User's Manual

32

RX23W Group

User's Manual: Hardware

RENESAS 32-Bit MCU
RX Family／RX200 Series

All information contained in these materials, including products and product specifications, represents information on the product at the time of publication and is subject to change by Renesas Electronics Corp. without notice. Please review the latest information published by Renesas Electronics Corp. through various means, including the Renesas Electronics Corp. website (http://www.renesas.com).

Renesas Electronics
www.renesas.com

Rev.1.20 Apr 2023

Downloaded from Arrow.com.

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
6. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.
   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.
   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.
7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (“Vulnerability Issues”). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
8. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
14. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.5.0-1 October 2020)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Contact Information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:
www.renesas.com/contact/

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation.
All trademarks and registered trademarks are the property of their respective owners.
SuperFlash® is a registered trademark of Silicon Storage Technology, Inc. in several countries including the United States and Japan.
Bluetooth® is a trademark of the Bluetooth SIG, Inc.

© 2023 Renesas Electronics Corporation. All rights reserved.

Downloaded from Arrow.com.

General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Precaution against Electrostatic Discharge (ESD)
   A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.
2. Processing at power-on
   The state of the product is undefined at the time when power is supplied. The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the time when power is supplied. In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the time when power is supplied until the reset process is completed. In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the time when power is supplied until the power reaches the level at which resetting is specified.
3. Input of signal during power-off state
   Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.
4. Handling of unused pins
   Handle unused pins in accordance with the directions given under handling of unused pins in the manual. The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of the LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible.
5. Clock signals
   After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is stabilized. When the clock signal is generated with an external resonator or from an external oscillator during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Additionally, when switching to a clock signal produced with an external resonator or by an external oscillator while program execution is in progress, wait until the target clock signal is stable.
6. Voltage application waveform at input pin
   Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (Max.) and V_{IH} (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (Max.) and V_{IH} (Min.).
7. Prohibition of access to reserved addresses
   Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.
8. Differences between products
   Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

Downloaded from Arrow.com.

How to Use This Manual

1. Purpose and Target Readers

This manual is designed to provide the user with an understanding of the hardware functions and electrical characteristics of the MCU. It is intended for users designing application systems incorporating the MCU. A basic knowledge of electric circuits, logical circuits, and MCUs is necessary in order to use this manual.
The manual comprises an overview of the product, descriptions of the CPU, system control functions, peripheral functions, electrical characteristics, and usage notes.

Particular attention should be paid to the precautionary notes when using the manual. These notes can be found within the body of the text, at the end of each section, and in the Usage Notes section.

The revision history summarizes the locations of revisions and additions. It does not list all revisions. Refer to the text of the manual for details.

The following documents apply to the RX23W Group. Make sure to refer to the latest versions of these documents. The latest versions of the listed documents are available from the Renesas Electronics website.

Document Type	Description	Document Title	Document No.
Datasheet	Hardware overview and electrical characteristics	RX23W Group Datasheet	R01DS0342EJ
User's Manual: Hardware	Hardware specifications (pin assignments, memory maps, peripheral specifications, electrical characteristics, and timing charts) and operation description Note: Refer to the application notes for details on using peripheral functions.	RX23W Group User's Manual: Hardware	This User's manual
User's Manual: Software	Description of CPU instruction set	RX Family RXv2 Instruction Set Architecture User's Manual: Software	R01US0071EJ
Application Note	Notes on Printed Circuit Board Patterns	RX Family Hardware Design Guide	R01AN1411EJ
	Information on using peripheral functions and application examples Sample programs	Available from Renesas Electronics website.	
Renesas Technical Update	Product specifications, updates on documents, etc.		

Downloaded from Arrow.com.

2. Notation of Numbers and Symbols

Each register description includes a bit chart, illustrating the arrangement of bits, and a table of bits, describing the meanings of the bit settings. The standard format and notation for bit charts and tables are described below.

X.X.X ... Register

Address(es): xxxx xxxxh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	...[1:0]		...4	—	—	—	...0
Value after reset:	x	0	0	0	0	0	0	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	...0	.....	0: ..... 1: Setting prohibited (3)	R/W (1)
b3 to b1	—	Reserved (2)	These bits are read as 0. The write value should be 0.	R/W
b4	...4	.....	0: 1:	R
b6, b5	...[1:0]	.....	0 0: ..... 0 1: ..... Settings other than above are prohibited. (3)	R/(W)*1
b7	—	Reserved	The read value is undefined. Writing to this bit has no effect.	R

(1) R/W: The bit or field is readable and writable.
R/(W): The bit or field is readable and writable. However, writing to this bit or field has some limitations. For details on the limitations, see the description or notes of respective registers.
R: The bit or field is readable. Writing to this bit or field has no effect.

(2) Reserved.
Use the specified value when writing to this bit or field; otherwise, the correct operation is not guaranteed.

(3) Setting prohibited. The correct operation is not guaranteed if such a setting is performed.

Downloaded from Arrow.com.

3. List of Abbreviations and Acronyms

Abbreviation	Full Form
ACIA	Asynchronous Communications Interface Adapter
bps	bits per second
CRC	Cyclic Redundancy Check
DMA	Direct Memory Access
DMAC	Direct Memory Access Controller
GSM	Global System for Mobile Communications
Hi-Z	High Impedance
IEBus	Inter Equipment Bus
I/O	Input / Output
IrDA	Infrared Data Association
LSB	Least Significant Bit
MSB	Most Significant Bit
NC	Non-Connect
PLL	Phase Locked Loop
PWM	Pulse Width Modulation
SFR	Special Function Registers
SIM	Subscriber Identity Module
UART	Universal Asynchronous Receiver / Transmitter
VCO	Voltage Controlled Oscillator

Caution: This product uses SuperFlash® technology licensed from Silicon Storage Technology, Inc.
The use of **Bluetooth**® in this product is under license from the Bluetooth SIG, Inc.

Downloaded from Arrow.com.

Contents

Features 51

1. Overview 52

1.1 Outline of Specifications 52

1.2 List of Products 58

1.3 Block Diagram 59

1.4 Pin Functions 61

1.5 Pin Assignments 65

1.5.1 85-Pin BGA 65

1.5.2 83-Pin LGA 66

1.5.3 56-Pin QFN 67

1.6 List of Pins and Pin Functions 68

1.6.1 85-Pin BGA 68

1.6.2 83-Pin LGA 70

1.6.3 56-Pin QFN 72

2. CPU 74

2.1 Features 74

2.2 Register Set of the CPU 75

2.2.1 General-Purpose Registers (R0 to R15) 76

2.2.2 Control Registers 76

2.2.2.1 Interrupt Stack Pointer (ISP)/User Stack Pointer (USP) 77

2.2.2.2 Exception Table Register (EXTB) 77

2.2.2.3 Interrupt Table Register (INTB) 77

2.2.2.4 Program Counter (PC) 77

2.2.2.5 Processor Status Word (PSW) 78

2.2.2.6 Backup PC (BPC) 79

2.2.2.7 Backup PSW (BPSW) 80

2.2.2.8 Fast Interrupt Vector Register (FINTV) 80

2.2.2.9 Floating-Point Status Word (FPSW) 81

2.2.3 Accumulator 83

2.3 Processor Mode 84

2.3.1 Supervisor Mode 84

2.3.2 User Mode 84

2.3.3 Privileged Instruction 84

2.3.4 Switching Between Processor Modes 84

2.4 Data Types 85

2.4.1 Integer 85

2.4.2 Floating-Points 86

2.4.3 Bitwise Operations 86

2.4.4 Strings 87

2.5 Endian 88

2.5.1 Switching the Endian 88

Downloaded from Arrow.com.

2.5.2 Access to I/O Registers 91
2.5.3 Notes on Access to I/O Registers 91
2.5.4 Data Arrangement 92
2.5.4.1 Data Arrangement in Registers 92
2.5.4.2 Data Arrangement in Memory 92
2.5.5 Notes on the Allocation of Instruction Codes 92
2.6 Vector Table 93
2.6.1 Exception Vector Table 93
2.6.2 Interrupt Vector Table 94
2.7 Operation of Instructions 95
2.7.1 Restrictions on RMPA and String-Manipulation Instructions 95
2.7.1.1 Transfer Size and Data Prefetching 95
2.7.1.2 Access to the External Space 95
2.7.1.3 Access to I/O Registers 95
2.8 Number of Cycles 96
2.8.1 Instruction and Number of Cycle 96
2.8.2 Numbers of Cycles for Response to Interrupts 100
3. Operating Modes 101
3.1 Operating Mode Types and Selection 101
3.2 Register Descriptions 102
3.2.1 Mode Monitor Register (MDMONR) 102
3.2.2 System Control Register 1 (SYSCR1) 103
3.3 Details of Operating Modes 104
3.3.1 Single-Chip Mode 104
3.3.2 Boot Mode 104
3.3.2.1 Boot Mode (USB Interface) 104
3.3.2.2 Boot Mode (SCI) 104
3.4 Transitions of Operating Modes 105
3.4.1 Operating Mode Transitions Determined by the Mode-Setting Pins 105
4. Address Space 106
4.1 Address Space 106
5. I/O Registers 108
5.1 I/O Register Addresses (Address Order) 110
6. Resets 141
6.1 Overview 141
6.2 Register Descriptions 143
6.2.1 Reset Status Register 0 (RSTSR0) 143
6.2.2 Reset Status Register 1 (RSTSR1) 144
6.2.3 Reset Status Register 2 (RSTSR2) 145
6.2.4 Software Reset Register (SWRR) 146
6.3 Operation 147

Downloaded from Arrow.com.

6.3.1 RES# Pin Reset 147
6.3.2 Power-On Reset and Voltage Monitoring 0 Reset 147
6.3.3 Voltage Monitoring 1 Reset 149
6.3.4 Independent Watchdog Timer Reset 150
6.3.5 Watchdog Timer Reset 150
6.3.6 Software Reset 150
6.3.7 Determination of Cold/Warm Start 151
6.3.8 Determination of Reset Generation Source 152
7. Option-Setting Memory (OFSM) 153
7.1 Overview 153
7.2 Register Descriptions 154
7.2.1 Option Function Select Register 0 (OFS0) 154
7.2.2 Option Function Select Register 1 (OFS1) 158
7.2.3 Endian Select Register (MDE) 159
7.3 Usage Note 160
7.3.1 Setting Example of Option-Setting Memory 160
8. Voltage Detection Circuit (LVDAb) 161
8.1 Overview 161
8.2 Register Descriptions 163
8.2.1 Voltage Monitoring 1 Circuit Control Register 1 (LVD1CR1) 163
8.2.2 Voltage Monitoring 1 Circuit Status Register (LVD1SR) 164
8.2.3 Voltage Monitoring Circuit Control Register (LVCMPCR) 165
8.2.4 Voltage Detection Level Select Register (LVDLVLR) 166
8.2.5 Voltage Monitoring 1 Circuit Control Register 0 (LVD1CR0) 167
8.3 VCC Input Voltage Monitor 168
8.3.1 Monitoring Vdet0 168
8.3.2 Monitoring Vdet1 168
8.4 Reset from Voltage Monitor 0 169
8.5 Interrupt and Reset from Voltage Monitoring 1 170
8.6 Event Link Output 172
8.6.1 Interrupt Handling and Event Linking 172
9. Clock Generation Circuit 173
9.1 Overview 173
9.2 Register Descriptions 178
9.2.1 System Clock Control Register (SCKCR) 178
9.2.2 System Clock Control Register 3 (SCKCR3) 180
9.2.3 PLL Control Register (PLLCR) 181
9.2.4 PLL Control Register 2 (PLLCR2) 182
9.2.5 USB-dedicated PLL Control Register (UPLLCR) 183
9.2.6 USB-dedicated PLL Control Register 2 (UPLLCR2) 184
9.2.7 Main Clock Oscillator Control Register (MOSCCR) 185

Downloaded from Arrow.com.

9.2.8 Sub-Clock Oscillator Control Register (SOSCCR) 186
9.2.9 Low-Speed On-Chip Oscillator Control Register (LOCOCR) 187
9.2.10 IWDT-Dedicated On-Chip Oscillator Control Register (ILOCOCR) 188
9.2.11 High-Speed On-Chip Oscillator Control Register (HOCOCR) 189
9.2.12 High-Speed On-Chip Oscillator Control Register 2 (HOCOCR2) 190
9.2.13 Oscillation Stabilization Flag Register (OSCOVFSR) 191
9.2.14 Oscillation Stop Detection Control Register (OSTDCR) 193
9.2.15 Oscillation Stop Detection Status Register (OSTDSR) 194
9.2.16 Main Clock Oscillator Wait Control Register (MOSCWTCR) 195
9.2.17 CLKOUT Output Control Register (CKOCR) 196
9.2.18 Main Clock Oscillator Forced Oscillation Control Register (MOFCR) 197
9.2.19 Memory Wait Cycle Setting Register (MEMWAIT) 198
9.2.20 Low-Speed On-Chip Oscillator Trimming Register (LOCOTRR) 200
9.2.21 IWDT-Dedicated On-Chip Oscillator Trimming Register (ILOCOTRR) 200
9.2.22 High-Speed On-Chip Oscillator Trimming Register n (HOCOTRRn) (n = 0, 3) 201
9.3 Main Clock Oscillator 202
9.3.1 Connecting a Crystal 202
9.3.2 External Clock Input 203
9.3.3 Handling of Pins When the Main Clock is Not Used 203
9.3.4 Notes on the External Clock Input 203
9.4 Sub-Clock Oscillator 204
9.4.1 Connecting 32.768-kHz Crystal 204
9.4.2 Handling of Pins When Sub-Clock is Not Used 205
9.5 Dedicated Clock Oscillator for Bluetooth 206
9.5.1 Connecting the Oscillator 206
9.5.2 Connecting the Bluetooth-Dedicated Clock Output Pin 206
9.6 Oscillation Stop Detection Function 208
9.6.1 Oscillation Stop Detection and Operation after Detection 208
9.6.2 Oscillation Stop Detection Interrupts 209
9.7 PLL Circuit 210
9.8 Internal Clock 210
9.8.1 System Clock 210
9.8.2 Peripheral Module Clock 210
9.8.3 FlashIF Clock 210
9.8.4 USB Clock 211
9.8.5 CAN Clock 211
9.8.6 CAC Clock 211
9.8.7 RTC-Dedicated Clock 211
9.8.8 IWDT-Dedicated Clock 211
9.8.9 SSI Clock 211
9.8.10 Clocks for BLE 211

Downloaded from Arrow.com.

9.8.11 Low-Power Timer Clock 211
9.9 Usage Notes 212
9.9.1 Notes on Clock Generation Circuit 212
9.9.2 Note on Rewriting the SCKCR3 Register 212
9.9.3 Notes on Resonator 212
9.9.4 Notes on Board Design 212
9.9.5 Notes on Resonator Connection Pins 214
9.9.6 Notes on Sub-Clock 214
10. Clock Frequency Accuracy Measurement Circuit (CAC) 218
10.1 Overview 218
10.2 Register Descriptions 220
10.2.1 CAC Control Register 0 (CACR0) 220
10.2.2 CAC Control Register 1 (CACR1) 221
10.2.3 CAC Control Register 2 (CACR2) 222
10.2.4 CAC Interrupt Request Enable Register (CAICR) 223
10.2.5 CAC Status Register (CASTR) 224
10.2.6 CAC Upper-Limit Value Setting Register (CAULVR) 225
10.2.7 CAC Lower-Limit Value Setting Register (CALLVR) 225
10.2.8 CAC Counter Buffer Register (CACNTBR) 225
10.3 Operation 226
10.3.1 Measuring Clock Frequency 226
10.3.2 Digital Filtering of Signals on the CACREF Pin 227
10.4 Interrupt Requests 227
10.5 Usage Notes 228
10.5.1 Module Stop Function Setting 228
11. Low Power Consumption 229
11.1 Overview 229
11.2 Register Descriptions 233
11.2.1 Standby Control Register (SBYCR) 233
11.2.2 Module Stop Control Register A (MSTPCRA) 234
11.2.3 Module Stop Control Register B (MSTPCRB) 235
11.2.4 Module Stop Control Register C (MSTPCRC) 237
11.2.5 Module Stop Control Register D (MSTPCRD) 238
11.2.6 Operating Power Control Register (OPCCR) 239
11.2.7 Sub Operating Power Control Register (SOPCCR) 240
11.2.8 Sleep Mode Return Clock Source Switching Register (RSTCKCR) 245
11.3 Reducing Power Consumption by Switching Clock Signals 247
11.4 Module Stop Function 247
11.5 Function for Lower Operating Power Consumption 247
11.5.1 Setting Operating Power Control Mode 247
11.6 Low Power Consumption Modes 249

Downloaded from Arrow.com.

11.6.1 Sleep Mode 249
11.6.1.1 Entry to Sleep Mode 249
11.6.1.2 Exit from Sleep Mode 250
11.6.1.3 Sleep Mode Return Clock Source Switching Function 250
11.6.2 Deep Sleep Mode 251
11.6.2.1 Entry to Deep Sleep Mode 251
11.6.2.2 Exit from Deep Sleep Mode 252
11.6.3 Software Standby Mode 253
11.6.3.1 Entry to Software Standby Mode 253
11.6.3.2 Exit from Software Standby Mode 254
11.6.3.3 Example of Software Standby Mode Application 255
11.7 Usage Notes 256
11.7.1 I/O Port States 256
11.7.2 Module Stop State of DMAC and DTC 256
11.7.3 On-Chip Peripheral Module Interrupts 256
11.7.4 Write Access to MSTPCRA, MSTPCRB, MSTPCRC, and MSTPCRD 256
11.7.5 Timing of WAIT Instructions 256
11.7.6 Rewrite the Register by DMAC and DTC in Sleep Mode 256
12. Battery Backup Function 257
12.1 Overview 257
12.2 Register Descriptions 258
12.2.1 VBATT Control Register (VBATTCR) 258
12.2.2 VBATT Status Register (VBATTSR) 259
12.2.3 VBATT Pin Voltage Drop Detection Interrupt Control Register (VBTLVDICR) 260
12.3 Operation 261
12.3.1 Battery Backup Function 261
12.3.2 VBATT Pin Voltage Monitoring Function 262
12.4 Usage Notes 263
13. Register Write Protection Function 264
13.1 Register Descriptions 265
13.1.1 Protect Register (PRCR) 265
14. Exception Handling 266
14.1 Exception Events 266
14.1.1 Undefined Instruction Exception 267
14.1.2 Privileged Instruction Exception 267
14.1.3 Access Exceptions 267
14.1.4 Floating-Point Exception 267
14.1.5 Reset 267
14.1.6 Non-Maskable Interrupt 267
14.1.7 Interrupt 267
14.1.8 Unconditional Trap 267

Downloaded from Arrow.com.

14.2 Exception Handling Procedure 268
14.3 Acceptance of Exception Events 270
14.3.1 Acceptance Timing and Saved PC Value 270
14.3.2 Vector and Site for Saving the Values in the PC and PSW 271
14.4 Hardware Processing for Accepting and Returning from Exceptions 272
14.5 Hardware Pre-Processing 273
14.5.1 Undefined Instruction Exception 273
14.5.2 Privileged Instruction Exception 273
14.5.3 Access Exceptions 273
14.5.4 Floating-Point Exception 273
14.5.5 Reset 273
14.5.6 Non-Maskable Interrupt 274
14.5.7 Interrupt 274
14.5.8 Unconditional Trap 274
14.6 Return from Exception Handling Routine 275
14.7 Priority of Exception Events 275
15. Interrupt Controller (ICUb) 276
15.1 Overview 276
15.2 Register Descriptions 278
15.2.1 Interrupt Request Register n (IRn) (n = interrupt vector number) 278
15.2.2 Interrupt Request Enable Register m (IERm) (m = 02h to 1Fh) 279
15.2.3 Interrupt Source Priority Register n (IPRn) (n = interrupt vector number) 280
15.2.4 Fast Interrupt Set Register (FIR) 281
15.2.5 Software Interrupt Generation Register (SWINTR) 282
15.2.6 DTC Transfer Request Enable Register n (DTCERn) (n = interrupt vector number) 283
15.2.7 DMAC Trigger Select Register m (DMRSRm) (m = DMAC channel number) 284
15.2.8 IRQ Control Register i (IRQCRi) (i = 0, 1, and 4 to 7) 285
15.2.9 IRQ Pin Digital Filter Enable Register 0 (IRQFLTE0) 286
15.2.10 IRQ Pin Digital Filter Setting Register 0 (IRQFLTC0) 287
15.2.11 Non-Maskable Interrupt Status Register (NMISR) 288
15.2.12 Non-Maskable Interrupt Enable Register (NMIER) 290
15.2.13 Non-Maskable Interrupt Status Clear Register (NMICLR) 292
15.2.14 NMI Pin Interrupt Control Register (NMICR) 293
15.2.15 NMI Pin Digital Filter Enable Register (NMIFLTE) 293
15.2.16 NMI Pin Digital Filter Setting Register (NMIFLTC) 294
15.3 Vector Table 295
15.3.1 Interrupt Vector Table 295
15.3.2 Fast Interrupt Vector Table 301
15.3.3 Non-maskable Interrupt Vector Area 301
15.4 Interrupt Operation 302

Downloaded from Arrow.com.

15.4.1 Detecting Interrupts 302
15.4.1.1 Operation of Status Flags for Edge-Detected Interrupts 302
15.4.1.2 Operation of Status Flags for Level-Detected Interrupts 304
15.4.2 Enabling and Disabling Interrupt Sources 305
15.4.3 Selecting Interrupt Request Destinations 306
15.4.4 Determining Priority 308
15.4.5 Multiple Interrupts 308
15.4.6 Fast Interrupt 308
15.4.7 Digital Filter 309
15.4.8 External Pin Interrupts 309
15.5 Non-maskable Interrupt Operation 310
15.6 Return from Low Power Consumption States 311
15.6.1 Return from Sleep Mode or Deep Sleep Mode 311
15.6.2 Return from Software Standby Mode 311
15.7 Usage Note 312
15.7.1 Note on WAIT Instruction Used with Non-Maskable Interrupt 312
16. Buses 313
16.1 Overview 313
16.2 Description of Buses 315
16.2.1 CPU Buses 315
16.2.2 Memory Buses 315
16.2.3 Internal Main Buses 315
16.2.4 Internal Peripheral Buses 316
16.2.5 Write Buffer Function (Internal Peripheral Bus) 317
16.2.6 Parallel Operation 318
16.3 Register Descriptions 319
16.3.1 Bus Error Status Clear Register (BERCLR) 319
16.3.2 Bus Error Monitoring Enable Register (BEREN) 319
16.3.3 Bus Error Status Register 1 (BERSR1) 320
16.3.4 Bus Error Status Register 2 (BERSR2) 320
16.3.5 Bus Priority Control Register (BUSPRI) 321
16.4 Bus Error Monitoring Section 323
16.4.1 Types of Bus Error 323
16.4.1.1 Illegal Address Access 323
16.4.1.2 Timeout 323
16.4.2 Operations When a Bus Error Occurs 324
16.4.3 Conditions Leading to Bus Errors 324
16.5 Interrupt 325
16.5.1 Interrupt Source 325
17. Memory-Protection Unit (MPU) 326
17.1 Overview 326

Downloaded from Arrow.com.

17.1.1 Types of Access Control 328
17.1.2 Regions for Access Control 328
17.1.3 Background Region 328
17.1.4 Overlap between Regions 328
17.1.5 Instructions and Data that Span Regions 328
17.2 Register Descriptions 329
17.2.1 Region-n Start Page Number Register (RSPAGEn) (n = 0 to 7) 329
17.2.2 Region-n End Page Number Register (REPAGEn) (n = 0 to 7) 330
17.2.3 Memory-Protection Enable Register (MPEN) 331
17.2.4 Background Access Control Register (MPBAC) 332
17.2.5 Memory-Protection Error Status-Clearing Register (MPECLR) 333
17.2.6 Memory-Protection Error Status Register (MPESTS) 334
17.2.7 Data Memory-Protection Error Address Register (MPDEA) 335
17.2.8 Region Search Address Register (MPSA) 335
17.2.9 Region Search Operation Register (MPOPS) 336
17.2.10 Region Invalidation Operation Register (MPOPI) 336
17.2.11 Instruction-Hit Region Register (MHITI) 337
17.2.12 Data-Hit Region Register (MHITD) 339
17.3 Functions 341
17.3.1 Memory Protection 341
17.3.2 Region Search 341
17.3.3 Protection of Registers Related to the Memory-Protection Unit 341
17.3.4 Flow for Determination of Access by the Memory-Protection Function 342
17.4 Procedures for Using Memory Protection 344
17.4.1 Setting Access-Control Information 344
17.4.2 Enabling Memory Protection 344
17.4.3 Transition to User Mode 344
17.4.4 Processing in Response to Memory-Protection Errors 344
18. DMA Controller (DMACA) 346
18.1 Overview 346
18.2 Register Descriptions 348
18.2.1 DMA Source Address Register (DMSAR) 348
18.2.2 DMA Destination Address Register (DMDAR) 348
18.2.3 DMA Transfer Count Register (DMCRA) 349
18.2.4 DMA Block Transfer Count Register (DMCRB) 351
18.2.5 DMA Transfer Mode Register (DMTMD) 352
18.2.6 DMA Interrupt Setting Register (DMINT) 353
18.2.7 DMA Address Mode Register (DMAMD) 355
18.2.8 DMA Offset Register (DMOFR) 358
18.2.9 DMA Transfer Enable Register (DMCNT) 359
18.2.10 DMA Software Start Register (DMREQ) 360

Downloaded from Arrow.com.

18.2.11 DMA Status Register (DMSTS) 361
18.2.12 DMA Activation Source Flag Control Register (DMCSL) 363
18.2.13 DMA Module Activation Register (DMAST) 364
18.3 Operation 365
18.3.1 Transfer Mode 365
18.3.2 Extended Repeat Area Function 369
18.3.3 Address Update Function Using Offset 371
18.3.4 Activation Sources 375
18.3.5 Operation Timing 376
18.3.6 DMAC Execution Cycles 377
18.3.7 Activating the DMAC 378
18.3.8 Starting DMA Transfer 379
18.3.9 Registers during DMA Transfer 379
18.3.10 Channel Priority 380
18.4 Ending DMA Transfer 381
18.4.1 Transfer End by Completion of Specified Total Number of Transfer Operations 381
18.4.2 Transfer End by Repeat Size End Interrupt 381
18.4.3 Transfer End by Interrupt on Extended Repeat Area Overflow 382
18.5 Interrupts 383
18.6 Event Link Function 384
18.7 Low Power Consumption Function 385
18.8 Usage Notes 386
18.8.1 DMA Transfer to Peripheral Modules 386
18.8.2 Access to the Registers during DMA Transfer 386
18.8.3 DMA Transfer to Reserved Areas 386
18.8.4 Interrupt Request by the DMA Activation Source Flag Control Register (DMCSL) at the End of each Transfer 386
18.8.5 Setting of DMAC Activation Source Select Register of the Interrupt Controller (ICU.DMRSRm) 386
18.8.6 Suspending or Restarting DMA Activation 386
19. Data Transfer Controller (DTCa) 387
19.1 Overview 387
19.2 Register Descriptions 389
19.2.1 DTC Mode Register A (MRA) 389
19.2.2 DTC Mode Register B (MRB) 390
19.2.3 DTC Transfer Source Register (SAR) 391
19.2.4 DTC Transfer Destination Register (DAR) 391
19.2.5 DTC Transfer Count Register A (CRA) 392
19.2.6 DTC Transfer Count Register B (CRB) 393
19.2.7 DTC Control Register (DTCCR) 393
19.2.8 DTC Vector Base Register (DTCVBR) 394
19.2.9 DTC Address Mode Register (DTCADMOD) 394

Downloaded from Arrow.com.

19.2.10 DTC Module Start Register (DTCST) 395
19.2.11 DTC Status Register (DTCSTS) 396
19.3 Request Sources 397
19.3.1 Allocating Transfer Information and DTC Vector Table 397
19.4 Operation 399
19.4.1 Transfer Information Read Skip Function 401
19.4.2 Transfer Information Write-Back Skip Function 402
19.4.3 Normal Transfer Mode 403
19.4.4 Repeat Transfer Mode 404
19.4.5 Block Transfer Mode 405
19.4.6 Chain Transfer 406
19.4.7 Operation Timing 407
19.4.8 Execution Cycles of the DTC 410
19.4.9 DTC Bus Mastership Release Timing 410
19.5 DTC Setting Procedure 411
19.6 Examples of DTC Usage 412
19.6.1 Normal Transfer 412
19.6.2 Chain Transfer When the Counter is 0 413
19.7 Interrupt Source 414
19.8 Event Link 414
19.9 Low Power Consumption Function 415
19.10 Usage Notes 416
19.10.1 Start Address of Transfer Information 416
19.10.2 Allocating Transfer Information 416
19.10.3 Setting the DTC Transfer Request Enable Register in the Interrupt Controller (ICU.DTCERn) 417
20. Event Link Controller (ELC) 418
20.1 Overview 418
20.2 Register Descriptions 419
20.2.1 Event Link Control Register (ELCR) 419
20.2.2 Event Link Setting Register n (ELSRn) (n = 1 to 4, 7, 8, 10, 12, 14 to 16, 18 to 29) 420
20.2.3 Event Link Option Setting Register A (ELOPA) 423
20.2.4 Event Link Option Setting Register B (ELOPB) 423
20.2.5 Event Link Option Setting Register C (ELOPC) 424
20.2.6 Event Link Option Setting Register D (ELOPD) 424
20.2.7 Port Group Setting Register n (PGRn) (n = 1, 2) 425
20.2.8 Port Group Control Register n (PGCn) (n = 1, 2) 426
20.2.9 Port Buffer Register n (PDBFn) (n = 1, 2) 427
20.2.10 Event Link Port Setting Register m (PELm) (m = 0 to 3) 428
20.2.11 Event Link Software Event Generation Register (ELSEGR) 429
20.3 Operation 430

Downloaded from Arrow.com.

20.3.1 Relation between Interrupt Handling and Event Linking 430
20.3.2 Event Linkage 431
20.3.3 Operation of Peripheral Timer Modules When Event Signal is Input 432
20.3.4 Operation of CTSU When Event Signal is Input 432
20.3.5 Operation of A/D and D/A Converters When Event Signal is Input 432
20.3.6 I/O Port Operation When Event Signal is Input and Event Generation 432
20.3.7 Example of Procedure for Linking Events 436
20.4 Usage Notes 437
20.4.1 Setting ELSRn Register 437
20.4.2 Setting Bit-Rotating Operation of Output Port Groups 437
20.4.3 Linking DMA/DTC Transfer End Signal as Event 437
20.4.4 Clock Settings 437
20.4.5 Module Stop Function Setting 437
21. I/O Ports 438
21.1 Overview 438
21.2 I/O Port Configuration 440
21.3 Register Descriptions 449
21.3.1 Port Direction Register (PDR) 449
21.3.2 Port Output Data Register (PODR) 450
21.3.3 Port Input Data Register (PIDR) 451
21.3.4 Port Mode Register (PMR) 452
21.3.5 Open Drain Control Register 0 (ODR0) 453
21.3.6 Open Drain Control Register 1 (ODR1) 454
21.3.7 Pull-Up Control Register (PCR) 455
21.3.8 Drive Capacity Control Register (DSCR) 456
21.4 Initialization of the Port Direction Register (PDR) 457
21.5 Handling of Unused Pins 458
22. Multi-Function Pin Controller (MPC) 459
22.1 Overview 459
22.2 Register Descriptions 465
22.2.1 Write-Protect Register (PWPR) 465
22.2.2 P0n Pin Function Control Register (P0nPFS) (n = 3, 5, 7) 466
22.2.3 P1n Pin Function Control Registers (P1nPFS) (n = 4 to 7) 467
22.2.4 P2n Pin Function Control Register (P2nPFS) (n = 1, 2, 5 to 7) 468
22.2.5 P3n Pin Function Control Registers (P3nPFS) (n = 0, 1) 469
22.2.6 P4n Pin Function Control Registers (P4nPFS) (n = 0 to 7) 470
22.2.7 PBn Pin Function Control Registers (PBnPFS) (n = 0, 1, 3, 5, 7) 471
22.2.8 PCn Pin Function Control Registers (PCnPFS) (n = 0, 2 to 7) 472
22.2.9 PDn Pin Function Control Registers (PDnPFS) (n = 3) 473
22.2.10 PEn Pin Function Control Registers (PEnPFS) (n = 0 to 4) 474
22.2.11 PJn Pin Function Control Registers (PJnPFS) (n = 3) 476

Downloaded from Arrow.com.

22.3 Usage Notes 477
22.3.1 Procedure for Specifying Input/Output Pin Function 477
22.3.2 Notes on MPC Register Setting 477
22.3.3 Note on Using Analog Functions 478
22.3.4 Notes on Using the CTSU Function of the Capacitive Touch Sensing Unit 478
23. Multi-Function Timer Pulse Unit 2 (MTU2a) 479
23.1 Overview 479
23.2 Register Descriptions 484
23.2.1 Timer Control Register (TCR) 484
23.2.2 Timer Mode Register (TMDR) 487
23.2.3 Timer I/O Control Register (TIOR) 489
23.2.4 Timer Interrupt Enable Register (TIER) 499
23.2.5 Timer Status Register (TSR) 501
23.2.6 Timer Buffer Operation Transfer Mode Register (TBTM) 502
23.2.7 Timer Input Capture Control Register (TICCR) 503
23.2.8 Timer A/D Converter Start Request Control Register (TADCR) 504
23.2.9 Timer A/D Converter Start Request Cycle Set Registers A and B (TADCORA and TADCORB) 505
23.2.10 Timer A/D Converter Start Request Cycle Set Buffer Registers A and B (TADCOBRA and TADCOBRB) 506
23.2.11 Timer Counter (TCNT) 506
23.2.12 Timer General Register (TGR) 507
23.2.13 Timer Start Register (TSTR) 508
23.2.14 Timer Synchronous Register (TSYR) 509
23.2.15 Timer Read/Write Enable Register (TRWER) 510
23.2.16 Timer Output Master Enable Register (TOER) 511
23.2.17 Timer Output Control Register 1 (TOCR1) 512
23.2.18 Timer Output Control Register 2 (TOCR2) 514
23.2.19 Timer Output Level Buffer Register (TOLBR) 516
23.2.20 Timer Gate Control Register (TGCR) 517
23.2.21 Timer Subcounter (TCNTS) 518
23.2.22 Timer Dead Time Data Register (TDDR) 518
23.2.23 Timer Cycle Data Register (TCDR) 519
23.2.24 Timer Cycle Buffer Register (TCBR) 519
23.2.25 Timer Interrupt Skipping Set Register (TITCR) 520
23.2.26 Timer Interrupt Skipping Counter (TITCNT) 521
23.2.27 Timer Buffer Transfer Set Register (TBTER) 522
23.2.28 Timer Dead Time Enable Register (TDER) 523
23.2.29 Timer Waveform Control Register (TWCR) 524
23.2.30 Noise Filter Control Registers (NFCR) 525
23.2.31 Bus Master Interface 526

Downloaded from Arrow.com.

23.3 Operation 527
23.3.1 Basic Functions 527
23.3.2 Synchronous Operation 533
23.3.3 Buffer Operation 535
23.3.4 Cascaded Operation 539
23.3.5 PWM Modes 544
23.3.6 Phase Counting Mode 548
23.3.7 Reset-Synchronized PWM Mode 554
23.3.8 Complementary PWM Mode 557
23.3.9 A/D Converter Start Request Delaying Function 588
23.3.10 Noise Filter 592
23.4 Interrupt Sources 593
23.4.1 Interrupt Sources and Priorities 593
23.4.2 DTC/DMAC Activation 594
23.4.3 A/D Converter Activation 594
23.5 Operation Timing 596
23.5.1 Input/Output Timing 596
23.5.2 Interrupt Signal Timing 602
23.6 Usage Notes 604
23.6.1 Module Clock Stop Mode Setting 604
23.6.2 Count Clock Restrictions 604
23.6.3 Notes on Cycle Setting 604
23.6.4 Contention between TCNT Write and Clear Operations 605
23.6.5 Contention between TCNT Write and Increment Operations 605
23.6.6 Contention between TGR Write Operation and Compare Match 606
23.6.7 Contention between Buffer Register Write Operation and Compare Match 606
23.6.8 Contention between Buffer Register Write and TCNT Clear Operations 607
23.6.9 Contention between TGR Read Operation and Input Capture 607
23.6.10 Contention between TGR Write Operation and Input Capture 608
23.6.11 Contention between Buffer Register Write Operation and Input Capture 608
23.6.12 Contention between MTU2.TCNT Write Operation and Overflow/Underflow in Cascaded Operation 609
23.6.13 Counter Value When Count Operation is Stopped in Complementary PWM Mode 610
23.6.14 Buffer Operation Setting in Complementary PWM Mode 610
23.6.15 Buffer Operation and Compare Match Flags in Reset-Synchronized PWM Mode 611
23.6.16 Overflow Flags in Reset-Synchronized PWM Mode 612
23.6.17 Contention between Overflow/Underflow and Counter Clearing 613
23.6.18 Contention between TCNT Write Operation and Overflow/Underflow 613
23.6.19 Notes on Transition from Normal Mode or PWM Mode 1 to Reset-Synchronized PWM Mode 614
23.6.20 Output Level in Complementary PWM Mode or Reset-Synchronized PWM Mode 614
23.6.21 Interrupts during Periods in the Module Stop State 614

Downloaded from Arrow.com.

23.6.22 Simultaneous Input Capture in MTU1.TCNT and MTU2.TCNT in Cascade Connection 614
23.6.23 Notes When Complementary PWM Mode Output Protection Functions are Not Used 615
23.6.24 Points for Caution to Prevent Malfunctions in Synchronous Clearing for Complementary PWM Mode 615
23.6.25 Continuous Output of Interrupt Signal in Response to a Compare Match 617
23.6.26 Usage Notes on A/D Converter Delaying Function in Complementary PWM Mode 617
23.7 MTU Output Pin Initialization 619
23.7.1 Operating Modes 619
23.7.2 Operation in Case of Re-Setting Due to Error during Operation 619
23.7.3 Overview of Pin Initialization Procedures and Mode Transitions in Case of Error during Operation 620
23.8 Operations Linked by the ELC 646
23.8.1 Event Signal Output to the ELC 646
23.8.2 MTU Operations in Response to Receiving Event Signals from the ELC 646
23.8.3 Notes on MTU by Event Signal Reception from the ELC 647
24. Port Output Enable 2 (POE2a) 648
24.1 Overview 648
24.2 Register Descriptions 651
24.2.1 Input Level Control/Status Register 1 (ICSR1) 651
24.2.2 Output Level Control/Status Register 1 (OCSR1) 653
24.2.3 Input Level Control/Status Register 2 (ICSR2) 654
24.2.4 Software Port Output Enable Register (SPOER) 655
24.2.5 Port Output Enable Control Register 1 (POECR1) 656
24.2.6 Port Output Enable Control Register 2 (POECR2) 657
24.2.7 Input Level Control/Status Register 3 (ICSR3) 658
24.3 Operation 659
24.3.1 Input Level Detection Operation 661
24.3.2 Output-Level Compare Operation 662
24.3.3 High-Impedance Control Using Registers 663
24.3.4 High-Impedance Control on Detection of Stopped Oscillation 663
24.3.5 High-Impedance Control in Response to Receiving an Event Signal from the ELC 663
24.3.6 Release from the High-Impedance 663
24.4 Interrupts 664
24.5 Usage Notes 664
24.5.1 Transitions to Software Standby Mode 664
24.5.2 When the POE Is Not Used 664
24.5.3 Specifying Pins Corresponding to the MTU 664
24.5.4 Notes on High-Impedance Control by Event Signal Reception from the ELC 664
25. 16-Bit Timer Pulse Unit (TPUa) 665
25.1 Overview 665
25.2 Register Descriptions 669

Downloaded from Arrow.com.

25.2.1 Timer Control Register (TCR) 669
25.2.2 Timer Mode Register (TMDR) 673
25.2.3 Timer I/O Control Register (TIORH, TIORL, TIOR) 674
25.2.4 Timer Interrupt Enable Register (TIER) 682
25.2.5 Timer Status Register (TSR) 683
25.2.6 Timer Counter (TCNT) 686
25.2.7 Timer General Register A (TGRA), Timer General Register B (TGRB), Timer General Register C (TGRC), Timer General Register D (TGRD) 686
25.2.8 Timer Start Register (TSTR) 687
25.2.9 Timer Synchronous Register (TSYR) 688
25.2.10 Noise Filter Control Register (NFCR) 689
25.3 Operation 691
25.3.1 Basic Functions 691
25.3.2 Synchronous Operation 697
25.3.3 Buffer Operation 699
25.3.4 Cascaded Operation 702
25.3.5 PWM Modes 704
25.3.6 Phase Counting Mode 709
25.3.6.1 Phase Counting Mode Application Example 714
25.3.7 Noise Filters 715
25.4 Interrupt Sources 716
25.5 DTC Activation 717
25.6 DMAC Activation 717
25.7 A/D Converter Activation 717
25.8 Operation Timing 718
25.8.1 Input/Output Timing 718
25.8.2 Interrupt Signal Timing 722
25.9 Usage Notes 724
25.9.1 Module Stop Function Setting 724
25.9.2 Input Clock Restrictions 724
25.9.3 Notes on Cycle Setting 724
25.9.4 Conflict between TPUm.TCNT Write and Clear Operations 725
25.9.5 Conflict between TPUm.TCNT Write and Increment Operations 725
25.9.6 Conflict between TPUm.TGRy Write and Compare Match 726
25.9.7 Conflict between Buffer Register Write and Compare Match 726
25.9.8 Conflict between TPUm.TGRy Read and Input Capture 727
25.9.9 Conflict between TPUm.TGRy Write and Input Capture 727
25.9.10 Conflict between Buffer Register Write and Input Capture 728
25.9.11 TCNT Simultaneous Input Capture in Cascade Operation 728
25.9.12 Conflict between Overflow/Underflow and Counter Clearing 729
25.9.13 Conflict between TPUm.TCNT Write and Overflow/Underflow 730

Downloaded from Arrow.com.

25.9.14 Multiplexing of I/O Pins 730
25.9.15 Continuous Output of Compare-Match Pulse Interrupt Signal 731
25.9.16 Continuous Output of Input-Capture Pulse Interrupt Signal 732
25.9.17 Continuous Output of Underflow Pulse Interrupt Signal 733
26. 8-Bit Timer (TMR) 734
26.1 Overview 734
26.2 Register Descriptions 739
26.2.1 Timer Counter (TCNT) 739
26.2.2 Time Constant Register A (TCORA) 740
26.2.3 Time Constant Register B (TCORB) 740
26.2.4 Timer Control Register (TCR) 741
26.2.5 Timer Counter Control Register (TCCR) 742
26.2.6 Timer Control/Status Register (TCSR) 744
26.2.7 Timer Counter Start Register (TCSTR) 746
26.3 Operation 747
26.3.1 Pulse Output 747
26.3.2 External Counter Reset Input 748
26.4 Operation Timing 749
26.4.1 TCNT Count Timing 749
26.4.2 Timing of Interrupt Signal Output on a Compare Match 750
26.4.3 Timing of Timer Output Signal at Compare Match 750
26.4.4 Timing of Counter Clear by Compare Match 751
26.4.5 Timing of the External Reset for TCNT 751
26.4.6 Timing of Interrupt Signal Output on an Overflow 752
26.5 Operation with Cascaded Connection 753
26.5.1 16-Bit Count Mode 753
26.5.2 Compare Match Count Mode 753
26.6 Interrupt Sources 754
26.6.1 Interrupt Sources and DTC Activation 754
26.7 Link Operation by ELC 755
26.7.1 Event Signal Output to ELC 755
26.7.2 TMR Operation when Receiving an Event Signal from ELC 755
26.7.3 Notes on Operating TMR According to an Event Signal from ELC 756
26.8 Usage Notes 757
26.8.1 Module Stop State Setting 757
26.8.2 Notes on Setting Cycle 757
26.8.3 Conflict between TCNT Write and Counter Clear 757
26.8.4 Conflict between TCNT Write and Increment 758
26.8.5 Conflict between TCORA or TCORB Write and Compare Match 758
26.8.6 Conflict between Compare Matches A and B 759
26.8.7 Switching of Internal Clocks and TCNT Operation 759

Downloaded from Arrow.com.

26.8.8 Clock Source Setting with Cascaded Connection 761
26.8.9 Continuous Output of Compare Match Interrupt Signal 761
27. Compare Match Timer (CMT) 762
27.1 Overview 762
27.2 Register Descriptions 763
27.2.1 Compare Match Timer Start Register 0 (CMSTR0) 763
27.2.2 Compare Match Timer Start Register 1 (CMSTR1) 763
27.2.3 Compare Match Timer Control Register (CMCR) 764
27.2.4 Compare Match Counter (CMCNT) 765
27.2.5 Compare Match Constant Register (CMCOR) 765
27.3 Operation 766
27.3.1 Periodic Count Operation 766
27.3.2 CMCNT Count Timing 766
27.4 Interrupts 767
27.4.1 Interrupt Sources 767
27.4.2 Timing of Compare Match Interrupt Generation 767
27.5 Link Operations by ELC 768
27.5.1 Event Signal Output to ELC 768
27.5.2 CMT Operation When Receiving an Event Signal from ELC 768
27.5.3 Notes on Operating CMT According to an Event Signal from ELC 768
27.6 Usage Notes 769
27.6.1 Setting the Module Stop Function 769
27.6.2 Conflict between CMCNT Counter Writing and Compare Match 769
27.6.3 Conflict between CMCNT Counter Writing and Incrementing 769
28. Realtime Clock (RTCe) 770
28.1 Overview 770
28.2 Register Descriptions 772
28.2.1 64-Hz Counter (R64CNT) 772
28.2.2 Second Counter (RSECCNT)/Binary Counter 0 (BCNT0) 773
28.2.3 Minute Counter (RMINCNT)/Binary Counter 1 (BCNT1) 774
28.2.4 Hour Counter (RHRCNT)/Binary Counter 2 (BCNT2) 775
28.2.5 Day-of-Week Counter (RWKCNT)/Binary Counter 3 (BCNT3) 776
28.2.6 Date Counter (RDAYCNT) 777
28.2.7 Month Counter (RMONCNT) 778
28.2.8 Year Counter (RYRCNT) 779
28.2.9 Second Alarm Register (RSECAR)/Binary Counter 0 Alarm Register (BCNT0AR) 780
28.2.10 Minute Alarm Register (RMINAR)/Binary Counter 1 Alarm Register (BCNT1AR) 781
28.2.11 Hour Alarm Register (RHRAR)/Binary Counter 2 Alarm Register (BCNT2AR) 782
28.2.12 Day-of-Week Alarm Register (RWKAR)/Binary Counter 3 Alarm Register (BCNT3AR) 783
28.2.13 Date Alarm Register (RDAYAR)/Binary Counter 0 Alarm Enable Register (BCNT0AER) 784

Downloaded from Arrow.com.

28.2.14 Month Alarm Register (RMONAR)/Binary Counter 1 Alarm Enable Register (BCNT1AER) 785
28.2.15 Year Alarm Register (RYRAR)/Binary Counter 2 Alarm Enable Register (BCNT2AER) 786
28.2.16 Year Alarm Enable Register (RYRAREN)/Binary Counter 3 Alarm Enable Register (BCNT3AER) 787
28.2.17 RTC Control Register 1 (RCR1) 788
28.2.18 RTC Control Register 2 (RCR2) 789
28.2.19 RTC Control Register 3 (RCR3) 791
28.2.19.1 Notes on using a low CL crystal unit 791
28.2.20 Time Error Adjustment Register (RADJ) 792
28.2.21 Time Capture Control Register n (RTCCRn) (n = 0, 1) 793
28.2.22 Second Capture Register n (RSECCPn) (n = 0, 1)/BCNT0 Capture Register n (BCNT0CPn) (n = 0, 1) 795
28.2.23 Minute Capture Register n (RMINCPn) (n = 0, 1)/BCNT1 Capture Register n (BCNT1CPn) (n = 0, 1) 796
28.2.24 Hour Capture Register n (RHRCPn) (n = 0, 1)/BCNT2 Capture Register n (BCNT2CPn) (n = 0, 1) 797
28.2.25 Date Capture Register n (RDAYCPn) (n = 0, 1)/BCNT3 Capture Register n (BCNT3CPn) (n = 0, 1) 798
28.2.26 Month Capture Register n (RMONCPn) (n = 0, 1) 799
28.3 Operation 800
28.3.1 Outline of Initial Settings of Registers after Power On 800
28.3.2 Clock and Count Mode Setting Procedure 801
28.3.3 Setting the Time 802
28.3.4 30-Second Adjustment 802
28.3.5 Reading 64-Hz Counter and Time 803
28.3.6 Alarm Function 804
28.3.7 Procedure for Disabling Alarm Interrupt 805
28.3.8 Time Error Adjustment Function 805
28.3.8.1 Automatic Adjustment 805
28.3.8.2 Adjustment by Software 806
28.3.8.3 Procedure for Changing the Mode of Adjustment 807
28.3.8.4 Procedure for Stopping Adjustment 807
28.3.9 Time Capture Function 808
28.4 Interrupt Sources 809
28.5 Event Link Output 811
28.5.1 Interrupt Handling and Event Linking 811
28.6 Usage Notes 812
28.6.1 Register Writing during Counting 812
28.6.2 Use of Periodic Interrupts 812
28.6.3 RTCOUT (1-Hz/64-Hz) Clock Output 812
28.6.4 Transitions to Low Power Consumption Modes after Setting Registers 813
28.6.5 Notes on Writing to and Reading from Registers 813

Downloaded from Arrow.com.

28.6.6 Changing the Count Mode ... 813
28.6.7 Initialization Procedure When the Realtime Clock is Not to be Used ... 814
29. Low-Power Timer (LPT) ... 815
29.1 Overview ... 815
29.2 Register Descriptions ... 816
29.2.1 Low-Power Timer Control Register 1 (LPTCR1) ... 816
29.2.2 Low-Power Timer Control Register 2 (LPTCR2) ... 817
29.2.3 Low-Power Timer Control Register 3 (LPTCR3) ... 818
29.2.4 Low-Power Timer Period Setting Register (LPTPRD) ... 819
29.2.5 Low-Power Timer Compare Register 0 (LPCMR0) ... 822
29.2.6 Low-Power Timer Standby Wakeup Enable Register (LPWUCR) ... 823
29.3 Operation ... 824
29.3.1 Periodic Counting Operation ... 824
29.3.2 Count Timing of Low-Power Timer Counter ... 826
29.3.3 Clearing Timing of Low-Power Timer Counter ... 826
29.4 Wakeup from Software Standby Mode by an Interrupt through the Event Link Controller (ELC) ... 827
29.5 Usage Notes ... 827
29.5.1 Notes on Transition to Software Standby Mode ... 827
30. Watchdog Timer (WDTA) ... 828
30.1 Overview ... 828
30.2 Register Descriptions ... 829
30.2.1 WDT Refresh Register (WDTRR) ... 829
30.2.2 WDT Control Register (WDTCR) ... 830
30.2.3 WDT Status Register (WDTSR) ... 833
30.2.4 WDT Reset Control Register (WDTRCR) ... 834
30.2.5 Option Function Select Register 0 (OFS0) ... 834
30.3 Operation ... 835
30.3.1 Count Operation in Each Start Mode ... 835
30.3.1.1 Register Start Mode ... 835
30.3.1.2 Auto-Start Mode ... 837
30.3.2 Control over Writing to the WDTCR and WDTRCR Registers ... 839
30.3.3 Refresh Operation ... 839
30.3.4 Reset Output ... 840
30.3.5 Interrupt Source ... 840
30.3.6 Reading the Down-Counter Value ... 841
30.3.7 Correspondence between Option Function Select Register 0 (OFS0) and WDT Registers ... 841
31. Independent Watchdog Timer (IWDTa) ... 842
31.1 Overview ... 842
31.2 Register Descriptions ... 844
31.2.1 IWDT Refresh Register (IWDTRR) ... 844

Downloaded from Arrow.com.

31.2.2 IWDT Control Register (IWDTCR) 845
31.2.3 IWDT Status Register (IWDTSR) 848
31.2.4 IWDT Reset Control Register (IWDTRCR) 849
31.2.5 IWDT Count Stop Control Register (IWDTCSTPR) 850
31.2.6 Option Function Select Register 0 (OFS0) 850
31.3 Operation 851
31.3.1 Count Operation in Each Start Mode 851
31.3.1.1 Register Start Mode 851
31.3.1.2 Auto-Start Mode 853
31.3.2 Control over Writing to the IWDTCR, IWDTRCR, and IWDTCSTPR Registers 855
31.3.3 Refresh Operation 856
31.3.4 Status Flags 858
31.3.5 Reset Output 858
31.3.6 Interrupt Sources 858
31.3.7 Reading the Counter Value 859
31.3.8 Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers 860
31.4 Link Operation by ELC 860
31.5 Usage Notes 860
31.5.1 Refresh Operations 860
31.5.2 Clock Divide Ratio Setting 860
32. USB 2.0 Host/Function Module (USBc) 861
32.1 Overview 861
32.2 Register Descriptions 863
32.2.1 System Configuration Control Register (SYSCFG) 863
32.2.2 System Configuration Status Register 0 (SYSSTS0) 865
32.2.3 Device State Control Register 0 (DVSTCTR0) 866
32.2.4 CFIFO Port Register (CFIFO),
D0FIFO Port Register (D0FIFO),
D1FIFO Port Register (D1FIFO) 869
32.2.5 CFIFO Port Select Register (CFIFOSEL),
D0FIFO Port Select Register (D0FIFOSEL),
D1FIFO Port Select Register (D1FIFOSEL) 871
32.2.6 CFIFO Port Control Register (CFIFOCTR),
D0FIFO Port Control Register (D0FIFOCTR),
D1FIFO Port Control Register (D1FIFOCTR) 875
32.2.7 Interrupt Enable Register 0 (INTENB0) 877
32.2.8 Interrupt Enable Register 1 (INTENB1) 878
32.2.9 BRDY Interrupt Enable Register (BRDYENB) 879
32.2.10 NRDY Interrupt Enable Register (NRDYENB) 880
32.2.11 BEMP Interrupt Enable Register (BEMPENB) 881
32.2.12 SOF Output Configuration Register (SOFCFG) 882
32.2.13 Interrupt Status Register 0 (INTSTS0) 883
32.2.14 Interrupt Status Register 1 (INTSTS1) 886

Downloaded from Arrow.com.

32.2.15 BRDY Interrupt Status Register (BRDYSTS) 889
32.2.16 NRDY Interrupt Status Register (NRDYSTS) 890
32.2.17 BEMP Interrupt Status Register (BEMPSTS) 891
32.2.18 Frame Number Register (FRMNUM) 892
32.2.19 USB Request Type Register (USBREQ) 893
32.2.20 USB Request Value Register (USBVAL) 894
32.2.21 USB Request Index Register (USBINDX) 894
32.2.22 USB Request Length Register (USBLENG) 895
32.2.23 DCP Configuration Register (DCPCFG) 896
32.2.24 DCP Maximum Packet Size Register (DCPMAXP) 897
32.2.25 DCP Control Register (DCPCTR) 898
32.2.26 Pipe Window Select Register (PIPESEL) 901
32.2.27 Pipe Configuration Register (PIPECFG) 902
32.2.28 Pipe Maximum Packet Size Register (PIPEMAXP) 904
32.2.29 Pipe Cycle Control Register (PIPEPERI) 905
32.2.30 Pipe n Control Registers (PIPEnCTR) (n = 1 to 9) 906
32.2.31 Pipe n Transaction Counter Enable Register (PIPEnTRE) (n = 1 to 5) 914
32.2.32 Pipe n Transaction Counter Register (PIPEnTRN) (n = 1 to 5) 915
32.2.33 Device Address n Configuration Register (DEVADDn) (n = 0 to 5) 916
32.2.34 USB Module Control Register (USBMC) 917
32.2.35 BC Control Register 0 (USBBCCTRL0) 918
32.3 Operation 920
32.3.1 System Control 920
32.3.1.1 Setting Data to the USB Related Register 920
32.3.1.2 Controller Function Selection 920
32.3.1.3 Controlling USB Data Bus Resistors 920
32.3.1.4 Example of USB External Connection Circuit 921
32.3.2 Interrupt Sources 926
32.3.3 Interrupt Descriptions 928
32.3.3.1 BRDY Interrupt 928
32.3.3.2 NRDY Interrupt 932
32.3.3.3 BEMP Interrupt 935
32.3.3.4 Device State Transition Interrupt 936
32.3.3.5 Control Transfer Stage Transition Interrupt 937
32.3.3.6 Frame Update Interrupt 938
32.3.3.7 VBUS Interrupt 938
32.3.3.8 Resume Interrupt 938
32.3.3.9 OVRCR Interrupt 938
32.3.3.10 BCHG Interrupt 938
32.3.3.11 DTCH Interrupt 938
32.3.3.12 SACK Interrupt 939

Downloaded from Arrow.com.

32.3.3.13 SIGN Interrupt 939
32.3.3.14 ATTCH Interrupt 939
32.3.3.15 EOFERR Interrupt 939
32.3.3.16 Portable Device Detection Interrupt 939
32.3.4 Pipe Control 940
32.3.4.1 Pipe Control Register Switching Procedures 941
32.3.4.2 Transfer Types 941
32.3.4.3 Endpoint Number 941
32.3.4.4 Maximum Packet Size Setting 942
32.3.4.5 Transaction Counter (For Pipes 1 to 5 in Reading Direction) 942
32.3.4.6 Response PID 943
32.3.4.7 Data PID Sequence Bit 944
32.3.4.8 Response PID = NAK Function 944
32.3.4.9 Auto Response Mode 944
32.3.4.10 OUT-NAK Mode 944
32.3.4.11 Null Auto Response Mode 945
32.3.5 FIFO Buffer Memory 945
32.3.5.1 FIFO Buffer Memory 945
32.3.5.2 FIFO Buffer Clearing 946
32.3.5.3 FIFO Port Functions 947
32.3.5.4 DMA Transfers (D0FIFO and D1FIFO Ports) 948
32.3.6 Control Transfers Using DCP 949
32.3.6.1 Control Transfers When the Host Controller is Selected 949
32.3.6.2 Control Transfers When the Function Controller is Selected 950
32.3.7 Bulk Transfers (Pipes 1 to 5) 951
32.3.8 Interrupt Transfers (Pipes 6 to 9) 951
32.3.8.1 Interval Counter during Interrupt Transfers When the Host Controller is Selected 951
32.3.9 Isochronous Transfers (Pipes 1 and 2) 952
32.3.9.1 Error Detection in Isochronous Transfers 952
32.3.9.2 Data PID 953
32.3.9.3 Interval Counter 953
32.3.10 SOF Interpolation Function 960
32.3.11 Pipe Schedule 961
32.3.11.1 Conditions for Generating a Transaction 961
32.3.11.2 Transfer Schedule 961
32.3.11.3 Enabling USB Communication 961
32.4 Usage Notes 962
32.4.1 Setting the Module-Stop Function 962
32.5 Battery Charging Detection Processing 963
32.5.1 Processing When Function Controller is Selected 963
32.5.2 Processing When Host Controller is Selected 965

Downloaded from Arrow.com.

33. Serial Communications Interface (SCIg, SCIh) 968
33.1 Overview 968
33.2 Register Descriptions 976
33.2.1 Receive Shift Register (RSR) 976
33.2.2 Receive Data Register (RDR) 976
33.2.3 Receive Data Register H, L, HL (RDRH, RDRL, RDRHL) 977
33.2.4 Transmit Data Register (TDR) 978
33.2.5 Transmit Data Register H, L, HL (TDRH, TDRL, TDRHL) 979
33.2.6 Transmit Shift Register (TSR) 979
33.2.7 Serial Mode Register (SMR) 980
33.2.8 Serial Control Register (SCR) 984
33.2.9 Serial Status Register (SSR) 988
33.2.10 Smart Card Mode Register (SCMR) 993
33.2.11 Bit Rate Register (BRR) 995
33.2.12 Modulation Duty Register (MDDR) 1003
33.2.13 Serial Extended Mode Register (SEMR) 1004
33.2.14 Noise Filter Setting Register (SNFR) 1007
33.2.15 I2C Mode Register 1 (SIMR1) 1008
33.2.16 I2C Mode Register 2 (SIMR2) 1009
33.2.17 I2C Mode Register 3 (SIMR3) 1010
33.2.18 I2C Status Register (SISR) 1012
33.2.19 SPI Mode Register (SPMR) 1013
33.2.20 Extended Serial Module Enable Register (ESMER) 1014
33.2.21 Control Register 0 (CR0) 1015
33.2.22 Control Register 1 (CR1) 1015
33.2.23 Control Register 2 (CR2) 1016
33.2.24 Control Register 3 (CR3) 1017
33.2.25 Port Control Register (PCR) 1017
33.2.26 Interrupt Control Register (ICR) 1018
33.2.27 Status Register (STR) 1019
33.2.28 Status Clear Register (STCR) 1020
33.2.29 Control Field 0 Data Register (CF0DR) 1020
33.2.30 Control Field 0 Compare Enable Register (CF0CR) 1021
33.2.31 Control Field 0 Receive Data Register (CF0RR) 1021
33.2.32 Primary Control Field 1 Data Register (PCF1DR) 1021
33.2.33 Secondary Control Field 1 Data Register (SCF1DR) 1022
33.2.34 Control Field 1 Compare Enable Register (CF1CR) 1022
33.2.35 Control Field 1 Receive Data Register (CF1RR) 1022
33.2.36 Timer Control Register (TCR) 1023
33.2.37 Timer Mode Register (TMR) 1023
33.2.38 Timer Prescaler Register (TPRE) 1024

Downloaded from Arrow.com.

33.2.39 Timer Count Register (TCNT) 1024
33.3 Operation in Asynchronous Mode 1025
33.3.1 Serial Data Transfer Format 1025
33.3.2 Receive Data Sampling Timing and Reception Margin in Asynchronous Mode 1027
33.3.3 Clock 1028
33.3.4 Double-Speed Mode 1028
33.3.5 CTS and RTS Functions 1029
33.3.6 SCI Initialization (Asynchronous Mode) 1030
33.3.7 Serial Data Transmission (Asynchronous Mode) 1032
33.3.8 Serial Data Reception (Asynchronous Mode) 1036
33.4 Multi-Processor Communications Function 1040
33.4.1 Multi-Processor Serial Data Transmission 1041
33.4.2 Multi-Processor Serial Data Reception 1042
33.5 Operation in Clock Synchronous Mode 1045
33.5.1 Clock 1045
33.5.2 CTS and RTS Functions 1046
33.5.3 SCI Initialization (Clock Synchronous Mode) 1047
33.5.4 Serial Data Transmission (Clock Synchronous Mode) 1048
33.5.5 Serial Data Reception (Clock Synchronous Mode) 1052
33.5.6 Simultaneous Serial Data Transmission and Reception (Clock Synchronous Mode) 1055
33.6 Operation in Smart Card Interface Mode 1056
33.6.1 Sample Connection 1056
33.6.2 Data Format (Except in Block Transfer Mode) 1057
33.6.3 Block Transfer Mode 1058
33.6.4 Receive Data Sampling Timing and Reception Margin 1059
33.6.5 SCI Initialization (Smart Card Interface Mode) 1060
33.6.6 Serial Data Transmission (Except in Block Transfer Mode) 1062
33.6.7 Serial Data Reception (Except in Block Transfer Mode) 1065
33.6.8 Clock Output Control 1067
33.7 Operation in Simple I^2C Mode 1068
33.7.1 Generation of Start, Restart, and Stop Conditions 1069
33.7.2 Clock Synchronization 1071
33.7.3 SSDA Output Delay 1072
33.7.4 SCI Initialization (Simple I^2C Mode) 1073
33.7.5 Operation in Master Transmission (Simple I^2C Mode) 1074
33.7.6 Master Reception (Simple I^2C Mode) 1076
33.7.7 Recovery from Bus Hang-up 1078
33.8 Operation in Simple SPI Mode 1079
33.8.1 States of Pins in Master and Slave Modes 1080
33.8.2 SS Function in Master Mode 1080
33.8.3 SS Function in Slave Mode 1080

Downloaded from Arrow.com.

33.8.4 Relationship between Clock and Transmit/Receive Data 1081
33.8.5 SCI Initialization (Simple SPI Mode) 1082
33.8.6 Transmission and Reception of Serial Data (Simple SPI Mode) 1082
33.9 Bit Rate Modulation Function 1082
33.10 Extended Serial Mode Control Section: Description of Operation 1083
33.10.1 Serial Transfer Protocol 1083
33.10.2 Transmitting a Start Frame 1084
33.10.3 Receiving a Start Frame 1087
33.10.3.1 Priority Interrupt Bit 1092
33.10.4 Detection of Bus Collisions 1093
33.10.5 Digital Filter for Input on the RXDX12 Pin 1094
33.10.6 Bit Rate Measurement 1095
33.10.7 Selectable Timing for Sampling Data Received through RXDX12 1096
33.10.8 Timer 1097
33.11 Noise Cancellation Function 1099
33.12 Interrupt Sources 1100
33.12.1 Buffer Operations for TXI and RXI Interrupts 1100
33.12.2 Interrupts in Asynchronous Mode, Clock Synchronous Mode, and Simple SPI Mode 1100
33.12.3 Interrupts in Smart Card Interface Mode 1101
33.12.4 Interrupts in Simple I^2C Mode 1102
33.12.5 Interrupt Requests from the Extended Serial Mode Control Section 1103
33.13 Event Linking 1104
33.14 Usage Notes 1105
33.14.1 Setting the Module Stop Function 1105
33.14.2 Break Detection and Processing 1105
33.14.3 Mark State and Sending Breaks 1105
33.14.4 Receive Error Flags and Transmit Operations (Clock Synchronous Mode and Simple SPI Mode) 1105
33.14.5 Writing Data to the TDR Register 1105
33.14.6 Restrictions on Clock Synchronous Transmission (Clock Synchronous Mode and Simple SPI Mode) 1106
33.14.7 Restrictions on Using DMAC or DTC 1107
33.14.8 Notes on Starting Transfer 1107
33.14.9 SCI Operations during Low Power Consumption State 1107
33.14.10 External Clock Input in Clock Synchronous Mode and Simple SPI Mode 1110
33.14.11 Limitations on Simple SPI Mode 1111
33.14.12 Limitation 1 on Usage of the Extended Serial Mode Control Section 1112
33.14.13 Limitation 2 on Usage of the Extended Serial Mode Control Section 1112
33.14.14 Note on Transmit Enable Bit (TE Bit) 1113
33.14.15 Note on Stopping Reception When Using the RTS Function in Asynchronous Mode 1113
34. IrDA Interface 1114
34.1 Overview 1114

Downloaded from Arrow.com.

34.2 Register Descriptions 1115
34.2.1 IrDA Control Register (IRCR) 1115
34.3 Operation 1117
34.3.1 Transmission 1117
34.3.2 Reception 1118
34.3.3 Selecting High-Level Pulse Width 1118
34.4 Usage Notes 1119
34.4.1 Module Stop Function Setting 1119
34.4.2 SCI5 Setting 1119
34.4.3 Minimum Pulse-Width during Reception 1119
34.4.4 Notes on IrDA Initial Setting/Resetting 1119
35. I2C-bus Interface (RIICa) 1120
35.1 Overview 1120
35.2 Register Descriptions 1123
35.2.1 I2C-bus Control Register 1 (ICCR1) 1123
35.2.2 I2C-bus Control Register 2 (ICCR2) 1125
35.2.3 I2C-bus Mode Register 1 (ICMR1) 1129
35.2.4 I2C-bus Mode Register 2 (ICMR2) 1130
35.2.5 I2C-bus Mode Register 3 (ICMR3) 1132
35.2.6 I2C-bus Function Enable Register (ICFER) 1134
35.2.7 I2C-bus Status Enable Register (ICSER) 1136
35.2.8 I2C-bus Interrupt Enable Register (ICIER) 1138
35.2.9 I2C-bus Status Register 1 (ICSR1) 1140
35.2.10 I2C-bus Status Register 2 (ICSR2) 1143
35.2.11 Slave Address Register Ly (SARLy) (y = 0 to 2) 1146
35.2.12 Slave Address Register Uy (SARUy) (y = 0 to 2) 1147
35.2.13 I2C-bus Bit Rate Low-Level Register (ICBRL) 1148
35.2.14 I2C-bus Bit Rate High-Level Register (ICBRH) 1149
35.2.15 I2C-bus Transmit Data Register (ICDRT) 1151
35.2.16 I2C-bus Receive Data Register (ICDRR) 1151
35.2.17 I2C-bus Shift Register (ICDRS) 1151
35.3 Operation 1152
35.3.1 Communication Data Format 1152
35.3.2 Initial Settings 1153
35.3.3 Master Transmit Operation 1154
35.3.4 Master Receive Operation 1157
35.3.5 Slave Transmit Operation 1163
35.3.6 Slave Receive Operation 1166
35.4 SCL Synchronization Circuit 1168
35.5 SDA Output Delay Function 1169
35.6 Digital Noise Filters 1170

Downloaded from Arrow.com.

35.7 Address Match Detection 1171
35.7.1 Slave-Address Match Detection 1171
35.7.2 Detection of the General Call Address 1173
35.7.3 Device-ID Address Detection 1174
35.7.4 Host Address Detection 1176
35.8 Automatic Low-Hold Function for SCL 1177
35.8.1 Function to Prevent Wrong Transmission of Transmit Data 1177
35.8.2 NACK Reception Transfer Suspension Function 1178
35.8.3 Function to Prevent Failure to Receive Data 1179
35.9 Arbitration-Lost Detection Functions 1181
35.9.1 Master Arbitration-Lost Detection (MALE Bit) 1181
35.9.2 NACK Transmission Arbitration-Lost Detection Function (NALE Bit) 1183
35.9.3 Slave Arbitration-Lost Detection (SALE Bit) 1184
35.10 Start Condition/Restart Condition/Stop Condition Generating Function 1185
35.10.1 Generating a Start Condition 1185
35.10.2 Generating a Restart Condition 1185
35.10.3 Generating a Stop Condition 1186
35.11 Bus Hanging 1187
35.11.1 Timeout Function 1187
35.11.2 Additional SCL Output Function 1189
35.11.3 RIIC Reset and Internal Reset 1190
35.12 SMBus Operation 1191
35.12.1 SMBus Timeout Measurement 1191
35.12.2 Packet Error Code (PEC) 1192
35.12.3 SMBus Host Notification Protocol (Notify ARP Master Command) 1192
35.13 Interrupt Sources 1193
35.13.1 Buffer Operation for TXI and RXI Interrupts 1193
35.14 Initialization of Registers and Functions When a Reset is Applied or a Condition is Detected 1194
35.15 Event Link Function (Output) 1195
35.15.1 Interrupt Handling and Event Linking 1195
35.16 Usage Notes 1196
35.16.1 Setting Module Stop Function 1196
35.16.2 Notes on Starting Transfer 1196
36. CAN Module (RSCAN) 1197
36.1 Overview 1197
36.2 Register Descriptions 1200
36.2.1 Bit Configuration Register L (CFGL) 1200
36.2.2 Bit Configuration Register H (CFGH) 1201
36.2.3 Control Register L (CTRL) 1202
36.2.4 Control Register H (CTRH) 1204
36.2.5 Status Register L (STSL) 1206

Downloaded from Arrow.com.

36.2.6 Status Register H (STSH) 1207
36.2.7 Error Flag Register L (ERFLL) 1208
36.2.8 Error Flag Register H (ERFLH) 1210
36.2.9 Global Configuration Register L (GCFGL) 1211
36.2.10 Global Configuration Register H (GCFGH) 1212
36.2.11 Global Control Register L (GCTRL) 1213
36.2.12 Global Control Register H (GCTRH) 1214
36.2.13 Global Status Register (GSTS) 1214
36.2.14 Global Error Flag Register (GERFLL) 1215
36.2.15 Global Transmit Interrupt Status Register (GTINTSTS) 1216
36.2.16 Timestamp Register (GTSC) 1217
36.2.17 Receive Rule Number Configuration Register (GAFLCFG) 1217
36.2.18 Receive Rule Entry Register jAL (GAFLIDLj) (j = 0 to 15) 1218
36.2.19 Receive Rule Entry Register jAH (GAFLIDHj) (j = 0 to 15) 1219
36.2.20 Receive Rule Entry Register jBL (GAFLMLj) (j = 0 to 15) 1220
36.2.21 Receive Rule Entry Register jBH (GAFLMHj) (j = 0 to 15) 1221
36.2.22 Receive Rule Entry Register jCL (GAFLPLj) (j = 0 to 15) 1222
36.2.23 Receive Rule Entry Register jCH (GAFLPHj) (j = 0 to 15) 1223
36.2.24 Receive Buffer Number Configuration Register (RMNB) 1224
36.2.25 Receive Buffer Receive Complete Flag Register (RMND0) 1224
36.2.26 Receive Buffer Register nAL (RMIDLn) (n = 0 to 15) 1225
36.2.27 Receive Buffer Register nAH (RMIDHn) (n = 0 to 15) 1226
36.2.28 Receive Buffer Register nBL (RMTSn) (n = 0 to 15) 1227
36.2.29 Receive Buffer Register nBH (RMPTRn) (n = 0 to 15) 1228
36.2.30 Receive Buffer Register nCL (RMDF0n) (n = 0 to 15) 1229
36.2.31 Receive Buffer Register nCH (RMDF1n) (n = 0 to 15) 1229
36.2.32 Receive Buffer Register nDL (RMDF2n) (n = 0 to 15) 1230
36.2.33 Receive Buffer Register nDH (RMDF3n) (n = 0 to 15) 1230
36.2.34 Receive FIFO Control Register m (RFCCm) (m = 0, 1) 1231
36.2.35 Receive FIFO Status Register m (RFSTSm) (m = 0, 1) 1232
36.2.36 Receive FIFO Pointer Control Register m (RFPCTRm) (m = 0, 1) 1233
36.2.37 Receive FIFO Access Register mAL (RFIDLm) (m = 0, 1) 1234
36.2.38 Receive FIFO Access Register mAH (RFIDHm) (m = 0, 1) 1234
36.2.39 Receive FIFO Access Register mBL (RFTSm) (m = 0, 1) 1235
36.2.40 Receive FIFO Access Register mBH (RFPTRm) (m = 0, 1) 1235
36.2.41 Receive FIFO Access Register mCL (RFDF0m) (m = 0, 1) 1236
36.2.42 Receive FIFO Access Register mCH (RFDF1m) (m = 0, 1) 1236
36.2.43 Receive FIFO Access Register mDL (RFDF2m) (m = 0, 1) 1237
36.2.44 Receive FIFO Access Register mDH (RFDF3m) (m = 0, 1) 1237
36.2.45 Transmit/Receive FIFO Control Register 0L (CFCCL0) 1238
36.2.46 Transmit/Receive FIFO Control Register 0H (CFCCH0) 1240

Downloaded from Arrow.com.

36.2.47 Transmit/Receive FIFO Status Register 0 (CFSTS0) 1241
36.2.48 Transmit/Receive FIFO Pointer Control Register 0 (CFPCTR0) 1243
36.2.49 Transmit/Receive FIFO Access Register 0AL (CFIDL0) 1244
36.2.50 Transmit/Receive FIFO Access Register 0AH (CFIDH0) 1245
36.2.51 Transmit/Receive FIFO Access Register 0BL (CFTS0) 1246
36.2.52 Transmit/Receive FIFO Access Register 0BH (CFPTR0) 1247
36.2.53 Transmit/Receive FIFO Access Register 0CL (CFDF00) 1248
36.2.54 Transmit/Receive FIFO Access Register 0CH (CFDF10) 1248
36.2.55 Transmit/Receive FIFO Access Register 0DL (CFDF20) 1249
36.2.56 Transmit/Receive FIFO Access Register 0DH (CFDF30) 1249
36.2.57 Receive FIFO Message Lost Status Register (RFMSTS) 1250
36.2.58 Transmit/Receive FIFO Message Lost Status Register (CFMSTS) 1250
36.2.59 Receive FIFO Interrupt Status Register (RFISTS) 1251
36.2.60 Transmit/Receive FIFO Receive Interrupt Status Register (CFISTS) 1251
36.2.61 Transmit Buffer Control Register p (TMCp) (p = 0 to 3) 1252
36.2.62 Transmit Buffer Status Register p (TMSTSp) (p = 0 to 3) 1253
36.2.63 Transmit Buffer Transmit Request Status Register (TMTRSTS) 1254
36.2.64 Transmit Buffer Transmit Complete Status Register (TMTCSTS) 1255
36.2.65 Transmit Buffer Transmit Abort Status Register (TMTASTS) 1256
36.2.66 Transmit Buffer Interrupt Enable Register (TMIEC) 1257
36.2.67 Transmit Buffer Register pAL (TMIDLp) (p = 0 to 3) 1257
36.2.68 Transmit Buffer Register pAH (TMIDHp) (p = 0 to 3) 1258
36.2.69 Transmit Buffer Register pBH (TMPTRp) (p = 0 to 3) 1259
36.2.70 Transmit Buffer Register pCL (TMDF0p) (p = 0 to 3) 1260
36.2.71 Transmit Buffer Register pCH (TMDF1p) (p = 0 to 3) 1260
36.2.72 Transmit Buffer Register pDL (TMDF2p) (p = 0 to 3) 1261
36.2.73 Transmit Buffer Register pDH (TMDF3p) (p = 0 to 3) 1261
36.2.74 Transmit History Buffer Control Register (THLCC0) 1262
36.2.75 Transmit History Buffer Status Register (THLSTS0) 1263
36.2.76 Transmit History Buffer Access Register (THLACC0) 1264
36.2.77 Transmit History Buffer Pointer Control Register (THLPCTR0) 1265
36.2.78 Global RAM Window Control Register (GRWCR) 1266
36.2.79 Global Test Configuration Register (GTSTCFG) 1267
36.2.80 Global Test Control Register (GTSTCTRL) 1267
36.2.81 Global Test Protection Unlock Register (GLOCKK) 1268
36.2.82 RAM Test Register r (RPGACCr) (r = 0 to 127) 1268
36.3 CAN Modes 1269
36.3.1 Global Modes 1269
36.3.2 Channel Modes 1271
36.4 Reception Function 1275
36.4.1 Data Processing Using the Receive Rule Table 1275

Downloaded from Arrow.com.

36.4.2 Timestamp 1277
36.5 Transmission Functions 1278
36.5.1 Transmit Priority Determination 1278
36.5.2 Transmission Using Transmit Buffers 1279
36.5.3 Transmission Using FIFO Buffers 1279
36.5.4 Transmit History Function 1282
36.6 Test Function 1283
36.6.1 Standard Test Mode 1283
36.6.2 Listen-Only Mode 1283
36.6.3 Self-Test Mode (Loopback Mode) 1284
36.6.4 RAM Test 1284
36.7 Interrupt 1285
36.8 RAM Window 1288
36.9 Initial Settings 1289
36.9.1 Clock Setting 1290
36.9.2 Bit Timing Setting 1290
36.9.3 Communication Speed Setting 1291
36.9.4 Receive Rule Setting 1292
36.9.5 Buffer Setting 1293
36.10 Reception Procedure 1294
36.10.1 Receive Buffer Reading Procedure 1294
36.10.2 FIFO Buffer Reading Procedure 1296
36.11 Transmission Procedure 1298
36.11.1 Procedure for Transmission from Transmit Buffers 1298
36.11.2 Procedure for Transmission from Transmit/Receive FIFO Buffers 1301
36.11.3 Transmit History Buffer Reading Procedure 1304
36.12 Test Settings 1305
36.12.1 Self-Test Mode Setting Procedure 1305
36.12.2 Protection Unlock Procedure 1306
36.12.3 RAM Test Setting Procedure 1307
36.13 Notes on the CAN Module 1308
37. Serial Sound Interface (SSI) 1309
37.1 Overview 1309
37.2 Register Descriptions 1311
37.2.1 Control Register (SSICR) 1311
37.2.2 Status Register (SSISR) 1315
37.2.3 FIFO Control Register (SSIFCR) 1317
37.2.4 FIFO Status Register (SSIFSR) 1319
37.2.5 Transmit FIFO Data Register (SSIFTDR) 1321
37.2.6 Receive FIFO Data Register (SSIFRDR) 1321
37.2.7 TDM Mode Register (SSITDMR) 1322

Downloaded from Arrow.com.

37.3 Operation 1323
37.3.1 Bus Format 1323
37.3.2 Non-Compressed Mode 1323
37.3.3 WS Continue Mode 1329
37.3.4 Operating States 1330
37.3.5 Transmit Operation 1331
37.3.6 Receive Operation 1334
37.3.7 Serial Bit Clock Control 1336
37.4 Interrupt Sources 1336
37.5 Usage Notes 1337
37.5.1 Setting the Module Stop Function 1337
37.5.2 Notes on Changing Transmission Modes 1337
37.5.3 Limits on WS Continue Mode 1337
38. Serial Peripheral Interface (RSPIa) 1338
38.1 Overview 1338
38.2 Register Descriptions 1342
38.2.1 RSPI Control Register (SPCR) 1342
38.2.2 RSPI Slave Select Polarity Register (SSLP) 1344
38.2.3 RSPI Pin Control Register (SPPCR) 1345
38.2.4 RSPI Status Register (SPSR) 1346
38.2.5 RSPI Data Register (SPDR) 1349
38.2.6 RSPI Sequence Control Register (SPSCR) 1352
38.2.7 RSPI Sequence Status Register (SPSSR) 1353
38.2.8 RSPI Bit Rate Register (SPBR) 1354
38.2.9 RSPI Data Control Register (SPDCR) 1355
38.2.10 RSPI Clock Delay Register (SPCKD) 1357
38.2.11 RSPI Slave Select Negation Delay Register (SSLND) 1358
38.2.12 RSPI Next-Access Delay Register (SPND) 1359
38.2.13 RSPI Control Register 2 (SPCR2) 1360
38.2.14 RSPI Command Register m (SPCMDm) (m = 0 to 7) 1362
38.3 Operation 1365
38.3.1 Overview of RSPI Operations 1365
38.3.2 Controlling RSPI Pins 1366
38.3.3 RSPI System Configuration Examples 1367
38.3.3.1 Single Master/Single Slave (with This MCU Acting as Master) 1367
38.3.3.2 Single Master/Single Slave (with This MCU Acting as Slave) 1368
38.3.3.3 Single Master/Multi-Slave (with This MCU Acting as Master) 1369
38.3.3.4 Single Master/Multi-Slave (with This MCU Acting as Slave) 1370
38.3.3.5 Multi-Master/Multi-Slave (with This MCU Acting as Master) 1371
38.3.3.6 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) (with This MCU Acting as Master) 1372

Downloaded from Arrow.com.

38.3.3.7 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) (with This MCU Acting as Slave) ... 1372
38.3.4 Data Format ... 1373
38.3.4.1 When Parity is Disabled (SPCR2.SPPE = 0) ... 1374
38.3.4.2 When Parity is Enabled (SPCR2.SPPE = 1) ... 1378
38.3.5 Transfer Format ... 1382
38.3.5.1 CPHA = 0 ... 1382
38.3.5.2 CPHA = 1 ... 1383
38.3.6 Communications Operating Mode ... 1384
38.3.6.1 Full-Duplex Synchronous Serial Communications (SPCR.TXMD = 0) ... 1384
38.3.6.2 Transmit Operations Only (SPCR.TXMD = 1) ... 1385
38.3.7 Transmit Buffer Empty/Receive Buffer Full Interrupts ... 1386
38.3.8 Error Detection ... 1388
38.3.8.1 Overrun Error ... 1389
38.3.8.2 Parity Error ... 1391
38.3.8.3 Mode Fault Error ... 1392
38.3.9 Initializing RSPI ... 1393
38.3.9.1 Initialization by Clearing the SPE Bit ... 1393
38.3.9.2 System Reset ... 1393
38.3.10 SPI Operation ... 1394
38.3.10.1 Master Mode Operation ... 1394
38.3.10.2 Slave Mode Operation ... 1404
38.3.11 Clock Synchronous Operation ... 1408
38.3.11.1 Master Mode Operation ... 1408
38.3.11.2 Slave Mode Operation ... 1412
38.3.12 Loopback Mode ... 1414
38.3.13 Self-Diagnosis of Parity Bit Function ... 1415
38.3.14 Interrupt Sources ... 1416
38.4 Link Operation by Event Linking ... 1417
38.4.1 Receive Buffer Full Event Output ... 1417
38.4.2 Transmit Buffer Empty Event Output ... 1417
38.4.3 Mode Fault, Overrun, or Parity Error Event Output ... 1417
38.4.4 RSPI Idle Event Output ... 1418
38.4.5 Transmission-Completed Event Output ... 1418
38.5 Usage Notes ... 1419
38.5.1 Setting Module Stop Function ... 1419
38.5.2 Note on Low Power Consumption Functions ... 1419
38.5.3 Notes on Starting Transfer ... 1419
38.5.4 Notes on the SPRF and SPTEF flags ... 1419
39. CRC Calculator (CRC) ... 1420
39.1 Overview ... 1420

Downloaded from Arrow.com.

39.2 Register Descriptions 1421
39.2.1 CRC Control Register (CRCCR) 1421
39.2.2 CRC Data Input Register (CRCDIR) 1421
39.2.3 CRC Data Output Register (CRCDOR) 1422
39.3 Operation 1423
39.4 Usage Notes 1426
39.4.1 Module Stop Function Setting 1426
39.4.2 Note on Transmission 1426
40. SD Host Interface (SDHIa) 1427
40.1 Overview 1427
40.2 Register Details 1428
40.2.1 Command Register (SDCMD) 1428
40.2.2 Argument Register (SDARG) 1430
40.2.3 Data Stop Register (SDSTOP) 1430
40.2.4 Block Count Register (SDBLKCNT) 1431
40.2.5 Response Register 10 (SDRSP10), Response Register 32 (SDRSP32), Response Register 54 (SDRSP54), Response Register 76 (SDRSP76) 1432
40.2.6 SD Status Register 1 (SDSTS1) 1433
40.2.7 SD Status Register 2 (SDSTS2) 1436
40.2.8 SD Interrupt Mask Register 1 (SDIMSK1) 1440
40.2.9 SD Interrupt Mask Register 2 (SDIMSK2) 1441
40.2.10 SDHI Clock Control Register (SDCLKCR) 1442
40.2.11 Transfer Data Size Register (SDSIZE) 1443
40.2.12 Card Access Option Register (SDOPT) 1444
40.2.13 SD Error Status Register 1 (SDERSTS1) 1445
40.2.14 SD Error Status Register 2 (SDERSTS2) 1446
40.2.15 SD Buffer Register (SDBUFR) 1447
40.2.16 SDIO Mode Control Register (SDIOMD) 1447
40.2.17 SDIO Status Register (SDIOSTS) 1449
40.2.18 SDIO Interrupt Mask Register (SDIOIMSK) 1450
40.2.19 DMA Transfer Enable Register (SDDMAEN) 1451
40.2.20 SDHI Software Reset Register (SDRST) 1452
40.2.21 Swap Control Register (SDSWAP) 1453
40.3 SDHI Operation 1454
40.3.1 Data Block Format of the SD Card 1454
40.3.2 SD Buffer and the SDBUFR Register 1455
40.3.3 SD Card Detection 1456
40.3.3.1 Using the SDHI_CD Pin to Detect an SD Card 1456
40.3.3.2 Using the SDHI_D3 Pin to Detect an SD Card 1456
40.3.4 SD Card Write Protection 1457
40.3.4.1 Using the SDHI_WP Pin to Enable Write Protection 1457

Downloaded from Arrow.com.

40.3.4.2 Using a Command to Enable Write Protection 1457
40.3.5 Communication Errors and Timeouts 1458
40.3.6 Examples of Issuing a Command 1459
40.3.6.1 Command Absent of Response Reception and Data Transfer 1459
40.3.6.2 Command Absent of Data Transfer 1460
40.3.6.3 Single Block Read Command (CMD17) 1461
40.3.6.4 Single Block Write Command (CMD24) 1463
40.3.6.5 Multi-Block Read Command (CMD18) 1465
40.3.6.6 Multi-Block Write Command (CMD25) 1467
40.3.6.7 IO_RW_DIRECT Command (CMD52) 1469
40.3.6.8 IO_RW_EXTENDED Command (CMD53 (Multi-Block Read)) 1470
40.3.6.9 IO_RW_EXTENDED (CMD53 Multi-Block Write) 1472
40.3.6.10 DMA Transfer 1474
40.4 Interrupts 1476
40.4.1 DMA Transfer Triggered by Interrupt Requests 1477
40.5 Notes on Using the SDHI 1478
40.5.1 Illegal Read Access During a Multi-Block Read and How To Avoid It 1478
40.5.2 SDBUFR Register Illegal Write Error 1478
40.5.3 Automatic Control of the SDHI Clock Output 1479
40.5.4 Restrictions on Setting the C52PUB Bit During a Multi-Block Write Sequence 1479
40.5.5 Note on Setting the SDCLKCR Register 1479
40.5.6 Writing to the SDSTOP Register During a Multi-Block Read Sequence 1479
40.5.7 Controlling Module Operation 1479
41. Bluetooth Low Energy (BLE) 1480
41.1 Overview 1480
41.2 Operation 1484
41.2.1 State Transitions 1484
41.3 Interrupts 1485
41.4 Certificates of Compliance 1486
41.4.1 Radio-Related Laws 1486
41.4.2 Bluetooth SIG Certification 1486
41.4.3 Labeling and Information RX23W Users Should Provide to End Users of Their Products 1486
41.5 Usage Notes 1489
41.5.1 RF Transceiver Power-Supply 1489
41.5.2 Wireless Standards 1490
41.5.3 Notes on Board Design 1490
42. Trusted Secure IP (TSIP-Lite) 1491
42.1 Overview 1491
42.2 Operation 1493
42.2.1 Operating Modes and State Transitions 1493

Downloaded from Arrow.com.

42.2.2 Encryption Engine 1494
42.2.3 Key Installation 1495
42.2.4 Encryption and Decryption 1496
42.2.5 Generating Key Generation Information (by Using Random Numbers) 1499
42.2.6 Random Number Generation 1499
42.3 Interrupt 1500
42.4 Usage Notes 1500
42.4.1 Standby Mode 1500
42.4.2 Setting the Module Stop Function 1500
42.4.3 TSIP-Lite Library 1500
43. Capacitive Touch Sensing Unit (CTSU) 1501
43.1 Overview 1502
43.2 Register Descriptions 1503
43.2.1 CTSU Control Register 0 (CTSUCR0) 1503
43.2.2 CTSU Control Register 1 (CTSUCR1) 1505
43.2.3 CTSU Synchronous Noise Reduction Setting Register (CTSUSDPRS) 1506
43.2.4 CTSU Sensor Stabilization Wait Control Register (CTSUSST) 1507
43.2.5 CTSU Measurement Channel Register 0 (CTSUMCH0) 1508
43.2.6 CTSU Measurement Channel Register 1 (CTSUMCH1) 1509
43.2.7 CTSU Channel Enable Control Register 0 (CTSUCHAC0) 1510
43.2.8 CTSU Channel Enable Control Register 1 (CTSUCHAC1) 1511
43.2.9 CTSU Channel Enable Control Register 2 (CTSUCHAC2) 1512
43.2.10 CTSU Channel Enable Control Register 3 (CTSUCHAC3) 1513
43.2.11 CTSU Channel Enable Control Register 4 (CTSUCHAC4) 1514
43.2.12 CTSU Channel Transmit/Receive Control Register 0 (CTSUCHTRC0) 1515
43.2.13 CTSU Channel Transmit/Receive Control Register 1 (CTSUCHTRC1) 1516
43.2.14 CTSU Channel Transmit/Receive Control Register 2 (CTSUCHTRC2) 1517
43.2.15 CTSU Channel Transmit/Receive Control Register 3 (CTSUCHTRC3) 1518
43.2.16 CTSU Channel Transmit/Receive Control Register 4 (CTSUCHTRC4) 1519
43.2.17 CTSU High-Pass Noise Reduction Control Register (CTSUDCLKC) 1520
43.2.18 CTSU Status Register (CTSUST) 1521
43.2.19 CTSU High-Pass Noise Reduction Spectrum Diffusion Control Register (CTSUSSC) 1523
43.2.20 CTSU Sensor Offset Register 0 (CTSUSO0) 1524
43.2.21 CTSU Sensor Offset Register 1 (CTSUSO1) 1525
43.2.22 CTSU Sensor Counter (CTSUSC) 1526
43.2.23 CTSU Reference Counter (CTSURC) 1527
43.2.24 CTSU Error Status Register (CTSUERRS) 1528
43.3 Operation 1530
43.3.1 Principles of Measurement Operation 1530
43.3.2 Measurement Modes 1532
43.3.2.1 Initial Setting Flowchart 1533

Downloaded from Arrow.com.

43.3.2.2 Status Counter 1534
43.3.2.3 Self-Capacitance Single Scan Mode Operation 1535
43.3.2.4 Self-Capacitance Multi-Scan Mode Operation 1537
43.3.2.5 Mutual Capacitance Full Scan Mode Operation 1539
43.3.3 Items Common to Multiple Modes 1542
43.3.3.1 Sensor Stabilization Wait Time and Measurement Time 1542
43.3.3.2 Interrupts 1543
43.4 Usage Notes 1545
43.4.1 Measurement Result Data (CTSUSC and CTSURC Counters) 1545
43.4.2 Software Trigger 1545
43.4.3 External Trigger 1545
43.4.4 Notes on Forcibly Stopping Operation 1546
43.4.5 TSCAP Pin 1546
43.4.6 Notes during Measurement Operation (CTSUCR0.CTSUSTRT Bit = 1) 1546
44. 12-Bit A/D Converter (S12ADE) 1547
44.1 Overview 1547
44.2 Register Descriptions 1551
44.2.1 A/D Data Registers y (ADDRy) (y = 0 to 7, 16 to 20, 27), A/D Data Duplication Register (ADDBLDR), A/D Temperature Sensor Data Register (ADTSDR), A/D Internal Reference Voltage Data Register (ADOCDR) 1551
44.2.2 A/D Self-Diagnosis Data Register (ADRD) 1553
44.2.3 A/D Control Register (ADCSR) 1554
44.2.4 A/D Channel Select Register A0 (ADANSA0) 1557
44.2.5 A/D Channel Select Register A1 (ADANSA1) 1558
44.2.6 A/D Channel Select Register B0 (ADANSB0) 1559
44.2.7 A/D Channel Select Register B1 (ADANSB1) 1560
44.2.8 A/D-Converted Value Addition/Average Function Select Register 0 (ADADS0) 1561
44.2.9 A/D-Converted Value Addition/Average Function Select Register 1 (ADADS1) 1562
44.2.10 A/D-Converted Value Addition/Average Count Select Register (ADADC) 1564
44.2.11 A/D Control Extended Register (ADCER) 1565
44.2.12 A/D Conversion Start Trigger Select Register (ADSTRGR) 1567
44.2.13 A/D Conversion Extended Input Control Register (ADEXICR) 1569
44.2.14 A/D Sampling State Register n (ADSSTRn) (n = 0 to 7, L, T, O) 1571
44.2.15 A/D Disconnection Detection Control Register (ADDISCR) 1572
44.2.16 A/D Event Link Control Register (ADELCCR) 1573
44.2.17 A/D Group Scan Priority Control Register (ADGSPCR) 1574
44.2.18 A/D Compare Function Control Register (ADCMPCR) 1575
44.2.19 A/D Compare Function Window A Channel Select Register 0 (ADCMPANSR0) 1577
44.2.20 A/D Compare Function Window A Channel Select Register 1 (ADCMPANSR1) 1578
44.2.21 A/D Compare Function Window A Extended Input Select Register (ADCMPANSER) 1579

Downloaded from Arrow.com.

44.2.22 A/D Compare Function Window A Comparison Condition Setting Register 0 (ADCMPLR0) 1580
44.2.23 A/D Compare Function Window A Comparison Condition Setting Register 1 (ADCMPLR1) 1582
44.2.24 A/D Compare Function Window A Extended Input Comparison Condition Setting Register (ADCMPLER) 1583
44.2.25 A/D Compare Function Window A Lower-Side Level Setting Register (ADCMPDR0) 1584
44.2.26 A/D Compare Function Window A Upper-Side Level Setting Register (ADCMPDR1) 1586
44.2.27 A/D Compare Function Window A Channel Status Register 0 (ADCMPSR0) 1588
44.2.28 A/D Compare Function Window A Channel Status Register 1 (ADCMPSR1) 1589
44.2.29 A/D Compare Function Window A Extended Input Channel Status Register (ADCMPSER) 1590
44.2.30 A/D High-Potential/Low-Potential Reference Voltage Control Register (ADHVREFCNT) 1591
44.2.31 A/D Compare Function Window A/B Status Monitor Register (ADWINMON) 1592
44.2.32 A/D Compare Function Window B Channel Select Register (ADCMPBNSR) 1593
44.2.33 A/D Compare Function Window B Lower-Side Level Setting Register (ADWINLLB) 1595
44.2.34 A/D Compare Function Window B Upper-Side Level Setting Register (ADWINULB) 1597
44.2.35 A/D Compare Function Window B Channel Status Register (ADCMPBSR) 1598
44.2.36 A/D Data Storage Buffer Register n (ADBUFn) (n = 0 to 15) 1599
44.2.37 A/D Data Storage Buffer Enable Register (ADBUFEN) 1600
44.2.38 A/D Data Storage Buffer Pointer Register (ADBUFPTR) 1601
44.3 Operation 1602
44.3.1 Scanning Operation 1602
44.3.2 Single Scan Mode 1603
44.3.2.1 Basic Operation 1603
44.3.2.2 Channel Selection and Self-Diagnosis 1604
44.3.2.3 A/D Conversion of Temperature Sensor Output/Internal Reference Voltage 1605
44.3.2.4 A/D Conversion in Double Trigger Mode 1606
44.3.3 Continuous Scan Mode 1607
44.3.3.1 Basic Operation 1607
44.3.3.2 Channel Selection and Self-Diagnosis 1608
44.3.4 Group Scan Mode 1609
44.3.4.1 Basic Operation 1609
44.3.4.2 A/D Conversion in Double Trigger Mode 1610
44.3.4.3 Operation under Group-A Priority Control 1611
44.3.5 Compare Function (Window A, Window B) 1621
44.3.5.1 Compare Function Window A/B 1621
44.3.5.2 ELC Output of Compare Function 1623
44.3.5.3 Using Data Buffers 1625
44.3.5.4 Restrictions for Compare Function 1626
44.3.6 Analog Input Sampling Time and Scan Conversion Time 1626

Downloaded from Arrow.com.

44.3.7 Usage Example of A/D Data Register Automatic Clearing Function 1628
44.3.8 A/D-Converted Value Addition/Average Mode 1629
44.3.9 Disconnection Detection Assist Function 1629
44.3.10 Starting A/D Conversion with Asynchronous Trigger 1631
44.3.11 Starting A/D Conversion with Synchronous Trigger from Peripheral Module 1631
44.4 Interrupt Sources and DTC/DMAC Transfer Requests 1631
44.4.1 Interrupt Requests 1631
44.5 Event Link Function 1632
44.5.1 Event Output to the ELC 1632
44.5.2 12-Bit A/D Converter Operation by Event from the ELC 1632
44.5.3 Note on 12-Bit A/D Converter When an Event Is Input from the ELC 1632
44.6 Selecting Reference Voltage 1632
44.7 Allowable Impedance of Signal Source 1633
44.8 Usage Notes 1634
44.8.1 Notes on Reading Data Registers 1634
44.8.2 Notes on Stopping A/D Conversion 1634
44.8.3 A/D Conversion Restarting Timing and Termination Timing 1635
44.8.4 Notes on Scan End Interrupt Handling 1635
44.8.5 Module Stop Function Setting 1635
44.8.6 Notes on Entering Low Power Consumption States 1635
44.8.7 Notes on Canceling Software Standby Mode 1635
44.8.8 Pin Setting when the 12-bit A/D Converter is Used 1635
44.8.9 Error in Absolute Accuracy When Disconnection Detection Assistance is in Use 1635
44.8.10 ADHSC Bit Rewriting Procedure 1636
44.8.11 Voltage Range of Analog Power Supply Pins 1637
44.8.12 Notes on Board Design 1638
44.8.13 Notes on Noise Prevention 1638
45. 12-Bit D/A Converter (R12DAA) 1639
45.1 Overview 1639
45.2 Register Descriptions 1640
45.2.1 D/A Data Register m (DADRm) (m = 0, 1) 1640
45.2.2 D/A Control Register (DACR) 1641
45.2.3 Data Register Format Select Register (DADPR) 1641
45.2.4 D/A A/D Synchronous Start Control Register (DAADSCR) 1642
45.2.5 D/A VREF Control Register (DAVREFCR) 1643
45.3 Operation 1644
45.3.1 Measure against Interference between D/A and A/D Conversion 1645
45.3.2 Notes on Using the Internal Reference Voltage as the Reference Voltage 1647
45.4 Event Link Operation Setting Procedure 1648
45.5 Usage Notes on Event Link Operation 1648
45.6 Usage Notes 1649

Downloaded from Arrow.com.

45.6.1 Module Stop Function Setting 1649
45.6.2 Operation of the D/A Converter in Module Stop State 1649
45.6.3 Operation of the D/A Converter in Software Standby Mode 1649
45.6.4 Note on Usage When Measure against Interference between D/A and A/D Conversion is Enabled 1649
46. Temperature Sensor (TEMPSA) 1650
46.1 Overview 1650
46.2 Register Descriptions 1651
46.2.1 Temperature Sensor Calibration Data Register (TSCDRH, TSCDRL) 1651
46.3 Using the Temperature Sensor 1652
46.3.1 Preparation for Using the Temperature Sensor 1652
46.3.2 Setting the 12-Bit A/D Converter 1653
47. Comparator B (CMPBa) 1654
47.1 Overview 1654
47.2 Register Descriptions 1657
47.2.1 Comparator B1 Control Register 1 (CPB1CNT1) 1657
47.2.2 Comparator B1 Control Register 2 (CPB1CNT2) 1657
47.2.3 Comparator B1 Flag Register (CPB1FLG) 1658
47.2.4 Comparator B1 Interrupt Control Register (CPB1INT) 1659
47.2.5 Comparator B1 Filter Select Register (CPB1F) 1660
47.2.6 Comparator B1 Mode Select Register (CPB1MD) 1660
47.2.7 Comparator B1 Reference Input Voltage Select Register (CPB1REF) 1661
47.2.8 Comparator B1 Output Control Register (CPB1OCR) 1662
47.3 Operation 1663
47.3.1 Comparator Bn Digital Filter (n = 2, 3) 1667
47.3.2 Comparator Bn Output Function (n = 2, 3) 1668
47.3.3 Example of Using Comparator B to Exit Software Standby Mode 1668
47.4 Comparator B2 and Comparator B3 Interrupts 1669
47.5 Usage Note 1669
47.5.1 Module Stop Function Setting 1669
48. Data Operation Circuit (DOC) 1670
48.1 Overview 1670
48.2 Register Descriptions 1671
48.2.1 DOC Control Register (DOCR) 1671
48.2.2 DOC Data Input Register (DODIR) 1672
48.2.3 DOC Data Setting Register (DODSR) 1672
48.3 Operation 1673
48.3.1 Data Comparison Mode 1673
48.3.2 Data Addition Mode 1674
48.3.3 Data Subtraction Mode 1675
48.4 Interrupt Requests 1675

Downloaded from Arrow.com.

48.5 Event Link Output 1676
48.5.1 Interrupt Handling and Event Linking 1676
48.6 Usage Note 1676
48.6.1 Module Stop Function Setting 1676
49. RAM 1677
49.1 Overview 1677
49.2 Operation 1677
49.2.1 Low Power Consumption Function 1677
49.2.2 Notes on Self-Diagnosis of the RAM 1677
50. Flash Memory (FLASH) 1678
50.1 Overview 1678
50.2 ROM Area and Block Configuration 1679
50.3 E2 DataFlash Area and Block Configuration 1680
50.4 Register Descriptions 1681
50.4.1 E2 DataFlash Control Register (DFLCTL) 1681
50.4.2 Flash P/E Mode Entry Register (FENTRYR) 1682
50.4.3 Protection Unlock Register (FPR) 1683
50.4.4 Protection Unlock Status Register (FPSR) 1683
50.4.5 Flash P/E Mode Control Register (FPMCR) 1684
50.4.6 Flash Initial Setting Register (FISR) 1685
50.4.7 Flash Reset Register (FRESETR) 1687
50.4.8 Flash Area Select Register (FASR) 1687
50.4.9 Flash Control Register (FCR) 1688
50.4.10 Flash Extra Area Control Register (FEXCR) 1690
50.4.11 Flash Processing Start Address Register H (FSARH) 1691
50.4.12 Flash Processing Start Address Register L (FSARL) 1691
50.4.13 Flash Processing End Address Register H (FEARH) 1692
50.4.14 Flash Processing End Address Register L (FEARL) 1692
50.4.15 Flash Write Buffer Register n (FWBn) (n = 0 to 3) 1693
50.4.16 Flash Status Register 0 (FSTATR0) 1694
50.4.17 Flash Status Register 1 (FSTATR1) 1696
50.4.18 Flash Error Address Monitor Register H (FEAMH) 1697
50.4.19 Flash Error Address Monitor Register L (FEAML) 1697
50.4.20 Flash Start-Up Setting Monitor Register (FSCMR) 1698
50.4.21 Flash Access Window Start Address Monitor Register (FAWSMR) 1698
50.4.22 Flash Access Window End Address Monitor Register (FAWEMR) 1699
50.4.23 Unique ID Register n (UIDRn) (n = 0 to 3) 1699
50.5 Start-Up Program Protection 1700
50.6 Area Protection 1701
50.7 Programming and Erasure 1702
50.7.1 Sequencer Modes 1702

Downloaded from Arrow.com.

50.7.1.1 E2 DataFlash Access Disabled Mode 1702
50.7.1.2 Read Mode 1703
50.7.1.3 P/E Modes 1703
50.7.2 Mode Transitions 1703
50.7.2.1 Transition from E2 DataFlash Access Disable Mode to Read Mode 1703
50.7.2.2 Transition from Read Mode to P/E Mode 1704
50.7.2.3 Transition from P/E Mode to Read Mode 1706
50.7.3 Software Commands 1708
50.7.4 Software Command Usage 1709
50.7.4.1 Program 1709
50.7.4.2 Block Erase 1711
50.7.4.3 All-Block Erase 1713
50.7.4.4 Blank Check 1715
50.7.4.5 Start-Up Area Information Program/Access Window Information Program 1717
50.7.4.6 Forced Stop of Software Commands 1718
50.7.5 Interrupt 1718
50.8 Boot Mode 1719
50.8.1 Boot Mode (USB Interface) 1720
50.8.1.1 Operating Conditions in Boot Mode (USB Interface) 1720
50.8.2 Boot Mode (SCI Interface) 1723
50.8.2.1 Operating Conditions in Boot Mode (SCI Interface) 1723
50.8.2.2 Starting Up in Boot Mode (SCI Interface) 1724
50.8.3 Boot Mode (FINE Interface) 1725
50.8.3.1 Operating Conditions in Boot Mode (FINE Interface) 1725
50.9 Flash Memory Protection 1726
50.9.1 ID Code Protection 1726
50.9.1.1 Boot Mode ID Code Protection 1727
50.9.1.2 On-Chip Debugging Emulator ID Code Protection 1728
50.10 Communication Protocol 1729
50.10.1 State Transition in Boot Mode (SCI Interface) 1729
50.10.2 Command and Response Configuration 1730
50.10.3 Response to Undefined Commands 1730
50.10.4 Boot Mode Status Inquiry 1731
50.10.5 Inquiry Commands 1732
50.10.5.1 Supported Device Inquiry 1732
50.10.5.2 Data Area Availability Inquiry 1733
50.10.5.3 User Area Information Inquiry 1733
50.10.5.4 Data Area Information Inquiry 1734
50.10.5.5 Block Information Inquiry 1734
50.10.6 Setting Commands 1735
50.10.6.1 Device Select 1735

Downloaded from Arrow.com.

50.10.6.2 Operating Frequency Select 1736
50.10.6.3 Program/Erase Host Command Wait State Transition 1737
50.10.7 ID Code Authentication Command 1738
50.10.7.1 ID Code Check 1738
50.10.8 Program/Erase Commands 1739
50.10.8.1 User/Data Area Program Preparation 1739
50.10.8.2 Program 1740
50.10.8.3 Data Area Program 1741
50.10.8.4 Erase Preparation 1742
50.10.8.5 Block Erase 1742
50.10.9 Read-Check Commands 1743
50.10.9.1 Memory Read 1743
50.10.9.2 User Area Checksum 1744
50.10.9.3 Data Area Checksum 1745
50.10.9.4 User Area Blank Check 1745
50.10.9.5 Data Area Blank Check 1746
50.10.9.6 Access Window Information Program 1746
50.10.9.7 Access Window Read 1747
50.11 Serial Programmer Operation in Boot Mode (SCI Interface) 1748
50.11.1 Bit Rate Automatic Adjustment 1749
50.11.2 Receive the MCU Information 1750
50.11.3 Select the Device and Change the Bit Rate 1751
50.11.4 Enter the Program/Erase Host Command Wait State 1752
50.11.5 Unlock Boot Mode ID Code Protection 1753
50.11.6 Erase the User Area and Data Area 1754
50.11.7 Program the User Area and Data Area 1755
50.11.8 Check Data in the User Area 1756
50.11.9 Check Data in the Data Area 1757
50.11.10 Set the Access Window in the User Area 1758
50.12 Rewriting by Self-Programming 1759
50.12.1 Overview 1759
50.13 Usage Notes 1760
50.14 Usage Notes in Boot Mode 1761
51. Electrical Characteristics 1762
51.1 Absolute Maximum Ratings 1762
51.2 DC Characteristics 1764
51.2.1 Normal I/O Pin Output Characteristics (1) 1781
51.2.2 Normal I/O Pin Output Characteristics (2) 1783
51.2.3 Normal I/O Pin Output Characteristics (3) 1785
51.3 AC Characteristics 1787
51.3.1 Clock Timing 1787

Downloaded from Arrow.com.

51.3.2 Reset Timing 1792
51.3.3 Timing of Recovery from Low Power Consumption Modes 1794
51.3.4 Control Signal Timing 1797
51.3.5 Timing of On-Chip Peripheral Modules 1798
51.3.5.1 Timing of I/O Ports 1798
51.3.5.2 Timing of MTU/TPU 1799
51.3.5.3 Timing of POE 1800
51.3.5.4 Timing of TMR 1800
51.3.5.5 Timing of SCI 1801
51.3.5.6 Timing of RIIC 1807
51.3.5.7 Timing of RSPI 1809
51.3.5.8 Timing of SSI 1813
51.3.5.9 Timing of SDHI 1815
51.3.5.10 Timing of A/D Converter Trigger 1816
51.3.5.11 Timing of CAC 1816
51.3.5.12 Timing of CLKOUT 1817
51.3.5.13 Timing of CLKOUT_RF 1818
51.4 USB Characteristics 1819
51.5 A/D Conversion Characteristics 1821
51.6 D/A Conversion Characteristics 1828
51.7 Temperature Sensor Characteristics 1830
51.8 Comparator Characteristics 1830
51.9 CTSU Characteristics 1832
51.10 Characteristics of Power-On Reset Circuit and Voltage Detection Circuit 1832
51.11 Oscillation Stop Detection Timing 1835
51.12 Battery Backup Function Characteristics 1836
51.13 ROM (Flash Memory for Code Storage) Characteristics 1837
51.14 E2 DataFlash Characteristics (Flash Memory for Data Storage) 1839
51.15 BLE Characteristics 1840
51.15.1 Transmission Characteristics 1840
51.15.2 Reception Characteristics (2 Mbps) 1840
51.15.3 Reception Characteristics (1 Mbps) 1841
51.15.4 Reception Characteristics (500 kbps) 1841
51.15.5 Reception Characteristics (125 kbps) 1842
51.16 Usage Notes 1843
51.16.1 Connecting VCL Capacitor and Bypass Capacitors 1843
Appendix 1. Port States in Each Processing Mode 1844
Appendix 2. Package Dimensions 1845
REVISION HISTORY 1851

Downloaded from Arrow.com.

54-MHz 32-bit RX MCUs, built-in FPU, 88.56 DMIPS, up to 512-KB flash memory, *Bluetooth*® 5.0, various communication functions including USB 2.0 full-speed host/function/OTG, CAN, SD host interface, serial sound interface, capacitive touch sensing unit, 12-bit A/D converter, 12-bit D/A converter, RTC, Encryption functions

Features

■ 32-bit RXv2 CPU core
- Max. operating frequency: 54 MHz
  Capable of 88.56 DMIPS in operation at 54 MHz
- Enhanced DSP: 32-bit multiply-accumulate and 16-bit multiply-subtract instructions supported
- Built-in FPU: 32-bit single-precision floating point (compliant to IEEE754)
- Divider (fastest instruction execution takes two CPU clock cycles)
- Fast interrupt
- CISC Harvard architecture with 5-stage pipeline
- Variable-length instructions, ultra-compact code
- On-chip debugging circuit
- Memory protection unit (MPU) supported

■ Low power design and architecture
- Operation from a single 1.8-V to 3.6-V supply
- RTC capable of operating on the battery backup power supply
- Three low power consumption modes
- Low power timer (LPT) that operates during the software standby state

■ On-chip flash memory for code
- 384- to 512-Kbyte capacities
- On-board or off-board user programming
- Programmable at 1.8 V
- For instructions and operands

■ On-chip data flash memory
- 8 Kbytes (1,000,000 program/erase cycles (typ.))
- BGO (Background Operation)

■ On-chip SRAM, no wait states
- 64-Kbyte size capacities

■ Data transfer functions
- DMAC: Incorporates four channels
- DTC: Four transfer modes

■ ELC
- Module operation can be initiated by event signals without using interrupts.
- Linked operation between modules is possible while the CPU is sleeping.

■ Reset and supply management
- Eight types of reset, including the power-on reset (POR)
- Low voltage detection (LVD) with voltage settings

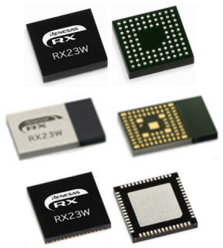
■ Clock functions
- Main clock oscillator frequency: 1 to 20 MHz
- External clock input frequency: Up to 20 MHz
- Sub-clock oscillator frequency: 32.768 kHz
- Frequency of Bluetooth-dedicated clock oscillator: 32 MHz
- PLL circuit input: 4 MHz to 12.5 MHz
- On-chip low- and high-speed oscillators, dedicated on-chip low-speed oscillator for the IWDT
- USB-dedicated PLL circuit: 4, 6, 8, or 12 MHz
  54 MHz can be set for the system clock and 48 MHz for the USB clock
- Generation of a dedicated 32.768-kHz clock for the RTC
- Clock frequency accuracy measurement circuit (CAC)

■ Realtime clock
- Adjustment functions (30 seconds, leap year, and error)
- Calendar count mode or binary count mode selectable
- Time capture function
- Time capture on event-signal input through external pins

■ Independent watchdog timer
- 15-kHz on-chip oscillator produces a dedicated clock signal to drive IWDT operation.

■ Useful functions for IEC60730 compliance
- Self-diagnostic and disconnection-detection assistance functions for the A/D converter, clock frequency accuracy measurement circuit, independent watchdog timer, RAM test assistance functions using the DOC, etc.

■ Capacitive touch sensing unit
- Self-capacitance method: A single pin configures a single key, supporting up to 12 keys
- Mutual capacitance method: Matrix configuration with 12 pins, supporting up to 36 keys



PTBG0085KB-A 5.5 × 5.5 mm, 0.5 mm pitch

PTLG0083KA-A 6.1 × 9.5 mm, 0.5 mm pitch
PTLG0083KA-B 6.1 × 9.5 mm, 0.5 mm pitch

PVQN0056LA-A 7 × 7 mm, 0.4 mm pitch

■ Up to 12 communication functions
- Bluetooth Low Energy (1 channel)
  An RF transceiver and link layer compliant with the Bluetooth 5.0 Low Energy specification
  LE 1M PHY, LE 2M PHY, LE Coded PHY (125 kbps and 500 kbps), and LE Advertising extension support
  On-chip Bluetooth-dedicated AES-CCM (128-bit blocks) encryption circuit
  The 83-pin LGA product has been certified as compliant with radio-related laws (in Japan, North America, and Europe).
  The 83-pin LGA product includes a small PCB trace antenna.
- USB 2.0 host/function/On-The-Go (OTG) (one channel), full-speed = 12 Mbps, low-speed = 1.5 Mbps, isochronous transfer, and BC (Battery Charger) supported
- CAN (one channel) compliant to ISO11898-1:
  Transfer at up to 1 Mbps
- SCI with many useful functions (up to 4 channels)
  Asynchronous mode, clock synchronous mode, smart card interface
  Reduction of errors in communications using the bit modulation function
- IrDA interface (one channel, in cooperation with the SCI5)
- I^2C bus interface: Transfer at up to 400 kbps, capable of SMBus operation (one channel)
- RSPI (one channel): Transfer at up to 16 Mbps
- Serial sound interface (one channel)
- SD host interface (optional: one channel) SD memory/ SDIO 1-bit or 4-bit SD bus supported

■ Up to 19 extended-function timers
- 16-bit MTU: input capture, output compare, complementary PWM output, phase counting mode (five channels)
- 16-bit TPU: input capture, output compare, phase counting mode (six channels)
- 8-bit TMR (four channels)
- 16-bit compare-match timers (four channels)

■ 12-bit A/D converter
- Capable of conversion within 0.83 µs
- 14 channels
- Sampling time can be set for each channel
- Self-diagnostic function and analog input disconnection detection assistance function

■ 12-bit D/A converter
- Two channels

■ Analog comparator
- Two channels × one unit

■ General I/O ports
- 5-V tolerant, open drain, input pull-up, switching of driving capacity

■ Encryption functions (TSIP-Lite)
- Unauthorized access to the encryption engine is disabled and imposture and falsification of information are prevented
- Safe management of keys
- 128- or 256-bit key length of AES for ECB, CBC, GCM, others
- True random number generator

■ Temperature sensor

■ Operating temperature range
- –40 to +85°C

■ Applications
- 85-pin BGA, 56-pin QFN: General industrial and consumer equipment
- 83-pin LGA: Consumer equipment

Downloaded from Arrow.com.

1. Overview

1.1 Outline of Specifications

Table 1.1 lists the specifications, and Table 1.2 gives a comparison of the functions of the products in different packages.
Table 1.1 is for products with the greatest number of functions, so the number of peripheral modules and channels will differ in accordance with the package type. For details, see Table 1.2, Comparison of Functions for Different Packages.

Table 1.1 Outline of Specifications (1/5)

Classification	Module/Function	Description
CPU	CPU	• Maximum operating frequency: 54 MHz • 32-bit RX CPU (RX v2) • Minimum instruction execution time: One instruction per clock cycle • Address space: 4-Gbyte linear • Register set General purpose: Sixteen 32-bit registers Control: Ten 32-bit registers Accumulator: Two 72-bit registers • Basic instructions: 75 (variable-length instruction format) • Floating-point instructions: 11 • DSP instructions: 23 • Addressing modes: 10 • Data arrangement Instructions: Little endian Data: Selectable as little endian or big endian • On-chip 32-bit multiplier: 32-bit × 32-bit → 64-bit • On-chip divider: 32-bit ÷ 32-bit → 32 bits • Barrel shifter: 32 bits • Memory protection unit (MPU)
	FPU	• Single precision (32-bit) floating point • Data types and floating-point exceptions in conformance with the IEEE754 standard
Memory	ROM	• Capacity: 384/512 Kbytes • Up to 32 MHz: No-wait memory access 32 to 54 MHz: Wait state required. No wait state if the instruction is served by a ROM accelerator hit. • Programming/erasing method: Serial programming (asynchronous serial communication/USB communication), self-programming
	RAM	• Capacity: 64 Kbytes • 54 MHz, no-wait memory access
	E2 DataFlash	• Capacity: 8 Kbytes • Number of erase/write cycles: 1,000,000 (typ)
MCU operating mode		Single-chip mode
Clock	Clock generation circuit	• Main clock oscillator, sub-clock oscillator, low-speed on-chip oscillator, high-speed on-chip oscillator, PLL frequency synthesizer, USB-dedicated PLL frequency synthesizer, and IWDT-dedicated on-chip oscillator, Bluetooth-dedicated clock oscillator, Bluetooth-dedicated low-speed on-chip oscillator • Oscillation stop detection: Available • Clock frequency accuracy measurement circuit (CAC) • Independent settings for the system clock (ICLK), peripheral module clock (PCLK), and FlashIF clock (FCLK) The CPU and system sections such as other bus masters run in synchronization with the system clock (ICLK): 54 MHz (at max.) MTU2a runs in synchronization with the PCLKA: 54 MHz (at max.) The ADCLK for the S12AD runs in synchronization with the PCLKD: 54 MHz (at max.) Peripheral modules other than MTU2a and S12ADE run in synchronization with the PCLKB: 32 MHz (at max.) The flash peripheral circuit runs in synchronization with the FCLK: 32 MHz (at max.)
Resets		RES# pin reset, power-on reset, voltage monitoring reset, watchdog timer reset, independent watchdog timer reset, and software reset
Voltage detection	Voltage detection circuit (LVDAb)	• When the voltage on VCC falls below the voltage detection level, an internal reset or internal interrupt is generated. Voltage detection circuit 0 is capable of selecting the detection voltage from 3 levels Voltage detection circuit 1 is capable of selecting the detection voltage from 10 levels

Downloaded from Arrow.com.

Table 1.1 Outline of Specifications (2/5)

Classification	Module/Function	Description
Low power consumption	Low power consumption functions	• Module stop function • Three low power consumption modes Sleep mode, deep sleep mode, and software standby mode • Low power timer that operates during the software standby state
	Function for lower operating power consumption	• Operating power control modes High-speed operating mode, middle-speed operating mode, and low-speed operating mode
Interrupt	Interrupt controller (ICUb)	• Interrupt vectors: 148 • External interrupts: 7 (NMI, IRQ0, IRQ1, IRQ4 to IRQ7 pins) • Non-maskable interrupts: 6 (NMI pin, oscillation stop detection interrupt, voltage monitoring 1 interrupt, WDT interrupt, IWDT interrupt, and VBATT power monitoring interrupt) • 16 levels specifiable for the order of priority
DMA	DMA controller (DMACA)	• 4 channels • Three transfer modes: Normal transfer, repeat transfer, and block transfer • Activation sources: Software trigger, external interrupts, and interrupt requests from peripheral functions
	Data transfer controller (DTCa)	• Transfer modes: Normal transfer, repeat transfer, and block transfer • Activation sources: Interrupts • Chain transfer function
I/O ports	General I/O ports	85-pin/83-pin/56-pin • I/O: 43/43/29 • Input: 1/1/1 • Pull-up resistors: 43/43/29 • Open-drain outputs: 31/31/24 • 5-V tolerance: 3/3/2
Event link controller (ELC)		• Event signals of 59 types can be directly connected to the module • Operations of timer modules are selectable at event input • Capable of event link operation for port B and port E
Multi-function pin controller (MPC)		Capable of selecting the input/output function from multiple pins
Timers	16-bit timer pulse unit (TPUa)	• (16 bits × 6 channels) × 1 unit • Maximum of 10 pulse-input/output possible • Select from among seven or eight counter-input clock signals for each channel • Supports the input capture/output compare function • Output of PWM waveforms in up to 9 phases in PWM mode • Support for buffered operation, phase-counting mode (two-phase encoder input) and cascade connected operation (32 bits × 2 channels) depending on the channel. • Capable of generating conversion start triggers for the A/D converters • Signals from the input capture pins are input via a digital filter • Clock frequency measuring method
	Multi-function timer pulse unit 2 (MTU2a)	• (16 bits × 5 channels) × 1 unit • Up to 15 pulse-input/output lines are available based on the six 16-bit timer channels • Select from among eight or seven counter-input clock signals for each channel (PCLK/1, PCLK/4, PCLK/16, PCLK/64, PCLK/256, PCLK/1024, MTCLKA, MTCLKB, MTCLKC, MTCLKD). • Input capture function • 18 output compare/input capture registers • Pulse output mode • Complementary PWM output mode • Reset synchronous PWM mode • Phase-counting mode • Capable of generating conversion start triggers for the A/D converter
	Port output enable 2 (POE2a)	Controls the high-impedance state of the MTU's waveform output pins
	Compare match timer (CMT)	• (16 bits × 2 channels) × 2 units • Select from among four clock signals (PCLK/8, PCLK/32, PCLK/128, PCLK/512)
	Watchdog timer (WDTA)	• 14 bits × 1 channel • Select from among six counter-input clock signals (PCLK/4, PCLK/64, PCLK/128, PCLK/512, PCLK/2048, PCLK/8192)

Downloaded from Arrow.com.

Table 1.1 Outline of Specifications (3/5)

Classification	Module/Function	Description
Timers	Independent watchdog timer (IWDTa)	• 14 bits × 1 channel • Count clock: Dedicated low-speed on-chip oscillator for the IWDT Frequency divided by 1, 16, 32, 64, 128, or 256
	Realtime clock (RTCe)	• Clock source: Sub-clock • Time/calendar • Interrupts: Alarm interrupt, periodic interrupt, and carry interrupt • Time-capture facility for two values
	Low power timer (LPT)	• 16 bits × 1 channel • Clock source: Sub-clock, Dedicated low-speed on-chip oscillator for the IWDT Frequency divided by 2, 4, 8, 16, or 32
	8-bit timer (TMR)	• (8 bits × 2 channels) × 2 units • Seven internal clocks (PCLK/1, PCLK/2, PCLK/8, PCLK/32, PCLK/64, PCLK/1024, and PCLK/8192) and an external clock can be selected • Pulse output and PWM output with any duty cycle are available • Two channels can be cascaded and used as a 16-bit timer
Communication functions	Serial communications interfaces (SCIg, SCIh)	• 4 channels (channel 1, 5, 8: SCIg, channel 12: SCIh) • SCIg Serial communications modes: Asynchronous, clock synchronous, and smart-card interface Multi-processor function On-chip baud rate generator allows selection of the desired bit rate Choice of LSB-first or MSB-first transfer Average transfer rate clock can be input from TMR timers for SCI5, and SCI12 Start-bit detection: Level or edge detection is selectable. Simple I^2C Simple SPI 9-bit transfer mode Bit rate modulation Event linking by the ELC (only on channel 5) • SCIh (The following functions are added to SCIg) Supports the serial communications protocol, which contains the start frame and information frame Supports the LIN format
	IrDA interface (IRDA)	• 1 channel (SCI5 used) • Supports encoding/decoding of waveforms conforming to IrDA standard 1.0
	I^2C bus interface (RIICa)	• 1 channel • Communications formats: I^2C bus format/SMBus format • Master mode or slave mode selectable • Supports fast mode
	Serial peripheral interface (RSPIa)	• 1 channel • Transfer facility Using the MOSI (master out, slave in), MISO (master in, slave out), SSL (slave select), and RSPCK (RSPI clock) enables serial transfer through SPI operation (four lines) or clock-synchronous operation (three lines) • Capable of handling serial transfer as a master or slave • Data formats • Choice of LSB-first or MSB-first transfer The number of bits in each transfer can be changed to 8, 9, 10, 11, 12, 13, 14, 15, 16, 20, 24, or 32 bits. 128-bit buffers for transmission and reception Up to four frames can be transmitted or received in a single transfer operation (with each frame having up to 32 bits) • Double buffers for both transmission and reception
	USB 2.0 host/function module (USBc)	• USB Device Controller (UDC) and transceiver for USB 2.0 are incorporated. • Host/function module: 1 port • Compliant with USB version 2.0 • Transfer speed: Full-speed (12 Mbps), low-speed (1.5 Mbps) • OTG (ON-The-Go) is supported. • Isochronous transfer is supported. • BC1.2 (Battery Charging Specification Revision 1.2) is supported.
	CAN module (RSCAN)	• 1 channel • Compliance with the ISO11898-1 specification (standard frame and extended frame) • 16 Message boxes

Downloaded from Arrow.com.

Table 1.1 Outline of Specifications (4/5)

Classification	Module/Function	Description
Communication functions	Serial Sound Interface (SSI)	• 1 channel • Capable of duplex communications • Various serial audio formats supported • Master/slave function supported • Programmable word clock or bit clock generation function • 8/16/18/20/22/24/32-bit data formats supported • On-chip 8-stage FIFO for transmission/reception • Supports WS continue mode in which the SSIWS signal is not stopped.
	SD Host Interface (SDHIa)	• 1 channel • Transfer speed: Default speed mode (8MB/s) • SD memory card interface (1 bit / 4bits SD bus) • MMC, eMMC Backward-compatible are supported. • SD Specifications Part 1: Compliant with Physical Layer Specification Ver.3.01 (Not support DDR) Part E1: SDIO Specification Ver. 3.00 • Error check function: CRC7 (command), CRC16 (data) • Interrupt Source: Card access interrupt, SDIO access interrupt, Card detection interrupt, SD buffer access interrupt • DMA transfer sources: SD_BUF write, SD_BUF read • Card detection, Write protection
	Bluetooth low energy (BLE)	• On-chip RF transceiver and link layer compliant with the Bluetooth 5.0 Low Energy specification • Bit rates: 1 Mbps, 2 Mbps, 500 kbps, and 125 kbps • LE Advertising extension support • Includes an RF transceiver power supply (selectable as a DC-to-DC converter or linear regulator) • On-chip matching circuit to help reduce the number of external parts • Transmission power: +4 dBm support • Small PCB trace antenna • Certified as compliant with radio-related laws • Bluetooth-dedicated clock oscillator
Encryption functions		• Access management circuit • Encryption engine 128- or 256-bit key sizes of AES Block cipher mode of operation: GCM, ECB, CBC, CMAC, XTS, CTR, GCTR • Hash function • True random number generator • Prevention from illicit copying of a key
12-bit A/D converter (S12ADE)		• 12 bits (14 channels × 1 unit) • 12-bit resolution • Minimum conversion time: 0.83 µs per channel when the ADCLK is operating at 54 MHz • Operating modes Scan mode (single scan mode, continuous scan mode, and group scan mode) Group A priority control (only for group scan mode) • Sampling variable Sampling time can be set up for each channel. • Self-diagnostic function • Double trigger mode (A/D conversion data duplicated) • Detection of analog input disconnection • A/D conversion start conditions A software trigger, a trigger from a timer (MTU, TPU), an external trigger signal, or ELC • Event linking by the ELC
Temperature sensor (TEMPSA)		• 1 channel • The voltage output from the temperature sensor is converted into a digital value by the 12-bit A/D converter.
12-bit D/A converter (R12DAA)		• 2 channels • 12-bit resolution • Output voltage: 0.4 to AVCC0-0.5V
CRC calculator (CRC)		• CRC code generation for arbitrary amounts of data in 8-bit units • Select any of three generating polynomials: $X^8 + X^2 + X + 1$, $X^{16} + X^{15} + X^2 + 1$, or $X^{16} + X^{12} + X^5 + 1$ • Generation of CRC codes for use with LSB-first or MSB-first communications is selectable.
Comparator B (CMPBa)		• 2 channels × 1 unit • Function to compare the reference voltage and the analog input voltage • Window comparator operation or standard comparator operation is selectable
Capacitive touch sensing unit (CTSU)		Detection pin: 12 channels
Data operation circuit (DOC)		Comparison, addition, and subtraction of 16-bit data
Power supply voltages/Operating frequencies		VCC = 1.8 to 2.4 V: 8 MHz, VCC = 2.4 to 2.7 V: 16 MHz, VCC = 2.7 to 3.6 V: 54 MHz
Operating temperature range		D version: −40 to +85°C

Downloaded from Arrow.com.

Table 1.1 Outline of Specifications (5/5)

Classification	Module/Function	Description
Packages		85-pin BGA (PTBG0085KB-A) 5.5 × 5.5 mm, 0.5 mm pitch 83-pin LGA (PTLG0083KA-A, PTLG0083KA-B) 6.1 × 9.5 mm, 0.5 mm pitch 56-pin QFN (PVQN0056LA-A) 7 × 7 mm, 0.4 mm pitch
Debugging interfaces		FINE interface

Downloaded from Arrow.com.

Table 1.2 Comparison of Functions for Different Packages

Module/Functions		RX23W Group		
		56 Pins	83 Pins	85 Pins
External bus	External bus	Not supported		
Interrupts	External interrupts	NMI, IRQ0, IRQ1, IRQ4 to IRQ7		
DMA	DMA controller	4 channels (DMAC0 to DMAC3)		
	Data transfer controller	Available		
Timers	16-bit timer pulse unit	5 channels (TPU0 to TPU3, TPU5)	6 channels (TPU0 to TPU5)	
	Multi-function timer pulse unit 2	5 channels (MTU0 to MTU4)		
	Port output enable 2	POE0#, POE8#	POE0#, POE1#, POE3#, POE8#	
	8-bit timer	2 channels × 2 units		
	Compare match timer	2 channels × 2 units		
	Low power timer	1 channel		
	Realtime clock	Available		
	Watchdog timer	Available		
	Independent watchdog timer	Available		
Communication functions	Serial communications interfaces (SCIg)	3 channels (SCI1, 5, 8)		
	IrDA interface	1 channel (SCI5)		
	Serial communications interfaces (SCIh)	Not supported	1 channel (SCI12)	
	I^2C bus interface	1 channel		
	CAN module	1 channel		
	Serial peripheral interface	1 channel		
	USB 2.0 host/function module	1 channel		
	Serial sound interface	1 channel		
	SD Host Interface	Not supported	1 channel	
	Bluetooth low energy	An RF transceiver and link layer compliant with Bluetooth 5.0 low energy specification		
Capacitive touch sensing unit		9 channels	12 channels	
12-bit A/D converter (including high-precision channels)		8 channels (4 channels)	14 channels (8 channels)	
Temperature sensor		Available		
D/A converter		1 channel	2 channels	
CRC calculator		Available		
Event link controller		Available		
Comparator B		2 channels		
RF transceiver power supply		DC-to-DC converter and linear regulator are selectable	Linear regulator	DC-to-DC converter and linear regulator are selectable
Small PCB trace antenna		Not supported	Included	Not supported
Dedicated crystal for the bluetooth 32 MHz		Not supported	Included	Not supported
Certificates of compliance with radio-related laws (technical standards, FCC, ISED, and CE)		—	Confirmed	—
Packages		56-pin QFN	83-pin LGA	85-pin BGA

Downloaded from Arrow.com.

1.2 List of Products

Table 1.3 is a list of products, and Figure 1.1 shows how to read the product part no., memory capacity, and package type.

Table 1.3 List of Products: D Version (T_a = –40 to +85°C)

Group	Part No.	Order Part No.	Package	ROM Capacity	RAM Capacity	E2 DataFlash	Operating Frequency	Security Function	Antenna	Operating Temperature
RX23W	R5F523W8ADBL	R5F523W8ADBL#20	PTBG0085KB-A	512 Kbytes	64 Kbytes	8 Kbytes	54 MHz	Not available	Not included	–40 to +85°C
	R5F523W8CDLN	R5F523W8CDLN#U0	PTLG0083KA-A					Not available	Included	
		R5F523W8CDLN#20	PTLG0083KA-B							
	R5F523W8ADNG	R5F523W8ADNG#30	PVQN0056LA-A					Not available	Not included	
	R5F523W8BDBL	R5F523W8BDBL#20	PTBG0085KB-A					Available	Not included	
	R5F523W8DDLN	R5F523W8DDLN#U0	PTLG0083KA-A					Available	Included	
		R5F523W8DDLN#20	PTLG0083KA-B							
	R5F523W8BDNG	R5F523W8BDNG#30	PVQN0056LA-A					Available	Not included	
	R5F523W7ADBL	R5F523W7ADBL#20	PTBG0085KB-A	384 Kbytes				Not available	Not included	
	R5F523W7ADNG	R5F523W7ADNG#30	PVQN0056LA-A					Not available	Not included	
	R5F523W7BDBL	R5F523W7BDBL#20	PTBG0085KB-A					Available	Not included	
	R5F523W7BDNG	R5F523W7BDNG#30	PVQN0056LA-A					Available	Not included	

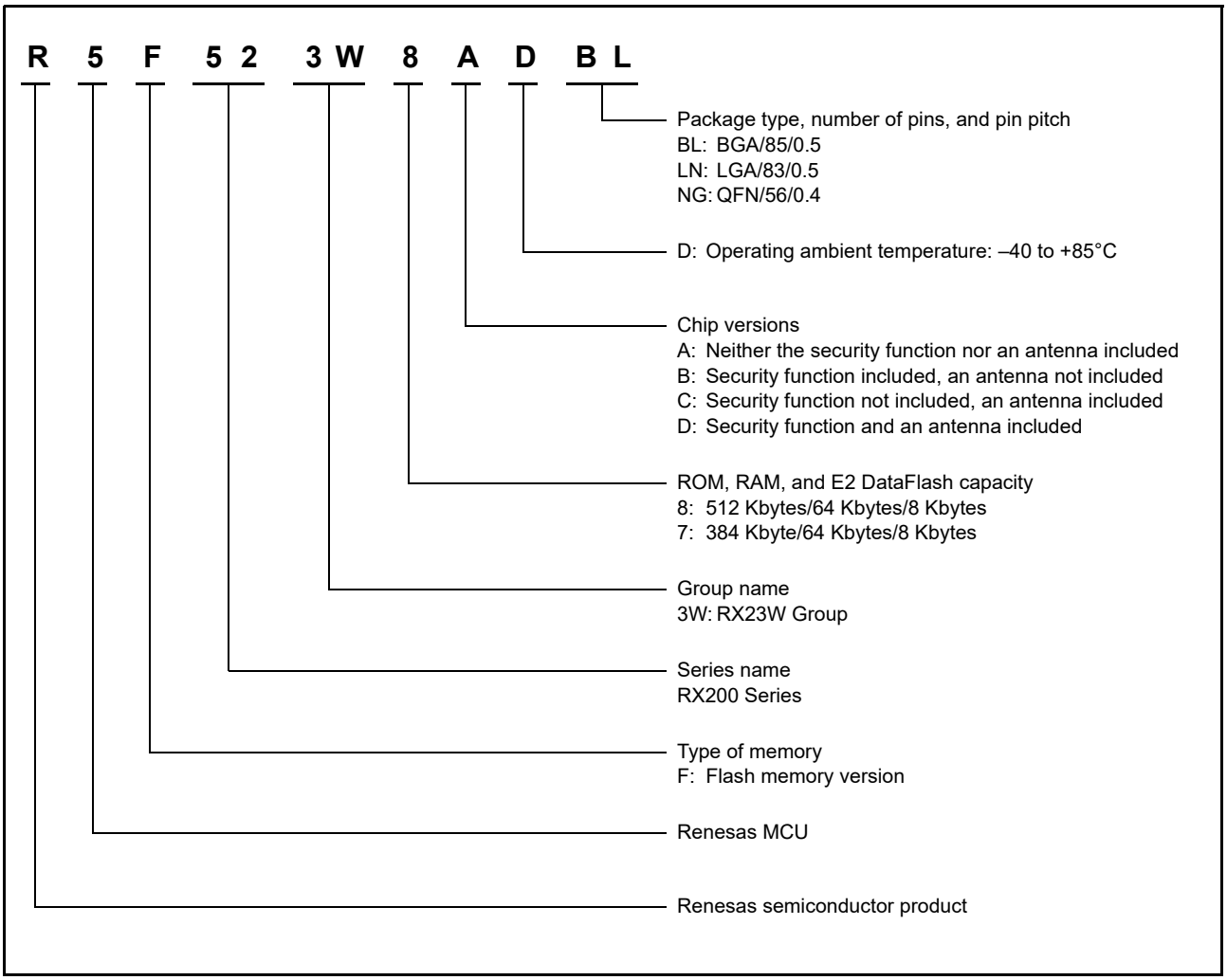
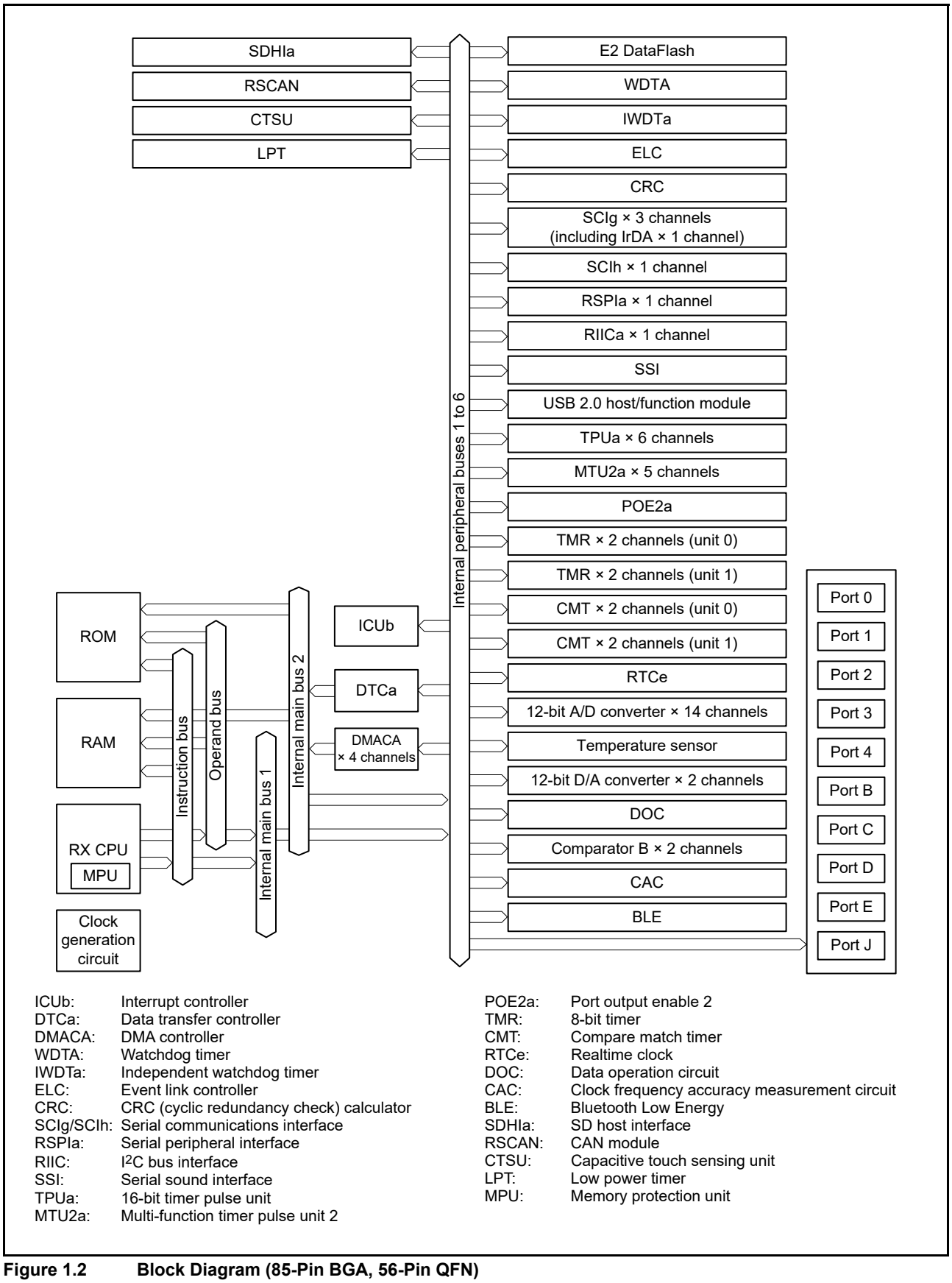


Figure 1.1 How to Read the Product Part Number

Downloaded from Arrow.com.

1.3 Block Diagram

Figure 1.2 shows a block diagram for the 85-pin BGA or 56-pin QFN product.



ICUb: Interrupt controller
DTCa: Data transfer controller
DMACA: DMA controller
WDTA: Watchdog timer
IWDTa: Independent watchdog timer
ELC: Event link controller
CRC: CRC (cyclic redundancy check) calculator
SCIg/SCIh: Serial communications interface
RSPIa: Serial peripheral interface
RIIC: I²C bus interface
SSI: Serial sound interface
TPUa: 16-bit timer pulse unit
MTU2a: Multi-function timer pulse unit 2
POE2a: Port output enable 2
TMR: 8-bit timer
CMT: Compare match timer
RTCe: Realtime clock
DOC: Data operation circuit
CAC: Clock frequency accuracy measurement circuit
BLE: Bluetooth Low Energy
SDHIa: SD host interface
RSCAN: CAN module
CTSU: Capacitive touch sensing unit
LPT: Low power timer
MPU: Memory protection unit

Figure 1.2 Block Diagram (85-Pin BGA, 56-Pin QFN)

Downloaded from Arrow.com.

Figure 1.3 shows a block diagram for the 83-pin LGA product.

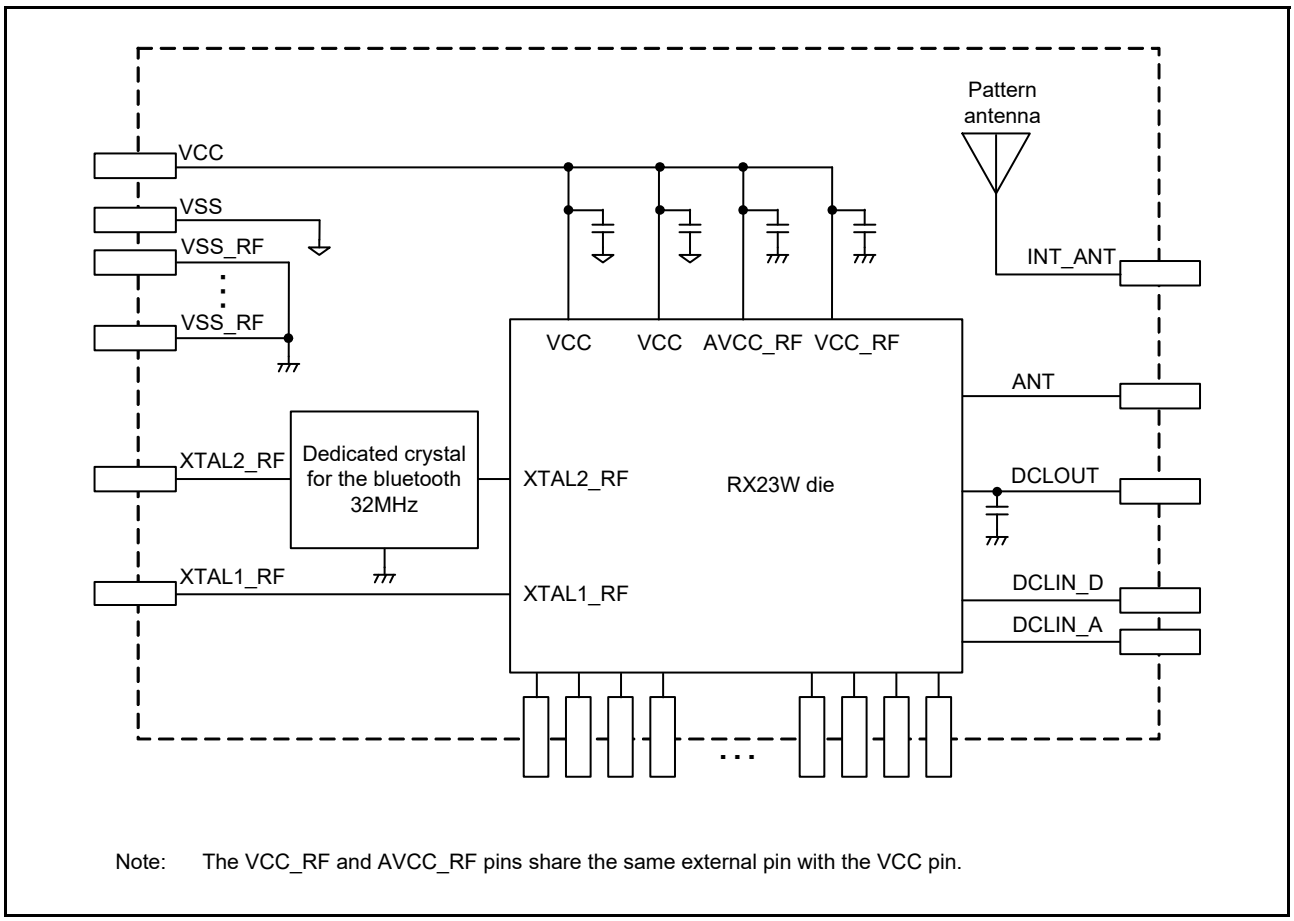


Figure 1.3 Block Diagram (83-Pin LGA)

Downloaded from Arrow.com.

1.4 Pin Functions

Table 1.4 lists the pin functions.

Table 1.4 Pin Functions (1/4)

Classifications	Pin Name	I/O	Description
Power supply	VCC	Input	Power supply pin. Connect it to the system power supply.
	VCL	—	Connect this pin to the VSS pin via a 4.7 μF smoothing capacitor used to stabilize the internal power supply. Place the capacitor close to the pin.
	VSS	Input	Ground pin. Connect it to the system power supply (0 V).
	VBATT	Input	Backup power pin
Clock	XTAL	Output	Pins for connecting a crystal. An external clock can be input through the EXTAL pin.
	EXTAL	Input	
	XCIN	Input	Input/output pins for the sub-clock oscillator. Connect a crystal between XCIN and XCOUT.
	XCOUT	Output	
	CLKOUT_RF	Output	Bluetooth-dedicated clock output pin for output of a 1-, 2-, or 4-MHz signal
	XTAL1_RF	Input	Pins for connecting the Bluetooth-dedicated clock oscillator. Connect a 32-MHz oscillator to these pins. The 83-pin LGA product includes a 32-MHz crystal resonator. XTAL1_RF and XTAL2_R should thus be externally connected to each other.
	XTAL2_RF	Output	
	CLKOUT	Output	Clock output pin.
Operating mode control	MD	Input	Pin for setting the operating mode. The signal levels on this pin must not be changed during operation.
	UB	Input	Pin used for boot mode (USB interface).
	UPSEL	Input	Pin used for boot mode (USB interface).
System control	RES#	Input	Reset pin. This MCU enters the reset state when this signal goes low.
CAC	CACREF	Input	Input pin for the clock frequency accuracy measurement circuit.
On-chip emulator	FINED	I/O	FINE interface pin.
Interrupts	NMI	Input	Non-maskable interrupt request pin.
	IRQ0, IRQ1, IRQ4 to IRQ7	Input	Interrupt request pins.
16-bit timer pulse unit	TIOCB0	I/O	The TGRB0 inputs capture input/output compare output/PWM output pins.
	TIOCB1	I/O	The TGRB1 inputs capture input/output compare output/PWM output pins.
	TIOCB2	I/O	The TGRB2 inputs capture input/output compare output/PWM output pins.
	TIOCA3, TIOCB3, TIOCC3, TIOCD3	I/O	The TGRA3 to TGRD3 input capture input/output compare output/PWM output pins.
	TIOCA4, TIOCB4	I/O	The TGRA4 and TGRB4 input capture input/output compare output/PWM output pins.
	TIOCB5	I/O	The TGRB5 inputs capture input/output compare output/PWM output pins.
	TCLKA, TCLKB, TCLKC, TCLKD	Input	Input pins for external clock signals.
Multi-function timer pulse unit 2	MTIOC0A, MTIOC0B, MTIOC0C	I/O	The TGRA0 to TGRC0 input capture input/output compare output/PWM output pins.
	MTIOC1A, MTIOC1B	I/O	The TGRA1 and TGRB1 input capture input/output compare output/PWM output pins.
	MTIOC2A, MTIOC2B	I/O	The TGRA2 and TGRB2 input capture input/output compare output/PWM output pins.
	MTIOC3A, MTIOC3B, MTIOC3C, MTIOC3D	I/O	The TGRA3 to TGRD3 input capture input/output compare output/PWM output pins.
	MTIOC4A, MTIOC4B, MTIOC4C, MTIOC4D	I/O	The TGRA4 to TGRD4 input capture input/output compare output/PWM output pins.
	MTCLKA, MTCLKB, MTCLKC, MTCLKD	Input	Input pins for the external clock.

Downloaded from Arrow.com.

Table 1.4 Pin Functions (2/4)

Classifications	Pin Name	I/O	Description
Port output enable 2	POE0#, POE1#, POE3#, POE8#	Input	Input pins for request signals to place the MTU pins in the high impedance state.
Realtime clock	RTCOUT	Output	Output pin for the 1-Hz/64-Hz clock.
	RTCIC0, RTCIC1	Input	Time capture event input pins.
8-bit timer	TMO0 to TMO2	Output	Compare match output pins.
	TMCI0 to TMCI3	Input	Input pins for the external clock to be input to the counter.
	TMRI1 to TMRI3	Input	Counter reset input pins.
Serial communications interface (SCIg)	• Asynchronous mode/clock synchronous mode		
	SCK1, SCK5, SCK8	I/O	Input/output pins for the clock.
	RXD1, RXD5, RXD8	Input	Input pins for received data.
	TXD1, TXD5, TXD8	Output	Output pins for transmitted data.
	CTS1#, CTS5#, CTS8#	Input	Input pins for controlling the start of transmission and reception.
	RTS1#, RTS5#, RTS8#	Output	Output pins for controlling the start of transmission and reception.
	• Simple I^2C mode		
	SSCL1, SSCL5, SSCL8	I/O	Input/output pins for the I^2C clock.
	SSDA1, SSDA5, SSDA8	I/O	Input/output pins for the I^2C data.
	• Simple SPI mode		
	SCK1, SCK5, SCK8	I/O	Input/output pins for the clock.
	SMISO1, SMISO5, SMISO8	I/O	Input/output pins for slave transmit data.
	SMOSI1, SMOSI5, SMOSI8	I/O	Input/output pins for master transmit data.
	SS1#, SS5#, SS8#	Input	Slave-select input pins.
IrDA interface	IRTXD5	Output	Data output pin in the IrDA format.
	IRRXD5	Input	Data input pin in the IrDA format.
Serial communications interface (SCIh)	• Asynchronous mode/clock synchronous mode		
	SCK12	I/O	Input/output pin for the clock.
	RXD12	Input	Input pin for receiving data.
	TXD12	Output	Output pin for transmitting data.
	CTS12#	Input	Input pin for controlling the start of transmission and reception.
	RTS12#	Output	Output pin for controlling the start of transmission and reception.
	• Simple I^2C mode		
	SSCL12	I/O	Input/output pin for the I^2C clock.
	SSDA12	I/O	Input/output pin for the I^2C data.
	• Simple SPI mode		
	SCK12	I/O	Input/output pin for the clock.
	SMISO12	I/O	Input/output pin for slave transmit data.
	SMOSI12	I/O	Input/output pin for master transmit data.
	SS12#	Input	Slave-select input pin.
	• Extended serial mode		
	RXDX12	Input	Input pin for data reception by SCIf.
	TXDX12	Output	Output pin for data transmission by SCIf.
	SIOX12	I/O	Input/output pin for data reception or transmission by SCIf.
I^2C bus interface	SCL	I/O	Input/output pin for I^2C bus interface clocks. Bus can be directly driven by the N-channel open drain output.
	SDA	I/O	Input/output pin for I^2C bus interface data. Bus can be directly driven by the N-channel open drain output.

Downloaded from Arrow.com.

Table 1.4 Pin Functions (3/4)

Classifications	Pin Name	I/O	Description
Serial peripheral interface	RSPCKA	I/O	Input/output pin for the RSPI clock.
	MOSIA	I/O	Input/output pin for transmitting data from the RSPI master.
	MISOA	I/O	Input/output pin for transmitting data from the RSPI slave.
	SSLA0	I/O	Input/output pin to select the slave for the RSPI.
	SSLA1, SSLA3	Output	Output pins to select the slave for the RSPI.
Serial sound interface	SSISCK0	I/O	SSI serial bit clock pin.
	SSIWS0	I/O	Word selection pin.
	SSITXD0	Output	Serial data output pin.
	SSIRXD0	Input	Serial data input pin.
	AUDIO_MCLK	Input	Master clock pin for audio.
CAN module	CRXD0	Input	Input pin
	CTXD0	Output	Output pin
SD host interface	SDHI_CLK	Output	SD clock output pin
	SDHI_CMD	I/O	SD command output, response input signal pin
	SDHI_D3 to SDHI_D0	I/O	SD data bus pins
	SDHI_CD	Input	SD card detection pin
	SDHI_WP	Input	SD write-protect signal
USB 2.0 host/ function module	VCC_USB	Input	Power supply pin for USB. Connect this pin to VCC or connect this pin to VSS via a 0.33 µF smoothing capacitor for stabilizing the internal power supply.
	VSS_USB	Input	Ground pin for USB. Connect this pin to VSS.
	USB0_DP	I/O	D+ I/O pin of the USB on-chip transceiver.
	USB0_DM	I/O	D- I/O pin of the USB on-chip transceiver.
	USB0_VBUS	Input	USB cable connection monitor pin.
	USB0_EXICEN	Output	Low-power control signal for the OTG chip.
	USB0_VBUSEN	Output	VBUS (5 V) supply enable signal for the OTG chip.
	USB0_OVRCURA, USB0_OVRCURB	Input	External overcurrent detection pins.
	USB0_ID	Input	Mini-AB connector ID input pin during operation in OTG mode.
12-bit A/D converter	AN000 to AN007, AN016 to AN020, AN027	Input	Input pins for the analog signals to be processed by the A/D converter.
	ADTRG0#	Input	Input pin for the external trigger signal that start the A/D conversion.
12-bit D/A converter	DA0, DA1	Output	Analog output pins of the D/A converter.
Comparator B	CMPB2, CMPB3	Input	Input pin for the analog signal to be processed by comparator B.
	CVREFB2, CVREFB3	Input	Analog reference voltage supply pin for comparator B.
	CMPOB2, CMPOB3	Output	Output pin for comparator B.
CTSU	TS2 to TS4, TS7, TS8, TS12, TS13, TS22, TS23, TS27, TS30, TS35	Output	Electrostatic capacitance measurement pins (touch pins).
	TSCAP	Output	LPF connection pin.
Analog power supply	AVCC0	Input	Analog voltage supply pin for the 12-bit A/D converter and D/A converter. Connect this pin to VCC when not using the 12-bit A/D converter and D/A converter.
	AVSS0	Input	Analog ground pin for the 12-bit A/D converter and D/A converter. Connect this pin to VSS when not using the 12-bit A/D converter and D/A converter.
	VREFH0	Input	Analog reference voltage supply pin for the 12-bit A/D converter.
	VREFL0	Input	Analog reference ground pin for the 12-bit A/D converter.

Downloaded from Arrow.com.

Table 1.4 Pin Functions (4/4)

Classifications	Pin Name	I/O	Description
I/O ports	P03, P05, P07	I/O	3-bit input/output pins.
	P14 to P17	I/O	4-bit input/output pins.
	P21, P22, P25 to P27	I/O	5-bit input/output pins.
	P30, P31, P35 to P37	I/O	5-bit input/output pins (P35 input pin).
	P40 to P47	I/O	8-bit input/output pins.
	PB0, PB1, PB3, PB5, PB7	I/O	5-bit input/output pins.
	PC0, PC2 to PC7	I/O	7-bit input/output pins.
	PD3	I/O	1-bit input/output pins.
	PE0 to PE4	I/O	5-bit input/output pins.
	PJ3	I/O	1-bit input/output pin.
Bluetooth low energy	ANT	I/O	RF single I/O pin for RF transceiver Set the impedance of the signal line to 50 Ω.
	INT_ANT	I/O	Internal antenna connection pin Externally connect this pin to the ANT pin.
	DCLOUT	Output	RF transceiver power-supply output pin
	DCLIN_A, DCLIN_D	Input	RF transceiver power-supply output connection pin In the case of an 83-pin LGA product, these pins should be externally connected to the DCLOUT pin.
	VCC_RF	Input	RF transceiver power supply pin
	AVCC_RF	Input	RF transceiver power supply pin
	VSS_RF	Input	RF transceiver ground pin

Downloaded from Arrow.com.

1.5 Pin Assignments

1.5.1 85-Pin BGA

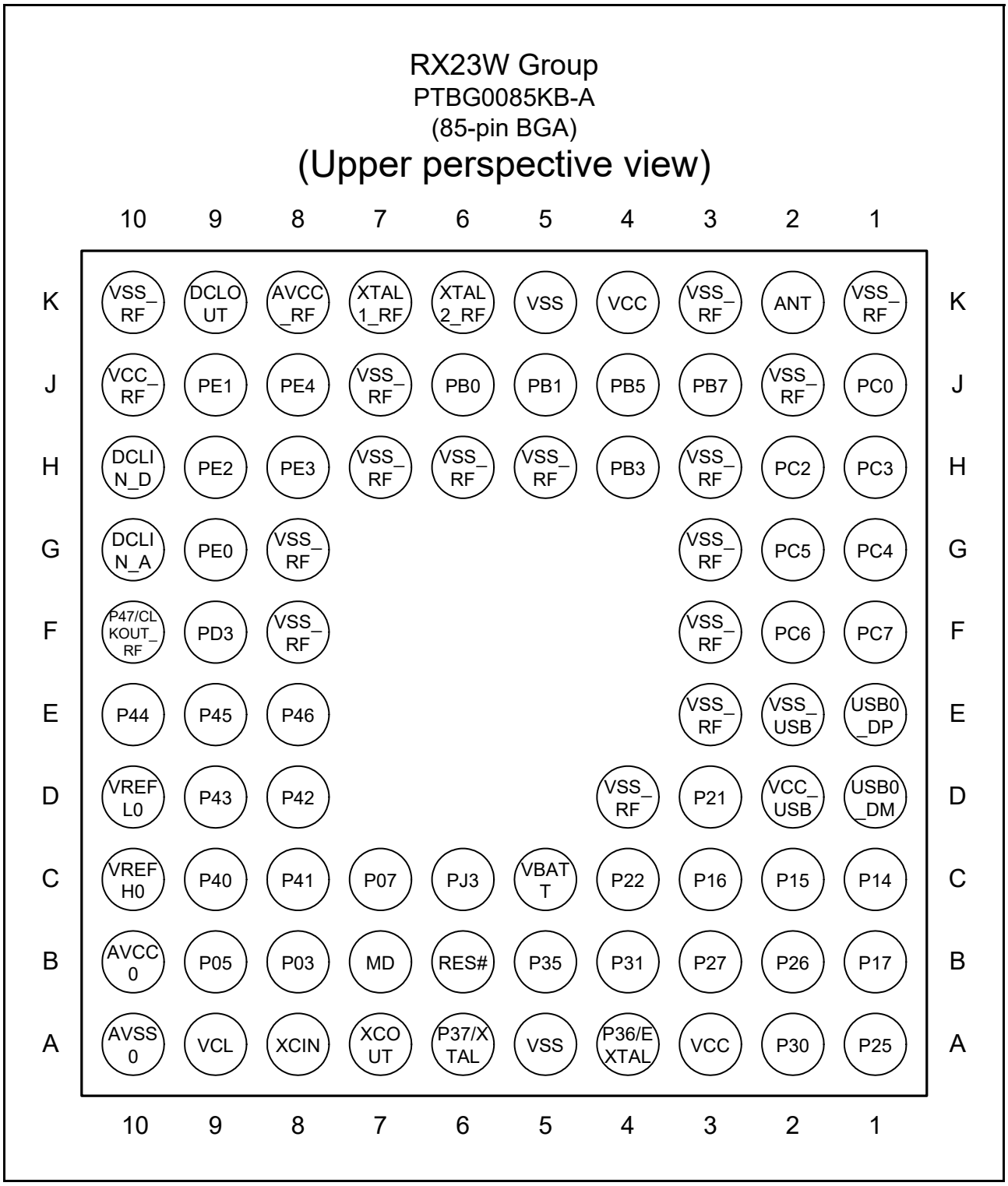


Figure 1.4 Pin Assignments of the 85-Pin BGA

Downloaded from Arrow.com.

1.5.2 83-Pin LGA

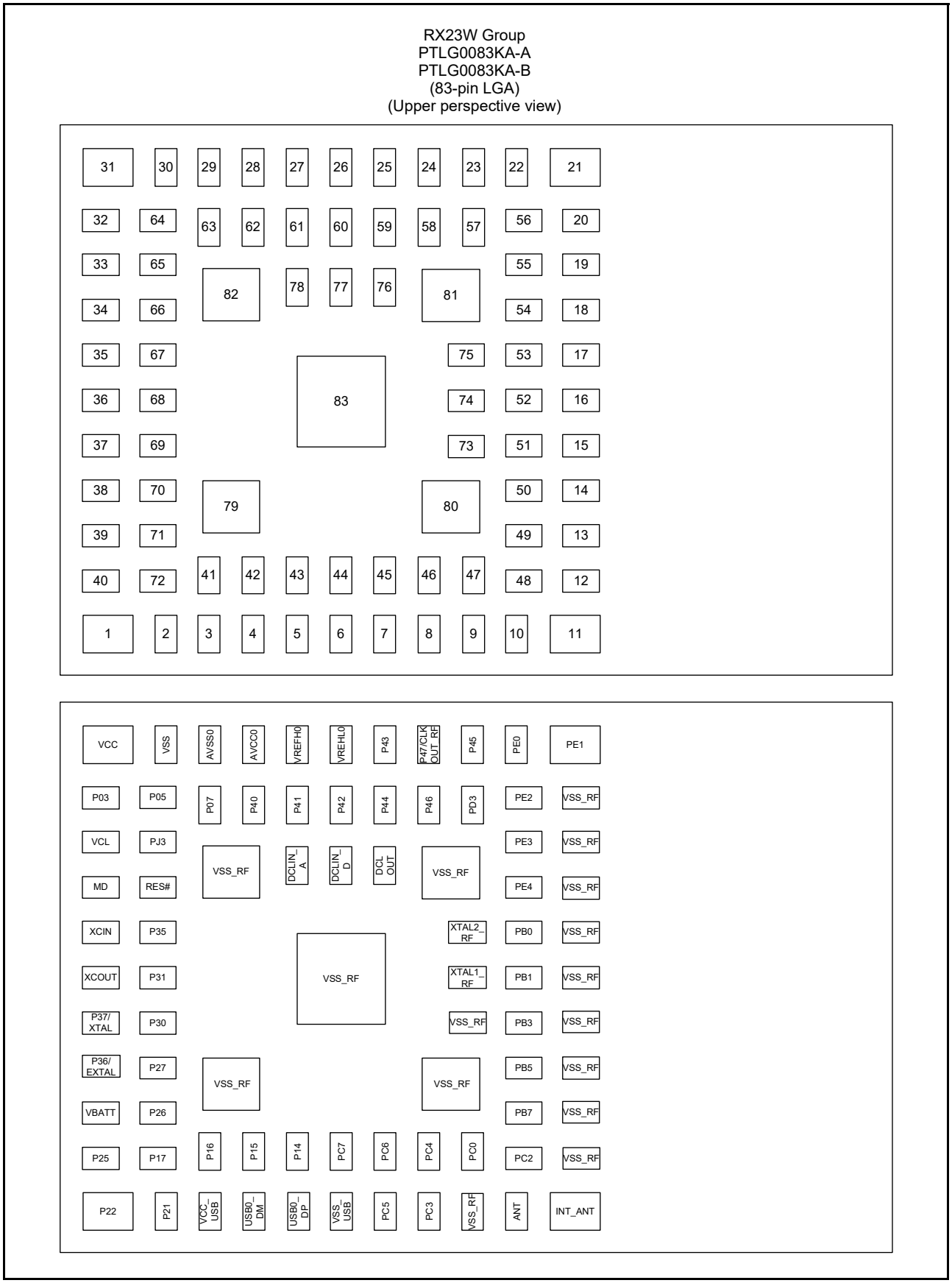
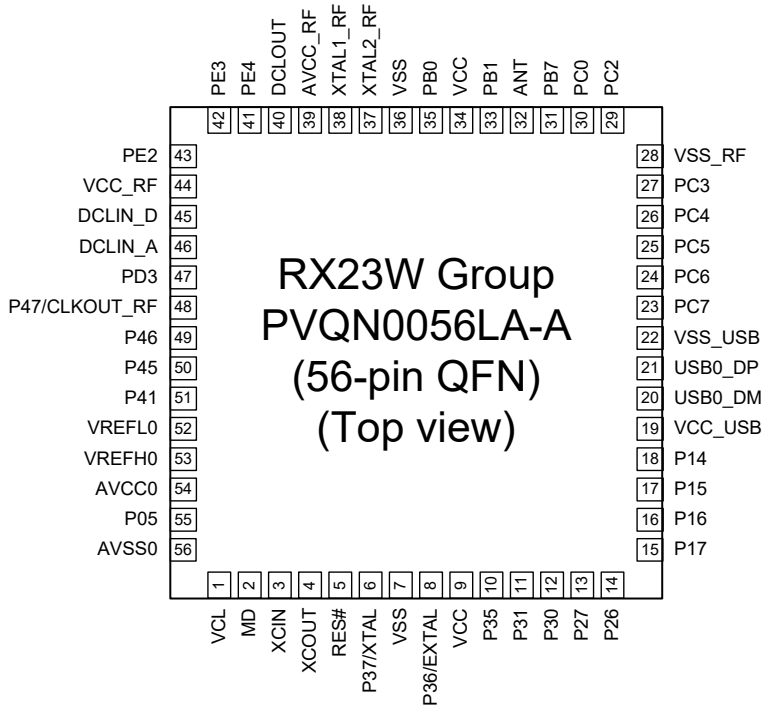


Figure 1.5 Pin Assignments of the 83-Pin LGA

Downloaded from Arrow.com.

1.5.3 56-Pin QFN



Note: VSS_RF is assigned as the exposed die pad. For details, refer to Appendix 2, Package Dimensions.

Figure 1.6 Pin Assignments of the 56-Pin QFN

Downloaded from Arrow.com.

1.6 List of Pins and Pin Functions

1.6.1 85-Pin BGA

Table 1.5 List of Pins and Pin Functions (85-Pin BGA) (1/2)

Pin No.	Power Supply, Clock, System Control	I/O Port	Timers (MTU, TPU, TMR, RTC, CMT, POE, CAC)	Communications (SCI, RSPI, RIIC, RSCAN, USB, SSI)	Memory Interface (SDHI)	Touch sensing	Others
A1		P25	MTIOC4C/MTCLKB/TIOCA4			TS4	ADTRG0#
A2		P30	MTIOC4B/TMRI3/POE8#/ RTCIC0	RXD1/SMISO1/SSCL1/ AUDIO_MCLK			IRQ0/CMPOB3
A3	VCC						
A4	EXTAL	P36					
A5	VSS						
A6	XTAL	P37					
A7	XCOUT						
A8	XCIN						
A9	VCL						
A10	AVSS0						
B1		P17	MTIOC3A/MTIOC3B/TMO1/ POE8#/TIOCB0/TCLKD	SCK1/MISOA/SDA/SSITXD0			IRQ7/CMPOB2
B2		P26	MTIOC2A/TMO1	TXD1/SMOSI1/SSDA1/SSIRXD0/ USB0_VBUSEN		TS3	CMPB3
B3		P27	MTIOC2B/TMCI3	SCK1/SSIWS0		TS2	CVREFB3
B4		P31	MTIOC4D/TMCI2/RTCIC1	CTS1#/RTS1#/SS1#/SSISCK0			IRQ1
B5	UPSEL	P35					NMI
B6	RES#						
B7	MD						FINED
B8		P03					DA0
B9		P05					DA1
B10	AVCC0						
C1		P14	MTIOC3A/MTCLKA/TMRI2/ TIOCB5/TCLKA	CTS1#/RTS1#/SS1#/CTXD0/ USB0_OVRCURA		TS13	IRQ4/ CVREFB2
C2		P15	MTIOC0B/MTCLKB/TMCI2/ TIOCB2/TCLKB	RXD1/SMISO1/SSCL1/CRXD0		TS12	IRQ5/CMPB2
C3		P16	MTIOC3C/MTIOC3D/TMO2/ TIOCB1/TCLKC/RTCOUT	TXD1/SMOSI1/SSDA1/MOSIA/SCL/ USB0_VBUS/USB0_VBUSEN/ USB0_OVRCURB			IRQ6/ ADTRG0#
C4		P22	MTIOC3B/MTCLKC/TMO0/ TIOCC3	USB0_OVRCURB/AUDIO_MCLK		TS7	
C5	VBATT						
C6		PJ3	MTIOC3C				
C7		P07					ADTRG0#
C8		P41					AN001
C9		P40					AN000
C10	VREFH0						
D1				USB0_DM			
D2	VCC_USB						
D3		P21	MTIOC1B/TMCI0/TIOCA3	USB0_EXICEN/SSIWS0		TS8	
D4	VSS_RF						
D8		P42					AN002
D9		P43					AN003
D10	VREFL0						
E1				USB0_DP			
E2	VSS_USB						
E3	VSS_RF						
E8		P46					AN006
E9		P45					AN005
E10		P44					AN004
F1	UB	PC7	MTIOC3A/MTCLKB/TMO2	TXD8/SMOSI8/SSDA8/MISOA			CACREF

Downloaded from Arrow.com.

Table 1.5 List of Pins and Pin Functions (85-Pin BGA) (2/2)

Pin No.	Power Supply, Clock, System Control	I/O Port	Timers (MTU, TPU, TMR, RTC, CMT, POE, CAC)	Communications (SCI, RSPI, RIIC, RSCAN, USB, SSI)	Memory Interface (SDHI)	Touch sensing	Others
F2		PC6	MTIOC3C/MTCLKA/TMCI2	RXD8/SMISO8/SSCL8/MOSIA/ USB0_EXICEN		TS22	
F3	VSS_RF						
F8	VSS_RF						
F9		PD3	POE8#				AN027
F10		P47					AN007/ CLKOUT_RF
G1		PC4	MTIOC3D/MTCLKC/TMCI1/ POE0#	CTS8#/RTS8#/SS8#/SSLA0/SCK5	SDHI_D1	TSCAP	
G2		PC5	MTIOC3B/MTCLKD/TMRI2	SCK8/RSPCKA/USB0_ID		TS23	
G3	VSS_RF						
G8	VSS_RF						
G9		PE0		SCK12			AN016
G10	DCLIN_A						
H1		PC3	MTIOC4D/TCLKB	TXD5/SMOSI5/SSDA5/IRTXD5	SDHI_D0	TS27	
H2		PC2	MTIOC4B/TCLKA	RXD5/SMISO5/SSCL5/SSLA3/ IRRXD5	SDHI_D3	TS30	
H3	VSS_RF						
H4		PB3	MTIOC0A/MTIOC4A/TMO0/ POE3#/TIOCD3/TCLKD		SDHI_WP		
H5	VSS_RF						
H6	VSS_RF						
H7	VSS_RF						
H8		PE3	MTIOC4B/POE8#	CTS12#/RTS12#/SS12#/ AUDIO_MCLK			AN019/ CLKOUT
H9		PE2	MTIOC4A	RXD12/RXDX12/SMISO12/SSCL12			IRQ7/AN018
H10	DCLIN_D						
J1		PC0	MTIOC3C/TCLKC	CTS5#/RTS5#/SS5#/SSLA1		TS35	
J2	VSS_RF						
J3		PB7	MTIOC3B/TIOCB5		SDHI_D2		
J4		PB5	MTIOC2A/MTIOC1B/TMRI1/ POE1#/TIOCB4	USB0_VBUS	SDHI_CD		
J5		PB1	MTIOC0C/MTIOC4C/TMCI0/ TIOCB3		SDHI_CLK		IRQ4
J6		PB0	TIOCA3	RSPCKA	SDHI_CMD		
J7	VSS_RF						
J8		PE4	MTIOC4D/MTIOC1A				AN020/ CLKOUT
J9		PE1	MTIOC4C	TXD12/TXDX12/SIOX12/SMOSI12/ SSDA12			AN017
J10	VCC_RF						
K1	VSS_RF						
K2							ANT
K3	VSS_RF						
K4	VCC						
K5	VSS						
K6	XTAL2_RF						
K7	XTAL1_RF						
K8	AVCC_RF						
K9	DCLOUT						
K10	VSS_RF						

Downloaded from Arrow.com.

1.6.2 83-Pin LGA

Table 1.6 List of Pins and Pin Functions (83-Pin LGA) (1/2)

Pin No.	Power Supply, Clock, System Control	I/O Port	Timers (MTU, TPU, TMR, RTC, CMT, POE, CAC)	Communications (SCI, RSPI, RIIC, RSCAN, USB, SSI)	Memory Interface (SDHI)	Touch sensing	Others
1		P22	MTIOC3B/MTCLKC/TMO0/ TIOCC3	USB0_OVRCURB/AUDIO_MCLK		TS7	
2		P21	MTIOC1B/TMCI0/TIOCA3	USB0_EXICEN/SSIWS0		TS8	
3	VCC_USB						
4				USB0_DM			
5				USB0_DP			
6	VSS_USB						
7		PC5	MTIOC3B/MTCLKD/TMRI2	SCK8/RSPCKA/USB0_ID		TS23	
8		PC3	MTIOC4D/TCLKB	TXD5/SMOSI5/SSDA5/IRTXD5	SDHI_D0	TS27	
9	VSS_RF						
10							ANT
11							INT_ANT
12	VSS_RF						
13	VSS_RF						
14	VSS_RF						
15	VSS_RF						
16	VSS_RF						
17	VSS_RF						
18	VSS_RF						
19	VSS_RF						
20	VSS_RF						
21		PE1	MTIOC4C	TXD12/TXDX12/SIOX12/SMOSI12/ SSDA12			AN017
22		PE0		SCK12			AN016
23		P45					AN005
24		P47					AN007/ CLKOUT_RF
25		P43					AN003
26	VREFL0						
27	VREFH0						
28	AVCC0						
29	AVSS0						
30	VSS						
31	VCC						
32		P03					DA0
33	VCL						
34	MD						FINED
35	XCIN						
36	XCOUT						
37	XTAL	P37					
38	EXTAL	P36					
39	VBATT						
40		P25	MTIOC4C/MTCLKB/TIOCA4			TS4	ADTRG0#
41		P16	MTIOC3C/MTIOC3D/TMO2/ TIOCB1/TCLKC/RTCOUT	TXD1/SMOSI1/SSDA1/MOSIA/SCL/ USB0_VBUS/USB0_VBUSEN/ USB0_OVRCURB			IRQ6/ ADTRG0#
42		P15	MTIOC0B/MTCLKB/TMCI2/ TIOCB2/TCLKB	RXD1/SMISO1/SSCL1/CRXD0		TS12	IRQ5/CMPB2
43		P14	MTIOC3A/MTCLKA/TMRI2/ TIOCB5/TCLKA	CTS1#/RTS1#/SS1#/CTXD0/ USB0_OVRCURA		TS13	IRQ4/ CVREFB2
44	UB	PC7	MTIOC3A/MTCLKB/TMO2	TXD8/SMOSI8/SSDA8/MISOA			CACREF
45		PC6	MTIOC3C/MTCLKA/TMCI2	RXD8/SMISO8/SSCL8/MOSIA/ USB0_EXICEN		TS22	
46		PC4	MTIOC3D/MTCLKC/TMCI1/ POE0#	CTS8#/RTS8#/SS8#/SSLA0/SCK5	SDHI_D1	TSCAP	

Downloaded from Arrow.com.

Table 1.6 List of Pins and Pin Functions (83-Pin LGA) (2/2)

Pin No.	Power Supply, Clock, System Control	I/O Port	Timers (MTU, TPU, TMR, RTC, CMT, POE, CAC)	Communications (SCI, RSPI, RIIC, RSCAN, USB, SSI)	Memory Interface (SDHI)	Touch sensing	Others
47		PC0	MTIOC3C/TCLKC	CTS5#/RTS5#/SS5#/SSLA1		TS35	
48		PC2	MTIOC4B/TCLKA	RXD5/SMISO5/SSCL5/SSLA3/ IRRXD5	SDHI_D3	TS30	
49		PB7	MTIOC3B/TIOCB5		SDHI_D2		
50		PB5	MTIOC2A/MTIOC1B/TMRI1/ POE1#/TIOCB4	USB0_VBUS	SDHI_CD		
51		PB3	MTIOC0A/MTIOC4A/TMO0/ POE3#/TIOCD3/TCLKD		SDHI_WP		
52		PB1	MTIOC0C/MTIOC4C/TMCI0/ TIOCB3		SDHI_CLK		IRQ4
53		PB0	TIOCA3	RSPCKA	SDHI_CMD		
54		PE4	MTIOC4D/MTIOC1A				AN020/ CLKOUT
55		PE3	MTIOC4B/POE8#	CTS12#/RTS12#/SS12#/ AUDIO_MCLK			AN019/ CLKOUT
56		PE2	MTIOC4A	RXD12/RXDX12/SMISO12/SSCL12			IRQ7/AN018
57		PD3	POE8#				AN027
58		P46					AN006
59		P44					AN004
60		P42					AN002
61		P41					AN001
62		P40					AN000
63		P07					ADTRG0#
64		P05					DA1
65		PJ3	MTIOC3C				
66	RES#						
67	UPSEL	P35					NMI
68		P31	MTIOC4D/TMCI2/RTCIC1	CTS1#/RTS1#/SS1#/SSISCK0			IRQ1
69		P30	MTIOC4B/TMRI3/POE8#/ RTCIC0	RXD1/SMISO1/SSCL1/ AUDIO_MCLK			IRQ0/CMPOB3
70		P27	MTIOC2B/TMCI3	SCK1/SSIWS0		TS2	CVREFB3
71		P26	MTIOC2A/TMO1	TXD1/SMOSI1/SSDA1/SSIRXD0/ USB0_VBUSEN		TS3	CMPB3
72		P17	MTIOC3A/MTIOC3B/TMO1/ POE8#/TIOCB0/TCLKD	SCK1/MISOA/SDA/SSITXD0			IRQ7/CMPOB2
73	VSS_RF						
74	XTAL1_RF						
75	XTAL2_RF						
76	DCLOUT						
77	DCLIN_D						
78	DCLIN_A						
79	VSS_RF						
80	VSS_RF						
81	VSS_RF						
82	VSS_RF						
83	VSS_RF						

Downloaded from Arrow.com.

1.6.3 56-Pin QFN

Table 1.7 List of Pins and Pin Functions (56-Pin QFN) (1/2)

Pin No.	Power Supply, Clock, System Control	I/O Port	Timers (MTU, TPU, TMR, RTC, CMT, POE, CAC)	Communications (SCI, RSPI, RIIC, RSCAN, USB, SSI)	Touch sensing	Others
1	VCL					
2	MD					FINED
3	XCIN					
4	XCOUT					
5	RES#					
6	XTAL	P37				
7	VSS					
8	EXTAL	P36				
9	VCC					
10	UPSEL	P35				NMI
11		P31	MTIOC4D/TMCI2/RTCIC1	CTS1#/RTS1#/SS1#/SSISCK0		IRQ1
12		P30	MTIOC4B/TMRI3/POE8#/RTCIC0	RXD1/SMISO1/SSCL1/AUDIO_MCLK		IRQ0/CMPOB3
13		P27	MTIOC2B/TMCI3	SCK1/SSIWS0	TS2	CVREFB3
14		P26	MTIOC2A/TMO1	TXD1/SMOSI1/SSDA1/SSIRXD0/ USB0_VBUSEN	TS3	CMPB3
15		P17	MTIOC3A/MTIOC3B/TMO1/POE8#/ TIOCB0/TCLKD	SCK1/MISOA/SDA/SSITXD0		IRQ7/CMPOB2
16		P16	MTIOC3C/MTIOC3D/TMO2/TIOCB1/ TCLKC/RTCOUT	TXD1/SMOSI1/SSDA1/MOSIA/SCL/ USB0_VBUS/USB0_VBUSEN/ USB0_OVRCURB		IRQ6/ADTRG0#
17		P15	MTIOC0B/MTCLKB/TMCI2/TIOCB2/ TCLKB	RXD1/SMISO1/SSCL1/CRXD0	TS12	IRQ5/CMPB2
18		P14	MTIOC3A/MTCLKA/TMRI2/TIOCB5/ TCLKA	CTS1#/RTS1#/SS1#/CTXD0/ USB0_OVRCURA	TS13	IRQ4/CVREFB2
19	VCC_USB					
20				USB0_DM		
21				USB0_DP		
22	VSS_USB					
23	UB	PC7	MTIOC3A/MTCLKB/TMO2	TXD8/SMOSI8/SSDA8/MISOA		CACREF
24		PC6	MTIOC3C/MTCLKA/TMCI2	RXD8/SMISO8/SSCL8/MOSIA/ USB0_EXICEN	TS22	
25		PC5	MTIOC3B/MTCLKD/TMRI2	SCK8/RSPCKA/USB0_ID	TS23	
26		PC4	MTIOC3D/MTCLKC/TMCI1/POE0#	CTS8#/RTS8#/SS8#/SSLA0/SCK5	TSCAP	
27		PC3	MTIOC4D/TCLKB	TXD5/SMOSI5/SSDA5/IRTXD5	TS27	
28	VSS_RF					
29		PC2	MTIOC4B/TCLKA	RXD5/SMISO5/SSCL5/SSLA3/IRRXD5	TS30	
30		PC0	MTIOC3C/TCLKC	CTS5#/RTS5#/SS5#/SSLA1	TS35	
31		PB7	MTIOC3B/TIOCB5			
32						ANT
33		PB1	MTIOC0C/MTIOC4C/TMCI0/TIOCB3			IRQ4
34	VCC					
35		PB0	TIOCA3	RSPCKA		
36	VSS					
37	XTAL2_RF					
38	XTAL1_RF					
39	AVCC_RF					
40	DCLOUT					
41		PE4	MTIOC4D/MTIOC1A			AN020/CLKOUT
42		PE3	MTIOC4B/POE8#	AUDIO_MCLK		AN019/CLKOUT
43		PE2	MTIOC4A			IRQ7/AN018
44	VCC_RF					
45	DCLIN_D					
46	DCLIN_A					
47		PD3	POE8#			AN027

Downloaded from Arrow.com.

Table 1.7 List of Pins and Pin Functions (56-Pin QFN) (2/2)

Pin No.	Power Supply, Clock, System Control	I/O Port	Timers (MTU, TPU, TMR, RTC, CMT, POE, CAC)	Communications (SCI, RSPI, RIIC, RSCAN, USB, SSI)	Touch sensing	Others
48		P47				AN007/ CLKOUT_RF
49		P46				AN006
50		P45				AN005
51		P41				AN001
52	VREFL0					
53	VREFH0					
54	AVCC0					
55		P05				DA1
56	AVSS0					

Note: VSS_RF is assigned as the exposed die pad. For details, refer to Appendix 2, Package Dimensions.

Downloaded from Arrow.com.

2. CPU

The RXv2 instruction set architecture (RXv2) has upward compatibility with the RXv1 instruction set architecture (RXv1).

- Adoption of variable-length instruction format
  As with RXv1, the RXv2 CPU has short formats for frequently used instructions, facilitating the development of efficient programs that take up less memory.
- Powerful instruction set
  The RXv2 supports 109 selected instructions. Moreover, DSP instructions and floating-point operation instructions are added, thus realizing high-speed arithmetic processing.
- Versatile addressing modes
  The RXv2 CPU has 11 versatile addressing modes, with register-register operations, register-memory operations, and bitwise operations included. Data transfer between memory locations is also possible.

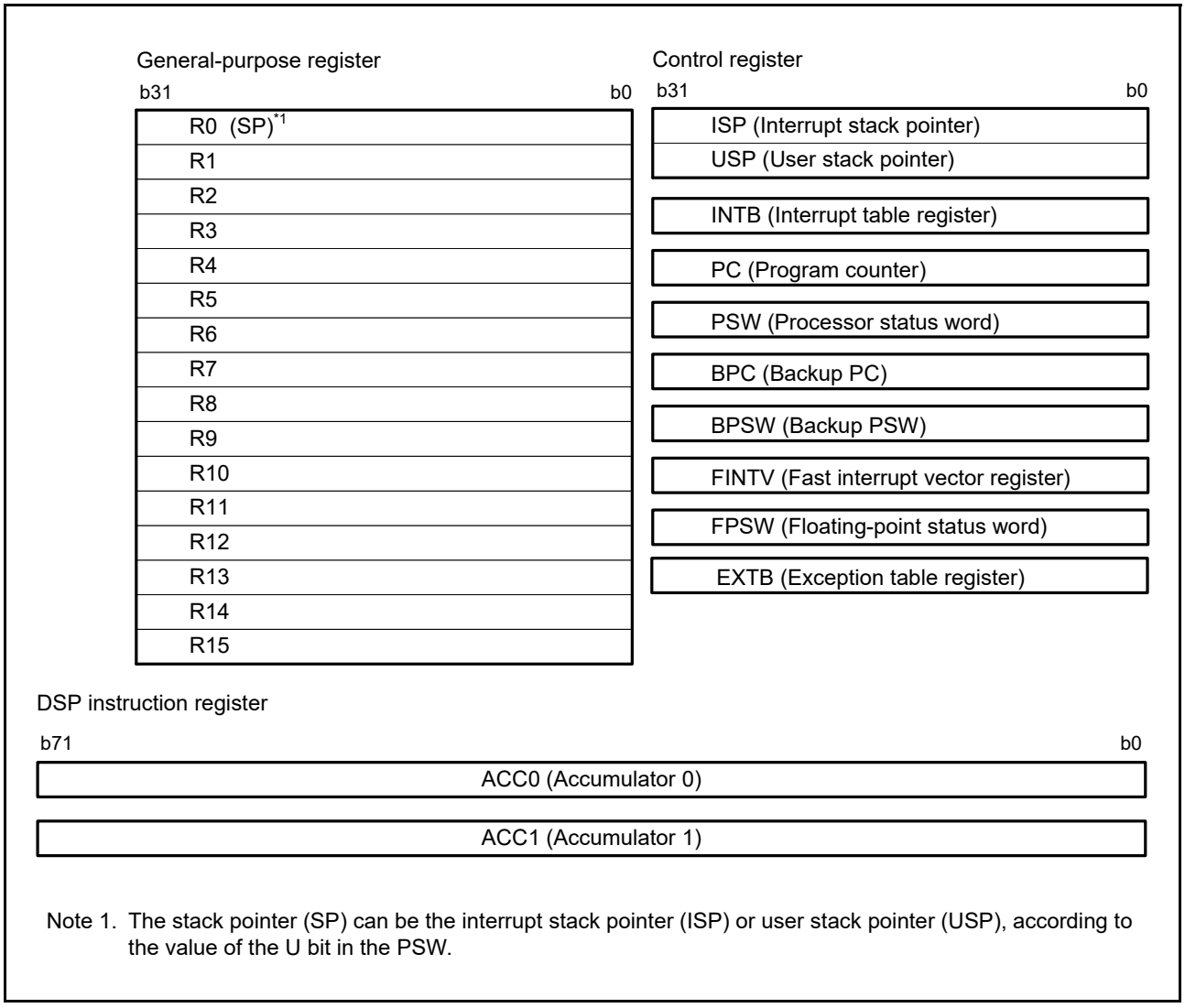
2.1 Features

- Minimum instruction execution rate: One clock cycle
- Address space: 4-Gbyte linear addresses
- Register set of the CPU
  General purpose: Sixteen 32-bit registers
  Control: Ten 32-bit registers
  Accumulator: Two 72-bit registers
- Variable-length instruction format (lengths from one to eight bytes)
- 109 instructions/11 addressing modes
  Basic instructions: 75
  Floating-point operation instructions: 11
  DSP instructions: 23
- Processor modes
  Supervisor mode and user mode
- Vector tables
  Exception vector table and interrupt vector table
- Memory protection unit
- Data arrangement
  Selectable as little endian or big endian

Downloaded from Arrow.com.

2.2 Register Set of the CPU

The RXv2 CPU has sixteen general-purpose registers, ten control registers, and two accumulator used for DSP instructions.



Note 1. The stack pointer (SP) can be the interrupt stack pointer (ISP) or user stack pointer (USP), according to the value of the U bit in the PSW.

Figure 2.1 Register Set of the CPU

Downloaded from Arrow.com.

2.2.1 General-Purpose Registers (R0 to R15)

This CPU has sixteen 32-bit general-purpose registers (R0 to R15). R0 to R15 can be used as data registers or address registers.
R0, a general-purpose register, also functions as the stack pointer (SP).
The stack pointer is switched to operate as the interrupt stack pointer (ISP) or user stack pointer (USP) by the value of the stack pointer select bit (U) in the processor status word (PSW).

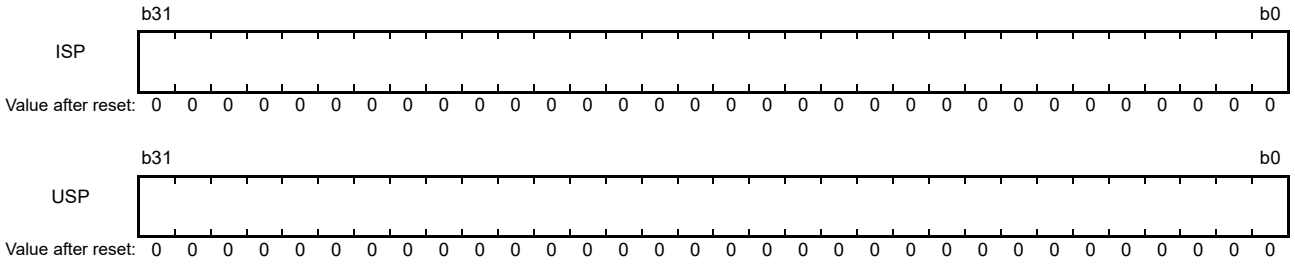
2.2.2 Control Registers

This CPU has the following ten control registers.

- Interrupt stack pointer (ISP)
- User stack pointer (USP)
- Exception table register (EXTB)
- Interrupt table register (INTB)
- Program counter (PC)
- Processor status word (PSW)
- Backup PC (BPC)
- Backup PSW (BPSW)
- Fast interrupt vector register (FINTV)
- Floating-point status word (FPSW)

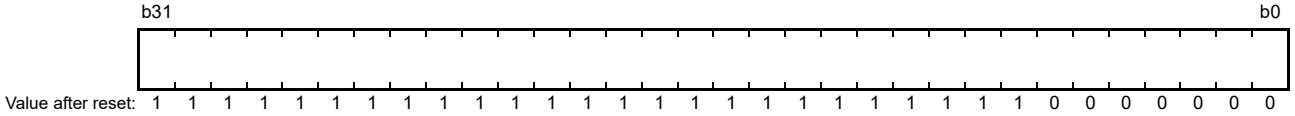
Downloaded from Arrow.com.

2.2.2.1 Interrupt Stack Pointer (ISP)/User Stack Pointer (USP)



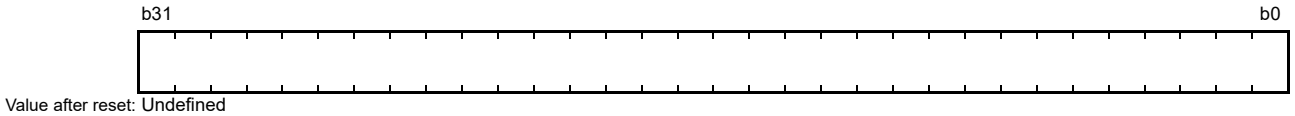
The stack pointer (SP) can be either of two types, the interrupt stack pointer (ISP) or the user stack pointer (USP). Whether the stack pointer operates as the ISP or USP depends on the value of the stack pointer select bit (U) in the processor status word (PSW).
Set the ISP or USP to a multiple of 4 to reduce the number of cycles required to execute interrupt sequences and instructions entailing stack manipulation.

2.2.2.2 Exception Table Register (EXTB)



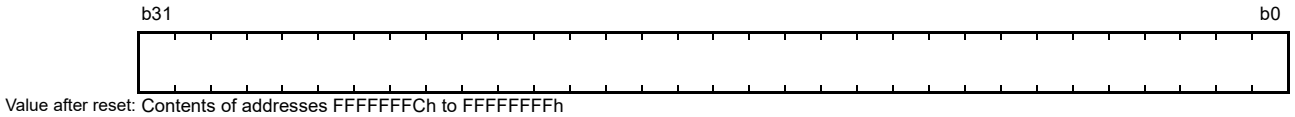
The exception table register (EXTB) specifies the address where the exception vector table starts.
Set the EXTB to a multiple of 4 to reduce the number of cycles required to execute interrupt sequences and instructions entailing stack manipulation.

2.2.2.3 Interrupt Table Register (INTB)



The interrupt table register (INTB) specifies the address where the interrupt vector table starts.
Set the INTB to a multiple of 4 to reduce the number of cycles required to execute interrupt sequences and instructions entailing stack manipulation.

2.2.2.4 Program Counter (PC)



The program counter (PC) indicates the address of the instruction being executed.

Downloaded from Arrow.com.

2.2.2.5 Processor Status Word (PSW)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	IPL[3:0]				—	—	—	PM	—	—	U	I
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	O	S	Z	C
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	C	Carry Flag	0: No carry has occurred. 1: A carry has occurred.	R/W
b1	Z	Zero Flag	0: Result is non-zero. 1: Result is 0.	R/W
b2	S	Sign Flag	0: Result is a positive value or 0. 1: Result is a negative value.	R/W
b3	O	Overflow Flag	0: No overflow has occurred. 1: An overflow has occurred.	R/W
b15 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	I*1	Interrupt Enable	0: Interrupt disabled. 1: Interrupt enabled.	R/W
b17	U*1	Stack Pointer Select	0: Interrupt stack pointer (ISP) is selected. 1: User stack pointer (USP) is selected.	R/W
b19, b18	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b20	PM*1, *2, *3	Processor Mode Select	0: Supervisor mode is selected. 1: User mode is selected.	R/W
b23 to b21	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b27 to b24	IPL[3:0]*1	Processor Interrupt Priority Level	b27 b24 0 0 0 0: Priority level 0 (lowest) 0 0 0 1: Priority level 1 0 0 1 0: Priority level 2 0 0 1 1: Priority level 3 0 1 0 0: Priority level 4 0 1 0 1: Priority level 5 0 1 1 0: Priority level 6 0 1 1 1: Priority level 7 1 0 0 0: Priority level 8 1 0 0 1: Priority level 9 1 0 1 0: Priority level 10 1 0 1 1: Priority level 11 1 1 0 0: Priority level 12 1 1 0 1: Priority level 13 1 1 1 0: Priority level 14 1 1 1 1: Priority level 15 (highest)	R/W
b31 to b28	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. In user mode, writing to the IPL[3:0], PM, U, and I bits by an MVTC or a POPC instruction is ignored. Writing to the IPL[3:0] bits by an MVTIPL instruction generates a privileged instruction exception.
Note 2. In supervisor mode, writing to the PM bit by an MVTC or a POPC instruction is ignored, but writing to the other bits is possible.
Note 3. Switching from supervisor mode to user mode requires execution of an RTE instruction after having set the PSW.PM bit saved on the stack to 1 or executing an RTFI instruction after having set the BPSW.PM bit to 1.

The processor status word (PSW) indicates the results of instruction execution or the state of the CPU.

Downloaded from Arrow.com.

C Flag (Carry Flag)

This flag retains the state of the bit after a carry, borrow, or shift-out has occurred.

Z Flag (Zero Flag)

This flag is set to 1 if the result of an operation is 0; otherwise its value is cleared to 0.

S Flag (Sign Flag)

This flag is set to 1 if the result of an operation is negative; otherwise its value is cleared to 0.

O Flag (Overflow Flag)

This flag is set to 1 if the result of an operation overflows; otherwise its value is cleared to 0.

I Bit (Interrupt Enable)

This bit enables interrupt requests. When a WAIT instruction is executed, the value of this bit becomes 1. It becomes 0 when an exception is accepted.

U Bit (Stack Pointer Select)

This bit specifies the stack pointer as either the ISP or USP. When an exception request is accepted, this bit is set to 0. When the processor mode is switched from supervisor mode to user mode, this bit is set to 1.

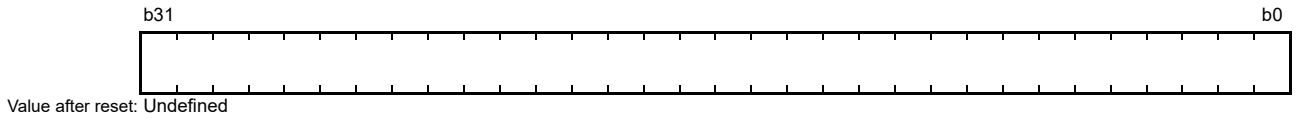
PM Bit (Processor Mode Select)

This bit specifies the processor mode. When an exception is accepted, the value of this bit becomes 0.

IPL[3:0] Bits (Processor Interrupt Priority Level)

The IPL[3:0] bits specify the processor interrupt priority level as one of sixteen levels from zero to fifteen, wherein priority level zero is the lowest and priority level fifteen the highest. When the priority level of a requested interrupt is higher than the processor interrupt priority level, the interrupt is enabled. Setting the IPL[3:0] bits to level fifteen (Fh) disables all interrupt requests. The IPL[3:0] bits are set to level fifteen (Fh) when a non-maskable interrupt is generated. When interrupts in general are generated, the bits are set to the priority levels of accepted interrupts.

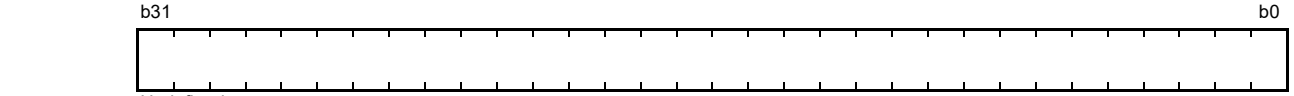
2.2.2.6 Backup PC (BPC)



The backup PC (BPC) is provided to speed up response to interrupts.
After a fast interrupt has been generated, the contents of the program counter (PC) are saved in the BPC register.

Downloaded from Arrow.com.

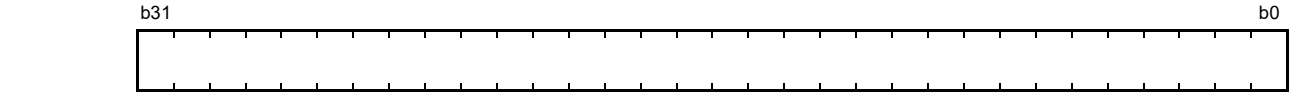
2.2.2.7 Backup PSW (BPSW)



Value after reset: Undefined

The backup PSW (BPSW) is provided to speed up response to interrupts.
After a fast interrupt has been generated, the contents of the processor status word (PSW) are saved in the BPSW. The allocation of bits in the BPSW corresponds to that in the PSW.

2.2.2.8 Fast Interrupt Vector Register (FINTV)



Value after reset: Undefined

The fast interrupt vector register (FINTV) is provided to speed up response to interrupts.
The FINTV register specifies a branch destination address when a fast interrupt has been generated.

Downloaded from Arrow.com.

2.2.2.9 Floating-Point Status Word (FPSW)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	FS	FX	FU	FZ	FO	FV	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	EX	EU	EZ	EO	EV	—	DN	CE	CX	CU	CZ	CO	CV	RM[1:0]	
Value after reset:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	RM[1:0]	Floating-Point Rounding-Mode Setting	b1 b0 0 0: Rounding towards the nearest value 0 1: Rounding towards 0 1 0: Rounding towards +∞ 1 1: Rounding towards –∞	R/W
b2	CV	Invalid Operation Cause Flag	0: No invalid operation has been encountered. 1: Invalid operation has been encountered.	R/(W) *1
b3	CO	Overflow Cause Flag	0: No overflow has occurred. 1: Overflow has occurred.	R/(W) *1
b4	CZ	Division-by-Zero Cause Flag	0: No division-by-zero has occurred. 1: Division-by-zero has occurred.	R/(W) *1
b5	CU	Underflow Cause Flag	0: No underflow has occurred. 1: Underflow has occurred.	R/(W) *1
b6	CX	Inexact Cause Flag	0: No inexact exception has been generated. 1: Inexact exception has been generated.	R/(W) *1
b7	CE	Unimplemented Processing Cause Flag	0: No unimplemented processing has been encountered. 1: Unimplemented process has been encountered.	R/(W) *1
b8	DN	0 Flush Bit of Denormalized Number	0: A denormalized number is handled as a denormalized number. 1: A denormalized number is handled as 0.*2	R/W
b9	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b10	EV	Invalid Operation Exception Enable	0: Invalid operation exception is masked. 1: Invalid operation exception is enabled.	R/W
b11	EO	Overflow Exception Enable	0: Overflow exception is masked. 1: Overflow exception is enabled.	R/W
b12	EZ	Division-by-Zero Exception Enable	0: Division-by-zero exception is masked. 1: Division-by-zero exception is enabled.	R/W
b13	EU	Underflow Exception Enable	0: Underflow exception is masked. 1: Underflow exception is enabled.	R/W
b14	EX	Inexact Exception Enable	0: Inexact exception is masked. 1: Inexact exception is enabled.	R/W
b25 to b15	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b26	FV*3	Invalid Operation Flag	0: No invalid operation has been encountered. 1: Invalid operation has been encountered.*8	R/W
b27	FO*4	Overflow Flag	0: No overflow has occurred. 1: Overflow has occurred.*8	R/W
b28	FZ*5	Division-by-Zero Flag	0: No division-by-zero has occurred. 1: Division-by-zero has occurred.*8	R/W
b29	FU*6	Underflow Flag	0: No underflow has occurred. 1: Underflow has occurred.*8	R/W
b30	FX*7	Inexact Flag	0: No inexact exception has been generated. 1: Inexact exception has been generated.*8	R/W

Downloaded from Arrow.com.

Bit	Symbol	Bit Name	Description	R/W
b31	FS	Floating-Point Error Summary Flag	This bit reflects the logical OR of the FU, FZ, FO, and FV flags.	R

Note 1. Writing 0 to the bit clears it. Writing 1 to the bit does not affect its value.
Note 2. Positive denormalized numbers are treated as +0, negative denormalized numbers as –0.
Note 3. When the EV bit is set to 0, the FV flag is enabled.
Note 4. When the EO bit is set to 0, the FO flag is enabled.
Note 5. When the EZ bit is set to 0, the FZ flag is enabled.
Note 6. When the EU bit is set to 0, the FU flag is enabled.
Note 7. When the EX bit is set to 0, the FX flag is enabled.
Note 8. Once the bit has been set to 1, this value is retained until it is cleared to 0 by software.

The floating-point status word (FPSW) indicates the results of floating-point operations.
When an exception handling enable bit (Ej) enables the exception handling (Ej = 1), the exception cause can be identified by checking the corresponding Cj flag in the exception handling routine. If the exception handling is masked (Ej = 0), the occurrence of exception can be checked by reading the Fj flag at the end of a series of processing. Once the Fj flag has been set to 1, this value is retained until it is cleared to 0 by software (j = X, U, Z, O, or V).

RM[1:0] Bits (Floating-Point Rounding-Mode Setting)

These bits specify the floating-point rounding-mode.

Explanation of Floating-Point Rounding Modes

- Rounding towards the nearest value (the default behavior) : An inexact result is rounded to the available value that is closest to the result which would be obtained with an infinite number of digits. If two available values are equally close, rounding is to the even alternative.
- Rounding towards 0 : An inexact result is rounded to the smallest available absolute value, i.e. in the direction of zero (simple truncation).
- Rounding towards +∞ : An inexact result is rounded to the nearest available value in the direction of positive infinity.
- Rounding towards –∞ : An inexact result is rounded to the nearest available value in the direction of negative infinity.

(1) Rounding to the nearest value is specified as the default mode and returns the most accurate value.
(2) Modes such as rounding towards 0, rounding towards +∞, and rounding towards –∞ are used to ensure precision when interval arithmetic is employed.

CV Flag (Invalid Operation Cause Flag), CO Flag (Overflow Cause Flag), CZ Flag (Division-by-Zero Cause Flag), CU Flag (Underflow Cause Flag), CX Flag (Inexact Cause Flag), and CE Flag (Unimplemented Processing Cause Flag)

Floating-point exceptions include the five specified in the IEEE754 standard, namely overflow, underflow, inexact, division-by-zero, and invalid operation. For a further floating-point exception that is generated upon detection of unimplemented processing, the corresponding flag (CE) is set to 1.

- The bit that has been set to 1 is cleared to 0 when the FPU instruction is executed.
- When 0 is written to the bit by the MVTC and POPC instructions, the bit is set to 0; the bit retains the previous value when 1 is written by the instruction.

DN Flag (0 Flush Bit of Denormalized Number)

When this bit is set to 0, a denormalized number is handled as a denormalized number. When this bit is set to 1, a denormalized number is handled as 0.

EV Bit (Invalid Operation Exception Enable), EO Bit (Overflow Exception Enable), EZ Bit (Division-by-Zero Exception Enable), EU Bit (Underflow Exception Enable), and EX Bit (Inexact Exception Enable)

When any of five floating-point exceptions specified in the IEEE754 standard is generated by the floating-point

Downloaded from Arrow.com.

operation instruction, the bit decides whether the CPU will start handling the exception. When the bit is set to 0, the exception handling is masked; when the bit is set to 1, the exception handling is enabled.

FV Flag (Invalid Operation Flag), FO Flag (Overflow Flag), FZ Flag (Division-by-Zero Flag), FU Flag (Underflow Flag), and FX Flag (Inexact Flag)

While the exception handling enable bit (Ej) is 0 (exception handling is masked), if any of five floating-point exceptions specified in the IEEE754 standard is generated, the corresponding bit is set to 1.

- When Ej is 1 (exception handling is enabled), the value of the flag remains.
- When the corresponding flag is set to 1, it remains 1 until it is cleared to 0 by software. (accumulation flag)

FS Flag (Floating-Point Error Summary Flag)

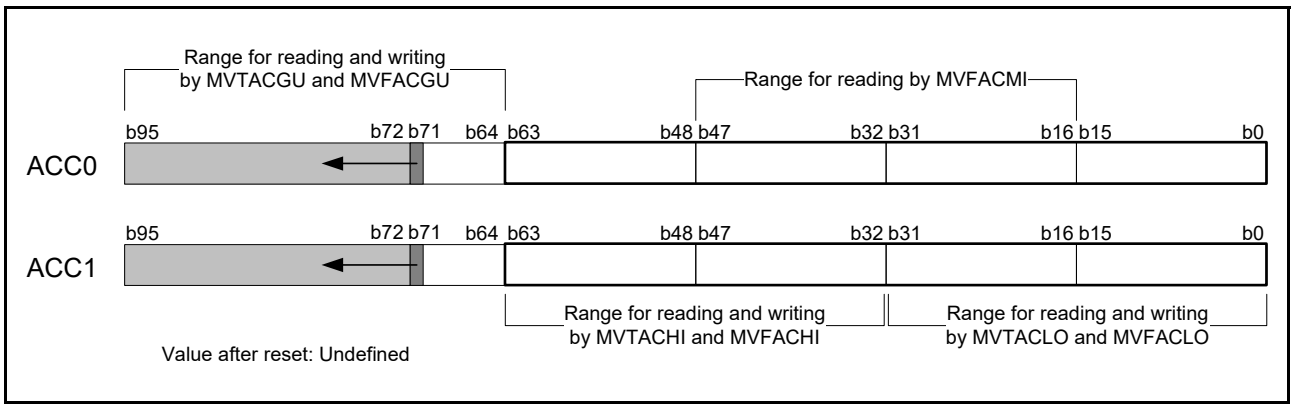
This bit reflects the logical OR of the FU, FZ, FO, and FV flags.

2.2.3 Accumulator

The accumulator (ACC0 or ACC1) is a 72-bit register used for DSP instructions. The accumulator is handled as a 96-bit register for reading and writing. At this time, when bits 95 to 72 of the accumulator are read, the value where the value of bit 71 is sign extended is read. Writing to bits 95 to 72 of the accumulator is ignored. ACC0 is also used for the multiply and multiply-and-accumulate instructions; EMUL, EMULU, FMUL, MUL, and RMPA, in which case the prior value in ACC0 is modified by execution of the instruction.

Use the MVTACGU, MVTACHI, and MVTACLO instructions for writing to the accumulator. The MVTACGU, MVTACHI, and MVTACLO instructions write data to bits 95 to 64, the higher-order 32 bits (bits 63 to 32), and the lower-order 32 bits (bits 31 to 0), respectively.

Use the MVFACGU, MVFACHI, MVFACMI, and MVFACLO instructions for reading data from the accumulator. The MVFACGU, MVFACHI, MVFACMI, and MVFACLO instructions read data from the guard bits (bits 95 to 64), higher-order 32 bits (bits 63 to 32), the middle 32 bits (bits 47 to 16), and the lower-order 32 bits (bits 31 to 0), respectively.



Note: The value of bit 71 is sign extended for bits 95 to 72 and the extended value is always read. Writing to this area is ignored.

Downloaded from Arrow.com.

2.3 Processor Mode

The RXv2 CPU supports two processor modes, supervisor and user. These processor modes and the memory protection function enable the realization of a hierarchical CPU resource protection and memory protection mechanism. Each processor mode imposes a level on rights of access to memory and the instructions that can be executed. Supervisor mode carries greater rights than user mode. The initial state after a reset is supervisor mode.

2.3.1 Supervisor Mode

In supervisor mode, all CPU resources are accessible and all instructions are available. However, writing to the processor mode select bit (PM) in the processor status word (PSW) by executing an MVTC or a POPC instruction will be ignored. For details on how to write to the PM bit, refer to section 2.2.2.5, Processor Status Word (PSW).

2.3.2 User Mode

In user mode, write access to the CPU resources listed below is restricted. The restriction applies to any instruction capable of write access.

- Some bits (bits IPL[3:0], PM, U, and I) in the processor status word (PSW)
- Interrupt stack pointer (ISP)
- Exception table register (EXTB)
- Interrupt table register (INTB)
- Backup PSW (BPSW)
- Backup PC (BPC)
- Fast interrupt vector register (FINTV)

2.3.3 Privileged Instruction

Privileged instructions can only be executed in supervisor mode. Executing a privileged instruction in user mode produces a privileged instruction exception. Privileged instructions include the RTFI, MVTIPL, RTE, and WAIT instructions.

2.3.4 Switching Between Processor Modes

Manipulating the processor mode select bit (PM) in the processor status word (PSW) switches the processor mode. However, rewriting to the PM bit by executing an MVTC or a POPC instruction is prohibited. Switch the processor mode by following the procedures described below.

(1) Switching from user mode to supervisor mode

After an exception has been generated, the PSW.PM bit is set to 0 and the CPU switches to supervisor mode. The hardware pre-processing is executed in supervisor mode. The state of the processor mode before the exception was generated is retained in the copy of PSW.PM bit is saved on the stack.

(2) Switching from supervisor mode to user mode

Executing an RTE instruction when the value of the copy of the PSW.PM bit that has been preserved on the stack is 1 or an RTFI instruction when the value of the copy of the PSW.PM bit that has been preserved in the backup PSW (BPSW) is 1 causes a transition to user mode. In the transition to user mode, the value of the stack pointer designation bit (the U bit in the PSW) becomes 1.

Downloaded from Arrow.com.

2.4 Data Types

The RXv2 CPU can handle four types of data: integer, floating-point, bit, and string.
For details, refer to RX Family RXv2 Instruction Set Architecture User's Manual: Software.

2.4.1 Integer

An integer can be signed or unsigned. For signed integers, negative values are represented by two's complements.

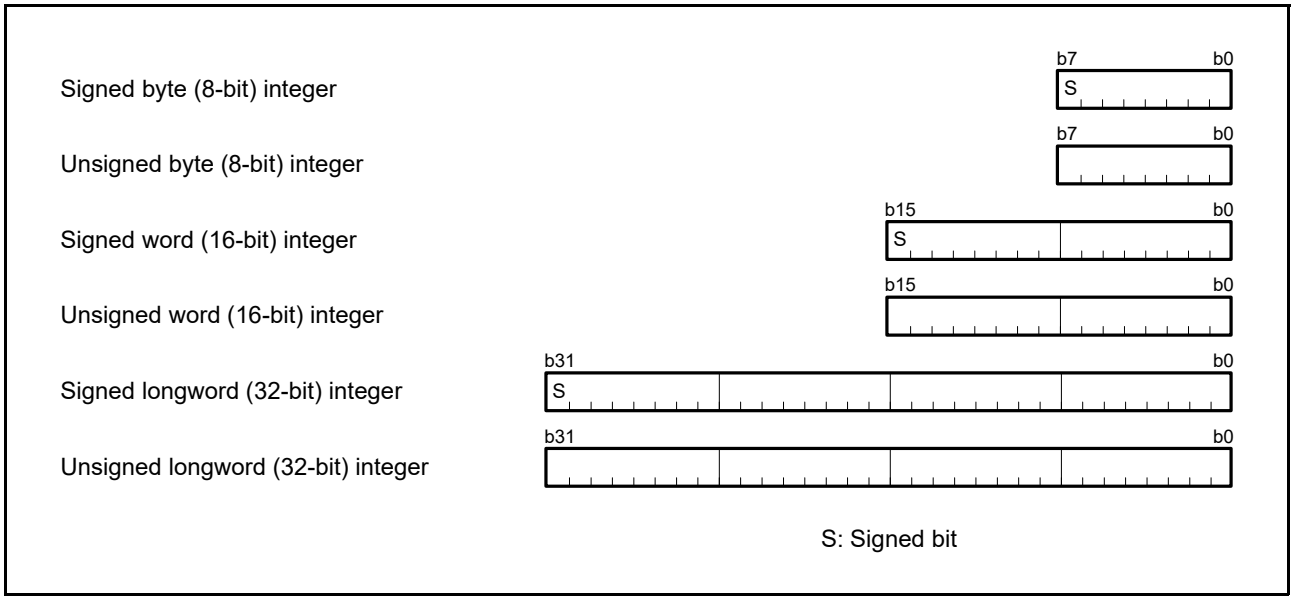


Figure 2.2 Integer

Downloaded from Arrow.com.

2.4.2 Floating-Points

Floating-point support is for the single-precision floating-point type specified in the IEEE754 standard; operands of this type can be used in eleven floating-point operation instructions: FADD, FCMP, FDIV, FMUL, FSQRT, FSUB, FTOI, FTOU, ITOF, ROUND, and UTOF.

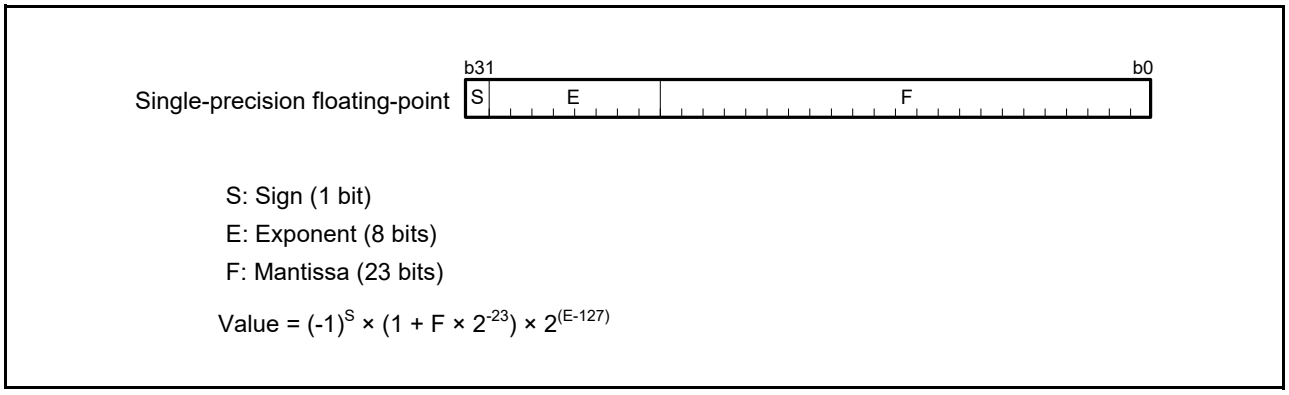


Figure 2.3 Floating-Point

The floating-point format supports the values listed below.

0 < E < 255 (normal numbers)
E = 0 and F = 0 (signed zero)
E = 0 and F > 0 (denormalized numbers)*1
E = 255 and F = 0 (infinity)
E = 255 and F > 0 (NaN: Not-a-Number)

Note 1. The number is treated as 0 when the FPSW.DN bit is 1. When the DN bit is 0, an unimplemented processing exception is generated.

2.4.3 Bitwise Operations

Five bit-manipulation instructions are provided for bitwise operations: BCLR, BMC*nd*, BNOT, BSET, and BTST.
A bit in a register is specified as the destination register and a bit number in the range from 31 to 0.
A bit in memory is specified as the destination address and a bit number from 7 to 0. The addressing modes available to specify addresses are register indirect and register relative.

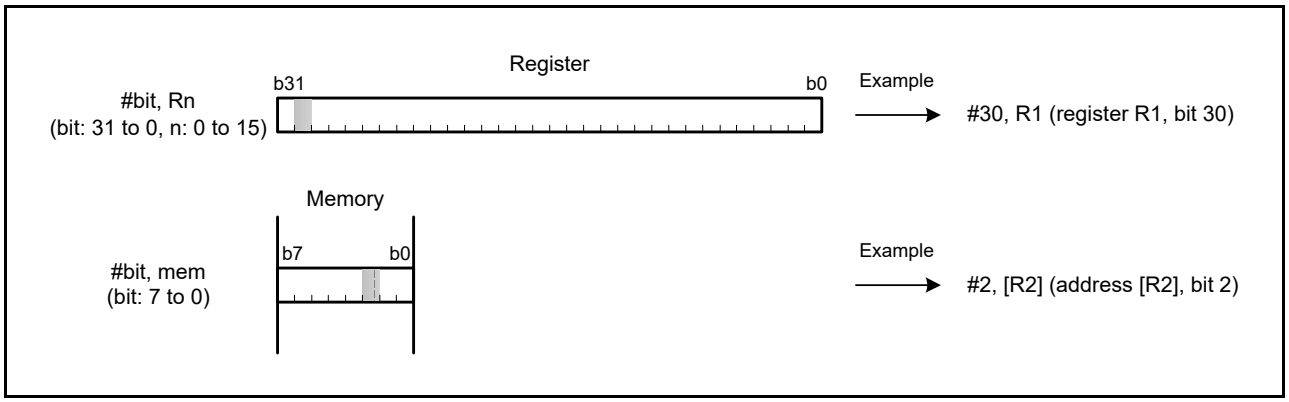


Figure 2.4 Bit

Downloaded from Arrow.com.

2.4.4 Strings

The string data type consists of an arbitrary number of consecutive byte (8-bit), word (16-bit), or longword (32-bit) units. Seven string manipulation instructions are provided for use with strings: SCMPU, SMOVB, SMOVF, SMOVU, SSTR, SUNTIL, and SWHILE.

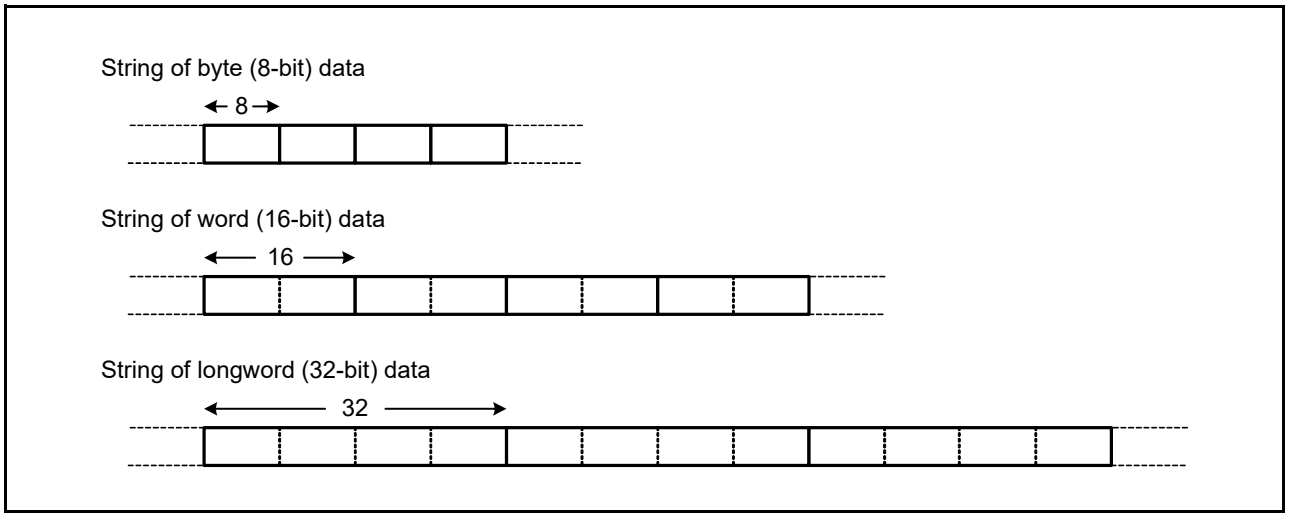


Figure 2.5 String

Downloaded from Arrow.com.

2.5 Endian

For the RXv2 CPU, instructions are little endian, but the treatment of data is selectable as little or big endian.

2.5.1 Switching the Endian

As arrangements of bytes, this MCU supports both big endian, where the higher-order byte (MSB) is at location 0, and little endian, where the lower-order byte (LSB) is at location 0.
For details on the endian setting, see section 3, Operating Modes.
Operations for access differ according to the endian setting and, depending on the instruction, whether 8-, 16- or 32-bit access has been selected. Operations for access in the various possible cases are described in Table 2.1 to Table 2.12.
In the tables,

LL indicates bits D7 to D0 of the general-purpose register,
LH indicates bits D15 to D8 of the general-purpose register,
HL indicates bits D23 to D16 of the general-purpose register, and
HH indicates bits D31 to D24 of the general-purpose register.

	D31 to D24	D23 to D16	D15 to D8	D7 to D0
General purpose register: Rm	HH	HL	LH	LL

Table 2.1 32-Bit Read Operations when Little Endian has been Selected

Operation / Address of src	Reading a 32-bit unit from address 0	Reading a 32-bit unit from address 1	Reading a 32-bit unit from address 2	Reading a 32-bit unit from address 3	Reading a 32-bit unit from address 4
Address 0	Transfer to LL	—	—	—	—
Address 1	Transfer to LH	Transfer to LL	—	—	—
Address 2	Transfer to HL	Transfer to LH	Transfer to LL	—	—
Address 3	Transfer to HH	Transfer to HL	Transfer to LH	Transfer to LL	—
Address 4	—	Transfer to HH	Transfer to HL	Transfer to LH	Transfer to LL
Address 5	—	—	Transfer to HH	Transfer to HL	Transfer to LH
Address 6	—	—	—	Transfer to HH	Transfer to HL
Address 7	—	—	—	—	Transfer to HH

Table 2.2 32-Bit Read Operations when Big Endian has been Selected

Operation / Address of src	Reading a 32-bit unit from address 0	Reading a 32-bit unit from address 1	Reading a 32-bit unit from address 2	Reading a 32-bit unit from address 3	Reading a 32-bit unit from address 4
Address 0	Transfer to HH	—	—	—	—
Address 1	Transfer to HL	Transfer to HH	—	—	—
Address 2	Transfer to LH	Transfer to HL	Transfer to HH	—	—
Address 3	Transfer to LL	Transfer to LH	Transfer to HL	Transfer to HH	—
Address 4	—	Transfer to LL	Transfer to LH	Transfer to HL	Transfer to HH
Address 5	—	—	Transfer to LL	Transfer to LH	Transfer to HL
Address 6	—	—	—	Transfer to LL	Transfer to LH
Address 7	—	—	—	—	Transfer to LL

Downloaded from Arrow.com.

Table 2.3 32-Bit Write Operations when Little Endian has been Selected

Operation / Address of dest	Writing a 32-bit unit to address 0	Writing a 32-bit unit to address 1	Writing a 32-bit unit to address 2	Writing a 32-bit unit to address 3	Writing a 32-bit unit to address 4
Address 0	Transfer from LL	—	—	—	—
Address 1	Transfer from LH	Transfer from LL	—	—	—
Address 2	Transfer from HL	Transfer from LH	Transfer from LL	—	—
Address 3	Transfer from HH	Transfer from HL	Transfer from LH	Transfer from LL	—
Address 4	—	Transfer from HH	Transfer from HL	Transfer from LH	Transfer from LL
Address 5	—	—	Transfer from HH	Transfer from HL	Transfer from LH
Address 6	—	—	—	Transfer from HH	Transfer from HL
Address 7	—	—	—	—	Transfer from HH

Table 2.4 32-Bit Write Operations when Big Endian has been Selected

Operation / Address of dest	Writing a 32-bit unit to address 0	Writing a 32-bit unit to address 1	Writing a 32-bit unit to address 2	Writing a 32-bit unit to address 3	Writing a 32-bit unit to address 4
Address 0	Transfer from HH	—	—	—	—
Address 1	Transfer from HL	Transfer from HH	—	—	—
Address 2	Transfer from LH	Transfer from HL	Transfer from HH	—	—
Address 3	Transfer from LL	Transfer from LH	Transfer from HL	Transfer from HH	—
Address 4	—	Transfer from LL	Transfer from LH	Transfer from HL	Transfer from HH
Address 5	—	—	Transfer from LL	Transfer from LH	Transfer from HL
Address 6	—	—	—	Transfer from LL	Transfer from LH
Address 7	—	—	—	—	Transfer from LL

Table 2.5 16-Bit Read Operations when Little Endian has been Selected

Operation / Address of src	Reading a 16-bit unit from address 0	Reading a 16-bit unit from address 1	Reading a 16-bit unit from address 2	Reading a 16-bit unit from address 3	Reading a 16-bit unit from address 4	Reading a 16-bit unit from address 5	Reading a 16-bit unit from address 6
Address 0	Transfer to LL	—	—	—	—	—	—
Address 1	Transfer to LH	Transfer to LL	—	—	—	—	—
Address 2	—	Transfer to LH	Transfer to LL	—	—	—	—
Address 3	—	—	Transfer to LH	Transfer to LL	—	—	—
Address 4	—	—	—	Transfer to LH	Transfer to LL	—	—
Address 5	—	—	—	—	Transfer to LH	Transfer to LL	—
Address 6	—	—	—	—	—	Transfer to LH	Transfer to LL
Address 7	—	—	—	—	—	—	Transfer to LH

Downloaded from Arrow.com.

Table 2.6 16-Bit Read Operations when Big Endian has been Selected

Operation / Address of src	Reading a 16-bit unit from address 0	Reading a 16-bit unit from address 1	Reading a 16-bit unit from address 2	Reading a 16-bit unit from address 3	Reading a 16-bit unit from address 4	Reading a 16-bit unit from address 5	Reading a 16-bit unit from address 6
Address 0	Transfer to LH	—	—	—	—	—	—
Address 1	Transfer to LL	Transfer to LH	—	—	—	—	—
Address 2	—	Transfer to LL	Transfer to LH	—	—	—	—
Address 3	—	—	Transfer to LL	Transfer to LH	—	—	—
Address 4	—	—	—	Transfer to LL	Transfer to LH	—	—
Address 5	—	—	—	—	Transfer to LL	Transfer to LH	—
Address 6	—	—	—	—	—	Transfer to LL	Transfer to LH
Address 7	—	—	—	—	—	—	Transfer to LL

Table 2.7 16-Bit Write Operations when Little Endian has been Selected

Operation / Address of dest	Writing a 16-bit unit to address 0	Writing a 16-bit unit to address 1	Writing a 16-bit unit to address 2	Writing a 16-bit unit to address 3	Writing a 16-bit unit to address 4	Writing a 16-bit unit to address 5	Writing a 16-bit unit to address 6
Address 0	Transfer from LL	—	—	—	—	—	—
Address 1	Transfer from LH	Transfer from LL	—	—	—	—	—
Address 2	—	Transfer from LH	Transfer from LL	—	—	—	—
Address 3	—	—	Transfer from LH	Transfer from LL	—	—	—
Address 4	—	—	—	Transfer from LH	Transfer from LL	—	—
Address 5	—	—	—	—	Transfer from LH	Transfer from LL	—
Address 6	—	—	—	—	—	Transfer from LH	Transfer from LL
Address 7	—	—	—	—	—	—	Transfer from LH

Table 2.8 16-Bit Write Operations when Big Endian has been Selected

Operation / Address of dest	Writing a 16-bit unit to address 0	Writing a 16-bit unit to address 1	Writing a 16-bit unit to address 2	Writing a 16-bit unit to address 3	Writing a 16-bit unit to address 4	Writing a 16-bit unit to address 5	Writing a 16-bit unit to address 6
Address 0	Transfer from LH	—	—	—	—	—	—
Address 1	Transfer from LL	Transfer from LH	—	—	—	—	—
Address 2	—	Transfer from LL	Transfer from LH	—	—	—	—
Address 3	—	—	Transfer from LL	Transfer from LH	—	—	—
Address 4	—	—	—	Transfer from LL	Transfer from LH	—	—
Address 5	—	—	—	—	Transfer from LL	Transfer from LH	—
Address 6	—	—	—	—	—	Transfer from LL	Transfer from LH
Address 7	—	—	—	—	—	—	Transfer from LL

Table 2.9 8-Bit Read Operations when Little Endian has been Selected

Operation / Address of src	Reading an 8-bit unit from address 0	Reading an 8-bit unit from address 1	Reading an 8-bit unit from address 2	Reading an 8-bit unit from address 3
Address 0	Transfer to LL	—	—	—
Address 1	—	Transfer to LL	—	—
Address 2	—	—	Transfer to LL	—
Address 3	—	—	—	Transfer to LL

Downloaded from Arrow.com.

Table 2.10 8-Bit Read Operations when Big Endian has been Selected

Operation / Address of src	Reading an 8-bit unit from address 0	Reading an 8-bit unit from address 1	Reading an 8-bit unit from address 2	Reading an 8-bit unit from address 3
Address 0	Transfer to LL	—	—	—
Address 1	—	Transfer to LL	—	—
Address 2	—	—	Transfer to LL	—
Address 3	—	—	—	Transfer to LL

Table 2.11 8-Bit Write Operations when Little Endian has been Selected

Operation / Address of dest	Writing an 8-bit unit to address 0	Writing an 8-bit unit to address 1	Writing an 8-bit unit to address 2	Writing an 8-bit unit to address 3
Address 0	Transfer from LL	—	—	—
Address 1	—	Transfer from LL	—	—
Address 2	—	—	Transfer from LL	—
Address 3	—	—	—	Transfer from LL

Table 2.12 8-Bit Write Operations when Big Endian has been Selected

Operation / Address of dest	Writing an 8-bit unit to address 0	Writing an 8-bit unit to address 1	Writing an 8-bit unit to address 2	Writing an 8-bit unit to address 3
Address 0	Transfer from LL	—	—	—
Address 1	—	Transfer from LL	—	—
Address 2	—	—	Transfer from LL	—
Address 3	—	—	—	Transfer from LL

2.5.2 Access to I/O Registers

The addresses of I/O registers are fixed, and this is regardless of whether the setting is for little endian or big endian. Accordingly, changes to the endian do not affect access to I/O registers. For the arrangements of I/O registers, refer to the descriptions of registers in the relevant sections.

2.5.3 Notes on Access to I/O Registers

Ensure that access to I/O registers is in accord with the following rules.

- With I/O registers for which a bus width of eight bits is indicated, use instructions having operands of the same width (eight bits). That is, access these registers by using instructions with .B as the size specifier (.size), or with .B or .UB as the size-extension specifier (.memex).
- With I/O registers for which a bus width of 16 bits is indicated, use instructions having operands of the same width (16 bits). That is, access these registers by using instructions with .W as the size specifier (.size), or with .W or .UW as the size-extension specifier (.memex).
- With I/O registers for which a bus width of 32 bits is indicated, use instructions having operands of the same width (32 bits). That is, access these registers by using instructions with .L as the size specifier (.size), or with .L size-extension specifier (.memex).

Downloaded from Arrow.com.

2.5.4 Data Arrangement

2.5.4.1 Data Arrangement in Registers

Figure 2.6 shows the relation between the sizes of registers and bit numbers.

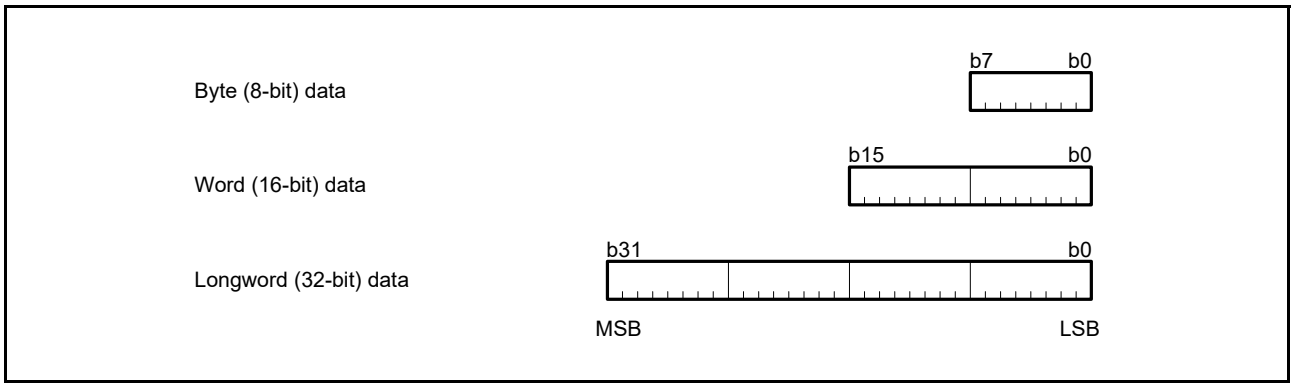


Figure 2.6 Data Arrangement in Registers

2.5.4.2 Data Arrangement in Memory

Data in memory have three sizes: byte (8-bit), word (16-bit), and longword (32-bit). The data arrangement is selectable as little endian or big endian. Figure 2.7 shows the arrangement of data in memory.

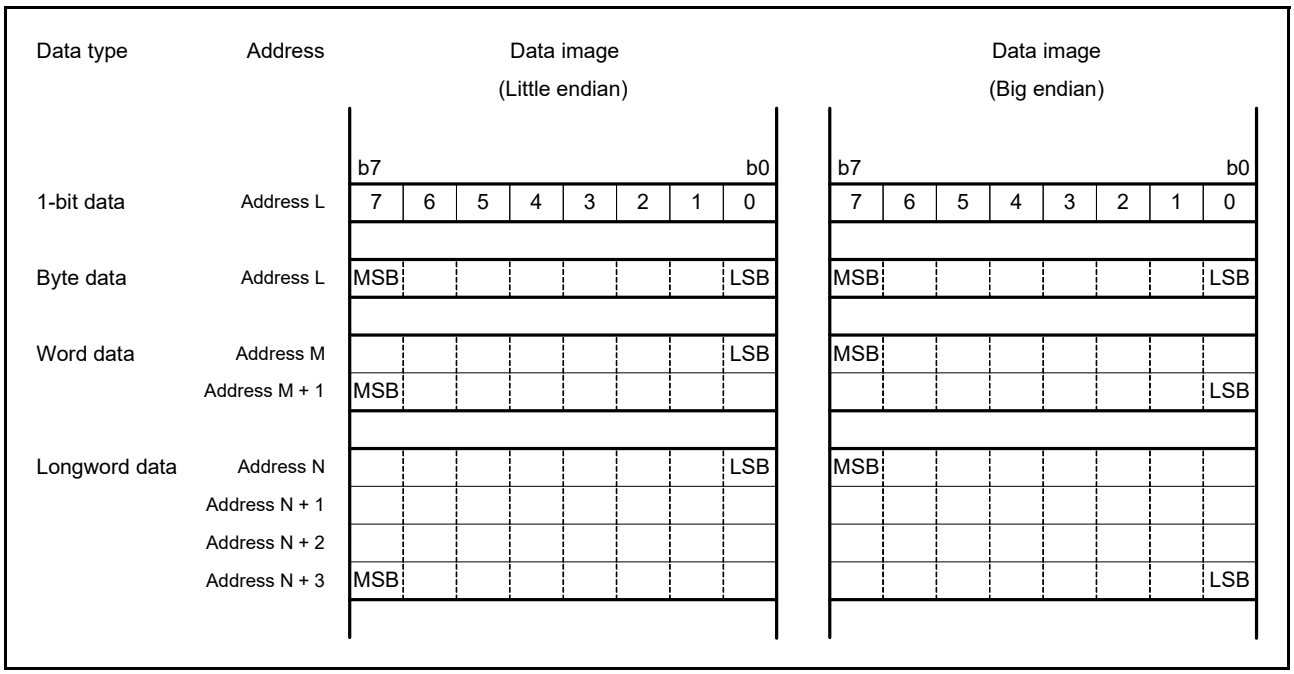


Figure 2.7 Data Arrangement in Memory

2.5.5 Notes on the Allocation of Instruction Codes

The allocation of instruction codes to an external space where the endian differs from that of the chip is prohibited. If the instruction codes are allocated to the external space, they must be allocated to areas where the endian setting is the same as that for the chip.

Downloaded from Arrow.com.

2.6 Vector Table

There are two types of vector table: exception and interrupt. Each vector in the vector table consists of four bytes and specifies the address where the corresponding exception handling routine starts.

2.6.1 Exception Vector Table

In the exception vector table, the individual vectors for the privileged instruction exception, access exception, undefined instruction exception, floating-point exception, and non-maskable interrupt are allocated to the 124-byte area where the value indicated by the exception table register (EXTB) is used as the starting address (ExtBase). The reset vector is always allocated to FFFFFFFCh, regardless of the value of the exception vector table.
Figure 2.8 shows the exception vector table.

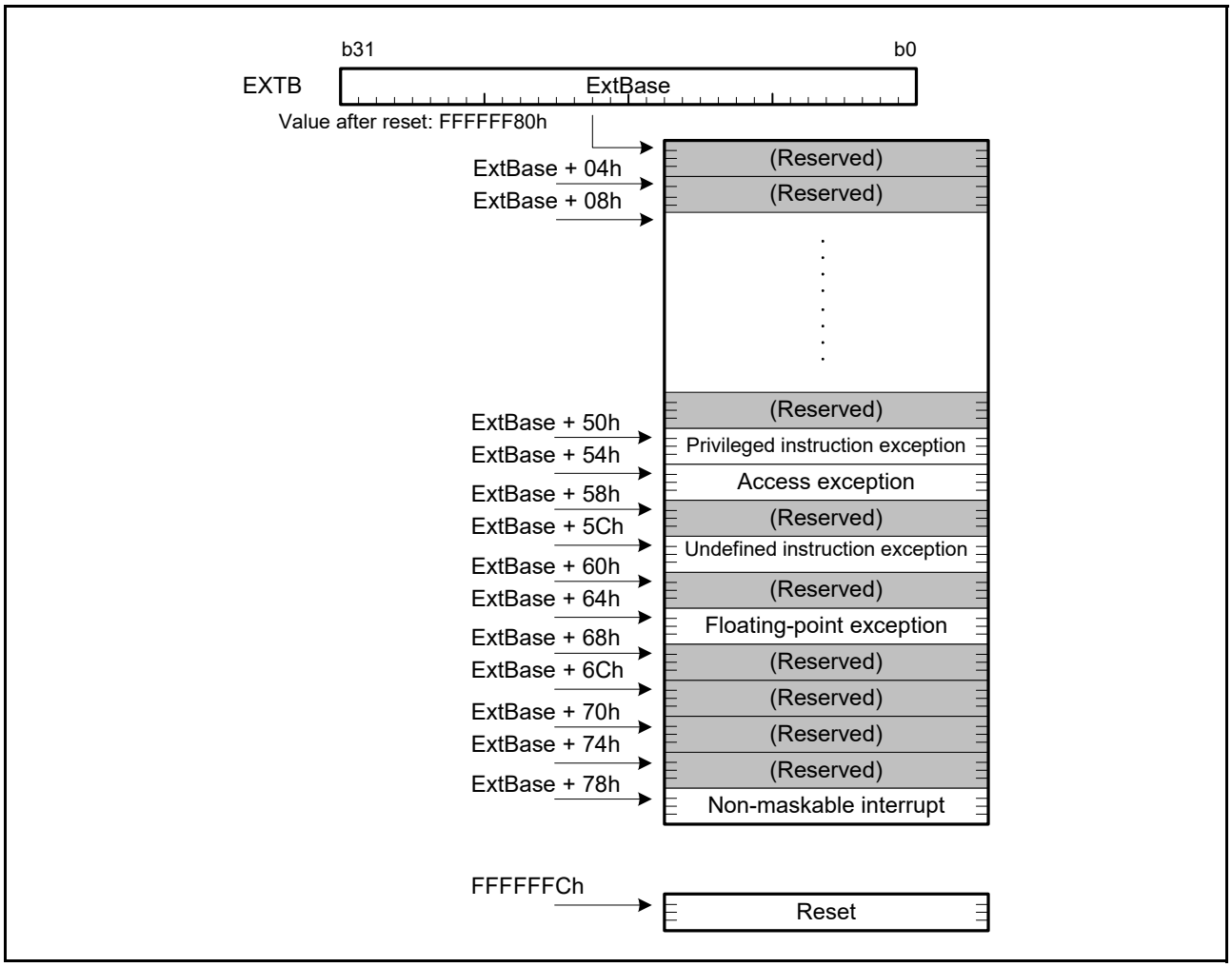


Figure 2.8 Exception Vector Table

Downloaded from Arrow.com.

2.6.2 Interrupt Vector Table

The address where the interrupt vector table is placed can be adjusted. The table is a 1,024-byte region that contains all vectors for unconditional traps and interrupts and starts at the address (IntBase) specified in the interrupt table register (INTB). Figure 2.9 shows the interrupt vector table.

Each vector in the interrupt vector table has a vector number from 0 to 255. Each of the INT instructions, which act as the sources of unconditional traps, is allocated to the vector that has the same number as is specified as the operand of the instruction itself (from 0 to 255). The BRK instruction is allocated to the vector with number 0. Furthermore, vector numbers (from 0 to 255) are allocated to interrupt requests in a fixed way for each product. For more on interrupt vector numbers, see section 15.3.1, Interrupt Vector Table.

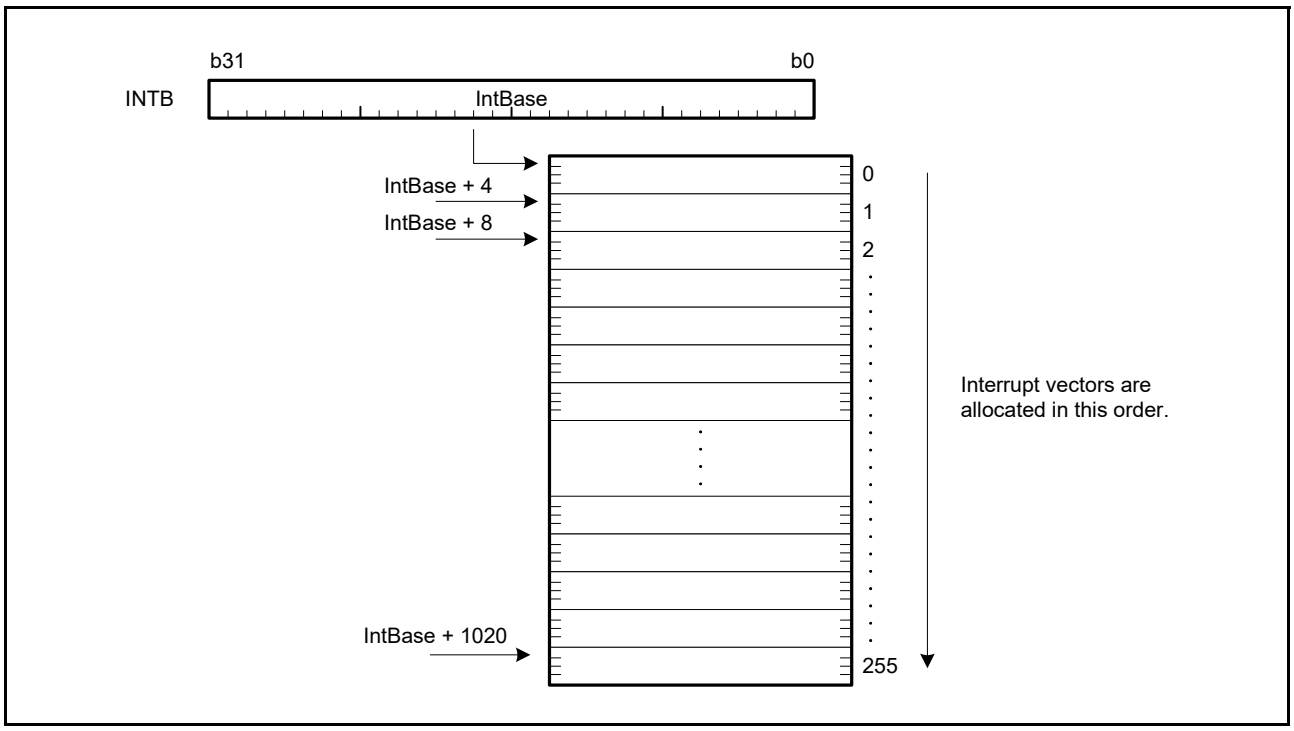


Figure 2.9 Interrupt Vector Table

Downloaded from Arrow.com.

2.7 Operation of Instructions

2.7.1 Restrictions on RMPA and String-Manipulation Instructions

2.7.1.1 Transfer Size and Data Prefetching

The RMPA instruction and the string-manipulation instructions (SCMPU, SMOVB, SMOVF, SMOVU, SSTR, SUNTIL, and SWHILE instructions) transfer data in longword units to speed up the reading of data from and writing of data to the memory. If the last of the data to be processed is less than a longword, data transfer proceeds with the sizes described below:

- RMPA, SSTR, SUNTIL, and SWHILE instructions: Size specified by the size specifier
- SCMPU, SMOVB, SMOVF, and SMOVU instructions: Byte

Additionally, in the above processing, the RMPA instruction and the string-manipulation instructions other than the SSTR instruction (that is, the SCMPU, SMOVB, SMOVF, SMOVU, SUNTIL, and SWHILE instructions) prefetch data when reading data from the memory. Data is prefetched from the prefetching start position with three bytes as the upper limit. The prefetching start positions of each operation are shown below.

- RMPA instruction: The multiplicand address specified by R1, and the multiplier address specified by R2
- SCMPU instruction: The source address specified by R1 for comparison, and the destination address specified by R2 for comparison
- SUNTIL and SWHILE instructions: The destination address specified by R1 for comparison
- SMOVB, SMOVF, and SMOVU instructions: The source address specified by R2 for transfer

2.7.1.2 Access to the External Space

Although the external space has a per-area ending-switching facility (only for data), the allocation of data to be handled by RMPA or string-manipulation instructions (SCMPU, SMOVB, SMOVF, SMOVU,SSTR, SUNTIL, and SWHILE instructions) to an area where the endian differs from that of the chip is prohibited, and operation is not guaranteed if this restriction is not observed. If data to be handled by RMPA or string-manipulation instructions are allocated to the external space, they must be allocated to areas where the endian setting is the same as that for the chip.

2.7.1.3 Access to I/O Registers

The allocation of data to be handled by RMPA or string-manipulation instructions (SCMPU, SMOVB, SMOVF, SMOVU, SSTR, SUNTIL, and SWHILE instructions) to I/O registers is prohibited, and operation is not guaranteed if this restriction is not observed.

Downloaded from Arrow.com.

2.8 Number of Cycles

2.8.1 Instruction and Number of Cycle

Table 2.13 to Table 2.20 show the number of cycles in operation of each instruction. The listed numbers of cycles for access to memory are the numbers of cycles during no-wait access. The operands in the table below indicate the following meanings.

#IMM: Immediate
flag: bit, flag
Rs, Rs2, Rd, Rd2, Ri, Rb: General-purpose register
As, Ad: Accumulator
CR: Control register
dsp: displacement
pcdsp: displacement

Table 2.13 Number of Cycles for Arithmetic/logic Instructions

Instruction	Mnemonic (indicates the common operation when the size is omitted)	Number of Cycles
Arithmetic/logic instructions (register-register, immediate-register)	• {ABS, NEG, NOT} “Rd”/“Rs, Rd” • {ADC, MAX, MIN, ROTL, ROTR, XOR} “#IMM, Rd”/“Rs, Rd” • ADD “#IMM, Rd”/“Rs, Rd”/“#IMM, Rs, Rd”/“Rs, Rs2, Rd” • {AND, MUL, OR, SUB} “#IMM, Rd”/“Rs, Rd”/“Rs, Rs2, Rd” • {CMP, TST} “#IMM, Rs”/“Rs, Rs2” • NOP • {ROLC, RORC, SAT} “Rd” • SBB “Rs, Rd” • {SHAR, SHLL, SHLR} “#IMM, Rd”/“Rs, Rd”/“#IMM, Rs, Rd”	1
	• DIV “#IMM, Rd”/“Rs, Rd”	3 to 20*1
	• DIVU “#IMM, Rd”/“Rs, Rd”	2 to 18*1
	• {EMUL, EMULU} “#IMM, Rd”/“Rs, Rd”	2
	• SATR	3
Arithmetic/logic instructions (memory source operand)	• {ADC, ADD, AND, MAX, MIN, MUL, OR, SBB, SUB, XOR} “[Rs], Rd”/“dsp[Rs], Rd” • {CMP, TST} “[Rs], Rs2”/“dsp[Rs], Rs2”	3
	• DIV “[Rs], Rd / dsp[Rs], Rd”	5 to 22
	• DIVU“[Rs], Rd / dsp[Rs], Rd”	4 to 20
	• {EMUL, EMULU} “[Rs], Rd”/“dsp[Rs], Rd”	4
	• RMPA.B	6+7×floor(n/4)+4×(n%4) n: Number of processing bytes*2
	• RMPA.W	6+5×floor(n/2)+4×(n%2) n: Number of processing words*2
	• RMPA.L	6+4n n: Number of processing longwords*2

Note 1. The number of cycles for the dividing instruction varies according to the divisor and dividend.
Note 2. floor (x): Max. integer that is smaller than x.

Downloaded from Arrow.com.

Table 2.14 Number of Cycles for Transfer Instructions

Instruction	Mnemonic (indicates the common operation when the size is omitted)	Number of Cycles
Transfer instructions (register-register, immediate-register)	• MOV “#IMM, Rd”/“Rs, Rd” • {MOVU, REVL, REVW} “Rs, Rd” • SCC*nd* “Rd” • {STNZ, STZ} “#IMM, Rd”/ “Rs, Rd”	1
	• XCHG “Rs, Rd”	2
Transfer instructions (load operation)	• {MOV, MOVU} “[Rs], Rd”/“dsp[Rs], Rd”/“[Rs+], Rd”/“[-Rs], Rd”/“[Ri, Rb], Rd” • MOVLI “[Rs], Rd” • POP “Rd”	Throughput: 1 Latency: 2*1
	• POPC “CR”	Throughput: 3 Latency: 4*1
	• POPM “Rd-Rd2”	Throughput: n Latency: n+1 n: Number of registers*1, *2
Transfer instructions (store operation)	• MOV “Rs, [Rd]”/“Rs, dsp[Rd]”/“Rs, [Rd+]”/“Rs, [-Rd]”/“Rs, [Ri, Rb]”/“#IMM, dsp[Rd]”/“#IMM, [Rd]” • PUSH “Rs” • PUSHC “CR” • SCCnd “[Rd]”/“dsp[Rd]” • MOVCO “Rs, [Rd]”	1
	• PUSHM “Rs-Rs2”	n n: Number of registers*3
Transfer instructions (memory-register)	• XCHG “[Rs], Rd”/“dsp[Rs], Rd”	2
Transfer instructions (memory-memory)	• MOV “[Rs], [Rd]”/“dsp[Rs], [Rd]”/“[Rs], dsp[Rd]”/“dsp[Rs], dsp[Rd]” • PUSH “[Rs]”/“dsp[Rs]”	3

Note 1. When the load data is used by the subsequent instruction, the number of cycles described as “latency” is counted as the number of cycles for the memory load instruction. For the cycles other than the memory load instruction, the number of cycles described as “throughput” is counted.

Note 2. The POPM instruction is converted into multiple load operations. The processing is the same as the one for the load operations of the MOV instruction, where the operation is repeated for the number of specified registers.

Note 3. The PUSHM instruction is converted into multiple store operations. The processing is the same as the one for the store operations of the MOV instruction, where the operation is repeated for the number of specified registers.

Table 2.15 Number of Cycles for Bit Manipulation Instructions

Instruction	Mnemonic (indicates the common operation when the size is omitted)	Number of Cycles
Bit manipulation instructions (register)	• {BCLR, BNOT, BSET} “#IMM, Rd”/“Rs, Rd” • BM*Cnd* “#IMM, Rd” • BTST “#IMM, Rs”/“Rs, Rs2”	1
Bit manipulation instructions (memory source operand)	• {BCLR, BNOT, BSET} “#IMM, [Rd]”/“#IMM, dsp[Rd]”/“Rs, [Rd]”/“Rs, dsp[Rd]” • BM*Cnd* “#IMM, [Rd]”/“#IMM, dsp[Rd]” • BTST “#IMM, [Rs]”/“#IMM, dsp[Rs]”/“Rs, [Rs2]”/“Rs, dsp[Rs2]”	3

Downloaded from Arrow.com.

Table 2.16 Number of Cycles for Branch Instructions

Instruction	Mnemonic (indicates the common operation when the size is omitted)	Number of Cycles
Branch instructions	• B*Cnd* “pcdsp” • {BRA, BSR} “pcdsp”/“Rs” • {JMP, JSR} “Rs”	Branch taken: 3 Branch not taken: 1
	• RTE	6
	• RTFI	3
	• RTS	5
	• RTSD “#IMM”	5
	• RTSD “#IMM, Rd-Rd2”	Throughput: n<5?5:1+n Latency: n<4?5:2+n n: Number of registers*1

?: Conditional operator

Note 1. When the load data is used by the subsequent instruction, the number of cycles described as “latency” is counted as the number of cycles for the memory load instruction. For the cycles other than the memory load instruction, the number of cycles described as “throughput” is counted.

Table 2.17 Number of Cycles for Floating-Point Operation Instructions

Instruction	Mnemonic (indicates the common operation when the size is omitted)	Number of Cycles
Floating-point operation instructions (register-register, immediate-register)	• {FADD, FSUB} “#IMM, Rd”/ “Rs, Rd”/ “Rs, Rs2, Rd”	2
	• FCMP “#IMM, Rs”/“Rs, Rs2”	1
	• FDIV “#IMM, Rd”/“Rs, Rd”	16
	• FMUL “#IMM, Rd”/ “Rs, Rd” / “Rs, Rs2, Rd”	2
	• FSQRT “Rs, Rd”	16
	• {FTOI, ROUND, ITOF} “Rs, Rd”	2
	• {FTOU, UTOF} “Rs, Rd”	2
Floating-point operation instructions (memory source operand)	• {FADD, FSUB} “[Rs], Rd”/“dsp[Rs], Rd”	4
	• FCMP “[Rs], Rs2”/“dsp[Rs], Rs2”	3
	• FDIV “[Rs], Rd”/“dsp[Rs], Rd”	18
	• FMUL “[Rs], Rd”/“dsp[Rs], Rd”	4
	• FSQRT “[Rs], Rd”/“dsp[Rs], Rd”	18
	• {FTOI, ROUND, ITOF} “[Rs], Rd”/“dsp[Rs], Rd”	4
	• {FTOU, UTOF} “[Rs], Rd”/“dsp[Rs], Rd”	4

Table 2.18 Number of Cycles for DSP Instructions

Instruction	Mnemonic (indicates the common operation when the size is omitted)	Number of Cycles
DSP instructions	• {EMULA, EMACA, EMSBA, MULLH, MULHI, MULLO, MACLH, MACHI, MACLO, MSBLH, MSBHI, MSBLO}“Rs, Rs2, Ad” • {MVFACHI, MVFACMI, MVFACLO, MVFACGU} “#IMM, As, Rd” • {MVTACHI, MVTACLO, MVTACGU} “As, Rd” • {RDACW, RDACL, RACW, RACL} “#IMM, Ad”	1

Downloaded from Arrow.com.

Table 2.19 Number of Cycles for String Manipulation Instructions

Instruction	Mnemonic (indicates the common operation when the size is omitted)	Number of Cycles
String manipulation instructions*1	• SCMPU	2+4×floor(n/4)+4×(n%4) n: Number of comparison bytes*2
	• SMOVB	n>3?6+3×floor(n/4)+3×(n%4):2+3n n: Number of transfer bytes*2
	• SMOVF, SMOVU	2+3×floor(n/4)+3×(n%4) n: Number of transfer bytes*2
	• SSTR.B	2+floor(n/4)+n%4 n: Number of transfer bytes*2
	• SSTR.W	2+floor(n/2)+n%2 n: Number of transfer words*2
	• SSTR.L	2+n n: Number of transfer longwords
	• SUNTIL.B, SWHILE.B	3+3×floor(n/4)+3×(n%4) n: Number of comparison bytes*2
	• SUNTIL.W, SWHILE.W	3+3×floor(n/2)+3×(n%2) n: Number of comparison words*2
	• SUNTIL.L, SWHILE.L	3+3×n n: Number of comparison longwords

?: Conditional operator

Note 1. Each of the SCMPU, SMOVU, SWHILE, and SUNTIL instructions ends the execution regardless of the specified cycles, if the end condition is satisfied during execution.
Note 2. floor (x): Max. integer that is smaller than x.

Table 2.20 Number of Cycles for System Manipulation Instructions

Instruction	Mnemonic (indicates the common operation when the size is omitted)	Number of Cycles
System manipulation instructions	• {CLRPSW, SETPSW}“flag” • MVTC “#IMM, CR”/“Rs, CR” • MVFC “CR, Rd” • MVTIPL “#IMM”	1
	• RTE	6
	• RTFI	3

Downloaded from Arrow.com.

2.8.2 Numbers of Cycles for Response to Interrupts

Table 2.21 lists numbers of cycles taken by processing for response to interrupts.

Table 2.21 Numbers of Cycles for Response to Interrupts

Type of Interrupt Request/Details of Processing	Fast Interrupt	Other Interrupts
ICU Judgment of priority order	2 cycles	
CPU Number of cycles from notification to acceptance of the interrupt request	N cycles (varies with the instruction being executed at the time the interrupt was received)	
CPU Pre-processing by hardware Saving the current PC and PSW values in RAM (or in control registers in the case of the fast interrupt) Reading of the vector Branching to the start of the exception handling routine	4 cycles	6 cycles

Times calculated from the values in Table 2.21 will be applicable when access to memory from the CPU is processed with no waiting. The RAM and code flash memory in this MCU allow such access. Numbers of cycles for response to interrupts can be minimized by placing program code (and vectors) in code flash memory and the stack in RAM. Furthermore, place the addresses where the exception handling routine start on 8-byte boundaries.
For information on the number of cycles from notification to acceptance of the interrupt request, indicated by N in the table above, see Table 2.13 to Table 2.20.
The timing of interrupt acceptance depends on the execution state of the instruction. For more information on this, see section 14.3.1, Acceptance Timing and Saved PC Value.

Downloaded from Arrow.com.

3. Operating Modes

3.1 Operating Mode Types and Selection

There are two types of operating-mode selection: one is be selected by the level on pins at the time of release from the reset state, and the other is selected by software after release from the reset state.
Table 3.1 shows the relationship between levels on the mode-setting pins (MD, UB) on release from the reset state and the operating mode selected at that time. For details on each of the operating modes, see section 3.3, Details of Operating Modes.

Table 3.1 Selection of Operating Modes by the Mode-Setting Pins

Mode-Setting Pin		
MD*1	UB	Operating Mode
Low	High	Boot mode (USB Interface)
Low	Low	Boot mode (SCI)
High	—	Single-chip mode

Note 1. Do not change the level on the MD pin while the MCU is operating.

The endian is selectable in single-chip mode. Endian is set by the MDE.MDE[2:0] bits in the option-setting memory. For the correspondence between the setting and endian, see Table 3.2.

Table 3.2 Selection of Endian

MDE.MDE[2:0] Bit Setting	Endian
000b	Big endian
111b	Little endian

Downloaded from Arrow.com.

3.2 Register Descriptions

3.2.1 Mode Monitor Register (MDMONR)

Address(es): 0008 0000h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MD
Value after reset:	0	0	0	0	0	0	0	x	0	0	0	0	0	0	0	0/1*1

Bit	Symbol	Bit Name	Description	R/W
b0	MD	MD Pin Status Flag	0: The MD pin is low. 1: The MD pin is high.	R
b7 to b1	—	Reserved	These bits are read as 0.	R
b8	—	Reserved	The read value is undefined.	R
b15 to b9	—	Reserved	These bits are read as 0.	R

Note 1. This affects the level on the MD pin at the time of release from the reset state.

Downloaded from Arrow.com.

3.2.2 System Control Register 1 (SYSCR1)

Address(es): 0008 0008h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RAME
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	RAME	RAM Enable	0: The RAM is disabled. 1: The RAM is enabled.	R/W
b15 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

RAME Bit (RAM Enable)

The RAME bit enables or disables the RAM.

A 0 should not be written to this bit during access to the RAM. When accessing the RAM immediately after changing the RAME bit from 0 (RAM disabled) to 1 (RAM enabled), make sure that the RAME bit is 1 before the access.

Even when the RAME bit is cleared to 0, the RAM retains its value. To retain the value in the RAM, keep the specified RAM standby voltage (VRAM). For details, see section 51, Electrical Characteristics.

Downloaded from Arrow.com.

3.3 Details of Operating Modes

3.3.1 Single-Chip Mode

In this mode, all I/O ports can be used as general input/output ports, peripheral function input/output, or interrupt input pins.
The chip starts up in single-chip mode if the high level is on the MD pin on release from the reset state.

3.3.2 Boot Mode

In this mode, the on-chip flash memory modifying program (boot program) stored in a dedicated area within the MCU operates. The on-chip flash memory (ROM and E2 DataFlash) can be modified from outside the MCU by using a USB or universal asynchronous receiver/transmitter (SCI1). For details, see section 50, Flash Memory (FLASH).
When a reset is released while the MD pin is low, boot mode is selected.

3.3.2.1 Boot Mode (USB Interface)

When a reset is released while the MD pin is low and the UB pin is high, boot mode (USB interface) is selected. For details on boot mode (USB interface), refer to section 50.8.1, Boot Mode (USB Interface).

3.3.2.2 Boot Mode (SCI)

When a reset is released while the MD pin is low and the UB pin is low, boot mode (SCI) is selected. For details on boot mode (SCI), refer to section 50.8.2, Boot Mode (SCI Interface).

Downloaded from Arrow.com.

3.4 Transitions of Operating Modes

3.4.1 Operating Mode Transitions Determined by the Mode-Setting Pins

Figure 3.1 shows operating mode transitions determined by the settings of the MD pin and the UB pin.

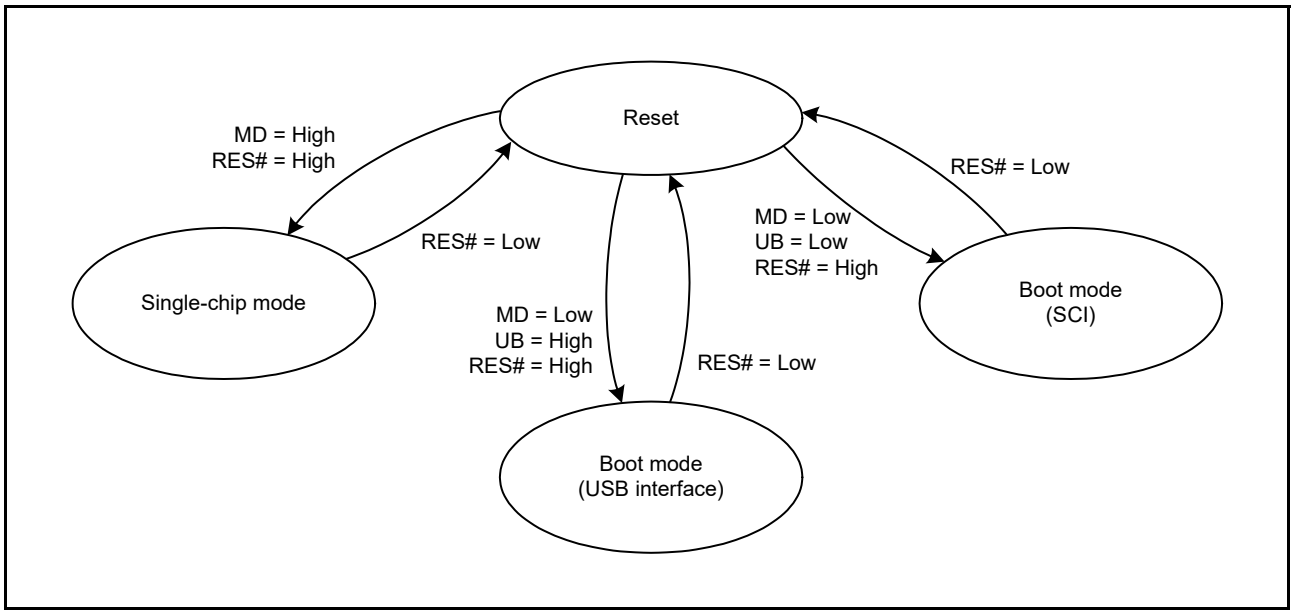


Figure 3.1 Mode-Setting Pin Levels and Operating Modes

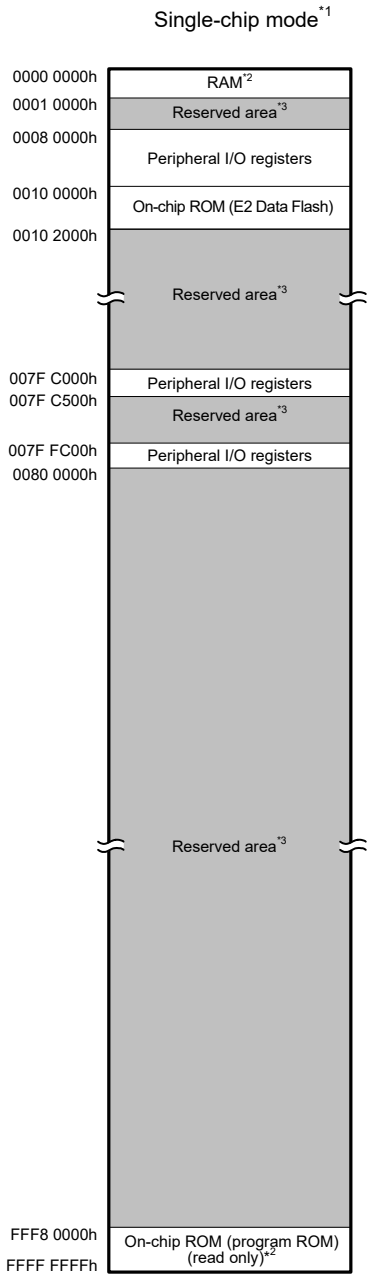
Downloaded from Arrow.com.

4. Address Space

4.1 Address Space

This MCU has a 4-Gbyte address space, consisting of the range of addresses from 0000 0000h to FFFF FFFFh. That is, linear access to an address space of up to 4 Gbytes is possible, and this contains both program and data areas.
Figure 4.1 shows the memory maps in the respective operating modes.

Downloaded from Arrow.com.



Note 1. The capacity of ROM differs depending on the products.

ROM (bytes)		RAM (bytes)	
Capacity	**Address**	**Capacity**	**Address**
512 Kbytes	FFF8 0000h to FFFF FFFFh	64 Kbytes	0000 0000h to 0000 FFFFh
384 Kbytes	FFFA 0000h to FFFF FFFFh		

Note: See Table 1.3 List of Products, for the product type name.

Figure 4.1 Memory Map in Each Operating Mode

Downloaded from Arrow.com.

5. I/O Registers

This section provides information on the on-chip I/O register addresses and bit configuration. The information is given as shown below. Notes on writing to registers are also given below.

(1) I/O register addresses (address order)

- Registers are listed from the lower allocation addresses.
- Registers are classified according to module symbols.
- Numbers of cycles for access indicate numbers of cycles of the given base clock.
- Among the internal I/O register area, addresses not listed in the list of registers are reserved. Reserved addresses must not be accessed. Do not access these addresses; otherwise, the operation when accessing these bits and subsequent operations cannot be guaranteed.

(2) Notes on writing to I/O registers

When writing to an I/O register, the CPU starts executing the subsequent instruction before completing I/O register write. This may cause the subsequent instruction to be executed before the post-update I/O register value is reflected on the operation.
As described in the following examples, special care is required for the cases in which the subsequent instruction must be executed after the post-update I/O register value is actually reflected.

[Examples of cases requiring special care]

- The subsequent instruction must be executed while an interrupt request is disabled with the IENj bit in IERn of the ICU (interrupt request enable bit) cleared to 0.
- A WAIT instruction is executed immediately after the preprocessing for causing a transition to the low power consumption state.

In the above cases, after writing to an I/O register, wait until the write operation is completed using the following procedure and then execute the subsequent instruction.

(a) Write to an I/O register.
(b) Read the value from the I/O register to a general register.
(c) Execute the operation using the value read.
(d) Execute the subsequent instruction.

[Instruction examples]

- Byte-size I/O registers

```
MOV.L  #SFR_ADDR, R1
MOV.B  #SFR_DATA, [R1]
CMP    [R1].UB, R1
;; Next process
```

- Word-size I/O registers

```
MOV.L  #SFR_ADDR, R1
MOV.W  #SFR_DATA, [R1]
CMP    [R1].W, R1
;; Next process
```


Downloaded from Arrow.com.

- Longword-size I/O registers

```
MOV.L  #SFR_ADDR, R1
MOV.L  #SFR_DATA, [R1]
CMP    [R1].L, R1
;; Next process
```

If multiple registers are written to and a subsequent instruction should be executed after the write operations are entirely completed, only read the I/O register that was last written to and execute the operation using the value; it is not necessary to read or execute operation for all the registers that were written to.

(3) Number of Access Cycles to I/O Registers

For numbers of clock cycles for access to I/O registers, see Table 5.1, List of I/O Registers (Address Order).
The number of access cycles to I/O registers is obtained by following equation.*1

Number of access cycles to I/O registers = Number of bus cycles for internal main bus 1 +
Number of divided clock synchronization cycles +
Number of bus cycles for internal peripheral bus 1 to 6

The number of bus cycles of internal peripheral bus 1 to 6 differs according to the register to be accessed.
When peripheral functions connected to internal peripheral bus 2 to 6 are accessed, the number of divided clock synchronization cycles is added.
The number of divided clock synchronization cycles differs depending on the frequency ratio between ICLK and PCLK (or FCLK) or bus access timing.
In the peripheral function unit, when the frequency ratio of ICLK is equal to or greater than that of PCLK (or FCLK), the sum of the number of bus cycles for internal main bus 1 and the number of the divided clock synchronization cycles will be one cycle of PCLK (or FCLK) at a maximum. Therefore, one PCLK (or FCLK) has been added to the number of access cycles shown in Table 5.1.
When the frequency ratio of ICLK is lower than that of PCLK (or FCLK), the subsequent bus access is started from the ICLK cycle following the completion of the access to the peripheral functions. Therefore, the access cycles are described on an ICLK basis.

Note 1. This applies to the number of cycles when the access from the CPU does not conflict with the instruction fetching to the external memory or bus access from the different bus master (DMAC or DTC).

(4) Restrictions in Relation to RMPA and String-Manipulation Instructions

The allocation of data to be handled by RMPA or string-manipulation instructions to I/O registers is prohibited, and operation is not guaranteed if this restriction is not observed.

(5) Notes on Sleep Mode and Mode Transitions

During sleep mode or mode transitions, do not write to the system control related registers (indicated by 'SYSTEM' in the Module Symbol column in Table 5.1, List of I/O Registers (Address Order)).

Downloaded from Arrow.com.

5.1 I/O Register Addresses (Address Order)

Table 5.1 List of I/O Registers (Address Order) (1/31)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK <PCLK	
0008 0000h	SYSTEM	Mode Monitor Register	MDMONR	16	16	3 ICLK		section 3.
0008 0008h	SYSTEM	System Control Register 1	SYSCR1	16	16	3 ICLK		section 3.
0008 000Ch	SYSTEM	Standby Control Register	SBYCR	16	16	3 ICLK		section 11.
0008 0010h	SYSTEM	Module Stop Control Register A	MSTPCRA	32	32	3 ICLK		section 11.
0008 0014h	SYSTEM	Module Stop Control Register B	MSTPCRB	32	32	3 ICLK		section 11.
0008 0018h	SYSTEM	Module Stop Control Register C	MSTPCRC	32	32	3 ICLK		section 11.
0008 001Ch	SYSTEM	Module Stop Control Register D	MSTPCRD	32	32	3 ICLK		section 11.
0008 0020h	SYSTEM	System Clock Control Register	SCKCR	32	32	3 ICLK		section 9.
0008 0026h	SYSTEM	System Clock Control Register 3	SCKCR3	16	16	3 ICLK		section 9.
0008 0028h	SYSTEM	PLL Control Register	PLLCR	16	16	3 ICLK		section 9.
0008 002Ah	SYSTEM	PLL Control Register 2	PLLCR2	8	8	3 ICLK		section 9.
0008 002Ch	SYSTEM	USB-dedicated PLL Control Register	UPLLCR	16	16	3 ICLK		section 9.
0008 002Eh	SYSTEM	USB-dedicated PLL Control Register 2	UPLLCR2	8	8	3 ICLK		section 9.
0008 0031h	SYSTEM	Memory Wait Cycle Setting Register	MEMWAIT	8	8	3 ICLK		section 9.
0008 0032h	SYSTEM	Main Clock Oscillator Control Register	MOSCCR	8	8	3 ICLK		section 9.
0008 0033h	SYSTEM	Sub-Clock Oscillator Control Register	SOSCCR	8	8	3 ICLK		section 9.
0008 0034h	SYSTEM	Low-Speed On-Chip Oscillator Control Register	LOCOCR	8	8	3 ICLK		section 9.
0008 0035h	SYSTEM	IWDT-Dedicated On-Chip Oscillator Control Register	ILOCOCR	8	8	3 ICLK		section 9.
0008 0036h	SYSTEM	High-Speed On-Chip Oscillator Control Register	HOCOCR	8	8	3 ICLK		section 9.
0008 0037h	SYSTEM	High-Speed On-Chip Oscillator Control Register 2	HOCOCR2	8	8	3 ICLK		section 9.
0008 003Ch	SYSTEM	Oscillation Stabilization Flag Register	OSCOVFSR	8	8	3 ICLK		section 9.
0008 003Eh	SYSTEM	CLKOUT Output Control Register	CKOCR	16	16	3 ICLK		section 9.
0008 0040h	SYSTEM	Oscillation Stop Detection Control Register	OSTDCR	8	8	3 ICLK		section 9.
0008 0041h	SYSTEM	Oscillation Stop Detection Status Register	OSTDSR	8	8	3 ICLK		section 9.
0008 0060h	SYSTEM	Low-Speed On-Chip Oscillator Trimming Register	LOCOTRR	8	8	3 ICLK		section 9.
0008 0064h	SYSTEM	IWDT-Dedicated On-Chip Oscillator Trimming Register	ILOCOTRR	8	8	3 ICLK		section 9.
0008 0068h	SYSTEM	High-Speed On-Chip Oscillator Trimming Register 0	HOCOTRR0	8	8	3 ICLK		section 9.
0008 006Bh	SYSTEM	High-Speed On-Chip Oscillator Trimming Register 3	HOCOTRR3	8	8	3 ICLK		section 9.
0008 00A0h	SYSTEM	Operating Power Control Register	OPCCR	8	8	3 ICLK		section 11.
0008 00A1h	SYSTEM	Sleep Mode Return Clock Source Switching Register	RSTCKCR	8	8	3 ICLK		section 11.
0008 00A2h	SYSTEM	Main Clock Oscillator Wait Control Register	MOSCWTCR	8	8	3 ICLK		section 9.
0008 00AAh	SYSTEM	Sub Operating Power Control Register	SOPCCR	8	8	3 ICLK		section 11.
0008 00B0h	LPT	Low-Power Timer Control Register 1	LPTCR1	8	8	3 ICLK		section 29.
0008 00B1h	LPT	Low-Power Timer Control Register 2	LPTCR2	8	8	3 ICLK		section 29.
0008 00B2h	LPT	Low-Power Timer Control Register 3	LPTCR3	8	8	3 ICLK		section 29.
0008 00B4h	LPT	Low-Power Timer Period Setting Register	LPTPRD	16	16	3 ICLK		section 29.
0008 00B8h	LPT	Low-Power Timer Compare Register 0	LPCMR0	16	16	3 ICLK		section 29.
0008 00BCh	LPT	Low-Power Timer Standby Wakeup Enable Register	LPWUCR	16	16	3 ICLK		section 29.
0008 00C0h	SYSTEM	Reset Status Register 2	RSTSR2	8	8	3 ICLK		section 6.
0008 00C2h	SYSTEM	Software Reset Register	SWRR	16	16	3 ICLK		section 6.
0008 00E0h	SYSTEM	Voltage Monitoring 1 Circuit Control Register 1	LVD1CR1	8	8	3 ICLK		section 8.
0008 00E1h	SYSTEM	Voltage Monitoring 1 Circuit Status Register	LVD1SR	8	8	3 ICLK		section 8.
0008 03FEh	SYSTEM	Protect Register	PRCR	16	16	3 ICLK		section 13.
0008 1300h	BSC	Bus Error Status Clear Register	BERCLR	8	8	2 ICLK		section 16.
0008 1304h	BSC	Bus Error Monitoring Enable Register	BEREN	8	8	2 ICLK		section 16.
0008 1308h	BSC	Bus Error Status Register 1	BERSR1	8	8	2 ICLK		section 16.
0008 130Ah	BSC	Bus Error Status Register 2	BERSR2	16	16	2 ICLK		section 16.
0008 1310h	BSC	Bus Priority Control Register	BUSPRI	16	16	2 ICLK		section 16.
0008 2000h	DMAC0	DMA Source Address Register	DMSAR	32	32	2 ICLK		section 18.

Downloaded from Arrow.com.

Table 5.1 List of I/O Registers (Address Order) (2/31)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK <PCLK	
0008 2004h	DMAC0	DMA Destination Address Register	DMDAR	32	32	2 ICLK		section 18.
0008 2008h	DMAC0	DMA Transfer Count Register	DMCRA	32	32	2 ICLK		section 18.
0008 200Ch	DMAC0	DMA Block Transfer Count Register	DMCRB	16	16	2 ICLK		section 18.
0008 2010h	DMAC0	DMA Transfer Mode Register	DMTMD	16	16	2 ICLK		section 18.
0008 2013h	DMAC0	DMA Interrupt Setting Register	DMINT	8	8	2 ICLK		section 18.
0008 2014h	DMAC0	DMA Address Mode Register	DMAMD	16	16	2 ICLK		section 18.
0008 2018h	DMAC0	DMA Offset Register	DMOFR	32	32	2 ICLK		section 18.
0008 201Ch	DMAC0	DMA Transfer Enable Register	DMCNT	8	8	2 ICLK		section 18.
0008 201Dh	DMAC0	DMA Software Start Register	DMREQ	8	8	2 ICLK		section 18.
0008 201Eh	DMAC0	DMA Status Register	DMSTS	8	8	2 ICLK		section 18.
0008 201Fh	DMAC0	DMA Activation Source Flag Control Register	DMCSL	8	8	2 ICLK		section 18.
0008 2040h	DMAC1	DMA Source Address Register	DMSAR	32	32	2 ICLK		section 18.
0008 2044h	DMAC1	DMA Destination Address Register	DMDAR	32	32	2 ICLK		section 18.
0008 2048h	DMAC1	DMA Transfer Count Register	DMCRA	32	32	2 ICLK		section 18.
0008 204Ch	DMAC1	DMA Block Transfer Count Register	DMCRB	16	16	2 ICLK		section 18.
0008 2050h	DMAC1	DMA Transfer Mode Register	DMTMD	16	16	2 ICLK		section 18.
0008 2053h	DMAC1	DMA Interrupt Setting Register	DMINT	8	8	2 ICLK		section 18.
0008 2054h	DMAC1	DMA Address Mode Register	DMAMD	16	16	2 ICLK		section 18.
0008 205Ch	DMAC1	DMA Transfer Enable Register	DMCNT	8	8	2 ICLK		section 18.
0008 205Dh	DMAC1	DMA Software Start Register	DMREQ	8	8	2 ICLK		section 18.
0008 205Eh	DMAC1	DMA Status Register	DMSTS	8	8	2 ICLK		section 18.
0008 205Fh	DMAC1	DMA Activation Source Flag Control Register	DMCSL	8	8	2 ICLK		section 18.
0008 2080h	DMAC2	DMA Source Address Register	DMSAR	32	32	2 ICLK		section 18.
0008 2084h	DMAC2	DMA Destination Address Register	DMDAR	32	32	2 ICLK		section 18.
0008 2088h	DMAC2	DMA Transfer Count Register	DMCRA	32	32	2 ICLK		section 18.
0008 208Ch	DMAC2	DMA Block Transfer Count Register	DMCRB	16	16	2 ICLK		section 18.
0008 2090h	DMAC2	DMA Transfer Mode Register	DMTMD	16	16	2 ICLK		section 18.
0008 2093h	DMAC2	DMA Interrupt Setting Register	DMINT	8	8	2 ICLK		section 18.
0008 2094h	DMAC2	DMA Address Mode Register	DMAMD	16	16	2 ICLK		section 18.
0008 209Ch	DMAC2	DMA Transfer Enable Register	DMCNT	8	8	2 ICLK		section 18.
0008 209Dh	DMAC2	DMA Software Start Register	DMREQ	8	8	2 ICLK		section 18.
0008 209Eh	DMAC2	DMA Status Register	DMSTS	8	8	2 ICLK		section 18.
0008 209Fh	DMAC2	DMA Activation Source Flag Control Register	DMCSL	8	8	2 ICLK		section 18.
0008 20C0h	DMAC3	DMA Source Address Register	DMSAR	32	32	2 ICLK		section 18.
0008 20C4h	DMAC3	DMA Destination Address Register	DMDAR	32	32	2 ICLK		section 18.
0008 20C8h	DMAC3	DMA Transfer Count Register	DMCRA	32	32	2 ICLK		section 18.
0008 20CCh	DMAC3	DMA Block Transfer Count Register	DMCRB	16	16	2 ICLK		section 18.
0008 20D0h	DMAC3	DMA Transfer Mode Register	DMTMD	16	16	2 ICLK		section 18.
0008 20D3h	DMAC3	DMA Interrupt Setting Register	DMINT	8	8	2 ICLK		section 18.
0008 20D4h	DMAC3	DMA Address Mode Register	DMAMD	16	16	2 ICLK		section 18.
0008 20DCh	DMAC3	DMA Transfer Enable Register	DMCNT	8	8	2 ICLK		section 18.
0008 20DDh	DMAC3	DMA Software Start Register	DMREQ	8	8	2 ICLK		section 18.
0008 20DEh	DMAC3	DMA Status Register	DMSTS	8	8	2 ICLK		section 18.
0008 20DFh	DMAC3	DMA Activation Source Flag Control Register	DMCSL	8	8	2 ICLK		section 18.
0008 2200h	DMAC	DMA Module Activation Register	DMAST	8	8	2 ICLK		section 18.
0008 2400h	DTC	DTC Control Register	DTCCR	8	8	2 ICLK		section 19.
0008 2404h	DTC	DTC Vector Base Register	DTCVBR	32	32	2 ICLK		section 19.
0008 2408h	DTC	DTC Address Mode Register	DTCADMOD	8	8	2 ICLK		section 19.
0008 240Ch	DTC	DTC Module Start Register	DTCST	8	8	2 ICLK		section 19.
0008 240Eh	DTC	DTC Status Register	DTCSTS	16	16	2 ICLK		section 19.
0008 6400h	MPU	Region-0 Start Page Number Register	RSPAGE0	32	32	1 ICLK		section 17.
0008 6404h	MPU	Region-0 End Page Number Register	REPAGE0	32	32	1 ICLK		section 17.

Downloaded from Arrow.com.

Table 5.1 List of I/O Registers (Address Order) (3/31)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK <PCLK	
0008 6408h	MPU	Region-1 Start Page Number Register	RSPAGE1	32	32	1 ICLK		section 17.
0008 640Ch	MPU	Region-1 End Page Number Register	REPAGE1	32	32	1 ICLK		section 17.
0008 6410h	MPU	Region-2 Start Page Number Register	RSPAGE2	32	32	1 ICLK		section 17.
0008 6414h	MPU	Region-2 End Page Number Register	REPAGE2	32	32	1 ICLK		section 17.
0008 6418h	MPU	Region-3 Start Page Number Register	RSPAGE3	32	32	1 ICLK		section 17.
0008 641Ch	MPU	Region-3 End Page Number Register	REPAGE3	32	32	1 ICLK		section 17.
0008 6420h	MPU	Region-4 Start Page Number Register	RSPAGE4	32	32	1 ICLK		section 17.
0008 6424h	MPU	Region-4 End Page Number Register	REPAGE4	32	32	1 ICLK		section 17.
0008 6428h	MPU	Region-5 Start Page Number Register	RSPAGE5	32	32	1 ICLK		section 17.
0008 642Ch	MPU	Region-5 End Page Number Register	REPAGE5	32	32	1 ICLK		section 17.
0008 6430h	MPU	Region-6 Start Page Number Register	RSPAGE6	32	32	1 ICLK		section 17.
0008 6434h	MPU	Region-6 End Page Number Register	REPAGE6	32	32	1 ICLK		section 17.
0008 6438h	MPU	Region-7 Start Page Number Register	RSPAGE7	32	32	1 ICLK		section 17.
0008 643Ch	MPU	Region-7 End Page Number Register	REPAGE7	32	32	1 ICLK		section 17.
0008 6500h	MPU	Memory-Protection Enable Register	MPEN	32	32	1 ICLK		section 17.
0008 6504h	MPU	Background Access Control Register	MPBAC	32	32	1 ICLK		section 17.
0008 6508h	MPU	Memory-Protection Error Status-Clearing Register	MPECLR	32	32	1 ICLK		section 17.
0008 650Ch	MPU	Memory-Protection Error Status Register	MPESTS	32	32	1 ICLK		section 17.
0008 6514h	MPU	Data Memory-Protection Error Address Register	MPDEA	32	32	1 ICLK		section 17.
0008 6520h	MPU	Region Search Address Register	MPSA	32	32	1 ICLK		section 17.
0008 6524h	MPU	Region Search Operation Register	MPOPS	16	16	1 ICLK		section 17.
0008 6526h	MPU	Region Invalidation Operation Register	MPOPI	16	16	1 ICLK		section 17.
0008 6528h	MPU	Instruction-Hit Region Register	MHITI	32	32	1 ICLK		section 17.
0008 652Ch	MPU	Data-Hit Region Register	MHITD	32	32	1 ICLK		section 17.
0008 7010h to 0008 70FFh	ICU	Interrupt Request Register 016 to Interrupt Request Register 255	IR016 to IR255	8	8	2 ICLK		section 15.
0008 711Bh to 0008 71FFh	ICU	DTC Transfer Request Enable Register 027 to DTC Transfer Request Enable Register 255	DTCER027 to DTCER255	8	8	2 ICLK		section 15.
0008 7202h to 0008 721Fh	ICU	Interrupt Request Enable Register 02 to Interrupt Request Enable Register 1F	IER02 to IER1F	8	8	2 ICLK		section 15.
0008 72E0h	ICU	Software Interrupt Generation Register	SWINTR	8	8	2 ICLK		section 15.
0008 72F0h	ICU	Fast Interrupt Set Register	FIR	16	16	2 ICLK		section 15.
0008 7300h to 0008 73FFh	ICU	Interrupt Source Priority Register 000 to Interrupt Source Priority Register 255	IPR000 to IPR255	8	8	2 ICLK		section 15.
0008 7400h	ICU	DMAC Trigger Select Register 0	DMRSR0	8	8	2 ICLK		section 15.
0008 7404h	ICU	DMAC Trigger Select Register 1	DMRSR1	8	8	2 ICLK		section 15.
0008 7408h	ICU	DMAC Trigger Select Register 2	DMRSR2	8	8	2 ICLK		section 15.
0008 740Ch	ICU	DMAC Trigger Select Register 3	DMRSR3	8	8	2 ICLK		section 15.
0008 7500h to 0008 7507h	ICU	IRQ Control Register 0 to IRQ Control Register 7	IRQCR0 to IRQCR7	8	8	2 ICLK		section 15.
0008 7510h	ICU	IRQ Pin Digital Filter Enable Register 0	IRQFLTE0	8	8	2 ICLK		section 15.
0008 7514h	ICU	IRQ Pin Digital Filter Setting Register 0	IRQFLTC0	16	16	2 ICLK		section 15.
0008 7580h	ICU	Non-Maskable Interrupt Status Register	NMISR	8	8	2 ICLK		section 15.
0008 7581h	ICU	Non-Maskable Interrupt Enable Register	NMIER	8	8	2 ICLK		section 15.
0008 7582h	ICU	Non-Maskable Interrupt Status Clear Register	NMICLR	8	8	2 ICLK		section 15.
0008 7583h	ICU	NMI Pin Interrupt Control Register	NMICR	8	8	2 ICLK		section 15.
0008 7590h	ICU	NMI Pin Digital Filter Enable Register	NMIFLTE	8	8	2 ICLK		section 15.
0008 7594h	ICU	NMI Pin Digital Filter Setting Register	NMIFLTC	8	8	2 ICLK		section 15.
0008 8000h	CMT	Compare Match Timer Start Register 0	CMSTR0	16	16	2 or 3 PCLKB	2 ICLK	section 27.
0008 8002h	CMT0	Compare Match Timer Control Register	CMCR	16	16	2 or 3 PCLKB	2 ICLK	section 27.
0008 8004h	CMT0	Compare Match Counter	CMCNT	16	16	2 or 3 PCLKB	2 ICLK	section 27.

Downloaded from Arrow.com.

Table 5.1 List of I/O Registers (Address Order) (4/31)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK <PCLK	
0008 8006h	CMT0	Compare Match Constant Register	CMCOR	16	16	2 or 3 PCLKB	2 ICLK	section 27.
0008 8008h	CMT1	Compare Match Timer Control Register	CMCR	16	16	2 or 3 PCLKB	2 ICLK	section 27.
0008 800Ah	CMT1	Compare Match Counter	CMCNT	16	16	2 or 3 PCLKB	2 ICLK	section 27.
0008 800Ch	CMT1	Compare Match Constant Register	CMCOR	16	16	2 or 3 PCLKB	2 ICLK	section 27.
0008 8010h	CMT	Compare Match Timer Start Register 1	CMSTR1	16	16	2 or 3 PCLKB	2 ICLK	section 27.
0008 8012h	CMT2	Compare Match Timer Control Register	CMCR	16	16	2 or 3 PCLKB	2 ICLK	section 27.
0008 8014h	CMT2	Compare Match Counter	CMCNT	16	16	2 or 3 PCLKB	2 ICLK	section 27.
0008 8016h	CMT2	Compare Match Constant Register	CMCOR	16	16	2 or 3 PCLKB	2 ICLK	section 27.
0008 8018h	CMT3	Compare Match Timer Control Register	CMCR	16	16	2 or 3 PCLKB	2 ICLK	section 27.
0008 801Ah	CMT3	Compare Match Counter	CMCNT	16	16	2 or 3 PCLKB	2 ICLK	section 27.
0008 801Ch	CMT3	Compare Match Constant Register	CMCOR	16	16	2 or 3 PCLKB	2 ICLK	section 27.
0008 8020h	WDT	WDT Refresh Register	WDTRR	8	8	2 or 3 PCLKB	2 ICLK	section 30.
0008 8022h	WDT	WDT Control Register	WDTCR	16	16	2 or 3 PCLKB	2 ICLK	section 30.
0008 8024h	WDT	WDT Status Register	WDTSR	16	16	2 or 3 PCLKB	2 ICLK	section 30.
0008 8026h	WDT	WDT Reset Control Register	WDTRCR	8	8	2 or 3 PCLKB	2 ICLK	section 30.
0008 8030h	IWDT	IWDT Refresh Register	IWDTRR	8	8	2 or 3 PCLKB	2 ICLK	section 31.
0008 8032h	IWDT	IWDT Control Register	IWDTCR	16	16	2 or 3 PCLKB	2 ICLK	section 31.
0008 8034h	IWDT	IWDT Status Register	IWDTSR	16	16	2 or 3 PCLKB	2 ICLK	section 31.
0008 8036h	IWDT	IWDT Reset Control Register	IWDTRCR	8	8	2 or 3 PCLKB	2 ICLK	section 31.
0008 8038h	IWDT	IWDT Count Stop Control Register	IWDTCSTPR	8	8	2 or 3 PCLKB	2 ICLK	section 31.
0008 8040h	DA	D/A Data Register 0	DADR0	16	16	2 or 3 PCLKB	2 ICLK	section 45.
0008 8042h	DA	D/A Data Register 1	DADR1	16	16	2 or 3 PCLKB	2 ICLK	section 45.
0008 8044h	DA	D/A Control Register	DACR	8	8	2 or 3 PCLKB	2 ICLK	section 45.
0008 8045h	DA	Data Register Format Select Register	DADPR	8	8	2 or 3 PCLKB	2 ICLK	section 45.
0008 8046h	DA	D/A A/D Synchronous Start Control Register	DAADSCR	8	8	2 or 3 PCLKB	2 ICLK	section 45.
0008 8047h	DA	D/A VREF Control Register	DAVREFCR	8	8	2 or 3 PCLKB	2 ICLK	section 45.
0008 8100h	TPU	Timer Start Register	TSTR	8	8	2 or 3 PCLKB	2 ICLK	section 25.
0008 8101h	TPU	Timer Synchronous Register	TSYR	8	8	2 or 3 PCLKB	2 ICLK	section 25.
0008 8108h	TPU0	Noise Filter Control Register	NFCR	8	8	2 or 3 PCLKB	2 ICLK	section 25.
0008 8109h	TPU1	Noise Filter Control Register	NFCR	8	8	2 or 3 PCLKB	2 ICLK	section 25.
0008 810Ah	TPU2	Noise Filter Control Register	NFCR	8	8	2 or 3 PCLKB	2 ICLK	section 25.
0008 810Bh	TPU3	Noise Filter Control Register	NFCR	8	8	2 or 3 PCLKB	2 ICLK	section 25.
0008 810Ch	TPU4	Noise Filter Control Register	NFCR	8	8	2 or 3 PCLKB	2 ICLK	section 25.
0008 810Dh	TPU5	Noise Filter Control Register	NFCR	8	8	2 or 3 PCLKB	2 ICLK	section 25.
0008 8110h	TPU0	Timer Control Register	TCR	8	8	2 or 3 PCLKB	2 ICLK	section 25.
0008 8111h	TPU0	Timer Mode Register	TMDR	8	8	2 or 3 PCLKB	2 ICLK	section 25.
0008 8112h	TPU0	Timer I/O Control Register H	TIORH	8	8	2 or 3 PCLKB	2 ICLK	section 25.
0008 8114h	TPU0	Timer Interrupt Enable Register	TIER	8	8	2 or 3 PCLKB	2 ICLK	section 25.
0008 8115h	TPU0	Timer Status Register	TSR	8	8	2 or 3 PCLKB	2 ICLK	section 25.
0008 8116h	TPU0	Timer Counter	TCNT	16	16	2 or 3 PCLKB	2 ICLK	section 25.
0008 8118h	TPU0	Timer General Register A	TGRA	16	16	2 or 3 PCLKB	2 ICLK	section 25.
0008 811Ah	TPU0	Timer General Register B	TGRB	16	16	2 or 3 PCLKB	2 ICLK	section 25.
0008 811Ch	TPU0	Timer General Register C	TGRC	16	16	2 or 3 PCLKB	2 ICLK	section 25.
0008 811Eh	TPU0	Timer General Register D	TGRD	16	16	2 or 3 PCLKB	2 ICLK	section 25.
0008 8120h	TPU1	Timer Control Register	TCR	8	8	2 or 3 PCLKB	2 ICLK	section 25.
0008 8121h	TPU1	Timer Mode Register	TMDR	8	8	2 or 3 PCLKB	2 ICLK	section 25.
0008 8122h	TPU1	Timer I/O Control Register	TIOR	8	8	2 or 3 PCLKB	2 ICLK	section 25.
0008 8124h	TPU1	Timer Interrupt Enable Register	TIER	8	8	2 or 3 PCLKB	2 ICLK	section 25.
0008 8125h	TPU1	Timer Status Register	TSR	8	8	2 or 3 PCLKB	2 ICLK	section 25.
0008 8126h	TPU1	Timer Counter	TCNT	16	16	2 or 3 PCLKB	2 ICLK	section 25.
0008 8128h	TPU1	Timer General Register A	TGRA	16	16	2 or 3 PCLKB	2 ICLK	section 25.
0008 812Ah	TPU1	Timer General Register B	TGRB	16	16	2 or 3 PCLKB	2 ICLK	section 25.

Downloaded from Arrow.com.

Table 5.1 List of I/O Registers (Address Order) (5/31)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK <PCLK	
0008 8130h	TPU2	Timer Control Register	TCR	8	8	2 or 3 PCLKB	2 ICLK	section 25.
0008 8131h	TPU2	Timer Mode Register	TMDR	8	8	2 or 3 PCLKB	2 ICLK	section 25.
0008 8132h	TPU2	Timer I/O Control Register	TIOR	8	8	2 or 3 PCLKB	2 ICLK	section 25.
0008 8134h	TPU2	Timer Interrupt Enable Register	TIER	8	8	2 or 3 PCLKB	2 ICLK	section 25.
0008 8135h	TPU2	Timer Status Register	TSR	8	8	2 or 3 PCLKB	2 ICLK	section 25.
0008 8136h	TPU2	Timer Counter	TCNT	16	16	2 or 3 PCLKB	2 ICLK	section 25.
0008 8138h	TPU2	Timer General Register A	TGRA	16	16	2 or 3 PCLKB	2 ICLK	section 25.
0008 813Ah	TPU2	Timer General Register B	TGRB	16	16	2 or 3 PCLKB	2 ICLK	section 25.
0008 8140h	TPU3	Timer Control Register	TCR	8	8	2 or 3 PCLKB	2 ICLK	section 25.
0008 8141h	TPU3	Timer Mode Register	TMDR	8	8	2 or 3 PCLKB	2 ICLK	section 25.
0008 8142h	TPU3	Timer I/O Control Register H	TIORH	8	8	2 or 3 PCLKB	2 ICLK	section 25.
0008 8143h	TPU3	Timer I/O Control Register L	TIORL	8	8	2 or 3 PCLKB	2 ICLK	section 25.
0008 8144h	TPU3	Timer Interrupt Enable Register	TIER	8	8	2 or 3 PCLKB	2 ICLK	section 25.
0008 8145h	TPU3	Timer Status Register	TSR	8	8	2 or 3 PCLKB	2 ICLK	section 25.
0008 8146h	TPU3	Timer Counter	TCNT	16	16	2 or 3 PCLKB	2 ICLK	section 25.
0008 8148h	TPU3	Timer General Register A	TGRA	16	16	2 or 3 PCLKB	2 ICLK	section 25.
0008 814Ah	TPU3	Timer General Register B	TGRB	16	16	2 or 3 PCLKB	2 ICLK	section 25.
0008 814Ch	TPU3	Timer General Register C	TGRC	16	16	2 or 3 PCLKB	2 ICLK	section 25.
0008 814Eh	TPU3	Timer General Register D	TGRD	16	16	2 or 3 PCLKB	2 ICLK	section 25.
0008 8150h	TPU4	Timer Control Register	TCR	8	8	2 or 3 PCLKB	2 ICLK	section 25.
0008 8151h	TPU4	Timer Mode Register	TMDR	8	8	2 or 3 PCLKB	2 ICLK	section 25.
0008 8152h	TPU4	Timer I/O Control Register	TIOR	8	8	2 or 3 PCLKB	2 ICLK	section 25.
0008 8154h	TPU4	Timer Interrupt Enable Register	TIER	8	8	2 or 3 PCLKB	2 ICLK	section 25.
0008 8155h	TPU4	Timer Status Register	TSR	8	8	2 or 3 PCLKB	2 ICLK	section 25.
0008 8156h	TPU4	Timer Counter	TCNT	16	16	2 or 3 PCLKB	2 ICLK	section 25.
0008 8158h	TPU4	Timer General Register A	TGRA	16	16	2 or 3 PCLKB	2 ICLK	section 25.
0008 815Ah	TPU4	Timer General Register B	TGRB	16	16	2 or 3 PCLKB	2 ICLK	section 25.
0008 8160h	TPU5	Timer Control Register	TCR	8	8	2 or 3 PCLKB	2 ICLK	section 25.
0008 8161h	TPU5	Timer Mode Register	TMDR	8	8	2 or 3 PCLKB	2 ICLK	section 25.
0008 8162h	TPU5	Timer I/O Control Register	TIOR	8	8	2 or 3 PCLKB	2 ICLK	section 25.
0008 8164h	TPU5	Timer Interrupt Enable Register	TIER	8	8	2 or 3 PCLKB	2 ICLK	section 25.
0008 8165h	TPU5	Timer Status Register	TSR	8	8	2 or 3 PCLKB	2 ICLK	section 25.
0008 8166h	TPU5	Timer Counter	TCNT	16	16	2 or 3 PCLKB	2 ICLK	section 25.
0008 8168h	TPU5	Timer General Register A	TGRA	16	16	2 or 3 PCLKB	2 ICLK	section 25.
0008 816Ah	TPU5	Timer General Register B	TGRB	16	16	2 or 3 PCLKB	2 ICLK	section 25.
0008 8200h	TMR0	Timer Control Register	TCR	8	8	2 or 3 PCLKB	2 ICLK	section 26.
0008 8201h	TMR1	Timer Control Register	TCR	8	8	2 or 3 PCLKB	2 ICLK	section 26.
0008 8202h	TMR0	Timer Control/Status Register	TCSR	8	8	2 or 3 PCLKB	2 ICLK	section 26.
0008 8203h	TMR1	Timer Control/Status Register	TCSR	8	8	2 or 3 PCLKB	2 ICLK	section 26.
0008 8204h	TMR0	Time Constant Register A	TCORA	8	8	2 or 3 PCLKB	2 ICLK	section 26.
0008 8204h	TMR01	Time Constant Register A	TCORA	16	16	2 or 3 PCLKB	2 ICLK	section 26.
0008 8205h	TMR1	Time Constant Register A	TCORA	8	8*1	2 or 3 PCLKB	2 ICLK	section 26.
0008 8206h	TMR0	Time Constant Register B	TCORB	8	8	2 or 3 PCLKB	2 ICLK	section 26.
0008 8206h	TMR01	Time Constant Register B	TCORB	16	16	2 or 3 PCLKB	2 ICLK	section 26.
0008 8207h	TMR1	Time Constant Register B	TCORB	8	8*1	2 or 3 PCLKB	2 ICLK	section 26.
0008 8208h	TMR0	Timer Counter	TCNT	8	8	2 or 3 PCLKB	2 ICLK	section 26.
0008 8208h	TMR01	Timer Counter	TCNT	16	16	2 or 3 PCLKB	2 ICLK	section 26.
0008 8209h	TMR1	Timer Counter	TCNT	8	8*1	2 or 3 PCLKB	2 ICLK	section 26.
0008 820Ah	TMR0	Timer Counter Control Register	TCCR	8	8	2 or 3 PCLKB	2 ICLK	section 26.
0008 820Ah	TMR01	Timer Counter Control Register	TCCR	16	16	2 or 3 PCLKB	2 ICLK	section 26.
0008 820Bh	TMR1	Timer Counter Control Register	TCCR	8	8*1	2 or 3 PCLKB	2 ICLK	section 26.
0008 820Ch	TMR0	Timer Counter Start Register	TCSTR	8	8	2 or 3 PCLKB	2 ICLK	section 26.

Downloaded from Arrow.com.

Table 5.1 List of I/O Registers (Address Order) (6/31)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK <PCLK	
0008 8210h	TMR2	Timer Control Register	TCR	8	8	2 or 3 PCLKB	2 ICLK	section 26.
0008 8211h	TMR3	Timer Control Register	TCR	8	8	2 or 3 PCLKB	2 ICLK	section 26.
0008 8212h	TMR2	Timer Control/Status Register	TCSR	8	8	2 or 3 PCLKB	2 ICLK	section 26.
0008 8214h	TMR2	Time Constant Register A	TCORA	8	8	2 or 3 PCLKB	2 ICLK	section 26.
0008 8214h	TMR23	Time Constant Register A	TCORA	16	16	2 or 3 PCLKB	2 ICLK	section 26.
0008 8215h	TMR3	Time Constant Register A	TCORA	8	8*1	2 or 3 PCLKB	2 ICLK	section 26.
0008 8216h	TMR2	Time Constant Register B	TCORB	8	8	2 or 3 PCLKB	2 ICLK	section 26.
0008 8216h	TMR23	Time Constant Register B	TCORB	16	16	2 or 3 PCLKB	2 ICLK	section 26.
0008 8217h	TMR3	Time Constant Register B	TCORB	8	8*1	2 or 3 PCLKB	2 ICLK	section 26.
0008 8218h	TMR2	Timer Counter	TCNT	8	8	2 or 3 PCLKB	2 ICLK	section 26.
0008 8218h	TMR23	Timer Counter	TCNT	16	16	2 or 3 PCLKB	2 ICLK	section 26.
0008 8219h	TMR3	Timer Counter	TCNT	8	8*1	2 or 3 PCLKB	2 ICLK	section 26.
0008 821Ah	TMR2	Timer Counter Control Register	TCCR	8	8	2 or 3 PCLKB	2 ICLK	section 26.
0008 821Ah	TMR23	Timer Counter Control Register	TCCR	16	16	2 or 3 PCLKB	2 ICLK	section 26.
0008 821Bh	TMR3	Timer Counter Control Register	TCCR	8	8*1	2 or 3 PCLKB	2 ICLK	section 26.
0008 821Ch	TMR2	Timer Counter Start Register	TCSTR	8	8	2 or 3 PCLKB	2 ICLK	section 26.
0008 8280h	CRC	CRC Control Register	CRCCR	8	8	2 or 3 PCLKB	2 ICLK	section 39.
0008 8281h	CRC	CRC Data Input Register	CRCDIR	8	8	2 or 3 PCLKB	2 ICLK	section 39.
0008 8282h	CRC	CRC Data Output Register	CRCDOR	16	16	2 or 3 PCLKB	2 ICLK	section 39.
0008 8300h	RIIC0	I²C-bus Control Register 1	ICCR1	8	8	2 or 3 PCLKB	2 ICLK	section 35.
0008 8301h	RIIC0	I²C-bus Control Register 2	ICCR2	8	8	2 or 3 PCLKB	2 ICLK	section 35.
0008 8302h	RIIC0	I²C-bus Mode Register 1	ICMR1	8	8	2 or 3 PCLKB	2 ICLK	section 35.
0008 8303h	RIIC0	I²C-bus Mode Register 2	ICMR2	8	8	2 or 3 PCLKB	2 ICLK	section 35.
0008 8304h	RIIC0	I²C-bus Mode Register 3	ICMR3	8	8	2 or 3 PCLKB	2 ICLK	section 35.
0008 8305h	RIIC0	I²C-bus Function Enable Register	ICFER	8	8	2 or 3 PCLKB	2 ICLK	section 35.
0008 8306h	RIIC0	I²C-bus Status Enable Register	ICSER	8	8	2 or 3 PCLKB	2 ICLK	section 35.
0008 8307h	RIIC0	I²C-bus Interrupt Enable Register	ICIER	8	8	2 or 3 PCLKB	2 ICLK	section 35.
0008 8308h	RIIC0	I²C-bus Status Register 1	ICSR1	8	8	2 or 3 PCLKB	2 ICLK	section 35.
0008 8309h	RIIC0	I²C-bus Status Register 2	ICSR2	8	8	2 or 3 PCLKB	2 ICLK	section 35.
0008 830Ah	RIIC0	Slave Address Register L0	SARL0	8	8	2 or 3 PCLKB	2 ICLK	section 35.
0008 830Bh	RIIC0	Slave Address Register U0	SARU0	8	8	2 or 3 PCLKB	2 ICLK	section 35.
0008 830Ch	RIIC0	Slave Address Register L1	SARL1	8	8	2 or 3 PCLKB	2 ICLK	section 35.
0008 830Dh	RIIC0	Slave Address Register U1	SARU1	8	8	2 or 3 PCLKB	2 ICLK	section 35.
0008 830Eh	RIIC0	Slave Address Register L2	SARL2	8	8	2 or 3 PCLKB	2 ICLK	section 35.
0008 830Fh	RIIC0	Slave Address Register U2	SARU2	8	8	2 or 3 PCLKB	2 ICLK	section 35.
0008 8310h	RIIC0	I²C-bus Bit Rate Low-Level Register	ICBRL	8	8	2 or 3 PCLKB	2 ICLK	section 35.
0008 8311h	RIIC0	I²C-bus Bit Rate High-Level Register	ICBRH	8	8	2 or 3 PCLKB	2 ICLK	section 35.
0008 8312h	RIIC0	I²C-bus Transmit Data Register	ICDRT	8	8	2 or 3 PCLKB	2 ICLK	section 35.
0008 8313h	RIIC0	I²C-bus Receive Data Register	ICDRR	8	8	2 or 3 PCLKB	2 ICLK	section 35.
0008 8380h	RSPI0	RSPI Control Register	SPCR	8	8	2 or 3 PCLKB	2 ICLK	section 38.
0008 8381h	RSPI0	RSPI Slave Select Polarity Register	SSLP	8	8	2 or 3 PCLKB	2 ICLK	section 38.
0008 8382h	RSPI0	RSPI Pin Control Register	SPPCR	8	8	2 or 3 PCLKB	2 ICLK	section 38.
0008 8383h	RSPI0	RSPI Status Register	SPSR	8	8	2 or 3 PCLKB	2 ICLK	section 38.
0008 8384h	RSPI0	RSPI Data Register	SPDR	32	16, 32	2 or 3 PCLKB	2 ICLK	section 38.
0008 8388h	RSPI0	RSPI Sequence Control Register	SPSCR	8	8	2 or 3 PCLKB	2 ICLK	section 38.
0008 8389h	RSPI0	RSPI Sequence Status Register	SPSSR	8	8	2 or 3 PCLKB	2 ICLK	section 38.
0008 838Ah	RSPI0	RSPI Bit Rate Register	SPBR	8	8	2 or 3 PCLKB	2 ICLK	section 38.
0008 838Bh	RSPI0	RSPI Data Control Register	SPDCR	8	8	2 or 3 PCLKB	2 ICLK	section 38.
0008 838Ch	RSPI0	RSPI Clock Delay Register	SPCKD	8	8	2 or 3 PCLKB	2 ICLK	section 38.
0008 838Dh	RSPI0	RSPI Slave Select Negation Delay Register	SSLND	8	8	2 or 3 PCLKB	2 ICLK	section 38.
0008 838Eh	RSPI0	RSPI Next-Access Delay Register	SPND	8	8	2 or 3 PCLKB	2 ICLK	section 38.
0008 838Fh	RSPI0	RSPI Control Register 2	SPCR2	8	8	2 or 3 PCLKB	2 ICLK	section 38.

Downloaded from Arrow.com.

Table 5.1 List of I/O Registers (Address Order) (7/31)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK <PCLK	
0008 8390h	RSPI0	RSPI Command Register 0	SPCMD0	16	16	2 or 3 PCLKB	2 ICLK	section 38.
0008 8392h	RSPI0	RSPI Command Register 1	SPCMD1	16	16	2 or 3 PCLKB	2 ICLK	section 38.
0008 8394h	RSPI0	RSPI Command Register 2	SPCMD2	16	16	2 or 3 PCLKB	2 ICLK	section 38.
0008 8396h	RSPI0	RSPI Command Register 3	SPCMD3	16	16	2 or 3 PCLKB	2 ICLK	section 38.
0008 8398h	RSPI0	RSPI Command Register 4	SPCMD4	16	16	2 or 3 PCLKB	2 ICLK	section 38.
0008 839Ah	RSPI0	RSPI Command Register 5	SPCMD5	16	16	2 or 3 PCLKB	2 ICLK	section 38.
0008 839Ch	RSPI0	RSPI Command Register 6	SPCMD6	16	16	2 or 3 PCLKB	2 ICLK	section 38.
0008 839Eh	RSPI0	RSPI Command Register 7	SPCMD7	16	16	2 or 3 PCLKB	2 ICLK	section 38.
0008 8410h	IRDA	IrDA Control Register	IRCR	8	8	2 or 3 PCLKB	2 ICLK	section 34.
0008 8900h	POE	Input Level Control/Status Register 1	ICSR1	16	8, 16	2 or 3 PCLKB	2 ICLK	section 24.
0008 8902h	POE	Output Level Control/Status Register 1	OCSR1	16	8, 16	2 or 3 PCLKB	2 ICLK	section 24.
0008 8908h	POE	Input Level Control/Status Register 2	ICSR2	16	8, 16	2 or 3 PCLKB	2 ICLK	section 24.
0008 890Ah	POE	Software Port Output Enable Register	SPOER	8	8	2 or 3 PCLKB	2 ICLK	section 24.
0008 890Bh	POE	Port Output Enable Control Register 1	POECR1	8	8	2 or 3 PCLKB	2 ICLK	section 24.
0008 890Ch	POE	Port Output Enable Control Register 2	POECR2	8	8	2 or 3 PCLKB	2 ICLK	section 24.
0008 890Eh	POE	Input Level Control/Status Register 3	ICSR3	16	8, 16	2 or 3 PCLKB	2 ICLK	section 24.
0008 9000h	S12AD	A/D Control Register	ADCSR	16	16	2 or 3 PCLKB	2 ICLK	section 44.
0008 9004h	S12AD	A/D Channel Select Register A0	ADANSA0	16	16	2 or 3 PCLKB	2 ICLK	section 44.
0008 9006h	S12AD	A/D Channel Select Register A1	ADANSA1	16	16	2 or 3 PCLKB	2 ICLK	section 44.
0008 9008h	S12AD	A/D-Converted Value Addition/Average Function Select Register 0	ADADS0	16	16	2 or 3 PCLKB	2 ICLK	section 44.
0008 900Ah	S12AD	A/D-Converted Value Addition/Average Function Select Register 1	ADADS1	16	16	2 or 3 PCLKB	2 ICLK	section 44.
0008 900Ch	S12AD	A/D-Converted Value Addition/Average Count Select Register	ADADC	8	8	2 or 3 PCLKB	2 ICLK	section 44.
0008 900Eh	S12AD	A/D Control Extended Register	ADCER	16	16	2 or 3 PCLKB	2 ICLK	section 44.
0008 9010h	S12AD	A/D Conversion Start Trigger Select Register	ADSTRGR	16	16	2 or 3 PCLKB	2 ICLK	section 44.
0008 9012h	S12AD	A/D Conversion Extended Input Control Register	ADEXICR	16	16	2 or 3 PCLKB	2 ICLK	section 44.
0008 9014h	S12AD	A/D Channel Select Register B0	ADANSB0	16	16	2 or 3 PCLKB	2 ICLK	section 44.
0008 9016h	S12AD	A/D Channel Select Register B1	ADANSB1	16	16	2 or 3 PCLKB	2 ICLK	section 44.
0008 9018h	S12AD	A/D Data Duplication Register	ADDBLDR	16	16	2 or 3 PCLKB	2 ICLK	section 44.
0008 901Ah	S12AD	A/D Temperature Sensor Data Register	ADTSDR	16	16	2 or 3 PCLKB	2 ICLK	section 44.
0008 901Ch	S12AD	A/D Internal Reference Voltage Data Register	ADOCDR	16	16	2 or 3 PCLKB	2 ICLK	section 44.
0008 901Eh	S12AD	A/D Self-Diagnosis Data Register	ADRD	16	16	2 or 3 PCLKB	2 ICLK	section 44.
0008 9020h	S12AD	A/D Data Register 0	ADDR0	16	16	2 or 3 PCLKB	2 ICLK	section 44.
0008 9022h	S12AD	A/D Data Register 1	ADDR1	16	16	2 or 3 PCLKB	2 ICLK	section 44.
0008 9024h	S12AD	A/D Data Register 2	ADDR2	16	16	2 or 3 PCLKB	2 ICLK	section 44.
0008 9026h	S12AD	A/D Data Register 3	ADDR3	16	16	2 or 3 PCLKB	2 ICLK	section 44.
0008 9028h	S12AD	A/D Data Register 4	ADDR4	16	16	2 or 3 PCLKB	2 ICLK	section 44.
0008 902Ah	S12AD	A/D Data Register 5	ADDR5	16	16	2 or 3 PCLKB	2 ICLK	section 44.
0008 902Ch	S12AD	A/D Data Register 6	ADDR6	16	16	2 or 3 PCLKB	2 ICLK	section 44.
0008 902Eh	S12AD	A/D Data Register 7	ADDR7	16	16	2 or 3 PCLKB	2 ICLK	section 44.
0008 9040h	S12AD	A/D Data Register 16	ADDR16	16	16	2 or 3 PCLKB	2 ICLK	section 44.
0008 9042h	S12AD	A/D Data Register 17	ADDR17	16	16	2 or 3 PCLKB	2 ICLK	section 44.
0008 9044h	S12AD	A/D Data Register 18	ADDR18	16	16	2 or 3 PCLKB	2 ICLK	section 44.
0008 9046h	S12AD	A/D Data Register 19	ADDR19	16	16	2 or 3 PCLKB	2 ICLK	section 44.
0008 9048h	S12AD	A/D Data Register 20	ADDR20	16	16	2 or 3 PCLKB	2 ICLK	section 44.
0008 9056h	S12AD	A/D Data Register 27	ADDR27	16	16	2 or 3 PCLKB	2 ICLK	section 44.
0008 907Ah	S12AD	A/D Disconnection Detection Control Register	ADDISCR	8	8	2 or 3 PCLKB	2 ICLK	section 44.
0008 907Dh	S12AD	A/D Event Link Control Register	ADELCCR	8	8	2 or 3 PCLKB	2 ICLK	section 44.
0008 9080h	S12AD	A/D Group Scan Priority Control Register	ADGSPCR	16	16	2 or 3 PCLKB	2 ICLK	section 44.
0008 908Ah	S12AD	A/D High-Potential/Low-Potential Reference Voltage Control Register	ADHVREFCNT	8	8	2 or 3 PCLKB	2 ICLK	section 44.
0008 908Ch	S12AD	A/D Compare Function Window A/B Status Monitor Register	ADWINMON	8	8	2 or 3 PCLKB	2 ICLK	section 44.

Downloaded from Arrow.com.

Table 5.1 List of I/O Registers (Address Order) (8/31)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK <PCLK	
0008 9090h	S12AD	A/D Compare Function Control Register	ADCMPCR	16	16	2 or 3 PCLKB	2 ICLK	section 44.
0008 9092h	S12AD	A/D Compare Function Window A Extended Input Select Register	ADCMPANSER	8	8	2 or 3 PCLKB	2 ICLK	section 44.
0008 9093h	S12AD	A/D Compare Function Window A Extended Input Comparison Condition Setting Register	ADCMPLER	8	8	2 or 3 PCLKB	2 ICLK	section 44.
0008 9094h	S12AD	A/D Compare Function Window A Channel Select Register 0	ADCMPANSR0	16	16	2 or 3 PCLKB	2 ICLK	section 44.
0008 9096h	S12AD	A/D Compare Function Window A Channel Select Register 1	ADCMPANSR1	16	16	2 or 3 PCLKB	2 ICLK	section 44.
0008 9098h	S12AD	A/D Compare Function Window A Comparison Condition Setting Register 0	ADCMPLR0	16	16	2 or 3 PCLKB	2 ICLK	section 44.
0008 909Ah	S12AD	A/D Compare Function Window A Comparison Condition Setting Register 1	ADCMPLR1	16	16	2 or 3 PCLKB	2 ICLK	section 44.
0008 909Ch	S12AD	A/D Compare Function Window A Lower-Side Level Setting Register	ADCMPDR0	16	16	2 or 3 PCLKB	2 ICLK	section 44.
0008 909Eh	S12AD	A/D Compare Function Window A Upper-Side Level Setting Register	ADCMPDR1	16	16	2 or 3 PCLKB	2 ICLK	section 44.
0008 90A0h	S12AD	A/D Compare Function Window A Channel Status Register 0	ADCMPSR0	16	16	2 or 3 PCLKB	2 ICLK	section 44.
0008 90A2h	S12AD	A/D Compare Function Window A Channel Status Register 1	ADCMPSR1	16	16	2 or 3 PCLKB	2 ICLK	section 44.
0008 90A4h	S12AD	A/D Compare Function Window A Extended Input Channel Status Register	ADCMPSER	16	16	2 or 3 PCLKB	2 ICLK	section 44.
0008 90A6h	S12AD	A/D Compare Function Window B Channel Select Register	ADCMPBNSR	8	8	2 or 3 PCLKB	2 ICLK	section 44.
0008 90A8h	S12AD	A/D Compare Function Window B Lower-Side Level Setting Register	ADWINLLB	16	16	2 or 3 PCLKB	2 ICLK	section 44.
0008 90AAh	S12AD	A/D Compare Function Window B Upper-Side Level Setting Register	ADWINULB	16	16	2 or 3 PCLKB	2 ICLK	section 44.
0008 90ACh	S12AD	A/D Compare Function Window B Channel Status Register	ADCMPBSR	8	8	2 or 3 PCLKB	2 ICLK	section 44.
0008 90B0h	S12AD	A/D Data Storage Buffer Register 0	ADBUF0	16	16	2 or 3 PCLKB	2 ICLK	section 44.
0008 90B2h	S12AD	A/D Data Storage Buffer Register 1	ADBUF1	16	16	2 or 3 PCLKB	2 ICLK	section 44.
0008 90B4h	S12AD	A/D Data Storage Buffer Register 2	ADBUF2	16	16	2 or 3 PCLKB	2 ICLK	section 44.
0008 90B6h	S12AD	A/D Data Storage Buffer Register 3	ADBUF3	16	16	2 or 3 PCLKB	2 ICLK	section 44.
0008 90B8h	S12AD	A/D Data Storage Buffer Register 4	ADBUF4	16	16	2 or 3 PCLKB	2 ICLK	section 44.
0008 90BAh	S12AD	A/D Data Storage Buffer Register 5	ADBUF5	16	16	2 or 3 PCLKB	2 ICLK	section 44.
0008 90BCh	S12AD	A/D Data Storage Buffer Register 6	ADBUF6	16	16	2 or 3 PCLKB	2 ICLK	section 44.
0008 90BEh	S12AD	A/D Data Storage Buffer Register 7	ADBUF7	16	16	2 or 3 PCLKB	2 ICLK	section 44.
0008 90C0h	S12AD	A/D Data Storage Buffer Register 8	ADBUF8	16	16	2 or 3 PCLKB	2 ICLK	section 44.
0008 90C2h	S12AD	A/D Data Storage Buffer Register 9	ADBUF9	16	16	2 or 3 PCLKB	2 ICLK	section 44.
0008 90C4h	S12AD	A/D Data Storage Buffer Register 10	ADBUF10	16	16	2 or 3 PCLKB	2 ICLK	section 44.
0008 90C6h	S12AD	A/D Data Storage Buffer Register 11	ADBUF11	16	16	2 or 3 PCLKB	2 ICLK	section 44.
0008 90C8h	S12AD	A/D Data Storage Buffer Register 12	ADBUF12	16	16	2 or 3 PCLKB	2 ICLK	section 44.
0008 90CAh	S12AD	A/D Data Storage Buffer Register 13	ADBUF13	16	16	2 or 3 PCLKB	2 ICLK	section 44.
0008 90CCh	S12AD	A/D Data Storage Buffer Register 14	ADBUF14	16	16	2 or 3 PCLKB	2 ICLK	section 44.
0008 90CEh	S12AD	A/D Data Storage Buffer Register 15	ADBUF15	16	16	2 or 3 PCLKB	2 ICLK	section 44.
0008 90D0h	S12AD	A/D Data Storage Buffer Enable Register	ADBUFEN	8	8	2 or 3 PCLKB	2 ICLK	section 44.
0008 90D2h	S12AD	A/D Data Storage Buffer Pointer Register	ADBUFPTR	8	8	2 or 3 PCLKB	2 ICLK	section 44.
0008 90DDh	S12AD	A/D Sampling State Register L	ADSSTRL	8	8	2 or 3 PCLKB	2 ICLK	section 44.
0008 90DEh	S12AD	A/D Sampling State Register T	ADSSTRT	8	8	2 or 3 PCLKB	2 ICLK	section 44.
0008 90DFh	S12AD	A/D Sampling State Register O	ADSSTRO	8	8	2 or 3 PCLKB	2 ICLK	section 44.
0008 90E0h	S12AD	A/D Sampling State Register 0	ADSSTR0	8	8	2 or 3 PCLKB	2 ICLK	section 44.
0008 90E1h	S12AD	A/D Sampling State Register 1	ADSSTR1	8	8	2 or 3 PCLKB	2 ICLK	section 44.
0008 90E2h	S12AD	A/D Sampling State Register 2	ADSSTR2	8	8	2 or 3 PCLKB	2 ICLK	section 44.
0008 90E3h	S12AD	A/D Sampling State Register 3	ADSSTR3	8	8	2 or 3 PCLKB	2 ICLK	section 44.
0008 90E4h	S12AD	A/D Sampling State Register 4	ADSSTR4	8	8	2 or 3 PCLKB	2 ICLK	section 44.
0008 90E5h	S12AD	A/D Sampling State Register 5	ADSSTR5	8	8	2 or 3 PCLKB	2 ICLK	section 44.
0008 90E6h	S12AD	A/D Sampling State Register 6	ADSSTR6	8	8	2 or 3 PCLKB	2 ICLK	section 44.
0008 90E7h	S12AD	A/D Sampling State Register 7	ADSSTR7	8	8	2 or 3 PCLKB	2 ICLK	section 44.

Downloaded from Arrow.com.

Table 5.1 List of I/O Registers (Address Order) (9/31)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK <PCLK	
0008 A020h	SCI1	Serial Mode Register	SMR	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 A020h	SMCI1	Serial Mode Register	SMR	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 A021h	SCI1	Bit Rate Register	BRR	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 A022h	SCI1	Serial Control Register	SCR	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 A022h	SMCI1	Serial Control Register	SCR	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 A023h	SCI1	Transmit Data Register	TDR	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 A024h	SCI1	Serial Status Register	SSR	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 A024h	SMCI1	Serial Status Register	SSR	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 A025h	SCI1	Receive Data Register	RDR	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 A026h	SCI1	Smart Card Mode Register	SCMR	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 A026h	SMCI1	Smart Card Mode Register	SCMR	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 A027h	SCI1	Serial Extended Mode Register	SEMR	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 A028h	SCI1	Noise Filter Setting Register	SNFR	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 A029h	SCI1	I^2C Mode Register 1	SIMR1	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 A02Ah	SCI1	I^2C Mode Register 2	SIMR2	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 A02Bh	SCI1	I^2C Mode Register 3	SIMR3	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 A02Ch	SCI1	I^2C Status Register	SISR	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 A02Dh	SCI1	SPI Mode Register	SPMR	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 A02Eh	SCI1	Transmit Data Register HL	TDRHL	16	16	4 or 5 PCLKB	2 ICLK	section 33.
0008 A02Eh	SCI1	Transmit Data Register H	TDRH	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 A02Fh	SCI1	Transmit Data Register L	TDRL	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 A030h	SCI1	Receive Data Register HL	RDRHL	16	16	4 or 5 PCLKB	2 ICLK	section 33.
0008 A030h	SCI1	Receive Data Register H	RDRH	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 A031h	SCI1	Receive Data Register L	RDRL	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 A032h	SCI1	Modulation Duty Register	MDDR	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 A0A0h	SCI5	Serial Mode Register	SMR	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 A0A0h	SMCI5	Serial Mode Register	SMR	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 A0A1h	SCI5	Bit Rate Register	BRR	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 A0A2h	SCI5	Serial Control Register	SCR	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 A0A2h	SMCI5	Serial Control Register	SCR	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 A0A3h	SCI5	Transmit Data Register	TDR	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 A0A4h	SCI5	Serial Status Register	SSR	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 A0A4h	SMCI5	Serial Status Register	SSR	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 A0A5h	SCI5	Receive Data Register	RDR	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 A0A6h	SCI5	Smart Card Mode Register	SCMR	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 A0A6h	SMCI5	Smart Card Mode Register	SCMR	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 A0A7h	SCI5	Serial Extended Mode Register	SEMR	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 A0A8h	SCI5	Noise Filter Setting Register	SNFR	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 A0A9h	SCI5	I^2C Mode Register 1	SIMR1	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 A0AAh	SCI5	I^2C Mode Register 2	SIMR2	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 A0ABh	SCI5	I^2C Mode Register 3	SIMR3	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 A0ACh	SCI5	I^2C Status Register	SISR	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 A0ADh	SCI5	SPI Mode Register	SPMR	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 A0AEh	SCI5	Transmit Data Register HL	TDRHL	16	16	4 or 5 PCLKB	2 ICLK	section 33.
0008 A0AEh	SCI5	Transmit Data Register H	TDRH	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 A0AFh	SCI5	Transmit Data Register L	TDRL	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 A0B0h	SCI5	Receive Data Register HL	RDRHL	16	16	4 or 5 PCLKB	2 ICLK	section 33.
0008 A0B0h	SCI5	Receive Data Register H	RDRH	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 A0B1h	SCI5	Receive Data Register L	RDRL	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 A0B2h	SCI5	Modulation Duty Register	MDDR	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 A100h	SCI8	Serial Mode Register	SMR	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 A100h	SMCI8	Serial Mode Register	SMR	8	8	2 or 3 PCLKB	2 ICLK	section 33.

Downloaded from Arrow.com.

Table 5.1 List of I/O Registers (Address Order) (10/31)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK <PCLK	
0008 A101h	SCI8	Bit Rate Register	BRR	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 A102h	SCI8	Serial Control Register	SCR	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 A102h	SMCI8	Serial Control Register	SCR	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 A103h	SCI8	Transmit Data Register	TDR	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 A104h	SCI8	Serial Status Register	SSR	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 A104h	SMCI8	Serial Status Register	SSR	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 A105h	SCI8	Receive Data Register	RDR	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 A106h	SCI8	Smart Card Mode Register	SCMR	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 A106h	SMCI8	Smart Card Mode Register	SCMR	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 A107h	SCI8	Serial Extended Mode Register	SEMR	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 A108h	SCI8	Noise Filter Setting Register	SNFR	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 A109h	SCI8	I^2C Mode Register 1	SIMR1	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 A10Ah	SCI8	I^2C Mode Register 2	SIMR2	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 A10Bh	SCI8	I^2C Mode Register 3	SIMR3	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 A10Ch	SCI8	I^2C Status Register	SISR	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 A10Dh	SCI8	SPI Mode Register	SPMR	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 A10Eh	SCI8	Transmit Data Register HL	TDRHL	16	16	4 or 5 PCLKB	2 ICLK	section 33.
0008 A10Eh	SCI8	Transmit Data Register H	TDRH	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 A10Fh	SCI8	Transmit Data Register L	TDRL	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 A110h	SCI8	Receive Data Register HL	RDRHL	16	16	4 or 5 PCLKB	2 ICLK	section 33.
0008 A110h	SCI8	Receive Data Register H	RDRH	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 A111h	SCI8	Receive Data Register L	RDRL	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 A112h	SCI8	Modulation Duty Register	MDDR	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 A500h	SSI0	Control Register	SSICR	32	32	2 or 3 PCLKB	2 ICLK	section 37.
0008 A504h	SSI0	Status Register	SSISR	32	32	2 or 3 PCLKB	2 ICLK	section 37.
0008 A510h	SSI0	FIFO Control Register	SSIFCR	32	32	2 or 3 PCLKB	2 ICLK	section 37.
0008 A514h	SSI0	FIFO Status Register	SSIFSR	32	32	2 or 3 PCLKB	2 ICLK	section 37.
0008 A518h	SSI0	Transmit FIFO Data Register	SSIFTDR	32	32	2 or 3 PCLKB	2 ICLK	section 37.
0008 A51Ch	SSI0	Receive FIFO Data Register	SSIFRDR	32	32	2 or 3 PCLKB	2 ICLK	section 37.
0008 A520h	SSI0	TDM Mode Register	SSITDMR	32	32	2 or 3 PCLKB	2 ICLK	section 37.
0008 AC00h	SDHI	Command Register	SDCMD	32	32	3 or 4 PCLKB cycles when reading, 2 or 3 PCLKB cycles when writing	3 ICLK cycles when reading, 2 ICLK cycles when writing	section 40.
0008 AC08h	SDHI	Argument Register	SDARG	32	32	3 or 4 PCLKB cycles when reading, 2 or 3 PCLKB cycles when writing	3 ICLK cycles when reading, 2 ICLK cycles when writing	section 40.
0008 AC10h	SDHI	Data Stop Register	SDSTOP	32	32	3 or 4 PCLKB cycles when reading, 2 or 3 PCLKB cycles when writing	3 ICLK cycles when reading, 2 ICLK cycles when writing	section 40.
0008 AC14h	SDHI	Block Count Register	SDBLKCNT	32	32	3 or 4 PCLKB cycles when reading, 2 or 3 PCLKB cycles when writing	3 ICLK cycles when reading, 2 ICLK cycles when writing	section 40.
0008 AC18h	SDHI	Response Register 10	SDRSP10	32	32	3 or 4 PCLKB cycles when reading, 2 or 3 PCLKB cycles when writing	3 ICLK cycles when reading, 2 ICLK cycles when writing	section 40.

Downloaded from Arrow.com.

Table 5.1 List of I/O Registers (Address Order) (11/31)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK <PCLK	
0008 AC20h	SDHI	Response Register 32	SDRSP32	32	32	3 or 4 PCLKB cycles when reading, 2 or 3 PCLKB cycles when writing	3 ICLK cycles when reading, 2 ICLK cycles when writing	section 40.
0008 AC28h	SDHI	Response Register 54	SDRSP54	32	32	3 or 4 PCLKB cycles when reading, 2 or 3 PCLKB cycles when writing	3 ICLK cycles when reading, 2 ICLK cycles when writing	section 40.
0008 AC30h	SDHI	Response Register 76	SDRSP76	32	32	3 or 4 PCLKB cycles when reading, 2 or 3 PCLKB cycles when writing	3 ICLK cycles when reading, 2 ICLK cycles when writing	section 40.
0008 AC38h	SDHI	SD Status Register 1	SDSTS1	32	32	3 or 4 PCLKB cycles when reading, 2 or 3 PCLKB cycles when writing	3 ICLK cycles when reading, 2 ICLK cycles when writing	section 40.
0008 AC3Ch	SDHI	SD Status Register 2	SDSTS2	32	32	3 or 4 PCLKB cycles when reading, 2 or 3 PCLKB cycles when writing	3 ICLK cycles when reading, 2 ICLK cycles when writing	section 40.
0008 AC40h	SDHI	SD Interrupt Mask Register 1	SDIMSK1	32	32	3 or 4 PCLKB cycles when reading, 2 or 3 PCLKB cycles when writing	3 ICLK cycles when reading, 2 ICLK cycles when writing	section 40.
0008 AC44h	SDHI	SD Interrupt Mask Register 2	SDIMSK2	32	32	3 or 4 PCLKB cycles when reading, 2 or 3 PCLKB cycles when writing	3 ICLK cycles when reading, 2 ICLK cycles when writing	section 40.
0008 AC48h	SDHI	SDHI Clock Control Register	SDCLKCR	32	32	3 or 4 PCLKB cycles when reading, 2 or 3 PCLKB cycles when writing	3 ICLK cycles when reading, 2 ICLK cycles when writing	section 40.
0008 AC4Ch	SDHI	Transfer Data Size Register	SDSIZE	32	32	3 or 4 PCLKB cycles when reading, 2 or 3 PCLKB cycles when writing	3 ICLK cycles when reading, 2 ICLK cycles when writing	section 40.
0008 AC50h	SDHI	Card Access Option Register	SDOPT	32	32	3 or 4 PCLKB cycles when reading, 2 or 3 PCLKB cycles when writing	3 ICLK cycles when reading, 2 ICLK cycles when writing	section 40.
0008 AC58h	SDHI	SD Error Status Register 1	SDERSTS1	32	32	3 or 4 PCLKB cycles when reading, 2 or 3 PCLKB cycles when writing	3 ICLK cycles when reading, 2 ICLK cycles when writing	section 40.
0008 AC5Ch	SDHI	SD Error Status Register 2	SDERSTS2	32	32	3 or 4 PCLKB cycles when reading, 2 or 3 PCLKB cycles when writing	3 ICLK cycles when reading, 2 ICLK cycles when writing	section 40.
0008 AC60h	SDHI	SD Buffer Register	SDBUFR	32	32	3 or 4 PCLKB cycles when reading, 2 or 3 PCLKB cycles when writing	3 ICLK cycles when reading, 2 ICLK cycles when writing	section 40.

Downloaded from Arrow.com.

Table 5.1 List of I/O Registers (Address Order) (12/31)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK <PCLK	
0008 AC68h	SDHI	SDIO Mode Control Register	SDIOMD	32	32	3 or 4 PCLKB cycles when reading, 2 or 3 PCLKB cycles when writing	3 ICLK cycles when reading, 2 ICLK cycles when writing	section 40.
0008 AC6Ch	SDHI	SDIO Status Register	SDIOSTS	32	32	3 or 4 PCLKB cycles when reading, 2 or 3 PCLKB cycles when writing	3 ICLK cycles when reading, 2 ICLK cycles when writing	section 40.
0008 AC70h	SDHI	SDIO Interrupt Mask Register	SDIOIMSK	32	32	3 or 4 PCLKB cycles when reading, 2 or 3 PCLKB cycles when writing	3 ICLK cycles when reading, 2 ICLK cycles when writing	section 40.
0008 ADB0h	SDHI	DMA Transfer Enable Register	SDDMAEN	32	32	3 or 4 PCLKB cycles when reading, 2 or 3 PCLKB cycles when writing	3 ICLK cycles when reading, 2 ICLK cycles when writing	section 40.
0008 ADC0h	SDHI	SDHI Software Reset Register	SDRST	32	32	3 or 4 PCLKB cycles when reading, 2 or 3 PCLKB cycles when writing	3 ICLK cycles when reading, 2 ICLK cycles when writing	section 40.
0008 ADE0h	SDHI	Swap Control Register	SDSWAP	32	32	3 or 4 PCLKB cycles when reading, 2 or 3 PCLKB cycles when writing	3 ICLK cycles when reading, 2 ICLK cycles when writing	section 40.
0008 B000h	CAC	CAC Control Register 0	CACR0	8	8	2 or 3 PCLKB	2 ICLK	section 10.
0008 B001h	CAC	CAC Control Register 1	CACR1	8	8	2 or 3 PCLKB	2 ICLK	section 10.
0008 B002h	CAC	CAC Control Register 2	CACR2	8	8	2 or 3 PCLKB	2 ICLK	section 10.
0008 B003h	CAC	CAC Interrupt Request Enable Register	CAICR	8	8	2 or 3 PCLKB	2 ICLK	section 10.
0008 B004h	CAC	CAC Status Register	CASTR	8	8	2 or 3 PCLKB	2 ICLK	section 10.
0008 B006h	CAC	CAC Upper-Limit Value Setting Register	CAULVR	16	16	2 or 3 PCLKB	2 ICLK	section 10.
0008 B008h	CAC	CAC Lower-Limit Value Setting Register	CALLVR	16	16	2 or 3 PCLKB	2 ICLK	section 10.
0008 B00Ah	CAC	CAC Counter Buffer Register	CACNTBR	16	16	2 or 3 PCLKB	2 ICLK	section 10.
0008 B080h	DOC	DOC Control Register	DOCR	8	8	2 or 3 PCLKB	2 ICLK	section 48.
0008 B082h	DOC	DOC Data Input Register	DODIR	16	16	2 or 3 PCLKB	2 ICLK	section 48.
0008 B084h	DOC	DOC Data Setting Register	DODSR	16	16	2 or 3 PCLKB	2 ICLK	section 48.
0008 B100h	ELC	Event Link Control Register	ELCR	8	8	2 or 3 PCLKB	2 ICLK	section 20.
0008 B102h	ELC	Event Link Setting Register 1	ELSR1	8	8	2 or 3 PCLKB	2 ICLK	section 20.
0008 B103h	ELC	Event Link Setting Register 2	ELSR2	8	8	2 or 3 PCLKB	2 ICLK	section 20.
0008 B104h	ELC	Event Link Setting Register 3	ELSR3	8	8	2 or 3 PCLKB	2 ICLK	section 20.
0008 B105h	ELC	Event Link Setting Register 4	ELSR4	8	8	2 or 3 PCLKB	2 ICLK	section 20.
0008 B108h	ELC	Event Link Setting Register 7	ELSR7	8	8	2 or 3 PCLKB	2 ICLK	section 20.
0008 B109h	ELC	Event Link Setting Register 8	ELSR8	8	8	2 or 3 PCLKB	2 ICLK	section 20.
0008 B10Bh	ELC	Event Link Setting Register 10	ELSR10	8	8	2 or 3 PCLKB	2 ICLK	section 20.
0008 B10Dh	ELC	Event Link Setting Register 12	ELSR12	8	8	2 or 3 PCLKB	2 ICLK	section 20.
0008 B10Fh	ELC	Event Link Setting Register 14	ELSR14	8	8	2 or 3 PCLKB	2 ICLK	section 20.
0008 B110h	ELC	Event Link Setting Register 15	ELSR15	8	8	2 or 3 PCLKB	2 ICLK	section 20.
0008 B111h	ELC	Event Link Setting Register 16	ELSR16	8	8	2 or 3 PCLKB	2 ICLK	section 20.
0008 B113h	ELC	Event Link Setting Register 18	ELSR18	8	8	2 or 3 PCLKB	2 ICLK	section 20.
0008 B114h	ELC	Event Link Setting Register 19	ELSR19	8	8	2 or 3 PCLKB	2 ICLK	section 20.
0008 B115h	ELC	Event Link Setting Register 20	ELSR20	8	8	2 or 3 PCLKB	2 ICLK	section 20.
0008 B116h	ELC	Event Link Setting Register 21	ELSR21	8	8	2 or 3 PCLKB	2 ICLK	section 20.
0008 B117h	ELC	Event Link Setting Register 22	ELSR22	8	8	2 or 3 PCLKB	2 ICLK	section 20.
0008 B118h	ELC	Event Link Setting Register 23	ELSR23	8	8	2 or 3 PCLKB	2 ICLK	section 20.

Downloaded from Arrow.com.

Table 5.1 List of I/O Registers (Address Order) (13/31)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK <PCLK	
0008 B119h	ELC	Event Link Setting Register 24	ELSR24	8	8	2 or 3 PCLKB	2 ICLK	section 20.
0008 B11Ah	ELC	Event Link Setting Register 25	ELSR25	8	8	2 or 3 PCLKB	2 ICLK	section 20.
0008 B11Bh	ELC	Event Link Setting Register 26	ELSR26	8	8	2 or 3 PCLKB	2 ICLK	section 20.
0008 B11Ch	ELC	Event Link Setting Register 27	ELSR27	8	8	2 or 3 PCLKB	2 ICLK	section 20.
0008 B11Dh	ELC	Event Link Setting Register 28	ELSR28	8	8	2 or 3 PCLKB	2 ICLK	section 20.
0008 B11Eh	ELC	Event Link Setting Register 29	ELSR29	8	8	2 or 3 PCLKB	2 ICLK	section 20.
0008 B11Fh	ELC	Event Link Option Setting Register A	ELOPA	8	8	2 or 3 PCLKB	2 ICLK	section 20.
0008 B120h	ELC	Event Link Option Setting Register B	ELOPB	8	8	2 or 3 PCLKB	2 ICLK	section 20.
0008 B121h	ELC	Event Link Option Setting Register C	ELOPC	8	8	2 or 3 PCLKB	2 ICLK	section 20.
0008 B122h	ELC	Event Link Option Setting Register D	ELOPD	8	8	2 or 3 PCLKB	2 ICLK	section 20.
0008 B123h	ELC	Port Group Setting Register 1	PGR1	8	8	2 or 3 PCLKB	2 ICLK	section 20.
0008 B124h	ELC	Port Group Setting Register 2	PGR2	8	8	2 or 3 PCLKB	2 ICLK	section 20.
0008 B125h	ELC	Port Group Control Register 1	PGC1	8	8	2 or 3 PCLKB	2 ICLK	section 20.
0008 B126h	ELC	Port Group Control Register 2	PGC2	8	8	2 or 3 PCLKB	2 ICLK	section 20.
0008 B127h	ELC	Port Buffer Register 1	PDBF1	8	8	2 or 3 PCLKB	2 ICLK	section 20.
0008 B128h	ELC	Port Buffer Register 2	PDBF2	8	8	2 or 3 PCLKB	2 ICLK	section 20.
0008 B129h	ELC	Event Link Port Setting Register 0	PEL0	8	8	2 or 3 PCLKB	2 ICLK	section 20.
0008 B12Ah	ELC	Event Link Port Setting Register 1	PEL1	8	8	2 or 3 PCLKB	2 ICLK	section 20.
0008 B12Bh	ELC	Event Link Port Setting Register 2	PEL2	8	8	2 or 3 PCLKB	2 ICLK	section 20.
0008 B12Ch	ELC	Event Link Port Setting Register 3	PEL3	8	8	2 or 3 PCLKB	2 ICLK	section 20.
0008 B12Dh	ELC	Event Link Software Event Generation Register	ELSEGR	8	8	2 or 3 PCLKB	2 ICLK	section 20.
0008 B300h	SCI12	Serial Mode Register	SMR	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 B300h	SMCI12	Serial Mode Register	SMR	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 B301h	SCI12	Bit Rate Register	BRR	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 B302h	SCI12	Serial Control Register	SCR	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 B302h	SMCI12	Serial Control Register	SCR	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 B303h	SCI12	Transmit Data Register	TDR	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 B304h	SCI12	Serial Status Register	SSR	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 B304h	SMCI12	Serial Status Register	SSR	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 B305h	SCI12	Receive Data Register	RDR	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 B306h	SCI12	Smart Card Mode Register	SCMR	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 B306h	SMCI12	Smart Card Mode Register	SCMR	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 B307h	SCI12	Serial Extended Mode Register	SEMR	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 B308h	SCI12	Noise Filter Setting Register	SNFR	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 B309h	SCI12	I^2C Mode Register 1	SIMR1	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 B30Ah	SCI12	I^2C Mode Register 2	SIMR2	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 B30Bh	SCI12	I^2C Mode Register 3	SIMR3	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 B30Ch	SCI12	I^2C Status Register	SISR	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 B30Dh	SCI12	SPI Mode Register	SPMR	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 B30Eh	SCI12	Transmit Data Register HL	TDRHL	16	16	4 or 5 PCLKB	2 ICLK	section 33.
0008 B30Eh	SCI12	Transmit Data Register H	TDRH	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 B30Fh	SCI12	Transmit Data Register L	TDRL	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 B310h	SCI12	Receive Data Register HL	RDRHL	16	16	4 or 5 PCLKB	2 ICLK	section 33.
0008 B310h	SCI12	Receive Data Register H	RDRH	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 B311h	SCI12	Receive Data Register L	RDRL	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 B312h	SCI12	Modulation Duty Register	MDDR	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 B320h	SCI12	Extended Serial Module Enable Register	ESMER	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 B321h	SCI12	Control Register 0	CR0	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 B322h	SCI12	Control Register 1	CR1	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 B323h	SCI12	Control Register 2	CR2	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 B324h	SCI12	Control Register 3	CR3	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 B325h	SCI12	Port Control Register	PCR	8	8	2 or 3 PCLKB	2 ICLK	section 33.

Downloaded from Arrow.com.

Table 5.1 List of I/O Registers (Address Order) (14/31)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK <PCLK	
0008 B326h	SCI12	Interrupt Control Register	ICR	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 B327h	SCI12	Status Register	STR	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 B328h	SCI12	Status Clear Register	STCR	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 B329h	SCI12	Control Field 0 Data Register	CF0DR	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 B32Ah	SCI12	Control Field 0 Compare Enable Register	CF0CR	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 B32Bh	SCI12	Control Field 0 Receive Data Register	CF0RR	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 B32Ch	SCI12	Primary Control Field 1 Data Register	PCF1DR	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 B32Dh	SCI12	Secondary Control Field 1 Data Register	SCF1DR	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 B32Eh	SCI12	Control Field 1 Compare Enable Register	CF1CR	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 B32Fh	SCI12	Control Field 1 Receive Data Register	CF1RR	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 B330h	SCI12	Timer Control Register	TCR	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 B331h	SCI12	Timer Mode Register	TMR	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 B332h	SCI12	Timer Prescaler Register	TPRE	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 B333h	SCI12	Timer Count Register	TCNT	8	8	2 or 3 PCLKB	2 ICLK	section 33.
0008 C000h	PORT0	Port Direction Register	PDR	8	8	2 or 3 PCLKB	2 ICLK	section 21.
0008 C001h	PORT1	Port Direction Register	PDR	8	8	2 or 3 PCLKB	2 ICLK	section 21.
0008 C002h	PORT2	Port Direction Register	PDR	8	8	2 or 3 PCLKB	2 ICLK	section 21.
0008 C003h	PORT3	Port Direction Register	PDR	8	8	2 or 3 PCLKB	2 ICLK	section 21.
0008 C004h	PORT4	Port Direction Register	PDR	8	8	2 or 3 PCLKB	2 ICLK	section 21.
0008 C00Bh	PORTB	Port Direction Register	PDR	8	8	2 or 3 PCLKB	2 ICLK	section 21.
0008 C00Ch	PORTC	Port Direction Register	PDR	8	8	2 or 3 PCLKB	2 ICLK	section 21.
0008 C00Dh	PORTD	Port Direction Register	PDR	8	8	2 or 3 PCLKB	2 ICLK	section 21.
0008 C00Eh	PORTE	Port Direction Register	PDR	8	8	2 or 3 PCLKB	2 ICLK	section 21.
0008 C012h	PORTJ	Port Direction Register	PDR	8	8	2 or 3 PCLKB	2 ICLK	section 21.
0008 C020h	PORT0	Port Output Data Register	PODR	8	8	2 or 3 PCLKB	2 ICLK	section 21.
0008 C021h	PORT1	Port Output Data Register	PODR	8	8	2 or 3 PCLKB	2 ICLK	section 21.
0008 C022h	PORT2	Port Output Data Register	PODR	8	8	2 or 3 PCLKB	2 ICLK	section 21.
0008 C023h	PORT3	Port Output Data Register	PODR	8	8	2 or 3 PCLKB	2 ICLK	section 21.
0008 C024h	PORT4	Port Output Data Register	PODR	8	8	2 or 3 PCLKB	2 ICLK	section 21.
0008 C02Bh	PORTB	Port Output Data Register	PODR	8	8	2 or 3 PCLKB	2 ICLK	section 21.
0008 C02Ch	PORTC	Port Output Data Register	PODR	8	8	2 or 3 PCLKB	2 ICLK	section 21.
0008 C02Dh	PORTD	Port Output Data Register	PODR	8	8	2 or 3 PCLKB	2 ICLK	section 21.
0008 C02Eh	PORTE	Port Output Data Register	PODR	8	8	2 or 3 PCLKB	2 ICLK	section 21.
0008 C032h	PORTJ	Port Output Data Register	PODR	8	8	2 or 3 PCLKB	2 ICLK	section 21.
0008 C040h	PORT0	Port Input Data Register	PIDR	8	8	3 or 4 PCLKB	3 ICLK	section 21.
0008 C041h	PORT1	Port Input Data Register	PIDR	8	8	3 or 4 PCLKB	3 ICLK	section 21.
0008 C042h	PORT2	Port Input Data Register	PIDR	8	8	3 or 4 PCLKB	3 ICLK	section 21.
0008 C043h	PORT3	Port Input Data Register	PIDR	8	8	3 or 4 PCLKB	3 ICLK	section 21.
0008 C044h	PORT4	Port Input Data Register	PIDR	8	8	3 or 4 PCLKB	3 ICLK	section 21.
0008 C04Bh	PORTB	Port Input Data Register	PIDR	8	8	3 or 4 PCLKB	3 ICLK	section 21.
0008 C04Ch	PORTC	Port Input Data Register	PIDR	8	8	3 or 4 PCLKB	3 ICLK	section 21.
0008 C04Dh	PORTD	Port Input Data Register	PIDR	8	8	3 or 4 PCLKB	3 ICLK	section 21.
0008 C04Eh	PORTE	Port Input Data Register	PIDR	8	8	3 or 4 PCLKB	3 ICLK	section 21.
0008 C052h	PORTJ	Port Input Data Register	PIDR	8	8	3 or 4 PCLKB	3 ICLK	section 21.
0008 C060h	PORT0	Port Mode Register	PMR	8	8	2 or 3 PCLKB	2 ICLK	section 21.
0008 C061h	PORT1	Port Mode Register	PMR	8	8	2 or 3 PCLKB	2 ICLK	section 21.
0008 C062h	PORT2	Port Mode Register	PMR	8	8	2 or 3 PCLKB	2 ICLK	section 21.
0008 C063h	PORT3	Port Mode Register	PMR	8	8	2 or 3 PCLKB	2 ICLK	section 21.
0008 C064h	PORT4	Port Mode Register	PMR	8	8	2 or 3 PCLKB	2 ICLK	section 21.
0008 C06Bh	PORTB	Port Mode Register	PMR	8	8	2 or 3 PCLKB	2 ICLK	section 21.
0008 C06Ch	PORTC	Port Mode Register	PMR	8	8	2 or 3 PCLKB	2 ICLK	section 21.
0008 C06Dh	PORTD	Port Mode Register	PMR	8	8	2 or 3 PCLKB	2 ICLK	section 21.

Downloaded from Arrow.com.

Table 5.1 List of I/O Registers (Address Order) (15/31)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK <PCLK	
0008 C06Eh	PORTE	Port Mode Register	PMR	8	8	2 or 3 PCLKB	2 ICLK	section 21.
0008 C072h	PORTJ	Port Mode Register	PMR	8	8	2 or 3 PCLKB	2 ICLK	section 21.
0008 C083h	PORT1	Open Drain Control Register 1	ODR1	8	8, 16	2 or 3 PCLKB	2 ICLK	section 21.
0008 C084h	PORT2	Open Drain Control Register 0	ODR0	8	8, 16	2 or 3 PCLKB	2 ICLK	section 21.
0008 C085h	PORT2	Open Drain Control Register 1	ODR1	8	8, 16	2 or 3 PCLKB	2 ICLK	section 21.
0008 C086h	PORT3	Open Drain Control Register 0	ODR0	8	8, 16	2 or 3 PCLKB	2 ICLK	section 21.
0008 C087h	PORT3	Open Drain Control Register 1	ODR1	8	8, 16	2 or 3 PCLKB	2 ICLK	section 21.
0008 C096h	PORTB	Open Drain Control Register 0	ODR0	8	8, 16	2 or 3 PCLKB	2 ICLK	section 21.
0008 C097h	PORTB	Open Drain Control Register 1	ODR1	8	8, 16	2 or 3 PCLKB	2 ICLK	section 21.
0008 C098h	PORTC	Open Drain Control Register 0	ODR0	8	8, 16	2 or 3 PCLKB	2 ICLK	section 21.
0008 C099h	PORTC	Open Drain Control Register 1	ODR1	8	8, 16	2 or 3 PCLKB	2 ICLK	section 21.
0008 C09Ch	PORTE	Open Drain Control Register 0	ODR0	8	8, 16	2 or 3 PCLKB	2 ICLK	section 21.
0008 C09Dh	PORTE	Open Drain Control Register 1	ODR1	8	8, 16	2 or 3 PCLKB	2 ICLK	section 21.
0008 C0A4h	PORTJ	Open Drain Control Register 0	ODR0	8	8, 16	2 or 3 PCLKB	2 ICLK	section 21.
0008 C0C0h	PORT0	Pull-Up Control Register	PCR	8	8	2 or 3 PCLKB	2 ICLK	section 21.
0008 C0C1h	PORT1	Pull-Up Control Register	PCR	8	8	2 or 3 PCLKB	2 ICLK	section 21.
0008 C0C2h	PORT2	Pull-Up Control Register	PCR	8	8	2 or 3 PCLKB	2 ICLK	section 21.
0008 C0C3h	PORT3	Pull-Up Control Register	PCR	8	8	2 or 3 PCLKB	2 ICLK	section 21.
0008 C0C4h	PORT4	Pull-Up Control Register	PCR	8	8	2 or 3 PCLKB	2 ICLK	section 21.
0008 C0CBh	PORTB	Pull-Up Control Register	PCR	8	8	2 or 3 PCLKB	2 ICLK	section 21.
0008 C0CCh	PORTC	Pull-Up Control Register	PCR	8	8	2 or 3 PCLKB	2 ICLK	section 21.
0008 C0CDh	PORTD	Pull-Up Control Register	PCR	8	8	2 or 3 PCLKB	2 ICLK	section 21.
0008 C0CEh	PORTE	Pull-Up Control Register	PCR	8	8	2 or 3 PCLKB	2 ICLK	section 21.
0008 C0D2h	PORTJ	Pull-Up Control Register	PCR	8	8	2 or 3 PCLKB	2 ICLK	section 21.
0008 C0E1h	PORT1	Drive Capacity Control Register	DSCR	8	8	2 or 3 PCLKB	2 ICLK	section 21.
0008 C0E2h	PORT2	Drive Capacity Control Register	DSCR	8	8	2 or 3 PCLKB	2 ICLK	section 21.
0008 C0E3h	PORT3	Drive Capacity Control Register	DSCR	8	8	2 or 3 PCLKB	2 ICLK	section 21.
0008 C0EBh	PORTB	Drive Capacity Control Register	DSCR	8	8	2 or 3 PCLKB	2 ICLK	section 21.
0008 C0ECh	PORTC	Drive Capacity Control Register	DSCR	8	8	2 or 3 PCLKB	2 ICLK	section 21.
0008 C0EDh	PORTD	Drive Capacity Control Register	DSCR	8	8	2 or 3 PCLKB	2 ICLK	section 21.
0008 C0EEh	PORTE	Drive Capacity Control Register	DSCR	8	8	2 or 3 PCLKB	2 ICLK	section 21.
0008 C0F2h	PORTJ	Drive Capacity Control Register	DSCR	8	8	2 or 3 PCLKB	2 ICLK	section 21.
0008 C11Fh	MPC	Write-Protect Register	PWPR	8	8	2 or 3 PCLKB	2 ICLK	section 22.
0008 C143h	MPC	P03 Pin Function Control Register	P03PFS	8	8	2 or 3 PCLKB	2 ICLK	section 22.
0008 C145h	MPC	P05 Pin Function Control Register	P05PFS	8	8	2 or 3 PCLKB	2 ICLK	section 22.
0008 C147h	MPC	P07 Pin Function Control Register	P07PFS	8	8	2 or 3 PCLKB	2 ICLK	section 22.
0008 C14Ch	MPC	P14 Pin Function Control Register	P14PFS	8	8	2 or 3 PCLKB	2 ICLK	section 22.
0008 C14Dh	MPC	P15 Pin Function Control Register	P15PFS	8	8	2 or 3 PCLKB	2 ICLK	section 22.
0008 C14Eh	MPC	P16 Pin Function Control Register	P16PFS	8	8	2 or 3 PCLKB	2 ICLK	section 22.
0008 C14Fh	MPC	P17 Pin Function Control Register	P17PFS	8	8	2 or 3 PCLKB	2 ICLK	section 22.
0008 C151h	MPC	P21 Pin Function Control Register	P21PFS	8	8	2 or 3 PCLKB	2 ICLK	section 22.
0008 C152h	MPC	P22 Pin Function Control Register	P22PFS	8	8	2 or 3 PCLKB	2 ICLK	section 22.
0008 C155h	MPC	P25 Pin Function Control Register	P25PFS	8	8	2 or 3 PCLKB	2 ICLK	section 22.
0008 C156h	MPC	P26 Pin Function Control Register	P26PFS	8	8	2 or 3 PCLKB	2 ICLK	section 22.
0008 C157h	MPC	P27 Pin Function Control Register	P27PFS	8	8	2 or 3 PCLKB	2 ICLK	section 22.
0008 C158h	MPC	P30 Pin Function Control Register	P30PFS	8	8	2 or 3 PCLKB	2 ICLK	section 22.
0008 C159h	MPC	P31 Pin Function Control Register	P31PFS	8	8	2 or 3 PCLKB	2 ICLK	section 22.
0008 C160h	MPC	P40 Pin Function Control Register	P40PFS	8	8	2 or 3 PCLKB	2 ICLK	section 22.
0008 C161h	MPC	P41 Pin Function Control Register	P41PFS	8	8	2 or 3 PCLKB	2 ICLK	section 22.
0008 C162h	MPC	P42 Pin Function Control Register	P42PFS	8	8	2 or 3 PCLKB	2 ICLK	section 22.
0008 C163h	MPC	P43 Pin Function Control Register	P43PFS	8	8	2 or 3 PCLKB	2 ICLK	section 22.
0008 C164h	MPC	P44 Pin Function Control Register	P44PFS	8	8	2 or 3 PCLKB	2 ICLK	section 22.

Downloaded from Arrow.com.

Table 5.1 List of I/O Registers (Address Order) (16/31)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK <PCLK	
0008 C165h	MPC	P45 Pin Function Control Register	P45PFS	8	8	2 or 3 PCLKB	2 ICLK	section 22.
0008 C166h	MPC	P46 Pin Function Control Register	P46PFS	8	8	2 or 3 PCLKB	2 ICLK	section 22.
0008 C167h	MPC	P47 Pin Function Control Register	P47PFS	8	8	2 or 3 PCLKB	2 ICLK	section 22.
0008 C198h	MPC	PB0 Pin Function Control Register	PB0PFS	8	8	2 or 3 PCLKB	2 ICLK	section 22.
0008 C199h	MPC	PB1 Pin Function Control Register	PB1PFS	8	8	2 or 3 PCLKB	2 ICLK	section 22.
0008 C19Bh	MPC	PB3 Pin Function Control Register	PB3PFS	8	8	2 or 3 PCLKB	2 ICLK	section 22.
0008 C19Dh	MPC	PB5 Pin Function Control Register	PB5PFS	8	8	2 or 3 PCLKB	2 ICLK	section 22.
0008 C19Fh	MPC	PB7 Pin Function Control Register	PB7PFS	8	8	2 or 3 PCLKB	2 ICLK	section 22.
0008 C1A0h	MPC	PC0 Pin Function Control Register	PC0PFS	8	8	2 or 3 PCLKB	2 ICLK	section 22.
0008 C1A2h	MPC	PC2 Pin Function Control Register	PC2PFS	8	8	2 or 3 PCLKB	2 ICLK	section 22.
0008 C1A3h	MPC	PC3 Pin Function Control Register	PC3PFS	8	8	2 or 3 PCLKB	2 ICLK	section 22.
0008 C1A4h	MPC	PC4 Pin Function Control Register	PC4PFS	8	8	2 or 3 PCLKB	2 ICLK	section 22.
0008 C1A5h	MPC	PC5 Pin Function Control Register	PC5PFS	8	8	2 or 3 PCLKB	2 ICLK	section 22.
0008 C1A6h	MPC	PC6 Pin Function Control Register	PC6PFS	8	8	2 or 3 PCLKB	2 ICLK	section 22.
0008 C1A7h	MPC	PC7 Pin Function Control Register	PC7PFS	8	8	2 or 3 PCLKB	2 ICLK	section 22.
0008 C1ABh	MPC	PD3 Pin Function Control Register	PD3PFS	8	8	2 or 3 PCLKB	2 ICLK	section 22.
0008 C1B0h	MPC	PE0 Pin Function Control Register	PE0PFS	8	8	2 or 3 PCLKB	2 ICLK	section 22.
0008 C1B1h	MPC	PE1 Pin Function Control Register	PE1PFS	8	8	2 or 3 PCLKB	2 ICLK	section 22.
0008 C1B2h	MPC	PE2 Pin Function Control Register	PE2PFS	8	8	2 or 3 PCLKB	2 ICLK	section 22.
0008 C1B3h	MPC	PE3 Pin Function Control Register	PE3PFS	8	8	2 or 3 PCLKB	2 ICLK	section 22.
0008 C1B4h	MPC	PE4 Pin Function Control Register	PE4PFS	8	8	2 or 3 PCLKB	2 ICLK	section 22.
0008 C1D3h	MPC	PJ3 Pin Function Control Register	PJ3PFS	8	8	2 or 3 PCLKB	2 ICLK	section 22.
0008 C290h	SYSTEM	Reset Status Register 0	RSTSR0	8	8	4 or 5 PCLKB	2 or 3 ICLK	section 6.
0008 C291h	SYSTEM	Reset Status Register 1	RSTSR1	8	8	4 or 5 PCLKB	2 or 3 ICLK	section 6.
0008 C293h	SYSTEM	Main Clock Oscillator Forced Oscillation Control Register	MOFCR	8	8	4 or 5 PCLKB	2 or 3 ICLK	section 9.
0008 C297h	SYSTEM	Voltage Monitoring Circuit Control Register	LVCMPCR	8	8	4 or 5 PCLKB	2 or 3 ICLK	section 8.
0008 C298h	SYSTEM	Voltage Detection Level Select Register	LVDLVLR	8	8	4 or 5 PCLKB	2 or 3 ICLK	section 8.
0008 C29Ah	SYSTEM	Voltage Monitoring 1 Circuit Control Register 0	LVD1CR0	8	8	4 or 5 PCLKB	2 or 3 ICLK	section 8.
0008 C29Dh	SYSTEM	VBATT Control Register	VBATTCR	8	8	4 or 5 PCLKB	2 or 3 ICLK	section 12.
0008 C29Eh	SYSTEM	VBATT Status Register	VBATTSR	8	8	4 or 5 PCLKB	2 or 3 ICLK	section 12.
0008 C29Fh	SYSTEM	VBATT Pin Voltage Drop Detection Interrupt Control Register	VBTLVDICR	8	8	4 or 5 PCLKB	2 or 3 ICLK	section 12.
0008 C400h	RTC	64-Hz Counter	R64CNT	8	8	2 or 3 PCLKB	2 ICLK	section 28.
0008 C402h	RTC	Second Counter	RSECCNT	8	8	2 or 3 PCLKB	2 ICLK	section 28.
0008 C402h	RTC	Binary Counter 0	BCNT0	8	8	2 or 3 PCLKB	2 ICLK	section 28.
0008 C404h	RTC	Minute Counter	RMINCNT	8	8	2 or 3 PCLKB	2 ICLK	section 28.
0008 C404h	RTC	Binary Counter 1	BCNT1	8	8	2 or 3 PCLKB	2 ICLK	section 28.
0008 C406h	RTC	Hour Counter	RHRCNT	8	8	2 or 3 PCLKB	2 ICLK	section 28.
0008 C406h	RTC	Binary Counter 2	BCNT2	8	8	2 or 3 PCLKB	2 ICLK	section 28.
0008 C408h	RTC	Day-of-Week Counter	RWKCNT	8	8	2 or 3 PCLKB	2 ICLK	section 28.
0008 C408h	RTC	Binary Counter 3	BCNT3	8	8	2 or 3 PCLKB	2 ICLK	section 28.
0008 C40Ah	RTC	Date Counter	RDAYCNT	8	8	2 or 3 PCLKB	2 ICLK	section 28.
0008 C40Ch	RTC	Month Counter	RMONCNT	8	8	2 or 3 PCLKB	2 ICLK	section 28.
0008 C40Eh	RTC	Year Counter	RYRCNT	16	16	2 or 3 PCLKB	2 ICLK	section 28.
0008 C410h	RTC	Second Alarm Register	RSECAR	8	8	2 or 3 PCLKB	2 ICLK	section 28.
0008 C410h	RTC	Binary Counter 0 Alarm Register	BCNT0AR	8	8	2 or 3 PCLKB	2 ICLK	section 28.
0008 C412h	RTC	Minute Alarm Register	RMINAR	8	8	2 or 3 PCLKB	2 ICLK	section 28.
0008 C412h	RTC	Binary Counter 1 Alarm Register	BCNT1AR	8	8	2 or 3 PCLKB	2 ICLK	section 28.
0008 C414h	RTC	Hour Alarm Register	RHRAR	8	8	2 or 3 PCLKB	2 ICLK	section 28.
0008 C414h	RTC	Binary Counter 2 Alarm Register	BCNT2AR	8	8	2 or 3 PCLKB	2 ICLK	section 28.
0008 C416h	RTC	Day-of-Week Alarm Register	RWKAR	8	8	2 or 3 PCLKB	2 ICLK	section 28.
0008 C416h	RTC	Binary Counter 3 Alarm Register	BCNT3AR	8	8	2 or 3 PCLKB	2 ICLK	section 28.

Downloaded from Arrow.com.

Table 5.1 List of I/O Registers (Address Order) (17/31)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK <PCLK	
0008 C418h	RTC	Date Alarm Register	RDAYAR	8	8	2 or 3 PCLKB	2 ICLK	section 28.
0008 C418h	RTC	Binary Counter 0 Alarm Enable Register	BCNT0AER	8	8	2 or 3 PCLKB	2 ICLK	section 28.
0008 C41Ah	RTC	Month Alarm Register	RMONAR	8	8	2 or 3 PCLKB	2 ICLK	section 28.
0008 C41Ah	RTC	Binary Counter 1 Alarm Enable Register	BCNT1AER	8	8	2 or 3 PCLKB	2 ICLK	section 28.
0008 C41Ch	RTC	Year Alarm Register	RYRAR	16	16	2 or 3 PCLKB	2 ICLK	section 28.
0008 C41Ch	RTC	Binary Counter 2 Alarm Enable Register	BCNT2AER	16	16	2 or 3 PCLKB	2 ICLK	section 28.
0008 C41Eh	RTC	Year Alarm Enable Register	RYRAREN	8	8	2 or 3 PCLKB	2 ICLK	section 28.
0008 C41Eh	RTC	Binary Counter 3 Alarm Enable Register	BCNT3AER	8	8	2 or 3 PCLKB	2 ICLK	section 28.
0008 C422h	RTC	RTC Control Register 1	RCR1	8	8	2 or 3 PCLKB	2 ICLK	section 28.
0008 C424h	RTC	RTC Control Register 2	RCR2	8	8	2 or 3 PCLKB	2 ICLK	section 28.
0008 C426h	RTC	RTC Control Register 3	RCR3	8	8	2 or 3 PCLKB	2 ICLK	section 28.
0008 C42Eh	RTC	Time Error Adjustment Register	RADJ	8	8	2 or 3 PCLKB	2 ICLK	section 28.
0008 C440h	RTC	Time Capture Control Register 0	RTCCR0	8	8	2 or 3 PCLKB	2 ICLK	section 28.
0008 C442h	RTC	Time Capture Control Register 1	RTCCR1	8	8	2 or 3 PCLKB	2 ICLK	section 28.
0008 C452h	RTC	Second Capture Register 0	RSECCP0	8	8	2 or 3 PCLKB	2 ICLK	section 28.
0008 C452h	RTC	BCNT0 Capture Register 0	BCNT0CP0	8	8	2 or 3 PCLKB	2 ICLK	section 28.
0008 C454h	RTC	Minute Capture Register 0	RMINCP0	8	8	2 or 3 PCLKB	2 ICLK	section 28.
0008 C454h	RTC	BCNT1 Capture Register 0	BCNT1CP0	8	8	2 or 3 PCLKB	2 ICLK	section 28.
0008 C456h	RTC	Hour Capture Register 0	RHRCP0	8	8	2 or 3 PCLKB	2 ICLK	section 28.
0008 C456h	RTC	BCNT2 Capture Register 0	BCNT2CP0	8	8	2 or 3 PCLKB	2 ICLK	section 28.
0008 C45Ah	RTC	Date Capture Register 0	RDAYCP0	8	8	2 or 3 PCLKB	2 ICLK	section 28.
0008 C45Ah	RTC	BCNT3 Capture Register 0	BCNT3CP0	8	8	2 or 3 PCLKB	2 ICLK	section 28.
0008 C45Ch	RTC	Month Capture Register 0	RMONCP0	8	8	2 or 3 PCLKB	2 ICLK	section 28.
0008 C462h	RTC	Second Capture Register 1	RSECCP1	8	8	2 or 3 PCLKB	2 ICLK	section 28.
0008 C462h	RTC	BCNT0 Capture Register 1	BCNT0CP1	8	8	2 or 3 PCLKB	2 ICLK	section 28.
0008 C464h	RTC	Minute Capture Register 1	RMINCP1	8	8	2 or 3 PCLKB	2 ICLK	section 28.
0008 C464h	RTC	BCNT1 Capture Register 1	BCNT1CP1	8	8	2 or 3 PCLKB	2 ICLK	section 28.
0008 C466h	RTC	Hour Capture Register 1	RHRCP1	8	8	2 or 3 PCLKB	2 ICLK	section 28.
0008 C466h	RTC	BCNT2 Capture Register 1	BCNT2CP1	8	8	2 or 3 PCLKB	2 ICLK	section 28.
0008 C46Ah	RTC	Date Capture Register 1	RDAYCP1	8	8	2 or 3 PCLKB	2 ICLK	section 28.
0008 C46Ah	RTC	BCNT3 Capture Register 1	BCNT3CP1	8	8	2 or 3 PCLKB	2 ICLK	section 28.
0008 C46Ch	RTC	Month Capture Register 1	RMONCP1	8	8	2 or 3 PCLKB	2 ICLK	section 28.
0008 C5A0h	CMPB	Comparator B1 Control Register 1	CPB1CNT1	8	8	2 or 3 PCLKB	2 ICLK	section 47.
0008 C5A1h	CMPB	Comparator B1 Control Register 2	CPB1CNT2	8	8	2 or 3 PCLKB	2 ICLK	section 47.
0008 C5A2h	CMPB	Comparator B1 Flag Register	CPB1FLG	8	8	2 or 3 PCLKB	2 ICLK	section 47.
0008 C5A3h	CMPB	Comparator B1 Interrupt Control Register	CPB1INT	8	8	2 or 3 PCLKB	2 ICLK	section 47.
0008 C5A4h	CMPB	Comparator B1 Filter Select Register	CPB1F	8	8	2 or 3 PCLKB	2 ICLK	section 47.
0008 C5A5h	CMPB	Comparator B1 Mode Select Register	CPB1MD	8	8	2 or 3 PCLKB	2 ICLK	section 47.
0008 C5A6h	CMPB	Comparator B1 Reference Input Voltage Select Register	CPB1REF	8	8	2 or 3 PCLKB	2 ICLK	section 47.
0008 C5A7h	CMPB	Comparator B1 Output Control Register	CPB1OCR	8	8	2 or 3 PCLKB	2 ICLK	section 47.
000A 0000h	USB0	System Configuration Control Register	SYSCFG	16	16	3, 4 PCLKB	2 ICLK	section 32.
000A 0004h	USB0	System Configuration Status Register 0	SYSSTS0	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9 × (frequency ratio of ICLK/ PCLKB)*2	section 32.
000A 0008h	USB0	Device State Control Register 0	DVSTCTR0	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9 × (frequency ratio of ICLK/ PCLKB)*2	section 32.
000A 0014h	USB0	CFIFO Port Register	CFIFO	16	16	3, 4 PCLKB	2 ICLK	section 32.
000A 0018h	USB0	D0FIFO Port Register	D0FIFO	16	16	3, 4 PCLKB	2 ICLK	section 32.

Downloaded from Arrow.com.

Table 5.1 List of I/O Registers (Address Order) (18/31)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK <PCLK	
000A 001Ch	USB0	D1FIFO Port Register	D1FIFO	16	16	3, 4 PCLKB	2 ICLK	section 32.
000A 0020h	USB0	CFIFO Port Select Register	CFIFOSEL	16	16	3, 4 PCLKB	2 ICLK	section 32.
000A 0022h	USB0	CFIFO Port Control Register	CFIFOCTR	16	16	3, 4 PCLKB	2 ICLK	section 32.
000A 0028h	USB0	D0FIFO Port Select Register	D0FIFOSEL	16	16	3, 4 PCLKB	2 ICLK	section 32.
000A 002Ah	USB0	D0FIFO Port Control Register	D0FIFOCTR	16	16	3, 4 PCLKB	2 ICLK	section 32.
000A 002Ch	USB0	D1FIFO Port Select Register	D1FIFOSEL	16	16	3, 4 PCLKB	2 ICLK	section 32.
000A 002Eh	USB0	D1FIFO Port Control Register	D1FIFOCTR	16	16	3, 4 PCLKB	2 ICLK	section 32.
000A 0030h	USB0	Interrupt Enable Register 0	INTENB0	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*2	section 32.
000A 0032h	USB0	Interrupt Enable Register 1	INTENB1	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*2	section 32.
000A 0036h	USB0	BRDY Interrupt Enable Register	BRDYENB	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*2	section 32.
000A 0038h	USB0	NRDY Interrupt Enable Register	NRDYENB	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*2	section 32.
000A 003Ah	USB0	BEMP Interrupt Enable Register	BEMPENB	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*2	section 32.
000A 003Ch	USB0	SOF Output Configuration Register	SOFCFG	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*2	section 32.
000A 0040h	USB0	Interrupt Status Register 0	INTSTS0	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*2	section 32.
000A 0042h	USB0	Interrupt Status Register 1	INTSTS1	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*2	section 32.
000A 0046h	USB0	BRDY Interrupt Status Register	BRDYSTS	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*2	section 32.
000A 0048h	USB0	NRDY Interrupt Status Register	NRDYSTS	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*2	section 32.
000A 004Ah	USB0	BEMP Interrupt Status Register	BEMPSTS	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*2	section 32.
000A 004Ch	USB0	Frame Number Register	FRMNUM	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*2	section 32.
000A 0054h	USB0	USB Request Type Register	USBREQ	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*2	section 32.
000A 0056h	USB0	USB Request Value Register	USBVAL	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*2	section 32.

Downloaded from Arrow.com.

Table 5.1 List of I/O Registers (Address Order) (19/31)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK <PCLK	
000A 0058h	USB0	USB Request Index Register	USBINDX	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*2	section 32.
000A 005Ah	USB0	USB Request Length Register	USBLENG	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*2	section 32.
000A 005Ch	USB0	DCP Configuration Register	DCPCFG	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*2	section 32.
000A 005Eh	USB0	DCP Maximum Packet Size Register	DCPMAXP	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*2	section 32.
000A 0060h	USB0	DCP Control Register	DCPCTR	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*2	section 32.
000A 0064h	USB0	Pipe Window Select Register	PIPESEL	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*2	section 32.
000A 0068h	USB0	Pipe Configuration Register	PIPECFG	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*2	section 32.
000A 006Ch	USB0	Pipe Maximum Packet Size Register	PIPEMAXP	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*2	section 32.
000A 006Eh	USB0	Pipe Cycle Control Register	PIPEPERI	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*2	section 32.
000A 0070h	USB0	Pipe 1 Control Register	PIPE1CTR	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*2	section 32.
000A 0072h	USB0	Pipe 2 Control Register	PIPE2CTR	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*2	section 32.
000A 0074h	USB0	Pipe 3 Control Register	PIPE3CTR	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*2	section 32.
000A 0076h	USB0	Pipe 4 Control Register	PIPE4CTR	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*2	section 32.
000A 0078h	USB0	Pipe 5 Control Register	PIPE5CTR	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*2	section 32.
000A 007Ah	USB0	Pipe 6 Control Register	PIPE6CTR	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*2	section 32.
000A 007Ch	USB0	Pipe 7 Control Register	PIPE7CTR	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*2	section 32.

Downloaded from Arrow.com.

Table 5.1 List of I/O Registers (Address Order) (20/31)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK <PCLK	
000A 007Eh	USB0	Pipe 8 Control Register	PIPE8CTR	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*2	section 32.
000A 0080h	USB0	Pipe 9 Control Register	PIPE9CTR	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*2	section 32.
000A 0090h	USB0	Pipe 1 Transaction Counter Enable Register	PIPE1TRE	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*2	section 32.
000A 0092h	USB0	Pipe 1 Transaction Counter Register	PIPE1TRN	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*2	section 32.
000A 0094h	USB0	Pipe 2 Transaction Counter Enable Register	PIPE2TRE	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*2	section 32.
000A 0096h	USB0	Pipe 2 Transaction Counter Register	PIPE2TRN	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*2	section 32.
000A 0098h	USB0	Pipe 3 Transaction Counter Enable Register	PIPE3TRE	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*2	section 32.
000A 009Ah	USB0	Pipe 3 Transaction Counter Register	PIPE3TRN	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*2	section 32.
000A 009Ch	USB0	Pipe 4 Transaction Counter Enable Register	PIPE4TRE	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*2	section 32.
000A 009Eh	USB0	Pipe 4 Transaction Counter Register	PIPE4TRN	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*2	section 32.
000A 00A0h	USB0	Pipe 5 Transaction Counter Enable Register	PIPE5TRE	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*2	section 32.
000A 00A2h	USB0	Pipe 5 Transaction Counter Register	PIPE5TRN	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*2	section 32.
000A 00B0h	USB0	BC Control Register 0	USBBCCTRL0	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*2	section 32.
000A 00CCh	USB0	USB Module Control Register	USBMC	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*2	section 32.
000A 00D0h	USB0	Device Address 0 Configuration Register	DEVADD0	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*2	section 32.
000A 00D2h	USB0	Device Address 1 Configuration Register	DEVADD1	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*2	section 32.

Downloaded from Arrow.com.

Table 5.1 List of I/O Registers (Address Order) (21/31)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK <PCLK	
000A 00D4h	USB0	Device Address 2 Configuration Register	DEVADD2	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*2	section 32.
000A 00D6h	USB0	Device Address 3 Configuration Register	DEVADD3	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*2	section 32.
000A 00D8h	USB0	Device Address 4 Configuration Register	DEVADD4	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*2	section 32.
000A 00DAh	USB0	Device Address 5 Configuration Register	DEVADD5	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*2	section 32.
000A 0900h	CTSU	CTSU Control Register 0	CTSUCR0	8	8	2 or 3 PCLKB	2 ICLK	section 43.
000A 0901h	CTSU	CTSU Control Register 1	CTSUCR1	8	8	2 or 3 PCLKB	2 ICLK	section 43.
000A 0902h	CTSU	CTSU Synchronous Noise Reduction Setting Register	CTSUSDPRS	8	8	2 or 3 PCLKB	2 ICLK	section 43.
000A 0903h	CTSU	CTSU Sensor Stabilization Wait Control Register	CTSUSST	8	8	2 or 3 PCLKB	2 ICLK	section 43.
000A 0904h	CTSU	CTSU Measurement Channel Register 0	CTSUMCH0	8	8	2 or 3 PCLKB	2 ICLK	section 43.
000A 0905h	CTSU	CTSU Measurement Channel Register 1	CTSUMCH1	8	8	2 or 3 PCLKB	2 ICLK	section 43.
000A 0906h	CTSU	CTSU Channel Enable Control Register 0	CTSUCHAC0	8	8	2 or 3 PCLKB	2 ICLK	section 43.
000A 0907h	CTSU	CTSU Channel Enable Control Register 1	CTSUCHAC1	8	8	2 or 3 PCLKB	2 ICLK	section 43.
000A 0908h	CTSU	CTSU Channel Enable Control Register 2	CTSUCHAC2	8	8	2 or 3 PCLKB	2 ICLK	section 43.
000A 0909h	CTSU	CTSU Channel Enable Control Register 3	CTSUCHAC3	8	8	2 or 3 PCLKB	2 ICLK	section 43.
000A 090Ah	CTSU	CTSU Channel Enable Control Register 4	CTSUCHAC4	8	8	2 or 3 PCLKB	2 ICLK	section 43.
000A 090Bh	CTSU	CTSU Channel Transmit/Receive Control Register 0	CTSUCHTRC0	8	8	2 or 3 PCLKB	2 ICLK	section 43.
000A 090Ch	CTSU	CTSU Channel Transmit/Receive Control Register 1	CTSUCHTRC1	8	8	2 or 3 PCLKB	2 ICLK	section 43.
000A 090Dh	CTSU	CTSU Channel Transmit/Receive Control Register 2	CTSUCHTRC2	8	8	2 or 3 PCLKB	2 ICLK	section 43.
000A 090Eh	CTSU	CTSU Channel Transmit/Receive Control Register 3	CTSUCHTRC3	8	8	2 or 3 PCLKB	2 ICLK	section 43.
000A 090Fh	CTSU	CTSU Channel Transmit/Receive Control Register 4	CTSUCHTRC4	8	8	2 or 3 PCLKB	2 ICLK	section 43.
000A 0910h	CTSU	CTSU High-Pass Noise Reduction Control Register	CTSUDCLKC	8	8	2 or 3 PCLKB	2 ICLK	section 43.
000A 0911h	CTSU	CTSU Status Register	CTSUST	8	8	2 or 3 PCLKB	2 ICLK	section 43.
000A 0912h	CTSU	CTSU High-Pass Noise Reduction Spectrum Diffusion Control Register	CTSUSSC	16	16	2 or 3 PCLKB	2 ICLK	section 43.
000A 0914h	CTSU	CTSU Sensor Offset Register 0	CTSUSO0	16	16	2 or 3 PCLKB	2 ICLK	section 43.
000A 0916h	CTSU	CTSU Sensor Offset Register 1	CTSUSO1	16	16	2 or 3 PCLKB	2 ICLK	section 43.
000A 0918h	CTSU	CTSU Sensor Counter	CTSUSC	16	16	2 or 3 PCLKB	2 ICLK	section 43.
000A 091Ah	CTSU	CTSU Reference Counter	CTSURC	16	16	2 or 3 PCLKB	2 ICLK	section 43.
000A 091Ch	CTSU	CTSU Error Status Register	CTSUERRS	16	16	2 or 3 PCLKB	2 ICLK	section 43.
000A 8300h	RSCAN0	Bit Configuration Register L	CFGL	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8302h	RSCAN0	Bit Configuration Register H	CFGH	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8304h	RSCAN0	Control Register L	CTRL	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8306h	RSCAN0	Control Register H	CTRH	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8308h	RSCAN0	Status Register L	STSL	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 830Ah	RSCAN0	Status Register H	STSH	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 830Ch	RSCAN0	Error Flag Register L	ERFLL	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 830Eh	RSCAN0	Error Flag Register H	ERFLH	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8322h	RSCAN	Global Configuration Register L	GCFGL	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8324h	RSCAN	Global Configuration Register H	GCFGH	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8326h	RSCAN	Global Control Register L	GCTRL	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8328h	RSCAN	Global Control Register H	GCTRH	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 832Ah	RSCAN	Global Status Register	GSTS	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 832Ch	RSCAN	Global Error Flag Register	GERFLL	8	8	2 or 3 PCLKB	2 ICLK	section 36.

Downloaded from Arrow.com.

Table 5.1 List of I/O Registers (Address Order) (22/31)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK <PCLK	
000A 832Eh	RSCAN	Timestamp Register	GTSC	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8330h	RSCAN	Receive Rule Number Configuration Register	GAFLCFG	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8332h	RSCAN	Receive Buffer Number Configuration Register	RMNB	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8334h	RSCAN	Receive Buffer Receive Complete Flag Register	RMND0	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8338h	RSCAN	Receive FIFO Control Register 0	RFCC0	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 833Ah	RSCAN	Receive FIFO Control Register 1	RFCC1	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8340h	RSCAN	Receive FIFO Status Register 0	RFSTS0	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8342h	RSCAN	Receive FIFO Status Register 1	RFSTS1	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8348h	RSCAN	Receive FIFO Pointer Control Register 0	RFPCTR0	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 834Ah	RSCAN	Receive FIFO Pointer Control Register 1	RFPCTR1	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8350h	RSCAN0	Transmit/Receive FIFO Control Register 0L	CFCCL0	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8352h	RSCAN0	Transmit/Receive FIFO Control Register 0H	CFCCH0	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8358h	RSCAN0	Transmit/Receive FIFO Status Register 0	CFSTS0	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 835Ch	RSCAN0	Transmit/Receive FIFO Pointer Control Register 0	CFPCTR0	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8360h	RSCAN	Receive FIFO Message Lost Status Register	RFMSTS	8	8	2 or 3 PCLKB	2 ICLK	section 36.
000A 8361h	RSCAN0	Transmit/Receive FIFO Message Lost Status Register	CFMSTS	8	8	2 or 3 PCLKB	2 ICLK	section 36.
000A 8362h	RSCAN	Receive FIFO Interrupt Status Register	RFISTS	8	8	2 or 3 PCLKB	2 ICLK	section 36.
000A 8363h	RSCAN	Transmit/Receive FIFO Receive Interrupt Status Register	CFISTS	8	8	2 or 3 PCLKB	2 ICLK	section 36.
000A 8364h	RSCAN0	Transmit Buffer Control Register 0	TMC0	8	8	2 or 3 PCLKB	2 ICLK	section 36.
000A 8365h	RSCAN0	Transmit Buffer Control Register 1	TMC1	8	8	2 or 3 PCLKB	2 ICLK	section 36.
000A 8366h	RSCAN0	Transmit Buffer Control Register 2	TMC2	8	8	2 or 3 PCLKB	2 ICLK	section 36.
000A 8367h	RSCAN0	Transmit Buffer Control Register 3	TMC3	8	8	2 or 3 PCLKB	2 ICLK	section 36.
000A 836Ch	RSCAN0	Transmit Buffer Status Register 0	TMSTS0	8	8	2 or 3 PCLKB	2 ICLK	section 36.
000A 836Dh	RSCAN0	Transmit Buffer Status Register 1	TMSTS1	8	8	2 or 3 PCLKB	2 ICLK	section 36.
000A 836Eh	RSCAN0	Transmit Buffer Status Register 2	TMSTS2	8	8	2 or 3 PCLKB	2 ICLK	section 36.
000A 836Fh	RSCAN0	Transmit Buffer Status Register 3	TMSTS3	8	8	2 or 3 PCLKB	2 ICLK	section 36.
000A 8374h	RSCAN0	Transmit Buffer Transmit Request Status Register	TMTRSTS	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8376h	RSCAN0	Transmit Buffer Transmit Complete Status Register	TMTCSTS	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8378h	RSCAN0	Transmit Buffer Transmit Abort Status Register	TMTASTS	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 837Ah	RSCAN0	Transmit Buffer Interrupt Enable Register	TMIEC	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 837Ch	RSCAN0	Transmit History Buffer Control Register	THLCC0	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8380h	RSCAN0	Transmit History Buffer Status Register	THLSTS0	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8384h	RSCAN0	Transmit History Buffer Pointer Control Register	THLPCTR0	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8388h	RSCAN	Global Transmit Interrupt Status Register	GTINTSTS	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 838Ah	RSCAN	Global RAM Window Control Register	GRWCR	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 838Ch	RSCAN	Global Test Configuration Register	GTSTCFG	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 838Eh	RSCAN	Global Test Control Register	GTSTCTRL	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8394h	RSCAN	Global Test Protection Unlock Register	GLOCKK	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83A0h	RSCAN	Receive Rule Entry Register 0AL	GAFLIDL0	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83A0h	RSCAN	Receive Buffer Register 0AL	RMIDL0	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83A2h	RSCAN	Receive Rule Entry Register 0AH	GAFLIDH0	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83A2h	RSCAN	Receive Buffer Register 0AH	RMIDH0	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83A4h	RSCAN	Receive Rule Entry Register 0BL	GAFLML0	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83A4h	RSCAN	Receive Buffer Register 0BL	RMTS0	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83A6h	RSCAN	Receive Rule Entry Register 0BH	GAFLMH0	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83A6h	RSCAN	Receive Buffer Register 0BH	RMPTR0	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83A8h	RSCAN	Receive Rule Entry Register 0CL	GAFLPL0	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83A8h	RSCAN	Receive Buffer Register 0CL	RMDF00	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83AAh	RSCAN	Receive Rule Entry Register 0CH	GAFLPH0	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83AAh	RSCAN	Receive Buffer Register 0CH	RMDF10	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83ACh	RSCAN	Receive Rule Entry Register 1AL	GAFLIDL1	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83ACh	RSCAN	Receive Buffer Register 0DL	RMDF20	16	16	2 or 3 PCLKB	2 ICLK	section 36.

Downloaded from Arrow.com.

Table 5.1 List of I/O Registers (Address Order) (23/31)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK <PCLK	
000A 83AEh	RSCAN	Receive Rule Entry Register 1AH	GAFLIDH1	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83AEh	RSCAN	Receive Buffer Register 0DH	RMDF30	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83B0h	RSCAN	Receive Rule Entry Register 1BL	GAFLML1	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83B0h	RSCAN	Receive Buffer Register 1AL	RMIDL1	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83B2h	RSCAN	Receive Rule Entry Register 1BH	GAFLMH1	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83B2h	RSCAN	Receive Buffer Register 1AH	RMIDH1	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83B4h	RSCAN	Receive Rule Entry Register 1CL	GAFLPL1	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83B4h	RSCAN	Receive Buffer Register 1BL	RMTS1	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83B6h	RSCAN	Receive Rule Entry Register 1CH	GAFLPH1	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83B6h	RSCAN	Receive Buffer Register 1BH	RMPTR1	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83B8h	RSCAN	Receive Rule Entry Register 2AL	GAFLIDL2	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83B8h	RSCAN	Receive Buffer Register 1CL	RMDF01	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83BAh	RSCAN	Receive Rule Entry Register 2AH	GAFLIDH2	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83BAh	RSCAN	Receive Buffer Register 1CH	RMDF11	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83BCh	RSCAN	Receive Rule Entry Register 2BL	GAFLML2	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83BCh	RSCAN	Receive Buffer Register 1DL	RMDF21	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83BEh	RSCAN	Receive Rule Entry Register 2BH	GAFLMH2	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83BEh	RSCAN	Receive Buffer Register 1DH	RMDF31	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83C0h	RSCAN	Receive Rule Entry Register 2CL	GAFLPL2	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83C0h	RSCAN	Receive Buffer Register 2AL	RMIDL2	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83C2h	RSCAN	Receive Rule Entry Register 2CH	GAFLPH2	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83C2h	RSCAN	Receive Buffer Register 2AH	RMIDH2	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83C4h	RSCAN	Receive Rule Entry Register 3AL	GAFLIDL3	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83C4h	RSCAN	Receive Buffer Register 2BL	RMTS2	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83C6h	RSCAN	Receive Rule Entry Register 3AH	GAFLIDH3	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83C6h	RSCAN	Receive Buffer Register 2BH	RMPTR2	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83C8h	RSCAN	Receive Rule Entry Register 3BL	GAFLML3	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83C8h	RSCAN	Receive Buffer Register 2CL	RMDF02	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83CAh	RSCAN	Receive Rule Entry Register 3BH	GAFLMH3	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83CAh	RSCAN	Receive Buffer Register 2CH	RMDF12	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83CCh	RSCAN	Receive Rule Entry Register 3CL	GAFLPL3	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83CCh	RSCAN	Receive Buffer Register 2DL	RMDF22	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83CEh	RSCAN	Receive Rule Entry Register 3CH	GAFLPH3	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83CEh	RSCAN	Receive Buffer Register 2DH	RMDF32	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83D0h	RSCAN	Receive Rule Entry Register 4AL	GAFLIDL4	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83D0h	RSCAN	Receive Buffer Register 3AL	RMIDL3	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83D2h	RSCAN	Receive Rule Entry Register 4AH	GAFLIDH4	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83D2h	RSCAN	Receive Buffer Register 3AH	RMIDH3	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83D4h	RSCAN	Receive Rule Entry Register 4BL	GAFLML4	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83D4h	RSCAN	Receive Buffer Register 3BL	RMTS3	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83D6h	RSCAN	Receive Rule Entry Register 4BH	GAFLMH4	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83D6h	RSCAN	Receive Buffer Register 3BH	RMPTR3	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83D8h	RSCAN	Receive Rule Entry Register 4CL	GAFLPL4	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83D8h	RSCAN	Receive Buffer Register 3CL	RMDF03	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83DAh	RSCAN	Receive Rule Entry Register 4CH	GAFLPH4	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83DAh	RSCAN	Receive Buffer Register 3CH	RMDF13	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83DCh	RSCAN	Receive Rule Entry Register 5AL	GAFLIDL5	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83DCh	RSCAN	Receive Buffer Register 3DL	RMDF23	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83DEh	RSCAN	Receive Rule Entry Register 5AH	GAFLIDH5	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83DEh	RSCAN	Receive Buffer Register 3DH	RMDF33	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83E0h	RSCAN	Receive Rule Entry Register 5BL	GAFLML5	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83E0h	RSCAN	Receive Buffer Register 4AL	RMIDL4	16	16	2 or 3 PCLKB	2 ICLK	section 36.

Downloaded from Arrow.com.

Table 5.1 List of I/O Registers (Address Order) (24/31)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK <PCLK	
000A 83E2h	RSCAN	Receive Rule Entry Register 5BH	GAFLMH5	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83E2h	RSCAN	Receive Buffer Register 4AH	RMIDH4	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83E4h	RSCAN	Receive Rule Entry Register 5CL	GAFLPL5	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83E4h	RSCAN	Receive Buffer Register 4BL	RMTS4	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83E6h	RSCAN	Receive Rule Entry Register 5CH	GAFLPH5	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83E6h	RSCAN	Receive Buffer Register 4BH	RMPTR4	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83E8h	RSCAN	Receive Rule Entry Register 6AL	GAFLIDL6	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83E8h	RSCAN	Receive Buffer Register 4CL	RMDF04	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83EAh	RSCAN	Receive Rule Entry Register 6AH	GAFLIDH6	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83EAh	RSCAN	Receive Buffer Register 4CH	RMDF14	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83ECh	RSCAN	Receive Rule Entry Register 6BL	GAFLML6	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83ECh	RSCAN	Receive Buffer Register 4DL	RMDF24	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83EEh	RSCAN	Receive Rule Entry Register 6BH	GAFLMH6	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83EEh	RSCAN	Receive Buffer Register 4DH	RMDF34	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83F0h	RSCAN	Receive Rule Entry Register 6CL	GAFLPL6	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83F0h	RSCAN	Receive Buffer Register 5AL	RMIDL5	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83F2h	RSCAN	Receive Rule Entry Register 6CH	GAFLPH6	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83F2h	RSCAN	Receive Buffer Register 5AH	RMIDH5	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83F4h	RSCAN	Receive Rule Entry Register 7AL	GAFLIDL7	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83F4h	RSCAN	Receive Buffer Register 5BL	RMTS5	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83F6h	RSCAN	Receive Rule Entry Register 7AH	GAFLIDH7	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83F6h	RSCAN	Receive Buffer Register 5BH	RMPTR5	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83F8h	RSCAN	Receive Rule Entry Register 7BL	GAFLML7	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83F8h	RSCAN	Receive Buffer Register 5CL	RMDF05	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83FAh	RSCAN	Receive Rule Entry Register 7BH	GAFLMH7	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83FAh	RSCAN	Receive Buffer Register 5CH	RMDF15	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83FCh	RSCAN	Receive Rule Entry Register 7CL	GAFLPL7	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83FCh	RSCAN	Receive Buffer Register 5DL	RMDF25	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83FEh	RSCAN	Receive Rule Entry Register 7CH	GAFLPH7	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 83FEh	RSCAN	Receive Buffer Register 5DH	RMDF35	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8400h	RSCAN	Receive Rule Entry Register 8AL	GAFLIDL8	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8400h	RSCAN	Receive Buffer Register 6AL	RMIDL6	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8402h	RSCAN	Receive Rule Entry Register 8AH	GAFLIDH8	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8402h	RSCAN	Receive Buffer Register 6AH	RMIDH6	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8404h	RSCAN	Receive Rule Entry Register 8BL	GAFLML8	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8404h	RSCAN	Receive Buffer Register 6BL	RMTS6	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8406h	RSCAN	Receive Rule Entry Register 8BH	GAFLMH8	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8406h	RSCAN	Receive Buffer Register 6BH	RMPTR6	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8408h	RSCAN	Receive Rule Entry Register 8CL	GAFLPL8	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8408h	RSCAN	Receive Buffer Register 6CL	RMDF06	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 840Ah	RSCAN	Receive Rule Entry Register 8CH	GAFLPH8	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 840Ah	RSCAN	Receive Buffer Register 6CH	RMDF16	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 840Ch	RSCAN	Receive Rule Entry Register 9AL	GAFLIDL9	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 840Ch	RSCAN	Receive Buffer Register 6DL	RMDF26	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 840Eh	RSCAN	Receive Rule Entry Register 9AH	GAFLIDH9	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 840Eh	RSCAN	Receive Buffer Register 6DH	RMDF36	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8410h	RSCAN	Receive Rule Entry Register 9BL	GAFLML9	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8410h	RSCAN	Receive Buffer Register 7AL	RMIDL7	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8412h	RSCAN	Receive Rule Entry Register 9BH	GAFLMH9	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8412h	RSCAN	Receive Buffer Register 7AH	RMIDH7	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8414h	RSCAN	Receive Rule Entry Register 9CL	GAFLPL9	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8414h	RSCAN	Receive Buffer Register 7BL	RMTS7	16	16	2 or 3 PCLKB	2 ICLK	section 36.

Downloaded from Arrow.com.

Table 5.1 List of I/O Registers (Address Order) (25/31)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK <PCLK	
000A 8416h	RSCAN	Receive Rule Entry Register 9CH	GAFLPH9	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8416h	RSCAN	Receive Buffer Register 7BH	RMPTR7	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8418h	RSCAN	Receive Rule Entry Register 10AL	GAFLIDL10	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8418h	RSCAN	Receive Buffer Register 7CL	RMDF07	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 841Ah	RSCAN	Receive Rule Entry Register 10AH	GAFLIDH10	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 841Ah	RSCAN	Receive Buffer Register 7CH	RMDF17	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 841Ch	RSCAN	Receive Rule Entry Register 10BL	GAFLML10	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 841Ch	RSCAN	Receive Buffer Register 7DL	RMDF27	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 841Eh	RSCAN	Receive Rule Entry Register 10BH	GAFLMH10	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 841Eh	RSCAN	Receive Buffer Register 7DH	RMDF37	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8420h	RSCAN	Receive Rule Entry Register 10CL	GAFLPL10	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8420h	RSCAN	Receive Buffer Register 8AL	RMIDL8	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8422h	RSCAN	Receive Rule Entry Register 10CH	GAFLPH10	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8422h	RSCAN	Receive Buffer Register 8AH	RMIDH8	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8424h	RSCAN	Receive Rule Entry Register 11AL	GAFLIDL11	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8424h	RSCAN	Receive Buffer Register 8BL	RMTS8	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8426h	RSCAN	Receive Rule Entry Register 11AH	GAFLIDH11	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8426h	RSCAN	Receive Buffer Register 8BH	RMPTR8	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8428h	RSCAN	Receive Rule Entry Register 11BL	GAFLML11	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8428h	RSCAN	Receive Buffer Register 8CL	RMDF08	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 842Ah	RSCAN	Receive Rule Entry Register 11BH	GAFLMH11	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 842Ah	RSCAN	Receive Buffer Register 8CH	RMDF18	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 842Ch	RSCAN	Receive Rule Entry Register 11CL	GAFLPL11	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 842Ch	RSCAN	Receive Buffer Register 8DL	RMDF28	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 842Eh	RSCAN	Receive Rule Entry Register 11CH	GAFLPH11	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 842Eh	RSCAN	Receive Buffer Register 8DH	RMDF38	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8430h	RSCAN	Receive Rule Entry Register 12AL	GAFLIDL12	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8430h	RSCAN	Receive Buffer Register 9AL	RMIDL9	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8432h	RSCAN	Receive Rule Entry Register 12AH	GAFLIDH12	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8432h	RSCAN	Receive Buffer Register 9AH	RMIDH9	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8434h	RSCAN	Receive Rule Entry Register 12BL	GAFLML12	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8434h	RSCAN	Receive Buffer Register 9BL	RMTS9	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8436h	RSCAN	Receive Rule Entry Register 12BH	GAFLMH12	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8436h	RSCAN	Receive Buffer Register 9BH	RMPTR9	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8438h	RSCAN	Receive Rule Entry Register 12CL	GAFLPL12	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8438h	RSCAN	Receive Buffer Register 9CL	RMDF09	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 843Ah	RSCAN	Receive Rule Entry Register 12CH	GAFLPH12	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 843Ah	RSCAN	Receive Buffer Register 9CH	RMDF19	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 843Ch	RSCAN	Receive Rule Entry Register 13AL	GAFLIDL13	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 843Ch	RSCAN	Receive Buffer Register 9DL	RMDF29	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 843Eh	RSCAN	Receive Rule Entry Register 13AH	GAFLIDH13	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 843Eh	RSCAN	Receive Buffer Register 9DH	RMDF39	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8440h	RSCAN	Receive Rule Entry Register 13BL	GAFLML13	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8440h	RSCAN	Receive Buffer Register 10AL	RMIDL10	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8442h	RSCAN	Receive Rule Entry Register 13BH	GAFLMH13	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8442h	RSCAN	Receive Buffer Register 10AH	RMIDH10	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8444h	RSCAN	Receive Rule Entry Register 13CL	GAFLPL13	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8444h	RSCAN	Receive Buffer Register 10BL	RMTS10	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8446h	RSCAN	Receive Rule Entry Register 13CH	GAFLPH13	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8446h	RSCAN	Receive Buffer Register 10BH	RMPTR10	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8448h	RSCAN	Receive Rule Entry Register 14AL	GAFLIDL14	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8448h	RSCAN	Receive Buffer Register 10CL	RMDF010	16	16	2 or 3 PCLKB	2 ICLK	section 36.

Downloaded from Arrow.com.

Table 5.1 List of I/O Registers (Address Order) (26/31)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK <PCLK	
000A 844Ah	RSCAN	Receive Rule Entry Register 14AH	GAFLIDH14	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 844Ah	RSCAN	Receive Buffer Register 10CH	RMDF110	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 844Ch	RSCAN	Receive Rule Entry Register 14BL	GAFLML14	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 844Ch	RSCAN	Receive Buffer Register 10DL	RMDF210	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 844Eh	RSCAN	Receive Rule Entry Register 14BH	GAFLMH14	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 844Eh	RSCAN	Receive Buffer Register 10DH	RMDF310	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8450h	RSCAN	Receive Rule Entry Register 14CL	GAFLPL14	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8450h	RSCAN	Receive Buffer Register 11AL	RMIDL11	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8452h	RSCAN	Receive Rule Entry Register 14CH	GAFLPH14	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8452h	RSCAN	Receive Buffer Register 11AH	RMIDH11	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8454h	RSCAN	Receive Rule Entry Register 15AL	GAFLIDL15	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8454h	RSCAN	Receive Buffer Register 11BL	RMTS11	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8456h	RSCAN	Receive Rule Entry Register 15AH	GAFLIDH15	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8456h	RSCAN	Receive Buffer Register 11BH	RMPTR11	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8458h	RSCAN	Receive Rule Entry Register 15BL	GAFLML15	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8458h	RSCAN	Receive Buffer Register 11CL	RMDF011	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 845Ah	RSCAN	Receive Rule Entry Register 15BH	GAFLMH15	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 845Ah	RSCAN	Receive Buffer Register 11CH	RMDF111	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 845Ch	RSCAN	Receive Rule Entry Register 15CL	GAFLPL15	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 845Ch	RSCAN	Receive Buffer Register 11DL	RMDF211	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 845Eh	RSCAN	Receive Rule Entry Register 15CH	GAFLPH15	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 845Eh	RSCAN	Receive Buffer Register 11DH	RMDF311	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8460h	RSCAN	Receive Buffer Register 12AL	RMIDL12	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8462h	RSCAN	Receive Buffer Register 12AH	RMIDH12	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8464h	RSCAN	Receive Buffer Register 12BL	RMTS12	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8466h	RSCAN	Receive Buffer Register 12BH	RMPTR12	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8468h	RSCAN	Receive Buffer Register 12CL	RMDF012	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 846Ah	RSCAN	Receive Buffer Register 12CH	RMDF112	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 846Ch	RSCAN	Receive Buffer Register 12DL	RMDF212	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 846Eh	RSCAN	Receive Buffer Register 12DH	RMDF312	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8470h	RSCAN	Receive Buffer Register 13AL	RMIDL13	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8472h	RSCAN	Receive Buffer Register 13AH	RMIDH13	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8474h	RSCAN	Receive Buffer Register 13BL	RMTS13	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8476h	RSCAN	Receive Buffer Register 13BH	RMPTR13	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8478h	RSCAN	Receive Buffer Register 13CL	RMDF013	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 847Ah	RSCAN	Receive Buffer Register 13CH	RMDF113	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 847Ch	RSCAN	Receive Buffer Register 13DL	RMDF213	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 847Eh	RSCAN	Receive Buffer Register 13DH	RMDF313	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8480h	RSCAN	Receive Buffer Register 14AL	RMIDL14	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8482h	RSCAN	Receive Buffer Register 14AH	RMIDH14	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8484h	RSCAN	Receive Buffer Register 14BL	RMTS14	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8486h	RSCAN	Receive Buffer Register 14BH	RMPTR14	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8488h	RSCAN	Receive Buffer Register 14CL	RMDF014	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 848Ah	RSCAN	Receive Buffer Register 14CH	RMDF114	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 848Ch	RSCAN	Receive Buffer Register 14DL	RMDF214	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 848Eh	RSCAN	Receive Buffer Register 14DH	RMDF314	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8490h	RSCAN	Receive Buffer Register 15AL	RMIDL15	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8492h	RSCAN	Receive Buffer Register 15AH	RMIDH15	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8494h	RSCAN	Receive Buffer Register 15BL	RMTS15	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8496h	RSCAN	Receive Buffer Register 15BH	RMPTR15	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8498h	RSCAN	Receive Buffer Register 15CL	RMDF015	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 849Ah	RSCAN	Receive Buffer Register 15CH	RMDF115	16	16	2 or 3 PCLKB	2 ICLK	section 36.

Downloaded from Arrow.com.

Table 5.1 List of I/O Registers (Address Order) (27/31)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK <PCLK	
000A 849Ch	RSCAN	Receive Buffer Register 15DL	RMDF215	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 849Eh	RSCAN	Receive Buffer Register 15DH	RMDF315	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8580h to 000A 859Fh	RSCAN	RAM Test Register 0 to RAM Test Register 15	RPGACC0 to RPGACC15	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 85A0h	RSCAN	Receive FIFO Access Register 0AL	RFIDL0	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 85A0h	RSCAN	RAM Test Register 16	RPGACC16	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 85A2h	RSCAN	Receive FIFO Access Register 0AH	RFIDH0	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 85A2h	RSCAN	RAM Test Register 17	RPGACC17	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 85A4h	RSCAN	Receive FIFO Access Register 0BL	RFTS0	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 85A4h	RSCAN	RAM Test Register 18	RPGACC18	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 85A6h	RSCAN	Receive FIFO Access Register 0BH	RFPTR0	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 85A6h	RSCAN	RAM Test Register 19	RPGACC19	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 85A8h	RSCAN	Receive FIFO Access Register 0CL	RFDF00	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 85A8h	RSCAN	RAM Test Register 20	RPGACC20	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 85AAh	RSCAN	Receive FIFO Access Register 0CH	RFDF10	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 85AAh	RSCAN	RAM Test Register 21	RPGACC21	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 85ACh	RSCAN	Receive FIFO Access Register 0DL	RFDF20	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 85ACh	RSCAN	RAM Test Register 22	RPGACC22	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 85AEh	RSCAN	Receive FIFO Access Register 0DH	RFDF30	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 85AEh	RSCAN	RAM Test Register 23	RPGACC23	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 85B0h	RSCAN	Receive FIFO Access Register 1AL	RFIDL1	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 85B0h	RSCAN	RAM Test Register 24	RPGACC24	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 85B2h	RSCAN	Receive FIFO Access Register 1AH	RFIDH1	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 85B2h	RSCAN	RAM Test Register 25	RPGACC25	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 85B4h	RSCAN	Receive FIFO Access Register 1BL	RFTS1	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 85B4h	RSCAN	RAM Test Register 26	RPGACC26	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 85B6h	RSCAN	Receive FIFO Access Register 1BH	RFPTR1	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 85B6h	RSCAN	RAM Test Register 27	RPGACC27	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 85B8h	RSCAN	Receive FIFO Access Register 1CL	RFDF01	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 85B8h	RSCAN	RAM Test Register 28	RPGACC28	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 85BAh	RSCAN	Receive FIFO Access Register 1CH	RFDF11	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 85BAh	RSCAN	RAM Test Register 29	RPGACC29	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 85BCh	RSCAN	Receive FIFO Access Register 1DL	RFDF21	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 85BCh	RSCAN	RAM Test Register 30	RPGACC30	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 85BEh	RSCAN	Receive FIFO Access Register 1DH	RFDF31	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 85BEh	RSCAN	RAM Test Register 31	RPGACC31	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 85C0h to 000A 85DEh	RSCAN	RAM Test Register 32 to RAM Test Register 47	RPGACC32 to RPGACC47	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 85E0h	RSCAN0	Transmit/Receive FIFO Access Register 0AL	CFIDL0	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 85E0h	RSCAN	RAM Test Register 48	RPGACC48	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 85E2h	RSCAN0	Transmit/Receive FIFO Access Register 0AH	CFIDH0	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 85E2h	RSCAN	RAM Test Register 49	RPGACC49	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 85E4h	RSCAN0	Transmit/Receive FIFO Access Register 0BL	CFTS0	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 85E4h	RSCAN	RAM Test Register 50	RPGACC50	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 85E6h	RSCAN0	Transmit/Receive FIFO Access Register 0BH	CFPTR0	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 85E6h	RSCAN	RAM Test Register 51	RPGACC51	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 85E8h	RSCAN0	Transmit/Receive FIFO Access Register 0CL	CFDF00	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 85E8h	RSCAN	RAM Test Register 52	RPGACC52	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 85EAh	RSCAN0	Transmit/Receive FIFO Access Register 0CH	CFDF10	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 85EAh	RSCAN	RAM Test Register 53	RPGACC53	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 85ECh	RSCAN0	Transmit/Receive FIFO Access Register 0DL	CFDF20	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 85ECh	RSCAN	RAM Test Register 54	RPGACC54	16	16	2 or 3 PCLKB	2 ICLK	section 36.

Downloaded from Arrow.com.

Table 5.1 List of I/O Registers (Address Order) (28/31)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK <PCLK	
000A 85EEh	RSCAN0	Transmit/Receive FIFO Access Register 0DH	CFDF30	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 85EEh	RSCAN	RAM Test Register 55	RPGACC55	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 85F0h to 000A 85FEh	RSCAN	RAM Test Register 56 to RAM Test Register 63	RPGACC56 to RPGACC63	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8600h	RSCAN0	Transmit Buffer Register 0AL	TMIDL0	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8600h	RSCAN	RAM Test Register 64	RPGACC64	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8602h	RSCAN0	Transmit Buffer Register 0AH	TMIDH0	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8602h	RSCAN	RAM Test Register 65	RPGACC65	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8604h	RSCAN	RAM Test Register 66	RPGACC66	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8606h	RSCAN0	Transmit Buffer Register 0BH	TMPTR0	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8606h	RSCAN	RAM Test Register 67	RPGACC67	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8608h	RSCAN0	Transmit Buffer Register 0CL	TMDF00	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8608h	RSCAN	RAM Test Register 68	RPGACC68	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 860Ah	RSCAN0	Transmit Buffer Register 0CH	TMDF10	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 860Ah	RSCAN	RAM Test Register 69	RPGACC69	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 860Ch	RSCAN0	Transmit Buffer Register 0DL	TMDF20	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 860Ch	RSCAN	RAM Test Register 70	RPGACC70	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 860Eh	RSCAN0	Transmit Buffer Register 0DH	TMDF30	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 860Eh	RSCAN	RAM Test Register 71	RPGACC71	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8610h	RSCAN0	Transmit Buffer Register 1AL	TMIDL1	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8610h	RSCAN	RAM Test Register 72	RPGACC72	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8612h	RSCAN0	Transmit Buffer Register 1AH	TMIDH1	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8612h	RSCAN	RAM Test Register 73	RPGACC73	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8614h	RSCAN	RAM Test Register 74	RPGACC74	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8616h	RSCAN0	Transmit Buffer Register 1BH	TMPTR1	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8616h	RSCAN	RAM Test Register 75	RPGACC75	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8618h	RSCAN0	Transmit Buffer Register 1CL	TMDF01	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8618h	RSCAN	RAM Test Register 76	RPGACC76	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 861Ah	RSCAN0	Transmit Buffer Register 1CH	TMDF11	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 861Ah	RSCAN	RAM Test Register 77	RPGACC77	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 861Ch	RSCAN0	Transmit Buffer Register 1DL	TMDF21	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 861Ch	RSCAN	RAM Test Register 78	RPGACC78	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 861Eh	RSCAN0	Transmit Buffer Register 1DH	TMDF31	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 861Eh	RSCAN	RAM Test Register 79	RPGACC79	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8620h	RSCAN0	Transmit Buffer Register 2AL	TMIDL2	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8620h	RSCAN	RAM Test Register 80	RPGACC80	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8622h	RSCAN0	Transmit Buffer Register 2AH	TMIDH2	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8622h	RSCAN	RAM Test Register 81	RPGACC81	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8624h	RSCAN	RAM Test Register 82	RPGACC82	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8626h	RSCAN0	Transmit Buffer Register 2BH	TMPTR2	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8626h	RSCAN	RAM Test Register 83	RPGACC83	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8628h	RSCAN0	Transmit Buffer Register 2CL	TMDF02	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8628h	RSCAN	RAM Test Register 84	RPGACC84	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 862Ah	RSCAN0	Transmit Buffer Register 2CH	TMDF12	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 862Ah	RSCAN	RAM Test Register 85	RPGACC85	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 862Ch	RSCAN0	Transmit Buffer Register 2DL	TMDF22	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 862Ch	RSCAN	RAM Test Register 86	RPGACC86	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 862Eh	RSCAN0	Transmit Buffer Register 2DH	TMDF32	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 862Eh	RSCAN	RAM Test Register 87	RPGACC87	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8630h	RSCAN0	Transmit Buffer Register 3AL	TMIDL3	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8630h	RSCAN	RAM Test Register 88	RPGACC88	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8632h	RSCAN0	Transmit Buffer Register 3AH	TMIDH3	16	16	2 or 3 PCLKB	2 ICLK	section 36.

Downloaded from Arrow.com.

Table 5.1 List of I/O Registers (Address Order) (29/31)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK <PCLK	
000A 8632h	RSCAN	RAM Test Register 89	RPGACC89	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8634h	RSCAN	RAM Test Register 90	RPGACC90	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8636h	RSCAN0	Transmit Buffer Register 3BH	TMPTR3	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8636h	RSCAN	RAM Test Register 91	RPGACC91	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8638h	RSCAN0	Transmit Buffer Register 3CL	TMDF03	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8638h	RSCAN	RAM Test Register 92	RPGACC92	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 863Ah	RSCAN0	Transmit Buffer Register 3CH	TMDF13	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 863Ah	RSCAN	RAM Test Register 93	RPGACC93	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 863Ch	RSCAN0	Transmit Buffer Register 3DL	TMDF23	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 863Ch	RSCAN	RAM Test Register 94	RPGACC94	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 863Eh	RSCAN0	Transmit Buffer Register 3DH	TMDF33	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 863Eh	RSCAN	RAM Test Register 95	RPGACC95	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8640h to 000A 867Eh	RSCAN	RAM Test Register 96 to RAM Test Register 127	RPGACC96 to RPGACC127	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000A 8680h	RSCAN0	Transmit History Buffer Access Register	THLACC0	16	16	2 or 3 PCLKB	2 ICLK	section 36.
000D 0A00h	MTU3	Timer Control Register	TCR	8	8	2 or 3 PCLKA	2 ICLK	section 23.
000D 0A01h	MTU4	Timer Control Register	TCR	8	8	2 or 3 PCLKA	2 ICLK	section 23.
000D 0A02h	MTU3	Timer Mode Register	TMDR	8	8	2 or 3 PCLKA	2 ICLK	section 23.
000D 0A03h	MTU4	Timer Mode Register	TMDR	8	8	2 or 3 PCLKA	2 ICLK	section 23.
000D 0A04h	MTU3	Timer I/O Control Register H	TIORH	8	8	2 or 3 PCLKA	2 ICLK	section 23.
000D 0A05h	MTU3	Timer I/O Control Register L	TIORL	8	8	2 or 3 PCLKA	2 ICLK	section 23.
000D 0A06h	MTU4	Timer I/O Control Register H	TIORH	8	8	2 or 3 PCLKA	2 ICLK	section 23.
000D 0A07h	MTU4	Timer I/O Control Register L	TIORL	8	8	2 or 3 PCLKA	2 ICLK	section 23.
000D 0A08h	MTU3	Timer Interrupt Enable Register	TIER	8	8	2 or 3 PCLKA	2 ICLK	section 23.
000D 0A09h	MTU4	Timer Interrupt Enable Register	TIER	8	8	2 or 3 PCLKA	2 ICLK	section 23.
000D 0A0Ah	MTU	Timer Output Master Enable Register	TOER	8	8	2 or 3 PCLKA	2 ICLK	section 23.
000D 0A0Dh	MTU	Timer Gate Control Register	TGCR	8	8	2 or 3 PCLKA	2 ICLK	section 23.
000D 0A0Eh	MTU	Timer Output Control Register 1	TOCR1	8	8	2 or 3 PCLKA	2 ICLK	section 23.
000D 0A0Fh	MTU	Timer Output Control Register 2	TOCR2	8	8	2 or 3 PCLKA	2 ICLK	section 23.
000D 0A10h	MTU3	Timer Counter	TCNT	16	16	2 or 3 PCLKA	2 ICLK	section 23.
000D 0A12h	MTU4	Timer Counter	TCNT	16	16	2 or 3 PCLKA	2 ICLK	section 23.
000D 0A14h	MTU	Timer Cycle Data Register	TCDR	16	16	2 or 3 PCLKA	2 ICLK	section 23.
000D 0A16h	MTU	Timer Dead Time Data Register	TDDR	16	16	2 or 3 PCLKA	2 ICLK	section 23.
000D 0A18h	MTU3	Timer General Register A	TGRA	16	16	2 or 3 PCLKA	2 ICLK	section 23.
000D 0A1Ah	MTU3	Timer General Register B	TGRB	16	16	2 or 3 PCLKA	2 ICLK	section 23.
000D 0A1Ch	MTU4	Timer General Register A	TGRA	16	16	2 or 3 PCLKA	2 ICLK	section 23.
000D 0A1Eh	MTU4	Timer General Register B	TGRB	16	16	2 or 3 PCLKA	2 ICLK	section 23.
000D 0A20h	MTU	Timer Subcounter	TCNTS	16	16	2 or 3 PCLKA	2 ICLK	section 23.
000D 0A22h	MTU	Timer Cycle Buffer Register	TCBR	16	16	2 or 3 PCLKA	2 ICLK	section 23.
000D 0A24h	MTU3	Timer General Register C	TGRC	16	16	2 or 3 PCLKA	2 ICLK	section 23.
000D 0A26h	MTU3	Timer General Register D	TGRD	16	16	2 or 3 PCLKA	2 ICLK	section 23.
000D 0A28h	MTU4	Timer General Register C	TGRC	16	16	2 or 3 PCLKA	2 ICLK	section 23.
000D 0A2Ah	MTU4	Timer General Register D	TGRD	16	16	2 or 3 PCLKA	2 ICLK	section 23.
000D 0A2Ch	MTU3	Timer Status Register	TSR	8	8	2 or 3 PCLKA	2 ICLK	section 23.
000D 0A2Dh	MTU4	Timer Status Register	TSR	8	8	2 or 3 PCLKA	2 ICLK	section 23.
000D 0A30h	MTU	Timer Interrupt Skipping Set Register	TITCR	8	8	2 or 3 PCLKA	2 ICLK	section 23.
000D 0A31h	MTU	Timer Interrupt Skipping Counter	TITCNT	8	8	2 or 3 PCLKA	2 ICLK	section 23.
000D 0A32h	MTU	Timer Buffer Transfer Set Register	TBTER	8	8	2 or 3 PCLKA	2 ICLK	section 23.
000D 0A34h	MTU	Timer Dead Time Enable Register	TDER	8	8	2 or 3 PCLKA	2 ICLK	section 23.
000D 0A36h	MTU	Timer Output Level Buffer Register	TOLBR	8	8	2 or 3 PCLKA	2 ICLK	section 23.
000D 0A38h	MTU3	Timer Buffer Operation Transfer Mode Register	TBTM	8	8	2 or 3 PCLKA	2 ICLK	section 23.
000D 0A39h	MTU4	Timer Buffer Operation Transfer Mode Register	TBTM	8	8	2 or 3 PCLKA	2 ICLK	section 23.

Downloaded from Arrow.com.

Table 5.1 List of I/O Registers (Address Order) (30/31)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK <PCLK	
000D 0A40h	MTU4	Timer A/D Converter Start Request Control Register	TADCR	16	16	2 or 3 PCLKA	2 ICLK	section 23.
000D 0A44h	MTU4	Timer A/D Converter Start Request Cycle Set Register A	TADCORA	16	16	2 or 3 PCLKA	2 ICLK	section 23.
000D 0A46h	MTU4	Timer A/D Converter Start Request Cycle Set Register B	TADCORB	16	16	2 or 3 PCLKA	2 ICLK	section 23.
000D 0A48h	MTU4	Timer A/D Converter Start Request Cycle Set Buffer Register A	TADCOBRA	16	16	2 or 3 PCLKA	2 ICLK	section 23.
000D 0A4Ah	MTU4	Timer A/D Converter Start Request Cycle Set Buffer Register B	TADCOBRB	16	16	2 or 3 PCLKA	2 ICLK	section 23.
000D 0A60h	MTU	Timer Waveform Control Register	TWCR	8	8, 16	2 or 3 PCLKA	2 ICLK	section 23.
000D 0A80h	MTU	Timer Start Register	TSTR	8	8, 16	2 or 3 PCLKA	2 ICLK	section 23.
000D 0A81h	MTU	Timer Synchronous Register	TSYR	8	8, 16	2 or 3 PCLKA	2 ICLK	section 23.
000D 0A84h	MTU	Timer Read/Write Enable Register	TRWER	8	8, 16	2 or 3 PCLKA	2 ICLK	section 23.
000D 0A90h	MTU0	Noise Filter Control Register	NFCR	8	8, 16	2 or 3 PCLKA	2 ICLK	section 23.
000D 0A91h	MTU1	Noise Filter Control Register	NFCR	8	8, 16	2 or 3 PCLKA	2 ICLK	section 23.
000D 0A92h	MTU2	Noise Filter Control Register	NFCR	8	8, 16	2 or 3 PCLKA	2 ICLK	section 23.
000D 0A93h	MTU3	Noise Filter Control Register	NFCR	8	8, 16	2 or 3 PCLKA	2 ICLK	section 23.
000D 0A94h	MTU4	Noise Filter Control Register	NFCR	8	8, 16	2 or 3 PCLKA	2 ICLK	section 23.
000D 0B00h	MTU0	Timer Control Register	TCR	8	8	2 or 3 PCLKA	2 ICLK	section 23.
000D 0B01h	MTU0	Timer Mode Register	TMDR	8	8	2 or 3 PCLKA	2 ICLK	section 23.
000D 0B02h	MTU0	Timer I/O Control Register H	TIORH	8	8	2 or 3 PCLKA	2 ICLK	section 23.
000D 0B03h	MTU0	Timer I/O Control Register L	TIORL	8	8	2 or 3 PCLKA	2 ICLK	section 23.
000D 0B04h	MTU0	Timer Interrupt Enable Register	TIER	8	8	2 or 3 PCLKA	2 ICLK	section 23.
000D 0B05h	MTU0	Timer Status Register	TSR	8	8	2 or 3 PCLKA	2 ICLK	section 23.
000D 0B06h	MTU0	Timer Counter	TCNT	16	16	2 or 3 PCLKA	2 ICLK	section 23.
000D 0B08h	MTU0	Timer General Register A	TGRA	16	16	2 or 3 PCLKA	2 ICLK	section 23.
000D 0B0Ah	MTU0	Timer General Register B	TGRB	16	16	2 or 3 PCLKA	2 ICLK	section 23.
000D 0B0Ch	MTU0	Timer General Register C	TGRC	16	16	2 or 3 PCLKA	2 ICLK	section 23.
000D 0B0Eh	MTU0	Timer General Register D	TGRD	16	16	2 or 3 PCLKA	2 ICLK	section 23.
000D 0B20h	MTU0	Timer General Register E	TGRE	16	16	2 or 3 PCLKA	2 ICLK	section 23.
000D 0B22h	MTU0	Timer General Register F	TGRF	16	16	2 or 3 PCLKA	2 ICLK	section 23.
000D 0B24h	MTU0	Timer Interrupt Enable Register 2	TIER2	8	8	2 or 3 PCLKA	2 ICLK	section 23.
000D 0B26h	MTU0	Timer Buffer Operation Transfer Mode Register	TBTM	8	8	2 or 3 PCLKA	2 ICLK	section 23.
000D 0B80h	MTU1	Timer Control Register	TCR	8	8	2 or 3 PCLKA	2 ICLK	section 23.
000D 0B81h	MTU1	Timer Mode Register	TMDR	8	8	2 or 3 PCLKA	2 ICLK	section 23.
000D 0B82h	MTU1	Timer I/O Control Register	TIOR	8	8	2 or 3 PCLKA	2 ICLK	section 23.
000D 0B84h	MTU1	Timer Interrupt Enable Register	TIER	8	8	2 or 3 PCLKA	2 ICLK	section 23.
000D 0B85h	MTU1	Timer Status Register	TSR	8	8	2 or 3 PCLKA	2 ICLK	section 23.
000D 0B86h	MTU1	Timer Counter	TCNT	16	16	2 or 3 PCLKA	2 ICLK	section 23.
000D 0B88h	MTU1	Timer General Register A	TGRA	16	16	2 or 3 PCLKA	2 ICLK	section 23.
000D 0B8Ah	MTU1	Timer General Register B	TGRB	16	16	2 or 3 PCLKA	2 ICLK	section 23.
000D 0B90h	MTU1	Timer Input Capture Control Register	TICCR	8	8	2 or 3 PCLKA	2 ICLK	section 23.
000D 0C00h	MTU2	Timer Control Register	TCR	8	8	2 or 3 PCLKA	2 ICLK	section 23.
000D 0C01h	MTU2	Timer Mode Register	TMDR	8	8	2 or 3 PCLKA	2 ICLK	section 23.
000D 0C02h	MTU2	Timer I/O Control Register	TIOR	8	8	2 or 3 PCLKA	2 ICLK	section 23.
000D 0C04h	MTU2	Timer Interrupt Enable Register	TIER	8	8	2 or 3 PCLKA	2 ICLK	section 23.
000D 0C05h	MTU2	Timer Status Register	TSR	8	8	2 or 3 PCLKA	2 ICLK	section 23.
000D 0C06h	MTU2	Timer Counter	TCNT	16	16	2 or 3 PCLKA	2 ICLK	section 23.
000D 0C08h	MTU2	Timer General Register A	TGRA	16	16	2 or 3 PCLKA	2 ICLK	section 23.
000D 0C0Ah	MTU2	Timer General Register B	TGRB	16	16	2 or 3 PCLKA	2 ICLK	section 23.
007F C090h	FLASH	E2 DataFlash Control Register	DFLCTL	8	8	2 or 3 FCLK	2 ICLK	section 50.
007F C0ACh	TEMPSA	Temperature Sensor Calibration Data Register L	TSCDRL	8	8	2 or 3 PCLKA	2ICLK	section 46.
007F C0ADh	TEMPSA	Temperature Sensor Calibration Data Register H	TSCDRH	8	8	2 or 3 PCLKA	2ICLK	section 46.
007F C100h	FLASH	Flash P/E Mode Control Register	FPMCR	8	8	2 or 3 FCLK	2 ICLK	section 50.
007F C104h	FLASH	Flash Area Select Register	FASR	8	8	2 or 3 FCLK	2 ICLK	section 50.

Downloaded from Arrow.com.

Table 5.1 List of I/O Registers (Address Order) (31/31)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK <PCLK	
007F C108h	FLASH	Flash Processing Start Address Register L	FSARL	16	16	2 or 3 FCLK	2 ICLK	section 50.
007F C110h	FLASH	Flash Processing Start Address Register H	FSARH	16	16	2 or 3 FCLK	2 ICLK	section 50.
007F C114h	FLASH	Flash Control Register	FCR	8	8	2 or 3 FCLK	2 ICLK	section 50.
007F C118h	FLASH	Flash Processing End Address Register L	FEARL	16	16	2 or 3 FCLK	2 ICLK	section 50.
007F C120h	FLASH	Flash Processing End Address Register H	FEARH	16	16	2 or 3 FCLK	2 ICLK	section 50.
007F C124h	FLASH	Flash Reset Register	FRESETR	8	8	2 or 3 FCLK	2 ICLK	section 50.
007F C12Ch	FLASH	Flash Status Register 1	FSTATR1	8	8	2 or 3 FCLK	2 ICLK	section 50.
007F C130h	FLASH	Flash Write Buffer Register 0	FWB0	16	16	2 or 3 FCLK	2 ICLK	section 50.
007F C138h	FLASH	Flash Write Buffer Register 1	FWB1	16	16	2 or 3 FCLK	2 ICLK	section 50.
007F C140h	FLASH	Flash Write Buffer Register 2	FWB2	16	16	2 or 3 FCLK	2 ICLK	section 50.
007F C144h	FLASH	Flash Write Buffer Register 3	FWB3	16	16	2 or 3 FCLK	2 ICLK	section 50.
007F C180h	FLASH	Protection Unlock Register	FPR	8	8	2 or 3 FCLK	2 ICLK	section 50.
007F C184h	FLASH	Protection Unlock Status Register	FPSR	8	8	2 or 3 FCLK	2 ICLK	section 50.
007F C1C0h	FLASH	Flash Start-Up Setting Monitor Register	FSCMR	16	16	2 or 3 FCLK	2 ICLK	section 50.
007F C1C8h	FLASH	Flash Access Window Start Address Monitor Register	FAWSMR	16	16	2 or 3 FCLK	2 ICLK	section 50.
007F C1D0h	FLASH	Flash Access Window End Address Monitor Register	FAWEMR	16	16	2 or 3 FCLK	2 ICLK	section 50.
007F C1D8h	FLASH	Flash Initial Setting Register	FISR	8	8	2 or 3 FCLK	2 ICLK	section 50.
007F C1DCh	FLASH	Flash Extra Area Control Register	FEXCR	8	8	2 or 3 FCLK	2 ICLK	section 50.
007F C1E0h	FLASH	Flash Error Address Monitor Register L	FEAML	16	16	2 or 3 FCLK	2 ICLK	section 50.
007F C1E8h	FLASH	Flash Error Address Monitor Register H	FEAMH	8	8	2 or 3 FCLK	2 ICLK	section 50.
007F C1F0h	FLASH	Flash Status Register 0	FSTATR0	8	8	2 or 3 FCLK	2 ICLK	section 50.
007F C350h	FLASHCONST	Unique ID Register 0	UIDR0	32	32	2 or 3 FCLK	2 ICLK	section 50.
007F C354h	FLASHCONST	Unique ID Register 1	UIDR1	32	32	2 or 3 FCLK	2 ICLK	section 50.
007F C358h	FLASHCONST	Unique ID Register 2	UIDR2	32	32	2 or 3 FCLK	2 ICLK	section 50.
007F C35Ch	FLASHCONST	Unique ID Register 3	UIDR3	32	32	2 or 3 FCLK	2 ICLK	section 50.
007F FFB2h	FLASH	Flash P/E Mode Entry Register	FENTRYR	16	16	2 or 3 FCLK	2 ICLK	section 50.

Note 1. Odd addresses cannot be accessed in 16-bit units. Table 26.4 lists register allocation for 16-bit access.
Note 2. When the register is accessed while the USB is operating, a delay may be generated in accessing.

Downloaded from Arrow.com.

6. Resets

6.1 Overview

The following resets are implemented: RES# pin reset, power-on reset, voltage monitoring 0 reset, voltage monitoring 1 reset, independent watchdog timer reset, watchdog timer reset, and software reset.
Table 6.1 lists the reset names and sources.

Table 6.1 Reset Names and Sources

Reset Name	Source
RES# pin reset	Voltage input to the RES# pin is driven low.
Power-on reset	VCC rises (voltage monitored: VPOR)*1
Voltage monitoring 0 reset	VCC falls (voltage monitored: Vdet0)*1
Voltage monitoring 1 reset	VCC falls (voltage monitored: Vdet1)*1
Independent watchdog timer reset	The independent watchdog timer underflows, or a refresh error occurs.
Watchdog timer reset	The watchdog timer underflows, or a refresh error occurs.
Software reset	Register setting

Note 1. For the voltages to be monitored (VPOR, Vdet0, and Vdet1), see section 8, Voltage Detection Circuit (LVDAb) and section 51, Electrical Characteristics.

Downloaded from Arrow.com.

The internal state and pins are initialized by a reset.
Table 6.2 lists the reset targets to be initialized.

Table 6.2 Targets Initialized by Each Reset Source

Target to be Initialized	Reset Source						
	RES# Pin Reset	Power-On Reset	Voltage Monitoring 0 Reset	Independent Watchdog Timer Reset	Watchdog Timer Reset	Voltage Monitoring 1 Reset	Software Reset
The power-on reset detect flag (RSTSR0.PORF)	✓	—	—	—	—	—	—
Register related to the cold start/warm start determination flag (RSTSR1.CWSF)	—*1	✓	—	—	—	—	—
Voltage monitoring 0 reset detect flag (RSTSR0.LVD0RF)	✓	✓	—	—	—	—	—
Registers related to the battery backup function (VBATTCR, VBATTSR, VBTLVDICR)	✓	✓	—	—	—	—	—
The independent watchdog timer reset detect flag (RSTSR2.IWDTRF)	✓	✓	✓	—	—	—	—
Registers related to the independent watchdog timer (IWDTRR, IWDTCR, IWDTSR, IWDTRCR, IWDTCSTPR, ILOCOCR)	✓	✓	✓	—	—	—	—
The watchdog timer reset detect flag (RSTSR2.WDTRF)	✓	✓	✓	✓	—	—	—
Registers related to the watchdog timer (WDTRR, WDTCR, WDTSR, WDTRCR)	✓	✓	✓	✓	—	—	—
The voltage monitoring 1 reset detect flag (RSTSR0.LVD1RF)	✓	✓	✓	✓	✓	—	—
Registers related to voltage monitor function 1 (LVD1CR0, LVCMPCR.LVD1E, LVDLVLR.LVD1LVL[3:0])	✓	✓	✓	✓	✓	—	—
(LVD1CR1, LVD1SR)	✓	✓	✓	✓	✓	—	—
The software reset detect flag (RSTSR2.SWRF)	✓	✓	✓	✓	✓	✓	—
Register related to the realtime clock*2	—	—	—	—	—	—	—
Registers other than the above, CPU, and internal state	✓	✓	✓	✓	✓	✓	✓

✓: Targets to be initialized, —: No change occurs.

Note 1. Initialized at a power-on.

Note 2. Some control bits (RCR1.CIE, RCR1.RTCOS, RCR2.RTCOE, ADJ30, and RESET) are initialized by all types of reset. For details on the target bits, refer to section 28, Realtime Clock (RTCe).

When a reset is canceled, the reset exception handling starts. For the reset exception handling, see section 14, Exception Handling.
Table 6.3 lists the pin related to the reset.

Table 6.3 Pin Related to Reset

Pin Name	I/O	Function
RES#	Input	Reset pin

Downloaded from Arrow.com.

6.2 Register Descriptions

6.2.1 Reset Status Register 0 (RSTSR0)

Address(es): 0008 C290h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	LVD1RF	LVD0RF	PORF
Value after reset:	0	0	0	0	0	0*1	0*1	0*1

Bit	Symbol	Bit Name	Description	R/W
b0	PORF	Power-On Reset Detect Flag	0: Power-on reset not detected. 1: Power-on reset detected.	R/(W) *2
b1	LVD0RF	Voltage Monitoring 0 Reset Detect Flag	0: Voltage monitoring 0 reset not detected. 1: Voltage monitoring 0 reset detected.	R/(W) *2
b2	LVD1RF	Voltage Monitoring 1 Reset Detect Flag	0: Voltage monitoring 1 reset not detected. 1: Voltage monitoring 1 reset detected.	R/(W) *2
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. The value after reset depends on the reset source.
Note 2. Only 0 can be written to clear the flag.

PORF Flag (Power-On Reset Detect Flag)

The PORF flag indicates that a power-on reset has occurred.
[Setting condition]

- When a power-on reset occurs.

[Clearing conditions]

- When resets shown in Table 6.2 occur.
- When PORF is read as 1 and then 0 is written to PORF.

LVD0RF Flag (Voltage Monitoring 0 Reset Detect Flag)

The LVD0RF flag indicates that VCC voltage has fallen below Vdet0.
[Setting condition]

- When Vdet0-level VCC voltage is detected.

[Clearing conditions]

- When resets listed in Table 6.2 occur.
- When LVD0RF is read as 1 and then 0 is written to LVD0RF.

LVD1RF Flag (Voltage Monitoring 1 Reset Detect Flag)

The LVD1RF flag indicates that VCC voltage has fallen below Vdet1.
[Setting condition]

- When Vdet1-level VCC voltage is detected.

[Clearing conditions]

- When a reset listed in Table 6.2 occurs.
- When LVD1RF is read as 1 and then 0 is written to LVD1RF.

Downloaded from Arrow.com.

6.2.2 Reset Status Register 1 (RSTSR1)

Address(es): 0008 C291h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	CWSF
Value after reset:	0	0	0	0	0	0	0	0/1*1

Bit	Symbol	Bit Name	Description	R/W
b0	CWSF	Cold/Warm Start Determination Flag	0: Cold start 1: Warm start	R/(W) *2
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. The value after reset depends on the reset source.
Note 2. Only 1 can be written to set the flag.

RSTSR1 determines whether a power-on reset has caused the reset processing (cold start) or a reset signal input during operation has caused the reset processing (warm start).

CWSF Flag (Cold/Warm Start Determination Flag)

The CWSF flag indicates the type of reset processing: cold start or warm start.
The CWSF flag is initialized at a power-on.
[Setting condition]

- When 1 is written through programming; it is not set to 0 even when 0 is written.

[Clearing condition]

- When a reset listed in Table 6.2 occurs.

Downloaded from Arrow.com.

6.2.3 Reset Status Register 2 (RSTSR2)

Address(es): 0008 00C0h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	SWRF	WDTRF	IWDTRF
Value after reset:	0	0	0	0	0	0*1	0*1	0*1

Bit	Symbol	Bit Name	Description	R/W
b0	IWDTRF	Independent Watchdog Timer Reset Detect Flag	0: Independent watchdog timer reset not detected. 1: Independent watchdog timer reset detected.	R/(W) *2
b1	WDTRF	Watchdog Timer Reset Detect Flag	0: Watchdog timer reset not detected. 1: Watchdog timer reset detected.	R/(W) *2
b2	SWRF	Software Reset Detect Flag	0: Software reset not detected. 1: Software reset detected.	R/(W) *2
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. The value after reset depends on the reset source.
Note 2. Only 0 can be written to clear the flag.

IWDTRF Flag (Independent Watchdog Timer Reset Detect Flag)

The IWDTRF flag indicates that an independent watchdog timer reset has occurred.
[Setting condition]

- When an independent watchdog timer reset occurs.

[Clearing conditions]

- When a reset listed in Table 6.2 occurs.
- When IWDTRF is read as 1 and then 0 is written to IWDTRF.

WDTRF Flag (Watchdog Timer Reset Detect Flag)

The WDTRF flag indicates that a watchdog timer reset has occurred.
[Setting condition]

- When a watchdog timer reset occurs.

[Clearing conditions]

- When a reset listed in Table 6.2 occurs.
- When WDTRF is read as 1 and then 0 is written to WDTRF.

SWRF Flag (Software Reset Detect Flag)

The SWRF flag indicates that a software reset has occurred.
[Setting condition]

- When a software reset occurs.

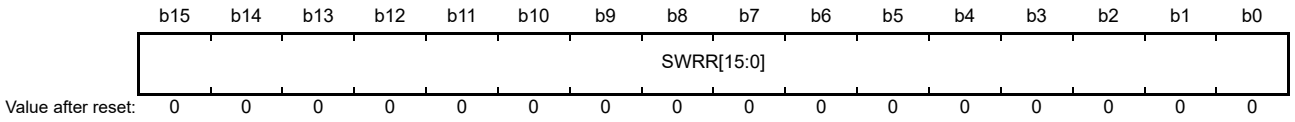
[Clearing conditions]

- When a reset listed in Table 6.2 occurs.
- When SWRF is read as 1 and then 0 is written to SWRF.

Downloaded from Arrow.com.

6.2.4 Software Reset Register (SWRR)

Address(es): 0008 00C2h



Bit	Symbol	Bit Name	Description	R/W
b15 to b0	SWRR[15:0]	Software Reset	Writing A501h resets the MCU. These bits are read as 0000h.	R/W

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

Downloaded from Arrow.com.

6.3 Operation

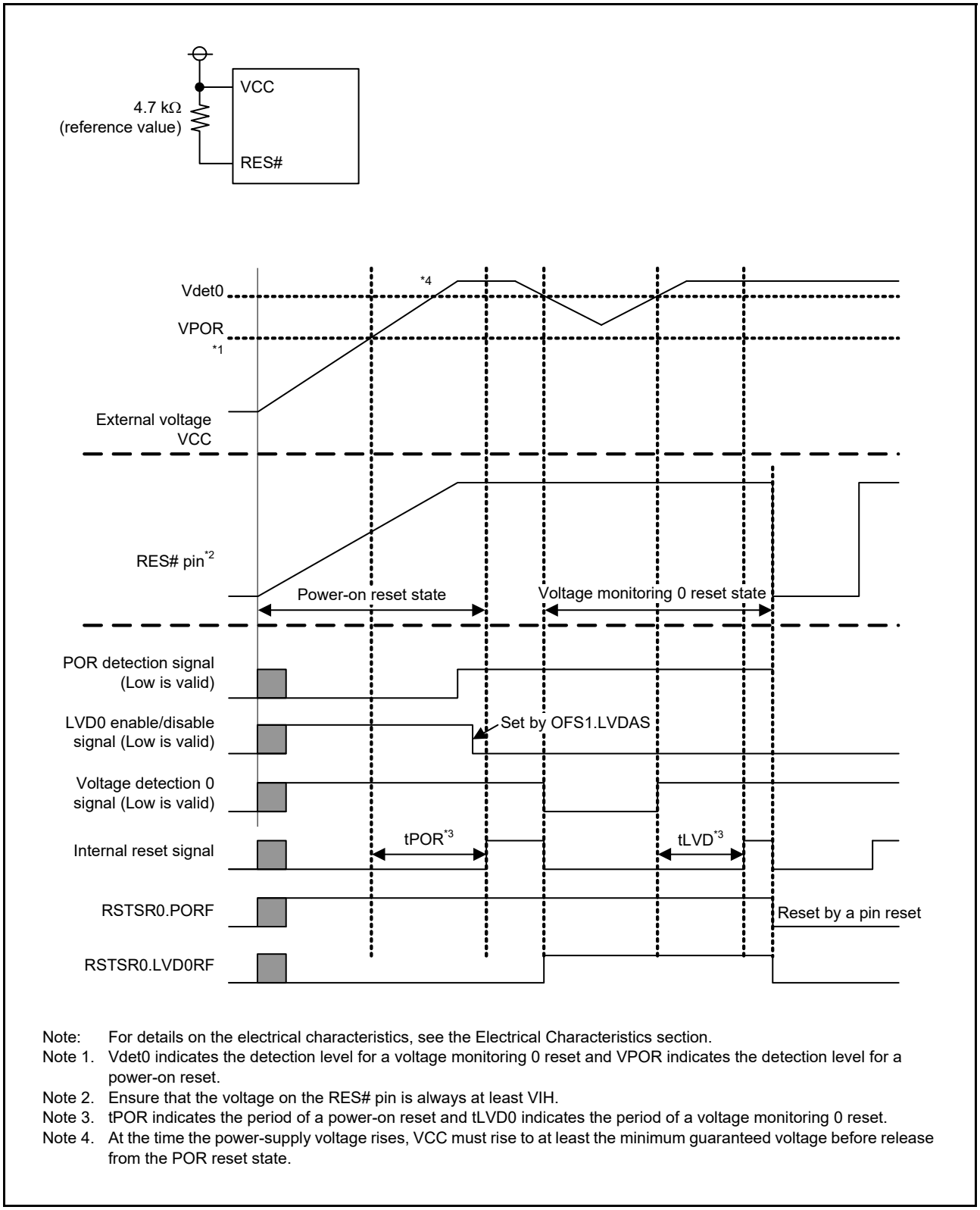
6.3.1 RES# Pin Reset

This is a reset generated by the RES# pin.
When the RES# pin is driven low, all the processing in progress is aborted and the LSI enters a reset state.
In order to unfailingly reset the LSI, the RES# pin should be held low for the specified power supply stabilization time at a power-on.
When the RES# pin is driven high from low, the internal reset is canceled after the post-RES# cancellation wait time (tRESWT) has elapsed, and then the CPU starts the reset exception handling.
For details, see section 51, Electrical Characteristics.

6.3.2 Power-On Reset and Voltage Monitoring 0 Reset

The power-on reset is an internal reset generated by the power-on reset circuit. A power-on reset is generated when power is supplied to the RES# pin while it is connected to VCC via a resistor. When connecting a capacitor to the RES# pin, also ensure that the voltage on the RES# pin is always at least VIH. For details on VIH, refer to section 51, Electrical Characteristics. After VCC has exceeded VPOR and the specified period (power-on reset time) has elapsed, the internal reset is canceled and the CPU starts the reset exception handling. The power-on reset time is a stabilization period for the external power supply and the MCU circuit. After a power-on reset has been generated, the PORF flag in RSTSR0 is set to 1. The PORF flag is initialized by RES# pin reset.
The voltage monitoring 0 reset is an internal reset generated by the voltage monitoring circuit. If the voltage detection circuit 0 start bit (LVDAS) in option function select register 1 (OFS1) is 0 (voltage monitoring 0 reset is enabled after a reset) and VCC falls below Vdet0, the RSTSR0.LVD0RF flag becomes 1 and the voltage detection circuit generates voltage monitoring 0 reset. Clear the OFS1.LVDAS bit to 0 if the voltage monitoring 0 reset is to be used.
Release from the voltage monitoring 0 reset state occurs when VCC rises above Vdet0 and the LVD0 reset time (tLVD0) elapses, and then the CPU starts the reset exception handling.
Figure 6.1 shows operations during a power-on reset and voltage monitoring 0 reset.
For details on voltage monitoring 0 reset, refer to section 8, Voltage Detection Circuit (LVDAb).

Downloaded from Arrow.com.



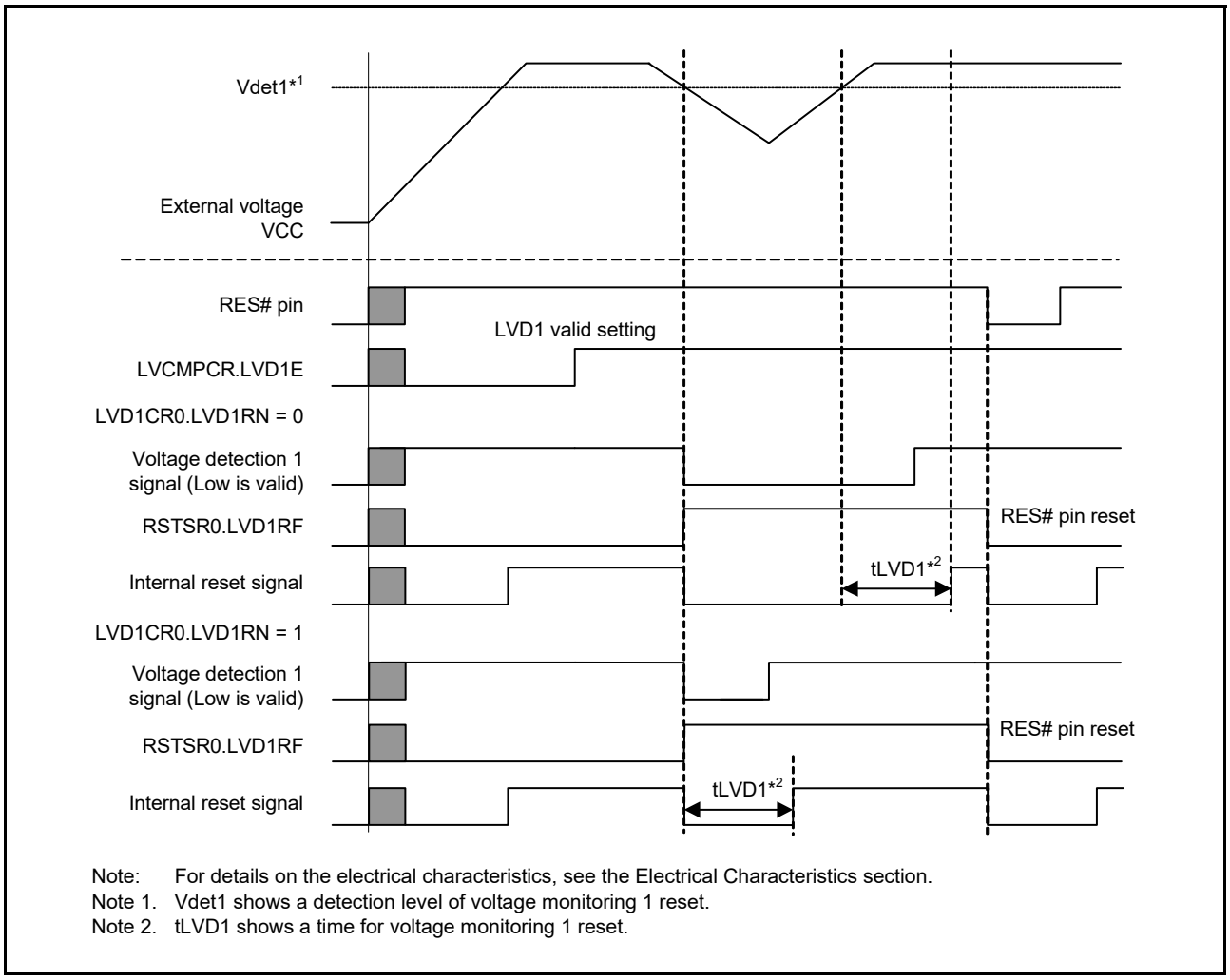
Note: For details on the electrical characteristics, see the Electrical Characteristics section.
Note 1. Vdet0 indicates the detection level for a voltage monitoring 0 reset and VPOR indicates the detection level for a power-on reset.
Note 2. Ensure that the voltage on the RES# pin is always at least VIH.
Note 3. tPOR indicates the period of a power-on reset and tLVD0 indicates the period of a voltage monitoring 0 reset.
Note 4. At the time the power-supply voltage rises, VCC must rise to at least the minimum guaranteed voltage before release from the POR reset state.

Figure 6.1 Operation Examples During a Power-On Reset and Voltage Monitoring 0 Reset

Downloaded from Arrow.com.

6.3.3 Voltage Monitoring 1 Reset

The voltage monitoring 1 reset is internal resets generated by the voltage monitoring circuit.
When the voltage monitoring 1 interrupt/reset enable bit (LVD1RIE) is set to 1 (enabling generation of a reset or interrupt by the voltage detection circuit) and the voltage monitoring 1 circuit mode select bit (LVD1RI) is set to 1 (selecting generation of a reset in response to detection of a low voltage) in the voltage monitoring 1 circuit control register 0 (LVD1CR0), the RSTSR0.LVD1RF flag is set to 1 and the voltage-detection circuit generates a voltage monitoring 1 reset if VCC falls to or below Vdet1.
Timing for release from the voltage monitoring 1 reset state is selectable with the voltage monitoring 1 reset negation select bit (LVD1RN) in the LVD1CR0 register. When the LVD1CR0.LVD1RN bit is 0 and VCC has fallen to or below Vdet1, the CPU is released from the internal reset state and starts reset exception handling once the voltage monitoring 1 reset time (tLVD1) has elapsed after VCC has risen above Vdet1. When the LVD1CR0.LVD1RN bit is 1 and VCC has fallen to or below Vdet1, the CPU is released from the internal reset state and starts reset exception handling once the voltage monitoring 1 reset time (tLVD1) has elapsed.
Detection levels Vdet1 can be changed by settings in the voltage detection level select register (LVDLVLR).
Figure 6.2 shows examples of operations during voltage monitoring 1 reset.
For details on the voltage monitoring 1 reset, refer to section 8, Voltage Detection Circuit (LVDAb).



Note: For details on the electrical characteristics, see the Electrical Characteristics section.
Note 1. Vdet1 shows a detection level of voltage monitoring 1 reset.
Note 2. tLVD1 shows a time for voltage monitoring 1 reset.

Figure 6.2 Operation Examples During Voltage Monitoring 1 Reset

Downloaded from Arrow.com.

6.3.4 Independent Watchdog Timer Reset

Independent watchdog timer reset is an internal reset generated by the independent watchdog timer.
Output of the independent watchdog timer reset from the independent watchdog timer can be selected by setting the IWDT reset control register (IWDTRCR) and option function select register 0 (OFS0).
When output of the independent watchdog timer reset is selected, an independent watchdog timer reset is generated if the independent watchdog timer underflows, or if data is written outside the refresh-permitted period. When the internal reset time (tRESW2) has elapsed after the independent watchdog timer reset has been generated, the internal reset is canceled and the CPU starts the reset exception handling.
For details on the independent watchdog timer reset, see section 31, Independent Watchdog Timer (IWDTa).

6.3.5 Watchdog Timer Reset

The watchdog-timer reset is an internal reset from the watchdog timer.
Output of the independent watchdog timer reset from the independent watchdog timer can be selected by setting the WDT reset control register (WDTRCR) and option function select register 0 (OFS0).
When output of the independent watchdog timer reset is selected, a watchdog timer reset is generated if the watchdog timer underflows, or if data is written outside the refresh-permitted period. When the internal reset time (tRESW2) has elapsed after the watchdog timer reset is generated, the internal reset is canceled and the CPU starts the reset exception handling.
For details, see section 30, Watchdog Timer (WDTA).

6.3.6 Software Reset

The software reset is an internal reset generated by the software reset circuit.
When A501h is written to SWRR, a software reset is generated. When the internal reset time (tRESW2) has elapsed after the software reset is generated, the internal reset is canceled and the CPU starts the reset exception handling.

Downloaded from Arrow.com.

6.3.7 Determination of Cold/Warm Start

By reading the CWSF flag in RSTSR1, the type of reset processing caused can be identified; that is, whether a power-on reset has caused the reset processing (cold start) or a reset signal input during operation has caused the reset processing (warm start).

The CWSF flag in RSTSR1 is set to 0 when a power-on reset occurs (cold start); otherwise the flag is not set to 0. The flag is set to 1 when 1 is written to it through programming; it is not set to 0 even when 0 is written.

Figure 6.3 shows an example of cold/warm start determination operation.

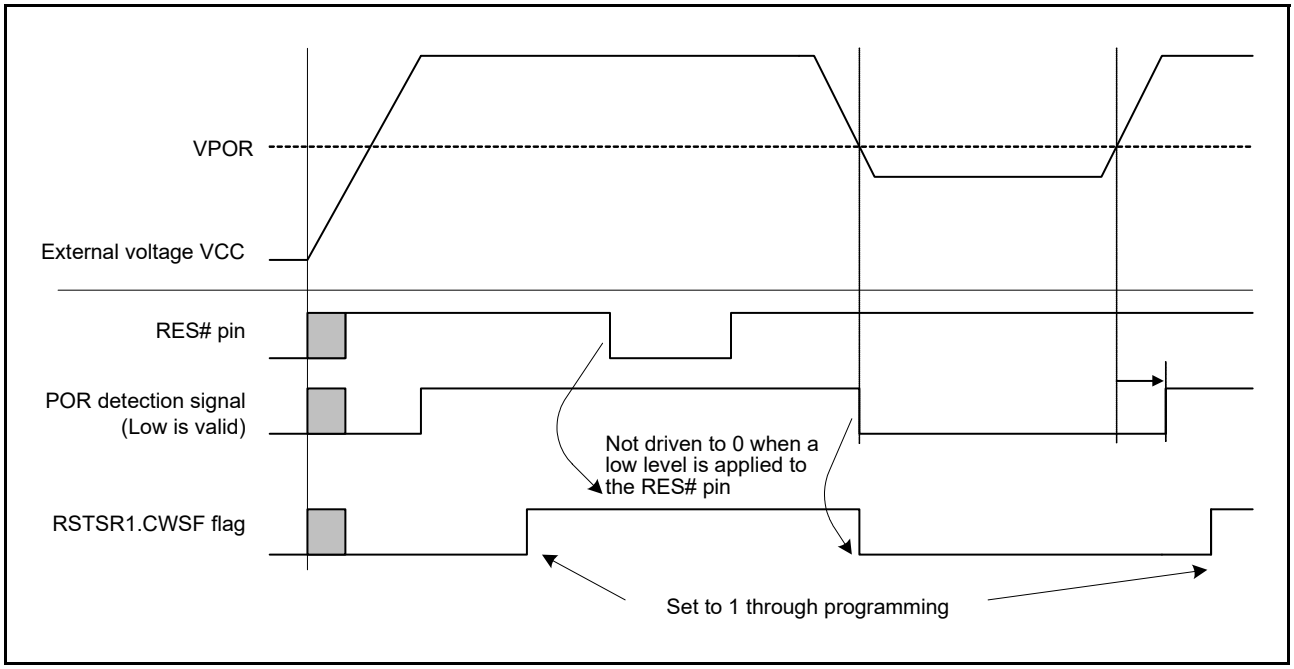


Figure 6.3 Example of Cold/Warm Start Determination Operation

Downloaded from Arrow.com.

6.3.8 Determination of Reset Generation Source

Reading RSTSR0 and RSTSR2 determines which reset was used to execute the reset exception handling.
Figure 6.4 shows an example of the flow to identify a reset generation source.

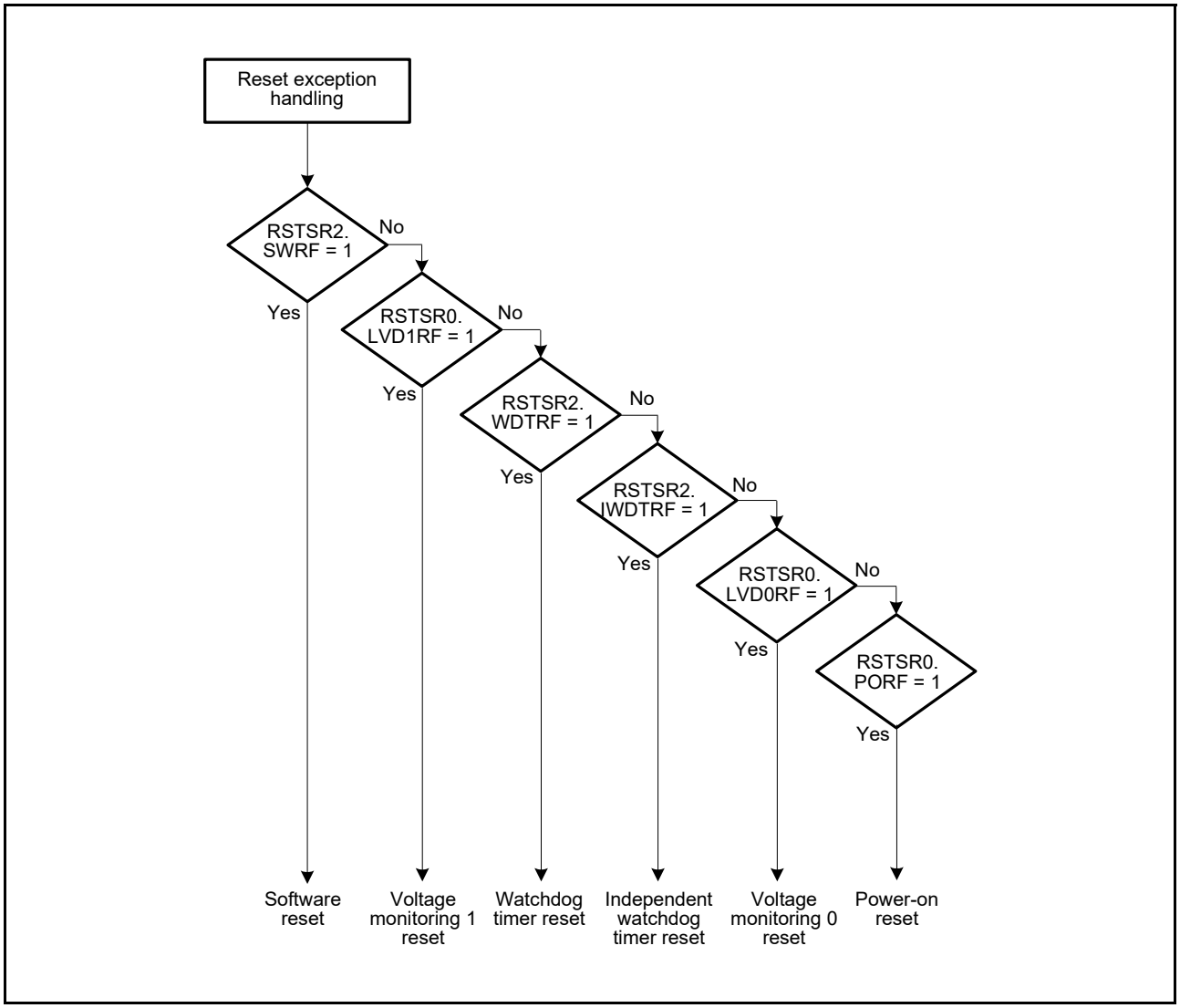


Figure 6.4 Example of Reset Generation Source Determination Flow

Downloaded from Arrow.com.

7. Option-Setting Memory (OFSM)

7.1 Overview

Option-setting memory (OFSM) refers to a set of registers that are provided for selecting the state of the microcontroller after a reset. The option-setting memory is allocated in the ROM.

Figure 7.1 shows the option-setting memory area.

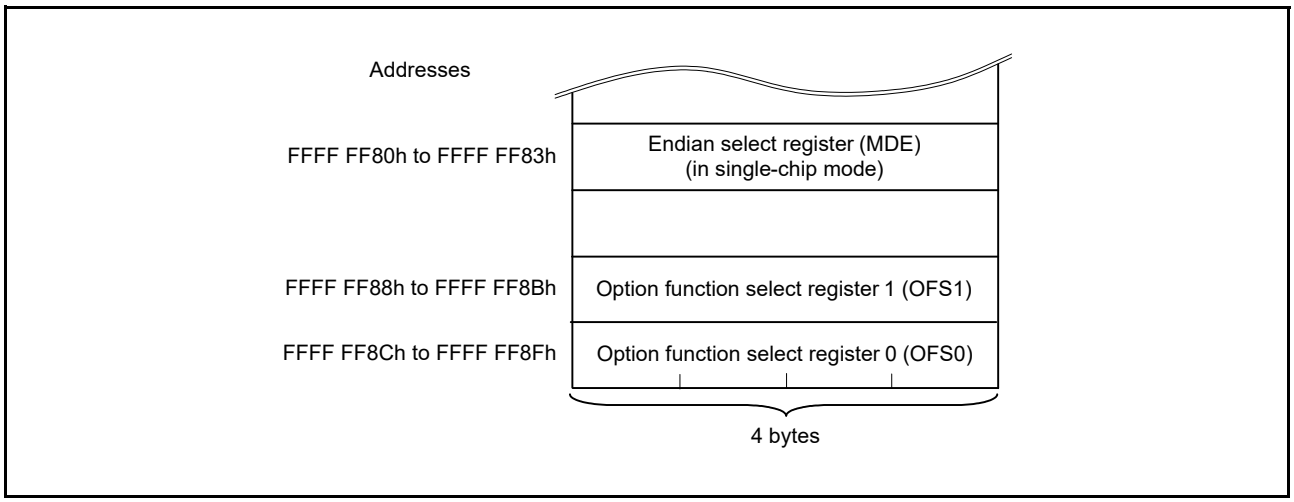


Figure 7.1 Option-Setting Memory Area

Downloaded from Arrow.com.

7.2 Register Descriptions

7.2.1 Option Function Select Register 0 (OFS0)

Address(es): OFSM.OFS0 FFFF FF8Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	WDTRSTIRQS	WDTRPSS[1:0]		WDTRPES[1:0]		WDTCKS[3:0]				WDTTOPS[1:0]		WDTSTRT	—
Value after reset:	The value set by the user*1															

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	IWDTSLCSTP	—	IWDTRSTIRQS	IWDTRPSS[1:0]		IWDTRPES[1:0]		IWDTCKS[3:0]				IWDTTOPS[1:0]		IWDTSTRT	—
Value after reset:	The value set by the user*1															

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	When reading, this bit returns the value written by the user. The write value should be 1.	R
b1	IWDTSTRT	IWDT Start Mode Select	0: IWDT is automatically activated in auto-start mode after a reset 1: IWDT is halted after a reset	R
b3, b2	IWDTTOPS[1:0]	IWDT Timeout Period Select	b3 b2 0 0: 128 cycles (007Fh) 0 1: 512 cycles (01FFh) 1 0: 1024 cycles (03FFh) 1 1: 2048 cycles (07FFh)	R
b7 to b4	IWDTCKS[3:0]	IWDT Clock Frequency Division Ratio Select	b7 b4 0 0 0 0: No division 0 0 1 0: Divide-by-16 0 0 1 1: Divide-by-32 0 1 0 0: Divide-by-64 1 1 1 1: Divide-by-128 0 1 0 1: Divide-by-256 Settings other than above are prohibited.	R
b9, b8	IWDTRPES[1:0]	IWDT Window End Position Select	b9 b8 0 0: 75% 0 1: 50% 1 0: 25% 1 1: 0% (No window end position setting)	R
b11, b10	IWDTRPSS[1:0]	IWDT Window Start Position Select	b11 b10 0 0: 25% 0 1: 50% 1 0: 75% 1 1: 100% (No window start position setting)	R
b12	IWDTRSTIRQS	IWDT Reset Interrupt Request Select	0: Non-maskable interrupt request is enabled 1: Reset is enabled	R
b13	—	Reserved	When reading, this bit returns the value written by the user. The write value should be 1.	R
b14	IWDTSLCSTP	IWDT Sleep Mode Count Stop Control	0: Counting stop is disabled 1: Counting stop is enabled when entering sleep, software standby, or deep sleep mode	R
b16, b15	—	Reserved	When reading, these bits return the value written by the user. The write value should be 1.	R
b17	WDTSTRT	WDT Start Mode Select	0: WDT is automatically activated in auto-start mode after a reset 1: WDT is stopped after a reset	R

Downloaded from Arrow.com.

Bit	Symbol	Bit Name	Description	R/W
b19, b18	WDTTOPS[1:0]	WDT Timeout Period Select	b19 b18 0 0: 1024 cycles (03FFh) 0 1: 4096 cycles (0FFFh) 1 0: 8192 cycles (1FFFh) 1 1: 16384 cycles (3FFFh)	R
b23 to b20	WDTCKS[3:0]	WDT Clock Frequency Division Ratio Select	b23 b20 0 0 0 1: Divide-by-4 0 1 0 0: Divide-by-64 1 1 1 1: Divide-by-128 0 1 1 0: Divide-by-512 0 1 1 1: Divide-by-2048 1 0 0 0: Divide-by-8192 Settings other than above are prohibited.	R
b25, b24	WDTRPES[1:0]	WDT Window End Position Select	b25 b24 0 0: 75% 0 1: 50% 1 0: 25% 1 1: 0% (No window end position setting)	R
b27, b26	WDTRPSS[1:0]	WDT Window Start Position Select	b27 b26 0 0: 25% 0 1: 50% 1 0: 75% 1 1: 100% (No window start position setting)	R
b28	WDTRSTIRQS	WDT Reset Interrupt Request Select	0: Non-maskable interrupt request is enabled 1: Reset is enabled	R
b31 to b29	—	Reserved	When reading, these bits return the value written by the user. The write value should be 1.	R

Note 1. The value of the blank product is FFFF FFFFh. This register is set to a specified value after programming of the flash memory with the user program.

The OFS0 register is allocated in the ROM. Set this register at the same time as writing the program. After writing to the OFS0 register once, do not write to it again.
When erasing the block including the OFS0 register, the OFS0 register value becomes FFFF FFFFh.
The setting in the OFS0 register is ignored in boot mode, and this register functions similarly when it is set to FFFF FFFFh.

IWDTSTRT Bit (IWDT Start Mode Select)

This bit selects the mode in which the IWDT is activated after a reset (stopped state or activated in auto-start mode). When activated in auto-start mode, the OFS0 register setting for the IWDT is effective.

IWDTTOPS[1:0] Bits (IWDT Timeout Period Select)

These bits select the timeout period, i.e. the time it takes for the down-counter to underflow, as 128, 512, 1024, or 2048 cycles of the frequency-divided clock set by the IWDTCKS[3:0] bits. The time (number of IWDT-dedicated clock cycles) it takes to underflow after a refresh operation is determined by the combination of the IWDTCKS[3:0] bits and IWDTTOPS[1:0] bits.
For details, see section 31, Independent Watchdog Timer (IWDTa).

IWDTCKS[3:0] Bits (IWDT Clock Frequency Division Ratio Select)

These bits select, from 1/1, 1/16, 1/32, 1/64, 1/128, and 1/256, the division ratio of the prescaler to divide the frequency of the IWDT-dedicated clock. Using the setting of these bits together with the IWDTTOPS[1:0] bit setting, the IWDT counting period can be set from 128 to 524288 IWDT-dedicated clock cycles.
For details, see section 31, Independent Watchdog Timer (IWDTa).

IWDTRPES[1:0] Bits (IWDT Window End Position Select)

These bits select the position of the end of the window for the down-counter as 0%, 25%, 50%, or 75% of the value being

Downloaded from Arrow.com.

counted by the counter. The value of the window end position must be smaller than the value of the window start position (window start position > window end position). If the value for the window end position is greater than the value for the window start position, only the value for the window start position is effective.
The counter values corresponding to the settings for the start and end positions of the window in the IWDTRPSS[1:0] and IWDTRPES[1:0] bits vary with the setting of the IWDTTOPS[1:0] bits.
For details, refer to section 31, Independent Watchdog Timer (IWDTa).

IWDTRPSS[1:0] Bits (IWDT Window Start Position Select)

These bits select the position where the window for the down-counter starts as 25%, 50%, 75%, or 100% of the value being counted (the point at which counting starts is 100% and the point at which an underflow occurs is 0%). The interval between the positions where the window starts and ends becomes the period in which refreshing is possible, and refreshing is not possible outside this period.
For details, refer to section 31, Independent Watchdog Timer (IWDTa).

IWDTRSTIRQS Bit (IWDT Reset Interrupt Request Select)

The setting of this bit selects the operation on an underflow of the down-counter or generation of a refresh error. Either an independent watchdog timer reset or a non-maskable interrupt request is selectable.
For details, refer to section 31, Independent Watchdog Timer (IWDTa).

IWDTSLCSTP Bit (IWDT Sleep Mode Count Stop Control)

This bit selects whether to stop counting when entering sleep, software standby, or deep sleep mode.
For details, see section 31, Independent Watchdog Timer (IWDTa).

WDTSTRT Bit (WDT Start Mode Select)

This bit selects the mode in which the WDT is activated after a reset (stopped state or activated in auto-start mode). When activated in auto-start mode, the OFS0 register setting for the WDT is effective.

WDTTOPS[1:0] Bits (WDT Timeout Period Select)

These bits select the timeout period, i.e. the time it takes for the down-counter to underflow, as 1024, 4096, 8192, or 16384 cycles of the frequency-divided clock set by the WDTCKS[3:0] bits. The time (number of PCLKB cycles) it takes to underflow after a refresh operation is determined by a combination of the WDTCKS[3:0] bits and WDTTOPS[1:0] bits.
For details, see section 30, Watchdog Timer (WDTA).

WDTCKS[3:0] Bits (WDT Clock Frequency Division Ratio Select)

These bits select, from 1/4, 1/64, 1/128, 1/512, 1/2048, and 1/8192, the division ratio of the prescaler to divide the frequency of PCLKB. Using the setting of these bits together with the WDTTOPS[1:0] bit setting, the WDT counting period can be set from 4096 to 134217728 PCLKB cycles.
For details, see section 30, Watchdog Timer (WDTA).

WDTRPES[1:0] Bits (WDT Window End Position Select)

These bits select the position of the end of the window on the down-counter as 0%, 25%, 50%, or 75% of the value being counted by the counter. The value of the window end position must be smaller than the value of the window start position (window start position > window end position). If the value for the window end position is greater than the value for the window start position, only the value for the window start position is effective.
The counter values corresponding to the settings for the start and end positions of the window in the WDTRPSS[1:0] and WDTRPES[1:0] bits vary with the setting of the WDTTOPS[1:0] bits.
For details, refer to section 30, Watchdog Timer (WDTA).

Downloaded from Arrow.com.

WDTRPSS[1:0] Bits (WDT Window Start Position Select)

These bits select the position where the window for the down-counter starts as 25%, 50%, 75%, or 100% of the value being counted (the point at which counting starts is 100% and the point at which an underflow occurs is 0%). The interval between the positions where the window starts and ends becomes the period in which refreshing is possible, and refreshing is not possible outside this period.
For details, refer to section 30, Watchdog Timer (WDTA).

WDTRSTIRQS Bit (WDT Reset Interrupt Request Select)

The setting of this bit selects the operation on an underflow of the down-counter or generation of a refresh error. Either a watchdog timer reset or a non-maskable interrupt request is selectable.
For details, refer to section 30, Watchdog Timer (WDTA).

Downloaded from Arrow.com.

7.2.2 Option Function Select Register 1 (OFS1)

Address(es): OFSM.OFS1 FFFF FF88h, (FFFF 7F88h)*1

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	The value set by the user*2															

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	—	—	—	—	—	—	—	HOCOEN	—	—	—	—	FASTSTUP	LVDAS	VDSEL[1:0]	
	The value set by the user*2															

Bit	Symbol	Bit Name	Description	R/W
b1, b0	VDSEL[1:0]	Voltage Detection 0 Level Select	b1 b0 0 0: Settings prohibited 0 1: 2.82 V is selected 1 0: 2.51 V is selected 1 1: 1.90 V is selected	R
b2	LVDAS	Voltage Detection 0 Circuit Start	0: Voltage monitoring 0 reset is enabled after a reset 1: Voltage monitoring 0 reset is disabled after a reset	R
b3	FASTSTUP	Power-On Fast Startup Time	0: Fast startup time at power on 1: Normal startup	R
b7 to b4	—	Reserved	When reading, these bits return the value written by the user. The write value should be 1.	R
b8	HOCOEN	HOCO Oscillation Enable	0: HOCO oscillation is enabled after a reset 1: HOCO oscillation is disabled after a reset	R
b31 to b9	—	Reserved	When reading, these bits return the value written by the user. The write value should be 1.	R

Note 1. Set the same value to this address when using start-up program protection.
Note 2. The value of the blank product is FFFF FFFFh. This register is set to a specified value after programming of the flash memory with the user program.

The OFS1 register is allocated in the ROM. Set this register at the same time as writing the program. After writing, do not write additions to this register.
When erasing the block including the OFS1 register, the OFS1 register value becomes FFFF FFFFh.
The setting in the OFS1 register is ignored in boot mode, and this register functions similarly when it is set to FFFF FFFFh.

VDSEL[1:0] Bits (Voltage Detection 0 Level Select)

These bits select the voltage detection level to be monitored by the voltage detection 0 circuit.
Set the voltage detection 0 level to 2.51 V (set VDSEL[1:0] bits to 10b) before using the battery backup function (setting the VBATTCR.VBATTDIS bit to 0).

LVDAS Bit (Voltage Detection 0 Circuit Start)

This bit selects whether the voltage monitoring 0 reset is enabled or disabled after a reset.
The Vdet0 voltage to be monitored by the voltage detection 0 circuit is selected by the VDSEL[1:0] bits.

FASTSTUP Bit (Power-On Fast Startup Time)

The startup time can be reduced by setting this bit to 0 (fast startup time at power on) when it is possible to meet the power-on VCC rising gradient (during fast startup time) shown in Electrical Characteristics. Do not set this bit to 0 when it is not possible to meet the power-on VCC rising gradient (during fast startup time).

Downloaded from Arrow.com.

HOCOEN Bit (HOCO Oscillation Enable)

This bit selects whether the HOCO oscillation is effective or not after a reset.
Setting the HOCOEN bit to 0 allows the HOCO oscillation to be started before the CPU starts operation, and therefore reduces the wait time for oscillation stabilization.
Note that even if the HOCOEN bit is set to 0, the system clock source is not switched to HOCO. The system clock source is switched to HOCO only by modifying the clock source select bits (SCKCR3.CKSEL[2:0]) from the CPU.
Also, when the HOCOEN bit is set to 0, the HOCO oscillation stabilization time (tHOCO) is secured by hardware, so the clock with the accuracy of the HOCO oscillation frequency (fHOCO) shown in Electrical Characteristics is supplied after release from the CPU reset state.

7.2.3 Endian Select Register (MDE)

Address(es): OFSM.MDE FFFF FF80h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	The value set by the user*1															

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	MDE[2:0]		
Value after reset:	The value set by the user*1															

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	MDE[2:0]	Endian Select	b2 b0 0 0 0: Big endian 1 1 1: Little endian Settings other than above are prohibited.	R
b31 to b3	—	Reserved	When reading, these bits return the value written by the user. The write value should be 1.	R

Note 1. The value of the blank product is FFFF FFFFh. This register is set to a specified value after programming of the flash memory with the user program.

The MDE register selects the endian for the CPU. In single-chip mode, the endian select register (MDE) at address FFFF FF80h is used to select the endian.
MDE is allocated in the ROM. Set the register at the same time as writing the program. After writing to the register once, do not write to it again.
When erasing the block including the MDE register, the MDE register value becomes FFFF FFFFh.

MDE[2:0] Bits (Endian Select)

These bits select little endian or big endian for the CPU.
The endian is determined by the value at address FFFF FF80h in the user area when operating in single-chip mode.

Downloaded from Arrow.com.

7.3 Usage Note

7.3.1 Setting Example of Option-Setting Memory

Since the option-setting memory is allocated in the ROM, values cannot be written by executing instructions. Write appropriate values when writing the program. Examples of the settings are shown below.

- To set FFFF FFF8h in the OFS0 register

```
.ORG    0FFFFFF8CH
.LWORD  0FFFFFFF8H
```

- To set FFFF FEF0h in the OFS1 register

```
.ORG    0FFFFFF88H
.LWORD  0FFFFFEF0H
.ORG    0FFFF7F88H
.LWORD  0FFFFFEF0H
```

When neither the voltage monitoring 0 reset nor power-on fast startup time is used, the value set in the address FFFF 7F88h can be FFFF FFFFh.

Note: Programming formats vary depending on the compiler. Refer to the compiler manual for details.

Downloaded from Arrow.com.

8. Voltage Detection Circuit (LVDAb)

The voltage detection circuit (LVD) monitors the voltage level input to the VCC pin using a program.

8.1 Overview

In voltage detection 0, the detection voltage can be selected from three levels using option function select register 1 (OFS1).
In voltage detection 1, the detection voltage can be selected from 10 levels using the voltage detection level select register (LVDLVLR).
Voltage monitoring 0 reset and voltage monitoring 1 reset/interrupt can be used.
Table 8.1 lists the specifications of the voltage detection circuit. Figure 8.1 is a block diagram of the voltage detection circuit. Figure 8.2 is a block diagram of the voltage monitoring 1 interrupt/reset circuit.

Table 8.1 LVD Specifications

Item		Voltage Monitoring 0	Voltage Monitoring 1
VCC monitoring	Monitored voltage	Vdet0	Vdet1
	Detection voltage	Voltage selectable from four levels using OFS1	Voltage selectable from 10 levels using the LVDLVLR.LVD1LVL[3:0] bits
	Monitoring flag	Not available	LVD1SR.LVD1MON flag: Monitors whether voltage is higher or lower than Vdet1
			LVD1SR.LVD1DET flag: Vdet1 passage detection
Process upon voltage detection	Reset	Voltage monitoring 0 reset	Voltage monitoring 1 reset
		Reset when Vdet0 > VCC CPU restart after specified time with VCC > Vdet0	Reset when Vdet1 > VCC CPU restart timing selectable: after specified time with VCC > Vdet1 or Vdet1 > VCC
	Interrupt	Not available	Voltage monitoring 1 interrupt
			Non-maskable or maskable interrupt is selectable
			Interrupt request issued when Vdet1 > VCC and VCC > Vdet1 or either
Event link function		Not available	Available Vdet1 passage detection event output

Downloaded from Arrow.com.

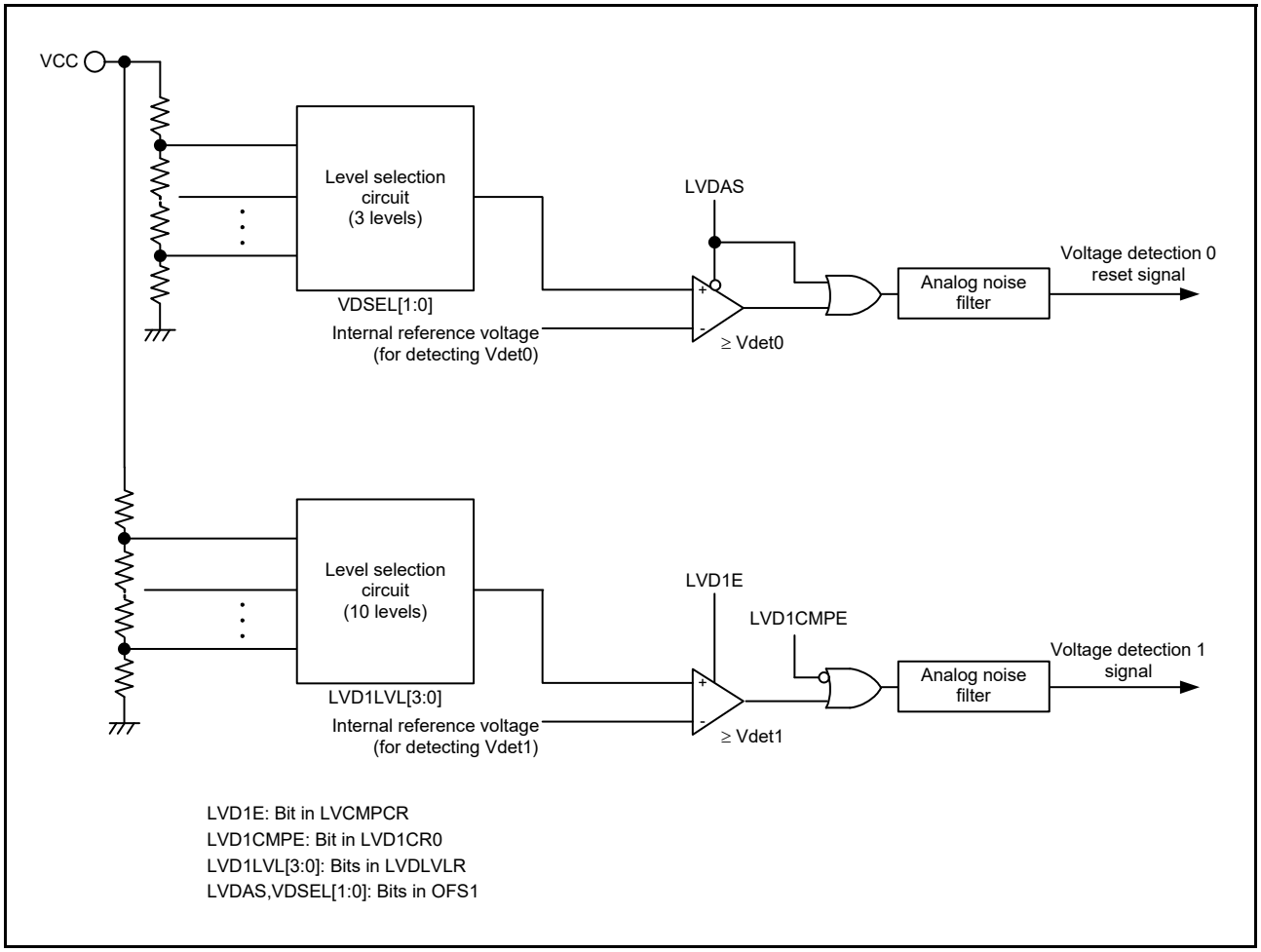


Figure 8.1 Block Diagram of the LVD

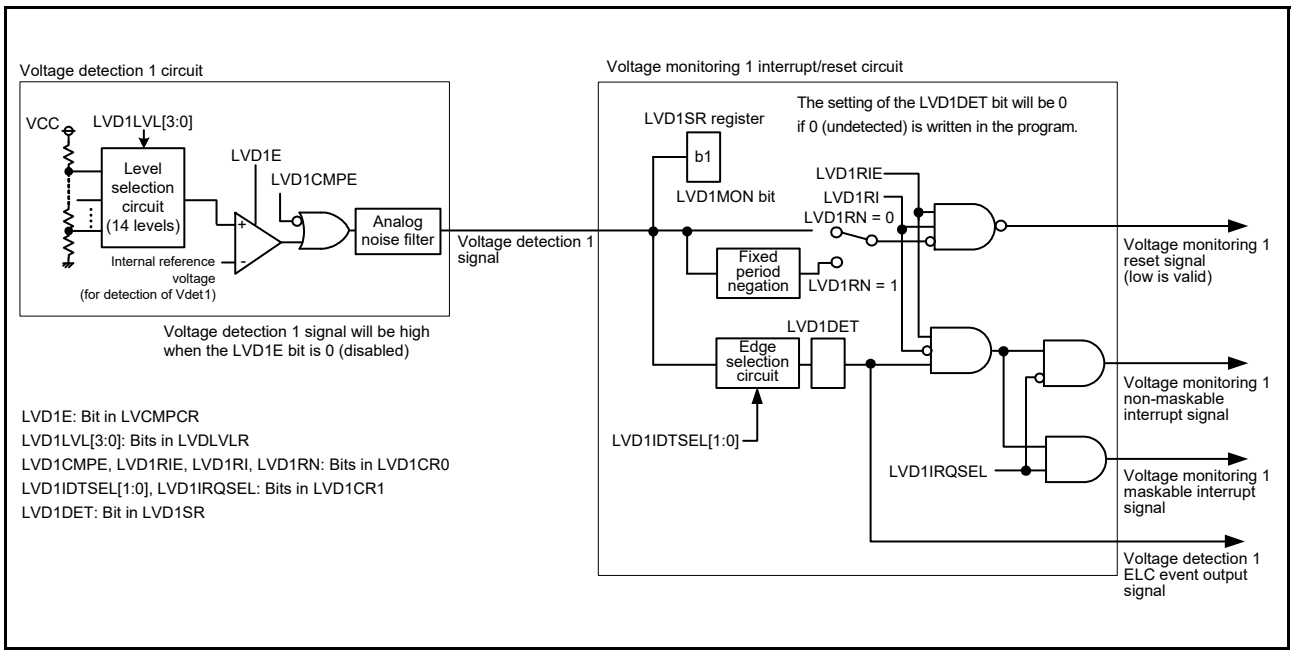


Figure 8.2 Block Diagram of Voltage Monitoring 1 Interrupt/Reset Circuit

Downloaded from Arrow.com.

8.2 Register Descriptions

8.2.1 Voltage Monitoring 1 Circuit Control Register 1 (LVD1CR1)

Address(es): 0008 00E0h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	LVD1IRQSEL	LVD1IDTSEL[1:0]	
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b1, b0	LVD1IDTSEL [1:0]	Voltage Monitoring 1 Interrupt ELC Event Generation Condition Select	b1 b0 0 0: When VCC ≥ Vdet1 (rise) is detected 0 1: When VCC < Vdet1 (drop) is detected 1 0: When drop and rise are detected 1 1: Setting prohibited	R/W
b2	LVD1IRQSEL	Voltage Monitoring 1 Interrupt Type Select	0: Non-maskable interrupt 1: Maskable interrupt	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.

Downloaded from Arrow.com.

8.2.2 Voltage Monitoring 1 Circuit Status Register (LVD1SR)

Address(es): 0008 00E1h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	LVD1MON	LVD1DET
Value after reset:	0	0	0	0	0	0	1	0

Bit	Symbol	Bit Name	Description	R/W
b0	LVD1DET	Voltage Monitoring 1 Voltage Change Detection Flag	0: Not detected 1: Vdet1 passage detection	R/(W) *1
b1	LVD1MON	Voltage Monitoring 1 Signal Monitor Flag	0: VCC < Vdet1 1: VCC ≥ Vdet1 or LVD1MON circuit is disabled	R
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.
Note 1. Only 0 can be written to this bit. After writing 0 to this bit, it takes two system clock cycles for the bit to be read as 0.

LVD1DET Flag (Voltage Monitoring 1 Voltage Change Detection Flag)

The LVD1DET flag is enabled when the LVCMPCR.LVD1E bit is 1 (voltage detection 1 circuit enabled) and the LVD1CR0.LVD1CMPE bit is 1 (voltage monitoring 1 circuit comparison result output enabled).
The LVD1DET flag should be set to 0 after LVD1CR0.LVD1RIE is set to 0 (disabled). LVD1CR0.LVD1RIE can be set to 1 (enabled) again after a period of two or more cycles of PCLKB has elapsed.
With read access to an I/O register which access cycle number is defined by PCLKB, two or more cycles of PCLKB may have to be secured as waiting time.

LVD1MON Flag (Voltage Monitoring 1 Signal Monitor Flag)

The LVD1MON flag is enabled when the LVCMPCR.LVD1E bit is 1 (voltage detection 1 circuit enabled) and the LVD1CR0.LVD1CMPE bit is 1 (voltage monitoring 1 circuit comparison result output enabled).

Downloaded from Arrow.com.

8.2.3 Voltage Monitoring Circuit Control Register (LVCMPCR)

Address(es): 0008 C297h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	LVD1E	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5	LVD1E	Voltage Detection 1 Enable	0: Voltage detection 1 circuit disabled 1: Voltage detection 1 circuit enabled	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.

LVD1E Bit (Voltage Detection 1 Enable)

When using voltage detection 1 interrupt/reset or the LVD1SR.LVD1MON flag, set the LVD1E bit to 1. The voltage detection 1 circuit starts once td(E-A) passes after the LVD1E bit value is changed from 0 to 1.

Downloaded from Arrow.com.

8.2.4 Voltage Detection Level Select Register (LVDLVLR)

Address(es): 0008 C298h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	LVD1LVL[3:0]			
Value after reset:	0	0	0	0	0	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	LVD1LVL[3:0]	Voltage Detection 1 Level Select (Standard voltage during drop in voltage)	b3 b0 0 1 0 0: 3.10 V 0 1 0 1: 3.00 V 0 1 1 0: 2.90 V 0 1 1 1: 2.79 V 1 0 0 0: 2.68 V 1 0 0 1: 2.58 V 1 0 1 0: 2.48 V 1 0 1 1: 2.20 V 1 1 0 0: 1.96 V 1 1 0 1: 1.86 V Settings other than those listed above are prohibited.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.

When changing the LVDLVLR register, first set the LVCMPCR.LVD1E bit to 0 (voltage detection 1 circuit disabled). For details on the voltage detection level range, refer to section 51, Electrical Characteristics.

Downloaded from Arrow.com.

8.2.5 Voltage Monitoring 1 Circuit Control Register 0 (LVD1CR0)

Address(es): 0008 C29Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	LVD1RN	LVD1RI	—	—	—	LVD1CMPE	—	LVD1RIE
Value after reset:	1	0	0	0	X	0	0	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	LVD1RIE	Voltage Monitoring 1 Interrupt/Reset Enable	0: Disabled 1: Enabled	R/W
b1	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b2	LVD1CMPE	Voltage Monitoring 1 Circuit Comparison Result Output Enable	0: Voltage monitoring 1 circuit comparison results output disabled 1: Voltage monitoring 1 circuit comparison results output enabled	R/W
b3	—	Reserved	The read value is undefined. The write value should be 0.	R/W
b5, b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	LVD1RI	Voltage Monitoring 1 Circuit Mode Select	0: Voltage monitoring 1 interrupt occurs when the voltage passes Vdet1 1: Voltage monitoring 1 reset occurs when the voltage falls below Vdet1	R/W
b7	LVD1RN	Voltage Monitoring 1 Reset Negation Select	0: Negation follows a stabilization time (tLVD1) after VCC > Vdet1 is detected. 1: Negation follows a stabilization time (tLVD1) after assertion of the voltage monitoring 1 reset.	R/W

Note: Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.

LVD1RIE Bit (Voltage Monitoring 1 Interrupt/Reset Enable)

The LVD1RIE bit is enabled when the LVCMPCR.LVD1E bit is set to 1 (voltage detection 1 circuit enabled) and the LVD1CMPE bit is set to 1 (voltage monitoring 1 circuit comparison results output enabled).

Ensure that neither a voltage monitoring 1 reset nor a voltage monitoring 1 non-maskable interrupt is generated during programming or erasure of the flash memory.

LVD1RN Bit (Voltage Monitoring 1 Reset Negation Select)

If the LVD1RN bit is to be set to 1 (negation follows a stabilization time after assertion of the voltage monitoring 1 reset), set the LOCOCR.LCSTP bit to 0 (LOCO is operating). Furthermore, if a transition to software standby mode, the only possible value for the LVD1RN bit is 0 (negation follows a stabilization time after VCC > Vdet1 is detected). Do not set the LVD1RN bit to 1 (negation follows a stabilization time after assertion of the voltage monitoring 1 reset).

Downloaded from Arrow.com.

8.3 VCC Input Voltage Monitor

8.3.1 Monitoring Vdet0

Monitoring Vdet0 is not possible.

8.3.2 Monitoring Vdet1

After making the following settings, the LVD1SR.LVD1MON flag can be used to monitor the results of comparison by voltage monitor 1.

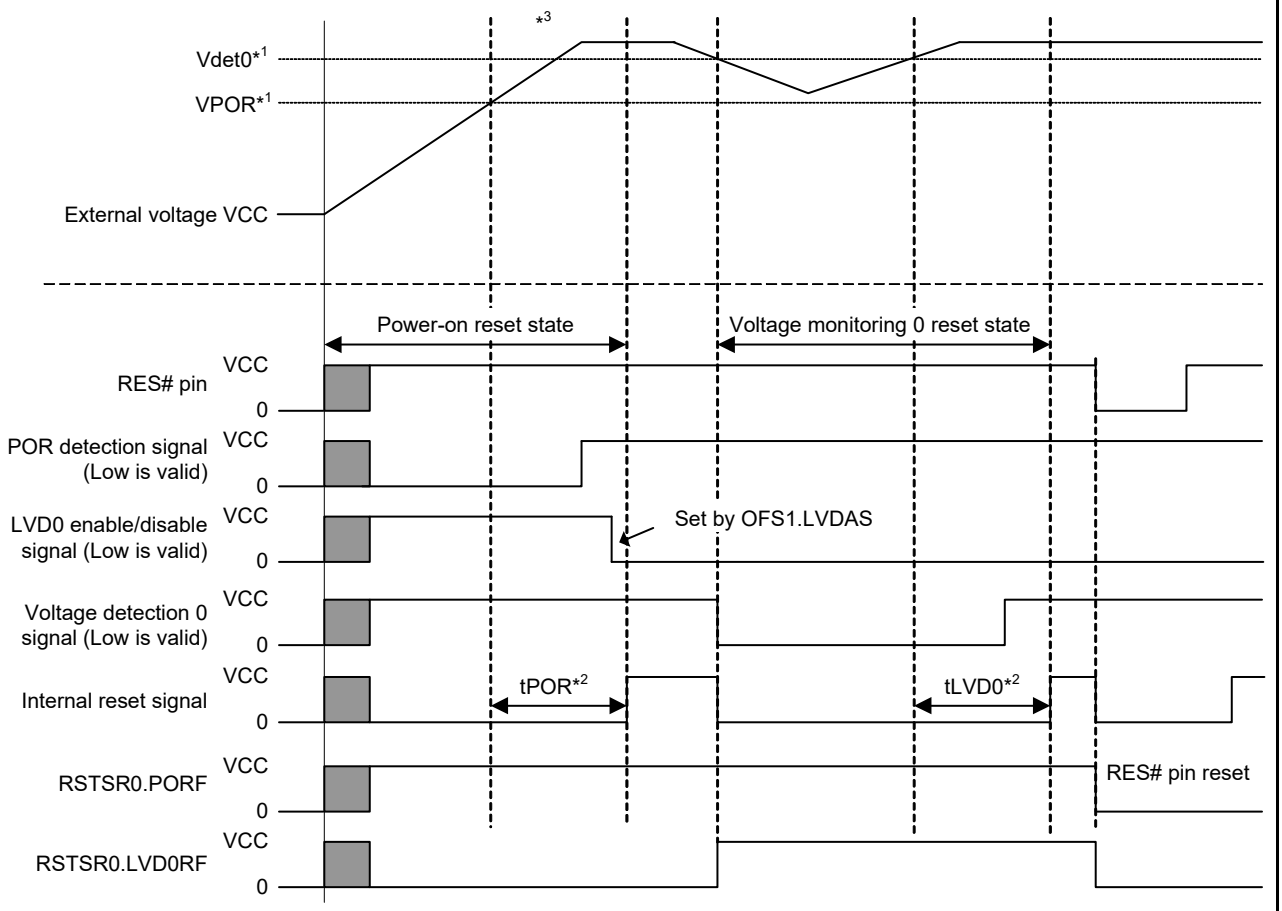
(1) Specify the detection voltage by setting the LVDLVLR.LVD1LVL[3:0] bits (voltage detection 1 level select).
(2) Set the LVCMPCR.LVD1E bit to 1 (voltage detection 1 circuit enabled).
(3) After waiting for td(E-A), set the LVD1CR0.LVD1CMPE bit to 1 (voltage monitoring 1 circuit comparison results output enabled).

Downloaded from Arrow.com.

8.4 Reset from Voltage Monitor 0

When using the reset from voltage monitor 0, clear the voltage detection 0 circuit start bit (OFS1.LVDAS) to 0 (enabling the voltage monitor 0 reset after a reset).

Figure 8.3 shows an example of operations for a voltage monitoring 0 reset.



Note: For details on the electrical characteristics, see the Electrical Characteristics section.
Note 1. Vdet0 indicates the detection level for a voltage monitoring 0 reset and VPOR indicates the detection level for a power-on reset.
Note 2. tPOR indicates the period of a power-on reset and tLVD0 indicates the period of a voltage monitoring 0 reset.
Note 3. At the time the power-supply voltage rises, VCC must rise to at least the minimum guaranteed voltage before release from the POR reset state.

Figure 8.3 Example of Voltage Monitoring 0 Reset Operation

Downloaded from Arrow.com.

8.5 Interrupt and Reset from Voltage Monitoring 1

Table 8.2 shows the procedures for setting bits related to the voltage monitoring 1 interrupt and voltage monitoring 1 reset. Table 8.3 shows the procedures for stopping bits related to the voltage monitoring 1 interrupt and voltage monitoring 1 reset. Figure 8.4 shows an example of operations for a voltage monitoring 1 interrupt. For the operation of the voltage monitoring 1 reset, see Figure 6.2 in section 6, Resets.

Table 8.2 Procedures for Setting Bits Related to the Voltage Monitoring 1 Interrupt and Voltage Monitoring 1 Reset

Step	Voltage Monitoring 1 Interrupt, Voltage Monitoring 1 ELC Event Output	Voltage Monitoring 1 Reset
1*1	Select the detection voltage by setting the LVDLVLR.LVD1LVL[3:0] bits.	
2*1	Set the LVD1CR0.LVD1RI bit to 0 (voltage monitoring 1 interrupt).	Set the LVD1CR0.LVD1RI bit to 1 (voltage monitoring 1 reset). Select the type of reset negation by setting the LVD1CR0.LVD1RN bit.
3	Select the timing of interrupt requests by setting the LVD1CR1.LVD1IDTSEL[1:0] bits. Select the type of interrupt by setting the LVD1CR1.LVD1IRQSEL bit.	—
4	—	Set the LVD1CR0.LVD1RIE bit to 1 (voltage monitoring 1 interrupt/reset enabled).
5*1	Set the LVCMPCR.LVD1E bit to 1 (voltage detection 1 circuit enabled).	
6*1	Wait for at least td(E-A).	
7	Set the LVD1CR0.LVD1CMPE bit to 1 (voltage monitoring 1 circuit comparison results output enabled).	
8	Wait for at least 2 µs.	—
9	Set the LVD1SR.LVD1DET bit to 0.	—
10	Set the LVD1CR0.LVD1RIE bit to 1 (voltage monitoring 1 interrupt/reset enabled).	—

Note 1. Steps 1, 2, 5, and 6 are not required if operation is with the setting to select the voltage monitoring 1 interrupt (LVD1CR0.LVD1RI = 0) and operation can be restarted by simply changing the settings of the LVD1CR1.LVD1IRQSEL and LVD1IDTSEL[1:0] bits after monitoring is stopped or if restarting is in a case where the settings related to the voltage-detection circuit were not changed after monitoring was stopped. When changes are to be made and operation is with the setting to select the voltage monitoring 1 reset (LVD1CR0.LVD1RI = 1), proceed through all steps from 1 to 10.

Table 8.3 Procedures for Stopping Bits Related to the Voltage Monitoring 1 Interrupt and Voltage Monitoring 1 Reset

Step	Voltage Monitoring 1 Interrupt, Voltage Monitoring 1 ELC Event Output	Voltage Monitoring 1 Reset
1	Set the LVD1CR0.LVD1RIE bit to 0 (voltage monitoring 1 interrupt/reset disabled).	—
2	Set the LVD1CR0.LVD1CMPE bit to 0 (voltage monitoring 1 circuit comparison results output disabled).	
3*1	Set the LVCMPCR.LVD1E bit to 0 (voltage detection 1 circuit disabled).	
4	—	Set the LVD1CR0.LVD1RIE bit to 0 (voltage monitoring 1 interrupt/reset disabled).
5	Modify settings of bits related to the voltage detection circuit registers other than LVCMPCR.LVD1E, LVD1CR0.LVD1RIE, and LVD1CR0.LVD1CMPE.	

Note 1. Step 3 is not required if operation is with the setting to select the voltage monitoring 1 interrupt (LVD1CR0.LVD1RI = 0) and operation can be restarted by simply changing the settings of the LVD1CR1.LVD1IRQSEL and LVD1IDTSEL[1:0] bits after monitoring is stopped or if restarting is in a case where the settings related to the voltage-detection circuit were not changed after monitoring was stopped. When changes are to be made and operation is with the setting to select the voltage monitoring 1 reset (LVD1CR0.LVD1RI = 1), proceed through all steps from 1 to 5.

Downloaded from Arrow.com.

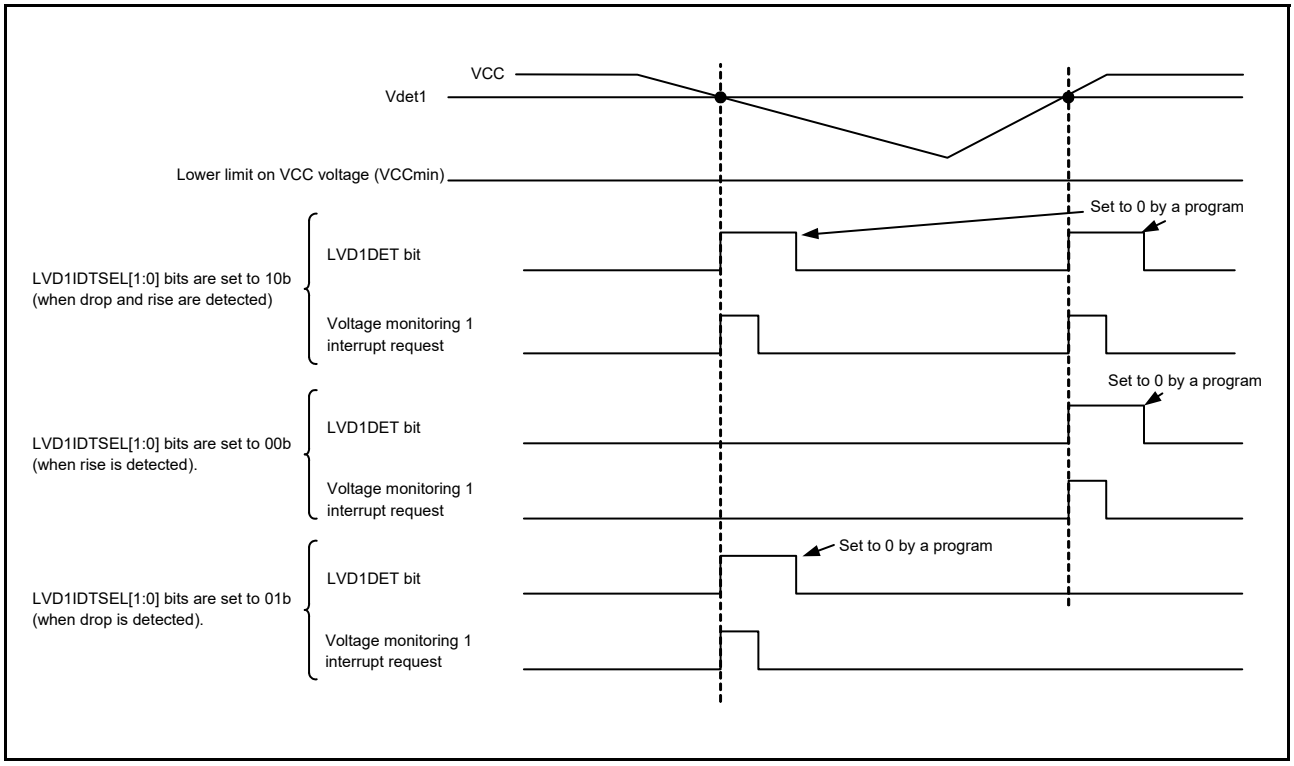


Figure 8.4 Example of Voltage Monitoring 1 Interrupt Operation

Downloaded from Arrow.com.

8.6 Event Link Output

The LVD can output the event signals to the event link controller (ELC).

(1) Vdet1 passage detection event output

The LVD outputs the event signal when it is detected that the voltage has passed the Vdet1 voltage while both the voltage detection 1 circuit and the voltage monitoring 1 circuit comparison result output are enabled.

When enabling the LVD's event link output function, be sure to make settings for enabling the LVD before enabling the LVD event link function of the ELC. To stop the LVD's event link output function, be sure to make settings for stopping the LVD after disabling the LVD event link function of the ELC.

8.6.1 Interrupt Handling and Event Linking

The LVD has the bits to separately enable or disable the voltage monitoring 1 interrupt. When an interrupt source is generated and the interrupt is enabled by the interrupt enable bit, the interrupt request signal is output to the CPU.
On the contrary, as soon as an interrupt source is generated, the event link signal is output as the event signal to the other module via the ELC regardless of the state of the interrupt enable bit.
It is possible to output voltage monitoring 1 interrupt in software standby. The event signal for the ELC, however, is output as follows:

- When the event of passing Vdet1 is detected in software standby mode, no event signals are generated for the ELC because no clock is presented in software standby mode. Since the Vdet1 passage detection flag is preserved, however, when the supply of the clock is resumed after restoring from software standby mode, the event signal for the ELC is output according to the state of the Vdet1 passage detection flag.

Downloaded from Arrow.com.

9. Clock Generation Circuit

9.1 Overview

This MCU incorporates a clock generation circuit.
Table 9.1 lists the specifications of the clock generation circuit. Figure 9.1 and Figure 9.2 show a block diagram of the clock generation circuit.

Table 9.1 Specifications of Clock Generation Circuit (1/2)

Item	Specification
Uses	• Generates the system clock (ICLK) to be supplied to the CPU, DMAC, DTC, ROM, and RAM. • Generates the peripheral module clocks (PCLKA, PCLKB, and PCLKD) to be supplied to peripheral modules. The peripheral module clock PCLKA is the operating clock for the MTU2, the peripheral module clock PCLKD is for the S12AD, and PCLKB is for other modules. • Generates the FlashIF clock (FCLK) to be supplied to the FlashIF. • Generates the USB clock (UCLK) to be supplied to the USB. • Generates the CAC clock (CACCLK) to be supplied to the CAC. • Generates the RTC-dedicated sub-clock (RTCSCLK) to be supplied to the RTC. • Generates the IWDT-dedicated low-speed clock (IWDTCLK) to be supplied to the IWDT. • Generates the CAN clock (CANMCLK) to be supplied to the RSCAN. • Generates the SSI clock (SSISCK) to be supplied to the SSI. • Generates the LPT clock (LPTCLK) to be supplied to the LPT. • Generation of Bluetooth-dedicated clock (BLECK) • Generation of Bluetooth-dedicated low speed clock (BLELOCO)
Operating frequencies*1	• ICLK: 54 MHz (max) • PCLKA: 54 MHz (max) • PCLKB: 32 MHz (max) • PCLKD: 54 MHz (max) • FCLK: 1 to 32 MHz (for programming and erasing the ROM and E2 DataFlash) 32 MHz (max) (for reading from the E2 DataFlash) • UCLK: 48 MHz • CACCLK: Same frequency as each oscillator • RTCSCLK: 32.768 kHz • IWDTCLK: 15 kHz • CANMCLK: 20 MHz (max) • SSISCK: 20 MHz (max) • LPTCLK: The same frequency as that of the selected oscillator • BLECK: 32 MHz • BLELOCO: 32.768 kHz
Main clock oscillator*2	• Resonator frequency: 1 to 20 MHz (VCC ≥ 2.4 V), 1 to 8 MHz (VCC < 2.4 V) • External clock input frequency: 20 MHz (max) • Connectable resonator or additional circuit: ceramic resonator, crystal • Connection pins: EXTAL, XTAL • Oscillation stop detection function: When a main clock oscillation stop is detected, the system clock source is switched to LOCO and MTU output can be forcedly driven to high-impedance. • Drive capacity switching function
Sub-clock oscillator	• Resonator frequency: 32.768 kHz • Connectable resonator or additional circuit: crystal • Connection pin: XCIN, XCOUT • Drive capacity switching function
PLL circuit*3	• Input clock source: Main clock • Input pulse frequency division ratio: Selectable from 1, 2, and 4 • Input frequency: 4 to 12.5 MHz • Frequency multiplication ratio: Selectable from 4 to 13.5 (increments of 0.5) • VCO oscillation frequency: 24 to 54 MHz (VCC ≥ 2.4 V)
USB-dedicated PLL circuit*3	• Input clock source: Main clock • Input pulse frequency division ratio: Selectable from 1, 2, and 4 • Input frequency: 4, 6, 8, and 12 MHz • Frequency multiplication ratio: Selectable from 4, 6, 8, and 12 • VCO oscillation frequency: 48 MHz (VCC ≥ 2.4 V)

Downloaded from Arrow.com.

Table 9.1 Specifications of Clock Generation Circuit (2/2)

Item	Specification
High-speed on-chip oscillator (HOCO)	Oscillation frequency: 32 and 54 MHz
Low-speed on-chip oscillator (LOCO)	Oscillation frequency: 4 MHz
IWDT-dedicated on-chip oscillator	Oscillation frequency: 15 kHz
Bluetooth-dedicated clock oscillator	• Frequency of oscillation: 32 MHz • Connectable resonator: Crystal (The 83-pin LGA package product includes a crystal resonator.) • Connection pin: XTAL1_RF, and XTAL2_RF
Bluetooth-dedicated low-speed on-chip oscillator	Frequency of oscillation: 32.768 kHz

Note 1. The maximum operating frequency in high-speed operating mode. For the maximum operating frequency in the other operating modes, refer to section 11.2.6, Operating Power Control Register (OPCCR).
Note 2. When oscillating USB-dedicated PLL at 48 MHz and the PLL at 54 MHz, the frequency of the main clock oscillator should be set to 4, 6, 8, or 12 MHz.
Note 3. The PLL and USB-dedicated PLL can be used when the external voltage (VCC) is 2.4 V or above.

Downloaded from Arrow.com.

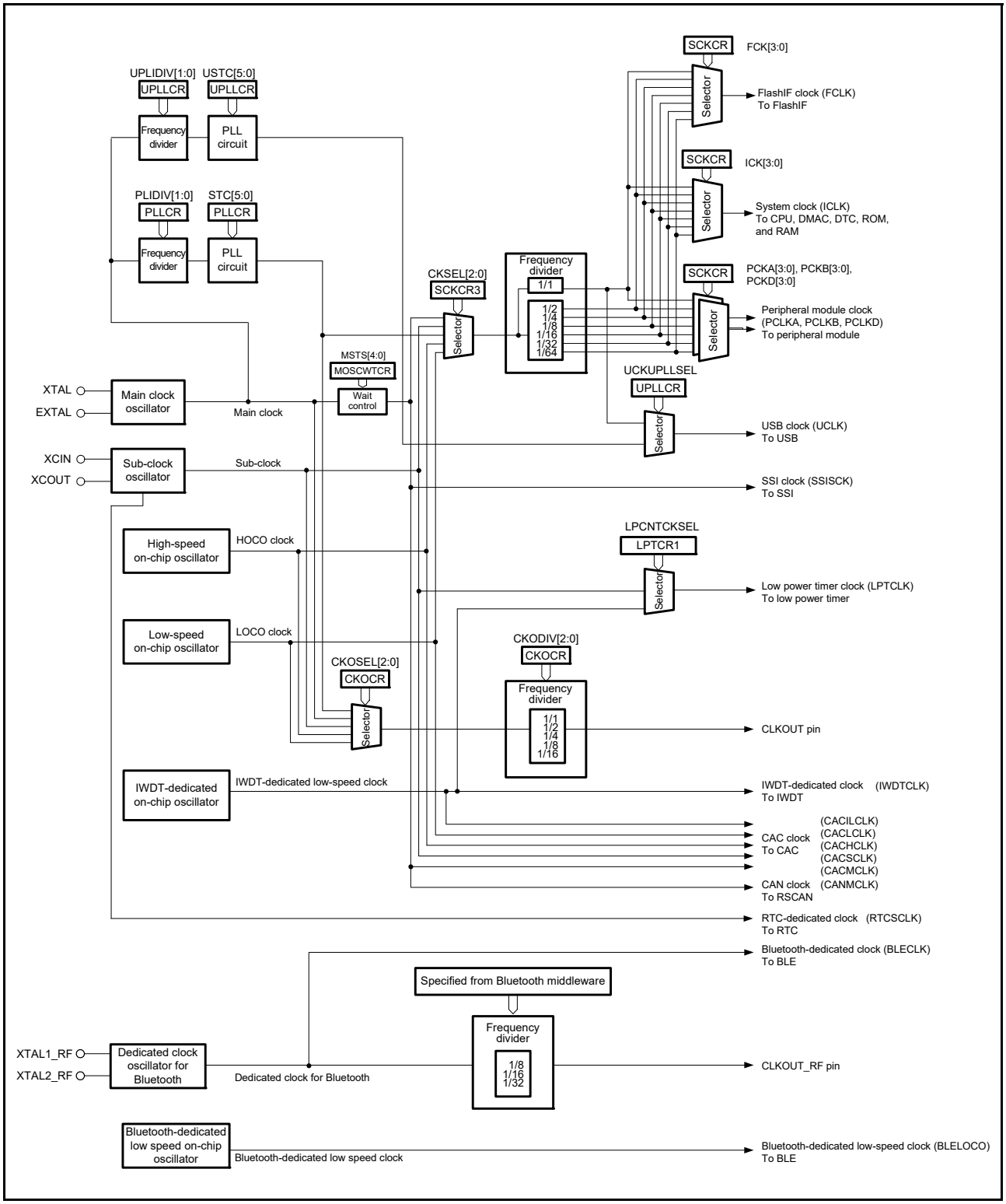


Figure 9.1 Block Diagram of Clock Generation Circuit (85-Pin BGA, 56-Pin QFN)

Downloaded from Arrow.com.

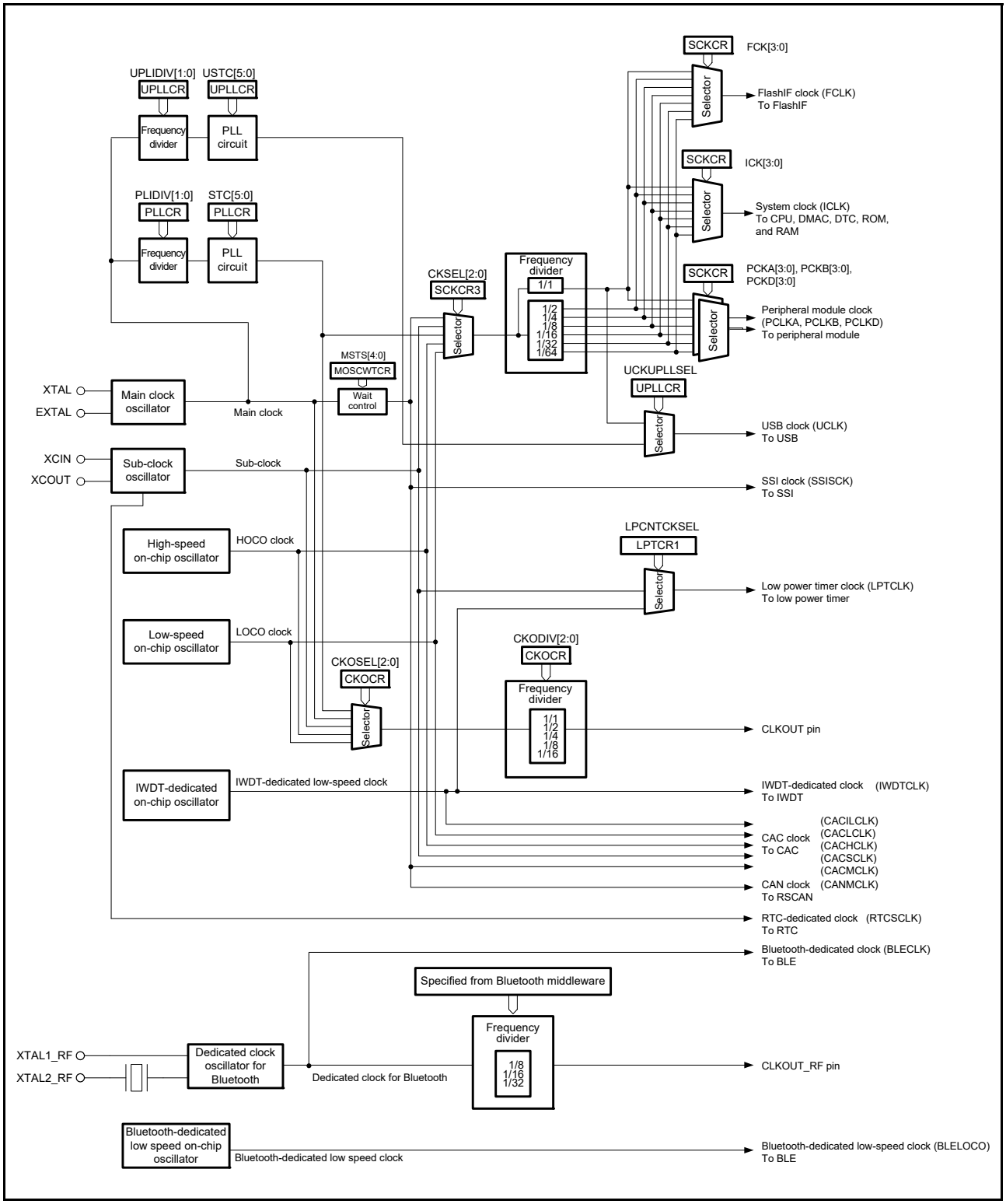


Figure 9.2 Block Diagram of Clock Generation Circuit (83-Pin LGA)

Downloaded from Arrow.com.

Table 9.2 lists the I/O pins of the clock generation circuit.

Table 9.2 I/O Pins of Clock Generation Circuit

Pin Name	I/O	Description
XTAL	Output	These pins are used to connect a crystal. The EXTAL pin can also be used to input an external clock. For details, refer to section 9.3.2, External Clock Input.
EXTAL	Input	
XCIN	Input	These pins are used to connect a 32.768-kHz crystal.
XCOUT	Output	
CLKOUT	Output	Clock output pin
XTAL1_RF	Input	Connect a 32-MHz oscillator. The 83-pin LGA product incorporates a 32-MHz crystal.
XTAL2_RF	Output	
CLKOUT_RF	Output	Bluetooth-dedicated clock output pin

Downloaded from Arrow.com.

9.2 Register Descriptions

9.2.1 System Clock Control Register (SCKCR)

Address(es): 0008 0020h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	FCK[3:0]				ICK[3:0]				—	—	—	—	—	—	—	—
Value after reset:	0	0	1	1	0	0	1	1	0	0	0	0	0	0	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PCKA[3:0]				PCKB[3:0]				—	—	—	—	PCKD[3:0]			
Value after reset:	0	0	1	1	0	0	1	1	0	0	0	0	0	0	1	1

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	PCKD[3:0]	Peripheral Module Clock D (PCLKD) Select	b3 b0 0 0 0 0: ×1 0 0 0 1: ×1/2 0 0 1 0: ×1/4 0 0 1 1: ×1/8 0 1 0 0: ×1/16 0 1 0 1: ×1/32 0 1 1 0: ×1/64 Settings other than above are prohibited.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11 to b8	PCKB[3:0]	Peripheral Module Clock B (PCLKB) Select	b11 b8 0 0 0 0: ×1 0 0 0 1: ×1/2 0 0 1 0: ×1/4 0 0 1 1: ×1/8 0 1 0 0: ×1/16 0 1 0 1: ×1/32 0 1 1 0: ×1/64 Settings other than above are prohibited.	R/W
b15 to b12	PCKA[3:0]	Peripheral Module Clock A (PCLKA) Select	b15 b12 0 0 0 0: ×1 0 0 0 1: ×1/2 0 0 1 0: ×1/4 0 0 1 1: ×1/8 0 1 0 0: ×1/16 0 1 0 1: ×1/32 0 1 1 0: ×1/64 Settings other than above are prohibited.	R/W
b19 to b16	—	Reserved	Set the value to whichever is greater of the values of the ICK[3:0] and PCKB[3:0] bits.	R/W
b23 to b20	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b27 to b24	ICK[3:0] *1	System Clock (ICLK) Select	b27 b24 0 0 0 0: ×1 0 0 0 1: ×1/2 0 0 1 0: ×1/4 0 0 1 1: ×1/8 0 1 0 0: ×1/16 0 1 0 1: ×1/32 0 1 1 0: ×1/64 Settings other than above are prohibited.	R/W

Downloaded from Arrow.com.

Bit	Symbol	Bit Name	Description	R/W
b31 to b28	FCK[3:0]	FlashIF Clock (FCLK) Select	b31 b28 0 0 0 0: ×1 0 0 0 1: ×1/2 0 0 1 0: ×1/4 0 0 1 1: ×1/8 0 1 0 0: ×1/16 0 1 0 1: ×1/32 0 1 1 0: ×1/64 Settings other than above are prohibited.	R/W

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
Note 1. Do not set the frequency division ratio of ICLK = 1 when a clock of frequency higher than 32 MHz is selected by the SCKCR3.CKSEL[2:0] bits and MEMWAIT.MEMWAIT = 0.

This register cannot be rewritten while the flash memory is being programmed or erased.
When an instruction for writing to SCKCR or SCKCR3 is to follow writing to the SCKCR register, do so in accord with the procedure below.

1. Write to the SCKCR register.
2. Confirm that the value has actually been written to the SCKCR register.
3. Proceed to the next step.

PCKD[3:0] Bits (Peripheral Module Clock D (PCLKD) Select)

These bits select the frequency of peripheral module clock D (PCLKD).

PCKB[3:0] Bits (Peripheral Module Clock B (PCLKB) Select)

These bits select the frequency of peripheral module clock B (PCLKB).

PCKA[3:0] Bits (Peripheral Module Clock A (PCLKA) Select)

These bits select the frequency of peripheral module clock A (PCLKA).

ICK[3:0] Bits (System Clock (ICLK) Select)

These bits select the frequency of the system clock (ICLK).

FCK[3:0] Bits (FlashIF Clock (FCLK) Select)

These bits select the frequency of the FlashIF clock (FCLK).

Downloaded from Arrow.com.

9.2.2 System Clock Control Register 3 (SCKCR3)

Address(es): 0008 0026h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	CKSEL[2:0]			—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b10 to b8	CKSEL[2:0] *1	Clock Source Select	b10 b8 0 0 0: LOCO 0 0 1: HOCO 0 1 0: Main clock oscillator 0 1 1: Sub-clock oscillator 1 0 0: PLL circuit Settings other than above are prohibited.	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
Note 1. Do not select a clock of frequency higher than 32 MHz when divided by 1 is selected by SCKCR.ICK and MEMWAIT.MEMWAIT = 0.

This register cannot be rewritten while the flash memory is being programmed or erased.

CKSEL[2:0] Bits (Clock Source Select)

These bits select the source of the system clock (ICLK), peripheral module clock (PCLKA, PCLKB, and PCLKD), FlashIF clock (FCLK), and USB clock (UCLK) from low-speed on-chip oscillator (LOCO), high-speed on-chip oscillator (HOCO), the main clock oscillator, the sub-clock oscillator, and the PLL circuit.
Transitions to clock sources which are not in operation are prohibited.

Downloaded from Arrow.com.

9.2.3 PLL Control Register (PLLCR)

Address(es): 0008 0028h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	STC[5:0]						—	—	—	—	—	—	PLIDIV[1:0]	
Value after reset:	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	PLIDIV[1:0]	PLL Input Frequency Division Ratio Select	b1 b0 0 0: ×1 0 1: ×1/2 1 0: ×1/4 1 1: Setting prohibited	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b13 to b8	STC[5:0]	Frequency Multiplication Factor Select	b13 b8 0 0 0 1 1 1: ×4 0 0 1 0 0 0: ×4.5 0 0 1 0 0 1: ×5 0 0 1 0 1 0: ×5.5 0 0 1 0 1 1: ×6 0 0 1 1 0 0: ×6.5 0 0 1 1 0 1: ×7 0 0 1 1 1 0: ×7.5 0 0 1 1 1 1: ×8 0 1 0 0 0 0: ×8.5 0 1 0 0 0 1: ×9 0 1 0 0 1 0: ×9.5 0 1 0 0 1 1: ×10 0 1 0 1 0 0: ×10.5 0 1 0 1 0 1: ×11 0 1 0 1 1 0: ×11.5 0 1 0 1 1 1: ×12 0 1 1 0 0 0: ×12.5 0 1 1 0 0 1: ×13 0 1 1 0 1 0: ×13.5 Settings other than above are prohibited.	R/W
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

Writing to the PLLCR is prohibited when the PLLCR2.PLLEN bit is 0 (PLL is operating).

PLIDIV[1:0] Bits (PLL Input Frequency Division Ratio Select)

These bits select the frequency division ratio of the PLL clock source.

Set these bits so that the frequency of PLL input signal is within the range of 4 MHz to 12.5 MHz.

STC[5:0] Bits (Frequency Multiplication Factor Select)

These bits select the frequency multiplication factor of the PLL circuit.

Set these bits so that the PLL oscillation frequency is within the range of 24 MHz to 54 MHz.

Downloaded from Arrow.com.

9.2.4 PLL Control Register 2 (PLLCR2)

Address(es): 0008 002Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	PLLEN
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	PLLEN	PLL Stop Control	0: PLL is operating. 1: PLL is stopped.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

PLLEN Bit (PLL Stop Control)

This bit runs or stops the PLL circuit.

After setting the PLLEN bit to 0 (PLL is operating), confirm that the OSCOVFSR.PLOVF bit is 1 before switching the system clock to the PLL clock.

That is, a fixed time for stabilization is required after the setting for PLL operation. A fixed time is also required for oscillation to stop after the setting to stop PLL operation. Accordingly, take note of the following limitations when starting and stopping PLL operation.

- After stopping the PLL, confirm that the OSCOVFSR.PLOVF bit is 0 before restarting the PLL.
- Confirm that the PLL is operating and that the OSCOVFSR.PLOVF bit is 1 before stopping the PLL.
- Regardless of whether or not it is selected as the system clock, confirm that the OSCOVFSR.PLOVF bit is 1 before executing a WAIT instruction to place the MCU in software standby mode.
- After stopping the PLL, confirm that the OSCOVFSR.PLOVF bit is 0 and execute a WAIT instruction before entering software standby mode.

When the PLL clock is selected by the SCKCR3.CKSEL[2:0] bits, do not set the PLLEN bit (PLL is stopped) to 1.

When the external voltage (VCC) is below 2.4 V or low-speed operating mode is selected by the SOPCCR.SOPCM bit, do not set the PLLEN bit to 0 (PLL is operating).

Downloaded from Arrow.com.

9.2.5 USB-dedicated PLL Control Register (UPLLCR)

Address(es): 0008 002Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	USTC[5:0]						—	—	—	UCKUPLLSEL	—	—	UPLIDIV[1:0]	
Value after reset:	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	UPLIDIV[1:0]	USB-dedicated PLL Input Frequency Division Ratio Select	b1 b0 0 0: ×1 0 1: ×1/2 1 0: ×1/4 Settings other than above are prohibited.	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	UCKUPLLSEL	UCLK Source USB-Dedicated PLL Select	0: System clock is selected as UCLK 1: USB-dedicated PLL is selected as UCLK	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b13 to b8	USTC[5:0]	Frequency Multiplication Factor Select	b13 b8 0 0 0 1 1 1: ×4 0 0 1 0 1 1: ×6 0 0 1 1 1 1: ×8 0 1 0 1 1 1: ×12 Settings other than above are prohibited.	R/W
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

Writing to the UPLLCR is prohibited when the UPLLCR2.UPLLEN bit is 0 (USB-dedicated PLL is operating).

UPLIDIV[1:0] Bits (USB-dedicated PLL Input Frequency Division Ratio Select)

These bits select the input frequency division ratio of the USB-dedicated PLL clock source.
Set these bits so that the input frequency of the USB-dedicated PLL input is 4, 6, 8, 12 MHz.

UCKUPLLSEL Bit (UCLK Source USB-Dedicated PLL Select)

This bit selects the USB-dedicated PLL clock. When this bit is set to 1, the USB-dedicated PLL (48 MHz) is supplied to the UCLK, and the clock of the oscillator (up to 54 MHz) selected by the SCKCR3.CKSEL[2:0] bits is supplied to the system clock.
When setting this bit to change the UCLK source clock, set the MSTPCRB.MSTPB19 bit (this module clock is disabled) and wait for at least three cycles of the clock before the change.

USTC[5:0] Bits (Frequency Multiplication Factor Select)

These bits select the frequency multiplication factor of the USB-dedicated PLL circuit.
Set these bits so that the USB-dedicated PLL oscillation frequency is 48 MHz.

Downloaded from Arrow.com.

9.2.6 USB-dedicated PLL Control Register 2 (UPLLCR2)

Address(es): 0008 002Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	UPLLEN
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	UPLLEN	USB-dedicated PLL Stop Control	0: USB-dedicated PLL is operating. 1: USB-dedicated PLL is stopped.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

The UPLLCR2 register runs or stops the USB-dedicated PLL circuit.

UPLLEN Bit (USB-dedicated PLL Stop Control)

This bit runs or stops the USB-dedicated PLL circuit.
After setting the UPLLEN bit to 0 (USB-dedicated PLL is operating), read the OSCOVFSR.UPLOVF flag to confirm that it has become 1, and then start using the USB-dedicated PLL clock.
That is, a fixed time for stabilization is required after the setting for USB-dedicated PLL operation. A fixed time is also required for oscillation to stop after the setting to stop USB-dedicated PLL operation. Accordingly, take note of the following limitations when starting and stopping USB-dedicated PLL operation.

- After stopping the USB-dedicated PLL, confirm that the OSCOVFSR.UPLOVF bit is 0 before restarting the USB-dedicated PLL.
- Confirm that the USB-dedicated PLL is operating and that the OSCOVFSR.UPLOVF bit is 1 before stopping the USB-dedicated PLL.
- Regardless of whether or not it is selected as the system clock, confirm that the OSCOVFSR.UPLOVF bit is 1 before executing a WAIT instruction to place the MCU in software standby mode.
- After stopping the USB-dedicated PLL, confirm that the OSCOVFSR.UPLOVF bit is 0 and execute a WAIT instruction before entering software standby mode.

When the external voltage (VCC) is below 2.4 V or low-speed operating mode is selected by the SOPCCR.SOPCM bit, do not set the UPLLEN bit to 0 (USB-dedicated PLL is operating).
When the USB-dedicated PLL is selected by the UPLLCR.UCKUPLLSEL bit and the MSTPCRB.MSTPB19 bit is set to 0 (this module clock is enabled), do not set the UPLLEN bit to 1 (USB-dedicated PLL is stopped).

Downloaded from Arrow.com.

9.2.7 Main Clock Oscillator Control Register (MOSCCR)

Address(es): 0008 0032h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	MOSTP
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	MOSTP	Main Clock Oscillator Stop	0: Main clock oscillator is operating. 1: Main clock oscillator is stopped.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

Set this register after setting up the main clock oscillator wait control register.

MOSTP Bit (Main Clock Oscillator Stop)

This bit runs or stops the main clock oscillator.
After setting the MOSTP bit to 0 (main clock oscillator is operating), read the OSCOVFSR.MOOVF bit to confirm that is has become 1, and then use the main clock.
For the main clock oscillator, a fixed time is required for oscillation to become stable after the settings for operation have been made. Furthermore, a fixed time is required for oscillation to actually stop after the settings to stop oscillation have been made. Accordingly, take note of the following limitations when starting and stopping operation.

- After stopping the main clock oscillator, confirm that the OSCOVFSR.MOOVF bit is 0 before restarting the main clock oscillator.
- Confirm that the main clock oscillator is operating and that the OSCOVFSR.MOOVF bit is 1 before stopping the main clock oscillator.
- Regardless of whether or not it is selected as the system clock, confirm that the OSCOVFSR.MOOVF bit is 1 and execute a WAIT instruction in order to operate the main clock oscillator and place the MCU in software standby mode.
- After stopping the main clock oscillator, confirm that the OSCOVFSR.MOOVF bit is 0 and execute a WAIT instruction before entering software standby mode.

Do not set the MOSTP bit to 1 when one of the following condition is met.

- When the main clock is selected as the clock source for the system clock (SCKCR3.CKSEL[2:0] = 010b)
- When the PLL clock is selected as the clock source for the system clock (SCKCR3.CKSEL[2:0] = 100b)
- When the PLL is operating (PLLCR2.PLLEN = 0)
- When the USB-dedicated PLL is operating (UPLLCR2.PLLEN = 0)

Do not set the MOSTP bit to 0 when the following condition is met.

- When low-speed operating mode is selected by the SOPCCR.SOPCM bit

Downloaded from Arrow.com.

9.2.8 Sub-Clock Oscillator Control Register (SOSCCR)

Address(es): 0008 0033h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	SOSTP
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SOSTP	Sub-Clock Oscillator Stop	0: Sub-clock oscillator is operating. 1: Sub-clock oscillator is stopped.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

SOSTP Bit (Sub-Clock Oscillator Stop)

This bit runs or stops the sub-clock oscillator.

The SOSTP bit and the sub-clock oscillator control bit in RTC control register 3 (RCR3.RTCEN) controls whether to operate or stop the sub-clock oscillator. If one of these bits is set so as to enable the operation, the sub-clock oscillator runs.

When changing the value of the SOSTP bit or RCR3.RTCEN bit, execute subsequent instructions after reading the bit and checking that its value has actually been updated (refer to (2), Notes on writing to I/O registers, in section 5, I/O Registers).

After the setting of the SOSTP bit or the RCR3.RTCEN bit has been changed so that the sub-clock oscillator operates, only start using the sub-clock after the sub-clock oscillation stabilization time (t_{SUBOSC}) has elapsed.

That is, a fixed time for stabilization is required after the setting for sub-clock oscillator operation. A fixed time is also required for oscillation to stop after the setting to stop the oscillator. Accordingly, take note of the following limitations when starting and stopping the oscillator.

- When restarting the sub-clock oscillator after it has been stopped, allow at least five cycles of the sub-clock as an interval over which it is still stopped.
- Ensure that oscillation by the sub-clock oscillator is stable when making the setting to stop the sub-clock oscillator.
- Regardless of whether or not it is selected as the system clock, ensure that oscillation by the sub-clock oscillator is stable before executing a WAIT instruction to place the chip on software standby.
- When a transition to software standby mode is to follow the setting to stop the sub-clock oscillator, wait for at least two cycles of the sub-clock oscillator after the setting to stop the sub-clock oscillator and before executing the WAIT instruction.

While the sub-clock oscillator is selected by the SCKCR3.CKSEL[2:0] bits, do not set the SOSTP bit to 1 (sub-clock oscillator is stopped).

Downloaded from Arrow.com.

9.2.9 Low-Speed On-Chip Oscillator Control Register (LOCOCR)

Address(es): 0008 0034h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	LCSTP
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	LCSTP	LOCO Stop	0: LOCO is operating. 1: LOCO is stopped.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

LCSTP Bit (LOCO Stop)

This bit runs or stops the LOCO.

After the setting of the LCSTP bit has been changed so that the LOCO operates, only start using the LOCO clock after the LOCO clock oscillation stabilization time (t_{LOCO}) has elapsed.

That is, a fixed time for stabilization of oscillation is required after the setting for LOCO operation. A fixed time is also required for oscillation to stop after the setting to stop the oscillator. Accordingly, take note of the following limitations when starting and stopping the oscillator.

- When restarting the LOCO after it has been stopped, allow at least five cycles of the LOCO as an interval over which it is still stopped.
- Ensure that oscillation by the LOCO is stable when making the setting to stop the LOCO.
- Regardless of whether or not it is selected as the system clock, ensure that oscillation by the LOCO is stable before executing a WAIT instruction to place the chip on software standby.
- When a transition to software standby mode is to follow the setting to stop the LOCO, wait for at least three cycles of the LOCO after the setting to stop the LOCO and before executing the WAIT instruction.

While the LOCO is selected by the SCKCR3.CKSEL[2:0] bits, do not set the LCSTP bit to 1 (LOCO is stopped).

While low-speed operating mode is selected by the SOPCCR.SOPCM bit, do not set the LCSTP bit to 0 (LOCO is operating).

Downloaded from Arrow.com.

9.2.10 IWDT-Dedicated On-Chip Oscillator Control Register (ILOCOCR)

Address(es): 0008 0035h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	ILCSTP
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	ILCSTP	IWDT-Dedicated On-Chip Oscillator Stop	0: IWDT-dedicated on-chip oscillator is operating. 1: IWDT-dedicated on-chip oscillator is stopped.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

When the IWDT start mode select bit in option function select register 0 (OFS0.IWDTSTRT) is 0 (IWDT is operating), the setting of this register is invalid; it is valid only when the OFS0.IWDTSTRT bit is set to 1 (IWDT is stopped). The ILCSTP bit cannot be changed from 0 (IWDT-dedicated on-chip oscillator is operating) to 1 (IWDT-dedicated on-chip oscillator is stopped) while ILOCOCR is valid.

ILCSTP Bit (IWDT-Dedicated On-Chip Oscillator Stop)

This bit runs or stops the IWDT-dedicated on-chip oscillator.

After the setting of the ILCSTP bit has been changed so that the IWDT-dedicated on-chip oscillator operates, supply of the clock is started the MCU internally after a fixed time corresponding to the IWDT-dedicated clock oscillation stabilization time (t_{ILOCO}) has elapsed.

If the IWDT-dedicated clock is to be used, only start using the oscillator after this wait time has elapsed.

Ensure that oscillation by the IWDT-dedicated on-chip oscillator is stable before executing a WAIT instruction to place the chip on software standby mode.

Downloaded from Arrow.com.

9.2.11 High-Speed On-Chip Oscillator Control Register (HOCOCR)

Address(es): 0008 0036h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	HCSTP
Value after reset:	0	0	0	0	0	0	0	0/1*1

Bit	Symbol	Bit Name	Description	R/W
b0	HCSTP	HOCO Stop	0: HOCO is operating. 1: HOCO is stopped.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
Note 1. The HCSTP bit value after a reset is 0 when the HOCO oscillation enable bit in option function select register 1 (OFS1.HOCOEN) is 0. The HCSTP bit value after a reset is 1 when the OFS1.HOCOEN bit is 1.

HCSTP Bit (HOCO Stop)

This bit runs or stops the HOCO.

When changing the HCSTP bit from 1 to 0 (i.e. changing the HOCO clock from stopped to operating), confirm that the OSCOVFSR.HCOVF bit is 1 before switching the system clock to the HOCO clock.

That is, a fixed time for stabilization of oscillation is required after the setting for HOCO operation. A fixed time is also required for oscillation to stop after the setting to stop the oscillator. Accordingly, take note of the following limitations when starting and stopping the oscillator.

- After stopping the HOCO, confirm that the OSCOVFSR.HCOVF bit is 0 before restarting the HOCO.
- Confirm that the HOCO is operating and that the OSCOVFSR.HCOVF bit is 1 before stopping the HOCO.
- Regardless of whether or not it is selected as the system clock, confirm that the OSCOVFSR.HCOVF bit is 1 before executing a WAIT instruction to place the MCU in software standby mode.
- After stopping the HOCO, confirm that the OSCOVFSR.HCOVF bit is 0 and execute a WAIT instruction before entering software standby mode.

While the HOCO is selected by the SCKCR3.CKSEL[2:0] bits, do not set the HCSTP bit to 1 (HOCO is stopped).

While low-speed operating mode is selected by the SOPCCR.SOPCM bit, do not set the HCSTP bit to 0 (HOCO is operating).

Downloaded from Arrow.com.

9.2.12 High-Speed On-Chip Oscillator Control Register 2 (HOCOCR2)

Address(es): 0008 0037h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	HCFRQ[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	HCFRQ[1:0]	HOCO Frequency Setting	b1 b0 0 0: 32 MHz 1 1: 54 MHz Settings other than above are prohibited.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

High-speed on-chip control register 2 is used to set operation of the high-speed on-chip oscillator.
When the HOCOCR.HCSTP bit is 0 (HOCO is operating), do not write to the HOCOCR2 register.

HCFRQ[1:0] Bits (HOCO Frequency Setting)

These bits set the frequency of the HOCO.

Downloaded from Arrow.com.

9.2.13 Oscillation Stabilization Flag Register (OSCOVFSR)

Address(es): 0008 003Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	UPLOVF	—	HCOVF	PLOVF	—	MOOVF
Value after reset:	0	0	0	0	0/1*1	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MOOVF	Main Clock Oscillation Stabilization Flag	0: Main clock is stopped 1: Oscillation is stable and the clock can be used as the system clock*2	R
b1	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b2	PLOVF	PLL Clock Oscillation Stabilization Flag	0: PLL is stopped or not stabilized 1: Oscillation is stable and the clock can be used as the system clock	R
b3	HCOVF	HOCO Clock Oscillation Stabilization Flag	0: HOCO is stopped or not stabilized 1: Oscillation is stable and the clock can be used as the system clock*2	R
b4	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5	UPLOVF	USB-Dedicated PLL Clock Oscillation Stabilization Flag	0: USB-dedicated PLL is stopped or not stabilized 1: Oscillation is stable and the clock can be used as the UCLK	R
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. The HCOVF value after a reset is 1 when the HOCO oscillation enable bit in option function selection register 1 (OFS1.HOCOEN) is 0. The HCOVF value after a reset is 0 when the OFS1.HOCOEN bit is 1.
Note 2. When an appropriate value is set in the wait control register for each oscillator. If a set value (wait time) is not adequate, clock supply starts before oscillation becomes stable.

The OSCOVFSR register monitors whether oscillation of each oscillator has become stable.
If a wait control register is provided for each oscillator, specify a wait time that is longer than or equal to the stabilization time of the corresponding oscillation circuit.

MOOVF Flag (Main Clock Oscillation Stabilization Flag)

This flag indicates whether oscillation of the main clock is stable.
[Setting condition]

- After the MOSCCR.MOSTP bit is set to 0 (main clock oscillator is operating) when the MOSTP bit is 1 (main clock oscillator is stopped), the corresponding time set in the MOSCWTCR register has elapsed and supply of the main clock is started to the MCU internally.

[Clearing condition]

- After the MOSCCR.MOSTP bit is set to 1, the processing to stop the oscillation of the main clock oscillator is completed.

PLOVF Flag (PLL Clock Oscillation Stabilization Flag)

This flag indicates whether oscillation of the PLL clock is stable.
[Setting condition]
After the PLLCR2.PLLEN is set to 0 (PLL is operating) when the PLLEN bit is 1 (PLL is stopped), the MOOVF flag becomes 1, the PLL clock oscillation stabilization time (tPLL) has elapsed, and supply of the PLL clock is started to the MCU internally.
[Clearing condition]
After the PLLCR2.PLLEN bit is set to 1, the processing to stop the oscillation of the PLL is completed.

Downloaded from Arrow.com.

HCOVF Flag (HOCO Clock Oscillation Stabilization Flag)

This flag indicates whether oscillation of the HOCO clock is stable.

[Setting condition]

- After the HOCOCR.HCSTP bit is set to 0 (HOCO is operating) when the HCSTP bit is 1 (HOCO is stopped), supply of the HOCO clock is started to the MCU internally.

[Clearing condition]

- After the HOCOCR.HCSTP bit is set to 1, the processing to stop the oscillation of the HOCO is completed.

UPLOVF Flag (USB-Dedicated PLL Clock Oscillation Stabilization Flag)

This flag indicates whether oscillation of the USB-dedicated PLL clock is stable.

[Setting condition]

- After the UPLLCR2.UPLLEN bit is set to 0 (USB-dedicated PLL is operating) when the UPLLEN bit is 1 (USB-dedicated PLL is stopped), the MOOVF flag becomes 1 and the PLL clock stabilization wait time (tPLL) has elapsed, and then supply of the USB-dedicated PLL clock is started to the MCU internally.

[Clearing condition]

- After the UPLLCR2.UPLLEN bit is set to 1, the processing to stop the oscillation of the USB-dedicated PLL clock is completed.

Downloaded from Arrow.com.

9.2.14 Oscillation Stop Detection Control Register (OSTDCR)

Address(es): 0008 0040h

	b7	b6	b5	b4	b3	b2	b1	b0
	OSTDE	—	—	—	—	—	—	OSTDIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	OSTDIE	Oscillation Stop Detection Interrupt Enable	0: The oscillation stop detection interrupt is disabled. Oscillation stop detection is not notified to the POE. 1: The oscillation stop detection interrupt is enabled. Oscillation stop detection is notified to the POE.	R/W
b6 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	OSTDE	Oscillation Stop Detection Function Enable	0: Oscillation stop detection function is disabled. 1: Oscillation stop detection function is enabled.	R/W

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

OSTDIE Bit (Oscillation Stop Detection Interrupt Enable)

If the oscillation stop detection flag in the oscillation stop detection status register (OSTDSR.OSTDF) requires clearing, do this after setting the OSTDIE bit to 0. Wait for at least two cycles of PCLKB before again setting the OSTDIE bit to 1. According to the number of cycles for access to read a given I/O register, wait time longer than two cycles of PCLKB may have to be secured.

OSTDE Bit (Oscillation Stop Detection Function Enable)

This bit enables or disables the oscillation stop detection function.

When the OSTDE bit is 1 (oscillation stop detection function enabled), the LOCO stop bit (LOCOCR.LCSTP) is set to 0 and the LOCO operation is started. The LOCO cannot be stopped while the oscillation stop detection function is enabled; writing 1 (LOCO is stopped) to the LOCOCR.LCSTP bit is invalid.

When the oscillation stop detection flag in the oscillation stop detection status register (OSTDSR.OSTDF) is 1 (main clock oscillation stop has been detected), writing 0 to the OSTDE bit is invalid.

When the OSTDE bit is 1, a transition cannot be made to software standby mode. To make a transition to software standby mode, execute the WAIT instruction with the OSTDE bit being 0.

Downloaded from Arrow.com.

9.2.15 Oscillation Stop Detection Status Register (OSTDSR)

Address(es): 0008 0041h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	OSTDF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	OSTDF	Oscillation Stop Detection Flag	0: The main clock oscillation stop has not been detected. 1: The main clock oscillation stop has been detected.	R/(W) *1
b7 to b1	—	Reserved	These bits are read as 0 and cannot be modified.	R

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
Note 1. This bit can only be set to 0.

OSTDF Flag (Oscillation Stop Detection Flag)

This bit is a flag to indicate the main clock status. When the OSTDF flag is 1, it indicates that the main clock oscillation stop has been detected.

Once the main clock oscillation stop is detected, the OSTDF flag is not set to 0 even though the main clock oscillation is restarted. The OSTDF flag is set to 0 by reading 1 from the bit and then writing 0. At least three ICLK cycles of wait time is necessary between writing 0 to the OSTDF flag and reading the OSTDF flag as 0. If the OSTDF flag is set to 0 while the main clock oscillation is stopped, the OSTDF flag becomes 0 and then returns to 1.

When the main clock oscillator (010b) or PLL (100b) is selected by the clock source select bits in system clock control register 3 (SCKCR3.CKSEL[2:0]), the OSTDF flag cannot be modified to 0. The OSTDF flag should be set to 0 after switching the clock source to a source other than the main clock oscillator and the PLL.

[Setting condition]

- The main clock oscillation is stopped with the OSTDCR.OSTDE bit being 1 (oscillation stop detection function enabled).

[Clearing condition]

- 1 is read and then 0 is written when the SCKCR3.CKSEL[2:0] bits are neither 010b nor 100b.

Downloaded from Arrow.com.

9.2.16 Main Clock Oscillator Wait Control Register (MOSCWTCR)

Address(es): 0008 00A2h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	MSTS[4:0]				
Value after reset:	0	0	0	0	0	1	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	MSTS[4:0]	Main Clock Oscillator Wait Time	b4 b0 0 0 0 0 0: Wait time = 2 cycles (0.5 µs) 0 0 0 0 1: Wait time = 1024 cycles (256 µs) 0 0 0 1 0: Wait time = 2048 cycles (512 µs) 0 0 0 1 1: Wait time = 4096 cycles (1.024 ms) 0 0 1 0 0: Wait time = 8192 cycles (2.048 ms) 0 0 1 0 1: Wait time = 16384 cycles (4.096 ms) 0 0 1 1 0: Wait time = 32768 cycles (8.192 ms) 0 0 1 1 1: Wait time = 65536 cycles (16.384 ms) Settings other than above are prohibited. Wait time when LOCO = 4.0 MHz (0.25 µs, TYP.)	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

MSTS[4:0] Bits (Main Clock Oscillator Wait Time)

Set these bits to select the oscillation stabilization wait time of the main clock oscillator.

Set the main clock oscillation stabilization time to longer than or equal to the stabilization time recommended by the oscillator manufacturer. When the main clock is externally input, set these bits to 00000b because the oscillation stabilization time is not required.

The wait time set by the MSTS[4:0] bits is counted using the LOCO clock. The LOCO automatically oscillates when necessary, regardless of the value of the LOCOCR.LCSTP bit.

After the set wait time has elapsed, supply of the main clock is started to the MCU internally and the OSCOVFSR.MOOVF flag becomes 1. If the set wait time is short, supply of the main clock is started before oscillation of the clock becomes stable.

Only rewrite the MOSCWTCR register when the MOSCCR.MOSTP bit is 1 and the OSCOVFSR.MOOVF flag is 0. Do not rewrite this register under any other conditions.

Downloaded from Arrow.com.

9.2.17 CLKOUT Output Control Register (CKOCR)

Address(es): 0008 003Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CKOSTP	CKODIV[2:0]			CKOSEL[3:0]				—	—	—	—	—	—	—	—
Value after reset:	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11 to b8	CKOSEL[3:0]	CLKOUT Output Source Select	b11 b8 0 0 0 0: LOCO clock 0 0 0 1: HOCO clock 0 0 1 0: Main clock oscillator 0 0 1 1: Sub-clock oscillator 0 1 0 0: PLL Settings other than above are prohibited.	R/W
b14 to b12	CKODIV[2:0]	CLKOUT Output Division Ratio Select	b14 b2 0 0 0: No division 0 0 1: ×1/2 0 1 0: ×1/4 0 1 1: ×1/8 1 0 0: ×1/16 Settings other than above are prohibited.	R/W
b15	CKOSTP	CLKOUT Output Stop Control	0: CLKOUT pin output enabled*1 1: CLKOUT pin output disabled (always output the low level)	R/W

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
Note 1. It is also necessary to set the pin function control register and port mode register for the corresponding pin.

CKOSEL[3:0] Bits (CLKOUT Output Source Select)

Set these bits to select the LOCO clock, HOCO clock, main clock, sub-clock, or PLL as the source of the clock to be output from the CLKOUT pin.

CKODIV[2:0] Bits (CLKOUT Output Division Ratio Select)

Set these bits to select the division ratio of the clock selected by the CKOSEL[3:0] bits.
Set the CKOSTP bit to 1 when changing the division ratio.
The division ratio of the output clock frequency should be set to no higher than 8 MHz when VCC is 2.7 V or above, and no higher than 4 MHz when VCC is below 2.7 V.
For details on the characteristics of the clock output from the CLKOUT pin, see Table 51.46, Timing of CLKOUT_RF.

CKOSTP Bit (CLKOUT Output Stop Control)

Set this bit to enable or disable output from the CLKOUT pin.
When this bit is set to 1, the selected clock is output. When this bit is set to 1, a low level is output.
If the CKOSTP bit is rewritten while the clock is still oscillating, a glitch may be generated in the output.

Downloaded from Arrow.com.

9.2.18 Main Clock Oscillator Forced Oscillation Control Register (MOFCR)

Address(es): 0008 C293h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	MOSEL	MODRV21	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5	MODRV21	Main Clock Oscillator Drive Capability Switch	VCC ≥ 2.4 V 0: 1 MHz or higher and lower than 10 MHz 1: 10 MHz to 20 MHz VCC < 2.4 V 0: 1 MHz to 8 MHz 1: Setting prohibited	R/W
b6	MOSEL	Main Clock Oscillator Switch	0: Resonator 1: External oscillator input	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

The EXTAL/XTAL pin is also used as a port. In the initial setting state, the pin is set as a port.

MODRV21 Bit (Main Clock Oscillator Drive Capability Switch)

These bits select the drive capability of the main clock oscillator.

MOSEL Bit (Main Clock Oscillator Switch)

This bit selects the oscillation source of the main clock oscillator.

Downloaded from Arrow.com.

9.2.19 Memory Wait Cycle Setting Register (MEMWAIT)

Address(es): 0008 0031h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	MEMWAIT
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MEMWAIT	Memory Wait Cycle Setting*1	0: No wait states 1: Wait states	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
Note 1. Do not select the MEMWAIT bit = 0 (no wait states) when divided by 1 is selected by the SCKCR.ICK[3:0] bits and a clock of frequency is higher than 32 MHz is selected as the system clock (ICLK) by the SCKCR3.CKSEL[2:0] bits. When a clock of frequency is lower than 32 MHz is selected as the ICLK, it is not necessary to set the MEMWAIT bit to 1 (wait states).

The MEMWAIT register is used to control the wait cycle of the ROM.

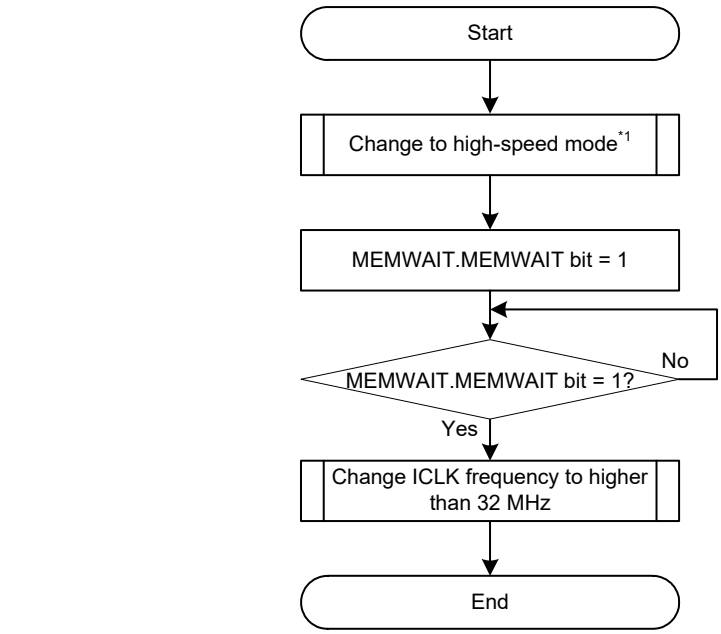
MEMWAIT Bit (Memory Wait Cycle Setting)

This bit is used to set the wait cycle of the ROM.
This bit is set to “no wait states” immediately after a reset.
When selecting a clock of frequency higher than 32 MHz as the system clock (ICLK), set the bit to 1 (wait sates).
When setting the MEMWAIT bit to 1 (wait states), make sure that high-speed mode is selected. After the value of the MEMWAIT bit is changed to 1, change the system clock to a clock of frequency higher than 32 MHz.
When setting the MEMWAIT bit to 0 (no wait states), make sure that the frequency of the system clock (ICLK) is 32 MHz or lower. When changing the operating power control state, make sure that the value of the MEMWAIT bit is changed to 0.
Table 9.3 lists the restrictions on setting the MEMWAIT bit, and Figure 9.3 and Figure 9.4 show the procedure for changing the MEMWAIT bit.

Table 9.3 Restrictions on Setting the MEMWAIT Bit

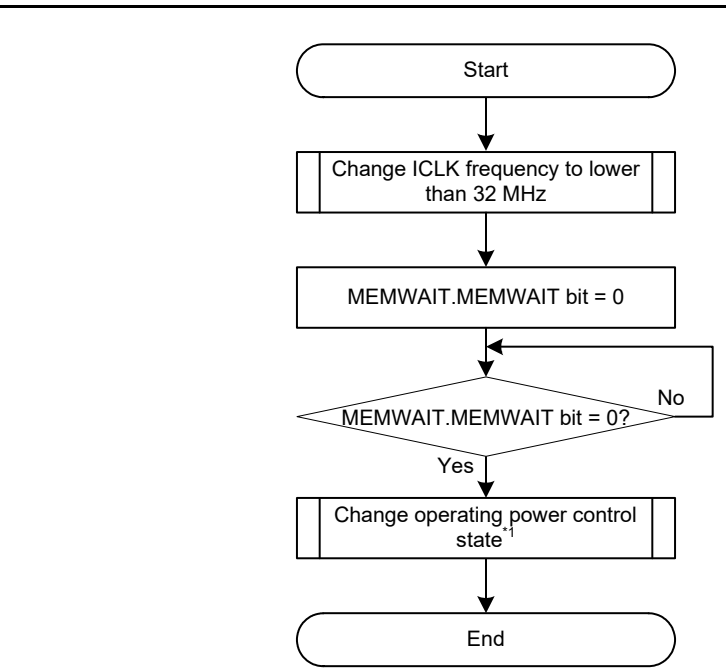
	Operating Power Control State			
	High-Speed Operating Mode			
MEMWAIT Bit	ICLK ≤ 32 MHz	ICLK > 32 MHz	Middle-Speed Operating Mode	Low-Speed Operating Mode
0	Can be set	Cannot be set	Can be set	Can be set
1	Can be set	Can be set	Cannot be set	Cannot be set

Downloaded from Arrow.com.



Note 1. Resetting is not necessary in high-speed mode.

Figure 9.3 Example of MEMWAIT Bit Setting Procedure When Changing ICLK Frequency to Higher than 32 MHz



Note 1. Resetting is not necessary when the state is not changed.

Figure 9.4 Example of MEMWAIT Bit Setting Procedure When Changing ICLK Frequency to Lower than 32 MHz

Downloaded from Arrow.com.

9.2.20 Low-Speed On-Chip Oscillator Trimming Register (LOCOTRR)

Address(es): 0008 0060h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	LOCOTRD[4:0]				
Value after reset:	0	0	0	*1	*1	*1	*1	*1

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	LOCOTRD[4:0]	Low-Speed On-Chip Oscillator Frequency Adjustment	b4 b0 1 0 0 0 0: -16 (Frequency: Low) 1 0 0 0 1: -15 : : 0 1 1 1 0: 14 0 1 1 1 1: 15 (Frequency: High)	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
Note 1. Chip-specific value

LOCOTRD[4:0] Bits (Low-Speed On-Chip Oscillator Frequency Adjustment)

Set the frequency adjustment value for the low-speed on-chip oscillator.
The setting range is -16 (10h) to 15 (0Fh) by two's complements. The greater the set value is, the higher the frequency is. The frequency is adjusted under certain conditions before shipment, so the value after reset varies with the chip. After a reset, the oscillation frequency returns to the factory default.

9.2.21 IWDT-Dedicated On-Chip Oscillator Trimming Register (ILOCOTRR)

Address(es): 0008 0064h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	ILOCOTRD[4:0]				
Value after reset:	0	0	0	*1	*1	*1	*1	*1

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	ILOCOTRD[4:0]	IWDT-Dedicated On-Chip Oscillator Frequency Adjustment	b4 b0 0 0 0 0 0: 0 (Frequency: Low) 0 0 0 0 1: 1 : : 1 1 1 1 0: 30 1 1 1 1 1: 31 (Frequency: High)	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
Note 1. Chip-specific value

ILOCOTRD[4:0] Bits (IWDT-Dedicated On-Chip Oscillator Frequency Adjustment)

Set the frequency adjustment value for the IWDT-dedicated on-chip oscillator.
The setting range is from 0 (00h) to 31 (1Fh) by binary numbers. The greater the set value is, the higher the frequency is. The frequency is adjusted under certain conditions before shipment, so the value after reset varies with the chip. After a reset, the oscillation frequency returns to the factory default.

Downloaded from Arrow.com.

9.2.22 High-Speed On-Chip Oscillator Trimming Register n (HOCOTRRn) (n = 0, 3)

Address(es): HOCOTRR0 0008 0068h, HOCOTRR3 0008 006Bh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	HOCOTRD[5:0]					
Value after reset:	0	0	*1	*1	*1	*1	*1	*1

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	HOCOTRD[5:0]	High-Speed On-Chip Oscillator Frequency Adjustment	b5 b0 0 0 0 0 0 0: 0 (Frequency: Low) 0 0 0 0 0 1: 1 : : 1 1 1 1 1 0: 62 1 1 1 1 1 1: 63 (Frequency: High)	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
Note 1. Chip-specific value

HOCOTRR0 corresponds to 32 MHz, and HOCOTRR3 corresponds to 54 MHz.

HOCOTRD[5:0] Bits (High-Speed On-Chip Oscillator Frequency Adjustment)

Set the frequency adjustment value for the high-speed on-chip oscillator.

The setting range is from 0 (00h) to 63 (3Fh) by binary numbers. The greater the set value is, the higher the frequency is.

The frequency is adjusted under certain conditions before shipment, so the value after reset varies with the chip. After a reset, the oscillation frequency returns to the factory default.

Downloaded from Arrow.com.

9.3 Main Clock Oscillator

There are two ways of supplying the clock signal from the main clock oscillator: connecting an oscillator or the input of an external clock signal.

9.3.1 Connecting a Crystal

Figure 9.5 shows an example of connecting a crystal.
A damping resistor (Rd) should be added, if necessary. Since the resistor values vary depending on the resonator and the oscillation drive capability, use values recommended by the resonator manufacturer. If use of an external feedback resistor (Rf) is directed by the resonator manufacturer, insert an Rf between EXTAL and XTAL by following the instruction.
When connecting a resonator to supply the clock, the frequency of the resonator should be in the frequency range of the resonator for the main clock oscillator described in Table 9.1.

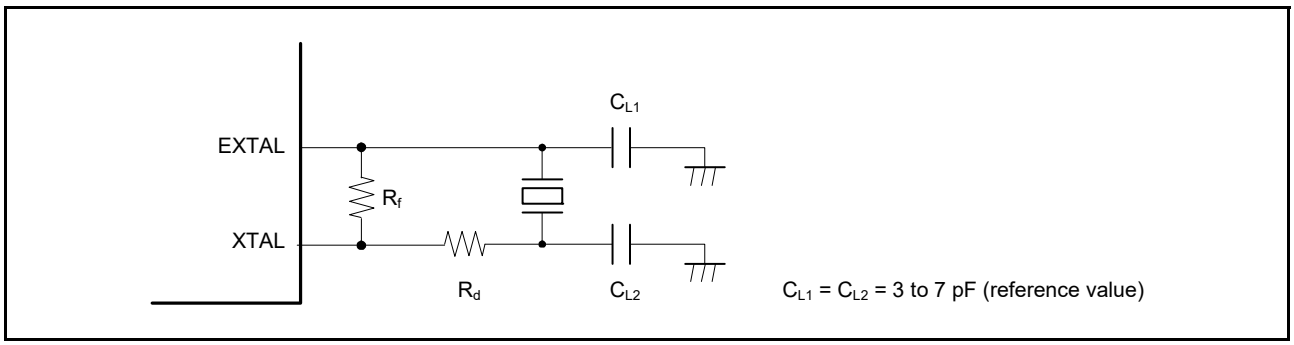


Figure 9.5 Example of Crystal Connection

Table 9.4 Damping Resistance (Reference Values)

Frequency (MHz)	2	8	16	20
Rd (Ω)	0	0	0	0

Figure 9.6 shows an equivalent circuit of the crystal. Use a crystal that has the characteristics shown in Table 9.5 as a reference.

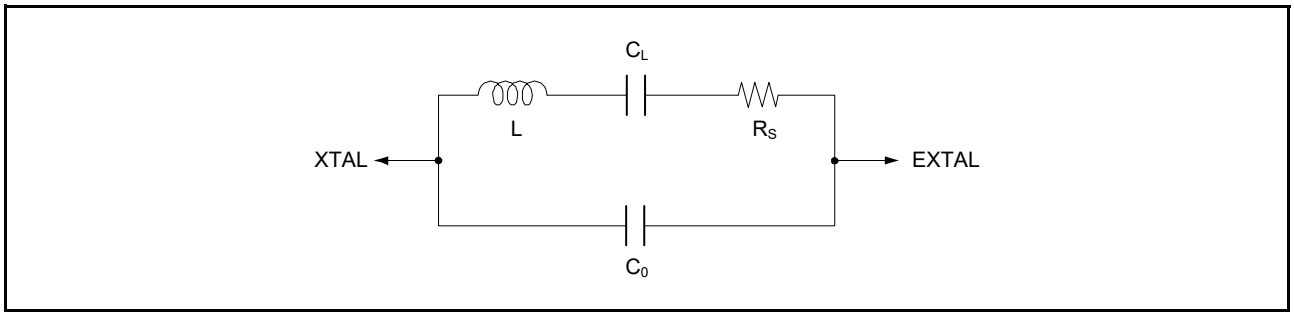


Figure 9.6 Equivalent Circuit of Crystal

Table 9.5 Crystal Characteristics (Reference Values)

Frequency (MHz)	8	12	16
R_S max (Ω)	200	120	56
C_0 max (pF)	1.3	1.3	1.4

Downloaded from Arrow.com.

9.3.2 External Clock Input

Figure 9.7 shows connection of an external clock. If operation is to be driven by an external clock, set the MOFCR.MOSEL bit to 1 and leave the XTAL pin open-circuit.

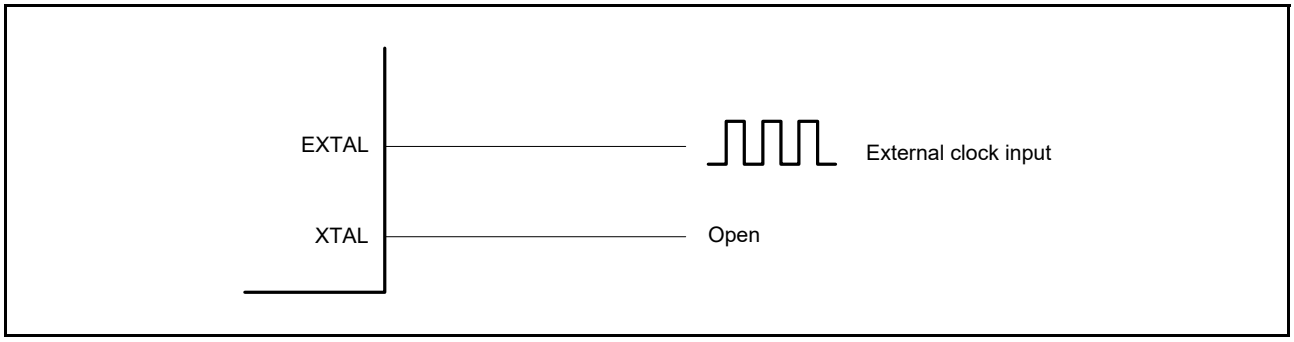


Figure 9.7 Connection Example of External Clock

9.3.3 Handling of Pins When the Main Clock is Not Used

For details on pin handling when the main clock is not used, refer to section 21.5, Handling of Unused Pins.

9.3.4 Notes on the External Clock Input

The frequency of the external clock input can only be changed while the main clock oscillator is stopped. Do not change the frequency of the external clock input while the setting of the main clock oscillator stop bit (MOSCCR.MOSTP) is 0 (main clock oscillator is operating).

Downloaded from Arrow.com.

9.4 Sub-Clock Oscillator

The only way of supplying the clock signal from the sub-clock oscillator is connecting a crystal.

9.4.1 Connecting 32.768-kHz Crystal

To supply a clock to the sub-clock oscillator, connect a 32.768-kHz crystal, as shown in Figure 9.8.
A damping resistor Rd should be added, if necessary. Since the resistor values vary depending on the resonator and the oscillation drive capability, use values recommended by the resonator manufacturer. If use of an external feedback resistor (Rf) is directed by the resonator manufacturer, insert an Rf between XCIN and XCOUT by following the instruction. When connecting a resonator to supply the clock, the frequency of the resonator should be in the frequency range of the resonator for the sub-clock oscillator described in Table 9.1.

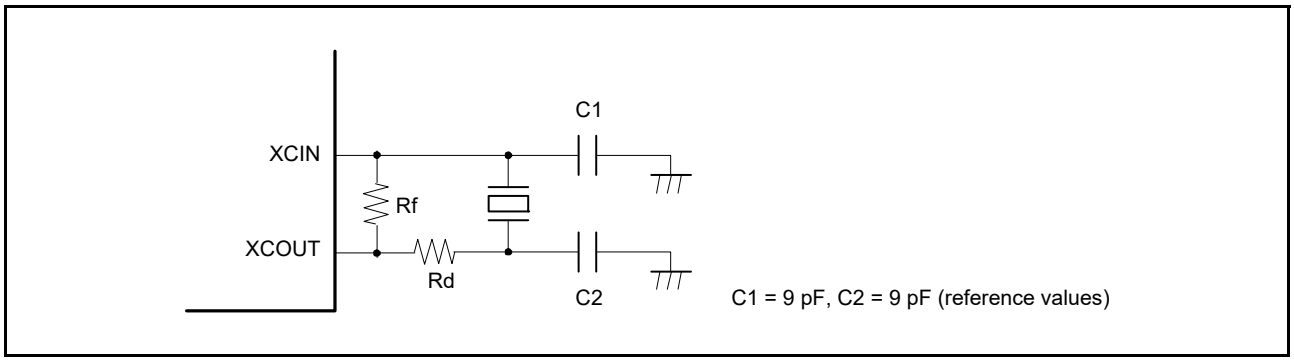


Figure 9.8 Connection Example of 32.768-kHz Crystal

Figure 9.9 shows an equivalent circuit for the 32.768-kHz crystal. Use a crystal that has the characteristics listed in Table 9.6.

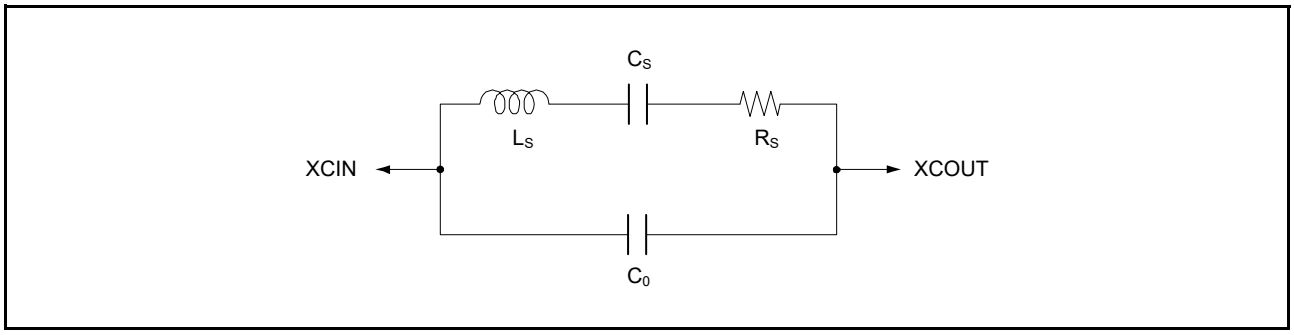


Figure 9.9 Equivalent Circuit for Crystal

Table 9.6 Crystal Characteristics (Reference Values)

Frequency (kHz)	32.768 (Low CL)
R_S max (kΩ)	37
C_0 max (pF)	0.9

Downloaded from Arrow.com.

9.4.2 Handling of Pins When Sub-Clock is Not Used

If the sub-clock is not in use, connect the XCIN pin to VSS via a resistor (to pull VSS down) and leave the XCOUT pin open-circuit as shown in Figure 9.10.
In addition, if an oscillator is not connected, set the sub-clock oscillator stop bit (SOSCCR.SOSTP) to 1 (stopping the oscillator) and the sub-clock oscillator control bit in RTC control register 3 (RCR3.RTCEN) to 0 (stopping the sub-clock oscillator). The value of some RTC registers related to the sub-clock will be undefined after a cold start. Accordingly, be sure to set these bits after a cold start.

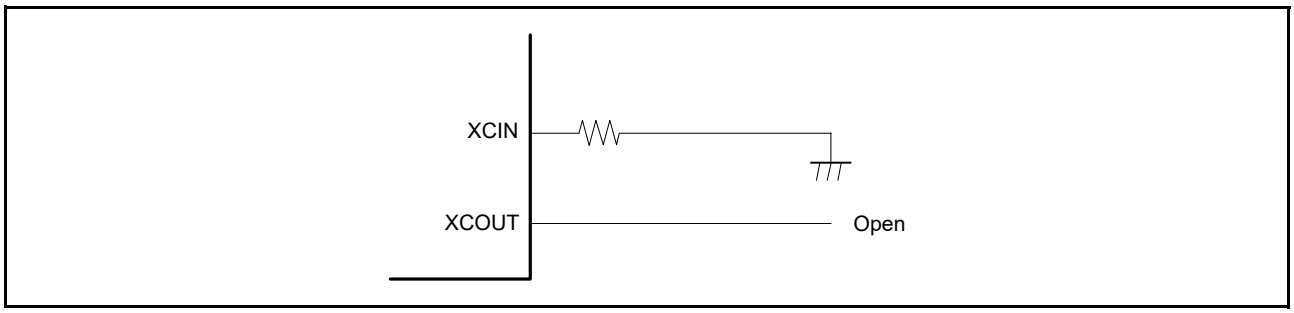


Figure 9.10 Pin Handling when Sub-Clock is Not Used

Downloaded from Arrow.com.

9.5 Dedicated Clock Oscillator for Bluetooth

In the case of the 85-pin BGA and 56-pin QFN products, operating Bluetooth requires connection to the 32-MHz clock oscillator. The 83-pin LGA product does not require the connection of a resonator because the package includes a 32-MHz crystal resonator. Note that handling of the XTAL1_RF and XTAL2_RF pins of the 83-pin LGA package product differs from that of the other products in that these pins should be externally connected to each other rather than to a resonator.

9.5.1 Connecting the Oscillator

An example of connection of the oscillator is shown in Figure 9.11.
In the case of the 85-pin BGA and 56-pin QFN package products, insert a damping resistor (Rd) as required. Set the resistance to the value recommended by the oscillator manufacturer as the value depends on the oscillator and its driving capability. If the oscillator manufacturer states that the addition of a feedback resistor (Rf) to the oscillator is required, insert Rf between XTAL1_RF and XTAL2_RF according to the instructions.
In the case of an 83-pin LGA package product, XTAL1_RF and XTAL2_RF should be externally connected to each other.
To control the oscillator, use the Bluetooth middleware provided by Renesas. The Bluetooth middleware is also able to control the settings of the on-chip variable capacitors, CL1 and CL2, to adjust the frequency of oscillation. Adjustment of the frequency of the Bluetooth-dedicated clock oscillator is explained in the application note Procedure for Adjusting the Frequency of the Bluetooth-Dedicated Clock Oscillator (R01AN4762). Obtain the latest version of this document from the Renesas website.

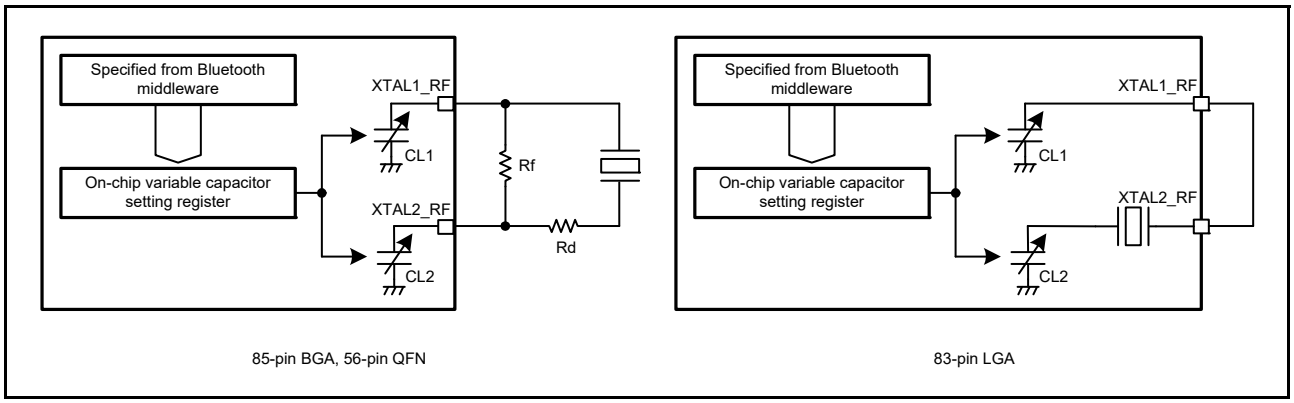


Figure 9.11 Example of the Connection of a 32-MHz Crystal Resonator

9.5.2 Connecting the Bluetooth-Dedicated Clock Output Pin

When the Bluetooth middleware sets up the clock output and the BLE shifts from waiting mode to RF power-down mode, the frequency-divided clock for the Bluetooth-dedicated clock is output. If the BLE subsequently shifts to RF low-power mode, the RF clock output is disabled. The Bluetooth middleware is able to specify the frequency-divisor for the Bluetooth-dedicated clock.
The Bluetooth-dedicated clock output can also be used as the input of the external clock for the main clock oscillator by externally connecting the CLKOUT_RF and EXTAL pins on the board. Figure 9.11 shows an example of a configuration where the Bluetooth-dedicated clock output is used as the input of the external clock for the main clock oscillator.

Downloaded from Arrow.com.

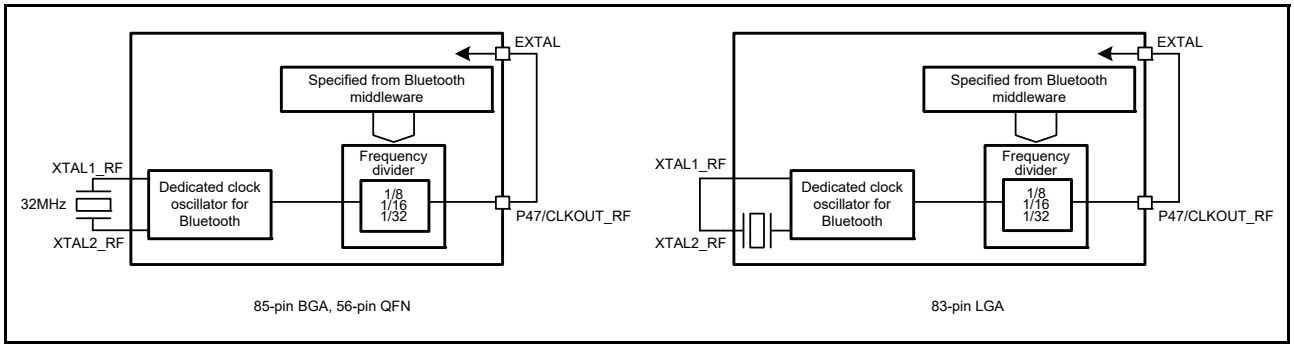


Figure 9.12 Example of Connection of the Bluetooth-dedicated clock output pin

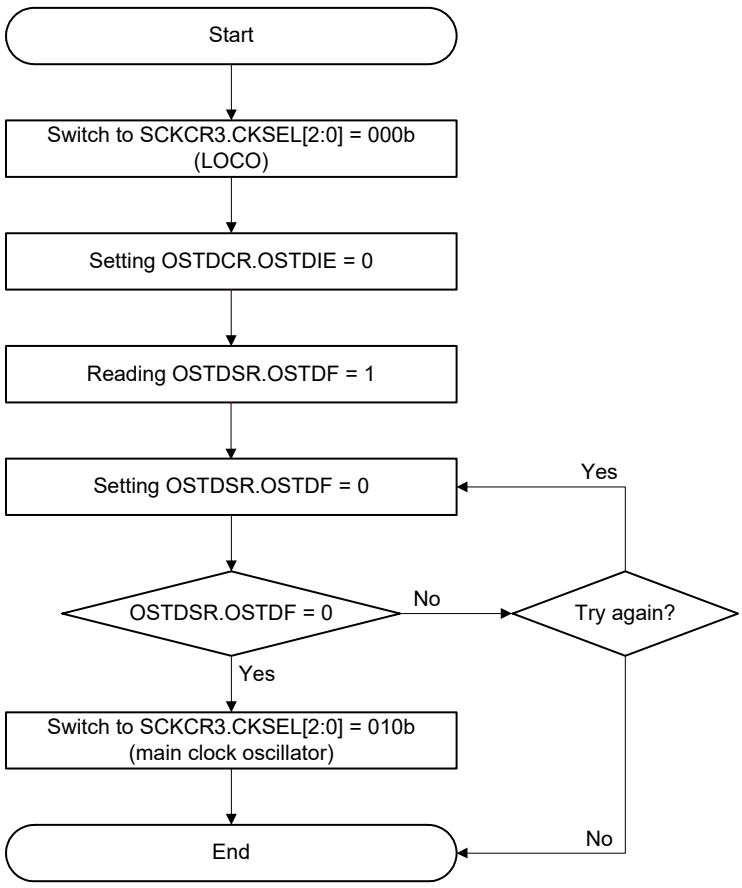
Downloaded from Arrow.com.

9.6 Oscillation Stop Detection Function

9.6.1 Oscillation Stop Detection and Operation after Detection

The oscillation stop detection function is used to detect the main clock oscillator stop and to supply LOCO clock pulses from the low-speed on-chip oscillator as the system clock source instead of the main clock.
An oscillation stop detection interrupt request can be generated when an oscillation stop is detected. In addition, the MTU output can be forcedly driven to the high-impedance on the detection. For details, refer to section 23, Multi-Function Timer Pulse Unit 2 (MTU2a) and section 24, Port Output Enable 2 (POE2a).
In the MCU, the main clock oscillation stop is detected when the input clock remains to be 0 or 1 for a certain period, for example, due to a malfunction of the main clock oscillator (refer to section 51, Electrical Characteristics).
When an oscillation stop is detected, the main clock selected by the clock source select bits (SCKCR3.CKSEL[2:0]) is switched to the LOCO clock by the corresponding selectors in the former stage. Therefore, if an oscillation stop is detected with the main clock selected as the system clock source, the system clock source is switched to the LOCO clock without a change of CKSEL[2:0].
If an oscillation stop is detected while the PLL clock is selected by the clock source select bits (SCKCR3.CKSEL[2:0]) in system clock control register 3, the SCKCR3.CKSEL[2:0] bit value does not change and the PLL clock remains the system clock source. However, the frequency becomes a free-running oscillation frequency.
Also, if an oscillation stop is detected while the USB-dedicated PLL clock is selected by the UPLLCR.UCKPLLSEL bit, the set value of these bits does not change and the USB-dedicated PLL clock remains the UCLK clock source. However, the frequency becomes a free-running oscillation frequency.
Switching between the main clock and LOCO clock is controlled by the oscillation stop detection flag (OSTDSR.OSTDF). The clock source is switched to the LOCO clock when the OSTDF flag is 1, and is switched to the main clock again when the OSTDF flag is set to 0. At this time, if the main clock or PLL clock is selected with the CKSEL[2:0] bits, the OSTDF flag cannot be set to 0. To switch the clock source to the main clock or PLL clock again after the oscillation stop detection, set the CKSEL[2:0] bits to a clock source other than the main clock or PLL clock and set the OSTDF flag to 0. After that, check that the OSTDF flag is not 1, and then set the CKSEL[2:0] bits to the main clock or PLL clock after the specified oscillation stabilization time has elapsed.
After a reset is released, the main clock oscillator is stopped and the oscillation stop detection function is disabled. To enable the oscillation stop detection function, activate the main clock oscillator and write 1 to the oscillation stop detection function enable bit (OSTDCR.OSTDE) after a specified oscillation stabilization time has elapsed.
The oscillation stop detection function is provided against the main clock stop by an external cause. Therefore, the oscillation stop detection function should be disabled before the main clock oscillator is stopped by the software or a transition is made to software standby mode.
When the system clock with the main clock selected as the system clock source, CAC main clock (CACMCLK), SSI clock (SSISCK), and CAN clock (CANMCLK) are selected, these clocks are switched to the LOCO clock by the oscillation stop detection. The system clock (ICLK) frequency during the LOCO clock operation is specified by the LOCO oscillation frequency and the division ratio set by the system clock (ICLK) select bits (SCKCR.ICK[3:0]).
When the PLL clock is selected as the system clock source and the USB-dedicated PLL clock is selected as the UCLK source, these clocks operate at the PLL free-running frequency by the oscillation stop detection.

Downloaded from Arrow.com.



Note: On return from the oscillation-stopped state, the factor responsible for stopping the main clock oscillation circuit must be removed on the user system to allow the return of oscillation.

Figure 9.13 Flow of Recovery from Detection of Oscillator Stop

9.6.2 Oscillation Stop Detection Interrupts

The main clock oscillator stop is notified to port output enable 23 (POE) if the oscillation-stop detection flag (OSTDSR.OSTDF) becomes 1 while the oscillation-stop detection interrupt enable bit (OSTDCR.OSTDIE) is 1 (oscillation stop detection interrupt enabled). On accepting the notification of the oscillation stop, the POE sets the OSTST high-impedance flag in input level control/status register 3 (ICSR3.OSTSTF) to 1. After the oscillation stop is detected, wait for at lease 10 cycles of PCLK before writing to this ICSR3.OSTSTF flag. When the OSTDSR.OSTDF flag requires clearing, do so setting the oscillation stop detection interrupt enable bit (OSTDCR.OSTDIE) to 0. Wait for at least two cycles of PCLKB clock before again setting the OSTDCR.OSTDIE bit to 1. According to the number of cycles for access to read a given I/O register, wait time longer than two cycles of PCLKB may have to be secured.

The oscillation stop detection interrupt is a non-maskable interrupt. Since non-maskable interrupts are disabled in the initial state after a reset release, enable the non-maskable interrupts by the software before using oscillation stop detection interrupts. For details, refer to section 15, Interrupt Controller (ICUb).

When the PLL detects an oscillation stop and is running at its own oscillation frequency, this indicates the occurrence of some system failure. An emergency measure should be taken to handle the failure.

Downloaded from Arrow.com.

9.7 PLL Circuit

The PLL circuit has a function to multiply the frequency from the oscillator.

9.8 Internal Clock

Clock sources of internal clock signals are the main clock, sub-clock, HOCO clock, LOCO clock, PLL clock, USB-dedicated PLL clock, and dedicated low-speed clock for the IWDT. The internal clocks listed below are produced from these sources.

(1) Operating clock of the CPU, DMAC, DTC, ROM, and RAM: System clock (ICLK)
(2) Operating clock of peripheral modules: Peripheral module clock (PCLKA, PCLKB, and PCLKD)
(3) Operating clock of the FlashIF: FlashIF clock (FCLK)
(4) Operating clock of USB modules: USB clock (UCLK)
(5) Operating clock of CAN modules: CAN clock (CANMCLK)
(6) Operating clock for the CAC: CAC clock (CACCLK)
(7) Operating clock for the RTC: RTC-dedicated sub-clock (RTCSCLK)
(8) Operating clock for the IWDT: IWDT-dedicated low-speed clock (IWDTCLK)
(9) Operating clock for the SSI: SSI clock (SSISCK)
(10) Operating clocks for BLE: The Bluetooth-dedicated clock (BLECLK), and the Bluetooth-dedicated low-speed clock (BLELOCO)
(11) Operating clock for the low-power timer: LPT clock (LPTCLK)

Frequencies of the internal clocks are set by the combination of the divisors selected by the SCKCR.FCK[3:0], ICK[3:0], PCKA[3:0], PCKB[3:0], and PCKD[3:0] bits, the clock source selected by the SCKCR3.CKSEL[2:0] bits, and the bits that select the frequency of the PLL circuit (PLLCR.STC[5:0] and PLIDIV[1:0] bits, UPLLCR.UPLIDIV[1:0], USTC[5:0] bits, and HOCOCR2.HCFRQ[1:0] bits). If the value of any of these bits is changed, subsequent operation will be at the frequency determined by the new value.

9.8.1 System Clock

The system clock (ICLK) is used as the operating clock of the CPU, DMAC, DTC, ROM, and RAM.
The ICLK frequency is specified by the SCKCR.ICK[3:0] bits, and the SCKCR3.CKSEL[2:0] bits, and the PLLCR.STC[5:0] and PLIDIV[1:0] bits.

9.8.2 Peripheral Module Clock

The peripheral module clocks (PCLKA, PCLKB, and PCLKD) are the operating clocks for use by peripheral modules.
The PCLKA, PCLKB, and PCLKD frequencies are specified by the SCKCR.PCKA[3:0], PCKB[3:0], and PCKD[3:0] bits, the SCKCR3.CKSEL[2:0] bits, and the PLLCR.STC[5:0] and PLIDIV[1:0] bits.
The peripheral module clock used as the operating clock is PCLKD for S12AD, and PCLKA and PCLKB are for other modules.

9.8.3 FlashIF Clock

The FlashIF clock (FCLK) is used as the operating clock of the FlashIF.
The FCLK frequency is specified by the SCKCR.FCK[3:0] bits, and the SCKCR3.CKSEL[2:0] bits, and the PLLCR.STC[5:0] and PLIDIV[1:0] bits.

Downloaded from Arrow.com.

9.8.4 USB Clock

The USB clock (UCLK) is an operating clock for the USB module. The UCLK frequency is specified by the SCKCR3.CKSEL[2:0] bits, and the PLLCR.STC[5:0], PLIDIV[1:0] bits, and the UPLLCR.UPLIDIV[1:0] and USTC[5:0] bits.
A 48-MHz clock must be supplied to the USB module. When the USB module is used, setting must be made so that UCLK is 48 MHz.

9.8.5 CAN Clock

The CAN clock (CANMCLK) is an operating clock for the CAN module.
CANMCLK is generated by the main clock oscillator.

9.8.6 CAC Clock

The CAC clock (CACCLK) is an operating clock for the CAC module.
The CACCLK clocks include CACMCLK which is generated by the main clock oscillator, CACSCLK which is generated by the sub-clock oscillator, CACHCLK which is generated by the high-speed on-chip oscillator, CACLCLK which is generated by the low-speed on-chip oscillator, and CACILCLK which is generated by the IWDT-dedicated on-chip oscillator.

9.8.7 RTC-Dedicated Clock

The RTC-dedicated clock (RTCSCLK) is the operating clock for the RTC.
RTCSCLK is generated by the sub-clock oscillator.

9.8.8 IWDT-Dedicated Clock

The IWDT-dedicated clock (IWDTCLK) is the operating clock for the IWDT.
IWDTCLK is internally generated by the IWDT-dedicated on-chip oscillator.

9.8.9 SSI Clock

The SSI clock (SSISCK) is the operating clock for the SSI.
SSISCK is generated by the main clock oscillator.

9.8.10 Clocks for BLE

The Bluetooth-dedicated clock (BLECLK) and the Bluetooth-dedicated low-speed clock (BLELOCO) are the operating clocks for the BLE. To control these clocks, use the Bluetooth middleware provided by Renesas.

9.8.11 Low-Power Timer Clock

The low-power timer clock (LPTCLK) is an operating clock for the low-power timer. The LPTCLK clocks include a clock generated by the sub-clock oscillator and a clock generated by the IWDT-dedicated on-chip oscillator.

Downloaded from Arrow.com.

9.9 Usage Notes

9.9.1 Notes on Clock Generation Circuit

(1) The frequencies of the system clock (ICLK), peripheral module clocks (PCLKA, PCLKB, and PCLKD), and FlashIF clock (FCLK) supplied to each module can be selected by the SCKCR register. Each frequency should meet the following:
Select each frequency that is within the operation guaranteed range of clock cycle time (tcyc) specified in AC characteristics of electrical characteristics.
The frequencies must not exceed the ranges listed in Table 9.1.
The peripheral modules operate on the PCLKA, PCLKB and PCLKD. Note therefore that the operating speed of modules such as the timer and SCI varies before and after the frequency is changed.

(2) The relationship of frequencies of the system clock (ICLK), peripheral module clocks A, B, and D (PCLKA, PCLKB, and PCLKD), and FlashIF clock (FCLK) must be set as follows.
ICLK:FCLK = N:1 or 1:N (N is an integer)
ICLK:PCLKA, PCLKB, and PCLKD = N:1 or 1:N (N is an integer)

(3) To secure the processing after the clock frequency is changed, modify the pertinent clock control register to change the frequency, and then read the value from the register, and then perform the subsequent processing.

9.9.2 Note on Rewriting the SCKCR3 Register

When the SCKCR3.CKSEL[2:0] bits have been rewritten, the clock output is temporarily stopped to prevent the switch of the clock source from generating a clock pulse of short duration (glitch). The interrupt controller and event link controller may not detect the following signals that were input during this period.

(1) An external pin interrupt or NMI pin interrupt with a pulse width shorter than 4 cycles of the PCLKB after the switch while the PCLKB frequency is the 1/1 of the clock source (the SCKCR.PCKB[3:0] bits are 0000b).

(2) An external pin interrupt or NMI pin interrupt with a pulse width shorter than 2.5 cycles of the PCLKB after the switch while the PCLKB frequency is the 1/2 of the clock source (the SCKCR.PCKB[3:0] bits are 0001b).

(3) An RTC periodic interrupt or RTC periodic event output when the SCKCR3.CKSEL[2:0] bits are changed to “011b” (sub-clock).

When the external pin interrupt or NMI pin interrupt is in use, input the signal with enough pulse width to exceed the time condition described in (1) and (2). When the RTC periodic interrupt or RTC periodic event output is in use, switch the clock source after a periodic interrupt is generated and before the next periodic interrupt is generated.

9.9.3 Notes on Resonator

Since various resonator characteristics relate closely to the user's board design, adequate evaluation is required on the user side before use, referencing the resonator connection example shown in this section. The circuit constants for the resonator depend on the resonator to be used and the stray capacitance of the mounting circuit. Therefore, the circuit constants should be determined in full consultation with the resonator manufacturer. The voltage to be applied between the resonator pins must be within the absolute maximum rating.

9.9.4 Notes on Board Design

When using a crystal, place the resonator and its load capacitors as close to the XTAL and EXTAL pins as possible. Other signal lines should be routed away from the oscillation circuit as shown in Figure 9.14 to prevent electromagnetic induction from interfering with correct oscillation.

Downloaded from Arrow.com.

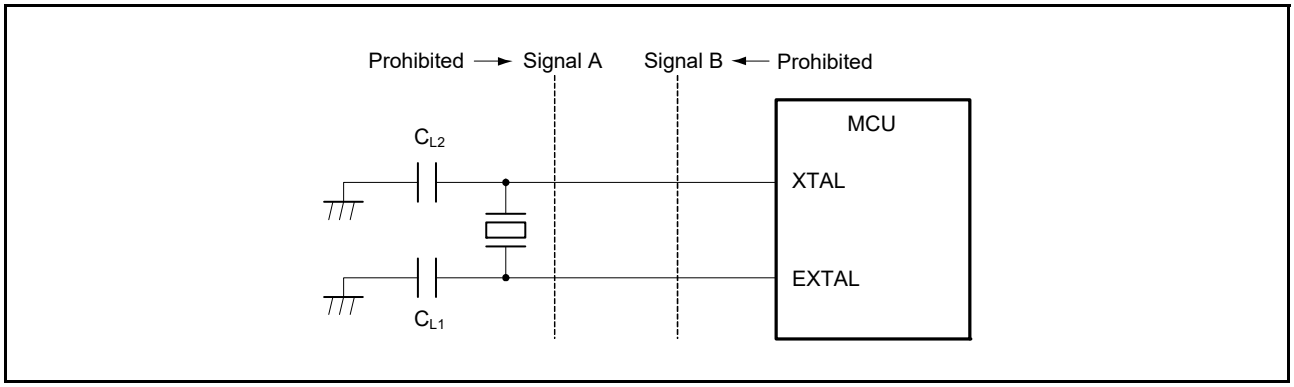


Figure 9.14 Point for Caution on Board Design for the Oscillation Circuit (Applicable to the Main Clock Oscillator as well as the Sub Clock Oscillator and Bluetooth-Dedicated Osillator)

Downloaded from Arrow.com.

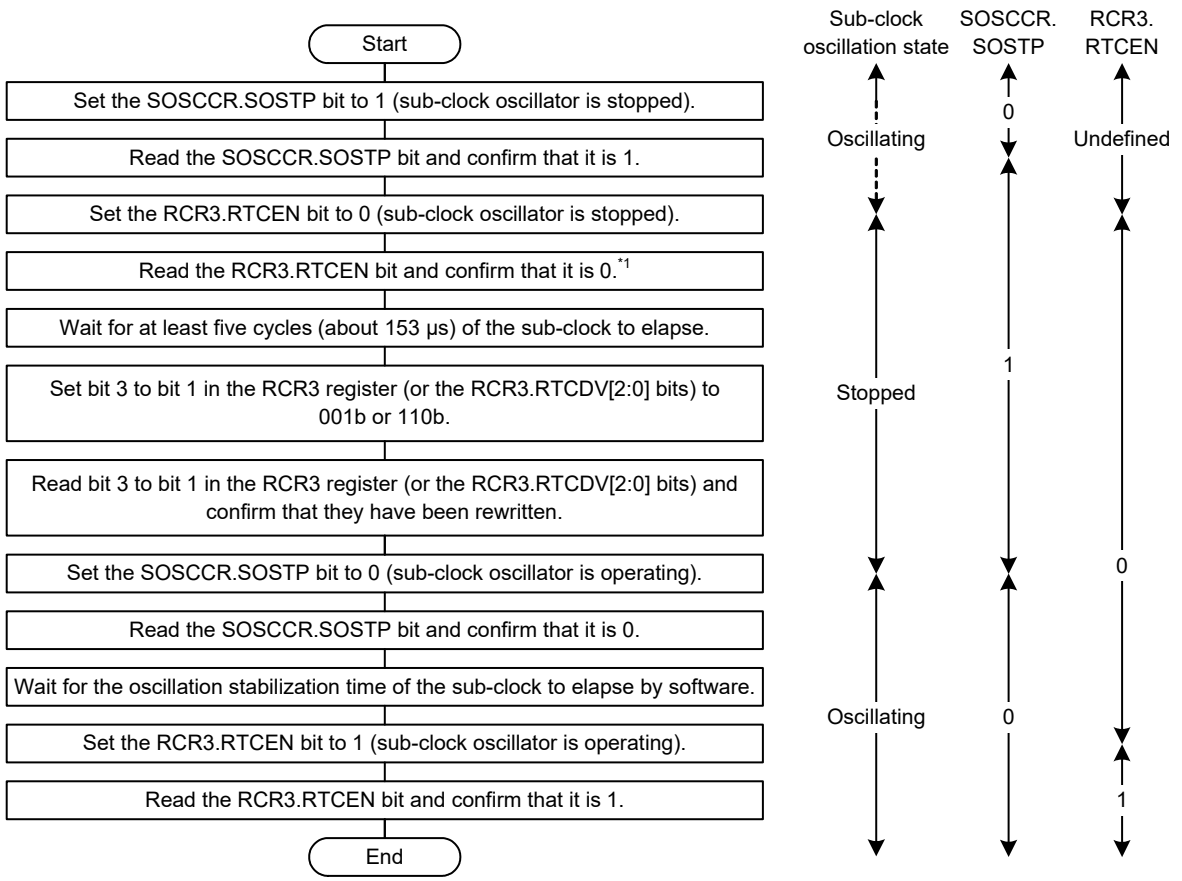
9.9.5 Notes on Resonator Connection Pins

When the main clock is not used, the EXTAL and XTAL pins can be used as general ports P36 and P37. When using these pins as general ports, be sure to stop the main clock (MOSCCR.MOSTP = 1). However, do not use the EXTAL and XTAL pins as general ports P36 and P37 in a system that uses the main clock.
When the main clock is used, do not set P36 and P37 to output.

9.9.6 Notes on Sub-Clock

The sub-clock can be used as the system clock, as the count source for the realtime clock, or as both. Take note of the following limitations and points for caution regarding the settings, including when the sub-clock is not in use.

- With regard to making the sub-clock oscillator run or stop, setting either the sub-clock oscillator stop bit in the sub-clock oscillator control register (SOSCCR.SOSTP) or the sub-clock oscillator control bit in RTC control register 3 (RCR3.RTCEN) will make the oscillator run.
- To use the sub-clock as the system clock and as the count source of the realtime clock simultaneously, perform initial settings according to the flowchart example shown in Figure 9.15. After that, perform the clock setting procedure shown in section 28.3.2, Clock and Count Mode Setting Procedure.

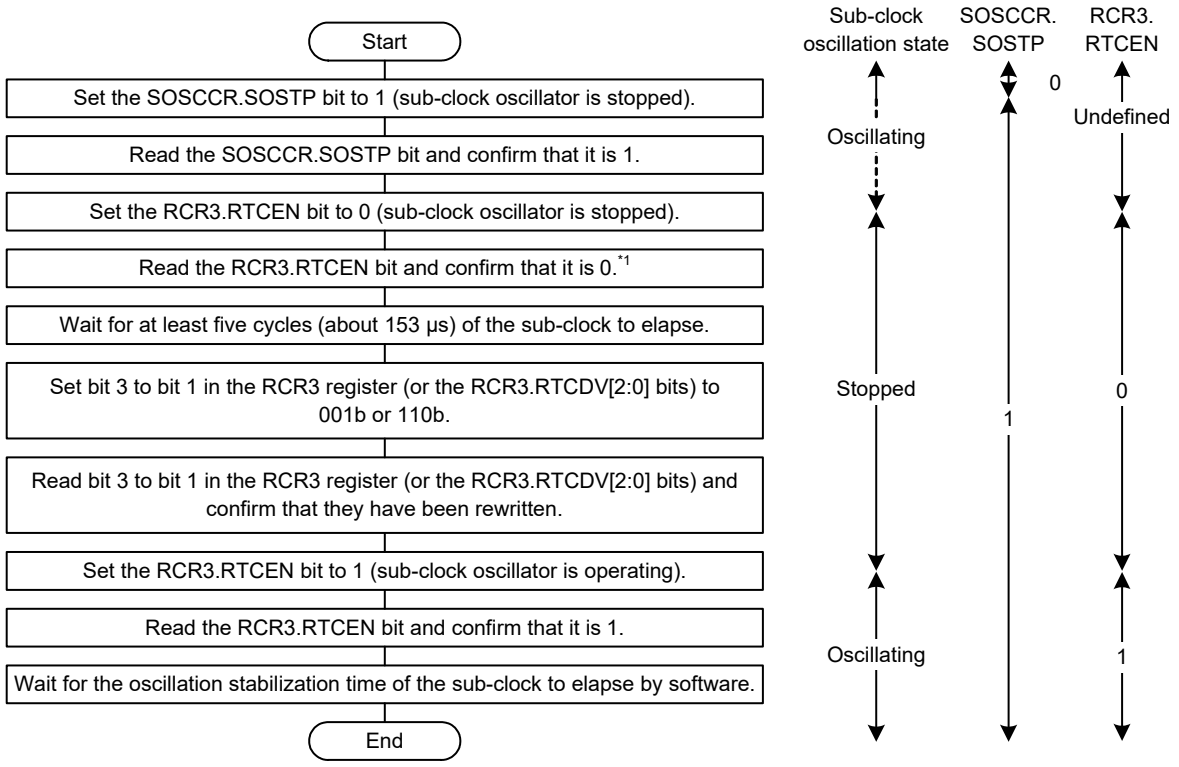


Note 1. The values of bit 7 to bit 1 are undefined. Confirm only the value of the RCR3.RTCEN bit.

Figure 9.15 Example of Initialization Flowchart When Sub-Clock is Used as Count Source of Realtime Clock

Downloaded from Arrow.com.

- When using the sub-clock only as the count source of the realtime clock, perform initial settings according to the flowchart example shown in Figure 9.16. After that, perform the clock setting procedure shown in section 28.3.2, Clock and Count Mode Setting Procedure.

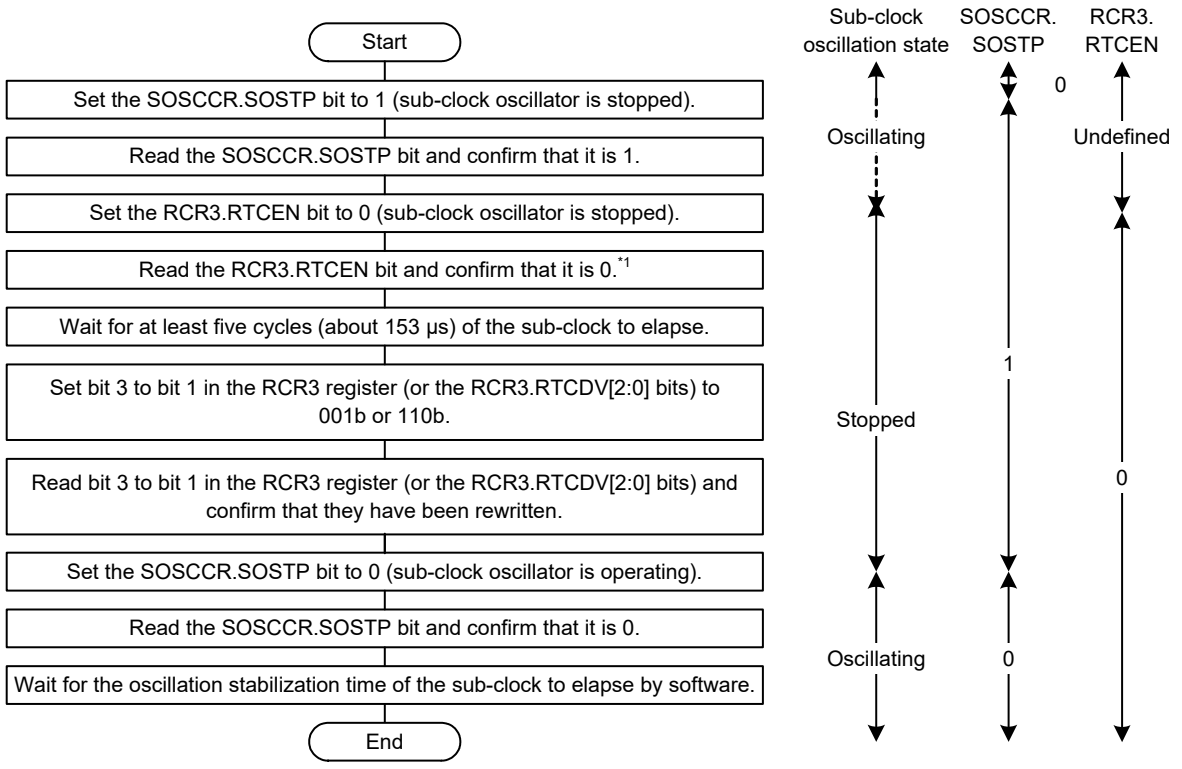


Note 1. The values of bit 7 to bit 1 are undefined. Confirm only the value of the RCR3.RTCEN bit.

Figure 9.16 Example of Initialization Flowchart When Sub-Clock is Used Only as Count Source of Realtime Clock

Downloaded from Arrow.com.

- When using the sub-clock only as the system clock, perform initial settings according to the flowchart example shown in Figure 9.17.

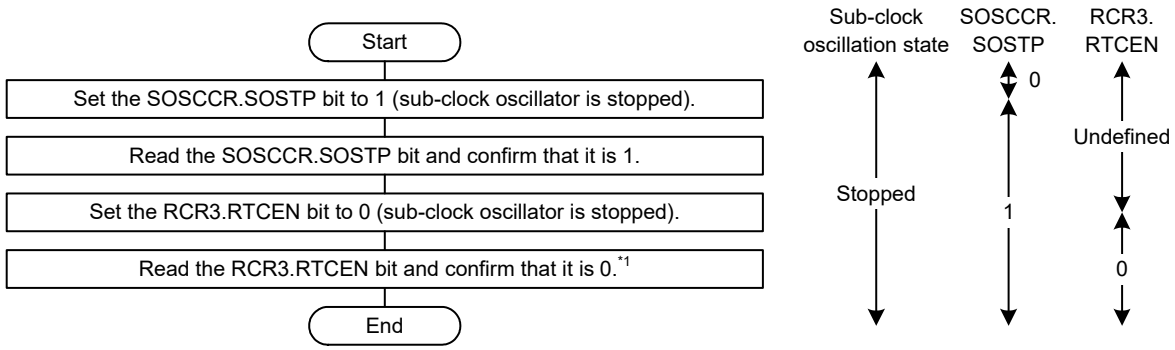


Note 1. The values of bit 7 to bit 1 are undefined. Confirm only the value of the RCR3.RTCEN bit.

Figure 9.17 Example of Initialization Flowchart When Sub-Clock is Used Only as System Clock

Downloaded from Arrow.com.

- When not using the sub-clock, perform initial settings according to the flowchart example in Figure 9.18.



Note 1. The values of bit 7 to bit 1 are undefined. Confirm only the value of the RCR3.RTCEN bit.

Figure 9.18 Example of Initialization Flowchart When Sub-Clock is Not Used

- Regardless of the RCR3.RTCEN bit setting, wait until the oscillator stabilization wait time elapses before rewriting the SOSCCR.SOSTP bit to 0 (sub-clock oscillator is operating).
- Since the sub-clock control circuit is in an unstable state after a cold start, it must be initialized regardless of whether or not the sub-clock is in use. The sub-clock is initialized by setting the SOSCCR.SOSTP bit to 1 and the RCR3.RTCEN bit to 0 (sub-clock oscillator is stopped). See section 28.2.19, RTC Control Register 3 (RCR3), for instructions to initialize the RCR3.RTCEN bit.
- The RCR3.RTCDV[2:0] bits must also be set when operating the sub-clock oscillator. Set these bits while the sub-clock oscillator is stopped. Do not rewrite these bits while the sub-clock oscillator is operating.
- When successively rewriting the SOSCCR.SOSTP bit followed by the RCR3.RTCEN bit or vice versa, confirm that the first bit rewrite was completed successfully before rewriting the second bit.

Downloaded from Arrow.com.

10. Clock Frequency Accuracy Measurement Circuit (CAC)

10.1 Overview

The clock frequency accuracy measurement circuit (CAC) counts pulses of the clock to be measured (measurement target clock) within the time generated by the clock to be used as a measurement reference (measurement reference clock), and determines the accuracy depending on whether the number of pulses is within the allowable range.
When measurement is completed or the number of pulses within the time generated by the measurement reference clock is not within the allowable range, an interrupt request is generated.
Table 10.1 lists the specifications of the CAC and Figure 10.1 shows a block diagram of the CAC.

Table 10.1 CAC Specifications

Item	Description
Measurement target clocks	The frequency of the following clocks can be measured. • Main clock • Sub-clock • HOCO clock • LOCO clock • IWDT-dedicated clock (IWDTCLK) • Peripheral module clock B (PCLKB)
Measurement reference clocks	• External clock input to the CACREF pin • Main clock • Sub-clock • HOCO clock • LOCO clock • IWDT-dedicated clock (IWDTCLK) • Peripheral module clock B (PCLKB)
Selectable function	Digital filter function
Interrupt sources	• Measurement end interrupt • Frequency error interrupt • Overflow interrupt
Low power consumption function	Module stop state can be set.

Downloaded from Arrow.com.

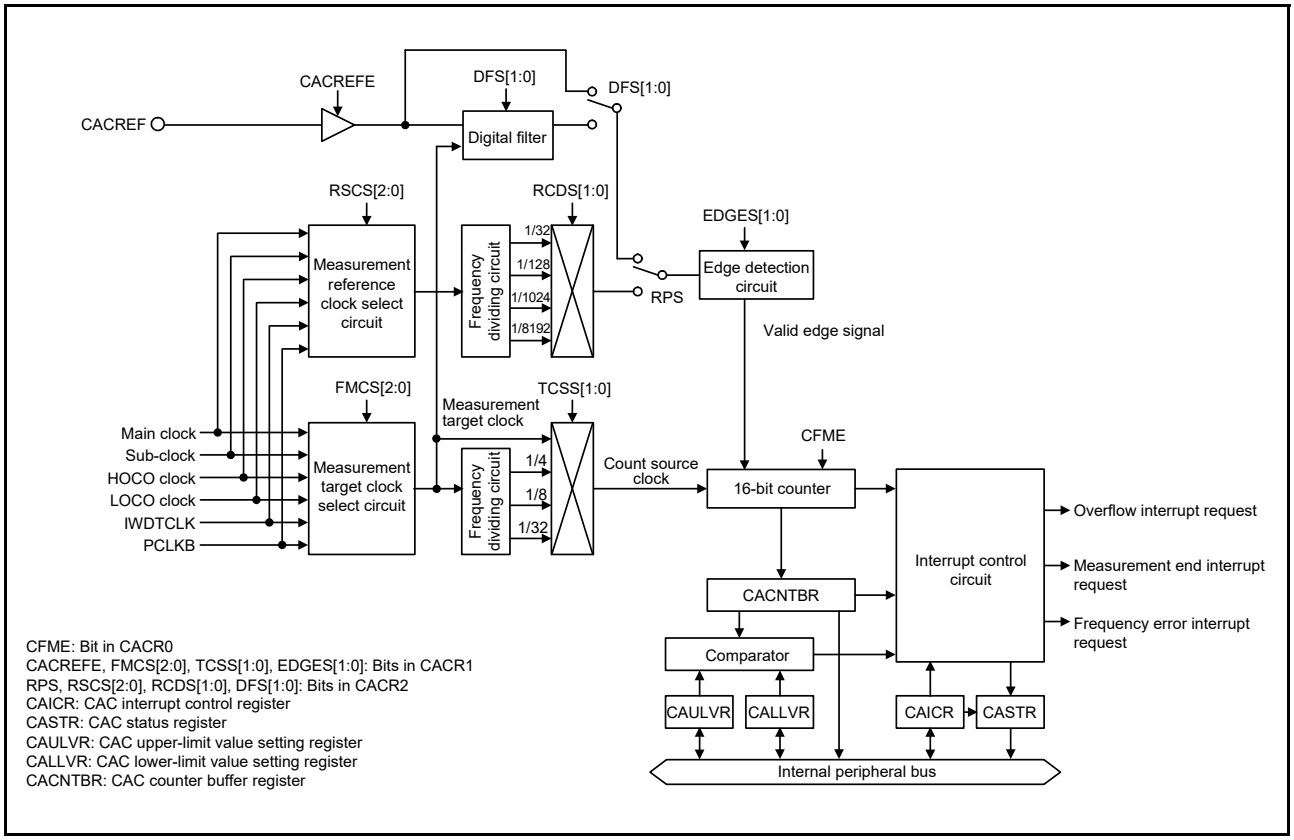


Figure 10.1 CAC Block Diagram

Table 10.2 shows the pin configuration of the CAC.

Table 10.2 Pin Configuration of CAC

Pin Name	I/O	Function
CACREF	Input	Measurement reference clock input pin

Downloaded from Arrow.com.

10.2 Register Descriptions

10.2.1 CAC Control Register 0 (CACR0)

Address(es): 0008 B000h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	CFME
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CFME	Clock Frequency Measurement Enable	0: Clock frequency measurement is disabled. 1: Clock frequency measurement is enabled.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

CFME Bit (Clock Frequency Measurement Enable)

This bit specifies whether clock frequency measurement is enabled or disabled.

When rewriting this bit, more time is required than other bits for the new value to be reflected in the register. Further write access to this bit are ignored until the current write access is reflected in the register. Read the bit to confirm that the rewrite has been reflected in the register.

Downloaded from Arrow.com.

10.2.2 CAC Control Register 1 (CACR1)

Address(es): 0008 B001h

	b7	b6	b5	b4	b3	b2	b1	b0
	EDGES[1:0]		TCSS[1:0]		FMCS[2:0]			CACREFE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CACREFE	CACREF Pin Input Enable	0: CACREF pin input is disabled. 1: CACREF pin input is enabled.	R/W
b3 to b1	FMCS[2:0]	Measurement Target Clock Select	b3 b1 0 0 0: Main clock 0 0 1: Sub-clock 0 1 0: HOCO clock 0 1 1: LOCO clock 1 0 0: IWDT-dedicated clock (IWDTCLK) 1 0 1: Peripheral module clock B (PCLKB) Settings other than above are prohibited.	R/W
b5, b4	TCSS[1:0]	Timer Count Clock Source Select	b5 b4 0 0: No division 0 1: ×1/4 clock 1 0: ×1/8 clock 1 1: ×1/32 clock	R/W
b7, b6	EDGES[1:0]	Valid Edge Select	b7 b6 0 0: Rising edge 0 1: Falling edge 1 0: Both rising and falling edges 1 1: Setting prohibited	R/W

Note: Set the CACR1 register when the CACR0.CFME bit is 0.

CACREFE Bit (CACREF Pin Input Enable)

This bit specifies whether the CACREF pin input is enabled or disabled.

FMCS[2:0]Bits (Measurement Target Clock Select)

These bits select the measurement target clock whose frequency is to be measured.

TCSS[1:0] Bits (Timer Count Clock Source Select)

These bits select the count clock source for the clock frequency accuracy measurement circuit.

EDGES[1:0]Bits (Valid Edge Select)

These bits select the valid edge for the reference signal.

Downloaded from Arrow.com.

10.2.3 CAC Control Register 2 (CACR2)

Address(es): 0008 B002h

	b7	b6	b5	b4	b3	b2	b1	b0
	DFS[1:0]		RCDS[1:0]		RSCS[2:0]			RPS
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RPS	Reference Signal Select	0: CACREF pin input 1: Internal clock (internally generated signal)	R/W
b3 to b1	RSCS[2:0]	Measurement Reference Clock Select	b3 b1 0 0 0: Main clock 0 0 1: Sub-clock 0 1 0: HOCO clock 0 1 1: LOCO clock 1 0 0: IWDT-dedicated clock (IWDTCLK) 1 0 1: Peripheral module clock B (PCLKB) Settings other than above are prohibited.	R/W
b5, b4	RCDS[1:0]	Measurement Reference Clock Frequency Division Ration Select	b5 b4 0 0: ×1/32 clock 0 1: ×1/128 clock 1 0: ×1/1024 clock 1 1: ×1/8192 clock	R/W
b7, b6	DFS[1:0]	Digital Filter Select	b7 b6 0 0: Digital filtering is disabled. 0 1: The sampling clock for the digital filter is the measurement target clock. 1 0: The sampling clock for the digital filter is the measurement target clock divided by 4. 1 1: The sampling clock for the digital filter is the measurement target clock divided by 16.	R/W

Note: Set the CACR2 register when the CACR0.CFME bit is 0.

RPS Bit (Reference Signal Select)

This bit selects whether to use the CACREF pin input or an internal clock (internally generated signal) as the reference signal.

RSCS[2:0]Bits (Measurement Reference Clock Select)

These bits select the clock source for generating the measurement reference clock.

RCDS[1:0]Bits (Measurement Reference Clock Frequency Division Ration Select)

These bits select the frequency division ratio of the measurement reference clock.

DFS[1:0]Bits (Digital Filter Select)

The setting of these bits enables or disables the digital filter and selects its sampling clock.

Downloaded from Arrow.com.

10.2.4 CAC Interrupt Request Enable Register (CAICR)

Address(es): 0008 B003h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	OVFFCL	MENDFCL	FERRFCL	—	OVFIE	MENDIE	FERRIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	FERRIE	Frequency Error Interrupt Request Enable	0: Frequency error interrupt request is disabled. 1: Frequency error interrupt request is enabled.	R/W
b1	MENDIE	Measurement End Interrupt Request Enable	0: Measurement end interrupt request is disabled. 1: Measurement end interrupt request is enabled.	R/W
b2	OVFIE	Overflow Interrupt Request Enable	0: Overflow interrupt request is disabled. 1: Overflow interrupt request is enabled.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	FERRFCL	FERRF Clear	When 1 is written to this bit, the CASTR.FERRF flag is cleared. This bit is read as 0.	R/W
b5	MENDFCL	MENDF Clear	When 1 is written to this bit, the CASTR.MENDF flag is cleared. This bit is read as 0.	R/W
b6	OVFFCL	OVFF Clear	When 1 is written to this bit, the CASTR.OVFF flag is cleared. This bit is read as 0.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

FERRIE Bit (Frequency Error Interrupt Request Enable)

This bit specifies whether the frequency error interrupt request is enabled or disabled.

MENDIE Bit (Measurement End Interrupt Request Enable)

This bit specifies whether the measurement end interrupt request is enabled or disabled.

OVFIE Bit (Overflow Interrupt Request Enable)

This bit specifies whether the overflow interrupt request is enabled or disabled.

FERRFCL Bit (FERRF Clear)

Setting this bit to 1 clears the CASTR.FERRF flag.

MENDFCL Bit (MENDF Clear)

Setting this bit to 1 clears the CASTR.MENDF flag.

OVFFCL Bit (OVFF Clear)

Setting this bit to 1 clears the CASTR.OVFF flag.

Downloaded from Arrow.com.

10.2.5 CAC Status Register (CASTR)

Address(es): 0008 B004h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	OVFF	MENDF	FERRF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	FERRF	Frequency Error Flag	0: The clock frequency is within the range corresponding to the settings. 1: The clock frequency has deviated beyond the range corresponding to the settings (frequency error).	R
b1	MENDF	Measurement End Flag	0: Measurement is in progress. 1: Measurement has ended.	R
b2	OVFF	Overflow Flag	0: The counter has not overflowed. 1: The counter has overflowed.	R
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

FERRF Flag (Frequency Error Flag)

This flag indicates deviation of the clock frequency from the set value (frequency error).

[Setting condition]

- The clock frequency is outside of the setting range.

[Clearing condition]

- 1 is written to the CAICR.FERRFCL bit.

MENDF Flag (Measurement End Flag)

This flag indicates the end of measurement.

[Setting condition]

- Measurement has finished.

[Clearing condition]

- 1 is written to the CAICR.MENDFCL bit.

OVFF Flag (Overflow Flag)

This flag indicates that the counter has overflowed.

[Setting condition]

- The counter has overflowed.

[Clearing condition]

- 1 is written to the CAICR.OVFFCL bit.

Downloaded from Arrow.com.

10.2.6 CAC Upper-Limit Value Setting Register (CAULVR)

Address(es): 0008 B006h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CAULVR is a 16-bit readable/writable register that specifies the upper-limit value of the counter used for measuring the frequency. When the frequency rises above the value specified in this register, a frequency error is detected.
Write to this register when the CACR0.CFME bit is 0.
The counter value held in CACNTBR can vary with the difference between the phases of the digital filter and edge-detection circuit on the one hand and the signal on the CACREF pin on the other, so ensure that this setting allows an adequate margin.

10.2.7 CAC Lower-Limit Value Setting Register (CALLVR)

Address(es): 0008 B008h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CALLVR is a 16-bit readable/writable register that specifies the lower-limit value of the counter used for measuring the frequency. When the frequency falls below the value specified in this register, a frequency error is detected.
Write to this register when the CACR0.CFME bit is 0.
The counter value held in CACNTBR can vary with the difference between the phases of the digital filter and edge-detection circuit on the one hand and the signal on the CACREF pin on the other, so ensure that this setting allows an adequate margin.

10.2.8 CAC Counter Buffer Register (CACNTBR)

Address(es): 0008 B00Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CACNTBR is a 16-bit read-only register that retains the counter value at the time a valid reference signal edge is input.

Downloaded from Arrow.com.

10.3 Operation

10.3.1 Measuring Clock Frequency

The clock frequency accuracy measurement circuit measures the clock frequency using the CACREF pin input or the internal clock as a reference. Figure 10.2 shows an operating example of the clock frequency accuracy measurement circuit.

The clock frequency accuracy measurement circuit operates as shown below when measuring the clock frequency.

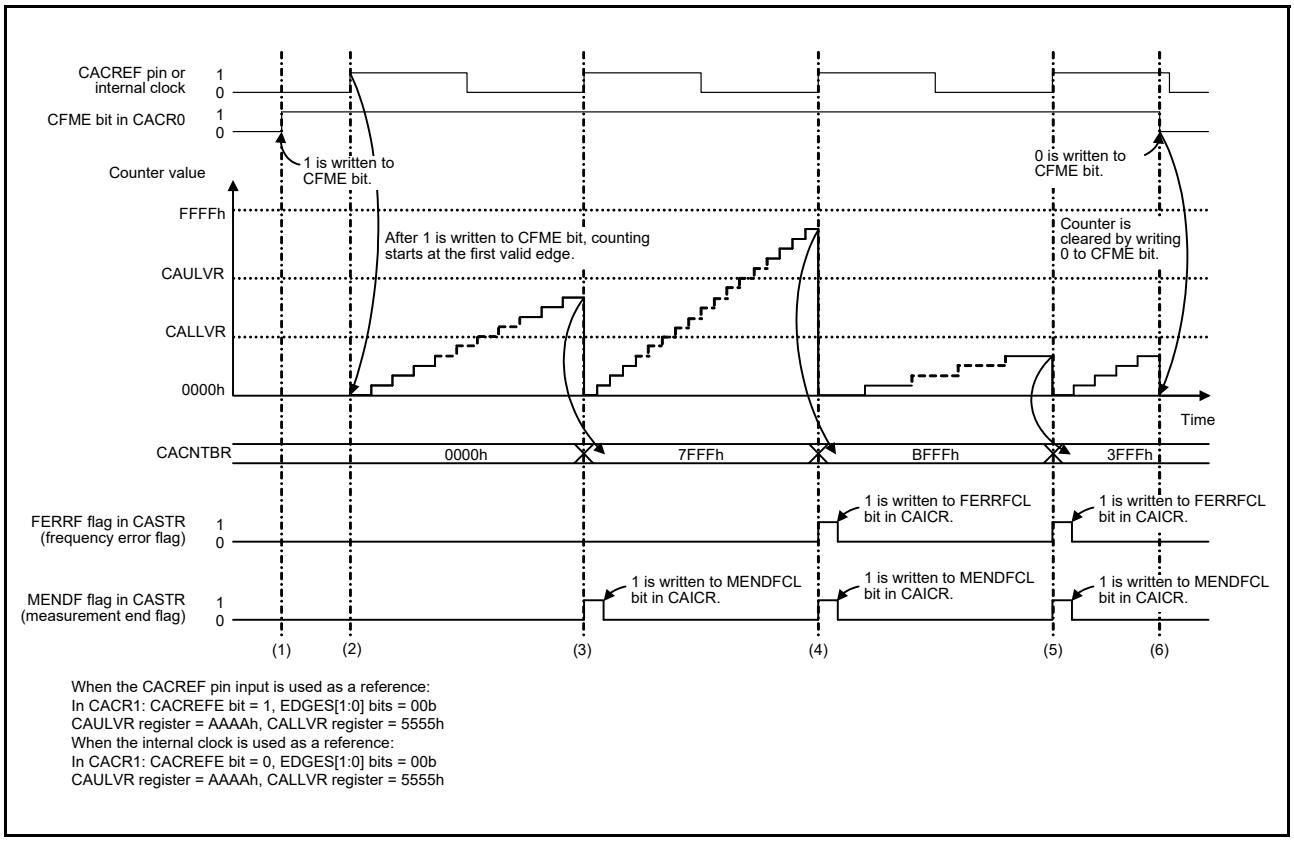


Figure 10.2 Operating Example of Clock Frequency Accuracy Measurement Circuit

(1) When the CACREF pin input is used as a reference (CACR1.CACREFE bit = 1), clock frequency measurement is enabled by writing 1 to the CACR0.CFME bit while the CACR2.RPS bit is 0 and the CACR1.CACREFE bit is 1. On the other hand, when the internal clock is used as a reference (CACR1.CACREFE bit = 0), clock frequency measurement is enabled by writing 1 to the CACR0.CFME bit while the CACR2.RPS bit is 1.

(2) When the CACREF pin input is used as a reference, the timer starts up-counting if the valid edge selected by the CACR1.EDGES[1:0] bits is input from the CACREF pin after 1 is written to the CFME bit. The valid edge is a rising edge (CACR1.EDGES[1:0] = 00b) in Figure 10.2.
When the internal clock is used as a reference, the timer starts up-counting if the valid edge selected by the CACR1.EDGES[1:0] bits is input based on the clock source selected by the CACR2.RSCS[2:0] bits after 1 is written to the CFME bit. The valid edge is a rising edge (CACR1.EDGES[1:0] = 00b) in Figure 10.2.

(3) When the next valid edge is input, the counter value is transferred in CACNTBR and compared with the values of CAULVR and CALLVR. If both CACNTBR ≤ CAULVR and CACNTBR ≥ CALLVR are satisfied, only the MENDF flag in CASTR is set to 1 because the clock frequency is correct. If the MENDIE bit in CAICR is 1, a measurement end interrupt is generated.

(4) When the next valid edge is input, the counter value is transferred in CACNTBR and compared with the values of CAULVR and CALLVR. In the case of CACNTBR > CAULVR, the FERRF flag in CASTR is set to 1 because the

Downloaded from Arrow.com.

clock frequency is erroneous. If the FERRIE bit in CAICR is 1, a frequency error interrupt is generated. Also, the MENDF flag in CASTR is set to 1. If the MENDIE bit in CAICR is 1, a measurement end interrupt is generated.

(5) When the next valid edge is input, the counter value is transferred in CACNTBR and compared with the values of CAULVR and CALLVR. In the case of CACNTBR < CALLVR, the FERRF flag in CASTR is set to 1 because the clock frequency is erroneous. If the FERRIE bit in CAICR is 1, a frequency error interrupt is generated. Also, the MENDF flag in CASTR is set to 1. If the MENDIE bit in CAICR is 1, a measurement end interrupt is generated.

(6) While the CFME bit in CACR0 is 1, the counter value is transferred in CACNTBR and compared with the values of CAULVR and CALLVR every time a valid edge is input. Writing 0 to the CFME bit in CACR0 clears the counter and stops up-counting.

10.3.2 Digital Filtering of Signals on the CACREF Pin

The CACREF pin has a digital filter. Levels on the target pin for sampling are conveyed to the internal circuitry after matching three consecutive times at the selected sampling interval and the same level continues to be conveyed internally until the level on the pin again matches three consecutive times.

Enabling and disabling of the digital filter and its sampling clock are selectable.

The counter value transferred in CACNTBR may be in error by up to one cycle of the sampling clock due to the difference between the phases of the digital filter and the signal input to the CACREF pin.

When a frequency dividing clock is selected as a count source clock, the counter value error is obtained by the following formula:

Counter value error = (One cycle of the count source clock) / (One cycle of the sampling clock)

10.4 Interrupt Requests

The CAC generates three types of interrupt request: frequency error interrupt, measurement end interrupt, and overflow interrupt. When an interrupt source is generated, the corresponding status flag becomes 1. Table 10.3 lists details on the interrupt requests of the clock frequency accuracy measurement circuit.

Table 10.3 Interrupt Requests of Clock Frequency Accuracy Measurement Circuit

Interrupt Request	Interrupt Enable Bit	Status Flag	Interrupt Source
Frequency error interrupt	CAICR.FERRIE	CASTR.FERRF	The result of comparing CACNTBR to CAULVR and CALLVR is either CACNTBR > CAULVR or CACNTBR < CALLVR.
Measurement end interrupt	CAICR.MENDIE	CASTR.MENDF	A valid edge is input from the CACREF pin. Note however that a measurement end interrupt does not occur at the first valid edge after writing 1 to the CACR0.CFME bit.
Overflow interrupt	CAICR.OVFIE	CASTR.OVFF	The counter has overflowed.

Downloaded from Arrow.com.

10.5 Usage Notes

10.5.1 Module Stop Function Setting

CAC operation can be disabled or enabled using module stop control register C (MSTPCRC). The initial setting is for the CAC to be halted. Register access is enabled by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

Downloaded from Arrow.com.

11. Low Power Consumption

11.1 Overview

This MCU has several functions for reducing power consumption, by setting clock dividers, stopping modules, changing to low power consumption mode in normal operation, and changing to operating power control mode.
Table 11.1 lists the specifications of low power consumption functions, and Table 11.2 lists the conditions to change to low power consumption modes, states of the CPU and peripheral modules, and the method for exiting each mode.
After a reset, this MCU returns to normal mode, but modules except the DMAC, DTC, and RAM are stopped.

Table 11.1 Specifications of Low Power Consumption Functions

Item	Specification
Clock divider functions	The frequency division ratio can be set independently for the system clock (ICLK), high speed peripheral module clock (PCLKA), peripheral module clock (PCLKB), S12AD clock (PCLKD), and FlashIF clock (FCLK).*1
Module stop function	Each peripheral module can be stopped independently by the module stop control register.
Function for transition to low power consumption mode	Transition to a low power consumption mode in which the CPU, peripheral modules, or oscillators are stopped is enabled.
Low power consumption modes	• Sleep mode • Deep sleep mode • Software standby mode
Operating power control modes	• Power consumption can be reduced in normal operation, sleep mode, and deep sleep mode by selecting an appropriate operating power control mode according to the operating frequency and operating voltage. • Three operating power control modes are available High-speed operating mode Middle-speed operating mode Low-speed operating mode

Note 1. For details, refer to section 9, Clock Generation Circuit.

Downloaded from Arrow.com.

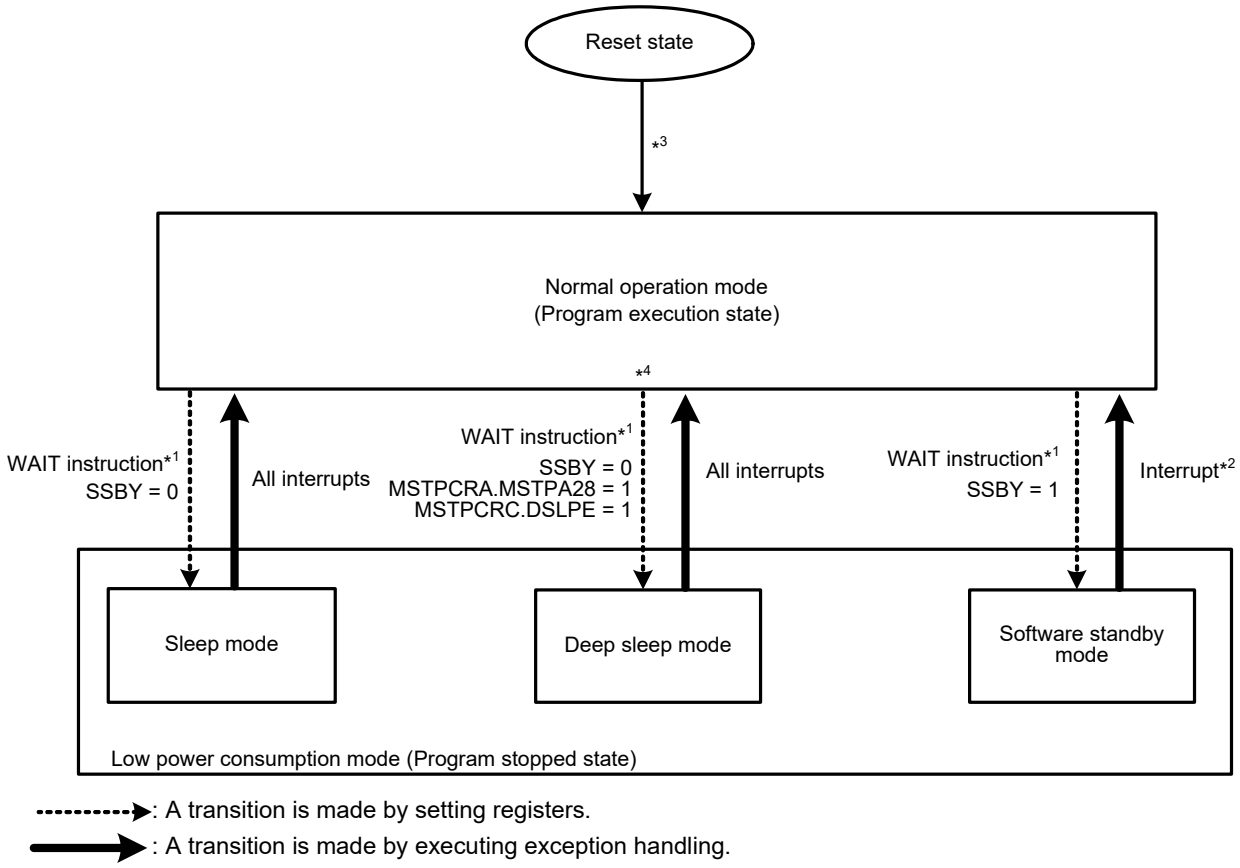
Table 11.2 Operating Conditions of Each Power Consumption Mode

Entering and Exiting Low Power Consumption Modes and Operating States	Sleep Mode	Deep Sleep Mode	Software Standby Mode
Entry trigger	Control register + instruction	Control register + instruction	Control register + instruction
Exit trigger	Interrupt	Interrupt	Interrupt*1
After exiting from each mode, CPU begins from*2	Interrupt handling	Interrupt handling	Interrupt handling
Main clock oscillator	Operating possible	Operating possible	Stopped
Sub-clock oscillator	Operating possible	Operating possible	Operating possible
High-speed on-chip oscillator	Operating possible	Operating possible	Stopped
Low-speed on-chip oscillator	Operating possible	Operating possible	Stopped
IWDT-dedicated on-chip oscillator	Operating possible*3	Operating possible*3	Operating possible*3
PLL	Operating possible	Operating possible	Stopped
USB-dedicated PLL	Operating possible	Operating possible	Stopped
CPU	Stopped (Retained)	Stopped (Retained)	Stopped (Retained)
RAM (0000 0000h to 0000 FFFFh)	Operating possible (Retained)	Stopped (Retained)	Stopped (Retained)
DMAC	Operating possible*5	Stopped (Retained)	Stopped (Retained)
DTC	Operating possible*5	Stopped (Retained)	Stopped (Retained)
Flash memory	Operating	Stopped (Retained)	Stopped (Retained)
Watchdog timer (WDT)	Stopped (Retained)	Stopped (Retained)	Stopped (Retained)
Independent watchdog timer (IWDT)	Operating possible*3	Operating possible*3	Operating possible*3
Realtime clock (RTC)	Operating possible	Operating possible	Operating possible
Low power timer (LPT)	Operating possible*8	Operating possible*8	Operating possible*8
Voltage detection circuit (LVD)	Operating possible	Operating possible	Operating possible
Power-on reset circuit	Operating	Operating	Operating
Peripheral modules	Operating possible	Operating possible	Stopped (Retained)*4
I/O ports	Operating	Operating	Retained
RTCOUT	Operating possible	Operating possible	Operating possible
CLKOUT	Operating possible	Operating possible	Operating possible*6
Comparator B	Operating possible	Operating possible	Operating possible*7

“Operating possible” means that operating or stopped can be controlled by the register setting.
“Stopped (Retained)” means that internal register values are retained and internal operations are suspended.

Note 1. “Interrupts” here indicates an external pin interrupt (the NMI, IRQ0, IRQ1, or IRQ4 to IRQ7) or any of peripheral interrupts (the RTC alarm, RTC interval, IWDT, voltage monitoring, VBATT pin voltage drop detection, USB and ELC (LPT-dedicated interrupt) interrupts).
Note 2. This does not include a RES# pin reset, power-on reset, voltage monitoring reset, or independent watchdog-timer reset. One of these reset sources initiate transition to reset state.
Note 3. Operating or stopping is selected by setting the IWDT sleep mode count stop control bit (IWDTSLCSTP) in option function select register 0 (OFS0) in IWDT auto-start mode. In any mode other than IWDT auto-start mode, operating or stopping is selected by the setting of the sleep mode count stop control bit (SLCSTP) in the IWDT count stop control register (IWDTCSTPR).
Note 4. The peripheral logic states are retained.
Note 5. During sleep mode, do not write to the system control related registers (indicated by ‘SYSTEM’ in the Module Symbol column in Table 5.1, List of I/O Registers (Address Order).
Note 6. Stopped when the clock output source select bits (CKOCR.CKOSEL[2:0]) are set to a value other than 011b (sub-clock oscillator).
Note 7. Using the digital filter function is prohibited. Operation for outputting the comparison result to the CMPOBn pin is possible.
Note 8. When the low-power timer clock source select bit in the low-power timer control register 1 (LPTCR1.LPCNTCKSEL) is 1 (selecting IWDT-dedicated on-chip oscillator), operating or stopping can be selected as described in Note 3. When the value of the bit is 0 (selecting the sub-clock oscillator), only operating is possible.

Downloaded from Arrow.com.



Note 1. If an interrupt is accepted while entering low power consumption mode, the transition to the program stop state is exited and the interrupt exception handling is executed.
Note 2. “Interrupts” here indicates an external pin interrupt (the NMI, IRQ0, IRQ1, or IRQ4 to IRQ7) or any of peripheral interrupts (the RTC alarm, RTC interval, IWDT, voltage monitoring, VBATT pin voltage drop detection interrupts USB and ELC (LPT-dedicated interrupt) interrupts).
Note 3. The LOCO is the clock source following a transition from the reset state to normal mode.
Note 4. Makes a transition from sleep mode, deep sleep mode, or software standby mode to normal operating mode by an interrupt. In the case of exiting sleep mode, the clock source after exiting is selectable. For details, refer to the description of the RSTCKCR register.
For deep sleep mode and software standby mode, the clock source after exiting is the same as that of the clock before entering software standby mode.

Transition to the reset state is made at any state instantly after a RES# pin reset, power on reset, voltage monitoring reset, WDT reset, IWDT reset, or software reset is generated.

Figure 11.1 Mode Transitions

Downloaded from Arrow.com.

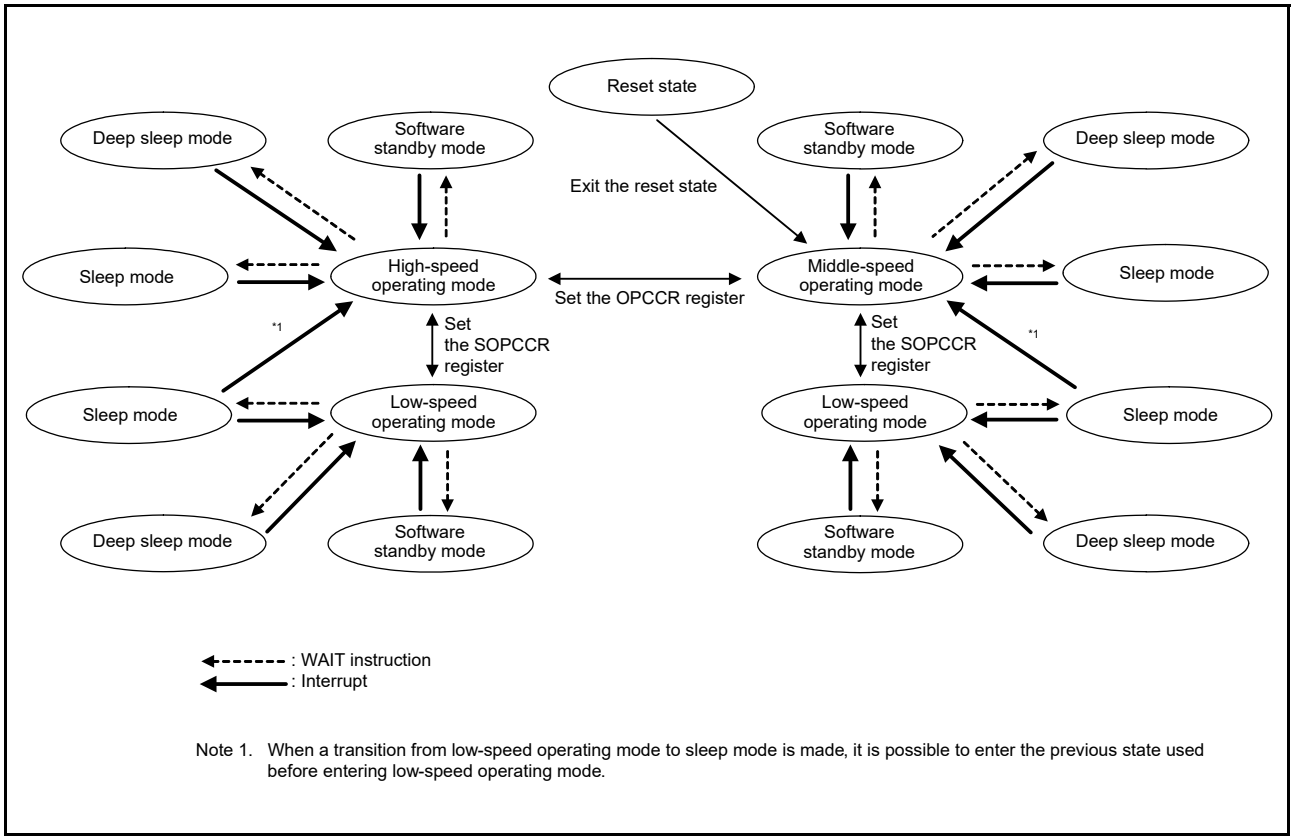


Figure 11.2 Operating Modes

- The sub-clock oscillator does not stop when entering software standby mode.
- It is possible to return from sleep mode to the previous operating state used before entering sleep mode. However, when a transition from low-speed operating mode to sleep mode is made, it is possible to enter the previous state used before entering low-speed operating mode.
- After exiting the reset state, operation starts in middle-speed operating mode.

Table 11.3 Oscillator Usability in Each Mode

	PLL	USB-dedicated PLL	HOCO	LOCO	IWDTCLK	Main Clock Oscillator	Sub-Clock Oscillator
High-speed operating mode	Usable*1	Usable*1	Usable	Usable	Usable	Usable	Usable
Middle-speed operating mode	Usable*1	Usable*1	Usable	Usable	Usable	Usable	Usable
Low-speed operating mode	Not usable	Not usable	Not usable	Not usable	Usable	Not usable	Usable

Note 1. The PLL and USB-dedicated PLL are usable when the power supply voltage is 2.4 V or above.

Downloaded from Arrow.com.

11.2 Register Descriptions

11.2.1 Standby Control Register (SBYCR)

Address(es): 0008 000Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SSBY	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b14 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	SSBY	Software Standby	0: Set entry to sleep mode or deep sleep mode after the WAIT instruction is executed 1: Set entry to software standby mode after the WAIT instruction is executed	R/W

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

SSBY Bit (Software Standby)

The SSBY bit specifies the transition destination after the WAIT instruction is executed.
When the SSBY bit is set to 1, the MCU enters software standby mode after execution of the WAIT instruction. When the MCU returns to normal mode after an interrupt has triggered and exits from software standby mode, the SSBY bit remains 1. The SSBY bit can be cleared by writing 0 to the SSBY bit.

When the oscillation stop detection function enable bit (OSTDCR.OSTDE) in the oscillation stop detection control register is 1, the set value of the SSBY bit is invalid. Even if the SSBY bit is 1, the MCU will enter sleep mode or deep sleep mode after execution of the WAIT instruction.

Downloaded from Arrow.com.

11.2.2 Module Stop Control Register A (MSTPCRA)

Address(es): 0008 0010h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	MSTPA28	—	—	—	—	—	—	—	—	MSTPA19	—	MSTPA17	—
Value after reset:	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MSTPA15	MSTPA14	MSTPA13	—	—	—	MSTPA9	—	—	—	MSTPA5	MSTPA4	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b4	MSTPA4	8-bit Timer 3 and 2 (Unit 1) Module Stop	Target module: TMR3, TMR2 0: This module clock is enabled 1: This module clock is disabled	R/W
b5	MSTPA5	8-bit Timer 1 and 0 (Unit 0) Module Stop	Target module: TMR1, TMR0 0: This module clock is enabled 1: This module clock is disabled	R/W
b8 to b6	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b9	MSTPA9	Multifunction Timer Pulse Unit 2 Module Stop	Target module: MTU (MTU0 to MTU4) 0: This module clock is enabled 1: This module clock is disabled	R/W
b12 to b10	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b13	MSTPA13	16-Bit Timer Pulse Unit Module Stop	Target module: TPU (TPU0 to TPU5) 0: The module stop state is canceled 1: Transition to the module stop state is made	R/W
b14	MSTPA14	Compare Match Timer 1 (Unit 1) Module Stop	Target module: CMT unit 1 (CMT2, CMT3) 0: This module clock is enabled 1: This module clock is disabled	R/W
b15	MSTPA15	Compare Match Timer (Unit 0) Module Stop	Target module: CMT unit 0 (CMT0, CMT1) 0: This module clock is enabled 1: This module clock is disabled	R/W
b16	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b17	MSTPA17	12-Bit A/D Converter Module Stop	Target module: S12AD 0: This module clock is enabled 1: This module clock is disabled	R/W
b18	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b19	MSTPA19	12-Bit D/A Converter Module Stop	Target module: DA 0: This module clock is enabled 1: This module clock is disabled	R/W
b27 to b20	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b28	MSTPA28	DMA Controller / Data Transfer Controller Module Stop	Target module: DMAC/DTC 0: This module clock is enabled 1: This module clock is disabled	R/W
b31 to b29	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

Downloaded from Arrow.com.

11.2.3 Module Stop Control Register B (MSTPCRB)

Address(es): 0008 0014h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	MSTPB30	—	—	—	MSTPB26	MSTPB25	—	MSTPB23	—	MSTPB21	—	MSTPB19	—	MSTPB17	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	MSTPB10	MSTPB9	—	—	MSTPB6	—	MSTPB4	—	—	—	MSTPB0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b0	MSTPB0*1	RSCAN0 Module Stop	Target module: RSCAN0 0:This module clock is enabled 1: This module clock is disabled	R/W
b3 to b1	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b4	MSTPB4	Serial Communication Interface SCIh Module Stop	Target module: SCIh (SCI12) 0: This module clock is enabled 1: This module clock is disabled	R/W
b5	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b6	MSTPB6	DOC Module Stop	Target module: DOC 0: This module clock is enabled 1: This module clock is disabled	R/W
b8, b7	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b9	MSTPB9	ELC Module Stop	Target module: ELC 0: This module clock is enabled 1: This module clock is disabled	R/W
b10	MSTPB10	Comparator B Module Stop	Target module: Comparator B 0: This module clock is enabled 1: This module clock is disabled	R/W
b16 to b11	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b17	MSTPB17	Serial Peripheral Interface 0 Module Stop	Target module: RSPI0 0: This module clock is enabled 1: This module clock is disabled	R/W
b18	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b19	MSTPB19 *2	USB0 Module Stop	Target module: USB0 0: This module clock is enabled 1: This module clock is disabled	R/W
b20	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b21	MSTPB21	I^2C Bus Interface 0 Module Stop	Target module: RIIC0 0: This module clock is enabled 1: This module clock is disabled	R/W
b22	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b23	MSTPB23	CRC Calculator Module Stop	Target module: CRC 0: This module clock is enabled 1: This module clock is disabled	R/W
b24	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b25	MSTPB25	Bluetooth Low Energy Module Stop	Target module: BLE 0: This module clock is enabled 1: This module clock is disabled	R/W
b26	MSTPB26	Serial Communication Interface 5 Module Stop	Target module: SCI5 0: This module clock is enabled 1: This module clock is disabled	R/W

Downloaded from Arrow.com.

Bit	Symbol	Bit Name	Description	R/W
b29 to b27	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b30	MSTPB30	Serial Communication Interface 1 Module Stop	Target module: SCI1 0: This module clock is enabled 1: This module clock is disabled	R/W
b31	—	Reserved	This bit is read as 1. The write value should be 1.	R/W

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

Note 1. This bit should be rewritten while the oscillation of the clock to be controlled by this bit is stable. When entering software standby mode after rewriting this bit, wait for two cycles of the CANMCLK after rewriting, and execute a WAIT instruction.

Note 2. This bit should be rewritten while the oscillation of the clock to be controlled by this bit is stable. The clock should be set for oscillation when this bit is 0 (this module clock is enabled). When entering software standby mode after rewriting this bit, wait for two cycles of the UCLK after rewriting, and execute a WAIT instruction. When stopping the clock after rewriting this bit to 1 (this module clock is disabled), wait for two cycles of the UCLK after rewriting, and stop the clock.

Downloaded from Arrow.com.

11.2.4 Module Stop Control Register C (MSTPCRC)

Address(es): 0008 0018h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DSLPE	—	—	—	MSTPC27	—	—	—	—	—	—	MSTPC20	MSTPC19	—	—	—
Value after reset:	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MSTPC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MSTPC0	RAM Module Stop*1	Target module: RAM (0000 0000h to 0000 FFFFh) 0: RAM operating 1: RAM stopped	R/W
b15 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b18 to b16	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b19	MSTPC19	Clock Frequency Accuracy Measurement Circuit Module Stop*2	Target module: CAC 0: This module clock is enabled 1: This module clock is disabled	R/W
b20	MSTPC20	IrDA Module Stop	Target module: IRDA 0: This module clock is enabled 1: This module clock is disabled	R/W
b26 to b21	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b27	MSTPC27	Serial Communication Interface 8 Module Stop	Target module: SCI8 0: This module clock is enabled 1: This module clock is disabled	R/W
b30 to b28	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b31	DSLPE	Deep Sleep Mode Enable	0: Deep sleep mode is disabled 1: Deep sleep mode is enabled	R/W

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.
Note 1. The MSTPC0 bit should not be set to 1 during access to the RAM. The RAM should not be accessed while the MSTPC0 bit is 1.
Note 2. The MSTPC19 bit should be rewritten while the oscillation of the clock to be controlled by this bit is stable. For entering software standby mode after rewriting this bit, wait for two cycles of the slowest clock among the clocks output by the oscillators actually oscillating and execute the WAIT instruction.

DSLPE Bit (Deep Sleep Mode Enable)

The DSLPE bit enables or disables a transition to deep sleep mode.

When the CPU executes the WAIT instruction with the DSLPE bit set to 1 and the SBYCR.SSBY and MSTPCRA.MSTPA28 bits meet specified conditions, the MCU enters deep sleep mode. For details, refer to section 11.6.2, Deep Sleep Mode.

Downloaded from Arrow.com.

11.2.5 Module Stop Control Register D (MSTPCRD)

Address(es): 0008 001Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	MSTPD31	—	—	—	—	—	—	—	—	—	—	—	MSTPD19	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MSTPD15	—	—	—	—	MSTPD10	—	—	—	—	—	—	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b9, b8	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b10	MSTPD10	Touch Sensor Control Unit Module Stop	Target module: CTSU 0: This module clock is enabled 1: This module clock is disabled	R/W
b14 to b11	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b15	MSTPD15*1	Serial Sound Interface Module Stop	Target module: SSI 0: This module clock is enabled 1: This module clock is disabled	R/W
b18 to b16	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b19	MSTPD19	SD Host Interface (SDHI) Module Stop	Target module: SDHI 0: This module clock is enabled 1: This module clock is disabled	R/W
b30 to b20	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b31	MSTPD31	Trusted Secure IP Function Module Stop*2, *3	Target module: Trusted Secure IP 0: This module clock is enabled 1: This module clock is disabled	R/W

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.
Note 1. When the main clock oscillator is used by the serial sound interface
This bit should be rewritten when the oscillation of the main clock oscillator is stable. When entering software standby mode after rewriting this bit, wait for two cycles of the main clock after rewriting, and execute a WAIT instruction. When stopping the main clock oscillator after rewriting this bit, wait for two cycles of the main clock after rewriting, and stop the main clock oscillator.
Note 2. This bit is reserved in chip version A. Set this bit once to 0 at the beginning of the program to initialize unused circuits.
Note 3. Set this bit once to 0 at the beginning of the program to initialize unused circuits even if the Trusted Secure IP function is not used in chip version B.

Downloaded from Arrow.com.

11.2.6 Operating Power Control Register (OPCCR)

Address(es): 0008 00A0h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	OPCMTSF	—	OPCM[2:0]		
Value after reset:	0	0	0	0	0	0	1	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	OPCM[2:0]	Operating Power Control Mode Select	b2 b0 0 0 0: High-speed operating mode 0 1 0: Middle-speed operating mode Settings other than above are prohibited.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	OPCMTSF	Operating Power Control Mode Transition Status Flag	0: Transition completed 1: During transition	R
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

The OPCCR register is used to reduce power consumption in normal operating mode, sleep mode, and deep sleep mode. Power consumption can be reduced according to the operating frequency and operating voltage to be used by the OPCCR setting.

The OPCCR register cannot be rewritten under the following conditions:

- When the OPCCR.OPCMTSF flag is 1 (during transition)
- Time period from WAIT instruction execution for a sleep mode transition, until exit from sleep mode to normal operation
- Time period from WAIT instruction execution for a deep sleep mode transition, until exit from deep sleep mode to normal operation
- When the SOPCCR.SOPCM bit is 1 (low-speed operating mode)

The OPCCR register cannot be rewritten while the flash memory is being programmed or erased (P/E).
For the procedures of changing operating power control modes, refer to Function in section 11.5, Function for Lower Operating Power Consumption.
During sleep mode or mode transitions, do not write to the registers related to system control (indicated by 'SYSTEM' in the Module Symbol column in Table 5.1, List of I/O Registers (Address Order)).

OPCM[2:0] Bits (Operating Power Control Mode Select)

The OPCM[2:0] bits select operating power control mode in normal operating mode, sleep mode, and deep sleep mode. Table 11.4 shows the relationship between operating power control modes, the OPCM[2:0] and SOPCM bit settings, and the operating frequency and voltage ranges.

OPCMTSF Flag (Operating Power Control Mode Transition Status Flag)

This flag indicates the switching control state during and after operating power mode transition.
This flag becomes 1 when the value of the OPCM[2:0] bits is rewritten, and 0 when mode transition is completed. Read this flag and confirm that it is 0 before proceeding to the next processing. Only rewrite the OPCM[2:0] bits when this flag is 0.

Downloaded from Arrow.com.

11.2.7 Sub Operating Power Control Register (SOPCCR)

Address(es): 0008 00AAh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	SOPCMTSF	—	—	—	SOPCM
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SOPCM	Sub Operating Power Control Mode Select	0: High-speed operating mode or middle-speed operating mode*1 1: Low-speed operating mode	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	SOPCMTSF	Sub Operating Power Control Mode Transition Status Flag	0: Transition completed 1: During transition	R
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.
Note 1. Depends on the setting of OPCCR.OPCM[2:0].

The SOPCCR register is used to reduce power consumption in normal operating mode, sleep mode, and deep sleep mode by controlling a transition to low-speed operating mode.
Setting this register initiates entry to/exit from low-speed operating mode.
Low-speed operating mode is used for the sub-clock oscillator only.
The OPCCR register cannot be rewritten when the SOPCM bit is 1 (low-speed operating mode).

The SOPCCR register cannot be rewritten under the following conditions:

- When the SOPCCR.SOPCMTSF flag is 1 (during transition)
- Time period from WAIT instruction execution for a sleep mode transition, until exit from sleep mode to normal operation
- Time period from WAIT instruction execution for a deep sleep mode transition, until exit from deep sleep mode to normal operation

This register cannot be rewritten while the flash memory is being programmed or erased (P/E).
For the procedures for changing operating power control modes, refer to Function in section 11.5, Function for Lower Operating Power Consumption.
During sleep mode or mode transitions, do not write to the registers related to system control (indicated by 'SYSTEM' in the Module Symbol column in Table 5.1, List of I/O Registers (Address Order)).

SOPCM Bit (Sub Operating Power Control Mode Select)

The SOPCM bit selects operating power control in normal operating mode and sleep mode.
Setting this bit to 1 allows a transition to low-speed operating mode. Setting this bit to 0 allows a return to the operating mode (operating mode set by OPCCR.OPCM[2:0]) before the transition to low-speed operating mode.
Table 11.4 shows the relationship between operating power control modes, the OPCM[2:0] and SOPCM bit settings, and the operating frequency and voltage ranges.

Downloaded from Arrow.com.

SOPCMTSF Flag (Sub Operating Power Control Mode Transition Status Flag)

The SOPCMTSF flag indicates the switching control state when the sub operating power control mode is switched. This flag becomes 1 when the value of the SOPCM bit is rewritten, and 0 when mode transition is completed. Read this flag and confirm that it is 0 before proceeding to the next processing. Only rewrite the SOPCM bit when this flag is 0.

Table 11.4 Operating Frequency and Voltage Ranges in Operating Power Control Modes

Operating Power Control Mode	OPCM[2:0] Bits	SOPCM Bit	Operating Voltage Range	Operating Frequency Range						MEMWAIT Bit
				Flash Memory Read Frequency					Flash Memory Programming /Erasure Frequency	
				ICLK	FCLK	PCLKD	PCLKB	PCLKA	FCLK	
High-speed operating mode	000b	0	2.7 to 3.6 V	32 to 54 MHz	Up to 32 MHz	Up to 54 MHz	Up to 32 MHz	Up to 54 MHz	1 to 32 MHz	1*1
				Up to 32 MHz						0/1*1
			2.4 to 2.7 V	Up to 16 MHz	Up to 16 MHz	Up to 32 MHz	Up to 16 MHz	Up to 16 MHz	—	0*2
			1.8 to 2.4 V	Up to 8 MHz	Up to 8 MHz	Up to 8 MHz	Up to 8 MHz	Up to 8 MHz	—	0*2
Middle-speed operating mode	010b	0	2.4 to 3.6 V	Up to 12 MHz	Up to 12 MHz	Up to 12 MHz	Up to 12 MHz	Up to 12 MHz	1 to 12 MHz	0
			1.8 to 2.4 V	Up to 8 MHz	Up to 8 MHz	Up to 8 MHz	Up to 8 MHz	Up to 8 MHz	1 to 8 MHz	0
Low-speed operating mode	000b	1	1.8 to 3.6 V	Up to 32.768 kHz	Up to 32.768 kHz	Up to 32.768 kHz	Up to 32.768 kHz	Up to 32.768 kHz	—	0
	010b	1	1.8 to 3.6 V							

Note: When using the FCLK at lower than 4 MHz during programming or erasing the flash memory, the frequency can be set to 1, 2, or 3 MHz.
Note 1. Change the value according to the procedure for setting the MEMWAIT bit.
Note 2. It is not necessary to set the MEMWAIT bit to 1.

Each operating power control mode is described below.

Downloaded from Arrow.com.

- High-Speed Operating Mode

The maximum operating frequency during FLASH read is 54 MHz for ICLK, PCLKA, and PCLKD; 32 MHz for PCLKB, and FCLK. The operating voltage range is 1.8 to 3.6 V during FLASH read. However, the maximum operating frequency during FLASH read is 16 MHz for ICLK, FCLK, PCLKA, and PCLKB, and 32 MHz for PCLKD when the operating voltage is 2.4 V or larger and smaller than 2.7 V. The maximum operating frequency during FLASH read is 8 MHz for all the clocks when the operating voltage is 1.8 V or larger and smaller than 2.4 V.
During FLASH programming/erasure, the operating frequency range is 1 to 32 MHz and the operating voltage range is 2.7 to 3.6 V.
The following restriction applies when high-speed operating mode is selected.

- The PLL and USB-dedicated PLL can be used when the operating voltage is 2.4 V or above.

However, the USB cannot be used when the operating voltage is below 3.0 V.
Figure 11.3 shows the operating voltages and frequencies in high-speed operating mode.

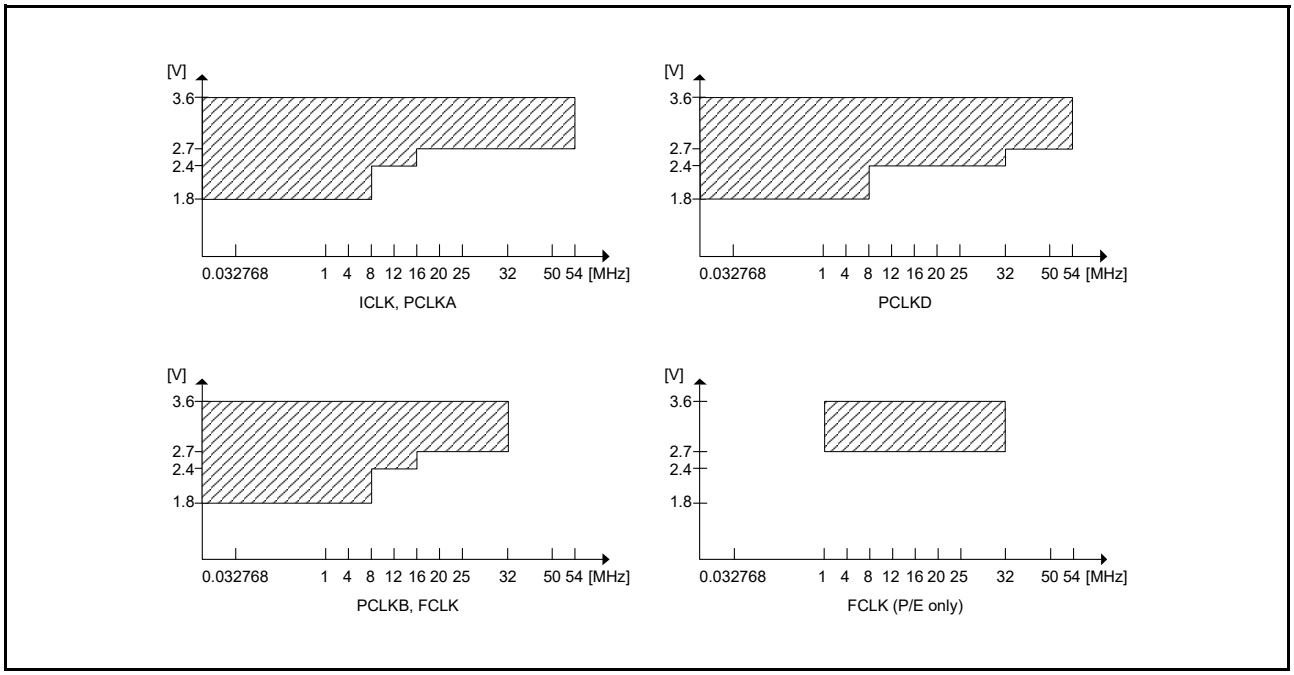


Figure 11.3 Operating Voltages and Frequencies in High-Speed Operating Mode

Note: When using the FCLK at lower than 4 MHz during programming or erasing the flash memory, the frequency can be set to 1, 2, or 3 MHz.

Downloaded from Arrow.com.

- Middle-Speed Operating Mode

As compared to high-speed operating mode, this mode reduces power consumption for low-speed operation.
The maximum operating frequency during FLASH read is 12 MHz for ICLK, FCLK, PCLKA, PCLKB, and PCLKD.
The operating voltage range is 1.8 to 3.6 V during FLASH read. The maximum operating frequency during FLASH read is 8 MHz for all the clocks when the operating voltage is 1.8 V or larger and smaller than 2.4 V.
During FLASH programming/erasure, the operating frequency range is 1 to 12 MHz and the operating voltage range is 1.8 to 3.6 V. The maximum operating frequency during FLASH programming/erasure is 8 MHz when the operating voltage is 1.8 V or larger and smaller than 2.4 V.
The power consumption of this mode is lower than that of high speed mode under the same conditions.
After a reset is canceled, operation is started from this mode.
The following restriction applies when middle-speed operating mode is selected:

- The PLL and USB-dedicated PLL can be used when the operating voltage is 2.4 V or above.

However, the USB cannot be used when the operating voltage is below 3.0 V.
Figure 11.4 shows the operating voltages and frequencies in middle-speed operating mode.

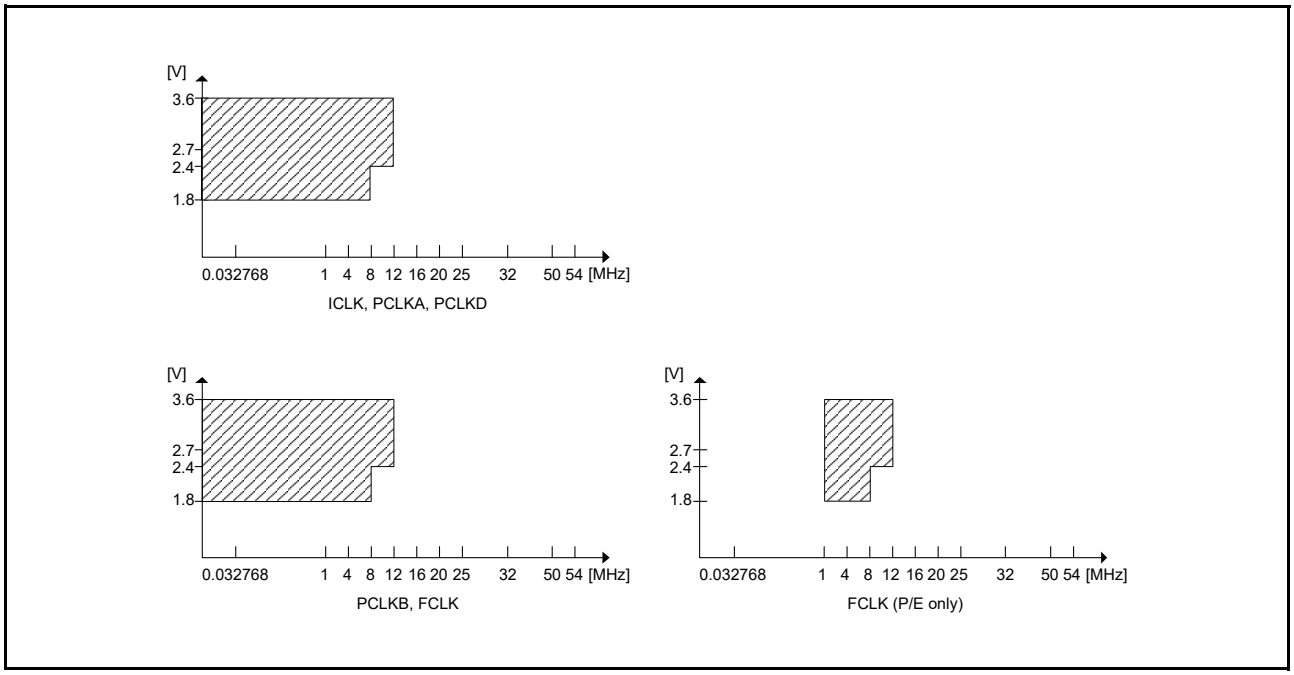


Figure 11.4 Operating Voltages and Frequencies in Middle-Speed Operating Mode

Note: When using the FCLK at lower than 4 MHz during programming or erasing the flash memory, the frequency can be set to 1, 2, or 3 MHz.

Downloaded from Arrow.com.

- Low-Speed Operating Mode

A transition to low-speed operating mode is set by writing 1 to the SOPCM bit in the SOPCCR register. The setting of the OPCM[2:0] bits cannot be modified during low-speed operating mode. This mode is used only for the sub oscillator of 32.768 kHz.

During reading the flash memory (FLASH), the maximum operating frequency of ICLK, FCLK, PCLKA, PCLKB, and PCLKD is 32.768 kHz. The operating voltage is in the range of 1.8 to 3.6 V.

The following restrictions apply when low-speed operating mode is selected:

- P/E operations for flash memory are prohibited.
- The PLL/USB-dedicated PLL, main clock oscillator, LOCO, and HOCO cannot be used.

Note: The SOPCM bit cannot be set to 1 when the PLLCR2.PLLEN bit is 0 (PLL is operating).
The SOPCM bit cannot be set to 1 when the UPLLCR2.UPLLEN bit is 0 (USB-dedicated PLL is operating).
The SOPCM bit cannot be set to 1 when the HOCOCR.HCSTP bit is 0 (HOCO is operating).
The SOPCM bit cannot be set to 1 when the MOSCCR.MOSTP bit is 0 (main clock oscillator is operating).
The SOPCM bit cannot be set to 1 when the LOCOCR.LCSTP bit is 0 (LOCO is operating).

Figure 11.5 shows the operating voltages and frequencies in low-speed operating mode.

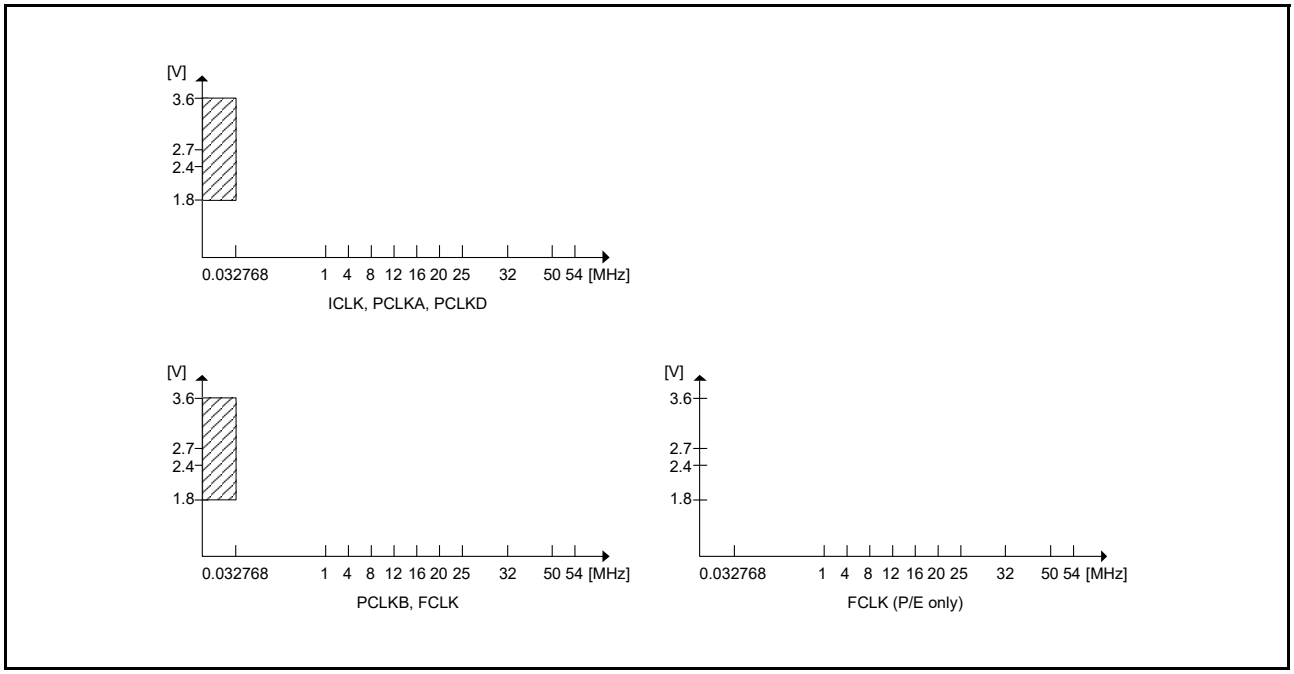


Figure 11.5 Operating Voltages and Frequencies in Low-Speed Operating Mode

Downloaded from Arrow.com.

11.2.8 Sleep Mode Return Clock Source Switching Register (RSTCKCR)

Address(es): 0008 00A1h

	b7	b6	b5	b4	b3	b2	b1	b0
	RSTCKEN	—	—	—	—	RSTCKSEL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	RSTCKSEL [2:0]	Sleep Mode Return Clock Source Select	b2 b0 0 0 0: LOCO is selected 0 0 1: HOCO is selected*1 0 1 0: Main clock oscillator is selected Settings other than above are prohibited when the RSTCKEN bit is 1.	R/W
b6 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	RSTCKEN	Sleep Mode Return Clock Source Switching Enable	0: Clock source switching at exit from sleep mode is disabled 1: Clock source switching at exit from sleep mode is enabled	R/W

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.
Note 1. HOCO can only be selected when entering high-speed operating mode.

RSTCKCR is used to control clock source switching at exit from sleep mode.
When exit from sleep mode is initiated by setting RSTCKCR, the main clock oscillator stop bit in the main clock oscillator control register (MOSCCR.MOSTP), the HOCO stop bit in the high-speed on-chip oscillator control register (HOCOCR.HCSTP), and the LOCO stop bit in the low-speed on-chip oscillator control register (LOCOCR.LCSTP) are automatically modified to the operating state corresponding to the clock source to be used after transition. The value of the RSTCKSEL[2:0] bits is automatically reloaded to the clock source select bits in system clock control register 3 (SCKCR3.CKSEL[2:0]).

RSTCKSEL[2:0] Bits (Sleep Mode Return Clock Source Select)

The RSTCKSEL[2:0] bits select the clock source to be used at exit from sleep mode.
The clock source selected by the RSTCKSEL[2:0] bits is enabled only when the RSTCKEN bit is 1.
As shown in Figure 11.2, Operating Modes, when returning from sleep mode to high-speed operating mode, the LOCO, HOCO, or main clock oscillator can be selected. When returning from sleep mode to middle-speed operating mode, the LOCO or main clock oscillator can be selected. However, in this case, the frequency of each clock (ICLK, FCLK, PCLKA, PCLKB, and PCLKD) must be lower than 12 MHz when the power supply voltage is 2.4 V or above, and lower than 8 MHz when the voltage is below 2.4 V.

Table 11.5 When Exiting Sleep Mode to High-Speed Operating Mode and Middle-Speed Operating Mode

Operating Mode during Sleep	Clock Source during Sleep	RSTCKSEL	Operating Mode after Exiting	Clock Source after Exiting
High-speed operating mode or low-speed operating mode after exit from high-speed operating mode	Sub-clock oscillator	000b (LOCO)	High-speed operating mode	LOCO
		001b (HOCO)		HOCO
		010b (main clock oscillator)		Main clock oscillator
Middle-speed operating mode or low-speed operating mode after exit from middle-speed operating mode	Sub-clock oscillator	000b (LOCO)	Middle-speed operating mode	LOCO
		010b (main clock oscillator)		Main clock oscillator*1

Note 1. The frequency of each clock (ICLK, FCLK, PCLKA, PCLKB, and PCLKD) must be lower than 12 MHz when the power supply voltage is 2.4 V or above, and lower than 8 MHz when the voltage is below 2.4 V.

Downloaded from Arrow.com.

RSTCKEN Bit (Sleep Mode Return Clock Source Switching Enable)

The RSTCKEN bit enables or disables clock source switching when sleep mode is exited.

The clock source can be switched when exiting sleep mode only while the sub-clock oscillator is selected as the clock for entering sleep mode. Do not enable this bit when entering sleep mode while the HOCO, LOCO, main clock oscillator, or PLL is selected as the clock source.

When returning from sleep mode while this bit is enabled, the SOPCM bit in the SOPCCR register is automatically rewritten to 0 (middle-speed operating mode or high-speed operating mode).

The value of the frequency division setting (in the SCKCR register) is retained.

To exit sleep mode to middle-speed operating mode when the main clock oscillator is selected, the frequency of each clock must be lower than 12 MHz when the power supply voltage is 2.4 V or above, and lower than 8 MHz when the voltage is below 2.4 V.

Downloaded from Arrow.com.

11.3 Reducing Power Consumption by Switching Clock Signals

The clock frequency can change by setting the SCKCR.FCK[3:0], ICK[3:0], PCKA[3:0], PCKB[3:0], and PCKD[3:0] bits. The CPU, DMAC, DTC, ROM, and RAM clocks can be set by the ICK[3:0] bits. The peripheral module clocks can be set by the PCKA[3:0], PCKB[3:0], and PCKD[3:0] bits.
The flash memory clock can be set by the FCK[3:0] bits.
For details, refer to section 9, Clock Generation Circuit.

11.4 Module Stop Function

The module stop function can be set for each on-chip peripheral module.
When the MSTPmi bit (m = A to D; i = 0 to 31) in MSTPCRA to MSTPCRD is set to 1, the specified module stops operating and enters the module stop state, but the CPU continues to operate independently. When the corresponding MSTPmi bit is set to 0, the module exits the module state and restarts operating at the end of the bus cycle. The internal states of modules are retained in the module stop state.
After a reset is canceled, all modules other than the DMAC, DTC, and on-chip RAM are in the module stop state.
Basically the registers in the module stop state cannot be read or written. However, note that data may be written to these registers if write access is made immediately after the setting of the module stop state. To avoid this, always write to the module stop registers after confirming that the last register setting is done.

11.5 Function for Lower Operating Power Consumption

By selecting an appropriate operating power control mode according to the operating frequency and operating voltage, power consumption can be reduced in normal mode, sleep mode, and deep sleep mode.

11.5.1 Setting Operating Power Control Mode

Examples of the procedures for switching operating power control modes are shown below:

(1) Switching from Normal Power Consumption Mode to Low Power Consumption Mode

- Example 1: From high-speed operating mode to middle-speed operating mode

(High-speed operation in high-speed operating mode)
↓
Set the frequency of each clock to lower than the maximum operating frequency for middle-speed operating mode
↓
Confirm that the OPCCR.OPCMTSF flag is 0 (transition completed)
↓
Set the OPCCR.OPCM[2:0] bits to 010b (middle-speed operating mode)
↓
Confirm that the OPCCR.OPCMTSF flag is 0 (transition completed)
↓
(Middle-speed operation in middle-speed operating mode)

Downloaded from Arrow.com.

- Example 2: From high-speed/middle-speed operating mode to low-speed operating mode

(High-speed operation in high-speed operating mode/middle-speed operation in middle-speed operating mode)
↓
Set the frequency of each clock to lower than the maximum operating frequency for low-speed operating mode
↓
Confirm that all clock sources but the sub-clock oscillator are stopped
↓
Confirm that the SOPCCR.SOPCMTSF flag is 0 (transition completed)
↓
Set the SOPCCR.SOPCM bit to 1 (low-speed operating mode)
↓
Confirm that the SOPCCR.SOPCMTSF flag is 0 (transition completed)
↓
Low-speed operation in low-speed operating mode

(2) Switching from Low Power Consumption Mode to Normal Power Consumption Mode

- Example 1: From low-speed operating mode to high-speed/middle-speed operating mode

Low-speed operation in low-speed operating mode
↓
Confirm that the SOPCCR.SOPCMTSF flag is 0 (transition completed)
↓
Set the SOPCCR.SOPCM bit to 0 (high-speed operating mode or middle-speed operating mode)
↓
Confirm that the SOPCCR.SOPCMTSF flag is 0 (transition completed)
↓
Set the frequency of each clock to lower than the maximum operating frequency for high-speed/middle-speed operating mode
↓
(High-speed operation in high-speed operating mode/middle-speed operation in middle-speed operating mode)

- Example 2: From middle-speed operating mode to high-speed operating mode

Middle-speed operation in middle-speed operating mode
↓
Confirm that the OPCCR.OPCMTSF flag is 0 (transition completed)
↓
Set the OPCCR.OPCM[2:0] bit to 0 (high-speed operating mode)
↓
Confirm that the OPCCR.OPCMTSF flag is 0 (transition completed)
↓
Set the frequency of each clock to lower than the maximum operating frequency for high-speed operating mode
↓
High-speed operation in high-speed operating mode

Downloaded from Arrow.com.

11.6 Low Power Consumption Modes

11.6.1 Sleep Mode

11.6.1.1 Entry to Sleep Mode

When the WAIT instruction is executed while the SBYCR.SSBY bit is 0, the CPU enters sleep mode. In sleep mode, the CPU stops operating but the contents of its internal registers are retained. Other peripheral functions do not stop.
When the WDT is used, the WDT stops counting when sleep mode is entered.
Counting by the IWDT stops if a transition to sleep mode is made while the IWDT is being used in auto-start mode and the OFS0.IWDTSLCSTP bit is 1. In the same way, counting by the IWDT stops if a transition to sleep mode is made while the IWDT is being used in register start mode and the IWDTCSTPR.SLCSTP bit is 1.
Furthermore, counting by the IWDT continues if a transition to sleep mode is made while the IWDT is being used in auto-start mode and the OFS0.IWDTSLCSTP bit is 0 (counting by the IWDT continues through transitions to low power consumption modes). In the same way, counting by the IWDT continues if a transition to sleep mode is made while the IWDT is being used in register start mode and the IWDTCSTPR.SLCSTP bit is 0.
To use sleep mode, make the following settings and then execute a WAIT instruction.

(1) Set the PSW.I bit*1 of the CPU to 0.
(2) Set the interrupt request destination*2 to be used for exit from sleep mode.
(3) Set the priority*3 of the interrupt to be used for exit from sleep mode to a level higher than the setting of the PSW.IPL[3:0] bits*1 of the CPU.
(4) Set the IERm.IENj bit*3 to 1 for the interrupt.
(5) Read the I/O register that is written last and confirm that the written value has been reflected.
(6) Execute the WAIT instruction (this automatically sets the I bit*1 in the PSW of the CPU to 1).

Note 1. For details, refer to section 2, CPU.
Note 2. For details, refer to section 15.4.3, Selecting Interrupt Request Destinations.
Note 3. For details, refer to section 15, Interrupt Controller (ICUb).

Downloaded from Arrow.com.

11.6.1.2 Exit from Sleep Mode

Exit from sleep mode is initiated by any interrupt, a RES# pin reset, a power-on reset, a voltage monitoring reset, or a reset caused by an IWDT underflow.

- Initiated by an interrupt
  An interrupt initiates exit from sleep mode and the interrupt exception handling starts. If a maskable interrupt has been masked by the CPU (the priority level*1 of the interrupt has been set to a value lower than that of the PSW.IPL[3:0] bits*2 of the CPU), sleep mode is not exited.
- Initiated by a RES# pin reset
  When the RES# pin is driven low, the MCU enters the reset state. When the RES# pin is driven high after the reset signal is input for a predetermined time period, the CPU starts the reset exception handling.
- Initiated by a power-on reset
  A power-on reset asserts a reset to the MCU.
  When a power-on reset is negated by a rise in the supply voltage, the CPU starts the reset exception handling.
- Initiated by a voltage monitoring reset
  A voltage monitoring reset asserts a reset to the MCU.
  When a voltage monitoring reset is negated by a rise in the supply voltage, the CPU starts the reset exception handling.
- Initiated by an independent watchdog timer reset
  An internal reset generated by an IWDT underflow asserts a reset to the MCU. However, when IWDT counting is stopped in sleep mode by setting OFS0.IWDTSTRT = 0 and OFS0.IWDTSLCSTP = 1, or OFS0.IWDTSTRT = 1 and IWDTCSTPR.SLCSTP = 1, the IWDT is stopped in sleep mode and sleep mode is not exited by the independent watchdog timer reset.

Note 1. For details, refer to section 15, Interrupt Controller (ICUb).
Note 2. For details, refer to section 2, CPU.

11.6.1.3 Sleep Mode Return Clock Source Switching Function

To switch the clock source used for exit from sleep mode, set the sleep mode return clock source switching register (RSTCKCR) and the wait control register for each clock. When the return interrupt is generated, after oscillation settling of the oscillator specified as the return clock, the clock source is automatically switched, and then operation exits sleep mode. At this time, the registers related to clock source switching are automatically rewritten.
For details, refer to section 11.2.8, Sleep Mode Return Clock Source Switching Register (RSTCKCR).
For details on settings the oscillation stabilization wait time, refer to section 9.2.16, Main Clock Oscillator Wait Control Register (MOSCWTCR).

Downloaded from Arrow.com.

11.6.2 Deep Sleep Mode

11.6.2.1 Entry to Deep Sleep Mode

When a WAIT instruction is executed with the MSTPCRC.DSLPE bit set to 1, the MSTPCRA.MSTPA28 bit set to 1, and the SBYCR.SSBY bit cleared to 0, a transition to deep sleep mode is made.
In deep sleep mode, the CPU and the DMAC, DTC, ROM, and RAM clocks stop. Peripheral functions do not stop.
When the WDT is used, the WDT stops counting when deep sleep mode is entered.
Counting by the IWDT stops if a transition to deep sleep mode is made while the IWDT is being used in auto-start mode and the OFS0.IWDTSLCSTP bit is 1. In the same way, counting by the IWDT stops if a transition to deep sleep mode is made while the IWDT is being used in register start mode and the IWDTCSTPR.SLCSTP bit is 1.
Furthermore, counting by the IWDT continues if a transition to deep sleep mode is made while the IWDT is being used in auto-start mode and the OFS0.IWDTSLCSTP bit is 0 (counting by the IWDT continues through transitions to low power consumption modes). In the same way, counting by the IWDT continues if a transition to deep sleep mode is made while the IWDT is being used in register start mode and the IWDTCSTPR.SLCSTP bit is 0.

To use deep sleep mode, make the following settings and then execute a WAIT instruction.

(1) Set the PSW.I bit*1 of the CPU to 0.
(2) Set the interrupt request destination*2 to be used for exit from deep sleep mode.
(3) Set the priority*3 of the interrupt to be used for exit from deep sleep mode to a level higher than the setting of the PSW.IPL[3:0] bits*1 of the CPU.
(4) Set the IERm.IENj bit*3 to 1 for the interrupt.
(5) Read the I/O register that is written last and confirm that the written value has been reflected.
(6) Execute a WAIT instruction (executing a WAIT instruction causes automatic setting of the PSW.I bit*1 of the CPU to 1).

Note 1. For details, refer to section 2, CPU.
Note 2. For details, refer to section 15.4.3, Selecting Interrupt Request Destinations.
Note 3. For details, refer to section 15, Interrupt Controller (ICUb).

Downloaded from Arrow.com.

11.6.2.2 Exit from Deep Sleep Mode

Exit from deep sleep mode is initiated by any interrupt, a RES# pin reset, a power-on reset, a voltage monitoring reset, or a reset caused by an IWDT underflow.

- Initiated by an interrupt
  An interrupt initiates exit from deep sleep mode and the interrupt exception handling starts. If a maskable interrupt has been masked by the CPU (the priority level*1 of the interrupt has been set to a value lower than that of the PSW.IPL[3:0] bits*2 of the CPU), deep sleep mode is not exited.
- Initiated by the RES# pin reset
  When the RES# pin is driven low, the MCU enters the reset state. When the RES# pin is driven high after the reset signal is input for a predetermined time period, the CPU starts the reset exception handling.
- Initiated by a power-on reset
  A power-on reset asserts a reset to the MCU.
  When a power-on reset is negated by a rise in the supply voltage, the CPU starts the reset exception handling.
- Initiated by a voltage monitoring reset
  A voltage monitoring reset asserts a reset to the MCU.
  When a voltage monitoring reset is negated by a rise in the supply voltage, the CPU starts the reset exception handling.
- Initiated by the independent watchdog timer
  An internal reset generated by an IWDT underflow asserts a reset to the MCU. However, when IWDT counting is stopped in deep sleep mode by setting OFS0.IWDTSTRT = 0 and OFS0.IWDTSLCSTP = 1, or OFS0.IWDTSTRT = 1 and IWDTCSTPR.SLCSTP = 1, the IWDT is stopped in deep sleep mode and deep sleep mode is not exited by the independent watchdog timer reset.

Note 1. For details, refer to section 15, Interrupt Controller (ICUb).
Note 2. For details, refer to section 2, CPU.

Downloaded from Arrow.com.

11.6.3 Software Standby Mode

11.6.3.1 Entry to Software Standby Mode

When a WAIT instruction is executed with the SBYCR.SSBY bit set to 1, a transition to software standby mode is made. In this mode, the CPU, on-chip peripheral functions, and all the other functions except the sub-clock oscillator stop. However, the contents of the CPU internal registers, RAM data, the states of on-chip peripheral functions, the I/O ports, and the sub-clock oscillator are retained. Software standby mode allows significant reduction in power consumption because the oscillator stops in this mode.
Set the DMAST.DMST bit of the DMAC and the DTCST.DTCST bit of the DTC to 0 before executing the WAIT instruction.
When the WDT is used, the WDT stops counting when software standby mode is entered.
Counting by the IWDT stops if a transition to software standby mode is made while the IWDT is being used in auto-start mode and the OFS0.IWDTSLCSTP bit is 1. In the same way, counting by the IWDT stops if a transition to software standby mode is made while the IWDT is being used in register start mode and the IWDTCSTPR.SLCSTP bit is 1.

Furthermore, counting by the IWDT continues if a transition to software standby mode is made while the IWDT is being used in auto-start mode and the OFS0.IWDTSLCSTP bit is 0 (counting by the IWDT continues through transitions to low power consumption modes). In the same way, counting by the IWDT continues if a transition to software standby mode is made while the IWDT is being used in register start mode and the IWDTCSTPR.SLCSTP bit is 0.
To use software standby mode, make the following settings and then execute a WAIT instruction.

(1) Set the PSW.I bit*1 of the CPU to 0.
(2) Set the interrupt request destination*2 to be used for recovery from software standby mode to the CPU.
(3) Set the priority*3 of the interrupt to be used for recovery from software standby mode to a level higher than the setting of the PSW.IPL[3:0] bits*1 of the CPU.
(4) Set the IERm.IENj bit*3 to 1 for the interrupt.
(5) Read the I/O register that is written last and confirm that the written value has been reflected.
(6) Execute a WAIT instruction (executing a WAIT instruction causes automatic setting of the PSW.I bit*1 of the CPU to 1).

Note 1. For details, refer to section 2, CPU.
Note 2. For details, refer to section 15.4.3, Selecting Interrupt Request Destinations.
Note 3. For details, refer to section 15, Interrupt Controller (ICUb).

Downloaded from Arrow.com.

11.6.3.2 Exit from Software Standby Mode

Exit from software standby mode is initiated by an external pin interrupt (the NMI, IRQ0, IRQ1, IRQ4 to IRQ7), peripheral function interrupts (the RTC alarm, RTC interval, IWDT, voltage monitoring, VBATT pin voltage drop detection, USB, and ELC (LPT-dedicated interrupt)), a RES# pin reset, a power-on reset, a voltage monitoring reset, or an independent watchdog timer reset. When any trigger which initiates exit from software standby mode is asserted, the oscillators which were operating before entry to software standby mode restart operation. After the oscillation of all these oscillators has been stabilized, operation returns from software standby mode.

- Initiated by an interrupt
  When an interrupt request from among the NMI, IRQ0, IRQ1, IRQ4 to IRQ7, RTC alarm, RTC interval, IWDT, voltage monitoring, VBATT pin voltage drop detection, USB, and ELC (LPT-dedicated interrupt) interrupts is generated, each of the oscillators which was operating before the transition to software standby mode resumes oscillation. After the oscillation stabilization wait time of each oscillator set by the MOSCWTCR.MSTS[4:0] bits has elapsed, the MCU exits software standby mode and interrupt exception processing starts.
- Initiated by a RES# pin reset
  Clock oscillation starts when the low level is applied to the RES# pin. Clock supply for the MCU starts at the same time. Keep the level on the RES# pin low over the time required for oscillation of the clocks to become stable. Reset exception processing starts when the high level is applied to the RES# pin.
- Initiated by a power-on reset
  A power-on reset asserts a reset to the MCU.
  When a power-on reset is negated by a rise in the supply voltage, the CPU starts the reset exception handling.
- Initiated by a voltage monitoring reset
  A voltage monitoring reset asserts a reset to the MCU.
  When a voltage monitoring reset is negated by a rise in the supply voltage, the CPU starts the reset exception handling.
- Initiated by an independent watchdog timer reset
  An internal reset generated by an IWDT underflow asserts a reset to the MCU.
  Note that the independent watchdog timer is stopped in software standby mode due to the register settings (OFS0.IWDTSTRT = 0 and OFS0.IWDTSLCSTP = 1, or OFS0.IWDTSTRT = 1 and IWDTCSTPR.SLCSTP = 1) in software standby mode. In that case, exit from software standby mode by the independent watchdog timer reset cannot be done.

Downloaded from Arrow.com.

11.6.3.3 Example of Software Standby Mode Application

Figure 11.6 shows an example of entry to software standby mode by the falling edge of the IRQn pin, and exit from software standby mode by the rising edge of the IRQn pin.
In this example, an IRQn interrupt is accepted with the IRQCRi.IRQMD[1:0] bits of the ICU set to 01b (falling edge), and then the IRQCRi.IRQMD[1:0] bits are set to 10b (rising edge). After that, the SBYCR.SSBY bit is set to 1 and the WAIT instruction is executed. Thus entry to software standby mode is completed. After that, exit from software standby mode is initiated by the rising edge of the IRQn pin.
To exit software standby mode, settings of the interrupt controller (ICU) are also necessary. For details, refer to section 15, Interrupt Controller (ICUb).

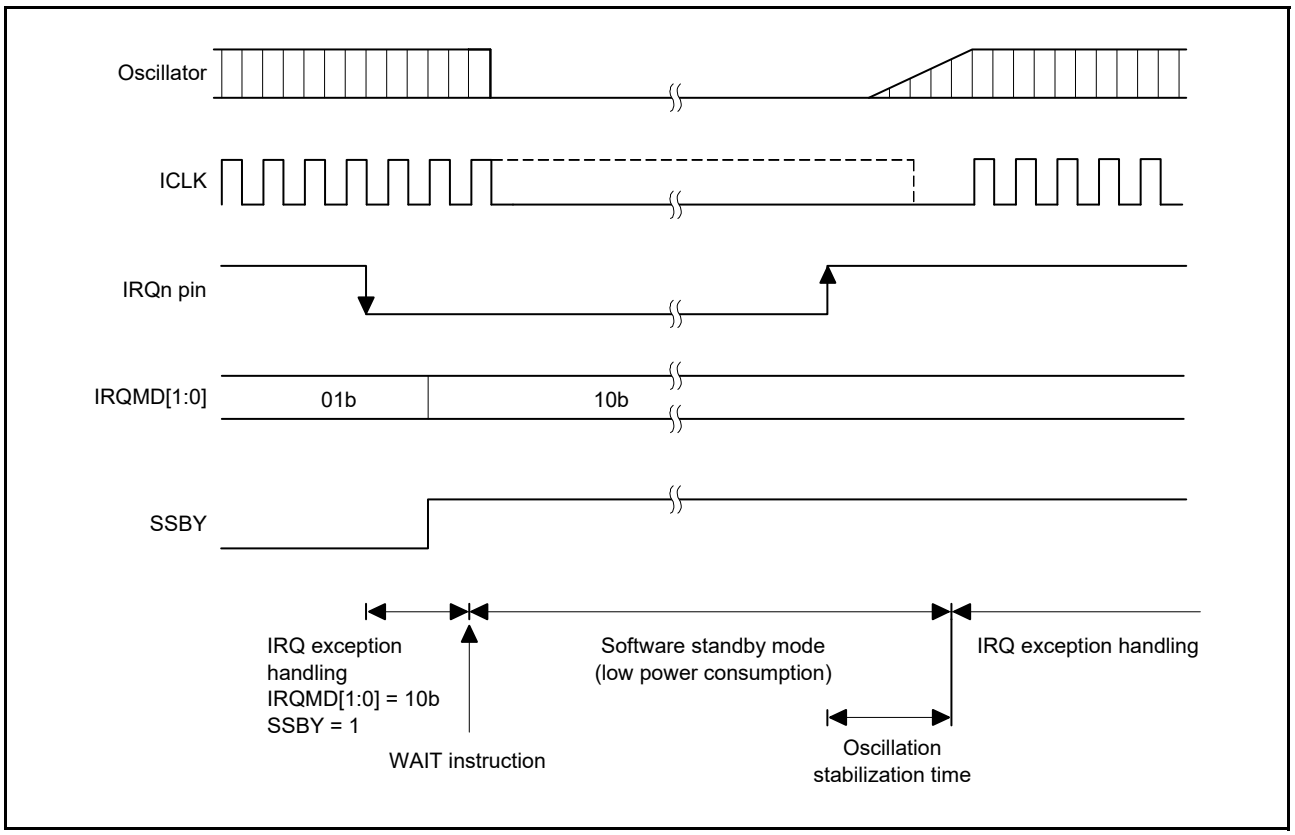


Figure 11.6 Example of Software Standby Mode Application

Downloaded from Arrow.com.

11.7 Usage Notes

11.7.1 I/O Port States

I/O port states are retained in software standby mode. Therefore, the supply current is not reduced if output signals are high level.

11.7.2 Module Stop State of DMAC and DTC

Before setting the MSTPCRA.MSTPA28 bit to 1, set the DMAST.DMST bit of the DMAC and the DTCST.DTCST bit of the DTC to 0 to avoid activating the DMAC and DTC.
For details, refer to section 18, DMA Controller (DMACA) and section 19, Data Transfer Controller (DTCa).

11.7.3 On-Chip Peripheral Module Interrupts

Interrupts do not operate in the module stop state. Therefore, if the module stop state is made after an interrupt request is generated, a CPU interrupt source or a DTC startup source cannot be cleared. For this reason, disable interrupts before entering the module stop state.

11.7.4 Write Access to MSTPCRA, MSTPCRB, MSTPCRC, and MSTPCRD

Write accesses to MSTPCRA, MSTPCRB, MSTPCRC, and MSTPCRD should be made only by the CPU.

11.7.5 Timing of WAIT Instructions

The WAIT instruction is executed before completion of the preceding register write. The WAIT instruction being executed before the register setting is modified may cause unintended operation. To avoid this, always execute the WAIT instruction after confirming that the last register setting is done.

11.7.6 Rewrite the Register by DMAC and DTC in Sleep Mode

The WDT stops in sleep mode. Do not set up the DMAC and DTC to rewrite any registers related to the WDT while the chip is in sleep mode.
Depending on the settings of the OFS0.IWDTSLCSTP bit and IWDTCSTPR.SLCSTP bit, the IWDT may also stop in sleep mode. To avoid this, do not set up the DMAC and DTC to rewrite any registers related to the IWDT in sleep mode.
The RSTCKCR register is a register that switches the clock source at exit from sleep mode. Changing the RSTCKCR register in sleep mode causes unintended operation, so do not write to this register in sleep mode.

Downloaded from Arrow.com.

12. Battery Backup Function

12.1 Overview

When the voltage at the VCC pin is dropped, power can be supplied to the realtime clock (RTC) and the sub-clock oscillator placed in the battery backup power area from the battery backup power pin (VBATT pin).
When the battery backup function is not used, connect the VBATT pin to the VCC pin and disable the battery backup function (set the VBATTCR.VBATTDIS bit to 1).
The VBATT pin voltage level and battery backup power voltage drop detection can be confirmed by reading the VBATT status register.
Figure 12.1 shows the configuration of the battery backup function.

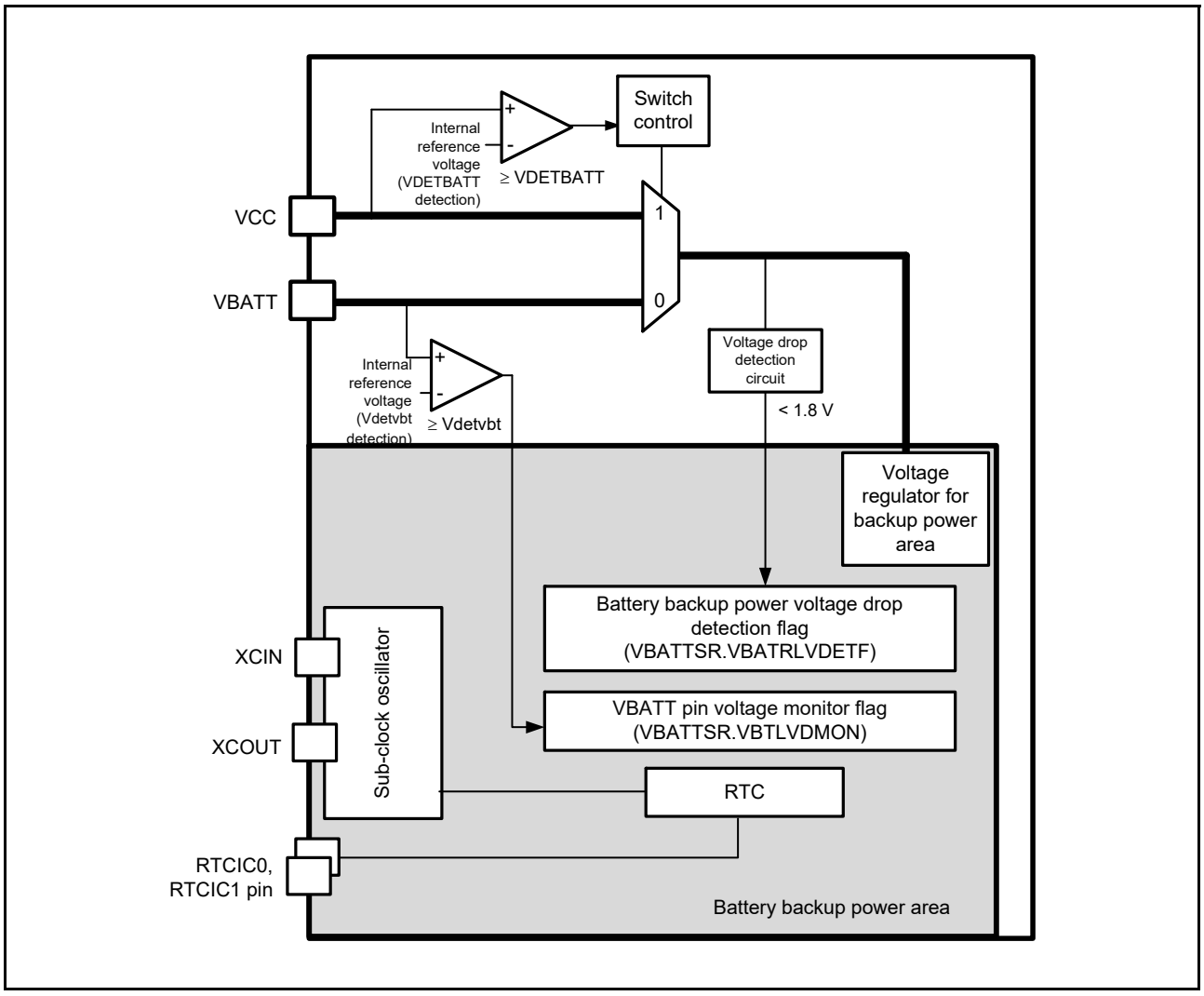


Figure 12.1 Configuration of Battery Backup Function

Downloaded from Arrow.com.

12.2 Register Descriptions

12.2.1 VBATT Control Register (VBATTCR)

Address(es): 0008 C29Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	VBTLVDLVL[1:0]		—	VBTLVDEN	—	—	—	VBATTDIS
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	VBATTDIS	Battery Backup Function Disable	0: Battery backup function enabled 1: Battery backup function disabled	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	VBTLVDEN	VBATT Pin Voltage Drop Detection Enable	0: VBATT pin voltage drop detection disabled 1: VBATT pin voltage drop detection enabled	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7, b6	VBTLVDLVL[1:0]	VBATT Pin Voltage Drop Detection Level Select	b7 b6 1 0: 2.20 V 1 1: 2.00 V Settings other than above are prohibited.	R/W

Note: Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.

The VBATTCR register is used to control operation of the battery backup function.

VBATTDIS Bit (Battery Backup Function Disable)

This bit is used to enable or disable the function for switching the power supply to the backup power area from VCC supply to battery backup power pin (VBATT pin) supply when the voltage at the VCC drops.
When the battery backup function is not used, set the VBATTDIS bit to 1 (battery backup function disabled).

VBTLVDEN Bit (VBATT Pin Voltage Drop Detection Enable)

This bit is used to enable or disable the voltage drop detection function of the VBATT pin.

VBTLVDLVL[1:0] Bits (VBATT Pin Voltage Drop Detection Level Select)

These bits are used to select the detection voltage level (Vdetvbt) when the voltage drop detection function of the VBATT pin is enabled.
VBTLVDLVL bits are enabled when the VBATTCR.VBATTDIS bit is 0 (battery backup function enabled).

Downloaded from Arrow.com.

12.2.2 VBATT Status Register (VBATTSR)

Address(es): 0008 C29Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	VBTLVDMON	VBATRLVDETF
Value after reset:	0	0	0	0	0	0	1	X

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	VBATRLVDETF	Battery Backup Power Voltage Drop Detection Flag	0: Battery backup power voltage drop (< 1.8 V) not detected 1: Battery backup power voltage drop (< 1.8 V) detected	R/W*1
b1	VBTLVDMON	VBATT Pin Voltage Monitor Flag	0: VBATT pin voltage < Vdetvbt 1: VBATT pin voltage ≥ Vdetvbt or VBATT pin voltage drop detection disabled	R
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.
Note 1. Only 0 can be written after 1 is read.

VBATRLVDETF Flag (Battery Backup Power Voltage Drop Detection Flag)

This flag indicates whether the battery backup power voltage has dropped below 1.8 V.

[Setting condition]

- When the battery backup power voltage drops below 1.8 V.

[Clearing condition]

- When 0 is written to the VBATRLVDETF flag after the VBATTSR register is read while this flag is 1.

By setting the VBATRLVDETF flag to 0 after reading as 1 and enabling the backup function (by setting VBATTCR.VBATTDIS to 0) after power-on, whether the battery backup power voltage has dropped while the VCC is low can be confirmed when the VCC rises again after the VCC drops and the power supply is switched to VBATT pin supply.

VBTLVDMON Flag (VBATT Pin Voltage Monitor Flag)

This flag indicates whether the VBATT pin voltage is below Vdetvbt.

This flag is enabled when the VBATTCR.VBATTDIS bit is 0 (battery backup function enabled) and the VBATTCR.VBTLVDEN bit is 1 (VBATT pin voltage drop detection enabled).

Downloaded from Arrow.com.

12.2.3 VBATT Pin Voltage Drop Detection Interrupt Control Register (VBTLVDICR)

Address(es): 0008 C29Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	VBTLVDISEL	VBTLVDIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	VBTLVDIE	VBATT Pin Voltage Drop Detection Interrupt Enable	0: VBATT pin voltage drop detection interrupt disabled 1: VBATT pin voltage drop detection interrupt enabled	R/W
b1	VBTLVDISEL	VBATT Pin Voltage Drop Detection Interrupt Enable Type Selects	0: Non-maskable interrupt 1: Maskable interrupt	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.

Downloaded from Arrow.com.

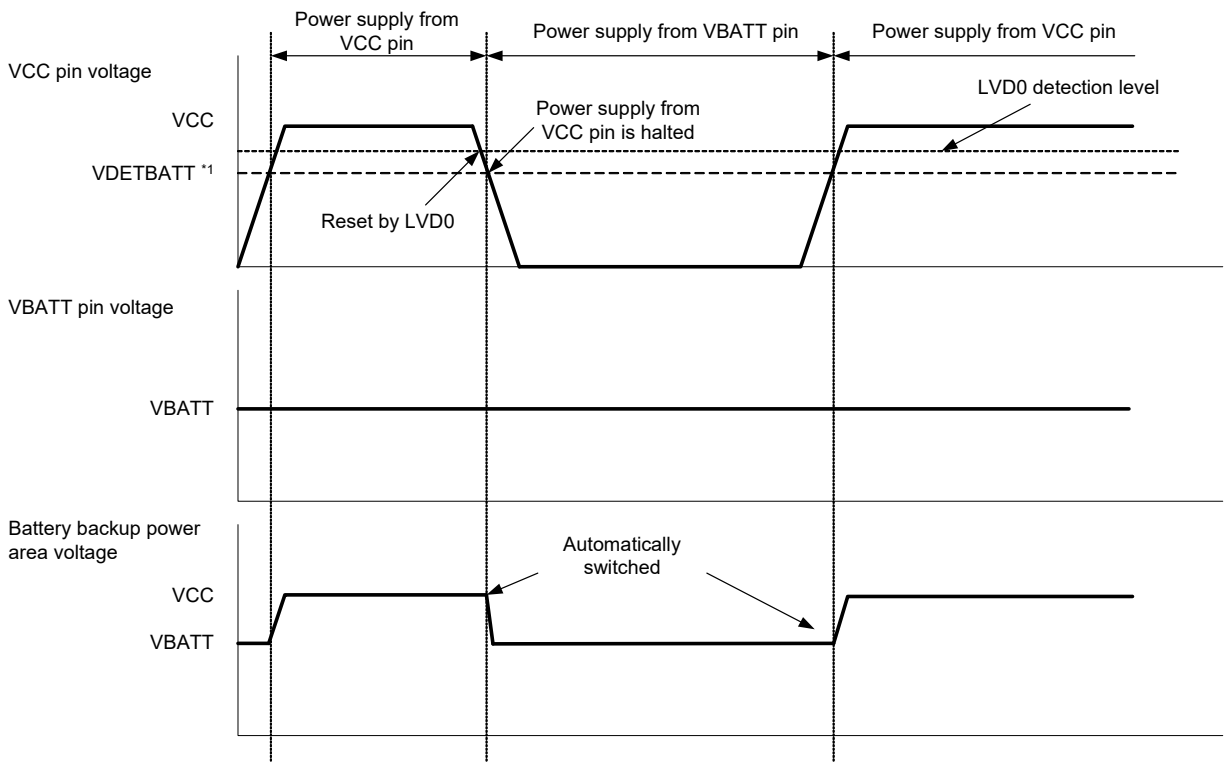
12.3 Operation

12.3.1 Battery Backup Function

When the voltage at the VCC pin is dropped, power can be supplied to the RTC and sub-clock oscillator from the VBATT pin. When the power supply reduction from the VCC pin is detected, connection to power is switched to the power supply from the VBATT pin. The power supply from the VCC pin is resumed when the voltage at the VCC pin exceeds VDETBATT while the RTC is operating on the power supply from the VBATT pin. This power supply change does not affect the RTC operation. When the voltage level at the VBATT pin voltage falls below the operation guaranteed voltage, operation of the RTC cannot be guaranteed.
To use the battery backup function, enable the voltage monitoring 0 reset (set the OFS1.LVDAS bit to 0) and select 2.51 V as the voltage detection 0 level (set OFS1.VDSEL[1:0] to 10b) in advance.
The power is supplied to the following modules from the VBATT pin.

- RTC
- Sub-clock oscillator (including XCIN and XCOUT pins)
- RTCIC0 (P30), RTCIC1 (P31)

Figure 12.2 shows the operation for switching to the battery backup function.
Figure 12.3 shows the operation example of the VBATT pin power voltage monitor flag, and Figure 12.4 shows the operation example of the battery backup power voltage drop detection flag.



Note: For details on electrical characteristics, see section 51., Electrical Characteristics.
Note 1. VDETBATT indicates the judgment level of the power supply change between the VCC pin and the VBATT pin.

Figure 12.2 Operation for Switching to Battery Backup Function

Downloaded from Arrow.com.

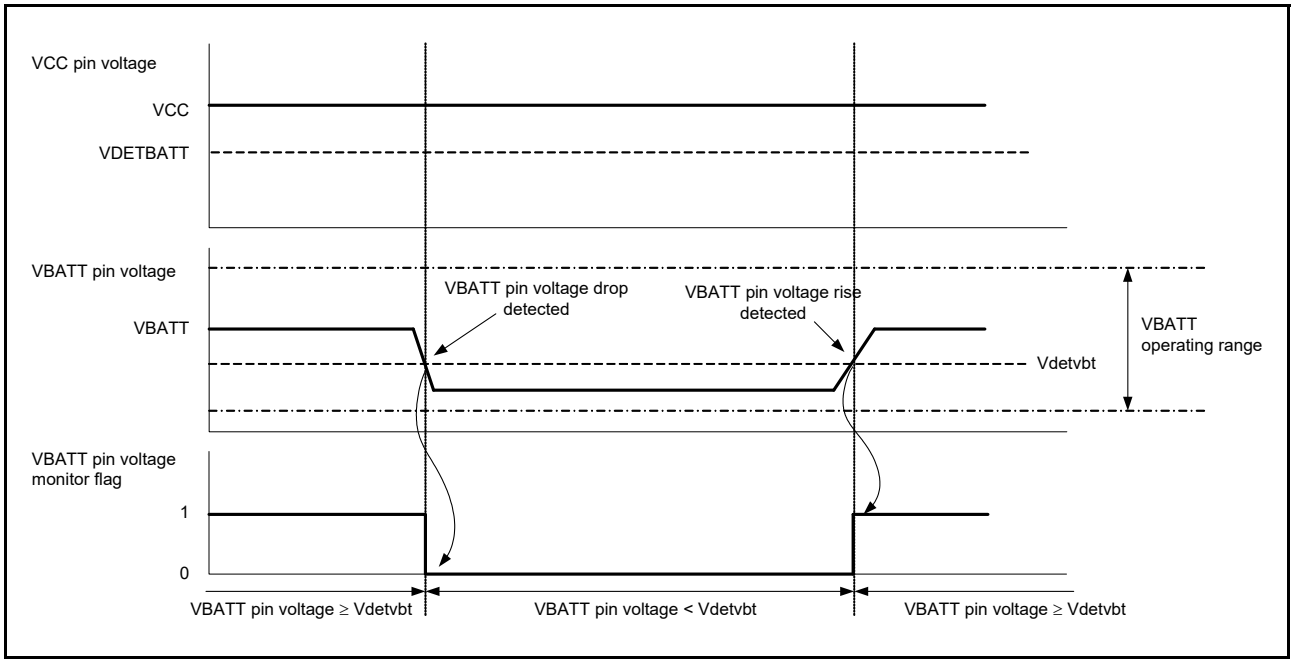


Figure 12.3 Operation Example of the VBATT Pin Power Voltage Monitor Flag

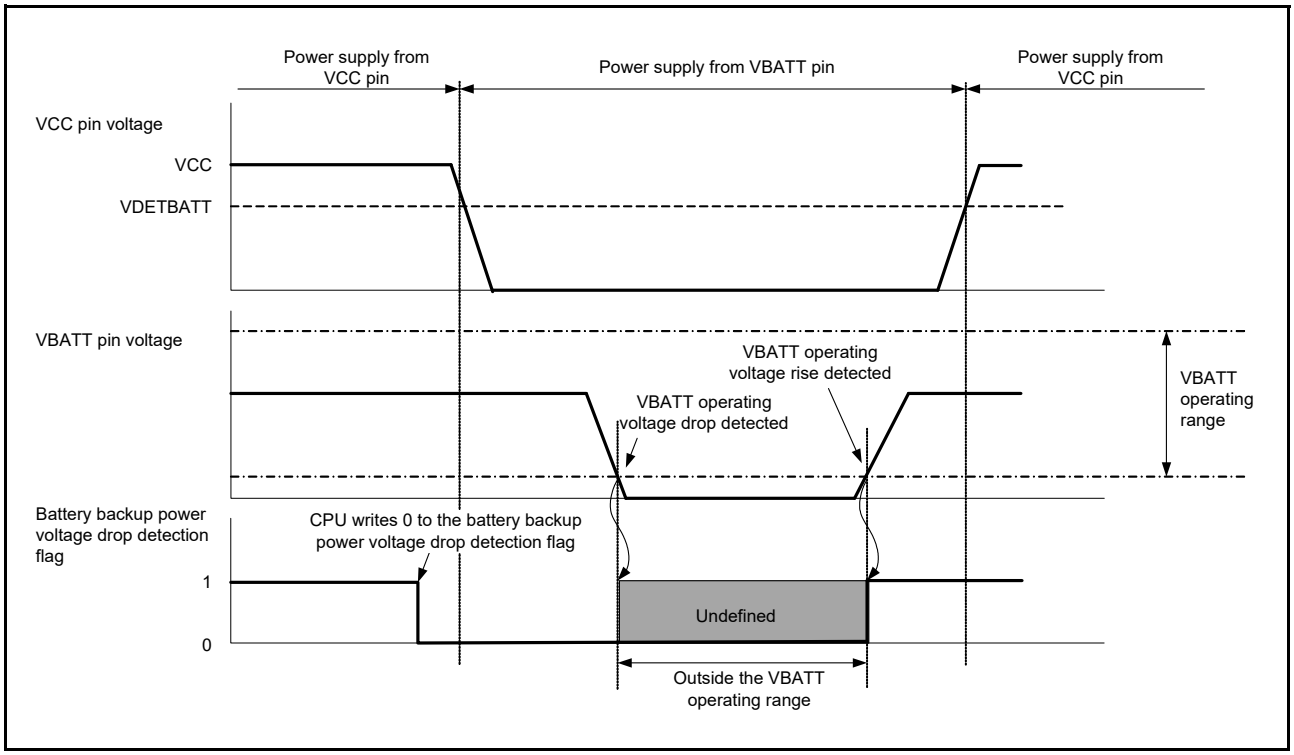


Figure 12.4 Operation Example of the Battery Backup Power Voltage Drop Detection Flag

12.3.2 VBATT Pin Voltage Monitoring Function

When the VCC is supplied, the VBATT pin voltage can be monitored. This function is disabled after a reset.
After the VBATT pin voltage drop detection enable bit (VBATTCR.VBTLVDEN) is set to 1, the VBATT pin voltage level can be monitored by reading the VBATT pin voltage monitor flag (VBATTSR.VBTLVDMON).

Downloaded from Arrow.com.

12.4 Usage Notes

1. When the VBATT pin is not used, connect the VBATT pin to the VCC pin.
2. When the battery backup function is not used, set the VBATTCR.VBATTDIS bit to 1 (battery backup function disabled).
3. When the voltage level at the VBATT is lower than the guaranteed operation range, operation of the sub-clock and RTC cannot be guaranteed. The RTC must be initialized to restart power supply after the VBATT pin falls below the operation guaranteed voltage.
4. Writing to the RTC registers should be performed while power is being supplied from the VCC pin.
5. When VCC is higher than VDETBATT, the VCC pin and VBATT pin are not connected by means of circuitry. When VCC is lower than VDETBATT and the switch is connected to the VBATT pin, if the voltage at the VBATT pin becomes lower than VCC - 0.6 V, current may flow into the VBATT pin through the parasitic diode between the VCC and VBATT pins.

Downloaded from Arrow.com.

13. Register Write Protection Function

The register write protection function protects important registers from being overwritten for in case a program runs out of control. The registers to be protected are set with the protect register (PRCR).

Table 13.1 lists the association between the PRCR bits and the registers to be protected.

Table 13.1 Association between PRCR Bits and Registers to be Protected

PRCR Bit	Register to be Protected
PRC0	• Registers related to the clock generation circuit: SCKCR, SCKCR3, PLLCR, PLLCR2, MOSCCR, SOSCCR, LOCOCR, ILOCOCR, HOCOCR, OSTDCR, OSTDSR, CKOCR, UPLLCR, UPLLCR2, HOCOCR2, MEMWAIT, LOCOTRR, ILOCOTRR, HOCOTRR0, HOCOTRR3
PRC1	• Register related to the operating modes: SYSCR1 • Registers related to low power consumption functions: SBYCR, MSTPCRA, MSTPCRB, MSTPCRC, MSTPCRD, OPCCR, RSTCKCR, SOPCCR • Registers related to the clock generation circuit: MOFCR, MOSCWTCR • Software reset register: SWRR
PRC2	• Registers related to the low power timer: LPTCR1, LPTCR2, LPTCR3, LPTPRD, LPCMR0, LPWUCR
PRC3	• Registers related to the LVD: LVCMPCR, LVDLVLR, LVD1CR0, LVD1CR1, LVD1SR • Registers related to the battery backup function: VBATTCR, VBATTSR, VBTLVDICR

Downloaded from Arrow.com.

13.1 Register Descriptions

13.1.1 Protect Register (PRCR)

Address(es): 0008 03FEh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PRKEY[7:0]								—	—	—	—	PRC3	PRC2	PRC1	PRC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PRC0	Protect Bit 0	Enables writing to the registers related to the clock generation circuit. 0: Write disabled 1: Write enabled	R/W
b1	PRC1	Protect Bit 1	Enables writing to the registers related to operating modes, low power consumption functions, the clock generation circuit, and software reset. 0: Write disabled 1: Write enabled	R/W
b2	PRC2	Protect Bit 2	Enables writing to the registers related to the low power timer. 0: Write disabled 1: Write enabled	R/W
b3	PRC3	Protect Bit 3	Enables writing to the registers related to the LVD. 0: Write disabled 1: Write enabled	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	PRKEY[7:0]	PRC Key Code	These bits control permission and prohibition of writing to the PRCR register. To modify the PRCR register, write A5h to the 8 higher-order bits and the desired value to the 8 lower-order bits as a 16-bit unit.	R/W*1

Note 1. Write data is not retained.

PRCi Bits (Protect Bit i) (i = 0 to 3)

These bits enable or disable writing to the corresponding registers to be protected.

Setting the PRCi bits to 1 and 0 enable and disable writing to the corresponding registers to be protected, respectively.

Downloaded from Arrow.com.

14. Exception Handling

14.1 Exception Events

During execution of a program by the CPU, the occurrence of a certain event may cause execution of that program to be suspended and execution of another program to be started. Such kinds of events are called exception events.
The RXv2 CPU supports eight types of exceptions. The types of exception events are shown in Figure 14.1.
The occurrence of an exception causes the processor mode to shift to supervisor mode.

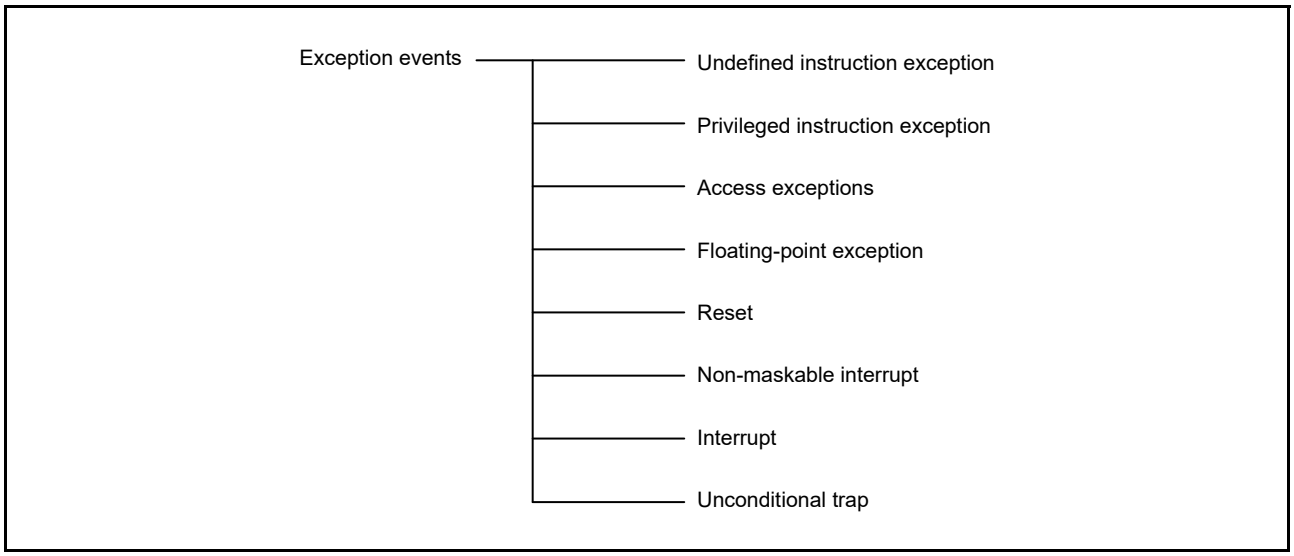


Figure 14.1 Types of Exception Events

Downloaded from Arrow.com.

14.1.1 Undefined Instruction Exception

An undefined instruction exception occurs when execution of an undefined instruction (an instruction not implemented) is detected.

14.1.2 Privileged Instruction Exception

A privileged instruction exception occurs when execution of a privileged instruction is detected in user mode. Privileged instructions can be executed only in supervisor mode.

14.1.3 Access Exceptions

An access exception occurs when an error is detected in access to memory by the CPU. If the memory-protection unit detects an instruction memory-protection error, an instruction-access exception occurs, and if the unit detects a data memory protection error, an operand-access exception occurs.

14.1.4 Floating-Point Exception

Floating-point exceptions include the five exception events (overflow, underflow, inexact, division-by-zero, and invalid operation) specified in the IEEE754 standard and another floating-point exception that is generated on detection of unimplemented processing. The exception handling of floating-point exceptions is prohibited when the EX, EU, EZ, EO, or EV bit in FPSW is 0.

14.1.5 Reset

A reset is generated by input of a reset signal to the CPU. This has the highest priority of any exception and is always accepted.

14.1.6 Non-Maskable Interrupt

The non-maskable interrupt is generated by input of a non-maskable interrupt signal to the CPU and is only used when a fatal fault is considered to have occurred in the system. Never use the non-maskable interrupt with an attempt to return to the program that was being executed at the time of interrupt generation after the exception handling routine is ended.

14.1.7 Interrupt

Interrupts are generated by the input of interrupt signals to the CPU. A fast interrupt can be selected as the interrupt with the highest priority. In the case of the fast interrupt, hardware pre-processing and hardware post-processing are handled fast. The priority level of the fast interrupt is 15 (the highest). The exception handling of interrupts is masked when the I bit in PSW is 0.

14.1.8 Unconditional Trap

An unconditional trap is generated when the INT or BRK instruction is executed.

Downloaded from Arrow.com.

14.2 Exception Handling Procedure

In the exception handling, part of the processing is handled automatically by hardware and part of it is handled by a program (exception handling routine) that has been written by the user. Figure 14.2 shows the processing procedure when an exception other than a reset is accepted.

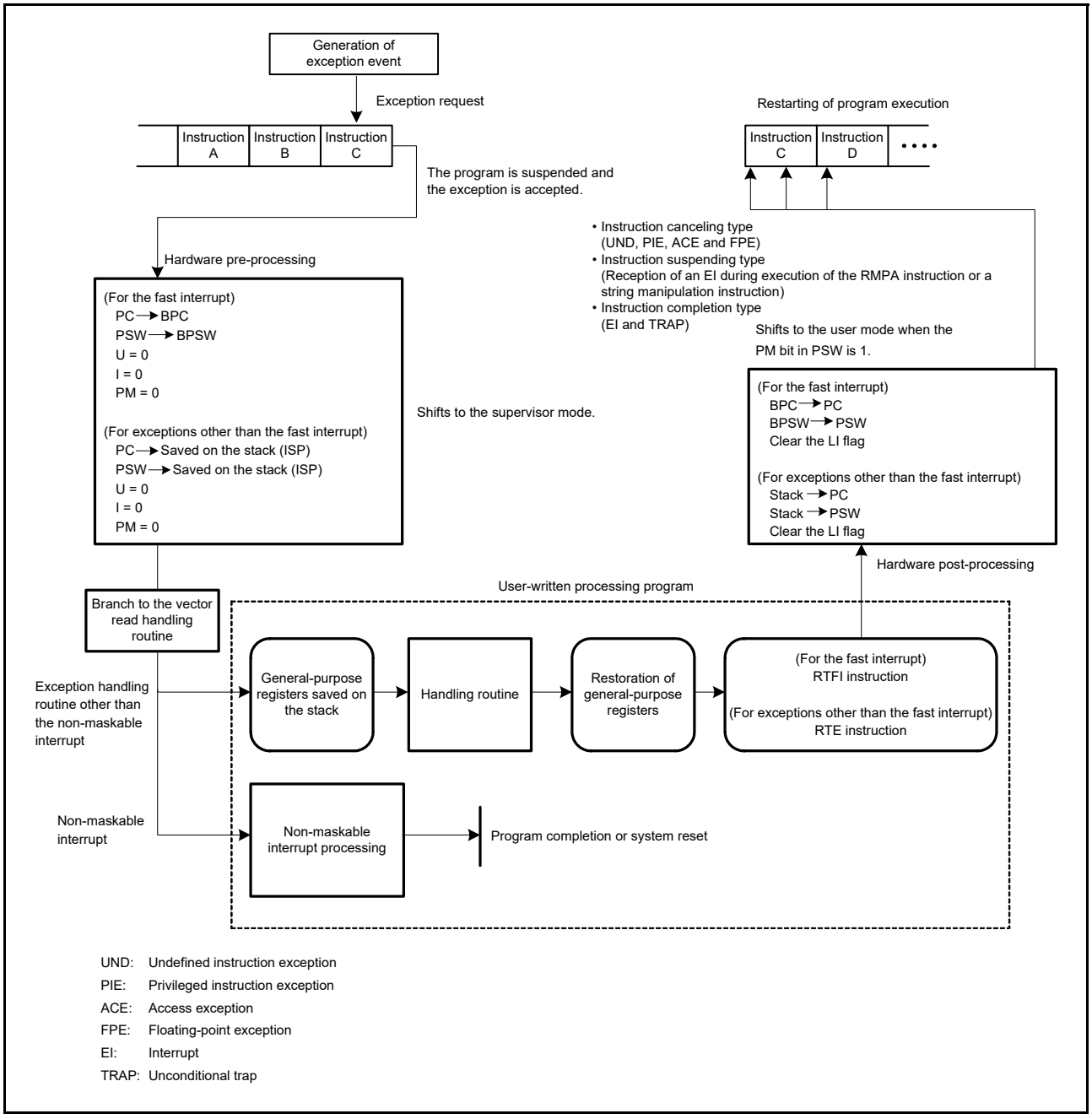


Figure 14.2 Outline of Exception Handling Procedure

Downloaded from Arrow.com.

When an exception is accepted, hardware processing by the RXv2 CPU is followed by access to the vector to acquire the address of the branch destination. In the vector, a vector address is allocated to each exception, and the branch destination address of the exception handling routine is written to each vector address.
Hardware pre-processing by the RXv2 CPU handles saving of the contents of the program counter (PC) and processor status word (PSW). In the case of a fast interrupt, the contents are saved in the backup PC (BPC) and the backup PSW (BPSW), respectively. In the case of exceptions other than a fast interrupt, the contents are saved in the stack area.
General purpose registers and control registers other than the PC and PSW that are to be used within the exception handling routine must be saved on the stack by a user program at the start of the exception handling routine.
On completion of processing by an exception handling routine, registers saved on the stack are restored and the RTE instruction is executed to restore execution from the exception handling routine to the original program. For return from a fast interrupt, the RTFI instruction is used instead. In the case of a non-maskable interrupt, however, finish the program or reset the system without returning to the original program.
Hardware post-processing by the RXv2 CPU handles restoration of the contents of PC and PSW. In the case of a fast interrupt, the values of BPC and BPSW are restored to PC and PSW, respectively. In the case of exceptions other than a fast interrupt, the values are restored from the stack to PC and PSW.

Downloaded from Arrow.com.

14.3 Acceptance of Exception Events

When an exception occurs, the CPU suspends the execution of the program and processing branches to the exception handling routine.

14.3.1 Acceptance Timing and Saved PC Value

Table 14.1 lists the timing of acceptance and the program counter (PC) value to be saved for each exception event.

Table 14.1 Acceptance Timing and Saved PC Value

Exception Event		Type of Handling	Acceptance Timing	Value Saved in BPC or on the Stack
Undefined instruction exception		Instruction canceling type	During instruction execution	PC value of the instruction that generated the exception
Privileged instruction exception		Instruction canceling type	During instruction execution	PC value of the instruction that generated the exception
Access exception		Instruction canceling type	During instruction execution	PC value of the instruction that generated the exception
Floating-point exception		Instruction canceling type	During instruction execution	PC value of the instruction that generated the exception
Reset		Instruction abandonment type	Any machine cycle	None
Non-maskable interrupt	During execution of the RMPA, SCMPU, SMOVB, SMOVF, SMOVU, SSTR, SUNTIL, and SWHILE instructions	Instruction suspending type	During instruction execution	PC value of the instruction being executed
	Other than above	Instruction completion type	At the next break between instructions	PC value of the next instruction
Interrupt	During execution of the RMPA, SCMPU, SMOVB, SMOVF, SMOVU, SSTR, SUNTIL, and SWHILE instructions	Instruction suspending type	During instruction execution	PC value of the instruction being executed
	Other than above	Instruction completion type	At the next break between instructions	PC value of the next instruction
Unconditional trap		Instruction completion type	At the next break between instructions	PC value of the next instruction

Downloaded from Arrow.com.

14.3.2 Vector and Site for Saving the Values in the PC and PSW

The vector for each type of exception and the site for saving the values of the program counter (PC) and processor status word (PSW) are listed in Table 14.2. The addresses where the exception vector table and interrupt vector table start must be set. For details, see section 2.6, Vector Table.

Table 14.2 Vector and Site for Saving the Values in the PC and PSW

Exception		Vector	Site for Saving the Values in the PC and PSW
Undefined instruction exception		Exception vector table (EXTB)	Stack
Privileged instruction exception		Exception vector table (EXTB)	Stack
Access exception		Exception vector table (EXTB)	Stack
Floating-point exception		Exception vector table (EXTB)	Stack
Reset		Exception vector table (EXTB)	Nowhere
Non-maskable interrupt		Exception vector table (EXTB)	Stack
Interrupt	Fast interrupt	FINTV	BPC and BPSW
	Other than above	Interrupt vector table (INTB)	Stack
Unconditional trap		Interrupt vector table (INTB)	Stack

Downloaded from Arrow.com.

14.4 Hardware Processing for Accepting and Returning from Exceptions

This section describes the hardware processing for accepting and returning from exceptions other than a reset.

(1) Hardware Pre-Processing for Accepting an Exception

(a) Saving PSW

- For a fast interrupt
  PSW → BPSW
- For exceptions other than a fast interrupt
  PSW → Stack

Note: The values in FPSW are not saved by hardware pre-processing. Therefore, if floating-point instructions are to be used within an exception handling routine, the user must save these values on the stack within the exception handling routine.

(b) Updating PM, U, and I Bits in PSW

I: Set to 0
U: Set to 0
PM: Set to 0

(c) Saving PC

- For a fast interrupt
  PC → BPC
- For exceptions other than a fast interrupt
  PC → Stack

(d) Setting Branch Destination Address of Exception Handling Routine in PC

Processing is shifted to the exception handling routine by acquiring the vector corresponding to the exception and then branching accordingly.

(2) Hardware Post-Processing for Execution of RTE and RTFI Instructions

(a) Restoring PSW

- For a fast interrupt
  BPSW → PSW
- For exceptions other than a fast interrupt
  Stack → PSW

(b) Restoring PC

- For a fast interrupt
  BPC → PC
- For exceptions other than a fast interrupt
  Stack → PC

(c) Clearing the LI flag

Downloaded from Arrow.com.

14.5 Hardware Pre-Processing

The hardware pre-processing from reception of each exception request to execution of the associated exception handling routine are explained below.

14.5.1 Undefined Instruction Exception

1. The value of the processor status word (PSW) is saved on the stack (ISP).
2. The processor mode select bit (PM), the stack pointer select bit (U), and the interrupt enable bit (I) in PSW are cleared to 0.
3. The value of the program counter (PC) is saved on the stack (ISP).
4. The vector is fetched from the value of EXTB + address 0000 005Ch.
5. The fetched vector is set to the PC and processing branches to the exception handling routine.

14.5.2 Privileged Instruction Exception

1. The value in the processor status word (PSW) is saved on the stack (ISP).
2. The processor mode select bit (PM), the stack pointer select bit (U), and the interrupt enable bit (I) in PSW are cleared to 0.
3. The value of the program counter (PC) is saved on the stack (ISP).
4. The vector is fetched from the value of EXTB + address 0000 0050h.
5. The fetched vector is set to the PC and processing branches to the exception handling routine.

14.5.3 Access Exceptions

1. The value in the processor status word (PSW) is saved on the stack (ISP).
2. The processor mode select bit (PM), the stack pointer select bit (U), and the interrupt enable bit (I) in PSW are cleared to 0.
3. The value of the program counter (PC) is saved on the stack (ISP).
4. The vector is fetched from the value of EXTB + address 0000 0054h.
5. The fetched vector is set to the PC and processing branches to the exception handling routine.

14.5.4 Floating-Point Exception

1. The value in the processor status word (PSW) is saved on the stack (ISP).
2. The processor mode select bit (PM), the stack pointer select bit (U), and the interrupt enable bit (I) in PSW are cleared to 0.
3. The value of the program counter (PC) is saved on the stack (ISP).
4. The vector is fetched from the value of EXTB + address 0000 0064h.
5. The fetched vector is set to the PC and processing branches to the exception handling routine.

14.5.5 Reset

1. The control registers are initialized.
2. The vector is fetched from address FFFF FFFCh.
3. The fetched vector is set to the PC.

Downloaded from Arrow.com.

14.5.6 Non-Maskable Interrupt

1. The value of the processor status word (PSW) is saved on the stack (ISP).
2. The processor mode select bit (PM), the stack pointer select bit (U), and the interrupt enable bit (I) in PSW are cleared to 0.
3. If the interrupt was generated during the execution of an RMPA, SCMPU, SMOVB, SMOVF, SMOVU, SSTR, SUNTIL, or SWHILE instruction, the value of the program counter (PC) for that instruction is saved on the stack (ISP). For other instructions, the PC value of the next instruction is saved.
4. The processor interrupt priority level bits (IPL[3:0]) in PSW are set to Fh.
5. The vector is fetched from the value of EXTB + address 0000 0078h.
6. The fetched vector is set to the PC and processing branches to the exception handling routine.

14.5.7 Interrupt

1. The value of the processor status word (PSW) is saved on the stack (ISP) or, for the fast interrupt, in the backup PSW (BPSW).
2. The processor mode select bit (PM), the stack pointer select bit (U), and the interrupt enable bit (I) in PSW are cleared to 0.
3. If the interrupt was generated during the execution of an RMPA, SCMPU, SMOVB, SMOVF, SMOVU, SSTR, SUNTIL, or SWHILE instruction, the value of the program counter (PC) for that instruction is saved. For other instructions, the PC value of the next instruction is saved. Saving of the PC is in the backup PC (BPC) for fast interrupts.
4. The processor interrupt priority level bits (IPL[3:0]) in PSW indicate the interrupt priority level of the interrupt.
5. The vector for an interrupt source other than the fast interrupt is fetched from the interrupt vector table. For the fast interrupt, the address is fetched from the fast interrupt vector register (FINTV).
6. The fetched vector is set to the PC and processing branches to the exception handling routine.

14.5.8 Unconditional Trap

1. The value in the processor status word (PSW) is saved on the stack (ISP).
2. The processor mode select bit (PM), the stack pointer select bit (U), and the interrupt enable bit (I) in PSW are cleared to 0.
3. The value of the program counter (PC) for the next instruction is saved on the stack (ISP).
4. For the INT instruction, the value at the vector corresponding to the INT instruction number is fetched from the interrupt vector table.
   For the BRK instruction, the value at the vector from the start address is fetched from the interrupt vector table.
5. The fetched vector is set to the PC and processing branches to the exception handling routine.

Downloaded from Arrow.com.

14.6 Return from Exception Handling Routine

Executing the instruction listed in Table 14.3 at the end of the corresponding exception handling routine restores the values of the program counter (PC) and processor status word (PSW) that were saved on the stack or in the control registers (BPC and BPSW) immediately before the exception handling sequence.

Table 14.3 Return from Exception Handling Routine

Exception		Instruction for Return
Undefined instruction exception		RTE
Privileged instruction exception		RTE
Access exception		RTE
Floating-point exception		RTE
Reset		Return is impossible
Non-maskable interrupt		Prohibited
Interrupt	Fast interrupt	RTFI
	Other than above	RTE
Unconditional trap		RTE

14.7 Priority of Exception Events

The priority of exception events is listed in Table 14.4. When multiple exceptions are generated at the same time, the exception with the highest priority is accepted first.

Table 14.4 Priority of Exception Events

Priority		Exception Event
High	1	Reset
↑	2	Non-maskable interrupt
	3	Interrupt
	4	Instruction access exception
	5	Undefined instruction exception Privileged instruction exception
	6	Unconditional trap
	7	Operand access exception
Low	8	Floating-point exception

Downloaded from Arrow.com.

15. Interrupt Controller (ICUb)

15.1 Overview

The interrupt controller receives interrupt requests from peripheral modules and external pins, and generates an interrupt request to the CPU and a transfer request to the DTC and DMAC.

Table 15.1 lists the specifications of the interrupt controller, and Figure 15.1 shows a block diagram of the interrupt controller.

Table 15.1 Specifications of Interrupt Controller

Item		Description
Interrupts	Peripheral function interrupts	• Interrupts from peripheral modules • Interrupt detection: Edge detection/level detection Edge detection or level detection is fixed for each source of connected peripheral modules.
	External pin interrupts	• Interrupts from pins IRQ0, IRQ1, and IRQ4 to IRQ7 • Number of sources: 6 • Interrupt detection: Low level/falling edge/rising edge/rising and falling edges One of these detection methods can be set for each source. • Digital filter function: Supported
	Software interrupt	• Interrupt generated by writing to a register • One interrupt source
	Event link interrupt	The ELSR8I, ELSR18I, or ELSR19I interrupt is generated by an ELC event
	Interrupt priority	Specified by registers.
	Fast interrupt function	Faster interrupt processing of the CPU can be set only for a single interrupt source.
	DTC/DMAC control	Interrupt sources can be used to start the DTC and DMAC.*1
Non-maskable interrupts	NMI pin interrupt	• Interrupt from the NMI pin • Interrupt detection: Falling edge/rising edge • Digital filter function: Supported
	Oscillation stop detection interrupt	Interrupt on detection of oscillation having stopped
	WDT underflow/ refresh error	Interrupt on an underflow of the down counter or occurrence of a refresh error
	IWDT underflow/ refresh error	Interrupt on an underflow of the down counter or occurrence of a refresh error
	Voltage monitoring 1 interrupt	Voltage monitoring interrupt of voltage monitoring circuit 1 (LVD1)
	VBATT voltage monitoring interrupt	Voltage monitoring interrupt of the VBATT
Return from low power consumption states	Sleep mode Deep sleep mode	Return is initiated by any non-maskable interrupt or any interrupt.
	Software standby mode	Return is initiated by Non-maskable interrupts (excluding oscillation stop detection interrupt), external pin interrupts (IRQ0, IRQ1, and IRQ4 to IRQ7), peripheral interrupts (voltage monitoring 1, VBATT voltage monitoring, RTC alarm/periodic, USB0 resume), or ELSR8I interrupt (LPT dedicated interrupt).

Note 1. For the DTC and DMAC triggers, refer to Table 15.3, Interrupt Vector Table.

Downloaded from Arrow.com.

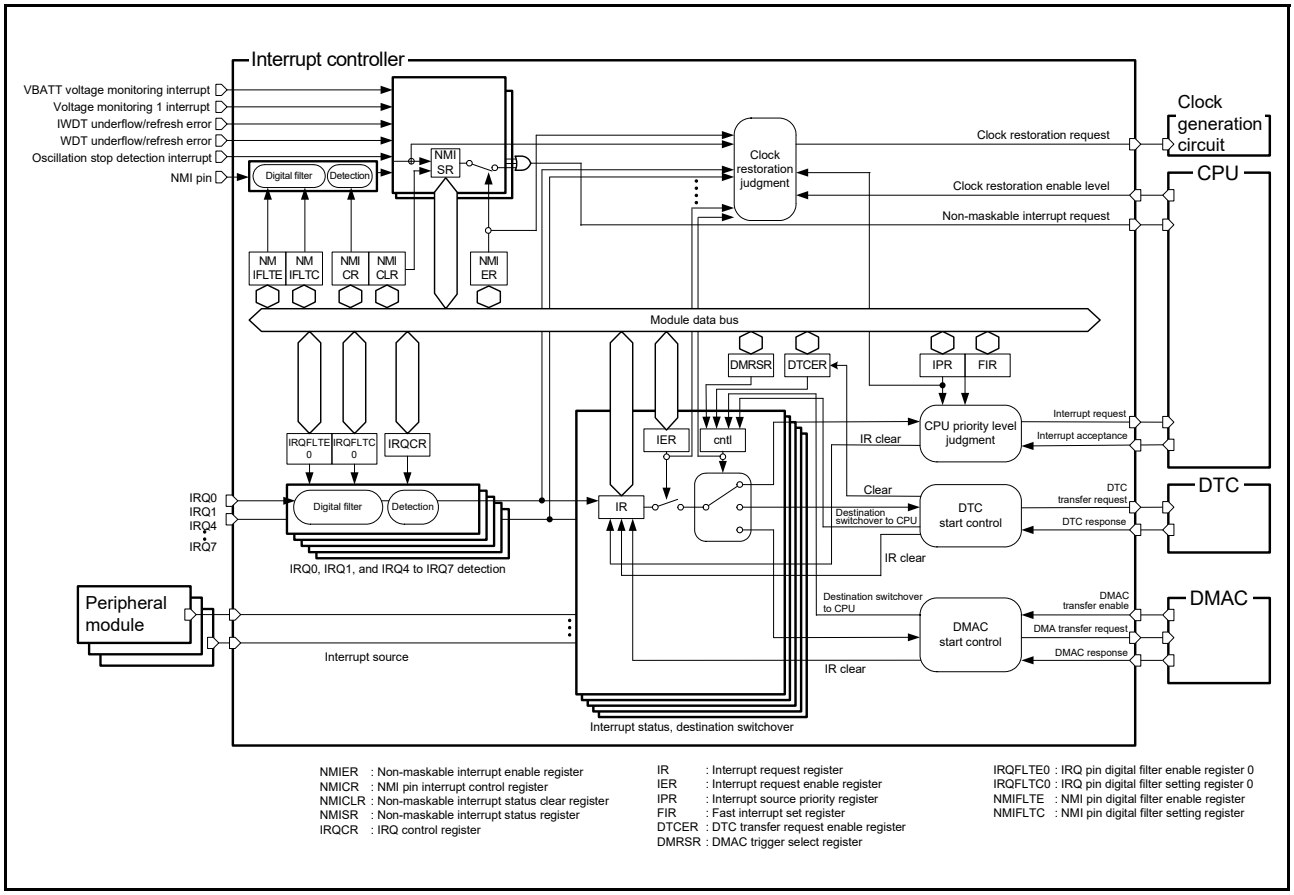


Figure 15.1 Block Diagram of Interrupt Controller

Table 15.2 lists the input/output pins of the interrupt controller.

Table 15.2 Pin Configuration of Interrupt Controller

Pin Name	I/O	Description
NMI	Input	Non-maskable interrupt request pin
IRQ0, IRQ1, and IRQ4 to IRQ7	Input	External interrupt request pins

Downloaded from Arrow.com.

15.2 Register Descriptions

15.2.1 Interrupt Request Register n (IRn) (n = interrupt vector number)

Address(es): ICU.IR016 0008 7010h to ICU.IR255 0008 70FFh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	IR
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	IR	Interrupt Status Flag	0: No interrupt request is generated 1: An interrupt request is generated	R/(W) *1
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. For an edge detection interrupt, only 0 can be written to this bit; do not write 1.
For a level detection interrupt, neither 0 nor 1 can be written.

IRn is provided for each interrupt source, where “n” indicates the interrupt vector number.
For the correspondence between interrupt sources and interrupt vector numbers, see Table 15.3, Interrupt Vector Table.

IR Flag (Interrupt Status Flag)

This flag is the status flag of an individual interrupt request. This flag is set to 1 when the corresponding interrupt request is generated. To detect an interrupt request, the interrupt request output should be enabled by the corresponding peripheral module interrupt enable bit.
There are two interrupt request detection methods: edge detection and level detection. For interrupts from peripheral modules, either edge detection or level detection is determined per interrupt source. For interrupts from IRQi (i = 0, 1, and 4 to 7) pins, edge detection or level detection is selected by setting the corresponding IRQCRi.IRQMD[1:0] bits. For detection of the various interrupt sources, see Table 15.3, Interrupt Vector Table.

(1) Edge detection

[Setting condition]

- The flag is set to 1 in response to the generation of an interrupt request from the corresponding peripheral module or IRQi pin. For interrupt generation by the various peripheral modules, refer to the sections describing the modules.

[Clearing conditions]

- The flag is cleared to 0 when the interrupt request destination accepts the interrupt request.
- The IR flag is cleared to 0 by writing 0 to it. Note, however, that writing 0 to the IR flag is prohibited if the destination of the interrupt request is the DTC or DMAC.

(2) Level detection

[Setting condition]

- The flag remains set to 1 while an interrupt request is being sent from the corresponding peripheral module or IRQi pin. For interrupt generation by the various peripheral modules, refer to the sections describing the modules.

[Clearing condition]

- The flag is cleared to 0 when the source of the interrupt request is cleared (it is not cleared when the interrupt request destination accepts the interrupt request). For clearing interrupts from the various peripheral modules, refer to the sections describing the modules.

When level detection has been selected for an IRQi pin, the interrupt request is withdrawn by driving the IRQi pin high.
Do not write 0 or 1 to the IR flag while level detection is selected.

Downloaded from Arrow.com.

15.2.2 Interrupt Request Enable Register m (IERm) (m = 02h to 1Fh)

Address(es): ICU.IER02 0008 7202h to ICU.IER1F 0008 721Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	IEN7	IEN6	IEN5	IEN4	IEN3	IEN2	IEN1	IEN0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	IEN0	Interrupt Request Enable 0	0: Interrupt request is disabled 1: Interrupt request is enabled	R/W
b1	IEN1	Interrupt Request Enable 1		R/W
b2	IEN2	Interrupt Request Enable 2		R/W
b3	IEN3	Interrupt Request Enable 3		R/W
b4	IEN4	Interrupt Request Enable 4		R/W
b5	IEN5	Interrupt Request Enable 5		R/W
b6	IEN6	Interrupt Request Enable 6		R/W
b7	IEN7	Interrupt Request Enable 7		R/W

Note: Write 0 to the bit that corresponds to the vector number for reservation. These bits are read as 0.

IENj Bit (Interrupt Request Enable j) (j = 0 to 7)

When an IENj bit is 1, the corresponding interrupt request will be output to the destination selected for the request.
When an IENj bit is 0, the corresponding interrupt request will not be output to the destination selected for the request.
The setting of an IENj bit does not affect the IRn.IR flag (n = interrupt vector number). Even if the corresponding IENj bit is 0, the IR flag value changes according to the descriptions in section 15.2.1, Interrupt Request Register n (IRn) (n = interrupt vector number).
The IERm.IENj bit is set for each request source (vector number).
For the correspondence between interrupt sources and IERm.IENj bits, see Table 15.3, Interrupt Vector Table.
For the procedure for setting IERm.IENj bits during the selection of destinations for interrupt requests, refer to section 15.4.3, Selecting Interrupt Request Destinations.

Downloaded from Arrow.com.

15.2.3 Interrupt Source Priority Register n (IPRn) (n = interrupt vector number)

Address(es): ICU.IPR000 0008 7300h to ICU.IPR255 0008 73FFh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	IPR[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	IPR[3:0]	Interrupt Priority Level Select	b3 b0 0 0 0 0: Level 0 (interrupt disabled)*1 0 0 0 1: Level 1 0 0 1 0: Level 2 0 0 1 1: Level 3 0 1 0 0: Level 4 0 1 0 1: Level 5 0 1 1 0: Level 6 0 1 1 1: Level 7 1 0 0 0: Level 8 1 0 0 1: Level 9 1 0 1 0: Level 10 1 0 1 1: Level 11 1 1 0 0: Level 12 1 1 0 1: Level 13 1 1 1 0: Level 14 1 1 1 1: Level 15 (highest)	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. When the interrupt is specified as a fast interrupt, it can be issued even if the priority level is level 0.

For the correspondence between interrupt sources and IPRn registers, see Table 15.3, Interrupt Vector Table.

IPR[3:0] Bits (Interrupt Priority Level Select)

These bits specify the priority level of the corresponding interrupt source.

Priority levels specified by the IPR[3:0] bits are used only to determine the priority of interrupt requests to be transferred to the CPU, and do not affect transfer requests to the DTC or DMAC.

The CPU accepts only interrupt requests higher than the priority level specified by the IPL[3:0] bits in PSW, and handles accepted interrupts.

If two or more interrupt requests are generated at the same time, their priority levels are compared with the value of the IPR[3:0] bits. If interrupt requests of the same priority level are generated at the same time, an interrupt source with a smaller vector number takes precedence.

These bits should be written to while an interrupt request is disabled (IERm.IENj bit = 0 (m = 02h to 1Fh; j = 0 to 7)).

Downloaded from Arrow.com.

15.2.4 Fast Interrupt Set Register (FIR)

Address(es): ICU.FIR 0008 72F0h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	FIEN	—	—	—	—	—	—	—	FVCT[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	FVCT[7:0]	Fast Interrupt Vector Number	Specify the vector number of an interrupt source to be a fast interrupt.	R/W
b14 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	FIEN	Fast Interrupt Enable	0: Fast interrupt is disabled 1: Fast interrupt is enabled	R/W

The fast interrupt function based on the FIR register setting is applicable only to interrupts to the CPU. It will not affect any transfer request to the DTC or DMAC.
Before writing to this register, be sure to disable interrupt requests (IERm.IENj bit = 0 (m = 02h to 1Fh; j = 0 to 7)).

FVCT[7:0] Bits (Fast Interrupt Vector Number)

The FVCT[7:0] bits specify the vector number of an interrupt source that uses the fast interrupt function.

FIEN Bit (Fast Interrupt Enable)

This bit enables the fast interrupt.
Setting this bit to 1 makes the interrupt request of the vector number specified by the FVCT[7:0] bits a fast interrupt.
When an interrupt request of the vector number specified by the FVCT[7:0] bits is generated and the interrupt request destination is the CPU while the FIEN bit is 1, the interrupt request is output to the CPU as a fast interrupt regardless of the setting of the IPRn register (n = interrupt vector number). When using the fast interrupt for returning from the software standby mode, see section 15.6.2, Return from Software Standby Mode.
If the setting of the IERm.IENj bit has disabled interrupt requests from the interrupt source with the vector number in this register, fast interrupt requests are not output to the CPU.
For settable vector numbers, see Table 15.3, Interrupt Vector Table.
Do not write any reserved vector numbers to the FVCT[7:0] bits.

For details on the fast interrupt, see section 14, Exception Handling, and section 15.4.6, Fast Interrupt.

Downloaded from Arrow.com.

15.2.5 Software Interrupt Generation Register (SWINTR)

Address(es): ICU.SWINTR 0008 72E0h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	SWINT
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SWINT	Software Interrupt Generation	This bit is read as 0. Writing 1 issues a software interrupt request. Writing 0 to this bit has no effect.	R/(W) *1
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Only 1 can be written.

SWINT Bit (Software Interrupt Generation)

When 1 is written to the SWINT bit, the interrupt request register 027 (IR027) is set to 1.

If 1 is written to the SWINT bit when the DTC transfer request enable register 027 (DTCER027) is set to 0, an interrupt to the CPU is generated.

If 1 is written to the SWINT bit when the DTC transfer request enable register 027 (DTCER027) is set to 1, a DTC transfer request is issued.

Downloaded from Arrow.com.

15.2.6 DTC Transfer Request Enable Register n (DTCERn) (n = interrupt vector number)

Address(es): ICU.DTCER027 0008 711Bh to ICU.DTCER255 0008 71FFh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	DTCE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DTCE	DTC Transfer Request Enable	0: The corresponding interrupt source is not selected as the DTC trigger. 1: The corresponding interrupt source is selected as the DTC trigger.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

An interrupt source that has been selected as a DMAC trigger should not be specified as a DTC trigger. See Table 15.3, Interrupt Vector Table, for the interrupt sources that are selectable as the DTC trigger.

DTCE Bit (DTC Transfer Request Enable)

When the DTCE bit is set to 1, the corresponding interrupt source is selected as the DTC trigger.

[Setting condition]

- When 1 is written to the DTCE bit

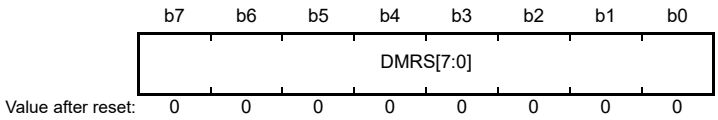
[Clearing conditions]

- When the specified number of transfers is completed (for the chain transfer, the number of transfers for the last chain transfer is completed)
- When 0 is written to the DTCE bit

Downloaded from Arrow.com.

15.2.7 DMAC Trigger Select Register m (DMRSRm) (m = DMAC channel number)

Address(es): ICU.DMRSR0 0008 7400h, ICU.DMRSR1 0008 7404h, ICU.DMRSR2 0008 7408h, ICU.DMRSR3 0008 740Ch



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	DMRS[7:0]	DMAC Trigger Select	These bits specify the vector number for the DMA transfer request.	R/W

To specify the same interrupt source for multiple DMRSRm registers is disabled. The interrupt source that has been selected for the DMRSRm trigger should not be specified as the DTC trigger. Otherwise, the correct operation is not guaranteed.

DMRS[7:0] Bits (DMAC Trigger Select)

The vector number of the interrupt source used as the DMAC trigger is specified in 8 bits. Do not set the vector numbers that are not assigned for the DMAC trigger.

For the correspondence between interrupt sources and interrupt vector numbers, see Table 15.3, Interrupt Vector Table.

Write to the DMRSRm register while the DMA transfer enable bit of the DMA transfer enable register (DMACm.DMCNT.DTE) is cleared to 0.

Downloaded from Arrow.com.

15.2.8 IRQ Control Register i (IRQCRi) (i = 0, 1, and 4 to 7)

Address(es): ICU.IRQCR0 0008 7500h, ICU.IRQCR1 0008 7501h, ICU.IRQCR4 0008 7504h to ICU.IRQCR7 0008 7507h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	IRQMD[1:0]		—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b3, b2	IRQMD[1:0]	IRQ Detection Sense Select	b3 b2 0 0: Low level 0 1: Falling edge 1 0: Rising edge 1 1: Rising and falling edges	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Only change the settings of this register while the corresponding interrupt request enable bit is prohibiting the interrupt request (IENj bit in IERm (m = 02h to 1Fh; j = 0 to 7) is 0). After changing the setting, clear the IR flag in IRn before setting the interrupt enable bit. However, when the change is to the low level, the IR flag does not require clearing.

IRQMD[1:0] Bits (IRQ Detection Sense Select)

These bits select the interrupt detection sensing method of IRQi pin.

For the external pin interrupt detection setting, see section 15.4.8, External Pin Interrupts.

Downloaded from Arrow.com.

15.2.9 IRQ Pin Digital Filter Enable Register 0 (IRQFLTE0)

Address(es): ICU.IRQFLTE0 0008 7510h

	b7	b6	b5	b4	b3	b2	b1	b0
	FLTEN7	FLTEN6	FLTEN5	FLTEN4	—	—	FLTEN1	FLTEN0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	FLTEN0	IRQ0 Digital Filter Enable	0: Digital filter is disabled 1: Digital filter is enabled	R/W
b1	FLTEN1	IRQ1 Digital Filter Enable		R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	FLTEN4	IRQ4 Digital Filter Enable	0: Digital filter is disabled 1: Digital filter is enabled	R/W
b5	FLTEN5	IRQ5 Digital Filter Enable		R/W
b6	FLTEN6	IRQ6 Digital Filter Enable		R/W
b7	FLTEN7	IRQ7 Digital Filter Enable		R/W

FLTENi Bit (IRQi Digital Filter Enable) (i = 0, 1, and 4 to 7)

This bit enables the digital filter used for the IRQi pin.

The digital filter is enabled when the FLTENi bit is 1, and disabled when the FLTENi bit is 0.

The IRQi pin level is sampled at the sampling clock cycle specified with the IRQFLTC0.FCLKSELi[1:0] bits. When the sampled level matches three times, the output level from the digital filter changes.

For details of the digital filter, see section 15.4.7, Digital Filter.

Downloaded from Arrow.com.

15.2.10 IRQ Pin Digital Filter Setting Register 0 (IRQFLTC0)

Address(es): ICU.IRQFLTC0 0008 7514h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	FCLKSEL7[1:0]		FCLKSEL6[1:0]		FCLKSEL5[1:0]		FCLKSEL4[1:0]		—		—		FCLKSEL1[1:0]		FCLKSEL0[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	FCLKSEL0[1:0]	IRQ0 Digital Filter Sampling Clock	0 0: PCLK 0 1: PCLK/8 1 0: PCLK/32 1 1: PCLK/64	R/W
b3, b2	FCLKSEL1[1:0]	IRQ1 Digital Filter Sampling Clock		R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b9, b8	FCLKSEL4[1:0]	IRQ4 Digital Filter Sampling Clock	0 0: PCLK 0 1: PCLK/8 1 0: PCLK/32 1 1: PCLK/64	R/W
b11, b10	FCLKSEL5[1:0]	IRQ5 Digital Filter Sampling Clock		R/W
b13, b12	FCLKSEL6[1:0]	IRQ6 Digital Filter Sampling Clock		R/W
b15, b14	FCLKSEL7[1:0]	IRQ7 Digital Filter Sampling Clock		R/W

FCLKSELi[1:0] Bits (IRQi Digital Filter Sampling Clock) (i = 0, 1, and 4 to 7)

These bits select the cycle of the digital filter sampling clock for the IRQi pin.

The sampling clock cycle can be selected from among the PCLK (every cycle), PCLK/8 (once every eight cycles), PCLK/32 (once every 32 cycles), and PCLK/64 (once every 64 cycles).

For details of the digital filter, see section 15.4.7, Digital Filter.

Downloaded from Arrow.com.

15.2.11 Non-Maskable Interrupt Status Register (NMISR)

Address(es): ICU.NMISR 0008 7580h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	VBATST	—	LVD1ST	IWDTST	WDTST	OSTST	NMIST
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	NMIST	NMI Status Flag	0: NMI pin interrupt is not requested 1: NMI pin interrupt is requested	R
b1	OSTST	Oscillation Stop Detection Interrupt Status Flag	0: Oscillation stop detection interrupt is not requested 1: Oscillation stop detection interrupt is requested	R
b2	WDTST	WDT Underflow/Refresh Error Status Flag	0: WDT underflow/refresh error interrupt is not requested 1: WDT underflow/refresh error interrupt is requested	R
b3	IWDTST	IWDT Underflow/Refresh Error Status Flag	0: IWDT underflow/refresh error interrupt is not requested 1: IWDT underflow/refresh error interrupt is requested	R
b4	LVD1ST	Voltage Monitoring 1 Interrupt Status Flag	0: Voltage monitoring 1 interrupt is not requested 1: Voltage monitoring 1 interrupt is requested	R
b5	—	Reserved	This bit is read as 0. Writing to this bit has no effect.	R
b6	VBATST	VBATT Voltage Monitoring Interrupt Status Flag	0: VBATT voltage monitoring interrupt request is not requested 1: VBATT voltage monitoring interrupt request is requested	R
b7	—	Reserved	This bit is read as 0. Writing to this bit has no effect.	R

The NMISR register monitors the status of a non-maskable interrupt source. Writing to the NMISR register is ignored. The setting in the non-maskable interrupt enable register (NMIER) does not affect the status flags in NMISR.
Before the end of the non-maskable interrupt handler, read the NMISR register and confirm the generation status of other non-maskable interrupts. Be sure to confirm that all of the bits in the NMISR register are set to 0 before the end of the handler.

NMIST Flag (NMI Status Flag)

This flag indicates the NMI pin interrupt request.
The NMIST flag is read-only, and cleared by the NMICLR.NMICLR bit.
[Setting condition]

- When an edge specified by the NMICR.NMIMD bit is input to the NMI pin

[Clearing condition]

- When 1 is written to the NMICLR.NMICLR bit

OSTST Flag (Oscillation Stop Detection Interrupt Status Flag)

This flag indicates the oscillation stop detection interrupt request.
The OSTST flag is read-only, and cleared by the NMICLR.OSTCLR bit.
[Setting condition]

- When the oscillation stop detection interrupt is generated

[Clearing condition]

- When 1 is written to the NMICLR.OSTCLR bit

Downloaded from Arrow.com.

WDTST Flag (WDT Underflow/Refresh Error Status Flag)

This flag indicates the WDT underflow/refresh error interrupt request.
The WDTST flag is read-only, and cleared by the NMICLR.WDTCLR bit.
[Setting condition]

- When the WDT underflow/refresh error interrupt is generated

[Clearing condition]

- When 1 is written to the NMICLR.WDTCLR bit

IWDTST Flag (IWDT Underflow/Refresh Error Status Flag)

This flag indicates the IWDT underflow/refresh error interrupt request.
The IWDTST flag is read-only, and cleared by the NMICLR.IWDTCLR bit.
[Setting condition]

- When the IWDT underflow/refresh error interrupt is generated while this interrupt is enabled at its source.

[Clearing condition]

- When 1 is written to the NMICLR.IWDTCLR bit

LVD1ST Flag (Voltage Monitoring 1 Interrupt Status Flag)

This flag indicates the request for voltage monitoring 1 interrupt.
The LVD1ST flag is read-only, and cleared by the NMICLR.LVD1CLR bit.
[Setting condition]

- When the voltage monitoring 1 interrupt is generated while this interrupt is enabled at its source.

[Clearing condition]

- When 1 is written to the NMICLR.LVD1CLR bit

VBATST Flag (VBATT Voltage Monitoring Interrupt Status Flag)

This flag indicates the request for the VBATT voltage monitoring interrupt.
[Setting condition]

- When the VBATT voltage monitoring interrupt is generated while this interrupt is enabled at its source.

[Clearing condition]

- When 1 is written to the NMICLR.VBATCLR bit.

Downloaded from Arrow.com.

15.2.12 Non-Maskable Interrupt Enable Register (NMIER)

Address(es): ICU.NMIER 0008 7581h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	VBATEN	—	LVD1EN	IWDTEN	WDTEN	OSTEN	NMIEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	NMIEN	NMI Pin Interrupt Enable	0: NMI pin interrupt is disabled 1: NMI pin interrupt is enabled	R/(W) *1
b1	OSTEN	Oscillation Stop Detection Interrupt Enable	0: Oscillation stop detection interrupt is disabled 1: Oscillation stop detection interrupt is enabled	R/(W) *1
b2	WDTEN	WDT Underflow/Refresh Error Enable	0: WDT underflow/refresh error interrupt is disabled 1: WDT underflow/refresh error interrupt is enabled	R/(W) *1
b3	IWDTEN	IWDT Underflow/Refresh Error Enable	0: IWDT underflow/refresh error interrupt is disabled 1: IWDT underflow/refresh error interrupt is enabled	R/(W) *1
b4	LVD1EN	Voltage Monitoring 1 Interrupt Enable	0: Voltage monitoring 1 interrupt is disabled 1: Voltage monitoring 1 interrupt is enabled	R/(W) *1
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	VBATEN	VBATT Voltage Monitoring Interrupt Enable	0: VBATT voltage monitoring interrupt is disabled 1: VBATT voltage monitoring interrupt is enabled	R/W *1
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note 1. A 1 can be written to this bit only once, and subsequent write accesses are no longer enabled.

NMIEN Bit (NMI Pin Interrupt Enable)

This bit enables the NMI pin interrupt.
A 1 can be written to this bit only once, and subsequent write accesses are no longer enabled.
Writing 0 to this bit is disabled.

OSTEN Bit (Oscillation Stop Detection Interrupt Enable)

This bit enables the oscillation stop detection interrupt.
A 1 can be written to this bit only once, and subsequent write accesses are no longer enabled.
Writing 0 to this bit is disabled.

WDTEN Bit (WDT Underflow/Refresh Error Enable)

This bit enables the WDT underflow/refresh error interrupt.
A 1 can be written to this bit only once, and subsequent write accesses are no longer enabled.
Writing 0 to this bit is disabled.

IWDTEN Bit (IWDT Underflow/Refresh Error Enable)

This bit enables the IWDT underflow/refresh error interrupt.
A 1 can be written to this bit only once, and subsequent write accesses are no longer enabled.
Writing 0 to this bit is disabled.

LVD1EN Bit (Voltage Monitoring 1 Interrupt Enable)

This bit enables the voltage monitoring 1 interrupt.
A 1 can be written to this bit only once, and subsequent write accesses are no longer enabled.
Writing 0 to this bit is disabled.

Downloaded from Arrow.com.

VBATEN Bit (VBATT Voltage Monitoring Interrupt Enable)

This bit enables the VBATT voltage monitoring interrupt.

A 1 can be written to this bit only once, and subsequent write accesses are no longer enabled.

Writing 0 to this bit is disabled.

Downloaded from Arrow.com.

15.2.13 Non-Maskable Interrupt Status Clear Register (NMICLR)

Address(es): ICU.NMICLR 0008 7582h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	VBATCLR	—	LVD1CLR	IWDTCLR	WDTCLR	OSTCLR	NMICLR
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	NMICLR	NMI Clear	This bit is read as 0. Writing 1 to this bit clears the NMISR.NMIST flag. Writing 0 to this bit has no effect.	R/(W) *1
b1	OSTCLR	OST Clear	This bit is read as 0. Writing 1 to this bit clears the NMISR.OSTST flag. Writing 0 to this bit has no effect.	R/(W) *1
b2	WDTCLR	WDT Clear	This bit is read as 0. Writing 1 to this bit clears the NMISR.WDTST flag. Writing 0 to this bit has no effect.	R/(W) *1
b3	IWDTCLR	IWDT Clear	This bit is read as 0. Writing 1 to this bit clears the NMISR.IWDTST flag. Writing 0 to this bit has no effect.	R/(W) *1
b4	LVD1CLR	LVD1 Clear	This bit is read as 0. Writing 1 to this bit clears the NMISR.LVD1ST flag. Writing 0 to this bit has no effect.	R/(W) *1
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	VBATCLR	VBAT Clear	This bit is read as 0. Writing 1 to this bit clears the NMISR.VBATST flag. Writing 0 has no effect.	R/(W) *1
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note 1. Only 1 can be written to this bit.

NMICLR Bit (NMI Clear)

Writing 1 to the NMICLR bit clears the NMISR.NMIST flag. This bit is read as 0.

OSTCLR Bit (OST Clear)

Writing 1 to the OSTCLR bit clears the NMISR.OSTST flag. This bit is read as 0.

WDTCLR Bit (WDT Clear)

Writing 1 to the WDTCLR bit clears the NMISR.WDTST flag. This bit is read as 0.

IWDTCLR Bit (IWDT Clear)

Writing 1 to the IWDTCLR bit clears the NMISR.IWDTST flag. This bit is read as 0.

LVD1CLR Bit (LVD1 Clear)

Writing 1 to the LVD1CLR bit clears the NMISR.LVD1ST flag. This bit is read as 0.

VBATCLR Bit (VBAT Clear)

Writing 1 to the VBATCLR bit clears the NMISR.VBATST flag. This bit is read as 0.

Downloaded from Arrow.com.

15.2.14 NMI Pin Interrupt Control Register (NMICR)

Address(es): ICU.NMICR 0008 7583h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	NMIMD	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b3	NMIMD	NMI Detection Set	0: Falling edge 1: Rising edge	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Change the setting of the NMICR register before the NMI pin interrupt is enabled (before setting the NMIER.NMIEN bit to 1).

NMIMD Bit (NMI Detection Set)

This bit specifies the detection edge of the NMI pin interrupt.

15.2.15 NMI Pin Digital Filter Enable Register (NMIFLTE)

Address(es): ICU.NMIFLTE 0008 7590h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	NFLTEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	NFLTEN	NMI Digital Filter Enable	0: Digital filter is disabled 1: Digital filter is enabled	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

NFLTEN Bit (NMI Digital Filter Enable)

This bit enables the digital filter used for the NMI pin interrupt.

The digital filter is enabled when the NFLTEN bit is 1, and disabled when the NFLTEN bit is 0.

The NMI pin level is sampled at the sampling clock cycle specified with the NMIFLTC.NFCLKSEL[1:0] bits. When the sampled level matches three times, the output level from the digital filter changes.

For details of the digital filter, see section 15.4.7, Digital Filter.

Downloaded from Arrow.com.

15.2.16 NMI Pin Digital Filter Setting Register (NMIFLTC)

Address(es): ICU.NMIFLTC 0008 7594h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	NFCLKSEL[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	NFCLKSEL[1:0]	NMI Digital Filter Sampling Clock	b1 b0 0 0: PCLK 0 1: PCLK/8 1 0: PCLK/32 1 1: PCLK/64	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

NFCLKSEL[1:0] Bits (NMI Digital Filter Sampling Clock)

These bits select the cycle of the digital filter sampling clock for the NMI pin interrupt.

The sampling clock cycle can be selected from among the PCLK (every cycle), PCLK/8 (once every eight cycles), PCLK/32 (once every 32 cycles), and PCLK/64 (once every 64 cycles).

For details of the digital filter, see section 15.4.7, Digital Filter.

Downloaded from Arrow.com.

15.3 Vector Table

There are two types of interrupts detected by the interrupt controller: maskable interrupts and non-maskable interrupts. When the CPU accepts an interrupt or non-maskable interrupt, it acquires a 4-byte vector address from the vector table.

15.3.1 Interrupt Vector Table

The interrupt vector table is placed in the 1024-byte range (4 bytes × 256 sources) beginning at the address specified in the interrupt table register (INTB) of the CPU. Write a value to the INTB register before enabling interrupts. The value written to the INTB register should be a multiple of 4.
Executing an INT instruction or BRK instruction leads to the generation of an unconditional trap. The same range of memory as shown in Table 15.3, Interrupt Vector Table, is used for the vectors for unconditional traps. The vector for BRK instructions is vector 0 while the vector numbers for INT instructions are specifiable as numbers in the range from 0 to 255.
Table 15.3 lists details of the interrupt vectors. Details of the headings in Table 15.3 are listed below.

Item	Description
Source of interrupt request generation	Name of the source for generation of the interrupt request
Name	Name of the interrupt
Vector no.	Vector number for the interrupt
Vector address offset	Value of the offset from the base address for the vector table
Interrupt detection method	“Edge” or “level” as the method for detection of the interrupt
CPU interrupt	“✓” in this column indicates usability as a CPU interrupt.
Start DTC	“✓” in this column indicates usability as a request for DTC transfer.
Start DMAC	“✓” in this column indicates usability as a request for DMA transfer.
SSBY return	“✓” in this column indicates usability as a request for return from software-standby mode.
IER	Name of the IER register and bit corresponding to the vector number
IPR	Name of the IPR register corresponding to the interrupt source
DTCER	Name of the DTCER register corresponding to the DTC trigger

Downloaded from Arrow.com.

Table 15.3 Interrupt Vector Table (1/6)

Source of Interrupt Request Generation	Name	Vector No.*1	Vector Address Offset	Interrupt Detection Method	CPU Interrupt	Start DTC	Start DMAC	SSBY Return	IER	IPR	DTCER
—	For an unconditional trap	0	0000h	—	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	1	0004h	—	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	2	0008h	—	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	3	000Ch	—	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	4	0010h	—	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	5	0014h	—	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	6	0018h	—	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	7	001Ch	—	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	8	0020h	—	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	9	0024h	—	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	10	0028h	—	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	11	002Ch	—	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	12	0030h	—	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	13	0034h	—	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	14	0038h	—	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	15	003Ch	—	N/A	N/A	N/A	N/A	—	—	—
BSC	BUSERR	16	0040h	Level	✓	N/A	N/A	N/A	IER02.IEN0	IPR000	—
—	Reserved	17	0044h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	18	0048h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	19	004Ch	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	20	0050h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	21	0054h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	22	0058h	—	N/A	N/A	N/A	N/A	—	—	—
FCU	FRDYI	23	005Ch	Edge	✓	N/A	N/A	N/A	IER02.IEN7	IPR002	—
—	Reserved	24	0060h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	25	0064h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	26	0068h	—	N/A	N/A	N/A	N/A	—	—	—
ICU	SWINT	27	006Ch	Edge	✓	✓	N/A	N/A	IER03.IEN3	IPR003	DTCER027
CMT0	CMI0	28	0070h	Edge	✓	✓	✓	N/A	IER03.IEN4	IPR004	DTCER028
CMT1	CMI1	29	0074h	Edge	✓	✓	✓	N/A	IER03.IEN5	IPR005	DTCER029
CMT2	CMI2	30	0078h	Edge	✓	✓	✓	N/A	IER03.IEN6	IPR006	DTCER030
CMT3	CMI3	31	007Ch	Edge	✓	✓	✓	N/A	IER03.IEN7	IPR007	DTCER031
CAC	FERRF	32	0080h	Level	✓	N/A	N/A	N/A	IER04.IEN0	IPR032	—
	MENDF	33	0084h	Level	✓	N/A	N/A	N/A	IER04.IEN1	IPR033	—
	OVFF	34	0088h	Level	✓	N/A	N/A	N/A	IER04.IEN2	IPR034	—
—	Reserved	35	008Ch	—	N/A	N/A	N/A	N/A	—	—	—
USB0	D0FIFO0	36	0090h	Edge	✓	✓	✓	N/A	IER04.IEN4	IPR036	DTCER036
	D1FIFO0	37	0094h	Edge	✓	✓	✓	N/A	IER04.IEN5	IPR037	DTCER037
	USBI0	38	0098h	Edge	✓	N/A	N/A	N/A	IER04.IEN6	IPR038	—
—	Reserved	39	009Ch	—	N/A	N/A	N/A	N/A	—	—	—
SDHI	SBFAI	40	00A0h	Edge	✓	✓	✓	N/A	IER05.IEN0	IPR040	DTCER040
	CDETI	41	00A4h	Level	✓	N/A	N/A	N/A	IER05.IEN1	IPR041	—
	CACI	42	00A8h	Level	✓	N/A	N/A	N/A	IER05.IEN2	IPR042	—
	SDACI	43	00ACh	Level	✓	N/A	N/A	N/A	IER05.IEN3	IPR043	—

Downloaded from Arrow.com.

Table 15.3 Interrupt Vector Table (2/6)

Source of Interrupt Request Generation	Name	Vector No.*1	Vector Address Offset	Interrupt Detection Method	CPU Interrupt	Start DTC	Start DMAC	SSBY Return	IER	IPR	DTCER
RSPI0	SPEI0	44	00B0h	Level	✓	N/A	N/A	N/A	IER05.IEN4	IPR044	—
	SPRI0	45	00B4h	Edge	✓	✓	✓	N/A	IER05.IEN5		DTCER045
	SPTI0	46	00B8h	Edge	✓	✓	✓	N/A	IER05.IEN6		DTCER046
	SPII0	47	00BCh	Level	✓	N/A	N/A	N/A	IER05.IEN7		—
—	Reserved	48	00C0h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	49	00C4h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	50	00C8h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	51	00CCh	—	N/A	N/A	N/A	N/A	—	—	—
RSCAN	COMFRXINT	52	00D0h	Edge	✓	✓	✓	N/A	IER06.IEN4	IPR052	DTCER052
	RXFINT	53	00D4h	Level	✓	N/A	N/A	N/A	IER06.IEN5	IPR053	—
	TXINT	54	00D8h	Level	✓	N/A	N/A	N/A	IER06.IEN6	IPR054	—
	CHERRINT	55	00DCh	Level	✓	N/A	N/A	N/A	IER06.IEN7	IPR055	—
	GLERRINT	56	00E0h	Level	✓	N/A	N/A	N/A	IER07.IEN0	IPR056	—
DOC	DOPCF	57	00E4h	Level	✓	N/A	N/A	N/A	IER07.IEN1	IPR057	—
—	Reserved	58	00E8h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	59	00ECh	—	N/A	N/A	N/A	N/A	—	—	—
CTSU	CTSUWR	60	00F0h	Edge	✓	✓	✓	N/A	IER07.IEN4	IPR060	DTCER060
	CTSURD	61	00F4h	Edge	✓	✓	✓	N/A	IER07.IEN5		DTCER061
	CTSUFN	62	00F8h	Edge	✓	N/A	N/A	N/A	IER07.IEN6		—
RTC	CUP	63	00FCh	Edge	✓	N/A	N/A	N/A	IER07.IEN7	IPR063	—
ICU	IRQ0	64	0100h	Edge/Level	✓	✓	✓	✓	IER08.IEN0	IPR064	DTCER064
	IRQ1	65	0104h	Edge/Level	✓	✓	✓	✓	IER08.IEN1	IPR065	DTCER065
—	Reserved	66	0108h	—	N/A	N/A	N/A	N/A	—	—	—
BLE	BLEIRQ	67	010Ch	Edge	✓	N/A	✓	N/A	IER08.IEN3	IPR067	—
ICU	IRQ4	68	0110h	Edge/Level	✓	✓	N/A	✓	IER08.IEN4	IPR068	DTCER068
	IRQ5	69	0114h	Edge/Level	✓	✓	N/A	✓	IER08.IEN5	IPR069	DTCER069
	IRQ6	70	0118h	Edge/Level	✓	✓	N/A	✓	IER08.IEN6	IPR070	DTCER070
	IRQ7	71	011Ch	Edge/Level	✓	✓	N/A	✓	IER08.IEN7	IPR071	DTCER071
—	Reserved	72	0120h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	73	0124h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	74	0128h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	75	012Ch	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	76	0130h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	77	0134h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	78	0138h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	79	013Ch	—	N/A	N/A	N/A	N/A	—	—	—
ELC	ELSR8I	80	0140h	Edge	✓	N/A	N/A	✓	IER0A.IEN0	IPR080	—
—	Reserved	81	0144h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	82	0148h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	83	014Ch	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	84	0150h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	85	0154h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	86	0158h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	87	015Ch	—	N/A	N/A	N/A	N/A	—	—	—
LVD	LVD1	88	0160h	Edge	✓	N/A	N/A	✓	IER0B.IEN0	IPR088	—
—	Reserved	89	0164h	—	N/A	N/A	N/A	N/A	—	—	—

Downloaded from Arrow.com.

Table 15.3 Interrupt Vector Table (3/6)

Source of Interrupt Request Generation	Name	Vector No.*1	Vector Address Offset	Interrupt Detection Method	CPU Interrupt	Start DTC	Start DMAC	SSBY Return	IER	IPR	DTCER
USB0	USBR0	90	0168h	Level	✓	N/A	N/A	✓	IER0B.IEN2	IPR090	—
VBATT	VBTLVDI	91	016Ch	Edge	✓	N/A	N/A	✓	IER0B.IEN3	IPR091	—
RTC	ALM	92	0170h	Edge	✓	N/A	N/A	✓	IER0B.IEN4	IPR092	—
	PRD	93	0174h	Edge	✓	N/A	N/A	✓	IER0B.IEN5	IPR093	—
—	Reserved	94	0178h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	95	017Ch	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	96	0180h	—	N/A	N/A	N/A	N/A	—	—	—
	Reserved	97	0184h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	98	0188h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	99	018Ch	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	100	0190h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	101	0194h	—	N/A	N/A	N/A	N/A	—	—	—
S12AD	S12ADI0	102	0198h	Edge	✓	✓	✓	N/A	IER0C.IEN6	IPR102	DTCER102
	GBADI	103	019Ch	Edge	✓	✓	✓	N/A	IER0C.IEN7	IPR103	DTCER103
CMPB1	CMPB2	104	01A0h	Edge	✓	✓	✓	N/A	IER0D.IEN0	IPR104	DTCER104
	CMPB3	105	01A4h	Edge	✓	✓	✓	N/A	IER0D.IEN1	IPR105	DTCER105
ELC	ELSR18I	106	01A8h	Edge	✓	✓	✓	N/A	IER0D.IEN2	IPR106	DTCER106
	ELSR19I	107	01ACh	Edge	✓	✓	✓	N/A	IER0D.IEN3	IPR107	DTCER107
SSI0	SSIF0	108	01B0h	Level	✓	N/A	N/A	N/A	IER0D.IEN4	IPR108	—
	SSIRXI0	109	01B4h	Edge	✓	✓	✓	N/A	IER0D.IEN5		DTCER109
	SSITXI0	110	01B8h	Edge	✓	✓	✓	N/A	IER0D.IEN6		DTCER110
TSIP	RD	111	01BCh	Edge	✓	✓	✓	N/A	IER0D.IEN7	IPR111	DTCER111
	WR	112	01C0h	Edge	✓	✓	✓	N/A	IER0E.IEN0		DTCER112
	ERR	113	01C4h	Edge	✓	N/A	N/A	N/A	IER0E.IEN1	IPR113	—
MTU0	TGIA0	114	01C8h	Edge	✓	✓	✓	N/A	IER0E.IEN2	IPR114	DTCER114
	TGIB0	115	01CCh	Edge	✓	✓	N/A	N/A	IER0E.IEN3		DTCER115
	TGIC0	116	01D0h	Edge	✓	✓	N/A	N/A	IER0E.IEN4		DTCER116
	TGID0	117	01D4h	Edge	✓	✓	N/A	N/A	IER0E.IEN5		DTCER117
	TCIV0	118	01D8h	Edge	✓	N/A	N/A	N/A	IER0E.IEN6	IPR118	—
	TGIE0	119	01DCh	Edge	✓	N/A	N/A	N/A	IER0E.IEN7		—
	TGIF0	120	01E0h	Edge	✓	N/A	N/A	N/A	IER0F.IEN0		—
MTU1	TGIA1	121	01E4h	Edge	✓	✓	✓	N/A	IER0F.IEN1	IPR121	DTCER121
	TGIB1	122	01E8h	Edge	✓	✓	N/A	N/A	IER0F.IEN2		DTCER122
	TCIV1	123	01ECh	Edge	✓	N/A	N/A	N/A	IER0F.IEN3	IPR123	—
	TCIU1	124	01F0h	Edge	✓	N/A	N/A	N/A	IER0F.IEN4		—
MTU2	TGIA2	125	01F4h	Edge	✓	✓	✓	N/A	IER0F.IEN5	IPR125	DTCER125
	TGIB2	126	01F8h	Edge	✓	✓	N/A	N/A	IER0F.IEN6		DTCER126
	TCIV2	127	01FCh	Edge	✓	N/A	N/A	N/A	IER0F.IEN7	IPR127	—
	TCIU2	128	0200h	Edge	✓	N/A	N/A	N/A	IER10.IEN0		—
MTU3	TGIA3	129	0204h	Edge	✓	✓	✓	N/A	IER10.IEN1	IPR129	DTCER129
	TGIB3	130	0208h	Edge	✓	✓	N/A	N/A	IER10.IEN2		DTCER130
	TGIC3	131	020Ch	Edge	✓	✓	N/A	N/A	IER10.IEN3		DTCER131
	TGID3	132	0210h	Edge	✓	✓	N/A	N/A	IER10.IEN4		DTCER132
	TCIV3	133	0214h	Edge	✓	N/A	N/A	N/A	IER10.IEN5	IPR133	—

Downloaded from Arrow.com.

Table 15.3 Interrupt Vector Table (4/6)

Source of Interrupt Request Generation	Name	Vector No.*1	Vector Address Offset	Interrupt Detection Method	CPU Interrupt	Start DTC	Start DMAC	SSBY Return	IER	IPR	DTCER
MTU4	TGIA4	134	0218h	Edge	✓	✓	✓	N/A	IER10.IEN6	IPR134	DTCER134
	TGIB4	135	021Ch	Edge	✓	✓	N/A	N/A	IER10.IEN7		DTCER135
	TGIC4	136	0220h	Edge	✓	✓	N/A	N/A	IER11.IEN0		DTCER136
	TGID4	137	0224h	Edge	✓	✓	N/A	N/A	IER11.IEN1		DTCER137
	TCIV4	138	0228h	Edge	✓	✓	N/A	N/A	IER11.IEN2	IPR138	DTCER138
—	Reserved	139	022Ch	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	140	0230h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	141	0234h	—	N/A	N/A	N/A	N/A	—	—	—
TPU0	TGI0A	142	0238h	Edge	✓	✓	✓	N/A	IER11.IEN6	IPR142	DTCER142
	TGI0B	143	023Ch	Edge	✓	✓	N/A	N/A	IER11.IEN7		DTCER143
	TGI0C	144	0240h	Edge	✓	✓	N/A	N/A	IER12.IEN0		DTCER144
	TGI0D	145	0244h	Edge	✓	✓	N/A	N/A	IER12.IEN1		DTCER145
	TCI0V	146	0248h	Edge	✓	N/A	N/A	N/A	IER12.IEN2	IPR146	—
TPU1	TGI1A	147	024Ch	Edge	✓	✓	✓	N/A	IER12.IEN3	IPR147	DTCER147
	TGI1B	148	0250h	Edge	✓	✓	N/A	N/A	IER12.IEN4		DTCER148
	TCI1V	149	0254h	Edge	✓	N/A	N/A	N/A	IER12.IEN5	IPR149	—
	TCI1U	150	0258h	Edge	✓	N/A	N/A	N/A	IER12.IEN6		—
TPU2	TGI2A	151	025Ch	Edge	✓	✓	✓	N/A	IER12.IEN7	IPR151	DTCER151
	TGI2B	152	0260h	Edge	✓	✓	N/A	N/A	IER13.IEN0		DTCER152
	TCI2V	153	0264h	Edge	✓	N/A	N/A	N/A	IER13.IEN1	IPR153	—
	TCI2U	154	0268h	Edge	✓	N/A	N/A	N/A	IER13.IEN2		—
TPU3	TGI3A	155	026Ch	Edge	✓	✓	✓	N/A	IER13.IEN3	IPR155	DTCER155
	TGI3B	156	0270h	Edge	✓	✓	N/A	N/A	IER13.IEN4		DTCER156
	TGI3C	157	0274h	Edge	✓	✓	N/A	N/A	IER13.IEN5		DTCER157
	TGI3D	158	0278h	Edge	✓	✓	N/A	N/A	IER13.IEN6		DTCER158
	TCI3V	159	027Ch	Edge	✓	N/A	N/A	N/A	IER13.IEN7	IPR159	—
TPU4	TGI4A	160	0280h	Edge	✓	✓	✓	N/A	IER14.IEN0	IPR160	DTCER160
	TGI4B	161	0284h	Edge	✓	✓	N/A	N/A	IER14.IEN1		DTCER161
	TCI4V	162	0288h	Edge	✓	N/A	N/A	N/A	IER14.IEN2	IPR162	—
	TCI4U	163	028Ch	Edge	✓	N/A	N/A	N/A	IER14.IEN3		—
TPU5	TGI5A	164	0290h	Edge	✓	✓	✓	N/A	IER14.IEN4	IPR164	DTCER164
	TGI5B	165	0294h	Edge	✓	✓	N/A	N/A	IER14.IEN5		DTCER165
	TCI5V	166	0298h	Edge	✓	N/A	N/A	N/A	IER14.IEN6	IPR166	—
	TCI5U	167	029Ch	Edge	✓	N/A	N/A	N/A	IER14.IEN7		—
—	Reserved	168	02A0h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	169	02A4h	—	N/A	N/A	N/A	N/A	—	—	—
POE	OEI1	170	02A8h	Level	✓	N/A	N/A	N/A	IER15.IEN2	IPR170	—
	OEI2	171	02ACh	Level	✓	N/A	N/A	N/A	IER15.IEN3	IPR171	—
—	Reserved	172	02B0h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	173	02B4h	—	N/A	N/A	N/A	N/A	—	—	—
TMR0	CMIA0	174	02B8h	Edge	✓	✓	N/A	N/A	IER15.IEN6	IPR174	DTCER174
	CMIB0	175	02BCh	Edge	✓	✓	N/A	N/A	IER15.IEN7		DTCER175
	OVI0	176	02C0h	Edge	✓	N/A	N/A	N/A	IER16.IEN0		—
TMR1	CMIA1	177	02C4h	Edge	✓	✓	N/A	N/A	IER16.IEN1	IPR177	DTCER177
	CMIB1	178	02C8h	Edge	✓	✓	N/A	N/A	IER16.IEN2		DTCER178
	OVI1	179	02CCh	Edge	✓	N/A	N/A	N/A	IER16.IEN3		—

Downloaded from Arrow.com.

Table 15.3 Interrupt Vector Table (5/6)

Source of Interrupt Request Generation	Name	Vector No.*1	Vector Address Offset	Interrupt Detection Method	CPU Interrupt	Start DTC	Start DMAC	SSBY Return	IER	IPR	DTCER
TMR2	CMIA2	180	02D0h	Edge	✓	✓	N/A	N/A	IER16.IEN4	IPR180	DTCER180
	CMIB2	181	02D4h	Edge	✓	✓	N/A	N/A	IER16.IEN5		DTCER181
	OVI2	182	02D8h	Edge	✓	N/A	N/A	N/A	IER16.IEN6		—
TMR3	CMIA3	183	02DCh	Edge	✓	✓	N/A	N/A	IER16.IEN7	IPR183	DTCER183
	CMIB3	184	02E0h	Edge	✓	✓	N/A	N/A	IER17.IEN0		DTCER184
	OVI3	185	02E4h	Edge	✓	N/A	N/A	N/A	IER17.IEN1		—
—	Reserved	186	02E8h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	187	02ECh	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	188	02F0h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	189	02F4h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	190	02F8h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	191	02FCh	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	192	0300h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	193	0304h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	194	0308h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	195	030Ch	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	196	0310h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	197	0314h	—	N/A	N/A	N/A	N/A	—	—	—
DMAC	DMAC0I	198	0318h	Edge	✓	✓	N/A	N/A	IER18.IEN6	IPR198	DTCER198
	DMAC1I	199	031Ch	Edge	✓	✓	N/A	N/A	IER18.IEN7	IPR199	DTCER199
	DMAC2I	200	0320h	Edge	✓	✓	N/A	N/A	IER19.IEN0	IPR200	DTCER200
	DMAC3I	201	0324h	Edge	✓	✓	N/A	N/A	IER19.IEN1	IPR201	DTCER201
—	Reserved	202	0328h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	203	032Ch	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	204	0330h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	205	0334h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	206	0338h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	207	033Ch	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	208	0340h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	209	0344h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	210	0348h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	211	034Ch	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	212	0350h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	213	0354h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	214	0358h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	215	035Ch	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	216	0360h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	217	0364h	—	N/A	N/A	N/A	N/A	—	—	—
SCI1	ERI1	218	0368h	Level	✓	N/A	N/A	N/A	IER1B.IEN2	IPR218	—
	RXI1	219	036Ch	Edge	✓	✓	✓	N/A	IER1B.IEN3		DTCER219
	TXI1	220	0370h	Edge	✓	✓	✓	N/A	IER1B.IEN4		DTCER220
	TEI1	221	0374h	Level	✓	N/A	N/A	N/A	IER1B.IEN5		—
SCI5	ERI5	222	0378h	Level	✓	N/A	N/A	N/A	IER1B.IEN6	IPR222	—
	RXI5	223	037Ch	Edge	✓	✓	✓	N/A	IER1B.IEN7		DTCER223
	TXI5	224	0380h	Edge	✓	✓	✓	N/A	IER1C.IEN0		DTCER224
	TEI5	225	0384h	Level	✓	N/A	N/A	N/A	IER1C.IEN1		—

Downloaded from Arrow.com.

Table 15.3 Interrupt Vector Table (6/6)

Source of Interrupt Request Generation	Name	Vector No.*1	Vector Address Offset	Interrupt Detection Method	CPU Interrupt	Start DTC	Start DMAC	SSBY Return	IER	IPR	DTCER
BLE	ERI	226	0388h	Level	✓	N/A	N/A	N/A	IER1C.IEN2	IPR226	—
	RXI	227	038Ch	Edge	✓	✓	✓	N/A	IER1C.IEN3		DTCER227
	TXI	228	0390h	Edge	✓	✓	✓	N/A	IER1C.IEN4		DTCER228
	TEI	229	0394h	Level	✓	N/A	N/A	N/A	IER1C.IEN5		—
SCI8	ERI8	230	0398h	Level	✓	N/A	N/A	N/A	IER1C.IEN6	IPR230	—
	RXI8	231	039Ch	Edge	✓	✓	✓	N/A	IER1C.IEN7		DTCER231
	TXI8	232	03A0h	Edge	✓	✓	✓	N/A	IER1D.IEN0		DTCER232
	TEI8	233	03A4h	Level	✓	N/A	N/A	N/A	IER1D.IEN1		—
—	Reserved	234	03A8h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	235	03ACh	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	236	03B0h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	237	03B4h	—	N/A	N/A	N/A	N/A	—	—	—
SCI12	ERI12	238	03B8h	Level	✓	N/A	N/A	N/A	IER1D.IEN6	IPR238	—
	RXI12	239	03BCh	Edge	✓	✓	✓	N/A	IER1D.IEN7		DTCER239
	TXI12	240	03C0h	Edge	✓	✓	✓	N/A	IER1E.IEN0		DTCER240
	TEI12	241	03C4h	Level	✓	N/A	N/A	N/A	IER1E.IEN1		—
	SCIX0	242	03C8h	Level	✓	N/A	N/A	N/A	IER1E.IEN2	IPR242	—
	SCIX1	243	03CCh	Level	✓	N/A	N/A	N/A	IER1E.IEN3	IPR243	—
	SCIX2	244	03D0h	Level	✓	N/A	N/A	N/A	IER1E.IEN4	IPR244	—
	SCIX3	245	03D4h	Level	✓	N/A	N/A	N/A	IER1E.IEN5	IPR245	—
RIIC0	EEI0	246	03D8h	Level	✓	N/A	N/A	N/A	IER1E.IEN6	IPR246	—
	RXI0	247	03DCh	Edge	✓	✓	✓	N/A	IER1E.IEN7	IPR247	DTCER247
	TXI0	248	03E0h	Edge	✓	✓	✓	N/A	IER1F.IEN0	IPR248	DTCER248
	TEI0	249	03E4h	Level	✓	N/A	N/A	N/A	IER1F.IEN1	IPR249	—
—	Reserved	250	03E8h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	251	03ECh	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	252	03F0h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	253	03F4h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	254	03F8h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	255	03FCh	—	N/A	N/A	N/A	N/A	—	—	—

Note 1. An interrupt source with a smaller vector number takes precedence.

15.3.2 Fast Interrupt Vector Table

The address of the entry in the interrupt vector table that corresponds to the vector number of the fast interrupt is placed in the fast interrupt vector register (FINTV) of the CPU.

15.3.3 Non-maskable Interrupt Vector Area

Non-maskable interrupts use the vector area in the exception vector table.
The exception vector table is allocated to the 128-byte area (4 bytes × 32 sources) beginning with the address set in the EXTB register in the CPU. Set a multiple of 4 in the EXTB register.

Downloaded from Arrow.com.

15.4 Interrupt Operation

The interrupt controller performs the following processing.

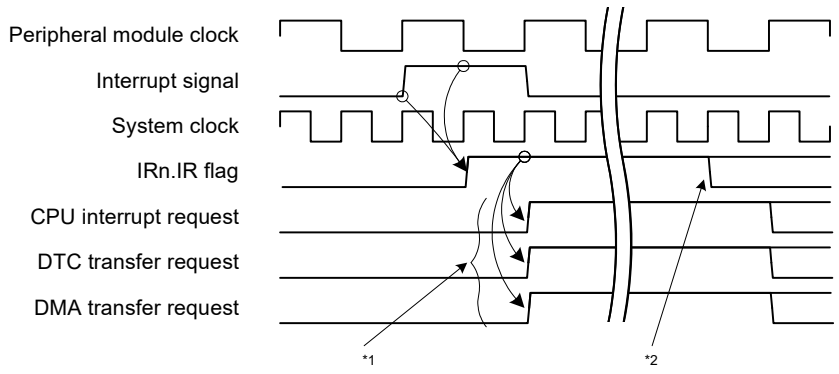
- Detecting interrupts
- Enabling and disabling interrupts
- Selecting interrupt request destinations (CPU interrupt, DTC trigger, or DMAC trigger)
- Determining priority

15.4.1 Detecting Interrupts

Interrupt requests are detected in either of two ways: the detection of edges of the interrupt signal or the detection of a level of the interrupt signal.
Edge detection or level detection is selected for the IRQi pins (i = 0, 1, and 4 to 7) as external interrupt requests by the setting of the IRQMD[1:0] bits in IRQCRi.
For interrupts from peripheral modules, either edge detection or level detection is determined per interrupt source.
For the correspondence between interrupt sources and methods of detection, see Table 15.3, Interrupt Vector Table.

15.4.1.1 Operation of Status Flags for Edge-Detected Interrupts

Figure 15.2 shows the operation of the IR flag in IRn (n = interrupt vector number) in the case of edge detection of an interrupt from a peripheral module or on an external pin.
The IR flag in IRn is set to 1 immediately after the transition of the interrupt signal due to generation of the interrupt. If the CPU is the request destination for the interrupt, the IR flag is automatically cleared to 0 on acceptance of the interrupt. If the DMAC or DTC is the request destination for the interrupt, the IRn.IR flag operation differs according to the DMAC/DTC transfer settings and transfer count. For details, see Table 15.4, Operation When Starting the DMAC/DTC.



Note 1. One of the following requests is issued: CPU interrupt request, DTC transfer request, and DMA transfer request. For details of the setting, see section 15.4.3, Selecting Interrupt Request Destinations.
Note 2. When the CPU interrupt request is specified, this flag is set to 0 on acceptance of a CPU interrupt. For the timing of 0 setting at the DTC transfer request or DMA transfer request, see Table 15.4, Operation When Starting the DMAC/DTC.

Figure 15.2 IRn.IR Flag Operation for Edge Detection Interrupts

Downloaded from Arrow.com.

Figure 15.3 to Figure 15.6 show the interrupt signals of the interrupt controller. Note that the timings of the interrupts with interrupt vector numbers 64 to 95 are different from those of other interrupts. For the IRQ pin interrupts with interrupt vector numbers 64 to 79, “internal delay + 2 PCLK cycles” of delay is added after the IRQ pin input. For the interrupts with interrupt vector numbers 80 to 95, “2 PCLK cycles” of delay is added.

If an interrupt signal is generated every clock cycle, the subsequent interrupts cannot be detected; secure two or more clock cycles of the system clock or peripheral module clock, whichever is slower, between issuance of continuous interrupt requests.

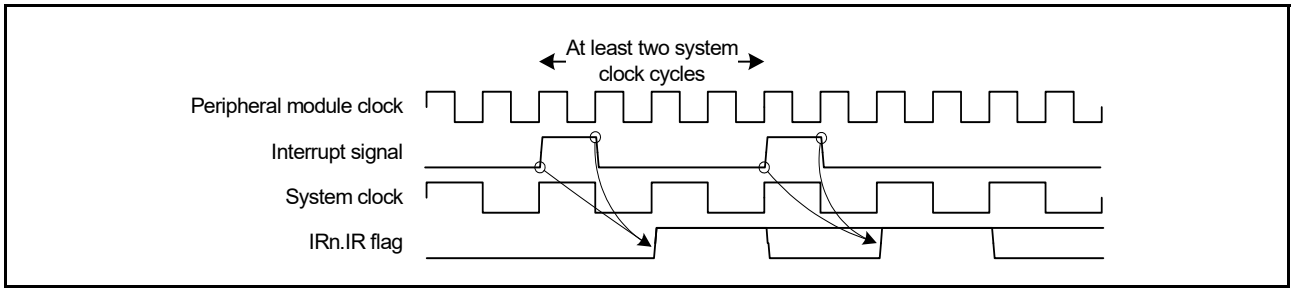


Figure 15.3 Interval Required between Issuance of Continuous Interrupt Requests (when the Frequency of System Clock is Slower than that of the Peripheral Module Clock)

While the IRn.IR flag is 1 after an interrupt request is generated, the interrupt request that is generated again will be ignored.*1
Figure 15.4 shows the timing for IRn.IR flag re-setting.

Note 1. When the transmission or reception interrupt of the SCI, RSPI, RIIC, USB, SSI, SDHI, or RSCAN is generated with the IRn.IR flag being 1, the interrupt request is retained. After the IRn.IR flag is cleared to 0, the IRn.IR flag is set to 1 again by the retained request. For details, see descriptions of the interrupts in section 33, Serial Communications Interface (SCIg, SCIh), section 35, I2C-bus Interface (RIICa), and section 38, Serial Peripheral Interface (RSPIa).

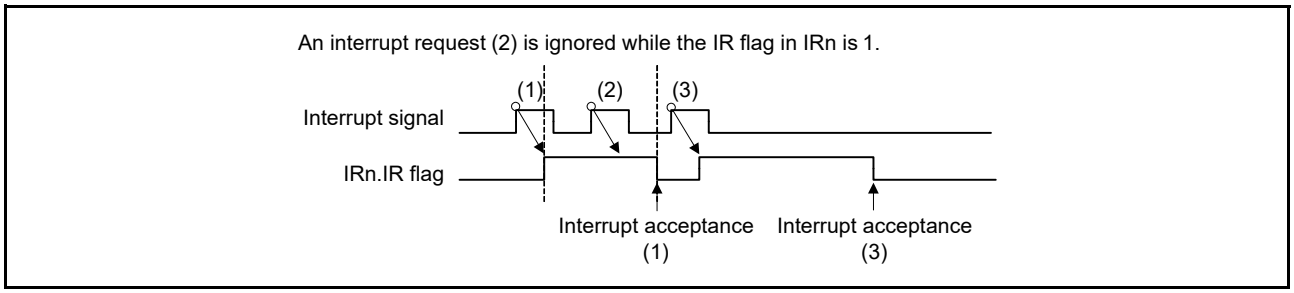


Figure 15.4 Timing for IRn.IR Flag Re-Setting

If an interrupt is disabled after the IRn.IR flag is set to 1 (output of the interrupt request is disabled by the interrupt enable bit of the relevant peripheral module), the IRn.IR flag is not affected but retains its state. Figure 15.5 shows operation when the interrupt is disabled.

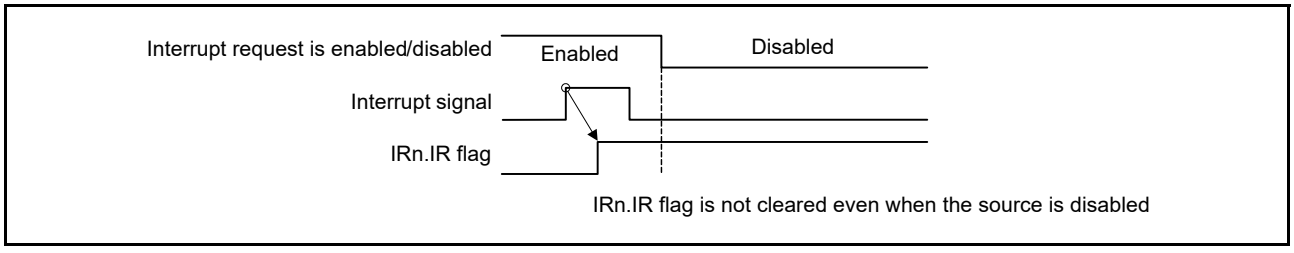


Figure 15.5 Relationship between IRn.IR Flag Operation and Disabling of Interrupt Request

Downloaded from Arrow.com.

15.4.1.2 Operation of Status Flags for Level-Detected Interrupts

Figure 15.6 shows the operation of the interrupt status flag (IR flag) in IRn (n = interrupt vector number) in the case of level detection of an interrupt from a peripheral module or an external pin.
The IR flag in IRn remains set to 1 as long as the interrupt signal is asserted. To clear the IRn.IR flag to 0, clear the interrupt request in the source generating the interrupt. Confirm that the interrupt request flag in the source generating the interrupt has been cleared to 0 and that the IRn.IR flag has been cleared to 0, and then complete the interrupt handling.

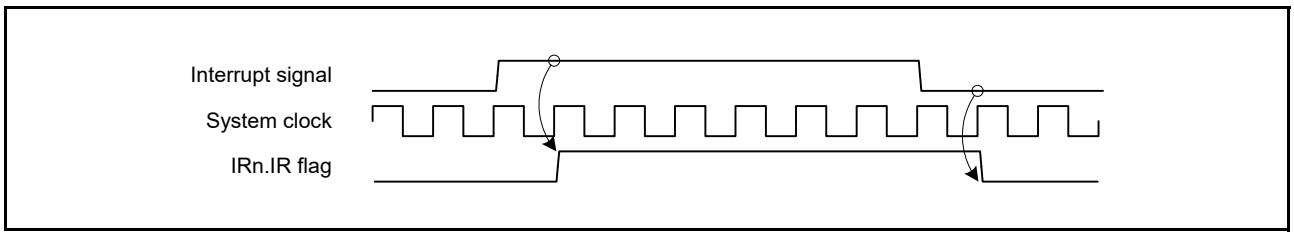


Figure 15.6 IRn.IR Flag Operation for Level Detection Interrupts

Figure 15.7 shows the procedure for handling level detection interrupts.

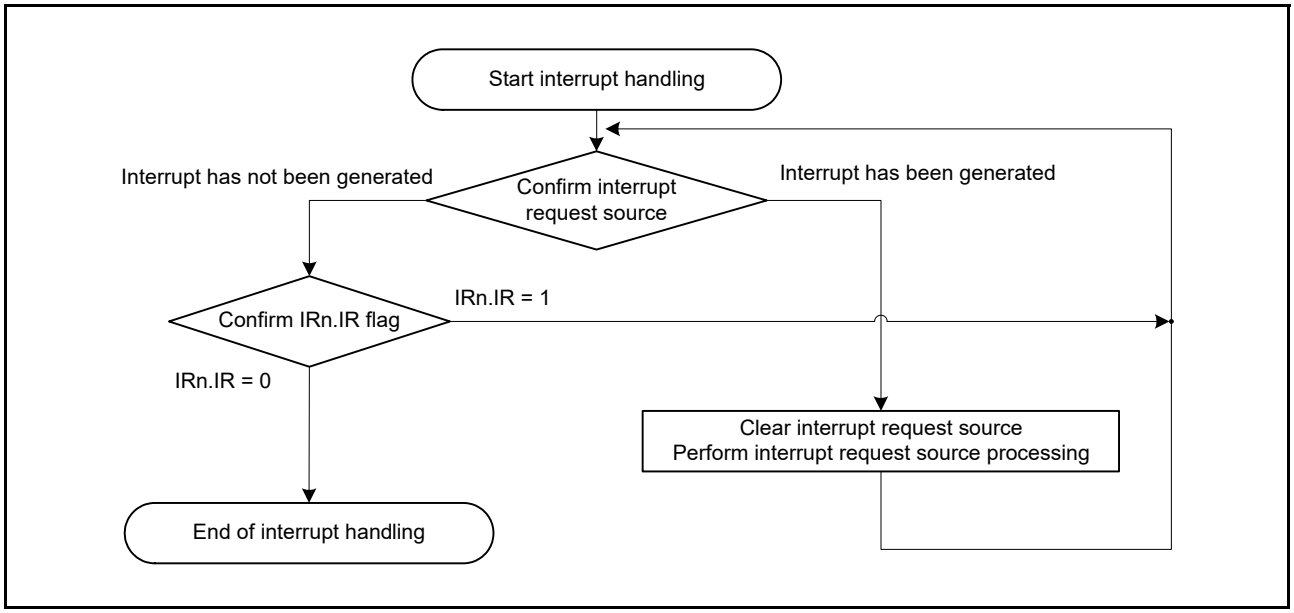


Figure 15.7 Procedure for Handling Level Detection Interrupts

Downloaded from Arrow.com.

15.4.2 Enabling and Disabling Interrupt Sources

Enabling requests from a given interrupt source requires the following settings.

1. In the case of interrupt requests from peripheral modules, setting the interrupt enable bit for the peripheral module to permit the output of interrupt requests from the source
2. Enabling of the interrupt by the IERm.IENj bit (m = 02h to 1Fh; j = 0 to 7)

When an interrupt request that is enabled at the corresponding source is generated, the corresponding IRn.IR flag (n = interrupt vector number) is set to 1.
Setting the IERm.IENj bit to enable an interrupt request allows the interrupt request for which the corresponding IRn.IR is 1 to be output to the interrupt request destination. Setting the IERm.IENj bit to disable an interrupt request suspends the output of the interrupt request for which the corresponding IRn.IR is 1.
The IRn.IR flag is not affected by the IERm.IENj bit.

Use the following procedure to disable interrupt requests.

1. Set the IERm.IENj bit to disable interrupt requests.
2. Set the peripheral module interrupt output enable bit to disable the output. Read the last written register and confirm that writing is completed.
3. Check the IRn.IR flag, and clear the IRn.IR flag if necessary.*1

Note 1. To disable the transmission or reception interrupt of the SCI, RSPI, or RIIC from the enabled state, clear the IRn.IR flag to 0 using the above procedure. For details, see descriptions of the interrupts in section 33, Serial Communications Interface (SCIg, SCIh), section 35, I2C-bus Interface (RIICa), and section 38, Serial Peripheral Interface (RSPIa).

Downloaded from Arrow.com.

15.4.3 Selecting Interrupt Request Destinations

Possible settings for the request destination of each interrupt are fixed. That is, settings for request destination other than those indicated in Table 15.3, Interrupt Vector Table, are not possible. Do not make an interrupt request destination setting that is not indicated by a "✓" in Table 15.3.
If the DMAC or DTC is selected as the destination for requests from an IRQi pin (i = 0, 1, and 4 to 7), be sure to set the IRQMD[1:0] bits in IRQCRi for that interrupt to select edge detection.
The following describes how to specify the destinations of interrupt requests.

(1) DMAC Trigger

Make the following settings for each source while the IERm.IENj bit (m = 02h to 1Fh; j = 0 to 7) is 0.

1. Specify the vector number of the desired interrupt in the DMAC trigger select register (DMRSRm) for the required channel of the DMAC.*1
2. Set the trigger for the target DMAC channel (DMACm.DMTMD.DCTG[1:0]) to 01b (interrupt module detection).
3. Set the DMAC transfer request enable bit for the target DMAC channel (DMACm.DMCNT.DTE) to 1.

After making the above settings, set the IERm.IENj bit to 1.
In addition, set the DMAC operation enable bit (DMAST.DMST) to 1. The order of making settings for each interrupt and enabling the DMAC operation enable bit does not matter.
For the DMAC setting procedure, refer to section 18.3.7, Activating the DMAC in section 18, DMA Controller (DMACA).

(2) DTC Trigger

Make the following settings for each source while the IERm.IENj bit (m = 02h to 1Fh; j = 0 to 7) is 0.

- Set the DTC transfer request enable bit in the DTC transfer request enable register (DTCERn.DTCE (n = interrupt vector number)) for the pertinent source to 1.*1

After making the above settings, set the IERm.IENj bit to 1.
In addition, set the DTC module start bit (DTCST.DTCST) to 1. The order of making settings for each interrupt and enabling the DTC module start bit does not matter.
For the DTC setting procedure, refer to section 19.5, DTC Setting Procedure, in section 19, Data Transfer Controller (DTCa).

Note 1. Do not set a DTC transfer request enable bit (DTCERn.DTCE) and a DMAC trigger select register (DMRSRm) to select the same source. Do not select the same source in more than one DMRSRm register.

Downloaded from Arrow.com.

(3) CPU Interrupt Request

If the interrupt request destination is neither the DMAC nor the DTC, the interrupt request is sent to the CPU. Set the IERm.IENj bit (m = 02h to 1Fh; j = 0 to 7) to 1 while neither the DMAC trigger settings nor the DTC trigger settings described above are in place.

Table 15.4 shows operation when the DMAC or DTC is the request destination.

Table 15.4 Operation When Starting the DMAC/DTC

Interrupt Request Destination	DISEL *1	Remaining Number of Transfer Operations	Operation per Request	IR*2	Interrupt Request Destination after Transfer
DMAC	1	≠ 0	DMA transfer → CPU interrupt	Cleared on interrupt acceptance by the CPU	DMAC
		= 0	DMA transfer → CPU interrupt	Cleared on interrupt acceptance by the CPU	The DMACm.DMCNT.DTE bit is cleared and the CPU becomes the destination.
	0	≠ 0	DMA transfer	Cleared at the start of DMAC transfer	DMAC
		= 0	DMA transfer*3	Cleared at the start of DMAC transfer*3	The DMACm.DMCNT.DTE bit is cleared and the CPU becomes the destination.
DTC*4	1	≠ 0	DTC transfer → CPU interrupt	Cleared on interrupt acceptance by the CPU	DTC
		= 0	DTC transfer → CPU interrupt	Cleared on interrupt acceptance by the CPU	The DTCERn.DTCE bit is cleared and the CPU becomes the destination.
	0	≠ 0	DTC transfer	Cleared at the start of DTC data transfer after reading DTC transfer information	DTC
		= 0	DTC transfer → CPU interrupt*3	Cleared on interrupt acceptance by the CPU*3	The DTCERn.DTCE bit is cleared and the CPU becomes the destination.

Note 1. DISEL for the DMAC is set by the DMACm.DMCSL.DISEL bit; DISEL for the DTC is set by the DTC.MRB.DISEL bit.
Note 2. When the IRn.IR flag is 1, an interrupt request (DTC or DMA transfer request) that is generated again will be ignored.
Note 3. When the DISEL bit is 0, operation with the remaining number of transfer operations being 0 differs according to whether the source is for the DTC or DMAC.
Note 4. For chain transfer, DTC transfer continues until the last chain transfer ends. Whether a CPU interrupt is generated at the end of chain transfer, the IRn.IR flag clear timing, and the interrupt request destination after transfer are determined by the state of DISEL and the remaining transfer count at the end of chain transfer. For the chain transfer, see Table 19.3, Chain Transfer Conditions in section 19, Data Transfer Controller (DTCa).

The request destination for an interrupt should be changed while the IERm.IENj bit is 0.

When a source is to be changed to an interrupt request or the DMAC trigger source is to be changed while a transfer is not complete (i.e. while the DMACm.DMCNT.DTE bit has not been cleared) after the settings described under (1) DMAC Trigger have been made, follow the procedure below.

1. For both the source to be withdrawn and the source that will have a new trigger, clear the IENj bits in IERm to 0.
2. Check the state of transfer by the DMAC. If transfer is in progress, wait for its completion.
3. Make the settings described under (1) DMAC Trigger.

When a source is to be changed to an interrupt request or the DTC transfer information is to be changed while a transfer is not complete (i.e. while the DTCERn.DTCE bit (n = interrupt vector number) has not been cleared) after the settings described under (2) DTC Trigger have been made, follow the procedure below.

1. For both the source to be withdrawn and the source that will have a new trigger, clear the IENj bits in IERm to 0.
2. Check the state of transfer by the DTC. If transfer is in progress, wait for its completion.
3. Make the settings described under (2) DTC Trigger.

Downloaded from Arrow.com.

15.4.4 Determining Priority

Interrupt priority is determined for each interrupt request destination.
The priority for each interrupt request destination is determined as follows.

(1) Determining Priority when the CPU is the Request Destination of the Interrupt

A source selected for the fast interrupt has the highest priority. After that, an interrupt source with a larger value of the interrupt priority level select bits (IPR[3:0]) in IPRn takes priority. If interrupts with the same priority level are generated by multiple sources, the source with the smallest vector number takes precedence.

(2) Determining Priority when the DTC is the Request Destination of the Interrupt

The IPR[3:0] bits in IPRn (n = interrupt vector number) have no effect. An interrupt source with a smaller vector number takes precedence.

(3) Determining Priority when the DMAC is the Request Destination of the Interrupt

The IPR[3:0] bits in IPRn have no effect. Regarding the order of priority of DMAC channels, see section 18, DMA Controller (DMACA).

15.4.5 Multiple Interrupts

To enable multiple interrupts of the CPU, set the PSW.I bit to 1 (interrupt enabled) in the handling routine of accepted interrupts.
The PSW.IPL[3:0] bits immediately after processing branches to the interrupt handling routine are set to the same value as the interrupt priority level of the accepted interrupt request. If an interrupt request which has an interrupt level higher than that of the PSW.IPL[3:0] bits is generated at this time, this interrupt request (for multiple interrupts) is accepted.
If the interrupt priority level of the accepted interrupt request is 15 (fast interrupt or interrupt when IPR[3:0] are set to 1111b), multiple interrupts are not generated.

15.4.6 Fast Interrupt

The fast interrupt is an interrupt for executing a faster interrupt response by the CPU, so only one of the interrupt sources can be assigned.
The interrupt priority level of the fast interrupt is 15 (highest) regardless of the setting of the IPR[3:0] bits in IPRn (n = interrupt vector number). In addition, the fast interrupt is accepted with precedence over other interrupt sources with level 15. However, when the value of the PSW.IPL[3:0] bits are 1111b (priority level 15), even the fast interrupt cannot be accepted.
To assign an interrupt source to the fast interrupt, specify the vector number of the source in the FIR.FVCT[7:0] bits, and set the FIR.FIEN bit to 1 (fast interrupt is enabled).
For details on the fast interrupt, see section 2, CPU and section 14, Exception Handling.

Downloaded from Arrow.com.

15.4.7 Digital Filter

The digital filter function is provided for the external interrupt request IRQi pins (i = 0, 1, and 4 to 7) and NMI pin interrupt.

The digital filter samples input signals at the filter sampling clock (PCLK) and removes the pulses of which length is less than three sampling cycles.

To use the digital filter for the IRQi pin, set the sampling clock cycle (PCLK, PCLK/8, PCLK/32, or PCLK/64) with the IRQFLTC0.FCLKSELi[1:0] bits and set the IRQFLTE0.FLTENi bit to 1 (digital filter enabled).

To use the digital filter for the NMI pin, set the sampling clock cycle (PCLK, PCLK/8, PCLK/32, or PCLK/64) with the NMIFLTC.NFCLKSEL[1:0] bits and set the NMIFLTE.NFLTEN bit to 1 (digital filter enabled).

Figure 15.8 shows an example of digital filter operation.

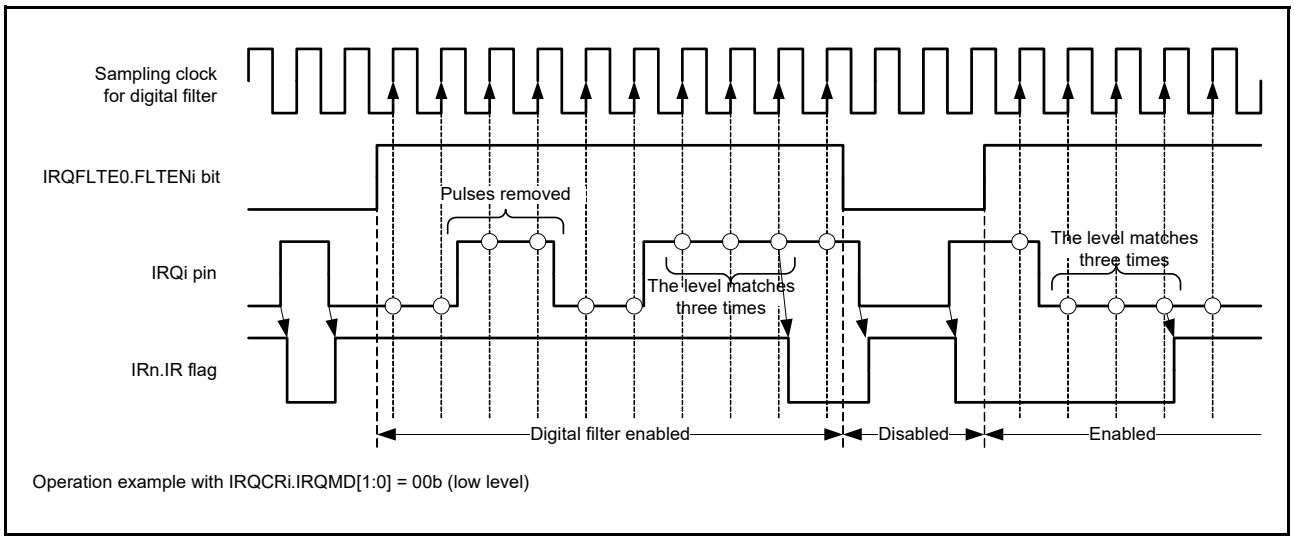


Figure 15.8 Digital Filter Operation Example

Before software standby mode is entered, set the IRQFLTE0.FLTENi and NMIFLTE.NFLTEN bits to 0 (digital filter disabled). To use the digital filter again after return from software standby mode, set the IRQFLTE0.FLTENi or NMIFLTE.NFLTEN bit to 1 (digital filter enabled).

15.4.8 External Pin Interrupts

The procedure for using the signal on an external pin as an interrupt is as follows.

1. Clear the IERm.IENj bit (m = 02h to 1Fh; j = 0 to 7) to 0 (interrupt request disabled).
2. Clear the IRQFLTE0.FLTENi bit (i = 0, 1, and 4 to 7) to 0 (digital filter disabled).*1
3. Set the digital filter sampling clock with the IRQFLTC0.FCLKSELi[1:0] bits.*1
4. Make or confirm the I/O port settings.
5. Set the method of detection for the interrupt in the IRQCRi.IRQMD[1:0] bits.
6. Clear the corresponding IRn.IR flag (n = interrupt vector number) to 0 (if edge detection is in use).
7. Set the IRQFLTE0.FLTENi bit to 1 (digital filter enabled).*1
8. If the interrupt is to be used for DMAC trigger, set the DMRSRm.DMRS[7:0] bits. If the interrupt is to be used for DTC trigger, set the DTCERn.DTCE bit. The interrupt will be a CPU interrupt if neither of these settings is made.
9. Set the IERm.IENj bit to 1 (interrupt request enabled).

Note 1. To use the digital filter function, settings must be made beforehand.

Downloaded from Arrow.com.

15.5 Non-maskable Interrupt Operation

There are six types of non-maskable interrupt: the NMI pin interrupt, oscillation stop detection interrupt, WDT underflow/refresh error, IWDT underflow/refresh error, voltage monitoring 1 interrupt, and VBATT voltage monitoring interrupt. Non-maskable interrupts are only usable as interrupts for the CPU; that is, they are not capable of DTC or DMAC trigger. Non-maskable interrupts take precedence over all interrupts, including the fast interrupt.
Non-maskable interrupt requests are accepted regardless of the states of the I (interrupt enable) bit and IPL[3:0] (processor interrupt priority level) bits in the PSW of the CPU. The current states of the non-maskable interrupts can be checked in the non-maskable interrupt status register (NMISR).
Confirm that all bits in the NMISR have returned to 0 from within the handler for the non-maskable interrupt, before ending the handler.
Non-maskable interrupts are disabled by default. If a system is to use non-maskable interrupts, the following procedure must be followed at the beginning of program processing.

Non-maskable interrupt usage procedure:

1. Set the stack pointer (SP).
2. To use the NMI pin, clear the NMIFLTE.NFLTEN bit to 0 (digital filter disabled).*1
3. To use the NMI pin, set the digital filter sampling clock with the NMIFLTC.NFCLKSEL[1:0] bits.*1
4. To use the NMI pin, set the NMI pin detection sense with the NMICR.NMIMD bit.
5. To use the NMI pin, write 1 to the NMICLR.NMICLR bit to clear the NMISR.NMIST flag to 0.
6. To use the NMI pin, set the NMIFLTE.NFLTEN bit to 1 (digital filter enabled).*1
7. Enable the non-maskable interrupt by writing 1 to the corresponding bit in the non-maskable interrupt enable register (NMIER).

Note 1. To use the digital filter function, settings must be made beforehand.

After 1 is written to the NMIER register, subsequent write access to the NMIEN bit in NMIER is ignored. The NMI interrupt cannot be disabled. It can be disabled only by a reset.
For the flow of non-maskable interrupt processing, see section 14, Exception Handling.
Writing 1 to the NMICLR.NMICLR bit clears the NMI status flag (NMISR.NMIST) to 0.
Writing 1 to the NMICLR.OSTCLR bit clears the oscillation stop detection interrupt status flag (NMISR.OSTST) to 0.
Writing 1 to the NMICLR.WDTCLR bit clears the WDT underflow/refresh error status flag (NMISR.WDTST) to 0.
Writing 1 to the NMICLR.IWDTCLR bit clears the IWDT underflow/refresh error status flag (NMISR.IWDTST) to 0.
Writing 1 to the NMICLR.LVD1CLR bit clears the voltage monitoring 1 interrupt status flag (NMISR.LVD1ST) to 0.
Writing 1 to the NMICLR.VBATCLR bit clears the VBATT voltage monitoring interrupt status flag (NMISR.VBATST) to 0.

Downloaded from Arrow.com.

15.6 Return from Low Power Consumption States

Interrupts can be used to return from sleep mode, deep sleep mode, or software standby mode.
Refer to section 11, Low Power Consumption for details. This section describes the procedure to set an interrupt source for returning from each low power consumption mode.

15.6.1 Return from Sleep Mode or Deep Sleep Mode

All non-maskable interrupts and all interrupts can be used to return from sleep mode and deep sleep mode. The following conditions must be satisfied.

(1) Non-maskable interrupts

- Generation of the interrupt that is used for return is enabled in the NMIER register.

(2) Interrupts

- The CPU is selected as the interrupt request destination.
- The interrupt request is enabled by the IERm.IENj bit (m = 02h to 1Fh; j = 0 to 7).
- The priority level is higher than the PSW.IPL[3:0] bit value in the CPU.

15.6.2 Return from Software Standby Mode

Non-maskable interrupts (excluding oscillation stop detection interrupt) and interrupts that have a "✓" in the SSBY return column in Table 15.3, Interrupt Vector Table can be used to return from software standby mode. The conditions below must be satisfied.

(1) Non-maskable interrupts

- Generation of the interrupt that is used for return is enabled in the NMIER register.
- When using the NMI pin interrupt, the digital filter is disabled.

(2) Interrupts

- The interrupt source can be used to return from software standby mode.
- The CPU is selected as the interrupt request destination.
- The interrupt request is enabled by the IERm.IENj bit (m = 02h to 1Fh; j = 0 to 7).
- The priority level is higher than the PSW.IPL[3:0] bit value in the CPU.
  When using the fast interrupt, set not only the FIR register but also the corresponding IPRn.IPR[3:0] bits (n = interrupt vector number). Set the IPRn.IPR[3:0] bits to a higher level than the PSW.IPL[3:0] bit value in the CPU.
- When using the external pin interrupt, the digital filter of the IRQi pin used is disabled.

Refer to section 15.4.7, Digital Filter for details on the procedure to set the digital filter.

Downloaded from Arrow.com.

15.7 Usage Note

15.7.1 Note on WAIT Instruction Used with Non-Maskable Interrupt

Before executing the WAIT instruction, check to see that all the status flags in NMISR are 0.

Downloaded from Arrow.com.

16. Buses

16.1 Overview

Table 16.1 lists the bus specifications, Figure 16.1 shows the bus configuration, and Table 16.2 lists the addresses assigned for each bus.

Table 16.1 Bus Specifications

Bus Type		Description
CPU bus	Instruction bus	• Connected to the CPU (for instructions) • Connected to on-chip memory (RAM, ROM) • Operates in synchronization with the system clock (ICLK)
	Operand bus	• Connected to the CPU (for operands) • Connected to on-chip memory (RAM, ROM) • Operates in synchronization with the system clock (ICLK)
Memory bus	Memory bus 1	• Connected to RAM
	Memory bus 2	• Connected to ROM
Internal main bus	Internal main bus 1	• Connected to the CPU • Operates in synchronization with the system clock (ICLK)
	Internal main bus 2	• Connected to the DMAC and DTC • Connected to on-chip memory (RAM, ROM) • Operates in synchronization with the system clock (ICLK)
Internal peripheral bus	Internal peripheral bus 1	• Connected to peripheral modules (DTC, DMAC, interrupt controller, and bus error monitoring section) • Operates in synchronization with the system clock (ICLK)
	Internal peripheral bus 2	• Connected to peripheral modules (modules other than those connected to internal peripheral buses 1, 3, and 4) • Operates in synchronization with the peripheral-module clock (PCLKB)
	Internal peripheral bus 3	• Connected to peripheral modules (USB0, RSCAN, and CTSU) • Operates in synchronization with the peripheral-module clock (PCLKB)
	Internal peripheral bus 4	• Connected to peripheral modules (MTU2) • Operates in synchronization with the peripheral-module clock (PCLKA)
	Internal peripheral bus 6	• Connected to the flash control module and E2 DataFlash • Operates in synchronization with the FlashIF clock (FCLK)

Downloaded from Arrow.com.

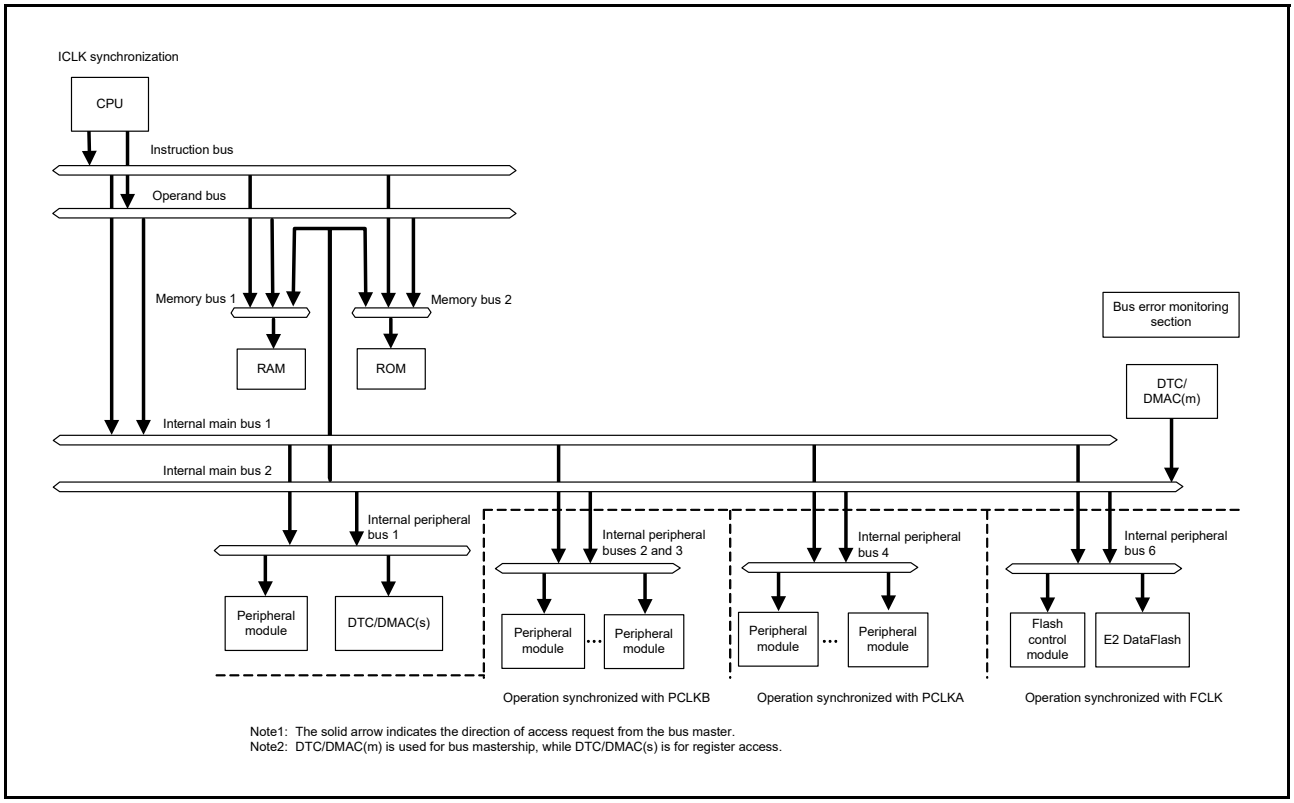


Figure 16.1 Bus Configuration

Table 16.2 Addresses Assigned for Each Bus

Address	Bus	Area
0000 0000h to 0000 FFFFh	Memory bus 1	RAM
0001 0000h to 0007 FFFFh		Reserved area
0008 0000h to 0008 7FFFh	Internal peripheral bus 1	Peripheral I/O registers
0008 8000h to 0009 FFFFh	Internal peripheral bus 2	
000A 0000h to 000B FFFFh	Internal peripheral bus 3	
000C 0000h to 000D FFFFh	Internal peripheral bus 4	
000E 0000h to 000F FFFFh	Reserved area	Reserved area
0010 0000h to 007F FFFFh	Internal peripheral bus 6	Flash control module and E2 DataFlash
0080 0000h to 00FF FFFFh		Reserved area
0100 0000h to 04FF FFFFh	Reserved area	
0500 0000h to 07FF FFFFh		
0800 0000h to 0FFF FFFFh		
1000 0000h to 7FFF FFFFh		
8000 0000h to FEFF FFFFh	Memory bus 2	ROM (for reading only)
FF00 0000h to FFFF FFFFh		

Downloaded from Arrow.com.

16.2 Description of Buses

16.2.1 CPU Buses

The CPU buses consist of the instruction and operand buses, which are connected to internal main bus 1. As the names suggest, the instruction bus is used to fetch instructions for the CPU, while the operand bus is used for operand access. The instruction bus is 64 bits while the operand bus is 32 bits.
Connection of the instruction and operand buses to RAM and ROM provides the CPU with direct access to these areas, i.e. access is not via internal main bus 1. However, only reading is possible in direct access to ROM by the CPU; programming and erasure are handled via an internal peripheral bus.
Bus requests for instruction fetching and operand access are arbitrated through internal main bus 1. The order of priority is operand access then instruction fetching.
If instruction fetching and operand access are requested for different buses (memory bus 1, memory bus 2, and internal main bus 1), the bus-access operations can proceed simultaneously. For example, parallel access to ROM and RAM is possible.

16.2.2 Memory Buses

The memory buses consist of memory bus 1 and memory bus 2. RAM is connected to memory bus 1 and ROM is connected to memory bus 2. The memory buses are 64 bits. Requests for bus mastership from the CPU buses (instruction fetching and operand) and internal main bus 2 are arbitrated through memory buses 1 and 2.
The priority order of the buses can be set using the memory bus 1 (RAM) priority control bits (BPRA[1:0]) and memory bus 2 (ROM) priority control bits (BPRO[1:0]) in the bus priority control register (BUSPRI) for the corresponding memory buses. When the priority order is fixed, internal main bus 2 has priority over the CPU bus (operand over instruction fetching). When the priority order is toggled, a bus has a lower priority when the request of that bus is accepted.

16.2.3 Internal Main Buses

The internal main buses consist of a bus for use by the CPU (internal main bus 1) and a bus for use by the other bus-master modules, i.e. the DTC and DMAC (internal main bus 2).
Bus requests for instruction fetching and operand access are arbitrated through internal main bus 1. The order of priority is operand access then instruction fetching.
Requests for bus mastership from the DTC and DMAC are arbitrated by internal main bus 2. The priority order is the DMAC and then DTC as listed in Table 16.3.
Between the DTC and DMAC, only the one that accepted the activation request issues the bus mastership request. The priority order of activation requests between the DTC and DMAC is DMAC0, DMAC1, DMAC2, DMAC3, and then DTC, regardless of the BUSPRI setting.
If the CPU and another bus master are requesting access to different buses (on-chip memory, internal peripheral buses 1 to 4, and 6), the respective bus-access operations can proceed simultaneously.
However, when the CPU executes the XCHG instruction, requests for bus access from masters other than the CPU are not accepted until data transfer for the XCHG instruction is completed regardless of the bus priority control register (BUSPRI) setting. Furthermore, requests for bus access from masters other than the DTC are not accepted during reading and writing-back of transfer control information for the DTC.

Downloaded from Arrow.com.

Table 16.3 Order of Priority for Bus Masters

Priority	Internal main buses	Bus Master
High ↑ Low	2	DMAC
		DTC
	1	CPU

Note: The above applies when the priority order of the buses is fixed.
The priority order of internal main bus 1 and another bus (internal main bus 2) can be toggled by the bus priority control register (BUSPRI) (round-robin method).

16.2.4 Internal Peripheral Buses

Connection of peripheral modules to the internal peripheral buses is as described in Table 16.4.

Table 16.4 Connection of Peripheral Modules to the Internal Peripheral Buses

Type of Bus	Peripheral Modules
Internal peripheral bus 1	DTC, DMAC, interrupt controller, and bus error monitoring section
Internal peripheral bus 2	Peripheral modules other than those connected to internal peripheral buses 1, 3, and 4
Internal peripheral bus 3	USB0, RSCAN, and CTSU
Internal peripheral bus 4	MTU2
Internal peripheral bus 6	Flash control module and E2 DataFlash

Requests for bus mastership from the CPU (internal main bus 1) and other bus masters (internal main bus 2) are arbitrated through internal peripheral buses 1 to 4, and 6.
The priority order of two internal main buses can be set using the bus priority control register (BUSPRI). The priority order can be set with the internal peripheral bus 1 priority control bits (BUSPRI.BPIB[1:0]), internal peripheral bus 2 and 3 priority control bits (BUSPRI.BPGB[1:0]), internal peripheral bus 4 priority control bits (BUSPRI.BPHB[1:0]), and internal peripheral bus 6 priority control bits (BUSPRI.BPFB[1:0]) for the corresponding internal peripheral buses.
When the priority order is fixed, internal main bus 2 has priority over internal main bus 1. When the priority order is toggled, a bus has a lower priority when the request of that bus is accepted (round-robin method).
The order of accepting requests may change depending on the BUSPRI setting (Refer to Figure 16.2).

Downloaded from Arrow.com.

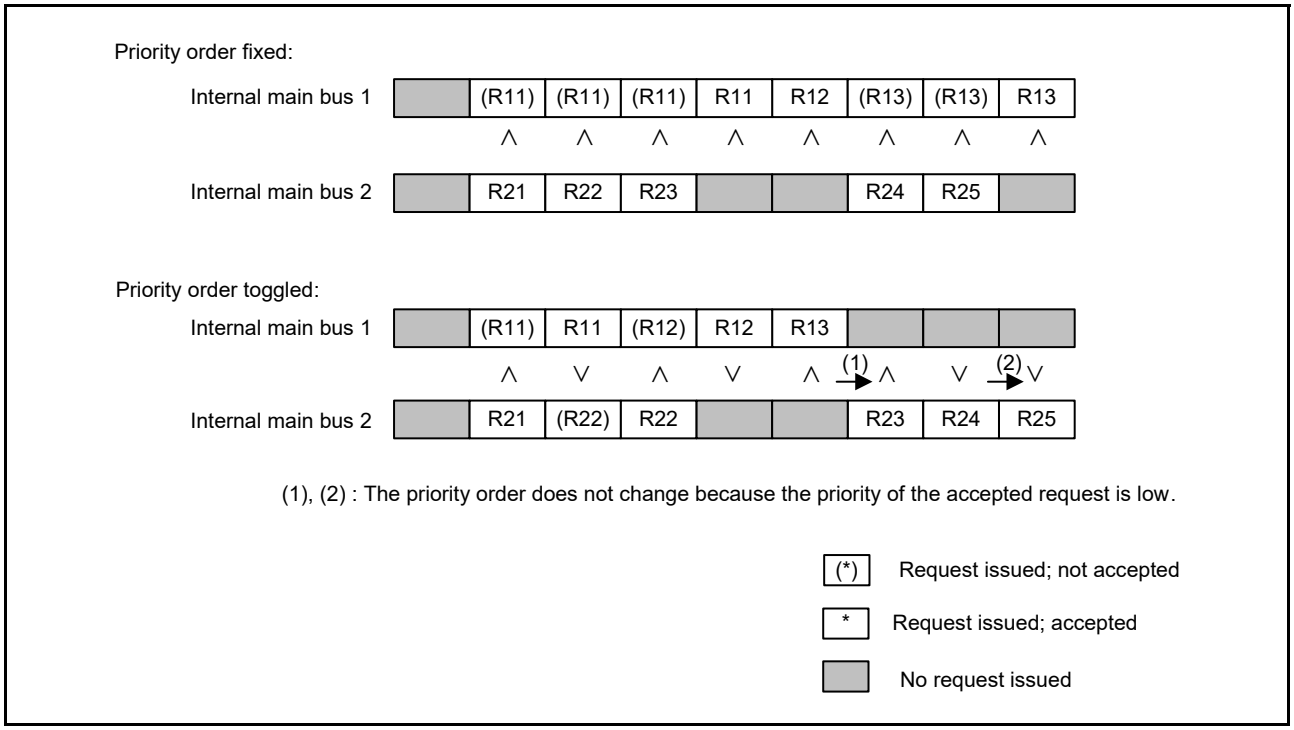


Figure 16.2 Priority Order between Internal Peripheral Bus Accesses

16.2.5 Write Buffer Function (Internal Peripheral Bus)

The internal peripheral bus has the write buffer function, which allows the next round of bus access to start, before the current write access is completed, in write access. However, if the following round of bus access is from the same bus master but to the different internal peripheral bus, it is suspended until the bus operations already in progress are completed. When read access to the internal memory is scheduled after the write access to the internal peripheral bus from the CPU, the following round of bus access can be started before the current bus operation is completed and thus the order of accesses may be changed (Refer to Figure 16.3).

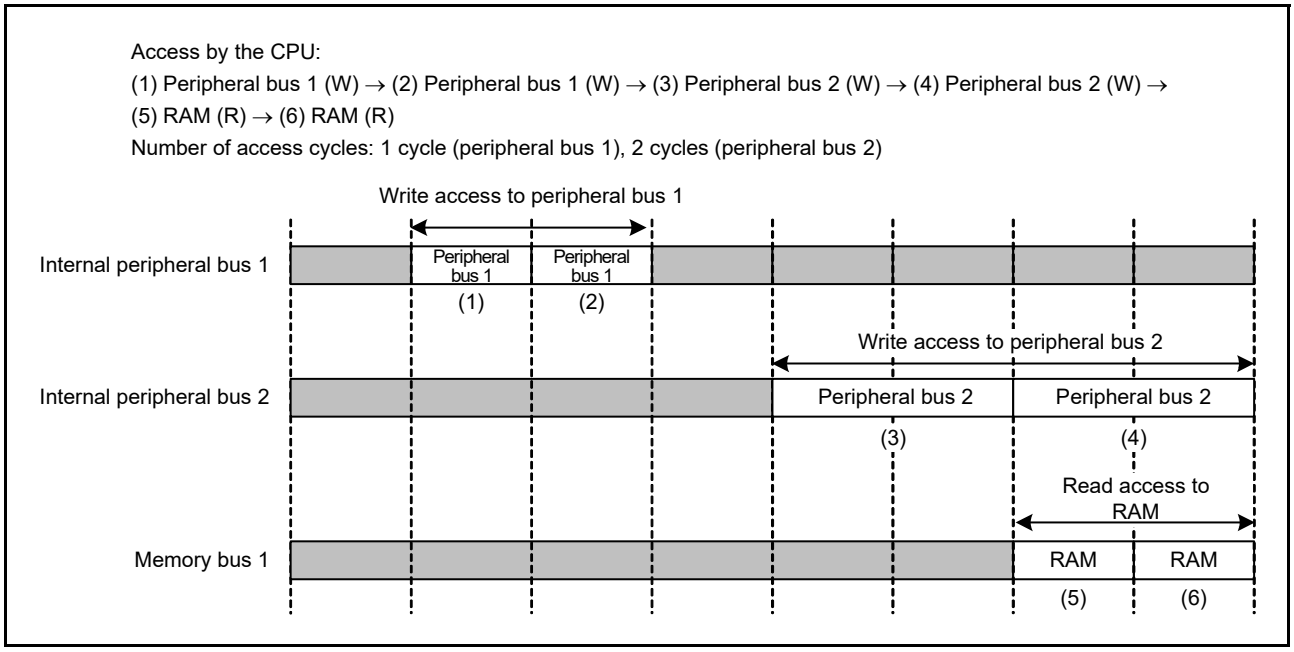


Figure 16.3 Write Buffer Function

Downloaded from Arrow.com.

16.2.6 Parallel Operation

Parallel operation is possible when different bus-master modules are requesting access to different slave modules. For example, if the CPU is fetching an instruction from ROM and an operand from RAM, the DMAC is able to handle transfer between a peripheral bus and the peripheral bus at the same time.
An example of parallel operations is shown in Figure 16.4. In this example, the CPU is able to employ the instruction and operand buses for simultaneous access to ROM and RAM, respectively. Furthermore, the DMAC simultaneously employs internal main bus 2 for access to a peripheral bus during access to RAM and ROM by the CPU.

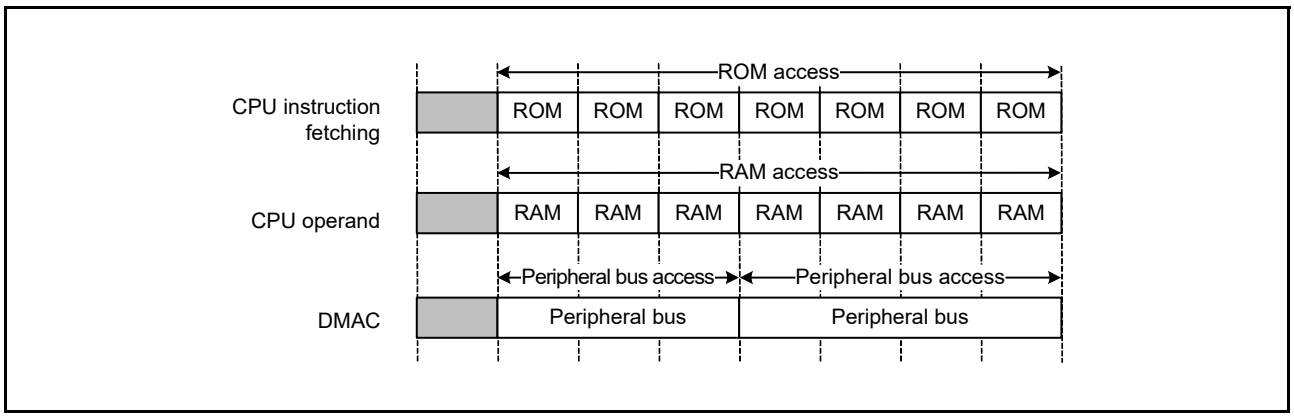


Figure 16.4 Example of Parallel Operations

Downloaded from Arrow.com.

16.3 Register Descriptions

16.3.1 Bus Error Status Clear Register (BERCLR)

Address(es): 0008 1300h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	STSCLR
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	STSCLR	Status Clear	0: Invalid 1: Bus error status register cleared	(W)*1
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Only writing 1 is effective; i.e. writing 0 has no effect.

STSCLR Bit (Status Clear)

Writing 1 to this bit clears the bus error status registers 1 and 2 (BERSR1 and BERSR2).
Writing 0 has no effect. It is read as 0.

16.3.2 Bus Error Monitoring Enable Register (BEREN)

Address(es): 0008 1304h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	TOEN	IGAEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	IGAEN	Illegal Address Access Detection Enable	0: Illegal address access detection is disabled. 1: Illegal address access detection is enabled.	R/W
b1	TOEN	Timeout Detection Enable*1, *2	0: Bus timeout detection is disabled. 1: Bus timeout detection is enabled.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. When detection is disabled (the TOEN bit is cleared to 0), bus access can cause the bus to freeze.
Note 2. Do not clear the TOEN bit to 0 (bus timeout detection disabled) while timeout errors are being detected.

Downloaded from Arrow.com.

16.3.3 Bus Error Status Register 1 (BERSR1)

Address(es): 0008 1308h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	MST[2:0]			—	—	TO	IA
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	IA	Illegal Address Access	0: Illegal address access not made 1: Illegal address access made	R
b1	TO	Timeout	0: Timeout not generated 1: Timeout generated	R
b3, b2	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b6 to b4	MST[2:0]	Bus Master Code	b6 b4 0 0 0: CPU 0 0 1: Reserved 0 1 0: Reserved 0 1 1: DTC/DMAC 1 0 0: Reserved 1 0 1: Reserved 1 1 0: Reserved 1 1 1: Reserved	R
b7	—	Reserved	This bit is read as 0. Writing to this bit has no effect.	R

MST[2:0] Bits (Bus Master Code)

These bits indicate the bus master that accessed a bus when a bus error occurred.

16.3.4 Bus Error Status Register 2 (BERSR2)

Address(es): 0008 130Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ADDR[12:0]													—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b15 to b3	ADDR[12:0]	Bus Error Occurrence Address	The upper 13 bits of an address that was accessed when a bus error occurred (in units of 512 Kbytes).	R

Downloaded from Arrow.com.

16.3.5 Bus Priority Control Register (BUSPRI)

Address(es): 0008 1310h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	BPFB[1:0]		BPHB[1:0]		BPGB[1:0]		BPIB[1:0]		BPRO[1:0]		BPRA[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	BPRA[1:0]	Memory Bus 1 (RAM) Priority Control	b1 b0 0 0: The order of priority is fixed. 0 1: The order of priority is toggled. 1 0: Setting prohibited 1 1: Setting prohibited	R/(W) *1
b3, b2	BPRO[1:0]	Memory Bus 2 (ROM) Priority Control	b3 b2 0 0: The order of priority is fixed. 0 1: The order of priority is toggled. 1 0: Setting prohibited 1 1: Setting prohibited	R/(W) *1
b5, b4	BPIB[1:0]	Internal Peripheral Bus 1 Priority Control	b5 b4 0 0: The order of priority is fixed. 0 1: The order of priority is toggled. 1 0: Setting prohibited 1 1: Setting prohibited	R/(W) *1
b7, b6	BPGB[1:0]	Internal Peripheral Bus 2 and 3 Priority Control	b7 b6 0 0: The order of priority is fixed. 0 1: The order of priority is toggled. 1 0: Setting prohibited 1 1: Setting prohibited	R/(W) *1
b9, b8	BPHB[1:0]	Internal Peripheral Bus 4 Priority Control	b9 b8 0 0: The order of priority is fixed. 0 1: The order of priority is toggled. 1 0: Setting prohibited 1 1: Setting prohibited	R/(W) *1
b11, b10	BPFB[1:0]	Internal Peripheral Bus 6 Priority Control	b11 b10 0 0: The order of priority is fixed. 0 1: The order of priority is toggled. 1 0: Setting prohibited 1 1: Setting prohibited	R/(W) *1
b15, b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. These bits can be written to only once while the DTC and DMAC are stopped. When they are written to more than one time, the operation is not guaranteed.

BPRA[1:0] Bits (Memory Bus 1 (RAM) Priority Control)

These bits specify the priority order for memory bus 1 (RAM).

When the priority order is fixed, internal main bus 2 has priority over internal main bus 1.

When the priority order is toggled, a bus has a lower priority when the request of that bus is accepted.

BPRO[1:0] Bits (Memory Bus 2 (ROM) Priority Control)

These bits specify the priority order for memory bus 2 (ROM).

When the priority order is fixed, internal main bus 2 has priority over internal main bus 1.

When the priority order is toggled, a bus has a lower priority when the request of that bus is accepted.

BPIB[1:0] Bits (Internal Peripheral Bus 1 Priority Control)

These bits specify the priority order for internal peripheral bus 1.

When the priority order is fixed, internal main bus 2 has priority over internal main bus 1.

Downloaded from Arrow.com.

When the priority order is toggled, a bus has a lower priority when the request of that bus is accepted.

BPGB[1:0] Bits (Internal Peripheral Bus 2 and 3 Priority Control)

These bits specify the priority order for internal peripheral buses 2 and 3.
When the priority order is fixed, internal main bus 2 has priority over internal main bus 1.
When the priority order is toggled, a bus has a lower priority when the request of that bus is accepted.

BPHB[1:0] Bits (Internal Peripheral Bus 4 Priority Control)

These bits specify the priority order for internal peripheral bus 4.
When the priority order is fixed, internal main bus 2 has priority over internal main bus 1.
When the priority order is toggled, a bus has a lower priority when the request of that bus is accepted.

BPFB[1:0] Bits (Internal Peripheral Bus 6 Priority Control)

These bits specify the priority order for internal peripheral bus 6.
When the priority order is fixed, internal main bus 2 has priority over internal main bus 1.
When the priority order is toggled, a bus has a lower priority when the request of that bus is accepted.

Downloaded from Arrow.com.

16.4 Bus Error Monitoring Section

The bus error monitoring section monitors the individual areas for bus errors, and when a bus error occurs, the error is indicated to the bus master.

16.4.1 Types of Bus Error

There are two types of bus error: illegal address access and timeout.
Illegal address access is the detection of illegal access to an area, and time-out is the detection of a bus-access operation not being completed within 768 cycles.

16.4.1.1 Illegal Address Access

When the illegal address access detection enable bit (IGAEN) in the bus error monitoring enable register (BEREN) is set to 1, access to the illegal address ranges will lead to illegal address access errors.
The address ranges where access will lead to illegal address access errors are indicated in Table 16.5.

16.4.1.2 Timeout

When the timeout detection enable bit (TOEN) in the bus error monitoring enable register (BEREN) is set to 1, bus access that is not completed within 768 cycles leads to a timeout error.

- Internal peripheral buses (2 and 3): Bus access is not completed within 768 peripheral module clock (PCLKB) cycles from the start of the access. In this MCU, a timeout error does not occur.
  Once a timeout error occurs, accesses from the bus master are rejected for 256 PCLKB cycles.
- Internal peripheral buses (4): Bus access is not completed within 768 peripheral module clock (PCLKA) cycles from the start of the access.
  Once a timeout error occurs, accesses from the bus master are rejected for 256 PCLKA cycles.
- Internal peripheral bus (6): Bus access is not completed within 768 FlashIF clock (FCLK) cycles from the start of the access.
  Once a timeout error occurs, accesses from the bus master are rejected for 256 FCLK cycles. In this MCU, a timeout error does not occur.

Downloaded from Arrow.com.

16.4.2 Operations When a Bus Error Occurs

When a bus error occurs, the error is indicated to the CPU. Operation is not guaranteed when a bus error occurs.

- Bus error indication to the CPU
  An interrupt is generated. The IERn register in the ICU can specify whether to generate an interrupt in the case of a bus error.

16.4.3 Conditions Leading to Bus Errors

Table 16.5 lists the types of bus errors for each area in the respective address space.
If an illegal address access error or timeout is detected when no bus error has occurred (bus error status register n (BERSRn; n = 1 or 2) is cleared), the detected error is reflected on the BERSRn. Once a bus error occurs, no subsequent bus errors are reflected on the register unless the register is cleared.
If bus errors are simultaneously caused by two or more bus masters, error information of only one bus master is reflected. Once a bus error occurs, the status is retained until BERSRn is cleared.

Table 16.5 Types of Bus Errors

Address	Type of Area	Type of Error	
		Illegal Address Access	Timeout
0000 0000h to 0007 FFFFh	Memory bus 1	—	—
0008 0000h to 0008 7FFFh	Internal peripheral bus 1	—	—
0008 8000h to 0009 FFFFh	Internal peripheral bus 2	Δ	—
000A 0000h to 000B FFFFh	Internal peripheral bus 3	Δ	—
000C 0000h to 000D FFFFh	Internal peripheral bus 4	Δ	✓
000E 0000h to 000F FFFFh	Reserved area	—	—
0010 0000h to 0011 FFFFh	Internal peripheral bus 6	—	—
0012 0000h to 007F FFFFh		Δ	—
0080 0000h to 00FF FFFFh		—	—
0100 0000h to 07FF FFFFh	Reserved area	—	—
0800 0000h to 0FFF FFFFh		—	—
1000 0000h to 7FFF FFFFh		✓	—
8000 0000h to FEFF FFFFh	Memory bus 2	—	—
FF00 0000h to FF7F FFFFh		—	—
FF80 0000h to FFFF FFFFh		—	—

—: A bus error does not result.
Δ: A bus error may or may not result.
✓: A bus error results.
Note: The capacity of the RAM, E2 DataFlash, and ROM differs depending on the product. For details, see section 49, RAM, section 50, Flash Memory (FLASH).

Downloaded from Arrow.com.

16.5 Interrupt

16.5.1 Interrupt Source

An illegal address access error or detection of a timeout leads to a bus error signal for the interrupt controller.

Table 16.6 Interrupt Source

Name	Interrupt Source	DTC Activation	DMAC Activation
BUSERR	Illegal address access error or timeout	Not possible	Not possible

Downloaded from Arrow.com.

17. Memory-Protection Unit (MPU)

17.1 Overview

The RXv2 CPU incorporates a memory-protection unit that checks the addresses of CPU access to the overall address space (0000 0000h to FFFF FFFFh).

Access-control information can be set for up to eight regions, and permission for access to each region is in accord with this information. The default response to the detection of access to a region where permission has not been set is the generation of a memory-protection error.

The supported access-control information for the individual regions consists of permission to read, permission to write, and permission to execute. This access-control information is effective when the processor mode of the CPU is user mode. Memory protection is not applied when the CPU is in supervisor mode.

Table 17.1 lists the specifications of the memory-protection unit, and Figure 17.1 shows a block diagram of the memory-protection unit.

Table 17.1 Specifications of Memory Protection

Specifictaions	Description
Region to be covered by memory protection and processor mode	0000 0000h to FFFF FFFFh (in user mode) No memory protection in supervisor mode
Number of regions	8
Page size (smallest unit of protection)	16 bytes
Specifying addresses of individual regions	Setting the page numbers where regions start and end
Setting to make memory protection effective or ineffective in individual regions	A V (valid) bit in each region-n end page number register (REPAGEn) makes the settings effective or ineffective for the corresponding region (n = 0 to 7).
Access-control information settings for individual regions	Instruction execution: Permission to execute Operand access: Permission to read, permission to write
Start of memory-protection operations	After the memory-protection unit has been enabled, access monitoring starting up with the transition to user mode.
Memory-protection error processing	Generation of access exceptions
Addresses where memory-protection errors are generated	Address in instruction execution: The PC value is preserved on the stack. Address in operand access: The address is stored in the data memory-protection error address register (MPDEA).
Determining the reasons for memory-protection errors	The memory-protection error status register (MPESTS) holds indicators of the reason.
Background region setting	Access-control information can be set for the background region (the whole address space).
Processing where regions overlap	The access-control information for access to an overlap between regions is the logical OR of the attributes for the given regions, and permission is given priority.

Downloaded from Arrow.com.

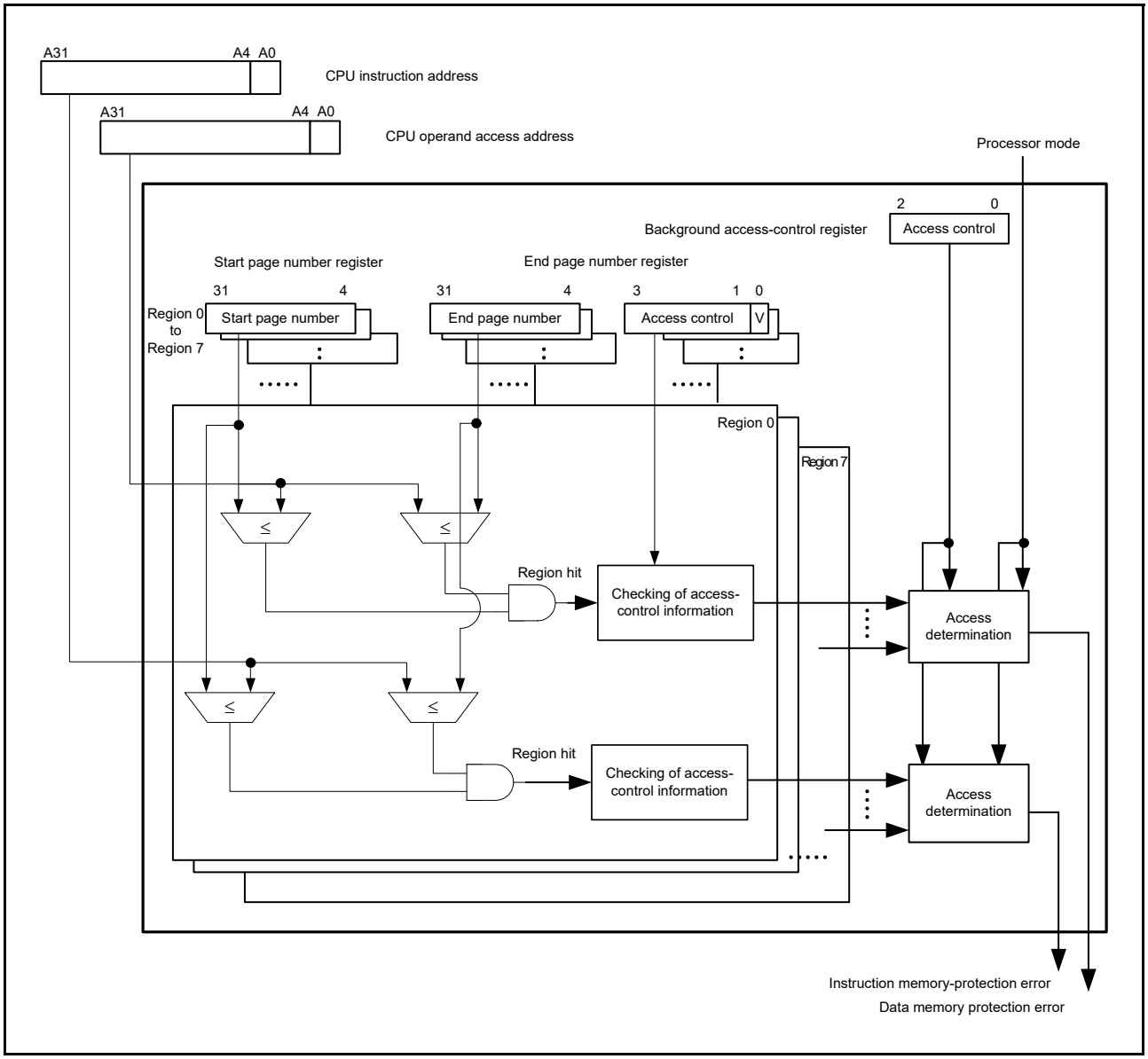


Figure 17.1 Block Diagram of the Memory-Protection Unit

Downloaded from Arrow.com.

17.1.1 Types of Access Control

There are three types of access control information: permission for instruction execution, permission to read operands, and permission to write operands. Violations of these types of access control are only detected when programs are running in user mode. Violations are not detected when programs are running in supervisor mode.

17.1.2 Regions for Access Control

Up to eight regions for access control are definable. Settings of the range of memory for each access-control region are made in the corresponding region-n start page number register (RSPAGEn) and region-n end page number register (REPAGEn), where n = 0 to 7.
The minimum unit for control of access is the “page”, by which the address space is divided into 16-byte units. The 28 higher-order bits ([31:4]) of the address [31:0] bits correspond to the page number.
The REPAGEn register specifies the access-control information for each area and whether the area is enabled or not.

17.1.3 Background Region

“Background region” refers to the whole address space (0000 0000h to FFFF FFFFh). Access-control information for the background region is set in the background-region access-control register (MPBAC). In contrast to the access-control information for the eight individual regions, protection information for the background region is effective as long as memory protection is enabled (the MPEN bit in the MPEN register is 1).

17.1.4 Overlap between Regions

In cases of overlap between multiple regions, the access-control information becomes the logical OR of the access-control bits for the overlapping regions (including the background region), with permission given priority.

17.1.5 Instructions and Data that Span Regions

Operations in response to the detection of memory-protection errors when instructions or data span regions for which different access-control settings have been made are undefined. Ensure that instructions and data do not span regions for which different access-control settings have been made.

Downloaded from Arrow.com.

17.2 Register Descriptions

17.2.1 Region-n Start Page Number Register (RSPAGEn) (n = 0 to 7)

Address(es): RSPAGE0 0008 6400h, RSPAGE1 0008 6408h, RSPAGE2 0008 6410h, RSPAGE3 0008 6418h, RSPAGE4 0008 6420h, RSPAGE5 0008 6428h, RSPAGE6 0008 6430h, RSPAGE7 0008 6438h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RSPN[27:0]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RSPN[27:0]												—	—	—	—
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	The read value is 0. The write value should always be 0.	R/W
b31 to b4	RSPN[27:0]	Region-Start Page Number	Page number where the region starts, for use in region determination	R/W

RSPN[27:0] Bits (Region-Start Page Number)

These bits specify the page number where the region starts.

Downloaded from Arrow.com.

17.2.2 Region-n End Page Number Register (REPAGEn) (n = 0 to 7)

Address(es): REPAGE0 0008 6404h, REPAGE1 0008 640Ch, REPAGE2 0008 6414h, REPAGE3 0008 641Ch, REPAGE4 0008 6424h, REPAGE5 0008 642Ch, REPAGE6 0008 6434h, REPAGE7 0008 643Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	REPN[27:0]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	REPN[27:0]												UAC[2:0]			V
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	V	Valid Bit	0: Region setting invalid 1: Region setting valid	R/W
b3 to b1	UAC[2:0]	Access Control Bits in User Mode	b3 0: Reading prohibited 1: Reading permitted b2 0: Writing prohibited 1: Writing permitted b1 0: Execution prohibited 1: Execution permitted	R/W
b31 to b4	REPN[27:0]	Region-End Page Number	Page number where the region ends, for use in region determination	R/W

V Bit (Valid Bit)

This bit enables or disables the settings for the corresponding region.

This bit is cleared to 0 when the region invalidation operation register (MPOPI) invalidates all access-controlled areas.

UAC[2:0] Bits (Access Control Bits in User Mode)

These bits specify the access control in user mode.

REPN[27:0] Bits (Region-End Page Number)

These bits specify the page number where the region ends.

Specify a value that is greater than or equal to the page number where the corresponding region starts. The page specified by the region-end page number is part of the target region for memory protection.

Downloaded from Arrow.com.

17.2.3 Memory-Protection Enable Register (MPEN)

Address(es): 0008 6500h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MPEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MPEN	Memory-Protection Enable	1: The memory protection is enabled. 0: The memory protection is disabled.	R/W
b31 to b1	—	Reserved	The read value is 0. The write value should always be 0	R/W

MPEN Bit (Memory-Protection Enable)

This bit enables or disables the memory protection.

After 1 has been written to this bit, address checking for memory protection by the CPU starts on the execution of a branch instruction (RTE or RTFI) that shifts operation to the user mode.

Downloaded from Arrow.com.

17.2.4 Background Access Control Register (MPBAC)

Address(es): 0008 6504h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	UBAC[2:0]			—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	The read value is 0. The write value should always be 0.	R/W
b3 to b1	UBAC[2:0]	Background Access Control Bits in User Mode	b3 0: Reading prohibited 1: Reading permitted b2 0: Writing prohibited 1: Writing permitted b1 0: Execution prohibited 1: Execution permitted	R/W
b31 to b4	—	Reserved	The read value is 0. The write value should always be 0.	R/W

UBAC[2:0] Bits (Background Access Control Bits in User Mode)

These bits specify the background access control in user mode.

Downloaded from Arrow.com.

17.2.5 Memory-Protection Error Status-Clearing Register (MPECLR)

Address(es): 0008 6508h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CLR	Error Status-Clearing	[Reading] 0: Fixed value for reading [Writing] 0: Nothing is done. 1: The DMPER and IMPER bits in MPESTS are cleared to 0.	R/W
b31 to b1	—	Reserved	The read value is 0. The write value should always be 0.	R/W

CLR Bit (Error Status-Clearing)

This bit clears the data read/write bit (DRW), the data memory-protection error generation bit (DMPER), and the instruction memory-protection error generation bit (IMPER) in the memory-protection error status register (MPESTS) to 0.

Downloaded from Arrow.com.

17.2.6 Memory-Protection Error Status Register (MPESTS)

Address(es): 0008 650Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	DRW	DMPER	IMPER
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	IMPER	Instruction Memory-Protection Error Generation	0: No instruction memory-protection error was generated. 1: Instruction memory-protection error was generated.	R
b1	DMPER	Data Memory-Protection Error Generation	0: No data memory-protection error was generated. 1: Data memory-protection error was generated.	R
b2	DRW	Data Read/Write	0: Data were read. 1: Data were written.	R
b31 to b3	—	Reserved	The read value is 0. The write value should always be 0.	R/W

IMPER Bit (Instruction Memory-Protection Error Generation)

This bit indicates the state of memory-protection error generation by instruction execution.

Setting the error status-clearing bit (CLR) in the memory-protection error status-clearing register (MPECLR) to 1 clears this bit to 0.

DMPER Bit (Data Memory-Protection Error Generation)

This bit indicates the state of memory-protection error generation by operand access.

Setting the error status-clearing bit (CLR) in the memory-protection error status-clearing register (MPECLR) to 1 clears this bit to 0.

DRW Bit (Data Read/Write)

For a memory-protection error produced by operand access, this bit indicates the read/write attribute of the access operation. This bit is only valid when the DMPER bit is 1.

Setting the error status-clearing bit (CLR) in the memory-protection error status-clearing register (MPECLR) to 1 clears this bit to 0.

Downloaded from Arrow.com.

17.2.7 Data Memory-Protection Error Address Register (MPDEA)

Address(es): 0008 6514h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DEA[31:0]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DEA[31:0]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	DEA[31:0]	Data Memory-Protection Error Address	Data memory-protection error address	R

DEA[31:0] Bits (Data Memory-Protection Error Address)

These bits retain the address for which operand access generated a memory-protection error.

17.2.8 Region Search Address Register (MPSA)

Address(es): 0008 6520h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	SA[31:0]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SA[31:0]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	SA[31:0]	Region Search Address	Address for region searching	R/W

SA[31:0] Bits (Region Search Address)

These bits specify the address for use in comparison with region-start addresses in the region-n start page number registers (RSPAGEn) and region-end addresses in the region-n end page number registers (REPAGEn).

Downloaded from Arrow.com.

17.2.9 Region Search Operation Register (MPOPS)

Address(es): 0008 6524h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	S
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	S	Region Search Operation Activation	[Reading] 0: Fixed value for reading [Writing] 0: Nothing is done. 1: A region-search operation proceeds.	R/W
b15 to b1	—	Reserved	The read value is 0. The write value should always be 0.	R/W

S Bit (Region Search Operation Activation)

Setting this bit to 1 makes the memory-protection unit perform a region-search operation. The address specified in the region search address register (MPSA) is compared with the address information for individual regions to search for a hitting region.

The result of searching is stored in the data-hit region bits (HITD[7:0]) of the data-hit region register (MHITD). Moreover, the logical OR of the respective access control bits for hitting regions is stored in the data-hit region access control bits (UHACD[2:0]) in user mode.

17.2.10 Region Invalidation Operation Register (MPOPI)

Address(es): 0008 6526h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	INV
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	INV	Region Invalidate Start	[Reading] 0: Fixed value for reading [Writing] 0: Nothing is done. 1: All access-controlled areas are invalidated.	R/W
b15 to b1	—	Reserved	The read value is 0. The write value should always be 0.	R/W

INV Bit (Region Invalidate Start)

Setting this bit to 1 clears the valid (V) bits in all of the region-n end page number registers (REPAGEn) to 0. After a V bit is cleared to 0, all settings other than background access-control settings are invalid.

Downloaded from Arrow.com.

17.2.11 Instruction-Hit Region Register (MHITI)

Address(es): 0008 6528h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	HITI[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	UHACI[2:0]			—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	The read value is 0. The write value should always be 0.	R/W
b3 to b1	UHACI[2:0]	Instruction-Hit Region Access Control Bits in User Mode	b3 0: Reading prohibited 1: Read permitted b2 0: Writing prohibited 1: Writing permitted b1 0: Execution prohibited 1: Execution is permitted.	R
b15 to b4	—	Reserved	The read value is 0. The write value should always be 0.	R/W
b23 to b16	HITI[7:0]	Instruction-Hit Region	When the instruction memory-protection error generation bit (MPESTS.IMPER) = 1, [b23:b16] = 0000 0000b indicates that attempted access to the background region led to an instruction memory-protection error. Other than above b23 0: Instruction memory-protection error was not generated in region 7. 1: Instruction memory-protection error was generated in region 7. b22 0: Instruction memory-protection error was not generated in region 6. 1: Instruction memory-protection error was generated in region 6. b21 0: Instruction memory-protection error was not generated in region 5. 1: Instruction memory-protection error was generated in region 5. b20 0: Instruction memory-protection error was not generated in region 4. 1: Instruction memory-protection error was generated in region 4. b19 0: Instruction memory-protection error was not generated in region 3. 1: Instruction memory-protection error was generated in region 3. b18 0: Instruction memory-protection error was not generated in region 2. 1: Instruction memory-protection error was generated in region 2. b17 0: Instruction memory-protection error was not generated in region 1. 1: Instruction memory-protection error was generated in region 1. b16 0: Instruction memory-protection error was not generated in region 0. 1: Instruction memory-protection error was generated in region 0.	R
b31 to b24	—	Reserved	The read value is 0. The write value should always be 0.	R/W

Downloaded from Arrow.com.

UHACI[2:0] Bits (Instruction-Hit Region Access Control Bits in User Mode)

These bits hold the user-mode access control bits (REPAGEn.UAC[2:0]) for the region where the instruction memory-protection error was generated.

If the error was generated in an overlap between regions, the value stored here is the logical OR of the user-mode access control bits for the corresponding regions (including the background region).

HITI[7:0] Bits (Instruction-Hit Region)

These bits indicate the region where an instruction memory-protection error was generated. These bits are set to 0000 0000b in response to the generation of an instruction memory-protection error in the background region.

Downloaded from Arrow.com.

17.2.12 Data-Hit Region Register (MHITD)

Address(es): 0008 652Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	HITD[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	UHACD[2:0]			—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	The read value is 0. The write value should always be 0.	R/W
b3 to b1	UHACD[2:0]	Data-Hit Region Access Control Bits in User Mode	b3 0: Reading prohibited 1: Reading permitted b2 0: Writing prohibited 1: Writing permitted b1 0: Execution prohibited 1: Execution permitted	R
b15 to b4	—	Reserved	The read value is 0. The write value should always be 0.	R/W
b23 to b16	HITD[7:0]	Data-Hit Region	When the data memory-protection error generation bit (MPESTS.DMPER) = 1, [b23:b16] = 0000 0000b indicates that attempted access to the background region led to a data memory-protection error. Other than above b23 0: Neither a data memory-protection error nor a search hit was generated in region 7. 1: A data memory-protection error or search hit was generated in region 7. b22 0: Neither a data memory-protection error nor a search hit was generated in region 6. 1: A data memory-protection error or search hit was generated in region 6. b21 0: Neither a data memory-protection error nor a search hit was generated in region 5. 1: A data memory-protection error or search hit was generated in region 5. b20 0: Neither a data memory-protection error nor a search hit was generated in region 4. 1: A data memory-protection error or search hit was generated in region 4. b19 0: Neither a data memory-protection error nor a search hit was generated in region 3. 1: A data memory-protection error or search hit was generated in region 3. b18 0: Neither a data memory-protection error nor a search hit was generated in region 2. 1: A data memory-protection error or search hit was generated in region 2. b17 0: Neither a data memory-protection error nor a search hit was generated in region 1. 1: A data memory-protection error or search hit was generated in region 1. b16 0: Neither a data memory-protection error nor a search hit was generated in region 0. 1: A data memory-protection error or search hit was generated in region 0.	R
b31 to b24	—	Reserved	The read value is 0. The write value should always be 0.	R/W

UHACD[2:0] Bits (Data-Hit Region Access Control Bits in User Mode)

These bits hold the user-mode access control bits (REPAGEn.UAC[2:0]) that have been set for the region where a data memory-protection error was generated or the region that produced a hit in region searching.

When an error is generated in an overlap between regions or a hit was generated in region searching, the value stored here is the logical OR of the user-mode access control bits for the corresponding regions (including the background region).

Downloaded from Arrow.com.

HITD[7:0] Bits (Data-Hit Region)

These bits indicate the region where a data memory-protection error was generated or the region that produced a hit in a region search. These bits are set to 0000 0000b for a data memory-protection error generated in the background region.

Note: When access to a register of memory protection unit in user mode generates a data memory-protection error, the value in this register is cleared to 0000 0000h.

Downloaded from Arrow.com.

17.3 Functions

17.3.1 Memory Protection

Memory protection means monitoring, in accord with the access-control information that has been set for the individual access-control regions and the background region, whether or not access by programs running in user mode violates the access-control settings. The memory-protection unit notifies the CPU of access-control violations (or memory-protection errors) when they are detected, causing the CPU to start access-exception processing.
Memory protection is enabled by setting the memory-protection enable (MPEN) bit in the memory-protection enable (MPEN) register to 1.
An instruction memory-protection error is generated on detection of an instruction-execution violation and a data memory-protection error is generated on detection of an operand-access reading or writing violation. Operand access that leads to a data memory-protection error is not actually executed.

17.3.2 Region Search

Region search means enquiry as to which of the eight specified access regions was “hit” and how the access-control information (permission to execute, to read, and to write) is set.
When the region search operation (S) bit in the region-search operation (MPOP) register is set to 1, the address specified in the region search address (MPSA) register is compared with the addresses for the individual regions. After a region search is executed, the data-hit region register (MHITD) indicates the logical OR of the access-control information for the region which was “hit” and for the other regions.

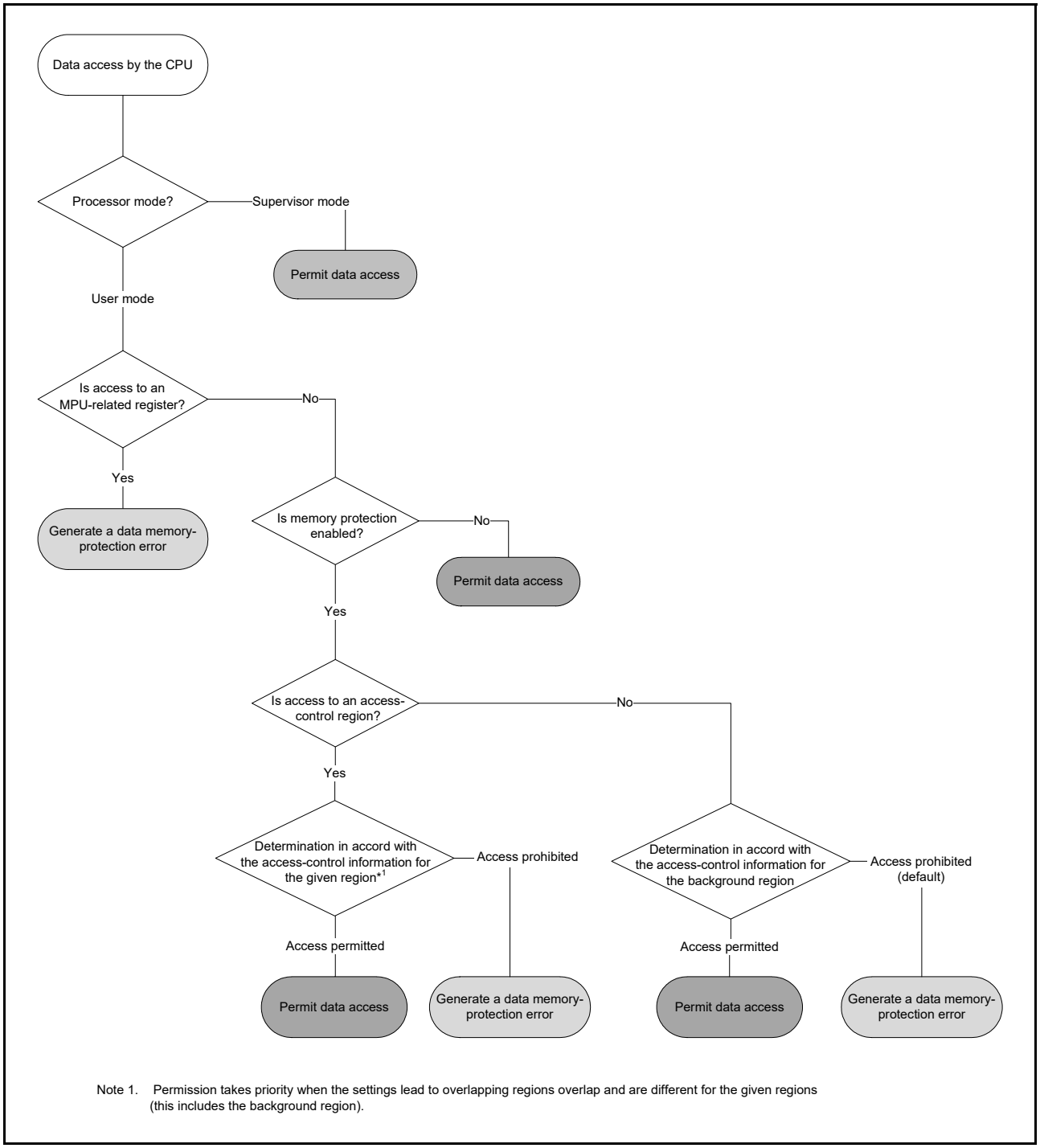
17.3.3 Protection of Registers Related to the Memory-Protection Unit

Registers related to the memory-protection unit are not accessible through means of access other than operand access by the CPU (i.e. by instruction fetching or DMA). Attempted access to registers related to the memory-protection unit in user mode through operand access by the CPU leads to a data memory-protection error regardless of whether or not memory protection is in effect at the given location.

Downloaded from Arrow.com.

17.3.4 Flow for Determination of Access by the Memory-Protection Function

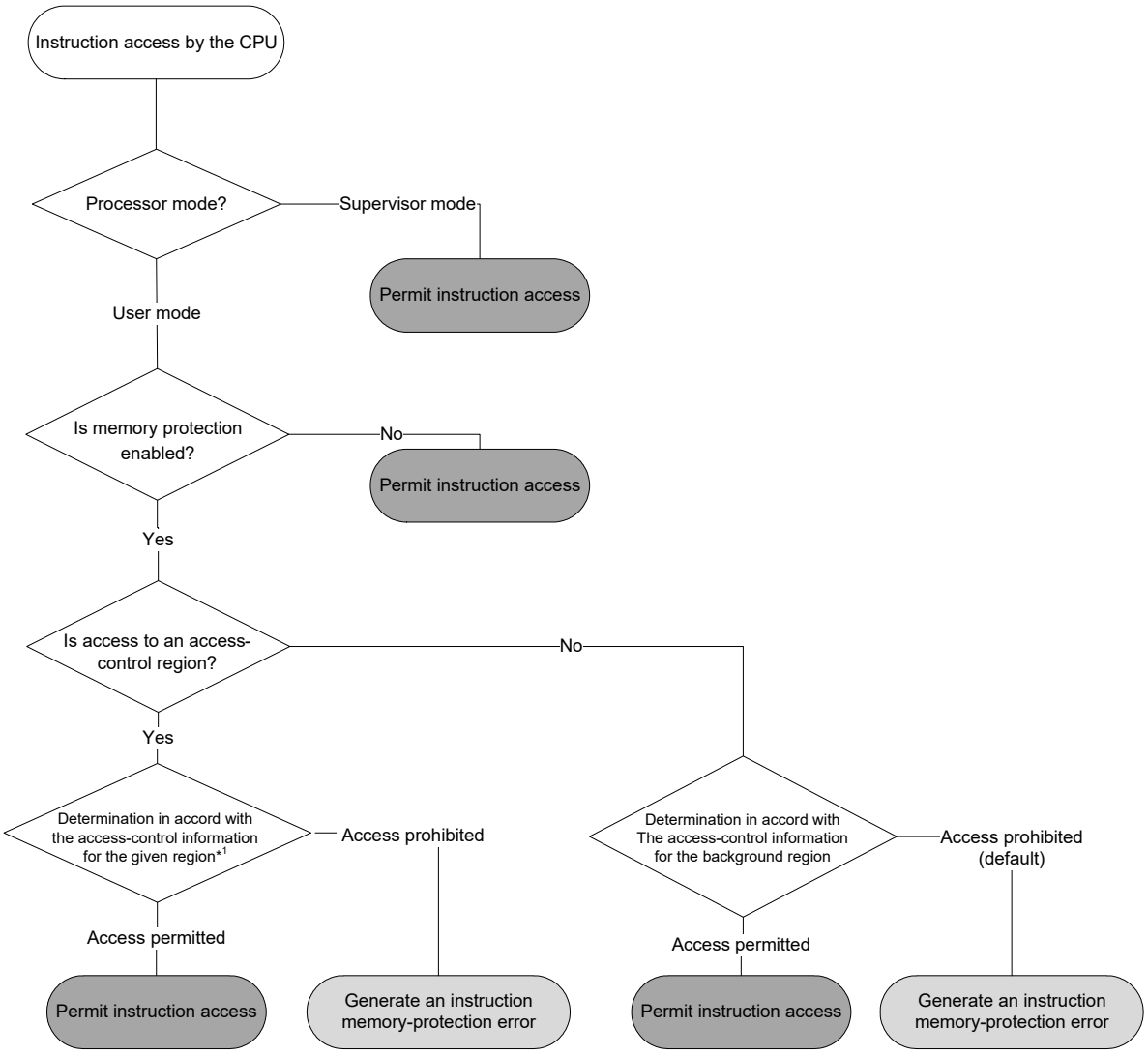
Figure 17.2 shows the flow of determination in the case of data access and Figure 17.3 shows the flow of determination in the case of instruction access.



Note 1. Permission takes priority when the settings lead to overlapping regions overlap and are different for the given regions (this includes the background region).

Figure 17.2 Flow of Determination for Data Access

Downloaded from Arrow.com.



Note 1. Permission takes priority when the settings lead to overlapping regions overlap and are not the same for the given regions (this includes the background region).

Figure 17.3 Flow of Determination for Instruction Access

Downloaded from Arrow.com.

17.4 Procedures for Using Memory Protection

17.4.1 Setting Access-Control Information

Access-control information for the various regions is set in supervisor mode.
Settings for up to eight access-control regions are made in the region-n start page number registers (RSPAGEn) and region-n end page number registers (REPAGEn), where n = 0 to 7.
Settings for the background access-control region are made in the background access-control register (MPBAC).

17.4.2 Enabling Memory Protection

Setting the memory-protection enable (MPEN) bit in the memory-protection enable (MPEN) register to 1 while operation is in supervisor mode enables memory protection.

17.4.3 Transition to User Mode

After updating the registers related to the memory-protection unit, read any of these registers and check that the settings have been made before the transition to user mode.
Either of the methods below can be used for the transition from supervisor mode to user mode.

- Set the processor mode setting (PM) bit in the copy of the processor status word (PSW) saved in the stack area to 1 (the setting for user mode) and then execute an RTE instruction.
- Set the PM bit in the backup processor status word (BPSW) to 1 and then execute an RTFI instruction.

Note: Using an MVTC or POPC instruction to write to the PSW.PM bit is invalid. Use an RTE or RTFI instruction to update the value of the PSW.PM bit.

The memory-protection unit starts checking instruction-execution access and operand access by the CPU on the transition to user mode.

17.4.4 Processing in Response to Memory-Protection Errors

The CPU starts access-exception processing on detection of a violation of protection set up by the access-control information (i.e. a memory-protection error). For details on CPU operations in access-exception processing, refer to section 14, Exception Handling.
To determine whether an instruction memory-protection error or data memory-protection error has been generated, check the values of the instruction memory-protection error generation (IMPER) and data memory-protection error generation (DMPER) bits in the memory-protection error status register (MPESTS) from within the exception-processing routine.
After confirming the type of error, clear the memory-protection error status register (MPESTS) by writing 1 to the status clearing (MPE) bit in the memory-protection error status clearing register (MPECLR).

Downloaded from Arrow.com.

(1) When a data memory-protection error is generated

Access-exception processing by the CPU saves the address of the instruction that led to the memory-protection error on the stack. Furthermore, the address of the operand for which access led to a memory-protection error is stored in the data memory-protection error address register (MPDEA) and the region information for the region where the memory-protection error was generated is stored in the data-hit region register (MHITD).

- Violations of access control in access to valid regions 0 to 7

The data-hit region bits (MHITD.HITD[7:0]) with the same region number as the region where the error occurred is set to 1. In user mode, the logical “or” of the region access-control information for the location where the error occurred is set in the data-hit region access-control bits (MHITD.UHACD[2:0]).

- Violations of access control for the background region, besides access to outside valid regions 0 to 7

The data-hit region bits (MHITD.HITD[7:0]) are set to 0000 0000b. In user mode, the access-control information for the background region is set in the data-hit region access-control bits (MHITD.UHACD[2:0]).

Referring to this information can pinpoint the sources of errors.

(2) When an instruction memory-protection error is generated

Access-exception processing by the CPU saves the address of the instruction that led to the memory-protection error on the stack. Furthermore, the region information for the region where the memory-protection error was generated is stored in the instruction-hit region register (MHITI).

- Violations of access control in access to valid regions 0 to 7

The instruction-hit region bit (MHITD.HITI[7:0]) with the same region number as the region where the error occurred is set to 1. In user mode, the logical “or” of the region access-control information for the location where the error occurred is set in the instruction-hit region access-control bits (MHITI.UHACI[2:0]).

- Violations of access control for the background region, besides access to outside valid regions 0 to 7

The instruction-hit region bits (MHITI.HITI[7:0]) are set to 0000 0000b. In user mode, the access-control information for the background region is set in the instruction-hit region access-control bits (MHITI.UHACI[2:0]).

Referring to this information can pinpoint the sources of errors.

Downloaded from Arrow.com.

18. DMA Controller (DMACA)

This MCU incorporates a 4-channel direct memory access controller (DMAC).
The DMAC module performs data transfers without the CPU. When a DMA transfer request is generated, the DMAC transfers data stored at the transfer source address to the transfer destination address.

18.1 Overview

Table 18.1 lists the specifications of the DMAC, and Figure 18.1 shows a block diagram of the DMAC.

Table 18.1 Specifications of DMAC

Item		Description
Number of channels		4 (DMACm (m = 0 to 3))
Transfer space		512 Mbytes (0000 0000h to 0FFF FFFFh and F000 0000h to FFFF FFFFh excluding reserved areas)
Maximum transfer volume		1M data (Maximum number of transfers in block transfer mode: 1,024 data × 1,024 blocks)
DMA request source		• Activation source selectable for each channel Software trigger Interrupt requests from peripheral modules or trigger input to external interrupt input pins*1
Channel priority		Channel 0 > Channel 1 > Channel 2 > Channel 3 (Channel 0: Highest)
Transfer data	Single data	Bit length: 8, 16, 32 bits
	Block size	Number of data: 1 to 1,024
Transfer mode	Normal transfer mode	• One data transfer by one DMA transfer request • Free running mode (setting in which total number of data transfers is not specified) settable
	Repeat transfer mode	• One data transfer by one DMA transfer request • Program returns to the transfer start address on completion of the repeat size of data transfer specified for the transfer source or destination. • Maximum settable repeat size: 1,024
	Block transfer mode	• One block data transfer by one DMA transfer request • Maximum settable block size: 1,024 data
Selective functions	Extended repeat area function	• Function in which data can be transferred by repeating the address values in the specified range with the upper bit values in the transfer address register fixed • Area of 2 bytes to 128 Mbytes separately settable as extended repeat area for transfer source and destination
Interrupt request	Transfer end interrupt	Generated on completion of transferring data volume specified by the transfer counter.
	Transfer escape end interrupt	Generated when the repeat size of data transfer is completed or the extended repeat area overflows.
Power consumption reduction function		Module stop state can be set.
Event link function		Event link request is generated after one data transfer (for block, after one block transfer).

Note 1. For details on DMAC activation sources, see Table 15.3, Interrupt Vector Table in section 15, Interrupt Controller (ICUb).

Downloaded from Arrow.com.

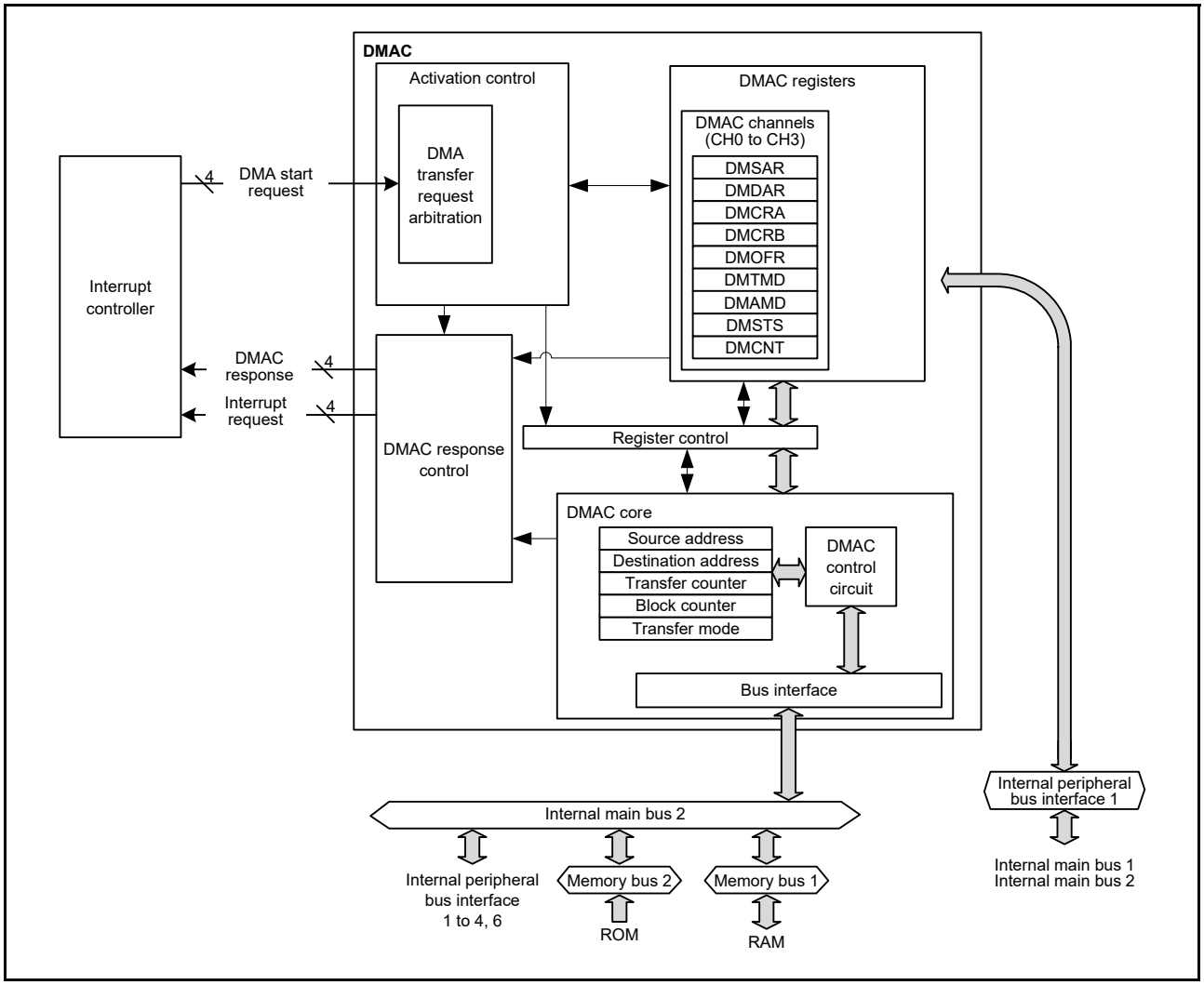


Figure 18.1 Block Diagram of DMAC

Downloaded from Arrow.com.

18.2 Register Descriptions

18.2.1 DMA Source Address Register (DMSAR)

Address(es): DMAC0.DMSAR 0008 2000h, DMAC1.DMSAR 0008 2040h, DMAC2.DMSAR 0008 2080h, DMAC3.DMSAR 0008 20C0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Description	Setting Range	R/W
b31 to b0	Specifies the transfer source start address.	0000 0000h to 0FFF FFFFh (256 Mbytes) F000 0000h to FFFF FFFFh (256 Mbytes)	R/W

Set DMSAR while DMAC activation is disabled (the DMST bit in DMAST = 0) or DMA transfer is disabled (the DTE bit in DMCNT = 0).
Setting bits 31 to 29 is invalid; a value of bit 28 is extended to bits 31 to 29. Reading DMSAR returns the extended value.

18.2.2 DMA Destination Address Register (DMDAR)

Address(es): DMAC0.DMDAR 0008 2004h, DMAC1.DMDAR 0008 2044h, DMAC2.DMDAR 0008 2084h, DMAC3.DMDAR 0008 20C4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Description	Setting Range	R/W
b31 to b0	Specifies the transfer destination start address.	0000 0000h to 0FFF FFFFh (256 Mbytes) F000 0000h to FFFF FFFFh (256 Mbytes)	R/W

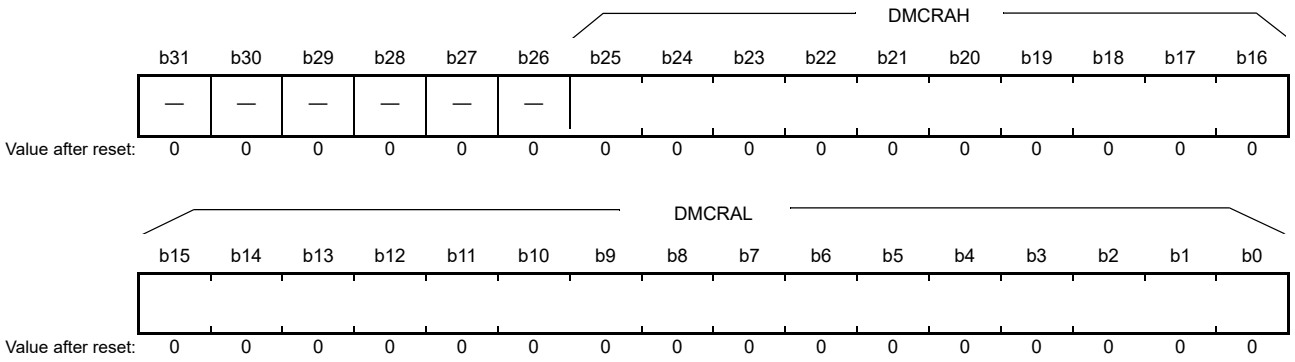
DMDAR specifies the start address of the transfer destination. Set DMDAR while DMAC activation is disabled (the DMST bit in DMAST = 0) or DMA transfer is disabled (the DTE bit in DMCNT = 0).
Setting bits 31 to 29 is invalid; a value of bit 28 is extended to bits 31 to 29. Reading DMDAR returns the extended value.

Downloaded from Arrow.com.

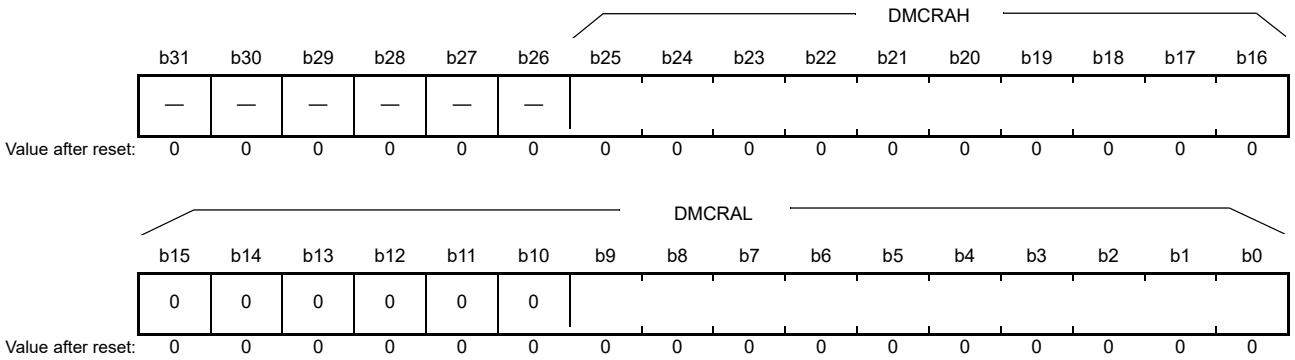
18.2.3 DMA Transfer Count Register (DMCRA)

Address(es): DMAC0.DMCRA 0008 2008h, DMAC1.DMCRA 0008 2048h, DMAC2.DMCRA 0008 2088h, DMAC3.DMCRA 0008 20C8h

- Normal transfer mode



- Repeat transfer mode, block transfer mode



Symbol	Bit Name	Description	R/W
DMCRAL	Lower bits of transfer count	Specifies the number of transfer operations	R/W
DMCRAH	Upper bits of transfer count		R/W

Note: Set the same value for DMCRAH and DMCRAL in repeat transfer mode and block transfer mode.

(1) Normal Transfer Mode (MD[1:0] Bits in DMACm.DMTMD = 00b)

DMCRAL functions as a 16-bit transfer counter.

The number of transfer operations is one when the setting is 0001h, and 65535 when it is FFFFh. The value is decremented by one each time data is transferred.

When the setting is 0000h, no specific number of transfer operations is set; data transfer is performed with the transfer counter stopped (free running mode).

DMCRAH is not used in normal transfer mode. Write 0000h to DMCRAH.

Downloaded from Arrow.com.

(2) Repeat Transfer Mode (MD[1:0] Bits in DMACm.DMTMD = 01b)

DMCRAH specifies the repeat size and DMCRAL functions as a 10-bit transfer counter.
The number of transfer operations is one when the setting is 001h, 1023 when it is 3FFh, and 1024 when it is 000h. In repeat transfer mode, a value in the range of 000h to 3FFh (1 to 1024) can be set for DMCRAH and DMCRAL.
Setting bits 15 to 10 in DMCRAL is invalid. Write 0 to these bits.
The value in DMCRAL is decremented by one each time data is transferred until it reaches 000h, at which the value in DMCRAH is loaded into DMCRAL.

(3) Block Transfer Mode (MD[1:0] Bits in DMACm.DMTMD = 10b)

DMCRAH specifies the block size and DMCRAL functions as a 10-bit block size counter.
The block size is one when the setting is 001h, 1023 when it is 3FFh, and 1024 when it is 000h. In block transfer mode, a value in the range of 000h to 3FFh can be set for DMCRAH and DMCRAL.
Setting bits 15 to 10 in DMCRAL is invalid. Write 0 to these bits.
The value in DMCRAL is decremented by one each time data is transferred until it reaches 000h, at which the value in DMCRAH is loaded into DMCRAL.

Downloaded from Arrow.com.

18.2.4 DMA Block Transfer Count Register (DMCRB)

Address(es): DMAC0.DMCRB 0008 200Ch, DMAC1.DMCRB 0008 204Ch, DMAC2.DMCRB 0008 208Ch, DMAC3.DMCRB 0008 20CCh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Description	Setting Range	R/W
b9 to b0	Specifies the number of block transfer operations or repeat transfer operations.	001h to 3FFh (1 to 1023) 000h (1024)	R/W
b15 to b10	Reserved	These bits are read as 0. The write value should be 0.	R/W

DMCRB specifies the number of block transfer operations and repeat transfer operations in block and repeat transfer mode, respectively.

The number of transfer operations is one when the setting is 001h, 1023 when it is 3FFh, and 1024 when it is 000h.

In repeat transfer mode, the value is decremented by one when the final data of one repeat size is transferred.

In block transfer mode, the value is decremented by one when the final data of one block size is transferred.

In normal transfer mode, DMCRB is not used. The setting is invalid.

Downloaded from Arrow.com.

18.2.5 DMA Transfer Mode Register (DMTMD)

Address(es): DMAC0.DMTMD 0008 2010h, DMAC1.DMTMD 0008 2050h, DMAC2.DMTMD 0008 2090h, DMAC3.DMTMD 0008 20D0h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MD[1:0]		DTS[1:0]		—	—	SZ[1:0]		—	—	—	—	—	—	DCTG[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	DCTG[1:0]	DMA Request Source Select	b1 b0 0 0: Software 0 1: Interrupts*1 from peripheral modules or external interrupt input pins 1 0: Setting prohibited 1 1: Setting prohibited	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b9, b8	SZ[1:0]	Transfer Data Size Select	b9 b8 0 0: 8 bits 0 1: 16 bits 1 0: 32 bits 1 1: Setting prohibited	R/W
b11, b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b13, b12	DTS[1:0]	Repeat Area Select	b13 b12 0 0: The destination is specified as the repeat area or block area. 0 1: The source is specified as the repeat area or block area. 1 0: The repeat area or block area is not specified. 1 1: Setting prohibited	R/W
b15, b14	MD[1:0]	Transfer Mode Select	b15 b14 0 0: Normal transfer 0 1: Repeat transfer 1 0: Block transfer 1 1: Setting prohibited	R/W

Note 1. DMAC activation source is selected using the DMRSRm registers of the ICU. For details on DMAC activation sources, see Table 15.3, Interrupt Vector Table in section 15, Interrupt Controller (ICUb).

DTS[1:0] Bits (Repeat Area Select)

DTS[1:0] select either the source or destination as the repeat area in repeat or block transfer mode. In normal transfer mode, setting these bits is invalid.

Downloaded from Arrow.com.

18.2.6 DMA Interrupt Setting Register (DMINT)

Address(es): DMAC0.DMINT 0008 2013h, DMAC1.DMINT 0008 2053h, DMAC2.DMINT 0008 2093h, DMAC3.DMINT 0008 20D3h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	DTIE	ESIE	RPTIE	SARIE	DARIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DARIE	Destination Address Extended Repeat Area Overflow Interrupt Enable	0: Disables an interrupt request for an extended repeat area overflow on the destination address 1: Enables an interrupt request for an extended repeat area overflow on the destination address	R/W
b1	SARIE	Source Address Extended Repeat Area Overflow Interrupt Enable	0: Disables an interrupt request for an extended repeat area overflow on the source address 1: Enables an interrupt request for an extended repeat area overflow on the source address	R/W
b2	RPTIE	Repeat Size End Interrupt Enable	0: Disables the repeat size end interrupt request. 1: Enables the repeat size end interrupt request.	R/W
b3	ESIE	Transfer Escape End Interrupt Enable	0: Disables the transfer escape end interrupt request. 1: Enables the transfer escape end interrupt request.	R/W
b4	DTIE	Transfer End Interrupt Enable	0: Disables the transfer end interrupt request. 1: Enables the transfer end interrupt request.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

DARIE Bit (Destination Address Extended Repeat Area Overflow Interrupt Enable)

When an extended repeat area overflow on the destination address occurs while this bit is set to 1, the DTE bit in DMCNT is cleared to 0. At the same time, the ESIF flag in DMSTS is set to 1 to indicate that an interrupt by an extended repeat area overflow on the destination address is requested.

When block transfer mode is used with the extended repeat area function, an interrupt is requested after completion of a 1-block size transfer. When setting 1 in the DTE bit in DMACm.DMCNT of the channel for which a transfer has been stopped, the transfer is resumed from the state when the transfer is stopped.

When the extended repeat area is not specified for the destination address, this bit is ignored.

SARIE Bit (Source Address Extended Repeat Area Overflow Interrupt Enable)

When an extended repeat area overflow on the source address occurs while this bit is set to 1, the DTE bit in DMCNT is cleared to 0. At the same time, the ESIF flag in DMSTS is set to 1 to indicate that an interrupt by an extended repeat area overflow on the source address is requested.

When block transfer mode is used with the extended repeat area function, an interrupt is requested after completion of a 1-block size transfer. When setting 1 in the DTE bit in DMACm.DMCNT of the channel for which a transfer has been stopped, the transfer is resumed from the state when the transfer is stopped.

When the extended repeat area is not specified for the source address, this bit is ignored.

Downloaded from Arrow.com.

RPTIE Bit (Repeat Size End Interrupt Enable)

When this bit is set to 1 in repeat transfer mode, the DTE bit in DMCNT is cleared to 0 after completion of a 1-repeat size data transfer. At the same time, the ESIF flag in DMSTS is set to 1 to indicate that the repeat size end interrupt request has been generated. The repeat size end interrupt request can be generated even when the DTS[1:0] bits in DMTMD are 10b (= repeat area or block area is not specified).

When this bit is set to 1 in block transfer mode, the DTE bit in DMCNT is cleared to 0 after completion of a 1-block data transfer in the same way as repeat transfer mode. At the same time, the ESIF flag in DMSTS is set to 1 to indicate that the repeat size end interrupt request has been generated. The repeat size end interrupt request can be generated even when the DTS[1:0] bits in DMTMD are 10b (= repeat area or block area is not specified).

ESIE Bit (Transfer Escape End Interrupt Enable)

This bit enables or disables the transfer escape end interrupt requests (repeat size end interrupt request and extended repeat area overflow interrupt request) that are generated during DMA transfer.

The transfer escape end interrupt is generated when the ESIF flag in DMSTS is set to 1 with this bit set to 1. The transfer escape end interrupt is cleared by clearing this bit or the ESIF flag in DMSTS to 0.

DTIE Bit (Transfer End Interrupt Enable)

This bit enables or disables the transfer end interrupt request to be generated on completion of a specified number of data transfers.

The transfer end interrupt is generated when the DTIF bit in DMSTS is set to 1 with this bit set to 1. The transfer end interrupt is cleared by clearing this bit or the DTIF bit in DMSTS to 0.

Downloaded from Arrow.com.

18.2.7 DMA Address Mode Register (DMAMD)

Address(es): DMAC0.DMAMD 0008 2014h, DMAC1.DMAMD 0008 2054h, DMAC2.DMAMD 0008 2094h, DMAC3.DMAMD 0008 20D4h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SM[1:0]		—	SARA[4:0]					DM[1:0]		—	DARA[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	DARA[4:0]	Destination Address Extended Repeat Area	Specifies the extended repeat area on the destination address. For details on the settings, see Table 18.2.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7, b6	DM[1:0]	Destination Address Update Mode	b7 b6 0 0: Destination address is fixed. 0 1: Offset addition*1 1 0: Destination address is incremented. 1 1: Destination address is decremented.	R/W
b12 to b8	SARA[4:0]	Source Address Extended Repeat Area	Specifies the extended repeat area on the source address. For details on the settings, see Table 18.2.	R/W
b13	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b15, b14	SM[1:0]	Source Address Update Mode	b15 b14 0 0: Source address is fixed. 0 1: Offset addition*1 1 0: Source address is incremented. 1 1: Source address is decremented.	R/W

Note 1. Offset addition can be specified only for DMAC0.

DARA[4:0] Bits (Destination Address Extended Repeat Area)

These bits specify the extended repeat area on the destination address. The extended repeat area function is realized by updating the specified lower address bits with the remaining upper address bits fixed. The size of the extended repeat area can be any power of two between 2 bytes and 128 Mbytes.

When the lower address overflows the extended repeat area by address increment, the start address of the extended repeat area is set. Similarly, when the lower address underflows the extended repeat area by address decrement, the end address of the extended repeat area is set.

When the repeat area or block area is specified as a transfer destination, do not specify the extended repeat area on the destination address. When repeat transfer or block transfer is selected, or when DMACm.DMTMD.DTS[1:0] = 00b (the transfer destination is specified as the repeat area or block area), write 00000b in the DARA[4:0] bits.

An interrupt can be requested when an overflow or underflow occurs in the extended repeat area with the DARIE bit in DMINT set to 1. Table 18.2 lists the settings and the corresponding extended repeat areas.

DM[1:0] Bits (Destination Address Update Mode)

These bits select the mode of updating the destination address.

When increment is selected and the SZ[1:0] bits in DMTMD are set to 00b, 01b, and 10b, the destination address is incremented by 1, 2, and 4, respectively.

When decrement is selected and the SZ[1:0] bits in DMTMD are set to 00b, 01b, and 10b, the destination address is decremented by 1, 2, and 4, respectively.

When offset addition is selected, the offset specified by the DMAC0.DMOFR register is added to the address.

Offset addition can be specified only for DMAC0.

Downloaded from Arrow.com.

SARA[4:0] Bits (Source Address Extended Repeat Area)

These bits specify the extended repeat area on the source address. The extended repeat area function is realized by updating the specified lower address bits with the remaining upper address bits fixed. The size of the extended repeat area can be any power of two between 21 (2 bytes) and 217 (128 Mbytes).

When the lower address overflows the extended repeat area by address increment, the start address of the extended repeat area is set. Similarly, when the lower address underflows the extended repeat area by address decrement, the end address of the extended repeat area is set.

When the repeat area or block area is specified as a transfer source, do not specify the extended repeat area on the source address. When repeat transfer or block transfer is selected, or when DMACm.DMTMD.DTS[1:0] = 01b (the transfer source is specified as the repeat area or block area), write 00000b in the SARA[4:0] bits.

An interrupt can be requested when an overflow or underflow occurs in the extended repeat area with the SARIE bit in DMINT set to 1. Table 18.2 lists the settings and the corresponding extended repeat areas.

SM[1:0] Bit (Source Address Update Mode)

These bits select the mode of updating the source address.

When increment is selected and the SZ[1:0] bits in DMTMD are set to 00b, 01b, and 10b, the source address is incremented by 1, 2, and 4, respectively.

When decrement is selected and the SZ[1:0] bits in DMTMD are set to 00b, 01b, and 10b, the source address is decremented by 1, 2, and 4, respectively.

When offset addition is selected, the offset specified by the DMAC0.DMOFR register is added to the address. Offset addition can be specified only for DMAC0.

Downloaded from Arrow.com.

Table 18.2 SARA[4:0] or DARA[4:0] Settings and Corresponding Repeat Areas

SARA4 to SARA0 or DARA4 to DARA0	Extended Repeat Area
00000b	Not specified
00001b	2 bytes specified as extended repeat area by the lower 1 bit of the address
00010b	4 bytes specified as extended repeat area by the lower 2 bits of the address
00011b	8 bytes specified as extended repeat area by the lower 3 bits of the address
00100b	16 bytes specified as extended repeat area by the lower 4 bits of the address
00101b	32 bytes specified as extended repeat area by the lower 5 bits of the address
00110b	64 bytes specified as extended repeat area by the lower 6 bits of the address
00111b	128 bytes specified as extended repeat area by the lower 7 bits of the address
01000b	256 bytes specified as extended repeat area by the lower 8 bits of the address
01001b	512 bytes specified as extended repeat area by the lower 9 bits of the address
01010b	1 Kbyte specified as extended repeat area by the lower 10 bits of the address
01011b	2 Kbytes specified as extended repeat area by the lower 11 bits of the address
01100b	4 Kbytes specified as extended repeat area by the lower 12 bits of the address
01101b	8 Kbytes specified as extended repeat area by the lower 13 bits of the address
01110b	16 Kbytes specified as extended repeat area by the lower 14 bits of the address
01111b	32 Kbytes specified as extended repeat area by the lower 15 bits of the address
10000b	64 Kbytes specified as extended repeat area by the lower 16 bits of the address
10001b	128 Kbytes specified as extended repeat area by the lower 17 bits of the address
10010b	256 Kbytes specified as extended repeat area by the lower 18 bits of the address
10011b	512 Kbytes specified as extended repeat area by the lower 19 bits of the address
10100b	1 Mbyte specified as extended repeat area by the lower 20 bits of the address
10101b	2 Mbytes specified as extended repeat area by the lower 21 bits of the address
10110b	4 Mbytes specified as extended repeat area by the lower 22 bits of the address
10111b	8 Mbytes specified as extended repeat area by the lower 23 bits of the address
11000b	16 Mbytes specified as extended repeat area by the lower 24 bits of the address
11001b	32 Mbytes specified as extended repeat area by the lower 25 bits of the address
11010b	64 Mbytes specified as extended repeat area by the lower 26 bits of the address
11011b	128 Mbytes specified as extended repeat area by the lower 27 bits of the address
11100b to 11111b	Setting prohibited.

Downloaded from Arrow.com.

18.2.8 DMA Offset Register (DMOFR)

Address(es): DMAC0.DMOFR 0008 2018h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Description	Setting Range	R/W
b31 to b0	Specifies the offset when offset addition is selected as the address update mode for transfer source or destination.	0000 0000h to 00FF FFFFh (0 bytes to (16 M – 1) bytes) FF00 0000h to FFFF FFFFh (–16 Mbytes to –1 byte)	R/W

Write to this register while the DMAC operation is stopped or DMA transfer is disabled (not during data transfer).
Setting bits 31 to 25 is invalid; a value of bit 24 is extended to bits 31 to 25. Reading DMOFR returns the extended value.

Downloaded from Arrow.com.

18.2.9 DMA Transfer Enable Register (DMCNT)

Address(es): DMAC0.DMCNT 0008 201Ch, DMAC1.DMCNT 0008 205Ch, DMAC2.DMCNT 0008 209Ch, DMAC3.DMCNT 0008 20DCh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	DTE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DTE	DMA Transfer Enable	0: Disables DMA transfer. 1: Enables DMA transfer.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

DTE Bit (DMA Transfer Enable)

When the DMST bit in DMAST is set to 1 (DMAC activation is enabled) and this bit is set to 1 (DMA transfer is enabled), DMA transfer can be started for the corresponding channel.

[Setting condition]

- When 1 is written to this bit.

[Clearing conditions]

- When 0 is written to this bit.
- When the specified total volume of data transfer is completed.
- When DMA transfer is stopped by the repeat size end interrupt.
- When DMA transfer is stopped by the extended repeat area overflow interrupt.

Downloaded from Arrow.com.

18.2.10 DMA Software Start Register (DMREQ)

Address(es): DMAC0.DMREQ 0008 201Dh, DMAC1.DMREQ 0008 205Dh, DMAC2.DMREQ 0008 209Dh, DMAC3.DMREQ 0008 20DDh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	CLRS	—	—	—	SWREQ
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SWREQ	DMA Software Start	0: DMA transfer is not requested. 1: DMA transfer is requested.	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	CLRS	DMA Software Start Bit Auto Clear Select	0: SWREQ bit is cleared after DMA transfer is started by software. 1: SWREQ bit is not cleared after DMA transfer is started by software.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

SWREQ Bit (DMA Software Start)

When 1 is written to this bit, a DMA transfer request is generated. After DMA transfer is started in response to the request, this bit is cleared to 0 if the CLRS bit is set to 0. This bit is not cleared to 0 while the CLRS bit is set to 1. In this case, a DMA transfer request can be issued again after completion of a transfer.

Note that, however, setting this bit is valid and DMA transfer by software is enabled only when the DCTG[1:0] bits in DMTMD are set to 00b (DMA activation source is software).

Setting this bit is invalid when the DCTG[1:0] bits in DMTMD are set to a value other than 00b.

To start DMA transfer by software with the CLRS bit being 0, ensure that the SWREQ bit is 0, and then write 1 to the SWREQ bit.

[Setting condition]

- When 1 is written to this bit.

[Clearing conditions]

- When a DMA transfer request by software is accepted and DMA transfer is started while the CLRS bit is set to 0 (the SWREQ bit is cleared after DMA transfer is started by software).
- When 0 is written to this bit.

CLRS Bit (DMA Software Start Bit Auto Clear Select)

This bit specifies whether to clear the SWREQ bit to 0 after DMA transfer is started in response to the DMA transfer request generated by setting the SWREQ bit to 1. With this bit set to 0, the SWREQ bit is cleared to 0 after DMA transfer is started. With this bit set to 1, the SWREQ bit is not cleared to 0. In this case, a DMA transfer request can be issued again after completion of a transfer.

Downloaded from Arrow.com.

18.2.11 DMA Status Register (DMSTS)

Address(es): DMAC0.DMSTS 0008 201Eh, DMAC1.DMSTS 0008 205Eh, DMAC2.DMSTS 0008 209Eh, DMAC3.DMSTS 0008 20DEh

	b7	b6	b5	b4	b3	b2	b1	b0
	ACT	—	—	DTIF	—	—	—	ESIF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ESIF	Transfer Escape End Interrupt Flag	0: A transfer escape end interrupt has not been generated. 1: A transfer escape end interrupt has been generated.	R/W*1
b3 to b1	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b4	DTIF	Transfer End Interrupt Flag	0: A transfer end interrupt has not been generated. 1: A transfer end interrupt has been generated.	R/W*1
b6, b5	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b7	ACT	DMA Active Flag	0: DMAC operation is suspended. 1: DMAC is operating.	R

Note 1. Only 0 can be written to clear the flag.

ESIF Flag (Transfer Escape End Interrupt Flag)

This flag indicates that the transfer escape end interrupt has been generated.

[Setting conditions]

- When 1-repeat size data transfer is completed in repeat transfer mode with the RPTIE bit in DMINT set to 1.
- When 1-block data transfer is completed in block transfer mode with the RPTIE bit in DMINT set to 1.
- When an extended repeat area overflow on the source address occurs while the SARIE bit in DMINT is set to 1 and the SARA[4:0] bits in DMAMD are set to a value other than 00000b (extended repeat area is specified on the transfer source address)
- When an extended repeat area overflow on the destination address occurs while the DARIE bit in DMINT is set to 1 and the DARA[4:0] bits in DMAMD are set to a value other than 00000b (extended repeat area is specified on the transfer destination address)

[Clearing conditions]

- When 0 is written to this bit.
- When 1 is written to the DTE bit in DMCNT.

Downloaded from Arrow.com.

DTIF Flag (Transfer End Interrupt Flag)

This flag indicates that the transfer end interrupt has been generated.

[Setting conditions]

- When the specified number of unit-transfers are completed in normal transfer mode (the value of DMCRAL becoming 0 on completion of transfer)
- When the specified number of repeat transfer operations are completed in repeat transfer mode (the value of DMCRB becoming 0 on completion of transfer))
- When the specified number of blocks have been transferred in block transfer mode (the value of DMCRB becoming 0 on completion of transfer)

[Clearing conditions]

- When 0 is written to this bit
- When 1 is written to the DTE bit in DMCNT

ACT Flag (DMA Active Flag)

This flag indicates whether the DMAC is in the idle or active state.

[Setting condition]

- When the DMAC starts data transfer operation

[Clearing condition]

- When data transfer in response to one transfer request is completed

Downloaded from Arrow.com.

18.2.12 DMA Activation Source Flag Control Register (DMCSL)

Address(es): DMAC0.DMCSL 0008 201Fh, DMAC1.DMCSL 0008 205Fh, DMAC2.DMCSL 0008 209Fh, DMAC3.DMCSL 0008 20DFh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	DISEL
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DISEL	Interrupt Select	0: At the beginning of transfer, clear the interrupt flag of the activation source to 0. 1: At the end of transfer, the interrupt flag of the activation source issues an interrupt to the CPU.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

DISEL Bit (Interrupt Select)

This bit selects whether the interrupt flag of the activation source of the DMAC is cleared to 0 or issues an interrupt to the CPU, at the beginning of transfer.

When DMTMD.DCTG[1:0] = 00b (activation by software), the setting of the DISEL bit does not affect the operation.

Downloaded from Arrow.com.

18.2.13 DMA Module Activation Register (DMAST)

Address(es): 0008 2200h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	DMST
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DMST	DMAC Operation Enable	0: DMAC activation is disabled. 1: DMAC activation is enabled.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

DMST Bit (DMAC Operation Enable)

When this bit is set to 1, DMAC activation is enabled for all channels.

When 1 is written to the DMACm.DMCNT.DTE bit (DMA transfer is enabled) of multiple channels and then this bit is set to 1 (DMAC activation is enabled), the corresponding multiple channels can be placed in the transfer request acceptable state at the same time.

When the DMST bit is cleared to 0 during DMA transfer, DMA transfer is suspended after completion of the current data transfer corresponding to a single transfer request. DMA transfer is resumed by setting the DMST bit to 1 again.

[Setting condition]

- When 1 is written to this bit

[Clearing condition]

- When 0 is written to this bit

Downloaded from Arrow.com.

18.3 Operation

18.3.1 Transfer Mode

(1) Normal Transfer Mode

In normal transfer mode, one data is transferred by one transfer request. A maximum of 65535 can be set as the number of transfer operations using the DMCRAL of DMACm. When these bits are set to 0000h, no specific number of transfer operations is set; data transfer is performed with the transfer counter stopped (free running mode). Setting DMCRB of DMACm is invalid in normal transfer mode. Except in free running mode, a transfer end interrupt request can be generated after completion of the specified number of transfer operations.
Table 18.3 summarizes the register update operation in normal transfer mode, and Figure 18.2 shows the operation in normal transfer mode.

Table 18.3 Register Update Operation in Normal Transfer Mode

Register	Function	Update Operation after Completion of a Transfer by One Transfer Request
DMACm.DMSAR	Transfer source address	Increment/decrement/fixed/offset addition*1
DMACm.DMDAR	Transfer destination address	Increment/decrement/fixed/offset addition*1
DMACm.DMCRAL	Transfer count	Decremented by one/not updated (in free running mode)
DMACm.DMCRAH	—	Not updated (Not used in normal transfer mode)
DMACm.DMCRB	—	Not updated (Not used in normal transfer mode)

Note 1. Offset addition can be specified only for DMAC0.

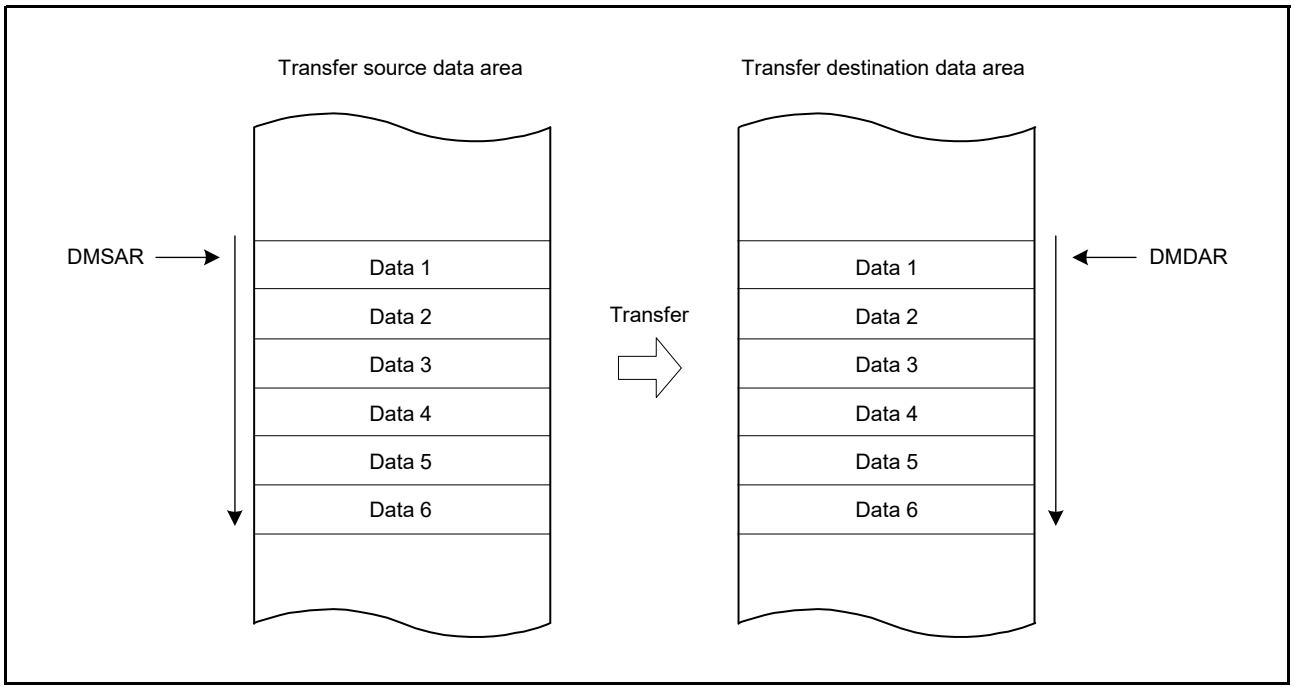


Figure 18.2 Operation in Normal Transfer Mode

Downloaded from Arrow.com.

(2) Repeat Transfer Mode

In repeat transfer mode, one data is transferred by one transfer request.
A maximum of 1K data can be set as a total repeat transfer size using DMCRA of the DMACm.
A maximum of 1K can be set as the number of repeat transfer operations using DMCRB of the DMACm; therefore, a maximum of 1M data (1K data × 1K count of repeat transfer operations) can be set as a total data transfer size.
Either the transfer source or transfer destination can be specified as a repeat area. When transfer of the repeat size data is completed, the address of the specified repeat area (DMSAR or DMDAR of the DMACm) returns to the transfer start address. When data of the specified repeat size has all been transferred in repeat transfer mode, DMA transfer can be stopped and the repeat size end interrupt can be requested. DMA transfer can be resumed by writing 1 to the DTE bit in DMCNT of DMACm in the repeat size end interrupt handling.
A transfer end interrupt request can be generated after completion of the specified number of repeat transfer operations.
Table 18.4 summarizes the register update operation in repeat transfer mode, and Figure 18.3 shows the operation in repeat transfer mode.

Table 18.4 Register Update Operation in Repeat Transfer Mode

		Update Operation after Completion of a Transfer by One Transfer Request	
Register	**Function**	**When DMACm.DMCRAL is not 1**	**When DMACm.DMCRAL is 1 (Transfer of the Last Data in Repeat Size)**
DMACm.DMSAR	Transfer source address	Increment/decrement/fixed/offset addition*1	• DMACm.DMTMD.DTS[1:0] = 00b Increment/decrement/fixed/offset addition*1 • DMACm.DMTMD.DTS[1:0] = 01b Initial value of DMACm.DMSAR • DMACm.DMTMD.DTS[1:0] = 10b Increment/decrement/fixed/offset addition*1
DMACm.DMDAR	Transfer destination address	Increment/decrement/fixed/offset addition*1	• DMACm.DMTMD.DTS[1:0] = 00b Initial value of DMACm.DMDAR • DMACm.DMTMD.DTS[1:0] = 01b Increment/decrement/fixed/offset addition*1 • DMACm.DMTMD.DTS[1:0] = 10b Increment/decrement/fixed/offset addition*1
DMACm.DMCRAH	Repeat size	Not updated	Not updated
DMACm.DMCRAL	Transfer count	Decremented by one	DMACm.DMCRAH
DMACm.DMCRB	Count of repeat transfer operations	Not updated	Decremented by one

Note 1. Offset addition can be specified only for DMAC0.

Downloaded from Arrow.com.

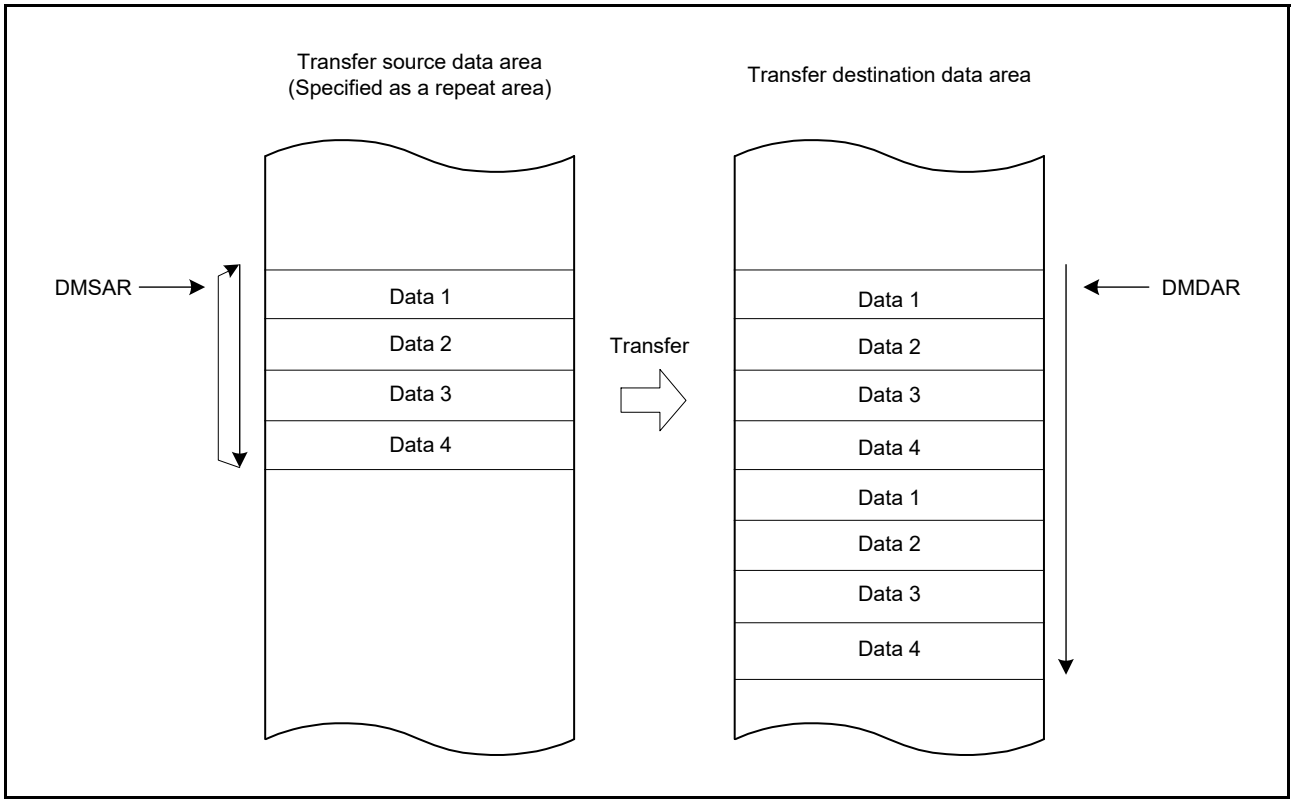


Figure 18.3 Operation in Repeat Transfer Mode

Downloaded from Arrow.com.

(3) Block Transfer Mode

In block transfer mode, a single block data is transferred by one transfer request.
A maximum of 1K data can be set as a total block transfer size using DMCRA of the DMACm.
A maximum of 1K can be set as the number of block transfer operations using DMCRB of the DMACm; therefore, a maximum of 1M data (1K data × 1K count of block transfer operations) can be set as a total data transfer size.
Either the transfer source or transfer destination can be specified as a block area. When transfer of a single block data is completed, the address of the specified block area (DMSAR or DMDAR of the DMACm) returns to the transfer start address. When a single block data has all been transferred in block transfer mode, DMA transfer can be stopped and the repeat size end interrupt can be requested. DMA transfer can be resumed by writing 1 to the DTE bit in DMCNT of DMACm in the repeat size end interrupt handling.
Transfer end interrupt request can be generated after completion of the specified number of block transfer operations.
Table 18.5 summarizes the register update operation in block transfer mode, and Figure 18.4 shows the operation in block transfer mode.

Table 18.5 Register Update Operation in Block Transfer Mode

Register	Function	Update Operation after Completion of Single-Block Transfer by One Transfer Request
DMACm.DMSAR	Transfer source address	• DMACm.DMTMD.DTS[1:0] = 00b Increment/decrement/fixed/offset addition*1 • DMACm.DMTMD.DTS[1:0] = 01b Initial value of DMACm.DMSAR • DMACm.DMTMD.DTS[1:0] = 10b Increment/decrement/fixed/offset addition*1
DMACm.DMDAR	Transfer destination address	• DMACm.DMTMD.DTS[1:0] = 00 b Initial value of DMACm.DMDAR • DMACm.DMTMD.DTS[1:0] = 01 b Increment/decrement/fixed/offset addition*1 • DMACm.DMTMD.DTS[1:0] = 10 b Increment/decrement/fixed/offset addition*1
DMACm.DMCRAH	Block size	Not updated
DMACm.DMCRAL	Transfer count	DMACm.DMCRAH
DMACm.DMCRB	Count of block transfer operations	Decremented by one

Note 1. Offset addition can be specified only for DMAC0.

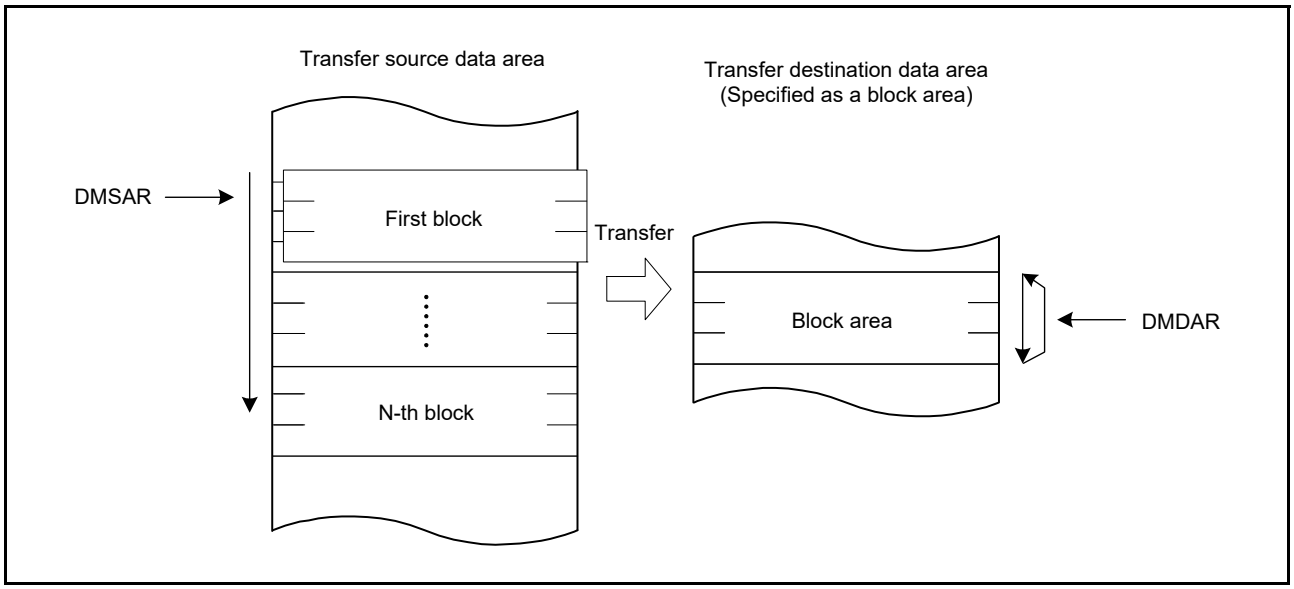


Figure 18.4 Operation in Block Transfer Mode

Downloaded from Arrow.com.

18.3.2 Extended Repeat Area Function

The DMAC supports a function to specify the extended repeat areas on the transfer source and destination addresses. With the extended repeat areas set, the address registers repeatedly indicate the addresses of the specified extended repeat areas.

The extended repeat areas can be specified separately to the transfer source address register (DMSAR) and transfer destination address register (DMDAR) of DMACm.

The extended repeat area on the source address is specified by the SARA[4:0] bits in DMAMD of DMACm. The extended repeat area on the destination address is specified by the DARA[4:0] bits in DMAMD of DMACm. The size can be specified separately for the source and destination sides.

However, the area (of transfer source or transfer destination) which is specified as the repeat area or block area should not be specified as the extended repeat area.

When the address register value reaches the end address of the extended repeat area and the extended repeat area overflows, DMA transfer is stopped and an interrupt by an extended repeat area overflow can be requested. When an overflow occurs in the extended repeat area on the transfer source while the SARIE bit in DMINT of DMACm is set to 1, the ESIF flag in DMSTS of DMACm is set to 1 and the DTE bit in DMCNT of DMACm is cleared to 0 to stop DMA transfer. At this time, if the ESIE bit in DMINT of DMACm is set to 1, an interrupt by an extended repeat area overflow is requested. When the DARIE bit in DMINT of DMACm is set to 1, the destination address register becomes a target to apply the function. DMA transfer can be resumed by writing 1 to the DTE bit in DMCNT of DMACm in the interrupt handling.

Figure 18.5 shows an example of the extended repeat area operation.

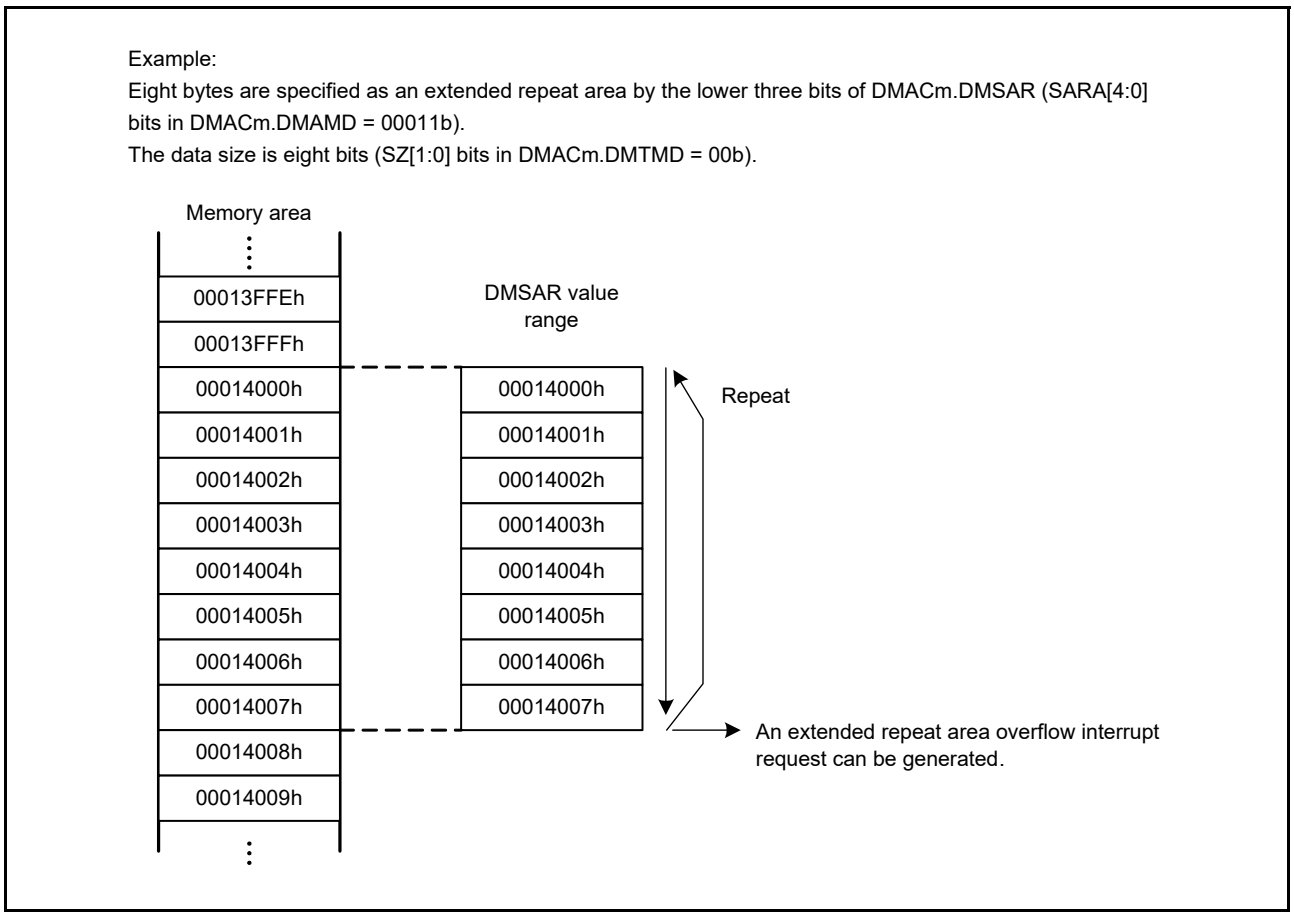


Figure 18.5 Example of Extended Repeat Area Operation

Downloaded from Arrow.com.

When an interrupt by an extended repeat area overflow is used in block transfer mode, the following should be taken into consideration.

When a transfer is stopped by an interrupt by an extended repeat area overflow, the address register must be set so that the block size is a power of 2 or the block size boundary is aligned with the extended repeat area boundary. When an overflow on the extended repeat area occurs during a transfer of one block, the interrupt by the overflow is suspended until transfer of the block is completed, and the transfer overruns.

Figure 18.6 shows an example when the extended repeat area function is used in block transfer mode.

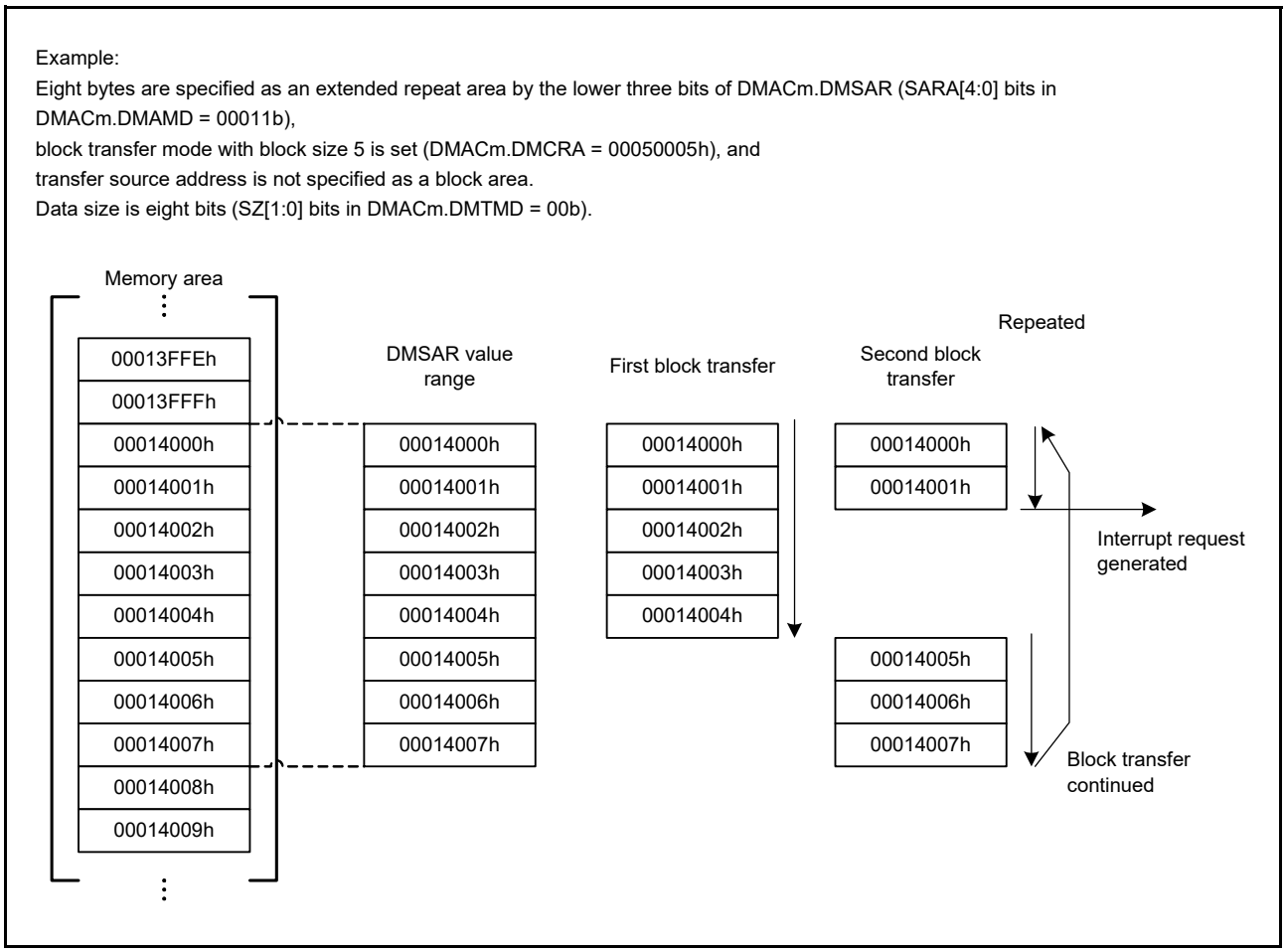


Figure 18.6 Example of Extended Repeat Area Function in Block Transfer Mode

Downloaded from Arrow.com.

18.3.3 Address Update Function Using Offset

The source and destination addresses can be updated by fixing, increment, decrement, or offset addition. When the offset addition is selected, the offset specified by the DMA offset register (DMOFR of DMAC0) is added to the address every time the DMAC performs one data transfer. This function realizes a data transfer where addresses are allocated to separated areas.

Offset subtraction can also be realized by setting a negative value in DMOFR of DMAC0. In this case, the negative value must be 2's complement.

Address update function using offset can be specified only for the DMAC0 channel.

Table 18.6 shows the address update method in each address update mode.

Table 18.6 Address Update Method in Each Address Update Mode

Address Update Mode	Settings of DMACm.DMAMD.SM[1:0] and DMACm.DMAMD.DM[1:0] for Address Update Modes	Address Update Method (for Different SZ[1:0] Settings in DMTMD of DMACm)		
		SZ[1:0] = 00b	SZ[1:0] = 01b	SZ[1:0] = 10b
Address fixed	00b	Fixed		
Offset addition	01b	+DMACm.DMOFR*1		
Increment	10b	+1	+2	+4
Decrement	11b	–1	–2	–4

Note 1. When setting a negative value in the DMA offset register, the value must be 2's complement. The 2's complement is obtained by the following formula.
2's complement of a negative offset value = ~ (offset) + 1 (~: bit inversion)

Downloaded from Arrow.com.

(1) Basic Transfer Using Offset Addition

Figure 18.7 shows an example of address updating using offset addition.

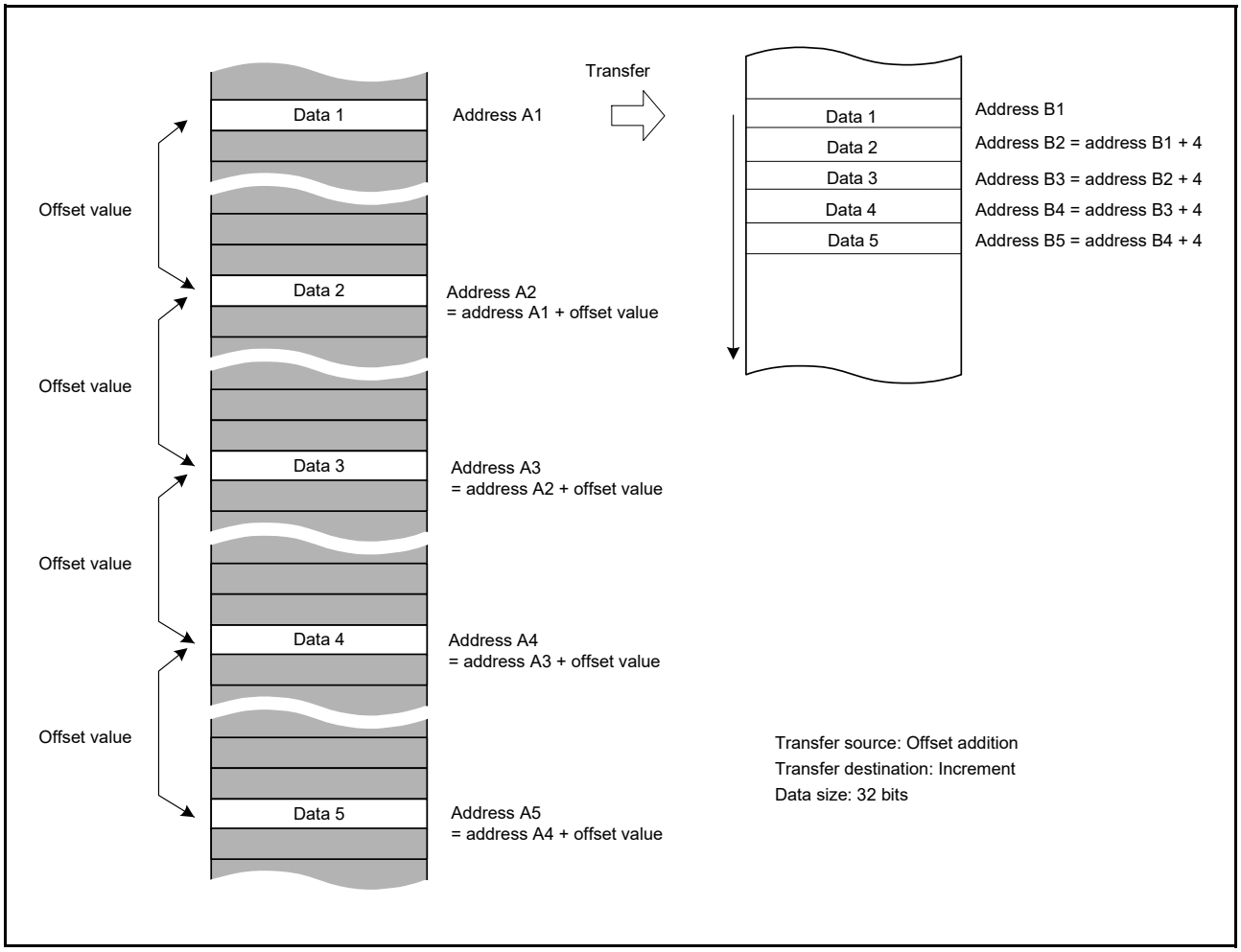


Figure 18.7 Example of Address Updating by Offset Addition

In Figure 18.7, the transfer data is 32 bits long, and offset addition and increment are set as the transfer source address update mode and transfer destination address update mode, respectively. The second and subsequent data is each read from the transfer source address obtained by adding the offset value to the previous address. The data read from the addresses at the specified intervals is written to the continuous locations on the destination.

Downloaded from Arrow.com.

(2) Example of XY Conversion Using Offset Addition

Figure 18.8 shows the XY conversion using offset addition in repeat transfer mode.
Settings are as follows:

- DMAC0.DMAMD: Transfer source address update mode: Offset addition
- DMAC0.DMAMD: Transfer destination address update mode: Destination address is incremented.
- DMAC0.DMTMD: Transfer data size select: 32 bits
- DMAC0.DMTMD: Transfer mode select: Repeat transfer
- DMAC0.DMTMD: Repeat area select: The source is specified as the repeat area.
- DMAC0.DMOFR: Offset address: 10h
- DMAC0.DMCRA: Repeat size: 4h
- DMAC0.DMINT: The repeat size end interrupt is enabled.

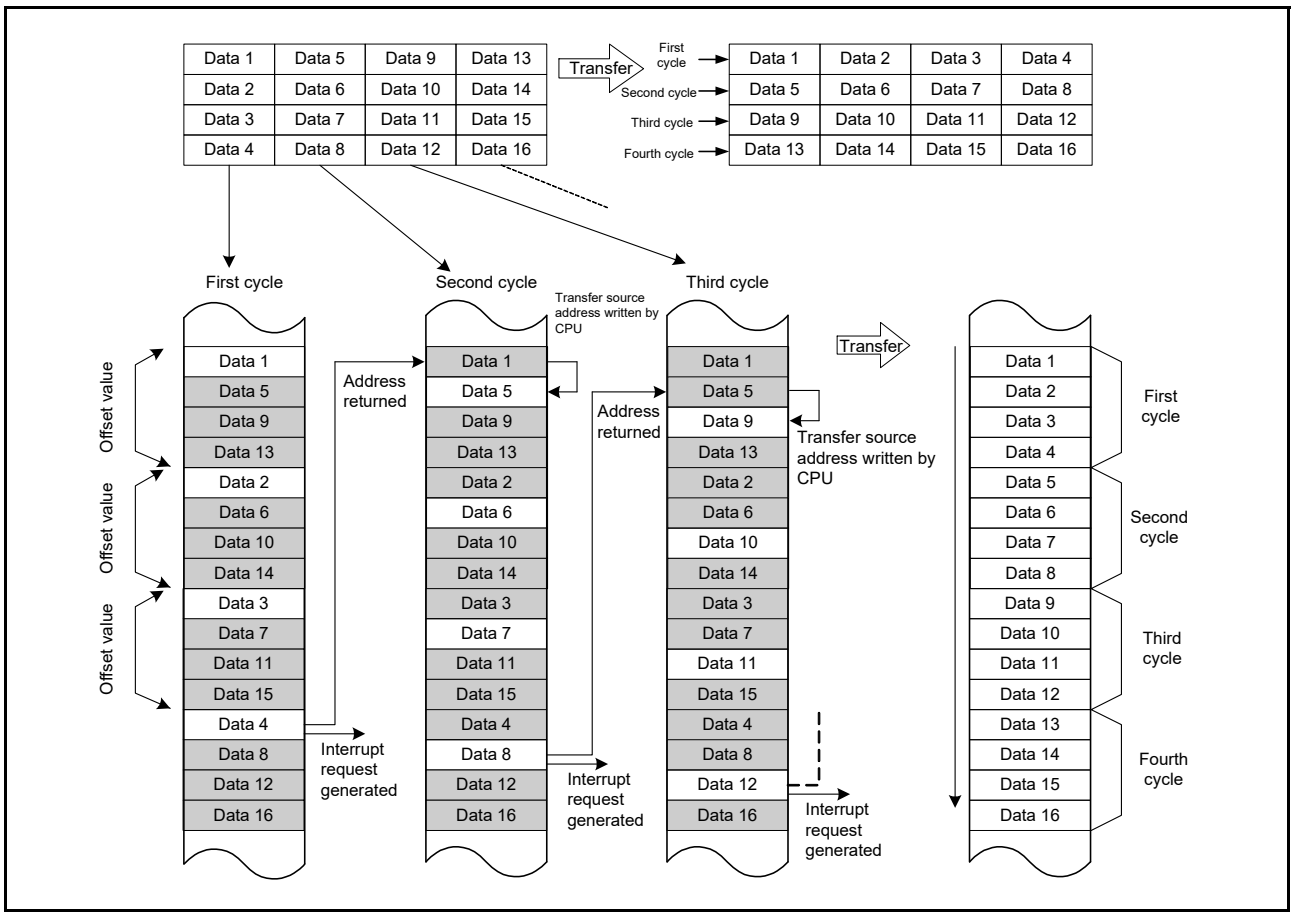


Figure 18.8 XY Conversion Operation Using Offset Addition in Repeat Transfer Mode

When a transfer starts, the offset value is added to the transfer source address every time data is transferred. The transfer data is written to the destination continuous addresses. When data 4 is transferred, which means that the repeat size of transfers is completed, the transfer source address returns to the transfer start address (address of data 1 on the transfer source) and a repeat size end interrupt is requested. While this interrupt stops the transfer temporarily, perform the following.

- DMAC0.DMSAR: Rewrite the DMA transfer source address to the address of data 5 (with the above example, the data 1 address + 4).
- DMAC0.DMCNT: Set the DTE bit to 1.

The DMA transfer is resumed from the state when the DMA transfer is stopped. After that, the operations described above are repeated until the transfer source data is transposed to the destination area (XY conversion).

Downloaded from Arrow.com.

Figure 18.9 shows a flowchart of the XY conversion.

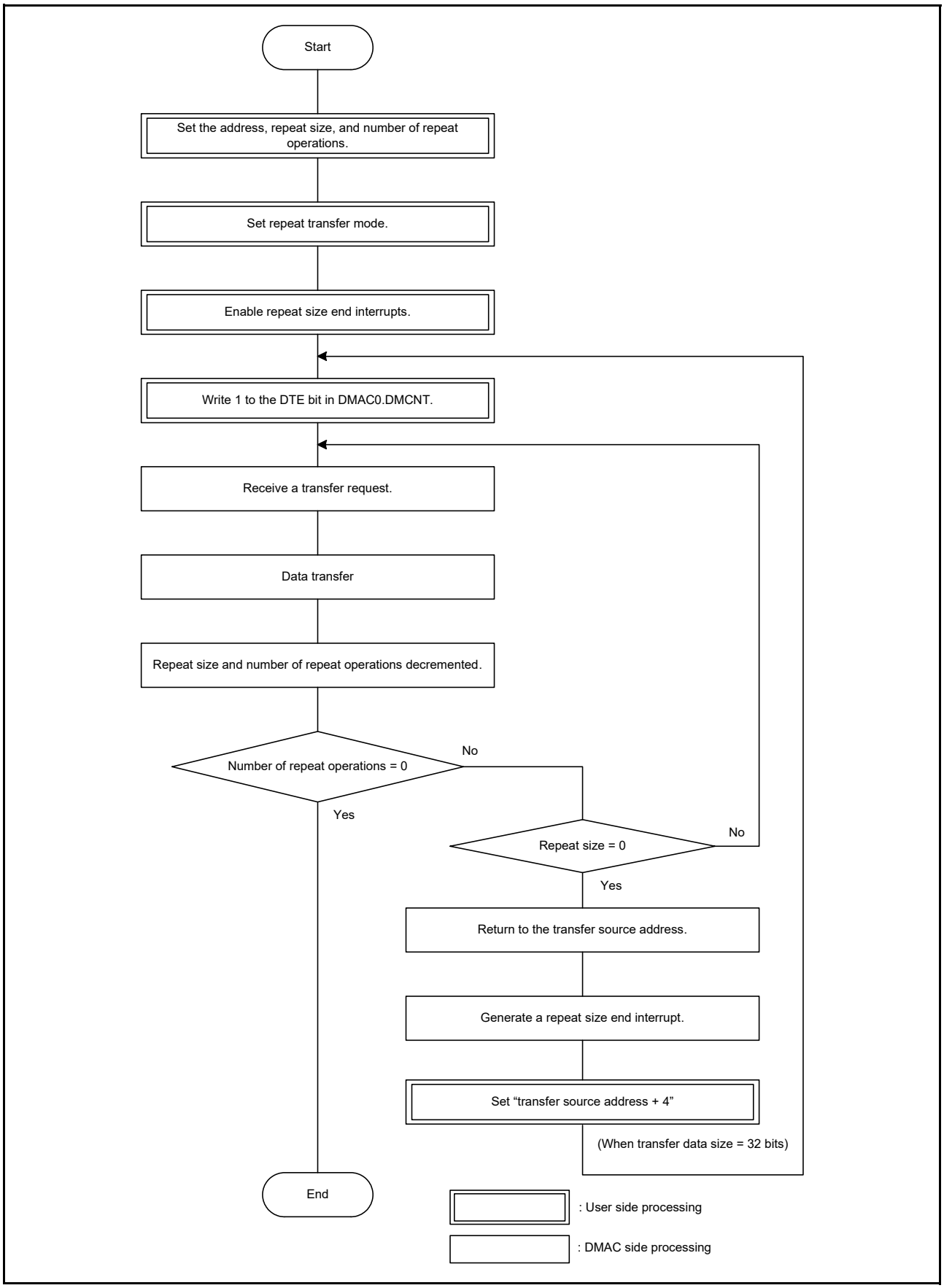


Figure 18.9 XY Conversion Flowchart Using Offset Addition in Repeat Transfer Mode

Downloaded from Arrow.com.

18.3.4 Activation Sources

Software, the interrupt requests from the peripheral modules, and the external interrupt requests can be specified as the DMAC activation sources. Setting the DCTG[1:0] bits in DMTMD of DMACm selects the activation source.

(1) DMAC Activation by Software

Setting the DCTG[1:0] bits in DMTMD of DMACm to 00b enables the DMAC activation by software.
To start DMA transfer by software, set the DCTG[1:0] bits in DMTMD of DMACm to 00b, and then set the DTE bit in DMCNT of DMACm to 1 (DMA transfer is enabled) and the SWREQ bit in DMREQ of DMACm to 1 (DMA transfer is requested) with the DMST bit in DMAST set to 1 (DMAC activation enabled).
When the DMAC is activated by software while the CLRS bit in DMREQ of DMACm is 0, the SWREQ bit in DMREQ of DMACm is cleared to 0 after data transfer is started in response to a DMA transfer request.
When the DMAC is activated by software while the CLRS bit is 1, the SWREQ bit is not cleared to 0 after data transfer is started. In this case, a DMA transfer request is issued again after completion of a transfer.

(2) DMAC Activation by Interrupt Requests from On-Chip Peripheral Modules or External Interrupt Requests

Interrupt requests from the on-chip peripheral modules and external interrupt requests can be specified as the DMAC activation sources. The activation source can be selected separately for each channel using the DMRSRm registers (m = 0 to 3) of the ICU.
The DMAC is activated when an interrupt request from the on-chip peripheral module or an external interrupt request is generated while the DCTG[1:0] bits in DMTMD of DMACm is set to 01b (interrupts from the peripheral modules and the external interrupt pins are selected), the DTE bit in DMCNT of DMACm is set to 1 (DMA transfer is enabled), and the DMST bit in DMAST is set to 1 (DMAC activation is enabled).

For interrupt requests specified as DMAC activation sources, see Table 15.3, Interrupt Vector Table, in section 15, Interrupt Controller (ICUb).

Downloaded from Arrow.com.

18.3.5 Operation Timing

Figure 18.10 and Figure 18.11 show DMAC operation timing examples.

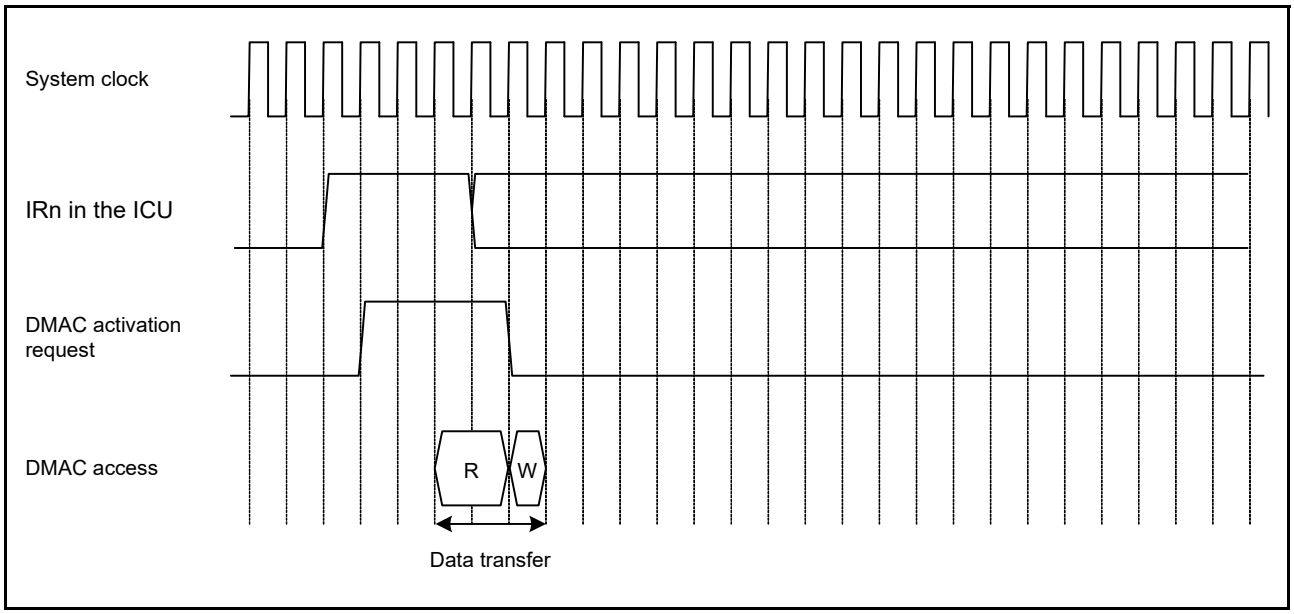


Figure 18.10 DMAC Operation Timing Example (1) (DMA Activation by Interrupt from Peripheral Module/ External Interrupt Input Pin, Normal Transfer Mode, Repeat Transfer Mode)

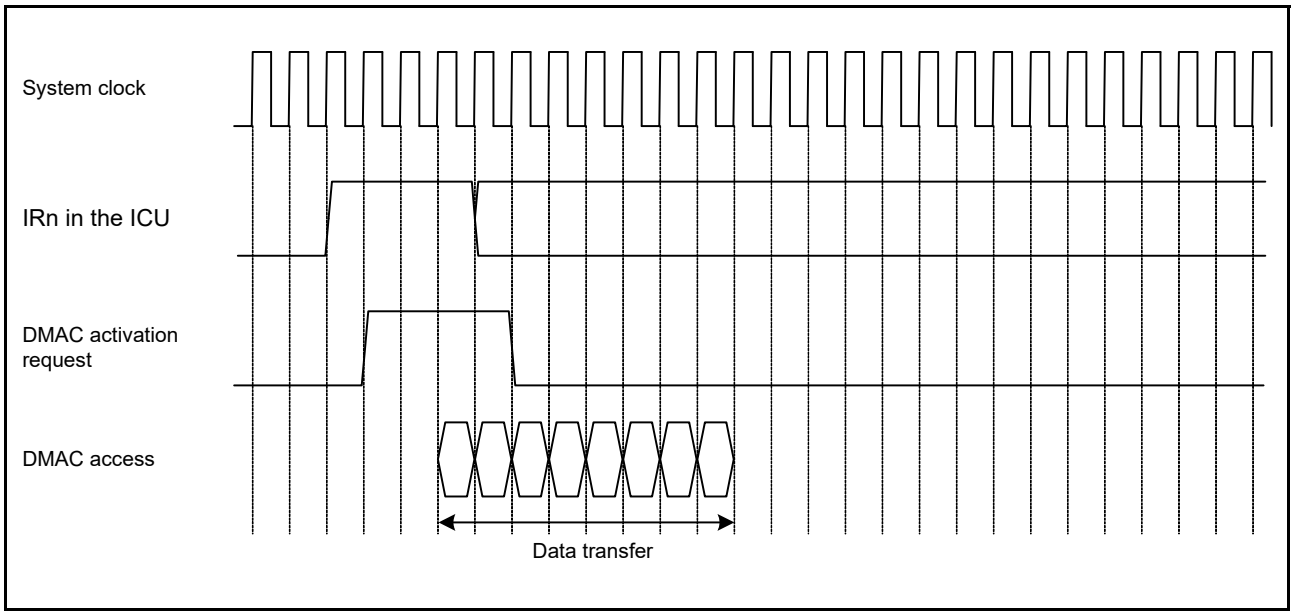


Figure 18.11 DMAC Operation Timing Example (2) (DMA Activation by Interrupt from Peripheral Module/ External Interrupt Input Pin, Block Transfer Mode, Block Size = 4)

Downloaded from Arrow.com.

18.3.6 DMAC Execution Cycles

Table 18.7 lists execution cycles in one DMAC data transfer operation.

Table 18.7 DMAC Execution Cycles

Transfer Mode	Data Transfer (Read)	Data Transfer (Write)
Normal	Cr+1	Cw
Repeat	Cr+1	Cw
Block*1	P × Cr	P × Cw

Note 1. This is the case when the block size is 2 or more. When the block size is 1, normal transfer cycle is applied.

P: Block size (DMCRAH register setting)
Cr: Data read destination access cycle
Cw: Data write destination access cycle

Cr and Cw depend on the access destination. For the number of cycles for each access destination, see section 49, RAM, section 50, Flash Memory (FLASH), and section 5, I/O Registers.
The unit for +1 in “Data Transfer (Read)” column is one system clock cycle (ICLK).
For the operation example, see section 18.3.5, Operation Timing.

Downloaded from Arrow.com.

18.3.7 Activating the DMAC

Figure 18.12 shows the register setting procedure.

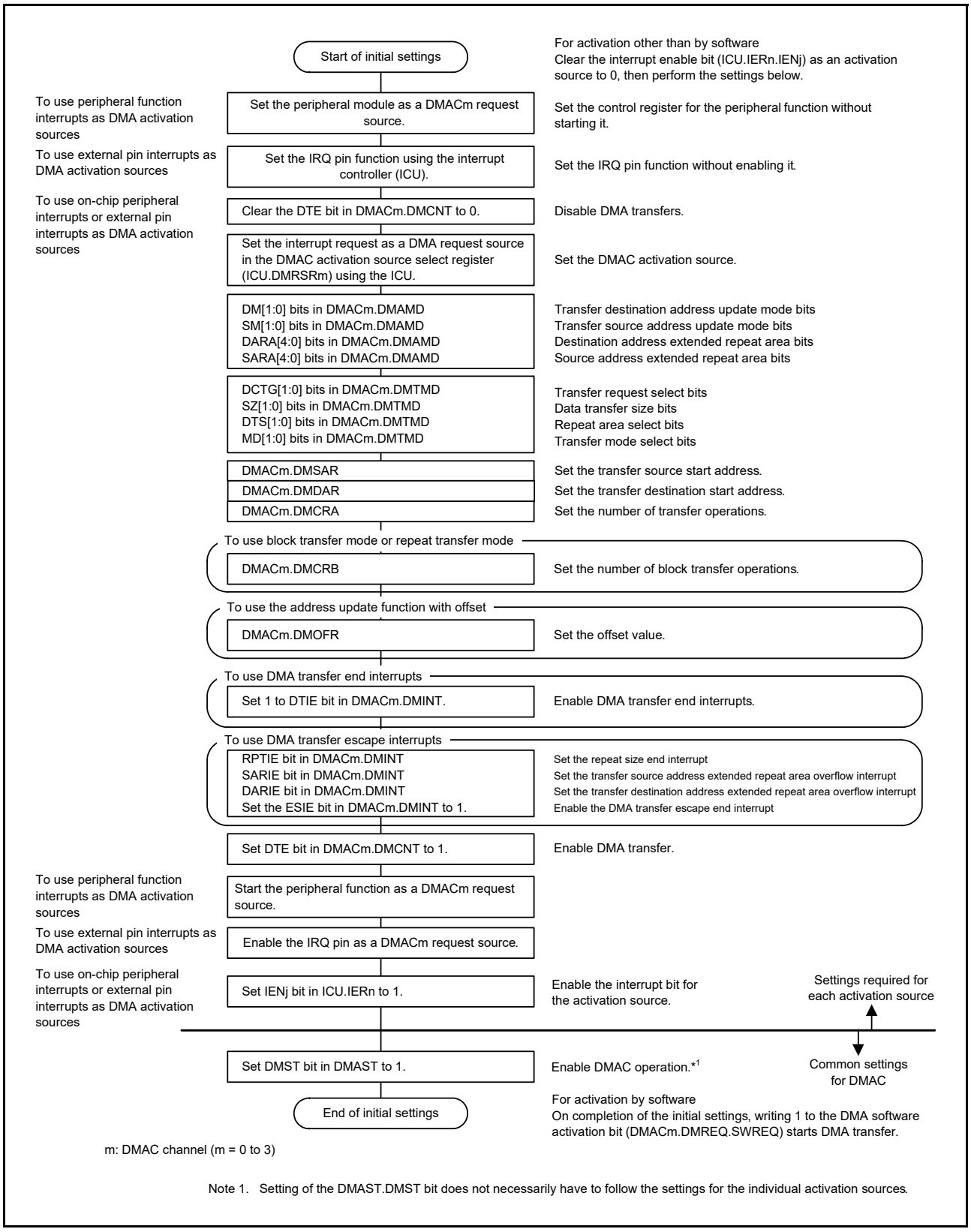


Figure 18.12 Register Setting Procedure

Downloaded from Arrow.com.

18.3.8 Starting DMA Transfer

Setting the DTE bit in DMCNT of DMACm to 1 (DMA transfer enabled) and setting the DMST bit in DMAST to 1 (DMAC start enabled) enable DMA transfer of channel m (m = 0 to 3).
Another activation request cannot be accepted during the transfer of other DMAC channel or DTC. When the proceeding transfer is completed, channel arbitration is performed where a DMA transfer request of the highest priority channel is accepted and DMA transfer of the channel starts. When DMA transfer starts, the ACT bit in DMSTS of DMACm is set to 1 (the DMAC is in the active state).

18.3.9 Registers during DMA Transfer

The DMAC registers are updated by a DMA transfer. The value to be updated differs according to the other settings and the transfer state. The registers to be updated are DMSAR, DMDAR, DMCRA, DMCRB, DMCNT, and DMSTS of DMACm.

(1) DMA Source Address Register (DMACm.DMSAR)

When data has been transferred in response to one transfer request, the contents of DMSAR are updated to the address to be accessed by the next transfer request.
For details on register update operation in each transfer mode, refer to Table 18.3 to Table 18.5.

(2) DMA Destination Address Register (DMACm.DMDAR)

When data has been transferred in response to one transfer request, the contents of DMDAR are updated to the address to be accessed by the next transfer request.
For details on register update operation in each transfer mode, refer to Table 18.3 to Table 18.5.

(3) DMA Transfer Count Register (DMACm.DMCRA)

When data has been transferred in response to one transfer request, the count value is updated. The update operation depends on the transfer mode selected.
For details on register update operation in each transfer mode, refer to Table 18.3 to Table 18.5.

(4) DMA Block Transfer Count Register (DMACm.DMCRB)

When data has been transferred in response to one transfer request, the count value is updated. The update operation depends on the transfer mode selected.
For details on register update operation in each transfer mode, refer to Table 18.3 to Table 18.5.

(5) DMA Transfer Enable Bit (DMACm.DMCNT.DTE)

Although the DMACm.DMCNT.DTE bit enables or disables data transfer by the register write access, it is automatically cleared to 0 by the DMAC according to the DMA transfer state.
The conditions for clearing this bit by the DMAC are as follows:

- When the specified total volume of data transfer is completed
- When DMA transfer is stopped by the repeat size end interrupt
- When DMA transfer is stopped by the extended repeat area overflow interrupt

Writing to the registers for the channels when the corresponding DMACm.DMCNT.DTE bit is set to 1 is prohibited (except for DMACm.DMCNT). In this case, writing must be performed after the bit is cleared to 0.

Downloaded from Arrow.com.

(6) DMA Active Flag (DMACm.DMSTS.ACT)

The ACT bit in DMSTS of DMACm indicates whether the DMACm is in the idle or active state.
This flag is set to 1 when the DMAC starts data transfer, and is cleared to 0 when data transfer in response to one transfer request is completed.
Even when DMA transfer is stopped by writing 0 to the DTE bit in DMCNT of DMACm during DMA transfer, this flag remains 1 until DMA transfer is completed.

(7) Transfer End Interrupt Flag (DMACm.DMSTS.DTIF)

The DTIF flag in DMSTS of DMACm is set to 1 after DMA transfer of the total transfer size of data is completed.
When both this flag and the DTIE bit in DMINT of DMACm are set to 1, a transfer end interrupt is requested.
This flag is set to 1 when the DMA transfer bus cycle is completed and the ACT flag in DMSTS of DMACm is cleared to 0 indicating the DMA transfer end.
This flag is automatically cleared to 0 when the DTE bit in DMCNT of DMACm is set to 1 during the interrupt handling.

(8) Transfer Escape End Interrupt Flag (DMACm.DMSTS.ESIF)

The ESIF flag in DMSTS of DMACm is set to 1 when a repeat size end interrupt or extended repeat area overflow interrupt is requested. When this bit and the ESIE bit in DMINT of DMACm are set to 1, a transfer escape end interrupt is requested.
This flag is set to 1 when the bus cycle of the DMA transfer having caused the interrupt request is completed and the ACT flag in DMSTS of DMACm is cleared to 0 indicating the DMA transfer end.
This flag is automatically cleared to 0 when the DTE bit in DMCNT of DMACm is set to 1 during an interrupt handling.
Before sending an interrupt request from the DMAC to the CPU or the DTC, the interrupt control register must be set. For details, see section 15, Interrupt Controller (ICUb).

18.3.10 Channel Priority

When multiple DMA transfer requests are present, the DMAC determines the priority of channels that have DMA transfer requests.
The channel priority is fixed as channel 0 > channel 1 > channel 2 > channel 3 (channel 0: highest).
When a DMA transfer request is generated during data transfer, channel arbitration is started after the final data has been transferred, and DMA transfer of the higher-priority channel starts.

Downloaded from Arrow.com.

18.4 Ending DMA Transfer

The operation for ending DMA transfer depends on the transfer end conditions. When DMA transfer ends, the DTE bit in DMCNT and the ACT flag in DMSTS of DMACm are changed from 1 to 0, indicating that DMA transfer has ended.

18.4.1 Transfer End by Completion of Specified Total Number of Transfer Operations

(1) In Normal Transfer Mode (DMACm.DMTMD.MD[1:0] = 00b)

When the value of DMCRAL of DMACm changes from 1 to 0, DMA transfer ends on the corresponding channel, and the DTE bit in DMCNT of DMACm is cleared to 0 and the DTIF bit in DMSTS of DMACm is set to 1 at the same time. If the DTIE bit in DMINT of DMACm is 1 at this time, a transfer end interrupt request is issued to the CPU or the DTC.

(2) In Repeat Transfer Mode (DMACm.DMTMD.MD[1:0] = 01b)

When the value of DMCRB of DMACm changes from 1 to 0, DMA transfer ends on the corresponding channel, and the DTE bit in DMCNT of DMACm is cleared to 0 and the DTIF bit in DMSTS of DMACm is set to 1 at the same time. If the DTIE bit in DMINT of DMACm is 1 at this time, an interrupt request is issued to the CPU or the DTC.

(3) In Block Transfer Mode (DMACm.DMTMD.MD[1:0] = 10b)

When the value of DMCRB of DMACm changes from 1 to 0, DMA transfer ends on the corresponding channel, and the DTE bit in DMCNT of DMACm is cleared to 0 and the DTIF bit in DMSTS of DMACm is set to 1 at the same time. If the DTIE bit in DMINT of DMACm is 1 at this time, an interrupt request is issued to the CPU or the DTC.
Before sending an interrupt request from the DMAC to the CPU or the DTC, the interrupt control register must be set. For details, see section 15, Interrupt Controller (ICUb).

18.4.2 Transfer End by Repeat Size End Interrupt

In repeat transfer mode, a repeat size end interrupt is requested when transfer of a 1-repeat size of data is completed while the RPTIE bit in DMINT of DMACm is set to 1. When the interrupt is requested to complete DMA transfer, the DTE bit in DMCNT of DMACm is cleared to 0 and the ESIF flag in DMSTS of DMACm is set to 1. If the ESIE bit in DMINT of DMACm is 1 at this time, an interrupt request is issued to the CPU or the DTC. Here, the transfer can be resumed by writing 1 to the DTE bit in DMCNT of DMACm.
A repeat size end interrupt can be requested also in block transfer mode. In block transfer mode, the interrupt is requested in the same way as in repeat transfer mode when transfer of a 1-block size data is completed.
Before sending an interrupt request from the DMAC to the CPU or the DTC, the interrupt control register must be set. For details, see section 15, Interrupt Controller (ICUb).

Downloaded from Arrow.com.

18.4.3 Transfer End by Interrupt on Extended Repeat Area Overflow

When an overflow on the extended repeat area occurs while the extended repeat area is specified and the SARIE or DARIE bit in DMINT of DMACm is set to 1, an interrupt by an extended repeat area overflow is requested. When the interrupt is requested, the DMA transfer is terminated, the DTE bit in DMCNT of DMACm is cleared to 0, and the ESIF flag in DMSTS of DMACm is set to 1. If the ESIE bit in DMINT of DMACm is 1 at this time, an interrupt request is issued to the CPU or the DTC.

Even if an interrupt by an extended repeat area overflow is requested during a read cycle, the following write cycle is performed.

In block transfer mode, even if an interrupt by an extended repeat area overflow is requested during a 1-block transfer, the remaining data in the block is transferred; transfer is terminated after a block transfer.

Before sending an interrupt request from the DMAC to the CPU or the DTC, the interrupt control register must be set. For details, see section 15, Interrupt Controller (ICUb).

Downloaded from Arrow.com.

18.5 Interrupts

Each DMAC channel can output an interrupt request to the CPU or the DTC after transfer in response to one request is completed. When the transfer destination is the on-chip peripheral bus, an interrupt request is generated upon completion of data write to the write buffer not to the actual transfer destination.

Table 18.8 lists the relation among the interrupt sources, the interrupt status flags, and the interrupt enable bits. Figure 18.13 shows the schematic logic diagram of interrupt outputs. Figure 18.14 shows the DMAC interrupt handling routine to resume or terminate DMA transfer.

Table 18.8 Relation among Interrupt Sources, Interrupt Status Flags, and Interrupt Enable Bits

Interrupt Sources		Interrupt Enable Bits	Interrupt Status Flags	Request Output Enable Bits
Transfer end		—	DMACm.DMSTS.DTIF	DMACm.DMINT.DTIE
Escape transfer end	Repeat size end	DMACm.DMINT.RPTIE	DMACm.DMSTS.ESIF	DMACm.DMINT.ESIE
	Source address extended repeat area overflow	DMACm.DMINT.SARIE		
	Destination address extended repeat area overflow	DMACm.DMINT.DARIE		

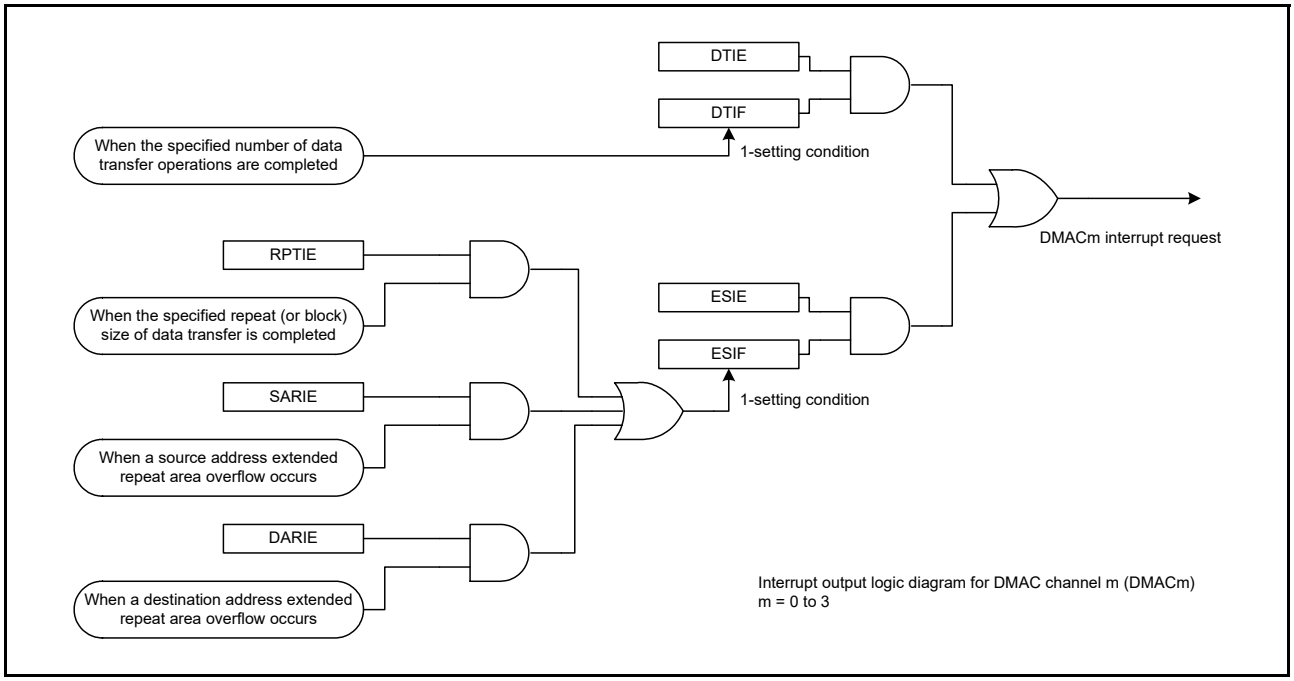


Figure 18.13 Schematic Logic Diagram of Interrupt Outputs

Downloaded from Arrow.com.

Specifically, the different procedures are used for canceling an interrupt to restart DMA transfer in the following two cases: (1) discontinuing or terminating DMA transfer and (2) continuing DMA transfer.

(1) When Discontinuing or Terminating DMA Transfer

Write 0 to the DTIF bit in DMSTS of DMACm to clear a transfer end interrupt, and to the ESIF bit in DMSTS of DMACm to clear a repeat size interrupt and an extended repeat area overflow interrupt. The DMACm remains in the stop state. When starting another DMA transfer after that, set the appropriate registers, and set the DTE bit in DMCNT of DMACm to 1 (DMA transfer enabled).

(2) When Continuing DMA Transfer

Write 1 to the DTE bit in DMCNT of DMACm. The ESIF bit in DMSTS of DMACm is automatically cleared to 0 (interrupt source cleared), and DMA transfer is resumed.

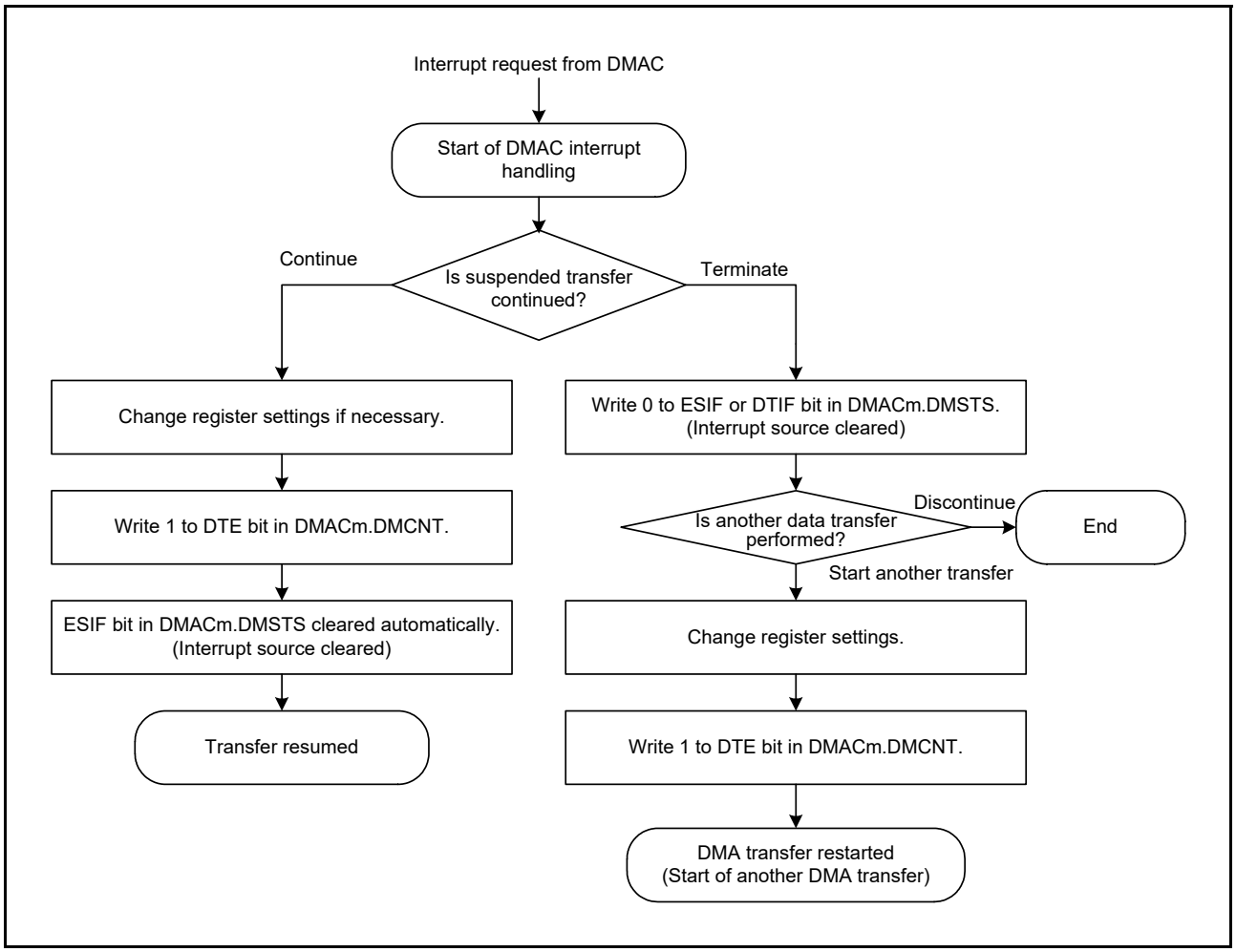


Figure 18.14 DMAC Interrupt Handling Routine to Resume/Terminate DMA Transfer

18.6 Event Link Function

Each DMAC channel outputs an event link request signal each time the channel completes data transfer (or block transfer in block transfer mode). However, when the transfer destination is the internal peripheral bus, an event link request signal is generated when the write to the write buffer is accepted.

Downloaded from Arrow.com.

18.7 Low Power Consumption Function

Before transition to the module stop state or software standby mode, clear the DMAST.DMST bit to 0 (DMAC activation is disabled), and then perform the following.

(1) Module Stop Function

Writing 1 to the MSTPA28 bit (transition to the module-stop state) in MSTPCRA enables the module-stop function of the DMAC. If DMA transfer is in progress at the time 1 is written to the MSTPA28 bit, the transition to the module-stop state proceeds after DMA transfer has ended. While the MSTPA28 bit is 1, accessing the DMAC registers are prohibited. Writing 0 to the MSTPA28 bit releases the DMAC from the module-stop state.

(2) Software Standby

Make settings in accord with the procedure under section 11.6.3.1, Entry to Software Standby Mode in section 11, Low Power Consumption.
If DMA transfer operations are in progress at the time the WAIT instruction is executed, the transition to software standby follows the completion of DMA transfer.

(3) Note on Low Power Consumption Function

For the WAIT instruction and the register setting procedure, see section 11.7.5, Timing of WAIT Instructions in section 11, Low Power Consumption.
To perform DMA transfer after returning from low power consumption mode, set the DMST bit in DMAST to 1 again.
To use a request that is generated in software standby mode as an interrupt request to the CPU but not as a DMAC startup request, specify the CPU as the interrupt request destination in accordance with the description in section 15.4.3, Selecting Interrupt Request Destinations in section 15, Interrupt Controller (ICUb), and then execute the WAIT instruction.

Downloaded from Arrow.com.

18.8 Usage Notes

18.8.1 DMA Transfer to Peripheral Modules

In DMA transfer to a peripheral module, the ACT bit in DMSTS of DMACm may be cleared to 0 (DMAC transfer suspended) during the period from the beginning of the final data write to the end of the peripheral bus access.

18.8.2 Access to the Registers during DMA Transfer

The DMSAR, DMDAR, DMCRA, DMCRB, DMTMD, DMINT, DMAMD, DMOFR, and DMCSL registers of DMACm must not be accessed while the ACT bit in DMSTS of the same channel is set to 1 (DMAC active state) or the DTE bit in DMCNT of the same channel is set to 1 (DMA transfer enabled).

18.8.3 DMA Transfer to Reserved Areas

DMA transfer to the reserved areas is prohibited. If such an access is made, transfer results are not guaranteed. For details on the reserved areas, see section 4, Address Space.

18.8.4 Interrupt Request by the DMA Activation Source Flag Control Register (DMCSL) at the End of each Transfer

While the DMACm.DMCSL.DISEL bit is 1, an interrupt is issued to the CPU at the end of each transfer that has been activated by one DMA request. Unlike the transfer end interrupt that the DMAC outputs or the escape end interrupt, the interrupt of this type is issued to the CPU at the end of DMA transfer without clearing the interrupt flag of the DMAC activation source to 0 by changing the interrupt request destination to the CPU. In this case, since the interrupt flag is not cleared to 0 at the end of DMAC transfer, it should be cleared to 0 by the CPU interrupt routine.
The interrupt flag is cleared when the CPU interrupt is accepted.
For the change of the settings on the interrupt flag or the interrupt request destination, see section 15, Interrupt Controller (ICUb). For the DMACm.DMCSL.DISEL bit setting, see section 18.2.12, DMA Activation Source Flag Control Register (DMCSL).

18.8.5 Setting of DMAC Activation Source Select Register of the Interrupt Controller (ICU.DMRSRm)

The DMAC activation source select register (ICU.DMRSRm) should be set while the DMA transfer enable bit (DMACm.DMCNT.DTE) is cleared to 0 (DMA transfer is disabled). Moreover, the DTC activation enable register (ICU.DTCERm) that corresponds to the same vector number that has been set by the ICU.DMRSRm register should not be set to 1. For details on the ICU.DTCERn and ICU.DMRSRm, see section 15, Interrupt Controller (ICUb).

18.8.6 Suspending or Restarting DMA Activation

To suspend a DMA activation request, write 0 to the interrupt enable bit for the activation source (ICU.IERn.IENj bit). To restart the DMA transfer, write 1 to the ICU.IERn.IENj bit with the setting shown in section 18.3.7, Activating the DMAC.

Downloaded from Arrow.com.

19. Data Transfer Controller (DTCa)

This MCU incorporates a data transfer controller (DTC).
The DTC is triggered by an interrupt request to perform data transfers.

19.1 Overview

Table 19.1 lists the specifications of the DTC, and Figure 19.1 shows a block diagram of the DTC.

Table 19.1 DTC Specifications

Item	Description
Number of transfer channels	• The same number as all interrupt sources that can start the DTC transfer.
Transfer modes	• Normal transfer mode A single transfer request leads to a single data transfer. • Repeat transfer mode A single transfer request leads to a single data transfer. The transfer address is returned to the transfer start address after the number of data transfers corresponding to “repeat size”. The maximum number of repeat transfers is 256, and the maximum data transfer size is 256 × 32 bits, 1024 bytes. • Block transfer mode A single transfer request leads to the transfer of a single block. The maximum block size is 256 × 32 bits = 1024 bytes.
Chain transfer	• Multiple types of data transfers can sequentially be executed in response to a single request. • Either “performed only when the transfer counter becomes 0” or “every time” can be selected.
Transfer space	• In short-address mode: 16 Mbytes (Areas from 0000 0000h to 007F FFFFh and FF80 0000h to FFFF FFFFh except reserved areas) • In full-address mode: 4 Gbytes (Area from 0000 0000h to FFFF FFFFh except reserved areas)
Data transfer units	• Single data: 1 byte (8 bits), 1 word (16 bits), 1 longword (32 bits) • Single block size: 1 to 256 data
CPU interrupt source	• An interrupt request can be generated to the CPU on a request source for a data transfer. • An interrupt request can be generated to the CPU after a single data transfer. • An interrupt request can be generated to the CPU after data transfer of specified volume.
Event link function	An event link request is generated after one data transfer (for block, after one block transfer).
Read skip	Reading of the transfer information can be skipped when the same transfer is repeated.
Write-back skip	Write-back of the transferred data that is not updated can be skipped when the address of the transfer source or destination is fixed.
Low power consumption function	Module stop state can be set.

Downloaded from Arrow.com.

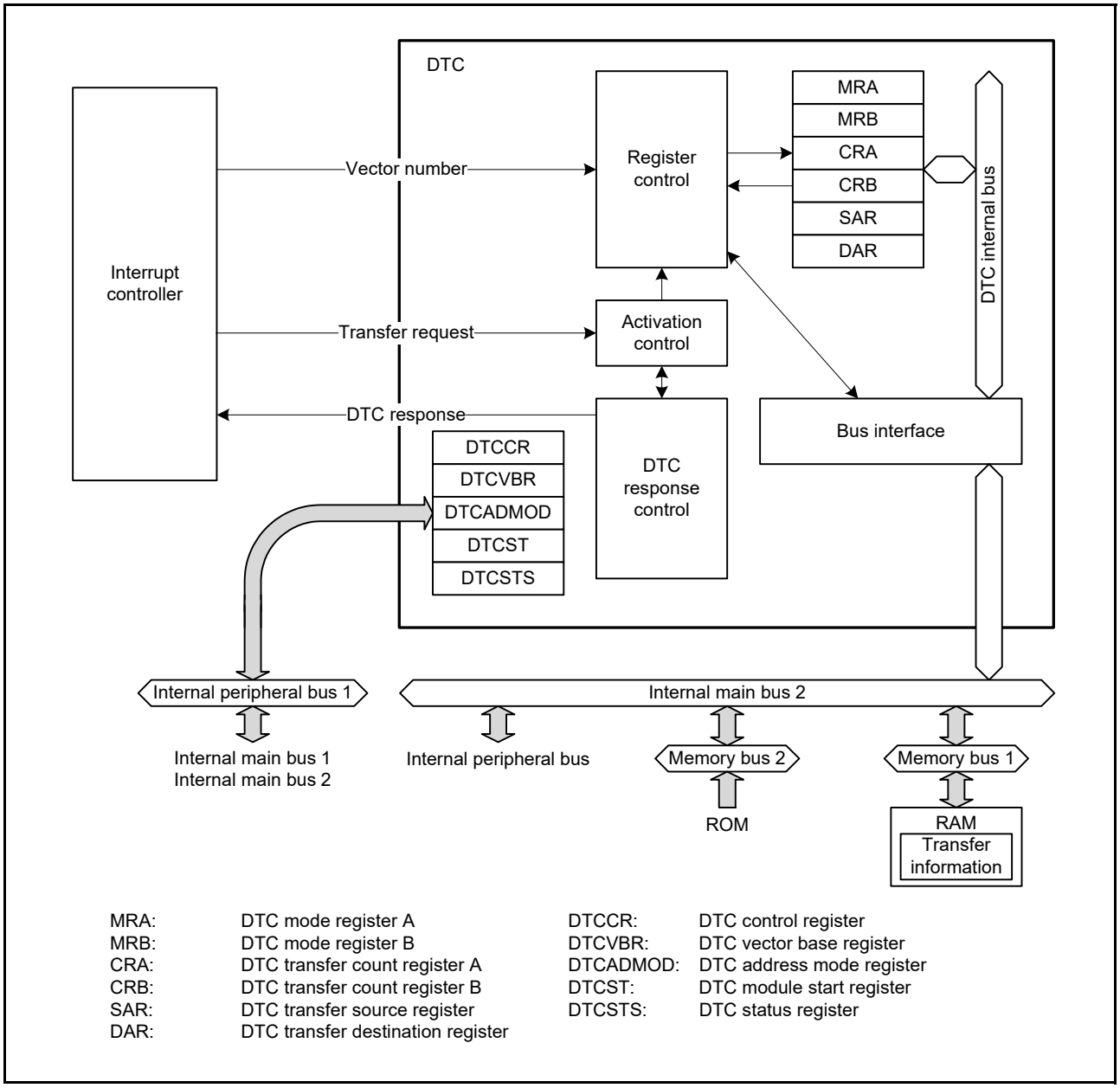


Figure 19.1 DTC Block Diagram

Downloaded from Arrow.com.

19.2 Register Descriptions

Registers MRA, MRB, SAR, DAR, CRA, and CRB are DTC internal registers, which cannot be directly accessed from the CPU. Values to be set in these DTC internal registers are placed in the RAM area as transfer information. When accepting a transfer request, the DTC reads the transfer information from the RAM area and sets it in the internal registers. After the data transfer ends, the values of the updated internal register are written back to the RAM area as transfer information.

19.2.1 DTC Mode Register A (MRA)

Address(es): (inaccessible directly from the CPU)

	b7	b6	b5	b4	b3	b2	b1	b0
	MD[1:0]		SZ[1:0]		SM[1:0]		—	—
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b1, b0	—	Reserved	Set these bits to 0.	—
b3, b2	SM[1:0]	Transfer Source Address Addressing Mode	b3 b2 0 0: The address in the SAR register is fixed. (write-back to SAR is skipped.) 0 1: The address in the SAR register is fixed. (write-back to SAR is skipped.) 1 0: The SAR value is incremented after a data transfer. (+1 when the SZ[1:0] bits are 00b, +2 when 01b, +4 when 10b) 1 1: The SAR value is decremented after a data transfer. (–1 when the SZ[1:0] bits are 00b, –2 when 01b, –4 when 10b)	—
b5, b4	SZ[1:0]	DTC Data Transfer Size	b5 b4 0 0: Byte (8-bit) transfer 0 1: Word (16-bit) transfer 1 0: Longword (32-bit) transfer 1 1: Setting prohibited	—
b7, b6	MD[1:0]	DTC Transfer Mode Select	b7 b6 0 0: Normal transfer mode 0 1: Repeat transfer mode 1 0: Block transfer mode 1 1: Setting prohibited	—

MRA register is used to select the DTC operating mode and cannot be accessed directly from the CPU.

Downloaded from Arrow.com.

19.2.2 DTC Mode Register B (MRB)

Address(es): (inaccessible directly from the CPU)

	b7	b6	b5	b4	b3	b2	b1	b0
	CHNE	CHNS	DISEL	DTS	DM[1:0]		—	—
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b1, b0	—	Reserved	Set these bits to 0.	—
b3, b2	DM[1:0]	Transfer Destination Address Addressing Mode	b3 b2 0 0: The address in the DAR register is fixed. (Write-back to DAR is skipped.) 0 1: The address in the DAR register is fixed. (Write-back to DAR is skipped.) 1 0: The DAR value is incremented after data transfer. (+1 when the MRA.SZ[1:0] bits are 00b, +2 when 01b, +4 when 10b) 1 1: The DAR value is decremented after data transfer. (–1 when the MRA.SZ[1:0] bits are 00b, –2 when 01b, –4 when 10b)	—
b4	DTS	DTC Transfer Mode Select	0: Transfer destination side is repeat area or block area. 1: Transfer source side is repeat area or block area.	—
b5	DISEL	DTC Interrupt Select	0: An interrupt request to the CPU is generated on completion of the specified number of data transfers. 1: An interrupt request to the CPU is generated for each data transfer.	—
b6	CHNS	DTC Chain Transfer Select	0: Chain transfer is performed on completion of each transfer. 1: Chain transfer is performed only when the transfer counter is changed from 1 to 0 or 1 to CRAH.	—
b7	CHNE	DTC Chain Transfer Enable	0: Chain transfer is disabled. 1: Chain transfer is enabled.	—

MRB register is used to select the DTC operating mode and cannot be accessed directly from the CPU.

DTS Bit (DTC Transfer Mode Select)

The DTS bit specifies the side (transfer source or destination) to be a repeat area or block area in repeat transfer mode or block transfer mode.

CHNS Bit (DTC Chain Transfer Select)

The CHNS bit selects the chain transfer condition.

When the CHNE bit is 0, setting of the CHNS bit is ignored. For details on the conditions to select the chain transfer, refer to Table 19.3, Chain Transfer Conditions.

When the next transfer is chain transfer, completion of the specified number of transfers is not determined, the interrupt status flag for the request source is not cleared, and an interrupt request to the CPU is not generated.

CHNE Bit (DTC Chain Transfer Enable)

The CHNE bit enables or disables chain transfer.

The chain transfer condition is selected by the CHNS bit.

For details of chain transfer, refer to section 19.4.6, Chain Transfer.

Downloaded from Arrow.com.

19.2.3 DTC Transfer Source Register (SAR)

Address(es): (inaccessible directly from the CPU)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

x: Undefined

SAR register is used to set the transfer source start address.
In full-address mode, 32 bits are valid.
In short-address mode, lower 24 bits are valid and upper 8 bits (b31 to b24) are ignored. The address of this register is extended by the value specified by b23.
SAR register cannot be accessed directly from the CPU.

19.2.4 DTC Transfer Destination Register (DAR)

Address(es): (inaccessible directly from the CPU)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

x: Undefined

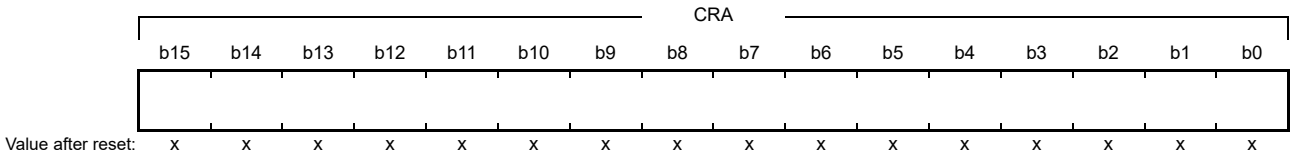
DAR register is used to set the transfer destination start address.
In full-address mode, 32 bits are valid.
In short-address mode, lower 24 bits are valid and upper 8 bits (b31 to b24) are ignored. The address of this register is extended by the value specified by b23.
DAR register cannot be accessed directly from the CPU.

Downloaded from Arrow.com.

19.2.5 DTC Transfer Count Register A (CRA)

- Normal transfer mode

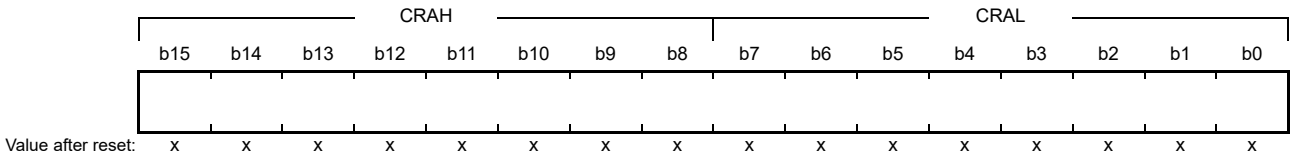
Address(es): (inaccessible directly from the CPU)



x: Undefined

- Repeat transfer mode/block transfer mode

Address(es): (inaccessible directly from the CPU)



x: Undefined

Symbol	Register Name	Description	R/W
CRAL	Transfer Counter A Lower Register	Set transfer count. This register functions as a transfer counter during data transfer.	—
CRAH	Transfer Counter A Upper Register	Set transfer count. This register functions as a reload register during data transfer.	—

Note: The function depends on transfer mode.
Note: Set CRAH and CRAL to the same value in repeat transfer mode and block transfer mode.

This register is for counting the number of transfers and cannot be accessed directly from the CPU.

(1) Normal transfer mode (MRA.MD[1:0] bits = 00b)

CRA register functions as a 16-bit transfer counter in normal transfer mode.
The transfer count is 1, 65535, and 65536 when the set value is 0001h, FFFFh, and 0000h, respectively.
The CRA value is decremented (–1) at each data transfer.

(2) Repeat transfer mode (MRA.MD[1:0] bits = 01b)

The CRAH register retains the transfer count and the CRAL register functions as an 8-bit transfer counter.
The transfer count is 1, 255, and 256 when the set value is 01h, FFh, and 00h, respectively.
The CRAL value is decremented (–1) at each data transfer. When it reaches 00h, the CRAH value is reloaded to the CRAL register.

(3) Block transfer mode (MRA.MD[1:0] bits = 10b)

The CRAH register retains the block size and the CRAL register functions as an 8-bit block size counter.
The transfer count is 1, 255, and 256 when the set value is 01h, FFh, and 00h, respectively.
The CRAL value is decremented (–1) at each data transfer. When it reaches 00h, the CRAH value is reloaded to the CRAL register.

Downloaded from Arrow.com.

19.2.6 DTC Transfer Count Register B (CRB)

Address(es): (inaccessible directly from the CPU)

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

x: Undefined

CRB register is used to set the block transfer count for block transfer mode and cannot be accessed directly from the CPU.
The transfer count is 1, 65535, and 65536 when the set value is 0001h, FFFFh, and 0000h, respectively.
The CRB value is decremented (–1) when the final data of a single block size is transferred.
When normal transfer mode or repeat transfer mode is selected, this register is not used and the set value is ignored.

19.2.7 DTC Control Register (DTCCR)

Address(es): DTC.DTCCR 0008 2400h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	RRS	—	—	—	—
Value after reset:	0	0	0	0	1	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b3	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b4	RRS	DTC Transfer Information Read Skip Enable	0: Transfer information read is not skipped. 1: Transfer information read is skipped when vector numbers match.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

DTCCR register is used to control the DTC operation.

RRS Bit (DTC Transfer Information Read Skip Enable)

The DTC vector number is compared with the vector number in the previous data transfer.
When these vector numbers match and the RRS bit is set to 1, DTC data transfer is performed without reading the transferred information. However, when the previous transfer was chain transfer, the transferred information is read regardless of the value of the RRS bit.
Furthermore, when the transfer counter (CRA register) became 0 during the previous normal transfer and when the transfer counter (CRB register) became 0 during the previous block transfer, the transferred information is read regardless of the RRS bit value.

Downloaded from Arrow.com.

19.2.8 DTC Vector Base Register (DTCVBR)

Address(es): DTC.DTCVBR 0008 2404h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The DTCVBR register is used to set the base address for calculating the address to which the DTC vector is allocated. Values for the upper 4 bits (b31 to b28) cannot be written but reflect the value written to b27. The lower 10 bits are reserved and the values are fixed to 0. Write 0 to the lower 10 bits if necessary.
It can be set in the range of 0000 0000h to 07FF FC00h and F800 0000h to FFFF FC00h in 1-Kbyte units.

19.2.9 DTC Address Mode Register (DTCADMOD)

Address(es): DTC.DTCADMOD 0008 2408h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	SHORT
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SHORT	Short-Address Mode Set	0: Full-address mode 1: Short-address mode	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

DTCADMOD register is used to specify the area accessible by the DTC.

SHORT Bit (Short-Address Mode Set)

This bit is used to select address mode of registers SAR and DAR.
Full-address mode allows the DTC to access to a 4-Gbyte space (0000 0000h to FFFF FFFFh).
Short-address mode allows the DTC to access to a 16-Mbyte space (0000 0000h to 007F FFFFh and FF80 0000h to FFFF FFFFh).

Downloaded from Arrow.com.

19.2.10 DTC Module Start Register (DTCST)

Address(es): DTC.DTCST 0008 240Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	DTCST
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DTCST	DTC Module Start	0: DTC module stop 1: DTC module start	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

DTCST Bit (DTC Module Start)

Set the DTCST bit to 1 to enable the DTC to accept transfer requests. When this bit is set to 0, transfer requests are no longer accepted.

If this bit is set to 0 during data transfer, the accepted transfer request is active until the processing is completed.

Set the DTCST bit to 0 before making a transition to the module stop state, deep sleep mode, or software standby mode.

Set the DTCST bit to 1 to resume the data transfer after returning from the module stop state, deep sleep mode, or software standby mode.

For details on transitions to the module stop state, deep sleep mode, and software standby mode, refer to section 19.9, Low Power Consumption Function, and section 11, Low Power Consumption.

Downloaded from Arrow.com.

19.2.11 DTC Status Register (DTCSTS)

Address(es): DTC.DTCSTS 0008 240Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ACT	—	—	—	—	—	—	—	VECN[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	VECN[7:0]	DTC Active Vector Number Monitoring Flag	These bits indicate the vector number for the request source when data transfer is in progress. The value is only valid if data transfer is in progress (the value of the ACT flag is 1).	R
b14 to b8	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b15	ACT	DTC Active Flag	0: Data transfer is not in progress. 1: Data transfer is in progress.	R

VECN[7:0] Flags (DTC Active Vector Number Monitoring Flag)

While data transfer is in progress, these bits indicate the vector number corresponding to the request source for the transfer.

When the DTCSTS register is read, the value of the VECN[7:0] flags is valid if the value of the ACT flag was 1 (data transfer is in progress) and invalid if the value of the ACT flag was 0 (data transfer is not in progress).

For the correspondence between the DTC request sources and the vector addresses, refer to section 15.3.1, Interrupt Vector Table in section 15, Interrupt Controller (ICUb).

ACT Flag (DTC Active Flag)

This flag indicates the state of data transfer operation.

[Setting condition]

- When the data transfer is started by a transfer request.

[Clearing condition]

- When the data transfer is completed in response to a transfer request.

Downloaded from Arrow.com.

19.3 Request Sources

The DTC data transfer is triggered by an interrupt request. Setting the ICU.DTCERn.DTCE bit (n = interrupt vector number) to 1 selects the corresponding interrupt request as a request source for the DTC.
For the correspondence between the DTC request sources and the vector addresses, refer to section 15.3.1, Interrupt Vector Table in section 15, Interrupt Controller (ICUb). For request by software, refer to section 15.2.5, Software Interrupt Generation Register (SWINTR) in section 15, Interrupt Controller (ICUb).
Once the DTC has accepted a transfer request, it does not accept another transfer request until transfer for that single request is completed, regardless of the priority of the requests.
When multiple transfer requests are generated during data transfer by the DMAC/DTC, the request with the highest priority on completion of the current transfer is accepted. When multiple transfer requests are generated while the DTCST.DTCST bit is 0 (DTC module stop), the request with the highest priority at the moment when the bit is subsequently set to 1 (DTC module start) is accepted.

The DTC performs the following operations at the start of a single data transfer (or the last of the consecutive transfers in the case of a chain transfer).

- On completion of a specified number of data transfer, the ICU.DTCERn.DTCE bit is set to 0 and an interrupt is requested to the CPU.
- If the MRB.DISEL bit is 1, an interrupt is requested to the CPU on completion of data transfer.
- For the other transfers, the interrupt status flag of the request source is set to 0 at the start of data transfer.

19.3.1 Allocating Transfer Information and DTC Vector Table

The DTC reads the start address of the transfer information corresponding to each request source from the vector table and reads the transfer information starting at that address.
The vector table should be located so that the lower 10 bits of the base address (start address) are 0. Use the DTC vector base register (DTCVBR) to set the base address of the DTC vector table.
Transfer information is allocated in the RAM area. The start address of the transfer information n with vector number n should be allocated at DTCVBR + 4n.
Transfer information should be aligned on a 4-byte boundary. The size of a transfer information is 12 bytes in short-address mode or 16 bytes in full-address mode. Use the DTCADMOD.SHORT bit to select short-address mode (SHORT bit = 1) or full-address mode (SHORT bit = 0).
Figure 19.2 shows the relationship between the DTC vector table and transfer information.
Figure 19.3 shows the allocation of transfer information in the RAM area. The lower addresses vary according to the endian of the corresponding allocation area. For details, refer to section 19.10.2, Allocating Transfer Information.

Downloaded from Arrow.com.

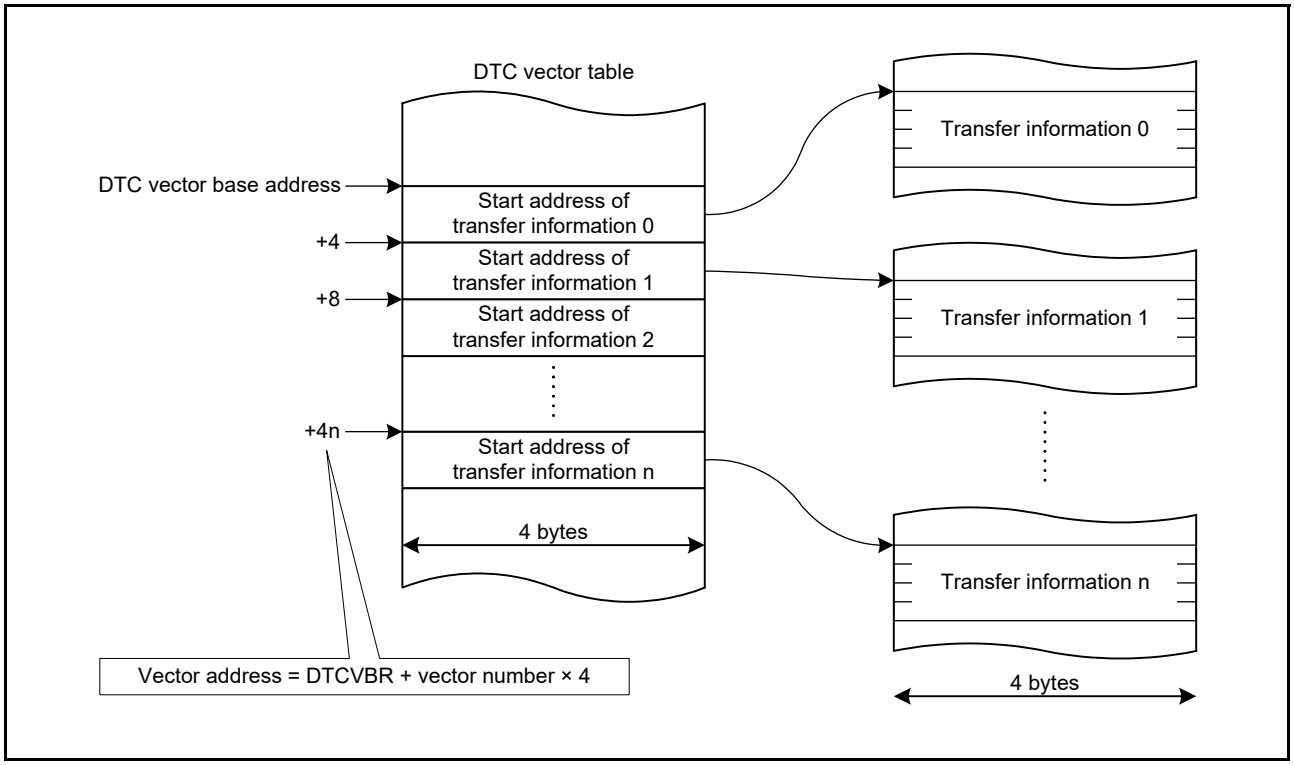


Figure 19.2 DTC Vector Table and Transfer Information

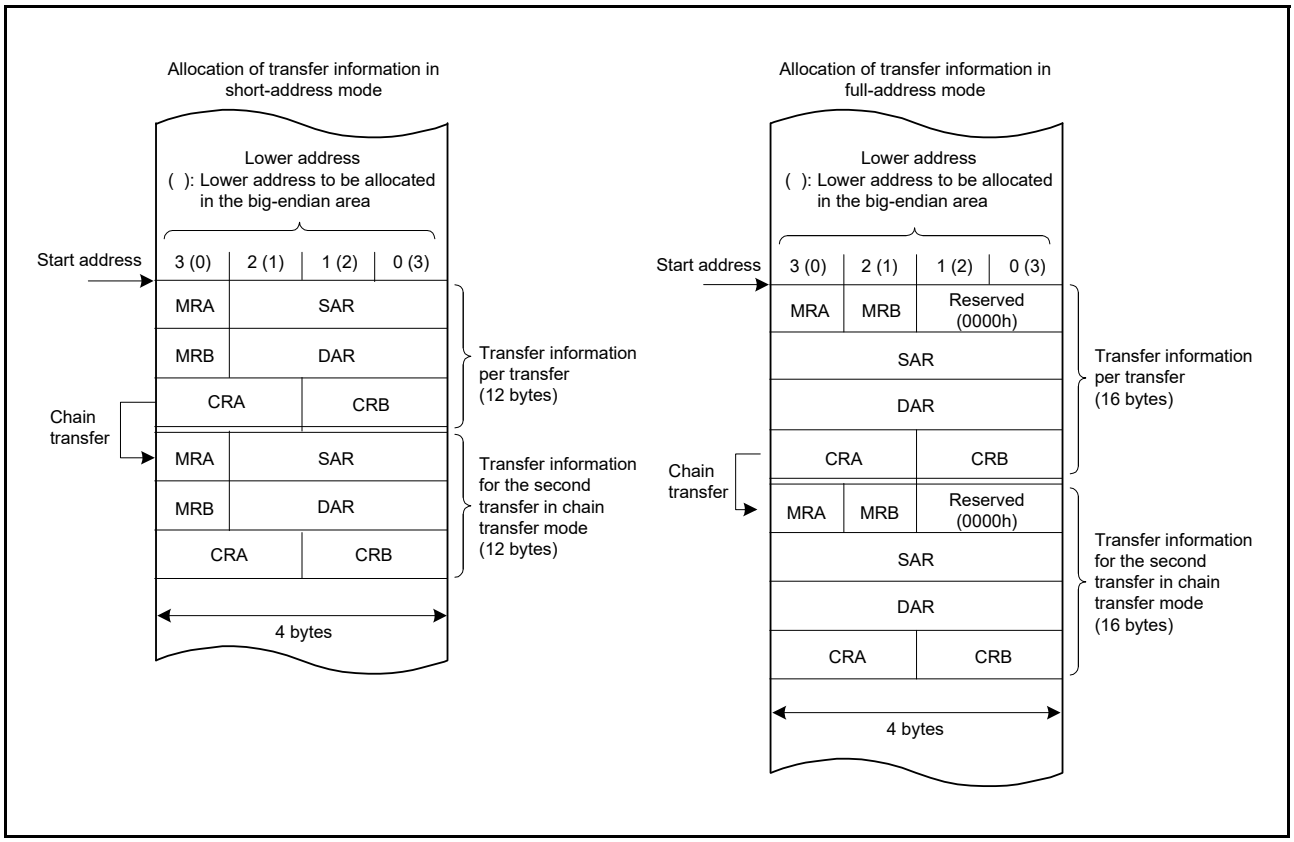


Figure 19.3 Allocation of Transfer Information in the RAM Area

Downloaded from Arrow.com.

19.4 Operation

The DTC transfers data in accordance with the transfer information. Storage of the transfer information in the RAM area is required before DTC operation.

When the DTC accepts a transfer request, it reads the DTC vector corresponding to the vector number. Next, the DTC reads transfer information from the address pointed by the DTC vector, transfers data, and then writes back the transfer information after the data transfer. Allocating transfer information in the RAM area allows data transfer of arbitrary number of channels.

There are three transfer modes: normal transfer mode, repeat transfer mode, and block transfer mode.

Set a transfer source address in the SAR register and a transfer destination address in the DAR register. The SAR and DAR registers are updated after the transfer according to the respective settings (increment, decrement, or fixed).

Table 19.2 lists transfer modes of the DTC.

Table 19.2 Transfer Modes of the DTC

Transfer Mode	Data Size Transferred on Single Transfer Request	Increment/Decrement of Memory Address	Settable Transfer Count
Normal transfer mode	1 byte/1 word/1 longword	Incremented/decremented by 1, 2, or 4 or address fixed	1 to 65536
Repeat transfer mode*1	1 byte/1 word/1 longword	Incremented/decremented by 1, 2, or 4 or address fixed	1 to 256*3
Block transfer mode*2	Block size specified in CRAH (1 to 256 bytes/1 to 256 words/1 to 256 longwords)	Incremented/decremented by 1, 2, or 4 or address fixed	1 to 65536

Note 1. Set transfer source or transfer destination in the repeat area.
Note 2. Set transfer source or transfer destination in the block area.
Note 3. After data transfer of the specified count, the initial state is restored and the operation is continued (repeated).

Setting the MRB.CHNE bit to 1 allows multiple transfers (chain transfer) on a single transfer request. The setting in combination with the MRB.CHNS bit enables a chain transfer when the specified number of data transfers is completed. Figure 19.4 shows the operation flowchart of the DTC. Table 19.3 lists chain transfer conditions.

Downloaded from Arrow.com.

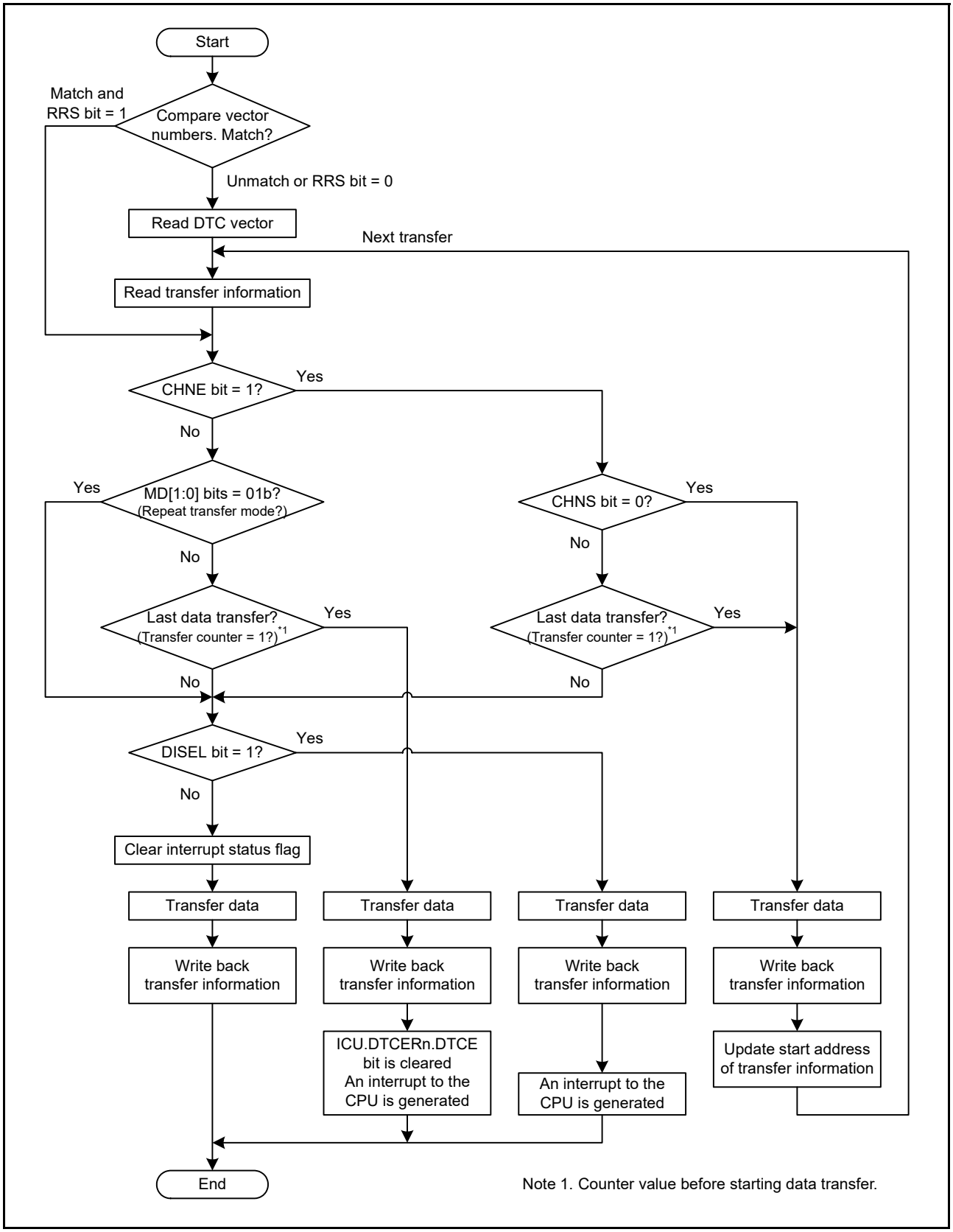


Figure 19.4 Operation Flowchart of the DTC

Downloaded from Arrow.com.

Table 19.3 Chain Transfer Conditions

First Transfer				Second Transfer*3				
CHNE Bit	CHNS Bit	DISEL Bit	Transfer Counter*1,*2	CHNE Bit	CHNS Bit	DISEL Bit	Transfer Counter*1,*2	Data Transfer
0	—	0	Other than (1 → 0)	—	—	—	—	Ends after the first transfer
0	—	0	(1 → 0)	—	—	—	—	Ends after the first transfer with an interrupt request to the CPU
0	—	1	—	—	—	—	—	
1	0	—	—	0	—	0	Other than (1 → 0)	Ends after the second transfer
				0	—	0	(1 → 0)	Ends after the second transfer with an interrupt request to the CPU
				0	—	1	—	
1	1	0	Other than (1 → ∗)	—	—	—	—	Ends after the first transfer
1	1	—	(1 → ∗)	0	—	0	Other than (1 → 0)	Ends after the second transfer
				0	—	0	(1 → 0)	Ends after the second transfer with an interrupt request to the CPU
				0	—	1	—	
1	1	1	Other than (1 → ∗)	—	—	—	—	Ends after the first transfer with an interrupt request to the CPU

Note 1. The transfer counters used depend on transfer modes as follows:
Normal transfer mode: CRA register
Repeat transfer mode: CRAL register
Block transfer mode: CRB register

Note 2. On completion of data transfer, the counters operate as follows:
1 → 0 in normal and block transfer modes
1 → CRAH in repeat transfer mode
(1 → ∗) in the table indicates both of the two operations above.

Note 3. Chain transfer can be selected for the second or subsequent transfers. The condition combination of “second transfer and the CHNE bit is 1” is omitted.

19.4.1 Transfer Information Read Skip Function

Reading of DTC vector and transfer information can be skipped by the setting of the DTCCR.RRS bit.

When a DTC transfer request is accepted, the current DTC vector number is compared with the DTC vector number in the previous data transfer. When these vector numbers match and the RRS bit is 1, the DTC does not read the DTC vector and transfer information, and transfers data according to the transfer information remained in the DTC.

However, when the previous transfer was chain transfer, the DTC vector and transfer information are read. Furthermore, when the transfer counter (CRA register) became 0 during the previous normal transfer and when the transfer counter (CRB register) became 0 during the previous block transfer, transfer information is read regardless of the value of the RRS bit. Figure 19.13 shows an example of transfer information read skip.

When updating the vector table and transfer information, set the RRS bit to 0, update the vector table and transfer information, and then set the RRS bit to 1. Setting the RRS bit to 0 discards the vector numbers retained in the DTC. The updated DTC vector table and transfer information are read in the next data transfer.

Downloaded from Arrow.com.

19.4.2 Transfer Information Write-Back Skip Function

When the MRA.SM[1:0] bits or the MRB.DM[1:0] bits are set to “address is fixed” (00b or 01b), a part of transfer information is not written back. This function is performed independently of the setting of short-address mode or full-address mode.

Table 19.4 lists transfer information write-back skip conditions and applicable registers. The CRA and CRB registers are written back independently of the setting of short-address mode or full-address mode.

Furthermore, in full-address mode, write-back of registers MRA and MRB is skipped.

Table 19.4 Transfer Information Write-Back Skip Conditions and Applicable Registers

MRA.SM[1:0] Bits		MRB.DM[1:0] Bits			
b3	b2	b3	b2	**SAR Register**	**DAR Register**
0	0	0	0	Skip	Skip
0	0	0	1		
0	1	0	0		
0	1	0	1		
0	0	1	0	Skip	Write-back
0	0	1	1		
0	1	1	0		
0	1	1	1		
1	0	0	0	Write-back	Skip
1	0	0	1		
1	1	0	0		
1	1	0	1		
1	0	1	0	Write-back	Write-back
1	0	1	1		
1	1	1	0		
1	1	1	1		

Downloaded from Arrow.com.

19.4.3 Normal Transfer Mode

This mode allows 1-byte, 1-word, or 1-longword data transfer on a single transfer request. The transfer count can be set to 1 to 65536.

Transfer source addresses and transfer destination addresses can be set to increment, decrement, or fixed independently.

This mode enables an interrupt request to the CPU to be generated at the end of specified-count transfer.

Table 19.5 lists register functions in normal transfer mode, and Figure 19.5 shows the memory map of normal transfer mode.

Table 19.5 Register Functions in Normal Transfer Mode

Register	Description	Value Written Back by Writing Transfer Information
SAR	Transfer source address	Increment/decrement/fixed*1
DAR	Transfer destination address	Increment/decrement/fixed*1
CRA	Transfer counter A	CRA – 1
CRB	Transfer counter B	Not updated

Note 1. Write-back operation is skipped when address is fixed.

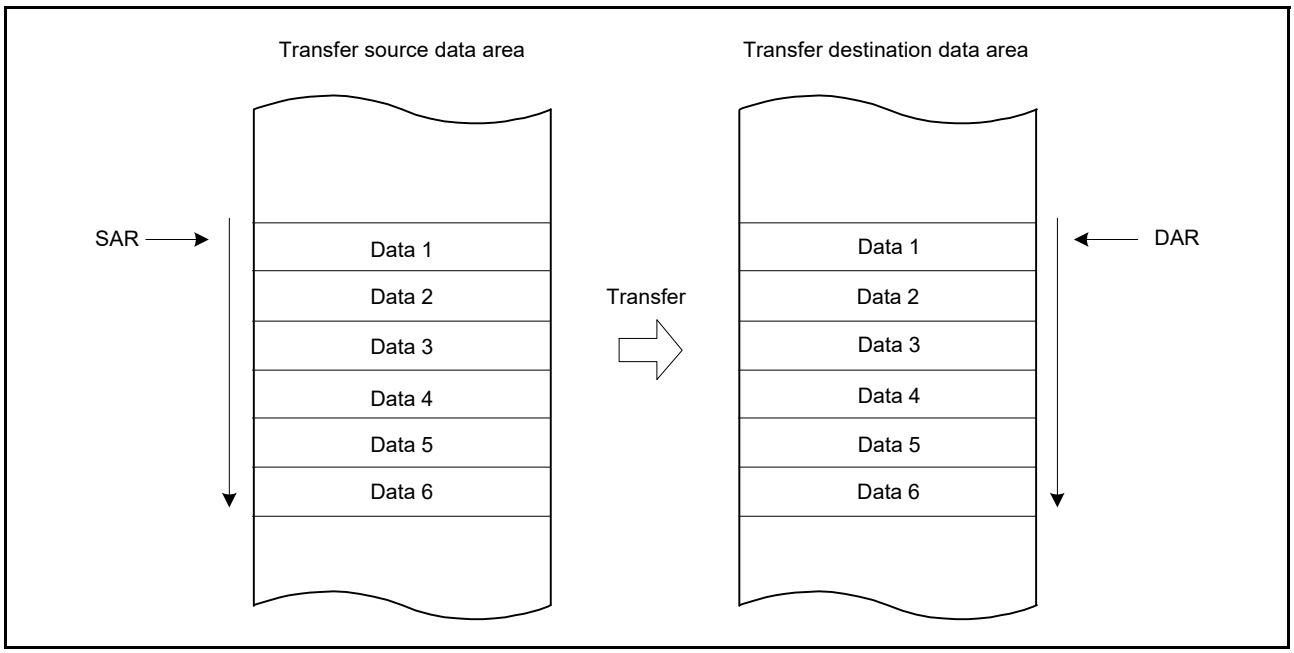


Figure 19.5 Memory Map of Normal Transfer Mode

Downloaded from Arrow.com.

19.4.4 Repeat Transfer Mode

This mode allows 1-byte, 1-word, or 1-longword data transfer on a single transfer request.
Specify either transfer source or transfer destination for the repeat area by the MRB.DTS bit. The transfer count can be set to 1 to 256. When the specified-count transfer is completed, the initial value of the address register specified in the transfer counter and the repeat area is restored and transfer is repeated. The other address register is incremented or decremented continuously or remains unchanged.
When the transfer counter CRAL is decreased to 00h in repeat transfer mode, the CRAL value is updated to the value set in the CRAH register. Thus the transfer counter does not become 00h, which disables an interrupt request to be generated to the CPU when the MRB.DISEL bit is set to 0 (an interrupt request to the CPU is generated on completion of the specified number of data transfers).
Table 19.6 lists the register functions in repeat transfer mode, and Figure 19.6 shows the memory map of repeat transfer mode.

Table 19.6 Register Functions in Repeat Transfer Mode

		Value Written Back by Writing Transfer Information		
			When CRAL = 1	
Register	**Description**	**When CRAL ≠ 1**	**When the MRB.DTS Bit is 0**	**When the MRB.DTS Bit is 1**
SAR	Transfer source address	Increment/decrement/fixed*1	Increment/decrement/fixed*1	SAR register initial value
DAR	Transfer destination address	Increment/decrement/fixed*1	DAR register initial value	Increment/decrement/fixed*1
CRAH	Retains initial value of transfer counter	CRAH	CRAH	
CRAL	Transfer counter A	CRAL – 1	CRAH	
CRB	Transfer counter B	Not updated	Not updated	

Note 1. Write-back operation is skipped when address is fixed.

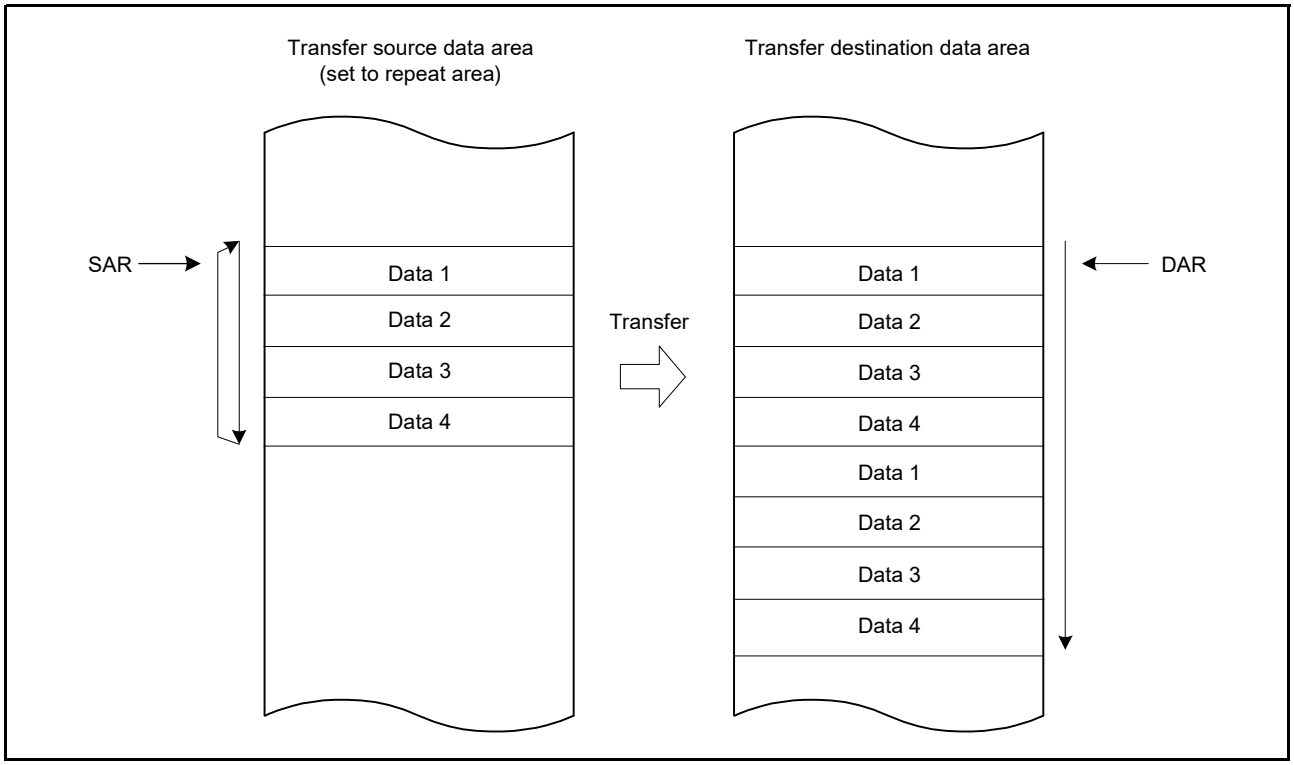


Figure 19.6 Memory Map of Repeat Transfer Mode (Transfer Source: Repeat Area)

Downloaded from Arrow.com.

19.4.5 Block Transfer Mode

This mode allows single-block data transfer on a single transfer request.
Specify either transfer source or transfer destination for the block area by the MRB.DTS bit. The block size can be set to 1 to 256 bytes, 1 to 256 words, or 1 to 256 longwords.
When transfer of the specified one block is completed, the initial values of the block size counter CRAL and the address register (the SAR register when the MRB.DTS bit is 1 or the DAR register when the DTS bit is 0) specified in the block area are restored. The other address register is incremented or decremented continuously or remains unchanged.
The transfer count (block count) can be set to 1 to 65536. This mode enables an interrupt request to the CPU to be generated at the end of specified-count block transfer.
Table 19.7 lists register functions in block transfer mode, and Figure 19.7 shows the memory map of block transfer mode.

Table 19.7 Register Functions in Block Transfer Mode

		Value Written Back by Writing Transfer Information	
Register	**Description**	**When MRB.DTS Bit is 0**	**When MRB.DTS Bit is 1**
SAR	Transfer source address	Increment/decrement/fixed*1	SAR register initial value
DAR	Transfer destination address	DAR register initial value	Increment/decrement/fixed*1
CRAH	Retains initial value of block size	CRAH	
CRAL	Block size counter	CRAH	
CRB	Block transfer counter	CRB – 1	

Note 1. Write-back operation is skipped when address is fixed.

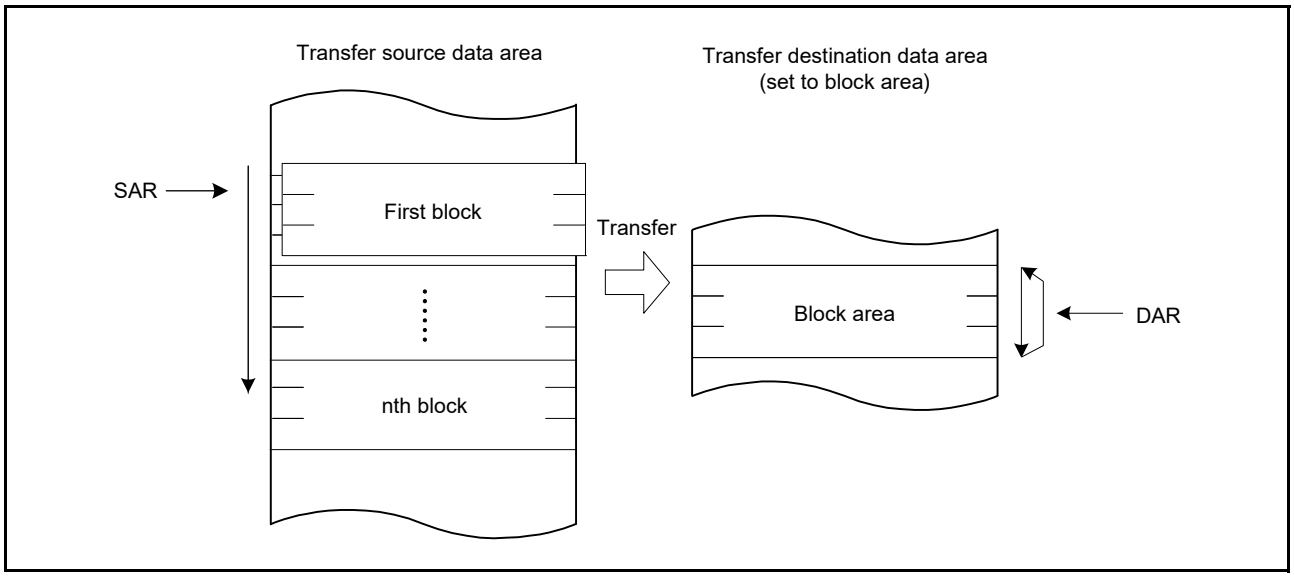


Figure 19.7 Memory Map of Block Transfer Mode (Transfer Destination: Block Area)

Downloaded from Arrow.com.

19.4.6 Chain Transfer

Setting the MRB.CHNE bit to 1 allows chain transfer to be performed continuously on a single transfer request. If the MRB.CHNE bit is 1 and the MRB.CHNS bit is 0, an interrupt request to the CPU is not generated when the specified number of data transfers is completed, or while the MRB.DISEL bit is 1 (an interrupt request to the CPU is generated for every data transfer). Data transfer has no effect on the interrupt status flag, which is the request source. The transfer information (SAR, DAR, CRA, CRB, MRA, and MRB) that define a data transfer can be specified independently of each other. Figure 19.8 shows chain transfer operation.

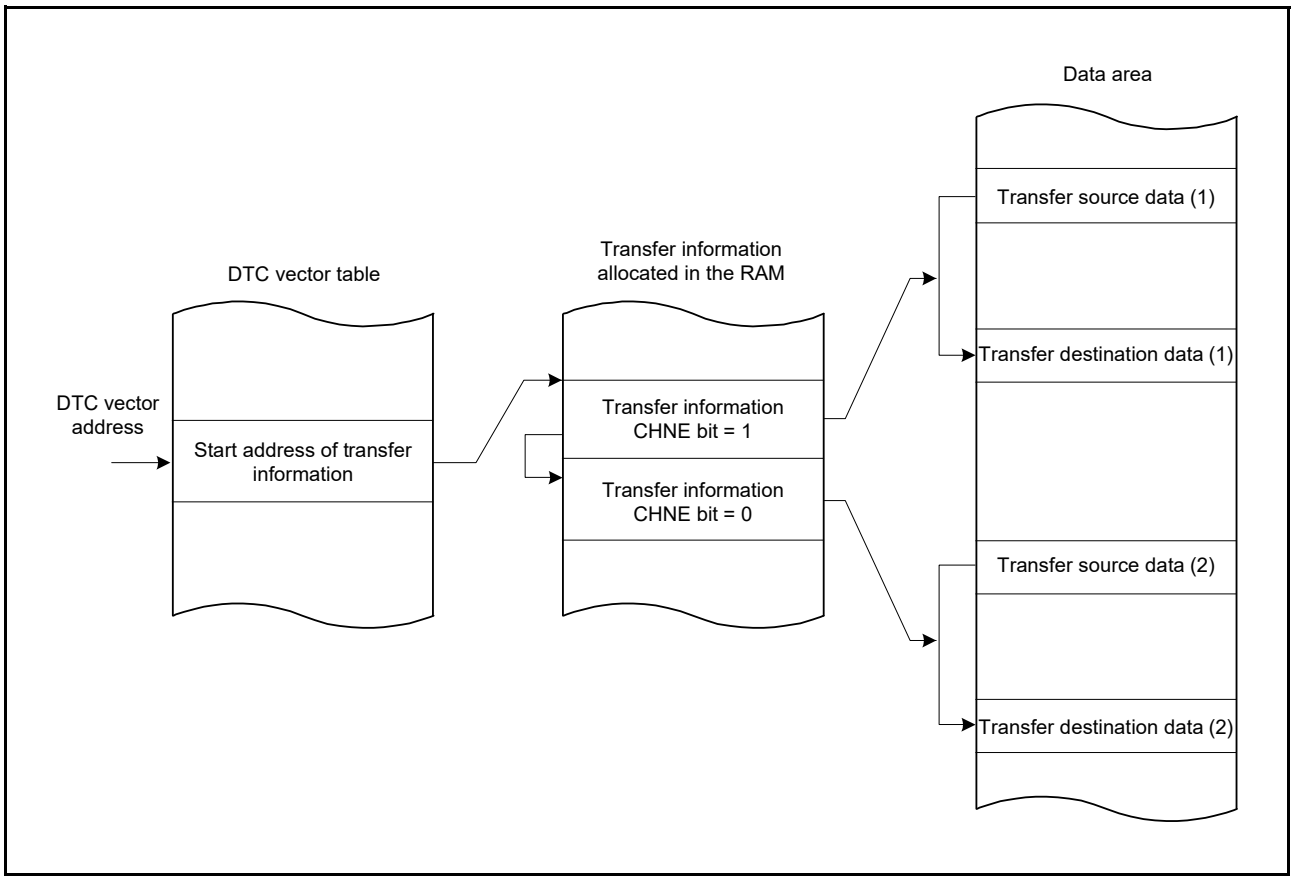


Figure 19.8 Chain Transfer Operation

If the MRB.CHNE bit is 1 and the CHNS bit is 1, chain transfer is performed only after completion of specified number of data transfers. In repeat transfer mode, chain transfer is performed after completion of specified number of data transfers.

For details on chain transfer conditions, refer to Table 19.3, Chain Transfer Conditions.

Downloaded from Arrow.com.

19.4.7 Operation Timing

Figure 19.9 to Figure 19.13 show examples of DTC operation timing.

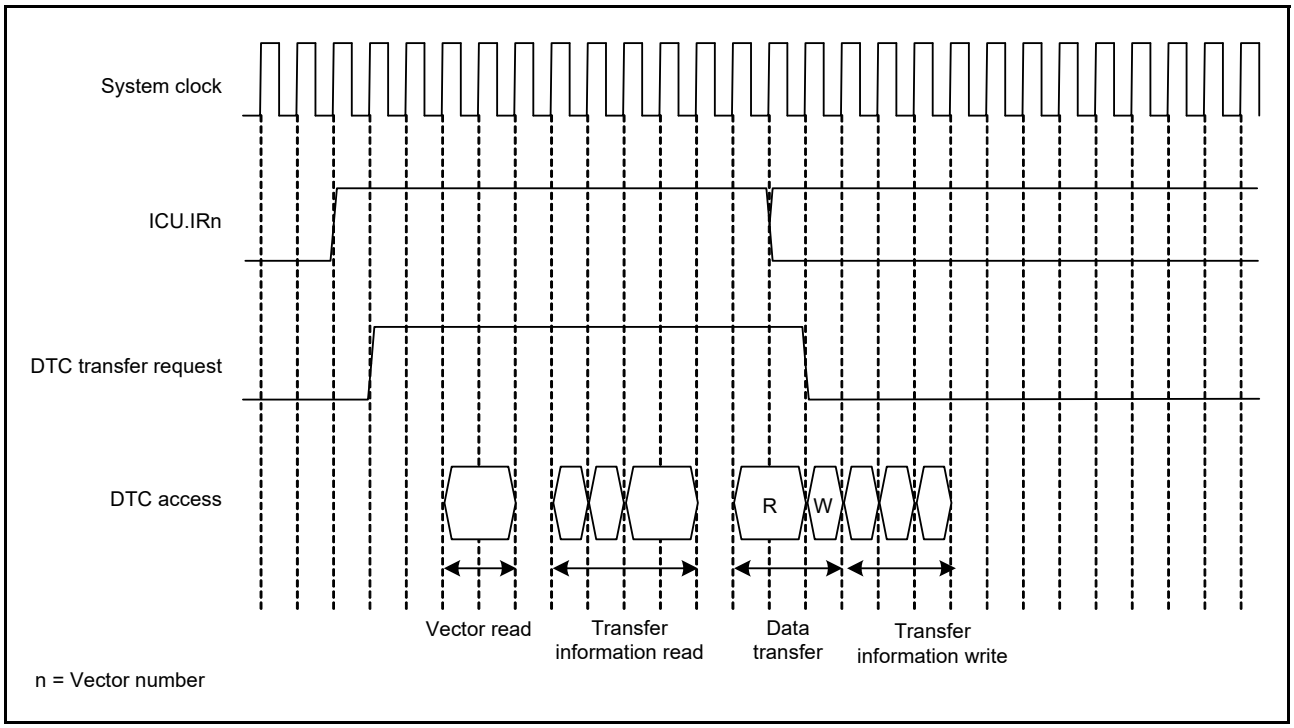


Figure 19.9 Example (1) of DTC Operation Timing (Short-Address Mode, Normal Transfer Mode, Repeat Transfer Mode)

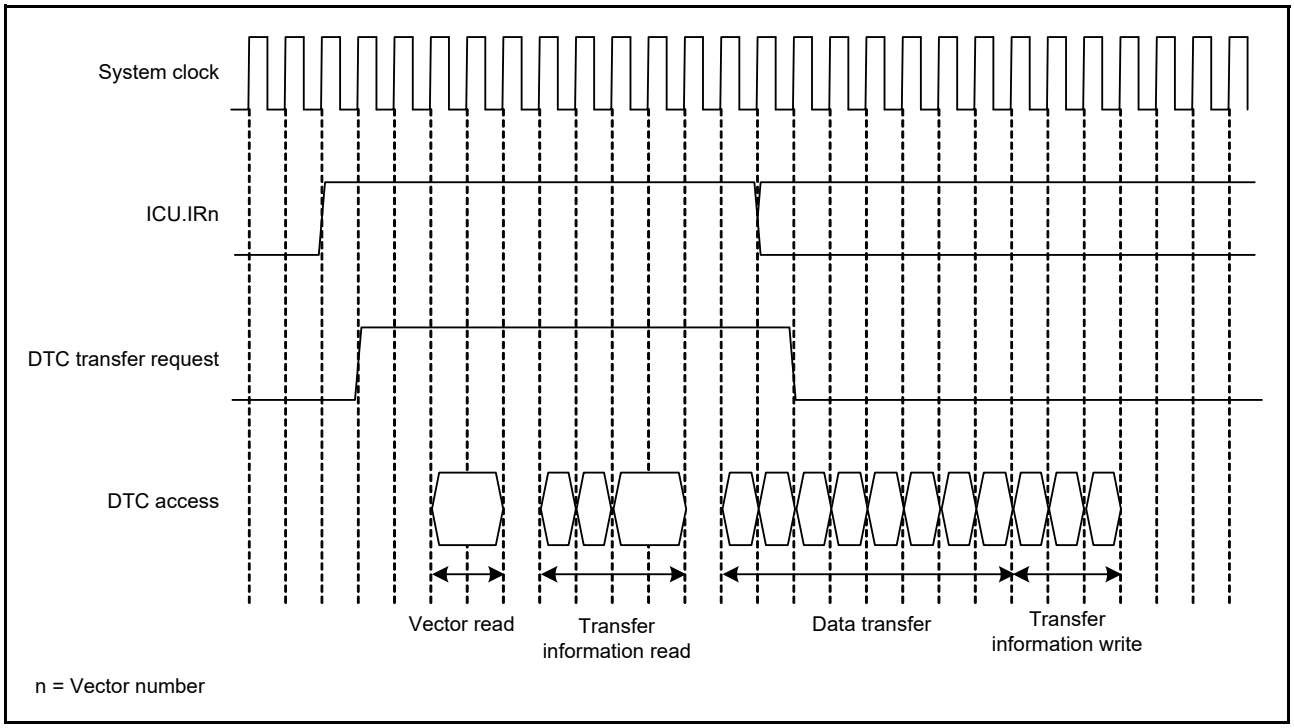


Figure 19.10 Example (2) of DTC Operation Timing (Short-Address Mode, Block Transfer Mode, Block Size = 4)

Downloaded from Arrow.com.

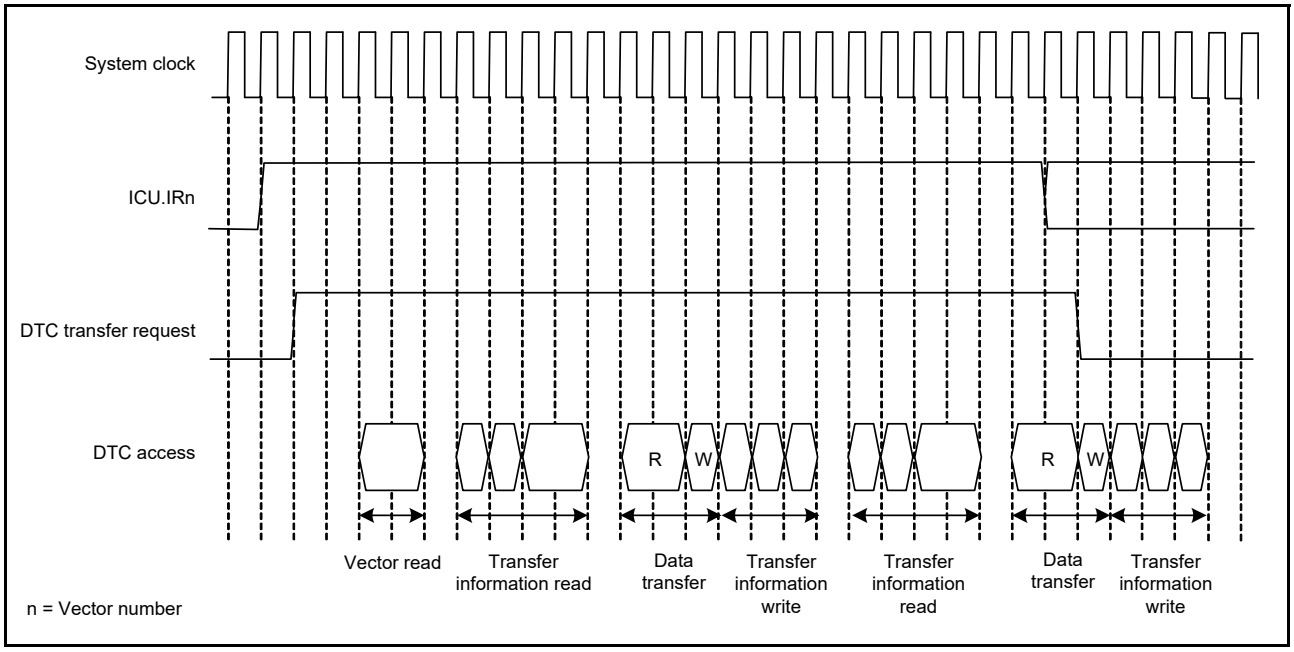


Figure 19.11 Example (3) of DTC Operation Timing (Short-Address Mode, Chain Transfer)

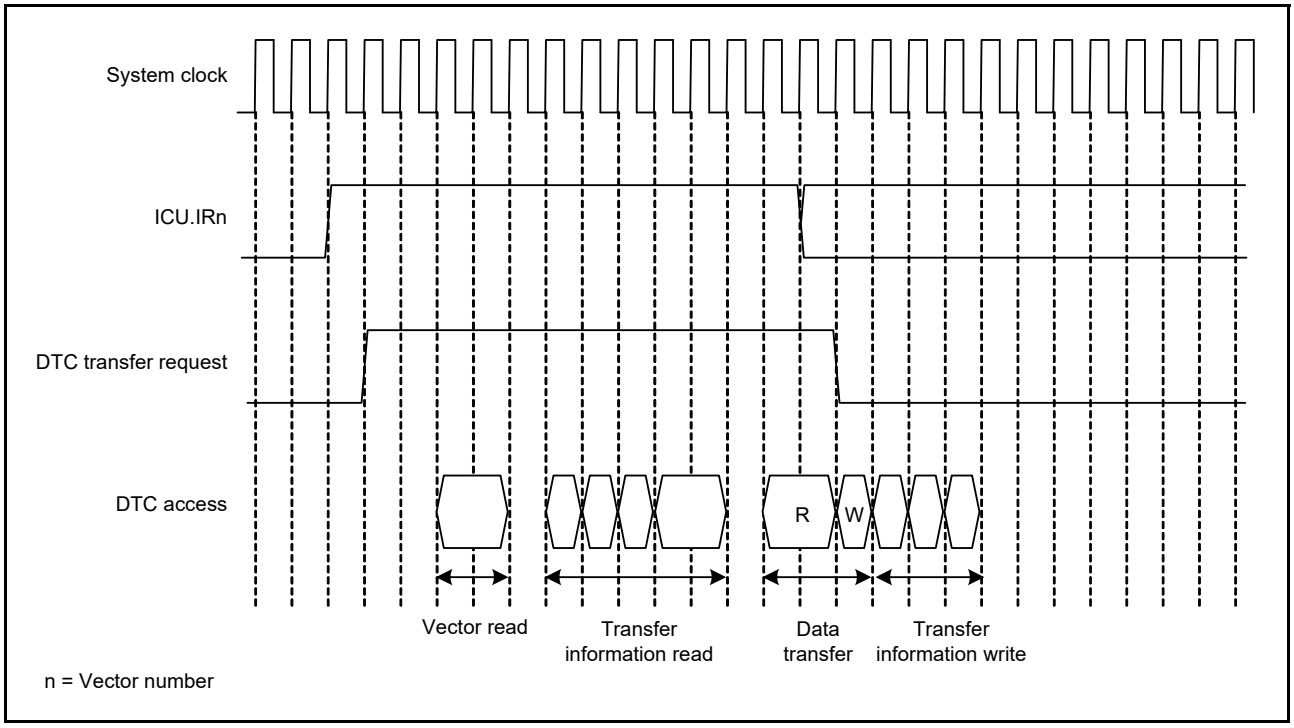


Figure 19.12 Example (4) of DTC Operation Timing (Full-Address Mode, Normal Transfer Mode, Repeat Transfer Mode)

Downloaded from Arrow.com.

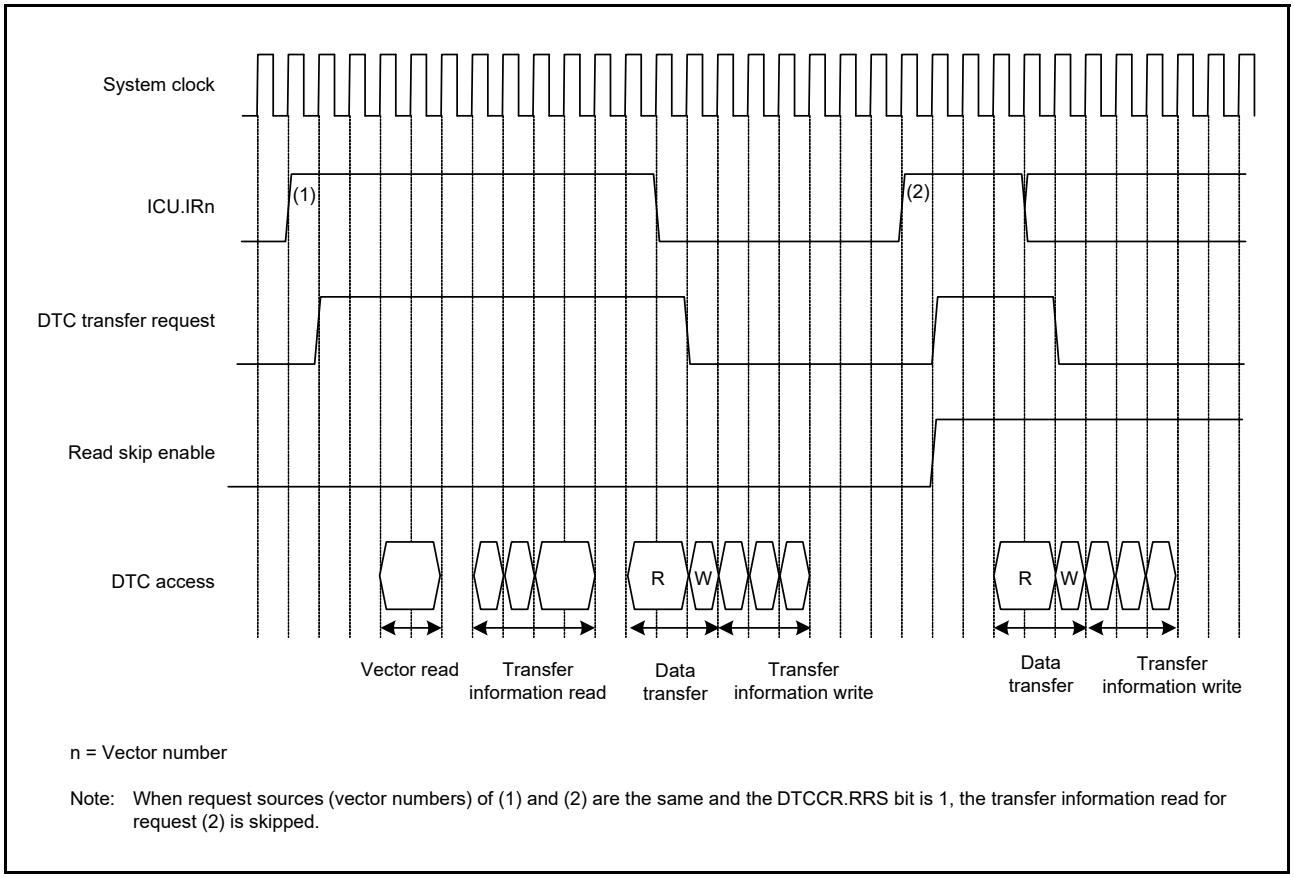


Figure 19.13 Example of Operation When Transfer Information Read Skip is Executed (Vector, Transfer Information, and Transfer Destination Data on the RAM, and Transfer Source Data on the Peripheral Module)

Downloaded from Arrow.com.

19.4.8 Execution Cycles of the DTC

Table 19.8 lists the execution cycles of single data transfer of the DTC.
For the order of the execution states, refer to section 19.4.7, Operation Timing.

Table 19.8 Execution Cycles of the DTC

Transfer Mode	Vector Read		Transfer Information Read			Transfer Information Write			Data Transfer		Internal Operation	
									Read	Write		
Normal	Cv + 1	0*1	4 × Ci + 1*2	3 × Ci + 1*3	0*1	3 × Ci*4	2 × Ci*5	Ci*6	Cr + 1	Cw	2	0*1
Repeat									Cr + 1	Cw		
Block*7									P × Cr	P × Cw		

Note 1. When transfer information read is skipped
Note 2. In full-address mode
Note 3. In short-address mode
Note 4. When neither SAR nor DAR is set to address-fixed
Note 5. When SAR or DAR is set to address-fixed
Note 6. When SAR and DAR are set to address-fixed
Note 7. When the block size is 2 or more. If the block size is 1, the cycle number for normal transfer is applied.

P: Block size (initial settings of CRAH and CRAL)
Cv: Cycles for access to vector transfer information storage destination
Ci: Cycles for access to transfer information storage destination address
Cr: Cycles for access to data read destination
Cw: Cycles for access to data write destination
(The unit is system clocks (ICLK) for “+ 1” in the Vector Read, Transfer Information Read, and Data Transfer Read columns and “2” in the Internal Operation column.)
(Cv, Ci, Cr, and Cw vary depending on the corresponding access destination. For the number of cycles for respective access destinations, refer to section 49, RAM, section 50, Flash Memory (FLASH), and section 5, I/O Registers.)

19.4.9 DTC Bus Mastership Release Timing

The DTC does not release the bus mastership during transfer information read and transfer information write. While transfer information is not read or written, bus arbitration is made according to the priority determined by the bus master arbitrator.
For bus arbitration, refer to section 16, Buses.

Downloaded from Arrow.com.

19.5 DTC Setting Procedure

Before using the DTC, set the DTC vector base register (DTCVBR).
Figure 19.14 shows the procedure to set the DTC.

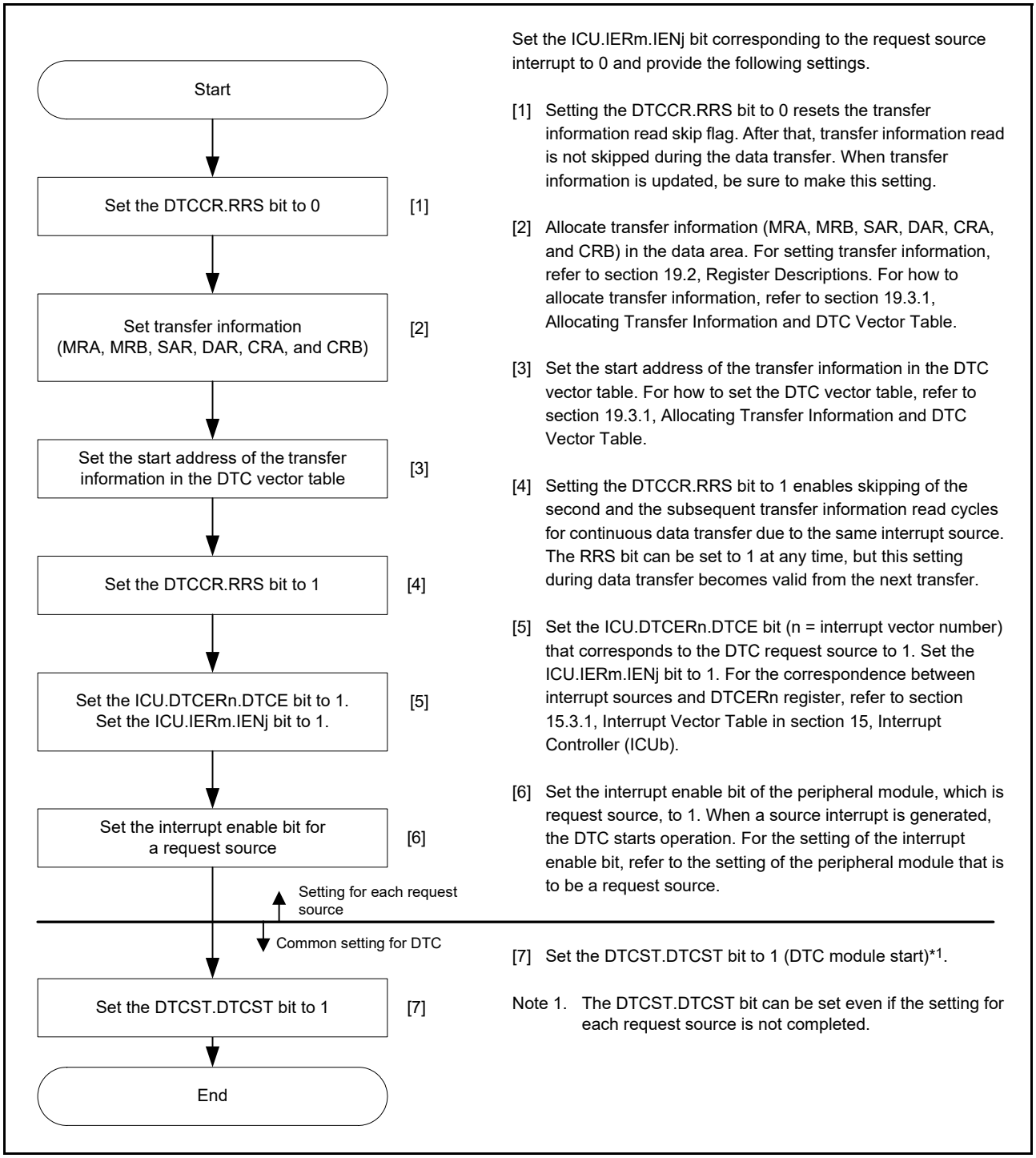


Figure 19.14 Procedure to Set the DTC

Downloaded from Arrow.com.

19.6 Examples of DTC Usage

19.6.1 Normal Transfer

As an example of DTC usage, its employment in the reception of 128 bytes of data by an SCI is described below.

(1) Transfer Information Setting

Set the MRA.MD[1:0] bits to 00b (normal transfer mode), the MRA.SZ[1:0] bits to 00b (byte transfer), and the MRA.SM[1:0] bits to 00b (source address is fixed). Set the MRB.CHNE bit to 0 (chain transfer is disabled), the MRB.DISEL bit to 0 (an interrupt request to the CPU is generated on completion of the specified number of data transfers), and the MRB.DM[1:0] bits to 10b (DAR is incremented after data transfer). The MRB.DTS bit can be set to any value. Set the RDR register address of the SCI in the SAR register, the start address of the RAM area for data storage in the DAR register, and 128 (0080h) in the CRA register. The CRB register can be set to any value.

(2) DTC Vector Table Setting

The start address of the transfer information for the RXI interrupt is set in the vector table for the DTC.

(3) ICU Setting and DTC Module Activation

Set the corresponding ICU.DTCERn.DTCE bit to 1 and the ICU.IERm.IENj bit to 1.
Set the DTCST.DTCST bit to 1.

(4) SCI Setting

Enable the RXI interrupt by setting the SCR.RIE bit in the SCI to 1. If a reception error occurs during the SCI receive operation, further reception is not performed. Accordingly, make settings so that the CPU can accept receive error interrupts.

(5) DTC Transfer

Every time the reception of 1 byte by the SCI is completed, an RXI interrupt is generated to start the data transfer. The DTC transfers the received byte from the RDR of the SCI to RAM, after which the DAR register is incremented and the CRA register is decremented.

(6) Interrupt Handling

After 128 times of data transfers have been completed and the value in the CRA register becomes 0, an RXI interrupt request is output to the CPU. Complete the process in the handling routine for this interrupt.

Downloaded from Arrow.com.

19.6.2 Chain Transfer When the Counter is 0

The second data transfer is performed only when the transfer counter is set to 0 in the first data transfer, and the first data transfer information is repeatedly changed in the second data transfer. Repeating this chain transfer enables transfers to be repeated more than 256 times.
The following shows an example of configuring a 128-Kbyte input buffer to addresses 20 0000h to 21 FFFFh (where the input buffer is set so that its lower address starts with 0000h). Figure 19.15 shows a chain transfer when the counter is 0.

(1) Set normal transfer mode for input data for the first data transfer. Set the following:
Transfer source address: Fixed, the CRA register is 0000h (65,536 times), the MRB.CHNE bit is 1 (chain transfer is enabled), the MRB.CHNS bit is 1 (chain transfer is performed only when the transfer counter becomes 0), and the MRB.DISEL bit is 0 (an interrupt request to the CPU is generated on completion of the specified number of data transfers).
(2) Prepare the upper 8 bits (in this case, 21h and 20h) of the start address at every 65,536 times of the transfer destination address for the first data transfer in another area (such as ROM).
(3) For the second data transfer, set repeat transfer mode (source is repeat area) for rewriting the transfer destination address of the first data transfer. The transfer destination is the address where the upper 8 bits of the DAR register in the first transfer information is allocated. In this case, set the MRB.CHNE bit to 0 (chain transfer is disabled) and the MRB.DISEL bit to 0 (an interrupt request to the CPU is generated on completion of the specified number of data transfers). In this case, set the transfer counter to 2.
(4) When a transfer request is accepted, the first data transfer is executed. When transfer is executed 65,536 times and the transfer counter of the first data transfer becomes 0, the second data transfer is started and the upper 8 bits of the transfer destination address of the first data transfer is set to 21h. At this time, the lower 16 bits of the transfer destination address and the transfer counter of the first data transfer have become 0000h.
(5) In succession, when another transfer request is accepted, the first data transfer is executed. When transfer is executed 65,536 times and the transfer counter of the first data transfer becomes 0, the second data transfer is started and the upper 8 bits of the transfer destination address of the first data transfer is set to 20h. At this time, the lower 16 bits of the transfer destination address and the transfer counter of the first data transfer have become 0000h.
(6) Steps (4) and (5) above are repeated infinitely. Because the second data transfer is in repeat transfer mode, no interrupt request to the CPU is generated.

Downloaded from Arrow.com.

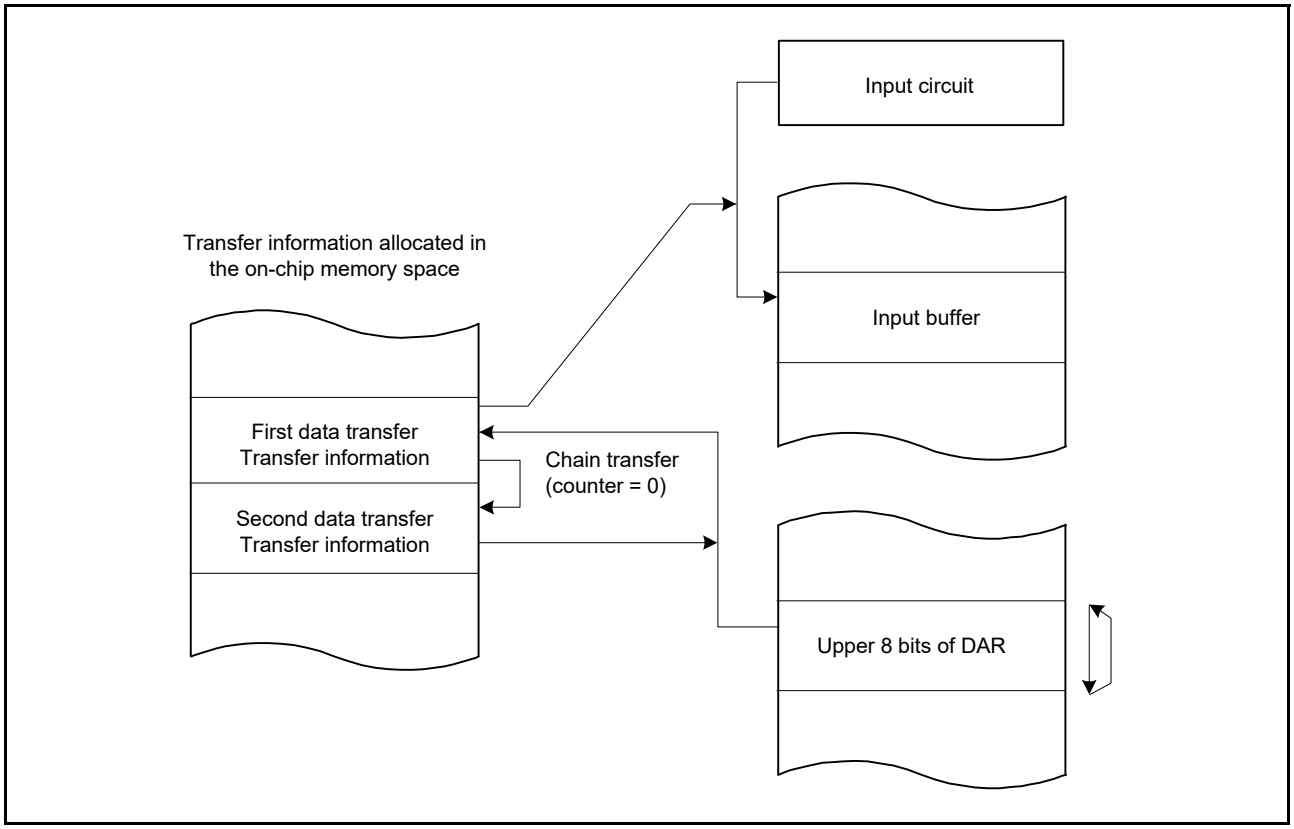


Figure 19.15 Chain Transfer When the Counter is 0

19.7 Interrupt Source

When the DTC has finished data transfer of specified count or when data transfer with the MRB.DISEL bit set to 1 (an interrupt request to the CPU is generated each time the data transfer is performed) has been completed, an interrupt to the CPU is generated by the DTC trigger source. Such interrupts to the CPU are controlled according to the PSW.I bit (interrupt enable) of the CPU, the PSW.IPL[3:0] bits (processor interrupt priority level), and the priority level of the interrupt controller.

19.8 Event Link

The DTC outputs an event signal on completing data transfer in response to one request.

Downloaded from Arrow.com.

19.9 Low Power Consumption Function

Before making a transition to the module stop state, deep sleep mode, or software standby mode, set the DTCST.DTCST bit to 0 (DTC module stop), and then perform the following.

(1) Module Stop Function

Writing 1 (transition to the module-stop state is made) to the MSTPCRA.MSTPA28 bit enables the module stop function of the DTC. If data transfer is in progress at the time 1 is written to the MSTPCRA.MSTPA28 bit, the transition to the module stop state proceeds after data transfer has ended. While the MSTPCRA.MSTPA28 bit is 1, accessing the DTC registers is prohibited.
Writing 0 (release from the module-stop state) to the MSTPCRA.MSTPA28 bit releases the DTC from the module stop state.

(2) Deep Sleep Mode

Make settings according to the procedure under section 11.6.2.1, Entry to Deep Sleep Mode, in section 11, Low Power Consumption.
If any data transfer is in progress at the time the WAIT instruction is executed, the transition to deep sleep mode follows the completion of the data transfer.
The DTC is released from the module stop state by writing 0 to the MSTPCRA.MSTPA28 bit following recovery from deep sleep mode.

(3) Software Standby Mode

Make settings according to the procedure under section 11.6.3.1, Entry to Software Standby Mode, in section 11, Low Power Consumption.
If any data transfer is in progress at the time the WAIT instruction is executed, the transition to software standby mode follows the completion of the data transfer.

(4) Notes on Low Power Consumption Function

For the WAIT instruction and the register setting procedure, refer to section 11.7.5, Timing of WAIT Instructions in section 11, Low Power Consumption.
To perform data transfer after returning from a low power consumption mode, set the DTCST.DTCST bit to 1 again.
To use a request that is generated in deep sleep mode or software standby mode as an interrupt request to the CPU but not as a DTC transfer request, specify the CPU as the interrupt request destination according to the description in section 15.4.3, Selecting Interrupt Request Destinations in section 15, Interrupt Controller (ICUb), and then execute the WAIT instruction.

Downloaded from Arrow.com.

19.10 Usage Notes

19.10.1 Start Address of Transfer Information

Set multiples of 4 for the start addresses of the transfer information to be specified in the DTC vector table. If any value other than a multiple of 4 is specified, access still proceeds with the lower 2 bits of the address regarded as 00b.

19.10.2 Allocating Transfer Information

Allocate transfer information in the memory area according to the endian of the area as shown in Figure 19.16. For example, when writing CRA and CRB settings in 16-bit units in big endian, write the CRA setting to the address plus 8h (Ch) and the CRB setting to the address plus Ah (Eh). In little endian, write the CRB setting to the address plus 8h (Ch) and the CRA setting to the address plus Ah (Eh). When writing CRA and CRB settings in 32-bit units, allocate the CRA setting at the MSB side of the 32 bits and the CRB setting at the LSB side, and write the settings to the address plus 8h (Ch), regardless of endian.

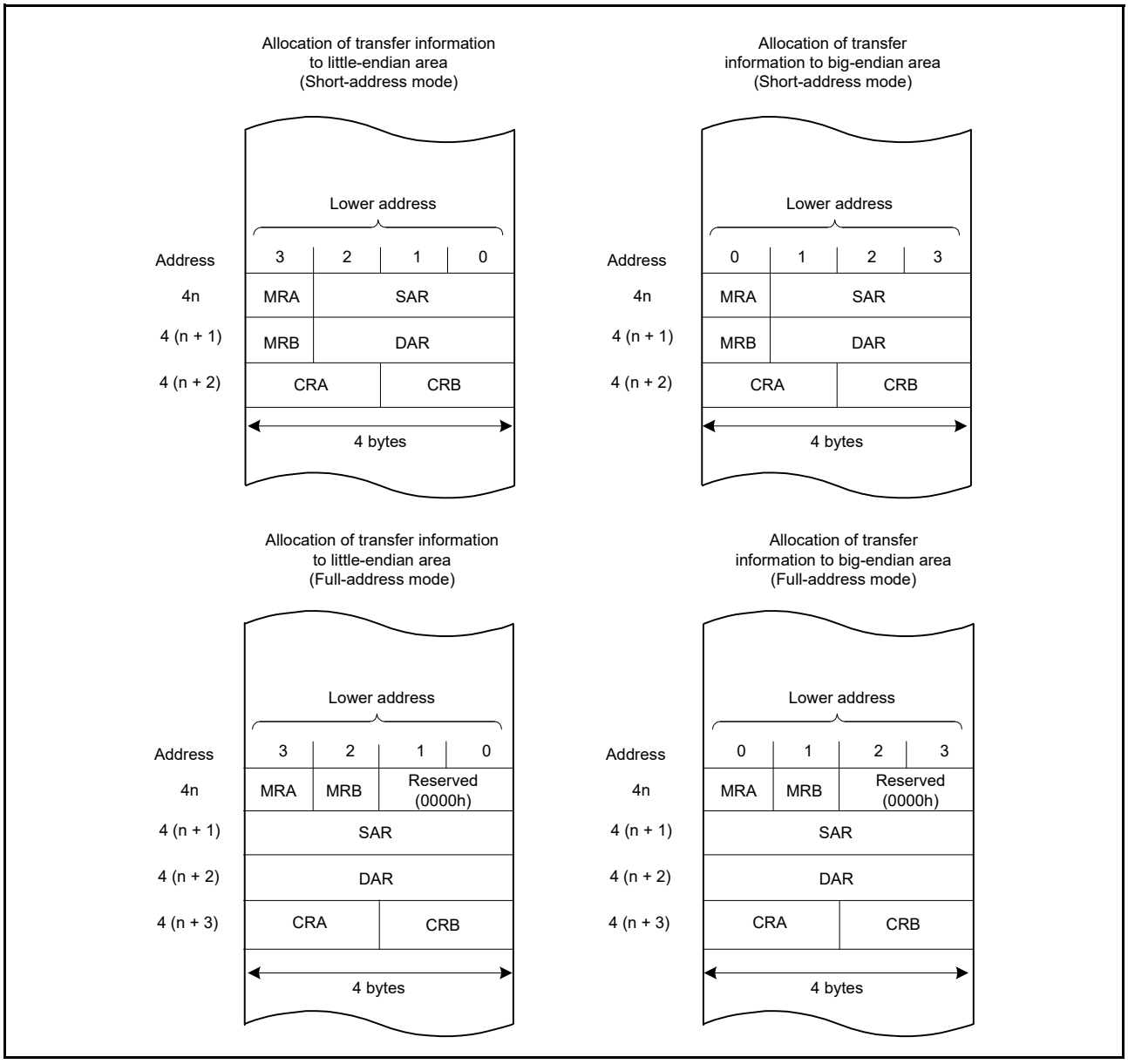


Figure 19.16 Allocation of Transfer Information

Downloaded from Arrow.com.

19.10.3 Setting the DTC Transfer Request Enable Register in the Interrupt Controller (ICU.DTCERn)

The DMA request should not be issued by setting the DMAC trigger select register (ICU.DMRSRm (m = DMAC channel number)) to the same vector number that has been specified by setting the ICU.DTCERn.DTCE bit to 1 (the corresponding interrupt source is selected as the DTC trigger). For details on the ICU.DTCERn and ICU.DMRSRm registers (m = DMAC channel number), refer to section 15, Interrupt Controller (ICUb).

Downloaded from Arrow.com.

20. Event Link Controller (ELC)

20.1 Overview

The event link controller (ELC) uses the interrupt requests generated by various peripheral modules as event signals, and interconnects (links) peripheral modules. As a result, peripheral modules can directly perform interlinked operation among them without using software.
Event signals can be output regardless of the settings of the corresponding interrupt request enable bits.

Table 20.1 lists the specifications of the ELC, and Figure 20.1 shows a block diagram of the ELC.

Table 20.1 ELC Specifications

Item	Description
Event link function	• 60 types of event signals can be directly interconnected to modules. • Operation for timer modules when inputting an event signal can be selected. • Event linkage operation is possible for port B and port E. Single port*1: Event linkage operation can be set in a single specified port. Port group*1: Event linkage operation can be set by grouping multiple specified ports among total of eight ports.
Low power consumption function	Module stop state can be set.

Note 1. When an input signal to a corresponding pin changes, an event is generated in a single port or in a port group specified as the input.

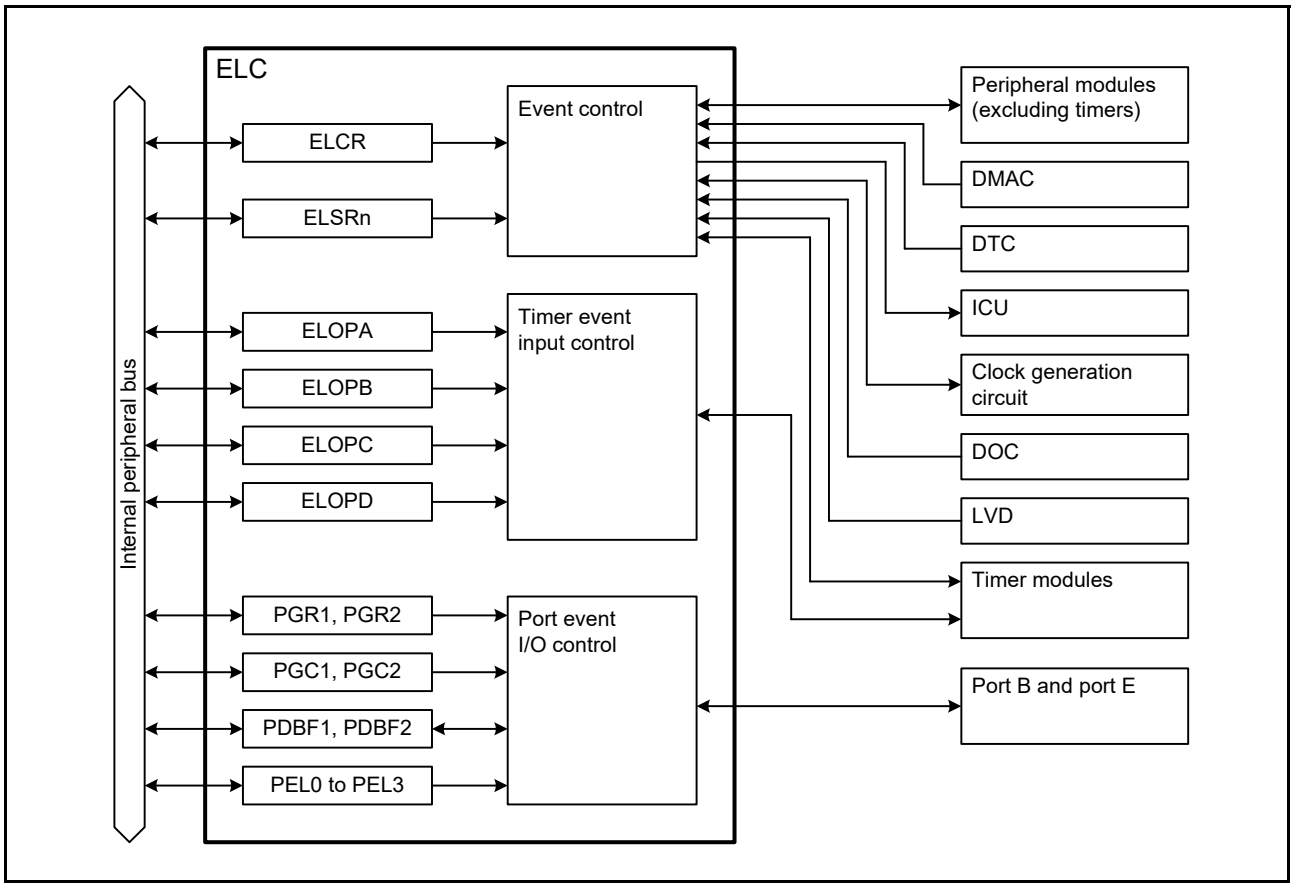


Figure 20.1 ELC Block Diagram (n = 1 to 4, 7, 8, 10, 12, 14 to 16, 18 to 29)

Downloaded from Arrow.com.

20.2 Register Descriptions

20.2.1 Event Link Control Register (ELCR)

Address(es): ELC.ELCR 0008 B100h

	b7	b6	b5	b4	b3	b2	b1	b0
	ELCON	—	—	—	—	—	—	—
Value after reset:	0	1	1	1	1	1	1	1

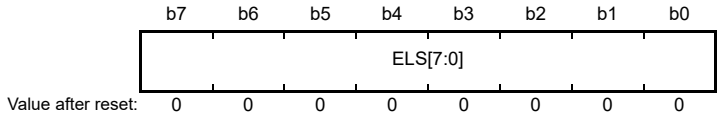
Bit	Symbol	Bit Name	Description	R/W
b6 to b0	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b7	ELCON	All Event Link Enable	0: ELC function is disabled. 1: ELC function is enabled.	R/W

The ELCR register controls operation of the ELC.

Downloaded from Arrow.com.

20.2.2 Event Link Setting Register n (ELSRn) (n = 1 to 4, 7, 8, 10, 12, 14 to 16, 18 to 29)

Address(es): ELC.ELSR1 0008 B102h, ELC.ELSR2 0008 B103h, ELC.ELSR3 0008 B104h, ELC.ELSR4 0008 B105h, ELC.ELSR7 0008 B108h, ELC.ELSR8 0008 B109h, ELC.ELSR10 0008 B10Bh, ELC.ELSR12 0008 B10Dh, ELC.ELSR14 0008 B10Fh, ELC.ELSR15 0008 B110h, ELC.ELSR16 0008 B111h, ELC.ELSR18 0008 B113h, ELC.ELSR19 0008 B114h, ELC.ELSR20 0008 B115h, ELC.ELSR21 0008 B116h, ELC.ELSR22 0008 B117h, ELC.ELSR23 0008 B118h, ELC.ELSR24 0008 B119h, ELC.ELSR25 0008 B11Ah, ELC.ELSR26 0008 B11Bh, ELC.ELSR27 0008 B11Ch, ELC.ELSR28 0008 B11Dh, ELC.ELSR29 0008 B11Eh



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	ELS[7:0]	Event Link Select	00h: Event signal output to the corresponding peripheral module is disabled. 08h to 6Ah: Set the number for the event signal to be linked. Settings other than above are prohibited.	R/W

The ELSRn register specifies an event signal to be linked to for each peripheral module. Table 20.2 shows the correspondence between the ELSRn register and the peripheral modules. Table 20.3 shows the correspondence between values set in the ELSRn register and event signals.

Table 20.2 Correspondence between the ELSRn Register and the Peripheral Modules

Register Name	Peripheral Module
ELSR1	MTU1
ELSR2	MTU2
ELSR3	MTU3
ELSR4	MTU4
ELSR7	CMT1
ELSR8	ICU (LPT dedicated interrupt)*1
ELSR10	TMR0
ELSR12	TMR2
ELSR14	CTSU
ELSR15	S12AD
ELSR16	DA0
ELSR18	ICU (Interrupt 1)*2
ELSR19	ICU (Interrupt 2)*2
ELSR20	Output port group 1
ELSR21	Output port group 2
ELSR22	Input port group 1
ELSR23	Input port group 2
ELSR24	Single port 0*3
ELSR25	Single port 1*3
ELSR26	Single port 2*3
ELSR27	Single port 3*3
ELSR28	Clock source switching to LOCO
ELSR29	POE

Note 1. Specify an event number to 32h (LPT compare match 0).
Note 2. Specify an event number from among 63h to 6Ah. Do not set other values.
Note 3. Do not set the DOC data operation condition met signal (6Ah) in the ELSR24, ELSR25, ELSR26, and ELSR27 registers.

Downloaded from Arrow.com.

Table 20.3 Correspondence between Values Set in ELSRn.ELS[7:0] Bits and Event Signals (1/2)

ELS[7:0] Bit Value	Peripheral Modules	Event Signal Set in ELSRn
08h	Multifunction timer pulse unit 2	MTU1 compare match 1A
09h		MTU1 compare match 1B
0Ah		MTU1 overflow
0Bh		MTU1 underflow
0Ch		MTU2 compare match 2A
0Dh		MTU2 compare match 2B
0Eh		MTU2 overflow
0Fh		MTU2 underflow
10h		MTU3 compare match 3A
11h		MTU3 compare match 3B
12h		MTU3 compare match 3C
13h		MTU3 compare match 3D
14h		MTU3 overflow
15h		MTU4 compare match 4A
16h		MTU4 compare match 4B
17h		MTU4 compare match 4C
18h		MTU4 compare match 4D
19h		MTU4 overflow
1Ah		MTU4 underflow
1Fh	Compare match timer	CMT1 compare match 1
22h	8-bit timers	TMR0 compare match A0
23h		TMR0 compare match B0
24h		TMR0 overflow
28h		TMR2 compare match A2
29h		TMR2 compare match B2
2Ah		TMR2 overflow
2Eh	Realtime clock	RTC periodic event (select 1/256, 1/128, 1/64, 1/32, 1/16, 1/8, 1/4, 1/2, 1, or 2 seconds)
31h	Independent watchdog timer	IWDT underflow or refresh error
32h	Low power timer	LPT compare match 0
34h	12-bit A/D converter	S12AD comparison conditions are met
35h		S12AD comparison conditions are not met
3Ah	Serial communications interfaces	SCI5 error (receive error or error signal detection)
3Bh		SCI5 receive data full
3Ch		SCI5 transmit data empty
3Dh		SCI5 transmit end
4Eh	I^2C-bus interface	RIIC0 communication error or event generation
4Fh		RIIC0 receive data full
50h		RIIC0 transmit data empty
51h		RIIC0 transmit end
52h	Serial peripheral interface	RSPI0 error (mode fault, overrun, or parity error)
53h		RSPI0 idle
54h		RSPI0 receive buffer full
55h		RSPI0 transmit buffer empty
56h		RSPI0 transmit end

Downloaded from Arrow.com.

Table 20.3 Correspondence between Values Set in ELSRn.ELS[7:0] Bits and Event Signals (2/2)

ELS[7:0] Bit Value	Peripheral Modules	Event Signal Set in ELSRn
58h	12-bit A/D converter	S12AD A/D conversion end
5Bh	Voltage detection circuit	LVD1 voltage detection
5Dh	DMA controller	DMAC0 transfer end
5Eh		DMAC1 transfer end
5Fh		DMAC2 transfer end
60h		DMAC3 transfer end
61h	Data transfer controller	DTC transfer end
62h	Clock generation circuit	Oscillation stop detection of clock generation circuit
63h	I/O ports	Input edge detection of input port group 1
64h		Input edge detection of input port group 2
65h		Input edge detection of single input port 0
66h		Input edge detection of single input port 1
67h		Input edge detection of single input port 2
68h		Input edge detection of single input port 3
69h	Event link controller	Software event
6Ah	Data operation circuit	DOC data operation condition met
Settings other than above are prohibited.		

Downloaded from Arrow.com.

20.2.3 Event Link Option Setting Register A (ELOPA)

Address(es): ELC.ELOPA 0008 B11Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	MTU3MD[1:0]		MTU2MD[1:0]		MTU1MD[1:0]		—	—
Value after reset:	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b1, b0	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b3, b2	MTU1MD[1:0]	MTU1 Operation Select	b3 b2 0 0: Counting is started. 0 1: Counting is restarted. 1 0: Input capture*1 1 1: Event output is disabled.	R/W
b5, b4	MTU2MD[1:0]	MTU2 Operation Select	b5 b4 0 0: Counting is started. 0 1: Counting is restarted. 1 0: Input capture*2 1 1: Event output is disabled.	R/W
b7, b6	MTU3MD[1:0]	MTU3 Operation Select	b7 b6 0 0: Counting is started. 0 1: Counting is restarted. 1 0: Input capture*3 1 1: Event output is disabled.	R/W

Note 1. The MTU1.TCNT value is captured into the MTU1.TGRA register.
Note 2. The MTU2.TCNT value is captured into the MTU2.TGRA register.
Note 3. The MTU3.TCNT value is captured into the MTU3.TGRA register.

The ELOPA register specifies the operations of MTU1 to MTU3 when an event signal is input. Set 11b (event output is disabled) when the ELC function is not used.

20.2.4 Event Link Option Setting Register B (ELOPB)

Address(es): ELC.ELOPB 0008 B120h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	MTU4MD[1:0]	
Value after reset:	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b1, b0	MTU4MD[1:0]	MTU4 Operation Select	b1 b0 0 0: Counting is started. 0 1: Counting is restarted. 1 0: Input capture*1 1 1: Event output is disabled.	R/W
b7 to b2	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

Note 1. The MTU4.TCNT value is captured into the MTU4.TGRA register.

The ELOPB register specifies the operation of MTU4 when an event signal is input. Set 11b (event output is disabled) when the ELC function is not used.

Downloaded from Arrow.com.

20.2.5 Event Link Option Setting Register C (ELOPC)

Address(es): ELC.ELOPC 0008 B121h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	LPTMD[1:0]		CMT1MD[1:0]		—	—
Value after reset:	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b1, b0	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b3, b2	CMT1MD[1:0]	CMT1 Operation Select	b3 b2 0 0: Counting is started. 0 1: Counting is restarted. 1 0: Event counter 1 1: Event output is disabled.	R/W
b5, b4	LPTMD[1:0]	LPT Operation Select	b5 b4 0 0: Output the LPT compare match 0 event to ICU as an interrupt request 1 1: Event output is disabled. Settings other than above are prohibited.	R/W
b7, b6	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

The ELOPC register specifies the operations of CMT1 and LPT when an event signal is input. Set 11b (event output is disabled) when the ELC function is not used.

20.2.6 Event Link Option Setting Register D (ELOPD)

Address(es): ELC.ELOPD 0008 B122h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	TMR2MD[1:0]		—	—	TMR0MD[1:0]	
Value after reset:	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b1, b0	TMR0MD[1:0]	TMR0 Operation Select	b1 b0 0 0: Counting is started. 0 1: Counting is restarted. 1 0: Event counter 1 1: Event output is disabled.	R/W
b3, b2	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b5, b4	TMR2MD[1:0]	TMR2 Operation Select	b5 b4 0 0: Counting is started. 0 1: Counting is restarted. 1 0: Event counter 1 1: Event output is disabled.	R/W
b7, b6	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

The ELOPD register specifies the operations of TMR0 and TMR2 when an event signal is input. Set 11b (event output is disabled) when the ELC function is not used.

Downloaded from Arrow.com.

20.2.7 Port Group Setting Register n (PGRn) (n = 1, 2)

Address(es): ELC.PGR1 0008 B123h, ELC.PGR2 0008 B124h

	b7	b6	b5	b4	b3	b2	b1	b0
	PGR7	PGR6	PGR5	PGR4	PGR3	PGR2	PGR1	PGR0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PGR0	Port Group Setting 0	0: Does not specify the port as a member of the port group. 1: Specifies the port as a member of the port group.	R/W
b1	PGR1	Port Group Setting 1		R/W
b2	PGR2	Port Group Setting 2		R/W
b3	PGR3	Port Group Setting 3		R/W
b4	PGR4	Port Group Setting 4		R/W
b5	PGR5	Port Group Setting 5		R/W
b6	PGR6	Port Group Setting 6		R/W
b7	PGR7	Port Group Setting 7		R/W

The PGRn register specifies a group of I/O ports. Among the ports, ports corresponding to bits set to 1 in the register are selected for a port group.

For example, when the PGR6 and PGR3 bits in the PGR1 register are set to 1, the PB6 and PB3 pins are selected to a port group.

Table 20.4 shows the PGRn register and corresponding ports.

Table 20.4 Registers Related to Port Groups and Corresponding Port Numbers

Port Number	Port Group Setting Register (PGRn)	Port Group Control Register (PGCn)	Port Buffer Register (PDBFn)
Port B	PGR1 register	PGC1 register	PDBF1 register
Port E	PGR2 register	PGC2 register	PDBF2 register

Downloaded from Arrow.com.

20.2.8 Port Group Control Register n (PGCn) (n = 1, 2)

Address(es): ELC.PGC1 0008 B125h, ELC.PGC2 0008 B126h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	PGCO[2:0]			—	PGCOVE	PGCI[1:0]	
Value after reset:	1	0	0	0	1	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	PGCI[1:0]	Event Output Edge Select	b1 b0 0 0: Event signal is output upon detection of the rising edge of the input signal to the port. 0 1: Event signal is output upon detection of the falling edge of the input signal to the port. 1 x: Event signal is output upon detection of both the rising and falling edges of the input signal to the port.	R/W
b2	PGCOVE	PDBF Overwrite	0: Overwriting the PDBFn register is disabled. 1: Overwriting the PDBFn register is enabled.	R/W
b3	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b6 to b4	PGCO[2:0]	Port Group Operation Select	b6 b4 0 0 0: Low is output when an event signal is input. 0 0 1: High is output when an event signal is input. 0 1 0: The output is toggled (inverted) when an event signal is input. 0 1 1: The buffer value is output when an event signal is input. 1 x x: The output data is rotated (from MSB to LSB) in the port group when an event signal is input.	R/W
b7	—	Reserved	This bit is read as 1. The write value should be 1.	R/W

x: Don’t care

For the port group set as an output, the PGCn register specifies the form of outputting the signal from the port when an event signal is input. For the port group set as an input, the PGCn register enables/disables overwriting of the PDBFn register and specifies the conditions of event generation (edge of the input signal).
Specify the I/O direction of the port by the corresponding bit in the PDR register.
Refer to Table 20.4 for the PGCn register and corresponding ports.

Downloaded from Arrow.com.

20.2.9 Port Buffer Register n (PDBFn) (n = 1, 2)

Address(es): ELC.PDBF1 0008 B127h, ELC.PDBF2 0008 B128h

	b7	b6	b5	b4	b3	b2	b1	b0
	PDBF7	PDBF6	PDBF5	PDBF4	PDBF3	PDBF2	PDBF1	PDBF0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PDBF0	Port Buffer 0	Specify the data to be transferred to the PODR register when an event signal is input. The setting value is valid when the PGCn.PGCO[2:0] bits are 011b or 1xxb. Write access to the bit specified as a member of the input port group is disabled. For details, refer to section 20.3, Operation.	R/W
b1	PDBF1	Port Buffer 1		R/W
b2	PDBF2	Port Buffer 2		R/W
b3	PDBF3	Port Buffer 3		R/W
b4	PDBF4	Port Buffer 4		R/W
b5	PDBF5	Port Buffer 5		R/W
b6	PDBF6	Port Buffer 6		R/W
b7	PDBF7	Port Buffer 7		R/W

The PDBFn register is an 8-bit readable/writable register used in combination with the PGRn register. Refer to section 20.3.6, I/O Port Operation When Event Signal is Input and Event Generation for the PDBFn register operations. Refer to Table 20.4 for the PDBFn register and corresponding ports.

Downloaded from Arrow.com.

20.2.10 Event Link Port Setting Register m (PELm) (m = 0 to 3)

Address(es): ELC.PEL0 0008 B129h, ELC.PEL1 0008 B12Ah, ELC.PEL2 0008 B12Bh, ELC.PEL3 0008 B12Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	PSM[1:0]		PSP[1:0]		PSB[2:0]		
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	PSB[2:0]	Bit Number Specification	Set a bit number for a port to be specified as a single port.	R/W
b4, b3	PSP[1:0]	Port Number Specification	b4 b3 0 0: Setting disabled 0 1: Port B (corresponding to PGR1) 1 0: Port E (corresponding to PGR2) 1 1: Setting prohibited	R/W
b6, b5	PSM[1:0]	Event Link Specification	• For the output port, specify the data to be output from the port. b6 b5 0 0: Low is output when an event signal is input. 0 1: High is output when an event signal is input. 1 x: The output is toggled (inverted) when an event signal is input. • For the input port, select the edge on which the event signal is to be output. b6 b5 0 0: Event signal is output upon detection of the rising edge. 0 1: Event signal is output upon detection of the falling edge. 1 x: Event signal is output upon detection of both the rising and falling edges.	R/W
b7	—	Reserved	This bit is read as 1. The write value should be 1.	R/W

x: Don't care

The PELm register specifies the single port, the operation upon an event signal input, and the conditions of event generation. This MCU can specify a total of four bits in port B and port E to respective single ports.
Specify the I/O direction of the port by the corresponding bit in the PDR register.

Downloaded from Arrow.com.

20.2.11 Event Link Software Event Generation Register (ELSEGR)

Address(es): ELC.ELSEGR 0008 B12Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	WI	WE	—	—	—	—	—	SEG
Value after reset:	1	0	1	1	1	1	1	0

Bit	Symbol	Bit Name	Description	R/W
b0	SEG	Software Event Generation	0: Normal operation 1: Software event is generated.	W
b5 to b1	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b6	WE	SEG Bit Write Enable	0: Write to SEG bit is disabled. 1: Write to SEG bit is enabled.	R/W
b7	WI	ELSEGR Register Write Disable	0: Write to ELSEGR register is enabled. 1: Write to ELSEGR register is disabled.	W

The MOV instruction must be used to write to this register.

SEG Bit (Software Event Generation)

When 1 is written to this bit while the WE bit is 1, a software event is generated.
This bit is read as 0. Even if 1 is written to this bit, this bit does not become 1.

WE Bit (SEG Bit Write Enable)

The SEG bit can be written to only when the WE bit is 1.
To set this bit to 1, write 0 to the WI bit and write 1 to this bit simultaneously.
To set this bit to 0, write 0 to the WI bit and write 0 to this bit simultaneously.

WI Bit (ELSEGR Register Write Disable)

The ELSEGR register can be written to only when the value to be written to the WI bit is 0.
This bit is read as 1.

Downloaded from Arrow.com.

20.3 Operation

20.3.1 Relation between Interrupt Handling and Event Linking

The peripheral modules incorporated in the MCU are provided with the interrupt request status flags and the interrupt enable bits to enable/disable these interrupt requests. When an interrupt request is generated in a peripheral module, the corresponding interrupt request status flag becomes 1. If the corresponding interrupt request is enabled then, the interrupt is requested to the CPU.

In contrast, the event link controller (ELC) uses the interrupt requests generated by various peripheral modules as event signals, interconnects (links) peripheral modules, and then, makes peripheral modules perform direct interlinked operation among them without using software. Event signals can be output regardless of the setting of the corresponding interrupt enable bit.

Figure 20.2 shows the relation between the interrupt handling and ELC.

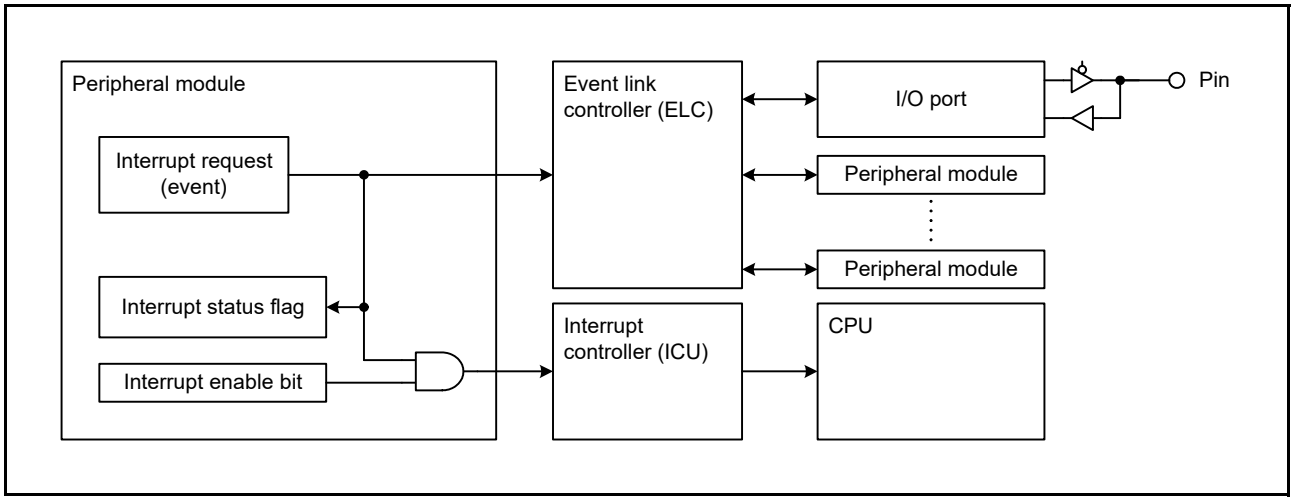


Figure 20.2 Relation between Interrupt Handling and ELC

Downloaded from Arrow.com.

20.3.2 Event Linkage

When events are specified in the ELSRn registers, the corresponding peripheral modules can be operated at generation of the specified events. A single peripheral module can link only with a single event. Set the ELSRn register after completing the initialization of the peripheral module to operate by an event. Table 20.5 lists the operations of peripheral modules when an event signal is input.

Table 20.5 Operations of Peripheral Modules When Event Signal is Input

Peripheral Module	Operations When Event Signal is Input		
MTU CMT TMR	The following operations can be selected by setting the ELOPA to ELOPD registers: • Starts counting when an event signal is input. • Restarts counting when an event signal is input. • Counts the input events (CMT, TMR). • Performs input-capture operation when an event signal is input (MTU).		
POE	The MTU complementary PWM output pins and MTU0 output pins become high-impedance when an event signal is input.		
CTSU	Starts measurement of electrostatic capacitance when an event signal is input.		
A/D converter	Starts A/D conversion when an event signal is input.		
D/A converter	Starts D/A conversion when an event signal is input.		
I/O ports (output)	The value of PODR register (port output data register) changes when an event signal is input (The level output from the corresponding pin changes).	Port group	• Changes the PODR register value to the specified value. • Transfers the PDBFn register value to the PODR register (n = 1, 2). • Rotates the PODR register.
		Single port	Changes the PODR register value to the specified value.
I/O ports (input)	When the signal level of the input pin changes	Port group	Generates an event.
		Single port	
	When an event signal is input	Port group	Transfers the signal level of the input pin to the PDBFn register.
		Single port	This combination cannot be used.
Clock generation circuit	Switches the clock source to the low-speed on-chip oscillator when an event signal is input.*1		
Interrupt controller	Request an interrupt to the CPU, starts DMA transfer, or starts DTC transfer when an event signal is input.		

Note 1. The SCKCR3.CKSEL[2:0] bits are modified to 000b (LOCO) regardless of the value of the protect register (PRCR.PRC0).

Downloaded from Arrow.com.

20.3.3 Operation of Peripheral Timer Modules When Event Signal is Input

For the timer modules, set the ELOPA to ELOPD register to specify the operation for when an event signal is input.

(1) Count Start Operation

When an event signal is input, the timer starts counting and the count start bit*1 in each timer control register becomes 1. An event signal that is input while the count start bit is 1 is ignored.

(2) Count Restart Operation

When an event signal is input, the timer counter is cleared. Since the count start bit*1 in each timer control register is retained, counting is restarted when an event signal is input while the count start bit is 1.

(3) Event Counter Operation

Event signal is selected as the timer count source. When an event signal is input, the timer counter is incremented.

(4) Input Capture Operation

When an event signal is input, the timer performs input-capture operation.

Note 1. Refer to the register descriptions on starting the timer in the relevant peripheral timer module section.

20.3.4 Operation of CTSU When Event Signal is Input

When an event signal is input while the CTSUCAP and CTSUSTRT bits in the CTSUCR0 register are set to 1, a measurement is started. Refer to the description of the CTSUCR0.CTSUSTRT bit for details.

20.3.5 Operation of A/D and D/A Converters When Event Signal is Input

When an event signal is input, the ADCSR.ADST bit and the DACR.DAOE0 bit*1 are set to 1 and the A/D and D/A converter start A/D and D/A conversion, respectively.

Note 1. Refer to the bit descriptions in the A/D converter and D/A converter sections.

20.3.6 I/O Port Operation When Event Signal is Input and Event Generation

The I/O port operation at an event signal input and conditions for event generation are set by the registers in ELC. The I/O ports that are used to set an event linkage are port B and port E.

(1) Single Ports and Port Groups

There are two event link modes: event link to single ports and event link to port groups. In the former mode, events can be interconnected to any one of the I/O ports. In the latter mode, events can be interconnected to port groups consisting of any two or more bits in the same I/O ports.

A single port can be set by the PELm.PSP[1:0] and PSB[2:0] bits (m = 0 to 3). A port group can be specified by setting two or more bits in the PGRn register (n = 1, 2) to 1. Among the ports corresponding to the bits set to 1 in the PGRn register, a port set as output becomes an output port group member, and a port set as input becomes an input port group member.

If an I/O port is specified as both a single port and a member of a port group, both functions are enabled when the corresponding port is input, whereas only the port group function is enabled when the corresponding port is output.

Set the PDR register to select the direction of the I/O ports.

Downloaded from Arrow.com.

(2) Event Generation in Single Input Ports

A single port that is set as input generates an event signal when the input signal to the corresponding pin changes. The event generation condition is specified using the PELm.PSM[1:0] bits (m = 0 to 3). An example of operation is shown in Figure 20.3 (1).

(3) Single Output Ports Operation When Event Signal is Input

When an event signal is input to a single port set as output, the output level (the PODR register value) of the corresponding pin changes as specified by the PELm.PSM[1:0] bits. An example of operation is shown in Figure 20.3 (2).

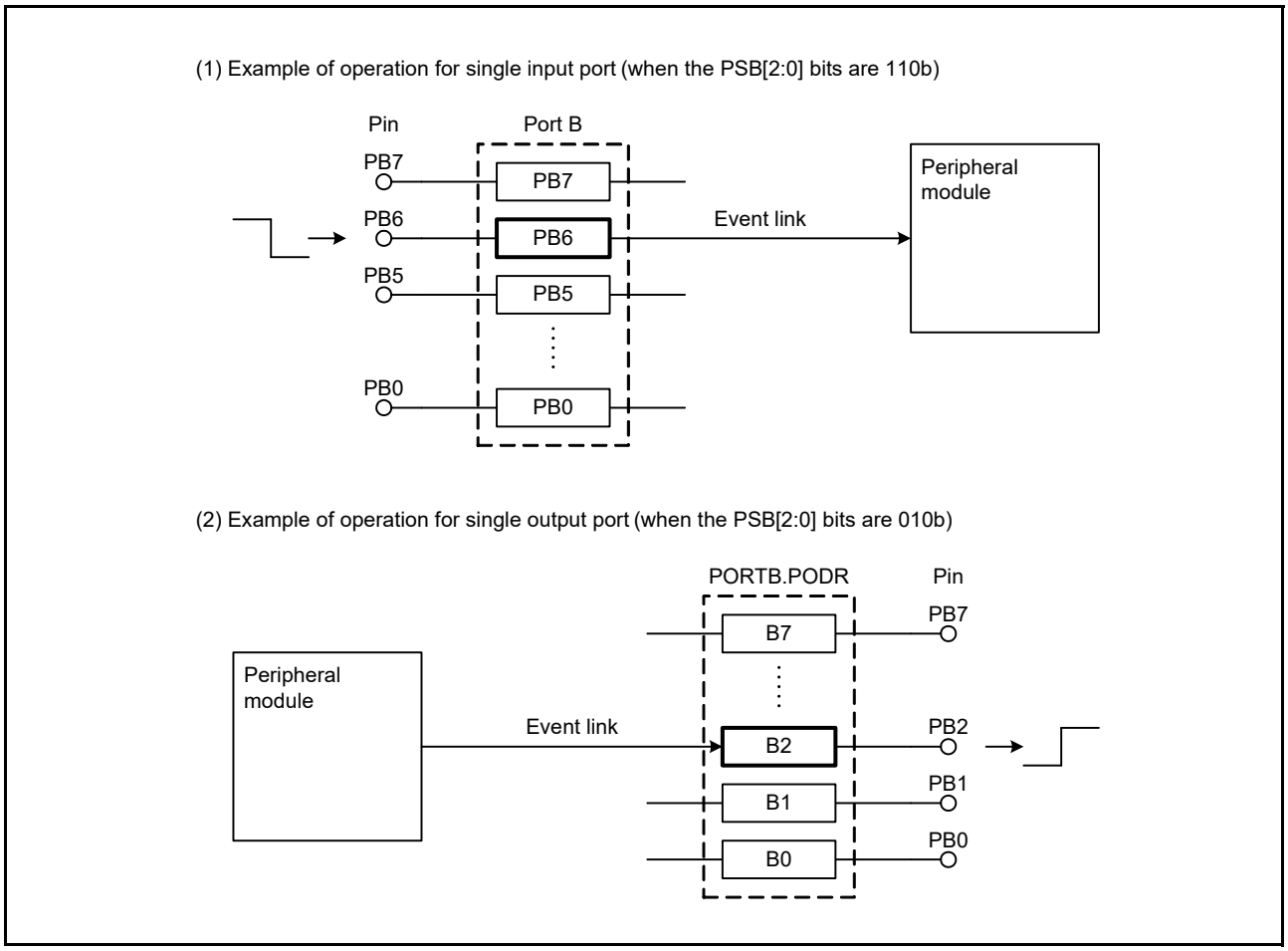


Figure 20.3 Event Linkage Related to Single Ports (Port B)

(4) Event Generation in Input Port Group

An input port group generates an event signal when any of input signals to the corresponding pins change. The event generation condition is specified using the PGCn.PGCI[1:0] bits (n = 1, 2).

Downloaded from Arrow.com.

(5) Input Port Group Operation When Event Signal is Input

When an event signal is input to an input port group, the level of the corresponding pins is transferred to the PDBFn register. Values of the bits corresponding to ports that are not specified as members of the input port group do not change. An example of operation is shown in Figure 20.4.

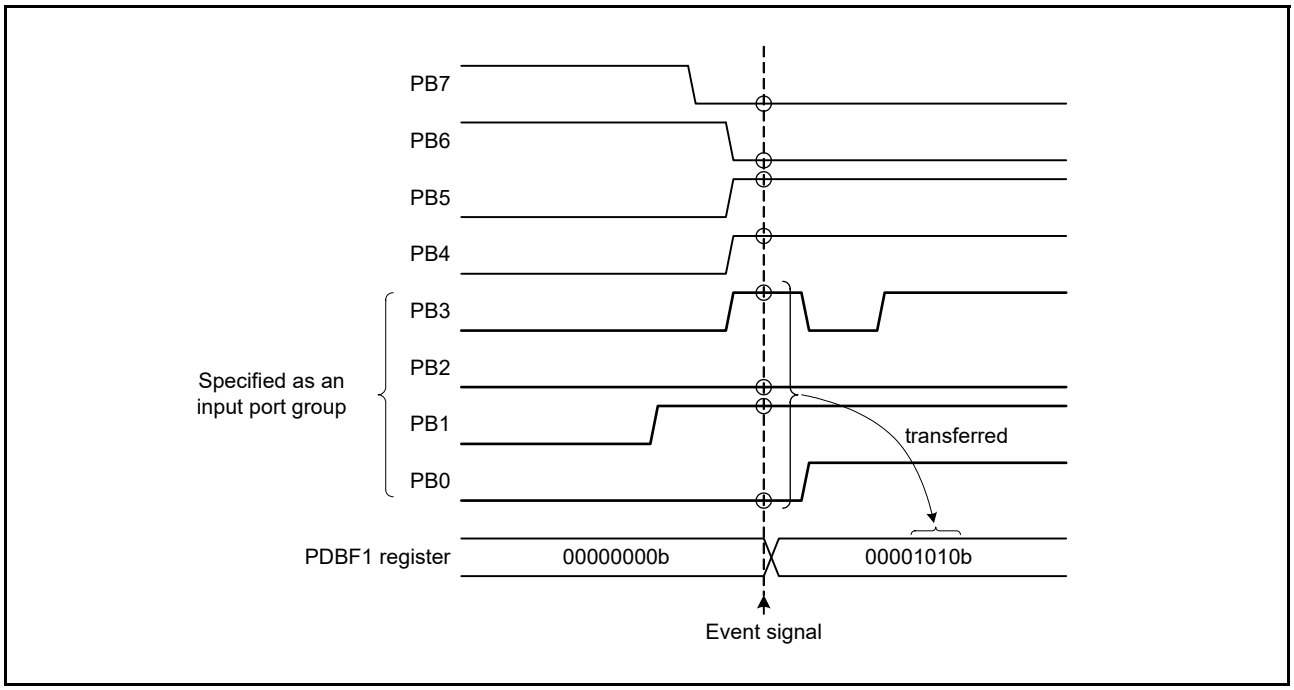


Figure 20.4 Event Linkage Related to Input Port Groups (Port B)

(6) Output Port Group Operation When Event Signal is Input

When an event signal is input to an output port group, the value of the corresponding PODR register changes according to a setting of the PGCn.PGCO[2:0] bits (n = 1, 2). An example of operation is shown in Figure 20.5.

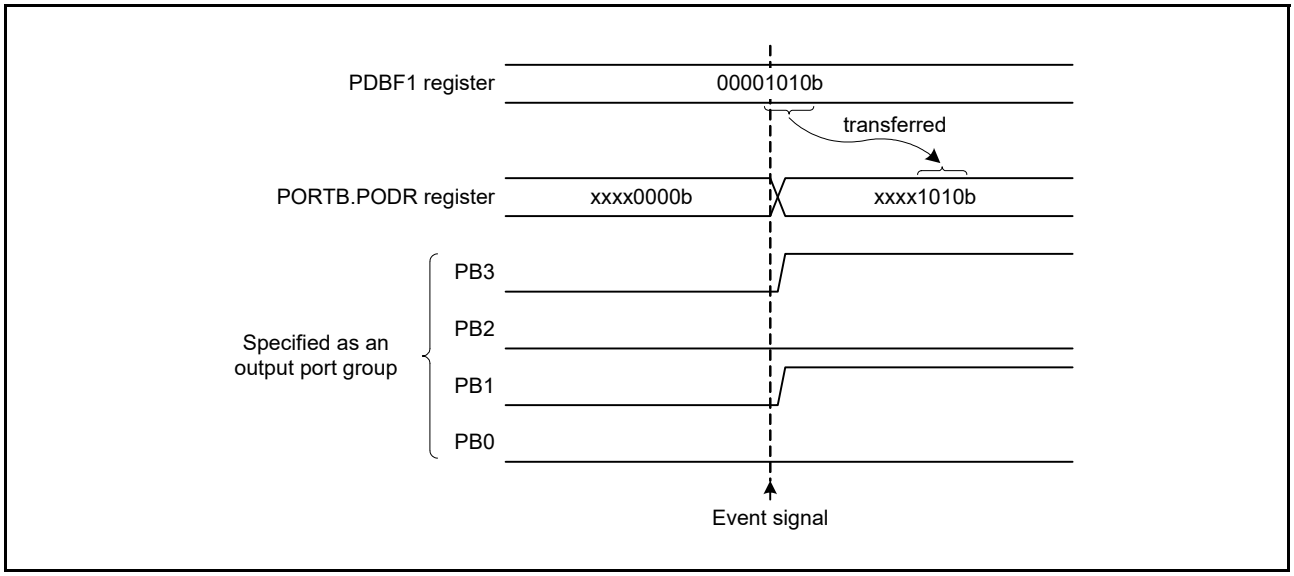


Figure 20.5 Event Linkage Related to Output Port Groups (Port B)

Downloaded from Arrow.com.

(7) Operation of the PDBFn Registers

(a) Input Port Groups

When an event signal is input to an input port group, the level of the corresponding pins is transferred to the PDBFn register (n = 1, 2). When another event signal is input to the input port group in this condition, different operations are performed depending on the PGCn.PGCOVE bit setting described as below.

- When the PGCn.PGCOVE bit is 0 (overwriting is disabled)
  When the value transferred to the PDBFn register after an input of the last event signal has already been read by the CPU or DTC, the level of the corresponding pins at the time is transferred to the PDBFn register. When the value has not been read, the level of the pins is not transferred to the PDBFn register, and the input event signal is ignored.
- When the PGCn.PGCOVE bit is 1 (overwriting is enabled)
  When another event signal is input to the input port group, the level of the corresponding pins is transferred to the PDBFn register.

(b) Output Port Groups

When an output port group is specified to output the PDBFn register value (PGCn.PGCO[2:0] bits = 011b), the PDBFn register value is transferred to the PODR register following an input of an event signal to the output port group. Data is not transferred to the bits corresponding to the ports that are not specified as members of the output port group.
When output data is specified to rotate in an output port group (PGCn.PGCO[2:0] bits = 1xxb), the data is transferred from the PDBFn register to the PODR register at first event signal, and the PODR register value is rotated from MSB to LSB within the relevant group at second and subsequent signals.
Examples of operation are shown in Figure 20.6.

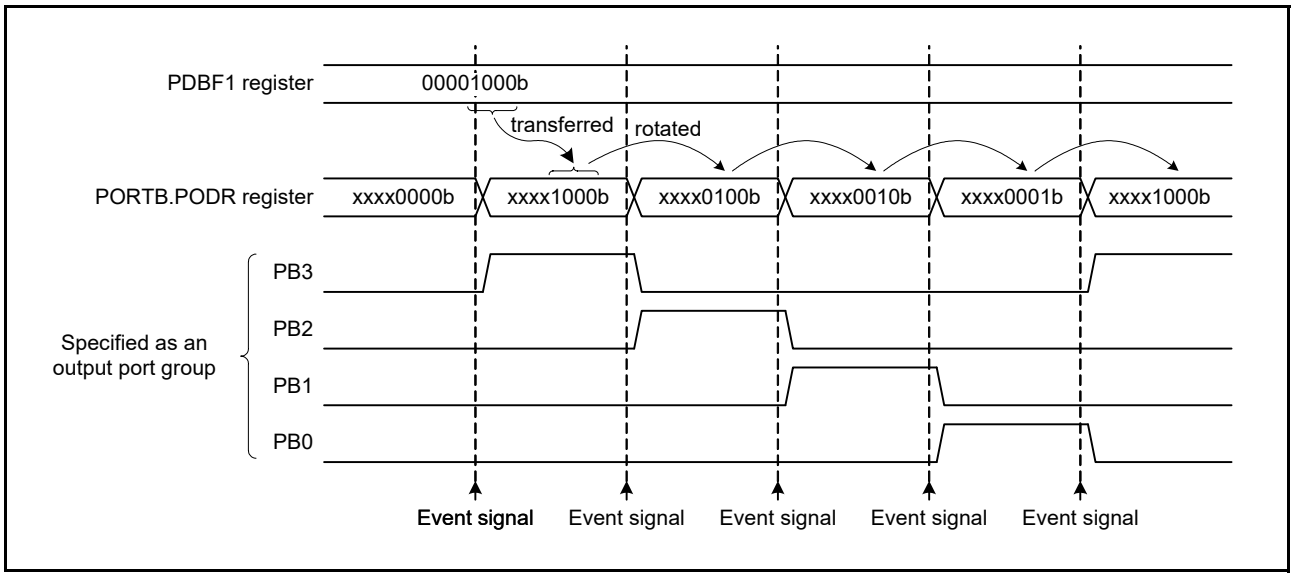


Figure 20.6 Bit-Rotating Operation of Output Port Groups (Port B)

Downloaded from Arrow.com.

(8) Restrictions on Writing to PODR and PDBF Registers

When the ELCR.ELCON bit is 1 (ELC function is enabled), write access to the PODR and PDBFn registers (n = 1, 2) becomes disabled at the following conditions.

- When a port is specified as a member of the input port group and when the event linkage is set, write access to the corresponding bit in the PDBFn register becomes disabled. However, when the DOC is selected for event signal, write access is enabled.
- When a port is specified as a member of the output port group, write access to the corresponding bit in the PODR register becomes disabled.
- When a port is specified as a single output port and when the event linkage for the port is set by the ELSRn register, write access to the corresponding bit in the PODR register becomes disabled.

20.3.7 Example of Procedure for Linking Events

The following describes the procedure for linking events.

(1) Initialize the peripheral module (destination) that operates based on an event signal.
(2) When event linkage is set to a port, set the following registers corresponding to the port.
PODR register: Set the initial values of the output ports.
PDR register: Set the I/O direction of the ports.
PGRn register: To operate ports for a port group, select ports to be specified as port group members (n = 1, 2).
PGCn register: Set the operation of the port group.
PELm register: When a port is operated as a single port, specify the port to be used, an operation of the port at an input of event signal, and the event generation condition (m = 0 to 3).
(3) Set the number of the event signal to the ELSRn register corresponding to the destination peripheral module.
(4) To link an event to a timer module, set any of the ELOPA to ELOPD registers corresponding to the timer as required.
(5) Set the ELCR.ELCON bit to 1, which enables linkage of all the events.
(6) Set the operation of the peripheral module (source) from which an event signal is output, and activate the module. The preset operation of the destination peripheral module is started by the event signal that is output from the source peripheral module.
(7) To stop event linkage of independent peripheral module, set 00h to the ELSRn register corresponding to the peripheral module. To stop linkage of all the events, set the ELCR.ELCON bit to 0.

Note: If event signal output from the RTC is to be used, make the ELC settings after the RTC settings (initialization, time setting, etc.). Unintended events may be generated if RTC settings are made after the ELC settings.
Note: When using event signal output from the LVD, set the LVD and then the ELC. Set the corresponding ELSRn register to 00h and then disable the LVD.

Downloaded from Arrow.com.

20.4 Usage Notes

20.4.1 Setting ELSRn Register

(1) Setting ELSR8 Register

Set this register to 32h (LPT compare match 0).

(2) Setting ELSR18 and ELSR19 Registers

Specify an event number from among 63h to 6Ah. Do not set the value other than preceding numbers.

(3) Setting ELSR24, ELSR25, ELSR26, and ELSR27 Registers

Do not set the DOC data operation condition met signal (6Ah).

20.4.2 Setting Bit-Rotating Operation of Output Port Groups

When the values of the PDBFn register (n = 1, 2) are changed in the bit-rotating operation mode of the output port group, set the ELSRn register again. Set intervals for generating the event as at least one PCLKB cycle when using it for bit-rotating operation.

20.4.3 Linking DMA/DTC Transfer End Signal as Event

When linking the DMA/DTC transfer end signal as an event signal, do not set the same peripheral module as the DMA/DTC transfer destination and event link destination. If set, the peripheral module might be started before DMA/DTC transfer to the peripheral module is completed.

20.4.4 Clock Settings

To link events, make sure that the ELC and the related peripheral modules are in an operational condition. The peripheral modules cannot operate if they are in the module stop state or in mode which they stop (software standby mode).

20.4.5 Module Stop Function Setting

ELC operation can be disabled or enabled using module stop control register B (MSTPCRB). After reset is released, the ELC function is disabled. Register access is enabled by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

Downloaded from Arrow.com.

21. I/O Ports

21.1 Overview

The I/O ports function as a general I/O port, an I/O pin of a peripheral module, an input pin for an interrupt, or a bus control pin.

Some of the pins are also configurable as an I/O pin of a peripheral module or an input pin for an interrupt. All pins function as input pins immediately after a reset, and pin functions are switched by register settings. The setting of each pin is specified by the registers for the corresponding I/O port and on-chip peripheral modules.

Each port has the port direction register (PDR) that selects input or output direction, the port output data register (PODR) that holds data for output, the port input data register (PIDR) that indicates the pin states, the open drain control register y (ODRy, y = 0, 1) that selects the output type of each pin, the pull-up control register (PCR) that controls on/off of the input pull-up MOS, the drive capacity control register (DSCR) that selects the drive capacity, and the port mode register (PMR) that specifies the pin function of each port.

For details on the PMR register, see section 22, Multi-Function Pin Controller (MPC).

The configuration of the I/O ports differs depending on the package. Table 21.1 lists the specifications of I/O ports, and Table 21.2 list the port functions.

Table 21.1 Specifications of I/O Ports

Port	Package		Package	
	85 Pins, 83 Pins	Number of Pin	56 Pins	Number of Pin
PORT0	P03, P05, P07	3	P05	1
PORT1	P14 to P17	4	P14 to P17	4
PORT2	P21, P22, P25 to P27	5	P26, P27	2
PORT3	P30, P31, P35 to P37	5	P30, P31, P35 to P37	5
PORT4	P40 to P47	8	P41, P45 to P47	4
PORTB	PB0, PB1, PB3, PB5, PB7	5	PB0, PB1, PB7	3
PORTC	PC0, PC2 to PC7	7	PC0, PC2 to PC7	7
PORTD	PD3	1	PD3	1
PORTE	PE0 to PE4	5	PE2 to PE4	3
PORTJ	PJ3	1	Not provided	0
	Total of Pins	44	Total of Pins	30

Downloaded from Arrow.com.

Table 21.2 Port Functions

Port	Pin	Input Pull-up	Open Drain Output	Drive Capacity Switching	5-V Tolerant
PORT0	P03, P05, P07	✓	—	Fixed to normal output	—
PORT1	P16, P17	✓	✓	✓	✓
	P14, P15	✓	✓	✓	—
PORT2	P21, P22, P25 to P27	✓	✓	✓	—
PORT3	P30, P31	✓	✓	✓	—
	P35	—	—	—	—
	P36, P37	✓	✓	Fixed to normal output	—
PORT4	P40 to P47	✓	—	Fixed to normal output	—
PORTB	PB0, PB1, PB3, PB7	✓	✓	✓	—
	PB5	✓	✓	✓	✓
PORTC	PC0, PC2 to PC7	✓	✓	✓	—
PORTD	PD3	✓	—	✓	—
PORTE	PE0 to PE4	✓	✓	✓	—
PORTJ	PJ3	✓	✓	✓	—

Specifying input pull-up, open-drain output, switching of drive capacity, or 5-V tolerance is available for other signals on pins that also function as general I/O pins.

Downloaded from Arrow.com.

21.2 I/O Port Configuration

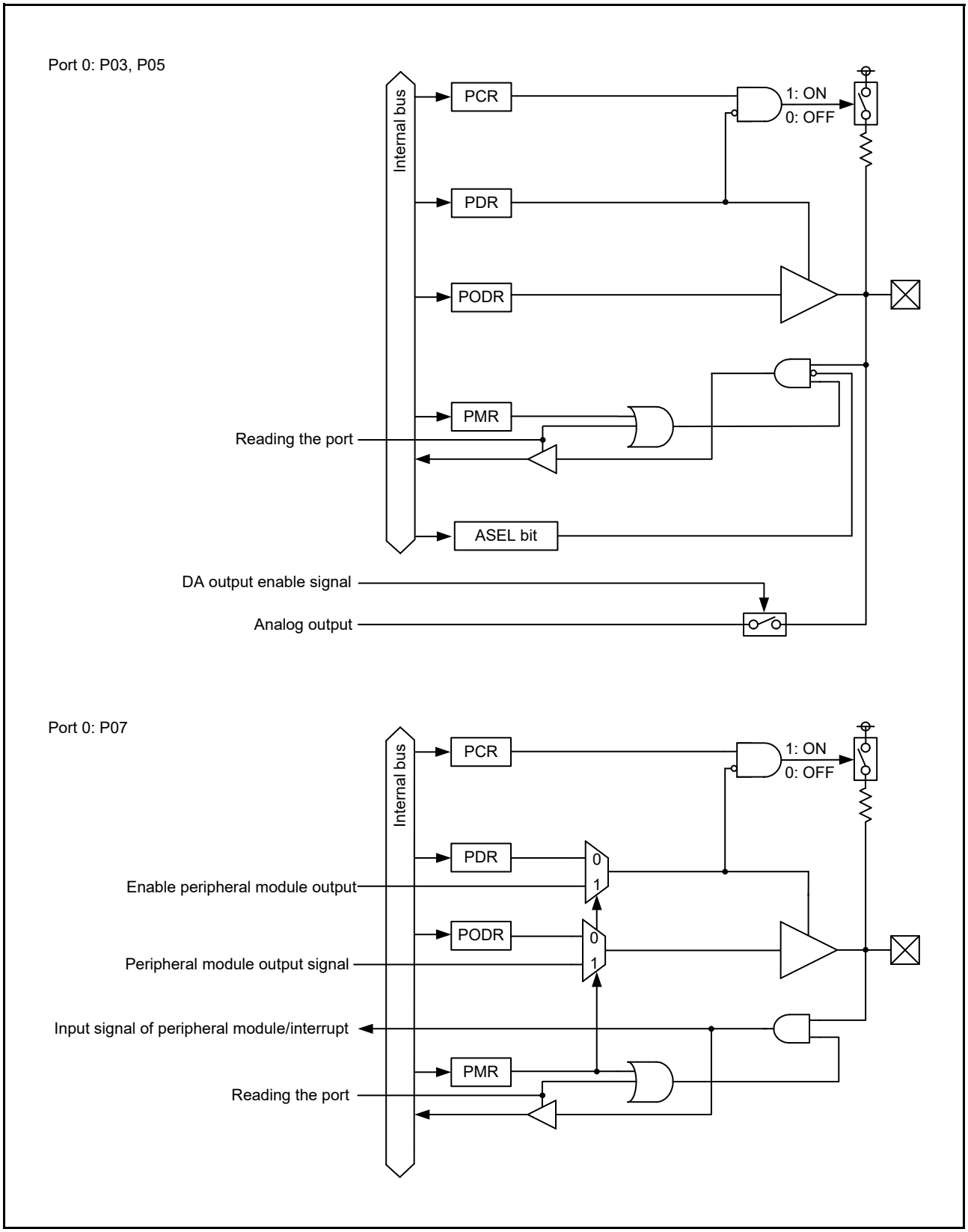


Figure 21.1 I/O Port Configuration (1)

Downloaded from Arrow.com.

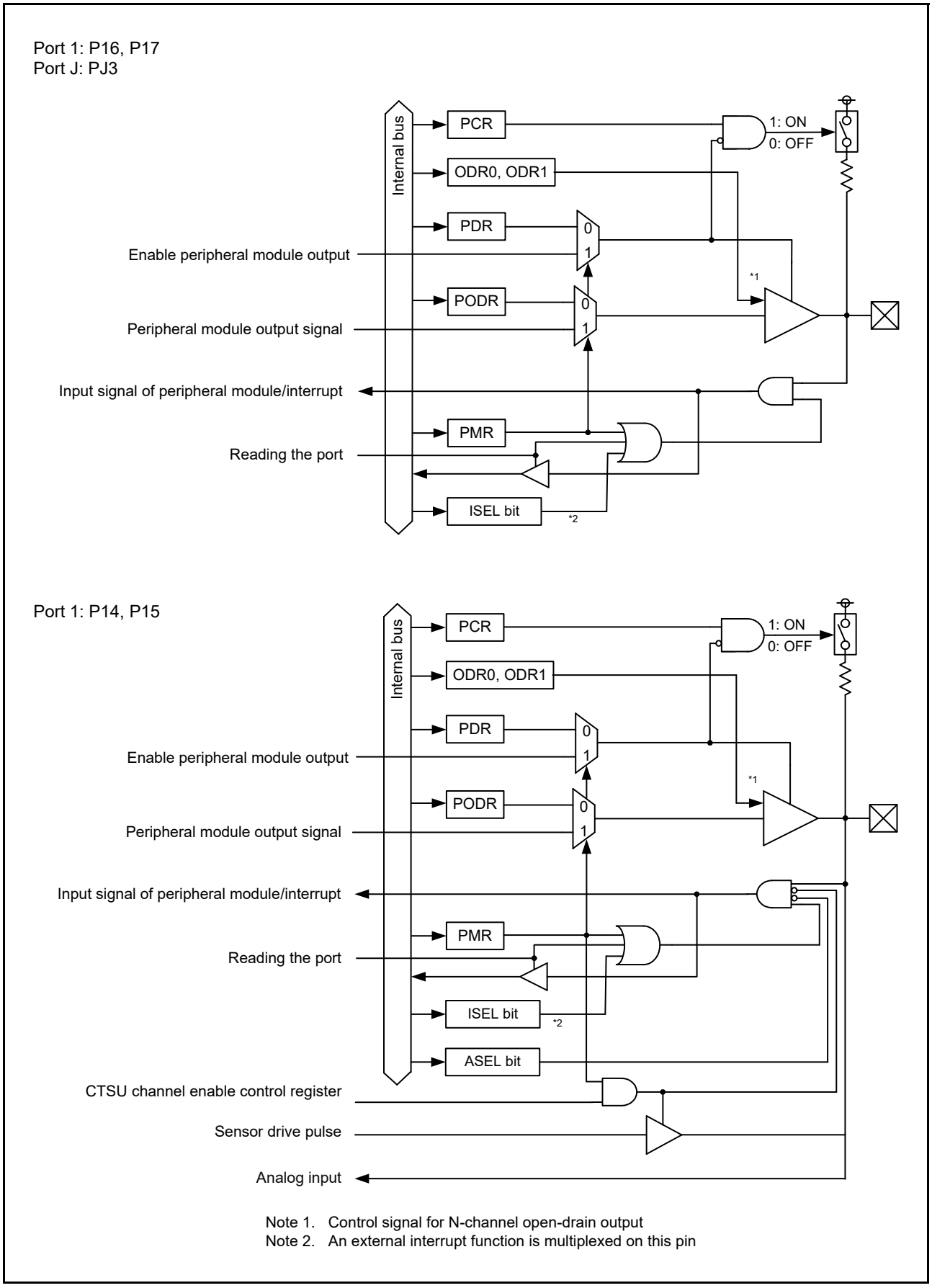
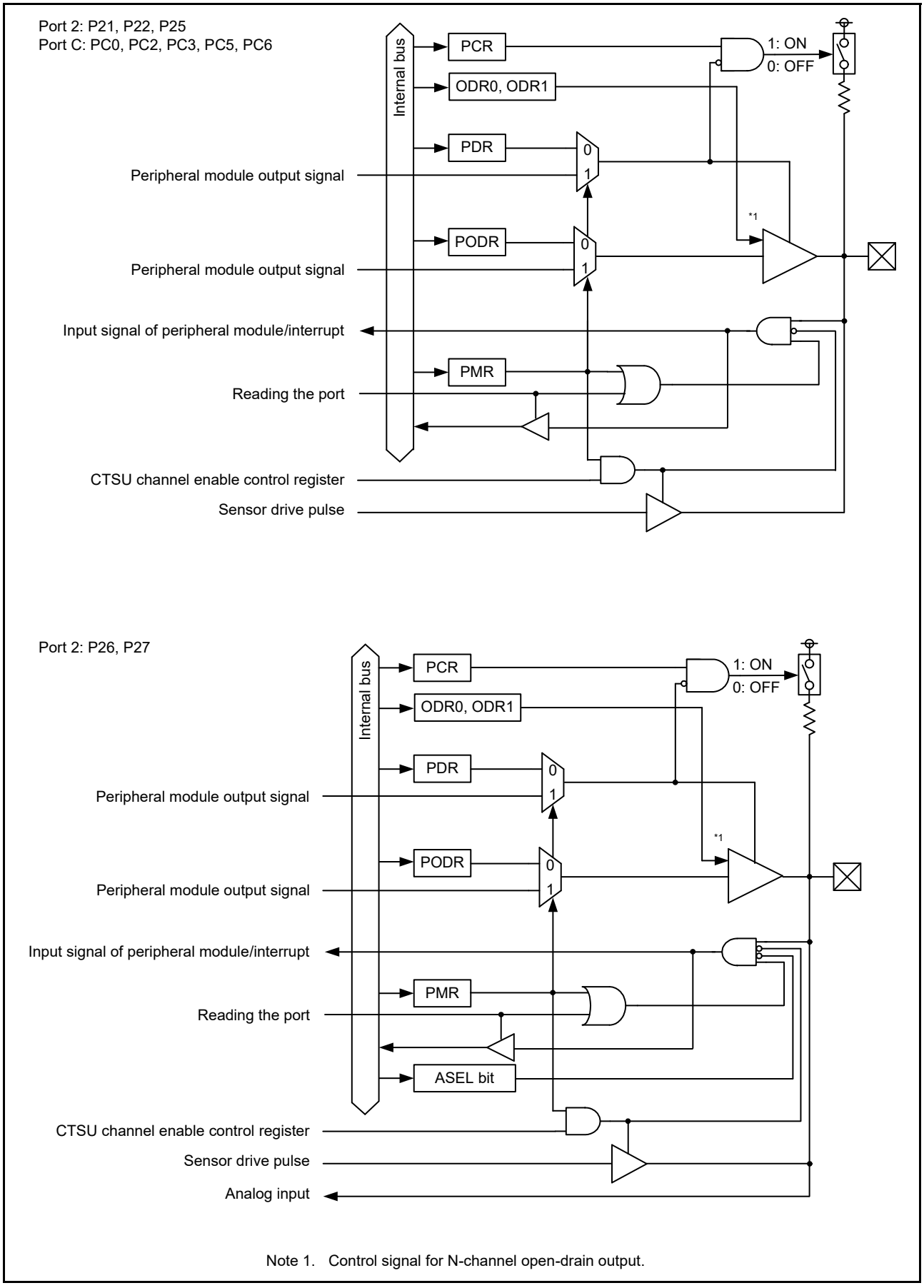


Figure 21.2 I/O Port Configuration (2)

Downloaded from Arrow.com.



Note 1. Control signal for N-channel open-drain output.

Figure 21.3 I/O Port Configuration (3)

Downloaded from Arrow.com.

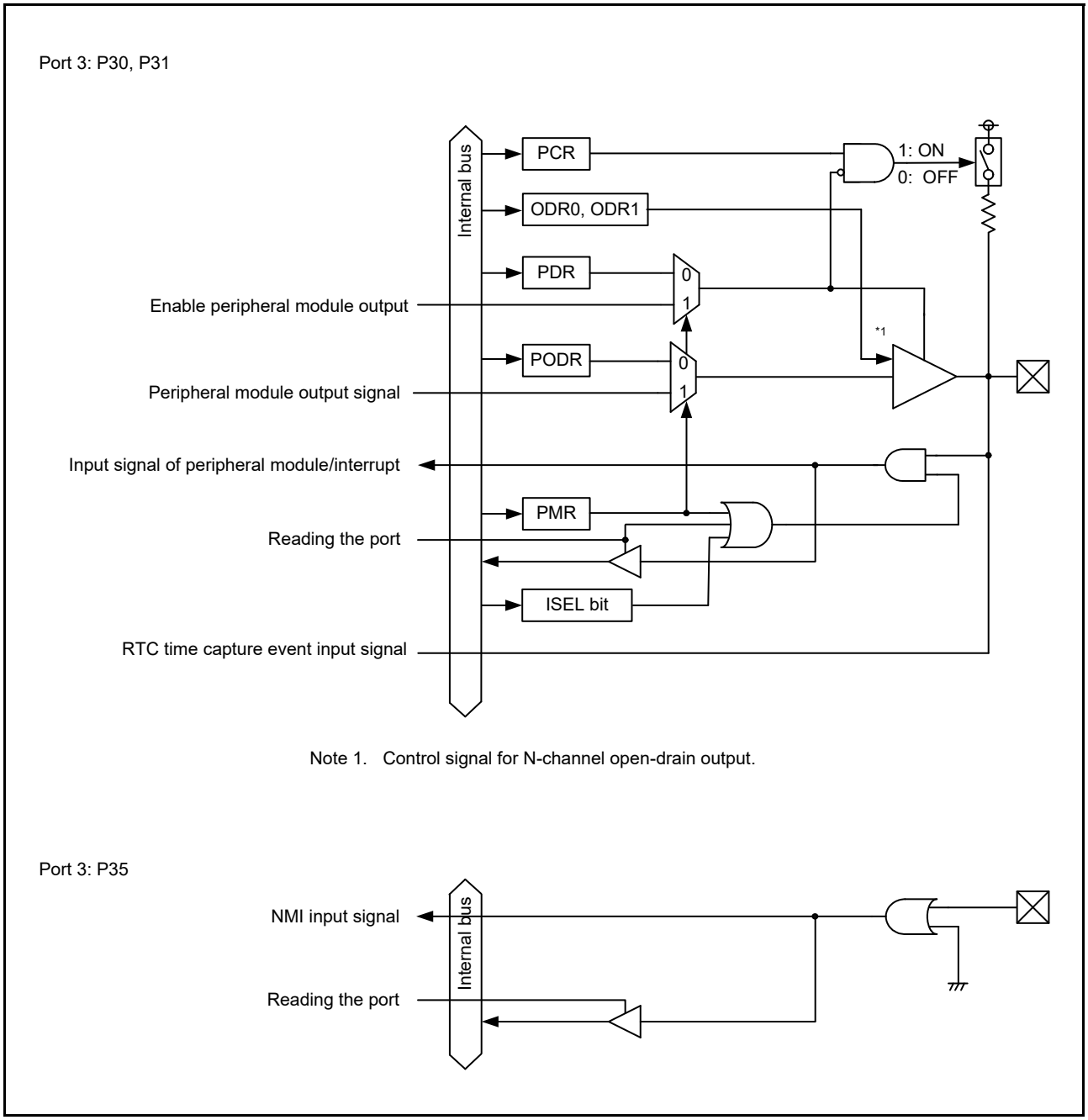
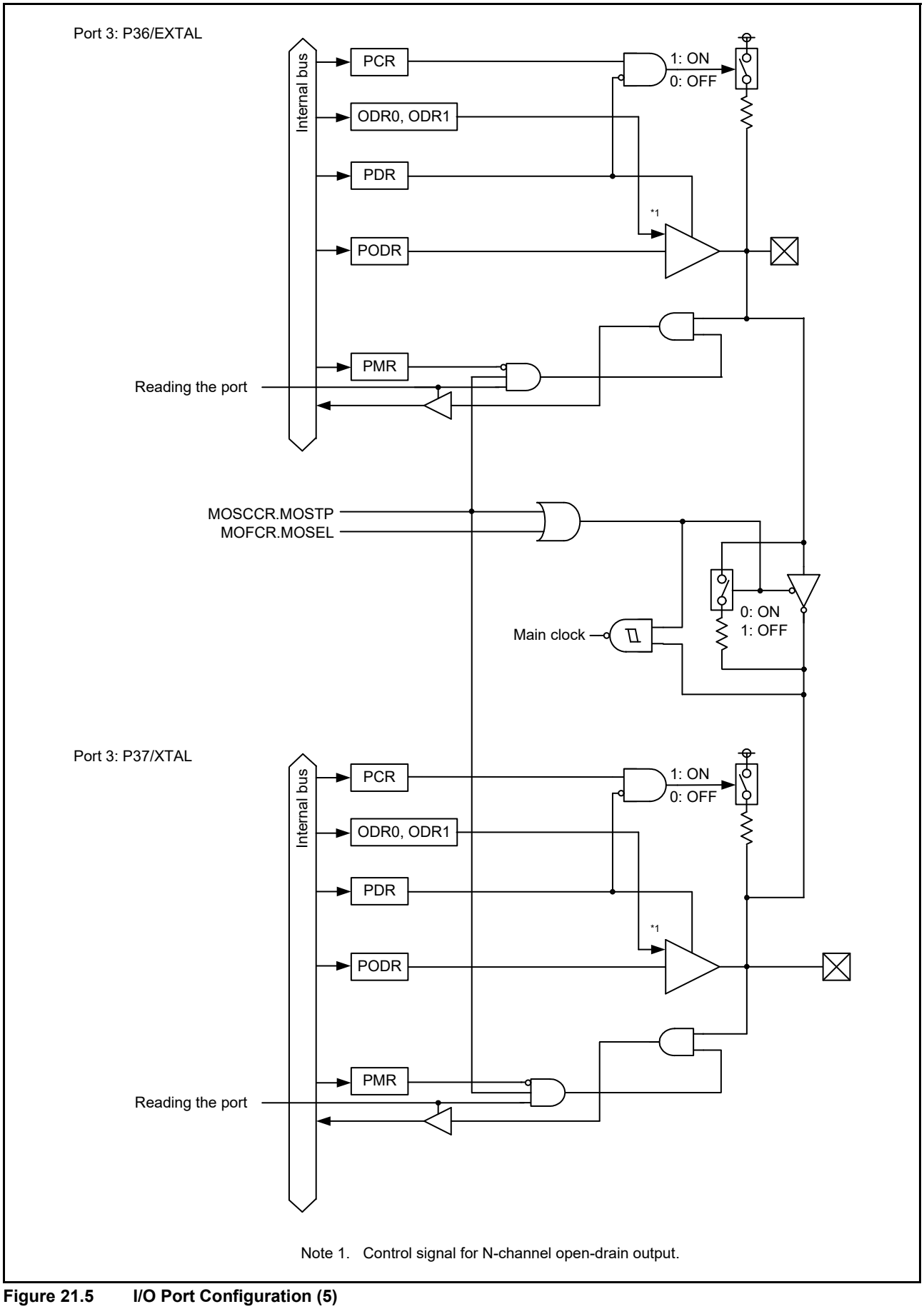


Figure 21.4 I/O Port Configuration (4)

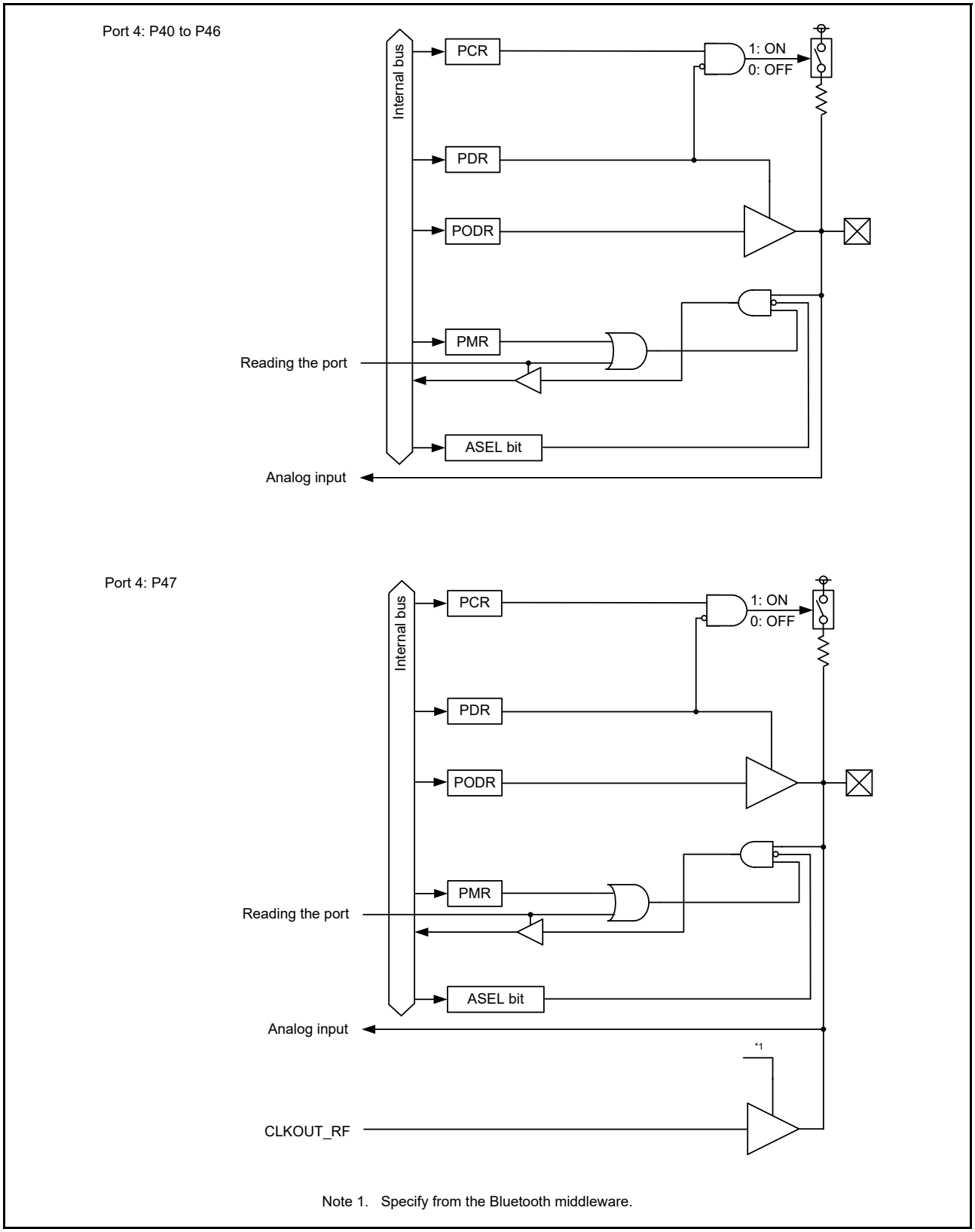
Downloaded from Arrow.com.



Note 1. Control signal for N-channel open-drain output.

Figure 21.5 I/O Port Configuration (5)

Downloaded from Arrow.com.



Note 1. Specify from the Bluetooth middleware.

Figure 21.6 I/O Port Configuration (6)

Downloaded from Arrow.com.

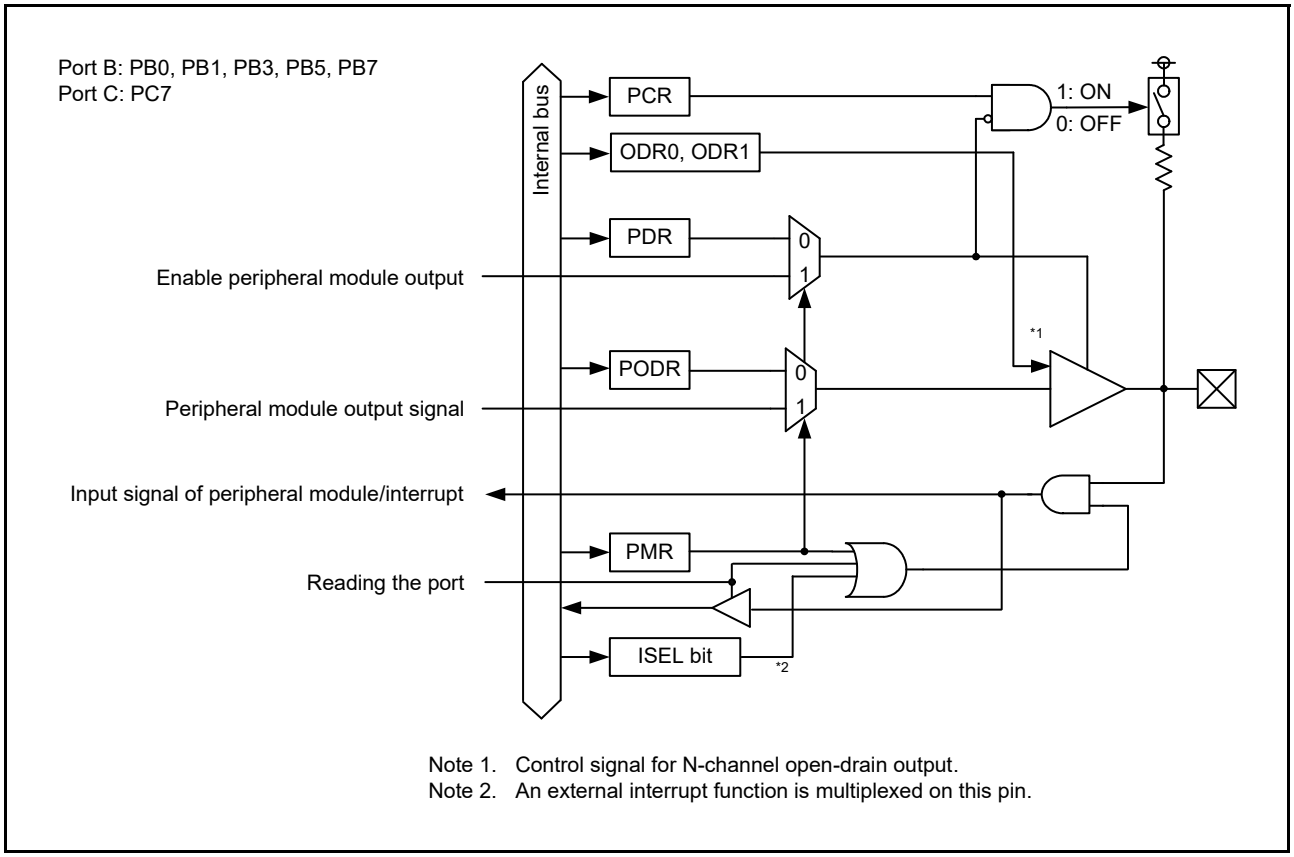
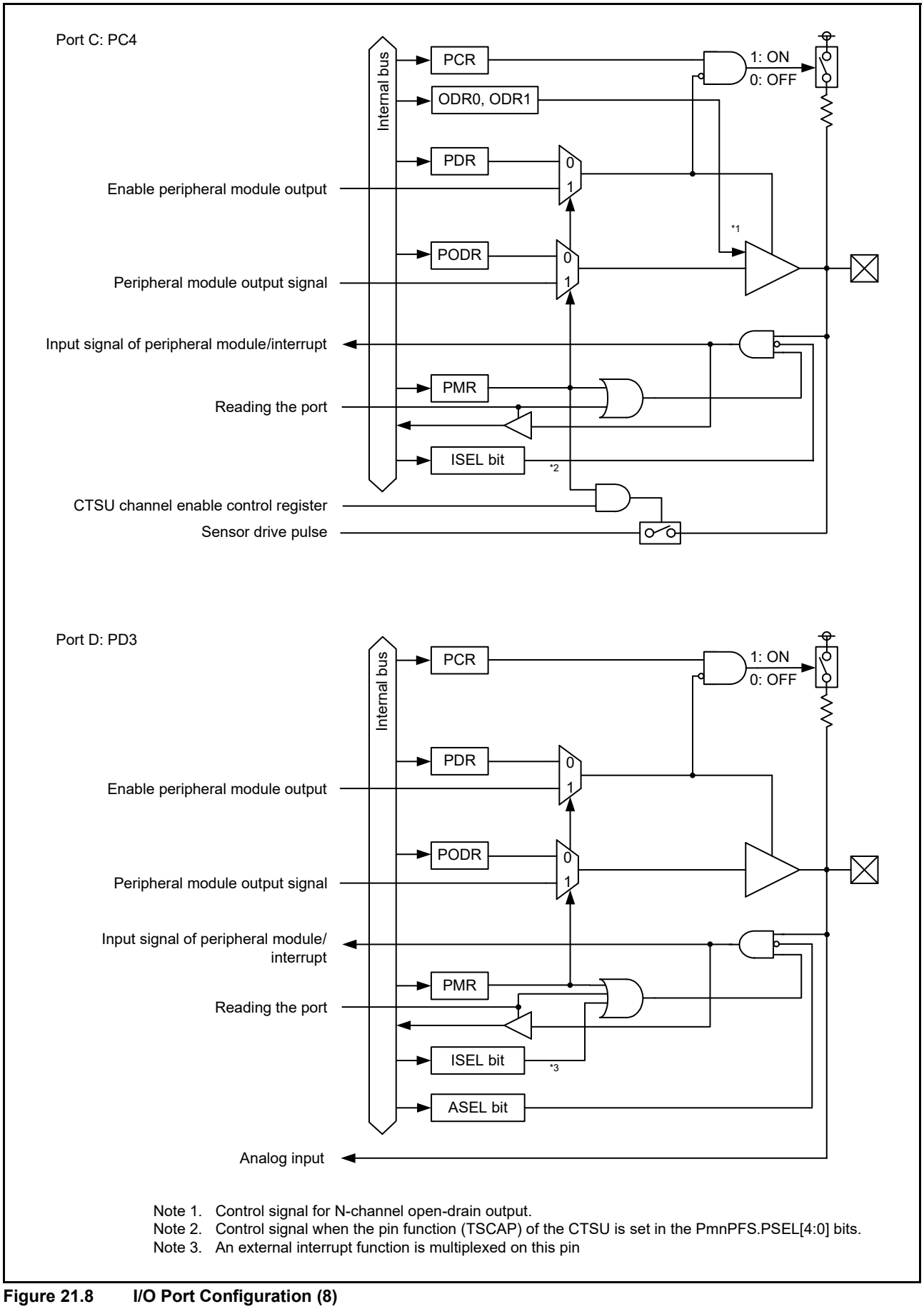


Figure 21.7 I/O Port Configuration (7)

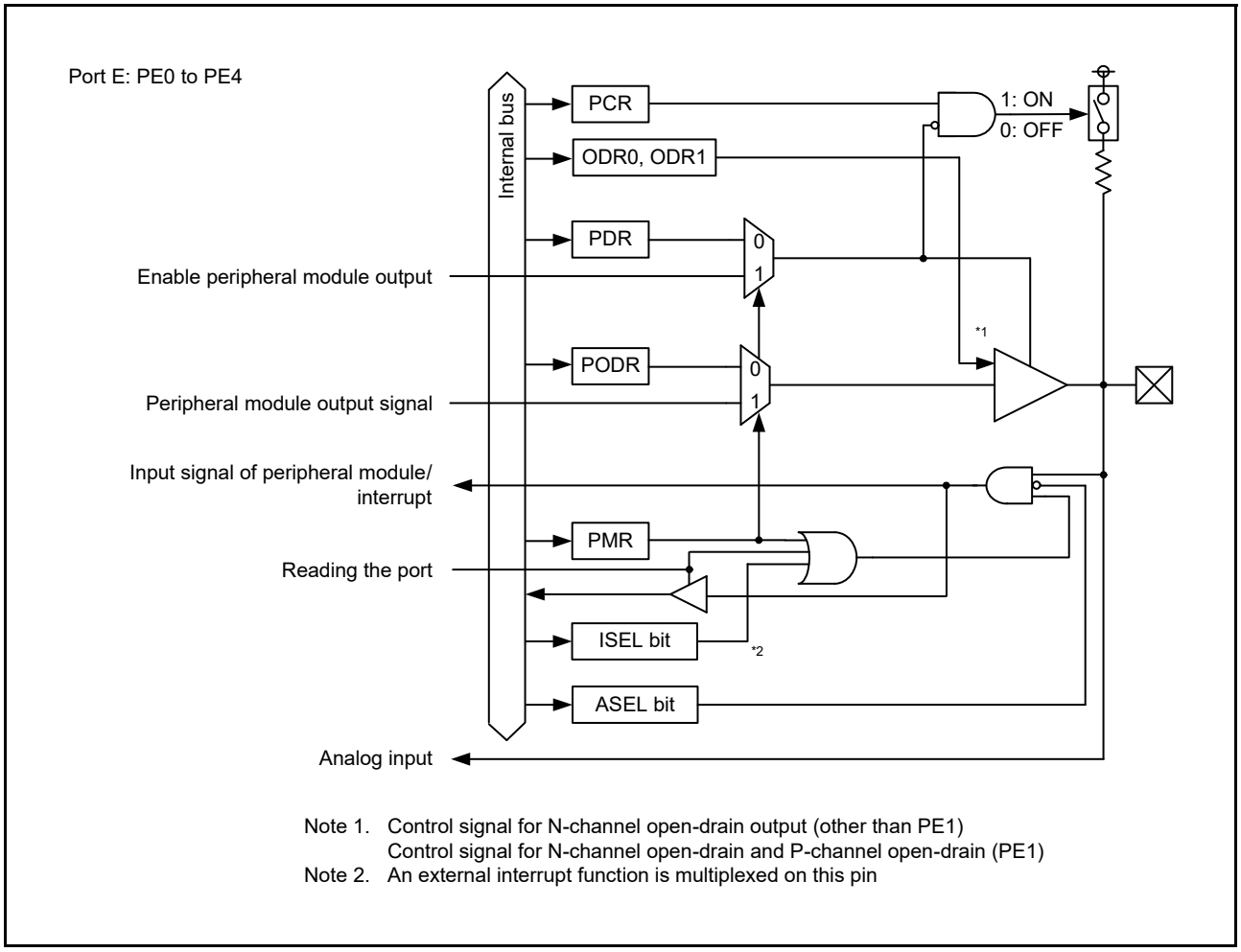
Downloaded from Arrow.com.



Note 1. Control signal for N-channel open-drain output.
Note 2. Control signal when the pin function (TSCAP) of the CTSU is set in the PmnPFS.PSEL[4:0] bits.
Note 3. An external interrupt function is multiplexed on this pin

Figure 21.8 I/O Port Configuration (8)

Downloaded from Arrow.com.



Note 1. Control signal for N-channel open-drain output (other than PE1)
Control signal for N-channel open-drain and P-channel open-drain (PE1)
Note 2. An external interrupt function is multiplexed on this pin

Figure 21.9 I/O Port Configuration (9)

Downloaded from Arrow.com.

21.3 Register Descriptions

21.3.1 Port Direction Register (PDR)

Address(es): PORT0.PDR 0008 C000h, PORT1.PDR 0008 C001h, PORT2.PDR 0008 C002h, PORT3.PDR 0008 C003h, PORT4.PDR 0008 C004h, PORTB.PDR 0008 C00Bh, PORTC.PDR 0008 C00Ch, PORTD.PDR 0008 C00Dh, PORTE.PDR 0008 C00Eh, PORTJ.PDR 0008 C012h

	b7	b6	b5	b4	b3	b2	b1	b0
	B7	B6	B5	B4	B3	B2	B1	B0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	B0	Pm0 I/O Select	0: Input (Functions as an input pin.) 1: Output (Functions as an output pin.)	R/W
b1	B1	Pm1 I/O Select		R/W
b2	B2	Pm2 I/O Select		R/W
b3	B3	Pm3 I/O Select		R/W
b4	B4	Pm4 I/O Select		R/W
b5	B5	Pm5 I/O Select		R/W
b6	B6	Pm6 I/O Select		R/W
b7	B7	Pm7 I/O Select		R/W

m = 0 to 4, B to E, J

PDR is used to select the input or output direction for individual pins of the corresponding port m when the pins are configured as the general I/O pins.

Each bit of PORTm.PDR corresponds to each pin of port m; I/O direction can be specified in 1-bit units.

Each bit of PDR corresponding to port m that does not exist is reserved. Make settings according to the description in section 21.4, Initialization of the Port Direction Register (PDR).

The B2 and B3 bits of PORT1.PDR, B0 bit of PORT2.PDR, and B2, B3, and B4 bits of PORT3.PDR register are also reserved. Values read from reserved bits are undefined. When writing, read the register and then rewrite the values that were read to the reserved bit or bits.

The PORT3.PDR.B5 bit is reserved, because the P35 pin is input only. A reserved bit is read as 0. The write value should be 0.

Downloaded from Arrow.com.

21.3.2 Port Output Data Register (PODR)

Address(es): PORT0.PODR 0008 C020h, PORT1.PODR 0008 C021h, PORT2.PODR 0008 C022h, PORT3.PODR 0008 C023h, PORT4.PODR 0008 C024h, PORTB.PODR 0008 C02Bh, PORTC.PODR 0008 C02Ch, PORTD.PODR 0008 C02Dh, PORTE.PODR 0008 C02Eh, PORTJ.PODR 0008 C032h

	b7	b6	b5	b4	b3	b2	b1	b0
	B7	B6	B5	B4	B3	B2	B1	B0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	B0	Pm0 Output Data Store	Holds output data.	R/W
b1	B1	Pm1 Output Data Store		R/W
b2	B2	Pm2 Output Data Store		R/W
b3	B3	Pm3 Output Data Store		R/W
b4	B4	Pm4 Output Data Store		R/W
b5	B5	Pm5 Output Data Store		R/W
b6	B6	Pm6 Output Data Store		R/W
b7	B7	Pm7 Output Data Store		R/W

m = 0 to 4, B to E, J

PODR holds the data to be output from the pins used for general output ports.
Bits corresponding to port m in the 85-pin product but do not exist in the 56-pin product are reserved. Write 0 to these bits.
The B2 and B3 bits of PORT1.PODR, B0 bit of PORT2.PODR, and B2, B3, and B4 bits of PORT3.PODR register are reserved. Values read from reserved bits are undefined. When writing, read the register and then rewrite the values that were read to the reserved bit or bits.
The PORT3.PODR.B5 bit is reserved, because the P35 pin is input only. The bit corresponding to a pin that does not exist is reserved. A reserved bit is read as 0. The write value should be 0.

Downloaded from Arrow.com.

21.3.3 Port Input Data Register (PIDR)

Address(es): PORT0.PIDR 0008 C040h, PORT1.PIDR 0008 C041h, PORT2.PIDR 0008 C042h, PORT3.PIDR 0008 C043h, PORT4.PIDR 0008 C044h, PORTB.PIDR 0008 C04Bh, PORTC.PIDR 0008 C04Ch, PORTD.PIDR 0008 C04Dh, PORTE.PIDR 0008 C04Eh, PORTJ.PIDR 0008 C052h

	b7	b6	b5	b4	b3	b2	b1	b0
	B7	B6	B5	B4	B3	B2	B1	B0
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	B0	Pm0	Indicates individual pin states of the corresponding port.	R
b1	B1	Pm1		R
b2	B2	Pm2		R
b3	B3	Pm3		R
b4	B4	Pm4		R
b5	B5	Pm5		R
b6	B6	Pm6		R
b7	B7	Pm7		R

m = 0 to 4, B to E, J

PIDR indicates individual pin states of port m.
The pin states of port m can be read with the PORTm.PIDR, regardless of the values of PORTm.PDR and PORTm.PMR.
The NMI pin state is reflected in the P35 bit.
The bit corresponding to a pin that does not exist is reserved. A reserved bit is read as undefined, and cannot be modified.

Note: When using P36 and P37 as general I/O ports, set the MOSCCR.MOSTP bit to 1 (main clock oscillator is stopped) and the P36 and P37 control bits in the PORT3.PMR register to 0 (use pin as general I/O port).

Downloaded from Arrow.com.

21.3.4 Port Mode Register (PMR)

Address(es): PORT0.PMR 0008 C060h, PORT1.PMR 0008 C061h, PORT2.PMR 0008 C062h, PORT3.PMR 0008 C063h, PORT4.PMR 0008 C064h, PORTB.PMR 0008 C06Bh, PORTC.PMR 0008 C06Ch, PORTD.PMR 0008 C06Dh, PORTE.PMR 0008 C06Eh, PORTJ.PMR 0008 C072h

	b7	b6	b5	b4	b3	b2	b1	b0
	B7	B6	B5	B4	B3	B2	B1	B0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	B0	Pm0 Pin Mode Control	0: Use pin as general I/O port. 1: Use pin as I/O port for peripheral functions.	R/W
b1	B1	Pm1 Pin Mode Control		R/W
b2	B2	Pm2 Pin Mode Control		R/W
b3	B3	Pm3 Pin Mode Control		R/W
b4	B4	Pm4 Pin Mode Control		R/W
b5	B5	Pm5 Pin Mode Control		R/W
b6	B6	Pm6 Pin Mode Control		R/W
b7	B7	Pm7 Pin Mode Control		R/W

m = 0 to 4, B to E, J

Each bit of PORTm.PMR corresponds to each pin of port m; pin function can be specified in 1-bit units.
Bits corresponding to port m in the 85-pin product but do not exist in the 56-pin product are reserved. Write 0 to these bits.
The bit corresponding to a pin that does not exist is reserved. A reserved bit is read as 0. The write value should be 0.
The B2 and B3 bits of PORT1.PMR, B0 bit of PORT2.PMR, and B2, B3, and B4 bits of PORT3.PMR register are reserved. Values read from reserved bits are undefined. When writing, read the register and then rewrite the values that were read to the reserved bit or bits.

Downloaded from Arrow.com.

21.3.5 Open Drain Control Register 0 (ODR0)

Address(es): PORT2.ODR0 0008 C084h, PORT3.ODR0 0008 C086h, PORTB.ODR0 0008 C096h, PORTC.ODR0 0008 C098h, PORTE.ODR0 0008 C09Ch, PORTJ.ODR0 0008 C0A4h

	b7	b6	b5	b4	b3	b2	b1	b0
	B7	B6	B5	B4	B3	B2	B1	B0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	B0	Pm0 Output Type Select	0: CMOS output 1: N-channel open-drain	R/W
b1	B1	Reserved	This bit is read as 0. The write value should be 0.	R/W
b2	B2	Pm1 Output Type Select	• P21, P31, PB1 b2 0: CMOS output 1: N-channel open-drain b3 This bit is read as 0. The write value should be 0. • PE1 b3 b2 0 0: CMOS output 0 1: N-channel open-drain 1 0: P-channel open-drain 1 1: Hi-Z	R/W
b3	B3			R/W
b4	B4	Pm2 Output Type Select	0: CMOS output 1: N-channel open-drain	R/W
b5	B5	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	B6	Pm3 Output Type Select	0: CMOS output 1: N-channel open-drain	R/W
b7	B7	Reserved	This bit is read as 0. The write value should be 0.	R/W

m = 2, 3, B, C, E, J

Bits corresponding to port m in the 85-pin product but do not exist in the 56-pin product are reserved. Write 0 to these bits.

The bits corresponding to a pin that does not exist or pins with no open-drain output allocation are reserved. A reserved bit is read as 0. The write value should be 0.

Downloaded from Arrow.com.

21.3.6 Open Drain Control Register 1 (ODR1)

Address(es): PORT1.ODR1 0008 C083h, PORT2.ODR1 0008 C085h, PORT3.ODR1 0008 C087h, PORTB.ODR1 0008 C097h, PORTC.ODR1 0008 C099h, PORTE.ODR1 0008 C09Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	B7	B6	B5	B4	B3	B2	B1	B0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	B0	Pm4 Output Type Select	0: CMOS output 1: N-channel open-drain	R/W
b1	B1	Reserved	This bit is read as 0. The write value should be 0.	R/W
b2	B2	Pm5 Output Type Select	0: CMOS output 1: N-channel open-drain	R/W
b3	B3	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	B4	Pm6 Output Type Select	0: CMOS output 1: N-channel open-drain	R/W
b5	B5	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	B6	Pm7 Output Type Select	0: CMOS output 1: N-channel open-drain	R/W
b7	B7	Reserved	This bit is read as 0. The write value should be 0.	R/W

m = 1 to 3, B, C, E

Bits corresponding to port m in the 85-pin product but do not exist in the 56-pin product are reserved. Write 0 to these bits.

The PORT3.ODR1.B2 bit is reserved, because the P35 pin is input only.

The bits corresponding to a pin that does not exist or pins with no open-drain output allocation are reserved. A reserved bit is read as 0. The write value should be 0.

Downloaded from Arrow.com.

21.3.7 Pull-Up Control Register (PCR)

Address(es): PORT0.PCR 0008 C0C0h, PORT1.PCR 0008 C0C1h, PORT2.PCR 0008 C0C2h, PORT3.PCR 0008 C0C3h, PORT4.PCR 0008 C0C4h, PORTB.PCR 0008 C0CBh, PORTC.PCR 0008 C0CCh, PORTD.PCR 0008 C0CDh, PORTE.PCR 0008 C0CEh, PORTJ.PCR 0008 C0D2h

	b7	b6	b5	b4	b3	b2	b1	b0
	B7	B6	B5	B4	B3	B2	B1	B0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	B0	Pm0 Input Pull-Up Resistor Control	0: Disables an input pull-up resistor. 1: Enables an input pull-up resistor.	R/W
b1	B1	Pm1 Input Pull-Up Resistor Control		R/W
b2	B2	Pm2 Input Pull-Up Resistor Control		R/W
b3	B3	Pm3 Input Pull-Up Resistor Control		R/W
b4	B4	Pm4 Input Pull-Up Resistor Control		R/W
b5	B5	Pm5 Input Pull-Up Resistor Control		R/W
b6	B6	Pm6 Input Pull-Up Resistor Control		R/W
b7	B7	Pm7 Input Pull-Up Resistor Control		R/W

m = 0 to 4, B to E, J

While a pin is in the input state with the corresponding bit in PORTm.PCR set to 1, the pull-up resistor connected to the pin is enabled.

When a pin is used as a general port output pin, or a peripheral function output pin, the pull-up resistor for the pin is disabled regardless of the settings of the PCR register.

The pull-up resistor is also disabled in the reset state.

The B5 bit in PORT3.PCR is reserved. The bit corresponding to a pin that does not exist is also reserved. A reserved bit is read as 0. The write value should be 0.

Downloaded from Arrow.com.

21.3.8 Drive Capacity Control Register (DSCR)

Address(es): PORT1.DSCR 0008 C0E1h, PORT2.DSCR 0008 C0E2h, PORT3.DSCR 0008 C0E3h, PORTB.DSCR 0008 C0EBh, PORTC.DSCR 0008 C0ECh, PORTD.DSCR 0008 C0EDh, PORTE.DSCR 0008 C0EEh, PORTJ.DSCR 0008 C0F2h

	b7	b6	b5	b4	b3	b2	b1	b0
	B7	B6	B5	B4	B3	B2	B1	B0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	B0	Pm0 Drive Capacity Control	0: Normal drive output 1: High-drive output	R/W
b1	B1	Pm1 Drive Capacity Control		R/W
b2	B2	Pm2 Drive Capacity Control		R/W
b3	B3	Pm3 Drive Capacity Control		R/W
b4	B4	Pm4 Drive Capacity Control		R/W
b5	B5	Pm5 Drive Capacity Control		R/W
b6	B6	Pm6 Drive Capacity Control		R/W
b7	B7	Pm7 Drive Capacity Control		R/W

m = 1 to 3, B to E, J

The bit corresponding to a pin with the fixed drive capacity can be read from or written to. However, the drive capacity cannot be changed.
When high-drive output is selected, switching noise increases compared to when normal output is selected. Carefully evaluate the effect of noise on the MCU caused by adjacent pins before selecting high-drive output.
The bit corresponding to a pin that does not exist is reserved. A reserved bit is read as 0. The write value should be 0.

Downloaded from Arrow.com.

21.4 Initialization of the Port Direction Register (PDR)

Initialize reserved bits in the PDR register according to Table 21.3 and Table 21.4.

- The blank columns in Table 21.3 and Table 21.4 indicate the bits corresponding to the pins listed in Table 21.1, Specifications of I/O Ports.
  The corresponding bits should be set to 1 (output) or 0 (input) depending on the user system.
  However, the PORT3.PDR.B5 bit of the input-only P35 pin is reserved.
  This bit should be set to 0 (input).
- The columns other than the blank columns in Table 21.3 and Table 21.4 indicate reserved bits.
  A reserved bit should be set to 0 (input) or 1 (output) according to Table 21.3 and Table 21.4.
  The B2 and B3 bits of PORT1.PDR, B0 bit of PORT2.PDR, and B2, B3, and B4 bits of PORT3.PDR register are reserved. Values read from reserved bits are undefined. When writing, read the register and then rewrite the values that were read to the reserved bit or bits.
  When setting a value to a reserved bit, access in byte units.

Table 21.3 PDR Register Settings in 85-Pin and 83-Pin Packages

	PDR Register							
Port Symbol	**b7**	**b6**	**b5**	**b4**	**b3**	**b2**	**b1**	**b0**
PORT0		1		1		1	1	1
PORT1					×	×	1	1
PORT2				1	1			×
PORT3			0	×	×	×		
PORT4								
PORTB		1		1		1		
PORTC							1	
PORTD	1	1	1	1		1	1	1
PORTE	1	1	1					
PORTJ	1	1	1	1		1	1	1

x: Undefined

Table 21.4 PDR Register Settings in 56-Pin Packages

	PDR Register							
Port Symbol	**b7**	**b6**	**b5**	**b4**	**b3**	**b2**	**b1**	**b0**
PORT0	1	1		1	1	1	1	1
PORT1					×	×	1	1
PORT2			1	1	1	1	1	×
PORT3			0	×	×	×		
PORT4				1	1	1		1
PORTB		1	1	1	1	1		
PORTC							1	
PORTD	1	1	1	1		1	1	1
PORTE	1	1	1				1	1
PORTJ	1	1	1	1	1	1	1	1

x: Undefined

Downloaded from Arrow.com.

21.5 Handling of Unused Pins

The configuration of unused pins is listed in Table 21.5.

Table 21.5 Unused Pin Configuration

Pin Name	Description
VBATT	Connect this pin to VCC.
MD	(Always used as mode pins)
RES#	Connect this pin to VCC via a pull-up resistor.
P35/NMI	Connect this pin to VCC via a pull-up resistor.
USB0_DM, USB0_DP	Leave this pin open.
P36/EXTAL	When the main clock is not used, set the MOSCCR.MOSTP bit to 1 (general port P36). When this pin is not used as port P36 either, it is configured in the same way as port 1 to 3, B to E, J.
P37/XTAL	When the main clock is not used, set the MOSCCR.MOSTP bit to 1 (general port P37). When this pin is not used as port P37 either, it is configured in the same way as port 1 to 3, B to E, J. When the external clock is input to the EXTAL pin, leave this pin open.
XCIN	Connect this pin to VSS via a pull-down resistor.
XCOUT	Leave this pin open.
Ports 1 to 3, Ports B to E, J	• If the direction setting is for input (PORTn.PDR = 0), the corresponding pin is connected to VCC (pulled up) via a resistor or to VSS (pulled down) via a resistor.*1 • If the direction setting is for output (PORTn.PDR = 1), the pin is released.*1, *2
Ports 0, 4	• If the direction setting is for input (PORTn.PDR = 0), the corresponding pin is connected to AVCC0 (pulled up) via a resistor or to AVSS0 (pulled down) via a resistor.*1 • If the direction setting is for output (PORTn.PDR = 1), the pin is released.*1, *2
VREFH0	Connect this pin to AVCC0.
VREFL0	Connect this pin to AVSS0.

Note 1. Clear the PORTn.PMR bit, the PmnPFS.ISEL bit and the PmnPFS.ASEL bit to 0.
Note 2. In the case of release when the setting is for output, the port is an input over the period from release from the reset state to the pin becoming an output. Since the voltage on the pin is undefined while it is an input, this may lead to an increase in the current drawn.

Downloaded from Arrow.com.

22. Multi-Function Pin Controller (MPC)

22.1 Overview

The multi-function pin controller (MPC) is used to allocate input and output signals for peripheral modules and input interrupt signals to pins from among multiple ports.

Table 22.1 shows the allocation of pin functions to multiple pins. The symbols ✓ and × in the table indicate whether the pins are or are not present on the given package. Allocating the same function to more than one pin is prohibited.

Table 22.1 Allocation of Pin Functions to Multiple Pins (1/6)

Module/Function	Channel	Pin Functions	Allocation Port	Package	
				85-pin, 83-pin	56-pin
Interrupt		NMI (input)	P35	✓	✓
Interrupt		IRQ0 (input)	P30	✓	✓
		IRQ1 (input)	P31	✓	✓
		IRQ4 (input)	PB1	✓	✓
			P14	✓	✓
		IRQ5 (input)	P15	✓	✓
		IRQ6 (input)	P16	✓	✓
		IRQ7 (input)	PE2	✓	✓
			P17	✓	✓
Clock generation circuit		CLKOUT (output)	PE3	✓	✓
			PE4	✓	✓
Multi-function timer unit 2	MTU0	MTIOC0A (input/output)	PB3	✓	×
		MTIOC0B (input/output)	P15	✓	✓
		MTIOC0C (input/output)	PB1	✓	✓
	MTU1	MTIOC1A (input/output)	PE4	✓	✓
		MTIOC1B (input/output)	P21	✓	×
			PB5	✓	×
	MTU2	MTIOC2A (input/output)	P26	✓	✓
			PB5	✓	×
		MTIOC2B (input/output)	P27	✓	✓
	MTU3	MTIOC3A (input/output)	P14	✓	✓
			P17	✓	✓
			PC7	✓	✓
		MTIOC3B (input/output)	P17	✓	✓
			P22	✓	×
			PB7	✓	✓
			PC5	✓	✓
		MTIOC3C (input/output)	P16	✓	✓
			PC0	✓	✓
			PC6	✓	✓
			PJ3	✓	×
		MTIOC3D (input/output)	P16	✓	✓
			PC4	✓	✓

Downloaded from Arrow.com.

Table 22.1 Allocation of Pin Functions to Multiple Pins (2/6)

Module/Function	Channel	Pin Functions	Allocation Port	Package	
				85-pin, 83-pin	56-pin
Multi-function timer unit 2	MTU4	MTIOC4A (input/output)	PB3	✓	×
			PE2	✓	✓
		MTIOC4B (input/output)	P30	✓	✓
			PC2	✓	✓
			PE3	✓	✓
		MTIOC4C (input/output)	P25	✓	×
			PB1	✓	✓
			PE1	✓	×
		MTIOC4D (input/output)	P31	✓	✓
			PC3	✓	✓
			PE4	✓	✓
	MTU	MTCLKA (input)	P14	✓	✓
			PC6	✓	✓
		MTCLKB (input)	P15	✓	✓
			P25	✓	×
			PC7	✓	✓
		MTCLKC (input)	P22	✓	×
			PC4	✓	✓
		MTCLKD (input)	PC5	✓	✓
Port output enable 2	POE0#	POE0# (input)	PC4	✓	✓
		POE1# (input)	PB5	✓	×
		POE3# (input)	PB3	✓	×
		POE8# (input)	P17	✓	✓
			P30	✓	✓
			PD3	✓	✓
			PE3	✓	✓
16-bit timer pulse unit	TPU0	TIOCB0 (input/output)	P17	✓	✓
	TPU1	TIOCB1 (input/output)	P16	✓	✓
	TPU2	TIOCB2 (input/output)	P15	✓	✓
	TPU3	TIOCA3 (input/output)	P21	✓	×
			PB0	✓	✓
		TIOCB3 (input/output)	PB1	✓	✓
		TIOCC3 (input/output)	P22	✓	×
		TIOCD3 (input/output)	PB3	✓	×
	TPU4	TIOCA4 (input/output)	P25	✓	×
		TIOCB4 (input/output)	PB5	✓	×
	TPU5	TIOCB5 (input/output)	P14	✓	✓
			PB7	✓	✓

Downloaded from Arrow.com.

Table 22.1 Allocation of Pin Functions to Multiple Pins (3/6)

Module/Function	Channel	Pin Functions	Allocation Port	Package	
				85-pin, 83-pin	56-pin
16-bit timer pulse unit	TPU	TCLKA (input)	P14	✓	✓
			PC2	✓	✓
		TCLKB (input)	P15	✓	✓
			PC3	✓	✓
		TCLKC (input)	P16	✓	✓
			PC0	✓	✓
		TCLKD (input)	P17	✓	✓
			PB3	✓	×
8-bit timer	TMR0	TMO0 (output)	P22	✓	×
			PB3	✓	×
		TMCI0 (input)	P21	✓	×
			PB1	✓	✓
	TMR1	TMO1 (output)	P17	✓	✓
			P26	✓	✓
		TMCI1 (input)	PC4	✓	✓
		TMRI1 (input)	PB5	✓	×
	TMR2	TMO2 (output)	P16	✓	✓
			PC7	✓	✓
		TMCI2 (input)	P15	✓	✓
			P31	✓	✓
			PC6	✓	✓
		TMRI2 (input)	P14	✓	✓
			PC5	✓	✓
	TMR3	TMCI3 (input)	P27	✓	✓
		TMRI3 (input)	P30	✓	✓
Serial communications interface	SCI1	RXD1 (input)/SMISO1 (input/output)/ SSCL1 (input/output)	P15	✓	✓
			P30	✓	✓
		TXD1 (output)/SMOSI1 (input/output)/ SSDA1 (input/output)	P16	✓	✓
			P26	✓	✓
		SCK1 (input/output)	P17	✓	✓
			P27	✓	✓
		CTS1# (input)/RTS1# (output)/SS1# (input)	P14	✓	✓
			P31	✓	✓
	SCI5	RXD5 (input)/SMISO5 (input/output)/ SSCL5 (input/output)	PC2	✓	✓
		TXD5 (output)/SMOSI5 (input/output)/ SSDA5 (input/output)	PC3	✓	✓
		SCK5 (input/output)	PC4	✓	✓
		CTS5# (input)/RTS5# (output)/SS5# (input/output)	PC0	✓	✓

Downloaded from Arrow.com.

Table 22.1 Allocation of Pin Functions to Multiple Pins (4/6)

Module/Function	Channel	Pin Functions	Allocation Port	Package	
				85-pin, 83-pin	56-pin
Serial communications interface	SCI8	RXD8 (input)/SMISO8 (input/output)/ SSCL8 (input/output)	PC6	✓	✓
		TXD8 (output)/SMOSI8 (input/output)/ SSDA8 (input/output)	PC7	✓	✓
		SCK8 (input/output)	PC5	✓	✓
		CTS8# (input)/RTS8# (output)/SS8# (input)	PC4	✓	✓
	SCI12	RXD12 (input)/SMISO12 (input/output)/ SSCL12 (input/output)/RXDX12 (input)	PE2	✓	×
		TXD12 (output)/SMOSI12 (input/ output)/SSDA12 (input/output)/TXDX12 (output)/SIOX12 (input/output)	PE1	✓	×
		SCK12 (input/output)	PE0	✓	×
		CTS12# (input)/RTS12# (output)/SS12# (input)	PE3	✓	×
I^2C bus interface	RIIC0	SCL (input/output)	P16	✓	✓
		SDA (input/output)	P17	✓	✓
Serial peripheral interface	RSPI0	RSPCKA (input/output)	PB0	✓	✓
			PC5	✓	✓
		MOSIA (input/output)	P16	✓	✓
			PC6	✓	✓
		MISOA (input/output)	P17	✓	✓
			PC7	✓	✓
		SSLA0 (input/output)	PC4	✓	✓
		SSLA1 (output)	PC0	✓	✓
		SSLA3 (output)	PC2	✓	✓
CAN module		CRXD0 (input)	P15	✓	✓
		CTXD0 (output)	P14	✓	✓
Realtime clock		RTCOUT (output)	P16	✓	✓
		RTCIC0 (input)*1	P30	✓	✓
		RTCIC1 (input)*1	P31	✓	✓
IrDA interface		IRTXD5 (output)	PC3	✓	✓
		IRRXD5 (input)	PC2	✓	✓
Serial sound interface		SSISCK0 (input/output)	P31	✓	✓
		SSIWS0 (input/output)	P21	✓	×
			P27	✓	✓
		SSITXD0 (output)	P17	✓	✓
		SSIRXD0 (input)	P26	✓	✓
		AUDIO_MCLK (input)	P22	✓	×
			P30	✓	✓
			PE3	✓	✓

Downloaded from Arrow.com.

Table 22.1 Allocation of Pin Functions to Multiple Pins (5/6)

Module/Function	Channel	Pin Functions	Allocation Port	Package	
				85-pin, 83-pin	56-pin
SD host interface		SDHI_CLK (output)	PB1	✓	×
		SDHI_CMD (input/output)	PB0	✓	×
		SDHI_D0 (input/output)	PC3	✓	×
		SDHI_D1 (input/output)	PC4	✓	×
		SDHI_D2 (input/output)	PB7	✓	×
		SDHI_D3 (input/output)	PC2	✓	×
		SDHI_CD (input)	PB5	✓	×
		SDHI_WP (input)	PB3	✓	×
USB 2.0 host/function module		USB0_VBUS (input)	P16	✓	✓
			PB5	✓	×
		USB0_EXICEN (output)	P21	✓	×
			PC6	✓	✓
		USB0_VBUSEN (output)	P16	✓	✓
			P26	✓	✓
		USB0_OVRCURA (input)	P14	✓	✓
		USB0_OVRCURB (input)	P16	✓	✓
			P22	✓	×
		USB0_ID (input)	PC5	✓	✓
12-bit A/D converter		AN000 (input)*1	P40	✓	×
		AN001 (input)*1	P41	✓	✓
		AN002 (input)*1	P42	✓	×
		AN003 (input)*1	P43	✓	×
		AN004 (input)*1	P44	✓	×
		AN005 (input)*1	P45	✓	✓
		AN006 (input)*1	P46	✓	✓
		AN007 (input)*1	P47	✓	✓
		AN016 (input)*1	PE0	✓	×
		AN017 (input)*1	PE1	✓	×
		AN018 (input)*1	PE2	✓	✓
		AN019 (input)*1	PE3	✓	✓
		AN020 (input)*1	PE4	✓	✓
		AN027 (input)	PD3	✓	✓
		ADTRG0# (input)	P07	✓	×
			P16	✓	✓
			P25	✓	×
12-bit D/A converter		DA0 (output)*1	P03	✓	×
		DA1 (output)*1	P05	✓	✓
Clock frequency accuracy measurement circuit		CACREF (input)	PC7	✓	✓

Downloaded from Arrow.com.

Table 22.1 Allocation of Pin Functions to Multiple Pins (6/6)

Module/Function	Channel	Pin Functions	Allocation Port	Package	
				85-pin, 83-pin	56-pin
Comparator B		CMPB2 (input)*1	P15	✓	✓
		CVREFB2 (input)*1	P14	✓	✓
		CMPB3 (input)*1	P26	✓	✓
		CVREFB3 (input)*1	P27	✓	✓
		CMPOB2 (output)	P17	✓	✓
		CMPOB3 (output)	P30	✓	✓
Capacitive touch sensing unit (CTSU)		TSCAP (output)	PC4	✓	✓
		TS2 (output)	P27	✓	✓
		TS3 (output)	P26	✓	✓
		TS4 (output)	P25	✓	×
		TS7 (output)	P22	✓	×
		TS8 (output)	P21	✓	×
		TS12 (output)	P15	✓	✓
		TS13 (output)	P14	✓	✓
		TS22 (output)	PC6	✓	✓
		TS23 (output)	PC5	✓	✓
		TS27 (output)	PC3	✓	✓
		TS30 (output)	PC2	✓	✓
		TS35 (output)	PC0	✓	✓

Note 1. Select general input (by setting the Bm bits for the given pin in the PDR and PMR for the given port to 0) for the pin if this pin function is to be used.

Downloaded from Arrow.com.

22.2 Register Descriptions

Registers and bits for pins that are not present due to differences according to the package are reserved. Write the value after a reset when writing to such bits.

22.2.1 Write-Protect Register (PWPR)

Address(es): 0008 C11Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	B0WI	PFSWE	—	—	—	—	—	—
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	PFSWE	PFS Register Write Enable	0: Writing to the PFS register is disabled 1: Writing to the PFS register is enabled	R/W
b7	B0WI	PFSWE Bit Write Disable	0: Writing to the PFSWE bit is enabled 1: Writing to the PFSWE bit is disabled	R/W

PFSWE Bit (PFS Register Write Enable)

Writing to PmnPFS register is enabled only when the PFSWE bit is set to 1.
To set the PFSWE bit to 1, write 1 to the PFSWE bit after writing 0 to the B0WI bit.

B0WI Bit (PFSWE Bit Write Disable)

Writing to the PFSWE bit is enabled only when the B0WI bit is set to 0.

Downloaded from Arrow.com.

22.2.2 P0n Pin Function Control Register (P0nPFS) (n = 3, 5, 7)

Address(es): P03PFS 0008 C143h, P05PFS 0008 C145h, P07PFS 0008 C147h

	b7	b6	b5	b4	b3	b2	b1	b0
	ASEL	—	—	PSEL[4:0]				
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see the tables below.	R/W
b6, b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	ASEL	Analog Function Select	0: Used other than as analog pin 1: Used as analog pin P03: DA0 (85/83 pins) P05: DA1 (85/83/56 pins)	R/W

The Pmn pin function control register (PmnPFS) selects the pin function.
Bits PSEL[4:0] select the peripheral function assigned to each port pin.
The ISEL bit is set when a pin is used as an IRQ input pin. This setting can be used with the combination of the peripheral function, though IRQn (external pin interrupt) of the same number should not be enabled by two or more pins.
The ASEL bit is set when a pin is used as an analog pin. When switching a pin to analog using the ASEL bit, set the corresponding port mode register bit (PORTm.PMR) to “general I/O port” and the port direction register bit (PORTm.PDR) to “input”. The pin state cannot be read at this point. The PmnPFS register is protected by the write-protect register (PWPR). Modify the register after releasing the protection.
The ISEL bit to which IRQn is not specified is reserved. The ASEL bit to which analog input/output is not specified is reserved.

Table 22.2 Register Settings for Input/Output Pin Function in 85-Pin and 83-Pin

PSEL[4:0] Settings	Pin		
	P03	P05	P07
00000b (initial value)	Hi-Z		
01001b	—	—	ADTRG0#

—: Do not specify this value.

Downloaded from Arrow.com.

22.2.3 P1n Pin Function Control Registers (P1nPFS) (n = 4 to 7)

Address(es): P14PFS 0008 C14Ch, P15PFS 0008 C14Dh, P16PFS 0008 C14Eh, P17PFS 0008 C14Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	ASEL	ISEL	—	PSEL[4:0]				
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see the tables below.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	ISEL	Interrupt Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin P14: IRQ4 input switch (85/83/56 pins) P15: IRQ5 input switch (85/83/56 pins) P16: IRQ6 input switch (85/83/56 pins) P17: IRQ7 input switch (85/83/56 pins)	R/W
b7	ASEL	Analog Function Select	0: Used other than as analog pin 1: Used as analog pin P14: CVREFB2 (85/83/56 pins) P15: CMPB2 (85/83/56 pins)	R/W

Table 22.3 Register Settings for Input/Output Pin Function in 85-Pin, 83-Pin, and 56-Pin

PSEL[4:0] Settings	Pin			
	P14	P15	P16	P17
00000b (initial value)	Hi-Z			
00001b	MTIOC3A	MTIOC0B	MTIOC3C	MTIOC3A
00010b	MTCLKA	MTCLKB	MTIOC3D	MTIOC3B
00011b	TIOCB5	TIOCB2	TIOCB1	TIOCB0
00100b	TCLKA	TCLKB	TCLKC	TCLKD
00101b	TMRI2	TMCI2	TMO2	TMO1
00111b	—	—	RTCOUT	POE8#
01001b	—	—	ADTRG0#	—
01010b	—	RXD1 SMISO1 SSCL1	TXD1 SMOSI1 SSDA1	SCK1
01011b	CTS1# RTS1# SS1#	—	—	—
01101b	—	—	MOSIA	MISOA
01111b	—	—	SCL	SDA
10000b	CTXD0	CRXD0	—	CMPOB2
10001b	USB0_OVRCURA	—	USB0_VBUS	—
10010b	—	—	USB0_VBUSEN	—
10011b	—	—	USB0_OVRCURB	—
10111b	—	—	—	SSITXD0
11001b	TS13	TS12	—	—

—: Do not specify this value.

Downloaded from Arrow.com.

22.2.4 P2n Pin Function Control Register (P2nPFS) (n = 1, 2, 5 to 7)

Address(es): P21PFS 0008 C151h, P22PFS 0008 C152h, P25PFS 0008 C155h, P26PFS 0008 C156h, P27PFS 0008 C157h

	b7	b6	b5	b4	b3	b2	b1	b0
	ASEL	—	—	PSEL[4:0]				
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see the tables below.	R/W
b6, b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	ASEL	Analog Function Select	0: Used other than as analog pin 1: Used as analog pin P26: CMPB3 (85/83/56 pins) P27: CVREFB3 (85/83/56 pins)	R/W

Table 22.4 Register Settings for Input/Output Pin Function in 85-Pin and 83-Pin

PSEL[4:0] Settings	Pin				
	P21	P22	P25	P26	P27
00000b (initial value)	Hi-Z				
00001b	MTIOC1B	MTIOC3B	MTIOC4C	MTIOC2A	MTIOC2B
00010b	—	MTCLKC	MTCLKB	—	—
00011b	TIOCA3	TIOCC3	TIOCA4	—	—
00101b	TMCI0	TMO0	—	TMO1	TMCI3
01001b	—	—	ADTRG0#	—	—
01010b	—	—	—	TXD1 SMOSI1 SSDA1	SCK1
10001b	USB0_EXICEN	USB0_OVRCURB	—	USB0_VBUSEN	—
10111b	SSIWS0	AUDIO_MCLK	—	SSIRXD0	SSIWS0
11001b	TS8	TS7	TS4	TS3	TS2

—: Do not specify this value.

Table 22.5 Register Settings for Input/Output Pin Function in 56-Pin

PSEL[4:0] Settings	Pin	
	P26	P27
00000b (initial value)	Hi-Z	
00001b	MTIOC2A	MTIOC2B
00101b	TMO1	TMCI3
01010b	TXD1 SMOSI1 SSDA1	SCK1
10001b	USB0_VBUSEN	—
10111b	SSIRXD0	SSIWS0
11001b	TS3	TS2

—: Do not specify this value.

Downloaded from Arrow.com.

22.2.5 P3n Pin Function Control Registers (P3nPFS) (n = 0, 1)

Address(es): P30PFS 0008 C158h, P31PFS 0008 C159h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	ISEL	—	PSEL[4:0]				
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see the tables below.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin P30: IRQ0 input switch (85/83/56 pins) P31: IRQ1 input switch (85/83/56 pins)	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Table 22.6 Register Settings for Input/Output Pin Function in 85-Pin, 83-Pin, and 56-Pin

PSEL[4:0] Settings	Pin	
	P30	P31
00000b (initial value)	Hi-Z	
00001b	MTIOC4B	MTIOC4D
00101b	TMRI3	TMCI2
00111b	POE8#	—
01010b	RXD1 SMISO1 SSCL1	—
01011b	—	CTS1# RTS1# SS1#
10000b	CMPOB3	—
10111b	AUDIO_MCLK	SSISCK0

—: Do not specify this value.

Downloaded from Arrow.com.

22.2.6 P4n Pin Function Control Registers (P4nPFS) (n = 0 to 7)

Address(es): P40PFS 0008 C160h, P41PFS 0008 C161h, P42PFS 0008 C162h, P43PFS 0008 C163h, P44PFS 0008 C164h, P45PFS 0008 C165h, P46PFS 0008 C166h, P47PFS 0008 C167h

	b7	b6	b5	b4	b3	b2	b1	b0
	ASEL	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	ASEL	Analog Function Select	0: Not used as an analog pin 1: Used as an analog pin P40: AN000 (85/83 pins) P41: AN001 (85/83/56 pins) P42: AN002 (85/83 pins) P43: AN003 (85/83 pins) P44: AN004 (85/83 pins) P45: AN005 (85/83/56 pins) P46: AN006 (85/83/56 pins) P47: AN007 (85/83/56 pins)	R/W

Downloaded from Arrow.com.

22.2.7 PBn Pin Function Control Registers (PBnPFS) (n = 0, 1, 3, 5, 7)

Address(es): PB0PFS 0008 C198h, PB1PFS 0008 C199h, PB3PFS 0008 C19Bh, PB5PFS 0008 C19Dh, PB7PFS 0008 C19Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	ISEL	—	PSEL[4:0]				
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see the tables below.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin PB1: IRQ4 (85/83/56 pins)	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Table 22.7 Register Settings for Input/Output Pin Function in 85-Pin and 83-Pin

PSEL[4:0] Settings	Pin				
	PB0	PB1	PB3	PB5	PB7
00000b (initial value)	Hi-Z				
00001b	—	MTIOC0C	MTIOC0A	MTIOC2A	MTIOC3B
00010b	—	MTIOC4C	MTIOC4A	MTIOC1B	—
00011b	TIOCA3	TIOCB3	TIOCD3	TIOCB4	TIOCB5
00100b	—	—	TCLKD	—	—
00101b	—	TMCI0	TMO0	TMRI1	—
00111b	—	—	POE3#	POE1#	—
01101b	RSPCKA	—	—	—	—
10001b	—	—	—	USB0_VBUS	—
11010b	SDHI_CMD	SDHI_CLK	SDHI_WP	SDHI_CD	SDHI_D2

—: Do not specify this value.

Table 22.8 Register Settings for Input/Output Pin Function in 56-Pin

PSEL[4:0] Settings	Pin		
	PB0	PB1	PB7
00000b (initial value)	Hi-Z		
00001b	—	MTIOC0C	MTIOC3B
00010b	—	MTIOC4C	—
00011b	TIOCA3	TIOCB3	TIOCB5
00101b	—	TMCI0	—
01101b	RSPCKA	—	—

—: Do not specify this value.

Downloaded from Arrow.com.

22.2.8 PCn Pin Function Control Registers (PCnPFS) (n = 0, 2 to 7)

Address(es): PC0PFS 0008 C1A0h, PC2PFS 0008 C1A2h, PC3PFS 0008 C1A3h, PC4PFS 0008 C1A4h, PC5PFS 0008 C1A5h, PC6PFS 0008 C1A6h, PC7PFS 0008 C1A7h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	PSEL[4:0]				
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see the tables below.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Table 22.9 Register Settings for Input/Output Pin Function in 85-Pin, 83-Pin, and 56-Pin

PSEL[4:0] Settings	Pin						
	PC0	PC2	PC3	PC4	PC5	PC6	PC7
00000b (initial value)	Hi-Z						
00001b	MTIOC3C	MTIOC4B	MTIOC4D	MTIOC3D	MTIOC3B	MTIOC3C	MTIOC3A
00010b	—	—	—	MTCLKC	MTCLKD	MTCLKA	MTCLKB
00011b	TCLKC	TCLKA	TCLKB	—	—	—	—
00101b	—	—	—	TMCI1	TMRI2	TMCI2	TMO2
00111b	—	—	—	POE0#	—	—	CACREF
10001b	—	—	—	—	USB0_ID	USB0_EXICEN	—
01010b	—	RXD5 SMISO5 SSCL5	TXD5 SMOSI5 SSDA5	SCK5	SCK8	RXD8 SMISO8 SSCL8	TXD8 SMOSI8 SSDA8
01011b	CTS5# RTS5# SS5#	—	—	CTS8# RTS8# SS8#	—	—	—
01101b	SSLA1	SSLA3	—	SSLA0	RSPCKA	MOSIA	MISOA
11001b	TS35	TS30	TS27	TSCAP	TS23	TS22	—
11010b	—	SDHI_D3*1	SDHI_D0*1	SDHI_D1*1	—	—	—

—: Do not specify this value.
Note 1. This setting is not supported by 56-pin products.

Downloaded from Arrow.com.

22.2.9 PDn Pin Function Control Registers (PDnPFS) (n = 3)

Address(es): PD3PFS 0008 C1ABh

	b7	b6	b5	b4	b3	b2	b1	b0
	ASEL	ISEL	—	PSEL[4:0]				
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see the tables below.	R/W
b6, b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	ASEL	Analog Function Select	0: Used other than as analog pin 1: Used as analog pin PD3: AN027 (85/83/56 pins)	R/W

Table 22.10 Register Settings for Input/Output Pin Function in 85-Pin, 83-Pin, and 56-Pin

PSEL[4:0] Settings	Pin
	PD3
00000b (initial value)	Hi-Z
00111b	POE8#

—: Do not specify this value.

Downloaded from Arrow.com.

22.2.10 PEn Pin Function Control Registers (PEnPFS) (n = 0 to 4)

Address(es): PE0PFS 0008 C1B0h, PE1PFS 0008 C1B1h, PE2PFS 0008 C1B2h, PE3PFS 0008 C1B3h, PE4PFS 0008 C1B4h

	b7	b6	b5	b4	b3	b2	b1	b0
	ASEL	ISEL	—	PSEL[4:0]				
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see the tables below.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin PE2: IRQ7 input switch (85/83/56 pins)	R/W
b7	ASEL	Analog Function Select	0: Not used as an analog pin 1: Used as an analog pin PE0:AN016 (85/83 pins) PE1:AN017 (85/83 pins) PE2:AN018 (85/83/56 pins) PE3:AN019 (85/83/56 pins) PE4:AN020 (85/83/56 pins)	R/W

Table 22.11 Register Settings for Input/Output Pin Function in 85-Pin and 83-Pin

PSEL[4:0] Settings	Pin				
	PE0	PE1	PE2	PE3	PE4
00000b (initial value)	Hi-Z				
00001b	—	MTIOC4C	MTIOC4A	MTIOC4B	MTIOC4D
00010b	—	—	—	—	MTIOC1A
00111b	—	—	—	POE8#	—
01001b	—	—	—	CLKOUT	CLKOUT
01100b	SCK12	TXD12 TXDX12 SIOX12 SMOSI12 SSDA12	RXD12 RXDX12 SMISO12 SSCL12	CTS12# RTS12# SS12#	—
10111b	—	—	—	AUDIO_MCLK	—

—: Do not specify this value.

Downloaded from Arrow.com.

Table 22.12 Register Settings for Input/Output Pin Function in 56-Pin

PSEL[4:0] Settings	Pin		
	PE2	PE3	PE4
00000b (initial value)	Hi-Z		
00001b	MTIOC4A	MTIOC4B	MTIOC4D
00010b	—	—	MTIOC1A
00111b	—	POE8#	—
01001b	—	CLKOUT	CLKOUT
10111b	—	AUDIO_MCLK	—

—: Do not specify this value.

Downloaded from Arrow.com.

22.2.11 PJn Pin Function Control Registers (PJnPFS) (n = 3)

Address(es): PJ3PFS 0008 C1D3h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	PSEL[4:0]				
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see the tables below.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Table 22.13 Register Settings for Input/Output Pin Function in 85-Pin and 83-Pin

PSEL[4:0] Settings	Pin
	PJ3
00000b (initial value)	Hi-Z
00001b	MTIOC3C

Downloaded from Arrow.com.

22.3 Usage Notes

22.3.1 Procedure for Specifying Input/Output Pin Function

Use the following procedure to specify the input/output pin functions.

(1) Clear the port mode register (PMR) to 0 to select the general I/O port function.
(2) Specify the assignments of input/output signals for peripheral functions to the desired pins.
(3) Enable writing to the Pmn pin function control register (PmnPFS) through the write-protect register (PWPR) setting. (m = 0 to 4, B to E, J; n = 0 to 7)
(4) Specify the input/output function for the pin through the PSEL[4:0] bit settings in the PmnPFS register.
(5) Clear the PFSWE bit in the PWPR register to 0 to disable writing to the PmnPFS register.
(6) Set the PMR to 1 as necessary to switch to the selected input/output function for the pin.

22.3.2 Notes on MPC Register Setting

(1) Settings of the Pmn pin function control register (PmnPFS) should be made only while the PMR register for the target pin is cleared to 0. If a Pmn pin function control register is set while the PMR register is 1, unexpected edges may be input through the input pin or unexpected pulses are output through the output pin.
(2) Only the allowed values (functions) should be specified in the Pmn pin function control registers. If a value that is not allowed for the register is specified, correct operation is not guaranteed.
(3) Do not assign a single function to multiple pins through the MPC settings.
(4) Analog input functions for the A/D converter are multiplexed with pins of ports 4 and E. If a pin is to be used as an analog input, avoid loss of resolution by setting the given bits of the port mode register (PMR) and of the port direction register (PDR) to 0, i.e. configuring the pin as a general-purpose input, and setting the PmnPFS.ASEL bit to 1.
(5) Values of the TCEN time capture event input pin enable bits of time capture control registers y (RTCCRy, y = 0, 1) are not defined after a reset. To prohibit unwanted input, set these bits to 0.
(6) Points to note regarding the port mode register (PMR), port direction register (PDR), and Pmn pin function control register (PmnPFS) settings for pins that have multiplexed pin functions are listed in Table 22.14.

Downloaded from Arrow.com.

Table 22.14 Register Settings

Item	PMR.Bn	PDR.Bn	PmnPFS			Point to Note
			ASEL	ISEL	PSEL[4:0]	
After a reset	0	0	0	0	00000b	Pins function as general input port pins after release from the reset state.
General input ports	0	0	0	0/1	x	Set the ISEL bit to 1 if these are multiplexed with interrupt inputs.
General output ports	0	1	0	0	x	
Peripheral functions	1	x	0	0/1	Peripheral functions (see Table 22.2 to Table 22.13)	Set the ISEL bit to 1 if these are multiplexed with interrupt inputs.
Interrupt inputs	0	0	0	1	x	
NMI	x	x	x	x*1	x	Register settings are not required.
Analog inputs and outputs	0	0	1	x*1	x	Set these as general input port pins so that the output buffers are turned off.
RTC time-capture event-input pins	0	0	x	0/1	x	Set these as general input port pins so that the output buffers are turned off.
EXTAL/XTAL	0	0	x	x*1	x	Set these as general input port pins so that the output buffers are turned off.
XCIN/XCOUT	0	0	x	x*1	x	Set these as general input port pins so that the output buffers are turned off.

x: Setting not required.
0/1: Setting the PmnPFS.ISEL bit to 0 makes the pin incapable of functioning as an IRQ pin.
Setting the PmnPFS.ISEL bit to 1 makes the pin capable of functioning as an IRQ pin (if the IRQ is selected from the multiplexed functions).
Note 1. The pin does not function as the IRQn input pin even if the PmnPFS.ISEL bit is set to 1.

Note: The pin state is readable when the PmnPFS.ASEL bit is 0.
- If the value of the PmnPFS.PSEL[4:0] bits is to be changed, do so while the PMR.Bn bit is 0.
- If an RIIC function is assigned to a port pin, clear the PCR.Bn (to 0); pulling up is automatically turned off for outputs from peripheral modules other than the RIIC.
- If an input pin for time-capture events is not in use, clear the time capture event input pin enable bit (TCEN) in time capture control register y (RTCCRy) to 0 (disabled).

22.3.3 Note on Using Analog Functions

When an analog function is in use, configure the pin as a general-purpose input by setting the given bits of the port mode register (PMR) and of the port direction register (PDR) to 0, and then set the ASEL bit in the Pmn pin function control register (PmnPFS) to 1.

22.3.4 Notes on Using the CTSU Function of the Capacitive Touch Sensing Unit

When using the CTSU function (TSCAP, TSm: m = 2 to 4, 7, 8, 12, 13, 22, 23, 27, 30, 35) of the capacitive touch sensing unit, set the given bits of the port mode register (PMR), the port direction register (PDR), and the pull-up control register to 0. Then, use the PmnPFS.PSEL[4:0] bits to select the CTSU function and set the PMR register to 1. When a pin function of the capacitive touch sensing unit, do not use the pin as the IRQ input pin regardless of the ISEL setting of the corresponding bit.

Downloaded from Arrow.com.

23. Multi-Function Timer Pulse Unit 2 (MTU2a)

In this section, “PCLK” is used to refer to PCLKA.

23.1 Overview

This MCU has an on-chip multi-function timer pulse unit 2 (MTU). Each unit comprises a 16-bit timer with five channels (MTU0 to MTU4).
Table 23.1 lists the specifications of the MTU, and Table 23.2 lists the functions of the MTU. Figure 23.1 shows a block diagram of the MTU.

Table 23.1 MTU Specifications

Item	Description
Pulse input/output	15 lines max.
Count clocks	Eight clocks or seven clocks for each channel
Available operations	[MTU0 to MTU4] • Waveform output at compare match • Input capture function (noise filter set function) • Counter clear operation • Simultaneous writing to multiple timer counters (TCNT) • Simultaneous clearing by compare match or input capture • Simultaneous register input/output by synchronous counter operation • A maximum of 11-phase PWM output is available in combination with synchronous operation
	[MTU0, MTU3, MTU4] • Buffer operation specifiable • AC synchronous motor (brushless DC motor) drive mode using complementary PWM output and reset-synchronized PWM output is settable and the selection of two types of waveform outputs (chopping and level) is possible.
	[MTU1, MTU2] • Phase counting mode specifiable independently • Cascade connection operation
	[MTU3, MTU4] • A total of 6-phase waveform output, which includes three phases each for positive and negative complementary PWM or reset-synchronized PWM output, by interlocking operation
Complementary PWM mode	• Interrupts at the crest and trough of the counter value • A/D converter start triggers can be skipped
Interrupt sources	25 sources
Buffer operation	Automatic transfer of register data
Trigger generation	A/D converter start trigger can be generated
Low power consumption function	Module stop state can be set.

Downloaded from Arrow.com.

Table 23.2 MTU Functions (1/2)

Item		MTU0	MTU1	MTU2	MTU3	MTU4
Count clocks		PCLK/1 PCLK/4 PCLK/16 PCLK/64 MTCLKA MTCLKB MTCLKC MTCLKD	PCLK/1 PCLK/4 PCLK/16 PCLK/64 PCLK/256 MTCLKA MTCLKB	PCLK/1 PCLK/4 PCLK/16 PCLK/64 PCLK/1024 MTCLKA MTCLKB MTCLKC	PCLK/1 PCLK/4 PCLK/16 PCLK/64 PCLK/256 PCLK/1024 MTCLKA MTCLKB	PCLK/1 PCLK/4 PCLK/16 PCLK/64 PCLK/256 PCLK/1024 MTCLKA MTCLKB
External clocks for phase counting mode		—	MTCLKA MTCLKB	MTCLKC MTCLKD	—	—
General registers (TGR)		TGRA TGRB TGRE	TGRA TGRB	TGRA TGRB	TGRA TGRB	TGRA TGRB
General registers/ buffer registers		TGRC TGRD TGRF	—	—	TGRC TGRD	TGRC TGRD
I/O pins		MTIOC0A MTIOC0B MTIOC0C	MTIOC1A MTIOC1B	MTIOC2A MTIOC2B	MTIOC3A MTIOC3B MTIOC3C MTIOC3D	MTIOC4A MTIOC4B MTIOC4C MTIOC4D
Counter clear function		TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture
Compare match output	Low output	✓	✓	✓	✓	✓
	High output	✓	✓	✓	✓	✓
	Toggle output	✓	✓	✓	✓	✓
Input capture function		✓	✓	✓	✓	✓
Synchronous operation		✓	✓	✓	✓	✓
PWM mode 1		✓	✓	✓	✓	✓
PWM mode 2		✓	✓	✓	—	—
Complementary PWM mode		—	—	—	✓	✓
Reset-synchronized PWM		—	—	—	✓	✓
AC synchronous motor drive mode		✓	—	—	✓	✓
Phase counting mode		—	✓	✓	—	—
Buffer operation		✓	—	—	✓	✓
Dead time compensation counter function		—	—	—	—	—
DMAC activation		TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture
DTC activation		TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture and TCNT overflow or underflow
A/D converter start trigger		TGRA compare match or input capture TGRB compare match or input capture TGRE compare match TGRF compare match	TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture TCNT underflow (trough) in complementary PWM mode

Downloaded from Arrow.com.

Table 23.2 MTU Functions (2/2)

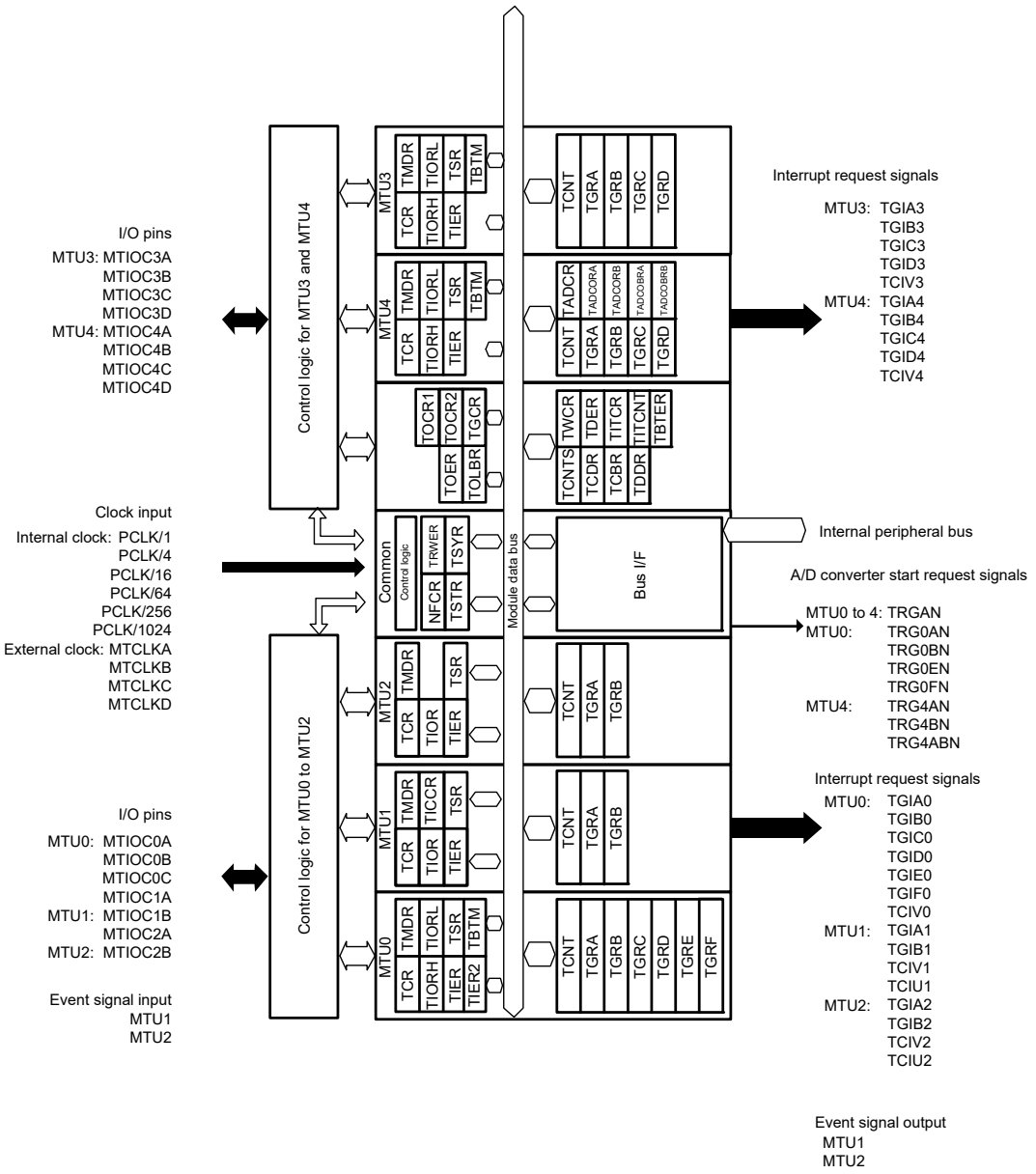
Item	MTU0	MTU1	MTU2	MTU3	MTU4
Interrupt sources	7 sources • Compare match or input capture 0A • Compare match or input capture 0B • Compare match or input capture 0C • Compare match or input capture 0D • Compare match 0E • Compare match 0F • Overflow	4 sources • Compare match or input capture 1A • Compare match or input capture 1B • Overflow • Underflow	4 sources • Compare match or input capture 2A • Compare match or input capture 2B • Overflow • Underflow	5 sources • Compare match or input capture 3A • Compare match or input capture 3B • Compare match or input capture 3C • Compare match or input capture 3D • Overflow	5 sources • Compare match or input capture 4A • Compare match or input capture 4B • Compare match or input capture 4C • Compare match or input capture 4D • Overflow or underflow
Event link function (output)	—	4 sources • Compare match 1A • Compare match 1B • Overflow • Underflow	4 sources • Compare match 2A • Compare match 2B • Overflow • Underflow	6 sources • Compare match 3A • Compare match 3B • Compare match 3C • Compare match 3D • Overflow • Underflow	6 sources • Compare match 4A • Compare match 4B • Compare match 4C • Compare match 4D • Overflow • Underflow
Event link function (input)	—	(1) Count start operation (2) Input capture operation (TRGA capture) (3) Count restart operation	(1) Count start operation (2) Input capture operation (TRGA capture) (3) Count restart operation	(1) Count start operation (2) Input capture operation (TRGA capture) (3) Count restart operation	(1) Count start operation (2) Input capture operation (TRGA capture) (3) Count restart operation
A/D converter start request delaying function	—	—	—	—	• A/D converter start request at a match between TADCORA and TCNT or A/D converter start request at a match between TADCORB and TCNT
Interrupt skipping function	—	—	—	• Skips TGRA compare match interrupts	• Skips TCIV interrupts
Module stop function	MSTPCRA.MSTPA9*1				

✓: Possible

—: Not possible

Note 1. For details on the module stop function, refer to section 11, Low Power Consumption.

Downloaded from Arrow.com.



TSTR:	Timer start register
TSYR:	Timer synchronous register
TCR:	Timer control register
NFCR:	Noise filter control register
TMDR:	Timer mode register
TIOR:	Timer I/O control register
TIORH:	Timer I/O control register H
TIORL:	Timer I/O control register L
TIER:	Timer interrupt enable register
TIER2:	Timer interrupt enable register 2
TGCR:	Timer gate control register
TOER:	Timer output master enable register
TOCR1:	Timer output control register 1
TOCR2:	Timer output control register 2
TSR:	Timer status register
TBTM:	Timer buffer operation transfer mode register
TCNT:	Timer counter
TCNTS:	Timer subcounter
TRWER:	Timer read/write enable register
TOLBR:	Timer output level buffer register
TCDR:	Timer cycle data register
TCBR:	Timer cycle buffer register
TDDR:	Timer dead time data register
TGRA:	Timer general register A
TGRB:	Timer general register B
TGRC:	Timer general register C
TGRD:	Timer general register D
TGRE:	Timer general register E
TGRF:	Timer general register F
TICCR:	Timer input capture control register
TWCR:	Timer waveform control register
TADCR:	Timer A/D converter start request control register
TADCORA:	Timer A/D converter start request cycle set register A
TADCORB:	Timer A/D converter start request cycle set register B
TADCOBRA:	Timer A/D converter start request cycle set buffer register A
TADCOBRB:	Timer A/D converter start request cycle set buffer register B
TITCR:	Timer interrupt skipping set registers
TITCNT:	Timer interrupt skipping counters register
TBTER:	Timer buffer transfer set register
TDER:	Timer dead time enable register

Figure 23.1 MTU Block Diagram

Downloaded from Arrow.com.

Table 23.3 lists the I/O pins to be used by the MTU.

Table 23.3 MTU I/O Pins

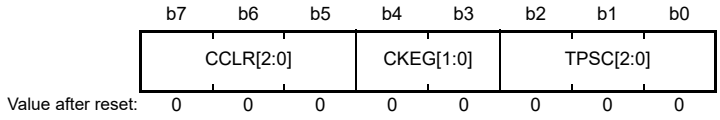
Module Symbol	Pin Name	I/O	Function
MTU	MTCLKA	Input	External clock A input pin (MTU1 phase counting mode A phase input)
	MTCLKB	Input	External clock B input pin (MTU1 phase counting mode B phase input)
	MTCLKC	Input	External clock C input pin (MTU2 phase counting mode A phase input)
	MTCLKD	Input	External clock D input pin (MTU2 phase counting mode B phase input)
MTU0	MTIOC0A	I/O	MTU0.TGRA input capture input/output compare output/PWM output pin
	MTIOC0B	I/O	MTU0.TGRB input capture input/output compare output/PWM output pin
	MTIOC0C	I/O	MTU0.TGRC input capture input/output compare output/PWM output pin
MTU1	MTIOC1A	I/O	MTU1.TGRA input capture input/output compare output/PWM output pin
	MTIOC1B	I/O	MTU1.TGRB input capture input/output compare output/PWM output pin
MTU2	MTIOC2A	I/O	MTU2.TGRA input capture input/output compare output/PWM output pin
	MTIOC2B	I/O	MTU2.TGRB input capture input/output compare output/PWM output pin
MTU3	MTIOC3A	I/O	MTU3.TGRA input capture input/output compare output/PWM output pin
	MTIOC3B	I/O	MTU3.TGRB input capture input/output compare output/PWM output pin
	MTIOC3C	I/O	MTU3.TGRC input capture input/output compare output/PWM output pin
	MTIOC3D	I/O	MTU3.TGRD input capture input/output compare output/PWM output pin
MTU4	MTIOC4A	I/O	MTU4.TGRA input capture input/output compare output/PWM output pin
	MTIOC4B	I/O	MTU4.TGRB input capture input/output compare output/PWM output pin
	MTIOC4C	I/O	MTU4.TGRC input capture input/output compare output/PWM output pin
	MTIOC4D	I/O	MTU4.TGRD input capture input/output compare output/PWM output pin

Downloaded from Arrow.com.

23.2 Register Descriptions

23.2.1 Timer Control Register (TCR)

Address(es): MTU0.TCR 000D 0B00h, MTU1.TCR 000D 0B80h, MTU2.TCR 000D 0C00h, MTU3.TCR 000D 0A00h, MTU4.TCR 000D 0A01h



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	TPSC[2:0]	Time Prescaler Select	Refer to Table 23.6 to Table 23.9.	R/W
b4, b3	CKEG[1:0]	Clock Edge Select	b4 b3 0 0: Count at rising edge 0 1: Count at falling edge 1 x: Count at both edges	R/W
b7 to b5	CCLR[2:0]	Counter Clear	Refer to Table 23.4 and Table 23.5.	R/W

x: Don't care

The MTU has a total of five TCR registers, one each for MTU0 to MTU4.
The TCR register controls the TCNT operation for each channel. The TCR register values should be specified only while the TCNT operation is stopped.

TPSC[2:0] Bits (Time Prescaler Select)

These bits select the TCNT count clock source. The count clock source can be selected independently for each channel. Refer to Table 23.6 to Table 23.9 for details.

CKEG[1:0] Bits (Clock Edge Select)

These bits select the clock edge. When the internal clock is counted at both edges, the count clock period is halved (e.g. PCLK/4 at both edges = PCLK/2 at rising edge). If phase counting mode is used on MTU1 and MTU2, the setting of these bits is ignored and the phase counting mode setting has priority. Internal clock edge selection is valid when the count clock source is PCLK/4 or slower. When PCLK/1 or the overflow/underflow in another channel is selected for the count clock source, a value can be written to these bits but counter operation compiles with the initial value.

CCLR[2:0] Bits (Counter Clear)

These bits select the TCNT counter clearing source. Refer to Table 23.4 and Table 23.5 for details.

Downloaded from Arrow.com.

Table 23.4 CCLR[2:0] (MTU0, MTU3, and MTU4)

Channel	Bit 7 CCLR[2]	Bit 6 CCLR[1]	Bit 5 CCLR[0]	Description
MTU0, MTU3, MTU4	0	0	0	TCNT clearing disabled
	0	0	1	TCNT cleared by TGRA compare match/input capture
	0	1	0	TCNT cleared by TGRB compare match/input capture
	0	1	1	TCNT cleared by counter clearing in another channel performing synchronous clearing/synchronous operation*1
	1	0	0	TCNT clearing disabled
	1	0	1	TCNT cleared by TGRC compare match/input capture*2
	1	1	0	TCNT cleared by TGRD compare match/input capture*2
	1	1	1	TCNT cleared by counter clearing in another channel performing synchronous clearing/synchronous operation*1

Note 1. Synchronous operation is selected by setting the TSYR.SYNCn bit (n = 0, 3, 4) to 1.
Note 2. When TGRC or TGRD is used as a buffer register, TCNT is not cleared because the buffer register setting has priority and compare match/input capture does not occur.

Table 23.5 CCLR[2:0] (MTU1 and MTU2)

Channel	Bit 7 Reserved*2	Bit 6 CCLR[1]	Bit 5 CCLR[0]	Description
MTU1, MTU2	0	0	0	TCNT clearing disabled
	0	0	1	TCNT cleared by TGRA compare match/input capture
	0	1	0	TCNT cleared by TGRB compare match/input capture
	0	1	1	TCNT cleared by counter clearing in another channel performing synchronous clearing/synchronous operation*1

Note 1. Synchronous operation is selected by setting the TSYR.SYNCn bit (n = 1, 2) to 1.
Note 2. Bit 7 is reserved in MTU1 and MTU2. This bit is read as 0. The write value should be 0.

Table 23.6 TPSC[2:0] (MTU0)

Channel	Bit 2 TPSC[2]	Bit 1 TPSC[1]	Bit 0 TPSC[0]	Description
MTU0	0	0	0	Internal clock: counts on PCLK/1
	0	0	1	Internal clock: counts on PCLK/4
	0	1	0	Internal clock: counts on PCLK/16
	0	1	1	Internal clock: counts on PCLK/64
	1	0	0	External clock: counts on MTCLKA pin input
	1	0	1	External clock: counts on MTCLKB pin input
	1	1	0	External clock: counts on MTCLKC pin input
	1	1	1	External clock: counts on MTCLKD pin input

Downloaded from Arrow.com.

Table 23.7 TPSC[2:0] (MTU1)

Channel	Bit 2 TPSC[2]	Bit 1 TPSC[1]	Bit 0 TPSC[0]	Description
MTU1	0	0	0	Internal clock: counts on PCLK/1
	0	0	1	Internal clock: counts on PCLK/4
	0	1	0	Internal clock: counts on PCLK/16
	0	1	1	Internal clock: counts on PCLK/64
	1	0	0	External clock: counts on MTCLKA pin input
	1	0	1	External clock: counts on MTCLKB pin input
	1	1	0	Internal clock: counts on PCLK/256
	1	1	1	Counts on MTU2.TCNT overflow/underflow

Note: This setting is ignored when MTU1 is in phase counting mode.

Table 23.8 TPSC[2:0] (MTU2)

Channel	Bit 2 TPSC[2]	Bit 1 TPSC[1]	Bit 0 TPSC[0]	Description
MTU2	0	0	0	Internal clock: counts on PCLK/1
	0	0	1	Internal clock: counts on PCLK/4
	0	1	0	Internal clock: counts on PCLK/16
	0	1	1	Internal clock: counts on PCLK/64
	1	0	0	External clock: counts on MTCLKA pin input
	1	0	1	External clock: counts on MTCLKB pin input
	1	1	0	External clock: counts on MTCLKC pin input
	1	1	1	Internal clock: counts on PCLK/1024

Note: This setting is ignored when MTU2 is in phase counting mode.

Table 23.9 TPSC[2:0] (MTU3 and MTU4)

Channel	Bit 2 TPSC[2]	Bit 1 TPSC[1]	Bit 0 TPSC[0]	Description
MTU3, MTU4	0	0	0	Internal clock: counts on PCLK/1
	0	0	1	Internal clock: counts on PCLK/4
	0	1	0	Internal clock: counts on PCLK/16
	0	1	1	Internal clock: counts on PCLK/64
	1	0	0	Internal clock: counts on PCLK/256
	1	0	1	Internal clock: counts on PCLK/1024
	1	1	0	External clock: counts on MTCLKA pin input
	1	1	1	External clock: counts on MTCLKB pin input

Downloaded from Arrow.com.

23.2.2 Timer Mode Register (TMDR)

Address(es): MTU0.TMDR 000D 0B01h, MTU1.TMDR 000D 0B81h, MTU2.TMDR 000D 0C01h, MTU3.TMDR 000D 0A02h, MTU4.TMDR 000D 0A03h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	BFE	BFB	BFA	MD[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	MD[3:0]	Mode Select	These bits specify the timer operating mode. Refer to Table 23.10 for details.	R/W
b4	BFA	Buffer Operation A	0: TGRA and TGRC operate normally 1: TGRA and TGRC used together for buffer operation	R/W
b5	BFB	Buffer Operation B	0: TGRB and TGRD operate normally 1: TGRB and TGRD used together for buffer operation	R/W
b6	BFE	Buffer Operation E	0: MTU0.TGRE and MTU0.TGRF operate normally 1: MTU0.TGRE and MTU0.TGRF used together for buffer operation	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

The TMDR register specifies the operating mode of each channel. The TMDR register values should be specified only while the TCNT operation is stopped.

Table 23.10 Operating Mode Setting by MD[3:0] Bits

Bit 3	Bit 2	Bit 1	Bit 0						
MD[3]	MD[2]	MD[1]	MD[0]	Description	MTU0	MTU1	MTU2	MTU3	MTU4
0	0	0	0	Normal mode	✓	✓	✓	✓	✓
0	0	0	1	Setting prohibited					
0	0	1	0	PWM mode 1	✓	✓	✓	✓	✓
0	0	1	1	PWM mode 2	✓	✓	✓		
0	1	0	0	Phase counting mode 1		✓	✓		
0	1	0	1	Phase counting mode 2		✓	✓		
0	1	1	0	Phase counting mode 3		✓	✓		
0	1	1	1	Phase counting mode 4		✓	✓		
1	0	0	0	Reset-synchronized PWM mode*1				✓	
1	0	0	1	Setting prohibited					
1	0	1	x	Setting prohibited					
1	1	0	0	Setting prohibited					
1	1	0	1	Complementary PWM mode 1 (transfer at crest)*1				✓	
1	1	1	0	Complementary PWM mode 2 (transfer at trough)*1				✓	
1	1	1	1	Complementary PWM mode 3 (transfer at crest and trough)*1				✓	

x: Don't care

Note: Only set the corresponding operating mode listed above for each channel.

Note 1. Reset-synchronized PWM mode and complementary PWM mode can only be set for MTU3. When MTU3 is set to reset-synchronized PWM mode or complementary PWM mode, the MTU4 settings become ineffective and conform to the MTU3 setting, respectively. MTU4 should be set to normal mode.

Downloaded from Arrow.com.

BFA Bit (Buffer Operation A)

This bit specifies normal operation for the TGRA register or buffered operation of the combination of registers TGRA and TGRC. When the TGRC register is used as a buffer register, the TGRC input capture/output compare does not take place in modes other than complementary PWM mode, but compare match with the TGRC register occurs in complementary PWM mode. If a compare match occurs on MTU4 in the Tb interval in complementary PWM mode, the MTU4.TIER.TGIEC bit should be set to 0.
When MTU3 or MTU4 is set to reset-synchronized PWM mode or complementary PWM mode, the buffer operation conforms to the MTU3 setting. Set the MTU4.TMDR.BFA bit to 0.
In MTU1 and MTU2, which have no TGRC register, this bit is reserved. It is read as 0. The write value should be 0.
Refer to Figure 23.40 for an illustration of the Tb interval in complementary PWM mode.

BFB Bit (Buffer Operation B)

This bit specifies normal operation for the TGRB register or buffered operation of the combination of registers TGRB and TGRD. When the TGRD register is used as a buffer register, the TGRD input capture/output compare does not take place in modes other than complementary PWM mode, but compare match with the TGRD register occurs in complementary PWM mode. If a compare match occurs in the Tb interval in complementary PWM mode, the MTU3.TIER.TGIED or MTU4.TIER.TGIED bit should be set to 0.
When MTU3 or MTU4 is set to reset-synchronized PWM mode or complementary PWM mode, the buffer operation conforms to the MTU3 setting. Set the MTU4.TMDR.BFB bit to 0.
In MTU1 and MTU2, which have no TGRD register, this bit is reserved. It is read as 0. The write value should be 0.
Refer to Figure 23.40 for an illustration of the Tb interval in complementary PWM mode.

BFE Bit (Buffer Operation E)

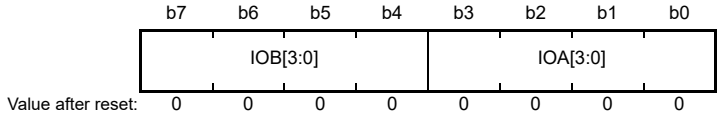
This bit specifies normal operation or buffered operation for registers MTU0.TGRE and MTU0.TGRF. Compare match with the TGRF register occurs even when the TGRF register is used as a buffer register.
In MTU1 to MTU4, this bit is reserved. It is read as 0. The write value should be 0.

Downloaded from Arrow.com.

23.2.3 Timer I/O Control Register (TIOR)

- MTU0.TIORH, MTU1.TIOR, MTU2.TIOR, MTU3.TIORH, MTU4.TIORH

Address(es): MTU0.TIORH 000D 0B02h, MTU1.TIOR 000D 0B82h, MTU2.TIOR 000D 0C02h, MTU3.TIORH 000D 0A04h, MTU4.TIORH 000D 0A06h

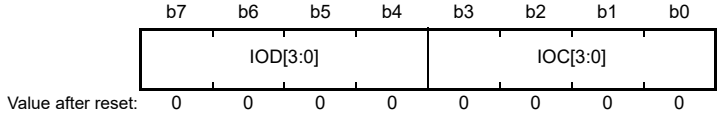


Bit	Symbol	Bit Name	Description	R/W
b3 to b0	IOA[3:0]	I/O Control A	Refer to the following tables.*1 MTU0.TIORH: Table 23.18 MTU1.TIOR: Table 23.20 MTU2.TIOR: Table 23.21 MTU3.TIORH: Table 23.22 MTU4.TIORH: Table 23.24	R/W
b7 to b4	IOB[3:0]	I/O Control B	Refer to the following tables.*1 MTU0.TIORH: Table 23.11 MTU1.TIOR: Table 23.12 MTU2.TIOR: Table 23.13 MTU3.TIORH: Table 23.14 MTU4.TIORH: Table 23.16	R/W

Note 1. If the IOm[3:0] (m = A, B) bits are changed to an “output prohibited” setting (0000b or 0100b) while output of the low or high level or toggling of the output in response to compare matches is in progress, the output becomes high-impedance.

- MTU0.TIORL, MTU3.TIORL, MTU4.TIORL

Address(es): MTU0.TIORL 000D 0B03h, MTU3.TIORL 000D 0A05h, MTU4.TIORL 000D 0A07h



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	IOC[3:0]	I/O Control C	Refer to the following tables.*1 MTU0.TIORL: Table 23.19 MTU3.TIORL: Table 23.23 MTU4.TIORL: Table 23.25	R/W
b7 to b4	IOD[3:0]	I/O Control D*2	Refer to the following tables.*1 MTU3.TIORL: Table 23.15 MTU4.TIORL: Table 23.17	R/W

Note 1. If the IOm[3:0] (m = C, D) bits are changed to an “output prohibited” setting (0000b or 0100b) while output of the low or high level or toggling of the output in response to compare matches is in progress, the output becomes high-impedance.
Note 2. In MTU0, which have no MTIOC0D pin, this bit is reserved. It is read as 0. The write value should be 0.

Downloaded from Arrow.com.

The MTU has a total of seven TIOR registers, one for MTU0, MTU1, and MTU2, two each for MTU3, and MTU4.
The TIOR register should be set when the TMDR register is set to select normal mode, PWM mode, or phase counting mode.
The initial output specified by the TIOR register is valid when the counter is stopped (the TSTR.CSTn bit is set to 0).
Note also that, in PWM mode 2, the output at the point at which the counter is set to 0 is specified.
When the TGRC or TGRD register is designated for buffer operation, this setting is invalid and the register operates as a buffer register.

Table 23.11 TIORH (MTU0)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOB[3]	**IOB[2]**	**IOB[1]**	**IOB[0]**	**MTU0.TGRB Function**	**MTIOC0B Pin Function**
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	0	0	0	Input capture register	Input capture at rising edge.
1	0	0	1		Input capture at falling edge.
1	0	1	x		Input capture at both edges.
1	1	x	x		Capture input source is count clock in MTU1. Input capture at MTU1.TCNT up-count/down-count.*1

x: Don't care

Note 1. When PCLK/1 is selected as the count clock for MTU1, MTU0 input capture is not generated. Do not select PCLK/1 as the count clock for MTU1.

Downloaded from Arrow.com.

Table 23.12 TIOR (MTU1)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOB[3]	**IOB[2]**	**IOB[1]**	**IOB[0]**	**MTU1.TGRB Function**	**MTIOC1B Pin Function**
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	0	0	0	Input capture register	Input capture at rising edge.
1	0	0	1		Input capture at falling edge.
1	0	1	x		Input capture at both edges.
1	1	x	x		Input capture at generation of MTU0.TGRC compare match/input capture.

x: Don't care

Downloaded from Arrow.com.

Table 23.13 TIOR (MTU2)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOB[3]	IOB[2]	IOB[1]	IOB[0]	MTU2.TGRB Function	MTIOC2B Pin Function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Table 23.14 TIORH (MTU3)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOB[3]	IOB[2]	IOB[1]	IOB[0]	MTU3.TGRB Function	MTIOC3B Pin Function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Downloaded from Arrow.com.

Table 23.15 TIORL (MTU3)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOD[3]	IOD[2]	IOD[1]	IOD[0]	MTU3.TGRD Function	MTIOC3D Pin Function
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register*1	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Note 1. When the MTU3.TMDR.BFB bit is set to 1 and the MTU3.TGRD register is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 23.16 TIORH (MTU4)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOB[3]	IOB[2]	IOB[1]	IOB[0]	MTU4.TGRB Function	MTIOC4B Pin Function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Downloaded from Arrow.com.

Table 23.17 TIORL (MTU4)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOD[3]	IOD[2]	IOD[1]	IOD[0]	MTU4.TGRD Function	MTIOC4D Pin Function
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register*1	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Note 1. When the MTU4.TMDR.BFB bit is set to 1 and the MTU4.TGRD register is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 23.18 TIORH (MTU0)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOA[3]	IOA[2]	IOA[1]	IOA[0]	MTU0.TGRA Function	MTIOC0A Pin Function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	0	0	0	Input capture register	Input capture at rising edge.
1	0	0	1		Input capture at falling edge.
1	0	1	x		Input capture at both edges.
1	1	x	x		Capture input source is count clock in MTU1. Input capture at MTU1.TCNT up-count/down-count.*1

x: Don't care

Note 1. When PCLK/1 is selected as the count clock for MTU1, MTU0 input capture is not generated. Do not select PCLK/1 as the count clock for MTU1.

Downloaded from Arrow.com.

Table 23.19 TIORL (MTU0)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOC[3]	IOC[2]	IOC[1]	IOC[0]	MTU0.TGRC Function	MTIOC0C Pin Function
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	0	0	0	Input capture register*1	Input capture at rising edge.
1	0	0	1		Input capture at falling edge.
1	0	1	x		Input capture at both edges.
1	1	x	x		Capture input source is count clock in MTU1. Input capture at MTU1.TCNT up-count/down-count.*2

x: Don't care

Note 1. When the MTU0.TMDR.BFA bit is set to 1 and the MTU0.TGRC register is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Note 2. When PCLK/1 is selected as the count clock for MTU1, MTU0 input capture is not generated. Do not select PCLK/1 as the count clock for MTU1.

Table 23.20 TIOR (MTU1)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOA[3]	IOA[2]	IOA[1]	IOA[0]	MTU1.TGRA Function	MTIOC1A Pin Function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	0	0	0	Input capture register	Input capture at rising edge.
1	0	0	1		Input capture at falling edge.
1	0	1	x		Input capture at both edges.
1	1	x	x		Input capture at generation of MTU0.TGRA compare match/input capture.

x: Don't care

Downloaded from Arrow.com.

Table 23.21 TIOR (MTU2)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOA[3]	IOA[2]	IOA[1]	IOA[0]	MTU2.TGRA Function	MTIOC2A Pin Function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Table 23.22 TIORH (MTU3)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOA[3]	IOA[2]	IOA[1]	IOA[0]	MTU3.TGRA Function	MTIOC3A Pin Function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Downloaded from Arrow.com.

Table 23.23 TIORL (MTU3)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOC[3]	IOC[2]	IOC[1]	IOC[0]	MTU3.TGRC Function	MTIOC3C Pin Function
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register*1	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Note 1. When the MTU3.TMDR.BFA bit is set to 1 and the MTU3.TGRC register is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 23.24 TIORH (MTU4)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOA[3]	IOA[2]	IOA[1]	IOA[0]	MTU4.TGRA Function	MTIOC4A Pin Function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Downloaded from Arrow.com.

Table 23.25 TIORL (MTU4)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOC[3]	**IOC[2]**	**IOC[1]**	**IOC[0]**	**MTU4.TGRC Function**	**MTIOC4C Pin Function**
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register*1	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Note 1. When the MTU4.TMDR.BFA bit is set to 1 and the MTU4.TGRC register is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Downloaded from Arrow.com.

23.2.4 Timer Interrupt Enable Register (TIER)

- MTU0.TIER, MTU3.TIER

Address(es): MTU0.TIER 000D 0B04h, MTU3.TIER 000D 0A08h

	b7	b6	b5	b4	b3	b2	b1	b0
	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
Value after reset:	0	0	0	0	0	0	0	0

- MTU1.TIER, MTU2.TIER

Address(es): MTU1.TIER 000D 0B84h, MTU2.TIER 000D 0C04h

	b7	b6	b5	b4	b3	b2	b1	b0
	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA
Value after reset:	0	0	0	0	0	0	0	0

- MTU4.TIER

Address(es): MTU4.TIER 000D 0A09h

	b7	b6	b5	b4	b3	b2	b1	b0
	TTGE	TTGE2	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TGIEA	TGR Interrupt Enable A	0: Interrupt requests (TGIA) disabled 1: Interrupt requests (TGIA) enabled	R/W
b1	TGIEB	TGR Interrupt Enable B	0: Interrupt requests (TGIB) disabled 1: Interrupt requests (TGIB) enabled	R/W
b2	TGIEC	TGR Interrupt Enable C	0: Interrupt requests (TGIC) disabled 1: Interrupt requests (TGIC) enabled	R/W
b3	TGIED	TGR Interrupt Enable D	0: Interrupt requests (TGID) disabled 1: Interrupt requests (TGID) enabled	R/W
b4	TCIEV	Overflow Interrupt Enable	0: Interrupt requests (TCIV) disabled 1: Interrupt requests (TCIV) enabled	R/W
b5	TCIEU	Underflow Interrupt Enable	0: Interrupt requests (TCIU) disabled 1: Interrupt requests (TCIU) enabled	R/W
b6	TTGE2	A/D Converter Start Request Enable 2	0: A/D converter start request generation by MTU4.TCNT underflow (trough) disabled 1: A/D converter start request generation by MTU4.TCNT underflow (trough) enabled	R/W
b7	TTGE	A/D Converter Start Request Enable	0: A/D converter start request generation disabled 1: A/D converter start request generation enabled	R/W

The MTU has a total of six TIER registers, two each for MTU0 and one each for MTU1 to MTU4.
The TIER register enables or disables interrupt requests in each channel.

Downloaded from Arrow.com.

TGIEA and TGIEB Bits (TGR Interrupt Enable A and B)

Each bit enables or disables interrupt requests (TGIm) (m = A, B).

TGIEC and TGIED Bits (TGR Interrupt Enable C and D)

Each bit enables or disables interrupt requests (TGIm) in MTU0, MTU3, and MTU4 (m = C, D).
In MTU1 and MTU2, these bits are reserved. They are read as 0. The write value should be 0.

TCIEV Bit (Overflow Interrupt Enable)

This bit enables or disables interrupt requests (TCIV).

TCIEU Bit (Underflow Interrupt Enable)

This bit enables or disables interrupt requests (TCIU) in MTU1 and MTU2.
In MTU0, MTU3, and MTU4, this bit is reserved. It is read as 0. The write value should be 0.

TTGE2 Bit (A/D Converter Start Request Enable 2)

This bit enables or disables generation of A/D converter start requests by MTU4.TCNT underflow (trough) in complementary PWM mode.
In MTU0 to MTU3, this bit is reserved. It is read as 0. The write value should be 0.

TTGE Bit (A/D Converter Start Request Enable)

This bit enables or disables generation of A/D converter start requests by the TGRA input capture/compare match.

- MTU0.TIER2

Address(es): MTU0.TIER2 000D 0B24h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	TGIEF	TGIEE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TGIEE	TGR Interrupt Enable E	0: Interrupt requests (TGIE) disabled 1: Interrupt requests (TGIE) enabled	R/W
b1	TGIEF	TGR Interrupt Enable F	0: Interrupt requests (TGIF) disabled 1: Interrupt requests (TGIF) enabled	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

TGIEE and TGIEF Bits (TGR Interrupt Enable E and F)

Each bit enables or disables interrupt requests by compare match between the MTU0.TCNT counter and the MTU0.TGRm register (m = E, F).

Downloaded from Arrow.com.

23.2.5 Timer Status Register (TSR)

Address(es): MTU0.TSR 000D 0B05h, MTU1.TSR 000D 0B85h, MTU2.TSR 000D 0C05h, MTU3.TSR 000D 0A2Ch, MTU4.TSR 000D 0A2Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	TCFD	—	—	—	—	—	—	—
Value after reset:	1	1	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	—	Reserved	These bits are read as undefined. The write value should be 1.	R/W
b6	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b7	TCFD	Count Direction Flag	0: TCNT counts down 1: TCNT counts up	R

The MTU has a total of five TSR registers, one each for MTU0 to MTU4.
The TSR register indicates the status of each channel.

TCFD Flag (Count Direction Flag)

Status flag that shows the direction in which the TCNT counter counts in MTU1 to MTU4.
In MTU0, this bit is reserved. It is read as 1. The write value should be 1.

Downloaded from Arrow.com.

23.2.6 Timer Buffer Operation Transfer Mode Register (TBTM)

- MTU0.TBTM

Address(es): MTU0.TBTM 000D 0B26h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	TTSE	TTSB	TTSA
Value after reset:	0	0	0	0	0	0	0	0

- MTU3.TBTM, MTU4.TBTM

Address(es): MTU3.TBTM 000D 0A38h, MTU4.TBTM 000D 0A39h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	TTSB	TTSA
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TTSA	Timing Select A	0: When compare match A occurs in each channel, data is transferred from TGRC to TGRA 1: When TCNT is cleared in each channel, data is transferred from TGRC to TGRA	R/W
b1	TTSB	Timing Select B	0: When compare match B occurs in each channel, data is transferred from TGRD to TGRB 1: When TCNT is cleared in each channel, data is transferred from TGRD to TGRB	R/W
b2	TTSE	Timing Select E	0: When compare match E occurs in MTU0, data is transferred from MTU0.TGRF to MTU0.TGRE 1: When MTU0.TCNT is cleared in MTU0, data is transferred from MTU0.TGRF to MTU0.TGRE	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The MTU has a total of three TBTM registers, one each for MTU0, MTU3, and MTU4.
The TBTM register specifies the timing for transferring data from the buffer register to the timer general register in PWM mode.

TTSA Bit (Timing Select A)

This bit specifies the timing for transferring data from the TGRC register to the TGRA register in each channel when they are used together for buffer operation. When a channel is not set to PWM mode, do not set the TTSA bit in the channel to 1.

TTSB Bit (Timing Select B)

This bit specifies the timing for transferring data from the TGRD register to the TGRB register in each channel when they are used together for buffer operation. When a channel is not set to PWM mode, do not set the TTSB bit in the channel to 1.

TTSE Bit (Timing Select E)

This bit specifies the timing for transferring data from the MTU0.TGRF register to the MTU0.TGRE register when they are used together for buffer operation. In MTU3 and MTU4, this bit is reserved and read as 0. The write value should be 0. When MTU0 is not set to PWM mode, do not set the TTSE bit to 1.

Downloaded from Arrow.com.

23.2.7 Timer Input Capture Control Register (TICCR)

Address(es): MTU1.TICCR 000D 0B90h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	I2BE	I2AE	I1BE	I1AE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	I1AE	Input Capture Enable	0: Does not include the MTIOC1A pin in the MTU2.TGRA input capture conditions 1: Includes the MTIOC1A pin in the MTU2.TGRA input capture conditions	R/W
b1	I1BE	Input Capture Enable	0: Does not include the MTIOC1B pin in the MTU2.TGRB input capture conditions 1: Includes the MTIOC1B pin in the MTU2.TGRB input capture conditions	R/W
b2	I2AE	Input Capture Enable	0: Does not include the MTIOC2A pin in the MTU1.TGRA input capture conditions 1: Includes the MTIOC2A pin in the MTU1.TGRA input capture conditions	R/W
b3	I2BE	Input Capture Enable	0: Does not include the MTIOC2B pin in the MTU1.TGRB input capture conditions 1: Includes the MTIOC2B pin in the MTU1.TGRB input capture conditions	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The MTU has one TICCR register for MTU1.

The TICCR register specifies input capture conditions when counters MTU1.TCNT and MTU2.TCNT are cascaded.

Downloaded from Arrow.com.

23.2.8 Timer A/D Converter Start Request Control Register (TADCR)

Address(es): MTU4.TADCR 000D 0A40h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	BF[1:0]		—	—	—	—	—	—	UT4AE	DT4AE	UT4BE	DT4BE	ITA3AE	ITA4VE	ITB3AE	ITB4VE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ITB4VE	TCIV4 Interrupt Skipping Link Enable*1, *2, *3	0: TCIV4 interrupt skipping is not linked 1: TCIV4 interrupt skipping is linked	R/W
b1	ITB3AE	TGIA3 Interrupt Skipping Link Enable*1, *2, *3	0: TGIA3 interrupt skipping is not linked 1: TGIA3 interrupt skipping is linked	R/W
b2	ITA4VE	TCIV4 Interrupt Skipping Link Enable*1, *2, *3	0: TCIV4 interrupt skipping is not linked 1: TCIV4 interrupt skipping is linked	R/W
b3	ITA3AE	TGIA3 Interrupt Skipping Link Enable*1, *2, *3	0: TGIA3 interrupt skipping is not linked 1: TGIA3 interrupt skipping is linked	R/W
b4	DT4BE	Down-Count TRG4BN Enable*3	0: A/D converter start requests (TRG4BN) disabled during MTU4.TCNT down-count operation 1: A/D converter start requests (TRG4BN) enabled during MTU4.TCNT down-count operation	R/W
b5	UT4BE	Up-Count TRG4BN Enable	0: A/D converter start requests (TRG4BN) disabled during MTU4.TCNT up-count operation 1: A/D converter start requests (TRG4BN) enabled during MTU4.TCNT up-count operation	R/W
b6	DT4AE	Down-Count TRG4AN Enable*3	0: A/D converter start requests (TRG4AN) disabled during MTU4.TCNT down-count operation 1: A/D converter start requests (TRG4AN) enabled during MTU4.TCNT down-count operation	R/W
b7	UT4AE	Up-Count TRG4AN Enable	0: A/D converter start requests (TRG4AN) disabled during MTU4.TCNT up-count operation 1: A/D converter start requests (TRG4AN) enabled during MTU4.TCNT up-count operation	R/W
b13 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15, b14	BF[1:0]	MTU4.TADCOBRA/TADCOBRB Transfer Timing Select	Refer to Table 23.26 for details.	R/W

Note: The TADCR register must not be accessed in 8-bit units; it should be accessed in 16-bit units.
Note 1. When interrupt skipping is disabled (the TITCR.T3AEN and T4VEN bits are set to 0 or the interrupt skipping count setting bits (T3ACOR[2:0] and T4VCOR[2:0]) in the TITCR register are set to 000b), do not link A/D converter start requests with interrupt skipping operation (set the TADCR.ITA3AE, ITA4VE, ITB3AE, and ITB4VE bits to 0).
Note 2. If link with interrupt skipping is enabled while interrupt skipping is disabled, A/D converter start requests will not be issued.
Note 3. Set b6 and b4 to b0 to 0 when complementary PWM mode is not selected.

The TADCR register enables or disables A/D converter start requests and specifies whether to link A/D converter start requests with interrupt skipping operation.

Downloaded from Arrow.com.

Table 23.26 Setting of Transfer Timing by TADCR.BF[1:0] Bits

Bit 15	Bit 14	Description			
BF[1]	BF[0]	In Complementary PWM Mode	In Reset-Synchronized PWM Mode	In PWM Mode 1	In Normal Mode
0	0	Data is not transferred from the cycle set buffer register (MTU4.TADCOBRA, MTU4.TADCOBRB) to the cycle set register (MTU4.TADCORA, MTU4.TADCORB).	Data is not transferred from the cycle set buffer register (MTU4.TADCOBRA, MTU4.TADCOBRB) to the cycle set register (MTU4.TADCORA, MTU4.TADCORB).	Data is not transferred from the cycle set buffer register (MTU4.TADCOBRA, MTU4.TADCOBRB) to the cycle set register (MTU4.TADCORA, MTU4.TADCORB).	Data is not transferred from the cycle set buffer register (MTU4.TADCOBRA, MTU4.TADCOBRB) to the cycle set register (MTU4.TADCORA, MTU4.TADCORB).
0	1	Data is transferred from the cycle set buffer register (MTU4.TADCOBRA, MTU4.TADCOBRB) to the cycle set register (MTU4.TADCORA, MTU4.TADCORB) at the crest of the MTU4.TCNT.	Data is transferred from the cycle set buffer register (MTU4.TADCOBRA, MTU4.TADCOBRB) to the cycle set register (MTU4.TADCORA, MTU4.TADCORB) when a compare match occurs between MTU3.TCNT and MTU3.TGRA.	Data is transferred from the cycle set buffer register (MTU4.TADCOBRA, MTU4.TADCOBRB) to the cycle set register (MTU4.TADCORA, MTU4.TADCORB) when a compare match occurs between MTU4.TCNT and MTU4.TGRA.	Data is transferred from the cycle set buffer register (MTU4.TADCOBRA, MTU4.TADCOBRB) to the cycle set register (MTU4.TADCORA, MTU4.TADCORB) when a compare match occurs between MTU4.TCNT and MTU4.TGRA.
1	0	Data is transferred from the cycle set buffer register (MTU4.TADCOBRA, MTU4.TADCOBRB) to the cycle set register (MTU4.TADCORA, MTU4.TADCORB) at the trough of the MTU4.TCNT.	Setting prohibited	Setting prohibited	Setting prohibited
1	1	Data is transferred from the cycle set buffer register (MTU4.TADCOBRA, MTU4.TADCOBRB) to the cycle set register (MTU4.TADCORA, MTU4.TADCORB) at the crest and trough of the MTU4.TCNT.	Setting prohibited	Setting prohibited	Setting prohibited

23.2.9 Timer A/D Converter Start Request Cycle Set Registers A and B (TADCORA and TADCORB)

Address(es): MTU4.TADCORA 000D 0A44h, MTU4.TADCORB 000D 0A46h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Note: MTU4.TADCORA and MTU4.TADCORB must not be accessed in 8-bit units; they should be accessed in 16-bit units.

The TADCORA and TADCORB registers specify the A/D converter start request cycle. When the MTU4.TCNT count reaches the value in TADCORA or TADCORB, a corresponding A/D converter start request will be issued.

Downloaded from Arrow.com.

23.2.10 Timer A/D Converter Start Request Cycle Set Buffer Registers A and B (TADCOBRA and TADCOBRB)

Address(es): MTU4.TADCOBRA 000D 0A48h, MTU4.TADCOBRB 000D 0A4Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Note: MTU4.TADCOBRA and MTU4.TADCOBRB must not be accessed in 8-bit units; they should be accessed in 16-bit units.

The TADCOBRA and TADCOBRB registers function as buffer registers for registers TADCORA and TADCORB, respectively. These registers specify the A/D converter start request cycle. When the crest or trough of the MTU4.TCNT count is reached, these register values are transferred to registers TADCORA and TADCORB, respectively.

23.2.11 Timer Counter (TCNT)

Address(es): MTU0.TCNT 000D 0B06h, MTU1.TCNT 000D 0B86h, MTU2.TCNT 000D 0C06h, MTU3.TCNT 000D 0A10h, MTU4.TCNT 000D 0A12h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Note: The TCNT counters must not be accessed in 8-bit units; they should be accessed in 16-bit units.

The MTU has a total of five TCNT counters, one each for MTU0 to MTU4.

Downloaded from Arrow.com.

23.2.12 Timer General Register (TGR)

Address(es): MTU0.TGRA 000D 0B08h, MTU0.TGRB 000D 0B0Ah, MTU0.TGRC 000D 0B0Ch, MTU0.TGRD 000D 0B0Eh, MTU0.TGRE 000D 0B20h, MTU0.TGRF 000D 0B22h, MTU1.TGRA 000D 0B88h, MTU1.TGRB 000D 0B8Ah, MTU2.TGRA 000D 0C08h, MTU2.TGRB 000D 0C0Ah, MTU3.TGRA 000D 0A18h, MTU3.TGRB 000D 0A1Ah, MTU3.TGRC 000D 0A24h, MTU3.TGRD 000D 0A26h, MTU4.TGRA 000D 0A1Ch, MTU4.TGRB 000D 0A1Eh, MTU4.TGRC 000D 0A28h, MTU4.TGRD 000D 0A2Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Note: The TGR registers must not be accessed in 8-bit units; they should be accessed in 16-bit units.

The MTU has a total of 18 TGR registers, six for MTU0, two each for MTU1 and MTU2, and four each for MTU3 and MTU4.

Registers TGRA, TGRB, TGRC, and TGRD function as either output compare or input capture registers. Registers TGRC and TGRD for MTU0, MTU3, and MTU4 can also be designated for operation as buffer registers. TGR buffer register combinations are TGRA and TGRC, and TGRB and TGRD.

Registers MTU0.TGRE and MTU0.TGRF function as compare registers. When the MTU0.TCNT count matches the MTU0.TGRE register value, an A/D converter start request can be issued. The TGRF register can also be designated for operation as a buffer register. TGR buffer register combination is TGRE and TGRF.

Downloaded from Arrow.com.

23.2.13 Timer Start Register (TSTR)

Address(es): MTU.TSTR 000D 0A80h

	b7	b6	b5	b4	b3	b2	b1	b0
	CST4	CST3	—	—	—	CST2	CST1	CST0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CST0	Counter Start 0	0: MTU0.TCNT performs count stop 1: MTU0.TCNT performs count operation	R/W
b1	CST1	Counter Start 1	0: MTU1.TCNT performs count stop 1: MTU1.TCNT performs count operation	R/W
b2	CST2	Counter Start 2	0: MTU2.TCNT performs count stop 1: MTU2.TCNT performs count operation	R/W
b5 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	CST3	Counter Start 3	0: MTU3.TCNT performs count stop 1: MTU3.TCNT performs count operation	R/W
b7	CST4	Counter Start 4	0: MTU4.TCNT performs count stop 1: MTU4.TCNT performs count operation	R/W

The TSTR register starts or stops the TCNT operation in MTU0 to MTU4.
Before setting the operating mode in the TMDR register or setting the TCNT count clock in the TCR register, be sure to stop the TCNT counter.

CSTn Bits (Counter Start n) (n = 0 to 4)

Each bit starts or stops the TCNT counter in the corresponding channel.
If 0 is written to the CSTn bit during operation with the MTIOC pin designated for output, the counter stops but the output compare signal level from the MTIOC pin is retained. If the TIOR register is written to while the CSTn bit is 0, the pin output level will be changed to the specified initial output value.

Downloaded from Arrow.com.

23.2.14 Timer Synchronous Register (TSYR)

Address(es): MTU.TSYR 000D 0A81h

	b7	b6	b5	b4	b3	b2	b1	b0
	SYNC4	SYNC3	—	—	—	SYNC2	SYNC1	SYNC0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SYNC0	Timer Synchronous Operation 0	0: MTU0.TCNT operates independently (TCNT setting/clearing is not related to other channels) 1: MTU0.TCNT performs synchronous operation. TCNT synchronous setting/synchronous clearing is enabled	R/W
b1	SYNC1	Timer Synchronous Operation 1	0: MTU1.TCNT operates independently (TCNT setting/clearing is not related to other channels) 1: MTU1.TCNT performs synchronous operation. TCNT synchronous setting/synchronous clearing is enabled	R/W
b2	SYNC2	Timer Synchronous Operation 2	0: MTU2.TCNT operates independently (TCNT setting/clearing is not related to other channels) 1: MTU2.TCNT performs synchronous operation. TCNT synchronous setting/synchronous clearing is enabled	R/W
b5 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	SYNC3	Timer Synchronous Operation 3	0: MTU3.TCNT operates independently (TCNT setting/clearing is not related to other channels). 1: MTU3.TCNT performs synchronous operation. TCNT synchronous setting/synchronous clearing is enabled.	R/W
b7	SYNC4	Timer Synchronous Operation 4	0: MTU4.TCNT operates independently (TCNT setting/clearing is not related to other channels). 1: MTU4.TCNT performs synchronous operation. TCNT synchronous setting/synchronous clearing is enabled.	R/W

The TSYR register selects independent operation or synchronous operation of the TCNT counter in MTU0 to MTU4. A channel performs synchronous operation when the corresponding bit in the TSYR register is set to 1.

SYNCn Bits (Timer Synchronous n Operation) (n = 0 to 4)

Each bit selects whether operation is independent of or synchronized with other channels.

When synchronous operation is selected, the TCNT synchronous setting of multiple channels and synchronous clearing by counter clearing on another channel are possible.

To set synchronous operation, the SYNCn bits for at least two channels must be set to 1. To set synchronous clearing, in addition to the SYNCn bit, the TCNT clearing source must also be set the TCR.CCLR[2:0] bits.

Downloaded from Arrow.com.

23.2.15 Timer Read/Write Enable Register (TRWER)

Address(es): MTU.TRWER 000D 0A84h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	RWE
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	RWE	Read/Write Enable	0: Read/write access to the registers is disabled 1: Read/write access to the registers is enabled	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The TRWER register enables or disables access to the registers and counters that have write-protection capability against accidental modification in MTU3 and MTU4.

RWE Bit (Read/Write Enable)

This bit enables or disables access to the registers that have write-protection capability against accidental modification.
[Clearing condition]

- When 0 is written to the RWE bit after reading the RWE bit = 1

- Registers and Counters having Write-Protection Capability against Accidental Modification

22 registers: MTUn.TCR, MTUn.TMDR, MTUn.TIORH, MTUn.TIORL, MTUn.TIER, MTUn.TGRA, MTUn.TGRB, MTU.TOER, MTU.TOCR1, MTU.TOCR2, MTU.TGCR, MTU.TCDR, MTU.TDDR, and MTUn.TCNT (n = 3, 4)

Downloaded from Arrow.com.

23.2.16 Timer Output Master Enable Register (TOER)

Address(es): MTU.TOER 000D 0A0Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	OE4D	OE4C	OE3D	OE4B	OE4A	OE3B
Value after reset:	1	1	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	OE3B	Master Enable MTIOC3B	0: MTU output is disabled*1 1: MTU output is enabled	R/W
b1	OE4A	Master Enable MTIOC4A	0: MTU output is disabled*1 1: MTU output is enabled	R/W
b2	OE4B	Master Enable MTIOC4B	0: MTU output is disabled*1 1: MTU output is enabled	R/W
b3	OE3D	Master Enable MTIOC3D	0: MTU output is disabled*1 1: MTU output is enabled	R/W
b4	OE4C	Master Enable MTIOC4C	0: MTU output is disabled*1 1: MTU output is enabled	R/W
b5	OE4D	Master Enable MTIOC4D	0: MTU output is disabled*1 1: MTU output is enabled	R/W
b7, b6	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

Note 1. To output a non-active level from each pin when MTU output is disabled, make necessary settings for non-active level output from general I/O ports in the data direction registers (PDR), port output data registers (PODR), and port mode register (PMR) in advance. For details, refer to the I/O Ports section.

The TOER register enables or disables output settings for output pins MTIOC4D, MTIOC4C, MTIOC3D, MTIOC4B, MTIOC4A, and MTIOC3B.

These pins do not output correctly if the bits in the TOER register have not been set. In MTU3 and MTU4, set the TOER register prior to setting the TIOR register.

Set the TOER register after setting the TSTR.CST3 and CST4 bits to 0 (refer to Figure 23.35 and Figure 23.38).

Downloaded from Arrow.com.

23.2.17 Timer Output Control Register 1 (TOCR1)

Address(es): MTU.TOCR1 000D 0A0Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	PSYE	—	—	TOCL	TOCS	OLSN	OLSP
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	OLSP	Output Level Select P*2,*3	Refer to Table 23.27.	R/W
b1	OLSN	Output Level Select N*2,*3	Refer to Table 23.28.	R/W
b2	TOCS	TOC Select	0: TOCR1 setting is selected 1: TOCR2 setting is selected	R/W
b3	TOCL	TOC Register Write Protection*1	0: Write access to the TOCS, OLSN, and OLSP bits is enabled 1: Write access to the TOCS, OLSN, and OLSP bits is disabled	R/W*4
b5, b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	PSYE	PWM Synchronous Output Enable	0: Toggle output is disabled 1: Toggle output is enabled	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note 1. Setting the TOCR1.TOCL bit to 1 prevents accidental modification when the CPU goes out of control.
Note 2. Setting the TOCR1.TOCS bit to 0 makes this bit setting valid.
Note 3. If dead-time is not generated, the negative-phase output is always the exact inverse of the positive-phase output. In this case, only the OLSP bit is valid.
Note 4. This bit can be set to 1 only once after a power-on reset. After 1 is written, 0 cannot be written to the bit.

The TOCR1 register enables or disables PWM-synchronized toggle output in complementary PWM mode and reset-synchronized PWM mode, and control inversion of PWM output level.

OLSP Bit (Output Level Select P)

This bit selects the positive-phase output level in reset-synchronized PWM mode and complementary PWM mode.

OLSN Bit (Output Level Select N)

This bit selects the negative-phase output level in reset-synchronized PWM mode and complementary PWM mode.

TOCS Bit (TOC Select)

This bit selects either the TOCR1 or TOCR2 register setting to be used for the output level in complementary PWM mode and reset-synchronized PWM mode.

TOCL Bit (TOC Register Write Protection)

This bit enables or disables write access to the TOCS, OLSN, and OLSP bits in the TOCR1 register.

PSYE Bit (PWM Synchronous Output Enable)

This bit enables or disables toggle output synchronized with the PWM cycle.

Downloaded from Arrow.com.

Table 23.27 Output Level Select Function

Bit 0	Function			
			Compare Match Output	
OLSP	Initial Output	Active Level	Up-Counting	Down-Counting
0	High	Low	Low	High
1	Low	High	High	Low

Table 23.28 Output Level Select Function

Bit 1	Function			
			Compare Match Output	
OLSN	Initial Output	Active Level	Up-Counting	Down-Counting
0	High	Low	High	Low
1	Low	High	Low	High

Note: The initial output value of negative-phase waveform changes to an active level after the dead time has passed since counting starts.

Figure 23.2 shows an example of output in complementary PWM mode (one phase) when OLSN = 1 and OLSP = 1.

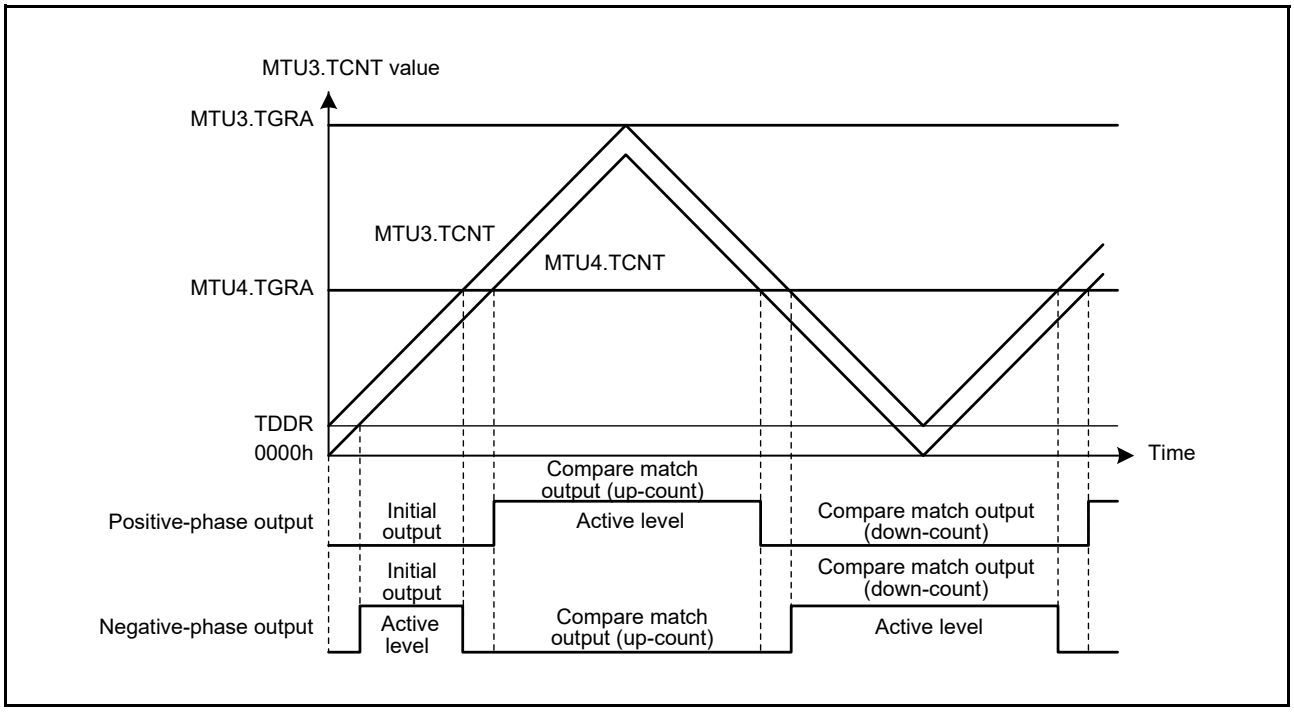


Figure 23.2 Example of Output in Complementary PWM Mode

Downloaded from Arrow.com.

23.2.18 Timer Output Control Register 2 (TOCR2)

Address(es): MTU.TOCR2 000D 0A0Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	BF[1:0]		OLS3N	OLS3P	OLS2N	OLS2P	OLS1N	OLS1P
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	OLS1P	Output Level Select 1P*1, *2	This bit selects the output level on MTIOC3B in reset-synchronized PWM mode and complementary PWM mode. Refer to Table 23.29.	R/W
b1	OLS1N	Output Level Select 1N*1, *2	This bit selects the output level on MTIOC3D in reset-synchronized PWM mode and complementary PWM mode. Refer to Table 23.30.	R/W
b2	OLS2P	Output Level Select 2P*1, *2	This bit selects the output level on MTIOC4A in reset-synchronized PWM mode and complementary PWM mode. Refer to Table 23.31.	R/W
b3	OLS2N	Output Level Select 2N*1, *2	This bit selects the output level on MTIOC4C in reset-synchronized PWM mode and complementary PWM mode. Refer to Table 23.32.	R/W
b4	OLS3P	Output Level Select 3P*1, *2	This bit selects the output level on MTIOC4B in reset-synchronized PWM mode and complementary PWM mode. Refer to Table 23.33.	R/W
b5	OLS3N	Output Level Select 3N*1, *2	This bit selects the output level on MTIOC4D in reset-synchronized PWM mode and complementary PWM mode. Refer to Table 23.34.	R/W
b7, b6	BF[1:0]	TOLBR Buffer Transfer Timing Select	These bits select the timing for transferring data from TOLBR to TOCR2. Refer to Table 23.35 for details.	R/W

Note 1. Setting the TOCR1.TOCS bit to 1 makes this bit setting valid.
Note 2. If dead-time is not generated, the negative-phase output is always the exact inverse of the positive-phase output. In these cases, only the OLSiP bits are valid (i = 1 to 3).

The TOCR2 register controls inversion of PWM output level in complementary PWM mode and reset-synchronized PWM mode.

Table 23.29 MTIOC3B Output Level Select Function

Bit 0	Function			
			Compare Match Output	
OLS1P	Initial Output	Active Level	Up-Counting	Down-Counting
0	High	Low	Low	High
1	Low	High	High	Low

Table 23.30 MTIOC3D Output Level Select Function

Bit 1	Function			
			Compare Match Output	
OLS1N	Initial Output	Active Level	Up-Counting	Down-Counting
0	High	Low	High	Low
1	Low	High	Low	High

Note: The initial output value of negative-phase waveform changes to an active level after the dead time has passed since counting starts.

Downloaded from Arrow.com.

Table 23.31 MTIOC4A Output Level Select Function

Bit 2	Function			
			Compare Match Output	
OLS2P	Initial Output	Active Level	Up-Counting	Down-Counting
0	High	Low	Low	High
1	Low	High	High	Low

Table 23.32 MTIOC4C Output Level Select Function

Bit 3	Function			
			Compare Match Output	
OLS2N	Initial Output	Active Level	Up-Counting	Down-Counting
0	High	Low	High	Low
1	Low	High	Low	High

Note: The initial output value of negative-phase waveform changes to an active level after the dead time has passed since counting starts.

Table 23.33 MTIOC4B Output Level Select Function

Bit 4	Function			
			Compare Match Output	
OLS3P	Initial Output	Active Level	Up-Counting	Down-Counting
0	High	Low	Low	High
1	Low	High	High	Low

Table 23.34 MTIOC4D Output Level Select Function

Bit 5	Function			
			Compare Match Output	
OLS3N	Initial Output	Active Level	Up-Counting	Down-Counting
0	High	Low	High	Low
1	Low	High	Low	High

Note: The initial output value of negative-phase waveform changes to an active level after the dead time has passed since counting starts.

Table 23.35 Setting of TOCR2.BF[1:0] Bits

Bit 7	Bit 6	Description	
BF[1]	BF[0]	Complementary PWM Mode	Reset-Synchronized PWM Mode
0	0	Does not transfer data from TOLBR to TOCR2.	Does not transfer data from TOLBR to TOCR2.
0	1	Transfers data from TOLBR to TOCR2 at the crest of the MTU4.TCNT count.	Transfers data from TOLBR to TOCR2 when MTU4.TCNT or MTU3.TCNT is cleared.
1	0	Transfers data from TOLBR to TOCR2 at the trough of the MTU4.TCNT count.	Setting prohibited
1	1	Transfers data from TOLBR to TOCR2 at the crest and trough of the MTU4.TCNT count.	Setting prohibited

Downloaded from Arrow.com.

23.2.19 Timer Output Level Buffer Register (TOLBR)

Address(es): MTU.TOLBR 000D 0A36h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	OLS3N	OLS3P	OLS2N	OLS2P	OLS1N	OLS1P
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	OLS1P	Output Level Select 1P	Specify the buffer value to be transferred to the OLS1P bit in TOCR2.	R/W
b1	OLS1N	Output Level Select 1N	Specify the buffer value to be transferred to the OLS1N bit in TOCR2.	R/W
b2	OLS2P	Output Level Select 2P	Specify the buffer value to be transferred to the OLS2P bit in TOCR2.	R/W
b3	OLS2N	Output Level Select 2N	Specify the buffer value to be transferred to the OLS2N bit in TOCR2.	R/W
b4	OLS3P	Output Level Select 3P	Specify the buffer value to be transferred to the OLS3P bit in TOCR2.	R/W
b5	OLS3N	Output Level Select 3N	Specify the buffer value to be transferred to the OLS3N bit in TOCR2.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The TOLBR register functions as a buffer register for the TOCR2 register and specify the PWM output level in complementary PWM mode and reset-synchronized PWM mode.

Figure 23.3 shows an example of the PWM output level setting procedure in buffer operation.

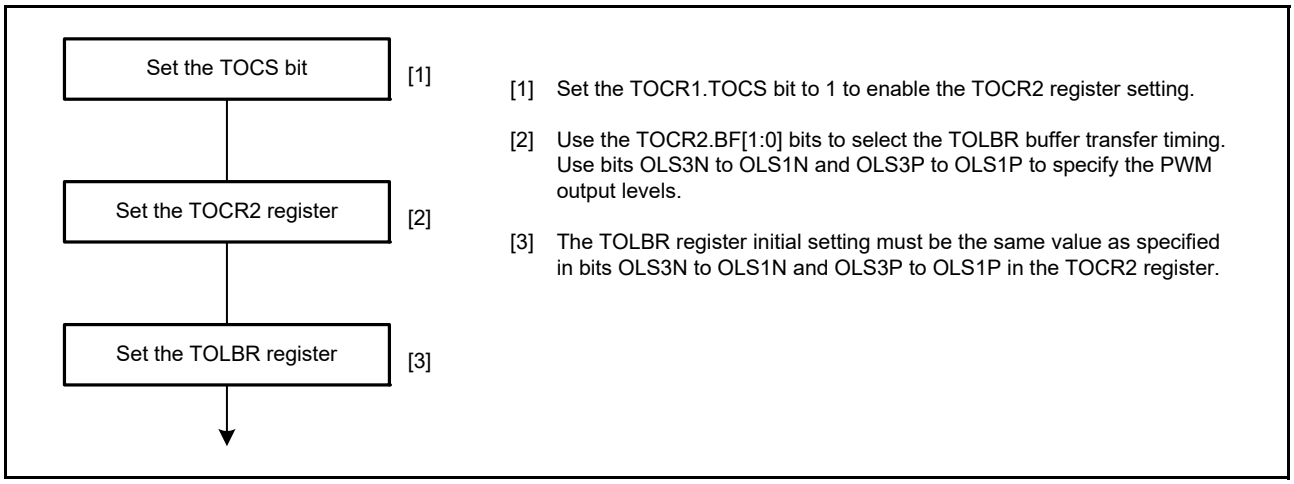


Figure 23.3 Example of PWM Output Level Setting Procedure in Buffer Operation

Downloaded from Arrow.com.

23.2.20 Timer Gate Control Register (TGCR)

Address(es): MTU.TGCR 000D 0A0Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	BDC	N	P	FB	WF	VF	UF
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	UF	Output Phase Switch	These bits turn on or off the positive-phase/negative-phase output. The setting of these bits is valid only when the TGCR.FB bit is set to 1. In this case, the setting of b0 to b2 is used instead of the external input. Refer to Table 23.36.	R/W
b1	VF			R/W
b2	WF			R/W
b3	FB	External Feedback Signal Enable	0: Output is switched by external input (input sources are TGRA, TGRB, and TGRC input capture signals in MTU0) 1: Output is switched by software (TGCR's UF, VF, and WF settings)	R/W
b4	P	Positive-Phase Output (P) Control	0: Level output 1: Reset-synchronized PWM or complementary PWM output	R/W
b5	N	Negative-Phase Output (N) Control	0: Level output 1: Reset-synchronized PWM or complementary PWM output	R/W
b6	BDC	Brushless DC Motor	0: Ordinary output 1: Functions of this register are made effective	R/W
b7	—	Reserved	This bit is read as 1. The write value should be 1.	R/W

The TGCR register controls the output waveform necessary for brushless DC motor control in reset-synchronized PWM mode and complementary PWM mode. These register settings are ineffective for anything other than complementary PWM mode and reset-synchronized PWM mode.

UF, VF, and WF Bits (Output Phase Switch)

The setting of these bits is valid only when the TGCR.FB bit is set to 1. In this case, the setting of b0 to b2 is used instead of the external input. Refer to Table 23.36.

FB Bit (External Feedback Signal Enable)

This bit selects whether the positive-/negative-phase output is switched automatically with the TGRA, TGRB, and TGRC input capture signals in MTU0 or by writing 0 or 1 to bits 2 to 0 in TGCR.

P Bit (Positive-Phase Output (P) Control)

This bit selects the level output or the reset-synchronized PWM/complementary PWM output for the positive-phase output pins (MTIOC3B, MTIOC4A, and MTIOC4B pins).

N Bit (Negative-Phase Output (N) Control)

This bit selects the level output or the reset-synchronized PWM/complementary PWM output for the negative-phase output pins (MTIOC3D, MTIOC4C, and MTIOC4D pins).

BDC Bit (Brushless DC Motor)

This bit selects whether to make the functions of the TGCR register effective or ineffective.

Downloaded from Arrow.com.

Table 23.36 Output Level Select Function

Bit 2	Bit 1	Bit 0	Function					
			MTIOC3B	MTIOC4A	MTIOC4B	MTIOC3D	MTIOC4C	MTIOC4D
WF	VF	UF	U Phase	V Phase	W Phase	U Phase	V Phase	W Phase
0	0	0	OFF	OFF	OFF	OFF	OFF	OFF
0	0	1	ON	OFF	OFF	OFF	OFF	ON
0	1	0	OFF	ON	OFF	ON	OFF	OFF
0	1	1	OFF	ON	OFF	OFF	OFF	ON
1	0	0	OFF	OFF	ON	OFF	ON	OFF
1	0	1	ON	OFF	OFF	OFF	ON	OFF
1	1	0	OFF	OFF	ON	ON	OFF	OFF
1	1	1	OFF	OFF	OFF	OFF	OFF	OFF

23.2.21 Timer Subcounter (TCNTS)

Address(es): MTU.TCNTS 000D 0A20h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Note: The TCNTS counters must not be accessed in 8-bit units; they should be accessed in 16-bit units.

The TCNTS counter is read-only counters that are used only in complementary PWM mode.

23.2.22 Timer Dead Time Data Register (TDDR)

Address(es): MTU.TDDR 000D 0A16h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Note: The TDDR registers must not be accessed in 8-bit units; they should be accessed in 16-bit units.

The TDDR register specifies the MTU3.TCNT and MTU4.TCNT counter offset value in complementary PWM mode. In complementary PWM mode, when the MTU3.TCNT and MTU4.TCNT counters are cleared and then restarted, the TDDR register value is loaded into the MTU3.TCNT counter and the count operation starts.

Downloaded from Arrow.com.

23.2.23 Timer Cycle Data Register (TCDR)

Address(es): MTU.TCDR 000D 0A14h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Note: The TCDR registers must not be accessed in 8-bit units; they should be accessed in 16-bit units.

The TCDR register specifies the count value to switch the count direction of the TCNTS counter. These registers are used only in complementary PWM mode. Set half the PWM cycle as the TCDR register value. The TCDR register is constantly compared with the TCNTS counter in complementary PWM mode, and when a match occurs, the TCNTS counter switches direction (down-count to up-count).

23.2.24 Timer Cycle Buffer Register (TCBR)

Address(es): MTU.TCBR 000D 0A22h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

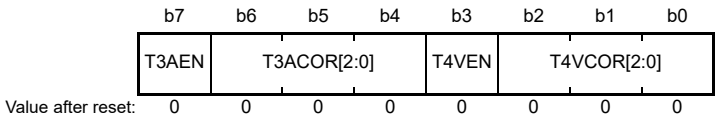
Note: The TCBR registers must not be accessed in 8-bit units; they should be accessed in 16-bit units.

The TCBR register functions as a buffer register for the TCDR register, and specifies the count value to switch the count direction of the TCNTS counter. This register is used only in complementary PWM mode. The TCBR register value is transferred to the TCDR register with the transfer timing set in the TMDR register.

Downloaded from Arrow.com.

23.2.25 Timer Interrupt Skipping Set Register (TITCR)

Address(es): MTU.TITCR 000D 0A30h



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	T4VCOR[2:0]	TCIV4 Interrupt Skipping Count Setting	These bits specify the TCIV4 interrupt skipping count within the range from 0 to 7.*1 For details, refer to Table 23.37.	R/W
b3	T4VEN	T4VEN	0: TCIV4 interrupt skipping disabled 1: TCIV4 interrupt skipping enabled	R/W
b6 to b4	T3ACOR[2:0]	TGIA3 Interrupt Skipping Count Setting	These bits specify the TGIA3 interrupt skipping count within the range from 0 to 7.*1 For details, refer to Table 23.38.	R/W
b7	T3AEN	T3AEN	0: TGIA3 interrupt skipping disabled 1: TGIA3 interrupt skipping enabled	R/W

Note 1. When 0 is specified for the interrupt skipping count, no interrupt skipping will be performed.
Before changing the interrupt skipping count, be sure to set the TITCR.T3AEN and TITCR.T4VEN bits to 0 to clear the TITCNT counter.

Table 23.37 Setting of Interrupt Skipping Count by T4VCOR[2:0] Bits

Bit 2	Bit 1	Bit 0	
T4VCOR[2]	T4VCOR[1]	T4VCOR[0]	Description
0	0	0	Does not perform TCIV4 interrupt skipping.
0	0	1	Sets the TCIV4 interrupt skipping count to 1.
0	1	0	Sets the TCIV4 interrupt skipping count to 2.
0	1	1	Sets the TCIV4 interrupt skipping count to 3.
1	0	0	Sets the TCIV4 interrupt skipping count to 4.
1	0	1	Sets the TCIV4 interrupt skipping count to 5.
1	1	0	Sets the TCIV4 interrupt skipping count to 6.
1	1	1	Sets the TCIV4 interrupt skipping count to 7.

Table 23.38 Setting of Interrupt Skipping Count by T3ACOR[2:0] Bits

Bit 6	Bit 5	Bit 4	
T3ACOR[2]	T3ACOR[1]	T3ACOR[0]	Description
0	0	0	Does not perform TGIA3 interrupt skipping.
0	0	1	Sets the TGIA3 interrupt skipping count to 1.
0	1	0	Sets the TGIA3 interrupt skipping count to 2.
0	1	1	Sets the TGIA3 interrupt skipping count to 3.
1	0	0	Sets the TGIA3 interrupt skipping count to 4.
1	0	1	Sets the TGIA3 interrupt skipping count to 5.
1	1	0	Sets the TGIA3 interrupt skipping count to 6.
1	1	1	Sets the TGIA3 interrupt skipping count to 7.

Downloaded from Arrow.com.

23.2.26 Timer Interrupt Skipping Counter (TITCNT)

Address(es): MTU.TITCNT 000D 0A31h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	T3ACNT[2:0]			—	T4VCNT[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	T4VCNT[2:0]	TCIV4 Interrupt Counter	While the T4VEN bit in TITCR is set to 1, the count in these bits is incremented every time a TCIV4 interrupt source occurs.	R
b3	—	Reserved	This bit is read as 0.	R
b6 to b4	T3ACNT[2:0]	TGIA3 Interrupt Counter	While the T3AEN bit in TITCR is set to 1, the count in these bits is incremented every time a TGIA3 interrupt source occurs.	R
b7	—	Reserved	This bit is read as 0.	R

Note: To clear the TITCNT counter, set the T3AEN and T4VEN bits in the TITCR register to 0.

The TITCNT counter counts the number of interrupt source occurrences for interrupt skipping. The TITCNT counter retains the values even after stopping the count operation of counters MTU4.TCNT and MTU3.TCNT.

T4VCNT[2:0] Bits (TCIV4 Interrupt Counter)

[Clearing conditions]

- When the TITCNT.T4VCNT[2:0] bits match the TITCR.T4VCOR[2:0] bits
- When the TITCR.T4VEN bit is set to 0
- When the TITCR.T4VCOR[2:0] bits are set to 000b

T3ACNT[2:0] Bits (TGIA3 Interrupt Counter)

[Clearing conditions]

- When the TITCNT.T3ACNT[2:0] bits match the TITCR.T3ACOR[2:0] bits
- When the TITCR.T3AEN bit is set to 0
- When the TITCR.T3ACOR[2:0] bits are set to 000b

Downloaded from Arrow.com.

23.2.27 Timer Buffer Transfer Set Register (TBTER)

Address(es): MTU.TBTER 000D 0A32h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	BTE[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	BTE[1:0]	Buffer Transfer Disable and Interrupt Skipping Link Setting	These bits enable or disable transfer from the buffer registers used in complementary PWM mode to the temporary registers and specify whether to link the transfer with interrupt skipping operation. Refer to Table 23.39 for details.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The TBTER register enables or disables transfer from the buffer registers used in complementary PWM mode to the temporary registers and specifies whether to link the transfer with interrupt skipping operation.

Table 23.39 Setting of TBTER.BTE[1:0] Bits

Bit 1	Bit 0	
BTE[1]	BTE[0]	Description
0	0	Enables transfer from the buffer registers to the temporary registers*1 and does not link the transfer with interrupt skipping operation.
0	1	Disables transfer from the buffer registers to the temporary registers.
1	0	Links transfer from the buffer registers to the temporary registers with interrupt skipping operation.*2
1	1	Setting prohibited

Note: Target buffer registers: MTU3.TGRC, MTU3.TGRD, MTU4.TGRC, MTU4.TGRD, and MTU.TCBR

Note 1. Data is transferred in accordance with the TMDR.MD[3:0] bit setting. For details, refer to section 23.3.8, Complementary PWM Mode.

Note 2. When interrupt skipping is disabled (the TITCR.T3AEN and T4VEN bits or the interrupt skipping count setting bits (T3ACOR[2:0] and T4VCOR[2:0]) in the TITCR register are set to 000b), be sure to disable link of buffer transfer with interrupt skipping (set the TBTER.BTE[1] bit to 0).
If link with interrupt skipping is enabled while interrupt skipping is disabled, buffer transfer will not be performed.

Downloaded from Arrow.com.

23.2.28 Timer Dead Time Enable Register (TDER)

Address(es): MTU.TDER 000D 0A34h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	TDER
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	TDER	Dead Time Enable	0: No dead time is generated 1: Dead time is generated*1	R/(W)
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. TDDR must be set to 1 or a larger value.

The TDER register specifies dead time generation in complementary PWM mode. The MTU3 has one TDER register. The TDER register should be modified only while the TCNT counter stops.

TDER Bit (Dead Time Enable)

This bit specifies whether to generate dead time.

[Clearing condition]

- When 0 is written to the TDER bit after reading the TDER bit = 1

Downloaded from Arrow.com.

23.2.29 Timer Waveform Control Register (TWCR)

Address(es): MTU.TWCR 000D 0A60h

	b7	b6	b5	b4	b3	b2	b1	b0
	CCE	—	—	—	—	—	—	WRE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	WRE	Initial Output Inhibition Enable	0: Initial value specified in TOCR is output 1: Initial output is inhibited	R/(W) *1
b6 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	CCE	Compare Match Clear Enable	0: Counters are not cleared at MTU3.TGRA compare match 1: Counters are cleared at MTU3.TGRA compare match	R/(W) *2

Note 1. Do not set this bit to 1 unless complementary PWM mode is selected.
Note 2. Do not set this bit to 1 unless complementary PWM mode 1 is selected.

The TWCR register controls the output waveform when synchronous counter clearing occurs in counters MTU3.TCNT and MTU4.TCNT in complementary PWM mode and specifies whether to clear the counters at the MTU3.TGRA compare match.
The TWCR.CCE bit and TWCR.WRE bit should be modified only while the TCNT counter stops.

WRE Bit (Initial Output Inhibition Enable)

This bit selects the waveform output when synchronous counter clearing occurs in complementary PWM mode.
The initial output is prohibited only when synchronous clearing occurs within the Tb interval at the trough in complementary PWM mode. When synchronous clearing occurs outside this interval, the initial value specified in the TOCR register is output regardless of the WRE bit setting. The initial value specified in the TOCR register is also output when synchronous clearing occurs in the Tb interval at the trough immediately after counters MTU3.TCNT and MTU4.TCNT start operation.
For the Tb interval at the trough in complementary PWM mode, refer to Figure 23.40.
[Setting condition]

- When 1 is written to the WRE bit after reading the WRE bit = 0

CCE Bit (Compare Match Clear Enable)

This bit specifies whether to clear counters at the MTU3.TGRA compare match in complementary PWM mode 1.
[Setting condition]

- When 1 is written to the CCE bit after reading the CCE bit = 0

Downloaded from Arrow.com.

23.2.30 Noise Filter Control Registers (NFCR)

Address(es): MTU0.NFCR 000D 0A90h, MTU1.NFCR 000D 0A91h, MTU2.NFCR 000D 0A92h, MTU3.NFCR 000D 0A93h, MTU4.NFCR 000D 0A94h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	NFCS[1:0]		NFDEN	NFCEN	NFBEN	NFAEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	NFAEN	Noise Filter A Enable	0: The noise filter for the MTIOCnA pin is disabled 1: The noise filter for the MTIOCnA pin is enabled	R/W
b1	NFBEN	Noise Filter B Enable	0: The noise filter for the MTIOCnB pin is disabled 1: The noise filter for the MTIOCnB pin is enabled	R/W
b2	NFCEN	Noise Filter C Enable*1	0: The noise filter for the MTIOCnC pin is disabled 1: The noise filter for the MTIOCnC pin is enabled	R/W
b3	NFDEN	Noise Filter D Enable*2	0: The noise filter for the MTIOCnD pin is disabled 1: The noise filter for the MTIOCnD pin is enabled	R/W
b5, b4	NFCS[1:0]	Noise Filter Clock Select	b5 b4 0 0: PCLK/1 0 1: PCLK/8 1 0: PCLK/32 1 1: The clock source for counting is the external clock	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. These bits are reserved in MTU1 and MTU2. These bits are read as 0, and writing to them is not possible.
Note 2. These bits are reserved in MTU0, MTU1 and MTU2. These bits are read as 0, and writing to them is not possible.

The MTUn.NFCR registers (n = 0 to 4) enable and disable the noise filters for the MTIOCnm (n = 0 to 4; m = A to D) pins and sets the sampling clocks for the noise filters.

NFAEN Bit (Noise Filter A Enable)

This bit disables or enables the noise filter for input from the MTIOCnA pin. Since unexpected edges may be internally generated when the value of the NFAEN bit is changed, select the output compare function in the timer I/O control register and set the TMDR.MD[3:0] bits to a value other than 0000b (normal mode) before changing the value.

NFBEN Bit (Noise Filter B Enable)

This bit disables or enables the noise filter for input from the MTIOCnB pin. Since unexpected edges may be internally generated when the value of the NFBEN bit is changed, select the output compare function in the timer I/O control register and set the TMDR.MD[3:0] bits to a value other than 0000b (normal mode) before changing the value.

NFCEN Bit (Noise Filter C Enable)

This bit disables or enables the noise filter for input from the MTIOCnC pin. Since unexpected edges may be internally generated when the value of the NFCEN bit is changed, select the output compare function in the timer I/O control register and set the TMDR.MD[3:0] bits to a value other than 0000b (normal mode) before changing the value.

NFDEN Bit (Noise Filter D Enable)

This bit disables or enables the noise filter for input from the MTIOCnD pin. Since unexpected edges may be internally generated when the value of the NFDEN bit is changed, select the output compare function in the timer I/O control register and set the TMDR.MD[3:0] bits to a value other than 0000b (normal mode) before changing the value.

Downloaded from Arrow.com.

NFCS[1:0] Bits (Noise Filter Clock Select)

These bits set the sampling interval for the noise filters. When setting the NFCS[1:0] bits, wait for two cycles of the selected sampling interval before setting the input-capture function. When the NFCS[1:0] bits are set to 11b, selecting the external clock as the source to drive counting, wait for two cycles of the external clock before setting the input-capture function.

23.2.31 Bus Master Interface

The timer counters (TCNT), timer general registers (TGR), timer subcounter (TCNTS), timer cycle buffer register (TCBR), timer dead time data register (TDDR), timer cycle data register (TCDR), timer A/D converter start request control register (TADCR), timer A/D converter start request cycle set registers (TADCORA/TADCORB), and timer A/D converter start request cycle set buffer registers (TADCOBRA/TADCOBRB) are 16-bit registers. A 16-bit data bus to the bus master enables 16-bit read/write access. 8-bit read/write is not allowed. Access the registers in 16-bit units.
All registers other than the above registers are 8-bit registers, so read/write access should be performed in 8-bit units.

Downloaded from Arrow.com.

23.3 Operation

23.3.1 Basic Functions

Each channel has the TCNT counter and the TGR register. The TCNT counter performs up-counting, and is also capable of free-running operation, periodic counting, and external event counting.
Each TGR register can be used as an input capture register or an output compare register.

(1) Counter Operation

When one of bits CST0 to CST4 in the TSTR register is set to 1, the TCNT counter for the corresponding channel begins counting. The TCNT counter can operate as a free-running counter, periodic counter, for example.

(a) Example of Count Operation Setting Procedure

Figure 23.4 shows an example of the count operation setting procedure.

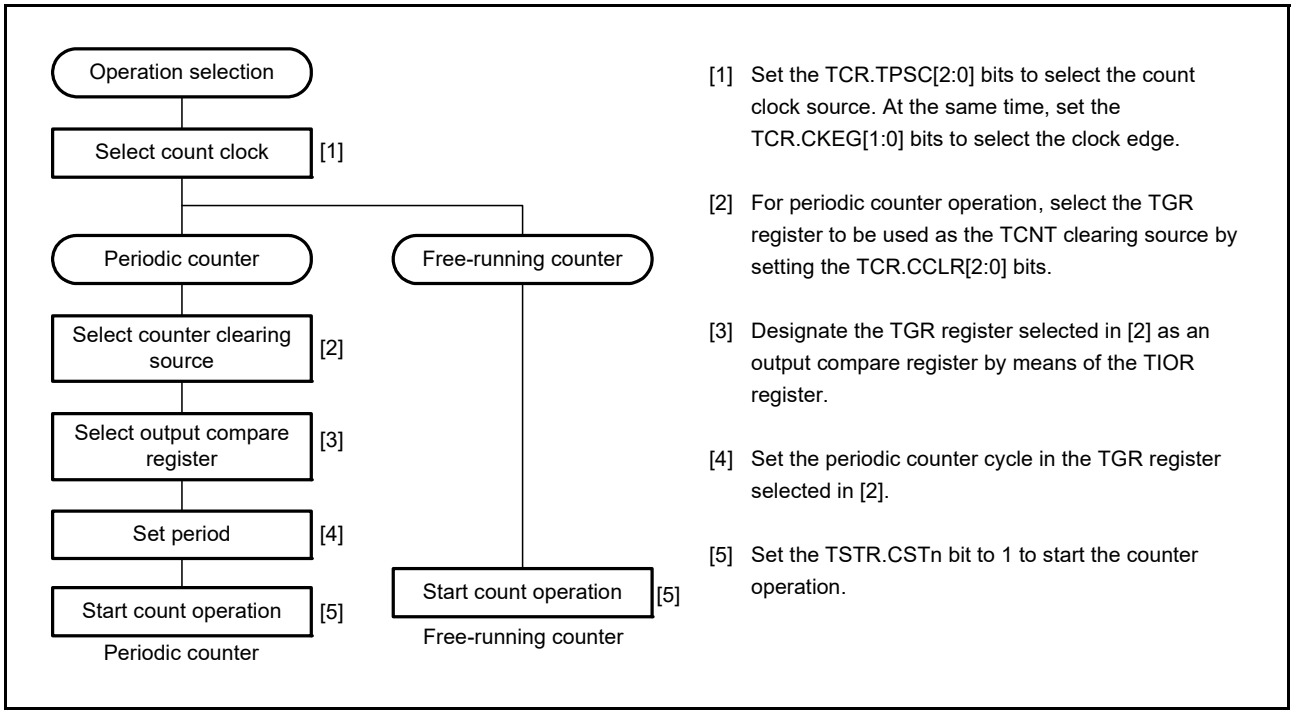


Figure 23.4 Example of Counter Operation Setting Procedure

Downloaded from Arrow.com.

(b) Free-Running Count Operation and Periodic Count Operation

Immediately after a reset, the MTU's TCNT counters are all designated as free-running counters. When the relevant CSTn bit in the TSTR register is set to 1, the corresponding TCNT counter starts up-count operation as a free-running counter. When TCNT overflows (from FFFFh to 0000h), the MTU requests an interrupt if the corresponding TCIEV bit in the TIER register is 1. After an overflow, the TCNT counter starts counting up again from 0000h.
Figure 23.5 illustrates free-running counter operation.

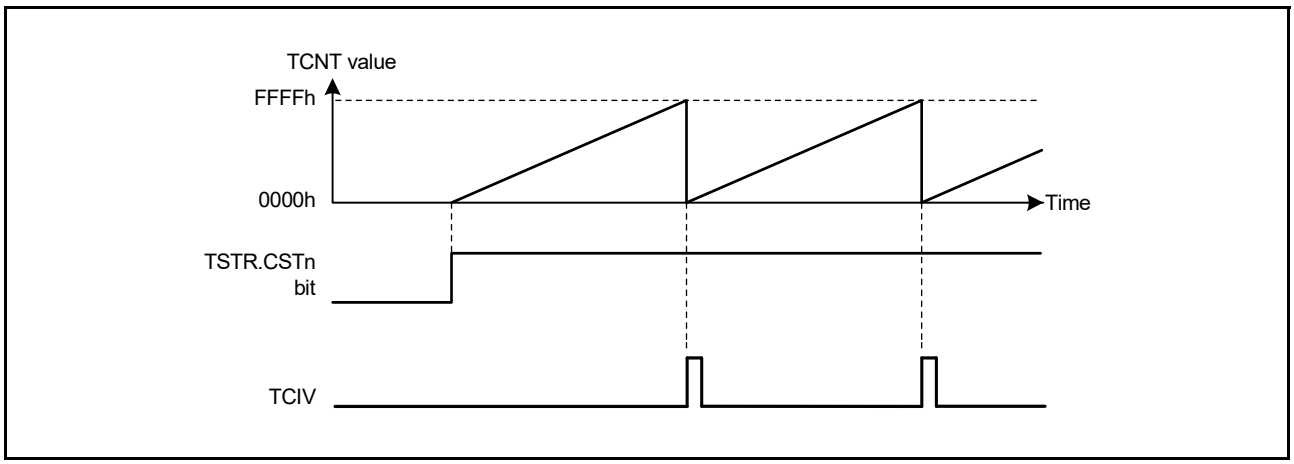


Figure 23.5 Free-Running Counter Operation

When compare match is selected as the TCNT clearing source, the TCNT counter for the relevant channel performs periodic count operation. The TGR register for setting the cycle is designated as an output compare register, and counter clearing by compare match is selected by means of the TCR.CCLR[2:0] bits. After the settings have been made, the TCNT counter starts up-count operation as a periodic counter when the corresponding bit in the TSTR register is set to 1. When the count matches the value in the TGR register, the TCNT counter is set to 0000h.
If the value of the corresponding TIER.TGIE bit is 1 at this point, the MTU requests an interrupt. After a compare match, the TCNT counter starts counting up again from 0000h.
Figure 23.6 illustrates periodic counter operation.

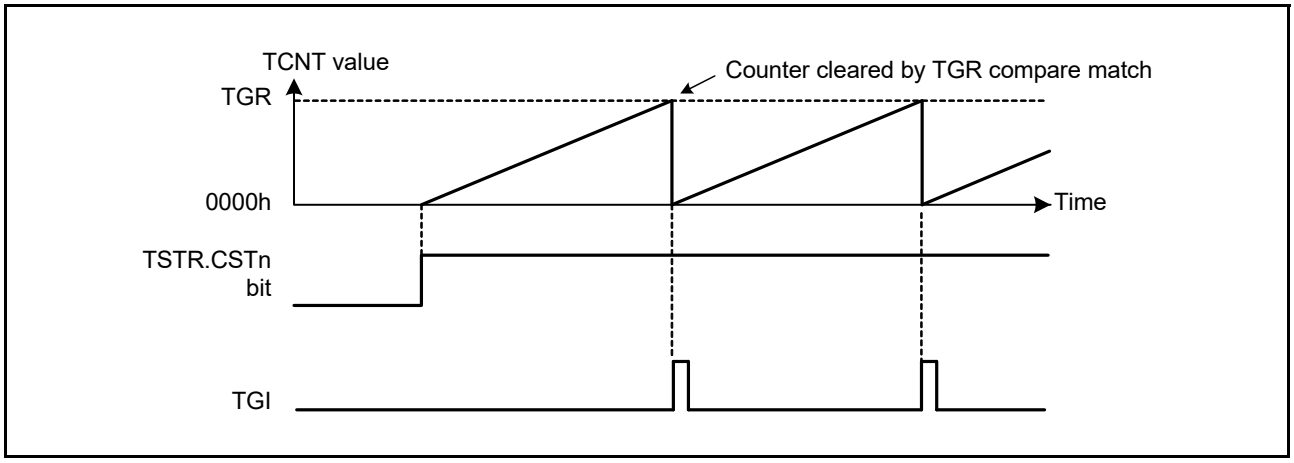


Figure 23.6 Periodic Counter Operation

Downloaded from Arrow.com.

(2) Waveform Output by Compare Match

The MTU can output low or high or toggle output from the corresponding output pin using compare match.

(a) Example of Procedure for Setting Waveform Output by Compare Match

Figure 23.7 shows an example of the procedure for setting waveform output by compare match.

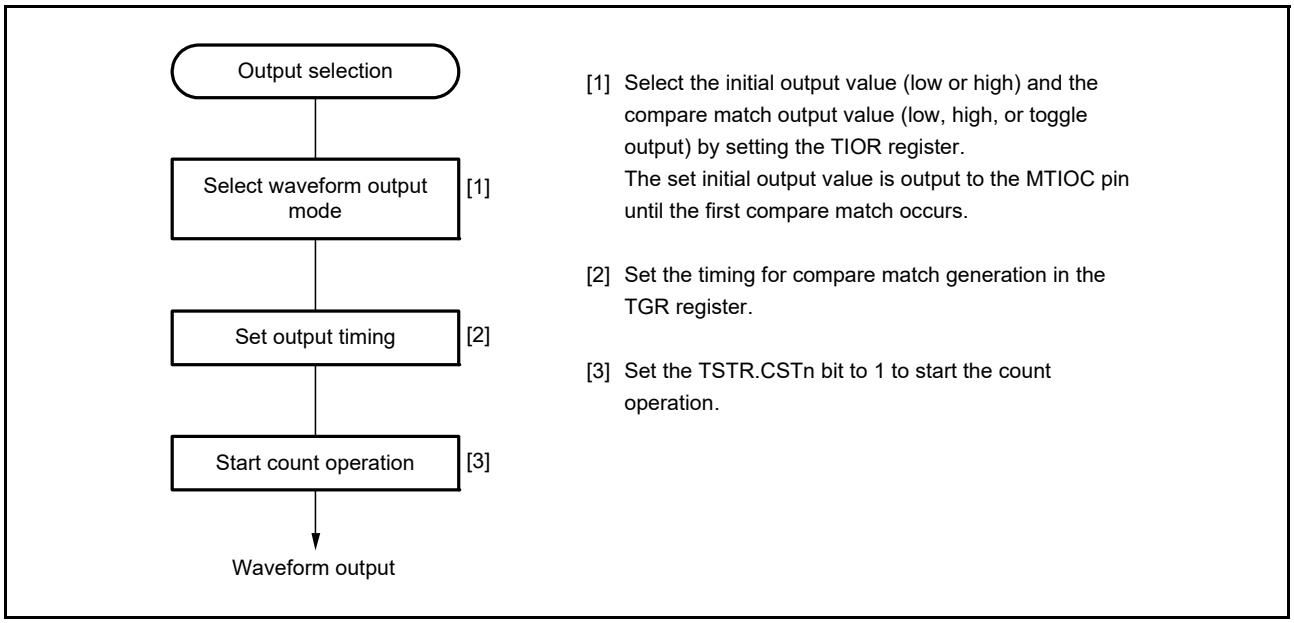


Figure 23.7 Example of Procedure for Setting Waveform Output by Compare Match

Downloaded from Arrow.com.

(b) Examples of Waveform Output Operation

Figure 23.8 shows an example of low output and high output.
In this example, the TCNT counter has been designated as a free-running counter, and settings have been made so that high is output by compare match A and low is output by compare match B. When the pin level is the same as the specified level, the pin level does not change.

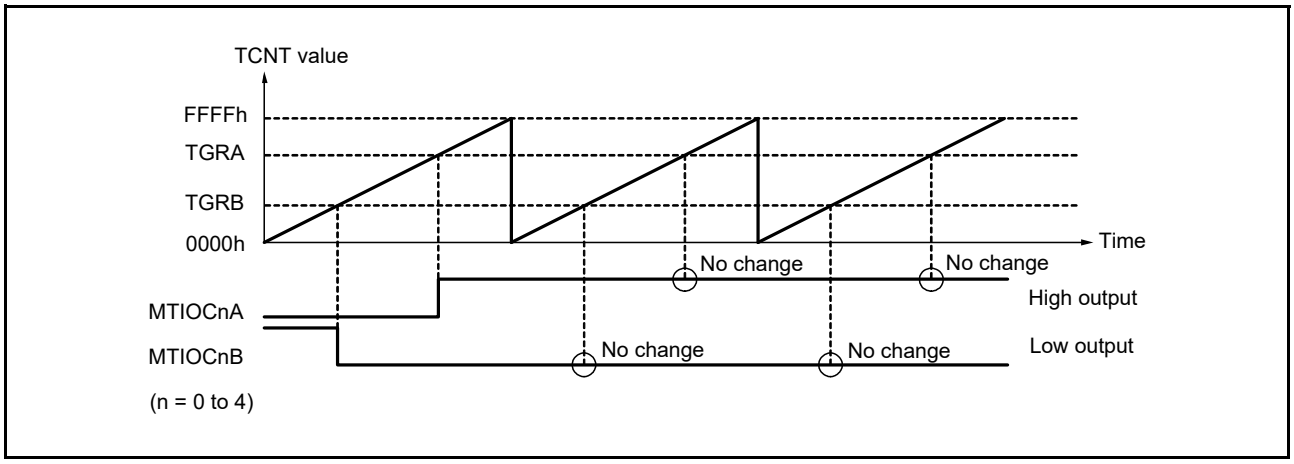


Figure 23.8 Example of Low Output and High Output Operation

Figure 23.9 shows an example of toggle output.
In this example, the TCNT counter has been designated as a periodic counter (with counter clearing on compare match B), and settings have been made so that the output is toggled by both compare match A and compare match B.

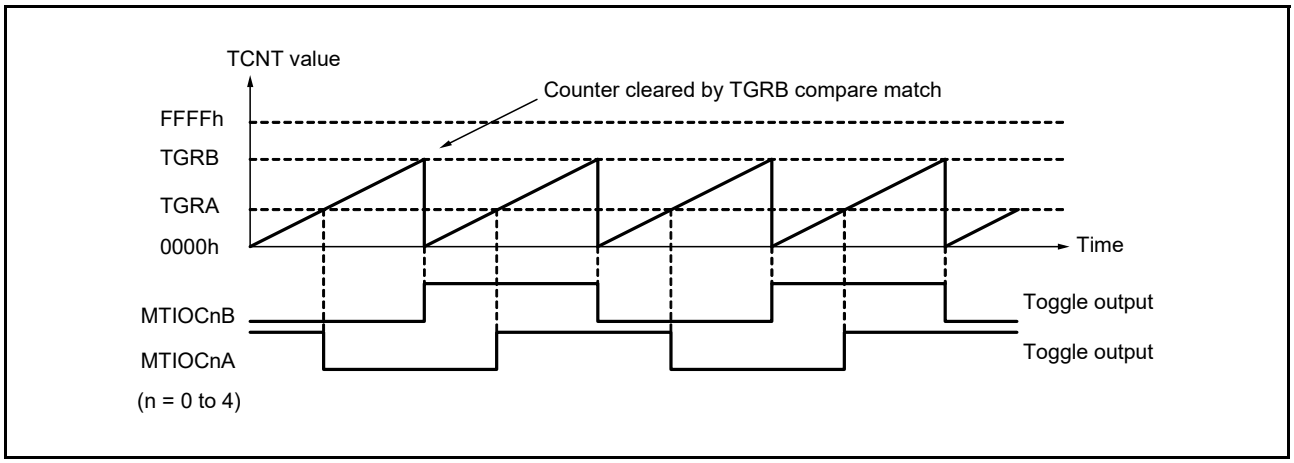


Figure 23.9 Example of Toggle Output Operation

Downloaded from Arrow.com.

(3) Input Capture Function

The TCNT value can be transferred to the TGR register on detection of the input edge of the MTIOCnm (n = 0 to 4; m = A to D) pin.
The rising edge, falling edge, or both edges can be selected as the detection edge. For MTU0 and MTU1, another channel's count clock or compare match signal can also be specified as the input capture source.

Note: When another channel's count clock is used as the input capture input for MTU0 and MTU1, PCLK/1 should not be selected as the count clock used for input capture input. Input capture will not be generated if PCLK/1 is selected.

(a) Example of Input Capture Operation Setting Procedure

Figure 23.10 shows an example of the input capture operation setting procedure.

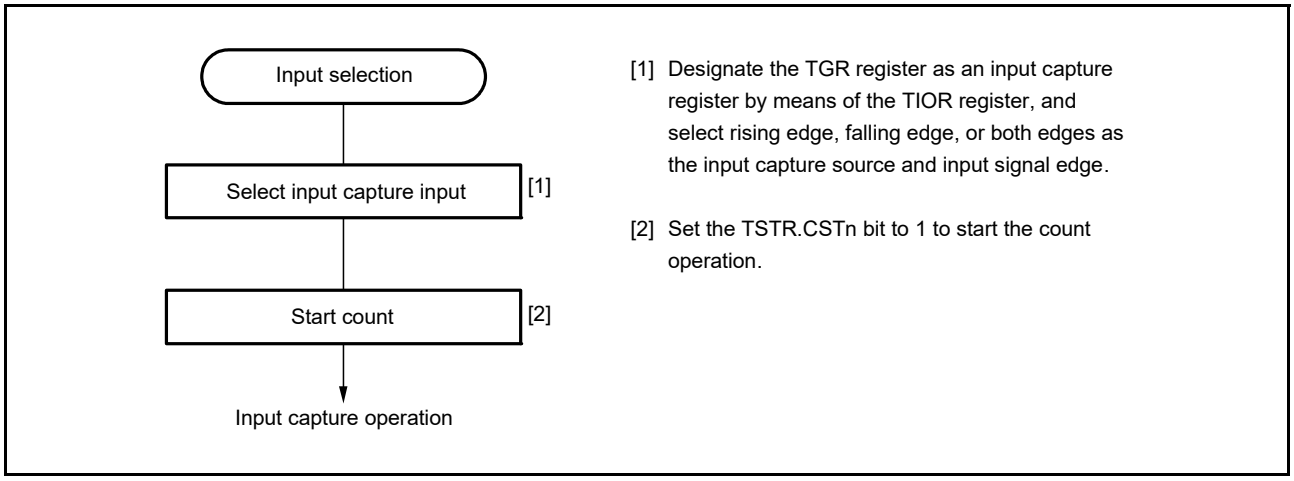


Figure 23.10 Example of Input Capture Operation Setting Procedure

Downloaded from Arrow.com.

(b) Example of Input Capture Operation

Figure 23.11 shows an example of input capture operation.
In this example, both rising and falling edges have been selected as the MTIOCnA pin input capture input edge, the falling edge has been selected as the MTIOCnB pin input capture input edge, and counter clearing by the TGRB input capture has been designated for the TCNT counter.

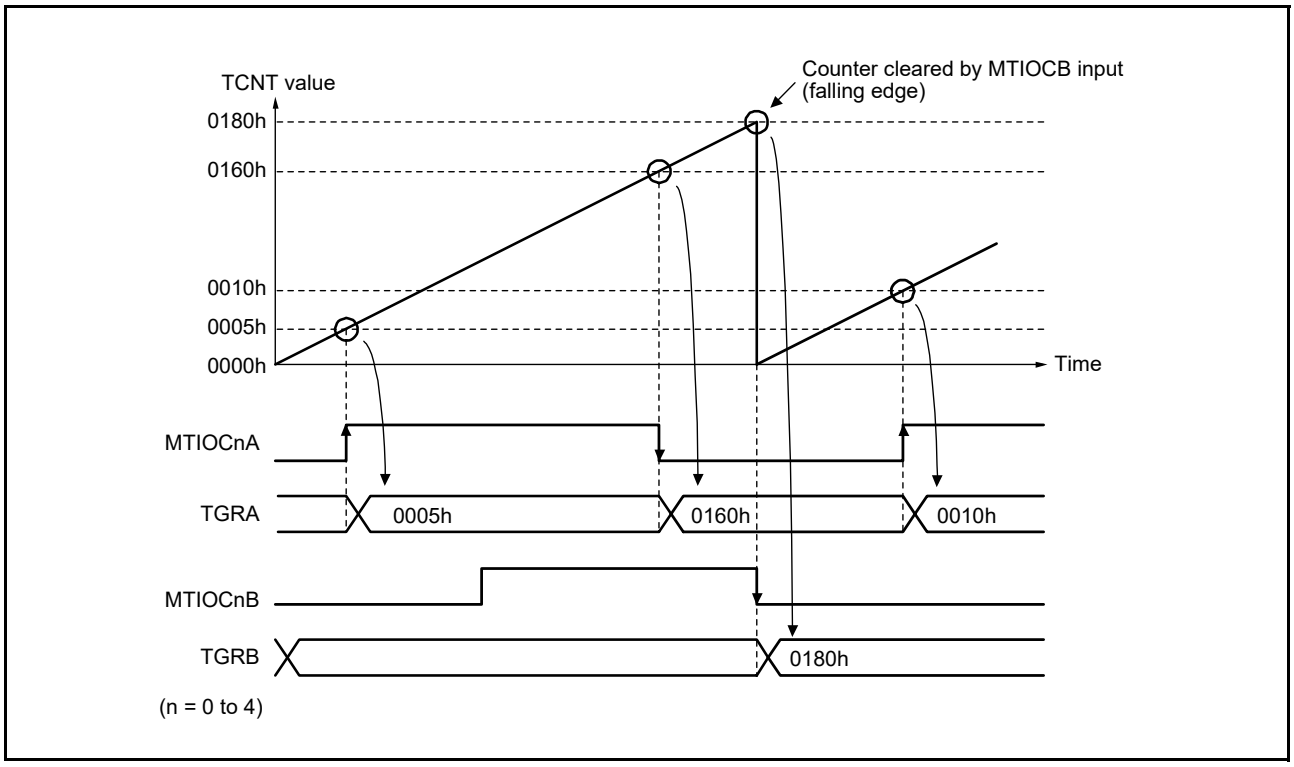


Figure 23.11 Example of Input Capture Operation

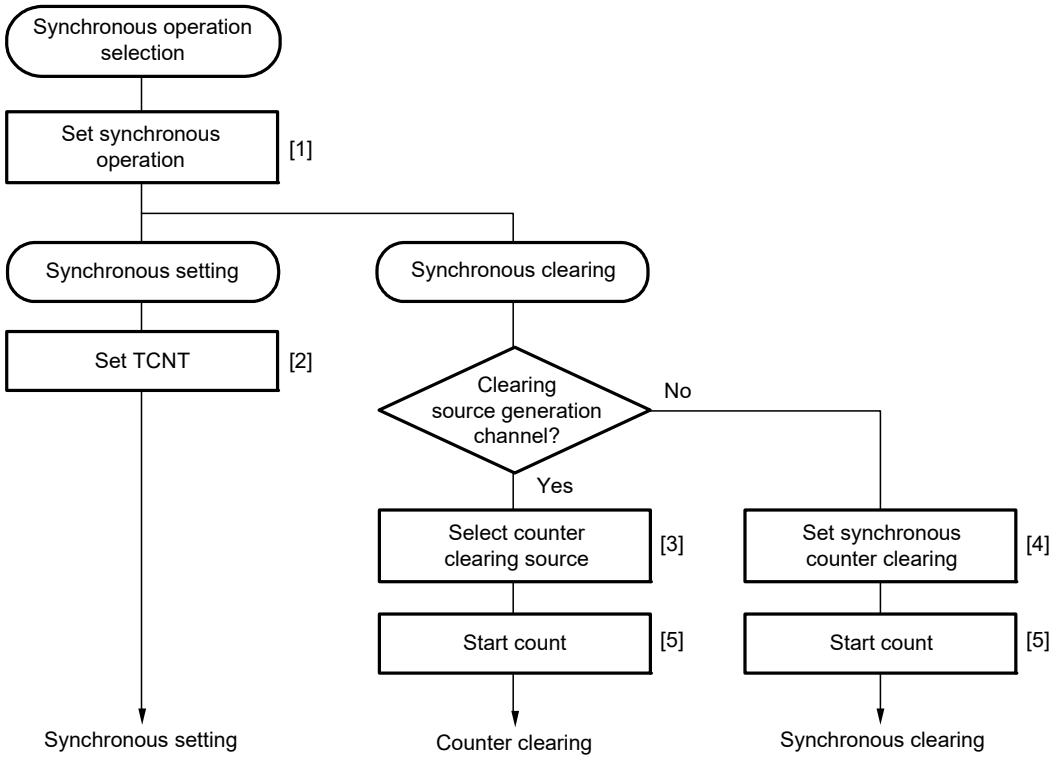
Downloaded from Arrow.com.

23.3.2 Synchronous Operation

In synchronous operation, the values in multiple TCNT counters can be modified simultaneously (synchronous setting). In addition, multiple TCNT counters can be cleared simultaneously (synchronous clearing) by making the appropriate setting in the TCR register.
Synchronous operation increases the number of the TGR registers assigned to a single time base.
MTU0 to MTU4 can all be designated for synchronous operation.

(1) Example of Synchronous Operation Setting Procedure

Figure 23.12 shows an example of the synchronous operation setting procedure.



[1] Set 1 in the TSYR.SYNCn bit (n = 0 to 4) corresponding to the channels to be designated for synchronous operation.

[2] When the TCNT counter of any of the channels designated for synchronous operation is written to, the same value is simultaneously written to the other TCNT counters.

[3] Use the TCR.CCLR[2:0] bits to specify TCNT clearing by input capture/output compare, etc.

[4] Use the TCR.CCLR[2:0] bits to designate synchronous clearing for the counter clearing source.

[5] Set 1 in the TSTR.CSTn bit for the relevant channels, to start the count operation.

Figure 23.12 Example of Synchronous Operation Setting Procedure

Downloaded from Arrow.com.

(2) Example of Synchronous Operation

Figure 23.13 shows an example of synchronous operation.
In this example, synchronous operation and PWM mode 1 have been designated for MTU0 to MTU2, compare match of the MTU0.TGRB register has been set as the counter clearing source in MTU0, and synchronous clearing has been set for the counter clearing source in MTU1 and MTU2.
Three-phase PWM waveforms are output from pins MTIOC0A, MTIOC1A, and MTIOC2A. At this time, synchronous setting and synchronous clearing by MTU0.TGRB compare match are performed for the TCNT counters in MTU0 to MTU2, and the data set in the MTU0.TGRB register is used as the PWM cycle.
For details of PWM modes, refer to section 23.3.5, PWM Modes.

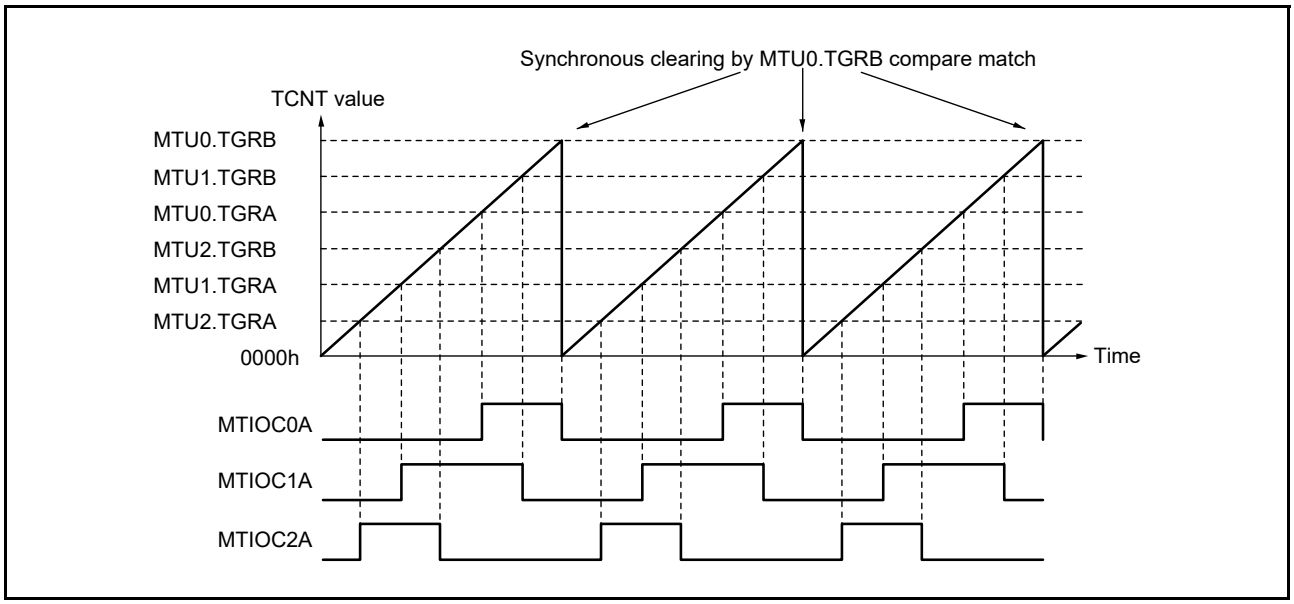


Figure 23.13 Example of Synchronous Operation

Downloaded from Arrow.com.

23.3.3 Buffer Operation

Buffer operation, provided for MTU0, MTU3, and MTU4, enables registers TGRC and TGRD to be used as buffer registers. In MTU0, TGRF register can also be used as a buffer register.
Buffer operation differs depending on whether TGR has been designated as an input capture register or as a compare match register.

Note: MTU0.TGRE register cannot be designated as an input capture register and can only operate as a compare match register.

Table 23.40 shows the register combinations used in buffer operation.

Table 23.40 Register Combinations in Buffer Operation

Channel	Timer General Register	Buffer Register
MTU0	TGRA	TGRC
	TGRB	TGRD
	TGRE	TGRF
MTU3	TGRA	TGRC
	TGRB	TGRD
MTU4	TGRA	TGRC
	TGRB	TGRD

- When TGR register is an output compare register

When a compare match occurs, the value in the buffer register for the corresponding channel is transferred to the timer general register.
This operation is illustrated in Figure 23.14.

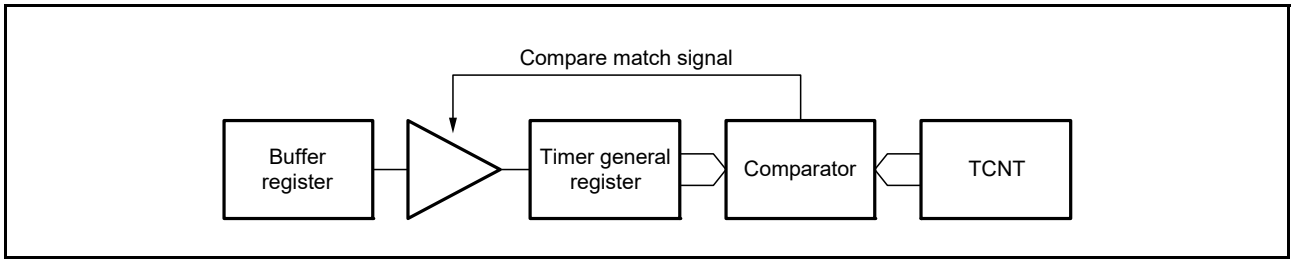


Figure 23.14 Compare Match Buffer Operation

- When TGR register is an input capture register

When an input capture occurs, the value in the TCNT counter is transferred to the TGR register and the value previously held in the TGR register is transferred to the buffer register.
This operation is illustrated in Figure 23.15.

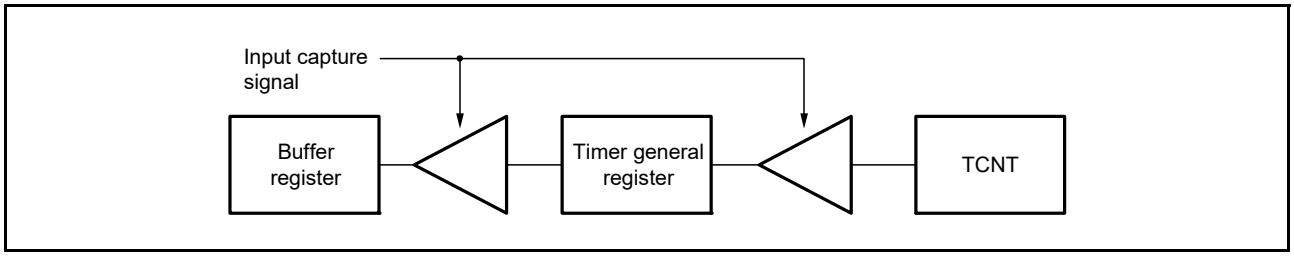


Figure 23.15 Input Capture Buffer Operation

Downloaded from Arrow.com.

(1) Example of Buffer Operation Setting Procedure

Figure 23.16 shows an example of the buffer operation setting procedure.

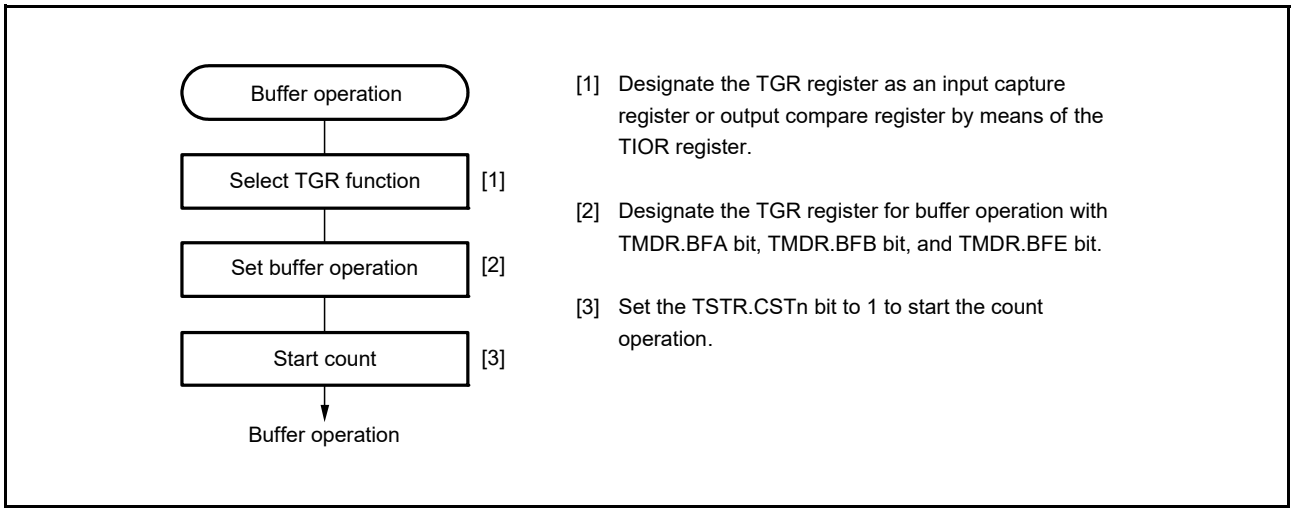


Figure 23.16 Example of Buffer Operation Setting Procedure

(2) Examples of Buffer Operation

(a) When TGR register is an Output Compare Register

Figure 23.17 shows an operation example in which PWM mode 1 has been designated for MTU0, and buffer operation has been designated for registers TGRA and TGRC. The settings used in this example are TCNT clearing by compare match B, high output at compare match A, and low output at compare match B. In this example, the TBTM.TTSA bit is set to 0.

As buffer operation has been set, when compare match A occurs, the output changes and the value in buffer register TGRC is simultaneously transferred to timer general register TGRA. This operation is repeated each time compare match A occurs.

For details of PWM modes, refer to section 23.3.5, PWM Modes.

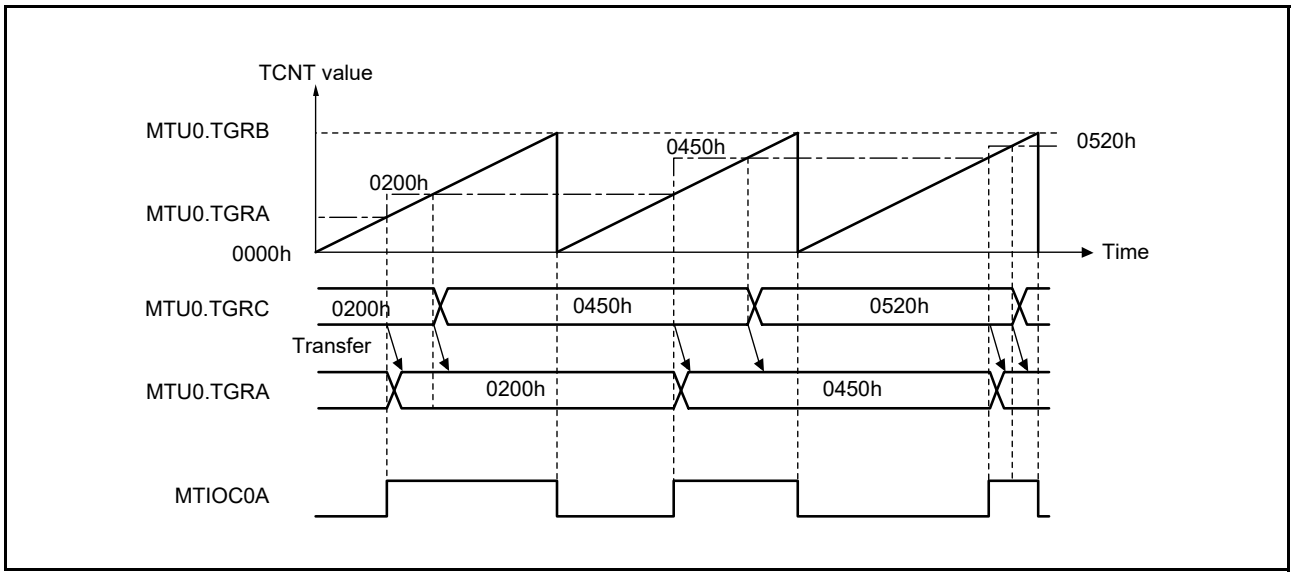


Figure 23.17 Example of Buffer Operation (1)

Downloaded from Arrow.com.

(b) When TGR register is an Input Capture Register

Figure 23.18 shows an operation example in which the TGRA register has been designated as an input capture register, and buffer operation has been designated for registers TGRA and TGRC.
Counter clearing by TGRA input capture has been set for the TCNT counter, and both rising and falling edges have been selected as the MTIOCnA pin input capture input edge.
As buffer operation has been set, when the TCNT value is transferred to the TGRA register upon occurrence of input capture A, the value previously stored in the TGRA register is simultaneously transferred to the TGRC register.

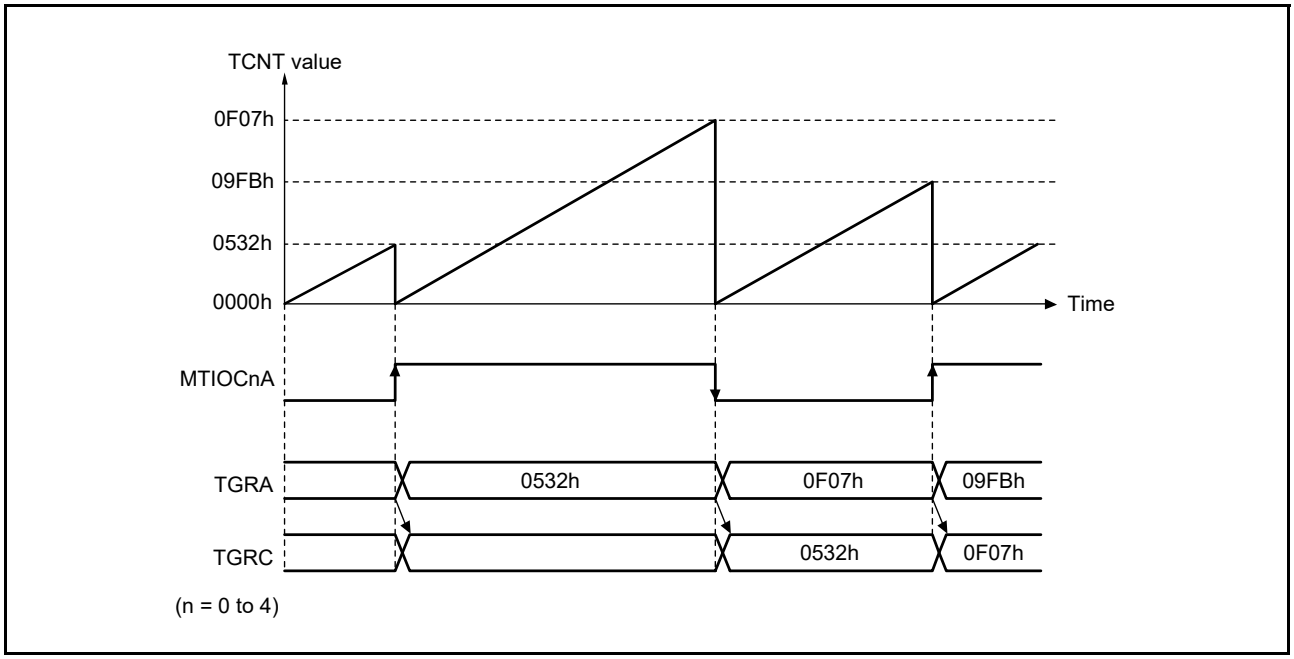


Figure 23.18 Example of Buffer Operation (2)

Downloaded from Arrow.com.

(3) Selecting Timing for Transfer from Buffer Registers to Timer General Registers in Buffer Operation

The timing for transfer from buffer registers to timer general registers can be selected in PWM mode 1 or 2 for MTU0 or in PWM mode 1 for MTU3 and MTU4 by setting the timer buffer operation transfer mode registers (MTU0.TBTM, MTU3.TBTM, and MTU4.TBTM). Either compare match (initial setting) or TCNT clearing can be selected for the transfer timing. TCNT clearing as transfer timing is one of the following cases.

- When the TCNT counter overflows (FFFFh → 0000h)
- When 0000h is written to the TCNT counter during counting
- When the TCNT counter is set to 0000h under the condition specified in the TCR.CCLR[2:0] bits

Note: The TBTM register must be modified only while the TCNT counter stops.

Figure 23.19 shows an operation example in which PWM mode 1 is designated for MTU0 and buffer operation is designated for registers MTU0.TGRA and MTU0.TGRC. The settings used in this example are MTU0.TCNT clearing by compare match B, high output at compare match A, and low output at compare match B. The MTU0.TBTM.TTSA bit is set to 1.

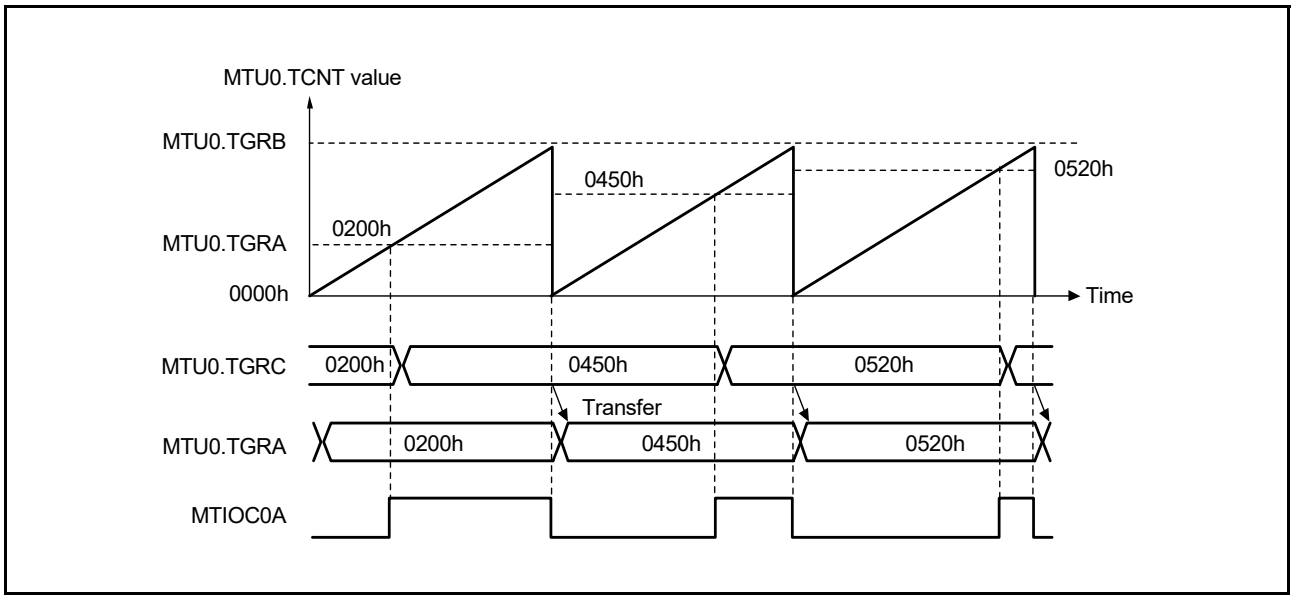


Figure 23.19 Example of Buffer Operation When MTU0.TCNT Clearing is Selected for MTU0.TGRC-to-MTU0.TGRA Transfer Timing

Downloaded from Arrow.com.

23.3.4 Cascaded Operation

In cascaded operation, 16-bit counters in different two channels are used together as a 32-bit counter.
This function works when overflow/underflow of the MTU2.TCNT counter is selected as the count clock for MTU1 through the TCR.TPSC[2:0] bits.
Underflow occurs only when the lower 16 bits of the TCNT counter is in phase counting mode.
Table 23.41 lists the register combinations used in cascaded operation.

Note: When phase counting mode is set for MTU1 or MTU2, the count clock setting is invalid and the counters operate independently in phase counting mode.

Table 23.41 Cascaded Combinations

Combination	Upper 16 Bits	Lower 16 Bits
MTU1 and MTU2	MTU1.TCNT	MTU2.TCNT

For simultaneous input capture of MTU1.TCNT and MTU2.TCNT during cascaded operation, additional input capture input pins can be specified by the TICCR register. The input-capture condition is of edges in the signal produced by taking the logical OR of the input level on the main input pin and the input level on the added input pin. Accordingly, if either is at the high, a change in the level of the other will not produce an edge for detection. For details, refer to (4) Cascaded Operation Example (c). For input capture in cascade connection, refer to section 23.6.22, Simultaneous Input Capture in MTU1.TCNT and MTU2.TCNT in Cascade Connection.
Table 23.42 lists the TICCR setting and input capture input pins.

Table 23.42 TICCR Setting and Input Capture Input Pins

Target Input Capture	TICCR Setting	Input Capture Input Pin
Input capture from MTU1.TCNT to MTU1.TGRA	I2AE bit = 0 (initial value)	MTIOC1A
	I2AE bit = 1	MTIOC1A, MTIOC2A
Input capture from MTU1.TCNT to MTU1.TGRB	I2BE bit = 0 (initial value)	MTIOC1B
	I2BE bit = 1	MTIOC1B, MTIOC2B
Input capture from MTU2.TCNT to MTU2.TGRA	I1AE bit = 0 (initial value)	MTIOC2A
	I1AE bit = 1	MTIOC2A, MTIOC1A
Input capture from MTU2.TCNT to MTU2.TGRB	I1BE bit = 0 (initial value)	MTIOC2B
	I1BE bit = 1	MTIOC2B, MTIOC1B

Downloaded from Arrow.com.

(1) Example of Cascaded Operation Setting Procedure

Figure 23.20 shows an example of the cascaded operation setting procedure.

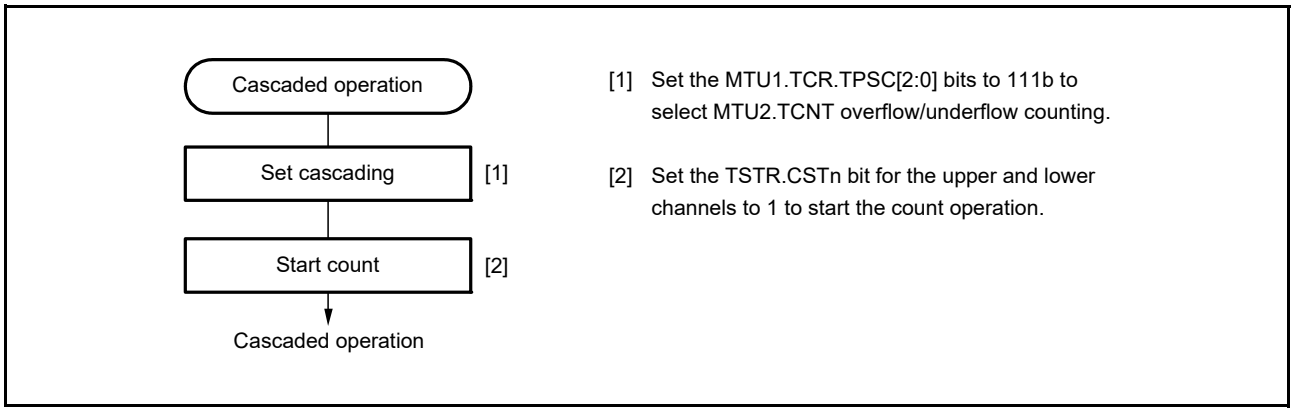


Figure 23.20 Cascaded Operation Setting Procedure

(2) Cascaded Operation Example (a)

Figure 23.21 shows the operation when the MTU1.TCNT counter is set for counting at MTU2.TCNT overflow/underflow and MTU2 is set for phase counting mode 1 while counters MTU1.TCNT and MTU2.TCNT are cascaded. The MTU1.TCNT counter is incremented by MTU2.TCNT overflow and decremented by MTU2.TCNT underflow.

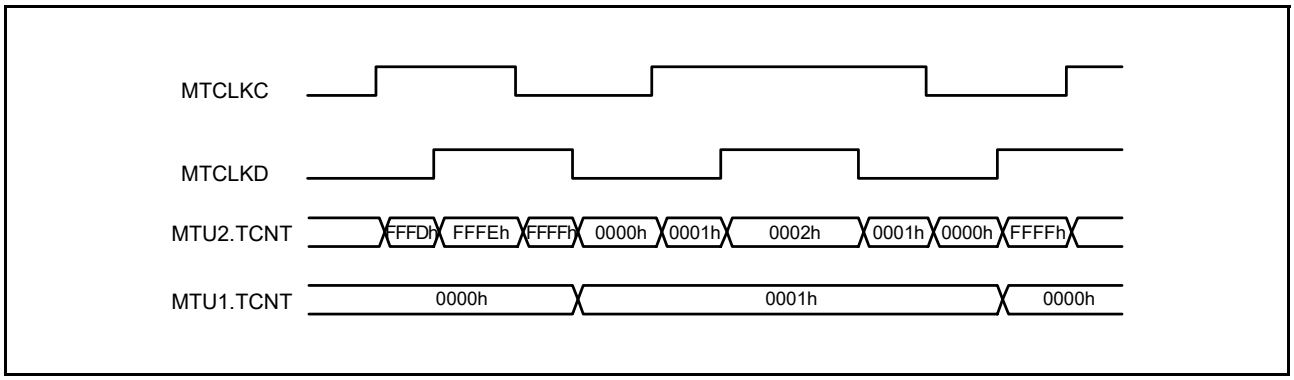


Figure 23.21 Cascaded Operation Example (a)

Downloaded from Arrow.com.

(3) Cascaded Operation Example (b)

Figure 23.22 illustrates the operation when counters MTU1.TCNT and MTU2.TCNT have been cascaded and the TICCR.I2AE bit has been set to 1 to include the MTIOC2A pin in the MTU1.TGRA input capture conditions. In this example, the MTU1.TIOR.IOA[3:0] bits have selected the MTIOC1A rising edge for the input capture timing while the MTU2.TIOR.IOA[3:0] bits have selected the MTIOC2A rising edge for the input capture timing.
Under these conditions, the rising edge of both MTIOC1A and MTIOC2A is used for the MTU1.TGRA input capture condition. For the MTU2.TGRA input capture condition, the MTIOC2A rising edge is used.

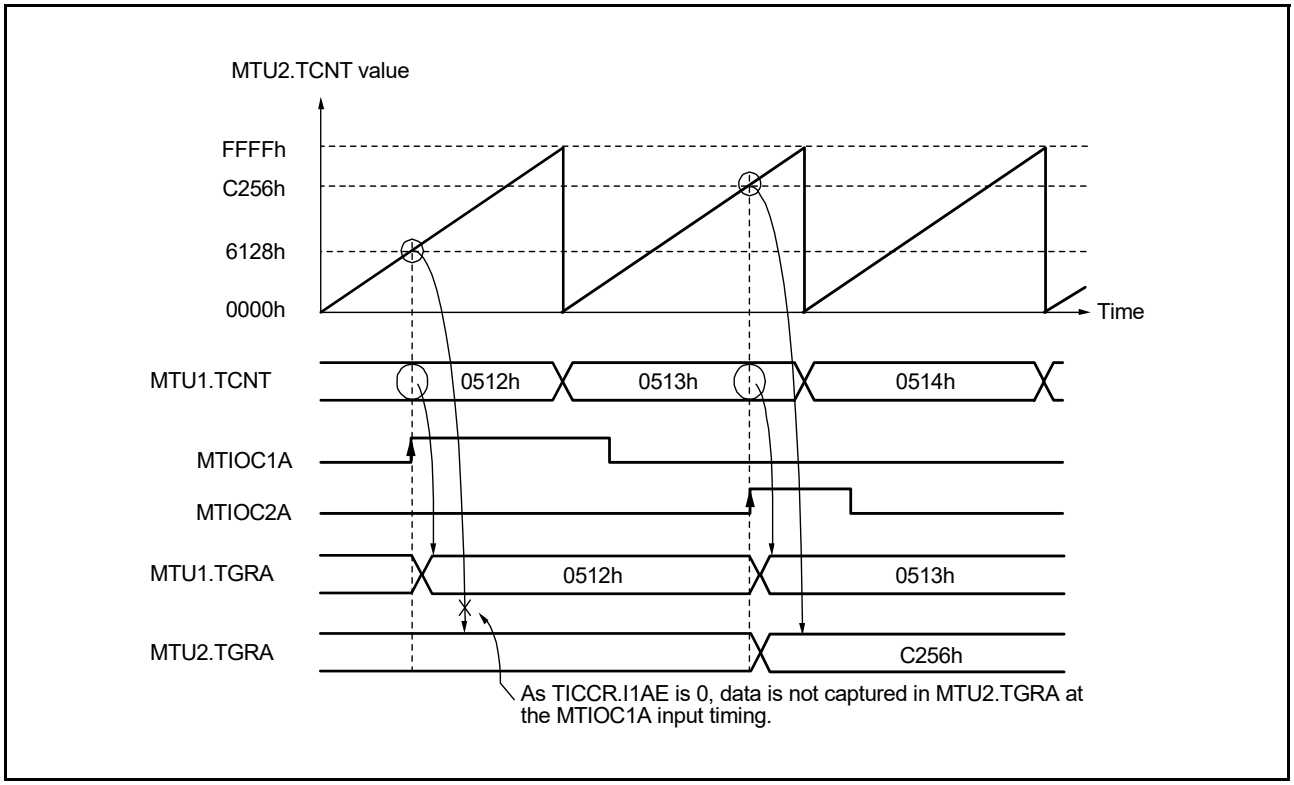
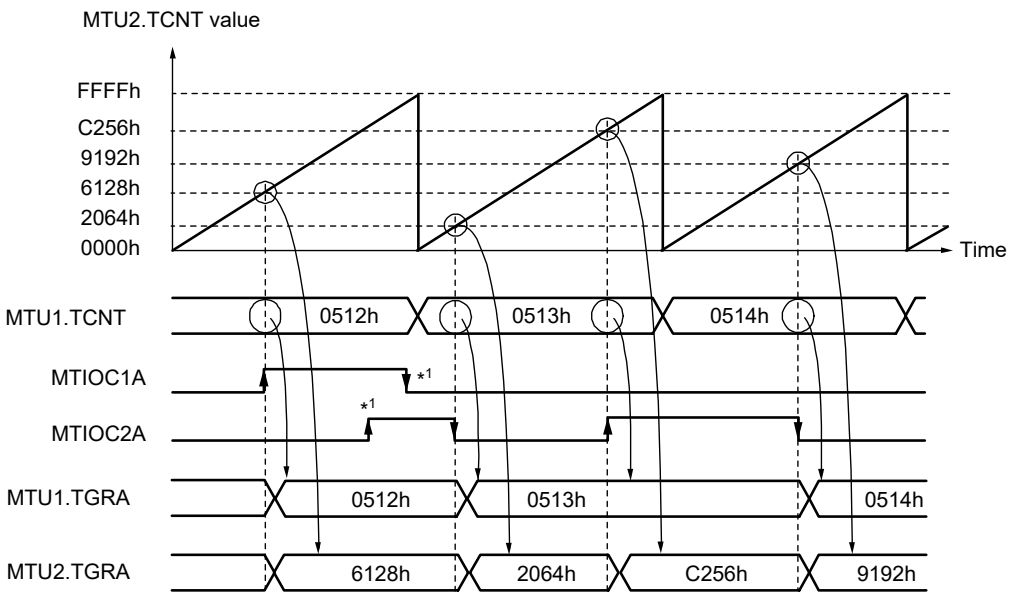


Figure 23.22 Cascaded Operation Example (b)

Downloaded from Arrow.com.

(4) Cascaded Operation Example (c)

Figure 23.23 illustrates the operation when counters MTU1.TCNT and MTU2.TCNT have been cascaded and the I2AE and I1AE bits in TICCR register have been set to 1 to include the MTIOC2A and MTIOC1A pins in the MTU1.TGRA and MTU2.TGRA input capture conditions, respectively. In this example, the IOA[3:0] bits in both MTU1.TIOR and MTU2.TIOR registers have selected both the rising and falling edges for the input capture timing. Under these conditions, the OR result of MTIOC1A and MTIOC2A input is used for the MTU1.TGRA and MTU2.TGRA input capture conditions.



Note 1. When either input signal is at the high level, an edge in the other input signal does not act as the condition for capture.

Figure 23.23 Cascaded Operation Example (c)

Downloaded from Arrow.com.

(5) Cascaded Operation Example (d)

Figure 23.24 illustrates the operation when counters MTU1.TCNT and MTU2.TCNT have been cascaded and the TICCR.I2AE bit has been set to 1 to include the MTIOC2A pin in the MTU1.TGRA input capture conditions. In this example, the MTU1.TIOR.IOA[3:0] bits have selected occurrence of MTU0.TGRA compare match or input capture for the input capture timing while the MTU2.TIOR.IOA[3:0] bits have selected the MTIOC2A rising edge for the input capture timing.

Under these conditions, as the MTU1.TIOR register has selected occurrence of MTU0.TGRA compare match or input capture for the input capture timing, the MTIOC2A edge is not used for MTU1.TGRA input capture condition although the TICCR.I2AE bit has been set to 1.

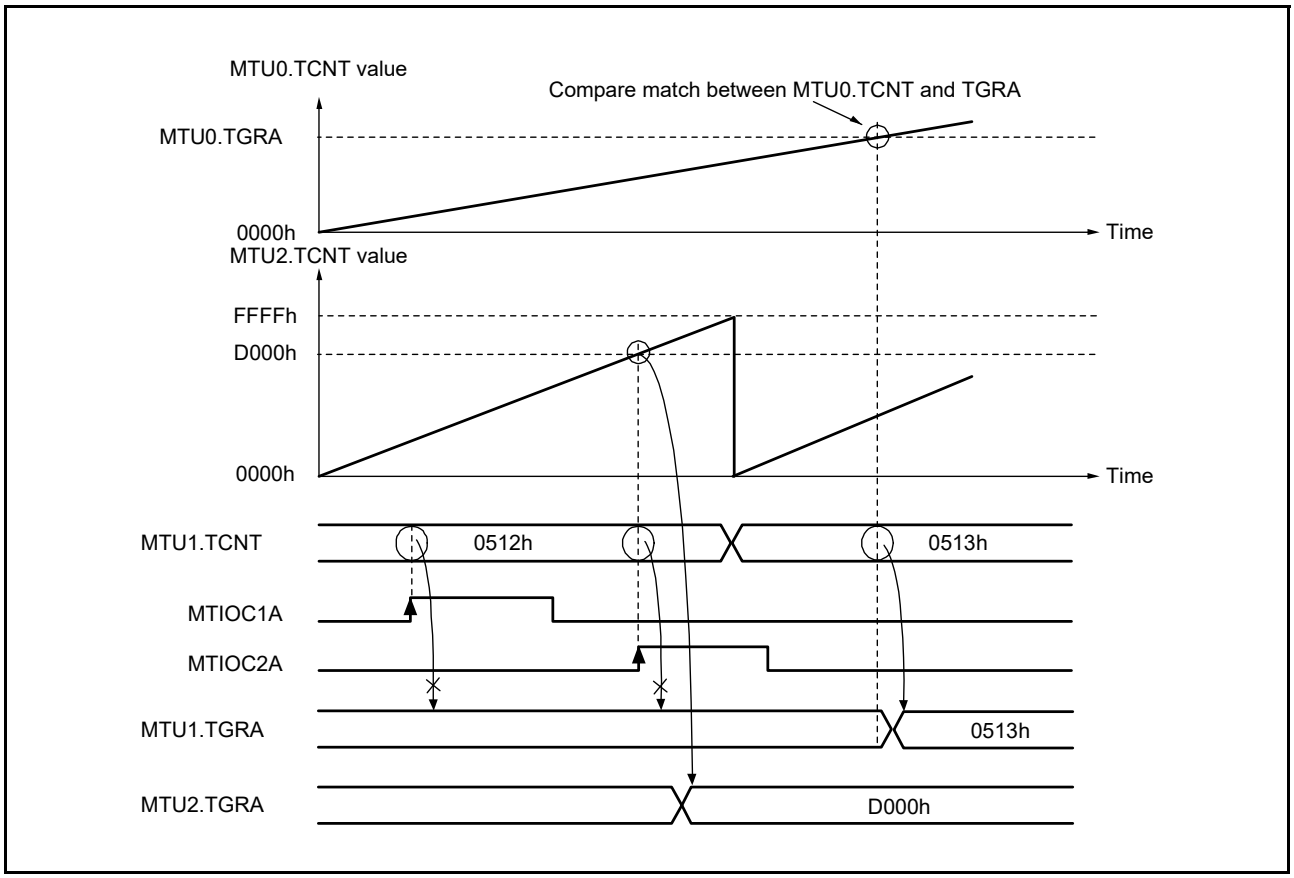


Figure 23.24 Cascaded Operation Example (d)

Downloaded from Arrow.com.

23.3.5 PWM Modes

PWM modes are provided to output PWM waveforms from the external pins. The output level can be selected as low, high, or toggle output in response to a compare match of each TGR register.
PWM waveforms in the range of 0% to 100% duty cycle can be output according to the TGR settings.
By designating TGR compare match as the counter clearing source, the PWM cycle can be specified in that register.
Every channel can be set to PWM mode independently. Channels set to PWM mode can perform synchronous operation with each other or other channels set to any other mode.
There are two PWM modes as described below.

(a) PWM Mode 1

PWM waveforms are output from the MTIOCnA and MTIOCnC pins by pairing the TGRA register with the TGRB register and the TGRC register with the TGRD register. The levels specified by the TIOR.IOA[3:0] and IOC[3:0] bits are output from the MTIOCnA and MTIOCnC pins at compare matches A and C, and the levels specified by the TIOR.IOB[3:0] and IOD[3:0] bits are output at compare matches B and D. The initial output value is set in the TGRA register or the TGRC register. If the values set in paired TGRs are identical, the output value does not change even when a compare match occurs.
In PWM mode 1, up to eight phases of PWM waveforms can be output.

(b) PWM Mode 2

PWM output is generated using one TGR register as the cycle register and the others as duty registers. The level specified in the TIOR register is output at compare matches. Upon counter clearing by a cycle register compare match, the initial value set in the TIOR register is output from each pin. If the values set in the cycle and duty registers are identical, the output value does not change even when a compare match occurs.
In PWM mode 2, up to eight phases of PWM waveforms can be output when using synchronous operation in combination.
The correspondence between PWM output pins and registers is listed in Table 23.43.

Table 23.43 PWM Output Registers and Output Pins

Channel	Register	Output Pins	
		PWM Mode 1	PWM Mode 2
MTU0	MTU0.TGRA	MTIOC0A	MTIOC0A
	MTU0.TGRB		MTIOC0B
	MTU0.TGRC	MTIOC0C	MTIOC0C
	MTU0.TGRD		No pin is assigned for this output.
MTU1	MTU1.TGRA	MTIOC1A	MTIOC1A
	MTU1.TGRB		MTIOC1B
MTU2	MTU2.TGRA	MTIOC2A	MTIOC2A
	MTU2.TGRB		MTIOC2B
MTU3	MTU3.TGRA	MTIOC3A	Setting prohibited
	MTU3.TGRB		
	MTU3.TGRC	MTIOC3C	
	MTU3.TGRD		
MTU4	MTU4.TGRA	MTIOC4A	
	MTU4.TGRB		
	MTU4.TGRC	MTIOC4C	
	MTU4.TGRD		

Note: In PWM mode 2, PWM output is not possible for the TGR register in which the PWM cycle is set.

Downloaded from Arrow.com.

(1) Example of PWM Mode Setting Procedure

Figure 23.25 shows an example of the PWM mode setting procedure.

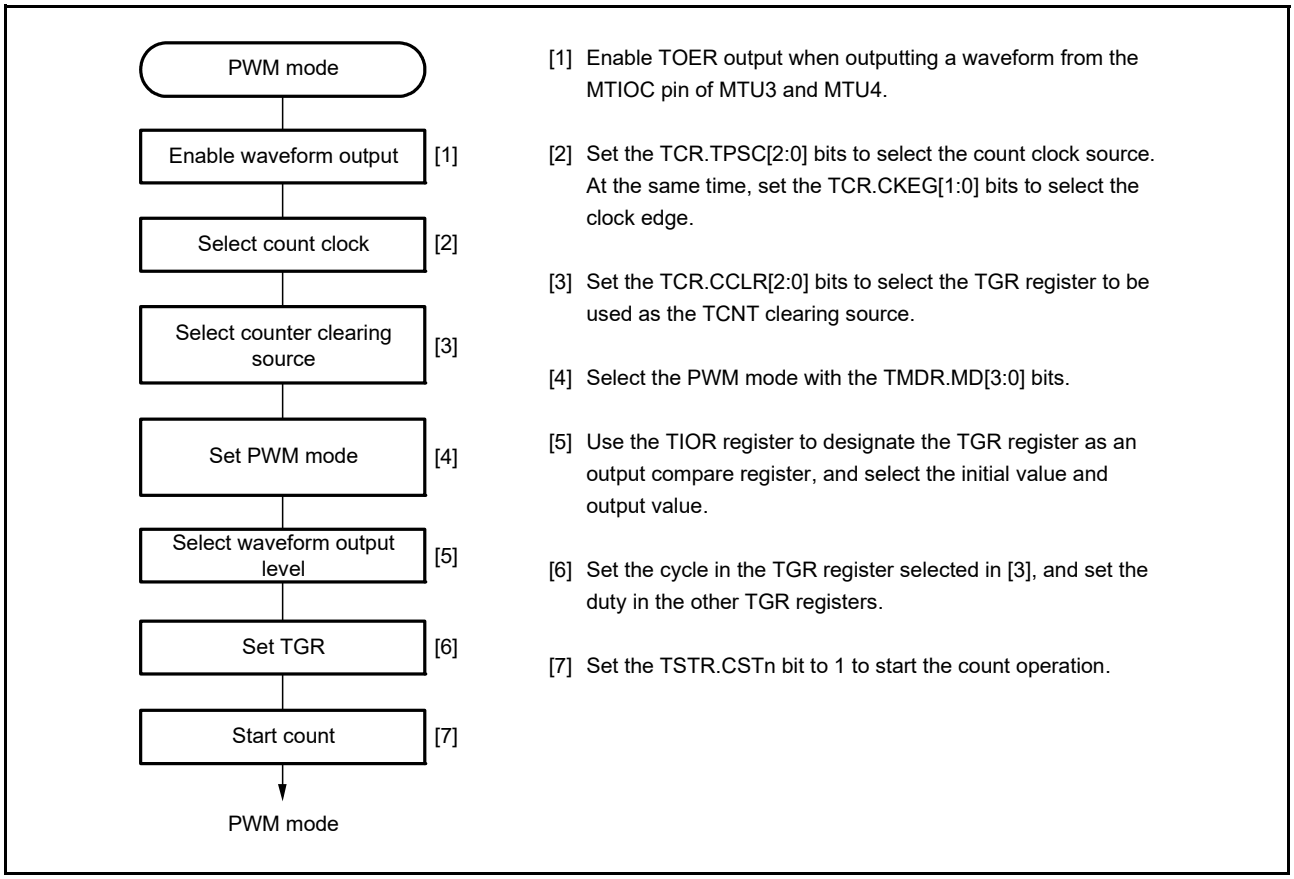


Figure 23.25 Example of PWM Mode Setting Procedure

(2) Examples of PWM Mode Operation

Figure 23.26 shows an example of operation in PWM mode 1.

In this example, TGRA compare match is set as the TCNT clearing source, a low level is set as the initial output value and output value for the TGRA register, and a high level is set as the output value for the TGRB register.

In this case, the value set in the TGRA register is used as the cycle, and the value set in the TGRB register is used as the duty.

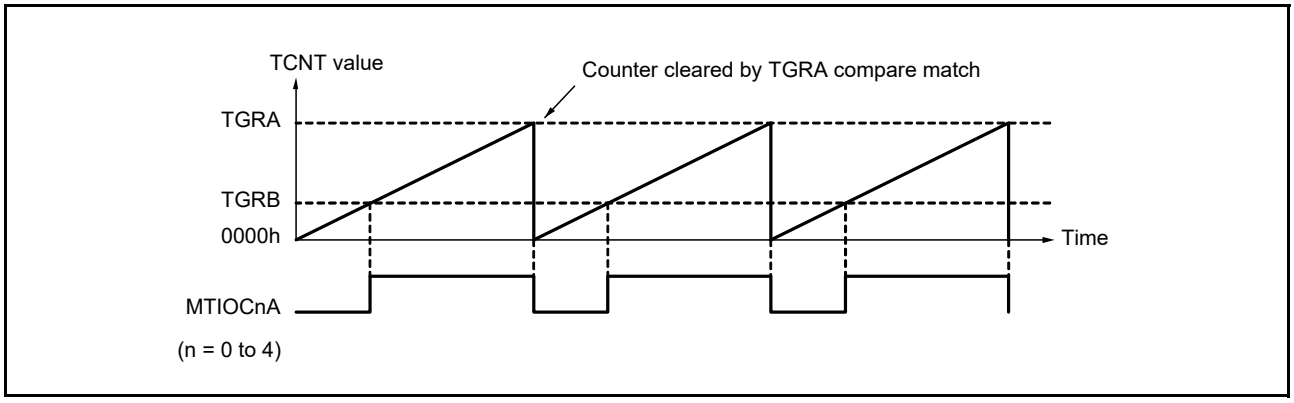


Figure 23.26 Example of PWM Mode Operation

Downloaded from Arrow.com.

Figure 23.27 shows an example of operation in PWM mode 2.
In this example, synchronous operation is designated for MTU0 and MTU1, MTU1.TGRB compare match is set as the TCNT clearing source, and a low level is set as the initial output value and a high level as the output value for the other TGR registers (MTU0.TGRA to MTU0.TGRC and MTU1.TGRA), outputting 4-phase PWM waveforms.
In this case, the value set in the MTU1.TGRB register is used as the cycle, and the values set in the other TGR registers are used as the duty.

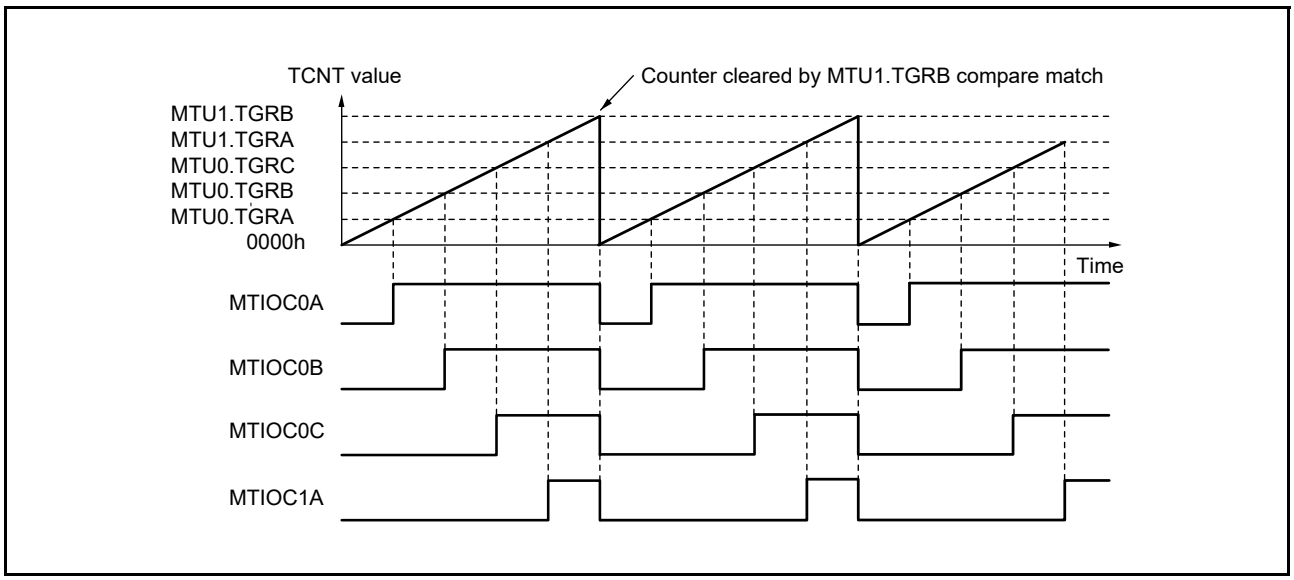


Figure 23.27 Example of PWM Mode Operation

Downloaded from Arrow.com.

Figure 23.28 shows examples of PWM waveform output with 0% duty and 100% duty in PWM mode 1.
In this example, TGRA compare match is set as the TCNT clearing source, a low level is set as the initial output value and output value for the TGRA register, and a high level is set as the output value for the TGRB register.

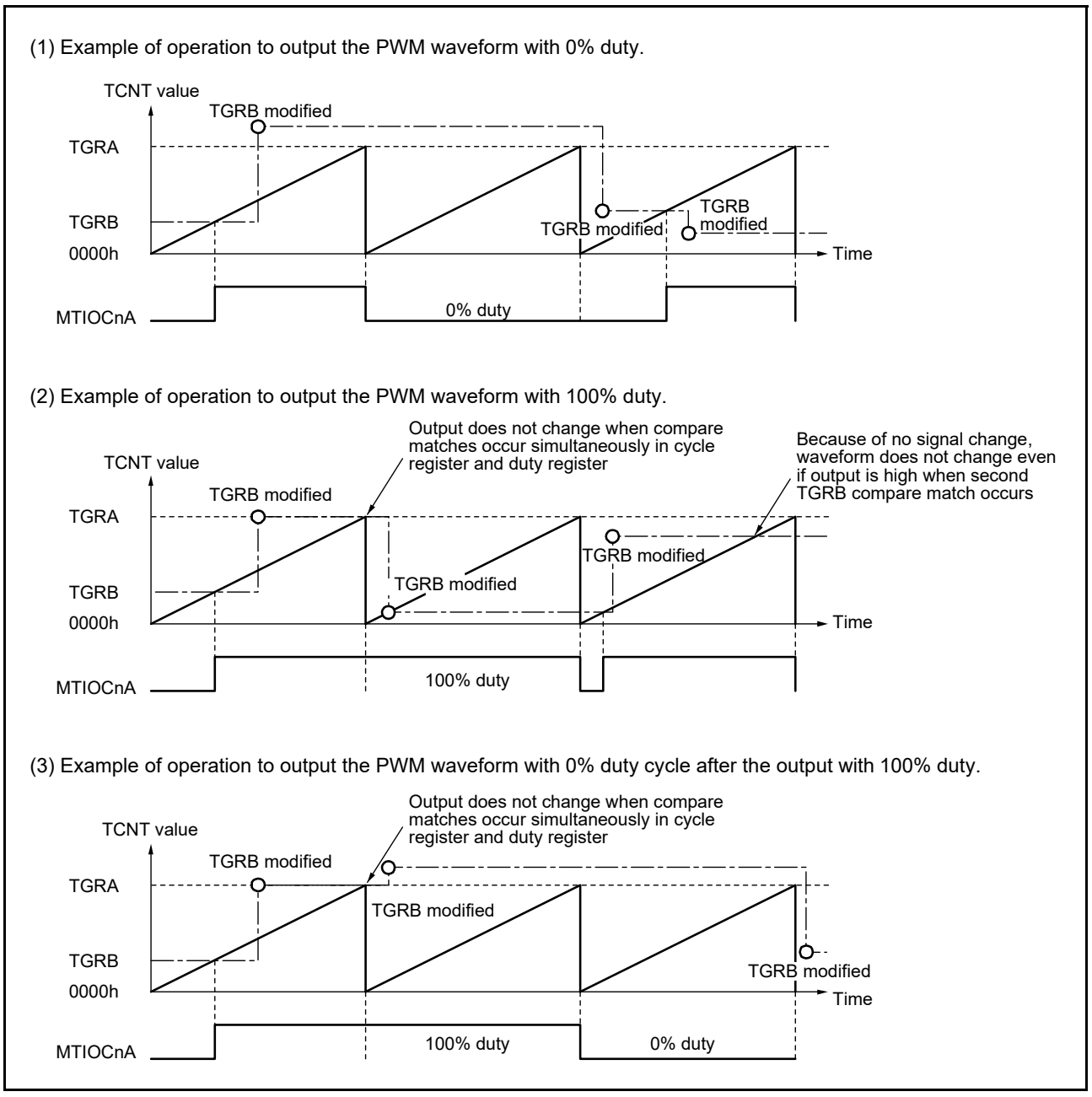


Figure 23.28 Examples of PWM Mode Operation (PWM Waveform Output with 0% Duty and 100% Duty) (n = 0 to 4)

Downloaded from Arrow.com.

23.3.6 Phase Counting Mode

When phase counting mode is specified, an external clock is selected as the count clock and the TCNT counter operates as an up-counter/down-counter regardless of the setting of the TCR.TPSC[2:0] bits and TCR.CKEG[1:0] bits. However, the functions of the TCR.CCLR[2:0] bits and of registers TIOR, TIER, and TGR are valid, and input capture/compare match and interrupt functions can be used.
This can be used for 2-phase encoder pulse input.
If an overflow occurs while the TCNT counter is counting up, a TCIV interrupt is generated while the corresponding TIER.TCIEV bit is 1. If an underflow occurs while the TCNT counter is counting down, a TCIU interrupt is generated while the corresponding TIER.TCIEU bit is 1.
The TSR.TCFD flag is the count direction flag. Read the TCFD flag to check whether the TCNT counter is counting up or down.
In phase counting mode, the external clock pins MTCLKA, MTCLKB, MTCLKC, and MTCLKD can be used as 2-phase encoder pulse input pins. Table 23.44 lists the correspondence between external clock pins and channels.

Table 23.44 Clock Input Pins in Phase Counting Mode

	External Clock Input Pins	
Channel	**A-Phase**	**B-Phase**
MTU1	MTCLKA	MTCLKB
MTU2	MTCLKC	MTCLKD

(1) Example of Phase Counting Mode Setting Procedure

Figure 23.29 shows an example of the phase counting mode setting procedure.

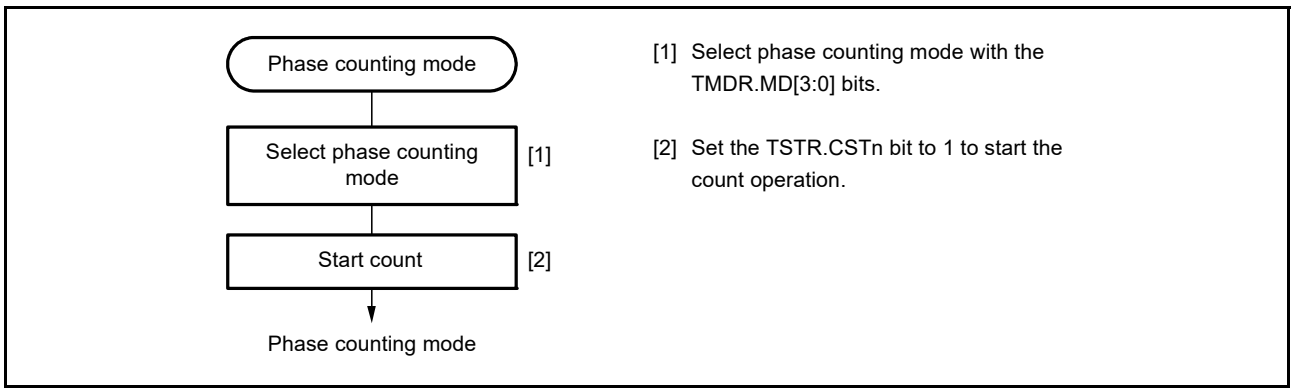


Figure 23.29 Example of Phase Counting Mode Setting Procedure

Downloaded from Arrow.com.

(2) Examples of Phase Counting Mode Operation

In phase counting mode, the TCNT counter is incremented or decremented according to the phase difference between two external clocks. There are four modes according to the count conditions.

(a) Phase Counting Mode 1

Figure 23.30 shows an example of operation in phase counting mode 1, and Table 23.45 lists the TCNT up-counting and down-counting conditions.

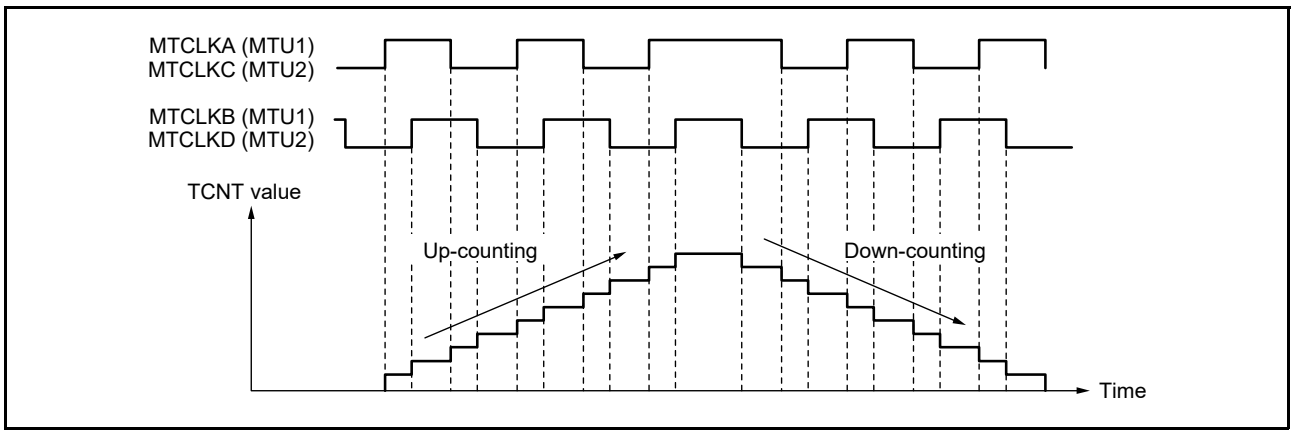


Figure 23.30 Example of Operation in Phase Counting Mode 1

Table 23.45 Up-Counting and Down-Counting Conditions in Phase Counting Mode 1

MTCLKA (MTU1) MTCLKC (MTU2)	MTCLKB (MTU1) MTCLKD (MTU2)	Operation
High	↑	Up-counting
Low	↓	
↑	Low	
↓	High	
High	↓	Down-counting
Low	↑	
↑	High	
↓	Low	

↑ : Rising edge

↓ : Falling edge

Downloaded from Arrow.com.

(b) Phase Counting Mode 2

Figure 23.31 shows an example of operation in phase counting mode 2, and Table 23.46 lists the TCNT up-counting and down-counting conditions.

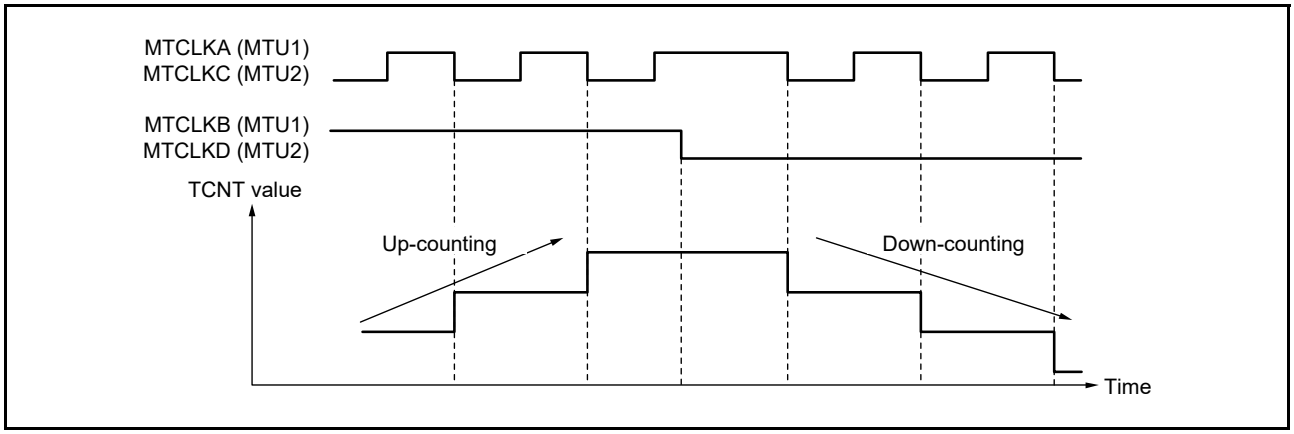


Figure 23.31 Example of Operation in Phase Counting Mode 2

Table 23.46 Up-Counting and Down-Counting Conditions in Phase Counting Mode 2

MTCLKA (MTU1) MTCLKC (MTU2)	MTCLKB (MTU1) MTCLKD (MTU2)	Operation
High	↑	None (Don't care)
Low	↓	None (Don't care)
↑	Low	None (Don't care)
↓	High	Up-counting
High	↓	None (Don't care)
Low	↑	None (Don't care)
↑	High	None (Don't care)
↓	Low	Down-counting

↑ : Rising edge

↓ : Falling edge

Downloaded from Arrow.com.

(c) Phase Counting Mode 3

Figure 23.32 shows an example of operation in phase counting mode 3, and Table 23.47 lists the TCNT up-counting and down-counting conditions.

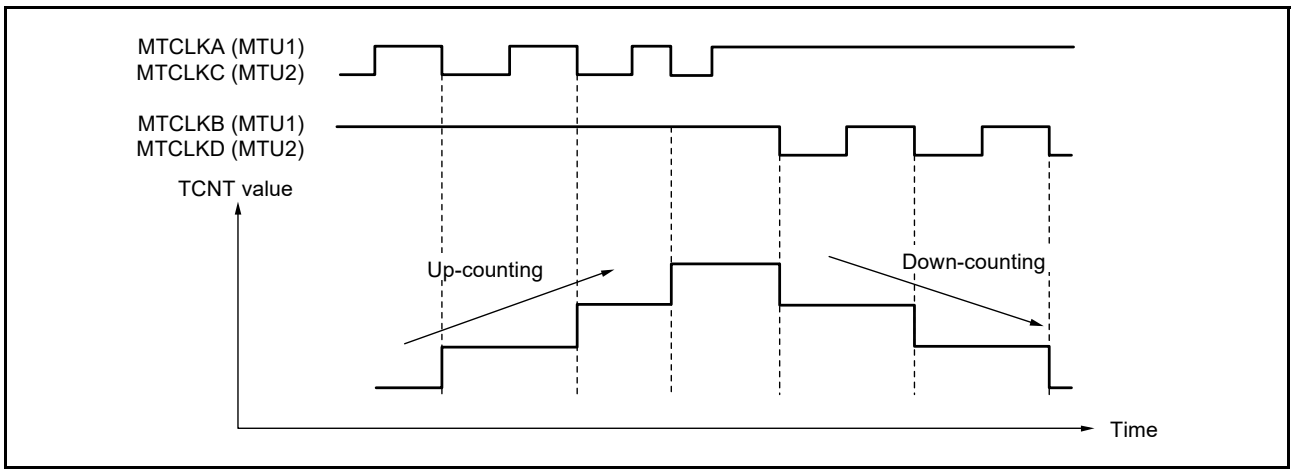


Figure 23.32 Example of Operation in Phase Counting Mode 3

Table 23.47 Up-Counting and Down-Counting Conditions in Phase Counting Mode 3

MTCLKA (MTU1) MTCLKC (MTU2)	MTCLKB (MTU1) MTCLKD (MTU2)	Operation
High	↑	None (Don't care)
Low	↓	None (Don't care)
↑	Low	None (Don't care)
↓	High	Up-counting
High	↓	Down-counting
Low	↑	None (Don't care)
↑	High	None (Don't care)
↓	Low	None (Don't care)

↑ : Rising edge

↓ : Falling edge

Downloaded from Arrow.com.

(d) Phase Counting Mode 4

Figure 23.33 shows an example of operation in phase counting mode 4, and Table 23.48 lists the TCNT up-counting and down-counting conditions.

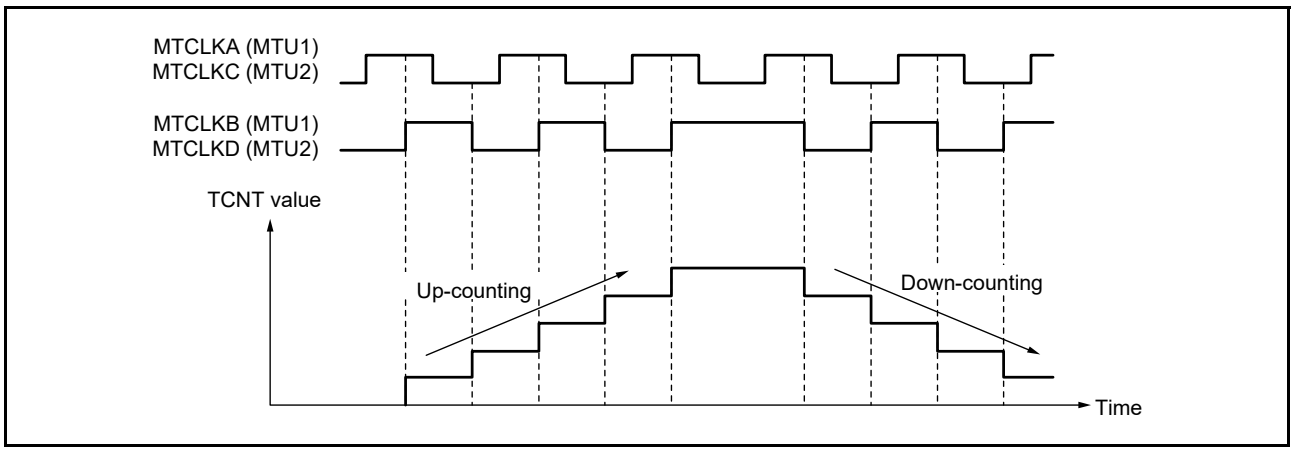


Figure 23.33 Example of Operation in Phase Counting Mode 4

Table 23.48 Up-Counting and Down-Counting Conditions in Phase Counting Mode 4

MTCLKA (MTU1) MTCLKC (MTU2)	MTCLKB (MTU1) MTCLKD (MTU2)	Operation
High	↑	Up-counting
Low	↓	
↑	Low	None (Don't care)
↓	High	
High	↓	Down-counting
Low	↑	
↑	High	None (Don't care)
↓	Low	

↑ : Rising edge

↓ : Falling edge

Downloaded from Arrow.com.

(3) Phase Counting Mode Application Example

Figure 23.34 shows an example in which MTU1 is in phase counting mode, and MTU1 is coupled with MTU0 to input 2-phase encoder pulses of a servo motor in order to detect position or speed.

MTU1 is set to phase counting mode 1, and the encoder pulse A-phase and B-phase are input to MTCLKA and MTCLKB.

MTU0.TGRC compare match is specified as the MTU0.TCNT clearing source and registers MTU0.TGRA and MTU0.TGRC are used for the compare match function and are set with the speed control cycle and position control cycle. The MTU0.TGRB register is used for input capture, with registers MTU0.TGRB and MTU0.TGRD operating in buffer mode. The MTU1 count clock is designated as the MTU0.TGRB input capture source, and the widths of 2-phase encoder 4-multiplication pulses are detected.

Registers MTU1.TGRA and MTU1.TGRB are designated for the input capture function and the MTU0.TGRA and MTU0.TGRC compare matches are selected as the input capture sources to store the up-counter/down-counter values for the control cycles.

This procedure enables the accurate detection of position and speed.

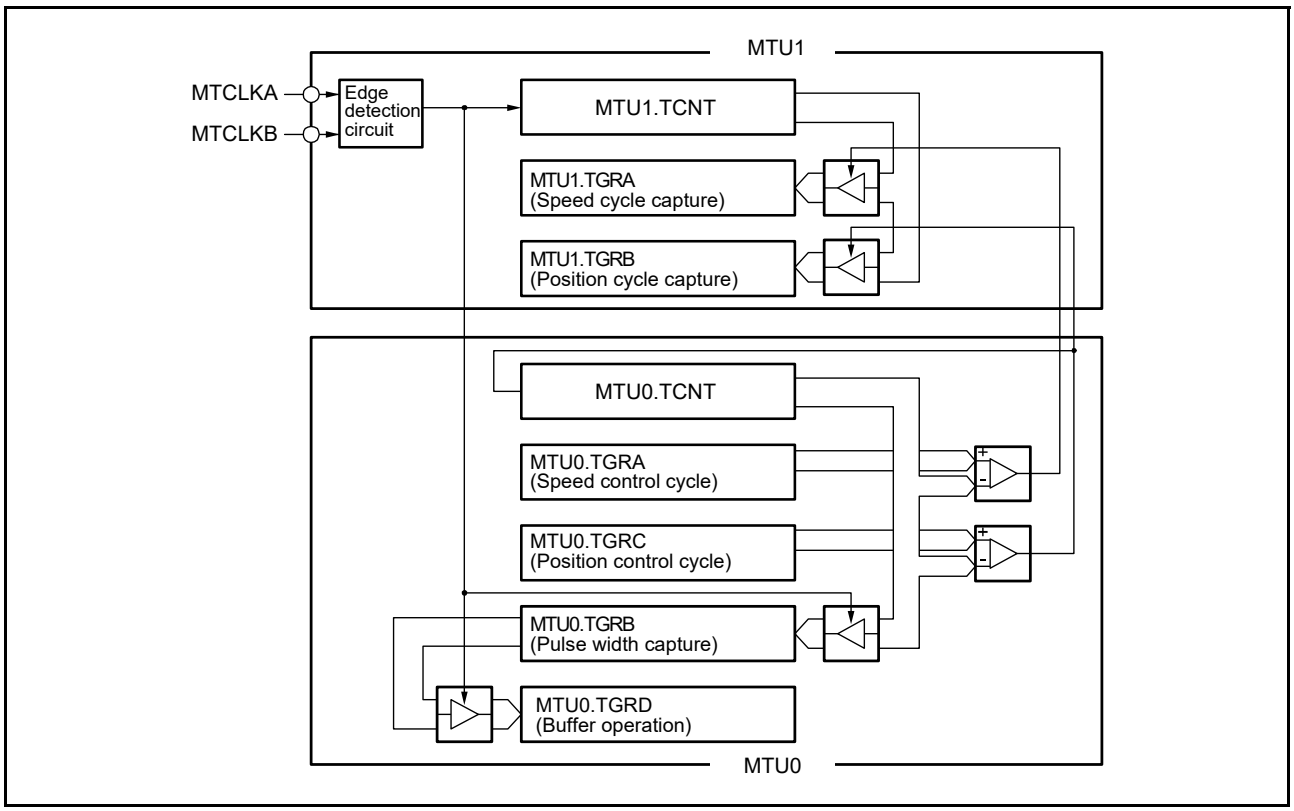


Figure 23.34 Phase Counting Mode Application Example

Downloaded from Arrow.com.

23.3.7 Reset-Synchronized PWM Mode

In the reset-synchronized PWM mode, six phases of positive and negative PWM waveforms that share a common wave transition point can be output by combining MTU3 and MTU4.
When set for reset-synchronized PWM mode, the MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4C, MTIOC4B, and MTIOC4D pins function as PWM output pins and the MTU3.TCNT counter functions as an up-counter.
Table 23.49 lists the PWM output pins. Table 23.50 lists the settings of the registers.

Table 23.49 Output Pins for Reset-Synchronized PWM Mode

Channel	Output Pin	Description
MTU3	MTIOC3B	PWM output pin 1
	MTIOC3D	PWM output pin 1' (negative-phase waveform of PWM output 1)
MTU4	MTIOC4A	PWM output pin 2
	MTIOC4C	PWM output pin 2' (negative-phase waveform of PWM output 2)
	MTIOC4B	PWM output pin 3
	MTIOC4D	PWM output pin 3' (negative-phase waveform of PWM output 3)

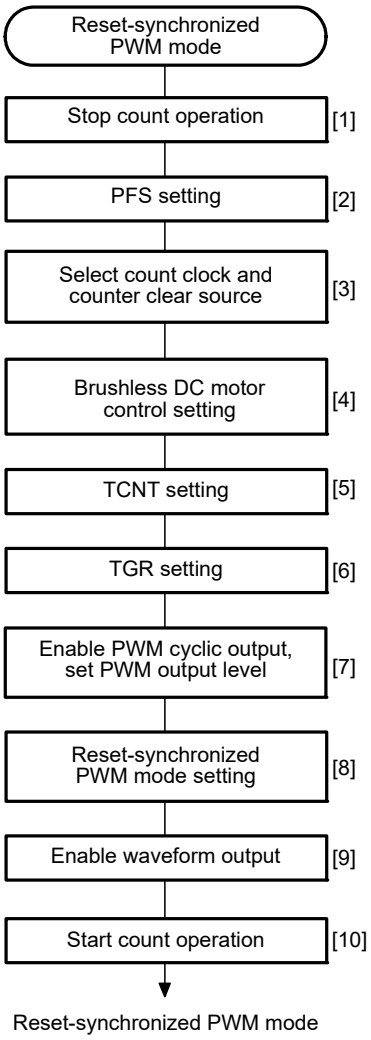
Table 23.50 Register Settings for Reset-Synchronized PWM Mode

Register	Setting
MTU3.TCNT	Initial setting (0000h)
MTU4.TCNT	Initial setting (0000h)
MTU3.TGRA	Set the count cycle for MTU3.TCNT
MTU3.TGRB	Set the transition point of the PWM waveform to be output from the MTIOC3B and MTIOC3D pins
MTU4.TGRA	Set the transition point of the PWM waveform to be output from the MTIOC4A and MTIOC4C pins
MTU4.TGRB	Set the transition point of the PWM waveform to be output from the MTIOC4B and MTIOC4D pins

Downloaded from Arrow.com.

(1) Example of Procedure for Setting Reset-Synchronized PWM Mode

Figure 23.35 shows an example of procedure for setting the reset-synchronized PWM mode.



[1] Set the CST3 and CST4 bits in the TSTR register to 0 to stop the TCNT operation. Specify the reset-synchronized PWM mode while MTU3.TCNT and MTU4.TCNT are stopped.

[2] Set the Pmn pin function control register and the port I/O register.

[3] Set bits TPSC[2:0] and CKEG[1:0] in the MTU3.TCR register to select the count clock source and clock edge for MTU3. Set the MTU3.TCR.CCLR[2:0] bits to select TGRA compare-match as a counter clear source.

[4] When performing brushless DC motor control, set the TGCR.BDC bit and set the feedback signal input source and output chopping or gate signal direct output.

[5] Set counters MTU3.TCNT and MTU4.TCNT to 0000h.

[6] The MTU3.TGRA register is the cycle register. Set the waveform cycle value in the MTU3.TGRA register. Set the transition timing of the PWM output waveforms in registers MTU3.TGRB, MTU4.TGRA, and MTU4.TGRB. Set times within the compare match range of the MTU3.TCNT counter.

X ≤ MTU3.TGRA (X: Setting value)

[7] Enable or disable toggle output synchronized with the PWM cycle using the TOCR1.PSYE bit, and set the PWM output level with bits OLSP and OLSN. When specifying the PWM output Level. In buffer operation of registers TOCR2 and TOLBR, refer to Figure 23.3.

[8] Set the MTU3.TMDR.MD[3:0] bits to 1000b to select the reset-synchronized PWM mode. Set the BFA bit to 1 for buffer operation of registers TGRA and TGRC, and the BFB bit to 1 for buffer operation of registers TGRB and TGRD, if necessary.
Set the initial value in the MTU4.TMDR register.

[9] Set the enabling/disabling of the PWM waveform output pin in the TOER register.

[10] Set the TSTR.CST3 bit to 1 to start the count operation.

Note: The output waveform starts to toggle at the point of MTU3.TCNT = MTU3.TGRA = X by setting X = MTU3.TGRA, i.e., cycle = duty cycle.

Figure 23.35 Procedure for Selecting Reset-Synchronized PWM Mode

Downloaded from Arrow.com.

(2) Example of Reset-Synchronized PWM Mode Operation

Figure 23.36 shows an example of operation in the reset-synchronized PWM mode.
Counters MTU3.TCNT and MTU4.TCNT operate as up-counters. The counters are cleared when a compare match occurs between the MTU3.TCNT counter and the MTU3.TGRA register, and then begin incrementing from 0000h. The output from the PWM pins toggles every time a compare match occurs in registers MTU3.TGRB, MTU4.TGRA, and MTU4.TGRB and the counters are cleared.

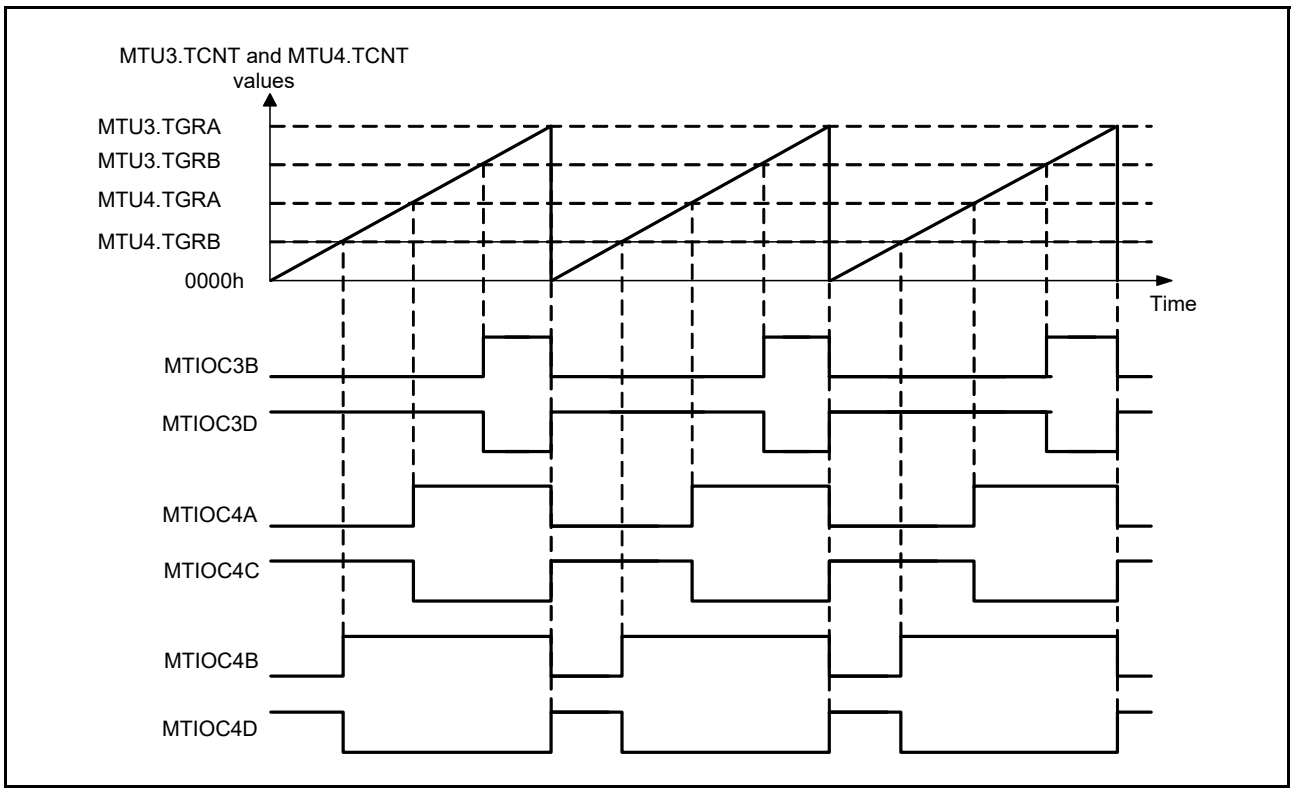


Figure 23.36 Example of Reset-Synchronized PWM Mode Operation (When TOCR1.OLSN = 1 and OLSP = 1)

Downloaded from Arrow.com.

23.3.8 Complementary PWM Mode

In complementary PWM mode, dead time can be set for PWM waveforms to be output. The dead time is the period during which the upper and lower arm transistors are set to the inactive level in order to prevent short-circuiting of the arms. Six phases of positive and negative PWM waveforms with dead time can be output by combining MTU3 and MTU4. PWM waveforms without dead time can also be output.
In complementary PWM mode, MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4B, MTIOC4C, and MTIOC4D pins function as PWM output pins, and the MTIOC3A pin can be set for toggle output synchronized with the PWM cycle. Counters MTU3.TCNT and MTU4.TCNT function as up/down-counters.
Table 23.51 lists the PWM output pins used. Table 23.52 lists the settings of the registers used.
A function to directly cut off the PWM output by using an external signal is supported as a port function.

Table 23.51 Output Pins for Complementary PWM Mode

Channel	Output Pin	Description
MTU3	MTIOC3A	Toggle output synchronized with PWM cycle (or I/O port)
	MTIOC3B	PWM output pin 1
	MTIOC3C	I/O port*1
	MTIOC3D	PWM output pin 1' (negative-phase waveform output of PWM output 1)
MTU4	MTIOC4A	PWM output pin 2
	MTIOC4C	PWM output pin 2' (negative-phase waveform output of PWM output 2)
	MTIOC4B	PWM output pin 3
	MTIOC4D	PWM output pin 3' (negative-phase waveform output of PWM output 3)

Note 1. Avoid setting the MTIOC3C pin as a timer I/O pin in complementary PWM mode.

Table 23.52 Register Settings for Complementary PWM Mode

Channel	Counter/ Register	Description	Read/Write from CPU
MTU3	MTU3.TCNT	Starts up-counting from the value set in the dead time register	Maskable by TRWER setting*1
	MTU3.TGRA	Set MTU3.TCNT upper limit value (1/2 carrier cycle + dead time)	Maskable by TRWER setting*1
	MTU3.TGRB	PWM output 1 compare register	Maskable by TRWER setting*1
	MTU3.TGRC	MTU3.TGRA buffer register	Readable/writable
	MTU3.TGRD	PWM output 1/MTU3.TGRB buffer register	Readable/writable
MTU4	MTU4.TCNT	Starts up-counting after being initialized to 0000h	Maskable by TRWER setting*1
	MTU4.TGRA	PWM output 2 compare register	Maskable by TRWER setting*1
	MTU4.TGRB	PWM output 3 compare register	Maskable by TRWER setting*1
	MTU4.TGRC	PWM output 2/MTU4.TGRA buffer register	Readable/writable
	MTU4.TGRD	PWM output 3/MTU4.TGRB buffer register	Readable/writable
Timer dead time data register (TDDR)		Set MTU4.TCNT and MTU3.TCNT offset value (dead time value)	Maskable by TRWER setting*1
Timer cycle data register (TCDR)		Set MTU4.TCNT upper limit value (1/2 carrier cycle)	Maskable by TRWER setting*1
Timer cycle buffer register (TCBR)		TCDR buffer register	Readable/writable
Subcounter (TCNTS)		Subcounter for dead time generation	Read-only
Temporary register 1 (TEMP1)		PWM output 1/MTU3.TGRB temporary register	Not readable/writable
Temporary register 2 (TEMP2)		PWM output 2/MTU4.TGRA temporary register	Not readable/writable
Temporary register 3 (TEMP3)		PWM output 3/MTU4.TGRB temporary register	Not readable/writable

Note 1. Access can be enabled or disabled according to the setting in the TRWER register (timer read/write enable register).

Downloaded from Arrow.com.

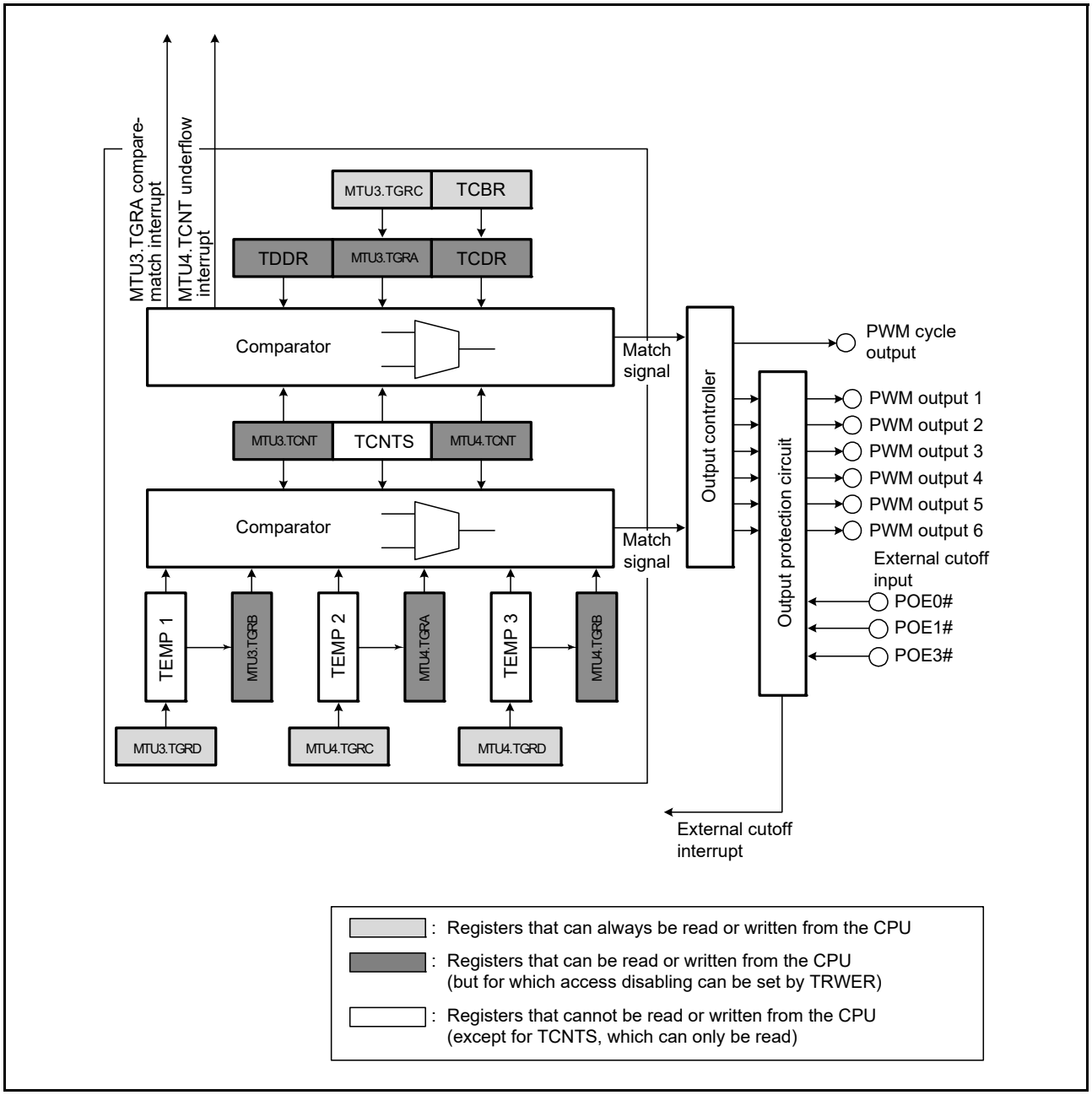
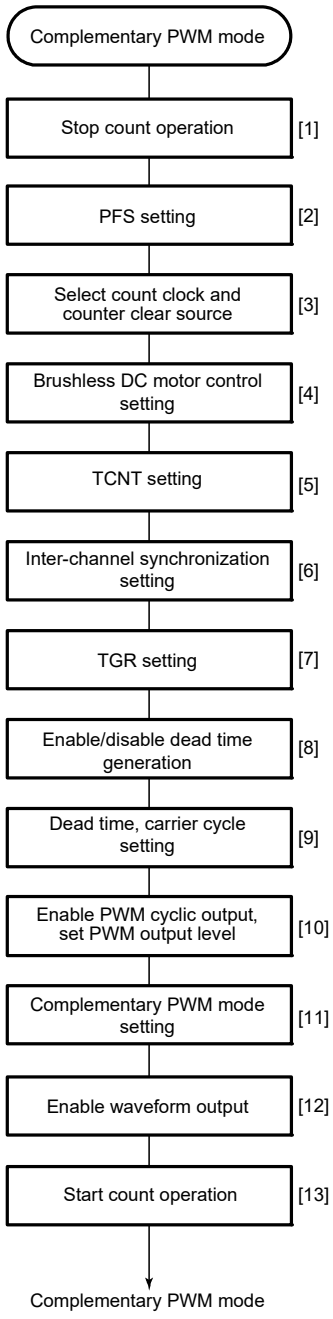


Figure 23.37 Block Diagram of MTU3 and MTU4 in Complementary PWM Mode

Downloaded from Arrow.com.

(1) Example of Complementary PWM Mode Setting Procedure

Figure 23.38 shows an example of the complementary PWM mode setting procedure.



[1] Set bits CST3 and CST4 in the TSTR register to 0 to stop TCNT operation. Specify complementary PWM mode while counters MTU3.TCNT and MTU4.TCNT are stopped.

[2] Set the Pmn pin function control register and the port I/O register.

[3] Set the same count clock source and clock edge for MTU3 and MTU4 with bits TPSC[2:0] and bits CKEG[1:0] in the TCR register. Use the CCLR[2:0] bits to set synchronous clearing only when restarting by synchronous clearing with another channel during complementary PWM mode operation.

[4] When performing brushless DC motor control, set TGCR.BDC bit and set the feedback signal input source and output chopping or gate signal direct output.

[5] Set the dead time in the MTU3.TCNT counter. Set the MTU4.TCNT counter to 0000h.

[6] Set only when restarting by synchronous clearing with another channel during complementary PWM mode operation. In this case, synchronize the channel generating the synchronous clear with MTU3 and MTU4 using the TSYR register.

[7] Set the output PWM duty cycles in the compare registers (MTU3.TGRB, MTU4.TGRA, and MTU4.TGRB) and buffer registers (MTU3.TGRD, MTU4.TGRC, and MTU4.TGRD). Set the same value in the buffer registers and the compare registers.

[8] This setting is necessary only when no dead time should be generated. Make appropriate settings in the TDER register so that no dead time is generated.

[9] Set the dead time in the TDDR register, 1/2 the carrier cycle in the TCDR register and TCBR register, and 1/2 the carrier cycle plus the dead time in registers MTU3.TGRA and MTU3.TGRC. When no dead time generation is selected, set 1 in the TDDR register and 1/2 the carrier cycle + 1 in registers MTU3.TGRA and MTU3.TGRC.

[10] Enable or disable toggle output synchronized with the TOCR1.PSYE bit, and set the PWM output level with bits OLSP and OLSN.
When specifying the PWM output level by using the TOLBR register as a buffer for the MTU.TOCR2 register, refer to Figure 23.3, Example of PWM Output Level Setting Procedure in Buffer Operation.

[11] Select complementary PWM mode in the MTU3.TMDR register. Set the BFA bit to 1 for buffer operation of registers TGRA and TGRC, and the BFB bit to 1 for buffer operation of registers TGRB and TGRD, if necessary. Do not set the MTU4.TMDR register.

[12] Set enabling/disabling of PWM waveform output pin output in the TOER register.

[13] Set bits CST3 and CST4 in the TSTR register to 1 simultaneously to start the count operation.

Figure 23.38 Example of Complementary PWM Mode Setting Procedure

Downloaded from Arrow.com.

(2) Outline of Complementary PWM Mode Operation

In complementary PWM mode, six phases (three positive and three negative) of PWM waveforms can be output. Figure 23.39 illustrates counter operation in complementary PWM mode, and Figure 23.40 shows an example of operation in complementary PWM mode.

(a) Counter Operation

In complementary PWM mode, three counters —MTU3.TCNT, MTU4.TCNT, and TCNTS— perform up-/down-count operations.

The MTU3.TCNT counter is automatically initialized to the value set in the TDDR register when complementary PWM mode is selected and the TSTR.CST3 bit is 0.

When the CST3 bit is set to 1, the MTU3.TCNT counter counts up to the value set in the MTU3.TGRA register, then switches to down-counting when it matches the MTU3.TGRA register. When the MTU3.TCNT value matches the TDDR register, the counter switches to up-counting, and the operation is repeated in this way.

The MTU4.TCNT counter should be initialized to 0000h.

When the CST4 bit is set to 1, the MTU4.TCNT counter counts up in synchronization with the MTU3.TCNT counter, and switches to down-counting when it matches the TCDR register. On reaching 0000h, the MTU4.TCNT counter switches to up-counting, and the operation is repeated in this way.

The TCNTS counter is a read-only counter. It does not need to be initialized.

When the MTU3.TCNT counter matches the TCDR register during up-/down-counting of the TCNT counter in MTU3 and MTU4, the TCNTS counter starts down-counting, and when the TCNTS counter matches the TCDR register, the operation switches to up-counting. When the TCNTS counter matches the MTU3.TGRA register, it becomes 0000h.

When the MTU4.TCNT counter matches the TDDR register during down-counting of counters MTU3.TCNT and MTU4.TCNT, the TCNTS counter starts up-counting, and when the TCNTS counter matches the TDDR register, the operation switches to down-counting. When the TCNTS counter reaches 0000h, it is set with the value in the MTU3.TGRA register.

The TCNTS counter is compared with the compare register and temporary register, in which the PWM duty is specified, only during the count operation.

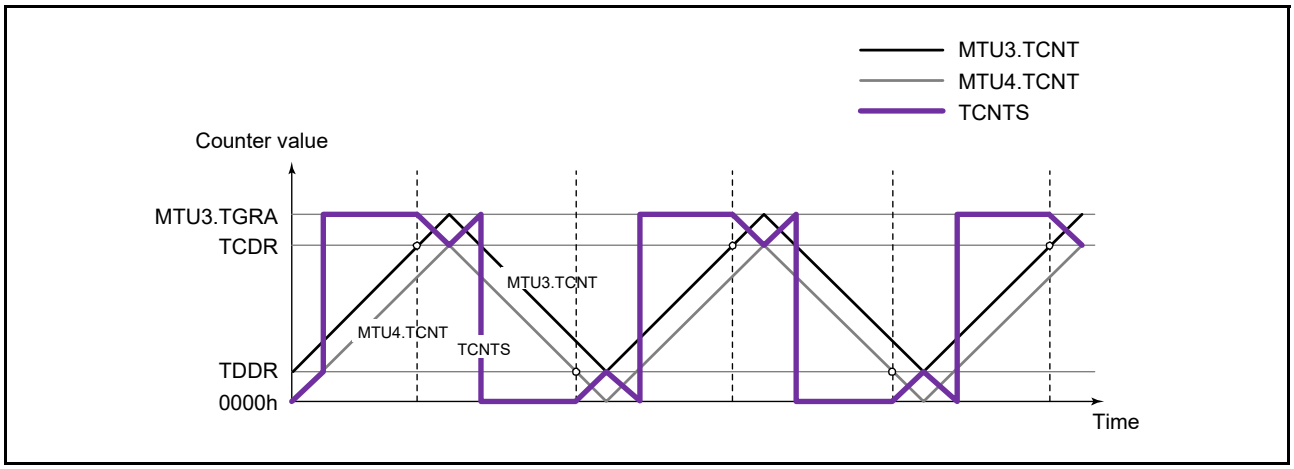


Figure 23.39 Counter Operation in Complementary PWM Mode

Downloaded from Arrow.com.

(b) Register Operation

In complementary PWM mode, nine registers (compare registers, buffer registers, and temporary registers) are used to control the duty ratio for the PWM output. Figure 23.40 shows an example of operation in complementary PWM mode.
Registers MTU3.TGRB, MTU4.TGRA, and MTU4.TGRB are constantly compared with the counters to generate PWM waveforms. When these registers match the counter, the value set in the TOCR1.OLSN and OLSP bits is output from the PWM output pin.
Registers MTU3.TGRD, MTU4.TGRC, and MTU4.TGRD are buffer registers for these compare registers. Between a buffer register and a compare register, there is a temporary register. The temporary registers cannot be accessed by the CPU.
Data in a compare register can be changed by writing new data to the corresponding buffer register. The buffer registers can be read or written at any time.
When modifying data in a buffer register, be sure to write to the MTU4.TGRD register last and enable data transfer from the buffer register to a temporary register. At this time, transfer from registers TCBR and MTU3.TGRC, which operate as buffer registers for the timer cycle registers, to temporary registers is also enabled. Data is transferred to all five temporary registers at the same time. When transfer is enabled in the Ta interval, data written to a buffer register is immediately transferred to the temporary register. Data is not transferred to the temporary register in the Tb1 and Tb2 intervals. Data enabled for transfer in this interval is transferred to the temporary register at the end of this interval.
The value transferred to a temporary register is transferred to the compare register at the end of the Tb1 interval (when matches the MTU3.TGRA register while the TCNTS counter is counting up), or at the end of the Tb2 interval (when matches 0000h while the TCNTS counter is counting down). The timing for transfer from the temporary register to the compare register can be selected with the TMDR.MD[3:0] bits. Figure 23.40 shows an example in which the trough is selected for the transfer timing.
In the Tb (Tb2 in Figure 23.40) interval in which data is not transferred to the temporary register, the temporary register has the same function as the compare register and is compared with the counter. In this interval, therefore, there are two compare match registers for one output phase; the compare register contains the pre-change data and the temporary register contains new data. In this interval, three counters (MTU3.TCNT, MTU4.TCNT and TCNTS) and two registers (compare register and temporary register) are compared, and PWM output is controlled accordingly.

Downloaded from Arrow.com.

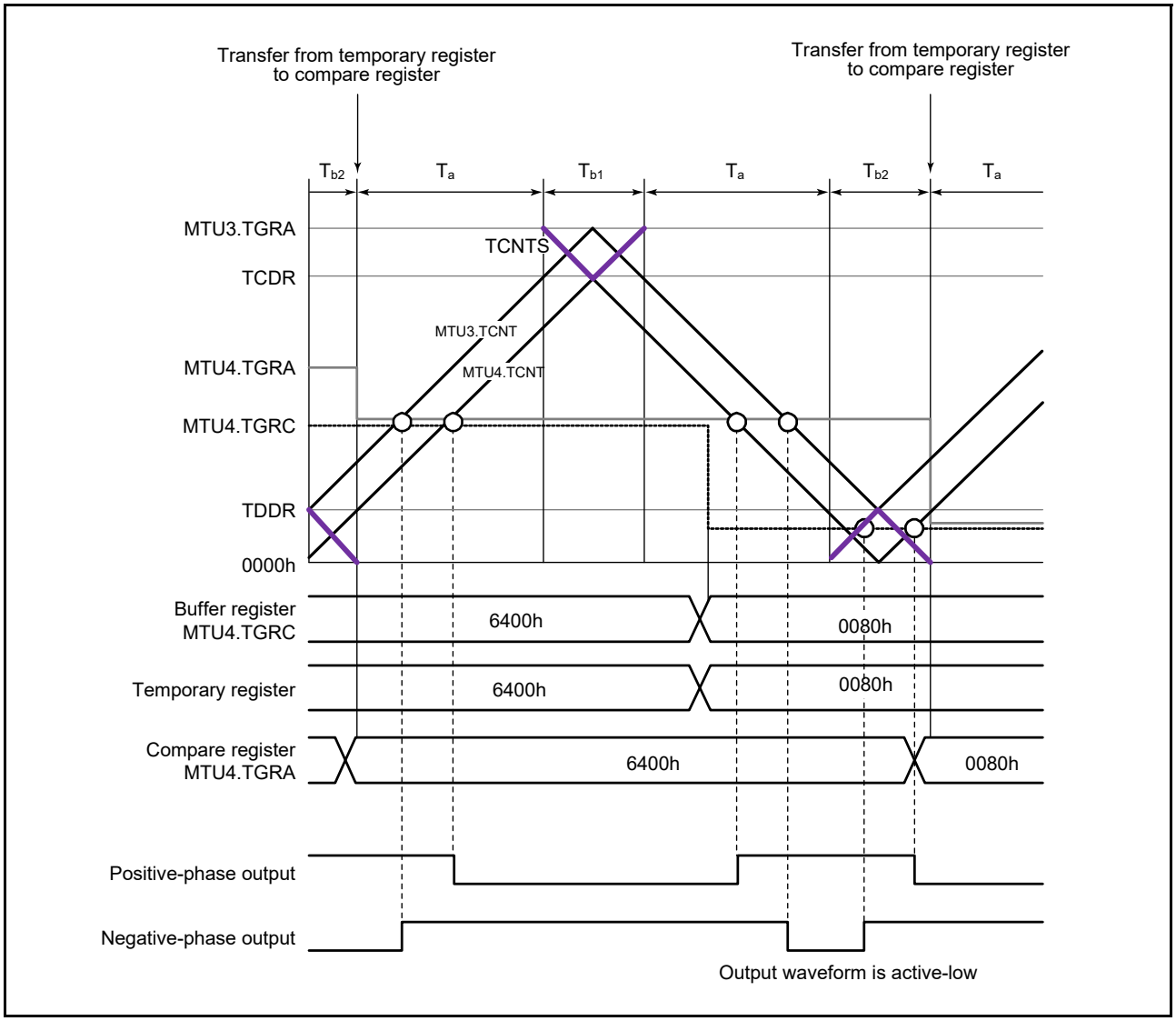


Figure 23.40 Example of Operation in Complementary PWM Mode

Downloaded from Arrow.com.

(c) Initial Setting

In complementary PWM mode, there are six registers that require initial setting. In addition, there is a register that specifies whether to generate dead time (it should be used only when dead time generation should be disabled).
Before setting complementary PWM mode with the TMDR.MD[3:0] bits, initial values should be set in the following registers.
The MTU3.TGRC register operates as the buffer register for the MTU3.TGRA register, and should be set with 1/2 the PWM cycle + dead time Td. The TCBR register operates as the buffer register for the TCDR register, and should be set with 1/2 the PWM cycle. Set dead time Td in the TDDR register.

When dead time is not needed, the TDER.TDER bit should be set to 0, registers MTU3.TGRC and MTU3.TGRA should be set to 1/2 the PWM cycle + 1, and the TDDR register should be set to 1.
Set the respective initial PWM duty values in three buffer registers MTU3.TGRD, MTU4.TGRC, and MTU4.TGRD. The values set in the five buffer registers excluding the TDDR register are transferred to the corresponding compare registers as soon as complementary PWM mode is set.
Set the MTU4.TCNT counter to 0000h before setting complementary PWM mode.

Table 23.53 Registers and Counters Requiring Initial Setting

Register and Counter	Setting
MTU3.TGRC	1/2 PWM cycle + dead time Td (1/2 PWM cycle + 1 when dead time generation is disabled by the TDER register)
TDDR	Dead time Td (1 when dead time generation is disabled by the TDER register)
TCBR	1/2 PWM cycle
MTU3.TGRD, MTU4.TGRC, MTU4.TGRD	Initial PWM duty value for each phase
MTU4.TCNT	0000h

Note: The value set in the MTU3.TGRC register should be the sum of 1/2 the PWM cycle set in the TCBR register and dead time Td set in the TDDR register. When dead time generation is disabled by the TDER register, the TGRC register should be set to 1/2 the PWM cycle + 1.

(d) PWM Output Level Setting

In complementary PWM mode, the PWM output level is set with bits OLSN and OLSP in the TOCR1 register or bits OLS1P to OLS3P and OLS1N to OLS3N in the TOCR2 register.
The output level can be set for each of the three positive phases and three negative phases of 6-phase output.
Complementary PWM mode should be cleared before setting or changing output levels.

(e) Dead Time Setting

In complementary PWM mode, dead time can be set for PWM output.
The dead time is set in the TDDR register. The value set in the TDDR register is used as the MTU3.TCNT counter start value and creates a dead time between counters MTU3.TCNT and MTU4.TCNT. Complementary PWM mode should be cleared before changing the contents of the TDDR register.

(f) Dead Time Suppressing

Dead time generation is suppressed by setting the TDER.TDER bit to 0. The TDER bit can be set to 0 only when 0 is written to it after reading it as 1.
Registers MTU3.TGRA and MTU3.TGRC should be set to 1/2 PWM cycle + 1 and the TDDR register should be set to 1.
By the above settings, PWM waveforms without dead time can be obtained. Figure 23.41 shows an example of operation without dead time.

Downloaded from Arrow.com.

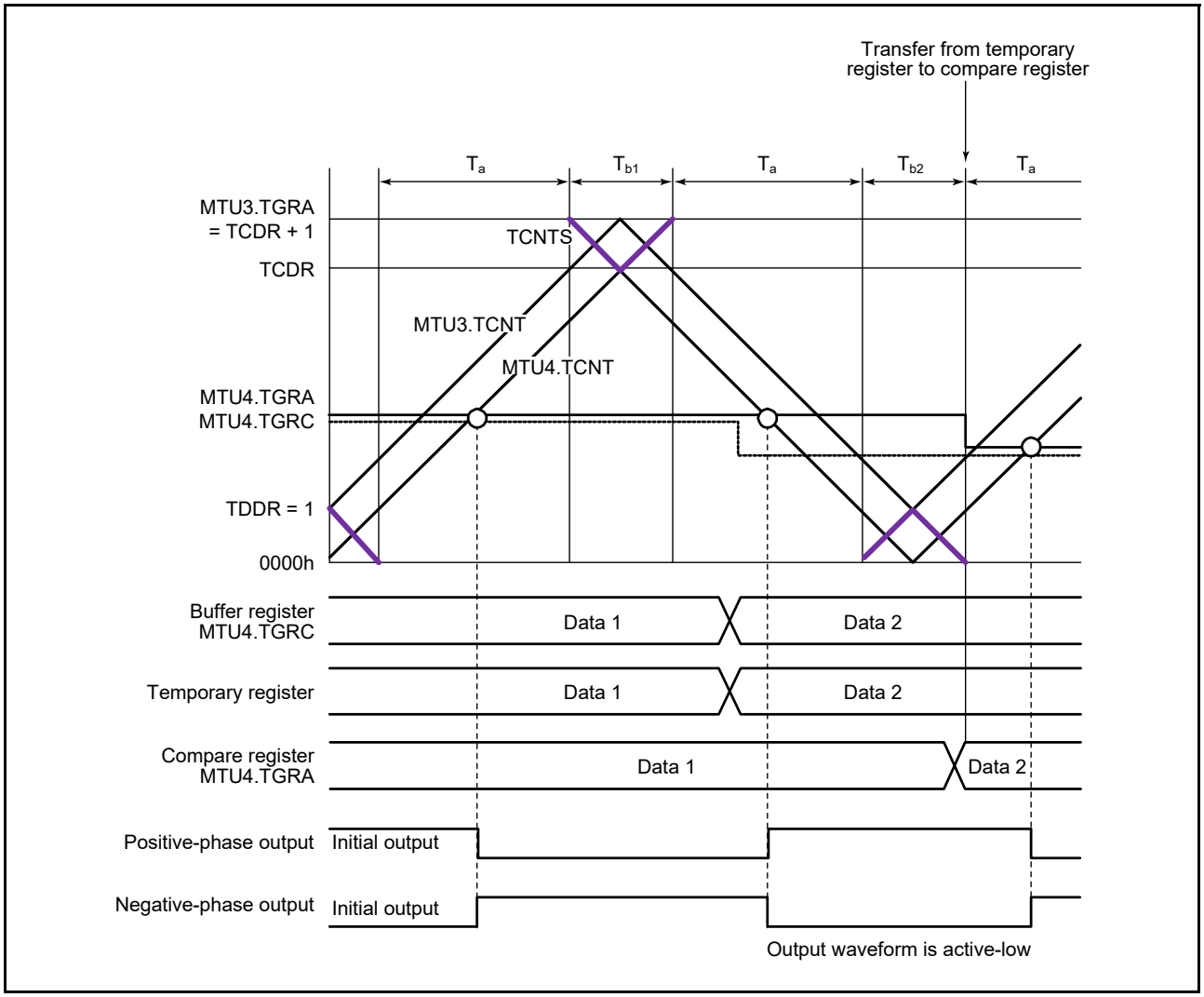


Figure 23.41 Example of Operation without Dead Time

Downloaded from Arrow.com.

(g) PWM Cycle Setting

In complementary PWM mode, the PWM cycle is set in two registers — the MTU3.TGRA register, in which the MTU3.TCNT counter upper limit value is set, and the TCDR register, in which the MTU4.TCNT counter upper limit value is set. The settings should be made so as to achieve the following relationship between these two registers:

With dead time: MTU3.TGRA setting = TCDR setting + TDDR setting
Without dead time: MTU3.TGRA setting = TCDR setting + 1

The settings should be made so as to achieve the following relationship between registers TCDR and TDDR.
TCDR setting > TDDR setting × 2 + 2

The MTU3.TGRA and TCDR settings are made by setting values in buffer registers MTU3.TGRC and TCBR. When data is written to the MTU4.TGRD register to enable transfers, the values set in registers MTU3.TGRC and TCBR are transferred simultaneously to registers MTU3.TGRA and TCDR with the transfer timing selected with the TMDR.MD[3:0] bits.
The new PWM cycle is reflected from the next cycle when data is updated at the crest, or from the current cycle when updated in the trough. Figure 23.42 illustrates the operation when the PWM cycle is updated at the crest.
Refer to the following section (h) Register Data Updating, for the method of updating the data in each buffer register.

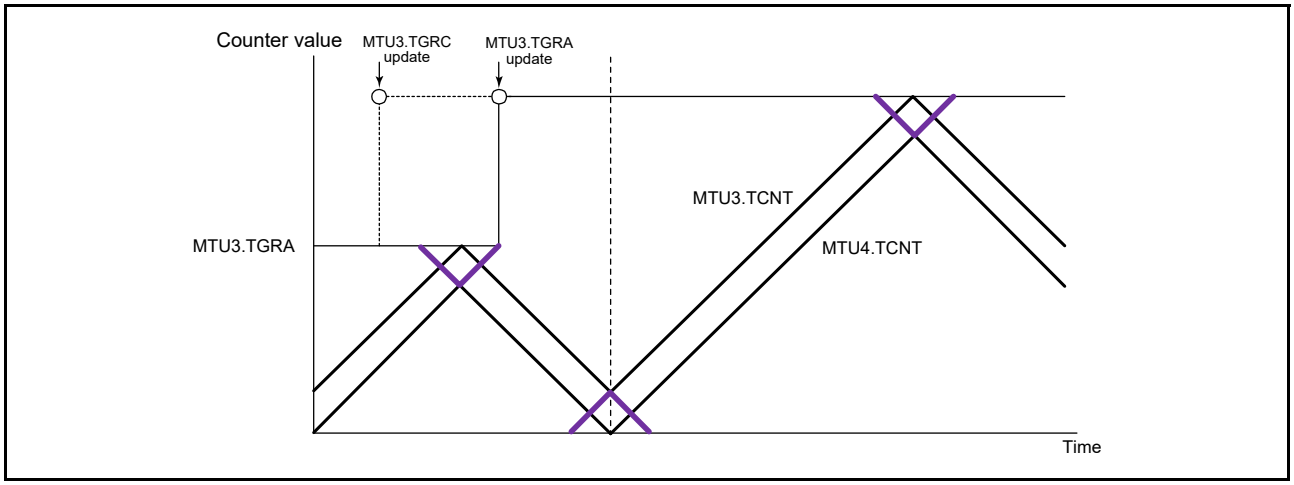


Figure 23.42 Example of PWM Cycle Updating

(h) Register Data Updating

In complementary PWM mode, the buffer register is used to update the data in a compare register. The update data can be written to the buffer register at any time. There are five registers (PWM duty and PWM cycle registers) that have buffer registers and can be updated during operation.
There is a temporary register between each of these registers and its buffer register. While subcounter TCNTS is not counting, if buffer register data is updated, the temporary register value also changes. Data is not transferred from buffer registers to temporary registers while the TCNTS counter is counting; in this case, the value written to a buffer register is transferred after the TCNTS counter halts.
The temporary register value is transferred to the compare register at the data update timing set with the TMDR.MD[3:0] bits. Figure 23.43 shows an example of data updating in complementary PWM mode. This example shows the mode in which data is updated at both the counter crest and trough.
When updating buffer register data, be sure to write to the MTU4.TGRD register at the end of the update. Data is transferred from buffer registers to the temporary registers simultaneously for all five registers after the write to the MTU4.TGRD register.
Even when not updating all five registers or when not updating the MTU4.TGRD data, be sure to write to the

Downloaded from Arrow.com.

MTU4.TGRD register after writing data to the registers to be updated. In this case, the data written to the MTU4.TGRD register should be the same as the data prior to the write operation.

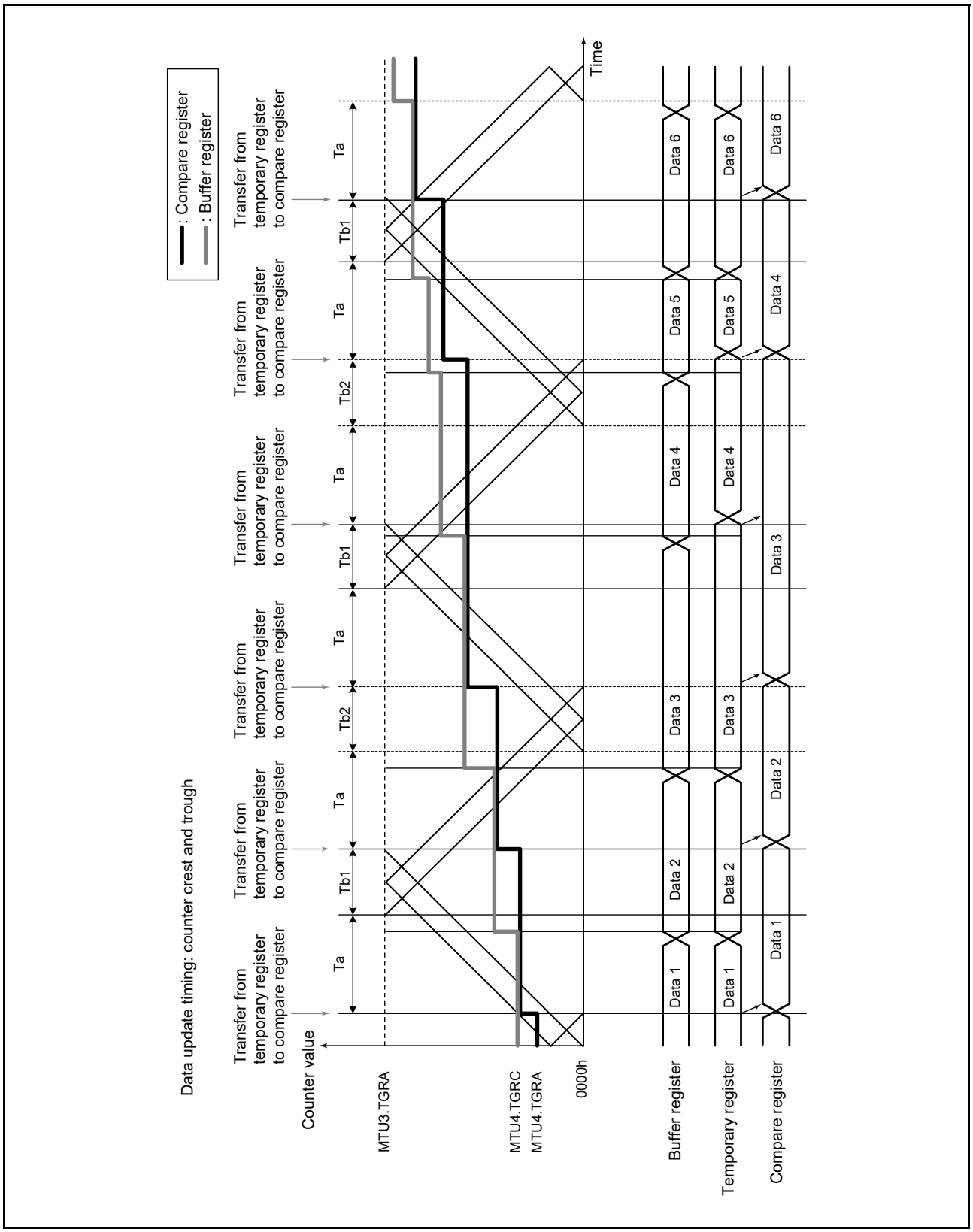


Figure 23.43 Example of Data Updating in Complementary PWM Mode

Downloaded from Arrow.com.

(i) Initial Output in Complementary PWM Mode

In complementary PWM mode, the initial output is determined by the setting of bits OLSN and OLSP in the TOCR1 register or bits OLS1N to OLS3N and OLS1P to OLS3P in the TOCR2 register.
This initial output is the non-active level of the PWM output and continues from when complementary PWM mode is set with the TMDR register until the MTU4.TCNT counter exceeds the value set in the TDDR register. Figure 23.44 shows an example of the initial output in complementary PWM mode.
An example of the waveform when the initial PWM duty value is smaller than the TDDR value is shown in Figure 23.45.

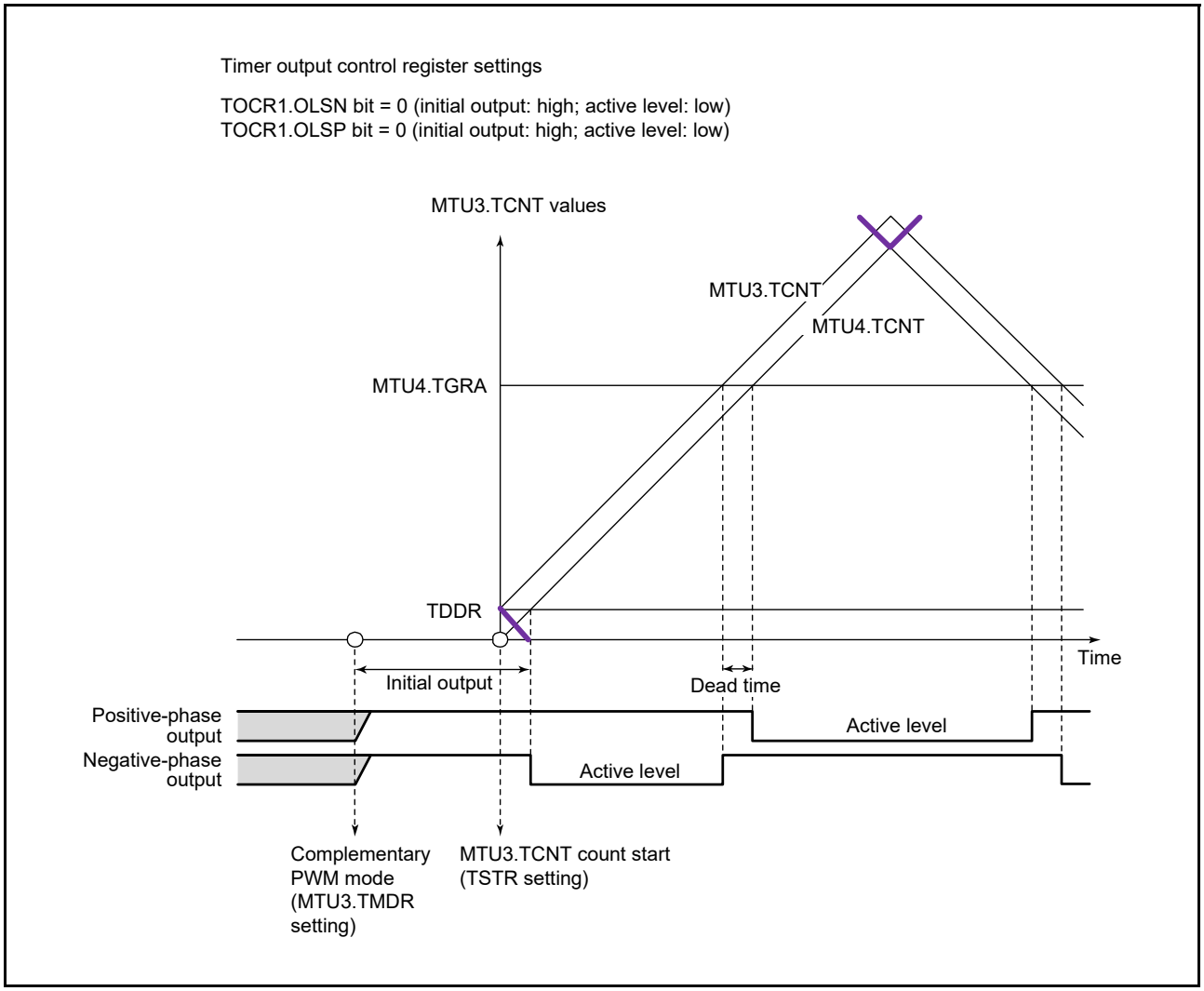


Figure 23.44 Example of Initial Output in Complementary PWM Mode (1)

Downloaded from Arrow.com.

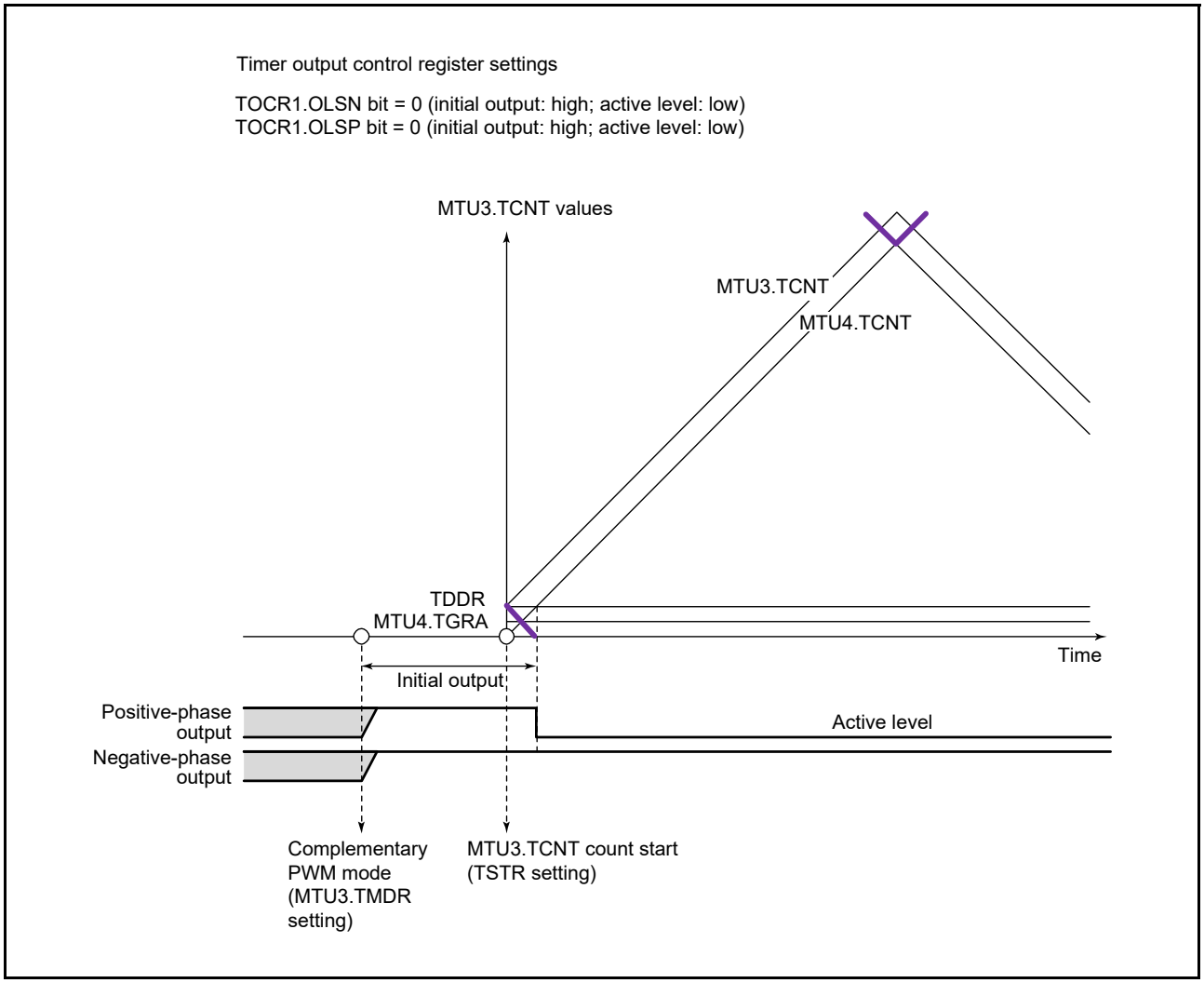


Figure 23.45 Example of Initial Output in Complementary PWM Mode (2)

Downloaded from Arrow.com.

(j) Method for Generating PWM Output in Complementary PWM Mode

In complementary PWM mode, six phases (three positive and three negative) of PWM waveforms can be output. Dead time can be set for PWM waveforms to be output.

A PWM waveform is generated by output of the level selected in the timer output control register in the event of a compare match between a counter and a compare register. While the TCNTS counter is counting, the compare register and temporary register values are simultaneously compared to generate consecutive PWM waveforms from 0 to 100% duty cycle. The relative timing of turn-on and turn-off compare match occurrence may vary, but the compare match that turns off each phase takes precedence to secure the dead time and ensure that the positive-phase and negative-phase turn-on times do not overlap. Figure 23.46 to Figure 23.48 show examples of waveform generation in complementary PWM mode.

The positive-phase and negative-phase turn-off timing is generated by a compare match with the counter indicated by a solid line, and the turn-on timing is generated by a compare match with the counter indicated by a dotted line, which operates with a delay equal to the dead time behind the counter indicated by a solid line. In the T1 period, compare match a that turns off the negative phase has the highest priority, and compare matches before a are ignored. In the T2 period, compare match c that turns off the positive phase has the highest priority, and compare matches before c are ignored.

In most cases, compare matches occur in the order a → b → c → d (or c → d → a' → b') as shown in Figure 23.46.

If compare matches deviate from the a → b → c → d order, since the time for which the negative phase is off is shorter than twice the dead time, the positive phase is not turned on. If compare matches deviate from the c → d → a' → b' order, since the time for which the positive phase is off is shorter than twice the dead time, the negative phase is not turned on.

As shown in Figure 23.47, if compare match c follows compare match a before compare match b, compare match b is ignored and the negative phase is turned on by compare match d. This is because turning off the positive phase has higher priority due to the occurrence of compare match c (positive-phase off timing) before compare match b (positive-phase on timing) (consequently, the waveform does not change because the positive phase goes from off to off).

Similarly, in the example in Figure 23.48, turning off the negative phase has priority due to the occurrence of compare match a' (negative-phase off timing) before compare match d (negative-phase on timing). As a result, the negative phase is not turned on.

Thus, in complementary PWM mode, compare matches at turn-off timings take precedence, and turn-on timing compare matches that occur before a turn-off timing compare match are ignored.

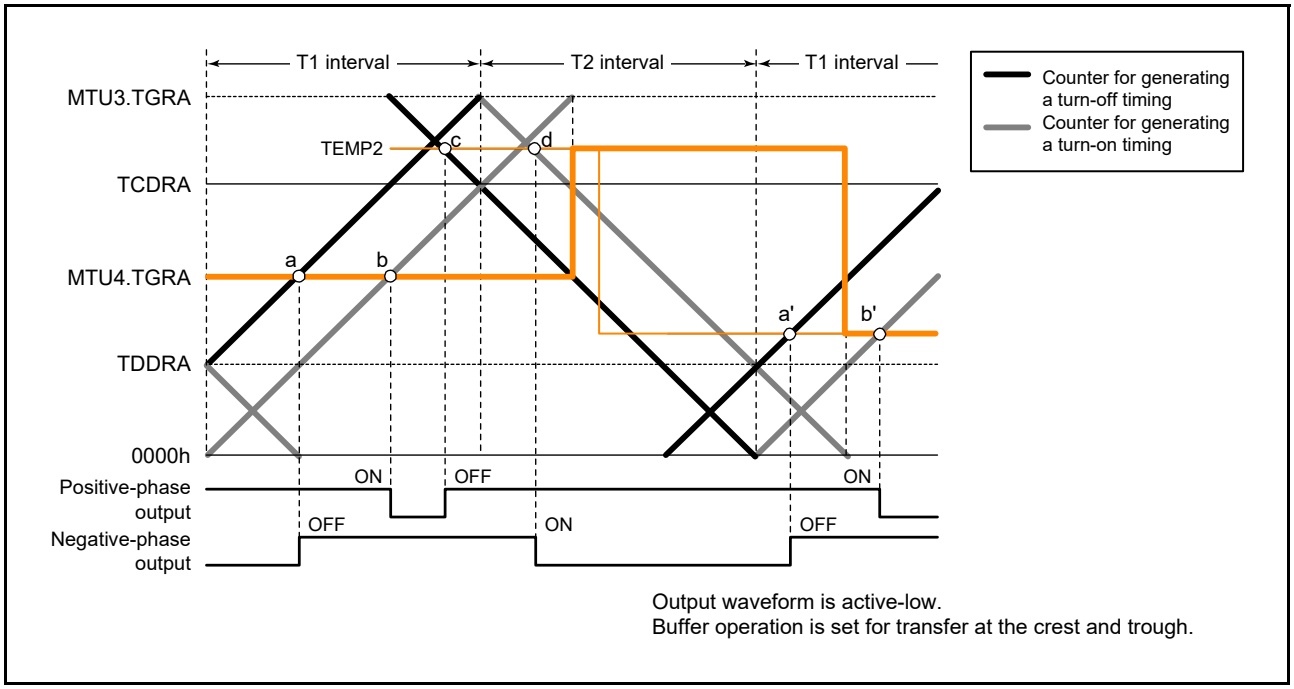


Figure 23.46 Example of Waveform Output in Complementary PWM Mode (1)

Downloaded from Arrow.com.

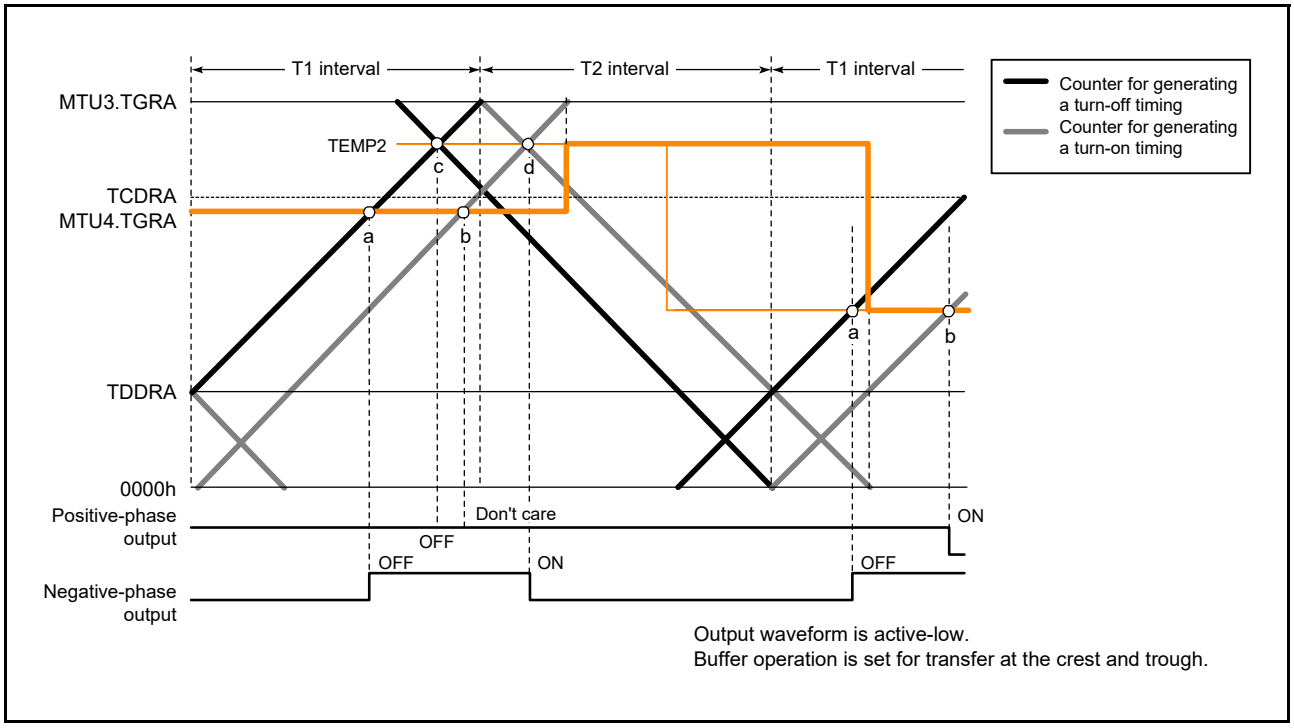


Figure 23.47 Example of Waveform Output in Complementary PWM Mode (2)

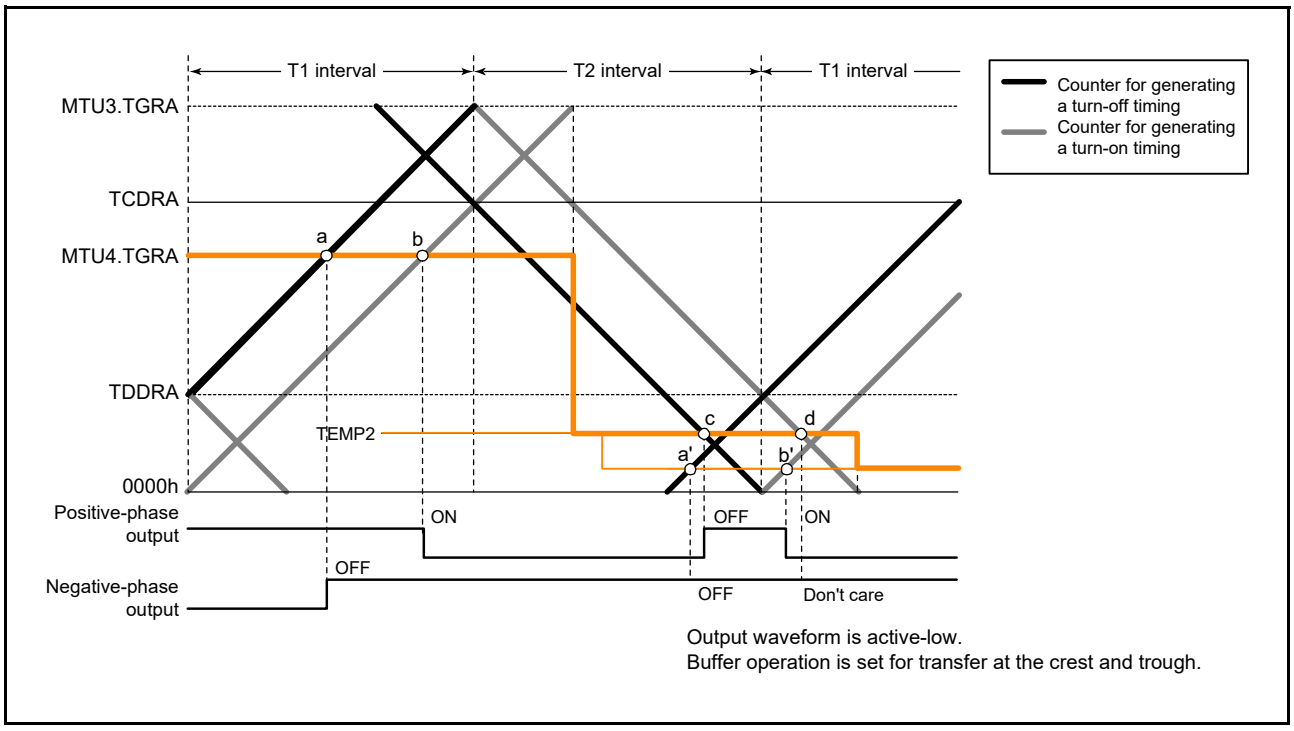


Figure 23.48 Example of Waveform Output in Complementary PWM Mode (3)

Downloaded from Arrow.com.

(k) 0% and 100% Duty Cycle Output in Complementary PWM Mode

In complementary PWM mode, 0% and 100% duty cycle PWM waveforms can be output as required. Figure 23.49 to Figure 23.53 show output examples.

A 100% duty cycle waveform is output when the data register value is set to 0000h. The waveform in this case has a positive phase with a 100% on-state. A 0% duty cycle waveform is output when the data register value is set to the same value as the MTU3.TGRA register. The waveform in this case has a positive phase with a 100% off-state.

On and off compare matches occur simultaneously, but if a turn-on compare match and turn-off compare match for the same phase occur simultaneously, both compare matches are ignored and the waveform does not change.

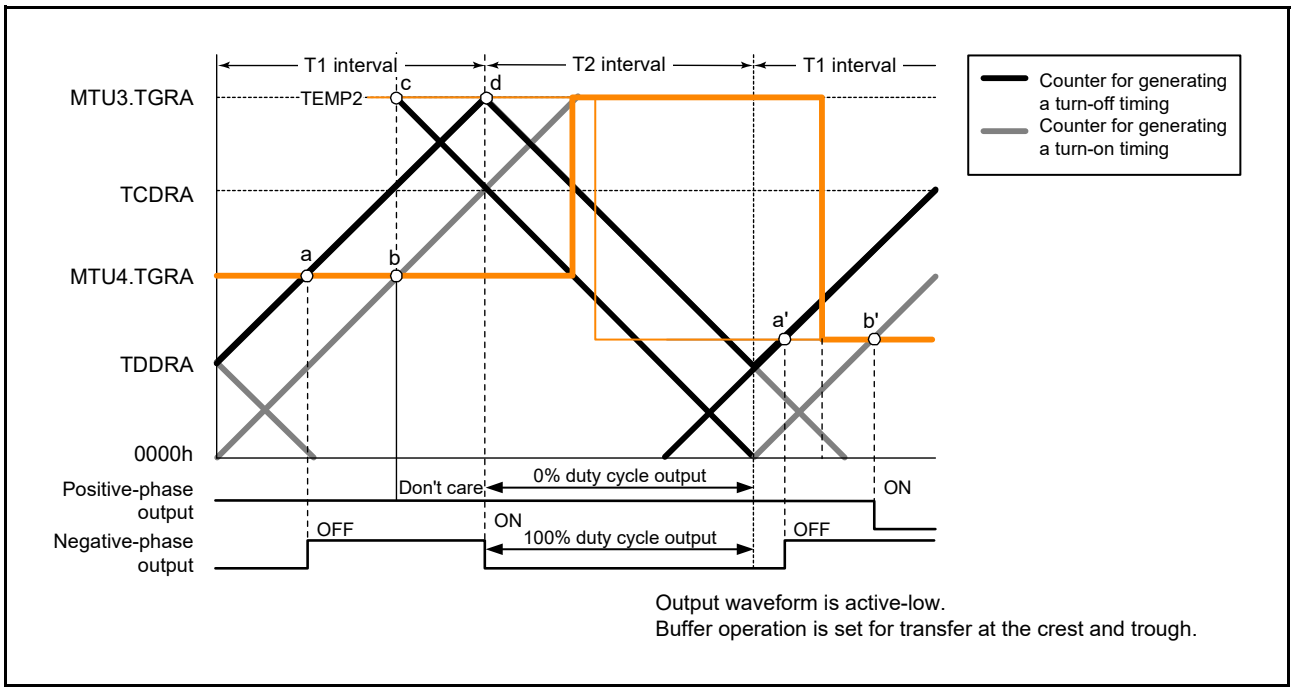


Figure 23.49 Example of 0% and 100% Waveform Output in Complementary PWM Mode (1)

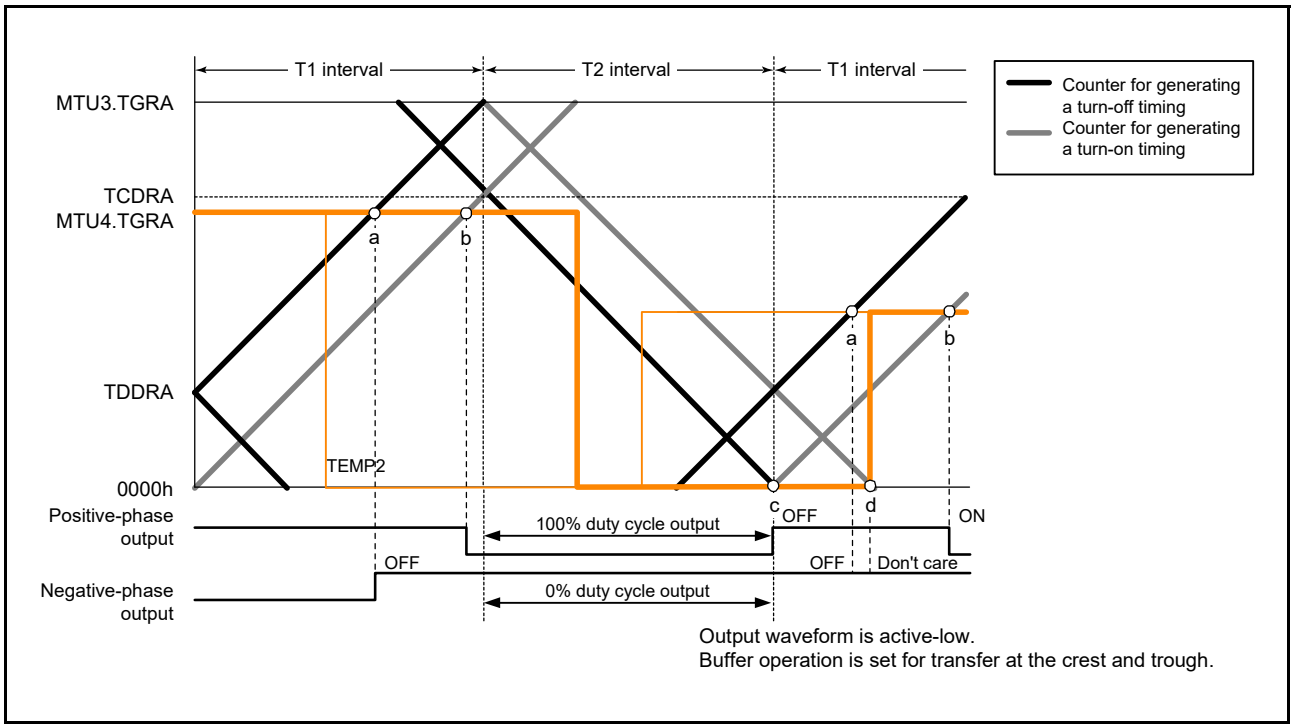


Figure 23.50 Example of 0% and 100% Waveform Output in Complementary PWM Mode (2)

Downloaded from Arrow.com.

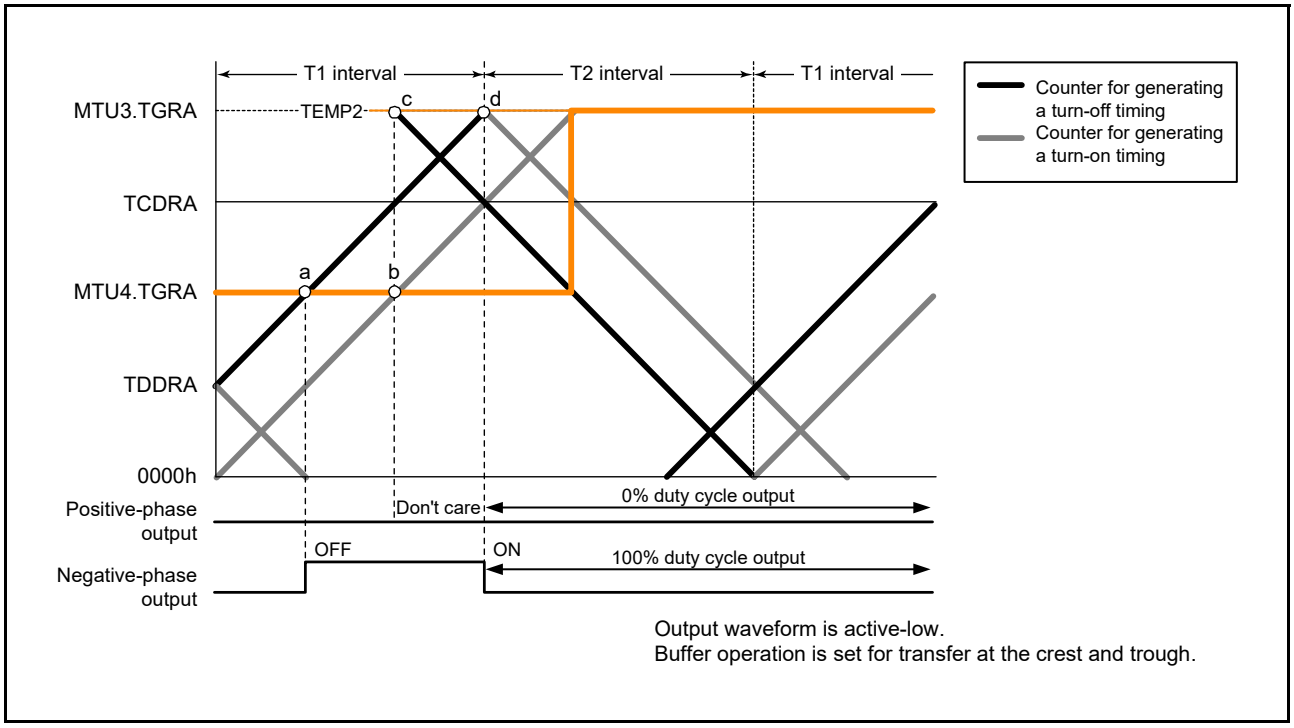


Figure 23.51 Example of 0% and 100% Waveform Output in Complementary PWM Mode (3)

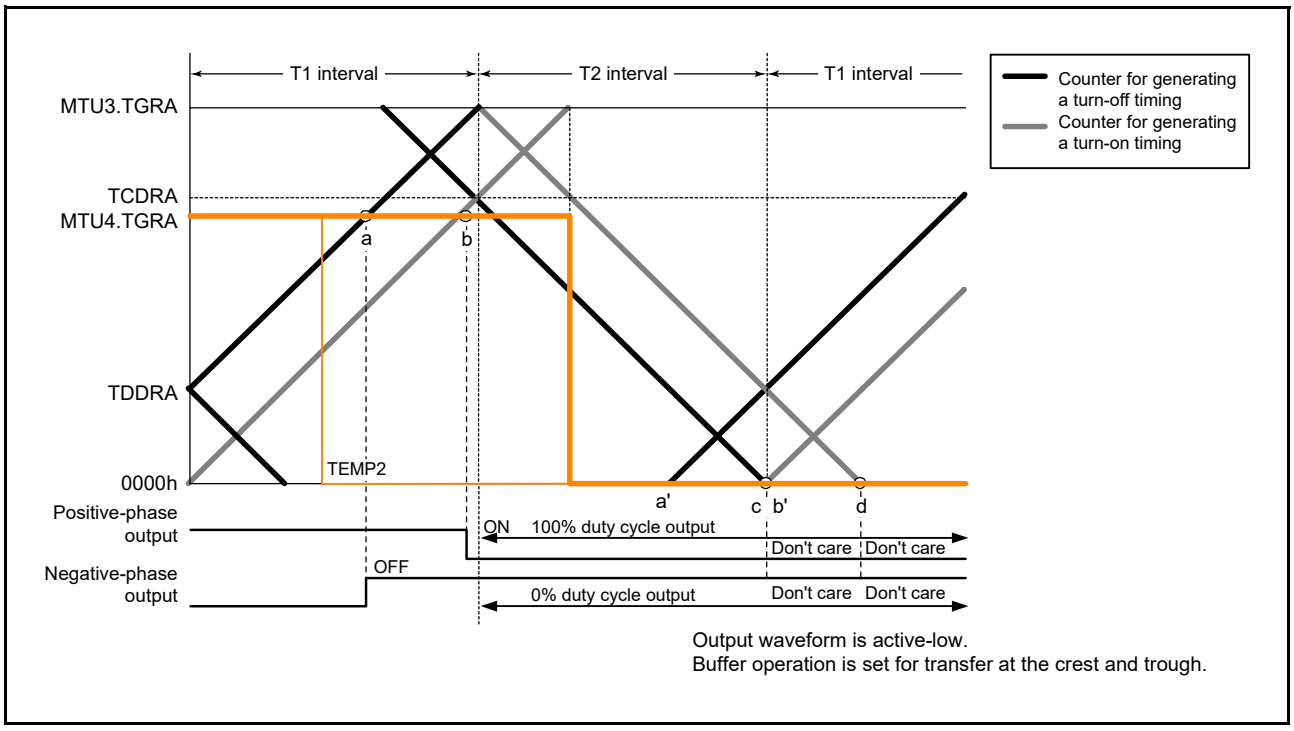


Figure 23.52 Example of 0% and 100% Waveform Output in Complementary PWM Mode (4)

Downloaded from Arrow.com.

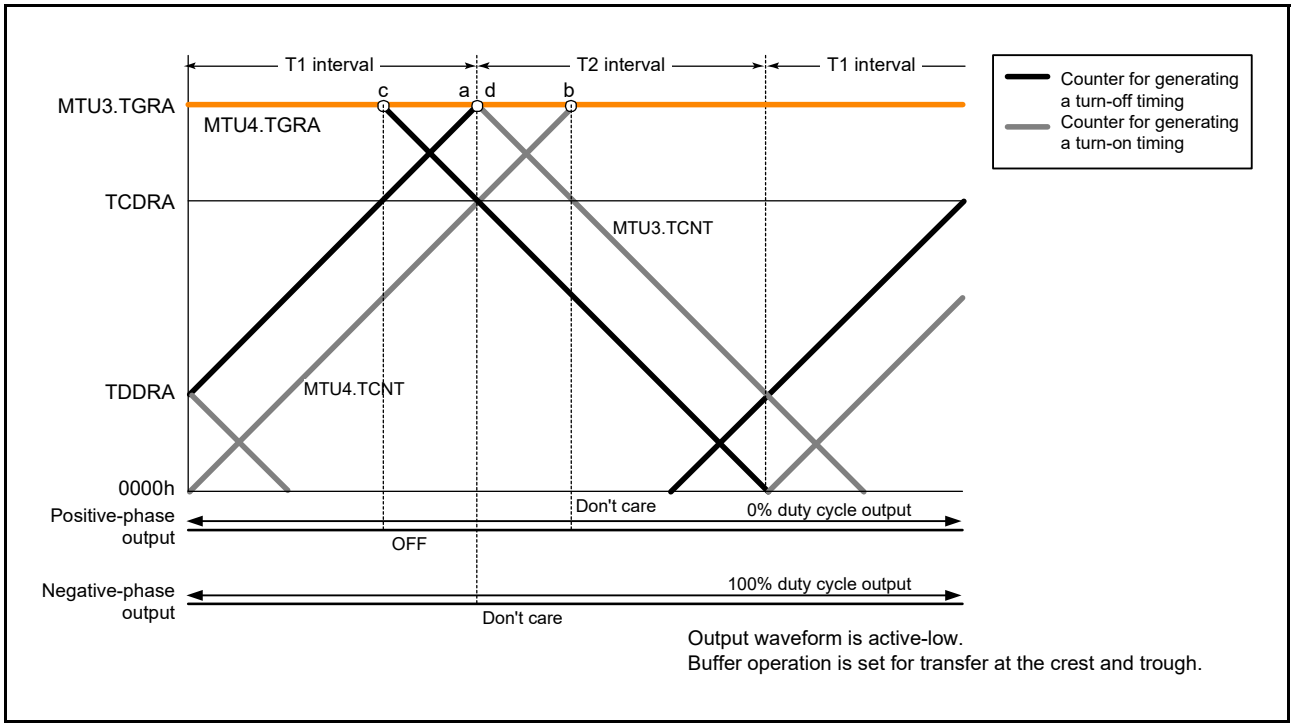


Figure 23.53 Example of 0% and 100% Waveform Output in Complementary PWM Mode (5)

(I) Toggle Output Synchronized with PWM Cycle

In complementary PWM mode, toggle output from the PWM output pin in synchronization with the PWM period can be enabled by setting the TOCR1.PSYE bit to 1. An example of a toggle output waveform is shown in Figure 23.54. This output is toggled by a compare match between the MTU3.TCNT counter and the MTU3.TGRA register and a compare match between the MTU4.TCNT counter and 0000h.

The MTIOC3A pin is assigned for this toggle output. The initial output is a high level.

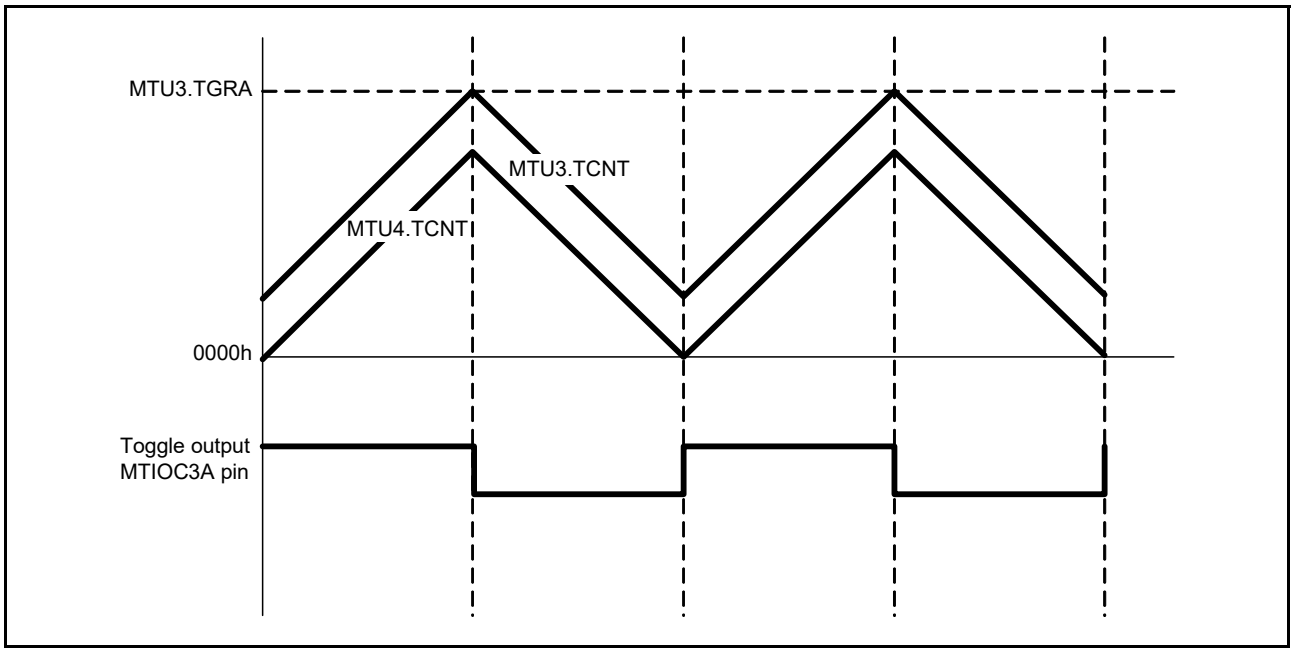


Figure 23.54 Example of Toggle Output Waveform Synchronized with PWM Output

Downloaded from Arrow.com.

(m) Counter Clearing by Another Channel

In complementary PWM mode, counters MTU3.TCNT, MTU4.TCNT, and TCNTS can be cleared by another channel source when a mode for synchronization with another channel is specified by the TSYR register and synchronous clearing is selected with the MTU3.TCR.CCLR[2:0] bits.

Figure 23.55 illustrates an example of this operation.

Use of this function enables a counter to be cleared and restarted through an external signal.

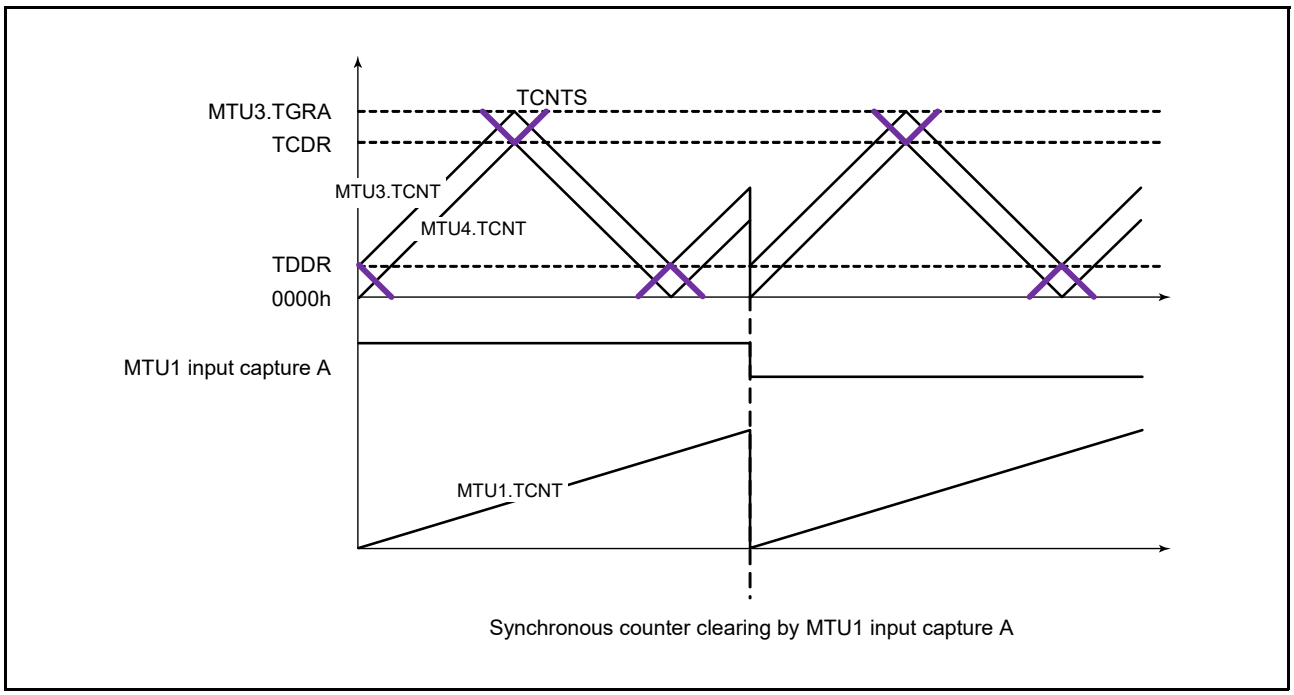


Figure 23.55 Counter Clearing Synchronized with Another Channel

Downloaded from Arrow.com.

(n) Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode

Setting the TWCR.WRE bit to 1 suppresses initial output when synchronous counter clearing occurs in the Tb interval (Tb2 interval) at the trough in complementary PWM mode and controls abrupt change in duty cycle at synchronous counter clearing.

Initial output suppression through setting the TWCR.WRE bit to 1 is applicable only when synchronous clearing occurs in the Tb2 interval as indicated by (10) or (11) in Figure 23.56. When synchronous clearing occurs outside that interval, the initial value specified by the TOCR1.OLSN bit and TOCR1.OLSP bit is output. Even in the Tb2 interval, if synchronous clearing occurs in the initial output period (indicated by (1) in Figure 23.56) immediately after the counters start operation, initial value output is not suppressed.

Synchronous clearing in any of MTU0 to MTU2 can cause counter clearing in MTU3 and MTU4.

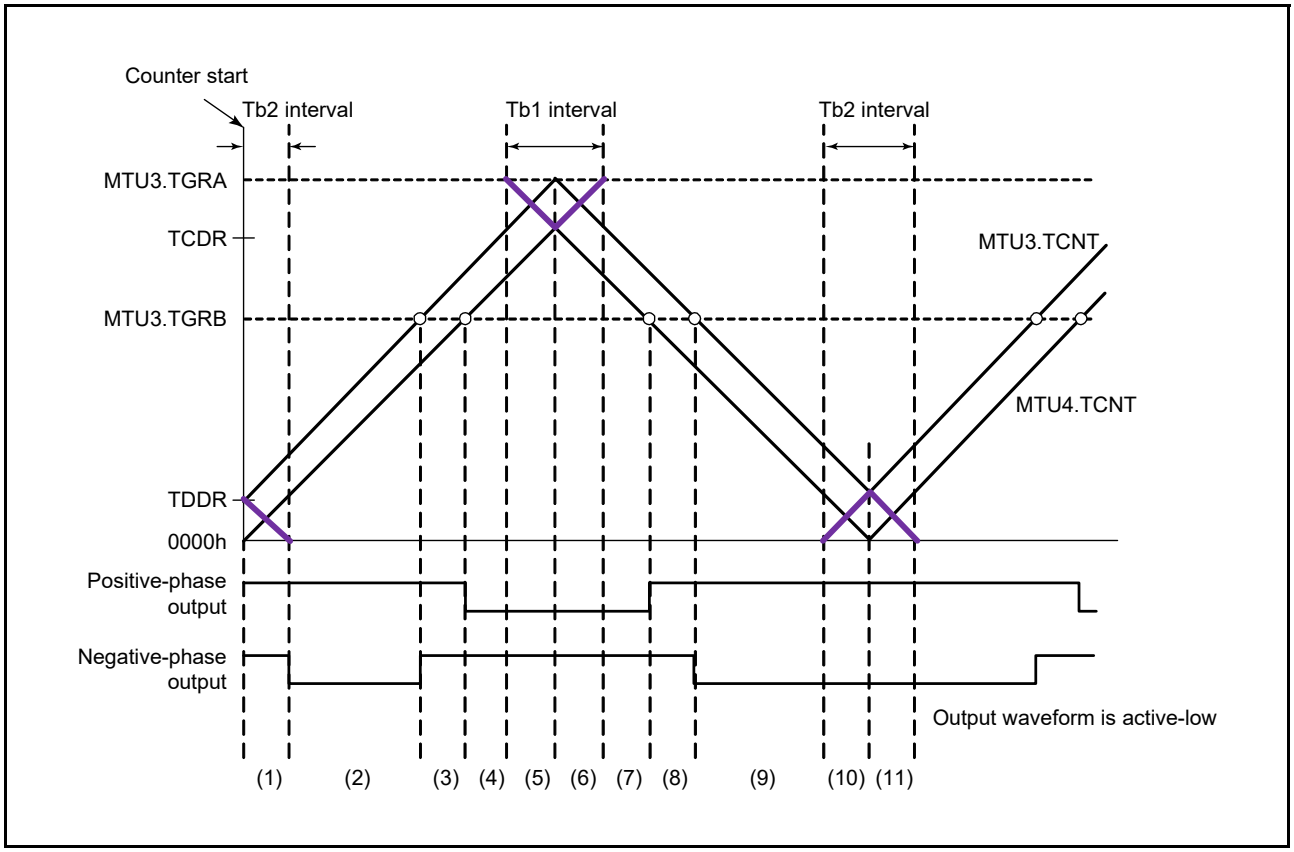


Figure 23.56 Timing for Synchronous Counter Clearing

Downloaded from Arrow.com.

- Example of Procedure for Setting Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode

An example of the procedure for setting output waveform control at synchronous counter clearing in complementary PWM mode is shown in Figure 23.57.

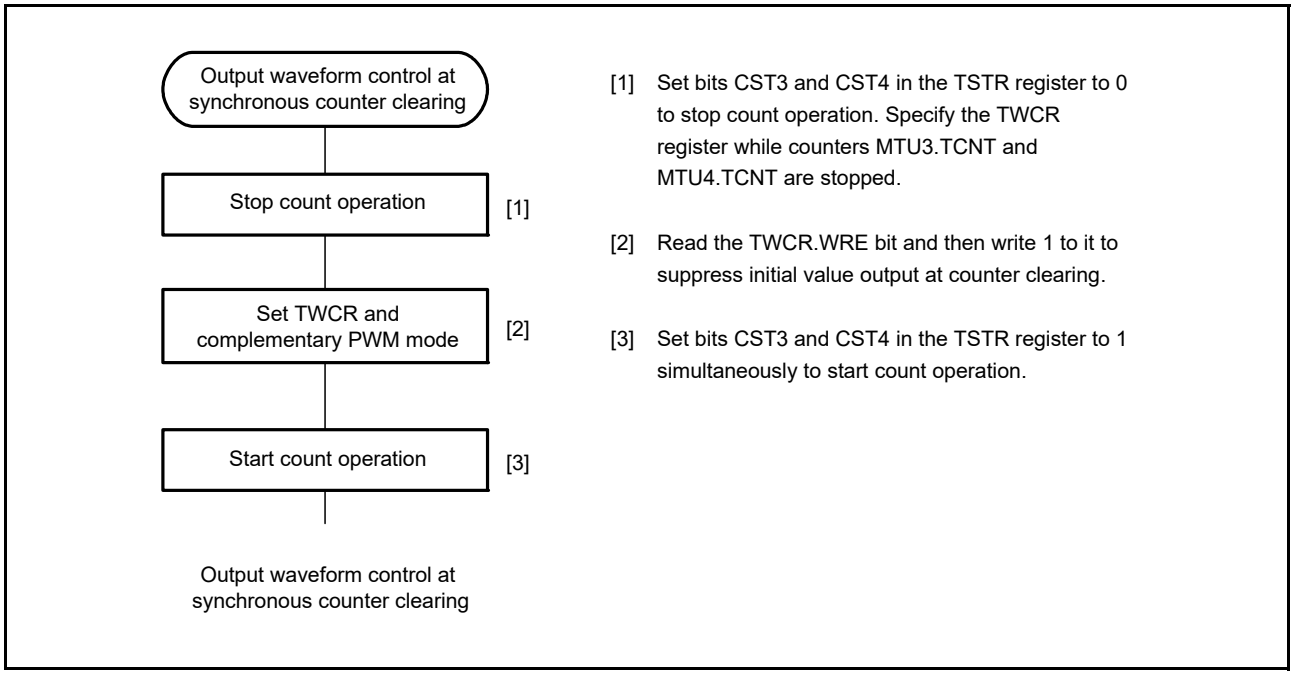


Figure 23.57 Example of Procedure for Setting Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode

- Examples of Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode

Figure 23.58 to Figure 23.61 show examples of output waveform control in which the MTU operates in complementary PWM mode and synchronous counter clearing is generated while the TWCR.WRE bit is set to 1. In the examples shown in Figure 23.58 to Figure 23.61, synchronous counter clearing occurs at timing (3), (6), (8), and (11) shown in Figure 23.56, respectively.

Downloaded from Arrow.com.

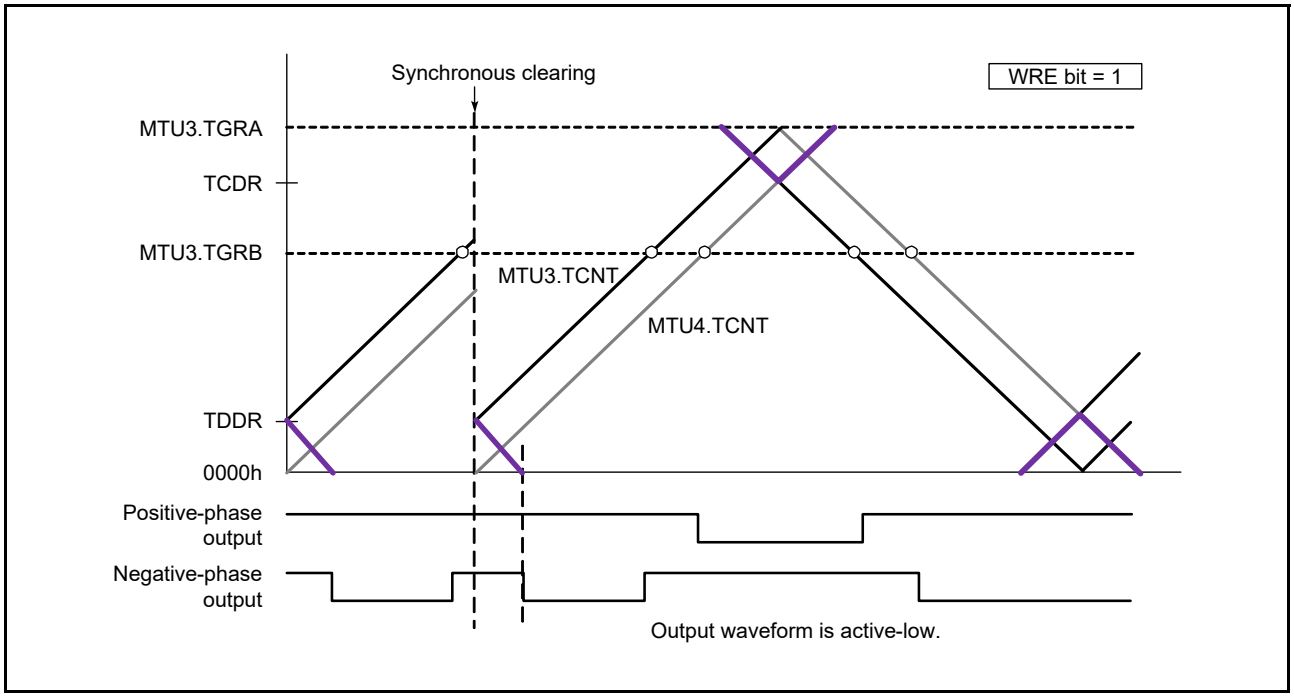


Figure 23.58 Example of Synchronous Clearing in Dead Time during Up-Counting (Timing (3) in Figure 23.56; TWCR.WRE Bit is 1)

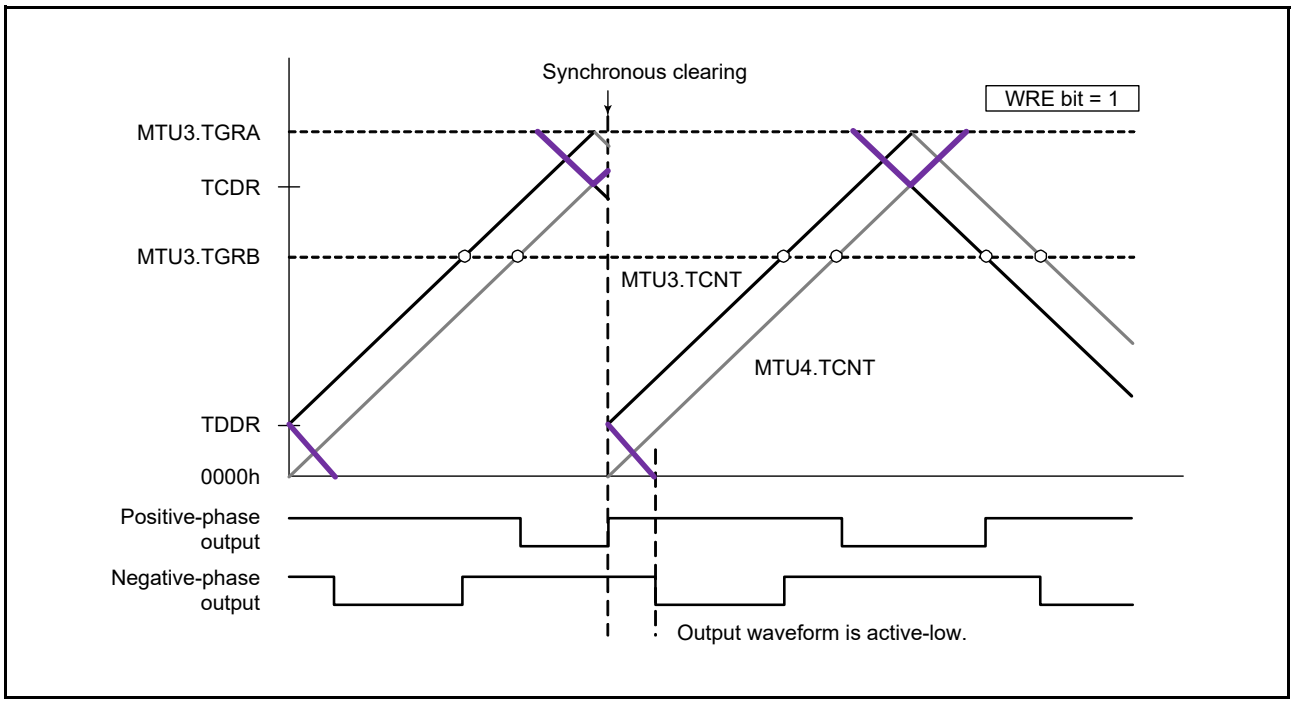


Figure 23.59 Example of Synchronous Clearing in Interval Tb at Crest (Timing (6) in Figure 23.56; TWCR.WRE Bit is 1)

Downloaded from Arrow.com.

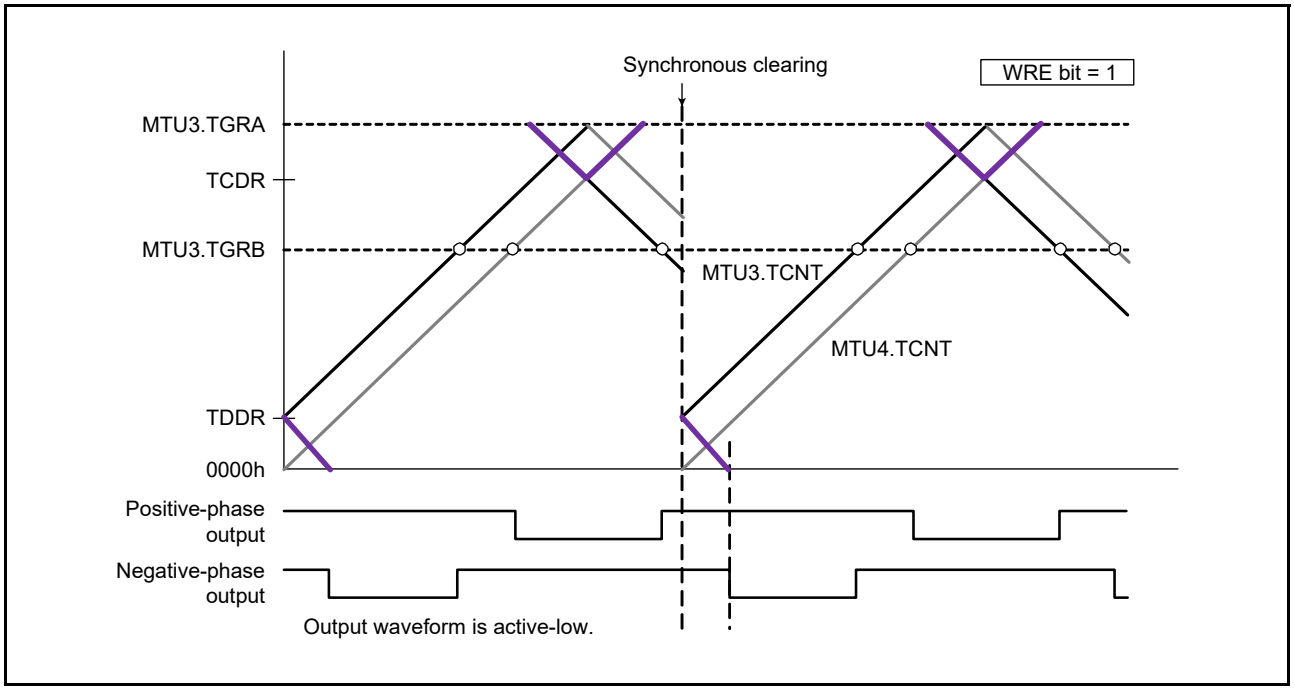


Figure 23.60 Example of Synchronous Clearing in Dead Time during Down-Counting (Timing (8) in Figure 23.56; TWCR.WRE Bit is 1)

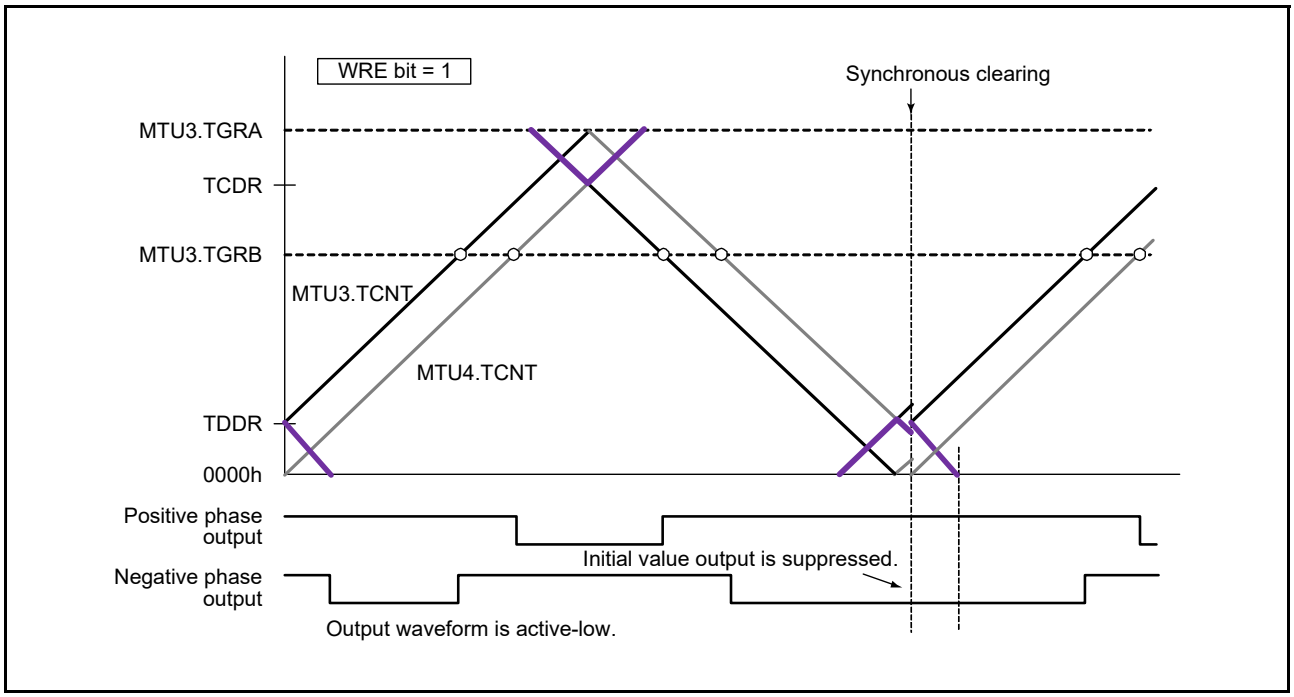


Figure 23.61 Example of Synchronous Clearing in Interval Tb at Trough (Timing (11) in Figure 23.56; TWCR.WRE Bit is 1)

Downloaded from Arrow.com.

(o) Counter Clearing by MTU3.TGRA Compare Match

In complementary PWM mode, counters MTU3.TCNT, MTU4.TCNT, and TCNTS can be cleared by MTU3.TGRA compare match when the TWCR.CCE bit is set.
Figure 23.62 shows an operation example.

Note: Use this function only in complementary PWM mode 1 (transfer at crest).
Note: Do not specify synchronous clearing by another channel (do not set the TSYR.SYNCn bits (n = 0 to 4) to 1).
Note: Do not set the PWM duty cycle value to 0000h.
Note: Do not set the TOCR1.PSYE bit to 1.

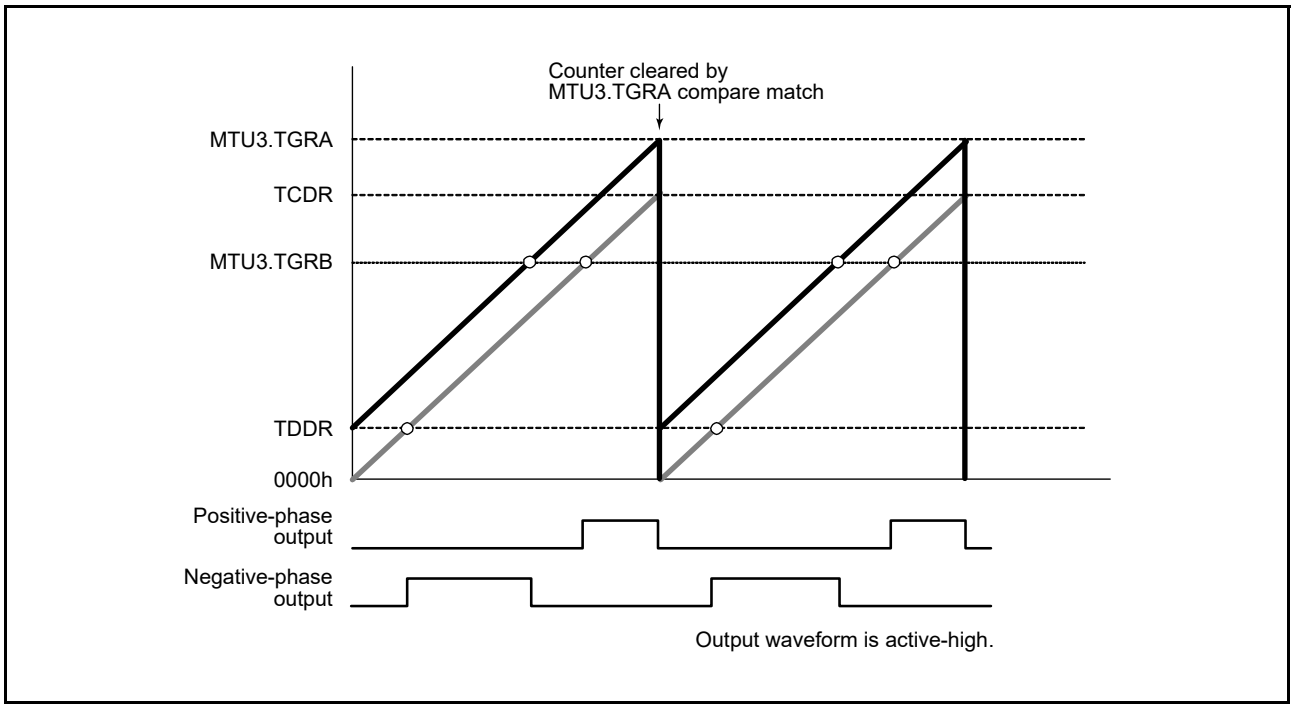


Figure 23.62 Example of Counter Clearing Operation by MTU3.TGRA Compare Match

Downloaded from Arrow.com.

(p) Example of Waveform Output for Driving AC Synchronous Motor (Brushless DC Motor)

In complementary PWM mode, a brushless DC motor can easily be controlled using the TGCR register. Figure 23.63 to Figure 23.66 show examples of brushless DC motor driving waveforms created using the TGCR register.

To switch the output phases for a 3-phase brushless DC motor by means of external signals detected with a Hall element, etc., set the TGCR.FB bit to 0. In this case, the external signals indicating the magnetic pole position should be input to timer input pins MTIOC0A, MTIOC0B, and MTIOC0C in MTU0. When an edge is detected at pin MTIOC0A, MTIOC0B, or MTIOC0C, the output on/off state is switched automatically.

When the TGCR.FB bit is 1, the output on/off state is switched when the TGCR.UF bit, TGCR.VF bit, or TGCR.WF bit is set to 0 or 1.

The driving waveforms are output from the 6-phase PWM output pins for complementary PWM mode.

With this 6-phase output, while the output is turned on, chopping output is available through complementary PWM mode output function by setting the TGCR.N bit or TGCR.P bit to 1. When the TGCR.N bit or TGCR.P bit is 0, the level output is selected.

The active level of the 6-phase output (on output level) can be set with the TOCR1.OLSN bit and TOCR1.OLSP bit regardless of the setting of the TGCR.N bit and TGCR.P bit.

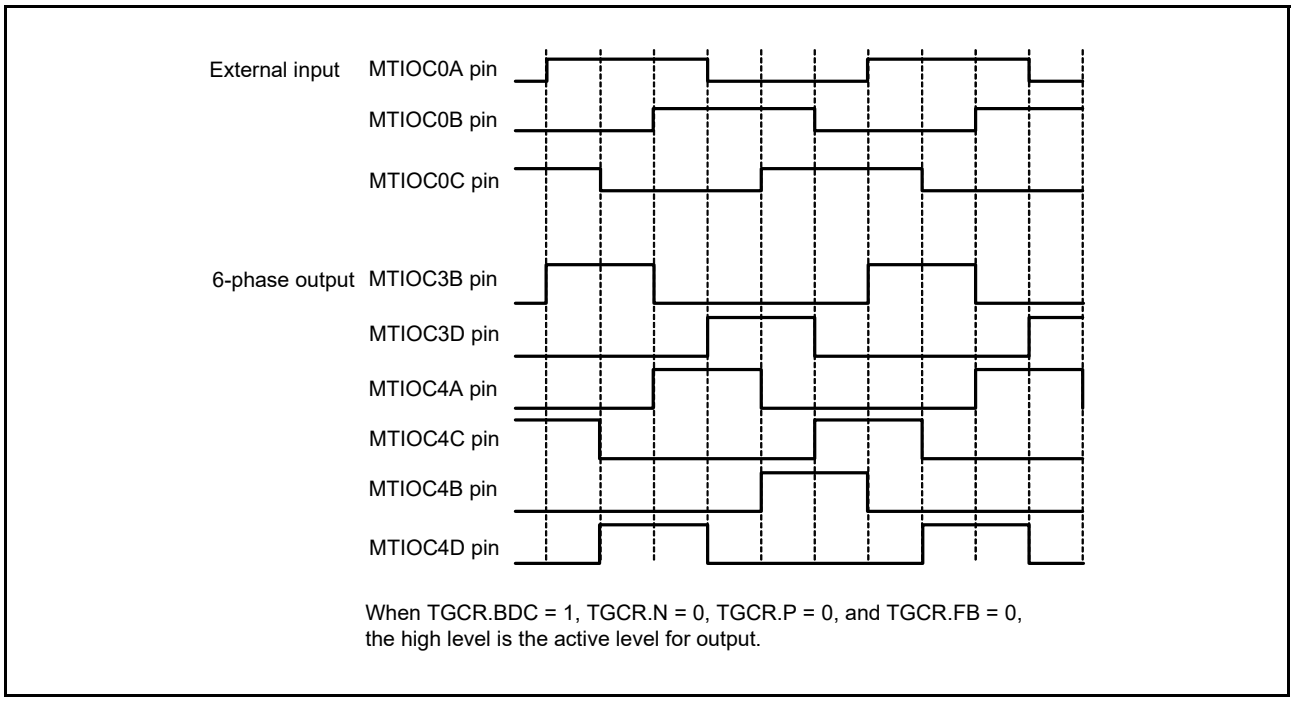


Figure 23.63 Example of Output Phase Switching by External Input (1)

Downloaded from Arrow.com.

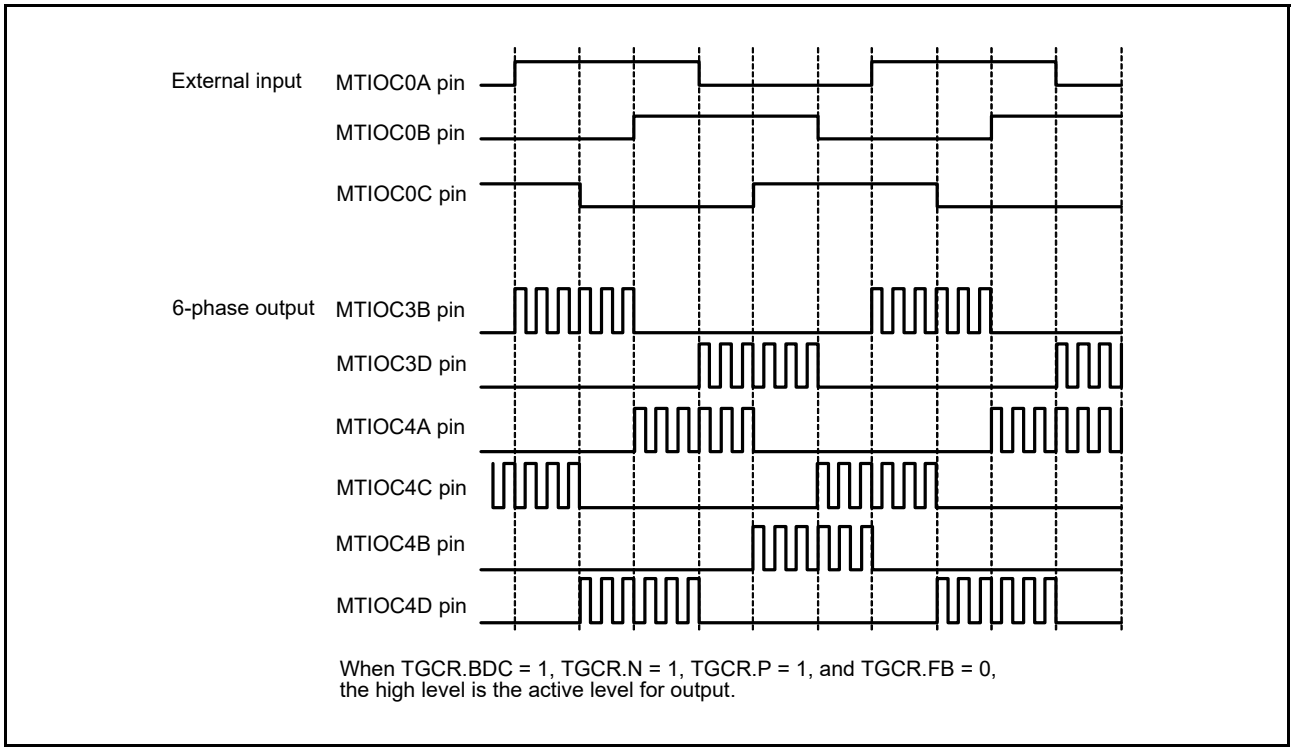


Figure 23.64 Example of Output Phase Switching by External Input (2)

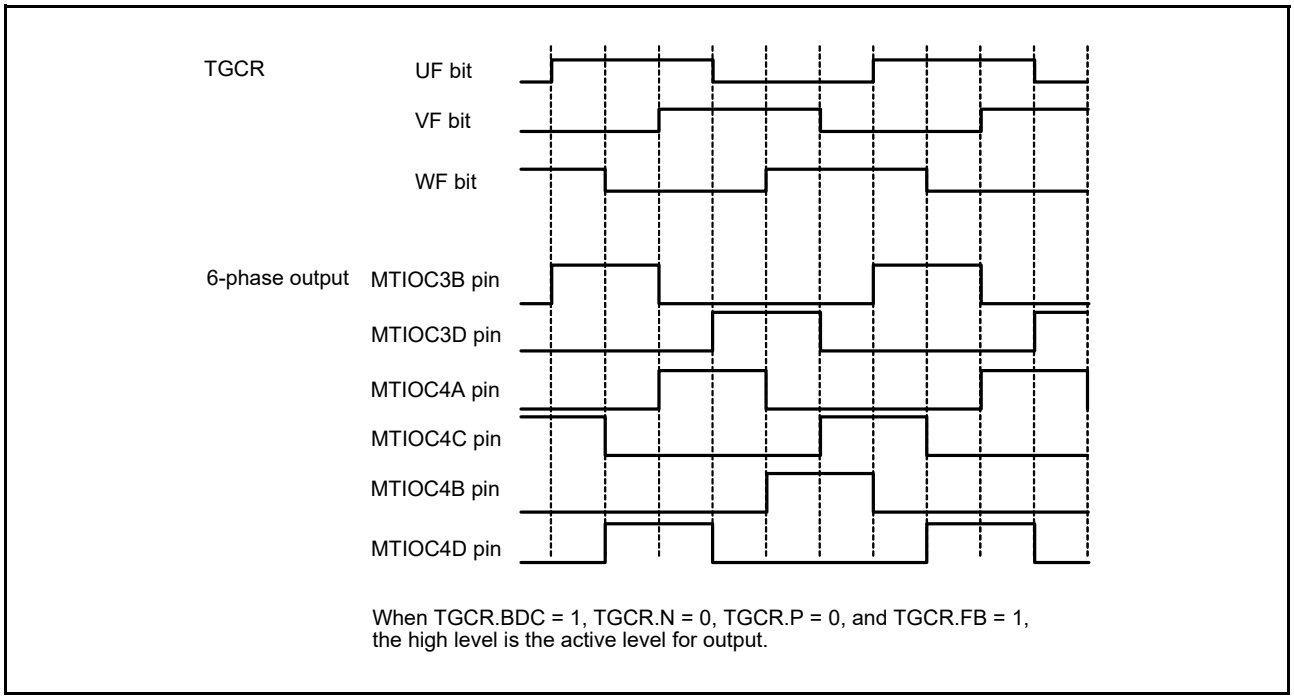


Figure 23.65 Example of Output Phase Switching through UF, VF, and WF Bit Settings (1)

Downloaded from Arrow.com.

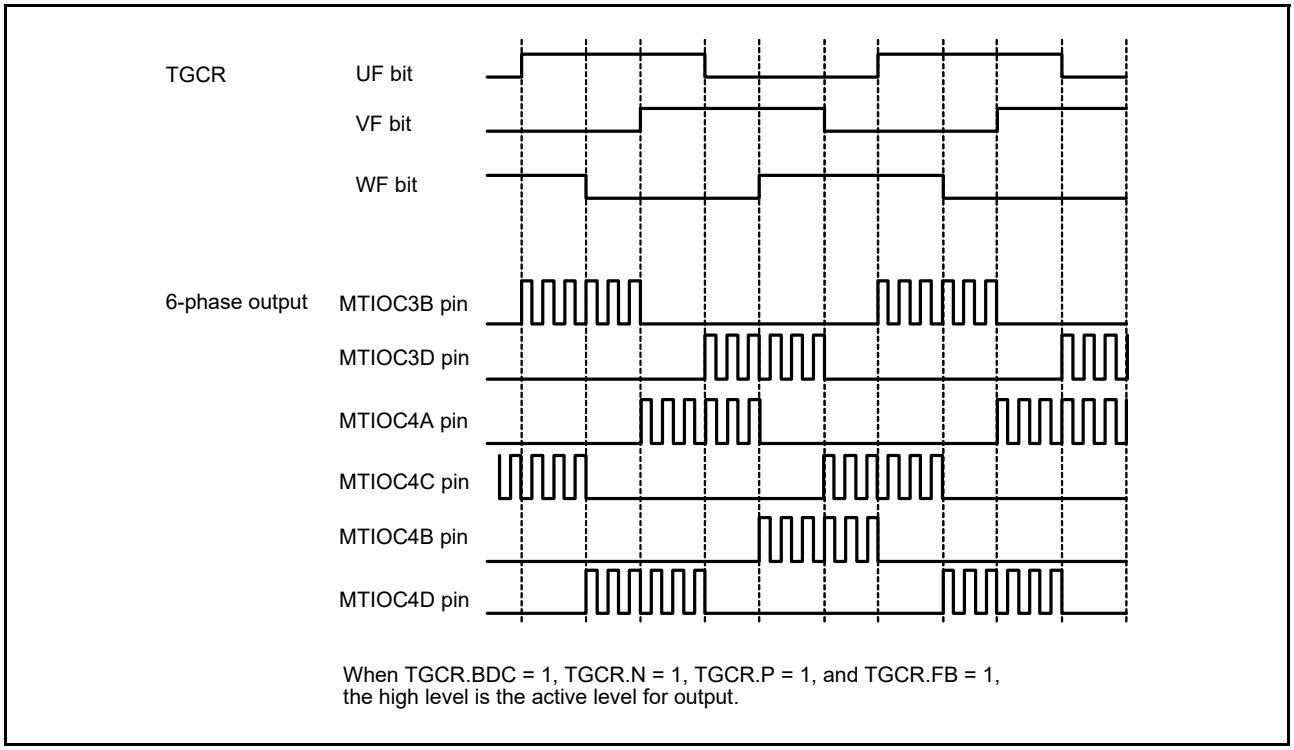


Figure 23.66 Example of Output Phase Switching through UF, VF, and WF Bit Settings (2)

(q) A/D Converter Start Request Setting

In complementary PWM mode, an A/D converter start request can be issued using the MTU3.TGRA compare match, the MTU4.TCNT underflow (trough), or compare match on a channel other than MTU3 and MTU4.

When start requests using the MTU3.TGRA compare match are specified, A/D conversion can be started at the crest of the MTU3.TCNT count.

A/D converter start requests can be specified by setting the TIER.TTGE bit to 1. To issue an A/D converter start request at an MTU4.TCNT underflow (trough), set the MTU4.TIER.TTGE2 bit to 1.

Downloaded from Arrow.com.

(3) Interrupt Skipping in Complementary PWM Mode

Interrupts TGIA3 (at the crest) and TCIV4 (at the trough) in MTU3 and MTU4 can be skipped up to seven times by setting the TITCR register.

Transfers from a buffer register to a temporary register or a compare register can be skipped in coordination with interrupt skipping by making settings in the TBTER register. For the linkage with buffer registers, refer to description (c) Buffer Transfer Control Linked with Interrupt Skipping, below.

A/D converter start requests generated by the A/D converter start request delaying function can also be skipped in coordination with interrupt skipping by making settings in the TADCR register. For the linkage with the A/D converter start request delaying function, refer to section 23.3.9, A/D Converter Start Request Delaying Function.

The TITCR register should be set while the TGIA3 and TCIV4 interrupt requests are disabled by the settings of registers MTU3.TIER and MTU4.TIER under the conditions in which compare match never occur and TGIA3 and TGIA4 interrupt requests by compare match are never generated. Before changing the skipping count, be sure to set the TITCR.T3AEN and TITCR.T4VEN bits to 0 to clear the skipping counter.

(a) Example of Interrupt Skipping Operation Setting Procedure

Figure 23.67 shows an example of the interrupt skipping operation setting procedure. Figure 23.68 shows the periods during which interrupt skipping count can be changed.

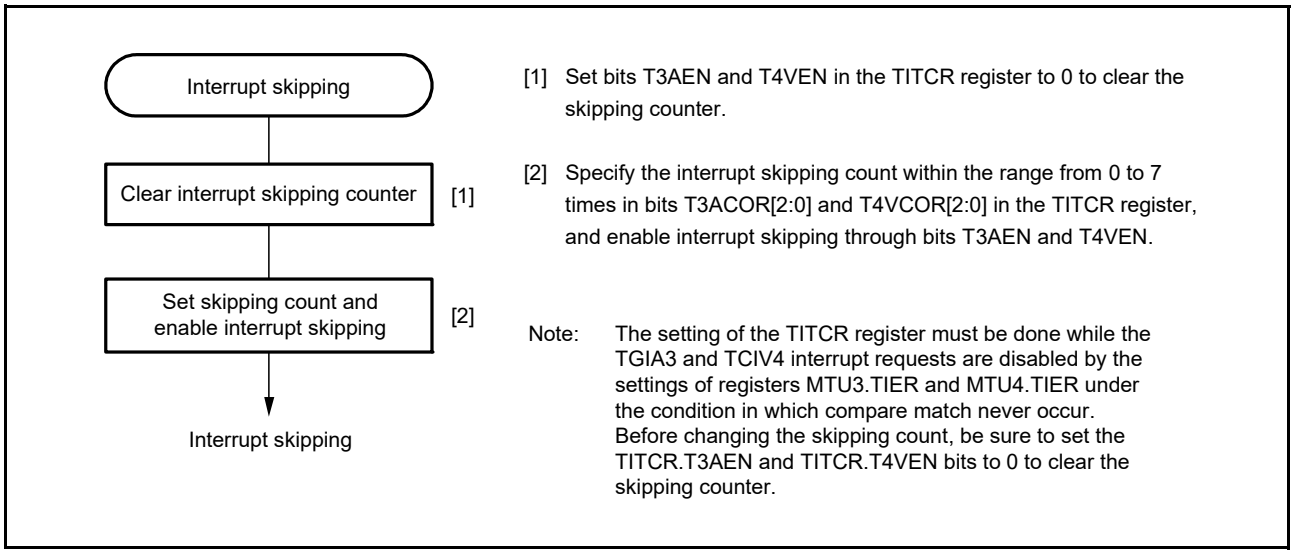


Figure 23.67 Example of Interrupt Skipping Operation Setting Procedure

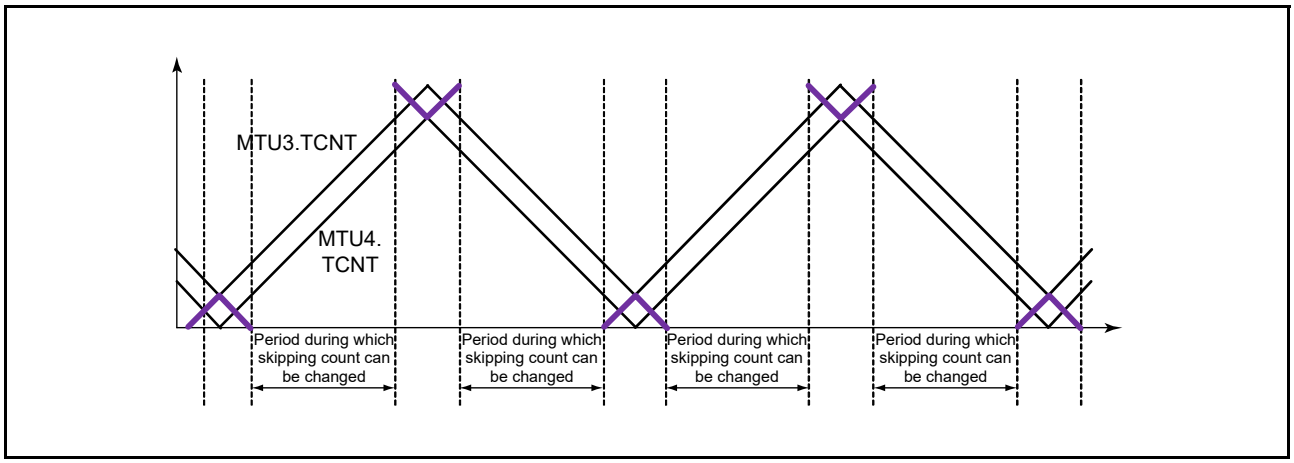


Figure 23.68 Periods during which Interrupt Skipping Count Can be Changed

Downloaded from Arrow.com.

(b) Example of Interrupt Skipping Operation

Figure 23.69 shows an example of MTU3.TGIA interrupt skipping in which the interrupt skipping count is set to three by the TITCR.T3ACOR[2:0] bits and the TITCR.T3AEN bit is set to 1.

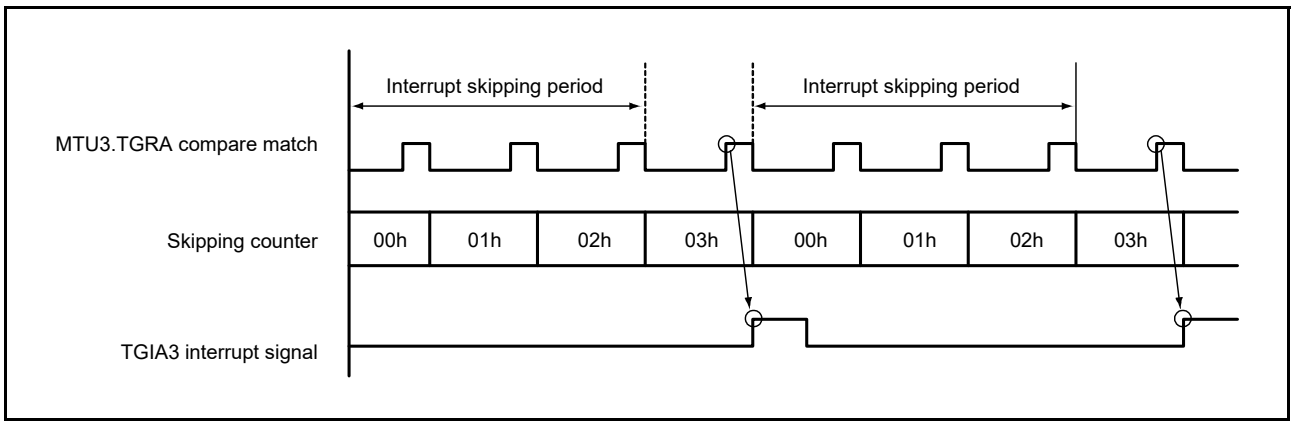


Figure 23.69 Example of Interrupt Skipping Operation

Downloaded from Arrow.com.

(c) Buffer Transfer Control Linked with Interrupt Skipping

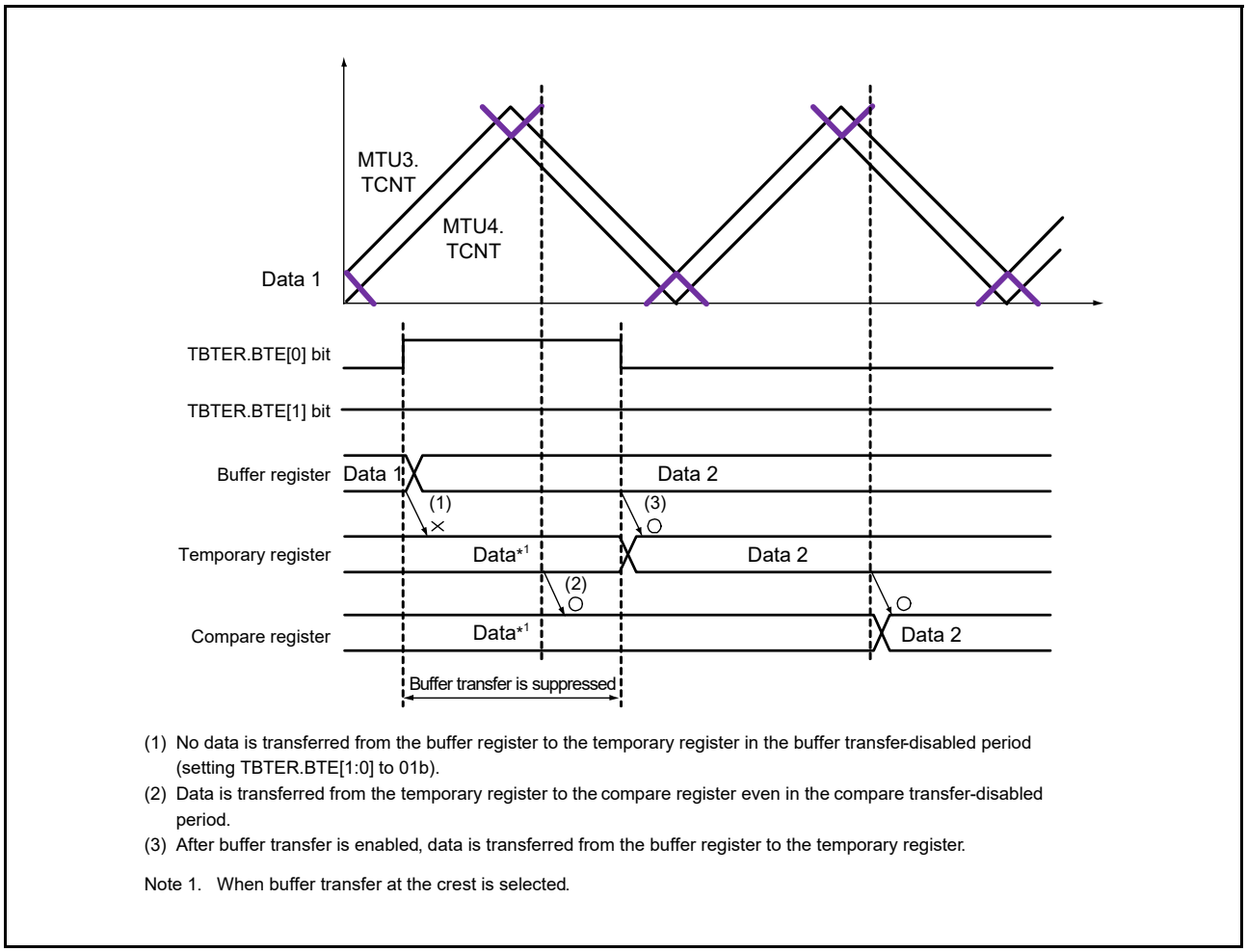
In complementary PWM mode, whether to transfer data from a buffer register to a temporary register and whether to link the transfer with interrupt skipping can be specified with the TBTER.BTE[1:0] bits.

Figure 23.70 shows an example of operation when buffer transfer is disabled (BTE[1:0] = 01b). While this setting is valid, data is not transferred from the buffer register to the temporary register.

Figure 23.71 shows an example of operation when buffer transfer is linked with interrupt skipping (BTE[1:0] = 10b). While this setting is valid, if data is written to the buffer register within the buffer transfer-enabled period, the data is transferred immediately from the buffer register to the temporary register. If data is written to the buffer register outside the buffer transfer-enabled period, the data is transferred from the buffer register to the temporary register at the timing when the next buffer transfer-enabled period starts.

Note that the buffer transfer-enabled period depends on the TITCR.T3AEN bit and TITCR.T4VEN bit settings. Figure 23.72 shows the relationship between the TITCR.T3AEN bit and TITCR.T4VEN bit settings and buffer transfer-enabled period.

Note: This function must always be used in combination with interrupt skipping.
When interrupt skipping is disabled (the T3AEN and T4VEN bits in the TITCR register are set to 0 or the skipping count setting bits (T3ACOR[2:0] and T4VCOR[2:0]) in the TITCR register are set to 000b), make sure that buffer transfer is not linked with interrupt skipping (set the TBTER.BTE[1] bit to 0).
If buffer transfer is linked with interrupt skipping while interrupt skipping is disabled, buffer transfer is never performed.



(1) No data is transferred from the buffer register to the temporary register in the buffer transfer-disabled period (setting TBTER.BTE[1:0] to 01b).
(2) Data is transferred from the temporary register to the compare register even in the compare transfer-disabled period.
(3) After buffer transfer is enabled, data is transferred from the buffer register to the temporary register.

Note 1. When buffer transfer at the crest is selected.

Figure 23.70 Example of Operation When Buffer Transfer is Disabled (TBTER.BTE[1:0] = 01b)

Downloaded from Arrow.com.

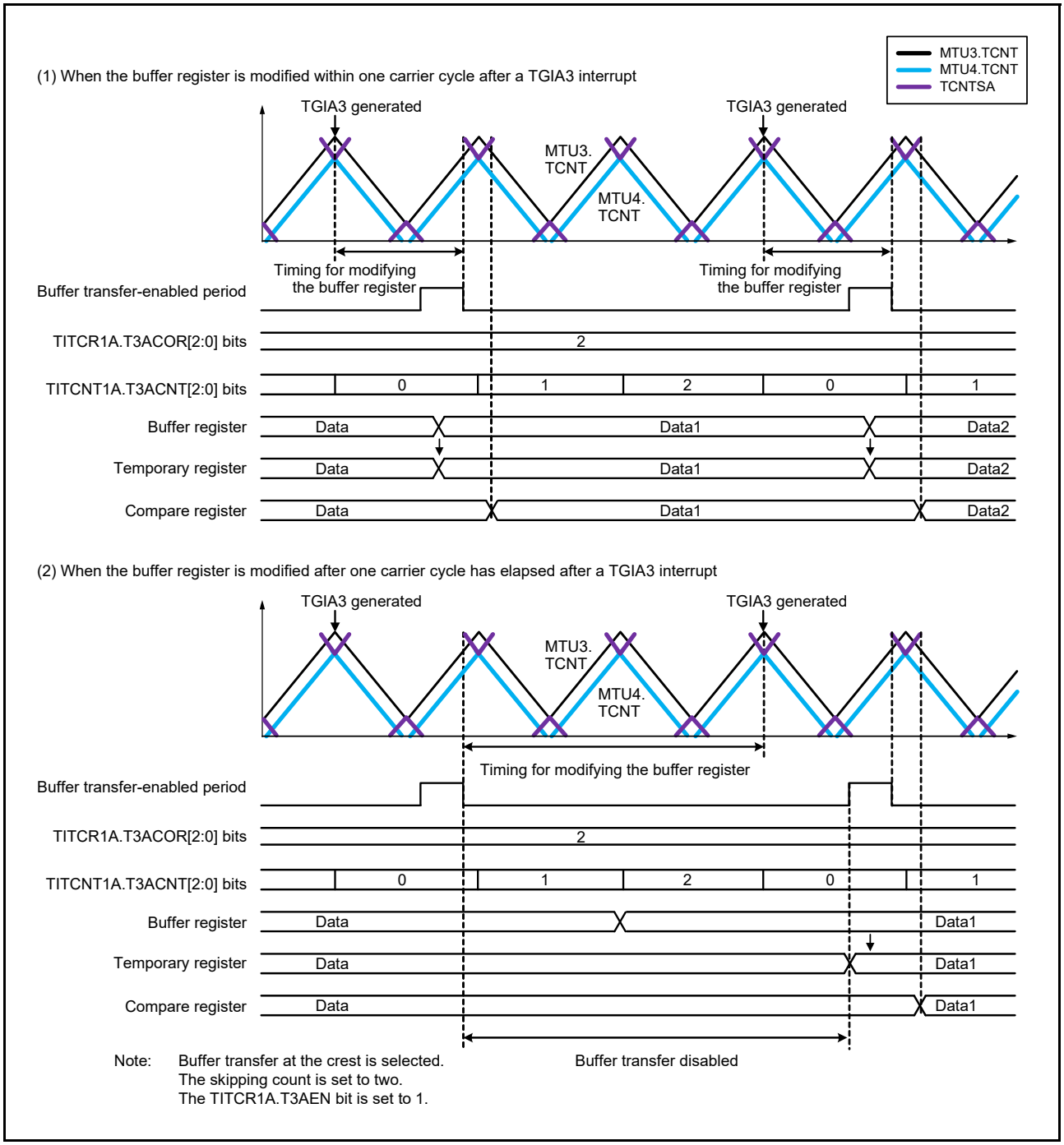


Figure 23.71 Example of Operation When Buffer Transfer is Linked with Interrupt Skipping (TBTER.BTE[1:0] = 10b)

Downloaded from Arrow.com.

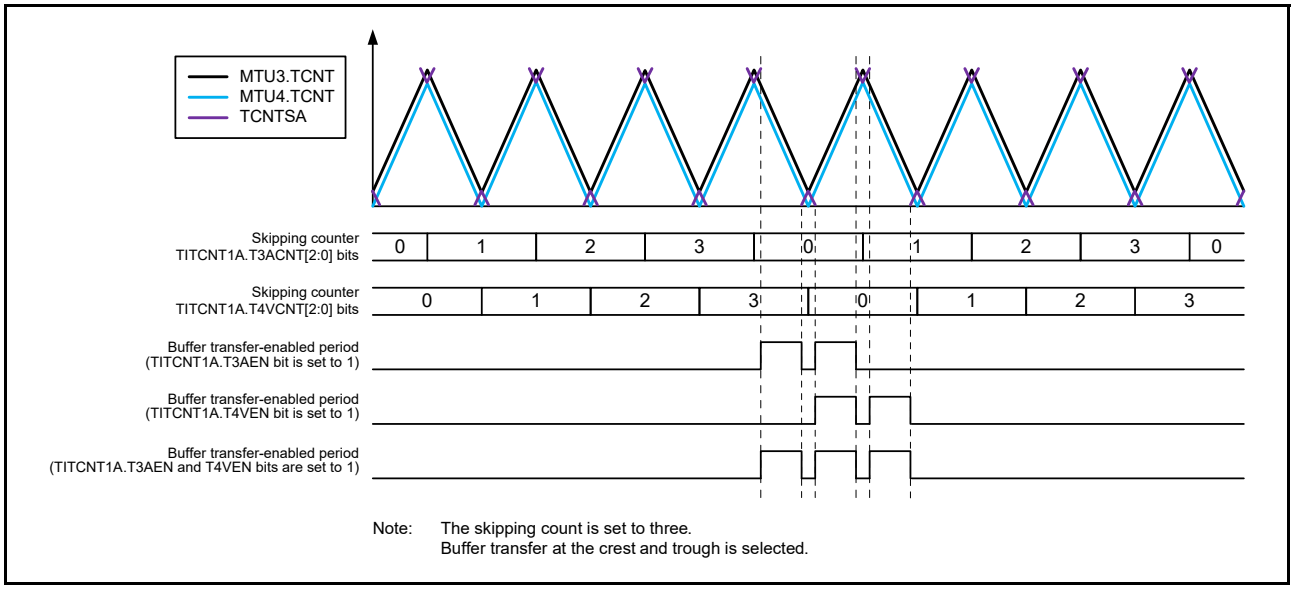


Figure 23.72 Relationship between Bits T3AEN and T4VEN in TITCR and Buffer Transfer-Enabled Period

(4) Complementary PWM Mode Output Protection Functions

The MTU provides the following protection functions for complementary PWM mode output.

(a) Register and Counter Miswrite Prevention Function

Access from the CPU to the mode registers, control registers, compare registers can be enabled or disabled by setting the TRWER.RWE bit. The applicable registers are some of the registers in MTU3 and MTU4 shown below:

22 registers in total

MTU3.TCR and MTU4.TCR, MTU3.TMDR and MTU4.TMDR, MTU3.TIORH and MTU4.TIORH, MTU3.TIORL and MTU4.TIORL, MTU3.TIER and MTU4.TIER, MTU3.TCNT and MTU4.TCNT, MTU3.TGRA and MTU4.TGRA, MTU3.TGRB and MTU4.TGRB, MTU.TOER, MTU.TOCR1, MTU.TOCR2, MTU.TGCR, MTU.TCDR, and MTU.TDDR

This function can disable CPU access to the mode registers, control registers, and counters to prevent miswriting due to CPU runaway. In the access-disabled state, the applicable registers are read as undefined and writing to these registers is ignored.

(b) Halting of PWM Output

The PWM output pins of MTU0, MTU3, and MTU4 can be set to the high-impedance state automatically.
Refer to section 24, Port Output Enable 2 (POE2a), for details.

Downloaded from Arrow.com.

23.3.9 A/D Converter Start Request Delaying Function

A/D converter start requests can be issued in MTU4 by making settings in registers TADCR, TADCORA, TADCORB, TADCOBRA, and TADCOBRB.

The A/D converter start request delaying function compares the MTU4.TCNT counter with the MTU4.TADCORA or MTU4.TADCORB register, and when their values match, the function issues a respective A/D converter start request (TRG4AN or TRG4BN).

A/D converter start requests (TRG4AN and TRG4BN) can be skipped in coordination with interrupt skipping by making settings in the TADCR.ITA3AE bit, TADCR.ITA4VE bit, TADCR.ITB3AE bit, and ITB4VE bit.

(1) Example of Procedure for Specifying A/D Converter Start Request Delaying Function

Figure 23.73 shows an example of procedure for specifying the A/D converter start request delaying function.

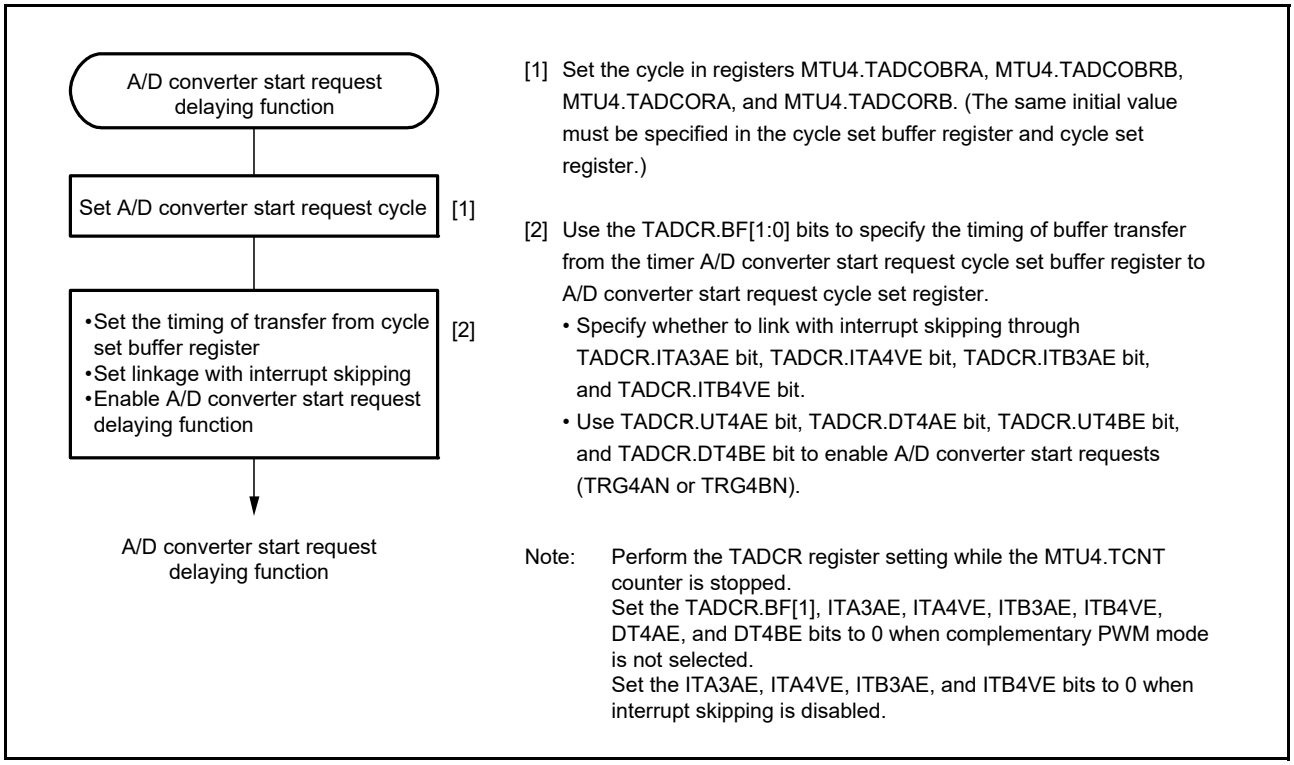
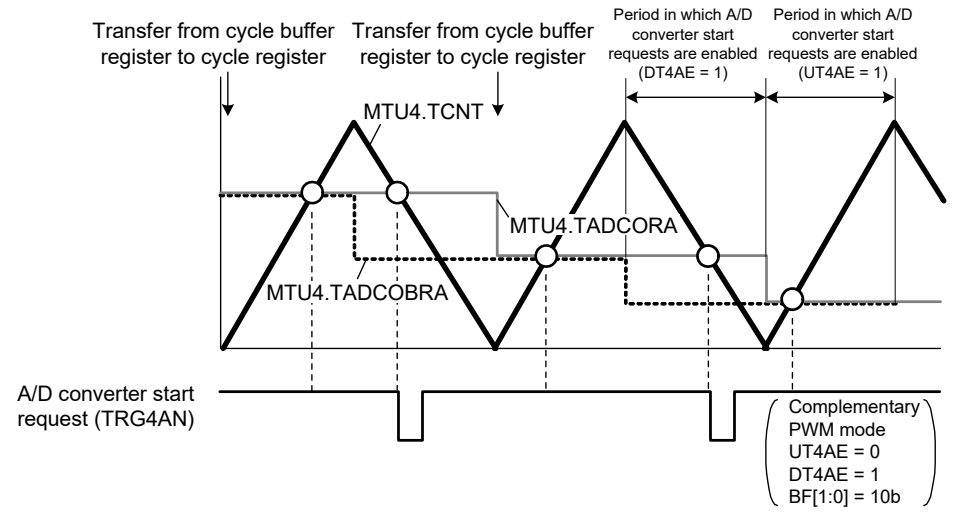


Figure 23.73 Example of Procedure for Specifying A/D Converter Start Request Delaying Function

Downloaded from Arrow.com.

(2) Basic Example of A/D Converter Start Request Delaying Function Operation

Figure 23.74 shows a basic example of A/D converter start request signal (TRG4AN) operation when the trough of the MTU4.TCNT counter is specified for the buffer transfer timing and an A/D converter start request signal is output during MTU4.TCNT down-counting.



Period in which A/D converter start requests are enabled (UT4AE = 1): 0 ≤ MTU4.TCNT ≤ TCDR – 1
Period in which A/D converter start requests are enabled (DT4AE = 1): TCDR ≥ MTU4.TCNT ≥ 1

UT4AE, DT4AE, BF[1:0]: Bits in TADCR

Figure 23.74 Basic Example of A/D Converter Start Request Signal (TRG4AN) Operation

(3) Period in Which A/D Converter Start Requests are Enabled

When the MTU4.TCNT counter and the MTU4.TADCORA or MTU4.TADCORB register match within the period enabled by the UT4AE and UT4BE bits, the corresponding A/D converter start request (TRG4AN or TRG4BN) is issued.

When the UT4AE and UT4BE bits in the MTU4.TADCR register are set to 1 in complementary PWM mode, A/D converter start requests are enabled during the MTU4.TCNT up-counting (0 ≤ MTU4.TCNT ≤ TCDR – 1). When the DT4AE and DT4BE bits in the MTU4.TADCR register are set to 1, A/D converter start requests are enabled during MTU4.TCNT down-counting (TCDR ≥ MTU4.TCNT ≥ 1). Refer to Figure 23.74.

(4) Buffer Transfer

The data in the timer A/D converter start request cycle set registers (MTU4.TADCORA and MTU4.TADCORB) is updated by writing data to the timer A/D converter start request cycle set buffer registers (MTU4.TADCOBRA and MTU4.TADCOBRB). Data is transferred from the buffer registers to the respective cycle set registers at the timing selected with the MTU4.TADCR.BF[1:0] bits.

There are notes on the timing for transferring data when using buffer transfer in complementary PWM mode. For details, section 23.6.26, Usage Notes on A/D Converter Delaying Function in Complementary PWM Mode.

In modes other than complementary PWM mode, set the BF[1] bit in the MTU4.TADCR register to 0.

Downloaded from Arrow.com.

(5) A/D Converter Start Request Delaying Function Linked with Interrupt Skipping

In complementary PWM mode, A/D converter start requests (TRG4AN and TRG4BN) can be issued in coordination with interrupt skipping by making settings in the TADCR.ITA3AE bit, TADCR.ITA4VE bit, TADCR.ITB3AE bit, and TADCR.ITB4VE bit. Figure 23.75 shows an example of A/D converter start request signal (TRG4AN) operation when TRG4AN output is enabled during MTU4.TCNT up-counting and down-counting and A/D converter start requests are linked with interrupt skipping.

Figure 23.76 shows another example of A/D converter start request signal (TRG4AN) operation when TRG4AN output is enabled during MTU4.TCNT up-counting and A/D converter start requests are linked with interrupt skipping.

In modes other than complementary PWM mode, do not use the A/D converter start request delaying function linked with interrupt skipping.

Set the ITA3AE, ITA4VE, ITB3AE, and ITB4VE bits in the MTU4.TADCR register to 0.

Note: This function should be used in combination with interrupt skipping.
When interrupt skipping is disabled (the TITCR.T3AEN bit and TITCR.T4VEN bit are set to 0 or the skipping count setting bits (T3ACOR[2:0] and T4VCOR[2:0]) in the TITCR register are set to 000b), make sure that A/D converter start requests are not linked with interrupt skipping (set the TADCR.ITA3AE bit, TADCR.ITA4VE bit, TADCR.ITB3AE bit, and TADCR.ITB4VE bit to 0).
Note that TRG4ABN (TRG4AN or TRG4BN) is output as the A/D converter start request signal in this case.
When this function is used, registers MTU4.TADCORA and MTU4.TADCORB should be set with the value ranging 0002h to the value set in the TCDRA register minus 2.

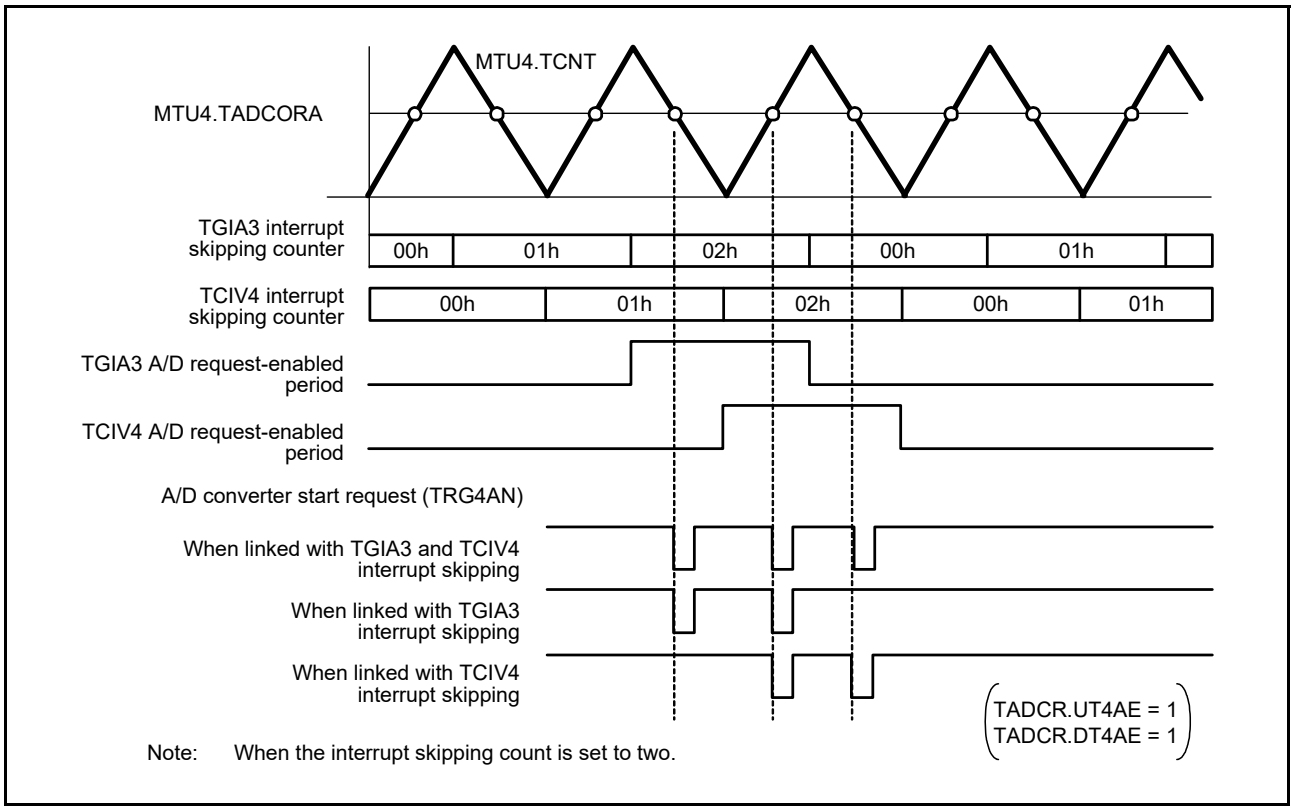


Figure 23.75 Example of A/D Converter Start Request Signal (TRG4AN) Operation Linked with Interrupt Skipping (when the output of TRG4AN in counting up and down by TCNT is enabled)

Downloaded from Arrow.com.

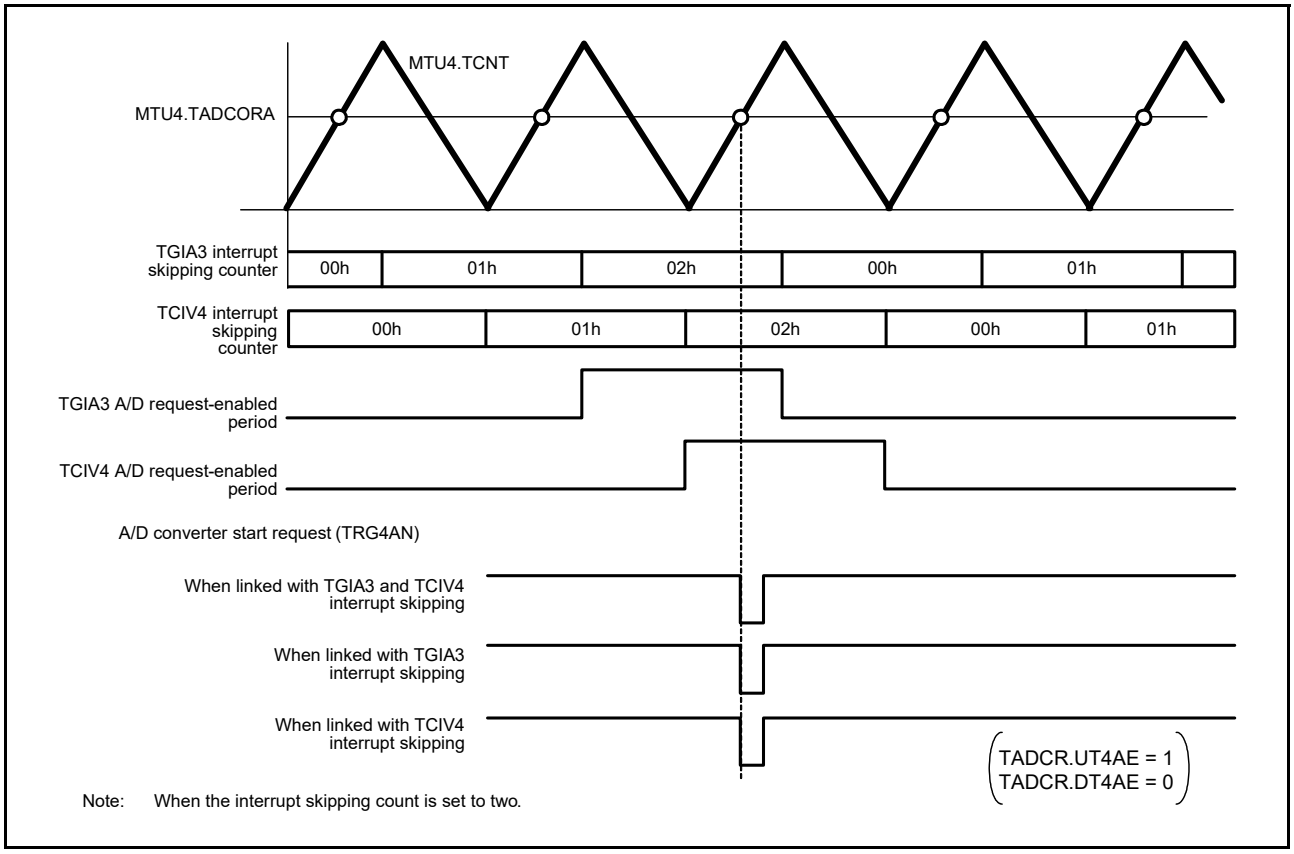


Figure 23.76 Example of A/D Converter Start Request Signal (TRG4AN) Operation Linked with Interrupt Skipping (when the output of TRG4AN in counting up by TCNT is enabled)

Downloaded from Arrow.com.

23.3.10 Noise Filter

Each pin for use in input capture and external pulse input to the MTU is equipped with a noise filter. The noise filter samples input signals at the sampling clock and removes the pulses of which length is less than three sampling cycles. The noise filter functionality includes enabling and disabling of the noise filter for each pin and setting of the sampling clock for each channel. Figure 23.77 shows the timing of noise filtering.

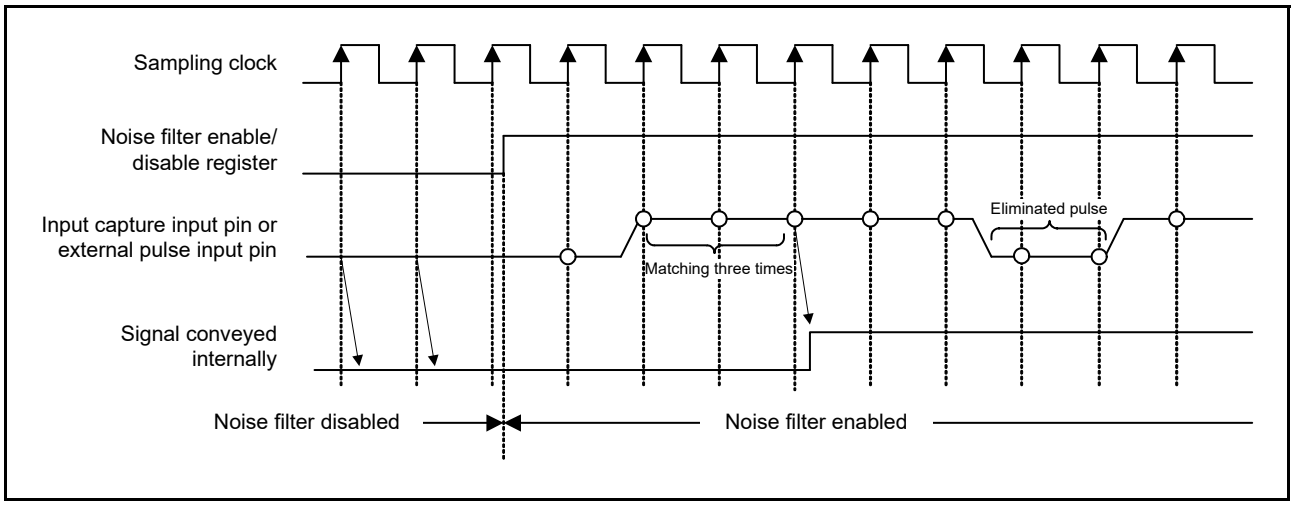


Figure 23.77 Timing of Noise Filtering

Downloaded from Arrow.com.

23.4 Interrupt Sources

23.4.1 Interrupt Sources and Priorities

There are three interrupt sources; the TGR input capture/compare match, the TCNT counter overflow, and the TCNT counter underflow. Each interrupt source has its own enable/disable bit, allowing the generation of interrupt request signals to be enabled or disabled individually.

When an interrupt source is detected, an interrupt is requested if the corresponding enable/disable bit in the TIER register is set to 1.

Relative channel priorities can be changed by the interrupt controller; however the priority within a channel is fixed. For details, refer to section 15, Interrupt Controller (ICUb).

Table 23.54 lists the MTU interrupt sources.

Table 23.54 MTU Interrupt Sources (1)

Channel	Name	Interrupt Source	DMAC Activation	DTC Activation	Priority
MTU0	TGIA0	MTU0.TGRA input capture/compare match	Possible	Possible	High
	TGIB0	MTU0.TGRB input capture/compare match	Not possible	Possible	↑
	TGIC0	MTU0.TGRC input capture/compare match	Not possible	Possible	
	TGID0	MTU0.TGRD input capture/compare match	Not possible	Possible	
	TCIV0	MTU0.TCNT overflow	Not possible	Not possible	
	TGIE0	MTU0.TGRE compare match	Not possible	Not possible	
	TGIF0	MTU0.TGRF compare match	Not possible	Not possible	
MTU1	TGIA1	MTU1.TGRA input capture/compare match	Possible	Possible	
	TGIB1	MTU1.TGRB input capture/compare match	Not possible	Possible	
	TCIV1	MTU1.TCNT overflow	Not possible	Not possible	
	TCIU1	MTU1.TCNT underflow	Not possible	Not possible	
MTU2	TGIA2	MTU2.TGRA input capture/compare match	Possible	Possible	
	TGIB2	MTU2.TGRB input capture/compare match	Not possible	Possible	
	TCIV2	MTU2.TCNT overflow	Not possible	Not possible	
	TCIU2	MTU2.TCNT underflow	Not possible	Not possible	
MTU3	TGIA3	MTU3.TGRA input capture/compare match	Possible	Possible	
	TGIB3	MTU3.TGRB input capture/compare match	Not possible	Possible	
	TGIC3	MTU3.TGRC input capture/compare match	Not possible	Possible	
	TGID3	MTU3.TGRD input capture/compare match	Not possible	Possible	
	TCIV3	MTU3.TCNT overflow	Not possible	Not possible	
MTU4	TGIA4	MTU4.TGRA input capture/compare match	Possible	Possible	
	TGIB4	MTU4.TGRB input capture/compare match	Not possible	Possible	
	TGIC4	MTU4.TGRC input capture/compare match	Not possible	Possible	
	TGID4	MTU4.TGRD input capture/compare match	Not possible	Possible	
	TCIV4	MTU4.TCNT overflow/underflow	Not possible	Possible	Low

Note: This table lists the initial state immediately after a reset. The relative channel priorities can be changed by the interrupt controller.

Downloaded from Arrow.com.

(1) Input Capture/Compare Match Interrupt

An interrupt is requested if the TIER.TGIE bit is set to 1 when a TGR input capture/compare match occurs on a channel. The MTU has 18 input capture/compare match interrupts (six for MTU0, four each for MTU3 and MTU4, and two each for MTU1 and MTU2).

(2) Overflow Interrupt

An interrupt is requested if the TIER.TGIE bit is set to 1 when a TCNT counter overflow occurs on a channel. The MTU has five overflow interrupts (one for each channel).

(3) Underflow Interrupt

An interrupt is requested if the TIER.TCIEU bit is set to 1 when a TCNT counter underflow occurs on a channel. The MTU has two underflow interrupts (one each for MTU1 and MTU2).

23.4.2 DTC/DMAC Activation

(1) DTC Activation

The DTC can be activated by the TGR input capture/compare match interrupt in each channel or the overflow interrupt in MTU4. For details, refer to section 19, Data Transfer Controller (DTCa).
The MTU provides a total of 17 input capture/compare match interrupts and overflow interrupts that can be used as DTC activation sources: four each for MTU0 and MTU3, two each for MTU1 and MTU2, and five for MTU4.

(2) DMAC Activation

The DMAC can be activated by the TGRA register input capture/compare match interrupt in each channel. For details, refer to section 18, DMA Controller (DMACA).
The MTU provides a total of five input capture/compare match interrupts that can be used as DMAC activation sources: one each for MTU0 to MTU4.
When the DMAC is activated by the MTU, the activation source is cleared when the DMAC requests the internal bus mastership. Therefore, there may be a wait period before DMAC transfer starts even when the activation source is cleared, depending on the internal bus state.

23.4.3 A/D Converter Activation

The A/D converter can be activated by one of the following five methods in the MTU. Table 23.55 lists the relationship between interrupt sources and A/D converter start request signals.

(1) A/D Converter Activation by TGRA Input Capture/Compare Match or at MTU4.TCNT Trough in Complementary PWM Mode

The A/D converter can be activated by the occurrence of a TGRA input capture/compare match in each channel. In addition, if complementary PWM mode operation is performed while the MTU4.TIER.TTGE2 bit is set to 1, the A/D converter can be activated at the trough of MTU4.TCNT count (MTU4.TCNT = 0000h).
A/D converter start request signal TRGAN is issued to the A/D converter under either of the following conditions.

- When a TGRA input capture/compare match occurs on a channel while the TIER.TTGE bit is set to 1
- When the MTU4.TCNT count reaches the trough (MTU4.TCNT = 0000h) during complementary PWM mode operation while the MTU4.TIER.TTGE2 bit is set to 1

When either condition is satisfied, if A/D converter start signal TRGAN from the MTU is selected as the trigger in the A/D converter, A/D conversion will start.

Downloaded from Arrow.com.

(2) A/D Converter Activation by Compare Match between MTU0.TCNT and MTU0.TGRE

A compare match between the MTU0.TCNT counter and the MTU0.TGRE register activates the A/D converter.
A/D converter start request signal TRG0EN is issued when a compare match occurs between the MTU0.TCNT counter and the MTU0.TGRE register. If A/D converter start signal TRG0EN from the MTU is selected as the trigger in the A/D converter, A/D conversion will start.

(3) A/D Converter Activation by Compare Match between MTU0.TCNT and MTU0.TGRF

An input capture or compare match between the MTU0.TCNT counter and the MTU0.TGRA or MTU0.TGRB register activates the A/D converter.
A compare match between the MTU0.TCNT counter and the MTU0.TGRF register activates the A/D converter.
A/D converter start request signal TRG0FN is issued when a compare match occurs between the MTU0.TCNT counter and the MTU0.TGRF register. If A/D converter start signal TRG0FN from the MTU is selected as the trigger in the A/D converter, A/D conversion will start.

(4) A/D Converter Activation by Input Capture or Compare Match with MTU0.TGRA or TGRB

The A/D converter can be activated when an input capture or compare match occurs between the MTU0.TCNT counter and the MTU0.TGRA or MTU0.TGRB register.
When an input capture or compare match occurs between the MTU0.TCNT counter and the MTU0.TGRA or MTU0.TGRB register. A/D converter start request signal TRG0AN or TRG0BN is issued. If A/D converter start signal TRG0AN or TRG0BN from the MTU is selected as the trigger in the A/D converter, A/D conversion will start.

(5) A/D Converter Activation by A/D Converter Start Request Delaying Function

The A/D converter can be activated by generating A/D converter start request signal TRG4AN or TRG4BN when the MTU4.TCNT count matches the TADCORA or TADCORB register value if the TADCR.UT4AE bit, TADCR.DT4AE bit, TADCR.UT4BE bit, or TADCR.DT4BE bit is set to 1. For details, refer to section 23.3.9, A/D Converter Start Request Delaying Function.
A/D conversion will start if A/D converter start signal TRG4ABN from the MTU is selected as the trigger in the A/D converter when TRG4AN or TRG4BN is generated.

Table 23.55 Interrupt Sources and A/D Converter Start Request Signals

Target Registers	A/D Start Request Source	A/D Converter Start Request Signal
MTU0.TGRA and MTU0.TCNT	Input capture/compare match	TRGAN
MTU1.TGRA and MTU1.TCNT		
MTU2.TGRA and MTU2.TCNT		
MTU3.TGRA and MTU3.TCNT		
MTU4.TGRA and MTU4.TCNT		
MTU4.TCNT	MTU4.TCNT trough in complementary PWM mode	
MTU0.TGRA and MTU0.TCNT	Input capture/compare match	TRG0AN
MTU0.TGRB and MTU0.TCNT		TRG0BN
MTU0.TGRE and MTU0.TCNT	Compare match	TRG0EN
MTU0.TGRF and MTU0.TCNT		TRG0FN
TADCORA and MTU4.TCNT		TRG4AN
TADCORB and MTU4.TCNT		TRG4BN
TADCORA and MTU4.TCNT or TADCORB and MTU4.TCNT		TRG4ABN

Downloaded from Arrow.com.

23.5 Operation Timing

23.5.1 Input/Output Timing

(1) TCNT Count Timing

Figure 23.78 show the TCNT count timing for TGI interrupt in internal clock operation, Figure 23.79 shows the TCNT count timing in external clock operation (normal mode), and Figure 23.80 shows the TCNT count timing in external clock operation (phase counting mode).

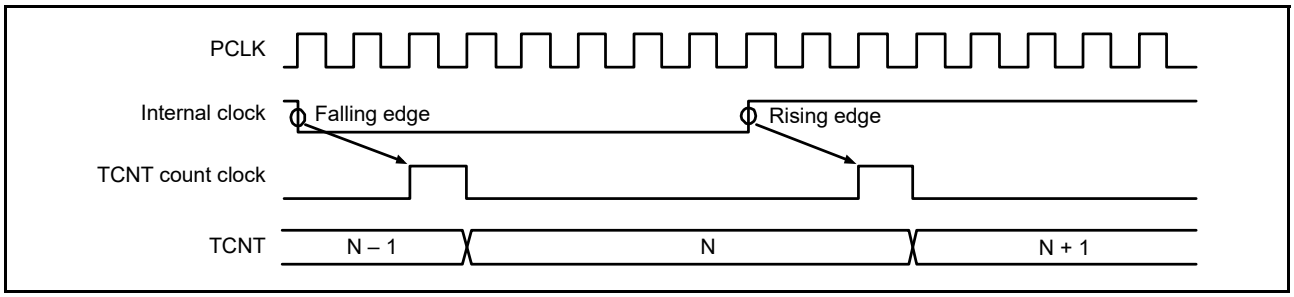


Figure 23.78 Count Timing in Internal Clock Operation (MTU0 to MTU4)

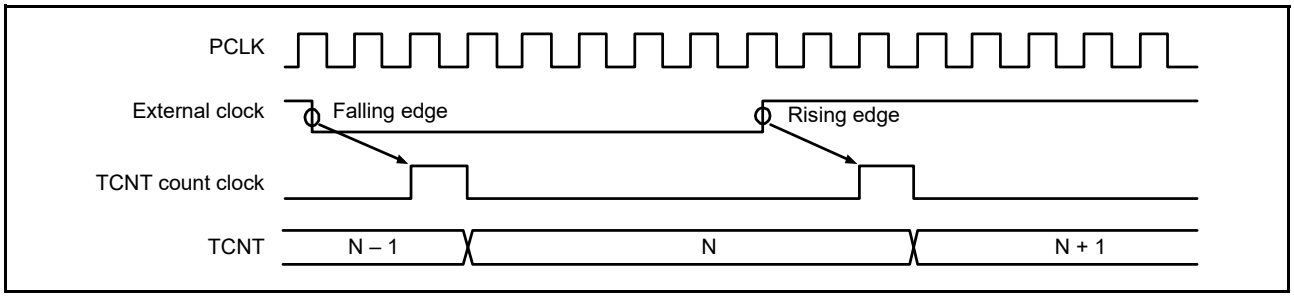


Figure 23.79 Count Timing in External Clock Operation (MTU0 to MTU4)

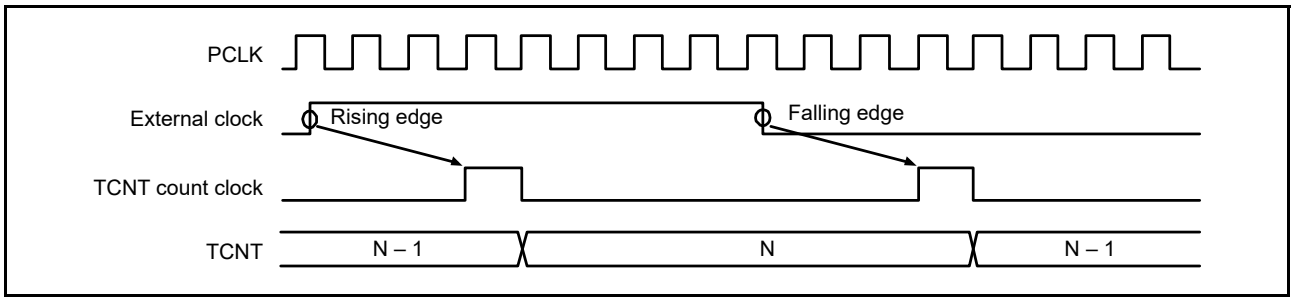


Figure 23.80 Count Timing in External Clock Operation (Phase Counting Mode)

Downloaded from Arrow.com.

(2) Output Compare Output Timing

A compare match signal is generated in the final state in which the TCNT counter and the TGR register match (the point at which the count value matched is updated by the TCNT counter). When a compare match signal is generated, the value set in the TIOR register is output to the output compare output pin (MTIOC pin). After a match between the TCNT counter and the TGR register, the compare match signal is not generated until the TCNT count clock is generated. Figure 23.81 shows the output compare output timing (normal mode or PWM mode) and Figure 23.82 shows the output compare output timing (complementary PWM mode or reset-synchronized PWM mode).

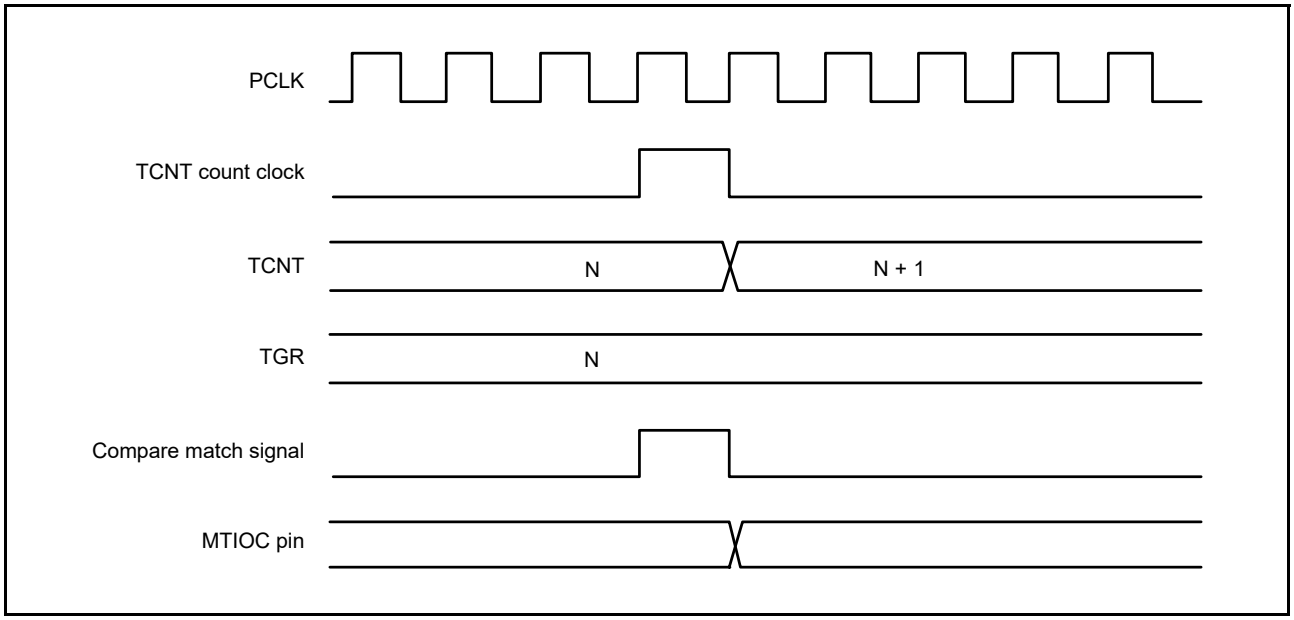


Figure 23.81 Output Compare Output Timing (Normal Mode or PWM Mode)

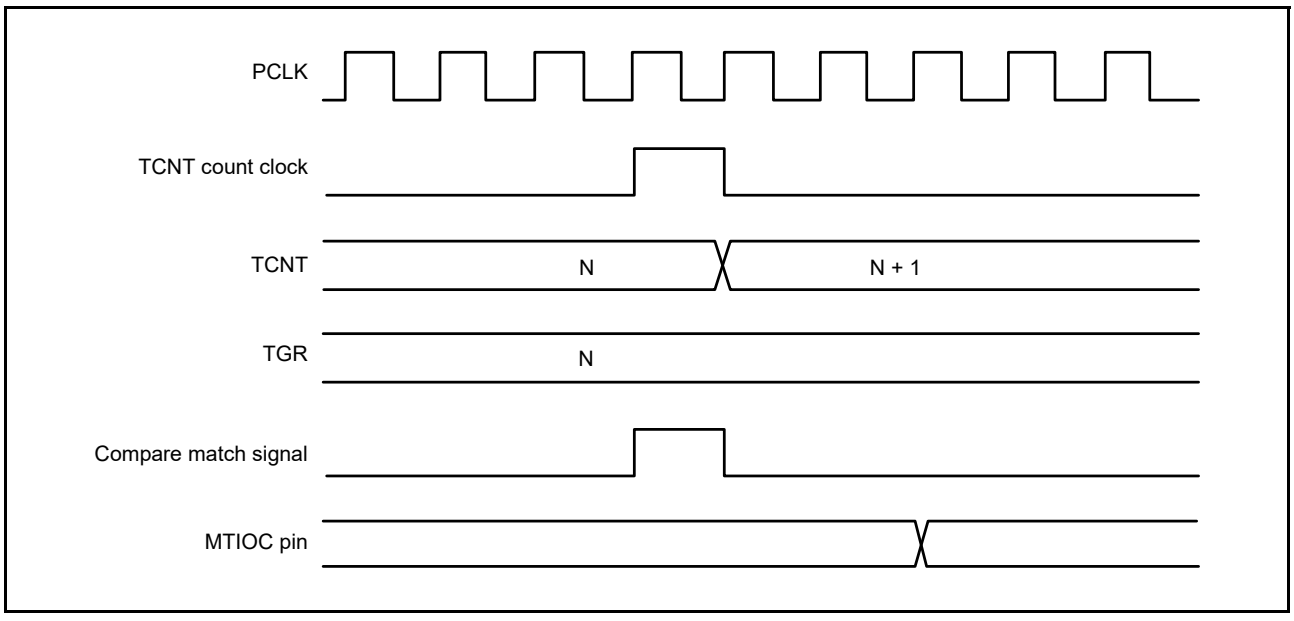


Figure 23.82 Output Compare Output Timing (Complementary PWM Mode or Reset-Synchronized PWM Mode)

Downloaded from Arrow.com.

(3) Input Capture Signal Timing

Figure 23.83 shows the input capture signal timing.

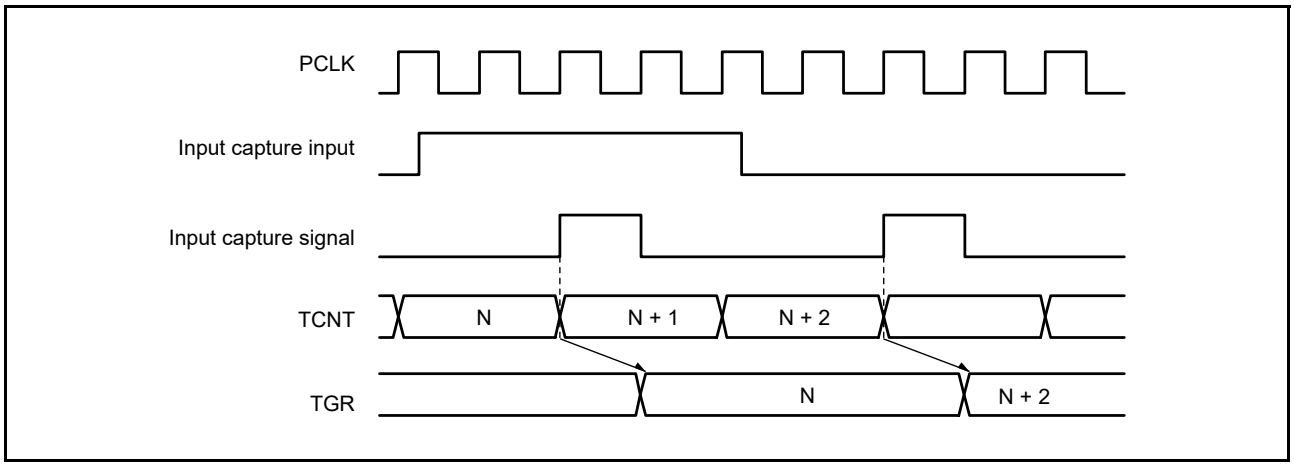


Figure 23.83 Input Capture Input Signal Timing

Downloaded from Arrow.com.

(4) Timing for Counter Clearing by Compare Match/Input Capture

Figure 23.84 show the timing when counter clearing on compare match is specified, and Figure 23.85 shows the timing when counter clearing on input capture is specified.

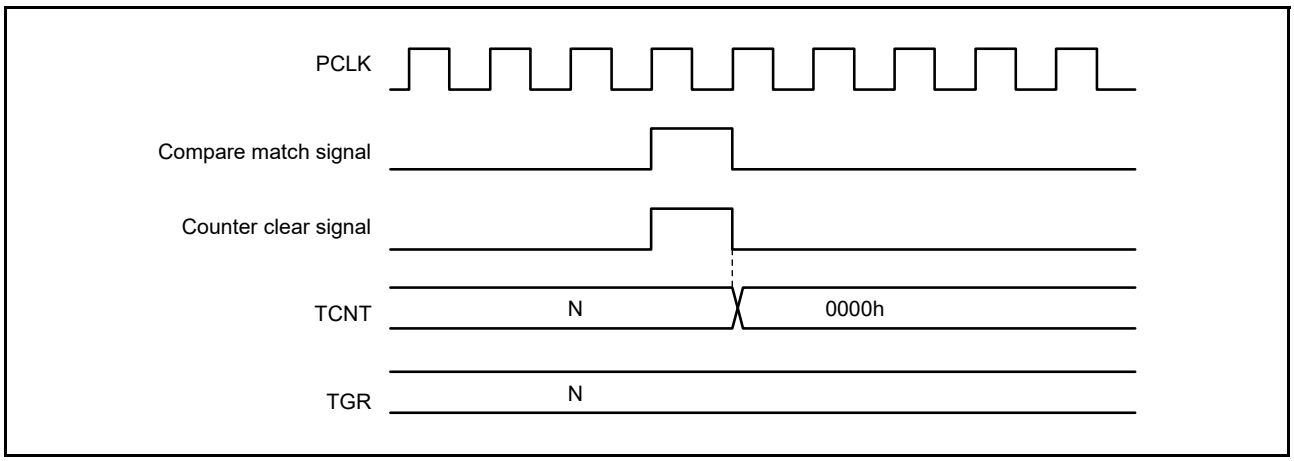


Figure 23.84 Counter Clear Timing (Compare Match) (MTU0 to MTU4)

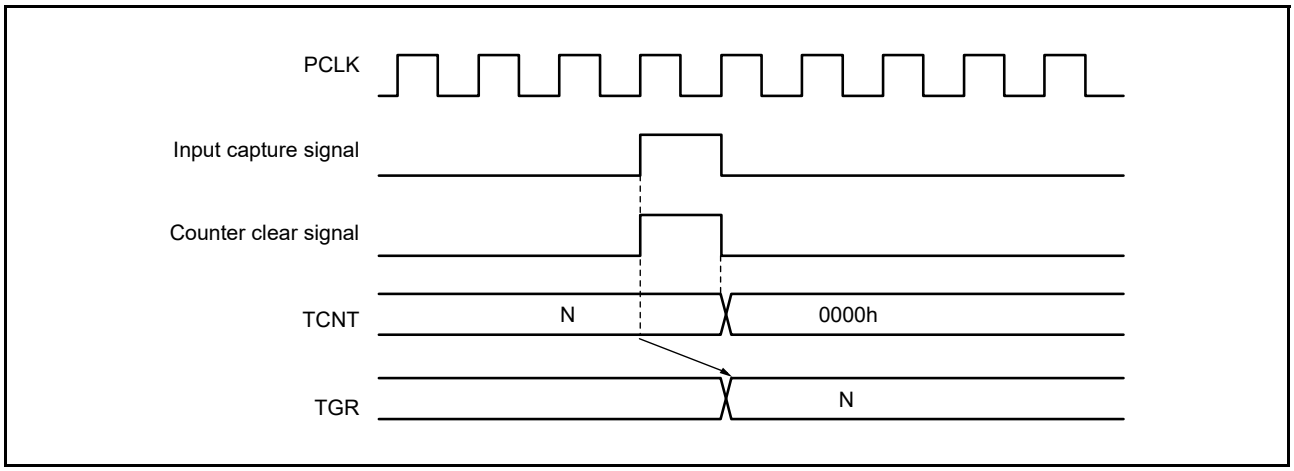


Figure 23.85 Counter Clear Timing (Input Capture) (MTU0 to MTU4)

Downloaded from Arrow.com.

(5) Buffer Operation Timing

Figure 23.86 to Figure 23.88 show the timing in buffer operation.

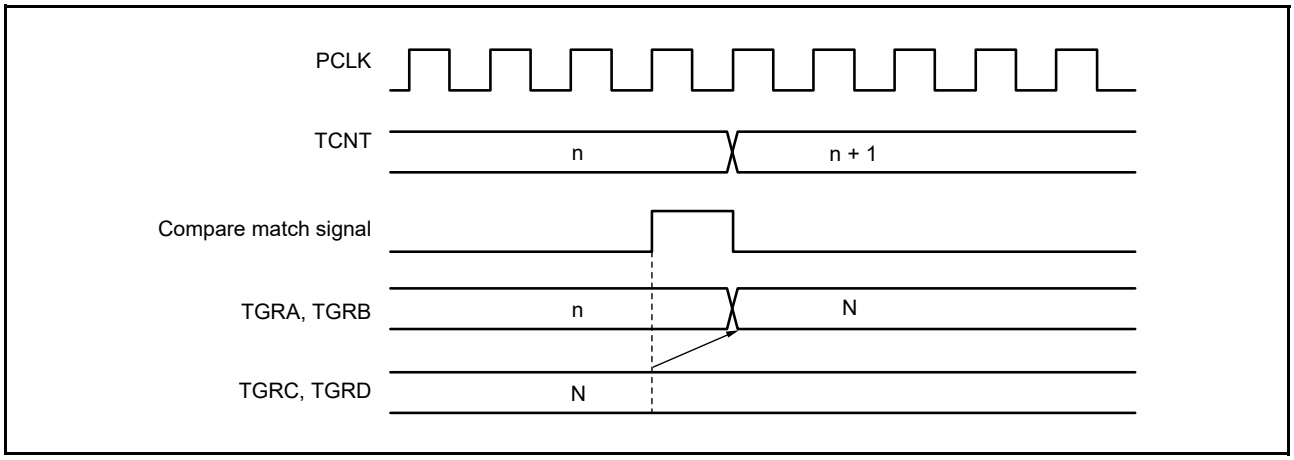


Figure 23.86 Buffer Operation Timing (Compare Match)

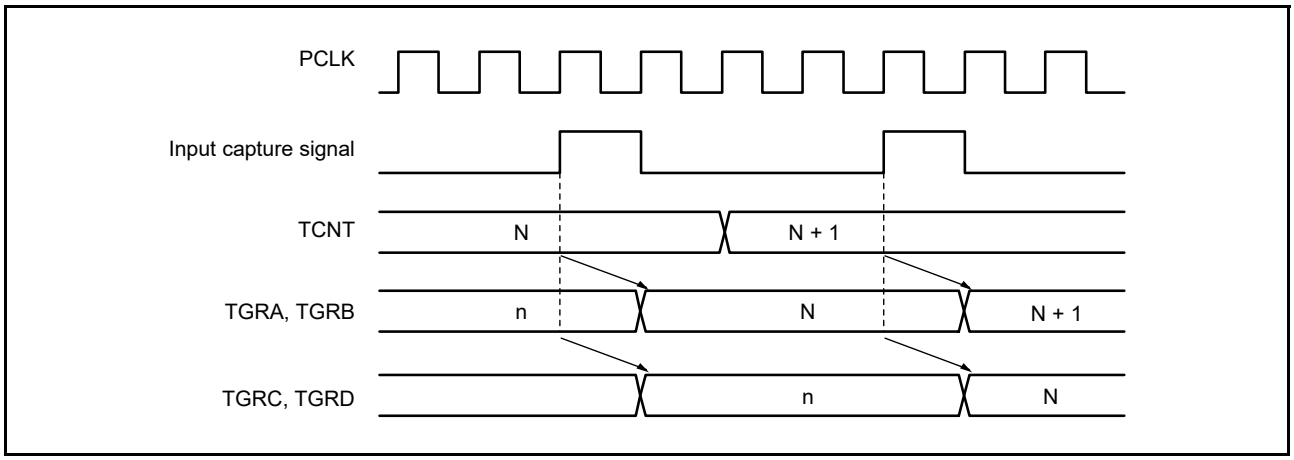


Figure 23.87 Buffer Operation Timing (Input Capture)

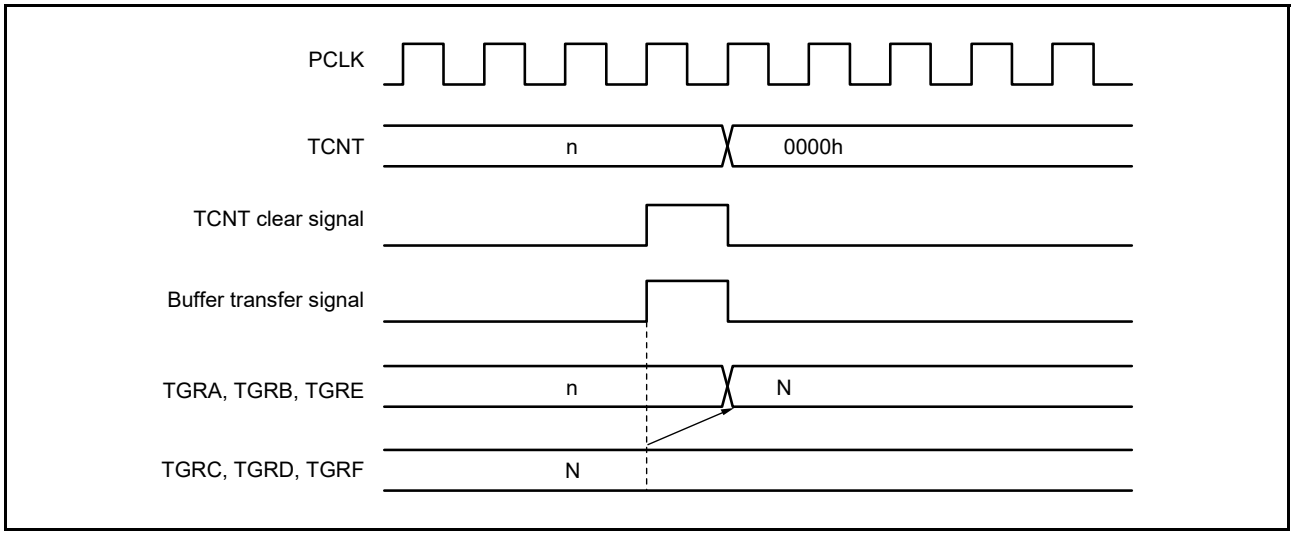


Figure 23.88 Buffer Operation Timing (When TCNT Cleared)

Downloaded from Arrow.com.

(6) Buffer Transfer Timing (Complementary PWM Mode)

Figure 23.89 to Figure 23.91 show the buffer transfer timing in complementary PWM mode.

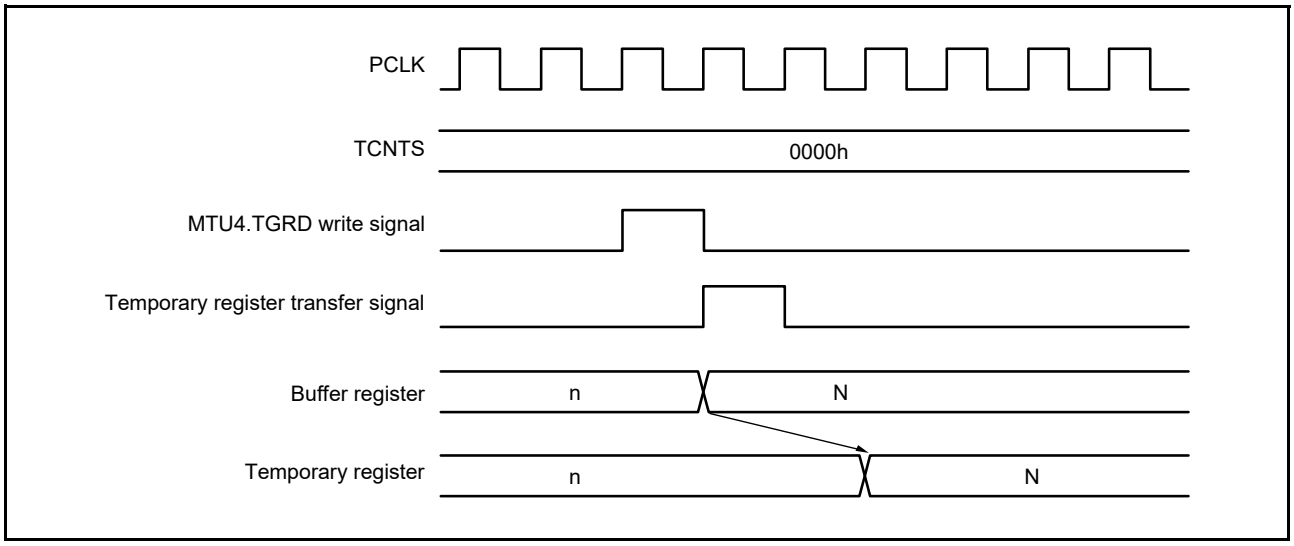


Figure 23.89 Transfer Timing from Buffer Register to Temporary Register (TCNTS Stop)

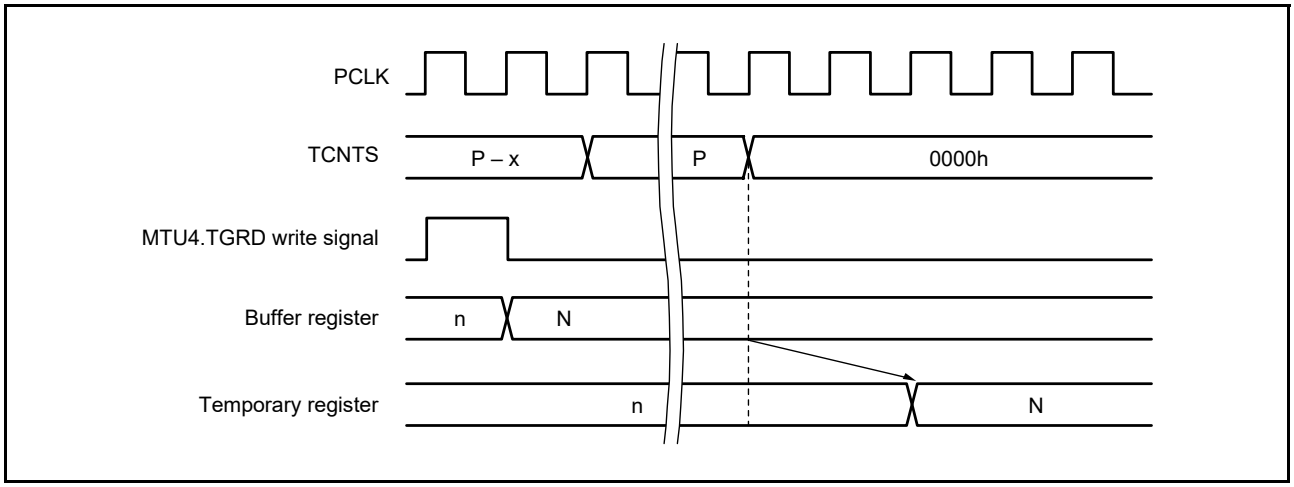


Figure 23.90 Transfer Timing from Buffer Register to Temporary Register (TCNTS Operating)

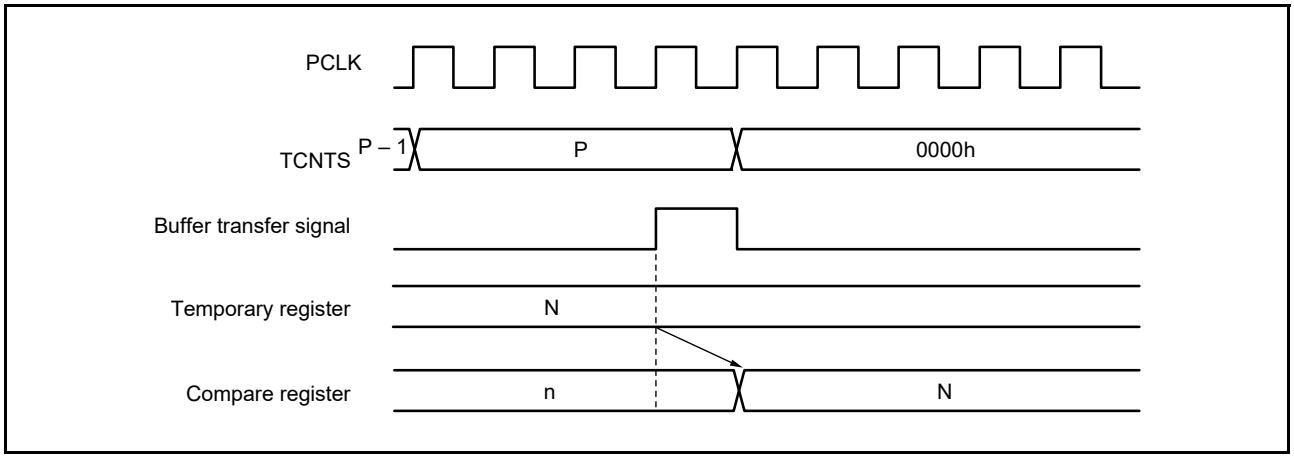


Figure 23.91 Transfer Timing from Temporary Register to Compare Register

Downloaded from Arrow.com.

23.5.2 Interrupt Signal Timing

(1) Timing for TGI Interrupt by Compare Match

Figure 23.92 show the TGI interrupt request signal timing on compare match.

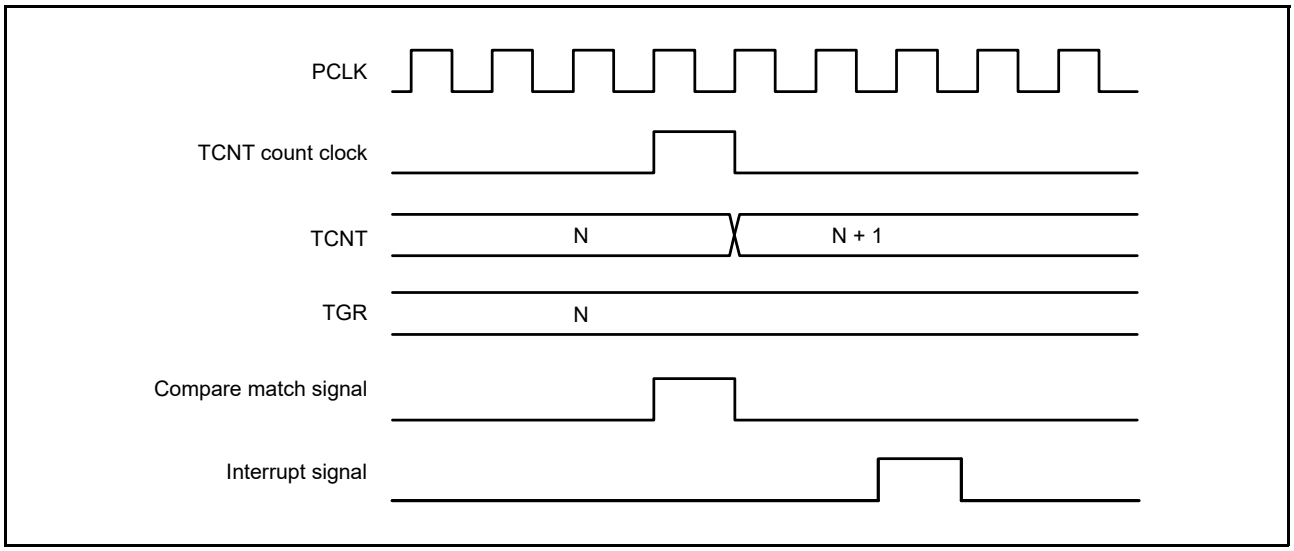


Figure 23.92 TGI Interrupt Timing (Compare Match) (MTU0 to MTU4)

(2) Timing for TGI Interrupt by Input Capture

Figure 23.93 show TGI interrupt request signal timing on input capture.

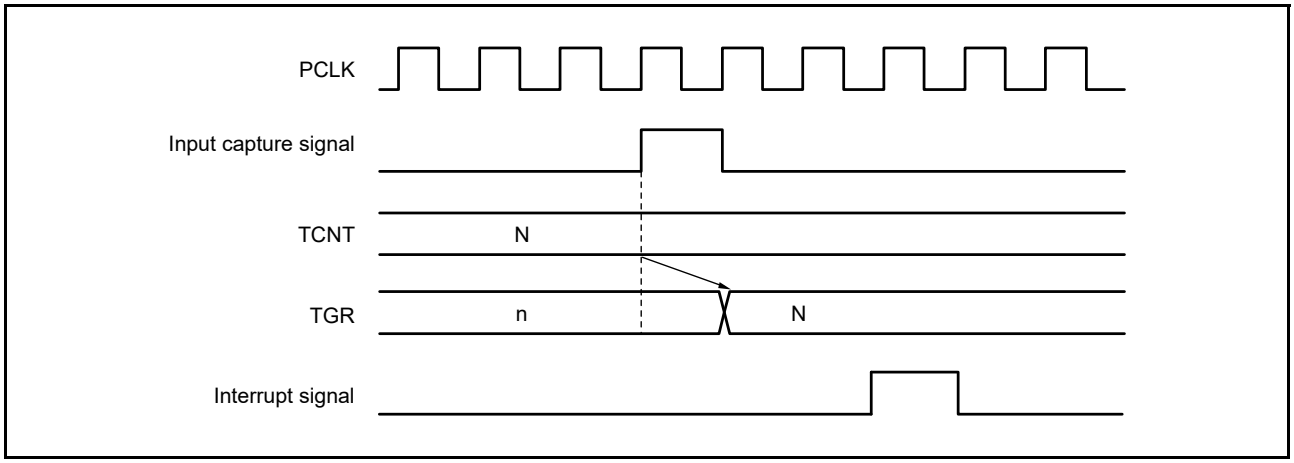


Figure 23.93 TGI Interrupt Timing (Input Capture) (MTU0 to MTU4)

Downloaded from Arrow.com.

(3) TCIV and TCIU Interrupt Timing

Figure 23.94 shows the TCIV interrupt request signal timing on overflow.
Figure 23.95 shows the TCIU interrupt request signal timing on underflow.

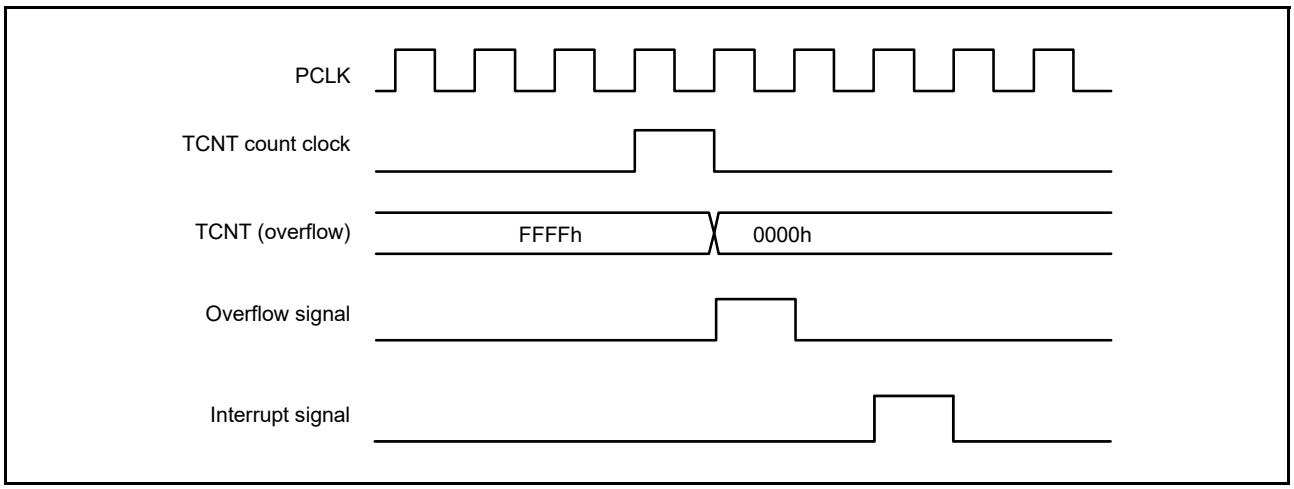


Figure 23.94 TCIV Interrupt Timing

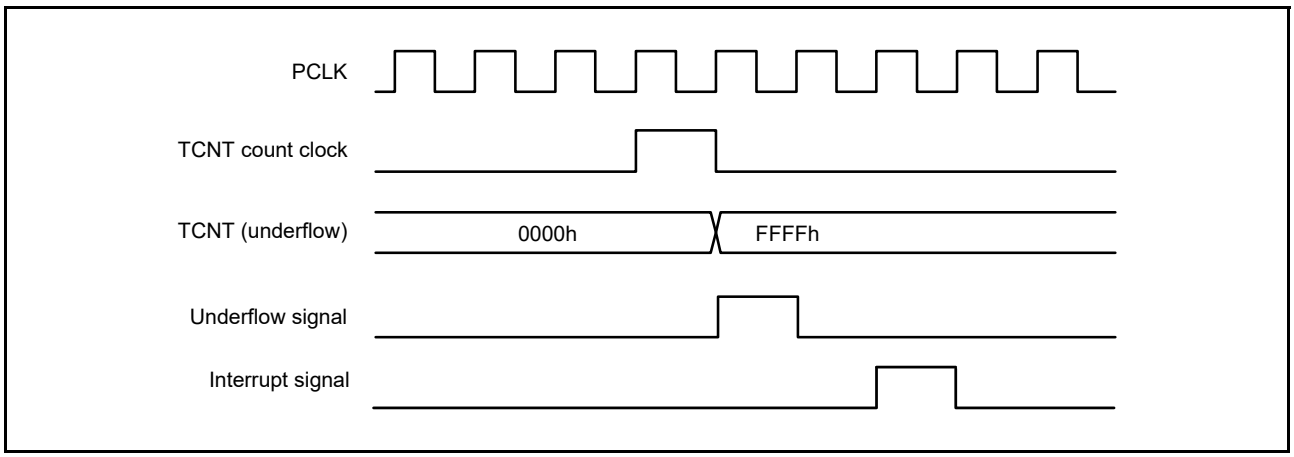


Figure 23.95 TCIU Interrupt Timing

Downloaded from Arrow.com.

23.6 Usage Notes

23.6.1 Module Clock Stop Mode Setting

MTU operation can be disabled or enabled using the module stop control register. MTU operation is stopped with the initial setting. Register access is enabled by releasing the module clock stop mode. For details, refer to section 11, Low Power Consumption.

23.6.2 Count Clock Restrictions

The count clock source pulse width must be at least 1.5 PCLK cycles for single-edge detection, and at least 2.5 PCLK cycles for both-edge detection. The MTU will not operate properly at narrower pulse widths.
In phase counting mode, the phase difference and overlap between the two input clocks must be at least 1.5 PCLK cycles, and the pulse width must be at least 2.5 PCLK cycles. Figure 23.96 shows the input clock conditions in phase counting mode.

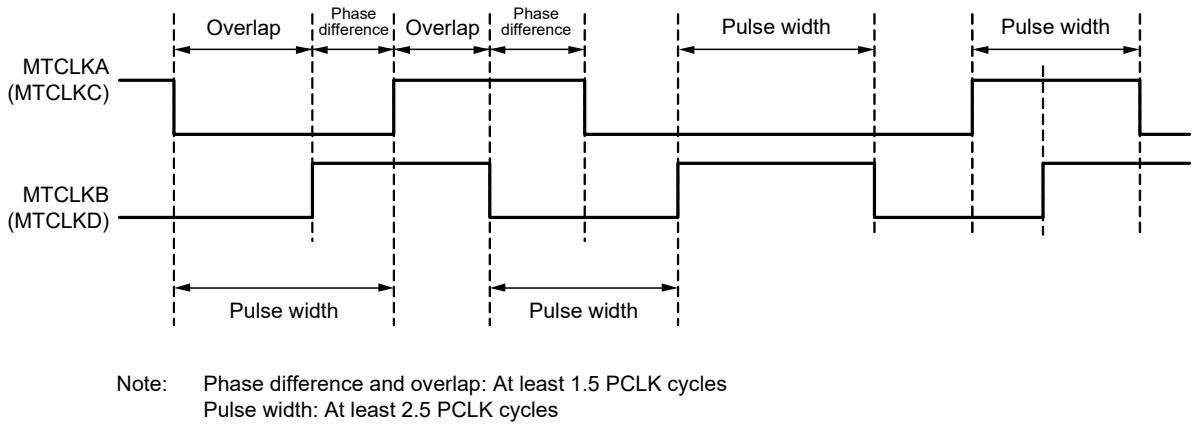


Figure 23.96 Phase Difference, Overlap, and Pulse Width in Phase Counting Mode

23.6.3 Notes on Cycle Setting

When counter clearing on compare match is set, the TCNT counter is cleared in the final state in which it matches the TGR register value (the point at which the TCNT counter updates the matched count value). Consequently, the actual counter frequency is given by the following formula:

- MTU0 to MTU4

$$f = \frac{CNTCLK}{N + 1}$$

f: Counter frequency
CNTCLK: The count clock frequency set by the TCR.TPSC[2:0] bits
N: The TGR register setting

Downloaded from Arrow.com.

23.6.4 Contention between TCNT Write and Clear Operations

If the counter clear signal is generated in a TCNT write cycle, the TCNT counter clearing takes precedence and the TCNT counter write operation is not performed.
Figure 23.97 shows the timing in this case.

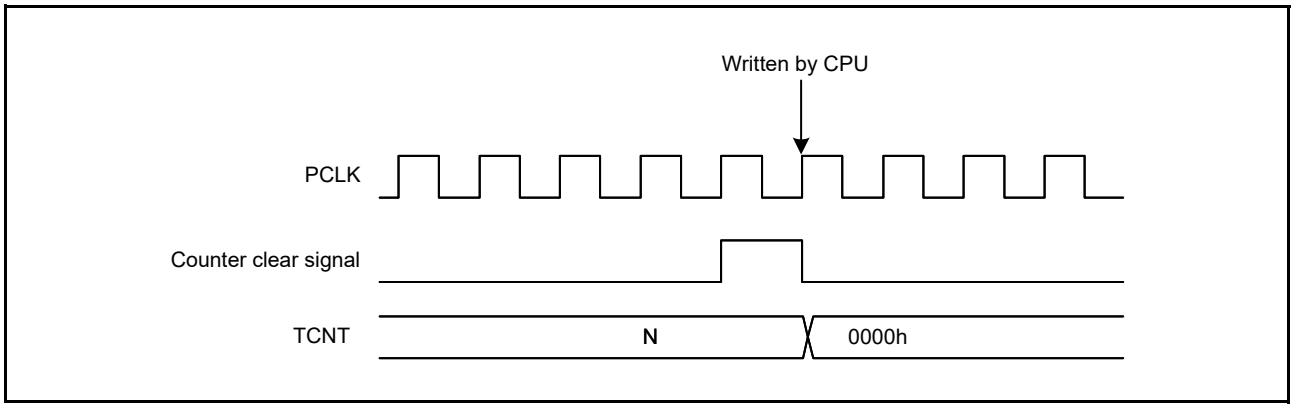


Figure 23.97 Contention between TCNT Write and Counter Clear Operations

23.6.5 Contention between TCNT Write and Increment Operations

If incrementing occurs in a TCNT write cycle, the TCNT counter write operation takes precedence and the TCNT counter is not incremented.
Figure 23.98 shows the timing in this case.

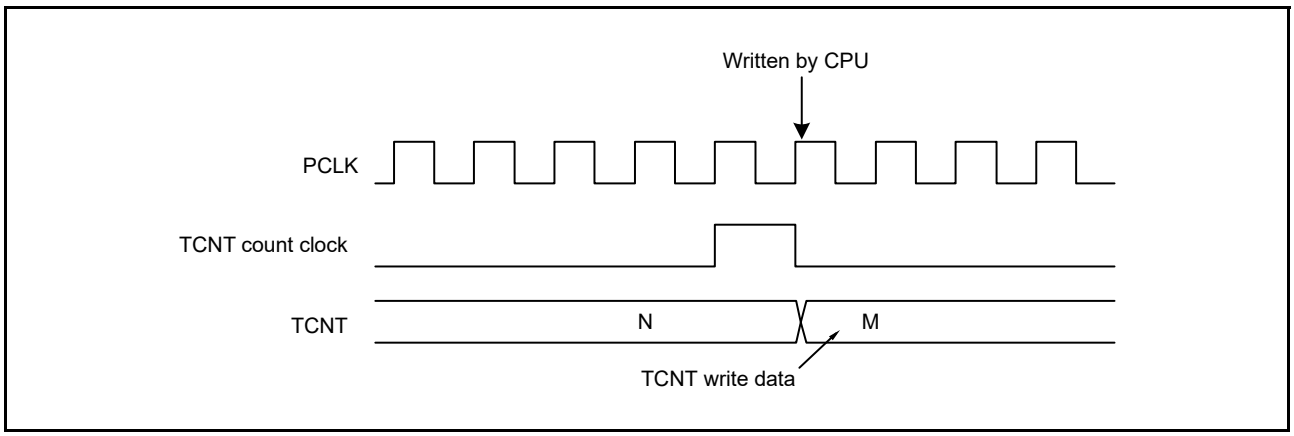


Figure 23.98 Contention between TCNT Write and Increment Operations

Downloaded from Arrow.com.

23.6.6 Contention between TGR Write Operation and Compare Match

If a compare match occurs in a TGR write cycle, the TGR register write operation is executed and the compare match signal is also generated.
Figure 23.99 shows the timing in this case.

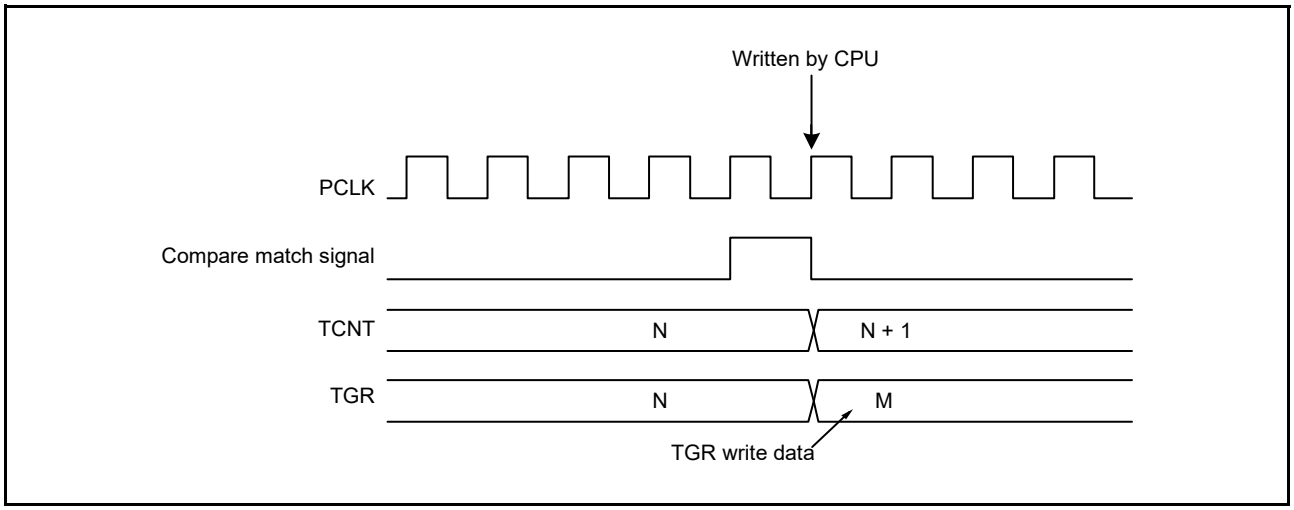


Figure 23.99 Contention between TGR Write Operation and Compare Match

23.6.7 Contention between Buffer Register Write Operation and Compare Match

If a compare match occurs in a TGR write cycle, the data before write operation is transferred to the TGR register by the buffer operation.
Figure 23.100 shows the timing in this case.

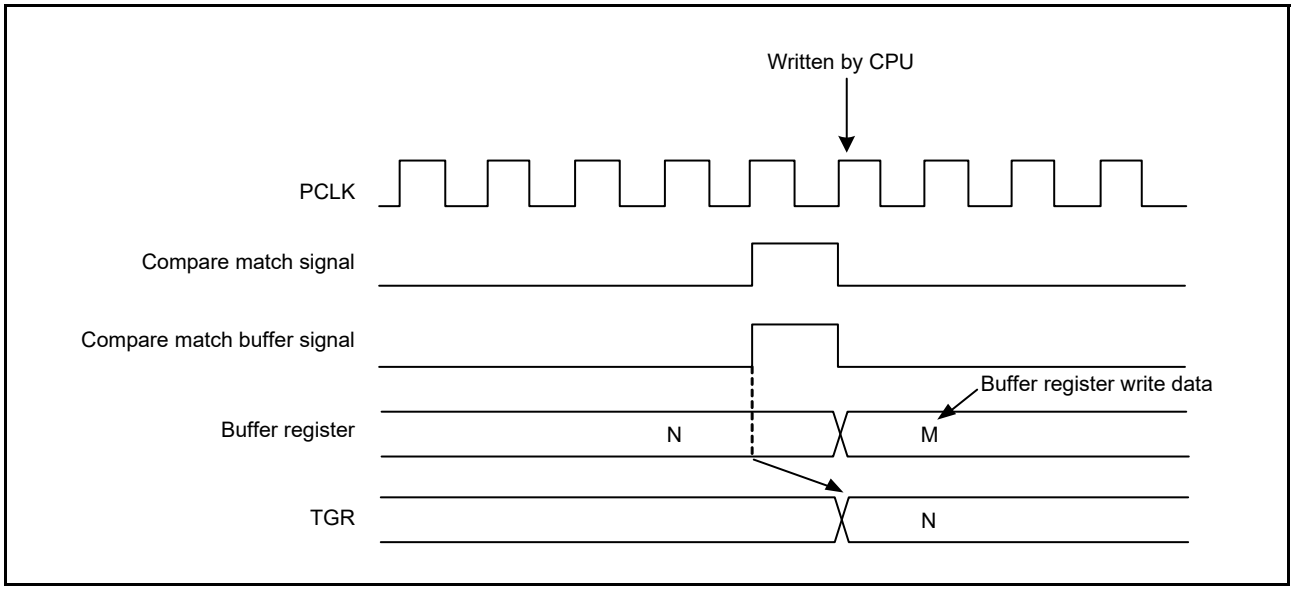


Figure 23.100 Contention between Buffer Register Write Operation and Compare Match

Downloaded from Arrow.com.

23.6.8 Contention between Buffer Register Write and TCNT Clear Operations

When the buffer transfer timing is set at the TCNT clear timing by the timer buffer operation transfer mode register (TBTM), if TCNT clearing occurs in a TGR write cycle, the data before write operation is transferred to TGR by the buffer operation.
Figure 23.101 shows the timing in this case.

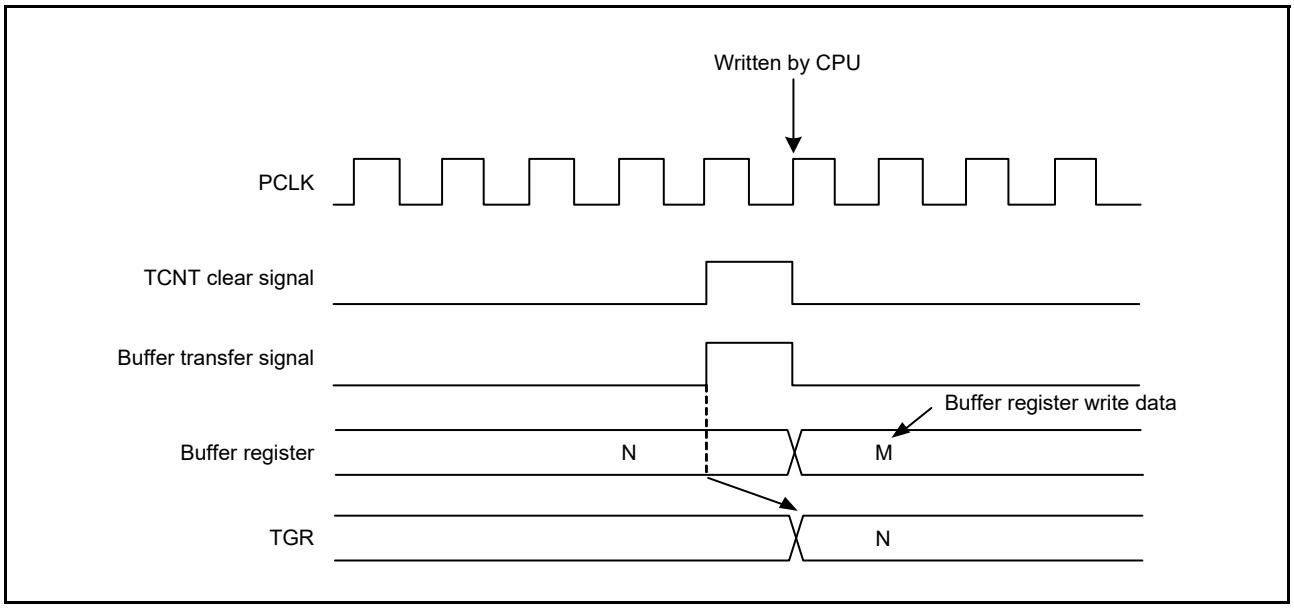


Figure 23.101 Contention between Buffer Register Write and TCNT Clear Operations

23.6.9 Contention between TGR Read Operation and Input Capture

If an input capture signal is generated in a TGR read cycle, the data before input capture transfer is read.
Figure 23.102 shows the timing in this case.

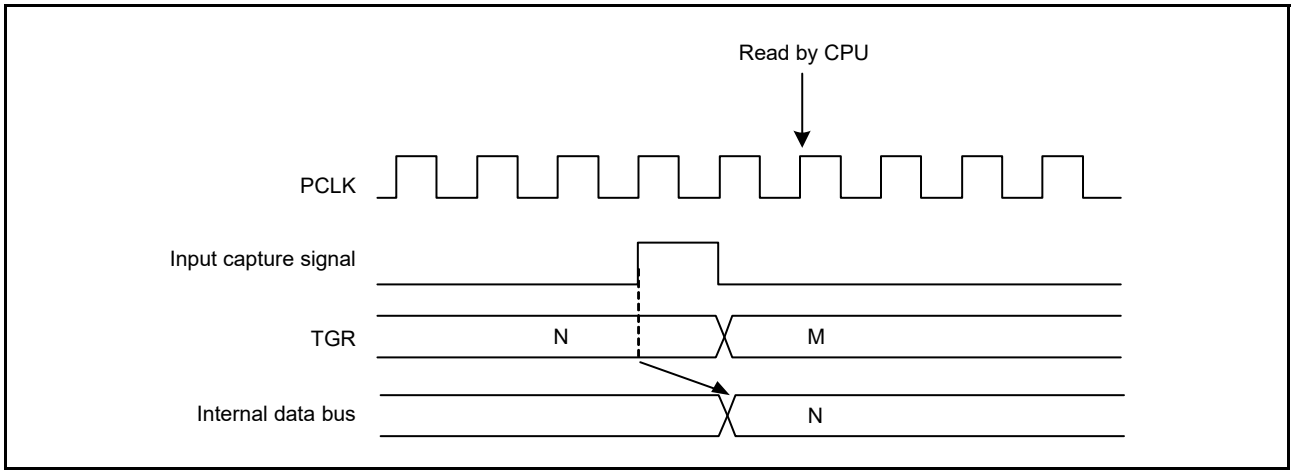


Figure 23.102 Contention between TGR Read Operation and Input Capture (MTU0 to MTU4)

Downloaded from Arrow.com.

23.6.10 Contention between TGR Write Operation and Input Capture

If an input capture signal is generated in a TGR write cycle, the input capture operation takes precedence and the TGR register write operation is not performed.
Figure 23.103 show the timing in this case.

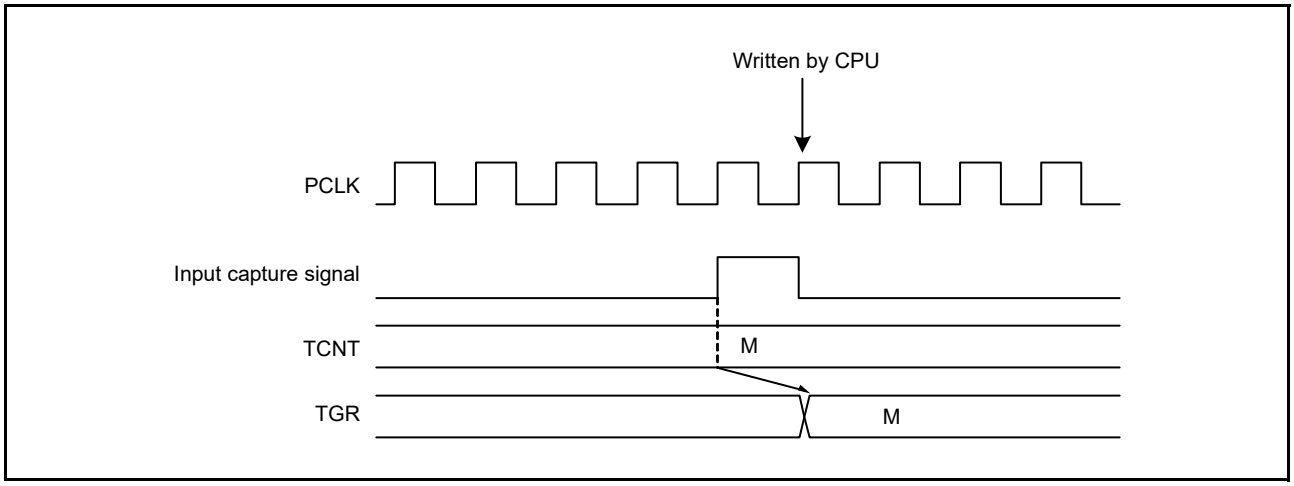


Figure 23.103 Contention between TGR Write Operation and Input Capture (MTU0 to MTU4)

23.6.11 Contention between Buffer Register Write Operation and Input Capture

If an input capture signal is generated in a buffer register write cycle, the buffer operation takes precedence and the buffer register write operation is not performed.
Figure 23.104 shows the timing in this case.

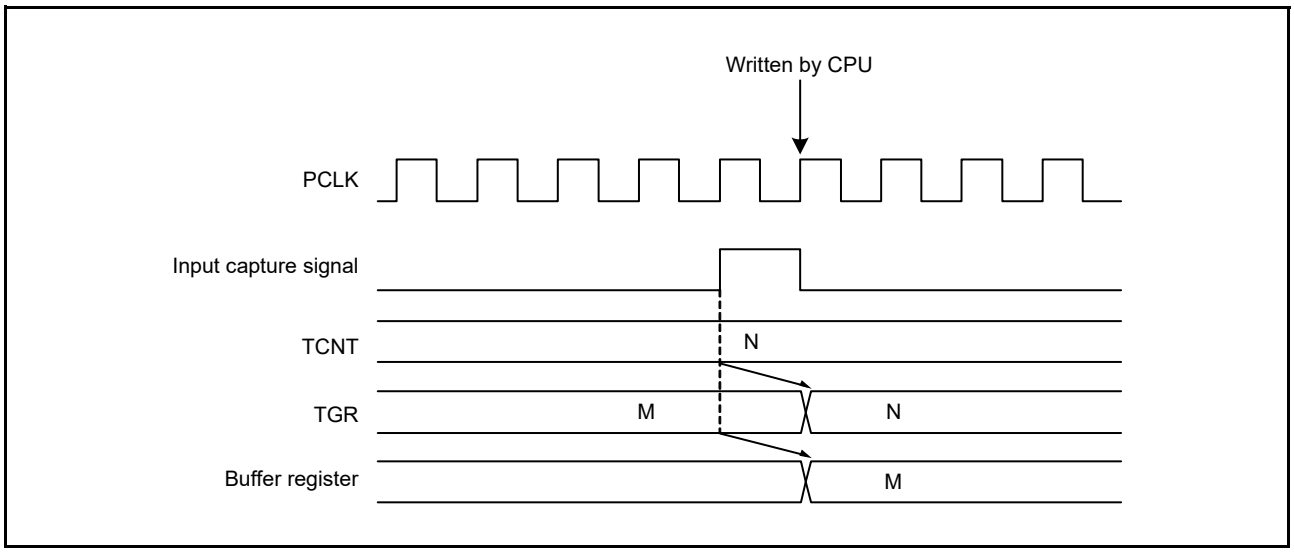


Figure 23.104 Contention between Buffer Register Write Operation and Input Capture

Downloaded from Arrow.com.

23.6.12 Contention between MTU2.TCNT Write Operation and Overflow/Underflow in Cascaded Operation

With timer counters MTU1.TCNT and MTU2.TCNT in a cascade, when a contention occurs between MTU1.TCNT counting (an MTU2.TCNT counter overflow/underflow) and the MTU2.TCNT write cycle, the MTU2.TCNT write operation is performed and the MTU1.TCNT count signal is disabled. In this case, if the MTU1.TGRA register works as a compare match register and there is a match between the MTU1.TGRA register and the MTU1.TCNT counter values, a compare match signal is issued.

Furthermore, when the MTU1.TCNT count clock is selected as the input capture source of MTU0, registers MTU0.TGRA to MTU0.TGRC work in input capture mode. In addition, when the MTU0.TGRC compare match/input capture is selected as the input capture source of the MTU1.TGRB register, the MTU1.TGRB register works in input capture mode.

Figure 23.105 shows the timing in this case.

When setting the TCNT clearing function in cascaded operation, be sure to synchronize MTU1 and MTU2.

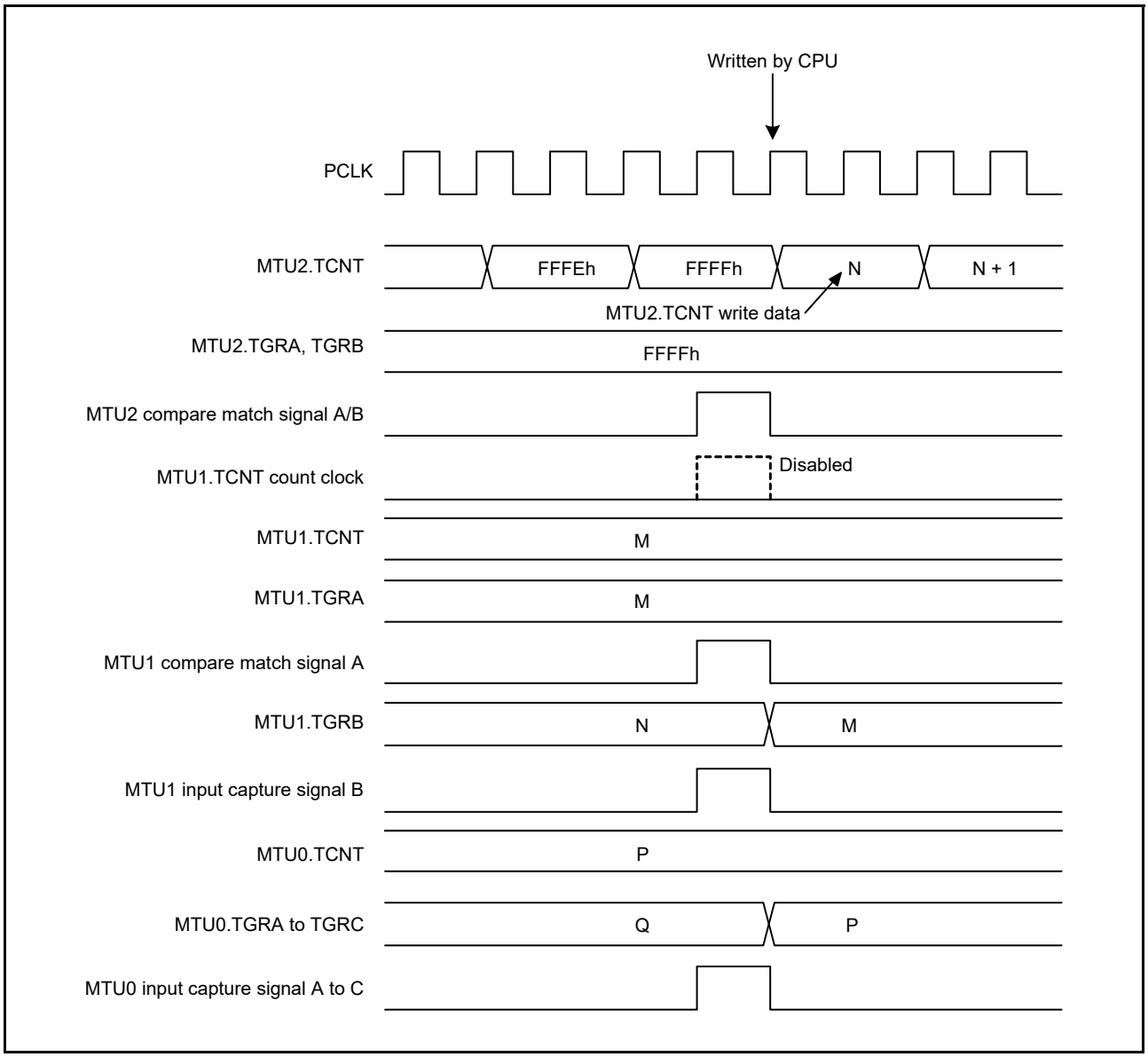


Figure 23.105 Contention between MTU2.TCNT Write Operation and Overflow/Underflow in Cascaded Operation

Downloaded from Arrow.com.

23.6.13 Counter Value When Count Operation is Stopped in Complementary PWM Mode

When counting operation in counters MTU3.TCNT and MTU4.TCNT is stopped in complementary PWM mode, the MTU3.TCNT counter is set to the TDDR register value and the MTU4.TCNT counter becomes 0000h.
When operation is restarted in complementary PWM mode, counting begins automatically from the initial setting state. Figure 23.106 shows this operation.
When counting begins in another operating mode, be sure to make initial settings in counters MTU3.TCNT and MTU4.TCNT.

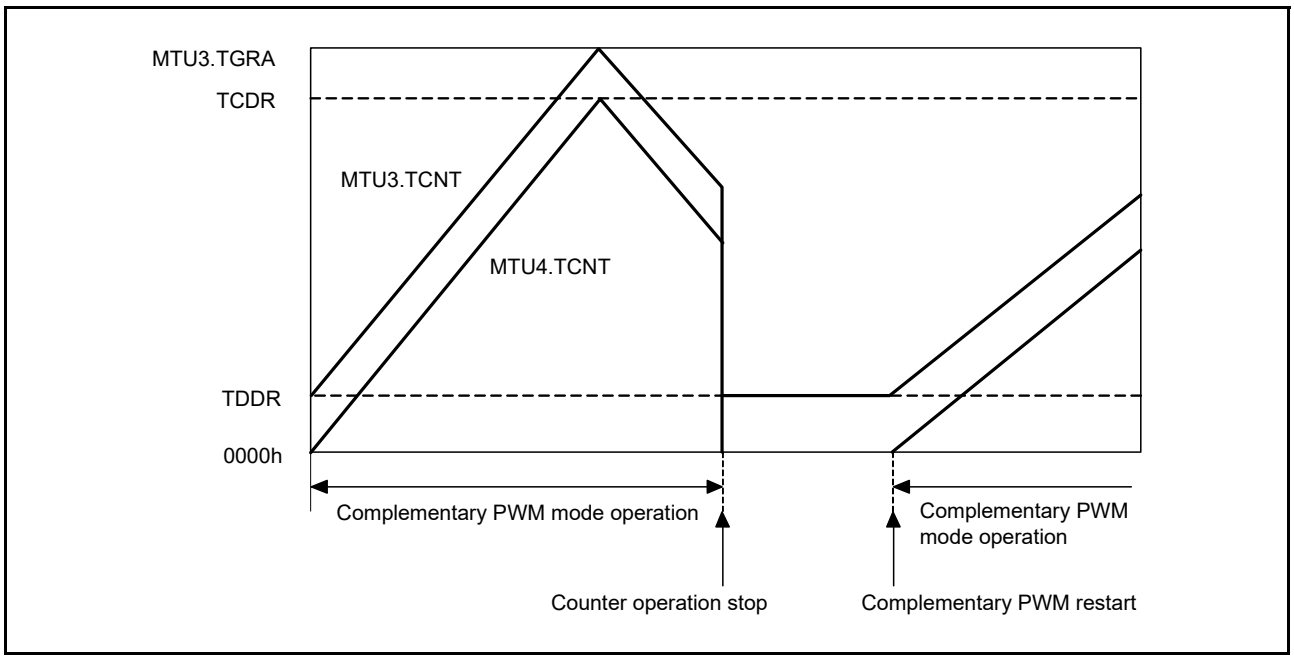


Figure 23.106 Counter Value When Stopped in Complementary PWM Mode (MTU3 and MTU4 Operation)

23.6.14 Buffer Operation Setting in Complementary PWM Mode

When modifying the PWM cycle set register (MTU3.TGRA), timer cycle data register (TCDR), and compare registers (MTU3.TGRB, MTU4.TGRA, and MTU4.TGRB) in complementary PWM mode, be sure to use buffer operation. Also, the MTU4.TMDR.BFA bit and MTU4.TMDR.BFB bit should be set to 0. Setting the MTU4.TMDR.BFA bit to 1 disables MTIOC4C pin waveform output. Setting the MTU4.TMDR.BFB bit to 1 also disables MTIOC4D pin waveform output.
In complementary PWM mode, buffer operation in MTU3 and MTU4 depends on the settings in bits BFA and BFB in the MTU3.TMDR register. When the MTU3.TMDR.BFA bit is set to 1, the MTU3.TGRC register functions as a buffer register for the MTU3.TGRA register. At the same time, the MTU4.TGRC register functions as a buffer register for the MTU4.TGRA register, and the TCBR register functions as a buffer register for the TCDR register.

Downloaded from Arrow.com.

23.6.15 Buffer Operation and Compare Match Flags in Reset-Synchronized PWM Mode

When setting buffer operation in reset-synchronized PWM mode, set the MTU4.TMDR.BFA bit and MTU4.TMDR.BFB bit to 0. Setting the MTU4.TMDR.BFA bit to 1 disables MTIOC4C pin waveform output. Setting the MTU4.TMDR.BFB bit to 1 also disables MTIOC4D pin waveform output.

In reset-synchronized PWM mode, buffer operation in MTU3 and MTU4 depends on the settings in the MTU3.TMDR.BFA bit and MTU3.TMDR.BFB bit. For example, if the MTU3.TMDR.BFA bit is set to 1, the MTU3.TGRC register functions as a buffer register for the MTU3.TGRA register. At the same time, the MTU4.TGRC register functions as a buffer register for the MTU4.TGRA register.

While the MTU3.TGRC and MTU3.TGRD registers are operating as buffer registers, the corresponding TGIC and TGID interrupt requests are never generated.

Figure 23.107 shows an example of the MTU3.TGR and MTU4.TGR registers, MTIOC3m, and MTIOC4m operation with the MTU3.TMDR.BFA bit and MTU3.TMDR.BFB bit set to 1 and the MTU4.TMDR.BFA bit and MTU4.TMDR.BFB bit set to 0. (m = A to D)

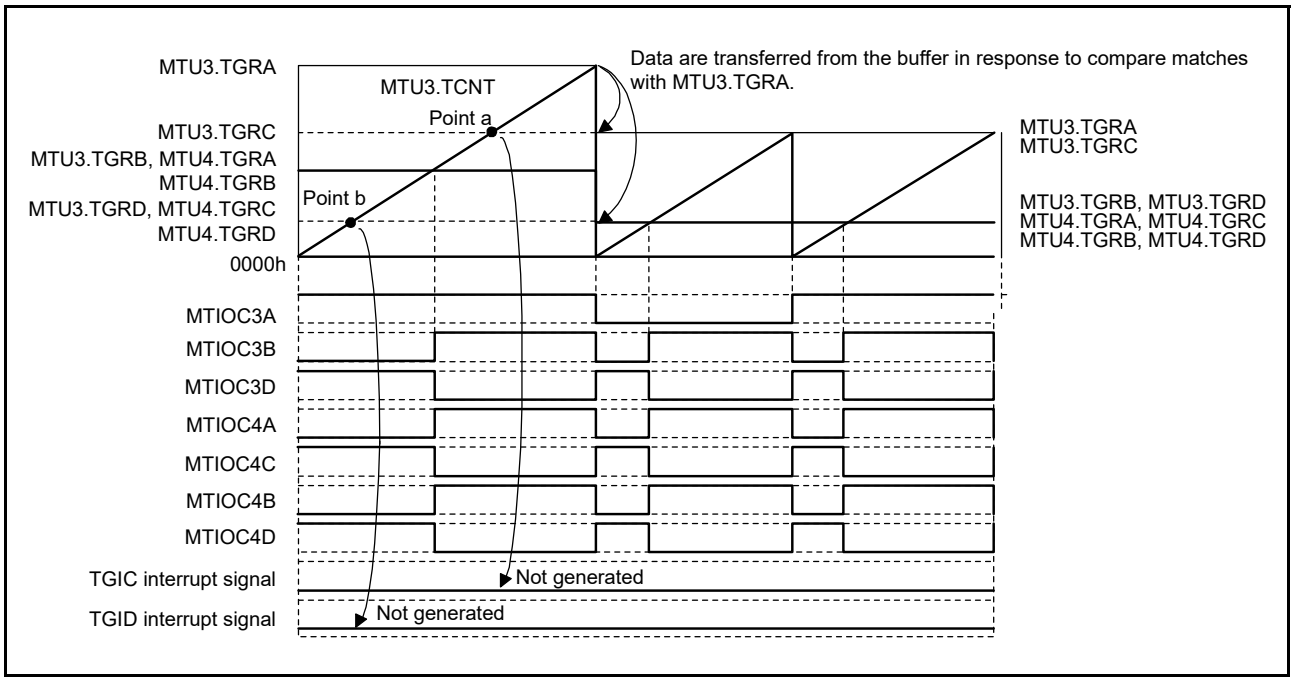


Figure 23.107 Buffer Operation and Compare Match Flags in Reset-Synchronized PWM Mode

Downloaded from Arrow.com.

23.6.16 Overflow Flags in Reset-Synchronized PWM Mode

After reset-synchronized PWM mode is selected, counters MTU3.TCNT and MTU4.TCNT start counting when the TSTR.CST3 bit is set to 1. In this state, the MTU4.TCNT count clock source and count edge are determined by the MTU3.TCR register setting.

In reset-synchronized PWM mode, with cycle register MTU3.TGRA set to FFFFh and the MTU3.TGRA compare match selected as the counter clearing source, counters MTU3.TCNT and MTU4.TCNT count up to FFFFh, then a compare match occurs with the MTU3.TGRA register, and counters MTU3.TCNT and MTU4.TCNT are both cleared. In this case, the corresponding TCIV interrupt request is not generated.

Figure 23.108 shows an operation example in reset-synchronized PWM mode with cycle register MTU3.TGRA set to FFFFh and the MTU3.TGRA compare match specified for the counter clearing source.

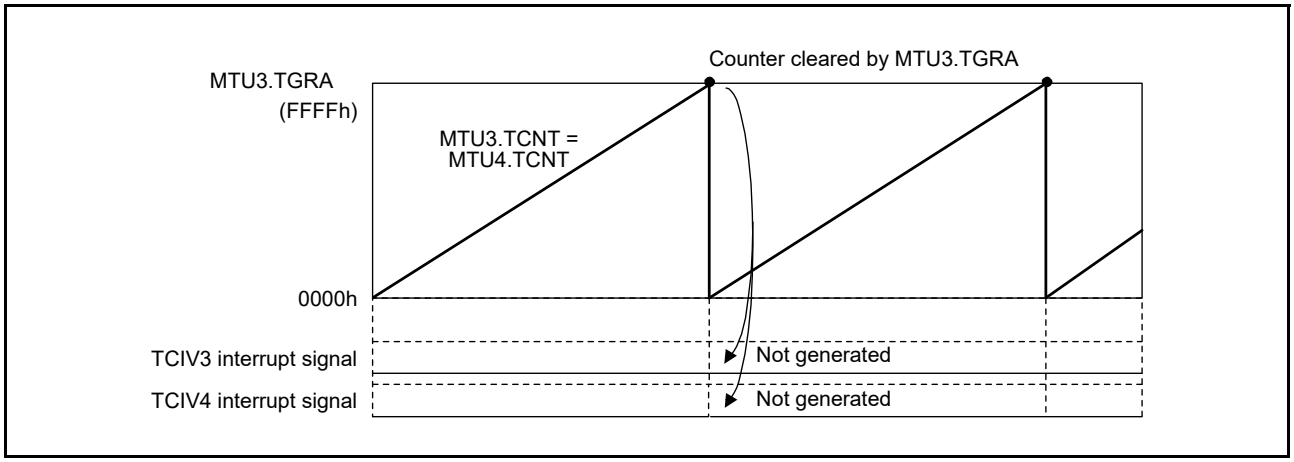


Figure 23.108 Overflow Flags in Reset-Synchronized PWM Mode

Downloaded from Arrow.com.

23.6.17 Contention between Overflow/Underflow and Counter Clearing

If an overflow/underflow and counter clearing occur simultaneously, the TCNT counter clearing takes precedence and the corresponding TCIV interrupt is not generated. If an overflow and counter clearing due to an input capture occur simultaneously, an input capture interrupt signal is output and an overflow interrupt signal is not output.
Figure 23.109 shows the operation timing when a TGR compare match is specified as the clearing source and the TGR register is set to FFFFh.

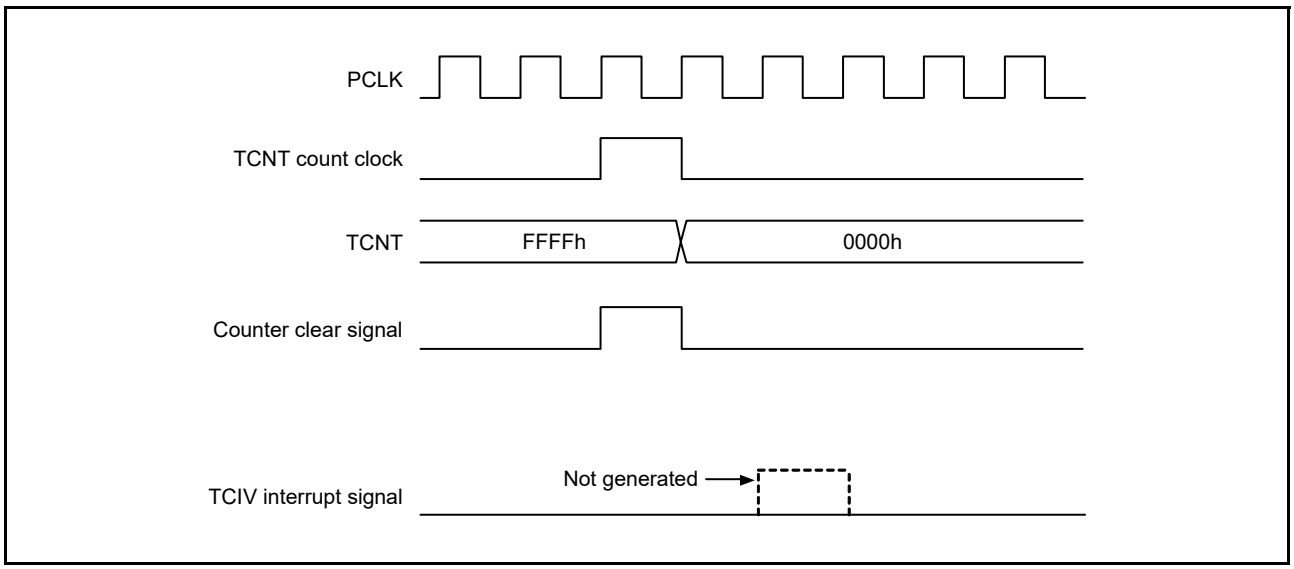


Figure 23.109 Contention between Overflow and Counter Clearing

23.6.18 Contention between TCNT Write Operation and Overflow/Underflow

If TCNT up-count or down-count in a TCNT write cycle and an overflow or an underflow occurs, the TCNT write operation takes precedence. The corresponding interrupt is not generated.
Figure 23.110 shows the operation timing when there is contention between TCNT write operation and overflow.

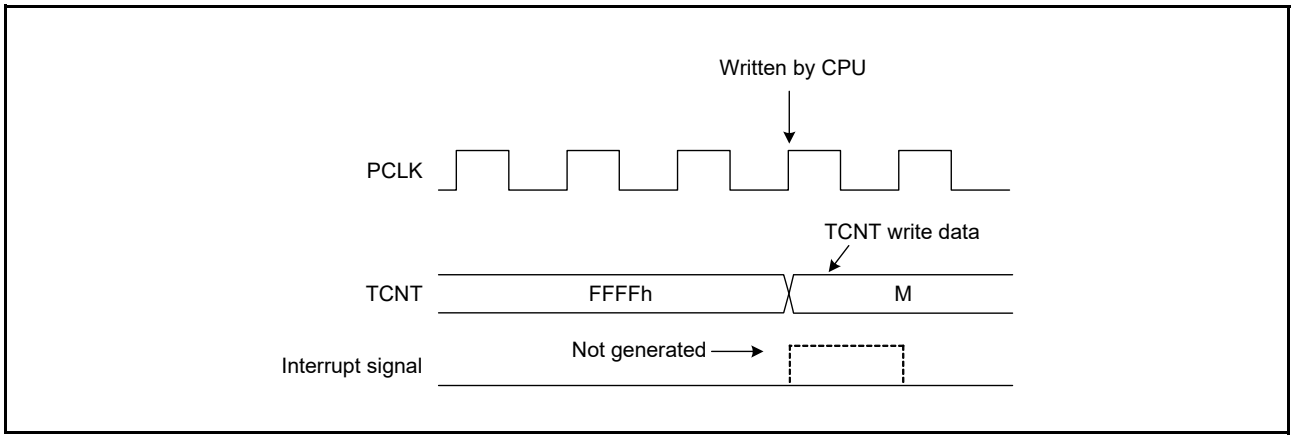


Figure 23.110 Contention between TCNT Write Operation and Overflow

Downloaded from Arrow.com.

23.6.19 Notes on Transition from Normal Mode or PWM Mode 1 to Reset-Synchronized PWM Mode

When making a transition from normal mode or PWM mode 1 to reset-synchronized PWM mode in MTU3 and MTU4, if the counter is stopped while the output pins (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4C, MTIOC4B, and MTIOC4D) are held at a high level and then operation is started after a transition to reset-synchronized PWM mode, the initial pin output will not be correct.
When making a transition from normal mode to reset-synchronized PWM mode, write 11h to registers MTU3.TIORH, MTU3.TIORL, MTU4.TIORH, and MTU4.TIORL to initialize the output pin state to a low level, then set the registers to the initial value (00h) before making the mode transition.
When making a transition from PWM mode 1 to reset-synchronized PWM mode, switch to normal mode, initialize the output pin state to a low level, and then set the registers to the initial value (00h) before making the transition to reset-synchronized PWM mode.

23.6.20 Output Level in Complementary PWM Mode or Reset-Synchronized PWM Mode

When complementary PWM mode or reset-synchronized PWM mode is selected for MTU3 or MTU4, use the TOCR1.OLSP bit and TOCR1.OLSN bit to set the levels for PWM waveform output. Also, when either of these modes is in use, set the TIOR register to 00h. The negative-phase output level when the TDER.TDER bit is set to 0 (no dead time is generated) in complementary PWM mode is the inverse of the positive-phase output level according to the TOCR1.OLSP bit setting, not the TOCR1.OLSN bit setting.

23.6.21 Interrupts during Periods in the Module Stop State

When an module that has issued an interrupt request enters the module stop state, clearing the source of the interrupt for the CPU or activation signal for the DTC/DMAC is not possible.
Accordingly, disable interrupts, etc. before making the settings for the module stop state.

23.6.22 Simultaneous Input Capture in MTU1.TCNT and MTU2.TCNT in Cascade Connection

When counters MTU1.TCNT and MTU2.TCNT operate as a 32-bit counter in cascade connection, the cascaded counter value cannot be captured successfully in some cases even if input-capture input is simultaneously done to MTIOC1A and MTIOC2A or to MTIOC1B and MTIOC2B. This is because the input timing of MTIOC1A and MTIOC2A or of MTIOC1B and MTIOC2B may not be the same when external input-capture signals input into counters MTU1.TCNT and MTU2.TCNT are taken in synchronization with the internal clock.
For example, the MTU1.TCNT counter (the counter for upper 16 bits) does not capture the count-up value by an overflow from the MTU2.TCNT counter (the counter for lower 16 bits) but captures the count value before the up-counting. In this case, the values of MTU1.TCNT = FFF1h and MTU2.TCNT = 0000h should be transferred to registers MTU1.TGRA and MTU2.TGRA or to registers MTU1.TGRB and MTU2.TGRB, but the values of MTU1.TCNT = FFF0h and MTU2.TCNT = 0000h are erroneously transferred.
The MTU has a function that allows simultaneous capture of counters MTU1.TCNT and MTU2.TCNT with a single input capture input. This function can be used to read the 32-bit counter such that counters MTU1.TCNT and MTU2.TCNT are captured at the same time. For details, refer to section 23.2.7, Timer Input Capture Control Register (TICCR).

Downloaded from Arrow.com.

23.6.23 Notes When Complementary PWM Mode Output Protection Functions are Not Used

The complementary PWM mode output protection functions are initially enabled. Refer to section 24, Port Output Enable 2 (POE2a), for details.

23.6.24 Points for Caution to Prevent Malfunctions in Synchronous Clearing for Complementary PWM Mode

If control of the output waveform is enabled (TWCR.WRE = 1) at the time of synchronous counter clearing in complementary PWM mode, satisfaction of either condition 1 or 2 below has the following effects.

- Dead time on the PWM output pins is shortened (or disappears).
- The active level is output on the PWM inverse-phase output pins beyond the period for active-level output.

Condition 1: In portion (10) of the initial output inhibition period in Figure 23.111, synchronous clearing occurs within the dead-time period for PWM output.

Condition 2: In portions (10) and (11) of the initial output inhibition period in Figure 23.112, synchronous clearing occurs when any condition from among MTU3.TGRB ≤ TDDR, MTU4.TGRA ≤ TDDR, or MTU4.TGRB ≤ TDDR is satisfied.

The following method avoids the above phenomena.

- Ensure that synchronous clearing proceeds with the value of each comparison register (MTU3.TGRB, MTU4.TGRA, and MTU4.TGRB) set to at least double the value of the dead time data register (TDDR).

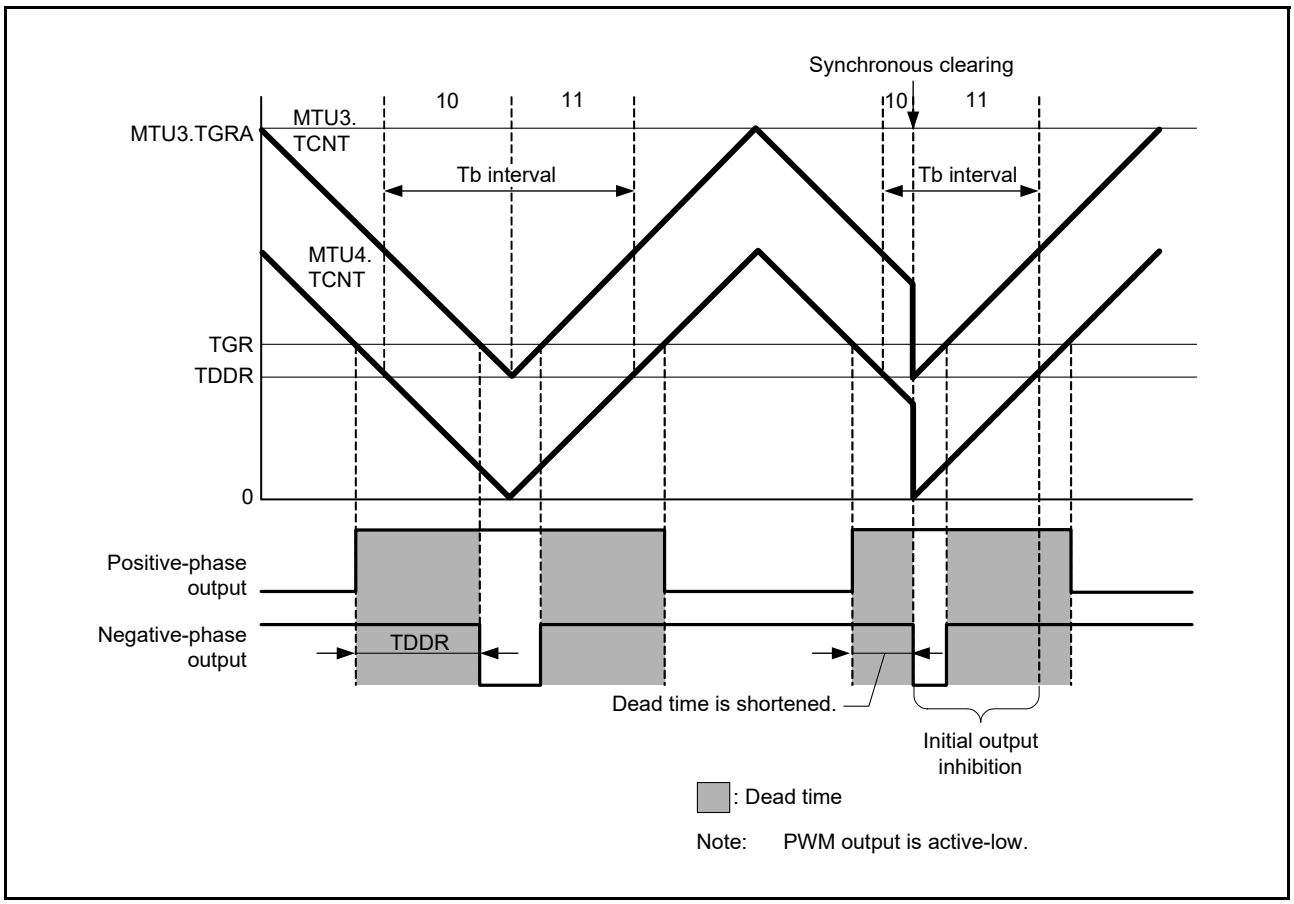


Figure 23.111 Example of Synchronous Clearing (When Condition 1 Applies)

Downloaded from Arrow.com.

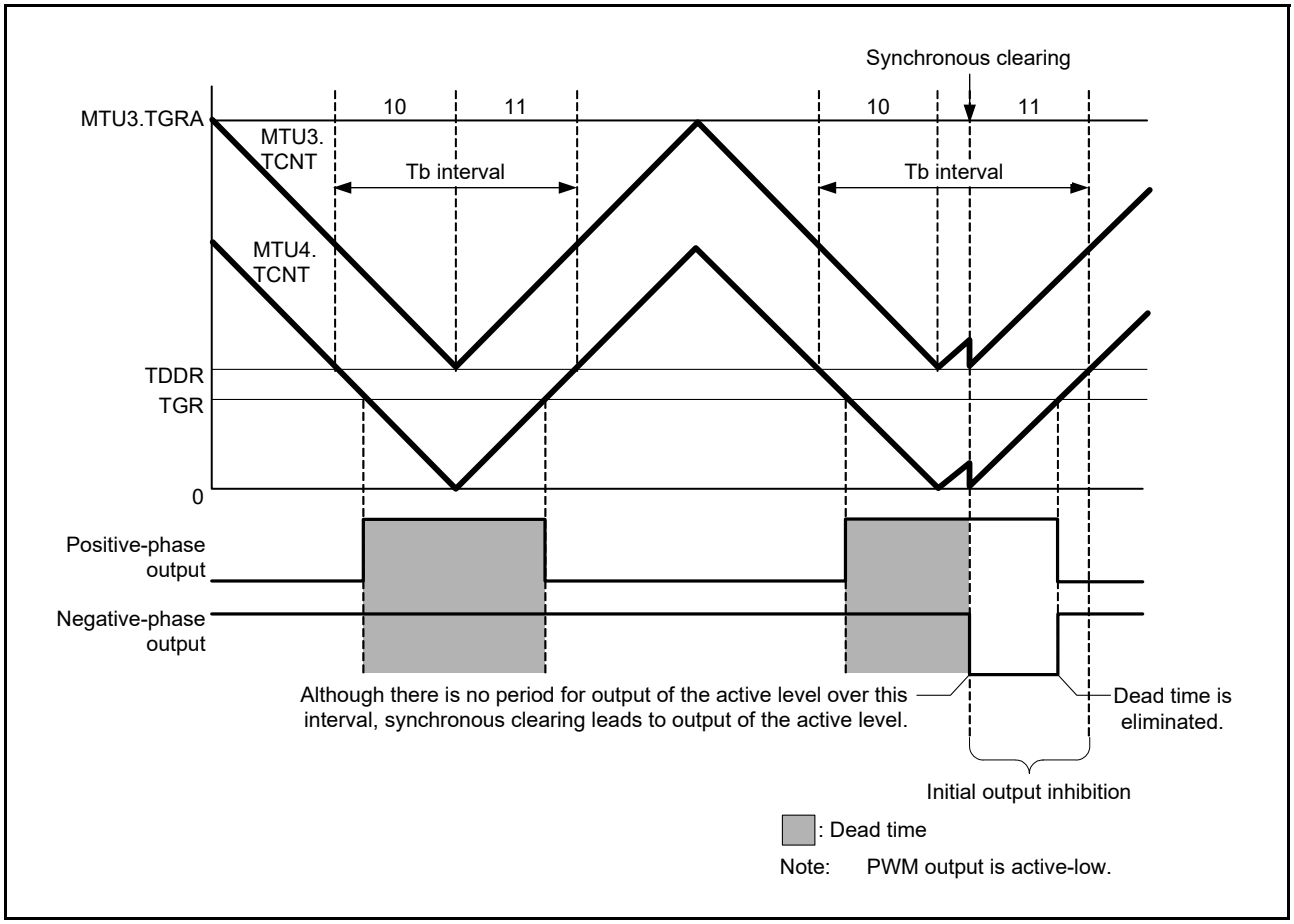


Figure 23.112 Example of Synchronous Clearing (When Condition 2 Applies)

Downloaded from Arrow.com.

23.6.25 Continuous Output of Interrupt Signal in Response to a Compare Match

When the TGR register is set to 0000h, PCLK/1 is set as the count clock, and compare match is set as the trigger for clearing of the count clock, the value of the TCNT counter remains 0000h, and the interrupt signal will be output continuously (i.e. its level will be flat) rather than output over a single cycle. Consequently, interrupts will not be detected in response to second and subsequent compare matches.

Figure 23.113 shows the timing for continuous output of the interrupt signal in response to a compare match.

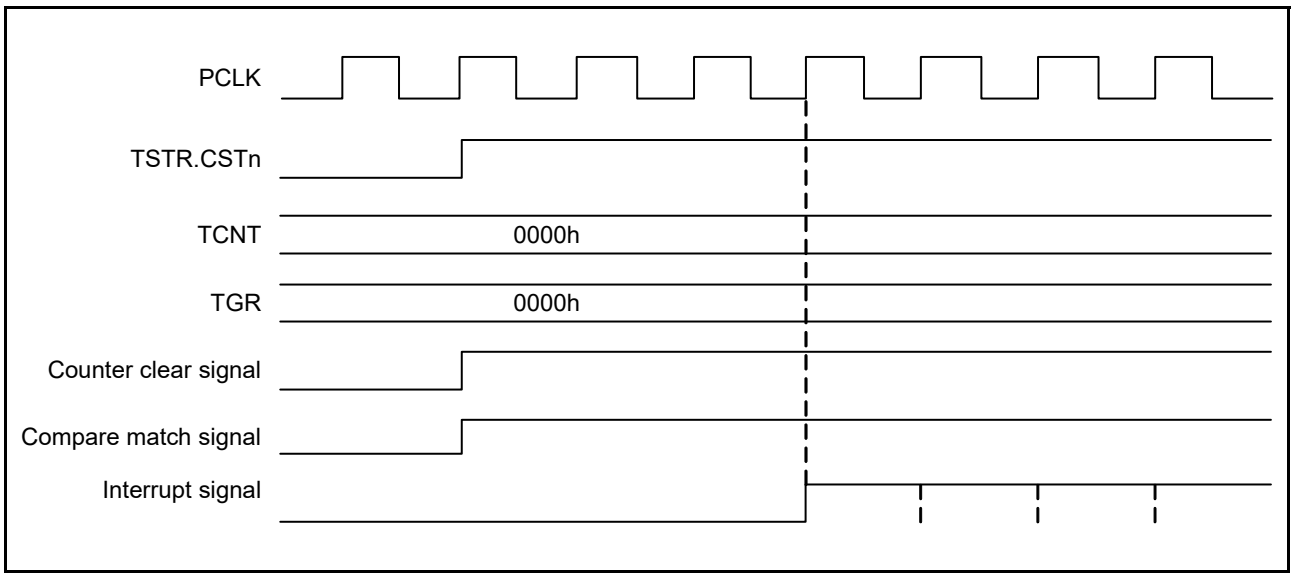


Figure 23.113 Continuous Output of Interrupt Signal in Response to a Compare Match

23.6.26 Usage Notes on A/D Converter Delaying Function in Complementary PWM Mode

- When data is transferred from a buffer register at the trough of the MTU4.TCNT counter while the MTU4.TADCOBRA and MTU4.TADCOBRB registers are set to 0 and the UT4AE and UT4BE bits in the MTU4.TADCR register are set to 1, no A/D converter start request is issued during up-counting immediately after transfer. Refer to Figure 23.114.
- When data is transferred from a buffer register at the crest of the MTU4.TCNT counter while the MTU4.TADCOBRA and MTU4.TADCOBRB registers are set to the same value as the TCDR register and the UT4AE and UT4BE bits in the MTU4.TADCR register are set to 1, no A/D converter start request is issued during down-counting immediately after transfer. Refer to Figure 23.115.
- To issue an A/D converter start request linked with interrupt skipping, set the MTU4.TADCORA and MTU4.TADCORB registers so that $2 \leq$ MTU4.TADCORA/TADCORB $\leq$ TCDR $- 2$ is satisfied.

Downloaded from Arrow.com.

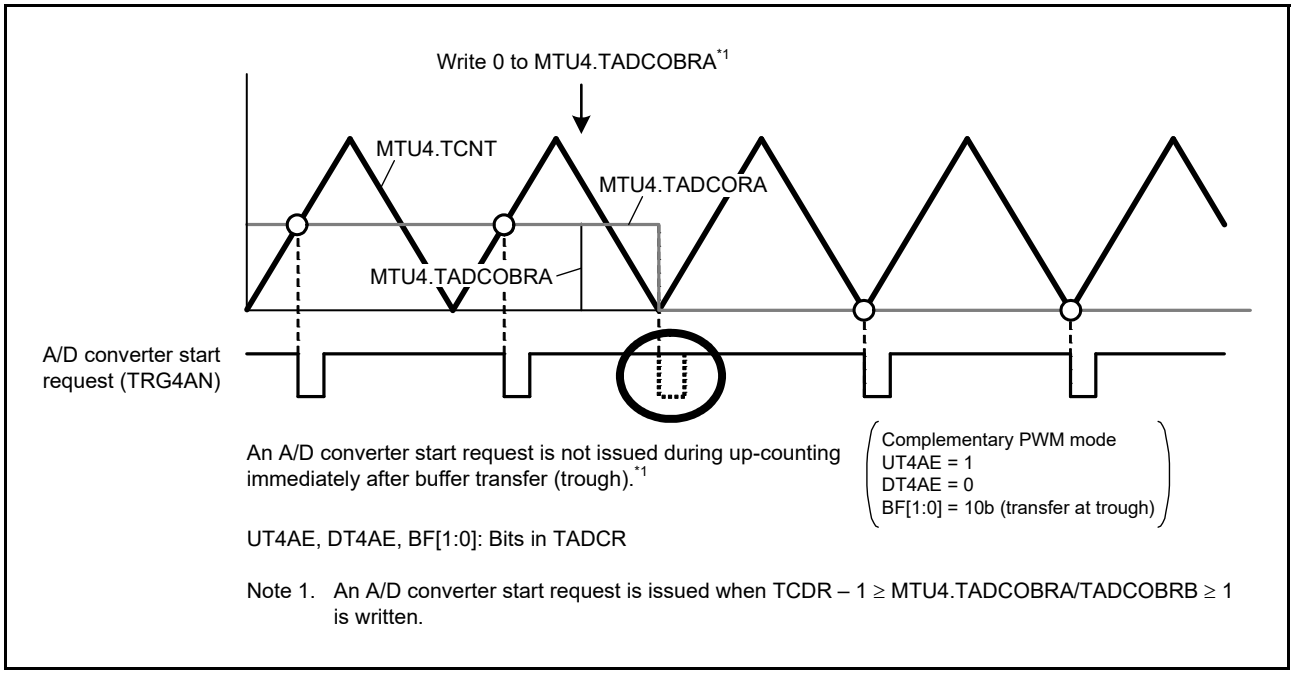


Figure 23.114 A/D Converter Start Request When 0 is Written to MTU4.TADCOBRA

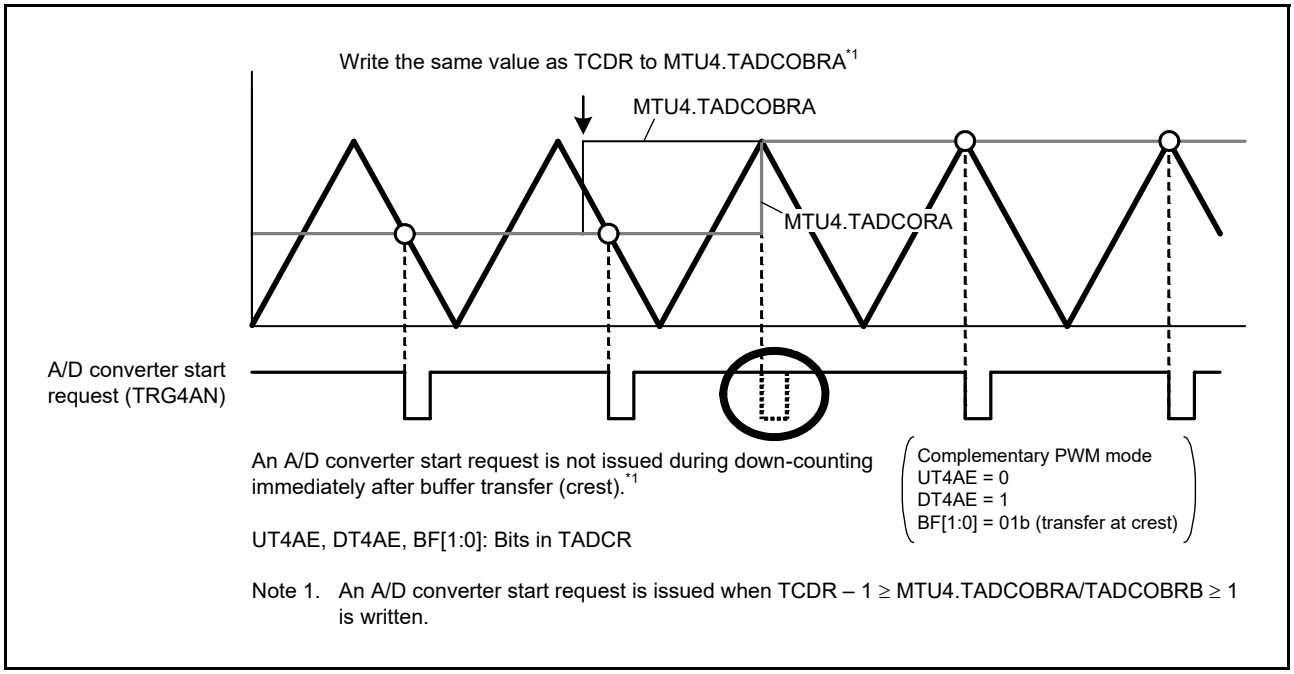


Figure 23.115 A/D Converter Start Request When the Same Value as TCDR is Written to MTU4.TADCOBRA

Downloaded from Arrow.com.

23.7 MTU Output Pin Initialization

23.7.1 Operating Modes

The MTU has the following six operating modes. Waveforms can be output in any of these modes.

- Normal mode (MTU0 to MTU4)
- PWM mode 1 (MTU0 to MTU4)
- PWM mode 2 (MTU0 to MTU2)
- Phase counting modes 1 to 4 (MTU1 and MTU2)
- Complementary PWM mode (MTU3 and MTU4)
- Reset-synchronized PWM mode (MTU3 and MTU4)

This section describes how to initialize the MTU output pins in each of these modes.

23.7.2 Operation in Case of Re-Setting Due to Error during Operation

If an error occurs during MTU operation, MTU output should be cut off by the system. For an I/O port that is shut down, set the port direction registers (PDR), the port output data register (PODR), and the port mode register (PMR) to switch the port pins to be general output pins and for output of the non-active level. Set the TIOR for the MTU pins to disable output. Set the TOER register for the complementary PWM output pins (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4B, MTIOC4C, and MTIOC4D). For PWM output pins, output can also be cut by hardware, using port output enable 2(POE). The pin initialization procedures for re-setting due to an error during operation and the procedures for restarting in a different mode after re-setting are described below.
The MTU has six operating modes, as stated above. There are thus 36 mode transition combinations, but some transitions are not available with certain channel and mode combinations. Available mode transition combinations are listed in Table 23.56.

Note that the following notations are used for operating modes.

Normal: Normal mode PWM1: PWM mode 1 PWM2: PWM mode 2
PCM: Phase counting modes 1 to 4 CPWM: Complementary PWM mode RPWM: Reset-synchronized PWM mode

Table 23.56 Mode Transition Combinations

	Normal	PWM1	PWM2	PCM	CPWM	RPWM
Normal	(1)	(2)	(3)	(4)	(5)	(6)
PWM1	(7)	(8)	(9)	(10)	(11)	(12)
PWM2	(13)	(14)	(15)	(16)	Not available	Not available
PCM	(17)	(18)	(19)	(20)	Not available	Not available
CPWM	(21)	(22)	Not available	Not available	(23), (24)	(25)
RPWM	(26)	(27)	Not available	Not available	(28)	(29)

Downloaded from Arrow.com.

23.7.3 Overview of Pin Initialization Procedures and Mode Transitions in Case of Error during Operation

- When making a transition to a mode (Normal, PWM1, PWM2, or PCM) in which the pin output level is selected by the TIOR register setting, initialize the pins by means of the TIOR register setting.
- In PWM mode 1, waveforms are not output to the MTIOCnB and MTIOCnD (n = 3, 4) pins. When a pin is configured for MTIOCnB or MTIOCnD, it enters high-impedance state. To output a specified level, set the pin to general output port.
- In PWM mode 2, waveforms are not output to the cycle register pins. When a pin is configured for MTIOCnm (n = 0 to 2; m = A to D), it enters high-impedance state. To output a specified level, set the pin to general output port.
- In normal mode or PWM 2 mode, if the TGRC and TGRD register operate as buffer registers, waveforms are not output to the corresponding pins (MTIOCnC or MTIOCnD (n = 0, 3, 4)). When a pin is configured for MTIOCnC or MTIOCnD, it enters high-impedance state. To output a specified level, set the pin to general output port.
- In PWM mode 1, if either TGRC or TGRD register operates as a buffer register, waveforms are not output to the corresponding pins (MTIOCnC or MTIOCnD (n = 0, 3, 4)). When a pin is configured for MTIOCnC or MTIOCnD, it enter high-impedance state. To output a specified level, set the pin to general output port.
- When making a transition to a mode (CPWM or RPWM) in which the pin output level is selected by the timer output control register (TOCR) setting, temporarily disable output in MTU3 and MTU4 with the TOER register. At this time, when a pin is configured for MTIOCnm (n = 3, 4; m = A to D), it enters high-impedance state. To output a specified level, set the pin to general output port. Switch to normal mode, perform initialization with the TIOR register, and restore the TIOR register to its initial value. After that, operate the MTU in accordance with the mode setting procedure (TOCR setting, TMDR setting, and TOER setting).

Note: Channel number is substituted for “n” indicated in this section unless otherwise specified.

Pin initialization procedures are described below for the numbered combinations in Table 23.56. The active level is assumed to be low.

Downloaded from Arrow.com.

(1) Operation When Error Occurs in Normal Mode and Operation is Restarted in Normal Mode

Figure 23.116 shows a case in which an error occurs in normal mode and operation is restarted in normal mode after re-setting.

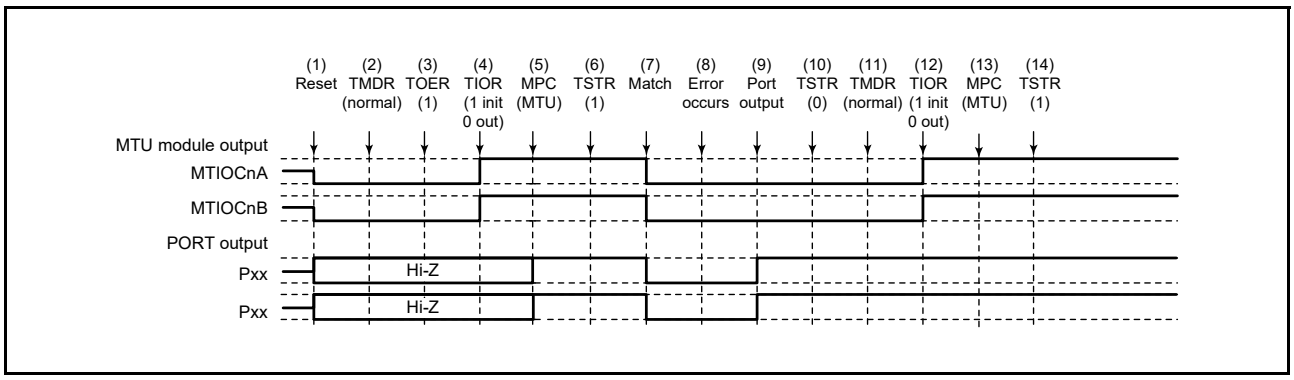


Figure 23.116 Error Occurrence in Normal Mode, Recovery in Normal Mode

(1) After a reset, the MTU output goes low and the ports enter high-impedance state.
(2) After a reset, the TMDR setting is for normal mode.
(3) For MTU3 and MTU4, enable output with the TOER register before initializing the pins with the TIOR register.
(4) Initialize the pins with the TIOR register. (In the example, the initial output is a high level, and a low level is output on compare match occurrence.)
(5) Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
(6) Start count operation by setting the TSTR register.
(7) Output goes low on compare match occurrence.
(8) An error occurs.
(9) Use the port direction register (PDR) and port mode register (PMR) for the input port pin to switch it to operate as a general output port pin, and the port output data register (PODR) to select output of the non-active level.
(10) Stop count operation by setting the TSTR register.
(11) This step is not necessary when restarting in normal mode.
(12) Initialize the pins with the TIOR register.
(13) Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
(14) Restart operation by setting the TSTR register.

Downloaded from Arrow.com.

(2) Operation When Error Occurs in Normal Mode and Operation is Restarted in PWM Mode 1

Figure 23.117 shows a case in which an error occurs in normal mode and operation is restarted in PWM mode 1 after re-setting.

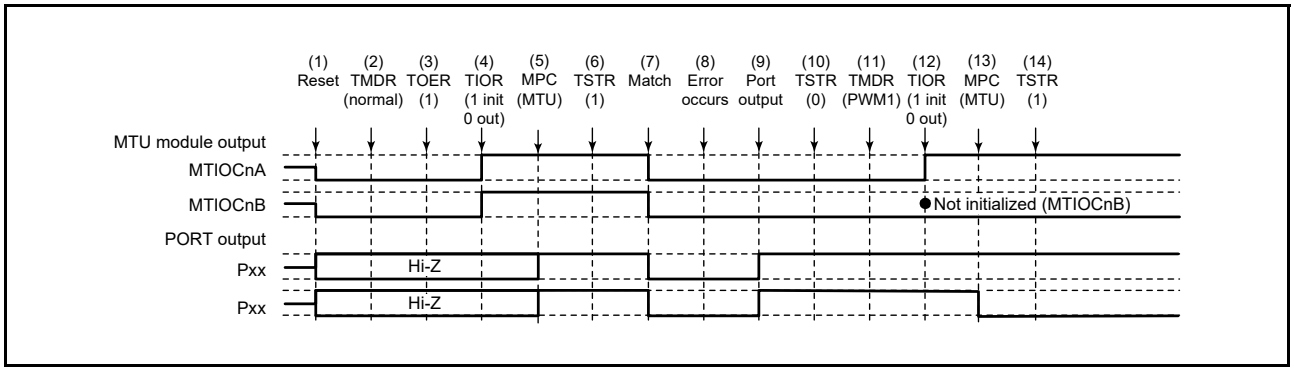


Figure 23.117 Error Occurrence in Normal Mode, Recovery in PWM Mode 1

(1) to (10) are the same as in Figure 23.116.
(11)Set PWM mode 1.
(12)Set the TIOR register to initialize pins, i.e. so that the MTIOCnB (or MTIOCnD) does not produce a waveform in PWM mode 1. If a particular level should be output, set the port direction register (PDR) and the port output data register (PODR) so that the pins of the I/O port operate as general outputs.
(13) Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
(14) Restart operation by setting the TSTR register.

(3) Operation When Error Occurs in Normal Mode and Operation is Restarted in PWM Mode 2

Figure 23.118 shows a case in which an error occurs in normal mode and operation is restarted in PWM mode 2 after re-setting.

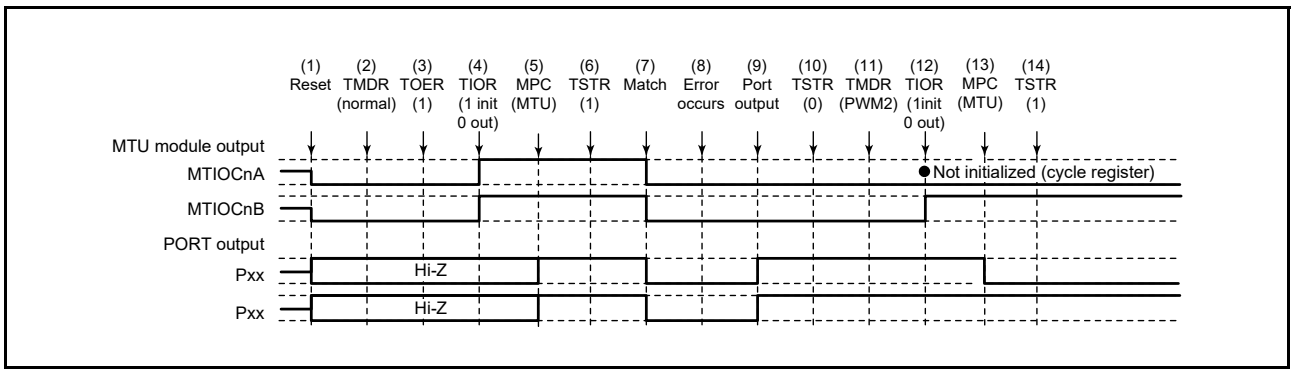


Figure 23.118 Error Occurrence in Normal Mode, Recovery in PWM Mode 2

(1) to (10) are the same as in Figure 23.116.
(11) Set PWM mode 2.
(12) Initialize the pins with the TIOR register. (In PWM mode 2, the cycle register pins are not initialized. If initialization is required, initialize in normal mode, then switch to PWM mode 2.)
(13) Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
(14) Restart operation by setting the TSTR register.

Note: PWM mode 2 can only be selected for MTU0 to MTU2, and therefore the TOER register setting is not necessary.

Downloaded from Arrow.com.

(4) Operation When Error Occurs in Normal Mode and Operation is Restarted in Phase Counting Mode

Figure 23.119 shows a case in which an error occurs in normal mode and operation is restarted in phase counting mode after re-setting.

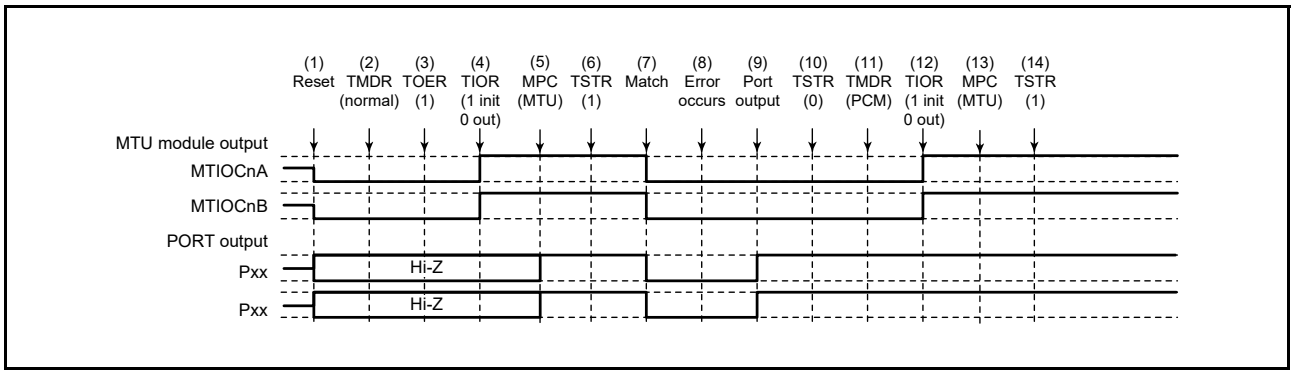


Figure 23.119 Error Occurrence in Normal Mode, Recovery in Phase Counting Mode

(1) to (10) are the same as in Figure 23.116.
(11) Set the phase counting mode.
(12) Initialize the pins with the TIOR register.
(13) Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
(14) Restart operation by setting the TSTR register.

Note: The phase counting mode can only be selected for MTU1 and MTU2, and therefore the TOER register setting is not necessary.

Downloaded from Arrow.com.

(5) Operation When Error Occurs in Normal Mode and Operation is Restarted in Complementary PWM Mode

Figure 23.120 shows a case in which an error occurs in normal mode and operation is restarted in complementary PWM mode after re-setting.

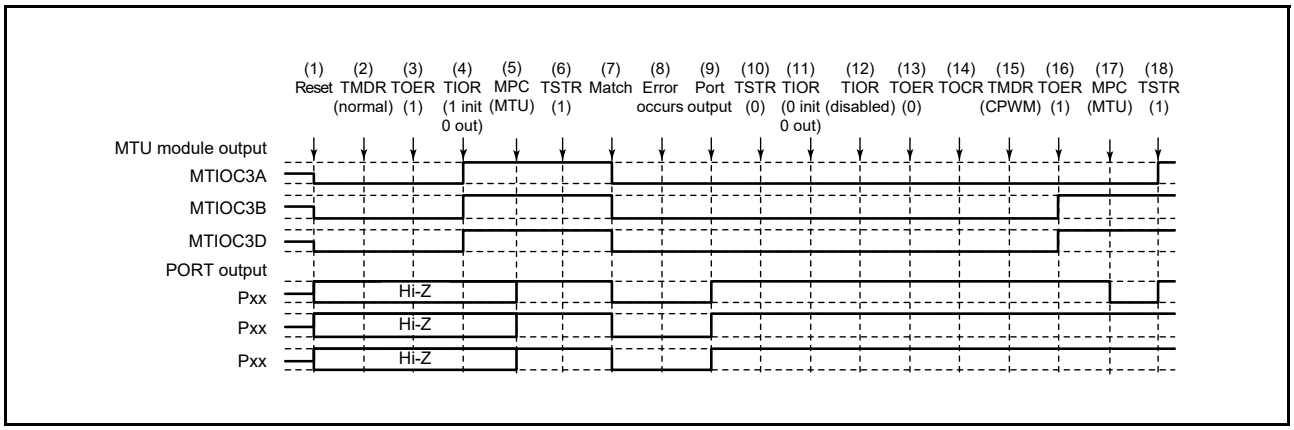


Figure 23.120 Error Occurrence in Normal Mode, Recovery in Complementary PWM Mode

(1) to (10) are the same as in Figure 23.116.
(11) Initialize the normal mode waveform generation block with the TIOR register.
(12) Disable operation of the normal mode waveform generation block with the TIOR register.
(13) Disable output in MTU3 and MTU4 with the TOER register.
(14) Select the complementary PWM mode output level and enable or disable cyclic output with the TOCR register.
(15) Set complementary PWM mode.
(16) Enable output in MTU3 and MTU4 with the TOER register.
(17) Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
(18) Restart operation by setting the TSTR register.

Downloaded from Arrow.com.

(6) Operation When Error Occurs in Normal Mode and Operation is Restarted in Reset-Synchronized PWM Mode

Figure 23.121 shows a case in which an error occurs in normal mode and operation is restarted in reset-synchronized PWM mode after re-setting.

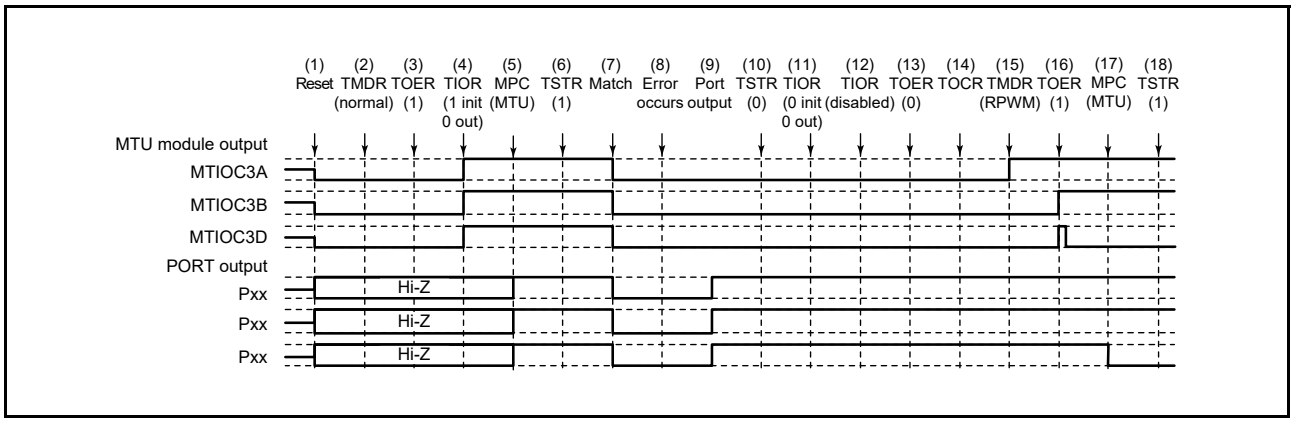


Figure 23.121 Error Occurrence in Normal Mode, Recovery in Reset-Synchronized PWM Mode

(1) to (13) are the same as in Figure 23.120.
(14) Select the reset-synchronized PWM mode output level and enable or disable cyclic output with the TOCR register.
(15) Set reset-synchronized PWM mode.
(16) Enable output in MTU3 and MTU4 with the TOER register.
(17) Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
(18) Restart operation by setting the TSTR register.

Downloaded from Arrow.com.

(7) Operation When Error Occurs in PWM Mode 1 and Operation is Restarted in Normal Mode

Figure 23.122 shows a case in which an error occurs in PWM mode 1 and operation is restarted in normal mode after re-setting.

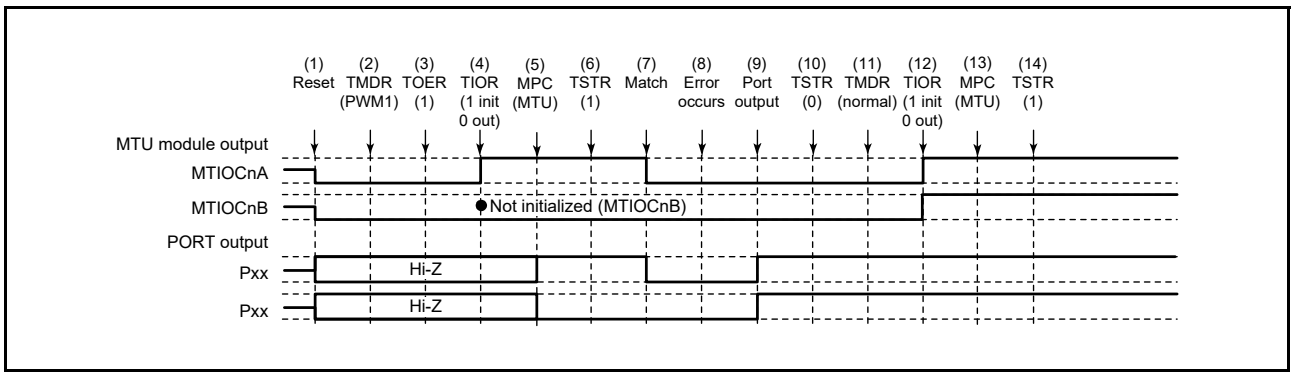


Figure 23.122 Error Occurrence in PWM Mode 1, Recovery in Normal Mode

(1) After a reset, the MTU output goes low and the ports enter high-impedance state.
(2) Set PWM mode 1.
(3) For MTU3 and MTU4, enable output with the TOER register before initializing the pins with the TIOR register.
(4) Initialize the pins with the TIOR register. (In the example, the initial output is a high level, and a low level is output on compare match occurrence. In PWM mode 1, the MTIOCnB side is not initialized.)
(5) Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
(6) Start count operation by setting the TSTR register.
(7) Output goes low on compare match occurrence.
(8) An error occurs.
(9) Use the port direction register (PDR) and port mode register (PMR) for the input port pin to switch it to operate as a general output port pin, and the port output data register (PODR) to select output of the non-active level.
(10) Stop count operation by setting the TSTR register.
(11) Set normal mode.
(12) Initialize the pins with the TIOR register.
(13) Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
(14) Restart operation by setting the TSTR register.

Downloaded from Arrow.com.

(8) Operation When Error Occurs in PWM Mode 1 and Operation is Restarted in PWM mode 1

Figure 23.123 shows a case in which an error occurs in PWM mode 1 and operation is restarted in PWM mode 1 after re-setting.

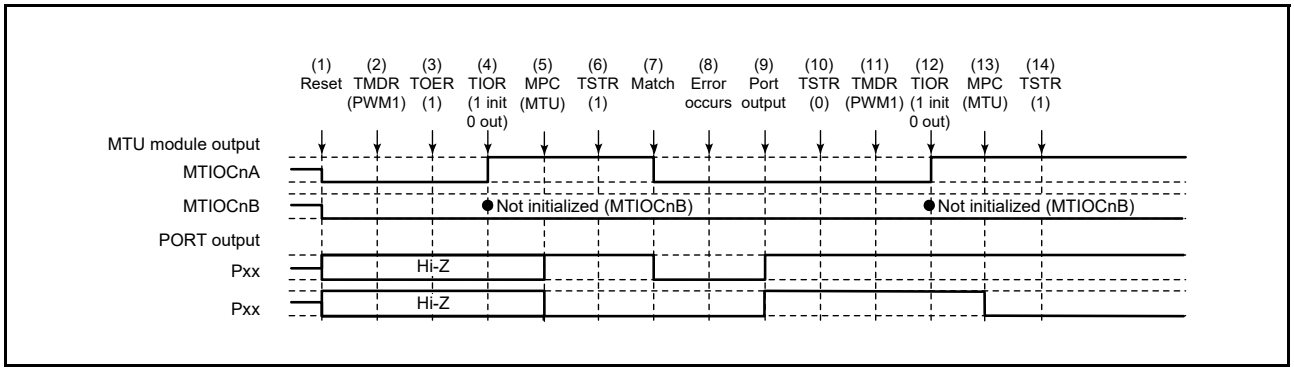


Figure 23.123 Error Occurrence in PWM Mode 1, Recovery in PWM Mode 1

(1) to (10) are the same as in Figure 23.122.
(11) This step is not necessary when restarting in PWM mode 1.
(12) Initialize the pins with the TIOR register. (In PWM mode 1, a waveform is not output on the MTIOCnB (MTIOCnD) pins. If a level is to be output, make the required general output port settings in the I/O port's port direction register (PDR) and port output data register (PODR).)
(13) Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
(14) Restart operation by setting the TSTR register.

(9) Operation When Error Occurs in PWM Mode 1 and Operation is Restarted in PWM mode 2

Figure 23.124 shows a case in which an error occurs in PWM mode 1 and operation is restarted in PWM mode 2 after re-setting.

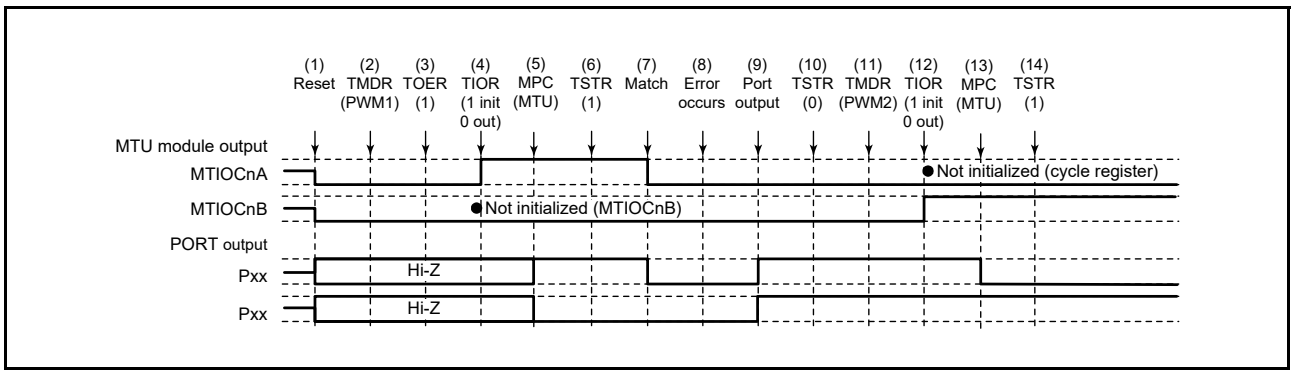


Figure 23.124 Error Occurrence in PWM Mode 1, Recovery in PWM Mode 2

(1) to (10) are the same as in Figure 23.122.
(11) Set PWM mode 2.
(12) Initialize the pins with the TIOR register. (In PWM mode 2, a waveform is not output on the cycle register pins. If a level is to be output, make the required general output port settings in the I/O port's port direction register (PDR) and port output data register (PODR).)
(13) Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
(14) Restart operation by setting the TSTR register.

Note: PWM mode 2 can only be selected for MTU0 to MTU2, and therefore the TOER register setting is not necessary.

Downloaded from Arrow.com.

(10) Operation When Error Occurs in PWM Mode 1 and Operation is Restarted in Phase Counting Mode

Figure 23.125 shows a case in which an error occurs in PWM mode 1 and operation is restarted in phase counting mode after re-setting.

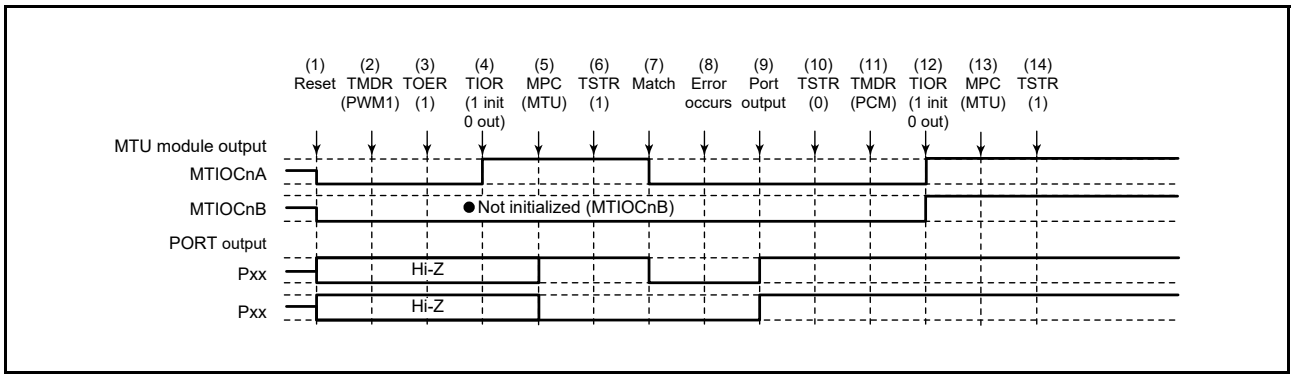


Figure 23.125 Error Occurrence in PWM Mode 1, Recovery in Phase Counting Mode

(1) to (10) are the same as in Figure 23.122.
(11) Set the phase counting mode.
(12) Initialize the pins with the TIOR register.
(13) Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
(14)Restart operation by setting the TSTR register.

Note: The phase counting mode can only be selected for MTU1 and MTU2, and therefore the TOER register setting is not necessary.

Downloaded from Arrow.com.

(11) Operation When Error Occurs in PWM Mode 1 and Operation is Restarted in Complementary PWM Mode

Figure 23.126 shows a case in which an error occurs in PWM mode 1 and operation is restarted in complementary PWM mode after re-setting.

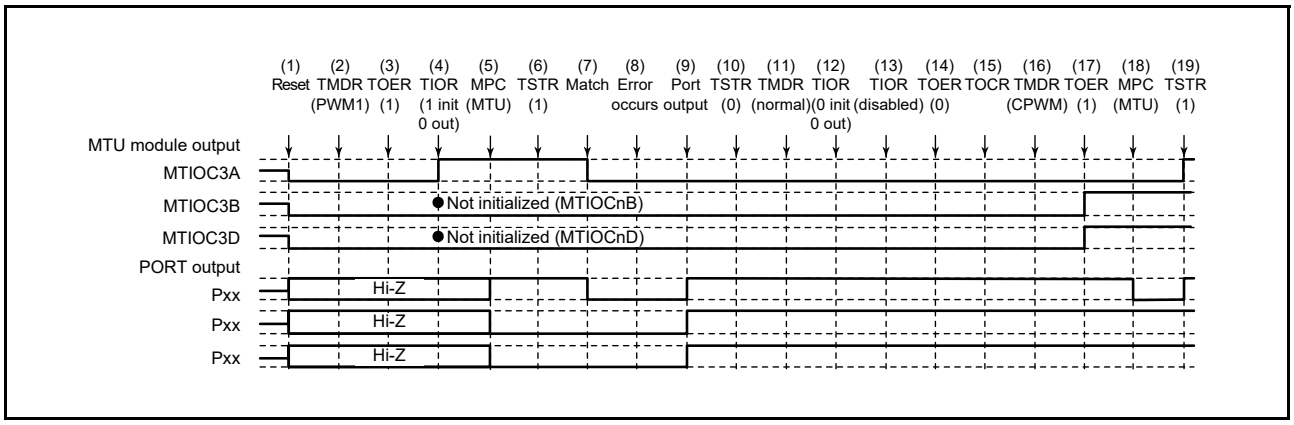


Figure 23.126 Error Occurrence in PWM Mode 1, Recovery in Complementary PWM Mode

(1) to (10) are the same as in Figure 23.122.
(11) Set normal mode to initialize the normal mode waveform generation block.
(12) Initialize the PWM mode 1 waveform generation block with the TIOR register.
(13) Disable operation of the PWM mode 1 waveform generation block with the TIOR register.
(14) Disable output in MTU3 and MTU4 with the TOER register.
(15) Select the complementary PWM mode output level and enable or disable cyclic output with the TOCR register.
(16) Set complementary PWM mode.
(17) Enable output in MTU3 and MTU4 with the TOER register.
(18) Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
(19) Restart operation by setting the TSTR register.

Downloaded from Arrow.com.

(12) Operation When Error Occurs in PWM Mode 1 and Operation is Restarted in Reset-Synchronized PWM Mode

Figure 23.127 shows a case in which an error occurs in PWM mode 1 and operation is restarted in reset-synchronized PWM mode after re-setting.

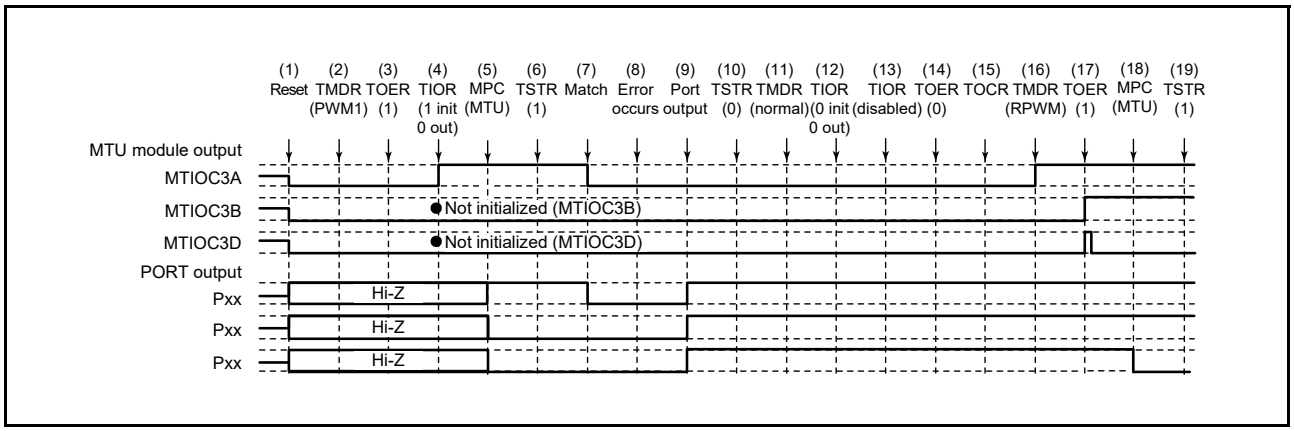


Figure 23.127 Error Occurrence in PWM Mode 1, Recovery in Reset-Synchronized PWM Mode

(1) to (14) are the same as in Figure 23.126.
(15) Select the reset-synchronized PWM mode output level and enable or disable cyclic output with the TOCR register.
(16) Set reset-synchronized PWM mode.
(17) Enable output in MTU3 and MTU4 with the TOER register.
(18) Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
(19) Restart operation by setting the TSTR register.

Downloaded from Arrow.com.

(13) Operation When Error Occurs in PWM Mode 2 and Operation is Restarted in Normal Mode

Figure 23.128 shows a case in which an error occurs in PWM mode 2 and operation is restarted in normal mode after re-setting.

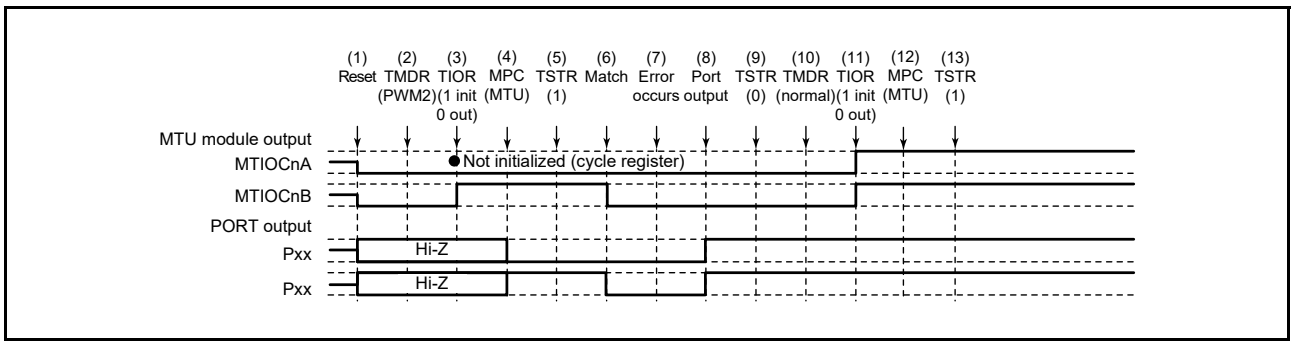


Figure 23.128 Error Occurrence in PWM Mode 2, Recovery in Normal Mode

(1) After a reset, the MTU output goes low and the ports enter high-impedance state.
(2) Set PWM mode 2.
(3) Initialize the pins with the TIOR register. (In the example, the initial output is a high level, and a low level is output on compare match occurrence. In PWM mode 2, the cycle register pins are not initialized. In the example, MTIOCnA is the cycle register.)
(4) Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
(5) Start count operation by setting the TSTR register.
(6) Output goes low on compare match occurrence.
(7) An error occurs.
(8) Use the port direction register (PDR) and port mode register (PMR) for the input port pin to switch it to operate as a general output port pin, and the port output data register (PODR) to select output of the non-active level.
(9) Stop count operation by setting the TSTR register.
(10) Set normal mode.
(11) Initialize the pins with the TIOR register.
(12) Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
(13) Restart operation by setting the TSTR register.

Downloaded from Arrow.com.

(14) Operation When Error Occurs in PWM Mode 2 and Operation is Restarted in PWM Mode 1

Figure 23.129 shows a case in which an error occurs in PWM mode 2 and operation is restarted in PWM mode 1 after re-setting.

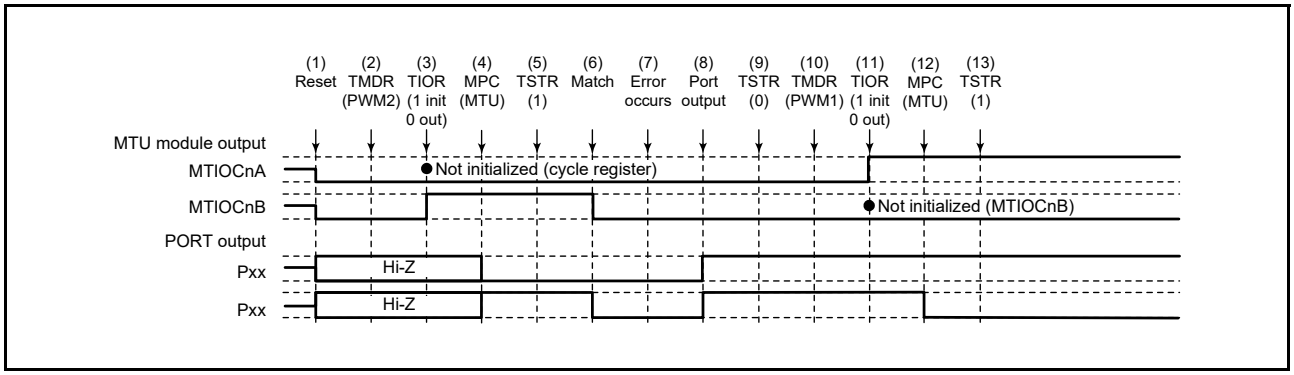


Figure 23.129 Error Occurrence in PWM Mode 2, Recovery in PWM Mode 1

(1) to (9) are the same as in Figure 23.128.
(10) Set PWM mode 1.
(11) Initialize the pins with the TIOR register. (In PWM mode 1, a waveform is not output on the MTIOCnB (MTIOCnD) pins. If a level is to be output, make the required general output port settings in the I/O port's port direction register (PDR) and port output data register (PODR).)
(12) Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
(13) Restart operation by setting the TSTR register.

(15) Operation When Error Occurs in PWM Mode 2 and Operation is Restarted in PWM Mode 2

Figure 23.130 shows a case in which an error occurs in PWM mode 2 and operation is restarted in PWM mode 2 after re-setting.

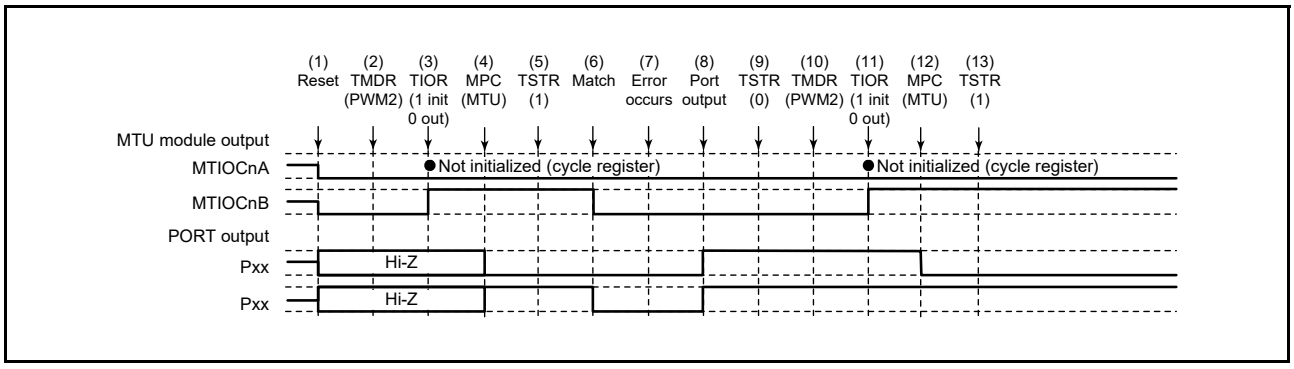


Figure 23.130 Error Occurrence in PWM Mode 2, Recovery in PWM Mode 2

(1) to (9) are the same as in Figure 23.128.
(10) This step is not necessary when restarting in PWM mode 2.
(11) Initialize the pins with the TIOR register. (In PWM mode 2, a waveform is not output on the cycle register pins. If a level is to be output, make the required general output port settings in the I/O port's port direction register (PDR) and port output data register (PODR).)
(12) Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
(13) Restart operation by setting the TSTR register.

Downloaded from Arrow.com.

(16) Operation When Error Occurs in PWM Mode 2 and Operation is Restarted in Phase Counting Mode

Figure 23.131 shows a case in which an error occurs in PWM mode 2 and operation is restarted in phase counting mode after re-setting.

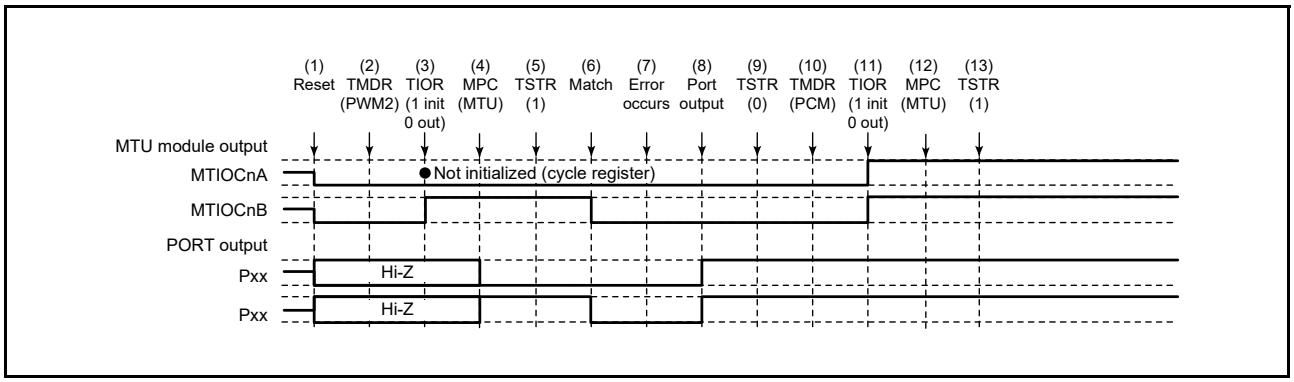


Figure 23.131 Error Occurrence in PWM Mode 2, Recovery in Phase Counting Mode

(1) to (9) are the same as in Figure 23.128.
(10) Set the phase counting mode.
(11) Initialize the pins with the TIOR register.
(12) Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
(13) Restart operation by setting the TSTR register.

Downloaded from Arrow.com.

(17) Operation When Error Occurs in Phase Counting Mode and Operation is Restarted in Normal Mode

Figure 23.132 shows a case in which an error occurs in phase counting mode and operation is restarted in normal mode after re-setting.

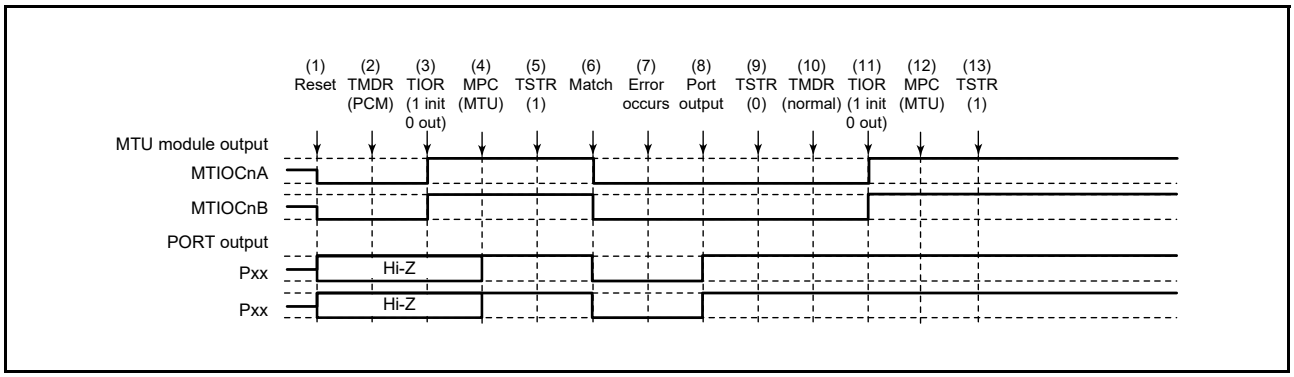


Figure 23.132 Error Occurrence in Phase Counting Mode, Recovery in Normal Mode

(1) After a reset, the MTU output goes low and the ports enter high-impedance state.
(2) Set phase counting mode.
(3) Initialize the pins with the TIOR register. (In the example, the initial output is a high level, and a low level is output on compare match occurrence.)
(4) Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
(5) Start count operation by setting the TSTR register.
(6) Output goes low on compare match occurrence.
(7) An error occurs.
(8) Use the port direction register (PDR) and port mode register (PMR) for the input port pin to switch it to operate as a general output port pin, and the port output data register (PODR) to select output of the non-active level.
(9) Stop count operation by setting the TSTR register.
(10) Set normal mode.
(11) Initialize the pins with the TIOR register.
(12) Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
(13) Restart operation by setting the TSTR register.

Downloaded from Arrow.com.

(18) Operation When Error Occurs in Phase Counting Mode and Operation is Restarted in PWM Mode 1

Figure 23.133 shows a case in which an error occurs in phase counting mode and operation is restarted in PWM mode 1 after re-setting.

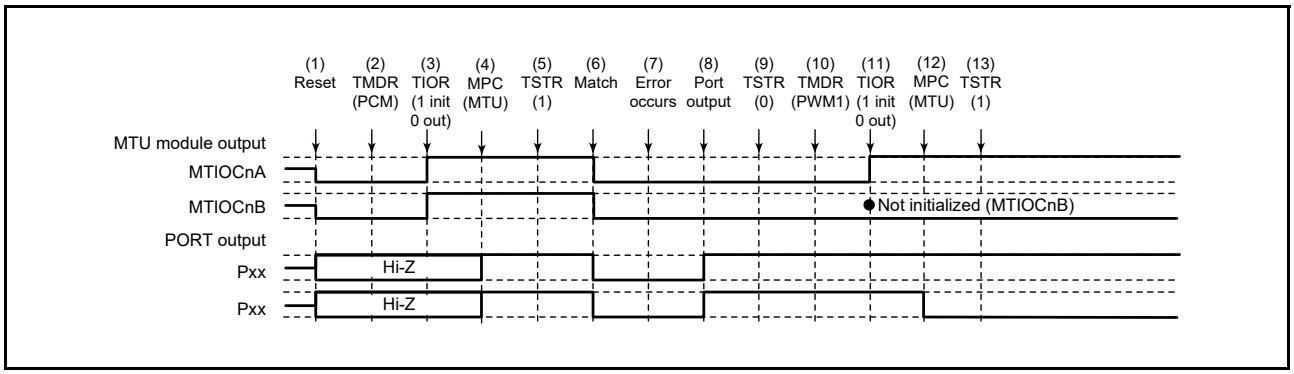


Figure 23.133 Error Occurrence in Phase Counting Mode, Recovery in PWM Mode 1

(1) to (9) are the same as in Figure 23.132.
(10) Set PWM mode 1.
(11) Initialize the pins with the TIOR register. (In PWM mode 1, a waveform is not output on the MTIOCnB (MTIOCnD) pins. If a level is to be output, make the required general output port settings in the I/O port's port direction register (PDR) and port output data register (PODR).)
(12) Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
(13) Restart operation by setting the TSTR register.

Downloaded from Arrow.com.

(19) Operation When Error Occurs in Phase Counting Mode and Operation is Restarted in PWM Mode 2

Figure 23.134 shows a case in which an error occurs in phase counting mode and operation is restarted in PWM mode 2 after re-setting.

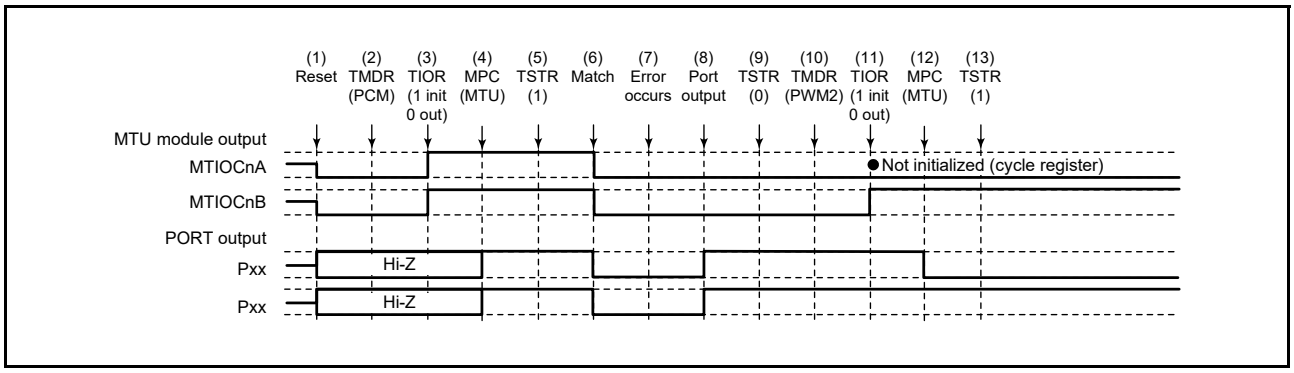


Figure 23.134 Error Occurrence in Phase Counting Mode, Recovery in PWM Mode 2

(1) to (9) are the same as in Figure 23.132.
(10) Set PWM mode 2.
(11) Initialize the pins with the TIOR register. (In PWM mode 1, a waveform is not output on the MTIOCnB (MTIOCnD) pins. If a level is to be output, make the required general output port settings in the I/O port's port direction register (PDR) and port output data register (PODR).)
(12) Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
(13) Restart operation by setting the TSTR register.

(20) Operation When Error Occurs in Phase Counting Mode and Operation is Restarted in Phase Counting Mode

Figure 23.135 shows a case in which an error occurs in phase counting mode and operation is restarted in phase counting mode after re-setting.

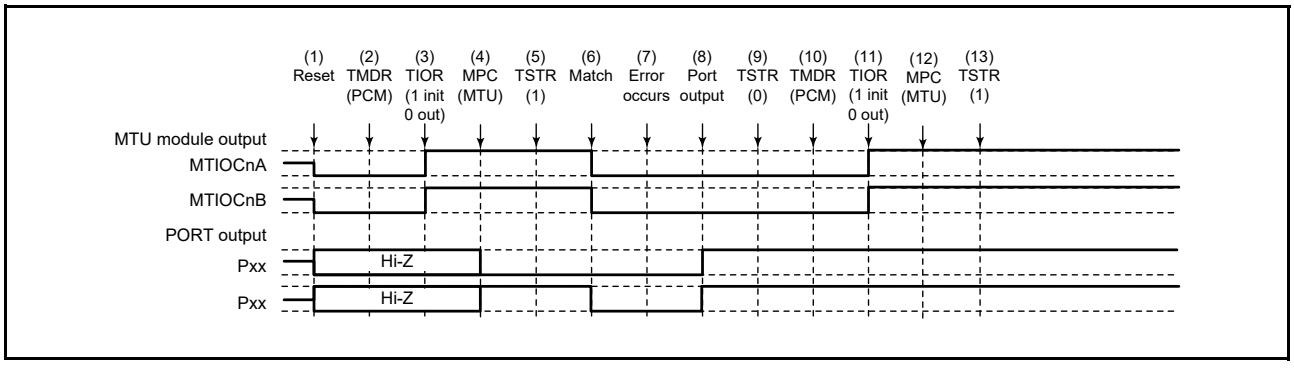


Figure 23.135 Error Occurrence in Phase Counting Mode, Recovery in Phase Counting Mode

(1) to (9) are the same as in Figure 23.132.
(10) This step is not necessary when restarting in phase counting mode.
(11) Initialize the pins with the TIOR register.
(12) Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
(13) Restart operation by setting the TSTR register.

Downloaded from Arrow.com.

(21) Operation When Error Occurs in Complementary PWM Mode and Operation is Restarted in Normal Mode

Figure 23.136 shows a case in which an error occurs in complementary PWM mode and operation is restarted in normal mode after re-setting.

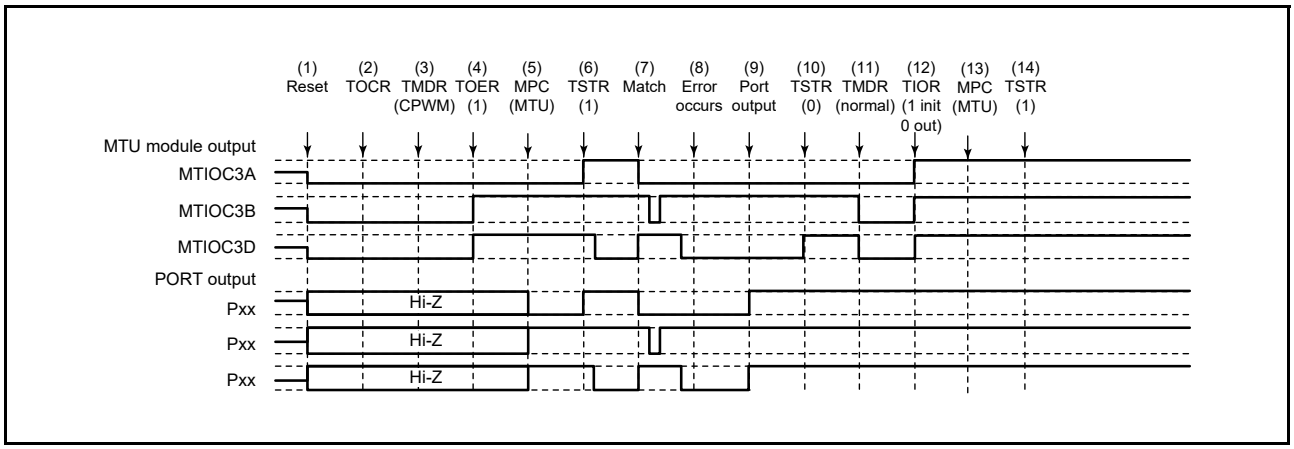


Figure 23.136 Error Occurrence in Complementary PWM Mode, Recovery in Normal Mode

(1) After a reset, the MTU output goes low and the ports enter high-impedance state.
(2) Select the complementary PWM mode output level and enable or disable cyclic output with the TOCR register.
(3) Set complementary PWM mode.
(4) Enable output in MTU3 and MTU4 with the TOER register.
(5) Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
(6) Start count operation by setting the TSTR register.
(7) The complementary PWM waveform is output on compare match occurrence.
(8) An error occurs.
(9) Use the port direction register (PDR) and port mode register (PMR) for the input port pin to switch it to operate as a general output port pin, and the port output data register (PODR) to select output of the non-active level.
(10) Stop count operation by setting the TSTR register. (MTU output becomes the initial complementary PWM output value).
(11) Set normal mode (MTU output goes low).
(12) Initialize the pins with the TIOR register.
(13) Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
(14) Restart operation by setting the TSTR register.

Downloaded from Arrow.com.

(22) Operation When Error Occurs in Complementary PWM Mode and Operation is Restarted in PWM Mode 1

Figure 23.137 shows a case in which an error occurs in complementary PWM mode and operation is restarted in PWM mode 1 after re-setting.

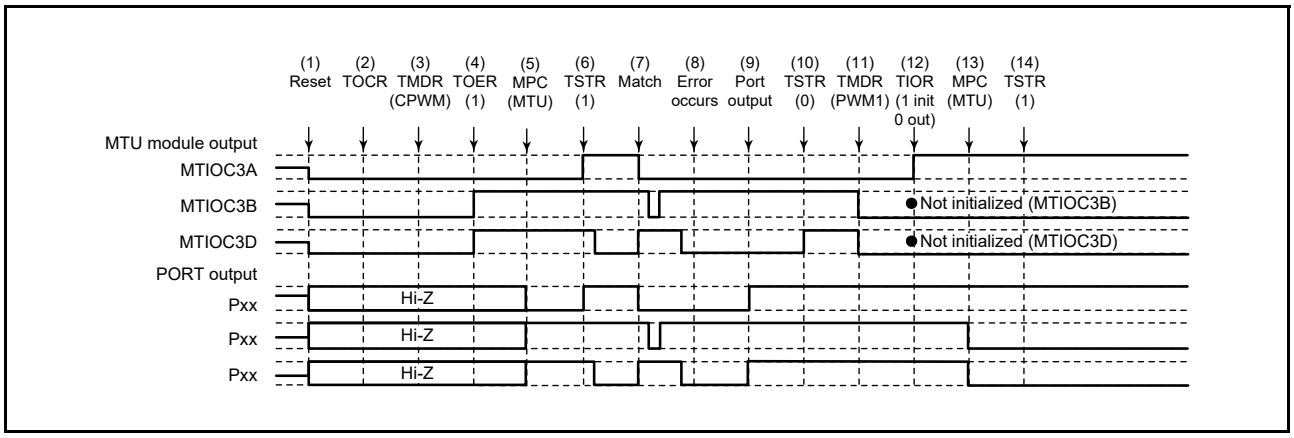


Figure 23.137 Error Occurrence in Complementary PWM Mode, Recovery in PWM Mode 1

(1) to (10) are the same as in Figure 23.136.
(11) Set PWM mode 1 (MTU output goes low).
(12) Initialize the pins with the TIOR register. (In PWM mode 1, a waveform is not output on the MTIOCnB (MTIOCnD) pins. If a level is to be output, make the required general output port settings in the I/O port's port direction register (PDR) and port output data register (PODR).)
(13) Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
(14) Restart operation by setting the TSTR register.

Downloaded from Arrow.com.

(23) Operation When Error Occurs in Complementary PWM Mode and Operation is Restarted in Complementary PWM Mode

Figure 23.138 shows a case in which an error occurs in complementary PWM mode and operation is restarted in complementary PWM mode after re-setting (when operation is restarted using the cycle and duty settings at the time of stopping the counter).

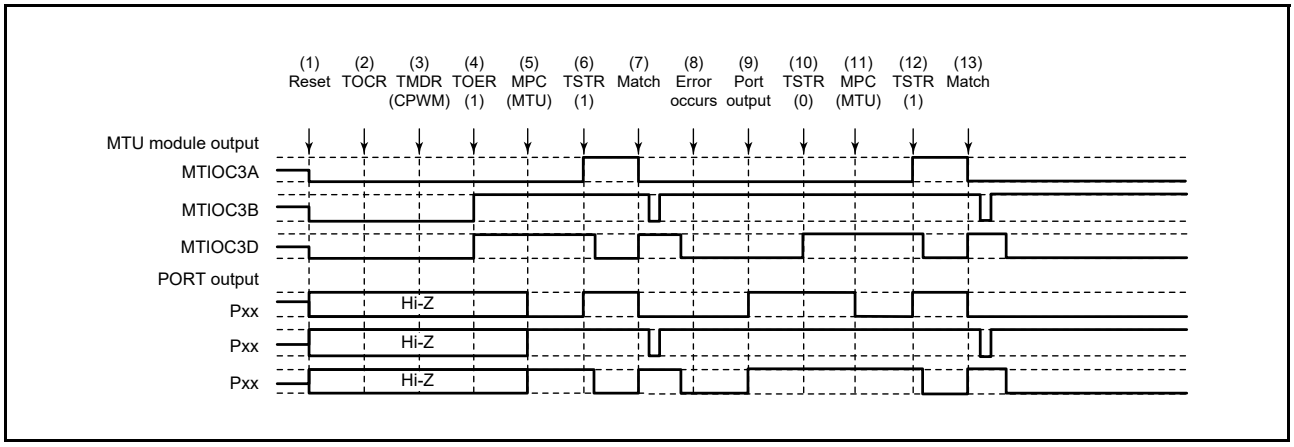


Figure 23.138 Error Occurrence in Complementary PWM Mode, Recovery in Complementary PWM Mode

(1) to (10) are the same as in Figure 23.136.
(11) Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
(12) Restart operation by setting the TSTR register.
(13) The complementary PWM waveform is output on compare match occurrence.

Downloaded from Arrow.com.

(24) Operation When Error Occurs in Complementary PWM Mode and Operation is Restarted in Complementary PWM Mode with New Settings

Figure 23.139 shows a case in which an error occurs in complementary PWM mode and operation is restarted in complementary PWM mode after re-setting (operation is restarted using new cycle and duty settings).

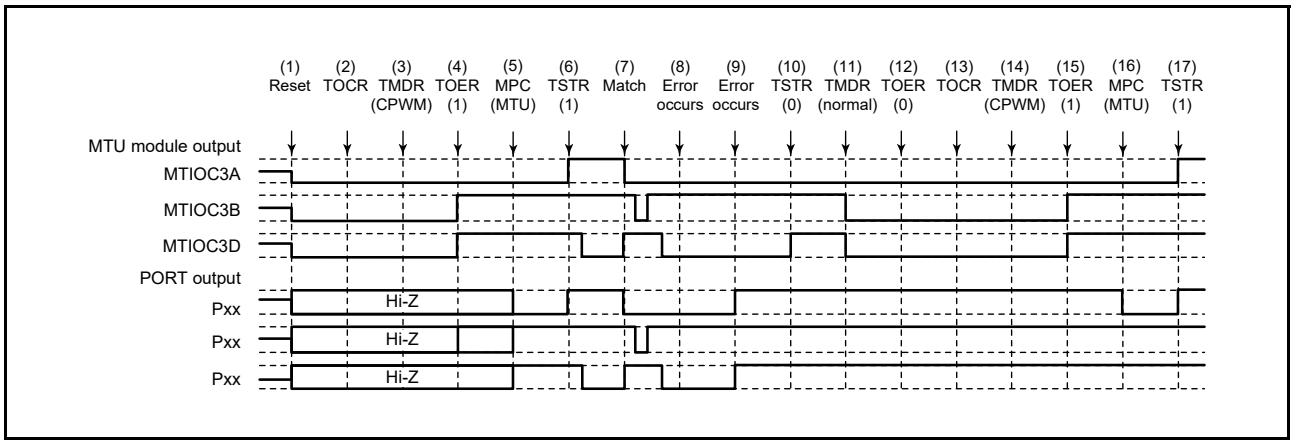


Figure 23.139 Error Occurrence in Complementary PWM Mode, Recovery in Complementary PWM Mode

(1) to (10) are the same as in Figure 23.136.
(11) Set normal mode and make new settings (MTU output goes low).
(12) Disable output in MTU3 and MTU4 with the TOER register.
(13) Select the complementary PWM mode output level and enable or disable cyclic output with the TOCR register.
(14) Set complementary PWM mode.
(15) Enable output in MTU3 and MTU4 with the TOER register.
(16) Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
(17) Restart operation by setting the TSTR register.

Downloaded from Arrow.com.

(25) Operation When Error Occurs in Complementary PWM Mode and Operation is Restarted in Reset-Synchronized PWM Mode

Figure 23.140 shows a case in which an error occurs in complementary PWM mode and operation is restarted in reset-synchronized PWM mode after re-setting.

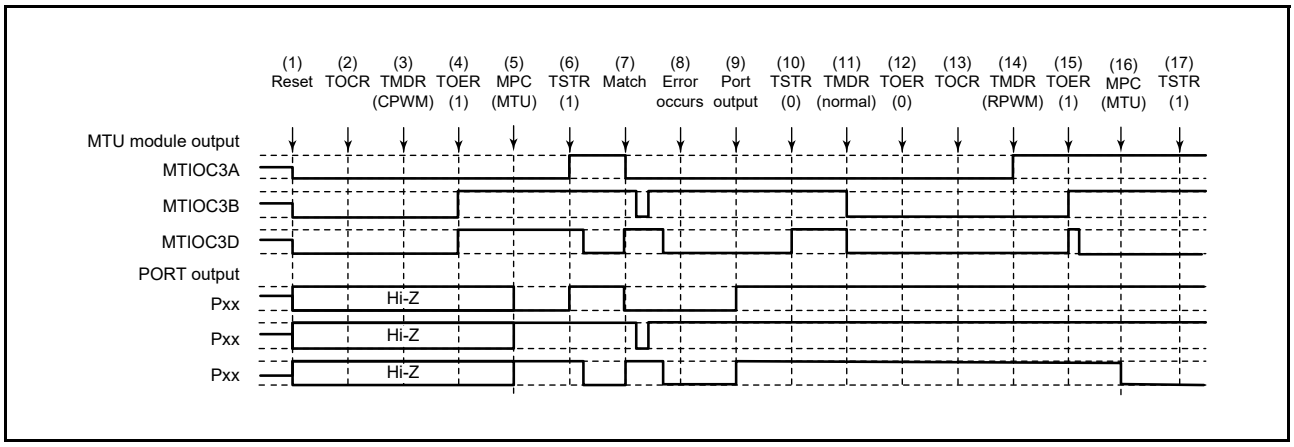


Figure 23.140 Error Occurrence in Complementary PWM Mode, Recovery in Reset-Synchronized PWM Mode

(1) to (10) are the same as in Figure 23.136.
(11) Set normal mode (MTU output goes low).
(12) Disable output in MTU3 and MTU4 with the TOER register.
(13) Select the reset-synchronized PWM mode output level and enable or disable cyclic output with the TOCR register.
(14) Set reset-synchronized PWM mode.
(15) Enable output in MTU3 and MTU4 with the TOER register.
(16) Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
(17) Restart operation by setting the TSTR register.

Downloaded from Arrow.com.

(26) Operation When Error Occurs in Reset-Synchronized PWM Mode and Operation is Restarted in Normal Mode

Figure 23.141 shows a case in which an error occurs in reset-synchronized PWM mode and operation is restarted in normal mode after re-setting.

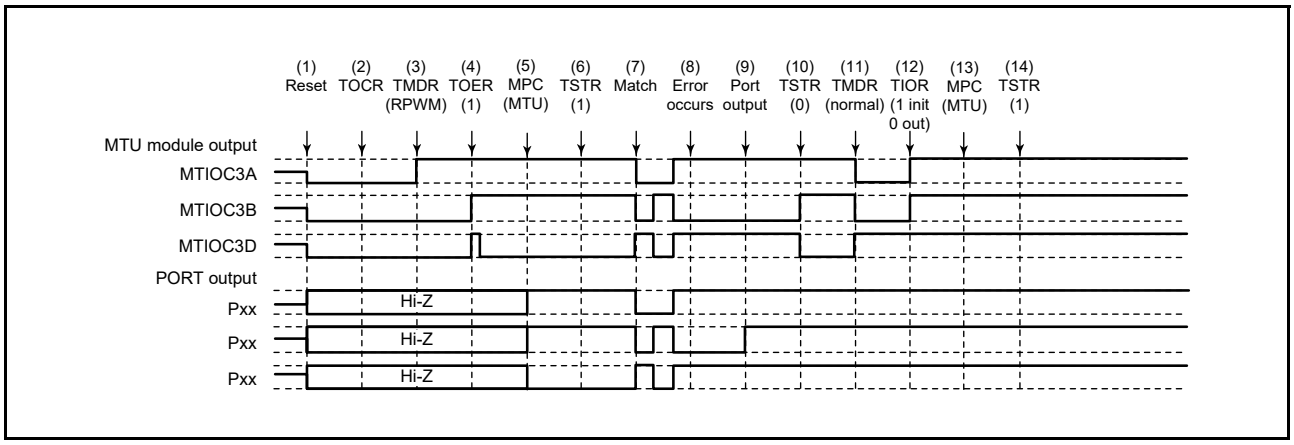


Figure 23.141 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Normal Mode

(1) After a reset, the MTU output goes low and the ports enter high-impedance state.
(2) Select the reset-synchronized PWM mode output level and enable or disable cyclic output with the TOCR register.
(3) Set reset-synchronized PWM mode.
(4) Enable output in MTU3 and MTU4 with the TOER register.
(5) Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
(6) Start count operation by setting the TSTR register.
(7) The reset-synchronized PWM waveform is output on compare match occurrence.
(8) An error occurs.
(9) Use the port direction register (PDR) and port mode register (PMR) for the input port pin to switch it to operate as a general output port pin, and the port output data register (PODR) to select output of the non-active level.
(10) Stop count operation by setting the TSTR register. (MTU output becomes the initial reset-synchronized PWM output value.)
(11) Set normal mode (positive-phase MTU output goes low, and negative-phase output goes high).
(12) Initialize the pins with the TIOR register.
(13) Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
(14) Restart operation by setting the TSTR register.

Downloaded from Arrow.com.

(27) Operation When Error Occurs in Reset-Synchronized PWM Mode and Operation is Restarted in PWM Mode 1

Figure 23.142 shows a case in which an error occurs in reset-synchronized PWM mode and operation is restarted in PWM mode 1 after re-setting.

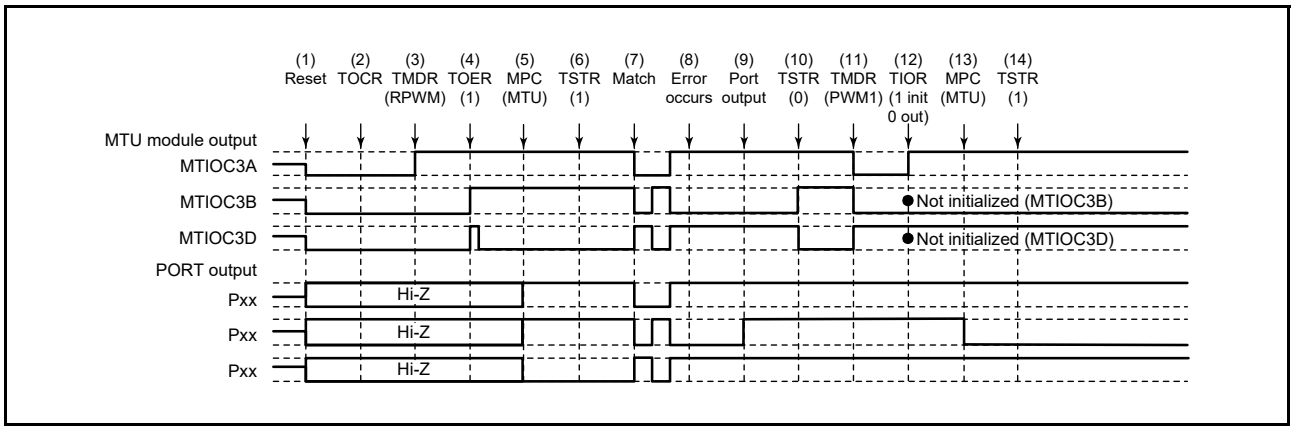


Figure 23.142 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in PWM Mode 1

(1) to (10) are the same as in Figure 23.141.
(11) Set PWM mode 1 (positive-phase MTU output goes low, and negative-phase output goes high).
(12) Initialize the pins with the TIOR register. (In PWM mode 1, a waveform is not output on the MTIOCnB (MTIOCnD) pins. If a level is to be output, make the required general output port settings in the I/O port's port direction register (PDR) and port output data register (PODR).)
(13) Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
(14) Restart operation by setting the TSTR register.

Downloaded from Arrow.com.

(28) Operation When Error Occurs in Reset-Synchronized PWM Mode and Operation is Restarted in Complementary PWM Mode

Figure 23.143 shows a case in which an error occurs in reset-synchronized PWM mode and operation is restarted in complementary PWM mode after re-setting.

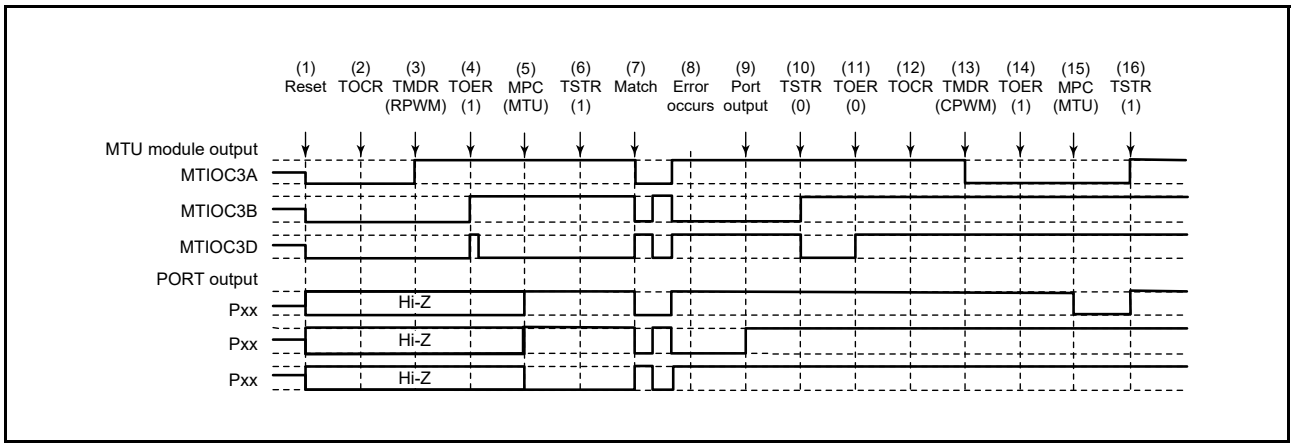


Figure 23.143 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Complementary PWM Mode

(1) to (10) are the same as in Figure 23.141.
(11) Disable output in MTU3 and MTU4 with the TOER register.
(12) Select the complementary PWM mode output level and enable or disable cyclic output with the TOCR register.
(13) Set complementary PWM mode (MTU cyclic output pin goes low).
(14) Enable output in MTU3 and MTU4 with the TOER register.
(15) Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
(16) Restart operation by setting the TSTR register.

Downloaded from Arrow.com.

(29) Operation When Error Occurs in Reset-Synchronized PWM Mode and Operation is Restarted in Reset-Synchronized PWM Mode

Figure 23.144 shows a case in which an error occurs in reset-synchronized PWM mode and operation is restarted in reset-synchronized PWM mode after re-setting.

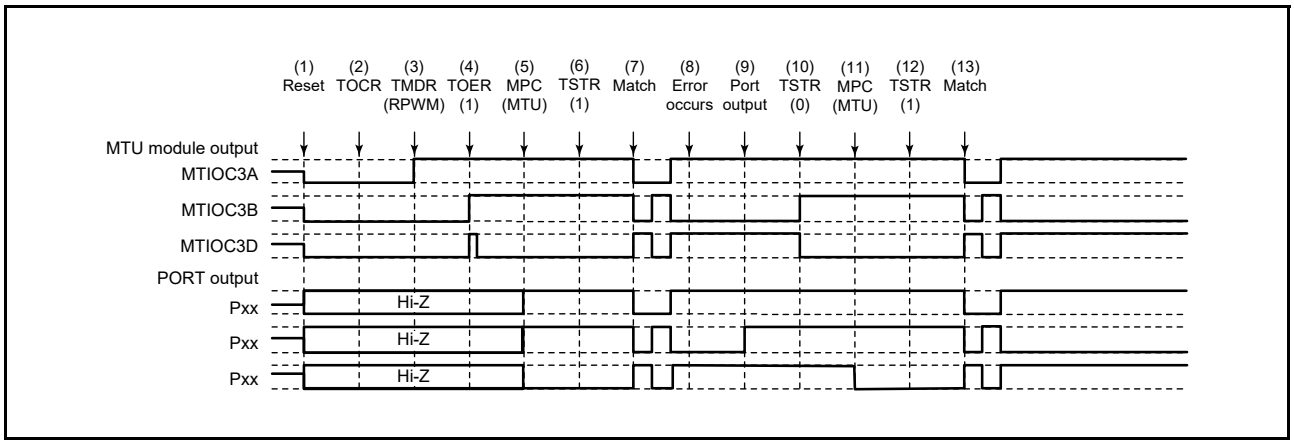


Figure 23.144 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Reset-Synchronized PWM Mode

(1) to (10) are the same as in Figure 23.141.
(11) Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
(12) Restart operation by setting the TSTR register.
(13) The reset-synchronized PWM waveform is output on compare match occurrence.

Downloaded from Arrow.com.

23.8 Operations Linked by the ELC

23.8.1 Event Signal Output to the ELC

The MTU is capable of operation linked with another module set in advance when its interrupt request signal is used as an event signal by the event link controller (ELC).
The MTU outputs the event signal regardless of the setting of the corresponding interrupt request enable bit.

23.8.2 MTU Operations in Response to Receiving Event Signals from the ELC

The MTU can perform the following operations in response to the event set in advance in the ELSRn register of the event link controller (ELC).

(1) Count Start Operation

The MTU is selected the count start operation when using the ELOPA and ELOPB registers setting of the ELC. The ELOPA register functions to MTU1 to MTU3, and ELOPB register functions to MTU4. The TMDR register of the channel set by MTU should be set to the value after reset, 00h. When the specified event is generated by the ELSRn register, the TSTR.CSTn bit shown in Table 23.57 is set to 1, then the MTU counter is started.
However, when the specified event is generated while the TSTR.CSTn bit is set to 1, the event is disabled. Table 23.57 lists the TSTR register bits used for each channel.
For details on the count start operation setting, refer to section 23.3.1, (1) Counter Operation.

Table 23.57 Linkage Operating TSTR Register by the ELC

Channel No.	TSTR Register
MTU1	TSTR.CST1 bit
MTU2	TSTR.CST2 bit
MTU3	TSTR.CST3 bit
MTU4	TSTR.CST4 bit

(2) Input Capture Operation

The MTU is selected the input capture operation when using the ELOPA and ELOPB registers setting of the ELC. The ELOPA register handles MTU1 to MTU3, and ELOPB register handles MTU4. The TMDR register of the channel set by MTU should be set to the value after reset, 00h. When the specified event is generated by the ELSRn register, then the TCNT counter value capture to TGR register. When using the input capture operation, after setting the bit of the TIOR register to the input capture, the TSTR.CSTn bit should be set to 1, and start the counter.
Then, the TIOCnA pin (input capture pin) input is disabled.
Table 23.58 lists the timer general register and timer I/O control register used in the input capture operation by the ELC.
For details on the input capture setting, refer to section 23.3.1, (3) Input Capture Function.

Table 23.58 Timer General Register and Timer I/O Control Register Used in the Input Capture Operation by the ELC

Channel No.	Register Name	Bit Name of TIOR Register
MTU1	MTU1.TGRA register	MTU1.TIOR.IOA[3:0] bits
MTU2	MTU2.TGRA register	MTU2.TIOR.IOA[3:0] bits
MTU3	MTU3.TGRA register	MTU3.TIORH.IOA[3:0] bits
MTU4	MTU4.TGRA register	MTU4.TIORH.IOA[3:0] bits

Downloaded from Arrow.com.

(3) Counter Restart Operation

The MTU is selected the count start operation when using the ELOPA and ELOPB registers setting of the ELC. The ELOPA register functions MTU1 to MTU3, and ELOPB register functions MTU4. The TMDR register of the channel set by MTU should be set to the value after reset, 00h. When the specified event is generated by the ELSRn register, then the TCNT counter value is rewritten to initial value. When the TSTR.CSTn bit is 1, count operation can be continued. For details on the TSTR.CSTn bit, refer to Table 23.57.

23.8.3 Notes on MTU by Event Signal Reception from the ELC

The following describes usage notes when using MTU by the event link operation.

(1) Count Start Operation

When the specified event is generated by the ELSRn register while write cycle is performed to the TSTR.CSTn bit, the write cycle is not performed to the TSTR.CSTn bit, and the setting to 1 takes precedence by generated event.

(2) Count Restart Operation

When the specified event is generated by the ELSRn register while write cycle is performed to the TCNT counter, the write cycle is not performed to the TCNT counter, and count value initialization takes precedence by generated event.

Downloaded from Arrow.com.

24. Port Output Enable 2 (POE2a)

The port output enable 2 (POE) module can be used to place the states of the pins for complementary PWM output by the MTU (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4B, MTIOC4C, and MTIOC4D), and the states of pins for MTU0 (MTIOC0A, MTIOC0B, and MTIOC0C) in the high-impedance in response to changes in the input levels on the POE0#, POE1#, POE3# and POE8# pins, in the output levels on pins for complementary PWM output by the MTU, oscillation stop detection by the clock generation circuit, and changes to register settings (SPOER) or event signal input from the event link controller (ELC).
It can also generate simultaneous interrupt requests.
In this section, “PCLK” is used to refer to PCLKB.

24.1 Overview

Table 24.1 lists the specifications of the POE, and Figure 24.1 shows a block diagram of the POE.

Table 24.1 POE Specifications

Item	Description
High-impedance is controlled by the input level detection	• Falling-edge detection or sampling of the low level 16 times at PCLK/8, PCLK/16, or PCLK/128 clock cycles can be set for each of the POE0#, POE1#, POE3# and POE8# input pins. • Pins for complementary PWM output from the MTU can be placed in the high-impedance on detection of falling edges or sampling of the low level on the POE0# to POE3# pins. • Pins for output from MTU0 can be placed in the high-impedance on detection of falling edges or sampling of the low level on the POE8# pin.
High-impedance is controlled by the output level comparison	• Levels output on pins for complementary PWM output from the MTU are compared, and when simultaneous output of the active level continues for one or more clock cycles, the pins can be placed in the high-impedance.
High-impedance is controlled by the oscillation stop detection	• Pins for complementary PWM output from the MTU and output pins for MTU0 can be placed in the high-impedance when oscillation by the clock generation circuit stops.
High-impedance is controlled by software (registers)	• Pins for complementary PWM output from the MTU and output pins for MTU0 can be placed in the high-impedance by modifying settings of POE registers.
High-impedance is controlled by the event signal	• Pins for complementary PWM output from the MTU and output pins for MTU0 can be placed in the high-impedance in response to an event signal from the event link controller (ELC).
Interrupts	• Interrupts can be generated in response to the results of POE0#, POE1#, POE3# and POE8# input-level detection and MTU complementary PWM output-level comparison.

The POE has input-level detection circuits, output-level comparison circuits, an input for the oscillation stop detection signal from the clock generation circuit, and a high-impedance request/interrupt request generating circuit as shown in Figure 24.1.

Downloaded from Arrow.com.

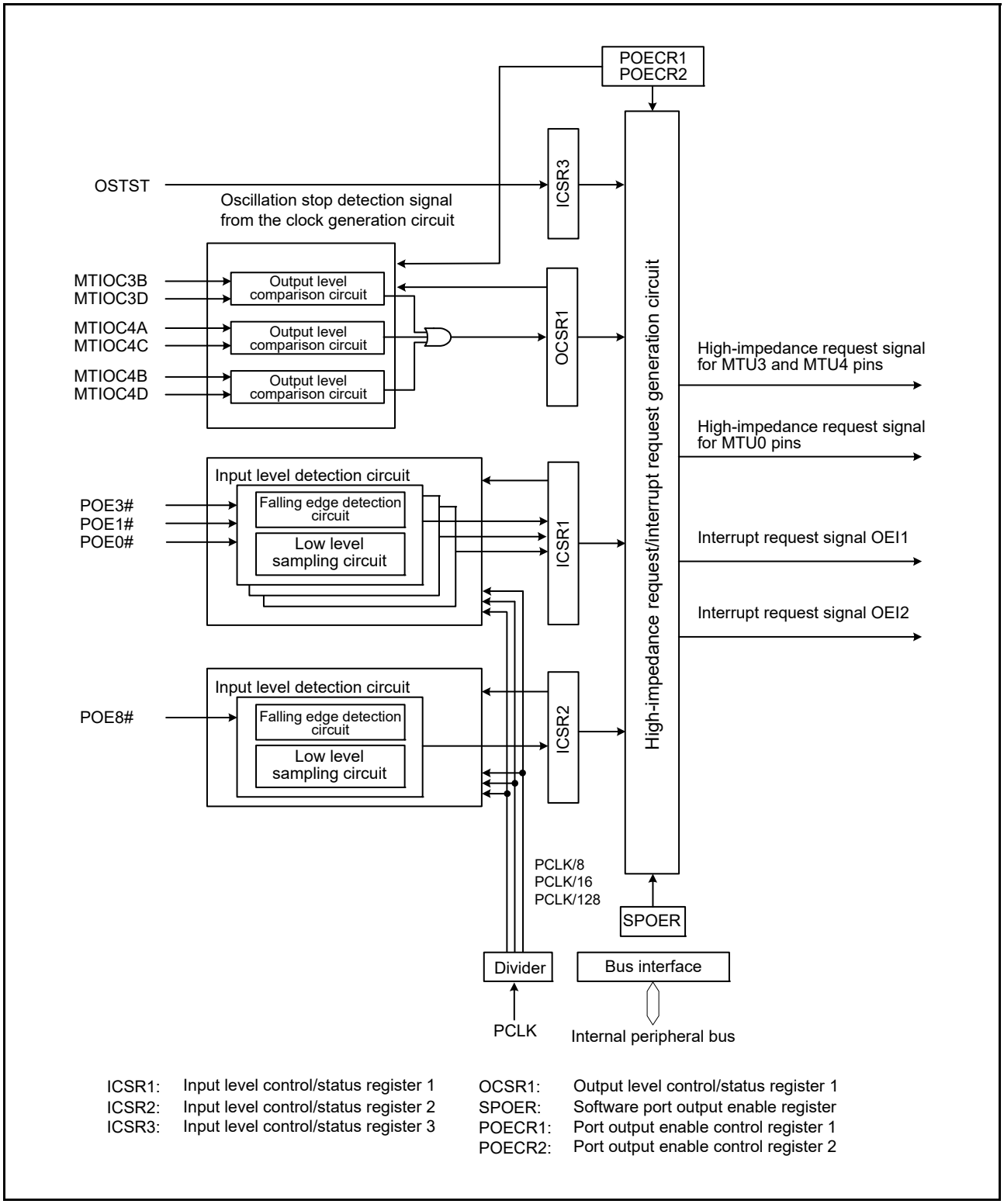


Figure 24.1 POE Block Diagram

Downloaded from Arrow.com.

Table 24.2 lists I/O pins to be used by the POE.

Table 24.2 POE I/O Pins

Pin Name	I/O	Description
POE0#, POE1#, POE3#	Input	Request signals to place the pins for MTU complementary PWM output in high-impedance.
POE8#	Input	Request signal to place the MTU0 output pins in high-impedance.
MTIOC3B	Output	MTU3 complementary PWM output pin
MTIOC3D	Output	MTU3 complementary PWM output pin
MTIOC4A	Output	MTU4 complementary PWM output pin
MTIOC4B	Output	MTU4 complementary PWM output pin
MTIOC4C	Output	MTU4 complementary PWM output pin
MTIOC4D	Output	MTU4 complementary PWM output pin
MTIOC0A	Output	MTU0 output pin
MTIOC0B	Output	MTU0 output pin
MTIOC0C	Output	MTU0 output pin

Table 24.3 lists output-level comparisons with pin combinations.

Table 24.3 Pin Combinations

Pin Combination	I/O	Description
MTIOC3B and MTIOC3D	Output	Pin combinations for output-level comparison and high-impedance control can be selected by POE registers. The pins for MTU complementary PWM output are placed in high-impedance when the pins simultaneously output an active level for one or more PCLK clock cycles. (When the MTU.TOCR1.TOCS bit = 0: The active level is low level if the MTU.TOCR1.OLSP and OLSN bits are 0, and the active level is high level if the MTU.TOCR1.OLSP and OLSN bits are 1. When the MTU.TOCR1.TOCS bit = 1: The active level is low level if the MTU.TOCR2.OLS3N, OLS3P, OLS2N, OLS2P, OLS1N, and OLS1P bits are 0, and the active level is high level if the MTU.TOCR2.OLS3N, OLS3P, OLS2N, OLS2P, OLS1N, and OLS1P bits are 1.)
MTIOC4A and MTIOC4C	Output	
MTIOC4B and MTIOC4D	Output	

Downloaded from Arrow.com.

24.2 Register Descriptions

24.2.1 Input Level Control/Status Register 1 (ICSR1)

Address(es): 0008 8900h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	POE3F	—	POE1F	POE0F	—	—	—	PIE1	POE3M[1:0]		—	—	POE1M[1:0]		POE0M[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	POE0M[1:0]	POE0 Mode Select	b1 b0 0 0: Accepts a high-impedance request on the falling edge of the POE0# pin input. 0 1: Accepts a high-impedance request when the POE0# pin input has been sampled 16 times at PCLK/8 clock cycles and all are low level. 1 0: Accepts a high-impedance request when POE0# input has been sampled 16 times at PCLK/16 clock pulses and all are low level. 1 1: Accepts a high-impedance request when POE0# input has been sampled 16 times at PCLK/128 clock pulses and all are low level.	R/W*1
b3, b2	POE1M[1:0]	POE1 Mode Select	b3 b2 0 0: Accepts a high-impedance request on the falling edge of the POE1# pin input. 0 1: Accepts a high-impedance request when the POE1# pin input has been sampled 16 times at PCLK/8 clock cycles and all are low level. 1 0: Accepts a high-impedance request when POE1# input has been sampled 16 times at PCLK/16 clock pulses and all are low level. 1 1: Accepts a high-impedance request when POE1# input has been sampled 16 times at PCLK/128 clock pulses and all are low level.	R/W*1
b5, b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7, b6	POE3M[1:0]	POE3 Mode Select	b7 b6 0 0: Accepts a high-impedance request on the falling edge of the POE3# pin input. 0 1: Accepts a high-impedance request when the POE3# pin input has been sampled 16 times at PCLK/8 clock cycles and all are low level. 1 0: Accepts a high-impedance request when POE3# input has been sampled 16 times at PCLK/16 clock pulses and all are low level. 1 1: Accepts a high-impedance request when POE3# input has been sampled 16 times at PCLK/128 clock pulses and all are low level.	R/W*1
b8	PIE1	Port Interrupt Enable 1	0: OEI1 interrupt requests by the input level detection disabled 1: OEI1 interrupt requests by the input level detection enabled	R/W
b11 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12	POE0F	POE0 Flag	0: Indicates that a high-impedance request has not been input to the POE0# pin. 1: Indicates that a high-impedance request has been input to the POE0# pin.	R/(W)*2
b13	POE1F	POE1 Flag	0: Indicates that a high-impedance request has not been input to the POE1# pin. 1: Indicates that a high-impedance request has been input to the POE1# pin.	R/(W)*2
b14	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b15	POE3F	POE3 Flag	0: Indicates that a high-impedance request has not been input to the POE3# pin. 1: Indicates that a high-impedance request has been input to the POE3# pin.	R/(W)*2

Note 1. Can be modified only once after a reset.
Note 2. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.

Downloaded from Arrow.com.

When low-level sampling has been set by the POE0M[1:0], POE1M[1:0], and POE3M[1:0] bits, writing 0 to the POE0F, POE1F, and POE3F flags requires high-level input on the POE0#, POE1F, and POE3# pins.
For details, refer to section 24.3.6, Release from the High-Impedance.

PIE1 Bit (Port Interrupt Enable 1)

This bit enables or disables OEI1 interrupt requests when any one of the POE0F, POE1F, and POE3F flags is set to 1.

POE0F Flag (POE0 Flag)

This flag indicates that a high-impedance request has been input to the POE0# pin.
[Setting condition]

- When the input set by POE0M[1:0] occurs at the POE0# pin

[Clearing condition]

- By writing 0 to POE0F after reading POE0F = 1

POE1F Flag (POE1 Flag)

This flag indicates that a high-impedance request has been input to the POE1# pin.
[Setting condition]

- When the input set by POE1M[1:0] occurs at the POE1# pin

[Clearing condition]

- By writing 0 to POE1F after reading POE1F = 1

POE3F Flag (POE3 Flag)

This flag indicates that a high-impedance request has been input to the POE3# pin.
[Setting condition]

- When the input set by POE3M[1:0] occurs at the POE3# pin

[Clearing condition]

- By writing 0 to POE3F after reading POE3F = 1

Downloaded from Arrow.com.

24.2.2 Output Level Control/Status Register 1 (OCSR1)

Address(es): 0008 8902h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	OSF1	—	—	—	—	—	OCE1	OIE1	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	OIE1	Output Short Interrupt Enable 1	0: OEI1 interrupt requests by the output level comparison disabled 1: OEI1 interrupt requests by the output level comparison enabled	R/W
b9	OCE1	Output Short High-Impedance Enable 1	0: Does not place the pins in high-impedance. 1: Places the pins in high-impedance.	R/W*1
b14 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	OSF1	Output Short Flag 1	0: Indicates that outputs have not simultaneously become an active level. 1: Indicates that outputs have simultaneously become an active level.	R/(W) *2

Note 1. Can be modified only once after a reset.
Note 2. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.

OIE1 Bit (Output Short Interrupt Enable 1)

This bit enables or disables OEI1 interrupt requests when the OSF1 flag is set to 1.

OCE1 Bit (Output Short High-Impedance Enable 1)

This bit specifies whether to place the MTU complementary PWM output pins in high-impedance when the OSF1 flag is set to 1.

OSF1 Flag (Output Short Flag 1)

This flag indicates that any one of the three pairs of two-phase outputs for MTU complementary PWM output to be compared in Table 24.3 has simultaneously become an active level. If the POECR2.PnCZEA (n = 1, 2, 3) bits are 0 or the output comparison function of the MTU is not enabled, the OSF1 flag will not be set to 1 even if both pins in the corresponding complementary output pair of the MTU are simultaneously active. The active levels are determined by the MTU.TOCR1 and TOCR2 registers.

[Setting condition]

- When any one of the three pairs of two-phase outputs has simultaneously become an active level*1

[Clearing condition]

- By writing 0 to OSF1 after reading OSF1 = 1
  The complementary output pins for the MTU must be at the inactive level when 0 is written to the flag.
  For details, refer to section 24.3.6, Release from the High-Impedance.

Note 1. The setting condition is judged only by the level of the pin regardless the setting of the MPC.PmnPFS register.

Downloaded from Arrow.com.

24.2.3 Input Level Control/Status Register 2 (ICSR2)

Address(es): 0008 8908h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	POE8F	—	—	POE8E	PIE2	—	—	—	—	—	—	POE8M[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	POE8M[1:0]	POE8 Mode Select	b1 b0 0 0: Accepts a high-impedance request on the falling edge of the POE8# pin input 0 1: Accepts a high-impedance request when the POE8# pin input has been sampled 16 times at PCLK/8 clock cycles and all are low level. 1 0: Accepts a high-impedance request when the POE8# pin input has been sampled 16 times at PCLK/16 clock cycles and all are low level. 1 1: Accepts a high-impedance request when the POE8# pin input has been sampled 16 times at PCLK/128 clock cycles and all are low level.	R/W*1
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	PIE2	Port Interrupt Enable 2	0: OEI2 interrupt requests disabled 1: OEI2 interrupt requests enabled	R/W
b9	POE8E	POE8 High-Impedance Enable	0: Does not place the MTIOC0A, MTIOC0B, and MTIOC0C pins in high-impedance. 1: Places the MTIOC0A, MTIOC0B, and MTIOC0C pins in high-impedance.	R/W*1
b11, b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12	POE8F	POE8 Flag	0: Indicates that a high-impedance request has not been input to the POE8# pin. 1: Indicates that a high-impedance request has been input to the POE8# pin.	R/(W)*2
b15 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.
Note 2. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.

PIE2 Bit (Port Interrupt Enable 2)

This bit enables or disables OEI2 interrupt requests when the POE8F flag is set to 1.

POE8E Bit (POE8 High-Impedance Enable)

This bit specifies whether to place the MTU0 pins in high-impedance when the POE8F flag is set to 1.

POE8F Flag (POE8 Flag)

This flag indicates that a high-impedance request has been input to the POE8# pin.

[Setting condition]

- When the input set by ICSR2.POE8M[1:0] bits occurs at the POE8# pin

[Clearing conditions]

- Writing 0 to POE8F after reading POE8F = 1
  When writing 0 to the flag while low-level sampling is selected for the ICSR2.POE8M[1:0] bits, the POE8# pin input must be at the high level.
  For details, refer to section 24.3.6, Release from the High-Impedance.

Downloaded from Arrow.com.

24.2.4 Software Port Output Enable Register (SPOER)

Address(es): 0008 890Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	CH0HIZ	CH34HIZ
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CH34HIZ	MTU3 and MTU4 Output High-Impedance Enable	0: Does not place the pins in high-impedance. 1: Places the pins in high-impedance.	R/W
b1	CH0HIZ	MTU0 Output High-Impedance Enable	0: Does not place the pins in high-impedance. 1: Places the pins in high-impedance.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

CH34HIZ Bit (MTU3 and MTU4 Output High-Impedance Enable)

This bit selects whether to place the MTU complementary PWM output pins (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4B, MTIOC4C, MTIOC4D) in high-impedance.

[Setting conditions]

- By writing 1 to CH34HIZ
- An event signal from the event link controller (ELC) is received.

[Clearing condition]

- By writing 0 to CH34HIZ after reading CH34HIZ = 1

CH0HIZ Bit (MTU0 Output High-Impedance Enable)

This bit selects whether to place the MTU0 pins (MTIOC0A, MTIOC0B, MTIOC0C) in high-impedance.

[Setting conditions]

- By writing 1 to CH0HIZ
- An event signal from the event link controller (ELC) is received.

[Clearing condition]

- By writing 0 to CH0HIZ after reading CH0HIZ = 1

Downloaded from Arrow.com.

24.2.5 Port Output Enable Control Register 1 (POECR1)

Address(es): 0008 890Bh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	PE2ZE	PE1ZE	PE0ZE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PE0ZE	MTIOC0A High-Impedance Enable	0: Does not place the pin in high-impedance. 1: Places the pin in high-impedance.	R/W*1
b1	PE1ZE	MTIOC0B High-Impedance Enable	0: Does not place the pin in high-impedance. 1: Places the pin in high-impedance.	R/W*1
b2	PE2ZE	MTIOC0C High-Impedance Enable	0: Does not place the pin in high-impedance. 1: Places the pin in high-impedance.	R/W*1
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.

Downloaded from Arrow.com.

24.2.6 Port Output Enable Control Register 2 (POECR2)

Address(es): 0008 890Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	P1CZEA	P2CZEA	P3CZEA	—	—	—	—
Value after reset:	0	1	1	1	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	P3CZEA	MTU Port 3 High-Impedance Enable	0: Comparison of output levels does not proceed and the pins are not placed in the high-impedance. 1: The pins are placed in the high-impedance.	R/W*1
b5	P2CZEA	MTU Port 2 High-Impedance Enable	0: Comparison of output levels does not proceed and the pins are not placed in the high-impedance. 1: The pins are placed in the high-impedance.	R/W*1
b6	P1CZEA	MTU Port 1 High-Impedance Enable	0: Comparison of output levels does not proceed and the pins are not placed in the high-impedance. 1: The pins are placed in the high-impedance.	R/W*1
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.

When this function is not used, write 00h to this register.

P3CZEA Bit (MTU Port 3 High-Impedance Enable)

This bit gives permission regarding whether or not the MTIOC4B and MTIOC4D pins for complementary PWM output from the MTU are placed in the high-impedance. It also gives permission regarding whether or not the levels on the MTIOC4B and MTIOC4D pins are compared.

P2CZEA Bit (MTU Port 2 High-Impedance Enable)

This bit gives permission regarding whether or not the MTIOC4A and MTIOC4C pins for complementary PWM output from the MTU are placed in the high-impedance. It also gives permission regarding whether or not the levels on the MTIOC4A and MTIOC4C pins are compared.

P1CZEA Bit (MTU Port 1 High-Impedance Enable)

This bit gives permission regarding whether or not the MTIOC3B and MTIOC3D pins for complementary PWM output from the MTU are placed in the high-impedance. It also gives permission regarding whether or not the levels on the MTIOC3B and MTIOC3D pins are compared.

Downloaded from Arrow.com.

24.2.7 Input Level Control/Status Register 3 (ICSR3)

Address(es): 0008 890Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	OSTSTF	—	—	OSTSTE	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b8 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b9	OSTSTE	OSTST High-Impedance Enable	0: Does not place the MTIOC0A, MTIOC0B, MTIOC0C, MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4B, MTIOC4C, and MTIOC4D pins in high-impedance. 1: Places the MTIOC0A, MTIOC0B, MTIOC0C, MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4B, MTIOC4C, and MTIOC4D pins in high-impedance.	R/W*1
b11, b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12	OSTSTF	OSTST High-Impedance Flag	0: Oscillation stop is not producing a request to place pins in the high-impedance. 1: Oscillation stop is producing a request to place pins in the high-impedance.	R/(W)*2
b15 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.
Note 2. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.

OSTSTE Bit (OSTST High-Impedance Enable)

This bit permits or prohibits placement of pins for complementary PWM output from MTU and output pins for MTU0 in the high-impedance on detection that oscillation has stopped.

OSTSTF Flag (OSTST High-Impedance Flag)

The OSTSTF flag is a status flag that indicates the state of requests to place pins in the high-impedance due to oscillation having stopped. The value of the flag becomes 1 when oscillation stops. Ensure that the oscillation-stopped detection signal is negated when clearing the flag by writing 0 to it. Writing 0 to the OSTSTF flag will not clear the flag while the oscillation-stopped detection signal is being asserted; in other words, it will not clear the flag before 10 PCLK clock cycles have elapsed after stopped oscillation was detected.

[Setting condition]

- Detection of the oscillation-stopped state

[Clearing condition]

- Writing 0 to the bit after having read its value as 1.

Downloaded from Arrow.com.

24.3 Operation

The target pins for high-impedance control and conditions to place the pins in high-impedance are described below.

(1) MTU0 pin (MTIOC0A)

When any of the following conditions is satisfied, the pin is placed to the high-impedance state.

- POE8# input level detection
  When the ICSR2.POE8F flag is set to 1 with POECR1.PE0ZE and ICSR2.POE8E set to 1.
- SPOER setting
  When the SPOER.CH0HIZ bit is set to 1 with POECR1.PE0ZE set to 1.
- Detection of stopped oscillation
  When the OSTSTF flag is set to 1 with POECR1.PE0ZE and ICSR3.OSTSTE set to 1.
- Event signal reception from the ELC

(2) MTU0 pin (MTIOC0B)

When any of the following conditions is satisfied, the pin is placed to the high-impedance state.

- POE8# input level detection
  When the ICSR2.POE8F flag is set to 1 with POECR1.PE1ZE and ICSR2.POE8E set to 1.
- SPOER setting
  When the SPOER.CH0HIZ bit is set to 1 with POECR1.PE1ZE set to 1.
- Detection of stopped oscillation
  When the OSTSTF flag is set to 1 with POECR1.PE1ZE and ICSR3.OSTSTE set to 1.
- Event signal reception from the ELC

(3) MTU0 pin (MTIOC0C)

When any of the following conditions is satisfied, the pin is placed to the high-impedance state.

- POE8# input level detection
  When the ICSR2.POE8F flag is set to 1 with POECR1.PE2ZE and ICSR2.POE8E set to 1.
- SPOER setting
  When the SPOER.CH0HIZ bit is set to 1 with POECR1.PE2ZE set to 1.
- Detection of stopped oscillation
  When the OSTSTF flag is set to 1 with POECR1.PE2ZE and ICSR3.OSTSTE set to 1.
- Event signal reception from the ELC

(4) MTU3 pins (MTIOC3B and MTIOC3D)

When any of the following conditions is satisfied, the pins are placed to the high-impedance state.

- POE0#, POE1#, and POE3# input level detection
  When the ICSR1.POE3F, POE1F, or POE0F flag is set to 1 with POECR2.P1CZEA set to 1.
- MTIOC3B and MTIOC3D output level comparison
  When the OCSR1.OSF1 flag is set to 1 with POECR2.P1CZEA and OCSR1.OCE1 set to 1.
- SPOER setting
  When the SPOER.CH34HIZ bit is set to 1 with POECR2.P1CZEA set to 1.
- Detection of stopped oscillation
  When the ICSR3.OSTSTF flag is set to 1 with POECR2.P1CZEA and ICSR3.OSTSTE set to 1.
- Event signal reception from the ELC

Downloaded from Arrow.com.

(5) MTU4 pins (MTIOC4A and MTIOC4C)

When any of the following conditions is satisfied, the pins are placed to the high-impedance state.

- POE0#, POE1#, and POE3# input level detection
  When the ICSR1.POE3F, POE1F, or POE0F flag is set to 1 with POECR2.P2CZEA set to 1.
- MTIOC4A and MTIOC4C output level comparison
  When the OCSR1.OSF1 flag is set to 1 with POECR2.P2CZEA and OCSR1.OCE1 set to 1.
- SPOER setting
  When the SPOER.CH34HIZ bit is set to 1 with POECR2.P2CZEA set to 1.
- Detection of stopped oscillation
  When the ICSR3.OSTSTF flag is set to 1 with POECR2.P2CZEA and ICSR3.OSTSTE set to 1.
- Event signal reception from the ELC

(6) MTU4 pins (MTIOC4B and MTIOC4D)

When any of the following conditions is satisfied, the pins are placed to the high-impedance state.

- POE0#, POE1#, and POE3# input level detection
  When the ICSR1.POE3F, POE1F, or POE0F flag is set to 1 with POECR2.P3CZEA set to 1.
- MTIOC4B and MTIOC4D output level comparison
  When the OCSR1.OSF1 flag is set to 1 with POECR2.P3CZEA and OCSR1.OCE1 set to 1.
- SPOER setting
  When the SPOER.CH34HIZ bit is set to 1 with POECR2.P3CZEA set to 1.
- Detection of stopped oscillation
  When the ICSR3.OSTSTF flag is set to 1 with POECR2.P3CZEA and ICSR3.OSTSTE set to 1.
- Event signal reception from the ELC

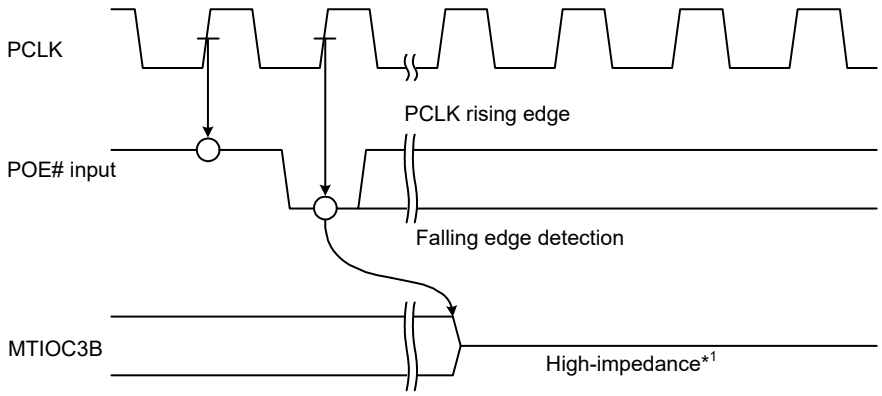
Downloaded from Arrow.com.

24.3.1 Input Level Detection Operation

If the input conditions set by the ICSR1 and ICSR2 registers occur on the POE0# to POE3# and POE8# pins, the pins for the MTU complementary PWM output and MTU0 are placed in high-impedance.

(1) Falling Edge Detection

When a change from a high to low level is input to the POE0#, POE1#, POE3# and POE8# pins, the pins for the MTU complementary PWM output and MTU0 are placed in high-impedance.
A falling edge is detected after PCLK causes sampling to proceed. If the low level is input to the POE0#, POE1#, POE3# or POE8# pin over less than one PCLK cycle, whether the falling edge will or will not be detected cannot be guaranteed.
Figure 24.2 shows the timing of sampling after the level changes in input to the POE0#, POE1#, POE3# and POE8# pins until the respective pins enter high-impedance.



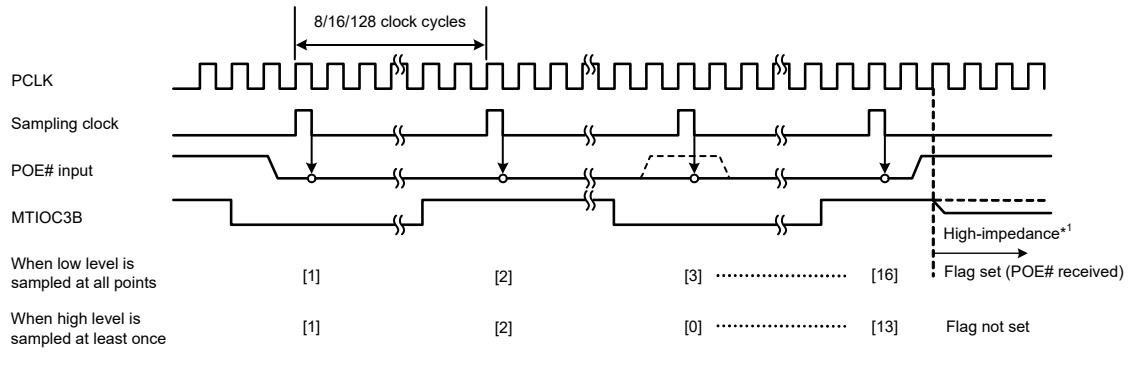
Note 1. The other MTU complementary PWM output pins and MTU0 pins also enter the high-impedance in the similar timing.

Figure 24.2 Falling Edge Detection

Downloaded from Arrow.com.

(2) Low-Level Detection

Figure 24.3 shows the low-level detection operation. When a low level is detected 16 times continuously with the sampling clock selected by the ICSR1 and ICSR2 registers, the detected level is recognized as low, and the pins for the MTU complementary PWM output and MTU0 are placed in high-impedance. If even one high level is detected during this interval, the detected level is not recognized as low. Furthermore, in an interval over which the sampling clock is not being output, changes to the levels on the POE0#, POE1#, POE3# and POE8# pins are ignored.

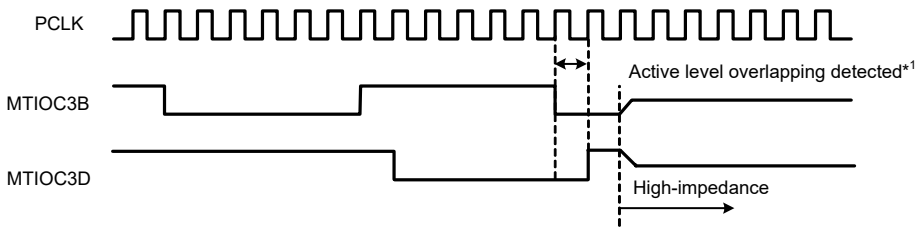


Note 1. The other MTU complementary PWM output pins and MTU0 pins also enter the high-impedance in the similar timing.

Figure 24.3 Low-Level Detection Operation

24.3.2 Output-Level Compare Operation

Figure 24.4 shows an example of the output-level compare operation for the combination of MTIOC3B and MTIOC3D (MTU complementary PWM output pins). The operation is the same for the other pin combinations.



Note 1. In this case, the low level has been set as the active level for the MTIOC3B and MTIOC3D pins.

Figure 24.4 Output-Level Compare Operation

Downloaded from Arrow.com.

24.3.3 High-Impedance Control Using Registers

The high-impedance of the MTU complementary PWM output and MTU0 pins can be directly controlled by writing to the software port output enable register (SPOER).
Setting the SPOER.CH34HIZ bit to 1 places the MTU complementary PWM output pins (MTU3 and MTU4) specified by the POECR2 register in the high-impedance.
Setting the SPOER.CH0HIZ bit to 1 places the MTU0 output pins specified by port output enable control register 1 (POECR1) in the high-impedance.

24.3.4 High-Impedance Control on Detection of Stopped Oscillation

When the oscillation stop detection function in the clock generation circuit detects stopped oscillation while the ICSR3.OSTSTE bit is 1, the MTU complementary PWM output pins specified by the POECR2 register and the MTU0 output pins specified by the POECR1 register are placed in the high-impedance.

24.3.5 High-Impedance Control in Response to Receiving an Event Signal from the ELC

The MTU complementary PWM output and MTU0 pins can be placed in the high-impedance state in response to an event signal from the ELC.
To control the high-impedance state of the MTU complementary PWM output and MTU0 pins, preset the corresponding register (POECR1 or POECR2) to enable the high-impedance state. When an event signal is received from the ELC, the corresponding bit (SPOER.CH0HIZ or SPOER.CH34HIZ) is set to 1, and the MTU complementary PWM output pins or MTU0 pins are placed in the high-impedance state.

24.3.6 Release from the High-Impedance

Pins for complementary PWM output from MTU and pins for MTU0 which have been placed in the high-impedance due to input-level detection can be released from that state by either returning them to their initial state with a reset or clearing all of the ICSR1.POE3F, POE1F and POE0F flags and the ICSR2.POE8F flag. Note, however, that when low-level sampling is selected by the ICSR1.POE3M[1:0], POE1M[1:0], and POE0M[1:0] bits, and the ICSR2.POE8M[1:0] bits, if a high level is being input to the corresponding pin from among POE0#, POE1#, POE3# and POE#8 but has not yet been detected, writing 0 to the flag is ignored (the flag is not cleared).
MTU complementary PWM output pins which have been placed in the high-impedance due to output-level comparison can be released from that state by either returning them to their initial state with a reset or clearing the OCSR1.OSF1 flag. Note, however, that if the inactive level is not yet being output from the MTU complementary PWM output pins, writing 0 to the flag is ignored (the flag is not cleared). Inactive-level outputs can be obtained by setting the MTU registers.
For MTU complementary PWM output pins and pins for MTU0 that have been placed in the high-impedance because oscillation by the clock generation circuit has stopped, clearing the ICSR3.OSTSTF or ICSR3.OSTSTE bit releases the pins from the high-impedance.
For MTU complementary PWM output pins and pins for MTU0 that have been placed in the high-impedance by the SPOER.CH34HIZ or SPOER.CH0HIZ bit, clearing the corresponding bits (SPOER.CH34HIZ and SPOER.CH0HIZ) releases the pins from the high-impedance.

Downloaded from Arrow.com.

24.4 Interrupts

The POE issues a request to generate an interrupt when the corresponding condition below is matched during input-level detection, output-level comparison, or oscillation stop by the clock generation circuit. Table 24.4 lists the interrupt sources and their request conditions. On acceptance of an OEI1 or OEI2 interrupt, the first line of the exception handling routine for the given interrupt should confirm that the flag for the given flag has been set to 1.

Table 24.4 Interrupt Sources and Conditions

Name	Interrupt Source	Interrupt Flag	Condition
OEI1	Output enable interrupt 1	POE0F, POE1F, POE3F, OSF1	When ICSR1.POE0F, POE1F, or POE3F flag is set to 1 with ICSR1.PIE1 set to 1, or when OCSR1.OSF1 flag is set to 1 with OCSR1.OIE1 set to 1.
OEI2	Output enable interrupt 2	POE8F	When ICSR2.POE8F flag is set to 1 with ICSR2.PIE2 set to 1.

24.5 Usage Notes

24.5.1 Transitions to Software Standby Mode

When the POE is used, do not make a transition to software standby mode. In this mode, the POE stops and thus the high-impedance of pins cannot be controlled.

24.5.2 When the POE Is Not Used

When the POE is not used, write 00h to port output enable control registers 1 and 2 (POECR1 and POECR2), respectively.

24.5.3 Specifying Pins Corresponding to the MTU

The POE controls high-impedance outputs only when a pin has been specified so that the pin corresponds to the MTU by setting the PMR and PmnPFS registers. When the pin has been specified as a general I/O pin, the POE does not control high-impedance outputs.

24.5.4 Notes on High-Impedance Control by Event Signal Reception from the ELC

When writing 0 to the SPOER.CH34HIZ or SPOER.CH0HIZ bit and receiving an event signal conflict, the event signal takes priority and the corresponding bit is set to 1. If the MTU complementary PWM output and MTU0 pins are placed in the high-impedance state when an event signal is received from the ELC, no interrupt request is generated.

Downloaded from Arrow.com.

25. 16-Bit Timer Pulse Unit (TPUa)

This MCU has on-chip 16-bit timer pulse units (TPU) comprising six-channel 16-bit timers.
In this section, “PCLK” is used to refer to PCLKB.

25.1 Overview

Specifications of the TPU are listed in Table 25.1. Functions of TPU are listed in Table 25.2.
Figure 25.1 shows a block diagram of TPU.

Table 25.1 Specifications of TPU

Item	Description
Pulse input/output	Maximum 10
Count clocks	Seven or eight types are provided for each channel.
Settable operations	• Waveform output at compare match • Input capture function (noise filters can be set) • Counter clear operation • Simultaneous writing to multiple timer counters (TCNT) • Simultaneous clearing by compare match and input capture • Synchronous input/output for registers by counter synchronous operation • Maximum of 9-phase PWM output by combination with synchronous operation • Cascaded operation
TPU0 and TPU3	Buffer operation can be set.
TPU1, TPU2, TPU4, and TPU5	Phase counting mode can be set.
Interrupt sources	26 sources
Buffer operation	Automatic transfer of register data
Generation of trigger	Conversion start trigger for the A/D converter can be generated.
Low power consumption function	Module stop state can be set.

Downloaded from Arrow.com.

Table 25.2 TPU Functions

Item		TPU0	TPU1	TPU2	TPU3	TPU4	TPU5
Count clocks		PCLK/1 PCLK/4 PCLK/16 PCLK/64 TCLKA TCLKB TCLKC TCLKD	PCLK/1 PCLK/4 PCLK/16 PCLK/64 PCLK/256 TCLKA TCLKB	PCLK/1 PCLK/4 PCLK/16 PCLK/64 PCLK/1024 TCLKA TCLKB TCLKC	PCLK/1 PCLK/4 PCLK/16 PCLK/64 PCLK/256 PCLK/1024 PCLK/4096 TCLKA	PCLK/1 PCLK/4 PCLK/16 PCLK/64 PCLK/1024 TCLKA TCLKC	PCLK/1 PCLK/4 PCLK/16 PCLK/64 PCLK/256 TCLKA TCLKC TCLKD
External clocks for phase counting mode		Not possible	TCLKA TCLKB	TCLKC TCLKD	Not possible	TCLKC TCLKD	TCLKA TCLKB
Timer general registers		TGRA TGRB TGRC*1 TGRD*1	TGRA TGRB	TGRA TGRB	TGRA TGRB TGRC*1 TGRD*1	TGRA TGRB	TGRA TGRB
I/O pins		TIOCB0	TIOCB1	TIOCB2	TIOCA3 TIOCB3 TIOCC3 TIOCD3	TIOCA4 TIOCB4	TIOCB5
Counter clear function (y = A to D)		TGRy compare match or input capture	TGRy compare match or input capture	TGRy compare match or input capture	TGRy compare match or input capture	TGRy compare match or input capture	TGRy compare match or input capture
Compare match output	Low output	Possible	Possible	Possible	Possible	Possible	Possible
	High output	Possible	Possible	Possible	Possible	Possible	Possible
	Toggle output	Possible	Possible	Possible	Possible	Possible	Possible
Input capture function		Possible	Possible	Possible	Possible	Possible	Possible
Synchronous operation		Possible	Possible	Possible	Possible	Possible	Possible
PWM mode		Possible	Possible	Possible	Possible	Possible	Possible
Phase counting mode		Not possible	Possible	Possible	Not possible	Possible	Possible
Buffer operation		Possible	Not possible	Not possible	Possible	Not possible	Not possible
DTC activation (y = A to D)		TGRy compare match or input capture	TGRy compare match or input capture	TGRy compare match or input capture	TGRy compare match or input capture	TGRy compare match or input capture	TGRy compare match or input capture
DMAC activation		TGRA compare match	TGRA compare match	TGRA compare match	TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match
A/D conversion start trigger		TGRA compare match	TGRA compare match	TGRA compare match	TGRA compare match or input capture	TGRA compare match or input capture	Not possible
Interrupt sources		5 sources • Compare match 0A • Compare match or input capture 0B • Compare match 0C • Compare match 0D • Overflow	4 sources • Compare match 1A • Compare match or input capture 1B • Overflow • Underflow	4 sources • Compare match 2A • Compare match or input capture 2B • Overflow • Underflow	5 sources • Compare match or input capture 3A • Compare match or input capture 3B • Compare match or input capture 3C • Compare match or input capture 3D • Overflow	4 sources • Compare match or input capture 4A • Compare match or input capture 4B • Overflow • Underflow	4 sources • Compare match 5A • Compare match or input capture 5B • Overflow • Underflow
Module stop setting*2		MSTPCRA.MSTPA13 bit					

Note 1. TGRC and TGRD can be set as a buffer register.
Note 2. For details, see section 11, Low Power Consumption.

Downloaded from Arrow.com.

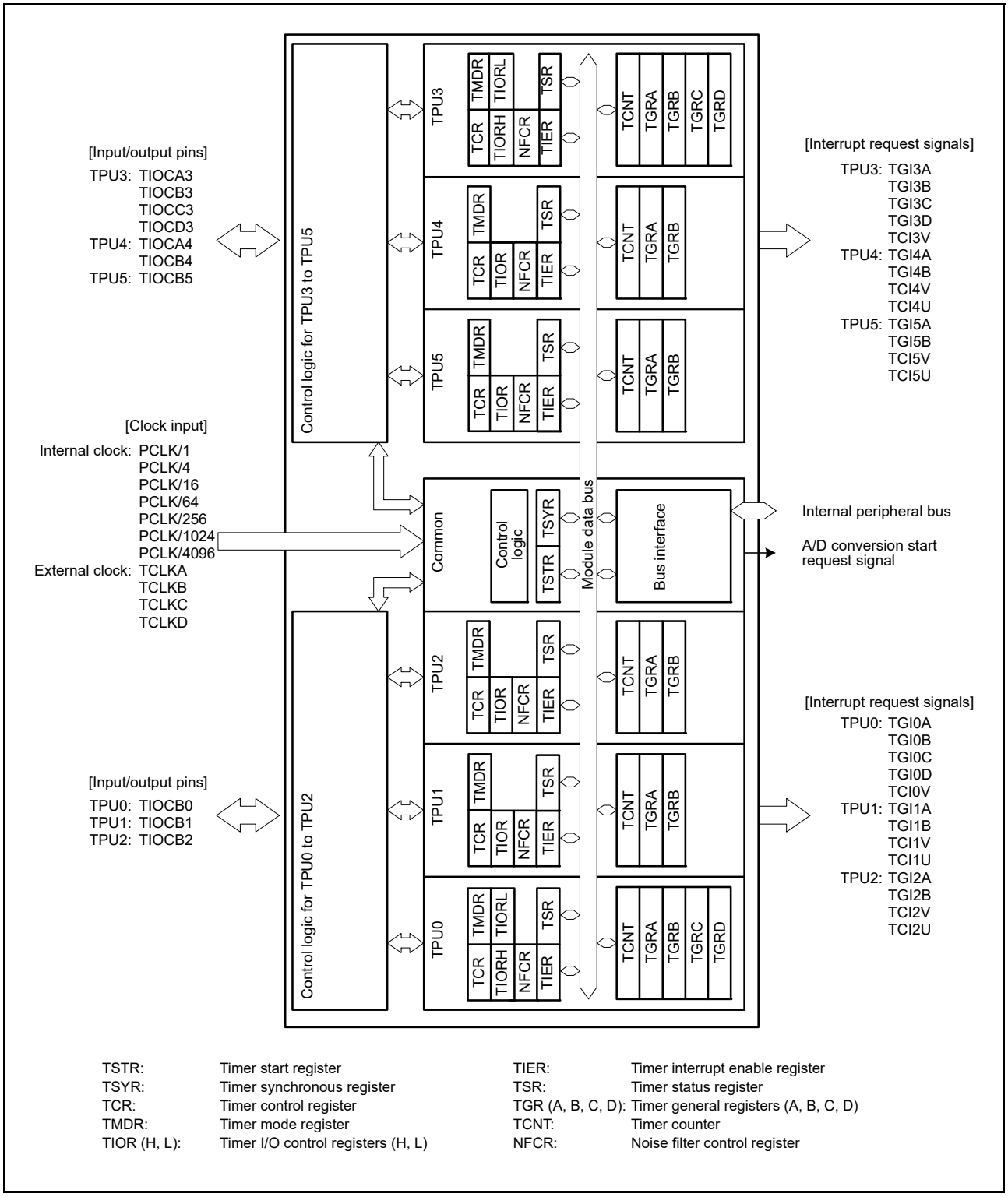


Figure 25.1 Block Diagram of TPU

Downloaded from Arrow.com.

Table 25.3 lists the input/output pins of the TPU.

Table 25.3 Pin Configuration of TPU

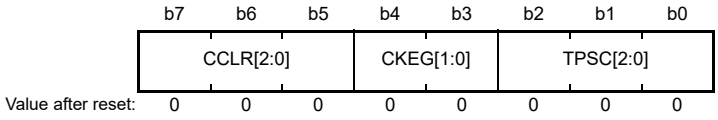
Channel	Pin Name	I/O	Description
Common	TCLKA	Input	External clock A input pin (TPU1 and TPU5 phase counting mode A phase input)
	TCLKB	Input	External clock B input pin (TPU1 and TPU5 phase counting mode B phase input)
	TCLKC	Input	External clock C input pin (TPU2 and TPU4 phase counting mode A phase input)
	TCLKD	Input	External clock D input pin (TPU2 and TPU4 phase counting mode B phase input)
TPU0	TIOCB0	I/O	TPU0.TGRB input capture input/output compare output/PWM output pin
TPU1	TIOCB1	I/O	TPU1.TGRB input capture input/output compare output/PWM output pin
TPU2	TIOCB2	I/O	TPU2.TGRB input capture input/output compare output/PWM output pin
TPU3	TIOCA3	I/O	TPU3.TGRA input capture input/output compare output/PWM output pin
	TIOCB3	I/O	TPU3.TGRB input capture input/output compare output/PWM output pin
	TIOCC3	I/O	TPU3.TGRC input capture input/output compare output/PWM output pin
	TIOCD3	I/O	TPU3.TGRD input capture input/output compare output/PWM output pin
TPU4	TIOCA4	I/O	TPU4.TGRA input capture input/output compare output/PWM output pin
	TIOCB4	I/O	TPU4.TGRB input capture input/output compare output/PWM output pin
TPU5	TIOCB5	I/O	TPU5.TGRB input capture input/output compare output/PWM output pin

Downloaded from Arrow.com.

25.2 Register Descriptions

25.2.1 Timer Control Register (TCR)

Address(es): TPU0.TCR 0008 8110h, TPU1.TCR 0008 8120h, TPU2.TCR 0008 8130h, TPU3.TCR 0008 8140h, TPU4.TCR 0008 8150h, TPU5.TCR 0008 8160h



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	TPSC[2:0]	Timer Prescaler Select	See Table 25.4 to Table 25.9.	R/W
b4, b3	CKEG[1:0]	Input Clock Edge Select	See Table 25.10.	R/W
b7 to b5	CCLR[2:0]	Counter Clear Source Select*1	See Table 25.11 and Table 25.12.	R/W

Note 1. Bit 7 is reserved in TPU1, TPU2, TPU4, and TPU5. These bits are read as 0. The write value should be 0.

TPUm.TCR settings should be made while TPUm.TCNT operation is stopped.

TPSC[2:0] Bits (Timer Prescaler Select)

These bits select the TCNT clock. The clock source can be selected independently for each channel.
To select the external clock as the clock source, set the bit in the port direction register (PDR) for the corresponding pin to 0 (input port), and set the bit in the port mode register (PMR) to 1 (uses the pin as an I/O port for peripheral functions). For details, see section 21, I/O Ports.

CKEG[1:0] Bits (Input Clock Edge Select)

These bits select the input clock edge.
When the internal clock is counted using both edges, the input clock period is halved (e.g. Both edges of PCLK/4 = PCLK/2 rising edge).
Internal clock edge selection is valid when the input clock is PCLK/4 or slower. This setting is ignored if the input clock is PCLK/1, or when overflow/underflow of another channel is selected.

Downloaded from Arrow.com.

Table 25.4 Bits TPSC[2:0] (TPU0)

Channel	Bits TPSC[2:0]			Description
	b2	b1	b0	
TPU0	0	0	0	Internal clock: counts on PCLK/1
	0	0	1	Internal clock: counts on PCLK/4
	0	1	0	Internal clock: counts on PCLK/16
	0	1	1	Internal clock: counts on PCLK/64
	1	0	0	External clock: counts on TCLKA pin input
	1	0	1	External clock: counts on TCLKB pin input
	1	1	0	External clock: counts on TCLKC pin input
	1	1	1	External clock: counts on TCLKD pin input

Table 25.5 Bits TPSC[2:0] (TPU1)

Channel	Bits TPSC[2:0]			Description
	b2	b1	b0	
TPU1	0	0	0	Internal clock: counts on PCLK/1
	0	0	1	Internal clock: counts on PCLK/4
	0	1	0	Internal clock: counts on PCLK/16
	0	1	1	Internal clock: counts on PCLK/64
	1	0	0	External clock: counts on TCLKA pin input
	1	0	1	External clock: counts on TCLKB pin input
	1	1	0	Internal clock: counts on PCLK/256
	1	1	1	Counts on TPU2.TCNT overflow/underflow

Note: This setting is invalid when TPU1 is in phase counting mode.

Table 25.6 Bits TPSC[2:0] (TPU2)

Channel	Bits TPSC[2:0]			Description
	b2	b1	b0	
TPU2	0	0	0	Internal clock: counts on PCLK/1
	0	0	1	Internal clock: counts on PCLK/4
	0	1	0	Internal clock: counts on PCLK/16
	0	1	1	Internal clock: counts on PCLK/64
	1	0	0	External clock: counts on TCLKA pin input
	1	0	1	External clock: counts on TCLKB pin input
	1	1	0	External clock: counts on TCLKC pin input
	1	1	1	Internal clock: counts on PCLK/1024

Note: This setting is invalid when TPU2 is in phase counting mode.

Downloaded from Arrow.com.

Table 25.7 Bits TPSC[2:0] (TPU3)

Channel	Bits TPSC[2:0]			Description
	b2	b1	b0	
TPU3	0	0	0	Internal clock: counts on PCLK/1
	0	0	1	Internal clock: counts on PCLK/4
	0	1	0	Internal clock: counts on PCLK/16
	0	1	1	Internal clock: counts on PCLK/64
	1	0	0	External clock: counts on TCLKA pin input
	1	0	1	Internal clock: counts on PCLK/1024
	1	1	0	Internal clock: counts on PCLK/256
	1	1	1	Internal clock: counts on PCLK/4096

Table 25.8 Bits TPSC[2:0] (TPU4)

Channel	Bits TPSC[2:0]			Description
	b2	b1	b0	
TPU4	0	0	0	Internal clock: counts on PCLK/1
	0	0	1	Internal clock: counts on PCLK/4
	0	1	0	Internal clock: counts on PCLK/16
	0	1	1	Internal clock: counts on PCLK/64
	1	0	0	External clock: counts on TCLKA pin input
	1	0	1	External clock: counts on TCLKC pin input
	1	1	0	Internal clock: counts on PCLK/1024
	1	1	1	Counts on TPU5.TCNT overflow/underflow

Note: This setting is invalid when TPU4 is in phase counting mode.

Table 25.9 Bits TPSC[2:0] (TPU5)

Channel	Bits TPSC[2:0]			Description
	b2	b1	b0	
TPU5	0	0	0	Internal clock: counts on PCLK/1
	0	0	1	Internal clock: counts on PCLK/4
	0	1	0	Internal clock: counts on PCLK/16
	0	1	1	Internal clock: counts on PCLK/64
	1	0	0	External clock: counts on TCLKA pin input
	1	0	1	External clock: counts on TCLKC pin input
	1	1	0	Internal clock: counts on PCLK/256
	1	1	1	External clock: counts on TCLKD pin input

Note: This setting is invalid when TPU5 is in phase counting mode.

Downloaded from Arrow.com.

Table 25.10 Bits CKEG[1:0]

Bits CKEG[1:0]		Input Clock	
b4	b3	Internal Clock	External clock
0	0	Counted at falling edge	Counted at rising edge
0	1	Counted at rising edge	Counted at falling edge
1	0	Counted at both edges	Counted at both edges
1	1	Counted at both edges	Counted at both edges

Table 25.11 Bits CCLR[2:0] (TPU0, TPU3)

Channel	Bits CCLR[2:0]			Description
	b7	b6	b5	
TPU0, TPU3	0	0	0	TCNT clearing disabled
	0	0	1	TCNT cleared by TGRA compare match/input capture
	0	1	0	TCNT cleared by TGRB compare match/input capture
	0	1	1	TCNT cleared by counter clearing for another channel performing synchronous clearing/ synchronous operation*2
	1	0	0	TCNT clearing disabled
	1	0	1	TCNT cleared by TGRC compare match/input capture*1
	1	1	0	TCNT cleared by TGRD compare match/input capture*1
	1	1	1	TCNT cleared by counter clearing for another channel performing synchronous clearing/ synchronous operation*2

Note 1. When TGRC or TGRD is used as a buffer register, TCNT is not cleared because the buffer register setting has priority, and compare match/input capture does not occur.
Note 2. Synchronous operation is selected by setting the TPU.TSYR.SYNCj bit (j = 0, 3) to 1.

Table 25.12 Bits CCLR[2:0] (TPU1, TPU2, TPU4, TPU5)

Channel	Bits CCLR[2:0]			Description
	b7*1	b6	b5	
TPU1, TPU2, TPU4, TPU5	0	0	0	TCNT clearing disabled
	0	0	1	TCNT cleared by TGRA compare match/input capture
	0	1	0	TCNT cleared by TGRB compare match/input capture
	0	1	1	TCNT cleared by counter clearing for another channel performing synchronous clearing/ synchronous operation*2
	1	0	0	Setting prohibited
	1	0	1	Setting prohibited
	1	1	0	Setting prohibited
	1	1	1	Setting prohibited

Note 1. This bit is reserved in TPU1, TPU2, TPU4, and TPU5. This bit is read as 0. The write value should be 0.
Note 2. Synchronous operation is selected by setting the TPU.TSYR.SYNCj bit (j = 1, 2, 4, 5) to 1.

Downloaded from Arrow.com.

25.2.2 Timer Mode Register (TMDR)

Address(es): TPU0.TMDR 0008 8111h, TPU1.TMDR 0008 8121h, TPU2.TMDR 0008 8131h, TPU3.TMDR 0008 8141h, TPU4.TMDR 0008 8151h, TPU5.TMDR 0008 8161h

	b7	b6	b5	b4	b3	b2	b1	b0
	ICSELD	ICSELB	BFB	BFA	MD[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	MD[3:0]	Mode Select	b3 b0 0 0 0 0: Normal operation 0 0 0 1: Setting prohibited 0 0 1 0: PWM mode 1 0 0 1 1: PWM mode 2 0 1 0 0: Phase counting mode 1*1 0 1 0 1: Phase counting mode 2*1 0 1 1 0: Phase counting mode 3*1 0 1 1 1: Phase counting mode 4*1 Settings other than above are prohibited.	R/W
b4	BFA	Buffer Operation A*2	0: TPUm.TGRA operates normally 1: TPUm.TGRA and TPUm.TGRC used together for buffer operation (m = 0, 3)	R/W
b5	BFB	Buffer Operation B*2	0: TPUm.TGRB operates normally 1: TPUm.TGRB and TPUm.TGRD used together for buffer operation (m = 0, 3)	R/W
b6	ICSELB	TGRB Input Capture Input Select*3	0: Input capture input source is TIOCBn pin 1: Input capture input source is TIOCAn pin (n = 3, 4)	R/W
b7	ICSELD	TGRD Input Capture Input Select*4	0: Input capture input source is TIOCDn pin 1: Input capture input source is TIOCCn pin (n = 3)	R/W

Note 1. Phase counting mode cannot be set for TPU0 and TPU3. A 0 should be written to bit 2 for them.
Note 2. These bits are reserved in TPU1, TPU2, TPU4, and TPU5. These bits are read as 0. The write value should be 0.
Note 3. In TPU0, TPU1, TPU2, and TPU5, which have no TIOCAn pin, these bits are reserved. These bits are read as 0. The write value should be 0.
Note 4. In TPU0, TPU1, TPU2, TPU4, and TPU5, which have no TIOCCn, TIOCDn pin, these bits are reserved. These bits are read as 0. The write value should be 0.

TPUm.TMDR settings should be made while TPUm.TCNT operation is stopped.

BFA Bit (Buffer Operation A)

Specifies whether TPUm.TGRA (m = 0, 3) is to normally operate, or TPUm.TGRA and TPUm.TGRC (m = 0, 3) are to be used together for buffer operation.
When TGRC is used as a buffer register, TGRC input capture/output compare is not generated.

BFB Bit (Buffer Operation B)

Specifies whether TPUm.TGRB (m = 0, 3) is to normally operate, or TPUm.TGRB and TPUm.TGRD (m = 0, 3) are to be used together for buffer operation.
When TGRD is used as a buffer register, TGRD input capture/output compare is not generated.

ICSELB Bit (TGRB Input Capture Input Select)

Selects the input capture input for TPUm.TGRB (m = 3, 4).
This function allows measurement of high-level width and period of the input pulse on a TIOCAn input pin.

Downloaded from Arrow.com.

ICSELD Bit (TGRD Input Capture Input Select)

Selects the input capture input for TPUm.TGRD (m = 3).

This function allows measurement of high-level width and period of the input pulse on a TIOCCn input pin.

25.2.3 Timer I/O Control Register (TIORH, TIORL, TIOR)

- TPU0.TIORH, TPU1.TIOR, TPU2.TIOR, TPU3.TIORH, TPU4.TIOR, TPU5.TIOR

Address(es): TPU0.TIORH 0008 8112h, TPU1.TIOR 0008 8122h, TPU2.TIOR 0008 8132h, TPU3.TIORH 0008 8142h, TPU4.TIOR 0008 8152h, TPU5.TIOR 0008 8162h

	b7	b6	b5	b4	b3	b2	b1	b0
	IOB[3:0]				IOA[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	IOA[3:0]	TGRA Control*2	See Table 25.13 to Table 25.18.*1	R/W
b7 to b4	IOB[3:0]	TGRB Control	See Table 25.13 to Table 25.18.*1	R/W

Note 1. If the IOn[3:0] bit (n = A, B) values are changed to output disabled (0000b or 0100b) during low/high/toggle output on compare match, the TIOCAn/TIOCBn pin (n = 0 to 5) is placed in high impedance state.

Note 2. In TPU0, TPU1, TPU2, and TPU5, which have no TIOCAn pin, these bits are reserved. These bits are read as 0. The write value should be 0.

- TPU3.TIORL

Address(es): TPU3.TIORL 0008 8143h

	b7	b6	b5	b4	b3	b2	b1	b0
	IOD[3:0]				IOC[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	IOC[3:0]	TGRC Control	See Table 25.16.*1	R/W
b7 to b4	IOD[3:0]	TGRD Control	See Table 25.16.*1	R/W

Note 1. If the IOn[3:0] bit (n = C, D) values are changed to output disabled (0000b or 0100b) during low/high/toggle output on compare match, the TIOCCn/TIOCDn pin (n = 0, 3) is placed in high impedance state.

Downloaded from Arrow.com.

TPU has two TIORH registers, one for TPU0 and TPU3, and one TIORL register for TPU3, and also has four TIOR registers, one for TPU1, TPU2, TPU4, and TPU5. Thus the TPU has seven timer I/O control registers in total.
TIORH, TIORL, and TIOR control registers TGRA, TGRB, TGRC, and TGRD.
Note that TIORH, TIORL, and TIOR are affected by the TMDR setting. For details, see Table 25.13 to Table 25.19.
The initial output specified by TIORH, TIORL, and TIOR is valid when the counter is stopped (the TPU.TSTR.CSTj bit (j = 0 to 5) is cleared to 0). In PWM mode 2, the output at the time when the TCNT is cleared to 0 is specified as the initial output.
When buffer operation has been selected for register TGRC or TGRD, the settings of the IOC[3:0] or IOD[3:0] bits become ineffective, and the TGRC or TGRD register simply operates as a buffer.
To specify the input capture pin in TIORH, TIORL, or TIOR, set the bit in the port direction register (PDR) for the corresponding pin to 0 (input port), and set the bit in the port mode register (PMR) to 1 (uses the pin as an I/O port for peripheral functions). For details, see section 21, I/O Ports.

IOA[3:0] Bits (TGRA Control)

Select the function of TPUm.TGRA (m = 3, 4).

IOB[3:0] Bits (TGRB Control)

Select the function of TPUm.TGRB (m = 0 to 5).

IOC[3:0] Bits (TGRC Control)

Select the function of TPUm.TGRC (m = 3).

IOD[3:0] Bits (TGRD Control)

Select the function of TPUm.TGRD (m = 3).

Downloaded from Arrow.com.

Table 25.13 TPU0.TIOR

Bits IOB[3:0]				Description	
b7	b6	b5	b4	TPU0.TGRB Function	TIOCB0 Pin Function and Related Issue
0	0	0	0	Output compare register	Output disabled
0	0	0	1		Initial output is low output; low output at compare match
0	0	1	0		Initial output is low output; high output at compare match
0	0	1	1		Initial output is low output; toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is high output; low output at compare match
0	1	1	0		Initial output is high output; high output at compare match
0	1	1	1		Initial output is high output; toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCB0 pin; input capture at rising edge
1	0	0	1		Capture input source is TIOCB0 pin; input capture at falling edge
1	0	1	x		Capture input source is TIOCB0 pin; input capture at both edges
1	1	x	x		Capture input source is TPU1 count clock; input capture at TPU1.TCNT count-up/ count-down*1

x: Don't care

Note 1. When the TPSC[2:0] bits in TPU1.TCR are set to 000b and PCLK/1 is used as the TPU1.TCNT count clock, this setting is invalid and input capture is not generated.

Table 25.14 TPU1.TIOR

Bits IOB[3:0]				Description	
b7	b6	b5	b4	TPU1.TGRB Function	TIOCB1 Pin Function and Related Issue
0	0	0	0	Output compare register	Output disabled
0	0	0	1		Initial output is low output; low output at compare match
0	0	1	0		Initial output is low output; high output at compare match
0	0	1	1		Initial output is low output; toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is high output; low output at compare match
0	1	1	0		Initial output is high output; high output at compare match
0	1	1	1		Initial output is high output; toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCB1 pin; input capture at rising edge
1	0	0	1		Capture input source is TIOCB1 pin; input capture at falling edge
1	0	1	x		Capture input source is TIOCB1 pin; input capture at both edges
1	1	x	x		Capture input source is TPU0.TGRC compare match/input capture; input capture at generation of TPU0.TGRC compare match/input capture

x: Don't care

Downloaded from Arrow.com.

Table 25.15 TPU2.TIOR

Bits IOB[3:0]				Description	
b7	b6	b5	b4	TPU2.TGRB Function	TIOCB2 Pin (Function and Related Issue
0	0	0	0	Output compare register	Output disabled
0	0	0	1		Initial output is low output; low output at compare match
0	0	1	0		Initial output is low output; high output at compare match
0	0	1	1		Initial output is low output; toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is high output; low output at compare match
0	1	1	0		Initial output is high output; high output at compare match
0	1	1	1		Initial output is high output; toggle output at compare match
1	x	0	0	Input capture register	Capture input source is TIOCB2 pin; input capture at rising edge
1	x	0	1		Capture input source is TIOCB2 pin; input capture at falling edge
1	x	1	x		Capture input source is TIOCB2 pin; input capture at both edges

x: Don’t care

Downloaded from Arrow.com.

Table 25.16 TPU3.TIORH

Bits IOA[3:0]				Description	
b3	b2	b1	b0	TPU3.TGRA Function	TIOCA3 Pin Function and Related Issue
0	0	0	0	Output compare register	Output disabled
0	0	0	1		Initial output is low output; low output at compare match
0	0	1	0		Initial output is low output; high output at compare match
0	0	1	1		Initial output is low output; toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is high output; low output at compare match
0	1	1	0		Initial output is high output; high output at compare match
0	1	1	1		Initial output is high output; toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCA3 pin; input capture at rising edge
1	0	0	1		Capture input source is TIOCA3 pin; input capture at falling edge
1	0	1	x		Capture input source is TIOCA3 pin; input capture at both edges
1	1	x	x		Capture input source is TPU4 count clock; input capture at TPU4.TCNT count-up/ count-down*1

Bits IOB[3:0]				Description	
b7	b6	b5	b4	TPU3.TGRB Function	TIOCB3 Pin Function and Related Issue
0	0	0	0	Output compare register	Output disabled
0	0	0	1		Initial output is low output; low output at compare match
0	0	1	0		Initial output is low output; high output at compare match
0	0	1	1		Initial output is low output; toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is high output; low output at compare match
0	1	1	0		Initial output is high output; high output at compare match
0	1	1	1		Initial output is high output; toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCB3 or TIOCA3 pin*2; input capture at rising edge
1	0	0	1		Capture input source is TIOCB3 or TIOCA3 pin*2; input capture at falling edge
1	0	1	x		Capture input source is TIOCB3 or TIOCA3 pin*2; input capture at both edges
1	1	x	x		Capture input source is TPU4 count clock; input capture at TPU4.TCNT count-up/ count-down*1

x: Don’t care

Note 1. When the TPSC[2:0] bits in TPU4.TCR are set to 000b and PCLK/1 is used as the TPU4.TCNT count clock, this setting is invalid and input capture is not generated.

Note 2. Selected by the ICSELB bit in TPU3.TMDR.

Downloaded from Arrow.com.

Table 25.17 TPU4.TIOR

Bits IOA[3:0]				Description	
b3	b2	b1	b0	TPU4.TGRA Function	TIOCA4 Pin Function and Related Issue
0	0	0	0	Output compare register	Output disabled
0	0	0	1		Initial output is low output; low output at compare match
0	0	1	0		Initial output is low output; high output at compare match
0	0	1	1		Initial output is low output; toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is high output; low output at compare match
0	1	1	0		Initial output is high output; high output at compare match
0	1	1	1		Initial output is high output; toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCA4 pin; input capture at rising edge
1	0	0	1		Capture input source is TIOCA4 pin; input capture at falling edge
1	0	1	x		Capture input source is TIOCA4 pin; input capture at both edges
1	1	x	x		Capture input source is TPU3.TGRA compare match/input capture; input capture at generation of TPU3.TGRA compare match/input capture

Bits IOB[3:0]				Description	
b7	b6	b5	b4	TPU4.TGRB Function	TIOCB4 Pin Function and Related Issue
0	0	0	0	Output compare register	Output disabled
0	0	0	1		Initial output is low output; low output at compare match
0	0	1	0		Initial output is low output; high output at compare match
0	0	1	1		Initial output is low output; toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is high output; low output at compare match
0	1	1	0		Initial output is high output; high output at compare match
0	1	1	1		Initial output is high output; toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCB4 or TIOCA4 pin*1; input capture at rising edge
1	0	0	1		Capture input source is TIOCB4 or TIOCA4 pin*1; input capture at falling edge
1	0	1	x		Capture input source is TIOCB4 or TIOCA4 pin*1; input capture at both edges
1	1	x	x		Capture input source is TPU3.TGRC compare match/input capture; input capture at generation of TPU3.TGRC compare match/input capture

x: Don’t care

Note 1. Selected by the ICSELB bit in TPU4.TMDR.

Downloaded from Arrow.com.

Table 25.18 TPU5.TIOR

Bits IOB[3:0]				Description	
b7	b6	b5	b4	TPU5.TGRB Function	TIOCB5 Pin Function and Related Issue
0	0	0	0	Output compare register	Output disabled
0	0	0	1		Initial output is low output; low output at compare match
0	0	1	0		Initial output is low output; high output at compare match
0	0	1	1		Initial output is low output; toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is high output; low output at compare match
0	1	1	0		Initial output is high output; high output at compare match
0	1	1	1		Initial output is high output; toggle output at compare match
1	x	0	0	Input capture register	Capture input source is TIOCB5 input capture at rising edge
1	x	0	1		Capture input source is TIOCB5 input capture at falling edge
1	x	1	x		Capture input source is TIOCB5 input capture at both edges

x: Don’t care

Downloaded from Arrow.com.

Table 25.19 TPU3.TIORL

Bits IOC[3:0]				Description	
b3	b2	b1	b0	TPU3.TGRC Function	TIOCC3 Pin Function and Related Issue
0	0	0	0	Output compare register*1	Output disabled
0	0	0	1		Initial output is low output; low output at compare match
0	0	1	0		Initial output is low output; high output at compare match
0	0	1	1		Initial output is low output; toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is high output; low output at compare match
0	1	1	0		Initial output is high output; high output at compare match
0	1	1	1		Initial output is high output; toggle output at compare match
1	0	0	0	Input capture register*1	Capture input source is TIOCC3 pin; input capture at rising edge
1	0	0	1		Capture input source is TIOCC3 pin; input capture at falling edge
1	0	1	x		Capture input source is TIOCC3 pin; input capture at both edges
1	1	x	x		Capture input source is TPU4 count clock; input capture at TPU4.TCNT count-up/ count-down*3

Bits IOD[3:0]				Description	
b7	b6	b5	b4	TPU3.TGRD Function	TIOCD3 Pin Function and Related Issue
0	0	0	0	Output compare register*2	Output disabled
0	0	0	1		Initial output is low output; low output at compare match
0	0	1	0		Initial output is low output; high output at compare match
0	0	1	1		Initial output is low output; toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is high output; low output at compare match
0	1	1	0		Initial output is high output; high output at compare match
0	1	1	1		Initial output is high output; toggle output at compare match
1	0	0	0	Input capture register*2	Capture input source is TIOCD3 or TIOCC3 pin*4; input capture at rising edge
1	0	0	1		Capture input source is TIOCD3 or TIOCC3 pin*4; input capture at falling edge
1	0	1	x		Capture input source is TIOCD3 or TIOCC3 pin*4; input capture at both edges
1	1	X	x		Capture input source is TPU4 count clock; input capture at TPU4.TCNT count-up/ count-down*3

x: Don’t care

Note 1. When the BFA bit in TPU3.TMDR is set to 1 (TPU3.TGRA and TPU3.TGRC are used for buffer operation) and TPU3.TGRC is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Note 2. When the BFB bit in TPU3.TMDR is set to 1 (TPU3.TGRB and TPU3.TGRD are used for buffer operation) and TPU3.TGRD is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Note 3. When the TPSC[2:0] bits in TPU4.TCR are set to 000b and PCLK/1 is used as the TPU3.TCNT count clock, this setting is invalid and input capture is not generated.

Note 4. Selected by the ICSELD bit in TPU3.TMDR.

Downloaded from Arrow.com.

25.2.4 Timer Interrupt Enable Register (TIER)

Address(es): TPU0.TIER 0008 8114h, TPU1.TIER 0008 8124h, TPU2.TIER 0008 8134h, TPU3.TIER 0008 8144h, TPU4.TIER 0008 8154h, TPU5.TIER 0008 8164h

	b7	b6	b5	b4	b3	b2	b1	b0
	TTGE	—	TCIEU	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
Value after reset:	0	1	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TGIEA	TGRA Interrupt Enable	0: Interrupt requests (TGImA) disabled 1: Interrupt requests (TGImA) enabled (m = 0 to 5)	R/W
b1	TGIEB	TGRB Interrupt Enable	0: Interrupt requests (TGImB) disabled 1: Interrupt requests (TGImB) enabled (m = 0 to 5)	R/W
b2	TGIEC	TGRC Interrupt Enable*1	0: Interrupt requests (TGImC) disabled 1: Interrupt requests (TGImC) enabled (m = 0, 3)	R/W
b3	TGIED	TGRD Interrupt Enable*1	0: Interrupt requests (TGImD) disabled 1: Interrupt requests (TGImD) enabled (m = 0, 3)	R/W
b4	TCIEV	Overflow Interrupt Enable	0: Interrupt requests (TCImV) disabled 1: Interrupt requests (TCImV) enabled (m = 0 to 5)	R/W
b5	TCIEU	Underflow Interrupt Enable*2	0: Interrupt requests (TCImU) disabled 1: Interrupt requests (TCImU) enabled (m = 1, 2, 4, 5)	R/W
b6	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b7	TTGE	A/D Conversion Start Request Enable*3	0: A/D conversion start request generation disabled 1: A/D conversion start request generation enabled	R/W

Note 1. These bits are reserved in TPU1, TPU2, TPU4, and TPU5. These bits are read as 0. The write value should be 0.
Note 2. This bit is reserved in TPU0 and TPU3. This bit is read as 0. The write value should be 0.
Note 3. This bit is reserved in TPU5. This bit is read as 0. The write value should be 0.

TTGE Bit (A/D Conversion Start Request Enable)

Enables/disables generation of A/D conversion start requests by TPUm.TGRA (m = 0 to 4) input capture/compare match.

Downloaded from Arrow.com.

25.2.5 Timer Status Register (TSR)

Address(es): TPU0.TSR 0008 8115h, TPU1.TSR 0008 8125h, TPU2.TSR 0008 8135h, TPU3.TSR 0008 8145h, TPU4.TSR 0008 8155h, TPU5.TSR 0008 8165h

	b7	b6	b5	b4	b3	b2	b1	b0
	TCFD	—	TCFU	TCFV	TGFD	TGFC	TGFB	TGFA
Value after reset:	1	1	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TGFA	Input Capture/Output Compare Flag A	0: Input capture to TPUm.TGRA or compare match with TPUm.TGRA has not occurred. 1: Input capture to TPUm.TGRA or compare match with TPUm.TGRA has occurred. (m = 0 to 5)	R/W*1
b1	TGFB	Input Capture/Output Compare Flag B	0: Input capture to TPUm.TGRB or compare match with TPUm.TGRB has not occurred. 1: Input capture to TPUm.TGRB or compare match with TPUm.TGRB has occurred. (m = 0 to 5)	R/W*1
b2	TGFC	Input Capture/Output Compare Flag C*2	0: Input capture to TPUm.TGRC or compare match with TPUm.TGRC has not occurred. 1: Input capture to TPUm.TGRC or compare match with TPUm.TGRC has occurred. (m = 0, 3)	R/W*1
b3	TGFD	Input Capture/Output Compare Flag D*2	0: Input capture to TPUm.TGRD or compare match with TPUm.TGRD has not occurred. 1: Input capture to TPUm.TGRD or compare match with TPUm.TGRD has occurred. (m = 0, 3)	R/W*1
b4	TCFV	Overflow Flag	0: TPUm.TCNT has not overflowed. 1: TPUm.TCNT has overflowed. (m = 0 to 5)	R/W*1
b5	TCFU	Underflow Flag*3	0: TPUm.TCNT has not underflowed. 1: TPUm.TCNT has underflowed. (m = 1, 2, 4, 5)	R/W*1
b6	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b7	TCFD	Counting Direction Flag*4	0: TPUm.TCNT counts down. 1: TPUm.TCNT counts up. (m = 1, 2, 4, 5)	R

Note 1. Only writing 0 to this bit is possible; this clears the flag.
Note 2. These bits are reserved in TPU1, TPU2, TPU4, and TPU5. The bits are read as 0. The write value should be 0.
Note 3. This bit is reserved in TPU0 and TPU3. The bit is read as 0. The write value should be 0.
Note 4. This bit is reserved in TPU0 and TPU3. The bit is read as 1. The write value should be 1.

Downloaded from Arrow.com.

TGFA Flag (Input Capture/Output Compare Flag A)

This status flag indicates that input capture to TPUm.TGRA or compare match with TPUm.TGRA (m = 0 to 5) has occurred.

[Setting conditions]

- When TPUm.TGRA holds the value for comparison in output-compare operations, TPUm.TCNT matches TPUm.TGRA.
- When TPUm.TGRA is serving as an input-capture register, the input-capture signal has caused transfer of the value in TPUm.TCNT to TPUm.TGRA.

[Clearing conditions]

- Activation of the DTC by the TGImA interrupt and clearing of the DTC.MRB.DISEL bit.
- Writing 0 to TGFA after reading its value as 1.

TGFB Flag (Input Capture/Output Compare Flag B)

This status flag indicates that input capture to TPUm.TGRB or compare match with TPUm.TGRB (m = 0 to 5) has occurred.

[Setting conditions]

- When TPUm.TGRB holds the value for comparison in output-compare operations, TPUm.TCNT matches TPUm.TGRB.
- When TPUm.TGRB is serving as an input-capture register, the input-capture signal has caused transfer of the value in TPUm.TCNT to TPUm.TGRB.

[Clearing conditions]

- Activation of the DTC by the TGImB interrupt and clearing of the DTC.MRB.DISEL bit.
- Writing 0 to TGFB after reading its value as 1.

TGFC Flag (Input Capture/Output Compare Flag C)

This status flag indicates that input capture to TPUm.TGRC or compare match with TPUm.TGRC (m = 0, 3) has occurred.

[Setting conditions]

- When TPUm.TGRC holds the value for comparison in output-compare operations, TPUm.TCNT matches TPUm.TGRC.
- When TPUm.TGRC is serving as an input-capture register, the input-capture signal has caused transfer of the value in TPUm.TCNT to TPUm.TGRC.

[Clearing conditions]

- Activation of the DTC by the TGImC interrupt and clearing of the DTC.MRB.DISEL bit.
- Writing 0 to TGFC after reading its value as 1.

TGFD Flag (Input Capture/Output Compare Flag D)

This status flag indicates that input capture to TPUm.TGRD or compare match with TPUm.TGRD (m = 0, 3) has occurred.

[Setting conditions]

- When TPUm.TGRD holds the value for comparison in output-compare operations, TPUm.TCNT matches TPUm.TGRD.
- When TPUm.TGRD is serving as an input-capture register, the input-capture signal has caused transfer of the value in TPUm.TCNT to TPUm.TGRD.

[Clearing conditions]

- Activation of the DTC by the TGImD interrupt and clearing of the DTC.MRB.DISEL bit.
- Writing 0 to TGFD after reading its value as 1.

Downloaded from Arrow.com.

TCFV Flag (Overflow Flag)

This status flag indicates an overflow of TPUm.TCNT (m = 0 to 5).

[Setting condition]

- Overflow of the value in TPUm.TCNT (TCNT counted from FFFFh to 0000h).

[Clearing condition]

- Writing 0 to TCFV after reading its value as 1.

TCFU Flag (Underflow Flag)

This status flag indicates an underflow of TPUm.TCNT (m = 1, 2, 4, 5).

[Setting condition]

- Underflow of the value in TPUm.TCNT (TCNT counted from 0000h to FFFFh).

[Clearing condition]

- Writing 0 to TCFU after reading its value as 1.

Downloaded from Arrow.com.

25.2.6 Timer Counter (TCNT)

Address(es): TPU0.TCNT 0008 8116h, TPU1.TCNT 0008 8126h, TPU2.TCNT 0008 8136h, TPU3.TCNT 0008 8146h, TPU4.TCNT 0008 8156h, TPU5.TCNT 0008 8166h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TPUm.TCNT is a readable/writable counter that counts the internal clock or external events.

25.2.7 Timer General Register A (TGRA), Timer General Register B (TGRB), Timer General Register C (TGRC), Timer General Register D (TGRD)

Address(es): TPU0.TGRA 0008 8118h, TPU0.TGRB 0008 811Ah, TPU0.TGRC 0008 811Ch, TPU0.TGRD 0008 811Eh,
TPU1.TGRA 0008 8128h, TPU1.TGRB 0008 812Ah,
TPU2.TGRA 0008 8138h, TPU2.TGRB 0008 813Ah,
TPU3.TGRA 0008 8148h, TPU3.TGRB 0008 814Ah, TPU3.TGRC 0008 814Ch, TPU3.TGRD 0008 814Eh,
TPU4.TGRA 0008 8158h, TPU4.TGRB 0008 815Ah,
TPU5.TGRA 0008 8168h, TPU5.TGRB 0008 816Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

TPU has 16 TGR registers in total, four each for TPU0 and TPU3, and two each for TPU1, TPU2, TPU4, and TPU5. TPUm.TGRA (m = 0 to 5), TPUm.TGRB (m = 0 to 5), TPUm.TGRC (m = 0, 3), and TPUm.TGRD (m = 0, 3) are readable/writable registers with a dual function as output compare and input capture registers.
TPUm.TGRC and TPUm.TGRD can also be specified for operation as buffer registers. Register combinations during buffer operations are TPUm.TGRA—TPUm.TGRC and TPUm.TGRB—TPUm.TGRD.

Downloaded from Arrow.com.

25.2.8 Timer Start Register (TSTR)

Address(es): TPU.TSTR 0008 8100h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	CST5	CST4	CST3	CST2	CST1	CST0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CST0	Counter Start 0	0: TCNT count operation is stopped 1: TCNT performs count operation	R/W
b1	CST1	Counter Start 1		R/W
b2	CST2	Counter Start 2		R/W
b3	CST3	Counter Start 3		R/W
b4	CST4	Counter Start 4		R/W
b5	CST5	Counter Start 5		R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

TPU.TSTR starts or stops TCNT operation for TPU0 to TPU5.
Before setting the operating mode in TPUm.TMDR or setting the TPUm.TCNT count clock in TPUm.TCR, stop the TPUm.TCNT operation.

CSTn Bit (Counter Start n) (n = 0 to 5)

This bit starts or stop the TCNT.
When the CSTn bit is cleared to 0 with CSTn = 1 and the corresponding TIOCyn pin (y = A to D; n = 0 to 5) specified for output, count operation stops but the output compare output level of the corresponding TIOCyn pin is retained.
If TIORH, TIORL, or TIOR is written to when the CSTn bit is 0, the pin output level will be changed to the set initial output value.

Downloaded from Arrow.com.

25.2.9 Timer Synchronous Register (TSYR)

Address(es): TPU.TSYR 0008 8101h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	SYNC5	SYNC4	SYNC3	SYNC2	SYNC1	SYNC0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SYNC0	Timer Synchronization 0	0: TCNT operates independently (TCNT setting/clearing is unrelated to other channels) 1: TCNT performs synchronous operation*1 (TCNT synchronous setting/synchronous clearing is possible)	R/W
b1	SYNC1	Timer Synchronization 1		R/W
b2	SYNC2	Timer Synchronization 2		R/W
b3	SYNC3	Timer Synchronization 3		R/W
b4	SYNC4	Timer Synchronization 4		R/W
b5	SYNC5	Timer Synchronization 5		R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. To set synchronous operation, the SYNCn bit (n = 0 to 5) for at least two channels must be set to 1. To set synchronous clearing, the TCNT clearing source must also be set by the TCR.CCLR[2:0] bits in addition to the SYNCn bit.

TPU.TSYR selects independent operation or synchronous operation for TCNT of TPU0 to TPU5.

SYNCn Bit (Timer Synchronization n) (n = 0 to 5)

This bit selects whether the TCNT operation is independent of or synchronized with TCNT of other channels. When synchronous operation is selected, synchronous setting of multiple TCNT and synchronous clearing through counter clearing on another channel are possible.

Downloaded from Arrow.com.

25.2.10 Noise Filter Control Register (NFCR)

Address(es): TPU0.NFCR 0008 8108h, TPU1.NFCR 0008 8109h, TPU2.NFCR 0008 810Ah, TPU3.NFCR 0008 810Bh, TPU4.NFCR 0008 810Ch, TPU5.NFCR 0008 810Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	NFCS[1:0]		NFDEN	NFCEN	NFBEN	NFAEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	NFAEN	Noise Filter Enable A*2	0: The noise filter for TIOCAm is disabled. 1: The noise filter for TIOCAm is enabled. (m = 3, 4)	R/W
b1	NFBEN	Noise Filter Enable B	0: The noise filter for TIOCBm is disabled. 1: The noise filter for TIOCBm is enabled. (m = 0 to 5)	R/W
b2	NFCEN	Noise Filter Enable C*1	0: The noise filter for TIOCCm is disabled. 1: The noise filter for TIOCCm is enabled. (m = 3)	R/W
b3	NFDEN	Noise Filter Enable D*1	0: The noise filter for TIOCDm is disabled. 1: The noise filter for TIOCDm is enabled. (m = 3)	R/W
b5, b4	NFCS[1:0]	Noise Filter Clock Select	b5 b4 0 0: PCLK/1 0 1: PCLK/8 1 0: PCLK/32 1 1: Clock source that drives counting	R/W
b7, b6	—	Reserved	These bits are read as 0. Writing to these bits is not possible.	R

Note 1. These bits are reserved in TPU0, TPU1, TPU2, TPU4, and TPU5. The bits are read as 0. Writing to these bits is not possible.
Note 2. These bits are reserved in TPU0, TPU1, TPU2, and TPU5. The bits are read as 0. Writing to these bits is not possible.

Only set the TPUm.NFCR registers while the TPUm.TCNT is stopped.

NFAEN Bit (Noise Filter Enable A)

This bit disables or enables the noise filter for the TIOCAm pin (m = 3, 4).
Since unexpected edges may be internally generated when the value of NFAEN is changed, select the output compare function in the timer I/O control register before changing the NFAEN value.

NFBEN Bit (Noise Filter Enable B)

This bit disables or enables the noise filter for the TIOCBm pin (m = 0 to 5).
Since unexpected edges may be internally generated when the value of NFBEN is changed, select the output compare function in the timer I/O control register before changing the NFBEN value.

NFCEN Bit (Noise Filter Enable C)

This bit disables or enables the noise filter for the TIOCCm pin (m = 3).
Since unexpected edges may be internally generated when the value of NFCEN is changed, select the output compare function in the timer I/O control register before changing the NFCEN value.

Downloaded from Arrow.com.

NFDEN Bit (Noise Filter Enable D)

This bit disables or enables the noise filter for the TIOCDm pin (m = 3).
Since unexpected edges may be internally generated when the value of NFDEN is changed, select the output compare function in the timer I/O control register before changing the NFDEN value.

NFCS[1:0] Bits (Noise Filter Clock Select)

These bits select the sampling clock for the noise filter.
When the count source is selected with NFCS[1:0] bits set to 11b, the clock that can be used as sampling clock are the internal clocks other than PCLK/1 specified with the TPSC[2:0] bits and the external clock. To select the PCLK/1 as both the count clock and the sampling clock, set the NFCS[1:0] bits to 00b.
The input-capture signal is sampled on rising edges of the selected clock signal. If the sampled levels match three times in a row, the given level is passed through as the input-capture signal. If the levels do not match, the existing value is retained.
After setting the NFCS[1:0] bits, wait for two selected sampling periods before setting the input capture function.

Downloaded from Arrow.com.

25.3 Operation

25.3.1 Basic Functions

Each channel has a TPUm.TCNT and a TPUm.TGRy register (y = A to D).
TCNT is a 16-bit up-counter, which can function as a free-running counter, periodic counter, or event counter.
TGRy can be used as an input capture register or output compare register.

(1) Counter Operation

When the CSTj bit (j = 0 to 5) in TPU.TSTR is set to 1, the TCNT for the corresponding channel starts counting.

(a) Example of count operation setting procedure

Figure 25.2 shows an example of the count operation setting procedure.

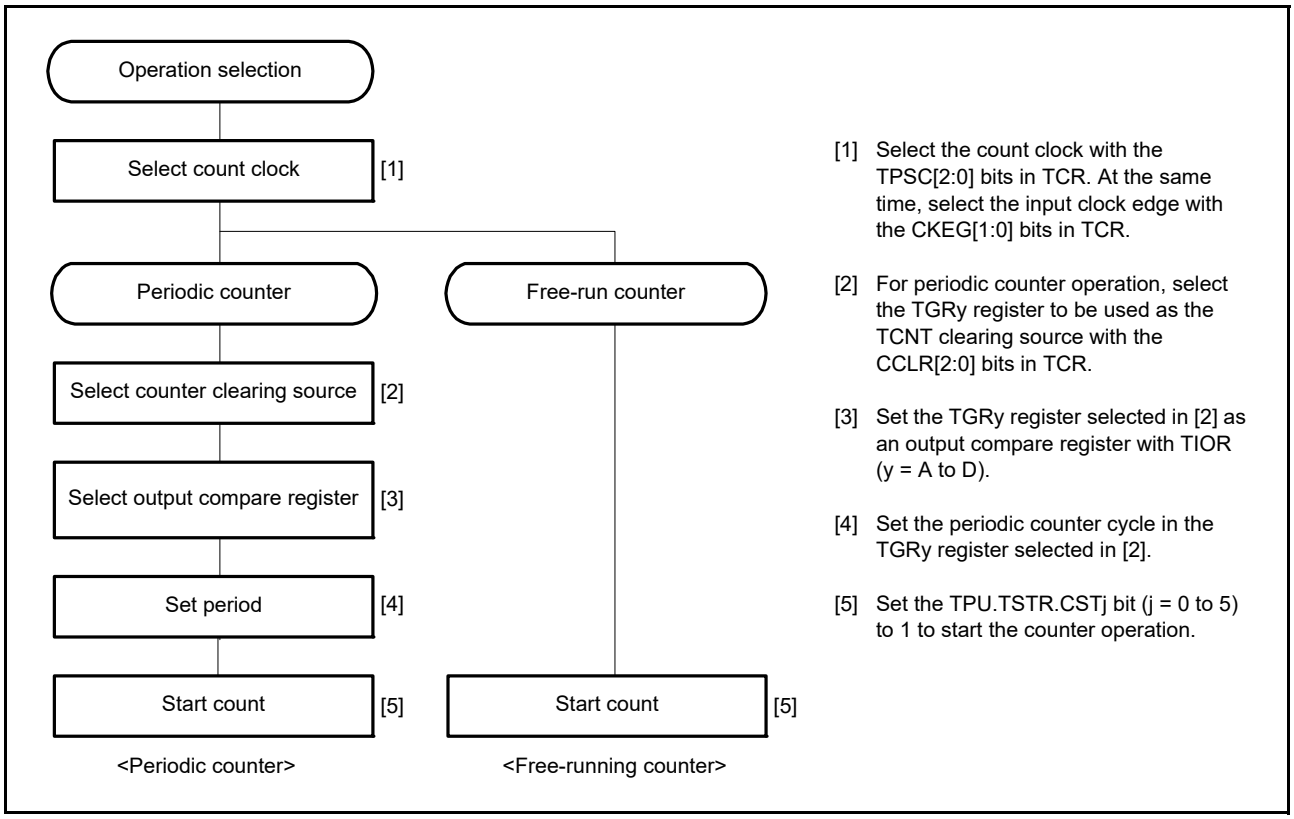


Figure 25.2 Example of Counter Operation Setting Procedure

Downloaded from Arrow.com.

(b) Free-running count operation and periodic count operation

Immediately after a reset, TPUm.TCNT are all set as free-running counters. When the relevant bit in TPU.TSTR is set to 1, the corresponding TCNT starts up-count operation as a free-running counter. When TCNT overflows (changes from FFFFh to 0000h), the TPU requests an interrupt. After an overflow, TCNT restarts counting up from 0000h.
Figure 25.3 shows free-running counter operation.

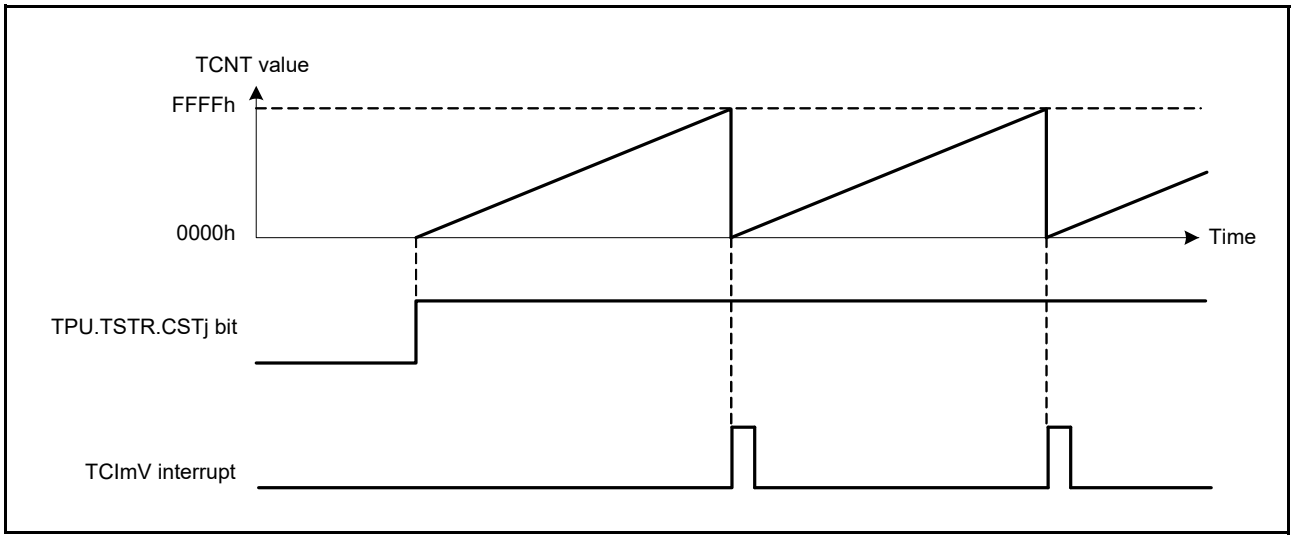


Figure 25.3 Free-Running Counter Operation

When compare match is selected as the TCNT clearing source, the TCNT for the relevant channel performs periodic count operation. The TPUm.TGRy for setting the period is set as an output compare register, and counter clearing by compare match is selected by the TPUm.TCR.CCLR[2:0] bits. After the settings have been made, TCNT starts count-up operation as a periodic counter when the corresponding bit in TPU.TSTR is set to 1. When the count value matches the TGRy value, TCNT is cleared to 0000h.
At this time, the TPU requests an interrupt. After a compare match, TCNT restarts counting up from 0000h.
Figure 25.4 shows periodic counter operation.

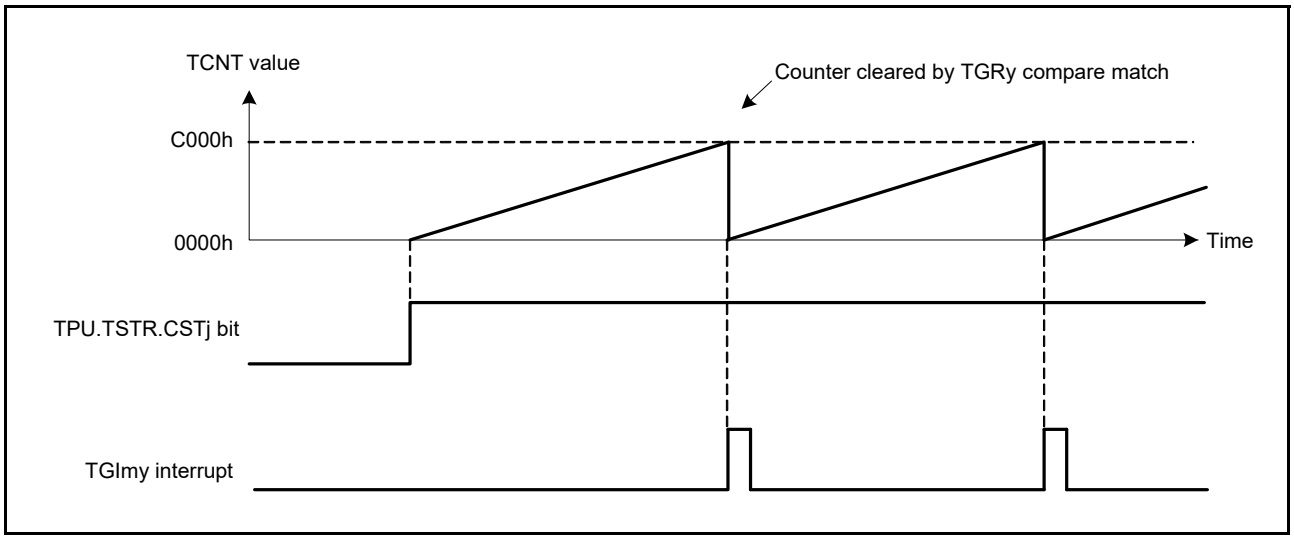


Figure 25.4 Periodic Counter Operation

Downloaded from Arrow.com.

(2) Waveform Output by Compare Match

The TPU can perform low, high, or toggle output from the corresponding output pin using a compare match.

(a) Example of setting procedure for waveform output by compare match

Figure 25.5 shows an example of the setting procedure for waveform output by a compare match.

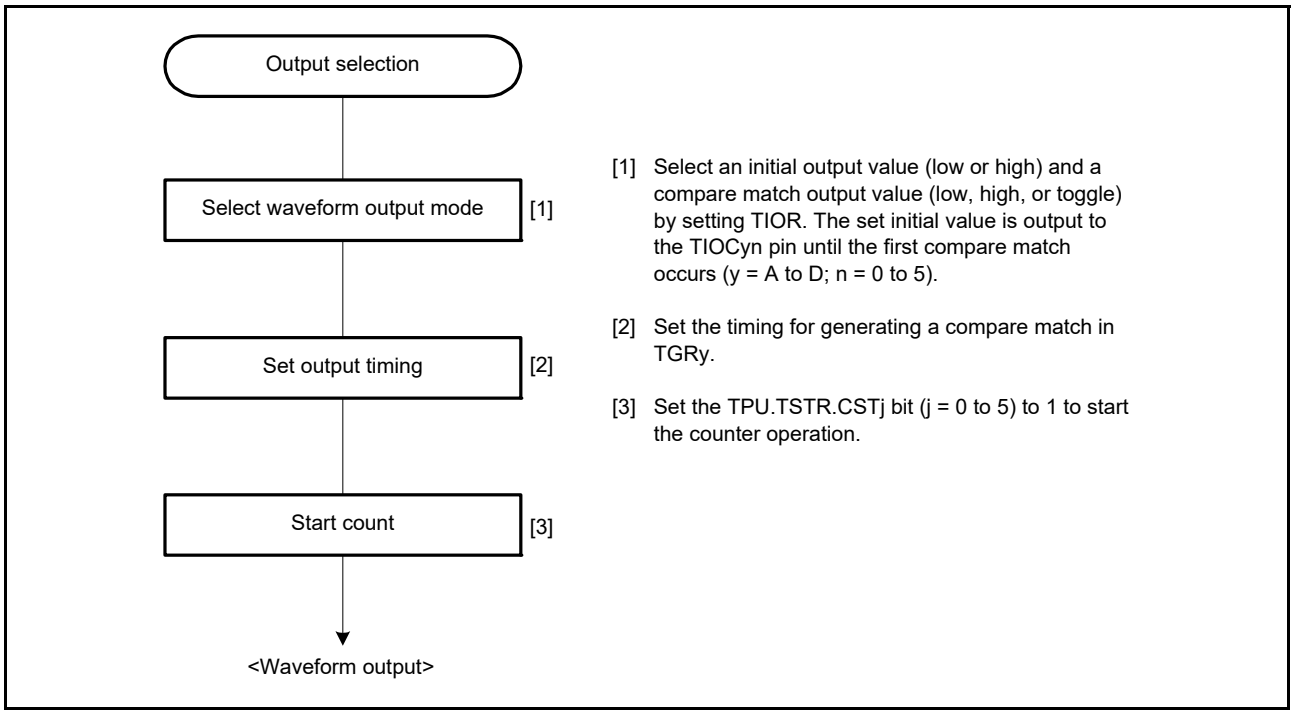


Figure 25.5 Example of Setting Procedure for Waveform Output by Compare Match

(b) Examples of waveform output operation

Figure 25.6 shows an example of low output/high output.

In this example, TPUm.TCNT has been set as a free-running counter, and settings have been made so that high is output by compare match A and low is output by compare match B. When the set level and the pin level match, the pin level does not change.

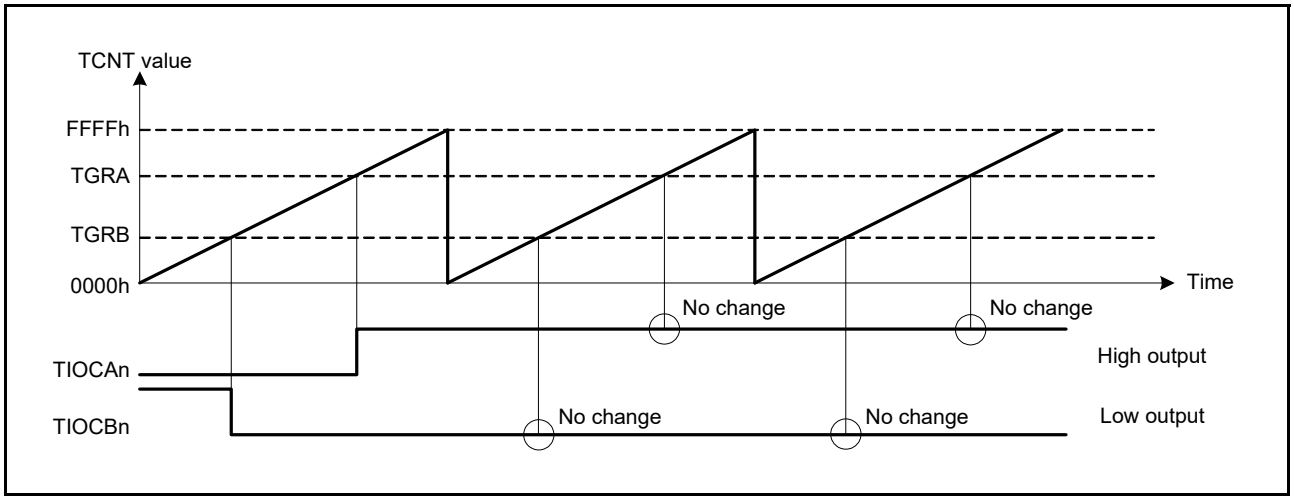


Figure 25.6 Example of Low-Output/High-Output Operation (n = 3, 4)

Downloaded from Arrow.com.

Figure 25.7 shows an example of toggle output.
In this example, TPUm.TCNT has been set as a periodic counter (with counter clearing performed by compare match B), and settings have been made so that output is toggled by both compare match A and compare match B.

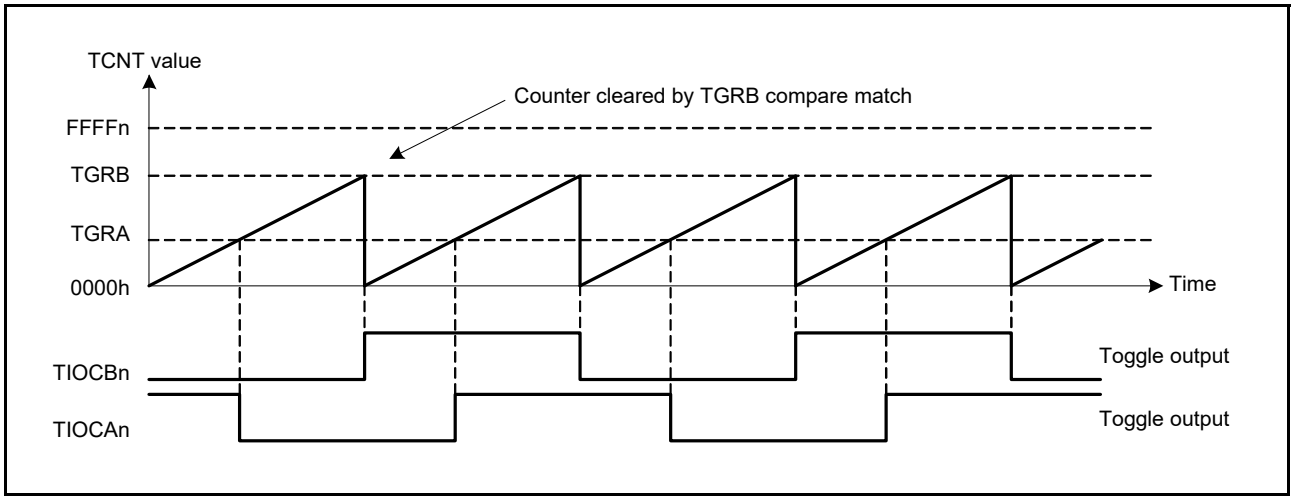


Figure 25.7 Example of Toggle Output Operation (n = 3, 4)

(3) Input Capture Function

The TPUm.TCNT value can be transferred to TPUm.TGRy on detection of the TIOCyn pin (y = A to D; n = 0 to 5) input edge.
The rising edge, the falling edge, or both edges can be selected as the detection edge. It is also possible to specify the count clock or compare match signal of TPU0, TPU1, TPU3, and TPU4 as the input capture source. Noise filtering can be applied to the input capture input.

Note: Even if the counter is halted, an input capture is generated, and flag and interrupt signals are generated.
Note: When another channel's count clock is used as the input capture input for TPU0 and TPU3, PCLK/1 should not be selected as the count clock used for input capture input. Input capture will not be generated if PCLK/1 is selected.

Downloaded from Arrow.com.

(a) Example of setting procedure for input capture operation

Figure 25.8 shows an example of the setting procedure for input capture operation.

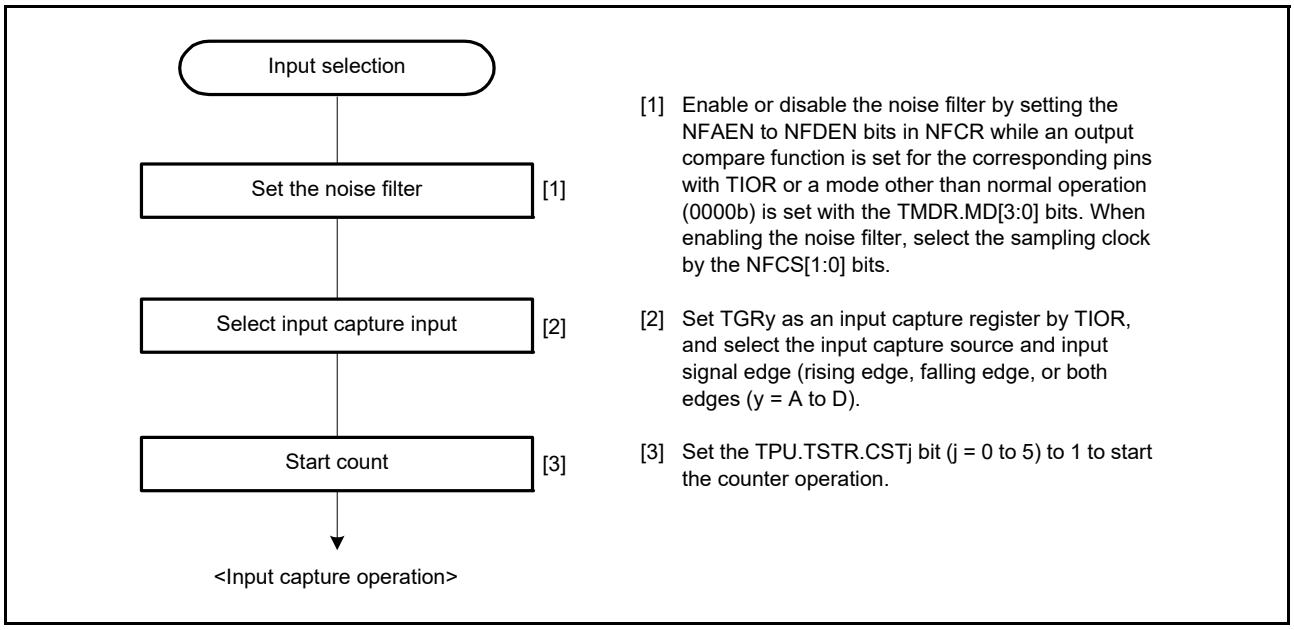


Figure 25.8 Example of Setting Procedure for Input Capture Operation

Downloaded from Arrow.com.

(b) Example of input capture operation

Figure 25.9 shows an example of input capture operation when the noise filter is stopped.
In this example, both rising and falling edges have been selected as the TIOCAn pin input capture input edge, the falling edge has been selected as the TIOCBn pin input capture input edge, and counter clearing by TPUm.TGRB input capture has been set for TPUm.TCNT.

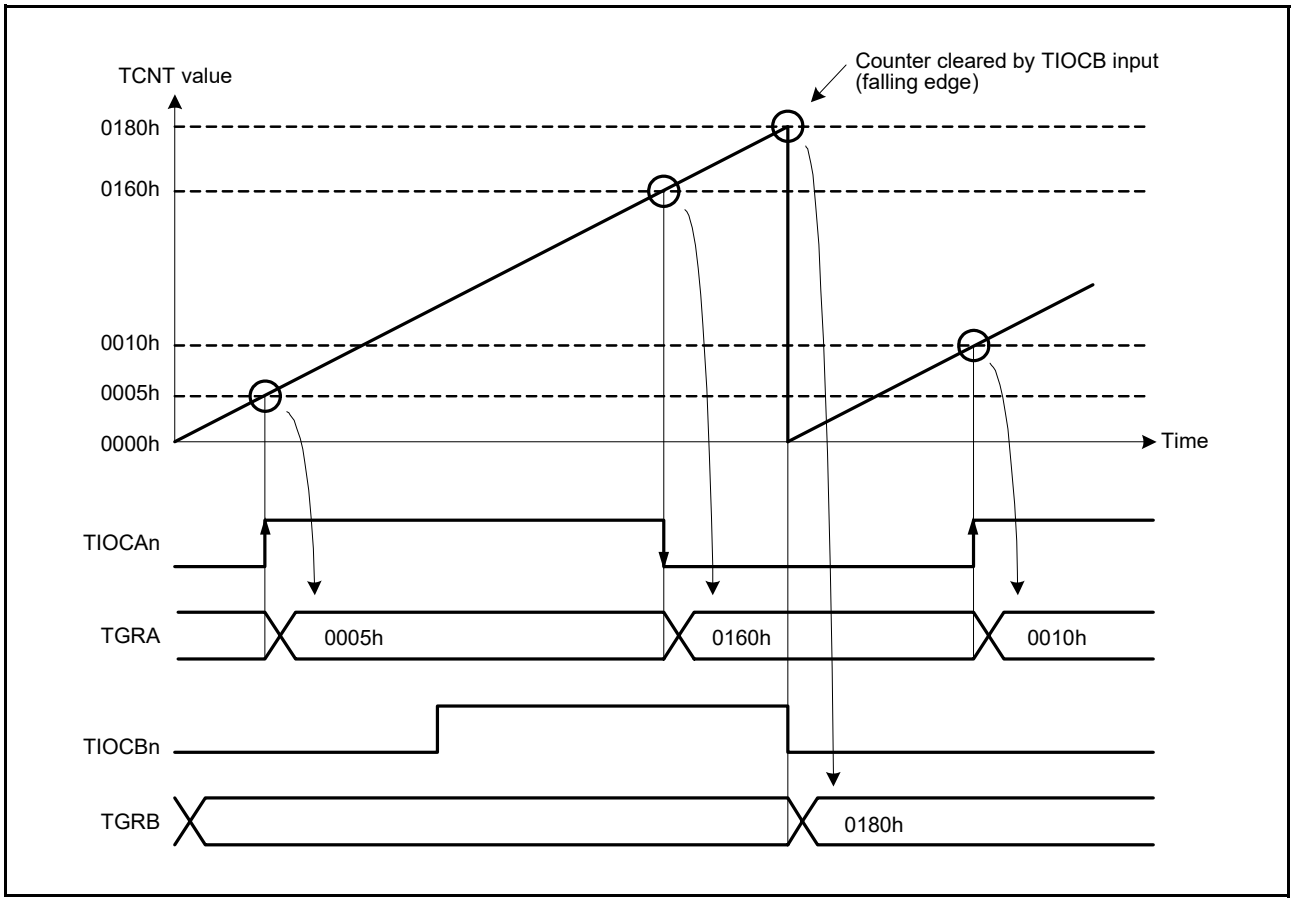


Figure 25.9 Example of Input Capture Operation (with Noise Filter Stopped) (n = 3, 4)

If noise filtering is enabled, input capture operation is performed on the edges of noise-filtered signal after a delay of (minimum sampling interval × 2 + PCLK) due to noise filtering for the input capture input.

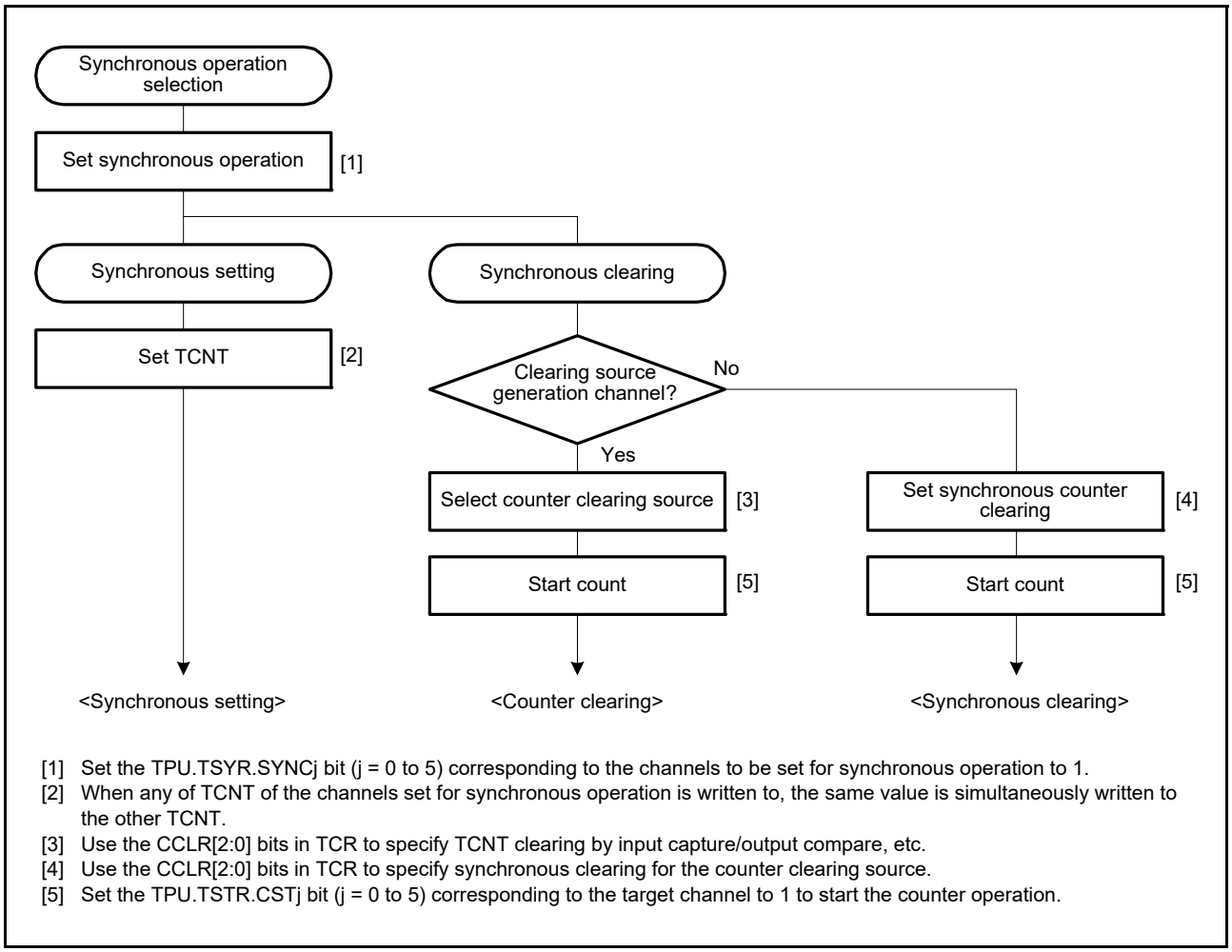
Downloaded from Arrow.com.

25.3.2 Synchronous Operation

In synchronous operation, the values in multiple TPUm.TCNT can be rewritten simultaneously (synchronous setting). Also, multiple TCNT can be cleared simultaneously (synchronous clearing) by making the appropriate setting in TPUm.TCR.
Synchronous operation enables TPUm.TGRy to be incremented with respect to a single time base.
TPU0 to TPU5 can all be set for synchronous operation.

(1) Example of Synchronous Operation Setting Procedure

Figure 25.10 shows an example of the synchronous operation setting procedure.



[1] Set the TPU.TSYR.SYNCj bit (j = 0 to 5) corresponding to the channels to be set for synchronous operation to 1.
[2] When any of TCNT of the channels set for synchronous operation is written to, the same value is simultaneously written to the other TCNT.
[3] Use the CCLR[2:0] bits in TCR to specify TCNT clearing by input capture/output compare, etc.
[4] Use the CCLR[2:0] bits in TCR to specify synchronous clearing for the counter clearing source.
[5] Set the TPU.TSTR.CSTj bit (j = 0 to 5) corresponding to the target channel to 1 to start the counter operation.

Figure 25.10 Example of Synchronous Operation Setting Procedure

Downloaded from Arrow.com.

(2) Example of Synchronous Operation

Figure 25.11 shows an example of synchronous operation.
In this example, synchronous operation and PWM mode 1 have been set for TPU0 to TPU2, TPU0.TGRA compare match has been set as the TPU0 counter clearing source, and synchronous clearing has been set for the TPU1 and TPU2 counter clearing source.
Three-phase PWM waveforms are output from pins TIOCB0, TIOCB1, and TIOCB2. At this time, synchronous setting and synchronous clearing by TPU0.TGRA compare match are performed for TPUm.TCNT of TPU0 to TPU2, and the data set in TPU0.TGRA is used as the PWM cycle.
For details on PWM modes, see section 25.3.5, PWM Modes.

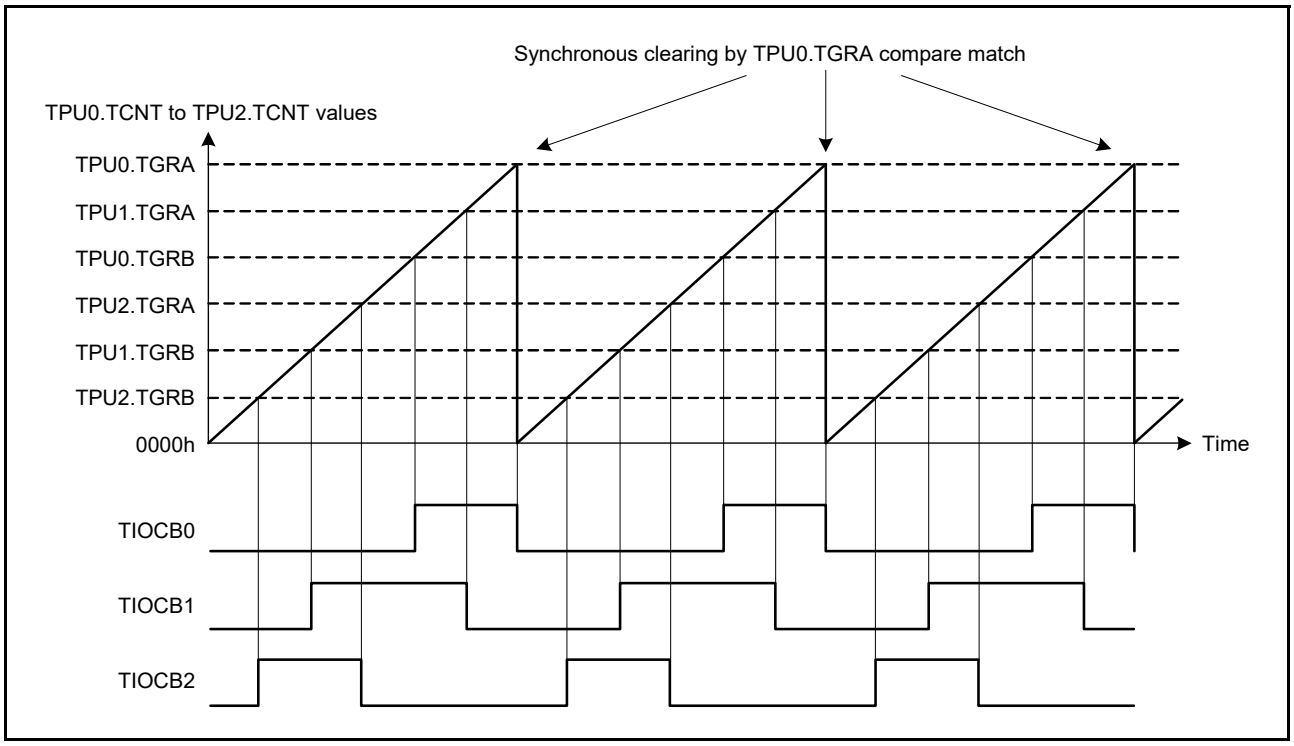


Figure 25.11 Example of Synchronous Operation

Downloaded from Arrow.com.

25.3.3 Buffer Operation

Buffer operation, provided for TPU0 and TPU3, enables TPUm.TGRC and TPUm.TGRD to be used as buffer registers. Buffer operation differs depending on whether TPUm.TGRy has been set as an input capture register or a compare match register.

Table 25.20 lists the register combinations used in buffer operation.

Table 25.20 Register Combinations

Channel	Timer General Register	Buffer Register
TPU0	TPU0.TGRA	TPU0.TGRC
	TPU0.TGRB	TPU0.TGRD
TPU3	TPU3.TGRA	TPU3.TGRC
	TPU3.TGRB	TPU3.TGRD

- When TPUm.TGRy is an output compare register

When a compare match occurs, the value in the buffer register for the corresponding channel is transferred to the timer general register.

This operation is shown in Figure 25.12.

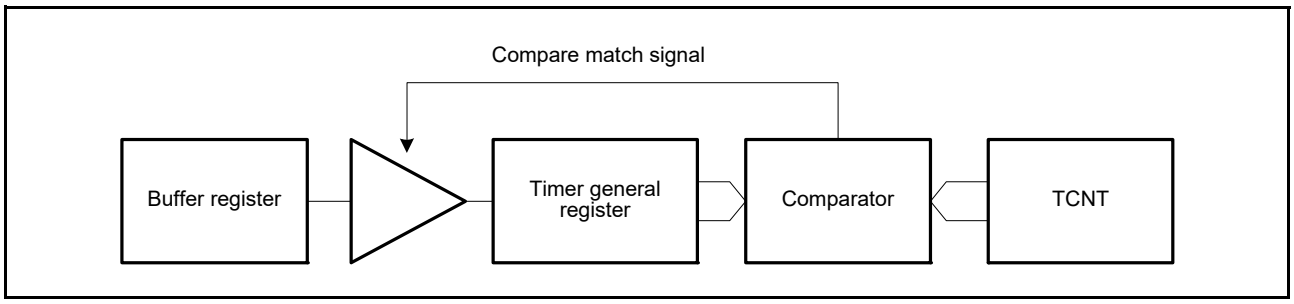


Figure 25.12 Compare Match Buffer Operation

- When TPUm.TGRy is an input capture register

When input capture occurs, the value in TPUm.TCNT is transferred to TGRy and the value previously held in TGRy is simultaneously transferred to the buffer register.

This operation is shown in Figure 25.13.

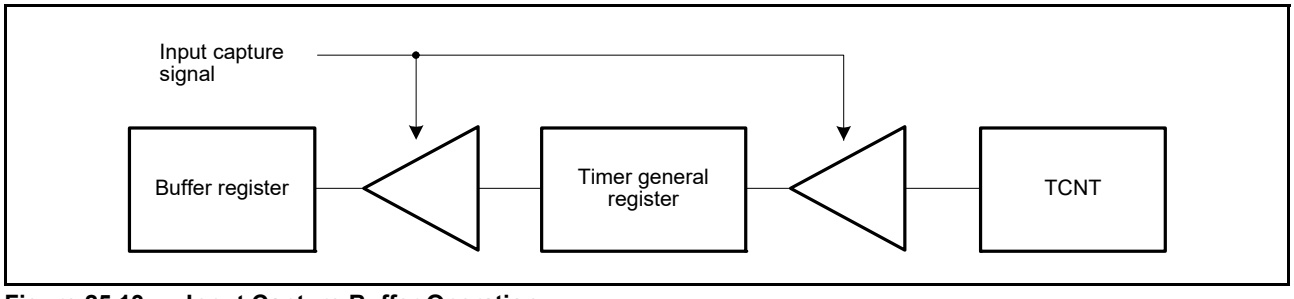


Figure 25.13 Input Capture Buffer Operation

Downloaded from Arrow.com.

(1) Example of Buffer Operation Setting Procedure

Figure 25.14 shows an example of the buffer operation setting procedure.

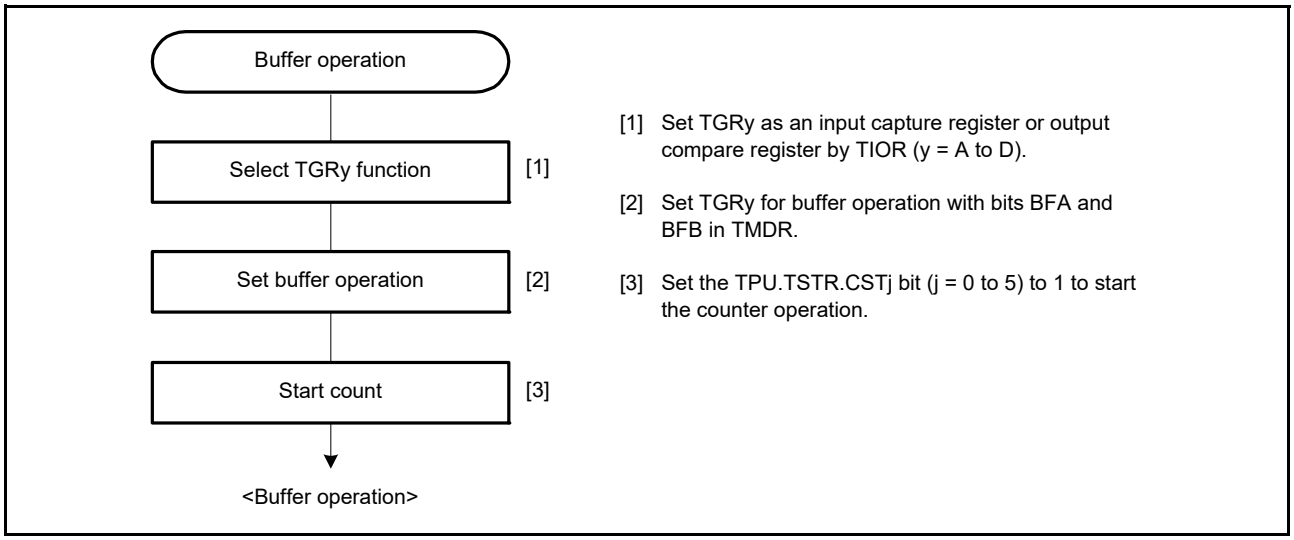


Figure 25.14 Example of Buffer Operation Setting Procedure

(2) Examples of Buffer Operation

(a) When TPUm.TGRy is an output compare register

Figure 25.15 shows an operation example in which PWM mode 1 has been set for TPU0, and buffer operation has been set for TPU0.TGRB and TPU0.TGRD. The settings used in this example are TPU0.TCNT clearing by compare match A, high output at compare match B, and low output at compare match A.

As buffer operation has been set, when compare match A occurs, the output changes and the TPU0.TGRD value is simultaneously transferred to TPU0.TGRB. This operation is repeated each time compare match B occurs.

For details on PWM modes, see section 25.3.5, PWM Modes.

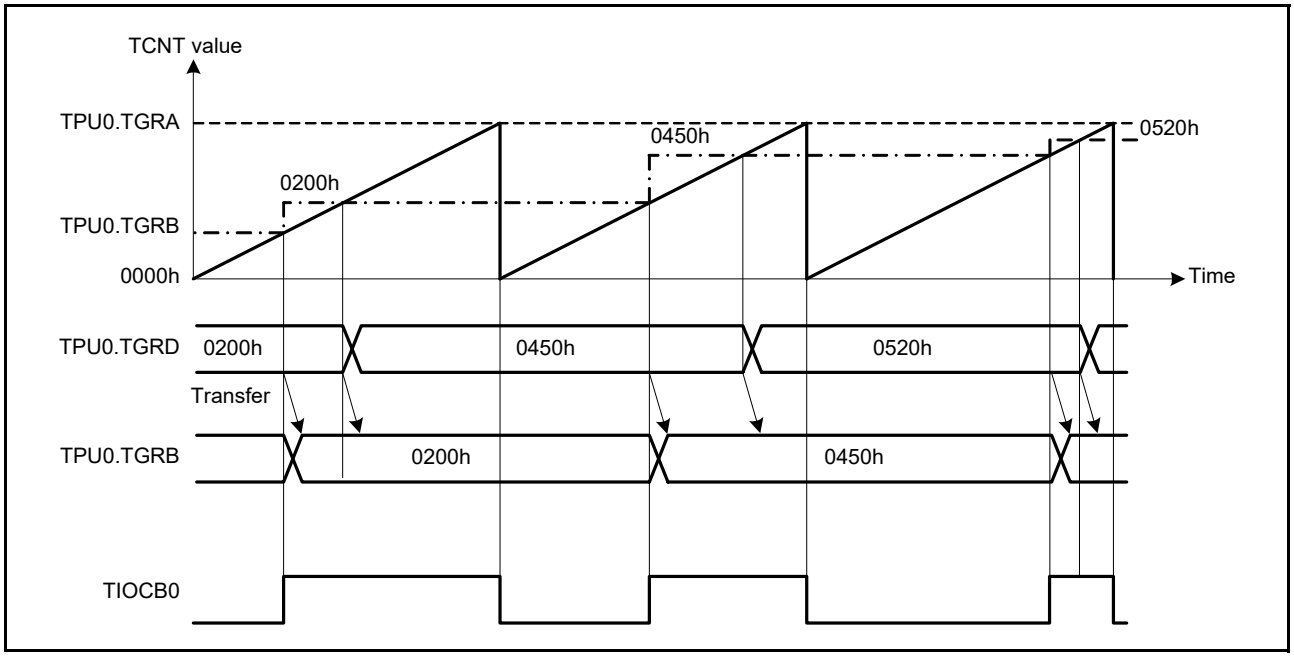


Figure 25.15 Example of Buffer Operation (1)

Downloaded from Arrow.com.

(b) When TPUm.TGRy is an input capture register

Figure 25.16 shows an operation example in which TPUm.TGRA has been set as an input capture register, and buffer operation has been set for the TGRA register and TPUm.TGRC.
Counter clearing by TGRA input capture has been set for TPUm.TCNT, and both rising and falling edges have been selected as the TIOCAn pin input capture input edge.
As buffer operation has been set, when the TCNT value is stored in TGRA upon occurrence of input capture A, the value previously stored in TGRA is simultaneously transferred to TGRC.

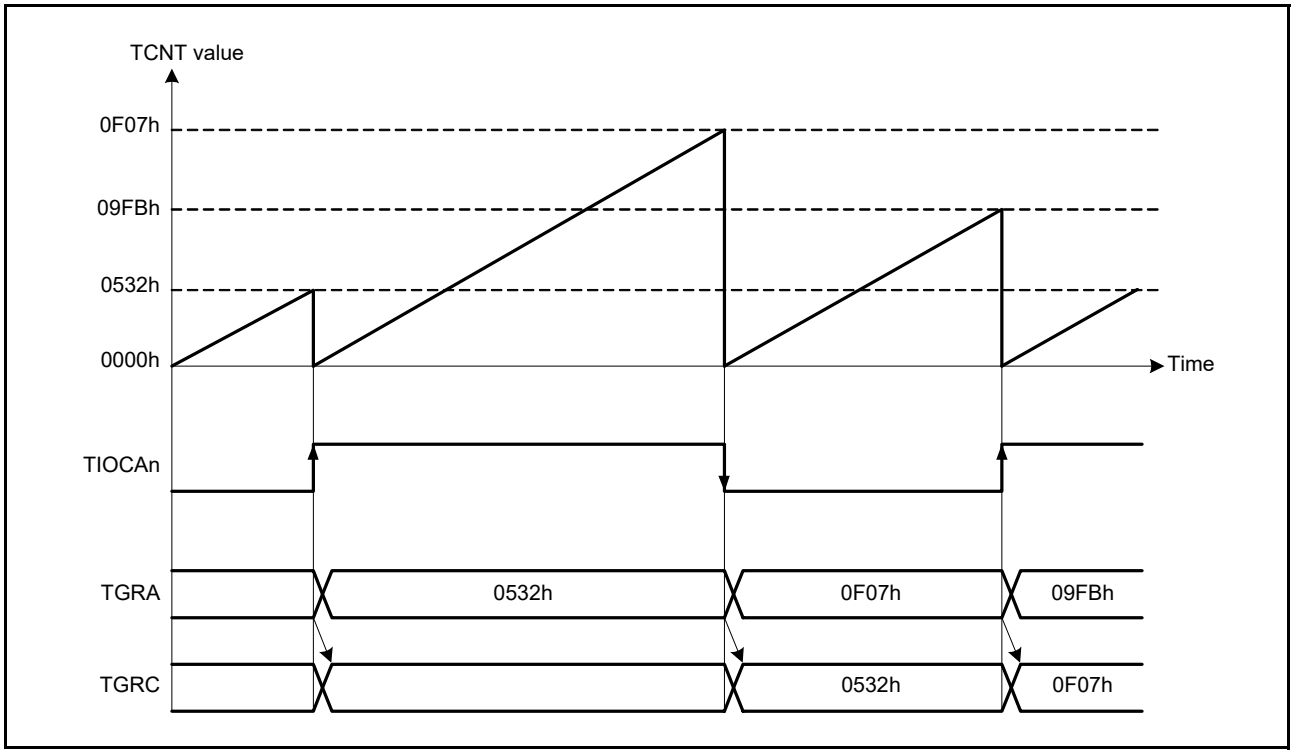


Figure 25.16 Example of Buffer Operation (2) (n = 3)

Downloaded from Arrow.com.

25.3.4 Cascaded Operation

In cascaded operation, two 16-bit counters for different channels are used together as a 32-bit counter.
This function works by counting the TPU1 (TPU4) count clock at overflow/underflow of TPU2.TCNT (TPU5.TCNT) as set by the TPSC[2:0] bits in TPU1.TCR (TPSC[2:0] bits in TPU4.TCR).
Underflow occurs only when the lower 16-bit TPUm.TCNT is in phase counting mode.
Table 25.21 lists the register combinations used in cascaded operation.

Note: When phase counting mode is set for TPU1 or TPU4, the count clock setting is invalid and the counter operates independently in phase counting mode.

Table 25.21 Cascaded Combinations

Combination	Upper 16 Bits	Lower 16 Bits
TPU1 and TPU2	TPU1.TCNT	TPU2.TCNT
TPU4 and TPU5	TPU4.TCNT	TPU5.TCNT

(1) Example of Cascaded Operation Setting Procedure

Figure 25.17 shows an example of the setting procedure for cascaded operation.

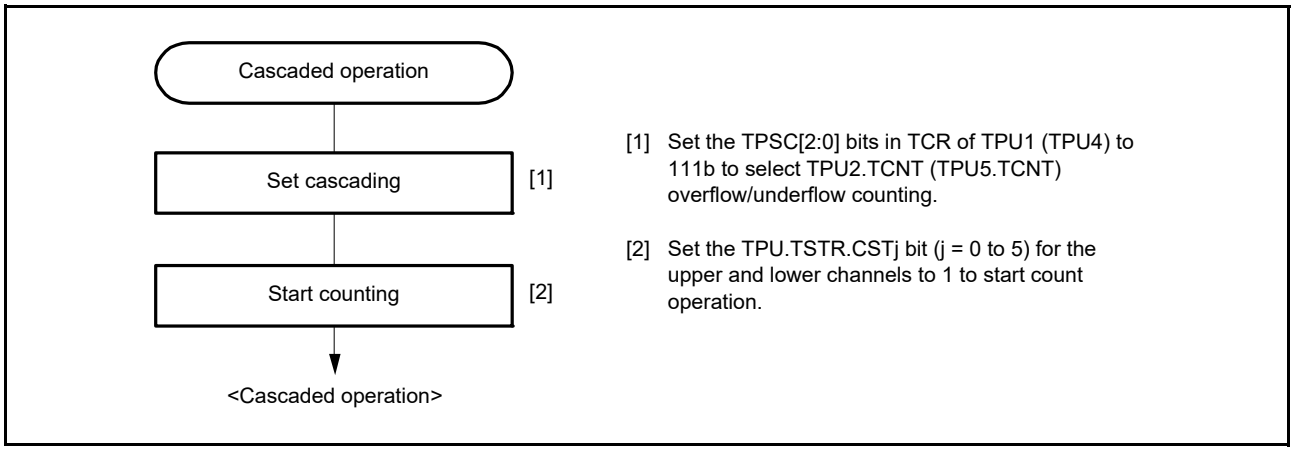


Figure 25.17 Cascaded Operation Setting Procedure

Downloaded from Arrow.com.

(2) Examples of Cascaded Operation

Figure 25.18 shows the operation when counting upon TPU2.TCNT overflow/underflow has been set for TPU1.TCNT, TPU1.TGRB and TPU2.TGRB have been set as input capture registers, and the rising edge of the TIOCB1 and TIOCB2 pins has been selected.

When a rising edge is input to the TIOCB1 and TIOCB2 pins simultaneously, the upper 16 bits of the 32-bit data are transferred to TPU1.TGRB, and the lower 16 bits to TPU2.TGRB. Note that a point for caution applies to simultaneous input capture in cascade operation, as described in section 25.9.11, TCNT Simultaneous Input Capture in Cascade Operation.

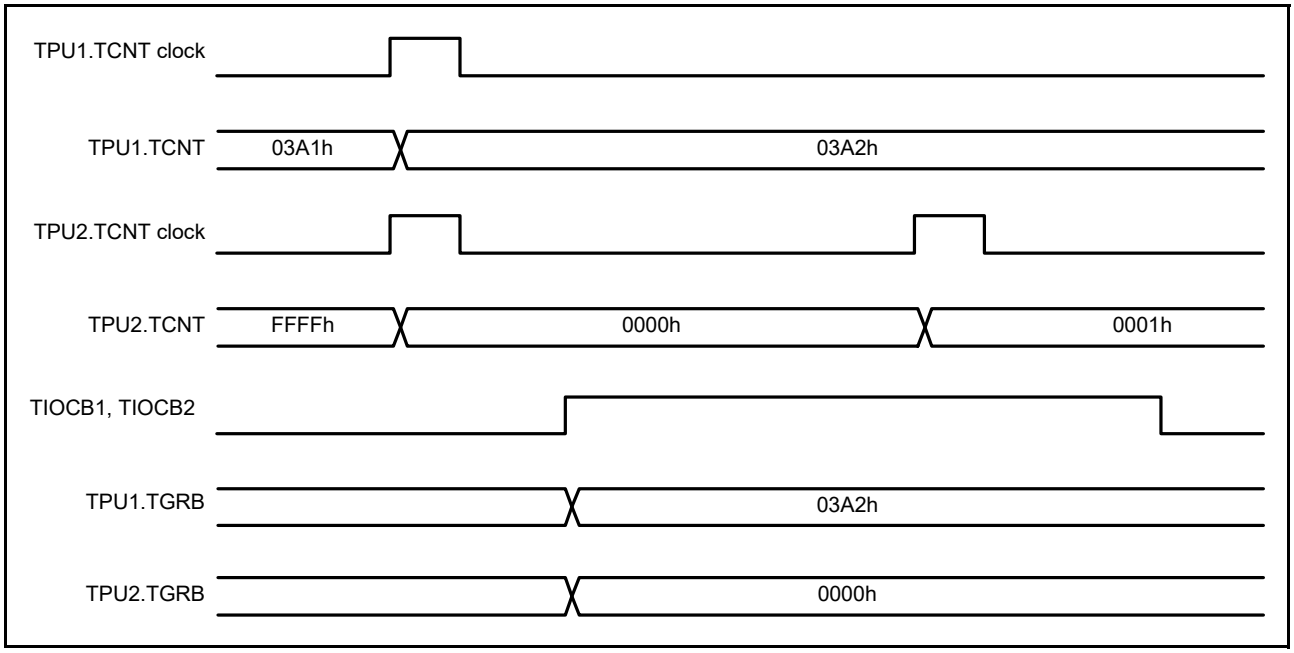


Figure 25.18 Example of Cascaded Operation (1)

Figure 25.19 shows the operation when counting upon TPU2.TCNT overflow/underflow has been set for TPU1.TCNT, and phase counting mode 1 has been specified for TPU2.

TPU1.TCNT is incremented by TPU2.TCNT overflow and decremented by TPU2.TCNT underflow.

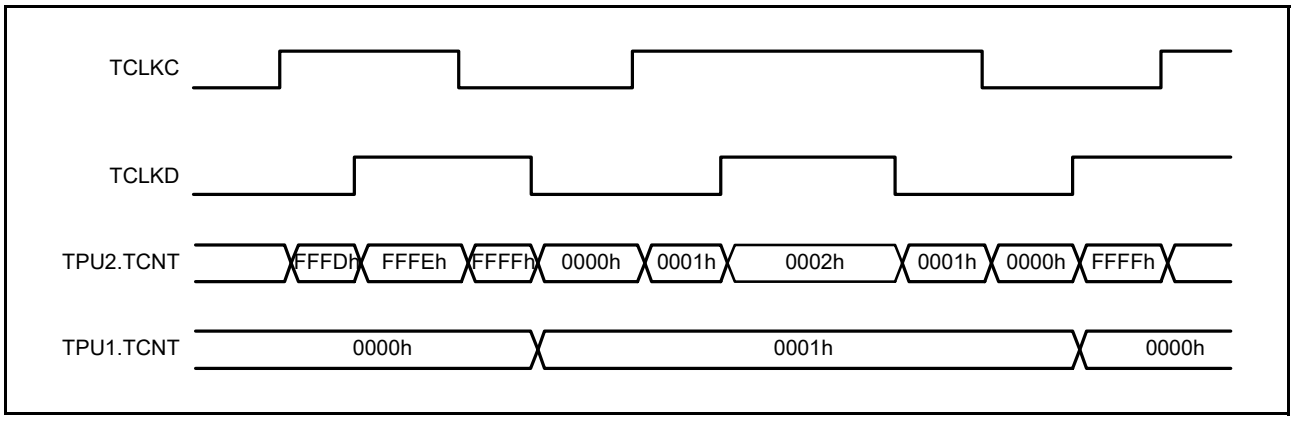


Figure 25.19 Example of Cascaded Operation (2)

Downloaded from Arrow.com.

25.3.5 PWM Modes

In PWM mode, PWM waveforms are output from the output pins. low, high, or toggle output can be selected as the output level in response to compare match of each TPUm.TGRy.

Settings of TGRy registers can output a PWM waveform in the range of 0% to 100% duty cycle.

Specifying TGRy compare match as the counter clearing source enables the cycle to be set in that register. All channels can be set for PWM mode independently. Synchronous operation is also possible.

There are two PWM modes, as described below.

1. PWM mode 1

   PWM waveform is generated from the TIOCAn and TIOCCn pins by pairing TPUm.TGRA with TPUm.TGRB and TPUm.TGRC with TPUm.TGRD. The outputs specified by the IOA[3:0] bits in TPUm.TIOR(H) and IOC[3:0] bits in TPUm.TIORL are output from the TIOCAn and TIOCCn pins at compare matches A and C, respectively. The outputs specified by the IOB[3:0] bits in TPUm.TIOR(H) and IOD[3:0] bits in TPUm.TIORL are output from the TIOCAn and TIOCCn pins at compare matches B and D, respectively. The initial output value is the value set in TGRA or TGRC. If the set values of paired TGRy registers are identical, the output value does not change even when a compare match occurs.

   In PWM mode 1, a maximum 3-phase PWM output is possible.

2. PWM mode 2

   PWM waveform is generated by using one TPUm.TGRy as the cycle register and the others as duty cycle registers. The output specified in TPUm.TIORH, TPUm.TIORL, or TPUm.TIOR is performed by compare matches. Upon counter clearing by a synchronous register compare match, the output value of each pin is the initial value set in TIORH, TIORL, or TIOR. If the set values of the cycle register and duty cycle register are identical, the output value does not change even when a compare match occurs.

   In PWM mode 2, a maximum 9-phase PWM waveform is possible by combined use with synchronous operation.

Downloaded from Arrow.com.

The correspondence between PWM output pins and registers is listed in Table 25.22.

Table 25.22 PWM Output Registers and Output Pins

Channel	Register	Output Pin	
		PWM Mode 1	PWM Mode 2
TPU0	TPU0.TGRA	No pin is assigned for this output.	No pin is assigned for this output.
	TPU0.TGRB		TIOCB0
	TPU0.TGRC	No pin is assigned for this output.	No pin is assigned for this output.
	TPU0.TGRD		No pin is assigned for this output.
TPU1	TPU1.TGRA	No pin is assigned for this output.	No pin is assigned for this output.
	TPU1.TGRB		TIOCB1
TPU2	TPU2.TGRA	No pin is assigned for this output.	No pin is assigned for this output.
	TPU2.TGRB		TIOCB2
TPU3	TPU3.TGRA	TIOCA3	TIOCA3
	TPU3.TGRB		TIOCB3
	TPU3.TGRC	TIOCC3	TIOCC3
	TPU3.TGRD		TIOCD3
TPU4	TPU4.TGRA	TIOCA4	TIOCA4
	TPU4.TGRB		TIOCB4
TPU5	TPU5.TGRA	No pin is assigned for this output.	No pin is assigned for this output.
	TPU5.TGRB		TIOCB5

Note: In PWM mode 2, PWM waveform output is not possible for the TPUm.TGRy register in which the cycle is set.

Downloaded from Arrow.com.

(1) Example of PWM Mode Setting Procedure

Figure 25.20 shows an example of the PWM mode setting procedure.

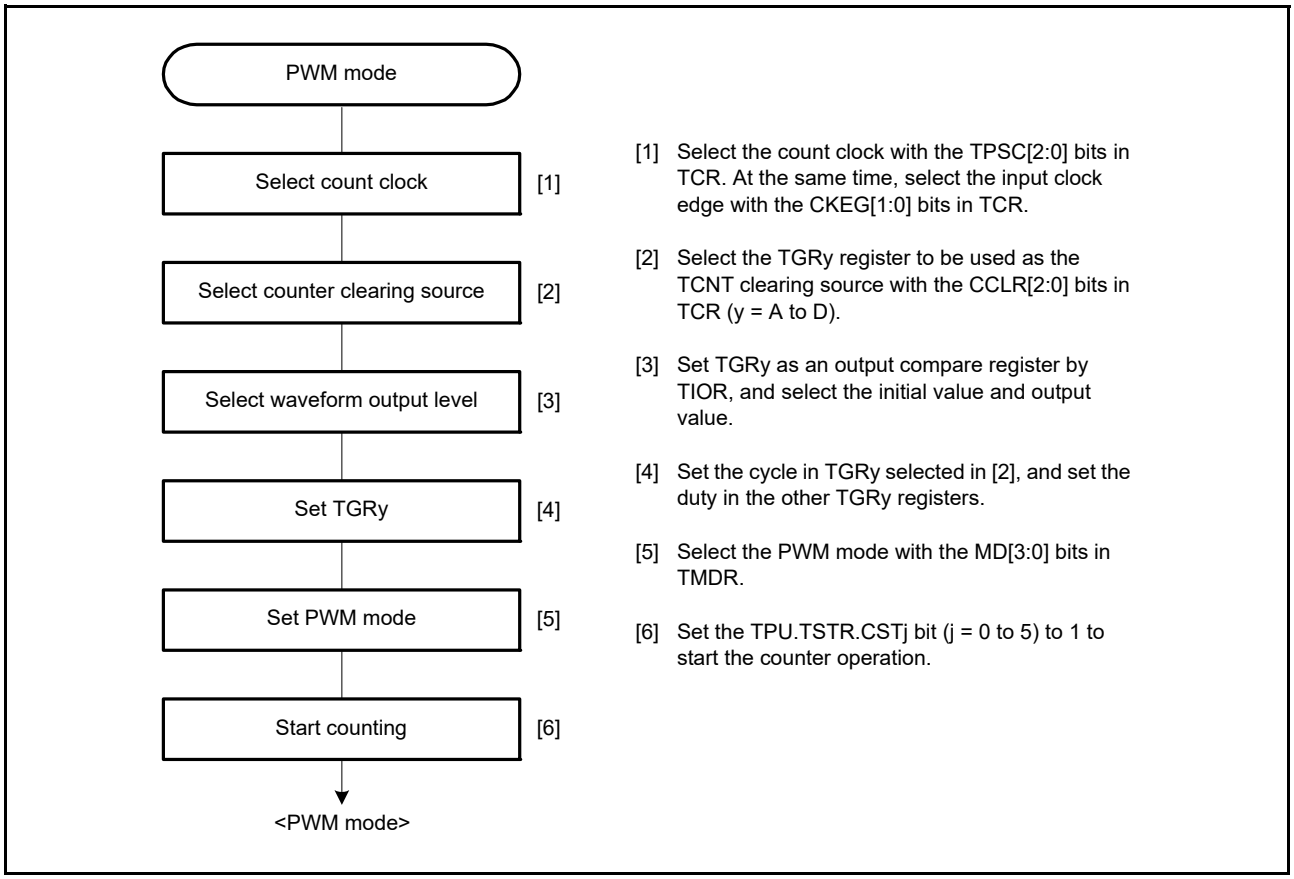


Figure 25.20 Example of PWM Mode Setting Procedure

Downloaded from Arrow.com.

(2) Examples of PWM Mode Operation

Figure 25.21 shows an example of PWM mode 1 operation.
In this example, TPUm.TGRA compare match is set as the TPUm.TCNT clearing source, low is set for the TGRA initial output value and output value, and high is set as the TPUm.TGRB output value.
In this case, the value set in TGRA is used as the cycle, and the value set in TGRB is used as the duty cycle.

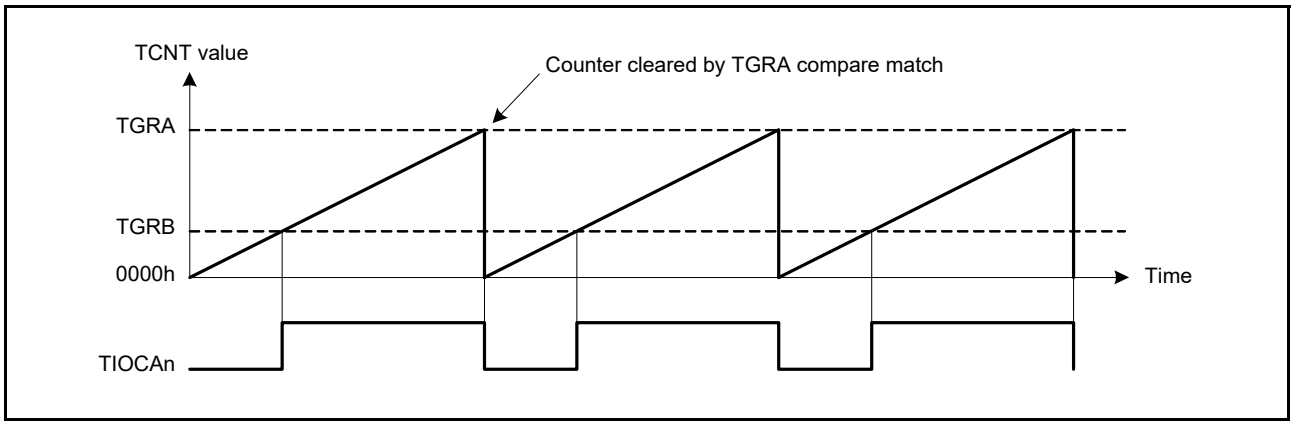


Figure 25.21 Example of PWM Mode Operation (1) (n = 3, 4)

Figure 25.22 shows an example of PWM mode 2 operation.
In this example, synchronous operation is specified for TPU3 and TPU4, TPU4.TGRB compare match is set as the TPUm.TCNT clearing source, and low is set for the initial output value and high for the output value of the other TPUm.TGRy registers (TPU3.TGRA to TPU3.TGRD and TPU4.TGRA), to output a 5-phase PWM waveform.
In this case, the value set in TPU1.TGRB is used as the cycle, and the values set in the other TGRy registers are used as the duty cycle.

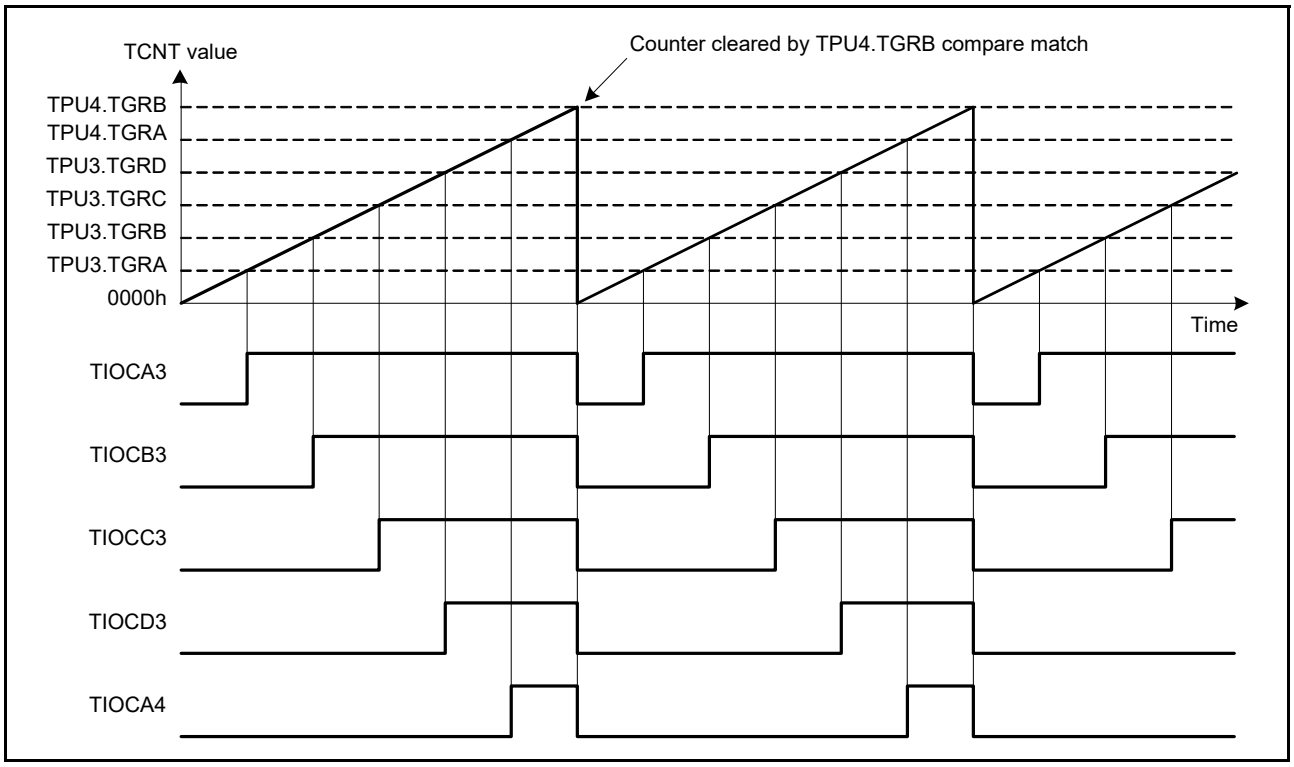


Figure 25.22 Example of PWM Mode Operation (2)

Downloaded from Arrow.com.

Figure 25.23 shows examples of PWM waveform output with 0% duty cycle and 100% duty cycle in PWM mode.

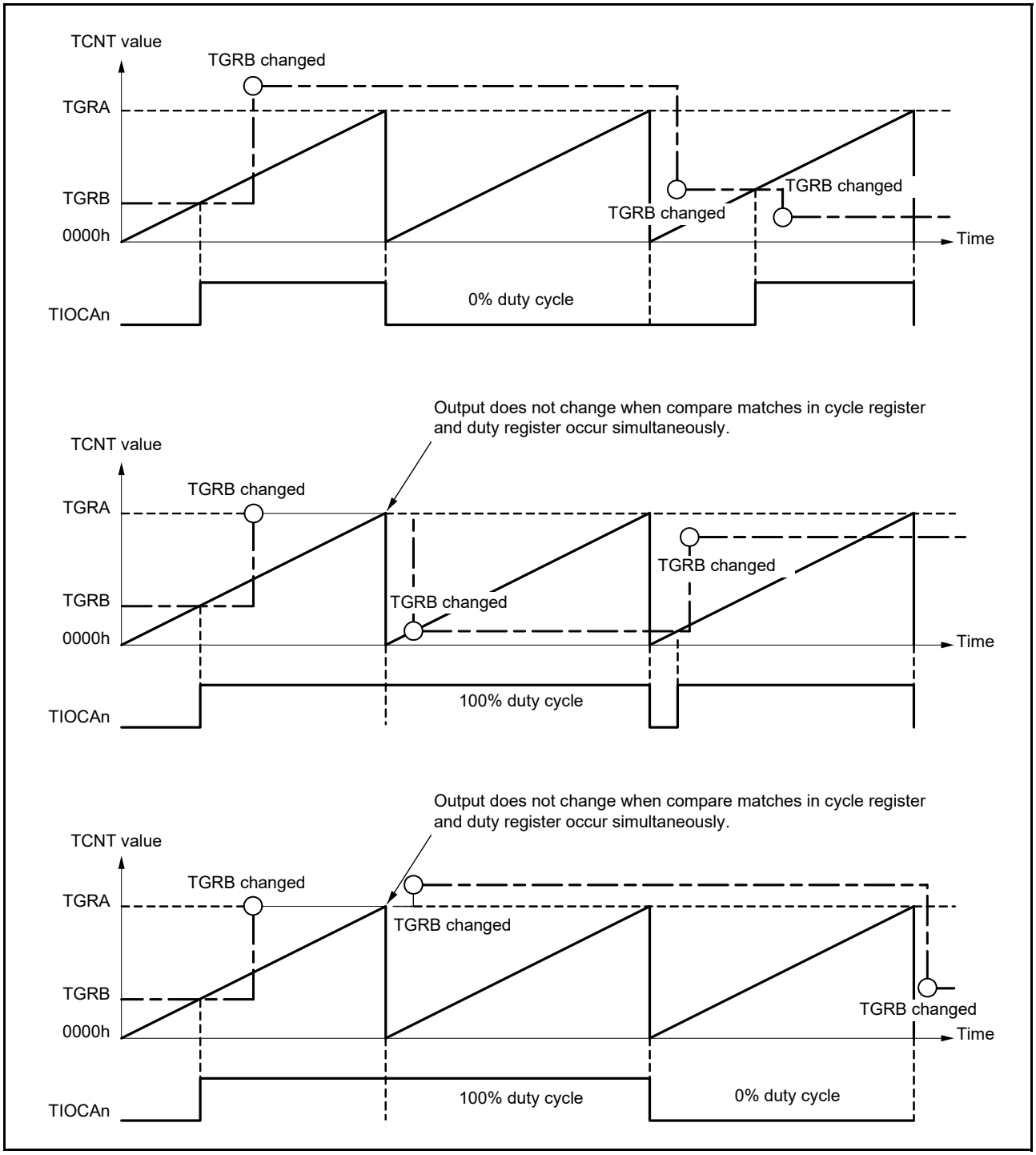


Figure 25.23 Example of PWM Mode Operation (3) (n = 3, 4)

Downloaded from Arrow.com.

25.3.6 Phase Counting Mode

In phase counting mode, the phase difference between two external clock inputs is detected by the settings for channels 1, 2, 4, and 5, and TPUm.TCNT is incremented/decremented accordingly.
When phase counting mode is set, an external clock is selected as the count clock and TCNT operates as an up-/down-counter regardless of the setting of the TPSC[2:0] bits and CKEG[1:0] bits in TPUm.TCR. However, the lower 2 bits of the CCLR[2:0] bits in TPUm.TCR and the functions of TPUm.TIORH, TPUm.TIORL, TPUm.TIOR, TPUm.TIER, and TPUm.TGRy are valid, and therefore input capture/compare match and interrupt functions are available.
When an overflow occurs while TCNT is counting up, a TCIV interrupt request is generated; when an underflow occurs while TCNT is counting down, a TCIU interrupt request is generated. The TCFD bit in TPUm.TSR is the count direction flag. Reading the TCFD flag provides an indication of whether TCNT is counting up or down.
In phase counting mode, the external clock pins TCLKA, TCLKB, TCLKC, and TCLKD can be used as 2-phase encoder pulse input.
Table 25.23 lists the correspondence between external clock pins and channels.

Table 25.23 Clock Input Pins in Phase Counting Mode

	External Clock Pins	
Channel	**A-Phase**	**B-Phase**
When TPU1 or TPU5 is set to phase counting mode	TCLKA	TCLKB
When TPU2 or TPU4 is set to phase counting mode	TCLKC	TCLKD

(1) Example of Phase Counting Mode Setting Procedure

Figure 25.24 shows an example of the phase counting mode setting procedure.

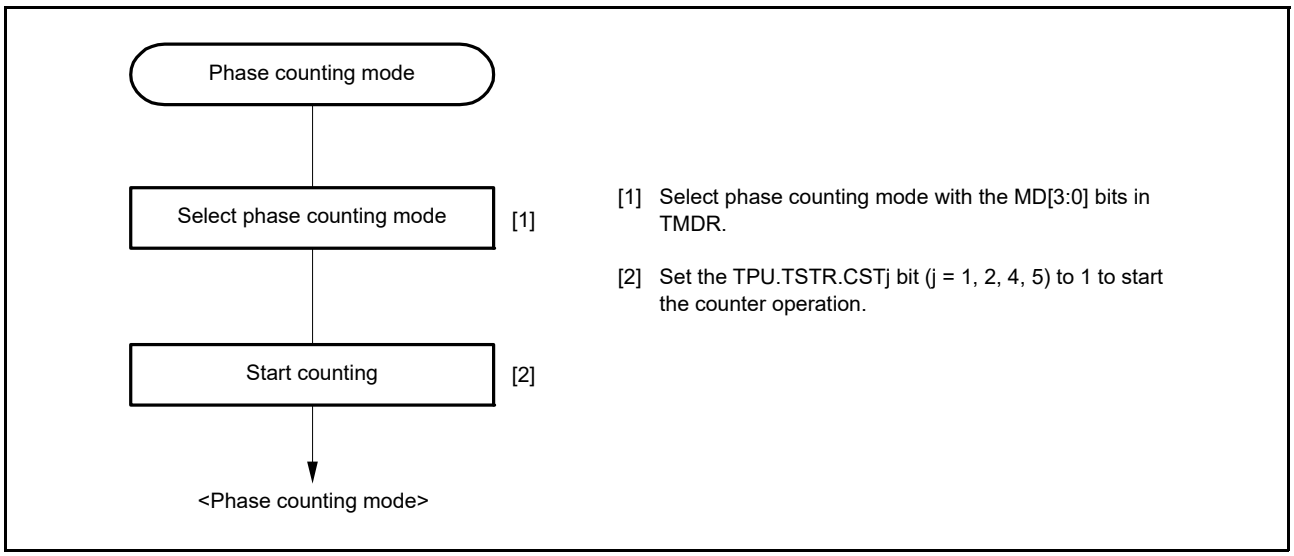


Figure 25.24 Example of Phase Counting Mode Setting Procedure

Downloaded from Arrow.com.

(2) Examples of Phase Counting Mode Operation

In phase counting mode, TPUm.TCNT counts up or down according to the phase difference between two external clocks. There are four modes, according to the count conditions.

(a) Phase counting mode 1

Figure 25.25 shows an example of phase counting mode 1 operation, and Table 25.24 lists the TPUm.TCNT up-/down-count conditions.

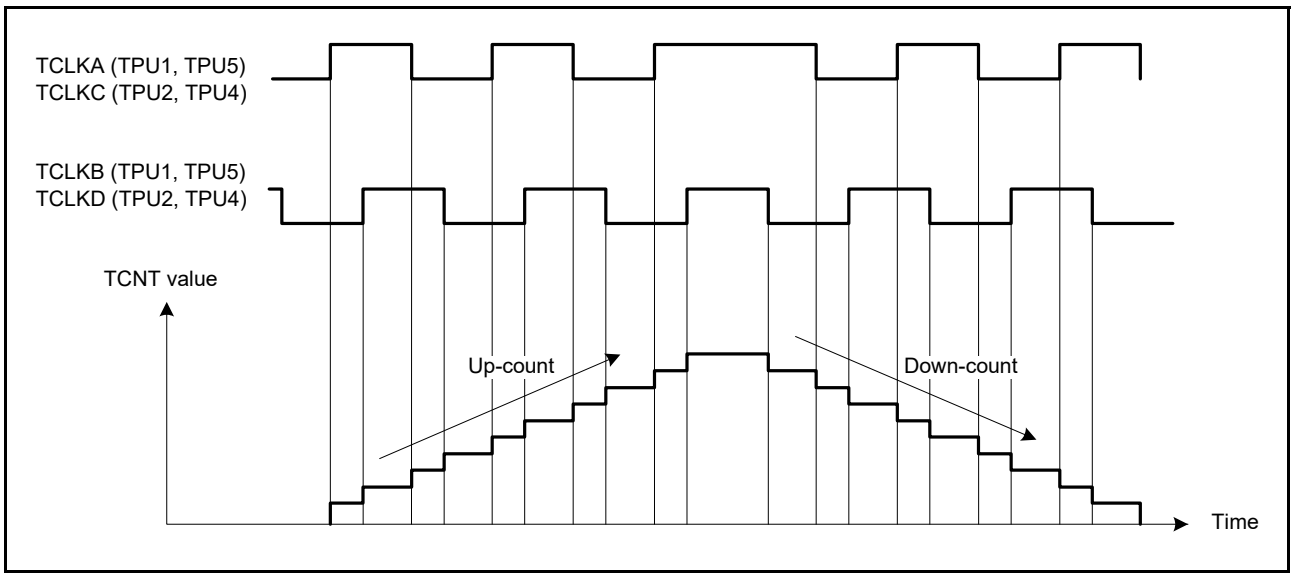


Figure 25.25 Example of Phase Counting Mode 1 Operation

Table 25.24 Up-/Down-Count Conditions in Phase Counting Mode 1

TCLKA (TPU1, TPU5) TCLKC (TPU2, TPU4)	TCLKB (TPU1, TPU5) TCLKD (TPU2, TPU4)	Operation
High	↑	Up-count
Low	↓	
↑	Low	
↓	High	
High	↓	Down-count
Low	↑	
↑	High	
↓	Low	

↑ : Rising edge

↓ : Falling edge

Downloaded from Arrow.com.

(b) Phase counting mode 2

Figure 25.26 shows an example of phase counting mode 2 operation, and Table 25.25 lists the TPUm.TCNT up-/down-count conditions.

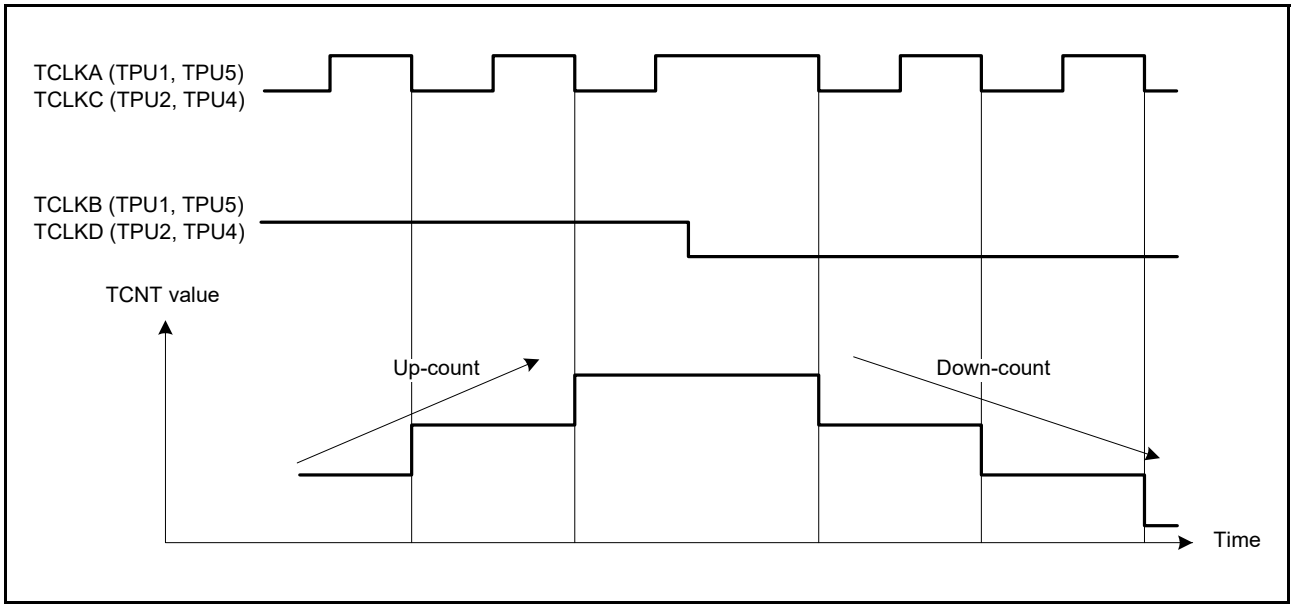


Figure 25.26 Example of Phase Counting Mode 2 Operation

Table 25.25 Up-/Down-Count Conditions in Phase Counting Mode 2

TCLKA (TPU1, TPU5) TCLKC (TPU2, TPU4)	TCLKB (TPU1, TPU5) TCLKD (TPU2, TPU4)	Operation
High	↑	Don't care
Low	↓	Don't care
↑	Low	Don't care
↓	High	Up-count
High	↓	Don't care
Low	↑	Don't care
↑	High	Don't care
↓	Low	Down-count

↑ : Rising edge

↓ : Falling edge

Downloaded from Arrow.com.

(c) Phase counting mode 3

Figure 25.27 shows an example of phase counting mode 3 operation, and Table 25.26 lists the TPUm.TCNT up-/down-count conditions.

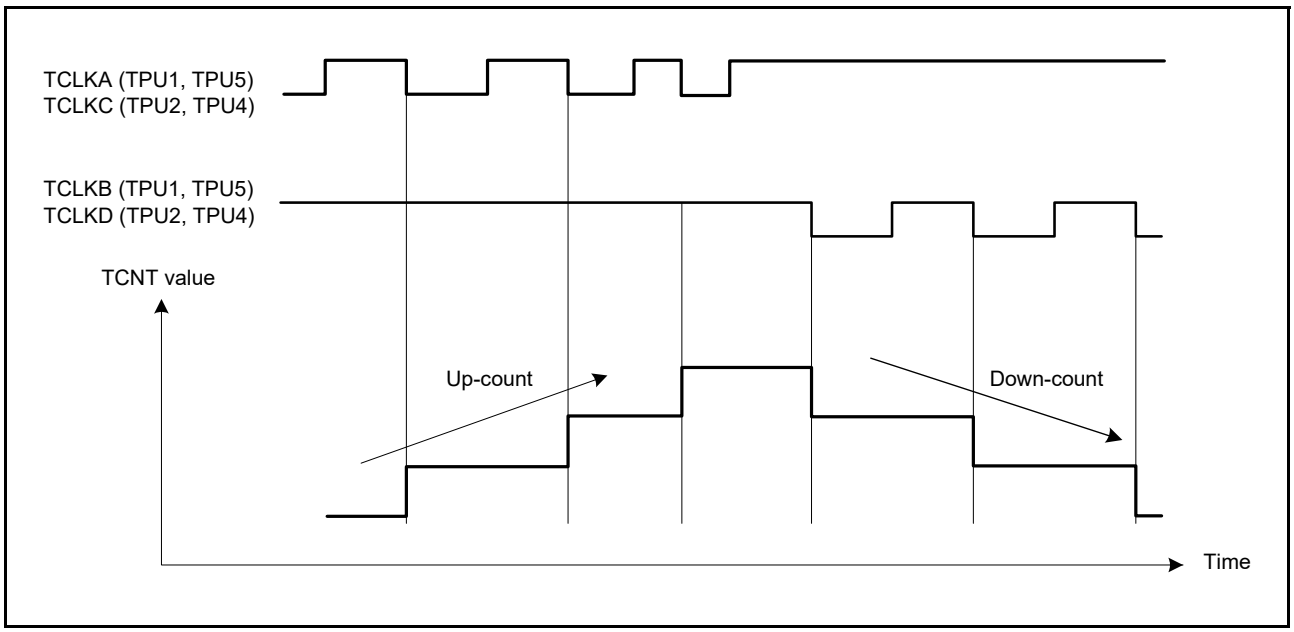


Figure 25.27 Example of Phase Counting Mode 3 Operation

Table 25.26 Up-/Down-Count Conditions in Phase Counting Mode 3

TCLKA (TPU1, TPU5) TCLKC (TPU2, TPU4)	TCLKB (TPU1, TPU5) TCLKD (TPU2, TPU4)	Operation
High	↑	Don't care
Low	↓	Don't care
↑	Low	Don't care
↓	High	Up-count
High	↓	Down-count
Low	↑	Don't care
↑	High	Don't care
↓	Low	Don't care

↑ : Rising edge

↓ : Falling edge

Downloaded from Arrow.com.

(d) Phase counting mode 4

Figure 25.28 shows an example of phase counting mode 4 operation, and Table 25.27 lists the TPUm.TCNT up-/down-count conditions.

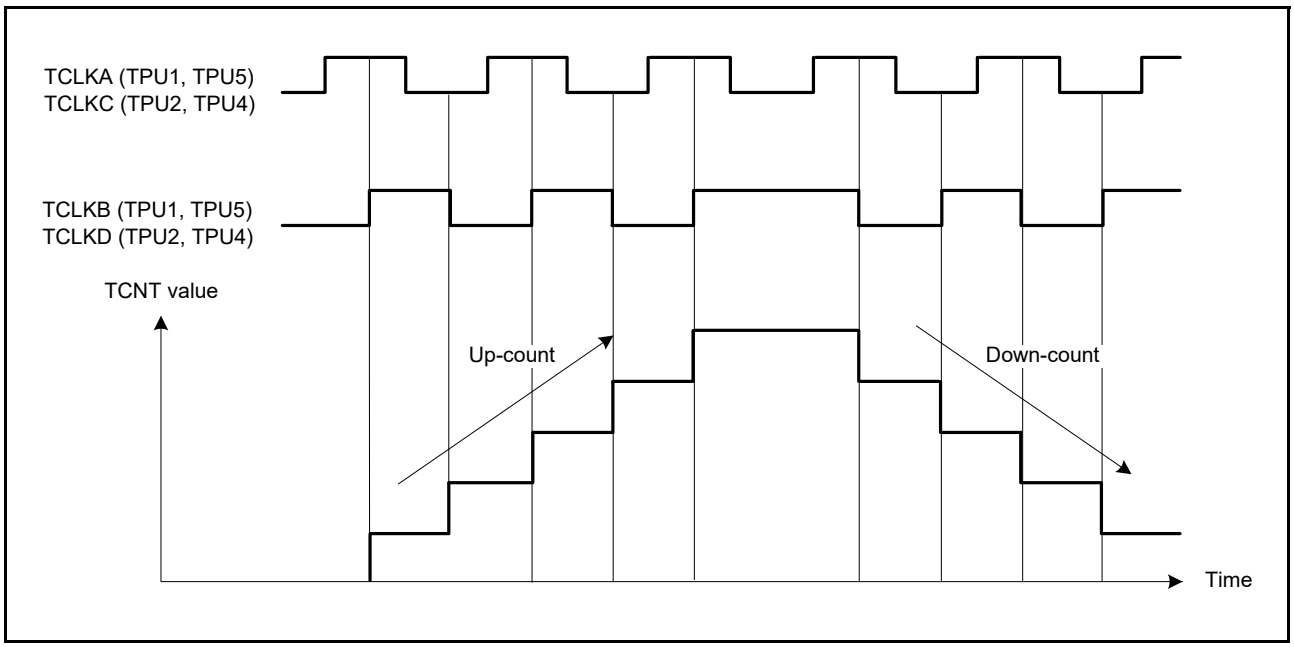


Figure 25.28 Example of Phase Counting Mode 4 Operation

Table 25.27 Up-/Down-Count Conditions in Phase Counting Mode 4

TCLKA (TPU1, TPU5) TCLKC (TPU2, TPU4)	TCLKB (TPU1, TPU5) TCLKD (TPU2, TPU4)	Operation
High	↑	Up-count
Low	↓	
↑	Low	Don't care
↓	High	
High	↓	Down-count
Low	↑	
↑	High	Don't care
↓	Low	

↑ : Rising edge

↓ : Falling edge

Downloaded from Arrow.com.

25.3.6.1 Phase Counting Mode Application Example

Figure 25.29 shows an example in which phase counting mode is set for TPU4, and TPU4 is coupled with TPU3 to input servo motor 2-phase encoder pulses in order to detect the position or speed.

TPU4 is set to phase counting mode 1, and the encoder pulse A-phase and B-phase are input to the TCLKA and TCLKB pins.

TPU3 operates with TPU3.TCNT clearing by TPU3.TGRC compare match; TPU3.TGRA and TPU3.TGRC are used for the compare match function and are set with the speed control cycle and position control cycle. TPU3.TGRB is used for input capture, with TPU3.TGRB and TPU3.TGRD operating in buffer mode. The TPU4 count clock is specified as the TPU3.TGRB input capture source, and the pulse width of 2-phase encoder 4-multiplication pulses is detected.

TPU4.TGRA and TPU4.TGRB for TPU4 are specified for input capture, TPU3.TGRA and TPU3.TGRC compare matches are selected as the input capture source, and the up-/down-counter values for the control cycles are stored.

This procedure enables accurate position/speed detection to be achieved.

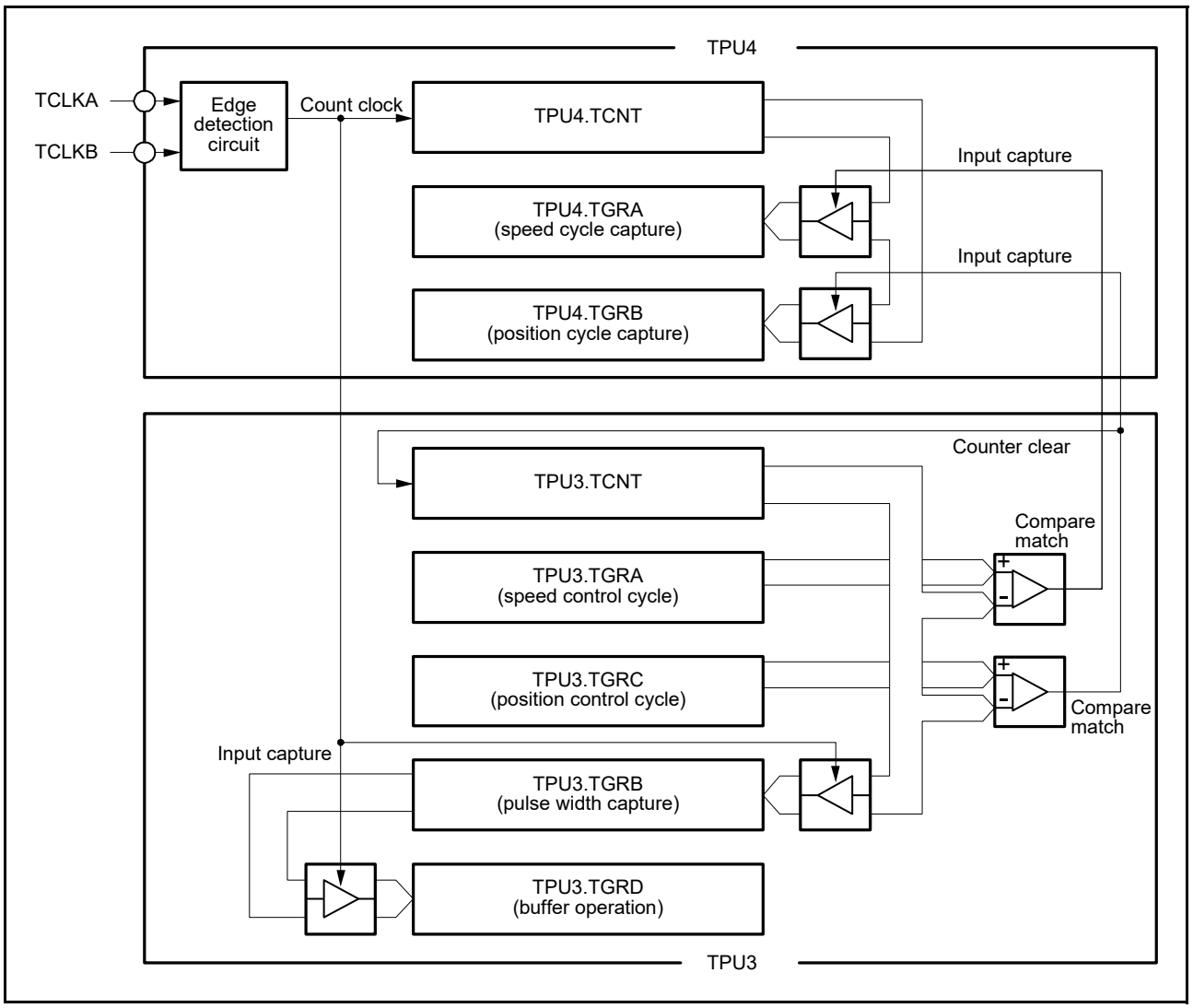


Figure 25.29 Phase Counting Mode Application Example

Downloaded from Arrow.com.

25.3.7 Noise Filters

Each pin for use in input capture by TPU is equipped with a noise filter. The noise filter samples the level on the pin three times at the selected sampling interval, conveys the level to the internal circuits if the samples match, and continues to convey that level until the other level is sampled from the pins three times in a row. The noise filter function can be enabled or disabled for each pin. Furthermore, sampling clock settings can be made for each channel.
Figure 25.30 is a timing chart for the noise filter.

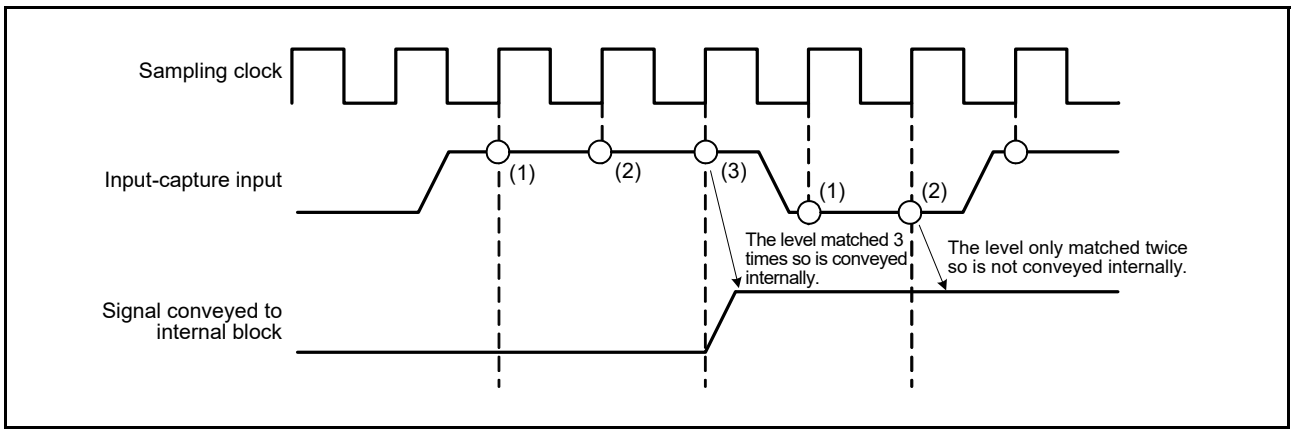


Figure 25.30 Timing Chart for the Noise Filter

Downloaded from Arrow.com.

25.4 Interrupt Sources

There are three kinds of TPU interrupt sources: TPUm.TGRy input capture/compare match, TPUm.TCNT overflow, and TPUm.TCNT underflow.

Relative channel priority levels can be changed by the interrupt controller, but the priority within a channel is fixed. For details, see section 15, Interrupt Controller (ICUb).

Table 25.28 lists the TPU interrupt sources.

Table 25.28 TPU Interrupt Sources

Channel	Name	Interrupt Source	DTC Activation	DMAC Activation
TPU0	TGI0A	TPU0.TGRA compare match	Possible	Possible
	TGI0B	TPU0.TGRB input capture/compare match	Possible	Not possible
	TGI0C	TPU0.TGRC compare match	Possible	Not possible
	TGI0D	TPU0.TGRD compare match	Possible	Not possible
	TCI0V	TPU0.TCNT overflow	Not possible	Not possible
TPU1	TGI1A	TPU1.TGRA compare match	Possible	Possible
	TGI1B	TPU1.TGRB input capture/compare match	Possible	Not possible
	TCI1V	TPU1.TCNT overflow	Not possible	Not possible
	TCI1U	TPU1.TCNT underflow	Not possible	Not possible
TPU2	TGI2A	TPU2.TGRA compare match	Possible	Possible
	TGI2B	TPU2.TGRB input capture/compare match	Possible	Not possible
	TCI2V	TPU2.TCNT overflow	Not possible	Not possible
	TCI2U	TPU2.TCNT underflow	Not possible	Not possible
TPU3	TGI3A	TPU3.TGRA input capture/compare match	Possible	Possible
	TGI3B	TPU3.TGRB input capture/compare match	Possible	Not possible
	TGI3C	TPU3.TGRC input capture/compare match	Possible	Not possible
	TGI3D	TPU3.TGRD input capture/compare match	Possible	Not possible
	TCI3V	TPU3.TCNT overflow	Not possible	Not possible
TPU4	TGI4A	TPU4.TGRA input capture/compare match	Possible	Possible
	TGI4B	TPU4.TGRB input capture/compare match	Possible	Not possible
	TCI4V	TPU4.TCNT overflow	Not possible	Not possible
	TCI4U	TPU4.TCNT underflow	Not possible	Not possible
TPU5	TGI5A	TPU5.TGRA compare match	Possible	Possible
	TGI5B	TPU5.TGRB input capture/compare match	Possible	Not possible
	TCI5V	TPU5.TCNT overflow	Not possible	Not possible
	TCI5U	TPU5.TCNT underflow	Not possible	Not possible

Note: This table lists the initial state immediately after a reset. The relative channel priority levels can be changed by the interrupt controller.

Downloaded from Arrow.com.

(1) Input Capture/Compare Match Interrupt

An interrupt is requested when the TGIEy bit (y = A, B, C, D) in TPUm.TIER is set to 1 by the occurrence of a TPUm.TGRy input capture/compare match on a channel. The TPU has 16 input capture/compare match interrupts, four each for TPU0 and TPU3, and two each for TPU1, TPU2, TPU4, and TPU5.

(2) Overflow Interrupt

An interrupt is requested when the TCIEV bit in TPUm.TIER is set to 1 by the occurrence of a TPUm.TCNT overflow on a channel. The TPU has six overflow interrupts, one for each channel.

(3) Underflow Interrupt

An interrupt is requested when the TCIEU bit in TPUm.TIER is set to 1 by the occurrence of a TPUm.TCNT underflow on a channel. The TPU has four underflow interrupts, one each for TPU1, TPU2, TPU4, and TPU5.

25.5 DTC Activation

The DTC can be activated by the TPUm.TGRy input capture/compare match interrupt of each channel. For details, see section 19, Data Transfer Controller (DTCa).
A total of 16 input capture/compare match interrupts can be used as DTC activation sources, four each for TPU0 and TPU3, and two each for TPU1, TPU2, TPU4, and TPU5.

25.6 DMAC Activation

The DMAC can be activated by the TPUm.TGRA input capture/compare match interrupt of each channel. For details, see section 18, DMA Controller (DMACA).
A total of six TPUm.TGRA input capture/compare match interrupts can be used as DMAC activation sources, one for each channel.

25.7 A/D Converter Activation

The TPU can activate the A/D converter by the TPUm.TGRA input capture/compare match for each channel.
When the TTGE bit in TPUm.TIER is set to 1, the TPU requests the A/D converter to start A/D conversion by the occurrence of a TPUm.TGRA input capture/compare match on a particular channel.

Downloaded from Arrow.com.

25.8 Operation Timing

25.8.1 Input/Output Timing

(1) TPUm.TCNT Count Timing

Figure 25.31 shows TPUm.TCNT count timing in internal clock operation, and Figure 25.32 shows TCNT count timing in external clock operation.

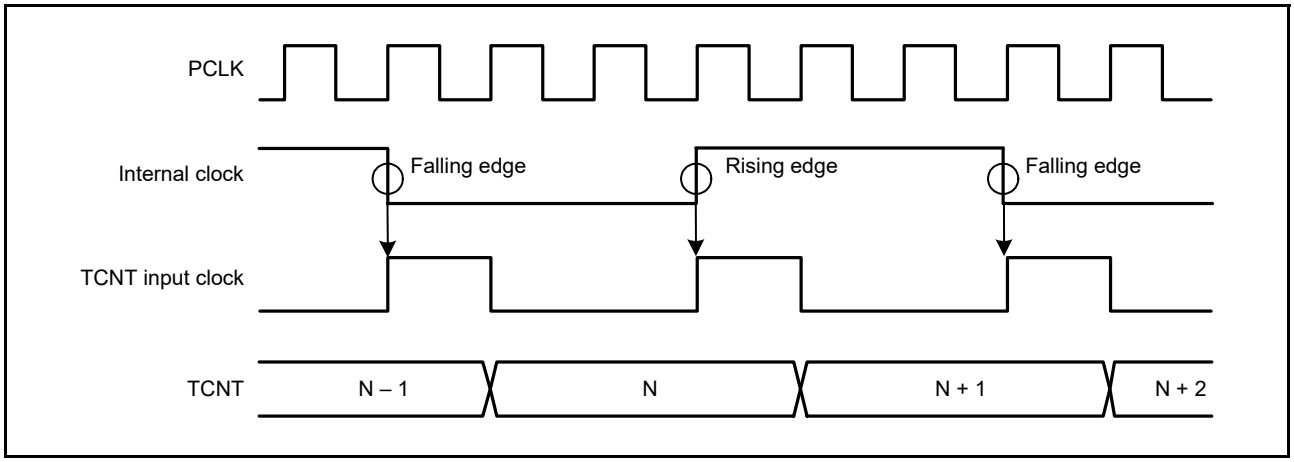


Figure 25.31 Count Timing in Internal Clock Operation

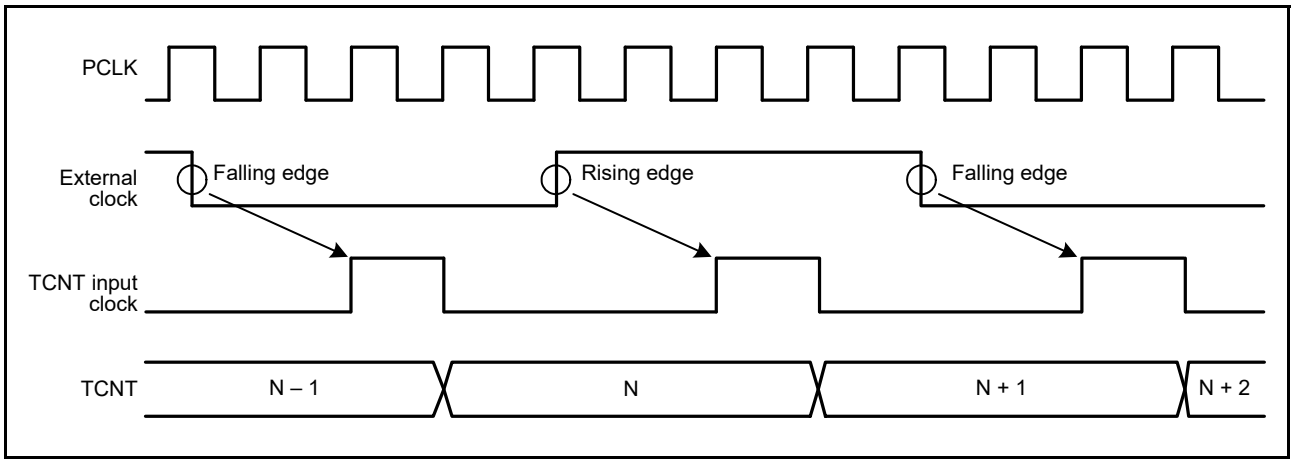


Figure 25.32 Count Timing in External Clock Operation

Downloaded from Arrow.com.

(2) Output Compare Output Timing

A compare match signal is generated in the final state in which TPUm.TCNT and TPUm.TGRy match (the point at which the count value matched by TCNT is updated). When a compare match signal is generated, the output value set in TPUm.TIORH, TPUm.TIORL, or TPUm.TIOR is output to the output compare output pin TIOCyn (y = A to D; n = 0 to 5). After a match between TCNT and TGRy, the compare match signal is not generated until the TCNT input clock is generated.

Figure 25.33 shows output compare output timing.

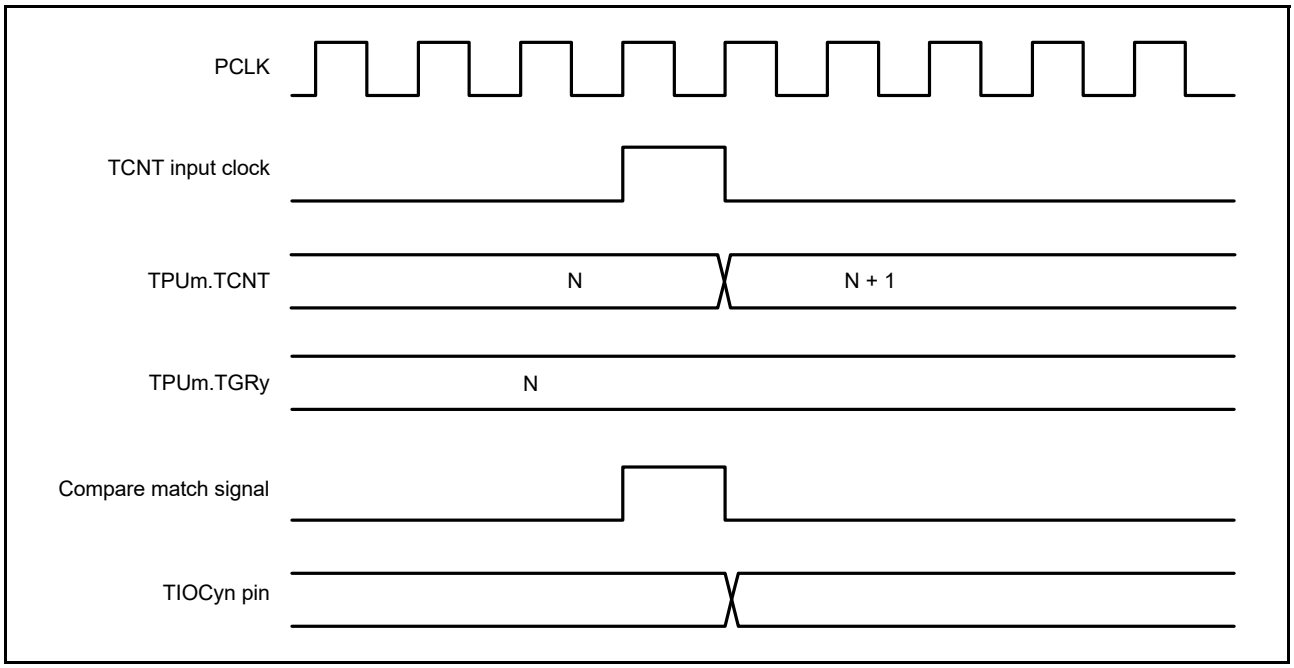


Figure 25.33 Output Compare Output Timing

(3) Input Capture Signal Timing

Figure 25.34 shows input capture signal timing.

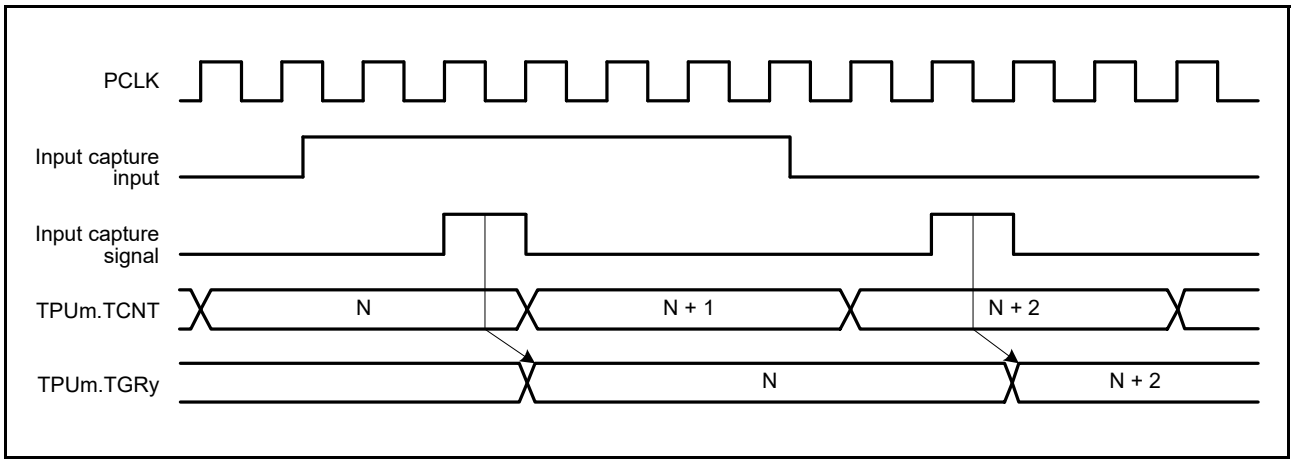


Figure 25.34 Input Capture Signal Timing

Downloaded from Arrow.com.

(4) Timing for Counter Clearing by Compare Match/Input Capture

Figure 25.35 shows the timing when counter clearing by compare match occurrence is specified, and Figure 25.36 shows the timing when counter clearing by input capture occurrence is specified.

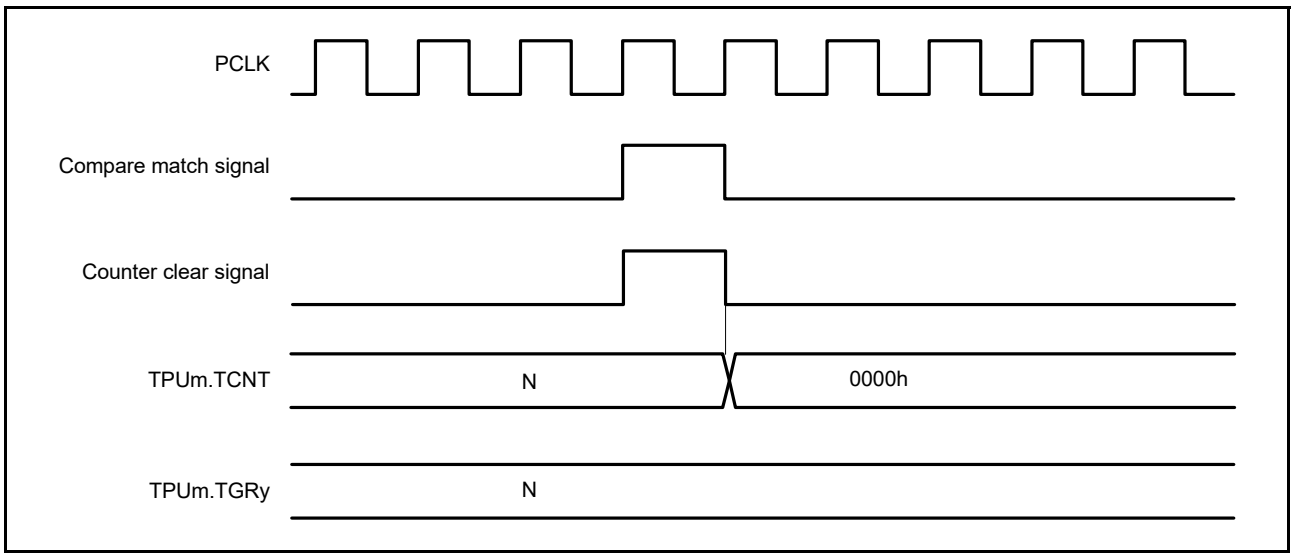


Figure 25.35 Counter Clear Timing (Compare Match)

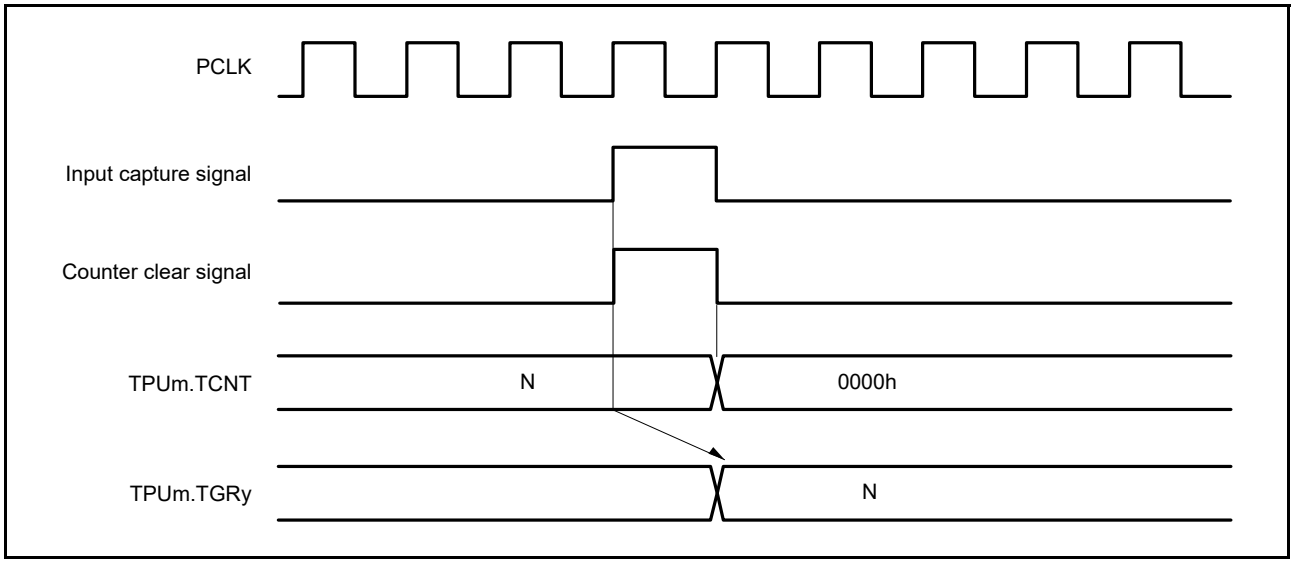


Figure 25.36 Counter Clear Timing (Input Capture)

Downloaded from Arrow.com.

(5) Buffer Operation Timing

Figure 25.37 and Figure 25.38 show the timings in buffer operation.

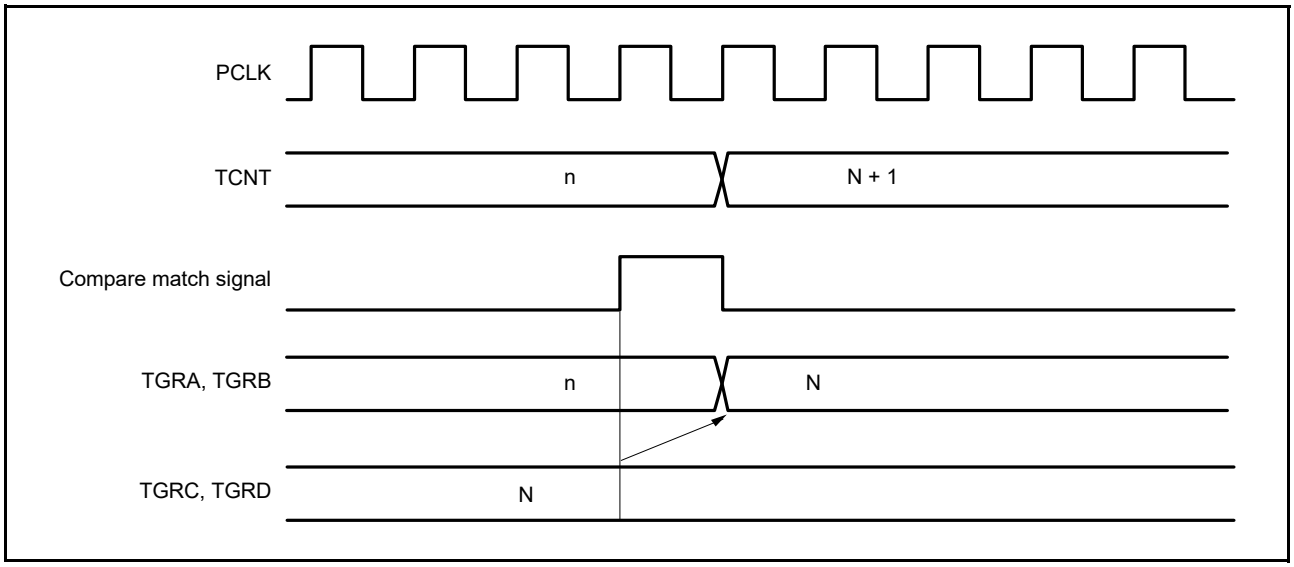


Figure 25.37 Buffer Operation Timing (Compare Match)

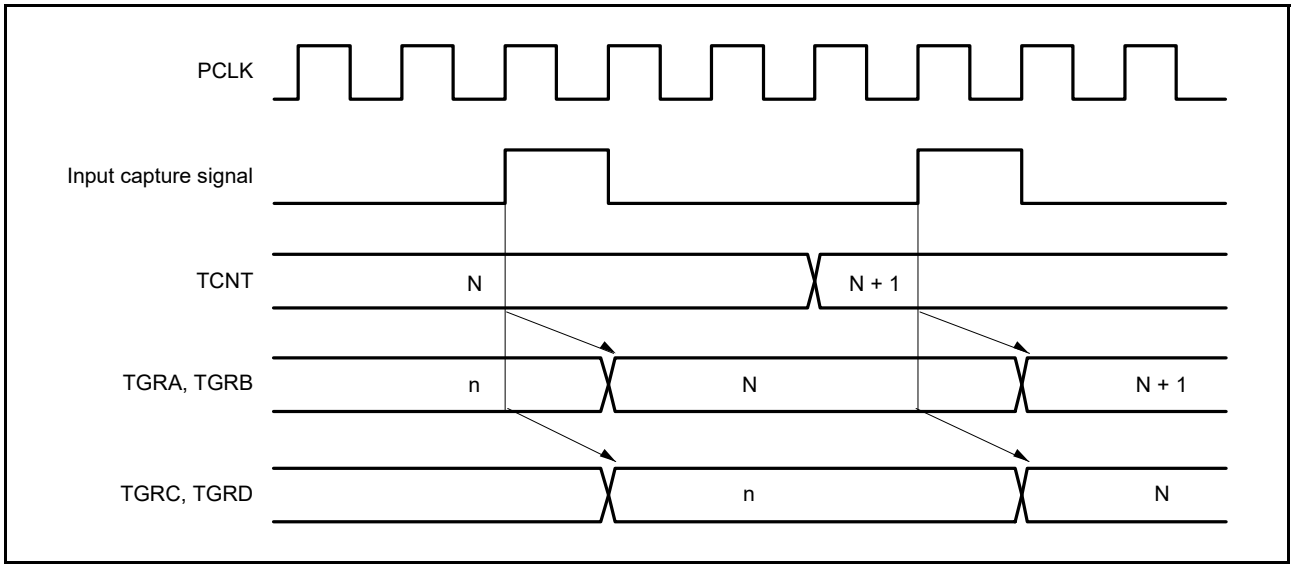


Figure 25.38 Buffer Operation Timing (Input Capture)

Downloaded from Arrow.com.

25.8.2 Interrupt Signal Timing

(1) Timing of Interrupt Signal Setting on Compare Match

Figure 25.39 shows the timing for setting the interrupt signal by compare match occurrence.

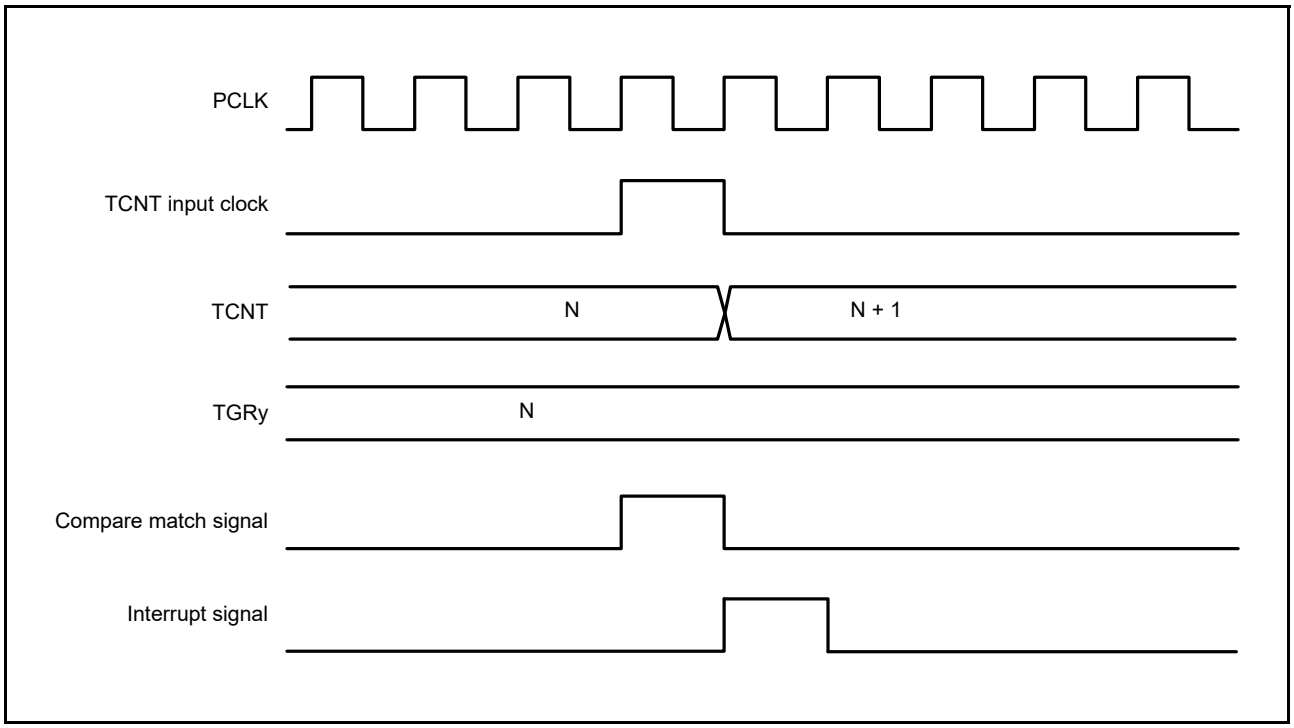


Figure 25.39 TGImy Interrupt Timing (Compare Match)

(2) Timing of Interrupt Signal Setting on Input Capture

Figure 25.40 shows the timing for setting the interrupt signal by input capture occurrence.

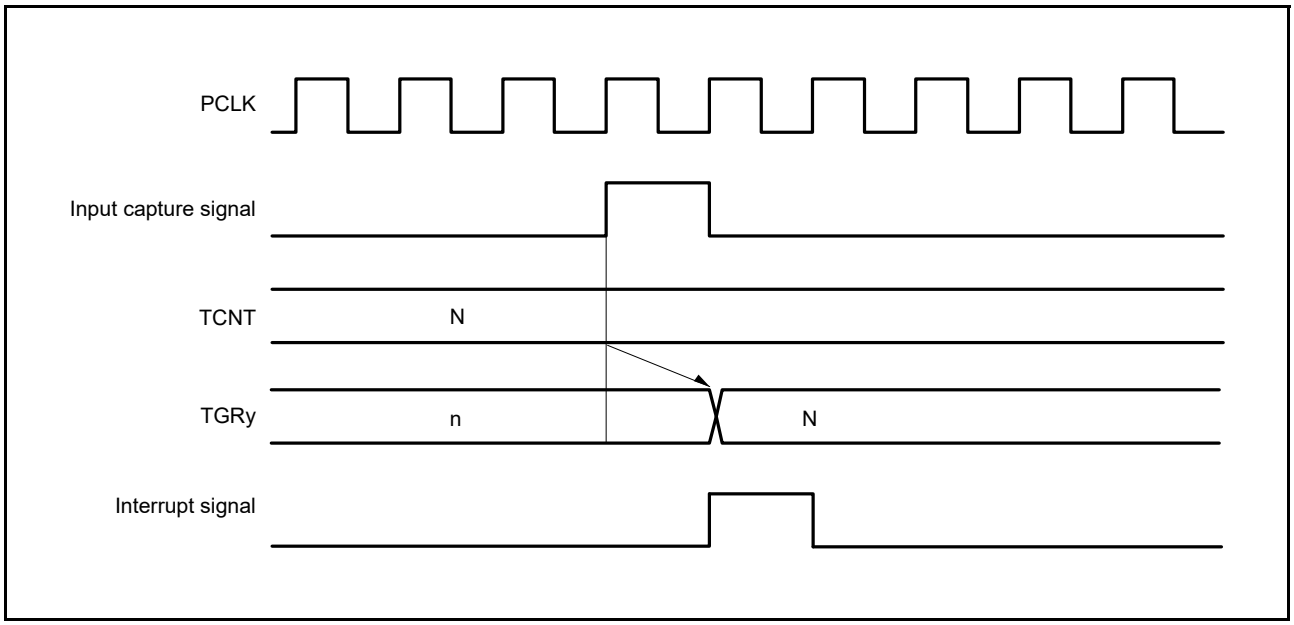


Figure 25.40 TGImy Interrupt Timing (Input Capture)

Downloaded from Arrow.com.

(3) Timing of TCImV/TCImU Interrupt Signal Setting

Figure 25.41 shows the timing for generating the TCImV interrupt signal by overflow occurrence.
Figure 25.42 shows the timing for generating the TCImU interrupt signal by underflow occurrence.

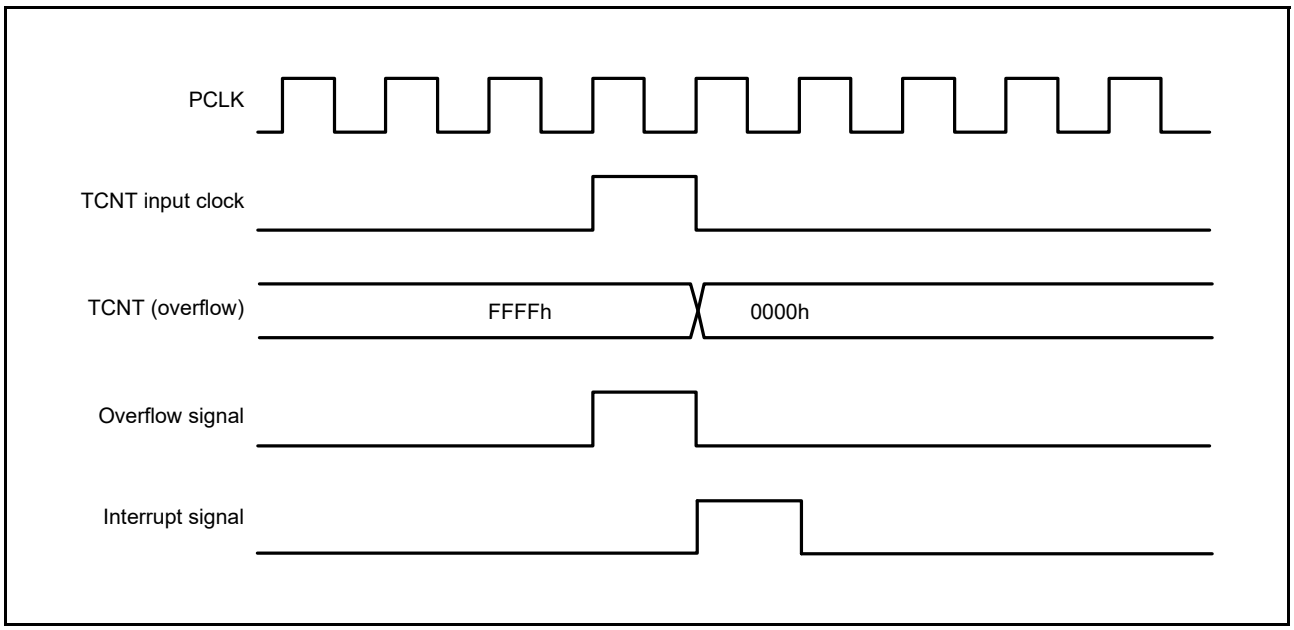


Figure 25.41 TCImV Interrupt Setting Timing

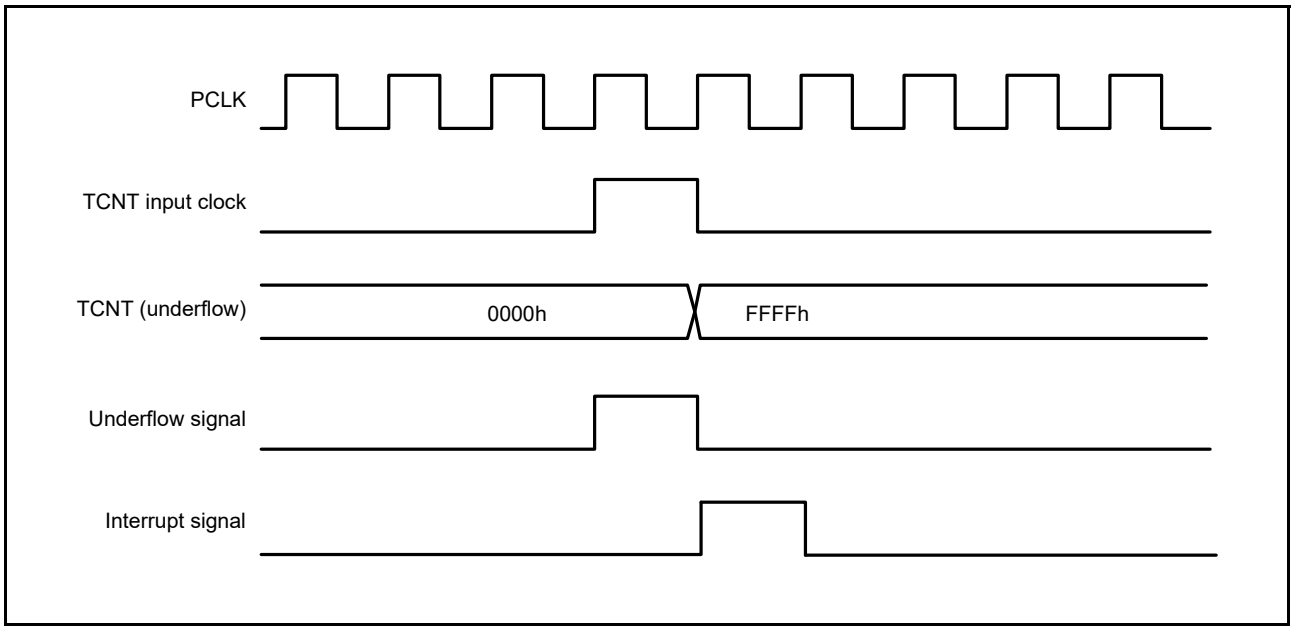


Figure 25.42 TCImU Interrupt Setting Timing

Downloaded from Arrow.com.

25.9 Usage Notes

25.9.1 Module Stop Function Setting

Operation of the TPU can be disabled or enabled using the module stop control register. The TPU does not operate with the initial setting. Register access is enabled by releasing the module stop state. For details, see section 11, Low Power Consumption.

25.9.2 Input Clock Restrictions

The input clock pulse width must be at least 1.5 PCLK cycles in the case of single-edge detection, and at least 2.5 PCLK cycles in the case of both-edge detection. The TPU will not operate properly with a narrower pulse width.
In phase counting mode, the phase difference and overlap between the two input clocks must be at least 1.5 PCLK cycles, and the pulse width must be at least 2.5 PCLK cycles. Figure 25.43 shows the input clock conditions in phase counting mode.

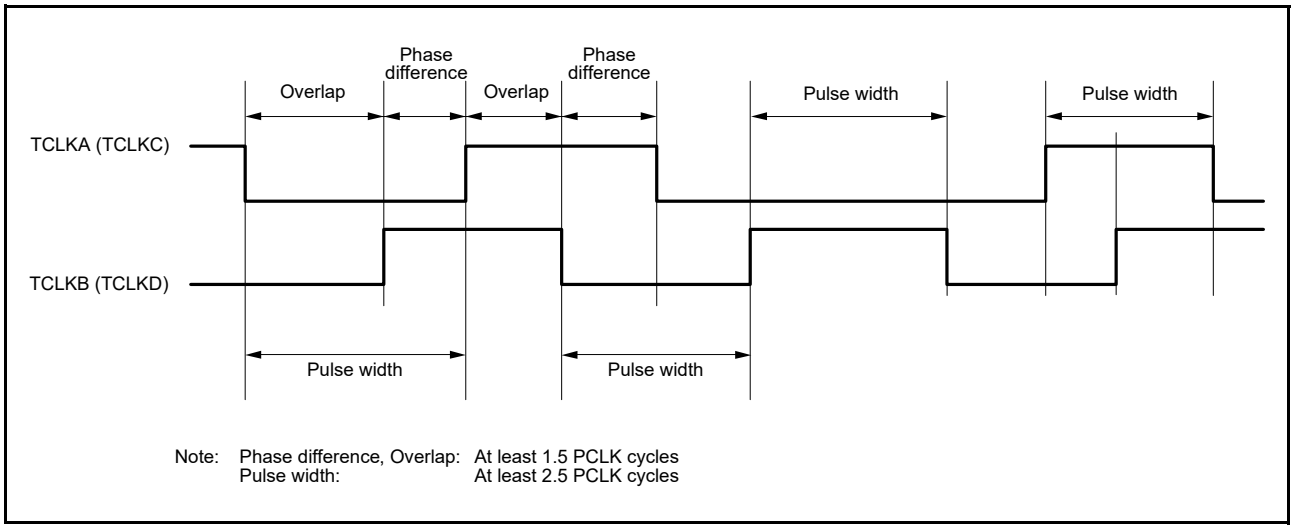


Figure 25.43 Phase Difference, Overlap, and Pulse Width in Phase Counting Mode

25.9.3 Notes on Cycle Setting

When counter clearing by compare match is set, TPUm.TCNT is cleared in the final state in which it matches the TPUm.TGRy value (the point at which the count value matched by TCNT is updated). Consequently, the actual counter frequency is given by the following formula:

$$f = \frac{f_{TCNT_CLK}}{(N + 1)}$$

f: Counter frequency
f_{TCNT_CLK}: Count clock frequency
N: TGRy set value

Downloaded from Arrow.com.

25.9.4 Conflict between TPUm.TCNT Write and Clear Operations

If the counter clearing signal is generated in a TCNT write cycle, TCNT clearing takes precedence and the TCNT write is not performed. Figure 25.44 shows the timing in this case.

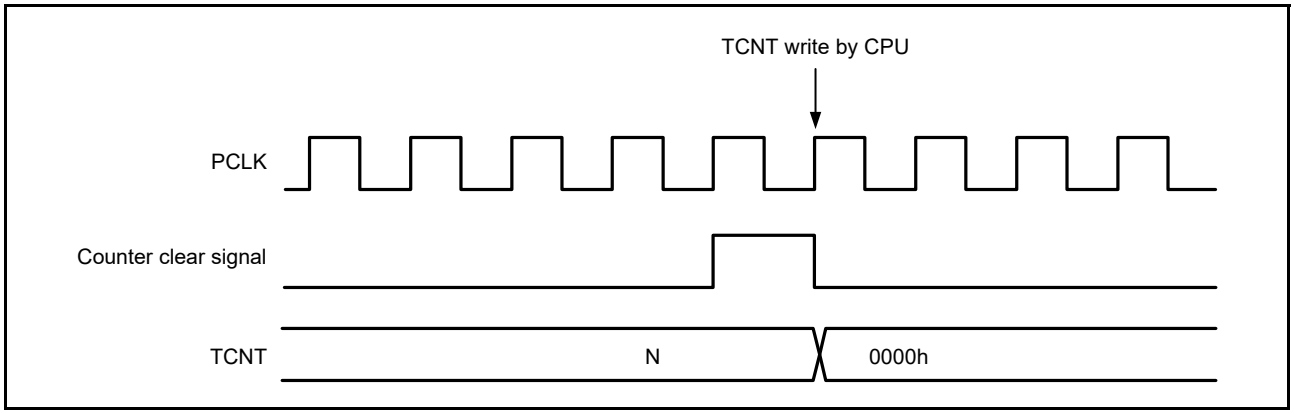


Figure 25.44 Conflict between TPUm.TCNT Write and Clear Operations

25.9.5 Conflict between TPUm.TCNT Write and Increment Operations

If incrementing occurs in a TCNT write cycle, the TCNT write takes precedence and TCNT is not incremented. Figure 25.45 shows the timing in this case.

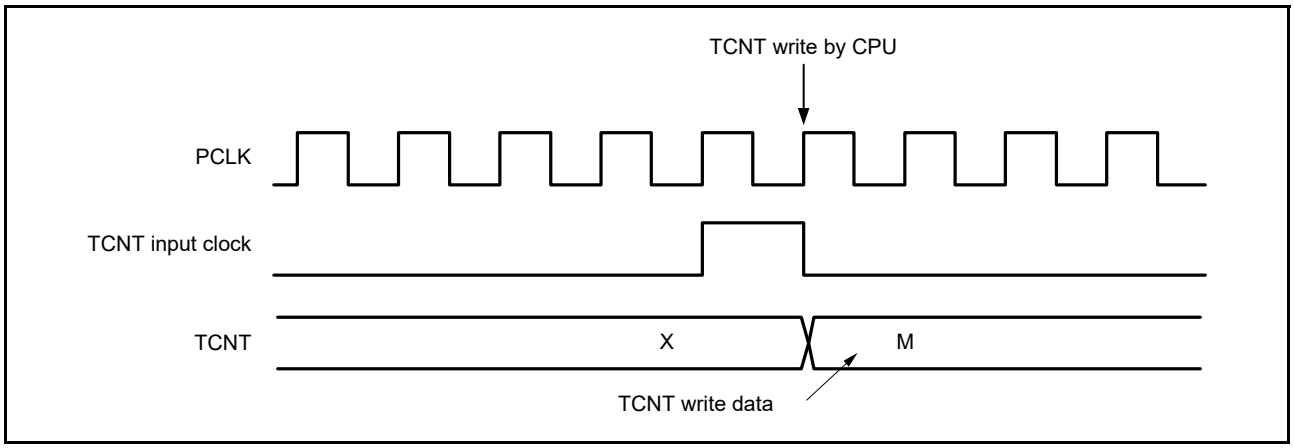


Figure 25.45 Conflict between TPUm.TCNT Write and Increment Operations

Downloaded from Arrow.com.

25.9.6 Conflict between TPUm.TGRy Write and Compare Match

If a compare match occurs in a TGRy write cycle, the TGRy write takes precedence and the compare match signal is disabled. A compare match also does not occur when the same value as before is written.
Figure 25.46 shows the timing in this case.

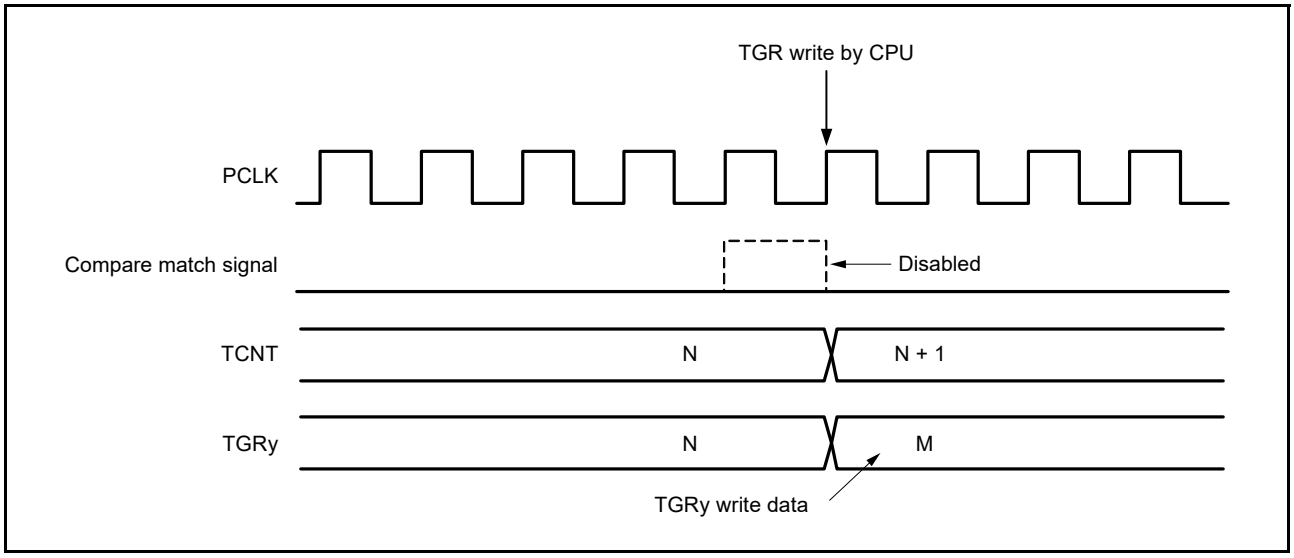


Figure 25.46 Conflict between TPUm.TGRy Write and Compare Match

25.9.7 Conflict between Buffer Register Write and Compare Match

If a compare match occurs in a TPUm.TGRy write cycle, the data transferred to TGRy by the buffer operation will be the data before writing.
Figure 25.47 shows the timing in this case.

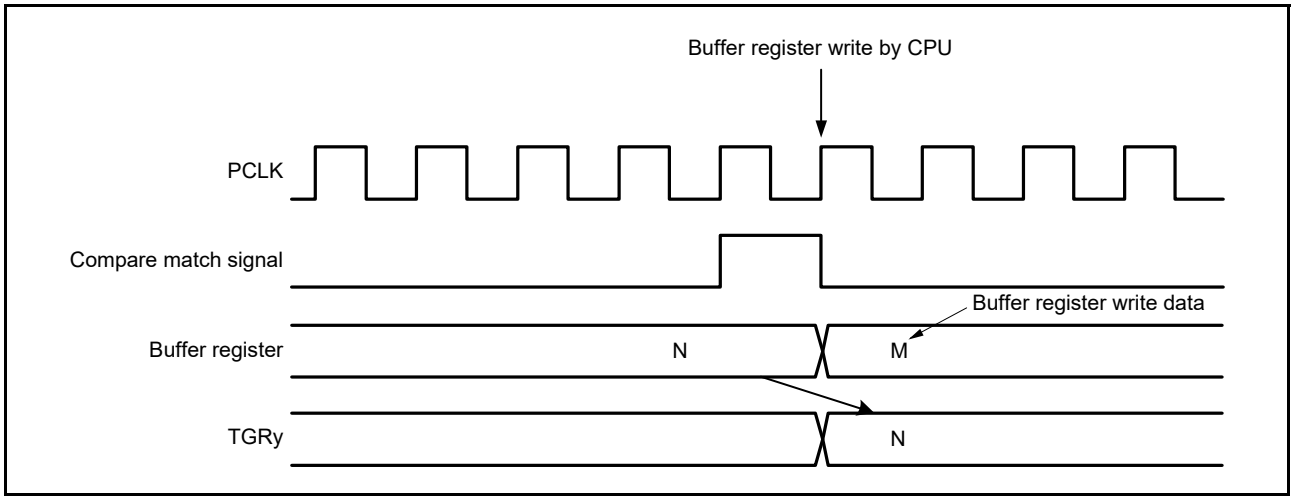


Figure 25.47 Conflict between Buffer Register Write and Compare Match

Downloaded from Arrow.com.

25.9.8 Conflict between TPUm.TGRy Read and Input Capture

If the input capture signal is generated in a TGRy read cycle, the data that is read will be the data before input capture transfer.

Figure 25.48 shows the timing in this case.

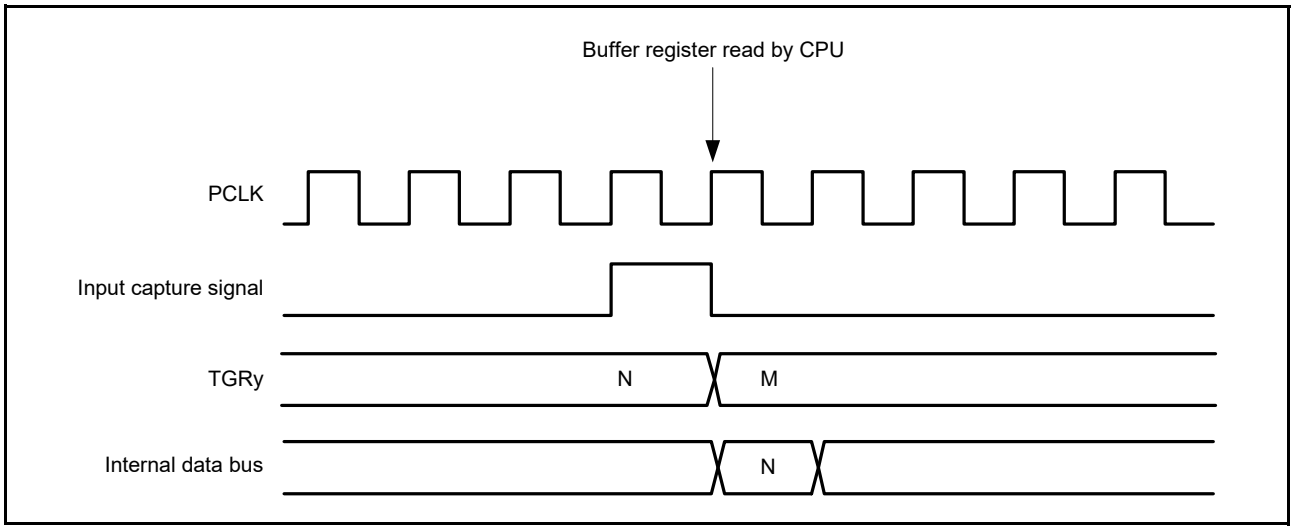


Figure 25.48 Conflict between TPUm.TGRy Read and Input Capture

25.9.9 Conflict between TPUm.TGRy Write and Input Capture

If the input capture signal is generated in a TGRy write cycle, the input capture operation takes precedence and the write to TGRy is not performed. Figure 25.49 shows the timing in this case.

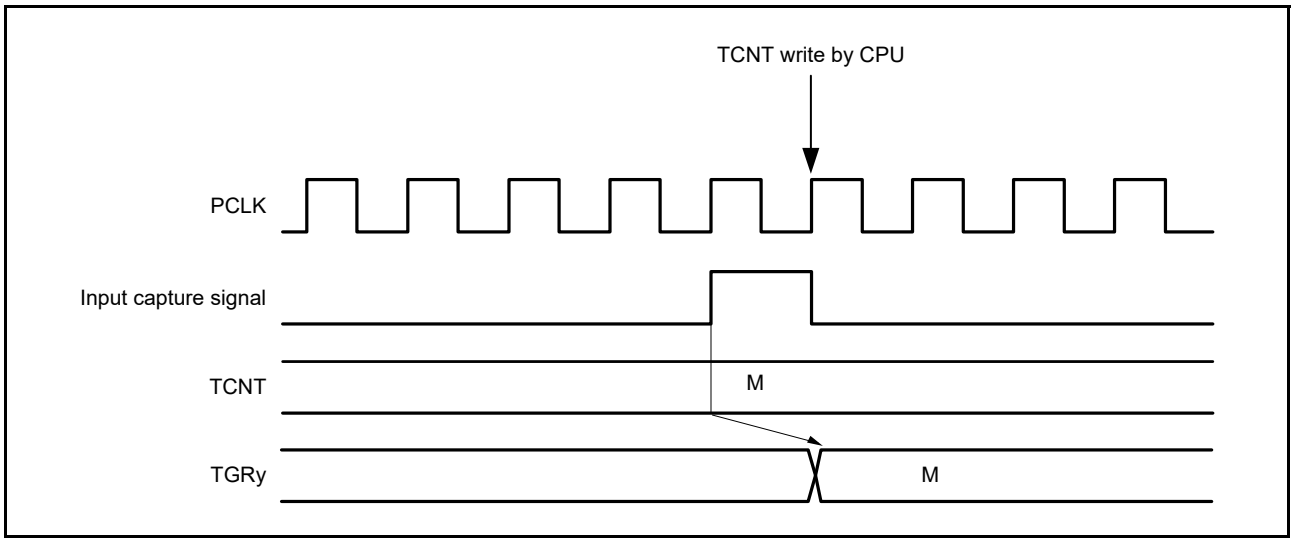


Figure 25.49 Conflict between TPUm.TGRy Write and Input Capture

Downloaded from Arrow.com.

25.9.10 Conflict between Buffer Register Write and Input Capture

If the input capture signal is generated in a buffer register write cycle, the buffer operation takes precedence and the write to the buffer register is not performed. Figure 25.50 shows the timing in this case.

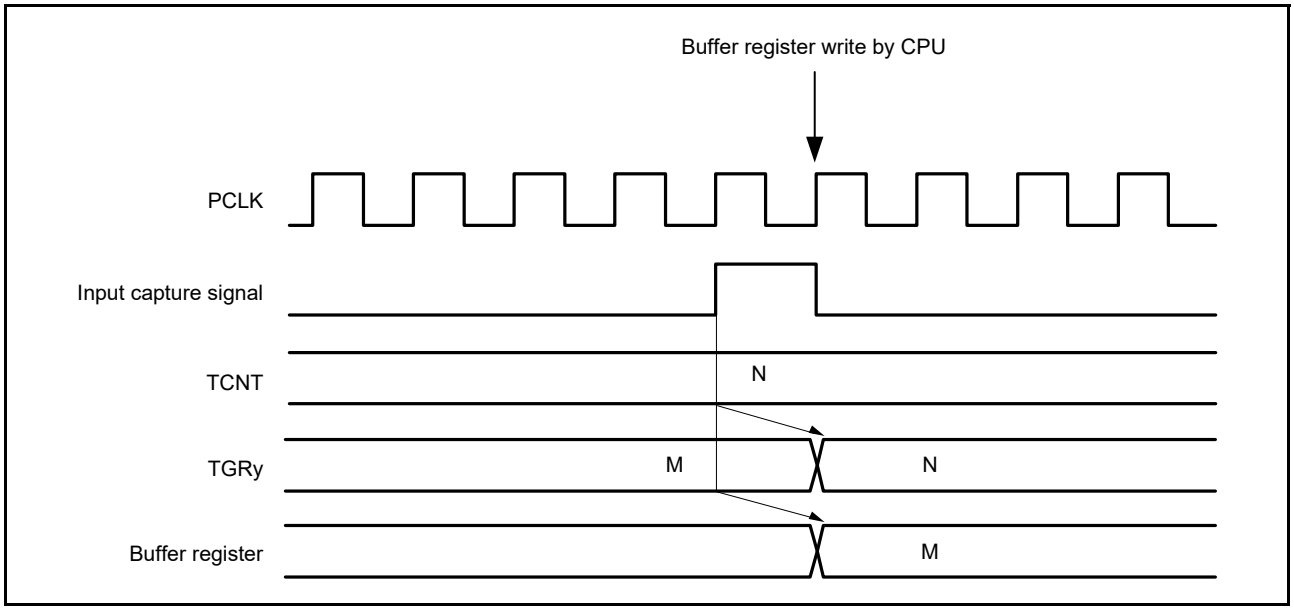


Figure 25.50 Conflict between Buffer Register Write and Input Capture

25.9.11 TCNT Simultaneous Input Capture in Cascade Operation

When TPU1.TCNT and TPU2.TCNT are cascaded for operation as a 32-bit counter, the counter value may not be captured correctly even if the input capture signal is input to the TIOC1B and TIOC2B pins at the same time. This is because a difference of up to 1 clock cycle in the timing of the capture signal to input to TPU1.TCNT and TPU2.TCNT may occur due to internal delays.

For example, the counter value is captured at an overflow of TPU2.TCNT as in counting up from 0A31 FFFFh to 0A32 0000h, the value captured may be 0A31 0000h or 0A32 FFFFh.

The same applies to cascaded operation of TPU4.TCNT and TPU5.TCNT.

Downloaded from Arrow.com.

25.9.12 Conflict between Overflow/Underflow and Counter Clearing

If overflow/underflow and counter clearing occur simultaneously, TPUm.TCNT is cleared with the generation of the compare match interrupt and an overflow interrupt is generated.

Figure 25.51 shows the operation timing when a TPUm.TGRy compare match is specified as the clearing source and FFFFh is set in TGRy.

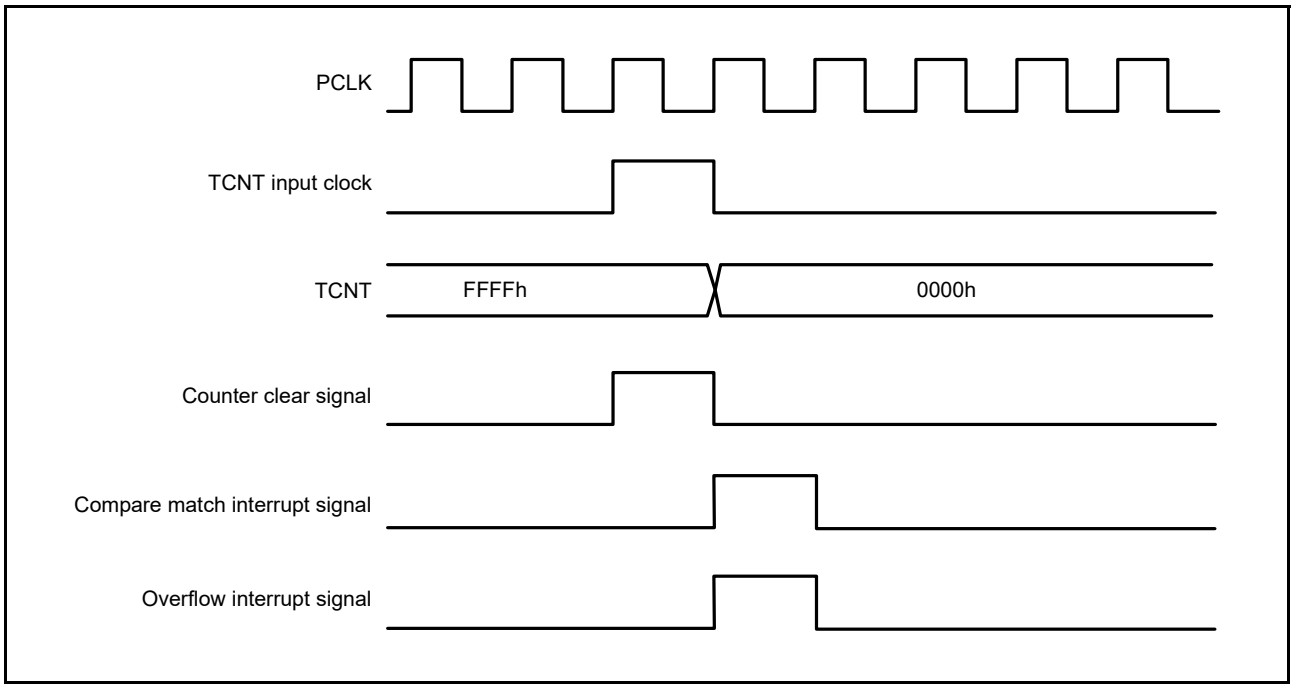


Figure 25.51 Conflict between Overflow and Counter Clearing

Downloaded from Arrow.com.

25.9.13 Conflict between TPUm.TCNT Write and Overflow/Underflow

If an overflow/underflow occurs due to increment/decrement in a TCNT write cycle, the TCNT write takes precedence. Figure 25.52 shows the operation timing when there is conflict between TCNT write and overflow.

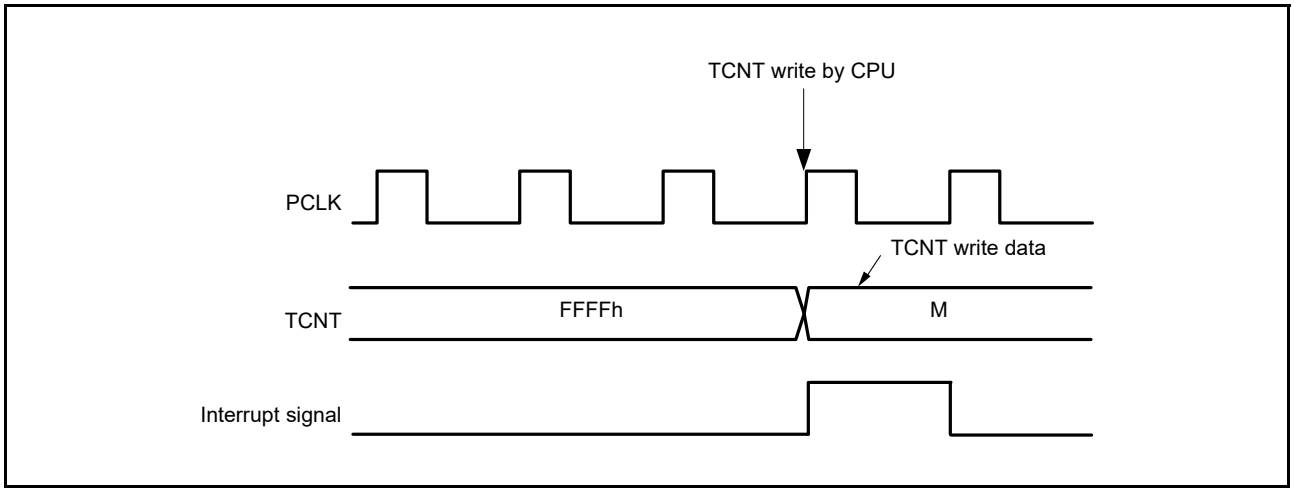


Figure 25.52 Conflict between TPUm.TCNT Write and Overflow

25.9.14 Multiplexing of I/O Pins

In this MCU, the TCLKA input pin is multiplexed with the TIOCB5 I/O pin, the TCLKB input pin with the TIOCB2 I/O pin, the TCLKC input pin with the TIOCB1 I/O pin, the TCLKD input pin with the TIOCB0 I/O pin, the TCLKC input pin with the TIOCC3 I/O pin, and the TCLKD input pin with the TIOCD3 I/O pin. When an external clock is input, compare match output should not be performed from a multiplexed pin.

Downloaded from Arrow.com.

25.9.15 Continuous Output of Compare-Match Pulse Interrupt Signal

When TGR is set to 0000h, PCLK/1 is set as the count clock, and compare match is set as the counter clear source, the TCNT remains 0000h and is not updated, and a compare-match pulse interrupt signal is output continuously to form a flat signal level.

When a pulse interrupt signal is used, the interrupt controller cannot detect the second and subsequent interrupts.

Figure 25.53 shows an operation timing when the compare-match pulse interrupt signal is continuously output.

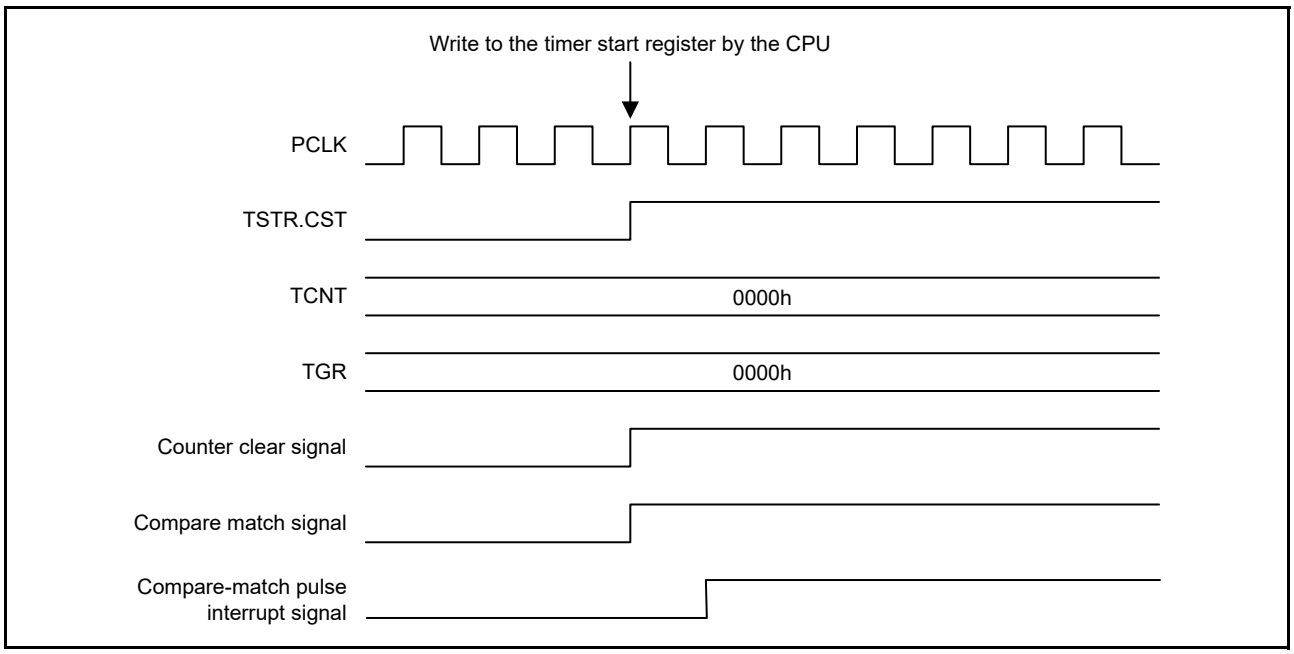


Figure 25.53 Continuous Output of Compare-Match Pulse Interrupt Signal

Downloaded from Arrow.com.

25.9.16 Continuous Output of Input-Capture Pulse Interrupt Signal

When input-capture signal is set on both edges and when the pulse width of the input-capture input equals to one PCLK cycle detected by internal sampling, input capture is generated continuously on the rising and falling edges. Therefore, an input-capture pulse interrupt signal is output continuously to form a flat signal level.
When a pulse interrupt signal is used, the interrupt controller cannot detect the second and subsequent interrupts.
Figure 25.54 shows an operation timing when the input-capture pulse interrupt signal is output continuously.

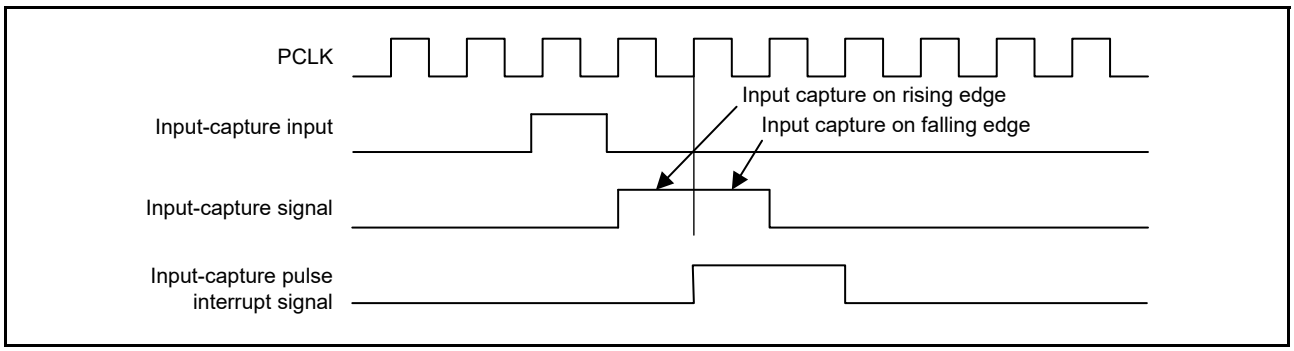


Figure 25.54 Continuous Output of Input-Capture Pulse Interrupt Signal

Downloaded from Arrow.com.

25.9.17 Continuous Output of Underflow Pulse Interrupt Signal

If two external clock signals' same direction edges to be phase counted are generated within two PCLK cycles in phase counting mode 1, with TGR being 0000h, and compare match set as the counter clear source, the TCNT remains 0000h and is not updated, and a compare-match pulse interrupt signal and an underflow interrupt signal are output continuously to form a flat signal level.

When a pulse interrupt signal is used, the interrupt controller cannot detect the second and subsequent interrupts.

Figure 25.55 shows an operation timing when the underflow pulse interrupt signal is output continuously.

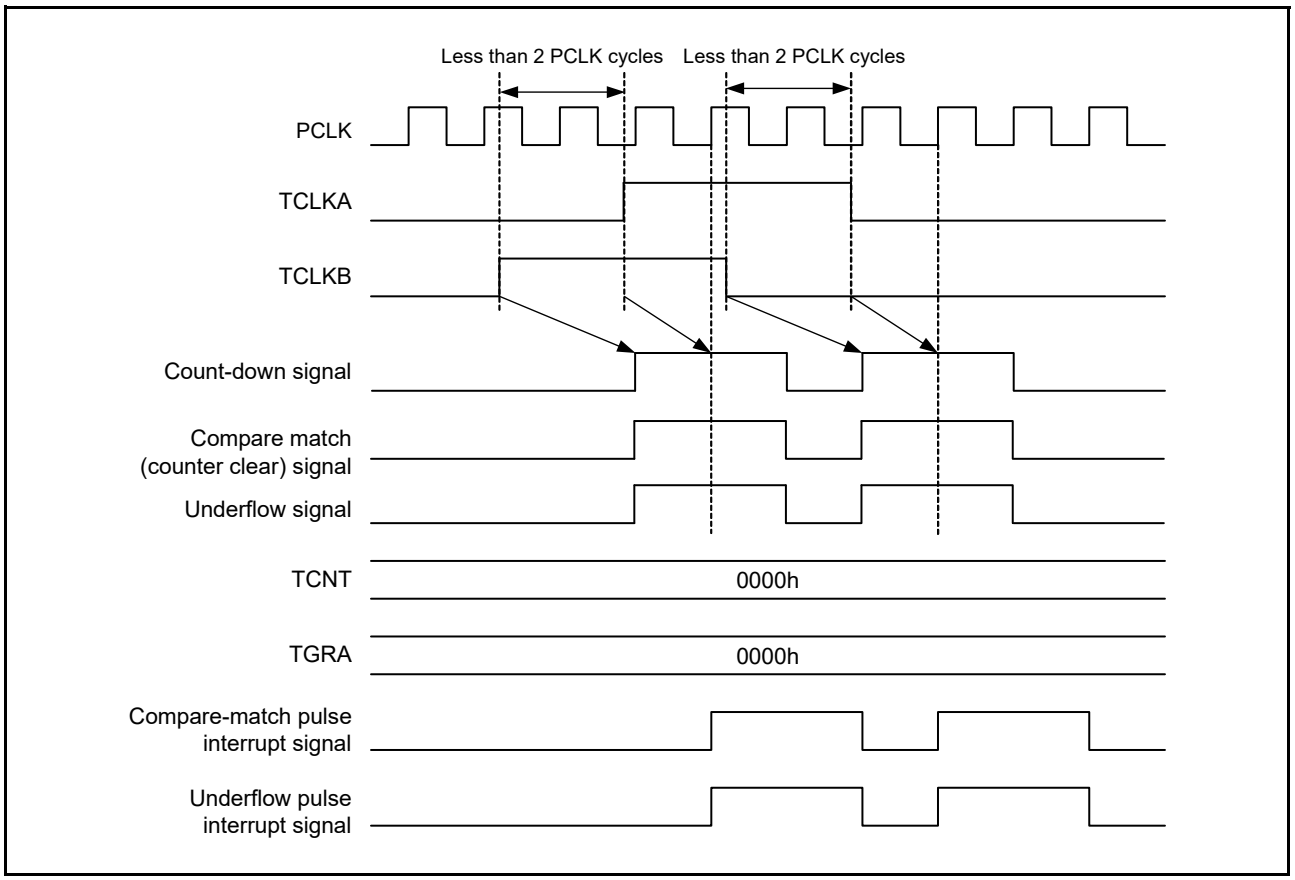


Figure 25.55 Continuous Output of Underflow Pulse Interrupt Signal

Downloaded from Arrow.com.

26. 8-Bit Timer (TMR)

This MCU has two units (unit 0, unit 1) of an on-chip 8-bit timer (TMR) module that comprise two 8-bit counter channels, totaling four channels. The 8-bit timer module can be used to count external events and also be used as a multi-function timer in a variety of applications, such as generation of counter reset signal, interrupt requests, and pulse output with a desired duty cycle using a compare-match signal with two registers.
Unit 0 and unit 1 have the same functions, and can generate a baud rate clock for the SCI.
In this section, “PCLK” is used to refer to PCLKB.

26.1 Overview

Table 26.1 lists the specifications of the TMR. Table 26.2 lists the TMR functions.
Figure 26.1 shows a block diagram of the 8-bit timer module (unit 0), and Figure 26.2 shows that of the 8-bit timer module (unit 1).

Table 26.1 Specifications of TMR

Item	Description
Count clock	• Internal clock: PCLK/1, PCLK/2, PCLK/8, PCLK/32, PCLK/64, PCLK/1024, PCLK/8192 • External clock: external count clock
Number of channels	(8 bits × 2 channels) × 2 units
Compare match	• 8-bit mode (compare match A, compare match B) • 16-bit mode (compare match A, compare match B)
Counter clear	Selected by compare match A or B, or an external counter reset signal.
Timer output	Output pulses with a desired duty cycle or PWM output
Cascading of two channels	• 16-bit count mode 16-bit timer using TMR0 for the upper 8 bits and TMR1 for the lower 8 bits (TMR2 for the upper 8 bits and TMR3 for the lower 8 bits) • Compare match count mode TMR1 can be used to count TMR0 compare matches (TMR3 can be used to count TMR2 compare matches).
Interrupt sources	Compare match A, compare match B, and overflow
Event link function (Output)	Compare match A, compare match B, and overflow (TMR0, TMR2)
Event link function (Input)	One of the following three operations proceeds in response to an event reception: (1) Counting start operation (TMR0, TMR2) (2) Event counting operation (TMR0, TMR2) (3) Counting restart operation (TMR0, TMR2)
DTC activation	DTC can be activated by compare match A interrupts or compare match B interrupts.
Capable of generating baud rate clock for SCI	Generates baud rate clock for SCI.*1
Low power consumption function	Each unit can be placed in a module stop state

Note 1. For details, refer to section 33, Serial Communications Interface (SCIg, SCIh).

Downloaded from Arrow.com.

Table 26.2 TMR Functions

Item		Unit 0			Unit 1		
Counter mode		**8 Bits**		**16 Bits**	**8 Bits**		**16 Bits**
Channel		**TMR0**	**TMR1**	**TMR0 + TMR1**	**TMR2**	**TMR3**	**TMR2 + TMR3**
Count clock		PCLK/1 PCLK/2 PCLK/8 PCLK/32 PCLK/64 PCLK/1024 PCLK/8192 TMCI0	PCLK/1 PCLK/2 PCLK/8 PCLK/32 PCLK/64 PCLK/1024 PCLK/8192 TMCI1	PCLK/1 PCLK/2 PCLK/8 PCLK/32 PCLK/64 PCLK/1024 PCLK/8192 TMCI1	PCLK/1 PCLK/2 PCLK/8 PCLK/32 PCLK/64 PCLK/1024 PCLK/8192 TMCI2	PCLK/1 PCLK/2 PCLK/8 PCLK/32 PCLK/64 PCLK/1024 PCLK/8192 TMCI3	PCLK/1 PCLK/2 PCLK/8 PCLK/32 PCLK/64 PCLK/1024 PCLK/8192 TMCI3
Counter clear		TMR0.TCORA TMR0.TCORB	TMR1.TCORA TMR1.TCORB TMRI1	TMR0.TCORA + TMR1.TCORA TMR0.TCORB + TMR1.TCORB	TMR2.TCORA TMR2.TCORB TMRI2	TMR3.TCORA TMR3.TCORB TMRI3	TMR2.TCORA + TMR3.TCORA TMR2.TCORB + TMR3.TCORB TMRI2
Compare match	Compare match A	✓	✓	✓	✓	✓	✓
	Compare match B	✓	✓	✓	✓	✓	✓
Timer output	Low output	✓	✓	✓	✓	—	✓
	High output	✓	✓	✓	✓	—	✓
	Toggle output	✓	✓	✓	✓	—	✓
DTC activation	Compare match A	✓	✓	✓	✓	✓	✓
	Compare match B	✓	✓	✓	✓	✓	✓
	TCNT overflow	—	—	—	—	—	—
Interrupt	Compare match A	CMIA0	CMIA1	CMIA0	CMIA2	CMIA3	CMIA2
	Compare match B	CMIB0	CMIB1	CMIB0	CMIB2	CMIB3	CMIB2
	TCNT overflow	OVI0	OVI1	OVI0	OVI2	OVI3	OVI2
Cascaded connection		TMR1 overflow	TMR0 compare match A	—	TMR3 overflow	TMR2 compare match A	—
SCI baud rate clock generation*1		✓		—	—		—
ELC output event	Compare match A	✓	—	✓	✓	—	✓
	Compare match B	✓	—	✓	✓	—	✓
	TCNT overflow	✓	—	✓	✓	—	✓
ELC input event	Counting start	✓	—	—	✓	—	—
	Event counting	✓	—	—	✓	—	—
	Counting restart	✓	—	—	✓	—	—
Module stop setting*2		MSTPCRA.MSTPA5 bit (unit 0), MSTPCRA.MSTPA4 bit (unit 1)					

✓: Possible
—: Impossible
Note 1. For details, refer to section 33, Serial Communications Interface (SCIg, SCIh).
Note 2. For details, refer to section 11, Low Power Consumption.

Downloaded from Arrow.com.

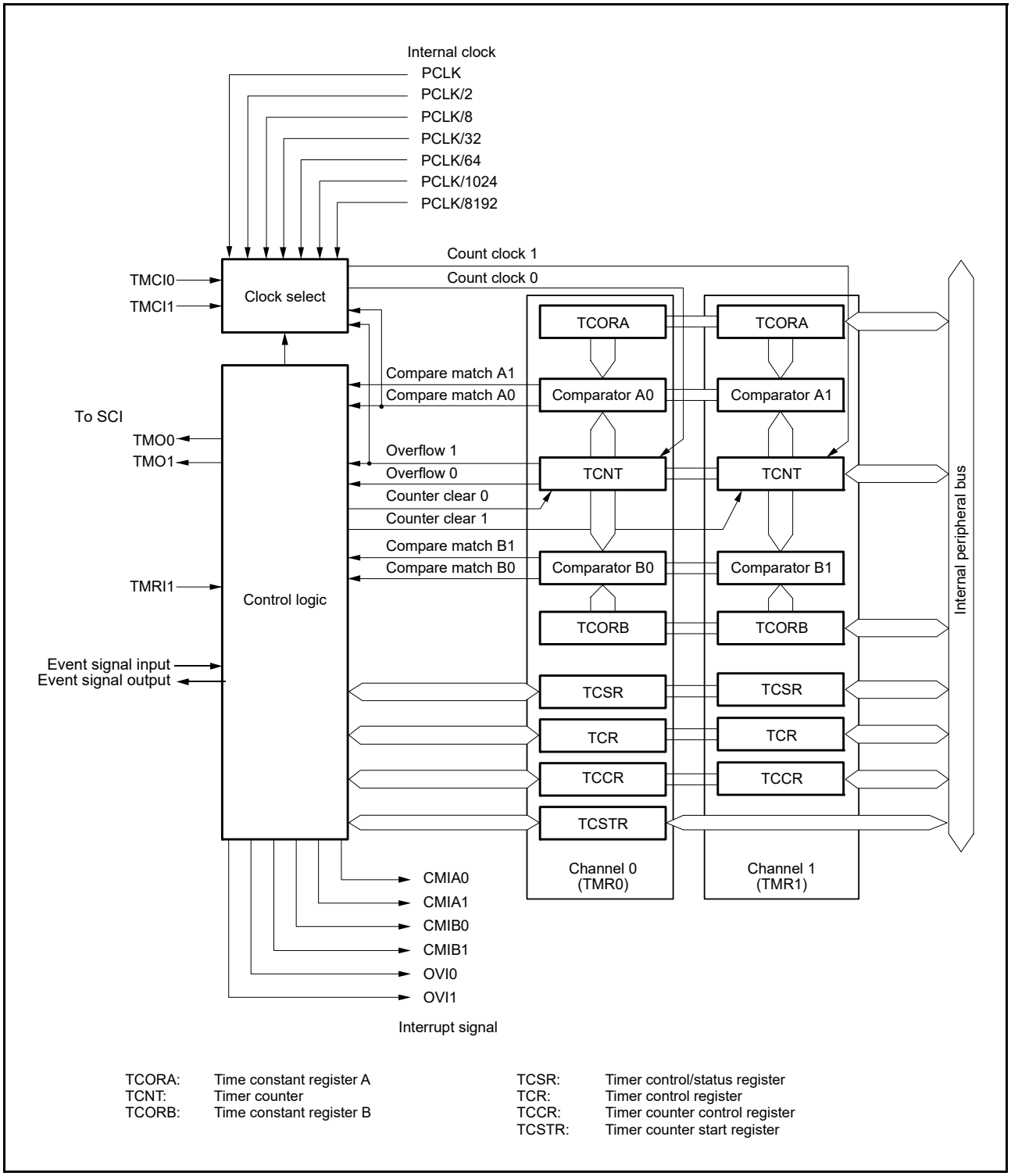


Figure 26.1 Block Diagram of TMR (Unit 0)

Downloaded from Arrow.com.

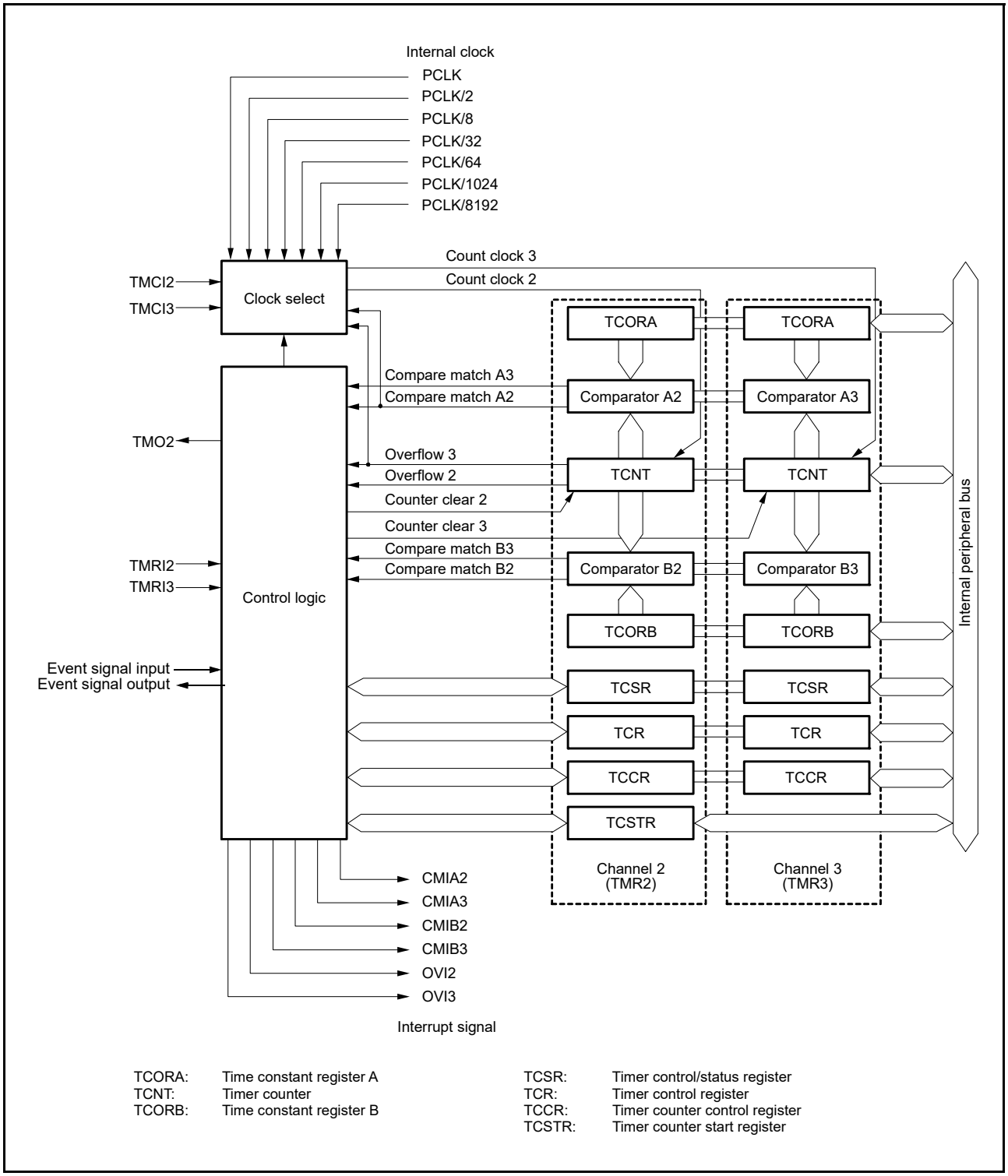


Figure 26.2 Block Diagram of TMR (Unit 1)

Downloaded from Arrow.com.

Table 26.3 lists the I/O pins of the TMR.

Table 26.3 Pin Configuration of TMR

Unit	Channel	Pin Name	I/O	Description
0	TMR0	TMO0	Output	Outputs compare match
		TMCI0	Input	Inputs external count clock
	TMR1	TMO1	Output	Outputs compare match
		TMCI1	Input	Inputs external count clock
		TMRI1	Input	Inputs external counter reset
1	TMR2	TMO2	Output	Outputs compare match
		TMCI2	Input	Inputs external count clock
		TMRI2	Input	Inputs external counter reset
	TMR3	TMCI3	Input	Inputs external count clock
		TMRI3	Input	Inputs external counter reset

Downloaded from Arrow.com.

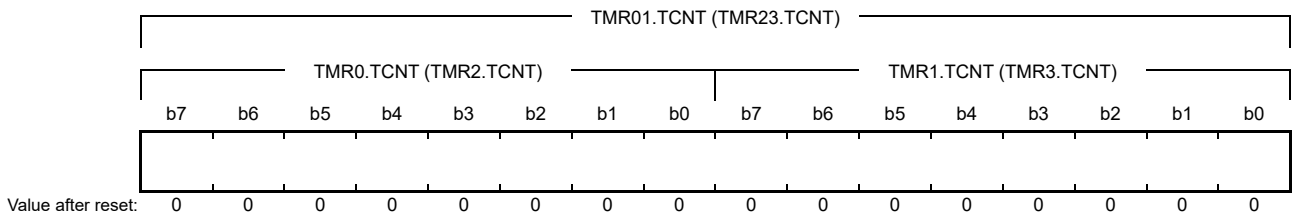
26.2 Register Descriptions

Table 26.4 Register Allocation for 16-Bit Access

Address	Register	Upper 8 Bits	Lower 8 Bits
0008 8208h	TMR01.TCNT	TMR0.TCNT	TMR1.TCNT
0008 8204h	TMR01.TCORA	TMR0.TCORA	TMR1.TCORA
0008 8206h	TMR01.TCORB	TMR0.TCORB	TMR1.TCORB
0008 820Ah	TMR01.TCCR	TMR0.TCCR	TMR1.TCCR
0008 8218h	TMR23.TCNT	TMR2.TCNT	TMR3.TCNT
0008 8214h	TMR23.TCORA	TMR2.TCORA	TMR3.TCORA
0008 8216h	TMR23.TCORB	TMR2.TCORB	TMR3.TCORB
0008 821Ah	TMR23.TCCR	TMR2.TCCR	TMR3.TCCR

26.2.1 Timer Counter (TCNT)

Address(es): TMR0.TCNT 0008 8208h, TMR1.TCNT 0008 8209h, TMR2.TCNT 0008 8218h, TMR3.TCNT 0008 8219h, TMR01.TCNT 0008 8208h, TMR23.TCNT 0008 8218h



TCNT is an 8-bit readable/writable up-counter.

TMR0.TCNT and TMR1.TCNT (TMR2.TCNT and TMR3.TCNT) comprise a single 16-bit counter (TMR01.TCNT, TMR23.TCNT) so they can be accessed together by a word transfer instruction.

The TCCR.CSS[1:0] and CKS[2:0] bits are used to select a count clock.

TCNT can be cleared by an external counter reset signal, compare match A, or compare match B. Which compare match to be used for clearing is selected by the TCR.CCLR[1:0] bits.

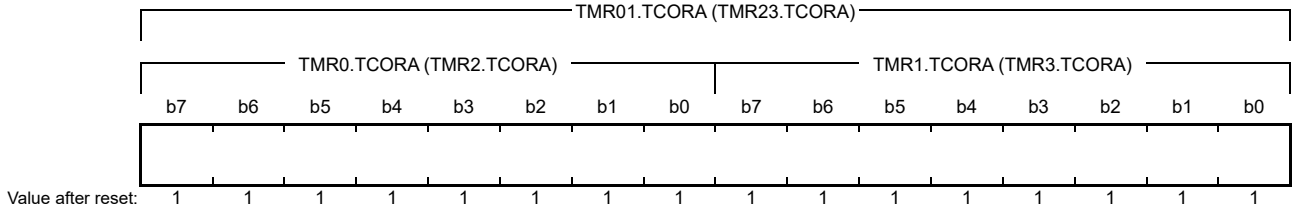
When TCNT overflows (its value changes from FFh to 00h), an overflow interrupt (low-level pulse) is output provided the interrupt request is enabled by the TCR.OVIE bit.

For details on the corresponding interrupt vector number, refer to section 15, Interrupt Controller (ICUb), and Table 26.6, TMR Interrupt Sources.

Downloaded from Arrow.com.

26.2.2 Time Constant Register A (TCORA)

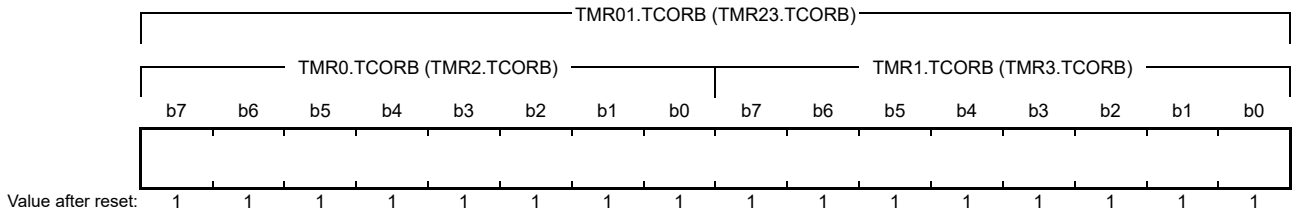
Address(es): TMR0.TCORA 0008 8204h, TMR1.TCORA 0008 8205h, TMR2.TCORA 0008 8214h, TMR3.TCORA 0008 8215h, TMR01.TCORA 0008 8204h, TMR23.TCORA 0008 8214h



TCORA is an 8-bit readable/writable register.
TMR0.TCORA and TMR1.TCORA (TMR2.TCORA and TMR3.TCORA) comprise a single 16-bit register (TMR01.TCORA, TMR23.TCORA) so they can be accessed together by a word transfer instruction.
The value in TCORA is continually compared with the value in TCNT. When a match is detected, the corresponding compare match A is generated, and a compare match A interrupt (low-level pulse) is output provided the interrupt request is enabled by the TCR.CMIEA bit.
However, comparison is not performed during writing to TCORA. The timer output from the TMOn pin can be freely controlled by this compare match A and the settings of the TCSR.OSA[1:0] bits.

26.2.3 Time Constant Register B (TCORB)

Address(es): TMR0.TCORB 0008 8206h, TMR1.TCORB 0008 8207h, TMR2.TCORB 0008 8216h, TMR3.TCORB 0008 8217h, TMR01.TCORB 0008 8206h, TMR23.TCORB 0008 8216h



TCORB is an 8-bit readable/writable register.
TMR0.TCORB and TMR1.TCORB (TMR2.TCORB and TMR3.TCORB) comprise a single 16-bit register (TMR01.TCORB, TMR23.TCORB) so they can be accessed together by a word transfer instruction.
The value in TCORB is continually compared with the value in TCNT. When a match is detected, the corresponding compare match B is generated, and a compare match B interrupt (low-level pulse) is output provided the interrupt request is enabled by the TCR.CMIEB bit.
However, comparison is not performed during writing to TCORB. The timer output from the TMOn pin can be freely controlled by this compare match B and the settings of the TCSR.OSB[1:0] bits.

Downloaded from Arrow.com.

26.2.4 Timer Control Register (TCR)

Address(es): TMR0.TCR 0008 8200h, TMR1.TCR 0008 8201h, TMR2.TCR 0008 8210h, TMR3.TCR 0008 8211h

	b7	b6	b5	b4	b3	b2	b1	b0
	CMIEB	CMIEA	OVIE	CCLR[1:0]		—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4, b3	CCLR[1:0]	Counter Clear*1	b4 b3 0 0: Clearing is disabled 0 1: Cleared by compare match A 1 0: Cleared by compare match B 1 1: Cleared by the external counter reset signal*2 (Select edge or level by the TMRIS bit in TCCR.)	R/W
b5	OVIE	Timer Overflow Interrupt Enable	0: Overflow interrupt requests (OVIn) are disabled 1: Overflow interrupt requests (OVIn) are enabled	R/W
b6	CMIEA	Compare Match Interrupt Enable A	0: Compare match A interrupt requests (CMIAn) are disabled 1: Compare match A interrupt requests (CMIAn) are enabled	R/W
b7	CMIEB	Compare Match Interrupt Enable B	0: Compare match B interrupt requests (CMIBn) are disabled 1: Compare match B interrupt requests (CMIBn) are enabled	R/W

Note 1. To use an external counter reset signal, set the corresponding pin function. For details, refer to section 21, I/O Ports and section 22, Multi-Function Pin Controller (MPC).
Note 2. Do not choose this option for TMR0.

CCLR[1:0] Bits (Counter Clear)

Select the condition by which TCNT is cleared.

OVIE Bit (Timer Overflow Interrupt Enable)

Selects whether overflow interrupt requests (OVIn) issued by TCNT are enabled or disabled.

CMIEA Bit (Compare Match Interrupt Enable A)

Selects whether compare match A interrupt requests (CMIAn) that are issued when the value of TCORA corresponds to that of TCNT are enabled or disabled.

CMIEB Bit (Compare Match Interrupt Enable B)

Selects whether compare match B interrupt requests (CMIBn) that are issued when the value of TCORB corresponds to that of TCNT are enabled or disabled.

Downloaded from Arrow.com.

26.2.5 Timer Counter Control Register (TCCR)

Address(es): TMR0.TCCR 0008 820Ah, TMR1.TCCR 0008 820Bh, TMR2.TCCR 0008 821Ah, TMR3.TCCR 0008 821Bh, TMR01.TCCR 0008 820Ah, TMR23.TCCR 0008 821Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	TMRIS	—	—	CSS[1:0]		CKS[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	CKS[2:0]	Clock Select*1	See Table 26.5.	R/W
b4, b3	CSS[1:0]	Clock Source Select	See Table 26.5.	R/W
b6, b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	TMRIS	Timer Reset Detection Condition Select*2	0: Cleared at rising edge of the external counter reset signal 1: Cleared when the external counter reset signal is high	R/W

Note 1. To use an external count clock, set the corresponding pin function. For details, refer to section 21, I/O Ports and section 22, Multi-Function Pin Controller (MPC).
Note 2. This bit is reserved in TMR0. The bit is read as 0. Writing to these bits is not possible.

TCCR register is a 8-bit register used to configure the basic operation of the counter. Two TCCR registers can be accessed simultaneously by accessing the address of the even channel TCCR register in 16-bit units.

CKS[2:0] Bits (Clock Select)
CSS[1:0] Bits (Clock Source Select)

The CKS[2:0] and CSS[1:0] bits select a clock. For details, see Table 26.5.

TMRIS Bit (Timer Reset Detection Condition Select)

This bit is enabled when the TCR.CCLR[1:0] bits are 11b (cleared by external counter reset signal) and selects the condition for detecting counter reset (level or edge).

Downloaded from Arrow.com.

Table 26.5 Clock Input to TCNT and Count Condition

Channel	TCCR Register CSS[1:0] b4	CSS[1:0] b3	CKS[2:0] b2	CKS[2:0] b1	CKS[2:0] b0	Description
TMR0 (TMR2)	0	0	—	0	0	Clock input prohibited
					1	Uses external count clock. Counts at rising edge*1.
				1	0	Uses external count clock. Counts at falling edge*1.
					1	Uses external count clock. Counts at both rising and falling edges*1.
	0	1	0	0	0	Uses internal clock. Counts at PCLK.
					1	Uses internal clock. Counts at PCLK/2.
				1	0	Uses internal clock. Counts at PCLK/8.
					1	Uses internal clock. Counts at PCLK/32.
			1	0	0	Uses internal clock. Counts at PCLK/64.
					1	Uses internal clock. Counts at PCLK/1024.
				1	0	Uses internal clock. Counts at PCLK/8192.
					1	Clock input prohibited
	1	0	—	—	—	Setting prohibited
	1	1	—	—	—	Counts at TMR1.TCNT (TMR3.TCNT) overflow signal*2.
TMR1 (TMR3)	0	0	—	0	0	Clock input prohibited
					1	Uses external count clock. Counts at rising edge*1.
				1	0	Uses external count clock. Counts at falling edge*1.
					1	Uses external count clock. Counts at both rising and falling edges*1.
	0	1	0	0	0	Uses internal clock. Counts at PCLK.
					1	Uses internal clock. Counts at PCLK/2.
				1	0	Uses internal clock. Counts at PCLK/8.
					1	Uses internal clock. Counts at PCLK/32.
			1	0	0	Uses internal clock. Counts at PCLK/64.
					1	Uses internal clock. Counts at PCLK/1024.
				1	0	Uses internal clock. Counts at PCLK/8192.
					1	Clock input prohibited
	1	0	—	—	—	Setting prohibited
	1	1	—	—	—	Counts at TMR0.TCNT (TMR2.TCNT) compare match A*2.

Note 1. To use an external count clock, set the corresponding pin function. For details, refer to section 21, I/O Ports and section 22, Multi-Function Pin Controller (MPC).

Note 2. If the clock input of TMR0 (TMR2) is the overflow signal of the TMR1.TCNT (TMR3.TCNT) counter and that of TMR1 (TMR3) is the compare match signal of the TMR0.TCNT (TMR2.TCNT) counter, no TCNT count clock is generated. Do not use this setting.

Downloaded from Arrow.com.

26.2.6 Timer Control/Status Register (TCSR)

- TMR0.TCSR, TMR2.TCSR

Address(es): TMR0.TCSR 0008 8202h, TMR2.TCSR 0008 8212h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	OSB[1:0]		OSA[1:0]	
Value after reset:	x	x	x	0	0	0	0	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b1, b0	OSA[1:0]	Output Select A*1	b1 b0 0 0: No change 0 1: Low is output 1 0: High is output 1 1: Output is inverted (toggle output)	R/W
b3, b2	OSB[1:0]	Output Select B*1	b3 b2 0 0: No change 0 1: Low is output 1 0: High is output 1 1: Output is inverted (toggle output)	R/W
b4	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7 to b5	—	Reserved	These bits are read as an undefined value. The write value should be 1.	R/W

Note 1. When all OSA[1:0] and OSB[1:0] bits are 0, the output enable signal corresponding to the TMOn pin is negated and a request for high-impedance output is issued to the I/O port. Timer output pin is driven low until the first compare-match occurs after a reset when at least one of the OSA[1:0] and OSB[1:0] bits is 1.

OSA[1:0] Bits (Output Select A)

These bits select a method of TMOn pin output when compare match A of TCORA and TCNT occurs.

OSB[1:0] Bits (Output Select B)

These bits select a method of TMOn pin output when compare match B of TCORB and TCNT occurs.

Downloaded from Arrow.com.

- TMR1.TCSR

Address(es): TMR1.TCSR 0008 8203h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	OSB[1:0]		OSA[1:0]	
Value after reset:	x	x	x	1	0	0	0	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b1, b0	OSA[1:0]	Output Select A*1	b1 b0 0 0: No change 0 1: Low is output 1 0: High is output 1 1: Output is inverted (toggle output)	R/W
b3, b2	OSB[1:0]	Output Select B*1	b3 b2 0 0: No change 0 1: Low is output 1 0: High is output 1 1: Output is inverted (toggle output)	R/W
b4	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b7 to b5	—	Reserved	These bits are read as an undefined value. The write value should be 1.	R/W

Note 1. When all OSA[1:0] and OSB[1:0] bits are 0, the output enable signal corresponding to the TMOn pin is negated and a request for high-impedance output is issued to the I/O port. Timer output pin is driven low until the first compare match occurs after a reset when at least one of the OSA[1:0] and OSB[1:0] bits is 1.

OSA[1:0] Bits (Output Select A)

These bits select a method of TMOn pin output when compare match A of TCORA and TCNT occurs.

OSB[1:0] Bits (Output Select B)

These bits select a method of TMOn pin output when compare match B of TCORB and TCNT occurs.

Downloaded from Arrow.com.

26.2.7 Timer Counter Start Register (TCSTR)

Address(es): TMR0.TCSTR 0008 820Ch, TMR2.TCSTR 0008 821Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	TCS
Value after reset:	x	x	x	x	x	x	x	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	TCS	Timer Counter Status	0: Count stopped state in response to ELC. 1: Count start state in response to ELC.	R/W
b7 to b1	—	Reserved	These bits are read as an undefined value. The write value should be 0.	R/W

TCS Bit (Timer Counter Status)

The TCS bit is used to check the state of the timer count in response to ELC.

When this bit is read as 1, it shows the timer start state in response to ELC. When this bit is read as 0, it shows the timer stopped state in response to ELC.

This bit is cleared by writing 0. Do not write 1 to this bit.

The TCS bit is valid only when the count start operation is selected by the ELOPD register of the event controller (ELC). For details, refer to section 26.7, Link Operation by ELC, or section 20, Event Link Controller (ELC).

Downloaded from Arrow.com.

26.3 Operation

26.3.1 Pulse Output

Figure 26.3 shows an example of the 8-bit timer being used to generate a pulse output with a desired duty cycle.

1. Set the TCR.CCLR[1:0] bits to 01b (cleared by compare match A) so that TCNT is cleared at a compare match of TCORA.
2. Set the TCSR.OSA[1:0] bits to 10b (high is output) and TCSR.OSB[1:0] bits to 01b (low is output), causing the output to change to high at a compare match of TCORA and to low at a compare match of TCORB.

With these settings, the 8-bit timer provides pulses output at a cycle determined by TCORA with a pulse width determined by TCORB. No software intervention is required.
The timer output pin is low after the TCSR.OSA[1:0] or TCSR.OSB[1:0] bits are set until the first compare match occurs after a reset.

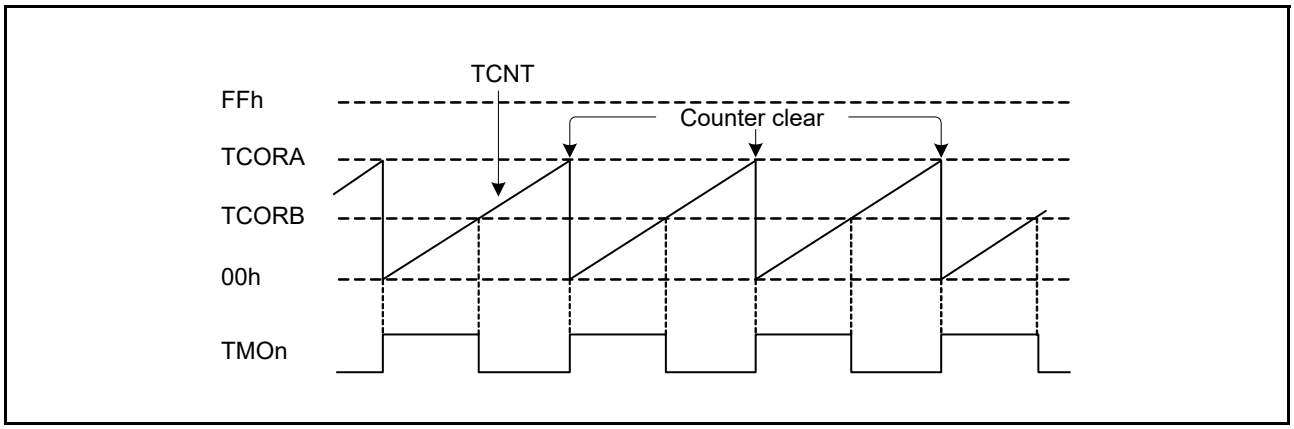


Figure 26.3 Example of Pulse Output (n = 0 to 2)

Downloaded from Arrow.com.

26.3.2 External Counter Reset Input

Figure 26.4 shows an example of the 8-bit timer being used to generate a pulse which is output after a desired delay time from a TMRIn input.

1. Set the TCR.CCLR[1:0] bits to 11b (cleared by external counter reset signal) and set the TMRIS bit in TCCR to 1 (cleared when the external counter reset signal is high) so that TCNT is cleared at the high level input of the TMRIn signal.
2. Set the TCSR.OSA[1:0] bits to 10b (high output) and the TCSR.OSB[1:0] bits to 01b (low output), causing the output to change to high at a compare match of TCORA and to low at a compare match of TCORB.

With these settings, the 8-bit timer provides pulses output at a desired delay time from a TMRIn input determined by TCORA and with a pulse width determined by TCORB and TCORA.

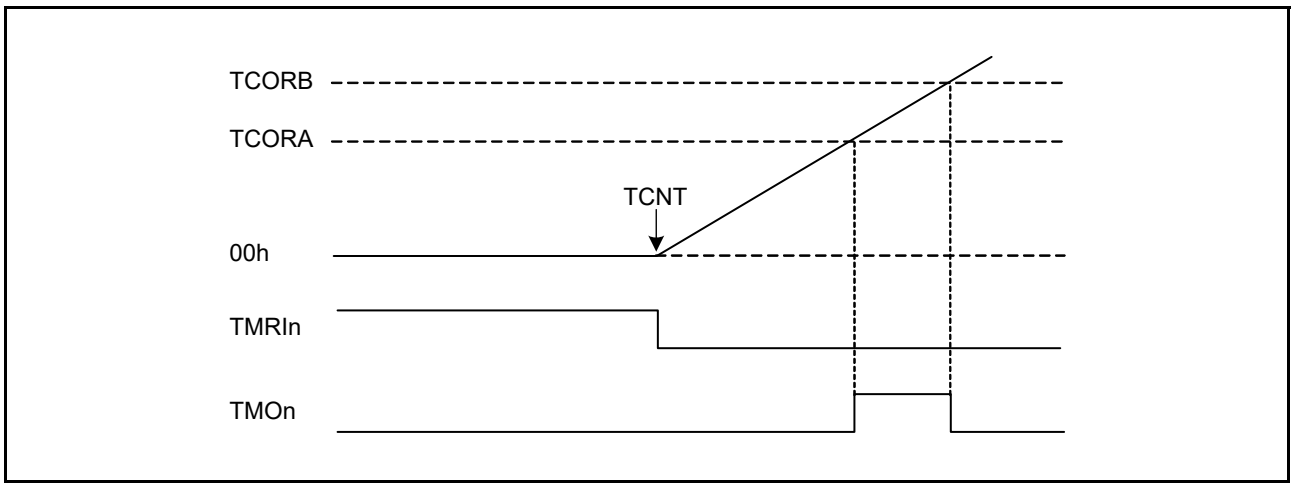


Figure 26.4 Example of External Counter Reset Signal Input (n = 0 to 2)

Downloaded from Arrow.com.

26.4 Operation Timing

26.4.1 TCNT Count Timing

Figure 26.5 shows the count timing of TCNT for internal clock. Figure 26.6 shows the count timing of TCNT for external clock.

Note that the external clock pulse width must be at least 1.5 PCLK cycles for increment at a single edge, and at least 2.5 PCLK cycles for increment at both edges. The counter will not increment correctly if the pulse width is less than these values.

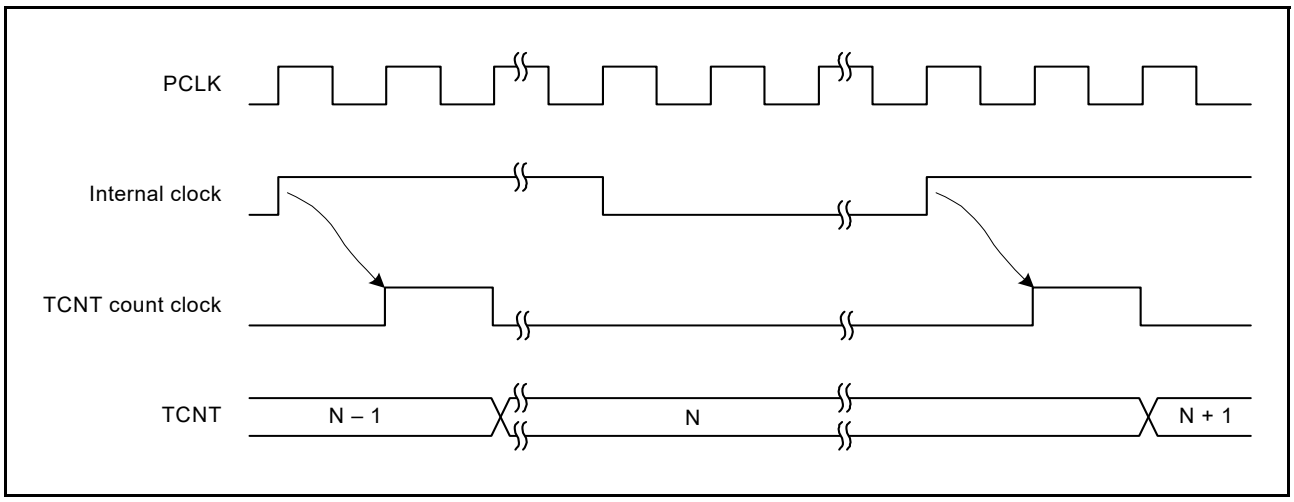


Figure 26.5 Count Timing for Internal Clock

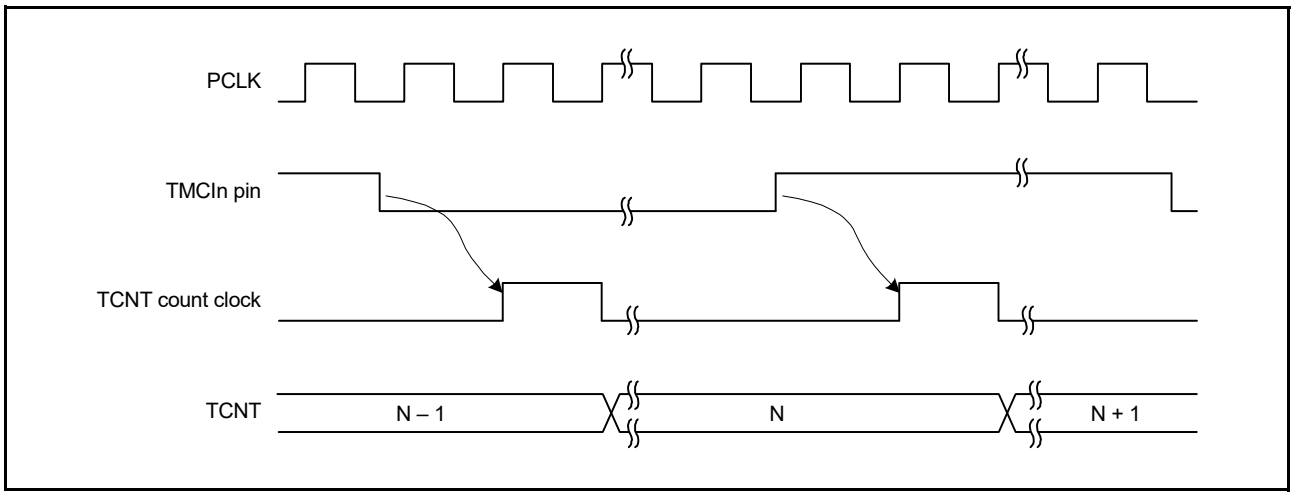


Figure 26.6 Count Timing for External Clock (at Both Edges)

Downloaded from Arrow.com.

26.4.2 Timing of Interrupt Signal Output on a Compare Match

A compare match refers to a match between the value of the TCORA or TCORB register and the TCNT, and a compare match interrupt signal is output at this time if the interrupt request is enabled. The compare match is generated in the last cycle in which the values match (at the time at which the value counted by TCNT to produce the match is updated). Accordingly, after a match between TCNT and the TCORA or TCORB register is detected, the compare match is not actually generated until the next cycle of the TCNT count clock. Figure 26.7 shows the timing of output of the interrupt signal.
For the corresponding interrupt vector number, refer to section 15, Interrupt Controller (ICUb) and Table 26.6.

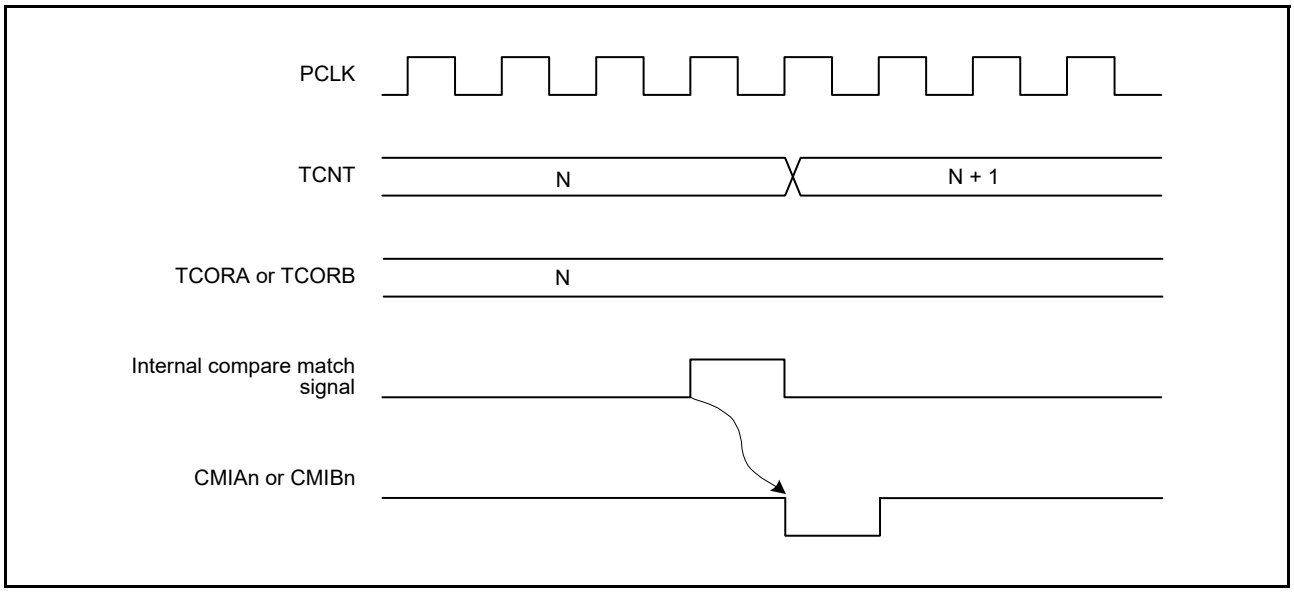


Figure 26.7 Timing of Interrupt Flag Setting to 1 at Compare Match (n = 0 to 3)

26.4.3 Timing of Timer Output Signal at Compare Match

When a compare match signal is generated, the output value specified by the TCSR.OSA[1:0] and OSB[1:0] bits is output on the timer output pin (TMOn).
Figure 26.8 shows the timing when the timer output is toggled by the compare match A signal.

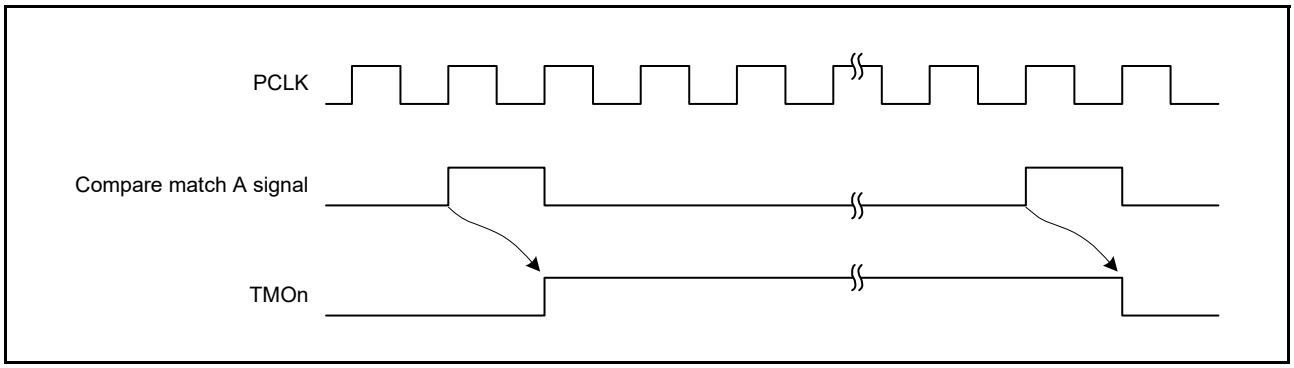


Figure 26.8 Timing of Timer Output Signal at Compare Match A Signal (n = 0 to 2)

Downloaded from Arrow.com.

26.4.4 Timing of Counter Clear by Compare Match

TCNT is cleared when compare match A or B occurs, depending on the settings of the TCR.CCLR[1:0] bits.
Figure 26.9 shows the timing of this operation.

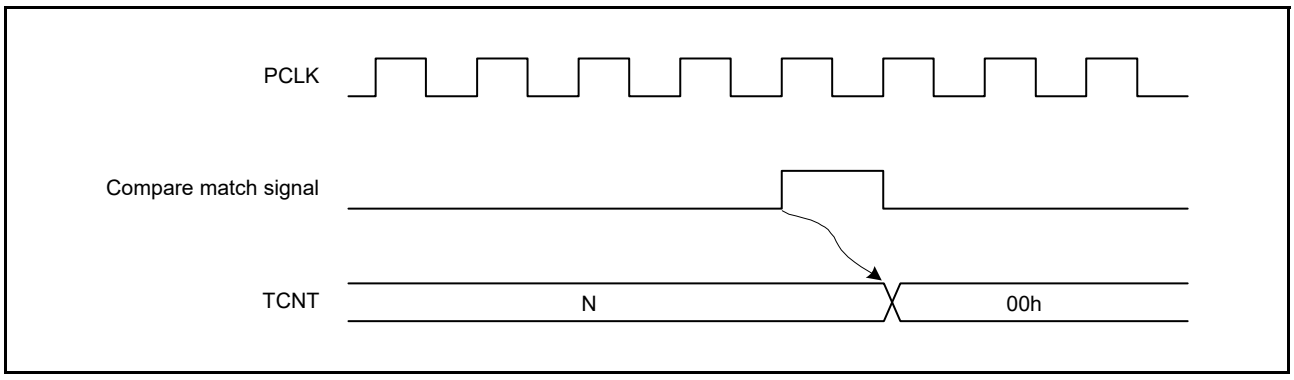


Figure 26.9 Timing of Counter Clear by Compare Match

26.4.5 Timing of the External Reset for TCNT

TCNT is cleared at the rising edge or high level of an external counter reset signal, depending on the settings of the TCR.CCLR[1:0] bits. At least 2 PCLK cycles are required from a reset input to clearing of TCNT.
Figure 26.10 and Figure 26.11 show the timing of this operation.

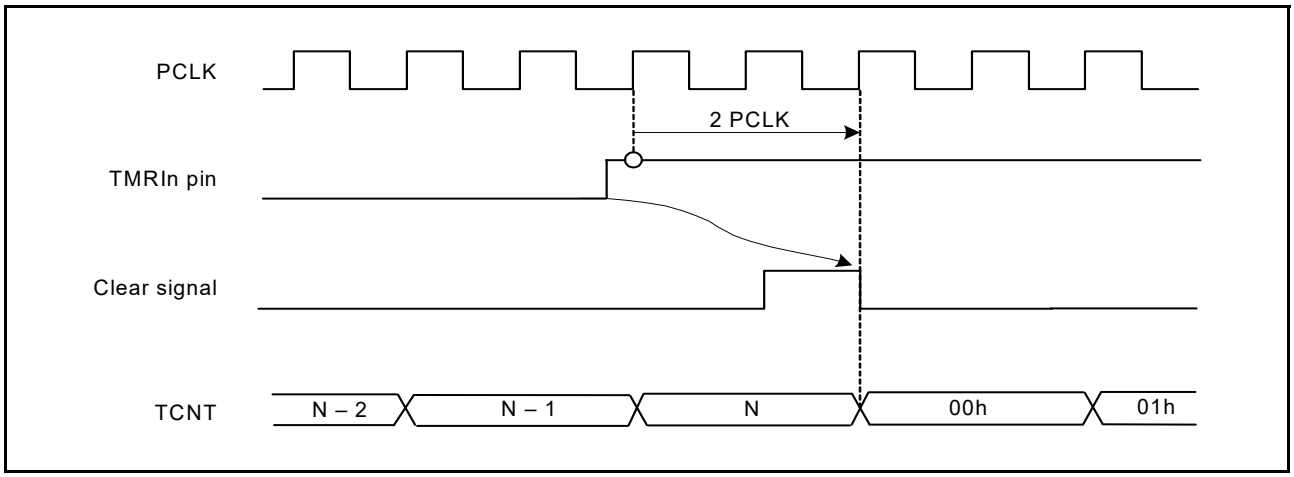


Figure 26.10 Clear Timing by External Counter Reset Signal (Rising Edge)

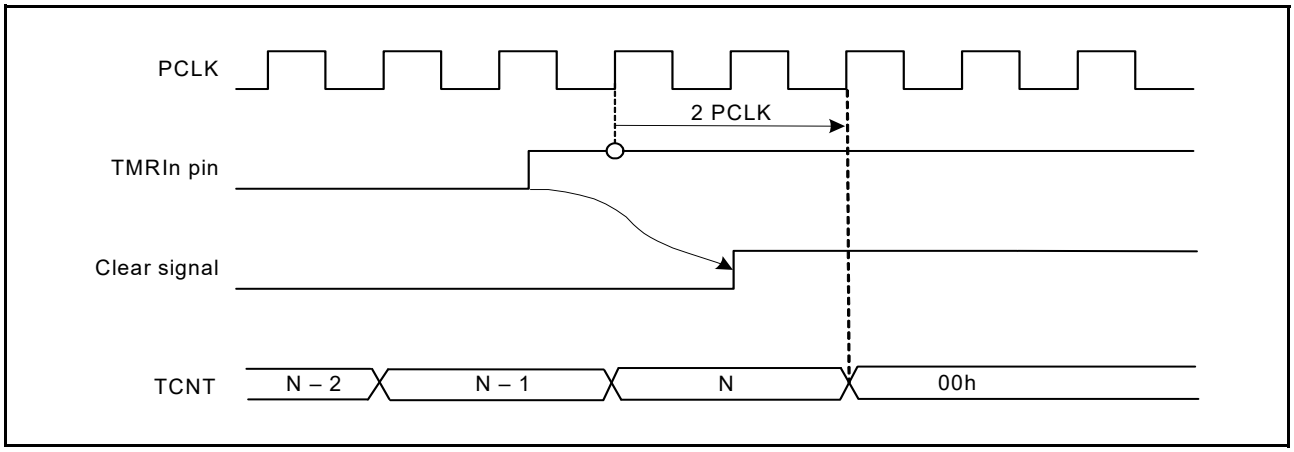


Figure 26.11 Clear Timing by External Counter Reset Signal (High Level)

Downloaded from Arrow.com.

26.4.6 Timing of Interrupt Signal Output on an Overflow

When TCNT overflows (changes from FFh to 00h), an overflow interrupt signal is output if this interrupt request is enabled.

Figure 26.12 shows the timing of output of the interrupt signal.

For the corresponding interrupt vector number, refer to section 15, Interrupt Controller (ICUb) and Table 26.6.

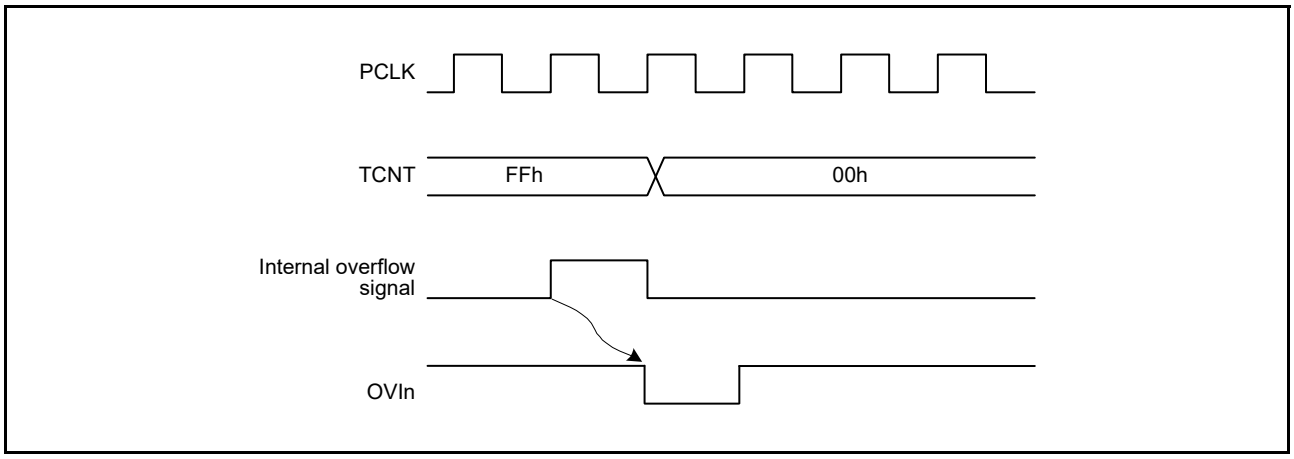


Figure 26.12 Timing of Overflow Interrupt Flag Setting to 1 (n = 0 to 3)

Downloaded from Arrow.com.

26.5 Operation with Cascaded Connection

If the CSS[1:0] bits in either TMR0.TCCR or TMR1.TCCR are set to 11b, the TMR of the two channels are cascaded. With this configuration, a single 16-bit timer could be used (16-bit counter mode) or compare matches of TMR0 could be counted by TMR1 (compare match count mode).
This section describes unit 0. When the two units are cascade-connected, the operation of unit 1 differs in some respects from that of unit 0.

26.5.1 16-Bit Count Mode

When the TMR0.TCCR.CSS[1:0] bits are set to 11b, the timer functions as a single 16-bit timer with TMR0 occupying the upper 8 bits and TMR1 occupying the lower 8 bits.

(1) Counter Clear Specification

- The settings of the TMR0.TCR.CCLR[1:0] bits become effective for the 16-bit counter. If the TMR0.TCR.CCLR[1:0] bits have been set for counter clear at compare match, the 16-bit counter (TMR0.TCNT and TMR1.TCNT together) is cleared when a 16-bit compare match event occurs. The counter of unit 1 can also be cleared by the signal on the TMRI2 pin (an external signal to reset the counter).
- The settings of the TMR1.TCR.CCLR[1:0] bits are ignored.

(2) Pin Output

- Control of output from the TMO0 pin by the TMR0.TCSR.OSA[1:0] and OSB[1:0] bits is in accordance with the 16-bit compare match conditions.
- Control of output from the TMO1 pin by the TMR1.TCSR.OSA[1:0] and OSB[1:0] bits is in accordance with the lower 8-bit compare match conditions. Output from TMR3 in unit 1 is not available since the chip does not have a pin equivalent to the TMO1 pin for TMR1 of unit 0.

26.5.2 Compare Match Count Mode

When the TMR1.TCCR.CSS[1:0] bits are set to 11b, TMR1.TCNT counts the number of occurrences of compare match A for TMR0. TMR0 and TMR1 are controlled independently. Conditions such as generation of interrupts, output from the TMOn pin (n = 0, 1), and counter clear are in accordance with the settings for each channel. Output from TMR3 in unit 1 is not available since the chip does not have a pin equivalent to the TMO1 pin for TMR1 of unit 0.

Downloaded from Arrow.com.

26.6 Interrupt Sources

26.6.1 Interrupt Sources and DTC Activation

There are three interrupt sources for TMRn: CMIAn, CMIBn, and OVIn. Their interrupt sources and priorities are listed in Table 26.6.
It is also possible to activate the DTC by means of CMIAn and CMIBn interrupts.

Table 26.6 TMR Interrupt Sources

Name	Interrupt Sources	DTC Activation	Priority
CMIA0	TMR0.TCORA compare match	Possible	High
CMIB0	TMR0.TCORB compare match	Possible	
OVI0	TMR0.TCNT overflow	Not possible	
CMIA1	TMR1.TCORA compare match	Possible	
CMIB1	TMR1.TCORB compare match	Possible	
OVI1	TMR1.TCNT overflow	Not possible	
CMIA2	TMR2.TCORA compare match	Possible	
CMIB2	TMR2.TCORB compare match	Possible	
OVI2	TMR2.TCNT overflow	Not possible	
CMIA3	TMR3.TCORA compare match	Possible	
CMIB3	TMR3.TCORB compare match	Possible	
OVI3	TMR3.TCNT overflow	Not possible	Low

Downloaded from Arrow.com.

26.7 Link Operation by ELC

26.7.1 Event Signal Output to ELC

The TMR uses the event link controller (ELC) to perform link operation to the previously specified module using the interrupt request signal as the event signal. The TMR outputs compare match A, compare match B, and overflow signals as event signals. Channels that can be used in this way are TMR0 and TMR2.
The event signal can be output regardless of the setting of the corresponding interrupt request enable bits (TMR0.TCR.OVIE or TMR2.TCR.OVIE, TMR0.TCR.CMIEA or TMR2.TCR.CMIEA, and TMR0.TCR.CMIEB or TMR2.TCR.CMIEB). For details, refer to section 20, Event Link Controller (ELC).
The event output function can be used for the cascaded operation.

26.7.2 TMR Operation when Receiving an Event Signal from ELC

The TMR can perform either of the following operations upon the event previously specified by the ELSRn register of the ELC. However, the ELC does not support the cascaded operation.

(1) Count Start

When the TMR count start operation is selected by the ELOPD register of the ELC and the event specified by ELSRn occurs, the TCSTR.TCS bit is set to 1, starting the TMR count operation. After the TMR count start operation is selected by the ELOPD register of the ELC, use the TCCR.CKS[2:0] and CSS[1:0] bits to select the count source.
If the specified event occurs while the TCS bit is 1, the event is ignored.
Write 0 to the TCSTR.TCS bit to stop counting.
When the count start event is input in the count stopped state, the TMR starts counting again according to the CKS[2:0] and CSS[1:0] bits.
The TCS bit is valid only when the ELOPD.TMR0MD[1:0] and ELOPD.TMR2MD[1:0] bits of the ELC select the count start operation.

(2) Event Count

When the TMR event count operation is selected by the ELOPD register of the ELC and the event specified by ELSRn occurs, the events are counted as the count source regardless of the TCCR.CKS[2:0] and CSS[1:0] bit settings. Reading the counter value returns the number of events that have been actually input.

(3) Count Restart

When the TMR count restart operation is selected by the ELOPD register of the ELC and the event specified by ELSRn occurs, the TCNT counter value is modified to the initial value. If the CKS[2:0] and CSS[1:0] bit settings are not disabling the clock input, the count operation is continued.

Downloaded from Arrow.com.

26.7.3 Notes on Operating TMR According to an Event Signal from ELC

The following describes the notes on operating the TMR using the event link feature.

(1) Count Start

When the event specified by ELSRn occurs during the write cycle to the TCSTR.TCS bit, the cycle is not completed; setting 1 according to the event occurrence takes priority.

(2) Event Count

When the event specified by ELSRn occurs during the write cycle to the TCNT, the cycle is not completed; event count operation according to the event occurrence takes priority.

(3) Count Restart

When the event specified by ELSRn occurs during the write cycle to the TCNT, the cycle is not completed; count value initialization according to the event occurrence takes priority.

Downloaded from Arrow.com.

26.8 Usage Notes

26.8.1 Module Stop State Setting

Operation of the TMR can be disabled or enabled by using the module stop control registers. The initial setting is for halting of TMR operation. Register access becomes possible after release from the module stop state. For details, refer to section 11, Low Power Consumption.

26.8.2 Notes on Setting Cycle

If the compare match is selected for counter clear, TCNT is cleared at the last PCLK in the cycle in which the value of TCNT matches with that of TCORA or TCORB. TCNT updates the counter value at this last state. Therefore, the counter frequency is obtained by the following formula (f: Counter frequency, PCLK: Operating frequency, N: TCORA and TCORB register setting value).

$$f = PCLK / (N + 1)$$

26.8.3 Conflict between TCNT Write and Counter Clear

If a counter clear signal is generated concurrently with CPU write to TCNT, the clear takes priority and the write is not performed as shown in Figure 26.13.

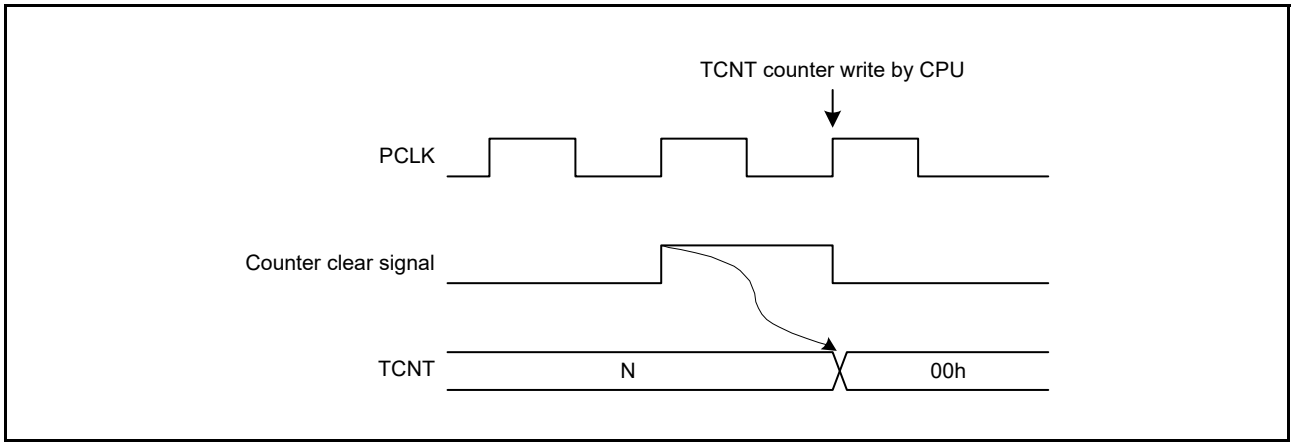


Figure 26.13 Conflict between TCNT Write and Counter Clear

Downloaded from Arrow.com.

26.8.4 Conflict between TCNT Write and Increment

Even if a counting-up signal is generated concurrently with CPU write to TCNT, the counting-up is not performed and the write takes priority as shown in Figure 26.14.

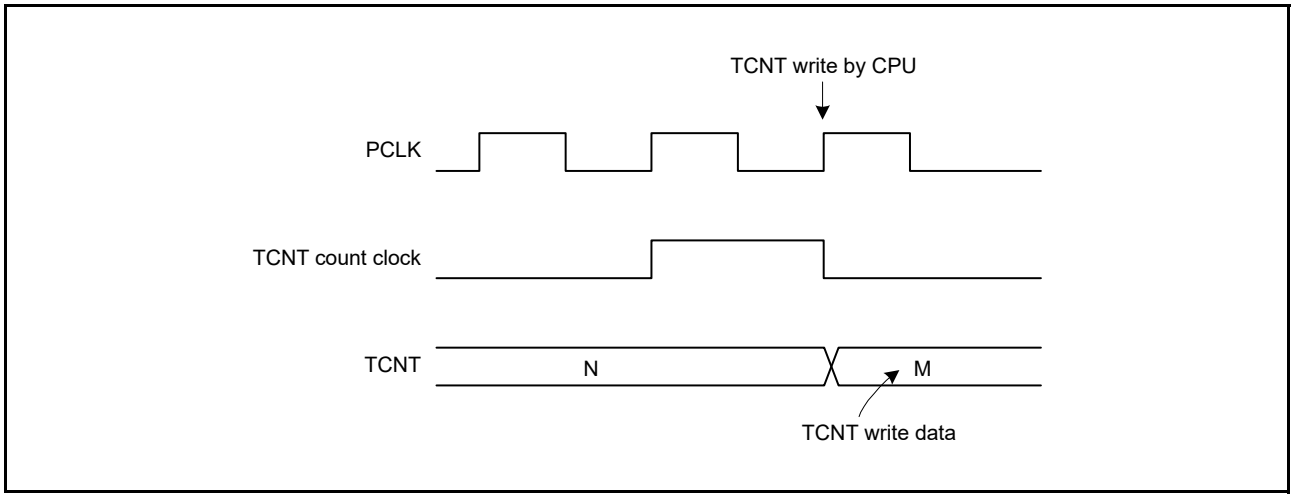


Figure 26.14 Conflict between TCNT Write and Increment

26.8.5 Conflict between TCORA or TCORB Write and Compare Match

Even if a compare match signal is generated simultaneously with CPU write to TCORA or TCORB as shown in Figure 26.15, the write takes priority and the compare match signal does not reach High level.

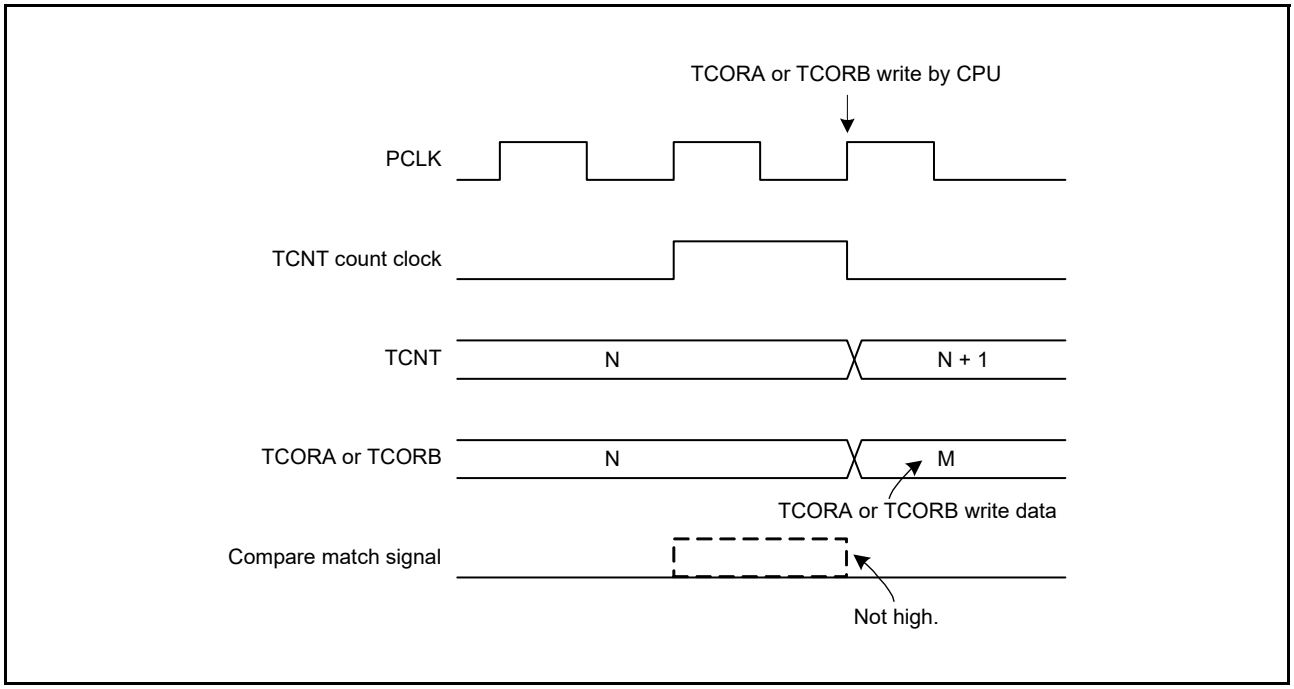


Figure 26.15 Conflict between TCORA or TCORB Write and Compare Match

Downloaded from Arrow.com.

26.8.6 Conflict between Compare Matches A and B

If compare match events A and B occur at the same time, the 8-bit timer operates in accordance with the priorities for the output methods high for compare match A and compare match B, as listed in Table 26.7.

Table 26.7 Timer Output Priorities

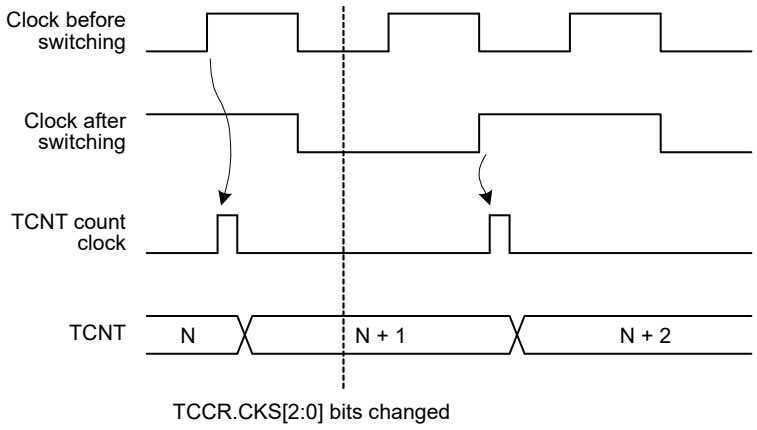
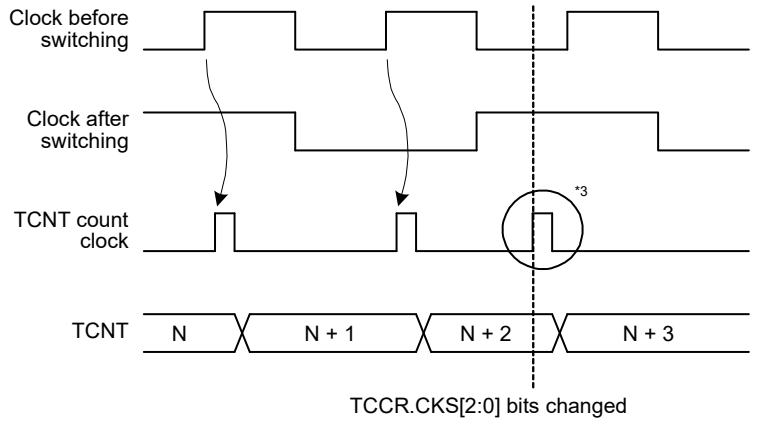
Output Setting	Priority
Toggle output	High
High output	↑
Low output	
No change	Low

26.8.7 Switching of Internal Clocks and TCNT Operation

TCNT may be incremented erroneously depending on when the internal clock is switched. Table 26.8 lists the relationship between the timing at which the internal clock is switched (by writing to the TCCR.CKS[2:0] bits) and the operation of TCNT.

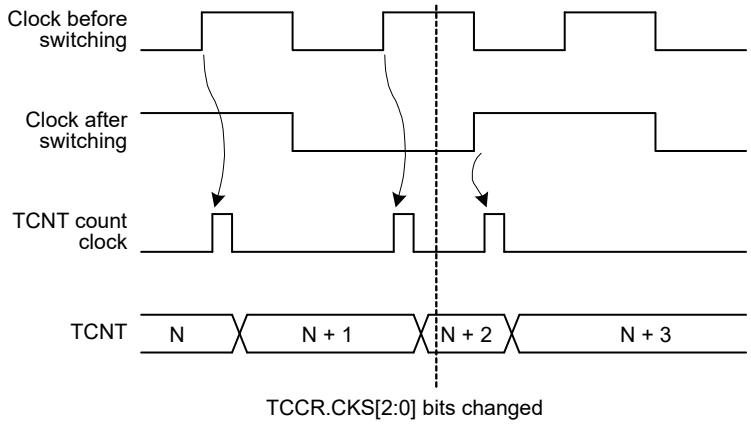
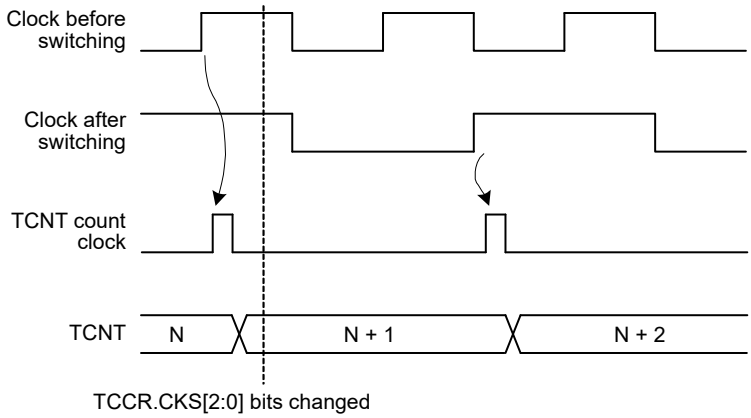
When TCNT count clock is generated from an internal clock, the rising edge of the internal clock pulse are always monitored. If the signal levels of the clocks before and after switching change from low to high as shown in No. 2 in Table 26.8, the change is considered as an edge. Therefore, a TCNT count clock is generated and TCNT is incremented. The erroneous increment of TCNT can also happen when switching between internal and internal clocks.

Table 26.8 Switching of Internal Clocks and TCNT Operation (1/2)

No.	Timing to Change the TCCR.CKS[2:0] Bits	TCNT Counter Operation
1	Switching from low to low*1	
2	Switching from low to high*2	

Downloaded from Arrow.com.

Table 26.8 Switching of Internal Clocks and TCNT Operation (2/2)

No.	Timing to Change the TCCR.CKS[2:0] Bits	TCNT Counter Operation
3	Switching from high to low*4	
4	Switching from high to high	

Note 1. Includes switching from low to stop, and from stop to low.
Note 2. Includes switching from stop to high.
Note 3. Generated because the change of the signal levels is considered as an edge; TCNT counter is incremented.
Note 4. Includes switching from high to stop.

Downloaded from Arrow.com.

26.8.8 Clock Source Setting with Cascaded Connection

If 16-bit counter mode and compare match count mode are specified at the same time, count clocks for TMR0.TCNT and TMR1.TCNT (TMR2.TCNT and TMR3.TCNT) are not generated, and the counter stops. Do not specify 16-bit counter mode and compare match count mode simultaneously.

26.8.9 Continuous Output of Compare Match Interrupt Signal

When TCORA or TCORB is set to 00h, PCLK/1 is set as the internal clock, and compare match is set as the counter clear source, the TCNT counter remains 00h and is not updated, and a compare match interrupt signal is output continuously to form a flat signal level.
At this time, the interrupt controller cannot detect the second and subsequent interrupts.
Figure 26.16 shows operation timing when the compare match interrupt signal is continuously output.

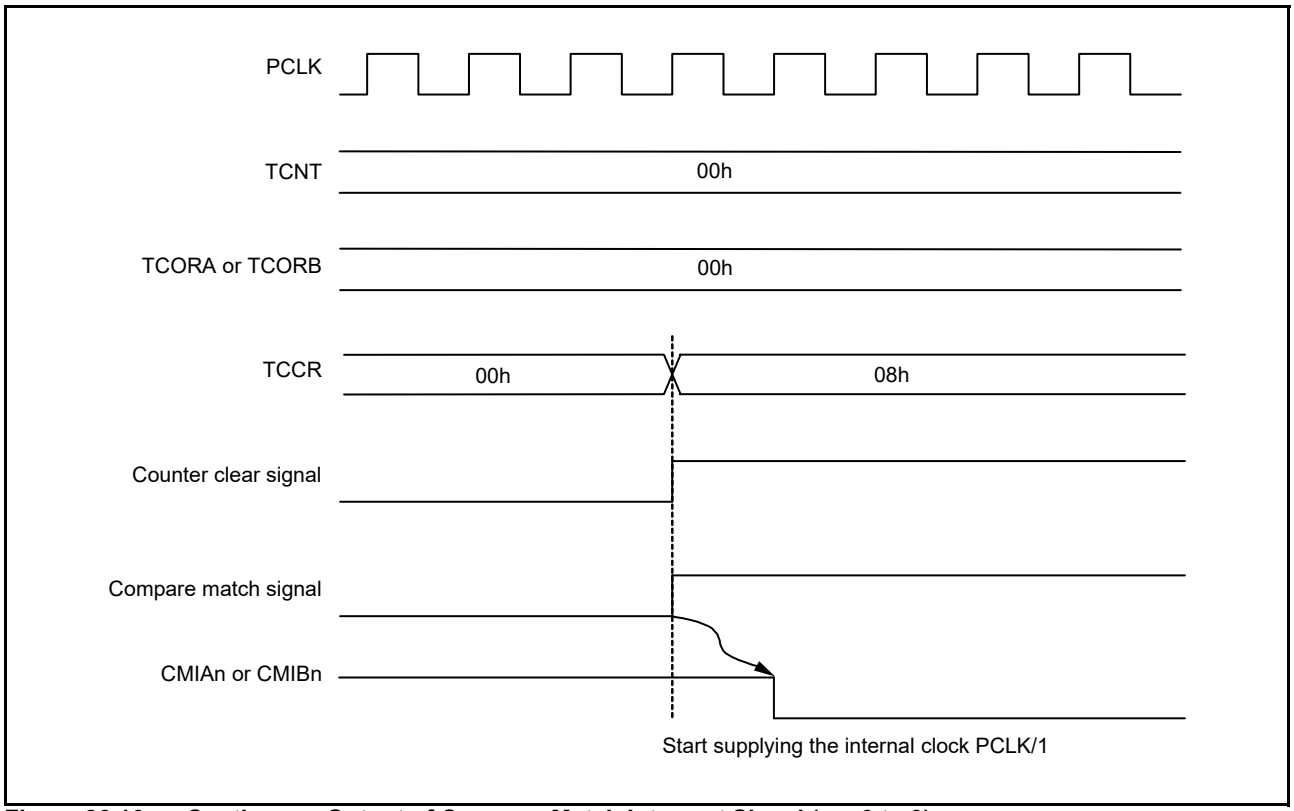


Figure 26.16 Continuous Output of Compare Match Interrupt Signal (n = 0 to 3)

Downloaded from Arrow.com.

27. Compare Match Timer (CMT)

This MCU has two on-chip compare match timer (CMT) units (unit 0 and unit 1), each consisting of a two-channel 16-bit timer (i.e., a total of four channels). The CMT has a 16-bit counter, and can generate interrupts at set intervals.
In this section, “PCLK” is used to refer to PCLKB.

27.1 Overview

Table 27.1 lists the specifications for the CMT.
Figure 27.1 shows a block diagram of the CMT (unit 0). A two-channel CMT constitutes a unit. Unit 0 and unit 1 are the same in terms of specifications. Compare match timer start register 0 (CMSTR0) and compare match interrupts (CMI0 and CMI1) of unit 0 correspond to compare match timer start register 1 (CMSTR1) and compare match interrupts (CMI2 and CMI3) of unit 1.

Table 27.1 CMT Specifications

Item	Description
Count clocks	• Four frequency dividing clocks One clock from PCLK/8, PCLK/32, PCLK/128, and PCLK/512 can be selected for each channel.
Interrupt	A compare match interrupt can be requested for each channel.
Event link function (output)	An event signal is output upon a CMT1 compare match.
Event link function (input)	Linking to the specified module is possible. CMT1 count start, event counter, or count restart operation is possible.
Low power consumption function	Each unit can be placed in a module stop state.

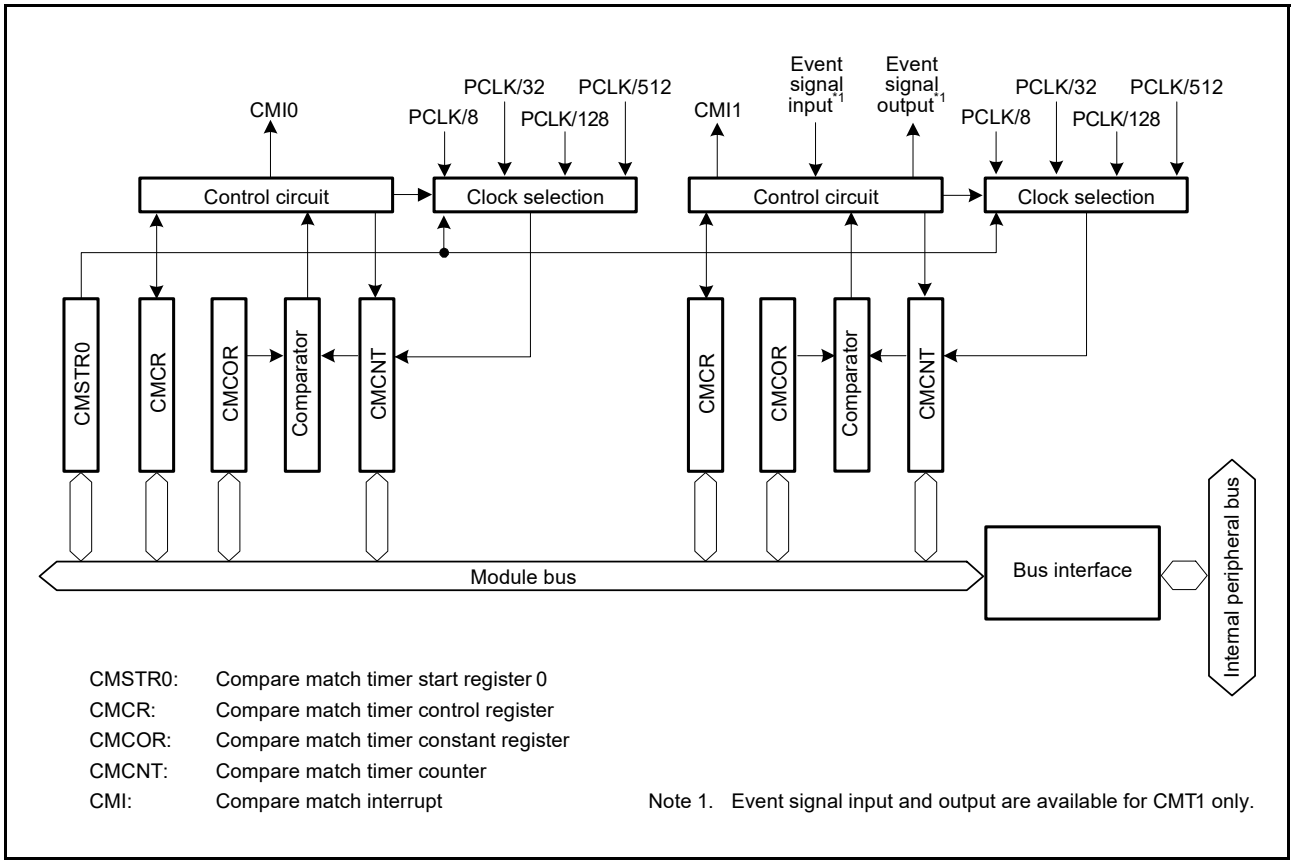


Figure 27.1 CMT (Unit 0) Block Diagram

Downloaded from Arrow.com.

27.2 Register Descriptions

27.2.1 Compare Match Timer Start Register 0 (CMSTR0)

Address(es): 0008 8000h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	STR1	STR0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	STR0	Count Start 0	0: CMT0.CMCNT count is stopped. 1: CMT0.CMCNT count is started.	R/W
b1	STR1	Count Start 1	0: CMT1.CMCNT count is stopped. 1: CMT1.CMCNT count is started.	R/W
b15 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

27.2.2 Compare Match Timer Start Register 1 (CMSTR1)

Address(es): 0008 8010h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	STR3	STR2
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	STR2	Count Start 2	0: CMT2.CMCNT count is stopped. 1: CMT2.CMCNT count is started.	R/W
b1	STR3	Count Start 3	0: CMT3.CMCNT count is stopped. 1: CMT3.CMCNT count is started.	R/W
b15 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Downloaded from Arrow.com.

27.2.3 Compare Match Timer Control Register (CMCR)

Address(es): CMT0.CMCR 0008 8002h, CMT1.CMCR 0008 8008h, CMT2.CMCR 0008 8012h, CMT3.CMCR 0008 8018h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	CMIE	—	—	—	—	CKS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	x	0	0	0	0	0	0	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b1, b0	CKS[1:0]	Clock Select	b1 b0 0 0: PCLK/8 0 1: PCLK/32 1 0: PCLK/128 1 1: PCLK/512	R/W
b5 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	CMIE	Compare Match Interrupt Enable	0: Compare match interrupt (CMIn) disabled 1: Compare match interrupt (CMIn) enabled	R/W
b7	—	Reserved	This bit is read as undefined. The write value should be 1.	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

CKS[1:0] Bits (Clock Select)

These bits select the count source from four frequency dividing clocks obtained by dividing the peripheral module clock (PCLK).

When the CMSTRm.STRn (m = 0, 1; n = 0 to 3) bit is set to 1, the CMCNT counter starts counting up on the clock selected with the CKS[1:0] bits.

CMIE Bit (Compare Match Interrupt Enable)

The CMIE bit enables or disables compare match interrupt (CMIn) (n = 0 to 3) generation when the CMCNT counter and the CMCOR register values match.

Downloaded from Arrow.com.

27.2.4 Compare Match Counter (CMCNT)

Address(es): CMT0.CMCNT 0008 8004h, CMT1.CMCNT 0008 800Ah, CMT2.CMCNT 0008 8014h, CMT3.CMCNT 0008 801Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The CMCNT counter is a readable/writable up-counter.
When an frequency dividing clock is selected by the CMCR.CKS[1:0] bits and the CMSTRm.STRn (m = 0, 1; n = 0 to 3) bit is set to 1, the CMCNT counter starts counting up using the selected clock.
When the value in the CMCNT counter and the value in the CMCOR register match, the CMCNT counter is set to 0000h. At the same time, a compare match interrupt (CMIn) (n = 0 to 3) is generated.

27.2.5 Compare Match Constant Register (CMCOR)

Address(es): CMT0.CMCOR 0008 8006h, CMT1.CMCOR 0008 800Ch, CMT2.CMCOR 0008 8016h, CMT3.CMCOR 0008 801Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

The CMCOR register is a readable/writable register to set a value for compare match with the CMCNT counter.

Downloaded from Arrow.com.

27.3 Operation

27.3.1 Periodic Count Operation

When an frequency dividing clock is selected by the CMCR.CKS[1:0] bits and the CMSTRm.STRn (m = 0, 1; n = 0 to 3) bit is set to 1, the CMCNT counter starts counting up using the selected clock.
When the value in the counter and the value in the register match, a compare match interrupt (CMIn) (n = 0 to 3) is generated. The CMCNT counter then starts counting up again from 0000h. Figure 27.2 shows the operation of the CMCNT counter.

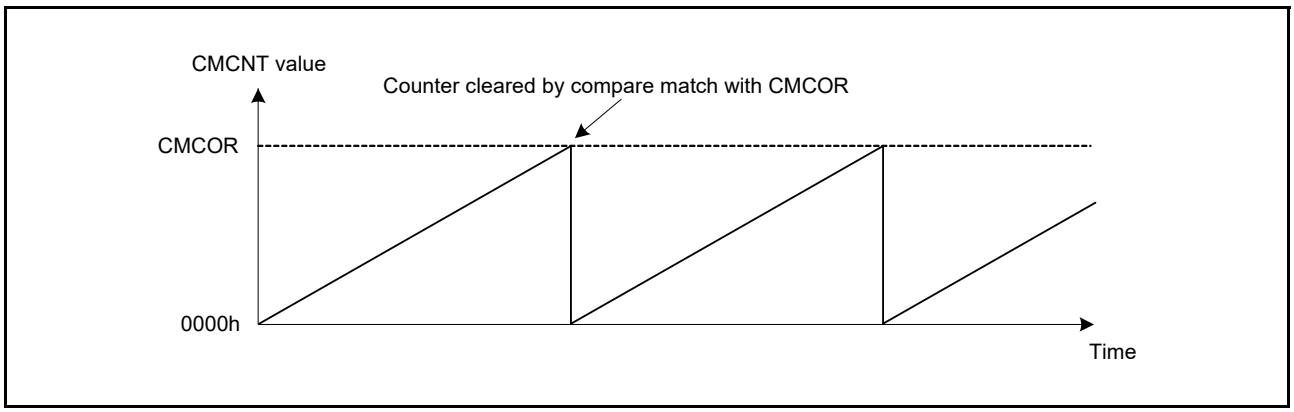


Figure 27.2 CMCNT Counter Operation

27.3.2 CMCNT Count Timing

As the count clock to be input to the CMCNT counter, one of four frequency dividing clocks (PCLK/8, PCLK/32, PCLK/128, and PCLK/512) obtained by dividing the peripheral module clock (PCLK) can be selected with the CMCR.CKS[1:0] bits. Figure 27.3 shows the timing of the CMCNT counter.

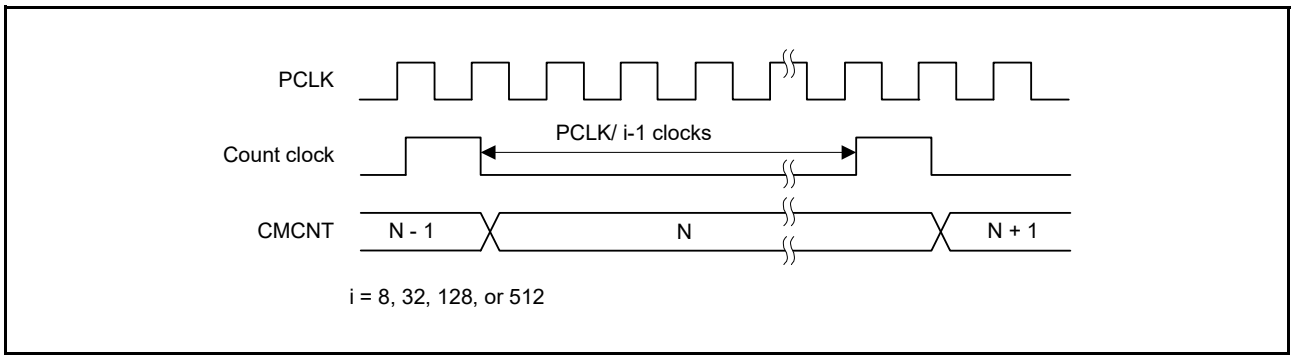


Figure 27.3 CMCNT Count Timing

Downloaded from Arrow.com.

27.4 Interrupts

27.4.1 Interrupt Sources

The CMT has channels and each of them to which a different vector address is allocated has a compare match interrupt (CMIn) (n = 0 to 3). When a compare match interrupt occurs, the corresponding interrupt request is output.
When the interrupt request is used to generate a CPU interrupt, the priority of channels can be changed by the interrupt controller settings. For details, see section 15, Interrupt Controller (ICUb).

Table 27.2 CMT Interrupt Sources

Name	Interrupt Sources	DTC Activation	DMAC Activation
CMI0	Compare match in CMT0	Possible	Possible
CMI1	Compare match in CMT1	Possible	Possible
CMI2	Compare match in CMT2	Possible	Possible
CMI3	Compare match in CMT3	Possible	Possible

27.4.2 Timing of Compare Match Interrupt Generation

When the CMCNT counter and the CMCOR register match, a compare match interrupt (CMIn) (n = 0 to 3) is generated. A compare match signal is generated at the last state in which the values match (the timing when the CMCNT counter updates the matched count value). That is, after a match between the CMCOR register and the CMCNT counter, the compare match signal is not generated until the next the CMCNT counter input clock.
Figure 27.4 shows the timing of a compare match interrupt.

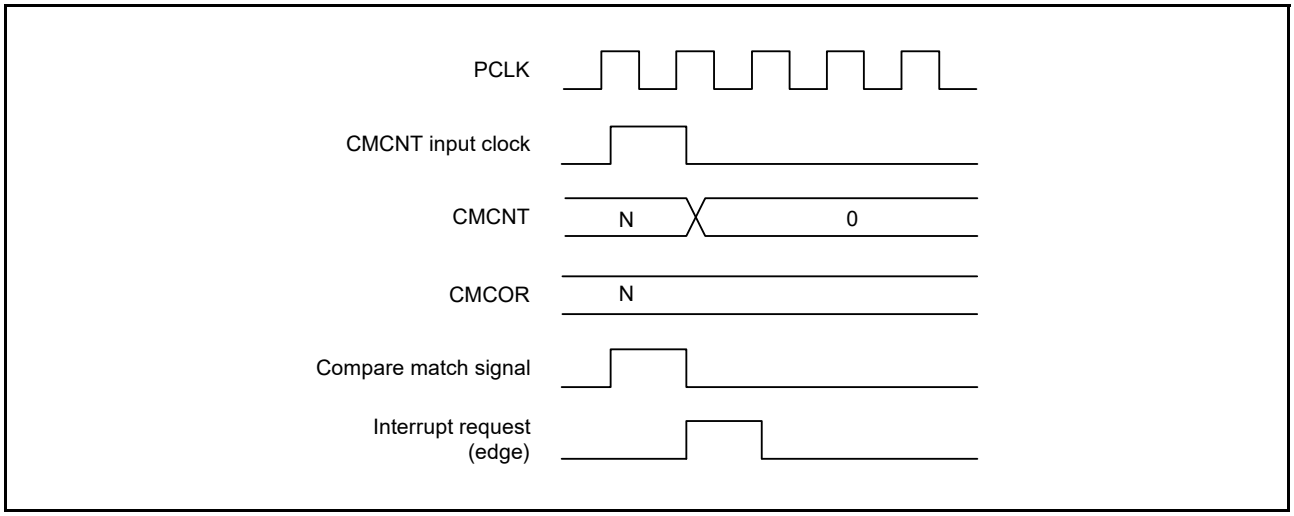


Figure 27.4 Timing of a Compare Match Interrupt

Downloaded from Arrow.com.

27.5 Link Operations by ELC

27.5.1 Event Signal Output to ELC

The CMT uses the event link controller (ELC) to perform link operation to a preset module using the interrupt request signal as the event signal. The CMT outputs the event signal upon a CMT1 compare match.
The event signal can be output regardless of the setting of the corresponding interrupt request enable bit (CMTn.CMCR.CMIE).

27.5.2 CMT Operation When Receiving an Event Signal from ELC

The CMT can perform either of the following operations upon the event preset by the ELSR7 register of the ELC.

(1) Count Start

When the CMT count start operation is selected by the ELOPC register of the ELC and the event specified by ELSR7 occurs, the CMSTR0.STR1 bit is set to 1, starting the CMT count operation.
However, if the specified event occurs while the CMSTR0.STR1 bit is 1, the event is ignored.

(2) Event Count

When the CMT event count operation is selected by the ELOPC register of the ELC and the event specified by ELSR7 occurs with the CMSTR0.STR1 bit being 1, the events are counted as the count source regardless of the CMT1.CMCR.CKS[1:0] bit setting. Reading the counter value returns the number of events that have been actually input.

(3) Count Restart

When the CMT count restart operation is selected by the ELOPC register of the ELC and the event specified by ELSR7 occurs, the CMT1.CMCNT counter value is modified to the initial value. If the CMSTR0.STR1 bit is 1 here, the count operation can be continued.

27.5.3 Notes on Operating CMT According to an Event Signal from ELC

The following describes the notes on operating the CMT using the event link feature.

(1) Count Start

When the event specified by ELSR7 occurs during the write cycle to the CMSTR0.STR1 bit, the cycle is not completed; setting 1 according to the event occurrence takes priority.

(2) Event Count

When the event specified by ELSR7 occurs during the write cycle to the CMT1.CMCNT, the cycle is not completed; event count operation according to the event occurrence takes priority.

(3) Count Restart

When the event specified by ELSR7 occurs during the write cycle to the CMT1.CMCNT, the cycle is not completed; count value initialization according to the event occurrence takes priority.

Downloaded from Arrow.com.

27.6 Usage Notes

27.6.1 Setting the Module Stop Function

The CMT can be enabled or disabled using the module stop control register. After a reset, the CMT is in the module stop state. The registers can be accessed by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

27.6.2 Conflict between CMCNT Counter Writing and Compare Match

When the compare match signal is generated while writing to the CMCNT counter, clearing the CMCNT counter has priority over writing to it. In this case, the CMCNT counter is not written to. Figure 27.5 shows the timing to clear the CMCNT counter.

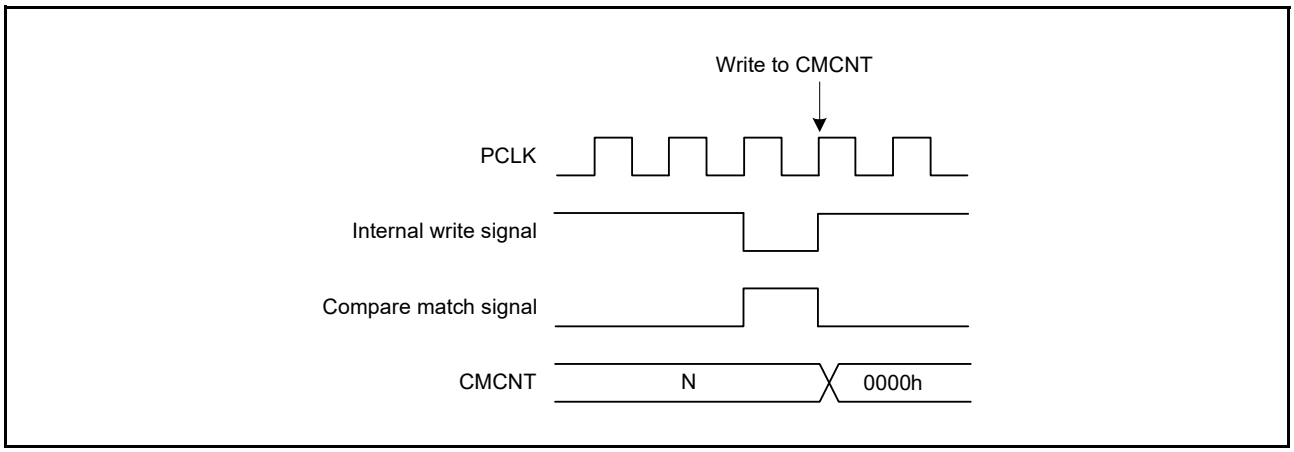


Figure 27.5 Conflict between CMCNT Counter Writing and Compare Match

27.6.3 Conflict between CMCNT Counter Writing and Incrementing

If writing to the counter and the incrementing conflict, the writing has priority over the incrementing. Figure 27.6 shows the timing to write the CMCNT counter.

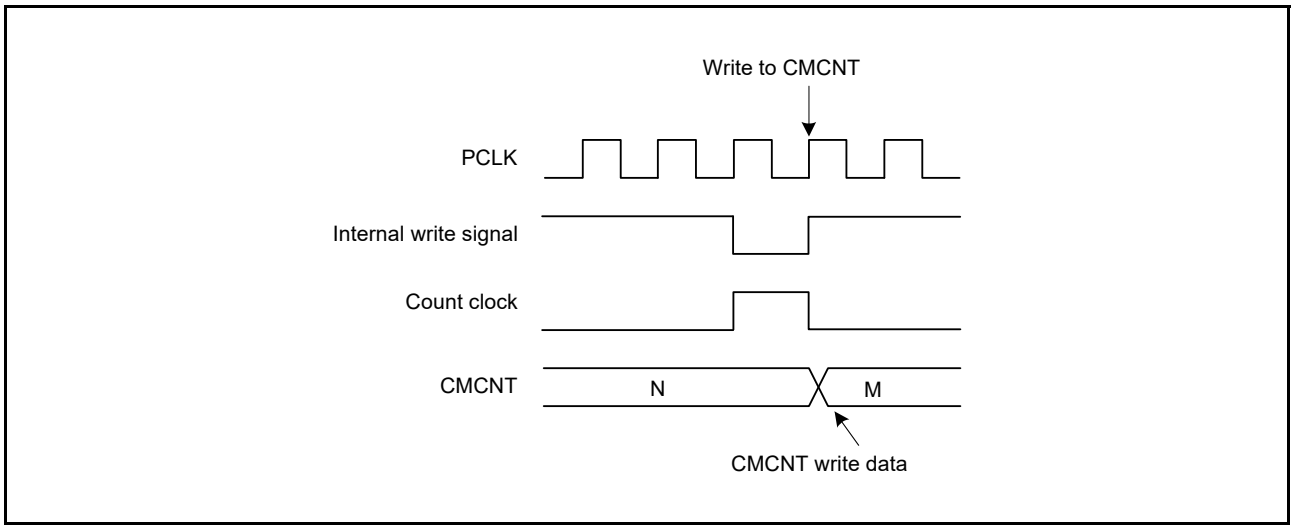


Figure 27.6 Conflict between CMCNT Counter Writing and Incrementing

Downloaded from Arrow.com.

28. Realtime Clock (RTCe)

In this section, “PCLK” is used to refer to PCLKB.

28.1 Overview

The RTC has two types of counting modes: calendar count mode and binary count mode. They are used by switching the register settings.
For calendar count mode, the RTC has a 100 year calendar from 2000 to 2099 and automatically adjusts dates for leap years.
For binary count mode, the RTC does not count in terms of years, months, dates, day-of-week, hours, or minutes; it counts seconds, and retains the information as a serial value. This mode can be used for calendars other than the Gregorian calendar.
The RTC uses the 128-Hz clock which is acquired by the count source divided by the prescaler as the reference clock.
Year, month, date, day-of-week, a.m./p.m. (in 12-hour mode), hour, minute, second, or 32-bit binary is counted in 1/128 second units.

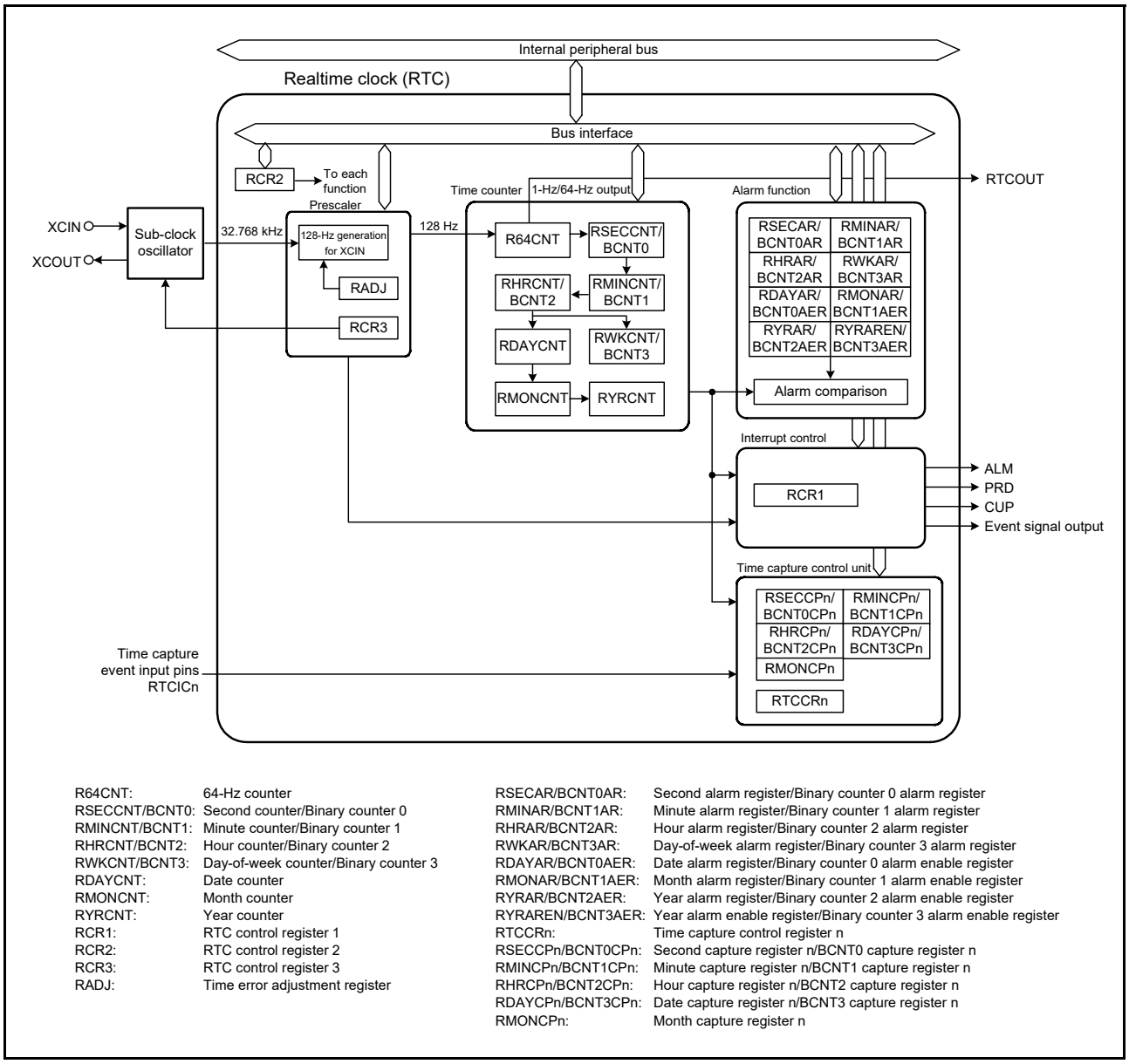
Table 28.1 lists the specifications of the RTC, Figure 28.1 shows a block diagram of the RTC, and Table 28.2 shows the pin configuration of the RTC.

Table 28.1 RTC Specifications

Item	Description
Count mode	Calendar count mode/binary count mode
Count source*1	Sub-clock (XCIN)
Clock and calendar functions	• Calendar count mode Year, month, date, day-of-week, hour, minute, second are counted, BCD display 12 hours/24 hours mode switching function 30 seconds adjustment function (a number less than 30 is rounded down to 00 seconds, and 30 seconds or more are rounded up to one minute) Automatic adjustment function for leap years • Binary count mode Count seconds in 32 bits, binary display • Common to both modes Start/stop function The sub-second digit is displayed in binary units (1 Hz, 2 Hz, 4 Hz, 8 Hz, 16 Hz, 32 Hz, or 64 Hz). Clock error correction function Clock (1 Hz/64 Hz) output
Interrupts	• Alarm interrupt (ALM) As an alarm interrupt condition, selectable which of the below is compared with: - Calendar count mode: Year, month, date, day-of-week, hour, minute, or second can be selected - Binary count mode: Each bit of the 32-bit binary counter • Periodic interrupt (PRD) 2 seconds, 1 second, 1/2 second, 1/4 second, 1/8 second, 1/16 second, 1/32 second, 1/64 second, 1/128 second, or 1/256 second can be selected as an interrupt period. • Carry interrupt (CUP) An interrupt is generated at either of the following timings: - When a carry from the 64-Hz counter to the second counter is generated. - When the 64-Hz counter is changed and the R64CNT register is read at the same time. • Recovery from software standby mode can be performed by an alarm interrupt or periodic interrupt
Time capture function	• Times can be captured when the edge of the time capture event input pin is detected. For every event input, month, date, hour, minute, and second are captured or 32-bit binary counter value is captured.
Event link function	Periodic event output

Note 1. Satisfy the frequency of the peripheral module clock (PCLKB) ≥ the frequency of the count source.

Downloaded from Arrow.com.



R64CNT: 64-Hz counter
RSECCNT/BCNT0: Second counter/Binary counter 0
RMINCNT/BCNT1: Minute counter/Binary counter 1
RHRCNT/BCNT2: Hour counter/Binary counter 2
RWKCNT/BCNT3: Day-of-week counter/Binary counter 3
RDAYCNT: Date counter
RMONCNT: Month counter
RYRCNT: Year counter
RCR1: RTC control register 1
RCR2: RTC control register 2
RCR3: RTC control register 3
RADJ: Time error adjustment register

RSECAR/BCNT0AR: Second alarm register/Binary counter 0 alarm register
RMINAR/BCNT1AR: Minute alarm register/Binary counter 1 alarm register
RHRAR/BCNT2AR: Hour alarm register/Binary counter 2 alarm register
RWKAR/BCNT3AR: Day-of-week alarm register/Binary counter 3 alarm register
RDAYAR/BCNT0AER: Date alarm register/Binary counter 0 alarm enable register
RMONAR/BCNT1AER: Month alarm register/Binary counter 1 alarm enable register
RYRAR/BCNT2AER: Year alarm register/Binary counter 2 alarm enable register
RYRAREN/BCNT3AER: Year alarm enable register/Binary counter 3 alarm enable register
RTCCRn: Time capture control register n
RSECCPn/BCNT0CPn: Second capture register n/BCNT0 capture register n
RMINCPn/BCNT1CPn: Minute capture register n/BCNT1 capture register n
RHRCPn/BCNT2CPn: Hour capture register n/BCNT2 capture register n
RDAYCPn/BCNT3CPn: Date capture register n/BCNT3 capture register n
RMONCPn: Month capture register n

Figure 28.1 Block Diagram of RTC (n = 0, 1)

Table 28.2 Pin Configuration of RTC

Pin Name	I/O	Function
XCIN	Input	Connect a 32.768-kHz crystal to these pins.
XCOUT	Output	
RTCOUT	Output	This pin is used to output a 1-Hz/64-Hz waveform.
RTCIC0	Input	Time capture event input pins
RTCIC1	Input	

Downloaded from Arrow.com.

28.2 Register Descriptions

When writing to or reading from RTC registers, do so in accordance with section 28.6.5, Notes on Writing to and Reading from Registers.

If the value in an RTC register after a reset is given as x (undefined bits) in the list, it is not initialized by a reset. When RTC enters the reset state or a low power consumption state during counting operations (i.e. while the RCR2.START bit is 1), the year, month, day of the week, date, hours, minutes, seconds, and 64-Hz counters continue to operate. Note that a reset generated during writing to or updating of a register might destroy the register value. In addition, do not allow the chip to enter software standby mode immediately after setting any of these registers. For details, refer to section 28.6.4, Transitions to Low Power Consumption Modes after Setting Registers.

28.2.1 64-Hz Counter (R64CNT)

Address(es): RTC.R64CNT 0008 C400h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	F1HZ	F2HZ	F4HZ	F8HZ	F16HZ	F32HZ	F64HZ
Value after reset:	0	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	F64HZ	64 Hz	Indicate the state between 1 Hz and 64 Hz of the sub-second digit.	R
b1	F32HZ	32 Hz		R
b2	F16HZ	16 Hz		R
b3	F8HZ	8 Hz		R
b4	F4HZ	4 Hz		R
b5	F2HZ	2 Hz		R
b6	F1HZ	1 Hz		R
b7	—	Reserved	This bit is read as 0. Writing to this bit has no effect.	R

The R64CNT counter is used in both calendar count mode and in binary count mode.

The 64-Hz counter (R64CNT) generates a period of one second by counting the 128-Hz reference clock.

The state in the sub-second range can be confirmed by reading this counter.

This counter is set to 00h by an RTC software reset or executing 30-second adjustment.

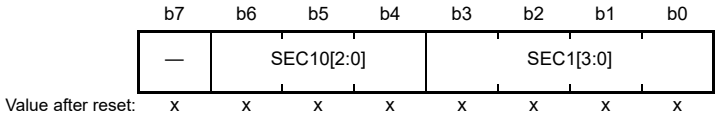
To read this counter, follow the procedure in section 28.3.5, Reading 64-Hz Counter and Time.

Downloaded from Arrow.com.

28.2.2 Second Counter (RSECCNT)/Binary Counter 0 (BCNT0)

(1) In calendar count mode:

Address(es): RTC.RSECCNT 0008 C402h



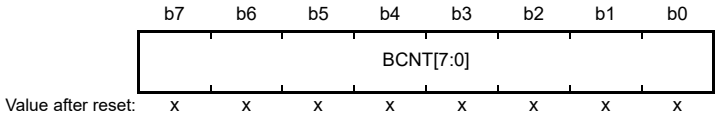
x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	SEC1[3:0]	1-Second Count	Counts from 0 to 9 every second. When a carry is generated, 1 is added to the tens place.	R/W
b6 to b4	SEC10[2:0]	10-Second Count	Counts from 0 to 5 for 60-second counting.	R/W
b7	—	Reserved	Set this bit to 0. It is read as the set value.	R/W

The RSECCNT counter is used for setting and counting the BCD-coded second value. It counts carries generated once per second in the 64-Hz counter.
The setting range is decimal 00 to 59. The RTC will not operate normally if any other value is set. Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.
After writing to the RSECCNT register, confirm that its value has actually changed before proceeding with further processing. Refer to section 28.6.5, Notes on Writing to and Reading from Registers for notes on accessing registers.

(2) In binary count mode:

Address(es): RTC.BCNT0 0008 C402h



x: Undefined

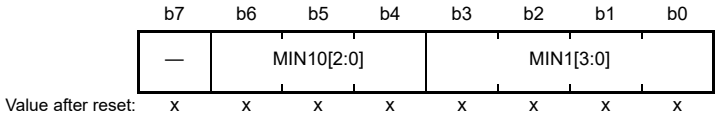
The BCNT0 counter is a readable/writable 32-bit binary counter b7 to b0.
The 32-bit binary counter performs count operation by a carry generating for each second of the 64-Hz counter.
Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.
To read this counter, follow the procedure in section 28.3.5, Reading 64-Hz Counter and Time.

Downloaded from Arrow.com.

28.2.3 Minute Counter (RMINCNT)/Binary Counter 1 (BCNT1)

(1) In calendar count mode:

Address(es): RTC.RMINCNT 0008 C404h



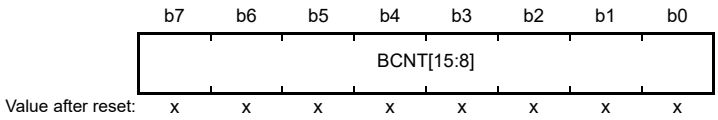
x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	MIN1[3:0]	1-Minute Count	Counts from 0 to 9 every minute. When a carry is generated, 1 is added to the tens place.	R/W
b6 to b4	MIN10[2:0]	10-Minute Count	Counts from 0 to 5 for 60-minute counting.	R/W
b7	—	Reserved	Set this bit to 0. It is read as the set value.	R/W

The RMINCNT counter is used for setting and counting the BCD-coded minute value. It counts carries generated once per minute in the second counter.
A value from 00 through 59 (in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly. Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.
After writing to the RMINCNT counter, confirm that its value has actually changed before proceeding with further processing. Refer to section 28.6.5, Notes on Writing to and Reading from Registers for notes on accessing registers.

(2) In binary count mode:

Address(es): RTC.BCNT1 0008 C404h



x: Undefined

The BCNT1 counter is a readable/writable 32-bit binary counter b15 to b8.
The 32-bit binary counter performs count operation by a carry generating for each second of the 64-Hz counter.
Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.
To read this counter, follow the procedure in section 28.3.5, Reading 64-Hz Counter and Time.

Downloaded from Arrow.com.

28.2.4 Hour Counter (RHRCNT)/Binary Counter 2 (BCNT2)

(1) In calendar count mode:

Address(es): RTC.RHRCNT 0008 C406h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	PM	HR10[1:0]		HR1[3:0]			
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	HR1[3:0]	1-Hour Count	Counts from 0 to 9 once per hour. When a carry is generated, 1 is added to the tens place.	R/W
b5, b4	HR10[1:0]	10-Hour Count	Counts from 0 to 2 once per carry from the ones place.	R/W
b6	PM	PM	Time Counter Setting for a.m./p.m. 0: a.m. 1: p.m.	R/W
b7	—	Reserved	Set this bit to 0. It is read as the set value.	R/W

The RHRCNT counter is used for setting and counting the BCD-coded hour value. It counts carries generated once per hour in the minute counter.
The specifiable time differs according to the setting in the hours mode bit (RCR2.HR24).
When the RCR2.HR24 bit is 0: From 00 to 11 (in BCD)
When the RCR2.HR24 bit is 1: From 00 to 23 (in BCD)
If a value outside of this range is specified, the RTC does not operate correctly. Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.
The PM bit is only enabled when the RCR2.HR24 bit is 0. Otherwise, the setting in the PM bit has no effect.
After writing to the RHRCNT counter, confirm that its value has actually changed before proceeding with further processing. Refer to section 28.6.5, Notes on Writing to and Reading from Registers for notes on accessing registers.

(2) In binary count mode:

Address(es): RTC.BCNT2 0008 C406h

	b7	b6	b5	b4	b3	b2	b1	b0
	BCNT[23:16]							
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

The BCNT2 counter is a readable/writable 32-bit binary counter b23 to b16.
The 32-bit binary counter performs count operation by a carry generating for each second of the 64-Hz counter.
Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.
To read this counter, follow the procedure in section 28.3.5, Reading 64-Hz Counter and Time.

Downloaded from Arrow.com.

28.2.5 Day-of-Week Counter (RWKCNT)/Binary Counter 3 (BCNT3)

(1) In calendar count mode:

Address(es): RTC.RWKCNT 0008 C408h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	DAYW[2:0]		
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	DAYW[2:0]	Day-of-Week Counting	b2 b0 0 0 0: Sunday 0 0 1: Monday 0 1 0: Tuesday 0 1 1: Wednesday 1 0 0: Thursday 1 0 1: Friday 1 1 0: Saturday 1 1 1: Setting Prohibited	R/W
b7 to b3	—	Reserved	Set these bits to 0. They are read as the set value.	R/W

The RWKCNT counter is used for setting and counting in the coded day-of-week value. It counts carries generated once per day in the hour counter.
A value from 0 through 6 can be specified; if a value outside of this range is specified, the RTC does not operate correctly. Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.
Refer to section 28.6.5, Notes on Writing to and Reading from Registers for notes on accessing registers.

(2) In binary count mode:

Address(es): RTC.BCNT3 0008 C408h

	b7	b6	b5	b4	b3	b2	b1	b0
	BCNT[31:24]							
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

The BCNT3 counter is a readable/writable 32-bit binary counter b31 to b24.
The 32-bit binary counter performs count operation by a carry generating for each second of the 64-Hz counter.
Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.
To read this counter, follow the procedure in section 28.3.5, Reading 64-Hz Counter and Time.

Downloaded from Arrow.com.

28.2.6 Date Counter (RDAYCNT)

Address(es): RTC.RDAYCNT 0008 C40Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	DATE10[1:0]		DATE1[3:0]			
Value after reset:	0	0	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	DATE1[3:0]	1-Day Count	Counts from 0 to 9 once per day. When a carry is generated, 1 is added to the tens place.	R/W
b5, b4	DATE10[1:0]	10-Day Count	Counts from 0 to 3 once per carry from the ones place.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The RDAYCNT counter is used in calendar count mode.
The RDAYCNT counter is used for setting and counting the BCD-coded date value. It counts carries generated once per day in the hour counter. The count operation depends on the month and whether the year is a leap year.
Leap years are determined according to whether the year counter (RYRCNT) value is divisible by 400, 100, and 4.
A value from 01 through 31 (in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly. (When specifying a value, note that the range of specifiable days depends on the month and whether the year is a leap year.) Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.
After writing to the RHRCNT counter, confirm that its value has actually changed before proceeding with further processing. Refer to section 28.6.5, Notes on Writing to and Reading from Registers for notes on accessing registers.

Downloaded from Arrow.com.

28.2.7 Month Counter (RMONCNT)

Address(es): RTC.RMONCNT 0008 C40Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	MON10	MON1[3:0]			
Value after reset:	0	0	0	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	MON1[3:0]	1-Month Count	Counts from 0 to 9 once per month. When a carry is generated, 1 is added to the tens place.	R/W
b4	MON10	10-Month Count	Counts from 0 to 1 once per carry from the ones place.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The RMONCNT counter is used in calendar count mode.
The RMONCNT counter is used for setting and counting the BCD-coded month value. It counts carries generated once per month in the date counter.
A value from 01 through 12 (in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly. Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.
After writing to the RMONCNT counter, confirm that its value has actually changed before proceeding with further processing. Refer to section 28.6.5, Notes on Writing to and Reading from Registers for notes on accessing registers.

Downloaded from Arrow.com.

28.2.8 Year Counter (RYRCNT)

Address(es): RTC.RYRCNT 0008 C40Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	YR10[3:0]				YR1[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	YR1[3:0]	1-Year Count	Counts from 0 to 9 once per year. When a carry is generated, 1 is added to the tens place.	R/W
b7 to b4	YR10[3:0]	10-Year Count	Counts from 0 to 9 once per carry from ones place. When a carry is generated in the tens place, 1 is added to the hundreds place.	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The RYRCNT counter is used in calendar count mode.

The RYRCNT counter is used for setting and counting the BCD-coded year value. It counts carries generated once per year in the month counter.

A value from 00 through 99 (in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly. Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.

After writing to the RYRCNT counter, confirm that its value has actually changed before proceeding with further processing. Refer to section 28.6.5, Notes on Writing to and Reading from Registers for notes on accessing registers.

Downloaded from Arrow.com.

28.2.9 Second Alarm Register (RSECAR)/Binary Counter 0 Alarm Register (BCNT0AR)

(1) In calendar count mode:

Address(es): RTC.RSECAR 0008 C410h

	b7	b6	b5	b4	b3	b2	b1	b0
	ENB	SEC10[2:0]			SEC1[3:0]			
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	SEC1[3:0]	1 Second	Value for the ones place of seconds	R/W
b6 to b4	SEC10[2:0]	10 Seconds	Value for the tens place of seconds	R/W
b7	ENB	ENB	0: The register value is not compared with the RSECCNT counter value. 1: The register value is compared with the RSECCNT counter value.	R/W

The RSECAR register is an alarm register corresponding to the BCD-coded second counter RSECCNT. When the ENB bit is set to 1, the RSECAR value is compared with the RSECCNT value. From among the alarm registers (RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, and RYRAREN), only those selected with the ENB bits set to 1 are compared with the corresponding counters. When the respective values all match, the IR flag corresponding to the ALM interrupt is set to 1.
RSECAR values from 00 through 59 (in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly.
After writing to the RSECAR register, confirm that its value has actually changed before proceeding with further processing. Refer to section 28.6.5, Notes on Writing to and Reading from Registers for notes on accessing registers.
This register is set to 00h by an RTC software reset.

(2) In binary count mode:

Address(es): RTC.BCNT0AR 0008 C410h

	b7	b6	b5	b4	b3	b2	b1	b0
	BCNTAR[7:0]							
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

The BCNT0AR counter is a readable/writable alarm register corresponding to 32-bit binary counter b7 to b0.
This register is set to 00h by an RTC software reset.

Downloaded from Arrow.com.

28.2.10 Minute Alarm Register (RMINAR)/Binary Counter 1 Alarm Register (BCNT1AR)

(1) In calendar count mode:

Address(es): RTC.RMINAR 0008 C412h

	b7	b6	b5	b4	b3	b2	b1	b0
	ENB	MIN10[2:0]			MIN1[3:0]			
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	MIN1[3:0]	1 Minute	Value for the ones place of minutes	R/W
b6 to b4	MIN10[2:0]	10 Minutes	Value for the tens place of minutes	R/W
b7	ENB	ENB	0: The register value is not compared with the RMINCNT counter value. 1: The register value is compared with the RMINCNT counter value.	R/W

The RMINAR register is an alarm register corresponding to the BCD-coded minute counter RMINCNT. When the ENB bit is set to 1, the RMINAR value is compared with the RMINCNT value. From among the alarm registers (RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, and RYRAREN), only those selected with the ENB bits set to 1 are compared with the corresponding counters. When the respective values all match, the IR flag corresponding to the ALM interrupt is set to 1.
RMINAR values from 00 through 59 (in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly.
After writing to the RMINAR register, confirm that its value has actually changed before proceeding with further processing. Refer to section 28.6.5, Notes on Writing to and Reading from Registers for notes on accessing registers.
This register is set to 00h by an RTC software reset.

(2) In binary count mode:

Address(es): RTC.BCNT1AR 0008 C412h

	b7	b6	b5	b4	b3	b2	b1	b0
	BCNTAR[15:8]							
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

The BCNT1AR counter is a readable/writable alarm register corresponding to 32-bit binary counter b15 to b8.
This register is set to 00h by an RTC software reset.

Downloaded from Arrow.com.

28.2.11 Hour Alarm Register (RHRAR)/Binary Counter 2 Alarm Register (BCNT2AR)

(1) In calendar count mode:

Address(es): RTC.RHRAR 0008 C414h

	b7	b6	b5	b4	b3	b2	b1	b0
	ENB	PM	HR10[1:0]		HR1[3:0]			
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	HR1[3:0]	1 Hour	Value for the ones place of hours	R/W
b5, b4	HR10[1:0]	10 Hours	Value for the tens place of hours	R/W
b6	PM	PM	Time Alarm Setting for a.m./p.m. 0: a.m. 1: p.m.	R/W
b7	ENB	ENB	0: The register value is not compared with the RHRCNT counter value. 1: The register value is compared with the RHRCNT counter value.	R/W

The RHRAR register is an alarm register corresponding to the BCD-coded hour counter RHRCNT. When the ENB bit is set to 1, the RHRAR value is compared with the RHRCNT value. From among the alarm registers (RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, and RYRAREN), only those selected with the ENB bits set to 1 are compared with the corresponding counters. When the respective values all match, the IR flag corresponding to the ALM interrupt is set to 1.
The specifiable time differs according to the setting in the hours mode bit (RCR2.HR24).
When the RCR2.HR24 bit is 0: From 00 to 11 (in BCD)
When the RCR2.HR24 bit is 1: From 00 to 23 (in BCD)
If a value outside of this range is specified, the RTC does not operate correctly.
When the RCR2.HR24 bit is 0, be sure to set the PM bit.
When the RCR2.HR24 bit is 1, the setting in the PM bit has no effect.
After writing to the RHRAR register, confirm that its value has actually changed before proceeding with further processing. Refer to section 28.6.5, Notes on Writing to and Reading from Registers for notes on accessing registers.
This register is set to 00h by an RTC software reset.

(2) In binary count mode:

Address(es): RTC.BCNT2AR 0008 C414h

	b7	b6	b5	b4	b3	b2	b1	b0
	BCNTAR[23:16]							
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

The BCNT2AR counter is a readable/writable alarm register corresponding to 32-bit binary counter b23 to b16.
This register is set to 00h by an RTC software reset.

Downloaded from Arrow.com.

28.2.12 Day-of-Week Alarm Register (RWKAR)/Binary Counter 3 Alarm Register (BCNT3AR)

(1) In calendar count mode:

Address(es): RTC.RWKAR 0008 C416h

	b7	b6	b5	b4	b3	b2	b1	b0
	ENB	—	—	—	—	DAYW[2:0]		
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	DAYW[2:0]	Day-of-Week Setting	b2 b0 0 0 0: Sunday 0 0 1: Monday 0 1 0: Tuesday 0 1 1: Wednesday 1 0 0: Thursday 1 0 1: Friday 1 1 0: Saturday 1 1 1: Setting Prohibited	R/W
b6 to b3	—	Reserved	Set these bits to 0. They are read as the set value.	R/W
b7	ENB	ENB	0: The register value is not compared with the RWKCNT counter value. 1: The register value is compared with the RWKCNT counter value.	R/W

The RWKAR register is an alarm register corresponding to the coded day-of-week counter RWKCNT. When the ENB bit is set to 1, the RWKAR value is compared with the RWKCNT value. From among the alarm registers (RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, and RYRAREN), only those selected with the ENB bits set to 1 are compared with the corresponding counters. When the respective values all match, the IR flag corresponding to the ALM interrupt is set to 1.

RWKAR values from 0 through 6 (in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly.

After writing to the RWKAR register, confirm that its value has actually changed before proceeding with further processing. Refer to section 28.6.5, Notes on Writing to and Reading from Registers for notes on accessing registers.

This register is set to 00h by an RTC software reset.

(2) In binary count mode:

Address(es): RTC.BCNT3AR 0008 C416h

	b7	b6	b5	b4	b3	b2	b1	b0
	BCNTAR[31:24]							
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

The BCNT3AR counter is a readable/writable alarm register corresponding to 32-bit binary counter b31 to b24.

This register is set to 00h by an RTC software reset.

Downloaded from Arrow.com.

28.2.13 Date Alarm Register (RDAYAR)/Binary Counter 0 Alarm Enable Register (BCNT0AER)

(1) In calendar count mode:

Address(es): RTC.RDAYAR 0008 C418h

	b7	b6	b5	b4	b3	b2	b1	b0
	ENB	—	DATE10[1:0]		DATE1[3:0]			
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	DATE1[3:0]	1 Day	Value for the ones place of days	R/W
b5, b4	DATE10[1:0]	10 Days	Value for the tens place of days	R/W
b6	—	Reserved	Set this bit to 0. It is read as the set value.	R/W
b7	ENB	ENB	0: The register value is not compared with the RDAYCNT counter value. 1: The register value is compared with the RDAYCNT counter value.	R/W

The RDAYAR register is an alarm register corresponding to the BCD-coded date counter RDAYCNT. When the ENB bit is set to 1, the RDAYAR value is compared with the RDAYCNT value. From among the alarm registers (RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, and RYRAREN), only those selected with the ENB bits set to 1 are compared with the corresponding counters. When the respective values all match, the IR flag corresponding to the ALM interrupt is set to 1.

RDAYAR values from 01 through 31 (in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly.

After writing to the RDAYAR register, confirm that its value has actually changed before proceeding with further processing. Refer to section 28.6.5, Notes on Writing to and Reading from Registers for notes on accessing registers.

This register is set to 00h by an RTC software reset.

(2) In binary count mode:

Address(es): RTC.BCNT0AER 0008 C418h

	b7	b6	b5	b4	b3	b2	b1	b0
	ENB[7:0]							
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

The BCNT0AER register is a readable/writable register for setting the alarm enable corresponding to 32-bit binary counter b7 to b0. Among the ENB[31:0] bits, the binary counter (BCNT[31:0]) corresponding to the bits which are set to 1 and the binary alarm register (BCNTAR[31:0]) are compared, and when all match, the IR flag corresponding to the ALM interrupt becomes 1.

This register is set to 00h by an RTC software reset.

Downloaded from Arrow.com.

28.2.14 Month Alarm Register (RMONAR)/Binary Counter 1 Alarm Enable Register (BCNT1AER)

(1) In calendar count mode:

Address(es): RTC.RMONAR 0008 C41Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	ENB	—	—	MON10	MON1[3:0]			
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	MON1[3:0]	1 Month	Value for the ones place of months	R/W
b4	MON10	10 Months	Value for the tens place of months	R/W
b6, b5	—	Reserved	Set these bits to 0. They are read as the set value.	R/W
b7	ENB	ENB	0: The register value is not compared with the RMONCNT counter value. 1: The register value is compared with the RMONCNT counter value.	R/W

The RMONAR register is an alarm register corresponding to the BCD-coded month counter RMONCNT. When the ENB bit is set to 1, the RMONAR value is compared with the RMONCNT value. From among the alarm registers (RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, and RYRAREN), only those selected with the ENB bits set to 1 are compared with the corresponding counters. When the respective values all match, the IR flag corresponding to the ALM interrupt is set to 1.
RMONAR values from 01 through 12 (in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly.
After writing to the RMONAR register, confirm that its value has actually changed before proceeding with further processing. Refer to section 28.6.5, Notes on Writing to and Reading from Registers for notes on accessing registers.
This register is set to 00h by an RTC software reset.

(2) In binary count mode:

Address(es): RTC.BCNT1AER 0008 C41Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	ENB[15:8]							
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

The BCNT1AER register is a readable/writable register for setting the alarm enable corresponding to 32-bit binary counter b15 to b8. Among the ENB[31:0] bits, the binary counter (BCNT[31:0]) corresponding to the bits which are set to 1 and the binary alarm register (BCNTAR[31:0]) are compared, and when all match, the IR flag corresponding to the ALM interrupt becomes 1.
This register is set to 00h by an RTC software reset.

Downloaded from Arrow.com.

28.2.15 Year Alarm Register (RYRAR)/Binary Counter 2 Alarm Enable Register (BCNT2AER)

(1) In calendar count mode:

Address(es): RTC.RYRAR 0008 C41Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	YR10[3:0]				YR1[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	YR1[3:0]	1 Year	Value for the ones place of years	R/W
b7 to b4	YR10[3:0]	10 Years	Value for the tens place of years	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The RYRAR register is an alarm register corresponding to the BCD-coded year counter RYRCNT.
RYRAR values from 00 through 99 (in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly.
After writing to the RYRAR register, confirm that its value has actually changed before proceeding with further processing. Refer to section 28.6.5, Notes on Writing to and Reading from Registers for notes on accessing registers.
This register is set to 0000h by an RTC software reset.

(2) In binary count mode:

Address(es): RTC.BCNT2AER 0008 C41Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	ENB[23:16]							
Value after reset:	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x

x: Undefined

The BCNT2AER register is a readable/writable register for setting the alarm enable corresponding to 32-bit binary counter b23 to b16. Among the ENB[31:0] bits, the binary counter (BCNT[31:0]) corresponding to the bits which are set to 1 and the binary alarm register (BCNTAR[31:0]) are compared, and when all match, the IR flag corresponding to the ALM interrupt becomes 1.
This register is set to 0000h by an RTC software reset.

Downloaded from Arrow.com.

28.2.16 Year Alarm Enable Register (RYRAREN)/Binary Counter 3 Alarm Enable Register (BCNT3AER)

(1) In calendar count mode:

Address(es): RTC.RYRAREN 0008 C41Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	ENB	—	—	—	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	—	Reserved	Set these bits to 0. They are read as the set value.	R/W
b7	ENB	ENB	0: The register value is not compared with the RYRCNT counter value. 1: The register value is compared with the RYRCNT counter value.	R/W

When the ENB bit in the RYRAREN register is set to 1, the RYRAR value is compared with the RYRCNT value. From among the alarm registers (RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, and RYRAREN), only those selected with the ENB bits set to 1 are compared with the corresponding counters. When the respective values all match, the IR flag corresponding to the ALM interrupt is set to 1.
This register is set to 00h by an RTC software reset.

(2) In binary count mode:

Address(es): RTC.BCNT3AER 0008 C41Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	ENB[31:24]							
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

The BCNT3AER register is a readable/writable register for setting the alarm enable corresponding to 32-bit binary counter b31 to b24. Among the ENB[31:0] bits, the binary counter (BCNT[31:0]) corresponding to the bits which are set to 1 and the binary alarm register (BCNTAR[31:0]) are compared, and when all match, the IR flag corresponding to the ALM interrupt becomes 1.
This register is set to 00h by an RTC software reset.

Downloaded from Arrow.com.

28.2.17 RTC Control Register 1 (RCR1)

Address(es): RTC.RCR1 0008 C422h

	b7	b6	b5	b4	b3	b2	b1	b0
	PES[3:0]				RTCOS	PIE	CIE	AIE
Value after reset:	x	x	x	x	0	x	0	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	AIE	Alarm Interrupt Enable	0: An alarm interrupt request is disabled. 1: An alarm interrupt request is enabled.	R/W
b1	CIE	Carry Interrupt Enable	0: A carry interrupt request is disabled. 1: A carry interrupt request is enabled.	R/W
b2	PIE	Periodic Interrupt Enable	0: A periodic interrupt request is disabled. 1: A periodic interrupt request is enabled.	R/W
b3	RTCOS	RTCOUT Output Select	0: RTCOUT outputs 1 Hz. 1: RTCOUT outputs 64 Hz.	R/W
b7 to b4	PES[3:0]	Periodic Interrupt Select	b7 b4 0 1 1 0: A periodic interrupt is generated every 1/256 second. 0 1 1 1: A periodic interrupt is generated every 1/128 second. 1 0 0 0: A periodic interrupt is generated every 1/64 second. 1 0 0 1: A periodic interrupt is generated every 1/32 second. 1 0 1 0: A periodic interrupt is generated every 1/16 second. 1 0 1 1: A periodic interrupt is generated every 1/8 second. 1 1 0 0: A periodic interrupt is generated every 1/4 second. 1 1 0 1: A periodic interrupt is generated every 1/2 second. 1 1 1 0: A periodic interrupt is generated every 1 second. 1 1 1 1: A periodic interrupt is generated every 2 seconds. Other than above: No periodic interrupts are generated.	R/W

The RCR1 register is used in both calendar count mode and in binary count mode.
Bits AIE, PIE, and PES[3:0] are updated synchronously with the count source. When the RCR1 register is modified, check that all the bits have been updated before proceeding to the next processing.

AIE Bit (Alarm Interrupt Enable)

This bit enables or disables alarm interrupt requests.

CIE Bit (Carry Interrupt Enable)

This bit enables and disables interrupt requests when a carry to the RSECCNT/BCNT0 register occurs, or when a carry to the 64-Hz counter (R64CNT) occurs while reading the 64-Hz counter.

PIE Bit (Periodic Interrupt Enable)

This bit enables or disabled a periodic interrupt.

RTCOS Bit (RTCOUT Output Select)

This bit selects the RTCOUT output period. The RTCOS bit must be rewritten while count operation is stopped (the RCR2.START bit is 0) and RTCOUT output is disabled (the RCR2.RTCOE bit is 0). When the RTCOUT is output to an external pin, the RCR2.RTCOE bit must be enabled. For details on controlling I/O ports, refer to section 22.3.1, Procedure for Specifying Input/Output Pin Function.

PES[3:0] Bits (Periodic Interrupt Select)

These bits specify the period for the periodic interrupt. A periodic interrupt is generated with the period specified by these bits.

Downloaded from Arrow.com.

28.2.18 RTC Control Register 2 (RCR2)

Address(es): RTC.RCR2 0008 C424h

	b7	b6	b5	b4	b3	b2	b1	b0
	CNTMD	HR24	AADJP	AADJE	RTCOE	ADJ30	RESET	START
Value after reset:	x	x	x	x	0	0	0	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	START	Start*3	0: Prescaler and counter are stopped. 1: Prescaler and counter operate normally.	R/W
b1	RESET	RTC Software Reset	• In writing 0: Writing is invalid. 1: The prescaler and the target registers for RTC software reset*1 are initialized. • In reading 0: In normal time operation, or an RTC software reset has completed. 1: During an RTC software reset	R/W
b2	ADJ30	30-Second Adjustment*2	• In writing 0: Writing is invalid. 1: 30-second adjustment is executed. • In reading 0: In normal time operation, or 30-second adjustment has completed. 1: During 30-second adjustment	R/W
b3	RTCOE	RTCOUT Output Enable	0: RTCOUT output disabled. 1: RTCOUT output enabled.	R/W
b4	AADJE	Automatic Adjustment Enable*3	0: Automatic adjustment is disabled. 1: Automatic adjustment is enabled.	R/W
b5	AADJP	Automatic Adjustment Period Select*3	0: The RADJ.ADJ[5:0] setting value is adjusted from the count value of the prescaler every minute (every 32 seconds in binary counter mode). 1: The RADJ.ADJ[5:0] setting value is adjusted from the count value of the prescaler every 10 seconds (every 8 seconds in binary counter mode).	R/W
b6	HR24	Hours Mode*2, *3	0: The RTC operates in 12-hour mode. 1: The RTC operates in 24-hour mode.	R/W
b7	CNTMD	Count Mode Select*3	0: The calendar count mode. 1: The binary count mode.	R/W

Note 1. R64CNT, RSECAR/BCNT0AR, RMINAR/BCNT1AR, RHRAR/BCNT2AR, RWKAR/BCNT3AR, RDAYAR/BCNT0AER, RMONAR/BCNT1AER, RYRAR/BCNT2AER, RYRAREN/BCNT3AER, RADJ, RTCCRn, RSECCPn/BCNT0CPn, RMINCPn/BCNT1CPn, RHRCPn/BCNT2CPn, RDAYCPn/BCNT3CPn, RMONCPn, RCR2.ADJ30, RCR2.AADJE, RCR2.AADJP

Note 2. This bit is reserved in binary counter mode. The write value should be 0.

Note 3. After writing to this bit, confirm that its value has actually changed before proceeding with further processing. Refer to section 28.6.5, Notes on Writing to and Reading from Registers regarding changes to the values of the AADJE, AADJP, and HR24 bits.

The RCR2 register is related to hours mode, automatic adjustment function, enabling the RTCOUT output, 30-second adjustment, RTC software reset, and controlling count operation.

START Bit (Start)

This bit stops or restarts the prescaler or counter (clock) operation.

The START bit is updated in synchronization with the count source. When the START bit is modified, check that the bit is updated before proceeding to the next processing.

Downloaded from Arrow.com.

RESET Bit (RTC Software Reset)

This bit initializes the prescaler and registers to be reset by RTC software reset.
When 1 is written to the RESET bit, the initialization starts in synchronization with the count source. When the initialization is completed, the RESET bit is automatically set to 0.
When 1 is written to the RESET bit, check that the bit is set to 0, and then make next settings.

ADJ30 Bit (30-Second Adjustment)

This bit is for 30-second adjustment.
When 1 is written to the ADJ30 bit, the RSECCNT value of 30 seconds or less is rounded down to 00 second and the value of 30 seconds or more is rounded up to 1 minute.
The 30-second adjustment is performed in synchronization with the count source. When 1 is written to this bit, the ADJ30 bit is automatically set to 0 after the 30-second adjustment is completed. In case when 1 is written to the ADJ30 bit, check that the bit is set to 0, and then make next settings.
When the 30-second adjustment is performed, the prescaler and R64CNT are also reset.
The ADJ30 bit is set to 0 by an RTC software reset.
This bit is reserved in binary counter mode. The write value should be 0.

RTCOE Bit (RTCOUT Output Enable)

This bit enables output of a 1-Hz/64-Hz clock signal from the RTCOUT pin.
Use the START bit to stop counting by the counters before changing the value of the RTCOE bit. Do not stop counting (write 0 to the START bit) and change the value of the RTCOE bit at the same time.
When RTCOUT signal is to be output from an external pin, set the RTCOE bit to 1 and set up the port control for the pin.

AADJE Bit (Automatic Adjustment Enable)

This bit controls (enables or disables) automatic adjustment.
Set the plus–minus bits (RADJ.PMADJ[1:0]) to 00b (adjustment is not performed) before changing the value of the AADJE bit.
The AADJE bit is set to 0 by an RTC software reset.

AADJP Bit (Automatic Adjustment Period Select)

This bit selects the automatic-adjustment period.
Set the plus–minus bits (RADJ.PMADJ[1:0]) to 00b (adjustment is not performed) before changing the value of the AADJP bit.
The AADJP bit is set to 0 by an RTC software reset.

HR24 Bit (Hours Mode)

This bit specifies whether the RTC will operate in 12- or 24-hour mode.
Use the START bit to stop counting by the counters before changing the value of the HR24 bit. Do not stop counting (write 0 to the START bit) and change the value of the HR24 bit at the same time.
This bit is reserved in binary counter mode. The write value should be 0.

CNTMD Bit (Count Mode Select)

This bit specifies whether the RTC count mode is operated in calendar count mode or in binary count mode.
After setting the count mode, execute an RTC software reset and start again from the initial settings.
The CNTMD bit is updated in synchronization with the count source, so when the value of the CNTMD bit has been changed, check that the value of the bit has actually been updated before applying the RTC software reset. The count mode changes to that which was specified beforehand in the CNTMD bit after the RTC software reset is applied.
For details on initial settings, refer to section 28.3.1, Outline of Initial Settings of Registers after Power On.

Downloaded from Arrow.com.

28.2.19 RTC Control Register 3 (RCR3)

Address(es): RTC.RCR3 0008 C426h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	RTCDV[2:0]			RTCEN
Value after reset:	0	0	0	0	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	RTCEN	Sub-Clock Oscillator Control	0: Sub-clock oscillator is stopped. 1: Sub-clock oscillator is operating.	R/W
b3 to b1	RTCDV[2:0]	Sub-Clock Oscillator Drive Capacity Control	b3 b1 0 0 0: Setting prohibited 0 0 1: Drive capacity for low CL 0 1 0: Setting prohibited 0 1 1: Setting prohibited 1 0 0: Setting prohibited 1 0 1: Setting prohibited 1 1 0: Drive capacity for standard CL 1 1 1: Setting prohibited	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The RCR3 register is used for controlling the sub-clock oscillator in the clock generation circuit. For details on controlling the sub-clock oscillator, refer to section 9, Clock Generation Circuit.
This register is a function common to calendar count mode and binary count mode.
When this register is modified, check that all the bits have been updated before proceeding to the next processing.

RTCEN Bit (Sub-Clock Oscillator Control)

The RTCEN bit and a clock generation circuit register (the SOSCCR.SOSTP bit) control whether to operate or stop the sub-clock oscillator. If one of the bits is set so as to enable the operation, the sub-clock oscillator runs.
When using the sub-clock as the count source to the RTC, set the sub-clock oscillator using the RTCEN bit.

RTCDV[2:0] Bits (Sub-Clock Oscillator Drive Capacity Control)

These bits control the drive capacity of the sub-clock oscillator. Set the RTCDV[2:0] bits when the SOSCCR.SOSTP bit is 1 and the RCR3.RTCEN bit is 0.

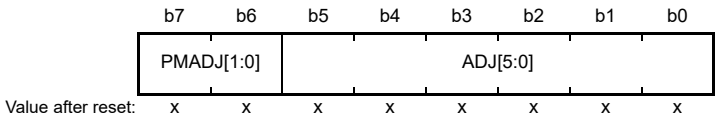
28.2.19.1 Notes on using a low CL crystal unit

When the signal level of any pin near the XCIN or XCOUT pin is changed, the oscillation accuracy of the sub-clock oscillator may be affected. The accuracy is affected differently depending on the board traces and how the signal level of any pin near the XCIN or XCOUT pin is changed. When designing a board using a low CL crystal unit, refer to the application note “Design Guide for Low CL Sub-clock Circuits” (R01AN1012EJ) to reduce the influence from noise.

Downloaded from Arrow.com.

28.2.20 Time Error Adjustment Register (RADJ)

Address(es): RTC.RADJ 0008 C42Eh



x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	ADJ[5:0]	Adjustment Value	These bits specify the adjustment value from the prescaler.	R/W
b7, b6	PMADJ[1:0]	Plus–Minus	b7 b6 0 0: Adjustment is not performed. 0 1: Adjustment is performed by the addition to the prescaler. 1 0: Adjustment is performed by the subtraction from the prescaler. 1 1: Setting prohibited	R/W

Adjustment is performed by the addition to or subtraction from the prescaler.
In case when the automatic adjustment enable (RCR2.AADJE) bit is 0, adjustment is performed when writing to the RADJ.
In case when the RCR2.AADJE bit is 1, adjustment is performed in the interval specified by the automatic adjustment period select (RCR2.AADJP) bit.
The current adjustment by software (disabling automatic adjustment) may be invalid if the following adjustment value is specified within 320 cycles of the count source after the register setting. To perform adjustment consecutively, wait for 320 cycles or more of the count source after the register setting and then specify the next adjustment value.
RADJ is updated in synchronization with the count source. When RADJ is modified, check that all the bits have been updated before continuing with further processing.
This register is set to 00h by an RTC software reset.

ADJ[5:0] Bits (Adjustment Value)

These bits specify the adjustment value (the number of sub-clock cycles) from the prescaler.

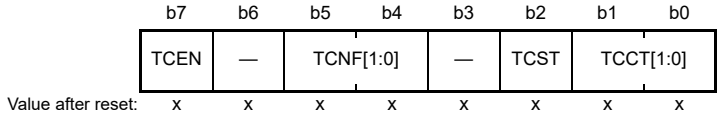
PMADJ[1:0] Bits (Plus–Minus)

These bits select whether the clock is set ahead or back depending on the error-adjustment value set in the ADJ[5:0] bits.

Downloaded from Arrow.com.

28.2.21 Time Capture Control Register n (RTCCRn) (n = 0, 1)

Address(es): RTC.RTCCR0 0008 C440h, RTC.RTCCR1 0008 C442h



x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b1, b0	TCCT[1:0]	Time Capture Control	b1 b0 0 0: No event is detected. 0 1: Rising edge is detected. 1 0: Falling edge is detected. 1 1: Both edges are detected.	R/W
b2	TCST	Time Capture Status	0: No event is detected. 1: An event is detected.*1	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5, b4	TCNF[1:0]	Time Capture Noise Filter Control	b5 b4 0 0: The noise filter is off. 0 1: Setting prohibited 1 0: The noise filter is on (count source). 1 1: The noise filter is on (count source by divided by 32).	R/W
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7	TCEN	Time Capture Event Input Pin Enable	0: The RTCICn pin is disabled as the time capture event input. 1: The RTCICn pin is enabled as the time capture event input.	R/W

Note 1. Indicates that an event has been detected. Writing 1 to this bit has no effect. Writing 0 sets this bit to 0.

The RTCCRn register is used both in calendar count mode and in binary count mode.
RTCCR0 and RTCCR1 control the RTCIC0 and RTCIC1 pins, respectively.
RTCCRn is updated in synchronization with the count source. When RTCCRn is modified, check that all the bits except for the TCST bit have been updated before continuing with further processing.
This register is set to 00h by an RTC software reset.

TCCT[1:0] Bits (Time Capture Control)

These bits control the edge detection of the time capture event input pins (RTCIC0 and RTCIC1). The detection edge is selectable. The TCCT[1:0] bits should be set while the TCEN bit is 1.

TCST Bit (Time Capture Status)

This bit indicates that an event of the time capture event input pins (RTCIC0 and RTCIC1) has been detected.
When the TCST bit is 0, no event is detected.
When the TCST bit is 1, this bit indicates that an event of the corresponding pin has been detected and the capture register is valid. When multiple events have been detected, the capture time for the first event is retained.
If an event is detected while the count operation is stopped (the RCR2.START bit is 0), the captured value is not guaranteed. In this case, set the TCST bit to 0 for deleting the captured value.
Writing 0 sets the TCST bit to 0. In addition, writing any other value except 0 has no effect.
Set the TCST bit while the TCCT[1:0] bits are 00b (no event is detected).
The TCST bit is set to 0 in synchronization with the count source. When the TCST bit is set to 0, check that the bit has been updated before continuing with further processing.

Downloaded from Arrow.com.

TCNF[1:0] Bits (Time Capture Noise Filter Control)

These bits control the noise filter of the time capture event input pins (RTCIC0 and RTCIC1).
When the noise filter is on, the count source divided by 1 or divided by 32 is selectable. In this case, when the input level on the time capture event input pin matches three consecutive times at the set sampling period, the input level is determined.
Set the TCNF[1:0] bits while the TCCT[1:0] bits are 00b (no event is detected). When the noise filter is used, set the TCNF[1:0] bits, wait for three cycles of the specified sampling period, and then set the TCCT[1:0] bits. Set the TCNF[1:0] bits when the TCEN bit is 1.

TCEN Bit (Time Capture Event Input Pin Enable)

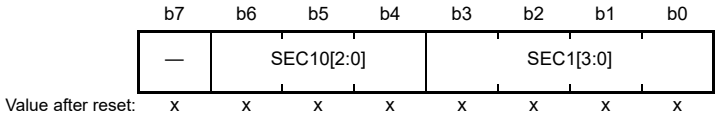
This bit enables or disables the time capture event input pins (RTCIC0 and RTCIC1).
When the sub-clock oscillator is stopped (RCR3.RTCEN bit = 0), the time capture event input pins (RTCIC0 and RTCIC1) are disabled regardless of the value of the TCEN bit.
When the time capture event input pin functions (RTCIC0 and RTCIC1) are allocated to the same pins as other multiplexed functions, make the general input port pin settings for the corresponding pins and then set this bit to 1. If the TCEN bit is set to 0, set also the TCCT[1:0] bits to 00b.

Downloaded from Arrow.com.

28.2.22 Second Capture Register n (RSECCPn) (n = 0, 1)/BCNT0 Capture Register n (BCNT0CPn) (n = 0, 1)

(1) In calendar count mode:

Address(es): RTC.RSECCP0 0008 C452h, RTC.RSECCP1 0008 C462h



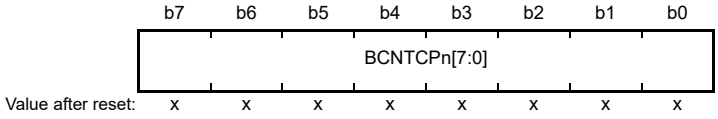
x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	SEC1[3:0]	1-Second Capture	Capture value for the ones place of seconds	R
b6 to b4	SEC10[2:0]	10-Second Capture	Capture value for the tens place of seconds	R
b7	—	Reserved	This bit is read as 0 after an RTC software reset.	R

RSECCPn is a read-only register that captures the RSECCNT value when a time capture event is detected.
The event detection times detected by the RTCIC0 and RTCIC1 pins are stored in the RSECCP0 and RSECCP1 registers, respectively.
This register is set to 00h by an RTC software reset.
Before reading from this register, be sure to stop the time capture event detection through the setting of the RTCCRn.TCCT[1:0] bits.

(2) In binary count mode:

Address(es): RTC.BCNT0CP0 0008 C452h, RTC.BCNT0CP1 0008 C462h



x: Undefined

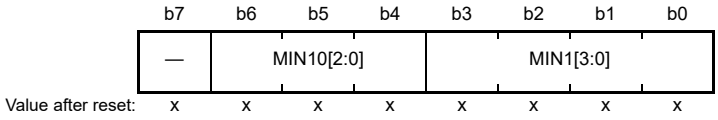
BCNT0CPn is a read-only register that captures the BCNT0 value when a time capture event is detected.
The event detection times detected by the RTCIC0 and RTCIC1 pins are stored in the BCNT0CP0 and BCNT0CP1 registers, respectively.
This register is set to 00h by an RTC software reset.
Before reading from this register, be sure to stop the time capture event detection through the setting of the RTCCRn.TCCT[1:0] bits.

Downloaded from Arrow.com.

28.2.23 Minute Capture Register n (RMINCPn) (n = 0, 1)/BCNT1 Capture Register n (BCNT1CPn) (n = 0, 1)

(1) In calendar count mode:

Address(es): RTC.RMINCP0 0008 C454h, RTC.RMINCP1 0008 C464h



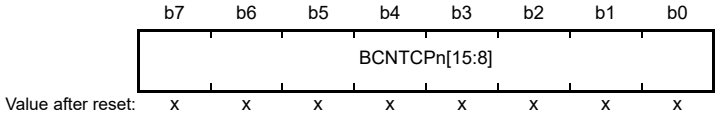
x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	MIN1[3:0]	1-Minute Capture	Capture value for the ones place of minutes	R
b6 to b4	MIN10[2:0]	10-Minute Capture	Capture value for the tens place of minutes	R
b7	—	Reserved	This bit is read as 0 after an RTC software reset.	R

RMINCPn is a read-only register that captures the RMINCNT value when a time capture event is detected.
The event detection times detected by the RTCIC0 and RTCIC1 pins are stored in the RMINCP0 and RMINCP1 registers, respectively.
This register is set to 00h by an RTC software reset.
Before reading from this register, be sure to stop the time capture event detection through the setting of the RTCCRn.TCCT[1:0] bits.

(2) In binary count mode:

Address(es): RTC.BCNT1CP0 0008 C454h, RTC.BCNT1CP1 0008 C464h



x: Undefined

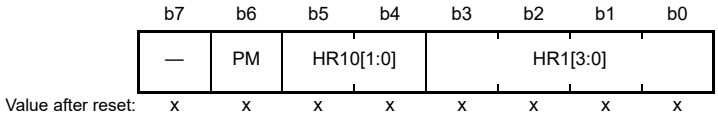
BCNT1CPn is a read-only register that captures the BCNT1 value when a time capture event is detected.
The event detection times detected by the RTCIC0 and RTCIC1 pins are stored in the BCNT1CP0 and BCNT1CP1 registers, respectively.
This register is set to 00h by an RTC software reset.
Before reading from this register, be sure to stop the time capture event detection through the setting of the RTCCRn.TCCT[1:0] bits.

Downloaded from Arrow.com.

28.2.24 Hour Capture Register n (RHRCPn) (n = 0, 1)/BCNT2 Capture Register n (BCNT2CPn) (n = 0, 1)

(1) In calendar count mode:

Address(es): RTC.RHRCP0 0008 C456h, RTC.RHRCP1 0008 C466h



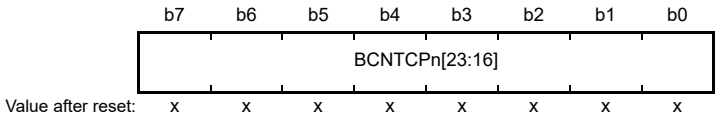
x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	HR1[3:0]	1-Hour Capture	Capture value for the ones place of hours	R
b5, b4	HR10[1:0]	10-Hour Capture	Capture value for the tens place of hours	R
b6	PM	PM	0: a.m. 1: p.m.	R
b7	—	Reserved	This bit is read as 0 after an RTC software reset.	R

RHRCPn is a read-only register that captures the RHRCNT value when a time capture event is detected.
The event detection times detected by the RTCIC0 and RTCIC1 pins are stored in the RHRCP0 and RHRCP1 registers, respectively.
The PM bit is only enabled when the RCR2.HR24 bit is 0 (in 12-hour mode).
This register is set to 00h by an RTC software reset.
Before reading from this register, be sure to stop the time capture event detection through the setting of the RTCCRn.TCCT[1:0] bits.

(2) In binary count mode:

Address(es): RTC.BCNT2CP0 0008 C456h, RTC.BCNT2CP1 0008 C466h



x: Undefined

BCNT2CPn is a read-only register that captures the BCNT2 value when a time capture event is detected.
The event detection times detected by the RTCIC0 and RTCIC1 pins are stored in the BCNT2CP0 and BCNT2CP1 registers, respectively.
This register is set to 00h by an RTC software reset.
Before reading from this register, be sure to stop the time capture event detection through the setting of the RTCCRn.TCCT[1:0] bits.

Downloaded from Arrow.com.

28.2.25 Date Capture Register n (RDAYCPn) (n = 0, 1)/BCNT3 Capture Register n (BCNT3CPn) (n = 0, 1)

(1) In calendar count mode:

Address(es): RTC.RDAYCP0 0008 C45Ah, RTC.RDAYCP1 0008 C46Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	DATE10[1:0]		DATE1[3:0]			
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	DATE1[3:0]	1-Day Capture	Capture value for the ones place of days	R
b5, b4	DATE10[1:0]	10-Day Capture	Capture value for the tens place of days	R
b7, b6	—	Reserved	These bits are read as 0 after an RTC software reset.	R

RDAYCPn is a read-only register that captures the RDAYCNT value when a time capture event is detected.
The event detection times detected by the RTCIC0 and RTCIC1 pins are stored in the RDAYCP0 and RDAYCP1 registers, respectively.
This register is set to 00h by an RTC software reset.
Before reading from this register, be sure to stop the time capture event detection through the setting of the RTCCRn.TCCT[1:0] bits.

(2) In binary count mode:

Address(es): RTC.BCNT3CP0 0008 C45Ah, RTC.BCNT3CP1 0008 C46Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	BCNTCPn[31:24]							
Value after reset:	x	x	x	x	x	x	x	x

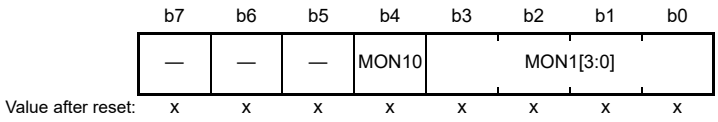
x: Undefined

BCNT3CPn is a read-only register that captures the BCNT3 value when a time capture event is detected.
The event detection times detected by the RTCIC0 and RTCIC1 pins are stored in the BCNT3CP0 and BCNT3CP1 registers, respectively.
This register is set to 00h by an RTC software reset.
Before reading from this register, be sure to stop the time capture event detection through the setting of the RTCCRn.TCCT[1:0] bits.

Downloaded from Arrow.com.

28.2.26 Month Capture Register n (RMONCPn) (n = 0, 1)

Address(es): RTC.RMONCP0 0008 C45Ch, RTC.RMONCP1 0008 C46Ch



x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	MON1[3:0]	1-Month Capture	Capture value for the ones place of months	R
b4	MON10	10-Month Capture	Capture value for the tens place of months	R
b7 to b5	—	Reserved	These bits are read as 0	R

RMONCPn is a read-only register that captures the RMONCNT value when a time capture event is detected.
The event detection times detected by the RTCIC0 and RTCIC1 pins are stored in the RMONCP0 and RMONCP1 registers, respectively.
This register is set to 00h by an RTC software reset.
Before reading from this register, be sure to stop the time capture event detection through the setting of the RTCCRn.TCCT[1:0] bits.

Downloaded from Arrow.com.

28.3 Operation

28.3.1 Outline of Initial Settings of Registers after Power On

After the power is turned on, the initial settings for the clock setting, count mode setting, time error adjustment, time setting, alarm, interrupt, and time capture control register should be performed.

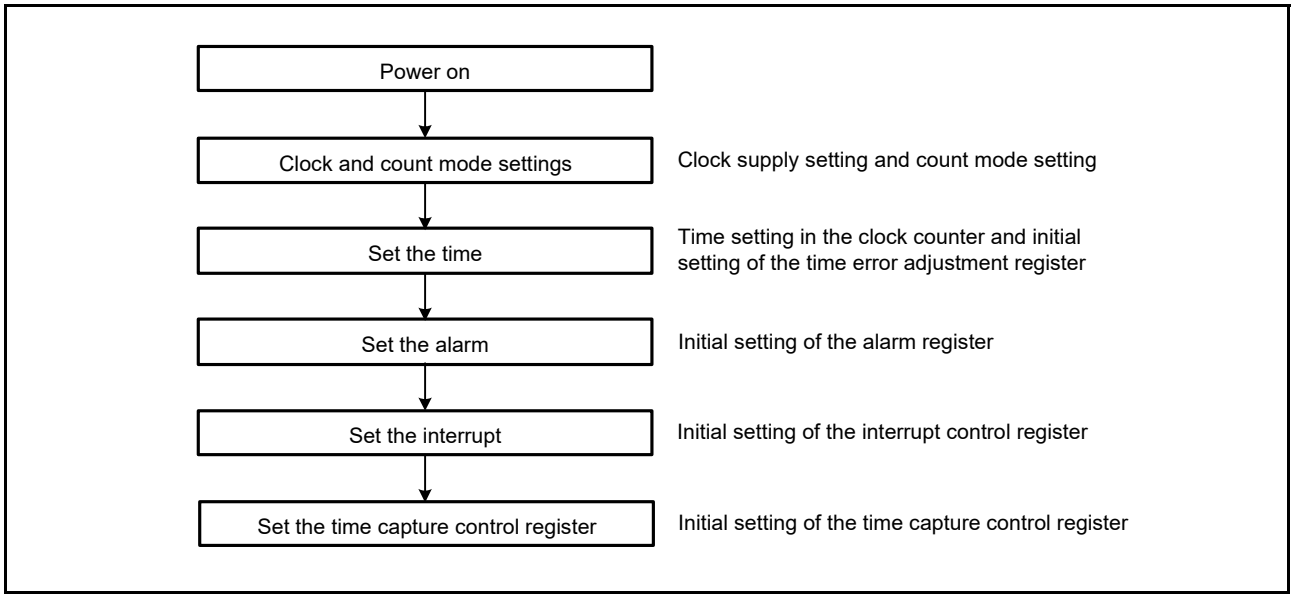
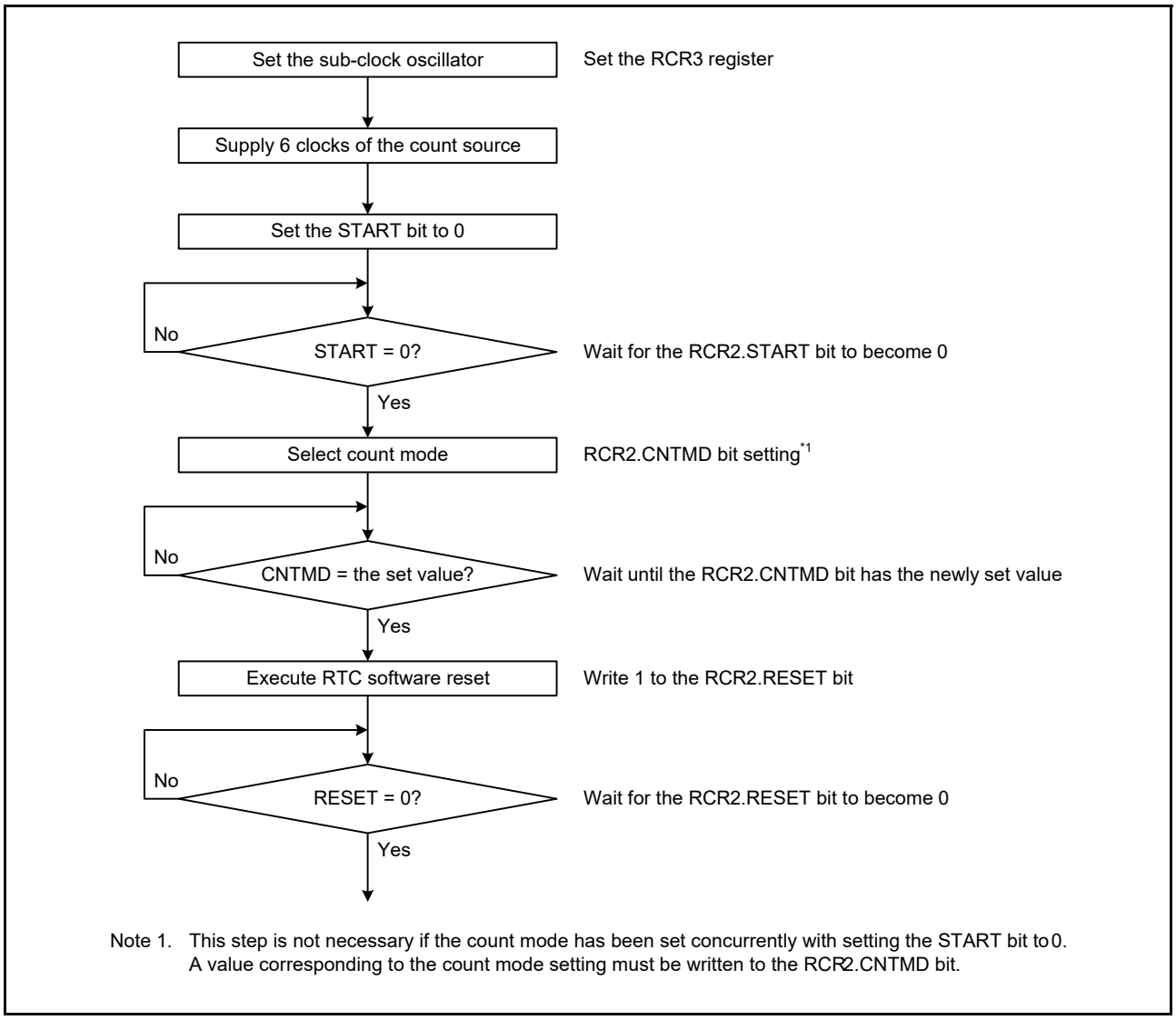


Figure 28.2 Outline of Initial Settings after Power On

Downloaded from Arrow.com.

28.3.2 Clock and Count Mode Setting Procedure

Figure 28.3 shows how to set the clock and the count mode.



Note 1. This step is not necessary if the count mode has been set concurrently with setting the START bit to 0. A value corresponding to the count mode setting must be written to the RCR2.CNTMD bit.

Figure 28.3 Clock and Count Mode Setting Procedure

Downloaded from Arrow.com.

28.3.3 Setting the Time

Figure 28.4 shows how to set the time.

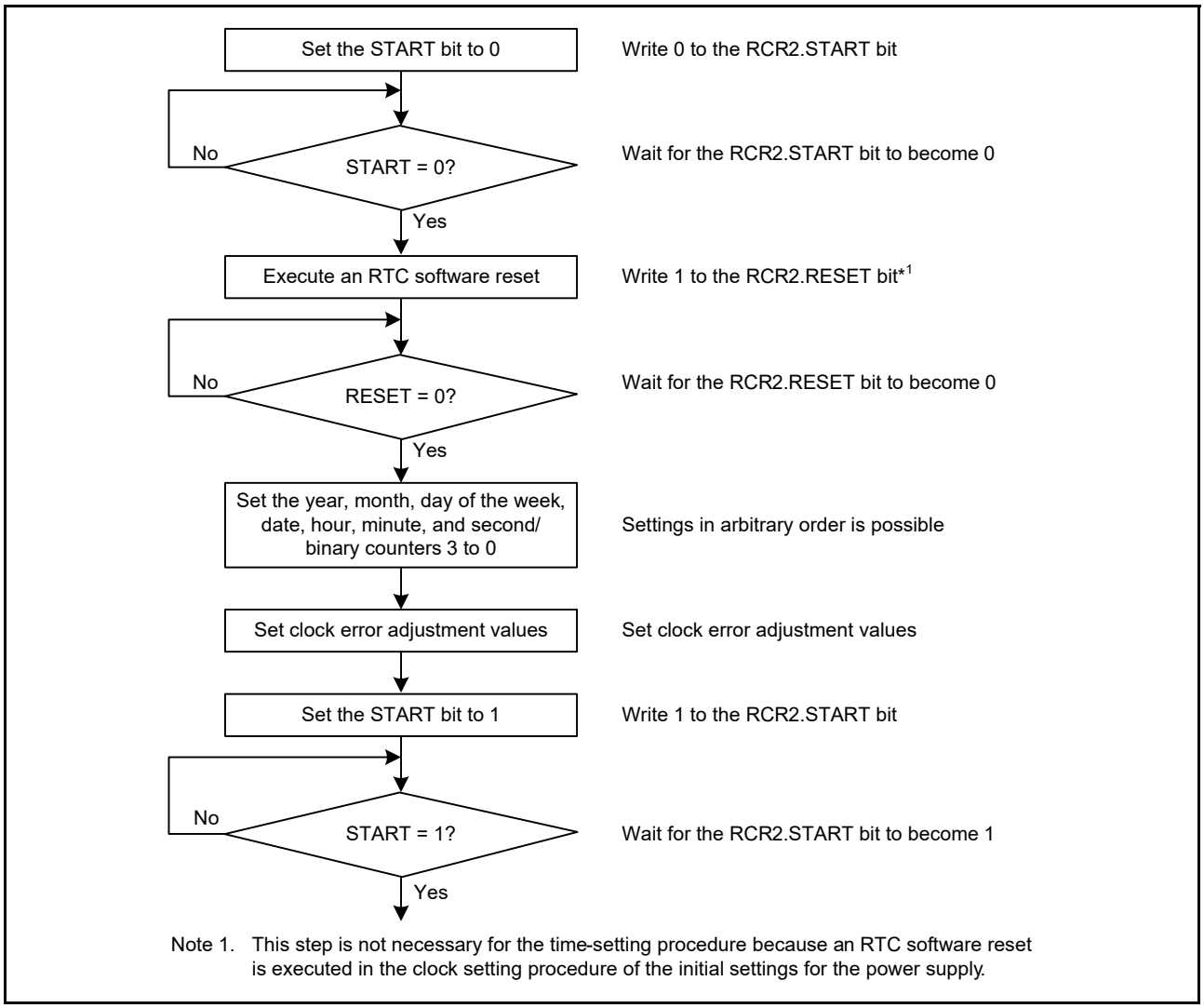


Figure 28.4 Setting the Time

28.3.4 30-Second Adjustment

Figure 28.5 shows how to execute 30-second adjustment.

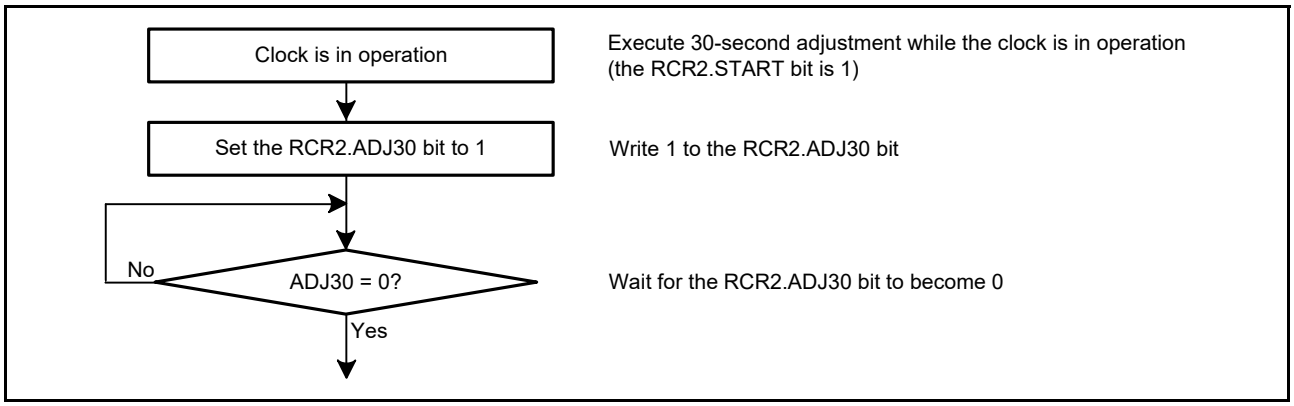


Figure 28.5 30-Second Adjustment

Downloaded from Arrow.com.

28.3.5 Reading 64-Hz Counter and Time

Figure 28.6 shows how to read the 64-Hz counter and time.

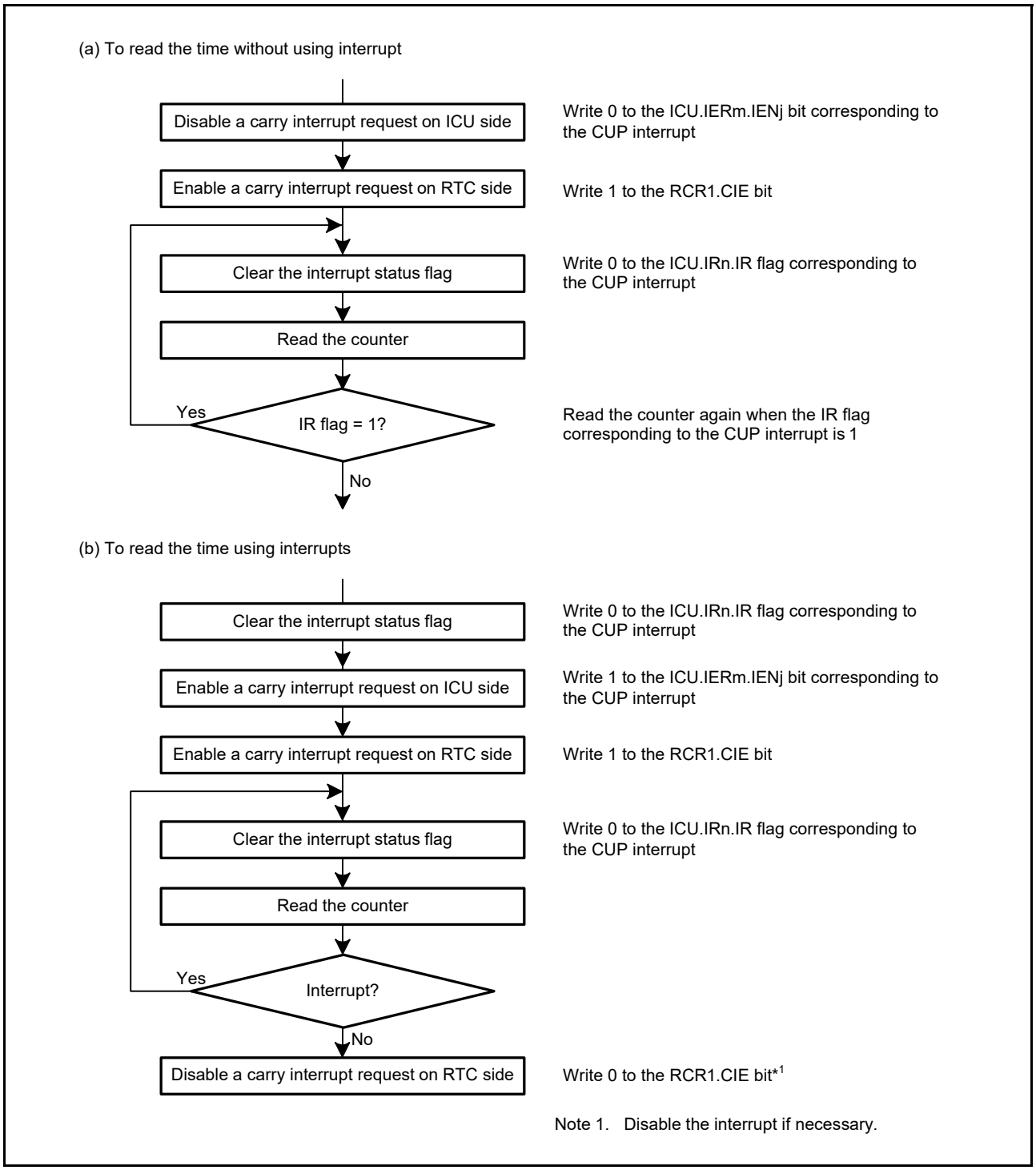


Figure 28.6 Reading Time

If a carry occurs while the 64-Hz counter and time are being read, the correct time will not be obtained, so they must be read again. The procedure for reading the time without using interrupts is shown in (a) in Figure 28.6, and the procedure using carry interrupts in (b). To keep the program simple, method (a) should be used in most cases.

Downloaded from Arrow.com.

28.3.6 Alarm Function

Figure 28.7 shows how to use the alarm function.

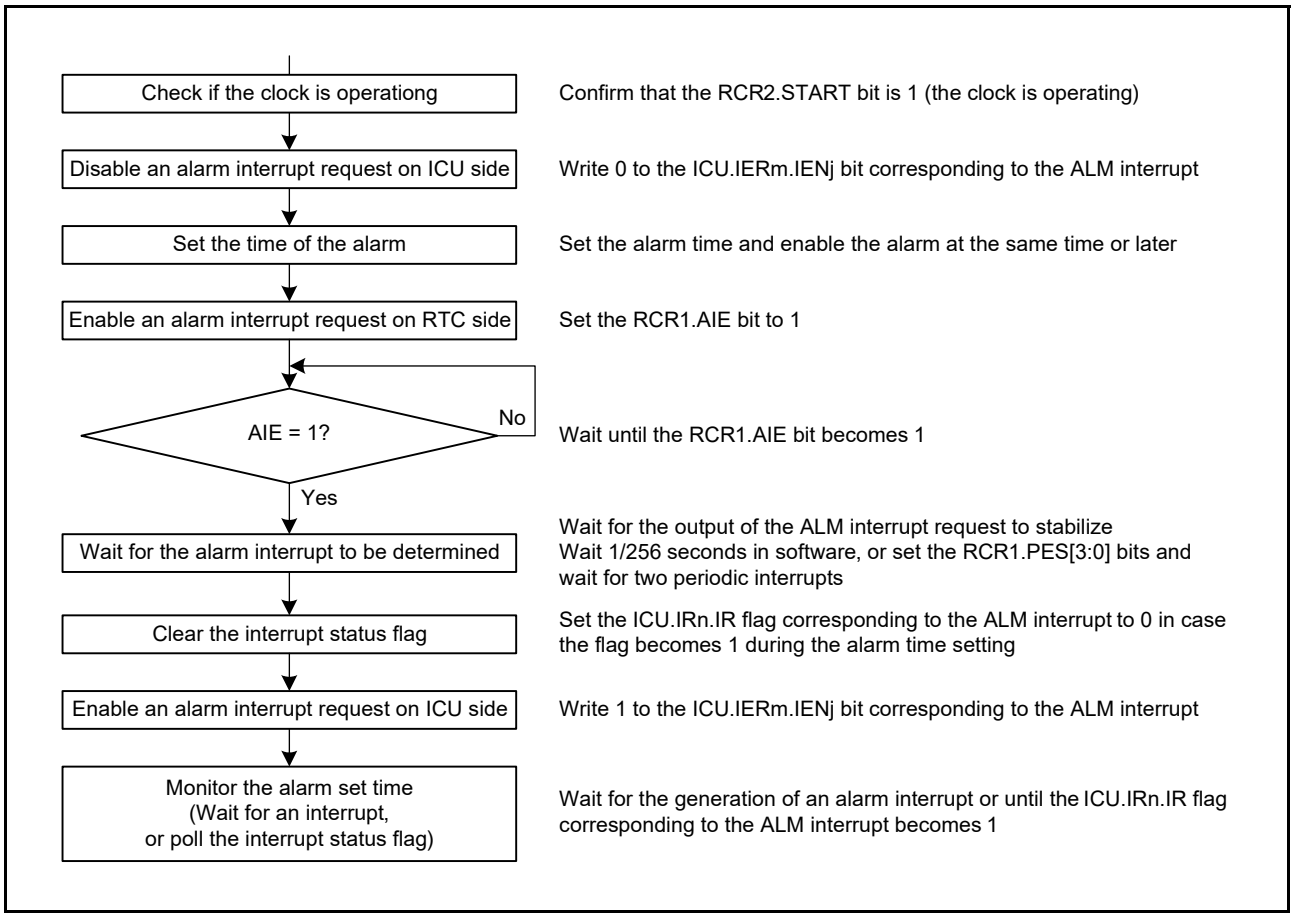


Figure 28.7 Using Alarm Function

In calendar count mode, an alarm can be generated by any one of year, month, date, day-of-week, hour, minute or second, or any combination of those. Write 1 to the ENB bit in the alarm registers involved in the alarm setting, and set the alarm time in the lower bits. Write 0 to the ENB bit in registers not involved in the alarm setting.

In binary count mode, an alarm can be generated in any bit combination of 32 bits. Write 1 to the ENB bit of the alarm enable register corresponding to the target bit of the alarm, and set the alarm time to the alarm register. For bits that are not target of the alarm, write 0 to the ENB bit of the alarm enable register.

When the counter and the alarm time match, the IR flag corresponding to the ALM interrupt is set to 1. Alarm detection can be confirmed by reading this bit, but an interrupt should be used in most cases. If 1 has been set in the interrupt request enable bit corresponding to the ALM interrupt, an alarm interrupt is generated in the event of alarm, enabling the alarm to be detected.

Writing 0 sets the IR flag corresponding to the ALM interrupt to 0.

When the counter and the alarm time match in a low power consumption state, the MCU returns from the low power consumption state.

Downloaded from Arrow.com.

28.3.7 Procedure for Disabling Alarm Interrupt

Figure 28.8 shows the procedure for disabling the enabled alarm interrupt request.

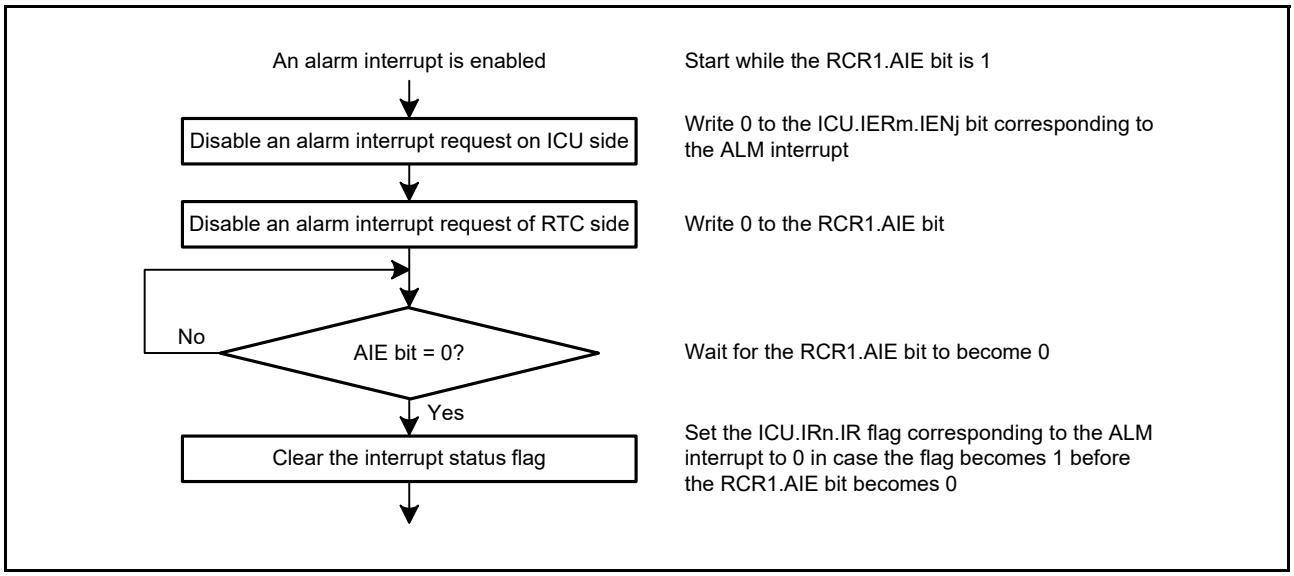


Figure 28.8 Procedure for Disabling Alarm Interrupt Request

28.3.8 Time Error Adjustment Function

The time error adjustment function is used to correct errors (running fast or slow) in the time due to the precision of oscillation by the sub-clock. Since 32,768 cycles of the sub-clock constitute 1 second of operation when the sub-clock is selected, the clock runs fast if the sub-clock frequency is high and slow if the sub-clock frequency is low. This function can be used to correct errors due to the clock running fast or slow.
Two types of time error adjustment functions are provided: automatic adjustment and adjustment by software.
Use the RCR2.AADJE bit to select automatic adjustment or adjustment by software.

28.3.8.1 Automatic Adjustment

Enable automatic adjustment by setting the RCR2.AADJE bit to 1.
Automatic adjustment is the addition or subtraction of the value counted by the prescaler to or from the value in the RADJ register every time the adjustment period selected by the RCR2.AADJE bit elapses.
Examples are shown below.

[Example 1] Sub-clock running at 32.769 kHz
Adjustment procedure:
When the sub-clock is running at 32.769 kHz, 1 second elapses every 32,769 clock cycles. The RTC is meant to run at 32,768 clock cycles, so the clock runs fast by one clock cycle every second. The time on the clock is fast by 60 clock cycles per minute, so adjustment can take the form of setting the clock back by 60 cycles every minute.

Register settings: (when RCR2.CNTMD = 0)

- RCR2.AADJP = 0 (adjustment every minute)
- RADJ.PMADJ[1:0] = 10b (adjustment is performed by the subtraction from the prescaler.)
- RADJ.ADJ[5:0] = 60 (3Ch)

Downloaded from Arrow.com.

[Example 2] Sub-clock running at 32.766 kHz
Adjustment procedure:
When the sub-clock is running at 32.766 kHz, 1 second elapses every 32,766 clock cycles. The RTC is meant to run at 32,768 clock cycles, so the clock runs slow by two clock cycles every second. The time on the clock is slow by 20 clock cycles every 10 seconds, so adjustment can take the form of setting the clock forward by 20 cycles every 10 seconds.

Register settings: (when RCR2.CNTMD = 0)

- RCR2.AADJP = 1 (adjustment every 10 seconds)
- RADJ.PMADJ[1:0] = 01b (adjustment is performed by the addition to the prescaler.)
- RADJ.ADJ[5:0] = 20 (14h)

[Example 3] Sub-clock running at 32.764 kHz
Adjustment procedure:
At 32.764 kHz, 1 second elapses on 32,764 clock cycles. Since the RTC operates for 32,768 clock cycles as 1 second, the clock is delayed for four clock cycles per second. In 8 seconds, the delay is 32 clock cycles, so correction can be made by proceeding the clock for 32 clock cycles every 8 seconds.

Register settings when the RCR2.CNTMD bit is 1

- RCR2.AADJP = 1 (adjustment every 8 seconds)
- RADJ.PMADJ[1:0] = 01b (adjustment is performed by the addition to the prescaler.)
- RADJ.ADJ[5:0] = 32 (20h)

28.3.8.2 Adjustment by Software

Enable adjustment by software by setting the RCR2.AADJE bit to 0.
Adjustment by software is the addition or subtraction of the value counted by the prescaler to or from the value in the RADJ register at the time of execution of an instruction for writing to the RADJ register.
An example is shown below.

[Example 1] Sub-clock running at 32.769 kHz
Adjustment procedure:
When the sub-clock is running at 32.769 kHz, 1 second elapses every 32,769 clock cycles. The RTC is meant to run at 32,768 clock cycles, so the clock runs fast by one clock cycle every second. The time on the clock is fast by one clock cycle per second, so adjustment can take the form of setting the clock back by one cycle every second.

Register settings:

- RADJ.PMADJ[1:0] = 10b (adjustment is performed by the subtraction from the prescaler.)
- RADJ.ADJ[5:0] = 1 (01h)
  This is written to the RADJ register once per 1-second interrupt.

Downloaded from Arrow.com.

28.3.8.3 Procedure for Changing the Mode of Adjustment

When changing the mode of adjustment, change the value of the AADJE bit in RCR2 after setting the RADJ.PMADJ[1:0] bits to 00b (adjustment is not performed).

Changing from adjustment by software to automatic adjustment:

(1) Set the RADJ.PMADJ[1:0] bits to 00b (adjustment is not performed).
(2) Set the RCR2.AADJE bit to 1 (automatic adjustment is enabled).
(3) Use the RCR2.AADJP bit to select the period of adjustment.
(4) In RADJ, set the PMADJ[1:0] bits for addition or subtraction and the ADJ[5:0] bits to the value for use in time error adjustment.

Changing from adjustment by software to automatic adjustment:

(1) Set the RADJ.PMADJ[1:0] bits to 00b (adjustment is not performed).
(2) Set the RCR2.AADJE bit to 0 (adjustment by software is enabled).
(3) Proceed with adjustment by setting the RADJ.PMADJ[1:0] bits for addition or subtraction and the RADJ.ADJ[5:0] bits to the value for use in time error adjustment at the desired time. After that, the time is adjusted every time a value is written to the RADJ register.

28.3.8.4 Procedure for Stopping Adjustment

Stop adjustment by setting the RADJ.PMADJ[1:0] bits to 00b (adjustment is not performed).

Downloaded from Arrow.com.

28.3.9 Time Capture Function

The RTC is capable of storing the month, date, hour, minute and second/binary counters 3 to 0 by detecting an edge of a signal on a time capture event input pin.

A noise filter can also be used on a time capture event input pin. If the noise filter is enabled, the TCST bit is set to 1 when the input level on the pin matches three times.

The noise filter can be switched on or off for each of the time capture event input pins. Operation when the noise filter is off is shown in Figure 28.9 and operation when the noise filter is on is shown in Figure 28.10.

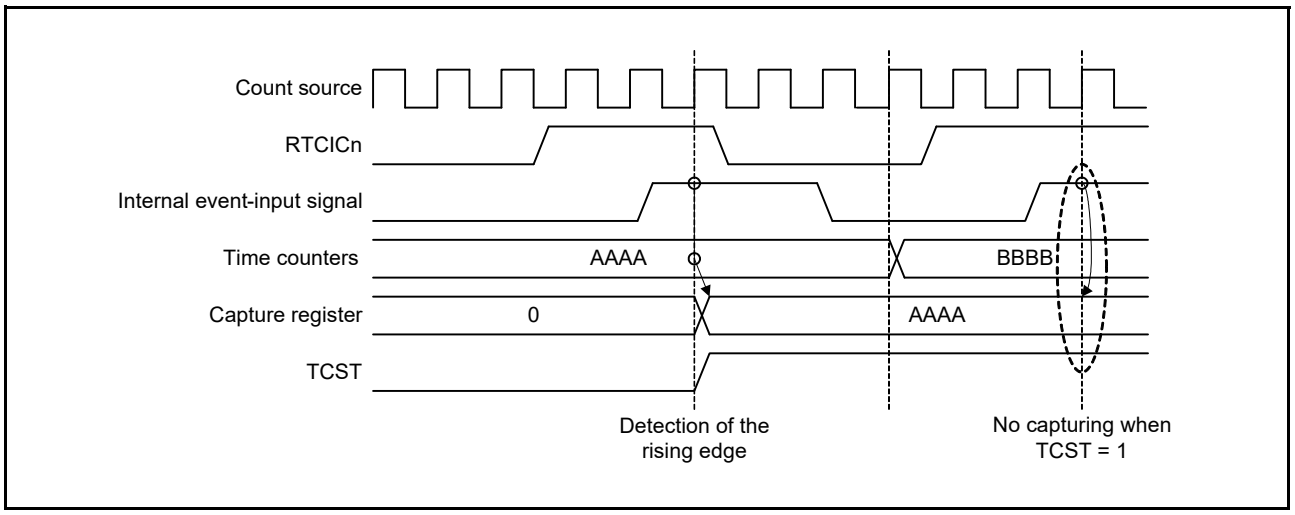


Figure 28.9 Timing of a Time Capture Operation (with the Filter Off) (n = 0, 1)

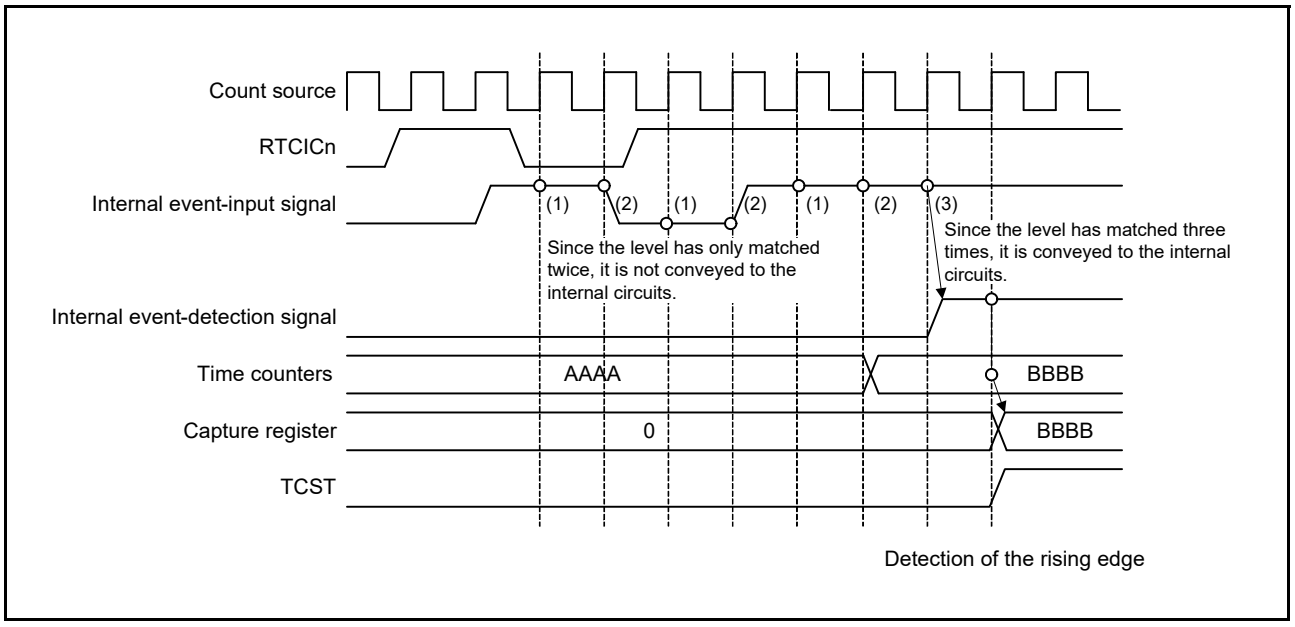


Figure 28.10 Timing of a Time Capture Operation (with the Filter On) (n = 0, 1)

Downloaded from Arrow.com.

28.4 Interrupt Sources

There are three interrupt sources in the realtime clock. Table 28.3 lists interrupt sources for the RTC.

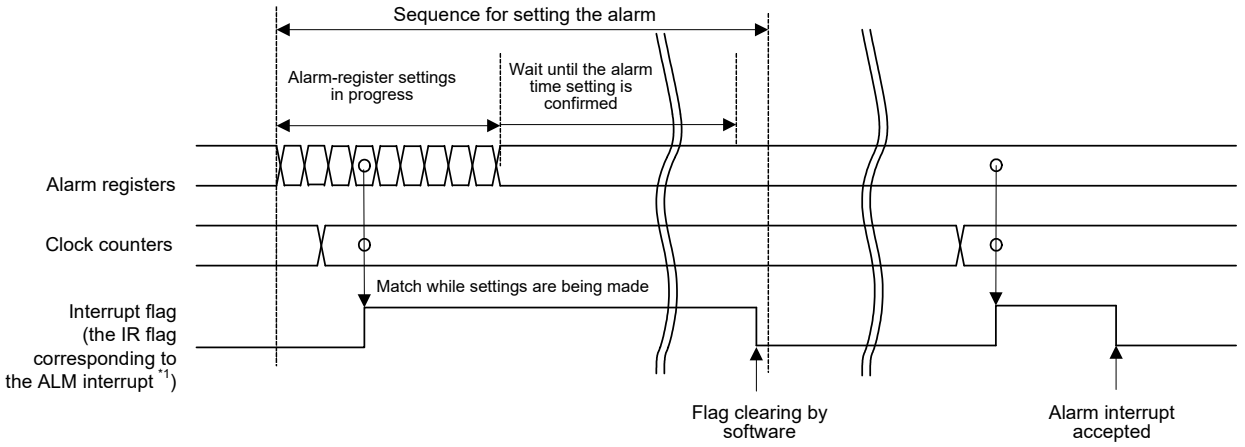
Table 28.3 RTC Interrupt Sources

Name	Interrupt Sources
ALM	Alarm interrupt
PRD	Periodic interrupt
CUP	Carry interrupt

(1) Alarm interrupt (ALM)

This interrupt is generated according to the result of comparison between the alarm registers and realtime clock counters (for details, refer to section 28.3.6, Alarm Function).

Since there is a possibility that the interrupt flag may be set to 1 when the settings of the alarm registers match the clock counters, wait for the alarm time settings to be confirmed and set the IR flag corresponding to the ALM interrupt to 0 again after modifying values of the alarm registers. Once the interrupt flag for the alarm interrupt has been set to 1 and the state has returned to non-matching of the alarm registers and clock counters, the flag will not be set again until there is a further match or the values of the alarm registers are modified again.



Note 1. Refer to the Interrupt Controller section for details on the corresponding interrupt vector number.

Figure 28.11 Timing Chart for the Alarm Interrupt (ALM)

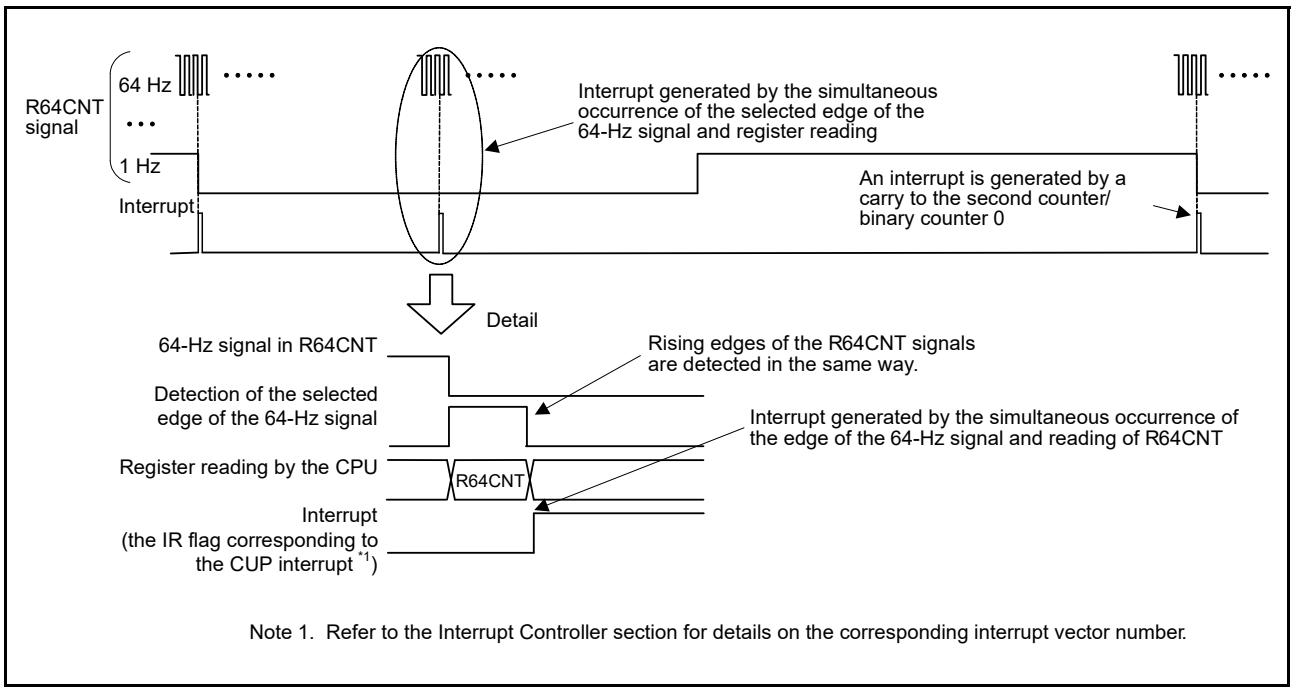
(2) Periodic interrupt (PRD)

This interrupt is generated at intervals of 2 seconds, 1 second, 1/2 second, 1/4 second, 1/8 second, 1/16 second, 1/32 second, 1/64 second, 1/128 second, or 1/256 second. The interrupt interval can be selected through the RCR1.PES[3:0] bits.

Downloaded from Arrow.com.

(3) Carry interrupt (CUP)

This interrupt is generated when a carry to the second counter/binary counter 0 occurred or a carry to the R64CNT counter occurred during read access to the 64-Hz counter.



Note 1. Refer to the Interrupt Controller section for details on the corresponding interrupt vector number.

Figure 28.12 Carry Interrupt (CUP) Timing Chart

Downloaded from Arrow.com.

28.5 Event Link Output

The RTC outputs the following event signals for the event link controller (ELC), and these can be used to initiate operations by other modules selected in advance.

(1) Periodic event output

The periodic event signal is output at the interval selected from among 1/256, 1/128, 1/64, 1/32, 1/16, 1/8, 1/4, 1/2, 1, and 2 seconds by the setting of the RCR1.PES[3:0] bits.

The event generation period immediately after the event generation is selected is not guaranteed.

Note: If event linking from the RTC is to be used, only make the ELC settings after making the RTC settings (initialization, time settings, etc.). Making the RTC settings after the ELC settings can lead to the output of unexpected event signals.

28.5.1 Interrupt Handling and Event Linking

The RTC has a bit to enable or disable periodic interrupts. An interrupt request signal is output for the CPU when an interrupt source is generated while the corresponding enable bit is enabled.

In contrast, an event link output signal is sent to other modules as an event signal via the ELC when an interrupt source is generated, regardless of the setting of the corresponding interrupt enable bit.

Note: Although alarm and periodic interrupts can still be output during software standby, the periodic event signals for the ELC are not output.

Downloaded from Arrow.com.

28.6 Usage Notes

28.6.1 Register Writing during Counting

The following registers should not be written to during counting (while the RCR2.START bit = 1).

RSECCNT/BCNT0, RMINCNT/BCNT1, RHRCNT/BCNT2, RDAYCNT, RWKCNT/BCNT3, RMONCNT, RYRCNT, RCR1.RTCOS, RCR2.RTCOE, RCR2.HR24

The counter must be stopped before writing to any of the above registers.

28.6.2 Use of Periodic Interrupts

The procedure for using periodic interrupts is shown in Figure 28.13.
The generation and period of the periodic interrupt can be changed by the setting of the RCR1.PES[3:0] bits. However, since the prescaler, R64CNT, and RSECCNT/BCNT0 are used to generate interrupts, the interrupt period is not guaranteed immediately after setting of the RCR1.PES[3:0] bits.
Furthermore, stopping/restarting or resetting counter operation, reset by RTC software, and the 30-second adjustment by changing the RCR2 value affects the interrupt period. When the time error adjustment function is used, the interrupt generation period after adjustment is added or subtracted according to the adjustment value.

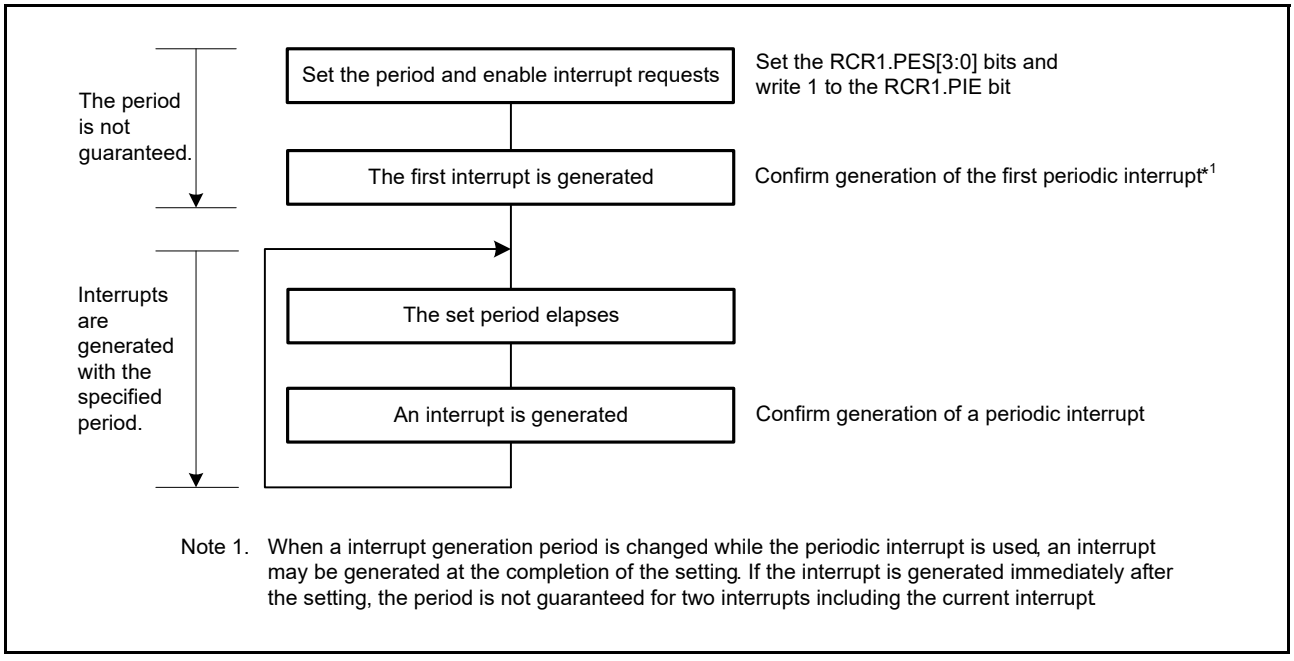


Figure 28.13 Using Periodic Interrupt Function

28.6.3 RTCOUT (1-Hz/64-Hz) Clock Output

Stopping/restarting or resetting counter operation, reset by RTC software, and the 30-second adjustment by changing the RCR2 value affects the period of RTCOUT (1-Hz/64-Hz) output. When the time error adjustment function is used, the period of RTCOUT (1-Hz/64-Hz) output after adjustment is added or subtracted according to the adjustment value.

Downloaded from Arrow.com.

28.6.4 Transitions to Low Power Consumption Modes after Setting Registers

A transition to a low power consumption state (software standby mode, or battery backup) during writing to or updating of an RTC register might destroy the register's value. After setting a register, confirm that the setting is in place before initiating a transition to a low power consumption state.

28.6.5 Notes on Writing to and Reading from Registers

- When reading a counter register such as the second counter after having written to the counter register, follow the procedure in section 28.3.5, Reading 64-Hz Counter and Time.
- The value written to the count registers, alarm registers, year alarm enable register, bits RCR2.AADJE, AADJP, and HR24, or RCR3 register is reflected when four read operations are performed after writing.
- The values written to the RCR1.CIE, RTCOS, and RCR2.RTCOE bits can be read immediately after writing.
- To read the value from the timer counter after return from a reset, software standby mode, or the battery backup state, wait for 1/128 second while the clock is operating (RCR2.START bit = 1).
- After a reset is generated, write to the RTC register when six cycles of the count source have elapsed.

28.6.6 Changing the Count Mode

When changing the count mode (calendar/binary), set the RCR2.START bit to 0, stop counting operation, then start again from the initial setting. For details on initial setting, refer to section 28.3.1, Outline of Initial Settings of Registers after Power On.

Downloaded from Arrow.com.

28.6.7 Initialization Procedure When the Realtime Clock is Not to be Used

Registers in the RTC are not initialized by a reset. Accordingly, depending on the initial state, the generation of an unintentional interrupt request or operation of the counter may lead to increased power consumption.
For products that do not require a realtime clock, initialize the registers by following the initialization procedure shown in Figure 28.14.

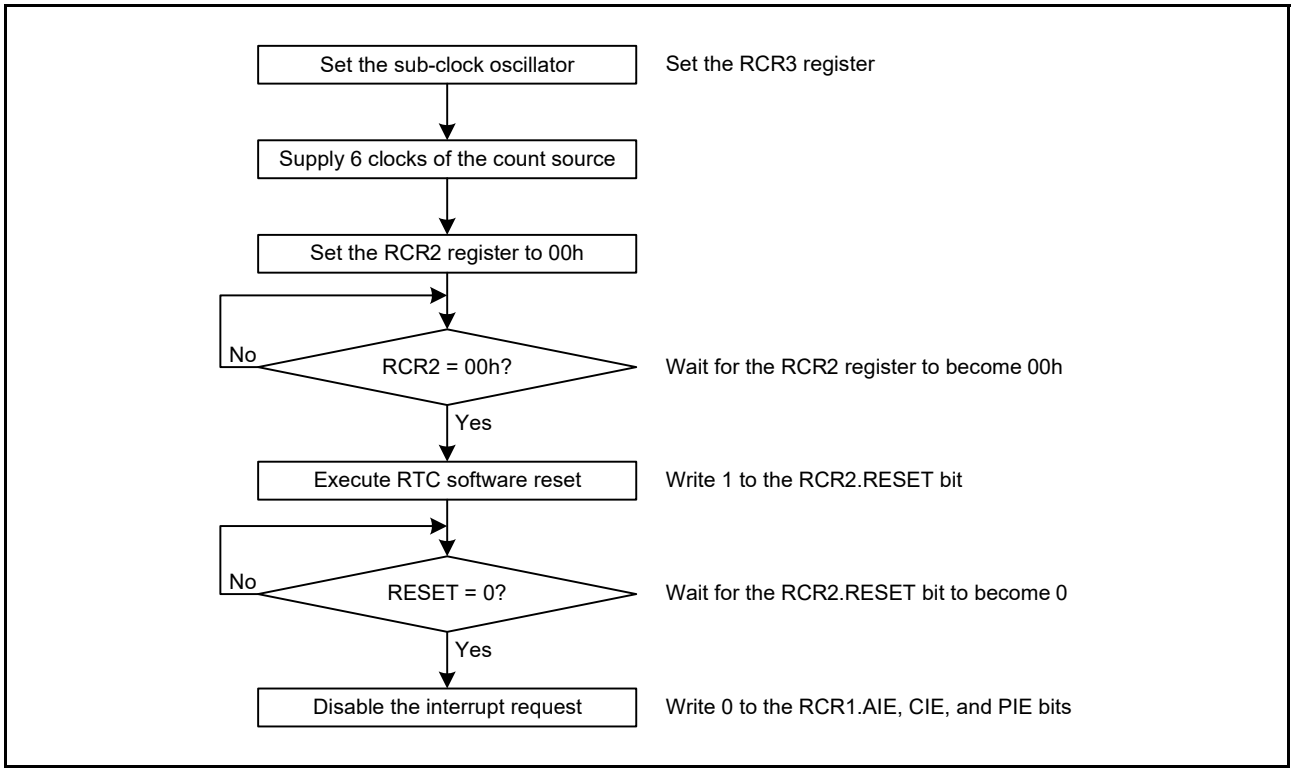


Figure 28.14 Initialization Procedure

Downloaded from Arrow.com.

29. Low-Power Timer (LPT)

29.1 Overview

This MCU integrates a low-power timer (LPT) that consists of a single-channel 16-bit timer. The LPT uses a sub-clock or IWDT-dedicated clock as the clock source, and can continue counting operation even in software standby mode. A compare match signal can be used to return from software standby mode to normal operating mode.
Table 29.1 lists the specifications of the LPT and Figure 29.1 shows a block diagram of the LPT.

Table 29.1 LPT Specifications

Item	Description
Clock source	Sub-clock or IWDT-dedicated clock
Clock division ratio	Divided by 2, 4, 8, 16, or 32
Counting operation	• Count up using the 16-bit up-counter • Counting operation can be continued even in software standby mode
Compare match	Compare match 0 (a compare match signal is generated only in software standby mode)
Event link function (output)	Compare match 0 (a compare match signal is generated only in software standby mode)

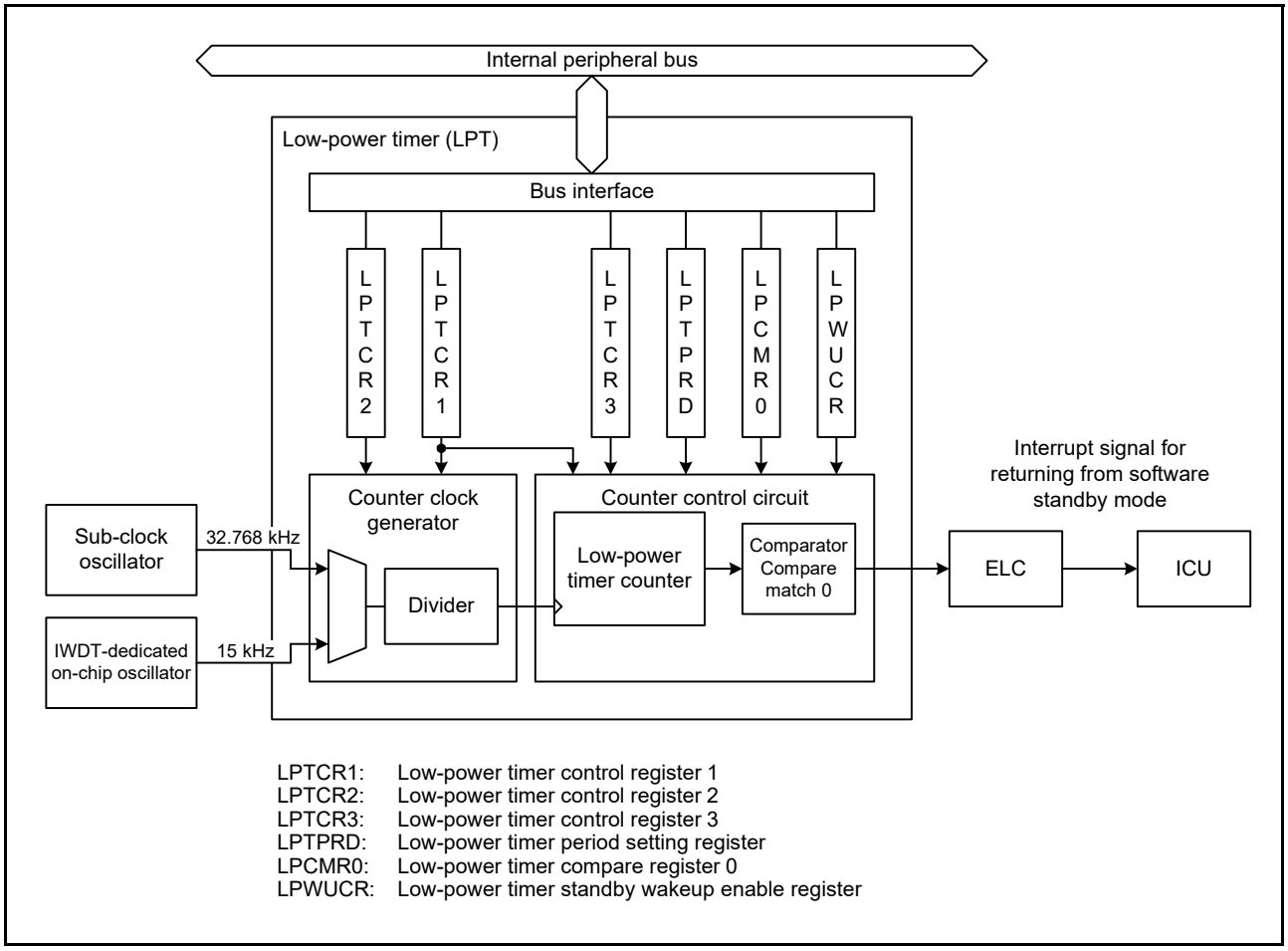


Figure 29.1 LPT Block Diagram

Downloaded from Arrow.com.

29.2 Register Descriptions

29.2.1 Low-Power Timer Control Register 1 (LPTCR1)

Address(es): LPT.LPTCR1 0008 00B0h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	LPCMRE0	—	LPCNTCKSEL	—	LPCNTPSSEL[2:0]		
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	LPCNTPSSEL[2:0]	Clock Division Ratio Select*1	b2 b0 0 0 1: Divided by 2 0 1 0: Divided by 4 0 1 1: Divided by 8 1 0 0: Divided by 16 1 0 1: Divided by 32 Settings other than above are prohibited.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	LPCNTCKSEL	Clock Source Select*1, *2	0: Sub-clock 1: IWDT-dedicated clock (IWDTCLK)*3	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	LPCMRE0	Compare Match 0 Enable*4	0: Compare match 0 is disabled 1: Compare match 0 is enabled	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC2 bit to 1 (write enabled) before rewriting this register.
Note 1. Rewrite these bits while the LPTCR2.LPCNTSTP bit is 1 (supply of clock to the low-power timer is stopped).
Note 2. Satisfy that the frequency of the system clock (ICLK) and peripheral module clock (PCLKB) ≥ 4 × (the frequency of the clock source).
Note 3. A clock generated by the IWDT-dedicated on-chip oscillator (IWDTCLK) is supplied to the low-power timer. When modifying this bit, make sure that the IWDT-dedicated on-chip oscillator is oscillating stably.
When the IWDTCLK is used as the clock source for the low-power timer, set the OFS0.IWDTSLCSTP bit to 0 (counting stop is disabled) in IWDT auto-start mode operation, and set the IWDTCSTPR.SLCSTP bit to 0 (count stop is disabled) in other modes. Without this setting, the IWDT-dedicated on-chip oscillator is stopped in software standby mode.
Note 4. Rewrite this bit while the LPTCR3.LPCNTEN bit is 0 (low-power timer counter stops).

The LPTCR1 register is used to control the low-power timer.

LPCNTCKSEL Bit (Clock Source Select)

This bit is used to select the sub-clock or IWDT-dedicated clock as the clock source for the low-power timer.

LPCMRE0 Bit (Compare Match 0 Enable)

This bit enables or disables low-power timer compare match 0.
When the low-power timer is put into operation and the MCU makes a transition to software standby mode while this bit and the LPWUCR.LPWKUPEN bit are set to 1 (wakeup from software standby mode using low-power timer is enabled), the MCU returns from software standby mode to normal operating mode through the event link controller (ELC) when the value of the low-power timer counter matches the value of the LPCMR0 register.
Settings for the interrupt and ELC are necessary to use a compare match 0 as a trigger source to return from software standby mode.
Refer to section 20, Event Link Controller (ELC) for details on the ELC settings, and refer to section 15, Interrupt Controller (ICUb) for details on the interrupt settings.

Downloaded from Arrow.com.

An interrupt request at compare match 0 is generated only in software standby mode. It is not generated in normal operating mode, sleep mode, and deep sleep mode.

29.2.2 Low-Power Timer Control Register 2 (LPTCR2)

Address(es): LPT.LPTCR2 0008 00B1h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	LPCNTSTP
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	LPCNTSTP	Clock Supply Control	0: Clock is supplied to the low-power timer. 1: Supply of clock to the low-power timer is stopped.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC2 bit to 1 (write enabled) before rewriting this register.

The LPTCR2 register is used to control supply of the clock to be used for the low-power timer.

LPCNTSTP Bit (Clock Supply Control)

This bit is used to supply or stop the clock to be used for the low-power timer. When this bit is set to 0, the clock signal is supplied to the low-power timer counter and divider.

Downloaded from Arrow.com.

29.2.3 Low-Power Timer Control Register 3 (LPTCR3)

Address(es): LPT.LPTCR3 0008 00B2h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	LPCNTRST	LPCNTEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	LPCNTEN	Low-Power Timer Counter Operation Control	0: Low-power timer counter stops 1: Low-power timer counter operates	R/W
b1	LPCNTRST	Low-Power Timer Counter Clear*1, *2	• When writing 0: Has no effect 1: Clears divider and counter • When reading 0: Clearing is completed 1: Clearing is in progress	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC2 bit to 1 (write enabled) before rewriting this register.
Note: Rewrite this register while the LPTCR2.LPCNTSTP bit is 0 (clock is supplied to the low-power timer).
Note 1. Rewrite this bit while the LPTCR3.LPCNTEN bit is 0 (low-power timer counter stops).
Note 2. When clearing the low-power timer counter successively, confirm that the LPCNTRST bit becomes 0, wait for at least one cycle of the clock selected by the LPTCR1.LPCNTCKSEL bit, and then write 1 to the LPCNTRST bit again.

The LPTCR3 register controls operations of and clears the low-power timer counter and divider.

LPCNTEN Bit (Low-Power Timer Counter Operation Control)

This bit is used to operate or stop the low-power timer counter and divider.
When this bit is set to 1 while the LPTCR2.LPCNTSTP bit is 0 (clock is supplied to the low-power timer), the low-power timer counter and divider start operating.
Do not write 1 to the LPCNTRST bit while this bit is 1.

LPCNTRST Bit (Low-Power Timer Counter Clear)

This bit is used to clear the low-power timer counter and divider.
When this bit is set to 1 while the LPTCR2.LPCNTSTP bit is 0 (clock is supplied to the low-power timer), the low-power timer counter and divider are cleared in synchronization with the clock used for the low-power timer. Once clearing is complete, this bit automatically becomes 0.
When 1 is written to this bit, confirm that its value becomes 0 before executing the next processing.

Downloaded from Arrow.com.

29.2.4 Low-Power Timer Period Setting Register (LPTPRD)

Address(es): LPT.LPTPRD 0008 00B4h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	LPCNTPRD[15:0]															
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	LPCNTPRD[15:0]	Low-Power Timer Period Setting	Set the period of the low-power timer. Setting range: 0001h to FFFFh	R/W

Note: Set the PRCR.PRC2 bit to 1 (write enabled) before rewriting this register.
Note: Rewrite this register while the LPTCR3.LPCNTEN bit is 0 (low-power timer counter stops).

The LPTPRD register is used to set the period of the low-power timer.
The period of the low-power timer is proportional to “LPTPRD + 1” and calculated by the following formula:

Period of low-power timer = period of clock source × division ratio × (LPTPRD + 1)

When the value of the low-power timer counter matches the set value, the counter is cleared to 0000h and continues counting.
Do not set this register to 0000h.

Table 29.2, and Table 29.3 list examples of setting the periods of the low-power timer. These examples show values most approximate to the target periods.

Downloaded from Arrow.com.

Table 29.2 Example of Low-Power Timer Period Settings for IWDTCLK

Division ratio	Divided by 2			Divided by 4			Divided by 8		
Target period (ms)	Set Value	Actual period (ms)	Error (%)	Set Value	Actual period (ms	Error (%)	Set Value	Actual period (ms	Error (%)
1	0006h	0.93	–6.67	0003h	1.07	6.67	0001h	1.07	6.67
2	000Dh	1.87	–6.67	0006h	1.87	–6.67	0003h	2.13	6.67
5	0024h	4.93	–1.33	0011h	4.80	–4.00	0008h	4.80	–4.00
10	004Ah	10.00	0.00	0024h	9.87	–1.33	0011h	9.60	–4.00
20	0095h	20.00	0.00	004Ah	20.00	0.00	0024h	19.73	–1.33
50	0176h	50.00	0.00	00BAh	49.87	–0.27	005Ch	49.60	–0.80
100	02EDh	100.00	0.00	0176h	100.00	0.00	00BAh	99.73	–0.27
200	05DBh	200.00	0.00	02EDh	200.00	0.00	0176h	200.00	0.00
500	0EA4h	499.87	–0.03	0751h	499.73	–0.05	03A8h	499.73	–0.05
1000	1D4Ah	999.87	–0.01	0EA4h	999.73	–0.03	0751h	999.47	–0.05
2000	3A96h	1999.87	–0.01	1D4Ah	1999.73	–0.01	0EA4h	1999.47	–0.03
5000	927Bh	5000.00	0.00	493Dh	5000.00	0.00	249Eh	5000.00	0.00
10000	—	—	—	—	—	—	493Dh	10000.00	0.00
20000	—	—	—	—	—	—	927Bh	20000.00	0.00
50000	—	—	—	—	—	—	—	—	—

Division ratio	Divided by 16			Divided by 32		
Target period (ms)	Set Value	Actual period (ms	Error (%)	Set Value	Actual period (ms	Error (%)
1	—	—	—	—	—	—
2	0001h	2.13	6.67	—	—	—
5	0004h	5.33	6.67	0001h	4.27	–14.67
10	0008h	9.60	–4.00	0004h	10.67	6.67
20	0011h	19.20	–4.00	0008h	19.20	–4.00
50	002Dh	49.07	–1.87	0016h	49.07	–1.87
100	005Ch	99.20	–0.80	002Dh	98.13	–1.87
200	00BAh	199.47	–0.27	005Ch	198.40	–0.80
500	01D3h	499.20	–0.16	00E9h	499.20	–0.16
1000	03A8h	999.47	–0.05	01D3h	998.40	–0.16
2000	0751h	1998.93	–0.05	03A8h	1998.93	–0.05
5000	124Eh	4999.47	–0.01	0926h	4998.40	–0.03
10000	249Eh	10000.00	0.00	124Eh	9998.93	–0.01
20000	493Dh	20000.00	0.00	249Eh	20000.00	0.00
50000	B71Ah	50000.00	0.00	5B8Ch	49998.93	0.00

Downloaded from Arrow.com.

Table 29.3 Example of Low-Power Timer Period Settings for Sub-Clock

Division ratio	Divided by 2			Divided by 4			Divided by 8		
Target period (ms)	Set Value	Actual period (ms)	Error (%)	Set Value	Actual period (ms	Error (%)	Set Value	Actual period (ms	Error (%)
1	000Fh	0.98	–2.34	0007h	0.98	–2.34	0003h	0.98	–2.34
2	001Fh	1.95	–2.34	000Fh	1.95	–2.34	0007h	1.95	–2.34
5	0050h	4.94	–1.12	0027h	4.88	–2.34	0013h	4.88	–2.34
10	00A2h	9.95	–0.51	0050h	9.89	–1.12	0027h	9.77	–2.34
20	0146h	19.96	–0.21	00A2h	19.90	–0.51	0050h	19.78	–1.12
50	0332h	49.99	–0.02	0198h	49.93	–0.15	00CBh	49.80	–0.39
100	0665h	99.98	–0.02	0332h	99.98	–0.02	0198h	99.85	–0.15
200	0CCBh	199.95	–0.02	0665h	199.95	–0.02	0332h	199.95	–0.02
500	1FFFh	500.00	0.00	0FFFh	500.00	0.00	07FFh	500.00	0.00
1000	3FFFh	1000.00	0.00	1FFFh	1000.00	0.00	0FFFh	1000.00	0.00
2000	7FFFh	2000.00	0.00	3FFFh	2000.00	0.00	1FFFh	2000.00	0.00
5000	—	—	—	9FFFh	5000.00	0.00	4FFFh	5000.00	0.00
10000	—	—	—	—	—	—	9FFFh	10000.00	0.00
20000	—	—	—	—	—	—	—	—	—
50000	—	—	—	—	—	—	—	—	—

Division ratio	Divided by 16			Divided by 32		
Target period (ms)	Set Value	Actual period (ms	Error (%)	Set Value	Actual period (ms	Error (%)
1	0001h	0.98	–2.34	—	—	—
2	0003h	1.95	–2.34	0001h	1.95	–2.34
5	0009h	4.88	–2.34	0004h	4.88	–2.34
10	0013h	9.77	–2.34	0009h	9.77	–2.34
20	0027h	19.53	–2.34	0013h	19.53	–2.34
50	0065h	49.80	–0.39	0032h	49.80	–0.39
100	00CBh	99.61	–0.39	0065h	99.61	–0.39
200	0198h	199.71	–0.15	00CBh	199.22	–0.39
500	03FFh	500.00	0.00	01FFh	500.00	0.00
1000	07FFh	1000.00	0.00	03FFh	1000.00	0.00
2000	0FFFh	2000.00	0.00	07FFh	2000.00	0.00
5000	27FFh	5000.00	0.00	13FFh	5000.00	0.00
10000	4FFFh	10000.00	0.00	27FFh	10000.00	0.00
20000	9FFFh	20000.00	0.00	4FFFh	20000.00	0.00
50000	—	—	—	C7FFh	50000.00	0.00

Downloaded from Arrow.com.

29.2.5 Low-Power Timer Compare Register 0 (LPCMR0)

Address(es): LPT.LPCMR0 0008 00B8h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	LPCMR0[15:0]															
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	LPCMR0[15:0]	Low-Power Timer Compare 0	Set the value of compare match 0 for comparison with the low-power timer counter.	R/W

Note: Set the PRCR.PRC2 bit to 1 (write enabled) before rewriting this register.
Note: Rewrite this register while the LPTCR3.LPCNTEN bit is 0 (low-power timer counter stops).

The LPCMR0 register is used to set the value of compare match 0 for comparison with the low-power timer counter. Set the LPCMR0 register to a value smaller than or equal to the value of the LPTPRD register.

Downloaded from Arrow.com.

29.2.6 Low-Power Timer Standby Wakeup Enable Register (LPWUCR)

Address(es): LPT.LPWUCR 0008 00BCh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	LPWKUPEN	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b14 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	LPWKUPEN	Low-Power Timer Standby Wakeup Enable*1	0: Wakeup from software standby mode using low-power timer is disabled 1: Wakeup from software standby mode using low-power timer is enabled	R/W

Note: Set the PRCR.PRC2 bit to 1 (write enabled) before rewriting this register.
Note 1. Rewrite this bit while the LPTCR3.LPCNTEN bit is 0 (low-power timer counter stops).

The LPWUCR register is used to enable the function that allows to return from software standby mode to normal operating mode when compare match 0 occurs in the low-power timer.

LPWKUPEN Bit (Low-Power Timer Standby Wakeup Enable)

This bit enables or disables the function that allows to return from software standby mode to normal operating mode when compare match 0 occurs in the low-power timer.

Downloaded from Arrow.com.

29.3 Operation

29.3.1 Periodic Counting Operation

The low-power timer is a 16-bit up-counter that operates regardless of the MCU operating mode*1.
When the LPTCR3.LPCNTEN bit is set to 1 (low-power timer counter operates) after setting the LPTCR1.LPCNTPSSEL[2:0] bits to select the division ratio, the LPTCR1.LPCNTCKSEL bit to select the clock source, and the LPTCR2.LPCNTSTP bit to 0 (clock is supplied to the low-power timer), the low-power timer counter starts counting with the selected clock.
When the value of the low-power timer counter matches the value of the LPTPRD register, the counter restarts counting from 0000h.
When the value of the low-power timer counter matches the value of the LPCMR0 register in software standby mode while the LPTCR1.LPCMRE0 bit is set to 1 (Compare match 0 is enabled) and the LPWUCR.LPWKUPEN bit is set to 1 (wakeup from software standby mode using low-power timer is enabled), the MCU returns from software standby mode to normal operating mode by the function of the event link controller (ELC).
Figure 29.2 shows operation of the low-power timer and Figure 29.3 shows an example of procedure for the initial settings.

Note 1. When the LPTCR1.LPCNTCKSEL bit is set to 1 (IWDT-dedicated clock), the counter stops because the IWDTCLK stops in the low-power consumption mode under the following settings:
The OFS0.IWDTSLCSTP bit is set to 1 (counting stop is enabled) while IWDT is activated in auto-start mode, or the IWDTCSTPR.SLCSTP bit is set to 1 (counting stop is enabled) while IWDT is activated in register start mode.

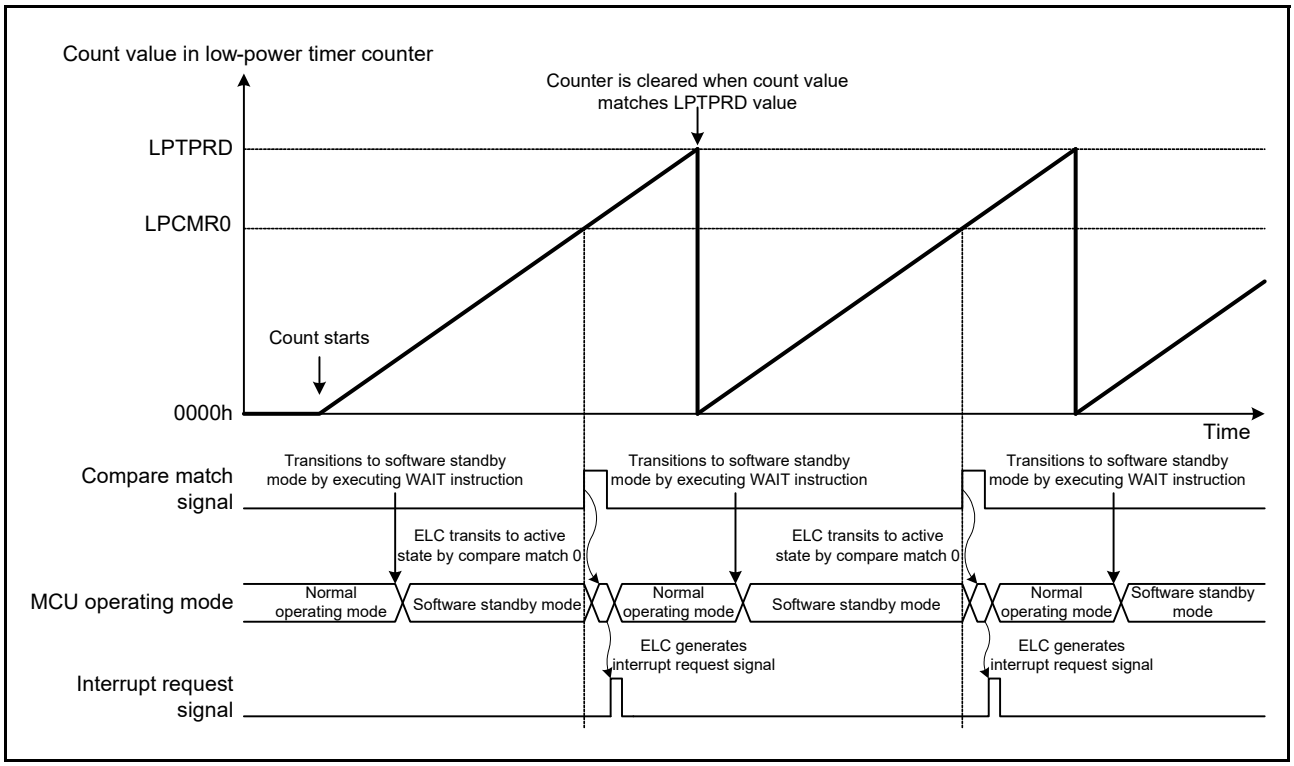


Figure 29.2 Operation of Low-Power Timer

Downloaded from Arrow.com.

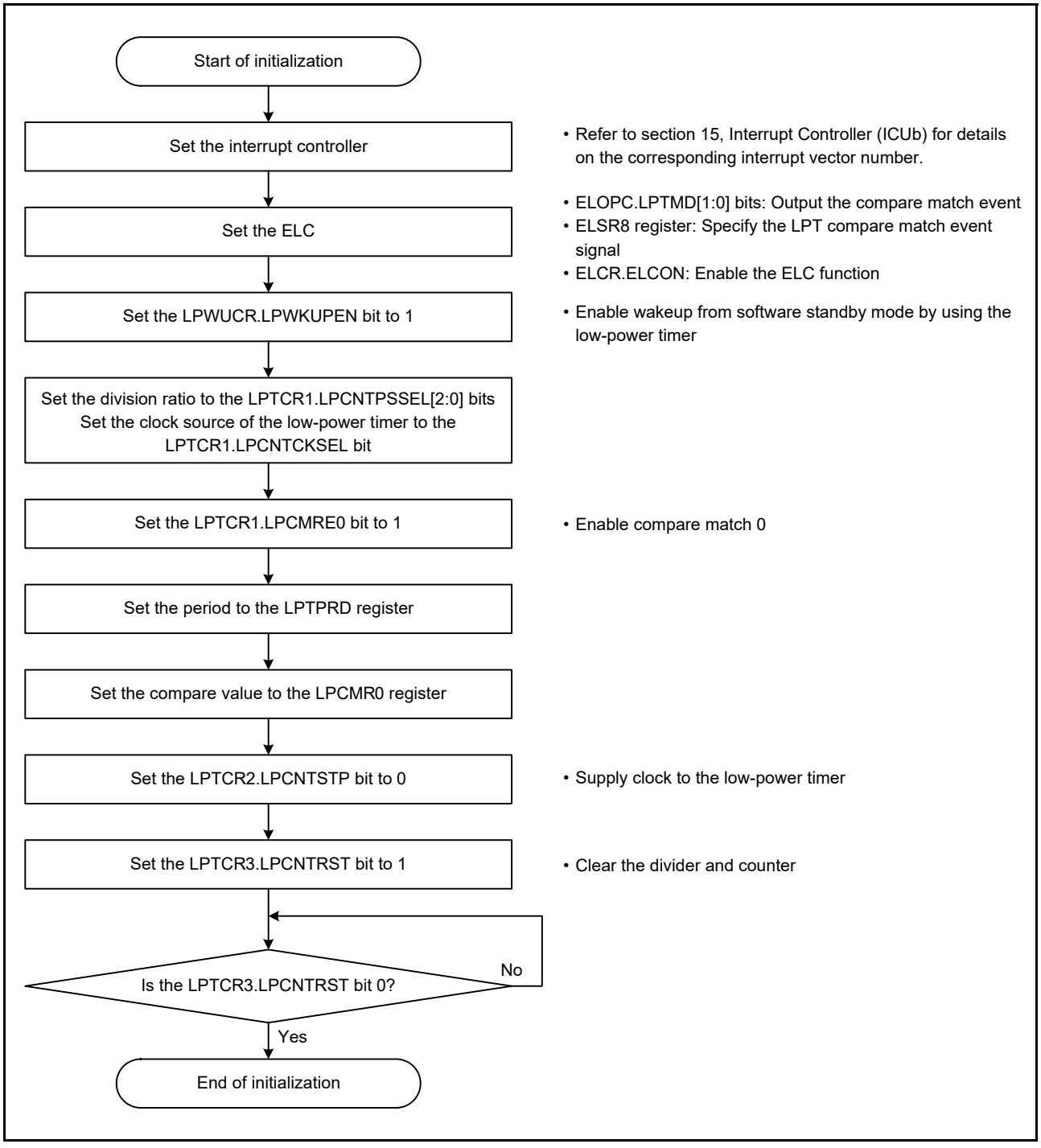


Figure 29.3 Example of Initial Settings

Downloaded from Arrow.com.

29.3.2 Count Timing of Low-Power Timer Counter

The LPTCR1.LPCNTPSSEL[2:0] bits are used to select the counter clock to be input to the low-power timer counter from among five clocks derived from dividing the clock source selected by the LPTCR1.LPCNTCKSEL bit by 2 to 32, respectively.
Figure 29.4 shows the count timing of the low-power timer counter in this case.

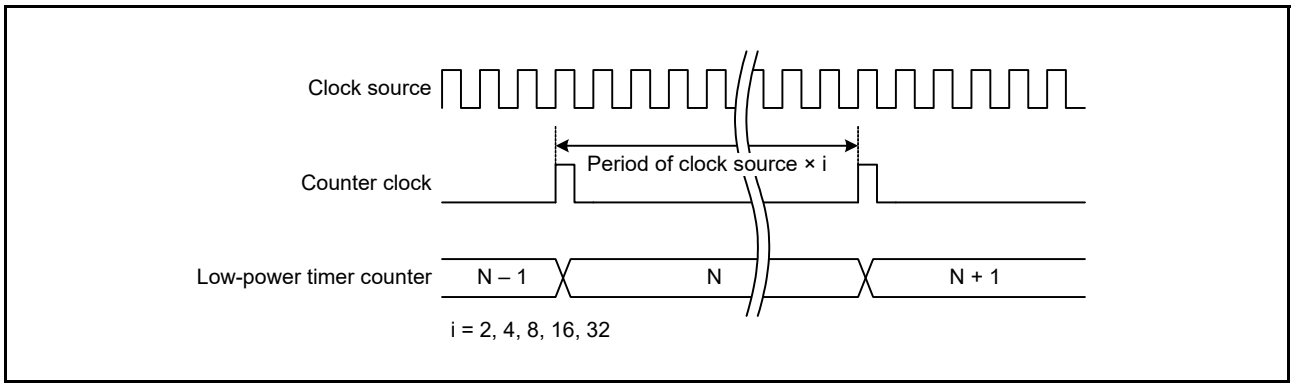


Figure 29.4 Count Timing of Low-Power Timer Counter

29.3.3 Clearing Timing of Low-Power Timer Counter

Writing 1 to the LPTCR3.LPCNTRST bit*1 clears the low-power timer counter.
This bit automatically becomes 0 when the clearing of the counter is completed.
Figure 29.5 shows the clearing timing of the low-power timer counter in this case.

Note 1. Write to the LPTCR3.LPCNTRST bit while the LPTCR3.LPCNTEN bit is 0 (low-power timer counter stops).

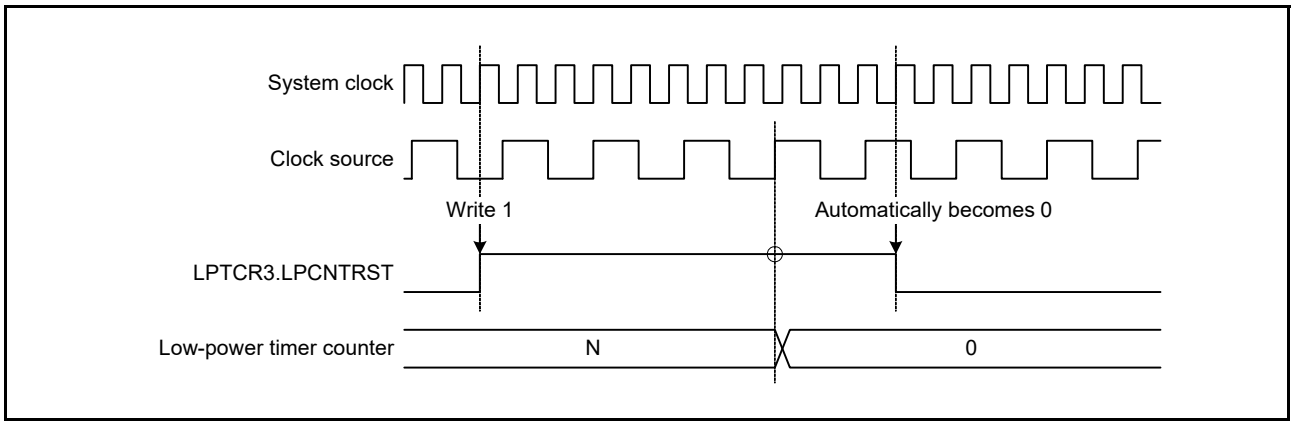


Figure 29.5 Clearing Timing of Low-Power Timer Counter

Downloaded from Arrow.com.

29.4 Wakeup from Software Standby Mode by an Interrupt through the Event Link Controller (ELC)

The low-power timer generates an event signal upon compare match 0 to the event link controller (ELC) only in software standby mode.
Setting the ELOPC.LPTMD[1:0] bits to 00b (output the compare match 0 event to ICU as an interrupt request) and the ELSR8 register to 32h (LPT compare match 0) leads to the generation of an interrupt from the event signal, and the MCU returns from software standby mode to normal operating mode.

29.5 Usage Notes

29.5.1 Notes on Transition to Software Standby Mode

When the MCU is to re-enter to software standby mode after returning from software standby mode to normal operating mode, wait for at least one cycle of the clock selected by the LPTCR1.LPCNTCKSEL bit before executing the WAIT instruction.

Downloaded from Arrow.com.

30. Watchdog Timer (WDTA)

The watchdog timer (WDT) is a 14-bit down-counter. It can be used to reset this MCU when the counter underflows because its value cannot be refreshed due to the system being out of control.
In addition, a non-maskable interrupt can be generated by an underflow.
The refresh-permitted period can be set to refresh the counter and used as the condition to detect when the system runs out of control.
In this section, “PCLK” is used to refer to PCLKB.

30.1 Overview

Table 30.1 lists the specifications of the WDT and Figure 30.1 shows a block diagram of the WDT.

Table 30.1 WDT Specifications

Item	Specifications
Count source	Peripheral module clock (PCLK)
Clock division ratio	Divide by 4, 64, 128, 512, 2048, or 8192
Counter operation	Counting down using a 14-bit down-counter
Conditions for starting the counter	• Auto-start mode: Counting automatically starts after a reset is released • Register start mode: Counting is started by refresh operation (writing 00h and then FFh to the WDTRR register)
Conditions for stopping the counter	• Reset • In low power consumption states • A counter underflows or a refresh error occurs (only in register start mode)
Window function	Window start and end positions can be specified (refresh-permitted and refresh-prohibited periods)
Watchdog timer Reset sources	• Down-counter underflows • Refreshing outside the refresh-permitted period (refresh error)
Non-maskable interrupt sources	• Down-counter underflows • Refreshing outside the refresh-permitted period (refresh error)
Reading the counter value	The down-counter value can be read by the WDTSR register.

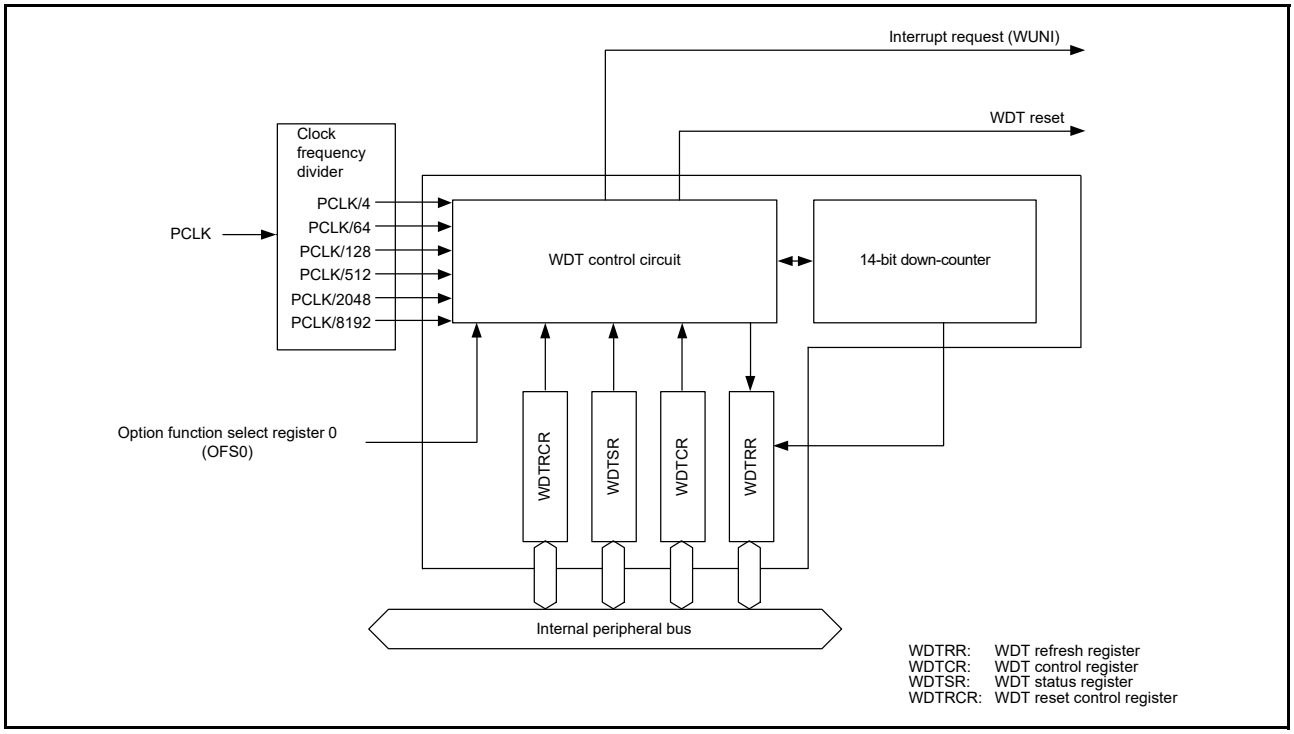


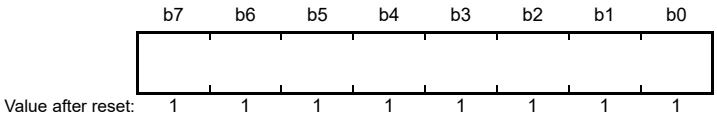
Figure 30.1 WDT Block Diagram

Downloaded from Arrow.com.

30.2 Register Descriptions

30.2.1 WDT Refresh Register (WDTRR)

Address(es): 0008 8020h



Bit	Description	R/W
b7 to b0	The down-counter is refreshed by writing 00h and then writing FFh to this register	R/W

WDTRR refreshes the down-counter of the WDT.
The down-counter of the WDT is refreshed by writing 00h and then writing FFh to WDTRR (refresh operation) within the refresh-permitted period.
After the down-counter has been refreshed, it starts counting down from the value selected by setting the WDTTOPS[1:0] bits in option function select register 0 (OFS0) in auto-start mode. In register start mode, counting down starts from the value selected by setting the WDTCR.TOPS[1:0] bits.
When 00h is written, the read value is 00h, when a value other than 00h is written, the read value is FFh.
For details of the refresh operation, refer to section 30.3.3, Refresh Operation.

Downloaded from Arrow.com.

30.2.2 WDT Control Register (WDTCR)

Address(es): 0008 8022h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	RPSS[1:0]		—	—	RPES[1:0]		CKS[3:0]				—	—	TOPS[1:0]	
Value after reset:	0	0	1	1	0	0	1	1	1	1	1	1	0	0	1	1

Bit	Symbol	Bit Name	Description	R/W
b1, b0	TOPS[1:0]	Timeout Period Selection	b1 b0 0 0: 1024 cycles (03FFh) 0 1: 4096 cycles (0FFFh) 1 0: 8192 cycles (1FFFh) 1 1: 16384 cycles (3FFFh)	R/W
b3, b2	—	Reserved	These bits are read as 0 and cannot be modified.	R
b7 to b4	CKS[3:0]	Clock Division Ratio Selection	b7 b4 0 0 0 1: Divide-by-4 0 1 0 0: Divide-by-64 1 1 1 1: Divide-by-128 0 1 1 0: Divide-by-512 0 1 1 1: Divide-by-2048 1 0 0 0: Divide-by-8192 Setting other than above are prohibited.	R/W
b9, b8	RPES[1:0]	Window End Position Selection	b9 b8 0 0: 75% 0 1: 50% 1 0: 25% 1 1: 0% (window end position is not specified)	R/W
b11, b10	—	Reserved	These bits are read as 0 and cannot be modified.	R
b13, b12	RPSS[1:0]	Window Start Position Selection	b13 b12 0 0: 25% 0 1: 50% 1 0: 75% 1 1: 100% (window start position is not specified)	R/W
b15, b14	—	Reserved	These bits are read as 0 and cannot be modified.	R

There are some restrictions on writing to the WDTCR register. For details, refer to section 30.3.2, Control over Writing to the WDTCR and WDTRCR Registers.

In auto-start mode, the settings in the WDTCR register are disabled, and the settings in option function select register 0 (OFS0) are enabled. The bit setting made to the WDTCR register can also be made in OFS0 register. For details, refer to section 30.3.7, Correspondence between Option Function Select Register 0 (OFS0) and WDT Registers.

TOPS[1:0] Bits (Timeout Period Selection)

These bits select the timeout period (period until the down-counter underflows) from among 1024, 4096, 8192, and 16384 cycles, taking the divided clock specified by the CKS[3:0] bits as one cycle.

After the down-counter is refreshed, the combination of the CKS[3:0] and TOPS[1:0] bits determines the time (number of PCLK cycles) until the counter underflows.

Relations between the CKS[3:0] and TOPS[1:0] bit settings, the timeout period, and the number of PCLK cycles are listed in Table 30.2.

Downloaded from Arrow.com.

Table 30.2 Timeout Period Settings

CKS[3:0] Bits				TOPS[1:0] Bits		Clock Division Ratio	Timeout Period (Number of Cycles)	Cycles of PCLK Clock
b7	b6	b5	b4	b1	b0			
0	0	0	1	0	0	Divide-by-4	1024	4096
				0	1		4096	16384
				1	0		8192	32768
				1	1		16384	65536
0	1	0	0	0	0	Divide-by-64	1024	65536
				0	1		4096	262144
				1	0		8192	524288
				1	1		16384	1048576
1	1	1	1	0	0	Divide-by-128	1024	131072
				0	1		4096	524288
				1	0		8192	1048576
				1	1		16384	2097152
0	1	1	0	0	0	Divide-by-512	1024	524288
				0	1		4096	2097152
				1	0		8192	4194304
				1	1		16384	8388608
0	1	1	1	0	0	Divide-by-2048	1024	2097152
				0	1		4096	8388608
				1	0		8192	16777216
				1	1		16384	33554432
1	0	0	0	0	0	Divide-by-8192	1024	8388608
				0	1		4096	33554432
				1	0		8192	67108864
				1	1		16384	134217728

CKS[3:0] Bits (Clock Division Ratio Selection)

These bits specify the division ration of the clock used for the down-counter. The division ration can be selected from among the peripheral module clock (PCLK) divided by 4, 64, 128, 512, 2048, and 8192. Combined with the TOPS[1:0] bit setting, a count period between 4,096 and 134,217,728 cycles of the PCLK clock can be selected for the WDT.

RPES[1:0] Bits (Window End Position Selection)

These bits specify the window end position that indicates the refresh-permitted period. 75%, 50%, 25%, or 0% of the timeout period can be selected for the window end position. The selected window end position should be a value smaller than the window start position (window start position > window end position). If the window end position is greater than the window start position, only the window start position setting is enabled.

RPSS[1:0] Bits (Window Start Position Selection)

These bits specify the window start position that indicates the refresh-permitted period. 25%, 50%, 75%, or 100% of the timeout period can be selected for the window end position. The window start position should be set to a value greater the window end position. If the window start position is set to a value smaller than or equal to the window end position, the window end position is set to 0%.

Downloaded from Arrow.com.

Table 30.3 lists the counter values for the window start and end positions and Figure 30.2 shows the refresh-permitted period set by the RPSS[1:0], RPES[1:0], and TOPS[1:0] bits.

Table 30.3 Relationship between Timeout Period and Window Start and End Counter Values

TOPS[1:0] Bits		Timeout Period		Window Start and End Counter Value			
		Cycles	Counter Value	100%	75%	50%	25%
0	0	1024	03FFh	03FFh	02FFh	01FFh	00FFh
0	1	4096	0FFFh	0FFFh	0BFFh	07FFh	03FFh
1	0	8192	1FFFh	1FFFh	17FFh	0FFFh	07FFh
1	1	16384	3FFFh	3FFFh	2FFFh	1FFFh	0FFFh

RPSS[1:0] Bits		RPES[1:0] Bits		Window	
b13	b12	b9	b8	Start (%)	End (%)
1	1	1	1	100	0
		1	0		25
		0	1		50
		0	0		75
1	0	1	1	75	0
		1	0		25
		0	1		50
		0	0		75
0	1	1	1	50	0
		1	0		25
		0	1		50
		0	0		75
0	0	1	1	25	0
		1	0		25
		0	1		50
		0	0		75

Note: If window end setting ≥ window start setting, the window end setting is set to 0%.

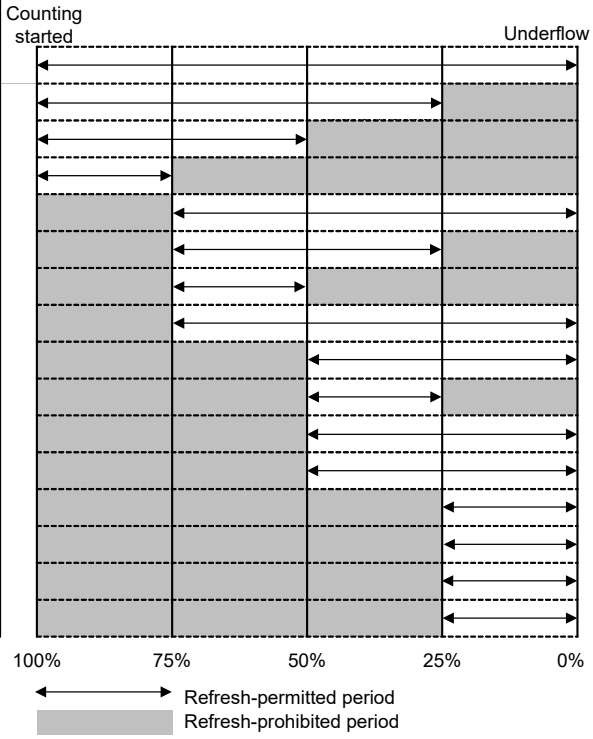


Figure 30.2 RPSS[1:0] and RPES[1:0] Bit Settings and the Refresh-Permitted Period

Downloaded from Arrow.com.

30.2.3 WDT Status Register (WDTSR)

Address(es): 0008 8024h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	REFEF	UNDFF	CNTVAL[13:0]													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b13 to b0	CNTVAL[13:0]	Down-Counter Value	Value counted by the down-counter	R
b14	UNDFF	Underflow Flag	0: No underflow occurred 1: Underflow occurred	R(/W) *1
b15	REFEF	Refresh Error Flag	0: No refresh error occurred 1: Refresh error occurred	R(/W) *1

Note 1. Only 0 can be written to clear the flag.

CNTVAL[13:0] Bits (Down-Counter Value)

Read these bits to confirm the value of the down-counter, but note that the read value may differ from the actual count by a value of one count.

UNDFF Flag (Underflow Flag)

Read this flag to confirm whether or not an underflow has occurred in the down-counter.
The value 1 indicates that the down-counter has underflowed. The value 0 indicates that the down-counter has not underflowed.
Write 0 to the UNDFF flag to set the value to 0. Writing 1 has no effect.

REFEF Flag (Refresh Error Flag)

Read this flag to confirm whether or not a refresh error (performing a refresh operation during a refresh-prohibited period) has occurred.
The value 1 indicates that a refresh error has occurred. The value 0 indicates that no refresh error has occurred.
Write 0 to the REFEF flag to set the value to 0. Writing 1 has no effect.

Downloaded from Arrow.com.

30.2.4 WDT Reset Control Register (WDTRCR)

Address(es): 0008 8026h

	b7	b6	b5	b4	b3	b2	b1	b0
	RSTIRQS	—	—	—	—	—	—	—
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	—	Reserved	These bits are read as 0 and cannot be modified.	R
b7	RSTIRQS	Reset Interrupt Request Selection	0: Non-maskable interrupt request output is enabled 1: Reset output is enabled	R/W

There are some restrictions on writing to the WDTRCR register. For details, refer to section 30.3.2, Control over Writing to the WDTCR and WDTRCR Registers.
In auto-start mode, the WDTRCR register settings are disabled, and the settings in option function select register 0 (OFS0) are enabled. The bit setting made to the WDTCR register can also be made in the OFS0 register. For details, refer to section 30.3.7, Correspondence between Option Function Select Register 0 (OFS0) and WDT Registers.

30.2.5 Option Function Select Register 0 (OFS0)

For details on the OFS0 register, refer to section 30.3.7, Correspondence between Option Function Select Register 0 (OFS0) and WDT Registers.

Downloaded from Arrow.com.

30.3 Operation

30.3.1 Count Operation in Each Start Mode

The WDT has two start modes: auto-start mode, in which counting automatically starts after a reset is released, and register start mode, in which counting is started by refresh operation (writing to the register).
In auto-start mode, counting automatically starts after a reset is released in accordance with the settings in option function select register 0 (OFS0) in the ROM.
In register start mode, counting is started by refresh operation (writing to the register) after the respective registers are set after a reset is released.
Select auto-start mode or register start mode by setting the OFS0.WDTSTRT bit.
When the auto-start mode is selected, the settings in the WDTCR and WDTRCR registers are disabled, and the settings in the OFS0 register are enabled.
On the other hand, when the register start mode is selected, the setting of the OFS0 register is disabled, and the settings of the WDTCR and WDTRCR registers are enabled.

30.3.1.1 Register Start Mode

When the OFS0.WDTSTRT bit is 1, register start mode is selected, and the WDTCR and WDTRCR registers are enabled.
After a reset is released, set the clock division ratio, window start and end positions, and timeout period in the WDTCR register, and the reset output or interrupt request output in the WDTRCR register. Then, refresh the down-counter to start counting down from the value set by the WDTCR.TOPS[1:0] bits.
Thereafter, as long as the counter is refreshed in the refresh-permitted period, the value in the counter is re-set each time the counter is refreshed and counting down continues. The WDT does not output the reset signal as long as this continues. However, if the down-counter underflows because the down-counter cannot be refreshed due to a program runaway, or if a refresh error occurs because the counter was refreshed outside the refresh-permitted period, the WDT outputs a reset signal or a non-maskable interrupt request (WUNI). Reset output or interrupt request output can be selected by setting the WDTRCR.RSTIRQS bit.

Figure 30.3 shows an example of operation under the following conditions.

- Register start mode (OFS0.WDTSTRT = 1)
- Reset output is enabled (WDTRCR.RSTIRQS = 1)
- The window start position is 75% (WDTCR.RPSS[1:0] = 10b)
- The window end position is 25% (WDTCR.RPES[1:0] = 10b)

Downloaded from Arrow.com.

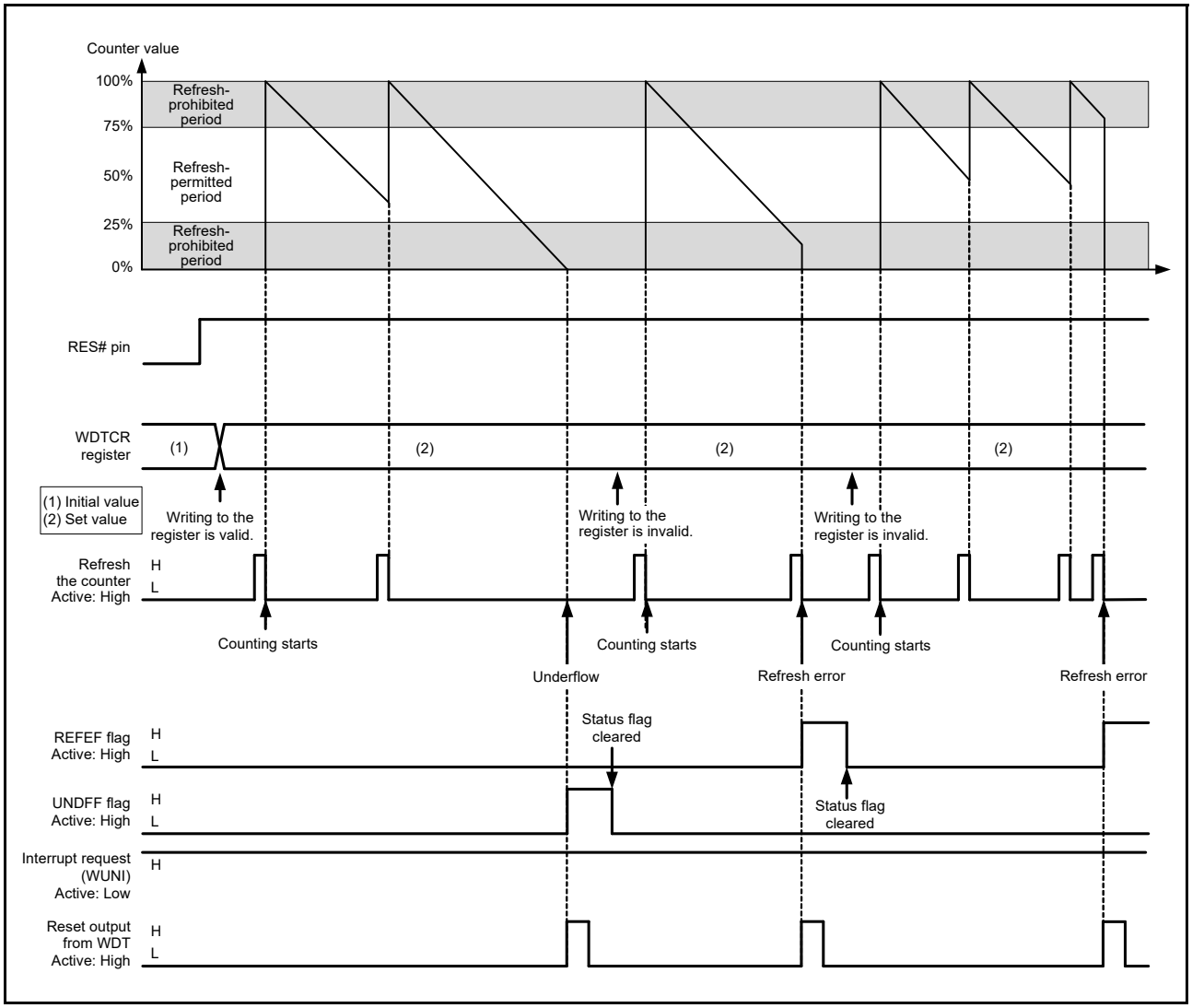


Figure 30.3 Operation Example in Register Start Mode

Downloaded from Arrow.com.

30.3.1.2 Auto-Start Mode

When the WDTSTRT bit in option function select register 0 (OFS0) is 0, auto-start mode is selected, the WDTCR and WDTRCR registers are disabled, and the settings in the OFS0 register are enabled.
Within the reset state, the setting values (clock division ratio, window start and end positions, timeout period, and reset output or interrupt request) of the OFS0 register are set in the WDT registers.
When the reset is released, the down-counter automatically starts counting down from the value set by the OFS0.WDTTOPS[1:0] bits.
After that, as long as the counter is refreshed in the refresh-permitted period, the value in the counter is re-set each time the counter is refreshed and counting down continues. The WDT does not output the reset signal as long as this continues.
However, if the down-counter underflows because refreshing of the down-counter is not possible due to the program having entered crashed execution or if a refresh error occurs due to refreshing outside the refresh-permitted period, the WDT outputs the reset signal or non-maskable interrupt request (WUNI).
After the reset signal or non-maskable interrupt request is output of for one cycle of counting, the value of the timeout period is set in the down-counter counting is restarted.
Reset output or interrupt request output can be selected by setting the OFS0.WDTRSTIRQS bit.

Figure 30.4 shows an example of operation (non-maskable interrupt) under the following conditions.

- Auto start mode (OFS0.WDTSTRT = 0)
- Non-maskable interrupt request output is enabled (OFS0.WDTRSTIRQS = 0)
- The window start position is 75% (OFS0.WDTRPSS[1:0] = 10b)
- The window end position is 25% (OFS0.WDTRPES[1:0] = 10b)

Downloaded from Arrow.com.

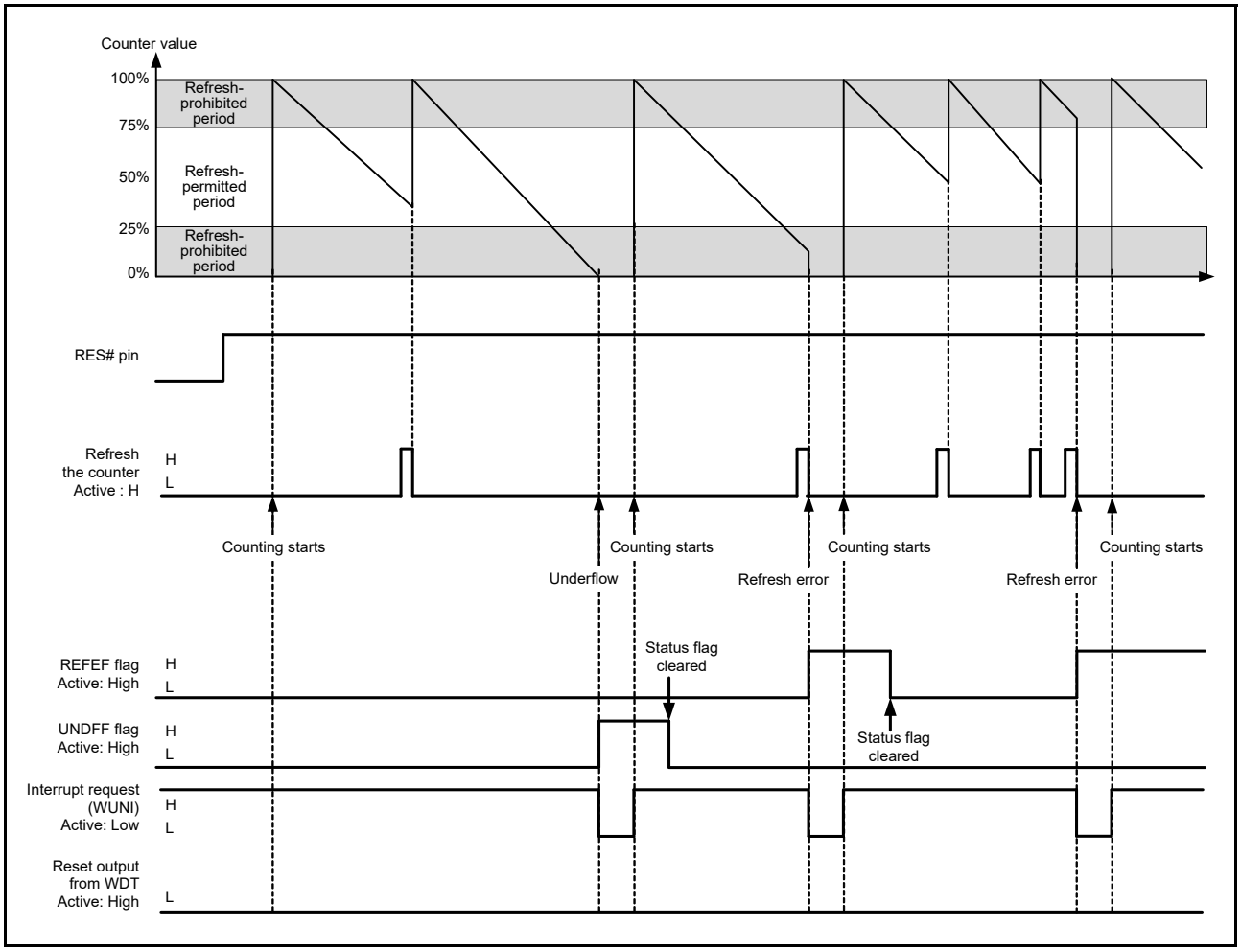


Figure 30.4 Operation Example in Auto-Start Mode

Downloaded from Arrow.com.

30.3.2 Control over Writing to the WDTCR and WDTRCR Registers

Writing to the WDTCR or WDTRCR register is only possible once between the release from the reset state and the first refresh operation.

After a refresh operation (counting starts) or by writing to the WDTCR or WDTRCR register, the protection signal in the WDT becomes 1 to protect the WDTCR and WDTRCR registers against subsequent attempts at writing.

This protection is released by the reset source of the WDT. With other reset sources, the protection is not released.

Figure 30.5 shows control waveforms produced in response to writing to the WDTCR register.

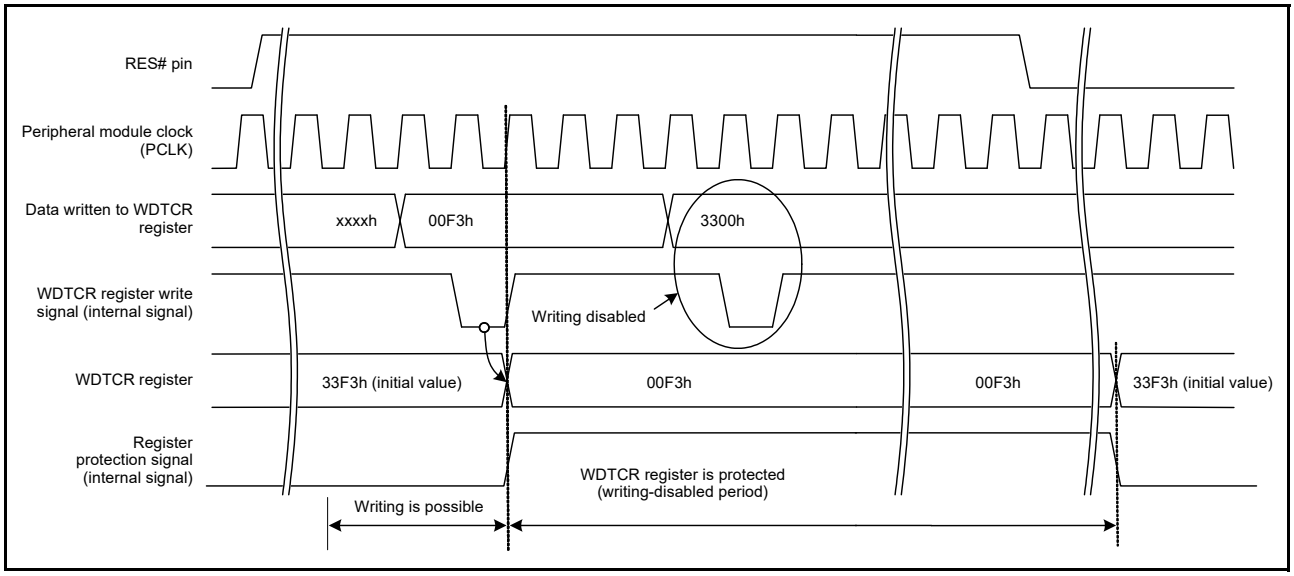


Figure 30.5 Control Waveforms Produced in Response to Writing to the WDTCR Register

30.3.3 Refresh Operation

The down-counter is refreshed by writing the values 00h and then FFh to the WDTRR register. If a value other than FFh is written after 00h, the down-counter is not refreshed. After such invalid writing, correct refreshing is performed by again writing to 00h and then FFh to the WDTRR register.

Even if a register other than the WDTRR register is accessed or the WDTRR register is read between writing 00h and writing FFh to the WDTRR register, correct refreshing will be done.

Writing to refresh the counter must be performed within the refresh-permitted period. Whether writing is done within the refresh-permitted period is determined when writing FFh. For this reason, correct refreshing will be done even if 00h is written outside the refresh-permitted period.

[Sample sequences of writing that are valid for refreshing the counter]

- 00h → FFh
- 00h (n–1-th time) → 00h (nth time) → FFh
- 00h → access to another register or read from the WDTRR register → FFh

[Sample sequences of writing that are not valid for refreshing the counter]

- 23h (a value other than 00h) → FFh
- 00h → 54h (a value other than FFh)
- 00h→ AAh (00h and a value other than FFh) → FFh

Downloaded from Arrow.com.

After FFh is written to the WDTRR register, refreshing the down-counter requires up to four cycles of the signal for counting. Therefore, writing FFh to the WDTRR register should be completed four-count cycles before the down-counter underflows.

Figure 30.6 shows the WDT refresh-operation waveforms when the clock division ratio = PCLK/64.

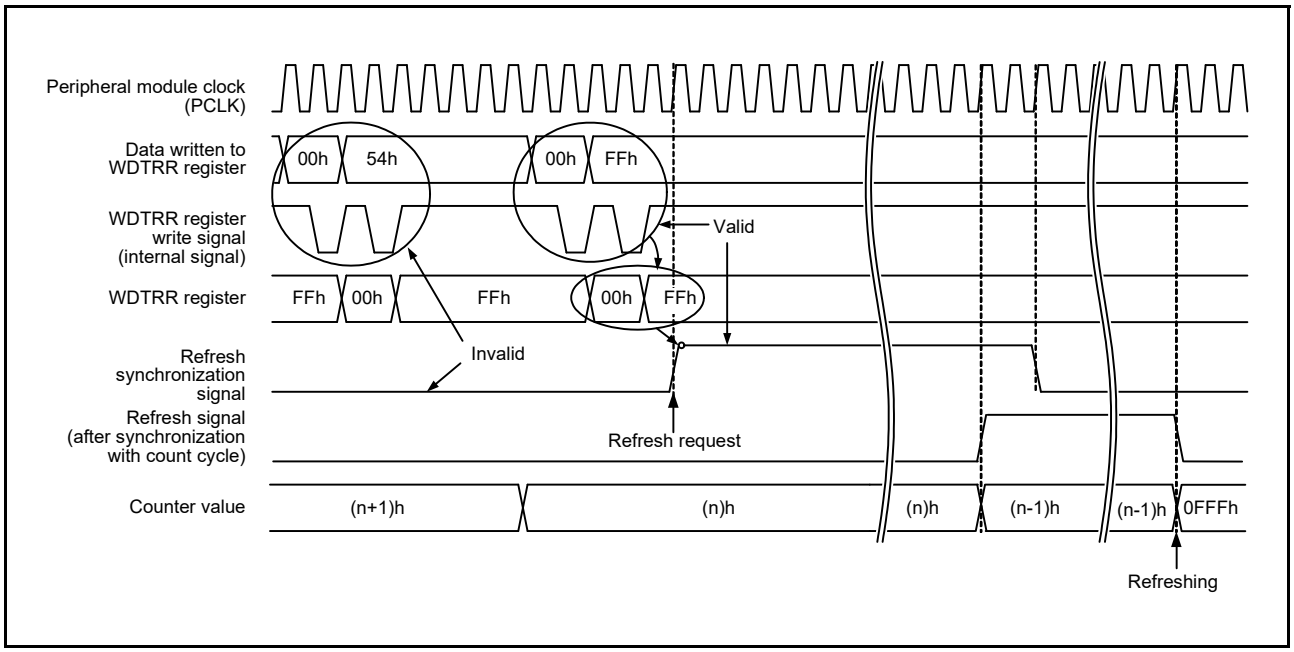


Figure 30.6 WDT Refresh Operation Waveforms (WDTCR.CKS[3:0] = 0100b, WDTCR.TOPS[1:0] = 01b)

30.3.4 Reset Output

When the WDTRCR.RSTIRQS bit is set to 1 in register start mode or when the WDTRSTIRQS bit in option function select register 0 (OFS0) is set to 1 in auto-start mode, a reset signal is output for one-count cycle when an underflow in the down-counter or a refresh error occurs.

In register start mode, the down-counter is initialized (all bits set to 0) and stopped in that state after output of the reset signal. After the reset is released and the program is restarted, the counter is set up again and counting down is started by refreshing.

In auto-start mode, counting down automatically starts after the reset is released.

30.3.5 Interrupt Source

When the WDTRCR.RSTIRQS bit is set to 0 in register start mode or when the OFS0.WDTRSTIRQS bit is set to 0 in auto-start mode, an interrupt (WUNI) signal is generated when an underflow in the counter or a refresh error occurs. This interrupt can be used as a non-maskable interrupt. For details, refer to section 15, Interrupt Controller (ICUb).

Table 30.4 WDT Interrupt Source

Name	Interrupt Source	DTC Activation	DMAC Activation
WUNI	Down-counter underflow Refresh error	Not possible	Not possible

Downloaded from Arrow.com.

30.3.6 Reading the Down-Counter Value

The WDT stores the counter value in the WDTSR.CNTVAL[13:0] bits. Thus, the counter value can be checked through the WDTSR.CNTVAL[13:0] bits.

Figure 30.7 shows the processing for reading the WDT down-counter value when the clock division ratio = PCLK/64.

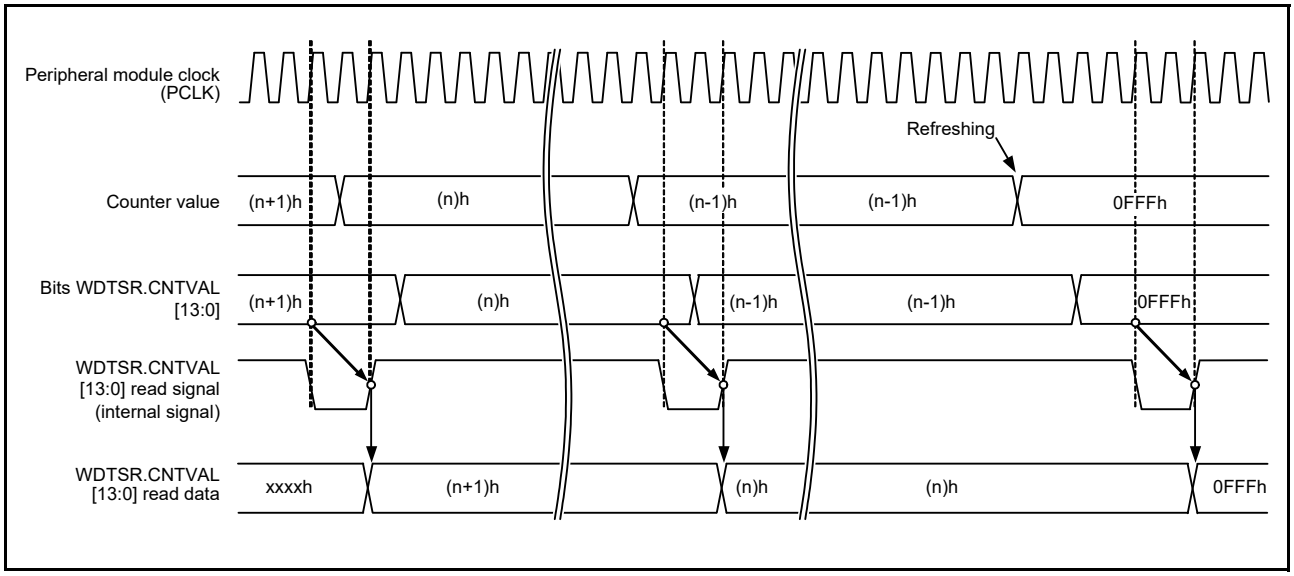


Figure 30.7 Processing for Reading WDT Down-Counter Value (WDTCR.CKS[3:0] = 0100b, WDTCR.TOPS[1:0] = 01b)

30.3.7 Correspondence between Option Function Select Register 0 (OFS0) and WDT Registers

Table 30.5 lists the correspondence between option function select register 0 (OFS0) used in auto-start mode and the registers used in register start mode.

Do not change the OFS0 register setting during WDT operation.

For details on the OFS0 register, refer to section 7.2.1, Option Function Select Register 0 (OFS0).

Table 30.5 Correspondence between Option Function Select Register 0 (OFS0) and WDT Registers

Target of Control	Function	OFS0 Register (Enabled in Auto-Start Mode) OFS0.WDTSTRT = 0	WDT Registers (Enabled in Register Start Mode) OFS0.WDTSTRT = 1
Down-counter	Timeout period selection	OFS0.WDTTOPS[1:0]	WDTCR.TOPS[1:0]
	Clock division ratio selection	OFS0.WDTCKS[3:0]	WDTCR.CKS[3:0]
	Window start position selection	OFS0.WDTRPSS[1:0]	WDTCR.RPSS[1:0]
	Window end position selection	OFS0.WDTRPES[1:0]	WDTCR.RPES[1:0]
Reset output or interrupt request output	Reset output or interrupt request output selection	OFS0.WDTRSTIRQS	WDTRCR.RSTIRQS

Downloaded from Arrow.com.

31. Independent Watchdog Timer (IWDTa)

In this section, “PCLK” is used to refer to PCLKB.

31.1 Overview

The independent watchdog timer (IWDT) can be used to detect programs being out of control.
The user can detect when a program runs out of control if an underflow occurs, by creating a program that refreshes the IWDT counter before it underflows.
The functions of the IWDT are different from those of the WDT in the following respects.

- The divided IWDT-dedicated clock (IWDTCLK) is used as the count source (not affected by the PCLK).
- When making a transition to sleep mode, software standby mode, or deep sleep mode, the IWDTCSTPR.SLCSTP bit or the OFS0.IWDTSLCSTP bit can be used to select whether to stop the counter or not.

Table 31.1 lists the specifications of the IWDT and Figure 31.1 shows a block diagram of the IWDT.

Table 31.1 IWDT Specifications

Item	Description
Count source*1	IWDT-dedicated clock (IWDTCLK)
Clock divide ratio	Divide by 1, 16, 32, 64, 128, or 256
Counter operation	Counting down using a 14-bit down-counter
Conditions for starting the counter	• Auto-start mode: Counting automatically starts after a reset is released • Register start mode: Counting is started by refresh operation (writing 00h and then FFh to the IWDTRR register).
Conditions for stopping the counter	• Reset • In low power consumption states (depends on the register setting*2) • A counter underflows or a refresh error occurs (only in register start mode)
Window function	Window start and end positions can be specified (refresh-permitted and refresh-prohibited periods)
Reset output sources	• Down-counter underflows • Refreshing outside the refresh-permitted period (refresh error)
Non-maskable interrupt sources	• Down-counter underflows • Refreshing outside the refresh-permitted period (refresh error)
Reading the counter value	The down-counter value can be read by the IWDTSR register.
Event link function (output)	• Down-counter underflow event output • Refresh error event output
Output signal (internal signal)	• Reset output • Interrupt request output • Sleep mode count stop control output
Auto-start mode (controlled by option function select register 0 (OFS0))	• Selecting the clock frequency divide ratio after a reset (OFS0.IWDTCKS[3:0] bits) • Selecting the timeout period of the independent watchdog timer (OFS0.IWDTTOPS[1:0] bits) • Selecting the window start position in the independent watchdog timer (OFS0.IWDTRPSS[1:0] bits) • Selecting the window end position in the independent watchdog timer (OFS0.IWDTRPES[1:0] bits) • Selecting the reset output or interrupt request output (OFS0.IWDTRSTIRQS bit) • Selecting the down-count stop function at transition to sleep mode, software standby mode, or deep sleep mode (OFS0.IWDTSLCSTP bit)
Register start mode (controlled by the IWDT registers)	• Selecting the clock frequency divide ratio after refreshing (IWDTCR.CKS[3:0] bits) • Selecting the timeout period of the independent watchdog timer (IWDTCR.TOPS[1:0] bits) • Selecting the window start position in the independent watchdog timer (IWDTCR.RPSS[1:0] bits) • Selecting the window end position in the independent watchdog timer (IWDTCR.RPES[1:0] bits) • Selecting the reset output or interrupt request output (IWDTRCR.RSTIRQS bit) • Selecting the down-count stop function at transition to sleep mode, software standby mode, or deep sleep mode (IWDTCSTPR.SLCSTP bit)

Note 1. Satisfy the frequency of the peripheral module clock (PCLK) ≥ 4 × (the frequency of the count source after divide).
Note 2. When the OFS0.IWDTSLCSTP bit is 1 in auto-start mode, and when the IWDTCSTPR.SLCSTP bit is 1 in register start mode.

Downloaded from Arrow.com.

To use the IWDT, the IWDT-dedicated clock (IWDTCLK) should be supplied so that the IWDT operates even if the peripheral module clock (PCLK) stops. The bus interface and registers operate with PCLK, and the 14-bit counter and control circuits operate with IWDTCLK.
Figure 31.1 is a block diagram of the IWDT.

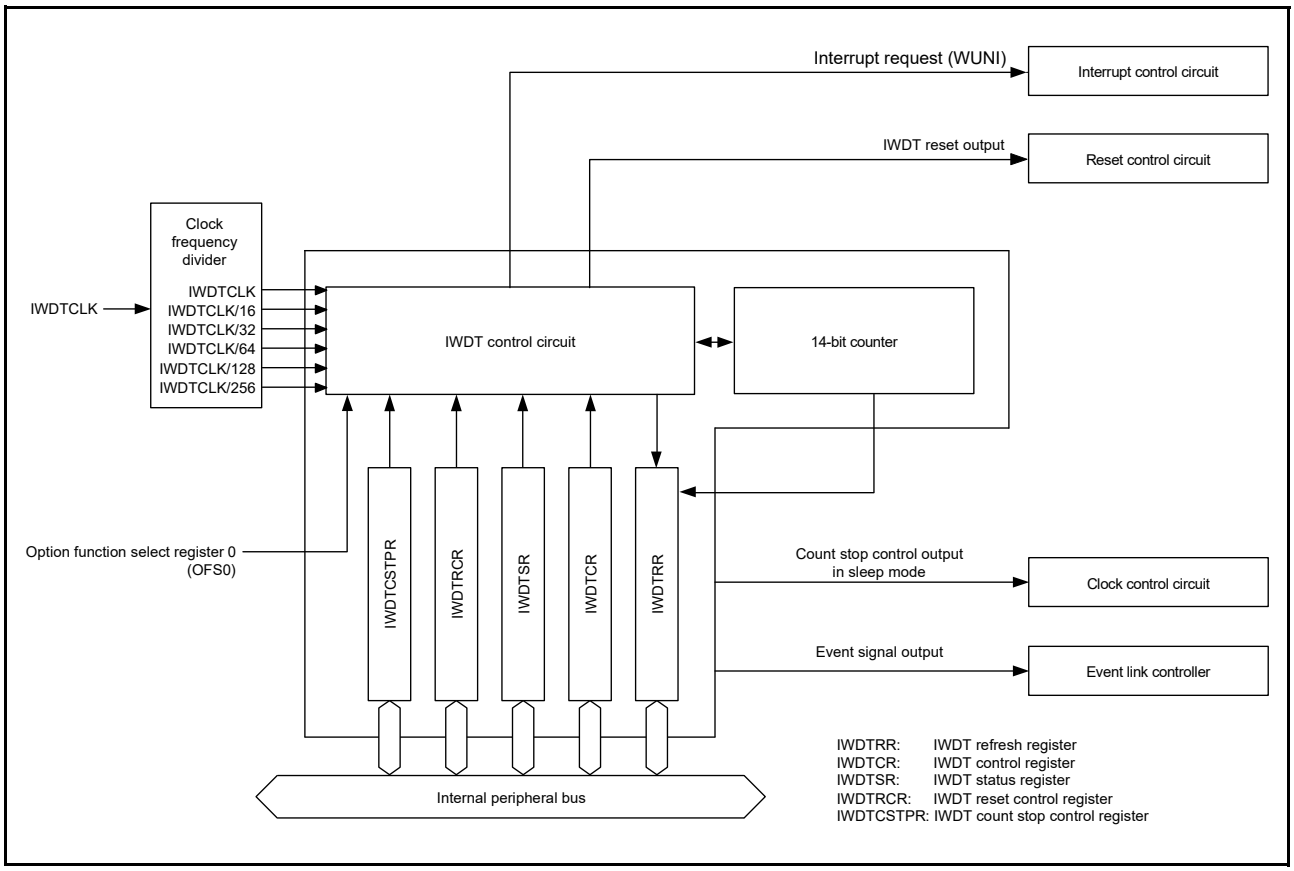


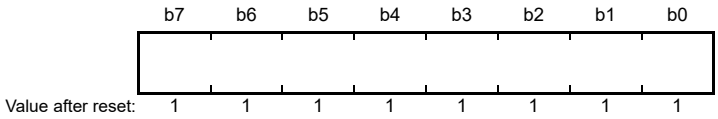
Figure 31.1 IWDT Block Diagram

Downloaded from Arrow.com.

31.2 Register Descriptions

31.2.1 IWDT Refresh Register (IWDTRR)

Address(es): IWDT.IWDTRR 0008 8030h



Bit	Description	R/W
b7 to b0	The counter is refreshed by writing 00h and then writing FFh to this register.	R/W

The IWDTRR register refreshes the counter of the IWDT.
The counter of the IWDT is refreshed by writing 00h and then writing FFh to the IWDTRR register (refresh operation) within the refresh-permitted period.
After the counter has been refreshed, it starts counting down from the value selected by the IWDTTOPS[1:0] bits in option function select register 0 (OFS0) in auto-start mode. In register start mode, counting down starts from the value selected by setting the IWDTCR.TOPS[1:0] bits in the first refresh operation after a reset is released.
When 00h is written, the read value is 00h. When a value other than 00h is written, the read value is FFh.
For details of the refresh operation, refer to section 31.3.3, Refresh Operation.

Downloaded from Arrow.com.

31.2.2 IWDT Control Register (IWDTCR)

Address(es): IWDT.IWDTCR 0008 8032h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	RPSS[1:0]		—	—	RPES[1:0]		CKS[3:0]				—	—	TOPS[1:0]	
Value after reset:	0	0	1	1	0	0	1	1	1	1	1	1	0	0	1	1

Bit	Symbol	Bit Name	Description	R/W
b1, b0	TOPS[1:0]	Timeout Period Select	b1 b0 0 0: 128 cycles (007Fh) 0 1: 512 cycles (01FFh) 1 0: 1024 cycles (03FFh) 1 1: 2048 cycles (07FFh)	R/W
b3, b2	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b7 to b4	CKS[3:0]	Clock Divide Ratio Select	b7 b4 0 0 0 0: No division 0 0 1 0: Divide-by-16 0 0 1 1: Divide-by-32 0 1 0 0: Divide-by-64 1 1 1 1: Divide-by-128 0 1 0 1: Divide-by-256 Other settings are prohibited.	R/W
b9, b8	RPES[1:0]	Window End Position Select	b9 b8 0 0: 75% 0 1: 50% 1 0: 25% 1 1: 0% (window end position is not specified.)	R/W
b11, b10	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b13, b12	RPSS[1:0]	Window Start Position Select	b13 b12 0 0: 25% 0 1: 50% 1 0: 75% 1 1: 100% (window start position is not specified.)	R/W
b15, b14	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R

There are some restrictions on writing to the IWDTCR register. For details, refer to section 31.3.2, Control over Writing to the IWDTCR, IWDTRCR, and IWDTCSTPR Registers.

In auto-start mode, the settings in the IWDTCR register are disabled, and the settings in option function select register 0 (OFS0) are enabled. The bit setting made to the IWDTCR register can also be made in the OFS0 register. For details, refer to section 31.3.8, Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers.

Downloaded from Arrow.com.

TOPS[1:0] Bits (Timeout Period Select)

These bits select the timeout period (period until the counter underflows) from among 128, 512, 1024, or 2048 cycles, taking the divided clock specified by the CKS[3:0] bits as one cycle.
After the counter is refreshed, the combination of the CKS[3:0] and TOPS[1:0] bits determines the time (number of IWDTCLK cycles) until the counter underflows.
Relations between the CKS[3:0] and TOPS[1:0] bit setting, the timeout period, and the number of IWDTCLK cycles are listed in Table 31.2.

Table 31.2 Settings and Timeout Periods

CKS[3:0] Bits				TOPS[1:0] Bits				
b7	b6	b5	b4	b1	b0	Clock Divide Ratio	Timeout Period (Number of Cycles)	Cycles of IWDTCLK
0	0	0	0	0	0	No division	128	128
				0	1		512	512
				1	0		1024	1024
				1	1		2048	2048
0	0	1	0	0	0	Divide-by-16	128	2048
				0	1		512	8192
				1	0		1024	16384
				1	1		2048	32768
0	0	1	1	0	0	Divide-by-32	128	4096
				0	1		512	16384
				1	0		1024	32768
				1	1		2048	65536
0	1	0	0	0	0	Divide-by-64	128	8192
				0	1		512	32768
				1	0		1024	65536
				1	1		2048	131072
1	1	1	1	0	0	Divide-by-128	128	16384
				0	1		512	65536
				1	0		1024	131072
				1	1		2048	262144
0	1	0	1	0	0	Divide-by-256	128	32768
				0	1		512	131072
				1	0		1024	262144
				1	1		2048	524288

CKS[3:0] Bits (Clock Divide Ratio Select)

These bits select the IWDTCLK clock divide ratio from among divide-by 1, 16, 32, 64, 128, and 256. Combination with the TOPS[1:0] bit setting, a count period between 128 and 524288 cycles of the IWDTCLK clock can be selected for the IWDT.

Downloaded from Arrow.com.

RPES[1:0] Bits (Window End Position Select)

These bits select 75%, 50%, 25% or 0% of the count period for the window end position of the counter. The window end position should be a value smaller than the window start position (window start position > window end position). If the window end position is greater than the window start position, only the window start position setting is enabled.
The counter values for the window start and end positions selected by setting the RPSS[1:0] and RPES[1:0] bits change depending on the TOPS[1:0] bit setting.
Table 31.3 lists the counter values for the window start and end positions corresponding to TOPS[1:0] bit values.

Table 31.3 Relationship between Timeout Period and Window Start and End Counter Values

TOPS[1:0] Bits		Timeout Period		Window Start and End Counter Value			
b1	b0	Cycles	Counter Value	100%	75%	50%	25%
0	0	128	007Fh	007Fh	005Fh	003Fh	001Fh
0	1	512	01FFh	01FFh	017Fh	00FFh	007Fh
1	0	1024	03FFh	03FFh	02FFh	01FFh	00FFh
1	1	2048	07FFh	07FFh	05FFh	03FFh	01FFh

RPSS[1:0] Bits (Window Start Position Select)

These bits select a counter window start position from 100%, 75%, 50%, or 25% of the count period (100% when the count starts and 0% when the counter underflows). The interval between the window start position and window end position is the refresh-permitted period and the other periods are refresh-prohibited periods.
Figure 31.2 shows the relationship between of the RPSS[1:0] and RPES[1:0] bit setting and the refresh-permitted and refresh-prohibited periods.

RPSS[1:0] Bits		RPES[1:0] Bits		Window	
b13	b12	b9	b8	Start (%)	End (%)
1	1	1	1	100	0
		1	0		25
		0	1		50
		0	0		75
1	0	1	1	75	0
		1	0		25
		0	1		50
		0	0		75
0	1	1	1	50	0
		1	0		25
		0	1		50
		0	0		75
0	0	1	1	25	0
		1	0		25
		0	1		50
		0	0		75

Note: If window end setting is greater than or equal to window start setting, the window end setting is set to 0%.

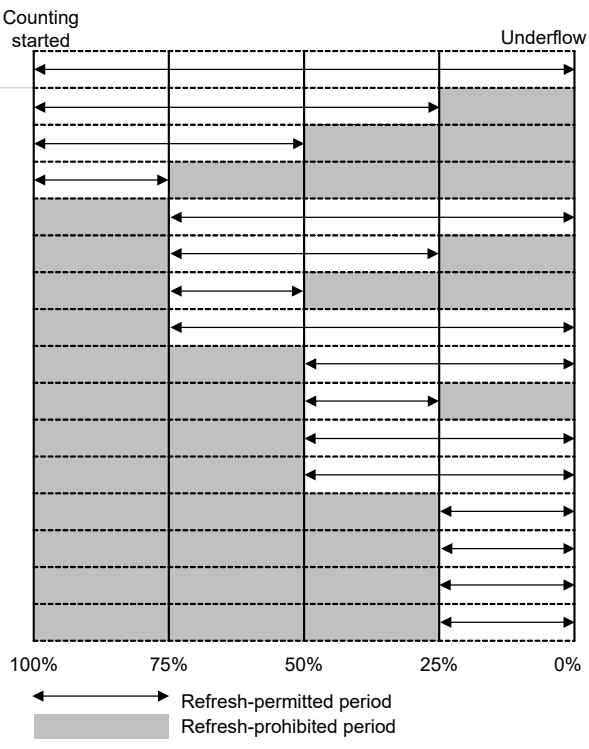


Figure 31.2 RPSS[1:0] and RPES[1:0] Bit Settings and the Refresh-Permitted Period

Downloaded from Arrow.com.

31.2.3 IWDT Status Register (IWDTSR)

Address(es): IWDT.IWDTSR 0008 8034h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	REFEF	UNDFF	CNTVAL[13:0]													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b13 to b0	CNTVAL[13:0]	Counter Value	Value counted by the counter	R
b14	UNDFF	Underflow Flag	0: No underflow occurred 1: Underflow occurred	R/(W) *1
b15	REFEF	Refresh Error Flag	0: No refresh error occurred 1: Refresh error occurred	R/(W) *1

Note 1. Only 0 can be written to clear the flag.

CNTVAL[13:0] Bits (Counter Value)

These bits are used to confirm the counter value of the counter, but note that the read value may differ from the actual count by a value of one count.

UNDFF Flag (Underflow Flag)

This bit is used to confirm whether or not an underflow has occurred in the counter.
The value 1 indicates that the counter has underflowed. The value 0 indicates that the counter has not underflowed.
Write 0 to the UNDFF flag to set the value to 0. Writing 1 has no effect.

REFEF Flag (Refresh Error Flag)

This bit is used to confirm whether or not a refresh error (performing a refresh operation during a refresh-prohibited period).
The value 1 indicates that a refresh error has occurred. The value 0 indicates that no refresh error has occurred.
Write 0 to the REFEF flag to set the value to 0. Writing 1 has no effect.

Downloaded from Arrow.com.

31.2.4 IWDT Reset Control Register (IWDTRCR)

Address(es): IWDT.IWDTRCR 0008 8036h

	b7	b6	b5	b4	b3	b2	b1	b0
	RSTIRQS	—	—	—	—	—	—	—
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b7	RSTIRQS	Reset Interrupt Request Select	0: Non-maskable interrupt request output is enabled. 1: Reset output is enabled.	R/W

There are some restrictions on writing to the IWDTRCR register. For details, refer to section 31.3.2, Control over Writing to the IWDTCR, IWDTRCR, and IWDTCSTPR Registers.

In auto-start mode, the IWDTRCR register setting are disabled, and the settings in option function select register 0 (OFS0) are enabled. The bit setting mode to the IWDTRCR register can also be made in the OFS0 register. For details, refer to section 31.3.8, Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers.

Downloaded from Arrow.com.

31.2.5 IWDT Count Stop Control Register (IWDTCSTPR)

Address(es): IWDT.IWDTCSTPR 0008 8038h

	b7	b6	b5	b4	b3	b2	b1	b0
	SLCSTP	—	—	—	—	—	—	—
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b7	SLCSTP	Sleep Mode Count Stop Control	0: Count stop is disabled. 1: Count is stopped at a transition to sleep mode, software standby mode, or deep sleep mode.	R/W

The IWDTCSTPR register controls whether to stop the IWDT counter in a low power consumption state. There are some restrictions on writing to the IWDTCSTPR register. For details, refer to section 31.3.2, Control over Writing to the IWDTCR, IWDTRCR, and IWDTCSTPR Registers.
In auto-start mode, the IWDTCSTPR register setting are disabled, and the settings in option function select register 0 (OFS0) are enabled. The bit setting mode to the IWDTCSTPR register can also be made in the OFS0 register. For details, refer to section 31.3.8, Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers.

SLCSTP Bit (Sleep Mode Count Stop Control)

This bit selects whether to stop counting at a transition to sleep mode, software standby mode, or deep sleep mode.

31.2.6 Option Function Select Register 0 (OFS0)

For option function select register 0 (OFS0), refer to section 31.3.8, Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers.

Downloaded from Arrow.com.

31.3 Operation

31.3.1 Count Operation in Each Start Mode

Select the IWDT start mode by setting the IWDTSTRT bit in option function select register 0 (OFS0).
When the OFS0.IWDTSTRT bit is 1 (register start mode), the IWDTCR, IWDTRCR, and IWDTCSTPR registers are enabled, and counting is started by refresh operation (writing) to the IWDTRR register. When the OFS0.IWDTSTRT bit is 0 (auto-start mode), the setting of the OFS0 register is enabled, and counting automatically starts after reset.

31.3.1.1 Register Start Mode

When the OFS0.IWDTSTRT bit in option function select register 0 is 1, register start mode is selected, and the IWDTCR, IWDTRCR, and IWDTCSTPR registers are enabled.
After a reset is released, set the clock divide ratio, window start and end positions, and timeout period in the IWDTCR register, the reset output or interrupt request output in the IWDTRCR register, and the counter stop control at transitions to low power consumption states in the IWDTCSTPR register. Then refresh the counter to start counting down from the value selected by setting the IWDTCR.TOPS[1:0] bits.
Thereafter, as long as the program continues normal operation and the counter is refreshed in the refresh-permitted period, the value in the counter is re-set each time the counter is refreshed and counting down continues. The IWDT does not output the reset signal as long as this continues. However, if the counter underflows because the counter cannot be refreshed due to a program runaway, or if a refresh error occurs because the counter was refreshed outside the refresh-permitted period, the IWDT outputs a reset signal or a non-maskable interrupt request (WUNI). Set the IWDTRCR.RSTIRQS bit to select either reset output or interrupt request output.

Figure 31.3 shows an example of operation under the following conditions.

- Register start mode (OFS0.IWDTSTRT = 1)
- Reset output is enabled (IWDTRCR.RSTIRQS = 1)
- The window start position is 75% (IWDTCR.RPSS[1:0] = 10b)
- The window end position is 25% (IWDTCR.RPES[1:0] = 10b)

Downloaded from Arrow.com.

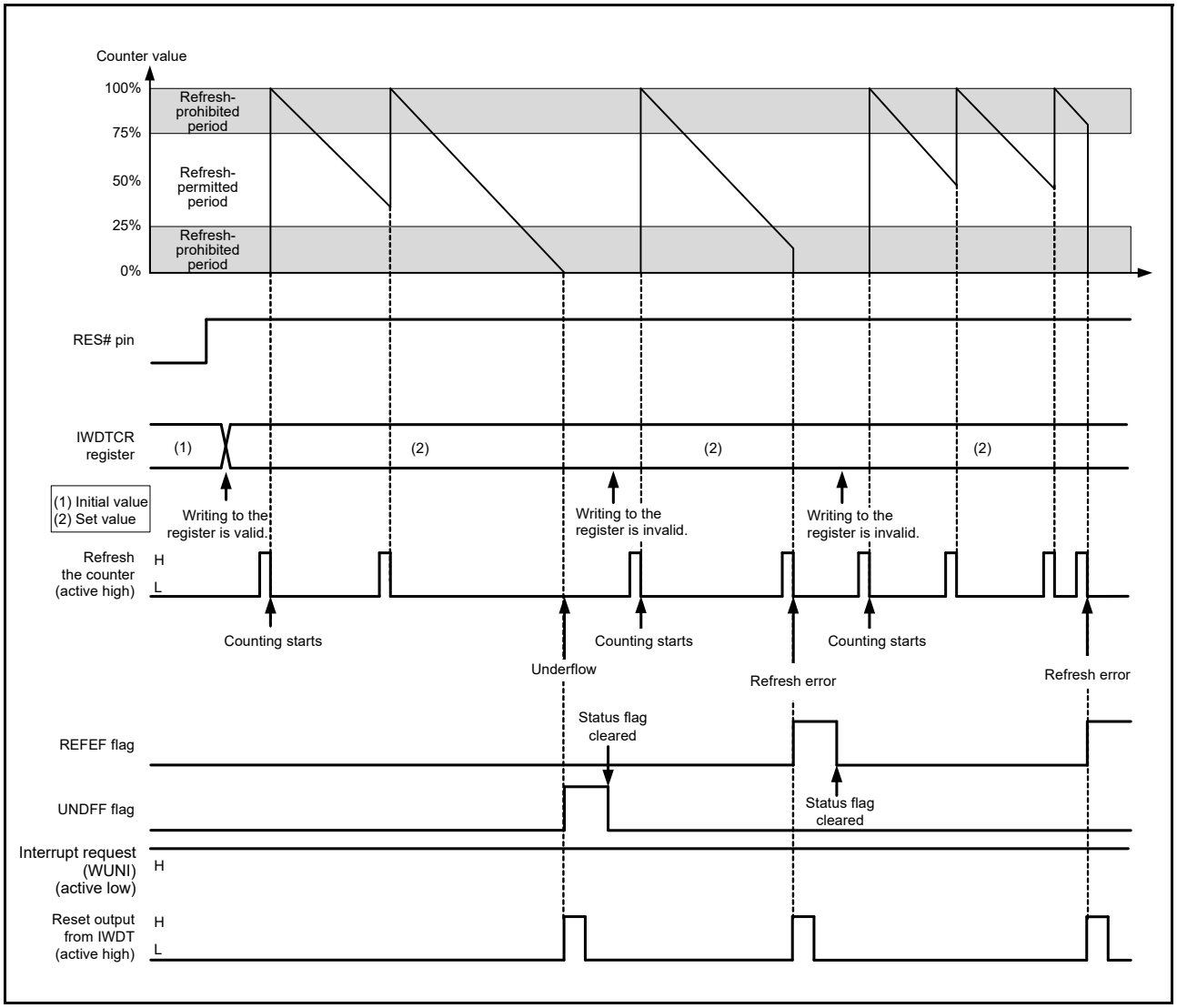


Figure 31.3 Operation Example in Register Start Mode

Downloaded from Arrow.com.

31.3.1.2 Auto-Start Mode

When the IWDTSTRT bit in option function select register 0 (OFS0) is 0, auto-start mode is selected, and the IWDTCR, IWDTRCR, and IWDTCSTPR registers are disabled.
Within the reset state, the clock divide ratio, window start and end positions, timeout period, reset output or interrupt request output, and counter stop control at transitions to low power consumption states are set using the values specified in the OFS0 register. When the reset is released, the counter automatically starts counting down from the value selected by the OFS0.IWDTTOPS[1:0] bits.
After that, as long as the program continues normal operation and the counter is refreshed in the refresh-permitted period, the value in the counter is re-set each time the counter is refreshed and counting down continues. The IWDT does not output the reset signal as long as this continues. However, if the counter underflows because refreshing of the counter is not possible due to the program having entered crashed execution or if a refresh error occurs due to refreshing outside the refresh-permitted period, the IWDT outputs the reset signal or non-maskable interrupt request (WUNI). After the reset signal or non-maskable interrupt request (WUNI) is generated, the counter reloads the timeout period after counting for one cycle, and restarts counting. Set the OFS0.IWDTRSTIRQS bit to select either reset output or interrupt request output.

Figure 31.4 shows an example of operation under the following conditions.

- Auto-start mode (OFS0.IWDTSTRT = 0)
- Non-maskable interrupt request output is enabled (OFS0.IWDTRSTIRQS = 0)
- The window start position is 75% (OFS0.IWDTRPSS[1:0] = 10b)
- The window end position is 25% (OFS0.IWDTRPES[1:0] = 10b)

Downloaded from Arrow.com.

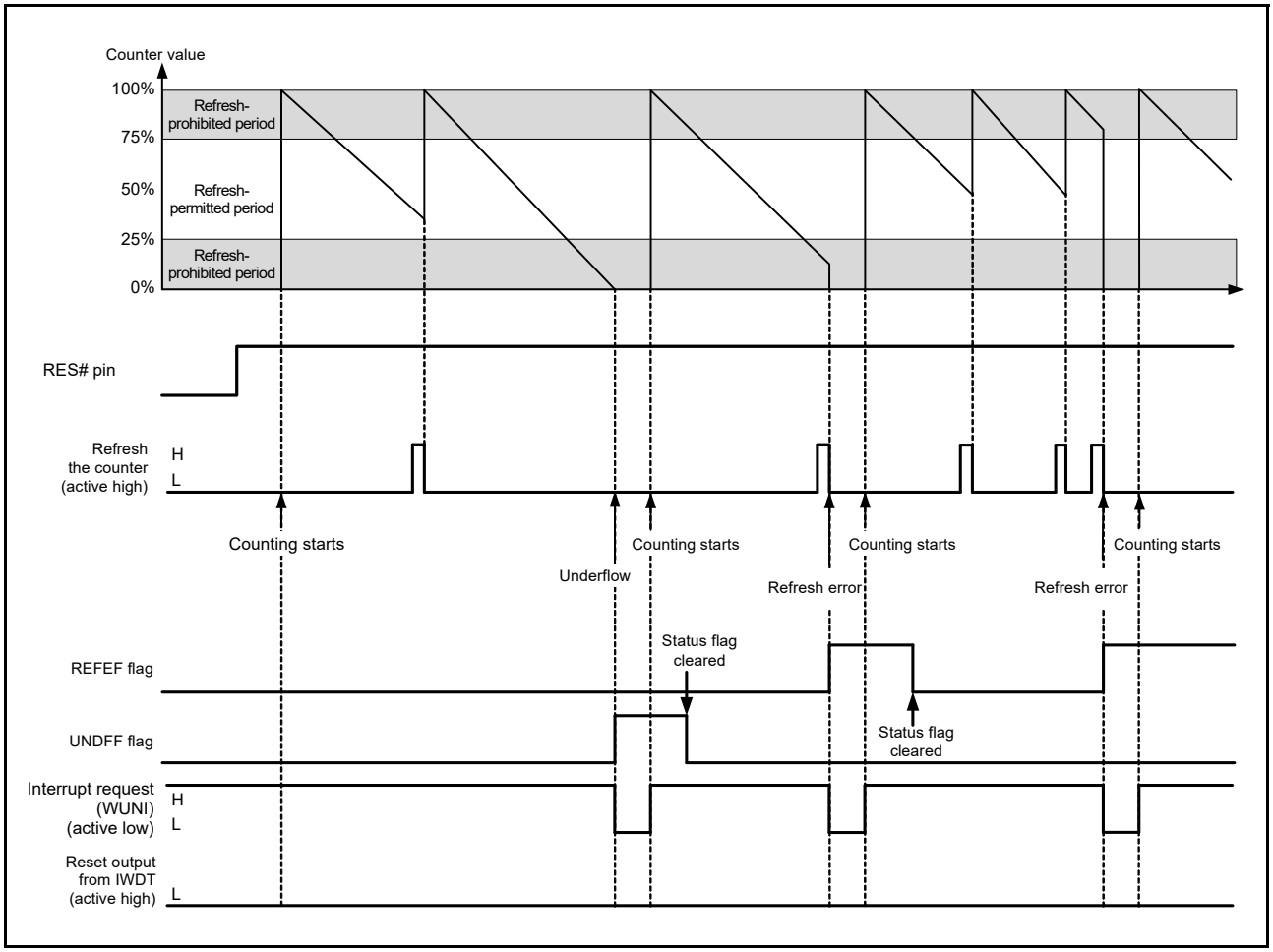


Figure 31.4 Operation Example in Auto-Start Mode

Downloaded from Arrow.com.

31.3.2 Control over Writing to the IWDTCR, IWDTRCR, and IWDTCSTPR Registers

Writing to the IWDTCR, IWDTRCR, or IWDTCSTPR register is only possible once between the release from the reset state and the first refresh operation.

After a refresh operation (counting starts) or the IWDTCR, IWDTRCR, or IWDTCSTPR register is written to, the protection signal in the IWDT becomes 1 to protect registers IWDTCR, IWDTRCR, and IWDTCSTPR against subsequent attempts at writing.

This protection is released by the reset source of the IWDT. With other reset sources, the protection is not released.

Figure 31.5 shows control waveforms produced in response to writing to the IWDTCR register.

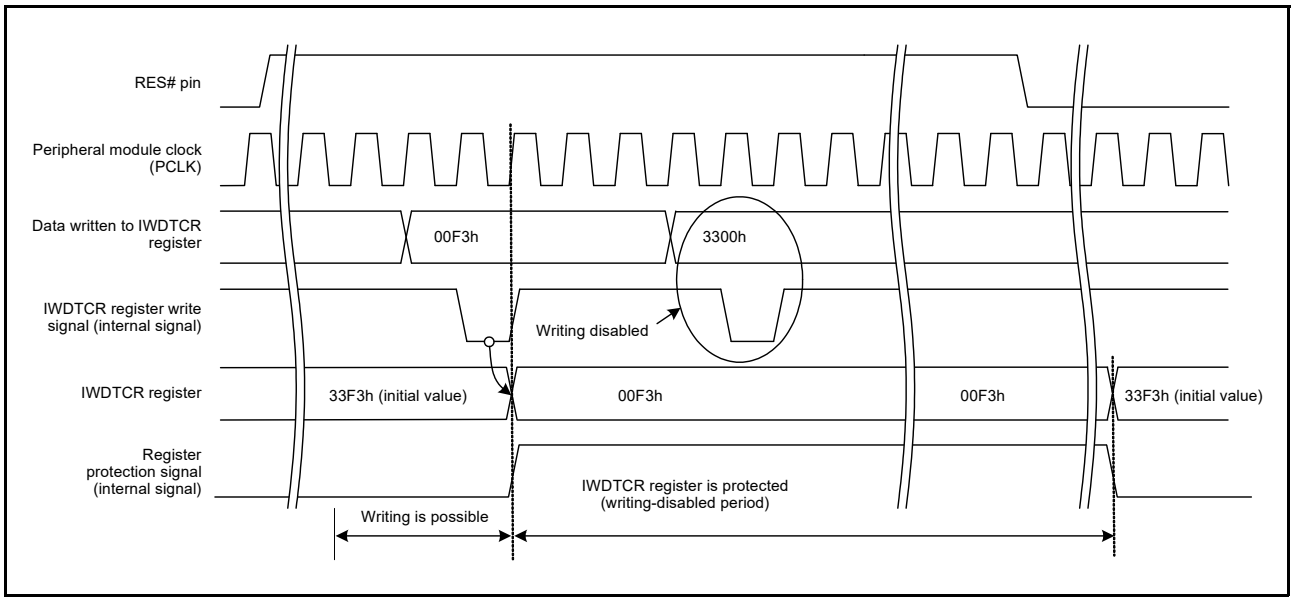


Figure 31.5 Control Waveforms Produced in Response to Writing to the IWDTCR Register

Downloaded from Arrow.com.

31.3.3 Refresh Operation

The counter is refreshed and starts operation (counting is started by refreshing) by writing the values 00h and then FFh to the IWDTRR register. If a value other than FFh is written after 00h, the counter is not refreshed. After such invalid writing, correct refreshing is performed by again writing 00h and then FFh to the IWDTRR register.
When writing is done in the order of 00h (first time) → 00h (second time), and if FFh is written after that, the writing order 00h → FFh is satisfied; writing 00h (n–1-th time) → 00h (nth time) → FFh is valid and correct refreshing will be done. Even when the first value written before 00h is not 00h, correct refreshing will be done if the operation contains the set of writing 00h → FFh. Moreover, even if a register other than the IWDTRR register is accessed or the IWDTRR register is read between writing 00h and writing FFh to the IWDTRR register, correct refreshing will be done.

[Sample sequences of writing that are valid for refreshing the counter]

- 00h → FFh
- 00h (n–1-th time) → 00h (nth time) → FFh
- 00h → access to another register or read from the IWDTRR register → FFh

[Sample sequences of writing that are not valid for refreshing the counter]

- 23h (a value other than 00h) → FFh
- 00h → 54h (a value other than FFh)
- 00h → AAh (00h and a value other than FFh) → FFh

Even when 00h is written to the IWDTRR register outside the refresh-permitted period, if FFh is written to the IWDTRR register in the refresh-permitted period, the writing sequence is valid and refreshing will be done.
After FFh is written to the IWDTRR register, refreshing the counter requires up to four cycles of the signal for counting (the IWDTCR.CKS[3:0] bits determine how many cycles of the IWDT-dedicated clock (IWDTCLK) make up one cycle for counting). Therefore, writing FFh to the IWDTRR register should be completed four-count cycles before the end position of the refresh-permitted period or a counter underflow. The value of the counter can be checked by the IWDTSR.CNTVAL[13:0] bits.

[Sample refreshing timings]

- When the window start position is set to 03FFh, even if 00h is written to the IWDTRR register before 03FFh is reached (0402h, for example), refreshing is done if FFh is written to the IWDTRR register after the value of the IWDTSR.CNTVAL[13:0] bits has reached 03FFh.
- When the window end position is set to 03FFh, refreshing is done if 0403h (four-count cycles before 03FFh) or a greater value is read from the IWDTSR.CNTVAL[13:0] bits immediately after writing 00h → FFh to the IWDTRR register.
- When the refresh-permitted period continues until count 0000h, refreshing can be done immediately before an underflow. In this case, if 0003h (four-count cycles before an underflow) or a greater value is read from the IWDTSR.CNTVAL[13:0] bits immediately after writing 00h → FFh to the IWDTRR register, no underflow occurs and refreshing is done.

Downloaded from Arrow.com.

Figure 31.6 shows the IWDT refresh-operation waveforms when PCLK > IWDTCLK and clock divide ratio = IWDTCLK.

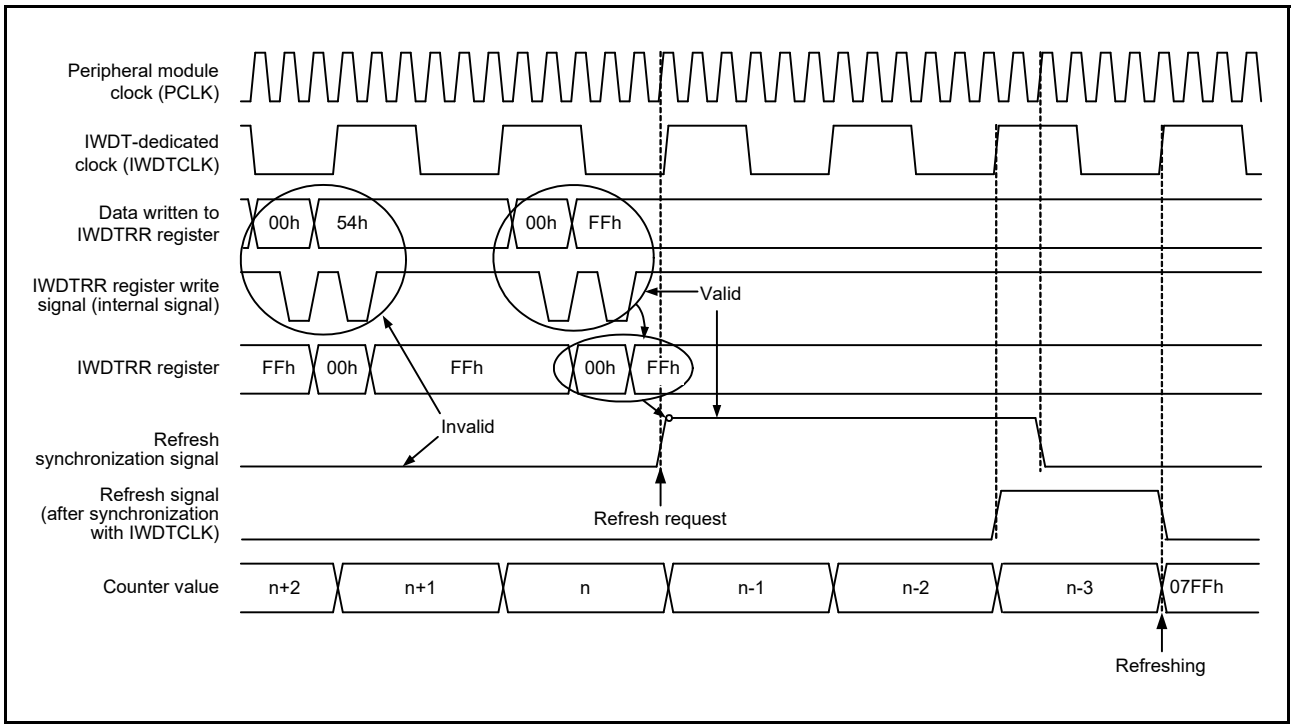


Figure 31.6 IWDT Refresh Operation Waveforms (IWDTCR.CKS[3:0] = 0000b, IWDTCR.TOPS[1:0] = 11b)

Downloaded from Arrow.com.

31.3.4 Status Flags

The IWDTSR.REFEF and IWDTSR.UNDFF flags retain the source of the reset signal output from the IWDT or the source of the interrupt request from the IWDT.
Thus, after release from the reset state or interrupt request generation, read the IWDTSR.REFEF and IWDTSR.UNDFF flags to check for the reset or interrupt source.
For each flag, writing 0 clears the bit and writing 1 has no effect.
Leaving the status flags unchanged does not affect operation. If the flags are not cleared, at the time of the next reset or interrupt request from the IWDT, the earlier reset or interrupt source is cleared and the new reset or interrupt source is written.
After 0 is written to each flag, up to three IWDTCLK cycles and two PCLK cycles are required before the value is reflected.

31.3.5 Reset Output

When the IWDTRCR.RSTIRQS bit is set to 1 in register start mode or when the IWDTRSTIRQS bit in option function select register 0 (OFS0) is set to 1 in auto-start mode, a reset signal is output when an underflow in the counter or a refresh error occurs.
In register start mode, the counter is initialized (0000h) and kept in that state after assertion of the reset signal. After the reset is released and the program is restarted, the counter is set up again and counting down is started by refreshing.
In auto-start mode, counting down automatically starts after the reset output.

31.3.6 Interrupt Sources

When the IWDTRCR.RSTIRQS bit is set to 0 in register start mode or when the OFS0.IWDTRSTIRQS bit is set to 0 in auto-start mode, an interrupt (WUNI) signal is output when an underflow in the counter or a refresh error occurs. This interrupt can be used as a non-maskable interrupt. For details, refer to section 15, Interrupt Controller (ICUb).

Table 31.4 IWDT Interrupt Source

Name	Interrupt Source	DTC Activation	DMAC Activation
WUNI	Counter underflow Refresh error	Not possible	Not possible

Downloaded from Arrow.com.

31.3.7 Reading the Counter Value

As the counter in IWDT-dedicated clock (IWDTCLK), the counter value cannot be read directly. The IWDT synchronizes the counter value with the peripheral module clock (PCLK) and stores it in the IWDTSR.CNTVAL[13:0] bits. Thus, the counter value can be checked indirectly through the IWDTSR.CNTVAL[13:0] bits.
Reading the counter value requires multiple PCLK clock cycles (up to four clock cycles), and the read counter value may differ from the actual counter value by a value of one count.
Figure 31.7 shows the processing for reading the IWDT counter value when PCLK > IWDTCLK and clock divide ratio = IWDTCLK.

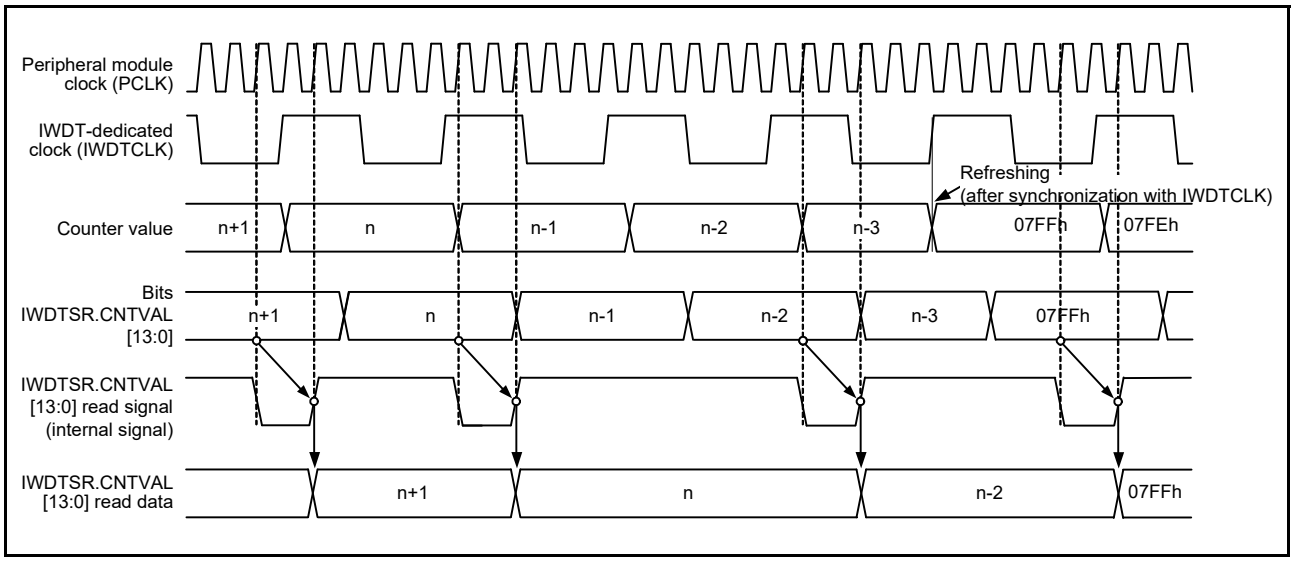


Figure 31.7 Processing for Reading IWDT Counter Value (IWDTCR.CKS[3:0] = 0000b, IWDTCR.TOPS[1:0] = 11b)

Downloaded from Arrow.com.

31.3.8 Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers

Table 31.5 lists the correspondence between option function select register 0 (OFS0) used in auto-start mode and the registers used in register start mode.
Do not change the OFS0 register setting during IWDT operation.
For details on the OFS0 register, refer to section 7.2.1, Option Function Select Register 0 (OFS0).

Table 31.5 Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers

Target of Control	Function	OFS0 Register (Enabled in Auto-Start Mode) OFS0.IWDTSTRT = 0	IWDT Registers (Enabled in Register Start Mode) OFS0.IWDTSTRT = 1
Counter	Timeout period selection	OFS0.IWDTTOPS[1:0]	IWDTCR.TOPS[1:0]
	Clock frequency divide ratio selection	OFS0.IWDTCKS[3:0]	IWDTCR.CKS[3:0]
	Window start position selection	OFS0.IWDTRPSS[1:0]	IWDTCR.RPSS[1:0]
	Window end position selection	OFS0.IWDTRPES[1:0]	IWDTCR.RPES[1:0]
Reset output or interrupt request output	Reset output or interrupt request output selection	OFS0.IWDTRSTIRQS	IWDTRCR.RSTIRQS
Count stop	Sleep mode count stop control	OFS0.IWDTSLCSTP	IWDTCSTPR.SLCSTP

31.4 Link Operation by ELC

The event link controller (ELC) can use the interrupt request signal generated by the IWDT as the event signal. Therefore, the ELC generates an event to the module specified previously when the IWDT outputs an interrupt request. The event signal is output by the counter underflow and refresh error.
An event signal is output regardless of the setting of the IWDTRCR.RSTIRQS bit in register start mode or the OFS0.IWDTRSTIRQS bit in auto-start mode. An event signal can also be output upon generation of the next interrupt source while the IWDTSR.REFEF or IWDTSR.UNDFF flag is 1.
For details, see section 20, Event Link Controller (ELC).

31.5 Usage Notes

31.5.1 Refresh Operations

When making the settings to control the timing of refreshing, consider variations in the range of errors due to the accuracy of the PCLK and IWDTCLK and set values which ensure that refreshing is possible.

31.5.2 Clock Divide Ratio Setting

Satisfy the frequency of the peripheral module clock (PCLK) ≥ 4 × (the frequency of the count source after divide).

Downloaded from Arrow.com.

32. USB 2.0 Host/Function Module (USBc)

32.1 Overview

This MCU incorporates a USB 2.0 host/function module (USB0).
The USB module is a USB controller that is equipped to operate as a host controller or function controller. The module supports full-speed and low-speed transfer as defined in Universal Serial Bus (USB) Specification 2.0. The module has an internal USB transceiver and supports all of the transfer types defined in USB Specification 2.0. It also supports Battery Charging Specification Revision 1.2.
The USB has buffer memory for data transfer, providing a maximum of 10 pipes. Pipes 1 to 9 can be assigned any endpoint number based on peripheral devices used for communication or based on the user system.
Table 32.1 shows the specifications of the USB.

Table 32.1 USB Specifications

Item	Specifications
Features	• USB Device Controller (UDC) and transceiver for USB 2.0 are incorporated. Host controller, function controller, and On-The-Go (OTG) are supported (one channel) • The host controller and the function controller can be switched by software. • Self-power mode or bus power mode can be selected. • BC1.2 (Battery Charging Specification Revision 1.2) is supported.
	When the host controller is selected: • Full-speed transfer (12 Mbps) and low-speed transfer (1.5 Mbps) are supported • Automatic scheduling for SOF and packet transmissions • Programmable intervals for isochronous and interrupt transfers
	When the function controller is selected: • Full-speed transfer (12 Mbps) and low-speed transfer (1.5 Mbps) are supported • Control transfer stage control function • Device state control function • Auto response function for SET_ADDRESS request • SOF interpolation function
Communication data transfer type	• Control transfer • Bulk transfer • Interrupt transfer • Isochronous transfer
Pipe configuration	• Buffer memory for USB communication is provided. • Up to 10 pipes can be selected (including the default control pipe). • Pipes 1 to 9 can be assigned any endpoint number.
	Transfer conditions that can be set for each pipe: • Pipe 0: Control transfer, 64-byte single buffer • Pipes 1 and 2: 64-byte double buffer can be specified for bulk transfer and 256-byte double buffer for isochronous transfer • Pipes 3 to 5: Bulk transfer, 64-byte double buffer • Pipes 6 to 9: Interrupt transfer, 64-byte single buffer
Others	• Reception ending function using transaction count • Function that changes the BRDY interrupt event notification timing (BFRE) • Function that automatically clears the buffer memory after the data for the pipe specified at the DnFIFO (n = 0, 1) port has been read (DCLRM) • NAK setting function for response PID generated by end of transfer (SHTNAK) • On-chip pull-up and pull-down resistors of D+/D–
Low power consumption function	Module-stop state can be set.

Downloaded from Arrow.com.

Figure 32.1 shows a block diagram of the USB.

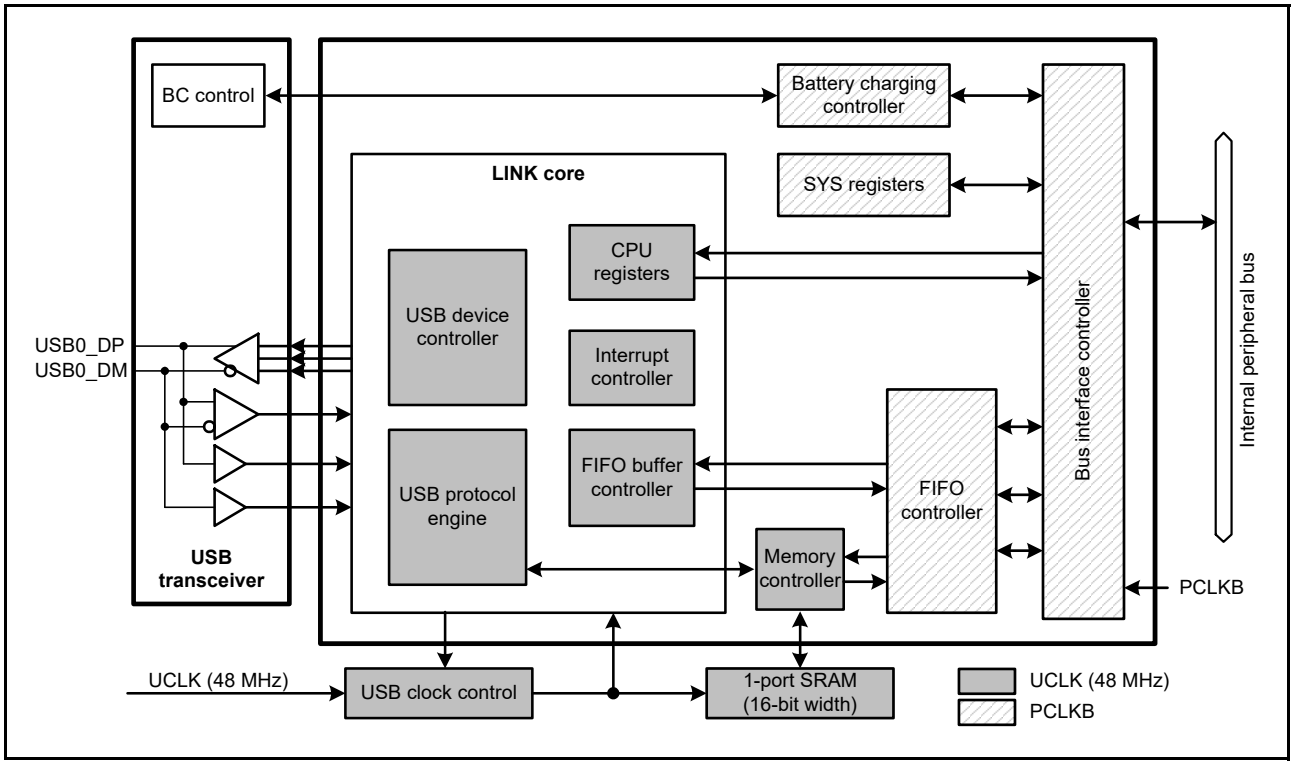


Figure 32.1 USB Block Diagram

Table 32.2 lists the I/O pins of the USB.

Table 32.2 USB Pin Configuration

Port	Pin Name	I/O	Function
USB0	USB0_DP	I/O	D+ I/O pin of the USB on-chip transceiver This pin should be connected to the D+ pin of the USB bus.
	USB0_DM	I/O	D– I/O pin of the USB on-chip transceiver This pin should be connected to the D– pin of the USB bus.
	USB0_VBUS	Input	USB cable connection monitor pin This pin should be connected to VBUS of the USB bus. The VBUS pin status (connected or disconnected) can be detected when the USB module is operating as a function controller.
	USB0_EXICEN	Output	Low-power control signal for external power supply (OTG) chip
	USB0_VBUSEN	Output	VBUS (5 V) supply enable signal for external power supply chip
	USB0_OVRCURA USB0_OVRCURB	Input	External overcurrent detection signals should be connected to these pins. VBUS comparator signals should be connected to these pins when the OTG power supply chip is connected.
	USB0_ID	Input	miniAB connector ID input signal should be connected to this pin during operation in OTG mode.
Common	VCC_USB	Input	USB power supply pin
	VSS_USB	Input	USB ground pin

Downloaded from Arrow.com.

32.2 Register Descriptions

32.2.1 System Configuration Control Register (SYSCFG)

Address(es): USB0.SYSCFG 000A 0000h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	SCKE	—	CNEN	—	DCFM	DRPD	DPRPU	DMRPU	—	—	USBE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	USBE	USB Operation Enable	0: USB operation is disabled. 1: USB operation is enabled.	R/W
b2, b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b3	DMRPU	D– Line Resistor Control*1	0: Pulling up the line is disabled. 1: Pulling up the line is enabled.	R/W
b4	DPRPU	D+ Line Resistor Control*1	0: Pulling up the line is disabled. 1: Pulling up the line is enabled.	R/W
b5	DRPD	D+/D– Line Resistor Control	0: Pulling down the lines is disabled. 1: Pulling down the lines is enabled.	R/W
b6	DCFM	Controller Function Select	0: Function controller is selected. 1: Host controller is selected.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b8	CNEN	CNEN Single End Receiver Enable	0: Single end receiver operation is disabled. 1: Single end receiver operation is enabled.	R/W
b9	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b10	SCKE	USB Clock Enable*2	0: Stops supplying the clock signal to the USB. 1: Enables supplying the clock signal to the USB.	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Do not enable the DMRPU and DPRPU bits at the same time.
Note 2. After writing 1 to the SCKE bit, read it and confirm it is set to 1.

USBE Bit (USB Operation Enable)

The USBE bit enables or disables operation of the USB.

Modifying the USBE bit from 1 to 0 initializes the register bits listed in Table 32.3.

This bit should be modified while the SCKE bit is 1.

When the host controller is selected, this bit should be set to 1 after setting the DRPD bit to 1, eliminating SYSSTS0.LNST[1:0] flag chattering, and confirming that the USB bus state is stabilized.

Table 32.3 Registers Initialized by Writing 0 to the SYSCFG.USBE Bit

Selected Function	Register	Bit	Remarks
Function controller	SYSSTS0	LNST[1:0]	The value is retained when the host controller is selected.
	DVSTCTR0	RHST[2:0]	
	INTSTS0	DVSQ[2:0]	The value is retained when the host controller is selected.
	USBREQ	BREQUEST[7:0], BMREQUESTTYPE[7:0]	The value is retained when the host controller is selected.
	USBVAL	—	The value is retained when the host controller is selected.
	USBINDX	—	The value is retained when the host controller is selected.
	USBLENG	—	The value is retained when the host controller is selected.
Host controller	DVSTCTR0	RHST[2:0]	
	FRMNUM	FRNM[10:0]	The value is retained when the function controller is selected.

Downloaded from Arrow.com.

DMRPU Bit (D– Line Resistor Control)

The DMRPU bit enables or disables pulling up the D– line when the function controller is selected.
When the DMRPU bit is set to 1 while the function controller is selected, the bit forces a pull-up of the D– line to notify the USB host of connection as a low-speed device. Modifying the DMRPU bit from 1 to 0 allows the USB to release the D– line, thus notifying the USB host of disconnection.
This bit should be set to 1 if the function controller is selected, and should be set to 0 if the host controller is selected.

DPRPU Bit (D+ Line Resistor Control)

The DPRPU bit enables or disables pulling up the D+ line when the function controller is selected.
When the DPRPU bit is set to 1 while the function controller is selected, the bit forces a pull-up of the D+ line to notify the USB host of connection as a full-speed device. Modifying the DPRPU bit from 1 to 0 allows the USB to release the D+ line, thus notifying the USB host of disconnection.
This bit should be set to 1 if the function controller is selected, and should be set to 0 if the host controller is selected.

DRPD Bit (D+/D– Line Resistor Control)

The DRPD bit enables or disables pulling down D+ and D– lines when the host controller is selected.
This bit should be set to 1 if the host controller is selected, and should be set to 0 if the function controller is selected.

DCFM Bit (Controller Function Select)

The DCFM bit selects the function of the USB.
This bit should be modified when the DMRPU, DPRPU, and DRPD bits are all 0.

CNEN Bit (CNEN Single End Receiver Enable)

Setting the CNEN bit to 1 allows the USB module to enable the single end receiver and set the LNST[1:0] flags to monitor the status of the D+ and D– lines.
The CNEN bit is used when the USB module operates as a portable device for battery charging.

SCKE Bit (USB Clock Enable)

The SCKE bit stops or enables supplying 48-MHz clock signals to the USB.
When this bit is 0, only SYSCFG can be read from and written to; the other registers related to the USB cannot be read from or written to.

Downloaded from Arrow.com.

32.2.2 System Configuration Status Register 0 (SYSSTS0)

Address(es): USB0.SYSSTS0 000A 0004h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	OVCMON[1:0]		—	—	—	—	—	—	—	HTACT	—	—	—	IDMON	LNST[1:0]	
Value after reset:	0*1	0*1	0	0	0	0	0	0	0	0	0	0	0	0*1	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	LNST[1:0]	USB Data Line Status Monitor Flag	Refer to Table 32.4.	R
b2	IDMON	ID Input Pin Monitor Flag	0: USB0_ID pin is low 1: USB0_ID pin is high	R
b5 to b3	—	Reserved	These bits are read as 0 and cannot be modified.	R
b6	HTACT	USB Host Sequencer Status Monitor Flag	0: Host sequencer of the USB is completely stopped. 1: Host sequencer of the USB is not completely stopped.	R
b13 to b7	—	Reserved	These bits are read as 0 and cannot be modified.	R
b15, b14	OVCMON[1:0]	OVRCURA/OVRCURB Input Pin Monitor Flag	The OVCMON[1] flag indicates the status of the USB0_OVRCURA pin. The OVCMON[0] flag indicates the status of the USB0_OVRCURB pin.	R

Note 1. Depends on the status of the USB0_OVRCURA/USB0_OVRCURB and USB0_ID pins.

LNST[1:0] Flags (USB Data Line Status Monitor Flag)

The LNST[1:0] flags indicate the state of the USB data lines (D+ and D– lines). Refer to Table 32.4.

The LNST[1:0] flags should be read after the connection processing (SYSCFG.DPRPU bit = 1) when the function controller is selected; whereas after enabling pull-down of the lines (SYSCFG.DRPD bit = 1) when the host controller is selected.

HTACT Flag (USB Host Sequencer Status Monitor Flag)

The HTACT flag is 0 when the host sequencer of the USB is completely stopped.

OVCMON[1:0] Flags (OVRCURA/OVRCURB Input Pin Monitor Flag)

The OVCMON[1:0] flags indicate the status of overcurrent from an external power supply chip.

Table 32.4 Status of USB Data Bus Lines (D+ Line, D– Line)

LNST[1:0] Flags	During Low-Speed Operation	During Full-Speed Operation
00b	SE0	SE0
01b	K-State	J-State
10b	J-State	K-State
11b	SE1	SE1

Downloaded from Arrow.com.

32.2.3 Device State Control Register 0 (DVSTCTR0)

Address(es): USB0.DVSTCTR0 000A 0008h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	HNPBTOA	EXICEN	VBUSEN	WKUP	RWUPE	USBRST	RESUME	UACT	—	RHST[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	RHST[2:0]	USB Bus Reset Status Flag	• When the host controller is selected b2 b0 0 0 0: Communication speed not determined (powered state or no connection) 1 x x: USB bus reset in progress 0 0 1: Low-speed connection 0 1 0: Full-speed connection • When the function controller is selected b2 b0 0 0 0: Communication speed not determined 0 0 1: USB bus reset in progress or low-speed connection 0 1 0: USB bus reset in progress or full-speed connection	R
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	UACT	USB Bus Enable	0: Downstream port is disabled (SOF transmission is disabled). 1: Downstream port is enabled (SOF transmission is enabled).	R/W
b5	RESUME	Resume Output	0: Resume signal is not output. 1: Resume signal is output.	R/W
b6	USBRST	USB Bus Reset Output	0: USB bus reset signal is not output. 1: USB bus reset signal is output.	R/W
b7	RWUPE	Wakeup Detection Enable	0: Downstream port wakeup is disabled. 1: Downstream port wakeup is enabled.	R/W
b8	WKUP	Wakeup Output	0: Remote wakeup signal is not output. 1: Remote wakeup signal is output.	R/W
b9	VBUSEN	VBUSEN Output Pin Control	0: USB0_VBUSEN pin outputs low. 1: USB0_VBUSEN pin outputs high.	R/W
b10	EXICEN	EXICEN Output Pin Control	0: USB0_EXICEN pin outputs low. 1: USB0_EXICEN pin outputs high.	R/W
b11	HNPBTOA	Host Negotiation Protocol (HNP) Control	This bit is used when switching from device B to device A while in OTG mode. If the HNPBTOA bit is 1, the internal function control keeps the suspended state until the HNP processing ends even though SYSCFG.DPRPU = 0 or SYSCFG.DCFM = 1.	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

x: Don’t care

Downloaded from Arrow.com.

RHST[2:0] Flags (USB Bus Reset Status Flag)

The RHST[2:0] flags indicate the status of the USB bus reset.
When the host controller is selected, the RHST[2:0] flags indicate 100b after the USBRST bit has been set to 1 by software.
The USB fixes the value of the RHST[2:0] flags when 0 is written to the USBRST bit by software and the USB completes SE0 driving.
When the function controller is selected, the RHST[2:0] flags indicate 010b (connection while DPRPU = 1) or 001b (disconnection while DMRPU = 1) when the USB detects the USB bus reset, and a DVST interrupt is generated.

UACT Bit (USB Bus Enable)

The UACT bit enables operation of the USB bus (controls the SOF packet transmission to the USB bus) when the host controller is selected.
With this bit set to 1, the USB puts the USB port to the USB-bus enabled state and performs SOF output and data transmission and reception.
This module starts outputting SOF packets within one frame after 1 has been written to the UACT bit by software.
With this bit set to 0, the USB enters the idle state after outputting SOF packets.
The USB sets the UACT bit to 0 on any of the following conditions.

- A DTCH interrupt is detected during communication (while UACT = 1).
- An EOFERR interrupt is detected during communication (while UACT = 1).

Writing 1 to this bit should be done at the end of the USB reset processing (writing 0 to the USBRST bit) or at the end of the resume processing from the suspended state (writing 0 to the RESUME bit).
This bit should be set to 0 if the function controller is selected.

RESUME Bit (Resume Output)

The RESUME bit controls the resume signal output when the host controller is selected.
Setting the RESUME bit to 1 allows the USB to drive the port to the K-state and output the resume signal.
The USB sets the RESUME bit to 1 on detecting the remote wakeup signal while RWUPE is 1 in the USB suspended state.
The USB continues outputting K-state while the RESUME bit = 1 (until the RESUME bit is set to 0 by software). The RESUME bit should be 1 (= resume period) for the time defined by USB Specification 2.0.
This bit should be set to 1 in the suspended state.
Write 1 to the UACT bit simultaneously with the end of the resume processing (writing 0 to the RESUME bit).
This bit should be set to 0 if the function controller is selected.

USBRST Bit (USB Bus Reset Output)

The USBRST bit controls the USB bus reset signal output when the host controller is selected.
When the host controller is selected, setting this bit to 1 allows the USB to drive SE0 of the USB port to reset the USB bus.
The USB continues outputting SE0 while USBRST = 1 (until the USBRST bit is set to 1 by software). The USBRST bit should be 1 (= USB bus reset period) for the time defined by USB Specification 2.0.
Writing 1 to this bit during communication (the UACT bit = 1) or during the resume processing (the RESUME bit = 1) prevents the USB from starting the USB bus reset processing until both the UACT and RESUME bits become 0.
Write 1 to the UACT bit simultaneously with the end of the USB bus reset processing (writing 0 to the USBRST bit).
This bit should be set to 0 if the function controller is selected.

Downloaded from Arrow.com.

RWUPE Bit (Wakeup Detection Enable)

The RWUPE bit enables or disables the downstream port peripheral device to use the remote wakeup function (resume signal output) when the host controller is selected.
With this bit set to 1, on detecting the remote wakeup signal, the USB detects the resume signal (K-state for 2.5 µs) from the downstream port device and performs the resume processing (drives the port to the K-state).
With this bit set to 0, the USB ignores the detected remote wakeup signal (K-state) from the peripheral device connected to the USB port.
While the RWUPE bit is 1, the internal clock should not be stopped even in the suspended state (SYSCFG.SCKE bit should be set to 1).
This bit should be set to 0 if the function controller is selected.

WKUP Bit (Wakeup Output)

The WKUP bit enables or disables outputting the remote wakeup signal (resume signal) to the USB bus when the function controller is selected.
The USB controls the output time of a remote wakeup signal. When this bit is set to 1, the USB sets this bit to 0 after outputting the 10-ms K-state.
According to USB Specification 2.0, the USB bus idle state must be kept for 5 ms or longer before a remote wakeup signal is sent. If the USB writes 1 to this bit right after detection of the suspended state, the K-state will be output after 2 ms.
Do not write 1 to this bit, unless the device state is in the suspended state (INTSTS0.DVSQ[2:0] flags = 1xxb) and the USB host enables the remote wakeup signal. When this bit is set to 1, the internal clock must not be stopped even in the suspended state (write 1 to this bit while the SYSCFG.SCKE bit = 1).
This bit should be set to 0 if the host controller is selected.

HNPBTOA Bit (Host Negotiation Protocol (HNP) Control)

The HNPBTOA bit is used when switching from device B to device A while in OTG mode. If the HNPBTOA bit is 1, the internal function control keeps the suspended state until the HNP processing ends even though the SYSCFG.DPRPU bit = 0 or SYSCFG.DCFM = 1 is set. Even if the falling edge of the D+ signal is detected at this time, no resume (RESM) interrupt is generated.
After this bit is set to 1, write 0 to this bit by software to terminate the HNP processing when connection to the host (pull-up on the target side) or timeout of the HNP processing is detected.

Downloaded from Arrow.com.

32.2.4 CFIFO Port Register (CFIFO), D0FIFO Port Register (D0FIFO), D1FIFO Port Register (D1FIFO)

(1) When the MBW bit is 1

Address(es): USB0.CFIFO 000A 0014h, USB0.D0FIFO 000A 0018h, USB0.D1FIFO 000A 001Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	—	FIFO Port	This port is used for reading receive data from the FIFO buffer and writing transmit data to the FIFO buffer.	R/W

(2) When the MBW bit is 0

Address(es): USB0.CFIFO.L 000A 0014h, USB0.D0FIFO.L 000A 0018h, USB0.D1FIFO.L 000A 001Ch

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	FIFO Port	This port is used for reading receive data from the FIFO buffer and writing transmit data to the FIFO buffer.	R/W

There are three FIFO ports: CFIFO, D0FIFO, and D1FIFO. Each FIFO port is configured of a port register (CFIFO, D0FIFO, or D1FIFO) that handles reading of data from the FIFO buffer memory and writing of data to the FIFO buffer memory, a port select register (CFIFOSEL, D0FIFOSEL, or D1FIFOSEL) that is used to select the pipe assigned to the FIFO port, and a port control register (CFIFOCTR, D0FIFOCTR, or D1FIFOCTR).
Each FIFO port has the following precautions.

- The FIFO buffer for DCP (control transfer) should be accessed through the CFIFO port.
- Accessing a FIFO buffer using DMA/DTC transfer should be performed through the D0FIFO or D1FIFO port.
- The D0FIFO and D1FIFO ports can also be accessed by the CPU.
- When using functions specific to a FIFO port, the pipe number (selected pipe) specified by the CURPIPE[3:0] bits in the port select register cannot be changed (when the DMA/DTC transfer function is used, etc.).
- The same pipe should not be assigned to two or more FIFO ports.
- There are two FIFO buffer states: the access right for a FIFO buffer can be on the CPU side or on the Serial Interface Engine (SIE) side. When the access right for a FIFO buffer is on the SIE side, the FIFO buffer cannot be accessed from the CPU.

Downloaded from Arrow.com.

FIFO Port Bit

Accessing the FIFO port bits allows reading the received data from the FIFO buffer or writing the transmit data to the FIFO buffer.

Each FIFO port register can be accessed only while the FRDY flag in each FIFO port control register (CFIFOCTR, D0FIFOCTR, or D1FIFOCTR) is 1.

The valid bits in a FIFO port register depend on the settings of the corresponding MBW bit of the FIFO port select register (CFIFOSEL, D0FIFOSEL, or D1FIFOSEL).

When the MBW bit is 1 (16-bit width), the data arrangement may differ from the data arrangement on the RAM depending on the value of the MDE.MDE[2:0] bits and the setting of the BIGEND bit (CFIFOSEL.BIGEND, D0FIFOSEL.BIGEND, or D1FIFOSEL.BIGEND). Table 32.5 lists the endian operation in 16-bit access. Note that if the total number of transmit data bytes is odd, access the lower byte in bytes when writing the last data.

When the MBW bit is 0 (8-bit width), access the lower byte in bytes.

Table 32.5 Endian Operation in 16-Bit Access

MDE.MDE[2:0] bits	CFIFOSEL.BIGEND Bit D0FIFOSEL.BIGEND Bit D1FIFOSEL.BIGEND Bit	Bits 15 to 8	Bits 7 to 0	Remarks
000b (big endian)	0 (little endian)	Data in address N + 1	Data in address N	Bytes reversed
	1 (big endian)	Data in address N	Data in address N + 1	
111b (little endian)	0 (little endian)	Data in address N + 1	Data in address N	
	1 (big endian)	Data in address N	Data in address N + 1	Bytes reversed

Downloaded from Arrow.com.

32.2.5 CFIFO Port Select Register (CFIFOSEL), D0FIFO Port Select Register (D0FIFOSEL), D1FIFO Port Select Register (D1FIFOSEL)

- CFIFOSEL

Address(es): USB0.CFIFOSEL 000A 0020h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RCNT	REW	—	—	—	MBW	—	BIGEND	—	—	ISEL	—	CURPIPE[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	CURPIPE [3:0]	CFIFO Port Access Pipe Specification	b3 b0 0 0 0 0: DCP (Default control pipe) 0 0 0 1: Pipe 1 0 0 1 0: Pipe 2 0 0 1 1: Pipe 3 0 1 0 0: Pipe 4 0 1 0 1: Pipe 5 0 1 1 0: Pipe 6 0 1 1 1: Pipe 7 1 0 0 0: Pipe 8 1 0 0 1: Pipe 9 Settings other than above are prohibited.	R/W
b4	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5	ISEL	CFIFO Port Access Direction When DCP is Selected	0: Reading from the buffer memory is selected. 1: Writing to the buffer memory is selected.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	BIGEND	CFIFO Port Endian Control	0: Little endian 1: Big endian	R/W
b9	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b10	MBW	CFIFO Port Access Bit Width	0: 8-bit width 1: 16-bit width	R/W
b13 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b14	REW	Buffer Pointer Rewind	0: The buffer pointer is not rewound. 1: The buffer pointer is rewound.	R/W*1
b15	RCNT	Read Count Mode	0: The CFIFOCTR.DTLN[8:0] flags are set to 000h when all of the receive data has been read from the CFIFO. (In double buffer mode, the DTLN[8:0] flag value is set to 000h when all the data has been read from only a single plane.) 1: The CFIFOCTR.DTLN[8:0] flags are decremented each time the receive data is read from the CFIFO.	R/W

Note 1. Only 0 can be read.

The same pipe should not be specified by the CURPIPE[3:0] bits in the CFIFOSEL, D0FIFOSEL, and D1FIFOSEL registers. When the CURPIPE[3:0] bits in the D0FIFOSEL and D1FIFOSEL registers are set to 0000b, no pipe is selected.

The pipe number should not be changed while the DMA/DTC transfer is enabled.

Downloaded from Arrow.com.

CURPIPE[3:0] Bits (CFIFO Port Access Pipe Specification)

The CURPIPE[3:0] bits specify the pipe number using which data is read or written through the CFIFO port.
After writing to these bits, read these bits to check that the written value agrees with the read value before proceeding to the next process.
Do not set the same pipe number to the CURPIPE[3:0] bits in the CFIFOSEL, D0FIFOSEL, and D1FIFOSEL registers.
Even if an attempt is made to modify the setting of these bits during access to the FIFO buffer, the current access setting is retained until the access is completed. Then, the modification becomes effective, thus enabling continuous access.

ISEL Bit (CFIFO Port Access Direction When DCP is Selected)

After writing to the ISEL bit with the DCP being a selected pipe, read this bit to check that the written value agrees with the read value before proceeding to the next process.
Set this bit and the CURPIPE[3:0] bits simultaneously.

MBW Bit (CFIFO Port Access Bit Width)

The MBW bit specifies the bit width for accessing the CFIFO port.
When the selected pipe is in the receiving direction, once reading data is started after setting this bit, this bit should not be modified until all the data has been read.
When the selected pipe is in the receiving direction, set the CURPIPE[3:0] bits to a different value once, and then set these bits and the MBW bit simultaneously. For the procedure for modifying the CURPIPE[3:0] bits, follow the description of these bits.
When the selected pipe is in the transmitting direction, the bit width cannot be changed from 8-bit width to 16-bit width while data is being written to the buffer memory.
An odd number of bytes can also be written through byte-access control even when 16-bit width is selected.

REW Bit (Buffer Pointer Rewind)

The REW bit specifies whether or not to rewind the buffer pointer.
When the selected pipe is in the receiving direction, setting the REW bit to 1 while the FIFO buffer is being read allows re-reading the FIFO buffer from the first data (in double buffer mode, re-reading the currently-read FIFO buffer plane from the first data is allowed).
Do not set the REW bit to 1 simultaneously with modifying the CURPIPE[3:0] bits. Before setting the REW bit to 1, be sure to check that the FRDY flag is 1.
To re-write to the FIFO buffer again from the first data for the pipe in the transmitting direction, use the BCLR bit.

Downloaded from Arrow.com.

- D0FIFOSEL, D1FIFOSEL

Address(es): USB0.D0FIFOSEL 000A 0028h, USB0.D1FIFOSEL 000A 002Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RCNT	REW	DCLRM	DREQE	—	MBW	—	BIGEND	—	—	—	—	CURPIPE[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	CURPIPE[3:0]	FIFO Port Access Pipe Select	b3 b0 0 0 0 0: DCP (Default control pipe) 0 0 0 1: Pipe 1 0 0 1 0: Pipe 2 0 0 1 1: Pipe 3 0 1 0 0: Pipe 4 0 1 0 1: Pipe 5 0 1 1 0: Pipe 6 0 1 1 1: Pipe 7 1 0 0 0: Pipe 8 1 0 0 1: Pipe 9 Settings other than above are prohibited.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	BIGEND	FIFO Port Endian Control	0: Little endian 1: Big endian	R/W
b9	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b10	MBW	FIFO Port Access Bit Width	0: 8-bit width 1: 16-bit width	R/W
b11	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b12	DREQE	DMA/DTC Transfer Request Enable	0: DMA/DTC transfer request is disabled. 1: DMA/DTC transfer request is enabled.	R/W
b13	DCLRM	Automatic FIFO Buffer Memory Clear Mode after Selected Pipe is Read	0: Automatic buffer clear mode is disabled. 1: Automatic buffer clear mode is enabled.	R/W
b14	REW	Buffer Pointer Rewind	0: The buffer pointer is not rewound. 1: The buffer pointer is rewound.	R/W*1
b15	RCNT	Read Count Mode	0: The DnFIFOCTR.DTLN[8:0] flags are set to 000h when all of the receive data has been read from the DnFIFO. (In double buffer mode, the DTLN[8:0] flag value is set to 000h when all the data has been read from only a single plane.) 1: The DnFIFOCTR.DTLN[8:0] flags are decremented each time the receive data is read from the DnFIFO. (n = 0, 1)	R/W

Note 1. Only 0 can be read.

The same pipe should not be specified by the CURPIPE[3:0] bits in the CFIFOSEL, D0FIFOSEL, and D1FIFOSEL registers. When the CURPIPE[3:0] bits in the D0FIFOSEL and D1FIFOSEL registers are set to 0000b, no pipe is selected.

The pipe number should not be changed while the DMA/DTC transfer is enabled.

Downloaded from Arrow.com.

CURPIPE[3:0] Bits (FIFO Port Access Pipe Select)

The CURPIPE[3:0] bits specify the pipe number using which data is read or written through the D0FIFO port or D1FIFO port.
After writing to these bits, read these bits to check that the written value agrees with the read value before proceeding to the next process.
Do not set the same pipe number to the CURPIPE[3:0] bits in the CFIFOSEL, D0FIFOSEL, and D1FIFOSEL registers.
Even if an attempt is made to modify the setting of these bits during access to the FIFO buffer, the current access setting is retained until the access is completed. Then, the modification becomes effective, thus enabling continuous access.

MBW Bit (FIFO Port Access Bit Width)

The MBW bit specifies the bit width for accessing the D0FIFO port or D1FIFO port.
When the selected pipe is in the receiving direction, once reading data is started after setting this bit, this bit should not be modified until all the data has been read.
When the selected pipe is in the receiving direction, set the CURPIPE[3:0] bits to a different value once, and then set these bits and the MBW bit simultaneously. For the procedure for modifying the CURPIPE[3:0] bits, follow the description of these bits.
When the selected pipe is in the transmitting direction, the bit width cannot be changed from 8-bit width to 16-bit width while data is being written to the buffer memory.
An odd number of bytes can also be written through byte-access control even when 16-bit width is selected.

DREQE Bit (DMA/DTC Transfer Request Enable)

The DREQE bit enables or disables the DMA/DTC transfer request to be issued.
Before setting the DREQE bit to 1 to enable the DMA/DTC transfer request to be issued, set the CURPIPE[3:0] bits.
When modifying the setting of the CURPIPE[3:0] bits, set this bit to 0 first.

DCLRM Bit (Automatic FIFO Buffer Memory Clear Mode after Selected Pipe is Read)

The DCLRM bit enables or disables the buffer memory to be cleared automatically after data in the selected pipe has been read.
With this bit set to 1, the USB sets the BCLR bit to 1 for the FIFO buffer of the selected pipe on receiving a zero-length packet while the FIFO buffer assigned to the selected pipe is empty, or on receiving a short packet and reading the data while the PIPECFG.BFRE bit is 1.
When using the USB with the SOFCFG.BRDYM bit set to 1, set this bit to 0.

REW Bit (Buffer Pointer Rewind)

The REW bit specifies whether or not to rewind the buffer pointer.
When the selected pipe is in the receiving direction, setting the REW bit to 1 while the FIFO buffer is being read allows re-reading the FIFO buffer from the first data (in double buffer mode, re-reading the currently-read FIFO buffer plane from the first data is allowed).
Do not set the REW bit to 1 simultaneously with modifying the CURPIPE[3:0] bits. Before setting the REW bit to 1, be sure to check that the FRDY flag is 1.
To re-write to the FIFO buffer again from the first data for the pipe in the transmitting direction, use the BCLR bit.

RCNT Bit (Read Count Mode)

The RCNT bit specifies the read mode for the value in the DnFIFOCTR.DTLN[8:0] flags.
When accessing DnFIFO with the PIPECFG.BFRE bit set to 1, set the RCNT bit to 0.

Downloaded from Arrow.com.

32.2.6 CFIFO Port Control Register (CFIFOCTR), D0FIFO Port Control Register (D0FIFOCTR), D1FIFO Port Control Register (D1FIFOCTR)

Address(es): USB0.CFIFOCTR 000A 0022h, USB0.D0FIFOCTR 000A 002Ah, USB0.D1FIFOCTR 000A 002Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	BVAL	BCLR	FRDY	—	—	—	—	DTLN[8:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b8 to b0	DTLN[8:0]	Receive Data Length Flag	Indicate the length of the receive data. These bits indicate different values depending on the setting of the RCNT bit in the port select register. For details, refer to the description on the DTLN[8:0] flags shown below.	R
b12 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b13	FRDY	FIFO Port Ready Flag	0: FIFO port access is disabled. 1: FIFO port access is enabled.	R
b14	BCLR	CPU Buffer Clear	0: Does not operate. 1: Clears the buffer memory on the CPU side.	R/W*1
b15	BVAL	Buffer Memory Valid	0: Invalid 1: Writing ended	R/W

Note 1. Only 0 can be read.

Registers CFIFOCTR, D0FIFOCTR, and D1FIFOCTR correspond to CFIFO, D0FIFO, and D1FIFO, respectively.

DTLN[8:0] Flags (Receive Data Length Flag)

The DTLN[8:0] flags indicate the length of the receive data.

While the FIFO buffer is being read, the DTLN[8:0] flags indicate different values depending on the DnFIFOSEL.RCNT bit (n = 0, 1) value as described below.

- RCNT = 0

  The USB sets the DTLN[8:0] flags to indicate the length of the receive data until the CPU or DMAC/DTC has read all the received data from a single FIFO buffer plane.

  While the PIPECFG.BFRE bit = 1, these bits retain the length of the receive data until the BCLR bit is set to 1 even after all the data has been read.

- RCNT = 1

  The USB decrements the value indicated by the DTLN[8:0] flags each time data is read from the FIFO buffer. (The value is decremented by one when the MBW bit is 0, and by two when the MBW bit is 1.)

  The USB sets these bits to 0 when all the data has been read from one FIFO buffer plane. However, in double buffer mode, if data has been received in one FIFO buffer plane before all the data has been read from the other plane, the USB sets these bits to indicate the length of the receive data in the former plane when all the data has been read from the latter plane.

Downloaded from Arrow.com.

FRDY Flag (FIFO Port Ready Flag)

The FRDY flag indicates whether the FIFO port can be accessed by the CPU or DMAC/DTC.
In the following cases, the USB sets the FRDY flag to 1 but data cannot be read via the FIFO port because there is no data to be read. In these cases, set the BCLR bit to 1 to clear the FIFO buffer, and enable transmission and reception of the next data.

- A zero-length packet is received when the FIFO buffer assigned to the selected pipe is empty.
- A short packet is received and the data is completely read while the PIPECFG.BFRE bit = 1.

BCLR Bit (CPU Buffer Clear)

The BCLR bit should be set to 1 to clear the FIFO buffer on the CPU side for the selected pipe.
When double buffer mode is set for the FIFO buffer assigned to the selected pipe, the USB clears only the FIFO buffer on the CPU side even when both planes are read-enabled.
When the selected pipe is the DCP, setting the BCLR bit to 1 allows the USB to clear the FIFO buffer regardless of whether the FIFO buffer is on the CPU side or SIE side. To clear the buffer on the SIE side, set the DCPCTR.PID[1:0] bits for the DCP to 00b (NAK) before setting the BCLR bit to 1.
When the selected pipe is in the transmitting direction, if 1 is written to the BVAL bit and the BCLR bit simultaneously, the USB clears the data that has been written before it, enabling transmission of a zero-length packet.
When the selected pipe is not the DCP, writing 1 to the BCLR bit should be done while the FRDY flag in the FIFO port control register is 1 (set by the USB).

BVAL Bit (Buffer Memory Valid)

The BVAL bit should be set to 1 when data has been completely written to the FIFO buffer on the CPU side for the pipe selected using the CURPIPE[3:0] bits (selected pipe).
When the selected pipe is in the transmitting direction, set the BVAL bit to 1 in the following cases. Then, the USB switches the FIFO buffer from the CPU side to the SIE side, thus enabling transmission.

- To transmit a short packet, set the BVAL bit to 1 after data has been written.
- To transmit a zero-length packet, set the BVAL bit to 1 before data is written to the FIFO buffer.

When data of the maximum packet size has been written for the pipe in continuous transfer mode, the USB sets the BVAL bit to 1 and switches the FIFO buffer from the CPU side to the SIE side, thus enabling transmission.
Writing 1 to the BVAL bit should be done while the FRDY flag is 1 (set by the USB).
When the selected pipe is in the receiving direction, do not set the BVAL bit to 1.

Downloaded from Arrow.com.

32.2.7 Interrupt Enable Register 0 (INTENB0)

Address(es): USB0.INTENB0 000A 0030h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	VBSE	RSME	SOFE	DVSE	CTRE	BEMPE	NRDYE	BRDYE	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	BRDYE	Buffer Ready Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b9	NRDYE	Buffer Not Ready Response Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b10	BEMPE	Buffer Empty Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b11	CTRE	Control Transfer Stage Transition Interrupt Enable *1	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b12	DVSE	Device State Transition Interrupt Enable*1	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b13	SOFE	Frame Number Update Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b14	RSME	Resume Interrupt Enable*1	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b15	VBSE	VBUS Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W

Note 1. The RSME, DVSE, and CTRE bits can be set to 1 only when the function controller is selected. Set these bits to 0 when the host controller is selected.

On detecting the interrupt corresponding to the bit in the INTENB0 register to which 1 has been set by software, the USB generates the USB interrupt request.

The USB sets 1 to each status bit in the INTSTS0 register when a detection condition of the corresponding interrupt source has been satisfied regardless of the setting in the INTENB0 register (regardless of whether the interrupt output is enabled or disabled).

While the status bit in the INTSTS0 register corresponding to the interrupt source indicates 1, the USB generates the USB interrupt request when the corresponding interrupt enable bit in the INTENB0 register is modified from 0 to 1 by software.

Downloaded from Arrow.com.

32.2.8 Interrupt Enable Register 1 (INTENB1)

Address(es): USB0.INTENB1 000A 0032h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	OVRCRE	BCHGE	—	DTCHE	ATTCHE	—	—	—	—	EOFERRE	SIGNE	SACKE	—	—	—	PDDETINTE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PDDETINTE0	Portable Device Detection Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	SACKE	Setup Transaction Normal Response Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b5	SIGNE	Setup Transaction Error Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b6	EOFERRE	EOF Error Detection Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b10 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11	ATTCHE	Connection Detection Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b12	DTCHE	Disconnection Detection Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b13	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b14	BCHGE	USB Bus Change Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b15	OVRCRE	Overcurrent Input Change Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W

Note: The bits in the INTENB1 register can be set to 1 only when the host controller is selected. Set these bits to 0 when the function controller is selected.

The INTENB1 register specifies the interrupt masks when the host controller is selected, and for the setup transaction. On detecting the interrupt corresponding to the bit in the INTENB1 register to which 1 has been set by software, the USB generates the USB interrupt request.

The USB sets 1 to each status bit in the INTSTS1 register when a detection condition of the corresponding interrupt source has been satisfied regardless of the setting in the INTENB1 register (regardless of whether the interrupt output is enabled or disabled).

While the status bit in the INTSTS1 register corresponding to the interrupt source indicates 1, the USB generates the USB interrupt request when the corresponding interrupt enable bit in the INTENB1 register is modified from 0 to 1 by software.

When the function controller is selected, the interrupts should not be enabled.

Downloaded from Arrow.com.

32.2.9 BRDY Interrupt Enable Register (BRDYENB)

Address(es): USB0.BRDYENB 000A 0036h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PIPE9BRDYE	PIPE8BRDYE	PIPE7BRDYE	PIPE6BRDYE	PIPE5BRDYE	PIPE4BRDYE	PIPE3BRDYE	PIPE2BRDYE	PIPE1BRDYE	PIPE0BRDYE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PIPE0BRDYE	BRDY Interrupt Enable for Pipe 0	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b1	PIPE1BRDYE	BRDY Interrupt Enable for Pipe 1	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b2	PIPE2BRDYE	BRDY Interrupt Enable for Pipe 2	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b3	PIPE3BRDYE	BRDY Interrupt Enable for Pipe 3	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b4	PIPE4BRDYE	BRDY Interrupt Enable for Pipe 4	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b5	PIPE5BRDYE	BRDY Interrupt Enable for Pipe 5	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b6	PIPE6BRDYE	BRDY Interrupt Enable for Pipe 6	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b7	PIPE7BRDYE	BRDY Interrupt Enable for Pipe 7	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b8	PIPE8BRDYE	BRDY Interrupt Enable for Pipe 8	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b9	PIPE9BRDYE	BRDY Interrupt Enable for Pipe 9	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The BRDYENB register enables or disables the INTSTS0.BRDY flag to be set to 1 when the BRDY interrupt is detected for each pipe.

On detecting the BRDY interrupt for the pipe corresponding to the bit in the BRDYENB register to which 1 has been set by software, the USB sets 1 to the corresponding BRDYSTS.PIPEnBRDY flag (n = 0 to 9) and the INTSTS0.BRDY flag. If INTENB0.BRDYE = 1 at this time, the USB generates the BRDY interrupt request.

While at least one PIPEnBRDY flag indicates 1, the USB generates the BRDY interrupt request when the corresponding interrupt enable bit in the BRDYENB register is modified from 0 to 1 by software.

Downloaded from Arrow.com.

32.2.10 NRDY Interrupt Enable Register (NRDYENB)

Address(es): USB0.NRDYENB 000A 0038h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PIPE9NRDYE	PIPE8NRDYE	PIPE7NRDYE	PIPE6NRDYE	PIPE5NRDYE	PIPE4NRDYE	PIPE3NRDYE	PIPE2NRDYE	PIPE1NRDYE	PIPE0NRDYE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PIPE0NRDYE	NRDY Interrupt Enable for Pipe 0	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b1	PIPE1NRDYE	NRDY Interrupt Enable for Pipe 1	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b2	PIPE2NRDYE	NRDY Interrupt Enable for Pipe 2	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b3	PIPE3NRDYE	NRDY Interrupt Enable for Pipe 3	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b4	PIPE4NRDYE	NRDY Interrupt Enable for Pipe 4	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b5	PIPE5NRDYE	NRDY Interrupt Enable for Pipe 5	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b6	PIPE6NRDYE	NRDY Interrupt Enable for Pipe 6	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b7	PIPE7NRDYE	NRDY Interrupt Enable for Pipe 7	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b8	PIPE8NRDYE	NRDY Interrupt Enable for Pipe 8	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b9	PIPE9NRDYE	NRDY Interrupt Enable for Pipe 9	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The NRDYENB register enables or disables the INTSTS0.NRDY flag to be set to 1 when the NRDY interrupt is detected for each pipe.

On detecting the NRDY interrupt for the pipe corresponding to the bit in the NRDYENB register to which 1 has been set by software, the USB sets 1 to the corresponding NRDYSTS.PIPEnNRDY flag (n = 0 to 9) and the INTSTS0.NRDY flag. If INTENB0.NRDYE = 1 at this time, the USB generates the NRDY interrupt request.

While at least one PIPEnNRDY flag indicates 1, the USB generates the NRDY interrupt request when the corresponding interrupt enable bit in the NRDYENB register is modified from 0 to 1 by software.

Downloaded from Arrow.com.

32.2.11 BEMP Interrupt Enable Register (BEMPENB)

Address(es): USB0.BEMPENB 000A 003Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PIPE9BEMPE	PIPE8BEMPE	PIPE7BEMPE	PIPE6BEMPE	PIPE5BEMPE	PIPE4BEMPE	PIPE3BEMPE	PIPE2BEMPE	PIPE1BEMPE	PIPE0BEMPE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PIPE0BEMPE	BEMP Interrupt Enable for Pipe 0	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b1	PIPE1BEMPE	BEMP Interrupt Enable for Pipe 1	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b2	PIPE2BEMPE	BEMP Interrupt Enable for Pipe 2	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b3	PIPE3BEMPE	BEMP Interrupt Enable for Pipe 3	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b4	PIPE4BEMPE	BEMP Interrupt Enable for Pipe 4	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b5	PIPE5BEMPE	BEMP Interrupt Enable for Pipe 5	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b6	PIPE6BEMPE	BEMP Interrupt Enable for Pipe 6	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b7	PIPE7BEMPE	BEMP Interrupt Enable for Pipe 7	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b8	PIPE8BEMPE	BEMP Interrupt Enable for Pipe 8	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b9	PIPE9BEMPE	BEMP Interrupt Enable for Pipe 9	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The BEMPENB register enables or disables the INTSTS0.BEMP flag to be set to 1 when the BEMP interrupt is detected for each pipe.

On detecting the BEMP interrupt for the pipe corresponding to the bit in the BEMPENB register to which 1 has been set by software, the USB sets 1 to the corresponding BEMPSTS.PIPEnBEMP flag (n = 0 to 9) and the INTSTS0.BEMP flag. If INTENB0.BEMPE = 1 at this time, the USB generates the BEMP interrupt request.

While at least one PIPEnBEMP flag in the BEMPSTS register indicates 1, the USB generates the BEMP interrupt request when the corresponding interrupt enable bit in the BEMPENB register is modified from 0 to 1 by software.

Downloaded from Arrow.com.

32.2.12 SOF Output Configuration Register (SOFCFG)

Address(es): USB0.SOFCFG 000A 003Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	TRNENSEL	—	BRDYM	—	EDGESTS	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	EDGESTS	Edge Interrupt Output Status Monitor Flag*1	Indicates 1 when the edge interrupt output signal is in the middle of the edge processing.	R
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	BRDYM	BRDY Interrupt Status Clear Timing	0: Software clears the status. 1: The USB clears the status when data has been read from the FIFO buffer or data has been written to the FIFO buffer.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b8	TRNENSEL	Transaction-Enabled Time Select*1	0: For non-low-speed communication 1: For low-speed communication	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Confirm that this bit is 0 before stopping the clock supply to the USB module.

EDGESTS Flag (Edge Interrupt Output Status Monitor Flag)

The EDGESTS flag indicates 1 when the edge interrupt output signal is in the middle of the edge processing.
Confirm that this flag is 0 before stopping the clock supply to the USB.

BRDYM Bit (BRDY Interrupt Status Clear Timing)

The BRDYM bit specifies the timing for clearing the BRDY interrupt status for each pipe.

TRNENSEL Bit (Transaction-Enabled Time Select)

The TRNENSEL bit selects, for full-speed or low-speed communication, the transaction-enabled time in which the USB issues tokens in a frame via the port.
Set the TRNENSEL bit to 1 when a low-speed device is connected.
The TRNENSEL bit is valid only when the host controller is selected.
This bit should be set to 0 if the function controller is selected.

Downloaded from Arrow.com.

32.2.13 Interrupt Status Register 0 (INTSTS0)

Address(es): USB0.INTSTS0 000A 0040h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	VBINT	RESM	SOFR	DVST	CTRT	BEMP	NRDY	BRDY	VBSTS	DVSQ[2:0]			VALID	CTSQ[2:0]		
Value after reset:	0	0	0	0/1*1	0	0	0	0	0*2	0*3	0*3	0/1*3	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	CTSQ[2:0]	Control Transfer Stage Flag	b2 b0 0 0 0: Idle or setup stage 0 0 1: Control read data stage 0 1 0: Control read status stage 0 1 1: Control write data stage 1 0 0: Control write status stage 1 0 1: Control write (no data) status stage 1 1 0: Control transfer sequence error	R
b3	VALID	USB Request Reception Flag	0: Setup packet is not received. 1: Setup packet is received.	R/W
b6 to b4	DVSQ[2:0]	Device State Flag	b6 b4 0 0 0: Powered state 0 0 1: Default state 0 1 0: Address state 0 1 1: Configured state 1 x x: Suspended state	R
b7	VBSTS	VBUS Input Status Flag	0: USB0_VBUS pin is low. 1: USB0_VBUS pin is high.	R
b8	BRDY	Buffer Ready Interrupt Status Flag	0: BRDY interrupts are not generated. 1: BRDY interrupts are generated.	R
b9	NRDY	Buffer Not Ready Interrupt Status Flag	0: NRDY interrupts are not generated. 1: NRDY interrupts are generated.	R
b10	BEMP	Buffer Empty Interrupt Status Flag	0: BEMP interrupts are not generated. 1: BEMP interrupts are generated.	R
b11	CTRT	Control Transfer Stage Transition Interrupt Status Flag*5	0: Control transfer stage transition interrupts are not generated. 1: Control transfer stage transition interrupts are generated.	R/W*4
b12	DVST	Device State Transition Interrupt Status Flag*5	0: Device state transition interrupts are not generated. 1: Device state transition interrupts are generated.	R/W*4
b13	SOFR	Frame Number Refresh Interrupt Status Flag	0: SOF interrupts are not generated. 1: SOF interrupts are generated.	R/W*4
b14	RESM	Resume Interrupt Status Flag *5,*6	0: Resume interrupts are not generated. 1: Resume interrupts are generated.	R/W*4
b15	VBINT	VBUS Interrupt Status Flag*6	0: VBUS interrupts are not generated. 1: VBUS interrupts are generated.	R/W*4

x: Don't care

Note 1. The value is 0 when the MCU is reset and 1 after a USB bus reset.
Note 2. The value is 1 when the USB0_VBUS pin is high and 0 when the USB0_VBUS pin is low.
Note 3. The value is 000b when the MCU is reset and 001b after a USB bus reset.
Note 4. To clear the VBINT, RESM, SOFR, DVST, CTRT, or VALID flag, write 0 only to the flags to be cleared; write 1 to the other flags. Do not write 0 to the status flags indicating 0.
Note 5. The status of the RESM, DVST, and CTRT flags are changed only when the function controller is selected. Set the corresponding interrupt enable bits to 0 (disabled) when the host controller is selected.
Note 6. A change in the status indicated by the VBINT and RESM flags can be detected even while the clock supply is stopped (the SCKE bit = 0), and the interrupts are output when the corresponding interrupt enable bits are enabled. Clearing the status through software should be done after enabling the clock supply.

Downloaded from Arrow.com.

CTSQ[2:0] Flags (Control Transfer Stage Flag)

When the host controller is selected, the read value is invalid.

VALID Flag (USB Request Reception Flag)

When the host controller is selected, the read value is invalid.

DVSQ[2:0] Flags (Device State Flag)

The DVSQ[2:0] flags are initialized by a USB bus reset.
When the host controller is selected, the read value is invalid.

BRDY Flag (Buffer Ready Interrupt Status Flag)

Indicates the BRDY interrupt status.
The USB sets the BRDY flag to 1 when at least one PIPEnBRDY flag (n = 0 to 9) is set to 1 among the PIPEnBRDY flags. These bits correspond to the BRDYENB.PIPEnBRDYE bits (n = 0 to 9) to which 1 has been set, when the USB detects the BRDY interrupt status in at least one pipe among the pipes for which the BRDY interrupt output is enabled by software.
For the conditions for PIPEnBRDY status assertion, refer to section 32.3.3.1, BRDY Interrupt.
The USB sets the BRDY flag to 0 when 0 is written by software to all the PIPEnBRDY flags corresponding to the PIPEnBRDYE bits that have been set to 1.
The BRDY flag cannot be set to 0 even if 0 is written to this bit by software.

NRDY Flag (Buffer Not Ready Interrupt Status Flag)

The USB sets the NRDY flag to 1 when at least one PIPEnNRDY flag (n = 0 to 9) is set to 1 among the PIPEnNRDY flags corresponding to the PIPEnNRDYE bits (n = 0 to 9) to which 1 has been set (when the USB detects the NRDY interrupt status in at least one pipe among the pipes for which software enables the NRDY interrupt output).
For the conditions for PIPEnNRDY status assertion, refer to section 32.3.3.2, NRDY Interrupt.
The USB sets the NRDY flag to 0 when 0 is written by software to all the PIPEnNRDY flags corresponding to the PIPEnNRDYE bits that have been set to 1.
The NRDY flag cannot be set to 0 even if 0 is written to this bit by software.

BEMP Flag (Buffer Empty Interrupt Status Flag)

The USB sets the BEMP flag to 1 when at least one PIPEnBEMP flag (n = 0 to 9) is set to 1 among the PIPEnBEMP flags corresponding to the PIPEnBEMPE bits (n = 0 to 9) to which 1 has been set (when the USB detects the BEMP interrupt status in at least one pipe among the pipes for which the BEMP interrupt output is enabled by software).
For the conditions for PIPEnBEMP status assertion, refer to section 32.3.3.3, BEMP Interrupt.
The USB sets the BEMP flag to 0 when 0 is written by software to all the PIPEnBEMP flags corresponding to the PIPEnBEMPE bits that have been set to 1.
The BEMP flag cannot be set to 0 even if 0 is written to this bit by software.

CTRT Flag (Control Transfer Stage Transition Interrupt Status Flag)

When the function controller is selected, the USB updates the value of the CTSQ[2:0] flags and sets the CTRT flag to 1 on detecting a change in the control transfer stage.
When a control transfer stage transition interrupt is generated, clear the status before the USB detects the next control transfer stage transition.
When the host controller is selected, the read value is invalid.

Downloaded from Arrow.com.

DVST Flag (Device State Transition Interrupt Status Flag)

When the function controller is selected, the USB updates the DVSQ[2:0] value and sets the DVST flag to 1 on detecting a change in the device state.
When a device state transition interrupt is generated, clear the status before the USB detects the next device state transition.
When the host controller is selected, the read value is invalid.

SOFR Flag (Frame Number Refresh Interrupt Status Flag)

(1) When the host controller is selected
The USB sets the SOFR flag to 1 on updating the frame number when the DVSTCTR0.UACT bit has been set to 1 by software. (A frame number refresh interrupt is detected every 1 ms.)
(2) When the function controller is selected
The USB sets the SOFR flag to 1 on updating the frame number. (A frame number refresh interrupt is detected every 1 ms.)
The USB can detect an SOFR interrupt through the internal interpolation function even when a damaged SOF packet is received from the USB host.

RESM Flag (Resume Interrupt Status Flag)

When the function controller is selected, the USB sets the RESM flag to 1 on detecting the falling edge of the signal on the USB0_DP pin in the suspended state (DVSQ[2:0] = 1xxb).
When the host controller is selected, the read value is invalid.

VBINT Flag (VBUS Interrupt Status Flag)

The USB sets the VBINT flag to 1 on detecting a level change (high to low or low to high) in the USB0_VBUS pin input value. The USB sets the VBSTS flag to indicate the USB0_VBUS pin input value. When the VBUS interrupt is generated, use software to repeat reading the VBSTS flag until the same value is read three or more times, and eliminate chattering.

Downloaded from Arrow.com.

32.2.14 Interrupt Status Register 1 (INTSTS1)

Address(es): USB0.INTSTS1 000A 0042h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	OVRCR	BCHG	—	DTCH	ATTCH	—	—	—	—	EOFERR	SIGN	SACK	—	—	—	PDDETINT0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PDDETINT0	Portable Device Detection Interrupt Status Flag	0: PDDETINT interrupts are not generated. 1: PDDETINT interrupts are generated.	R/W *1
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	SACK	Setup Transaction Normal Response Interrupt Status Flag	0: SACK interrupts are not generated. 1: SACK interrupts are generated.	R/W *1
b5	SIGN	Setup Transaction Error Interrupt Status Flag	0: SIGN interrupts are not generated. 1: SIGN interrupts are generated.	R/W *1
b6	EOFERR	EOF Error Detection Interrupt Status Flag	0: EOFERR interrupts are not generated. 1: EOFERR interrupts are generated.	R/W *1
b10 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11	ATTCH	ATTCH Interrupt Status Flag	0: ATTCH interrupts are not generated. 1: ATTCH interrupts are generated.	R/W *1
b12	DTCH	USB Disconnection Detection Interrupt Status Flag	0: DTCH interrupts are not generated. 1: DTCH interrupts are generated.	R/W *1
b13	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b14	BCHG	USB Bus Change Interrupt Status Flag *2	0: BCHG interrupts are not generated. 1: BCHG interrupts are generated.	R/W *1
b15	OVRCR	Overcurrent Input Change Interrupt Status Flag*2	0: OVRCR interrupts are not generated. 1: OVRCR interrupts are generated.	R/W *1

Note 1. To clear the status indicated by the flags in the INTSTS1 register, write 0 only to the flags to be cleared; write 1 to the other flags.

Note 2. A change in the status indicated by the OVRCR or BCHG flag can be detected even while the clock supply is stopped (while the SYSCFG.SCKE bit = 0), and the interrupt is output when the corresponding interrupt enable bit is enabled. Clearing the status through software should be done after setting the SYSCFG.SCKE bit to 1.
No interrupts other than those indicated by the BCHG and OVRCR flags can be detected while the clock supply is stopped (while the SYSCFG.SCKE bit = 0).

The INTSTS1 register is used to confirm the status of each interrupt when the host controller is selected.
The various status change interrupts indicated by the flags in the INTSTS1 register should be enabled only when the host controller is selected.

PDDETINT0 Flag (Portable Device Detection Interrupt Status Flag)

Indicates the status of the portable device detection interrupt when the host controller is selected.
This bit is set to 1 when the USB module detects when a level change (high to low or low to high) occurs in the input value to the VDPDET pin of the USB physical layer transceiver (PHY). The USB module sets the PDDETSTS0 flag to indicate the VDPDET input value.
When the PDDETINT interrupt is generated, use software to repeat reading the PDDETSTS0 flag until the same value is read three or more times, and eliminate chattering.

Downloaded from Arrow.com.

SACK Flag (Setup Transaction Normal Response Interrupt Status Flag)

Indicates the status of the setup transaction normal response interrupt when the host controller is selected.
The USB detects the SACK interrupt when ACK response is returned from the peripheral device during the setup transactions issued by the USB, and sets the SACK flag to 1. Here, if the corresponding interrupt enable bit has been set to 1 by software, the USB generates the SACK interrupt.
When the function controller is selected, the read value is invalid.

SIGN Flag (Setup Transaction Error Interrupt Status Flag)

Indicates the status of the setup transaction error interrupt when the host controller is selected.
The USB detects the SIGN interrupt when ACK response is not returned from the peripheral device three consecutive times during the setup transactions issued by this module, and sets the SIGN flag to 1. Here, if the corresponding interrupt enable bit has been set to 1 by software, the USB generates the SIGN interrupt.
Specifically, the USB detects the SIGN interrupt when any of the following response conditions occur for three consecutive setup transactions.

- Timeout is detected by the USB when the peripheral device has returned no response.
- A damaged ACK packet is received.
- A handshake other than ACK (NAK, NYET, or STALL) is received.

When the function controller is selected, the read value is invalid.

EOFERR Flag (EOF Error Detection Interrupt Status Flag)

Indicates the status of the EOFERR interrupt when the host controller is selected.
The USB detects the EOFERR interrupt on detecting that communication is not completed at the EOF2 timing prescribed by USB Specification 2.0, and sets the EOFERR flag to 1. Here, if the corresponding interrupt enable bit has been set to 1 by software, the USB generates the EOFERR interrupt.
After detecting the EOFERR interrupt, the USB controls hardware as described below (irrespective of the setting of the corresponding interrupt enable bit). All pipes in which communications are currently carried out for the USB port should be terminated by software and perform re-enumeration of the USB port.

- Modifies the DVSTCTR0.UACT bit for the port in which an EOFERR interrupt has been detected to 0.
- Puts the port in which an EOFERR interrupt has been generated into the idle state.

When the function controller is selected, the read value is invalid.

ATTCH Flag (ATTCH Interrupt Status Flag)

Indicates the status of the ATTCH interrupt when the host controller is selected.
The USB detects the ATTCH interrupt on detecting J-state or K-state of the full-speed or low-speed signal level for 2.5 µs, and sets the ATTCH flag to 1. Here, if the corresponding interrupt enable bit has been set to 1 by software, the USB generates the interrupt.
Specifically, the USB detects the ATTCH interrupt on any of the following conditions.

- K-state, SE0, or SE1 changes to J-state, and J-state continues for 2.5 µs.
- J-state, SE0, or SE1 changes to K-state, and K-state continues for 2.5 µs.

When the function controller is selected, the read value is invalid.

Downloaded from Arrow.com.

DTCH Flag (USB Disconnection Detection Interrupt Status Flag)

Indicates the status of the USB disconnection detection interrupt when the host controller is selected.
The USB detects the DTCH interrupt on detecting USB bus disconnection, and sets the DTCH flag to 1. Here, if the corresponding interrupt enable bit has been set to 1 by software, the USB generates the interrupt.
The USB detects bus disconnection based on USB Specification 2.0.
After detecting the DTCH interrupt, the USB controls hardware as described below (irrespective of the setting of the corresponding interrupt enable bit). All the pipes in which communications are currently carried out for the USB port should be terminated by software and make a transition to the wait state for bus connection to the USB port (wait state for ATTCH interrupt generation).

- Modifies the DVSTCTR0.UACT bit for the port in which a DTCH interrupt has been detected to 0.
- Puts the port in which a DTCH interrupt has been generated into the idle state.

When the function controller is selected, the read value is invalid.

BCHG Flag (USB Bus Change Interrupt Status Flag)

Indicates the status of the USB bus change interrupt.
The USB detects the BCHG interrupt when a change in the full-speed or low-speed signal level occurs on the USB port (a change from J-state, K-state, or SE0 to J-state, K-state, or SE0), and sets the BCHG flag to 1. Here, if the corresponding interrupt enable bit has been set to 1 by software, the USB generates the interrupt.
The USB sets the LNST[1:0] flags to indicate the current input state of the USB port. When the BCHG interrupt is generated, use software to repeat reading the LNST[1:0] flags until the same value is read three or more times, and eliminate chattering.
A change in the USB bus state can be detected even while the internal clock supply is stopped.
When the function controller is selected, the read value is invalid.

OVRCR Flag (Overcurrent Input Change Interrupt Status Flag)

Indicates the status of the USB0_OVRCURA and USB0_OVRCURB input pin change interrupt.
The USB detects the OVRCR interrupt when a change (high to low or low to high) occurs in at least one of the input values to the USB0_OVRCURA and USB0_OVRCURB pins, and sets the OVRCR flag to 1. Here, if the corresponding interrupt enable bit has been set to 1 by software, the USB generates the interrupt.

Downloaded from Arrow.com.

32.2.15 BRDY Interrupt Status Register (BRDYSTS)

Address(es): USB0.BRDYSTS 000A 0046h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PIPE9BRDY	PIPE8BRDY	PIPE7BRDY	PIPE6BRDY	PIPE5BRDY	PIPE4BRDY	PIPE3BRDY	PIPE2BRDY	PIPE1BRDY	PIPE0BRDY
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PIPE0BRDY	BRDY Interrupt Status Flag for Pipe 0*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b1	PIPE1BRDY	BRDY Interrupt Status Flag for Pipe 1*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b2	PIPE2BRDY	BRDY Interrupt Status Flag for Pipe 2*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b3	PIPE3BRDY	BRDY Interrupt Status Flag for Pipe 3*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b4	PIPE4BRDY	BRDY Interrupt Status Flag for Pipe 4*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b5	PIPE5BRDY	BRDY Interrupt Status Flag for Pipe 5*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b6	PIPE6BRDY	BRDY Interrupt Status Flag for Pipe 6*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b7	PIPE7BRDY	BRDY Interrupt Status Flag for Pipe 7*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b8	PIPE8BRDY	BRDY Interrupt Status Flag for Pipe 8*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b9	PIPE9BRDY	BRDY Interrupt Status Flag for Pipe 9*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. When the SOFCFG.BRDYM bit is set to 0, to clear the status indicated by the flags in the BRDYSTS register, write 0 only to the flags to be cleared; write 1 to the other flags.
Note 2. When the SOFCFG.BRDYM bit is set to 0, clearing BRDY Interrupts should be done before accessing the FIFO.

Downloaded from Arrow.com.

32.2.16 NRDY Interrupt Status Register (NRDYSTS)

Address(es): USB0.NRDYSTS 000A 0048h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PIPE9NRDY	PIPE8NRDY	PIPE7NRDY	PIPE6NRDY	PIPE5NRDY	PIPE4NRDY	PIPE3NRDY	PIPE2NRDY	PIPE1NRDY	PIPE0NRDY
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PIPE0NRDY	NRDY Interrupt Status Flag for Pipe 0	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b1	PIPE1NRDY	NRDY Interrupt Status Flag for Pipe 1	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b2	PIPE2NRDY	NRDY Interrupt Status Flag for Pipe 2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b3	PIPE3NRDY	NRDY Interrupt Status Flag for Pipe 3	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b4	PIPE4NRDY	NRDY Interrupt Status Flag for Pipe 4	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b5	PIPE5NRDY	NRDY Interrupt Status Flag for Pipe 5	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b6	PIPE6NRDY	NRDY Interrupt Status Flag for Pipe 6	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b7	PIPE7NRDY	NRDY Interrupt Status Flag for Pipe 7	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b8	PIPE8NRDY	NRDY Interrupt Status Flag for Pipe 8	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b9	PIPE9NRDY	NRDY Interrupt Status Flag for Pipe 9	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. To clear the status indicated by the flags in the NRDYSTS register, write 0 only to the flags to be cleared; write 1 to the other flags.

Downloaded from Arrow.com.

32.2.17 BEMP Interrupt Status Register (BEMPSTS)

Address(es): USB0.BEMPSTS 000A 004Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PIPE9BEMP	PIPE8BEMP	PIPE7BEMP	PIPE6BEMP	PIPE5BEMP	PIPE4BEMP	PIPE3BEMP	PIPE2BEMP	PIPE1BEMP	PIPE0BEMP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PIPE0BEMP	BEMP Interrupt Status Flag for Pipe 0	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b1	PIPE1BEMP	BEMP Interrupt Status Flag for Pipe 1	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b2	PIPE2BEMP	BEMP Interrupt Status Flag for Pipe 2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b3	PIPE3BEMP	BEMP Interrupt Status Flag for Pipe 3	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b4	PIPE4BEMP	BEMP Interrupt Status Flag for Pipe 4	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b5	PIPE5BEMP	BEMP Interrupt Status Flag for Pipe 5	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b6	PIPE6BEMP	BEMP Interrupt Status Flag for Pipe 6	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b7	PIPE7BEMP	BEMP Interrupt Status Flag for Pipe 7	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b8	PIPE8BEMP	BEMP Interrupt Status Flag for Pipe 8	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b9	PIPE9BEMP	BEMP Interrupt Status Flag for Pipe 9	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. To clear the status indicated by the flags in the BEMPSTS register, write 0 only to the flags to be cleared; write 1 to the other flags.

Downloaded from Arrow.com.

32.2.18 Frame Number Register (FRMNUM)

Address(es): USB0.FRMNUM 000A 004Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	OVRN	CRCE	—	—	—	FRNM[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b10 to b0	FRNM[10:0]	Frame Number Flag	Latest frame number	R
b13 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b14	CRCE	Receive Data Error Flag	0: No error 1: An error occurred	R/W*1
b15	OVRN	Overrun/Underrun Detection Status Flag	0: No error 1: An error occurred	R/W*1

Note 1. To clear the status, write 0 only to the bits to be cleared; write 1 to the other bits.

FRNM[10:0] Flags (Frame Number Flag)

These bits indicate the latest frame number for the USB after the issuing of an SOF packet every 1 ms or writing to the FRNM[10:0] flags at the SOF packet reception.

CRCE Flag (Receive Data Error Flag)

Indicates whether a CRC error or bit stuffing error has been detected in the pipe during isochronous transfer.
The CRCE flag can be set to 0 by writing 0 to the CRCE flag by software. Here, 1 should be written to the other bits in FRMNUM.
On detecting a CRC error, the USB generates the internal NRDY interrupt request.

OVRN Flag (Overrun/Underrun Detection Status Flag)

Indicates whether an overrun/underrun error has been detected in the pipe during isochronous transfer.
The OVRN flag can be set to 0 by writing 0 to the OVRN flag by software. Here, 1 should be written to the other bits in FRMNUM.
(1) When the host controller is selected
The USB sets the OVRN flag to 1 on any of the following conditions.

- For the isochronous transfer pipe in the transmitting direction, the time to issue an OUT token comes before all the transmit data has been written to the FIFO buffer.
- For the isochronous transfer pipe in the receiving direction, the time to issue an IN token comes when no FIFO buffer planes are empty.

(2) When the function controller is selected
The USB sets the OVRN flag to 1 on any of the following conditions.

- For the isochronous transfer pipe in the transmitting direction, the IN token is received before all the transmit data has been written to the FIFO buffer.
- For the isochronous transfer pipe in the receiving direction, the OUT token is received when no FIFO buffer planes are empty.

Downloaded from Arrow.com.

32.2.19 USB Request Type Register (USBREQ)

Address(es): USB0.USBREQ 000A 0054h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	BREQUEST[7:0]								BMREQUESTTYPE[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	BMREQUESTTYPE[7:0]	Request Type	These bits store the USB request bmRequestType value.	R/W *1
b15 to b8	BREQUEST[7:0]	Request	These bits store the USB request bRequest value.	R/W *1

Note 1. When the function controller is selected, these bits can only be read from, and writing to these bits is invalid. When the host controller is selected, these bits can be read from and written to.

The USBREQ register stores setup requests for control transfers.
When the function controller is selected, the values of bRequest and bmRequestType that have been received are stored.
When the host controller is selected, the values of bRequest and bmRequestType to be transmitted are set.
The USBREQ register is initialized by a USB bus reset.

BMREQUESTTYPE[7:0] Bits (Request Type)

These bits hold the value of the bmRequestType field of a USB request.

- When the host controller is selected:
  Set these bits to the value of the USB request data in setup transactions for transmission. Do not modify the value of the BMREQUESTTYPE[7:0] bits while the DCPCTR.SUREQ bit is 1.
- When the function controller is selected:
  These bits indicate the value of the USB request data in setup transactions for reception. Writing to the bits has no effect.

BREQUEST[7:0] Bits (Request)

These bits store bRequest value of the USB request.

- When the host controller is selected:
  Set these bits to the value of the USB request data in setup transactions for transmission. Do not modify the value of the BREQUEST[7:0] bits while the DCPCTR.SUREQ bit is 1.
- When the function controller is selected:
  These bits indicate the value of the USB request data in setup transactions for reception. Writing to the bits has no effect.

Downloaded from Arrow.com.

32.2.20 USB Request Value Register (USBVAL)

Address(es): USB0.USBVAL 000A 0056h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

When the function controller is selected, the value of wValue that has been received is stored in the USBVAL register.
When the host controller is selected, the value of wValue to be transmitted is set.
The USBVAL register is initialized by a USB bus reset.

- When the host controller is selected:
  Set these bits to the value of the wValue field in USB requests of setup transactions for transmission. Do not modify the value of the USBVAL register while the DCPCTR.SUREQ bit is 1.
- When the function controller is selected:
  These bits indicate the value of the wValue field in USB requests received in setup transactions for reception. Writing to the USBVAL register has no effect.

32.2.21 USB Request Index Register (USBINDX)

Address(es): USB0.USBINDX 000A 0058h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The USBINDX register stores setup requests for control transfers.
When the function controller is selected, the value of wIndex that has been received is stored. When the host controller is selected, the value of wIndex to be transmitted is set.
The USBINDX register is initialized by a USB bus reset.

- When the host controller is selected:
  Set these bits to the value of the wIndex field in USB requests of setup transactions for transmission. Do not modify the value of the USBINDX register while the DCPCTR.SUREQ bit is 1.
- When the function controller is selected:
  These bits indicate the value of the wIndex field in USB requests received in setup transactions for reception. Writing to the USBINDX register has no effect.

Downloaded from Arrow.com.

32.2.22 USB Request Length Register (USBLENG)

Address(es): USB0.USBLENG 000A 005Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The USBLENG register stores setup requests for control transfers.
When the function controller is selected, the value of wLength that has been received is stored. When the host controller is selected, the value of wLength to be transmitted is set.
The USBLENG register is initialized by a USB bus reset.

- When the host controller is selected:
  Set these bits to the value of the wLength field in USB requests of setup transactions for transmission. Do not modify the value of the USBLENG register while the DCPCTR.SUREQ bit is 1.
- When the function controller is selected:
  These bits indicate the value of the wLength field in USB requests received in setup transactions for reception. Writing to the USBLENG register has no effect.

Downloaded from Arrow.com.

32.2.23 DCP Configuration Register (DCPCFG)

Address(es): USB0.DCPCFG 000A 005Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	SHTNAK	—	—	DIR	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	DIR	Transfer Direction*1	0: Data receiving direction 1: Data transmitting direction	R/W
b6, b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	SHTNAK	Pipe Disabled at End of Transfer*1	0: Pipe continued at the end of transfer 1: Pipe disabled at the end of transfer	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Modify this bit while DCPCTR.PID[1:0] bits are 00b (NAK). Before modifying this bit after modifying the DCPCTR.PID[1:0] bits for the DCP from 01b (BUF) to 00b (NAK), check that the DCPCTR.PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.

DIR Bit (Transfer Direction)

When the host controller is selected, the DIR bit sets the transfer direction of the data stage and status stage.
When the function controller is selected, the DIR bit should be set to 0.

SHTNAK Bit (Pipe Disabled at End of Transfer)

The SHTNAK bit specifies whether to modify PID[1:0] bits to 00b (NAK) upon the end of transfer when the selected pipe is in the receiving direction.
The SHTNAK bit is valid when the selected pipe in the receiving direction.
When the SHTNAK bit is set to 1, the USB modifies the DCPCTR.PID[1:0] bits for the DCP to 00b (NAK) on determining the end of the transfer. The USB determines that the transfer has ended on the following condition.

- A short packet (including a zero-length packet) is successfully received.

Downloaded from Arrow.com.

32.2.24 DCP Maximum Packet Size Register (DCPMAXP)

Address(es): USB0.DCPMAXP 000A 005Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DEVSEL[3:0]				—	—	—	—	—	MXPS[6:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	MXPS[6:0]	Maximum Packet Size*1	These bits set the maximum amount of data (maximum packet size) in payloads for the DCP. b6 b0 0 0 0 1 0 0 0: 8 bytes 0 0 1 0 0 0 0: 16 bytes 0 0 1 1 0 0 0: 24 bytes 0 1 0 0 0 0 0: 32 bytes 0 1 0 1 0 0 0: 40 bytes 0 1 1 0 0 0 0: 48 bytes 0 1 1 0 0 0 0: 56 bytes 1 0 0 0 0 0 0: 64 bytes 1 0 0 1 0 0 0: 72 bytes 1 0 1 0 0 0 0: 80 bytes 1 0 1 1 0 0 0: 88 bytes 1 1 0 0 0 0 0: 96 bytes 1 1 0 1 0 0 0: 104 bytes 1 1 1 0 0 0 0: 112 bytes 1 1 1 1 0 0 0: 120 bytes Settings other than above are prohibited.	R/W
b11 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b12	DEVSEL[3:0]	Device Select*2	b15 b12 0 0 0 0: Address 0000 0 0 0 1: Address 0001 0 0 1 0: Address 0010 0 0 1 1: Address 0011 0 1 0 0: Address 0100 0 1 0 1: Address 0101 Settings other than above are prohibited.	R/W

Note 1. Modify the MXPS[6:0] bits while PID[1:0] bits are 00b (NAK). Before modifying these bits after modifying the DCPCTR.PID[1:0] bits for the DCP from 01b (BUF) to 00b (NAK), check that the DCPCTR.PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software. After modifying the MXPS[6:0] bits and the DCP has been set to the CURPIPE[3:0] bits in a port select register, clear the buffer by setting the BCLR bit the port control register to 1.

Note 2. Modify the DEVSEL[3:0] bits while PID[1:0] bits are 00b (NAK) and the DCPCTR.SUREQ bit is 0. To modify these bits after modifying the DCPCTR.PID[1:0] bits for the DCP from 01b (BUF) to 00b (NAK), check that the DCPCTR.PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.

MXPS[6:0] Bits (Maximum Packet Size)

The MXPS[6:0] bits specify the maximum amount of data (maximum packet size) in payloads for the DCP. The initial value of the bits is 40h (64 bytes).

Ensure that the setting of the MXPS[6:0] bits is in compliance with USB Specification 2.0.

Do not write to the FIFO buffer or set PID[1:0] = 01b (BUF) while the setting of the MXPS[6:0] bits is 0.

DEVSEL[3:0] Bits (Device Select)

When the host controller is selected, these bits specify the address of the peripheral device which is the communication target during control transfer.

The DEVSEL[3:0] bits should be set after setting the address to the DEVADDn (n = 0 to 5) register corresponding to the value to be set in the DEVSEL[3:0] bits. For example, before setting the DEVSEL[3:0] bits to 0010b, the address should be set to DEVADD2.

When the function controller is selected, the DEVSEL[3:0] bits should be set to 0000b.

Downloaded from Arrow.com.

32.2.25 DCP Control Register (DCPCTR)

Address(es): USB0.DCPCTR 000A 0060h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	BSTS	SUREQ	—	—	SUREQ CLR	—	—	SQCLR	SQSET	SQMON	PBUSY	—	—	CCPL	PID[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	PID[1:0]	Response PID	b1 b0 0 0: NAK response 0 1: BUF response (depending on the buffer state) 1 0: STALL response 1 1: STALL response	R/W
b2	CCPL	Control Transfer End Enable	0: Invalid 1: Completion of control transfer is enabled.	R/W
b4, b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5	PBUSY	Pipe Busy Flag	0: DCP is not used for the transaction. 1: DCP is used for the transaction.	R
b6	SQMON	Sequence Toggle Bit Monitor Flag	0: DATA0 1: DATA1	R
b7	SQSET	Sequence Toggle Bit Set*2	0: Invalid 1: Specifies DATA1.	R/W*1
b8	SQCLR	Sequence Toggle Bit Clear*2	0: Invalid 1: Specifies DATA0.	R/W*1
b10, b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11	SUREQCLR	SUREQ Bit Clear	0: Invalid 1: Clears the SUREQ bit to 0.	R/W
b13, b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b14	SUREQ	Setup Token Transmission	0: Invalid 1: Transmits the setup packet.	R/W
b15	BSTS	Buffer Status Flag	0: Buffer access is disabled. 1: Buffer access is enabled.	R

Note 1. Only 0 can be read.
Note 2. Write 1 to the SQSET and SQCLR bits while PID[1:0] bits are 00b (NAK). Before modifying these bits after modifying the PID[1:0] bits for the DCP from 01b (BUF) to 00b (NAK), check that the PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.

PID[1:0] Bits (Response PID)

The PID[1:0] bits control the response type of the USB during control transfer.

(1) When the host controller is selected

Modify the setting of the PID[1:0] bits from 00b (NAK) to 01b (BUF) using the following procedure.

- When the transmitting direction is set
  Write all the transmit data to the FIFO buffer while the DVSTCTR0.UACT bit is 1 and PID[1:0] bits are 00b (NAK), and then write 01b (BUF response). After PID[1:0] have been set to 01b (BUF), the USB executes the OUT transaction.
- When the receiving direction is set
  Check that the FIFO buffer is empty (or empty the buffer) while the DVSTCTR0.UACT bit is 1 and PID[1:0] bits are 00b (NAK), and then set PID[1:0] bits to 01b (BUF). After PID[1:0] bits have been set to 01b (BUF), the USB executes the IN transaction.

Downloaded from Arrow.com.

The USB modifies the setting of the PID[1:0] bits as follows.

- The USB sets PID[1:0] bits to 11b (STALL) on receiving the data of a size exceeding the maximum packet size when the PID[1:0] bits has been set to 01b (BUF) by software.
- The USB sets PID[1:0] bits to 00b (NAK) on detecting a receive error, such as a CRC error, three consecutive times.
- The USB also sets PID[1:0] bits to 11b (STALL) on receiving the STALL handshake.

(2) When the function controller is selected

The USB modifies the setting of the PID[1:0] bits as follows.

- The USB modifies the PID[1:0] bits to 00b (NAK) on receiving the setup packet. Here, the USB sets the INTSTS0.VALID flag to 1. The setting of the PID[1:0] bits cannot be modified until the VALID flag is set to 0 by software.
- The USB sets PID[1:0] bits to 11b (STALL) on receiving the data of a size exceeding the maximum packet size when the PID[1:0] bits have been set to 01b (BUF) by software.
- The USB sets PID[1:0] bits to 1xb (STALL) on detecting the control transfer sequence error.
- The USB sets PID[1:0] bits to 00b (NAK) on detecting the USB bus reset.

The USB does not check the setting of the PID[1:0] bits while the SET_ADDRESS request is processed.
The PID[1:0] bits are initialized by a USB bus reset.

CCPL Bit (Control Transfer End Enable)

When the function controller is selected, setting the CCPL bit to 1 enables the status stage of the control transfer to be completed.
When the CCPL bit is set to 1 by software while the corresponding PID[1:0] bits are set to 01b (BUF), the USB completes the control transfer status stage.
During control read transfer, the USB transmits the ACK handshake in response to the OUT transaction from the USB host, and transmits the zero-length packet in response to the IN transaction from the USB host during control write or no-data control transfer. However, on detecting the SET_ADDRESS request, the USB operates in auto response mode from the setup stage up to the status stage completion irrespective of the setting of the CCPL bit.
The USB modifies the CCPL bit from 1 to 0 on receiving a new setup packet.
1 cannot be written to the CCPL bit by software while the INTSTS0.VALID flag is 1.
The CCPL bit is initialized by a USB bus reset.
When the host controller is selected, be sure to write 0 to the CCPL bit.

PBUSY Flag (Pipe Busy Flag)

The PBUSY flag indicates whether DCP is used or not for the transaction when USB changes the PID[1:0] bits from 01b (BUF) to 00b (NAK).
The USB modifies the PBUSY flag from 0 to 1 upon start of the USB transaction for the relevant pipe, and modifies the PBUSY flag from 1 to 0 upon completion of one transaction.
Reading the PBUSY flag after the PID[1:0] bits have been set to 00b (NAK) by software allows checking whether modification of the pipe settings is possible.
For details, refer to section 32.3.4.1, Pipe Control Register Switching Procedures.

Downloaded from Arrow.com.

SQMON Flag (Sequence Toggle Bit Monitor Flag)

The SQMON flag indicates the expected value of the sequence toggle bit for the next transaction during the DCP transfer.
The USB allows the SQMON flag to toggle upon normal completion of the transaction. However, the SQMON flag is not allowed to toggle when a data PID mismatch occurs during the transfer in the receiving direction.
When the function controller is selected, the USB sets the SQMON flag to 1 (specifies DATA1 as the expected value) upon successful reception of the setup packet.
When the function controller is selected, the USB does not reference the SQMON flag during the IN/OUT transaction of the status stage, and does not allow the SQMON flag to toggle upon normal completion.

SQSET Bit (Sequence Toggle Bit Set)

The SQSET bit specifies DATA1 as the expected value of the sequence toggle bit for the next transaction during the DCP transfer.
Do not set the SQCLR and SQSET bits to 1 simultaneously.

SQCLR Bit (Sequence Toggle Bit Clear)

The SQCLR bit specifies DATA0 as the expected value of the sequence toggle bit for the next transaction during the DCP transfer. The SQCLR bit indicates 0.
Do not set the SQCLR and SQSET bits to 1 simultaneously.

SUREQCLR Bit (SUREQ Bit Clear)

When the host controller is selected, setting the SUREQCLR bit to 1 clears the SUREQ bit to 0. The SUREQCLR bit indicates 0.
Set the SUREQCLR bit to 1 through software when communication has stopped with the SUREQ bit being 1 during the setup transaction. However, for normal setup transactions, the USB automatically clears the SUREQ bit to 0 upon completion of the transaction; therefore, clearing the SUREQ bit through software is not necessary.
Controlling the SUREQ bit through the SUREQCLR bit must be done while the DVSTCTR0.UACT bit is 0 and thus communication is halted or while no transfer is being performed with bus disconnection detected.
When the function controller is selected, be sure to write 0 to the SUREQCLR bit.

SUREQ Bit (Setup Token Transmission)

The USB transmits the setup packet by setting the SUREQ bit to 1 when the host controller is selected.
After completing the setup transaction process, the USB generates either the SACK or SIGN interrupt and sets the SUREQ bit to 0.
The USB also sets the SUREQ bit to 0 when software sets the SUREQCLR bit to 1.
Before setting the SUREQ bit to 1, set the DCPMAXP.DEVSEL[3:0] bits, registers USBREQ, USBVAL, USBINDX, and USBLENG appropriately to transmit the desired USB request in the setup transaction. Before setting this bit to 1, check that the PID[1:0] bits for the DCP are set to 00b (NAK). After setting the SUREQ bit to 1, do not modify the DCPMAXP.DEVSEL[3:0] bits, registers USBREQ, USBVAL, USBINDX, or USBLENG until the setup transaction is completed (the SUREQ bit = 1).
Write 1 to the SUREQ bit only when transmitting the setup token; for other purposes, write 0.
When the function controller is selected, be sure to write 0 to the SUREQ bit.

BSTS Flag (Buffer Status Flag)

Indicates whether DCP FIFO buffer access is enabled or disabled.
The meaning of the BSTS flag depends on the setting of ISEL bit in the port select register as shown below.

- When the ISEL bit = 0, the BSTS flag indicates whether the received data can be read from the buffer.
- When the ISEL bit = 1, the BSTS flag indicates whether the data to be transmitted can be written to the buffer.

Downloaded from Arrow.com.

32.2.26 Pipe Window Select Register (PIPESEL)

Address(es): USB0.PIPESEL 000A 0064h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	PIPESEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	PIPESEL[3:0]	Pipe Window Select	b3 b0 0 0 0 0: No pipe selected 0 0 0 1: Pipe 1 0 0 1 0: Pipe 2 0 0 1 1: Pipe 3 0 1 0 0: Pipe 4 0 1 0 1: Pipe 5 0 1 1 0: Pipe 6 0 1 1 1: Pipe 7 1 0 0 0: Pipe 8 1 0 0 1: Pipe 9 Settings other than above are prohibited.	R/W
b15 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Pipes 1 to 9 should be set using registers PIPESEL, PIPECFG, PIPEMAXP, PIPEPERI, PIPEnCTR, PIPEnTRE, and PIPEnTRN.

After selecting the pipe using the PIPESEL register, functions of the pipe should be set using registers PIPECFG, PIPEMAXP, and PIPEPERI. PIPEnCTR, PIPEnTRE, and PIPEnTRN can be set regardless of the pipe selection in the PIPESEL register.

PIPESEL[3:0] Bits (Pipe Window Select)

The PIPESEL[3:0] bits select the pipe number corresponding to registers PIPECFG, PIPEMAXP, and PIPEPERI which data are written to or read from.

Selecting a pipe number through the PIPESEL[3:0] bits allows writing to and reading from registers PIPECFG, PIPEMAXP, and PIPEPERI which correspond to the selected pipe number.

When PIPESEL[3:0] = 0000b, 0 is read from all of the bits in registers PIPECFG, PIPEMAXP, and PIPEPERI. Writing to these bits is invalid.

Downloaded from Arrow.com.

32.2.27 Pipe Configuration Register (PIPECFG)

Address(es): USB0.PIPECFG 000A 0068h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TYPE[1:0]		—	—	—	BFRE	DBLB	—	SHTNAK	—	—	DIR	EPNUM[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	EPNUM[3:0]	Endpoint Number*1	These bits specify the endpoint number for the selected pipe. Setting 0000b means an unused pipe.	R/W
b4	DIR	Transfer Direction*2,*3	0: Receiving direction 1: Transmitting direction	R/W
b6, b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	SHTNAK	Pipe Disabled at End of Transfer*1	0: Pipe assignment continued at the end of transfer 1: Pipe assignment disabled at the end of transfer	R/W
b8	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b9	DBLB	Double Buffer Mode*2,*3	0: Single buffer 1: Double buffer	R/W
b10	BFRE	BRDY Interrupt Operation Specification*2,*3	0: BRDY interrupt upon transmitting or receiving data 1: BRDY interrupt upon completion of reading data	R/W
b13 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15, b14	TYPE[1:0]	Transfer Type*1	• Pipes 1 and 2 b15 b14 0 0: Pipe not used 0 1: Bulk transfer 1 0: Setting prohibited 1 1: Isochronous transfer • Pipes 3 to 5 b15 b14 0 0: Pipe not used 0 1: Bulk transfer 1 0: Setting prohibited 1 1: Setting prohibited • Pipes 6 to 9 b15 b14 0 0: Pipe not used 0 1: Setting prohibited 1 0: Interrupt transfer 1 1: Setting prohibited	R/W

Note 1. Modify the TYPE[1:0], SHTNAK, and EPNUM[3:0] bits while the PIPEnCTR.PID[1:0] bits are 00b (NAK). Before modifying these bits after modifying the PIPEnCTR.PID[1:0] bits for the selected pipe from 01b (BUF) to 00b (NAK), check that the PIPEnCTR.PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.

Note 2. Modify the BFRE, DBLB, and DIR bits while the PIPEnCTR.PID[1:0] bits are 00b (NAK) and before the pipe is selected by the CURPIPE[3:0] bits in the port select register. Before modifying these bits after modifying the PIPEnCTR.PID[1:0] bits for the selected pipe from 01b (BUF) to 00b (NAK), check that the PIPEnCTR.PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), checking the PIPEnCTR.PBUSY flag through software is not necessary.

Note 3. To modify the BFRE, DBLB, and DIR bits after completing USB communication using the selected pipe, write 1 and then 0 to the PIPEnCTR.ACLRM bit continuously through software to clear the FIFO buffer assigned to the selected pipe while the PID[1:0] bits and CURPIPE[3:0] bits are in the state described in the above note 2.

The PIPECFG register specifies the transfer type, buffer memory access direction, and endpoint numbers for pipes 1 to 9. It also selects single or double buffer mode, and whether to continue or disable pipe operation at the end of transfer.

Downloaded from Arrow.com.

EPNUM[3:0] Bits (Endpoint Number)

The EPNUM[3:0] bits specify the endpoint number for the selected pipe.
Setting 0000b means an unused pipe.
Do not make the settings such that the combination of the settings of the DIR and EPNUM[3:0] bits should be the same for two or more pipes (EPNUM[3:0] bits = 0000b can be set for all of the pipes).

DIR Bit (Transfer Direction)

The DIR bit specifies the transfer direction for the selected pipe.
When the DIR bit has been set to 0 by software, the USB uses the selected pipe in the receiving direction, and when software has set the DIR bit to 1, the USB uses the selected pipe in the transmitting direction.

SHTNAK Bit (Pipe Disabled at End of Transfer)

The SHTNAK bit specifies whether to modify the PID[1:0] bits to 00b (NAK) upon the end of transfer when the selected pipe is in the receiving direction.
The SHTNAK bit is valid when the selected pipe is pipe 1 to 5 in the receiving direction.
When the SHTNAK bit has been set to 1 by software for the selected pipe in the receiving direction, the USB modifies the PIPEnCTR.PID[1:0] bits corresponding to the selected pipe to 00b (NAK) on determining the end of the transfer. The USB determines that the transfer has ended on any of the following conditions.

- A short packet (including a zero-length packet) is successfully received.
- The transaction counter is used and the number of packets specified by the counter are successfully received.

DBLB Bit (Double Buffer Mode)

The DBLB bit selects either single or double buffer mode for the FIFO buffer used by the selected pipe.
The DBLB bit is valid when the selected pipe is pipe 1 to 5.

BFRE Bit (BRDY Interrupt Operation Specification)

The BFRE bit specifies the BRDY interrupt generation timing from the USB to the CPU with respect to the selected pipe.
When the BFRE bit has been set to 1 by software and the selected pipe is in the receiving direction, the USB detects the transfer completion and generates the BRDY interrupt on having read the relevant packet.
When the BRDY interrupt is generated with the above conditions, 1 should be written to the BCLR bit in the port control register by software. The FIFO buffer assigned to the selected pipe is not enabled for reception until 1 is written to the BCLR bit.
When the BFRE bit has been set to 1 by software and the selected pipe is in the transmitting direction, the USB does not generate the BRDY interrupt.
For details, refer to section 32.3.3.1, BRDY Interrupt.

TYPE[1:0] Bits (Transfer Type)

The TYPE[1:0] bits select the transfer type for the pipe selected by the PIPESEL.PIPESEL[3:0] bits (selected pipe).
Before setting the PID[1:0] bits to 01b (BUF) for the selected pipe (before starting USB communication using the selected pipe), set the TYPE[1:0] bits to a value other than 00b.

Downloaded from Arrow.com.

32.2.28 Pipe Maximum Packet Size Register (PIPEMAXP)

Address(es): USB0.PIPEMAXP 000A 006Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DEVSEL[3:0]				—	—	—	MXPS[8:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0/1 *1	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b8 to b0	MXPS[8:0]	Maximum Packet Size*2	• Pipes 1 and 2: 1 byte (001h) to 256 bytes (100h) • Pipes 3 to 5: 8 bytes (008h), 16 bytes (010h), 32 bytes (020h), 64 bytes (040h) • Pipes 6 to 9: 1 byte (001h) to 64 bytes (040h)	R/W
b11 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b12	DEVSEL[3:0]	Device Select*3	b3 b0 0 0 0 0: Address 0000 0 0 0 1: Address 0001 0 0 1 0: Address 0010 0 0 1 1: Address 0011 0 1 0 0: Address 0100 0 1 0 1: Address 0101 Settings other than above are prohibited.	R/W

Note 1. The value of these bits is 0000h when no pipe is selected with the PIPESEL.PIPESEL[3:0] bits and 0040h when a pipe is selected.

Note 2. Modify the MXPS[8:0] bits while the PIPEnCTR.PID[1:0] bits are 00b (NAK) and before the pipe is selected by the CURPIPE[3:0] bits in the port select register. Before modifying these bits after modifying the PIPEnCTR.PID[1:0] bits for the selected pipe from 01b (BUF) to 00b (NAK), check that the PIPEnCTR.PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.

Note 3. Modify the DEVSEL[3:0] bits while the PIPEnCTR.PID[1:0] bits are 00b (NAK). To modify these bits after modifying the PIPEnCTR.PID[1:0] bits for the selected pipe from 01b (BUF) to 00b (NAK), check that the PIPEnCTR.PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.

The PIPEMAXP register specifies the maximum packet size for pipes 1 to 9.

MXPS[8:0] Bits (Maximum Packet Size)

The MXPS[8:0] bits specify the maximum data payload (maximum packet size) for the selected pipe.

These bits should be set to the appropriate value for each transfer type based on USB Specification 2.0. Note that the maximum value of pipes 1 and 2 is 256. While MXPS[8:0] = 000h, do not write to the FIFO buffer or do not set the PID[1:0] bits to 01b (BUF).

DEVSEL[3:0] Bits (Device Select)

When the host controller is selected, these bits specify the USB device address of the peripheral device which is the communication target.

The DEVSEL[3:0] bits should be set after setting the address to the DEVADDn (n = 0 to 5) register corresponding to the value to be set in the DEVSEL[3:0] bits. For example, before setting the DEVSEL[3:0] bits to 0010b, the address should be set to DEVADD2.

When the function controller is selected, the DEVSEL[3:0] bits should be set to 0000b.

Downloaded from Arrow.com.

32.2.29 Pipe Cycle Control Register (PIPEPERI)

Address(es): USB0.PIPEPERI 000A 006Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	IFIS	—	—	—	—	—	—	—	—	—	IITV[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	IITV[2:0]	Interval Error Detection Interval *1	Specify the interval error detection timing for the selected pipe in terms of frames, which is expressed as nth power of 2.	R/W
b11 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12	IFIS	Isochronous IN Buffer Flush	0: The buffer is not flushed. 1: The buffer is flushed.	R/W
b15 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Modify the IITV[2:0] bits while the PIPEnCTR.PID[1:0] bits are 00b (NAK). To modify these bits after modifying the PIPEnCTR.PID[1:0] bits for the selected pipe from 01b (BUF) to 00b (NAK), check that the PIPEnCTR.PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.

The PIPEPERI register selects whether the buffer is flushed or not when an interval error occurred during isochronous IN transfer, and sets the interval error detection interval for pipes 1 to 9.

IITV[2:0] Bits (Interval Error Detection Interval)

Before modifying the IITV[2:0] bits after USB communication has been completed with the IITV[2:0] bits set to a certain value, set the PID[1:0] bits to 00b (NAK) and then set the PIPEnCTR.ACLRM bit to 1 to initialize the interval timer.

The IITV[2:0] bits are invalid for pipes 3 to 5; set the IITV[2:0] bits to 000b for pipes 3 to 5.

IFIS Bit (Isochronous IN Buffer Flush)

Specifies whether to flush the buffer when the pipe selected by the PIPESEL.PIPESEL[3:0] bits (selected pipe) is used for isochronous IN transfers.

When the function controller is selected and the selected pipe is for isochronous IN transfers, the USB automatically clears the FIFO buffer when the USB fails to receive the IN token from the USB host within the interval set by the IITV[2:0] bits in terms of frames.

In double buffer mode (the PIPECFG.DBLB bit = 1), the USB only clears the data in the plane used earlier.

The USB clears the FIFO buffer on receiving the SOF packet immediately after the frame in which the USB has expected to receive the IN token. Even if the SOF packet is damaged, the USB also clears the FIFO buffer at the right timing to receive the SOF packet by using the internal interpolation function.

When the host controller is selected, set the IITV[2:0] bits to 000b.

When the selected pipe is not for isochronous transfer, set the IITV[2:0] bits to 000b.

Downloaded from Arrow.com.

32.2.30 Pipe n Control Registers (PIPEnCTR) (n = 1 to 9)

- PIPEnCTR (n = 1 to 5)

Address(es): USB0.PIPE1CTR 000A 0070h, USB0.PIPE2CTR 000A 0072h, USB0.PIPE3CTR 000A 0074h, USB0.PIPE4CTR 000A 0076h, USB0.PIPE5CTR 000A 0078h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	BSTS	INBUFM	—	—	—	ATREPM	ACLRM	SQCLR	SQSET	SQMON	PBUSY	—	—	—	PID[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	PID[1:0]	Response PID	b1 b0 0 0: NAK response 0 1: BUF response (depending on the buffer state) 1 0: STALL response 1 1: STALL response	R/W
b4 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5	PBUSY	Pipe Busy Flag	0: The relevant pipe is not used for the transaction. 1: The relevant pipe is used for the transaction.	R
b6	SQMON	Sequence Toggle Bit Confirmation	0: DATA0 1: DATA1	R
b7	SQSET	Sequence Toggle Bit Set*2	0: Write disabled 1: Specifies DATA1.	R/W*1
b8	SQCLR	Sequence Toggle Bit Clear*2	0: Write disabled 1: Specifies DATA0.	R/W*1
b9	ACLRM	Auto Buffer Clear Mode*3	0: Disabled 1: Enabled (all buffers are initialized)	R/W
b10	ATREPM	Auto Response Mode*2	0: Auto response is disabled. 1: Auto response is enabled.	R/W
b13 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b14	INBUFM	Transmit Buffer Monitor Flag	0: There is no data to be transmitted in the buffer memory. 1: There is data to be transmitted in the buffer memory.	R
b15	BSTS	Buffer Status Flag	0: Buffer access by the CPU is disabled. 1: Buffer access by the CPU is enabled.	R

Note 1. Only 0 can be read.
Note 2. Modify the ATREPM bit or write 1 to the SQCLR or SQSET bit while the PID[1:0] bits are 00b (NAK). Before modifying these bits after modifying the PID[1:0] bits for the selected pipe from 01b (BUF) to 00b (NAK), check that the PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.
Note 3. Modify the ACLRM bit while PID[1:0] bits are 00b (NAK) and before the pipe is selected by the CURPIPE[3:0] bits in the port select register. Before modifying this bit after modifying the PID[1:0] bits for the selected pipe from 01b (BUF) to 00b (NAK), check that the PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.

The PIPEnCTR register can be set regardless of the pipe selection in the PIPESEL register.

Downloaded from Arrow.com.

PID[1:0] Bits (Response PID)

The PID[1:0] bits specify the response type for the next transaction of the relevant pipe.
The default setting of the PID[1:0] bits are 00b (NAK). Modify the setting of the PID[1:0] bits to 01b (BUF) to use the relevant pipe for USB transfer. Table 32.6 and Table 32.7 show the basic operation (operation when there are no errors in the transmitted and received packets) of the USB depending on the PID[1:0] bit setting.
After modifying the setting of the PID[1:0] bits through software from 01b (BUF) to 00b (NAK) during USB communication using the relevant pipe, check that the PBUSY flag is 1 to see if USB transfer using the relevant pipe has actually entered the NAK state. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.
The USB modifies the setting of the PID[1:0] bits in the following cases.

- The USB sets the PID[1:0] bits to 00b (NAK) on recognizing the completion of the transfer when the relevant pipe is in the receiving direction and the PIPECFG.SHTNAK bit for the selected pipe has been set to 1 by software.
- The USB sets the PID[1:0] bits to 11b (STALL) on receiving a data packet with a payload exceeding the maximum packet size of the relevant pipe.
- The USB sets the PID[1:0] bits to 00b (NAK) on detecting a USB bus reset when the function controller is selected.
- The USB sets the PID[1:0] bits to 00b (NAK) on detecting a receive error, such as a CRC error, three consecutive times when the host controller is selected.
- The USB sets the PID[1:0] bits to 11b (STALL) on receiving the STALL handshake when the host controller is selected.

To specify each response type, set the PID[1:0] bits as follows.

- To make a transition from NAK (00b) to STALL, set 10b.
- To make a transition from BUF (01b) to STALL, set 11b.
- To make a transition from STALL (11b) to NAK, set 10b and then 00b.
- To make a transition from STALL (11b) to BUF, set 10b, 00b, and then 01b.
- To make a transition from STALL (10b) to BUF, set 00b and then 01b.

PBUSY Flag (Pipe Busy Flag)

The PBUSY flag indicates whether the relevant pipe is being currently used or not for the transaction.
The USB modifies the PBUSY flag from 0 to 1 upon start of the USB transaction for the relevant pipe, and modifies the PBUSY flag from 1 to 0 upon completion of one transaction.
Reading the PBUSY flag after the PID[1:0] bits have been set to 00b (NAK) by software allows checking whether modification of the pipe settings is possible.
For details, refer to section 32.3.4.1, Pipe Control Register Switching Procedures.

SQMON Bit (Sequence Toggle Bit Confirmation)

The SQMON flag indicates the expected value of the sequence toggle bit for the next transaction of the relevant pipe.
When the relevant pipe is not for the isochronous transfer, the USB allows the SQMON flag to toggle upon normal completion of the transaction. However, the SQMON flag is not allowed to toggle when a data PID mismatch occurs during the transfer in the receiving direction.

SQSET Bit (Sequence Toggle Bit Set)

The SQSET bit should be set to 1 to set DATA1 as the expected value of the sequence toggle bit for the next transaction of the relevant pipe.
Setting the SQSET bit to 1 through software allows the USB to set DATA1 as the expected value of the sequence toggle bit of the relevant pipe. The USB sets the SQSET bit to 0.

Downloaded from Arrow.com.

SQCLR Bit (Sequence Toggle Bit Clear)

The SQCLR bit should be set to 1 to clear the expected value (to set DATA0 as the expected value) of the sequence toggle bit for the next transaction of the relevant pipe.
Setting the SQCLR bit to 1 through software allows the USB to set DATA0 as the expected value of the sequence toggle bit of the relevant pipe. The USB sets the SQCLR bit to 0.

ACLRM Bit (Auto Buffer Clear Mode)

Enables or disables auto buffer clear mode for the relevant pipe.
To delete the information in the FIFO buffer assigned to the relevant pipe completely, write 1 and then 0 to the ACLRM bit continuously.
Table 32.8 shows the information cleared by writing 1 and 0 to the ACLRM bit continuously and the cases in which clearing the information is necessary.

ATREPM Bit (Auto Response Mode)

Enables or disables auto response mode for the relevant pipe.
When the function controller is selected and the relevant pipe is for bulk transfer, the ATREPM bit can be set to 1.
When the ATREPM bit is set to 1, the USB responds to the token from the USB host as described below.
(1) When the relevant pipe is for bulk IN transfer (the PIPECFG.TYPE[1:0] bits = 01b and the PIPECFG.DIR bit = 1)
When the ATREPM bit = 1 and PID[1:0] = 01b (BUF), the USB transmits a zero-length packet in response to the IN token.
The USB updates (allows toggling of) the sequence toggle bit (data PID) each time the USB receives ACK from the USB host (in a single transaction, IN token is received, zero-length packet is transmitted, and then ACK is received.).
In this case, the USB does not generate the BRDY or BEMP interrupt.
(2) When the relevant pipe is for bulk OUT transfer (the PIPECFG.TYPE[1:0] bits = 01b and the PIPECFG.DIR bit = 0)
When the ATREPM bit = 1 and PID[1:0] = 01b (BUF), the USB returns NAK in response to the OUT token and generates the NRDY interrupt.
For USB communication in auto response mode, set the ATREPM bit to 1 while the FIFO buffer is empty. Do not write to the FIFO buffer during USB communication in auto response mode.
When the relevant pipe is for isochronous transfer, be sure to set the ATREPM bit to 0.
When the host controller is selected, be sure to set the ATREPM bit to 0.

INBUFM Flag (Transmit Buffer Monitor Flag)

Indicates the relevant FIFO buffer status when the relevant pipe is in the transmitting direction.
When the relevant pipe is in the transmitting direction (the PIPECFG.DIR bit = 1), the USB sets the INBUFM flag to 1 when the CPU or DMAC/DTC completes writing data to at least one FIFO buffer plane.
The USB sets the INBUFM flag to 0 when the USB completes transmitting the data from the FIFO buffer plane to which all the data has been written. In double buffer mode (the PIPECFG.DBLB bit = 1), the USB sets the INBUFM flag to 0 when the USB completes transmitting the data from the two FIFO buffer planes before the CPU or DMAC/DTC completes writing data to one FIFO buffer plane.
The INBUFM flag indicates the same value as the BSTS flag when the relevant pipe is in the receiving direction (the PIPECFG.DIR bit = 0).

Downloaded from Arrow.com.

BSTS Flag (Buffer Status Flag)

Indicates the FIFO buffer status for the relevant pipe.

The meaning of the BSTS flag depends on the settings of the PIPECFG.DIR bit, PIPECFG.BFRE bit, and DnFIFOSEL.DCLRM bits as shown in Table 32.9.

Table 32.6 Operation of USB depending on PID[1:0] Bits Setting (When Host Controller is Selected)

PID[1:0] Bits	Transfer Type	Transfer Direction (DIR Bit)	Operation of USB
00b (NAK)	Operation does not depend on the setting.	Operation does not depend on the setting.	Does not issue tokens.
01b (BUF)	Bulk or interrupt	Operation does not depend on the setting.	Issues tokens while the DVSTCTR0.UACT bit is 1 and the FIFO buffer corresponding to the relevant pipe is ready for transmission and reception. Does not issue tokens while the DVSTCTR0.UACT bit is 0 or the FIFO buffer corresponding to the relevant pipe is not ready for transmission or reception.
	Isochronous	Operation does not depend on the setting.	Issues tokens irrespective of the status of the FIFO buffer corresponding to the relevant pipe.
10b (STALL) or 11b (STALL)	Operation does not depend on the setting.	Operation does not depend on the setting.	Does not issue tokens.

Table 32.7 Operation of USB depending on PID[1:0] Bits Setting (When Function Controller is Selected)

PID[1:0] Bits	Transfer Type	Transfer Direction (DIR Bit)	Operation of USB
00b (NAK)	Bulk or interrupt	Operation does not depend on the setting.	Returns NAK in response to the token from the USB host.
	Isochronous	Operation does not depend on the setting.	Returns nothing in response to the token from the USB host.
01b (BUF)	Bulk	Receiving direction (DIR bit = 0)	Receives data and returns ACK in response to the OUT token from the USB host if the FIFO buffer corresponding to the relevant pipe is ready for reception.
	Interrupt	Receiving direction (DIR bit = 0)	Receives data and returns ACK in response to the OUT token from the USB host if the FIFO buffer corresponding to the relevant pipe is ready for reception.
	Bulk or interrupt	Transmitting direction (DIR bit = 1)	Transmits data in response to the token from the USB host if the corresponding FIFO buffer is ready for transmission. Returns NAK if not ready.
	Isochronous	Receiving direction (DIR bit = 0)	Receives data in response to the OUT token from the USB host if the FIFO buffer corresponding to the relevant pipe is ready for reception. Discards data if not ready.
	Isochronous	Transmitting direction (DIR bit = 1)	Transmits data in response to the token from the USB host if the corresponding FIFO buffer is ready for transmission. Transmits the zero-length packet if not ready.
10b (STALL) or 11b (STALL)	Bulk or interrupt	Operation does not depend on the setting.	Returns STALL in response to the token from the USB host.
	Isochronous	Operation does not depend on the setting.	Returns nothing in response to the token from the USB host.

Downloaded from Arrow.com.

Table 32.8 Information Cleared by USB by Setting ACLRM = 1

No.	Information Cleared by ACLRM Bit Manipulation	Cases in which Clearing Information is Necessary
1	All the information in the FIFO buffer assigned to the relevant pipe (both FIFO buffer planes are cleared when double buffer mode is selected)	When the pipe is to be initialized
2	The interval count value when the relevant pipe is for isochronous transfer	When the interval count value is to be reset
3	Internal flags concerning the PIPECFG.BFRE bit	When the PIPECFG.BFRE setting is modified
4	FIFO buffer toggle control	When the PIPECFG.DBLB setting is modified
5	Internal flags concerning the transaction count	When the transaction count function is forcibly terminated

Table 32.9 Operation of BSTS Flag

DIR Bit	BFRE Bit	DCLRM Bit	BSTS Flag Function
0	0	0	The received data can be read from the FIFO buffer. The received data has been completely read from the FIFO buffer.
		1	Setting prohibited
	1	0	The received data can be read from the FIFO buffer. The BCLR bit in the port control register has been set to 1 by software after the received data has been completely read from the FIFO buffer.
		1	The received data can be read from the FIFO buffer. The received data has been completely read from the FIFO buffer.
1	0	0	The transmit data can be written to the FIFO buffer. The transmit data has been completely written to the FIFO buffer.
		1	Setting prohibited
	1	0	Setting prohibited
		1	Setting prohibited

Downloaded from Arrow.com.

- PIPEnCTR (n = 6 to 9)

Address(es): USB0.PIPE6CTR 000A 007Ah, USB0.PIPE7CTR 000A 007Ch, USB0.PIPE8CTR 000A 007Eh, USB0.PIPE9CTR 000A 0080h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	BSTS	—	—	—	—	—	ACLRM	SQCLR	SQSET	SQMON	PBUSY	—	—	—	PID[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	PID[1:0]	Response PID	b1 b0 0 0: NAK response 0 1: BUF response (depending on the buffer state) 1 0: STALL response 1 1: STALL response	R/W
b4 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5	PBUSY	Pipe Busy Flag	0: The relevant pipe is not used at the USB bus. 1: The relevant pipe is used at the USB bus.	R
b6	SQMON	Sequence Toggle Bit Confirmation Flag	0: DATA0 1: DATA1	R
b7	SQSET	Sequence Toggle Bit Set*2	0: Invalid 1: Specifies DATA1.	R/W *1
b8	SQCLR	Sequence Toggle Bit Clear *2	0: Invalid 1: Specifies DATA0.	R/W *1
b9	ACLRM	Auto Buffer Clear Mode*2,*3	0: Auto buffer clear mode is disabled. 1: Auto buffer clear mode is enabled (all buffers are initialized)	R/W
b14 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	BSTS	Buffer Status Flag	0: Buffer access is disabled. 1: Buffer access is enabled.	R

Note 1. Only 0 can be read. Only 1 can be written.
Note 2. Write 1 to the SQCLR or SQSET bit while the PID[1:0] bits are 00b (NAK). Before modifying these bits after modifying the PID[1:0] bits for the selected pipe from 01b (BUF) to 00b (NAK), check that the PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.
Note 3. Modify the ACLRM bit while the PID[1:0] bits are 00b (NAK) and before the pipe is selected by the CURPIPE[3:0] bits in the port select register. Before modifying this bit after modifying the PID[1:0] bits for the selected pipe from 01b (BUF) to 00b (NAK), check that the PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.

Downloaded from Arrow.com.

PID[1:0] Bits (Response PID)

The PID[1:0] bits specify the response type for the next transaction of the relevant pipe.
The default setting of the PID[1:0] bits are 00b (NAK). Modify the setting of the PID[1:0] bits to 01b (BUF) to use the relevant pipe for USB transfer. Table 32.6 and Table 32.7 show the basic operation (operation when there are no errors in the transmitted and received packets) of the USB depending on the setting of the PID[1:0] bits.
After modifying the setting of the PID[1:0] bits through software from 01b (BUF) to 00b (NAK) during USB communication using the relevant pipe, check that the PBUSY flag is 1 to see if USB transfer using the relevant pipe has actually entered the NAK state. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.
The USB modifies the setting of the PID[1:0] bits in the following cases.

- The USB sets the PID[1:0] bits to 11b (STALL) on receiving a data packet with a payload exceeding the maximum packet size of the relevant pipe.
- The USB sets the PID[1:0] bits to 00b (NAK) on detecting a USB bus reset when the function controller is selected.
- The USB sets the PID[1:0] bits to 00b (NAK) on detecting a receive error, such as a CRC error, three consecutive times when the host controller is selected.
- The USB sets the PID[1:0] bits to 11b (STALL) on receiving the STALL handshake when the host controller is selected.

To specify each response type, set the PID[1:0] bits as follows.

- To make a transition from NAK (00b) to STALL, set 10b.
- To make a transition from BUF (01b) to STALL, set 11b.
- To make a transition from STALL (11b) to NAK, set 10b and then 00b.
- To make a transition from STALL (11b) to BUF, set 10b, 00b, and then 01b.
- To make a transition from STALL (10b) to BUF, set 00b and then 01b.

PBUSY Flag (Pipe Busy Flag)

The PBUSY flag indicates whether the relevant pipe is being currently used or not for the transaction.
The USB modifies the PBUSY flag from 0 to 1 upon start of the USB transaction for the relevant pipe, and modifies the PBUSY flag from 1 to 0 upon completion of one transaction.
Reading the PBUSY flag after the PID[1:0] bits has been set to 00b (NAK) by software allows checking whether modification of the pipe settings is possible.

SQMON Flag (Sequence Toggle Bit Confirmation Flag)

The SQMON flag indicates the expected value of the sequence toggle bit for the next transaction of the relevant pipe.
The USB allows the SQMON flag to toggle upon normal completion of the transaction. However, the SQMON flag is not allowed to toggle when a data PID mismatch occurs during the transfer in the receiving direction.

SQSET Bit (Sequence Toggle Bit Set)

The SQSET bit should be set to 1 to set DATA1 as the expected value of the sequence toggle bit for the next transaction of the relevant pipe.
Setting the SQSET bit through software allows the USB to set DATA1 as the expected value of the sequence toggle bit of the relevant pipe. The USB sets the SQSET bit to 0.

SQCLR Bit (Sequence Toggle Bit Clear)

The SQCLR bit should be set to 1 to clear the expected value (to set DATA0 as the expected value) of the sequence toggle bit for the next transaction of the relevant pipe.
Setting the SQCLR bit to 1 through software allows the USB to set DATA0 as the expected value of the sequence toggle bit of the relevant pipe. The USB sets the SQCLR bit to 0.

Downloaded from Arrow.com.

ACLRM Bit (Auto Buffer Clear Mode)

Enables or disables auto buffer clear mode for the relevant pipe.

To delete the information in the FIFO buffer assigned to the relevant pipe completely, write 1 and then 0 to the ACLRM bit continuously.

Table 32.10 shows the information cleared by writing 1 and 0 to the ACLRM bit continuously and the cases in which clearing the information is necessary.

BSTS Flag (Buffer Status Flag)

Indicates the FIFO buffer status for the relevant pipe.

The meaning of the BSTS flag depends on the settings of the PIPECFG.DIR bit, PIPECFG.BFRE bit, and DnFIFOSEL.DCLRM bits as shown in Table 32.9.

Table 32.10 Information Cleared by USB by Setting the ACLRM Bit = 1

No.	Information Cleared by ACLRM Bit Manipulation	Cases in which Clearing Information is Necessary
1	All the information in the FIFO buffer assigned to the selected pipe	When the pipe is to be initialized
2	The interval count value when the selected pipe is for interrupt transfer and the host controller is selected	When the interval count value is to be reset
3	Internal flags concerning the PIPECFG.BFRE bit	When the PIPECFG.BFRE setting is modified
4	Internal flags concerning the transaction count	When the transaction count function is forcibly terminated

Downloaded from Arrow.com.

32.2.31 Pipe n Transaction Counter Enable Register (PIPEnTRE) (n = 1 to 5)

Address(es): USB0.PIPE1TRE 000A 0090h, USB0.PIPE2TRE 000A 0094h, USB0.PIPE3TRE 000A 0098h, USB0.PIPE4TRE 000A 009Ch, USB0.PIPE5TRE 000A 00A0h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	TRENB	TRCLR	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	TRCLR	Transaction Counter Clear	0: Invalid 1: The current counter value is cleared.	R/W
b9	TRENB	Transaction Counter Enable	0: Transaction counter is disabled. 1: Transaction counter is enabled.	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Modify each bit in the PIPEnTRE register while the PID[1:0] bits are 00b (NAK). Before modifying these bits after modifying the PIPEnCTR.PID[1:0] bits for the selected pipe from 01b (BUF) to 00b (NAK), check that the PIPEnCTR.PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.

TRCLR Bit (Transaction Counter Clear)

Clears the current value of the transaction counter corresponding to the relevant pipe and then sets the TRCLR bit to 0.

TRENB Bit (Transaction Counter Enable)

Enables or disables the transaction counter.

For the pipe in the receiving direction, setting the TRENB bit to 1 after setting the total number of the packets to be received in the PIPEnTRN register through software allows the USB to control hardware as described below on having received the number of packets equal to the setting of the PIPEnTRN register.

- While the PIPECFG.SHTNAK bit is 1, the USB modifies the PID[1:0] bits to 00b (NAK) for the corresponding pipe on having received the number of packets equal to the setting of the PIPEnTRN register.
- While the PIPECFG.BFRE bit is 1, the USB asserts the BRDY interrupt on having received the number of packets equal to the setting of the PIPEnTRN register and then reading the last received data.

For the pipe in the transmitting direction, set the TRENB bit to 0.

When the transaction counter is not used, set the TRENB bit to 0.

When the transaction counter is used, set the PIPEnTRN register before setting the TRENB bit to 1. Set the TRENB bit to 1 before receiving the first packet to be counted by the transaction counter.

Downloaded from Arrow.com.

32.2.32 Pipe n Transaction Counter Register (PIPEnTRN) (n = 1 to 5)

Address(es): USB0.PIPE1TRN 000A 0092h, USB0.PIPE2TRN 000A 0096h, USB0.PIPE3TRN 000A 009Ah, USB0.PIPE4TRN 000A 009Eh, USB0.PIPE5TRN 000A 00A2h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The PIPEnTRN register retains the setting by a USB bus reset.

- When written to:
  This register specifies the total of packets (number of transactions) to be received in corresponding pipe.
- When read from:
  This register indicates the specified number of transactions if the PIPEnTRE.TRENB bit is 0.
  This register indicates the number of currently counted transactions if the PIPEnTRE.TRENB bit is 1.

The USB increments the value of the PIPEnTRN register by one when all of the following conditions are satisfied on receiving the packet.

- The PIPEnTRE.TRENB bit = 1
- (PIPEnTRN set value ≠ current counter value + 1) on receiving the packet.
- The payload of the received packet agrees with the setting of the PIPEMAXP.MXPS[8:0] bits.

The USB sets the value of the PIPEnTRN register to 0000h when any of the following conditions are satisfied.

(1) All of the following conditions are satisfied.
- The PIPEnTRE.TRENB bit = 1
- (PIPEnTRN set value = current counter value + 1) on receiving the packet.
- The payload of the received packet agrees with the setting of the PIPEMAXP.MXPS[8:0] bits.

(2) All of the following conditions are satisfied.
- The PIPEnTRE.TRENB bit = 1
- The USB has received a short packet.

(3) All of the following conditions are satisfied.
- The PIPEnTRE.TRENB bit = 1
- The PIPEnTRE.TRCLR bit has been set to 1 by software.

For the pipe in the transmitting direction, set the PIPEnTRN register to 0000h.
When the transaction counter is not used, set the PIPEnTRN register to 0000h.
Setting the number of transactions to be transferred to the PIPEnTRN register is only enabled when the PIPEnTRE.TRENB bit is 0. To modify the number of transactions to be transferred, set the TRCLR bit to 1 (to clear the current counter value) before setting the PIPEnTRE.TRENB bit to 1.

Downloaded from Arrow.com.

32.2.33 Device Address n Configuration Register (DEVADDn) (n = 0 to 5)

Address(es): USB0.DEVADD0 000A 00D0h, USB0.DEVADD1 000A 00D2h, USB0.DEVADD2 000A 00D4h, USB0.DEVADD3 000A 00D6h, USB0.DEVADD4 000A 00D8h, USB0.DEVADD5 000A 00DAh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	USBSPD[1:0]		—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7, b6	USBSPD[1:0]	Transfer Speed of Communication Target Device	b7 b6 0 0: DEVADDn is not used 0 1: Low-speed 1 0: Full-speed 1 1: Setting prohibited	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The DEVADDn register specifies the transfer speed of the peripheral device which is the communication target for pipes 0 to 9.

When the host controller is selected, the bits in the DEVADDn register should be set before starting communication using each pipe.

The bits in the DEVADDn register should be modified while no valid pipes are using the settings of the bits. Valid pipes refer to the pipes satisfying both of the following conditions:

- The DEVADDn register is selected by the DEVSEL[3:0] bits.
- The PID[1:0] bits are set to 01b (BUF) for the selected pipe or the selected pipe is the DCP with the DCPCTR.SUREQ bit set to 1.

USBSPD[1:0] Bits (Transfer Speed of Communication Target Device)

The USBSPD[1:0] bits specify the USB transfer speed of the communication target peripheral device.

Set these bits to 01b when a low-speed device is connected, whereas set them to 10b when a full-speed device is connected.

When the host controller is selected, the USB refers to the setting of the USBSPD[1:0] bits to generate packets.

When the function controller is selected, set these bits to 00b.

Downloaded from Arrow.com.

32.2.34 USB Module Control Register (USBMC)

Address(es): USB0.USBMC 000A 00CCh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	VDDUSBE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit	Symbol	Bit Name	Description	R/W
b0	VDDUSBE	USB Power Supply Circuit On/Off Control	0: USB power supply circuit off 1: USB power supply circuit on	R/W
b1	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b15 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

VDDUSBE Bit (USB Power Supply Circuit On/Off Control)

The USB power supply circuit generates the reference voltage for battery charging. Set this bit to 1 when using the battery charging function.

Downloaded from Arrow.com.

32.2.35 BC Control Register 0 (USBBCCTRL0)

Address(es): USB0.USBBCCTRL0 000A 00B0h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PDDETSTS0	CHGDETSTS0	BATCHGE0	—	VDMSRCE0	IDPSINKE0	VDPSRCE0	IDMSINKE0	IDPSRCE0	RPDME0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RPDME0	D– Pin Pull-Down Control	0: Pull-down off 1: Pull-down on	R/W
b1	IDPSRCE0	D+ Pin IDPSRC Output Control	0: Stop 1: 10 µA output	R/W
b2	IDMSINKE0	D– Pin 0.6 V Input Detection (Comparator and Sink) Control	0: Detection off 1: Detection on (comparator and sink current on)	R/W
b3	VDPSRCE0	D+ Pin VDPSRC (0.6 V) Output Control	0: Stop 1: 0.6 V output	R/W
b4	IDPSINKE0	D+ Pin 0.6 V Input Detection (Comparator and Sink) Control	0: Detection off 1: Detection on (comparator and sink current on)	R/W
b5	VDMSRCE0	D– Pin VDMSRC (0.6 V) Output Control	0: Stop 1: 0.6 V output	R/W
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7	BATCHGE0	BC (Battery Charger) Function Ch0 General Enable Control	0: Disabled 1: Enabled	R/W
b8	CHGDETSTS0	D– Pin 0.6 V Input Detection Status Flag*1	0: Not detected 1: Detected	R
b9	PDDETSTS0	D+ Pin 0.6 V Input Detection Status Flag*2	0: Not detected 1: Detected	R
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Valid when IDMSINKE0 = 1.
Note 2. Valid when IDPSINKE0 = 1.

RPDME0 Bit (D– Pin Pull-Down Control)

When using the battery charging function, set this bit to 1 to control the pull-down resistor of the D– pin.

IDPSRCE0 Bit (D+ Pin IDPSRC Output Control)

With this bit set to 1, when the function controller is selected, current output is enabled upon detection of the connection of the data pin and the D+ pin is pulled up.

IDMSINKE0 Bit (D– Pin 0.6 V Input Detection (Comparator and Sink) Control)

With this bit set to 1, when the function controller is selected, the USB module detects whether VDMSRC (0.6 V) that is output from the host to D– upon primary detection is connected, or whether VDPSRC (0.6 V) that is output from the function to D+ is connected to the function's D– via the host.

VDPSRCE0 Bit (D+ Pin VDPSRC (0.6 V) Output Control)

With this bit set to 1, when the function controller is selected, output is enabled upon primary detection and VDPSRC (0.6 V) is applied to D+.

Downloaded from Arrow.com.

IDPSINKE0 Bit (D+ Pin 0.6 V Input Detection (Comparator and Sink) Control)

With this bit set to 1, when the function controller selected, the USB module detects whether VDMSRC (0.6 V) that is output from the function to D– is connected to the function's D+ (DCP) via the host. When the host controller is selected, the USB module detects whether VDPSRC (0.6 V) that is output from the function to D+ upon primary detection is connected.

VDMSRCE0 Bit (D– Pin VDMSRC (0.6 V) Output Control)

With this bit set to 1, when the function controller selected, output is enabled upon secondary detection and VDMSRC (0.6 V) is applied to D–. When the host controller is selected, output is enabled upon primary detection and VDMSRC (0.6 V) is applied to D–.

CHGDETSTS0 Flag (D– Pin 0.6 V Input Detection Status Flag)

When the host controller is selected, this flag is set to 1 if the USB module detects whether VDMSRC (0.6 V) that is output from the host to D– during primary detection is connected, or whether VDPSRC (0.6 V) that is output from the function to D+ is connected to the function's D– via the host.

PDDETSTS0 Flag (D+ Pin 0.6 V Input Detection Status Flag)

When the function controller is selected, this flag is set to 1 if the USB module detects whether VDMSRC (0.6 V) that is output from the function to D– during secondary detection is connected to the function's D+ (DCP) via the host.
When the host controller is selected, this bit is set to 1 if the USB module detects whether VDPSRC (0.6 V) that is output from the function to D+ during primary detection is connected.

Downloaded from Arrow.com.

32.3 Operation

32.3.1 System Control

This section describes the register settings that are necessary for initialization of this module and power consumption control.

32.3.1.1 Setting Data to the USB Related Register

Setting the SYSCFG.USBE bit to 1 after starting the clock supply to the USB (SYSCFG.SCKE bit = 1) enables and starts USB operation.

32.3.1.2 Controller Function Selection

For the USB, the host or function controller can be selected using the SYSCFG.DCFM bit. Note that the DCFM bit should be modified in the initial settings immediately after a reset is released or when pull-up resistors of the D+ and D– lines and pull-down resistors of the D+ and D– lines are disabled (the SYSCFG.DPRPU bit = 0, DMRPU bit = 0, and DRPD bit = 0).

32.3.1.3 Controlling USB Data Bus Resistors

The USB has pull-up and pull-down resistors for the D+ and D– lines. Pull up or pull down these lines by setting the SYSCFG.DPRPU, DMRPU, and DRPD bits.
When the function controller is selected, confirm that connection to the USB host is made, then set the DPRPU or DMRPU bit to 1 and pull up the D+ line (during full-speed) and D– line (during low-speed).
When the DPRPU or DMRPU bit is set to 0 during communication with the PC, the USB module disables the pull-up resistor of the USB data line, thus notifying the USB host of disconnection.
When the host controller is selected, set the DRPD bit and pull down the D+ and D– lines.

Table 32.11 Controlling USB Data Bus Resistors

Settings			Controlling USB Data Bus Resistors		
DRPD	**DPRPU**	**DMRPU**	**D–**	**D+**	**Remarks**
0	0	0	Open	Open	When not used
0	1	0	Open	Pull-up	When operating as the function controller (full-speed)
0	0	1	Pull-up	Open	When operating as the function controller (low-speed)
1	0	0	Pull-down	Pull-down	When operating as the host controller
Other than above			—	—	Setting prohibited

Downloaded from Arrow.com.

32.3.1.4 Example of USB External Connection Circuit

Figure 32.2 shows an example of connection between OTG device and USB connector in the self-powered state.
The USB controls the signals for enabling pull-up resistors for the D+ and D– signals and pull-down resistors for the D+ and D– signals. These signals can be pulled up or down using the SYSCFG.DPRPU, DMRPU, and DRPD bits.
When the function controller is selected and the DPRPU or DMRPU bit is set to 0 during communication with the host controller, the pull-up resistor of the USB data line is disabled, making it possible to notify the USB host of the device disconnection.

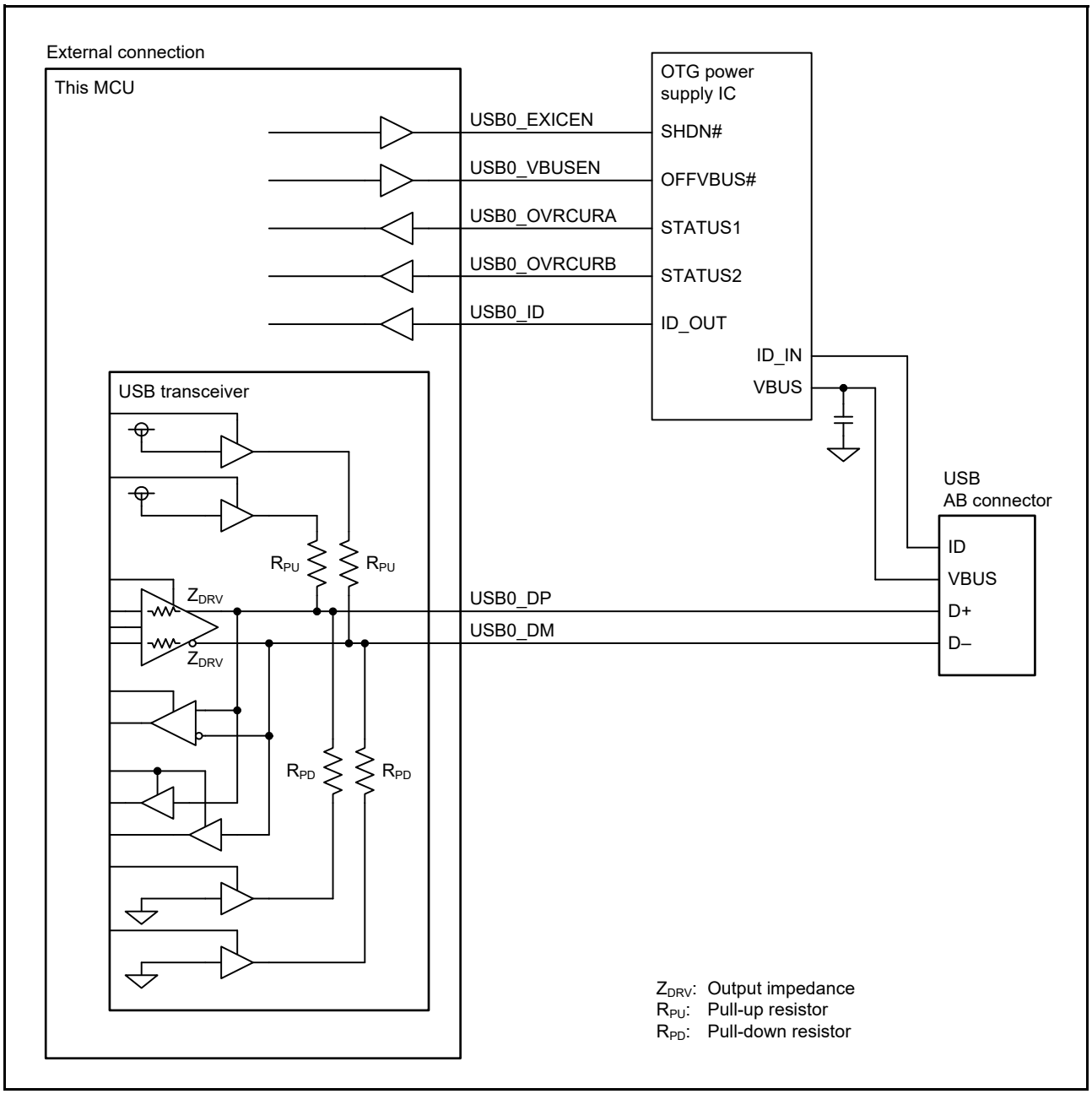


Figure 32.2 Example of Connection between Self-Powered OTG Device and USB Connector

Downloaded from Arrow.com.

Figure 32.3 shows an example of connection between function controller and USB connector in self-powered state.

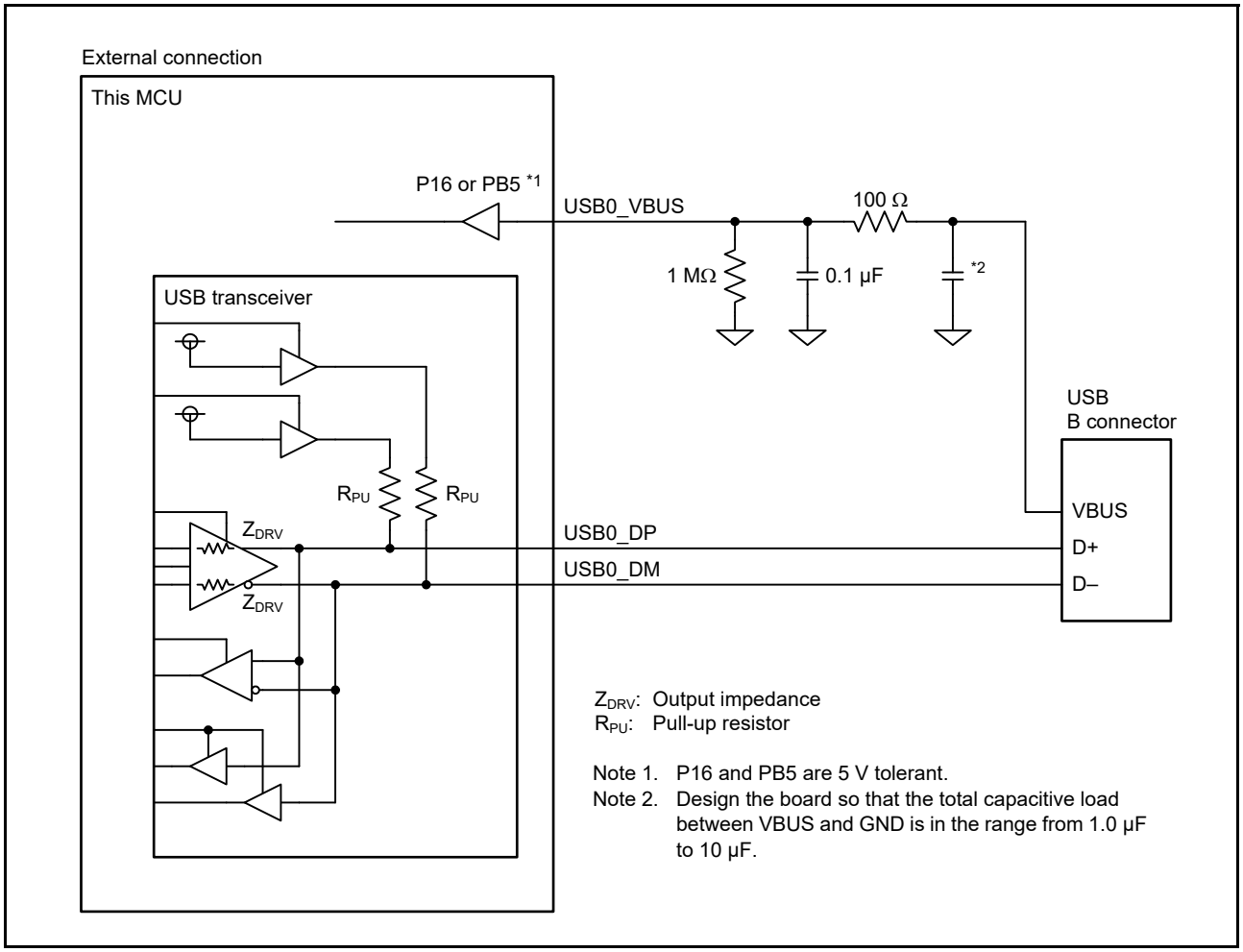


Figure 32.3 Example of Connection between Self-Powered Function and USB Connector

Downloaded from Arrow.com.

Figure 32.4 shows an example of connection between function controller and USB connector when Battery Charging Specification Revision 1.2 is supported.

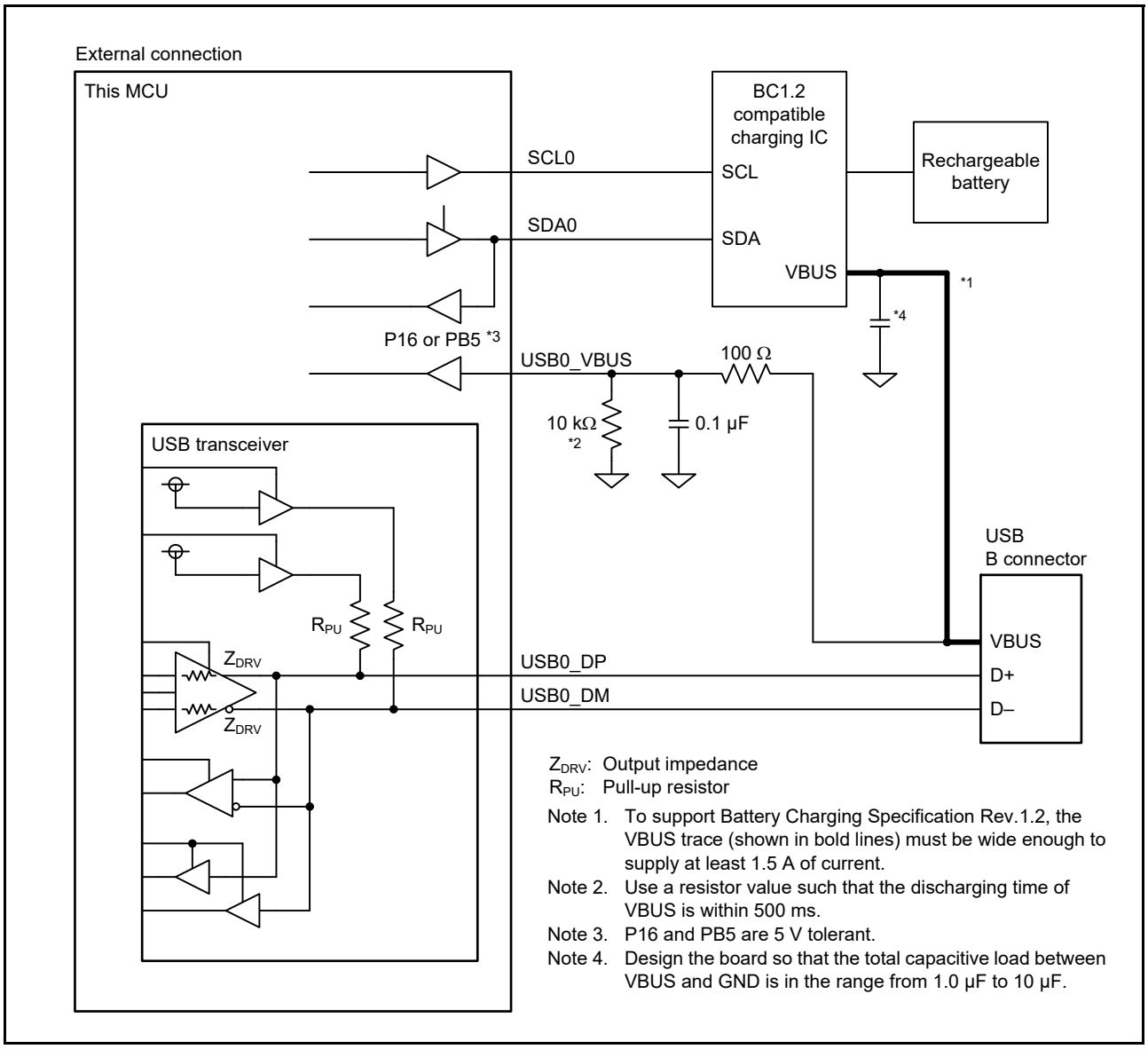


Figure 32.4 Example of Connection between Function and USB Connector When Battery Charging Specification Rev.1.2 is Supported

Downloaded from Arrow.com.

Figure 32.5 shows an example of connection between host controller and USB connector.

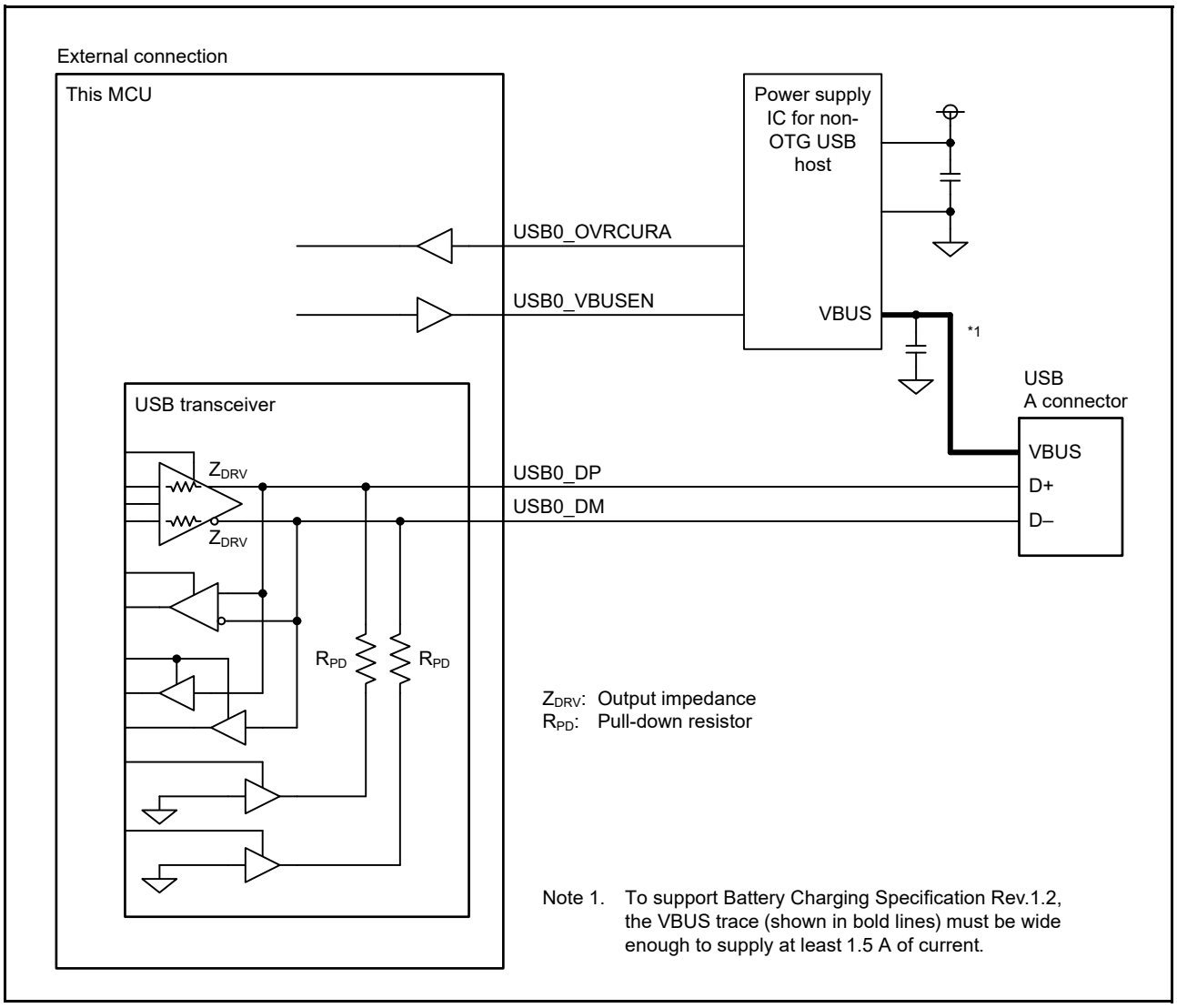


Figure 32.5 Example of Connection between Host and USB Connector

Downloaded from Arrow.com.

Figure 32.6 shows an example of connection between function controller and USB connector in bus powered state.

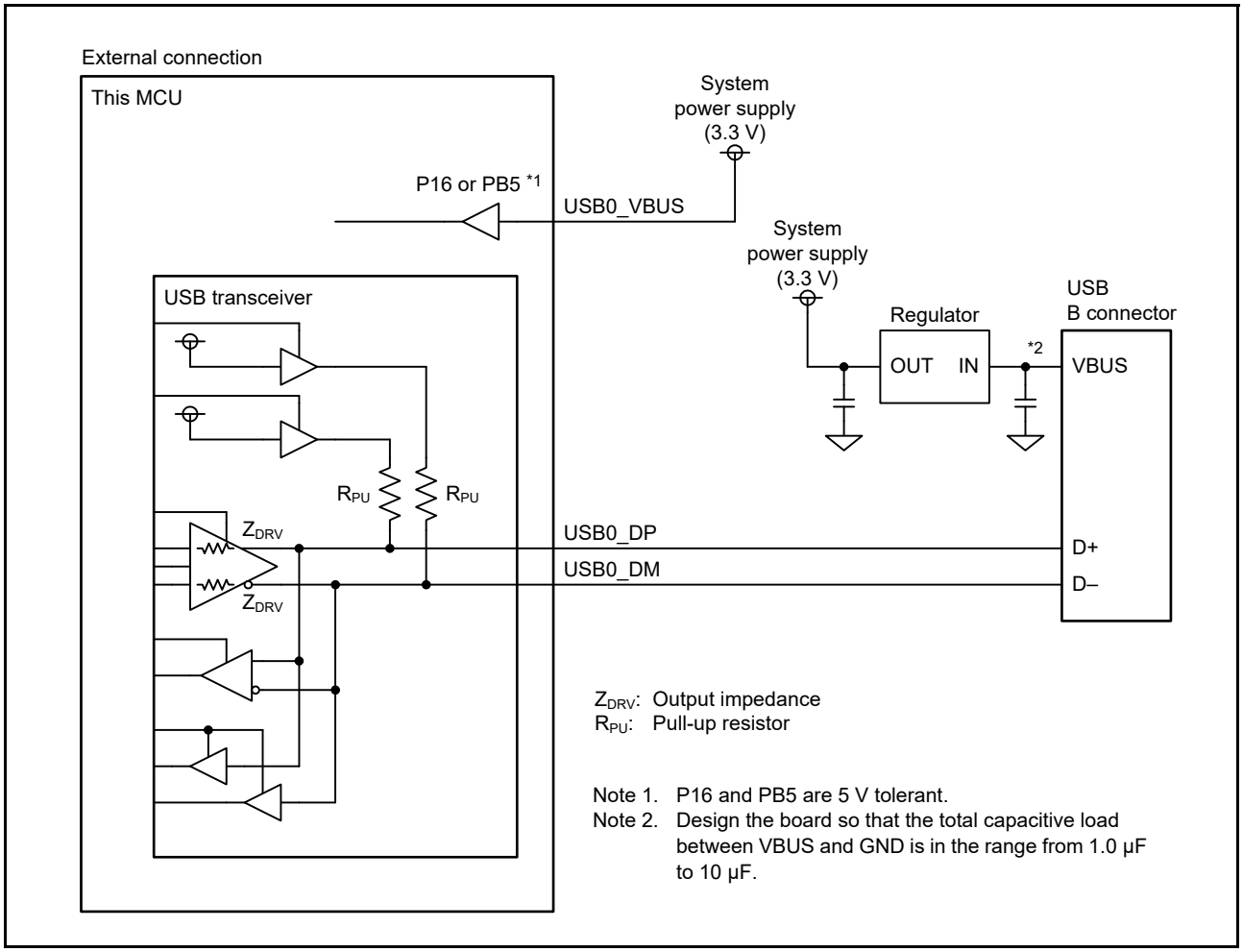


Figure 32.6 Example of Connection between Bus-Powered Function and USB Connector

The examples of external circuits given in this section are simplified circuits, and their operation in every system is not guaranteed.

Downloaded from Arrow.com.

32.3.2 Interrupt Sources

Table 32.12 lists the interrupt sources in the USB.

When an interrupt generation condition is satisfied and the interrupt output is enabled using the corresponding interrupt enable register, a USB interrupt request is issued the Interrupt Controller (ICU) and an USB interrupt will be generated.

Table 32.12 Interrupt Sources

Bit to be Set	Name	Interrupt Source	Function That Generates the Interrupt	Status Flag
VBINT	VBUS interrupt	• When a change in the state of the USB0_VBUS input pin has been detected (low to high or high to low)	Host/function*1	INTSTS0. VBSTS
RESM	Resume interrupt	• When a change in the state of the USB bus has been detected in the suspended state (J-state to K-state or J-state to SE0)	Function	—
SOFR	Frame number update interrupt	[Host controller is selected] • When an SOF packet with a different frame number has been transmitted [Function controller is selected] • When an SOF packet with a different frame number has been received	Host/function	—
DVST	Device state transition interrupt	• When a device state transition has been detected (any of the following conditions) A USB bus reset detected Suspended state detected SET_ADDRESS request received SET_CONFIGURATION request received	Function	INTSTS0. DVSQ[2:0]
CTRT	Control transfer stage transition interrupt	• When a stage transition has been detected in control transfer (any of the following conditions) Setup stage completed Control write transfer status stage transition Control read transfer status stage transition Control transfer completed A control transfer sequence error occurred	Function	INTSTS0. CTSQ[2:0]
BEMP	Buffer empty interrupt	• When transmission of all data in the buffer memory has been completed and the buffer has become empty • When a packet larger than the maximum packet size has been received	Host/function	BEMPSTS. PIPEnBEMP
NRDY	Buffer not ready interrupt	[Host controller is selected] • When STALL has been received from the peripheral device for the issued token • When a response has not been received correctly from the peripheral device for the issued token (no response was returned three consecutive times or a packet reception error occurred three consecutive times) • When an overrun/underrun occurred during isochronous transfer [Function controller is selected] • When NAK has been returned for an IN or OUT token while the PID[1:0] bits are 01b (BUF) • When a CRC error or a bit stuffing error occurred during data reception in isochronous transfer • When an overrun/underrun occurred during data reception in isochronous transfer	Host/function	NRDYSTS. PIPEnNRDY
BRDY	Buffer ready interrupt	• When the buffer has become ready (reading or writing is enabled)	Host/function	BRDYSTS. PIPEnBRDY
OVRCR	Overcurrent input change interrupt	• When a change in the state of the USB0_OVRCURA or USB0_OVRCURB input pin has been detected (low to high or high to low)	Host	INTSTS1. OVRCR
BCHG	Bus change interrupt	• When a change of USB bus state has been detected	Host/function	SYSSTS0. LNST[1:0]
DTCH	Disconnection detection during full-speed operation	• When disconnection of a peripheral device has been detected in full-speed operation	Host	DVSTCTR0. RHST[2:0]
ATTCH	Device connection detection	• When J-state or K-state is detected on the USB port for 2.5 µs. Used for checking whether a peripheral device is connected.	Host	—
EOFERR	EOF error detection	• When an EOF error of a peripheral device has been detected	Host	—
SACK	Normal setup operation	• When the normal response (ACK) for the setup transaction has been received	Host	—
SIGN	Setup error	• When a setup transaction error (no response or ACK packet corruption) was detected three consecutive times	Host	—
PDDETINT0	Portable device detection interrupt	• When connection of the portable device has been detected	Host	INTSTS1. PDDETINT0

Note 1. Though this interrupt can be generated while the host function is selected, it is not usually used with the host function.

Downloaded from Arrow.com.

Figure 32.7 shows the circuits related to the interrupts in the USB.

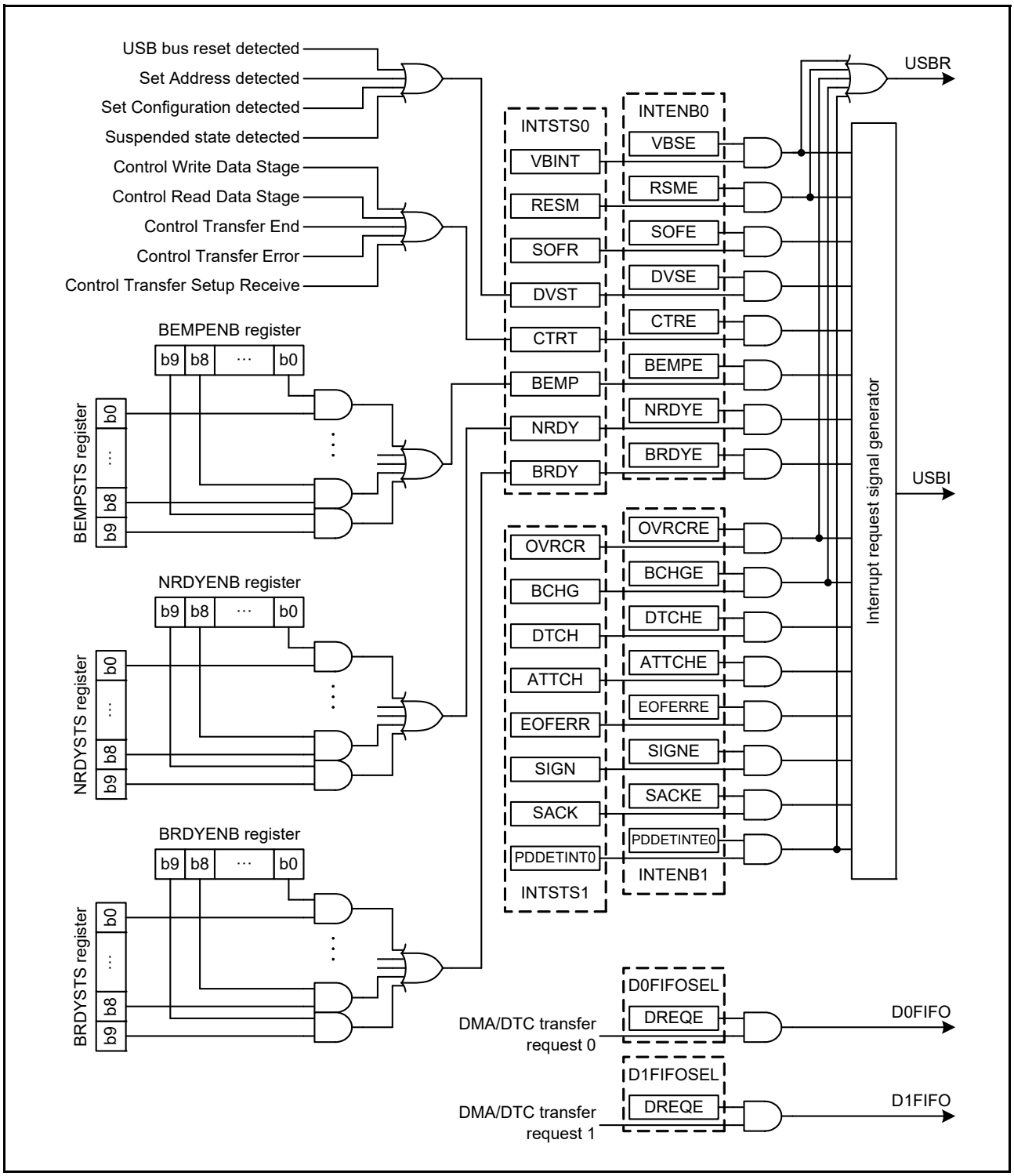


Figure 32.7 Circuits Related to Interrupts in USB

Downloaded from Arrow.com.

Table 32.13 shows the interrupts generated in the USB0.

Table 32.13 USB Interrupts

Interrupt Name	Interrupt Status Flag	DTC Activation	DMAC Activation
D0FIFO	DMA/DTC transfer request 0	Possible	Possible
D1FIFO	DMA/DTC transfer request 1	Possible	Possible
USBI	VBUS interrupt, resume interrupt, frame number update interrupt, device state transition interrupt, control transfer stage transition interrupt, buffer empty interrupt, buffer not ready interrupt, buffer ready interrupt, overcurrent input change interrupt, bus change interrupt, disconnection detection during full-speed operation, device connection detection, EOF error detection, normal setup operation, setup error, and portable device detection interrupt	Not possible	Not possible
USBR	VBUS interrupt, resume interrupt, overcurrent input change interrupt, bus change interrupt, and portable device detection interrupt	Not possible	Not possible

32.3.3 Interrupt Descriptions

32.3.3.1 BRDY Interrupt

The BRDY interrupt is generated when either of the host controller or function controller is selected. The following shows the conditions under which the USB sets 1 to a corresponding bit in the BRDYSTS register. Under this condition, the USB generates a BRDY interrupt if software has set 1 to the BRDYENB.PIPEnBRDYE bit that corresponds to the pipe and 1 to the INTENB0.BRDYE bit.
The conditions for generating and clearing the BRDY interrupt depend on the settings of the SOFCFG.BRDYM bit and PIPECFG.BFRE bit for each pipe as described below.

(1) When the SOFCFG.BRDYM Bit = 0 and the PIPECFG.BFRE Bit = 0

With these settings, the BRDY interrupt indicates that the FIFO port is accessible.
On any of the following conditions, the USB generates an internal BRDY interrupt request trigger and sets 1 to the BRDYSTS.PIPEnBRDY flag corresponding to the pertinent pipe.

(a) For the pipe in the transmitting direction:

- When the DIR bit is changed from 0 to 1 by software.
- When packet transmission is completed using the pertinent pipe while write-access from the CPU to the FIFO buffer for the pertinent pipe is disabled (when the BSTS flag is read as 0).
- When one FIFO buffer is empty on completion of writing data to the other FIFO buffer in double buffer mode.
- No request trigger is generated until completion of writing data to the currently-written FIFO buffer even if transmission to the other FIFO buffer is completed.
- When the hardware flushes the buffer of the pipe for isochronous transfers.
- When 1 is written to the PIPEnCTR.ACLRM bit, which causes the FIFO buffer to make transition from the write-disabled to write-enabled state.

No request trigger is generated for the DCP (that is, during data transmission for control transfers).

Downloaded from Arrow.com.

(b) For the pipe in the receiving direction:

- When packet reception is completed successfully thus enabling the FIFO buffer to be read while read-access from the CPU to the FIFO buffer for the pertinent pipe is disabled (when the BSTS flag is read as 0).
  No request trigger is generated for the transaction in which data PID mismatch has occurred.
- When one FIFO buffer is read-enabled on completion of reading data from the other FIFO buffer in double buffer mode.
  No request trigger is generated until completion of reading data from the currently-read FIFO buffer even if reception by the other FIFO buffer is completed.

When the function controller is selected, the BRDY interrupt is not generated in the status stage of control transfers.
The BRDY interrupt status of the pertinent pipe can be set to 0 by writing 0 to the corresponding PIPEnBRDY flag through software. In this case, 1s should be written to the PIPEnBRDY flags for the other pipes.
Clear the BRDY status before accessing the FIFO buffer.

(2) When the SOFCFG.BRDYM Bit = 0 and the PIPECFG.BFRE Bit = 1

With these settings, the USB generates a BRDY interrupt on completion of reading all data for a single transfer using the pipe in the receiving direction, and sets 1 to the bit in the BRDYSTS register corresponding to the pertinent pipe.
On any of the following conditions, the USB determines that the last data for a single transfer has been received.

- When a short packet including a zero-length packet is received.
- When the pipe n transaction counter register (PIPEnTRN) is used and the number of packets specified by the PIPEnTRN register are completely received.

When the pertinent data is completely read after any of the above conditions has been satisfied, the USB determines that all data for a single transfer has been completely read.
When a zero-length packet is received while the FIFO buffer is empty, the USB module determines that all data for a single transfer has been completely read when the FRDY flag in the FIFO port control register is 1 and the DTLN[8:0] flags are 0. In this case, to start the next transfer, write 1 to the BCLR bit in the corresponding port control register through software.
With these settings, the USB does not detect a BRDY interrupt for the pipe in the transmitting direction.
The BRDY interrupt status of the pertinent pipe can be set to 0 by writing 0 to the corresponding BRDYSTS.PIPEnBRDY flag through software. In this case, 1s should be written to the PIPEnBRDY flags for the other pipes.
In this mode, the PIPECFG.BFRE bit setting should not be modified until all data for a single transfer has been processed. When it is necessary to modify the PIPECFG.BFRE bit before completion of processing, all FIFO buffers for the pertinent pipe should be cleared using the PIPEnCTR.ACLRM bit.

Downloaded from Arrow.com.

(3) When the SOFCFG.BRDYM Bit = 1 and the PIPECFG.BFRE Bit = 0

With these settings, the BRDYSTS.PIPEnBRDY values are linked to the BSTS flag setting for each pipe. In other words, the BRDY interrupt status flags (PIPEnBRDY) are set to 1 or 0 by the USB depending on the FIFO buffer status.

(a) For the pipe in the transmitting direction:

The BRDY interrupt status bits are set to 1 when the FIFO buffer is ready for write access, and are set to 0 when it is not ready.
However, the BRDY interrupt is not generated even if the DCP in the transmitting direction is ready for write access.

(b) For the pipe in the receiving direction:

The BRDY interrupt status bits are set to 1 when the FIFO buffer is ready for read access, and are set to 0 when all data have been read (not ready for read access).
When a zero-length packet is received while the FIFO buffer is empty, the pertinent bit is set to 1 and the BRDY interrupt is continuously generated until BCLR = 1 is written through software.
With this setting, the PIPEnBRDY flag cannot be set to 0 through software.
When the SOFCFG.BRDYM bit is set to 1, all PIPECFG.BFRE bits (for all pipes) should be set to 0.

Downloaded from Arrow.com.

Figure 32.8 shows the timing of BRDY interrupt generation.

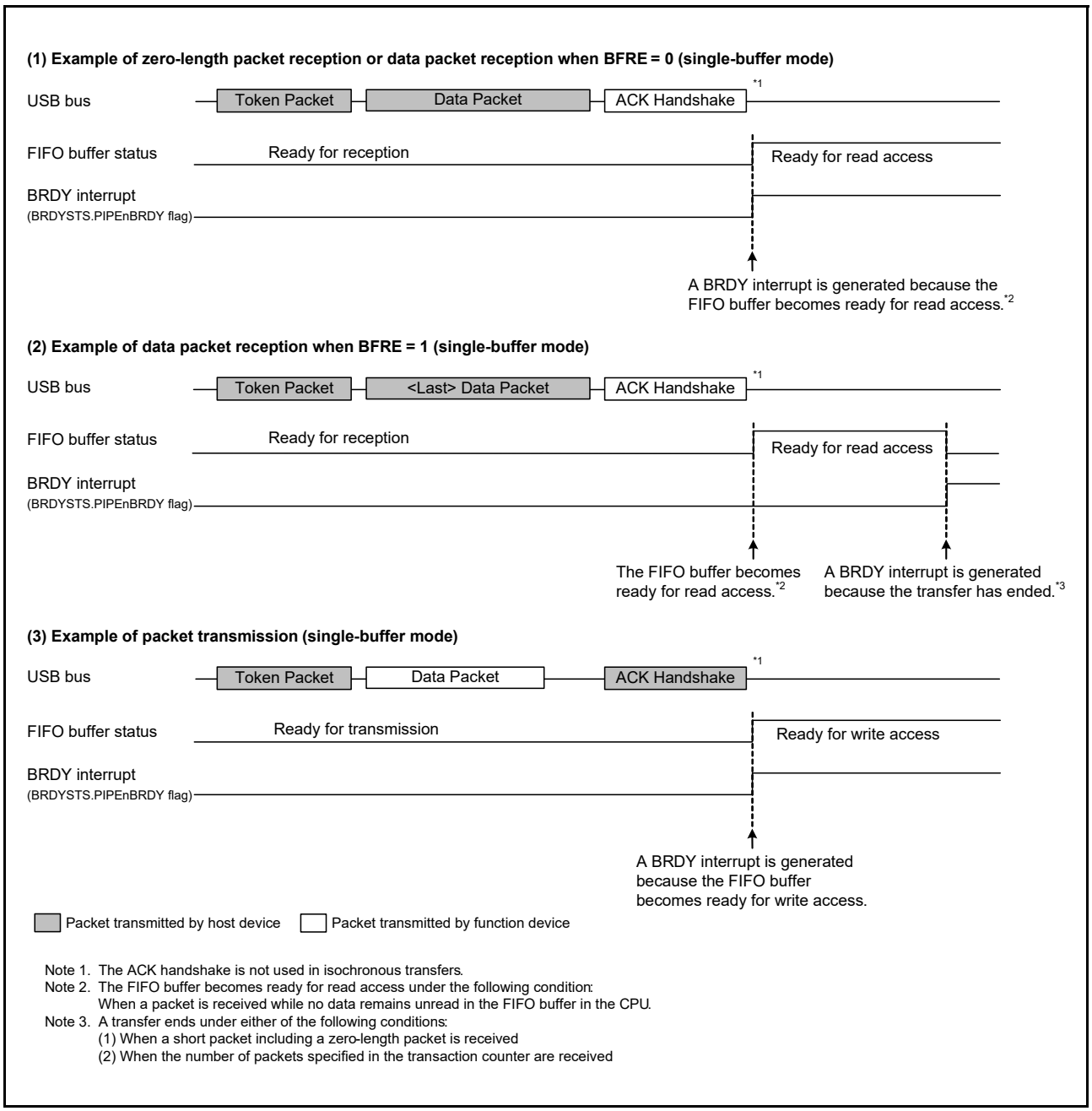


Figure 32.8 Timing of BRDY Interrupt Generation

The condition that USB clears the INTSTS0.BRDY flag depends on the SOFCFG.BRDYM bit setting. Table 32.14 shows the condition for clearing the BRDY flag.

Table 32.14 Condition for Clearing BRDY Flag

BRDYM Bit	Condition for Clearing BRDY Flag
0	The USB sets the BRDY flag to 0 when all bits in the BRDYSTS register have been set to 0 by software.
1	The USB sets the BRDY flag to 0 when the BSTS flags for all pipes have become 0.

Downloaded from Arrow.com.

32.3.3.2 NRDY Interrupt

On generating an internal NRDY interrupt request for the pipe whose PID[1:0] bits are set to 01b (BUF) by software, the USB sets the corresponding NRDYSTS.PIPEnNRDY flag to 1. If the corresponding bit in the NRDYENB register has been set to 1 by software, the USB sets the INTSTS0.NRDY flag to 1 and generates a USB interrupt.
The following describes the conditions on which the USB generates the internal NRDY interrupt request for a given pipe.
Note that the internal NRDY interrupt request is not generated during setup transaction execution when the host controller is selected. During setup transactions when the host controller is selected, the SACK or SIGN interrupt is detected.
The internal NRDY interrupt request is not generated during status stage execution of the control transfer when the function controller is selected.

(1) When Host Controller is Selected

(a) For the pipe in the transmitting direction:

On any of the following conditions, the USB detects an NRDY interrupt.

- For the pipe for isochronous transfers, when the time to issue an OUT token comes while there is no data to be transmitted in the FIFO buffer.
  In this case, the USB transmits a zero-length packet following the OUT token and sets the bit corresponding to the NRDYSTS.PIPEnNRDY flag and the FRMNUM.OVRN flag to 1.
- During communications other than setup transactions using the pipe for the transfers other than isochronous transfers, when any combination of the following two cases occur three consecutive times: 1) no response is returned from the peripheral device (when timeout is detected before detection of the handshake packet from the peripheral device) and 2) an error is detected in the packet from the peripheral device.
  In this case, the USB sets the bit corresponding to the PIPEnNRDY flag to 1 and modifies the setting of the PID[1:0] bits of the corresponding pipe to 00b (NAK).
- During communications other than setup transactions, when the STALL handshake is received from the peripheral device.
  In this case, the USB sets the bit corresponding to the PIPEnNRDY flag to 1 and modifies the setting of the PID[1:0] bits of the corresponding pipe to 11b (STALL).

(b) For the pipe in the receiving direction:

- For the pipe for isochronous transfers, when the time to issue an IN token comes while there is no space available in the FIFO buffer.
  In this case, the USB discards the received data for the IN token and sets the PIPEnNRDY flag corresponding to the pipe and the OVRN flag to 1.
  When a packet error is detected in the received data for the IN token, the USB also sets the FRMNUM.CRCE flag to 1.
- For the pipe for the transfers other than isochronous transfers, when any combination of the following two cases occur three consecutive times: 1) no response is returned from the peripheral device for the IN token issued by the USB (when timeout is detected before detection of the DATA packet from the peripheral device) and 2) an error is detected in the packet from the peripheral device.
  In this case, the USB sets the PIPEnNRDY flag corresponding to the pipe to 1 and modifies the setting of the PID[1:0] bits of the corresponding pipe to 00b (NAK).
- For the pipe for isochronous transfers, when no response is returned from the peripheral device for the IN token (when timeout is detected before detection of the DATA packet from the peripheral device) or an error is detected in the packet from the peripheral device.
  In this case, the USB sets the PIPEnNRDY flag corresponding to the pipe to 1. (The setting of the PID[1:0] bits of the pipe is not modified.)

Downloaded from Arrow.com.

- For the pipe for isochronous transfers, when a CRC error or a bit stuffing error is detected in the received data packet.
  In this case, the USB sets the PIPEnNRDY flag corresponding to the pipe and the CRCE flag to 1.
- When the STALL handshake is received.
  In this case, the USB sets the PIPEnNRDY flag corresponding to the pipe to 1 and modifies the setting of the PID[1:0] bits of the corresponding pipe to 11b (STALL).

(2) When Function Controller is Selected

(a) For the pipe in the transmitting direction:

- When an IN token is received while there is no data to be transmitted in the FIFO buffer.
  In this case, the USB generates a NRDY interrupt request at the reception of the IN token and sets the NRDYSTS.PIPEnNRDY flag to 1.
  For the pipe for the isochronous transfers in which an interrupt is generated, the USB transmits a zero-length packet and sets the FRMNUM.OVRN flag to 1.

(b) For the pipe in the receiving direction:

- When an OUT token is received while there is no space available in the FIFO buffer.
  For the pipe for the isochronous transfers in which an interrupt is generated, the USB generates a NRDY interrupt request at the reception of the OUT token and sets the PIPEnNRDY flag to 1 and OVRN flag to 1.
  For the pipe for the transfers other than isochronous transfers in which an interrupt is generated, the USB generates a NRDY interrupt request when a NAK handshake is transferred after the data following the OUT token is received, and sets the PIPEnNRDY flag to 1.
  However, during re-transmission (due to data PID mismatch), the NRDY interrupt request is not generated. In addition, if an error occurs in the DATA packet, the NRDY interrupt request is not generated.
- For the pipe for isochronous transfers, when a token is not received successfully within an interval frame.
  In this case, the USB generates a NRDY interrupt request when SOF is received, and sets the PIPEnNRDY flag to 1.

Downloaded from Arrow.com.

Figure 32.9 shows the timing of NRDY interrupt generation when the function controller is selected.

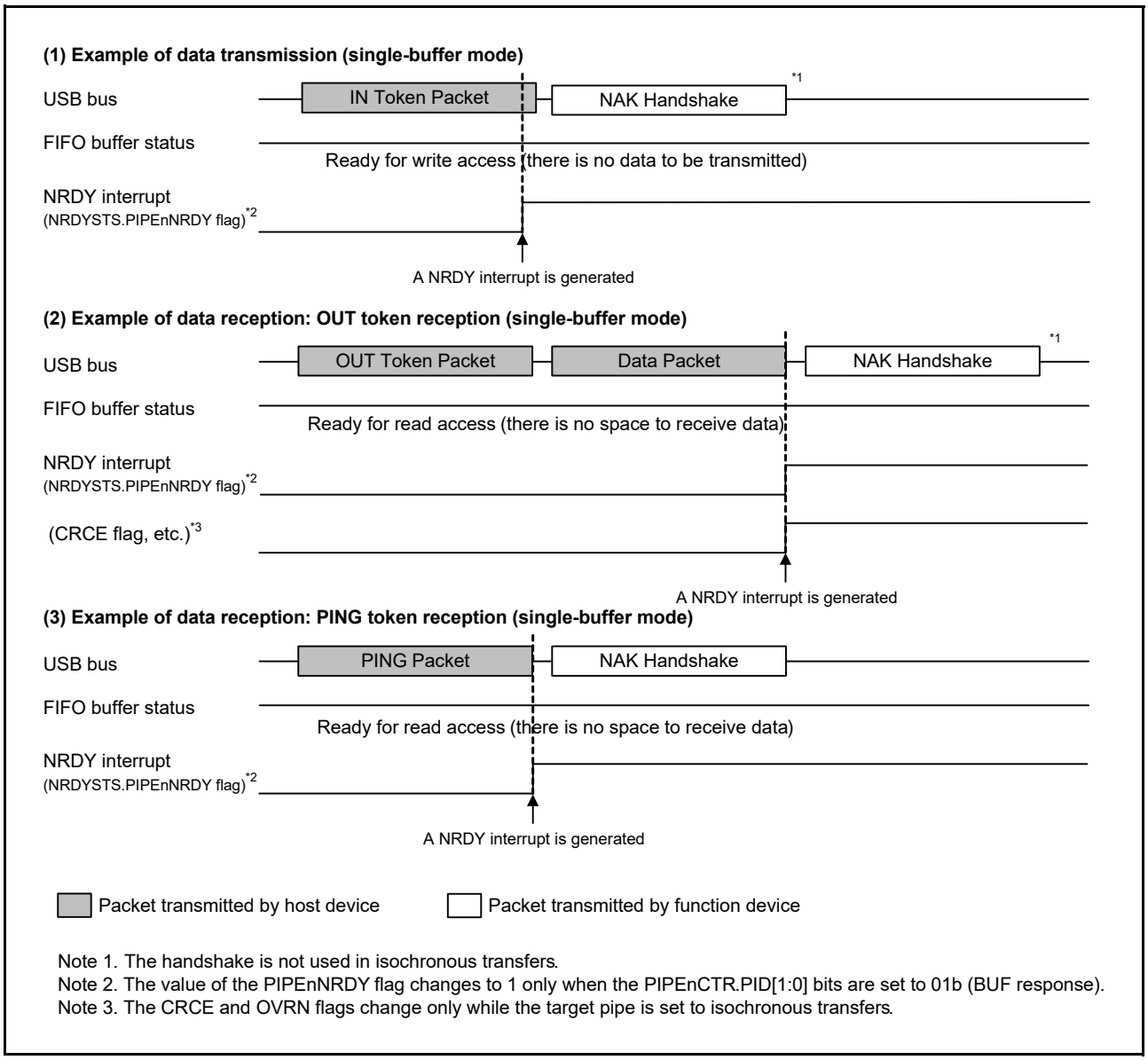


Figure 32.9 Timing of NRDY Interrupt Generation (When Function Controller is Selected)

Downloaded from Arrow.com.

32.3.3.3 BEMP Interrupt

On detecting a BEMP interrupt for the pipe whose PID[1:0] bits are set to 01b (BUF) by software, the USB sets the corresponding BEMPSTS.PIPEnBEMP flag to 1. If the corresponding bit in the BEMPENB register has been set to 1 by software, the USB sets the INTSTS0.BEMP flag to 1 and generates a USB interrupt.

The following describes the conditions on which the USB generates an internal BEMP interrupt request.

(1) For the pipe in the transmitting direction:

When the FIFO buffer of the corresponding pipe is empty on completion of transmission (including zero-length packet transmission).
In single buffer mode, an internal BEMP interrupt request is generated simultaneously with the BRDY interrupt for the pipe other than DCP. However, the internal BEMP interrupt request is not generated on any of the following conditions.

- When the CPU or DMAC/DTC has already started writing data to the FIFO buffer of the CPU on completion of transmitting data from one FIFO buffer in double buffer mode.
- When the buffer is cleared (emptied) by setting the PIPEnCTR.ACLRM or the BCLR bit in the port control register to 1.
- When IN transfer (zero-length packet transmission) is performed during the control transfer status stage while the function controller is selected.

(2) For the pipe in the receiving direction:

When the successfully-received data packet size exceeds the specified maximum packet size.
In this case, the USB generates a BEMP interrupt request, sets the corresponding BEMPSTS.PIPEnBEMP flag to 1, discards the received data, and modifies the setting of the PID[1:0] bits of the corresponding pipe to 11b (STALL). Here, the USB returns no response when used as the host controller, and returns STALL response when used as the function controller.
However, the internal BEMP interrupt request is not generated on any of the following conditions.

- When a CRC error or a bit stuffing error is detected in the received data.
- When a setup transaction is being performed,
  Writing 0 to the BEMPSTS.PIPEnBEMP flag clears the status.
  Writing 1 to the BEMPSTS.PIPEnBEMP flag has no effect.

Figure 32.10 shows the timing of BEMP interrupt generation when the function controller is selected.

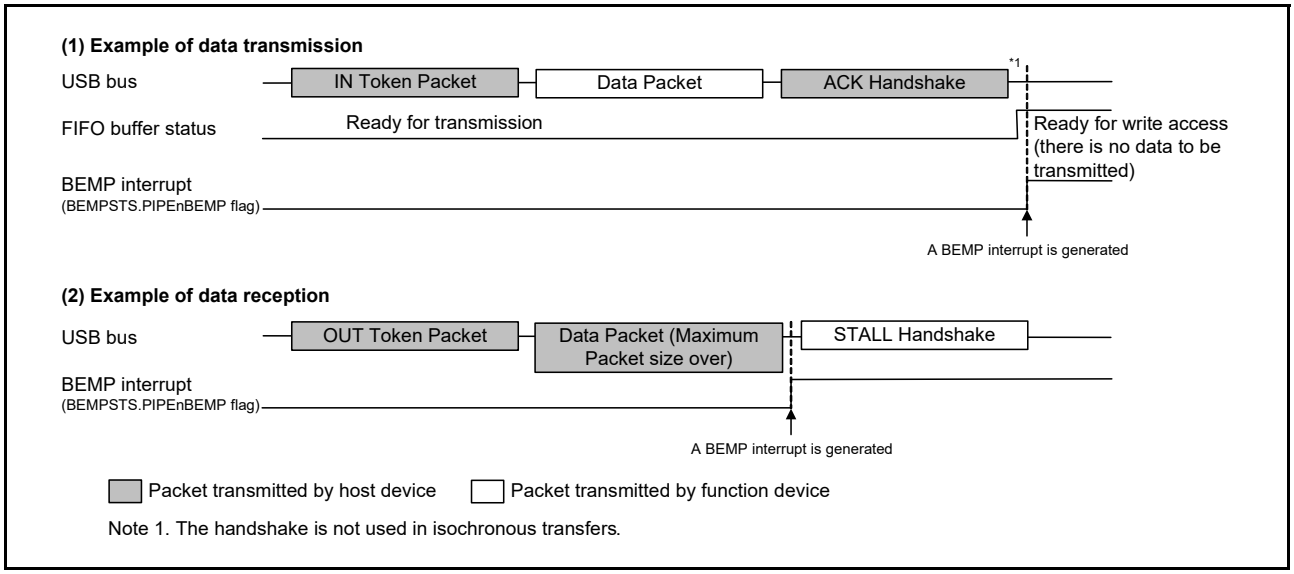


Figure 32.10 Timing of BEMP Interrupt Generation (When Function Controller is Selected)

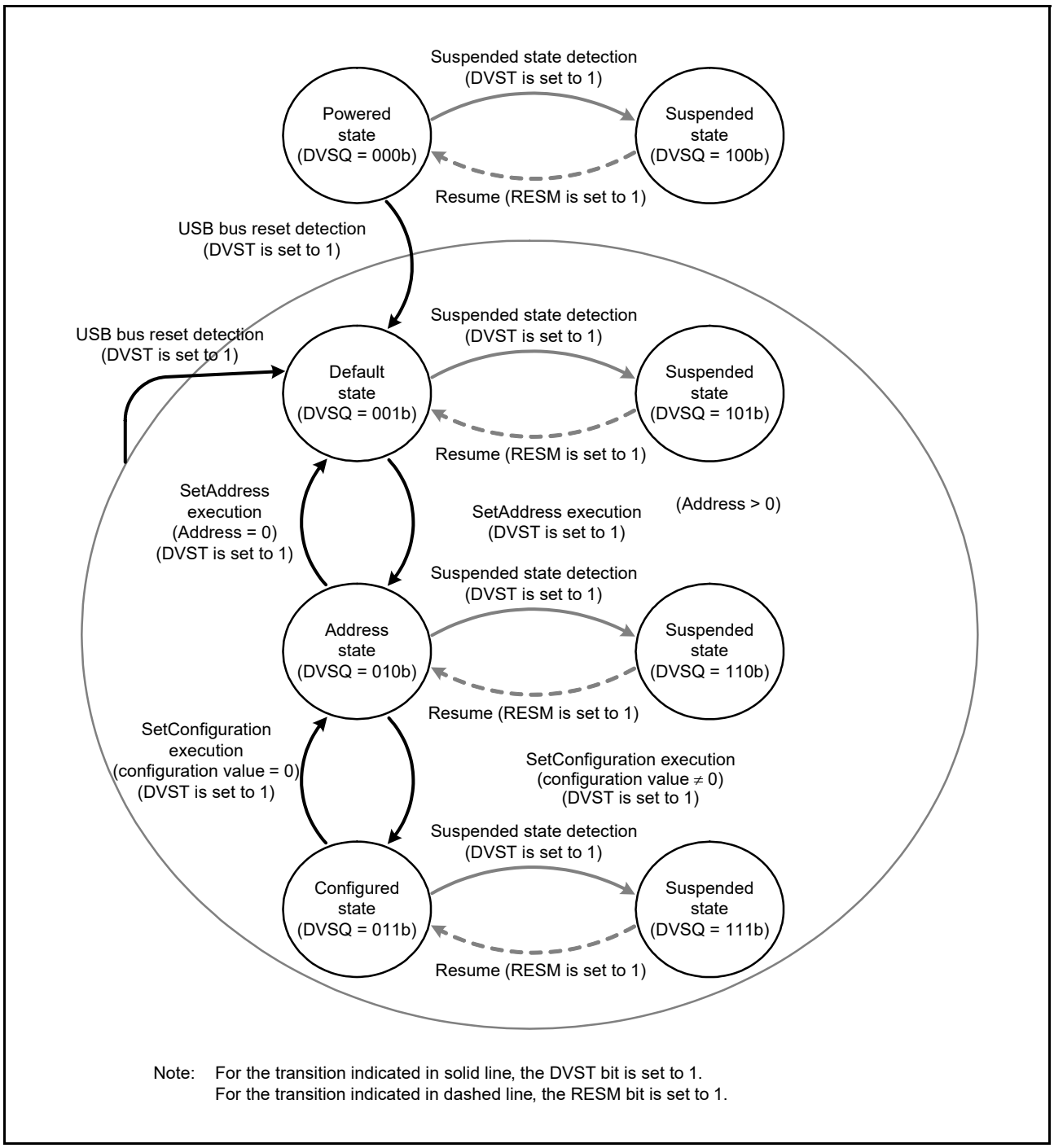
Downloaded from Arrow.com.

32.3.3.4 Device State Transition Interrupt

Figure 32.11 is a diagram of device state transitions in the USB. The USB controls device state and generates device state transition interrupts. However, recovery from the suspended state (resume signal detection) is detected by means of the resume interrupt. The device state transition interrupts can be enabled or disabled individually using INTENB0. The device state to which a transition was made can be confirmed using the INTSTS0.DVSQ[2:0] flags.

When a transition is made to the default state, a device state transition interrupt is generated after a USB bus reset is detected.

Device state can be controlled only when the function controller is selected. The device state transition interrupts can also be generated only when the function controller is selected.



Note: For the transition indicated in solid line, the DVST bit is set to 1.
For the transition indicated in dashed line, the RESM bit is set to 1.

Figure 32.11 Device State Transitions

Downloaded from Arrow.com.

32.3.3.5 Control Transfer Stage Transition Interrupt

Figure 32.12 is a diagram of control transfer stage transitions in the USB. The USB controls the control transfer sequence and generates control transfer stage transition interrupts. The control transfer stage transition interrupts can be enabled or disabled individually using INTENB0. The transfer stage to which a transition was made can be confirmed using the INTSTS0.CTSQ[2:0] flags.

Control transfer stage transition interrupts are generated only when the function controller is selected.

The control transfer sequence errors are listed below. If an error occurs, the DCPCTR.PID[1:0] bits are set to 1xb (STALL response).

During control read transfer:

- An OUT token is received while no data has been transferred for the IN token at the data stage.
- An IN token is received at the status stage.
- A data packet with PID = DATA0 is received at the status stage.

During control write transfer:

- An IN token is received while no ACK response has been returned for the OUT token at the data stage.
- A data packet with PID = DATA0 is received for the first data packet at the data stage.
- An OUT token is received at the status stage

During no-data control transfers:

- An OUT token is received at the status stage.

At the control write transfer data stage, if the number of receive data exceeds the wLength value of the USB request, it cannot be recognized as a control transfer sequence error. At the control read transfer status stage, packets other than zero-length packets are received by an ACK response and the transfer ends normally.

When a CTRT interrupt occurs in response to a sequence error (INTSTS0.CTRT = 1), CTSQ[2:0] = 110b value is retained until the CTRT flag = 0 is written from the system (the interrupt status is cleared). Therefore, while CTSQ[2:0] = 110b is being held, the CTRT interrupt that ends the setup stage will not be generated even if a new USB request is received. (The USB retains the setup stage end, and after the interrupt status has been cleared by software, a setup stage end interrupt is generated.)

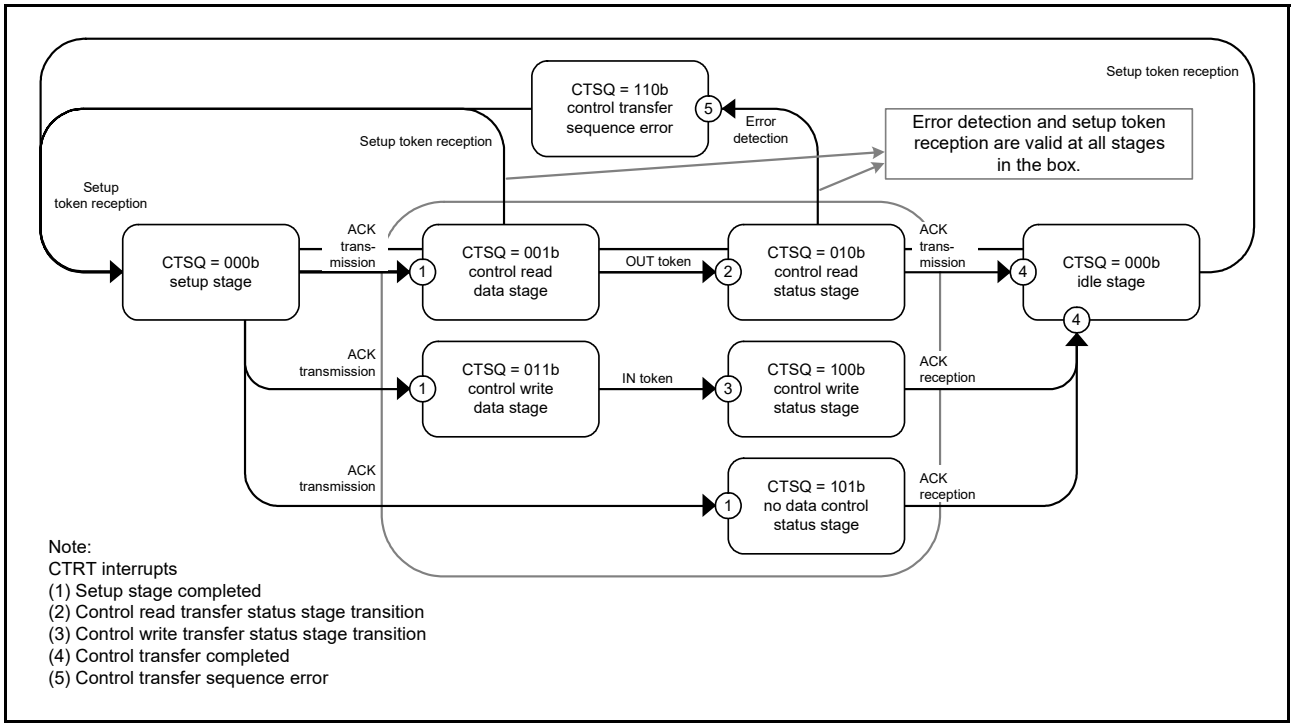


Figure 32.12 Control Transfer Stage Transitions

Downloaded from Arrow.com.

32.3.3.6 Frame Update Interrupt

With the host controller selected, an interrupt is generated at the timing when the frame number is updated. With the function controller selected, an SOFR interrupt is generated when the frame number is updated.
When the function controller is selected, the USB updates the frame number and generates an SOFR interrupt if it detects a new SOF packet during full-speed operation.

32.3.3.7 VBUS Interrupt

When the USB0_VBUS pin level changes, a VBUS interrupt is generated. The level of the USB0_VBUS pin can be checked with the INTSTS0.VBSTS flag. Whether the host controller is connected or disconnected can be confirmed using the VBUS interrupt. However, if the system is activated with the host controller connected, the first VBUS interrupt is not generated because there is no change in the USB0_VBUS pin level.

32.3.3.8 Resume Interrupt

When the function controller is selected, a resume interrupt is generated when the device state is the suspended state and the USB bus state has changed (from J-state to K-state, or from J-state to SE0). Recovery from the suspended state is detected by means of the resume interrupt.
When the host controller is selected, no resume interrupt is generated. Use the BCHG interrupt to detect a change in the USB bus state.

32.3.3.9 OVRCR Interrupt

An OVRCR interrupt is generated when the USB0_OVRCURA or USB0_OVRCURB pin level has changed. The levels of the USB0_OVRCURA and USB0_OVRCURB pins can be checked with the SYSSTS0.OVCMON[1:0] bits. The external power supply IC can check whether overcurrent has been detected using the OVRCR interrupt.
For OTG connection, whether a change has been detected in the VBUS comparator can be checked using the OVRCR interrupt.

32.3.3.10 BCHG Interrupt

A BCHG interrupt is generated when the USB bus state has changed. The BCHG interrupt can be used to detect whether the peripheral device is connected and can also be used to detect a remote wakeup when the host controller is selected. The BCHG interrupt is generated regardless of whether the host controller or function controller is selected.

32.3.3.11 DTCH Interrupt

A DTCH interrupt is generated when disconnection of the USB bus is detected while the host controller is selected. The USB detects bus disconnection based on USB Specification 2.0.
After detecting a DTCH interrupt, the USB controls hardware as described below (irrespective of the value set in the corresponding interrupt enable bit). All pipes in which communications are currently carried out for the pertinent port should be terminated by software and make a transition to the wait state for bus connection to the pertinent port (wait state for ATTCH interrupt generation).

- Modifies the DVSTCTR0.UACT bit for the port in which a DTCH interrupt has been detected to 0.
- Puts the port in which a DTCH interrupt has been generated into the idle state.

Downloaded from Arrow.com.

32.3.3.12 SACK Interrupt

A SACK interrupt is generated when an ACK response for the transmitted setup packet has been received from the peripheral device with the host controller selected. The SACK interrupt can be used to confirm that the setup transaction has been completed successfully.

32.3.3.13 SIGN Interrupt

A SIGN interrupt is generated when an ACK response for the transmitted setup packet has not been correctly received from the peripheral device three consecutive times with the host controller selected. The SIGN interrupt can be used to detect no ACK response transmitted from the peripheral device or corruption of an ACK packet.

32.3.3.14 ATTCH Interrupt

An ATTCH interrupt is generated when J-state or K-state of the full-speed signal level is detected on the USB port for 2.5 µs with the host controller selected. To be more specific, an ATTCH interrupt is detected on any of the following conditions.

- When K-state, SE0, or SE1 changes to J-state, and J-state continues 2.5 µs.
- When J-state, SE0, or SE1 changes to K-state, and K-state continues 2.5 µs.

32.3.3.15 EOFERR Interrupt

An EOFERR interrupt is generated when it is detected that communication is not completed at the EOF2 timing prescribed in USB Specification 2.0.
After detecting an EOFERR interrupt, the USB controls hardware as described below (irrespective of the value set in the corresponding interrupt enable bit). All pipes in which communications are currently carried out for the pertinent port should be terminated by software and perform re-enumeration of the pertinent port.

- Modifies the DVSTCTR0.UACT bit for the port in which an EOFERR interrupt has been detected to 0.
- Puts the port in which an EOFERR interrupt has been generated into the idle state.

32.3.3.16 Portable Device Detection Interrupt

A portable device detection interrupt is generated when the USB module detects a level change (high to low or low to high) in the PDDET output from the USB-PHY. When a portable device detection interrupt is generated, use software to repeat reading the PDDETSTS0 flag until the same value is read three or more times, and perform debouncing.

Downloaded from Arrow.com.

32.3.4 Pipe Control

Table 32.15 lists the pipe settings for the USB. With USB data transfer, data transfer is carried out using the pipe that the software has associated with the endpoint. The USB has ten pipes that are used for data transfer.
Appropriate settings should be made for each of the pipes according to the specifications of the system.

Table 32.15 Pipe Settings

Register Name	Bit Name	Setting	Remarks
DCPCFG PIPECFG	TYPE[1:0]	Specifies the transfer type	Pipes 1 to 9: Can be set
	BFRE	Selects the BRDY interrupt mode	Pipes 1 to 5: Can be set
	DBLB	Selects double buffer mode	Pipes 1 to 5: Can be set
	DIR	Selects transfer direction	IN or OUT can be set
	EPNUM[3:0]	Endpoint number	Pipes 1 to 9: Can be set A value other than 0000b should be set when the pipe is used.
	SHTNAK	Selects disabled state for pipe when transfer ends	Pipes 1 and 2: Can be set (only when bulk transfer has been selected) Pipes 3 to 5: Can be set
DCPMAXP PIPEMAXP	DEVSEL[3:0]	Selects a device	Referenced only when the host controller is selected.
	MXPS[8:0]	Maximum packet size	Compliant with USB Specification 2.0.
PIPEPERI	IFIS	Buffer flush	Pipes 1 and 2: Can be set (only when isochronous transfer has been selected) Pipes 3 to 9: Cannot be set
	IITV[2:0]	Interval counter	Pipes 1 and 2: Can be set (only when isochronous transfer has been selected) Pipes 3 to 5: Cannot be set Pipes 6 to 9: Can be set (only when the host controller has been selected)
DCPCTR PIPEnCTR	BSTS	Buffer status	For the DCP, receive buffer status and transmit buffer status are switched with the ISEL bit.
	INBUFM	IN buffer monitor	Available only for pipes 1 to 5.
	SUREQ	SETUP request	Can be set only for the DCP. Can be controlled only when the host controller has been selected.
	SUREQCLR	SUREQ clear	Can be set only for the DCP. Can be controlled only when the host controller has been selected.
	ATREPM	Auto response mode	Pipes 1 to 5: Can be set Can be set only when the function controller has been selected.
	ACLRM	Auto buffer clear	Pipes 1 to 9: Can be set
	SQCLR	Sequence clear	Clears the data toggle bit.
	SQSET	Sequence set	Sets the data toggle bit.
	SQMON	Sequence monitor	Monitors the data toggle bit.
	PBUSY	Pipe busy status	
	PID[1:0]	Response PID	Refer to section 32.3.4.6, Response PID.
PIPEnTRE	TRENB	Transaction counter enable	Pipes 1 to 5: Can be set
	TRCLR	Current transaction counter clear	Pipes 1 to 5: Can be set
PIPEnTRN	—	Transaction counter	Pipes 1 to 5: Can be set

Downloaded from Arrow.com.

32.3.4.1 Pipe Control Register Switching Procedures

The following bits in the pipe control registers can be modified only when USB communication is prohibited (PID[1:0] = 00b (NAK)).

The following shows the registers and bits that should not be modified when USB communication is enabled (PID[1:0] = 01b (BUF)).

- Bits in the DCPCFG and DCPMAXP registers
- The SQCLR and SQSET bits in the DCPCTR register
- Bits in registers PIPECFG, PIPEMAXP, and PIPEPERI
- The ATREPM, ACLRM, SQCLR, and SQSET bits in the PIPEnCTR register
- Bits in the PIPEnTRE and PIPEnTRN registers

In order to modify the above bits in the USB communication enabled (PID[1:0] = 01b (BUF)) state, follow the procedure shown below:

1. A request to modify bits in the pipe control register occurs.
2. Modify the PID[1:0] bits corresponding to the pipe to 00b (NAK).
3. Wait until the corresponding PBUSY flag is set to 0.
4. Modify the bits in the pipe control register.

The following bits in the pipe control registers can be modified only when the pertinent pipe information has not been set by the CURPIPE[3:0] bits in registers CFIFOSEL, D0FIFOSEL, and D1FIFOSEL.

Registers that should not be set when the CURPIPE[3:0] bits are set:

- Bits in the DCPCFG and DCPMAXP register
- Bits in registers PIPECFG, PIPEMAXP and PIPEPERI

In order to modify pipe information, the CURPIPE[3:0] bits in the port select registers should be set to a pipe other than the pipe to be modified. For the DCP, the buffer should be cleared using the BCLR bit in the port control register after the pipe information is modified.

32.3.4.2 Transfer Types

The PIPECFG.TYPE[1:0] bits are used to specify the transfer type for each pipe. The transfer types that can be set for the pipes are as follows.

- DCP: No setting is necessary (fixed at control transfer).
- Pipes 1 and 2: These should be set to bulk transfer or isochronous transfer.
- Pipes 3 to 5: These should be set to bulk transfer.
- Pipes 6 to 9: These should be set to interrupt transfer.

32.3.4.3 Endpoint Number

The PIPECFG.EPNUM[3:0] bits are used to set the endpoint number for each pipe. The DCP is fixed at endpoint 0. The other pipes can be set from endpoint 1 to endpoint 15.

- DCP: No setting is necessary (fixed at endpoint 0).
- Pipes 1 to 9: The endpoint numbers from 1 to 15 should be selected and set.
  These should be set so that the combination of the PIPECFG.DIR bit and EPNUM[3:0] bits is unique.

Downloaded from Arrow.com.

32.3.4.4 Maximum Packet Size Setting

The DCPMAXP.MXPS[6:0] bits and the PIPEMAXP.MXPS[8:0] bits are used to specify the maximum packet size for each pipe. DCP and pipes 1 to 5 can be set to any of the maximum pipe sizes defined by USB Specification 2.0. For pipes 6 to 9, 64 bytes are the upper limit of the maximum packet size. The maximum packet size should be set before beginning the transfer (PID[1:0] = 01b (BUF)).

- DCP: Set 8, 16, 32, or 64.
- Pipes 1 to 5: Set 8, 16, 32, or 64 when using bulk transfer.
- Pipes 1 and 2: Set a value between 1 and 256 when using isochronous transfer.
- Pipes 6 to 9: Set a value between 1 and 64.

32.3.4.5 Transaction Counter (For Pipes 1 to 5 in Reading Direction)

When the specified number of transactions has been completed in the data packet receiving direction, the USB recognizes that the transfer has ended. Two transaction counters are provided: one is the PIPEnTRN register that specifies the number of transactions to be executed and the other is the current counter that internally counts the number of executed transactions. With the PIPECFG.SHTNAK bit set to 1, when the current counter value matches the specified number of transactions, the corresponding PIPEnCTR.PID[1:0] bits are set to 00b (NAK) and the subsequent transfer is disabled. The transactions can be counted again from the beginning by initializing the current counter of the transaction counter function through the PIPEnTRE.TRCLR bit. The information read from PIPEnTRN differs depending on the setting of the PIPEnTRE.TRENB bit.

- The TRENB bit = 0: The specified transaction counter value can be read.
- The TRENB bit = 1: The current counter value indicating the internally counted number of executed transactions can be read.

When operating the TRCLR bit, the following should be noted.

- If the transactions are being counted and PID[1:0] = 01b (BUF), the current counter cannot be cleared.
- If there is any data left in the buffer, the current counter cannot be cleared.

Downloaded from Arrow.com.

32.3.4.6 Response PID

The PID[1:0] bits in the DCPCTR and PIPEnCTR registers are used to set the response PID for each pipe.
The following shows the USB operation with various response PID settings:

(1) Response PID settings when the host controller is selected:

The response PID is used to specify the execution of transactions.

- NAK setting: Using pipes is disabled. No transaction is executed.
- BUF setting: Transactions are executed based on the status of the buffer memory.
  For OUT direction: If there are transmit data in the buffer memory, an OUT token is issued.
  For IN direction: If there is an area to receive data in the buffer memory, an IN token is issued.
- STALL setting: Using pipes is disabled. No transaction is executed.

Note: Setup transactions for the DCP are set with the DCPCTR.SUREQ bit.

(2) Response PID settings when the function controller is selected:

The response PID is used to specify the response to transactions from the host.

- NAK setting: The NAK response is returned in response to the generated transaction.
- BUF setting: Responses are made to transactions according to the status of the buffer memory.
- STALL setting: The STALL response is returned in response to the generated transaction.

Note: For setup transactions, an ACK response is returned regardless of the PID[1:0] bits setting, and the USB request is stored in the register.

The USB may write to the PID[1:0] bits, depending on the results of the transaction as described below.

(3) When the host controller has been selected and the response PID is set by hardware:

- NAK setting: In the following cases, PID[1:0] = 00b (NAK) is set and issuing of tokens is automatically stopped:
  When a transfer other than isochronous transfer has been performed and an NRDY interrupt is generated.
  (For details, refer to section 32.3.3.2, NRDY Interrupt.)
  - If a short packet is received when the PIPECFG.SHTNAK bit has been set to 1 for bulk transfer.
  - If the transaction counting ends when the SHTNAK bit has been set to 1 for bulk transfer.
- BUF setting: There is no BUF writing by the USB.
- STALL setting: In the following cases, PID[1:0] = 1xb (STALL) is set and issuing of tokens is automatically stopped:
  When STALL is received in response to the transmitted token.
  When the size of the receive data packet exceeds the maximum packet size.

(4) When the function controller has been selected and the response PID is set by hardware:

- NAK setting: In the following cases, PID[1:0] = 00b (NAK) is set and NAK is returned in response to transactions:
  When the SETUP token is received normally (DCP only).
  If the transaction counting ends or a short packet is received when the PIPECFG.SHTNAK bit has been set to 1 for bulk transfer.
- BUF setting: There is no BUF writing by the USB.
- STALL setting: In the following cases, PID[1:0] = 1xb (STALL) is set and STALL is returned in response to transactions:
  When a maximum packet size exceeded error is detected in the received data packet.
  When a control transfer sequence error has been detected (DCP only).

Downloaded from Arrow.com.

32.3.4.7 Data PID Sequence Bit

The USB automatically toggles the sequence bit in the data PID when data is transferred successfully in the control transfer data stage, bulk transfer, and interrupt transfer. The sequence bit of the next data PID to be transmitted can be confirmed with the SQMON flag in the DCPCTR and PIPEnCTR registers. When data is transmitted, the sequence bit switches at the timing of ACK handshake reception. When data is received, the sequence bit switches at the timing of ACK handshake transmission. The DCPCTR.SQCLR bit and the PIPEnCTR.SQSET bit can be used to change the data PID sequence bit.

When the function controller has been selected and control transfer is used, the USB automatically sets the sequence bit when a stage transition is made. DATA1 is returned when the setup stage is ended. The sequence bit is not referenced and PID = DATA1 is returned in a status stage. Therefore, software settings are not required. However, when the host controller has been selected and control transfer is used, the sequence bit should be set by software at a stage transition. For the ClearFeature request transmission or reception, the data PID sequence bit should be set by software regardless of whether the host controller or function controller is selected.

32.3.4.8 Response PID = NAK Function

The USB has a function that disables pipe operation (response PID = NAK) at the timing at which the final data packet of a transaction is received (the USB automatically distinguishes this based on reception of a short packet or the transaction counter) by setting the PIPECFG.SHTNAK bit to 1.

When the double buffer mode is being used for the buffer memory, using this function enables reception of data packets in transfer units. If pipe operation has been disabled, software should set the pipe to the enabled state again (response PID = BUF).

The response PID = NAK function can be used only when bulk transfers are used.

32.3.4.9 Auto Response Mode

With the pipes for bulk transfer (pipe 1 to pipe 5), when the PIPEnCTR.ATREPM bit is set to 1, a transition is made to auto response mode. During an OUT transfer (the PIPECFG.DIR bit = 0), OUT-NAK mode is entered, and during an IN transfer (the DIR bit = 1), null auto response mode is entered.

32.3.4.10 OUT-NAK Mode

With the pipes for bulk OUT transfer, NAK is returned in response to an OUT token and an NRDY interrupt is output when the PIPEnCTR.ATREPM bit is set to 1. To make a transition from normal mode to OUT-NAK mode, OUT-NAK mode should be specified in the pipe operation disabled state (response PID = NAK) before enabling pipe operation (response PID = BUF). After pipe operation has been enabled, OUT-NAK mode becomes valid. However, if an OUT token is received immediately before pipe operation is disabled, the token data is normally received, and an ACK is returned to the host.

To make a transition from OUT-NAK mode to normal mode, OUT-NAK mode should be canceled in the pipe operation disabled state (response PID = NAK) before enabling pipe operation (response PID = BUF). In normal mode, reception of OUT data is enabled.

Downloaded from Arrow.com.

32.3.4.11 Null Auto Response Mode

With the pipes for bulk IN transfer, zero-length packets are continuously transmitted when the PIPEnCTR.ATREPM bit is set to 1.

To make a transition from normal mode to null auto response mode, null auto response mode should be set in the pipe operation disabled state (response PID = NAK) before enabling pipe operation (response PID = BUF). After pipe operation has been enabled, null auto response mode becomes valid. Before setting null auto response mode, the PIPEnCTR.INBUFM = 0 should be confirmed because the mode can be set only when the buffer is empty. If the INBUFM flag is 1, the buffer should be emptied with the PIPEnCTR.ACLRM bit. While a transition to null auto response mode is being made, data should not be written from the FIFO port.

To make a transition from null auto response mode to normal mode, pipe operation disabled state (response PID = NAK) should be retained for the period of zero-length packet transmission (about 10 µs) before canceling null auto response mode. In normal mode, data can be written from the FIFO port; therefore, packet transmission to the host is enabled by enabling pipe operation (response PID = BUF).

32.3.5 FIFO Buffer Memory

32.3.5.1 FIFO Buffer Memory

The USB has FIFO buffer memory for data transfer. The memory area used for each pipe is managed by the USB. The FIFO buffer memory has two states depending on whether the access right is assigned to the system (CPU side) or the USB (SIE side).

(1) Buffer Status

Table 32.16 and Table 32.17 show the buffer status in the USB. The buffer memory status can be confirmed using the DCPCTR.BSTS flag and the PIPEnCTR.INBUFM flag. The transfer direction for the buffer memory can be specified using either the PIPECFG.DIR bit or the CFIFOSEL.ISEL bit (when DCP is selected).

The INBUFM flag is valid for pipe 0 to pipe 5 in the transmitting direction.

When a transmitting pipe uses the double buffer configuration, software can read the BSTS flag to monitor the buffer memory status on the CPU side and the INBUFM flag to monitor the buffer memory status on the SIE side. When the BEMP interrupt may not show the buffer empty status because the write access to the FIFO port by the CPU or DMAC/DTC is slow, software can use the INBUFM flag to confirm the end of transmission.

Table 32.16 Buffer Status Indicated by the BSTS Flag

ISEL or DIR	BSTS	Buffer Memory Status
0 (receiving direction)	0	There is no received data, or data is being received. Reading from the FIFO port is disabled.
0 (receiving direction)	1	There is received data, or a zero-length packet has been received. Reading from the FIFO port is allowed. Note that when a zero-length packet is received, reading is not possible and the buffer must be cleared.
1 (transmitting direction)	0	The transmission has not been completed. Writing to the FIFO port is disabled.
1 (transmitting direction)	1	The transmission has been completed. CPU write is allowed.

Downloaded from Arrow.com.

Table 32.17 Buffer Status Indicated by the INBUFM Flag

DIR	INBUFM	Buffer Memory Status
0 (receiving direction)	Invalid	Invalid
1 (transmitting direction)	0	The transmission has been completed. There is no waiting data to be transmitted.
1 (transmitting direction)	1	The FIFO port has written data to the buffer. There is data to be transmitted.

32.3.5.2 FIFO Buffer Clearing

Table 32.18 shows the clearing of the FIFO buffer memory by the USB. The buffer memory can be cleared using the BCLR, DnFIFOSEL.DCLRM, and PIPEnCTR.ACLRM bit in the port control register.
Either a single or double buffer configuration can be selected for pipes 1 to 5, using the PIPECFG.DBLB bit.

Table 32.18 List of Buffer Clearing Methods

FIFO Buffer Clearing Mode	Clearing Buffer Memory on CPU Side	Mode for Automatically Clearing Buffer Memory after Reading Specified Pipe Data	Auto Buffer Clear Mode for Discarding All Received Packets
Register used	CFIFOCTR DnFIFOCTR	DnFIFOSEL	PIPEnCTR
Bit used	BCLR	DCLRM	ACLRM
Clearing condition	Cleared by writing 1	1: Mode valid 0: Mode invalid	1: Mode valid 0: Mode invalid

(1) Auto Buffer Clear Mode Function

With the USB, all received data packets are discarded if the PIPEnCTR.ACLRM bit is set to 1. If a correct data packet has been received, the ACK response is returned to the host controller. The auto buffer clear mode function can be set only in the buffer memory reading direction.
If the ACLRM bit is set to 1 and then to 0, the buffer memory of the selected pipe can be cleared regardless of the access direction.
An access cycle of at least 100 ns is required for the internal hardware sequence processing time between ACLRM = 1 and ACLRM = 0.

Downloaded from Arrow.com.

32.3.5.3 FIFO Port Functions

Table 32.19 shows the settings for the FIFO port functions of the USB. In write access, writing data until the maximum packet size is reached automatically enables transmission of the data. To enable transmission before the maximum packet size is reached, the BVAL bit in the port control register should be set to end writing. To send a zero-length packet, the BCLR bit in the register should be used to clear the buffer and then the BVAL bit set in order to end writing.
In reading, reception of new packets is automatically enabled when all data has been read. Data cannot be read when a zero-length packet has been received (the DTLN[8:0] flags = 0), so the BCLR bit in the register should be used to clear the buffer. The length of the receive data can be confirmed using the DTLN[8:0] flags in the port control register.

Table 32.19 FIFO Port Function Settings

Register	Bit	Description
CFIFOSEL, DnFIFOSEL (n = 0, 1)	RCNT	Selects DTLN[8:0] read mode.
	REW	Buffer memory rewind (re-read, rewrite).
	DCLRM	Automatically clears receive data for a specified pipe after the data has been read (only for DnFIFO).
	DREQE	Enables DMA/DTC transfers (only for DnFIFO).
	MBW	FIFO port access bit width.
	BIGEND	Selects FIFO port endian.
	ISEL	FIFO port access direction (only for DCP).
	CURPIPE[3:0]	Selects the current pipe.
CFIFOCTR, DnFIFOCTR (n = 0, 1)	BVAL	Ends writing to the buffer memory.
	BCLR	Clears the buffer memory on the CPU side.
	DTLN[8:0]	Checks the length of receive data.

(1) FIFO Port Selection

Table 32.20 shows the pipes that can be selected with the various FIFO ports. The pipe to be accessed should be selected using the CURPIPE[3:0] bits in the port select register. After the pipe is selected, whether the written value can be correctly read from the CURPIPE[3:0] bits should be checked. (If the previous pipe number is read, it indicates that the pipe modification is being executed by the USB controller.) Then, the FRDY flag in a port control register = 1 is checked.
In addition, the bus width to be accessed should be selected using the MBW bit in the port select register. The buffer memory access direction conforms to the PIPECFG.DIR bit. Only for the DCP, the ISEL bit in the port select register determines the direction.

Table 32.20 FIFO Port Access Categorized by Pipe

Pipe	Access Method	Port that can be Used
DCP	CPU access	CFIFO port register
Pipe 1 to pipe 9	CPU access	CFIFO port register D0FIFO/D1FIFO port register
	DMAC/DTC access	D0FIFO/D1FIFO port register

(2) REW Bit

It is possible to temporarily stop access to the pipe currently being accessed, access a different pipe, and then continue processing for the current pipe again. The REW bit in the port select register is used for this processing.
If a pipe is selected through the CURPIPE[3:0] bits in the port select register with the REW bit set to 1, the pointer used for reading from and writing to the buffer memory is reset, and reading or writing can be carried out from the first byte.
If a pipe is selected with 0 set for the REW bit, data can be read and written in continuation from the previous selection, without the pointer being reset.
To access the FIFO port, the FRDY flag in the port control register = 1 should be checked after selecting a pipe.

Downloaded from Arrow.com.

32.3.5.4 DMA Transfers (D0FIFO and D1FIFO Ports)

(1) Overview of DMA Transfers

For pipes 1 to 9, the FIFO port can be accessed using the DMAC. When accessing the buffer for the pipe targeted for DMA transfer is enabled, a DMA transfer request is issued.
The unit of transfer to the FIFO port should be selected using the DnFIFOSEL.MBW bit and the pipe targeted for the DMA transfer should be selected using the DnFIFOSEL.CURPIPE[3:0] bits. The selected pipe should not be changed during the DMA transfer.

(2) DnFIFO Auto Clear Mode (D0FIFO and D1FIFO Port Reading Direction)

If 1 is set in the DnFIFOSEL.DCLRM bit, the USB automatically clears the buffer memory of the selected pipe when reading of data from the buffer memory has been completed.
Table 32.21 shows the packet reception and buffer memory clearing processing by software for each of the various settings. As shown in Table 32.21, the buffer clearing conditions depend on the value set in the PIPECFG.BFRE bit. Using the DnFIFOSEL.DCLRM bit eliminates the need for the buffer to be cleared by software in any situation that requires buffer clearing. This enables DMA transfers without involving software.
The DnFIFO auto clear mode can be set only in the buffer memory reading direction.

Table 32.21 Packet Reception and Buffer Memory Clearing Processing by Software

Buffer Status When Packet is Received	Register Setting			
	DCLRM = 0		DCLRM = 1	
	BFRE = 0	BFRE = 1	BFRE = 0	BFRE = 1
Buffer full	Clearing is not necessary	Clearing is not necessary	Clearing is not necessary	Clearing is not necessary
Zero-length packet reception	Clearing is necessary	Clearing is necessary	Clearing is not necessary	Clearing is not necessary
Normal short packet reception	Clearing is not necessary	Clearing is necessary	Clearing is not necessary	Clearing is not necessary
Transaction count end	Clearing is not necessary	Clearing is necessary	Clearing is not necessary	Clearing is not necessary

Downloaded from Arrow.com.

32.3.6 Control Transfers Using DCP

In the data stage of control transfers, data is transferred using the default control pipe (DCP).
The DCP buffer memory is a 64-byte single buffer and is a fixed area that is shared for both control reading and control writing. The buffer memory can be accessed only through the CFIFO port.

32.3.6.1 Control Transfers When the Host Controller is Selected

(1) Setup Stage

Registers USBREQ, USBVAL, USBINDX, and USBLENG are the registers that are used to transmit a USB request for setup transactions. Writing setup packet data to the registers and writing 1 to the DCPCTR.SUREQ bit transmits the specified data for setup transactions. Upon completion of the transaction, the SUREQ bit is set to 0. The above USB request registers should not be modified while SUREQ = 1.
After the attached state of the connected function device is detected, the first setup transaction for the device should be issued by using the sequence described above with the DCPMAXP.DEVSEL[3:0] bits set to 0 and the DEVADD0.USBSPD[1:0] bits set appropriately.
After the connected function device is shifted to the Address state, setup transactions should be issued by using the sequence described above with the assigned USB address set in the DEVSEL[3:0] bits and the bits in the DEVADDn register corresponding to the specified USB address set appropriately. For example, when PIPEMAXP.DEVSEL[3:0] = 0010b, make appropriate settings in the DEVADD2 register; when PIPEMAXP.DEVSEL[3:0] = 0101b, make appropriate settings in the DEVADD5 register.
When the setup transaction data has been sent, an interrupt request is generated according to the response received from the peripheral device (SIGN or SACK flag in the INTSTS1 register), by means of which the result of the setup transactions can be confirmed.
A data packet of DATA0 (USB request) is transmitted as the data packet for a setup transaction regardless of the setting of the DCPCTR.SQMON flag.

(2) Data Stage

Data is transferred using the DCP buffer memory.
The access direction of the DCP buffer memory should be specified using the CFIFOSEL.ISEL bit. The transfer direction should be specified using the DCPCFG.DIR bit.
For the first data packet of the data stage, the data PID should be transferred as DATA1. Set data PID = DATA1 in the DCPCTR.SQSET bit and the PID[1:0] bits = 01b (BUF). Completion of data transfer is detected using the BRDY or BEMP interrupt.
For control write transfers, when the number of data bytes to be sent is an integer multiple of the maximum packet size, software should control so as to send a zero-length packet at the end.

(3) Status Stage

Zero-length packet data is transferred in the direction opposite to that in the data stage. As in the data stage, data is transferred using the DCP buffer memory. Transactions are done in the same manner as the data stage.
For the data packets of the status stage, the data PID should be set to DATA1 using the DCPCTR.SQSET bit.
For reception of a zero-length packet, the received data length should be confirmed using the CFIFOCTR.DTLN[8:0] flags after a BRDY interrupt is generated, and the buffer memory should then be cleared using the CFIFOCTR.BCLR bit.

Downloaded from Arrow.com.

32.3.6.2 Control Transfers When the Function Controller is Selected

(1) Setup Stage

The USB sends an ACK response for a correct setup packet targeted to the USB. The operation of the USB in the setup stage is described below.
When receiving a new setup packet, the USB sets the following bits.

- Set the INTSTS0.VALID flag to 1.
- Set the DCPCTR.PID[1:0] bits to 00b (NAK).
- Set the DCPCTR.CCPL bit to 0.

When receiving a data packet right after the setup packet, the USB stores the USB request parameters in registers USBREQ, USBVAL, USBINDX, and USBLENG.

Response processing with respect to the control transfer should be carried out after setting the VALID flag = 0. In the VALID flag = 1 state, PID[1:0] = 01b (BUF) cannot be set, and the data stage cannot be terminated.
Using the function of the VALID flag, the USB can suspend the current request processing when receiving a new USB request during a control transfer, and can send a response to the newest request.
In addition, the USB automatically detects the direction bit (bit 8 of bmRequestType) and the request data length (wLength) of the received USB request, distinguishes between control read transfer, control write transfer, and no-data control transfer, and controls stage transitions. For a wrong sequence, the sequence error of the control transfer stage transition interrupt is generated, and the software is notified of occurrence of the error. For the stage control of the USB, refer to Figure 32.12.

(2) Data Stage

Data transfers corresponding to received USB requests should be done using the DCP. Before accessing the DCP buffer memory, the access direction should be specified using the CFIFOSEL.ISEL bit.
If the transfer data is larger than the size of the DCP buffer memory, the data transfer should be carried out using the BRDY interrupt for control write transfers and the BEMP interrupt for control read transfers.

(3) Status Stage

Control transfers are terminated by setting the DCPCTR.CCPL bit to 1 while the DCPCTR.PID[1:0] bits are set to 01b (BUF).
After the above settings have been made, the USB automatically executes the status stage in accordance with the data transfer direction determined at the setup stage. The specific procedure is as follows.

- For control read transfers
  A zero-length packet is received from the USB host and an ACK response is sent.
- For control write transfers and no-data control transfers
  A zero-length packet is transmitted and an ACK response is received from the USB host.

(4) Control Transfer Auto Response Function

The USB automatically responds to a correct SET_ADDRESS request. If any of the following errors occurs in the SET_ADDRESS request, a response from the software is necessary.

- bmRequestType is not 00h: Any transfer other than a control write transfer
- wIndex is not 00h: Request error
- wIndex is not 00h: Any transfer other than a no-data control transfer
- wValue is larger than 7Fh: Request error
- INTSTS0.DVSQ[2:0] are 011b (Configured state): Control transfer of a device state error

For all requests other than the SET_ADDRESS request, a response is required from the corresponding software.

Downloaded from Arrow.com.

32.3.7 Bulk Transfers (Pipes 1 to 5)

The buffer memory usage (single/double buffer setting) can be selected for bulk transfers. The USB provides the following functions for bulk transfers.

- BRDY interrupt function (PIPECFG.BFRE bit: refer to section 32.3.3.1, (2) When the SOFCFG.BRDYM Bit = 0 and the PIPECFG.BFRE Bit = 1)
- Transaction count function
  (PIPEnTRE.TRENB and TRCLR bits and PIPEnTRN register: refer to section 32.3.4.5, Transaction Counter (For Pipes 1 to 5 in Reading Direction))
- Response PID = NAK function (PIPECFG.SHTNAK bit: refer to section 32.3.4.8, Response PID = NAK Function)
- Auto response mode (PIPEnCTR.ATREPM bit: refer to section 32.3.4.9, Auto Response Mode)

32.3.8 Interrupt Transfers (Pipes 6 to 9)

When the function controller is selected, the USB carries out interrupt transfers in accordance with the timing controlled by the host controller.
When the host controller is selected, the timing of issuing a token can be specified using the interval counter.

32.3.8.1 Interval Counter during Interrupt Transfers When the Host Controller is Selected

For interrupt transfers, intervals between transactions are set in the PIPEPERI.IITV[2:0] bits. The USB controller issues interrupt transfer tokens based on the specified intervals.

(1) Counter Initialization

The interval counter is initialized when the MCU is reset or when the PIPEnCTR.ACLRM bit is set to 1. Note that the PIPEPERI.IITV[2:0] bits are not initialized when the ACLRM bit is used for initialization.

Note that the interval counter is not initialized in the following case.

- USB bus reset or USB suspended
  The IITV[2:0] bits are not initialized. Setting 1 to the DVSTCTR0.UACT bit starts counting from the value before entering the USB bus reset state or USB suspended state.

(2) Operation When Transmission/Reception is Impossible at Token Issuance Timing

The USB cannot issue tokens even at token issuance timing in the following cases. In such a case, the USB attempts transactions at the subsequent interval.

- When the PID[1:0] bits are set to 00b (NAK) or 1xb (STALL).
- When the buffer memory is full at the token sending timing in the receiving (IN) direction.
- When there is no data to be sent in the buffer memory at the token sending timing in the transmitting (OUT) direction.

Downloaded from Arrow.com.

32.3.9 Isochronous Transfers (Pipes 1 and 2)

The USB has the following functions for isochronous transfers.

- Notification of isochronous transfer error information
- Interval counter (specified by the PIPEPERI.IITV[2:0] bits)
- Isochronous IN transfer data setup control (IDLY function)
- Isochronous IN transfer buffer flush function (specified by the PIPEPERI.IFIS bit)

32.3.9.1 Error Detection in Isochronous Transfers

The USB has a function for detecting the error information described below, so that when errors occur in isochronous transfers, they can be controlled by software. Table 32.22 and Table 32.23 show the priority in which errors are confirmed and the interrupts generated corresponding to errors.

(a) PID errors

- If the PID of the received packet is illegal.

(b) CRC errors and bit stuffing errors

- If an error occurs in the CRC of the received packet or the bit stuffing is illegal.

(c) Maximum packet size exceeded

- The data of the received packet is larger than the specified maximum packet size.

(d) Overrun and underrun errors

- When the host controller is selected
  When the buffer memory is full at the token sending timing in the IN (receiving) direction.
  When there is no data to be sent in the buffer memory at the token sending timing in the OUT (transmitting) direction.
- When the function controller is selected
  When there is no data to be sent in the buffer memory at the token receiving timing in the IN (transmitting) direction.
  When the buffer memory is full at the token receiving timing in the OUT (receiving) direction.

(e) Interval errors

An interval error is generated on any of the following conditions when the function controller is selected.

- During an isochronous IN transfer, an IN token could not be received in the interval frame.
- During an isochronous OUT transfer, an OUT token could not be received in the interval frame.

Table 32.22 Error Detection When a Token is Received

Detection Priority	Error	Generated Interrupt and Status
1	PID errors	No interrupts are generated in both cases when the host controller is selected and the function controller is selected (ignored as a corrupted packet).
2	CRC errors and bit stuffing errors	No interrupts generated in both cases when the host controller is selected and the function controller is selected (ignored as a corrupted packet).
3	Overrun and underrun errors	An NRDY interrupt is generated to set the FRMNUM.OVRN flag to 1 in both cases when the host controller is selected and function controller is selected. When the function controller is selected, a zero-length packet is transmitted in response to IN token. However, no data packets are received in response to OUT token.
4	Interval errors	An NRDY interrupt is generated when the function controller is selected. It is not generated when the host controller is selected.

Downloaded from Arrow.com.

Table 32.23 Error Detection When a Data Packet is Received

Detection Priority	Error	Generated Interrupt and Status
1	PID errors	No interrupts are generated (ignored as a corrupted packet).
2	CRC errors and bit stuffing errors	An NRDY interrupt is generated to set the FRMNUM.CRCE to 1 bit in both cases when the host controller is selected and the function controller is selected.
3	Maximum packet size exceeded errors	A BEMP interrupt is generated to set the PID[1:0] bits to 1xb (STALL) in both cases when the host controller is selected and the function controller is selected.

32.3.9.2 Data PID

When the function controller is selected, the USB operates as follows in response to the received PID.
IN direction

- DATA0: Sent as data packet PID
- DATA1: Not sent
- DATA2: Not sent
- mDATA: Not sent

OUT direction

- DATA0: Received normally as data packet PID
- DATA1: Received normally as data packet PID
- DATA2: Packets are ignored
- mDATA: Packets are ignored

32.3.9.3 Interval Counter

The isochronous transfer interval can be set using the PIPEPERI.IITV[2:0] bits. The interval counter enables the functions shown in Table 32.24 when the function controller is selected. When the host controller is selected, the USB generates the token issuance timing. When the host controller is selected, the interval counter operation is the same as that in the interrupt transfer.

Table 32.24 Interval Counter Function When the Function Controller is Selected

Transfer Direction	Function	Conditions for Detection
IN	Flushes transmit buffer	When an IN token cannot be successfully received in the interval frame during an isochronous IN transfer
OUT	Notifies that a token not being received	When an OUT token cannot be successfully received in the interval frame during an isochronous OUT transfer

The interval count is carried out when an SOF is received or for interpolated SOFs, so the isochronism can be maintained even if an SOF is damaged. The frame interval that can be set is the $2^{IITV[2:0]}$ frames.

Downloaded from Arrow.com.

(1) Counter Initialization When the Function Controller is Selected

The interval counter is initialized when the MCU is reset or when the PIPEnCTR.ACLRM bit is set to 1. Note that the PIPEPERI.IITV[2:0] bits are not initialized when the ACLRM bit is used for initialization.

After the interval counter has been initialized, counting is started under either of the following conditions 1 and 2 when a packet has been transferred successfully.

1. An SOF is received after transmission of data in response to an IN token while the PID[1:0] bits are 01b (BUF).
2. An SOF is received after reception of data of an OUT token while the PID[1:0] bits are 01b (BUF).

Note that the interval counter is not initialized under the following conditions.

- When the PID[1:0] bits are set to 00b (NAK) or 1xb (STALL)
  The interval timer does not stop. The USB attempts transactions at the subsequent interval.
- When the USB bus is reset or USB is suspended
  The IITV[2:0] bits are not initialized. When an SOF has been received, counting is restarted from the value prior to the reception of the SOF.

(2) Interval Counting and Transfer Control When the Host Controller is Selected

The USB controls the interval between token issuance operations based on the PIPEPERI.IITV[2:0] bit settings. Specifically, the USB issues a token for a selected pipe once every $2^{IITV[2:0]}$ frames.
The USB starts counting the token issuance interval at the frame following the frame in which the PID[1:0] bits have been set to 01b (BUF) by software.

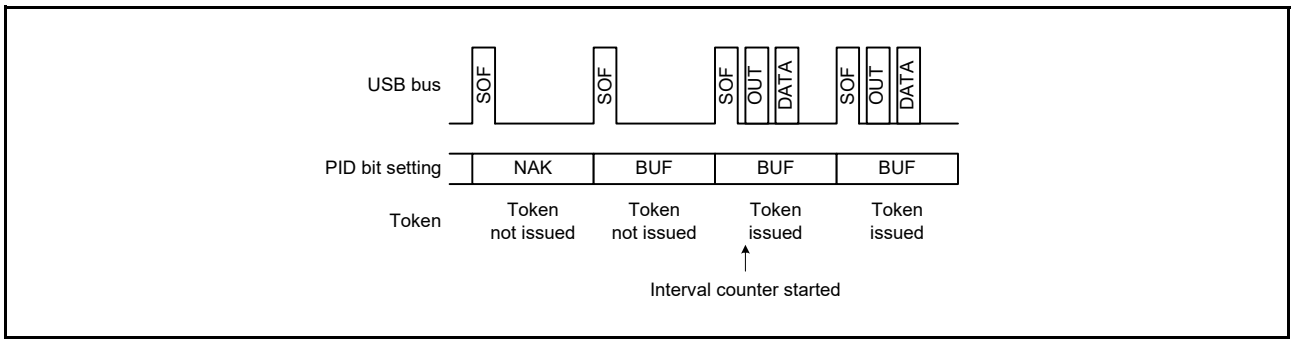


Figure 32.13 Token Issuance When IITV[2:0] = 000b

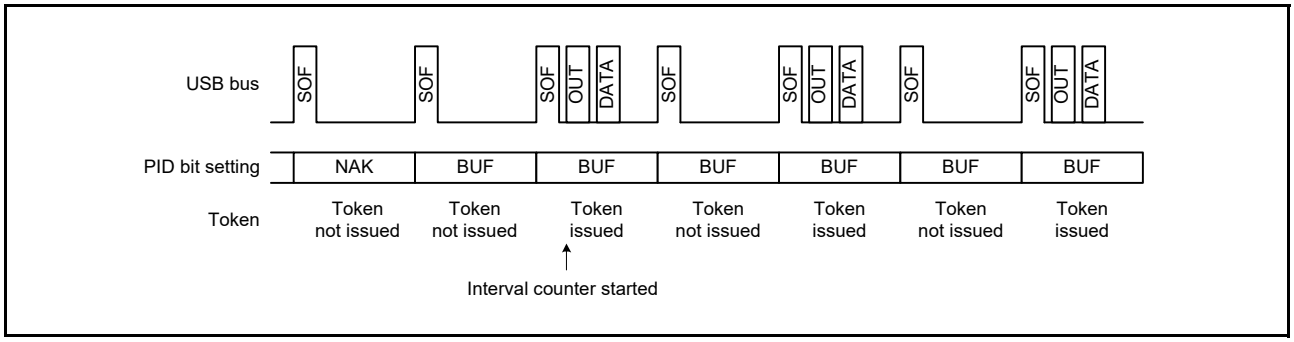


Figure 32.14 Token Issuance When IITV[2:0] = 001b

Downloaded from Arrow.com.

When the selected pipe is set for isochronous transfers, the USB carries out the following operation in addition to controlling the token issuance interval. The USB issues a token even when the NRDY interrupt generation condition is satisfied.

(a) When the selected pipe is for isochronous IN transfers

The USB generates an NRDY interrupt when the USB issues an IN token but does not receive a packet successfully from a peripheral device (no response or packet error).
The USB sets the FRMNUM.OVRN flag to 1 generating an NRDY interrupt when the time to issue an IN token comes while the USB cannot receive data because the FIFO buffer is full (due to the fact that the CPU or DMAC/DTC is too slow to read data from the FIFO buffer).

(b) When the selected pipe is for isochronous OUT transfers

The USB sets the OVRN flag to 1 generating an NRDY interrupt and transmitting a zero-length packet when the time to issue an OUT token comes while there is no data to be transmitted in the FIFO buffer (because the CPU or DMAC/DTC is too slow to write data to the FIFO buffer).
The token issuance interval is reset on any of the following conditions.

- When the USB is reset through a reset pin
  (The IITV[2:0] bits are also set to 0).
- When the PIPEnCTR.ACLRM bit has been set to 1 by software.

(3) Interval Counting and Transfer Control When the Function Controller is Selected

(a) When the selected pipe is for isochronous OUT transfers

The USB generates an NRDY interrupt when the USB fails to receive a data packet within the interval set by the PIPEPERI.IITV[2:0] bits.
The USB also generates an NRDY interrupt when the USB fails to receive data because of a CRC error or other errors contained in the data packet or because of the FIFO buffer being full.
The NRDY interrupt is generated at the timing of SOF packet reception. Even if the SOF packet is corrupted, the internal interpolation allows the interrupt to be generated at the timing to receive the SOF packet.
However, when the IITV[2:0] bits are set to a value other than 0, the USB generates an NRDY interrupt on receiving an SOF packet for every interval after starting interval counting operation.
When the PID[1:0] bits are set to 00b (NAK) by software after starting the interval timer, the USB does not generate an NRDY interrupt on receiving an SOF packet.
The timing to start interval counting depends on the setting of IITV[2:0] bits as shown below.

- When the IITV[2:0] = 000b: The interval counting starts when software has set the PID[1:0] bits for the selected pipe to 01b (BUF).

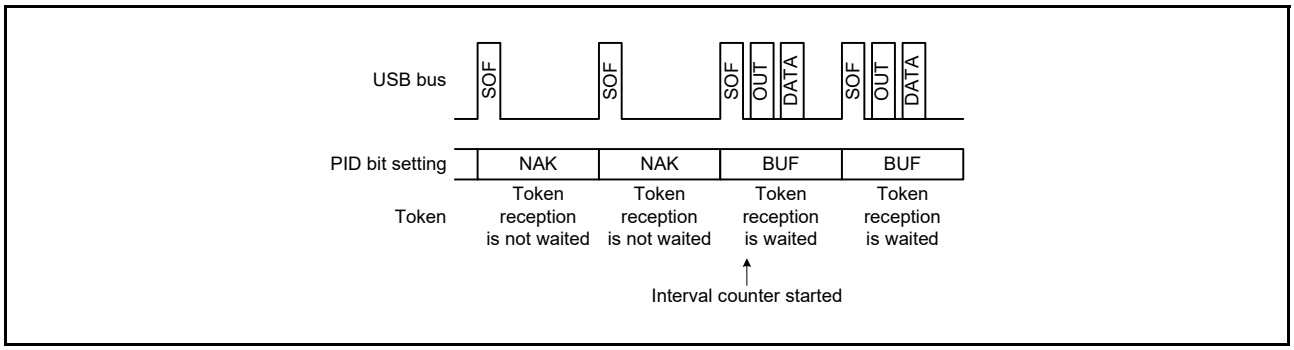


Figure 32.15 Relationship between Frames and Expected Token Reception When IITV[2:0] = 000b

Downloaded from Arrow.com.

- When the IITV[2:0] ≠ 000b: The interval counting starts on completion of successful reception of the first data packet after the PID[1:0] bits for the selected pipe have been modified to 01b (BUF).

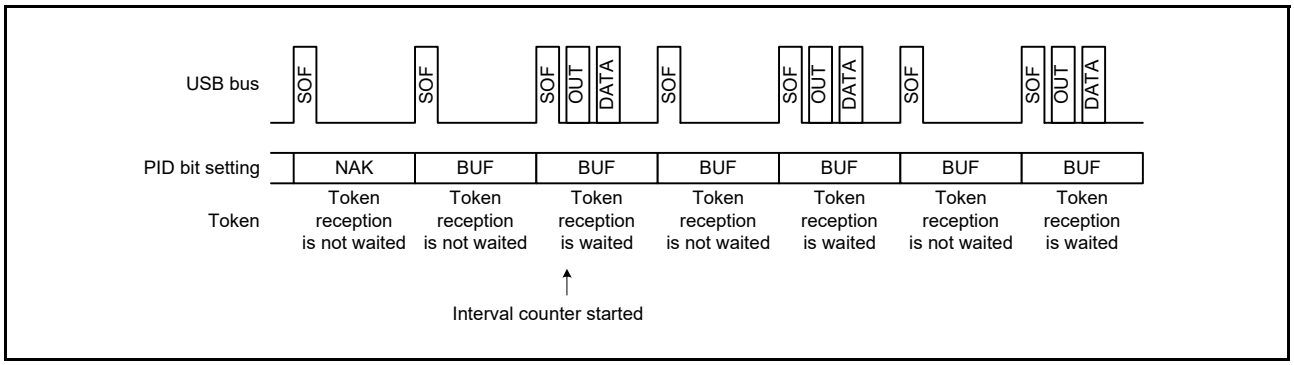


Figure 32.16 Relationship between Frames and Expected Token Reception When IITV[2:0] ≠ 000b

(b) When the selected pipe is for isochronous IN transfers

The PIPEPERI.IFIS bit should be 1 for this use. When IFIS = 0, the USB transmits a data packet in response to the received IN token irrespective of the setting of the PIPEPERI.IITV[2:0] bits.
When IFIS = 1, the USB clears the FIFO buffer when the USB fails to receive an IN token in the frame at the interval set by the IITV[2:0] bits while there is data to be transmitted in the FIFO buffer.
The USB also clears the FIFO buffer when the USB fails to receive an IN token successfully because of a bus error such as a CRC error contained in the IN token.
The FIFO buffer is cleared at the timing of SOF packet reception. Even if the SOF packet is corrupted, the internal interpolation allows the FIFO buffer to be cleared at the timing to receive the SOF packet.

The timing to start interval counting depends on the setting of the IITV[2:0] bits (similar to the timing during OUT transfers).
The interval is counted on any of the following conditions in function controller mode.

- When a hardware-reset is applied to the USB (here, the IITV[2:0] bits are also set to 000b).
- When the PIPEnCTR.ACLRM bit is set to 1 by software.
- When the USB detects a USB bus reset.

Downloaded from Arrow.com.

(4) Setup of Data to be Transmitted Using Isochronous Transfer When the Function Controller is Selected

With isochronous data transmission using the USB in the function controller, after data has been written to the buffer memory, a data packet can be transmitted with the next frame after the frame in which an SOF packet is detected. This function is called the isochronous transfer transmission data setup function, and it makes it possible to designate the frame from which transmission began.

In a double buffer configuration, even after the writing of data to both buffers has been completed, transmission will be enabled for only one buffer to which data writing was completed first. Accordingly, even if multiple IN tokens are received, only one packet of data is transmitted from a single buffer.

When an IN token is received, if the buffer memory is in the transmission enabled state, the USB transmits data as a normal response. If the buffer memory is not in the transmission enabled state, however, a zero-length packet is sent and an underrun error occurs.

Figure 32.17 shows an example of transmission using the isochronous transfer transmission data setup function with the USB when IITV[2:0] = 000b (every frame) has been set.

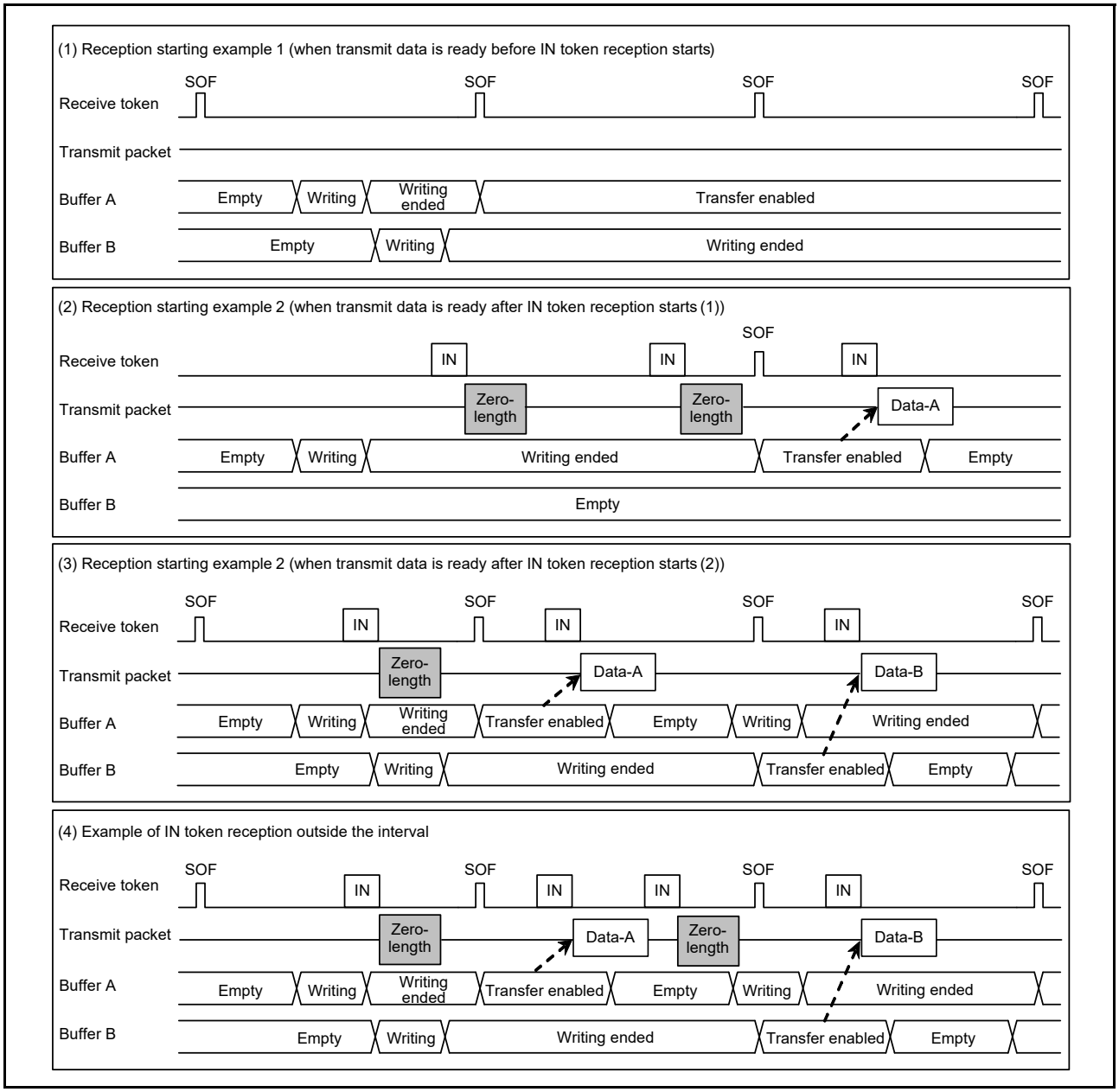


Figure 32.17 Example of Data Setup Function Operation

Downloaded from Arrow.com.

(5) Isochronous Transfer Transmission Buffer Flush When the Function Controller is Selected

If an SOF packet of the next frame is received without receiving an IN token in an interval frame during isochronous data transmission, the USB operates as if an IN token had been corrupted, and clears the buffer for which transmission is enabled, putting that buffer in the writing enabled state.
If a double buffer configuration is used and writing to both buffers has been completed, the buffer memory that was cleared is assumed as the data having been sent in the interval frame, and transmission is enabled for the buffer memory that is not cleared with SOF packet reception.
The timing of the buffer flush function depends on the setting of the PIPEPERI.IITV[2:0] bits.

- When the IITV[2:0] = 000b
  The buffer flush operation starts from the next frame after the pipe becomes valid.
- When the IITV[2:0] ≠ 000b
  The buffer flush operation is carried out after the first successful transaction.

Figure 32.18 shows an example of the buffer flush function in the USB. When an unanticipated token is received before the interval frame, the USB sends the write data or a zero-length packet as an underrun error according to the data setup state.

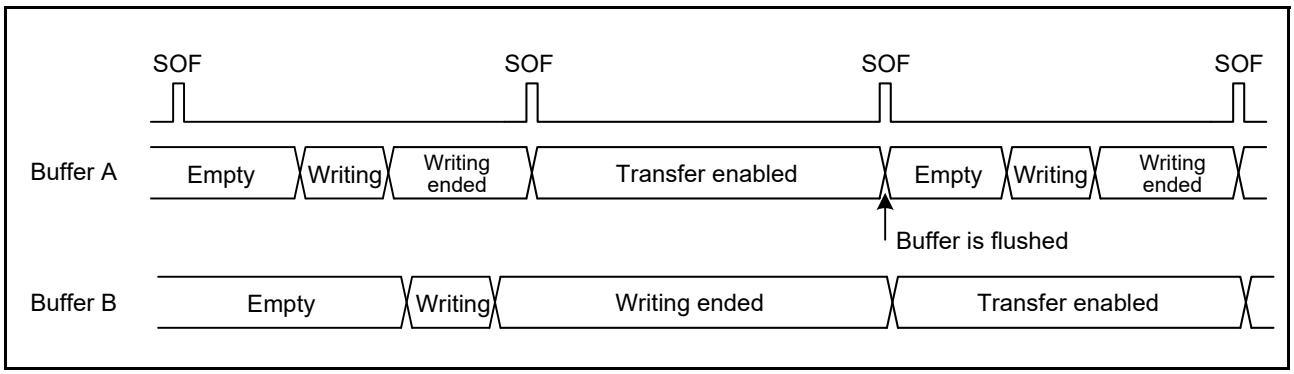


Figure 32.18 Example of Buffer Flush Operation

Downloaded from Arrow.com.

Figure 32.19 shows an example of interval error occurrence in the USB. There are five types of interval errors, as shown below. The interval error is generated at the timing indicated by ① in the figure, and the buffer flush function is activated.

If an interval error occurs during an IN transfers, the buffer flush function is activated; if it occurs during an OUT transfer, an NRDY interrupt is generated.

The FRMNUM.OVRN flag should be used to distinguish between NRDY interrupts such as received packet errors and overrun errors.

In response to tokens that are shaded in the figure, responses are sent according to the buffer memory status.

IN direction

- If the buffer is in the transmission enabled state, the data is transferred as a normal response.
- If the buffer is in the transmission disabled state, a zero-length packet is sent and an underrun error occurs.

OUT direction

- If the buffer is in the reception enabled state, the data is received as a normal response.
- If the buffer is in the reception disabled state, the data is discarded and an overrun error occurs.

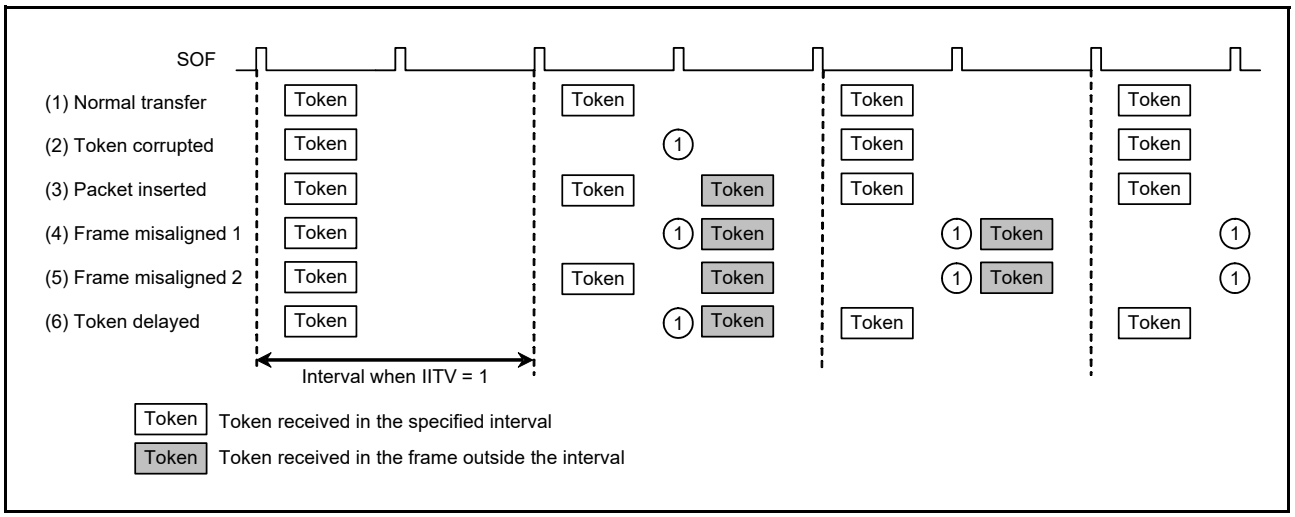


Figure 32.19 Example of Interval Error Occurrence When IITV[2:0] = 001b

Downloaded from Arrow.com.

32.3.10 SOF Interpolation Function

When the function controller is selected and if data could not be received at intervals of 1 ms because an SOF packet was corrupted or missing, the USB interpolates the SOF. The SOF interpolation operation begins when the USBE and SCKE bits in the SYSCFG register have been set to 1 and an SOF packet is received. The interpolation function is initialized under the following conditions.

- MCU reset
- USB bus reset
- Suspended state detected

The SOF interpolation operates as follows.

- The interpolation function is not activated until an SOF packet is received.
- After the first SOF packet is received, interpolation is carried out by counting 1 ms with an internal clock of 48 MHz.
- After the second and subsequent SOF packets are received, interpolation is carried out at the previous reception interval.
- Interpolation is not carried out in the suspended state or while a USB bus reset is being received.

The USB supports the following functions based on the SOF packet reception. These functions also operate normally with SOF interpolation, if the SOF packet was missing.

- Updating of the frame number
- SOFR interrupt timing
- Isochronous transfer interval count

If an SOF packet is missing, the FRMNUM.FRNM[10:0] flags are not updated.

Downloaded from Arrow.com.

32.3.11 Pipe Schedule

32.3.11.1 Conditions for Generating a Transaction

When the host controller is selected and the DVSTCTR0.UACT bit has been set to 1, the USB generates a transaction under the conditions shown in Table 32.25.

Table 32.25 Conditions for Generating a Transaction

	Conditions for Generation				
Transaction	**DIR**	**PID[1:0]**	**IITV[0]**	**Buffer State**	**SUREQ**
Setup	—*1	—*1	—*1	—*1	1 setting
Control transfer data stage, status stage, bulk transfer	IN	BUF	Invalid	Receive area exists	—*1
	OUT	BUF	Invalid	Transmit data exists	—*1
Interrupt transfer	IN	BUF	Valid	Receive area exists	—*1
	OUT	BUF	Valid	Transmit data exists	—*1
Isochronous transfer	IN	BUF	Valid	*2	—*1
	OUT	BUF	Valid	*3	—*1

Note 1. Symbols (—) in the table indicate that the condition is unrelated to the generating of tokens. "Valid" indicates that, for interrupt transfers and isochronous transfers, a transaction is generated only in transfer frames that are based on the interval counter. "Invalid" indicates that a transaction is generated regardless of the interval counter.
Note 2. This indicates that a transaction is generated regardless of whether there is a receive area. If there is no receive area, however, the received data is discarded.
Note 3. This indicates that a transaction is generated regardless of whether there is any data to be transmitted. If there is no data to be transmitted, however, a zero-length packet is transmitted.

32.3.11.2 Transfer Schedule

This section describes the transfer scheduling within a frame of the USB. After the USB sends an SOF, the transfer is carried out in the sequence described below.

1. Execution of periodic transfers
   A pipe is searched in the order of Pipe 1 → Pipe 2 → Pipe 6 → Pipe 7 → Pipe 8 → Pipe 9, and then, if there is a pipe for which an isochronous or interrupt transfer transaction can be generated, the transaction is generated.
2. Setup transactions for control transfers
   The DCP is checked, and if a setup transaction is possible, it is sent.
3. Execution of bulk transfers, control transfer data stages, and control transfer status stages
   A pipe is searched in the order of DCP → Pipe 1 → Pipe 2 → Pipe 3 → Pipe 4 → Pipe 5, and then, if there is a pipe for which a transaction for a bulk transfer, a control transfer data stage, or a control transfer status stage can be generated, the transaction is generated.
   When a transaction is generated, processing moves to the next pipe transaction regardless of whether the response from the peripheral device is ACK or NAK. If there is time for transfer within the frame, step 3 is repeated.

32.3.11.3 Enabling USB Communication

Setting the DVSTCTR0.UACT bit to 1 initiates SOF transmission and transaction generation is enabled.
Setting the UACT bit to 0 stops SOF transmission and a suspend state is entered. If the setting of the UACT bit is changed from 1 to 0, processing stops after the next SOF is sent.

Downloaded from Arrow.com.

32.4 Usage Notes

32.4.1 Setting the Module-Stop Function

Operation of the USB module can be disabled or enabled by setting a bit in the module stop control register B (MSTPCRB). The USB is initially disabled after a reset. Registers in the USB only become accessible after it has been released from the module-stop state. For details, refer to section 11, Low Power Consumption.

Downloaded from Arrow.com.

32.5 Battery Charging Detection Processing

It is possible to control the processing for data contact detection (D+ line contact check), primary detection (charger detection), and secondary detection (charger verification), which are defined in the battery charging specification. The following describes required operations for a function device and a host device, individually.

32.5.1 Processing When Function Controller is Selected

The following processing is required when operating the USB module as a portable device for battery charging.

(1) Detect when the data lines (D+ and D–) have made contact and start the processing for primary detection.
(2) After primary detection starts, wait 40 ms for masking, and then check the D– voltage level to confirm the primary detection result.
(3) If the charger is detected during primary detection, also start secondary detection.
(4) After secondary detection starts, wait 40 ms for masking, and then check the D+ voltage level to confirm the secondary detection result.

For step (1), after VBUS is detected using the VBIT interrupt and the VBSTS flag, wait for 300 to 900 ms by software, and then set the VDPSRCE0 and IDMSINKE0 bits in the USBBCCTRL0 register to 1. Or set the IDPSRCE0 bit to 1, and after a change from high to low on the D+ line is detected using the LNST[1:0] flags, set the IDPSRCE0 bit to 0 and set the VDPSRCE0 and IDMSINKE0 bits to 1. Set the VDPSRCE0 and IDMSINKE0 bits to 0 at the same time.*1

For step (2), set the VDPSRCE0 and IDMSINKE0 bits to 1 and wait 40 ms by software, and then use the CHGDETSTS0 flag to verify the primary detection result.*2

For step (3), if the CHGDETSTS0 flag is set to 1 in step (2), verify that the charger is detected, and then set the VDPSRCE0 and IDMSINKE0 bits to 0 and set the VDMSRCE0 and IDPSINKE0 bits to 1.

For step (4), set the VDMSRCE0 and IDPSINKE0 bits to 1 and wait for 40 ms by software, and then use the PDDETSTS0 flag to verify the secondary detection result.

The following shows the process flow.

Note 1. The battery charging specification describes two implementation methods of the process flow for data contact detection (D+/D– line contact check). One of the methods is to detect a change to logic low due to the pull-down resistor of the host device when the D+ and D– lines have made contact with the target while the D+ line is held at logic high by applying a current of 7 to 13 µA on the D+ line. The other method is to wait for 300 to 900 ms after VBUS is detected.
Note 2. During primary detection, when the voltage on the D– line is detected to be 0.25 to 0.4 V or above and 0.8 to 2.0 V or below, the target device is recognized as the host device for battery charging (charging downstream port). When using a PHY in which the CHGDETSTS0 flag only indicates that the voltage on the D– line is 0.25 to 0.4 V or above, add the processing to check that the voltage on D– line is 0.8 V to 2.0 V or below using the LNST[1:0] flags, as necessary.

Downloaded from Arrow.com.

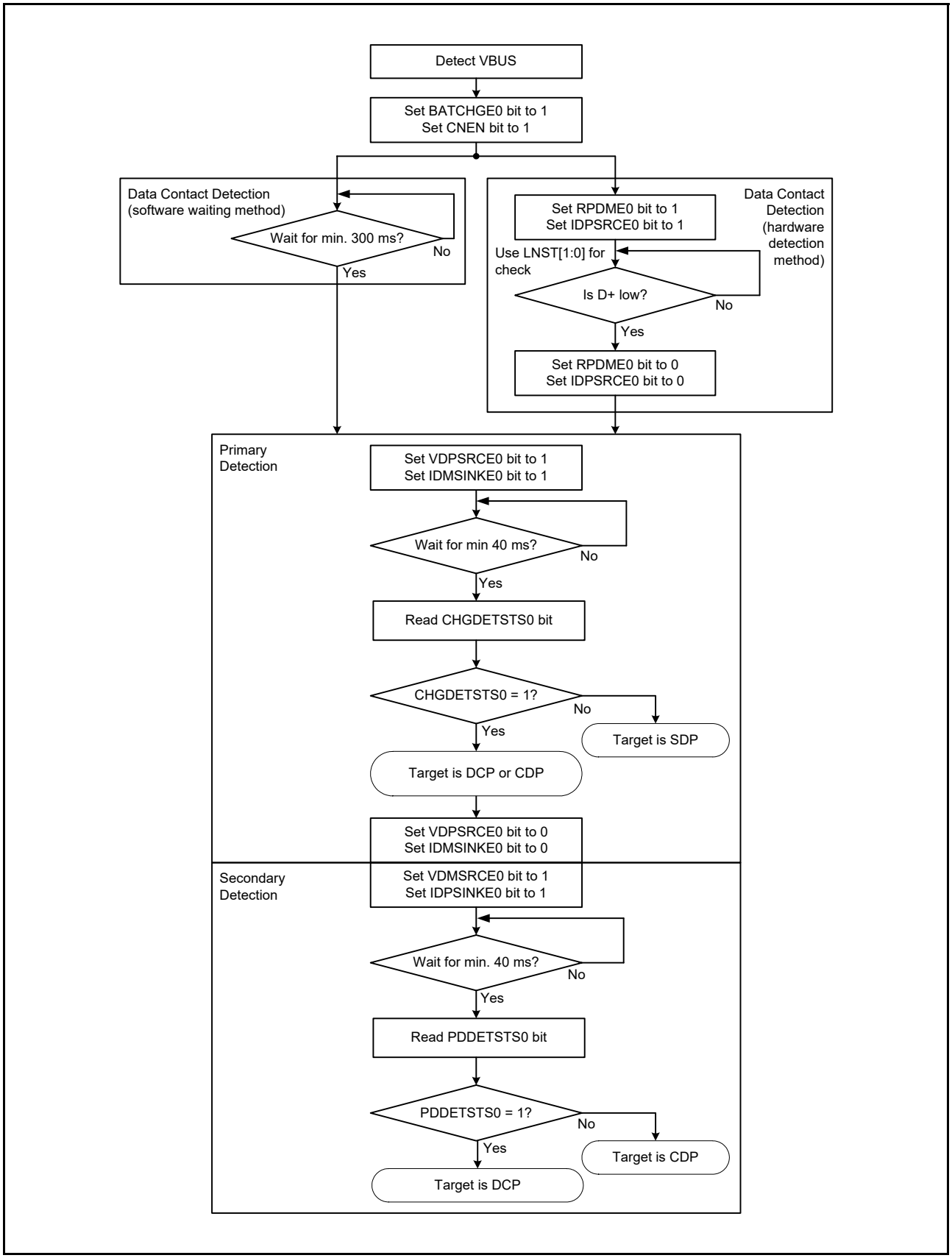


Figure 32.20 Process Flow for Operating as Portable Device

Downloaded from Arrow.com.

32.5.2 Processing When Host Controller is Selected

The following processing is required when operating the USB module as a charging downstream port for battery charging.

(1) Start driving the VBUS.
(2) Enable the portable device detection circuit.
(3) Monitor the portable device detection signal, and start driving the D– line if the detection signal is high.
(4) Detect when the portable device detection signal is low level and stop driving the D– line.

Or, the following processing can also be used in accordance with the battery charging specification.
(A) After disconnection is detected, start driving the D– line within 200 ms.
(B) After connection is detected, stop driving the D– line within 10 ms.

The D– line must be driven to allow the portable device to detect the primary detection described in section 32.5.1, Processing When Function Controller is Selected. The above steps (1) to (4) apply when the portable device detection function is provided by hardware. This method is to drive the D– line when the portable device is detected. Steps (A) and (B) apply when the portable device function is not provided or available by hardware. Regardless of detection of the portable device, the D– line is driven in the disconnected state and the line is not driven in the connected state. In the battery charging specification, either of these methods can be used.

For steps (3) and (4), after a change in the portable device detection signal is detected using the PDDETINT interrupt, the current signal state can be confirmed by reading the PDDETSTS0 flag.

Steps (A) and (B) can be performed only in a software timer.

The following show the process flow for steps (1) to (4) and the process flow for steps (A) to (B), respectively.

Downloaded from Arrow.com.

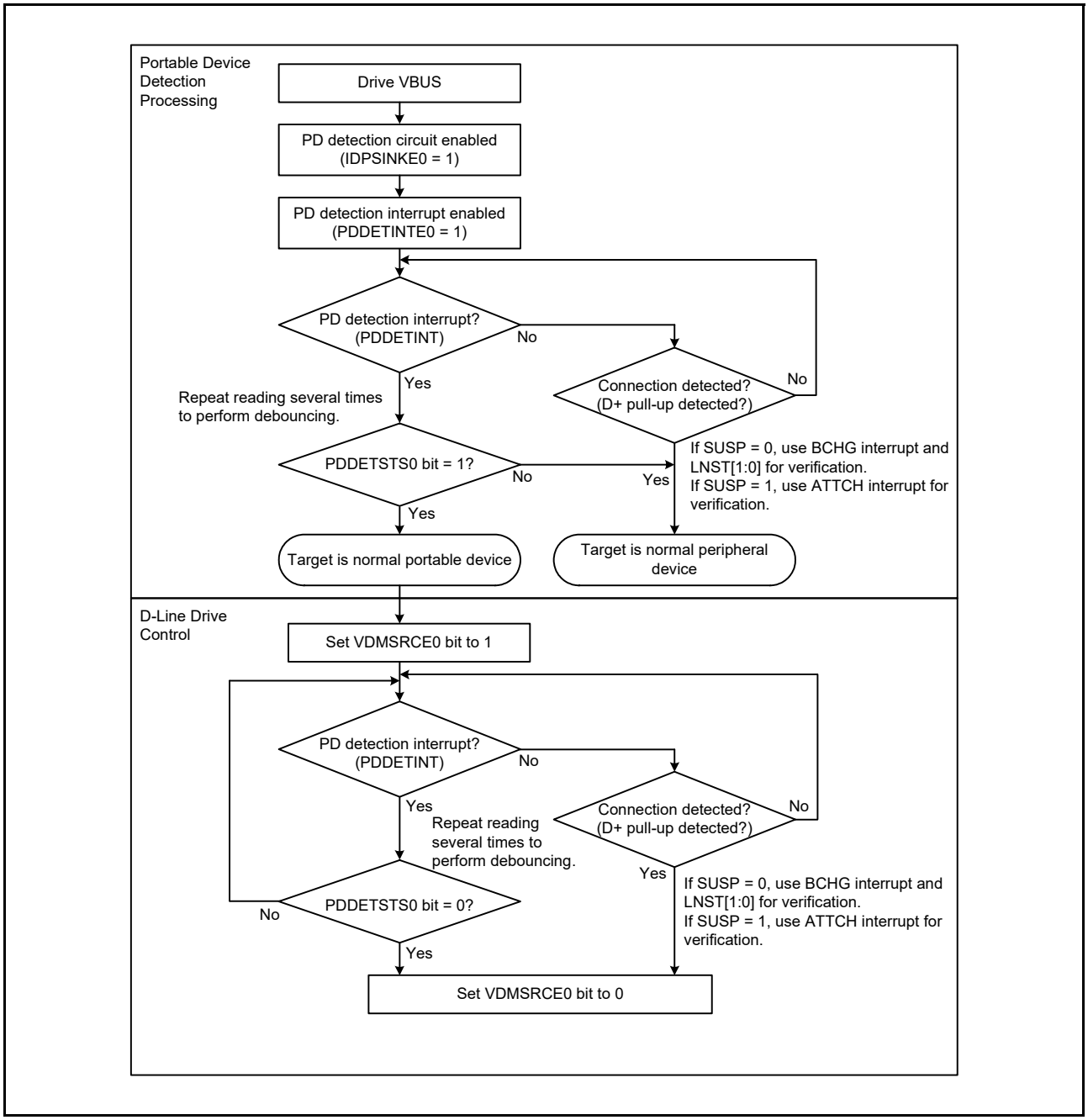


Figure 32.21 Process Flow for Operating as Charging Downstream Port (Steps (1) to (4))

Downloaded from Arrow.com.

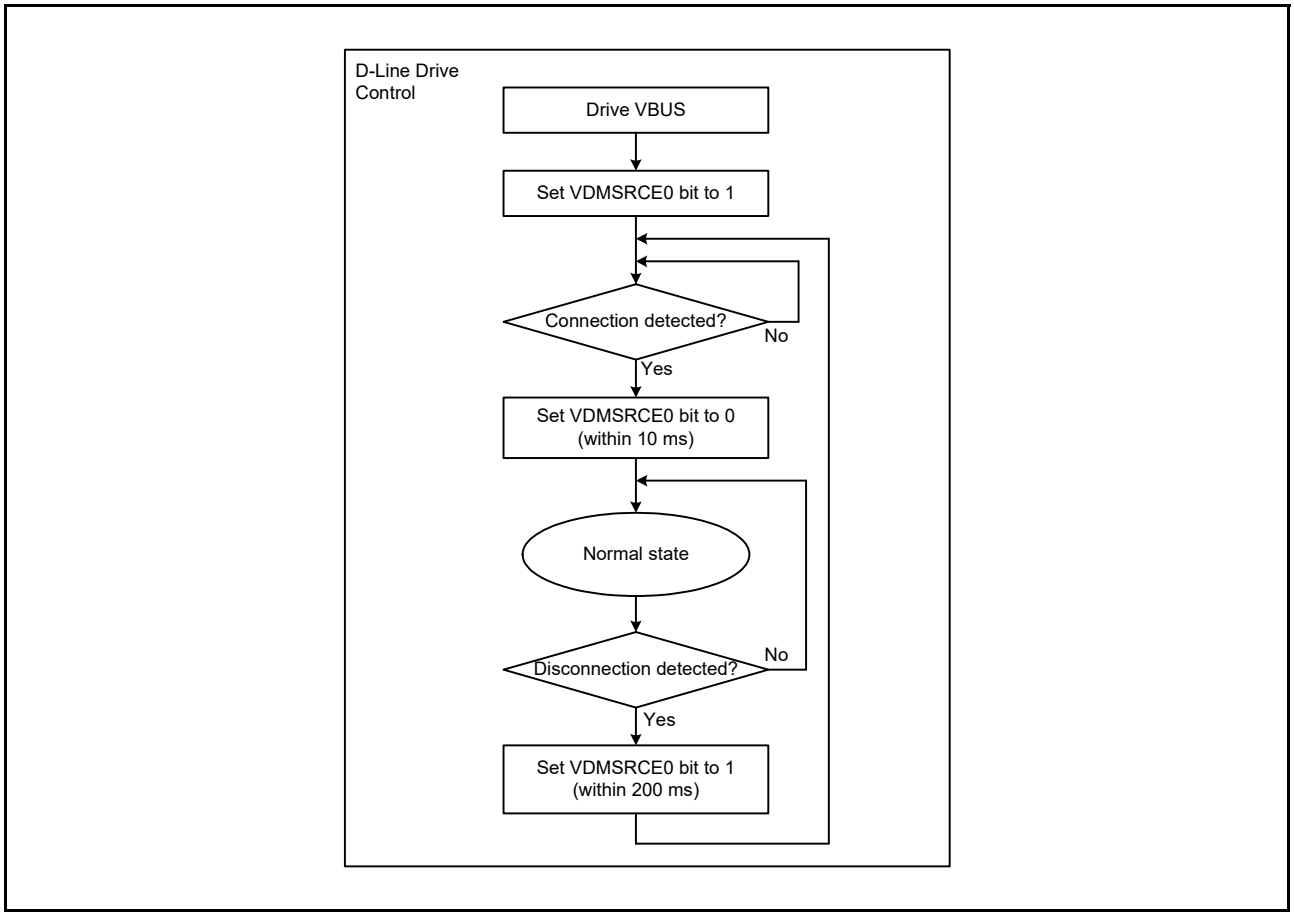


Figure 32.22 Process Flow for Operating as Charging Downstream Port (Steps (A) to (B))

Downloaded from Arrow.com.

33. Serial Communications Interface (SCIg, SCIh)

This MCU has four independent serial communications interface (SCI) channels. The SCI consists of the SCIg module (SCI1, SCI5, and SCI8) and the SCIh module (SCI12).

The SCIg module (SCI1, SCI5, and SCI8) can handle both asynchronous and clock synchronous serial communications. Asynchronous serial data communications can be carried out with standard asynchronous communications chips such as a Universal Asynchronous Receiver/Transmitter (UART) or Asynchronous Communications Interface Adapter (ACIA). As an extended function in asynchronous communications mode, the SCI also supports smart card (IC card) interfaces conforming to ISO/IEC 7816-3 (standard for Identification Cards). The SCI is also supports simple SPI interfaces, and simple I2C-bus interfaces when configured for single-master systems.

The SCIh module includes the functions of the SCIg module, and supports an extended serial communication protocol formed of Start Frames and Information Frames.

In this section, “PCLK” is used to refer to PCLKB.

33.1 Overview

Table 33.1 lists the specifications of the SCIg module, Table 33.2 lists the specifications of the SCIh module, and Table 33.3 lists the specifications of the individual SCI channels.

Figure 33.1 shows the block diagram of SCI1 and SCI8, Figure 33.2 shows the block diagram of SCI5, and Figure 33.3 shows the block diagram of SCI12 (SCIh).

Table 33.1 SCIg Specifications (1/2)

Item		Description
Serial communication modes		• Asynchronous • Clock synchronous • Smart card interface • Simple I2C-bus • Simple SPI bus
Transfer speed		Bit rate specifiable with the on-chip baud rate generator.
Full-duplex communications		Transmitter: Continuous transmission possible using double-buffer structure. Receiver: Continuous reception possible using double-buffer structure.
I/O pins		Refer to Table 33.4 to Table 33.6.
Data transfer		Selectable as LSB first or MSB first transfer*1
Interrupt sources		Transmit end, transmit data empty, receive data full, and receive error Completion of generation of a start condition, restart condition, or stop condition (for simple I2C mode)
Low power consumption function		Module stop state can be set for each channel.
Asynchronous mode	Data length	7, 8, or 9 bits
	Transmission stop bit	1 or 2 bits
	Parity	Even parity, odd parity, or no parity
	Receive error detection	Parity, overrun, and framing errors
	Hardware flow control	CTSn# and RTSn# pins can be used in controlling transmission/reception.
	Start-bit detection	Low level or falling edge is selectable.
	Break detection	When a framing error occurs, a break can be detected by reading the RXDn pin level directly.
	Clock source	An internal or external clock can be selected. Transfer rate clock input from the TMR can be used. (SCI5)
	Double-speed mode	Baud rate generator double-speed mode is selectable.
	Multi-processor communications function	Serial communication among multiple processors
	Noise cancellation	The signal paths from input on the RXDn pins incorporate digital noise filters.

Downloaded from Arrow.com.

Table 33.1 SCIg Specifications (2/2)

Item		Description
Clock synchronous mode	Data length	8 bits
	Receive error detection	Overrun error
	Hardware flow control	CTSn# and RTSn# pins can be used in controlling transmission/reception.
Smart card interface mode	Error processing	An error signal can be automatically transmitted when detecting a parity error during reception
		Data can be automatically retransmitted when receiving an error signal during transmission
	Data type	Both direct convention and inverse convention are supported.
Simple I2C mode	Transfer format	I2C-bus format
	Operating mode	Master (single-master operation only)
	Transfer rate	Fast mode is supported (refer to section 33.2.11, Bit Rate Register (BRR) to set the transfer rate).
	Noise cancellation	The signal paths from input on the SSCLn and SSDAn pins incorporate digital noise filters, and the interval for noise cancellation is adjustable.
Simple SPI bus	Data length	8 bits
	Detection of errors	Overrun error
	SS input pin function	Applying the high level to the SSn# pin can cause the output pins to enter the high-impedance state.
	Clock settings	Four kinds of settings for clock phase and clock polarity are selectable.
Bit rate modulation function		Correction of outputs from the on-chip baud rate generator can reduce errors.
Event link function (supported by SCI5 only)		Error (receive error or error signal detection) event output
		Receive data full event output
		Transmit data empty event output
		Transmit end event output

Note 1. In simple I2C mode, only MSB first is available.

Table 33.2 SCIh Specifications (1/2)

Item	Description
Serial communication modes	• Asynchronous • Clock synchronous • Smart card interface • Simple I2C-bus • Simple SPI bus
Transfer speed	Bit rate specifiable with the on-chip baud rate generator.
Full-duplex communications	Transmitter: Continuous transmission possible using double-buffer structure. Receiver: Continuous reception possible using double-buffer structure.
I/O pins	Refer to Table 33.4 to Table 33.7.
Data transfer	Selectable as LSB first or MSB first transfer*1
Interrupt sources	Transmit end, transmit data empty, receive data full, and receive error Completion of generation of a start condition, restart condition, or stop condition (for simple I2C mode)
Low power consumption function	Module stop state can be set.

Downloaded from Arrow.com.

Table 33.2 SCIh Specifications (2/2)

Item		Description
Asynchronous mode	Data length	7, 8, or 9 bits
	Transmission stop bit	1 or 2 bits
	Parity	Even parity, odd parity, or no parity
	Receive error detection	Parity, overrun, and framing errors
	Hardware flow control	CTSn# and RTSn# pins can be used in controlling transmission/reception.
	Start-bit detection	Low level or falling edge is selectable.
	Break detection	When a framing error occurs, a break can be detected by reading the RXDn pin level directly.
	Clock source	An internal or external clock can be selected. Transfer rate clock input from the TMR can be used.
	Double-speed mode	Baud rate generator double-speed mode is selectable.
	Multi-processor communications function	Serial communication among multiple processors
	Noise cancellation	The signal paths from input on the RXDn pins incorporate digital noise filters.
Clock synchronous mode	Data length	8 bits
	Receive error detection	Overrun error
	Hardware flow control	CTSn# and RTSn# pins can be used in controlling transmission/reception.
Smart card interface mode	Error processing	An error signal can be automatically transmitted when detecting a parity error during reception
		Data can be automatically retransmitted when receiving an error signal during transmission
	Data type	Both direct convention and inverse convention are supported.
Simple I^2C mode	Transfer format	I^2C-bus format
	Operating mode	Master (single-master operation only)
	Transfer rate	Fast mode is supported (refer to section 33.2.11, Bit Rate Register (BRR) to set the transfer rate).
	Noise cancellation	The signal paths from input on the SSCLn and SSDAn pins incorporate digital noise filters, and the interval for noise cancellation is adjustable.
Simple SPI bus	Data length	8 bits
	Detection of errors	Overrun error
	SS input pin function	Applying the high level to the SSn# pin can cause the output pins to enter the high-impedance state.
	Clock settings	Four kinds of settings for clock phase and clock polarity are selectable.
Extended serial mode	Start Frame transmission	• Output of a low level as the Break Field over a specified width and generation of interrupts on completion • Detection of bus collisions and the generation of interrupts on detection
	Start Frame reception	• Detection of the Break Field low width and generation of an interrupt on detection • Comparison of Control Fields 0 and 1 and generation of an interrupt when the two match • Two kinds of data for comparison (primary and secondary) can be set in Control Field 1. • A priority interrupt bit can be set in Control Field 1. • Handling of Start Frames that do not include a Break Field • Handling of Start Frames that do not include a Control Field 0 • Function for measuring bit rates
	I/O control function	• Selectable polarity for TXDX12 and RXDX12 signals • Selection of a digital filter for the RXDX12 signal • Half-duplex operation employing RXDX12 and TXDX12 signals multiplexed on the same pin • Selectable timing for the sampling of data received through RXDX12
	Timer function	• Usable as a reload timer
Bit rate modulation function		Correction of outputs from the on-chip baud rate generator can reduce errors.

Note 1. In simple I^2C mode, only MSB first is available.

Downloaded from Arrow.com.

Table 33.3 Functions of SCI Channels

Item	SCI1, SCI8	SCI5	SCI12
Asynchronous mode	Available	Available	Available
Clock synchronous mode	Available	Available	Available
Smart card interface mode	Available	Available	Available
Simple I²C mode	Available	Available	Available
Simple SPI mode	Available	Available	Available
Extended serial mode	Not available	Not available	Available
TMR clock input	Not available	Available	Available
Event link function	Not available	Available	Not available

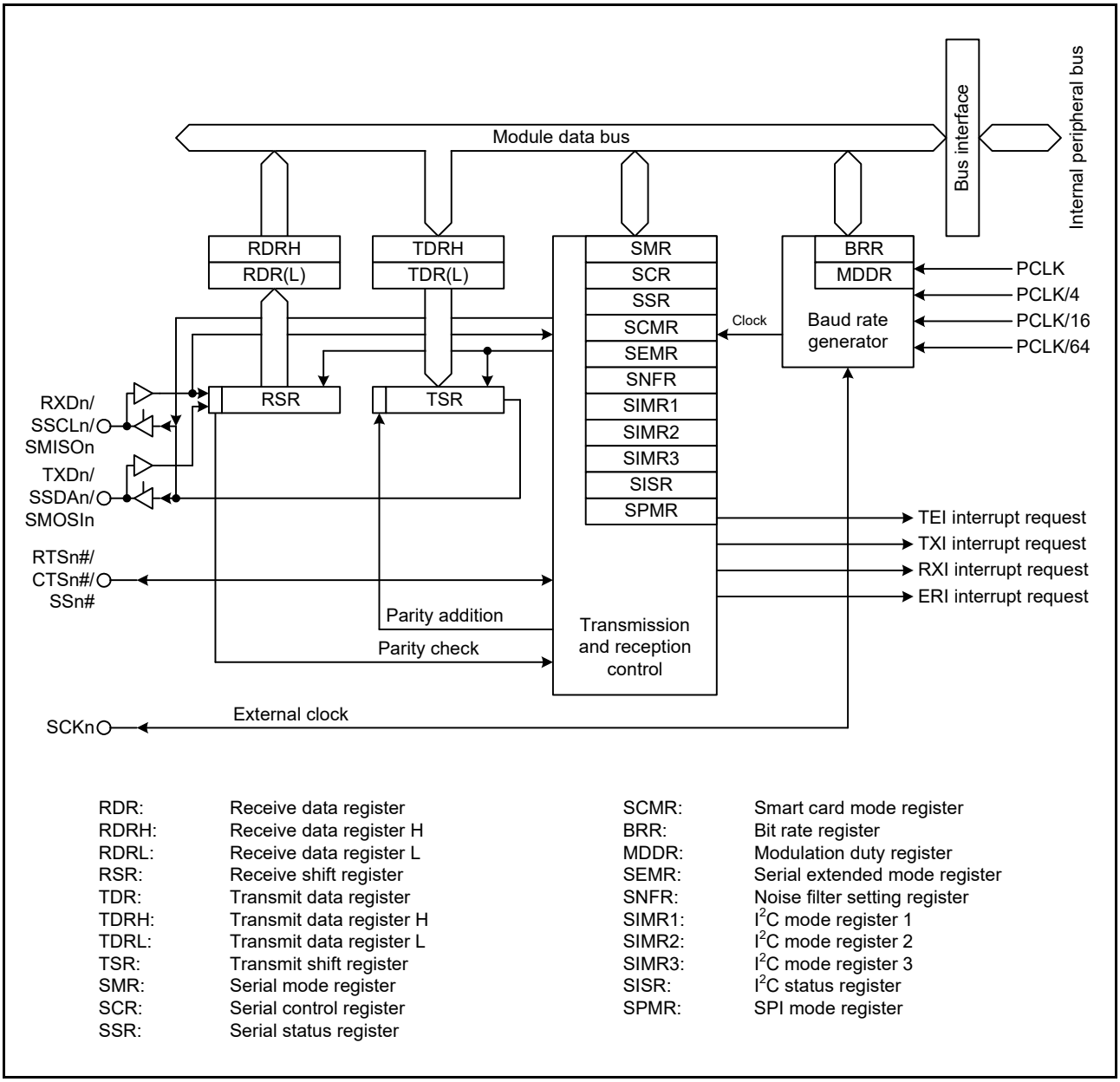


Figure 33.1 Block Diagram of SCIg (SCI1 and SCI8)

Downloaded from Arrow.com.

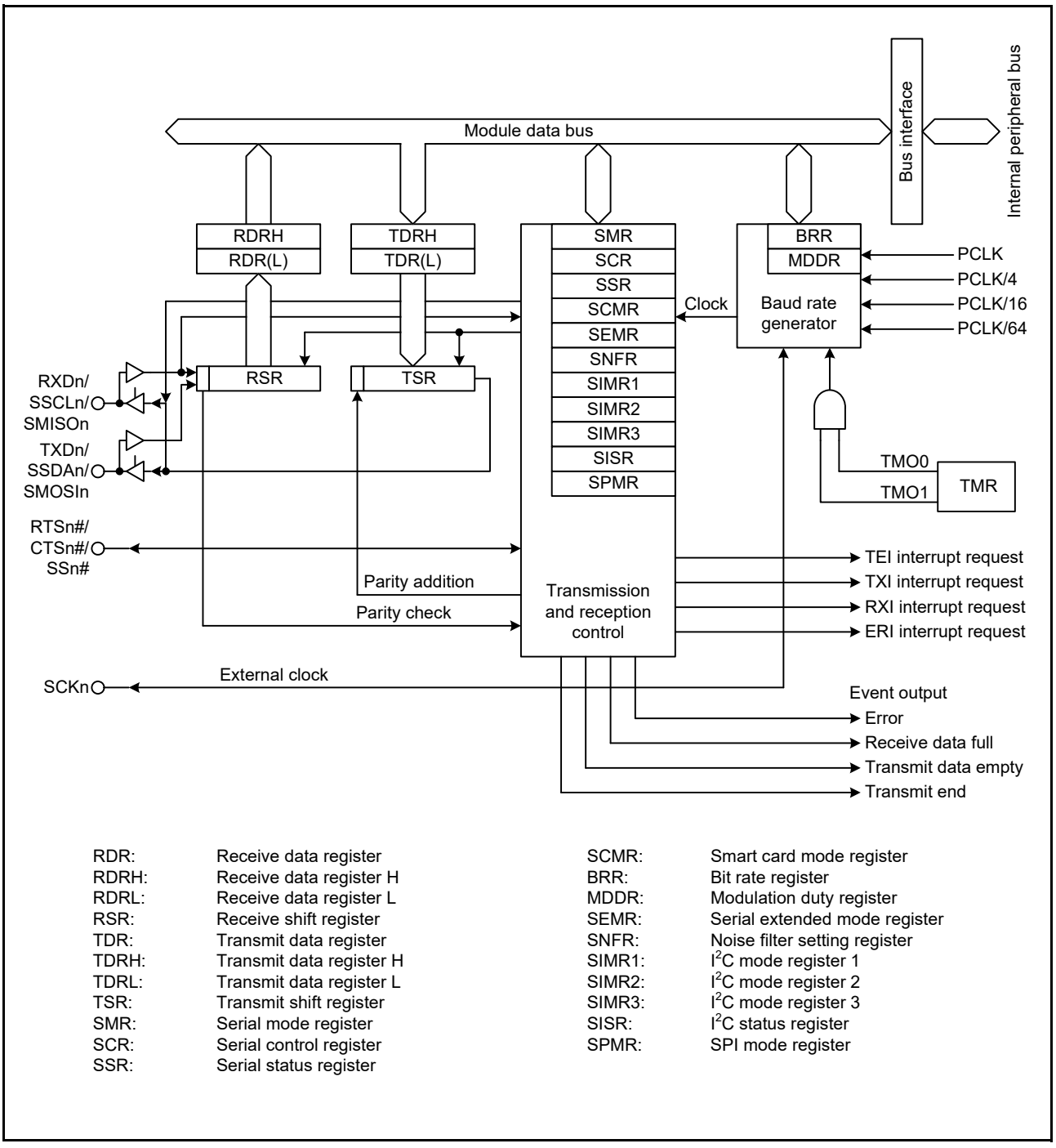


Figure 33.2 Block Diagram of SCIg (SCI5)

Downloaded from Arrow.com.

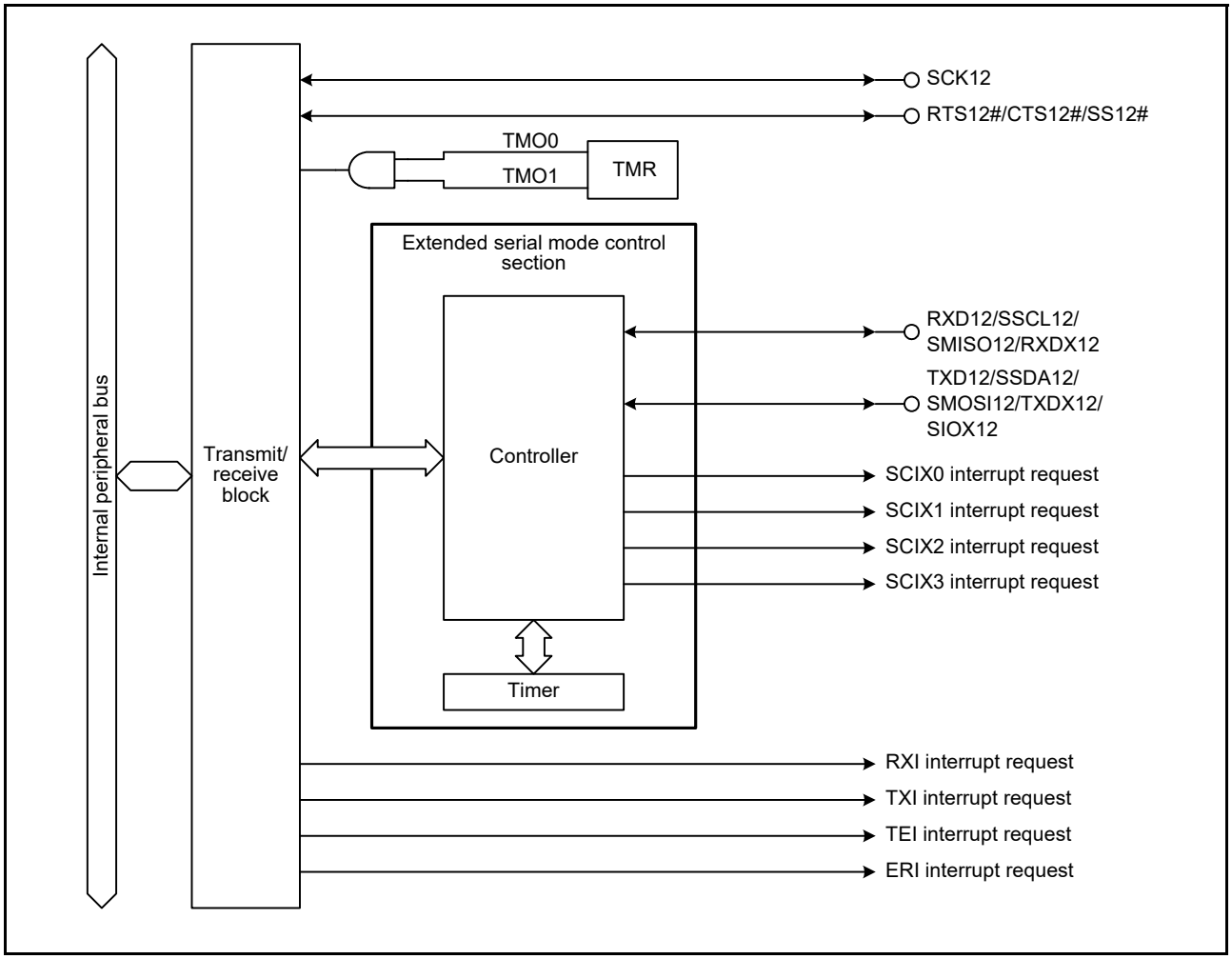


Figure 33.3 Block Diagram of SCIh (SCI12)

Downloaded from Arrow.com.

Table 33.4 to Table 33.7 list the pin configuration of the SCIs for the individual modes.

Table 33.4 SCI Pin Configuration in Asynchronous Mode and Clock Synchronous Mode

Channel	Pin Name	I/O	Function
SCI1	SCK1	I/O	SCI1 clock input/output
	RXD1	Input	SCI1 receive data input
	TXD1	Output	SCI1 transmit data output
	CTS1#/RTS1#	I/O	SCI1 transfer start control input/output
SCI5	SCK5	I/O	SCI5 clock input/output
	RXD5	Input	SCI5 receive data input
	TXD5	Output	SCI5 transmit data output
	CTS5#/RTS5#	I/O	SCI5 transfer start control input/output
SCI8	SCK8	I/O	SCI8 clock input/output
	RXD8	Input	SCI8 receive data input
	TXD8	Output	SCI8 transmit data output
	CTS8#/RTS8#	I/O	SCI8 transfer start control input/output
SCI12	SCK12	I/O	SCI12 clock input/output
	RXD12	Input	SCI12 receive data input
	TXD12	Output	SCI12 transmit data output
	CTS12#/RTS12#	I/O	SCI12 transfer start control input/output

Table 33.5 SCI Pin Configuration in Simple I^2C Mode

Channel	Pin Name	I/O	Function
SCI1	SSCL1	I/O	SCI1 I^2C clock input/output
	SSDA1	I/O	SCI1 I^2C data input/output
SCI5	SSCL5	I/O	SCI5 I^2C clock input/output
	SSDA5	I/O	SCI5 I^2C data input/output
SCI8	SSCL8	I/O	SCI8 I^2C clock input/output
	SSDA8	I/O	SCI8 I^2C data input/output
SCI12	SSCL12	I/O	SCI12 I^2C clock input/output
	SSDA12	I/O	SCI12 I^2C data input/output

Table 33.6 SCI Pin Configuration in Simple SPI Mode (1/2)

Channel	Pin Name	I/O	Function
SCI1	SCK1	I/O	SCI1 clock input/output
	SMISO1	I/O	SCI1 slave transmit data input/output
	SMOSI1	I/O	SCI1 master transmit data input/output
	SS1#	Input	SCI1 chip select input
SCI5	SCK5	I/O	SCI5 clock input/output
	SMISO5	I/O	SCI5 slave transmit data input/output
	SMOSI5	I/O	SCI5 master transmit data input/output
	SS5#	Input	SCI5 chip select input
SCI8	SCK8	I/O	SCI8 clock input/output
	SMISO8	I/O	SCI8 slave transmit data input/output
	SMOSI8	I/O	SCI8 master transmit data input/output
	SS8#	Input	SCI8 chip select input

Downloaded from Arrow.com.

Table 33.6 SCI Pin Configuration in Simple SPI Mode (2/2)

Channel	Pin Name	I/O	Function
SCI12	SCK12	I/O	SCI12 clock input/output
	SMISO12	I/O	SCI12 slave transmit data input/output
	SMOSI12	I/O	SCI12 master transmit data input/output
	SS12#	Input	SCI12 chip select input

Table 33.7 SCI Pin Configuration in Extended Serial Mode

Channel	Pin Name	I/O	Function
SCI12	RXDX12	Input	SCI12 receive data input
	TXDX12	Output	SCI12 transmit data output
	SIOX12	I/O	SCI12 transfer data input/output

Downloaded from Arrow.com.

33.2 Register Descriptions

33.2.1 Receive Shift Register (RSR)

The RSR register is a shift register which is used to receive serial data input from the RXDn pin and converts it into parallel data.
When one frame of data has been received, it is automatically transferred to the RDR register.
The RSR register cannot be directly accessed by the CPU.

33.2.2 Receive Data Register (RDR)

Address(es): SCI1.RDR 0008 A025h, SCI5.RDR 0008 A0A5h, SCI8.RDR 0008 A105h, SCI12.RDR 0008 B305h

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0

The RDR register is an 8-bit register that stores receive data.
When one frame of serial data has been received, the received serial data is transferred from the RSR register to the RDR register. Then the RSR register can receive the next data.
Since the RSR and RDR registers function as a double buffer in this way, continuous receive operations can be performed.
Read the RDR register only once after a receive data full interrupt (RXI) has occurred. Note that if next one frame of data is received before reading receive data from the RDR register, an overrun error occurs.
The RDR register cannot be written to by the CPU.

Downloaded from Arrow.com.

33.2.3 Receive Data Register H, L, HL (RDRH, RDRL, RDRHL)

- Receive Data Register H (RDRH)

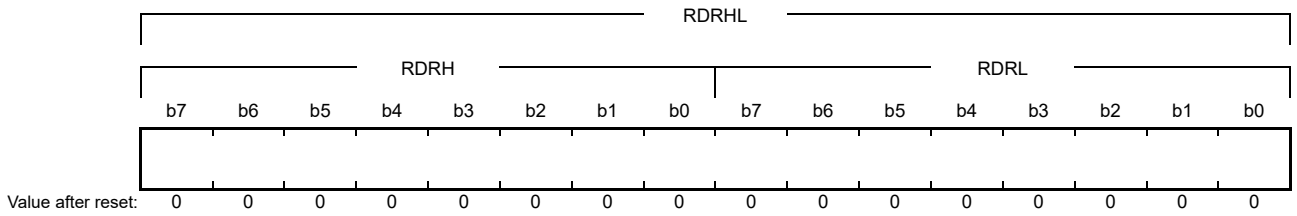
Address(es): SCI1.RDRH 0008 A030h, SCI5.RDRH 0008 A0B0h, SCI8.RDRH 0008 A110h, SCI12.RDRH 0008 B310h

- Receive Data Register L (RDRL)

Address(es): SCI1.RDRL 0008 A031h, SCI5.RDRL 0008 A0B1h, SCI8.RDRL 0008 A111h, SCI12.RDRL 0008 B311h

- Receive Data Register HL (RDRHL)

Address(es): SCI1.RDRHL 0008 A030h, SCI5.RDRHL 0008 A0B0h, SCI8.RDRHL 0008 A110h, SCI12.RDRHL 0008 B310h



The RDRH and RDRL registers are 8-bit registers that store receive data. Use these registers when asynchronous mode and 9-bit data length are selected.

The RDRL register is the shadow register of the RDR register; i.e. access to the RDRL register is equivalent to access to the RDR register.

After one frame of data is received, the received data is transferred from the RSR register to these registers, thus allowing the RSR register to receive the next data.

The RSR, RDRH and RDRL registers have a double-buffered construction to enable continuous reception.

Read the RDRH and RDRL registers should be performed only once in the order from the RDRH register to the RDRL register when a receive data full interrupt (RXI) request is issued. Note that an overrun error occurs when the next frame of data is received before the received data has been read from the RDRL register.

The CPU cannot write to the RDRH and RDRL registers. Bits 0 to 7 in the RDRH register are fixed to 0. These bits are read as 0.

The RDRHL register can be accessed in 16-bit units.

Downloaded from Arrow.com.

33.2.4 Transmit Data Register (TDR)

Address(es): SCI1.TDR 0008 A023h, SCI5.TDR 0008 A0A3h, SCI8.TDR 0008 A103h, SCI12.TDR 0008 B303h

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1

The TDR register is an 8-bit register that stores transmit data.
When the SCI detects that the TSR register is empty, it transfers the transmit data written in the TDR register to the TSR register and starts transmission.
The double-buffered structures of the TDR register and the TSR register enable continuous serial transmission. If the next transmit data has already been written to the TDR register when one frame of data is transmitted, the SCI transfers the written data to the TSR register to continue transmission.
The CPU is able to read from or write to the TDR register at any time. Only write transmit data to the TDR register once after each instance of the transmit data empty interrupt (TXI).

Downloaded from Arrow.com.

33.2.5 Transmit Data Register H, L, HL (TDRH, TDRL, TDRHL)

- Transmit Data Register H (TDRH)

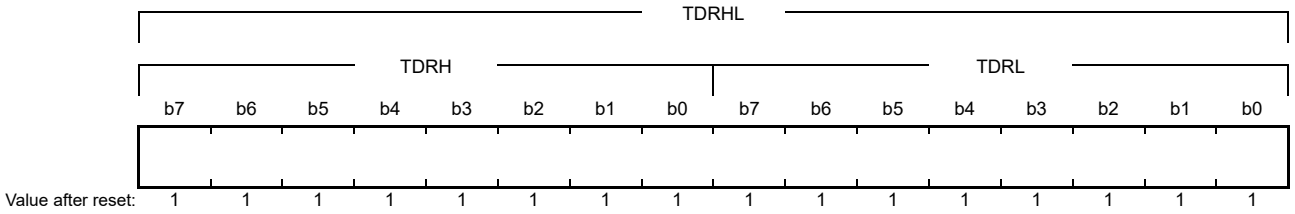
Address(es): SCI1.TDRH 0008 A02Eh, SCI5.TDRH 0008 A0AEh, SCI8.TDRH 0008 A10Eh, SCI12.TDRH 0008 B30Eh

- Transmit Data Register L (TDRL)

Address(es): SCI1.TDRL 0008 A02Fh, SCI5.TDRL 0008 A0AFh, SCI8.TDRL 0008 A10Fh, SCI12.TDRL 0008 B30Fh

- Transmit Data Register HL (TDRHL)

Address(es): SCI1.TDRHL 0008 A02Eh, SCI5.TDRHL 0008 A0AEh, SCI8.TDRHL 0008 A10Eh, SCI12.TDRHL 0008 B30Eh



The TDRH and TDRL registers are 8-bit registers that store transmit data. Use these registers when asynchronous mode and 9-bit data length are selected.
The TDRL register is the shadow register of the TDR register; i.e. access to the TDRL register is equivalent to access to the TDR register.
When empty space is detected in the TSR register, the transmit data stored in the TDRH and TDRL registers is transferred to the TSR register; i.e., transmitting is started.
The TSR, TDRH and TDRL registers have a double-buffered construction to realize continuous reception. When the next data to be transmitted is stored in the TDRL register after one frame of data has been transmitted, the transmitting operation is continued by transfer to the TSR register.
The CPU can read and write to the TDRH and TDRL registers. Bits 0 to 7 in the RDRH register are fixed to 1. These bits are read as 1. The write value should be 1.
Writing transmit data to the TDRH and TDRL registers should be performed only once in the order from the TDRH register to the TDRL register when a transmit data empty interrupt (TXI) request is issued.
The TDRHL register can be accessed in 16-bit units.

33.2.6 Transmit Shift Register (TSR)

The TSR register is a shift register that transmits serial data.
To perform serial data transmission, the SCI first automatically transfers transmit data from the TDR register to the TSR register, and then sends the data to the TXDn pin.
The TSR register cannot be directly accessed by the CPU.

Downloaded from Arrow.com.

33.2.7 Serial Mode Register (SMR)

Some bits in the SMR register have different functions in smart card interface mode and non-smart card interface mode.

(1) Non-Smart Card Interface Mode (SCMR.SMIF = 0)

Address(es): SCI1.SMR 0008 A020h, SCI5.SMR 0008 A0A0h, SCI8.SMR 0008 A100h, SCI12.SMR 0008 B300h

	b7	b6	b5	b4	b3	b2	b1	b0
	CM	CHR	PE	PM	STOP	MP	CKS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	CKS[1:0]	Clock Select	b1 b0 0 0: PCLK (n = 0)*1 0 1: PCLK/4 (n = 1)*1 1 0: PCLK/16 (n = 2)*1 1 1: PCLK/64 (n = 3)*1	R/W*4
b2	MP	Multi-Processor Mode	(Valid only in asynchronous mode) 0: Multi-processor communications function is disabled 1: Multi-processor communications function is enabled	R/W*4
b3	STOP	Stop Bit Length	(Valid only in asynchronous mode) 0: 1 stop bit 1: 2 stop bits	R/W*4
b4	PM	Parity Mode	(Valid only when the PE bit is 1) 0: Selects even parity 1: Selects odd parity	R/W*4
b5	PE	Parity Enable	(Valid only in asynchronous mode) • When transmitting 0: Parity bit addition is not performed 1: The parity bit is added • When receiving 0: Parity bit checking is not performed 1: The parity bit is checked	R/W*4
b6	CHR	Character Length	(Valid only in asynchronous mode*2) Selects in combination with the SCMR.CHR1 bit. CHR1 CHR 0 0: Transmit/receive in 9-bit data length 0 1: Transmit/receive in 9-bit data length 1 0: Transmit/receive in 8-bit data length (initial value) 1 1: Transmit/receive in 7-bit data length*3	R/W*4
b7	CM	Communications Mode	0: Asynchronous mode or simple I2C mode 1: Clock synchronous mode or simple SPI mode	R/W*4

Note 1. n is the decimal notation of the value of n in the BRR register (refer to section 33.2.11, Bit Rate Register (BRR)).
Note 2. In other than asynchronous mode, this bit setting is invalid and a fixed data length of 8 bits is used.
Note 3. LSB first is fixed and the MSB (bit 7) in the TDR register is not transmitted in transmission.
Note 4. Writable only when the SCR.TE bit is 0 and the SCR.RE bit is 0 (both serial transmission and reception are disabled).

CKS[1:0] Bits (Clock Select)

These bits select the clock source for the on-chip baud rate generator.

For the relation between the settings of these bits and the baud rate, refer to section 33.2.11, Bit Rate Register (BRR).

MP Bit (Multi-Processor Mode)

Disables/enables the multi-processor communications function. The settings of the PE bit and PM bit are invalid in multi-processor mode.

Downloaded from Arrow.com.

STOP Bit (Stop Bit Length)

Selects the stop bit length in transmission.
In reception, only the first stop bit is checked regardless of this bit setting. If the second stop bit is 0, it is treated as the start bit of the next transmit frame.

PM Bit (Parity Mode)

Selects the parity mode (even or odd) for transmission and reception.
The setting of the PM bit is invalid in multi-processor mode.

PE Bit (Parity Enable)

When this bit is set to 1, the parity bit is added to transmit data, and the parity bit is checked in reception.
Irrespective of the setting of the PE bit, the parity bit is not added or checked in multi-processor format.

CHR Bit (Character Length)

Selects the data length for transmission and reception.
Selects in combination with the SCMR.CHR1 bit.
In other than asynchronous mode, a fixed data length of 8 bits is used.

Downloaded from Arrow.com.

(2) Smart Card Interface Mode (SCMR.SMIF = 1)

Address(es): SMCI1.SMR 0008 A020h, SMCI5.SMR 0008 A0A0h, SMCI8.SMR 0008 A100h, SMCI12.SMR 0008 B300h

	b7	b6	b5	b4	b3	b2	b1	b0
	GM	BLK	PE	PM	BCP[1:0]		CKS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	CKS[1:0]	Clock Select	b1 b0 0 0: PCLK (n = 0)*1 0 1: PCLK/4 (n = 1)*1 1 0: PCLK/16 (n = 2)*1 1 1: PCLK/64 (n = 3)*1	R/W*2
b3, b2	BCP[1:0]	Base Clock Pulse	Selects the number of base clock cycles in combination with the SCMR.BCP2 bit. Table 33.8 lists the combinations of the SCMR.BCP2 bit and SMR.BCP[1:0] bits.	R/W*2
b4	PM	Parity Mode	(Valid only when the PE bit is 1) 0: Selects even parity 1: Selects odd parity	R/W*2
b5	PE	Parity Enable	When this bit is set to 1, a parity bit is added to transmit data, and the parity of received data is checked. Set this bit to 1 in smart card interface mode.	R/W*2
b6	BLK	Block Transfer Mode	0: Non-block transfer mode operation 1: Block transfer mode operation	R/W*2
b7	GM	GSM Mode	0: Non-GSM mode operation 1: GSM mode operation	R/W*2

Note 1. n is the decimal notation of the value of n in the BRR register (refer to section 33.2.11, Bit Rate Register (BRR)).
Note 2. Writable only when the SCR.TE bit is 0 and the SCR.RE bit is 0 (both serial transmission and reception are disabled).

CKS[1:0] Bits (Clock Select)

These bits select the clock source for the on-chip baud rate generator.

For the relationship between the settings of these bits and the baud rate, refer to section 33.2.11, Bit Rate Register (BRR).

BCP[1:0] Bits (Base Clock Pulse)

These bits select the number of base clock cycles in a 1-bit data transfer time in smart card interface mode.

Set these bits in combination with the SCMR.BCP2 bit.

For details, refer to section 33.6.4, Receive Data Sampling Timing and Reception Margin.

Table 33.8 Combinations of the SCMR.BCP2 Bit and SMR.BCP[1:0] Bits

SCMR.BCP2 Bit	SMR.BCP[1:0] Bits		Number of Base Clock Cycles for 1-Bit Transfer Period
0	0	0	93 clock cycles (S = 93)*1
0	0	1	128 clock cycles (S = 128)*1
0	1	0	186 clock cycles (S = 186)*1
0	1	1	512 clock cycles (S = 512)*1
1	0	0	32 clock cycles (S = 32)*1 (Initial Value)
1	0	1	64 clock cycles (S = 64)*1
1	1	0	372 clock cycles (S = 372)*1
1	1	1	256 clock cycles (S = 256)*1

Note 1. S is the value of S in the BRR register (refer to section 33.2.11, Bit Rate Register (BRR)).

Downloaded from Arrow.com.

PM Bit (Parity Mode)

Selects the parity mode for transmission and reception (even or odd).
For details on the usage of this bit in smart card interface mode, refer to section 33.6.2, Data Format (Except in Block Transfer Mode).

PE Bit (Parity Enable)

Set the PE bit to 1.
The parity bit is added to transmit data before transmission, and the parity bit is checked in reception.

BLK Bit (Block Transfer Mode)

Setting this bit to 1 allows block transfer mode operation.
For details, refer to section 33.6.3, Block Transfer Mode.

GM Bit (GSM Mode)

Setting this bit to 1 allows GSM mode operation.
In GSM mode, the SSR.TEND flag set timing is put forward to 11.0 etu (elementary time unit = 1-bit transfer time) from the start and the clock output control function is appended. For details, refer to section 33.6.6, Serial Data Transmission (Except in Block Transfer Mode) and section 33.6.8, Clock Output Control.

Downloaded from Arrow.com.

33.2.8 Serial Control Register (SCR)

Some bits in the SCR register have different functions in smart card interface mode and non-smart card interface mode.

(1) Non-Smart Card Interface Mode (SCMR.SMIF = 0)

Address(es): SCI1.SCR 0008 A022h, SCI5.SCR 0008 A0A2h, SCI8.SCR 0008 A102h, SCI12.SCR 0008 B302h

	b7	b6	b5	b4	b3	b2	b1	b0
	TIE	RIE	TE	RE	MPIE	TEIE	CKE[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	CKE[1:0]	Clock Enable	(Asynchronous mode) b1 b0 0 0: On-chip baud rate generator The SCKn pin becomes high-impedance. 0 1: On-chip baud rate generator The clock with the same frequency as the bit rate is output from the SCKn pin. 1 x: External clock or TMR clock*2 • The clock with a frequency 16 times the bit rate should be input from the SCKn pin. Input a clock signal with a frequency eight times the bit rate when the SEMR.ABCS bit is 1. • The SCKn pin becomes high-impedance when the TMR clock*2 is used. (Clock synchronous mode) b1 b0 0 x: Internal clock The SCKn pin functions as the clock output pin. 1 x: External clock The SCKn pin functions as the clock input pin.	R/W*1
b2	TEIE	Transmit End Interrupt Enable	0: A TEI interrupt request is disabled 1: A TEI interrupt request is enabled	R/W
b3	MPIE	Multi-Processor Interrupt Enable	(Valid in asynchronous mode when the SMR.MP bit is 1) 0: Normal reception 1: When the data with the multi-processor bit set to 0 is received, the data is not read, and setting the status flags RDRF, ORER, and FER in the SSR register to 1 is disabled. When the data with the multi-processor bit set to 1 is received, the MPIE bit is automatically cleared to 0, and normal reception is resumed.	R/W
b4	RE	Receive Enable	0: Serial reception is disabled 1: Serial reception is enabled	R/W*3
b5	TE	Transmit Enable	0: Serial transmission is disabled 1: Serial transmission is enabled	R/W*3
b6	RIE	Receive Interrupt Enable	0: RXI and ERI interrupt requests are disabled 1: RXI and ERI interrupt requests are enabled	R/W
b7	TIE	Transmit Interrupt Enable	0: A TXI interrupt request is disabled 1: A TXI interrupt request is enabled	R/W

x: Don't care

Note 1. Writable only when TE = 0 and RE = 0.

Note 2. TMR clock is selectable for SCI5 and SCI12.

Note 3. 1 can be written only when TE = 0 and RE = 0, while the SMR.CM bit is 1. After setting TE or RE to 1, only 0 can be written to TE and RE. While the SMR.CM bit is 0 and the SIMR1.IICM bit is 0, writing is enabled under any condition.

Downloaded from Arrow.com.

CKE[1:0] Bits (Clock Enable)

These bits select the clock source and SCKn pin function.
The combination of the settings of these bits and of the SEMR.ACS0 bit sets the internal TMR clock.

TEIE Bit (Transmit End Interrupt Enable)

Enables or disables a TEI interrupt request.
A TEI interrupt request is disabled by setting the TEIE bit to 0.
In simple I^2C mode, the TEI is allocated to the interrupt on completion of issuing a start, restart, or stop condition (STI). In this case, the TEIE bit can be used to enable or disable the STI.

MPIE Bit (Multi-Processor Interrupt Enable)

When this bit is set to 1 and the data with the multi-processor bit set to 0 is received, the data is not read and setting the status flags RDRF, ORER, and FER in the SSR register to 1 is disabled. When the data with the multi-processor bit set to 1 is received, the MPIE bit is automatically cleared to 0, and normal reception is resumed. For details, refer to section 33.4, Multi-Processor Communications Function.
When the data with the multi-processor bit set to 0 is received, the receive data is not transferred from the RSR register to the RDR register, a receive error is not detected, and setting the flags RDRF, ORER, and FER to 1 is disabled.
When the data with the multi-processor bit set to 1 is received, the MPB bit is set to 1, the MPIE bit is automatically cleared to 0, the RXI and ERI interrupt requests are enabled (if the SCR.RIE bit is set to 1), and setting the flags RDRF, ORER, and FER to 1 is enabled.
Set the MPIE bit to 0 if multi-processor communications function is not to be used.

RE Bit (Receive Enable)

Enables or disables serial reception.
When this bit is set to 1, serial reception is started by detecting the start bit in asynchronous mode or the synchronous clock input in clock synchronous mode. Note that the SMR register should be set prior to setting the RE bit to 1 in order to designate the reception format.
Even if reception is halted by setting the RE bit to 0, the ORER, FER, PER, and RDRF flags in the SSR register are not affected and the previous value is retained.

TE Bit (Transmit Enable)

Enables or disables serial transmission.
When this bit is set to 1, serial transmission is started by writing transmit data to the TDR register. Note that the SMR register should be set prior to setting the TE bit to 1 in order to designate the transmission format.

RIE Bit (Receive Interrupt Enable)

Enables or disables RXI and ERI interrupt requests.
An RXI interrupt request is disabled by setting the RIE bit to 0.
An ERI interrupt request can be canceled by reading 1 from the ORER, FER, or PER flag in the SSR register and then setting the flag to 0, or setting the RIE bit to 0.

TIE Bit (Transmit Interrupt Enable)

Enables or disables TXI interrupt request.
A TXI interrupt request is disabled by setting the TIE bit to 0.

Downloaded from Arrow.com.

(2) Smart Card Interface Mode (SCMR.SMIF = 1)

Address(es): SMCI1.SCR 0008 A022h, SMCI5.SCR 0008 A0A2h, SMCI8.SCR 0008 A102h, SMCI12.SCR 0008 B302h

	b7	b6	b5	b4	b3	b2	b1	b0
	TIE	RIE	TE	RE	MPIE	TEIE	CKE[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	CKE[1:0]	Clock Enable	• When SMR.GM = 0 b1 b0 0 0: Output disabled The SCKn pin becomes high-impedance. 0 1: Clock output 1 x: Setting prohibited • When SMR.GM = 1 b1 b0 0 0: Output fixed low x 1: Clock output 1 0: Output fixed high	R/W*1
b2	TEIE	Transmit End Interrupt Enable	This bit should be 0 in smart card interface mode.	R/W
b3	MPIE	Multi-Processor Interrupt Enable	This bit should be 0 in smart card interface mode.	R/W
b4	RE	Receive Enable	0: Serial reception is disabled 1: Serial reception is enabled	R/W*2
b5	TE	Transmit Enable	0: Serial transmission is disabled 1: Serial transmission is enabled	R/W*2
b6	RIE	Receive Interrupt Enable	0: RXI and ERI interrupt requests are disabled 1: RXI and ERI interrupt requests are enabled	R/W
b7	TIE	Transmit Interrupt Enable	0: A TXI interrupt request is disabled 1: A TXI interrupt request is enabled	R/W

x: Don't care

Note 1. Writable only when TE = 0 and RE = 0.

Note 2. 1 can be written only when TE = 0 and RE = 0. After setting TE or RE to 1, only 0 can be written in TE and RE.

For details on interrupt requests, refer to section 33.12, Interrupt Sources.

CKE[1:0] Bits (Clock Enable)

These bits control the clock output from the SCKn pin.

In GSM mode, clock output can be dynamically switched. For details, refer to section 33.6.8, Clock Output Control.

TEIE Bit (Transmit End Interrupt Enable)

This bit should be 0 in smart card interface mode.

MPIE Bit (Multi-Processor Interrupt Enable)

This bit should be 0 in smart card interface mode.

Downloaded from Arrow.com.

RE Bit (Receive Enable)

Enables or disables serial reception.

When this bit is set to 1, serial reception is started by detecting the start bit. Note that the SMR register should be set prior to setting the RE bit to 1 in order to designate the reception format.

Even if reception is halted by setting the RE bit to 0, the ORER, FER, and PER flags in the SSR register are not affected and the previous value is retained.

TE Bit (Transmit Enable)

Enables or disables serial transmission.

When this bit is set to 1, serial transmission is started by writing transmit data to the TDR register. Note that the SMR register should be set prior to setting the TE bit to 1 in order to designate the transmission format.

RIE Bit (Receive Interrupt Enable)

Enables or disables RXI and ERI interrupt requests.

An RXI interrupt request is disabled by setting the RIE bit to 0.

An ERI interrupt request can be canceled by reading 1 from the ORER, FER, or PER flag in the SSR register and then setting the flag to 0, or setting the RIE bit to 0.

TIE Bit (Transmit Interrupt Enable)

Enables or disables TXI interrupt request.

A TXI interrupt request is disabled by setting the TIE bit to 0.

Downloaded from Arrow.com.

33.2.9 Serial Status Register (SSR)

Some bits in the SSR register have different functions in smart card interface mode and non-smart card interface mode.

(1) Non-Smart Card Interface Mode (SCMR.SMIF = 0)

Address(es): SCI1.SSR 0008 A024h, SCI5.SSR 0008 A0A4h, SCI8.SSR 0008 A104h, SCI12.SSR 0008 B304h

	b7	b6	b5	b4	b3	b2	b1	b0
	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT
Value after reset:	1	0	0	0	0	1	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MPBT	Multi-Processor Bit Transfer	Sets the multi-processor bit for adding to the transmission frame 0: Data transmission cycles 1: ID transmission cycles	R/W
b1	MPB	Multi-Processor	Value of the multi-processor bit in the reception frame 0: Data transmission cycles 1: ID transmission cycles	R
b2	TEND	Transmit End Flag	0: A character is being transmitted. 1: Character transfer has been completed.	R
b3	PER	Parity Error Flag	0: No parity error occurred 1: A parity error has occurred	R/(W) *1
b4	FER	Framing Error Flag	0: No framing error occurred 1: A framing error has occurred	R/(W) *1
b5	ORER	Overrun Error Flag	0: No overrun error occurred 1: An overrun error has occurred	R/(W) *1
b6	RDRF	Receive Data Full Flag	0: No valid data is held in the RDR register 1: Received data is held in the RDR register	R/(W) *2
b7	TDRE	Transmit Data Empty Flag	0: Data to be transmitted is held in the TDR register 1: No data is held in the TDR register	R/(W) *2

Note 1. Only 0 can be written to this bit, to clear the flag. To clear this flag, confirm that the flag is 1 and then set it to 0.
Note 2. Write 1 when writing is necessary.

MPB Bit (Multi-Processor)

Holds the value of the multi-processor bit in the reception frame. This bit does not change when the SCR.RE bit is 0.

TEND Flag (Transmit End Flag)

Indicates completion of transmission.

[Setting conditions]

- When the SCR.TE bit is set to 0 (serial transmission is disabled)
  When the SCR.TE bit is changed from 0 to 1, the TEND flag is not affected and retains the value 1.
- When the TDR register is not updated at the time of transmission of the tail-end bit of a character being transmitted

[Clearing condition]

- When transmit data are written to the TDR register while the SCR.TE bit is 1
  When setting the TEND flag to 0 to complete the interrupt handling, refer to section 15.4.1.2, Operation of Status Flags for Level-Detected Interrupts.

Downloaded from Arrow.com.

PER Flag (Parity Error Flag)

Indicates that a parity error has occurred during reception in asynchronous mode and the reception ends abnormally.

[Setting condition]

- When a parity error is detected during reception
  Although receive data when the parity error occurs is transferred to the RDR register, no RXI interrupt request occurs. Note that when the PER flag is being set to 1, the subsequent receive data is not transferred to the RDR register.

[Clearing condition]

- When 0 is written to the PER flag after reading PER = 1
  When setting the PER flag to 0 to complete the interrupt handling, refer to section 15.4.1.2, Operation of Status Flags for Level-Detected Interrupts.
  Even when the SCR.RE bit is set to 0 (serial reception is disabled), the PER flag is not affected and retains its previous value.

FER Flag (Framing Error Flag)

Indicates that a framing error has occurred during reception in asynchronous mode and the reception ends abnormally.

[Setting condition]

- When the stop bit is 0
  In 2-stop-bit mode, only the first stop bit is checked whether it is 1 but the second stop bit is not checked. Note that although receive data when the framing error occurs is transferred to the RDR register, no RXI interrupt request occurs. In addition, when the FER flag is being set to 1, the subsequent receive data is not transferred to the RDR register.

[Clearing condition]

- When 0 is written to the FER flag after reading FER = 1
  When setting the FER flag to 0 to complete the interrupt handling, refer to section 15.4.1.2, Operation of Status Flags for Level-Detected Interrupts.
  Even when the SCR.RE bit is set to 0, the FER flag is not affected and retains its previous value.

ORER Flag (Overrun Error Flag)

Indicates that an overrun error has occurred during reception and the reception ends abnormally.

[Setting condition]

- When the next data is received before receive data is read from the RDR register
  In the RDR register, receive data prior to an overrun error occurrence is retained, but data received after the overrun error occurrence is lost. When the ORER flag is set to 1, subsequent serial reception cannot be performed. Note that, in clock synchronous mode, serial transmission also cannot continue.

[Clearing condition]

- When 0 is written to the ORER flag after reading ORER = 1
  When setting the ORER flag to 0 to complete the interrupt handling, refer to section 15.4.1.2, Operation of Status Flags for Level-Detected Interrupts.
  Even when the SCR.RE bit is set to 0, the ORER flag is not affected and retains its previous value.

RDRF Flag (Receive Data Full Flag)

Indicates whether the RDR register has received data.

[Setting condition]

- When data has been received normally, and transferred from the RSR register to the RDR register

[Clearing condition]

- When data is read from the RDR register

Downloaded from Arrow.com.

TDRE Flag (Transmit Data Empty Flag)

Indicates whether the TDR register has data to be transmitted.

[Setting condition]

- When data is transferred from the TDR register to the TSR register

[Clearing condition]

- When data is written to the TDR register

(2) Smart Card Interface Mode (SCMR.SMIF = 1)

Address(es): SMCI1.SSR 0008 A024h, SMCI5.SSR 0008 A0A4h, SMCI8.SSR 0008 A104h, SMCI12.SSR 0008 B304h

	b7	b6	b5	b4	b3	b2	b1	b0
	TDRE	RDRF	ORER	ERS	PER	TEND	MPB	MPBT
Value after reset:	1	0	0	0	0	1	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MPBT	Multi-Processor Bit Transfer	This bit should be set to 0 in smart card interface mode.	R/W
b1	MPB	Multi-Processor	This bit is not used in smart card interface mode. It should be set to 0.	R
b2	TEND	Transmit End Flag	0: A character is being transmitted. 1: Character transfer has been completed.	R
b3	PER	Parity Error Flag	0: No parity error occurred 1: A parity error has occurred	R/(W) *1
b4	ERS	Error Signal Status Flag	0: Low error signal not responded 1: Low error signal responded	R/(W) *1
b5	ORER	Overrun Error Flag	0: No overrun error occurred 1: An overrun error has occurred	R/(W) *1
b6	RDRF	Receive Data Full Flag	0: No valid data is held in the RDR register 1: Received data is held in the RDR register	R/(W) *2
b7	TDRE	Transmit Data Empty Flag	0: Data to be transmitted is held in the TDR register 1: No data is held in the TDR register	R/(W) *2

Note 1. Only 0 can be written to this bit, to clear the flag. To clear this flag, confirm that the flag is 1 and then set it to 0.
Note 2. Write 1 when writing is necessary.

Downloaded from Arrow.com.

TEND Flag (Transmit End Flag)

With no error signal from the receiving side, this bit is set to 1 when further data for transfer is ready to be transferred to the TDR register.

[Setting conditions]

- When the SCR.TE bit is 0 (serial transmission is disabled)
  When the SCR.TE bit is changed from 0 to 1, the TEND flag is not affected and retains the value 1.
- When a specified period has elapsed after the latest transmission of 1 byte, the ERS flag is 0, and the TDR register is not updated
  The set timing is determined by register settings as listed below.
  When SMR.GM = 0 and SMR.BLK = 0, 12.5 etu after the start of transmission
  When SMR.GM = 0 and SMR.BLK = 1, 11.5 etu after the start of transmission
  When SMR.GM = 1 and SMR.BLK = 0, 11.0 etu after the start of transmission
  When SMR.GM = 1 and SMR.BLK = 1, 11.0 etu after the start of transmission

[Clearing condition]

- When transmit data are written to the TDR register while the SCR.TE bit is 1
  When setting the TEND flag to 0 to complete the interrupt handling, refer to section 15.4.1.2, Operation of Status Flags for Level-Detected Interrupts.

PER Flag (Parity Error Flag)

Indicates that a parity error has occurred during reception in asynchronous mode and the reception ends abnormally.

[Setting condition]

- When a parity error is detected during reception
  Although receive data when the parity error occurs is transferred to the RDR register, no RXI interrupt request occurs. Note that when the PER flag is being set to 1, the subsequent receive data is not transferred to the RDR register.

[Clearing condition]

- When 0 is written to the PER flag after reading PER = 1
  When setting the PER flag to 0 to complete the interrupt handling, refer to section 15.4.1.2, Operation of Status Flags for Level-Detected Interrupts.
  Even when the SCR.RE bit is set to 0 (serial reception is disabled), the PER flag is not affected and retains its previous value.

ERS Flag (Error Signal Status Flag)

[Setting condition]

- When a low error signal is sampled

[Clearing condition]

- When 0 is written to the ERS flag after reading ERS = 1
  When setting the ERS flag to 0 to complete the interrupt handling, refer to section 15.4.1.2, Operation of Status Flags for Level-Detected Interrupts.
  Even when the SCR.RE bit is set to 0, the ERS flag is not affected and retains its previous value.

Downloaded from Arrow.com.

ORER Flag (Overrun Error Flag)

Indicates that an overrun error has occurred during reception and the reception ends abnormally.

[Setting condition]

- When the next data is received before receive data is read from the RDR register
  In the RDR register, the receive data prior to an overrun error occurrence is retained, but data received following the overrun error occurrence is lost. When the ORER flag is set to 1, subsequent serial reception cannot be performed.

[Clearing condition]

- When 0 is written to the ORER flag after reading ORER = 1
  When setting the ORER flag to 0 to complete the interrupt handling, refer to section 15.4.1.2, Operation of Status Flags for Level-Detected Interrupts.
  Even when the SCR.RE bit is set to 0, the ORER flag is not affected and retains its previous value.

RDRF Flag (Receive Data Full Flag)

Indicates whether the RDR register has received data.

[Setting condition]

- When data has been received normally, and transferred from the RSR register to the RDR register

[Clearing condition]

- When data is read from the RDR register

TDRE Flag (Transmit Data Empty Flag)

Indicates whether the TDR register has data to be transmitted.

[Setting condition]

- When data is transferred from the TDR register to the TSR register

[Clearing condition]

- When data is written to the TDR register

Downloaded from Arrow.com.

33.2.10 Smart Card Mode Register (SCMR)

Address(es): SCI1.SCMR 0008 A026h, SCI5.SCMR 0008 A0A6h, SCI8.SCMR 0008 A106h, SCI12.SCMR 0008 B306h, SMCI1.SCMR 0008 A026h, SMCI5.SCMR 0008 A0A6h, SMCI8.SCMR 0008 A106h, SMCI12.SCMR 0008 B306h

	b7	b6	b5	b4	b3	b2	b1	b0
	BCP2	—	—	CHR1	SDIR	SINV	—	SMIF
Value after reset:	1	1	1	1	0	0	1	0

Bit	Symbol	Bit Name	Description	R/W
b0	SMIF	Smart Card Interface Mode Select	0: Non-smart card interface mode (Asynchronous mode, clock synchronous mode, simple SPI mode, or simple I2C mode) 1: Smart card interface mode	R/W*1
b1	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b2	SINV	Transmitted/Received Data Invert*2, *3	0: Data bits in the TDR register are transferred to the TSR register as they are. Data bits in the RSR register are transferred to the RDR register as they are. 1: Data bits in the TDR register are transferred to the TSR register with inverting. Data bits in the RSR register are transferred to the RDR register with inverting.	R/W*1
b3	SDIR	Transmitted/Received Data Transfer Direction*2, *4	0: Transfer with LSB first 1: Transfer with MSB first	R/W*1
b4	CHR1	Character Length 1*5	Selects in combination with the SMR.CHR bit. CHR1 CHR 0 0: Transmit/receive in 9-bit data length 0 1: Transmit/receive in 9-bit data length 1 0: Transmit/receive in 8-bit data length (initial value) 1 1: Transmit/receive in 7-bit data length*6	R/W*1
b6, b5	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b7	BCP2	Base Clock Pulse 2	Selects the number of base clock cycles in combination with the SMR.BCP[1:0] bits. Table 33.9 lists the combinations of the SCMR.BCP2 bit and SMR.BCP[1:0] bits.	R/W*1

Note 1. Writable only when the SCR.TE bit is 0 and the SCR.RE bit is 0 (both serial transmission and reception are disabled).
Note 2. This bit can be used in the smart card interface mode, asynchronous mode (multi-processor mode), clock synchronous mode, and simple SPI mode.
Note 3. Set this bit to 0 if operation is to be in simple I2C mode.
Note 4. Set this bit to 1 if operation is to be in simple I2C mode.
Note 5. This bit is only valid in asynchronous mode. The setting is invalid and a fixed data length of 8 bits is used in modes other than asynchronous mode.
Note 6. LSB first should be selected and the value of MSB (b7) in the TDR register cannot be transmitted.

SMIF Bit (Smart Card Interface Mode Select)

When this bit is set to 1, smart card interface mode is selected.

When this bit is set to 0, non-smart card interface mode, i.e., asynchronous mode (including multi-processor mode), clock synchronous mode, simple SPI mode, or simple I2C mode is selected.

SINV Bit (Transmitted/Received Data Invert)

This bit is used to invert the logic level of the data bits when the data is transferred between data register and shift register. This bit does not affect the logic level of the parity bit. To invert the parity bit, invert the SMR.PM bit.

CHR1 Bit (Character Length 1)

Selects the data length of transmit/receive data.

Selects in combination with the SMR.CHR bit.

A fixed data length of 8 bits is used in modes other than asynchronous mode.

Downloaded from Arrow.com.

BCP2 Bit (Base Clock Pulse 2)

Selects the number of base clock cycles in a 1-bit data transfer time in smart card interface mode. Set this bit in combination with the SMR.BCP[1:0] bits.

Table 33.9 Combinations of the SCMR.BCP2 Bit and SMR.BCP[1:0] Bits

SCMR.BCP2 Bit	SMR.BCP[1:0] Bits		Number of Base Clock Cycles for 1-Bit Transfer Period
0	0	0	93 clock cycles (S = 93)*1
0	0	1	128 clock cycles (S = 128)*1
0	1	0	186 clock cycles (S = 186)*1
0	1	1	512 clock cycles (S = 512)*1
1	0	0	32 clock cycles (S = 32)*1 (Initial Value)
1	0	1	64 clock cycles (S = 64)*1
1	1	0	372 clock cycles (S = 372)*1
1	1	1	256 clock cycles (S = 256)*1

Note 1. S is the value of S in the BRR register (refer to section 33.2.11, Bit Rate Register (BRR)).

Downloaded from Arrow.com.

33.2.11 Bit Rate Register (BRR)

Address(es): SCI1.BRR 0008 A021h, SCI5.BRR 0008 A0A1h, SCI8.BRR 0008 A101h, SCI12.BRR 0008 B301h

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1

The BRR register is an 8-bit register that adjusts the bit rate.
As each SCI channel has independent baud rate generator control, different bit rates can be set for each. Table 33.10 shows the relationship between the setting (N) in the BRR register and the bit rate (B) for normal asynchronous mode, multi-processor communication, clock synchronous mode, smart card interface mode, simple SPI mode, and simple I²C mode.
The BRR register is writable only when the TE and RE bits in the SCR register are 0.

Table 33.10 Relationship between N Setting in the BRR Register and Bit Rate B

Mode	SEMR Settings		BRR Setting	Error (%)
	BGDM bit	ABCS bit		
Asynchronous, multi-processor communication	0	0	$N = \frac{PCLK \times 10^6}{64 \times 2^{2n-1} \times B} - 1$	$Error = \left\{ \frac{PCLK \times 10^6}{B \times 64 \times 2^{2n-1} \times (N+1)} - 1 \right\} \times 100$
	0	1	$N = \frac{PCLK \times 10^6}{32 \times 2^{2n-1} \times B} - 1$	$Error = \left\{ \frac{PCLK \times 10^6}{B \times 32 \times 2^{2n-1} \times (N+1)} - 1 \right\} \times 100$
	1	0		
	1	1	$N = \frac{PCLK \times 10^6}{16 \times 2^{2n-1} \times B} - 1$	$Error = \left\{ \frac{PCLK \times 10^6}{B \times 16 \times 2^{2n-1} \times (N+1)} - 1 \right\} \times 100$
Clock synchronous, simple SPI			$N = \frac{PCLK \times 10^6}{8 \times 2^{2n-1} \times B} - 1$	
Smart card interface			$N = \frac{PCLK \times 10^6}{S \times 2^{2n+1} \times B} - 1$	$Error = \left\{ \frac{PCLK \times 10^6}{B \times S \times 2^{2n+1} \times (N+1)} - 1 \right\} \times 100$
Simple I²C*1			$N = \frac{PCLK \times 10^6}{64 \times 2^{2n-1} \times B} - 1$	

B: Bit rate (bps)
N: BRR setting for on-chip baud rate generator ($0 \le N \le 255$)
PCLK: Operating frequency (MHz)
n and S: Determined by the settings of the SMR and SCMR registers as listed in Table 33.12 and Table 33.13.
Note 1. Adjust the bit rate so that the widths at high and low level of the SCL output in simple I²C mode satisfy the I²C-bus standard.

Table 33.11 Calculating Widths at High and Low Level for SCL

Mode	SCL	Formula (Result in Seconds)
I²C	High period (minimum value)	$(N+1) \times 4 \times 2^{2n-1} \times 7 \times \frac{1}{PCLK \times 10^6}$
	Low period (minimum value)	$(N+1) \times 4 \times 2^{2n-1} \times 8 \times \frac{1}{PCLK \times 10^6}$

Downloaded from Arrow.com.

Table 33.12 Clock Source Settings

SMR.CKS[1:0] Bit Setting	Clock Source	n
0 0	PCLK	0
0 1	PCLK/4	1
1 0	PCLK/16	2
1 1	PCLK/64	3

Table 33.13 Base Clock Settings in Smart Card Interface Mode

SCMR.BCP2 Bit Setting	SMR.BCP[1:0] Bit Setting	Base Clock Cycles for 1-bit Period	S
0	0 0	93 clock cycles	93
0	0 1	128 clock cycles	128
0	1 0	186 clock cycles	186
0	1 1	512 clock cycles	512
1	0 0	32 clock cycles	32
1	0 1	64 clock cycles	64
1	1 0	372 clock cycles	372
1	1 1	256 clock cycles	256

Table 33.14 lists examples of N settings in the BRR register in normal asynchronous mode. Table 33.15 lists the maximum bit rate settable for each operating frequency. Examples of BRR (N) settings in clock synchronous mode and simple SPI mode are listed in Table 33.18. Examples of BRR (N) settings in smart card interface mode are listed in Table 33.20. Examples of BRR (N) settings in simple I^2C mode are listed in Table 33.22. In smart card interface mode, the number of base clock cycles S in a 1-bit data transfer time can be selected. For details, refer to section 33.6.4, Receive Data Sampling Timing and Reception Margin. Table 33.16 and Table 33.19 list the maximum bit rates with external clock input.

When either the SEMR.ABCS or BGDM bit is set to 1 in asynchronous mode, the bit rate becomes twice that listed in Table 33.14. When both of those bits are set to 1, the bit rate becomes four times the listed value.

Downloaded from Arrow.com.

Table 33.14 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode)

Bit Rate (bps)	Operating Frequency PCLK (MHz)														
	8			9.8304			10			12			12.288		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	141	0.03	2	174	–0.26	2	177	–0.25	2	212	0.03	2	217	0.08
150	2	103	0.16	2	127	0.00	2	129	0.16	2	155	0.16	2	159	0.00
300	1	207	0.16	1	255	0.00	2	64	0.16	2	77	0.16	2	79	0.00
600	1	103	0.16	1	127	0.00	1	129	0.16	1	155	0.16	1	159	0.00
1200	0	207	0.16	0	255	0.00	1	64	0.16	1	77	0.16	1	79	0.00
2400	0	103	0.16	0	127	0.00	0	129	0.16	0	155	0.16	0	159	0.00
4800	0	51	0.16	0	63	0.00	0	64	0.16	0	77	0.16	0	79	0.00
9600	0	25	0.16	0	31	0.00	0	32	–1.36	0	38	0.16	0	39	0.00
19200	0	12	0.16	0	15	0.00	0	15	1.73	0	19	–2.34	0	19	0.00
31250	0	7	0.00	0	9	–1.70	0	9	0.00	0	11	0.00	0	11	2.40
38400	—	—	—	0	7	0.00	0	7	1.73	0	9	–2.34	0	9	0.00

Bit Rate (bps)	Operating Frequency PCLK (MHz)														
	14			16			17.2032			18			19.6608		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	248	–0.17	3	70	0.03	3	75	0.48	3	79	–0.12	3	86	0.31
150	2	181	0.16	2	207	0.16	2	223	0.00	2	233	0.16	2	255	0.00
300	2	90	0.16	2	103	0.16	2	111	0.00	2	116	0.16	2	127	0.00
600	1	181	0.16	1	207	0.16	1	223	0.00	1	233	0.16	1	255	0.00
1200	1	90	0.16	1	103	0.16	1	111	0.00	1	116	0.16	1	127	0.00
2400	0	181	0.16	0	207	0.16	0	223	0.00	0	233	0.16	0	255	0.00
4800	0	90	0.16	0	103	0.16	0	111	0.00	0	116	0.16	0	127	0.00
9600	0	45	–0.93	0	51	0.16	0	55	0.00	0	58	–0.69	0	63	0.00
19200	0	22	–0.93	0	25	0.16	0	27	0.00	0	28	1.02	0	31	0.00
31250	0	13	0.00	0	15	0.00	0	16	1.20	0	17	0.00	0	19	–1.70
38400	—	—	—	0	12	0.16	0	13	0.00	0	14	–2.34	0	15	0.00

Bit Rate (bps)	Operating Frequency PCLK (MHz)								
	20			25			30		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	88	–0.25	3	110	–0.02	3	132	0.13
150	3	64	0.16	3	80	0.47	3	97	–0.35
300	2	129	0.16	2	162	–0.15	2	194	0.16
600	2	64	0.16	2	80	0.47	2	97	–0.35
1200	1	129	0.16	1	162	–0.15	1	194	0.16
2400	1	64	0.16	1	80	0.47	1	97	–0.35
4800	0	129	0.16	0	162	–0.15	0	194	0.16
9600	0	64	0.16	0	80	0.47	0	97	–0.35
19200	0	32	–1.36	0	40	–0.76	0	48	–0.35
31250	0	19	0.00	0	24	0.00	0	29	0.00
38400	0	15	1.73	0	19	1.73	0	23	1.73

Note: This is an example when the ABCS and BGDM bits in the SEMR register are 0.
When either the ABCS bit or BGDM bit is set to 1, the bit rate doubles.
When both ABCS and BGDM bits in the SEMR register are set to 1, the bit rate increases four times.

Downloaded from Arrow.com.

Table 33.15 Maximum Bit Rate for Each Operating Frequency (Asynchronous Mode)

	SEMR Settings						SEMR Settings				
PCLK (MHz)	BGDM Bit	ABCS Bit	n	N	Maximum Bit Rate (bps)	PCLK (MHz)	BGDM Bit	ABCS Bit	n	N	Maximum Bit Rate (bps)
8	0	0	0	0	250000	17.2032	0	0	0	0	537600
		1	0	0	500000			1	0	0	1075200
	1	0	0	0			1	0	0	0	
		1	0	0	1000000			1	0	0	2150400
9.8304	0	0	0	0	307200	18	0	0	0	0	562500
		1	0	0	614400			1	0	0	1125000
	1	0	0	0			1	0	0	0	
		1	0	0	1228800			1	0	0	2250000
10	0	0	0	0	312500	19.6608	0	0	0	0	614400
		1	0	0	625000			1	0	0	1228800
	1	0	0	0			1	0	0	0	
		1	0	0	1250000			1	0	0	2457600
12	0	0	0	0	375000	20	0	0	0	0	625000
		1	0	0	750000			1	0	0	1250000
	1	0	0	0			1	0	0	0	
		1	0	0	1500000			1	0	0	2500000
12.288	0	0	0	0	384000	25	0	0	0	0	781250
		1	0	0	768000			1	0	0	1562500
	1	0	0	0			1	0	0	0	
		1	0	0	1536000			1	0	0	3125000
14	0	0	0	0	437500	30	0	0	0	0	937500
		1	0	0	875000			1	0	0	1875000
	1	0	0	0			1	0	0	0	
		1	0	0	1750000			1	0	0	3750000
16	0	0	0	0	500000						
		1	0	0	1000000						
	1	0	0	0							
		1	0	0	2000000						

Downloaded from Arrow.com.

Table 33.16 Maximum Bit Rate with External Clock Input (Asynchronous Mode)

		Maximum Bit Rate (bps)	
PCLK (MHz)	External Input Clock (MHz)	SEMR.ABCS Bit = 0	SEMR.ABCS Bit = 1
8	2.0000	125000	250000
9.8304	2.4576	153600	307200
10	2.5000	156250	312500
12	3.0000	187500	375000
12.288	3.0720	192000	384000
14	3.5000	218750	437500
16	4.0000	250000	500000
17.2032	4.3008	268800	537600
18	4.5000	281250	562500
19.6608	4.9152	307200	614400
20	5.0000	312500	625000
25	6.2500	390625	781250
30	7.5000	468750	937500

Table 33.17 Maximum Bit Rate with TMR Clock Input (Asynchronous Mode)

		Maximum Bit Rate (bps)	
PCLK (MHz)	TMR Clock (MHz)	SEMR.ABCS Bit = 0	SEMR.ABCS Bit = 1
8	4	250000	500000
9.8304	4.9152	307200	614400
10	5	312500	625000
12	6	375000	750000
12.288	6.144	384000	768000
14	7	437500	875000
16	8	500000	1000000
17.2032	8.6016	537600	1075200
18	9	562500	1125000
19.6608	9.8304	614400	1228800
20	10	625000	1250000
25	12.5	781250	1562500
30	15	937500	1875000

Downloaded from Arrow.com.

Table 33.18 BRR Settings for Various Bit Rates (Clock Synchronous Mode, Simple SPI Mode)

	Operating Frequency PCLK (MHz)											
	8		10		16		20		25		30	
Bit Rate (bps)	n	N	n	N	n	N	n	N	n	N	n	N
110												
250	3	124	3	155	3	249						
500	2	249	3	77	3	124	3	155	3	194	3	233
1 k	2	124	2	155	2	249	3	77	3	97	3	116
2.5 k	1	199	1	249	2	99	2	124	2	155	2	187
5 k	1	99	1	124	1	199	1	249	2	77	2	93
10 k	0	199	0	249	1	99	1	124	1	155	1	187
25 k	0	79	0	99	0	159	0	199	0	249	1	74
50 k	0	39	0	49	0	79	0	99	0	124	0	149
100 k	0	19	0	24	0	39	0	49	0	62	0	74
250 k	0	7	0	9	0	15	0	19	0	24	0	29
500 k	0	3	0	4	0	7	0	9	—	—	0	14
1 M	0	1			0	3	0	4	—	—		
2 M	0	0*1			0	1			—	—		
2.5 M			0	0*1			0	1			0	2
4 M					0	0*1						
5 M							0	0*1				
6.25 M									0	0*1		
7.5 M											0	0*1

Blank cell: Cannot be set since the bit rate error exceeds 5%.

—: Can be set, but a bit rate error of 1 to 5% will occur.

Note 1. Continuous transmission or reception is not possible. After transmitting/receiving one frame of data, there is an interval of a 1-bit period before starting transmitting/receiving the next frame of data. The output of the synchronization clock is stopped for a 1-bit period. For this reason, it takes 9 bits worth of time to transfer one frame (8 bits) of data, and the average transfer rate is 8/9 times the bit rate.

Table 33.19 Maximum Bit Rate with External Clock Input (Clock Synchronous Mode, Simple SPI Mode)

PCLK (MHz)	External Input Clock (MHz)	Maximum Bit Rate (Mbps)
8	1.3333	1.3333
10	1.6667	1.6667
12	2.0000	2.0000
14	2.3333	2.3333
16	2.6667	2.6667
18	3.0000	3.0000
20	3.3333	3.3333
25	4.1667	4.1667
30	5.0000	5.0000

Downloaded from Arrow.com.

Table 33.20 BRR Settings for Various Bit Rates (Smart Card Interface Mode, n = 0, S = 372)

Bit Rate (bps)	PCLK (MHz)	n	N	Error (%)
9600	7.1424	0	0	0.00
	10.00	0	1	–30.00
	10.7136	0	1	–25.00
	13.00	0	1	–8.99
	14.2848	0	1	0.00
	16.00	0	1	12.01
	18.00	0	2	–15.99
	20.00	0	2	–6.66
	25.00	0	3	–12.49
	30.00	0	3	5.01

Table 33.21 Maximum Bit Rate for Each Operating Frequency (Smart Card Interface Mode, S = 32)

PCLK (MHz)	Maximum Bit Rate (bps)	n	N
10.00	156250	0	0
10.7136	167400	0	0
13.00	203125	0	0
16.00	250000	0	0
18.00	281250	0	0
20.00	312500	0	0
25.00	390625	0	0
30.00	468750	0	0

Downloaded from Arrow.com.

Table 33.22 BRR Settings for Various Bit Rates (Simple I²C Mode)

Bit Rate (bps)	Operating Frequency PCLK (MHz)														
	8			10			16			20			25		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
10 k	0	24	0.0	0	31	–2.3	1	12	–3.8	1	15	–2.3	1	19	–2.3
25 k	0	9	0.0	0	12	–3.8	1	4	0.0	1	6	–10.7	1	7	–2.3
50 k	0	4	0.0	0	6	–10.7	1	2	–16.7	1	3	–21.9	1	3	–2.3
100 k	0	2	–16.7	0	3	–21.9	0	4	0.0	0	6	–10.7	1	1	–2.3
250 k	0	0	0.0	0	1	–37.5	0	1	0.0	0	2	–16.7	0	3	–21.9
350 k										0	1	–10.7	0	2	–25.6

Bit Rate (bps)	Operating Frequency PCLK (MHz)		
	30		
	n	N	Error (%)
10 k	1	23	–2.3
25 k	1	9	–6.3
50 k	1	4	–6.3
100 k	1	2	–21.9
250 k	0	3	–6.3
350 k	0	2	–10.7

Table 33.23 Minimum Widths at High and Low Level for SCL at Various Bit Rates (Simple I²C Mode)

Bit Rate (bps)	Operating Frequency PCLK (MHz)											
	8			10			16			20		
	n	N	Min. Widths at High/Low Level for SCL (µs)	n	N	Min. Widths at High/Low Level for SCL (µs)	n	N	Min. Widths at High/Low Level for SCL (µs)	n	N	Min. Widths at High/Low Level for SCL (µs)
10 k	0	24	43.75/50.00	0	31	44.80/51.20	1	12	45.50/52.00	1	15	44.80/51.20
25 k	0	9	17.50/20.00	0	12	18.20/20.80	1	4	17.50/20.00	1	6	19.60/22.40
50 k	0	4	8.75/10.00	0	6	9.80/11.20	1	2	10.50/12.00	1	3	11.20/12.80
100 k	0	2	5.25/6.00	0	3	5.60/6.40	0	4	4.37/5.00	0	6	4.90/5.60
250 k	0	0	1.75/2.00	0	1	2.80/3.20	0	1	1.75/2.00	0	2	2.10/2.40
350 k										0	1	1.40/1.60

Bit Rate (bps)	Operating Frequency PCLK (MHz)					
	25			30		
	n	N	Min. Widths at High/Low Level for SCL (µs)	n	N	Min. Widths at High/Low Level for SCL (µs)
10 k	1	19	44.80/51.20	1	23	44.80/51.20
25 k	1	7	17.92/20.48	1	9	18.66/21.33
50 k	1	3	8.96/10.24	1	4	9.33/10.66
100 k	1	1	4.48/5.12	1	2	5.60/6.40
250 k	0	3	2.24/2.56	0	3	1.86/2.13
350 k	0	2	1.68/1.92	0	2	1.40/1.60

Downloaded from Arrow.com.

33.2.12 Modulation Duty Register (MDDR)

Address(es): SCI1.MDDR 0008 A032h, SCI5.MDDR 0008 A0B2h, SCI8.MDDR 0008 A112h, SCI12.MDDR 0008 B312h

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1

The MDDR register corrects the bit rate adjusted by the BRR register.

When the SEMR.BRME bit is set to 1, the bit rate generated by the on-chip baud rate generator is evenly corrected according to the settings of the MDDR register (M/256). The relationship between the MDDR register setting (M) and the bit rate (B) is given in Table 33.24.

The range of the value that can be set in the MDDR register is from 80h to FFh. A value other than these cannot be set.

The MDDR register is writable only when the TE and RE bits in the SCR register are 0.

Table 33.24 Relationship between MDDR Setting (M) and Bit Rate (B) When Bit Rate Modulation Function is Used

Mode	SEMR Settings		BRR Setting	Error (%)
	BGDM Bit	ABCS Bit		
Asynchronous, multi-processor communication	0	0	$N = \frac{PCLK \times 10^6}{64 \times 2^{2n-1} \times \frac{256}{M} \times B} - 1$	$Error = \left\{ \frac{PCLK \times 10^6}{B \times 64 \times 2^{2n-1} \times \frac{256}{M} \times (N+1)} - 1 \right\} \times 100$
	1	0	$N = \frac{PCLK \times 10^6}{32 \times 2^{2n-1} \times \frac{256}{M} \times B} - 1$	$Error = \left\{ \frac{PCLK \times 10^6}{B \times 32 \times 2^{2n-1} \times \frac{256}{M} \times (N+1)} - 1 \right\} \times 100$
	0	1		
	1	1	$N = \frac{PCLK \times 10^6}{16 \times 2^{2n-1} \times \frac{256}{M} \times B} - 1$	$Error = \left\{ \frac{PCLK \times 10^6}{B \times 16 \times 2^{2n-1} \times \frac{256}{M} \times (N+1)} - 1 \right\} \times 100$
Clock synchronous, simple SPI*1			$N = \frac{PCLK \times 10^6}{8 \times 2^{2n-1} \times \frac{256}{M} \times B} - 1$	
Smart card interface			$N = \frac{PCLK \times 10^6}{S \times 2^{2n+1} \times \frac{256}{M} \times B} - 1$	$Error = \left\{ \frac{PCLK \times 10^6}{B \times S \times 2^{2n+1} \times \frac{256}{M} \times (N+1)} - 1 \right\} \times 100$
Simple I²C*2			$N = \frac{PCLK \times 10^6}{64 \times 2^{2n-1} \times \frac{256}{M} \times B} - 1$	

B: Bit rate (bps)
M: MDDR setting (128 ≤ MDDR ≤ 256)
N: BRR setting for baud rate generator (0 ≤ N ≤ 255)
PCLK: Operating frequency (MHz)
n and S: Determined by the settings of the SMR and SCMR registers as listed in Table 33.12 and Table 33.13, section 33.2.11, Bit Rate Register (BRR).

Note 1. Do not use this function in clock synchronous mode and in the highest speed settings in simple SPI mode (SMR.CKS[1:0] = 00b, SCR.CKE[1] = 0, and BRR = 0).

Note 2. Adjust the bit rate so that the widths at high and low level of the SCL output in simple I²C mode satisfy the I²C-bus standard.

Downloaded from Arrow.com.

Smaller settings of the SMR.CKS[1:0] bits and larger settings of the BRR register reduce difference in the length of the 1-bit period.

33.2.13 Serial Extended Mode Register (SEMR)

Address(es): SCI1.SEMR 0008 A027h, SCI5.SEMR 0008 A0A7h, SCI8.SEMR 0008 A107h, SCI12.SEMR 0008 B307h

	b7	b6	b5	b4	b3	b2	b1	b0
	RXDESEL	BGDM	NFEN	ABCS	—	BRME	—	ACS0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ACS0	Asynchronous Mode Clock Source Select	(Valid only in asynchronous mode) 0: External clock input 1: Logical AND of two compare matches output from TMR (valid for SCI5 and SCI12 only)	R/W*1
b1	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b2	BRME	Bit Rate Modulation Enable	0: Bit rate modulation function is disabled. 1: Bit rate modulation function is enabled.	R/W*1
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	ABCS	Asynchronous Mode Base Clock Select	(Valid only in asynchronous mode) 0: Selects 16 base clock cycles for 1-bit period. 1: Selects 8 base clock cycles for 1-bit period.	R/W*1
b5	NFEN	Digital Noise Filter Function Enable	(In asynchronous mode) 0: Noise cancellation function for the RXDn input signal is disabled. 1: Noise cancellation function for the RXDn input signal is enabled. (in simple I^2C mode) 0: Noise cancellation function for the SSCLn and SSDAn input signals is disabled. 1: Noise cancellation function for the SSCLn and SSDAn input signals is enabled. The NFEN bit should be 0 in any mode other than above.	R/W*1
b6	BGDM	Baud Rate Generator Double-Speed Mode Select	(Only valid the SCR.CKE[1] bit is 0 in asynchronous mode) 0: Baud rate generator outputs the clock with normal frequency. 1: Baud rate generator outputs the clock with doubled frequency.	R/W*1
b7	RXDESEL	Asynchronous Start Bit Edge Detection Select	(Valid only in asynchronous mode) 0: The low level on the RXDn pin is detected as the start bit. 1: A falling edge on the RXDn pin is detected as the start bit.	R/W*1

Note 1. Writable only when the SCR.TE bit is 0 and the SCR.RE bit is 0 (both serial transmission and reception are disabled).

The SEMR register is used to select a clock source for 1-bit period in asynchronous mode or a detection method of the start bit.

Downloaded from Arrow.com.

ACS0 Bit (Asynchronous Mode Clock Source Select)

Selects the clock source in the asynchronous mode.

The ACS0 bit is valid in asynchronous mode (SMR.CM bit = 0) and when an external clock input is selected (SCR.CKE[1:0] bits = 10b or 11b). This bit is used to select an external clock input or the logical AND of compare matches output from the internal TMR.

Set the ACS0 bit to 0 in other than asynchronous mode.

For SCI5 and SCI12, the TMOn output (n = 0, 1) of TMR unit 0 can be set as the base clock source. Refer to Table 33.25 for details.

The ACS0 bits for SCI1 and SCI8 are reserved. The write values to these bits for SCI1 and SCI8 should be 0.

Table 33.25 Correspondence between SCI Channels and Compare Match Outputs

SCI	TMR	Compare Match Output
SCI5	Unit 0	TMO0, TMO1
SCI12	Unit 0	TMO0, TMO1

Figure 33.4 shows a setting example of when TMO0 and TMO1 in the TMR unit 0 are selected for output.

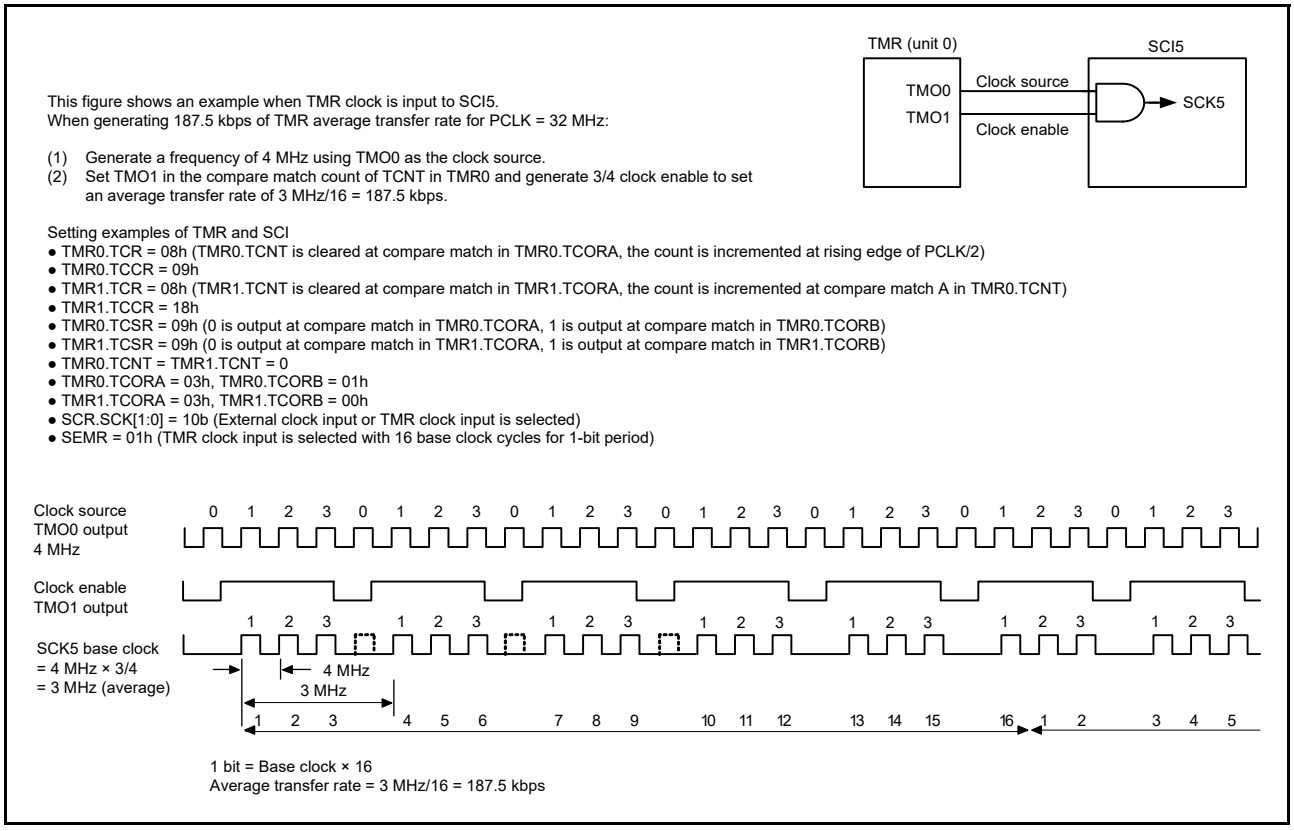


Figure 33.4 Example of Average Transfer Rate Setting When TMR Clock is Input

Downloaded from Arrow.com.

BRME Bit (Bit Rate Modulation Enable)

This bit enables and disables the bit rate modulation function. The bit rate generated by on-chip baud rate generator is evenly corrected when this function is enabled.

NFEN Bit (Digital Noise Filter Function Enable)

This bit enables or disables the digital noise filter function.
When the function is enabled, noise cancellation is applied to the RXDn input signal in asynchronous mode, and noise cancellation is applied to the SSDAn and SSCLn input signals in simple I^2C mode.
In any mode other than above, set the NFEN bit to 0 to disable the digital noise filter function.
When the function is disabled, input signals are transferred as is, as internal signals.

BGDM Bit (Baud Rate Generator Double-Speed Mode Select)

Selects the cycle of output clock for the baud rate generator.
This bit is valid when the on-chip baud rate generator is selected as the clock source (SCR.CKE[1] = 0) in asynchronous mode (SMR.CM = 0). For the clock output from the baud rate generator, either normal or doubled frequency can be selected. The base clock is generated by the clock output from the baud rate generator. When the BGDM bit is set to 1, the base clock cycle is halved and the bit rate is doubled.
Set this bit to 0 in modes other than asynchronous mode.

RXDESEL Bit (Asynchronous Start Bit Edge Detection Select)

Selects the detection method of the start bit for reception in asynchronous mode. When a break occurs, data receiving operation depends on the settings of this bit. Set this bit to 1 when reception should be stopped while a break occurs or when reception should be started without retaining the RXDn pin input at high level for the period of one data frame or longer after completion of the break.
Set this bit to 0 in modes other than asynchronous mode.

Downloaded from Arrow.com.

33.2.14 Noise Filter Setting Register (SNFR)

Address(es): SCI1.SNFR 0008 A028h, SCI5.SNFR 0008 A0A8h, SCI8.SNFR 0008 A108h, SCI12.SNFR 0008 B308h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	NFCS[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	NFCS[2:0]	Noise Filter Clock Select	In asynchronous mode, the standard setting for the base clock is as follows. b2 b0 0 0 0: The clock signal divided by 1 is used with the noise filter. In simple I²C mode, the standard settings for the clock source of the on-chip baud rate generator selected by the SMR.CKS[1:0] bits are given below. b2 b0 0 0 1: The clock signal divided by 1 is used with the noise filter. 0 1 0: The clock signal divided by 2 is used with the noise filter. 0 1 1: The clock signal divided by 4 is used with the noise filter. 1 0 0: The clock signal divided by 8 is used with the noise filter. Settings other than above are prohibited.	R/W*1
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Writing to these bits is only possible when the RE and TE bits in the SCR register are 0 (serial reception and transmission disabled).

NFCS[2:0] Bits (Noise Filter Clock Select)

These bits select the sampling clock for the digital noise filter. To use the noise filter in asynchronous mode, set these bits to 000b. In simple I²C mode, set the bits to a value in the range from 001b to 100b.

Downloaded from Arrow.com.

33.2.15 I²C Mode Register 1 (SIMR1)

Address(es): SCI1.SIMR1 0008 A029h, SCI5.SIMR1 0008 A0A9h, SCI8.SIMR1 0008 A109h, SCI12.SIMR1 0008 B309h

	b7	b6	b5	b4	b3	b2	b1	b0
	IICDL[4:0]					—	—	IICM
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	IICM	Simple I²C Mode Select	SMIF IICM 0 0: Asynchronous mode, Multi-processor mode, Clock synchronous mode (in asynchronous mode, synchronous, or simple SPI mode) 0 1: Simple I²C mode 1 0: Smart card interface mode 1 1: Setting prohibited.	R/W*1
b2, b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7 to b3	IICDL[4:0]	SSDA Output Delay Select	(Cycles below are of the clock signal from the on-chip baud rate generator.) b7 b3 0 0 0 0 0: No output delay 0 0 0 0 1: 0 to 1 cycle 0 0 0 1 0: 1 to 2 cycles 0 0 0 1 1: 2 to 3 cycles 0 0 1 0 0: 3 to 4 cycles 0 0 1 0 1: 4 to 5 cycles : : 1 1 1 1 0: 29 to 30 cycles 1 1 1 1 1: 30 to 31 cycles	R/W*1

Note 1. Writing to these bits is only possible when the RE and TE bits in the SCR register are 0 (both serial transmission and reception are disabled).

SIMR1 is used to select simple I²C mode and the number of delay stages for the SSDA output.

IICM Bit (Simple I²C Mode Select)

In conjunction with the SCMR.SMIF bit, this bit selects the operating mode.

IICDL[4:0] Bits (SSDA Output Delay Select)

These bits are used to set a delay for output on the SSDAn pin relative to the falling edge of the output on the SSCLn pin. The available delay settings range from no delay to 31 cycles, with the clock signal from the on-chip baud rate generator as the base. The signal obtained by frequency-dividing PCLK by the divisor set in the SMR.CKS[1:0] bits is supplied as the clock signal from the on-chip baud rate generator. Set these bits to 00000b unless operation is in simple I²C mode. In simple I²C mode, set the bits to a value in the range from 00001b to 11111b.

Downloaded from Arrow.com.

33.2.16 I²C Mode Register 2 (SIMR2)

Address(es): SCI1.SIMR2 0008 A02Ah, SCI5.SIMR2 0008 A0AAh, SCI8.SIMR2 0008 A10Ah, SCI12.SIMR2 0008 B30Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	IICACKT	—	—	—	IICCSC	IICINTM
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	IICINTM	I²C Interrupt Mode Select	0: Use ACK/NACK interrupts. 1: Use reception and transmission interrupts.	R/W*1
b1	IICCSC	Clock Synchronization	0: No synchronization with the clock signal 1: Synchronization with the clock signal	R/W*1
b4 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5	IICACKT	ACK Transmission Data	0: ACK transmission 1: NACK transmission and reception of ACK/NACK	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Writing to these bits is only possible when the RE and TE bits in the SCR register are 0 (serial reception and transmission disabled).

SIMR2 is used to select how reception and transmission are controlled in simple I²C mode.

IICINTM Bit (I²C Interrupt Mode Select)

This bit selects the sources of interrupt requests in simple I²C mode.

IICCSC Bit (Clock Synchronization)

Set the IICCSC bit to 1 if the internally generated SSCLn clock signal is to be synchronized when the SSCLn pin has been placed at the low level in the case of a wait inserted by the other device, etc.

The SSCLn clock signal is not synchronized if the IICCSC bit is 0. The SSCLn clock signal is generated in accord with the rate selected in the BRR regardless of the level being input on the SSCLn pin.

Set the IICCSC bit to 1 except during debugging.

IICACKT Bit (ACK Transmission Data)

Transmitted data contains ACK bits. Set this bit to 1 when ACK and NACK bits are received.

Downloaded from Arrow.com.

33.2.17 I2C Mode Register 3 (SIMR3)

Address(es): SCI1.SIMR3 0008 A02Bh, SCI5.SIMR3 0008 A0ABh, SCI8.SIMR3 0008 A10Bh, SCI12.SIMR3 0008 B30Bh

	b7	b6	b5	b4	b3	b2	b1	b0
	IICSCLS[1:0]		IICSDAS[1:0]		IICSTIF	IICSTP REQ	IICRST AREQ	IICSTA REQ
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	IICSTAREQ	Start Condition Generation	0: A start condition is not generated. 1: A start condition is generated.*1, *3, *4, *5	R/W
b1	IICRSTAREQ	Restart Condition Generation	0: A restart condition is not generated. 1: A restart condition is generated.*2, *3, *4, *5	R/W
b2	IICSTPREQ	Stop Condition Generation	0: A stop condition is not generated. 1: A stop condition is generated.*2, *3, *4, *5	R/W
b3	IICSTIF	Issuing of Start, Restart, or Stop Condition Completed Flag	0: There are no requests for generating conditions or a condition is being generated. 1: A start, restart, or stop condition is completely generated.	R/W
b5, b4	IICSDAS[1:0]	SSDA Output Select	b5 b4 0 0: Serial data output 0 1: Generate a start, restart, or stop condition. 1 0: Output the low level on the SSDAn pin. 1 1: Place the SSDAn pin in the high-impedance state.	R/W
b7, b6	IICSCLS[1:0]	SSCL Output Select	b7 b6 0 0: Serial clock output 0 1: Generate a start, restart, or stop condition. 1 0: Output the low level on the SSCLn pin. 1 1: Place the SSCLn pin in the high-impedance state.	R/W

Note 1. Generate a start condition only when the SSCLn and SSDAn pins are both high (the corresponding bits in the corresponding PIDR registers are 1).
Note 2. Generate a restart or stop condition only when the SSCLn pin is low (the corresponding bit in the PIDR register is 0).
Note 3. Do not set more than one from among the IICSTAREQ, IICRSTAREQ, and IICSTPREQ bits to 1 at a given time.
Note 4. Execute the generation of a condition after the value of the IICSTIF flag is 0.
Note 5. Do not write 0 to this bit while it is 1. Generation of a condition is suspended by writing 0 to this bit while it is 1.

SIMR3 is used to control the simple I2C mode start and stop conditions, and to hold the SSDAn and SSCLn pins at fixed levels.

IICSTAREQ Bit (Start Condition Generation)

When a start condition is to be generated, set both the IICSDAS[1:0] and IICSCLS[1:0] bits to 01b as well as setting the IICSTAREQ bit to 1.

[Setting condition]

- Writing 1 to the bit

[Clearing condition]

- Completion of generation of the start condition

Downloaded from Arrow.com.

IICRSTAREQ Bit (Restart Condition Generation)

When a restart condition is to be generated, set both the IICSDAS[1:0] and IICSCLS[1:0] bits to 01b as well as setting the IICRSTAREQ bit to 1.

[Setting condition]

- Writing 1 to the bit

[Clearing condition]

- Completion of generation of the restart condition

IICSTPREQ Bit (Stop Condition Generation)

When a stop condition is to be generated, set both the IICSDAS[1:0] and IICSCLS[1:0] bits to 01b as well as setting the IICSTPREQ bit to 1.

[Setting condition]

- Writing 1 to the bit

[Clearing condition]

- Completion of generation of the stop condition

IICSTIF Flag (Issuing of Start, Restart, or Stop Condition Completed Flag)

After generating a condition, this bit indicates that the generation is completed. When using the IICSTAREQ, IICRSTAREQ, or IICSTPREQ bit to cause generation of a condition, do so after setting the IICSTIF flag to 0.

When the IICSTIF flag is 1 while an interrupt request is enabled by setting the SCR.TEIE bit, an STI request is output.

[Setting condition]

- Completion of the generation of a start, restart, or stop condition (however, in cases where this conflicts with any of the conditions for the flag becoming 0 listed below, the other condition takes precedence)

[Clearing conditions]

- Writing 0 to the bit (confirm that the IICSTIF flag is 0 before doing so)
- Writing 0 to the SIMR1.IICM bit (when operation is not in simple I^2C mode)
- Writing 0 to the SCR.TE bit

IICSDAS[1:0] Bits (SSDA Output Select)

These bits control output from the SSDAn pin.

Set the IICSDAS[1:0] and IICSCLS[1:0] bits to the same value during normal operations.

IICSCLS[1:0] Bits (SSCL Output Select)

These bits control output from the SSCLn pin.

Set the IICSCLS[1:0] and IICSDAS[1:0] bits to the same value during normal operations.

Downloaded from Arrow.com.

33.2.18 I²C Status Register (SISR)

Address(es): SCI1.SISR 0008 A02Ch, SCI5.SISR 0008 A0ACh, SCI8.SISR 0008 A10Ch, SCI12.SISR 0008 B30Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	IICACKR
Value after reset:	0	0	x	x	0	x	0	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	IICACKR	ACK Reception Data Flag	0: ACK received 1: NACK received	R/W*1
b1	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b2	—	Reserved	The read value is undefined.	R
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5, b4	—	Reserved	The read value is undefined.	R
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Only 0 can be written to this bit, to clear the flag.

SISR is used to monitor state in relation to simple I²C mode.

IICACKR Flag (ACK Reception Data Flag)

Received ACK and NACK bits can be read from this bit.
The IICACKR flag is updated at the rising of SSCLn clock for the ACK/NACK receiving bit.

Downloaded from Arrow.com.

33.2.19 SPI Mode Register (SPMR)

Address(es): SCI1.SPMR 0008 A02Dh, SCI5.SPMR 0008 A0ADh, SCI8.SPMR 0008 A10Dh, SCI12.SPMR 0008 B30Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	CKPH	CKPOL	—	MFF	—	MSS	CTSE	SSE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SSE	SSn# Pin Function Enable	0: SSn# pin function is disabled. 1: SSn# pin function is enabled.	R/W*1
b1	CTSE	CTS Enable	0: CTS function is disabled (RTS output function is enabled). 1: CTS function is enabled.	R/W*1
b2	MSS	Master Slave Select	0: Transmission is through the SMOSIn pin and reception is through the SMISOn pin (master mode). 1: Reception is through the SMOSIn pin and transmission is through the SMISOn pin (slave mode).	R/W*1
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	MFF	Mode Fault Flag	0: No mode fault error 1: Mode fault error	R/W*2
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	CKPOL	Clock Polarity Select	0: Clock polarity is not inverted. 1: Clock polarity is inverted.	R/W*1
b7	CKPH	Clock Phase Select	0: Clock is not delayed. 1: Clock is delayed.	R/W*1

Note 1. Writing to these bits is only possible when the RE and TE bits in the SCR register are 0 (both serial transmission and reception are disabled).
Note 2. Only 0 can be written to these bits, which clears the flag.

SPMR is used to select the extension settings in asynchronous and clock synchronous modes.

SSE Bit (SSn# Pin Function Enable)

Set this bit to 1 if the SSn# pin is to be used in control of transmission and reception (in simple SPI mode). Set this bit to 0 in any other mode. Furthermore, even for usage in simple SPI mode, the SSn# pin on the master side is not required to control reception and transmission when master mode (SCR.CKE[1:0] = 00b and MSS = 0) is selected and there is a single master, so the setting for the SSE bit is 0. Do not set both the SSE and CTSE bits to enabled (even if this setting is made, operation is the same as that when these bits are set to 0).

CTSE Bit (CTS Enable)

Set this bit to 1 if the SSn# pin is to be used for inputting of the CTS control signal to control of transmission and reception. The RTS signal is output when this bit is set to 0. Set this bit to 0 in smart card interface mode, simple SPI mode, and simple I^2C mode. Do not set both the CTSE and SSE bits to enabled (even if this setting is made, operation is the same as that when these bits are set to 0).

MSS Bit (Master Slave Select)

This bit selects between master and slave operation in simple SPI mode. When the MSS bit is set to 1, data is received through the SMOSIn pin and transmitted through the SMISOn pin.
Set this bit to 0 in modes other than simple SPI mode.

Downloaded from Arrow.com.

MFF Flag (Mode Fault Flag)

This bit indicates mode fault errors.

In a multi-master configuration, determine the mode fault error occurrence by reading the MFF flag.

[Setting condition]

- Input on the SSn# pin being at the low level during master operation in simple SPI mode (SSE bit = 1 and MSS bit = 0)

[Clearing condition]

- Writing 0 to the bit after it was read as 1

CKPOL Bit (Clock Polarity Select)

This bit selects the polarity of the clock signal output through the SCKn pin. Refer to Figure 33.58 for details.

Set the bit to 0 in other than simple SPI mode and clock synchronous mode.

CKPH Bit (Clock Phase Select)

This bit selects the phase of the clock signal output through the SCKn pin. Refer to Figure 33.58 for details.

Set the bit to 0 in other than simple SPI mode and clock synchronous mode.

33.2.20 Extended Serial Module Enable Register (ESMER)

Address(es): SCI12.ESMER 0008 B320h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	ESME
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ESME	Extended Serial Mode Enable	0: The extended serial mode is disabled. 1: The extended serial mode is enabled.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

ESME Bit (Extended Serial Mode Enable)

When the ESME bit is 1, the facilities of the extended serial mode control section are enabled.

When the ESME bit is 0, the extended serial mode control section is initialized.

Table 33.26 Settings of the ESME Bit and Timer Operation Mode

ESME Bit	Timer Mode	Break Field Low Width Determination Mode	Break Field Low Width Output Mode
0	Available*1	Not available	Not available
1	Available	Available	Available

Note 1. Operation is only possible with PCLK selected.

Downloaded from Arrow.com.

33.2.21 Control Register 0 (CR0)

Address(es): SCI12.CR0 0008 B321h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	BRME	RXDSF	SFSF	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b1	SFSF	Start Frame Status Flag	0: Start Frame detection function is disabled. 1: Start Frame detection function is enabled.	R
b2	RXDSF	RXDX12 Input Status Flag	0: RXDX12 input is enabled. 1: RXDX12 input is disabled.	R
b3	BRME	Bit Rate Measurement Enable	0: Measurement of bit rate is disabled. 1: Measurement of bit rate is enabled.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

33.2.22 Control Register 1 (CR1)

Address(es): SCI12.CR1 0008 B322h

	b7	b6	b5	b4	b3	b2	b1	b0
	PIBS[2:0]			PIBE	CF1DS[1:0]		CF0RE	BFE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	BFE	Break Field Enable	0: Break Field detection is disabled. 1: Break Field detection is enabled.	R/W
b1	CF0RE	Control Field 0 Reception Enable	0: Reception of Control Field 0 is disabled. 1: Reception of Control Field 0 is enabled.	R/W
b3, b2	CF1DS[1:0]	Control Field 1 Data Register Select	b3 b2 0 0: Selects comparison with the value in the PCF1DR register. 0 1: Selects comparison with the value in the SCF1DR register. 1 0: Selects comparison with the values in the PCF1DR and SCF1DR registers. 1 1: Setting prohibited.	R/W
b4	PIBE	Priority Interrupt Bit Enable	0: The priority interrupt bit is disabled. 1: The priority interrupt bit is enabled.	R/W
b7 to b5	PIBS[2:0]	Priority Interrupt Bit Select	b7 b5 0 0 0: 0th bit of Control Field 1 0 0 1: 1st bit of Control Field 1 0 1 0: 2nd bit of Control Field 1 0 1 1: 3rd bit of Control Field 1 1 0 0: 4th bit of Control Field 1 1 0 1: 5th bit of Control Field 1 1 1 0: 6th bit of Control Field 1 1 1 1: 7th bit of Control Field 1	R/W

Downloaded from Arrow.com.

33.2.23 Control Register 2 (CR2)

Address(es): SCI12.CR2 0008 B323h

	b7	b6	b5	b4	b3	b2	b1	b0
	RTS[1:0]		BCCS[1:0]		—	DFCS[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	DFCS[2:0]	RXDX12 Signal Digital Filter Clock Select	b2 b0 0 0 0: Filter is disabled. 0 0 1: Filter clock is base clock*1, *2 0 1 0: Filter clock is PCLK/8 0 1 1: Filter clock is PCLK/16 1 0 0: Filter clock is PCLK/32 1 0 1: Filter clock is PCLK/64 1 1 0: Filter clock is PCLK/128 1 1 1: Setting prohibited	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5, b4	BCCS[1:0]	Bus Collision Detection Clock Select	• When SEMR.BGDM = 0 or SEMR.BGDM = 1 and SMR.CKS[1:0] = a value other than 00b b5 b4 0 0: Base clock 0 1: Base clock frequency divided by 2 1 0: Base clock frequency divided by 4 1 1: Setting prohibited • When SEMR.BGDM = 1 and SMR.CKS[1:0] = 00b b5 b4 0 0: Base clock frequency divided by 2 0 1: Base clock frequency divided by 4 1 0: Setting prohibited 1 1: Setting prohibited	R/W
b7, b6	RTS[1:0]	RXDX12 Reception Sampling Timing Select	• When SCI12.SEMR.ABCS = 0 b7 b6 0 0: Rising edge of the 8th cycle of base clock 0 1: Rising edge of the 10th cycle of base clock 1 0: Rising edge of the 12th cycle of base clock 1 1: Rising edge of the 14th cycle of base clock • When SCI12.SEMR.ABCS = 1 b7 b6 0 0: Rising edge of the 4th cycle of base clock 0 1: Rising edge of the 5th cycle of base clock 1 0: Rising edge of the 6th cycle of base clock 1 1: Rising edge of the 7th cycle of base clock	R/W

Note: The period of the base clock is 1/16 of a single bit period when the SCI12.SEMR.ABCS is 0, and 1/8 of a single bit period when the SCI12.SEMR.ABCS is 1.
Note 1. To use the base clock, set the SCI12.SCR.TE bit to 1.
Note 2. The base clock divided by 2 is the filter clock when the SEMR.BGDM bit is 1 and the SMR.CKS[1:0] bits are 00b.

Downloaded from Arrow.com.

33.2.24 Control Register 3 (CR3)

Address(es): SCI12.CR3 0008 B324h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	SDST
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SDST	Start Frame Detection Start	0: Detection of Start Frame is not performed. 1: Detection of Start Frame is performed.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

SDST Bit (Start Frame Detection Start)

Detection of a Start Frame begins when this bit is set to 1. The bit is read as 0.

33.2.25 Port Control Register (PCR)

Address(es): SCI12.PCR 0008 B325h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	SHARPS	—	—	RXDXPS	TXDXPS
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TXDXPS	TXDX12 Signal Polarity Select	0: The polarity of TXDX12 signal is not inverted for output. 1: The polarity of TXDX12 signal is inverted for output.	R/W
b1	RXDXPS	RXDX12 Signal Polarity Select	0: The polarity of RXDX12 signal is not inverted for input. 1: The polarity of RXDX12 signal is inverted for input.	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	SHARPS	TXDX12/RXDX12 Pin Multiplexing Select	0: The TXDX12 and RXDX12 pins are independent. 1: The TXDX12 and RXDX12 signals are multiplexed on the same pin.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

SHARPS Bit (TXDX12/RXDX12 Pin Multiplexing Select)

When this bit is set to 1, the TXDX12 and RXDX12 signals are multiplexed on the same pin so that half-duplex communications become possible.

Downloaded from Arrow.com.

33.2.26 Interrupt Control Register (ICR)

Address(es): SCI12.ICR 0008 B326h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	AEDIE	BCDIE	PIBDIE	CF1MIE	CF0MIE	BFDIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	BFDIE	Break Field Low Width Detected Interrupt Enable	0: Interrupts on detection of the low width for a Break Field are disabled. 1: Interrupts on detection of the low width for a Break Field are enabled.	R/W
b1	CF0MIE	Control Field 0 Match Detected Interrupt Enable	0: Interrupts on detection of a match with Control Field 0 are disabled. 1: Interrupts on detection of a match with Control Field 0 are enabled.	R/W
b2	CF1MIE	Control Field 1 Match Detected Interrupt Enable	0: Interrupts on detection of a match with Control Field 1 are disabled. 1: Interrupts on detection of a match with Control Field 1 are enabled.	R/W
b3	PIBDIE	Priority Interrupt Bit Detected Interrupt Enable	0: Interrupts on detection of the priority interrupt bit are disabled. 1: Interrupts on detection of the priority interrupt bit are enabled.	R/W
b4	BCDIE	Bus Collision Detected Interrupt Enable	0: Interrupts on detection of a bus collision are disabled. 1: Interrupts on detection of a bus collision are enabled.	R/W
b5	AEDIE	Valid Edge Detected Interrupt Enable	0: Interrupts on detection of a valid edge are disabled. 1: Interrupts on detection of a valid edge are enabled.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Downloaded from Arrow.com.

33.2.27 Status Register (STR)

Address(es): SCI12.STR 0008 B327h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	AEDF	BCDF	PIBDF	CF1MF	CF0MF	BFDF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	BFDF	Break Field Low Width Detection Flag	[Setting conditions] • Detection of the low width for a Break Field • Completion of the output of the low width for a Break Field • Underflow of the timer [Clearing condition] • Writing 1 to the STCR.BFDCL bit	R
b1	CF0MF	Control Field 0 Match Flag	[Setting condition] • A match between the value received in Control Field 0 and the set value. [Clearing condition] • Writing 1 to the STCR.CF0MCL bit	R
b2	CF1MF	Control Field 1 Match Flag	[Setting condition] • A match between the data received in Control Field 1 and the set values. [Clearing condition] • Writing 1 to the STCR.CF1MCL bit	R
b3	PIBDF	Priority Interrupt Bit Detection Flag	[Setting condition] • Detection of the priority interrupt bit [Clearing condition] • Writing 1 to the STCR.PIBDCL bit	R
b4	BCDF	Bus Collision Detected Flag	[Setting condition] • Detection of the bus collision [Clearing condition] • Writing 1 to the STCR.BCDCL bit	R
b5	AEDF	Valid Edge Detection Flag	[Setting condition] • Detection of a valid edge [Clearing condition] • Writing 1 to the STCR.AEDCL bit	R
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R

Downloaded from Arrow.com.

33.2.28 Status Clear Register (STCR)

Address(es): SCI12.STCR 0008 B328h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	AEDCL	BCDCL	PIBDCL	CF1MCL	CF0MCL	BFDCL
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	BFDCL	BFDF Clear	Setting this bit to 1 clears the STR.BFDF flag. This bit is read as 0.	R/W
b1	CF0MCL	CF0MF Clear	Setting this bit to 1 clears the STR.CF0MF flag. This bit is read as 0.	R/W
b2	CF1MCL	CF1MF Clear	Setting this bit to 1 clears the STR.CF1MF flag. This bit is read as 0.	R/W
b3	PIBDCL	PIBDF Clear	Setting this bit to 1 clears the STR.PIBDF flag. This bit is read as 0.	R/W
b4	BCDCL	BCDF Clear	Setting this bit to 1 clears the STR.BCDF flag. This bit is read as 0.	R/W
b5	AEDCL	AEDF Clear	Setting this bit to 1 clears the STR.AEDF flag. This bit is read as 0.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

33.2.29 Control Field 0 Data Register (CF0DR)

Address(es): SCI12.CF0DR 0008 B329h

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0

The CF0DR register is an 8-bit readable and writable register that holds a value for comparison with Control Field 0.

Downloaded from Arrow.com.

33.2.30 Control Field 0 Compare Enable Register (CF0CR)

Address(es): SCI12.CF0CR 0008 B32Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	CF0CE7	CF0CE6	CF0CE5	CF0CE4	CF0CE3	CF0CE2	CF0CE1	CF0CE0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CF0CE0	Control Field 0 Bit 0 Compare Enable	0: Comparison with bit 0 of Control Field 0 is disabled. 1: Comparison with bit 0 of Control Field 0 is enabled.	R/W
b1	CF0CE1	Control Field 0 Bit 1 Compare Enable	0: Comparison with bit 1 of Control Field 0 is disabled. 1: Comparison with bit 1 of Control Field 0 is enabled.	R/W
b2	CF0CE2	Control Field 0 Bit 2 Compare Enable	0: Comparison with bit 2 of Control Field 0 is disabled. 1: Comparison with bit 2 of Control Field 0 is enabled.	R/W
b3	CF0CE3	Control Field 0 Bit 3 Compare Enable	0: Comparison with bit 3 of Control Field 0 is disabled. 1: Comparison with bit 3 of Control Field 0 is enabled.	R/W
b4	CF0CE4	Control Field 0 Bit 4 Compare Enable	0: Comparison with bit 4 of Control Field 0 is disabled. 1: Comparison with bit 4 of Control Field 0 is enabled.	R/W
b5	CF0CE5	Control Field 0 Bit 5 Compare Enable	0: Comparison with bit 5 of Control Field 0 is disabled. 1: Comparison with bit 5 of Control Field 0 is enabled.	R/W
b6	CF0CE6	Control Field 0 Bit 6 Compare Enable	0: Comparison with bit 6 of Control Field 0 is disabled. 1: Comparison with bit 6 of Control Field 0 is enabled.	R/W
b7	CF0CE7	Control Field 0 Bit 7 Compare Enable	0: Comparison with bit 7 of Control Field 0 is disabled. 1: Comparison with bit 7 of Control Field 0 is enabled.	R/W

33.2.31 Control Field 0 Receive Data Register (CF0RR)

Address(es): SCI12.CF0RR 0008 B32Bh

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0

CF0RR is a readable register that holds the value received in Control Field 0.

33.2.32 Primary Control Field 1 Data Register (PCF1DR)

Address(es): SCI12.PCF1DR 0008 B32Ch

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0

PCF1DR is an 8-bit readable and writable register that holds the 8-bit primary value for comparison with Control Field 1.

Downloaded from Arrow.com.

33.2.33 Secondary Control Field 1 Data Register (SCF1DR)

Address(es): SCI12.SCF1DR 0008 B32Dh

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0

PCF1DR is an 8-bit readable and writable register that holds the 8-bit secondary value for comparison with Control Field 1.

33.2.34 Control Field 1 Compare Enable Register (CF1CR)

Address(es): SCI12.CF1CR 0008 B32Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	CF1CE7	CF1CE6	CF1CE5	CF1CE4	CF1CE3	CF1CE2	CF1CE1	CF1CE0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CF1CE0	Control Field 1 Bit 0 Compare Enable	0: Comparison with bit 0 of Control Field 1 is disabled. 1: Comparison with bit 0 of Control Field 1 is enabled.	R/W
b1	CF1CE1	Control Field 1 Bit 1 Compare Enable	0: Comparison with bit 1 of Control Field 1 is disabled. 1: Comparison with bit 1 of Control Field 1 is enabled.	R/W
b2	CF1CE2	Control Field 1 Bit 2 Compare Enable	0: Comparison with bit 2 of Control Field 1 is disabled. 1: Comparison with bit 2 of Control Field 1 is enabled.	R/W
b3	CF1CE3	Control Field 1 Bit 3 Compare Enable	0: Comparison with bit 3 of Control Field 1 is disabled. 1: Comparison with bit 3 of Control Field 1 is enabled.	R/W
b4	CF1CE4	Control Field 1 Bit 4 Compare Enable	0: Comparison with bit 4 of Control Field 1 is disabled. 1: Comparison with bit 4 of Control Field 1 is enabled.	R/W
b5	CF1CE5	Control Field 1 Bit 5 Compare Enable	0: Comparison with bit 5 of Control Field 1 is disabled. 1: Comparison with bit 5 of Control Field 1 is enabled.	R/W
b6	CF1CE6	Control Field 1 Bit 6 Compare Enable	0: Comparison with bit 6 of Control Field 1 is disabled. 1: Comparison with bit 6 of Control Field 1 is enabled.	R/W
b7	CF1CE7	Control Field 1 Bit 7 Compare Enable	0: Comparison with bit 7 of Control Field 1 is disabled. 1: Comparison with bit 7 of Control Field 1 is enabled.	R/W

33.2.35 Control Field 1 Receive Data Register (CF1RR)

Address(es): SCI12.CF1RR 0008 B32Fh

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0

CF1RR is a readable register that holds the value received in Control Field 1.

Downloaded from Arrow.com.

33.2.36 Timer Control Register (TCR)

Address(es): SCI12.TCR 0008 B330h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	TCST
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TCST	Timer Count Start	0: Stops the timer counting 1: Starts the timer counting	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

33.2.37 Timer Mode Register (TMR)

Address(es): SCI12.TMR 0008 B331h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	TCSS[2:0]			TWRC	—	TOMS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	TOMS[1:0]	Timer Operating Mode Select*1	b1 b0 0 0: Timer mode 0 1: Break Field low width determination mode 1 0: Break Field low width output mode 1 1: Setting prohibited	R/W
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3	TWRC	Counter Write Control	0: Data is written to the reload register and counter 1: Data is written to the reload register only	R/W
b6 to b4	TCSS[2:0]	Timer Count Clock Source Select*1	b6 b4 0 0 0: PCLK 0 0 1: PCLK/2 0 1 0: PCLK/4 0 1 1: PCLK/8 1 0 0: PCLK/16 1 0 1: PCLK/32 1 1 0: PCLK/64 1 1 1: PCLK/128	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note 1. Rewrite the TOMS[1:0] and TCSS[2:0] bits only when the timer is stopped (TCST = 0).

TWRC Bit (Counter Write Control)

This bit determines whether a value written to the TPRE or TCNT register is written to the reload register only or is written to both the reload register and the counter.

Downloaded from Arrow.com.

33.2.38 Timer Prescaler Register (TPRE)

Address(es): SCI12.TPRE 0008 B332h



TPRE consists of an 8-bit reload register, a read buffer, and a counter, each of which has FFh as its initial value. The counter counts down in synchronization with the counter clock selected by the TMR.TCSS[2:0] bits, and is reloaded with the value in the reload register when it underflows. Underflows of this register provide the clock source to drive counting by the TCNT register. The reload register and read buffer are allocated to the same address. Data is written to the reload register in writing, and the counter value that has been transferred to the read buffer is returned in reading.
It takes one PCLK cycle to load a value from the reload register to the counter.

33.2.39 Timer Count Register (TCNT)

Address(es): SCI12.TCNT 0008 B333h



TCNT consists of an 8-bit reload register, a read buffer, and a counter, each of which has FFh as its initial value. This down-counter counts underflows of the TPRE register until the TCNT register underflows, and is then reloaded with the value from the reload register. The reload register and read buffer are allocated to the same address. Data is written to the reload register in writing, and the counter value that has been transferred to the read buffer is returned in reading.
It takes one PCLK cycle to load a value from the reload register to the counter.

Downloaded from Arrow.com.

33.3 Operation in Asynchronous Mode

Figure 33.5 shows the general format for asynchronous serial communications.
One frame consists of a start bit (low level), transmit/receive data, a parity bit, and stop bits (high level).
In asynchronous serial communications, the communications line is usually held in the mark state (high level).
The SCI monitors the communications line. When the SCI detects a low, it regards that as a start bit and starts serial communication.
Inside the SCI, the transmitter and receiver are independent units, enabling full-duplex communications. Both the transmitter and the receiver also have a double-buffered structure, so that data can be read or written during transmission or reception, enabling continuous data transmission and reception.

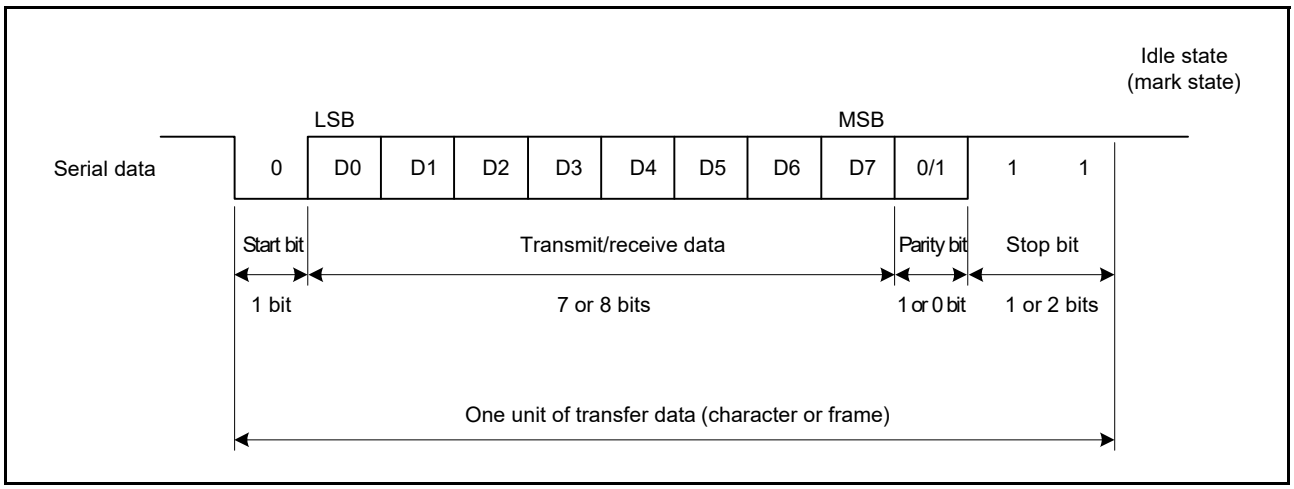


Figure 33.5 Data Format in Asynchronous Serial Communications (Example with 8-Bit Data, Parity, 2 Stop Bits)

33.3.1 Serial Data Transfer Format

Table 33.27 lists the serial data transfer formats that can be used in asynchronous mode.
Any of 18 transfer formats can be selected according to the SMR and SCMR setting. For details of multi-processor function, refer to section 33.4, Multi-Processor Communications Function.

Downloaded from Arrow.com.

Table 33.27 Serial Transfer Formats (Asynchronous Mode)

SCMR Setting	SMR Setting				Serial Transfer Format and Frame Length
CHR1	CHR	PE	MP	STOP	1 2 3 4 5 6 7 8 9 10 11 12 13
0	0	0	0	0	S 9-bit data STOP
0	0	0	0	1	S 9-bit data STOP STOP
0	0	1	0	0	S 9-bit data P STOP
0	0	1	0	1	S 9-bit data P STOP STOP
1	0	0	0	0	S 8-bit data STOP
1	0	0	0	1	S 8-bit data STOP STOP
1	0	1	0	0	S 8-bit data P STOP
1	0	1	0	1	S 8-bit data P STOP STOP
1	1	0	0	0	S 7-bit data STOP
1	1	0	0	1	S 7-bit data STOP STOP
1	1	1	0	0	S 7-bit data P STOP
1	1	1	0	1	S 7-bit data P STOP STOP
0	0	—	1	0	S 9-bit data MPB STOP
0	0	—	1	1	S 9-bit data MPB STOP STOP
1	0	—	1	0	S 8-bit data MPB STOP
1	0	—	1	1	S 8-bit data MPB STOP STOP
1	1	—	1	0	S 7-bit data MPB STOP
1	1	—	1	1	S 7-bit data MPB STOP STOP

S: Start bit
STOP: Stop bit
P: Parity bit
MPB: Multi-processor bit

Downloaded from Arrow.com.

33.3.2 Receive Data Sampling Timing and Reception Margin in Asynchronous Mode

In asynchronous mode, the SCI operates on a base clock with a frequency of 16 times*1 the bit rate.
In reception, the SCI samples the falling edge of the start bit using the base clock, and performs internal synchronization. Since receive data is sampled at the rising edge of the 8th pulse*1 of the base clock, data is latched at the middle of each bit, as shown in Figure 33.6. Thus the reception margin in asynchronous mode is determined by formula (1) below.

$$M = \left| \left(0.5 - \frac{1}{2N}\right) - (L - 0.5)F - \frac{|D - 0.5|}{N}(1 + F) \right| \times 100\ (\%) \quad \cdots \text{Formula (1)}$$

M: Reception margin
N: Ratio of bit rate to clock
(N = 16 when SEMR.ABCS = 0, N = 8 when SEMR.ABCS = 1)
D: Duty cycle of clock (D = 0.5 to 1.0)
L: Frame length (L = 9 to 13)
F: Absolute value of clock frequency deviation

Assuming values of F = 0 and D = 0.5 in formula (1), the reception margin is determined by the formula below.

M = {0.5 – 1/(2 × 16)} × 100 (%) = 46.875 (%)

However, this is only the computed value, and a margin of 20% to 30% should be allowed in system design.

Note 1. These are values when the SEMR.ABCS bit is 0. When the ABCS bit is 1, a frequency of 8 times the bit rate is used as a base clock and receive data is sampled at the rising edge of the 4th pulse of the base clock.

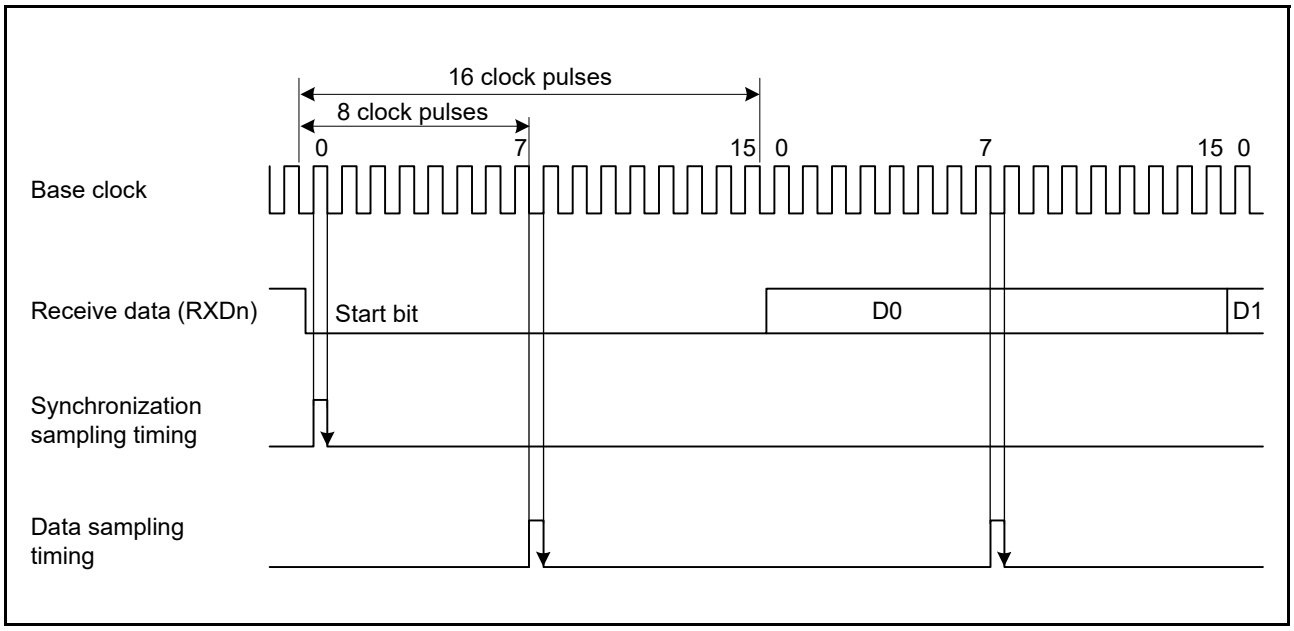


Figure 33.6 Receive Data Sampling Timing in Asynchronous Mode

Downloaded from Arrow.com.

33.3.3 Clock

Either an internal clock generated by the on-chip baud rate generator or an external clock input to the SCKn pin can be selected as the SCI's transfer clock, according to the setting of the SMR.CM bit and the SCR.CKE[1:0] bits.
When an external clock is input to the SCKn pin, the clock frequency should be 16 times the bit rate (when SEMR.ABCS bit = 0) and 8 times the bit rate (when SEMR.ABCS bit = 1). In addition, when an external clock is specified, the base clock of TMR0 can be selected by the SCIn.SEMR.ACS0 bit (n = 5, 12).
When the SCI is operated on an internal clock, the clock can be output from the SCKn pin. The frequency of the clock output in this case is equal to the bit rate, and the phase is such that the rising edge of the clock is in the middle of the transmit data, as shown in Figure 33.7.

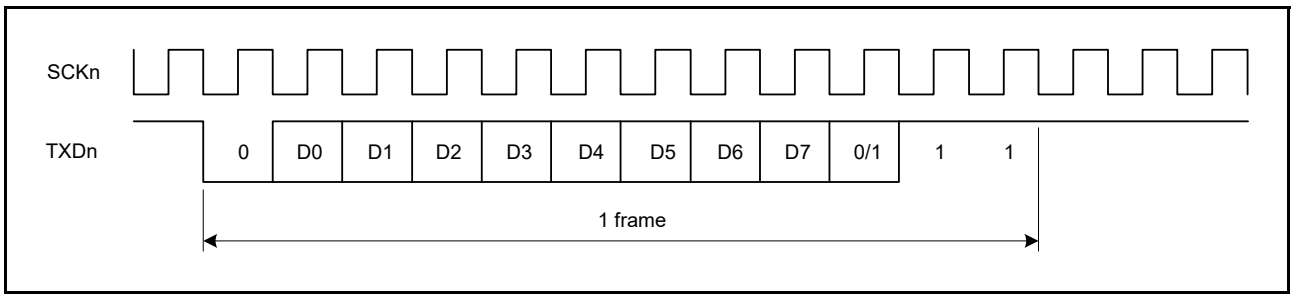


Figure 33.7 Phase Relationship between Output Clock and Transmit Data (Asynchronous Mode: SMR.CHR = 0, PE = 1, MP = 0, STOP = 1)

33.3.4 Double-Speed Mode

The output clock frequency of the on-chip baud rate generator is doubled by setting the SEMR.BGDM bit to 1, enabling high-speed communication at a doubled bit rate. If the SEMR.ABCS bit is set to 1 under the above condition, the number of base clock cycles changes from 16 to 8, so the bit rate becomes four times faster than the initial state.
As shown by Formula (1) in section 33.3.2, Receive Data Sampling Timing and Reception Margin in Asynchronous Mode, setting the SEMR.ABCS bit to 1 changes the number of cycles to 8, and the sampling interval becomes longer. This causes the reception margin to decrease. Therefore, setting the SEMR.BGDM bit to 1 and the SEMR.ABCS bit to 0 is recommended instead of setting the SEMR.BGDM bit to 0 and the SEMR.ABCS bit to 1 for high-speed operation at a doubled bit rate.

Downloaded from Arrow.com.

33.3.5 CTS and RTS Functions

The CTS function is the use of input on the CTSn# pin in transmission control. Setting the SPMR.CTSE bit to 1 enables the CTS function. When the CTS function is enabled, placing the low level on the CTSn# pin causes transmission to start.
Applying the high level to the CTS# pin while transmission is in progress does not affect transmission of the current frame, which continues.
In the RTS function, by using the function of output on the RTSn# pin, a low level is output when reception becomes possible. Conditions for output of the low and high level are shown below.

[Conditions for low-level output]
When the following conditions are all satisfied.

- The SCR.RE bit is 1.
- Reception is not in progress.
- There are no received data yet to be read.
- The ORER, FER, and PER flags in the SSR register are all 0.

[Condition for high-level output]
When the conditions for low-level output are not satisfied.

Note that either one of CTS and RTS can be selected.

Downloaded from Arrow.com.

33.3.6 SCI Initialization (Asynchronous Mode)

Before transmitting and receiving data, start by writing the initial value 00h to the SCR register and then continue through the procedure for SCI given in Figure 33.8. Whenever the operating mode or transfer format is changed, the SCR register must be initialized before the change is made.

When the external clock is used in asynchronous mode, ensure that the clock signal is supplied even during initialization.

Note that setting the SCR.RE bit to 0 initializes neither the ORER, FER, PER, and RDRF flags in the SSR register nor registers RDR, RDRH, and RDRL.

In addition, note that setting bits TIE, TE, and TEIE in the SCR register to 1 simultaneously leads to the generation of a transmit end interrupt (TEI) request before the generation of a transmit data empty interrupt (TXI) request.

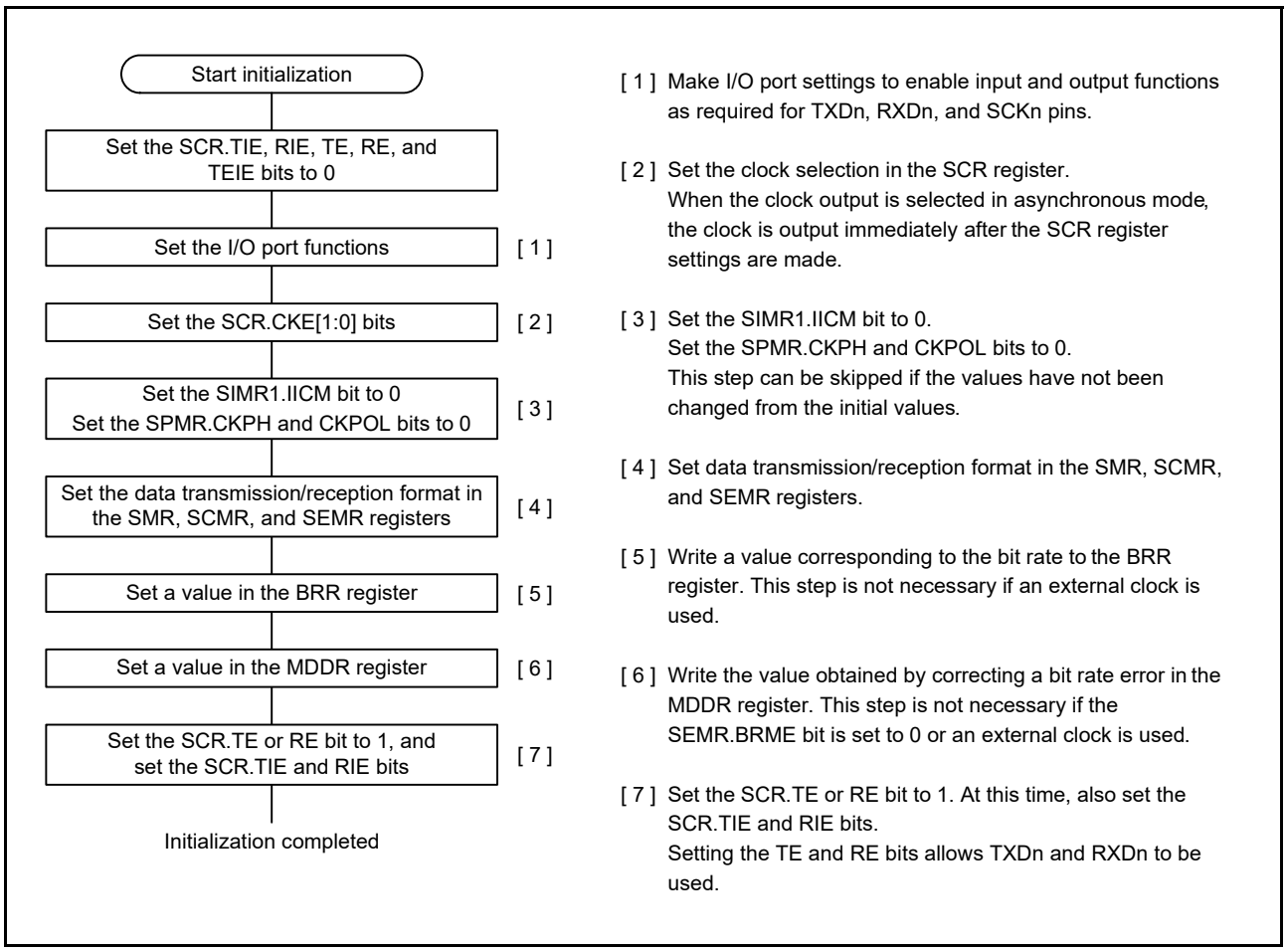


Figure 33.8 Sample SCI Initialization Flowchart (Asynchronous Mode)

Downloaded from Arrow.com.

Figure 33.9 shows an example of data transmission when the SCI is set to asynchronous mode according to the flow described in Figure 33.8 after a reset. When the pin function is set to the TXD pin, it is still high-impedance because the SCR.TE bit is 0. When the transmit data is written after setting the TE bit to 1, a data transmission starts. After the TE bit is set to 1, one frame of high is output from TXD pin (internal wait time) and then the data transmission starts.

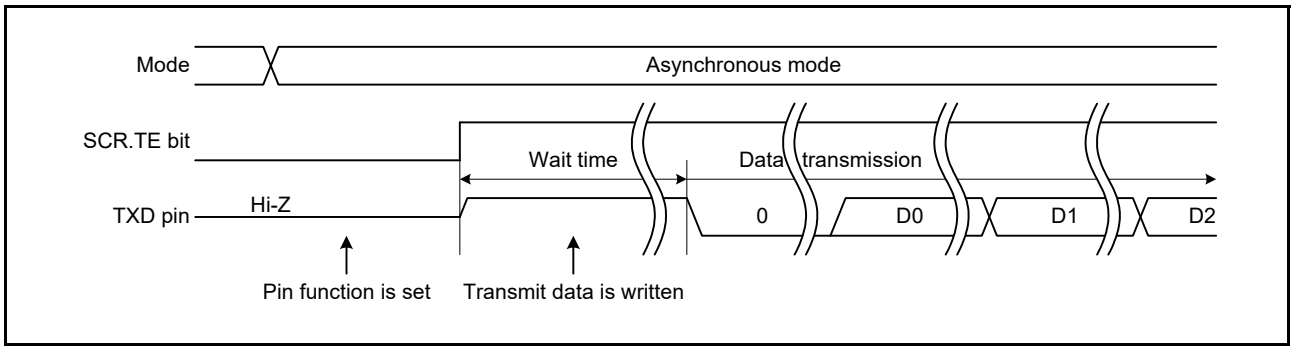


Figure 33.9 Example of Data Transmission Timing in Asynchronous Mode

Downloaded from Arrow.com.

33.3.7 Serial Data Transmission (Asynchronous Mode)

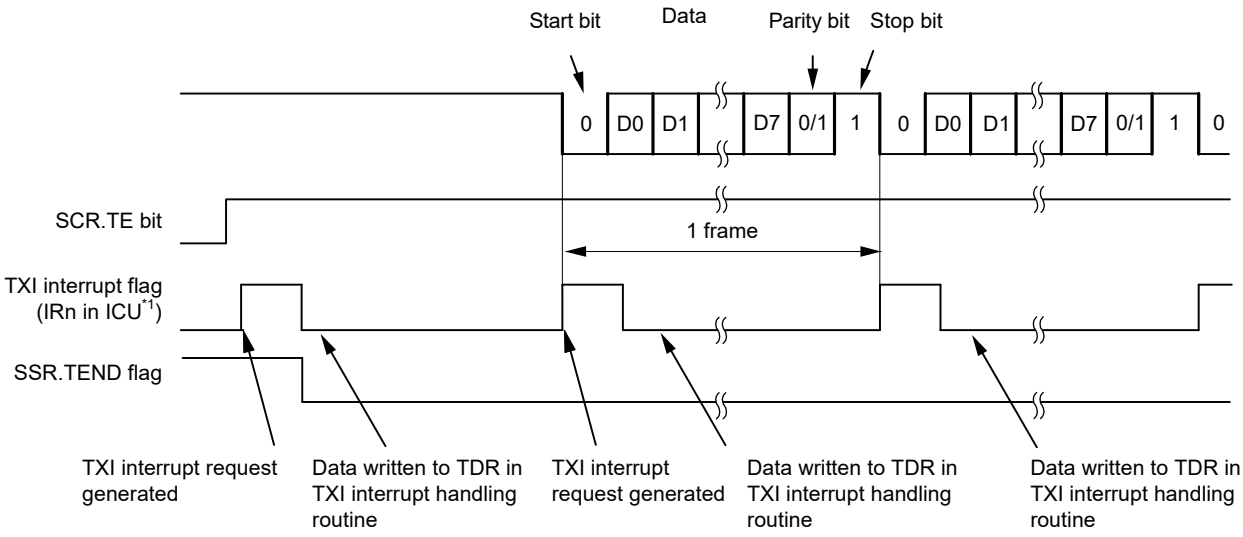
Figure 33.10 to Figure 33.12 show an example of the operation for serial transmission in asynchronous mode.
In serial transmission, the SCI operates as described below.

1. The SCI transfers data from the TDR register*1 to the TSR register when data is written to the TDR register*1 in the TXI interrupt handling routine. The TXI interrupt request at the beginning of transmission is generated when the SCR.TE bit is set to 1 after the SCR.TIE bit is set to 1 or when these 2 bits are set to 1 simultaneously by a single instruction.
2. Transmission starts after the SPMR.CTSE bit is set to 0 (CTS function is disabled) and a low level on the CTSn# pin causes data transfer from the TDR register*1 to the TSR register. If the SCR.TIE bit is 1 at this time, a TXI interrupt request is generated. Continuous transmission is obtainable by writing the next transmit data to the TDR register*1 in the TXI interrupt handling routine before transmission of the current transmit data is completed. When TEI interrupt requests are in use, set the SCR.TIE bit to 0 (a TXI interrupt request is disabled) and the SCR.TEIE bit to 1 (a TEI interrupt request is enabled) after the last of the data to be transmitted are written to the TDR register*1, *2 from the handling routine for TXI requests.
3. Data is sent from the TXDn pin in the following order: start bit, transmit data, parity bit or multi-processor bit (may be omitted depending on the format), and stop bit.
4. The SCI checks for updating of (writing to) the TDR register*3 at the time of stop bit output.
5. When the TDR register*3 is updated, setting of the SPMR.CTSE bit to 0 (CTS function is disabled) or a low level input on the CTSn# pin cause the next transfer of the next transmit data from the TDR register*1 to the TSR register and sending of the stop bit, after which serial transmission of the next frame starts.
6. If the TDR register*3 is not updated, the SCI sets the SSR.TEND flag to 1, sends the stop bit, and then outputs high to put the line in the mark state. If the SCR.TEIE bit is 1 at this time, the SSR.TEND flag is set to 1 and a TEI interrupt request is generated.

Note 1. Write data not to the TDR register but to the TDRH and TDRL registers when 9-bit data length is selected.
Note 2. Write data in the order from the TDRH register to the TDRL register when 9-bit data length is selected.
Note 3. The SCI checks for updating of the TDRL register only and does not check for updating of the TDRH register when 9-bit data length is selected.

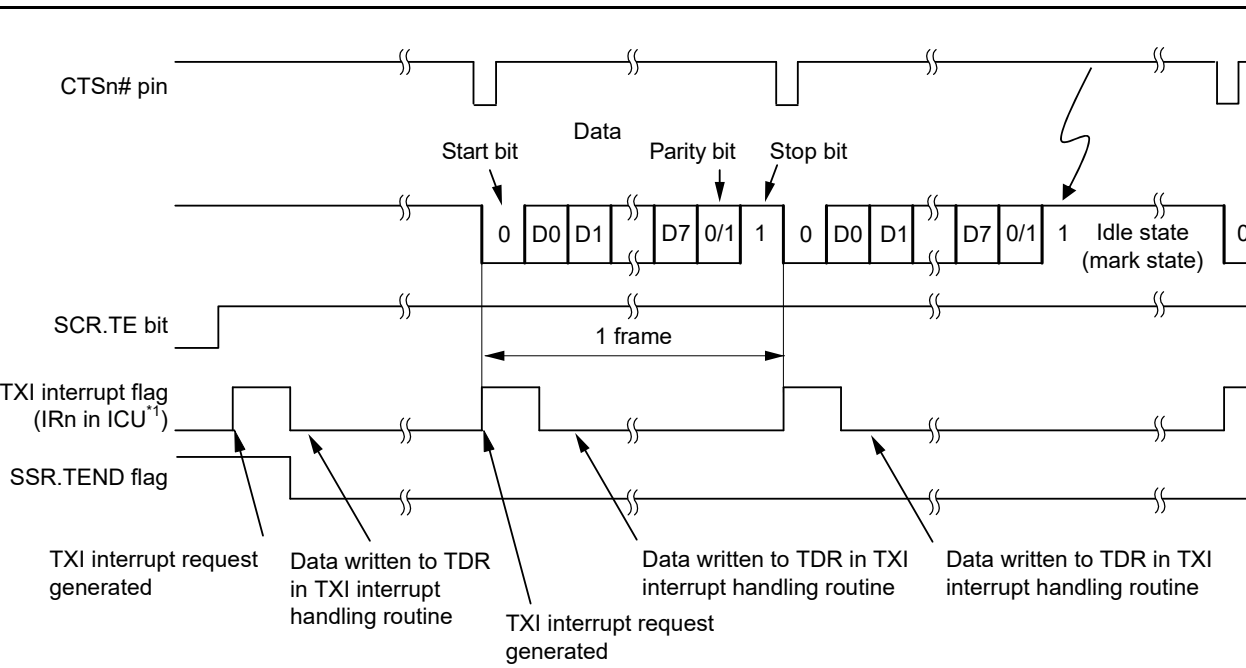
Figure 33.13 shows a sample flowchart for serial transmission in asynchronous mode.

Downloaded from Arrow.com.



Note 1. Refer to section 15, Interrupt Controller (ICUb) for details on the corresponding interrupt vector number.

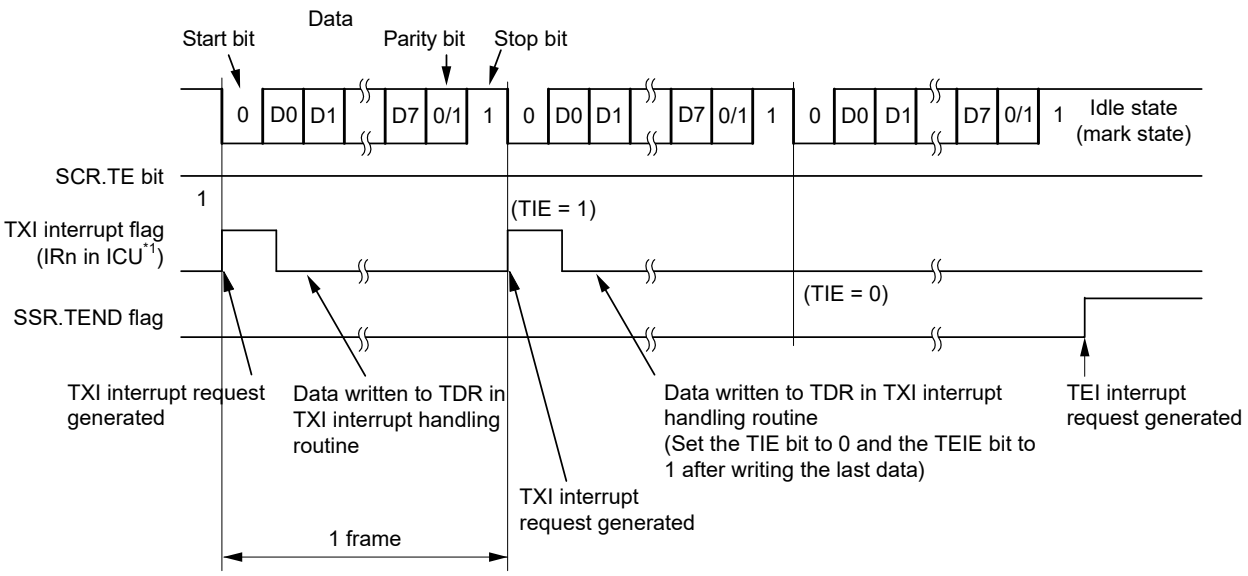
Figure 33.10 Example of Operation for Serial Transmission in Asynchronous Mode (1) (with 8-Bit Data, Parity, 1 Stop Bit, CTS Function Not Used, at the Beginning of Transmission)



Note 1. Refer to section 15, Interrupt Controller (ICUb) for details on the corresponding interrupt vector number.

Figure 33.11 Example of Operation for Serial Transmission in Asynchronous Mode (2) (with 8-Bit Data, Parity, 1 Stop Bit, CTS Function Used, at the Beginning of Transmission)

Downloaded from Arrow.com.



Note 1. Refer to section 15, Interrupt Controller (ICUb) for details on the corresponding interrupt vector number.

Figure 33.12 Example of Operation for Serial Transmission in Asynchronous Mode (3) (with 8-Bit Data, Parity, 1 Stop Bit, CTS Function Not Used, from the Middle of Transmission until Transmission Completion)

Downloaded from Arrow.com.

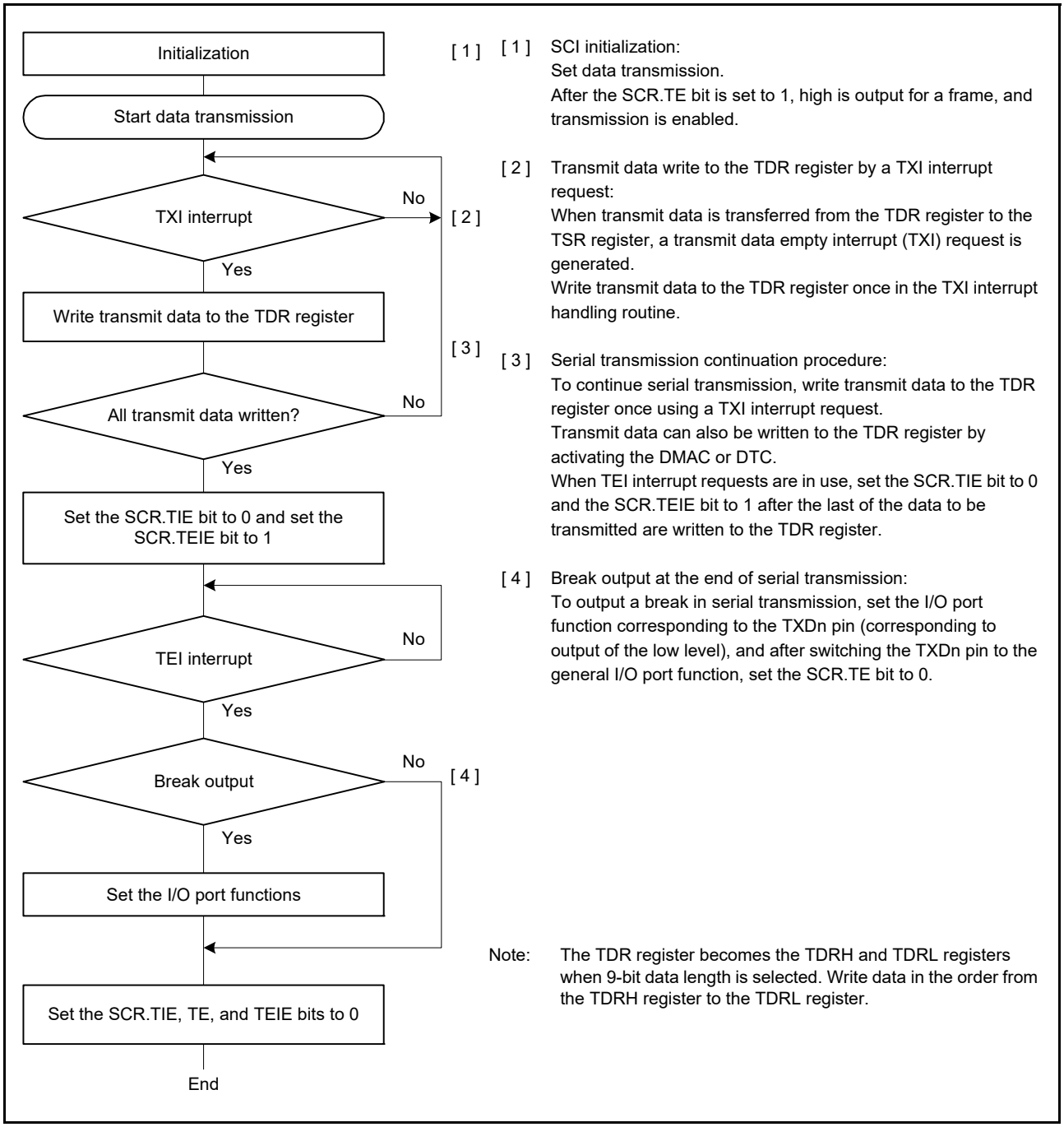


Figure 33.13 Example of Serial Transmission Flowchart in Asynchronous Mode

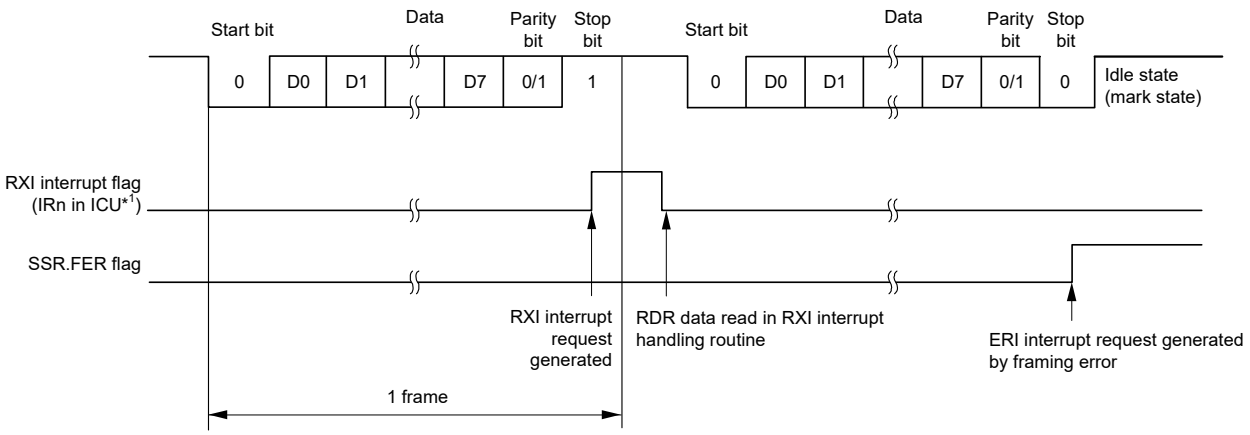
Downloaded from Arrow.com.

33.3.8 Serial Data Reception (Asynchronous Mode)

Figure 33.14 and Figure 33.15 show an example of the operation for serial data reception in asynchronous mode. In serial data reception, the SCI operates as described below.

1. When the value of the SCR.RE bit becomes 1, the output signal on the RTSn# pin goes to the low level.
2. When the SCI monitors the communications line and detects a start bit, it performs internal synchronization, stores receive data in the RSR register, and checks the parity bit and stop bit.
3. If an overrun error occurs, the SSR.ORER flag is set to 1. If the SCR.RIE bit is 1 at this time, an ERI interrupt request is generated. Receive data is not transferred to the RDR register*1.
4. If a parity error is detected, the SSR.PER flag is set to 1 and receive data is transferred to the RDR register*1. If the SCR.RIE bit is 1 at this time, an ERI interrupt request is generated.
5. If a framing error (when the stop bit is 0) is detected, the SSR.FER flag is set to 1 and receive data is transferred to the RDR register*1. If the SCR.RIE bit is 1 at this time, an ERI interrupt request is generated.
6. When reception finishes successfully, receive data is transferred to the RDR register*1. If the SCR.RIE bit is 1 at this time, an RXI interrupt request is generated. Continuous reception is enabled by reading the receive data transferred to the RDR register*1 in this RXI interrupt handling routine before reception of the next receive data is completed. Reading the received data that have been transferred to the RDR register*1 causes the RTSn# pin to output the low level.

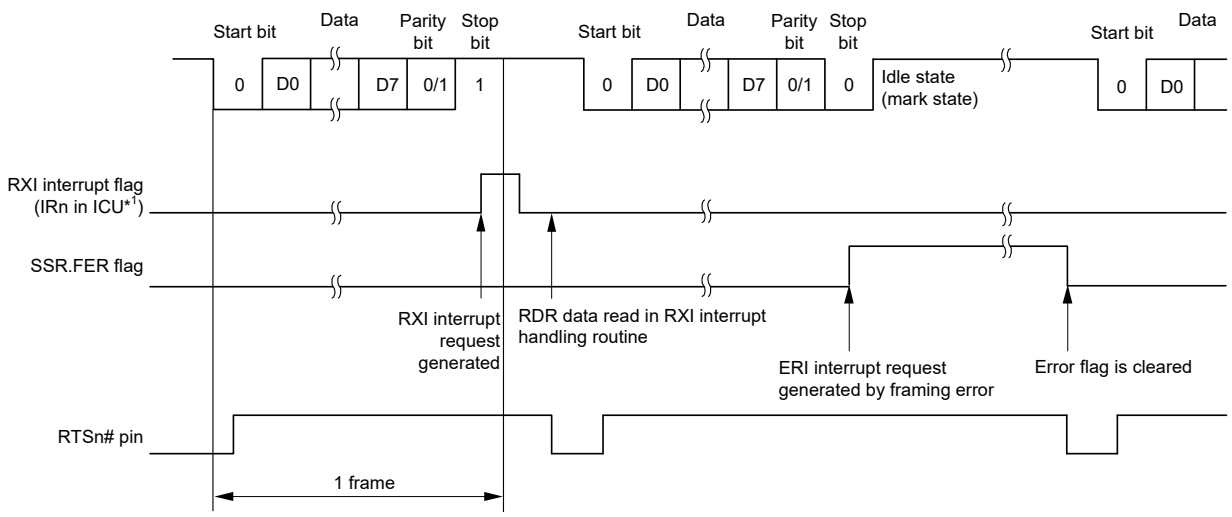
Note 1. Read data not in the RDR register but in the RDRH and RDRL registers when 9-bit data length is selected.
Note 2. The SCI checks for reading of the RDRL register only and does not check for reading of the RDRH register when 9-bit data length is selected.



Note 1. Refer to section 15, Interrupt Controller (ICUb) for details on the corresponding interrupt vector number.

Figure 33.14 Example of SCI Operation for Serial Reception in Asynchronous Mode (1) (When RTS Function is Not Used) (Example with 8-Bit Data, Parity, 1 Stop Bit)

Downloaded from Arrow.com.



Note 1. Refer to section 15, Interrupt Controller (ICUb) for details on the corresponding interrupt vector number.

Figure 33.15 Example of SCI Operation for Serial Reception in Asynchronous Mode (2) (When RTS Function is Used) (Example with 8-Bit Data, Parity, 1 Stop Bit)

Table 33.28 lists the states of the status flags in the SSR register and receive data handling when a receive error is detected.

If a receive error is detected, an ERI interrupt request is generated but an RXI interrupt request is not generated. Data reception cannot be resumed while the receive error flag is 1. Accordingly, set the ORER, FER, and PER flags to 0 before resuming reception. Moreover, be sure to read the RDR (or the RDRL) register during overrun error processing.

When a reception is forcibly terminated by setting the SCR.RE bit to 0 during operation, read the RDR (or the RDRL) register because received data which has not yet been read may be left in the RDR (or the RDRL) register.

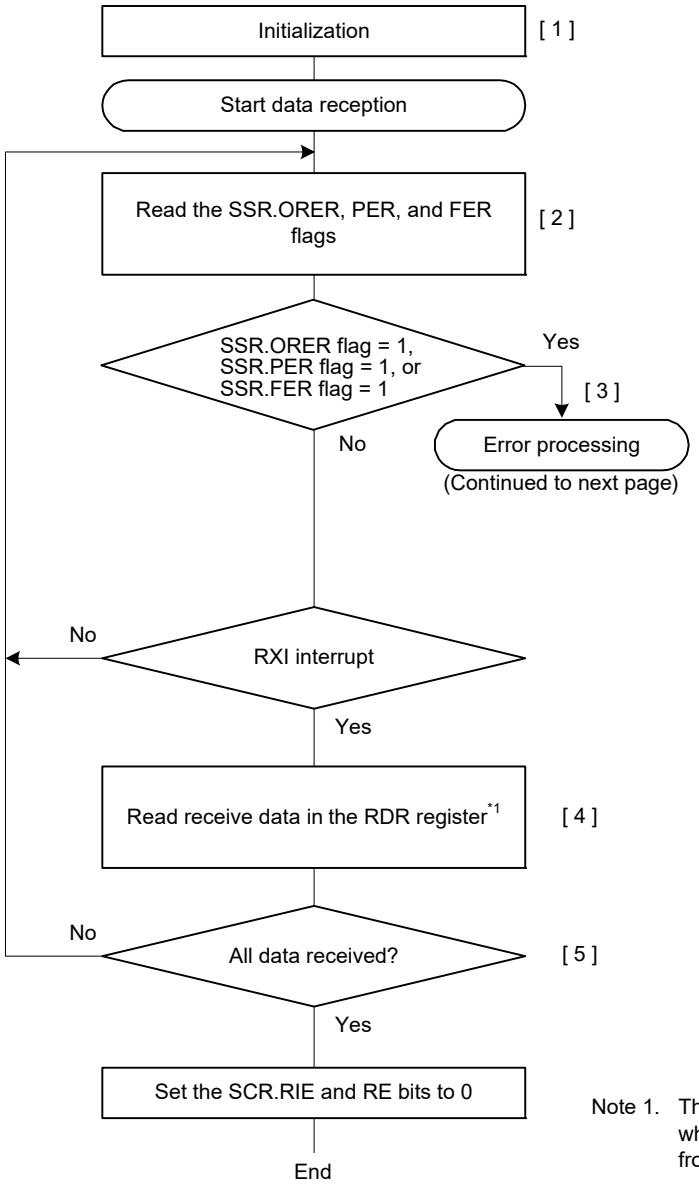
Figure 33.16 and Figure 33.17 show samples of flowcharts for serial data reception.

Table 33.28 Status Flags in the SSR Register and Receive Data Handling

Status Flags in the SSR Register				
ORER	**FER**	**PER**	**Receive Data**	**Receive Error Type**
1	0	0	Lost	Overrun error
0	1	0	Transferred to the RDR register*1	Framing error
0	0	1	Transferred to the RDR register*1	Parity error
1	1	0	Lost	Overrun error + framing error
1	0	1	Lost	Overrun error + parity error
0	1	1	Transferred to the RDR register*1	Framing error + parity error
1	1	1	Lost	Overrun error + framing error + parity error

Note 1. Read data not in the RDR register but in the RDRH and RDRL registers when 9-bit data length is selected.

Downloaded from Arrow.com.



[1] SCI initialization:
Set data reception.

[2] [3] Receive error processing and break detection:
If a receive error occurs, an ERI interrupt is generated. An error is identified by reading the ORER, PER, and FER flags in the SSR register. After performing the appropriate error processing, be sure to set the ORER, PER, and FER flags to 0. Reception cannot be resumed if any of these flags is set to 1. In the case of a framing error, a break can be detected by reading the value of the input port corresponding to the RXDn pin.

[4] Read the receive data in the RDR register once in the RXI interrupt handling routine.

[5] Serial reception continuation procedure:
To continue serial reception, before the stop bit of the current frame is received, read data from the RDR register in the RXI interrupt handling routine.
The RDR data can also be read by activating the DMAC or DTC.

Note 1. The RDR register becomes the RDRH and RDRL registers when 9-bit data length is selected. Read data in the order from the RDRH register to the RDRL register.

Figure 33.16 Example Flowchart of Serial Reception in Asynchronous Mode (1)

Downloaded from Arrow.com.

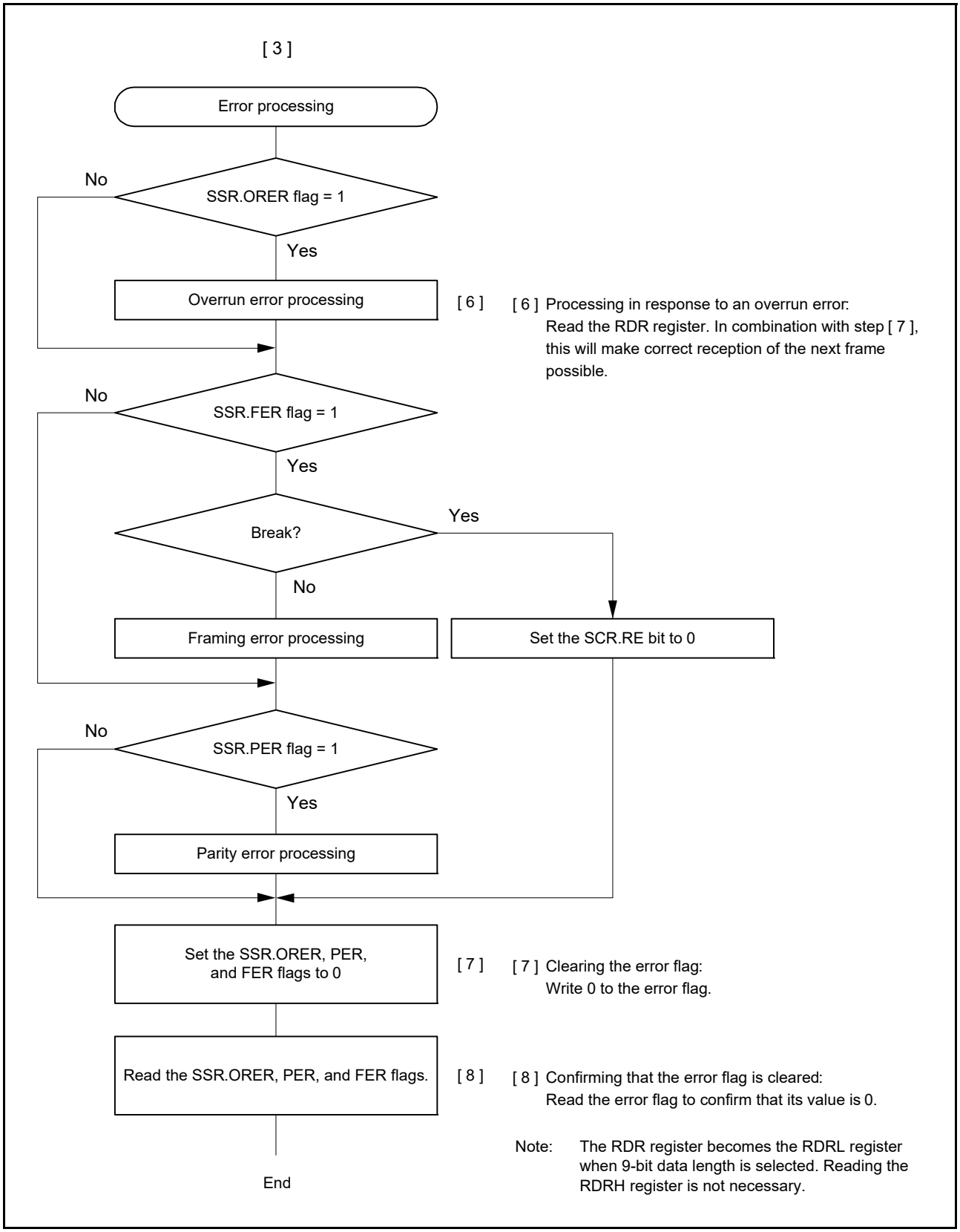


Figure 33.17 Example Flowchart of Serial Reception in Asynchronous Mode (2)

Downloaded from Arrow.com.

33.4 Multi-Processor Communications Function

Using the multi-processor communication functions enables to transmit and receive data by sharing a communication line between multiple processors by using asynchronous serial communication in which the multi-processor bit is added. In multi-processor communication, a unique ID code is allocated to each receiving station. Serial communication cycles consist of an ID transmission cycle to specify the receiving station and a data transmission cycle to transmit data to the specified receiving station. The multi-processor bit is used to distinguish between the ID transmission cycle and the data transmission cycle. When the multi-processor bit is set to 1, it indicates the ID transmission cycle and when the multi-processor bit is set to 0, it indicates the data transmission cycle. Figure 33.18 shows an example of communication between processors by using a multi-processor format. First, a transmitting station transmits communication data in which the multi-processor bit set to 1 is added to the ID code of the receiving station. Next, the transmitting station transmits the communication data in which the multi-processor bit set to 0 is added to the transmit data. Upon receiving the communication data in which the multi-processor bit is set to 1, the receiving station compares the received ID with the ID of the receiving station itself and if the two match, receives the communication data that is subsequently transmitted. If the received ID does not match with the ID of the receiving station, the receiving station skips the communication data until again receiving the communication data in which the multi-processor bit is set to 1.

For supporting this function, the SCI provides the SCR.MPIE bit. When the MPIE bit is set to 1, transfer of receive data from the RSR register to the RDR register (the RDRH and RDRL registers when 9-bit data length is selected), detection of a receive error, and setting the respective status flags RDRF, ORER, and FER in the SSR register are disabled until reception of data in which the multi-processor bit is set to 1. Upon receiving a reception character in which the multi-processor bit is set to 1, the SSR.MPB bit is set to 1 and the SCR.MPIE bit is automatically cleared, thus returning to a normal reception operation. During this time, an RXI interrupt is generated if the SCR.RIE bit is 1.

When the multi-processor format is specified, specification of the parity bit is disabled. Apart from this, there is no difference from the operation in the normal asynchronous mode. A clock which is used for the multi-processor communication is also the same as the clock used in the normal asynchronous mode.

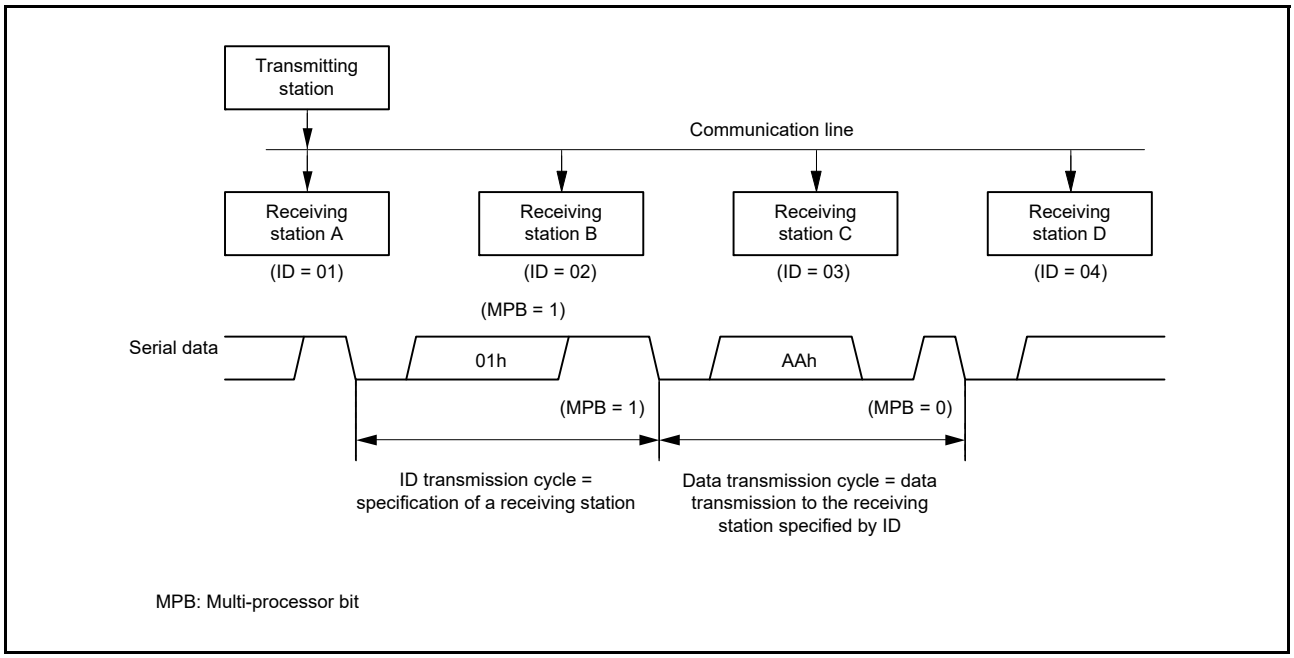


Figure 33.18 An Example of Communication using the Multi-Processor Format (Example of Transmission of Data AAh to Receiving Station A)

Downloaded from Arrow.com.

33.4.1 Multi-Processor Serial Data Transmission

Figure 33.19 is a sample flowchart of multi-processor data transmission. In the ID transmission cycle, the ID should be transmitted with the SSR.MPBT bit set to 1. In the data transmission cycle, the data should be transmitted with the MPBT bit set to 0. The other operations are the same as the operations in asynchronous mode.

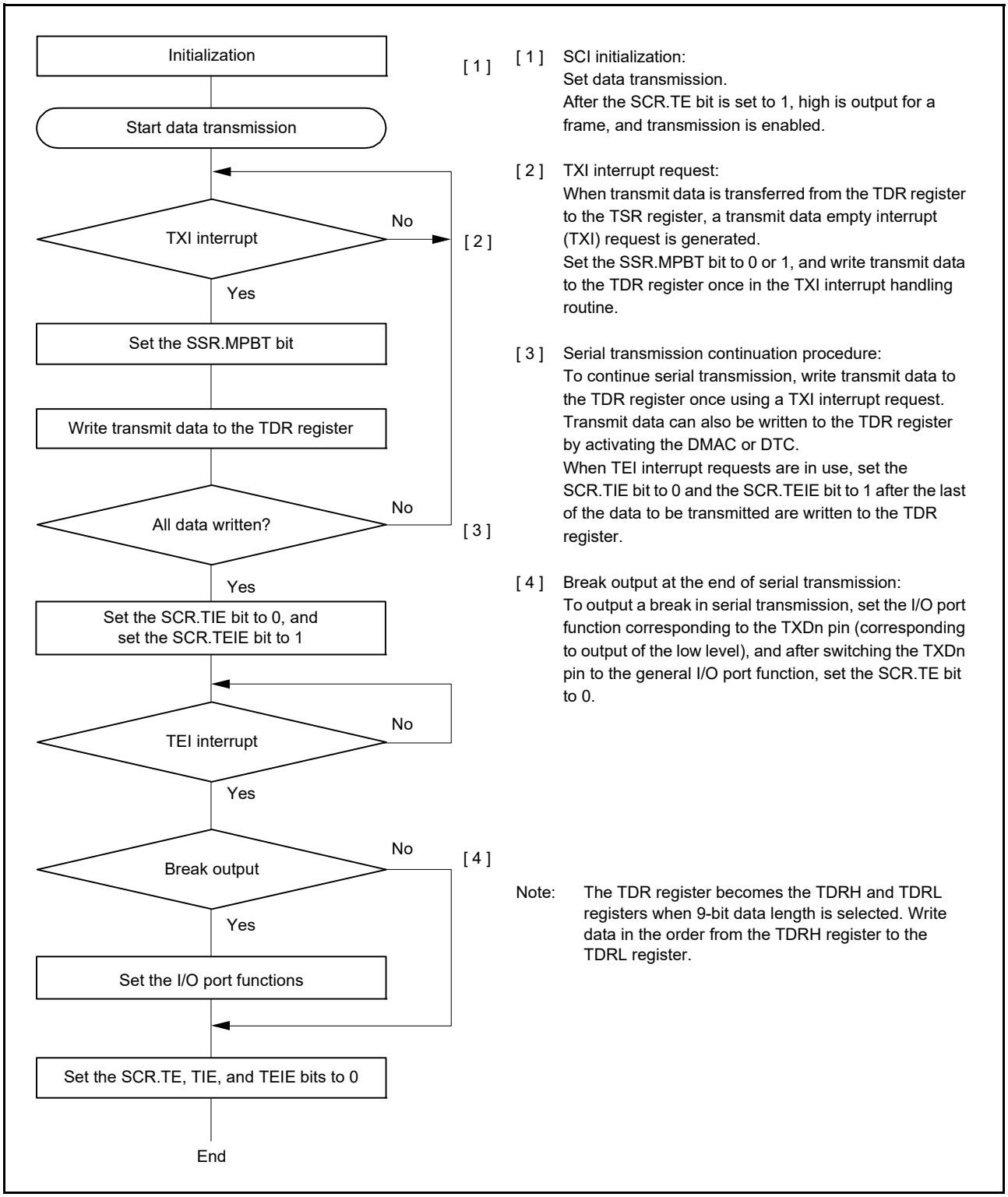
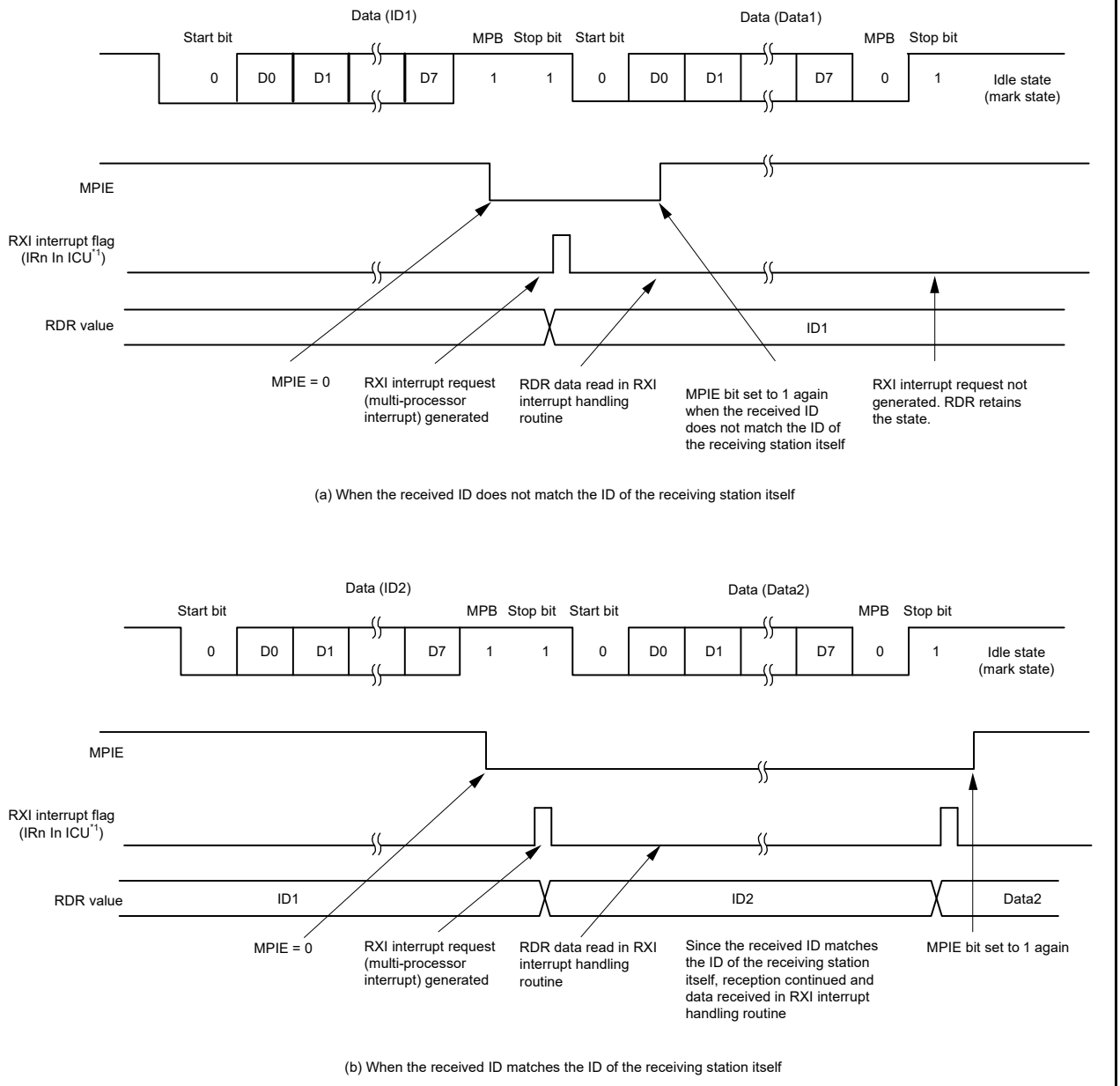


Figure 33.19 Example of Multi-Processor Serial Transmission Flowchart

Downloaded from Arrow.com.

33.4.2 Multi-Processor Serial Data Reception

Figure 33.21 and Figure 33.22 are sample flowcharts of multi-processor data reception. When the SCR.MPIE bit is set to 1, reading the communication data is skipped until reception of the communication data in which the multi-processor bit is set to 1. When the communication data in which the multi-processor bit is set to 1 is received, the received data is transferred to the RDR register (the RDRH and RDRL registers when 9-bit data length is selected). During this time, the RXI interrupt request is generated. The other operations are the same as the operations in asynchronous mode. Figure 33.20 is the example of operation for reception.



Note 1. Refer to section 15, Interrupt Controller (ICUb) for details on the corresponding interrupt vector number.

Figure 33.20 Example of SCI Reception (8-Bit Data/Multi-Processor Bit/1 Stop Bit)

Downloaded from Arrow.com.

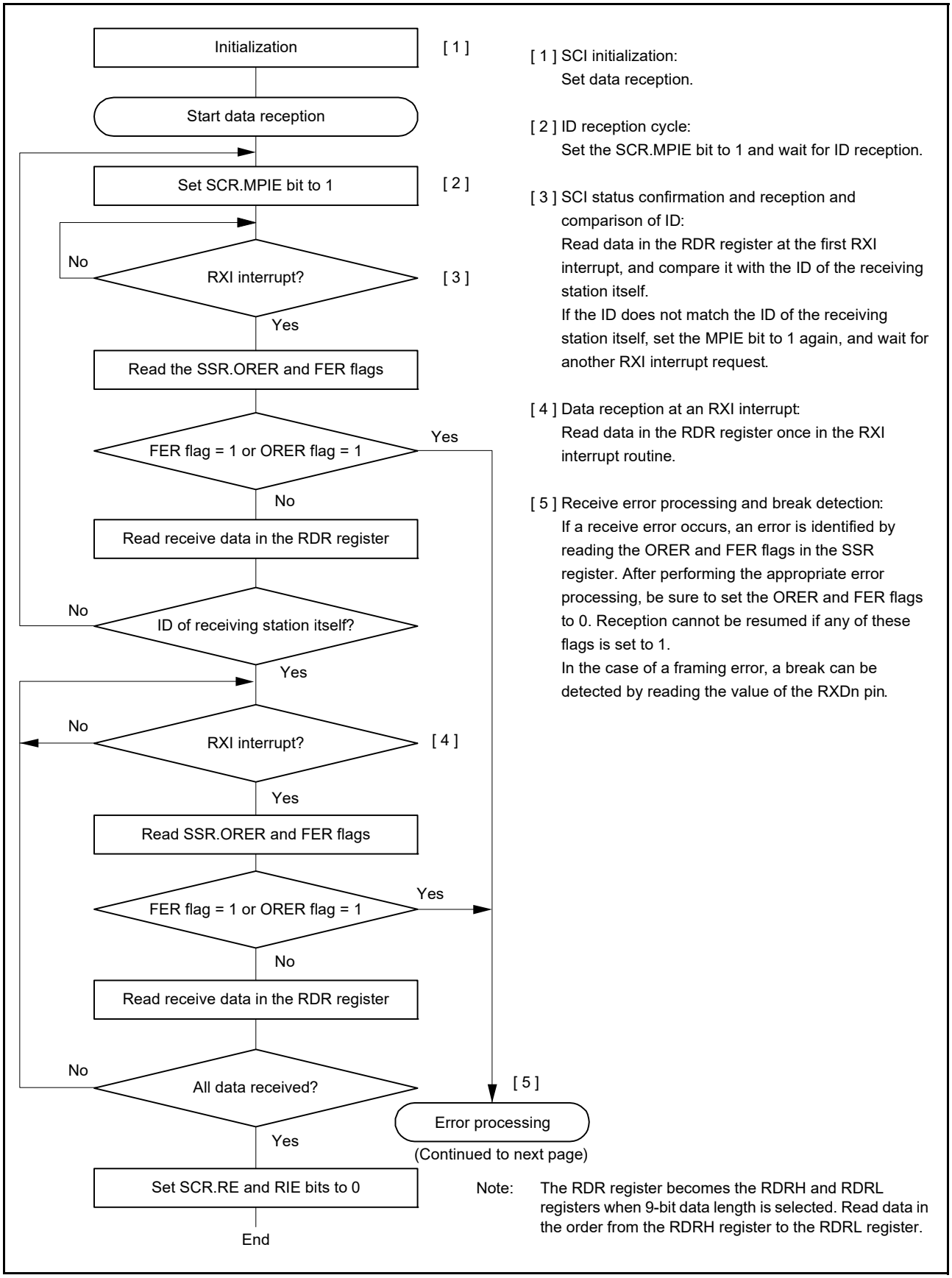


Figure 33.21 Example of Multi-Processor Serial Reception Flowchart (1)

Downloaded from Arrow.com.

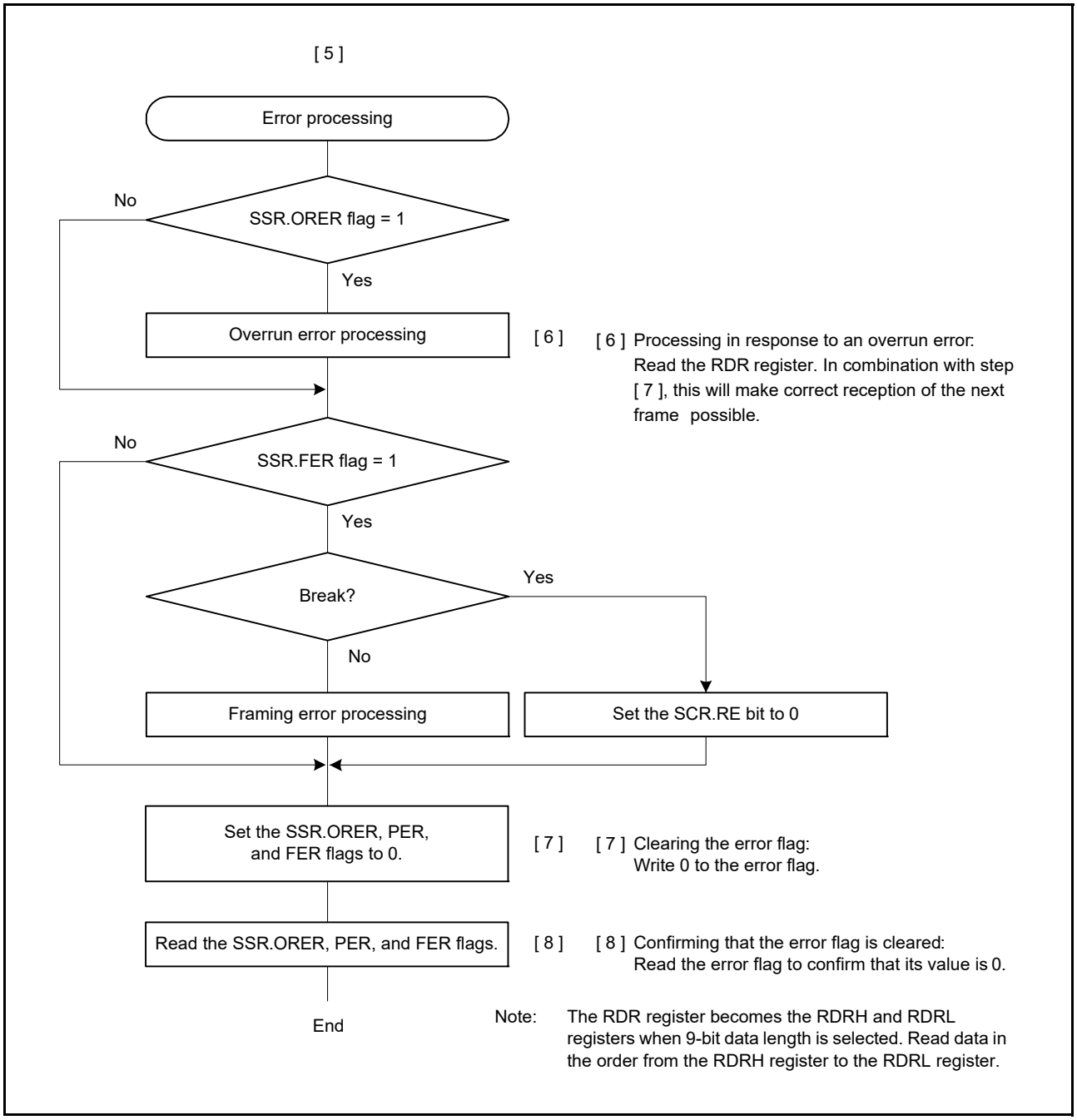


Figure 33.22 Example of Multi-Processor Serial Reception Flowchart (2)

Downloaded from Arrow.com.

33.5 Operation in Clock Synchronous Mode

Figure 33.23 shows the data format for clock synchronous serial data communications.
In clock synchronous mode, data is transmitted or received in synchronization with clock pulses. One character in transfer data consists of 8-bit data. In clock synchronous mode, no parity bit can be added.
In data transmission, the SCI outputs data from one falling edge of the synchronization clock to the next. In data reception, the SCI receives data in synchronization with the rising edge of the synchronization clock. After 8-bit data is output, the communication line holds the last bit output state.
Inside the SCI, the transmitter and receiver are independent units, enabling full-duplex communications by use of a common clock. Both the transmitter and the receiver also have a double-buffered structure, so that the next transmit data can be written during transmission or the previous receive data can be read during reception, enabling continuous data transfer.

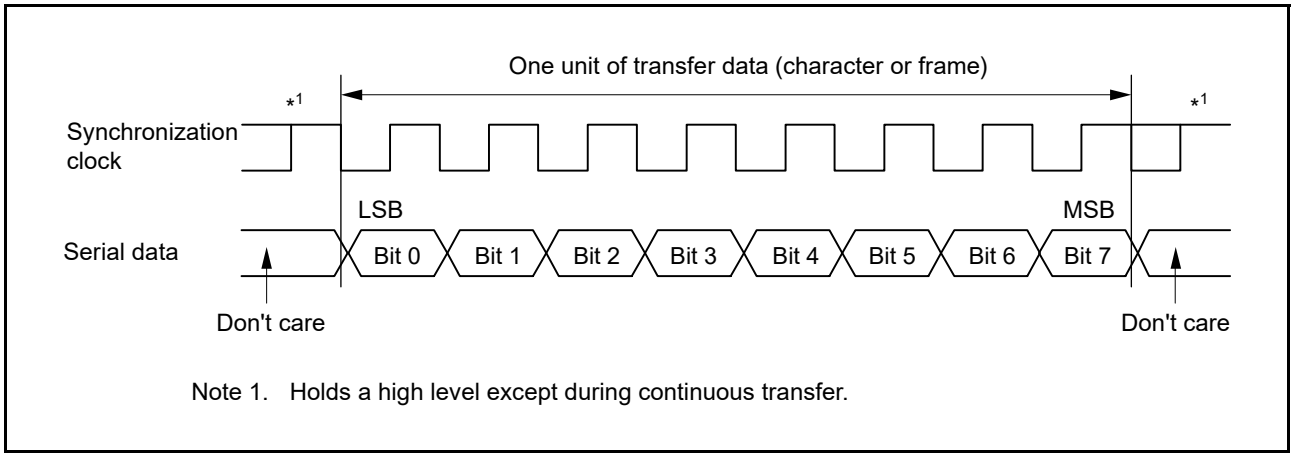


Figure 33.23 Data Format in Clock Synchronous Serial Communications (LSB First)

33.5.1 Clock

Either an internal clock generated by the on-chip baud rate generator or an external synchronization clock input at the SCKn pin can be selected, according to the setting of the SCR.CKE[1:0] bits.
When the SCI is operated on an internal clock, the synchronization clock is output from the SCKn pin. Eight synchronization clock pulses are output in the transfer of one character, and when no transfer is performed the clock is held high. However, when only data reception is performed while the CTS function is disabled, the synchronization clock output is started at the same time when the SCR.RE bit set to 1. The synchronization clock is stopped at the high level when an overrun error occurs or the SCR.RE bit is set to 0.
When only data reception is performed and the CTS function is enabled, the clock output is not started even when the SCR.RE bit set to 1 if the CTSn# pin input is high when the SCR.RE bit is 0. The synchronization clock output is started when the SCR.RE bit is set to 1 and the CTSn# pin input is low. After that, if the CTSn# pin input is high on completion of the frame reception, the synchronization clock output is stopped at the high level. If the CTSn# pin input continues to be low, the synchronization clock is stopped at the high level when an overrun error occurs or the SCR.RE bit is set to 0.

Downloaded from Arrow.com.

33.5.2 CTS and RTS Functions

In the CTS function, CTSn# pin input is used to control reception/transmission start when the clock source is the internal clock. Setting the SPMR.CTSE bit to 1 enables the CTS function.
When the CTS function is enabled, placing the low level on the CTSn# pin causes reception/transmission to start. Applying the high level to the CTS# pin while reception/transmission are in progress does not affect reception/transmission of the current frame, which continues.
In the RTS function, RTSn# pin output is used to request reception/transmission start when the clock source is an external synchronizing clock. A low level is output when serial communications become possible. Conditions for output of the low and high level are shown below.

[Conditions for low-level output]
When the following conditions are all satisfied.

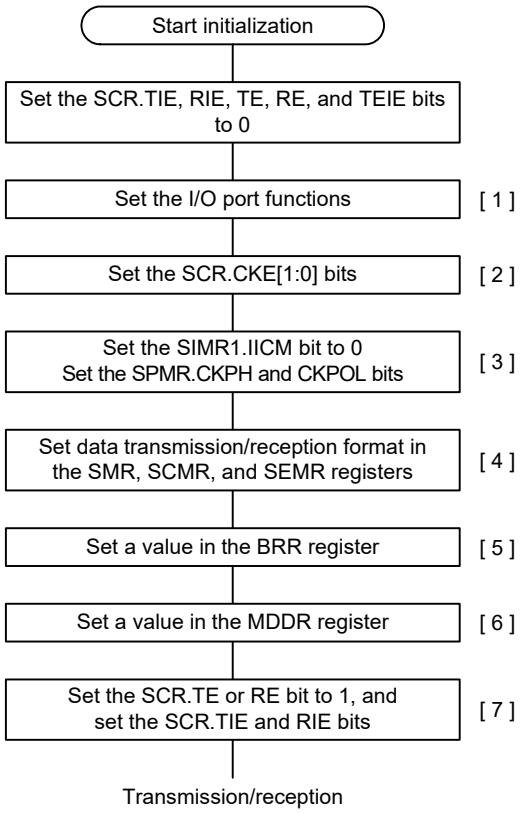
- The SCR.RE or SCR.TE bit is 1.
- Transmission or reception is not in progress.
- There are no received data yet to be read (when the SCR.RE bit is 1).
- Untransmitted data is present (when the SCR.TE bit is 1).
- The SSR.ORER flag is 0.

[Condition for high-level output]
The conditions for low-level output have not been satisfied.

Downloaded from Arrow.com.

33.5.3 SCI Initialization (Clock Synchronous Mode)

Before transmitting and receiving data, start by writing the initial value 00h to the SCR register and then continue through the procedure for SCI given in Figure 33.24. Whenever the operating mode or transfer format is changed, the SCR register must be initialized before the change is made.
Note that setting the SCR.RE bit to 0 initializes neither the ORER, FER, and PER flags in the SSR register nor the RDR register.



[1] Make I/O port settings to enable input and output functions as required for TXDn, RXDn, and SCKn pins.

[2] Set the clock selection in the SCR register.
When an internal clock is selected, the SCK pin functions as the clock output pin.

[3] Set the SIMR1.IICM bit to 0.
Set the SPMR.CKPH and CKPOL bits.
Step [3] can be skipped if the values have not been changed from the initial values.

[4] Set the data transmission/reception format in the SMR, SCMR, and SEMR registers.

[5] Write a value corresponding to the bit rate to the BRR register. This step is not necessary if an external clock is used.

[6] Write the value obtained by correcting a bit rate error in the MDDR register. This step is not necessary if the SEMR.BRME bit is set to 0 or an external clock is used.

[7] Set the SCR.TE or RE bit to 1. At this time, also set the SCR.TIE and RIE bits.
Setting the TE and RE bits allows TXDn and RXDn to be used.

Note: In simultaneous transmit and receive operations, the SCR.TE and RE bits should both be set to 0 or 1 simultaneously.

Figure 33.24 Example of SCI Initialization Flowchart (Clock Synchronous Mode)

Downloaded from Arrow.com.

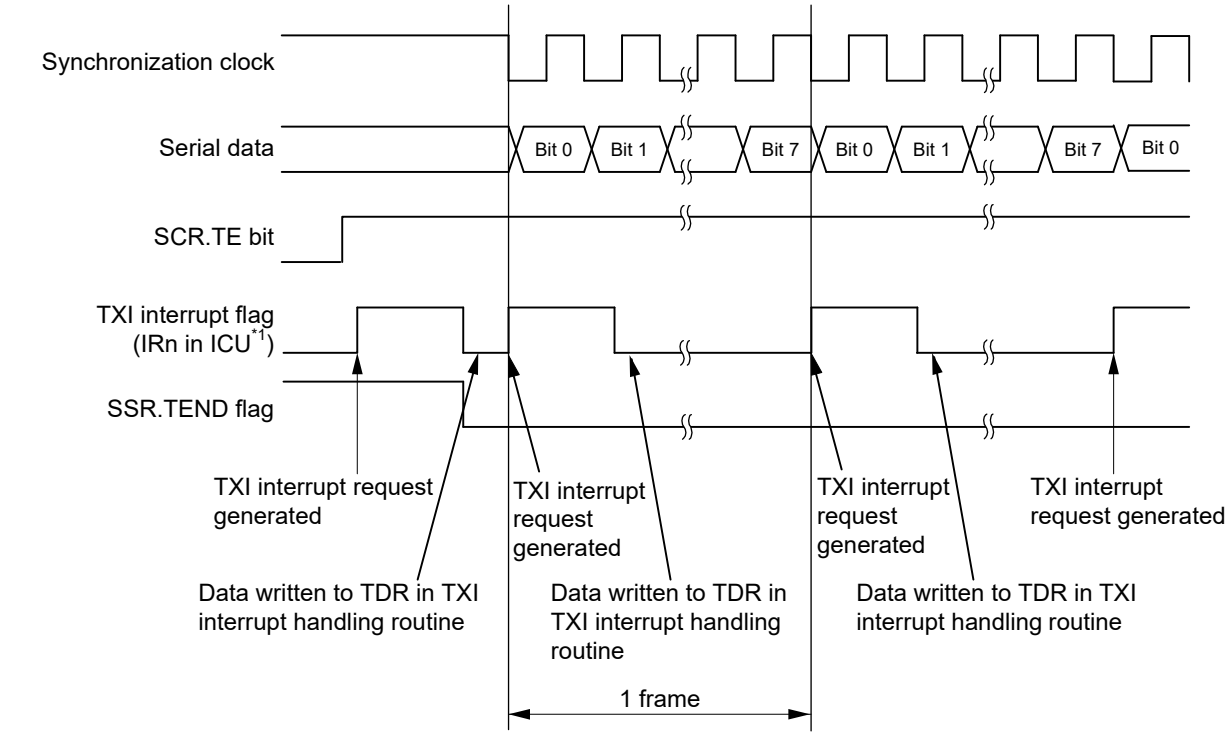
33.5.4 Serial Data Transmission (Clock Synchronous Mode)

Figure 33.25, Figure 33.26, and Figure 33.27 show an example of the operation for serial transmission in clock synchronous mode.
In serial data transmission, the SCI operates as described below.

1. The SCI transfers data from the TDR register to the TSR register when data is written to the TDR register in the TXI interrupt handling routine. The TXI interrupt request at the beginning of transmission is generated when the SCR.TE bit is set to 1 after the SCR.TIE bit is set to 1 or when these 2 bits are set to 1 simultaneously by a single instruction.
2. After transferring data from the TDR register to the TSR register, the SCI starts transmission. When the SCR.TIE bit is set to 1 at this time, a TXI interrupt request is generated. Continuous transmission is enabled by writing the next transmit data to the TDR register in this TXI interrupt handling routine before transmission of the current transmit data has finished. When TEI interrupt requests are in use, set the SCR.TIE bit to 0 (a TXI interrupt request is disabled) and the SCR.TEIE bit to 1 (a TEI interrupt request is enabled) after the last of the data to be transmitted are written to the TDR register from the handling routine for TXI requests.
3. 8-bit data is sent from the TXDn pin in synchronization with the output clock when clock output mode has been specified and in synchronization with the input clock when use of an external clock has been specified. Output of the clock signal is suspended until the input CTS signal is at the low level while the SPMR.CTSE bit is 1 (CTS function is enabled).
4. The SCI checks for updating of (writing to) the TDR register at the time of the last bit output.
5. When TDR is updated, the next transmit data is transferred from the TDR register to the TSR register, and serial transmission of the next frame is started.
6. If the TDR register is not updated, set the SSR.TEND flag to 1 and the TXDn pin retains the output state of the last bit. If the SCR.TEIE bit is 1 at this time, a TEI interrupt request is generated. The SCKn pin is held high.

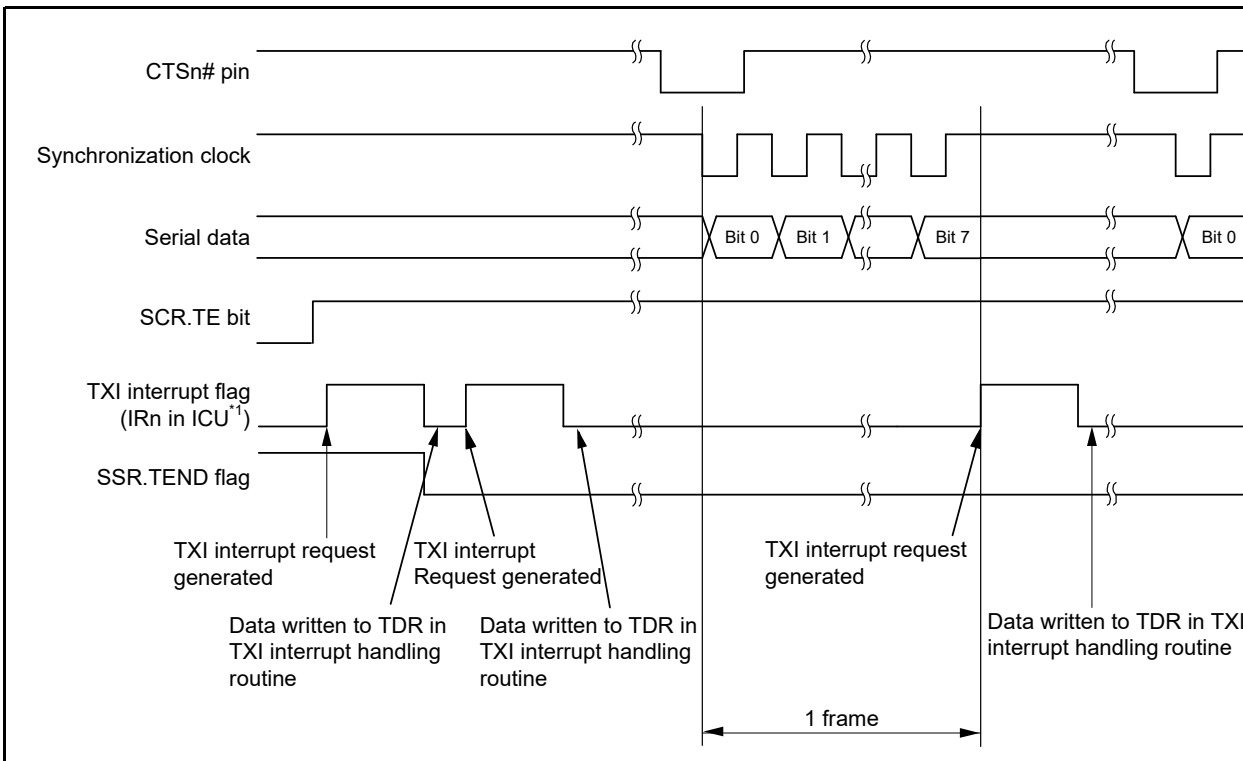
Figure 33.28 shows a sample flowchart of serial data transmission.
Transmission will not start while a receive error flag (ORER, FER, or PER in the SSR register) is set to 1. Be sure to set the receive error flags to 0 before starting transmission. Note that setting the SCR.RE bit to 0 does not clear the receive error flags.

Downloaded from Arrow.com.



Note 1. Refer to section 15, Interrupt Controller (ICUb) for details on the corresponding interrupt vector number.

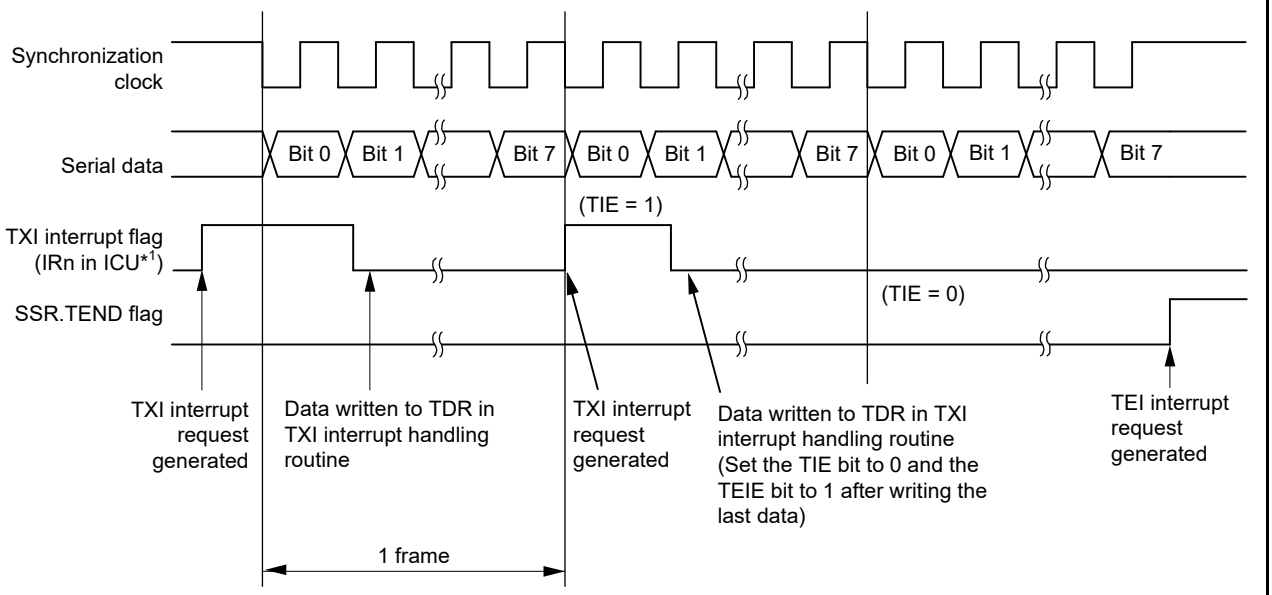
Figure 33.25 Example of Serial Data Transmission in Clock Synchronous Mode When the CTS Function is Not Used at the Beginning of Transmission



Note 1. Refer to section 15, Interrupt Controller (ICUb) for details on the corresponding interrupt vector number.

Figure 33.26 Example of Serial Data Transmission in Clock Synchronous Mode When the CTS Function is Used at the Beginning of Transmission

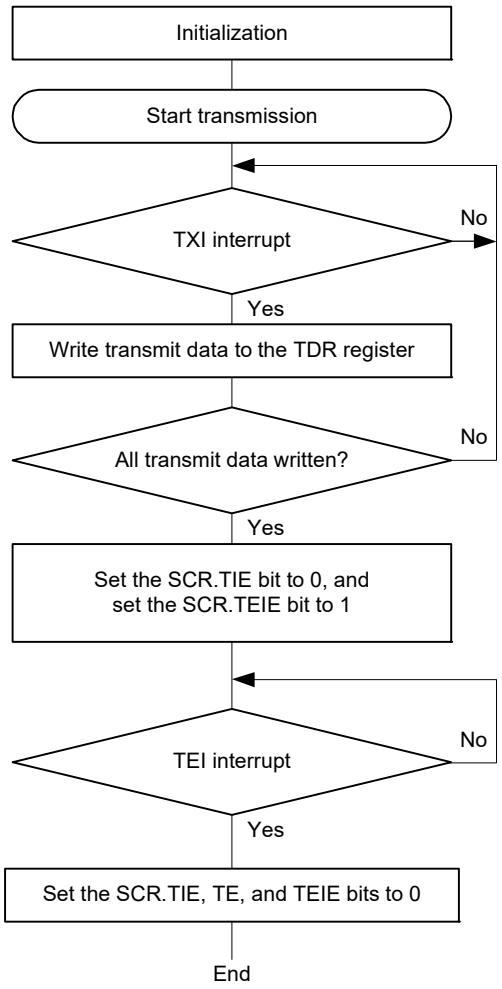
Downloaded from Arrow.com.



Note 1. Refer to section 15, Interrupt Controller (ICUb) for details on the corresponding interrupt vector number.

Figure 33.27 Example of Serial Data Transmission in Clock Synchronous Mode from the Middle of Transmission until Transmission Completion

Downloaded from Arrow.com.



[1] SCI initialization:
Set data transmission.

[2] Writing transmit data write to the TDR register by a TXI interrupt request:
When transmit data is transferred from the TDR register to the TSR register, a transmit data empty interrupt (TXI) request is generated.
Transmit data is written to the TDR register once from the handling routine for TXI requests.

[3] Serial transmission continuation procedure:
To continue serial transmission, write transmit data to the TDR register upon accepting a transmit data empty interrupt (TXI) request. Transmit data can also be written to the TDR register by activating the DMAC or DTC by the TXI interrupt request.
When TEI interrupt requests are in use, set the SCR.TIE bit to 0 and the SCR.TEIE bit to 1 after the last of the data to be transmitted are written to the TDR register.

Note: When the external clock is in use (the value of the SCR.CKE[1:0] bits is 10b or 11b), the rising edge on the SCK pin for the last bit sets the SSR.TEND flag to 1. Setting the SCR.TE bit to 0 immediately after this may lead to insufficient received-data hold time on the receiver side.

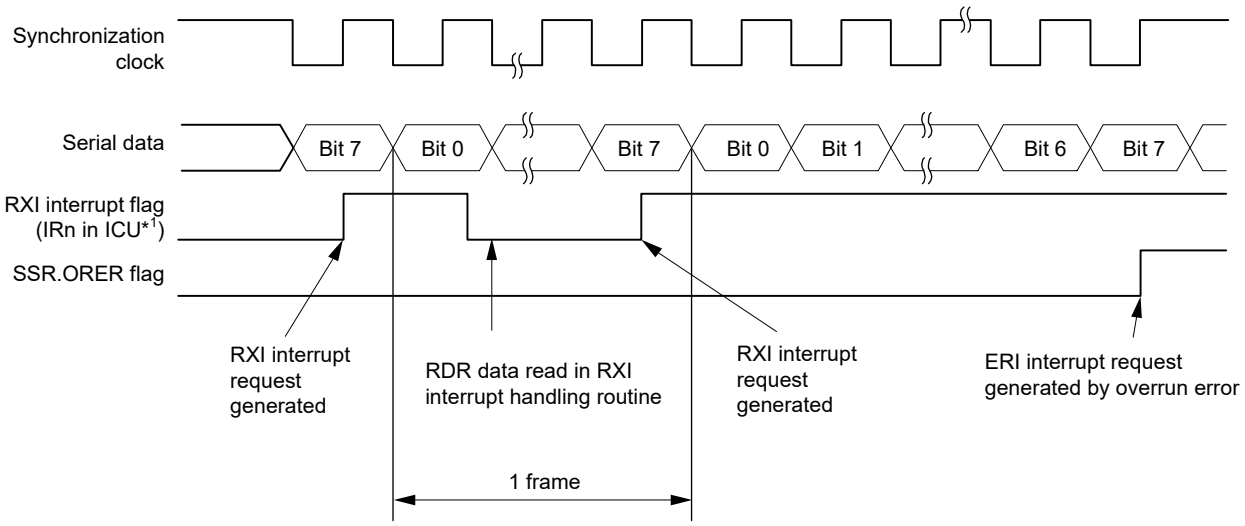
Figure 33.28 Example Flowchart of Serial Transmission in Clock Synchronous Mode

Downloaded from Arrow.com.

33.5.5 Serial Data Reception (Clock Synchronous Mode)

Figure 33.29 and Figure 33.30 show an example of SCI operation for serial reception in clock synchronous mode. In serial data reception, the SCI operates as described below.

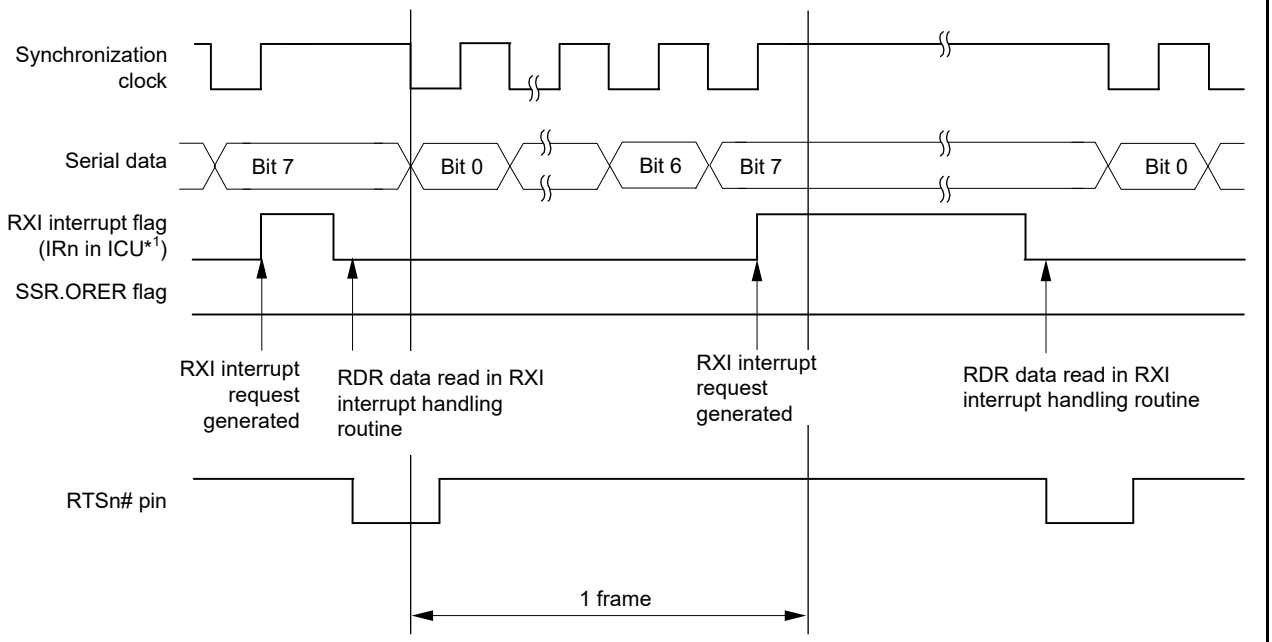
1. The value of the SCR.RE bit becoming 1 places the signal output on the RTSn# pin at the low level (when the RTS function is in use).
2. The SCI performs internal initialization and starts receiving data in synchronization with a synchronization clock input or output, and stores the receive data in the RSR register.
3. If an overrun error occurs, the SSR.ORER flag is set to 1. If the SCR.RIE bit is 1 at this time, an ERI interrupt request is generated. Receive data is not transferred to the RDR register.
4. When reception finishes successfully, receive data is transferred to the RDR register. If the SCR.RIE bit is 1 at this time, an RXI interrupt request is generated. Continuous reception is enabled by reading the receive data transferred to the RDR register in this RXI interrupt handling routine before reception of the next receive data is completed. Reading out the received data that have been transferred to the RDR register causes the RTSn# pin to output the low level (when the RTS function is in use).



Note 1. Refer to section 15, Interrupt Controller (ICUb) for details on the corresponding interrupt vector number.

Figure 33.29 Example of Operation for Serial Reception in Clock Synchronous Mode (1) (When RTS Function is Not Used)

Downloaded from Arrow.com.



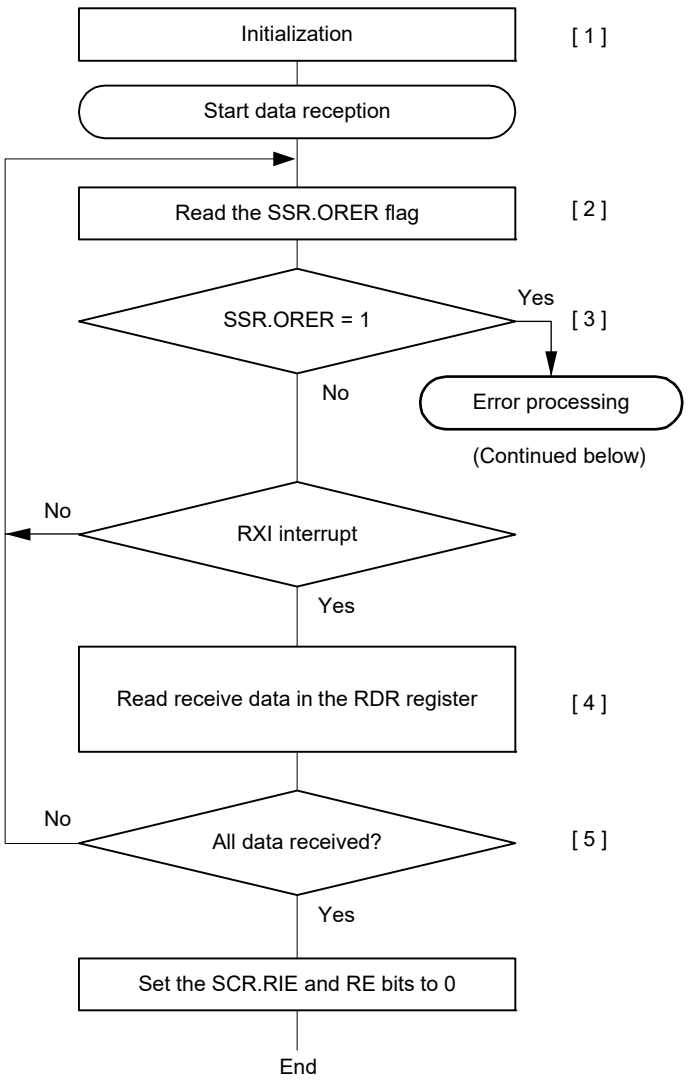
Note 1. Refer to section 15, Interrupt Controller (ICUb) for details on the corresponding interrupt vector number.

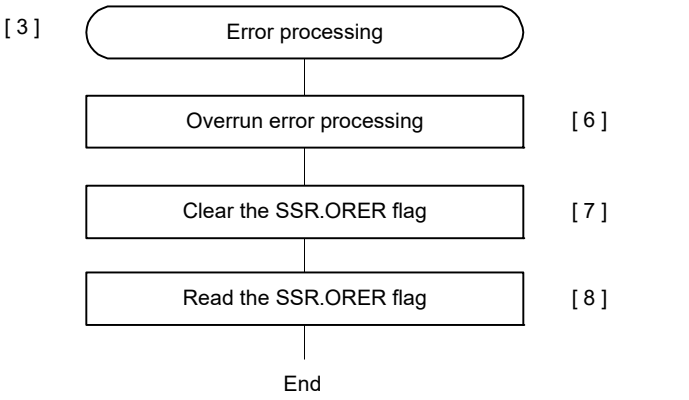
Figure 33.30 Example of Operation for Serial Reception in Clock Synchronous Mode (2) (When RTS Function is Used)

Data transfer cannot be resumed while a receive error flag is 1. Accordingly, clear the ORER, FER, and PER flags in the SSR register before resuming reception. Moreover, be sure to read the RDR register during overrun error processing. When a reception is forcibly terminated by setting the SCR.RE bit to 0 during operation, read the RDR register because received data which has not yet been read may be left in the RDR register.

Downloaded from Arrow.com.

Figure 33.31 shows a sample flowchart for serial data reception.





[1] SCI initialization:
Make input port-pin settings for pins to be used as RXDn pins.

[2] [3] Receive error processing:
If a receive error occurs, read the SSR.ORER flag, perform the relevant error processing, and then set the ORER flag to 0. Data reception cannot be resumed while the ORER flag is 1.

[4] Read the receive data in the RDR register once in the receive data full interrupt (RXI) request handling routine.

[5] Serial reception continuation procedure:
To continue serial reception, before the MSB (bit 7) of the current frame is received, finish reading the receive data in the RDR register. The RDR data can also be read by activating the DMAC or DTC by an RXI interrupt request.

[6] Processing in response to an overrun error:
Read the RDR register. In combination with step [7], this will make correct reception of the next frame possible.

[7] Clearing the error flag:
Write 0 to the error flag.

[8] Confirming that the error flag is cleared:
Read the error flag to confirm that its value is 0.

Figure 33.31 Example Flowchart of Serial Reception in Clock Synchronous Mode

Downloaded from Arrow.com.

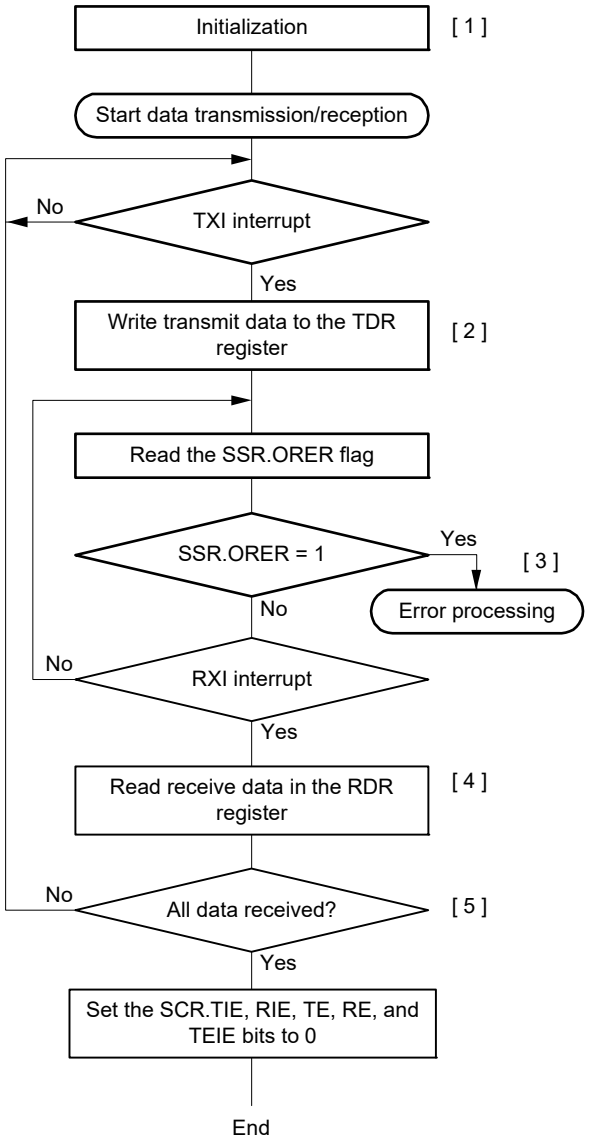
33.5.6 Simultaneous Serial Data Transmission and Reception (Clock Synchronous Mode)

Figure 33.32 shows a sample flowchart for simultaneous serial transmit and receive operations in clock synchronous mode.

After initializing the SCI, the following procedure should be used for simultaneous serial data transmit and receive operations.

To switch from transmit mode to simultaneous transmit and receive mode, check that the SCI has finished transmission by reading that the SSR.TEND flag is 1, and then initialize the SCR register. Then set the TIE, RIE, TE, and RE bits in the SCR register to 1 simultaneously by a single instruction.

To switch from receive mode to simultaneous transmit and receive mode, check that the SCI has finished reception, and then set the RIE and RE bits to 0. Then check that the receive error flags (ORER, FER, and PER in the SSR register) are 0, and then set the TIE, RIE, TE, and RE bits in the SCR register to 1 simultaneously by a single instruction.



[1] SCI initialization:
The TXDn pin can act as the output pin for transmitted data and the RXDn pin can act as the input pin for received data at the same time.

[2] Transmit data write:
Write transmit data to the TDR register once in the TXI interrupt handling routine.

[3] Receive error processing:
If a receive error occurs, read the SSR.ORER flag, perform the relevant error processing, and then set the ORER flag to 0. Data reception cannot be resumed while the ORER flag is 1.

[4] Reading receive data:
Read the receive data in the RDR register once in the RXI interrupt request handling routine.

[5] Serial transmission/reception continuation procedure:
To continue serial transmission and reception, before the MSB (bit 7) of the current frame is received, finish reading the receive data in the RDR register by the RXI interrupt. Also, before the MSB (bit 7) of the current frame is transmitted, write data to the TDR register by the TXI interrupt. Transmit data can also be written to the TDR register by activating the DMAC or DTC by a transmit data empty interrupt (TXI) request. Similarly, the RDR data can also be read by activating the DMAC or DTC by a receive data full interrupt (RXI) request.

Note: When switching from transmit or receive operation to simultaneous transmit and receive operations first set the TIE, RIE, TE, RE, and TEIE bits in the SCR register to 0, and then set TIE, RIE, TE, and RE bits to 1 simultaneously.

Figure 33.32 Example Flowchart of Simultaneous Serial Transmission and Reception in Clock Synchronous Mode

Downloaded from Arrow.com.

33.6 Operation in Smart Card Interface Mode

The SCI supports smart card (IC card) interfaces conforming to ISO/IEC 7816-3 (standard for Identification Cards), as an extended function of the SCI.
Smart card interface mode can be selected using the appropriate register.

33.6.1 Sample Connection

Figure 33.33 shows a sample connection between a smart card (IC card) and this MCU.
As in the figure, since this MCU communicates with an IC card using a single transmission line, interconnect the TXDn and RXDn pins and pull up the data transmission line to VCC using a resistor.
Setting the TE and RE bits in the SCR register to 1 with an IC card disconnected enables closed transmission/reception allowing self-diagnosis.
To supply an IC card with the clock pulses generated by the SCI, input the SCKn pin output to the CLK pin of an IC card.
The output port of the this MCU can be used to output a reset signal.

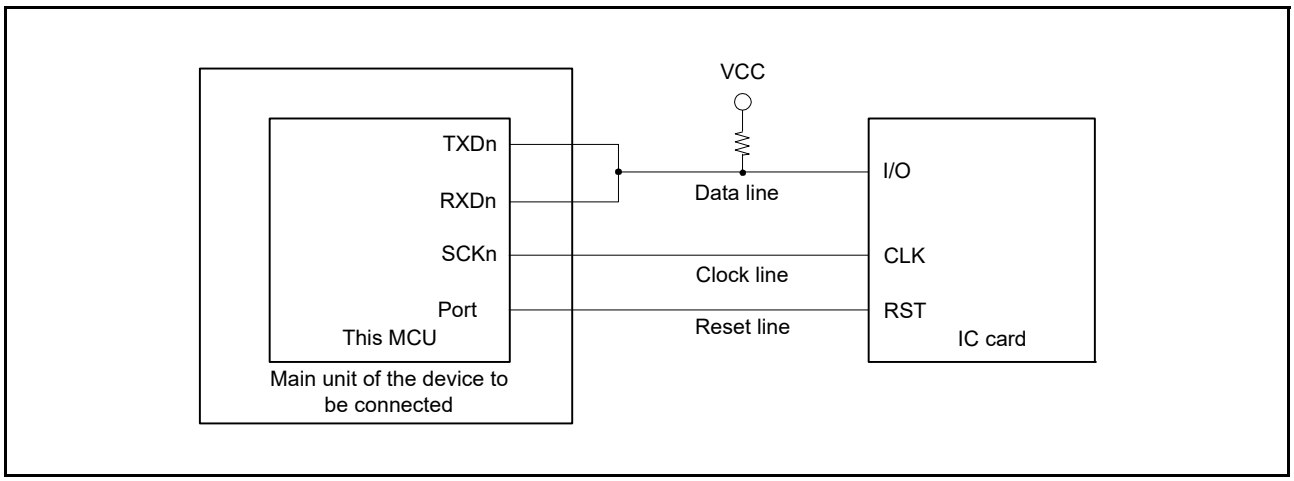


Figure 33.33 Sample Connection with a Smart Card (IC Card)

Downloaded from Arrow.com.

33.6.2 Data Format (Except in Block Transfer Mode)

Figure 33.34 shows the data transfer formats in smart card interface mode.

- One frame consists of 8-bit data and a parity bit in asynchronous mode.
- During transmission, at least 2 etu (elementary time unit: time required for transferring 1 bit) is secured as a guard time from the end of the parity bit until the start of the next frame.
- If a parity error is detected during reception, a low-level error signal is output for 1 etu after 10.5 etu has passed from the start bit.
- If an error signal is sampled during transmission, the same data is automatically retransmitted after at least 2 etu.

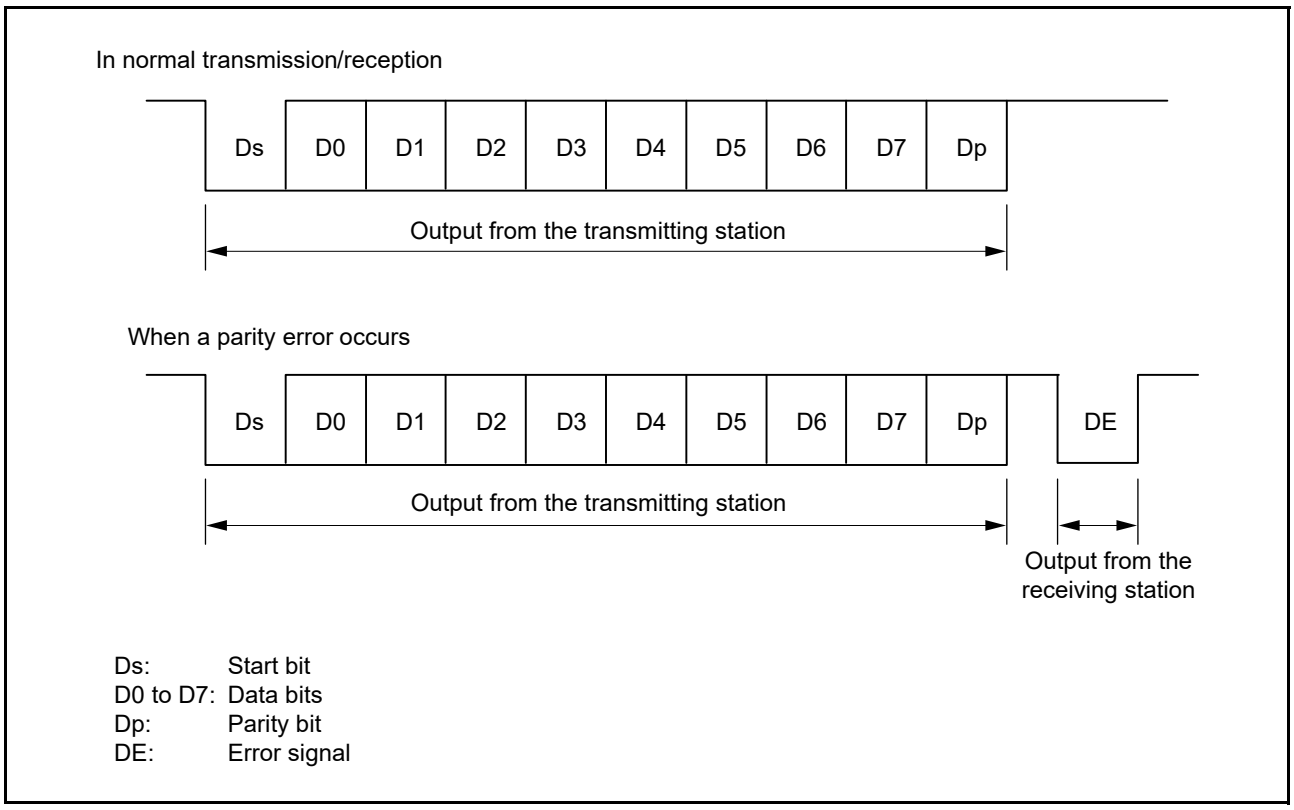


Figure 33.34 Data Formats in Smart Card Interface Mode

Downloaded from Arrow.com.

For communications with IC cards of the direct convention type and inverse convention type, follow the procedure below.

(1) Direct Convention Type

For the direct convention type, logic levels 1 and 0 correspond to states Z and A, respectively, and data is transferred with LSB first as the start character, as shown in Figure 33.35. Therefore, data in the start character in the figure is 3Bh. When using the direct convention type, write 0 to both the SDIR and SINV bits in the SCMR register. Write 0 to the SMR.PM bit in order to use even parity, which is prescribed by the smart card standard.

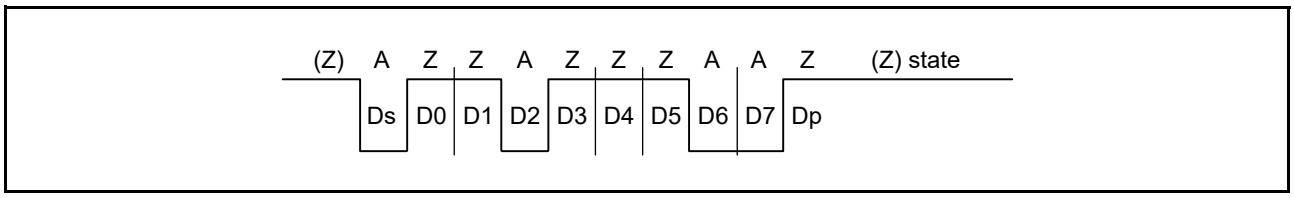


Figure 33.35 Direct Convention (SCMR.SDIR bit = 0, SCMR.SINV bit = 0, SMR.PM bit = 0)

(2) Inverse Convention Type

For the inverse convention type, logic levels 1 and 0 correspond to states A and Z, respectively and data is transferred with MSB first as the start character, as shown in Figure 33.36. Therefore, data in the start character in the figure is 3Fh. When using the inverse convention type, write 1 to both the SDIR and SINV bits in the SCMR register. The parity bit is logic level 0 to produce even parity, which is prescribed by the smart card standard, and corresponds to state Z. Since the SINV bit of the this MCU only inverts data bits D7 to D0, write 1 to the SMR.PM bit to invert the parity bit for both transmission and reception.

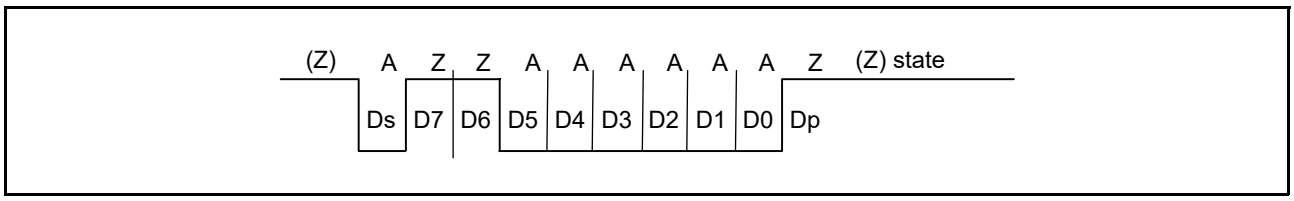


Figure 33.36 Inverse Convention (SCMR.SDIR bit = 1, SCMR.SINV bit = 1, SMR.PM bit = 1)

33.6.3 Block Transfer Mode

Block transfer mode is different from non-block transfer mode in the following respects.

- Even if a parity error is detected during reception, no error signal is output. Since the SSR.PER flag is set by error detection, clear the PER flag before receiving the parity bit of the next frame.
- During transmission, at least 1 etu is secured as a guard time from the end of the parity bit until the start of the next frame.
- Since the same data is not retransmitted during transmission, the SSR.TEND flag is set 11.5 etu after transmission start.
- In block transfer mode, the SSR.ERS flag indicates the error signal status as in non-block transfer mode, but the flag is read as 0 because no error signal is transferred.

Downloaded from Arrow.com.

33.6.4 Receive Data Sampling Timing and Reception Margin

Only the base clock generated by the on-chip baud rate generator can be used as a transmit/receive clock in smart card interface mode.

In this mode, the SCI can operate on a base clock with a frequency of 32, 64, 372, 256, 93, 128, 186, or 512 times the bit rate according to the settings of the SCMR.BCP2 bit and the SMR.BCP[1:0] bits.

For data reception, the falling edge of the start bit is sampled with the base clock to perform synchronization. Receive data is sampled on the 16th, 32nd, 186th, 128th, 46th, 64th, 93rd, and 256th rising edges of the base clock so that it can be latched at the middle of each bit as shown in Figure 33.37. The reception margin here is determined by the following formula.

$$M = \left| \left(0.5 - \frac{1}{2N}\right) - (L - 0.5)F - \frac{|D - 0.5|}{N}(1 + F) \right| \times 100\ (\%)$$

M: Reception margin (%)
N: Ratio of bit rate to clock (N = 32, 64, 372, 256)
D: Duty cycle of clock (D = 0 to 1.0)
L: Frame length (L = 10)
F: Absolute value of clock frequency deviation

Assuming values of F = 0, D = 0.5, and N = 372 in the above formula, the reception margin is determined by the formula below.

M = {0.5 – 1/(2 × 372)} × 100 (%) = 49.866 (%)

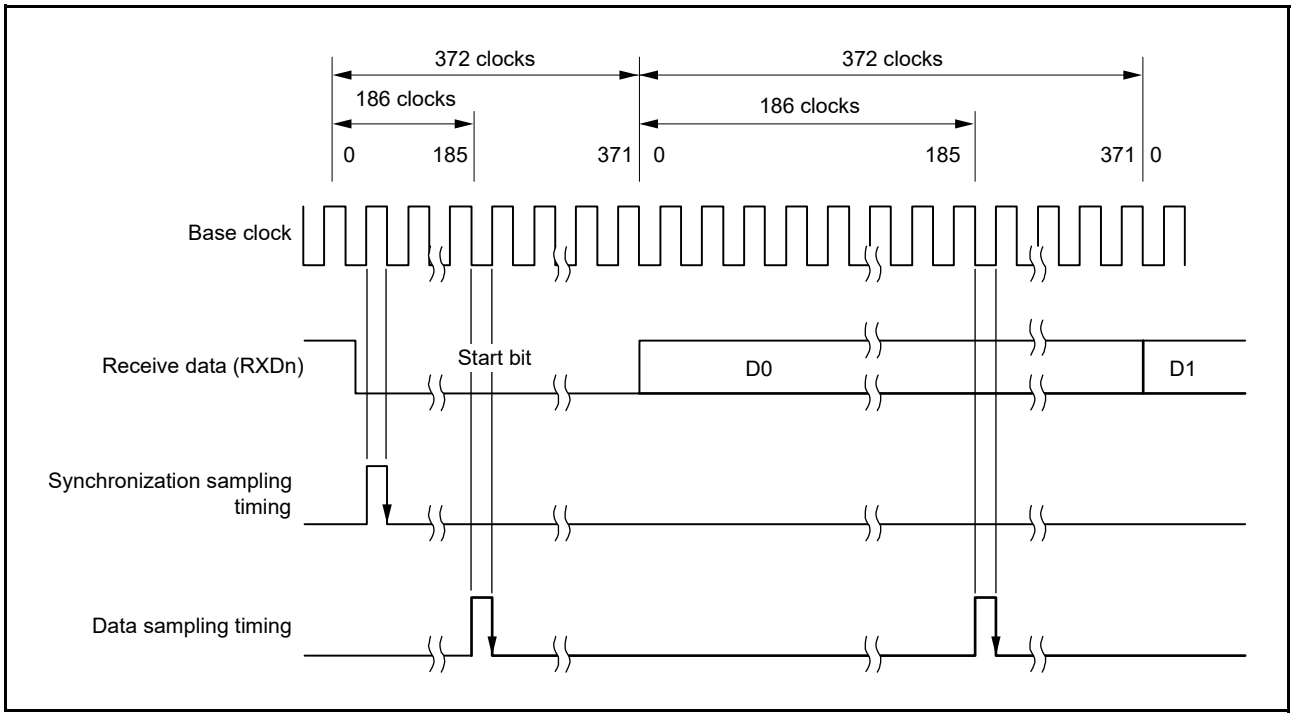


Figure 33.37 Receive Data Sampling Timing in Smart Card Interface Mode (When Clock Frequency is 372 Times the Bit Rate)

Downloaded from Arrow.com.

33.6.5 SCI Initialization (Smart Card Interface Mode)

Initialize the SCI following the example of flowchart shown in Figure 33.38.
Initialize the SCR and SSR registers before switching from transmit mode to receive mode and vice versa. When not changing the bit rate, it is not necessary to set the CKE[1:0] bits to 00b. Even if the RE bit is set to 0, the RDR register is not initialized.

To change receive mode to transmit mode, first check that reception has completed, and then execute steps [1] and [3] in Figure 33.38. Set TE = 1 and RE = 0 at step [11]. Reception completion can be verified by reading the RXI request, SSR.ORER, or SSR.PER flag.
To change transmit mode to receive mode, first check that transmission has completed, and then execute steps [1] and [3] in Figure 33.38. Set TE = 0 and RE = 1 at step [11]. Transmission completion can be verified by reading the SSR.TEND flag.

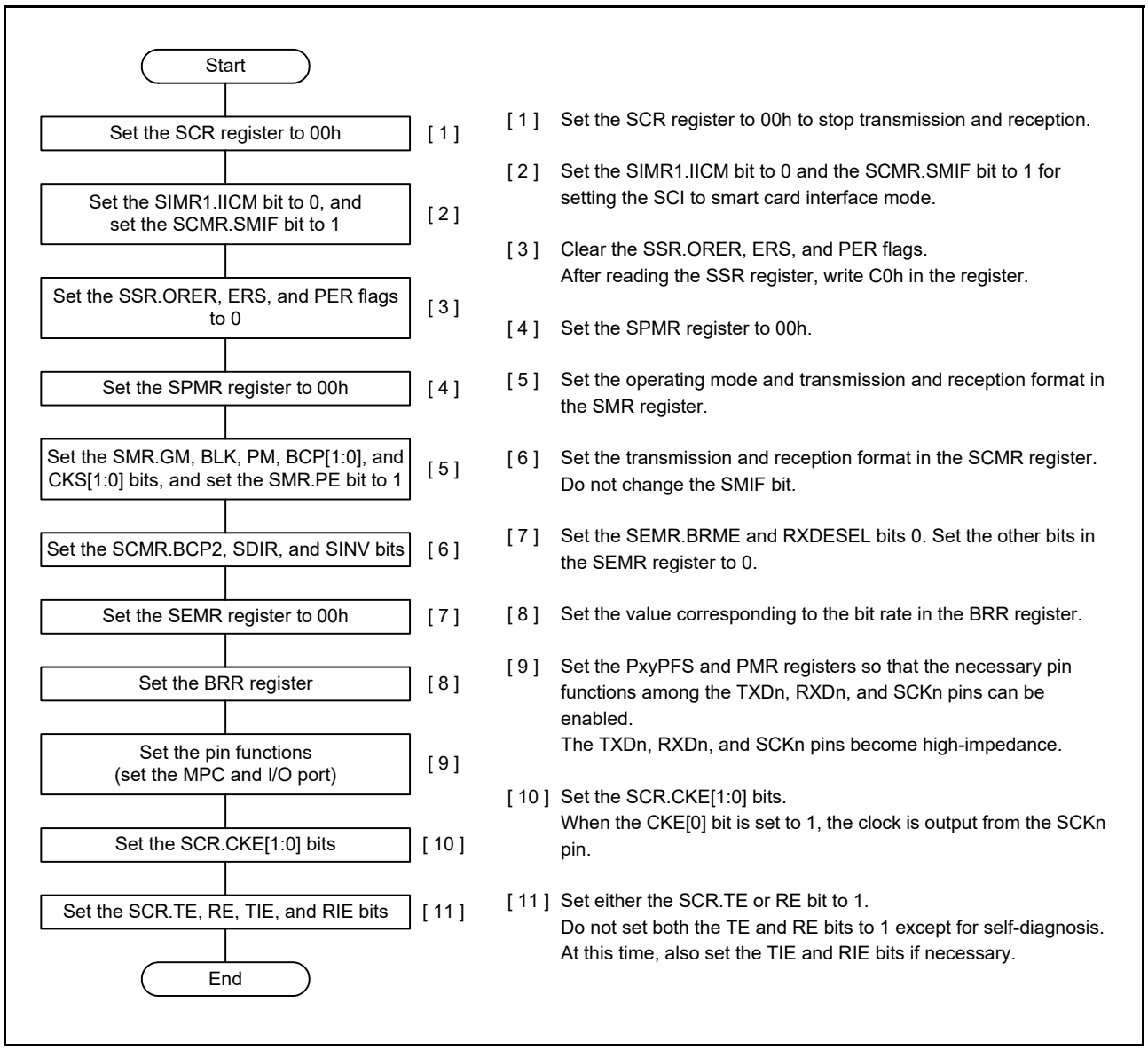


Figure 33.38 Example of SCI Initialization Flowchart (Smart Card Interface Mode)

Downloaded from Arrow.com.

Figure 33.39 shows an example of data transmission when the SCI is set to smart card interface mode according to the flow described in Figure 33.38 after a reset. When the pin functions are set to the SCK and TXD pins, they are still high-impedance because the SCR.CKE[0] and SCR.TE bits are 0. When the CKE[0] bit is set to 1, clock is output from the SCK pin. When the transmit data is written after setting the TE bit to 1, a data transmission starts. After the TE bit is set to 1, one frame of high-impedance is output from TXD pin (internal wait time) and then the data transmission starts. In smart card interface mode, the clock is continuously output while the CKE[0] bit is set to 1 (clock output) even if both the TE and RE bits are set to 0.

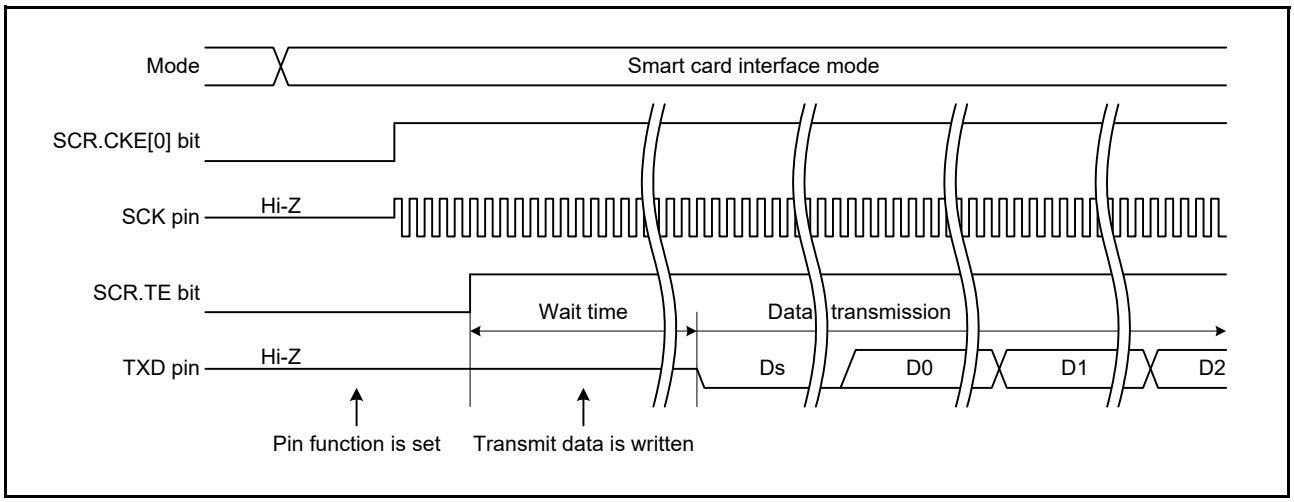


Figure 33.39 Example of Data Transmission Timing in Smart Card Interface Mode

Downloaded from Arrow.com.

33.6.6 Serial Data Transmission (Except in Block Transfer Mode)

Serial data transmission in smart card interface mode (except in block transfer mode), in that an error signal is sampled and data can be retransmitted, is different from that in non-smart card interface mode. Figure 33.40 shows the data retransmit operation during transmission.

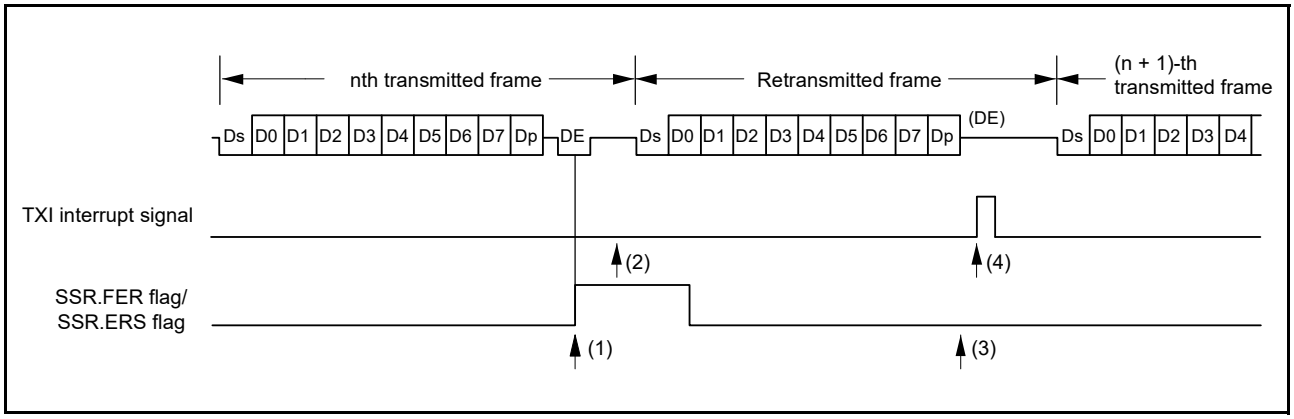


Figure 33.40 Data Retransmit Operation in SCI Transmit Mode

(1) When an error signal from the receiver end is sampled after one-frame data has been transmitted, the SSR.ERS flag is set to 1. If the SCR.RIE bit is 1 at this time, an ERI interrupt request is generated. Clear the ERS flag before the next parity bit is sampled.

(2) For a frame in which an error signal is received, the SSR.TEND flag is not set. Data is retransferred from the TDR register to the TSR register allowing automatic data retransmission.

(3) If no error signal is returned from the receiver, the ERS flag is not set to 1.

(4) In this case, the SCI judges that transmission of one-frame data (including retransmission) has been completed, and the TEND flag is set. If the SCR.TIE bit is 1 at this time, a TXI interrupt request is generated. Writing transmit data to the TDR register starts transmission of the next data.

Downloaded from Arrow.com.

Figure 33.41 shows a sample flowchart of serial transmission.

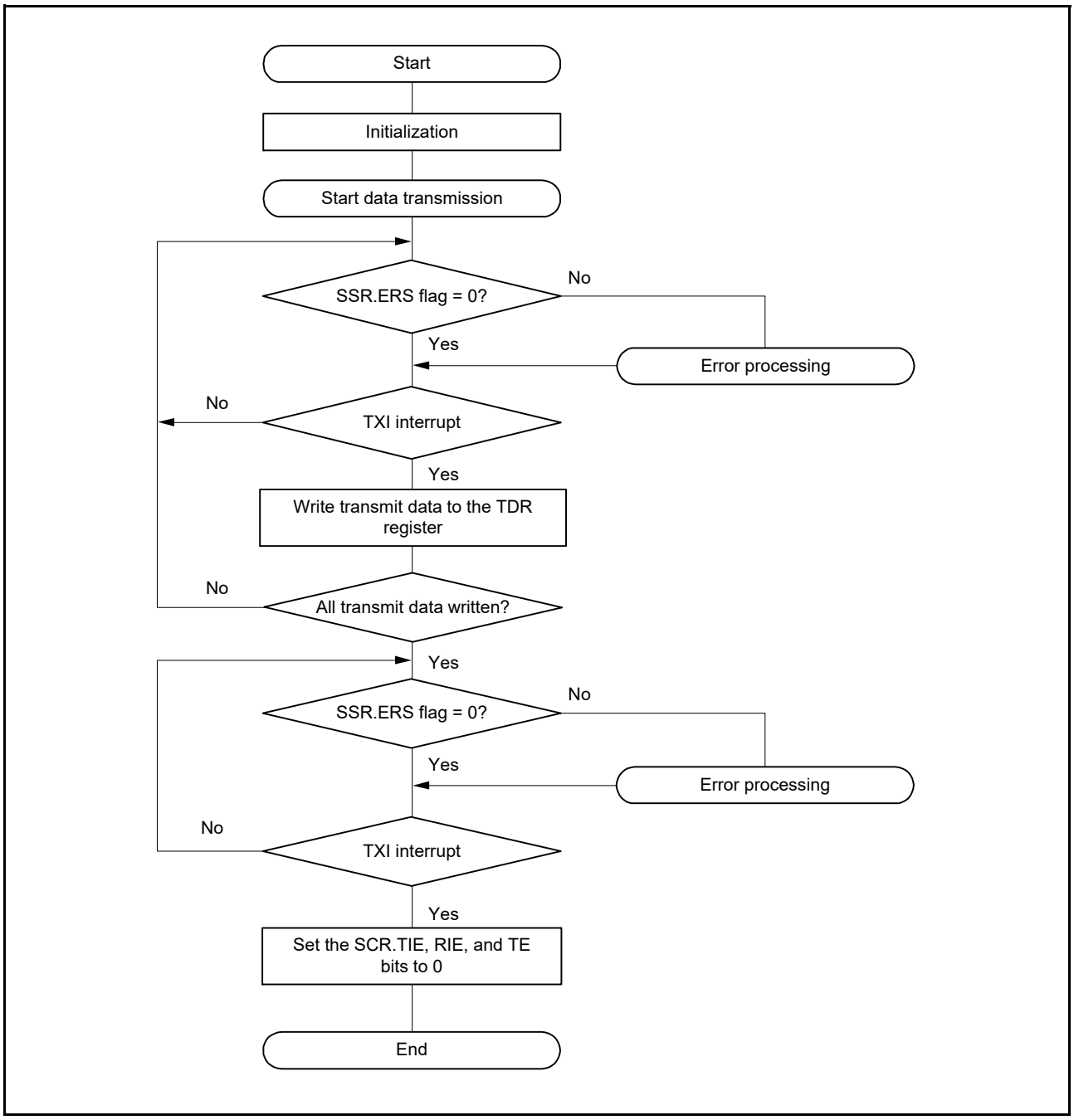


Figure 33.41 Sample Smart Card Interface Transmission Flowchart

All the processing steps are automatically performed using a TXI interrupt request to activate the DTC or DMAC. When the SSR.TEND flag is set to 1 in transmission, if the SCR.TIE bit is 1, a TXI interrupt request is generated. The DTC or DMAC is activated by a TXI interrupt request if the TXI interrupt request is specified as a source of DTC or DMAC activation beforehand, allowing transfer of transmit data. The TEND flag is automatically set to 0 when the DTC or DMAC transfers the data.

If an error occurs, the SCI automatically retransmits the same data. During this retransmission, the TEND flag is kept to 0 and the DTC or DMAC is not activated. Therefore, the SCI and DTC or DMAC automatically transmit the specified number of bytes, including retransmission in the case of error occurrence. However, since the ERS flag is not automatically cleared, set the RIE bit to 1 beforehand to enable an ERI interrupt request to be generated at error occurrence, and clear the ERS flag.

Downloaded from Arrow.com.

When transmitting/receiving data using the DTC or DMAC, be sure to make settings to enable the DTC or DMAC before making SCI settings.
For DTC or DMAC settings, refer to section 19, Data Transfer Controller (DTCa), section 18, DMA Controller (DMACA).

Note that the SSR.TEND flag is set in different timings depending on the SMR.GM bit setting. Figure 33.42 shows the TEND flag generation timing.

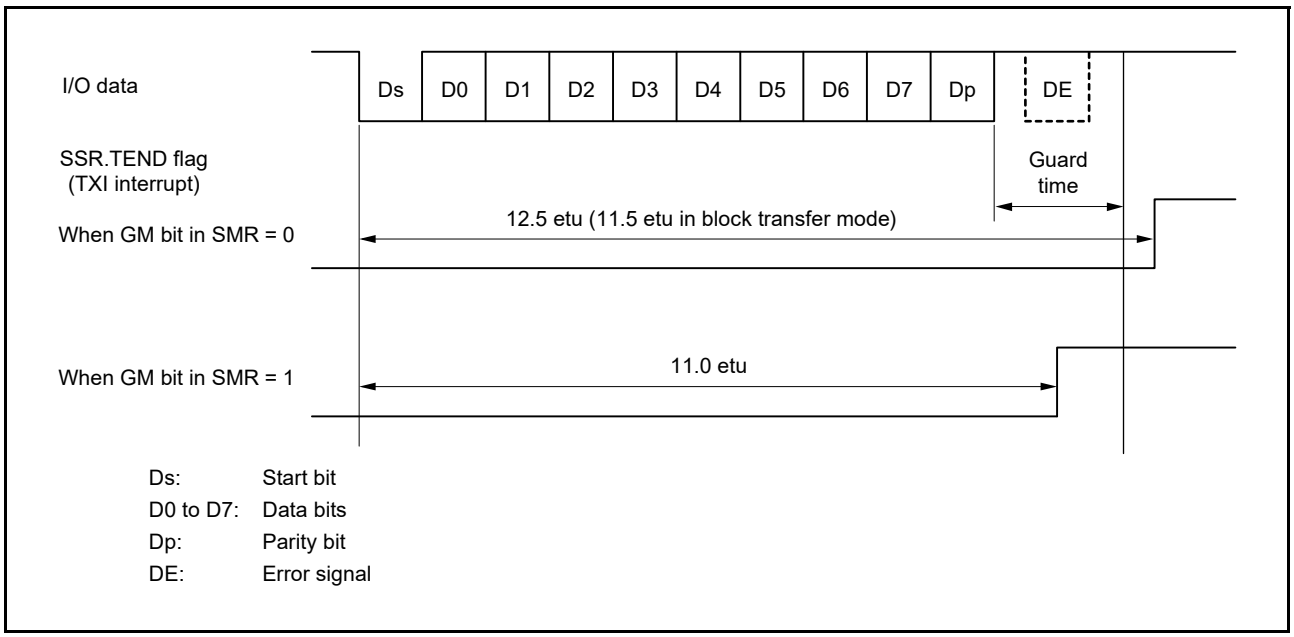


Figure 33.42 SSR.TEND Flag Generation Timing during Transmission

Downloaded from Arrow.com.

33.6.7 Serial Data Reception (Except in Block Transfer Mode)

Serial data reception in smart card interface mode is similar to that in non-smart card interface mode. Figure 33.43 shows the data retransmit operation in receive mode.

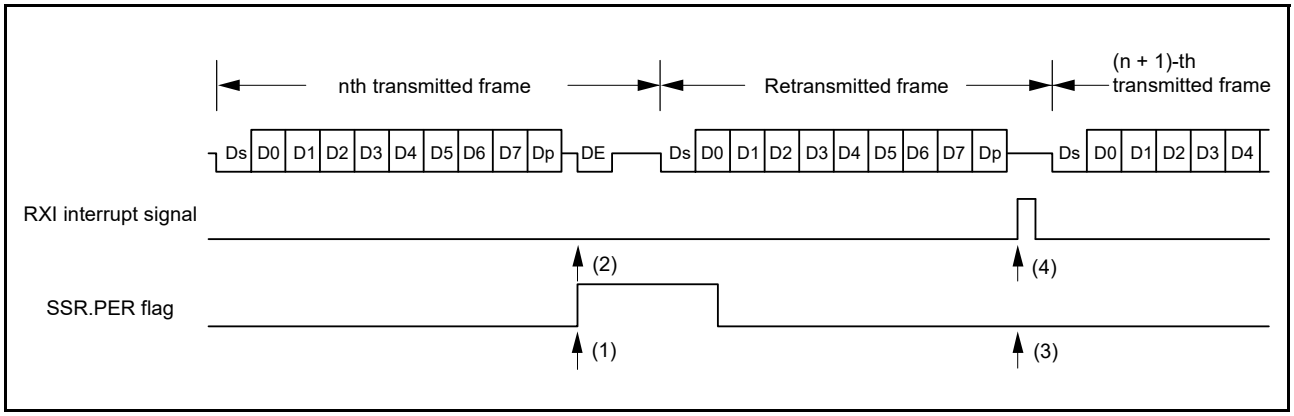


Figure 33.43 Data Retransmit Operation in SCI Receive Mode (Data Retransmit Operation during Reception)

(1) If a parity error is detected in receive data, the SSR.PER flag is set to 1. When the SCR.RIE bit is 1 at this time, an ERI interrupt request is generated. Clear the PER flag before the next parity bit is sampled.
(2) For a frame in which a parity error is detected, no RXI interrupt is generated.
(3) When no parity error is detected, the SSR.PER flag is not set to 1.
(4) In this case, data is determined to have been received successfully. When the SCR.RIE bit is 1, an RXI interrupt request is generated.

Downloaded from Arrow.com.

Figure 33.44 shows a sample flowchart for serial data reception.

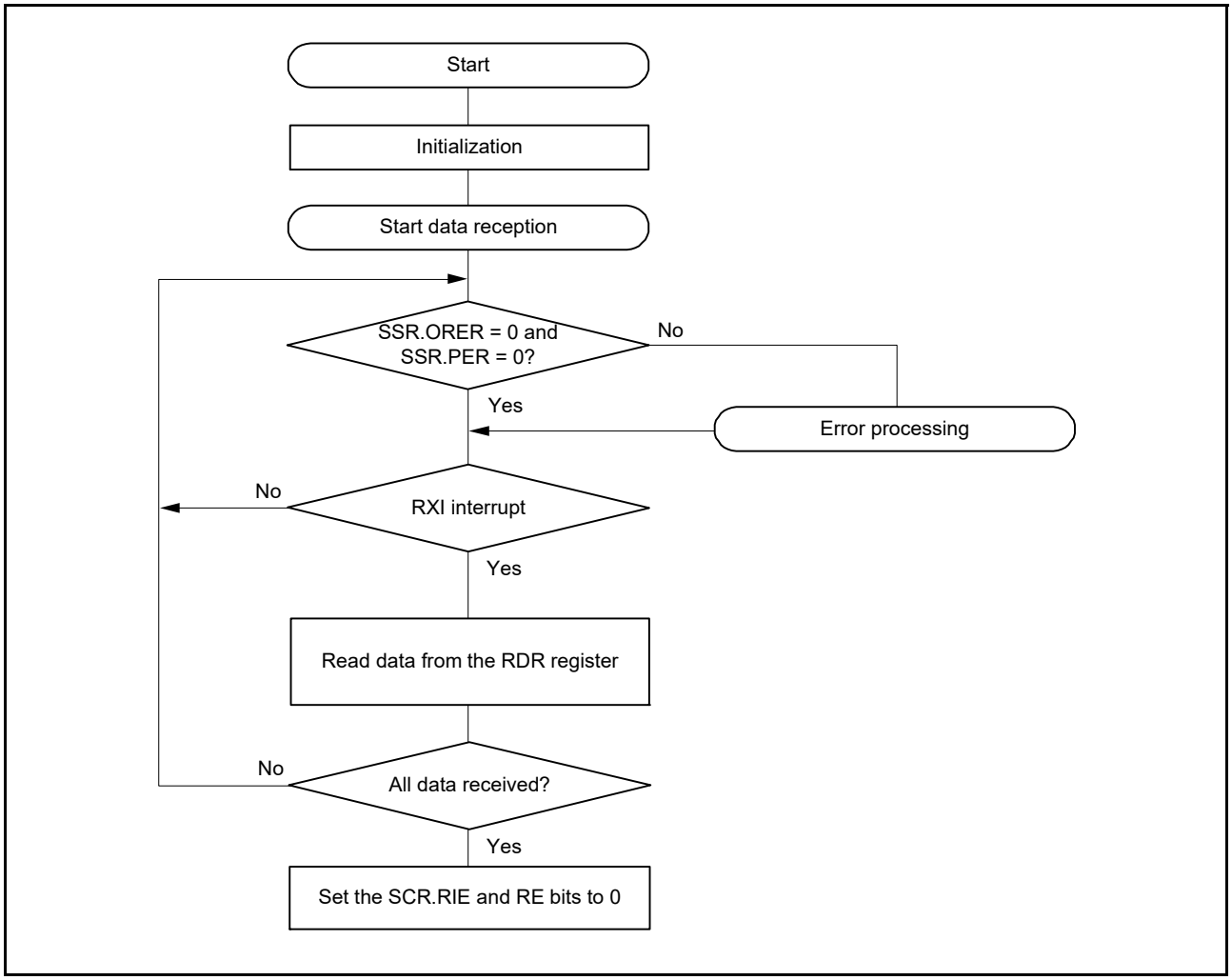


Figure 33.44 Sample Smart Card Interface Reception Flowchart

All the processing steps are automatically performed using an RXI interrupt request to activate the DTC or DMAC.
In reception, setting the RIE bit to 1 allows an RXI interrupt request to be generated. The DTC or DMAC is activated by an RXI interrupt request if the RXI interrupt request is specified as a source of DTC or DMAC activation beforehand, allowing transfer of receive data.
If an error occurs during reception and either the ORER or PER flag in the SSR register is set to 1, a receive error interrupt (ERI) request is generated. Clear the error flag after the error occurrence. If an error occurs, the DTC or DMAC is not activated and receive data is skipped. Therefore, the number of bytes of receive data specified in the DTC or DMAC is transferred.
Even if a parity error occurs and the PER flag is set to 1 during reception, receive data is transferred to the RDR register, thus allowing the data to be read.
When a reception is forcibly terminated by setting the SCR.RE bit to 0 during operation, read the RDR register because the received data which has not yet been read may be left in the RDR register.

Note 1. For operations in block transfer mode, refer to section 33.3, Operation in Asynchronous Mode.

Downloaded from Arrow.com.

33.6.8 Clock Output Control

Clock output can be fixed to high or low using the SCR.CKE[1:0] bits when the SMR.GM bit is 1. When the CKE[1:0] bits are set to 01b (clock output), the base clock is output from the SCK pin. For the settings of the base clock frequency (bit rate), refer to section 33.2.11, Bit Rate Register (BRR). When the CKE[1:0] bits are set to 00b (output fixed low) or 10b (output fixed to high), the SCK pin can be fixed to low or high.

Figure 33.45 shows a timing chart when the clock output is controlled.

If changing the CKE[1:0] bits while the SMR.GM bit is 0 (non-GSM mode), a pulse of unexpected width may output from SCK pin because the result is immediately reflected to the SCK pin.

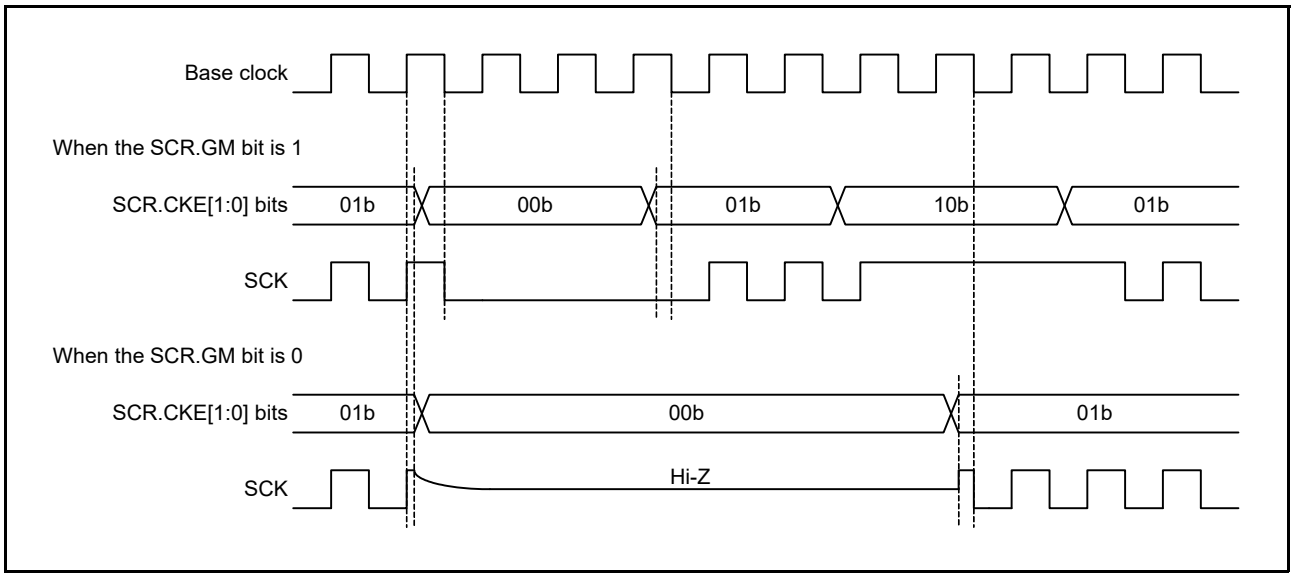


Figure 33.45 Clock Output Control

Downloaded from Arrow.com.

33.7 Operation in Simple I2C Mode

Simple I2C-bus format is composed of 8 data bits and an acknowledge bit. By continuing into a slave-address frame after a start condition or restart condition, a master device is able to specify a slave device as the partner for communications. The currently specified slave device remains valid until a new slave device is specified or a stop condition is satisfied.
The 8 data bits in all frames are transmitted in order from the MSB.
The I2C-bus format and timing of the I2C-bus are shown in Figure 33.46 and Figure 33.47.

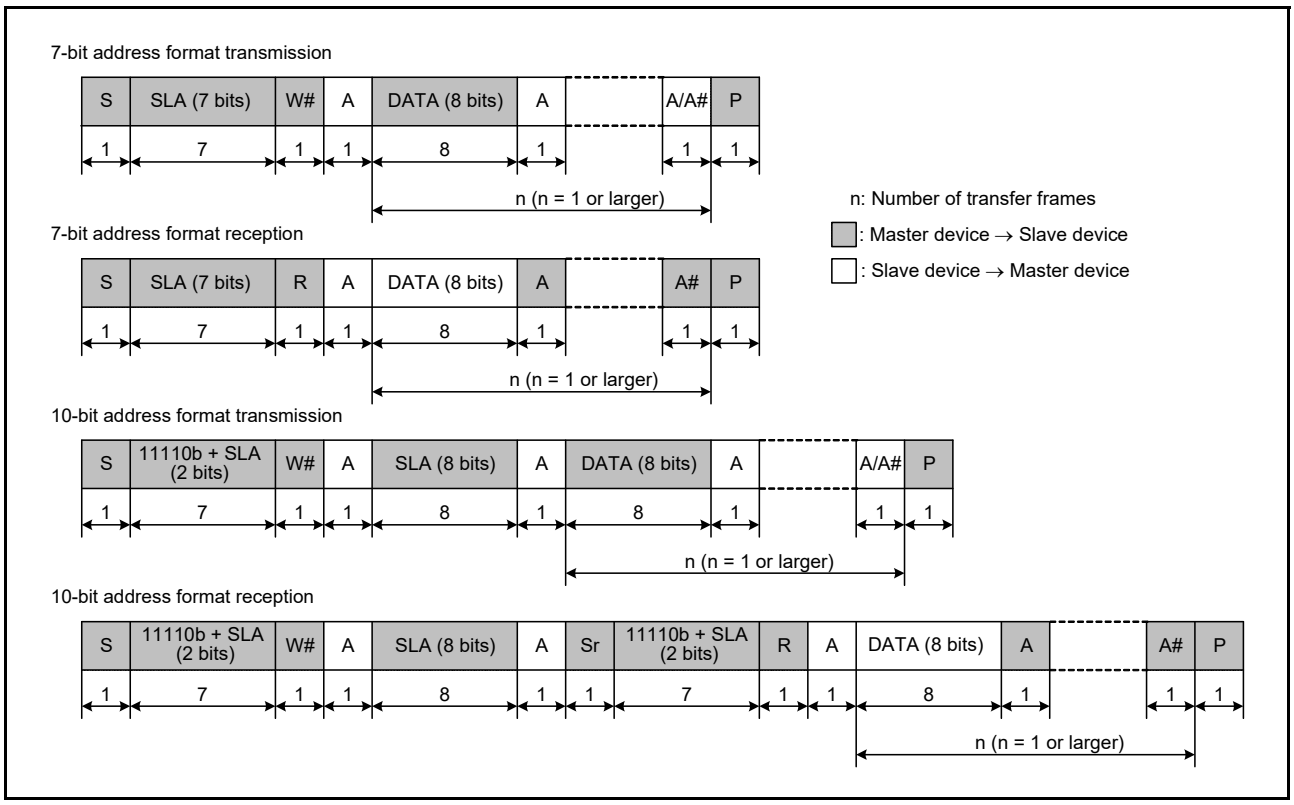


Figure 33.46 I2C-bus Format

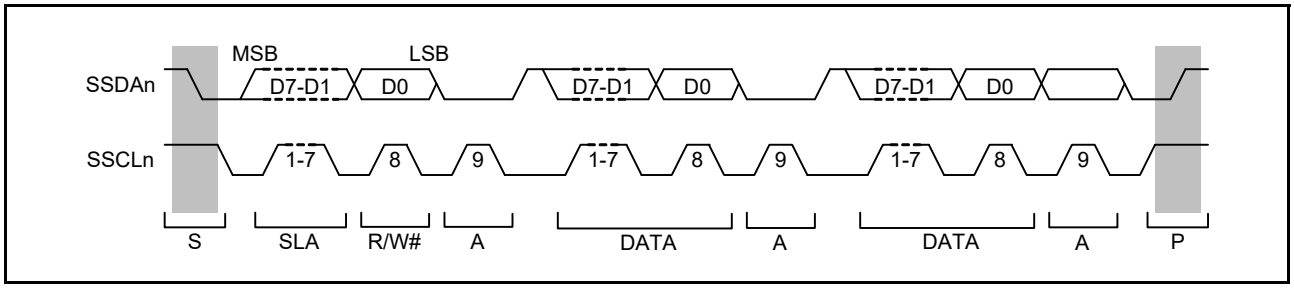


Figure 33.47 I2C-bus Timing (When SLA is 7 Bits)

S: Indicates a start condition, i.e. the master device changing the level on the SSDAn line from the high to the low level while the SSCLn line is at the high level.
SLA: Indicates a slave address, by which the master device selects a slave device.
R/W#: Indicates the direction of transfer (reception or transmission). The value 1 corresponds to transfer from the slave device to the master device and 0 corresponds to transfer from the master device to the slave device.
A/A#: Indicates an acknowledge bit. This is returned by the slave device for master transmission and by the master device for master reception. Return of the low level indicates ACK and return of the high level indicates NACK.
Sr: Indicates a restart condition, i.e. the master device changing the level on the SSDAn line from the high to the low level while the SSCLn line is at the high level and after the setup time has elapsed.
DATA: Indicates the data being received or transmitted.
P: Indicates a stop condition, i.e. the master device changing the level on the SSDAn line from the low to the high level while the SSCLn line is at the high level.

Downloaded from Arrow.com.

33.7.1 Generation of Start, Restart, and Stop Conditions

Writing 1 to the SIMR3.IICSTAREQ bit causes the generation of a start condition. The generation of a start condition proceeds through the following operations.

- The level on the SSDAn line falls (from the high level to the low level) and the SSCLn line is kept in the released state.
- The hold time for the start condition is secured as half of a bit period at the bit rate determined by the setting of the BRR.
- The level on the SSCLn line falls (from the high level to the low level), the SIMR3.IICSTAREQ bit is set (to 0), and a start-condition generated interrupt is output.

Writing 1 to the SIMR3.IICRSTAREQ bit causes the generation of a start condition. The generation of a start condition proceeds through the following operations.

- The SSDAn line is released and the SSCLn line is kept at the low level.
- The period at low level for the SSCLn line is secured as half of a bit period at the bit rate determined by the setting of the BRR.
- The SSCLn line is released (transition from the low to the high level).
- Once the high level on the SSCLn line is detected, the setup time for the restart condition is secured as half of a bit period at the bit rate determined by the setting of the BRR.
- The level on the SSDAn line falls (from the high level to the low level).
- The hold time for the restart condition is secured as half of a bit period at the bit rate determined by the setting of the BRR.
- The level on the SSCLn line falls (from the high level to the low level), the SIMR3.IICRSTAREQ bit is set (to 0), and a restart-condition generated interrupt is output.

Writing 1 to the SIMR3.IICSTPREQ bit causes the generation of a stop condition. The generation of a stop condition proceeds through the following operations.

- The level on the SSDAn line falls (from the high level to the low level) and the SSCLn line is kept at the low level.
- The period at low level for the SSCLn line is secured as half of a bit period at the bit rate determined by the setting of the BRR.
- The SSCLn line is released (transition from the low to the high level).
- Once the high level on the SSCLn line is detected, the setup time for the stop condition is secured as half of a bit period at the bit rate determined by the setting of the BRR.
- The SSDAn is released (transition from the low to the high level), the SIMR3.IICSTPREQ bit is set (to 0), and a stop-condition generated interrupt is output.

Downloaded from Arrow.com.

Figure 33.48 shows the timing of operations in the generation of start, restart, and stop conditions.

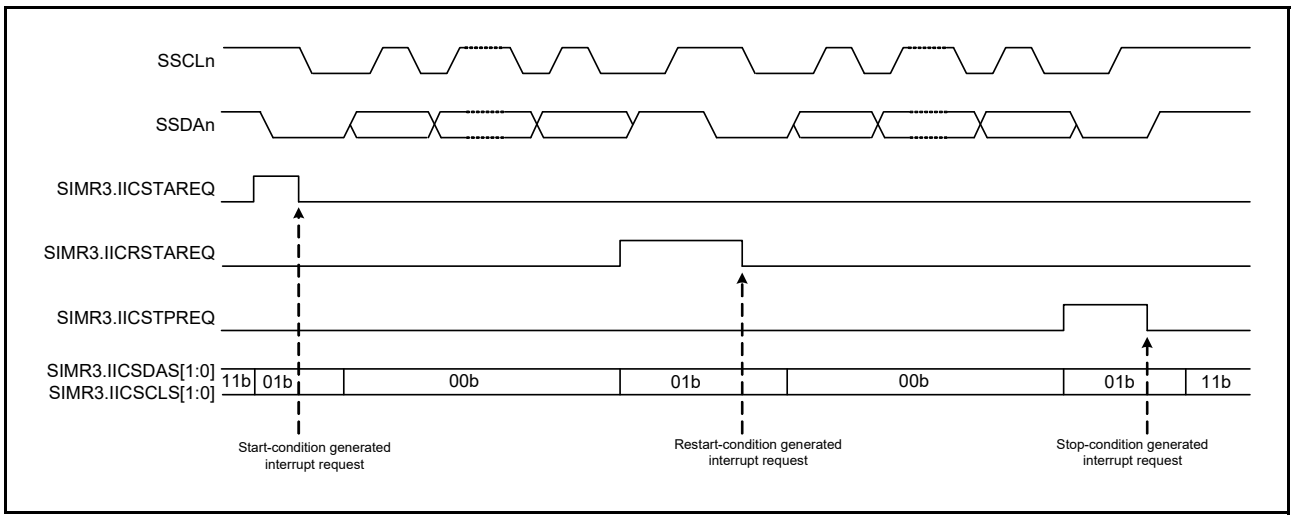


Figure 33.48 Timing of Operations in the Generation of Start, Restart, and Stop Conditions

Downloaded from Arrow.com.

33.7.2 Clock Synchronization

The SSCLn line may be placed at the low level in the case of a wait inserted by a slave device as the other side of transfer. Setting the SIMR2.IICCSC bit to 1 applies control to obtain synchronization when the levels of the internal SSCLn clock signal and the level being input on the SSCLn pin differ.

When the SIMR2.IICCSC bit is set to 1, the level of the internal SSCLn clock signal changes from low to high, counting to determine the period at high level is stopped while the low level is being input on the SSCLn pin, and counting to determine the period at high level starts after the transition of the input on the SSCLn pin to the high level. The interval from this time until counting to determine the period at high level starts on the transition of the SSCLn pin to the high level is the total of the delay of SSCLn output, delay for noise filtering of the input on the SSCLn pin (2 or 3 cycles of sampling clock for the noise filter), and delay for internal processing (1 or 2 cycles of PCLK). The period at high level of the internal SSCLn clock is extended even if other devices are not placing the low level on the SSCLn line. If the SIMR2.IICCSC bit is 1, synchronization is obtained for the transmission and reception of data by taking the logical AND of the input on the SSCLn pin and the internal SSCLn clock. If the SIMR2.IICCSC bit is 0, synchronization with the internal SSCLn clock is obtained for the transmission and reception of data.

If a slave device inserts a period of waiting into the interval until the transition of the internal SSCLn clock signal from the low to the high level after a request for the generation of a start, restart, or stop condition is issued, the time until generation is prolonged by that period.

If a slave device inserts a period of waiting after the transition of the internal SSCLn clock signal from the low to the high level, although the generation-completed interrupt is issued without stopping the period of waiting, generation of the condition itself is not guaranteed. Figure 33.49 shows an example of operations to synchronize the clocks.

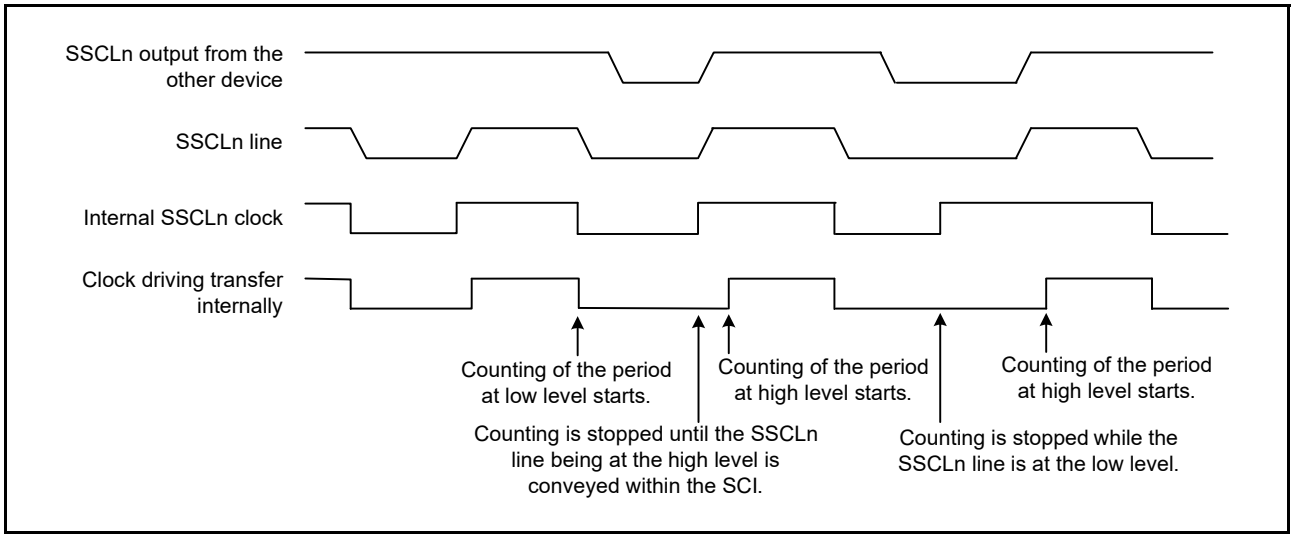


Figure 33.49 Example of Operations for Clock Synchronization

Downloaded from Arrow.com.

33.7.3 SSDA Output Delay

The SIMR1.IICDL[4:0] bits can be used to set a delay for output on the SSDAn pin relative to falling edges of output on the SSCLn pin. Delay-time settings from 0 to 31 are selectable, representing periods of the corresponding numbers of cycles of the clock signal from the on-chip baud rate generator (derived by frequency-dividing the base clock, PCLK, by the divisor selected by the SMR.CKS[1:0] bits). A delay for output on the SSDAn pin is for the start condition/restart condition/stop condition signal, 8-bit transmit data, and an acknowledge bit.

If the SSDA output delay is shorter than the time for the level on the SSCLn pin to fall, the change of the output on the SSDAn pin will start while the output level on the SSCLn pin is falling, creating a possibility of erroneous operation for slave devices. Ensure that settings for the delay of output on the SSDAn pin are for times greater than the time output on the SSCLn pin takes to fall (300 ns for I^2C-bus in normal mode and fast mode).

Figure 33.50 shows the timing of delays in SSDA output.

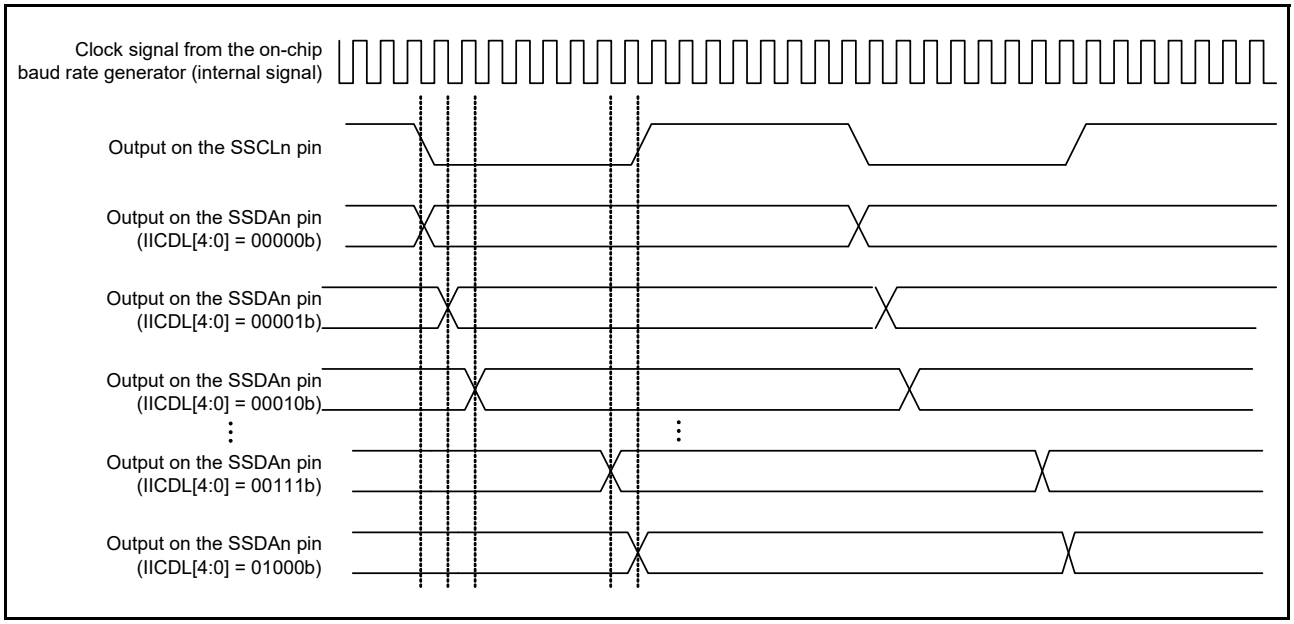


Figure 33.50 Timing of Delays in SSDA Output

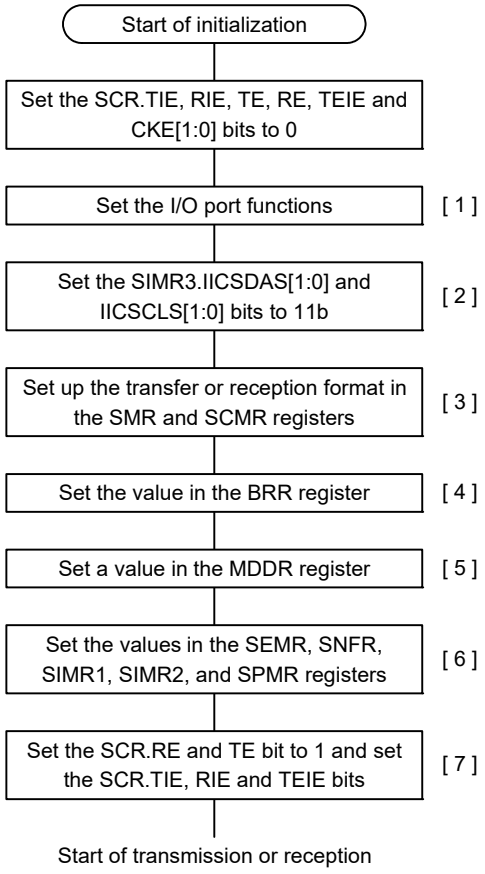
Downloaded from Arrow.com.

33.7.4 SCI Initialization (Simple I2C Mode)

Before transferring data, write the initial value (00h) to the SCR register and initialize the interface following the example shown in Figure 33.51.

When changing the operating mode, transfer format, and so on, be sure to set the SCR register to its initial value before proceeding with the changes.

In simple I2C mode, the open-drain setting for the communication ports should be made on the port side.



[1] Make I/O port settings that allow use (on N-channel open-drain output pins) of the SSCLn and SSDAn pin functions.

[2] Place the SSCLn and SSDAn pins in the high-impedance state until a start condition is to be generated.

[3] Set the format for transmission and reception in the SMR and SCMR registers.
In the SMR register, set the CKS[1:0] bits to the desired value and set the other bits to 0.
In the SCMR register, set the SDIR bit to 1 and the SINV and SMIF bits to 0.

[4] Write the value for the desired bit rate to the BRR register.

[5] Write the value obtained by correcting a bit rate error in the MDDR register. This step is not necessary if the SEMR.BRME bit is set to 0.

[6] Set the values in the SEMR, SNFR, SIMR1, SIMR2, and SPMR registers.
In the SEMR register, set the NFEN and BRME bits.
In the SNFR register, set the NFCS[2:0] bits.
In the SIMR1 register, set the IICM bit to 1 and the IICDL[4:0] bits as required.
In the SIMR2 register, set the IICACKT and IICCSC bits to 1 and the IICINTM bits as required.
In the SPMR register, set all the bits to 0.

[7] Set the SCR.RE and TE bits to 1. At this time, also set the SCR.TIE, RIE, and TEIE bits (for transmission and when the SIMR2.IICINTM bit is 1, set the RIE bit to 0).
Setting the TE and RE bits to 1 makes the SSCLn and SSDAn pin functions available.

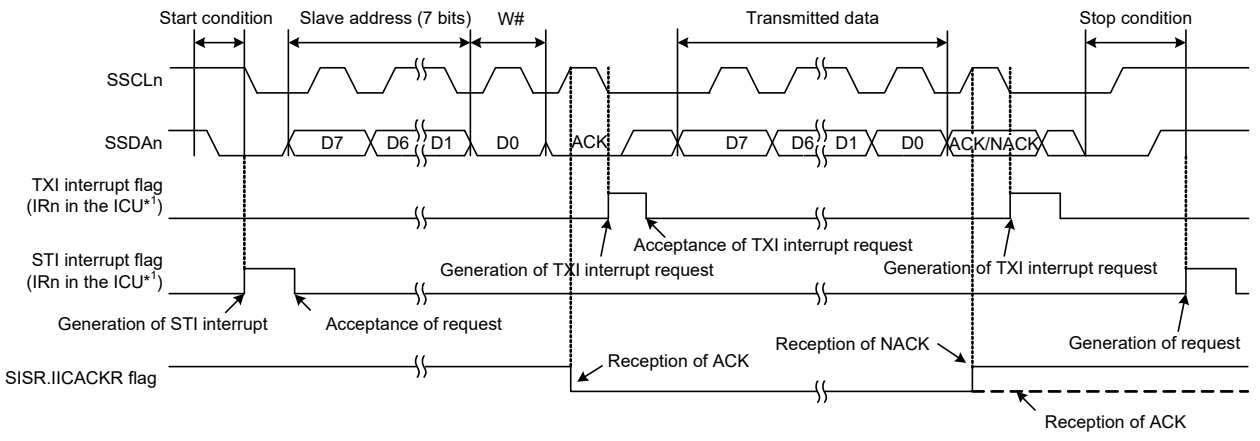
Note: Set the SCR.RE and TE bits to 0 or 1 at the same time.

Figure 33.51 Example of the Flowchart of SCI Initialization (for Simple I2C Mode)

Downloaded from Arrow.com.

33.7.5 Operation in Master Transmission (Simple I2C Mode)

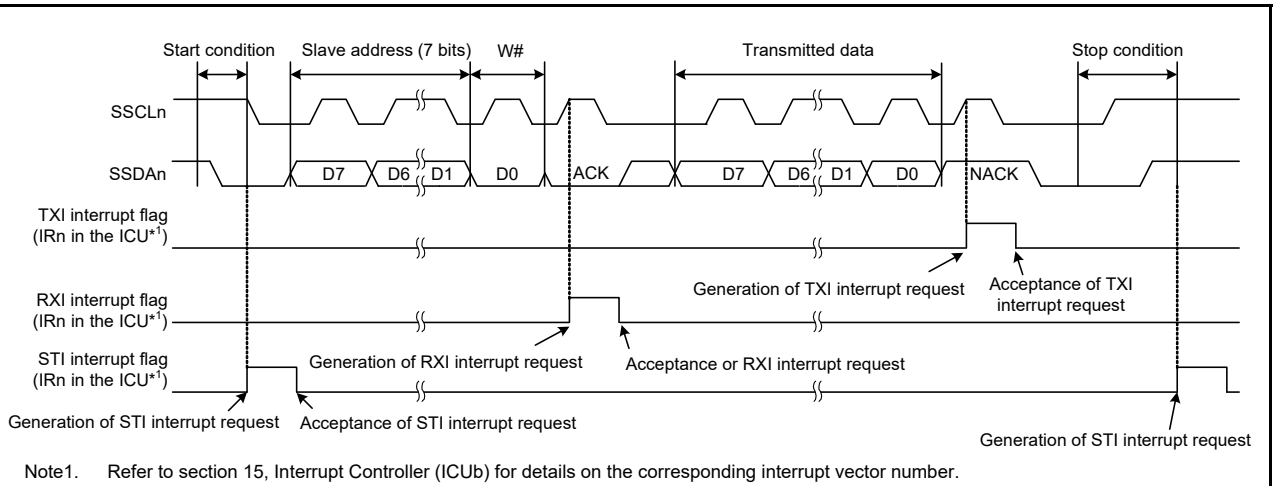
Figure 33.52 and Figure 33.53 show examples of operations in master transmission and Figure 33.54 is a flowchart showing the procedure for data transmission. Refer to Table 33.33 for more information on the STI interrupt.
When 10-bit slave addresses are in use, steps [3] and [4] in Figure 33.54 are repeated twice.
In simple I2C mode, the transmit data empty interrupt (TXI) is generated when communication of one frame is completed, unlike the TXI interrupt request generation timing during clock synchronous transmission.



Note1. Refer to section 15, Interrupt Controller (ICUb) for details on the corresponding interrupt vector number.

Figure 33.52 Example 1 of Operations for Master Transmission in Simple I2C-bus Mode (with 7-Bit Slave Addresses, Transmission Interrupts, and Reception Interrupts in Use)

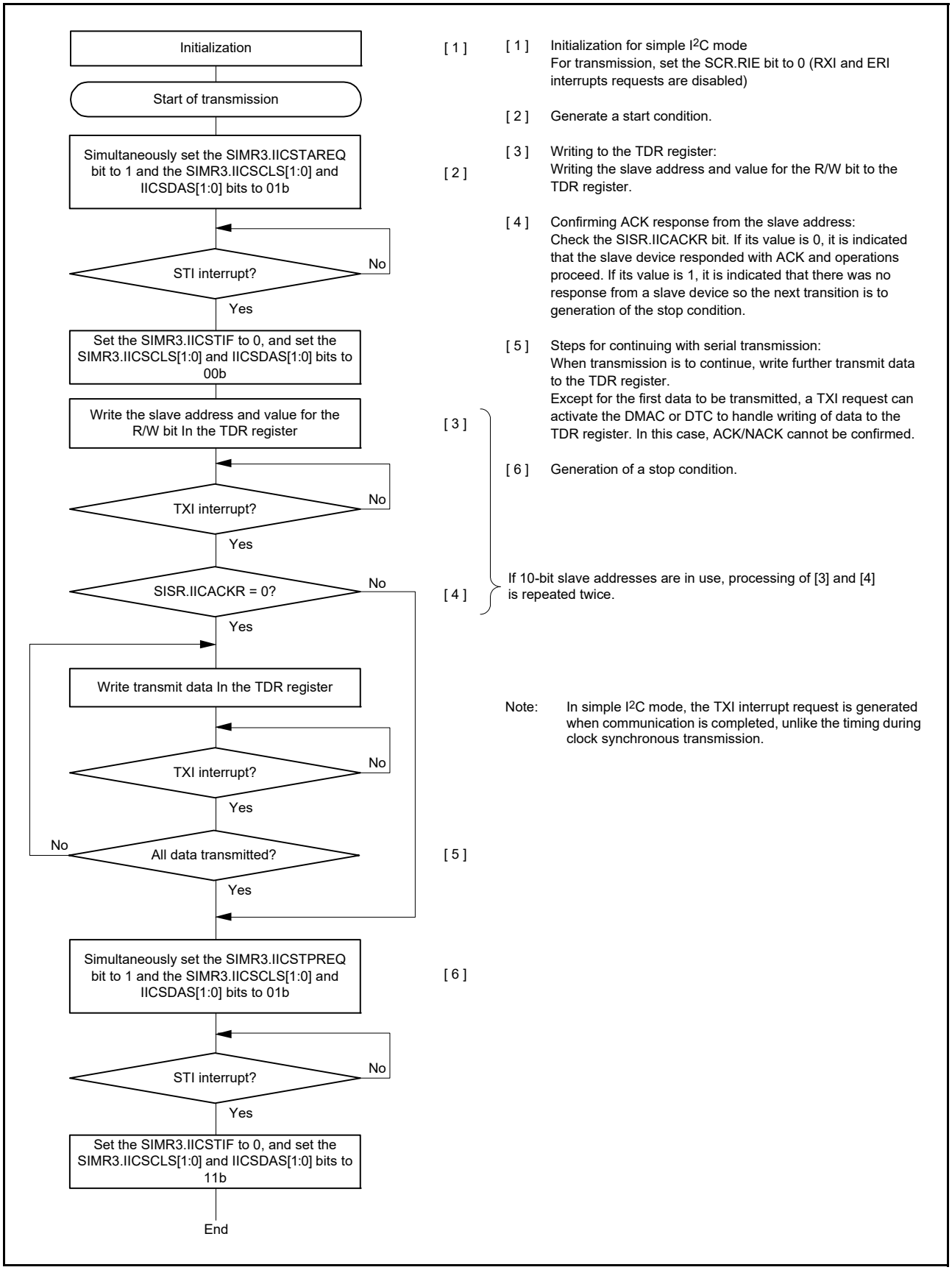
When the SIMR2.IICINTM bit is set to 0 (use ACK/NACK interrupts) during master transmission, the DTC or DMAC is activated by the ACK interrupt as the trigger and necessary number of data bytes are transmitted. When the NACK is received, error processing, such as transmission stop and retransmission, is performed by the NACK interrupt as the trigger.



Note1. Refer to section 15, Interrupt Controller (ICUb) for details on the corresponding interrupt vector number.

Figure 33.53 Example 2 of Operations for Master Transmission in Simple I2C-bus Mode (with 7-Bit Slave Addresses, ACK Interrupts, and NACK Interrupts in Use)

Downloaded from Arrow.com.



[1] Initialization for simple I²C mode
For transmission, set the SCR.RIE bit to 0 (RXI and ERI interrupts requests are disabled)

[2] Generate a start condition.

[3] Writing to the TDR register:
Writing the slave address and value for the R/W bit to the TDR register.

[4] Confirming ACK response from the slave address:
Check the SISR.IICACKR bit. If its value is 0, it is indicated that the slave device responded with ACK and operations proceed. If its value is 1, it is indicated that there was no response from a slave device so the next transition is to generation of the stop condition.

[5] Steps for continuing with serial transmission:
When transmission is to continue, write further transmit data to the TDR register.
Except for the first data to be transmitted, a TXI request can activate the DMAC or DTC to handle writing of data to the TDR register. In this case, ACK/NACK cannot be confirmed.

[6] Generation of a stop condition.

[3] to [4] If 10-bit slave addresses are in use, processing of [3] and [4] is repeated twice.

Note: In simple I²C mode, the TXI interrupt request is generated when communication is completed, unlike the timing during clock synchronous transmission.

Figure 33.54 Example of the Procedure for Master Transmission Operations in Simple I²C Mode (with Transmission Interrupts and Reception Interrupts in Use)

Downloaded from Arrow.com.

33.7.6 Master Reception (Simple I2C Mode)

Figure 33.55 shows an example of operations in simple I2C mode master reception and Figure 33.56 is a flowchart showing the procedure for master reception.

The value of the SIMR2.IICINTM bit is assumed to be 1 (use reception and transmission interrupts).

In simple I2C mode, the transmit data empty interrupt (TXI) is generated when communication of one frame is completed, unlike the TXI interrupt request generation timing during clock synchronous transmission.

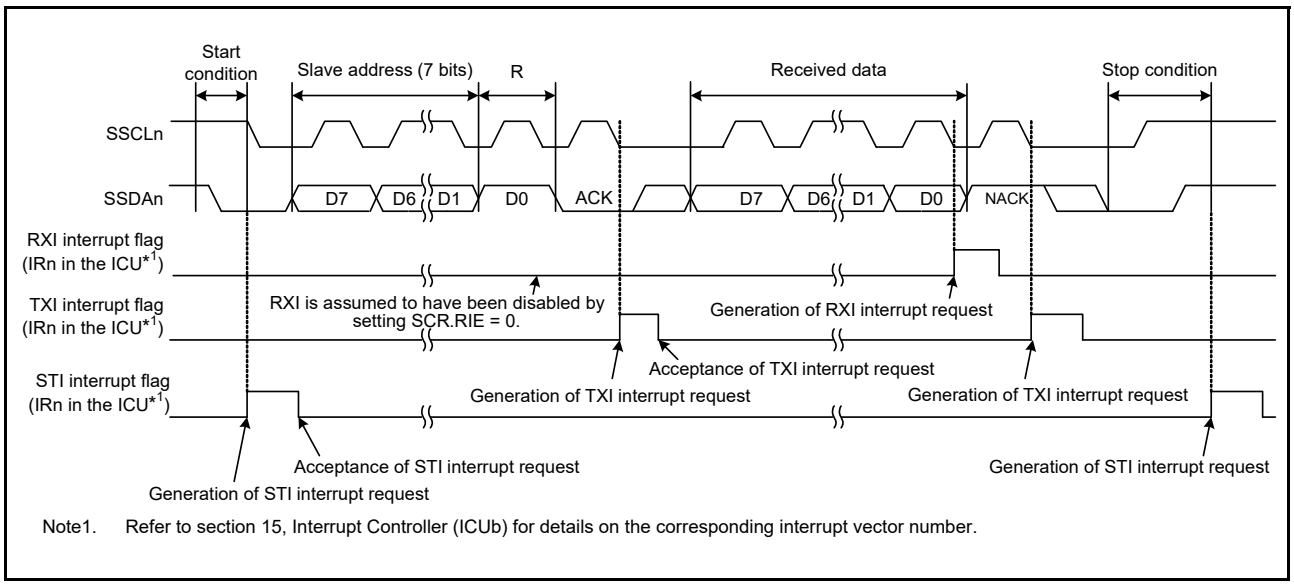
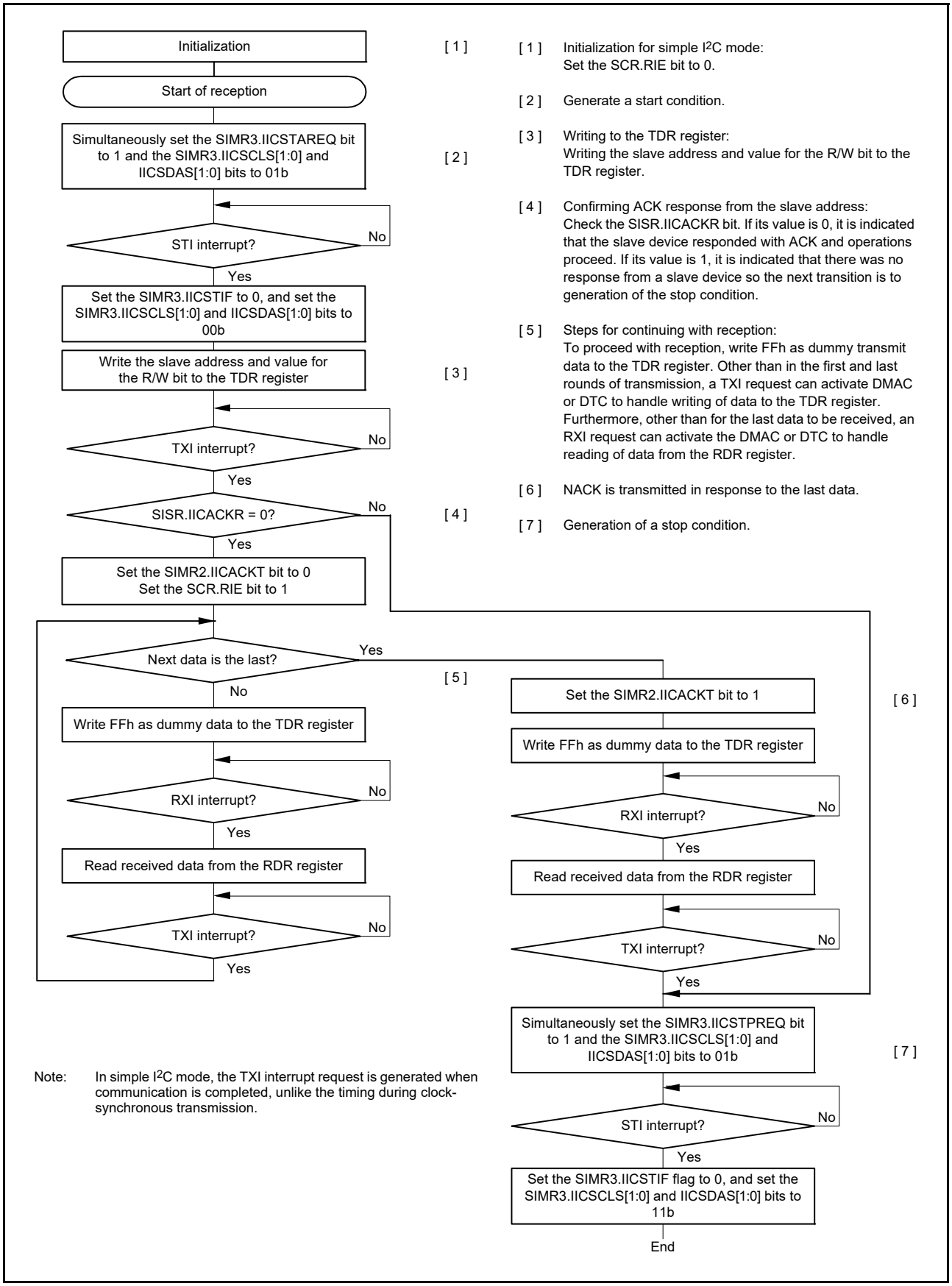


Figure 33.55 Example of Operations for Master Reception in Simple I2C-bus Mode (with 7-Bit Slave Addresses, Transmission Interrupts, and Reception Interrupts in Use)

Downloaded from Arrow.com.



[1] Initialization for simple I²C mode:
Set the SCR.RIE bit to 0.

[2] Generate a start condition.

[3] Writing to the TDR register:
Writing the slave address and value for the R/W bit to the TDR register.

[4] Confirming ACK response from the slave address:
Check the SISR.IICACKR bit. If its value is 0, it is indicated that the slave device responded with ACK and operations proceed. If its value is 1, it is indicated that there was no response from a slave device so the next transition is to generation of the stop condition.

[5] Steps for continuing with reception:
To proceed with reception, write FFh as dummy transmit data to the TDR register. Other than in the first and last rounds of transmission, a TXI request can activate DMAC or DTC to handle writing of data to the TDR register. Furthermore, other than for the last data to be received, an RXI request can activate the DMAC or DTC to handle reading of data from the RDR register.

[6] NACK is transmitted in response to the last data.

[7] Generation of a stop condition.

Note: In simple I²C mode, the TXI interrupt request is generated when communication is completed, unlike the timing during clock-synchronous transmission.

Figure 33.56 Example of the Procedure for Master Reception Operations in Simple I²C Mode (with Transmission Interrupts and Reception Interrupts in Use)

Downloaded from Arrow.com.

33.7.7 Recovery from Bus Hang-up

If the bus is stuck by an abnormal state in SCI because of the communication error, reset the SCI according to the following steps and release the bus.

(1) Set the SCR.TE and RE bit to 0 at the same time to reset SCI.
(2) Set the SIMR3 register to F0h to release the bus.
(3) If the SSR.RDRF flag is 1, dummy-read the RDR register to clear the flag.
(4) Set the SCR.TE and RE bit to 1 at the same time.

Downloaded from Arrow.com.

33.8 Operation in Simple SPI Mode

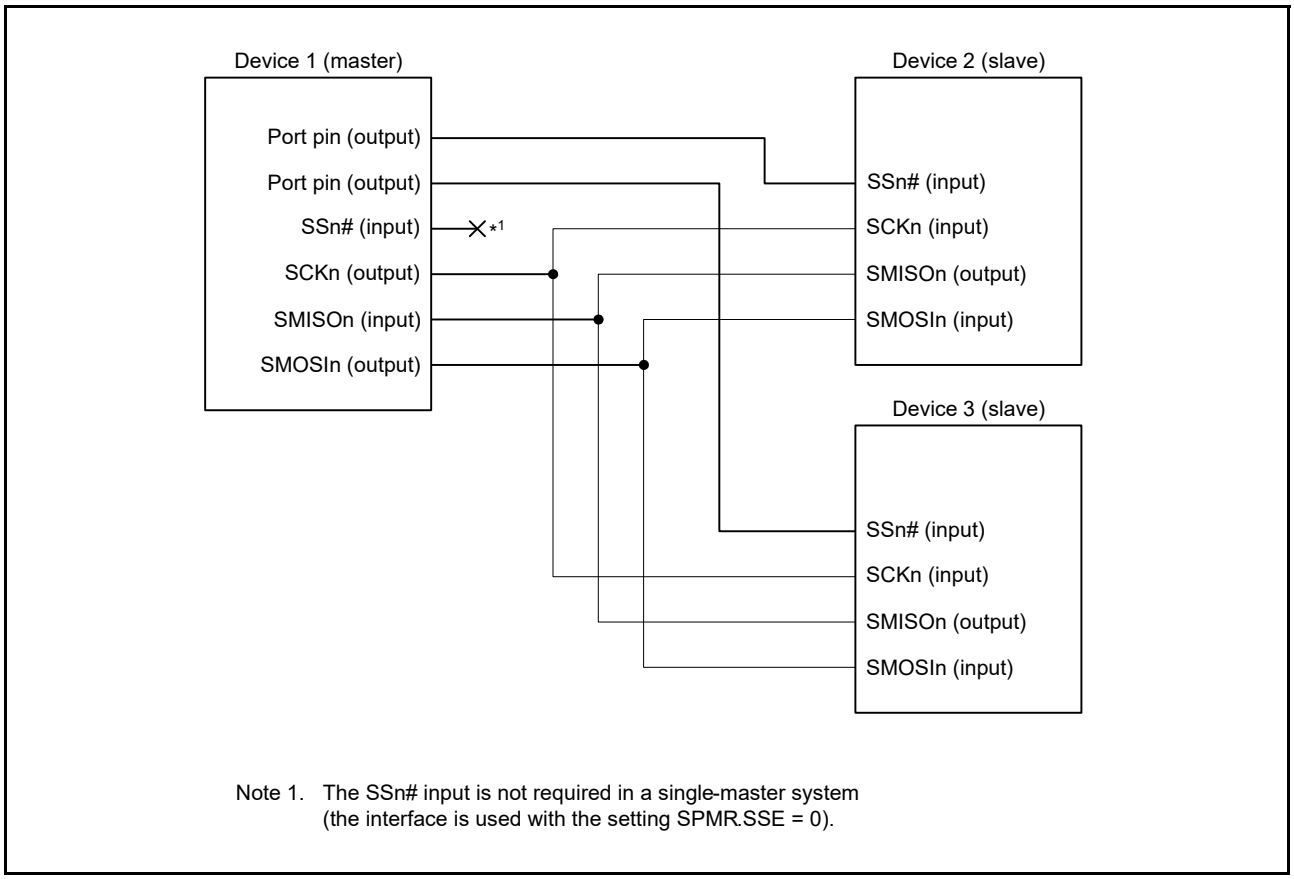
As an extended function, the SCI supports a simple SPI mode that handles transfer among one or multiple master devices and multiple slave devices.

Making the settings for clock synchronous mode (SCMR.SMIF = 0, SIMR1.IICM = 0, SMR.CM = 1) plus setting the SPMR.SSE bit to 1 places the SCI in simple SPI mode. However, the SS pin function on the master side is unnecessary for connection of the device used as the master in simple SPI mode when the configuration only has a single master, so set the SPMR.SSE bit to 0 in such cases.

Figure 33.57 shows an example of connections for simple SPI mode. Control a general port pin to produce the SS output signal from the master.

In simple SPI mode, data are transferred in synchronization with clock pulses in the same way as in clock synchronous mode. One character of data for transfer consists of 8 bits of data, and parity bits cannot be appended to this. The data can be inverted by setting the SCMR.SINV bit to 1.

Since the receiver and transmitter are independent of each other within the SCI module, full-duplex communications are possible, with a common clock signal. Furthermore, since both the transmitter and receiver have a double-buffered structure, writing of further transmit data while transmission is in progress and reading of previously received data while reception is in progress are both possible. Continuous transfer is thus possible.



Note 1. The SSn# input is not required in a single-master system (the interface is used with the setting SPMR.SSE = 0).

Figure 33.57 Example of Connections via a Simple SPI Mode (In Single Master Mode, SPMR.SSE Bit = 0)

Downloaded from Arrow.com.

33.8.1 States of Pins in Master and Slave Modes

The direction (input or output) of pins for the simple SPI mode interface differs according to whether the device is a master (SCR.CKE[1:0] = 00b or 01b and SPMR.MSS = 0) or slave (SCR.CKE[1:0] = 10b or 11b and SPMR.MSS = 1). Table 33.29 lists the states of pins according to the mode and the level on the SSn# pin.

Table 33.29 States of Pins by Mode and Input Level on the SSn# Pin

Mode	Input on SSn# Pin	State of SMOSIn Pin	State of SMISOn Pin	State of SCKn Pin
Master mode*1	High level (transfer can proceed)	Output for data transmission*2	Input for received data	Clock output*3
	Low level (transfer cannot proceed)	High-impedance	Input for received data (but disabled)	High-impedance
Slave mode	High level (transfer cannot proceed)	Input for received data (but disabled)	High-impedance	Clock input (but disabled)
	Low level (transfer can proceed)	Input for received data	Output for data transmission	Clock input

Note 1. When there is only a single master (SPMR.SSE = 0), transfer is possible regardless of the input level on the SSn# pin (this is equivalent to input of a high level on the SSn# pin). Since the SSn# pin function is not required, the pin is available for other purposes.
Note 2. The SMOSIn pin output is in the high-impedance state when serial transmission is disabled (SCR.TE bit = 0).
Note 3. The SCKn pin output is in the high-impedance state when serial transmission is disabled (SCR.TE and RE bits = 00b) in a multi-master configuration (SPMR.SSE = 1).

33.8.2 SS Function in Master Mode

Setting the SCR.CKE[1:0] bits to 00b and the SPMR.MSS bit to 0 selects master operation. The SSn# pin is not used in single-master configurations (SPMR.SSE = 0), so transmission or reception can proceed regardless of the value of the SSn# pin.

When the level on the SSn# pin is high in a multi-master configuration (SPMR.SSE = 1), a master device outputs clock signals from the SCKn pin before starting transmission or reception to indicate that there are no other masters or another master is performing reception or transmission. When the level on the SSn# pin is low in a multi-master configuration (SPMR.SSE = 1), there are other masters, and this indicates that transmission or reception is in progress. At this time the SMOSIn output and SCKn pins will be placed in the high-impedance state and starting transmission or reception will not be possible. Furthermore, the value of the SPMR.MFF bit will be 1, indicating a mode fault error. In a multi-master configuration, start error processing by reading SPMR.MFF flag. Also, even if a mode fault error occurs while transmission or reception is in progress, transmission or reception will not be stopped, but the SMOSIn and SCKn pin output will be placed in the high-impedance state after the completion of the transfer.

Control a general port pin to produce the SS output signal from the master.

33.8.3 SS Function in Slave Mode

Setting the SCR.CKE[1:0] bits to 10b and the SPMR.MSS bit to 1 selects slave operation. When the level on the SSn# pin is high, the SMISOn output pin will be in the high-impedance state and clock input through the SCKn pin will be ignored. When the level on the SSn# pin is low, clock input through the SCKn pin will be effective and transmission or reception can proceed.

If the input on the SSn# pin changes from low to high level during transmission or reception, the SMISOn output pin will be placed in the high-impedance state. Meanwhile, the internal processing for transmission or reception will continue at the rate of the clock input through the SCKn pin until processing for the character currently being transmitted or received is completed, after which it stops. At that time, an interrupt (the appropriate one from among TXI, RXI, and TEI) will be generated.

Downloaded from Arrow.com.

33.8.4 Relationship between Clock and Transmit/Receive Data

The CKPOL and CKPH bits in the SPMR register can be used to set up the clock for use in transmission and reception in four different ways. The relation between the clock signal and the transmission and reception of data is shown in Figure 33.58. The relation is the same for both master and slave operation.

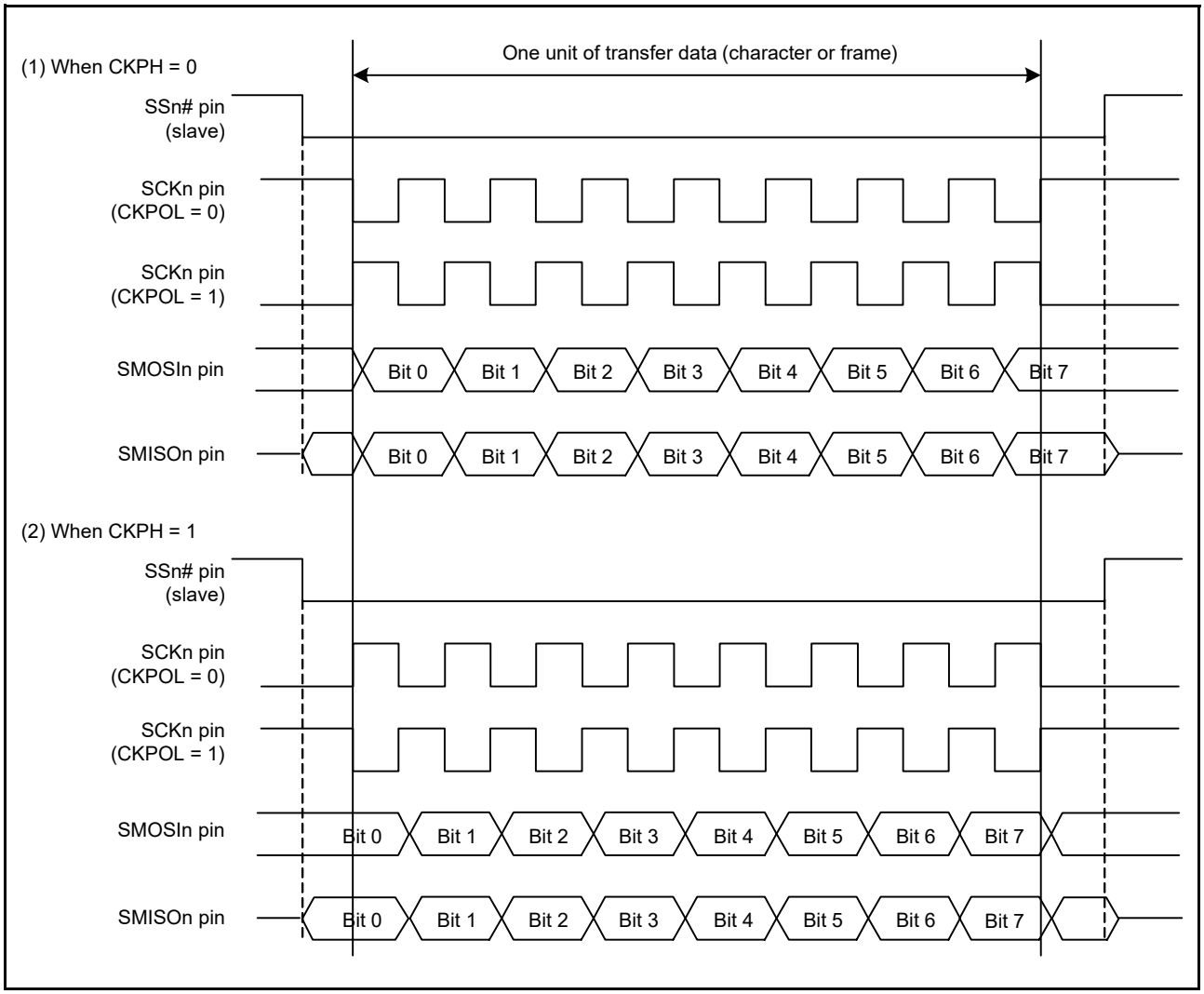


Figure 33.58 Relation between Clock Signal and Transmit/Receive Data in Simple SPI Mode

Downloaded from Arrow.com.

33.8.5 SCI Initialization (Simple SPI Mode)

The procedure is the same as for initialization in clock synchronous mode Figure 33.24, Sample SCI Initialization Flowchart. The CKPOL and CKPH bits in the SPMR register must be set to ensure that the kind of clock signal they select is suitable for both master and slave devices.
For initialization, changes to the operating mode, changes to the transfer format, and so on, initialize the SCR register before proceeding with changes.
As well as setting the RE bit to 0, note that the SSR.ORER, FER, and PER flags, as well as the RDR register, are not initialized.

33.8.6 Transmission and Reception of Serial Data (Simple SPI Mode)

In master operation, ensure that the SSn# pin of the slave device on the other side of the transfer is at the low level before starting the transfer and at the high level on completion of the transfer. Otherwise, the procedures are the same as in clock synchronous mode.

33.9 Bit Rate Modulation Function

The bit rate modulation function corrects the bit rate by thinning out the specified amount of clocks from those input to the baud rate generator.
When the SEMR.BRME bit is 1, the baud rate generator validates and counts the average interval of the number of clocks set in the MDDR register out of the total 256 clocks input.
Figure 33.59 assumes the SCI is in asynchronous mode, bits SMR.CKS[1:0] are 00b, the BRR register is 00h, and the MDDR register is 160. In this example, the cycle of the base clock is evenly corrected to 256/160, and the bit rate is corrected to 160/256. Note that there is an imbalance in thinning out the internal clock, and expansion and contraction occur in the pulse width of the base clock.

Note: Do not use this function in the highest speed settings (SMR.CKS[1:0] = 00b, SCR.CKE[1] = 0, and BRR = 0) in clock synchronous mode and simple SPI mode.

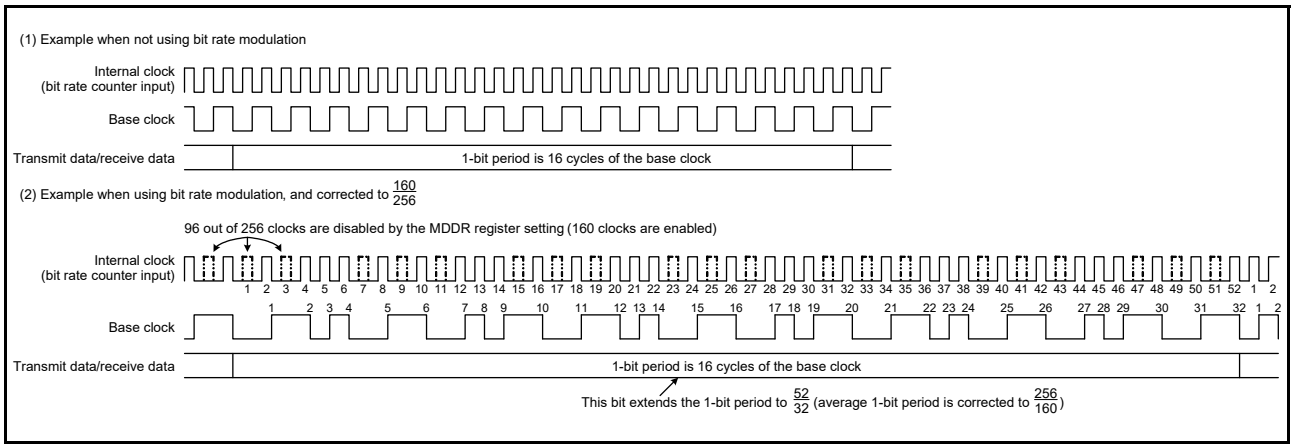


Figure 33.59 Example of the Base Clock When the Bit Rate Modulation Function is Used

The input of a clock signal with a shorter period to the baud rate generator reduces difference in the generated base clock period and, since the division ratio of the baud rate generator also becomes larger, reduces difference in the length of the 1-bit period.

Downloaded from Arrow.com.

33.10 Extended Serial Mode Control Section: Description of Operation

33.10.1 Serial Transfer Protocol

The extended serial mode control section of the SCI12 can realize the serial transfer protocol composed of Start Frames and Information Frames that is shown in Figure 33.60.
A Start Frame is composed of a Break Field, Control Field 0, and Control Field 1. An Information Frame is composed of a number of Data Fields, a CRC16 Upper Field, and a CRC16 Lower Field.

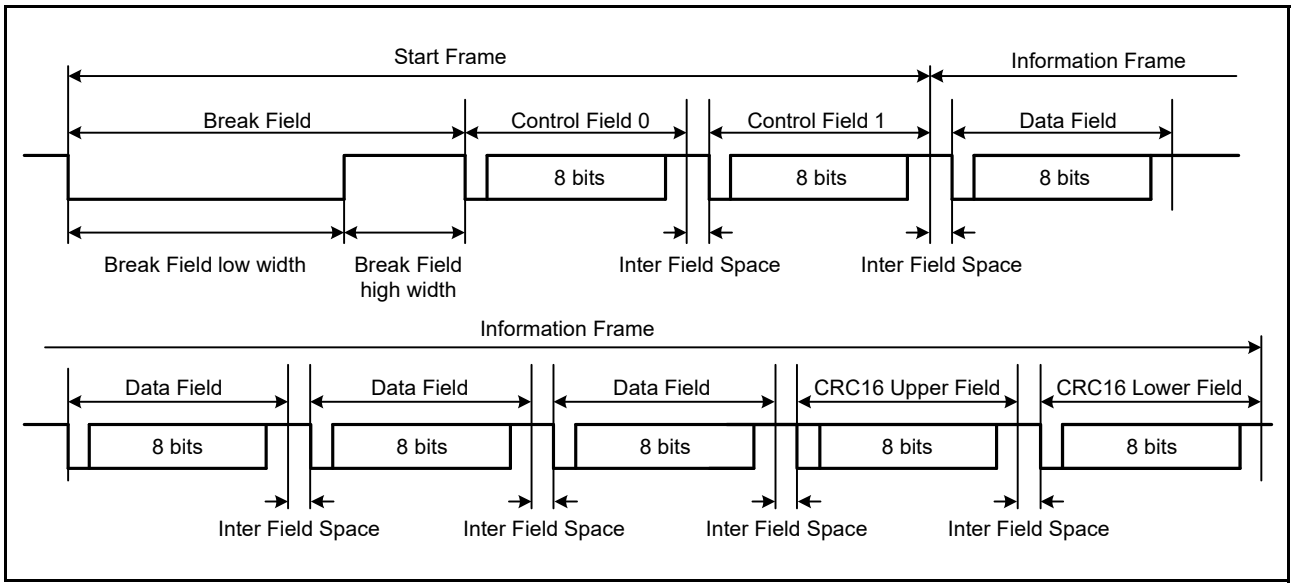


Figure 33.60 Protocol for Serial Transfer by the Extended Serial Mode Control Section

Downloaded from Arrow.com.

33.10.2 Transmitting a Start Frame

Figure 33.61 shows an example of operations to transmit a Start Frame, which is composed of the Break Field low width, Control Field 0, and Control Field 1. Figure 33.62 and Figure 33.63 are flowcharts for the transmission of a Start Frame.
Operations when the extended serial mode control section is to be used to transmit a Start Frame are as listed below. Be sure to use the SCI12 in asynchronous mode.

(1) With Break Field low width output mode as the operating mode for the timer, writing 1 to the TCR.TCST bit starts counting by the timer, and the low level will be output from the TXDX12 pin over the period corresponding to registers TCNT and TPRE settings.
(2) The output on the TXDX12 pin is inverted when the timer counter underflows, and the STR.BFDF flag is set to 1. An SCIX0 interrupt is also generated if the value of the ICR.BFDIE bit is 1.
(3) Write 0 to the TCR.TCST bit to stop counting by the timer, and send the data for Control Field 0. After the Break Field low width output, stop counting before the next underflow occurs.
(4) When the data for Control Field 0 have been transmitted, the data for Control Field 1 is transmitted.
(5) When the data for Control Field 1 have been transmitted, an Information Frame is transmitted.

Omit the Break Field and Control Field 0 to suit the structure of the Start Frame.

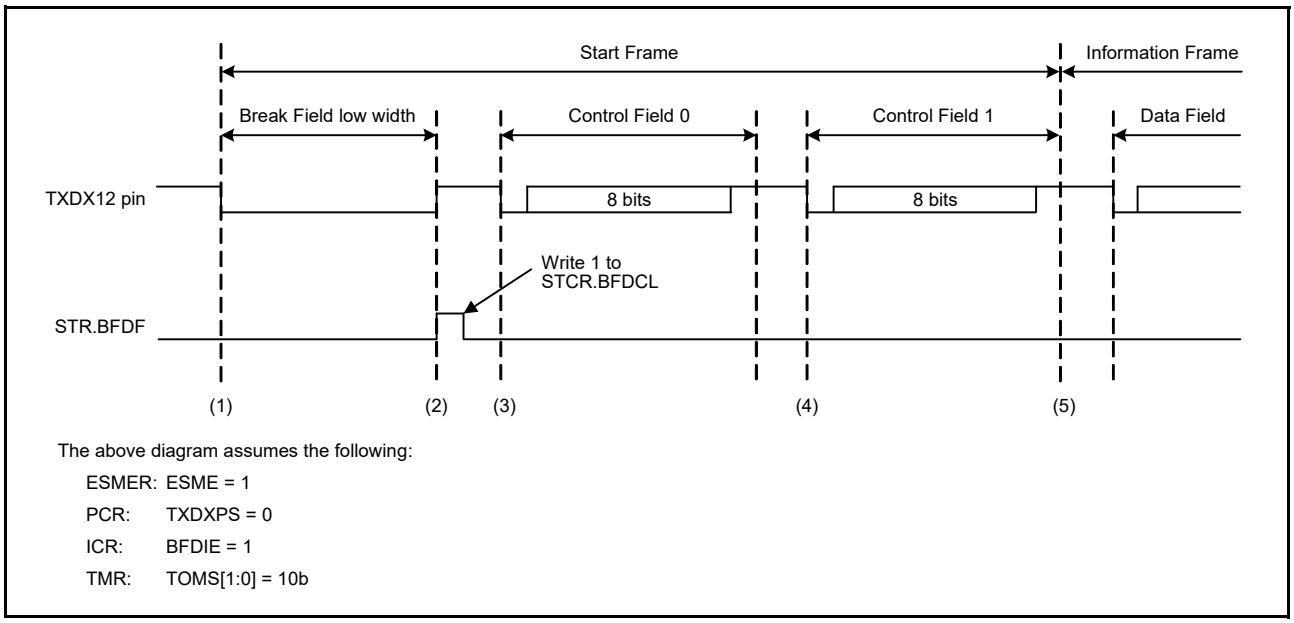


Figure 33.61 Example of Operations When Transmitting a Start Frame

Downloaded from Arrow.com.

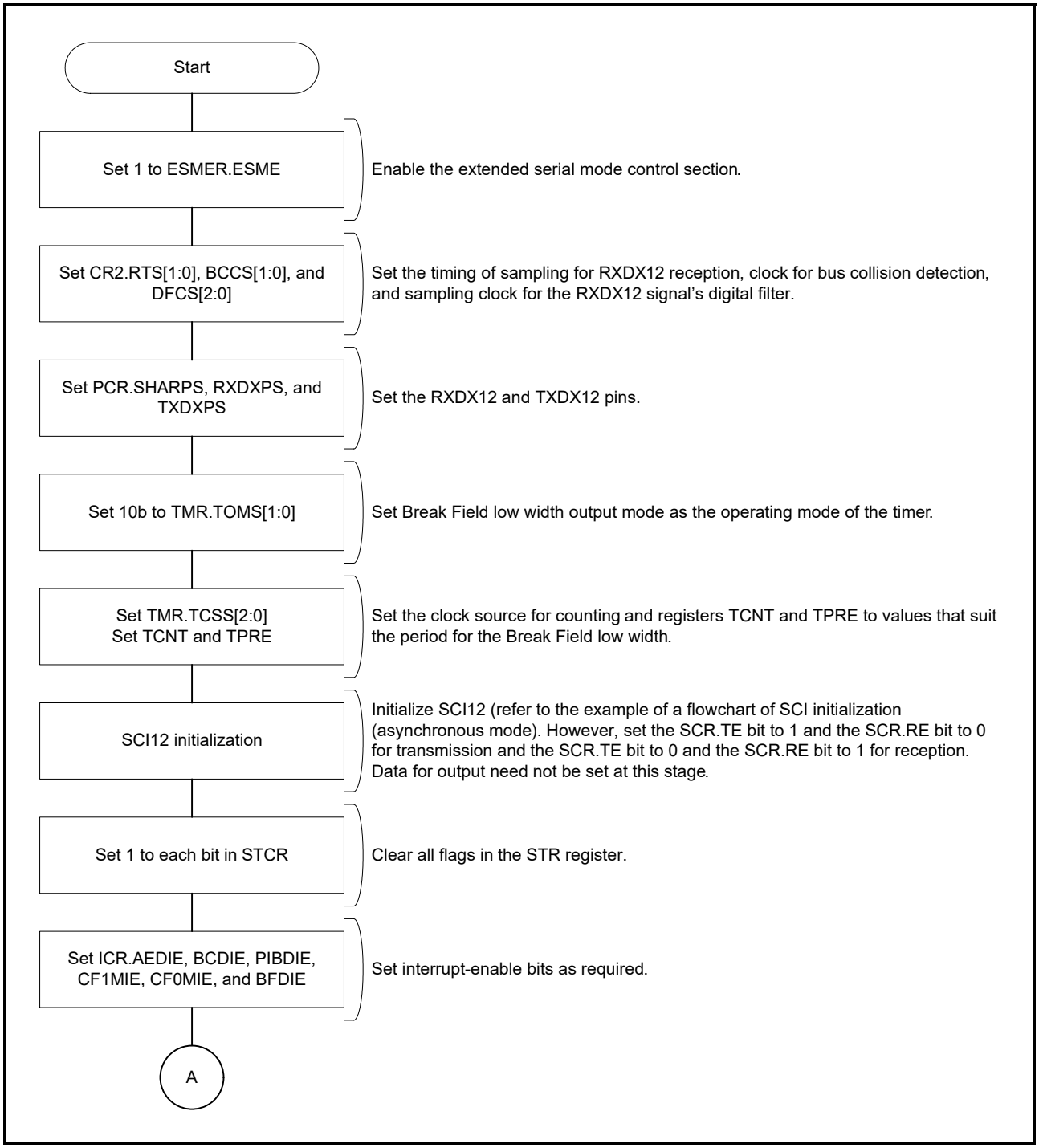


Figure 33.62 Example of Start Frame Transmission (1/2)

Downloaded from Arrow.com.

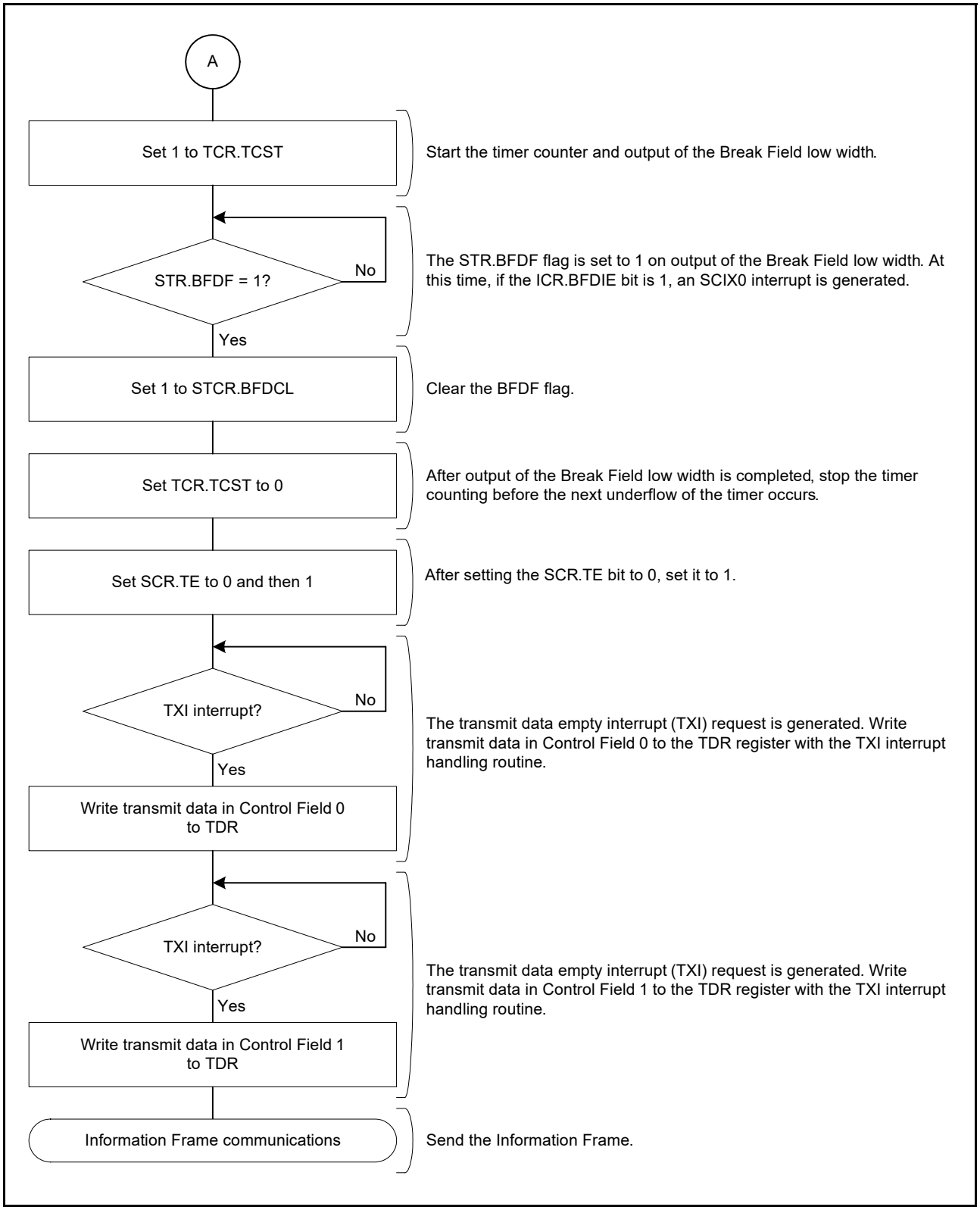


Figure 33.63 Example of Start Frame Transmission (2/2)

Downloaded from Arrow.com.

33.10.3 Receiving a Start Frame

The extended serial mode control section is capable of receiving Start Frames with the structures listed in Table 33.30.

Table 33.30 Structures of Start Frames

Bit Setting		Structures of Start Frames
BFE	CF0RE	
0	0	Control Field 1 (8 bits) → Information Frame
0	1	Control Field 0 (8 bits) → Control Field 1 (8 bits) → Information Frame
1	0	Break Field (low width) → Control Field 1 (8 bits) → Information Frame
1	1	Break Field (low width) → Control Field 0 (8 bits) → Control Field 1 (8 bits) → Information Frame

Figure 33.64 shows an example of operations to receive a Start Frame, which is composed of the Break Field low width, Control Field 0, and Control Field 1. Figure 33.65 and Figure 33.66 are flowcharts for the reception of a Start Frame, and Figure 33.67 is a state transition diagram when receiving a Start Frame.
Operations when the extended serial mode control section is to be used to receive a Start Frame are as listed below. Be sure to use the SCI12 in asynchronous mode.

(1) With Break Field low width detection mode as the operating mode for the timer, writing 1 to the CR3.SDST bit enables detection of the Break Field low width.
(2) Low-level input on the RXDX12 pin continuing over a period longer than that corresponding to the settings of registers TCNT and TPRE is detected as the Break Field low width. At this time, the STR.BFDF flag is set to 1. An SCIX0 interrupt is also generated if the value of the ICR.BFDIE bit is 1.
(3) When the input from the RXDX12 pin goes high after the Break Field low width, the CR0.RXDSF flag becomes 0 and reception of Control Field 0 starts.
(4) If the data received in Control Field 0 match the data set in the CF0DR register, the STR.CF0MF flag is set to 1. An SCIX1 interrupt is also generated if the value of the ICR.CF0MIE bit is 1. Reception of Control Field 1 starts after that. If the data received in Control Field 0 do not match the data set in the CF0DR register, a transition to the state prior to Break Field low width detection proceeds.
(5) If the data received in Control Field 1 match the data set in registers PCF1DR and SCF1DR, the STR.CF1MF flag is set to 1. An SCIX1 interrupt is also generated if the value of the ICR.CF1MIE bit is 1. Transfer of the Information Frame starts after that. If the data received in Control Field 1 do not match the data set in either or both of registers PCF1DR and SCF1DR, a transition to the state prior to Break Field low width detection proceeds.

Omit the Break Field and Control Field 0 to suit the structure of the Start Frame.

Downloaded from Arrow.com.

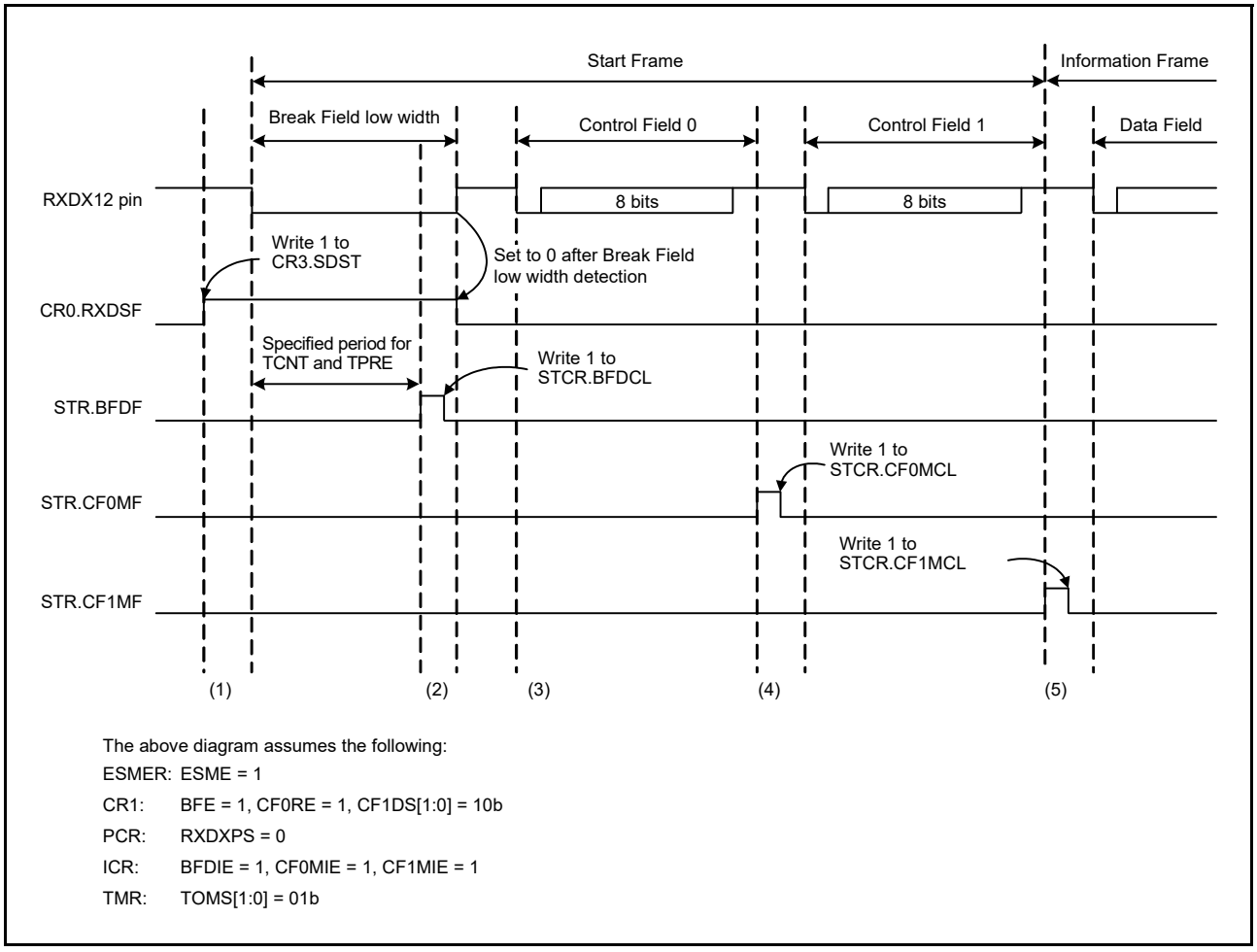


Figure 33.64 Example of Operations at the Time of Start Frame Reception

Downloaded from Arrow.com.

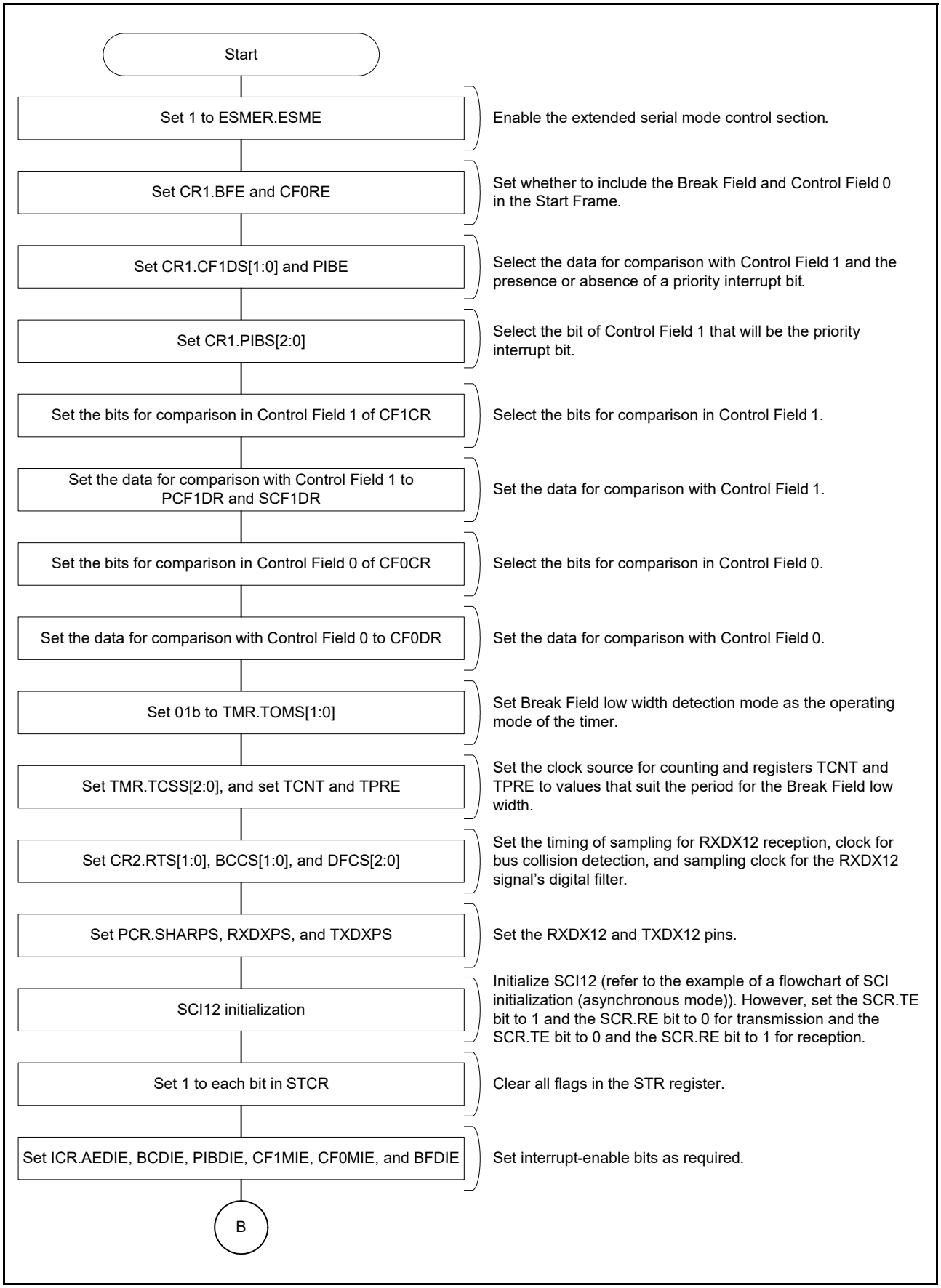


Figure 33.65 Sample Flowchart for Reception of a Start Frame (1)

Downloaded from Arrow.com.

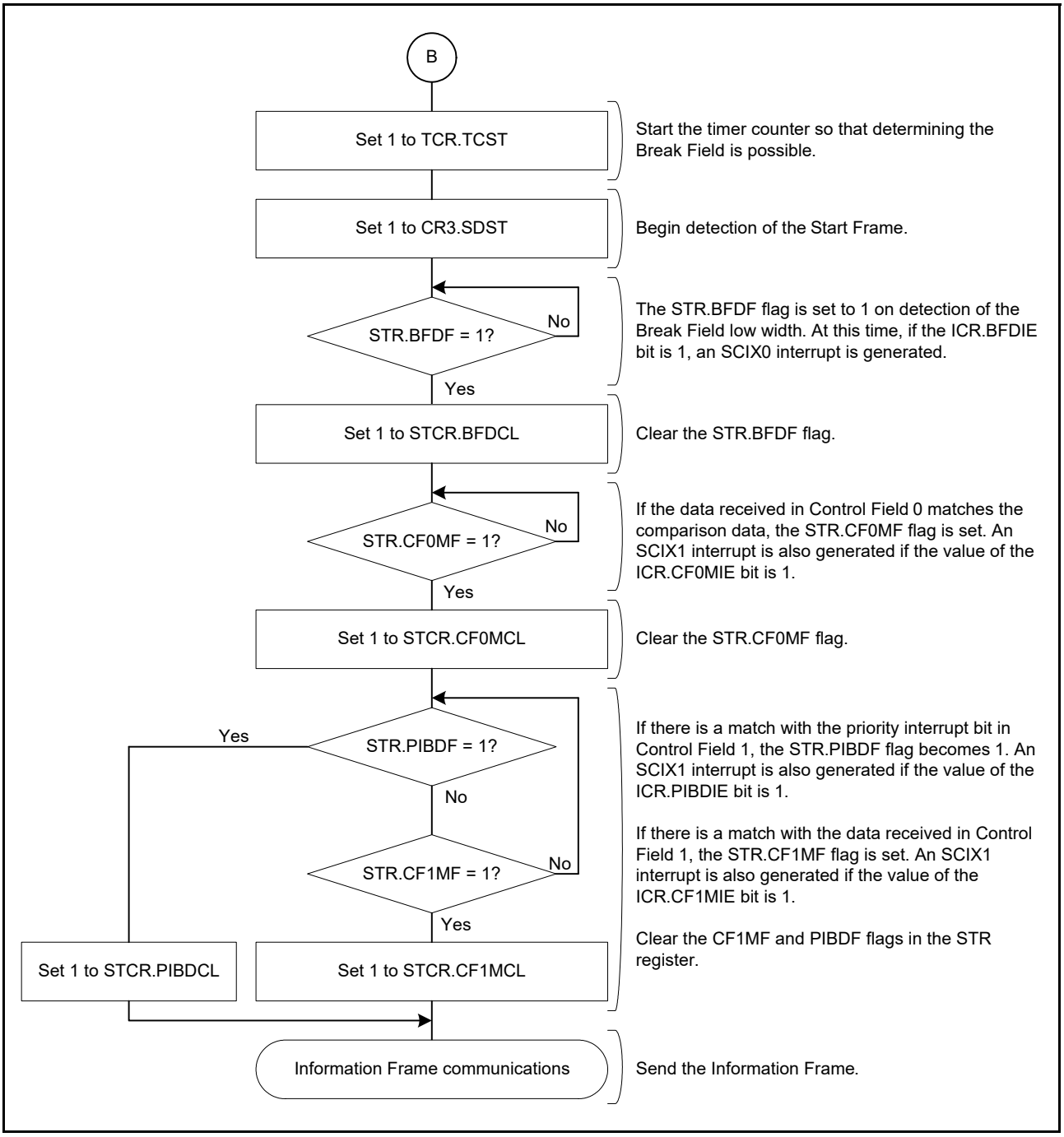


Figure 33.66 Sample Flowchart for Reception of a Start Frame (2)

Downloaded from Arrow.com.

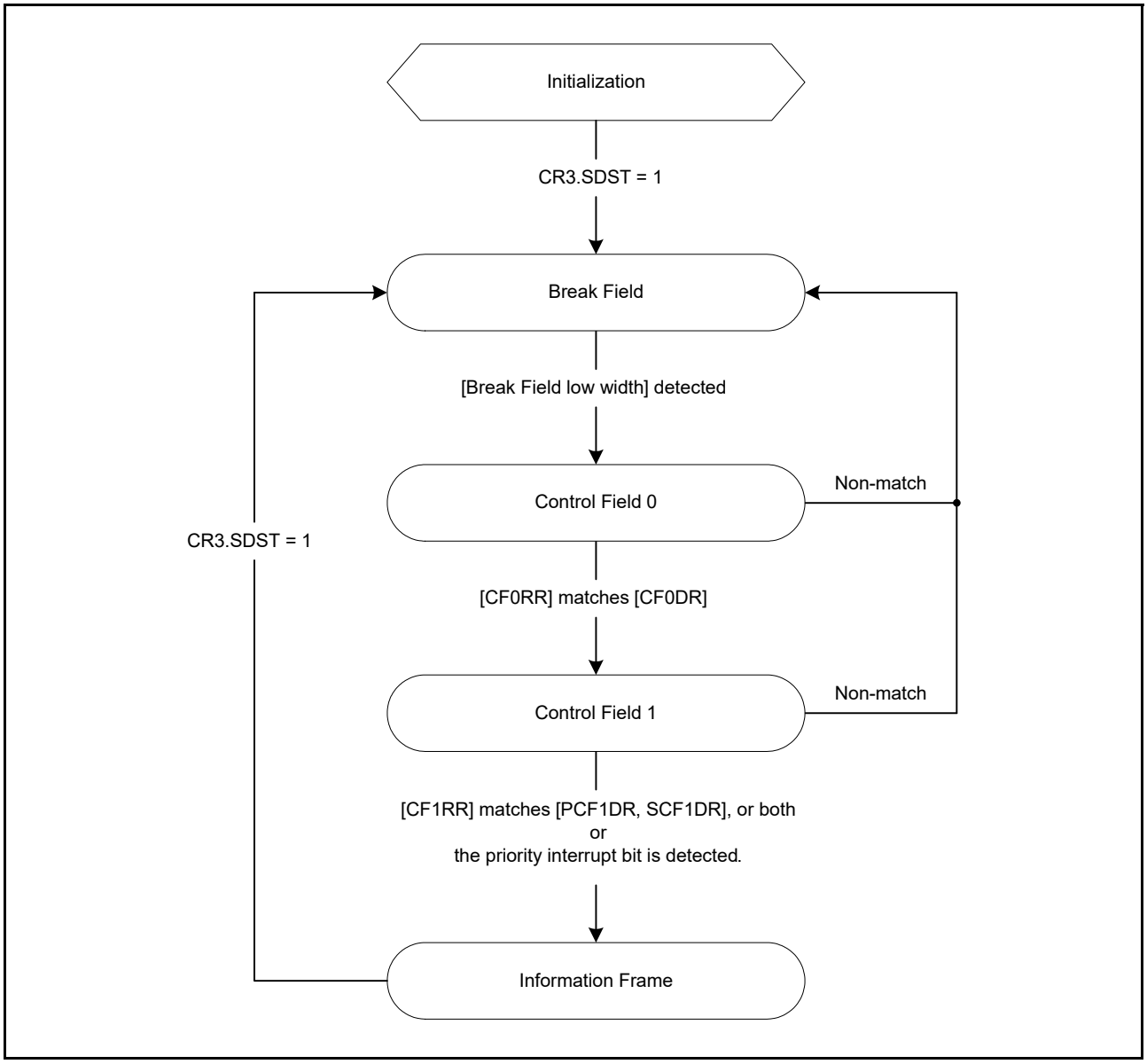


Figure 33.67 State Transitions When Receiving a Start Frame

Downloaded from Arrow.com.

33.10.3.1 Priority Interrupt Bit

Figure 33.68 shows an example of operation in Start Frame reception where a priority interrupt bit is in use. Setting the CR1.PIBE bit to 1 enables the use of a priority interrupt bit.
Operations of the extended serial mode control section in start Frame reception where a priority interrupt bit is in use are as described below.

Steps (1) to (4) are the same as in Figure 33.64, for Start Frame reception.

(5) If the value of the bit selected by the CR1.PIBS[2:0] bits matches the corresponding bit in the PCF1DR register, the STR.PIBDF flag is set to 1. An SCIX1 interrupt is also generated if the value of the ICR.PIBDIE bit is 1. Transfer of the Information Frame starts after that. If the data received in Control Field 1 do not match the data set in either or both of registers PCF1DR and SCF1DR and the priority interrupt bit is not detected, a transition to the state prior to Break Field low width detection proceeds.

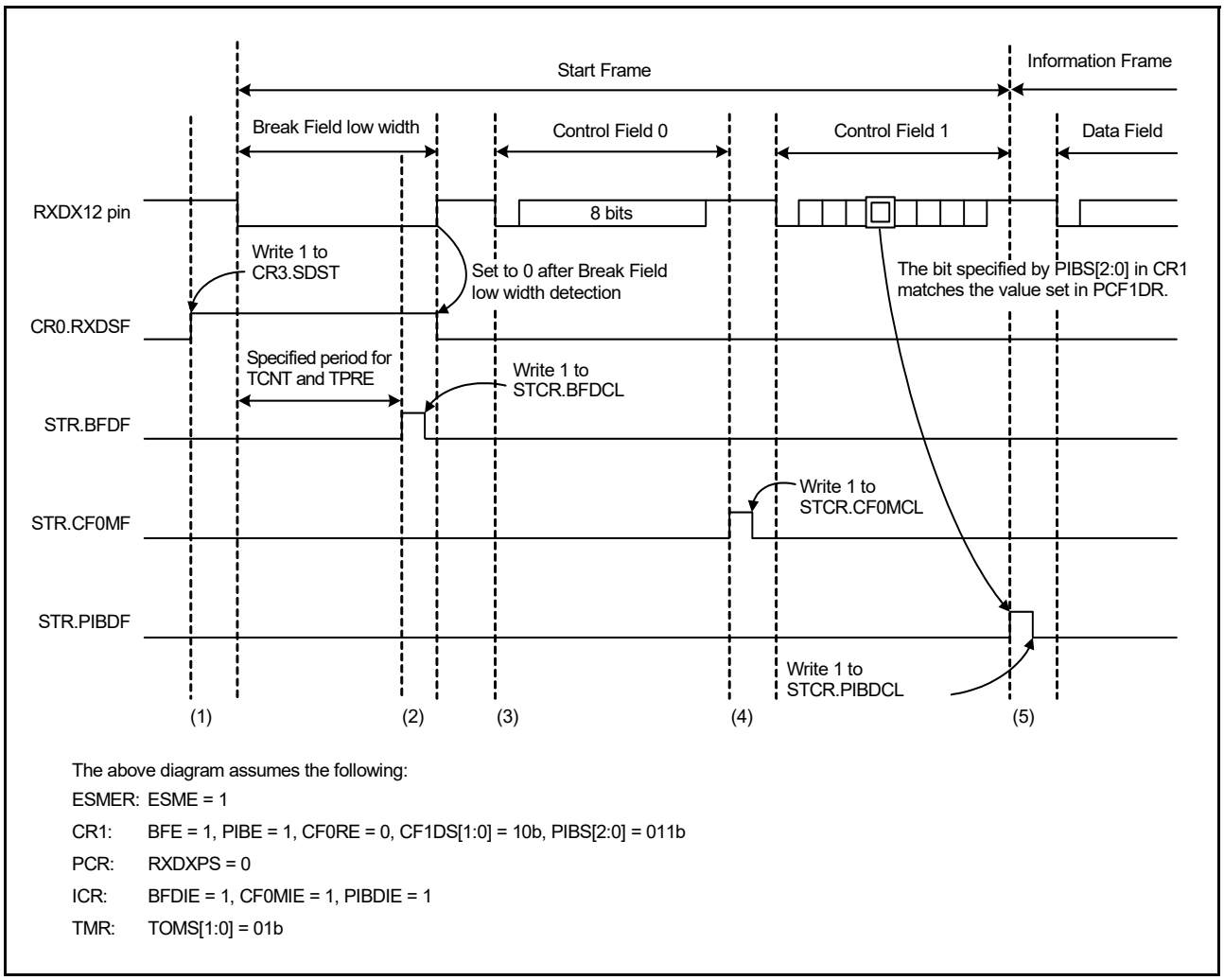


Figure 33.68 Example of Operations When Receiving a Start Frame While the CR1.PIBE Bit is 1

Downloaded from Arrow.com.

33.10.4 Detection of Bus Collisions

Detection of bus collisions operate for cases where output of the Break Field low width and transmission of data are in progress when the ESMER.ESME bit and the SCI.TE bit are set to 1.

Figure 33.69 shows an example of operations with bus collision detection. Signals output through TXDX12 and input through RXDX12 are sampled with the bus collision detection clock set with the CR2.BCCS[1:0] bits as the sampling clock, and the STR.BCDF flag is set to 1 if the signals fail to match three times in a row. An SCIX2 interrupt is also generated if the value of the ICR.BCDIE bit is 1.

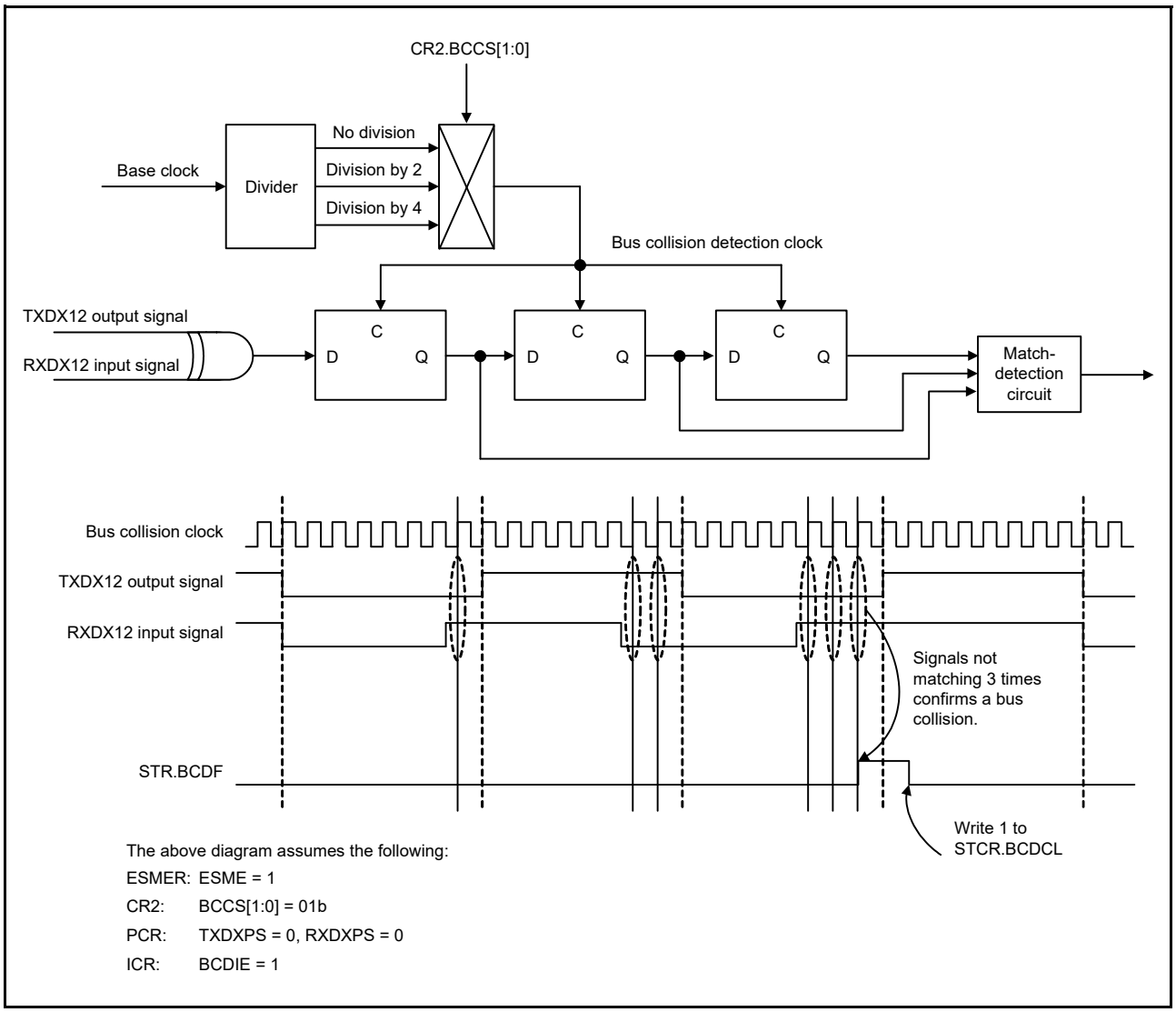


Figure 33.69 Example of Operations with Bus Collision Detection

Downloaded from Arrow.com.

33.10.5 Digital Filter for Input on the RXDX12 Pin

Signals input through the RXDX12 pin can be passed through a digital filter before they are conveyed to the internal circuits. The digital filter consists of three flip-flop circuit stages connected in series and a match-detecting circuit. The CR2.DFCS[2:0] bits select the sampling clock for the RXDX12 pin input signals. If the outputs of all three latches match, the given level is conveyed to subsequent circuits. If the levels do not match, the previous value is retained. In other words, levels are confirmed as being the signal if they are retained for at least three cycles of the sampling clock but judged to be noise rather than changes in the signal level if they change within three cycles of the sampling clock. Figure 33.70 shows an example of operations with the digital filter.

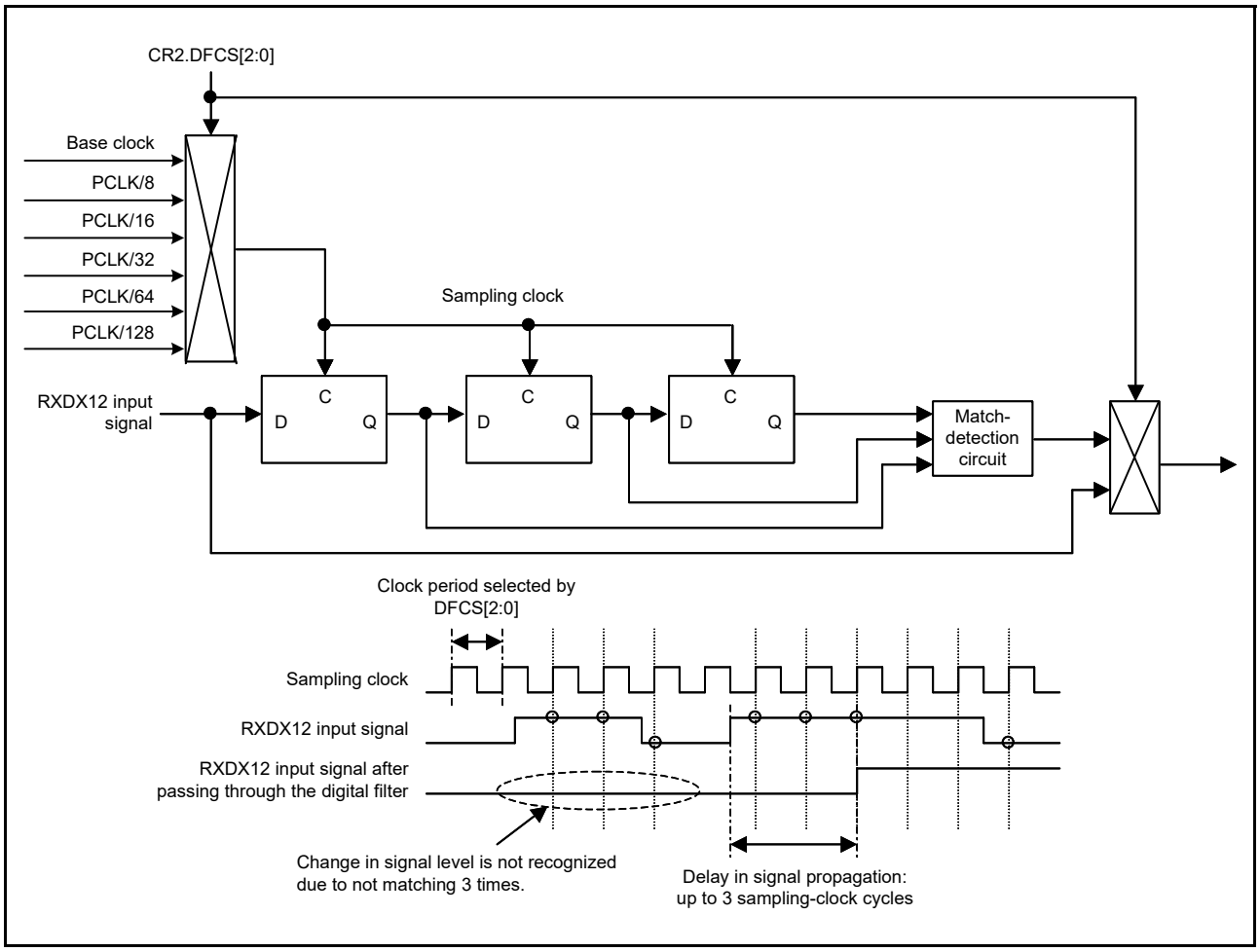


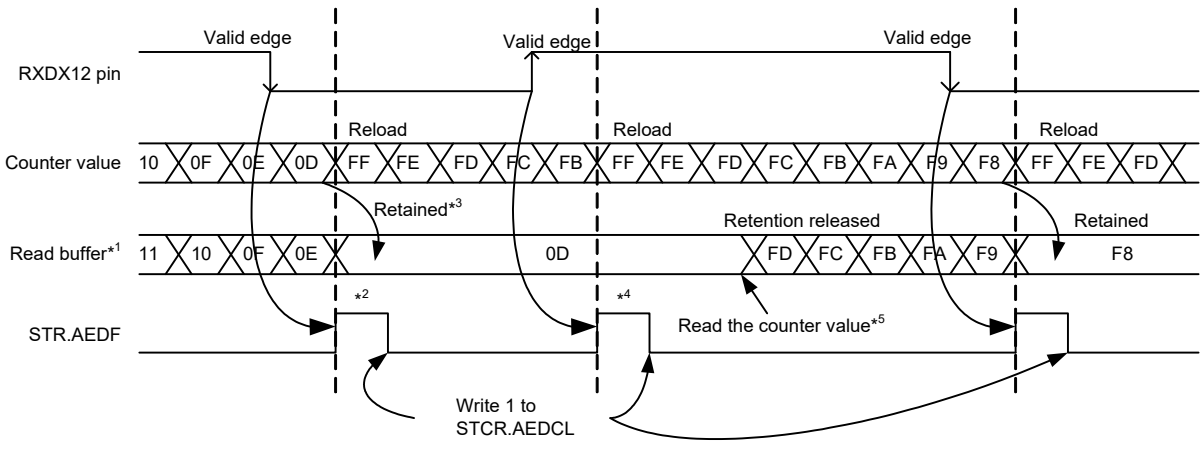
Figure 33.70 Example of Operations with the Digital Filter

Downloaded from Arrow.com.

33.10.6 Bit Rate Measurement

The bit rate measurement function measures the intervals between rising and falling edges and between falling and rising edges of the signal input from the RXDX12 pin. Figure 33.71 shows an example of operations for bit rate measurement.

(1) Writing 1 to the CR0.BRME bit enables bit rate measurement. Only set the BRME bit to 1 when you wish to proceed with bit rate measurement. Furthermore, bit rate measurement will not proceed during a Break Field, even if the BRME bit is set to 1.
(2) After detection of the Break Field low width, bit rate measurement starts when the level input on the RXDX12 pin becomes high.
(3) Once bit rate measurement has started, counter values from the timer are retained in the read buffers on the input of valid edges from the RXDX12 pin (rising and falling edges) and the counter is reloaded. An SCIX3 interrupt is also generated if the value of the ICR.AEDIE bit is 1. Retention by registers TCNT and TPRE is released by reading these registers.
(4) The bit rate as calculated from the values counted during intervals between valid edges can be used for adjusting the rate by changing the settings of the BRR register. To disable the bit rate measurement after a match with Control Field 1, write 0 to the CR0.BRME bit.



Note 1. The TCNT or TPRE register indicates the contents of the read buffers.
Note 2. The STR.AEDF flag becomes 1 on detection of a valid edge.
Note 3. When an effective edge is detected, the current value of the counter is stored in the read buffer.
Note 4. The value retained in the read buffers due to one valid edge is not updated to the new value even if a further valid edge is detected.
Note 5. Retention of a value in a read buffer due to a valid edge is released by reading the TCNT or TPRE register.

Figure 33.71 Example of Operations for Bit Rate Measurement

Downloaded from Arrow.com.

33.10.7 Selectable Timing for Sampling Data Received through RXDX12

The extended serial mode control section provides a way of adjusting the timing for the sampling of data received through the RXDX12 pin by setting the CR2.RTS[1:0] bits to select the rising edges of 8th, 10th, 12th, or 14th cycle of the base clock. If the value of the SEMR.ABCS bit is 1, the bits select the rising edges of 4th, 5th, 6th, or 7th cycle of the base clock. Figure 33.72 shows timing for the sampling of data received through RXDX12.

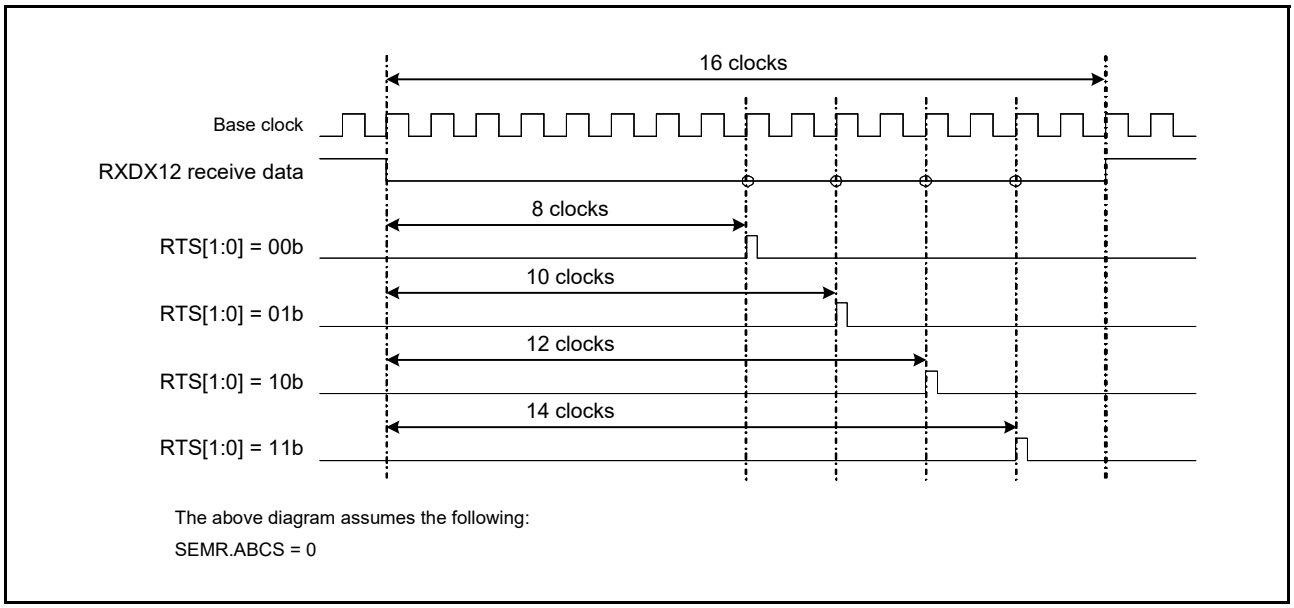


Figure 33.72 Timing for Sampling of Data Received through RXDX12

Downloaded from Arrow.com.

33.10.8 Timer

The timer has the following operating modes.

(1) Break Field Low Width Output Mode

This mode is for output through the TXDX12 pin of the low level over the Break Field low width at the transmission of a Start Frame. Setting the TMR.TOMS[1:0] bits to 10b switches operation to Break Field low width output mode. The TMR.TCSS[2:0] bits select the clock source for the counter. When the TCR.TCST bit is set to 1, the output on the TXDX12 pin goes to the low level and counting starts. When the timer underflows, the output on the TXDX12 pin goes to the high level and the STR.BFDF flag is set to 1. An SCIX0 interrupt is also generated if the value of the ICR.BFDIE bit is 1. When 0 is written to the TCR.TCST bit, counting stops after reloading of registers TPRE and TCNT. After output of the Break Field low width is completed, stop the timer before it underflows again. Figure 33.73 shows an example of operations in Break Field low width output mode.

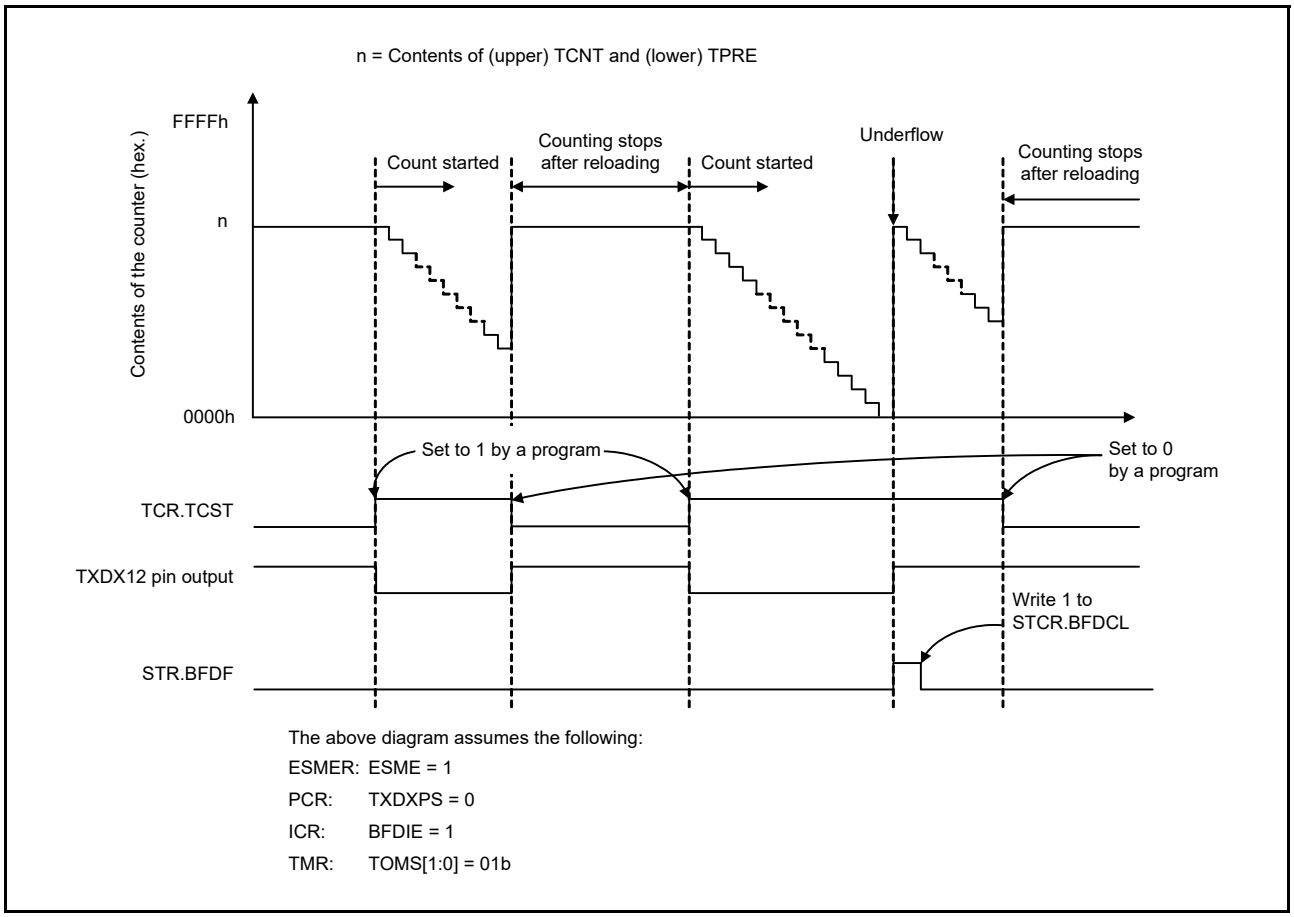


Figure 33.73 Example of Operations in Break Field Low Width Output Mode

Downloaded from Arrow.com.

(2) Break Field Low Width Determination Mode

This mode is for determining the Break Field low width in the input signal on the RXDX12 pin at the reception of a Start Frame. Setting the TMR.TOMS[1:0] bits to 01b switches operation to Break Field low width determination mode. The TMR.TCSS[2:0] bits select the clock source for the counter. When the TCR.TCST bit is set to 1, the interface enters the Break Field low width determinable state. Determination starts when a low level is input from the RXDX12 pin. When a high level is then input on the RXDX12 pin, registers TPRE and TCNT are reloaded and the interface enters the Break Field low width determinable state. When the timer underflows during Break Field low width determination, the STR.BFDF flag is set to 1. An SCIX0 interrupt is also generated if the value of the ICR.BFDIE bit is 1. If an underflow of the timer during data transfer cause a problem in the form of interrupt generation, stop the timer after Break Field low width determination. Figure 33.74 shows an example of operations in Break Field low width output mode.

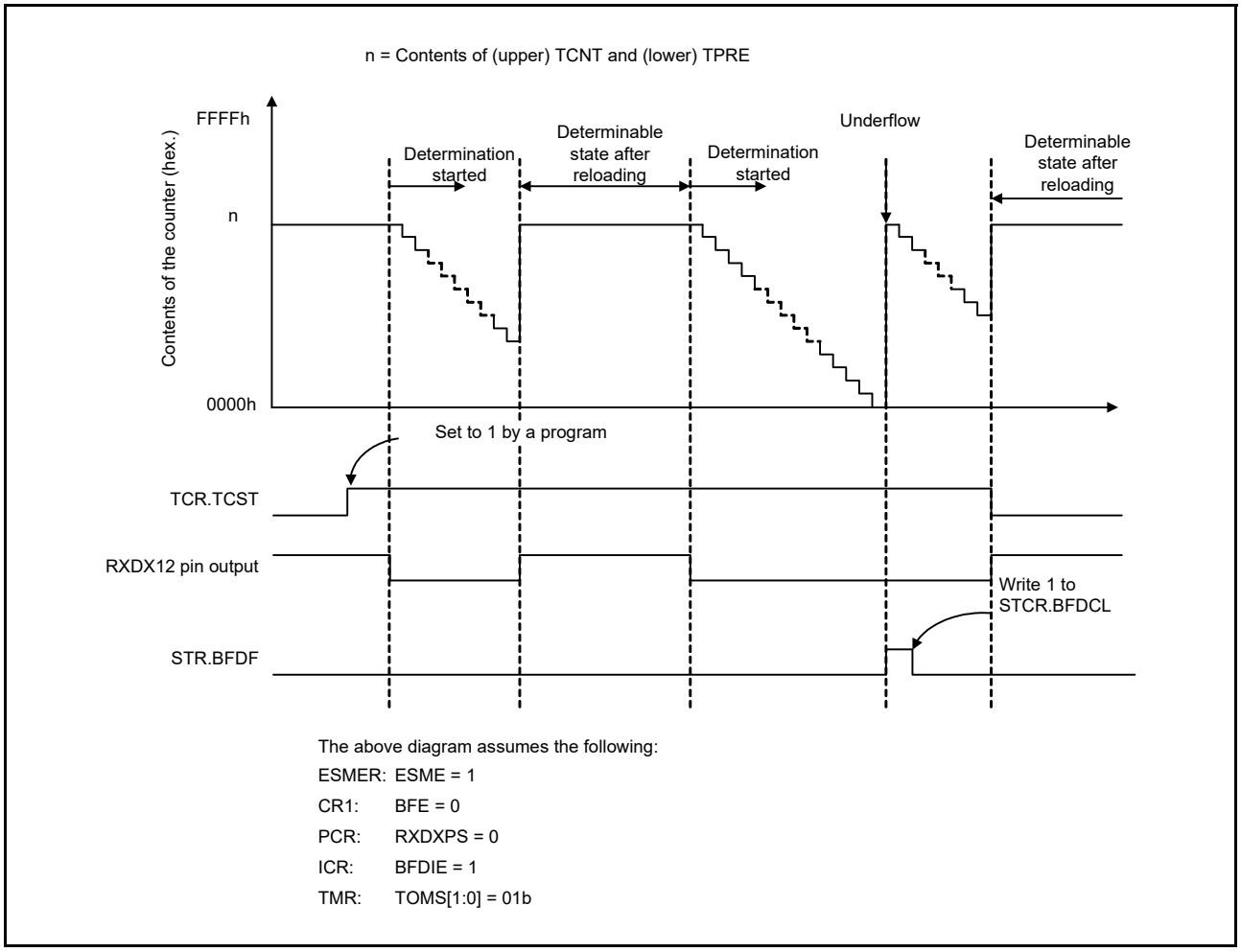


Figure 33.74 Example of Operations in Break Field Low Width Determination Mode

(3) Timer Mode

This mode is for counting cycles of the internal clock as the clock source. Setting the TMR.TOMS[1:0] bits to 00b switches operation to timer mode. The TMR.TCSS[2:0] bits select the clock source for the counter. Counting starts when 1 is written to the TCR.TCST bit and stops when 0 is written to the TCST bit. Registers TPRE and TCNT both count down. The TPRE register counts cycles of the clock source for counting, and underflows of the TPRE register provide the clock source for counting by the TCNT register. When the timer underflows, the STR.BFDF flag is set to 1. An SCIX0 interrupt is also generated if the value of the ICR.BFDIE bit is 1.

Downloaded from Arrow.com.

33.11 Noise Cancellation Function

Figure 33.75 shows the configuration of the noise filter used for noise cancellation. The noise filter consists of two stages of flip-flop circuits and a match-detection circuit. When the level on the pin matches in three consecutive samples taken at the set sampling interval, the matching level continues to be conveyed internally until the level on the pin again matches in three consecutive samples.

In asynchronous mode, the noise cancellation function can be applied on the RXDn input signal. The period of the base clock (1/16th of a bit-period when SEMR.ABCS = 0 and 1/8th of a bit-period when SEMR.ABCS = 1) is the sampling interval.

In simple I^2C mode, the noise cancellation function can be applied on the SSDAn and SSCLn input signals. The sampling clock is the clock signal produced by frequency-dividing the signal from the clock source for the internal baud-rate generator by one, two, four, or eight as selected by the setting of the SNFR.NFCS[2:0] bits.

If the base clock is stopped with the noise filter enabled and then the clock input is started again, the noise filter operation resumes from where the clock was stopped. If SCR.TE and SCR.RE are set to 0 during base clock input, all of the noise filter flip-flop values are initialized to 1. Accordingly, if the input data is 1 when reception operation resumes, it is determined that a level match is detected and is conveyed to the internal signal. When the level being input corresponds to 0, the initial output of the noise filter is retained until the level matches in three consecutive samples.

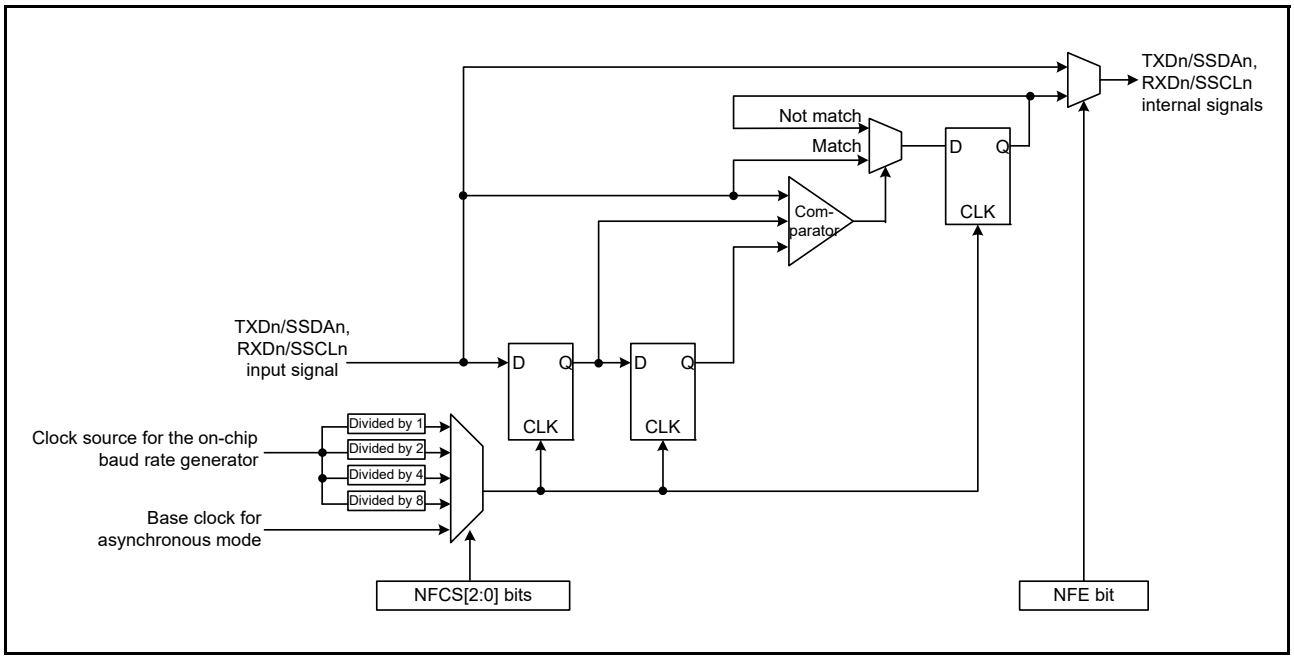


Figure 33.75 Block Diagram of Digital Noise Filter

Downloaded from Arrow.com.

33.12 Interrupt Sources

33.12.1 Buffer Operations for TXI and RXI Interrupts

If the conditions for a TXI and RXI interrupt are satisfied while the interrupt status flag in the interrupt controller is 1, the SCI does not output the interrupt request but retains it internally (with a capacity for retention of one request per source). When the value of the interrupt status flag in the interrupt controller becomes 0, the interrupt request retained within the SCI is output. The internally retained interrupt request is automatically discarded once the actual interrupt is output. Clearing of the corresponding interrupt enable bit (the TIE or RIE bit in the SCR register) can also be used to discard an internally retained interrupt request.

33.12.2 Interrupts in Asynchronous Mode, Clock Synchronous Mode, and Simple SPI Mode

Table 33.31 lists interrupt sources in asynchronous mode, clock synchronous mode, and simple SPI mode. Individual interrupt sources can be enabled or disabled with the enable bits in the SCR register.

If the SCR.TIE bit is 1, a TXI interrupt request is generated when transmit data is transferred from the TDR or TDRL register*1 to the TSR register. A TXI interrupt request can also be generated by setting the SCR.TE bit to 1 after setting the SCR.TIE bit to 1 or by using a single instruction to set the SCR.TE and SCR.TIE bit to 1 at the same time. A TXI interrupt request can activate the DTC or DMAC to handle data transfer.

A TXI interrupt request is not generated by setting the SCR.TE bit to 1 while the setting of the SCR.TIE bit is 0 or by setting the SCR.TIE bit to 1 while the setting of the SCR.TE bit is 1.*2

Note that setting the SCR.TE bit to 0 while the SCR.TIE bit is 1 leads to the generation of a TXI interrupt.

When new data is not written by the time of transmission of the last bit of the current transmit data and the setting of the SCR.TEIE bit is 1, the SSR.TEND flag becomes 1 and a TEI interrupt request is generated. Furthermore, when the setting of the SCR.TE bit is 1, the SSR.TEND flag retains the value 1 until further transmit data are written to the TDR or TDRL register*1, and setting the SCR.TEIE bit to 1 leads to the generation of a TEI interrupt request.

Writing data to the TDR or TDRL register*1 leads to clearing of the SSR.TEND flag and, after a certain time, discarding of the TEI interrupt request.

If the SCR.RIE bit is 1, an RXI interrupt request is generated when received data is stored in the RDR register. An RXI interrupt request can activate the DTC or DMAC to handle data transfer.

Setting of any from among the ORER, FER, and PER flags in the SSR register to 1 while the SCR.RIE bit is 1 leads to the generation of an ERI interrupt request. An RXI interrupt request is not generated at this time. Clearing all three flags (ORER, FER, and PER) leads to discarding of the ERI interrupt request.

Note 1. In the case where asynchronous mode and 9-bit data length are selected

Note 2. To temporarily disable TXI interrupts at the time of transmission of the last of the data and so on when you wish a new round of transmission to start after handling of the transmission-completed interrupt, control disabling and enabling of the interrupt by using the interrupt request enable bit in the interrupt controller rather than using the SCR.TIE bit. This can prevent the suppression of TXI interrupt requests in the transfer of new data.

Table 33.31 Interrupt Sources

Name	Interrupt Source	Interrupt Flag	DTC Activation	DMAC Activation
ERI	Receive error	ORER, FER, or PER	Not possible	Not possible
RXI	Receive data full	RDRF	Possible	Possible
TXI	Transmit data empty	TDRE	Possible	Possible
TEI	Transmit end	TEND	Not possible	Not possible

Downloaded from Arrow.com.

33.12.3 Interrupts in Smart Card Interface Mode

Table 33.32 lists interrupt sources in smart card interface mode. A transmit end interrupt (TEI) request cannot be used in this mode.

Table 33.32 SCI Interrupt Sources

Name	Interrupt Source	Interrupt Flag	DTC Activation	DMAC Activation
ERI	Receive error or error signal detection	ORER, PER, or ERS	Not possible	Not possible
RXI	Receive data full	—	Possible	Possible
TXI	Transmit data empty	TEND	Possible	Possible

Data transmission/reception using the DTC or DMAC is also possible in smart card interface mode, similar to in the normal SCI mode. In transmission, when the SSR.TEND flag is set to 1, a TXI interrupt request is generated. This TXI interrupt request activates the DTC or DMAC allowing transfer of transmit data if the TXI request is specified beforehand as a source of DTC or DMAC activation. The TEND flag is automatically set to 0 when the DTC or DMAC transfers the data.

If an error occurs, the SCI automatically retransmits the same data. During the retransmission, the TEND flag is kept to 0 and the DTC or DMAC is not activated. Therefore, the SCI and DTC or DMAC automatically transmit the specified number of bytes, including retransmission in the case of error occurrence. However, the SSR.ERS flag is not automatically cleared to 0 at error occurrence. Therefore, the ERS flag must be cleared by previously setting the SCR.RIE bit to 1 to enable an ERI interrupt request to be generated at error occurrence.

When transmitting/receiving data using the DTC or DMAC, be sure to make settings to enable the DTC or DMAC before making SCI settings. For DTC or DMAC settings, refer to section 19, Data Transfer Controller (DTCa) and section 18, DMA Controller (DMACA).

In reception, an RXI interrupt request is generated when receive data is set to the RDR register. This RXI interrupt request activates the DTC or DMAC allowing transfer of receive data if the RXI request is specified beforehand as a source of DTC or DMAC activation. If an error occurs, the error flag is set. Therefore, the DTC or DMAC is not activated and an ERI interrupt request is issued to the CPU instead; the error flag must be cleared.

Downloaded from Arrow.com.

33.12.4 Interrupts in Simple I2C Mode

The interrupt sources in simple I2C mode are listed in Table 33.33. The STI interrupt is allocated to the transmit end interrupt (TEI) request. The receive error interrupt (ERI) request cannot be used.

The DTC or DMAC can also be used to handle transfer in simple I2C mode.

When the value of the SIMR2.IICINTM bit is 1, an RXI request will be generated on the falling edge of the SSCLn signal for the eighth bit. If the RXI has been set up as an activating request for the DTC or DMAC beforehand, the RXI request will activate the DTC or DMAC to handle transfer of the received data. Furthermore, a TXI request is generated on the falling edge of the SSCLn signal for the ninth bit (acknowledge bit). If the TXI has been set up as an activating request for the DTC or DMAC beforehand, the TXI request will activate the DTC or DMAC to handle transfer of the transmit data.

When the value of the SIMR2.IICINTM bit is 0, an RXI request (ACK detection) if the input on the SSDAn pin is at the low level or a TXI request (NACK detection) if the input on the SSDAn pin is at the high level will be generated on the rising edge of the SSCLn signal for the ninth bit (acknowledge bit). If the RXI has been set up as an activating request for the DTC or DMAC beforehand, the RXI request will activate the DTC or DMAC to handle transfer of the received data. Also, if the DTC or DMAC is used for data transfer in reception or transmission, be sure to set up and enable the DTC or DMAC before setting up the SCI.

When the IICSTAREQ, IICRSTAREQ, and IICSTPREQ bits in the SIMR3 register are used to generate a start condition, restart condition, or stop condition, the STI request is issued when generation is complete.

Table 33.33 SCI Interrupt Sources

Name	Interrupt Source		Interrupt Flag	DTC Activation	DMAC Activation
	IICINTM bit = 0	IICINTM bit = 1			
RXI	ACK detection	Reception	—	Possible	Possible
TXI	NACK detection	Transmission	—	Possible*1	Possible*1
STI	Completion of generation of a start, restart, or stop condition		IICSTIF	Not possible	Not possible

Note 1. Activation of the DTC or DMAC is only possible when the SIMR2.IICINTM bit is 1 (use reception and transmission interrupts).

Downloaded from Arrow.com.

33.12.5 Interrupt Requests from the Extended Serial Mode Control Section

The extended serial mode control section has a total of six types of interrupt request for generating the SCIX0 interrupt (Break Field low width detected), SCIX1 interrupt (Control Field 0 match, Control Field 1 match, priority interrupt bit detected), SCIX2 interrupt (bus collision detected), and SCIX3 interrupt (valid edge detected). When any of the interrupt factors is generated, the corresponding status flag is set to 1. Details of all of the interrupt requests are listed in Table 33.34.

Table 33.34 Interrupt Sources of the Extended Serial Mode Control Section

Interrupt Request	Status Flag	Interrupt Factors
SCIX0 interrupt (Break Field low width detected)	BFDF	• Detection of a Break Field low width longer than the interval corresponding to the timer setting • Completion of the output of a Break Field low width over the interval corresponding to the timer setting • Underflow of the timer
SCIX1 interrupt (Control Field 0 match)	CF0MF	The data received in Control Field 0 matching the value set in the CF0DR register
SCIX1 interrupt (Control Field 1 match)	CF1MF	The data received in Control Field 1 matching the value set in the PCF1DR or SCF1DR register
SCIX1 interrupt (priority interrupt bit detected)	PIBDF	The value of the bit specified as the priority interrupt bit matching the value set in the PCF1DR register
SCIX2 interrupt (bus collision detected)	BCDF	The output level on the TXDX12 pin and the input level on the RXDX12 pin not matching on three consecutive cycles of the bus collision detection clock
SCIX3 interrupt (valid edge detected)	AEDF	Detection of a valid edge during bit rate measurement

Downloaded from Arrow.com.

33.13 Event Linking

By employing interrupt request signals as event signals, SCI5 is able to provide linked operation through the event link controller (ELC) for modules selected in advance.
Event signals can be output regardless of the values of the corresponding interrupt request enable bits.

(1) Error (receive error, error signal detected) event output

- Indicates abnormal termination due to a parity error during reception in asynchronous mode.
- Indicates abnormal termination due to a framing error during reception in asynchronous mode.
- Indicates abnormal termination due to an overrun error during reception.
- Indicates detection of the error signal during transmission in smart card interface mode.

(2) Receive data full event output

- Indicates that received data have been set in the receive data register (RDR or RDRL).
- Indicates that ACK has been detected if the SIMR2.IICINTM bit is 0 in simple I^2C mode.
- Indicates that the 8th-bit SSCL5 falling edge has been detected if the SIMR2.IICINTM bit is 1 in simple I^2C mode.
- When the SIMR2.IICINTM bit is 1 during master transmission in simple I^2C mode, set the event link controller (ELC) so that receive data full events are not used.

(3) Transmit data empty event output

- Indicates that the SCR.TE bit has been changed from 0 to 1.
- Indicates that transmit data have been transferred from the transmit data register (TDR or TDRL) to the transmit shift register (TSR).
- Indicates that transmission has been completed in smart card interface mode.
- Indicates that NACK has been detected if the SIMR2.IICINTM bit is 0 in simple I^2C mode.
- Indicates that the ninth-bit SSCL5 falling edge has been detected if the SIMR2.IICINTM bit is 1 in simple I^2C mode.

(4) Transmit end event output

- Indicates the completion of transmission.
- Indicates that the starting condition, resumption condition, or termination condition has been generated in simple I^2C mode.

Downloaded from Arrow.com.

33.14 Usage Notes

33.14.1 Setting the Module Stop Function

Module stop control register B (MSTPCRB) and module stop control register C (MSTPCRC) are used to stop and start SCI operations. With the value after a reset, SCI operations are stopped. Register access is enabled by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

33.14.2 Break Detection and Processing

When a framing error is detected, a break can be detected by reading the RXDn pin value directly. In a break, the input from the RXDn pin becomes all 0s, and so the SSR.FER flag is set to 1 (framing error has occurred), and the SSR.PER flag may also be set to 1 (parity error has occurred). When the SEMR.RXDESEL bit is 0, the SCI continues the receive operation even after a break is received. Therefore, note that even if the FER flag is set to 0 (no framing error occurred), it will be set to 1 again. When the SEMR.RXDESEL bit is 1, the SCI sets the SSR.FER flag to 1 and stops receiving operation until a start bit of the next data frame is detected. If the SSR.FER flag is set to 0 at this time, the SSR.FER flag retains 0 during the break. When the RXDn pin becomes high and the break ends, detecting the beginning of the start bit at the first falling edge of the RXDn pin allows the SCI to start the receiving operation.

33.14.3 Mark State and Sending Breaks

When the SCR.TE bit is 0 (serial transmission is disabled), the TXDn pin becomes high-impedance. To forcibly set the TXDn pin to mark or space state while the TE bit is 0, set the I/O port associated registers and switch the TXDn pin to general output port.

For holding the communication line in the mark (“1”) state until the TE bit is set to 1 (serial transmission is enabled), set the corresponding bit in the PODR register to 1 for high output from general output port. To start communications, set the TE bit to 1 and then the corresponding bit in the PMR register to 1.

To send a break (the space state for longer than a certain period of time) while data transmission, set the corresponding bit in the PODR register to 0 (low output), and set the corresponding bit in the PMR register to 0 (general I/O port). Then set the TE bit to 0 if necessary. When the TE bit is set to 0, the transmitter is initialized regardless of the current transmit status.

33.14.4 Receive Error Flags and Transmit Operations (Clock Synchronous Mode and Simple SPI Mode)

Transmission cannot be started when a receive error flag (SSR.ORER) is set to 1, even if data is written to the TDR register. Be sure to set the receive error flags to 0 before starting transmission. Note also that the receive error flags cannot be set to 0 even if the SCR.RE bit is set to 0 (serial reception is disabled).

33.14.5 Writing Data to the TDR Register

Data can be written to registers TDR, TDRH, and TDRL. However, if new data is written to registers TDR, TDRH, and TDRL when transmit data is remaining in registers TDR, TDRH, and TDRL, the previous data in registers TDR, TDRH, and TDRL is lost because it has not been transferred to the TSR register yet. Be sure to write transmit data to registers TDR, TDRH, and TDRL in the TXI interrupt request handling routine.

Downloaded from Arrow.com.

33.14.6 Restrictions on Clock Synchronous Transmission (Clock Synchronous Mode and Simple SPI Mode)

When the external clock source is used as a synchronization clock, the following restrictions apply.

(1) Start of transmission

Update the TDR register by the CPU, DMAC, or DTC and wait for at least five PCLK cycles before allowing the transmit clock to be input (refer to Figure 33.76).

(2) Continuous transmission

(a) Write the next transmit data to the TDR or TDRL register before the falling edge of the transmit clock (bit 7) (refer to Figure 33.76).

(b) When updating the TDR register after bit 7 has started to transmit, update the TDR register while the synchronization clock is in the low-level period, and set the high-level width of the transmit clock (bit 7) to four PCLK cycles or longer (refer to Figure 33.76).

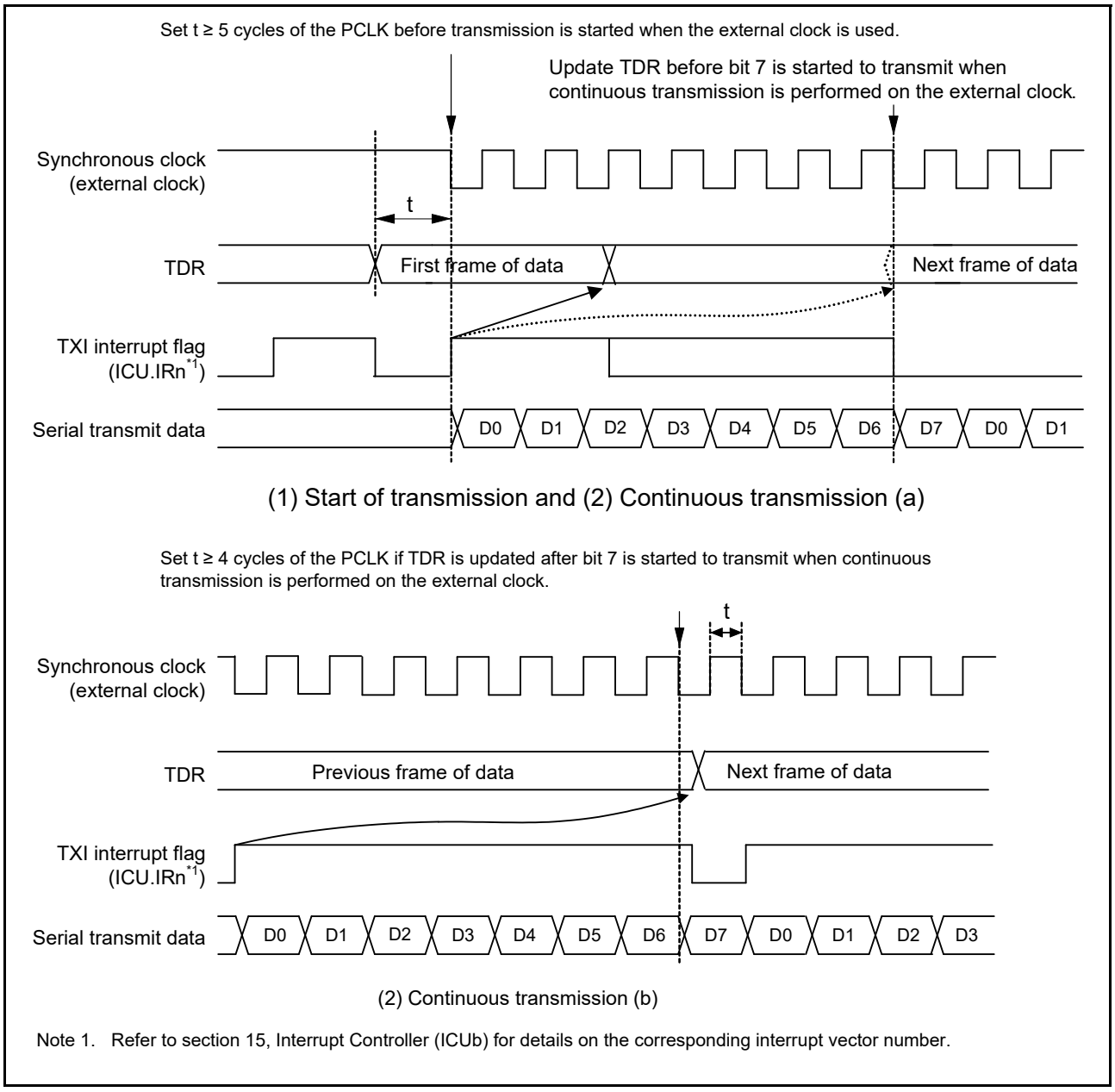


Figure 33.76 Restrictions on Use of External Clock in Clock Synchronous Transmission

Downloaded from Arrow.com.

33.14.7 Restrictions on Using DMAC or DTC

When using the DMAC or DTC to read the RDR, RDRH, and RDRL registers, be sure to set the receive data full interrupt (RXI) as the activation source of the relevant SCI.

33.14.8 Notes on Starting Transfer

At the point where transfer starts when the interrupt status flag (IRn.IR flag) in the interrupt controller is 1, follow the procedure below to clear interrupt requests before permitting operations (by setting the SCR.TE or SCR.RE bit to 1). For details on the interrupt status flag, refer to section 15, Interrupt Controller (ICUb).

- Confirm that transfer has stopped (the setting of the SCR.TE or SCR.RE bits is 0).
- Set the corresponding interrupt enable bit (SCR.TIE or SCR.RIE) to 0.
- Read the corresponding interrupt enable bit (SCR.TIE or SCR.RIE bit) to check that it has become 0.
- Set the interrupt status flag (IRn.IR flag) in the interrupt controller to 0.

33.14.9 SCI Operations during Low Power Consumption State

(1) Transmission

When making settings for the module stopped state or in transitions to software standby, stop operations (by setting the TIE, TE, and TEIE bits in the SCR register to 0) after switching the TXDn pin to the general I/O port pin function. Setting the TE bit to 0 resets the TSR register and the SSR.TEND flag. Depending on the port settings, output pins may output the level before a transition to the low power consumption state is made after release from the module stopped state or software standby mode. When transitions to these states are made during transmission, the data being transmitted become indeterminate.

To transmit data in the same transmit mode after cancellation of the low power consumption state, set the TE bit to 1, read the SSR register, and write data to the TDR register sequentially to start data transmission. To transmit data with a different transmit mode, initialize the SCI first.

Figure 33.77 shows a sample flowchart for transition to software standby mode during transmission. Figure 33.78 and Figure 33.79 show the port pin states during transition to software standby mode.

Before specifying the module stop state or making a transition to software standby mode from the transmit mode using DTC/DMA transfer, stop the transmit operations (TE = 0). To start transmission after cancellation using the DTC/DMAC, set the TE and TIE bits to 1. The TXI interrupt flag is set to 1 and transmission starts using the DTC/DMAC.

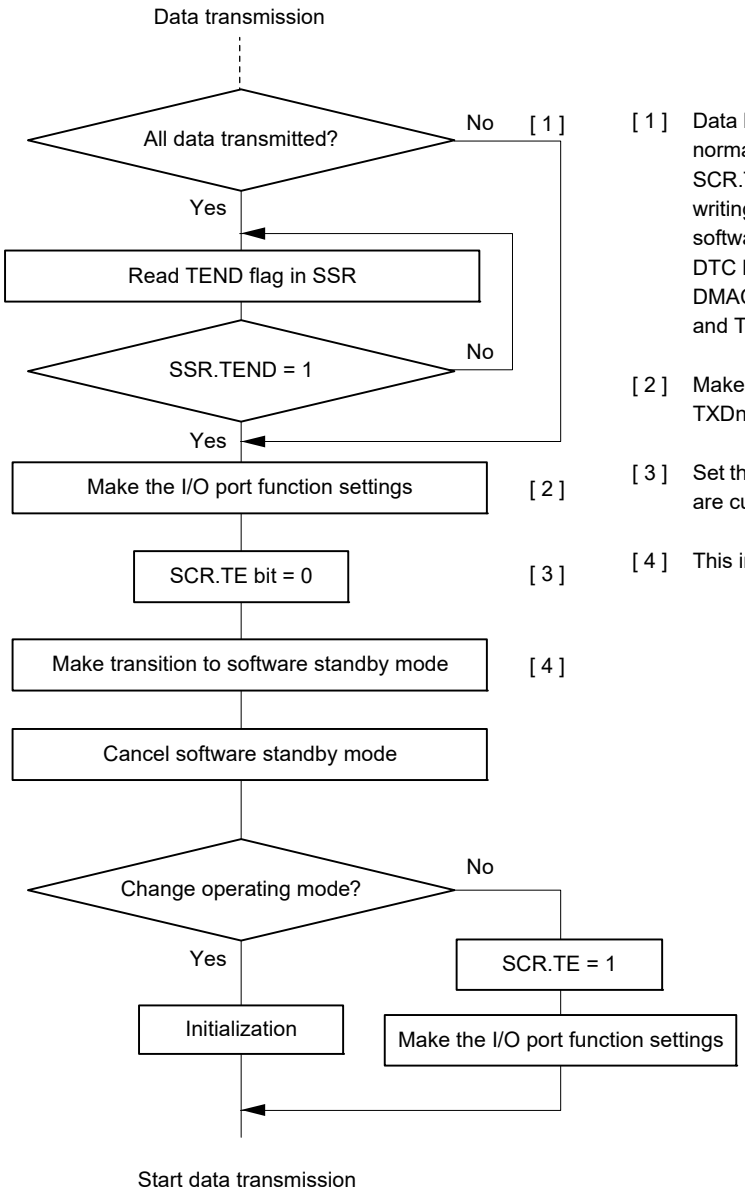
(2) Reception

Before specifying the module stop state or making a transition to software standby mode, stop the receive operations (SCR.RE = 0). If transition is made during data reception, the data being received will be invalid.

To receive data in the same receive mode after cancellation of the low power consumption state, set the RE bit to 1, and then start reception. To receive data in a different receive mode, initialize the SCI first.

Figure 33.80 shows a sample flowchart for transition to software standby mode during reception.

Downloaded from Arrow.com.



[1] Data being transmitted is lost halfway. Data can be normally transmitted from the CPU by setting the SCR.TE bit to 1, reading the SSR register, and writing data to the TDR register after canceling software standby mode. However, if the DMAC or DTC has been activated, the data remaining in the DMAC or DTC will be transmitted when both the TE and TIE bits in the SCR register are set to 1.

[2] Make the I/O port function settings to switch the TXDn pin to operate as a general I/O port.

[3] Set the TIE and TEIE bits in the SCR register if they are currently set to 1.

[4] This includes the setting for the module stop state.

Figure 33.77 Example of Flowchart for Transition to Software Standby Mode during Transmission

Downloaded from Arrow.com.

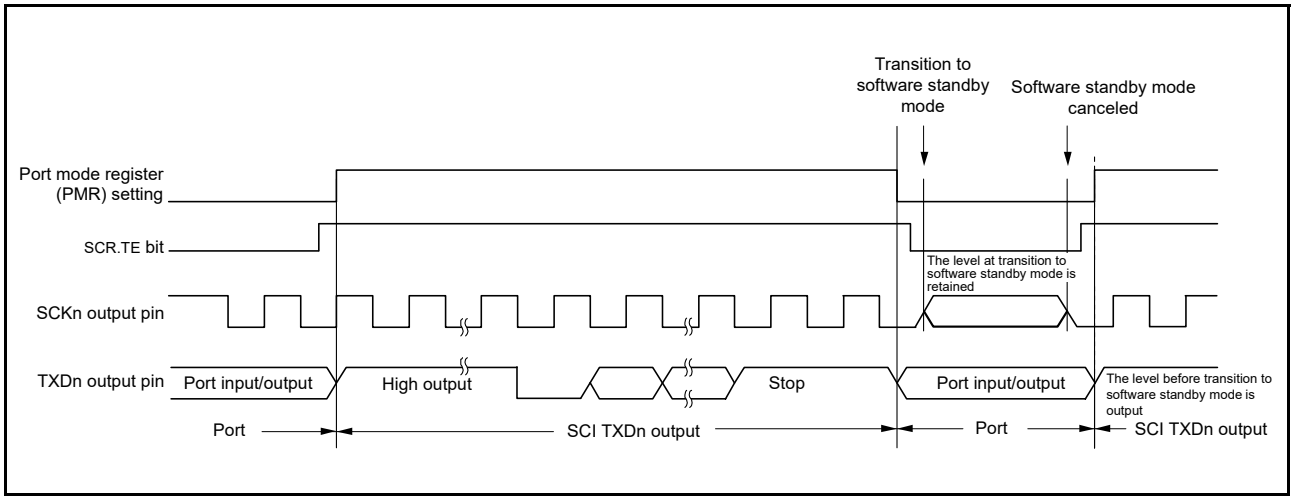


Figure 33.78 Port Pin States during Transition to Software Standby Mode (Internal Clock, Asynchronous Transmission)

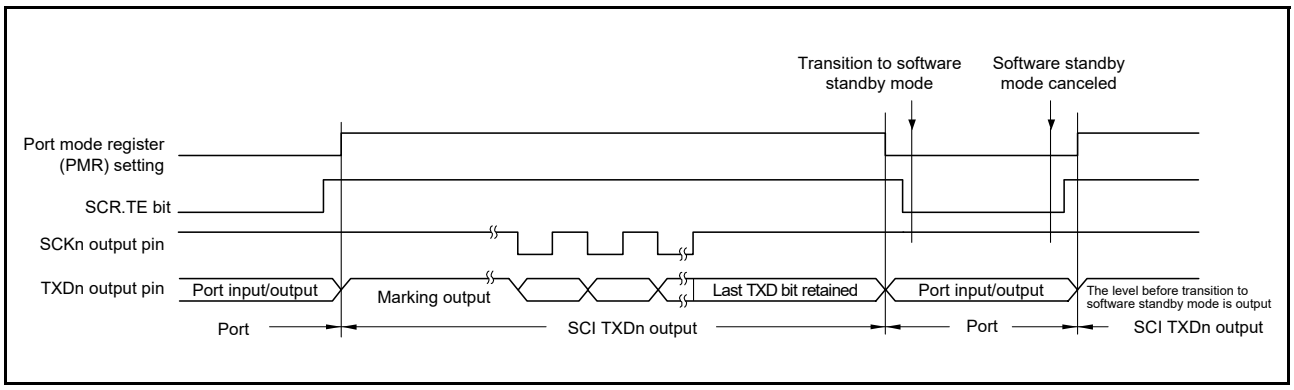


Figure 33.79 Port Pin States during Transition to Software Standby Mode (Internal Clock, Clock Synchronous Transmission)

Downloaded from Arrow.com.

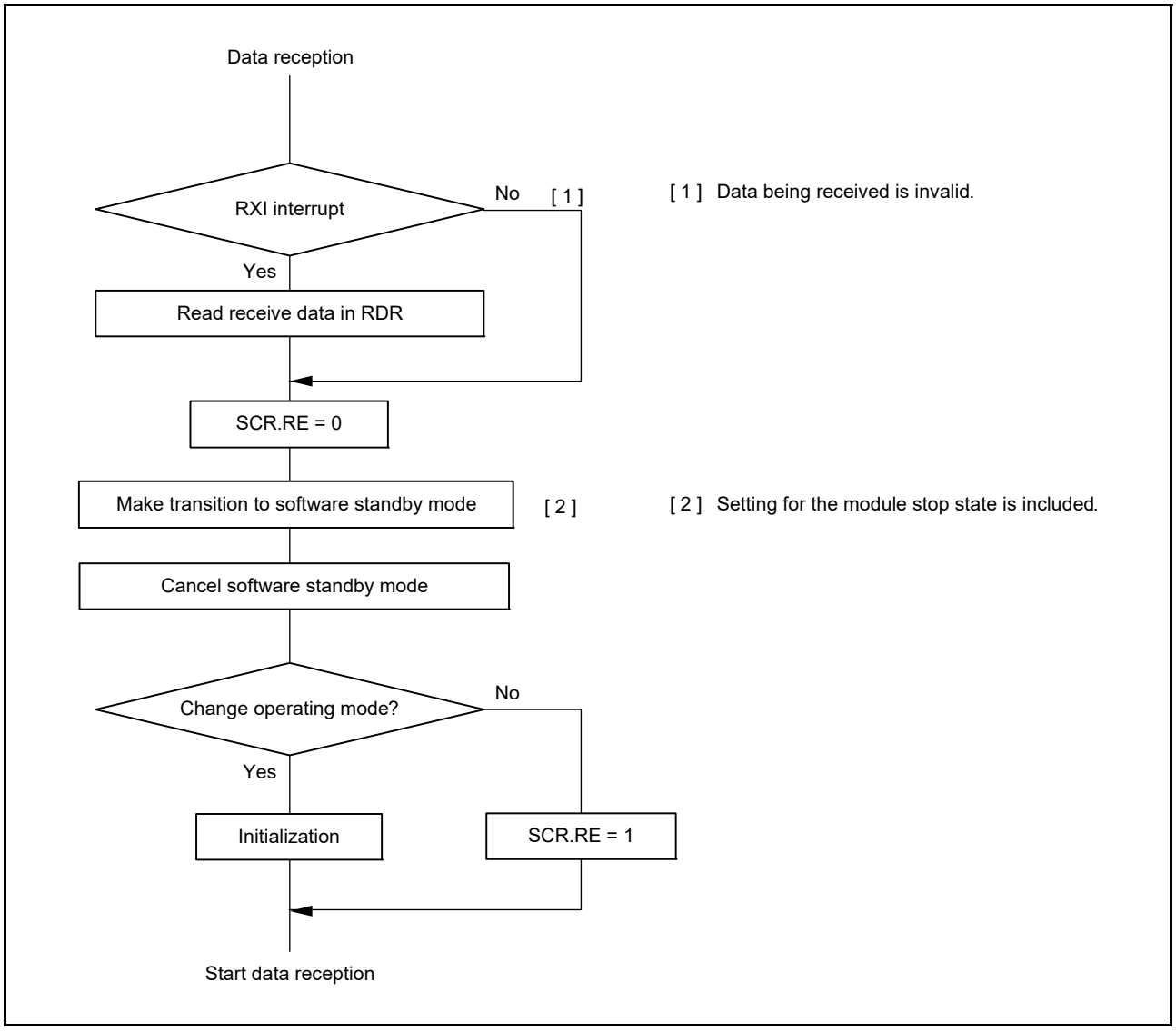


Figure 33.80 Example of Flowchart for Transition to Software Standby Mode during Reception

33.14.10 External Clock Input in Clock Synchronous Mode and Simple SPI Mode

In clock synchronous mode and simple SPI mode, the external clock SCKn must be input as follows:
High-pulse period, low-pulse period = 2 PCLK cycles or more, period = 6 PCLK cycles or more

Downloaded from Arrow.com.

33.14.11 Limitations on Simple SPI Mode

(1) Master Mode

- Use a resistor to pull up or pull down the clock line matching the initial settings for the transfer clock set by the SPMR.CKPH and CKPOL bits when the SPMR.SSE bit is 1.
  This prevents the clock line from being placed in the high-impedance state when the SCR.TE bit is set to 0 or unexpected edges from being generated on the clock line when the SCR.TE bit is changed from 0 to 1. When the SPMR.SSE bit is 0 in single master mode, pulling up or pulling down the clock line is not necessary because the clock line is not placed in the high-impedance state even when the SCR.TE bit is set to 0.
- In the case of the setting for clock delay (SPMR.CKPH bit is 1), the receive data full interrupt (RXI) is generated before the final clock edge on the SCKn pin as indicated in Figure 33.81. If the TE and RE bits in the SCR register become 0 at this time before the final edge of the clock signal on the SCKn pin, the SCKn pin is placed in the high-impedance state, so the width of the last clock pulse of the transfer clock is shortened. Furthermore, an RXI interrupt may lead to the input signal on the SSn# pin of a connected slave going to the high level before the final edge of the clock signal on the SCKn pin, leading to incorrect operation of the slave.
- In a multi-master configuration, take care because the SCKn pin output becomes high-impedance while the input on the SSn# pin is at the low level if a mode fault error occurs as the current character is being transferred, stopping supply of the clock signal to the connected slave. Remake the settings for the connected slave to avoid misaligned bits when transfer is restarted.

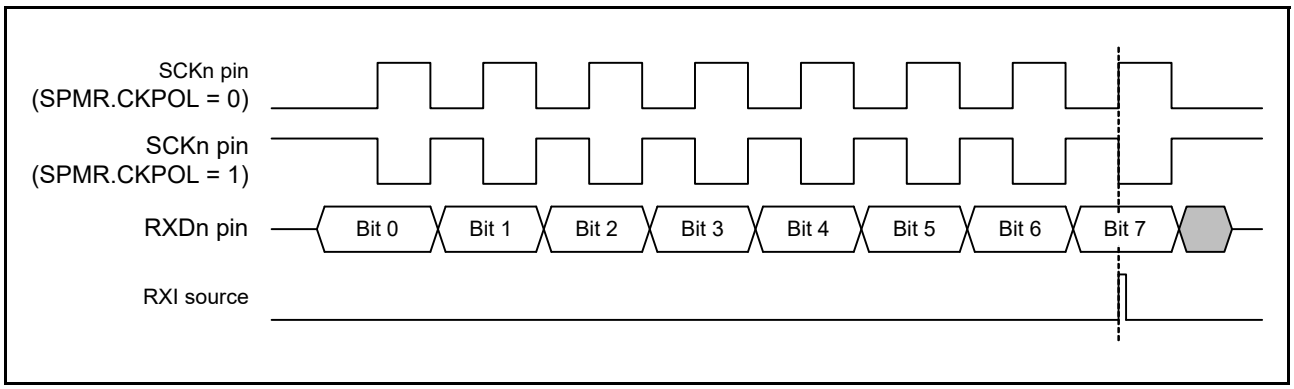


Figure 33.81 Timing of the RXI Interrupt in Simple SPI Mode (with Clock Delay)

(2) Slave Mode

- Secure at least five cycles of the PCLK from writing transmit data in the TDR register to start of the external clock input. Also secure at least five cycles of the PCLK from input of low level on the SSn# pin to start of the external clock input.
- Provide an external clock signal from the master the same as the transmit/receive data length.
- Control the input on the SSn# pin before the start and after the end of data transfer.
- When the level being input on the SSn# pin is to be changed from low to high while the current character is being transferred, set the TE and RE bits in the SCR register to 0 and, after remaking the settings, restart transfer of the first byte.

Downloaded from Arrow.com.

33.14.12 Limitation 1 on Usage of the Extended Serial Mode Control Section

When the PCR.SHARPS bit is set to 1, output on the TXDX12/RXDX12 pin is only possible when the following conditions apply.

- The timer is in Break Field low width output mode and the value of the TCR.TCST bit is 1 (when the TCST bit is set to 1, the high level continues to be output for up to one cycle of the clock source for counting by the timer counter before output of the low level)
- The value of the SCR.TE bit is 1.

33.14.13 Limitation 2 on Usage of the Extended Serial Mode Control Section

The TXI, RXI, ERI, and TEI interrupt requests are generated even if the extended serial mode is enabled. However, the RXI interrupt should not be enabled during reception of a Start Frame because the extended serial mode control section uses the receive data full signal.
To use the RXI interrupts during a reception of the Information Frame, use it in accordance with one of the following procedures. When a receive error is detected, clear the receive error flag and initialize the extended serial mode control section.

(1) Set the SCR.RIE bit to 0 to disable the output of interrupt requests. Check the error flags in the SSR register on completion of the reception of a Start Frame, because an ERI interrupt is not generated if a receive error occurs. After reception of the Start Frame is completed, set the SCR.RIE bit to 1 by the time the first byte of the Information Frame is received.

(2) Set the SCR.RIE bit to 1 to disable RXI interrupts and enable ERI interrupts for ICU.
Clear the IRn.IR flag to enable the acceptance of RXI interrupts by ICU by the time the first byte of the Information Frame is received after the completion of Start Frame reception.

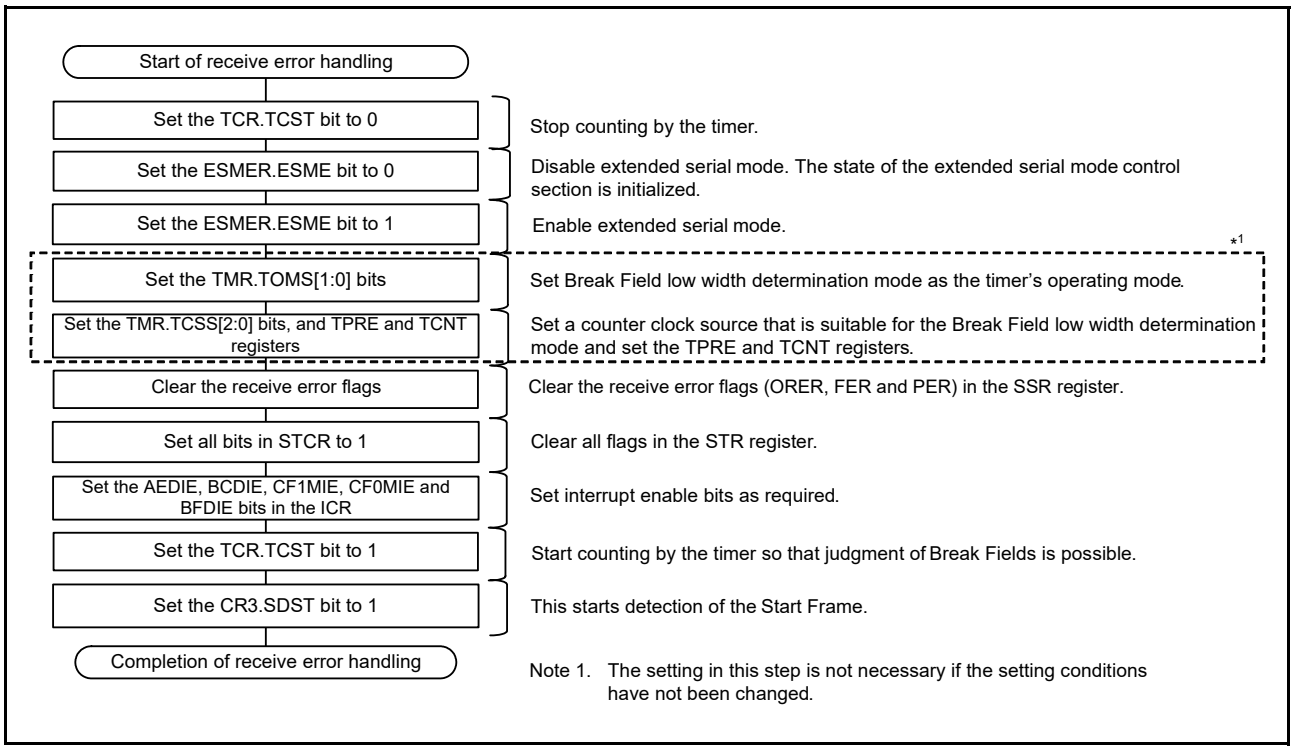


Figure 33.82 Example of Flowchart for Receive Error Handling (during Reception of the Start Frame)

Downloaded from Arrow.com.

33.14.14 Note on Transmit Enable Bit (TE Bit)

When setting the pin function to “TXDn” while the SCR.TE bit is 0 (serial transmission is disabled) or setting the TE bit to 0 while the pin function is “TXDn”, output of the TXDn pin becomes high-impedance.
Prevent the TXDn line from becoming high-impedance by any of the following ways:

(1) Connect a pull-up resistor to the TXDn line.
(2) Set the TE bit to 1*1 before changing the pin function to “TXDn”. Before setting the TE bit to 0, change the pin function to “general-purpose I/O port” and drive the pin high or low.

Note 1. An interrupt is generated when the TE bit is set to 1 while the TXI interrupt is enabled. If this creates a problem, change the pin function to “TXDn” first, and then set the corresponding ICU.IERm.IENj bit to 1.

33.14.15 Note on Stopping Reception When Using the RTS Function in Asynchronous Mode

One clock cycle of PCLK is required for the time from setting the SCR.RE bit to 0 to stopping the RTS signal generator in asynchronous mode.
When reading the RDR (or RDRL) register after setting the SCR.RE bit to 0, confirm that the RE bit has been set to 0 before reading the RDR (or RDRL) register to prevent these two processes from being performed consecutively.

Downloaded from Arrow.com.

34. IrDA Interface

The IrDA interface sends and receives IrDA data communication waveforms in cooperation with the SCI5 based on the IrDA (Infrared Data Association) standard 1.0.
In this section, “PCLK” is used to refer to PCLKB.

34.1 Overview

Enabling the IrDA function by using the IRE bit in the IRCR register allows encoding and decoding the TXD5 and RXD5 signals of the SCI5 to the waveforms conforming to the IrDA standard 1.0 (IRTXD5 and IRRXD5 pins). Connecting these waveforms to an infrared transmitter/receiver implements infrared data communication conforming to the IrDA standard 1.0 system.
With the IrDA standard 1.0 system, data transfer can be started at 9600 bps and the transfer rate can be changed whenever necessary. Since the IrDA interface cannot change the transfer rate automatically, the transfer rate should be changed through software.
Figure 34.1 is a block diagram showing cooperation between the IrDA and SCI5.

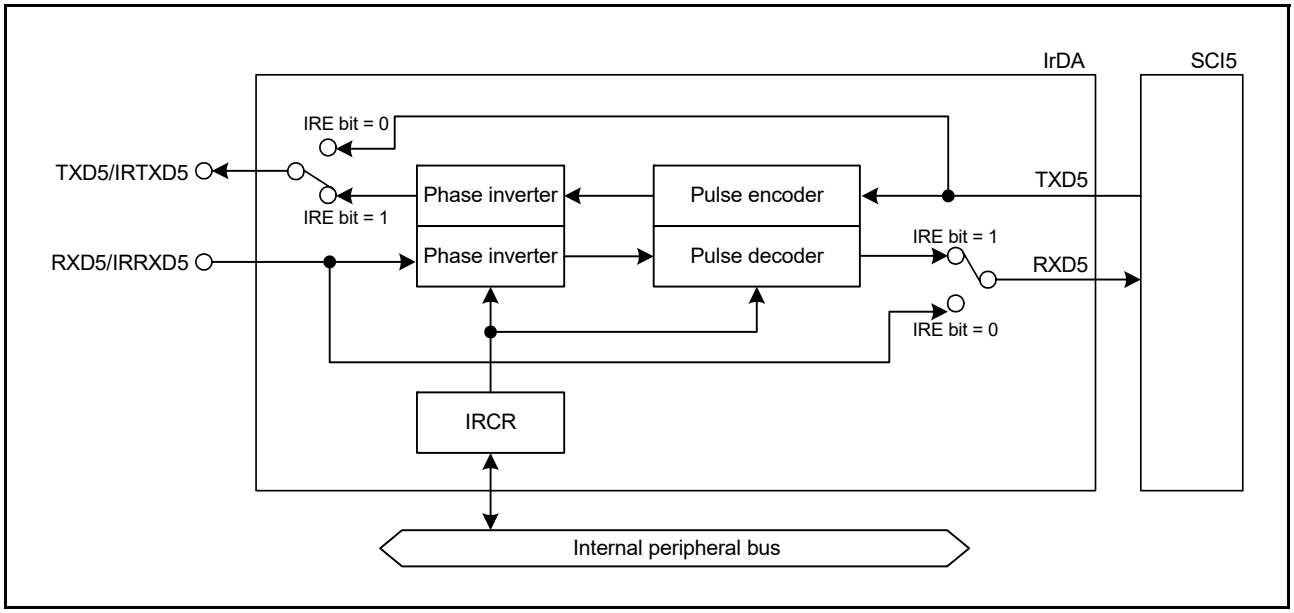


Figure 34.1 Block Diagram Showing Cooperation between IrDA and SCI5

Table 34.1 IrDA Interface I/O Pins

Pin Name	I/O	Function
IRTXD5	Output	Outputs data to be transmitted.
IRRXD5	Input	Inputs received data.

Downloaded from Arrow.com.

34.2 Register Descriptions

34.2.1 IrDA Control Register (IRCR)

Address(es): IRDA.IRCR 0008 8410h

	b7	b6	b5	b4	b3	b2	b1	b0
	IRE	IRCKS[2:0]			IRTXINV	IRRXINV	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b2	IRRXINV	IrRX Data Polarity Switching	0: IRRXD5 input is used as received data as is. 1: IRRXD5 input is used as received data after the polarity is inverted.	R/W
b3	IRTXINV	IrTX Data Polarity Switching	0: Data to be transmitted is output to IRTXD5 as is. 1: Data to be transmitted is output to IRTXD5 after the polarity is inverted.	R/W
b6 to b4	IRCKS[2:0]	IrDA Output Pulse Width Select	b6 b4 0 0 0: B × 3/16 (B = bit period) 0 0 1: PCLK/2 0 1 0: PCLK/4 0 1 1: PCLK/8 1 0 0: PCLK/16 1 0 1: PCLK/32 1 1 0: PCLK/64 1 1 1: PCLK/128	R/W
b7	IRE	IrDA Enable	0: Serial I/O pins are used for normal serial communication. 1: Serial I/O pins are used for IrDA data communication.	R/W

Note: The IRCR register values are retained in module stop state, sleep mode, deep sleep mode, and software standby mode.

IRRXINV Bit (IrRX Data Polarity Switching)

This bit inverts the logic level of the IrRXD input.
When IRRXINV = 0, the IrRXD input is used as received data as is.
When IRRXINV = 1, the IrRXD input is used after the polarity is inverted.
Set this bit when the IRE bit is 0 or at the same time as setting the IRE bit to 1.

IRTXINV Bit (IrTX Data Polarity Switching)

This bit inverts the logic level of the IrTXD output.
When IRTXINV = 0, data is transmitted to IrTXD as is, and the high-level pulse width specified by IRCKS is applied.
When IRTXINV = 1, data is transmitted to IrTXD after the polarity is inverted, and the low-level pulse width specified by IRCKS is applied.
Set this bit when the IRE bit is 0 or at the same time as setting the IRE bit to 1.

IRCKS[2:0] Bits (IrDA Output Pulse Width Select)

These bits set the high-level pulse width during IRTXD5 output encoding when the IrDA function is enabled.
Use the following procedure to set the IRCKS[2:0] bits.
(1) Set the IRCR register to specify the IrDA function (IRE bit = 1 (IrDA is enabled)).
(2) Set the IRCKS[2:0] bits to 000b.
(3) Set the SCI5.SCR.TE bit to 1 (transmission is enabled).
(4) Wait for a duration of 18/(16 × SCI5 bit rate).
(5) Set the IRCKS[2:0] bits to the target value.

Downloaded from Arrow.com.

IRE Bit (IrDA Enable)

This bit selects either normal serial communication or IrDA data communication as the function of the serial I/O pins.

Downloaded from Arrow.com.

34.3 Operation

34.3.1 Transmission

In transmission, the signals output from the SCI5 (UART frames) are converted to the IR frame data through the IrDA interface (see Figure 34.2). When the IRCR.IRTXINV bit is 0 and data is 0, high-level pulses with 3/16 of the bit period are output (initial setting). The high-level pulse width can be changed by setting the IRCR.IRCKS[2:0] bits. The standard prescribes that the minimum high-level pulse width should be 1.41 µs and the maximum high-level pulse width should be the bit period × (3/16 + 2.5%) or (the bit period × 3/16) + 1.08 µs. When the peripheral module clock PCLK is 20 MHz, the minimum high-level pulse width can be set to 1.6 µs (101b: PCLK/32) as shown in Table 34.2. When data is 1, no pulses are output.

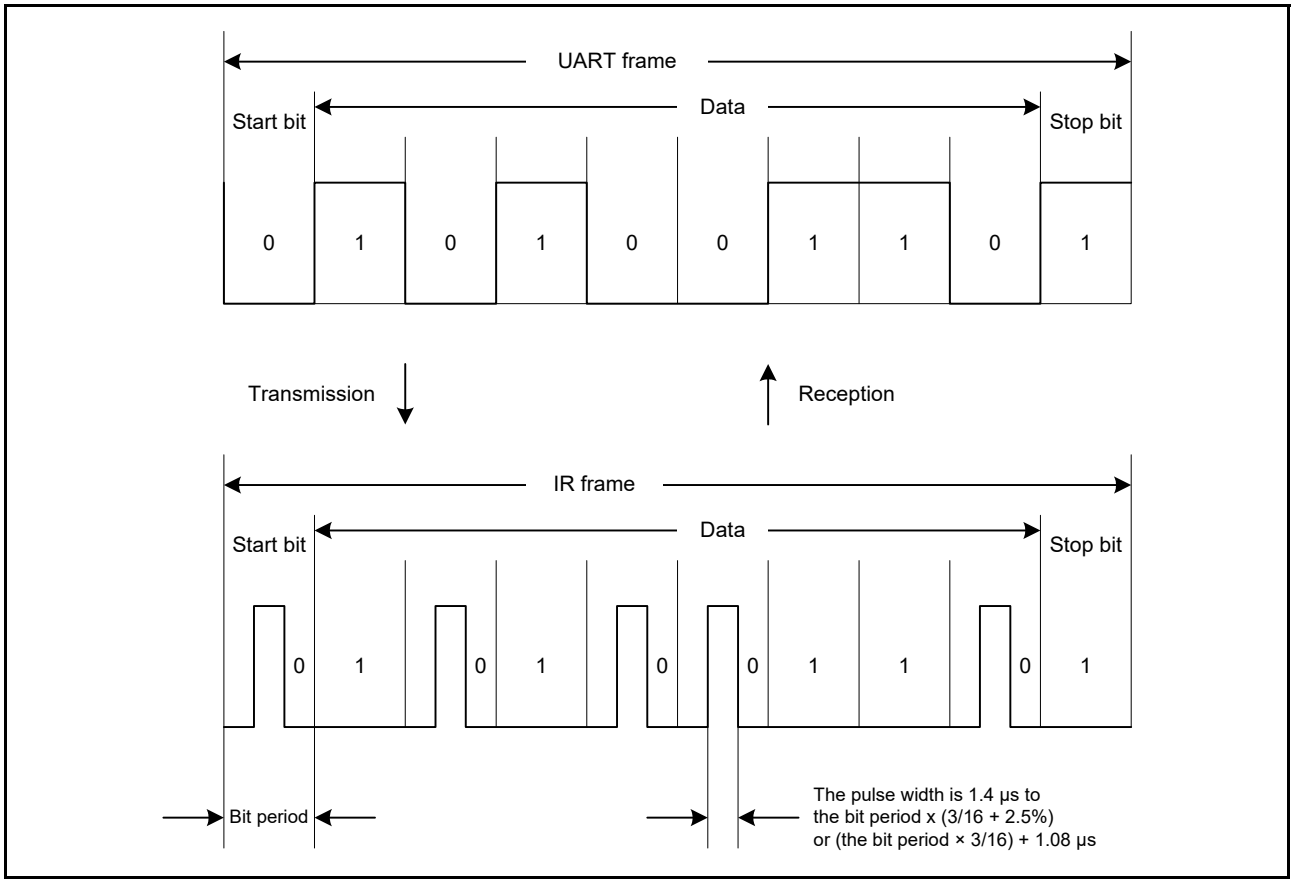


Figure 34.2 IrDA Transmission/Reception

Downloaded from Arrow.com.

34.3.2 Reception

In reception, the IR frame data is converted to the UART frame data through the IrDA interface and is input to the SCI5. Low-level data is input when the IRCR.IRRXINV bit is 0 and a high-level pulse is detected; high-level data is input when no pulse is detected for a 1-bit period.

34.3.3 Selecting High-Level Pulse Width

The IRCKS[2:0] bits need to be set to encode waveforms that meet the IrDA standard (the minimum pulse width should be 1.41 µs and the maximum should be the bit period × (3/16 + 2.5%) or (the bit period × 3/16) + 1.08 µs).
The default value of the IRCKS[2:0] bits is 000b (the bit period × 3/16). Table 34.2 lists IRCKS[2:0] bit settings and the corresponding operating frequencies and bit rates of this module, which can be used when setting the pulse width shorter than the default value.

Table 34.2 IRCKS[2:0] Bit Settings (When setting the pulse width shorter than IRCKS[2:0] = 000b (the bit period × 3/16))

Peripheral Module Operating Frequency PCLK (MHz)	Bit Rate (bps) (Upper Row)/Bit Period × 3/16 (µs) (Lower Row)*1					
	2400	9600	19200	38400	57600	115200
	78.13	19.53	9.77	4.88	3.26	1.63
4.9152	011b-111b	011b-110b	011b-101b	011b-100b	011b-100b	011b
5	011b-111b	011b-110b	011b-101b	011b-100b	011b-100b	011b
6	100b-111b	100b-110b	100b-101b	100b	100b	—*2
6.144	100b-111b	100b-110b	100b-101b	100b	100b	—*2
7.3728	100b-111b	100b-110b	100b-101b	100b-101b	100b	—*2
8	100b-111b	100b-111b	100b-110b	100b-101b	100b	—*2
9.8304	100b-111b	100b-111b	100b-110b	100b-101b	100b-101b	100b
10	100b-111b	100b-111b	100b-110b	100b-101b	100b-101b	100b
12	101b-111b	101b-111b	101b-110b	101b	101b	—*2
12.288	101b-111b	101b-111b	101b-110b	101b	101b	—*2
14	101b-111b	101b-111b	101b-111b	101b-110b	101b	—*2
14.7456	101b-111b	101b-111b	101b-111b	101b-110b	101b	—*2
16	101b-111b	101b-111b	101b-111b	101b-110b	101b	—*2
16.9344	101b-111b	101b-111b	101b-111b	101b-110b	101b	—*2
17.2032	101b-111b	101b-111b	101b-111b	101b-110b	101b	—*2
18	101b-111b	101b-111b	101b-111b	101b-110b	101b	—*2
19.6608	101b-111b	101b-111b	101b-111b	101b-110b	101b-110b	101b
20	101b-111b	101b-111b	101b-111b	101b-110b	101b-110b	101b
25	110b-111b	110b-111b	110b-111b	110b	110b	—*2

Note 1. Bit rates cannot be set for the IrDA interface. For details on bit rate settings, section 33, Serial Communications Interface (SCIg, SCIh).
Note 2. A pulse width which is less than the bit period × 3/16 cannot be set. Set the IRCKS[2:0] bits to 000b.

Downloaded from Arrow.com.

34.4 Usage Notes

34.4.1 Module Stop Function Setting

The IrDA can be enabled and disabled using module stop control register C (MSTPCRC). The IrDA is stopped after a reset. Registers can be accessed by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

34.4.2 SCI5 Setting

When using the IrDA, set the SCI5.SEMR.ABCS bit to 0 and SMR.STOP bit to 1 (2 stop bits).

34.4.3 Minimum Pulse-Width during Reception

For the input signals (IRRXD5) of this IrDA, input waveforms compliant to IrDA standard 1.0 (minimum pulse width: 1.4 us).

34.4.4 Notes on IrDA Initial Setting/Resetting

To change the value of the SCI5.SCR.TE or RE bit, set the IRCR.IRE bit to 1 (IrDA operates) and the IRCR.IRCKS[2:0] bits to 000b and then use the following procedure.

(1) Enabling transmission

- After the SCI.SCR.TE bit is set to 1 (transmission is enabled), wait for a duration of 18/(16 × SCI5 bit rate) and then switch the I/O port function to the IRTXD5 pin.

(2) Enabling reception

- After the I/O port function is switched to the IRTXD5 pin, wait for a duration of 18/(16 × SCI5 bit rate) and then set the SCI.SCR.TE bit to 1 (transmission is enabled).

When performing transmission and reception, a duration of 18/(16 × SCI5 bit rate) can be waited at the same time.

Downloaded from Arrow.com.

35. I2C-bus Interface (RIICa)

This MCU has a single-channel I2C-bus interface (RIIC).
The RIIC module conforms with the NXP I2C-bus (Inter-IC bus) interface and provides a subset of its functions.
In this section, “PCLK” is used to refer to PCLKB.

35.1 Overview

Table 35.1 lists the specifications of the RIIC, Figure 35.1 shows a block diagram of the RIIC. Table 35.2 lists the I/O pins of the RIIC.

Table 35.1 RIIC Specifications (1/2)

Item	Description
Communications format	• I2C-bus format or SMBus format • Master mode or slave mode selectable • Automatic securing of the various setup times, hold times, and bus-free times for the transfer rate
Transfer rate	Fast-mode is supported (up to 400 kbps)
Serial clock (SCL)	For master operation, the duty cycle of the SCL is selectable in the range from 4 to 96%.
Generating and detecting conditions	Start, restart, and stop conditions are automatically generated. Start conditions (including restart conditions) and stop conditions are detectable.
Slave address	• Up to three different slave addresses can be set. • 7-bit and 10-bit address formats are supported (along with the use of both at once). • General call addresses, device ID addresses, and SMBus host addresses are detectable.
Acknowledgment	• For transmission, the acknowledgment bit is automatically loaded. Transfer of the next data for transmission can be automatically suspended on detection of a not-acknowledge signal. • For reception, the acknowledgment bit is automatically transmitted. If a wait between the eighth and ninth clock pulses has been selected, software control of the acknowledgment in response to the received data is possible.
Wait function	• During reception, cycles of waiting by holding the SCL line low can be inserted at the following two types of timing: Waiting between the eighth and ninth clock pulses Waiting between the ninth clock pulse and the first clock pulse of the next byte
SDA output delay function	Changes in the timing of the output of data bits for transmission, and of the acknowledgment bit, can be delayed relative to the falling edge of SCL.
Arbitration	• For multi-master operation Clock synchronization for the SCL line in cases of conflict with the SCL signal from another master is possible. When generating the start condition would create conflict on the bus, loss in arbitration is detected by testing for non-matching between the internal data level and the actual level on the SDA line. During master operation, loss in arbitration is detected by testing for non-matching between the actual level on the SDA line and the internal data level. • Loss in arbitration due to detection of the start condition while the bus is busy is detectable (to prevent the generating of double start conditions). • Loss in arbitration in transfer of a not-acknowledge signal due to the internal signal (NACK) and the actual level on the SDA line not matching is detectable. • Loss in arbitration due to non-matching of internal data level and the actual level on the SDA line is detectable in slave transmission.
Timeout function	The internal timeout function is capable of detecting long-interval stop of the SCL.
Noise cancellation	The interface incorporates digital noise filters for both the SCL and SDA signals, and the width for noise cancellation by the filters is adjustable by software.
Interrupt sources	Four sources: • Communication error/communication event Detection of arbitration-lost, NACK, timeout, a start condition including a restart condition, or a stop condition • Receive data full (including matching with a slave address) • Transmit data empty (including matching with a slave address) • Transmission end

Downloaded from Arrow.com.

Table 35.1 RIIC Specifications (2/2)

Item	Description
Low power consumption function	Module stop state can be set.
RIIC operating modes	• Four Master transmit mode, master receive mode, slave transmit mode, and slave receive mode
Event link function (output)	Four sources (RIIC0): • Communication error/communication event Detection of arbitration-lost, NACK, timeout, a start condition including a restart condition, or a stop condition • Receive data full (including matching with a slave address) • Transmit data empty (including matching with a slave address) • Transmission end

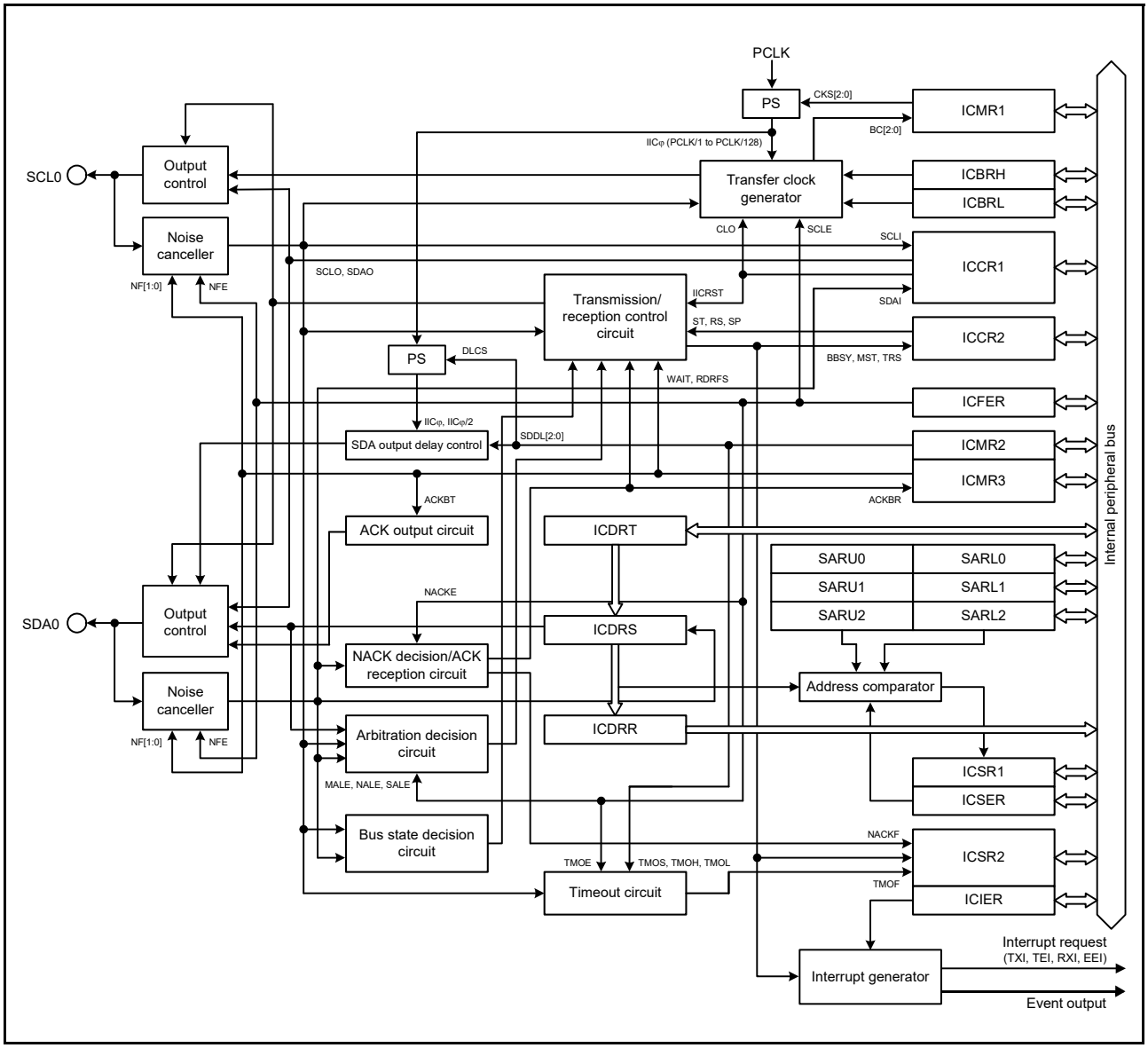


Figure 35.1 RIIC Block Diagram

The logic levels of the input signals for RIIC are CMOS when the I2C-bus is selected (ICMR3.SMBS bit is 0), or TTL when the SMBus is selected (ICMR3.SMBS bit is 1).

Downloaded from Arrow.com.

Table 35.2 RIIC Pin Configuration

Channel	Pin Name	I/O	Function
RIIC0	SCL0	I/O	RIIC0 serial clock I/O pin
	SDA0	I/O	RIIC0 serial data I/O pin

Downloaded from Arrow.com.

35.2 Register Descriptions

35.2.1 I2C-bus Control Register 1 (ICCR1)

Address(es): RIIC0.ICCR1 0008 8300h

	b7	b6	b5	b4	b3	b2	b1	b0
	ICE	IICRST	CLO	SOWP	SCLO	SDAO	SCLI	SDAI
Value after reset:	0	0	0	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b0	SDAI	SDA Line Monitor	0: SDA0 line is low. 1: SDA0 line is high.	R
b1	SCLI	SCL Line Monitor	0: SCL0 line is low. 1: SCL0 line is high.	R
b2	SDAO	SDA Output Control/Monitor	• Read: 0: The RIIC has driven the SDA0 pin low. 1: The RIIC has released the SDA0 pin. • Write: 0: The RIIC drives the SDA0 pin low. 1: The RIIC releases the SDA0 pin. (High level output is achieved through an external pull-up resistor.)	R/W
b3	SCLO	SCL Output Control/Monitor	• Read: 0: The RIIC has driven the SCL0 pin low. 1: The RIIC has released the SCL0 pin. • Write: 0: The RIIC drives the SCL0 pin low. 1: The RIIC releases the SCL0 pin. (High level output is achieved through an external pull-up resistor.)	R/W
b4	SOWP	SCLO/SDAO Write Protect	0: SCLO and SDAO bits can be written. 1: SCLO and SDAO bits are protected. (This bit is read as 1.)	R/W
b5	CLO	Additional SCL Output	0: Does not output an additional SCL (default). 1: Outputs an additional SCL. (The CLO bit is cleared automatically after one clock pulse is output.)	R/W
b6	IICRST	I2C-bus Interface Internal Reset	0: Releases the RIIC reset or internal reset. 1: Initiates the RIIC reset or internal reset. (Clears the bit counter and the SCL0/SDA0 output latch)	R/W
b7	ICE	I2C-bus Interface Enable	0: Disable (SCL0 and SDA0 pins in inactive state) 1: Enable (SCL0 and SDA0 pins in active state) (Combined with the IICRST bit to select either RIIC or internal reset.)	R/W

SDAO Bit (SDA Output Control/Monitor) and SCLO Bit (SCL Output Control/Monitor)

These bits are used to directly control the SDA0 and SCL0 signals output from the RIIC.

When writing to these bits, also write 0 to the SOWP bit.

The result of setting these bits is input to the RIIC via the input buffer. When slave mode is selected, a start condition may be detected and the bus may be released depending on the bit settings.

Do not rewrite these bits during a start condition, stop condition, restart condition, or during transmission or reception. Operation after rewriting under the above conditions is not guaranteed.

When reading these bits, the state of signals output from the RIIC can be read.

Downloaded from Arrow.com.

CLO Bit (Additional SCL Output)

This bit is used to output an additional SCL for debugging or error processing.
Normally, set the bit to 0. Setting the bit to 1 in a normal communication state causes a communication error.
For details on this function, refer to section 35.11.2, Additional SCL Output Function.

IICRST Bit (I2C-bus Interface Internal Reset)

This bit is used to reset the internal states of the RIIC.
Setting this bit to 1 initiates an RIIC reset or internal reset.
Whether an RIIC reset or internal reset is initiated is determined according to the combination with the ICE bit. Table 35.3 lists the resets of the RIIC.
The RIIC reset initializes all registers and internal states of the RIIC, and the internal reset initializes the bit counter (ICMR1.BC[2:0] bits), the I2C-bus shift register (ICDRS), and the I2C-bus status registers (ICSR1 and ICSR2) as well as the internal states of the RIIC. For the reset conditions for each register, refer to section 35.14, Initialization of Registers and Functions When a Reset is Applied or a Condition is Detected.
An internal reset initiated with the IICRST bit set to 1 during operation (with the ICE bit set to 1) resets the internal states of the RIIC without initializing the port settings and the control and setting registers of the RIIC when the bus or RIIC hangs up due to a communication error.
If the RIIC hangs up in a low level output state, resetting the internal states cancels the low level output state and releases the bus with the SCL0 pin and SDA0 pin at a high impedance.

Note: If an internal reset is initiated using the IICRST bit for a bus hang-up occurred during communication with the master device in slave mode, the states may become different between the slave device and the master device (due to the difference in the bit counter information). For this reason, do not initiate an internal reset in slave mode, but initiate restoration processing from the master device. If it is necessary to perform an internal reset in slave mode, perform it during bus free state. If an internal reset is necessary because the RIIC has hung with the SCL0 line in a low level output state in slave mode, initiate an internal reset and then generate a restart condition from the master device or resume communications from the start condition after having generated a stop condition. If communication is restarted by initiating a reset solely in the slave device without generating a start condition or restart condition from the master device, synchronization will be lost because the master and slave devices operate asynchronously.

Table 35.3 RIIC Resets

IICRST	ICE	State	Specifications
1	0	RIIC reset	Resets all registers and internal states of the RIIC.
	1	Internal reset	Resets the ICMR1.BC[2:0] bits, registers ICSR1, ICSR2, and ICDRS, and the internal states of the RIIC.

ICE Bit (I2C-bus Interface Enable)

This bit selects the active or inactive state of the SCL0 and SDA0 pins. It can also be combined with the IICRST bit to initiate two types of resets. See Table 35.3, RIIC Resets, for the types of resets.
Set the ICE bit to 1 when using the RIIC. The SCL0 and SDA0 pins are placed in the active state when the ICE bit is set to 1.
Set the ICE bit to 0 when the RIIC is not to be used. The SCL0 and SDA0 pins are placed in the inactive state when the ICE bit is set to 0. Do not assign the SCL0 or SDA0 pin to the RIIC when setting up the multi-function pin controller (MPC). Note that the slave address comparison operation is carried out if the pins are assigned to the RIIC.

Downloaded from Arrow.com.

35.2.2 I2C-bus Control Register 2 (ICCR2)

Address(es): RIIC0.ICCR2 0008 8301h

	b7	b6	b5	b4	b3	b2	b1	b0
	BBSY	MST	TRS	—	SP	RS	ST	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b1	ST	Start Condition Generation Request	0: Does not request to generate a start condition. 1: Requests to generate a start condition.	R/W
b2	RS	Restart Condition Generation Request	0: Does not request to generate a restart condition. 1: Requests to generate a restart condition.	R/W
b3	SP	Stop Condition Generation Request	0: Does not request to generate a stop condition. 1: Requests to generate a stop condition.	R/W
b4	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5	TRS	Transmit/Receive Mode	0: Receive mode 1: Transmit mode	R/W*1
b6	MST	Master/Slave Mode	0: Slave mode 1: Master mode	R/W*1
b7	BBSY	Bus Busy Detection Flag	0: The I2C-bus is released (bus free state). 1: The I2C-bus is occupied (bus busy state).	R

Note 1. When the ICMR1.MTWP bit is set to 1, bits MST and TRS can be written to.

ST Bit (Start Condition Generation Request)

This bit is used to request transition to master mode and generation of a start condition.

When this bit is set to 1 to request to generate a start condition, a start condition is generated when the BBSY flag is set to 0 (bus free state).

For details on the start condition generation, refer to section 35.10, Start Condition/Restart Condition/Stop Condition Generating Function.

[Setting condition]

- When 1 is written to the ST bit

[Clearing conditions]

- When 0 is written to the ST bit
- When a start condition has been generated (a start condition is detected)
- When the ICSR2.AL (arbitration-lost) flag is set to 1
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

Note: Set the ST bit to 1 (requests to generate a start condition) when the BBSY flag is set to 0 (bus free state). Note that arbitration may be lost due to a start condition generation error if the ST bit is set to 1 (requests to generate a start condition) when the BBSY flag is set to 1 (bus busy state).

Downloaded from Arrow.com.

RS Bit (Restart Condition Generation Request)

This bit is used to request that a restart condition be generated in master mode.
When this bit is set to 1 to request to generate a restart condition, a restart condition is generated when the BBSY flag is set to 1 (bus busy state) and the MST bit is set to 1 (master mode).
For details on the restart condition generation, refer to section 35.10, Start Condition/Restart Condition/Stop Condition Generating Function.
[Setting condition]

- When 1 is written to the RS bit with the ICCR2.BBSY flag set to 1

[Clearing conditions]

- When 0 is written to the RS bit
- When a restart condition has been generated (a start condition is detected)
- When the ICSR2.AL (arbitration-lost) flag is set to 1
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

Note: Do not set the RS bit to 1 while generating a stop condition.
Note: If 1 (requests to generate a restart condition) is written to the RS bit in slave mode, the restart condition is not generated but the RS bit remains set to 1. If the operating mode changes to master mode with the bit not being cleared, note that the restart condition may be generated.

SP Bit (Stop Condition Generation Request)

This bit is used to request that a stop condition be generated in master mode.
When this bit is set to 1 to request to generate a stop condition, a stop condition is generated when the BBSY flag is set to 1 (bus busy state) and the MST bit is set to 1 (master mode).
For details on the stop condition generation, refer to section 35.10, Start Condition/Restart Condition/Stop Condition Generating Function.
[Setting condition]

- When 1 is written to the SP bit with both the BBSY flag and the ICCR2.MST bit set to 1

[Clearing conditions]

- When 0 is written to the SP bit
- When a stop condition has been generated (a stop condition is detected)
- When the ICSR2.AL (arbitration-lost) flag is set to 1
- When a start condition and a restart condition are detected
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

Note: Writing to the SP bit is not possible while the setting of the BBSY flag is 0 (bus free state).
Note: Do not set the SP bit to 1 while a restart condition is being generated.

Downloaded from Arrow.com.

TRS Bit (Transmit/Receive Mode)

This bit indicates transmit or receive mode.
The RIIC is in receive mode when the TRS bit is set to 0 and is in transmit mode when the bit is set to 1. Combination of this bit and the MST bit indicates the operating mode of the RIIC.
The value of the TRS bit is automatically changed to 1 for transmission or 0 for reception in response to the generation or detection of a start condition and setting of the R/W# bit. Although writing to the TRS bit is possible when the ICMR1.MTWP bit is set to 1, writing to this bit is not necessary during normal usage.
[Setting conditions]

- When a start condition is generated normally according to the start condition generation request (when a start condition is detected with the ST bit set to 1)
- When a restart condition is generated normally according to the restart condition generation request (when a restart condition is detected with the RS bit set to 1)
- When the R/W# bit added to the slave address is set to 0 in master mode
- When the address received in slave mode matches the address enabled in the ICSER register, with the R/W# bit set to 1
- When 1 is written to the TRS bit with the ICMR1.MTWP bit set to 1

[Clearing conditions]

- When a stop condition is detected
- The ICSR2.AL (arbitration-lost) flag being set to 1
- In master mode, reception of a slave address to which an R/W# bit with the value 1 is appended
- In slave mode, a match between the received address and the address enabled in the ICSER register when the value of the received R/W# bit is 0 (including cases where the received address is the general call address)
- In slave mode, a restart condition is detected (that is, a start condition is detected while the ICCR2.BBSY flag is 1 and the ICCR2.MST bit is 0)
- When 0 is written to the TRS bit with the ICMR1.MTWP bit set to 1
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

MST Bit (Master/Slave Mode)

This bit indicates master or slave mode.
The RIIC is in slave mode when the MST bit is set to 0 and is in master mode when the bit is set to 1. Combination of this bit and the TRS bit indicates the operating mode of the RIIC.
The value of the MST bit is automatically changed to 1 for master mode or 0 for slave mode by generating of a start condition and generating or detection of a stop condition, etc. Although writing to the MST bit is possible when the ICMR1.MTWP bit is set to 1, writing to this bit is not necessary during normal usage.
[Setting conditions]

- When a start condition is generated normally according to the start condition generation request (when a start condition is detected with the ST bit set to 1)
- When 1 is written to the MST bit with the ICMR1.MTWP bit set to 1

[Clearing conditions]

- When a stop condition is detected
- When the ICSR2.AL (arbitration-lost) flag is set to 1
- When 0 is written to the MST bit with the ICMR1.MTWP bit set to 1
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

Downloaded from Arrow.com.

BBSY Flag (Bus Busy Detection Flag)

The BBSY flag indicates whether the I2C-bus is occupied (bus busy state) or released (bus free state).
This bit is set to 1 when the SDA0 line changes from high to low under the condition of SCL0 line = high, assuming that a start condition has been generated.
The RIIC recognizes the SDA0 line changing from low to high while the SCL0 line is high as generation of the stop condition. After that, this flag becomes 0 if the RIIC does not detect a start condition during the bus free time (the period set in the ICBRL register).
[Setting condition]

- When a start condition is detected

[Clearing conditions]

- When the bus free time (specified in the ICBRL register) start condition is not detected after detecting a stop condition
- When 1 is written to the ICCR1.IICRST bit with the ICCR1.ICE bit set to 0 (RIIC reset)

Downloaded from Arrow.com.

35.2.3 I2C-bus Mode Register 1 (ICMR1)

Address(es): RIIC0.ICMR1 0008 8302h

	b7	b6	b5	b4	b3	b2	b1	b0
	MTWP	CKS[2:0]			BCWP	BC[2:0]		
Value after reset:	0	0	0	0	1	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	BC[2:0]	Bit Counter	b2 b0 0 0 0: 9 bits 0 0 1: 2 bits 0 1 0: 3 bits 0 1 1: 4 bits 1 0 0: 5 bits 1 0 1: 6 bits 1 1 0: 7 bits 1 1 1: 8 bits	R/W*1
b3	BCWP	BC Write Protect	0: Enables a value to be written in the BC[2:0] bits. (This bit is read as 1.)	R/W*1
b6 to b4	CKS[2:0]	Internal Reference Clock Select	Select the internal reference clock (IICφ) source for the RIIC. b6 b4 0 0 0: PCLK/1 0 0 1: PCLK/2 0 1 0: PCLK/4 0 1 1: PCLK/8 1 0 0: PCLK/16 1 0 1: PCLK/32 1 1 0: PCLK/64 1 1 1: PCLK/128	R/W
b7	MTWP	MST/TRS Write Protect	0: Disables writing to the ICCR2.MST and TRS bits. 1: Enables writing to the ICCR2.MST and TRS bits.	R/W

Note 1. Rewrite the BC[2:0] bits and set the BCWP bit to 0 at the same time.

BC[2:0] Bits (Bit Counter)

These bits function as a counter that indicates the number of bits remaining to be transferred at the detection of a rising edge on the SCL0 line. Although these bits are writable and readable, it is not necessary to access these bits under normal conditions.

To write to these bits, specify the number of bits to be transferred plus one (data is transferred with an additional acknowledgment bit) between transferred bytes when the SCL0 line is low.

The values of the BC[2:0] bits return to 000b at the end of a data transfer including the acknowledgment bit or when a start condition including a restart condition is detected.

Downloaded from Arrow.com.

35.2.4 I2C-bus Mode Register 2 (ICMR2)

Address(es): RIIC0.ICMR2 0008 8303h

	b7	b6	b5	b4	b3	b2	b1	b0
	DLCS	SDDL[2:0]			—	TMOH	TMOL	TMOS
Value after reset:	0	0	0	0	0	1	1	0

Bit	Symbol	Bit Name	Description	R/W
b0	TMOS	Timeout Detection Time Select	0: Long mode is selected. 1: Short mode is selected.	R/W
b1	TMOL	Timeout L Count Control	0: Count-up is disabled while the SCL0 line is low. 1: Count-up is enabled while the SCL0 line is low.	R/W
b2	TMOH	Timeout H Count Control	0: Count-up is disabled while the SCL0 line is high. 1: Count-up is enabled while the SCL0 line is high.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6 to b4	SDDL[2:0]	SDA Output Delay Counter	• When ICMR2.DLCS bit is 0 (IICφ) b6 b4 0 0: No output delay 0 1: 1 IICφ cycle 1 0: 2 IICφ cycles 1 1: 3 IICφ cycles 0 0: 4 IICφ cycles 0 1: 5 IICφ cycles 1 0: 6 IICφ cycles 1 1: 7 IICφ cycles • When ICMR2.DLCS bit is 1 (IICφ/2) b6 b4 0 0: No output delay 0 1: 1 or 2 IICφ cycles 1 0: 3 or 4 IICφ cycles 1 1: 5 or 6 IICφ cycles 0 0: 7 or 8 IICφ cycles 0 1: 9 or 10 IICφ cycles 1 0: 11 or 12 IICφ cycles 1 1: 13 or 14 IICφ cycles	R/W
b7	DLCS	SDA Output Delay Clock Source Select	0: The internal reference clock (IICφ) is selected as the clock source of the SDA output delay counter. 1: The internal reference clock divided by 2 (IICφ/2) is selected as the clock source of the SDA output delay counter.*1	R/W

Note 1. The DLCS bit setting of 1 (IICφ/2) only becomes valid when SCL pin is low. When SCL pin is high, the DLCS bit setting of 1 becomes invalid and the clock source becomes the internal reference clock (IICφ).

TMOS Bit (Timeout Detection Time Select)

This bit is used to select long mode or short mode for the timeout detection time when the timeout function is enabled (ICFER.TMOE bit is 1). When this bit is set to 0, long mode is selected. When this bit is set to 1, short mode is selected. In long mode, the timeout detection internal counter functions as a 16-bit counter. In short mode, the counter functions as a 14-bit counter. While the SCL0 line is in the state that enables this counter as specified by bits TMOH and TMOL, the counter counts up in synchronization with the internal reference clock (IICφ) as a count source.

For details on the timeout function, refer to section 35.11.1, Timeout Function.

Downloaded from Arrow.com.

TMOL Bit (Timeout L Count Control)

This bit is used to enable or disable the internal counter of the timeout function to count up while the SCL0 line is held low when the timeout function is enabled (ICFER.TMOE bit is 1).

TMOH Bit (Timeout H Count Control)

This bit is used to enable or disable the internal counter of the timeout function to count up while the SCL0 line is held high when the timeout function is enabled (ICFER.TMOE bit is 1).

SDDL[2:0] Bits (SDA Output Delay Counter)

The SDA output can be delayed by the SDDL[2:0] setting. This counter works with the clock source selected by the DLCS bit. The setting of this function can be used for all types of SDA output, including the transmission of the acknowledgment bit.

Set the SDA output delay time to meet the I2C-bus specification (within the data valid time/data valid acknowledge time*1) or the SMBus specification (more than the data hold time (300 ns) and less than "clock low period – data setup time (250 ns)"). Note that, if a value outside the specification is set, communication with communication devices may malfunction or it may seemingly become a start condition or stop condition depending on the bus state.

For details on this function, refer to section 35.5, SDA Output Delay Function.

Note 1. Data valid time/data valid acknowledge time
3,450 ns (up to 100 kbps: Standard-mode (Sm))
900 ns (up to 400 kbps: Fast-mode (Fm))

Downloaded from Arrow.com.

35.2.5 I2C-bus Mode Register 3 (ICMR3)

Address(es): RIIC0.ICMR3 0008 8304h

	b7	b6	b5	b4	b3	b2	b1	b0
	SMBS	WAIT	RDRFS	ACKWP	ACKBT	ACKBR	NF[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	NF[1:0]	Noise Filter Stage Select	b1 b0 0 0: Noise of up to one IICφ cycle is filtered out (single-stage filter). 0 1: Noise of up to two IICφ cycles is filtered out (2-stage filter). 1 0: Noise of up to three IICφ cycles is filtered out (3-stage filter). 1 1: Noise of up to four IICφ cycles is filtered out (4-stage filter).	R/W
b2	ACKBR	Received Acknowledge	0: 0 is received as the acknowledgment bit (ACK reception). 1: 1 is received as the acknowledgment bit (NACK reception).	R
b3	ACKBT	Transmit Acknowledge	0: 0 is to be sent as the acknowledgment bit (ACK transmission). 1: 1 is to be sent as the acknowledgment bit (NACK transmission).	R/W*1
b4	ACKWP	ACKBT Write Protect	0: Modification of the ACKBT bit is disabled. 1: Modification of the ACKBT bit is enabled.	R/W*1
b5	RDRFS	RDRF Flag Set Timing Select	0: The RDRF flag is set at the rising edge of the ninth SCL. (The SCL0 line is not held low at the falling edge of the eighth clock pulse.) 1: The RDRF flag is set at the rising edge of the eighth SCL. (The SCL0 line is held low at the falling edge of the eighth clock pulse.) Low-hold is released by writing a value to the ACKBT bit.	R/W*2
b6	WAIT	WAIT	0: No WAIT (The period between ninth clock pulse and first clock pulse is not held low.) 1: WAIT (The period between ninth clock pulse and first clock pulse is held low.) Low-hold is released by reading the ICDRR register.	R/W*2
b7	SMBS	SMBus/I2C-bus Select	0: The I2C-bus is selected. 1: The SMBus is selected.	R/W

Note 1. Write to the ACKBT bit only while the ACKWP bit is already 1. If it is attempted to write 1 to both the ACKWP and ACKBT bits at the same time, the ACKBT bit will not be set to 1.

Note 2. The WAIT and RDRFS bits are valid only in receive mode (invalid in transmit mode).

NF[1:0] Bits (Noise Filter Stage Select)

These bits are used to select the number of stages in the digital noise filter.

For details on the digital noise filter function, refer to section 35.6, Digital Noise Filters.

Note: Set the noise range to be filtered out by the noise filter within a range less than the SCL0 line high period or low period. If the noise filter width is set to a value of [the shorter one of either SCL high width or SCL low width] – {1.5 × t_{IICcyc} (cycle time of internal reference clock (IICφ)) + 120 ns (pulse width suppressed by the analog noise filter, a reference value)} or a greater value, the serial clock is regarded as noise by the noise filter function of the RIIC, which may prevent the RIIC from operating normally.

Downloaded from Arrow.com.

ACKBR Bit (Received Acknowledge)

This bit is used to store the value of the acknowledgment bit received from the receiver in transmit mode.
[Setting condition]

- When 1 is received as the acknowledgment bit with the ICCR2.TRS bit set to 1

[Clearing conditions]

- When 0 is received as the acknowledgment bit with the ICCR2.TRS bit set to 1
- When 1 is written to the ICCR1.IICRST bit while the ICCR1.ICE bit is 0 (RIIC reset)

ACKBT Bit (Transmit Acknowledge)

This bit is used to set the bit to be sent at the acknowledgment timing in receive mode.
[Setting condition]

- When 1 is written to this bit with the ACKWP bit set to 1

[Clearing conditions]

- When 0 is written to this bit with the ACKWP bit set to 1
- When stop condition generation is detected (when a stop condition is detected with the ICCR2.SP bit set to 1)
- When 1 is written to the ICCR1.IICRST bit while the ICCR1.ICE bit is 0 (RIIC reset)

ACKWP Bit (ACKBT Write Protect)

This bit is used to control the modification of the ACKBT bit.

RDRFS Bit (RDRF Flag Set Timing Select)

This bit is used to select the RDRF flag set timing in receive mode and also to select whether to hold the SCL0 line low at the falling edge of the eighth SCL.
When the RDRFS bit is 0, the SCL0 line is not held low at the falling edge of the eighth SCL, and the RDRF flag is set to 1 at the rising edge of the ninth SCL.
When the RDRFS bit is 1, the RDRF flag is set to 1 at the rising edge of the eighth SCL and the SCL0 line is held low at the falling edge of the eighth SCL. The low-hold of the SCL0 line is released by writing a value to the ACKBT bit.
After data is received with this setting, the SCL0 line is automatically held low before the acknowledgment bit is sent. This enables processing to send ACK (ACKBT bit is 0) or NACK (ACKBT bit is 1) according to receive data.

WAIT Bit (WAIT)

This bit is used to control whether to hold the period between the ninth SCL and the first SCL low until the I2C-bus receive data register (ICDRR) is completely read each time single-byte data is received in receive mode.
When the WAIT bit is 0, the receive operation is continued without holding the period between the ninth and the first SCL low. When both the RDRFS and WAIT bits are 0, continuous receive operation is enabled with the double buffer.
When the WAIT bit is 1, the SCL0 line is held low from the falling edge of the ninth SCL until the ICDRR register value is read each time single-byte data is received. This enables receive operation in byte units.

Note: When the value of the WAIT bit is to be read, be sure to read the ICDRR register beforehand.

SMBS Bit (SMBus/I2C-bus Select)

Setting this bit to 1 selects the SMBus and enables the ICSER.HOAE bit.

Downloaded from Arrow.com.

35.2.6 I2C-bus Function Enable Register (ICFER)

Address(es): RIIC0.ICFER 0008 8305h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	SCLE	NFE	NACKE	SALE	NALE	MALE	TMOE
Value after reset:	0	1	1	1	0	0	1	0

Bit	Symbol	Bit Name	Description	R/W
b0	TMOE	Timeout Function Enable	0: The timeout function is disabled. 1: The timeout function is enabled.	R/W
b1	MALE	Master Arbitration-Lost Detection Enable	0: Master arbitration-lost detection is disabled. (Disables the arbitration-lost detection function and does not clear the ICCR2.MST and TRS bits automatically when arbitration is lost.) 1: Master arbitration-lost detection is enabled. (Enables the arbitration-lost detection function and clears the ICCR2.MST and TRS bits automatically when arbitration is lost.)	R/W
b2	NALE	NACK Transmission Arbitration-Lost Detection Enable	0: NACK transmission arbitration-lost detection is disabled. 1: NACK transmission arbitration-lost detection is enabled.	R/W
b3	SALE	Slave Arbitration-Lost Detection Enable	0: Slave arbitration-lost detection is disabled. 1: Slave arbitration-lost detection is enabled.	R/W
b4	NACKE	NACK Reception Transfer Suspension Enable	0: Transfer operation is not suspended during NACK reception (transfer suspension disabled). 1: Transfer operation is suspended during NACK reception (transfer suspension enabled).	R/W
b5	NFE	Digital Noise Filter Enable	0: Digital noise filters are not used. 1: Digital noise filters are used.	R/W
b6	SCLE	SCL Synchronization Enable	0: SCL synchronization is disabled. 1: SCL synchronization is enabled.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

TMOE Bit (Timeout Function Enable)

This bit is used to enable or disable the timeout function.

For details on the timeout function, refer to section 35.11.1, Timeout Function.

MALE Bit (Master Arbitration-Lost Detection Enable)

This bit is used to specify whether to use the arbitration-lost detection function in master mode. Normally, set this bit to 1.

NALE Bit (NACK Transmission Arbitration-Lost Detection Enable)

This bit is used to specify whether to cause arbitration to be lost when ACK is detected during transmission of NACK in receive mode (such as when slaves with the same address exist on the bus or when two or more masters select the same slave device simultaneously with different number of receive bytes).

SALE Bit (Slave Arbitration-Lost Detection Enable)

This bit is used to specify whether to cause arbitration to be lost when a value different from the value being transmitted is detected on the bus in slave transmit mode (such as when slaves with the same address exist on the bus or when a mismatch with the transmit data occurs due to noise).

Downloaded from Arrow.com.

NACKE Bit (NACK Reception Transfer Suspension Enable)

This bit is used to specify whether to continue or discontinue the data transfer when NACK is received in transmit mode. Normally, set this bit to 1.

When NACK is received with the NACKE bit set to 1, the next transfer operation is suspended.

When the NACKE bit is 0, the next transfer operation is continued regardless of the value of the received acknowledgment bit.

For details on the NACK reception transfer suspension function, refer to section 35.8.2, NACK Reception Transfer Suspension Function.

SCLE Bit (SCL Synchronization Enable)

This bit is used to specify whether to the SCL output is to be synchronized with the SCL input. Normally, set this bit to 1.

When the SCLE bit is set to 0 (SCL synchronization is disabled), the RIIC does not synchronize the SCL output with the SCL input. In this setting, the RIIC outputs the clock with the transfer rate set in registers ICBRH and ICBRL regardless of the SCL0 line state. For this reason, if the load of the I2C-bus line is much larger than the specification value or if the SCL output overlaps in multiple masters, the short-cycle SCL that does not meet the specification may be output. When the SCL synchronization is not used, it also affects the generation of a start condition, restart condition, and stop condition, and the continuous output of additional SCL.

This bit must not be set to 0 except for checking the output of the set transfer rate.

Downloaded from Arrow.com.

35.2.7 I2C-bus Status Enable Register (ICSER)

Address(es): RIIC0.ICSER 0008 8306h

	b7	b6	b5	b4	b3	b2	b1	b0
	HOAE	—	DIDE	—	GCAE	SAR2E	SAR1E	SAR0E
Value after reset:	0	0	0	0	1	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	SAR0E	Slave Address Register 0 Enable	0: Slave address in registers SARL0 and SARU0 is disabled. 1: Slave address in registers SARL0 and SARU0 is enabled.	R/W
b1	SAR1E	Slave Address Register 1 Enable	0: Slave address in registers SARL1 and SARU1 is disabled. 1: Slave address in registers SARL1 and SARU1 is enabled.	R/W
b2	SAR2E	Slave Address Register 2 Enable	0: Slave address in registers SARL2 and SARU2 is disabled. 1: Slave address in registers SARL2 and SARU2 is enabled.	R/W
b3	GCAE	General Call Address Enable	0: General call address detection is disabled. 1: General call address detection is enabled.	R/W
b4	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5	DIDE	Device-ID Address Detection Enable	0: Device-ID address detection is disabled. 1: Device-ID address detection is enabled.	R/W
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7	HOAE	Host Address Enable	0: Host address detection is disabled. 1: Host address detection is enabled.	R/W

SARyE Bit (Slave Address Register y Enable) (y = 0 to 2)

This bit is used to enable or disable the slave address set in registers SARLy and SARUy.

When this bit is set to 1, the slave address set in registers SARLy and SARUy is enabled and is compared with the received slave address.

When this bit is set to 0, the slave address set in registers SARLy and SARUy is disabled and is ignored even if it matches the received slave address.

GCAE Bit (General Call Address Enable)

This bit is used to specify whether to ignore the general call address (0000 000b + 0 (write): All 0) when it is received.

When this bit is set to 1, if the received slave address matches the general call address, the RIIC recognizes the received slave address as the general call address independently of the slave addresses set in registers SARLy and SARUy (y = 0 to 2) and performs data receive operation.

When this bit is set to 0, the received slave address is ignored even if it matches the general call address.

DIDE Bit (Device-ID Address Detection Enable)

This bit is used to specify whether to recognize and execute the device-ID address when a device ID (1111 100b) is received in the first byte after a start condition or restart condition is detected.

When this bit is set to 1, if the received first byte matches the device ID, the RIIC recognizes that the device-ID address has been received. When the following R/W# bit is 0 (write), the RIIC recognizes the second and the following bytes as slave addresses and continues the receive operation.

When this bit is set to 0, the RIIC ignores the received first byte even if it matches the device ID address and recognizes the first byte as a normal slave address.

For details on the device-ID address detection, refer to section 35.7.3, Device-ID Address Detection.

Downloaded from Arrow.com.

HOAE Bit (Host Address Enable)

This bit is used to specify whether to ignore received host address (0001 000b) when the ICMR3.SMBS bit is 1.

When this bit is set to 1 while the ICMR3.SMBS bit is 1, if the received slave address matches the host address, the RIIC recognizes the received slave address as the host address independently of the slave addresses set in registers SARLy and SARUy (y = 0 to 2) and performs the receive operation.

When the ICMR3.SMBS bit or the HOAE bit is set to 0, the received slave address is ignored even if it matches the host address.

Downloaded from Arrow.com.

35.2.8 I2C-bus Interrupt Enable Register (ICIER)

Address(es): RIIC0.ICIER 0008 8307h

	b7	b6	b5	b4	b3	b2	b1	b0
	TIE	TEIE	RIE	NAKIE	SPIE	STIE	ALIE	TMOIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TMOIE	Timeout Interrupt Request Enable	0: Timeout interrupt (TMOI) request is disabled. 1: Timeout interrupt (TMOI) request is enabled.	R/W
b1	ALIE	Arbitration-Lost Interrupt Request Enable	0: Arbitration-lost interrupt (ALI) request is disabled. 1: Arbitration-lost interrupt (ALI) request is enabled.	R/W
b2	STIE	Start Condition Detection Interrupt Request Enable	0: Start condition detection interrupt (STI) request is disabled. 1: Start condition detection interrupt (STI) request is enabled.	R/W
b3	SPIE	Stop Condition Detection Interrupt Request Enable	0: Stop condition detection interrupt (SPI) request is disabled. 1: Stop condition detection interrupt (SPI) request is enabled.	R/W
b4	NAKIE	NACK Reception Interrupt Request Enable	0: NACK reception interrupt (NAKI) request is disabled. 1: NACK reception interrupt (NAKI) request is enabled.	R/W
b5	RIE	Receive Data Full Interrupt Request Enable	0: Receive data full interrupt (RXI) request is disabled. 1: Receive data full interrupt (RXI) request is enabled.	R/W
b6	TEIE	Transmission End Interrupt Request Enable	0: Transmission end interrupt (TEI) request is disabled. 1: Transmission end interrupt (TEI) request is enabled.	R/W
b7	TIE	Transmit Data Empty Interrupt Request Enable	0: Transmit data empty interrupt (TXI) request is disabled. 1: Transmit data empty interrupt (TXI) request is enabled.	R/W

TMOIE Bit (Timeout Interrupt Request Enable)

This bit is used to enable or disable timeout interrupt (TMOI) requests when the ICSR2.TMOF flag is set to 1. A TMOI interrupt request is canceled by setting the TMOF flag or the TMOIE bit to 0.

ALIE Bit (Arbitration-Lost Interrupt Request Enable)

This bit is used to enable or disable arbitration-lost interrupt (ALI) requests when the ICSR2.AL flag is set to 1. An ALI interrupt request is canceled by setting the AL flag or the ALIE bit to 0.

STIE Bit (Start Condition Detection Interrupt Request Enable)

This bit is used to enable or disable start condition detection interrupt (STI) requests when the ICSR2.START flag is set to 1. An STI interrupt request is canceled by setting the START flag or the STIE bit to 0.

SPIE Bit (Stop Condition Detection Interrupt Request Enable)

This bit is used to enable or disable stop condition detection interrupt (SPI) requests when the ICSR2.STOP flag is set to 1. An SPI interrupt request is canceled by setting the STOP flag or the SPIE bit to 0.

NAKIE Bit (NACK Reception Interrupt Request Enable)

This bit is used to enable or disable NACK reception interrupt (NAKI) requests when the ICSR2.NACKF flag is set to 1. An NAKI interrupt request is canceled by setting the NACKF flag or the NAKIE bit to 0.

RIE Bit (Receive Data Full Interrupt Request Enable)

This bit is used to enable or disable receive data full interrupt (RXI) requests when the ICSR2.RDRF flag is set to 1.

Downloaded from Arrow.com.

TEIE Bit (Transmission End Interrupt Request Enable)

This bit is used to enable or disable transmission end interrupt (TEI) requests when the ICSR2.TEND flag is set to 1. An TEI interrupt request is canceled by setting the TEND flag or the TEIE bit to 0.

TIE Bit (Transmit Data Empty Interrupt Request Enable)

This bit is used to enable or disable transmit data empty interrupt (TXI) requests when the ICSR2.TDRE flag is set to 1.

Downloaded from Arrow.com.

35.2.9 I2C-bus Status Register 1 (ICSR1)

Address(es): RIIC0.ICSR1 0008 8308h

	b7	b6	b5	b4	b3	b2	b1	b0
	HOA	—	DID	—	GCA	AAS2	AAS1	AAS0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	AAS0	Slave Address 0 Detection Flag	0: Slave address 0 is not detected. 1: Slave address 0 is detected.	R/(W) *1
b1	AAS1	Slave Address 1 Detection Flag	0: Slave address 1 is not detected. 1: Slave address 1 is detected.	R/(W) *1
b2	AAS2	Slave Address 2 Detection Flag	0: Slave address 2 is not detected. 1: Slave address 2 is detected.	R/(W) *1
b3	GCA	General Call Address Detection Flag	0: General call address is not detected. 1: General call address is detected.	R/(W) *1
b4	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5	DID	Device-ID Address Detection Flag	0: Device-ID command is not detected. 1: Device-ID command is detected. • This bit is set to 1 when the first byte received immediately after a start condition is detected matches a value of (device ID (1111 100b) + 0 (write)).	R/(W) *1
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7	HOA	Host Address Detection Flag	0: Host address is not detected. 1: Host address is detected. • This bit is set to 1 when the received slave address matches the host address (0001 000b).	R/(W) *1

Note 1. Only 0 can be written to clear the flag.

AASy Flag (Slave Address y Detection Flag) (y = 0 to 2)

[Setting conditions]

For 7-bit address format: SARUy.FS bit = 0

- When the received slave address matches the SARLy.SVA[6:0] bits value with the ICSER.SARyE bit set to 1 (slave address y detection enabled)
  This flag is set to 1 at the rising edge of the ninth SCL in the first byte.

For 10-bit address format: SARUy.FS bit = 1

- When the received slave address matches a value of (11110b + SARUy.SVA[1:0] bits) and the following address matches the SARLy value with the ICSER.SARyE bit set to 1 (slave address y detection enabled)
  This flag is set to 1 at the rising edge of the ninth SCL in the second byte.

[Clearing conditions]

- When 0 is written to the AASy flag after reading the AASy flag to be 1
- When a stop condition is detected
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

For 7-bit address format: SARUy.FS bit = 0

- When the received slave address does not match the SARLy.SVA[6:0] bits value with the ICSER.SARyE bit set to 1 (slave address y detection enabled)
  This flag is set to 0 at the rising edge of the ninth SCL in the first byte.

Downloaded from Arrow.com.

For 10-bit address format: SARUy.FS bit = 1

- When the received slave address does not match a value of (11110b + SARUy.SVA[1:0] bits) with the ICSER.SARyE bit set to 1 (slave address y detection enabled)
  This flag is set to 0 at the rising edge of the ninth SCL in the first byte.
- When the received slave address matches a value of (11110b + SARUy.SVA[1:0] bits) and the following address does not match the SARLy value with the ICSER.SARyE bit set to 1 (slave address y detection enabled)
  This flag is set to 0 at the rising edge of the ninth SCL in the second byte.

GCA Flag (General Call Address Detection Flag)

[Setting condition]

- When the received slave address matches the general call address (0000 000b + 0 (write)) with the ICSER.GCAE bit set to 1 (general call address detection is enabled)
  This flag is set to 1 at the rising edge of the ninth SCL in the first byte.

[Clearing conditions]

- When 0 is written to the GCA flag after reading GCA flag to be 1
- When a stop condition is detected
- When the received slave address does not match the general call address (0000 000b + 0 (write)) with the ICSER.GCAE bit set to 1 (general call address detection is enabled)
  This flag is set to 0 at the rising edge of the ninth SCL in the first byte.
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

DID Flag (Device-ID Address Detection Flag)

[Setting condition]

- When the first byte received immediately after a start condition or restart condition is detected matches a value of (device ID (1111 100b) + 0 (write)) with the ICSER.DIDE bit set to 1 (device-ID address detection is enabled)
  This flag is set to 1 at the rising edge of the ninth SCL in the first byte.

[Clearing conditions]

- When 0 is written to the DID flag after reading DID flag to be 1
- When a stop condition is detected
- When the first byte received immediately after a start condition or restart condition is detected does not match a value of (device ID (1111 100b)) with the ICSER.DIDE bit set to 1 (device-ID address detection is enabled)
  This flag is set to 0 at the rising edge of the ninth SCL in the first byte.
- When the first byte received immediately after a start condition or restart condition is detected matches a value of (device ID (1111 100b) + 0 (write)) and the second byte does not match any of slave addresses 0 to 2 with the ICSER.DIDE bit set to 1 (device-ID address detection is enabled)
  This flag is set to 0 at the rising edge of the ninth SCL in the second byte.
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

HOA Flag (Host Address Detection Flag)

[Setting condition]

- When the received slave address matches the host address (0001 000b) with the ICSER.HOAE bit set to 1 (host address detection is enabled)
  This flag is set to 1 at the rising edge of the ninth SCL in the first byte.

[Clearing conditions]

- When 0 is written to the HOA flag after reading HOA flag to be 1
- When a stop condition is detected
- When the received slave address does not match the host address (0001 000b) with the ICSER.HOAE bit set to 1

Downloaded from Arrow.com.

(host address detection is enabled)
This flag is set to 0 at the rising edge of the ninth SCL in the first byte.

- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

Downloaded from Arrow.com.

35.2.10 I2C-bus Status Register 2 (ICSR2)

Address(es): RIIC0.ICSR2 0008 8309h

	b7	b6	b5	b4	b3	b2	b1	b0
	TDRE	TEND	RDRF	NACKF	STOP	START	AL	TMOF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TMOF	Timeout Detection Flag	0: Timeout is not detected. 1: Timeout is detected.	R/(W) *1
b1	AL	Arbitration-Lost Flag	0: Arbitration is not lost. 1: Arbitration is lost.	R/(W) *1
b2	START	Start Condition Detection Flag	0: Start condition is not detected. 1: Start condition is detected.	R/(W) *1
b3	STOP	Stop Condition Detection Flag	0: Stop condition is not detected. 1: Stop condition is detected.	R/(W) *1
b4	NACKF	NACK Detection Flag	0: NACK is not detected. 1: NACK is detected.	R/(W) *1
b5	RDRF	Receive Data Full Flag	0: The ICDRR register contains no receive data. 1: The ICDRR register contains receive data.	R/(W) *1
b6	TEND	Transmission End Flag	0: Data is being transmitted. 1: Data has been transmitted.	R/(W) *1
b7	TDRE	Transmit Data Empty Flag	0: The ICDRT register contains transmit data. 1: The ICDRT register contains no transmit data.	R

Note 1. Only 0 can be written to clear the flag.

TMOF Flag (Timeout Detection Flag)

This flag is set to 1 when the RIIC recognizes timeout after the SCL0 line state remains unchanged for a certain period.
[Setting condition]

- When the SCL0 line state remains unchanged for the period specified by bits ICMR2.TMOH, TMOL, and TMOS while the ICFER.TMOE bit is 1 (the timeout function is enabled) in master mode or in slave mode and the received slave address matches.

[Clearing conditions]

- When 0 is written to the TMOF bit after reading TMOF = 1
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

AL Flag (Arbitration-Lost Flag)

This flag shows that bus mastership has been lost (loss in arbitration) due to a bus conflict or some other reason when a start condition is generated or an address and data are transmitted. The RIIC monitors the level on the SDA0 line during transmission and, if the level on the line does not match the value of the bit being output, sets the value of the AL flag to 1 to indicate that the bus is occupied by another device.
The RIIC can also detect loss of arbitration during NACK transmission in receive mode or during data transmission in slave mode.

Downloaded from Arrow.com.

[Setting conditions]

When master arbitration-lost detection is enabled: ICFER.MALE = 1

- When the internal SDA output state does not match the SDA0 line level at the rising edge of SCL except for the ACK period during data (including slave address) transmission in master transmit mode (when the SDA0 line is driven low while the internal SDA output is high (the SDA0 pin is in the high-impedance state))
- When a start condition is detected while the ICCR2.ST bit is 1 (requests to generate a start condition) or the internal SDA output state does not match the SDA0 line level
- When the ICCR2.ST bit is set to 1 (requests to generate a start condition) with the ICCR2.BBSY flag set to 1.

When NACK arbitration-lost detection is enabled: ICFER.NALE = 1

- When the internal SDA output state does not match the SDA0 line level at the rising edge of SCL in the ACK period during NACK transmission in receive mode

When slave arbitration-lost detection is enabled: ICFER.SALE = 1

- When the internal SDA output state does not match the SDA0 line level at the rising edge of SCL except for the ACK period during data transmission in slave transmit mode

[Clearing conditions]

- When 0 is written to the AL flag after reading AL = 1
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

Table 35.4 Relationship between Arbitration-Lost Generation Sources and Arbitration-Lost Enable Functions

ICFER			ICSR2		
MALE	NALE	SALE	AL	Error	Arbitration-Lost Generation Source
1	×	×	1	Start condition generation error	When internal SDA output state does not match SDA0 line level when a start condition is detected while the ICCR2.ST bit is 1
					When ICCR2.ST bit is set to 1 with ICCR2.BBSY flag set to 1
			1	Transmit data mismatch	When transmit data (including slave address) does not match the bus state in master transmit mode
×	1	×	1	NACK transmission mismatch	When ACK is detected during transmission of NACK in master receive mode or slave receive mode
×	×	1	1	Transmit data mismatch	When transmit data does not match the bus state in slave transmit mode

×: Don't care

START Flag (Start Condition Detection Flag)

[Setting condition]

- When a start condition (or a restart condition) is detected

[Clearing conditions]

- When 0 is written to the START bit after reading START = 1
- When a stop condition is detected
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

STOP Flag (Stop Condition Detection Flag)

[Setting condition]

- When a stop condition is detected

[Clearing conditions]

- When 0 is written to the STOP bit after reading STOP = 1
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

Downloaded from Arrow.com.

NACKF Flag (NACK Detection Flag)

[Setting condition]

- When ACK is not received (NACK is received) from the receiver in transmit mode with the ICFER.NACKE bit set to 1 (transfer suspension enabled)

[Clearing conditions]

- When 0 is written to the NACKF bit after reading NACKF = 1
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

Note: When the NACKF flag is set to 1, the RIIC suspends data transmission/reception. Writing to the ICDRT register in transmit mode or reading from the ICDRR register in receive mode with the NACKF flag set to 1 does not enable data transmit/receive operation. To restart data transmission/reception, set the NACKF flag to 0.

RDRF Flag (Receive Data Full Flag)

[Setting conditions]

- When receive data has been transferred from the ICDRS register to the ICDRR register
  This flag is set to 1 at the rising edge of the eighth or ninth SCL (selected by the ICMR3.RDRFS bit)
- When the received slave address matches after a start condition (or a restart condition) is detected with the ICCR2.TRS bit set to 0

[Clearing conditions]

- When 0 is written to the RDRF bit after reading RDRF = 1
- When data is read from the ICDRR register
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

TEND Flag (Transmission End Flag)

[Setting condition]

- At the rising edge of the ninth SCL while the TDRE flag is 1

[Clearing conditions]

- When 0 is written to the TEND bit after reading TEND = 1
- When data is written to the ICDRT register
- When a stop condition is detected
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

TDRE Flag (Transmit Data Empty Flag)

[Setting conditions]

- When data has been transferred from the ICDRT register to the ICDRS register and the ICDRT register becomes empty
- When the ICCR2.TRS bit is set to 1
- When the received slave address matches while the TRS bit is 1

[Clearing conditions]

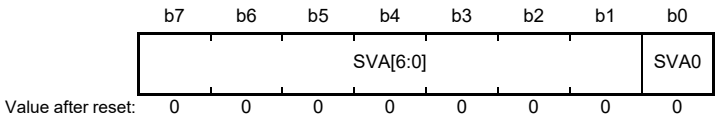
- When data is written to the ICDRT register
- When the ICCR2.TRS bit is set to 0
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

Note: The NACKF flag becoming 1 while the ICFER.NACKE bit is 1 suspends data transmission and reception by the RIIC. Even if the next data for transmission has already been written to the ICDRT register (the TDRE flag is 0), the data in the ICDRT register is retained but not transferred to the ICDRS register. At this point, the TDRE flag does not become 1.

Downloaded from Arrow.com.

35.2.11 Slave Address Register Ly (SARLy) (y = 0 to 2)

Address(es): RIIC0.SARL0 0008 830Ah, RIIC0.SARL1 0008 830Ch, RIIC0.SARL2 0008 830Eh



Bit	Symbol	Bit Name	Description	R/W
b0	SVA0	10-Bit Address LSB	Set a slave address	R/W
b7 to b1	SVA[6:0]	7-Bit Address/10-Bit Address Lower Bits	Set a slave address	R/W

SVA0 Bit (10-Bit Address LSB)

When the 10-bit address format is selected (SARUy.FS bit is 1), this bit functions as the LSB of a 10-bit address and forms the lower 8 bits of a 10-bit address in combination with the SVA[6:0] bits.
When the ICSER.SARyE bit is set to 1 (SARLy and SARUy enabled) and the SARUy.FS bit is 1, this bit is valid. While the SARUy.FS bit or SARyE bit is 0, the setting of this bit is ignored.

SVA[6:0] Bits (7-Bit Address/10-Bit Address Lower Bits)

When the 7-bit address format is selected (SARUy.FS bit is 0), these bits function as a 7-bit address. When the 10-bit address format is selected (SARUy.FS bit is 1), these bits function as the lower 8 bits of a 10-bit address in combination with the SVA0 bit.
While the ICSER.SARyE bit is 0, the setting of these bits is ignored.

Downloaded from Arrow.com.

35.2.12 Slave Address Register Uy (SARUy) (y = 0 to 2)

Address(es): RIIC0.SARU0 0008 830Bh, RIIC0.SARU1 0008 830Dh, RIIC0.SARU2 0008 830Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	SVA[1:0]		FS
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	FS	7-Bit/10-Bit Address Format Select	0: The 7-bit address format is selected. 1: The 10-bit address format is selected.	R/W
b2, b1	SVA[1:0]	10-Bit Address Upper Bits	Set a slave address	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

FS Bit (7-Bit/10-Bit Address Format Select)

This bit is used to select 7-bit address or 10-bit address for slave address y (in registers SARLy and SARUy).
When the ICSER.SARyE bit is set to 1 (registers SARLy and SARUy enabled) and the SARUy.FS bit is 0, the 7-bit address format is selected for slave address y, the SARLy.SVA[6:0] bits setting is valid, and the settings of the SVA[1:0] bits and the SARLy.SVA0 bit are ignored.
When the ICSER.SARyE bit is set to 1 (registers SARLy and SARUy enabled) and the SARUy.FS bit is 1, the 10-bit address format is selected for slave address y and the settings of the SVA[1:0] bits and SARLy are valid.
While the ICSER.SARyE bit is 0 (registers SARLy and SARUy disabled), the setting of the SARUy.FS bit is invalid.

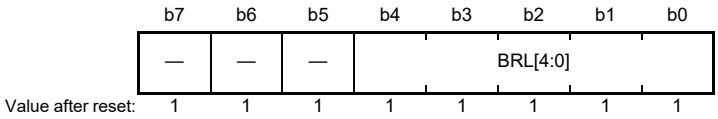
SVA[1:0] Bits (10-Bit Address Upper Bits)

When the 10-bit address format is selected (FS = 1), these bits function as the upper 2 bits of a 10-bit address.
When the ICSER.SARyE bit is set to 1 (SARLy and SARUy enabled) and the SARUy.FS bit is 1, these bits are valid.
While the SARUy.FS bit or SARyE bit is 0, the setting of these bits is ignored.

Downloaded from Arrow.com.

35.2.13 I2C-bus Bit Rate Low-Level Register (ICBRL)

Address(es): RIIC0.ICBRL 0008 8310h



Bit	Symbol	Bit Name	Description	R/W
b4 to b0	BRL[4:0]	Bit Rate Low Period	Low period of SCL	R/W
b7 to b5	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

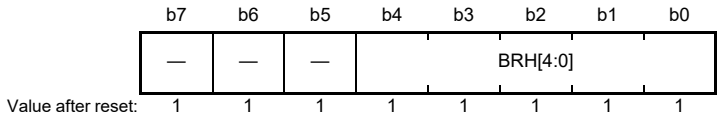
ICBRL is a 5-bit register to set the low period of SCL.
It also works to generate the data setup time for automatic SCL low-hold operation (refer to section 35.8, Automatic Low-Hold Function for SCL); when the RIIC is used only in slave mode, this register needs to be set to a value longer than the data setup time*1.
ICBRL counts the low period with the internal reference clock (IICφ) specified by the ICMR1.CKS[2:0] bits.
If the digital noise filter is enabled (the ICFER.NFE bit is 1), set the ICBRL register to a value at least one greater than the number of stages in the noise filter. Regarding the number of stages in the noise filter, see the description of the ICMR3.NF[1:0] bits.

Note 1. Data setup time (tSU: DAT)
250 ns (up to 100 kbps: Standard-mode (Sm))
100 ns (up to 400 kbps: Fast-mode (Fm))

Downloaded from Arrow.com.

35.2.14 I2C-bus Bit Rate High-Level Register (ICBRH)

Address(es): RIIC0.ICBRH 0008 8311h



Bit	Symbol	Bit Name	Description	R/W
b4 to b0	BRH[4:0]	Bit Rate High Period	High period of SCL	R/W
b7 to b5	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

ICBRH is a 5-bit register to set the high period of SCL. ICBRH is valid in master mode. If the RIIC is used only in slave mode, this register need not to set the high period.
ICBRH counts the high period with the internal reference clock (IICφ) specified by the ICMR1.CKS[2:0] bits.
If the digital noise filter is enabled (the ICFER.NFE bit is 1), set the ICBRH register to a value at least one greater than the number of stages in the noise filter. Regarding the number of stages in the noise filter, see the description of the ICMR3.NF[1:0] bits.

The I2C transfer rate and the SCL duty are calculated using the following expression.
Transfer rate = 1 / {[(ICBRH + 1) + (ICBRL + 1)] / IICφ*1 + SCL0 line rise time [tr] + SCL0 line fall time [tf]}
Duty cycle = {SCL0 line rise time [tr]*2 + (ICBRH + 1) / IICφ} / {SCL0 line fall time [tf]*2 + (ICBRL + 1) / IICφ}

Note 1. IICφ = PCLK × Division ratio
Note 2. The SCL0 line rise time [tr] and SCL0 line fall time [tf] depend on the total bus line capacitance [Cb] and the pull-up resistor [Rp]. For details, refer to the I2C-bus specification from NXP Semiconductors.

Table 35.5 lists examples of ICBRH/ICBRL settings.

Downloaded from Arrow.com.

Table 35.5 Examples of ICBRH/ICBRL Settings for Transfer Rate

Transfer Rate (kbps)	Operating Frequency PCLK (MHz)								
	8			10			12.5		
	CKS[2:0]	ICBRH	ICBRL	CKS[2:0]	ICBRH	ICBRL	CKS[2:0]	ICBRH	ICBRL
10	100b	22 (F6h)	25 (F9h)	101b	13 (EDh)	15 (EFh)	101b	16 (F0h)	20 (F4h)
50	010b	16 (F0h)	19 (F3h)	010b	21 (F5h)	24 (F8h)	011b	12 (ECh)	15 (EFh)
100	001b	15 (EFh)	18 (F2h)	001b	19 (F3h)	23 (F7h)	001b	24 (F8h)	29 (FDh)
400	000b	4 (E4h)	10 (EAh)	000b	5 (E5h)	12 (ECh)	000b	7 (E7h)	16 (F0h)

Transfer Rate (kbps)	Operating Frequency PCLK (MHz)								
	16			20			25		
	CKS[2:0]	ICBRH	ICBRL	CKS[2:0]	ICBRH	ICBRL	CKS[2:0]	ICBRH	ICBRL
10	101b	22 (F6h)	25 (F9h)	110b	13 (EDh)	15 (EFh)	110b	16 (F0h)	20 (F4h)
50	011b	16 (F0h)	19 (F3h)	011b	21 (F5h)	24 (F8h)	100b	12 (ECh)	15 (EFh)
100	010b	15 (EFh)	18 (F2h)	010b	19 (F3h)	23 (F7h)	010b	24 (F8h)	29 (FDh)
400	000b	9 (E9h)	20 (F4h)	000b	11 (EBh)	25 (F9h)	001b	7 (E7h)	16 (F0h)

Transfer Rate (kbps)	Operating Frequency PCLK (MHz)					
	30			32		
	CKS[2:0]	ICBRH	ICBRL	CKS[2:0]	ICBRH	ICBRL
10	110b	20 (F4h)	24 (F8h)	110b	22 (F6h)	25 (F9h)
50	100b	15 (EFh)	18 (F2h)	100b	16 (F0h)	19 (F3h)
100	011b	14 (EEh)	17 (F1h)	011b	15 (EFh)	18 (F2h)
400	001b	8 (E8h)	19 (F3h)	001b	9 (E9h)	20 (F4h)

Note: ICBRH/ICBRL settings in these tables are calculated using the following values:
SCL0 line rise time (tr): 100 kbps or less (Sm): 1000 ns, 400 kbps or less (Fm): 300 ns
SCL0 line fall time (tf): 400 kbps or less (Sm/Fm): 300 ns
For the specified values of rise time (tr) and fall time (tf) of the SCL0 signal, refer to the I2C-bus specification from NXP Semiconductors.

Downloaded from Arrow.com.

35.2.15 I2C-bus Transmit Data Register (ICDRT)

Address(es): RIIC0.ICDRT 0008 8312h

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1

When the ICDRT register detects a space in the I2C-bus shift register (ICDRS), it transfers the transmit data that has been written to the ICDRT register to the ICDRS register and starts transmitting data in transmit mode.
The double-buffer structure of the ICDRT register and the ICDRS register allows continuous transmit operation if the next transmit data has been written to the ICDRT register while the ICDRS register data is being transmitted.
The ICDRT register can always be read and written. Write transmit data to the ICDRT register once when a transmit data empty interrupt (TXI) request is generated.

35.2.16 I2C-bus Receive Data Register (ICDRR)

Address(es): RIIC0.ICDRR 0008 8313h

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0

When 1 byte of data has been received, the received data is transferred from the I2C-bus shift register (ICDRS) to the ICDRR register to enable the next data to be received.
The double-buffer structure of the ICDRS register and the ICDRR register allows continuous receive operation if the received data has been read from the ICDRR register while the ICDRS register is receiving data.
The ICDRR register cannot be written. Read data from the ICDRR register once when a receive data full interrupt (RXI) request is generated.
If the ICDRR register receives the next receive data before the current data is read from the ICDRR register (while the ICSR2.RDRF flag is 1), the RIIC automatically holds the SCL line low one cycle before the RDRF flag is set to 1 next.

35.2.17 I2C-bus Shift Register (ICDRS)

The ICDRS register is an 8-bit shift register to transmit and receive data.
During transmission, transmit data is transferred from the ICDRT register to the ICDRS register and is sent from the SDA0 pin. During reception, data is transferred from the ICDRS register to the ICDRR register after 1 byte of data has been received.
The ICDRS register cannot be accessed directly.

Downloaded from Arrow.com.

35.3 Operation

35.3.1 Communication Data Format

The I2C-bus format consists of 8-bit data and 1-bit acknowledgment. The first byte following a start condition or restart condition is an address byte used to specify a slave device with which the master device communicates. The specified slave is valid until a new slave is specified or a stop condition is generated.
Figure 35.2 shows the I2C-bus format, and Figure 35.3 shows the I2C-bus timing.

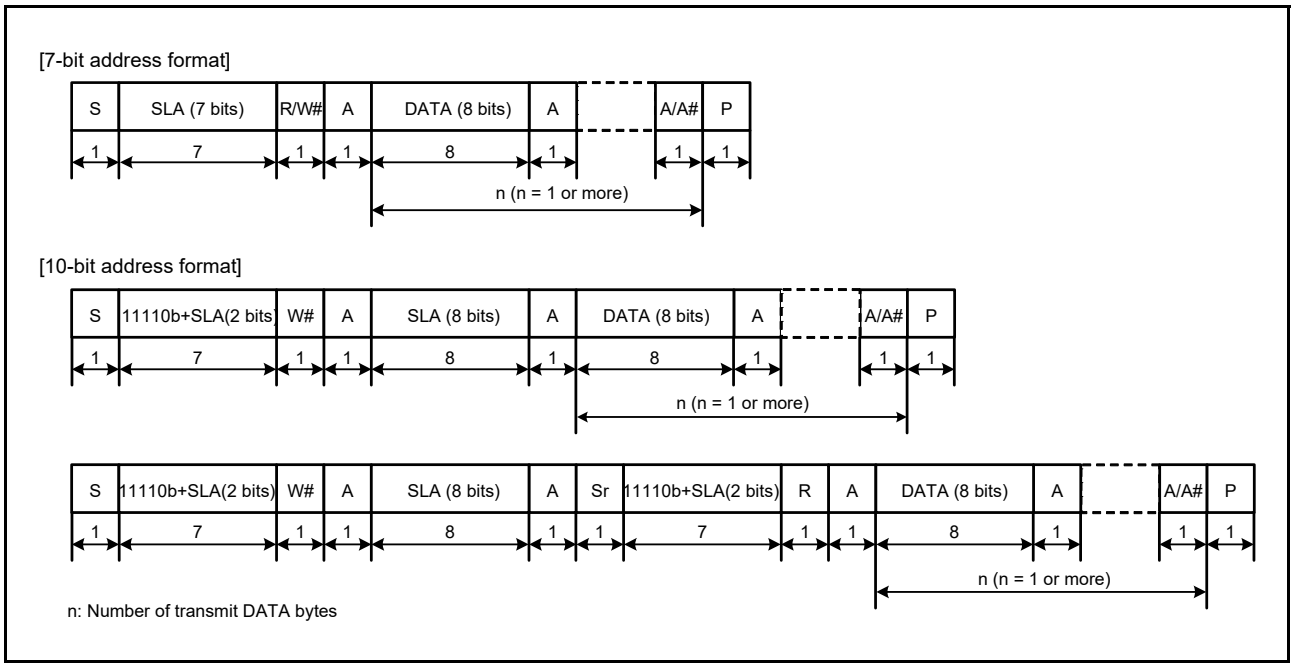


Figure 35.2 I2C-bus Format

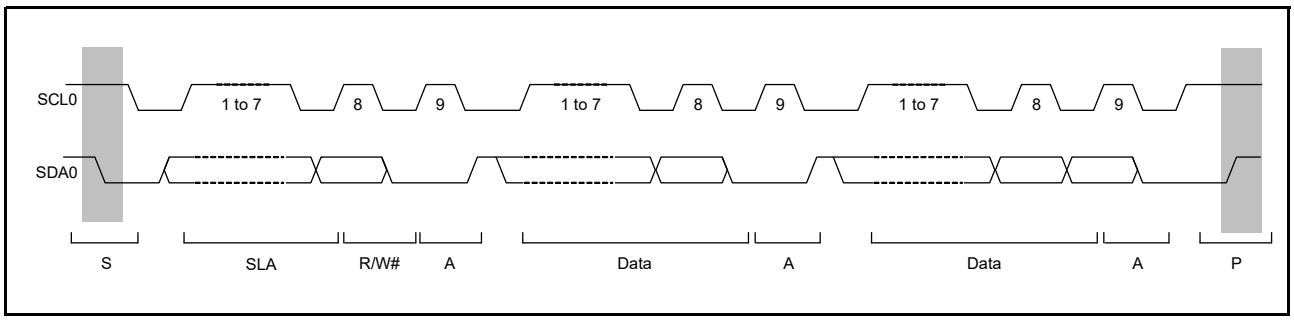


Figure 35.3 I2C-bus Timing (SLA = 7 Bits)

S: Start condition. The master device drives the SDA0 line low from high while the SCL0 line is high.

SLA: Slave address, by which the master device selects a slave device.

R/W#: Indicates the direction of data transfer: from the slave device to the master device when R/W is 1, or from the master device to the slave device when R/W is 0.

A: Acknowledge. The receiver drives the SDA0 line low. (In master transmit mode, the slave device returns acknowledge. In master receive mode, the master device returns acknowledge.)

A#: Not Acknowledge. The receiver drives the SDA0 line high.

Sr: Restart condition. The master device drives the SDA0 line low from high after the setup time has elapsed with the SCL0 line high.

DATA: Transmitted or received data

P: Stop condition. The master device drives the SDA0 line high from low while the SCL0 line is high.

Downloaded from Arrow.com.

35.3.2 Initial Settings

Before starting data transmission and reception, initialize the RIIC according to the procedure in Figure 35.4.
Set the ICCR1.ICE bit to 1 (internal reset) after setting the ICCR1.IICRST bit to 1 (RIIC reset) with the ICCR1.ICE bit set to 0 (SCL0 and SDA0 pins in inactive state). This initializes the various flags and internal state of the ICSR1 register. After that, set registers SARLy, SARUy, ICSER, ICMR1, ICBRH, and ICBRL (y = 0 to 2), and set the other registers as necessary (for initial settings of the RIIC, see Figure 35.4). When the necessary register settings have been completed, set the ICCR1.IICRST bit to 0 (releases the RIIC reset). This step is not necessary if initialization of the RIIC has already been completed.

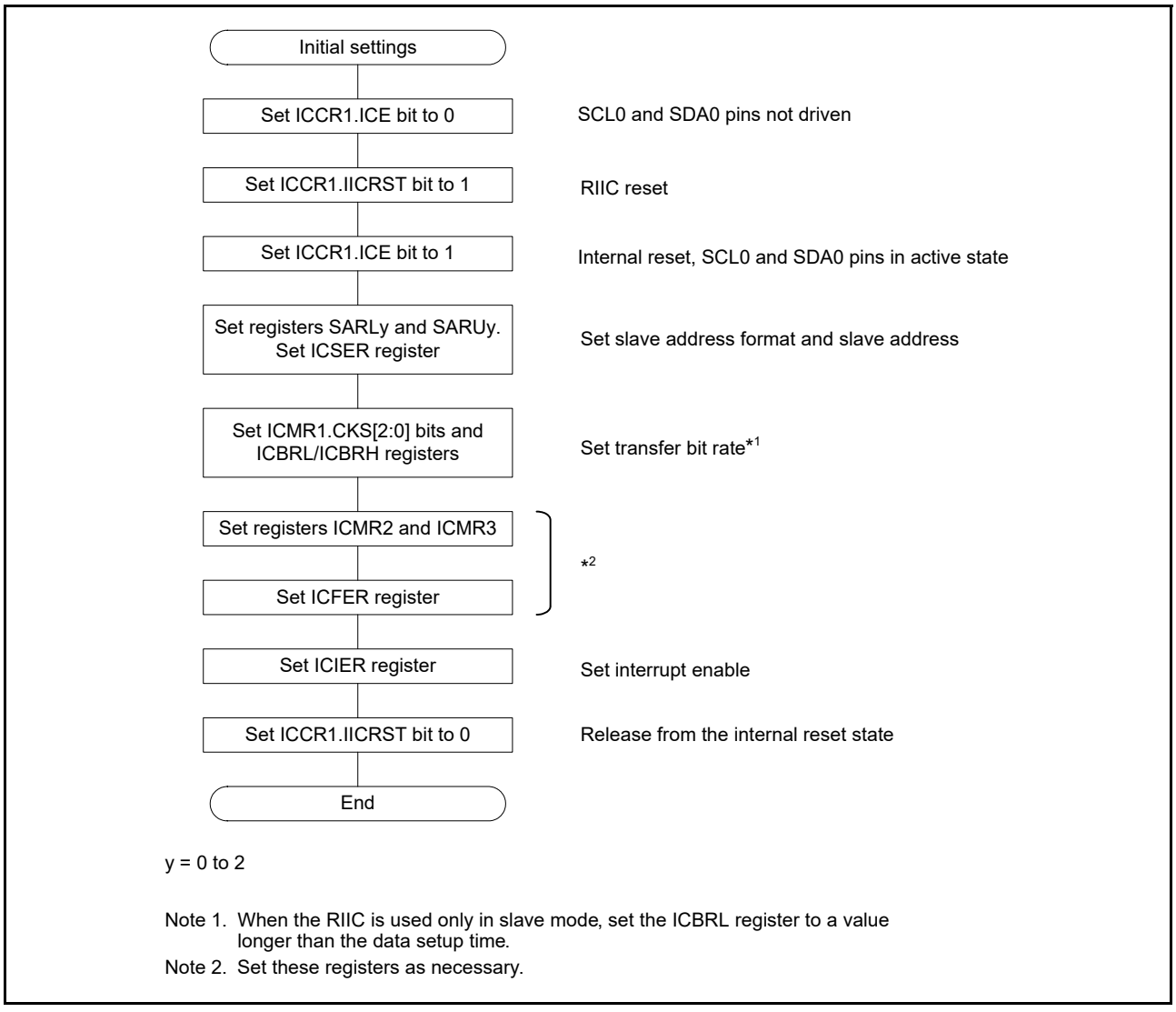


Figure 35.4 Example of RIIC Initialization Flowchart

Downloaded from Arrow.com.

35.3.3 Master Transmit Operation

In master transmit operation, the RIIC generates clock signals and sends data as the master device, and the slave device returns acknowledgments. Figure 35.5 shows an example of usage of master transmission and Figure 35.6 to Figure 35.8 show the timing of operations in master transmission.
The following describes the procedure and operations for master transmission.

(1) Initial settings. For details, refer to section 35.3.2, Initial Settings.
(2) Read the ICCR2.BBSY flag to check that the bus is open, and then set the ICCR2.ST bit to 1 (requests to generate a start condition). Upon receiving the request, the RIIC generates a start condition. At the same time, the BBSY flag and the ICSR2.START flag are automatically set to 1 and the ST bit is automatically set to 0. At this time, if the start condition is detected and the internal levels for the SDA output state and the levels on the SDA0 line have matched while the ST bit is 1, the RIIC recognizes that generating of the start condition as requested by the ST bit has been successfully completed, and bits MST and TRS in the ICCR2 register are automatically set to 1, placing the RIIC in master transmit mode. The ICSR2.TDRE flag is also automatically set to 1 in response to setting of the TRS bit to 1.
(3) Check that the ICSR2.TDRE flag is 1, and then write the value for transmission (the slave address and the R/W# bit) to the ICDRT register. Once the data for transmission are written to the ICDRT register, the TDRE flag is automatically set to 0, the data are transferred from the ICDRT register to the ICDRS register, and the TDRE flag is again set to 1. After the byte containing the slave address and R/W# bit has been transmitted, the value of the TRS bit is automatically updated to select master transmit or master receive mode in accord with the value of the transmitted R/W# bit. If the value of the R/W# bit was 0, the RIIC continues in master transmit mode.
Because the ICSR2.NACKF flag being 1 at this time indicates that no slave device recognized the address or there was an error in communications, write 1 to the ICCR2.SP bit to generate a stop condition.
For data transmission with an address in the 10-bit format, start by writing 1111 0b, the 2 higher-order bits of the slave address, and W to the ICDRT register as the first address transmission. Then, as the second address transmission, write the 8 lower-order bits of the slave address to the ICDRT register.
(4) After confirming that the ICSR2.TDRE flag is 1, write the data for transmission to the ICDRT register. The RIIC automatically holds the SCL0 line low until the data for transmission are ready or a stop condition is generated.
(5) After all bytes of data for transmission have been written to the ICDRT register, wait until the ICSR2.NACKF or ICSR2.TEND flag becomes 1, and then set the ICCR2.SP bit to 1 (requests to generate a stop condition). Upon receiving a stop condition generation request, the RIIC generates the stop condition.
(6) Upon detecting the stop condition, the RIIC automatically sets bits MST and TRS in the ICCR2 register to 00b and enters slave receive mode. Furthermore, it automatically sets the TDRE and TEND flags to 0, and sets the ICSR2.STOP flag to 1.
(7) After checking that the ICSR2.STOP flag is 1, set the ICSR2.NACKF and STOP flags to 0 for the next transfer operation.

Downloaded from Arrow.com.

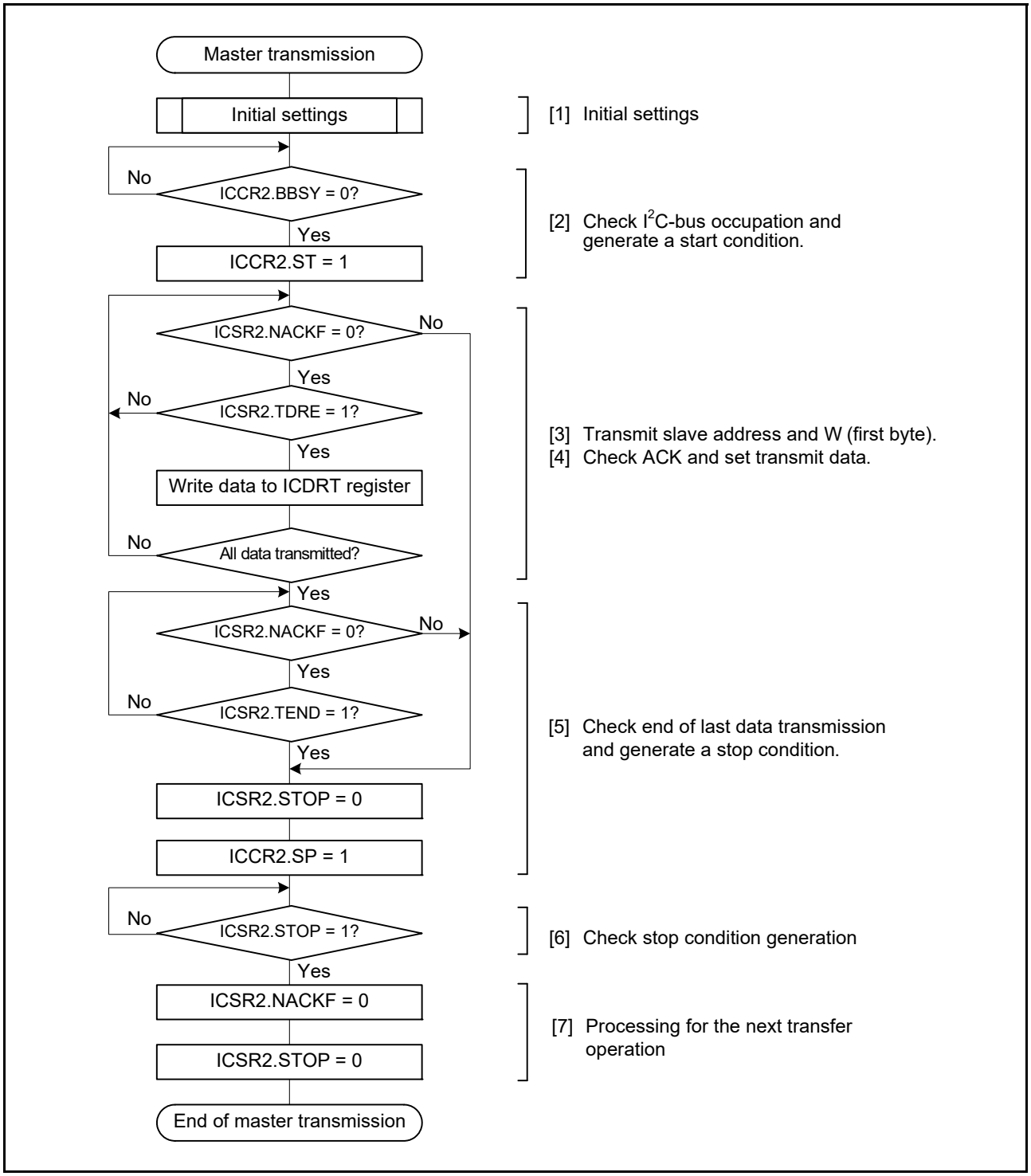


Figure 35.5 Example of Master Transmission Flowchart

Downloaded from Arrow.com.

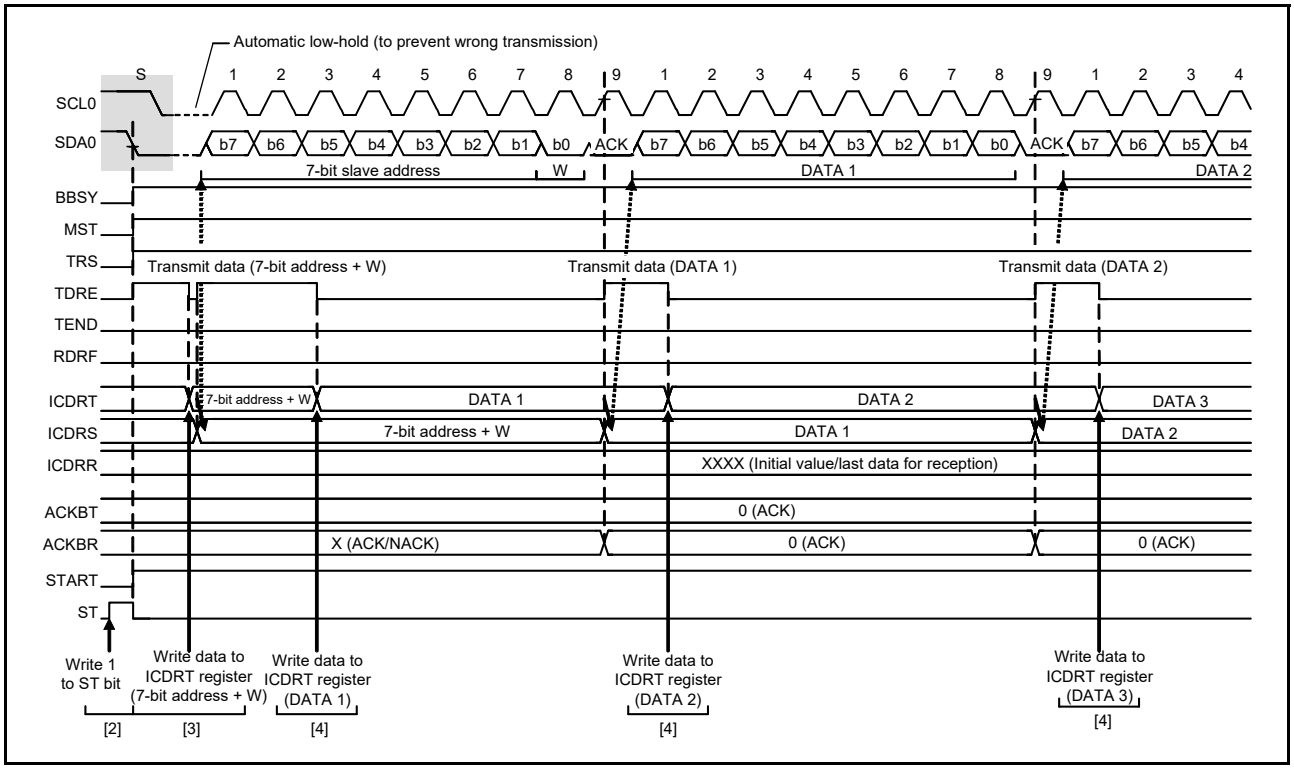


Figure 35.6 Master Transmit Operation Timing (1) (7-Bit Address Format)

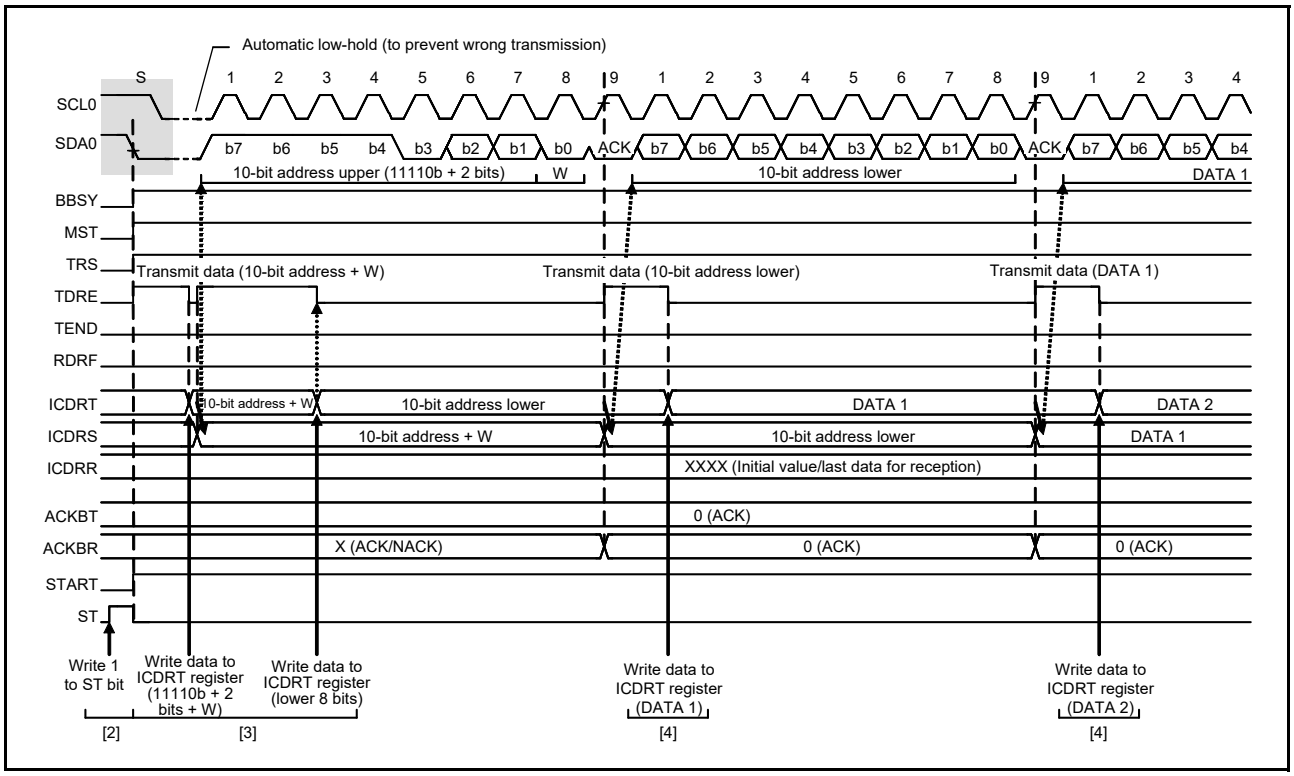


Figure 35.7 Master Transmit Operation Timing (2) (10-Bit Address Format)

Downloaded from Arrow.com.

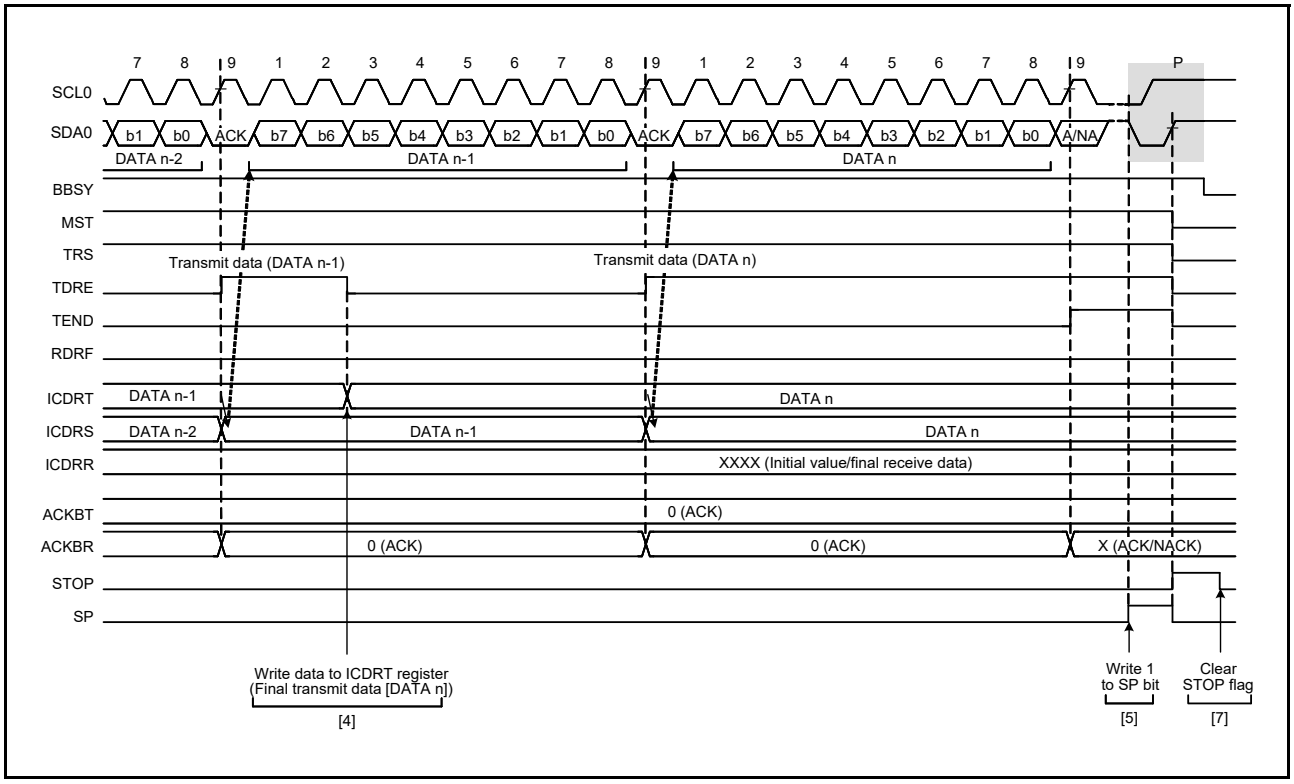


Figure 35.8 Master Transmit Operation Timing (3)

35.3.4 Master Receive Operation

In master receive operation, the RIIC as a master device generates clock signals, receives data from the slave device, and returns acknowledgments. Because the RIIC must start by sending a slave address to the corresponding slave device, this part of the procedure is performed in master transmit mode, but the subsequent steps are in master receive mode.
Figure 35.9 and Figure 35.10 show examples of usage of master reception (7-bit address format) and Figure 35.11 to Figure 35.13 show the timing of operations in master reception.
The following describes the procedure and operations for master reception.

(1) Initial settings. For details, refer to section 35.3.2, Initial Settings.

(2) Read the ICCR2.BBSY flag to check that the bus is open, and then set the ICCR2.ST bit to 1 (requests to generate a start condition). Upon receiving the request, the RIIC generates a start condition. When the RIIC detects the start condition, the BBSY flag and the ICSR2.START flag are automatically set to 1 and the ST bit is automatically set to 0. At this time, if the start condition is detected and the levels for the SDA output and the levels on the SDA0 line have matched while the ST bit is 1, the RIIC recognizes that generating of the start condition as requested by the ST bit has been successfully completed, and bits MST and TRS in the ICCR2 register are automatically set to 1, placing the RIIC in master transmit mode. The ICSR2.TDRE flag is also automatically set to 1 in response to setting of the TRS bit to 1.

(3) Check that the ICSR2.TDRE flag is 1, and then write the value for transmission (the first byte indicates the slave address and value of the R/W# bit) to the ICDRT register. Once the data for transmission are written to the ICDRT register, the TDRE flag is automatically set to 0, the data are transferred from the ICDRT register to the ICDRS register, and the TDRE flag is again set to 1. Once the byte containing the slave address and R/W# bit has been transmitted, the value of the ICCR2.TRS bit is automatically updated to select transmit or receive mode in accord with the value of the transmitted R/W# bit. If the value of the R/W# bit was 1, the TRS bit is set to 0 on the rising edge of the ninth SCL, placing the RIIC in master receive mode. At this time, the TDRE flag is set to 0 and the ICSR2.RDRF flag is automatically set to 1.

Downloaded from Arrow.com.

Because the ICSR2.NACKF flag being 1 at this time indicates that no slave device recognized the address or there was an error in communications, write 1 to the ICCR2.SP bit to generate a stop condition.
For master reception from a device with a 10-bit address, start by using master transmission to send the 10-bit address, and then generate a restart condition. After that, transmitting 1111 0b, the two higher-order bits of the slave address, and the R bit places the RIIC in master receive mode.

(4) Dummy read the ICDRR register after confirming that the ICSR2.RDRF flag is 1; this makes the RIIC start output of the SCL and start data reception.

(5) After 1 byte of data has been received, the ICSR2.RDRF flag is set to 1 on the rising edge of the eighth or ninth SCL (the clock signal) as selected by the ICMR3.RDRFS bit. Reading the ICDRR register at this time will produce the received data, and the RDRF flag is automatically set to 0 at the same time. Furthermore, the value of the acknowledgment bit received during the ninth SCL is returned as the value set in the ICMR3.ACKBT bit. Furthermore, if the next byte to be received is the next to last byte, set the ICMR3.WAIT bit to 1 (for wait insertion) before reading the ICDRR register (containing the second byte from last). As well as enabling NACK output even in the case of delays in processing to set the ICMR3.ACKBT bit to 1 (NACK) in step (6), due to other interrupts, etc., this fixes the SCL0 line to low on the falling edge of the ninth clock pulse in reception of the last byte, so the state is such that generating a stop condition is possible.

(6) When the ICMR3.RDRFS bit is 0 and the slave device must be notified that it is to end transfer for data reception after transfer of the next (final) byte, set the ICMR3.ACKBT bit to 1 (NACK).

(7) After reading the byte before last from the ICDRR register, if the value of the ICSR2.RDRF flag is confirmed to be 1, write 1 to the ICCR2.SP bit (requests to generate a stop condition) and then read the last byte from the ICDRR register. When the ICDRR register is read, the RIIC is released from the wait state and generates the stop condition after low-level output in the ninth clock pulse is completed or the SCL0 line is released from the low-hold state.

(8) Upon detecting the stop condition, the RIIC automatically sets bits MST and TRS in the ICCR2 register to 00b and enters slave receive mode. Furthermore, detection of the stop condition leads to setting of the ICSR2.STOP flag to 1.

(9) After checking that the ICSR2.STOP flag is 1, set the ICSR2.NACKF and STOP flags to 0 for the next transfer operation.

Downloaded from Arrow.com.

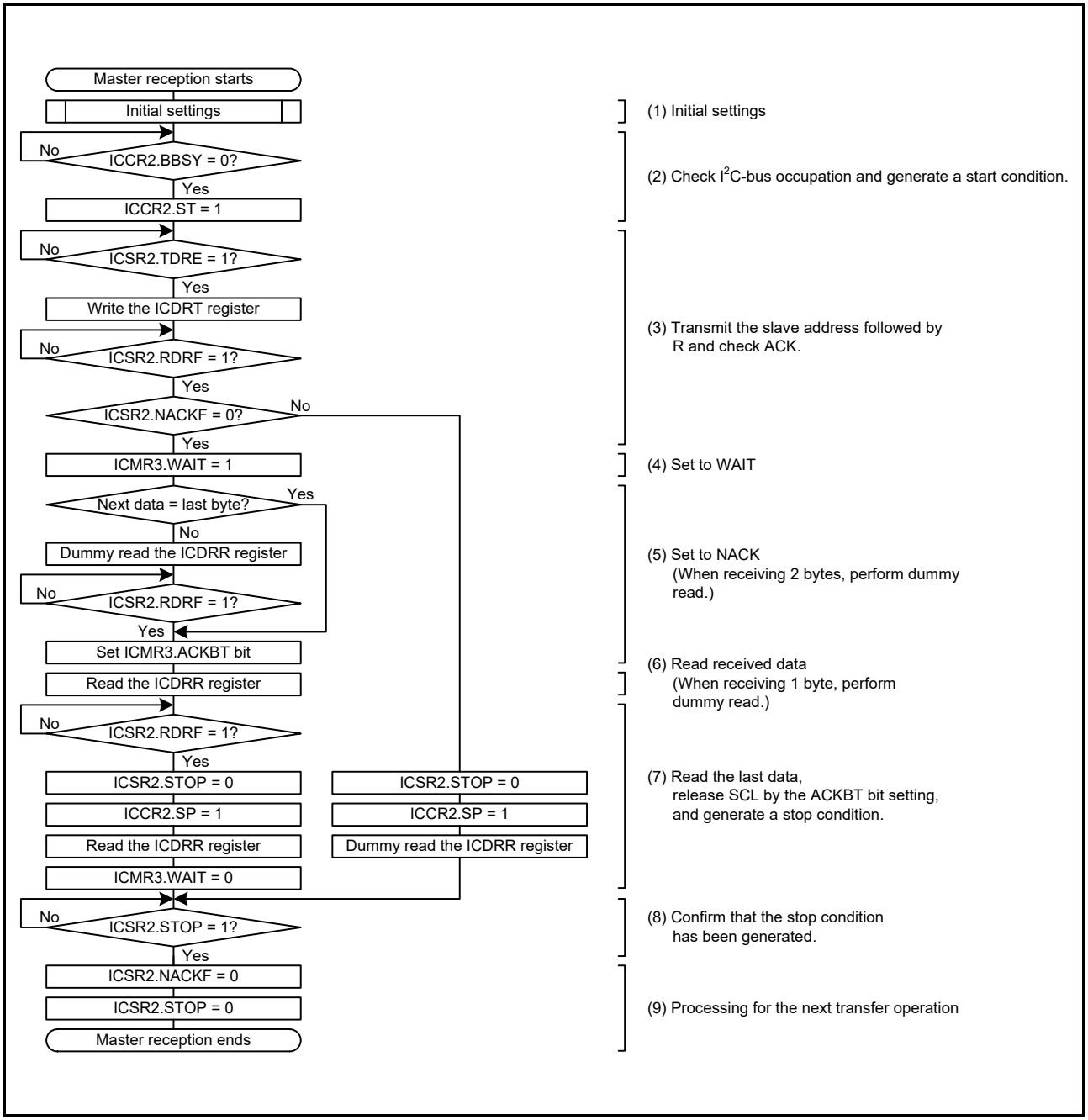


Figure 35.9 Example of Master Reception (7-Bit Address Format, 1 or 2 bytes)

Downloaded from Arrow.com.

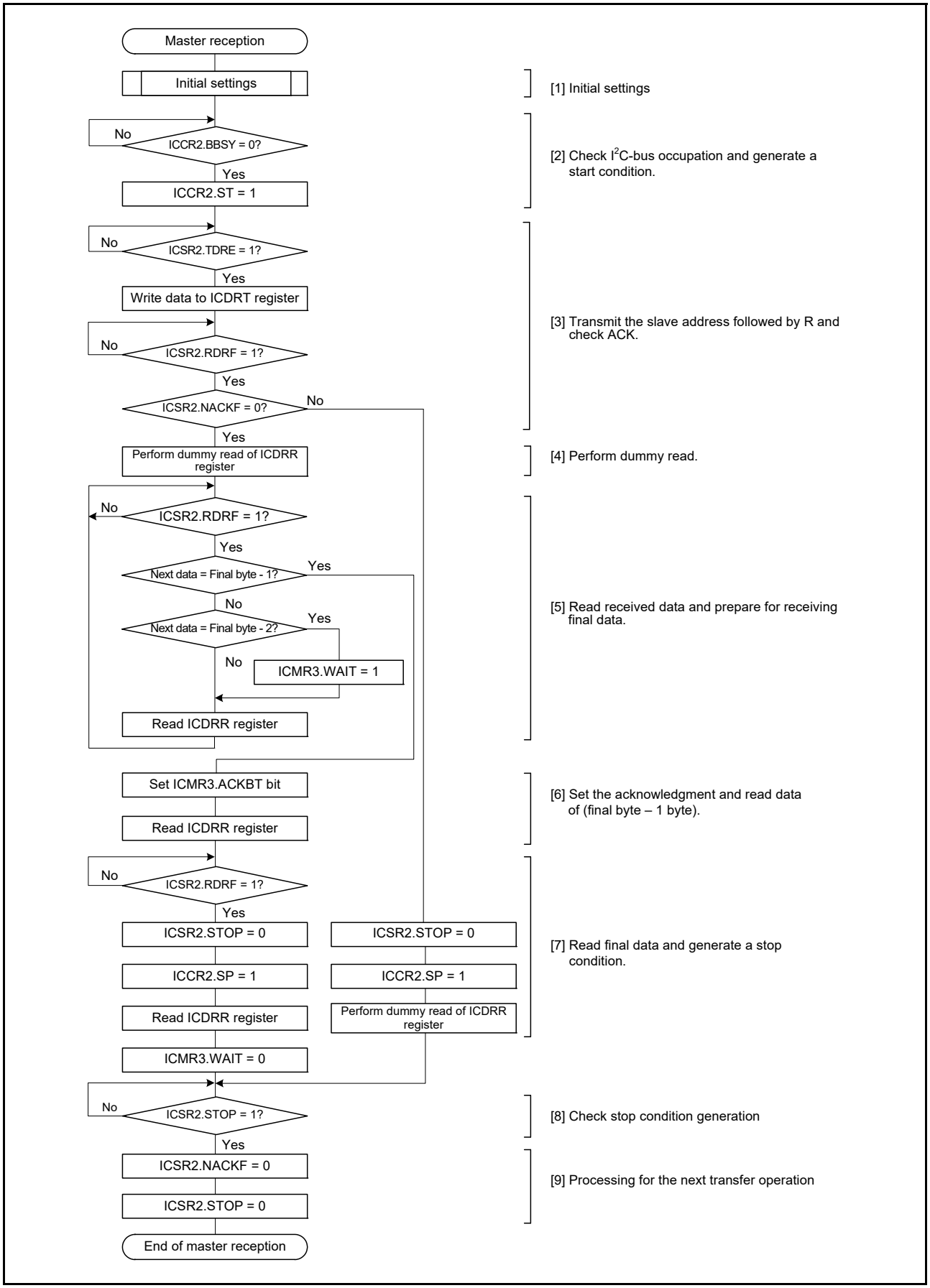


Figure 35.10 Example of Master Reception (7-Bit Address Format, 3 Bytes or More)

Downloaded from Arrow.com.

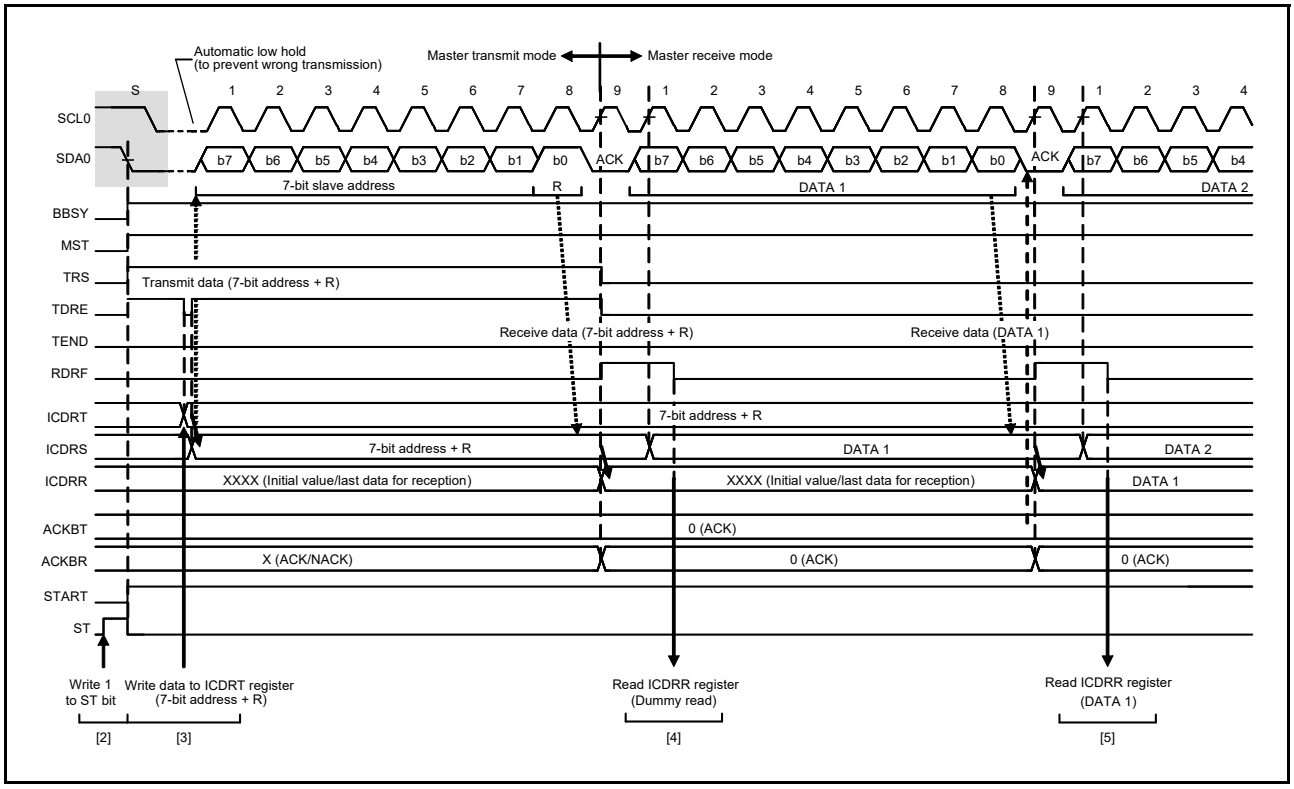


Figure 35.11 Master Receive Operation Timing (1) (7-Bit Address Format, When RDRFS bit is 0)

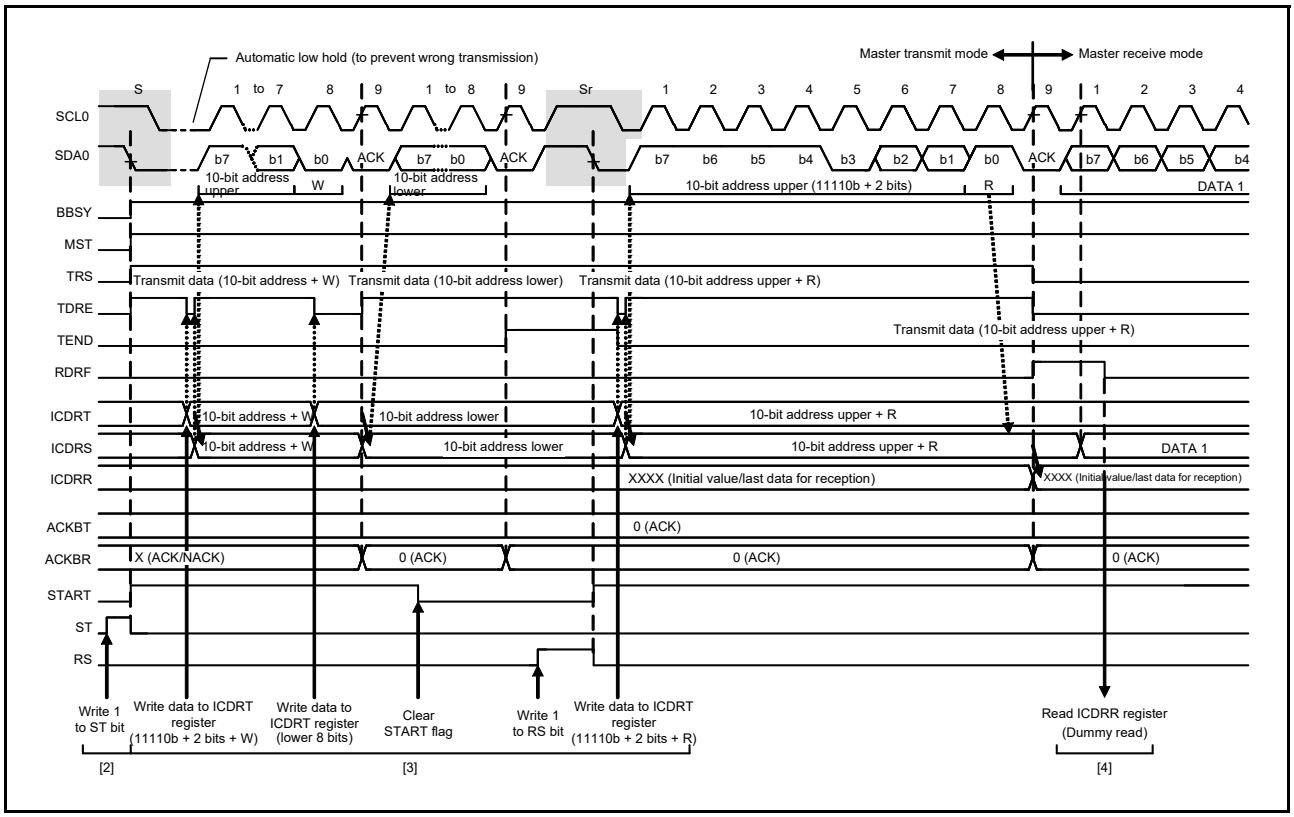


Figure 35.12 Master Receive Operation Timing (2) (10-Bit Address Format, When RDRFS bit is 0)

Downloaded from Arrow.com.

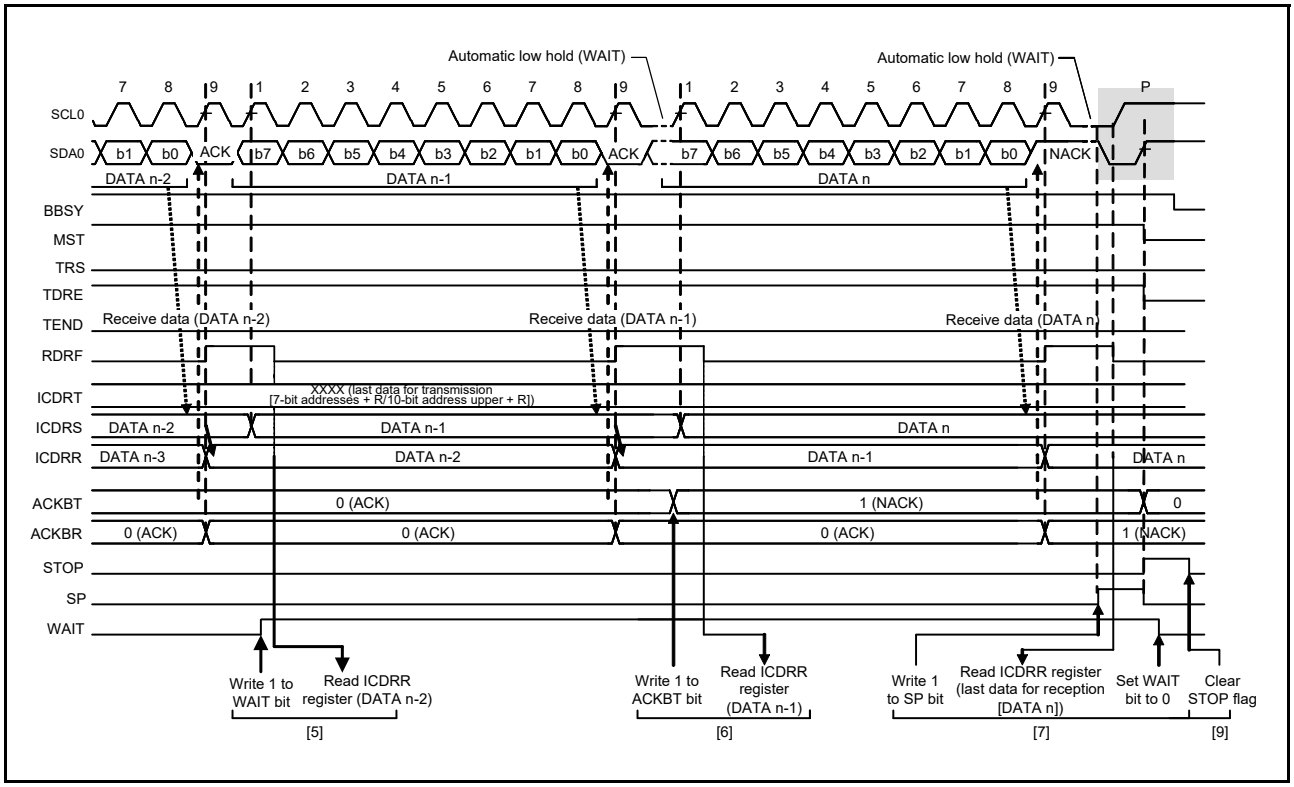


Figure 35.13 Master Receive Operation Timing (3) (When RDRFS bit is 0)

Downloaded from Arrow.com.

35.3.5 Slave Transmit Operation

In slave transmit operation, the master device outputs the SCL, the RIIC transmits data as a slave device, and the master device returns acknowledgments.

Figure 35.14 shows an example of usage of slave transmission and Figure 35.15 and Figure 35.16 show the timing of operations in slave transmission.

The following describes the procedure and operations for slave transmission.

(1) Initial settings. For details, refer to section 35.3.2, Initial Settings.
After initial settings, the RIIC will stay in the standby state until it receives a slave address that it matches.

(2) After receiving a matching slave address, the RIIC sets one of the corresponding bits ICSR1.HOA, GCA, and AASy (y = 0 to 2) to 1 on the rising edge of the ninth SCL (the clock signal) and outputs the value set in the ICMR3.ACKBT bit to the acknowledgment bit on the ninth SCL. If the value of the R/W# bit that was also received at this time is 1, the RIIC automatically places itself in slave transmit mode by setting both the ICCR2.TRS bit and the ICSR2.TDRE flag to 1.

(3) After the ICSR2.TDRE flag is confirmed to be 1, write the data for transmission to the ICDRT register. At this time, if the RIIC does not receive ACK from the master device (receives an NACK signal) while the ICFER.NACKE bit is 1, the RIIC suspends transfer of the next data.

(4) Wait until the ICSR2.TEND flag is set to 1 while the ICSR2.TDRE flag is 1, after the ICSR2.NACKF flag is set to 1 or the last byte for transmission is written to the ICDRT register. When the ICSR2.NACKF flag or the TEND flag is 1, the RIIC drives the SCL0 line low on the falling edge of ninth SCL.

(5) When the ICSR2.NACKF flag or the ICSR2.TEND flag is 1, dummy read the ICDRR register to complete the processing. This releases the SCL0 line.

(6) Upon detecting the stop condition, the RIIC automatically sets bits ICSR1.HOA, GCA, and AASy (y = 0 to 2), flags ICSR2.TDRE and TEND, and the ICCR2.TRS bit to 0, and enters slave receive mode.

(7) After checking that the ICSR2.STOP flag is 1, set the ICSR2.NACKF and STOP flags to 0 for the next transfer operation.

Downloaded from Arrow.com.

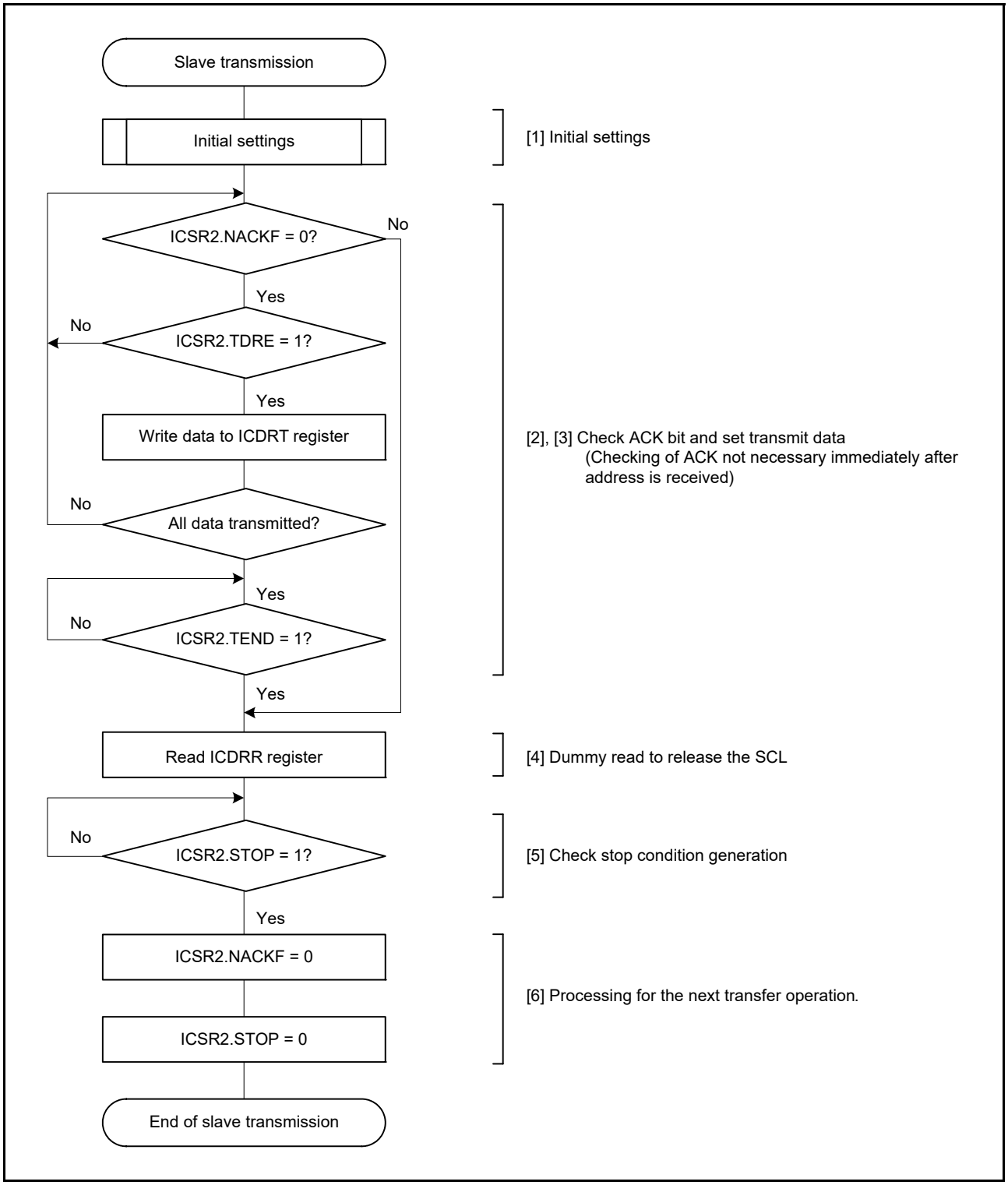


Figure 35.14 Example of Slave Transmission Flowchart

Downloaded from Arrow.com.

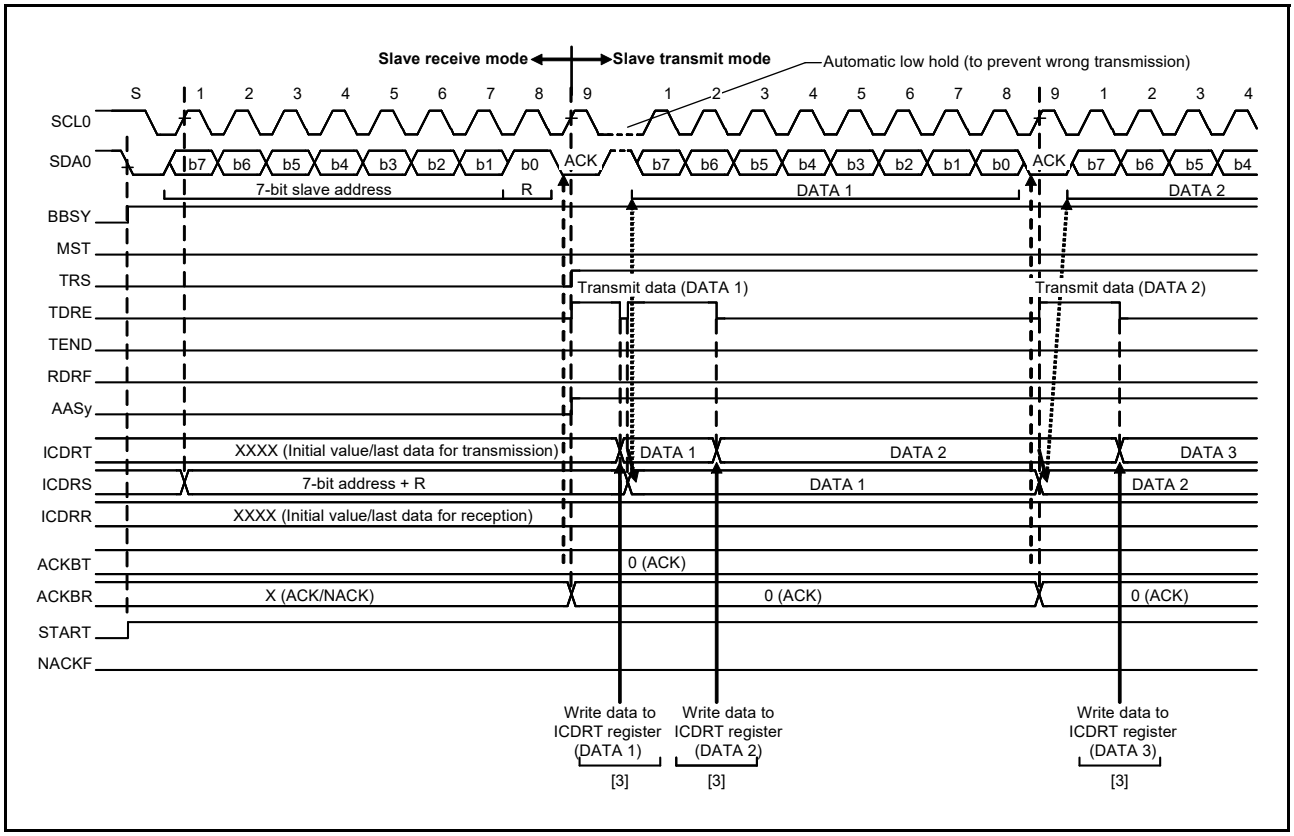


Figure 35.15 Slave Transmit Operation Timing (1) (7-Bit Address Format)

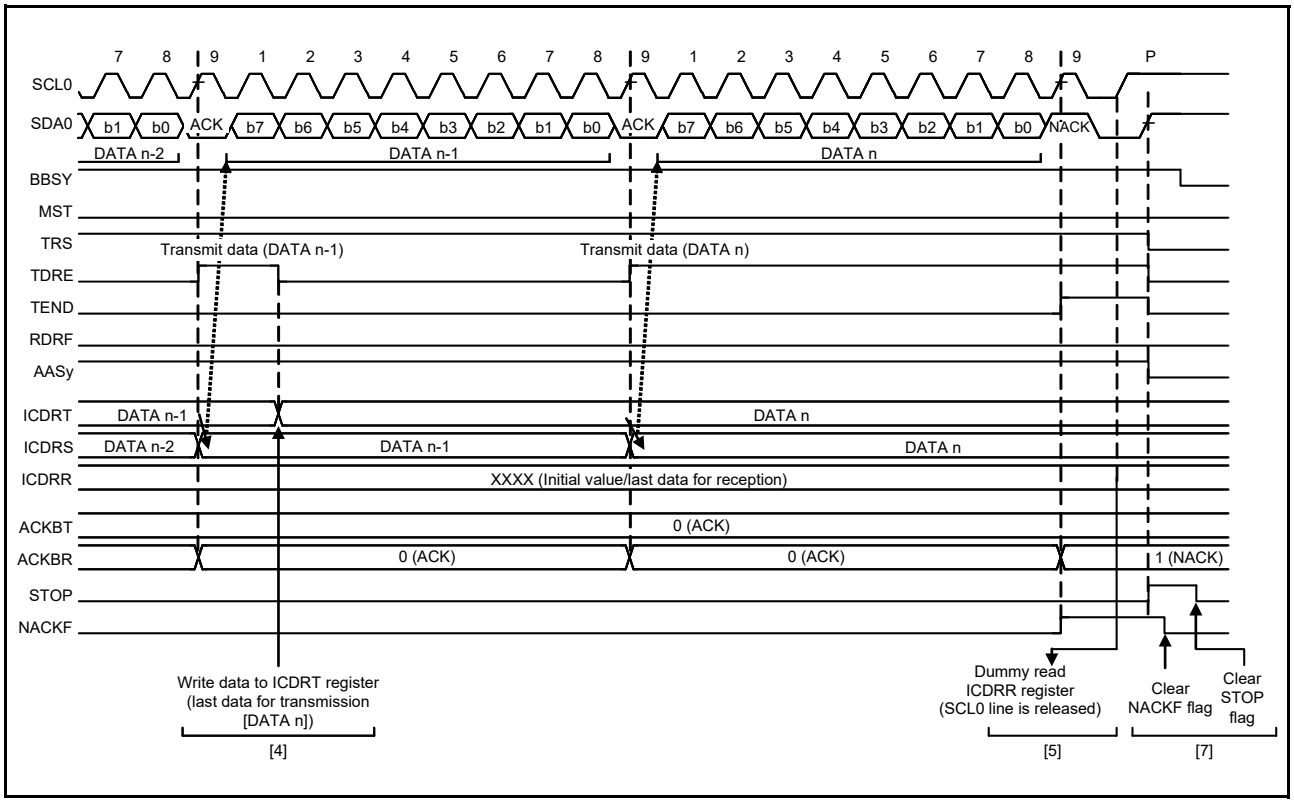


Figure 35.16 Slave Transmit Operation Timing (2)

Downloaded from Arrow.com.

35.3.6 Slave Receive Operation

In slave receive operation, the master device outputs the SCL and transmit data, and the RIIC returns acknowledgments as a slave device.

Figure 35.17 shows an example of usage of slave reception and Figure 35.18 and Figure 35.19 show the timing of operations in slave reception.

The following describes the procedure and operations for slave reception.

(1) Initial settings. For details, refer to section 35.3.2, Initial Settings.
After initial settings, the RIIC will stay in the standby state until it receives a slave address that it matches.

(2) After receiving a matching slave address, the RIIC sets one of the corresponding bits ICSR1.HOA, GCA, and AASy (y = 0 to 2) to 1 on the rising edge of the ninth SCL (the clock signal) and outputs the value set in the ICMR3.ACKBT bit to the acknowledgment bit on the ninth SCL. If the value of the R/W# bit that was also received at this time is 0, the RIIC continues to place itself in slave receive mode and sets the ICSR2.RDRF flag to 1.

(3) After the ICSR2.STOP flag is confirmed to be 0 and the ICSR2.RDRF flag to be 1, dummy read the ICDRR register (the dummy value consists of the slave address and R/W# bit when the 7-bit address format is selected, or the lower 8 bits when the 10-bit address format is selected).

(4) When the ICDRR register is read, the RIIC automatically sets the ICSR2.RDRF flag to 0. If reading of the ICDRR register is delayed and a next byte is received while the RDRF flag is still set to 1, the RIIC holds the SCL0 line low from one SCL cycle before the timing with which RDRF should be set. In this case, reading the ICDRR register releases the SCL0 line from being held low.
When the ICSR2.STOP flag is 1 and the ICSR2.RDRF flag is also 1 or when all the data is completely received, read the ICDRR register.

(5) Upon detecting the stop condition, the RIIC automatically clears bits ICSR1.HOA, GCA, and AASy (y = 0 to 2) to 0.

(6) After checking that the ICSR2.STOP flag is 1, set the ICSR2.STOP flag to 0 for the next transfer operation.

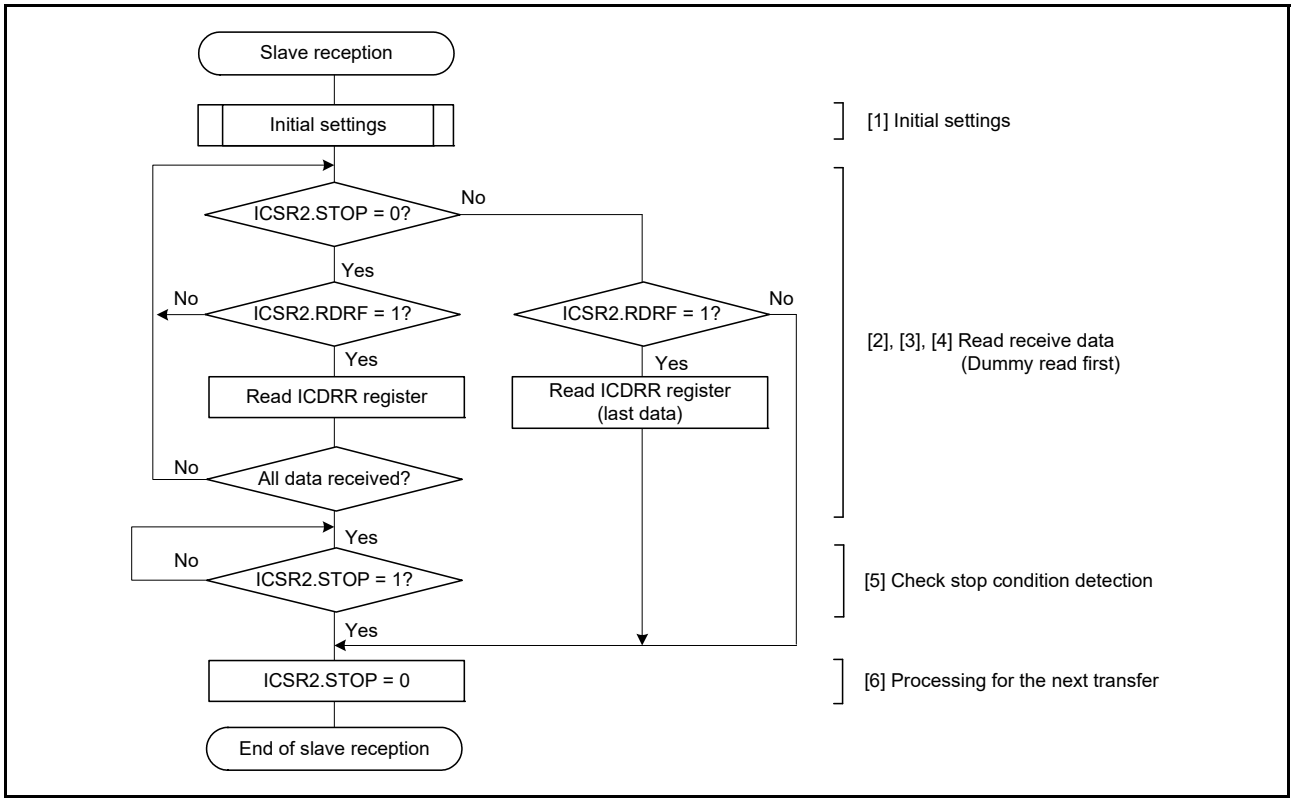


Figure 35.17 Example of Slave Reception Flowchart

Downloaded from Arrow.com.

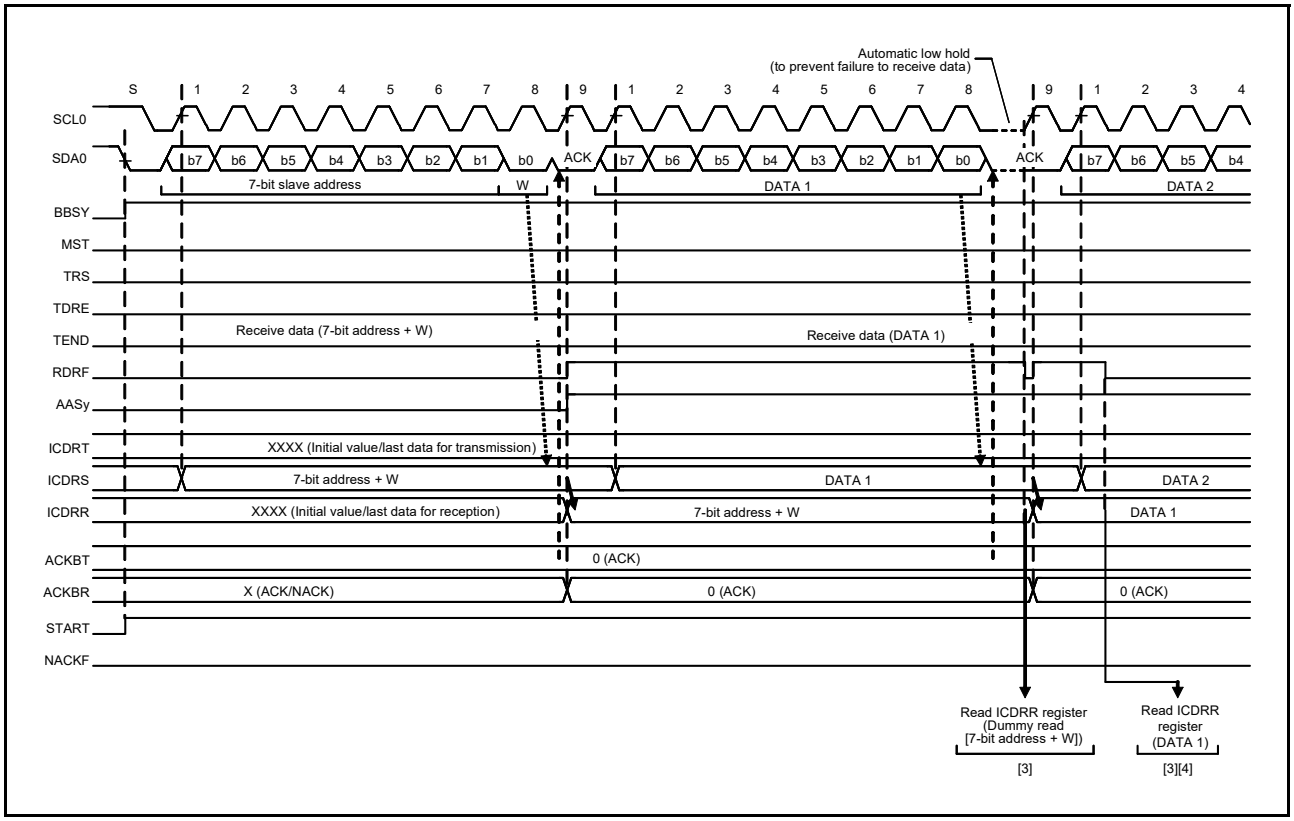


Figure 35.18 Slave Receive Operation Timing (1) (7-Bit Address Format, when RDRFS bit is 0)

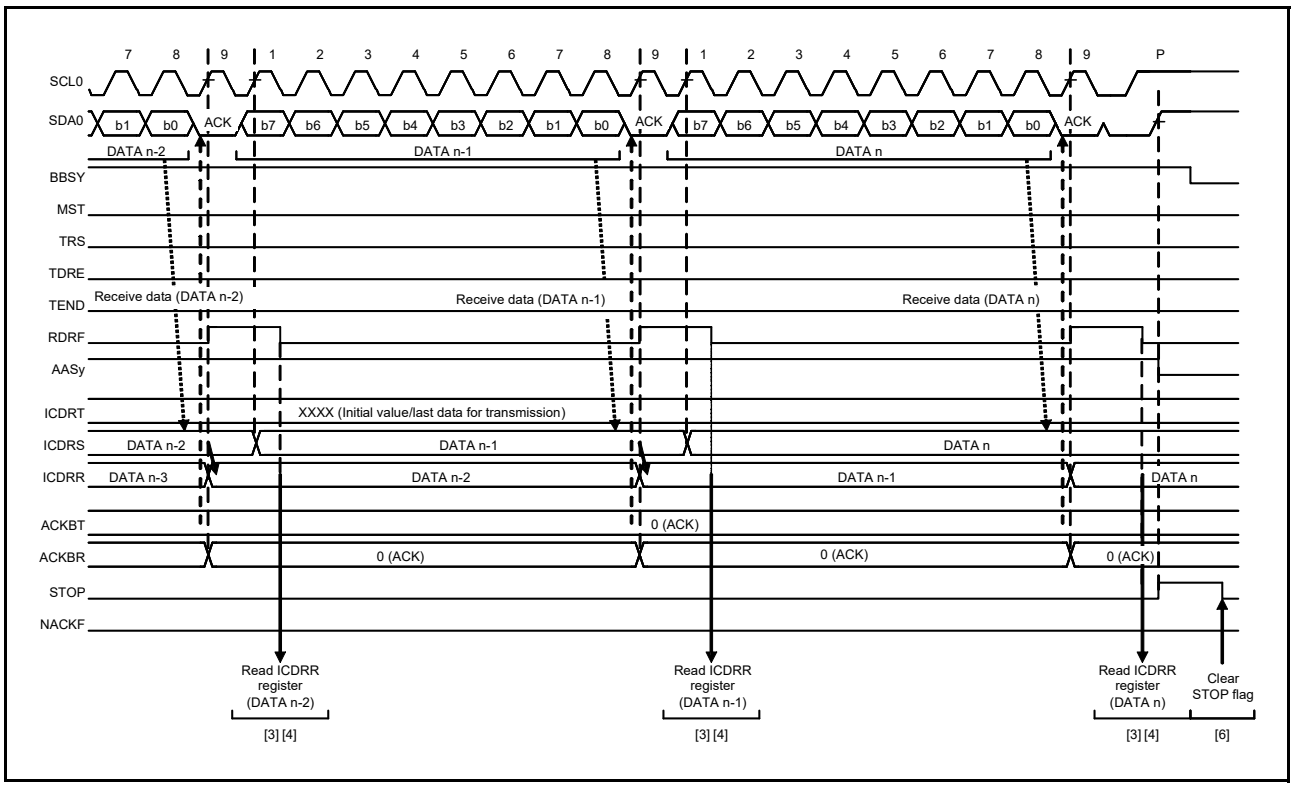


Figure 35.19 Slave Receive Operation Timing (2) (when RDRFS bit is 0)

Downloaded from Arrow.com.

35.4 SCL Synchronization Circuit

In generation of the SCL, the RIIC starts counting out the value for width at high level specified in the ICBRH register when it detects a rising edge on the SCL0 line and drives the SCL0 line low once counting of the width at high level is complete. When the RIIC detects the falling edge of the SCL0 line, it starts counting out the width at low period specified in the ICBRL register, and then stops driving the SCL0 line (releases the line) once counting of the width at low level is complete. The SCL is thus generated.

If multiple master devices are connected to the I2C-bus, a collision of SCL signals may arise due to contention with another master device. In such cases, the master devices have to synchronize their SCL signals. Since synchronization of the SCL signals must be handled bit by bit, the RIIC is equipped with a facility (the SCL synchronization circuit) for obtaining bit-by-bit synchronization of the SCL signals by monitoring the SCL0 line while in master mode.

When the RIIC has detected a rising edge on the SCL0 line and thus started counting out the width at high level specified in the ICBRH register, and the level on the SCL0 line falls because an SCL signal is being generated by another master device, the RIIC stops counting when it detects the falling edge, drives the level on the SCL0 line low, and starts counting out the width at low level specified in the ICBRL register. When the RIIC finishes counting out the width at low level, it stops driving the SCL0 line low (i.e. releases the line). At this time, if the width at low level of the SCL signal from the other master device is longer than the width at low level set in the RIIC, the width at low level of the SCL signal will be extended. Once the width at low level for the other master device has ended, the SCL signal rises because the SCL0 line has been released. When the RIIC finishes outputting the low period of the SCL, the SCL0 line is released and the SCL rises. That is, in cases of contention of SCL signals from more than one master, the width at high level of the SCL signal is synchronized with that of the clock having the narrower width, and the width at low level of the SCL signal is synchronized with that of the clock having the broader width. However, such synchronization of the SCL signal is only enabled when the ICFER.SCLE bit is set to 1.

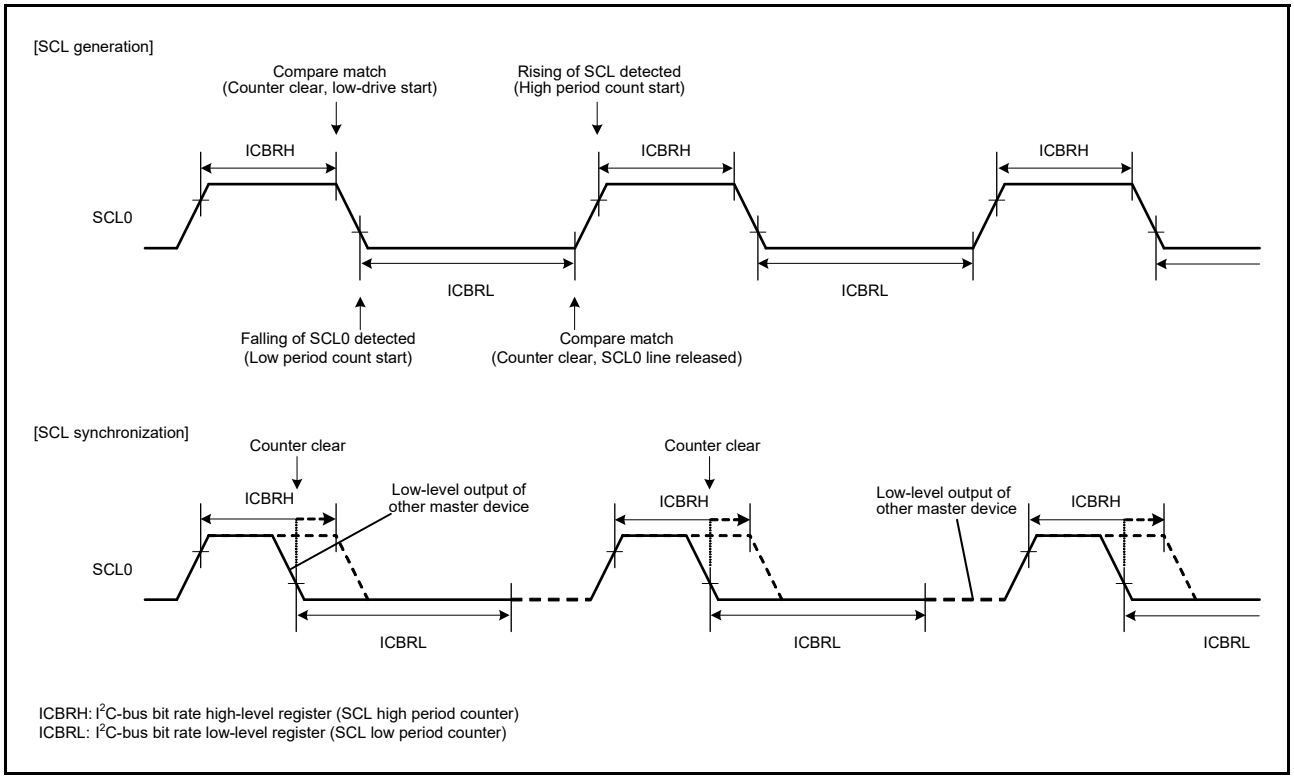


Figure 35.20 Generation and Synchronization of the SCL Signal from the RIIC

Downloaded from Arrow.com.

35.5 SDA Output Delay Function

The RIIC module incorporates a function for delaying output on the SDA line. The delay can be applied to all output (generation of the start, restart, and stop conditions, data, and the ACK and NACK signals) on the SDA line.

The SDA output delay function is used to delay the SDA output timing relative to falling edges of SCL to ensure that the SDA signal changes while the SCL is low and can be used to prevent erroneous operation of communications devices. This function is also used to satisfy the 300 ns (min.) data hold time prescribed by the SMBus specification.

The output delay function is enabled by setting the ICMR2.SDDL[2:0] bits to any value other than 000b, and disabled by setting the same bits to 000b.

When the SDA output delay function is enabled (the SDDL[2:0] bits are not “000b”), the SDA output delay counter counts the number of cycles set in the SDDL[2:0] bits of the count source selected by the ICMR2.DLCS bit (the internal reference clock (IICφ) or internal reference clock divided by 2 (IICφ/2)). On completion of counting of cycles of delay, the RIIC changes the bit being output as the SDA signal (generation of the start, restart, or stop condition, a new bit, or an ACK or NACK signal).

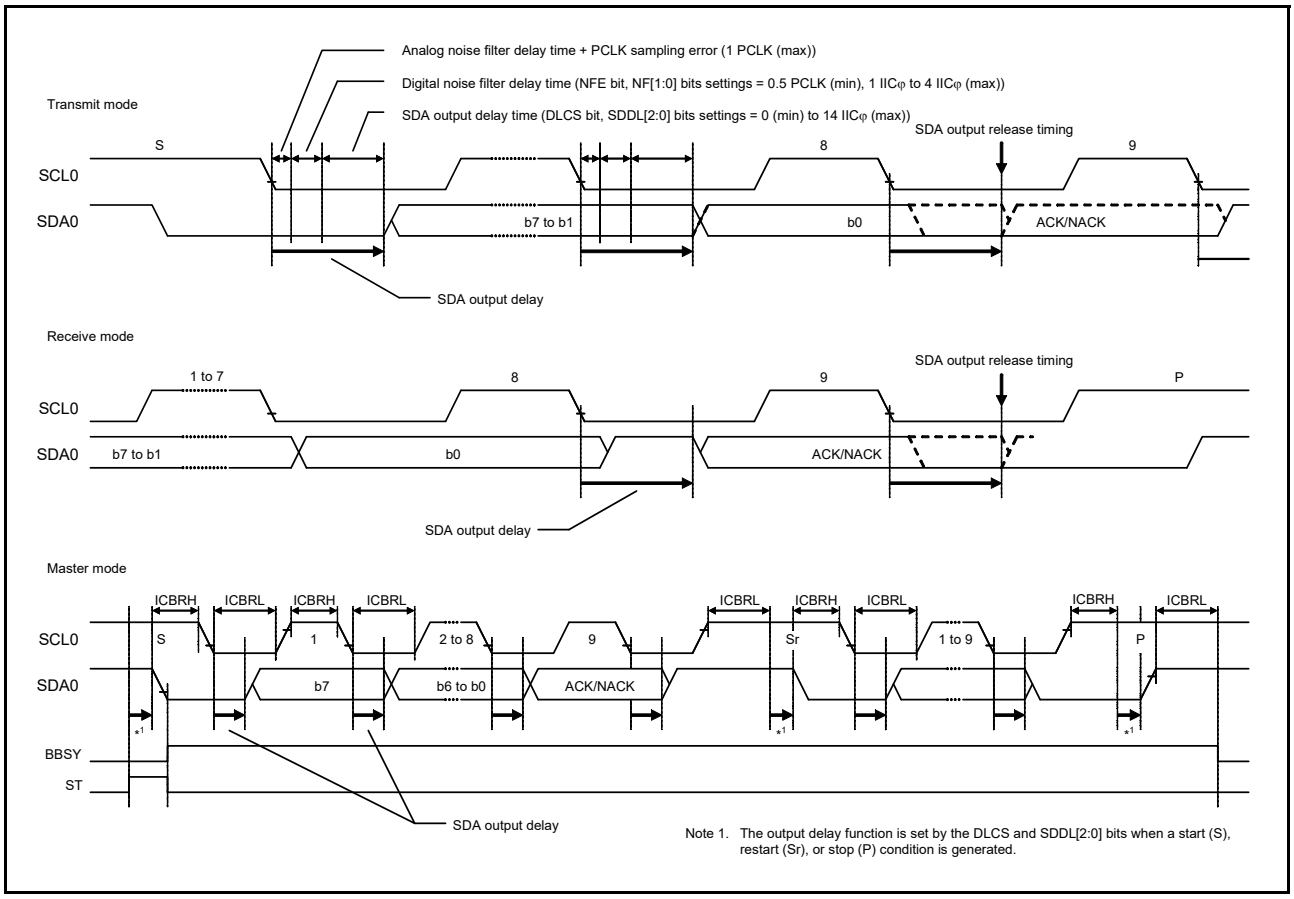


Figure 35.21 SDA Output Delay Function

Downloaded from Arrow.com.

35.6 Digital Noise Filters

The states of the SCL0 and SDA0 pins are conveyed to the internal circuitry through analog noise filters and digital noise filters. Figure 35.22 is a block diagram of the digital noise filter.

The on-chip digital noise filter of each RIIC consists of four flip-flop circuit stages connected in series and a match detection circuit.

The number of effective stages in the digital noise filter is selected by the ICMR3.NF[1:0] bits. The selected number of effective stages determines the noise-filtering capability as a period from one to four cycles of IICφ.

The input signal to the SCL0 pin (or SDA0 pin) is sampled on falling edges of the IICφ signal. When the input signal level matches the output level for the number of effective flip-flop circuit stages selected by the ICMR3.NF[1:0] bits, the signal level is conveyed to the subsequent stages. If the signal levels do not match, the previous value is retained.

If the ratio between the frequency of the internal operating clock (PCLK) and the transfer rate is relatively small (e.g. data transfer at 400 kbps with PCLK = 4 MHz), the characteristics of the digital noise filter may lead to the elimination of needed signals as noise. In such cases, it is possible to disable the digital noise filter (by setting the ICFER.NFE bit to 0) and use only the analog noise filter.

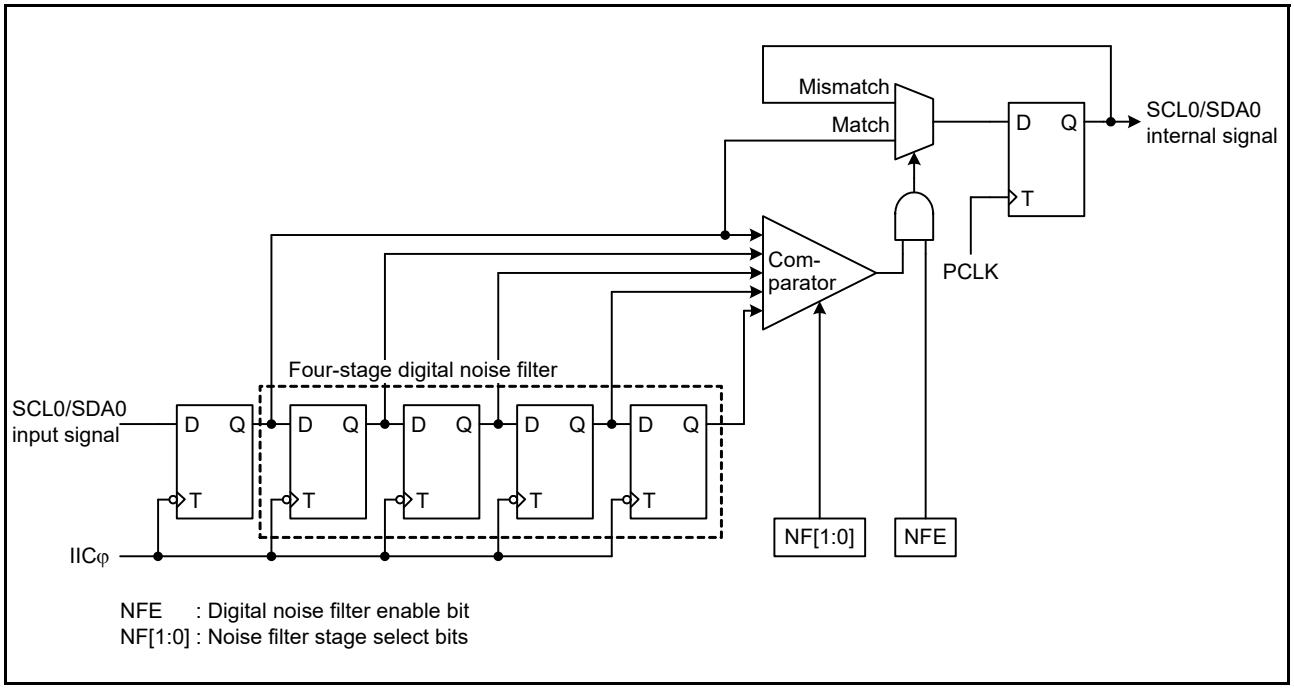


Figure 35.22 Block Diagram of the Digital Noise Filter

Downloaded from Arrow.com.

35.7 Address Match Detection

The RIIC can set three unique slave addresses in addition to the general call address and host address, and also can set 7-bit or 10-bit slave addresses.

35.7.1 Slave-Address Match Detection

The RIIC can set three unique slave addresses, and has a slave address detection function for each unique slave address. When the ICSER.SARyE bit (y = 0 to 2) is set to 1, the slave addresses set in registers SARUy and SARLy (y = 0 to 2) can be detected.

When the RIIC detects a match with its set slave address, the corresponding ICSR1.AASy flag (y = 0 to 2) is set to 1 on the rising edge of the ninth SCL, and the ICSR2.RDRF flag or the ICSR2.TDRE flag is set to 1 according to the level of the R/W# bit. This causes a receive data full interrupt (RXI) or transmit data empty interrupt (TXI) to be generated. The AASy flag is used to identify which slave address has been specified.

Figure 35.23 to Figure 35.25 show the AASy flag set timing in three cases.

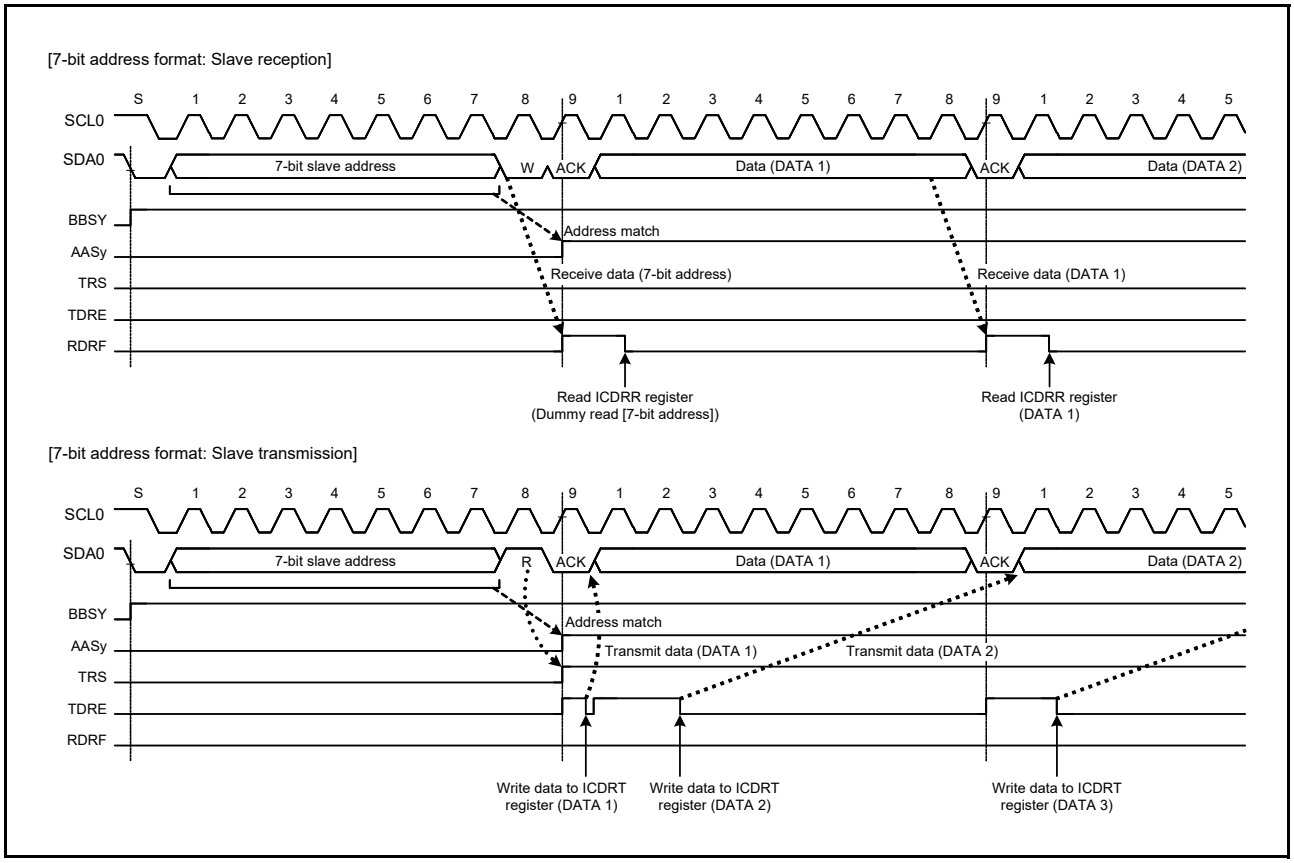


Figure 35.23 AASy Flag Set Timing with 7-Bit Address Format Selected

Downloaded from Arrow.com.

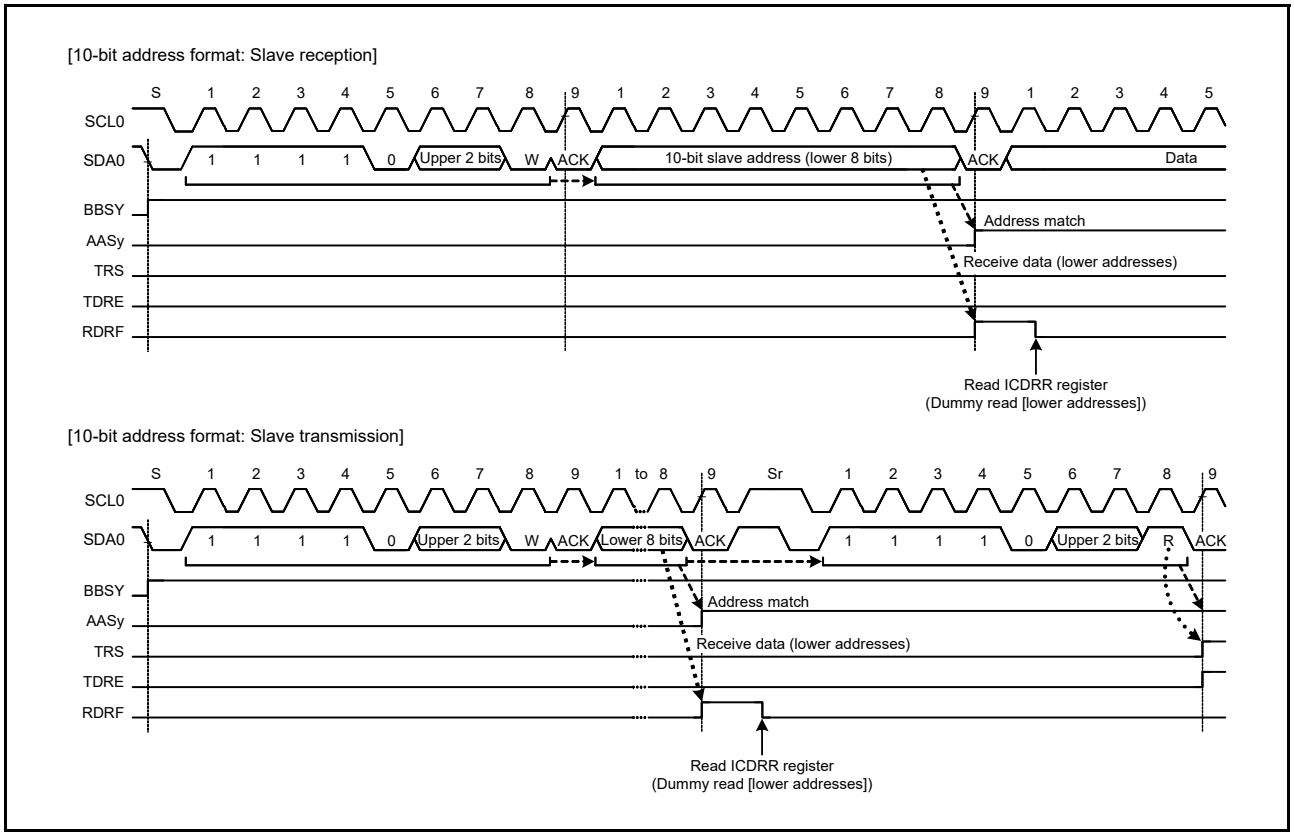


Figure 35.24 AASy Flag Set Timing with 10-Bit Address Format Selected

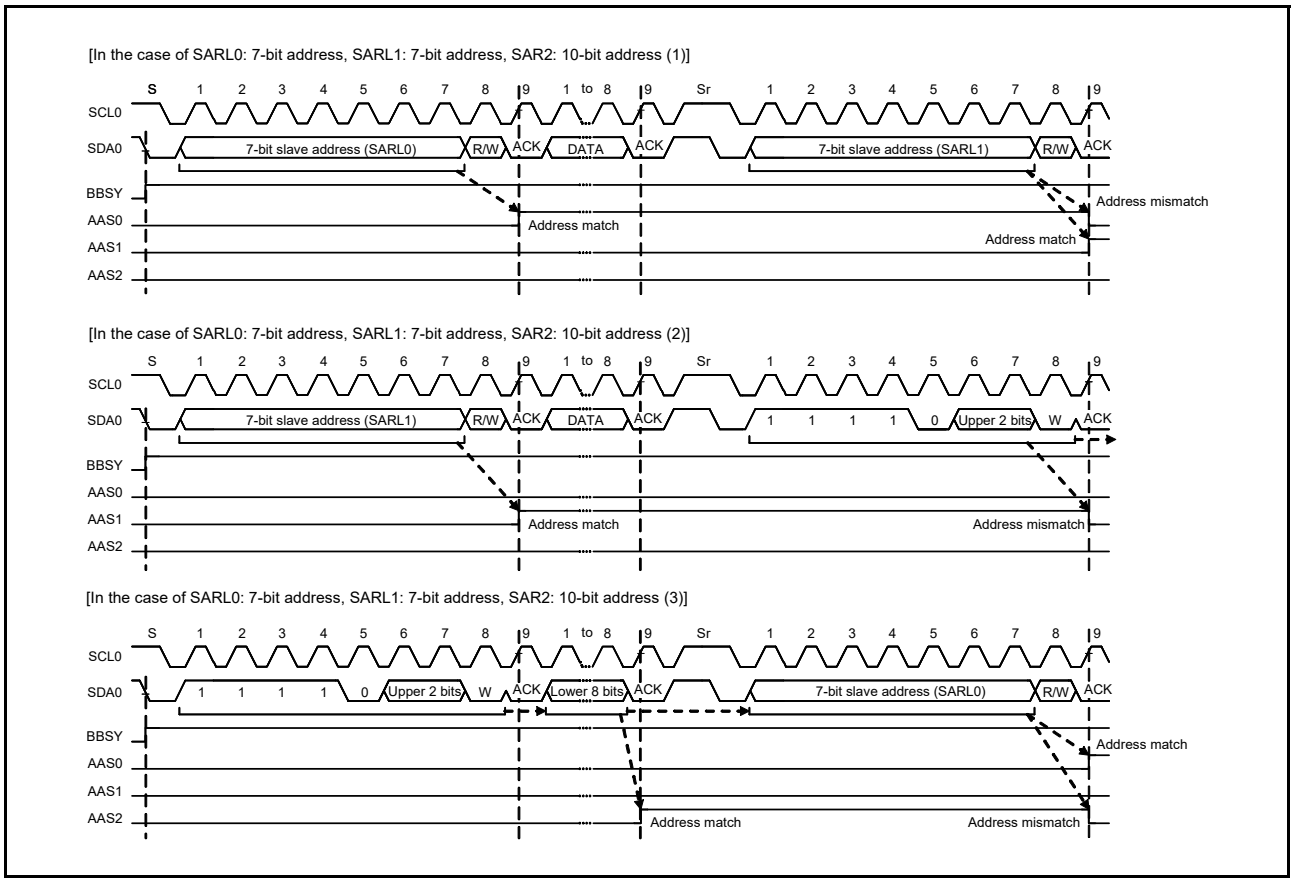


Figure 35.25 AASy Flag Set/Clear Timing with 7-Bit/10-Bit Address Formats Mixed

Downloaded from Arrow.com.

35.7.2 Detection of the General Call Address

The RIIC also has a facility for detecting the general call address (0000 000b + 0 (write)). This is enabled by setting the ICSER.GCAE bit to 1.

If the address following a start or restart condition is 0000 000b + 1 (read) (start byte), the RIIC recognizes this as the address of a slave device with an “all-zero” address but not as the general call address.

When the RIIC detects the general call address, both the ICSR1.GCA flag and the ICSR2.RDRF flag are set to 1 on the rising edge of the ninth SCL. This leads to the generation of a receive data full interrupt (RXI). The value of the GCA flag can be confirmed to recognize that the general call address has been transmitted.

Operation after detection of the general call address is the same as normal slave receive operation.

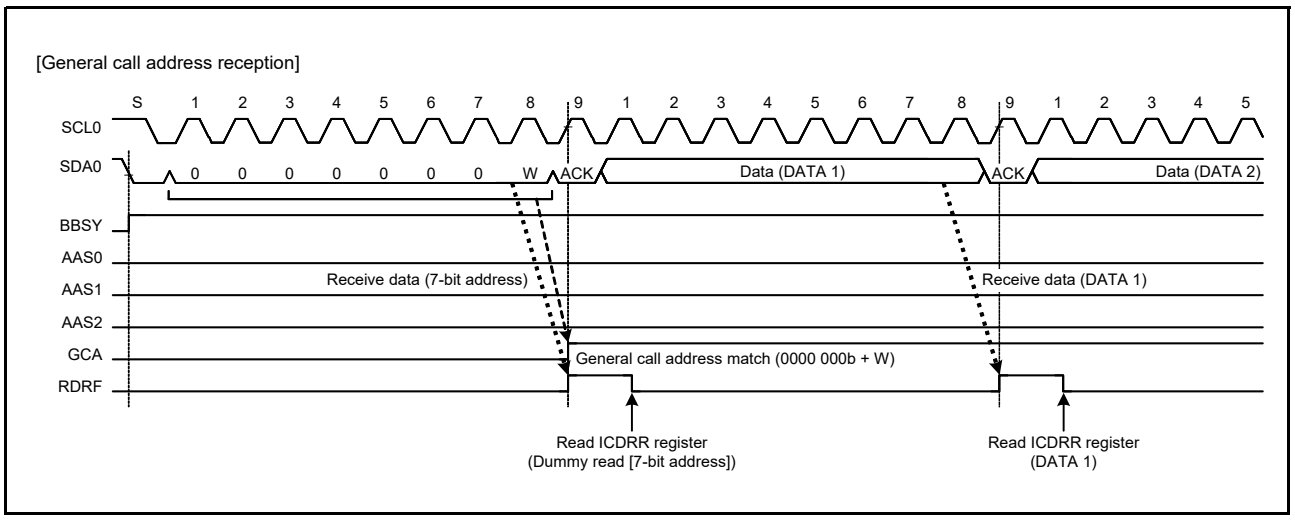


Figure 35.26 Timing of GCA Flag Setting during Reception of General Call Address

Downloaded from Arrow.com.

35.7.3 Device-ID Address Detection

The RIIC module has a function to detect device-ID addresses complying with the I2C-bus specification. When the RIIC receives 1111 100b as the first seven bits of the first byte following a start condition or a restart condition while the ICSER.DIDE bit is 1, the RIIC recognizes the address as a device-ID address, sets the ICSR1.DID flag to 1 on the rising edge of the ninth SCL when the following R/W# bit is 0, and then compares the second and following bytes with its own slave address. If the received address matches the value in the slave address register, the RIIC sets the corresponding ICSR1.AASy flag (y = 0 to 2) to 1.

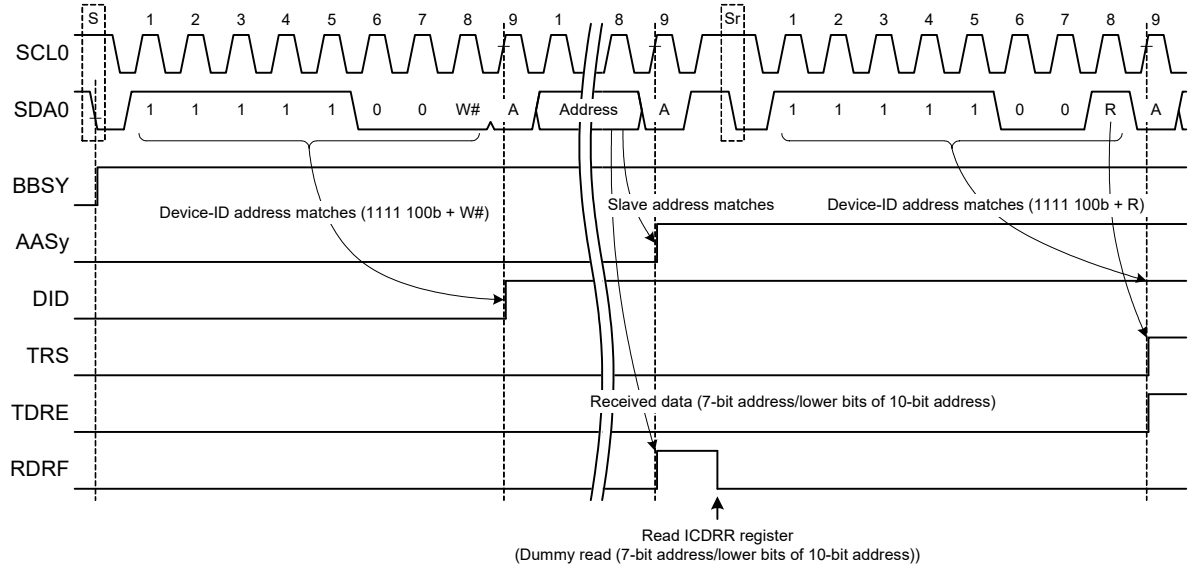
After that, when the first byte received after a start or restart condition is generated matches the device ID address (1111 100b) again and the following R/W# bit is 1, the RIIC does not compare the second and subsequent bytes and sets the ICSR2.TDRE flag to 1.

In the device-ID address detection function, the RIIC sets the DID flag to 0 if a match with the RIIC's own slave address is not obtained or a match with the device ID address is not obtained after a match with the RIIC's own slave address and the detection of a restart condition. If the first byte after detection of a start or restart condition matches the device ID address (1111 100b) and the R/W# bit is 0, the RIIC sets the DID flag to 1 and compares the second and subsequent bytes with the RIIC's slave address. If the R/W# bit is 1, the DID flag holds the previous value and the RIIC does not compare the second and subsequent bytes. Therefore, the reception of a device-ID address can be checked by reading the DID flag after confirming that TDRE flag is 1.

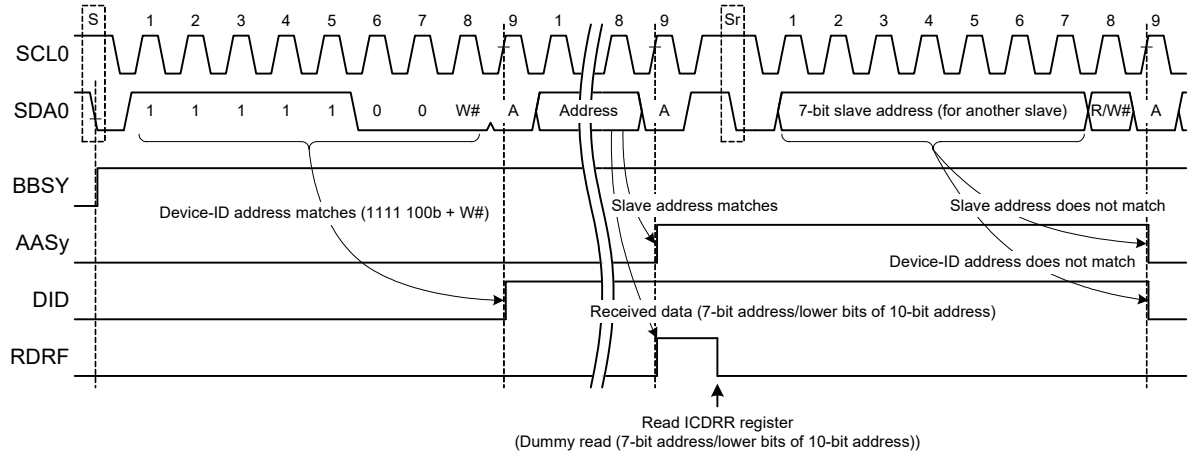
Furthermore, prepare the device-ID fields (3 bytes: 12 bits indicating the manufacturer + 9 bits identifying the part + 3 bits indicating the revision) that must be sent to the host after reception of a continuous device-ID field as normal data for transmission. For details of the information that must be included in device-ID fields, contact NXP Semiconductors.

Downloaded from Arrow.com.

[Device-ID reception]



[When the address following a restart condition is not a device-ID address]



[When the address before the device-ID address + R had not matched the slave address]

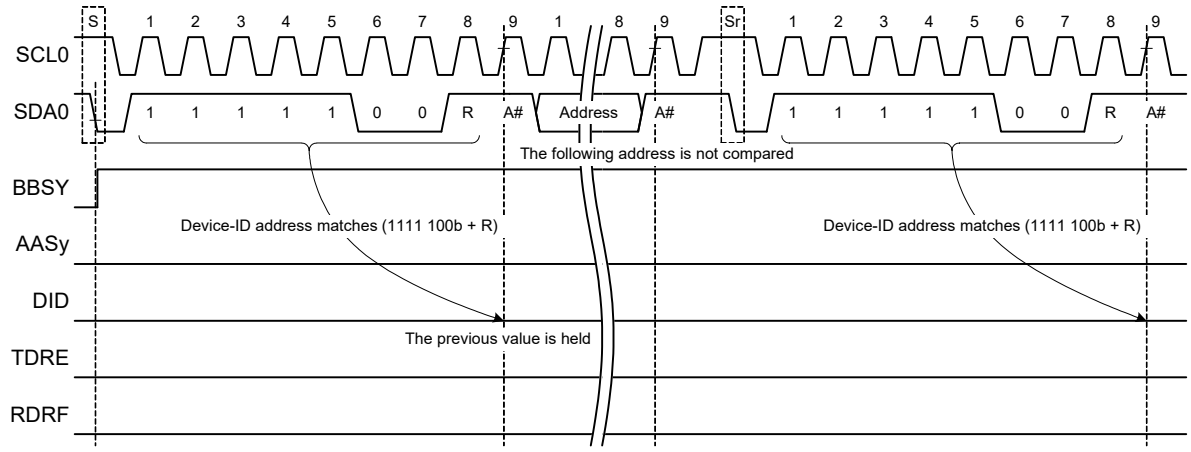


Figure 35.27 Set/Clear Timing of the AASy and DID Flags during Reception of Device-ID Address

Downloaded from Arrow.com.

35.7.4 Host Address Detection

The RIIC has a function to detect the host address while the SMBus is operating. When the ICSER.HOAE bit is set to 1 while the ICMR3.SMBS bit is 1, the RIIC can detect the host address (0001 000b) in slave receive mode (bits MST and TRS in the ICCR2 register are 00b).

When the RIIC detects the host address, the ICSR1.HOA flag is set to 1 at the rising edge of the ninth SCL, and at the same time, the ICSR2.RDRF flag is set to 1 when the R/W# bit is 0 (Wr bit). This causes a receive data full interrupt (RXI) to be generated. The HOA flag is used to recognize that the host address was sent from the smart battery or other devices.

If the bit following the host address (0001 000b) is an Rd bit (R/W# bit is 1), the RIIC can also detect the host address. After the host address is detected, the RIIC operates in the same manner as normal slave operation.

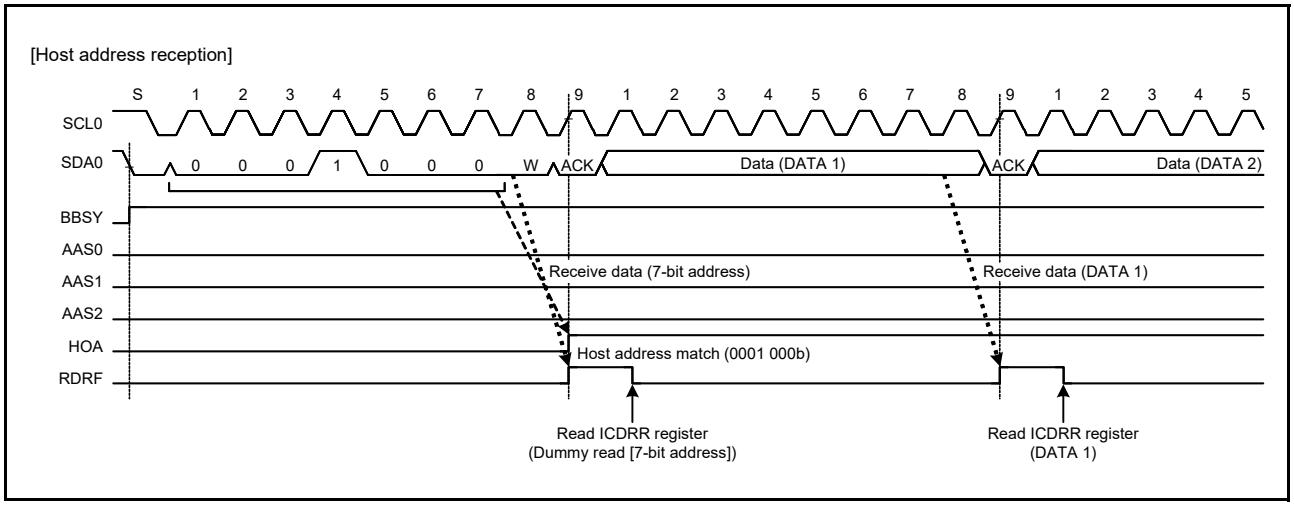


Figure 35.28 HOA Flag Set Timing during Reception of Host Address

Downloaded from Arrow.com.

35.8 Automatic Low-Hold Function for SCL

35.8.1 Function to Prevent Wrong Transmission of Transmit Data

If the shift register (ICDRS) is empty when data have not been written to the I2C-bus transmit data register (ICDRT) with the RIIC in transmission mode (ICCR2.TRS bit is 1), the SCL0 line is automatically held low over the intervals shown below. This low period is extended until data for transmission have been written, which prevents the unintended transmission of erroneous data.

Master transmit mode

- Low period after a start condition or restart condition is generated
- Low period between the ninth clock pulse of one transfer and the first clock pulse of the next

Slave transmit mode

- Low period between the ninth clock pulse of one transfer and the first clock pulse of the next

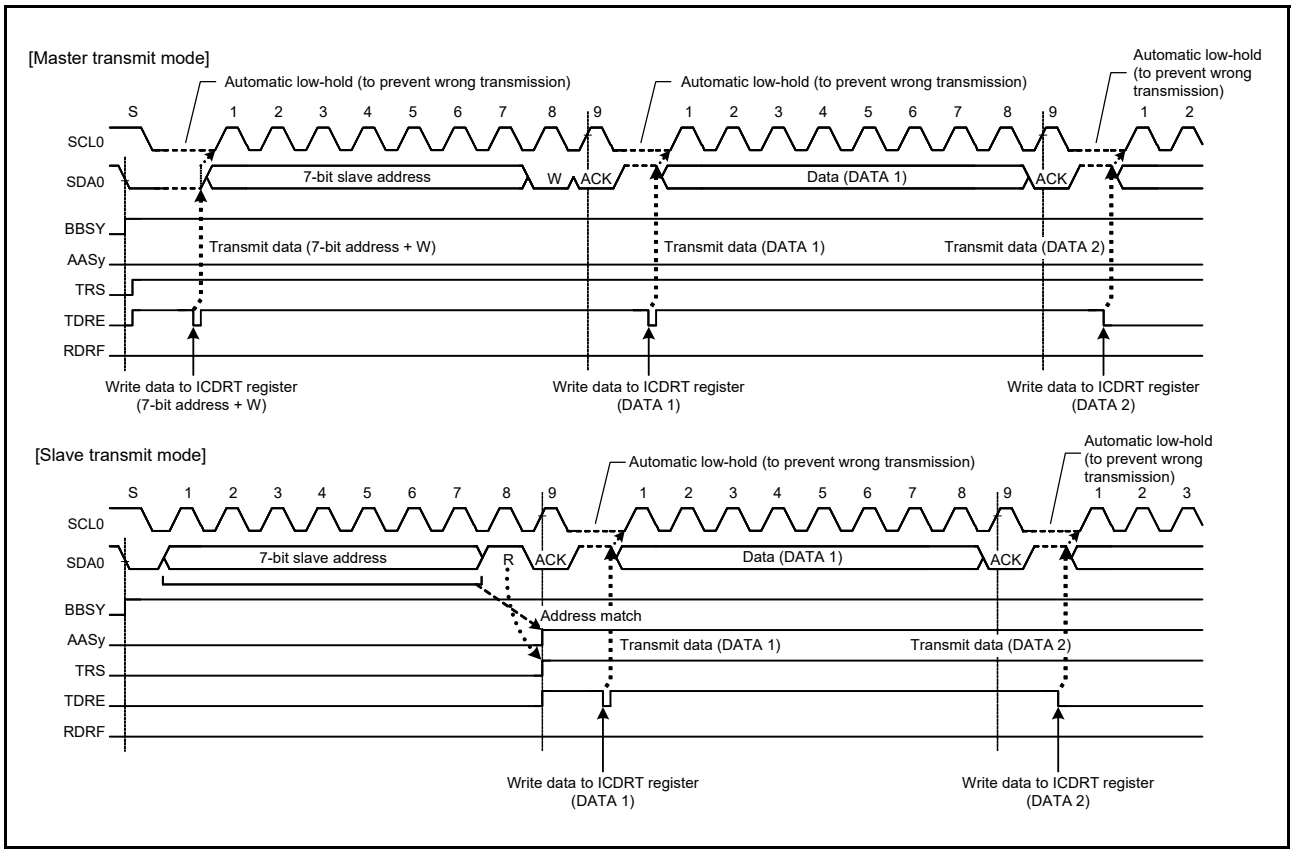


Figure 35.29 Automatic Low-Hold Operation in Transmit Mode

Downloaded from Arrow.com.

35.8.2 NACK Reception Transfer Suspension Function

The RIIC has a function to suspend transfer operation when NACK is received in transmit mode (ICCR2.TRS bit is 1). This function is enabled when the ICFER.NACKE bit is set to 1 (transfer suspension enabled). If the next transmit data has already been written (ICSR2.TDRE flag is 0) when NACK is received, next data transmission at the falling edge of the ninth SCL is automatically suspended. This prevents the SDA0 line output level from being held low when the MSB of the next transmit data is 0.

If the data transmission is suspended (ICSR2.NACKF flag is 1) by this function, the following data transmission and data reception are not started. To resume data transfer, set the NACKF flag to 0. In master transmit mode, restart data transfer by setting the NACKF flag to 0 after generating a restart condition, or restart data transfer from a start condition after generating a stop condition then setting the NACKF flag to 0.

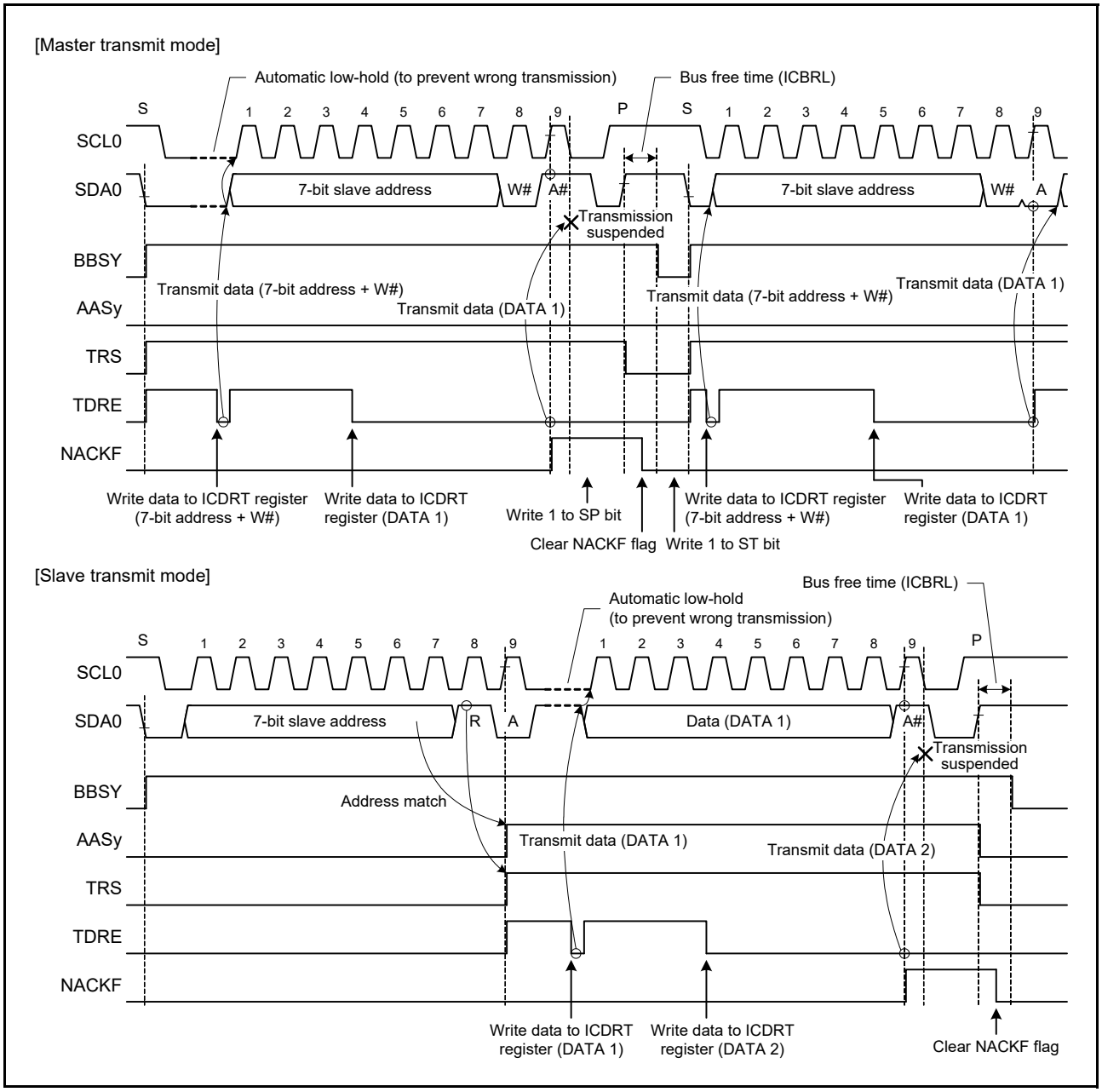


Figure 35.30 Suspension of Data Transmission When NACK is Received (NACKE = 1)

Downloaded from Arrow.com.

35.8.3 Function to Prevent Failure to Receive Data

If response processing is delayed when receive data (ICDRR) read is delayed for a period of one transfer byte or more with receive data full (ICSR2.RDRF flag is 1) in receive mode (ICCR2.TRS bit is 0), the RIIC holds the SCL0 line low automatically immediately before the next data is received to prevent failure to receive data.
This function to prevent failure to receive data using the automatic low-hold function is also enabled even if the read processing of the final receive data is delayed and, in the meantime, the RIIC's own slave address is designated after a stop condition is generated. This function does not disturb other communication because the RIIC does not hold the SCL0 line low when a mismatch with its own slave address occurs after a stop condition is generated.
Sections in which the SCL0 line is held low can be selected with a combination of the WAIT and RDRFS bits in the ICMR3 register.

(1) 1-Byte Receive Operation and Automatic Low-Hold Function Using the WAIT Bit

When the ICMR3.WAIT bit is set to 1, the RIIC performs 1-byte receive operation using the WAIT bit function. Furthermore, when the ICMR3.RDRFS bit is 0, the RIIC automatically sends the ICMR3.ACKBT bit value for the acknowledgment bit in the period from the falling edge of the eighth SCL to the falling edge of the ninth SCL, and automatically holds the SCL0 line low at the falling edge of the ninth SCL using the WAIT bit function. This low-hold is released by reading data from the ICDRR register, which enables bytewise receive operation.
The WAIT bit function is enabled for receive bytes after a match with the RIIC's own slave address (including the general call address and host address) is obtained in master receive mode or slave receive mode.

(2) 1-Byte Receive Operation (ACK/NACK Transmission Control) and Automatic Low-Hold Function Using the RDRFS Bit

When the ICMR3.RDRFS bit is set to 1, the RIIC performs 1-byte receive operation using the RDRFS bit function. When the RDRFS bit is set to 1, the ICSR2.RDRF flag (receive data full) is set to 1 at the rising edge of the eighth SCL, and the SCL0 line is automatically held low at the falling edge of the eighth SCL. This low-hold is released by writing a value to the ICMR3.ACKBT bit, but cannot be released by reading data from the ICDRR register, which enables receive operation by the ACK/NACK transmission control according to the data received in byte units.
The RDRFS bit function is enabled for receive bytes after a match with the RIIC's own slave address (including the general call address and host address) is obtained in master receive mode or slave receive mode.

Downloaded from Arrow.com.

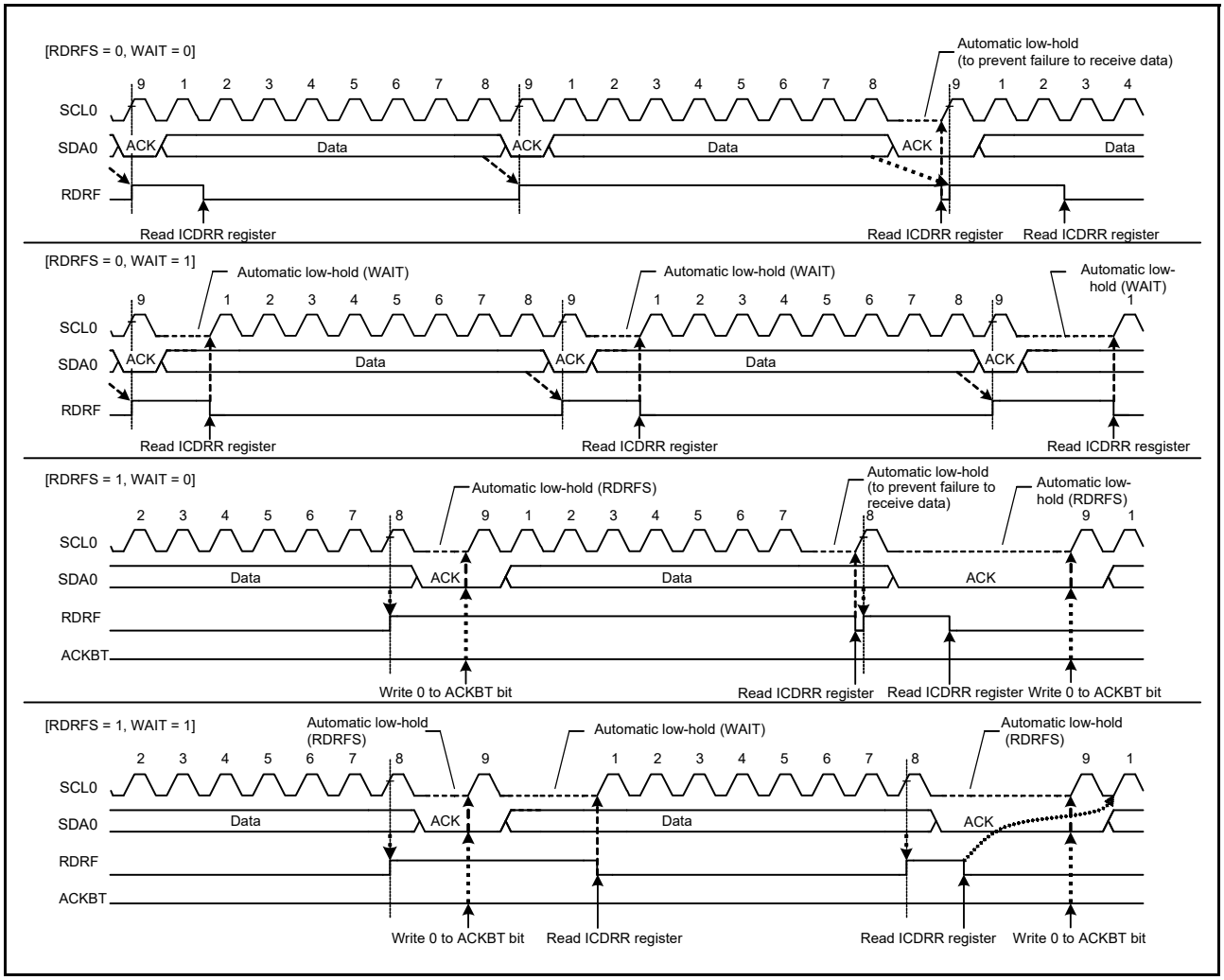


Figure 35.31 Automatic Low-Hold Operation in Receive Mode (Using RDRFS and WAIT Bits)

Downloaded from Arrow.com.

35.9 Arbitration-Lost Detection Functions

In addition to the normal arbitration-lost detection function defined by the I2C-bus specification, the RIIC has functions to prevent double-generation of a start condition, to detect arbitration-lost during transmission of NACK, and to detect arbitration-lost in slave transmit mode.

35.9.1 Master Arbitration-Lost Detection (MALE Bit)

The RIIC drives the SDA0 line low to generate a start condition. However, if the SDA0 line has already been driven low by another master device generating a start condition, the RIIC causes arbitration to be lost, so priority is given to transfer by the other master device. Similarly, if the ICCR2.ST bit is set to 1 while the ICCR2.BBSY flag is 1 (bus busy state), arbitration is lost, so priority is given to transfer by the other master device. No start condition is generated in this case.
When a start condition is generated successfully, if the data for transmission including the address bits (i.e. the internal SDA output level) and the level on the SDA0 line do not match (the high output as the internal SDA output; i.e. the SDA0 pin is in the high-impedance state) and the low is detected on the SDA0 line, the RIIC loses the arbitration.
After a master arbitration-lost is generated, the RIIC immediately enters slave receive mode. If a slave address (including the general call address) matches its own address at this time, the RIIC continues in slave operation.
A master arbitration-lost is detected when the following conditions are met while the ICFER.MALE bit is 1 (master arbitration-lost detection enabled).

Conditions for detecting master arbitration-lost

- Non-matching of the internal level for output on SDA and the level on the SDA0 line after a start condition was generated by setting the ICCR2.ST bit to 1 while the ICCR2.BBSY flag was set to 0 (erroneous generation of a start condition)
- Setting of the ICCR2.ST bit to 1 while the BBSY flag is set to 1 (start condition double-generation error)
- When the transmit data excluding acknowledgment bit (internal SDA output level) does not match the level on the SDA0 line in master transmit mode (bits MST and TRS in the ICCR2 register = 11b)

Downloaded from Arrow.com.

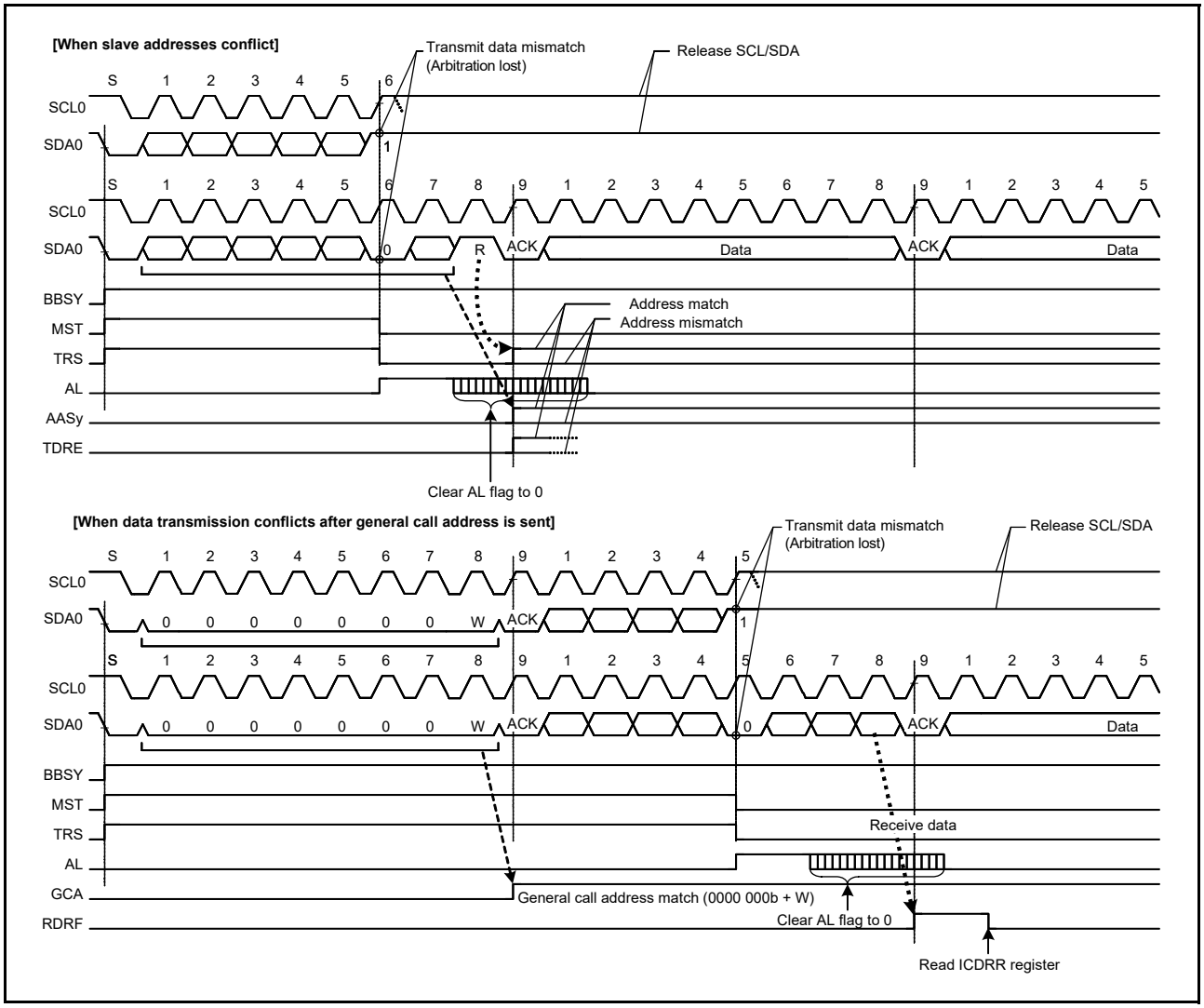


Figure 35.32 Examples of Master Arbitration-Lost Detection (MALE = 1)

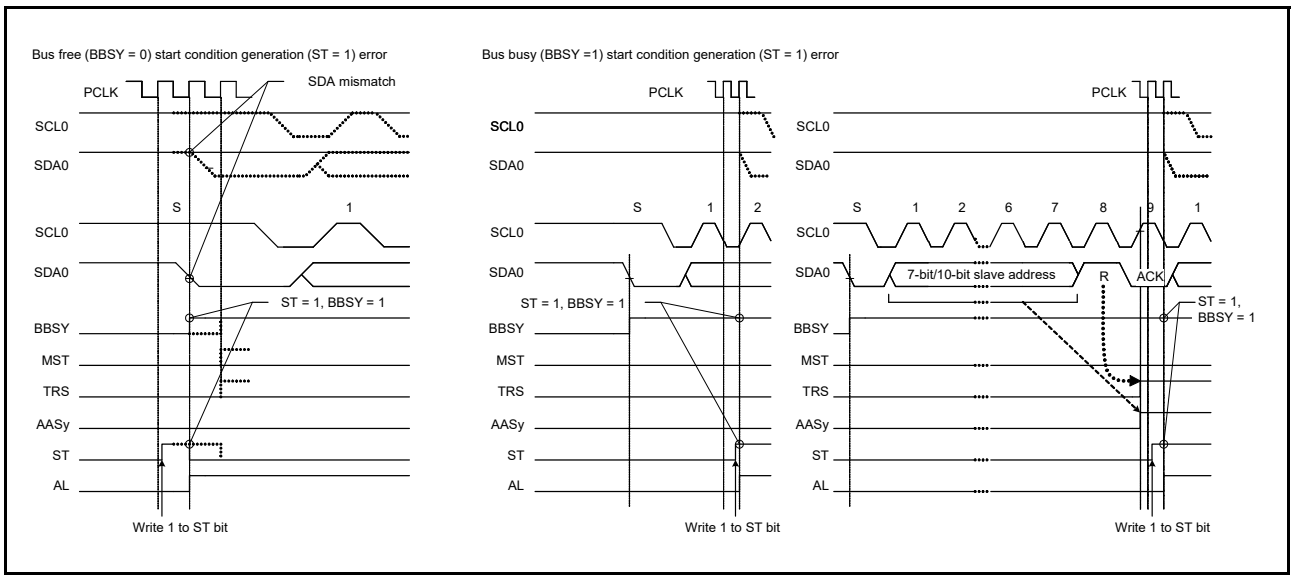


Figure 35.33 Arbitration-Lost When a Start Condition is Generated (MALE = 1)

Downloaded from Arrow.com.

35.9.2 NACK Transmission Arbitration-Lost Detection Function (NALE Bit)

The RIIC has a function to cause arbitration to be lost if the internal SDA output level does not match the level on the SDA0 line (the high output as the internal SDA output; i.e. the SDA0 pin is in the high-impedance state) and the low is detected on the SDA0 line during transmission of NACK in receive mode. Arbitration is lost due to a conflict of NACK transmission and ACK transmission when two or more master devices receive data from the same slave device simultaneously in a multi-master system. Such conflict occurs when multiple master devices send/receive the same information through a single slave device. Figure 35.34 shows an example of NACK transmission arbitration-lost detection.

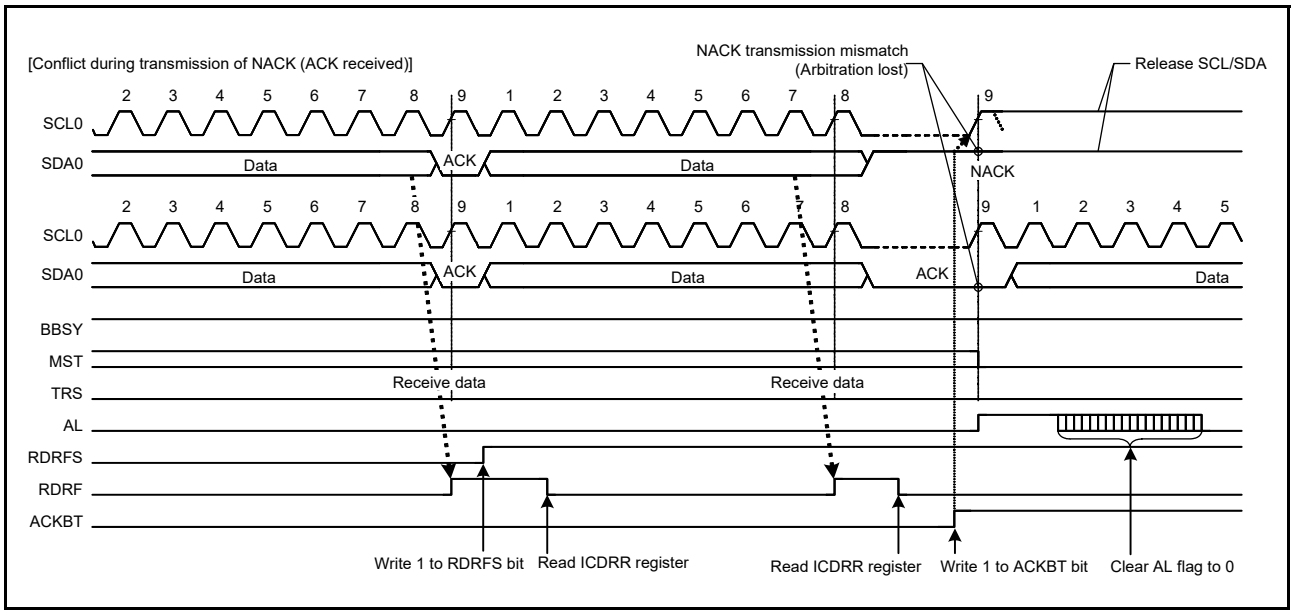


Figure 35.34 Example of NACK transmission arbitration-lost Detection (NALE = 1)

The following explains arbitration-lost detection using an example where two master devices (master A and master B) and a single slave device are connected through the bus. In this example, master A receives two bytes of data from the slave device, and master B receives four bytes of data from the slave device.

If master A and master B access the slave device simultaneously, because the slave address is identical, arbitration is not lost in both master A and master B during access to the slave device. Therefore, both master A and master B recognize that they have obtained the bus mastership and operate as such. Here, master A sends NACK when it has received second byte of data from the slave device. Meanwhile, master B sends ACK because it has not received necessary four bytes of data. At this time, the NACK transmission from master A and the ACK transmission from master B conflict. In general, if a conflict like this occurs, master A cannot detect ACK transmitted by master B and generates a stop condition. Therefore, the generation of the stop condition conflicts with the SCL output of master B, which disturbs communication.

When the RIIC receives ACK during transmission of NACK, it detects a defeat in conflict with other master devices and causes arbitration to be lost.

If a NACK transmission arbitration-lost occurs, the RIIC immediately cancels the slave address matched state and enters slave receive mode. This prevents a stop condition from being generated, preventing a communication failure on the bus. Also, in the ARP command processing of SMBus, it is possible to omit surplus processing (FFh transmission processing) after NACK transmission when the UDID (Unique Device Identifier) of “Assign Address” does not match and after the NACK transmission of the “Get UDID (General)” after the “Assign Address” is confirmed.

The RIIC detects NACK transmission arbitration-lost when the following condition is met with the ICFER.NALE bit set to 1 (NACK transmission arbitration-lost is enabled).

Downloaded from Arrow.com.

Condition for detecting NACK transmission arbitration-lost

- When the internal SDA output level does not match the SDA0 line (ACK is received) during transmission of NACK (ICMR3.ACKBT bit = 1)

35.9.3 Slave Arbitration-Lost Detection (SALE Bit)

The RIIC has a function to cause arbitration to be lost if the data for transmission (i.e. the internal SDA output level) and the level on the SDA0 line do not match (the high output as the internal SDA output; i.e. the SDA0 pin is in the high-impedance state and the low is detected on the SDA0 line in slave transmit mode). The slave arbitration-lost detection function is mainly used when transmitting a UDID (Unique Device Identifier) of SMBus.
When the slave arbitration-lost occurs, the RIIC is immediately released from the slave address matched state and enters slave receive mode. This function can detect a data conflict during UDID transmission of SMBus and omit surplus processing (FFh transmission processing) after the data conflict.
The RIIC detects slave arbitration-lost when the following condition is met with the ICFER.SALE bit set to 1 (slave arbitration-lost detection enabled).

Condition for detecting slave arbitration-lost

- When transmit data excluding acknowledgment bit (internal SDA output level) does not match the SDA0 line in slave transmit mode (bits MST and TRS in the ICCR2 register are 01b)

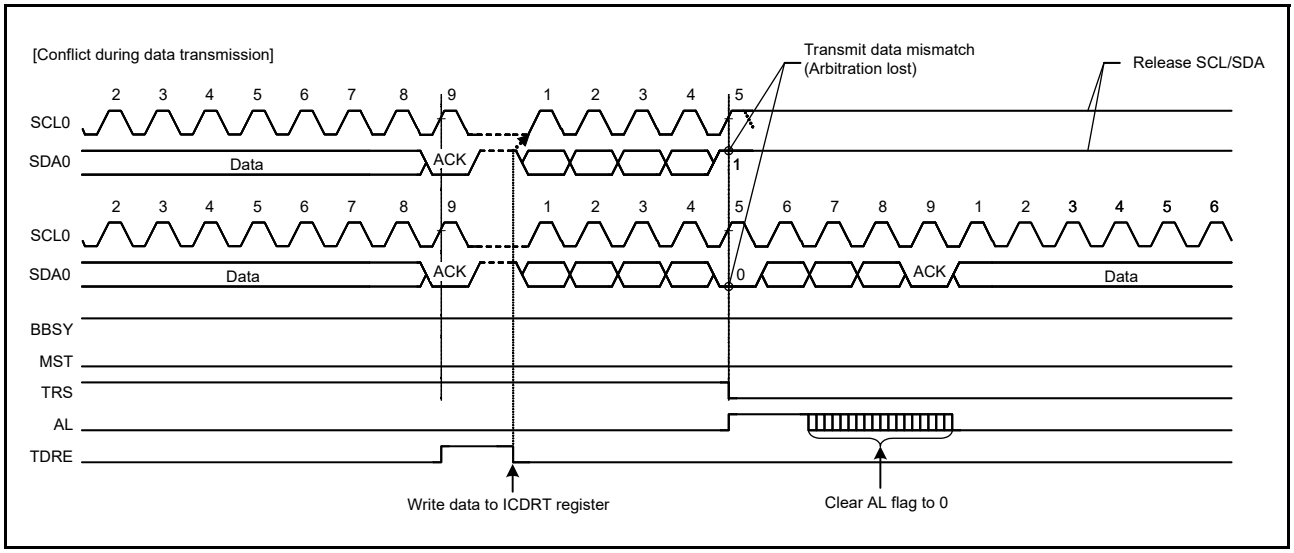


Figure 35.35 Example of Slave Arbitration-Lost Detection (SALE = 1)

Downloaded from Arrow.com.

35.10 Start Condition/Restart Condition/Stop Condition Generating Function

35.10.1 Generating a Start Condition

The RIIC generates a start condition when the ICCR2.ST bit is set to 1.
When the ST bit is set to 1, a start condition generation request is made and the RIIC generates a start condition when the ICCR2.BBSY flag is 0 (bus free state). When a start condition is generated normally, the RIIC automatically shifts to the master transmit mode.
A start condition is generated in the following sequence.

Start condition generation

(1) Drive the SDA0 line low (high to low).
(2) Secure the time set in the ICBRH register and the start condition hold time.
(3) Drive the SCL0 line low (high to low).
(4) Detect the low level on the SCL0 line and secure the low period of the signal on the SCL0 line set in the ICBRL register.

35.10.2 Generating a Restart Condition

The RIIC generates a restart condition when the ICCR2.RS bit is set to 1.
When the RS bit is set to 1, a restart condition generation request is made and the RIIC generates a restart condition when the ICCR2.BBSY flag is 1 (bus busy state) and the ICCR2.MST bit is 1 (master mode).
A restart condition is generated in the following sequence.

Restart condition generation

(1) Release the SDA0 line.
(2) Secure the low period of the signal on the SCL0 line set in the ICBRL register.
(3) Release the SCL0 line (low to high).
(4) Detect the high level on the SCL0 line and secure the time set in the ICBRL register and the restart condition setup time.
(5) Drive the SDA0 line low (high to low).
(6) Secure the time set in the ICBRH register and the restart condition hold time.
(7) Drive the SCL0 line low (high to low).
(8) Detect the low level on the SCL0 line and secure the low period of the signal on the SCL0 line set in the ICBRL register.

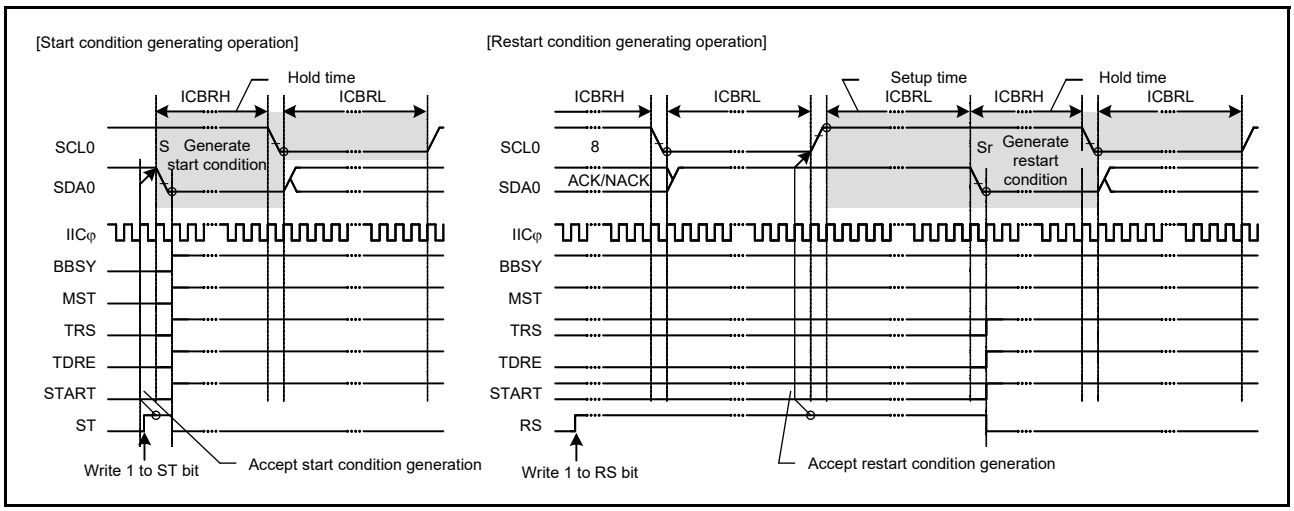


Figure 35.36 Start Condition/Restart Condition Generation Timing (ST and RS Bits)

Downloaded from Arrow.com.

35.10.3 Generating a Stop Condition

The RIIC generates a stop condition when the ICCR2.SP bit is set to 1.
When the SP bit is set to 1, a stop condition generation request is made and the RIIC generates a stop condition when the ICCR2.BBSY flag is 1 (bus busy state) and the ICCR2.MST bit is 1 (master mode).
A stop condition is generated in the following sequence.

Stop condition generation

(1) Drive the SDA0 line low (high to low).
(2) Secure the low period of the signal on the SCL0 line set in the ICBRL register.
(3) Release the SCL0 line (low to high).
(4) Detect the high level on the SCL0 line and secure the time set in the ICBRH register and the stop condition setup time.
(5) Release the SDA0 line (low to high).
(6) Secure the time set in the ICBRL register and the bus free time.
(7) Set the BBSY flag to 0 (to release the bus mastership).

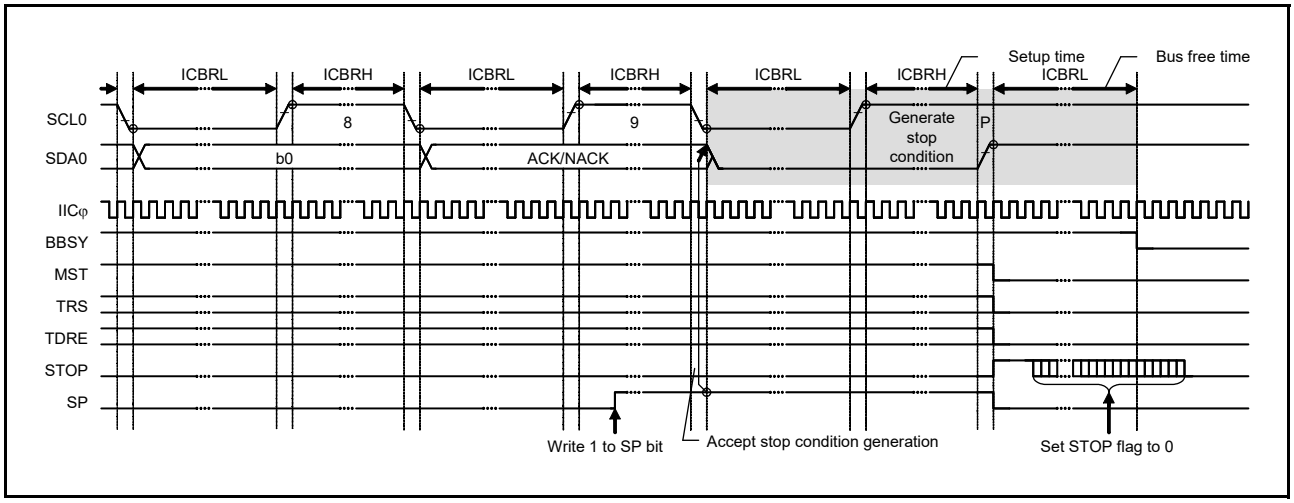


Figure 35.37 Stop Condition Generation Timing (SP Bit)

Downloaded from Arrow.com.

35.11 Bus Hanging

If the clock signals from the master and slave devices go out of synchronization due to noise or other factors, the I2C-bus might hang with a fixed level on the SCL0 line and/or SDA0 line.
As measures against the bus hanging, the RIIC has a timeout function to detect hanging by monitoring the SCL0 line, a function for the output of an additional SCL to release the bus from a hung state due to clock signals being out of synchronization, the RIIC reset function, and internal reset function.
By checking bits SCLO, SDAO, SCLI, and SDAI in the ICCR1 register, it is possible to see whether the RIIC or its partner in communications is placing the low level on the SCL0 or SDA0 lines.

35.11.1 Timeout Function

The RIIC includes a timeout function for detecting when the SCL0 line has been stuck longer than the predetermined time. The RIIC can detect an abnormal bus state by monitoring that the SCL0 line is stuck low or high for a predetermined time.
The timeout function monitors the SCL0 line state and counts the low period or high period using the internal counter. The timeout function resets the internal counter each time the SCL0 line changes (rising or falling), but continues to count unless the SCL0 line changes. If the internal counter overflows due to no SCL0 line change, the RIIC can detect the timeout and report the bus hung state.
This timeout function is enabled when the ICFER.TMOE bit is 1. It detects a hung state that the SCL0 line is stuck low or high during the following conditions:

- The bus is busy (ICCR2.BBSY flag is 1) in master mode (ICCR2.MST bit is 1).
- The RIIC's own slave address is detected (ICSR1 register is not 00h) and the bus is busy (ICCR2.BBSY flag is 1) in slave mode (ICCR2.MST bit is 0).
- The bus is free (ICCR2.BBSY flag is 0) while generation of a start condition is requested (ICCR2.ST bit is 1).

The internal counter of the timeout function works using the internal reference clock (IICφ) set by the ICMR1.CKS[2:0] bits as a count source. It functions as a 16-bit counter when long mode is selected (ICMR2.TMOS bit is 0) or a 14-bit counter when short mode is selected (TMOS bit is 1).
The SCL0 line level (low/high or both levels) during which this counter is activated can be selected by the setting of bits TMOH and TMOL in the ICMR2 register. If both bits TMOL and TMOH are set to 0, the internal counter does not work.

Downloaded from Arrow.com.

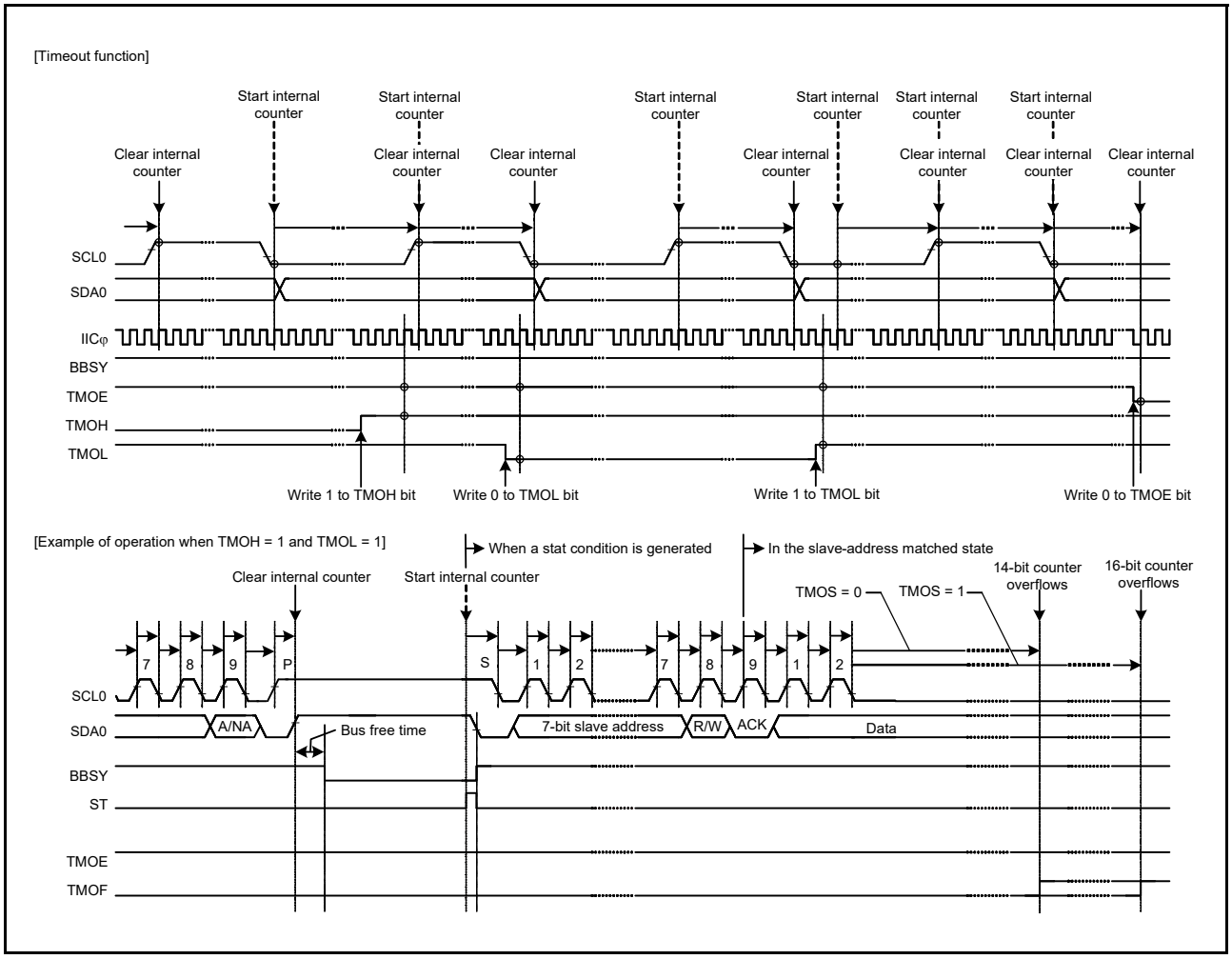


Figure 35.38 Timeout Function

Downloaded from Arrow.com.

35.11.2 Additional SCL Output Function

In master mode, the RIIC module has a facility for the output of additional SCL to release the SDA0 line from being held low by the slave device due to the master being out of synchronization with the slave device.
This function is mainly used in master mode to release the SDA0 line from the state of being stuck low by including additional SCL output from the RIIC with single cycles of the SCL as the unit if the RIIC cannot generate a stop condition because the slave device is holding the SDA0 line low. Do not use this function in normal situations. Using it when communications are proceeding correctly will lead to malfunctions.
When the ICCR1.CLO bit is set to 1, an additional clock pulse at the frequency set by the ICMR1.CKS[2:0] bits and the ICBRH and ICBRL registers is output from the SCL0 pin. After output of this clock pulse, the CLO bit automatically becomes 0. The SCL0 pin is held low when the ICCR2.BBSY flag is 1 and held high when the BBSY flag is 0.
Consecutive additional clock pulses can be output by writing 1 to the CLO bit after confirming the CLO bit to be 0.
When the RIIC module is in master mode and the slave device is holding the SDA0 line low because synchronization with the slave device has been lost due to the effects of noise, etc., the output of a stop condition is not possible. The additional SCL output function can be used to output additional clock pulses one by one to make the slave device release the SDA0 line from being held low, thus recovering the bus from an unusable state. Release of the SDA0 line by the slave device can be monitored by reading the ICCR1.SDAI bit. After confirming release of the SDA0 line by the slave device, complete communications by regenerating a stop condition.
Use this function with the ICFER.MALE bit set to 0 (master arbitration-lost detection disabled).

Conditions for using the ICCR1.CLO bit

- When the bus is free (ICCR2.BBSY flag is 0) or in master mode (ICCR2.MST bit is 1 and ICCR2.BBSY flag is 1)
- When the communication device does not hold the SCL0 line low

Figure 35.39 shows the operation timing of the additional SCL output function (CLO bit).

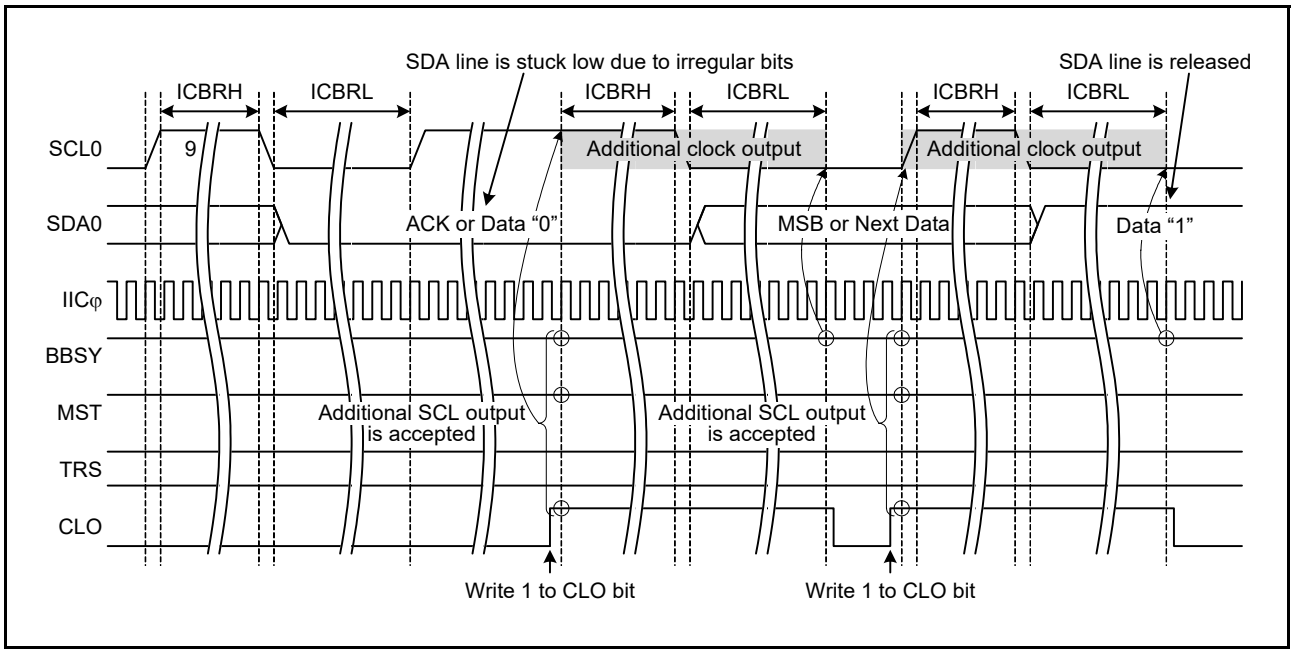


Figure 35.39 Additional SCL Output Function (CLO Bit)

Downloaded from Arrow.com.

35.11.3 RIIC Reset and Internal Reset

The RIIC module incorporates a function for resetting itself. There are two types of reset. One is referred to as an RIIC reset; this initializes all registers including the ICCR2.BBSY flag. The other is referred to as an internal reset; this releases the RIIC from the slave-address matched state and initializes the internal counter while retaining other settings. After applying a reset, be sure to set the ICCR1.IICRST bit to 0.

Both types of reset are effective for release from bus-hung states because both restore the output state of the SCL0 and SDA0 pins to the high-impedance state.

Applying a reset during slave operation may lead to a loss of synchronization between the master device clock and the slave device clock, so avoided this where possible. Note that monitoring of the bus state, such as for the presence of a start condition, is not possible during an RIIC reset (bits ICE and IICRST in the ICCR1 register are 01b).

For a detailed description of the RIIC and internal resets, refer to section 35.14, Initialization of Registers and Functions When a Reset is Applied or a Condition is Detected.

Downloaded from Arrow.com.

35.12 SMBus Operation

The RIIC is available for data communication conforming to the SMBus (Version 2.0). To perform SMBus communication, set the ICMR3.SMBS bit to 1. To use the transfer rate within a range of 10 kbps to 100 kbps of the SMBus specification, set the ICMR1.CKS[2:0] bits, the ICBRH register, and the ICBRL register. In addition, determine the values of the ICMR2.DLCS bit and the ICMR2.SDDL[2:0] bits to meet the data hold time (300 ns (min.)). If the RIIC is used only as a slave device, the transfer rate setting is not necessary, whereas the ICBRL register needs to be set to a value longer than the data setup time (250 ns).
For the SMBus device default address (1100 001b), use one of the slave address registers L0 to L2 (registers SARL0, SARL1, and SARL2), and set the corresponding SARUy.FS bit (7-bit/10-bit address format select) (y = 0 to 2) to 0 (7-bit address format).
When transmitting the UDID (Unique Device Identifier), set the ICFER.SALE bit to 1 to enable the slave arbitration-lost detection function.

35.12.1 SMBus Timeout Measurement

(1) Measuring timeout of slave device

The following period (timeout interval: $T_{LOW:SEXT}$) must be measured for slave devices in SMBus communication.

- From start condition to stop condition

To measure timeout for slave devices, measure the period from start condition detection to stop condition detection with the MTU or TMR timer using a start condition detection interrupt (STI) and stop condition detection interrupt (SPI) of the RIIC. The measured timeout period must be within the cumulative clock low extend time (slave device) ($T_{LOW:SEXT}$: 25 ms (max.)) of the SMBus specification.
If the time measured with the MTU or TMR exceeds the detect clock low timeout ($T_{TIMEOUT}$: 25 ms (min.)) of the SMBus specification, the slave device must release the bus by writing 1 to the ICCR1.IICRST bit to apply an internal reset of the RIIC. When an internal reset is applied to the RIIC, it stops driving the SCL0 and SDA0 pins of the bus and makes the SCL0/SDA0 pin outputs high-impedance, thus releasing the bus.

(2) Measuring timeout of master device

The following periods (timeout interval: $T_{LOW:MEXT}$) must be measured for master devices in SMBus communication.

- From start condition to acknowledgment bit
- Between acknowledgment bits
- From acknowledgment bit to stop condition

To measure timeout for master devices, measure these periods with the MTU or TMR timer using a start condition detection interrupt (STI), stop condition detection interrupt (SPI), and transmission end interrupt (TEI) or receive data full interrupt (RXI) of the RIIC. The measured timeout period must be within the cumulative clock low extend time (master device) ($T_{LOW:MEXT}$: 10 ms (max.)) of the SMBus specification, and the total of all $T_{LOW:MEXT}$ from start condition to stop condition must be within $T_{LOW:SEXT}$: 25 ms (max.).
For the ACK receive timing (rising edge of the ninth SCL), monitor the ICSR2.TEND flag in master transmit mode (master transmitter) and the ICSR2.RDRF flag in master receive mode (master receiver). For this reason, perform bytewise transmit operation in master transmit mode, and hold the ICMR3.RDRFS bit 0 until the byte just before reception of the final byte in master receive mode. While the RDRFS bit is 0, the RDRF flag is set to 1 at the rising edge of the ninth SCL.
If the period measured with the MTU or TMR exceeds the cumulative clock low extend time (master device) ($T_{LOW:MEXT}$: 10 ms (max.)) of the SMBus specification or the total of measured periods exceeds the detect clock low timeout ($T_{TIMEOUT}$: 25 ms (min.)) of the SMBus specification, the master device must stop the transaction by generating a stop condition. In master transmit mode, immediately stop the transmit operation (writing data to the ICDRT register).

Downloaded from Arrow.com.

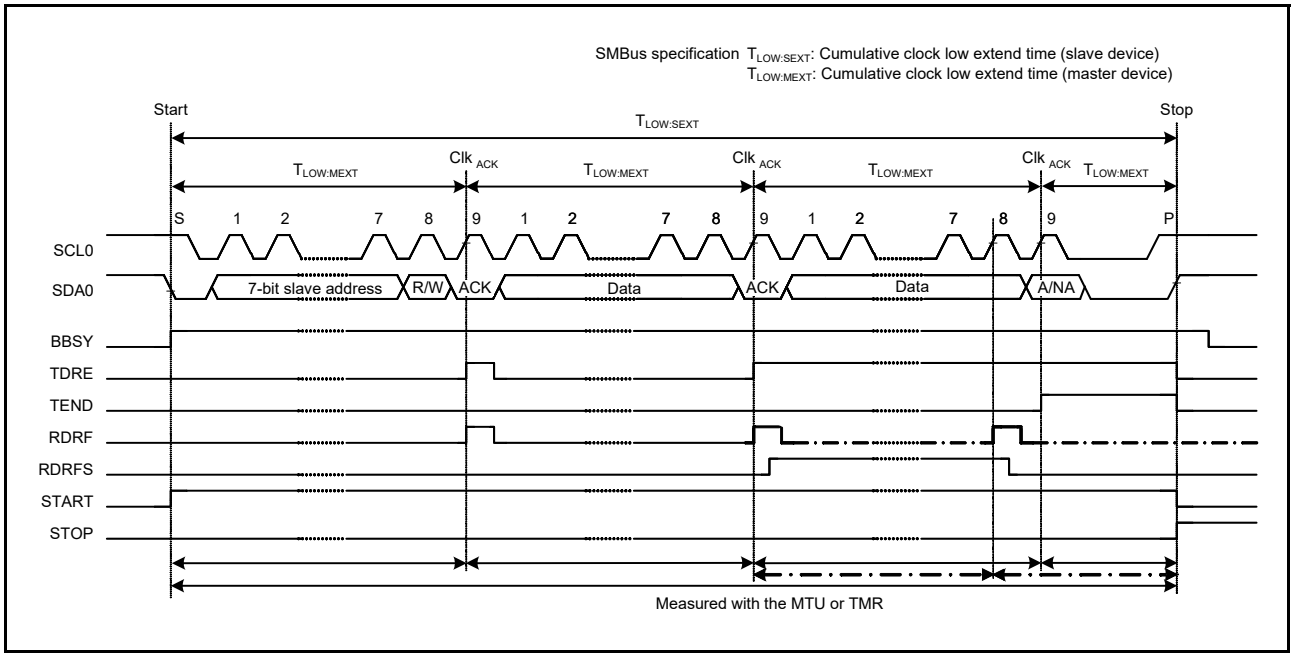


Figure 35.40 SMBus Timeout Measurement

35.12.2 Packet Error Code (PEC)

This MCU incorporates a CRC calculator. The CRC calculator enables transmission of a packet error code (PEC) or checking the received data of the SMBus in data communication of the RIIC. For the CRC generating polynomials of the CRC calculator, refer to section 39, CRC Calculator (CRC).

The PEC data in master transmit mode can be generated by writing all transmit data to the CRC data input register (CRCDIR) in the CRC calculator.

The PEC data in master receive mode can be checked by writing all receive data to CRCDIR in the CRC calculator and comparing the obtained value in the CRC data output register (CRCDOR) with the received PEC data.

To send ACK or NACK according to the match or mismatch result when the final byte is received as a result of the PEC code check, set the ICMR3.RDRFS bit to 1 before the rising edge of the eighth SCL during reception of the final byte, and hold the SCL0 line low at the falling edge of the eighth clock pulse.

35.12.3 SMBus Host Notification Protocol (Notify ARP Master Command)

In communications over an SMBus, a slave device can temporarily act as a master device to notify the SMBus host (or ARP master) of its own slave address or to request its own slave address from the SMBus host.

For a product of this MCU to operate as an SMBus host (or ARP master), the host address (0001 000b) sent from the slave device must be detected as a slave address, so the RIIC has a function for detecting the host address. To detect the host address as a slave address, set the ICMR3.SMBS bit and the ICSER.HOAE bit to 1. Operation after the host address has been detected is the same as normal slave operation.

Downloaded from Arrow.com.

35.13 Interrupt Sources

The RIIC generates four types of interrupt requests: communication error or communication event (arbitration-lost detection, NACK detection, timeout detection, start condition detection, and stop condition detection), receive data full, transmit data empty, and transmission end.

Table 35.6 lists details of the several interrupt requests. The receive data full and transmit data empty interrupt requests allow the DTC or DMAC to start data transfer.

Table 35.6 Interrupt Sources

Symbol	Interrupt Source	Interrupt Flag	Start DTC/DMA Transfer	Interrupt Generation Condition
EEI	Communication error/ communication event	AL	Not possible	AL = 1 and ALIE = 1
		NACKF		NACKF = 1 and NAKIE = 1
		TMOF		TMOF = 1 and TMOIE = 1
		START		START = 1 and STIE = 1
		STOP		STOP = 1 and SPIE = 1
RXI*2	Receive data full	RDRF	Possible	RDRF = 1 and RIE = 1
TXI*1	Transmit data empty	TDRE	Possible	TDRE = 1 and TIE = 1
TEI*3	Transmission end	TEND	Not possible	TEND = 1 and TEIE = 1

Note: There is a delay time between the execution of a write instruction for a peripheral module by the CPU and actual writing to the module. Thus, when an interrupt flag has been cleared or an interrupt request has been masked, read the relevant flag again to check whether clearing or masking has been completed, and then return from interrupt handling. Returning from interrupt handling without checking that writing to the module has been completed creates a possibility of repeated processing of the same interrupt.

Note 1. Because TXI is an edge-detected interrupted, it does not require clearing. Furthermore, the ICSR2.TDRE flag (a condition for TXI) is automatically set to 0 when data for transmission are written to the ICDRT register or a stop condition is detected (ICSR2.STOP flag is 1).

Note 2. Because RXI is an edge-detected interrupted, it does not require clearing. Furthermore, the ICSR2.RDRF flag (a condition for RXI) is automatically set to 0 when data are read from the ICDRR register.

Note 3. When using the TEI interrupt, clear the ICSR2.TEND flag in the TEI interrupt handling.
Note that the ICSR2.TEND flag is automatically set to 0 when data for transmission are written to the ICDRT register or a stop condition is detected (ICSR2.STOP flag is 1).

Clear the each flag or mask the interrupt request during interrupt handling.

35.13.1 Buffer Operation for TXI and RXI Interrupts

If the conditions for generating a TXI and RXI interrupt are satisfied while the corresponding ICU.IRn.IR flag is 1, the interrupt request is not output for the ICU but retained internally (the capacity for internal retention is one request per source).

An interrupt request that was being retained internally is output to the ICU when the value of the IR flag becomes 0.

Internally retained interrupt requests are automatically cleared under normal conditions of usage.

Internally retained interrupt requests can also be cleared by writing 0 to the corresponding interrupt enable bit in the ICIER register.

Downloaded from Arrow.com.

35.14 Initialization of Registers and Functions When a Reset is Applied or a Condition is Detected

The RIIC can be reset by MCU reset, RIIC reset, and internal reset functions. Table 35.7 lists the reset states of registers and functions when a reset is applied or a condition is detected.

Table 35.7 Reset States of Registers and Functions When a Reset is Applied or a Condition is Detected

		MCU Reset	RIIC Reset (ICE = 0, IICRST = 1)	Internal Reset (ICE = 1, IICRST = 1)	Start Condition/ Restart Condition Detection	Stop Condition Detection
ICCR1	SDAO, SCLO	To be reset	To be reset	To be reset	Retained	Retained
	IICRST, ICE		Retained	Retained		
	Others		To be reset			
ICCR2	ST, RS	To be reset	To be reset	To be reset	To be reset	Retained
	SP					To be reset
	TRS				See note 1	
	MST				See note 1	
	BBSY			Retained	Becomes 1	
ICMR1	BC[2:0]	To be reset	To be reset	To be reset	To be reset	Retained
	Others			Retained	Retained	
ICMR2		To be reset	To be reset	Retained	Retained	Retained
ICMR3	ACKBT	To be reset	To be reset	Retained	Retained	To be reset
	Others					Retained
ICFER		To be reset	To be reset	Retained	Retained	Retained
ICSER		To be reset	To be reset	Retained	Retained	Retained
ICIER		To be reset	To be reset	Retained	Retained	Retained
ICSR1		To be reset	To be reset	To be reset	Retained	To be reset
ICSR2	START	To be reset	To be reset	To be reset	Becomes 1	To be reset
	STOP				Retained	Becomes 1
	TEND					To be reset
	TDRE				See note 1	
	Others				Retained	Retained
SARL0, SARL1, SARL2, SARU0, SARU1, SARU2		To be reset	To be reset	Retained	Retained	Retained
ICBRH, ICBRL		To be reset	To be reset	Retained	Retained	Retained
ICDRT		To be reset	To be reset	Retained	Retained	Retained
ICDRR		To be reset	To be reset	Retained	Retained	Retained
ICDRS		To be reset	To be reset	To be reset	Retained	Retained
Timeout function		To be reset	To be reset	To be reset	Operation	Operation
Bus free time measurement		To be reset	To be reset	Operation	Operation	Operation

Note 1. This bit is not reset. This bit becomes 0 or 1 in accordance with the conditions.

Downloaded from Arrow.com.

35.15 Event Link Function (Output)

The RIIC0 handles event output for the event link controller (ELC) corresponding to the following sources.

- Communication error/communication event
- Receive data full
- Transmit data empty
- Transmission end

35.15.1 Interrupt Handling and Event Linking

The RIIC has four types of interrupts: communication error or communication event (arbitration-lost detection, NACK detection, timeout detection, start condition detection, or stop condition detection), receive data full, transmit data empty, and transmission end. Each of these has an enable bit to control enabling and disabling of the interrupt signal. An interrupt request signal is output for the ICU when an interrupt source condition is satisfied while the setting of the corresponding enable bit is enabled.
The corresponding event signals are sent to other modules via the ELC when the interrupt source conditions are satisfied, regardless of the settings of the interrupt enable bits.
For details on interrupt sources, see Table 35.6.

Downloaded from Arrow.com.

35.16 Usage Notes

35.16.1 Setting Module Stop Function

Module stop state can be entered or released using module stop control register B (MSTPCRB). The initial setting is for operation of the RIIC to be stopped. RIIC register access is enabled by releasing the module stop state.
For details on module stop control register B, refer to section 11, Low Power Consumption.

35.16.2 Notes on Starting Transfer

If the IR flag corresponding to the RIIC interrupt is 1 when transfer is started (ICCR1.ICE bit is 1), follow the procedure below to clear interrupts before enabling operations. Starting transfer with the IR flag set to 1 while the ICCR1.ICE bit is 1 leads to an interrupt request being internally retained after transfer starts, and this can lead to unanticipated behavior of the IR flag.

1. Confirm that the ICCR1.ICE bit is 0.
2. Set the relevant interrupt enable bits (ICIER.TIE, etc.) on the peripheral function side to 0.
3. Read the relevant interrupt enable bits (ICIER.TIE, etc.) on the peripheral function side and confirm that its value is 0.
4. Set the IR flag to 0.

Downloaded from Arrow.com.

36. CAN Module (RSCAN)

36.1 Overview

This MCU incorporates the Controller Area Network (CAN) module with one channel of CAN protocol controller conforming to the ISO 11898-1 standard. Table 36.1 shows the CAN module specifications. Figure 36.1 shows the CAN module block diagram. Table 36.2 lists the I/O pins of the CAN module.
In this section, the following variables indicate the number of registers.

- j (j = 0 to 15): Receive rule entry register number
  (GAFLIDLj, GAFLIDHj, GAFLMLj, GAFLMHj, GAFLPLj, GAFLPHj)
- m (m = 0, 1): Receive FIFO buffer number
- n (n = 0 to 15): Receive buffer number
- p (p = 0 to 3): Transmit buffer number
- r (r = 0 to 127): RAM test register (RPGACCr) number

Table 36.1 CAN Module Specifications (1/2)

Item	Specification
Number of channels	1
Protocol	ISO 11898-1 compliant
Communication speed	• Maximum 1 Mbps $\text{Communication speed (CAN bit time clock)} = \frac{1}{\text{CAN bit time}}$ CAN bit time = CANTq × Tq count per bit $\text{CANTq} = \frac{\text{CFGL.BRP[9:0] bits} + 1}{\text{fCAN}}$ Tq: Time quantum fCAN: Frequency of CAN clock source (selected by the GCFGL.DCS bit)
Buffer	20 buffers in total • Individual buffers: 4 buffers (4 buffers for one channel) Transmit buffer: 4 buffers per a channel • Shared buffers: 16 buffers Receive buffer: 0 to 16 buffers Receive FIFO buffer: 2 FIFO buffers (up to 16 buffers allocatable to each) Transmit/receive FIFO buffer: A FIFO buffer per a channel (up to 16 buffers allocatable to each)
Reception function	• Receives data frames and remote frames. • Selects ID format (standard ID, extended ID, or both IDs) to be received. • Sets interrupt enable/disable for each FIFO. • Mirror function (to receive messages transmitted from the own CAN node) • Timestamp function (to record message reception time as a 16-bit timer value)
Reception filter function	• Selects receive messages according to 16 receive rules. • Sets the number of receive rules (0 to 16) for each channel. • Acceptance filter processing: Sets ID and mask for each receive rule. • DLC filter processing: Sets DLC check value for each receive rule.
Receive message transfer function	• Routing function to transfer receive messages to arbitrary destinations (can be transferred to up to 2 buffers). Transfer destination: Receive buffer, receive FIFO buffer, and transmit/receive FIFO buffer • Label addition function Stores label information together when storing a message in a receive buffer and FIFO buffer.
Transmission function	• Transmits data frames and remote frames. • Selects ID format (standard ID, extended ID, or both IDs) to be transmitted. • Sets interrupt enable/disable for each transmit buffer and transmit/receive FIFO buffer. • Selects ID priority transmission or transmit buffer number priority transmission. • Transmit abort function (completion of the abort can be confirmed with the flag) • One-shot transmission function
Interval transmission function	Sets message transmission interval time (transmit mode of transmit/receive FIFO buffers)

Downloaded from Arrow.com.

Table 36.1 CAN Module Specifications (2/2)

Item	Specification
Transmit history function	Stores the history information of transmitted messages.
Bus off recovery mode selection	Selects a method of returning from bus off state. • ISO 11898-1 compliant • Automatic transition to channel halt mode at bus-off entry • Automatic transition to channel halt mode at bus-off end • Transition to channel halt mode by a program • Transition to the error-active state by a program (forcible return from the bus off state)
Error status monitoring	• Monitors CAN protocol errors (stuff error, form error, ACK error, CRC error, bit error, ACK delimiter error, and bus dominant lock). • Detects error status transitions (error warning, error passive, bus off entry, and bus off recovery) • Reads the error counter. • Monitors DLC errors.
Interrupt source	5 sources • Global (2 sources) Global receive FIFO interrupt Global error interrupt • Channel (3 sources/channel) Channel transmit interrupt – Transmit complete interrupt – Transmit abort interrupt – Transmit/receive FIFO transmit complete interrupt – Transmit history interrupt Transmit/receive FIFO receive interrupt Channel error interrupt
CAN clock source	Peripheral module clock (PCLK), CANMCLK
Test function	Test function for user evaluation • Listen-only mode • Self-test mode 0 (external loopback) • Self-test mode 1 (internal loopback) • RAM test (read/write test)
Low power consumption	Module stop state can be set.

Downloaded from Arrow.com.

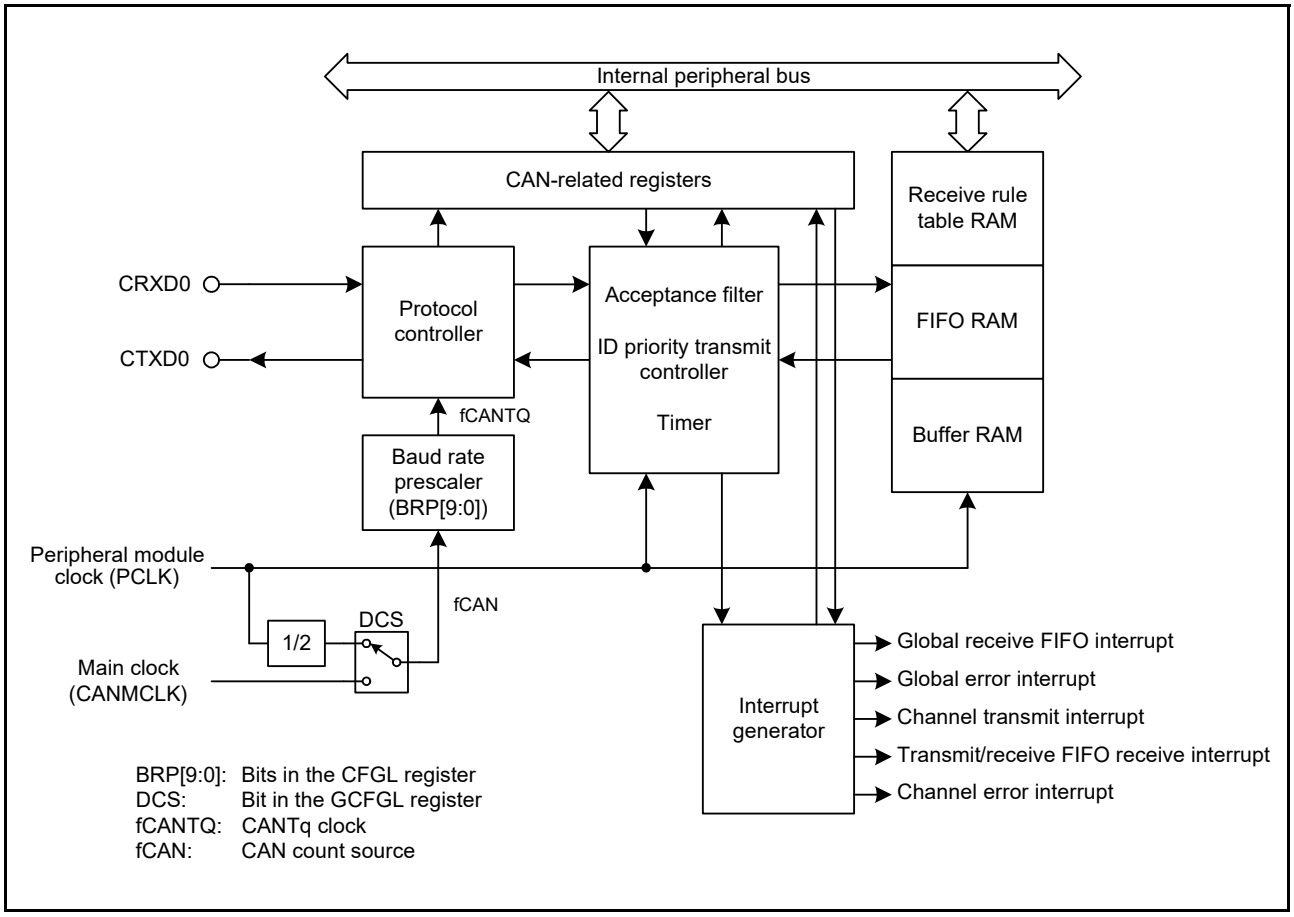


Figure 36.1 CAN Module Block Diagram

- CRXD0/CTXD0: CAN input/output pins
- Protocol controller: Handles CAN protocol processing such as bus arbitration, bit timing at transmission and reception, stuffing, and error handling, etc.
- Receive rule table RAM: Stores the rules for filtering received messages. Each receive rule specifies an ID/ frame format/data length code of the message to be received, a label to be attached to the message that has passed through the filter, and the location of such message to be stored.
- FIFO RAM: Includes three 16-stage FIFO buffers. There are two FIFOs for reception only and one FIFO for transmission or reception.
- Buffer RAM: Used as a transmit and receive buffer. There are 4 buffers for transmission and 16 buffers for reception.
- Acceptance filter: Performs filtering of received messages.
- Timer: There are a timer for timestamp function during reception and a timer which determines the message transmission intervals while using the transmit FIFO buffer.

Table 36.2 I/O Pins of the CAN Module

Pin Name	I/O	Description
CRXD0	Input	Receive data input pins of the RSCAN0
CTXD0	Output	Transmit data output pins of the RSCAN0

Downloaded from Arrow.com.

36.2 Register Descriptions

36.2.1 Bit Configuration Register L (CFGL)

Address(es): RSCAN0.CFGL 000A 8300h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	BRP[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b9 to b0	BRP[9:0]	Prescaler Division Ratio Set	When these bits are set to P (0 to 1023), the baud rate prescaler divides fCAN by P + 1.	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Modify the CFGL register only in channel reset mode or channel halt mode. Set this register in channel reset mode before making a transition to channel communication mode or channel halt mode. For setting bit timing, see section 36.9, Initial Settings.

BRP[9:0] Bits (Prescaler Division Ratio Set)

The CAN Tq clock (fCANTQ) is obtained by the CAN clock source (fCAN) and setting the clock division ratio with the BRP[9:0] bits and one clock cycle of the CAN Tq clock is 1 Time Quantum (Tq).

Downloaded from Arrow.com.

36.2.2 Bit Configuration Register H (CFGH)

Address(es): RSCAN0.CFGH 000A 8302h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	SJW[1:0]		—	TSEG2[2:0]			TSEG1[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	TSEG1[3:0]	Time Segment 1 Control	b3 b0 0 0 0 0: Setting prohibited 0 0 0 1: Setting prohibited 0 0 1 0: Setting prohibited 0 0 1 1: 4 Tq 0 1 0 0: 5 Tq 0 1 0 1: 6 Tq 0 1 1 0: 7 Tq 0 1 1 1: 8 Tq 1 0 0 0: 9 Tq 1 0 0 1: 10 Tq 1 0 1 0: 11 Tq 1 0 1 1: 12 Tq 1 1 0 0: 13 Tq 1 1 0 1: 14 Tq 1 1 1 0: 15 Tq 1 1 1 1: 16 Tq	R/W
b6 to b4	TSEG2[2:0]	Time Segment 2 Control	b6 b4 0 0 0: Setting prohibited 0 0 1: 2 Tq 0 1 0: 3 Tq 0 1 1: 4 Tq 1 0 0: 5 Tq 1 0 1: 6 Tq 1 1 0: 7 Tq 1 1 1: 8 Tq	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b9, b8	SJW[1:0]	Resynchronization Jump Width Control	b9 b8 0 0: 1 Tq 0 1: 2 Tq 1 0: 3 Tq 1 1: 4 Tq	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Modify the CFGH register only in channel reset mode or channel halt mode. Set this register in channel reset mode before making a transition to channel communication mode or channel halt mode. For setting bit timing, see section 36.9, Initial Settings.

TSEG1[3:0] Bits (Time Segment 1 Control)

These bits are used to specify a Tq value for the total length of the propagation time segment (PROP_SEG) and phase buffer segment 1 (PHASE_SEG1). A value of 4 Tq to 16 Tq can be set.

TSEG2[2:0] Bits (Time Segment 2 Control)

These bits are used to specify a Tq value for the length of phase buffer segment 2 (PHASE_SEG2).
A value of 2 Tq to 8 Tq can be set. Set a value smaller than the value of the TSEG1[3:0] bits.

Downloaded from Arrow.com.

SJW[1:0] Bits (Resynchronization Jump Width Control)

These bits are used to specify a Tq value for the resynchronization jump width. A value of 1 Tq to 4 Tq can be set. Set a value equal to or smaller than the value of the TSEG2[3:0] bits.

36.2.3 Control Register L (CTRL)

Address(es): RSCAN0.CTRL 000A 8304h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ALIE	BLIE	OLIE	BORIE	BOEIE	EPIE	EWIE	BEIE	—	—	—	—	RTBO	CSLPR	CHMDC[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Bit	Symbol	Bit Name	Description	R/W
b1, b0	CHMDC[1:0]	Mode Select	b1 b0 0 0: Channel communication mode. 0 1: Channel reset mode. 1 0: Channel halt mode. 1 1: Setting prohibited.	R/W
b2	CSLPR	Channel Stop Mode	0: Other than channel stop mode. 1: Channel stop mode.	R/W
b3	RTBO	Forcible Return from Bus-off	When this bit is set to 1, forcible return from the bus off state is made. This bit is read as 0.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	BEIE	Protocol Error Interrupt Enable	0: Protocol error interrupt is disabled. 1: Protocol error interrupt is enabled.	R/W
b9	EWIE	Error Warning Interrupt Enable	0: Error warning interrupt is disabled. 1: Error warning interrupt is enabled.	R/W
b10	EPIE	Error Passive Interrupt Enable	0: Error passive interrupt is disabled. 1: Error passive interrupt is enabled.	R/W
b11	BOEIE	Bus Off Entry Interrupt Enable	0: Bus off entry interrupt is disabled. 1: Bus off entry interrupt is enabled.	R/W
b12	BORIE	Bus Off Recovery Interrupt Enable	0: Bus off recovery interrupt is disabled. 1: Bus off recovery interrupt is enabled.	R/W
b13	OLIE	Overload Frame Transmit Interrupt Enable	0: Overload frame transmit interrupt is disabled. 1: Overload frame transmit interrupt is enabled.	R/W
b14	BLIE	Bus Lock Interrupt Enable	0: Bus lock interrupt is disabled. 1: Bus lock interrupt is enabled.	R/W
b15	ALIE	Arbitration Lost Interrupt Enable	0: Arbitration lost interrupt is disabled. 1: Arbitration lost interrupt is enabled.	R/W

CHMDC[1:0] Bits (Mode Select)

These bits are used to select a channel mode (channel communication mode, channel reset mode, or channel halt mode). For details, see section 36.3.2, Channel Modes. Setting the CSLPR bit to 1 in channel reset mode allows transition to channel stop mode. Do not set the CHMDC[1:0] bits to 11b. When the CAN module has transitioned to channel halt mode depending on the setting of the CTRH.BOM[1:0] bits, the CHMDC[1:0] bits automatically become 10b.

CSLPR Bit (Channel Stop Mode)

Setting this bit to 1 places the channel in channel stop mode.
Setting this bit to 0 makes the channel leave from channel stop mode.
Do not modify this bit in channel communication mode or channel halt mode.

Downloaded from Arrow.com.

RTBO Bit (Forcible Return from Bus-off)

Setting this bit to 1 (forcible return from the bus off state) in the bus off state forcibly returns the state from the bus off state to the error active state. This bit is automatically set to 0. Setting this bit to 1 sets the STSH.TEC[7:0] and STSH.REC[7:0] flags to 00h and also sets the STSL.BOSTS flag to 0 (not in bus off state).The other registers remain unchanged. No bus off recovery interrupt request due to return from the bus off state is generated. Use this bit only when the CTRH.BOM[1:0] bits are 00b (ISO 11898-1 compliant).

A delay of up to 1 CAN bit time occurs after the RTBO bit is set to 1 until the CAN module transitions to the error active state. Set this bit to 1 in channel communication mode.

BEIE Bit (Protocol Error Interrupt Enable)

When the ERFLL.BEF flag becomes 1 while the BEIE bit is 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

EWIE Bit (Error Warning Interrupt Enable)

When the ERFLL.EWF flag becomes 1 while the EWIE bit is 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

EPIE Bit (Error Passive Interrupt Enable)

When the ERFLL.EPF flag becomes 1 while the EPIE bit is 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

BOEIE Bit (Bus Off Entry Interrupt Enable)

When the ERFLL.BOEF flag becomes 1 while the BOEIE bit is 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

BORIE Bit (Bus Off Recovery Interrupt Enable)

When the ERFLL.BORF flag becomes 1 while the BORIE bit is 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

OLIE Bit (Overload Frame Transmit Interrupt Enable)

When the ERFLL.OVLF flag becomes 1 while the OLIE bit is 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

BLIE Bit (Bus Lock Interrupt Enable)

When the ERFLL.BLF flag becomes 1 while the BLIE bit is 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

ALIE Bit (Arbitration Lost Interrupt Enable)

When the ERFLL.ALF flag becomes 1 while the ALIE bit is 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

Downloaded from Arrow.com.

36.2.4 Control Register H (CTRH)

Address(es): RSCAN0.CTRH 000A 8306h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	CTMS[1:0]		CTME	ERRD	BOM[1:0]		—	—	—	—	TAIE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TAIE	Transmit Abort Interrupt Enable	0: Transmit abort interrupt is disabled. 1: Transmit abort interrupt is enabled.	R/W
b4 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6, b5	BOM[1:0]	Bus Off Recovery Mode Select	b6 b5 0 0: ISO 11898-1 compliant 0 1: Transition to channel halt mode at bus-off entry 1 0: Transition to channel halt mode at bus-off end 1 1: Transition to channel halt mode in the bus off state by a program request	R/W
b7	ERRD	Error Display Mode Select	0: Only the first error is indicated after b14 to b8 in the ERFLL register have all been cleared. 1: The error flags of all errors are indicated.	R/W
b8	CTME	Communication Test Mode Enable	0: Communication test mode is disabled. 1: Communication test mode is enabled.	R/W
b10, b9	CTMS[1:0]	Communication Test Mode Select	b10 b9 0 0: Standard test mode 0 1: Listen-only mode 1 0: Self-test mode 0 (external loopback mode) 1 1: Self-test mode 1 (internal loopback mode)	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

TAIE Bit (Transmit Abort Interrupt Enable)

When transmit abort of the transmit buffer is completed with the TAIE bit set to 1, an interrupt request is generated. Modify this bit only in channel reset mode.

BOM[1:0] Bits (Bus Off Recovery Mode Select)

These bits are used to select a bus off recovery mode of the CAN module.

When the BOM[1:0] bits are set to 00b, return to the error active state from the bus off state is compliant with the ISO 11898-1 standard. That is, the CAN module reenters the CAN communication (error active state) after 11 consecutive recessive bits are detected 128 times. A bus off recovery interrupt request is generated at the time of return from the bus off state. Even if the CTRL.CHMDC[1:0] bits are set to 10b (channel halt mode) before recessive bits are detected 128 times, the CAN module does not transition to channel halt mode until recessive bits are detected 128 times.

When the CAN module reaches the bus off state while the BOM[1:0] bits are set to 01b, the CTRL.CHMDC[1:0] bits are set to 10b and the CAN module transitions to channel halt mode. No bus off recovery interrupt request is generated at the time of return from the bus off state and the STSH.TEC[7:0] and STSH.REC[7:0] flags are set to 00h.

When the CAN module reaches the bus off state when the BOM[1:0] bits are set to 10b, the CTRL.CHMDC[1:0] bits are set to 10b and the CAN module transitions to channel halt mode after return from the bus off state (11 consecutive recessive bits are detected 128 times). A bus off recovery interrupt request is generated at the time of return from the bus off state and the STSH.TEC[7:0] and STSH.REC[7:0] flags are set to 00h.

When the BOM[1:0] bits are set to 11b and the CTRL.CHMDC[1:0] bits are set to 10b while the CAN module is in the bus off state, the CAN module transitions to channel halt mode. No bus off recovery interrupt request is generated at the time of return from the bus off state and the STSH.TEC[7:0] and STSH.REC[7:0] flags are set to 00h. However, if 11 consecutive recessive bits are detected 128 times and the CAN module has recovered to the error active state from the

Downloaded from Arrow.com.

bus off state before the CTRL.CHMDC[1:0] bits are set to 10b, a bus off recovery interrupt request is generated.
If the CPU requests transition to channel reset mode at the same time when the CAN module transitions to channel halt mode (at bus off entry when the BOM[1:0] bits are 01b or at bus off end when the BOM[1:0] bits are 10b), the CPU's request takes precedence. Modify these bits only in channel reset mode.

ERRD Bit (Error Display Mode Select)

This bit is used to control display mode of b14 to b8 in the ERFLL register.
When this bit is 0, only the flags of the first error become 1. If two or more errors occur first, all the flags of detected errors become 1.
When this bit is 1, all the flags of errors that have occurred become 1 regardless of the error occurrence order. Modify this bit only in channel reset mode or channel halt mode.

CTME Bit (Communication Test Mode Enable)

Setting this bit to 1 enables communication test mode. Modify this bit only in channel halt mode. This bit becomes 0 in channel reset mode.

CTMS[1:0] Bits (Communication Test Mode Select)

These bits are used to select a communication test mode. Modify these bits only in channel halt mode. These bits become 0 in channel reset mode.

Downloaded from Arrow.com.

36.2.5 Status Register L (STSL)

Address(es): RSCAN0.STSL 000A 8308h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	COMSTS	RECSTS	TRMSTS	BOSTS	EPSTS	CSLPSTS	CHLTSTS	CRSTSTS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	CRSTSTS	Channel Reset Status Flag	0: Not in channel reset mode 1: In channel reset mode	R
b1	CHLTSTS	Channel Halt Status Flag	0: Not in channel halt mode 1: In channel halt mode	R
b2	CSLPSTS	Channel Stop Status Flag	0: Not in channel stop mode 1: In channel stop mode	R
b3	EPSTS	Error Passive Status Flag	0: Not in error passive state 1: In error passive state	R
b4	BOSTS	Bus Off Status Flag	0: Not in bus off state 1: In bus off state	R
b5	TRMSTS	Transmit Status Flag	0: Bus idle or in reception 1: In transmission or bus off state	R
b6	RECSTS	Receive Status Flag	0: Bus idle, in transmission or bus off state 1: In reception	R
b7	COMSTS	Communication Status Flag	0: Communication is not ready. 1: Communication is ready.	R
b15 to b8	—	Reserved	These bits are read as 0.	R

CRSTSTS Flag (Channel Reset Status Flag)

This flag becomes 1 when the CAN module has transitioned to channel reset mode, and becomes 0 when the CAN module has transitioned to channel communication mode or channel halt mode. This flag remains 1 even if the CAN module transitions from channel reset mode to channel stop mode.

CHLTSTS Flag (Channel Halt Status Flag)

This flag becomes 1 when the CAN module has transitioned to channel halt mode, and becomes 0 when the CAN module has exited channel halt mode.

CSLPSTS Flag (Channel Stop Status Flag)

This flag becomes 1 when the CAN module has transitioned to channel stop mode, and becomes 0 when the CAN module has returned from channel stop mode.

EPSTS Flag (Error Passive Status Flag)

This flag becomes 1 when the CAN module has entered the error passive state (128 ≤ STSH.TEC[7:0] value ≤ 255 or 128 ≤ STSH.REC[7:0] value), and becomes 0 when the CAN module has exited the error passive state or has entered channel reset mode.

BOSTS Flag (Bus Off Status Flag)

This flag becomes 1 when the CAN module has entered the bus off state (STSH.TEC[7:0] value > 255), and becomes 0 when the CAN module has exited the bus off state.

Downloaded from Arrow.com.

TRMSTS Flag (Transmit Status Flag)

This flag becomes 1 when transmission has started, and becomes 0 when the bus has become idle or reception has started. This flag remains 1 in the bus off state.

RECSTS Flag (Receive Status Flag)

This flag becomes 1 when reception has started, and becomes 0 when the bus has become idle or transmission has started.

COMSTS Flag (Communication Status Flag)

This flag indicates that communication is ready.

This flag becomes 1 when the CAN module has detected 11 consecutive recessive bits after it has transitioned from channel reset mode or channel halt mode to channel communication mode. This flag becomes 0 in channel reset mode or channel halt mode.

36.2.6 Status Register H (STSH)

Address(es): RSCAN0.STSH 000A 830Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TEC[7:0]								REC[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Description	R/W
b7 to b0	REC[7:0]	The receive error counter (REC) can be read.	R
b15 to b8	TEC[7:0]	The transmit error counter (TEC) can be read.	R

REC[7:0] Flags

These flags indicate the receive error counter value. For receive error counter increment/decrement conditions, see the CAN standard (ISO 11898-1).

These flags become 00h in channel reset mode.

TEC[7:0] Flags

These flags indicate the transmit error counter value. For transmit error counter increment/decrement conditions, see the CAN standard (ISO 11898-1).

These flags become 00h in channel reset mode.

Downloaded from Arrow.com.

36.2.7 Error Flag Register L (ERFLL)

Address(es): RSCAN0.ERFLL 000A 830Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	ADERR	B0ERR	B1ERR	CERR	AERR	FERR	SERR	ALF	BLF	OVLF	BORF	BOEF	EPF	EWF	BEF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	BEF	Bus Error Flag	0: No channel bus error is detected. 1: Channel bus error is detected.	R/(W) *1
b1	EWF	Error Warning Flag	0: No error warning is detected. 1: Error warning is detected.	R/(W) *1
b2	EPF	Error Passive Flag	0: No error passive is detected. 1: Error passive is detected.	R/(W) *1
b3	BOEF	Bus Off Entry Flag	0: No bus off entry is detected. 1: Bus off entry is detected.	R/(W) *1
b4	BORF	Bus Off Recovery Flag	0: No bus off recovery is detected. 1: Bus off recovery is detected.	R/(W) *1
b5	OVLF	Overload Flag	0: No overload is detected. 1: Overload is detected.	R/(W) *1
b6	BLF	Bus Lock Flag	0: No channel bus lock is detected. 1: Channel bus lock is detected.	R/(W) *1
b7	ALF	Arbitration Lost Flag	0: No arbitration lost is detected. 1: Arbitration lost is detected.	R/(W) *1
b8	SERR	Stuff Error Flag	0: No stuff error is detected. 1: Stuff error is detected.	R/(W) *1
b9	FERR	Form Error Flag	0: No form error is detected. 1: Form error is detected.	R/(W) *1
b10	AERR	ACK Error Flag	0: No ACK error is detected. 1: ACK error is detected.	R/(W) *1
b11	CERR	CRC Error Flag	0: No CRC error is detected. 1: CRC error is detected.	R/(W) *1
b12	B1ERR	Recessive Bit Error Flag	0: No recessive bit error is detected. 1: Recessive bit error is detected.	R/(W) *1
b13	B0ERR	Dominant Bit Error Flag	0: No dominant bit error is detected. 1: Dominant bit error is detected.	R/(W) *1
b14	ADERR	ACK Delimiter Error Flag	0: No ACK delimiter error is detected. 1: ACK delimiter error is detected.	R/(W) *1
b15	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note 1. Only 0 can be written to this bit, to clear the flag. Writing 1 does not affect the flag value.

See the CAN standard (ISO 11898-1) if you want to check error occurrence conditions. To clear each flag of this register, write 0 by the program. These flags cannot be set to 1 by the program. If any of these flags becomes 1 at the timing when the program writes 0 to the flag, the flag becomes 1. Each flag becomes 0 in channel reset mode.
With respect to b14 to b8 in the ERFLL register, if an error is detected with all flags of b14 to b8 set to 0 when the CTRH.ERRD bit is set to 0 (only the first error information is displayed), the corresponding flag becomes 1.

BEF Flag (Bus Error Flag)

This flag becomes 1 when any one of the ADERR, B0ERR, B1ERR, CERR, AERR, FERR, and SERR flags becomes 1.

Downloaded from Arrow.com.

EWF Flag (Error Warning Flag)

This flag becomes 1 only when the STSH.REC[7:0] or STSH.TEC[7:0] value exceeds 95 for the first time. Therefore, if the program writes 0 to this flag with the STSH.REC[7:0] or STSH.TEC [7:0] value remaining over 95, this bit is not set to 1 until both STSH.REC[7:0] and STSH.TEC[7:0] values become 95 or less and then the STSH.REC[7:0] or STSH.TEC[7:0] value exceeds 95 again.

EPF Flag (Error Passive Flag)

This flag becomes 1 when the CAN module becomes error passive state (STSH.REC[7:0] or STSH.TEC[7:0] value > 127). This flag becomes 1 only when the STSH.REC[7:0] or STSH.TEC[7:0] value exceeds 127 for the first time. Therefore, if the program writes 0 to this flag with the STSH.REC[7:0] or STSH.TEC[7:0] value remaining over 127, this bit is not set to 1 until both STSH.REC[7:0] and STSH.TEC[7:0] values become 127 or less and then the STSH.REC[7:0] or STSH.TEC[7:0] value exceeds 127 again.

BOEF Flag (Bus Off Entry Flag)

This flag becomes 1 when the state becomes bus off state (STSH.TEC[7:0] value > 255). This flag also becomes 1 when the state becomes bus off state with the CTRH.BOM[1:0] bits set to 01b (transition to channel halt mode at bus off entry).

BORF Flag (Bus Off Recovery Flag)

This flag becomes 1 when 11 consecutive recessive bits have been detected 128 times and the CAN module returns from the bus off state. However, this flag is not set to 1 if the CAN module returns from the bus off state in any of the following ways before 11 consecutive recessive bits are detected 128 times.

- The CTRL.CHMDC[1:0] bits are set to 01b (channel reset mode).
- The CTRL.RTBO bit is set to 1 (forcible return from the bus off state is made).
- The CTRH.BOM[1:0] bits are set to 01b (transition to channel halt mode at bus off entry).
- The CTRL.CHMDC[1:0] bits are set to 10b (channel halt mode) before 11 consecutive recessive bits are detected 128 times with the CTRH.BOM[1:0] bits set to 11b (transition to channel halt mode upon a request from the program during bus off).

OVLF Flag (Overload Flag)

This flag becomes 1 when the overload frame transmit condition has been detected when performing reception or transmission.

BLF Flag (Bus Lock Flag)

This flag becomes 1 when 32 consecutive dominant bits have been detected on the CAN bus in channel communication mode. After that, detection of the bus lock becomes possible again if either of the following conditions is met.

- A recessive bit is detected after the BLF flag has been modified from 1 to 0.
- The CAN module transitions to channel reset mode and returns to channel communication mode after the BLF flag has been modified from 1 to 0.

ALF Flag (Arbitration Lost Flag)

This flag becomes 1 when an arbitration lost has been detected.

SERR Flag (Stuff Error Flag)

This flag becomes 1 when a stuff error has been detected.

FERR Flag (Form Error Flag)

This flag becomes 1 when a form error has been detected.

Downloaded from Arrow.com.

AERR Flag (ACK Error Flag)

This flag becomes 1 when an ACK error has been detected.

CERR Flag (CRC Error Flag)

This flag becomes 1 when a CRC error has been detected.

B1ERR Flag (Recessive Bit Error Flag)

This flag becomes 1 when a dominant bit has been detected though a recessive bit was transmitted.

B0ERR Flag (Dominant Bit Error Flag)

This flag becomes 1 when a recessive bit has been detected though a dominant bit was transmitted.

ADERR Flag (ACK Delimiter Error Flag)

This flag becomes 1 when a form error has been detected in the ACK delimiter during transmission.

36.2.8 Error Flag Register H (ERFLH)

Address(es): RSCAN0.ERFLH 000A 830Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	CRCREG[14:0]														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b14 to b0	CRCREG[14:0]	CRC Calculation Data	A CRC value calculated based on the transmit message or receive message is indicated.	R
b15	—	Reserved	This bit is read as 0.	R

CRCREG[14:0] Bits (CRC Calculation Data)

When the CTRH.CTME bit is set to 1 (communication test mode is enabled), the CRC value calculated based on the transmit or receive message can be read. When the CTRH.CTME bit is set to 0 (communication test mode is disabled), these bits are read as 0.

Downloaded from Arrow.com.

36.2.9 Global Configuration Register L (GCFGL)

Address(es): RSCAN.GCFGL 000A 8322h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	TSSS	TSP[3:0]				—	—	—	DCS	MME	DRE	DCE	TPRI
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TPRI	Transmit Priority Select	0: ID priority 1: Transmit buffer number priority	R/W
b1	DCE	DLC Check Enable	0: DLC check is disabled. 1: DLC check is enabled.	R/W
b2	DRE	DLC Replacement Enable	0: DLC replacement is disabled. 1: DLC replacement is enabled.	R/W
b3	MME	Mirror Function Enable	0: Mirror function is disabled. 1: Mirror function is enabled.	R/W
b4	DCS	CAN Clock Source Select	0: PCLK 1: CANMCLK (obtained from the main clock)	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11 to b8	TSP[3:0]	Timestamp Clock Source Division	b11 b10 b9 b8 0 0 0 0: Not divided 0 0 0 1: Divided by 2 0 0 1 0: Divided by 4 0 0 1 1: Divided by 8 0 1 0 0: Divided by 16 0 1 0 1: Divided by 32 0 1 1 0: Divided by 64 0 1 1 1: Divided by 128 1 0 0 0: Divided by 256 1 0 0 1: Divided by 512 1 0 1 0: Divided by 1024 1 0 1 1: Divided by 2048 1 1 0 0: Divided by 4096 1 1 0 1: Divided by 8192 1 1 1 0: Divided by 16384 1 1 1 1: Divided by 32768	R/W
b12	TSSS	Timestamp Clock Source Select	0: PCLK 1: CAN bit time clock	R/W
b15 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Modify the GCFGL register only in global reset mode.

TPRI Bit (Transmit Priority Select)

This bit is used to set the transmit priority.

When this bit is set to 0, ID priority is selected and the transmit priority complies with the CAN bus arbitration rule (ISO 11898-1 standard). When this bit is set to 1, transmit buffer number priority is selected and the minimum number of transmit buffer specified for transmission takes precedence.

DCE Bit (DLC Check Enable)

Setting this bit to 1 makes the DLC check function available. Set the GAFLPHj.GAFLDLC[3:0] bits to 0000b before setting the DCE bit to 0.

Downloaded from Arrow.com.

DRE Bit (DLC Replacement Enable)

When the DRE bit is set to 1, the DLC value of the receive rule is stored in the buffer instead of the DLC value of the received message after the DLC value has passed through the DLC filter. In this case, a value of 00h is stored in the data byte that exceeds the DLC value of the receive rule.
When the DCE bit is set to 1 (DLC check is enabled), the DLC replacement function is available.

MME Bit (Mirror Function Enable)

Setting this bit to 1 makes the mirror function available.

DCS Bit (CAN Clock Source Select)

When this bit is set to 0, the peripheral clock (PCLK) divided by 2 is used as the CAN clock source (fCAN).
When this bit is set to 1, CANMCLK obtained from the EXTAL pin is used as the CAN clock source (fCAN).

TSP[3:0] Bits (Timestamp Clock Source Division)

The clock obtained by dividing the clock source selected by the TSSS bit by the TSP[3:0] value is the count source of the timestamp counter.

TSSS Bit (Timestamp Clock Source Select)

This bit is used to select a clock source of the timestamp counter.

36.2.10 Global Configuration Register H (GCFGH)

Address(es): RSCAN.GCFGH 000A 8324h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ITRCP[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	ITRCP[15:0]	Interval Timer Prescaler Set	If the set value is M, PCLK is frequency-divided by M. Setting 0000h is prohibited when the interval timer is in use.	R/W

Modify the GCFGH register only in global reset mode.

ITRCP[15:0] Bits (Interval Timer Prescaler Set)

These bits are used to set a clock source division value of the interval timer for FIFO buffers. For details, see section 36.5.3 (1) Interval Transmission Function.

Downloaded from Arrow.com.

36.2.11 Global Control Register L (GCTRL)

Address(es): RSCAN.GCTRL 000A 8326h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	THLEIE	MEIE	DEIE	—	—	—	—	—	GSLPR	GMDC[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Bit	Symbol	Bit Name	Description	R/W
b1, b0	GMDC[1:0]	Global Mode Select	b1 b0 0 0: Global operating mode 0 1: Global reset mode 1 0: Global test mode 1 1: Setting prohibited	R/W
b2	GSLPR	Global Stop Mode	0: Other than global stop mode 1: Global stop mode	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	DEIE	DLC Error Interrupt Enable	0: DLC error interrupt is disabled. 1: DLC error interrupt is enabled.	R/W
b9	MEIE	FIFO Message Lost Interrupt Enable	0: FIFO message lost interrupt is disabled. 1: FIFO message lost interrupt is enabled.	R/W
b10	THLEIE	Transmit History Buffer Overflow Interrupt Enable	0: Transmit history buffer overflow interrupt is disabled. 1: Transmit history buffer overflow interrupt is enabled.	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

GMDC[1:0] Bits (Global Mode Select)

These bits are used to select the mode of entire CAN module (global operating mode, global reset mode, or global test mode). For details, see section 36.3.1, Global Modes. Setting the GSLPR bit to 1 in global reset mode places the CAN module in global stop mode.

GSLPR Bit (Global Stop Mode)

Setting this bit to 1 places the CAN module in global stop mode.
Setting this bit to 0 makes the CAN module leave from global stop mode.
Do not modify this bit in global operating mode or in global test mode.

DEIE Bit (DLC Error Interrupt Enable)

When the GERFLL.DEF flag becomes 1 while the DEIE bit is 1, an interrupt request is generated. Modify this bit only in global reset mode.

MEIE Bit (FIFO Message Lost Interrupt Enable)

When the GERFLL.MES flag becomes 1 while the MEIE bit is 1, an interrupt request is generated. Modify this bit only in global reset mode.

THLEIE Bit (Transmit History Buffer Overflow Interrupt Enable)

When the GERFLL.THLES flag becomes 1 while the THLEIE bit is 1, an interrupt request is generated. Modify this bit only in global reset mode.

Downloaded from Arrow.com.

36.2.12 Global Control Register H (GCTRH)

Address(es): RSCAN.GCTRH 000A 8328h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TSRST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TSRST	Timestamp Counter Reset	Setting the TSRST bit to 1 resets the timestamp counter. This bit is read as 0.	R/W
b15 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

TSRST Bit (Timestamp Counter Reset)

This bit is used to reset the timestamp counter. When this bit is set to 1, the GTSC register is set to 0000h.

36.2.13 Global Status Register (GSTS)

Address(es): RSCAN.GSTS 000A 832Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	GRAMINIT	GSLPSTS	GHLTSTS	GRSTSTS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	GRSTSTS	Global Reset Status Flag	0: Not in global reset mode 1: In global reset mode	R
b1	GHLTSTS	Global Test Status Flag	0: Not in global test mode 1: In global test mode	R
b2	GSLPSTS	Global Stop Status Flag	0: Not in global stop mode 1: In global stop mode	R
b3	GRAMINIT	CAN RAM Initialization Status Flag	0: CAN RAM initialization is completed. 1: CAN RAM initialization is ongoing.	R
b15 to b4	—	Reserved	These bits are read as 0.	R

GRSTSTS Flag (Global Reset Status Flag)

This flag becomes 1 when the CAN module has transitioned to global reset mode, and becomes 0 when the CAN module has exited global reset mode. This flag remains 1 even when the CAN module has transitioned from global reset mode to global stop mode.

GHLTSTS Flag (Global Test Status Flag)

This flag becomes 1 when the CAN module has transitioned to global test mode, and becomes 0 when the CAN module has exited global test mode.

GSLPSTS Flag (Global Stop Status Flag)

This flag becomes 1 when the CAN module has transitioned to global stop mode, and becomes 0 when the CAN module

Downloaded from Arrow.com.

has returned from global stop mode.

GRAMINIT Flag (CAN RAM Initialization Status Flag)

This flag indicates the initialization status of the CAN RAM.
This flag becomes 1 after the CAN module is enabled, and becomes 0 when CAN RAM initialization is completed.

36.2.14 Global Error Flag Register (GERFLL)

Address(es): RSCAN.GERFLL 000A 832Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	THLES	MES	DEF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DEF	DLC Error Flag	0: No DLC error is present. 1: A DLC error is present.	R/(W) *1
b1	MES	FIFO Message Lost Status Flag	0: No FIFO message lost error is present. 1: A FIFO message lost error is present.	R
b2	THLES	Transmit History Buffer Overflow Status Flag	0: No transmit history buffer overflow is present. 1: A transmit history buffer overflow is present.	R
b7 to b3	—	Reserved	The read value is undefined. The write value should be 0.	R/W

Note 1. Only 0 can be written to this bit, to clear the flag. Writing 1 does not affect the flag value.

All flags in the GERFLL register become 0 in global reset mode.

DEF Flag (DLC Error Flag)

The DEF flag becomes 1 when an error has been detected during the DLC check. This flag can be set to 0 by writing 0 by the program.

MES Flag (FIFO Message Lost Status Flag)

The MES flag becomes 1 when any one of the RFSTSm.RFMLT flags or the CFSTS0.CFMLT flag becomes 1.
This flag becomes 0 when all RFSTSm.RFMLT flags and the CFSTS0.CFMLT flag are set to 0.

THLES Flag (Transmit History Buffer Overflow Status Flag)

The THLES flag becomes 1 when the THLSTS0.THLELT flag becomes 1.
This flag becomes 0 when the THLSTS0.THLELT flag is set to 0.

Downloaded from Arrow.com.

36.2.15 Global Transmit Interrupt Status Register (GTINTSTS)

Address(es): RSCAN.GTINTSTS 000A 8388h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	THIF0	CFTIF0	TAIF0	TSIF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TSIF0	RSCAN0 Transmit Buffer Interrupt Status Flag	0: No transmit buffer transmit complete interrupt request is present. 1: A transmit buffer transmit complete interrupt request is present.	R
b1	TAIF0	RSCAN0 Transmit Buffer Abort Interrupt Status Flag	0: No transmit buffer abort interrupt request is present. 1: A transmit buffer abort interrupt request is present.	R
b2	CFTIF0	RSCAN0 Transmit/Receive FIFO Interrupt Status Flag	0: No transmit/receive FIFO transmit interrupt request is present. 1: A transmit/receive FIFO transmit interrupt request is present.	R
b3	THIF0	RSCAN0 Transmit History Interrupt Status Flag	0: No transmit history interrupt request is present. 1: A transmit history interrupt request is present.	R
b15 to b4	—	Reserved	These bits are read as 0.	R

All flags in the GTINTSTS register become 0 in global reset or channel reset mode.

TSIF0 Flag (RSCAN0 Transmit Buffer Interrupt Status Flag)

The TSIF0 flag becomes 1 when the TMIEC.TMIEp bit is set to 1 (enabling interrupts) and the corresponding TMSTSp.TMTRF[1:0] flags become 10b (transmission has been completed without transmit abort request) or 11b (transmission has been completed with transmit abort request).
This flag becomes 0 when all TMSTSp.TMTRF[1:0] flags that satisfy a condition for setting the TSIF0 flag to 1 are set to 00b. This flag also becomes 0 when the TMIEC.TMIEp bit is set to 0.

TAIF0 Flag (RSCAN0 Transmit Buffer Abort Interrupt Status Flag)

The TAIF0 flag becomes 1 when the CTRH.TAIE bit is set to 1 (enabling interrupts) and the TMSTSp.TMTRF[1:0] flags become 01b (transmit abort has been completed).
This flag becomes 0 when the TMSTSp.TMTRF[1:0] flags, which indicate that the abort of transmission has been completed, are set to 00b.

CFTIF0 Flag (RSCAN0 Transmit/Receive FIFO Interrupt Status Flag)

The CFTIF0 flag becomes 1 when the CFCCL0.CFTXIE bit is set to 1 (enabling interrupts) and the CFSTS0.CFTXIF flag becomes 1 (interrupt request present).
This flag becomes 0 when the CFSTS0.CFTXIF flag is set to 0. This flag also becomes 0 when the CFCCL0.CFTXIE bit is set to 0.

THIF0 Flag (RSCAN0 Transmit History Interrupt Status Flag)

The THIF0 flag becomes 1 when the THLCC0.THLIE bit is set to 1 (enabling interrupts) and the THLSTS0.THLIF flag becomes 1 (interrupt request present).
This flag becomes 0 when the THLSTS0.THLIF flag is set to 0. This flag also becomes 0 when the THLCC0.THLIE bit is set to 0.

Downloaded from Arrow.com.

36.2.16 Timestamp Register (GTSC)

Address(es): RSCAN.GTSC 000A 832Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TS[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Description	Counter Value	R/W
b15 to b0	TS[15:0]	The timestamp counter value can be read.	0000h to FFFFh	R

When the TS[15:0] bits are read, the read value shows the timestamp counter (16-bit free-running counter) value at that time. The TS[15:0] value is captured when the SOF is detected and then stored in the receive buffer or the FIFO buffer.
The timestamp counter is initialized in global reset mode.
The timestamp counter start timing and stop timing depend on the count source.

- When the GCFGL.TSSS value is 0 (PCLK is selected):
  The timestamp counter starts counting when the CAN module has transitioned to global operating mode.
  This counter stops counting when the CAN module has transitioned to global stop mode or global test mode.
- When the GCFGL.TSSS value is 1 (CAN bit time clock is selected):
  The timestamp counter starts counting when the corresponding channel has transitioned to channel communication mode.
  This counter stops counting when the corresponding channel has transitioned to channel reset mode or channel halt mode.

36.2.17 Receive Rule Number Configuration Register (GAFLCFG)

Address(es): RSCAN.GAFLCFG 000A 8330h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	RNC0[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	RNC0[4:0]	RSCAN0 Receive Rule Number Set	Set the number of receive rules of channel 0. Set these bits to a value within a range of 00h to 10h.	R/W
b15 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Modify the GAFLCFG register only in global reset mode.
Up to 16 rules can be registered in the receive rule table.

RNC0[4:0] Bits (RSCAN0 Receive Rule Number Set)

These bits are used to set the number of rules to be registered in the channel 0 receive rule table.
Set these bits to a value within a range of 00h to 10h.

Downloaded from Arrow.com.

36.2.18 Receive Rule Entry Register jAL (GAFLIDLj) (j = 0 to 15)

Address(es): RSCAN.GAFLIDL0 000A 83A0h, RSCAN.GAFLIDL1 000A 83ACh, RSCAN.GAFLIDL2 000A 83B8h, RSCAN.GAFLIDL3 000A 83C4h, RSCAN.GAFLIDL4 000A 83D0h, RSCAN.GAFLIDL5 000A 83DCh, RSCAN.GAFLIDL6 000A 83E8h, RSCAN.GAFLIDL7 000A 83F4h, RSCAN.GAFLIDL8 000A 8400h, RSCAN.GAFLIDL9 000A 840Ch, RSCAN.GAFLIDL10 000A 8418h, RSCAN.GAFLIDL11 000A 8424h, RSCAN.GAFLIDL12 000A 8430h, RSCAN.GAFLIDL13 000A 843Ch, RSCAN.GAFLIDL14 000A 8448h, RSCAN.GAFLIDL15 000A 8454h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	GAFLID[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	GAFLID[15:0]	ID Set L	Set the ID of the receive rule. For the standard ID, set the ID in b10 to b0 and set b15 to b11 to 0.	R/W

Modify the GAFLIDLj register only when the GRWCR.RPAGE bit is set to 0 in global reset mode.

GAFLID[15:0] Bits (ID Set L)

These bits are used to set the ID field of the receive rule. The ID value set by these bits is compared with the ID in the received message during the acceptance filter processing.

Downloaded from Arrow.com.

36.2.19 Receive Rule Entry Register jAH (GAFLIDHj) (j = 0 to 15)

Address(es): RSCAN.GAFLIDH0 000A 83A2h, RSCAN.GAFLIDH1 000A 83AEh, RSCAN.GAFLIDH2 000A 83BAh, RSCAN.GAFLIDH3 000A 83C6h, RSCAN.GAFLIDH4 000A 83D2h, RSCAN.GAFLIDH5 000A 83DEh, RSCAN.GAFLIDH6 000A 83EAh, RSCAN.GAFLIDH7 000A 83F6h, RSCAN.GAFLIDH8 000A 8402h, RSCAN.GAFLIDH9 000A 840Eh, RSCAN.GAFLIDH10 000A 841Ah, RSCAN.GAFLIDH11 000A 8426h, RSCAN.GAFLIDH12 000A 8432h, RSCAN.GAFLIDH13 000A 843Eh, RSCAN.GAFLIDH14 000A 844Ah, RSCAN.GAFLIDH15 000A 8456h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	GAFLIDE	GAFLRTR	GAFLLB	GAFLID[28:16]												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b12 to b0	GAFLID[28:16]	ID Set H	Set the ID of the receive rule. For the standard ID, set these bits to 0.	R/W
b13	GAFLLB	Receive Rule Target Message Select	0: When a message transmitted from another CAN node is received 1: When a message transmitted from own node is received	R/W
b14	GAFLRTR	RTR Select	0: Data frame 1: Remote frame	R/W
b15	GAFLIDE	IDE Select	0: Standard ID 1: Extended ID	R/W

Modify the GAFLIDHj register only when the GRWCR.RPAGE bit is set to 0 in global reset mode.

GAFLID[28:16] Bits (ID Set H)

These bits are used to set the ID field of the receive rule. The ID value set by these bits is compared with the ID in the received message during the acceptance filter processing.

GAFLLB Bit (Receive Rule Target Message Select)

When this bit is set to 0, data processing using the receive rule is performed when receiving messages transmitted from another CAN node.

When this bit is set to 1 when the mirror function is used, data processing using the receive rule is performed when receiving messages transmitted from the own CAN node.

GAFLRTR Bit (RTR Select)

This bit is used to select the frame format (data frame or remote frame) of the receive rule. This bit is compared with the RTR bit in the received message during the acceptance filter processing.

GAFLIDE Bit (IDE Select)

This bit is used to select the ID format (standard ID or extended ID) of the receive rule. This bit is compared with the IDE bit in the received message during the acceptance filter processing.

Downloaded from Arrow.com.

36.2.20 Receive Rule Entry Register jBL (GAFLMLj) (j = 0 to 15)

Address(es): RSCAN.GAFLML0 000A 83A4h, RSCAN.GAFLML1 000A 83B0h, RSCAN.GAFLML2 000A 83BCh, RSCAN.GAFLML3 000A 83C8h, RSCAN.GAFLML4 000A 83D4h, RSCAN.GAFLML5 000A 83E0h, RSCAN.GAFLML6 000A 83ECh, RSCAN.GAFLML7 000A 83F8h, RSCAN.GAFLML8 000A 8404h, RSCAN.GAFLML9 000A 8410h, RSCAN.GAFLML10 000A 841Ch, RSCAN.GAFLML11 000A 8428h, RSCAN.GAFLML12 000A 8434h, RSCAN.GAFLML13 000A 8440h, RSCAN.GAFLML14 000A 844Ch, RSCAN.GAFLML15 000A 8458h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	GAFLIDM[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	GAFLIDM[15:0]	ID Mask L	0: The corresponding ID bit is not compared. 1: The corresponding ID bit is compared.	R/W

Modify the GAFLMLj register only when the GRWCR.RPAGE bit is set to 0 in global reset mode.

GAFLIDM[15:0] Bits (ID Mask L)

These bits are used to mask the corresponding ID bit of the receive rule.

Downloaded from Arrow.com.

36.2.21 Receive Rule Entry Register jBH (GAFLMHj) (j = 0 to 15)

Address(es): RSCAN.GAFLMH0 000A 83A6h, RSCAN.GAFLMH1 000A 83B2h, RSCAN.GAFLMH2 000A 83BEh, RSCAN.GAFLMH3 000A 83CAh, RSCAN.GAFLMH4 000A 83D6h, RSCAN.GAFLMH5 000A 83E2h, RSCAN.GAFLMH6 000A 83EEh, RSCAN.GAFLMH7 000A 83FAh, RSCAN.GAFLMH8 000A 8406h, RSCAN.GAFLMH9 000A 8412h, RSCAN.GAFLMH10 000A 841Eh, RSCAN.GAFLMH11 000A 842Ah, RSCAN.GAFLMH12 000A 8436h, RSCAN.GAFLMH13 000A 8442h, RSCAN.GAFLMH14 000A 844Eh, RSCAN.GAFLMH15 000A 845Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	GAFLIDEM	GAFLRTRM	—	GAFLIDM[28:16]												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b12 to b0	GAFLIDM[28:16]	ID Mask H	0: The corresponding ID bit is not compared. 1: The corresponding ID bit is compared.	R/W
b13	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b14	GAFLRTRM	RTR Mask	0: The RTR bit is not compared. 1: The RTR bit is compared	R/W
b15	GAFLIDEM	IDE Mask	0: The IDE bit is not compared. 1: The IDE bit is compared.	R/W

Modify the GAFLMHj register only when the GRWCR.RPAGE bit is set to 0 in global reset mode.

GAFLIDM[28:16] Bits (ID Mask H)

These bits are used to mask the corresponding ID bit of the receive rule.

GAFLRTRM Bit (RTR Mask)

This bit is used to mask the RTR bit of the receive rule.

GAFLIDEM Bit (IDE Mask)

When this bit is set to 1, filter processing is performed only for messages of the ID format specified by the GAFLIDHj.GAFLIDE bit.

When this bit is set to 0, it is regarded that all received messages have matched the specified ID format. To set the GAFLIDEM bit to 0, set the GAFLMHj.GAFLIDM[28:16] bits and the GAFLMLj.GAFLIDM[15:0] bits to all 0s.

Downloaded from Arrow.com.

36.2.22 Receive Rule Entry Register jCL (GAFLPLj) (j = 0 to 15)

Address(es): RSCAN.GAFLPL0 000A 83A8h, RSCAN.GAFLPL1 000A 83B4h, RSCAN.GAFLPL2 000A 83C0h, RSCAN.GAFLPL3 000A 83CCh, RSCAN.GAFLPL4 000A 83D8h, RSCAN.GAFLPL5 000A 83E4h, RSCAN.GAFLPL6 000A 83F0h, RSCAN.GAFLPL7 000A 83FCh, RSCAN.GAFLPL8 000A 8408h, RSCAN.GAFLPL9 000A 8414h, RSCAN.GAFLPL10 000A 8420h, RSCAN.GAFLPL11 000A 842Ch, RSCAN.GAFLPL12 000A 8438h, RSCAN.GAFLPL13 000A 8444h, RSCAN.GAFLPL14 000A 8450h, RSCAN.GAFLPL15 000A 845Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	GAFLRMV	GAFLRMDP[6:0]							—	—	—	GAFLFDP4	—	—	GAFLFDP1	GAFLFDP0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	GAFLFDP0	Receive FIFO Buffer Select 0	0: Not select a receive FIFO buffer 0 1: Select a receive FIFO buffer 0	R/W
b1	GAFLFDP1	Receive FIFO Buffer Select 1	0: Not select a receive FIFO buffer 1 1: Select a receive FIFO buffer 1	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	GAFLFDP4	RSCAN0 Transmit/Receive FIFO Buffer Select 0	0: Not select an RSCAN0 transmit/receive FIFO buffer 0 1: Select an RSCAN0 transmit/receive FIFO buffer 0	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b14 to b8	GAFLRMDP[6:0]	Receive Buffer Number Select	Set the receive buffer number to store receive messages.	R/W
b15	GAFLRMV	Receive Buffer Enable	0: No receive buffer is used. 1: A receive buffer is used.	R/W

Modify the GAFLPLj register only when the GRWCR.RPAGE bit is set to 0 in global reset mode.

GAFLFDP0 Bit (Receive FIFO Buffer Select 0),
GAFLFDP1 Bit (Receive FIFO Buffer Select 1),
GAFLFDP4 Bit (RSCAN0 Transmit/Receive FIFO Buffer Select 0)

These bits are used to specify FIFO buffers that store receive messages that have passed through the filter. Up to two FIFO buffers are selectable. However, when the GAFLPLj.GAFLRMV bit is set to 1 (a receive buffer is used), up to one FIFO buffer is selectable. Only receive FIFO buffers and the transmit/receive FIFO buffer for which the CFCCH0.CFM[1:0] bits are set to 00b (receive mode) are selectable.

GAFLRMDP[6:0] Bits (Receive Buffer Number Select)

These bits are used to select the number of the receive buffer that stores receive messages that have passed through the filter when the GAFLRMV bit is set to 1. Set these bits to a value smaller than the value set by the RMNB.NRXMB[4:0] bits.

GAFLRMV Bit (Receive Buffer Enable)

When this bit is set to 1, receive messages that have passed through the filter are stored in the receive buffer selected by the GAFLRMDP[6:0] bits.

Downloaded from Arrow.com.

36.2.23 Receive Rule Entry Register jCH (GAFLPHj) (j = 0 to 15)

Address(es): RSCAN.GAFLPH0 000A 83AAh, RSCAN.GAFLPH1 000A 83B6h, RSCAN.GAFLPH2 000A 83C2h, RSCAN.GAFLPH3 000A 83CEh, RSCAN.GAFLPH4 000A 83DAh, RSCAN.GAFLPH5 000A 83E6h, RSCAN.GAFLPH6 000A 83F2h, RSCAN.GAFLPH7 000A 83FEh, RSCAN.GAFLPH8 000A 840Ah, RSCAN.GAFLPH9 000A 8416h, RSCAN.GAFLPH10 000A 8422h, RSCAN.GAFLPH11 000A 842Eh, RSCAN.GAFLPH12 000A 843Ah, RSCAN.GAFLPH13 000A 8446h, RSCAN.GAFLPH14 000A 8452h, RSCAN.GAFLPH15 000A 845Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	GAFLDLC[3:0]				GAFLPTR[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b11 to b0	GAFLPTR[11:0]	Receive Rule Label	Set the 12-bit label information.	R/W
b15 to b12	GAFLDLC[3:0]	Receive Rule DLC	b15 b12 0 0 0 0: 0 or more data bytes (DLC check is disabled) 0 0 0 1: 1 or more data bytes 0 0 1 0: 2 or more data bytes 0 0 1 1: 3 or more data bytes 0 1 0 0: 4 or more data bytes 0 1 0 1: 5 or more data bytes 0 1 1 0: 6 or more data bytes 0 1 1 1: 7 or more data bytes 1 x x x: 8 or more data bytes	R/W

x: Don’t care

Modify the GAFLPHj register only when the GRWCR.RPAGE bit is set to 0 in global reset mode.

GAFLPTR[11:0] Bits (Receive Rule Label)

These bits are used to set a 12-bit label to be attached to messages that have passed through the filter. A label is attached when a message is stored in the receive buffer or the FIFO buffer.

GAFLDLC[3:0] Bits (Receive Rule DLC)

These bits are used to set the minimum data length necessary for receiving messages. If the data length of a message that is being filtered is equal to or larger than the value set by the GAFLDLC[3:0] bits, the message passes the DLC check. Setting these bits to 0000b disables the DLC check function allowing messages with any data length to pass the DLC check.

Downloaded from Arrow.com.

36.2.24 Receive Buffer Number Configuration Register (RMNB)

Address(es): RSCAN.RMNB 000A 8332h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	NRXMB[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	NRXMB[4:0]	Receive Buffer Number Configuration	Set the number of receive buffers. Set a value of 0 to 16.	R/W
b15 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Modify the RMNB register only in global reset mode.

NRXMB[4:0] Bits (Receive Buffer Number Configuration)

These bits are used to set the total number of receive buffers of the CAN module. The maximum value is 16.
Setting these bits to all 0s makes receive buffers unavailable.

36.2.25 Receive Buffer Receive Complete Flag Register (RMND0)

Address(es): RSCAN.RMND0 000A 8334h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RMNS[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	RMNS[15:0]	Receive Buffer Receive Complete Flag n	0: Receive buffer n contains no new message (n = 0 to 15). 1: Receive buffer n contains a new message.	R/W

Write 0 to the RMND0 register in global operating mode or global test mode.

RMNS[15:0] Flags (Receive Buffer Receive Complete Flag n)

Each of the RMNS[15:0] flags becomes 1 when the processing for storing a message in the corresponding receive buffer starts.
To set these flags to 0, write 0 to the corresponding flag by the program. In this case, write this register in 16-bit unit to ensure that only the specified bit is set to 0 and the other bits are set to 1. These bits cannot be set to 0 while a message is being stored. It takes time of 10 clock cycles of PCLK for storing a message.
These flags become 0 in global reset mode.

Downloaded from Arrow.com.

36.2.26 Receive Buffer Register nAL (RMIDLn) (n = 0 to 15)

Address(es): RSCAN.RMIDL0 000A 83A0h, RSCAN.RMIDL1 000A 83B0h, RSCAN.RMIDL2 000A 83C0h, RSCAN.RMIDL3 000A 83D0h, RSCAN.RMIDL4 000A 83E0h, RSCAN.RMIDL5 000A 83F0h, RSCAN.RMIDL6 000A 8400h, RSCAN.RMIDL7 000A 8410h, RSCAN.RMIDL8 000A 8420h, RSCAN.RMIDL9 000A 8430h, RSCAN.RMIDL10 000A 8440h, RSCAN.RMIDL11 000A 8450h, RSCAN.RMIDL12 000A 8460h, RSCAN.RMIDL13 000A 8470h, RSCAN.RMIDL14 000A 8480h, RSCAN.RMIDL15 000A 8490h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RMID[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	RMID[15:0]	Receive Buffer ID Data L	The standard ID or extended ID of received message can be read. Read bits 10 to 0 for standard ID. Bits 15 to 11 are read as 0.	R

This register can be read when the GRWCR.RPAGE bit is 1.

RMID[15:0] Bits (Receive Buffer ID Data L)

These bits indicate the ID of the message stored in the receive buffer.

Downloaded from Arrow.com.

36.2.27 Receive Buffer Register nAH (RMIDHn) (n = 0 to 15)

Address(es): RSCAN.RMIDH0 000A 83A2h, RSCAN.RMIDH1 000A 83B2h, RSCAN.RMIDH2 000A 83C2h, RSCAN.RMIDH3 000A 83D2h, RSCAN.RMIDH4 000A 83E2h, RSCAN.RMIDH5 000A 83F2h, RSCAN.RMIDH6 000A 8402h, RSCAN.RMIDH7 000A 8412h, RSCAN.RMIDH8 000A 8422h, RSCAN.RMIDH9 000A 8432h, RSCAN.RMIDH10 000A 8442h, RSCAN.RMIDH11 000A 8452h, RSCAN.RMIDH12 000A 8462h, RSCAN.RMIDH13 000A 8472h, RSCAN.RMIDH14 000A 8482h, RSCAN.RMIDH15 000A 8492h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RMIDE	RMRTR	—	RMID[28:16]												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b12 to b0	RMID[28:16]	Receive Buffer ID Data H	The standard ID or extended ID of received message can be read. For standard ID, these bits are read as 0.	R
b13	—	Reserved	This bit is read as 0.	R
b14	RMRTR	Receive Buffer RTR	0: Data frame 1: Remote frame	R
b15	RMIDE	Receive Buffer IDE	0: Standard ID 1: Extended ID	R

This register can be read when the GRWCR.RPAGE bit is 1.

RMID[28:16] Bits (Receive Buffer ID Data H)

These bits indicate the ID of the message stored in the receive buffer.

RMRTR Bit (Receive Buffer RTR)

This bit indicates the frame format (data frame or remote frame) of the message stored in the receive buffer.

RMIDE Bit (Receive Buffer IDE)

This bit indicates the ID format (standard ID or extended ID) of the message stored in the receive buffer.

Downloaded from Arrow.com.

36.2.28 Receive Buffer Register nBL (RMTSn) (n = 0 to 15)

Address(es): RSCAN.RMTS0 000A 83A4h, RSCAN.RMTS1 000A 83B4h, RSCAN.RMTS2 000A 83C4h, RSCAN.RMTS3 000A 83D4h, RSCAN.RMTS4 000A 83E4h, RSCAN.RMTS5 000A 83F4h, RSCAN.RMTS6 000A 8404h, RSCAN.RMTS7 000A 8414h, RSCAN.RMTS8 000A 8424h, RSCAN.RMTS9 000A 8434h, RSCAN.RMTS10 000A 8444h, RSCAN.RMTS11 000A 8454h, RSCAN.RMTS12 000A 8464h, RSCAN.RMTS13 000A 8474h, RSCAN.RMTS14 000A 8484h, RSCAN.RMTS15 000A 8494h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RMTS[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	RMTS[15:0]	Receive Buffer Timestamp Data	Timestamp value of the received message can be read.	R

This register can be read when the GRWCR.RPAGE bit is 1.

RMTS[15:0] Bits (Receive Buffer Timestamp Data)

These bits indicate the timestamp value of the message stored in the receive buffer.

Downloaded from Arrow.com.

36.2.29 Receive Buffer Register nBH (RMPTRn) (n = 0 to 15)

Address(es): RSCAN.RMPTR0 000A 83A6h, RSCAN.RMPTR1 000A 83B6h, RSCAN.RMPTR2 000A 83C6h, RSCAN.RMPTR3 000A 83D6h, RSCAN.RMPTR4 000A 83E6h, RSCAN.RMPTR5 000A 83F6h, RSCAN.RMPTR6 000A 8406h, RSCAN.RMPTR7 000A 8416h, RSCAN.RMPTR8 000A 8426h, RSCAN.RMPTR9 000A 8436h, RSCAN.RMPTR10 000A 8446h, RSCAN.RMPTR11 000A 8456h, RSCAN.RMPTR12 000A 8466h, RSCAN.RMPTR13 000A 8476h, RSCAN.RMPTR14 000A 8486h, RSCAN.RMPTR15 000A 8496h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RMDLC[3:0]				RMPTR[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b11 to b0	RMPTR[11:0]	Receive Buffer Label Data	Label information of the received message can be read.	R
b15 to b12	RMDLC[3:0]	Receive Buffer DLC Data	b15 b12 0 0 0 0: 0 data bytes 0 0 0 1: 1 data byte 0 0 1 0: 2 data bytes 0 0 1 1: 3 data bytes 0 1 0 0: 4 data bytes 0 1 0 1: 5 data bytes 0 1 1 0: 6 data bytes 0 1 1 1: 7 data bytes 1 x x x: 8 data bytes	R

x: Don’t care

This register can be read when the GRWCR.RPAGE bit is 1.

RMPTR[11:0] Bits (Receive Buffer Label Data)

These bits indicate the label information of the message stored in the receive buffer.

RMDLC[3:0] Bits (Receive Buffer DLC Data)

These bits indicate the data length of the message stored in the receive buffer.

Downloaded from Arrow.com.

36.2.30 Receive Buffer Register nCL (RMDF0n) (n = 0 to 15)

Address(es): RSCAN.RMDF00 000A 83A8h, RSCAN.RMDF01 000A 83B8h, RSCAN.RMDF02 000A 83C8h, RSCAN.RMDF03 000A 83D8h, RSCAN.RMDF04 000A 83E8h, RSCAN.RMDF05 000A 83F8h, RSCAN.RMDF06 000A 8408h, RSCAN.RMDF07 000A 8418h, RSCAN.RMDF08 000A 8428h, RSCAN.RMDF09 000A 8438h, RSCAN.RMDF010 000A 8448h, RSCAN.RMDF011 000A 8458h, RSCAN.RMDF012 000A 8468h, RSCAN.RMDF013 000A 8478h, RSCAN.RMDF014 000A 8488h, RSCAN.RMDF015 000A 8498h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RMDB1[7:0]								RMDB0[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	RMDB0[7:0]	Receive Buffer Data Byte 0	Data in the message stored in the receive buffer can be read.	R
b15 to b8	RMDB1[7:0]	Receive Buffer Data Byte 1		R

When the RMPTRn.RMDLC[3:0] value is smaller than 1000b, data bytes for which no data is set are read as 00h.
This register can be read when the GRWCR.RPAGE bit is 1.

36.2.31 Receive Buffer Register nCH (RMDF1n) (n = 0 to 15)

Address(es): RSCAN.RMDF10 000A 83AAh, RSCAN.RMDF11 000A 83BAh, RSCAN.RMDF12 000A 83CAh, RSCAN.RMDF13 000A 83DAh, RSCAN.RMDF14 000A 83EAh, RSCAN.RMDF15 000A 83FAh, RSCAN.RMDF16 000A 840Ah, RSCAN.RMDF17 000A 841Ah, RSCAN.RMDF18 000A 842Ah, RSCAN.RMDF19 000A 843Ah, RSCAN.RMDF110 000A 844Ah, RSCAN.RMDF111 000A 845Ah, RSCAN.RMDF112 000A 846Ah, RSCAN.RMDF113 000A 847Ah, RSCAN.RMDF114 000A 848Ah, RSCAN.RMDF115 000A 849Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RMDB3[7:0]								RMDB2[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	RMDB2[7:0]	Receive Buffer Data Byte 2	Data in the message stored in the receive buffer can be read.	R
b15 to b8	RMDB3[7:0]	Receive Buffer Data Byte 3		R

When the RMPTRn.RMDLC[3:0] value is smaller than 1000b, data bytes for which no data is set are read as 00h.
This register can be read when the GRWCR.RPAGE bit is 1.

Downloaded from Arrow.com.

36.2.32 Receive Buffer Register nDL (RMDF2n) (n = 0 to 15)

Address(es): RSCAN.RMDF20 000A 83ACh, RSCAN.RMDF21 000A 83BCh, RSCAN.RMDF22 000A 83CCh, RSCAN.RMDF23 000A 83DCh, RSCAN.RMDF24 000A 83ECh, RSCAN.RMDF25 000A 83FCh, RSCAN.RMDF26 000A 840Ch, RSCAN.RMDF27 000A 841Ch, RSCAN.RMDF28 000A 842Ch, RSCAN.RMDF29 000A 843Ch, RSCAN.RMDF210 000A 844Ch, RSCAN.RMDF211 000A 845Ch, RSCAN.RMDF212 000A 846Ch, RSCAN.RMDF213 000A 847Ch, RSCAN.RMDF214 000A 848Ch, RSCAN.RMDF215 000A 849Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RMDB5[7:0]								RMDB4[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	RMDB4[7:0]	Receive Buffer Data Byte 4	Data in the message stored in the receive buffer can be read.	R
b15 to b8	RMDB5[7:0]	Receive Buffer Data Byte 5		R

When the RMPTRn.RMDLC[3:0] value is smaller than 1000b, data bytes for which no data is set are read as 00h.
This register can be read when the GRWCR.RPAGE bit is 1.

36.2.33 Receive Buffer Register nDH (RMDF3n) (n = 0 to 15)

Address(es): RSCAN.RMDF30 000A 83AEh, RSCAN.RMDF31 000A 83BEh, RSCAN.RMDF32 000A 83CEh, RSCAN.RMDF33 000A 83DEh, RSCAN.RMDF34 000A 83EEh, RSCAN.RMDF35 000A 83FEh, RSCAN.RMDF36 000A 840Eh, RSCAN.RMDF37 000A 841Eh, RSCAN.RMDF38 000A 842Eh, RSCAN.RMDF39 000A 843Eh, RSCAN.RMDF310 000A 844Eh, RSCAN.RMDF311 000A 845Eh, RSCAN.RMDF312 000A 846Eh, RSCAN.RMDF313 000A 847Eh, RSCAN.RMDF314 000A 848Eh, RSCAN.RMDF315 000A 849Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RMDB7[7:0]								RMDB6[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	RMDB6[7:0]	Receive Buffer Data Byte 6	Data in the message stored in the receive buffer can be read.	R
b15 to b8	RMDB7[7:0]	Receive Buffer Data Byte 7		R

When the RMPTRn.RMDLC[3:0] value is smaller than 1000b, data bytes for which no data is set are read as 00h.
This register can be read when the GRWCR.RPAGE bit is 1.

Downloaded from Arrow.com.

36.2.34 Receive FIFO Control Register m (RFCCm) (m = 0, 1)

Address(es): RSCAN.RFCC0 000A 8338h, RSCAN.RFCC1 000A 833Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RFIGCV[2:0]			RFIM	—	RFDC[2:0]			—	—	—	—	—	—	RFIE	RFE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RFE	Receive FIFO Buffer Enable	0: No receive FIFO buffer is used. 1: Receive FIFO buffers are used.	R/W
b1	RFIE	Receive FIFO Interrupt Enable	0: Receive FIFO interrupt is disabled. 1: Receive FIFO interrupt is enabled.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b10 to b8	RFDC[2:0]	Receive FIFO Buffer Depth Configuration	b10 b8 0 0 0: 0 messages 0 0 1: 4 messages 0 1 0: 8 messages 0 1 1: 16 messages 1 0 0: Setting prohibited 1 0 1: Setting prohibited 1 1 0: Setting prohibited 1 1 1: Setting prohibited	R/W
b11	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b12	RFIM	Receive FIFO Interrupt Source Select	0: An interrupt occurs when the condition set by the RFIGCV[2:0] bits is met. 1: An interrupt occurs each time a message has been received.	R/W
b15 to b13	RFIGCV[2:0]	Receive FIFO Interrupt Request Timing Select	b15 b13 0 0 0: When FIFO is 1/8 full. 0 0 1: When FIFO is 2/8 full. 0 1 0: When FIFO is 3/8 full. 0 1 1: When FIFO is 4/8 full. 1 0 0: When FIFO is 5/8 full. 1 0 1: When FIFO is 6/8 full. 1 1 0: When FIFO is 7/8 full. 1 1 1: When FIFO is full.	R/W

RFE Bit (Receive FIFO Buffer Enable)

Setting the RFE bit to 1 makes receive FIFO buffers available. Setting this bit to 0 sets the RFSTSm.RFEMP flag to 1 (the receive FIFO buffer contains no unread message (buffer empty)). Modify this bit only in global operating mode or global test mode.

RFIE Bit (Receive FIFO Interrupt Enable)

Setting the RFIE bit to 1 enables receive FIFO interrupts. Modify this bit when the RFE bit is set to 0 (no receive FIFO buffer is used).

RFDC[2:0] Bits (Receive FIFO Buffer Depth Configuration)

These bits are used to select the number of messages that can be stored in a single receive FIFO buffer. If these bits are set to 000b, do not use any receive FIFO buffer. Modify these bits only in global reset mode.

RFIM Bit (Receive FIFO Interrupt Source Select)

This bit is used to select a FIFO interrupt source. Modify this bit only in global reset mode.

Downloaded from Arrow.com.

RFIGCV[2:0] Bits (Receive FIFO Interrupt Request Timing Select)

These bits are used to specify the fraction of the transmit/receive FIFO buffer (the number of messages is selected by the setting of the RFDC[2:0] bits) that must be filled for the FIFO buffer to generate a receive interrupt request when the RFIM bit is set to 0.

When the RFDC[2:0] bits are set to 001b (4 messages), set the RFIGCV[2:0] bits to 001b, 011b, 101b, or 111b. Modify these bits only in global reset mode.

36.2.35 Receive FIFO Status Register m (RFSTSm) (m = 0, 1)

Address(es): RSCAN.RFSTS0 000A 8340h, RSCAN.RFSTS1 000A 8342h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	RFMC[5:0]						—	—	—	—	RFIF	RFMLT	RFFLL	RFEMP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	RFEMP	Receive FIFO Buffer Empty Status Flag	0: The receive FIFO buffer contains unread messages. 1: The receive FIFO buffer contains no unread message (buffer empty).	R
b1	RFFLL	Receive FIFO Buffer Full Status Flag	0: The receive FIFO buffer is not full. 1: The receive FIFO buffer is full.	R
b2	RFMLT	Receive FIFO Message Lost Flag	0: No receive FIFO message is lost. 1: A receive FIFO message is lost.	R/(W) *1
b3	RFIF	Receive FIFO Interrupt Request Flag	0: No receive FIFO interrupt request is present. 1: A receive FIFO interrupt request is present.	R/(W) *1
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b13 to b8	RFMC[5:0]	Receive FIFO Unread Message Counter	The number of unread messages stored in the receive FIFO buffer is displayed.	R
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Only 0 can be written to this bit, to clear the flag. Writing 1 does not affect the flag value.

RFEMP Flag (Receive FIFO Buffer Empty Status Flag)

This flag becomes 1 when all messages in the receive FIFO buffer have been read. This flag also becomes 1 when the RFCCm.RFE bit is 0 or in global reset mode.

This flag becomes 0 when even a single received message has been stored in the receive FIFO buffer.

RFFLL Flag (Receive FIFO Buffer Full Status Flag)

This flag becomes 1 when the number of messages stored in the receive FIFO buffer matches the FIFO buffer depth set by the RFCCm.RFDC[2:0] bits.

If the number of messages stored in the receive FIFO buffer becomes smaller than the FIFO buffer depth set by the RFCCm.RFDC[2:0] bits, this flag becomes 0. This flag also becomes 0 when the RFCCm.RFE bit is set to 0 (no receive FIFO buffer is used) or in global reset mode.

RFMLT Flag (Receive FIFO Message Lost Flag)

This flag becomes 1 when it is attempted to store a new message while the receive FIFO buffer is full. In this case, the new message is discarded.

This flag becomes 0 in global reset mode or by writing 0 to this flag.

Modify this bit only in global operating mode or global test mode.

Downloaded from Arrow.com.

RFIF Flag (Receive FIFO Interrupt Request Flag)

This flag becomes 1 when the receive FIFO interrupt request generation conditions set by the RFCCm.RFIGCV[2:0] bits (m = 0, 1) and the RFCCm.RFIM bit are met. This flag becomes 0 in global reset mode or by writing 0 to this flag. Modify this bit only in global operating mode or global test mode.

RFMC[5:0] Flags (Receive FIFO Unread Message Counter)

These flags indicate the number of unread messages in the receive FIFO buffer. This flag becomes 00h when the RFCCm.RFE bit is set to 0.

36.2.36 Receive FIFO Pointer Control Register m (RFPCTRm) (m = 0, 1)

Address(es): RSCAN.RFPCTR0 000A 8348h, RSCAN.RFPCTR1 000A 834Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	RFPC[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	RFPC[7:0]	Receive FIFO Pointer	When these bits are set to FFh, the read pointer moves to the next unread message in the receive FIFO buffer. The setting for these bits must be FFh.	W
b15 to b8	—	Reserved	The write value should be 0.	W

RFPC[7:0] Bits (Receive FIFO Pointer)

When the RFPC[7:0] bits are set to FFh, the read pointer moves to the next unread message in the receive FIFO buffer. At this time, the RFSTSm.RFMC[5:0] (receive FIFO unread message counter) value is decremented. Read the RFIDLm, RFIDHm, RFTSm, RFPTRm, and RFDF0m to RFDF3m registers to read messages in the receive FIFO buffer, and then write FFh to the RFPC[7:0] bits.

Write FFh to these bits when the RFCCm.RFE bit is set to 1 (receive FIFO buffers are used) and the RFSTSm.RFEMP flag is 0 (the receive FIFO buffer contains unread messages).

Downloaded from Arrow.com.

36.2.37 Receive FIFO Access Register mAL (RFIDLm) (m = 0, 1)

Address(es): RSCAN.RFIDL0 000A 85A0h, RSCAN.RFIDL1 000A 85B0h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RFID[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	RFID[15:0]	Receive FIFO Buffer ID Data L	The standard ID or extended ID of received message can be read. Read bits 10 to 0 for standard ID. Bits 15 to 11 are read as 0.	R

This register can be read when the GRWCR.RPAGE bit is 1.

RFID[15:0] Bits (Receive FIFO Buffer ID Data L)

These bits indicate the ID of the message stored in the receive FIFO buffer.

36.2.38 Receive FIFO Access Register mAH (RFIDHm) (m = 0, 1)

Address(es): RSCAN.RFIDH0 000A 85A2h, RSCAN.RFIDH1 000A 85B2h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RFIDE	RFRTR	—	RFID[28:16]												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b12 to b0	RFID[28:16]	Receive FIFO Buffer ID Data H	The standard ID or extended ID of received message can be read. For standard ID, these bits are read as 0.	R
b13	—	Reserved	This bit is read as 0.	R
b14	RFRTR	Receive FIFO Buffer RTR	0: Data frame 1: Remote frame	R
b15	RFIDE	Receive FIFO Buffer IDE	0: Standard ID 1: Extended ID	R

This register can be read when the GRWCR.RPAGE bit is 1.

RFID[28:16] Bits (Receive FIFO Buffer ID Data H)

These bits indicate the ID of the message stored in the receive FIFO buffer.

RFRTR Bit (Receive FIFO Buffer RTR)

This bit indicates the frame format (data frame or remote frame) of the message stored in the receive FIFO buffer.

RFIDE Bit (Receive FIFO Buffer IDE)

This bit indicates the ID format (standard ID or extended ID) of the message stored in the receive FIFO buffer.

Downloaded from Arrow.com.

36.2.39 Receive FIFO Access Register mBL (RFTSm) (m = 0, 1)

Address(es): RSCAN.RFTS0 000A 85A4h, RSCAN.RFTS1 000A 85B4h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RFTS[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	RFTS[15:0]	Receive FIFO Buffer Timestamp Data	Timestamp value of the received message can be read.	R

This register can be read when the GRWCR.RPAGE bit is 1.

RFTS[15:0] Bits (Receive FIFO Buffer Timestamp Data)

These bits indicate the timestamp value of the message stored in the receive FIFO buffer.

36.2.40 Receive FIFO Access Register mBH (RFPTRm) (m = 0, 1)

Address(es): RSCAN.RFPTR0 000A 85A6h, RSCAN.RFPTR1 000A 85B6h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RFDLC[3:0]				RFPTR[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b11 to b0	RFPTR[11:0]	Receive FIFO Buffer Label Data	Label information of the received message can be read.	R
b15 to b12	RFDLC[3:0]	Receive FIFO Buffer DLC Data	b15 b14 b13 b12 0 0 0 0: 0 data bytes 0 0 0 1: 1 data byte 0 0 1 0: 2 data bytes 0 0 1 1: 3 data bytes 0 1 0 0: 4 data bytes 0 1 0 1: 5 data bytes 0 1 1 0: 6 data bytes 0 1 1 1: 7 data bytes 1 x x x: 8 data bytes	R

x: Don't care

This register can be read when the GRWCR.RPAGE bit is 1.

RFPTR[11:0] Bits (Receive FIFO Buffer Label Data)

These bits indicate the label information of the message stored in the receive FIFO buffer.

RFDLC[3:0] Bits (Receive FIFO Buffer DLC Data)

These bits indicate the data length of the message stored in the receive FIFO buffer.

Downloaded from Arrow.com.

36.2.41 Receive FIFO Access Register mCL (RFDF0m) (m = 0, 1)

Address(es): RSCAN.RFDF00 000A 85A8h, RSCAN.RFDF01 000A 85B8h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RFDB1[7:0]								RFDB0[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	RFDB0[7:0]	Receive FIFO Buffer Data Byte 0	Data in the message stored in the receive FIFO buffer can be read.	R
b15 to b8	RFDB1[7:0]	Receive FIFO Buffer Data Byte 1		R

When the RFPTRm.RFDLC[3:0] value is smaller than 1000b, data bytes for which no data is set are read as 00h.
This register can be read when the GRWCR.RPAGE bit is 1.

36.2.42 Receive FIFO Access Register mCH (RFDF1m) (m = 0, 1)

Address(es): RSCAN.RFDF10 000A 85AAh, RSCAN.RFDF11 000A 85BAh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RFDB3[7:0]								RFDB2[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	RFDB2[7:0]	Receive FIFO Buffer Data Byte 2	Data in the message stored in the receive FIFO buffer can be read.	R
b15 to b8	RFDB3[7:0]	Receive FIFO Buffer Data Byte 3		R

When the RFPTRm.RFDLC[3:0] value is smaller than 1000b, data bytes for which no data is set are read as 00h.
This register can be read when the GRWCR.RPAGE bit is 1.

Downloaded from Arrow.com.

36.2.43 Receive FIFO Access Register mDL (RFDF2m) (m = 0, 1)

Address(es): RSCAN.RFDF20 000A 85ACh, RSCAN.RFDF21 000A 85BCh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RFDB5[7:0]								RFDB4[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	RFDB4[7:0]	Receive FIFO Buffer Data Byte 4	Data in the message stored in the receive FIFO buffer can be read.	R
b15 to b8	RFDB5[7:0]	Receive FIFO Buffer Data Byte 5		R

When the RFPTRm.RFDLC[3:0] value is smaller than 1000b, data bytes for which no data is set are read as 00h.
This register can be read when the GRWCR.RPAGE bit is 1.

36.2.44 Receive FIFO Access Register mDH (RFDF3m) (m = 0, 1)

Address(es): RSCAN.RFDF30 000A 85AEh, RSCAN.RFDF31 000A 85BEh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RFDB7[7:0]								RFDB6[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	RFDB6[7:0]	Receive FIFO Buffer Data Byte 6	Data in the message stored in the receive FIFO buffer can be read.	R
b15 to b8	RFDB7[7:0]	Receive FIFO Buffer Data Byte 7		R

When the RFPTRm.RFDLC[3:0] value is smaller than 1000b, data bytes for which no data is set are read as 00h.
This register can be read when the GRWCR.RPAGE bit is 1.

Downloaded from Arrow.com.

36.2.45 Transmit/Receive FIFO Control Register 0L (CFCCL0)

Address(es): RSCAN0.CFCCL0 000A 8350h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CFIGCV[2:0]			CFIM	—	CFDC[2:0]			—	—	—	—	—	CFTXIE	CFRXIE	CFE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CFE	Transmit/Receive FIFO Buffer Enable	0: No transmit/receive FIFO buffer is used. 1: Transmit/receive FIFO buffers are used.	R/W
b1	CFRXIE	Transmit/Receive FIFO Receive Interrupt Enable	0: Transmit/receive FIFO receive interrupt is disabled. 1: Transmit/receive FIFO receive interrupt is enabled.	R/W
b2	CFTXIE	Transmit/Receive FIFO Transmit Interrupt Enable	0: Transmit/receive FIFO transmit interrupt is disabled. 1: Transmit/receive FIFO transmit interrupt is enabled.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b10 to b8	CFDC[2:0]	Transmit/Receive FIFO Buffer Depth Configuration	b10 b8 0 0 0: 0 messages 0 0 1: 4 messages 0 1 0: 8 messages 0 1 1: 16 messages 1 0 0: Setting prohibited 1 0 1: Setting prohibited 1 1 0: Setting prohibited 1 1 1: Setting prohibited	R/W
b11	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b12	CFIM	Transmit/Receive FIFO Interrupt Source Select	0: • Receive mode When the number of received messages has met the condition set by the CFIGCV[2:0] bits, a FIFO receive interrupt request is generated. • Transmit mode When the buffer becomes empty upon completion of message transmission, a FIFO transmit interrupt request is generated. 1: • Receive mode A FIFO receive interrupt request is generated each time a message has been received. • Transmit mode A FIFO transmit interrupt request is generated each time a message has been transmitted.	R/W
b15 to b13	CFIGCV[2:0]	Transmit/Receive FIFO Receive Interrupt Request Timing Select	b15 b13 0 0 0: When FIFO is 1/8 full. 0 0 1: When FIFO is 2/8 full. 0 1 0: When FIFO is 3/8 full. 0 1 1: When FIFO is 4/8 full. 1 0 0: When FIFO is 5/8 full. 1 0 1: When FIFO is 6/8 full. 1 1 0: When FIFO is 7/8 full. 1 1 1: When FIFO is full.	R/W

CFE Bit (Transmit/Receive FIFO Buffer Enable)

Setting this bit to 1 makes transmit/receive FIFO buffers available.

When this bit is set to 0 in transmit mode, if a message in the transmit/receive FIFO buffer is being transmitted or to be transmitted next, the transmit/receive FIFO buffer becomes empty after completion of transmission, CAN bus error detection, or arbitration lost. In other cases or in receive mode, the transmit/receive FIFO buffer becomes empty immediately.

Downloaded from Arrow.com.

This bit is set to 0 when the following conditions are met.

- Receive mode: Global reset mode
- Transmit mode: Channel reset mode

Modify this bit only in the following mode.

- Receive mode: Global operating mode or global test mode
- Transmit mode: Channel communication mode or channel halt mode

CFRXIE Bit (Transmit/Receive FIFO Receive Interrupt Enable)

When the CFSTS0.CFRXIF flag becomes 1 while this bit is 1, a transmit/receive FIFO receive interrupt request is generated.
Modify this bit with the CFE bit set to 0.

CFTXIE Bit (Transmit/Receive FIFO Transmit Interrupt Enable)

When the CFSTS0.CFTXIF flag becomes 1 while this bit is 1, a transmit/receive FIFO transmit interrupt request is generated.
Modify this bit with the CFE bit set to 0 (no transmit/receive FIFO buffer is used).

CFDC[2:0] Bits (Transmit/Receive FIFO Buffer Depth Configuration)

These bits are used to set the number of messages that can be stored in a single transmit/receive FIFO buffer.
If these bits are set to 000b, do not use any receive FIFO buffer. Modify these bits only in global reset mode.

CFIM Bit (Transmit/Receive FIFO Interrupt Source Select)

This bit is used to select a transmit/receive FIFO interrupt source. Modify this bit only in global reset mode.

CFIGCV[2:0] Bits (Transmit/Receive FIFO Receive Interrupt Request Timing Select)

These bits are used to specify the fraction of the transmit/receive FIFO buffer (the number of messages is selected by the setting of the CFDC[2:0] bits) that must be filled for the FIFO buffer to generate a receive interrupt request when the CFCCH0.CFM[1:0] bits are set to 00b (receive mode) and the CFIM bit is set to 0.
When the CFDC[2:0] bits are set to 001b (4 messages), set the CFIGCV[2:0] bits to 001b, 011b, 101b, or 111b.
Modify these bits only in global reset mode.

Downloaded from Arrow.com.

36.2.46 Transmit/Receive FIFO Control Register 0H (CFCCH0)

Address(es): RSCAN0.CFCCH0 000A 8352h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CFITT[7:0]								—	—	CFTML[1:0]		CFITR	CFITSS	CFM[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	CFM[1:0]	Transmit/Receive FIFO Mode Select	b1 b0 0 0: Receive mode 0 1: Transmit mode 1 0: Setting prohibited 1 1: Setting prohibited	R/W
b2	CFITSS	Interval Timer Clock Source Select	0: Clock selected by the CFITR bit 1: CAN bit time clock	R/W
b3	CFITR	Transmit/Receive FIFO Interval Timer Resolution	0: Clock obtained by frequency-dividing PCLK by the ITRCP[15:0] value 1: Clock obtained by frequency-dividing PCLK by the ITRCP[15:0] value × 10	R/W
b5, b4	CFTML[1:0]	Transmit Buffer Link Configuration	Set the transmit buffer number to be linked to the transmit/receive FIFO buffer.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	CFITT[7:0]	Message Transmission Interval Configuration	Set a message transmission interval. Set these bits to a value within a range of 00h to FFh.	R/W

CFM[1:0] Bits (Transmit/Receive FIFO Mode Select)

These bits are used to select transmit/receive FIFO mode. Modify these bits only in global reset mode.

CFITSS Bit (Interval Timer Clock Source Select)

Setting this bit to 0 selects the clock selected by the CFITR bit as the clock source for counting by the interval timer.
Setting this bit to 1 selects the CAN bit time clock as the clock source for counting by the interval timer. Set the CFCCL0.CFE bit to 0 (no transmit/receive FIFO buffer is used) before modifying the CFITSS bit.

CFITR Bit (Transmit/Receive FIFO Interval Timer Resolution)

This bit is valid when the setting of the CFITSS bit is 0.
Setting this bit to 0 selects the clock obtained by frequency-dividing PCLK by the GCFGH.ITRCP[15:0] value.
Setting this bit to 1 selects the clock obtained by frequency-dividing PCLK by the GCFGH.ITRCP[15:0] value × 10.
Modifying this bit with the CFCCL0.CFE bit set to 0 (no transmit/receive FIFO buffer is used).

CFTML[1:0] Bits (Transmit Buffer Link Configuration)

These bits are used to set the number of transmit buffer to be linked to the transmit/receive FIFO buffer when the CFM[1:0] bits are set to 01b (transmit mode).
Setting the CFCCL0.CFDC[2:0] bits to 001b or more enables the setting of the CFTML[1:0] bits.
Modify these bits only in global reset mode.

CFITT[7:0] Bits (Message Transmission Interval Configuration)

These bits are used to set a message transmission interval when transmitting messages continuously from a transmit/receive FIFO buffer whose CFM[1:0] bits are set to 01b (transmit mode).
Set the CFCCL0.CFE bit to 0 (no transmit/receive FIFO buffer is used) and then modify the CFITT[7:0] bits.

Downloaded from Arrow.com.

36.2.47 Transmit/Receive FIFO Status Register 0 (CFSTS0)

Address(es): RSCAN0.CFSTS0 000A 8358h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	CFMC[5:0]						—	—	—	CFTXIF	CFRXIF	CFMLT	CFFLL	CFEMP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	CFEMP	Transmit/Receive FIFO Buffer Empty Status Flag	0: The transmit/receive FIFO buffer contains messages. 1: The transmit/receive FIFO buffer contains no message (buffer empty).	R
b1	CFFLL	Transmit/Receive FIFO Buffer Full Status Flag	0: The transmit/receive FIFO buffer is not full. 1: The transmit/receive FIFO buffer is full.	R
b2	CFMLT	Transmit/Receive FIFO Message Lost Flag	0: No transmit/receive FIFO message is lost. 1: A transmit/receive FIFO message is lost.	R/(W) *1
b3	CFRXIF	Transmit/Receive FIFO Receive Interrupt Request Flag	0: No transmit/receive FIFO receive interrupt request is present. 1: A transmit/receive FIFO receive interrupt request is present.	R/(W) *1
b4	CFTXIF	Transmit/Receive FIFO Transmit Interrupt Request Flag	0: No transmit/receive FIFO transmit interrupt request is present. 1: A transmit/receive FIFO transmit interrupt request is present.	R/(W) *1
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b13 to b8	CFMC[5:0]	Transmit/Receive FIFO Message Counter	The number of messages stored in the transmit/receive FIFO buffer is indicated.	R
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Only 0 can be written to this bit, to clear the flag. Writing 1 does not affect the flag value.

CFEMP Flag (Transmit/Receive FIFO Buffer Empty Status Flag)

[Setting conditions]

- When the CFCCH0.CFM[1:0] value is 00b: All messages have been read, or global reset mode.
- When the CFCCH0.CFM[1:0] value is 01b: All messages have been transmitted, or channel reset mode.
- When the CFCCL0.CFE value is 0 (no transmit/receive FIFO buffer is used).
  Note that this flag becomes 1 after transmission completion, CAN bus error detection, or arbitration lost when the message in the transmit/receive FIFO buffer is being transmitted or to be transmitted next.

[Clearing conditions]

- When the CFCCH0.CFM[1:0] value is 00b: Any one of received messages has been stored in the transmit/receive FIFO buffer.
- When the CFCCH0.CFM[1:0] value is 01b: A value FFh has been written to the CFPCTR0 register after data was written to the CFIDL0, CFIDH0, CFPTR0, and CFDF00 to CFDF30 registers.

CFFLL Flag (Transmit/Receive FIFO Buffer Full Status Flag)

[Setting condition]

- When the number of messages stored in the transmit/receive FIFO buffer matches the FIFO buffer depth set by the CFCCL0.CFDC[2:0] bits.

[Clearing conditions]

- When the number of messages stored in the transmit/receive FIFO buffer becomes smaller than the FIFO buffer depth set by the CFCCL0.CFDC[2:0] bits.

Downloaded from Arrow.com.

- When the CFCCL0.CFE value is 0 (no transmit/receive FIFO buffer is used).
  Note that this flag becomes 0 after transmission completion, CAN bus error detection, or arbitration lost when the message in the transmit/receive FIFO buffer is being transmitted or to be transmitted next.
- When CFCCH0.CFM[1:0] value is 00b: In global reset mode
- When CFCCH0.CFM[1:0] value is 01b: In channel reset mode

CFMLT Flag (Transmit/Receive FIFO Message Lost Flag)

[Setting condition]

- When it is attempted to store a new message while the transmit/receive FIFO buffer is full. In this case, the new message is discarded.

[Clearing conditions]

- Write 0 to the CFMLT flag
- When CFCCH0.CFM[1:0] value is 00b: In global reset mode
- When CFCCH0.CFM[1:0] value is 01b: In channel reset mode

Set this flag to 0 in global operating mode or global test mode.

CFRXIF Flag (Transmit/Receive FIFO Receive Interrupt Request Flag)

[Setting condition]

- When CFCCH0.CFM[1:0] value is 00b and interrupt source setting the CFCCL0.CFIM bit is generated.

[Clearing conditions]

- Write 0 to the CFRXIF flag
- When CFCCH0.CFM[1:0] value is 00b: In global reset mode
- When CFCCH0.CFM[1:0] value is 01b: In channel reset mode

Set this flag to 0 in global operating mode or global test mode.

CFTXIF Flag (Transmit/Receive FIFO Transmit Interrupt Request Flag)

[Setting condition]

- When CFCCH0.CFM[1:0] value is 01b and interrupt source setting the CFCCL0.CFIM bit is generated.

[Clearing conditions]

- Write 0 to the CFTXIF flag
- When CFCCH0.CFM[1:0] value is 00b: In global reset mode
- When CFCCH0.CFM[1:0] value is 01b: In channel reset mode

Set this flag to 0 in global operating mode or global test mode.

CFMC[5:0] Flags (Transmit/Receive FIFO Message Counter)

The CFMC[5:0] flags indicate the following values that depend on the setting of the CFCCH0.CFM[1:0] bits.

- When CFM[1:0] value is 01b (transmit mode): Number of untransmitted messages in the buffer
- When CFM[1:0] value is 00b (receive mode): Number of unread received messages in the buffer

These bits are set to 0 when any of the following conditions is met.

- When CFCCH0.CFM[1:0] value is 00b: In global reset mode
- When CFCCH0.CFM[1:0] value is 01b: In channel reset mode

Downloaded from Arrow.com.

36.2.48 Transmit/Receive FIFO Pointer Control Register 0 (CFPCTR0)

Address(es): RSCAN0.CFPCTR0 000A 835Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	CFPC[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	CFPC[7:0]	RSCAN0 Transmit/Receive FIFO Pointer	Receive mode: Writing FFh to these bits moves the read pointer to the next unread message in the transmit/receive FIFO buffer. Transmit mode: Writing FFh to these bits moves the write pointer to the next stage of the transmit/receive FIFO buffer.	W
b15 to b8	—	Reserved	The write value should be 0.	W

CFPC[7:0] Bits (RSCAN0 Transmit/Receive FIFO Pointer)

Receive mode (CFCCH0.CFM[1:0] value is 00b):
Writing FFh to the CFPC[7:0] bits moves the read pointer to the next unread message in the transmit/receive FIFO buffer. At this time, the CFSTS0.CFMC[5:0] value (transmit/receive FIFO message counter) is decremented. Read the CFIDL0, CFIDH0, CFTS0, CFPTR0, and CFDF00 to CFDF30 registers to read messages in the transmit/receive FIFO buffer, and then write FFh to the CFPC[7:0] bits.
Write FFh to these bits when the CFCCL0.CFE bit is 1 (transmit/receive FIFO buffers are used) and the CFSTS0.CFEMP flag is 0 (the transmit/receive FIFO buffer contains messages).

Transmit mode (CFCCH0.CFM[1:0] value is 01b):
Writing FFh to the CFPC[7:0] bits stores the data written to the CFIDL0, CFIDH0, CFPTR0, and CFDF00 to CFDF30 registers in the transmit/receive FIFO buffer and moves the write pointer to the next stage of the transmit/receive FIFO buffer. At this time, the CFSTS0.CFMC[5:0] value is incremented. Write transmit messages to the CFIDL0, CFIDH0, CFPTR0, and CFDF00 to CFDF30 registers and then write FFh to the CFPC[7:0] bits.
Write FFh to these bits when the CFCCL0.CFE bit is 1 and the CFSTS0.CFFLL flag is 0 (the transmit/receive FIFO buffer is not full).

Downloaded from Arrow.com.

36.2.49 Transmit/Receive FIFO Access Register 0AL (CFIDL0)

Address(es): RSCAN0.CFIDL0 000A 85E0h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CFID[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	CFID[15:0]	Transmit/Receive FIFO Buffer ID Data L	When CFCCH0.CFM[1:0] value is 01b (transmit mode): Set standard ID or extended ID. For standard ID, write an ID to b10 to b0 and write 0 to b15 to b11. When CFCCH0.CFM[1:0] value is 00b (receive mode): Standard ID or extended ID in the received message can be read. For standard ID, read b10 to b0. The value read from b15 to b11 are 0.	R/W

Modify this register only when the CFCCH0.CFM[1:0] value is 01b (transmit mode). This register is readable only when the CFCCH0.CFM[1:0] value is 00b (receive mode).
This register can be read/written when the GRWCR.RPAGE bit is 1.

CFID[15:0] Bits (Transmit/Receive FIFO Buffer ID Data L)

These bits indicate the ID of the received message stored in the transmit/receive FIFO buffer when the CFCCH0.CFM[1:0] value is 00b.
When the CFCCH0.CFM[1:0] value is 01b, set the ID of the message to be transmitted from the transmit/receive FIFO buffer.

Downloaded from Arrow.com.

36.2.50 Transmit/Receive FIFO Access Register 0AH (CFIDH0)

Address(es): RSCAN0.CFIDH0 000A 85E2h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CFIDE	CFRTR	THLEN	CFID[28:16]												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b12 to b0	CFID[28:16]	Transmit/Receive FIFO Buffer ID Data H	When CFCCH0.CFM[1:0] value is 01b (transmit mode): Set standard ID or extended ID. For standard ID, write 0 to these bits. When CFCCH0.CFM[1:0] value is 00b (receive mode): Standard ID or extended ID in the received message can be read. For standard ID, these bits are read as 0.	R/W
b13	THLEN	Transmit History Data Store Enable	This bit is valid only when the CFCCH0.CFM[1:0] value is 01b (transmit mode). 0: Transmit history data is not stored in the buffer. 1: Transmit history data is stored in the buffer.	R/W
b14	CFRTR	Transmit/Receive FIFO Buffer RTR	0: Data frame 1: Remote frame	R/W
b15	CFIDE	Transmit/Receive FIFO Buffer IDE	0: Standard ID 1: Extended ID	R/W

Modify this register only when the CFCCH0.CFM[1:0] value is 01b (transmit mode).
This register is readable only when the CFCCH0.CFM[1:0] value is 00b (receive mode).
This register can be read/written when the GRWCR.RPAGE bit is 1.

CFID[28:16] Bits (Transmit/Receive FIFO Buffer ID Data H)

These bits indicate the ID of the received message stored in the transmit/receive FIFO buffer when the CFCCH0.CFM[1:0] value is 00b.
When the CFCCH0.CFM[1:0] value is 01b, set the ID of the message to be transmitted from the transmit/receive FIFO buffer.

THLEN Bit (Transmit History Data Store Enable)

When this bit is set to 1, the transmit history data (label information, buffer number, and buffer type) of transmit messages is stored in the transmit history buffer after transmission is completed.
This bit is enabled when the CFCCH0.CFM[1:0] value is 01b (transmit mode).

CFRTR Bit (Transmit/Receive FIFO Buffer RTR)

This bit indicates the data format (data frame or remote frame) of the received message stored in the transmit/receive FIFO buffer when the CFCCH0.CFM[1:0] value is 00b. When the CFCCH0.CFM[1:0] value is 01b, set the data format of the message to be transmitted from the transmit/receive FIFO buffer.

CFIDE Bit (Transmit/Receive FIFO Buffer IDE)

This bit indicates the ID format (standard ID or extended ID) of the received message stored in the transmit/receive FIFO buffer when the CFCCH0.CFM[1:0] value is 00b. When the CFCCH0.CFM[1:0] value is 01b, set the ID format of the message to be transmitted from the transmit/receive FIFO buffer.

Downloaded from Arrow.com.

36.2.51 Transmit/Receive FIFO Access Register 0BL (CFTS0)

Address(es): RSCAN0.CFTS0 000A 85E4h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CFTS[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	CFTS[15:0]	Transmit/Receive FIFO Buffer Timestamp Data	These bits are valid only when the CFCCH0.CFM[1:0] value is 00b (receive mode). The timestamp value of the received message can be read.	R

This register can be read when the GRWCR.RPAGE bit is 1.

CFTS[15:0] Bits (Transmit/Receive FIFO Buffer Timestamp Data)

These bits indicate the timestamp value of the message stored in the transmit/receive FIFO buffer.

These bits are valid when the CFCCH0.CFM[1:0] value is 00b.

Downloaded from Arrow.com.

36.2.52 Transmit/Receive FIFO Access Register 0BH (CFPTR0)

Address(es): RSCAN0.CFPTR0 000A 85E6h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CFDLC[3:0]				CFPTR[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b11 to b0	CFPTR[11:0]	Transmit/Receive FIFO Buffer Label Data	When CFCCH0.CFM[1:0] value is 01b (transmit mode): Set the label information to be stored in the transmit history buffer. Only CFPTR[7:0] bits are valid. When CFCCH0.CFM[1:0] value is 00b (receive mode): The label information of the received message can be read.	R/W
b15 to b12	CFDLC[3:0]	Transmit/Receive FIFO Buffer DLC Data	b15 b12 0 0 0 0: 0 data bytes 0 0 0 1: 1 data byte 0 0 1 0: 2 data bytes 0 0 1 1: 3 data bytes 0 1 0 0: 4 data bytes 0 1 0 1: 5 data bytes 0 1 1 0: 6 data bytes 0 1 1 1: 7 data bytes 1 x x x: 8 data bytes	R/W

x: Don’t care

Modify this register only when the CFCCH0.CFM[1:0] value is 01b (transmit mode).
This register is readable only when the CFCCH0.CFM[1:0] value is 00b (receive mode).
This register can be read/written when the GRWCR.RPAGE bit is 1.

CFPTR[11:0] Bits (Transmit/Receive FIFO Buffer Label Data)

These bits indicate the label information attached to the received message stored in the transmit/receive FIFO buffer when the CFCCH0.CFM[1:0] value is 00b. When the CFCCH0.CFM[1:0] value is 01b, the CFPTR[7:0] value is stored in the transmit history buffer when message transmission has been completed.

CFDLC[3:0] Bits (Transmit/Receive FIFO Buffer DLC Data)

These bits indicate the data length of the received message stored in the transmit/receive FIFO buffer when the CFCCH0.CFM[1:0] value is 00b. When the CFCCH0.CFM[1:0] value is 01b, set the data length of the message to be transmitted from the transmit/receive FIFO buffer.
If 9-byte or more data length is set, 8 bytes of data is actually transmitted.

Downloaded from Arrow.com.

36.2.53 Transmit/Receive FIFO Access Register 0CL (CFDF00)

Address(es): RSCAN0.CFDF00 000A 85E8h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CFDB1[7:0]								CFDB0[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	CFDB0[7:0]	Transmit/Receive FIFO Buffer Data Byte 0	When CFCCH0.CFM[1:0] value is 01b (transmit mode): Set the transmit/receive FIFO buffer data. When CFCCH0.CFM[1:0] value is 00b (receive mode): The message data stored in the transmit/receive FIFO buffer can be read.	R/W
b15 to b8	CFDB1[7:0]	Transmit/Receive FIFO Buffer Data Byte 1		R/W

Modify this register only when the CFCCH0.CFM[1:0] value is 01b.
This register is readable only when the CFCCH0.CFM[1:0] value is 00b. When the CFPTR0.CFDLC[3:0] value is smaller than 1000b, data bytes for which no data is set are read as 00h.
This register can be read/written when the GRWCR.RPAGE bit is 1.

36.2.54 Transmit/Receive FIFO Access Register 0CH (CFDF10)

Address(es): RSCAN0.CFDF10 000A 85EAh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CFDB3[7:0]								CFDB2[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	CFDB2[7:0]	Transmit/Receive FIFO Buffer Data Byte 2	When CFCCH0.CFM[1:0] value is 01b (transmit mode): Set the transmit/receive FIFO buffer data. When CFCCH0.CFM[1:0] value is 00b (receive mode): The message data stored in the transmit/receive FIFO buffer can be read.	R/W
b15 to b8	CFDB3[7:0]	Transmit/Receive FIFO Buffer Data Byte 3		R/W

Modify this register only when the CFCCH0.CFM[1:0] value is 01b.
This register is readable only when the CFCCH0.CFM[1:0] value is 00b. When the CFPTR0.CFDLC[3:0] value is smaller than 1000b, data bytes for which no data is set are read as 00h.
This register can be read/written when the GRWCR.RPAGE bit is 1.

Downloaded from Arrow.com.

36.2.55 Transmit/Receive FIFO Access Register 0DL (CFDF20)

Address(es): RSCAN0.CFDF20 000A 85ECh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CFDB5[7:0]								CFDB4[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	CFDB4[7:0]	Transmit/Receive FIFO Buffer Data Byte 4	When CFCCH0.CFM[1:0] value is 01b (transmit mode): Set the transmit/receive FIFO buffer data. When CFCCH0.CFM[1:0] value is 00b (receive mode): The message data stored in the transmit/receive FIFO buffer can be read.	R/W
b15 to b8	CFDB5[7:0]	Transmit/Receive FIFO Buffer Data Byte 5		R/W

Modify this register only when the CFCCH0.CFM[1:0] value is 01b.
This register is readable only when the CFCCH0.CFM[1:0] value is 00b. When the CFPTR0.CFDLC[3:0] value is smaller than 1000b, data bytes for which no data is set are read as 00h.
This register can be read/written when the GRWCR.RPAGE bit is 1.

36.2.56 Transmit/Receive FIFO Access Register 0DH (CFDF30)

Address(es): RSCAN0.CFDF30 000A 85EEh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CFDB7[7:0]								CFDB6[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	CFDB6[7:0]	Transmit/Receive FIFO Buffer Data Byte 6	When CFCCH0.CFM[1:0] value is 01b (transmit mode): Set the transmit/receive FIFO buffer data. When CFCCH0.CFM[1:0] value is 00b (receive mode): The message data stored in the transmit/receive FIFO buffer can be read.	R/W
b15 to b8	CFDB7[7:0]	Transmit/Receive FIFO Buffer Data Byte 7		R/W

Modify this register only when the CFCCH0.CFM[1:0] value is 01b.
This register is readable only when the CFCCH0.CFM[1:0] value is 00b. When the CFPTR0.CFDLC[3:0] value is smaller than 1000b, data bytes for which no data is set are read as 00h.
This register can be read/written when the GRWCR.RPAGE bit is 1.

Downloaded from Arrow.com.

36.2.57 Receive FIFO Message Lost Status Register (RFMSTS)

Address(es): RSCAN.RFMSTS 000A 8360h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	RF1MLT	RF0MLT
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RF0MLT	Receive FIFO Buffer 0 Message Lost Status Flag	0: No receive FIFO buffer m message is lost (m = 0, 1). 1: A receive FIFO buffer m message is lost.	R
b1	RF1MLT	Receive FIFO Buffer 1 Message Lost Status Flag		R
b7 to b2	—	Reserved	These bits are read as 0.	R

The RFMSTS register is set to 00h in global reset mode.

RFmMLT Flag (Receive FIFO Buffer m Message Lost Status Flag)

The RFmMLT flag becomes 1 when the RFSTSm.RFMLT flag becomes 1 (a receive FIFO message is lost). When the RFSTSm.RFMLT flag is set to 0, the RFmMLT flag becomes 0.

36.2.58 Transmit/Receive FIFO Message Lost Status Register (CFMSTS)

Address(es): RSCAN0.CFMSTS 000A 8361h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	CF0MLT
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CF0MLT	RSCAN0 Transmit/Receive FIFO Buffer 0 Message Lost Status Flag	0: No RSCAN0 transmit/receive FIFO buffer 0 message is lost. 1: An RSCAN0 transmit/receive FIFO buffer 0 message is lost.	R
b7 to b1	—	Reserved	These bits are read as 0.	R

The CFMSTS register is set to 00h in global reset mode.

CF0MLT Flag (RSCAN0 Transmit/Receive FIFO Buffer 0 Message Lost Status Flag)

The CF0MLT flag becomes 1 when the CFSTS0.CFMLT flag becomes 1 (a transmit/receive FIFO message is lost). When the CFSTS0.CFMLT flag is set to 0, the CF0MLT flag becomes 0.

Downloaded from Arrow.com.

36.2.59 Receive FIFO Interrupt Status Register (RFISTS)

Address(es): RSCAN.RFISTS 000A 8362h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	RF1IF	RF0IF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RF0IF	Receive FIFO Buffer 0 Interrupt Request Status Flag	0: No receive FIFO buffer m interrupt request is present (m = 0, 1). 1: A receive FIFO buffer m interrupt request is present.	R
b1	RF1IF	Receive FIFO Buffer 1 Interrupt Request Status Flag		R
b7 to b2	—	Reserved	These bits are read as 0.	R

The RFISTS register is set to 00h in global reset mode.

RFmIF Flag (Receive FIFO Buffer m Interrupt Request Status Flag)

The RFmIF flag becomes 1 when the RFSTSm.RFIF flag becomes 1 (a receive FIFO interrupt request is present). When the RFSTSm.RFIF flag is set to 0, the RFmIF flag becomes 0.

36.2.60 Transmit/Receive FIFO Receive Interrupt Status Register (CFISTS)

Address(es): RSCAN.CFISTS 000A 8363h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	CF0IF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CF0IF	RSCAN0 Transmit/Receive FIFO Buffer 0 Receive Interrupt Request Status Flag	0: No RSCAN0 transmit/receive FIFO buffer 0 receive interrupt request is present. 1: An RSCAN0 transmit/receive FIFO buffer 0 receive interrupt request is present.	R
b7 to b1	—	Reserved	These bits are read as 0.	R

The CFISTS register is set to 00h in global reset mode.

CF0IF Flag (RSCAN0 Transmit/Receive FIFO Buffer 0 Receive Interrupt Request Status Flag)

The CF0IF flag becomes 1 when the CFSTS0.CFRXIF flag becomes 1 (a transmit/receive FIFO receive interrupt request is present). When the CFSTS0.CFRXIF flag is set to 0, the CF0IF flag becomes 0.

Downloaded from Arrow.com.

36.2.61 Transmit Buffer Control Register p (TMCp) (p = 0 to 3)

Address(es): RSCAN0.TMC0 000A 8364h, RSCAN0.TMC1 000A 8365h, RSCAN0.TMC2 000A 8366h, RSCAN0.TMC3 000A 8367h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	TMOM	TMTAR	TMTR
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TMTR	Transmit Request	0: Transmission is not requested. 1: Transmission is requested.	R/(W) *1
b1	TMTAR	Transmit Abort Request	0: Transmit abort is not requested. 1: Transmit abort is requested.	R/(W) *1
b2	TMOM	One-Shot Transmission Enable	0: One-shot transmission is disabled. 1: One-shot transmission is enabled.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Only 1 can be written to this bit. Writing 0 does not affect the bit value.

When the TMCp register meets the following condition, set it to 00h.

- The TMCp register corresponds to the transmit buffer number selected by the CFCCH0.CFTML[1:0] bits.

Bits in the TMCp register become all 0s in channel reset mode. Modify the TMCp register (p = 0 to 3) only in channel communication mode or channel halt mode.

TMTR Bit (Transmit Request)

Setting this bit to 1 transmits the message stored in the transmit buffer.
The TMTR bit becomes 0 when any of the following conditions is met, but does not become 0 by writing 0 by the program.

- Transmission has been completed.
- Transmit abort has been completed by setting the TMTAR bit to 1.
- An error or arbitration lost has been detected with the TMOM bit set to 1.

Set the TMTR bit to 1 when the TMSTSp.TMTRF[1:0] value is 00b.

TMTAR Bit (Transmit Abort Request)

Setting this bit to 1 generates a transmit abort request for the message stored in the transmit buffer. However, a message that is being transmitted or to be transmitted next cannot be aborted.
When the TMTR bit becomes 1, the TMTAR bit can be set to 1.
The TMTAR bit is set to 0 when any of the following conditions is met, but does not become 0 by writing 0 by the program.

- Transmission has been completed.
- Transmit abort has been completed.
- An error or arbitration lost has been detected.

If this bit becomes 0 at the timing when the program writes 1 to this bit, this bit becomes 0.

TMOM Bit (One-Shot Transmission Enable)

Setting this bit to 1 enables one-shot transmission. When transmission fails, retransmission defined in the CAN protocol is not performed.

Downloaded from Arrow.com.

Modify the TMOM bit when the TMSTSp.TMTRM flag is 0. To set the TMOM bit to 1, also set the TMTR bit together.

36.2.62 Transmit Buffer Status Register p (TMSTSp) (p = 0 to 3)

Address(es): RSCAN0.TMSTS0 000A 836Ch, RSCAN0.TMSTS1 000A 836Dh, RSCAN0.TMSTS2 000A 836Eh, RSCAN0.TMSTS3 000A 836Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	TMTARM	TMTRM	TMTRF[1:0]		TMTSTS
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TMTSTS	Transmit Buffer Transmit Status Flag	0: Transmission is not in progress. 1: Transmission is in progress.	R
b2, b1	TMTRF[1:0]	Transmit Buffer Transmit Result Flag	b2 b1 0 0: Transmission is in progress or no transmit request is present. 0 1: Transmit abort has been completed. 1 0: Transmission has been completed (without transmit abort request). 1 1: Transmission has been completed (with transmit abort request).	R/W
b3	TMTRM	Transmit Buffer Transmit Request Status Flag	0: No transmit request is present. 1: A transmit request is present.	R
b4	TMTARM	Transmit Buffer Transmit Abort Request Status Flag	0: No transmit abort request is present. 1: A transmit abort request is present.	R
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R

The TMSTSp register becomes all 0s in channel reset mode.

TMTSTS Flag (Transmit Buffer Transmit Status Flag)

This flag becomes 1 when transmission from the transmit buffer starts, and becomes 0 when transmission from the transmit buffer has been completed or terminated due to a bus error or arbitration lost.

TMTRF[1:0] Flags (Transmit Buffer Transmit Result Flag)

These flags indicate the result of transmission from the transmit buffer.
00b: Transmission is in progress or no transmit request is present.
01b: Transmission from the transmit buffer was aborted.
10b: Transmission has been completed with the TMCp.TMTAR bit set to 0 (transmit abort is not requested).
11b: Transmission has been completed with the TMCp.TMTAR bit set to 1 (transmit abort is requested).
Write 00b to the TMTRF[1:0] flags in channel communication mode or channel halt mode. Do not write any value other than 00b to these flags.

TMTRM Flag (Transmit Buffer Transmit Request Status Flag)

The TMTRM flag becomes 1 when the TMCp.TMTR bit is set to 1, and becomes 0 when the TMCp.TMTR bit is set to 0.

TMTARM Flag (Transmit Buffer Transmit Abort Request Status Flag)

The TMTARM flag becomes 1 when the TMCp.TMTAR bit is set to 1, and becomes 0 when the TMCp.TMTAR bit is set to 0.

Downloaded from Arrow.com.

36.2.63 Transmit Buffer Transmit Request Status Register (TMTRSTS)

Address(es): RSCAN0.TMTRSTS 000A 8374h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	TMTRSTS3	TMTRSTS2	TMTRSTS1	TMTRSTS0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TMTRSTS0	RSCAN0 Transmit Buffer 0 Transmit Request Status Flag	0: No transmit request is present. 1: A transmit request is present.	R
b1	TMTRSTS1	RSCAN0 Transmit Buffer 1 Transmit Request Status Flag		R
b2	TMTRSTS2	RSCAN0 Transmit Buffer 2 Transmit Request Status Flag		R
b3	TMTRSTS3	RSCAN0 Transmit Buffer 3 Transmit Request Status Flag		R
b15 to b4	—	Reserved	These bits are read as 0.	R

TMTRSTSp Flag (RSCAN0 Transmit Buffer p Transmit Request Status Flag) (p = 0 to 3)

This flag indicates the status of the TMCp.TMTR bit.

When the TMTR bit is set to 1 (transmission is requested), the corresponding TMTRSTSp flag becomes 1.

The corresponding TMTRSTSp flag becomes 0 when the TMTR bit is set to 0 (transmission is not requested) or in channel reset mode.

Downloaded from Arrow.com.

36.2.64 Transmit Buffer Transmit Complete Status Register (TMTCSTS)

Address(es): RSCAN0.TMTCSTS 000A 8376h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	TMTCSTS3	TMTCSTS2	TMTCSTS1	TMTCSTS0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TMTCSTS0	RSCAN0 Transmit Buffer 0 Transmit Complete Status Flag	0: Transmission has not been completed. 1: Transmission has been completed.	R
b1	TMTCSTS1	RSCAN0 Transmit Buffer 1 Transmit Complete Status Flag		R
b2	TMTCSTS2	RSCAN0 Transmit Buffer 2 Transmit Complete Status Flag		R
b3	TMTCSTS3	RSCAN0 Transmit Buffer 3 Transmit Complete Status Flag		R
b15 to b4	—	Reserved	These bits are read as 0.	R

TMTCSTSp Flag (RSCAN0 Transmit Buffer p Transmit Complete Status Flag) (p = 0 to 3)

When the TMSTSp.TMTRF[1:0] flags become 10b (transmission has been completed (without transmit abort request)) or 11b (transmission has been completed (with transmit abort request)), the corresponding TMTCSTSp flag becomes 1. This flag becomes 0 when the corresponding TMSTSp.TMTRF[1:0] flags are set to 00b or in channel reset mode.

Downloaded from Arrow.com.

36.2.65 Transmit Buffer Transmit Abort Status Register (TMTASTS)

Address(es): RSCAN0.TMTASTS 000A 8378h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	TMTASTS3	TMTASTS2	TMTASTS1	TMTASTS0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TMTASTS0	RSCAN0 Transmit Buffer 0 Transmit Abort Status Flag	0: Transmission is not aborted. 1: Transmission is aborted.	R
b1	TMTASTS1	RSCAN0 Transmit Buffer 1 Transmit Abort Status Flag		R
b2	TMTASTS2	RSCAN0 Transmit Buffer 2 Transmit Abort Status Flag		R
b3	TMTASTS3	RSCAN0 Transmit Buffer 3 Transmit Abort Status Flag		R
b15 to b4	—	Reserved	These bits are read as 0.	R

TMTASTSp Flag (RSCAN0 Transmit Buffer p Transmit Abort Status Flag) (p = 0 to 3)

When the TMSTSp.TMTRF[1:0] flags become 01b (transmit abort has been completed), the corresponding TMTASTSp flag becomes 1.

This flag becomes 0 when the corresponding TMSTSp.TMTRF[1:0] flags are set to 00b or in channel reset mode.

Downloaded from Arrow.com.

36.2.66 Transmit Buffer Interrupt Enable Register (TMIEC)

Address(es): RSCAN0.TMIEC 000A 837Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	TMIE3	TMIE2	TMIE1	TMIE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TMIE0	RSCAN0 Transmit Buffer 0 Interrupt Enable	0: Transmit buffer interrupt is disabled. 1: Transmit buffer interrupt is enabled.	R/W
b1	TMIE1	RSCAN0 Transmit Buffer 1 Interrupt Enable		R/W
b2	TMIE2	RSCAN0 Transmit Buffer 2 Interrupt Enable		R/W
b3	TMIE3	RSCAN0 Transmit Buffer 3 Interrupt Enable		R/W
b15 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

TMIEp Bit (RSCAN0 Transmit Buffer p Interrupt Enable) (p = 0 to 3)

When TMIEp bit is set to 1 and the corresponding transmission has been completed, a transmit buffer interrupt request is generated.

Modify this bit when the corresponding TMSTSp.TMTRM flag is 0 (no transmit request is present).

Write 0 to bits corresponding to transmit buffers linked to transmit/receive FIFO buffers.

36.2.67 Transmit Buffer Register pAL (TMIDLp) (p = 0 to 3)

Address(es): RSCAN0.TMIDL0 000A 8600h, RSCAN0.TMIDL1 000A 8610h, RSCAN0.TMIDL2 000A 8620h, RSCAN0.TMIDL3 000A 8630h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TMID[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	TMID[15:0]	Transmit Buffer ID Data L	Set standard ID or extended ID. For standard ID, write an ID to b10 to b0 and write 0 to b15 to b11.	R/W

Modify this register when the corresponding TMSTSp.TMTRM flag is 0 (no transmit request is present). If this register is linked to any transmit/receive FIFO buffer, do not write data to this register.

This register can be read/written when the GRWCR.RPAGE bit is 1.

TMID[15:0] Bits (Transmit Buffer ID Data L)

These bits are used to set the ID of the message to be transmitted from the transmit buffer.

Downloaded from Arrow.com.

36.2.68 Transmit Buffer Register pAH (TMIDHp) (p = 0 to 3)

Address(es): RSCAN0.TMIDH0 000A 8602h, RSCAN0.TMIDH1 000A 8612h, RSCAN0.TMIDH2 000A 8622h, RSCAN0.TMIDH3 000A 8632h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TMIDE	TMRTR	THLEN	TMID[28:16]												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b12 to b0	TMID[28:16]	Transmit Buffer ID Data H	Set standard ID or extended ID. For standard ID, write 0 to these bits.	R/W
b13	THLEN	Transmit History Data Store Enable	0: Transmit history data is not stored in the buffer. 1: Transmit history data is stored in the buffer.	R/W
b14	TMRTR	Transmit Buffer RTR	0: Data frame 1: Remote frame	R/W
b15	TMIDE	Transmit Buffer IDE	0: Standard ID 1: Extended ID	R/W

Modify this register when the corresponding TMSTSp.TMTRM flag is 0 (no transmit request is present). If this register is linked to any transmit/receive FIFO buffer, do not write data to this register.
This register can be read/written when the GRWCR.RPAGE bit is 1.

TMID[28:16] Bits (Transmit Buffer ID Data H)

These bits are used to set the ID of the message to be transmitted from the transmit buffer.

THLEN Bit (Transmit History Data Store Enable)

When this bit is set to 1, the transmit history data (label information, buffer number, and buffer type) of transmit messages is stored in the transmit history buffer after transmission is completed.

TMRTR Bit (Transmit Buffer RTR)

This bit is used to set the data format of the message to be transmitted from the transmit buffer.

TMIDE Bit (Transmit Buffer IDE)

This bit is used to set the ID format of the message to be transmitted from the transmit buffer.

Downloaded from Arrow.com.

36.2.69 Transmit Buffer Register pBH (TMPTRp) (p = 0 to 3)

Address(es): RSCAN0.TMPTR0 000A 8606h, RSCAN0.TMPTR1 000A 8616h, RSCAN0.TMPTR2 000A 8626h, RSCAN0.TMPTR3 000A 8636h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TMDLC[3:0]				—	—	—	—	TMPTR[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	TMPTR[7:0]	Transmit Buffer Label Data	Set the label information to be stored in the transmit history buffer.	R/W
b11 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b12	TMDLC[3:0]	Transmit Buffer DLC Data	b15 b12 0 0 0 0: 0 data bytes 0 0 0 1: 1 data byte 0 0 1 0: 2 data bytes 0 0 1 1: 3 data bytes 0 1 0 0: 4 data bytes 0 1 0 1: 5 data bytes 0 1 1 0: 6 data bytes 0 1 1 1: 7 data bytes 1 x x x: 8 data bytes	R/W

x: Don’t care

Modify this register when the corresponding TMSTSp.TMTRM flag is 0 (no transmit request is present). If this register is linked to any transmit/receive FIFO buffer, do not write data to this register.
This register can be read/written when the GRWCR.RPAGE bit is 1.

TMPTR[7:0] Bits (Transmit Buffer Label Data)

When message transmission has been completed, the TMPTR[7:0] value is stored in the transmit history buffer.

TMDLC[3:0] Bits (Transmit Buffer DLC Data)

These bits are used to set the data length of the message to be transmitted from the transmit buffer when the TMIDHp.TMRTR bit is set to 0 (data frame). If a 9-byte (or more) data length is set, 8 bytes of data is actually transmitted.
When the TMIDHp.TMRTR bit is set to 1 (remote frame), set the data length of messages to be requested.

Downloaded from Arrow.com.

36.2.70 Transmit Buffer Register pCL (TMDF0p) (p = 0 to 3)

Address(es): RSCAN0.TMDF00 000A 8608h, RSCAN0.TMDF01 000A 8618h, RSCAN0.TMDF02 000A 8628h, RSCAN0.TMDF03 000A 8638h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TMDB1[7:0]								TMDB0[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	TMDB0[7:0]	Transmit Buffer Data Byte 0	Set transmit buffer data.	R/W
b15 to b8	TMDB1[7:0]	Transmit Buffer Data Byte 1		R/W

Modify this register when the corresponding TMSTSp.TMTRM flag is 0 (no transmit request is present). If this register is linked to any transmit/receive FIFO buffer, do not write data to this register.
This register can be read/written when the GRWCR.RPAGE bit is 1.

36.2.71 Transmit Buffer Register pCH (TMDF1p) (p = 0 to 3)

Address(es): RSCAN0.TMDF10 000A 860Ah, RSCAN0.TMDF11 000A 861Ah, RSCAN0.TMDF12 000A 862Ah, RSCAN0.TMDF13 000A 863Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TMDB3[7:0]								TMDB2[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	TMDB2[7:0]	Transmit Buffer Data Byte 2	Set transmit buffer data.	R/W
b15 to b8	TMDB3[7:0]	Transmit Buffer Data Byte 3		R/W

Modify this register when the corresponding TMSTSp.TMTRM flag is 0 (no transmit request is present). If this register is linked to any transmit/receive FIFO buffer, do not write data to this register.
This register can be read/written when the GRWCR.RPAGE bit is 1.

Downloaded from Arrow.com.

36.2.72 Transmit Buffer Register pDL (TMDF2p) (p = 0 to 3)

Address(es): RSCAN0.TMDF20 000A 860Ch, RSCAN0.TMDF21 000A 861Ch, RSCAN0.TMDF22 000A 862Ch, RSCAN0.TMDF23 000A 863Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TMDB5[7:0]								TMDB4[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	TMDB4[7:0]	Transmit Buffer Data Byte 4	Set transmit buffer data.	R/W
b15 to b8	TMDB5[7:0]	Transmit Buffer Data Byte 5		R/W

Modify this register when the corresponding TMSTSp.TMTRM flag is 0 (no transmit request is present). If this register is linked to any transmit/receive FIFO buffer, do not write data to this register.
This register can be read/written when the GRWCR.RPAGE bit is 1.

36.2.73 Transmit Buffer Register pDH (TMDF3p) (p = 0 to 3)

Address(es): RSCAN0.TMDF30 000A 860Eh, RSCAN0.TMDF31 000A 861Eh, RSCAN0.TMDF32 000A 862Eh, RSCAN0.TMDF33 000A 863Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TMDB7[7:0]								TMDB6[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	TMDB6[7:0]	Transmit Buffer Data Byte 6	Set transmit buffer data.	R/W
b15 to b8	TMDB7[7:0]	Transmit Buffer Data Byte 7		R/W

Modify this register when the corresponding TMSTSp.TMTRM flag is 0 (no transmit request is present). If this register is linked to any transmit/receive FIFO buffer, do not write data to this register.
This register can be read/written when the GRWCR.RPAGE bit is 1.

Downloaded from Arrow.com.

36.2.74 Transmit History Buffer Control Register (THLCC0)

Address(es): RSCAN0.THLCC0 000A 837Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	THLDTE	THLIM	THLIE	—	—	—	—	—	—	—	THLE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	THLE	Transmit History Buffer Enable	0: Transmit history buffer is not used. 1: Transmit history buffer is used.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	THLIE	Transmit History Interrupt Enable	0: Transmit history interrupt is disabled. 1: Transmit history interrupt is enabled.	R/W
b9	THLIM	Transmit History Interrupt Source Select	0: When 6 sets of data have been stored in the transmit history buffer 1: When a single set of transmit history data has been stored	R/W
b10	THLDTE	Transmit History Target Buffer Select	0: Entry from transmit/receive FIFO buffers 1: Entry from transmit buffers, transmit/receive FIFO buffers	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

THLE Bit (Transmit History Buffer Enable)

Setting this bit to 1 makes the transmit history buffer available. When data transmission from the buffer selected by the THLDTE bit has been completed, the transmit history data of transmit messages is stored in the transmit history buffer. Modify this bit only in channel communication mode or channel halt mode.

THLIE Bit (Transmit History Interrupt Enable)

When the THLIE bit is set to 1 and the source selected by the THLIM bit has occurred, a transmit history interrupt request is generated. Modify the THLIE bit with the THLE bit set to 0.

THLIM Bit (Transmit History Interrupt Source Select)

This bit is used to select a transmit history interrupt source.
Modify this bit only in channel reset mode.

THLDTE Bit (Transmit History Target Buffer Select)

When this bit is set to 0, the transmit history data of messages transmitted from transmit/receive FIFO buffers is stored in the transmit history buffer. When this bit is set to 1, the transmit history data of messages transmitted from transmit buffers and transmit/receive FIFO buffers is stored in the transmit history buffer.
Modify this bit only in channel reset mode.

Downloaded from Arrow.com.

36.2.75 Transmit History Buffer Status Register (THLSTS0)

Address(es): RSCAN0.THLSTS0 000A 8380h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	THLMC[3:0]				—	—	—	—	THLIF	THLELT	THLFLL	THLEMP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	THLEMP	Transmit History Buffer Empty Status Flag	0: Transmit history buffer contains unread data. 1: Transmit history buffer contains no unread data (buffer empty).	R
b1	THLFLL	Transmit History Buffer Full Status Flag	0: Transmit history buffer is not full. 1: Transmit history buffer is full.	R
b2	THLELT	Transmit History Buffer Overflow Flag	0: Transmit history buffer overflow has not occurred. 1: Transmit history buffer overflow has occurred.	R/(W) *1
b3	THLIF	Transmit History Interrupt Request Flag	0: No transmit history interrupt request is present. 1: A transmit history interrupt request is present.	R/(W) *1
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11 to b8	THLMC[3:0]	Transmit History Buffer Unread Data Counter	These bits indicate the number of unread data sets stored in the transmit history buffer.	R
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Only 0 can be written to this bit, to clear the flag. Writing 1 does not affect the flag value.

THLEMP Flag (Transmit History Buffer Empty Status Flag)

The THLEMP flag becomes 0 when even a single set of transmit history data has been stored in the transmit history buffer.

This flag becomes 1 when all the data in the transmit history buffer has been read. This flag also becomes 1 in channel reset mode or when the THLCC0.THLE bit is set to 0 (transmit history buffer is not used).

THLFLL Flag (Transmit History Buffer Full Status Flag)

The THLFLL flag becomes 1 when 8 data sets have been stored in the transmit history buffer, and becomes 0 when the number of data sets stored in the transmit history buffer has decreased to less than 8.

This flag also becomes 0 in channel reset mode or when the THLCC0.THLE bit is set to 0 (transmit history buffer is not used).

THLELT Flag (Transmit History Buffer Overflow Flag)

The THLELT flag becomes 1 when it is attempted to store new transmit history data while the transmit history buffer is full. In this case, the new data is discarded.

This flag becomes 0 in channel reset mode or by writing 0 to this flag by the program.

THLIF Flag (Transmit History Interrupt Request Flag)

The THLIF flag becomes 1 when the interrupt source set by the THLCC0.THLIM bit has occurred.

This flag becomes 0 in channel reset mode or by writing 0 to this flag by the program.

THLMC[3:0] Flags (Transmit History Buffer Unread Data Counter)

These flags indicate the number of unread data sets stored in the transmit history buffer.

Downloaded from Arrow.com.

36.2.76 Transmit History Buffer Access Register (THLACC0)

Address(es): RSCAN0.THLACC0 000A 8680h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TID[7:0]								—	—	—	BN[1:0]		—	BT[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	BT[1:0]	Buffer Type Data	b1 b0 0 1: Transmit buffer 1 0: Transmit FIFO buffer	R
b2	—	Reserved	This bit is read as 0.	R
b4, b3	BN[1:0]	Buffer Number Data	The buffer number of transmit source (transmit buffer or transmit/receive FIFO) can be read.	R
b7 to b5	—	Reserved	These bits are read as 0.	R
b15 to b8	TID[7:0]	Label Data	The label information of stored data can be read.	R

This register can be read when the GRWCR.RPAGE bit is 1.

BT[1:0] Bits (Buffer Type Data)

These bits indicate the transmit source buffer type of transmit history data stored in the transmit history buffer.

BN[1:0] Bits (Buffer Number Data)

These bits indicate the transmit source buffer number of transmit history data stored in the transmit history buffer.

TID[7:0] Bits (Label Data)

These bits indicate the label information of transmit history data stored in the transmit history buffer.

Downloaded from Arrow.com.

36.2.77 Transmit History Buffer Pointer Control Register (THLPCTR0)

Address(es): RSCAN0.THLPCTR0 000A 8384h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	THLPC[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	THLPC[7:0]	Transmit History Buffer Pointer	Writing FFh to these bits moves the read pointer to the next unread data in the transmit history buffer.	W
b15 to b8	—	Reserved	The write value should be 0.	W

THLPC[7:0] Bits (Transmit History Buffer Pointer)

When the THLPC [7:0] bits are set to FFh, the read pointer moves to the next data in the transmit history buffer. At this time, the THLSTS0.THLMC[3:0] (transmit history buffer unread data counter) value is decremented. After reading the THLACC0 register, write FFh to the THLPC [7:0] bits.

Write FFh to the THLPC[7:0] bits when the THLCC0.THLE bit is set to 1 (transmit history buffer is used) and the THLSTS0.THLEMP flag is 0.

Downloaded from Arrow.com.

36.2.78 Global RAM Window Control Register (GRWCR)

Address(es): RSCAN.GRWCR 000A 838Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RPAGE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RPAGE	RAM Window Select	0: Selects window 0 (receive rule entry registers, RAM test registers) 1: Selects window 1 (receive buffer, receive FIFO buffer, transmit/receive FIFO buffer, transmit buffer, transmit history data access register)	R/W
b15 to b1	—	Reserved	The write value should be 0.	R/W

RPAGE Bit (RAM Window Select)

This bit is used to select a window for the switching of registers that are allocated to addresses from 000A 83A0h to 000A 8681h.

[Registers allocated when the RPAGE bit is set to 0 (window 0 selected)]

- Receive rule entry registers: GAFLIDLj, GAFLIDHj, GAFLMLj, GAFLMHj, GAFLPLj, GAFLPHj (j = 0 to 15)
- RAM test registers: RPGACCr (r = 0 to 127)

[Registers allocated when the RPAGE bit is set to 1 (window 1 selected)]

- Receive buffer registers: RMIDLn, RMIDHn, RMTSn, RMPTRn, RMDF0n to RMDF3n (n = 0 to 15)
- Receive FIFO access registers: RFIDLm, RFIDHm, RFTSm, RFPTRm, RFDF0m to RFDF3m (m = 0, 1)
- Transmit/receive FIFO access registers: CFIDL0, CFIDH0, CFTS0, CFPTR0, CFDF00 to CFDF30
- Transmit buffer registers: TMIDLp, TMIDHp, TMPTRp, TMDF0p to TMDF3p (p = 0 to 3)
- Transmit history buffer access register: THLACC0

Downloaded from Arrow.com.

36.2.79 Global Test Configuration Register (GTSTCFG)

Address(es): RSCAN.GTSTCFG 000A 838Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	RTMPS[2:0]			—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b10 to b8	RTMPS[2:0]	RAM Test Page Configuration	Set a value within a range of page 0 (00h) to page 2 (02h).	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Modify the GTSTCFG register only in global test mode.

RTMPS[2:0] Bits (RAM Test Page Configuration)

These bits are used to set the RAM test target page number for RAM test. Set a value from 00h to 02h.

36.2.80 Global Test Control Register (GTSTCTRL)

Address(es): RSCAN.GTSTCTRL 000A 838Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	RTME	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b2	RTME	RAM Test Enable	0: RAM test is disabled. 1: RAM test is enabled.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

RTME Bit (RAM Test Enable)

Setting this bit to 1 enables RAM test. Modify this bit only in global test mode.

(1) Set the GCTRL.GMDC[1:0] bits to 10b (global test mode).
(2) Unlock protection by successively writing 7575h and 8A8Ah to the GLOCKK register
(3) Set the RTME bit to 1.
(4) Check that the RTME bit is set to 1.

Downloaded from Arrow.com.

36.2.81 Global Test Protection Unlock Register (GLOCKK)

Address(es): RSCAN.GLOCKK 000A 8394h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	LOCK[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	LOCK[15:0]	Protection Unlock Data	Write protection unlock data to use test functions. These bits are read as 0000h.	W

Modify the GLOCKK register only in global test mode.

LOCK[15:0] Bits (Protection Unlock Data)

Write the protection unlock data shown in Table 36.3 to the LOCK[15:0] bits in succession to allow writing 1 to the target bit.

Table 36.3 Protection Unlock Data for Test Functions

Test Function	Protection Unlock Data 1	Protection Unlock Data 2	Target Bit
RAM test	7575h	8A8Ah	GTSTCTRL.RTME bit

Writing data to the CAN's register area (000A 8300h to 000A 839Fh) except the RAM area after protection is unlocked enables protection again.
Protection is not enabled even by reading data from the CAN's register area or reading/writing data from/to other areas.

36.2.82 RAM Test Register r (RPGACCr) (r = 0 to 127)

Address(es): RSCAN.RPGACC0 to RSCAN.RPGACC127 000A 8580h to 000A 867Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RDTA[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Description	R/W
Data can be read and written in CAN RAM.	R/W

Modify the RPGACCr register in global test mode with the GTSTCTRL.RTME bit set to 1 (RAM test is enabled). The RPGACCr register is readable and writable when the GTSTCTRL.RTME bit is set to 1.
This register can be read/written when the GRWCR.RPAGE bit is 0.

Downloaded from Arrow.com.

36.3 CAN Modes

The CAN module has four global modes to control entire CAN module status and four channel modes to control individual channel status.

Details of global modes are described in section 36.3.1, Global Modes, and details of channel modes are described in section 36.3.2, Channel Modes.

- Global stop mode: Stops clocks of entire module to achieve low power consumption.
- Global reset mode: Performs initial settings for entire module.
- Global test mode: Performs test settings and performs RAM test.
- Global operating mode: Makes entire module operable.
- Channel stop mode: Stops channel clock.
- Channel reset mode: Performs initial settings for channels.
- Channel halt mode: Stops CAN communication and enables channel test.
- Channel communication mode: Performs CAN communication.

36.3.1 Global Modes

Figure 36.2 shows the transitions of global modes.

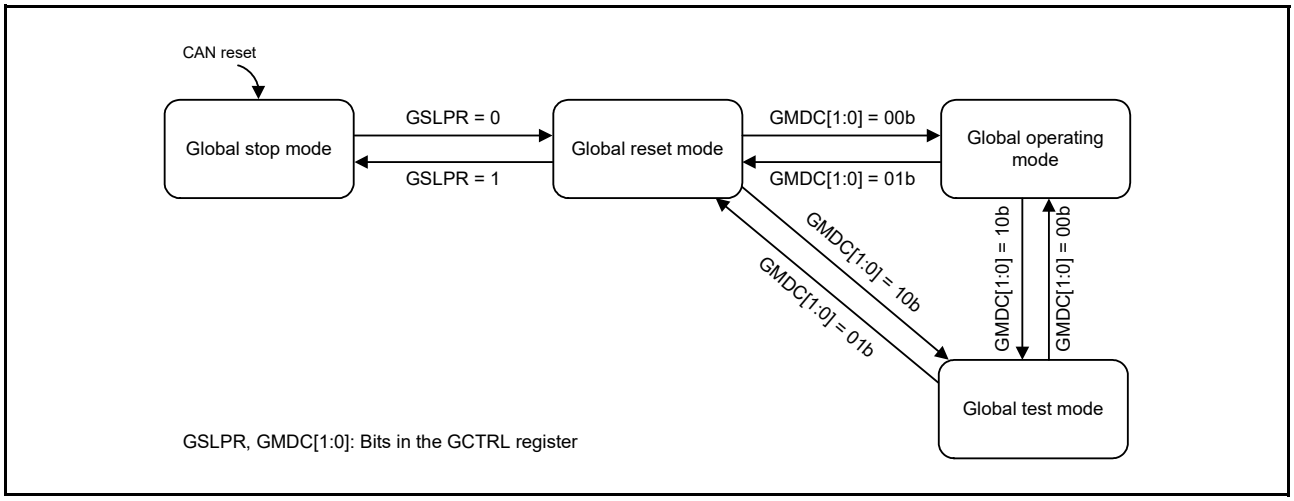


Figure 36.2 Transitions of Global Modes

Channel modes transition in some cases with transitions of global modes. Table 36.4 shows the transitions of channel modes depending on the global mode setting by the GCTRL.GMDC[1:0] bits and the GSLPR bit.

Table 36.4 Transitions of Channel Modes Depending on Global Mode Setting (GCTRL.GMDC[1:0] and GSLPR Bits)

	Channel Mode after Setting			
Channel Mode before Setting	**GMDC[1:0] = 00b GSLPR = 0 (Global Operation)**	**GMDC[1:0] = 10b GSLPR = 0 (Global Test)**	**GMDC[1:0] = 01b GSLPR = 0 (Global Reset)**	**GMDC[1:0] = 01b GSLPR = 1 (Global Stop)**
Channel communication	Channel communication	Channel communication	Channel reset	Transition prohibited
Channel halt	Channel halt	Channel halt	Channel reset	Transition prohibited
Channel reset	Channel reset	Channel reset	Channel reset	Channel stop
Channel stop	Channel stop	Channel stop	Channel stop	Channel stop

Downloaded from Arrow.com.

Table 36.5 shows the global mode transition time.

Table 36.5 Global Mode Transition Time

Mode before Transition	Mode after Transition	Maximum Transition Time
Global stop	Global reset	3 PCLK cycles
Global reset	Global stop	3 PCLK cycles
Global reset	Global test	10 PCLK cycles
Global reset	Global operating	10 PCLK cycles
Global test	Global reset	3 PCLK cycles
Global test	Global operating	3 PCLK cycles
Global operating	Global reset	3 PCLK cycles
Global operating	Global test	Two CAN frames

(1) Global Stop Mode

In global stop mode, clocks of the CAN do not run and therefore power consumption is reduced. CAN registers can be read, but writing data to them is prohibited. Register values are retained.
After the operation of the CAN module is enabled, the CAN module transitions to global stop mode. Setting the GCTRL.GSLPR bit to 1 (global stop mode) in global reset mode sets the CTRL.CSLPR bit to 1 (channel stop mode). If all channels are forcibly caused to transition to channel stop mode, the CAN module transitions to global stop mode. The GCTRL.GSLPR bit should not be modified in global operating mode and global test mode.

(2) Global Reset Mode

In global reset mode, CAN module settings are performed. When the CAN module transitions to global reset mode, some registers are initialized. Table 36.8 and Table 36.9 list the registers to be initialized.
Setting the GCTRL.GMDC[1:0] bits to 01b sets each of the CTRL.CHMDC[1:0] bits to 01b (channel reset mode). If all channels are forcibly caused to transition to channel reset mode, the CAN module transitions to global reset mode. Channels that are already in channel reset mode or channel stop mode do not transition (because the CTRL.CHMDC[1:0] bits have already been set to 01b).

(3) Global Test Mode

In global test mode, settings for test-related registers are performed. When the CAN module transitions to global test mode, all CAN communications are disabled.
Setting the GCTRL.GMDC[1:0] bits to 10b sets each of the CTRL.CHMDC[1:0] bits to 10b (channel halt mode). If all channels are forcibly caused to transition to channel halt mode, the CAN module transitions to global test mode. Channels that are in channel stop mode, channel reset mode, or channel halt mode do not transition.

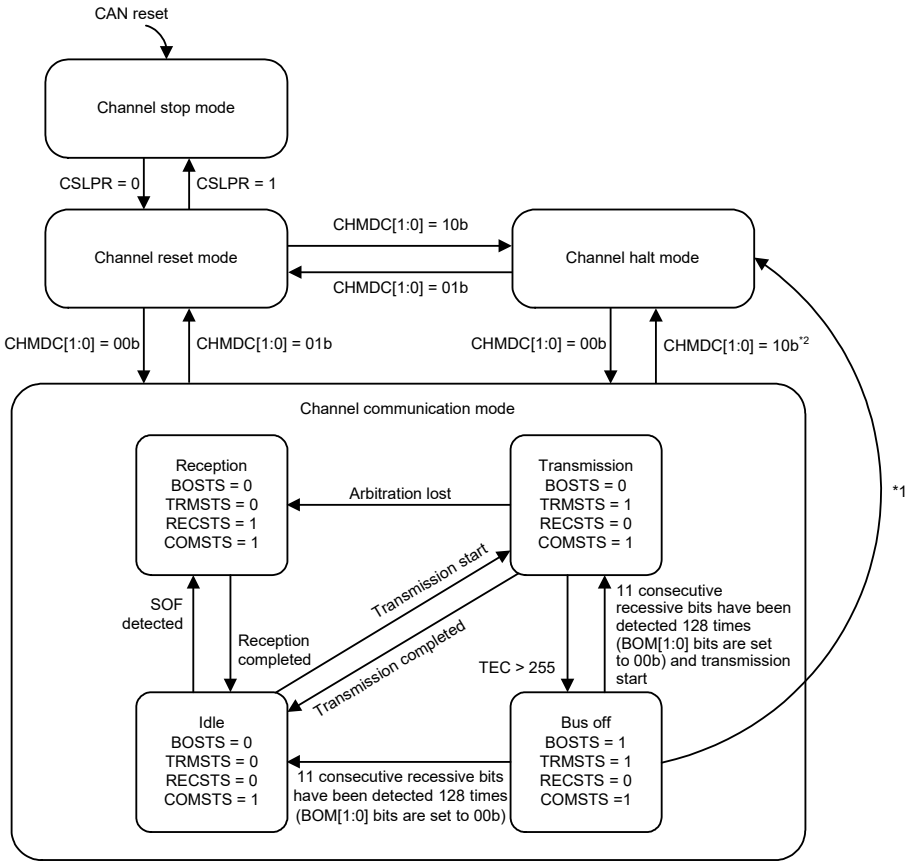
(4) Global Operating Mode

In global operating mode, entire CAN module operates.
When the GCTRL.GMDC[1:0] bits are set to 00b, the CAN module transitions to global operating mode.

Downloaded from Arrow.com.

36.3.2 Channel Modes

Figure 36.3 shows a channel mode state transition chart. Table 36.6 shows the channel mode transition time.



CHMDC[1:0], CSLPR: Bits in the CTRL register
BOM[1:0]: Bits in the CTRH register
BOSTS, TRMSTS, RECSTS, COMSTS: Bits in the STSL register

Note 1. Timing of transition from bus off state to channel halt mode
- When BOM[1:0] = 01b: Transition to channel halt mode when TEC exceeds 255
- When BOM[1:0] = 10b: Transition to channel halt mode when 11 consecutive recessive bits have been detected 128 times
- When BOM[1:0] = 11b: Transition to channel halt mode when the CHMDC[1:0] bits are set to 10b

Note 2. While the CAN bus is locked at the dominant level, transition to channel halt mode is not made. In that case, enter channel reset mode.

Figure 36.3 Channel Mode State Transition Chart

Table 36.6 Channel Mode Transition Time

Mode before Transition	Mode after Transition	Maximum Transition Time
Channel stop	Channel reset	3 PCLK cycles
Channel reset	Channel stop	3 PCLK cycles
Channel reset	Channel halt	3 CAN bit times
Channel reset	Channel communication	2 CAN bit times
Channel halt	Channel reset	3 PCLK cycles
Channel halt	Channel communication	3 CAN bit times
Channel communication	Channel reset	3 PCLK cycles
Channel communication	Channel halt	2 CAN frames

Downloaded from Arrow.com.

(1) Channel Stop Mode

In channel stop mode, clocks are not supplied to channels and therefore power consumption is reduced. CAN registers can be read, but writing data to them is prohibited. Register values are retained.
Each channel enters channel stop mode after the operation of the CAN module is enabled. The channel transitions to channel stop mode when the CTRL.CSLPR bit is set to 1 (channel stop mode) in channel reset mode.
The CSLPR bit should not be modified in channel communication mode and channel halt mode.

(2) Channel Reset Mode

In channel reset mode, channel settings are performed. When a channel transitions to channel reset mode, some channel-related registers are initialized. Table 36.8 lists the registers to be initialized.
When the CTRL.CHMDC[1:0] bits are set to 01b (channel reset mode) during CAN communication, communication is terminated before it is completed and the channel transitions to channel reset mode. Table 36.7 shows the operation when the CTRL.CHMDC[1:0] bits are set to 01b (channel reset mode) during CAN communication.

(3) Channel Halt Mode

In channel halt mode, settings for test-related registers of channels are performed. When a channel transitions to channel halt mode, CAN communication of the channel stops.
Table 36.7 shows operation when the CTRL.CHMDC[1:0] bits are set to 10b (channel halt mode) during CAN communication.

Table 36.7 Operation when a Channel Transitions to Channel Reset Mode/Channel Halt Mode

Mode	During Reception	During Transmission	Bus Off State
Channel reset (CHMDC[1:0] = 01b)	Transitions to channel reset mode before reception is completed.*1	Transitions to channel reset mode before transmission is completed.*1	Transitions to channel reset mode before bus off recovery.
Channel halt*3 (CHMDC[1:0] = 10b)	Transitions to channel halt mode after reception is completed.*2	Transitions to channel halt mode after transmission is completed.*2	[When BOM[1:0] = 00b] Transitions to channel halt mode (CHMDC[1:0] = 10b) only after bus off recovery. [When BOM[1:0] = 01b] Transitions to channel halt mode automatically when the condition for transition to bus off state is met. [When BOM[1:0] = 10b] Transitions to channel halt mode automatically after bus off recovery. [When BOM[1:0] = 11b] Transitions to channel halt mode immediately after the CHMDC[1:0] bits are set to 10b before bus off recovery.

Note 1. To allow transition to channel reset mode after communication is completed, set the CHMDC[1:0] bits to 10b and confirm that communication has been completed and transition to channel halt mode has been made, and then set the CHMDC[1:0] bits to 01b.
Note 2. While the CAN bus is locked at the dominant level, transition to channel halt mode is not made. In that case, enter channel reset mode. The CAN bus status can be confirmed with the ERFLL.BLF flag that becomes 1 when dominant lock is detected.
Note 3. In case of a transition from channel reset mode to channel halt mode, transition to channel halt mode after setting the CFGL and CFGH registers in channel reset mode.

(4) Channel Communication Mode

In channel communication mode, CAN communication is performed. Each channel has the following communication states during CAN communication.

- Idle: Neither reception nor transmission is in progress.
- Reception: Receiving a message sent from another node.
- Transmission: Transmitting a message.

Downloaded from Arrow.com.

- Bus off: Isolated from CAN communication.

When the CTRL.CHMDC[1:0] bits are set to 00b, the channel transitions to channel communication mode. After that, when 11 consecutive recessive bits have been detected, the STSL.COMSTS flag becomes 1 (communication is ready) and transmission and reception are enabled on the CAN network as an active node. At this time, transmission and reception of messages can be started.

(5) Bus Off State

A channel transitions to the bus off state according to the transmit/receive error counter increment/decrement rules of the ISO 11898-1 standard.
How to return from the bus off state is set by the CTRH.BOM[1:0] bits.

- When CTRH.BOM[1:0] = 00b:
  Bus off recovery is compliant with the ISO 11898-1 standard. After 11 consecutive recessive bits have been detected 128 times, a channel returns from the bus off state to the CAN communication ready state (error active state).
  At that time, the STSH.TEC[7:0] and STSH.REC[7:0] flags are initialized to 00h and the ERFLL.BORF flag becomes 1 (bus off recovery is detected). When the CTRL.CHMDC[1:0] bits are set to 10b (channel halt mode) in the bus off state, the channel transitions to channel halt mode after bus off recovery has been completed (11 consecutive recessive bits have been detected 128 times).
- When CTRH.BOM[1:0] = 01b:
  When a channel transitions to the bus off state, the CTRL.CHMDC[1:0] bits are set to 10b and the channel transitions to channel halt mode. At that time, the STSH.TEC[7:0] and STSH.REC[7:0] flags are initialized to 00h but the ERFLL.BORF flag is not set to 1.
- When CTRH.BOM[1:0] = 10b:
  When a channel has transitioned to the bus off state, the CTRL.CHMDC[1:0] bits are set to 10b. After bus off recovery has been completed (11 consecutive recessive bits have been detected 128 times), the channel transitions to channel halt mode. At that time, the STSH.TEC[7:0] and STSH.REC[7:0] flags are initialized to 00h and the ERFLL.BORF flag becomes 1.
- When CTRH.BOM[1:0] = 11b:
  When the CHMDC[1:0] bits are set to 10b in the bus off state, the channel transitions to channel halt mode before bus off recovery is completed. At that time, the STSH.TEC[7:0] and STSH.REC[7:0] flags are initialized to 00h but the ERFLL.BORF flag is not set to 1.
  However, the BORF flag becomes 1 if a CAN module transitions to error active state (by detecting 128 times of 11 consecutive recessive bits) before the CTRL.CHMDC[1:0] bits are set to 10b.

If the channel transitions to channel halt mode simultaneously when the program writes a value to the CTRL.CHMDC[1:0] bits, writing by the program takes precedence. An automatic transition to channel halt mode when the CTRH.BOM[1:0] bits are set to 01b or 10b is made only when the CTRL.CHMDC[1:0] bits are 00b (channel communication mode).
Furthermore, setting the CTRL.RTBO bit to 1 allows forcible return from the bus off state. As soon as the CTRL.RTBO bit is set to 1, the state changes to the error active state. After 11 consecutive recessive bits have been detected, the condition of CAN module becomes ready for communication. In this case, the ERFLL.BORF flag is not set to 1 and the STSH.TEC[7:0] and STSH.REC[7:0] flags are initialized to 00h. Write 1 to the CTRL.RTBO bit when the CTRH.BOM[1:0] value is 00b.

Downloaded from Arrow.com.

Table 36.8 Registers Initialized in Global Reset Mode or Channel Reset Mode

Register	Bit/Flag
CTRL	CHMDC[1:0]
CTRH	CTMS[1:0], CTME
STSL	CHLTSTS, EPSTS, BOSTS, TRMSTS, RECSTS, COMSTS
STSH	REC[7:0], TEC[7:0]
ERFLL	ADERR, B0ERR, B1ERR, CERR, AERR, FERR, SERR, ALF, BLF, OVLF, BORF, BOEF, EPF, EWF, BEF
ERFLH	CRCREG[14:0]
CFCCL0	When transmit/receive FIFO buffer is in transmit mode: CFE
CFSTS0	When transmit/receive FIFO buffer is in transmit mode: CFMC[5:0], CFTXIF, CFRXIF, CFMLT, CFFLL, CFEMP
TMCp	TMOM, TMTAR, TMTR
TMSTSp	TMTARM, TMTRM, TMTRF[1:0], TMTSTS
TMTRSTS	TMTRSTSp
TMTCSTS	TMTCSTSp
TMTASTS	TMTASTSp
THLCC0	THLE
THLSTS0	THLMC[3:0], THLIF, THLELT, THLFLL, THLEMP
GTINTSTS	THIF0, CFTIF0, TAIF0, TSIF0

Table 36.9 Registers Initialized Only in Global Reset Mode

Register	Bit/Flag
GSTS	GHLTSTS
GERFLL	THLES, MES, DEF
GTSC	TS[15:0]
RMND0	RMNSn
RFCCm	RFE
RFSTSm	RFMC[5:0], RFIF, RFMLT, RFFLL, RFEMP
CFCCL0	When transmit/receive FIFO buffer is in receive mode: CFE
CFSTS0	When transmit/receive FIFO buffer is in receive mode: CFMC[5:0], CFTXIF, CFRXIF, CFMLT, CFFLL, CFEMP
RFMSTS	RFmMLT
CFMSTS	CF0MLT
RFISTS	RFmIF
CFISTS	CF0IF
GTSTCFG	RTMPS[2:0]
GTSTCTRL	RTME

Downloaded from Arrow.com.

36.4 Reception Function

There are two reception types.

- Reception by receive buffers:
  Zero to 16 receive buffers can be shared by all channels. Since messages stored in receive buffers are overwritten at each reception, the latest receive data can always be read.
- Reception by receive FIFO buffers and transmit/receive FIFO buffers (receive mode):
  Two receive FIFO buffers can be shared by all channels and one dedicated transmit/receive FIFO buffer is provided for each channel. The FIFO buffers can hold the number of received messages set by the RFCCm.RFDC[2:0] bits and CFCCL0.CFDC[2:0] bits, and messages can be read sequentially from the oldest.

36.4.1 Data Processing Using the Receive Rule Table

Data processing using the receive rule table allows selected messages to be stored in the specified buffer. Data processing includes acceptance filter processing, DLC filter processing, routing processing, label addition processing, and mirror function processing.
Up to 16 receive rules can be registered per channel. If receive rules are not set, no message can be received. Figure 36.4 illustrates how receive rules are registered.

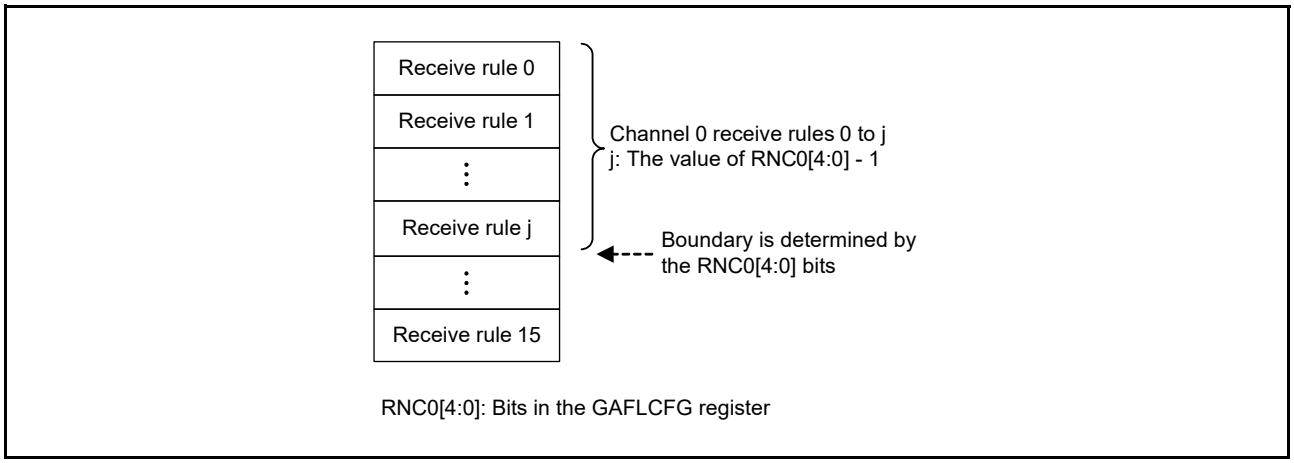


Figure 36.4 Entry of Receive Rules

Each receive rule consists of 12 bytes in the GAFLIDLj, GAFLIDHj, GAFLMLj, GAFLMHj, GAFLPLj, and GAFLPHj registers. The GAFLIDLj and GAFLIDHj registers are used to set ID, IDE bit, RTR bit, and the mirror function, the GAFLMLj and GAFLMHj registers are used to set mask, the GAFLPLj and GAFLPHj registers are used to set label information to be added, DLC value, and storage receive buffer, and storage FIFO buffer.

(1) Acceptance Filter Processing

In the acceptance filter processing, the ID data, IDE bit, and RTR bit in a received message are compared with the ID data, IDE bit, and RTR bit set in the receive rule of the corresponding channel. When all these bits match, the message passes through the acceptance filter processing. The ID data, IDE bit, and RTR bit in a received message which correspond to bits that are set to 0 (bits are not compared) in the GAFLMLj and GAFLMHj registers are not compared and are regarded as matched.
Check begins with the receive rule with the smallest rule number of the corresponding channel. When all the bits to be compared in a received message match the bits set in the receive rule or when all the receive rules are compared without any match, filter processing stops. If there is no matching receive rule, the received message is not stored in the receive buffer or FIFO buffer.

Downloaded from Arrow.com.

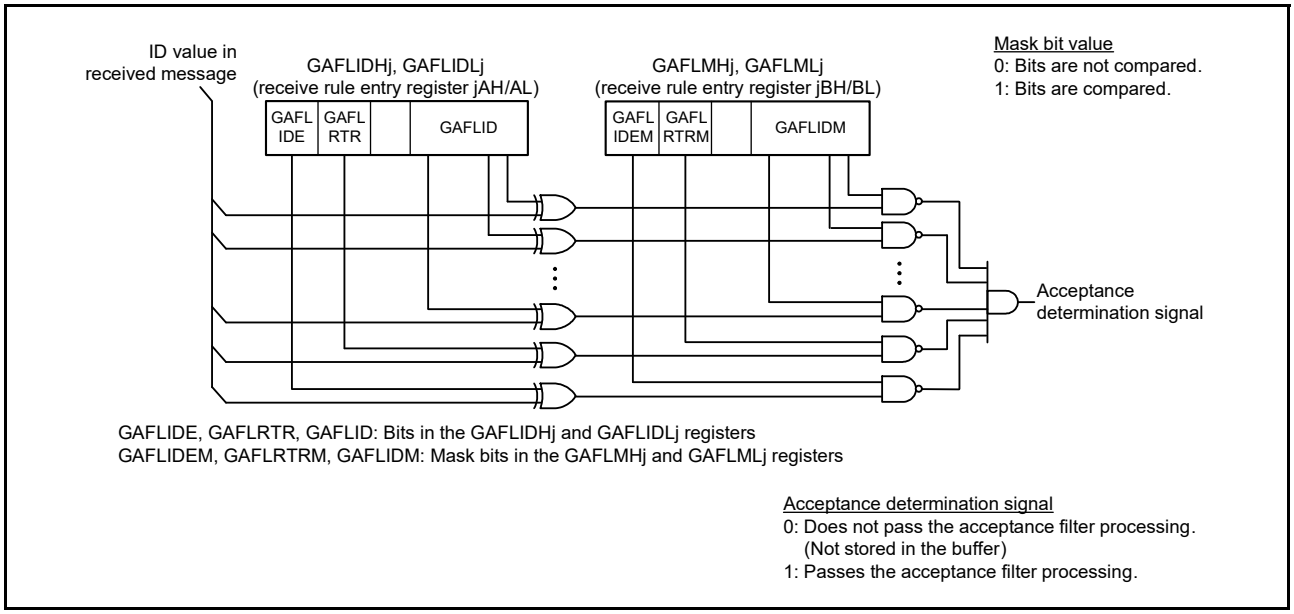


Figure 36.5 Acceptance Filter Function

(2) DLC Filter Processing

When the GCFGL.DCE bit is set to 1 (DLC check is enabled), DLC filter processing is added to messages that passed through the acceptance filter processing. When the DLC value in a message is equal to or larger than the DLC value set in the receive rule, the message passes through the DLC filter processing.

When a message has passed through the DLC filter processing with the GCFGL.DRE bit set to 0 (DLC replacement is disabled), the DLC value in the received message is stored in the buffer. In this case, all the data bytes in the received message are stored in the buffer.

When a message has passed through the DLC filter processing with the GCFGL.DRE bit set to 1 (DLC replacement is enabled), the DLC value in the receive rule is stored in the buffer instead of the DLC value in the received message. In this case, a value of 00h is written to data bytes that are larger than the DLC value in the receive rule.

When the DLC value in the received message is smaller than that in the receive rule, the message does not pass through the DLC filter processing. In this case, the message is not stored in the receive buffer or the FIFO buffer and the GERFLL.DEF flag becomes 1 (a DLC error is present).

(3) Routing Processing

Messages that passed through the acceptance filter processing and the DLC filter processing are stored in receive buffers, receive FIFO buffers, or transmit/receive FIFO buffers (set to receive mode). Message storage destination is set by the GAFLPLj.GAFLRMV, GAFLRMDP[6:0], GAFLFDP4, GAFLFDP1, and GAFLFDP0 bits. Messages that passed through the acceptance filter processing and the DLC filter processing can be stored in up to two buffers.

(4) Label Addition Processing

It is possible to add 12-bit label information to messages that passed through the filter processing and store them in buffers. This label information is set in the GAFLPHj.GAFLPTR[11:0] bits.

(5) Mirror Function Processing

The mirror function allows reception of messages transmitted from the own CAN node. The mirror function is made available by setting the GCFGL.MME bit to 1 (mirror function is enabled).

When the mirror function is in use, receive rules for which the GAFLIDHj.GAFLLB bit is set to 0 are used for data processing when receiving messages transmitted from other CAN nodes. When receiving messages transmitted from the own CAN node, receive rules for which the GAFLIDHj.GAFLLB bit is set to 1 are used for data processing.

Downloaded from Arrow.com.

36.4.2 Timestamp

The timestamp counter is a 16-bit free-running counter used for recording message receive time. The timestamp counter value is fetched at the start-of-frame (SOF) timing of a message and is then stored in a receive buffer or a FIFO buffer together with the message ID and data. PCLK or the CAN bit time clock is selectable as a timestamp counter clock source from the GCFGL.TSSS bit. The clock obtained by dividing the selected clock source by the GCFGL.TSP[3:0] value is used as the timestamp counter count source.

When the CAN bit time clock is used as a clock source, the timestamp counter stops when the corresponding channel transitions to channel reset mode or channel halt mode. When PCLK is used as a clock source, the timestamp function is not affected by channel mode.

The timestamp counter value is reset to 0000h by setting the GCTRH.TSRST bit to 1.

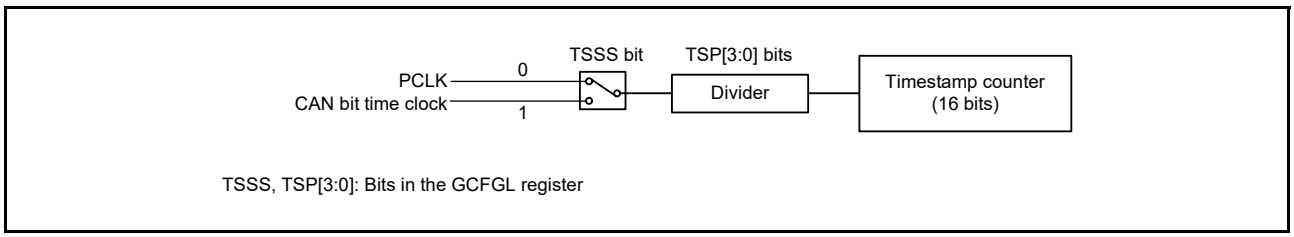


Figure 36.6 Timestamp Function Block Diagram

Downloaded from Arrow.com.

36.5 Transmission Functions

There are two types of transmission.

- Transmission using transmit buffers:
  Each channel has 4 buffers.
- Transmission using transmit/receive FIFO buffers (transmit mode):
  Each channel has one FIFO buffer. Up to 16 messages can be contained in a single FIFO buffer. Each FIFO buffer is used with a link to a transmit buffer. Only the message to be transmitted next in a FIFO buffer becomes the target of transmit priority determination. Messages are transmitted sequentially on a first-in, first-out basis.

Figure 36.7 shows the allocation of transmit/receive FIFO buffer link.

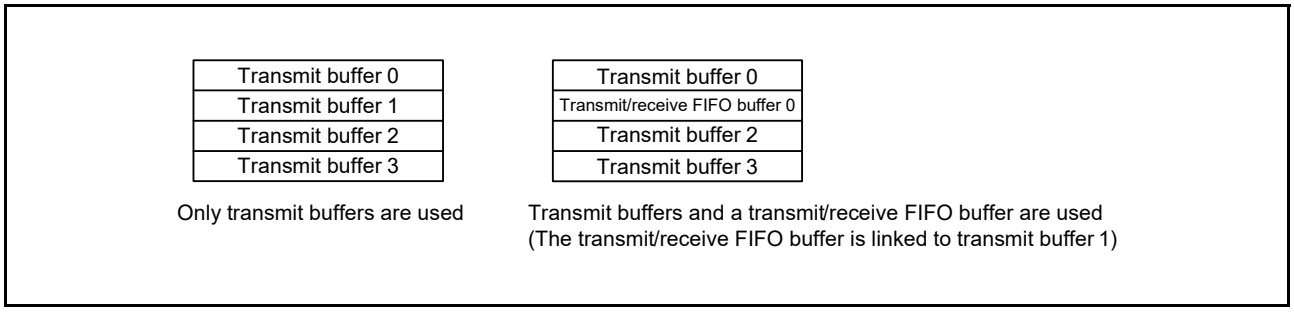


Figure 36.7 Allocation of Transmit/Receive FIFO Buffer Links

36.5.1 Transmit Priority Determination

If transmit requests are issued from multiple buffers in the same channel, transmit priority is determined.
The priority is determined by using one of the following methods.

- ID priority (GCFGL.TPRI bit = 0)
- Transmit buffer number priority (GCFGL.TPRI bit = 1)

The setting of the GCFGL.TPRI bit is enabled in all CAN channels.
When the GCFGL.TPRI bit is set to 0, messages are transmitted according to the priority of stored message IDs. Priority of IDs conforms to the CAN bus arbitration specification defined in the ISO 11898-1 standard. IDs of messages stored in transmit buffers and transmit/receive FIFO buffers (set to transmit mode) are targets of priority determination. When transmit/receive FIFO buffers are used, the oldest message in a FIFO buffer becomes the target of priority determination. When a message is being transmitted from a transmit/receive FIFO buffer, the next message in the FIFO buffer becomes the target of priority determination. When the GCFGL.TPRI bit is set to 1, the message in the transmit buffer of the minimum number among buffers with a transmit request is transmitted first. When transmit/receive FIFO buffers are linked to transmit buffers, transmit priority is determined according to linked transmit buffer numbers.
When messages are retransmitted due to an arbitration lost or an error, transmit priority determination is made again regardless of the GCFGL.TPRI bit setting.

Downloaded from Arrow.com.

36.5.2 Transmission Using Transmit Buffers

Setting the transmit request bit (TMCp.TMTR bit) in a transmit buffer to 1 (transmission is requested) allows transmission of data frames or remote frames.
Transmit result is shown by the corresponding TMSTSp.TMTRF[1:0] flags. When transmit completes successfully, the TMSTSp.TMTRF[1:0] flags become 10b (transmission has been completed (without transmit abort request)) or 11b (transmission has been completed (with transmit abort request)).

(1) Transmit Abort Function

With respect to transmit buffers for which the TMSTSp.TMTRM flag is 1 (a transmit request is present), when the TMCp.TMTAR bit is set to 1 (transmit abort is requested), the transmit request is canceled. When transmit abort is completed, the TMSTSp.TMTRF[1:0] flags become 01b (transmit abort has been completed) and the transmit request is canceled (the TMSTSp.TMTRM flag becomes 0).
A message that is being transmitted or a message to be transmitted next according to the transmit priority determination cannot be aborted. However, when an arbitration lost or an error has occurred while a message for which the TMCp.TMTAR bit is set to 1 is being transmitted, retransmission is not performed.

(2) One-Shot Transmission Function (Retransmission Disabling Function)

When the TMCp.TMOM bit is set to 1 (one-shot transmission is enabled), transmission is performed only once. Even if an arbitration lost or an error occurs, retransmission is not performed.
One-shot transmit result is shown by the corresponding TMSTSp.TMTRF[1:0] flags. When one-shot transmit completes successfully, the TMSTSp.TMTRF[1:0] flags become 10b or 11b. When an arbitration lost or an error has occurred, the TMSTSp.TMTRF[1:0] flags become 01b (transmit abort has been completed).

36.5.3 Transmission Using FIFO Buffers

Messages of a volume of the FIFO buffer depth set by the CFCCL0.CFDC[2:0] bits can be stored in a single transmit/receive FIFO buffer. Messages are transmitted sequentially on a first-in, first-out basis.
Transmit/receive FIFO buffers are linked to transmit buffers selected by the CFCCH0.CFTML[1:0] bits.
When the CFCCL0.CFE bit is set to 1 (transmit/receive FIFO buffers are used), transmit/receive FIFO buffers become targets of transmit priority determination. Priority determination is made for only the message to be transmitted next in a FIFO buffer.
When the CFCCL0.CFE bit is set to 0 (no transmit/receive FIFO buffer is used), the CFSTS0.CFEMP flag becomes 1 (the transmit/receive FIFO buffer contains no message (buffer empty)) at the timing below.

- The transmit/receive FIFO buffer becomes empty immediately when the message in it is not being transmitted or is not to be transmitted next.
- The transmit/receive FIFO buffer becomes empty after transmission completion, CAN bus error detection, or arbitration lost when the message in it is being transmitted or to be transmitted next.

When the CFCCL0.CFE bit is set to 0, all messages in transmit/receive FIFO buffers are lost and messages cannot be stored in FIFO buffers. Confirm that the CFSTS0.CFEMP flag becomes 1 before setting the CFCCL0.CFE bit to 1 again.

(1) Interval Transmission Function

To transmit messages from the same FIFO buffer while a transmit/receive FIFO buffer that is set to transmit mode is in use, message transmission interval time can be set.
Immediately after the first message has been transmitted successfully from the FIFO buffer with the CFCCL0.CFE bit set to 1, the interval timer starts counting (after EOF7 of the CAN protocol). After that, when the interval time has passed, the next message is transmitted. The interval timer stops in channel reset mode or by setting the CFCCL0.CFE bit to 0.
The interval time is set by the CFCCH0.CFITT[7:0] bits. When the interval timer is not used, set the

Downloaded from Arrow.com.

CFCCH0.CFITT[7:0] bits to 00h.
Select an interval timer count source by the CFCCH0.CFITR and CFITSS bits. When the CFCCH0.CFITR and CFITSS bits are set to 00b, the clock obtained by frequency-dividing PCLK by the GCFGH.ITRCP[15:0] value is used as a count source. When the CFCCH0.CFITR and CFITSS bits are set to 10b, the clock obtained by frequency-dividing PCLK by the GCFGH.ITRCP[15:0] value × 10 is used as a count source. When the CFCCH0.CFITR and CFITSS bits are set to x1b, the CAN bit time clock is used as a count source.
The interval time is calculated by the following equations where M is the set GCFGH.ITRCP[15:0] value and N is the set CFCCH0.CFITT[7:0] value.

- When CFCCH0.CFITR and CFITSS = 00b

$$\frac{1}{PCLK} \times M \times N$$

- When CFCCH0.CFITR and CFITSS = 10b

$$\frac{1}{PCLK} \times M \times 10 \times N$$

- When CFCCH0.CFITR and CFITSS = x1b
(fCANBIT is CAN bit time clock frequency)

$$\frac{1}{fCANBIT} \times N$$

Figure 36.8 shows the interval timer block diagram.

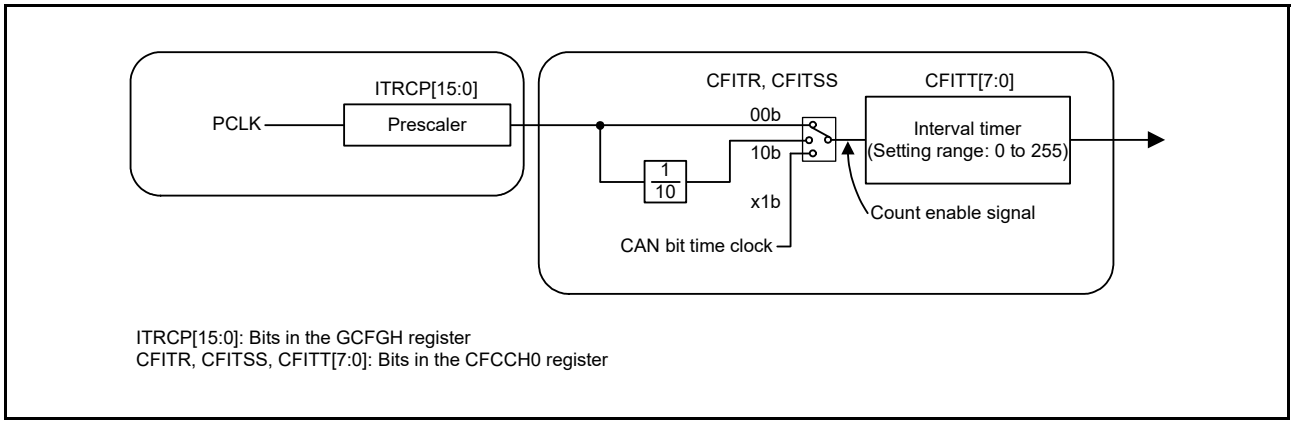


Figure 36.8 Interval Timer Block Diagram

Downloaded from Arrow.com.

Figure 36.9 shows the interval timer timing chart.

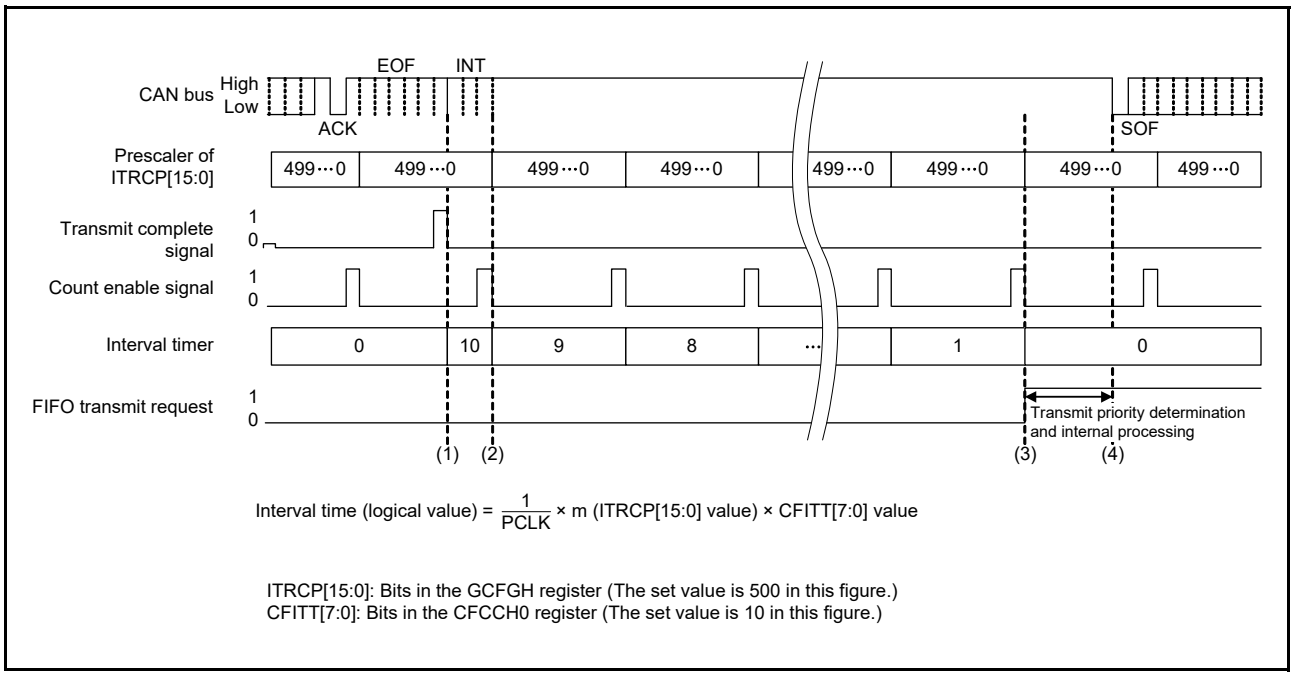


Figure 36.9 Interval Timer Timing Chart

(1) The interval timer starts counting upon completion of transmission. Since the prescaler is not initialized at the time of transmission completion, the first interval time contains an error of up to one count of the interval timer.
(2) The interval timer is decremented by the next count enable signal.
(3) When the interval timer has decreased to 0, the transmit/receive FIFO buffer issues a transmit request.
(4) The transmit/receive FIFO buffer is determined for the next transmission by the priority determination, it starts transmitting data. Transmission starts with a delay of three CAN bit time clock cycles or less from the issue of transmit request.

Downloaded from Arrow.com.

36.5.4 Transmit History Function

Information of transmitted messages can be stored in the transmit history buffer. Each channel has a single transmit history buffer that can contain 8 sets of transmit history data.
A message transmit source buffer type can be selected by the THLCC0.THLDTE bit. Whether to store transmit history data for each message can be set by the CFIDH0.THLEN bit.
After transmit completes successfully, information of the following transmit messages is stored in the transmit history buffer as transmit history data. After successful completion of transmit, process may be delayed by up to 38 clocks of PCLK before the transmit history data is stored.

- Buffer type 01b: Transmit buffer
  10b: Transmit/receive FIFO buffer
- Buffer number Number of source transmit buffer or transmit/receive FIFO buffer.
  This number depends on buffer types. See Table 36.10.
- Label data Label information of transmit message

Table 36.10 Transmit History Data Buffer Numbers

Buffer Number	Buffer Type	
	01b	10b
00b	Transmit buffer 0	Numbers of transmit buffers linked to transmit/receive FIFO buffers by the CFCCH0.CFTML[1:0] bits.
01b	Transmit buffer 1	
10b	Transmit buffer 2	
11b	Transmit buffer 3	

Label data is used to identify each message. A unique label data can be added to each message transmitted from a transmit buffer or transmit/receive FIFO buffer.
Transmit history data can be read from the THLACC0 register. If it is attempted to store new transmit history data while the buffer is full, the buffer overflows and the new data is discarded.

Downloaded from Arrow.com.

36.6 Test Function

The test function is classified into communication tests and global tests.
Communication tests: Performed for each channel.

- Standard test mode
- Listen-only mode
- Self-test mode 0 (external loopback mode)
- Self-test mode 1 (internal loopback mode)
  Global tests: Performed in entire module
- RAM test (read/write test)

36.6.1 Standard Test Mode

Standard test mode allows CRC test.

36.6.2 Listen-Only Mode

Listen-only mode allows reception of data frames and remote frames. Only recessive bits are transmitted on the CAN bus, and the ACK bit, overload flag, and active error flag are not transmitted. Listen-only mode is available for detecting the communication speed.
Do not make a transmit request from any buffer in listen-only mode.
Figure 36.10 shows the connection when listen-only mode is selected.

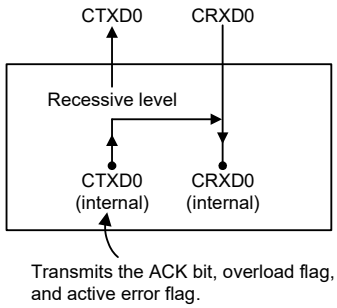


Figure 36.10 Connection When Listen-Only Mode is Selected

Downloaded from Arrow.com.

36.6.3 Self-Test Mode (Loopback Mode)

In self-test mode, transmitted messages are compared with the receive rule of the own channel and the messages are stored in a buffer if they have passed through the filter processing. Messages transmitted from other CAN nodes are compared only with the receive rule for which the GAFLIDHj.GAFLLB bit is set to 0 (when a message transmitted from another CAN node is received).
If the mirror function and self-test mode are both enabled, the self-test mode setting takes precedence.

(1) Self-Test Mode 0 (External Loopback Mode)

Self-test mode 0 is used to perform a loopback test within a channel including the CAN transceiver.
In self-test mode 0, transmitted messages are handled as messages received through the CAN transceiver and are stored in a buffer. An ACK bit is generated to receive messages transmitted from the own CAN node.
Figure 36.11 shows the connection when self-test mode 0 is selected.

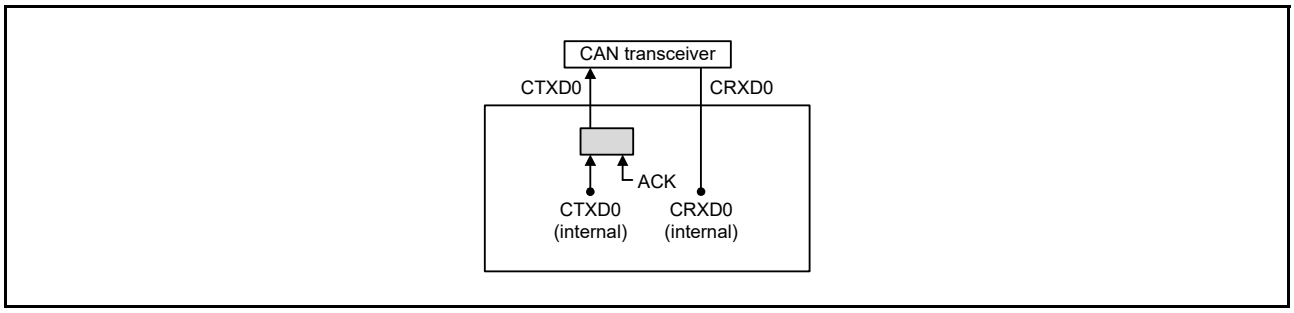


Figure 36.11 Connection When Self-Test Mode 0 is Selected

(2) Self-Test Mode 1 (Internal Loopback Mode)

In self-test mode 1, transmitted messages are handled as received messages and are stored in a buffer. An ACK bit is generated to receive messages transmitted from the own CAN node.
In self-test mode 1, internal feedback from the internal CTXD0 pin to the internal CRXD0 pin is performed. The external CRXD0 pin input is isolated. The external CTXD0 pin outputs only recessive bits.
Figure 36.12 shows the connection when self-test mode 1 is selected.

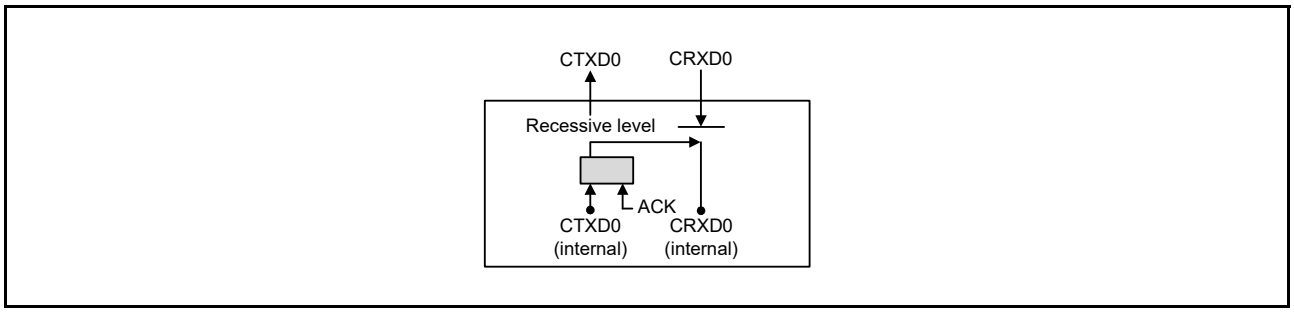


Figure 36.12 Connection When Self-Test Mode 1 is Selected

36.6.4 RAM Test

The RAM test function allows accesses to all CAN RAM addresses.
When the RAM test function is used, the RAM is divided into pages of 256 bytes each. RAM test page is set by the GTSTCFG.RTMPS[2:0] bits. Data in the set page can be read from and written to the RPGACCr register. The available total RAM size is 544 bytes (0220h).

Downloaded from Arrow.com.

36.7 Interrupt

The CAN module has 5 interrupts that are grouped into global interrupts and channel interrupts.

Global interrupts (2 interrupts):

- Global receive FIFO interrupt (RXFINT)
- Global error interrupt (GLERRINT)

Channel interrupts (3 interrupts per channel):

- Channel transmit interrupt (TXINT)
  - Transmit complete interrupt
  - Transmit abort interrupt
  - Transmit/receive FIFO transmit complete interrupt (transmit mode)
  - Transmit history interrupt
- Transmit/receive FIFO receive interrupt (COMFRXINT)
- Channel error interrupt (CHERRINT)

When an interrupt request is generated, the corresponding CAN module interrupt request flag becomes 1 (interrupt request present). In that case, when the interrupt enable bit is set to 1 (enabling interrupts), an interrupt request is output from the CAN module. (Generation of interrupts also is controlled by the interrupt function.)
Setting the interrupt request flag to 0 (no interrupt request present) or setting the interrupt enable bit to 0 (disabling interrupts) clears the current interrupt request. The next interrupt request is not generated until the interrupt request is cleared.

For details on the setting of the interrupt functions, refer to section 15, Interrupt Controller (ICUb).
In the following pages, Table 36.11 lists the CAN interrupt sources, Figure 36.13 shows the CAN global interrupt block diagram, and Figure 36.14 shows the CAN channel interrupt block diagram.

Downloaded from Arrow.com.

Table 36.11 List of CAN Interrupt Sources

Item	Interrupt Source		Corresponding Interrupt Request Flag*1	Corresponding Interrupt Enable Bit *1
Global interrupts	Global receive FIFO	Receive FIFO 0	RFSTS0.RFIF	RFCC0.RFIE
		Receive FIFO 1	RFSTS1.RFIF	RFCC1.RFIE
	Global error		GERFLL.DEF	GCTRL.DEIE
			GERFLL.MES	GCTRL.MEIE
			GERFLL.THLES	GCTRL.THLEIE
Channel interrupts	Channel transmit	Transmit complete	TMSTSp.TMTRF[1:0]	TMIEC.TMIEp
		Transmit abort	TMSTSp.TMTRF[1:0]	CTRH.TAIE
		Transmit/receive FIFO transmit	CFSTS0.CFTXIF	CFCCL0.CFTXIE
		Transmit history	THLSTS0.THLIF	THLCC0.THLIE
	Transmit/receive FIFO receive		CFSTS0.CFRXIF	CFCCL0.CFRXIE
	Channel error		ERFLL.BEF	CTRL.BEIE
			ERFLL.ALF	CTRL.ALIE
			ERFLL.BLF	CTRL.BLIE
			ERFLL.OVLF	CTRL.OLIE
			ERFLL.BORF	CTRL.BORIE
			ERFLL.BOEF	CTRL.BOEIE
			ERFLL.EPF	CTRL.EPIE
			ERFLL.EWF	CTRL.EWIE
	Wakeup		None	None

Note 1. For details on the interrupt request flags and interrupt enable bits, refer to section 15, Interrupt Controller (ICUb).

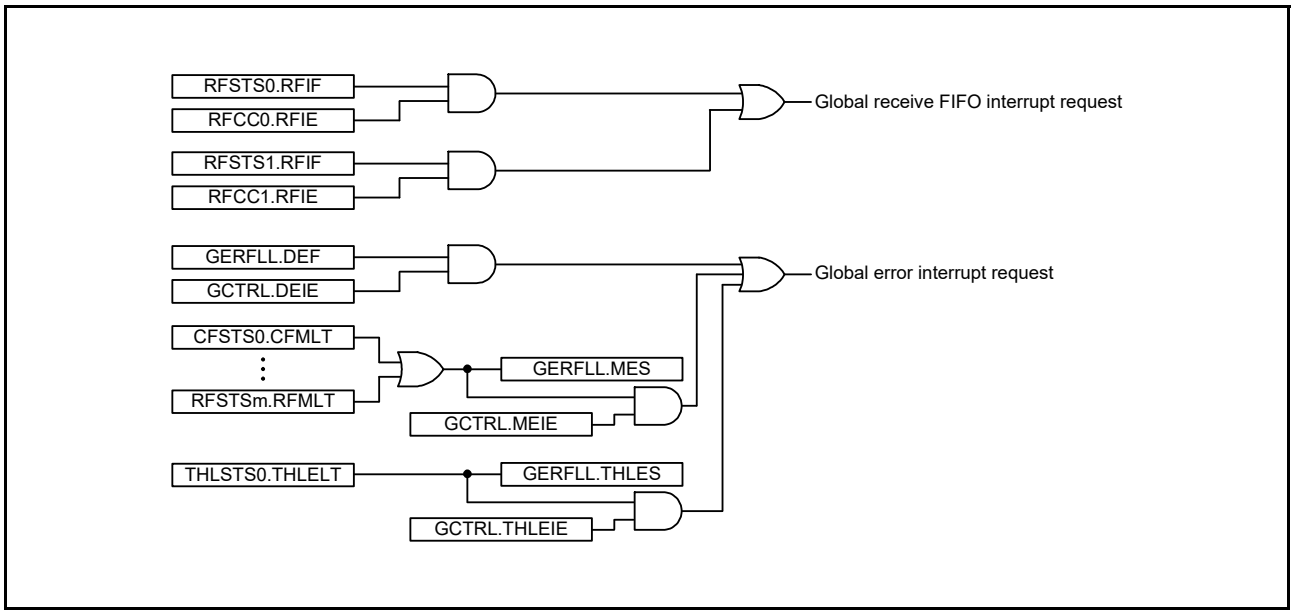


Figure 36.13 CAN Global Interrupt Block Diagram

Downloaded from Arrow.com.

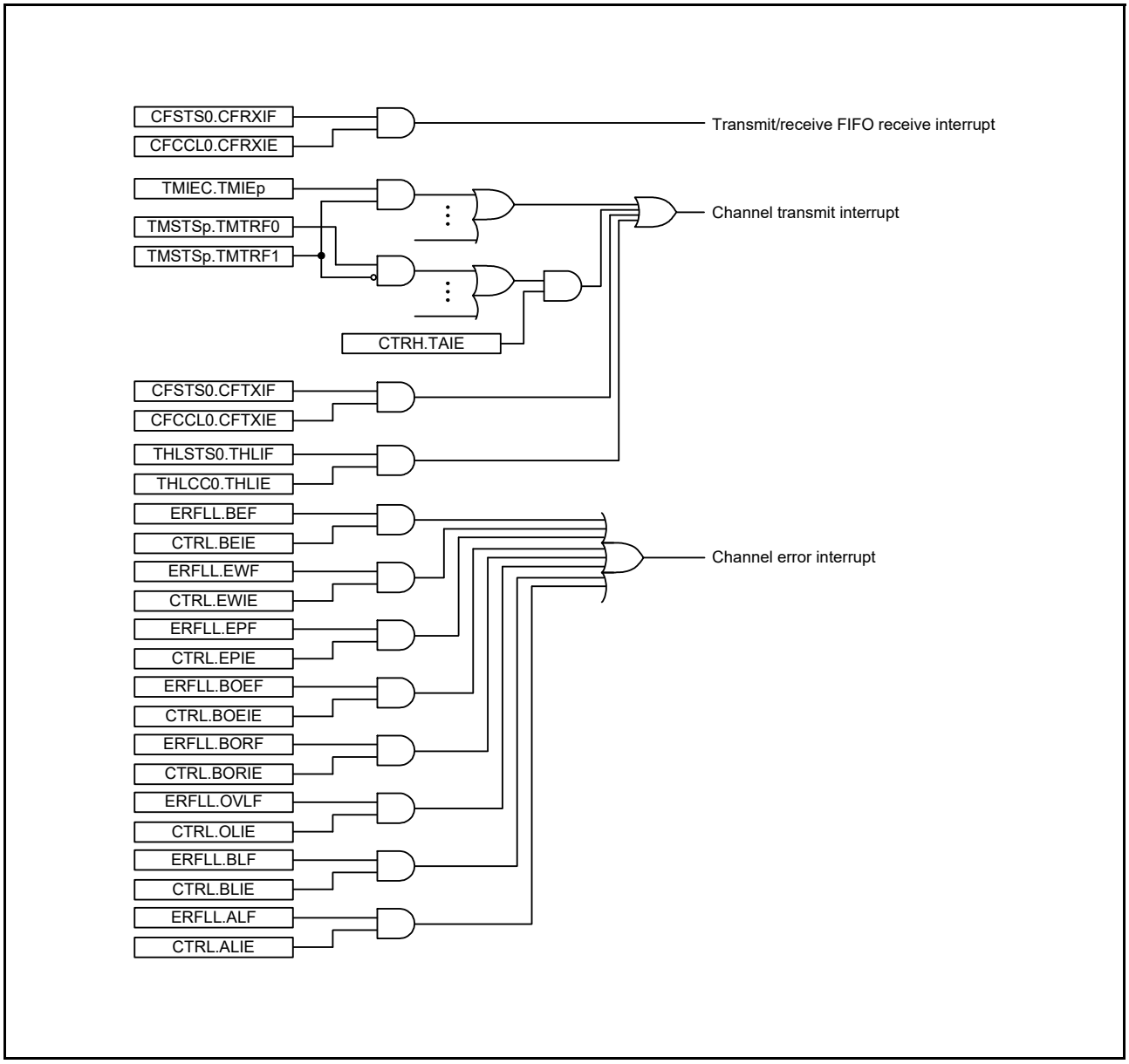


Figure 36.14 CAN Channel Interrupt Block Diagram

Downloaded from Arrow.com.

36.8 RAM Window

The CAN area from 000A 83A0h to 000A 8681h consists of two windows. The GRWCR.RPAGE bit is used to switch the allocation of registers.

- Registers allocated when the GRWCR.RPAGE bit is set to 0 (window 0 selected)
  Receive rule entry registers: GAFLIDLj, GAFLIDHj, GAFLMLj, GAFLMHj, GAFLPLj, GAFLPHj
  RAM test registers: RPGACCr

- Registers allocated when the GRWCR.RPAGE bit is set to 1 (window 1 selected)
  Receive buffer registers: RMIDLn, RMIDHn, RMTSn, RMPTRn, RMDF0n to RMDF3n
  Receive FIFO access registers: RFIDLm, RFIDHm, RFTSm, RFPTRm, RFDF0m to RFDF3m
  Transmit/receive FIFO access registers: CFIDL0, CFIDH0, CFTS0, CFPTR0, CFDF00 to CFDF30
  Transmit buffer registers: TMIDLp, TMIDHp, TMPTRp, TMDF0p to TMDF3p
  Transmit history buffer access register: THLACC0

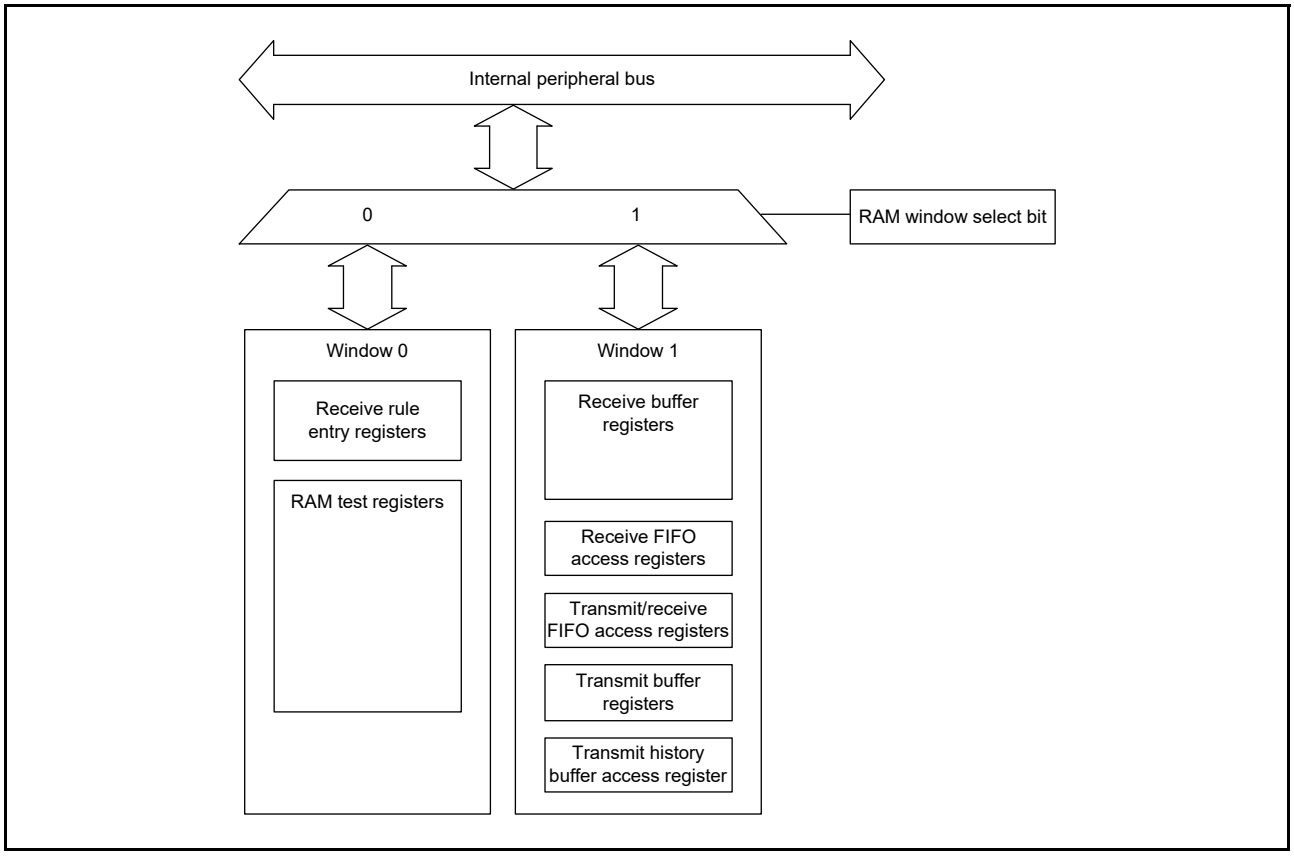


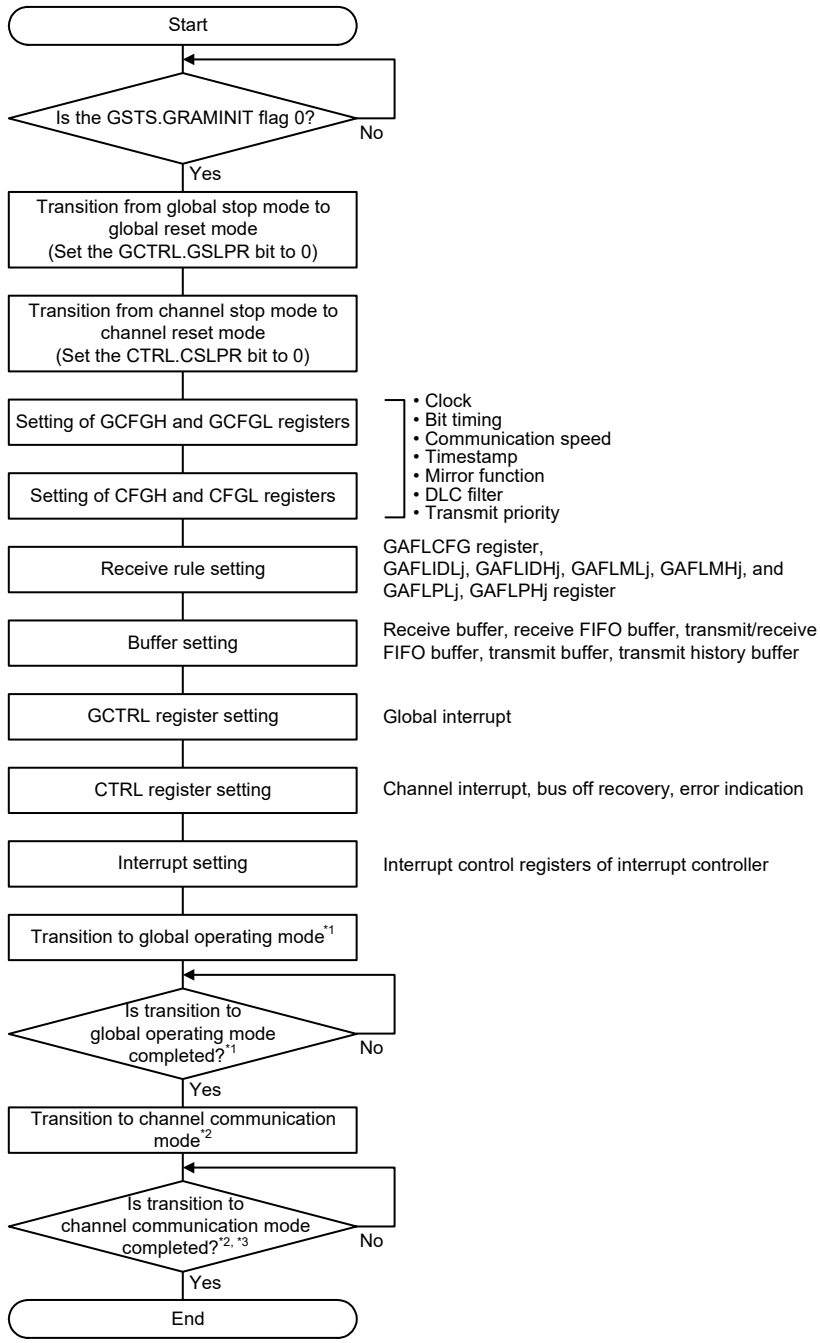
Figure 36.15 RAM Window

Downloaded from Arrow.com.

36.9 Initial Settings

The CAN module initializes the CAN RAM after the operation of the CAN module is enabled. The RAM initialization time is 276 cycles of PCLK. The GSTS.GRAMINIT flag becomes 1 (CAN RAM initialization is ongoing) during the RAM initialization and becomes 0 (CAN RAM initialization is finished) when the initialization is completed. Make CAN settings after the GSTS.GRAMINIT flag becomes 0.

Figure 36.16 shows the CAN setting procedure after the operation of the CAN module is enabled.



Note 1. When global mode is changed by the GCTRL.GSLPR and GMDC[1:0] bits, use the GSTS register to confirm that the mode is switched. Do not modify the GCTRL.GMDC[1:0] bits until the mode is switched.

Note 2. When channel mode is changed by the CTRL.CSLPR and CHMDC[1:0] bits, use the STSL register to confirm that the mode is switched. Do not modify the CTRL.CHMDC[1:0] bits until the mode is switched.

Note 3. After the channel transitions to channel communication mode, when 11 consecutive recessive bits have been detected, communication is ready (the STSL.COMSTS flag becomes 1) and transmission and reception are enabled on the CAN network as an active node. At this time, transmission and reception of messages can be started.

Figure 36.16 CAN Setting Procedure after the Operation of the CAN Module is Enabled

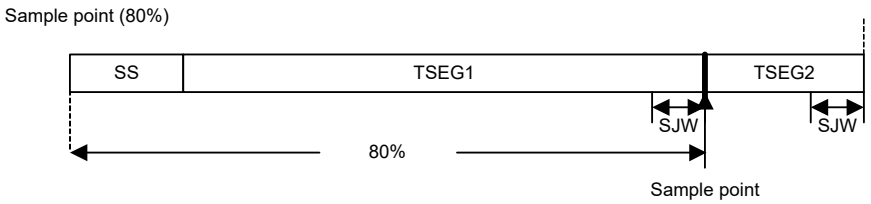
Downloaded from Arrow.com.

36.9.1 Clock Setting

Set the CAN clock source (fCAN) as a clock source of the CAN module. Select PCLK or CANMCLK with the GCFGL.DCS bit.

36.9.2 Bit Timing Setting

In the CAN protocol, one bit of a communication frame consists of three segments, SS, TSEG1, and TSEG2. Two of the segments, TSEG1 and TSEG2, can be set by the CFGH register for each channel. Sample point timing can be determined by setting two segments. This timing can be adjusted in units of 1 Time Quantum (referred to as Tq hereinafter). 1 Tq equals to one CAN Tq clock cycle. The CAN Tq clock is obtained by selecting the clock source with the GCFGL.DCS bit and selecting the clock division ratio with the CFGL.BRP[9:0] bits.
Figure 36.17 shows the bit timing chart. Table 36.12 shows an example of bit timing setting.



SS = 1 Tq fixed
Set TSEG1 to a range of 4 Tq to 16 Tq.
Set TSEG2 to a range of 2 Tq to 8 Tq.
Set SJW to a range of 1 Tq to 4 Tq.
Set SS + TSEG1 + TSEG2 to a range of 8 Tq to 25 Tq.
TSEG1 > TSEG2 ≥ SJW

- SS (synchronization segment):
  The SS is a segment that performs synchronization by monitoring the edge from recessive to dominant bits in the Interframe Space.
  Interframe Space consists of Intermission, Suspend Transmission, and Bus Idle. All nodes can start transmission during Bus Idle.
- TSEG1 (time segment 1):
  TSEG1 is a segment that absorbs physical delay on the CAN network. The physical delay on the CAN network is twice of the total of the delay on the CAN bus, the delay in the input comparator, and the delay in the output driver.
- TSEG2 (time segment 2):
  TSEG2 is a segment that compensates phase error due to an error in frequency.
- SJW (resynchronization jump width)
  The SJW is a length to extend or reduce time segment to compensate an error in phase due to phase error.

Figure 36.17 Bit Timing Chart

Table 36.12 Example of Bit Timing Setting

1 Bit	Set Value (Tq)				Sample Point (%) Note: See Figure 36.17
	SS	TSEG1	TSEG2	SJW	
8 Tq	1	4	3	1	62.50
	1	5	2	1	75.00
10 Tq	1	6	3	1	70.00
	1	7	2	1	80.00
16 Tq	1	10	5	1	68.75
	1	11	4	1	75.00
20 Tq	1	13	6	1	70.00
	1	15	4	3	80.00
24 Tq	1	15	8	1	66.67
	1	16	7	1	70.83

Downloaded from Arrow.com.

36.9.3 Communication Speed Setting

Set the CAN communication speed for each channel using the fCAN, baud rate prescaler division value (CFGL.BRP[9:0] bits), and Tq count per bit time.

Figure 36.18 shows the CAN clock control block diagram, and Table 36.13 shows an example of the communication speed setting.

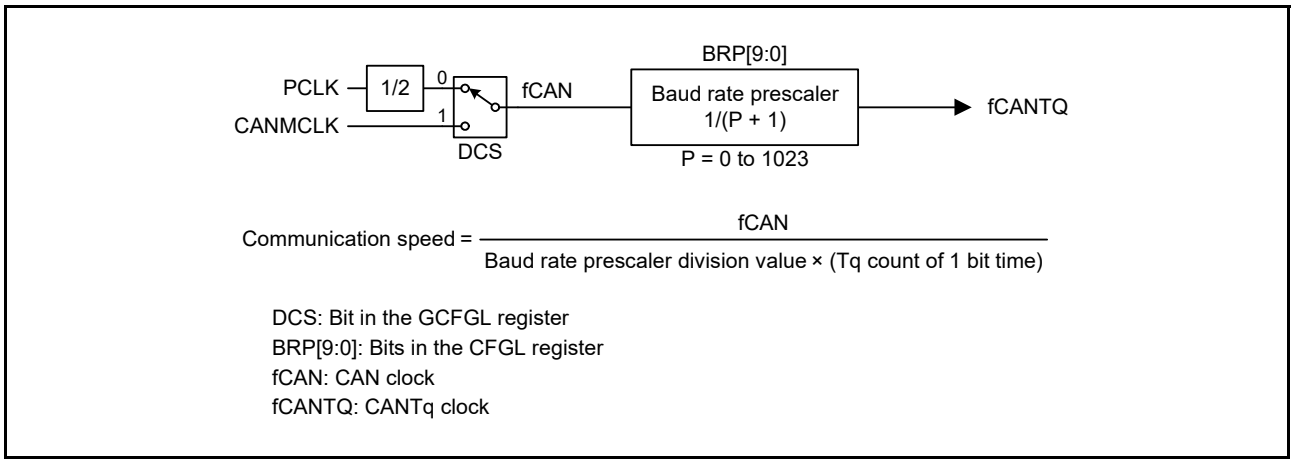


Figure 36.18 CAN Clock Control Block Diagram

Table 36.13 Example of Communication Speed Setting

	fCAN	
Communication Speed	**16 MHz**	**8 MHz**
1 Mbps	8 Tq (2) 16 Tq (1)	8 Tq (1)
500 kbps	8 Tq (4) 16 Tq (2)	8 Tq (2) 16 Tq (1)
250 kbps	8 Tq (8) 16 Tq (4)	8 Tq (4) 16 Tq (2)
83.3 kbps	8 Tq (24) 16 Tq (12)	8 Tq (12) 16 Tq (6)
33.3 kbps	8 Tq (60) 10 Tq (48) 16 Tq (30) 20 Tq (24)	8 Tq (30) 10 Tq (24) 16 Tq (15) 20 Tq (12)

Note: Values in ( ) are baud rate prescaler division values.

Downloaded from Arrow.com.

36.9.4 Receive Rule Setting

Receive rules can be set using receive rule-related registers.
Up to 16 receive rules can be registered.
Figure 36.19 shows the receive rule setting procedure.

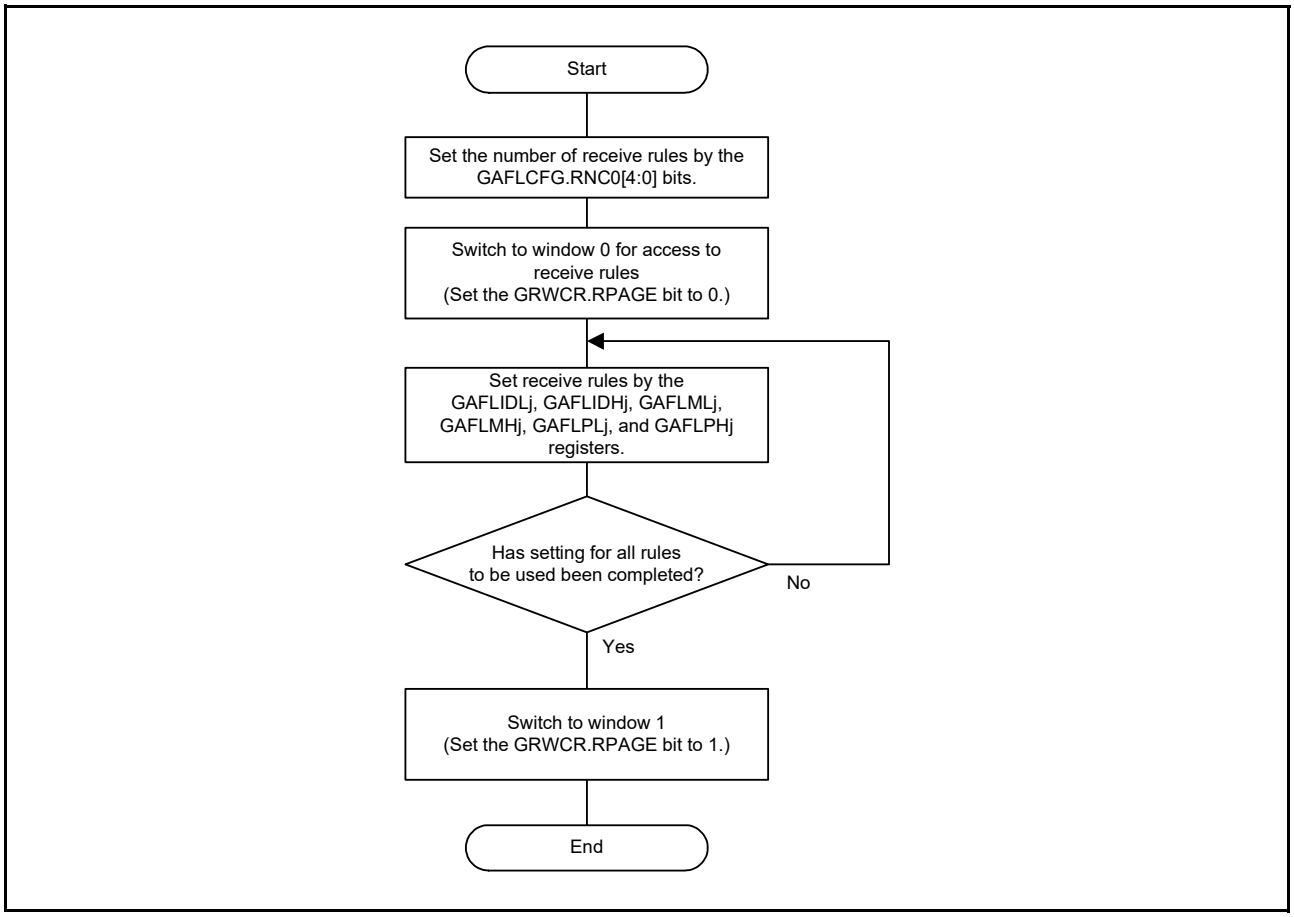


Figure 36.19 Receive Rule Setting Procedure

Downloaded from Arrow.com.

36.9.5 Buffer Setting

Set sizes and interrupt sources of buffers. For transmit/receive FIFO buffers that are set to transmit mode, set transmit buffers to be linked.

Figure 36.20 shows the buffer configuration. Figure 36.21 shows the buffer setting procedure.

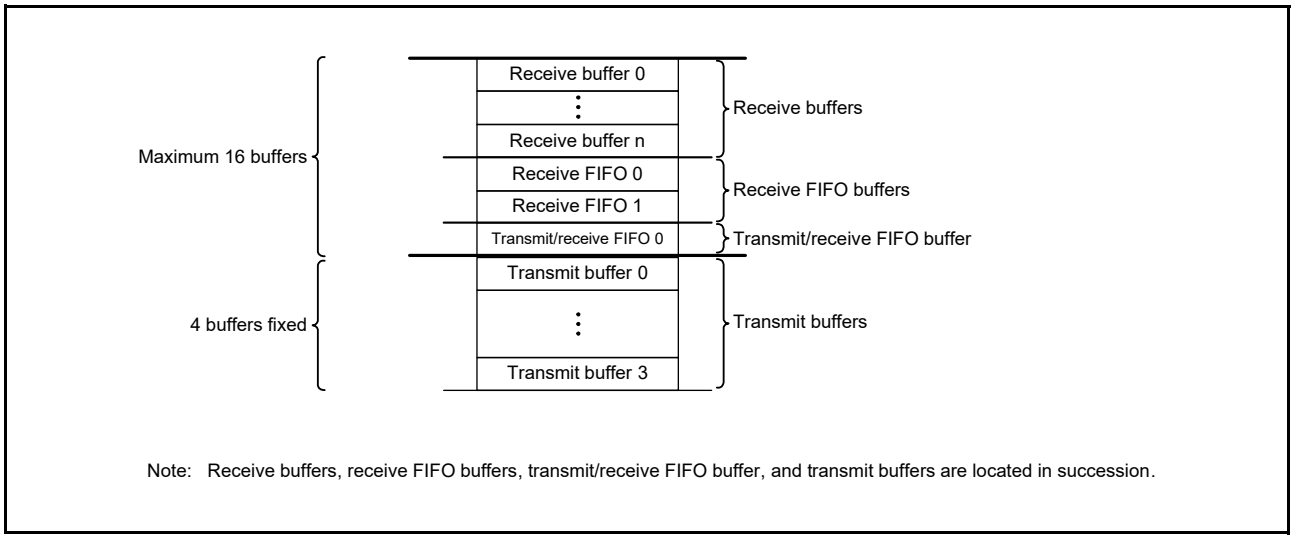


Figure 36.20 Buffer Configuration

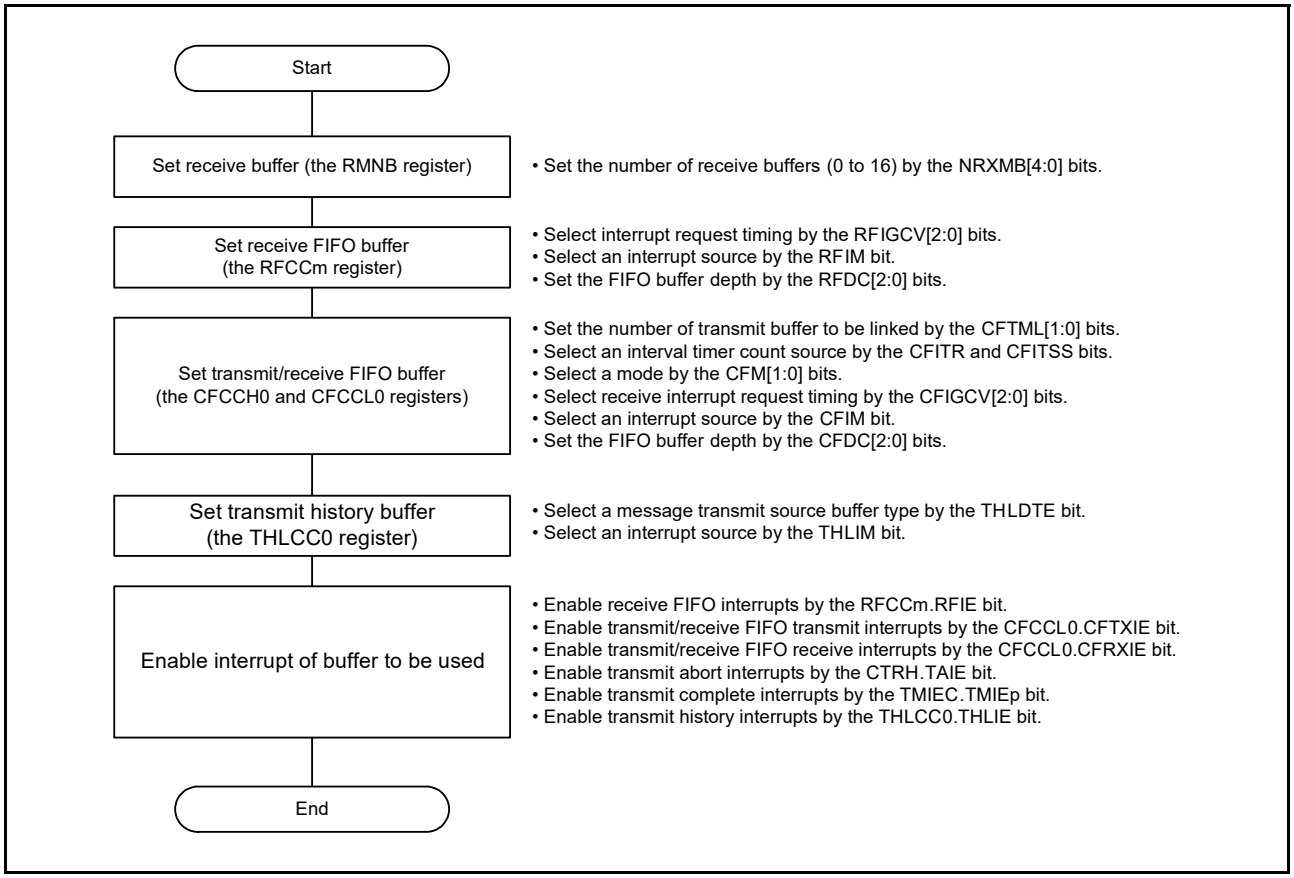


Figure 36.21 Buffer Setting Procedure

Downloaded from Arrow.com.

36.10 Reception Procedure

36.10.1 Receive Buffer Reading Procedure

When the processing to store received messages in a receive buffer starts, the RMND0.RMNSn flag becomes 1 (receive buffer n contains a new message). Messages can be read from the RMIDLn, RMIDHn, RMTSn, RMPTRn, and RMDF0n to RMDF3n registers. If the next message has been received before the current message is read from the receive buffer, the message is overwritten. Figure 36.22 shows the receive buffer reading procedure.

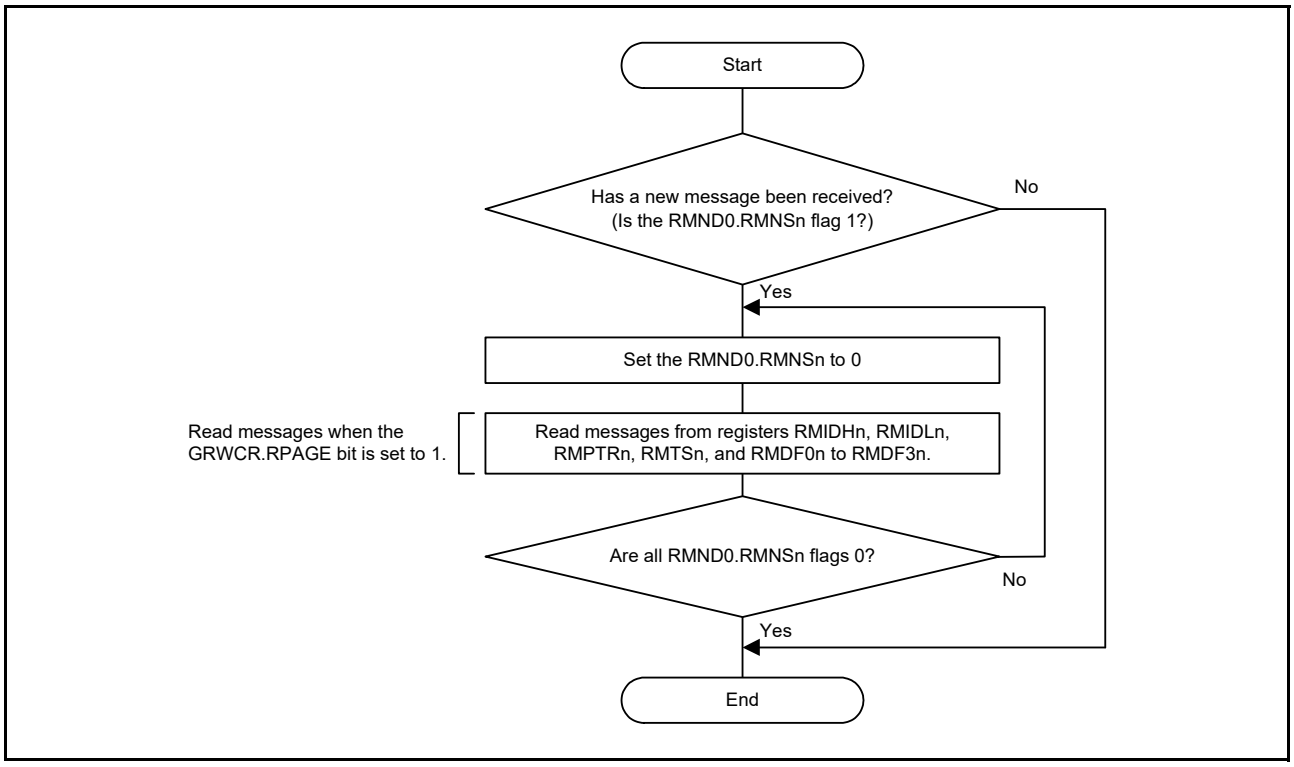


Figure 36.22 Receive Buffer Reading Procedure

Downloaded from Arrow.com.

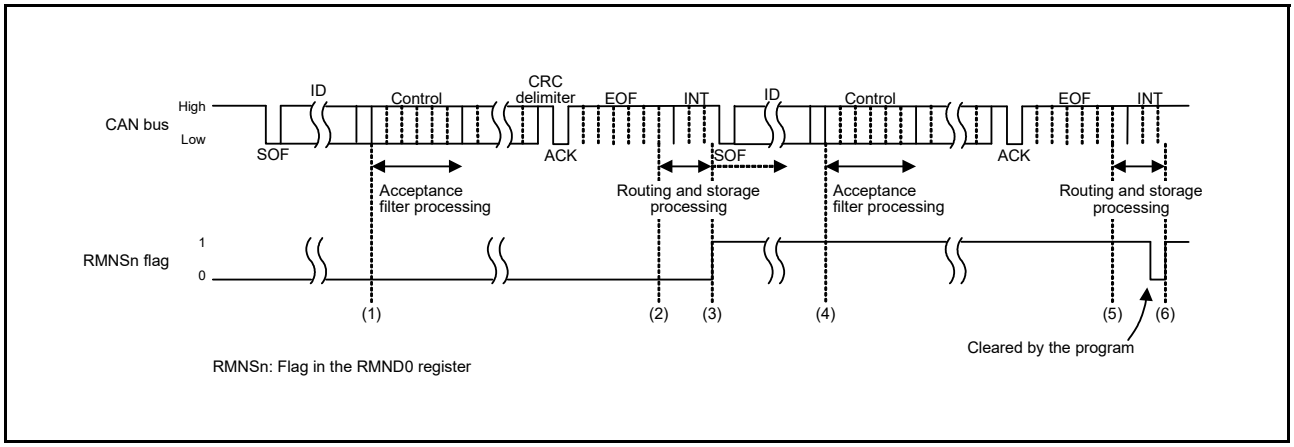


Figure 36.23 Receive Buffer Reception Timing Chart

(1) When the ID field in a message has been received, the acceptance filter processing starts.

(2) When the message matches the receive rule of the corresponding channel and the message has been successfully received, the routing processing to transfer the message to the specified buffer starts. When the GCFGL.DCE bit is set to 1 (DLC check is enabled), the DLC filter processing starts at this time.

(3) When the message has passed through the DLC filter processing, the processing to store the message in the specified receive buffer starts.
When the message storage processing starts, the corresponding RMND0.RMNSn flag becomes 1 (receive buffer contains a new message). If other channels are performing filter processing or transmit priority determination processing, the routing processing and the storage processing may be delayed.

(4) When the ID field of the next message has been received, the acceptance filter processing starts.

(5) When the message matches the receive rule of the corresponding channel and the message has been successfully received, the routing processing to transfer the message to the specified buffer starts. When the GCFGL.DCE bit is set to 1 (DLC check is enabled), the DLC filter processing starts at this time.

(6) When the corresponding RMND0.RMNSn flag becomes 0 (receive buffer contains no new message), this flag becomes 1 again when the message storage processing starts. Even if the RMND0.RMNSn flag remains 1, a new message is overwritten to the receive buffer. The RMND0.RMNSn flag should not be set to 0 during storage of messages.

Downloaded from Arrow.com.

36.10.2 FIFO Buffer Reading Procedure

When received messages have been stored in one or more receive FIFO buffers or a transmit/receive FIFO buffer that is set to receive mode, the corresponding message count display counter (RFSTSm.RFMC[5:0] flags or CFSTS0.CFMC[5:0] flags) is incremented. At this time, when the RFCCm.RFIE bit (receive FIFO interrupt is enabled) or the CFCCL0.CFRXIE bit (transmit/receive FIFO receive interrupt is enabled) is set to 1, an interrupt request is generated. Received messages can be read from the RFIDLm, RFIDHm, RFTSm, RFPTRm, and RFDF0m to RFDF3m registers (receive FIFO buffers) or the CFIDL0, CFIDH0, CFTS0, CFPTR0, and CFDF00 to CFDF30 registers (transmit/receive FIFO buffers). Messages in FIFO buffers can be read sequentially on a first-in, first-out basis.
When the message count display counter value matches the FIFO buffer depth (a value set by the RFCCm.RFDC[2:0] bits or the CFCCL0.CFDC[2:0] bits), the RFFLL or CFFLL flag becomes 1 (the receive FIFO buffer is full).
When all messages have been read out of the FIFO buffer, the RFSTSm.RFEMP flag or CFSTS0.CFEMP flag becomes 1 (the receive FIFO buffer contains no unread message (buffer empty)).
If the RFCCm.RFE bit or the CFCCL0.CFE bit is set to 0 (no receive FIFO buffer is used) with the interrupt request flag (RFSTSm.RFIF flag or CFSTS0.CFRXIF flag) set to 1 (a receive FIFO interrupt request is present), the interrupt request flag is not automatically set to 0. Set the interrupt request flag to 0 by the program.

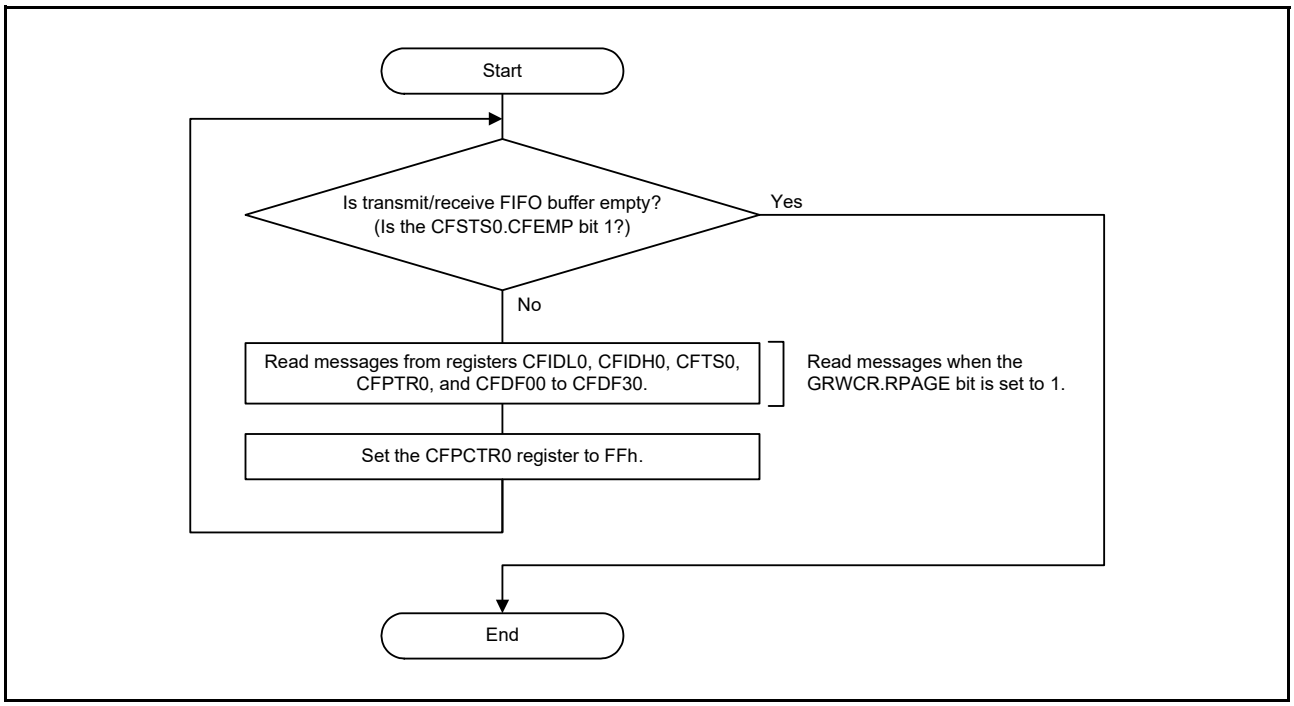


Figure 36.24 Transmit/Receive FIFO Buffer Reading Procedure

Downloaded from Arrow.com.

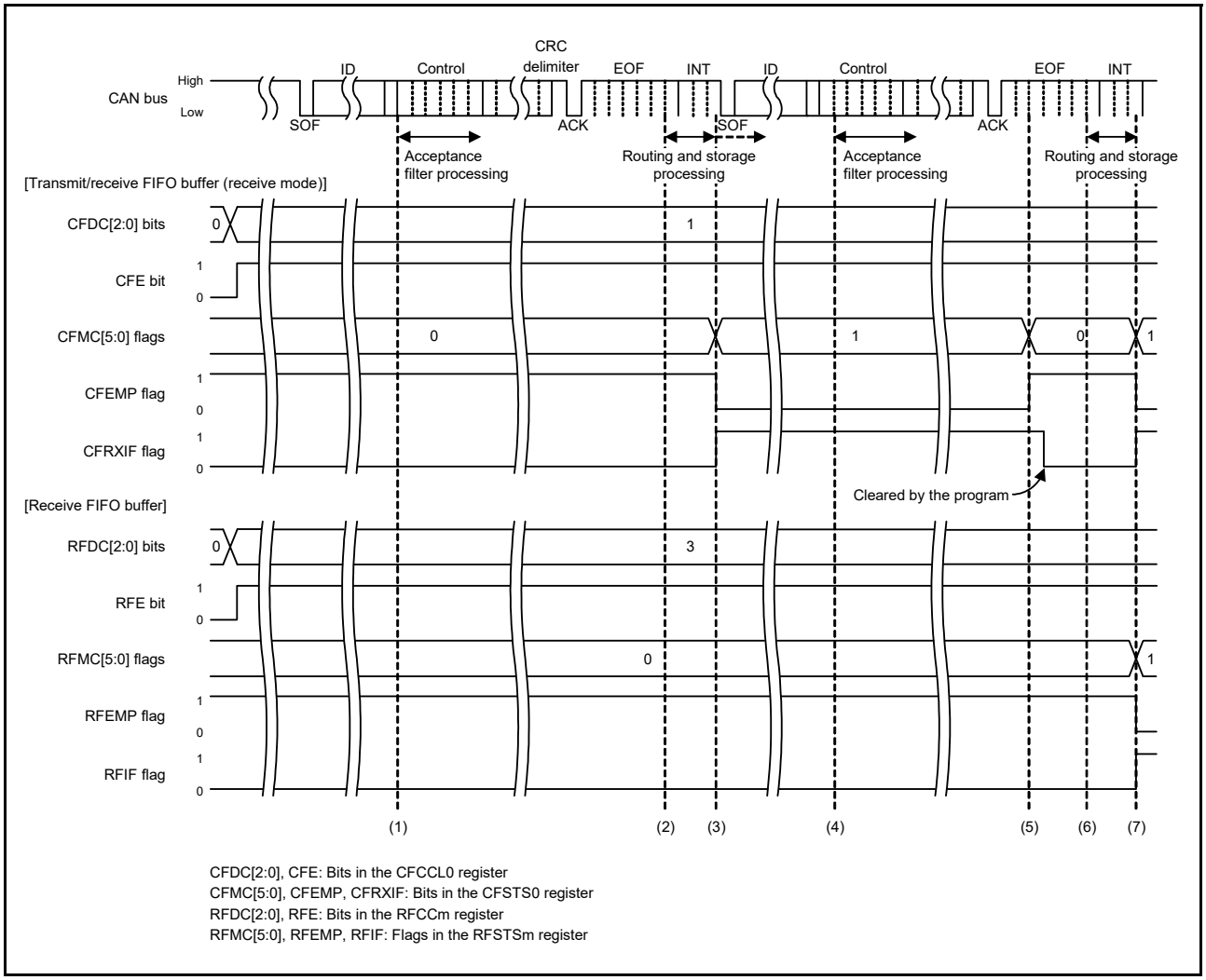


Figure 36.25 FIFO Buffer Reception Timing Chart

(1) When the ID field in a message has been received, the acceptance filter processing starts.

(2) When the message matches the receive rule of the corresponding channel and the message has been successfully received, the routing processing to transfer the message to the specified buffer starts. When the GCFGL.DCE bit is set to 1 (DLC check is enabled), the DLC filter processing starts at this time.

(3) When the message has passed through the DLC filter processing and the CFCCL0.CFE value is 1 (transmit/receive FIFO buffers are used) and the CFCCL0.CFDC[2:0] value is 001b or more, the message is stored in the transmit/receive FIFO buffer that is set to receive mode. The CFSTS0.CFMC[5:0] value is incremented and becomes 01h. When the CFCCL0.CFIM bit is set to 1 (a FIFO receive interrupt request is generated each time a message has been received), the CFSTS0.CFRXIF flag becomes 1 (a transmit/receive FIFO receive interrupt request is present). The CFSTS0.CFRXIF flag can be reset to 0 by the program.

(4) When the ID field of the next message has been received, the acceptance filter processing starts.

(5) Read received messages from the CFIDL0, CFIDH0, CFTS0, CFPTR0, and CFDF00 to CFDF30 registers and write FFh to the CFPCTR0 register. Thereby the CFSTS0.CFMC[5:0] flags are decremented and become 00h, and the CFSTS0.CFEMP flag becomes 1 (the transmit/receive FIFO buffer contains no message (buffer empty)).

(6) When the message matches the receive rule of the corresponding channel and the message has been successfully received, the routing processing to transfer the message to the specified buffer starts. When the GCFGL.DCE bit is set to 1 (DLC check is enabled), the DLC filter processing starts at this time.

(7) The message is stored in the transmit/receive FIFO buffer set in receive mode, when the message has passed through the DLC filter process if the CFCCL0.CFE bit is set to 1 (transmit/receive FIFO buffers are used) and the

Downloaded from Arrow.com.

CFCCL0.CFDC[2:0] bits are set to 001b or more.
The CFSTS0.CFMC[5:0] value is incremented to 01h. When the CFCCL0.CFIM bit is set to 1 (an interrupt occurs each time a message has been received), the CFSTS0.CFRXIF flag becomes 1 (a transmit/receive FIFO receive interrupt request is present).
The message is stored in the receive FIFO buffer, if the RFCCm.RFE bit is set to 1 (receive FIFO buffers are used) and RFCCm.RFDC[2:0] bits are set to 001b or more. The RFSTSm.RFMC[5:0] value is incremented to 01h. When the RFCCm.RFIM bit is set to 1 (an interrupt occurs each time a message has been received), the RFSTSm.RFIF flag becomes 1 (a receive FIFO interrupt request is present).

36.11 Transmission Procedure

36.11.1 Procedure for Transmission from Transmit Buffers

Figure 36.26 shows the procedure for transmission from transmit buffers.
Figure 36.27 shows a timing chart where messages are transmitted from two transmit buffers and transmission has been successfully completed. Figure 36.28 shows a timing chart where messages are transmitted from two transmit buffers and transmit abort has been completed.

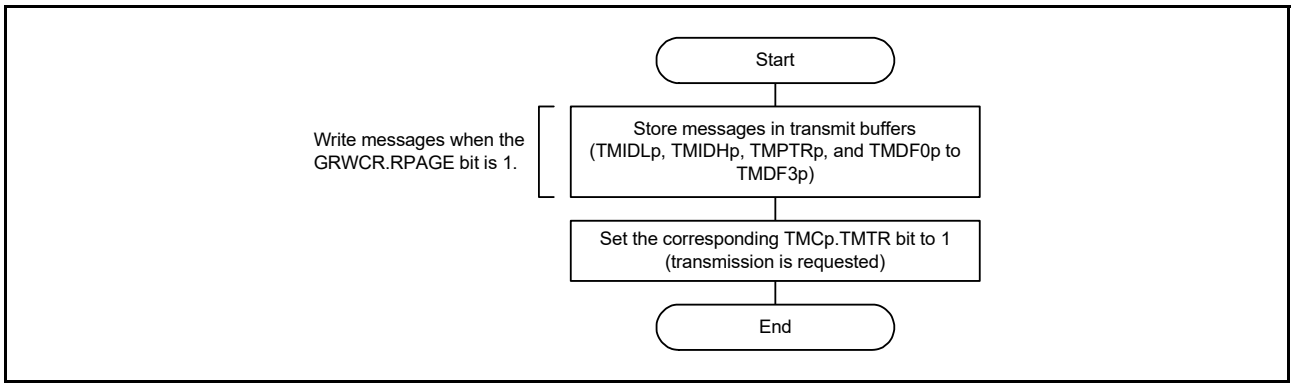


Figure 36.26 Procedure for Transmission from Transmit Buffers

Downloaded from Arrow.com.

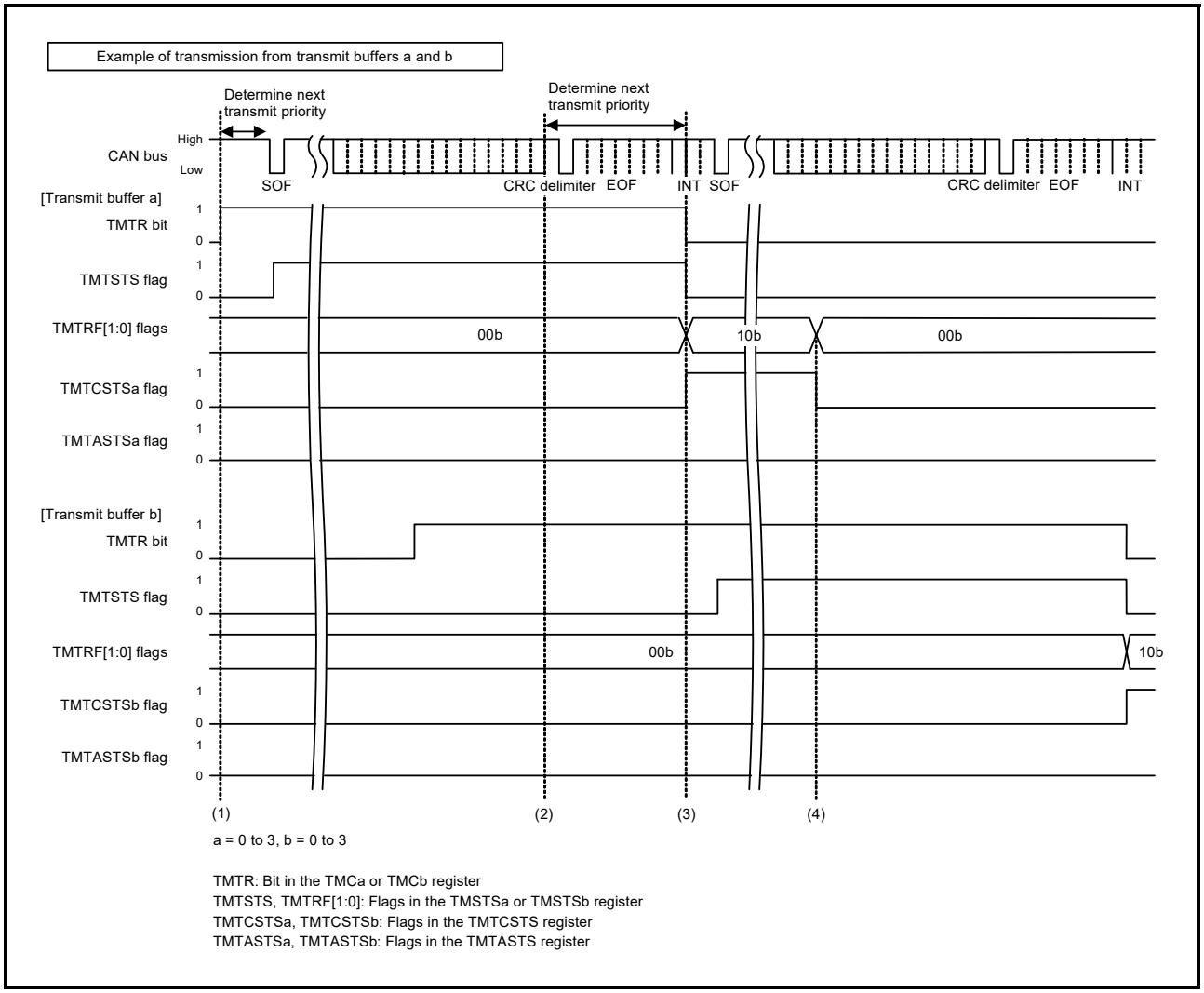


Figure 36.27 Transmit Buffer Transmission Timing Chart (Transmission Completed Successfully)

(1) When the TMCa.TMTR bit (a = 0 to 3) is set to 1 while the CAN bus is idle, the transmit priority determination processing starts to determine the highest-priority transmit buffer. If transmit buffer a is determined to be the highest-priority transmit buffer, the corresponding TMSTSa.TMTSTS flag becomes 1 (transmission is in progress) and the CAN channel starts transmitting data.

(2) When a transmit request from a buffer is present, the priority determination starts with the CRC delimiter for the next transmission.

(3) When transmit completes successfully, the TMSTSa.TMTRF[1:0] flags become 10b (transmission has been completed (without transmit abort request)), the TMSTSa.TMTSTS flag and the TMCa.TMTR bit become 0, and the TMTCSTS.TMTCSTSa flag becomes 1. When the TMIEC.TMIEa value is 1 (transmit buffer interrupt is enabled), a transmit interrupt request is generated. To clear the interrupt request, set the TMSTSa.TMTRF[1:0] flags to 00b (transmission is in progress or no transmit request is present).

(4) Before starting the next transmission, set the TMSTSa.TMTRF[1:0] flags to 00b. Write the next message to the transmit buffer, and then set the TMCa.TMTR bit to 1 (transmission is requested). The TMCa.TMTR bit can be set to 1 only when the TMSTSa.TMTRF[1:0] flag value is 00b.
If an arbitration lost has occurred after transmission is started, the TMSTSa.TMTSTS flag becomes 0. The transmit priority determination is reexecuted at the beginning of the CRC delimiter to search the highest-priority transmit buffer. If an error has occurred during transmission or after arbitration lost, the priority determination processing is reexecuted during transmission of an error frame.

Downloaded from Arrow.com.

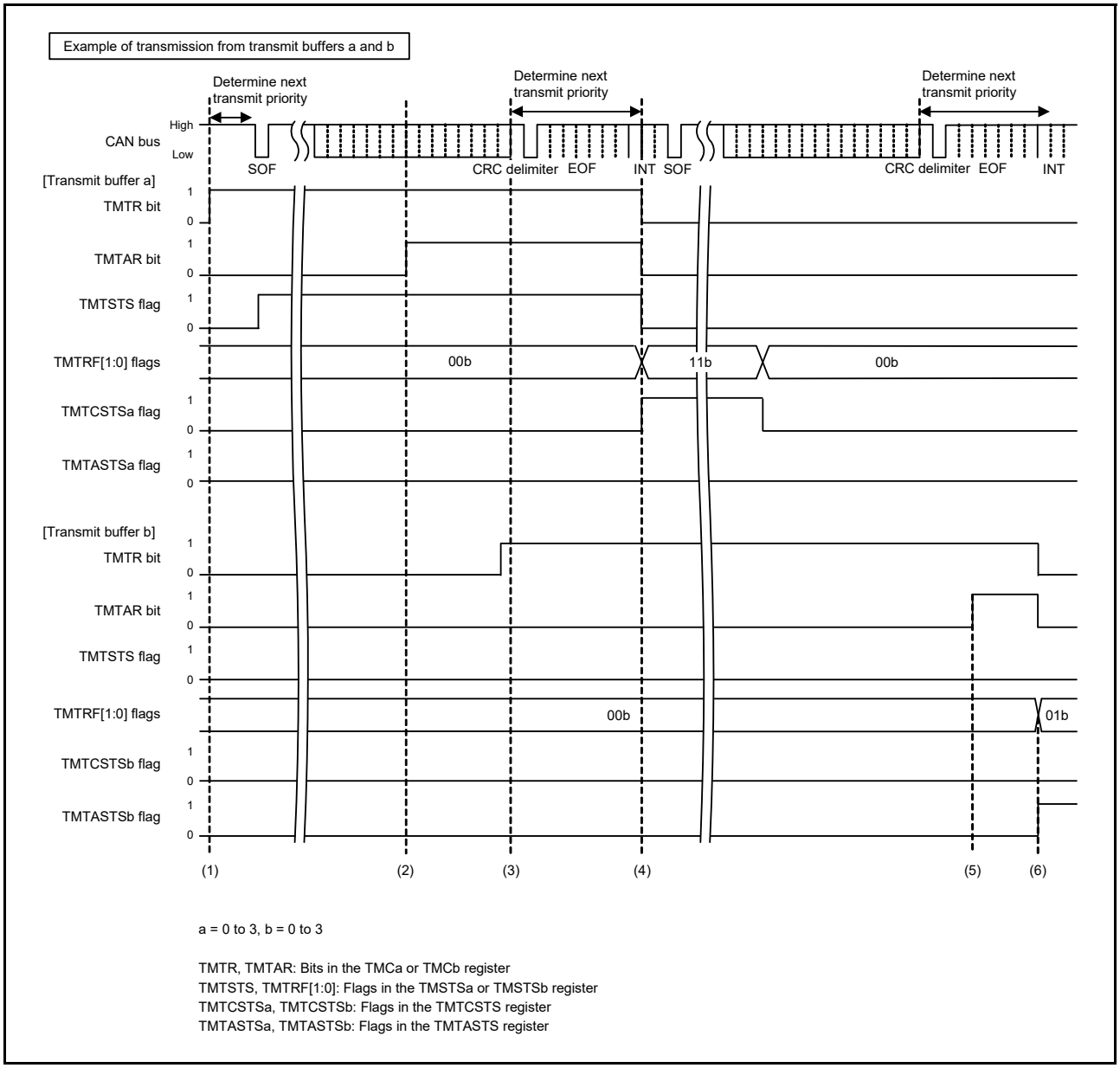


Figure 36.28 Transmit Buffer Transmission Timing Chart (Transmit Abort Completed)

(1) When the TMCa.TMTR bit (a = 0 to 3) is set to 1 while the CAN bus is idle, the transmit priority determination processing starts to determine the highest-priority transmit buffer. If transmit buffer a is determined to be the highest-priority transmit buffer, the corresponding TMSTSa.TMTSTS flag becomes 1 (transmission is in progress) and the CAN channel starts transmitting data.

(2) When it is determined that the transmit buffer is used for the next transmission or transmission is in progress, message transmission is not aborted unless an error or arbitration lost occurs even if the TMCa.TMTAR bit is set to 1 (transmit abort is requested).

(3) The priority determination starts with the CRC delimiter for the next transmission. In this timing chart, buffer b is not selected as the next transmit buffer.

(4) When transmit completes successfully, the TMSTSa.TMTRF[1:0] flags become 11b (transmission has been completed (with transmit abort request)), the TMSTSa.TMTSTS flag and the TMCa.TMTR bit become 0, and the TMTCSTS.TMTCSTSa flag becomes 1.
When the TMIEC.TMIEa value is 1 (transmit buffer interrupt is enabled), a transmit interrupt request is generated.

Downloaded from Arrow.com.

To clear the interrupt request, set the TMSTSa.TMTRF[1:0] flags to 00b (transmission is in progress or no transmit request is present).

(5) While another CAN node is transmitting data on the CAN bus (TMSTSa.TMTSTS flag = 0), if the TMCa.TMTAR bit is set to 1 while the corresponding channel is determining transmit priority, the TMCa.TMTR bit cannot be set to 0.

(6) After the internal processing time has passed, the transmission is terminated and the TMSTSa.TMTRF[1:0] flags become 01b and the TMTASTS.TMTASTSa flag becomes 1. When the transmit buffer is not transmitting data and is not selected as the next transmit buffer and priority determination is not being made, an abort request is immediately accepted and the TMSTSa.TMTRF flag becomes 01b. At this time, the TMCa.TMTR and TMTAR bits become 0.
When transmit abort is completed with the CTRH.TAIE bit set to 1 (transmit abort interrupt is enabled), an interrupt request is generated. To clear the interrupt request, set the TMSTSa.TMTRF[1:0] flags to 00b.

If an arbitration lost has occurred after the CAN channel started transmission, the TMSTSa.TMTSTS flag becomes 0. The transmit priority determination is reexecuted at the beginning of the CRC delimiter to search the highest-priority transmit buffer. If an error has occurred during transmission or after arbitration lost, the priority determination processing is reexecuted during transmission of an error frame.

36.11.2 Procedure for Transmission from Transmit/Receive FIFO Buffers

Figure 36.29 shows the procedure for transmission from transmit/receive FIFO buffers.
Figure 36.30 shows a timing chart where messages are transmitted from the transmit/receive FIFO buffers and transmission has been successfully completed. Figure 36.31 shows a timing chart where messages are transmitted from the transmit/receive FIFO buffers and transmit abort has been completed.

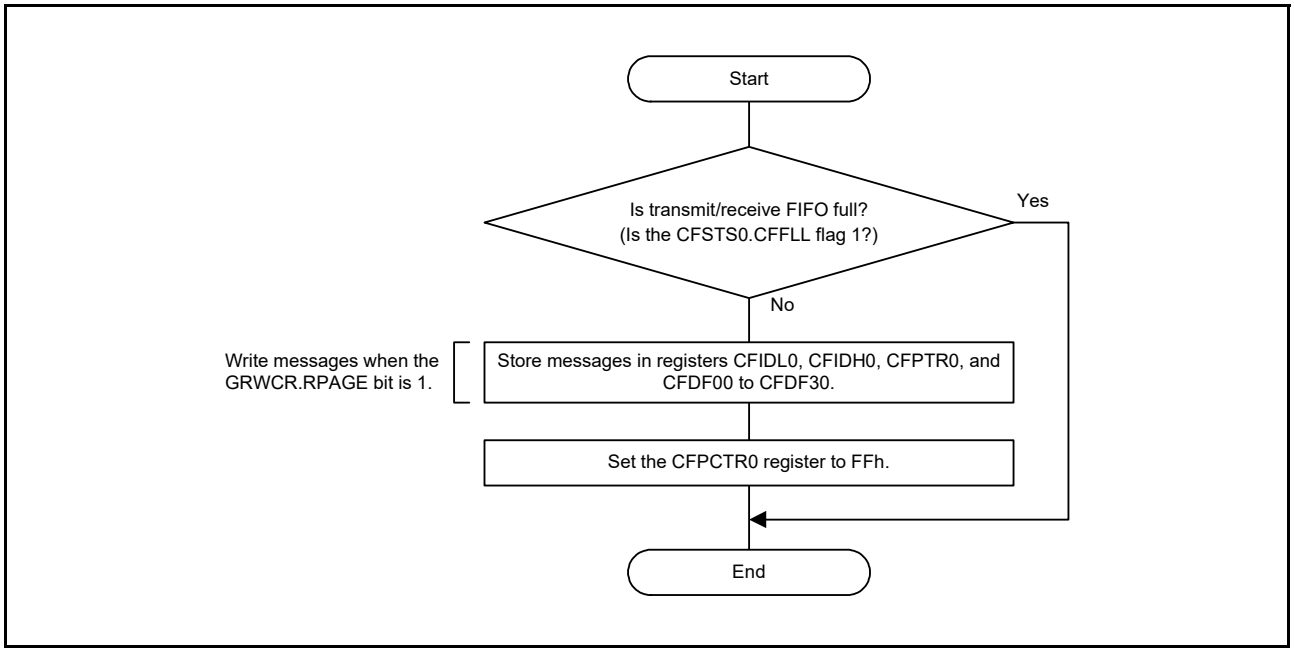


Figure 36.29 Procedure for Transmission from Transmit/Receive FIFO Buffers

Downloaded from Arrow.com.

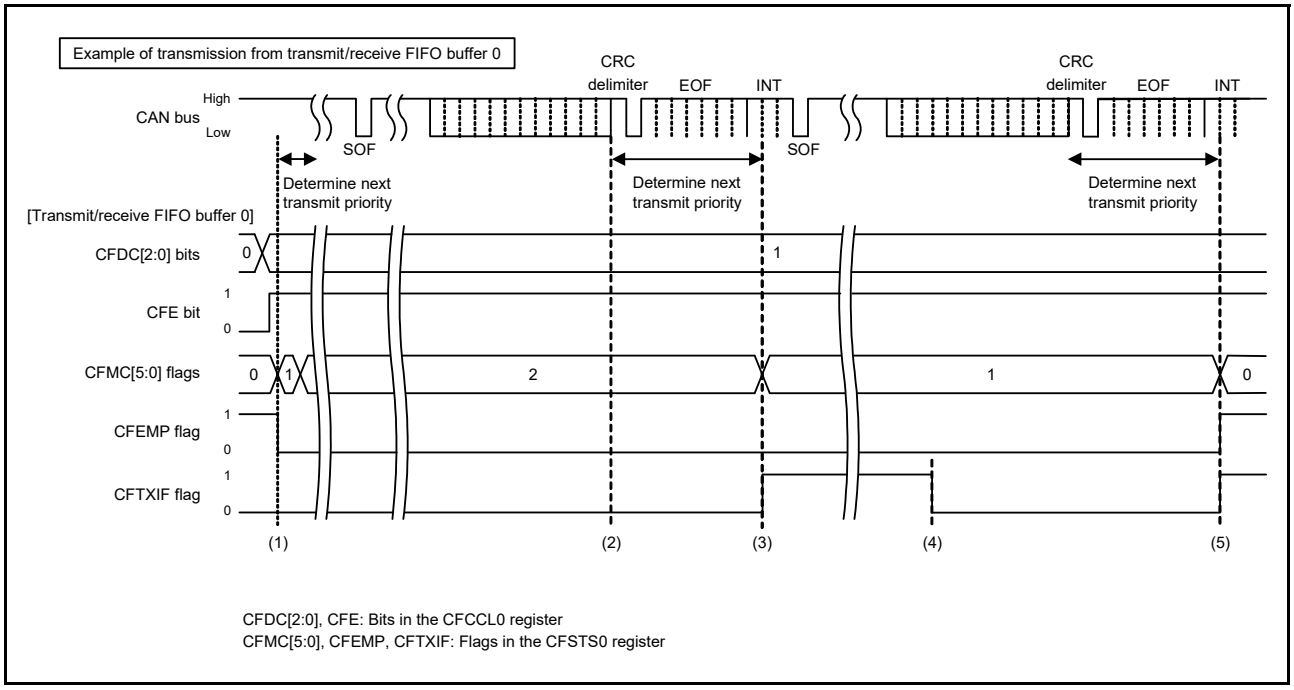


Figure 36.30 Transmit/Receive FIFO Buffer Transmission Timing Chart (Transmission Completed Successfully)

(1) While the CAN bus is idle, when the CFCCL0.CFE value is 1 (transmit/receive FIFO buffer 0 is used) and the CFCCL0.CFDC[2:0] value is 001b (4 messages) or more and the CFSTS0.CFMC[5:0] value is 01h or more, the priority determination processing starts to determine the highest-priority transmit message. When the highest-priority transmit message has been determined, transmission of the message starts.

(2) When a transmit request from a buffer is present, the priority determination starts with the CRC delimiter for the next transmission.

(3) When transmit completes successfully, the CFSTS0.CFMC[5:0] value is decremented. Setting the CFCCL0.CFIM bit to 1 (a FIFO transmit interrupt request is generated each time a message has been transmitted) sets the CFSTS0.CFTXIF flag to 1 (a transmit/receive FIFO transmit interrupt request is present).

(4) The CFSTS0.CFTXIF flag can be cleared by the program.

(5) Message transmission from transmit/receive FIFO buffer 0 has been completed and the CFSTS0.CFMC[5:0] value is decremented. The CFSTS0.CFMC[5:0] flags become 00h and therefore the CFSTS0.CFEMP flag becomes 1 (the transmit/receive FIFO buffer contains no message (buffer empty)).
Transmission is continued until the CFSTS0.CFEMP flag becomes 1. It is possible to continuously store transmit messages in FIFO buffers until the CFSTS0.CFFLL flag becomes 1 (the transmit/receive FIFO buffer is full).

Downloaded from Arrow.com.

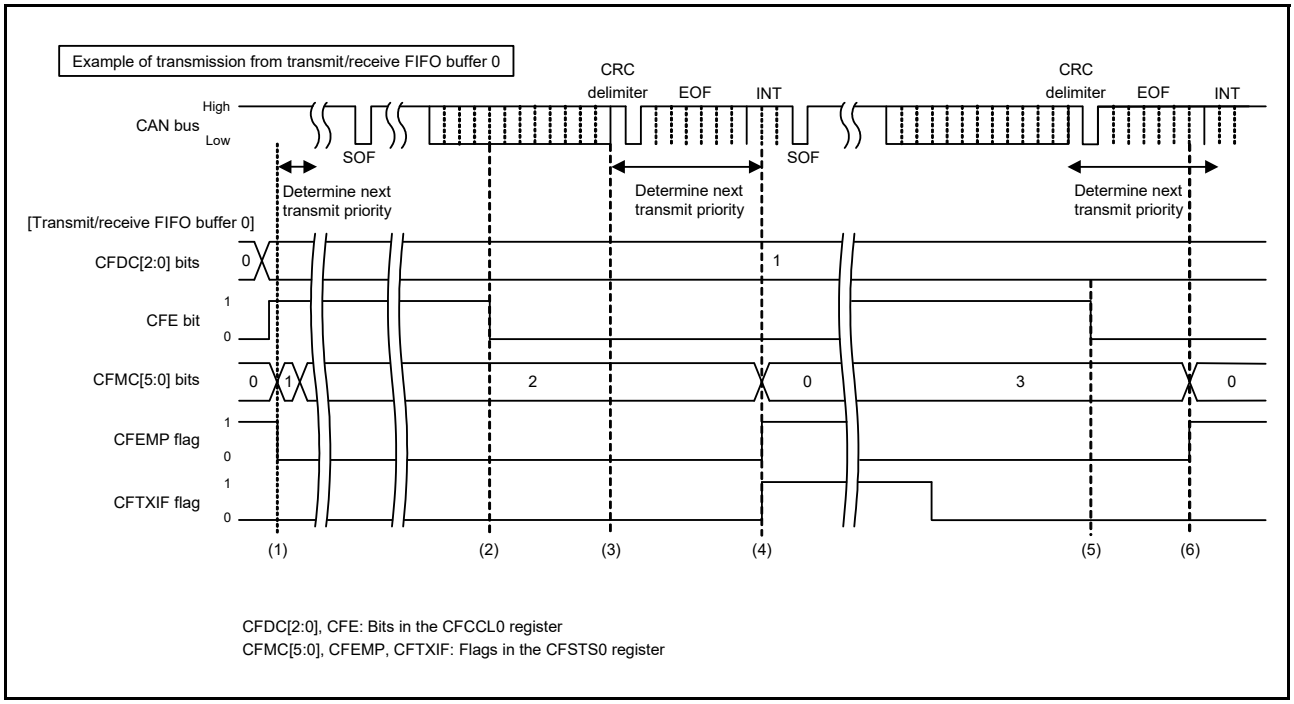


Figure 36.31 Transmit/Receive FIFO Buffer Transmission Timing Chart (Transmit Abort Completed)

(1) While the CAN bus is idle, when the CFCCL0.CFE value is 1 (transmit/receive FIFO buffer 0 is used) and the CFCCL0.CFDC[2:0] value is 001b (4 messages) or more and the CFSTS0.CFMC[5:0] value is 01h or more, the priority determination processing starts to determine the highest-priority transmit message. When the highest-priority transmit message has been determined, transmission of the message starts.

(2) When transmission is in progress or it is determined that the transmit/receive FIFO buffer is used for the next transmission, message transmission is not aborted unless an error or arbitration lost occurs even if the CFCCL0.CFE bit is set to 0 (no transmit/receive FIFO buffer 0 is used).

(3) When a transmit request from a buffer is present, the priority determination starts with the CRC delimiter for the next transmission. In this figure, transmit/receive FIFO buffer 0 is not selected as a buffer for the next transmission.

(4) When transmit completes successfully, the CFSTS0.CFMC[5:0] value becomes 00h. Setting the CFCCL0.CFIM bit to 1 (a FIFO transmit interrupt request is generated each time a message has been transmitted) sets the CFSTS0.CFTXIF flag to 1 (a transmit/receive FIFO transmit interrupt request is present). The CFSTS0.CFTXIF flag can be cleared by the program.

(5) If another CAN node on the CAN bus is transmitting data (not from transmit/receive FIFO buffer 0), transmit/receive FIFO buffer 0 cannot be disabled immediately even if the CFCCL0.CFE bit is set to 0 (no transmit/receive FIFO buffer 0 is used) during transmit priority determination. (The CFSTS0.CFEMP flag is not set to 1 (the transmit/receive FIFO buffer contains no message (buffer empty)) immediately.)

(6) After the internal processing time has passed, transmit/receive FIFO buffers are disabled and the CFSTS0.CFMC[5:0] flags become 00h and the CFSTS0.CFEMP flag becomes 1. When the transmit/receive FIFO buffer 0 is not transmitting data and is not selected as the next transmit buffer and priority determination is not in progress, the transmit/receive FIFO buffer 0 is immediately disabled. (The CFSTS0.CFMC[5:0] flags become 00h and the CFSTS0.CFEMP flag becomes 1.)

Downloaded from Arrow.com.

36.11.3 Transmit History Buffer Reading Procedure

Transmit history data can be read from the THLACC0 register. The next data can be accessed by writing FFh to the corresponding THLPCTR0 register after reading a set of data. Figure 36.32 shows the transmit history buffer reading procedure.

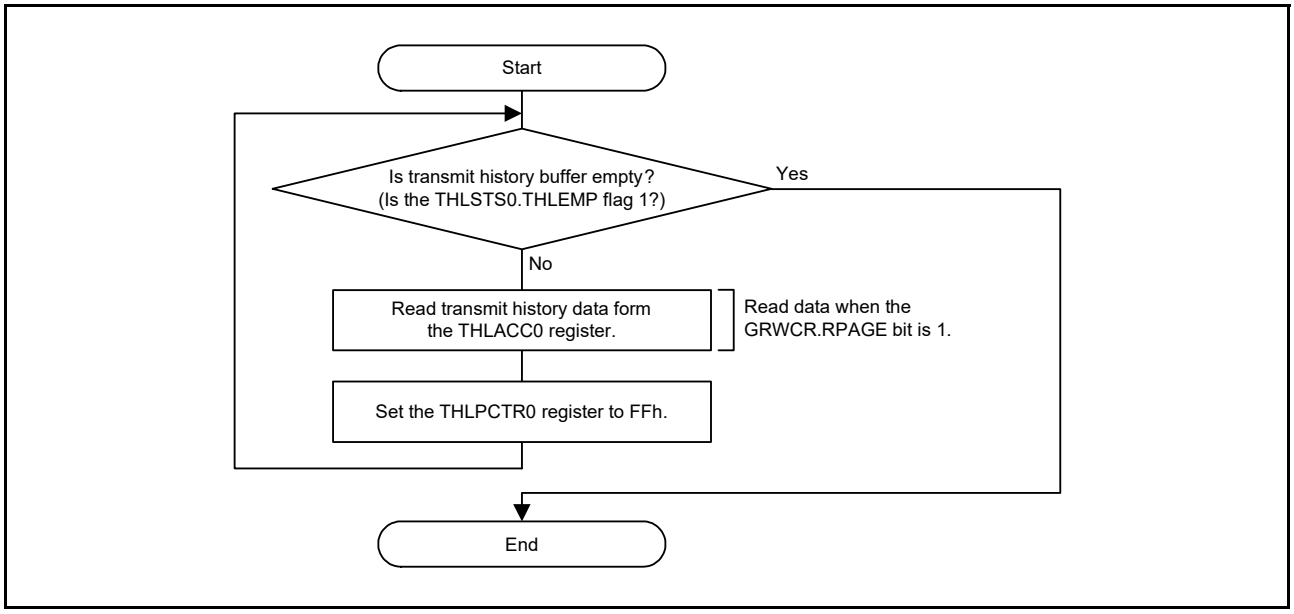


Figure 36.32 Transmit History Buffer Reading Procedure

Downloaded from Arrow.com.

36.12 Test Settings

36.12.1 Self-Test Mode Setting Procedure

Self-test mode allows communication test on a channel basis by receiving messages transmitted from the own node. Figure 36.33 shows the self-test mode setting procedure.

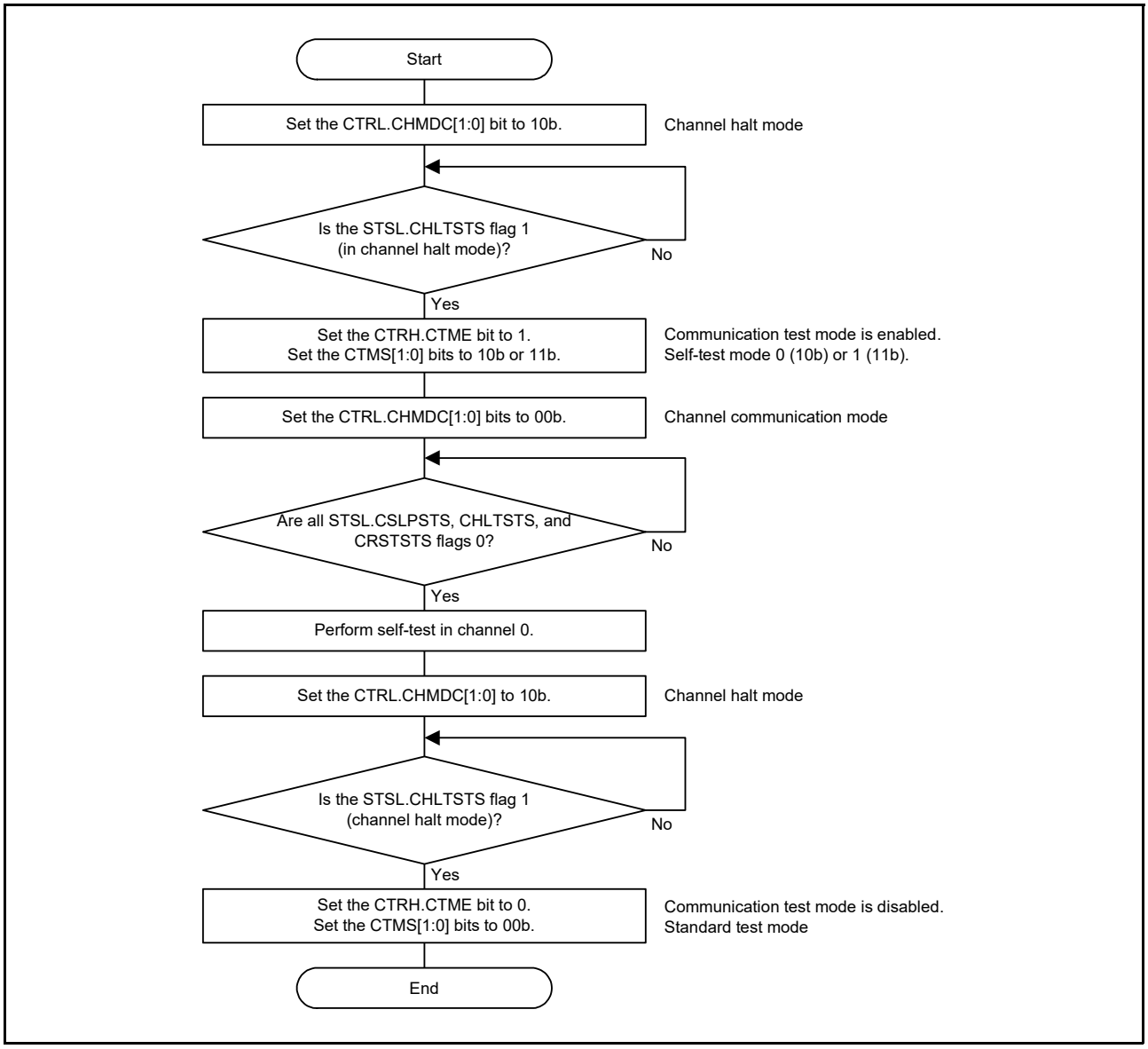


Figure 36.33 Self-Test Mode Setting Procedure

Downloaded from Arrow.com.

36.12.2 Protection Unlock Procedure

Since the global test functions shown in Table 36.14 are protected, write unlock data 1 and unlock data 2 in succession to the GLOCKK.LOCK[15:0] bits, and then set each test function bit to 1.

Table 36.14 Protection Unlock Data for Test Functions

Test Function	Protection Unlock Data 1	Protection Unlock Data 2	Target Bit
RAM test	7575h	8A8Ah	GTSTCTRL.RTME bit

If an incorrect value has been written to the GLOCKK.LOCK[15:0] bits, retry the procedure above from writing of unlock data 1.
Figure 36.34 shows the protection unlock procedure.

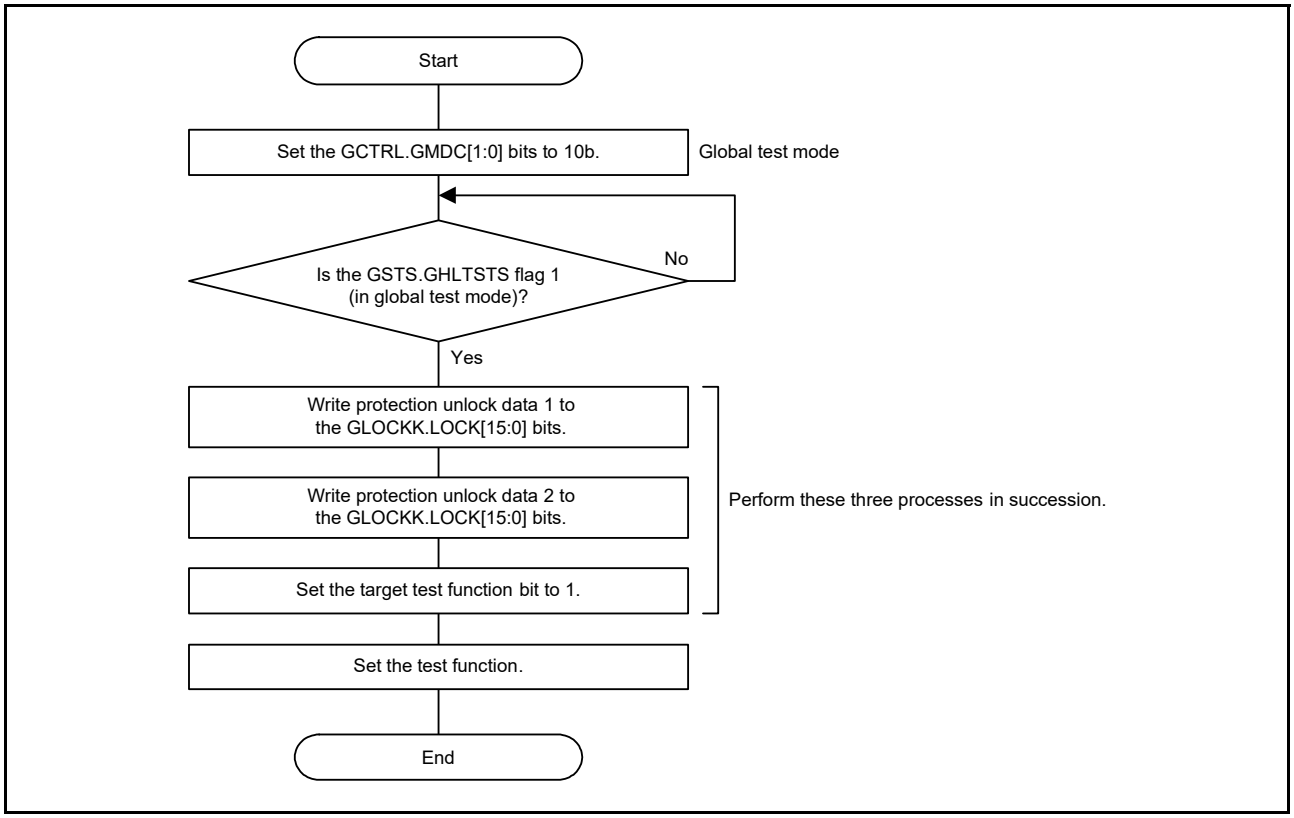


Figure 36.34 Protection Unlock Procedure

Downloaded from Arrow.com.

36.12.3 RAM Test Setting Procedure

RAM tests include CAN RAM read/write test. The read/write test verifies that data written to the RAM is read correctly. Before closing the RAM test, write 0000h to all pages of the CAN RAM.

Figure 36.35 shows the RAM test setting procedure.

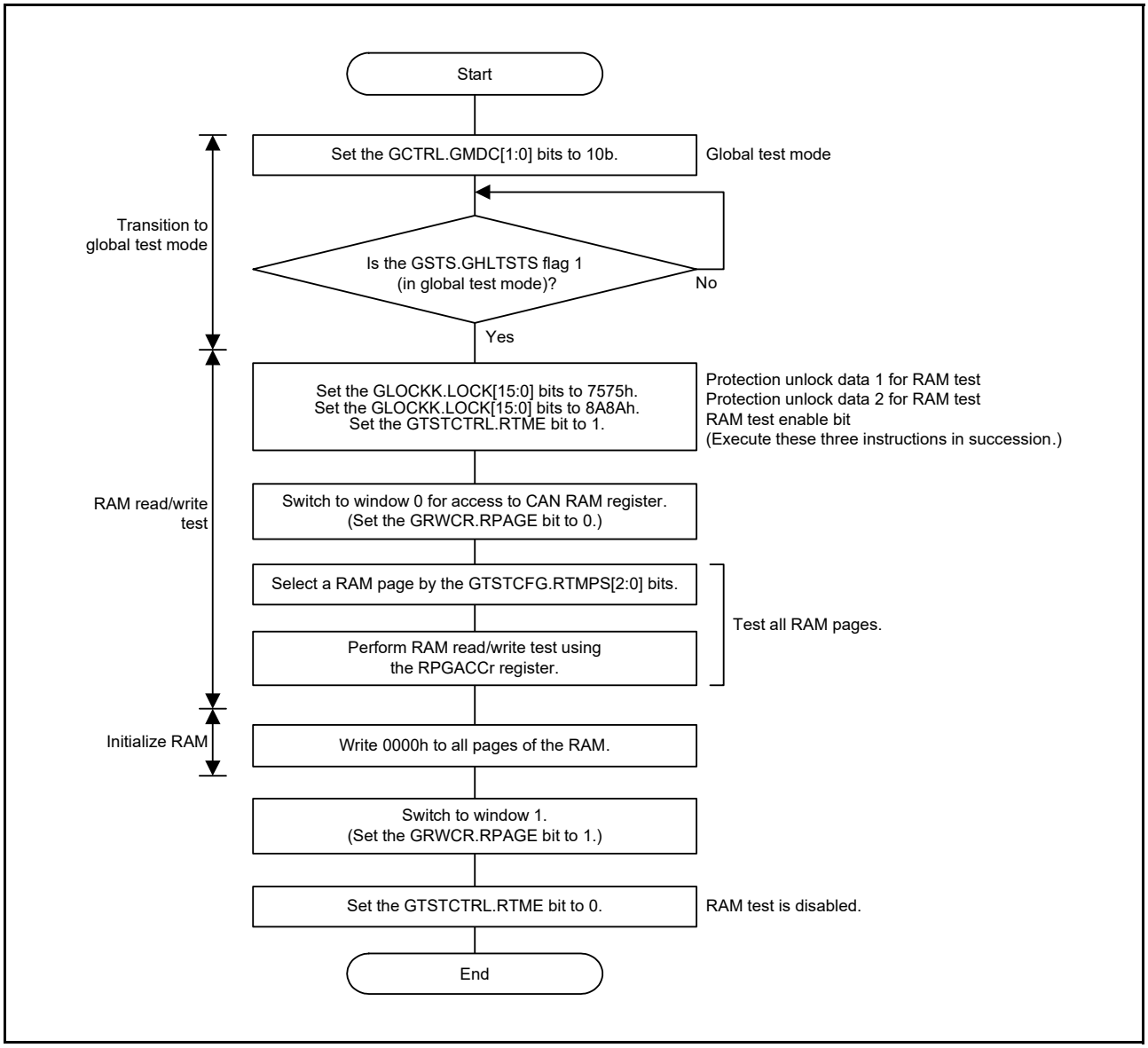


Figure 36.35 RAM Test Setting Procedure

Downloaded from Arrow.com.

36.13 Notes on the CAN Module

- When changing a global mode, check the GSTS.GSLPSTS, GHLTSTS, and GRSTSTS flags for transitions. When changing a channel mode, check the STSL.CSLPSTS, CHLTSTS, and CRSTSTS flags for transitions.
- The acceptance filter processing checks receive rules sequentially in ascending order from the minimum rule number. If the same ID, IDE bit, or RTR bit value is set for multiple receive rules, the minimum number of receive rule is used for the acceptance filter processing. If the message does not pass through the subsequent DLC filter processing, the data processing is terminated without returning to the acceptance filter processing and the message is not stored in the buffer.
- When linking transmit buffers to transmit/receive FIFO buffers, set the control register (TCMp) of the corresponding transmit buffer to 00h. The status register (TMSTSp) of the corresponding transmit buffer should not be used. Flags in other status registers (registers TMTRSTS, TMTCSTS, and TMTASTS), which correspond to transmit buffers linked to transmit/receive FIFO buffers remain unchanged. Set the enable bit in the corresponding interrupt enable register (the TMIEC register) to 0 (transmit buffer interrupt is disabled).
- When the CAN bit time clock is selected as a timestamp counter clock source, the timestamp counter stops when the corresponding channel has transitioned to channel reset mode or channel halt mode.
- In case of an attempt to store a new receive message when the receive FIFO buffer and the transmit/receive FIFO buffer are full, the new message is discarded. If you wish to store a new transmit message in the transmit/receive FIFO buffer, check that the transmit/receive FIFO buffer is not full.
- Since an interrupt request flag in the CAN module is not automatically set to 0 when an interrupt is accepted, the flags must be set to 0 by software. After the corresponding interrupt request flag has been set to 1, an interrupt is not generated even if an interrupt source condition is satisfied.
- In order to generate the CAN related interrupt that several interrupt sources are gathered, the following condition should be met:
  All interrupt request flags corresponding to these interrupt sources in the CAN module are set to 0 (note that this only applies to those interrupt request flags for which the corresponding interrupt enable bits shown in Table 36.11 are set to 1).
- The values of unused receive buffer registers (RMIDLn, RMIDHn, RMTSn, RMPTRn, and RMDF0n to RMDF3n (n = 0 to 15)), receive FIFO access registers (RFIDLm, RFIDHm, RFTSm, RFPTRm, and RFDF0m to RFDF3m (m = 0, 1)), and transmit/receive FIFO access registers (CFIDL0, CFIDH0, CFTS0, CFPTR0, and CFDF00 to CFDF30) become undefined once the CAN module exits from global reset mode and enters global operating mode or global test mode.

Downloaded from Arrow.com.

37. Serial Sound Interface (SSI)

This MCU integrates one channel of the serial sound interface (SSI) compliant with the I^{2}S bus specification. The SSI supports I^{2}S data format and MSB-first and left-justified/right-justified formats, so it can be used to send or receive audio data with various devices.

37.1 Overview

Table 37.1 SSI Specifications

Item	Specifications
Number of channels	One channel (SSI0)
Operating mode	Non-compressed mode
Transmission formats	• I^{2}S format supported • MSB-first supported Left-justified/right-justified formats selectable
Function	• Serves as both a transmitter and a receiver Channel 0 supports full-duplex communications. • Capable of various audio formats • SSISCK0 (serial bit clock) can be selected from among 16, 32, 48, and 64 fs (fs: Sampling rate) • The master clock (MCLK) can be selected from either of the following: Master clock pin for audio (AUDIO_MCLK): 1 to 25 MHz Main clock • Includes 8-stage FIFO buffers in transmitter and receiver • Capable of selecting whether to stop word select (SSIWS0) or not when data transmission is stopped
Interrupt sources	Three sources • Communication error Transmit underflow, transmit overflow, receive underflow, receive overflow, and idle • Receive data full • Transmit data empty
Low power consumption function	Module stop state can be set.

Downloaded from Arrow.com.

Figure 37.1 shows a block diagram of SSI (SSI0).

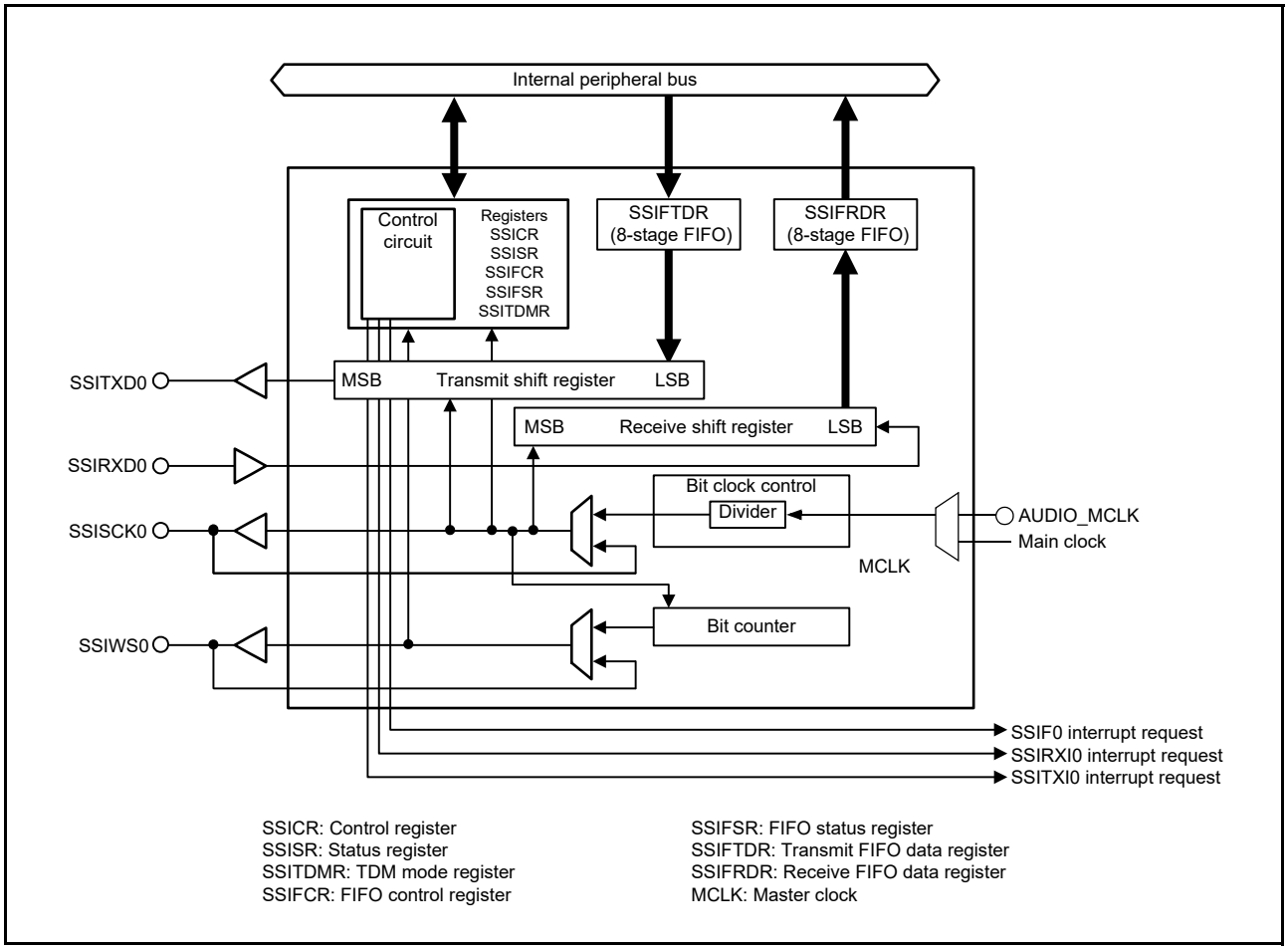


Figure 37.1 Block Diagram of SSI (SSI0)

Table 37.2 lists the I/O pins of the SSI.

Table 37.2 SSI I/O Pins

Channel	Pin Name	I/O	Description
SSI0	SSISCK0	I/O	Serial bit clock pin
	SSIWS0	I/O	Word selection pin
	SSITXD0	Output	Serial data output pin
	SSIRXD0	Input	Serial data input pin
	AUDIO_MCLK	Input	Master clock for audio pin (input master clock)

Downloaded from Arrow.com.

37.2 Register Descriptions

37.2.1 Control Register (SSICR)

Address(es): SSI0.SSICR 0008 A500h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	CKS	TUIEN	TOIEN	RUIEN	ROIEN	IIEN	—	CHNL[1:0]		DWL[2:0]			SWL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SCKD	SWSD	SCKP	SWSP	SPDP	SDTA	PDTA	DEL	CKDV[3:0]				MUEN	—	TEN	REN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	REN	Receive Enable	0: Disables receive operation. 1: Enables receive operation.	R/W
b1	TEN	Transmit Enable	0: Disables transmit operation. 1: Enables transmit operation.	R/W
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3	MUEN	Mute Enable*1	0: Not muted. 1: Muted.	R/W
b7 to b4	CKDV[3:0]	Serial Bit Clock Frequency Setting*3	b7 b6 b5 b4 0 0 0 0: MCLK 0 0 0 1: MCLK/2 0 0 1 0: MCLK/4 0 0 1 1: MCLK/8 0 1 0 0: MCLK/16 0 1 0 1: MCLK/32 0 1 1 0: MCLK/64 0 1 1 1: MCLK/128 1 0 0 0: MCLK/6 1 0 0 1: MCLK/12 1 0 1 0: MCLK/24 1 0 1 1: MCLK/48 1 1 0 0: MCLK/96 Settings other than above are prohibited.	R/W
b8	DEL	Serial Data Delay*3	0: I²S format compatibility One clock cycle delay between SSIWS0 and SSITXD0/SSIRXD0 1: MSB-first left-justified/right-justified format compatibility No delay between SSIWS0 and SSITXD0/SSIRXD0	R/W
b9	PDTA	Parallel Data Allocation*3	When data word length is 8 or 16 bits: 0: The lower bits of parallel data (SSIFTDR, SSIFRDR) are transferred prior to the upper bits. 1: The upper bits of parallel data (SSIFTDR, SSIFRDR) are transferred prior to the lower bits. When data word length is 18, 20, 22, or 24 bits: 0: Parallel data (SSIFTDR, SSIFRDR) is left-justified. 1: Parallel data (SSIFTDR, SSIFRDR) is right-justified.	R/W
b10	SDTA	Serial Data Alignment*3	0: Transmitting and receiving in the order of serial data and padding bits 1: Transmitting and receiving in the order of padding bits and serial data	R/W
b11	SPDP	Serial Padding Polarity*3	0: Padding data is 0. 1: Padding data is 1.	R/W

Downloaded from Arrow.com.

Bit	Symbol	Bit Name	Description	R/W
b12	SWSP	Word Select Polarity	0: SSIWS0 is low for the 1st system word, high for the 2nd system word. 1: SSIWS0 is high for the 1st system word, low for the 2nd system word.	R/W
b13	SCKP	Serial Bit Clock Polarity*3	0: SSIWS0 and SSITXD0 change at the SSISCK0 falling edge (SSIWS0 and SSIRXD0 are sampled at the SSISCK0 rising edge). 1: SSIWS0 and SSITXD0 change at the SSISCK0 rising edge (SSIWS0 and SSITXD0 are sampled at the SSISCK0 falling edge).	R/W
b14	SWSD	Word Select Direction*2, *3	0: SSIWS0 pin is input (slave mode). 1: SSIWS0 pin is output (master mode).	R/W
b15	SCKD	Serial Bit Clock Direction*2, *3	0: SSISCK0 pin is input (slave mode). 1: SSISCK0 pin is output (master mode).	R/W
b18 to b16	SWL[2:0]	System Word Length*3	Set the system word length to the bit clock frequency/2 fs. b18 b16 0 0 0: 8 bits (serial bit clock frequency = 16 fs) 0 0 1: 6 bits (serial bit clock frequency = 32 fs) 0 1 0: 24 bits (serial bit clock frequency = 48 fs) 0 1 1: 32 bits (serial bit clock frequency = 64 fs) Settings other than above are prohibited.	R/W
b21 to b19	DWL[2:0]	Data Word Length*3	b21 b19 0 0 0: 8 bits 0 0 1: 16 bits 0 1 0: 18 bits 0 1 1: 20 bits 1 0 0: 22 bits 1 0 1: 24 bits Settings other than above are prohibited.	R/W
b23, b22	CHNL[1:0]	Channels*3	b23 b22 0 0: One channel Settings other than above are prohibited.	R/W
b24	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b25	IIEN	Idle Interrupt Enable	0: Disables an idle interrupt. 1: Enables an idle interrupt.	R/W
b26	ROIEN	Receive Overflow Interrupt Enable	0: Disables a receive overflow interrupt. 1: Enables a receive overflow interrupt.	R/W
b27	RUIEN	Receive Underflow Interrupt Enable	0: Disables a receive underflow interrupt. 1: Enables a receive underflow interrupt.	R/W
b28	TOIEN	Transmit Overflow Interrupt Enable	0: Disables a transmit overflow interrupt. 1: Enables a transmit overflow interrupt.	R/W
b29	TUIEN	Transmit Underflow Interrupt Enable	0: Disables a transmit underflow interrupt. 1: Enables a transmit underflow interrupt.	R/W
b30	CKS	Audio Clock Select*3	0: AUDIO_MCLK input 1: Main clock	R/W
b31	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note 1. While this module is muted, low is transmitted regardless of the value of serial data, but data transmission is not stopped. Since the number of data in the transmit FIFO decreases, write dummy data to the SSIFTDR register to prevent the generation of a transmit underflow. When the MUEN bit is set to 1, the SSITXD0 pin immediately becomes low without synchronizing SSIWS0 pin.

Note 2. Set the SCKD and SWSD bits to the same value. Other settings are prohibited.

Note 3. Rewriting is allowed only in the idle state.

Downloaded from Arrow.com.

REN Bit (Receive Enable)

This bit enables or disables receive operation. Setting this bit to 1 starts receive operation.

TEN Bit (Transmit Enable)

This bit enables or disables transmit operation. Setting this bit to 1 starts transmit operation.
SSITXD0 pin of SSI0 is set as output while SSITXD0 is selected by the multi-function pin controller (MPC), regardless of the TEN bit setting.

Table 37.3 SSITXD0 and SSIRXD0 Pin States

Register Settings			SSI0	
MPC setting	TEN	REN	SSITXD0	SSIRXD0
SSI function	0	0	Output	Input
	0	1	Output	Input
	1	0	Output	Input
	1	1	Output	Input
Other than SSI function	x	x	Depends on the selected function	Depends on the selected function

x: Don't care
—: Settings prohibited.

CKDV[3:0] Bits (Serial Bit Clock Frequency Setting)

These bits select the frequency of the serial bit clock in master mode. Since the input clock from the SSISCK0 pin is used in slave mode, the setting of these bits is ignored. The serial bit clock is used as the operating clock of the shift register.

Calculation Example:
When fs (sampling rate) = the SSIWS0 frequency = 96 kHz and the system word length = 32 bits
The bit clock frequency = 96 kHz × 32 bits × 2 = 6.144 MHz is necessary, so set CKDV[3:0] = 0001b (MCLK/2) when MCLK = 12.288 MHz.

PDTA Bit (Parallel Data Allocation)

The setting of this bit specifies the allocation of data to be stored in the SSIFRDR register in receive mode and the SSIFTDR register in transmit mode.
During receive operation, the SSI stores the data received from the serial bus in the SSIFRDR register according to the PDTA bit setting.
During transmit operation, the SSI stores the data stored in the SSIFTDR register in the transmit shift register, and transmits the data to the serial bus according to the PDTA bit setting.

Downloaded from Arrow.com.

(1) When PDTA bit is 0

DWL[2:0] Bits	SSIFTDR and SSIFRDR Registers
000b	31 24 23 16 15 8 7 0 4th word | 3rd word | 2nd word | 1st word
001b	31 16 15 0 2nd word | 1st word
010b	31 14 13 0 Valid | Invalid
011b	31 12 11 0 Valid | Invalid
100b	31 10 9 0 Valid | Invalid
101b	31 8 7 0 Valid | Invalid

(2) When PDTA bit is 1

DWL[2:0] Bits	SSIFTDR and SSIFRDR Registers
000b	31 24 23 16 15 8 7 0 1st word | 2nd word | 3rd word | 4th word
001b	31 16 15 0 1st word | 2nd word
010b	31 18 17 0 Invalid | Valid
011b	31 20 19 0 Invalid | Valid
100b	31 22 21 0 Invalid | Valid
101b	31 24 23 0 Invalid | Valid

SCKP Bit (Serial Bit Clock Polarity)

This bit is used to select the polarity of SSISCK0 signal.

Table 37.4 lists the setting of SCKP bit and signal I/O timing.

Table 37.4 Setting of SCKP Bit and Signal Timing

	SCKP Bit = 0	SCKP Bit = 1
SSIRXD0 input sampling timing for reception	SSISCK0 rising edge	SSISCK0 falling edge
SSITXD0 output changing timing for transmission	SSISCK0 falling edge	SSISCK0 rising edge
SSIWS0 input sampling timing in slave mode (SWSD bit = 0)	SSISCK0 rising edge	SSISCK0 falling edge
SSIWS0 output changing timing in master mode (SWSD bit = 1)	SSISCK0 falling edge	SSISCK0 rising edge

CHNL[1:0] Bits (Channels)

These bits select the number of channels to be decoded in each system word. Set these bits to 00b in this module.

Downloaded from Arrow.com.

37.2.2 Status Register (SSISR)

Address(es): SSI0.SSISR 0008 A504h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	TUIRQ	TOIRQ	RUIRQ	ROIRQ	IIRQ	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	TCHNO[1:0]		TSWNO	RCHNO[1:0]		RSWNO	IDST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1

Bit	Symbol	Bit Name	Description	R/W
b0	IDST	Idle Status Flag	0: SSI communication is in progress. 1: SSI communication is idle.	R
b1	RSWNO	Receive System Word Number Flag	Receive word number	R
b3, b2	RCHNO[1:0]	Receive Channel Number Flag	These bits are read as 00b.	R
b4	TSWNO	Transmit System Word Number Flag	Transmit word number	R
b6, b5	TCHNO[1:0]	Transmit Channel Number Flag	These bits are read as 00b.	R
b24 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b25	IIRQ	Idle Interrupt Status Flag	0: Not in idle state 1: In idle state	R
b26	ROIRQ	Receive Overflow Interrupt Status Flag	0: No receive overflow has occurred. 1: A receive overflow has occurred.	R/(W) *1
b27	RUIRQ	Receive Underflow Interrupt Status Flag	0: No receive underflow has occurred. 1: A receive underflow has occurred.	R/(W) *1
b28	TOIRQ	Transmit Overflow Interrupt Status Flag	0: No transmit overflow has occurred. 1: A transmit overflow has occurred.	R/(W) *1
b29	TUIRQ	Transmit Underflow Interrupt Status Flag	0: No transmit underflow has occurred. 1: A transmit underflow has occurred.	R/(W) *1
b31, b30	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Writing 0 after confirming the flag to be 1 clears the flag. To clear flags, write 0 only to the flags to be cleared; write 1 to the other flags. Do not write 0 to a status flag indicating 0.

IDST Flag (Idle Status Flag)

This status flag indicates that the SSI is in idle state where communication is stopped.

This flag is set to 0 when communication starts after the SSICR.TEN bit or SSICR.REN bit is set to 1. Also, this flag is set to 1 if both the TEN and REN bits are set to 0 and system word communication is completed.

If the external device stops inputting the serial bit clock before communication is completed, this flag is not set to 1.

RSWNO Flag (Receive System Word Number Flag)

The initial value of this flag is 1, and its value is inverted when the data is transferred from the receive shift register to the SSIFRDR register.

This flag is initialized to 1 when the SSICR.REN bit value is changed from 0 to 1.

When the data word length specified by the SSICR.DWL[2:0] bits is 18 bits or more, this flag indicates which system word the data in the SSIFRDR register represents.

Downloaded from Arrow.com.

TSWNO Flag (Transmit System Word Number Flag)

This status flag indicates the current word number.
The initial value of this is 1, and its value is inverted when the data is transferred from the SSIFTDR register to the transmit shift register.
This flag is initialized to 1 when the SSICR.TEN bit value is changed from 0 to 1.
When the data word length specified by the SSICR.DWL[2:0] bits is 18 bits or more, this flag indicates the system word that is in the data transferred from the SSIFTDR register to the transmit shift register.

IIRQ Flag (Idle Interrupt Status Flag)

This status flag indicates whether this module is in idle state.
This flag is set regardless of the value of the SSICR.IIEN bit to allow polling.
The interrupt can be masked by setting the SSICR.IIEN bit to 0, but the interrupt cannot be cleared by writing 0 to this flag.
If IIRQ flag = 1 and SSICR.IIEN bit = 1, an interrupt occurs.

ROIRQ Flag (Receive Overflow Interrupt Status Flag)

This status flag indicates that receive data was supplied at a higher rate than was required. If a receive overflow occurs, stop reception and start from the beginning of the flowchart again.
This flag is set to 1 regardless of the setting of the SSICR.ROIEN bit. This flag can be set to 0 by writing 0 after confirming it to be 1.
If ROIRQ flag = 1 and SSICR.ROIEN bit = 1, an interrupt occurs.
If ROIRQ flag = 1, the data was transferred from the transmit shift register to the SSIFRDR register while the receive FIFO is full (SSIFSR.RDC[3:0] flags = 8h). This may lead to the loss of data.

Note: When an overflow occurs, the current data in the data buffer of this module is overwritten by the next incoming data from the SSI interface.

RUIRQ Flag (Receive Underflow Interrupt Status Flag)

This status flag indicates that receive data was supplied at a lower rate than was required. If a receive underflow occurs, stop reception and start the flowchart again from the beginning.
This flag is set to 1 regardless of the setting of the SSICR.RUIEN bit. This flag can be set to 0 by writing 0 after confirming it to be 1.
If RUIRQ flag = 1 and SSICR.RUIEN bit = 1, an interrupt occurs.
If RUIRQ flag = 1, the SSIFRDR register was read while the receive FIFO is empty (SSIFSR.RDC[3:0] flags = 0h). This may cause invalid receive data to be stored.

TOIRQ Flag (Transmit Overflow Interrupt Status Flag)

This status flag indicates that transmit data was supplied at a higher rate than was required. If a transmit overflow occurs, stop transmission and start from the beginning of the flowchart again.
This flag is set to 1 regardless of the setting of the SSICR.TOIEN bit. This flag can be set to 0 by writing 0 after confirming it to be 1.
If TOIRQ flag = 1 and SSICR.TOIEN bit = 1, an interrupt occurs.
If TOIRQ flag = 1, the SSIFTDR register had data written to it while the transmit FIFO is full (SSIFSR.TDC[3:0] flags = 8h). This may lead to the loss of data.

TUIRQ Flag (Transmit Underflow Interrupt Status Flag)

This status flag indicates that transmit data was supplied at a lower rate than was required. If a transmit underflow occurs, stop transmission and start from the beginning of the flowchart again.
This flag is set to 1 regardless of the setting of the SSICR.TUIEN bit. This flag can be set to 0 by writing 0 after

Downloaded from Arrow.com.

confirming it to be 1.
If TUIRQ flag = 1 and SSICR.TUIEN bit = 1, an interrupt occurs.
If TUIRQ flag = 1, the SSIFTDR register did not have data written to it before it was required for transmission. This may lead to the same data being transmitted once more.

Note: When a transmit underflow occurs, the last data input to the SSIFTDR register is transmitted until this module is in the idle state after transmission is stopped.

37.2.3 FIFO Control Register (SSIFCR)

Address(es): SSI0.SSIFCR 0008 A510h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	AUCKE	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SSIRST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	TTRG[1:0]		RTRG[1:0]		TIE	RIE	TFRST	RFRST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RFRST	Receive FIFO Data Register Reset*4	0: Release the receive FIFO data reset. 1: Initiates the receive FIFO data reset.	R/W
b1	TFRST	Transmit FIFO Data Register Reset*4	0: Release the transmit FIFO data reset. 1: Initiates the transmit FIFO data reset.	R/W
b2	RIE	Receive Data Full Interrupt Enable	0: Receive data full interrupt (RXI) request is disabled. 1: Receive data full interrupt (RXI) request is enabled.*1	R/W
b3	TIE	Transmit Data Empty Interrupt Enable	0: Transmit data empty interrupt (TXI) request is disabled. 1: Transmit data empty interrupt (TXI) request is enabled.*2	R/W
b5, b4	RTRG[1:0]	Receive FIFO Threshold Setting *4	b5 b4 0 0: 1 0 1: 2 1 0: 4 1 1: 6	R/W
b7, b6	TTRG[1:0]	Transmit FIFO Threshold Setting *4	b7 b6 0 0: 7 (1)*3 0 1: 6 (2)*3 1 0: 4 (4)*3 1 1: 2 (6)*3	R/W
b15 to b8	—	Reserved	These bits are read as undefined. The write value should be 0.	R/W
b16	SSIRST	SSI Software Reset	0: Clears the SSI software reset. 1: Initiates the SSI software reset.	R/W
b30 to b17	—	Reserved	These bits are read as undefined. The write value should be 0.	R/W
b31	AUCKE	Master Clock Enable*4	0: The master clock is disabled. 1: The master clock is enabled.	R/W

Note 1. The RXI request can be cleared by setting the SSIFSR.RDF flag to 0 (see the description of the SSIFSR.RDF flag for details) or RIE bit to 0.
Note 2. The TXI request can be cleared by setting the SSIFSR.TDE flag to 0 (see the description of the SSIFSR.TDE flag for details) or TIE bit to 0.
Note 3. The values in parenthesis are the number of empty stages in SSIFTDR at which the SSIFSR.TDE flag is set.
Note 4. Rewriting is allowed only in the idle state.

The SSIFCR register resets the number of the data bytes stored in the SSIFTDR and SSIFRDR registers, and specifies transmit FIFO and receive FIFO threshold values.

Downloaded from Arrow.com.

RFRST Bit (Receive FIFO Data Register Reset)

This bit invalidates the data in the SSIFRDR register to reset the FIFO to an empty state.

TFRST Bit (Transmit FIFO Data Register Reset)

This bit invalidates the data in the SSIFTDR register to reset the FIFO to an empty state.

RIE Bit (Receive Data Full Interrupt Enable)

This bit enables or disables generation of receive data full interrupt (RXI) requests when the SSIFSR.RDF flag is set to 1 during reception.

TIE Bit (Transmit Data Empty Interrupt Enable)

This bit enables or disables generation of transmit data empty interrupt (TXI) requests when the SSIFSR.TDE flag is set to 1 during transmit operation.

RTRG[1:0] Bits (Receive FIFO Threshold Setting)

These bits specify the receive FIFO threshold value. When the number of received data bytes stored in the SSIFRDR register (receive FIFO) has become equal to or greater than the value specified by the RTRG[1:0] bits, the SSIFSR.RDF flag is set to 1 and reading the received data is requested. If the SSIFCR.RIE bit is 1 at this time, a receive data full interrupt (RXI) request is generated.

TTRG[1:0] Bits (Transmit FIFO Threshold Setting)

These bits specify the transmit FIFO threshold value. When the number of transmit data bytes stored in the SSIFTDR register (transmit FIFO) has become equal to or less than the value specified by the TTRG[1:0], the SSIFSR.TDE flag is set to 1 and writing the transmit data is requested. If the SSIFCR.TIE bit is 1 at this time, a transmit data empty interrupt (TXI) request is generated.

SSIRST Bit (SSI Software Reset)

Writing 1 to this bit initializes the SSI internal status, registers other than the SSIFCR register, and bits other than this bit in the SSIFCR register. Since this bit is not automatically cleared to 0, confirm that 1 is written to it before writing 0. Do not write 0 to this bit and 1 to other bits at the same time. After modifying this bit, confirm that its value is modified before proceeding to the next processing.

Downloaded from Arrow.com.

37.2.4 FIFO Status Register (SSIFSR)

Address(es): SSI0.SSIFSR 0008 A514h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	TDC[3:0]				—	—	—	—	—	—	—	TDE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	RDC[3:0]				—	—	—	—	—	—	—	RDF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RDF	Receive Data Full Flag	0: Number of received data bytes in the SSIFRDR register is less than the specified receive FIFO threshold value. 1: Number of received data bytes in the SSIFRDR register is equal to or greater than the specified receive FIFO threshold value.	R/(W) *1
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11 to b8	RDC[3:0]	Receive Data Indicate Flag	Indicate the number of data units stored in the SSIFRDR register.	R
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	TDE	Transmit Data Empty Flag	0: Number of data bytes for transmission in the SSIFTDR register is greater than the specified transmit FIFO threshold value. 1: Number of data bytes for transmission in the SSIFTDR register is equal to or less than the specified transmit FIFO threshold value.*2	R/(W) *1
b23 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b27 to b24	TDC[3:0]	Transmit Data Indicate Flag	Indicate the number of data units stored in the SSIFTDR register.	R
b31 to b28	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Writing 0 after confirming the flag to be 1 clears the flag. To clear flags, write 0 only to the flags to be cleared; write 1 to the other flags. Do not write 0 to a status flag indicating 0.

Note 2. Since the SSIFTDR register is an 8-stage FIFO register, the amount of data that can be written to it while TDE flag = 1 is “8 - specified transmit FIFO threshold value” bytes at maximum. Writing more data will be ignored. The number of data bytes in the SSIFTDR register is indicated in the TDC[3:0] flags.

The SSIFSR register consists of status flags indicating the operating status of the SSIFTDR register and SSIFRDR register.

RDF Flag (Receive Data Full Flag)

This flag indicates that, when the received data is transferred to the SSIFRDR register, the number of data bytes in the SSIFRDR register has become equal to or greater than the receive FIFO threshold value, and thus reading the received data from the SSIFRDR register has been enabled.

[Setting condition]

- The number of receive data bytes that is equal to or greater than the value specified by the SSIFCR.RTRG[1:0] bits is stored in the SSIFRDR register.

[Clearing conditions]

- 0 is written to the RDF flag after the RDF flag is confirmed to be 1.
- Received data is read from the SSIFRDR register using DMA or DTC transfer (transfer of the last block in block transfer). Do not clear the RDF flag to 0 during DMA or DTC transfer.

Downloaded from Arrow.com.

Note: Since the SSIFRDR register is a 32-byte FIFO register, the maximum number of data bytes that can be read from it while the RDF flag is 1 is indicated in the RDC[3:0] flags. If reading data from the SSIFRDR register is continued after all the data is read, undefined values will be read.

RDC[3:0] Flags (Receive Data Indicate Flag)

These flags indicate the number of data bytes stored in the SSIFRDR register.
RDC[3:0] flags = 0h indicates no received data. RDC[3:0] flags = 8h indicates that 32 bytes of received data is stored in the SSIFRDR register.

TDE Flag (Transmit Data Empty Flag)

This flag indicates that, when data is transferred from the SSIFTDR register to the transmit shift register, the number of data bytes in the SSIFTDR register has become less than the transmit FIFO threshold value, and thus writing transmit data to the SSIFTDR register has been enabled.
[Setting condition]

- The number of the transmit data bytes written to the SSIFTDR register is equal to or less than the value specified by the SSIFCR.TTRG[1:0] bits.

[Clearing conditions]

- 0 is written to the TDE flag after the TDE flag is confirmed to be 1.
- Transmit data is written to the SSIFTDR register using DMA or DTC transfer (transfer of the last block in block transfer). Do not clear the TDE flag to 0 during DMA or DTC transfer.

Note: Since the SSIFTDR register is a 32-byte FIFO register, the maximum number of bytes that can be written to it while the TDE flag is 1 is 8 – TDC[3:0]. If writing data to the SSIFTDR register is continued after all the data is written, writing will be invalid and an overflow occurs.

TDC[3:0] Flags (Transmit Data Indicate Flag)

These flags indicate the number of data bytes stored in the SSIFTDR register.
TDC[3:0] flags = 0h indicates no data for transmission. TDC[3:0] flags = 8h indicates that 32 bytes of data for transmission is stored in the SSIFTDR register.

Downloaded from Arrow.com.

37.2.5 Transmit FIFO Data Register (SSIFTDR)

Address(es): SSI0.SSIFTDR 0008 A518h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

The SSIFTDR register is a write-only FIFO register consisting of eight stages of 32-bit registers for storing transmit data. Write transmit data to the SSIFTDR register in 64-bit (two stages of FIFO) units regardless of the data word length setting. If transmit data ends on a 32-bit boundary, write 00000000h for the other 32 bits, and stop transmission while 64-bit writing is completed. When the transmit shift register is empty, the SSI transfers the transmit data written to the SSIFTDR register to start serial transmission, which can be continued until the SSIFTDR register becomes empty.
Note that when the SSIFTDR register is full of data (32 bytes), the next data cannot be written to it. If writing is attempted, it will be ignored and an overflow occurs.

37.2.6 Receive FIFO Data Register (SSIFRDR)

Address(es): SSI0.SSIFRDR 0008 A51Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

The SSIFRDR register is a read-only FIFO register consisting of eight stages of 32-bit registers for storing received data. Each time 4 bytes of serial data is received, the SSI stores the received serial data in the SSIFRDR register from the receive shift register according to the PDTA bit setting. Receive operation can be continued until a maximum 32 bytes of data have been stored to in the SSIFRDR register. The SSIFRDR register can be read but cannot be written to. Note that when the SSIFRDR register is read when it stores no received data, undefined values will be read and a receive underflow occurs.
After the SSIFRDR register becomes full of received data, the data received thereafter will be lost and a receive overflow occurs.

Downloaded from Arrow.com.

37.2.7 TDM Mode Register (SSITDMR)

Address(es): SSI0.SSITDMR 0008 A520h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	CONT	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	CONT	WS Continue Mode*1	0: Disables WS continue mode. 1: Enables WS continue mode.	R/W
b31 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. This bit can be set only in master mode (SSICR.SCKD bit = 1 and SSICR.SWSD bit = 1).

The SSITDMR register is a readable/writable 32-bit register that enables or disables WS continue mode.

Downloaded from Arrow.com.

37.3 Operation

37.3.1 Bus Format

This module can operate as a transmitter or a receiver and can be configured into many serial bus formats in either mode. The bus format can be selected from one of the six modes shown in Table 37.5.

Table 37.5 Bus Format

	TEN	REN	SCKD	SWSD	MUEN	IIEN	TOIEN	TUIEN	ROIEN	RUIEN	CONT	SWSP	DEL	PDTA	SDTA	SPDP	SCKP	SWL[2:0]	DWL[2:0]	CHNL[1:0]
Non-compression slave receiver	0	1	0	0	Control bits							Configuration bits								
Non-compression slave transmitter	1	0	0	0																
Non-compression slave transceiver	1	1	0	0																
Non-compression master receiver	0	1	1	1																
Non-compression master transmitter	1	0	1	1																
Non-compression master transceiver	1	1	1	1																

37.3.2 Non-Compressed Mode

This SSI supports non-compressed mode only. It supports the I^2S compatible format as well as MSB-first and left-justified/right-justified.

(1) Slave Receiver

This mode allows the module to receive serial data from another device. The clock and word select signal used for the serial data stream is also supplied from an external device. If these signals do not conform to the format specified in the configuration fields of this module, operation is not guaranteed.

(2) Slave Transmitter

This mode allows the module to transmit serial data to another device. The clock and word select signal used for the serial data stream is also supplied from an external device. If these signals do not conform to the format specified in the configuration fields of this module, operation is not guaranteed.

(3) Slave Transceiver

This mode allows serial data transmission and reception between this module and another device. The clock and word select signal used for the serial data stream is also supplied from an external device. If these signals do not conform to the format specified in the configuration fields of this module, operation is not guaranteed.

(4) Master Receiver

This mode allows the module to receive serial data from another device. The clock and word select signals are internally derived from the master clock. The format of these signals is defined in the configuration fields of this module. If the incoming data does not follow the configured format, operation is not guaranteed.

(5) Master Transmitter

This mode allows the module to transmit serial data to another device. The clock and word select signals are internally derived from the master clock. The format of these signals is defined in the configuration fields of this module.

Downloaded from Arrow.com.

(6) Master Transceiver

This mode allows serial data transmission and reception between this module and another device. The clock and word select signals are internally derived from the master clock. The format of these signals is defined in the configuration fields of this module.

(7) Operating Settings Related to Word Length

All bits related to the SSICR register's word length are valid in non-compressed modes. There are many configurations this module supports, but some of the combinations are shown below for the I²S compatible format, MSB-first and left-justified format, and MSB-first and right-justified format.
In this section SSITXD0 and SSIRXD0 are referred to SSIDATA.

- I²S Compatible Format

Figure 37.2 and Figure 37.3 show the I²S compatible format both without and with padding.
Padding occurs when the data word length is smaller than the system word length.

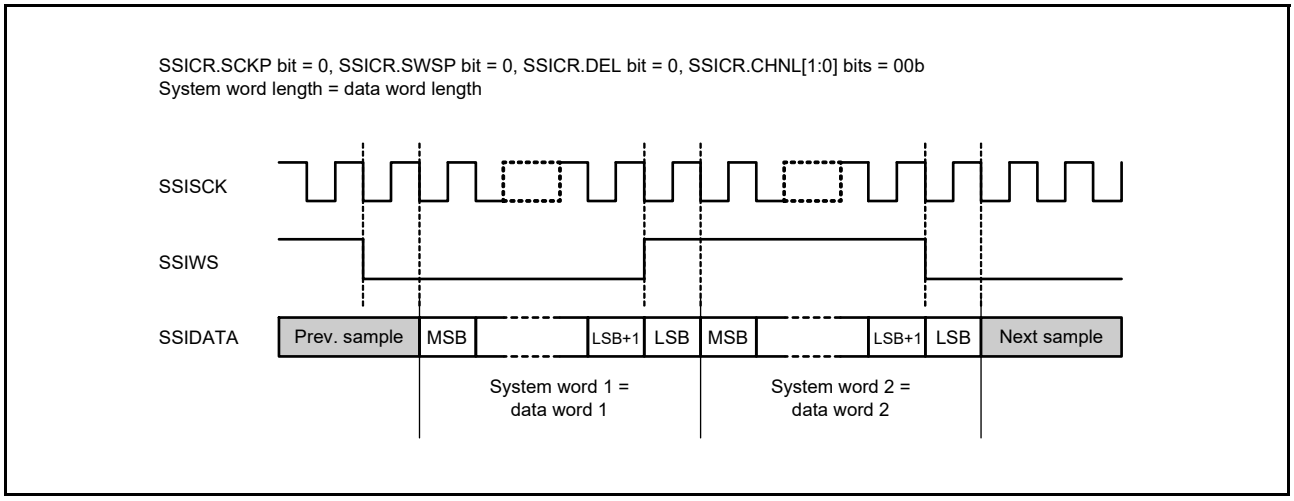


Figure 37.2 I²S Compatible Format (without Padding)

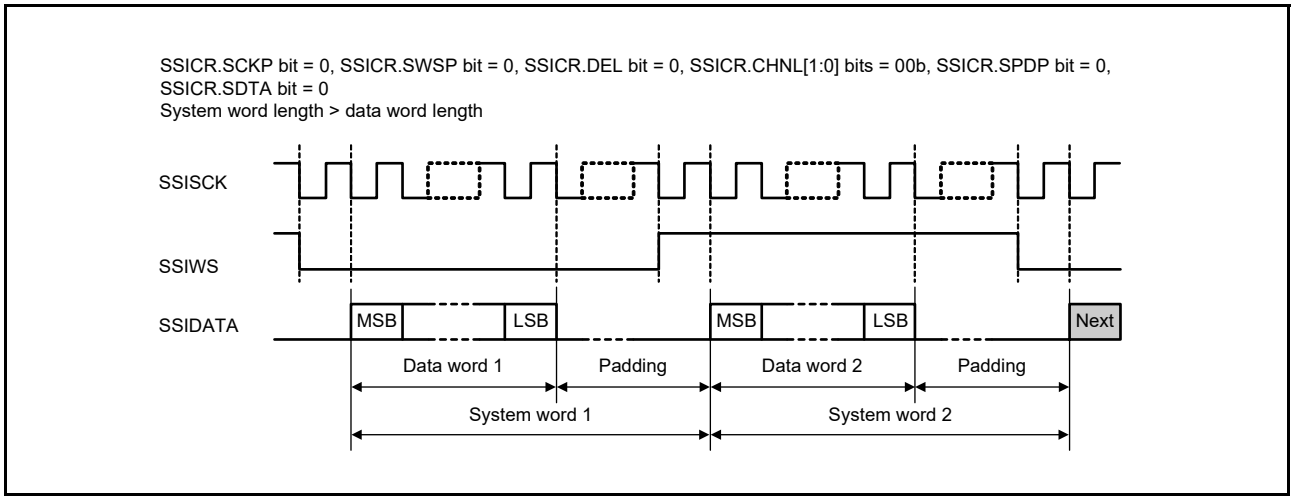


Figure 37.3 I²S Compatible Format (with Padding)

Downloaded from Arrow.com.

- MSB-First and Left-Justified Format

Figure 37.4 shows the MSB-first and left-justified format with padding.

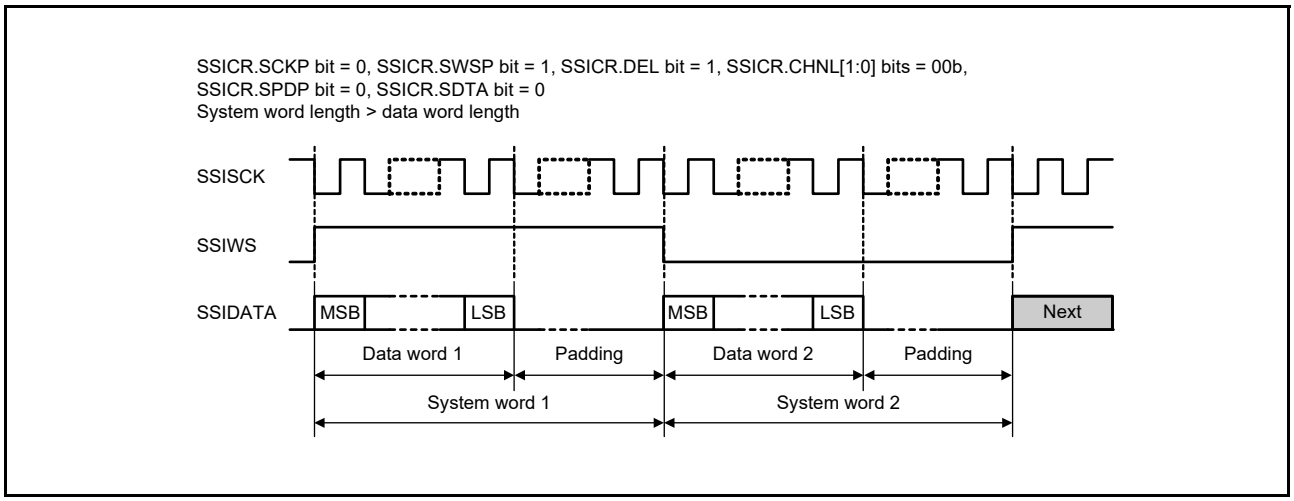


Figure 37.4 MSB-First and Left-Justified Format (Transmitted and Received in the Order of Serial Data and Padding Bits)

- MSB-First and Right-Justified Format

Figure 37.5 shows the MSB-first and right-justified format with padding.

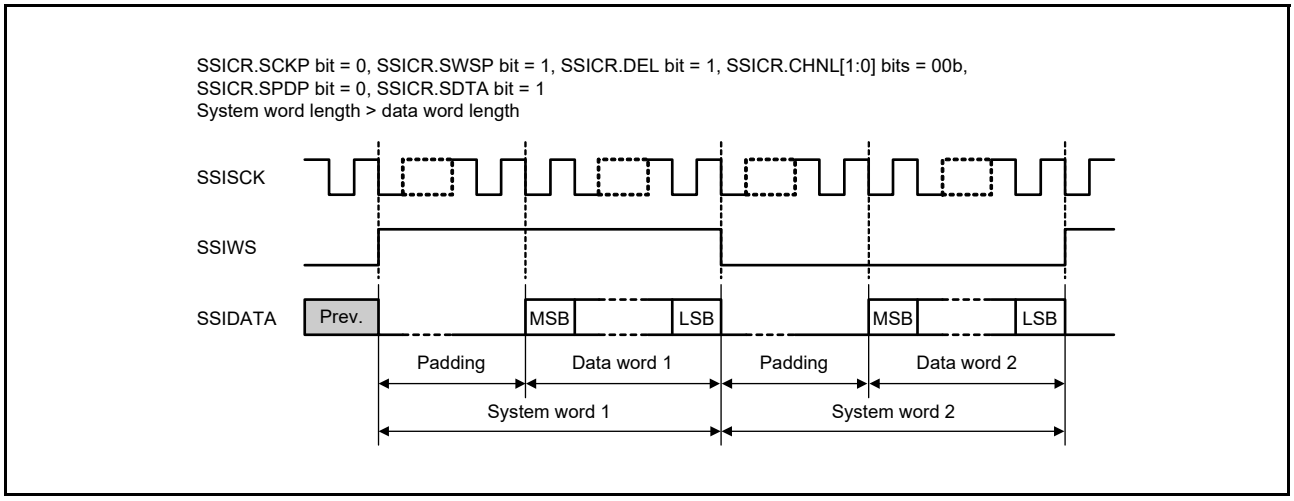


Figure 37.5 MSB-First and Right-Justified Format (Transmitted and Received in the Order of Padding Bits and Serial Data)

Table 37.6 shows the number of padding bits for each of the valid setting.

Table 37.6 Number of Padding Bits per System Word for Each Valid Setting

SSICR.CHNL[1:0] Bits (Decoded Channels per System Word)		SSICR.SWL[2:0] Bits (System Word Length)		SSICR.DWL[2:0] Bits (Data Word Length)					
				000b	001b	010b	011b	100b	101b
				8 bits	16 bits	18 bits	20 bits	22 bits	24 bits
00b	1 channel	000b	8 bits	0	—	—	—	—	—
		001b	16 bits	8	0	—	—	—	—
		010b	24 bits	16	8	6	4	2	0
		011b	32 bits	24	16	14	12	10	8

Downloaded from Arrow.com.

(8) Operating Settings Other than Word Length Related Settings

Several more configuration bits in non-compressed mode are shown below. These bits are not mutually exclusive, but some combinations may not be useful.

These configuration bits are described below with reference to the basic format sample in Figure 37.6.
In Figure 37.6 to Figure 37.14, a system word length of 6 bits and a data word length of 4 bits are used for simplification of these figures.

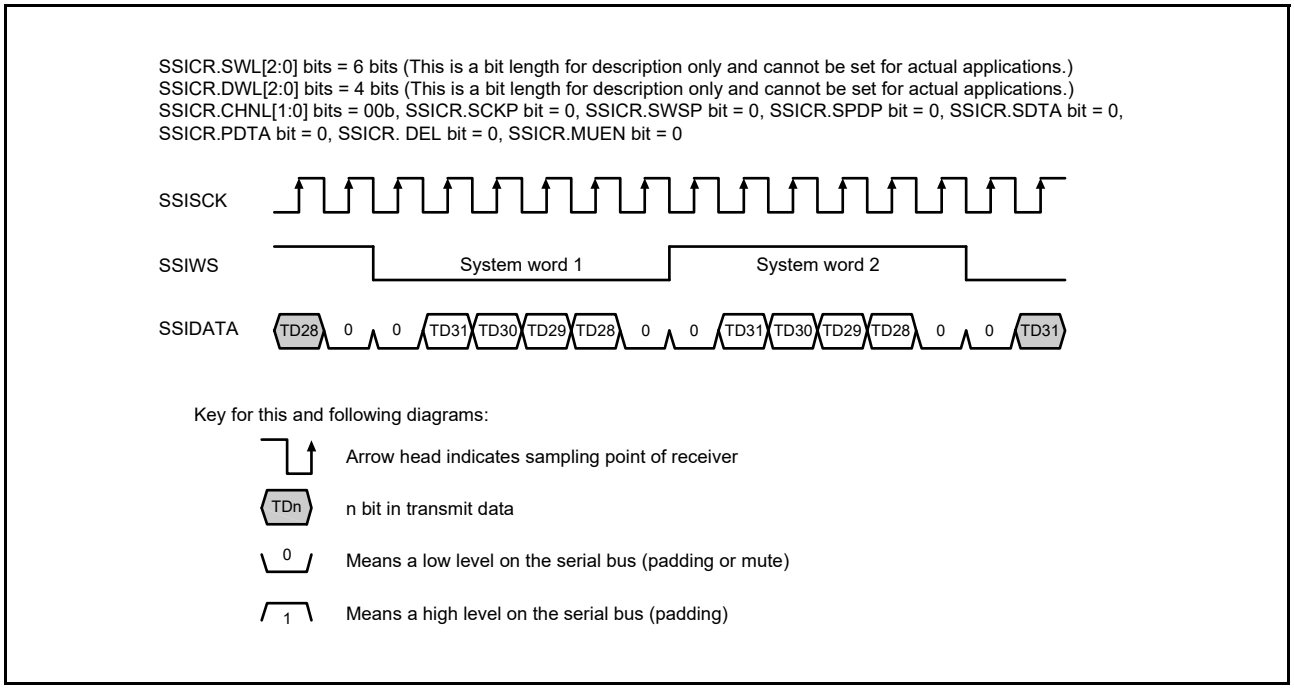


Figure 37.6 Basic Format Sample (Transmit Mode)

- Inverted Clock

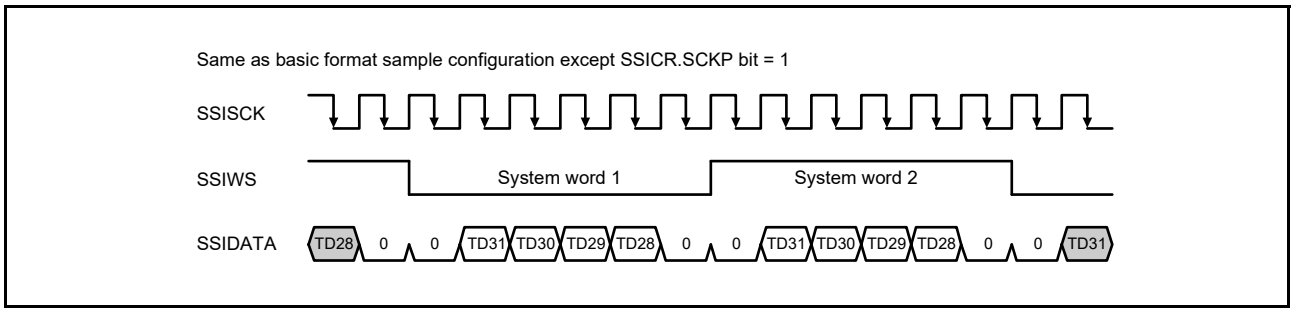


Figure 37.7 Inverted Clock

Downloaded from Arrow.com.

- Inverted Word Select

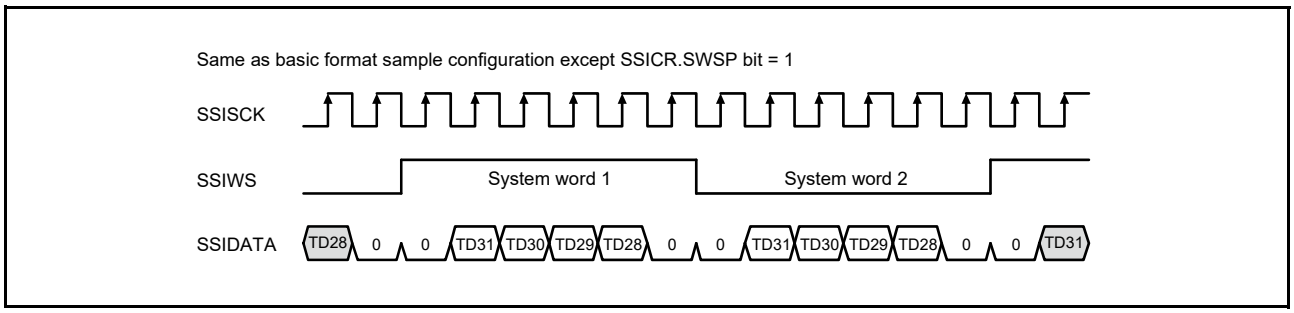


Figure 37.8 Inverted Word Select

- Inverted Padding Polarity

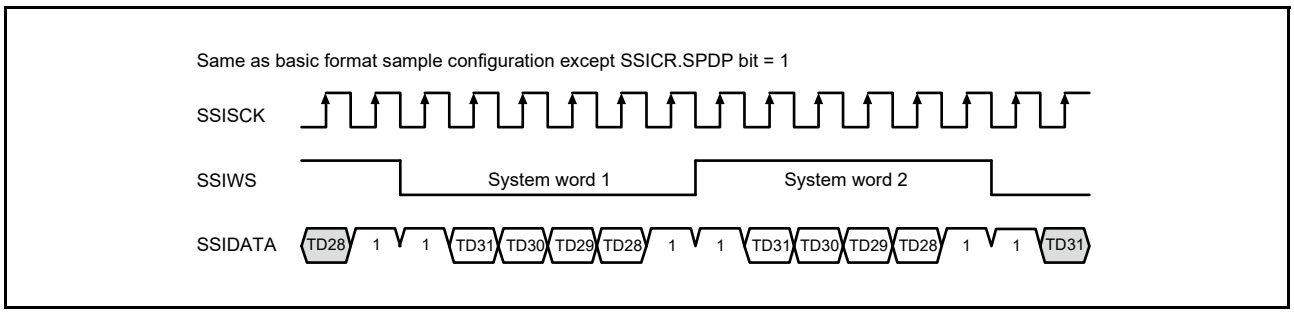


Figure 37.9 Inverted Padding Polarity

- Transmitting and Receiving in the Order of Padding Bits and Serial Data; with Delay

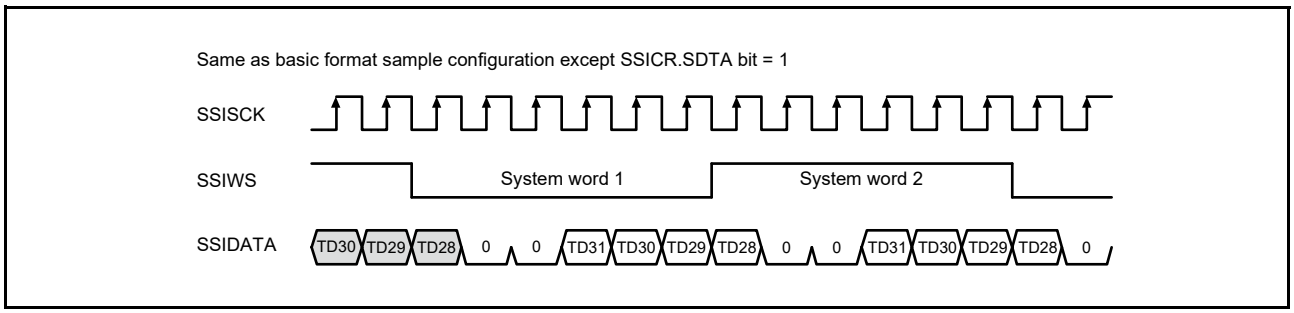


Figure 37.10 Transmitting and Receiving in the Order of Padding Bits and Serial Data; with Delay

- Transmitting and Receiving in the Order of Padding Bits and Serial Data; without Delay

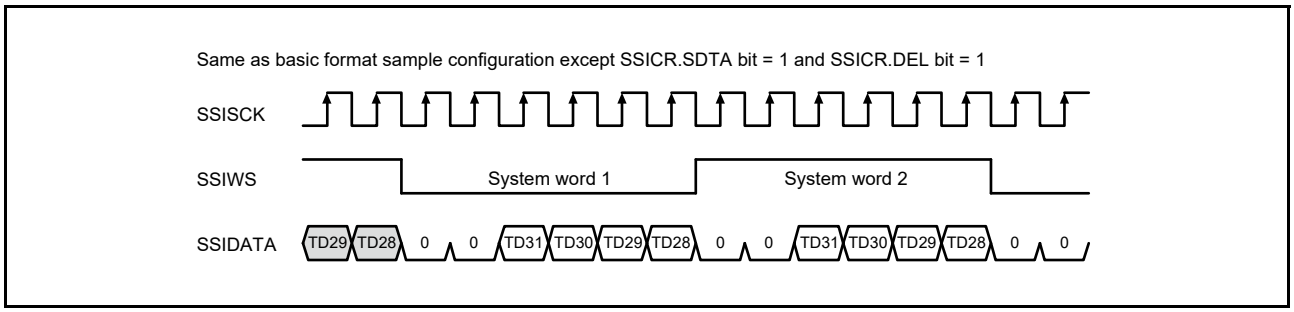


Figure 37.11 Transmitting and Receiving in the Order of Padding Bits and Serial Data; without Delay

Downloaded from Arrow.com.

- Transmitting and Receiving in the Order of Serial Data and Padding Bits; without Delay

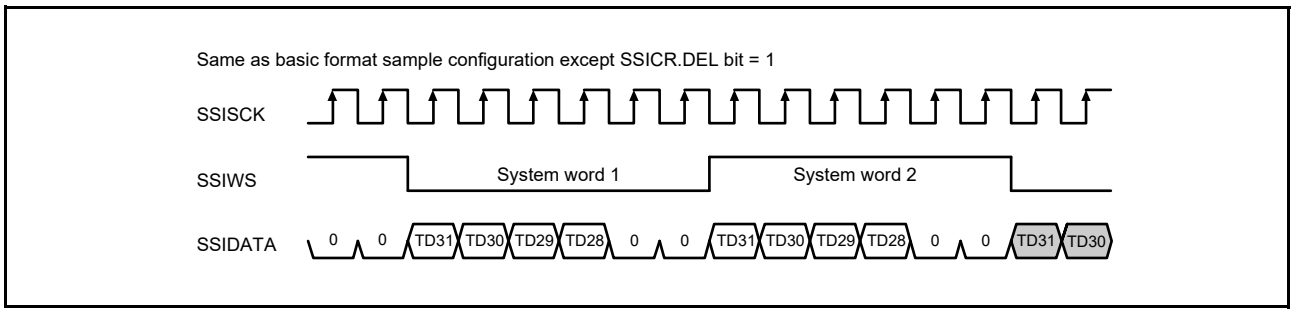


Figure 37.12 Transmitting and Receiving in the Order of Serial Data and Padding Bits; without Delay

- Parallel Right-Justified with Delay

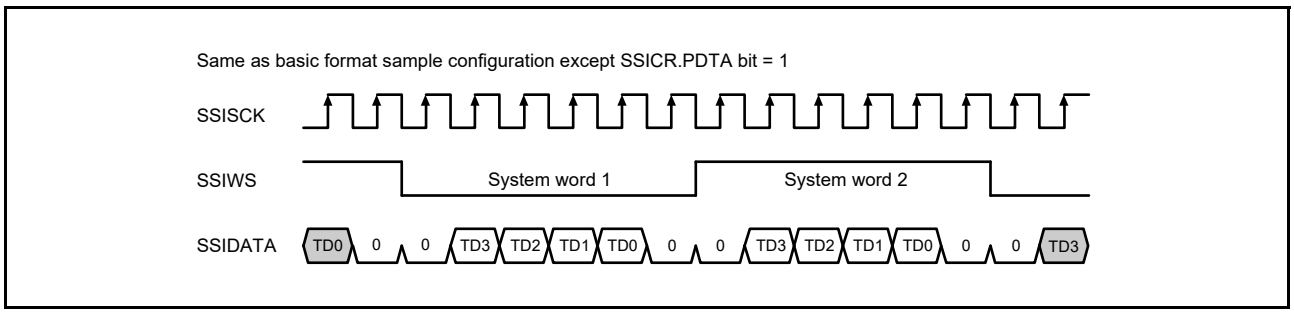


Figure 37.13 Parallel Right-Justified with Delay

- Mute Enabled

When the SSICR.MUEN bit is set to 1, the SSITXD0 pin becomes low (0) without synchronizing SSIWS0.

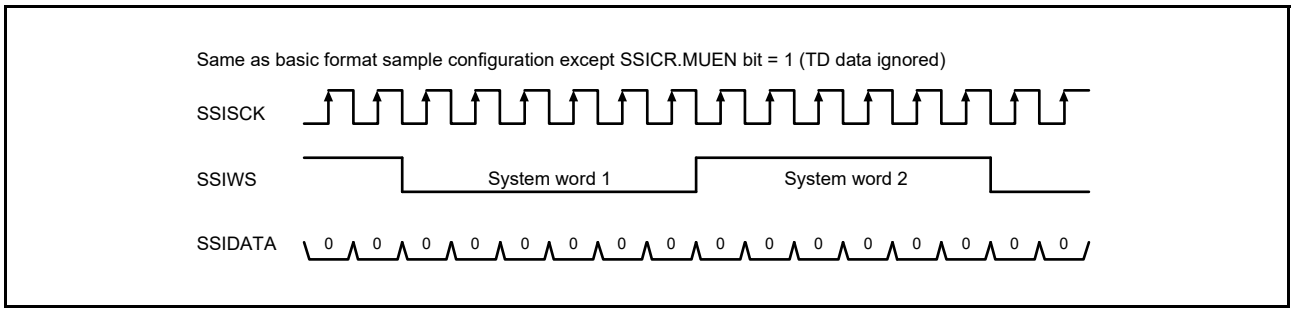


Figure 37.14 Mute Enabled

Downloaded from Arrow.com.

37.3.3 WS Continue Mode

In WS continue mode, the SSIWS0 signal continues to be toggled irrespective whether data transmission is enabled or disabled. This mode can be set using the SSITDMR.CONT bit. With this mode enabled, the SSIWS0 signal does not stop but continues toggling even if the SSICR.TEN and REN bits are both set to 0 (transmission disabled). With this mode disabled, the SSIWS0 signal stops if the SSICR.TEN and REN bits are both set to 0.

Figure 37.15 and Figure 37.16 show the operations with WS continue mode enabled and disabled, respectively.

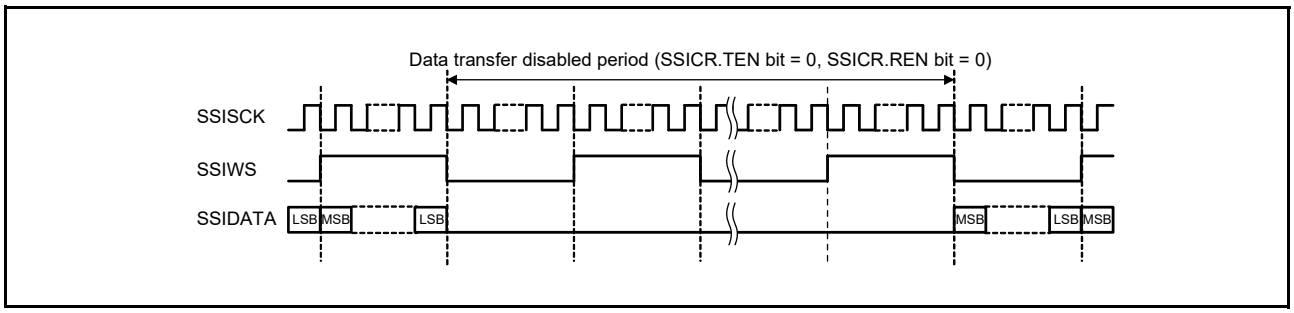


Figure 37.15 WS Continue Mode Enabled

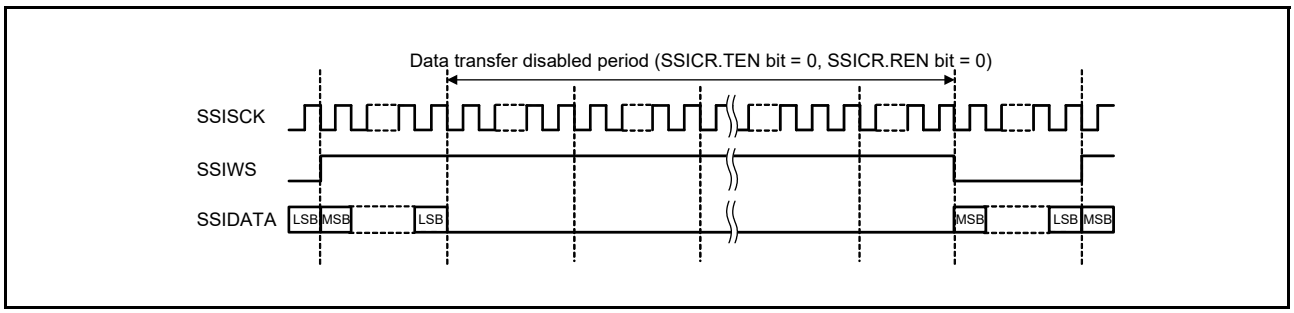


Figure 37.16 WS Continue Mode Disabled

Downloaded from Arrow.com.

37.3.4 Operating States

There are three states of operation: idle, communication, and waiting for idle. Figure 37.17 shows the operating state transitions.

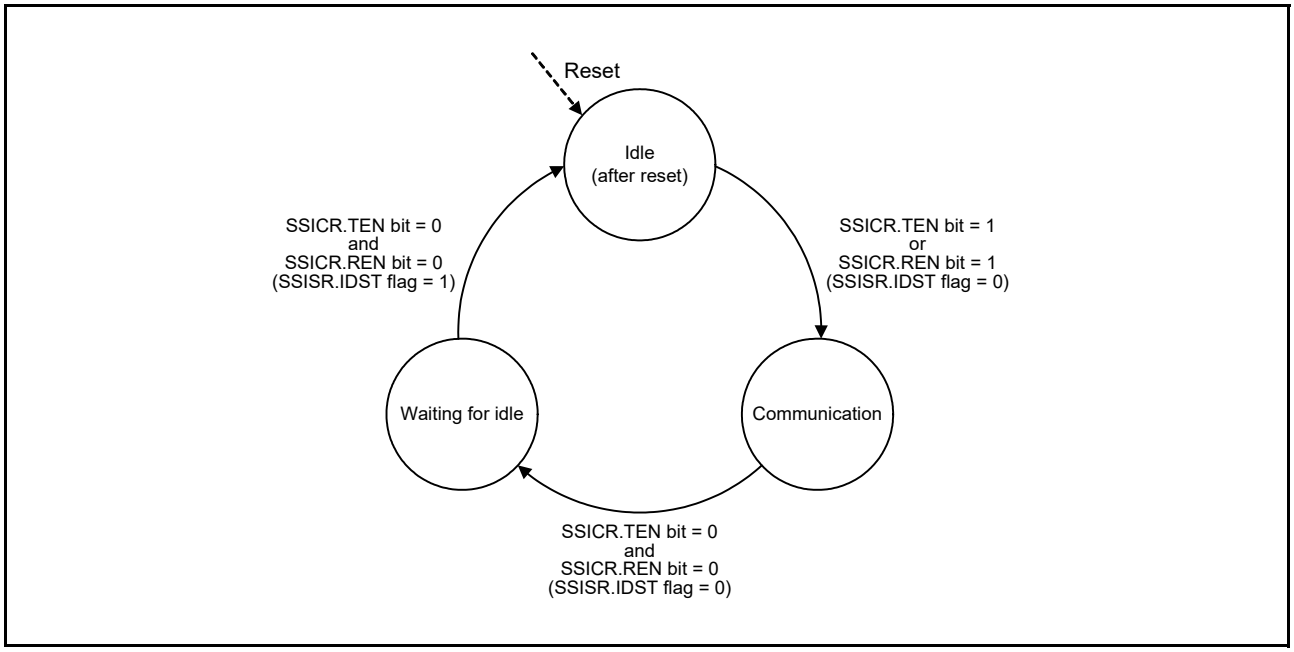


Figure 37.17 Operating State Transitions

(1) Idle State

This module enters this state when the MSTPCRD.MSTPD14 and MSTPD15 bits are set to 0 after a reset is released. All required configuration fields in the control register should be defined in this state. After the settings are made, the module enters communication state when the SSICR.TEN bit or SSICR.REN bit is set to 1.

(2) Communication State

Communication in this state depends on the selected operating state. For details, refer to section 37.3.5, Transmit Operation and section 37.3.6, Receive Operation.

(3) Waiting for Idle

This module enters this state when both the SSICR.TEN and SSICR.REN bits are set to 0 in communication state. If system word communication is completed in this state, the SSISR.IDST flag is set to 1 and this module enters the idle state.

Downloaded from Arrow.com.

37.3.5 Transmit Operation

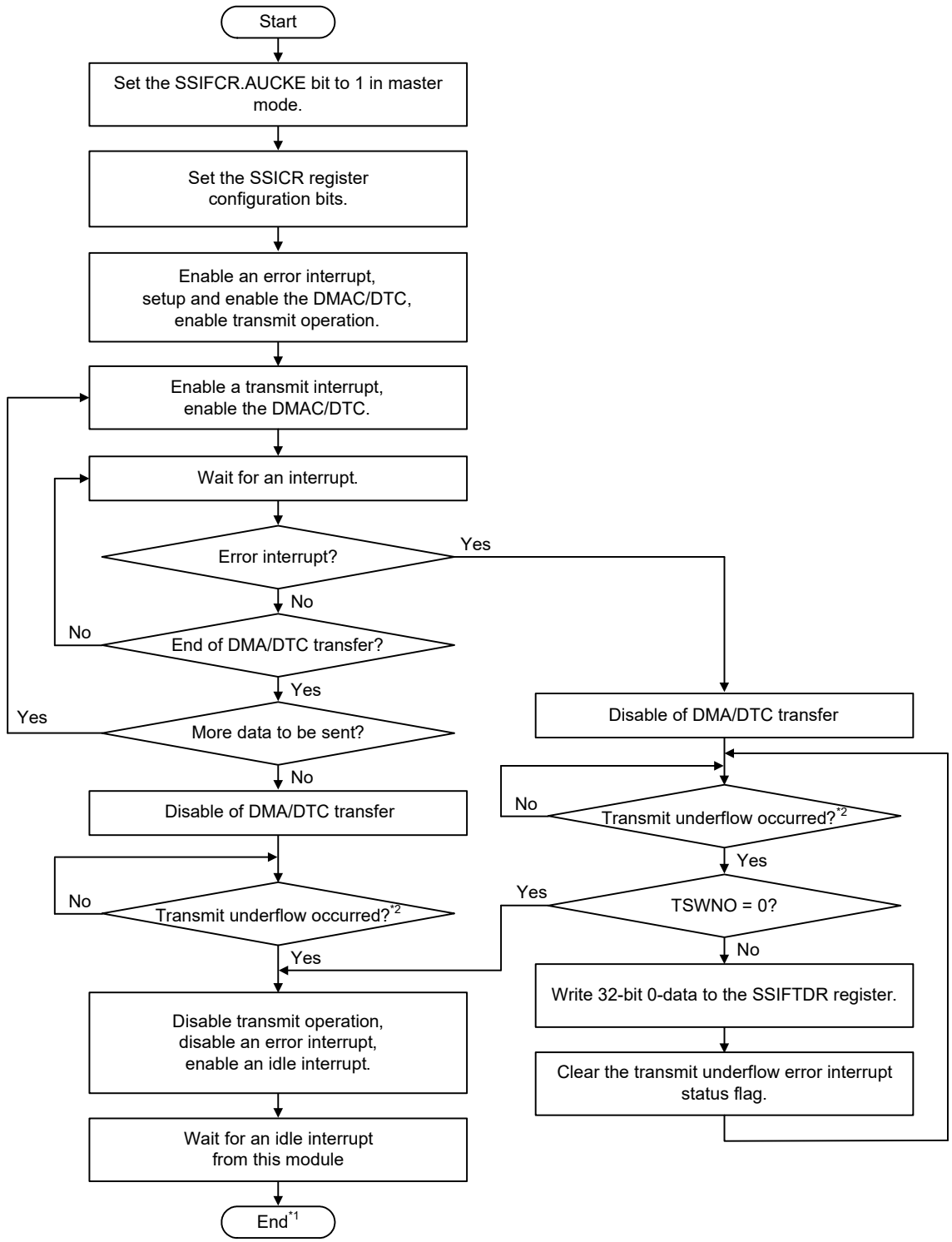
Transmission can be controlled either by DMA/DTC transfer or interrupt.
DMAC/DTC control is preferred to reduce the processor load. In transmission using the DMAC/DTC, the processor will only receive interrupts if there is an underflow or overflow of data or if DMA/DTC transfer has been completed. In transmission using DMA/DTC transfer, set the number of DMA/DTC transfers to multiples of 2 to write transmit data to the SSIFTDR register in 64-bit (two stages of FIFO) units.
The alternative method is using the interrupts that this module generates to supply data as required. In transmission using interrupts, write transmit data in 64-bit units regardless of the data format. If transmit data ends on a 32-bit boundary, write 00000000h after the last transmit data is written, and complete writing on a 64-bit boundary.
When stopping transmission, stop writing to the SSIFTDR register while 64-bit writing is completed. After writing is stopped, wait until a transmit underflow occurs before setting the SSICR.TEN bit to 0. During transmit underflow, the last data input to SSIFTDR register is continuously transmitted until this module enters the idle state. After setting the TEN bit to 0, continue to supply the clock*1 until the SSISR.IIRQ flag becomes 1 (in idle state). If a transmit underflow error or transmit overflow error occurs during data transmission, transmit data to SSIFTDR register may not be written in a 64-bit units. In that case, stop writing data, wait until a transmit underflow error occurs, and check the SSISR.TSWNO flag when the transmit underflow has occurred. When the TSWNO flag is 1, write 00000000h to SSIFTDR register and wait until an underflow occurs again. Once the TSWNO flag is confirmed to be 0, Set the TEN bit to 0 and continue to supply the clock*1 until the SSISR.IIRQ flag becomes 1 (in idle state).

Figure 37.18 shows transmission flow using the DMA/DTC, and Figure 37.19 shows transmission flow using interrupts.

Note 1. Input clock from the SSISCK0 pin when SSICR.SCKD bit = 0.
Master clock when SSICR.SCKD bit = 1.

Downloaded from Arrow.com.

(1) Transmission Using the DMAC/DTC

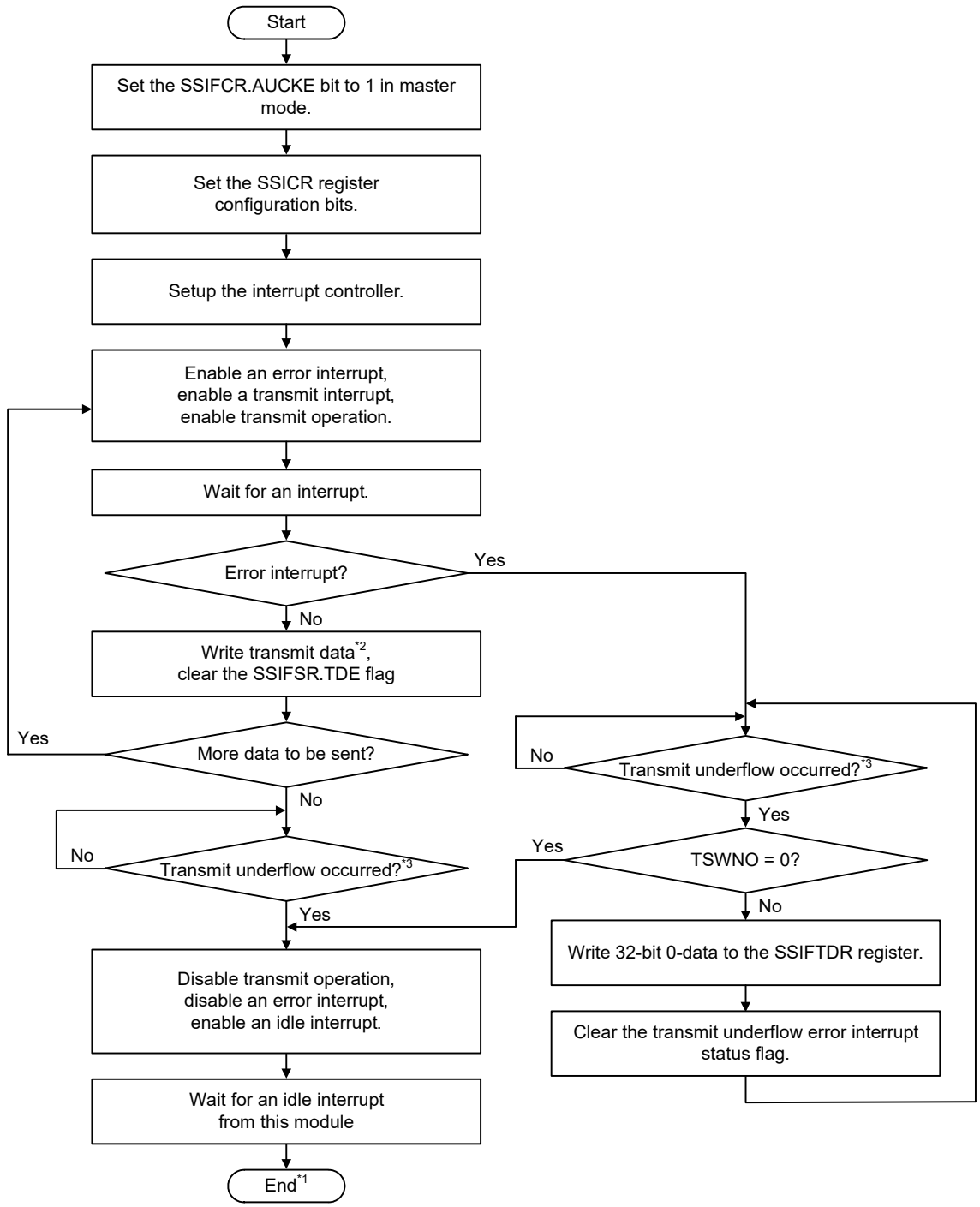


Note 1. When restarting transmission after transmit operation is disabled (TEN = 0) while WS continue mode is disabled, execute a software reset and go back to the start of the flowchart again.
Note 2. Stop writing transmit data to the SSIFTDR register and wait until a transmit underflow occurs. During transmit underflow, the last data input to SSIFTDR register is continuously transmitted until the module is in the idle state.

Figure 37.18 Transmission Using the DMAC/DTC

Downloaded from Arrow.com.

(2) Transmission Using Interrupts



Note 1. When restarting transmission after transmit operation is disabled (TEN = 0) while WS continue mode is disabled, execute a software reset and go back to the start of the flowchart again.
Note 2. Write transmit data in 64-bit units regardless of the data word setting.
Note 3. Stop writing transmit data to the SSIFTDR register and wait until a transmit underflow occurs. During transmit underflow, the last data input to SSIFTDR register is continuously transmitted until the module is in the idle state.

Figure 37.19 Transmission Using Interrupts

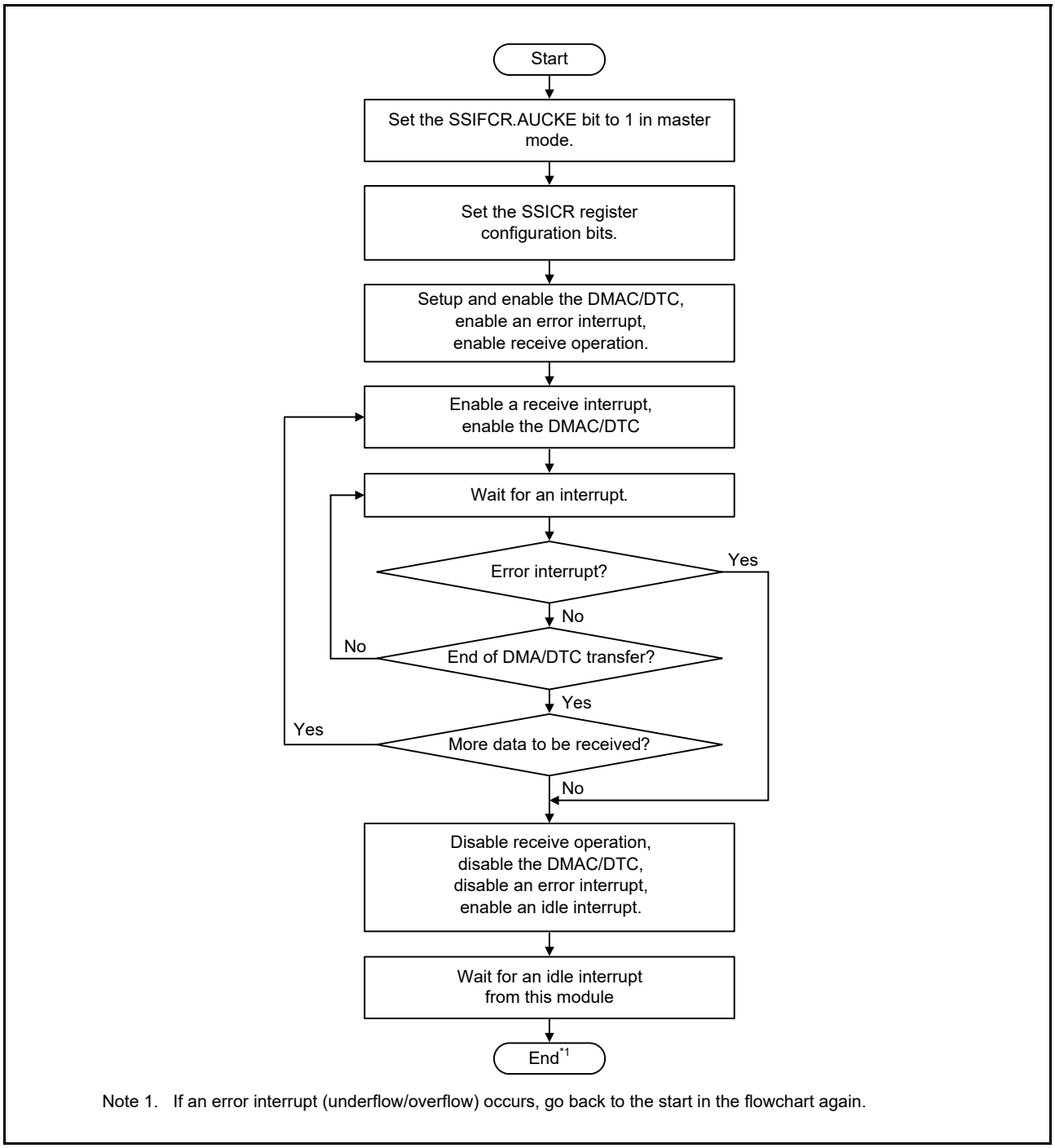
Downloaded from Arrow.com.

37.3.6 Receive Operation

Like transmission, reception can be controlled either by DMA/DTC transfer or interrupt.
Figure 37.20 and Figure 37.21 show the flow of operation.
When stopping reception, set the SSICR.REN bit to 0 and continue to supply the clock*1 until the SSISR.IIRQ flag becomes 1 (in idle state).

Note 1. Input clock from the SSISCK0 pin when SSICR.SCKD bit = 0.
Master clock when SSICR.SCKD bit = 1.

(1) Reception Using the DMAC/DTC

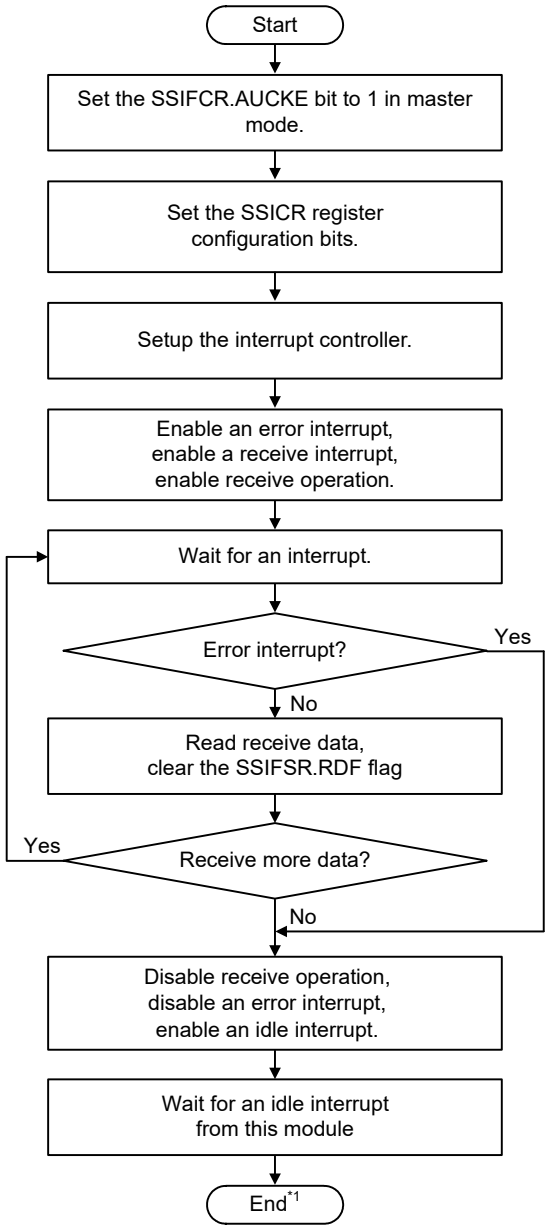


Note 1. If an error interrupt (underflow/overflow) occurs, go back to the start in the flowchart again.

Figure 37.20 Reception Using the DMAC/DTC

Downloaded from Arrow.com.

(2) Reception Using Interrupts



Note 1. If an error interrupt (underflow/overflow) occurs, go back to the start in the flowchart again.

Figure 37.21 Reception Using Interrupts

Downloaded from Arrow.com.

37.3.7 Serial Bit Clock Control

The SSI controls and selects the serial bit clock, according to the SSICR.SCKD and CKDV[3:0] bits setting.
If the serial bit clock direction is set to input (SCKD bit = 0), this module is in slave mode and the shift register uses the clock that was input to the SSISCK0 pin as the bit clock.

If the serial bit clock direction is set to output (SCKD bit = 1), this module is in master mode, and the shift register uses the master clock (MCLK) or a divided master clock as the bit clock. The master clock is divided by the ratio specified by the SSICR.CKDV[3:0] bits for use as the bit clock by the shift register.
In either case the module pin, SSISCK0, is the same as the bit clock.

37.4 Interrupt Sources

Table 37.7 lists the interrupt sources of the SSI. Each interrupt source can be enabled or disabled by the SSICR.TUIEN, TOIEN, RUIEN, ROIEN and IIEN bits, and the SSIFCR.TIE and RIE bits.

Table 37.7 SSI Interrupt Sources

Channel	Interrupt Source	Description	Interrupt Status Flag	Interrupt Enable Bit	DMAC/DTC Start Trigger
SSI0	SSIF0	Transmit underflow interrupt Transmit overflow interrupt Receive underflow interrupt Receive overflow interrupt Idle interrupt	SSISR.TUIRQ SSISR.TOIRQ SSISR.RUIRQ SSISR.ROIRQ SSISR.IIRQ	SSICR.TUIEN SSICR.TOIEN SSICR.RUIEN SSICR.ROIEN SSICR.IIEN	Not available
	SSIRXI0	Receive data full interrupt (RXI)	SSIFSR.RDF	SSIFCR.RIE	Available
	SSITXI0	Transmit data empty interrupt (TXI)	SSIFSR.TDE	SSIFCR.TIE	Available

Downloaded from Arrow.com.

37.5 Usage Notes

37.5.1 Setting the Module Stop Function

Module stop state can be entered or released using the MSTPCRD register. The initial setting of the SSI is in the module stop state. SSI register access is enabled by releasing the module stop state.
For details on the MSTPCRD register, refer to section 11, Low Power Consumption.

37.5.2 Notes on Changing Transmission Modes

For mode transitions between the transmitter, receiver, and transceiver while WS continue mode is disabled (SSITDMR.CONT = 0), set the SSICR.TEN and SSICR.REN bits to 0 and make a transition to the idle state once. Set the SSICR.TEN and SSICR.REN bits again while the module is in the idle state and restart transmission.

37.5.3 Limits on WS Continue Mode

If WS continue mode setting is changed, the operation of the SSISCK0 and SSIWS0 signals immediately after switching are not guaranteed. If it affects the device to be connected, do not change the setting dynamically.

Downloaded from Arrow.com.

38. Serial Peripheral Interface (RSPIa)

In this section, “PCLK” is used to refer to PCLKB.

38.1 Overview

This MCU includes one channel of Serial Peripheral Interface (RSPI).
The RSPI channels are capable of high-speed, full-duplex synchronous serial communications with multiple processors and peripheral devices.
Table 38.1 lists the specifications of the RSPI, and Figure 38.1 shows a block diagram of the RSPI.
In this section, m as used with the RSPI command registers (SPCMDm) indicates 0 to 7.

Table 38.1 RSPI Specifications (1/2)

Item	Description
Number of channels	One channel
RSPI transfer functions	• Use of MOSI (master out/slave in), MISO (master in/slave out), SSL (slave select), and RSPCK (RSPI clock) signals allows serial communications through SPI operation (4-wire method) or clock synchronous operation (3-wire method). • Transmit-only operation is available. • Communication mode: Full-duplex or transmit-only can be selected. • Switching of the polarity of RSPCK • Switching of the phase of RSPCK
Data format	• MSB first/LSB first selectable • Transfer bit length is selectable as 8, 9, 10, 11, 12, 13, 14, 15, 16, 20, 24, or 32 bits. • 128-bit transmit/receive buffers • Up to four frames can be transferred in one round of transmission/reception (each frame consisting of up to 32 bits).
Bit rate	• In master mode, the on-chip baud rate generator generates RSPCK by frequency-dividing PCLK (the division ratio ranges from divided by 2 to divided by 4096). • In slave mode, the minimum PCLK clock divided by 8 can be input as RSPCK (the maximum frequency of RSPCK is that of PCLK divided by 8). Width at high level: 4 cycles of PCLK; width at low level: 4 cycles of PCLK
Buffer configuration	• Double buffer configuration for the transmit/receive buffers • 128 bits for the transmit/receive buffers
Error detection	• Mode fault error detection • Overrun error detection*1 • Parity error detection
SSL control function	• Three SSL pins (SSLA0, SSLA1, and SSLA3) for each channel • In single-master mode, SSLA0, SSLA1, and SSLA3 pins are output. • In multi-master mode: SSLA0 pin for input, and SSLA1, SSLA3 pins for either output or unused. • In slave mode: SSLA0 pin for input, and SSLA1, SSLA3 pins for unused. • Controllable delay from SSL output assertion to RSPCK operation (RSPCK delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units) • Controllable delay from RSPCK stop to SSL output negation (SSL negation delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units) • Controllable wait for next-access SSL output assertion (next-access delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units) • Function for changing SSL polarity
Control in master transfer	• A transfer of up to eight commands can be executed sequentially in looped execution. • For each command, the following can be set: SSL signal value, bit rate, RSPCK polarity/phase, transfer data length, MSB/LSB first, burst, RSPCK delay, SSL negation delay, and next-access delay • A transfer can be initiated by writing to the transmit buffer. • MOSI signal value specifiable in SSL negation • RSPCK auto-stop function

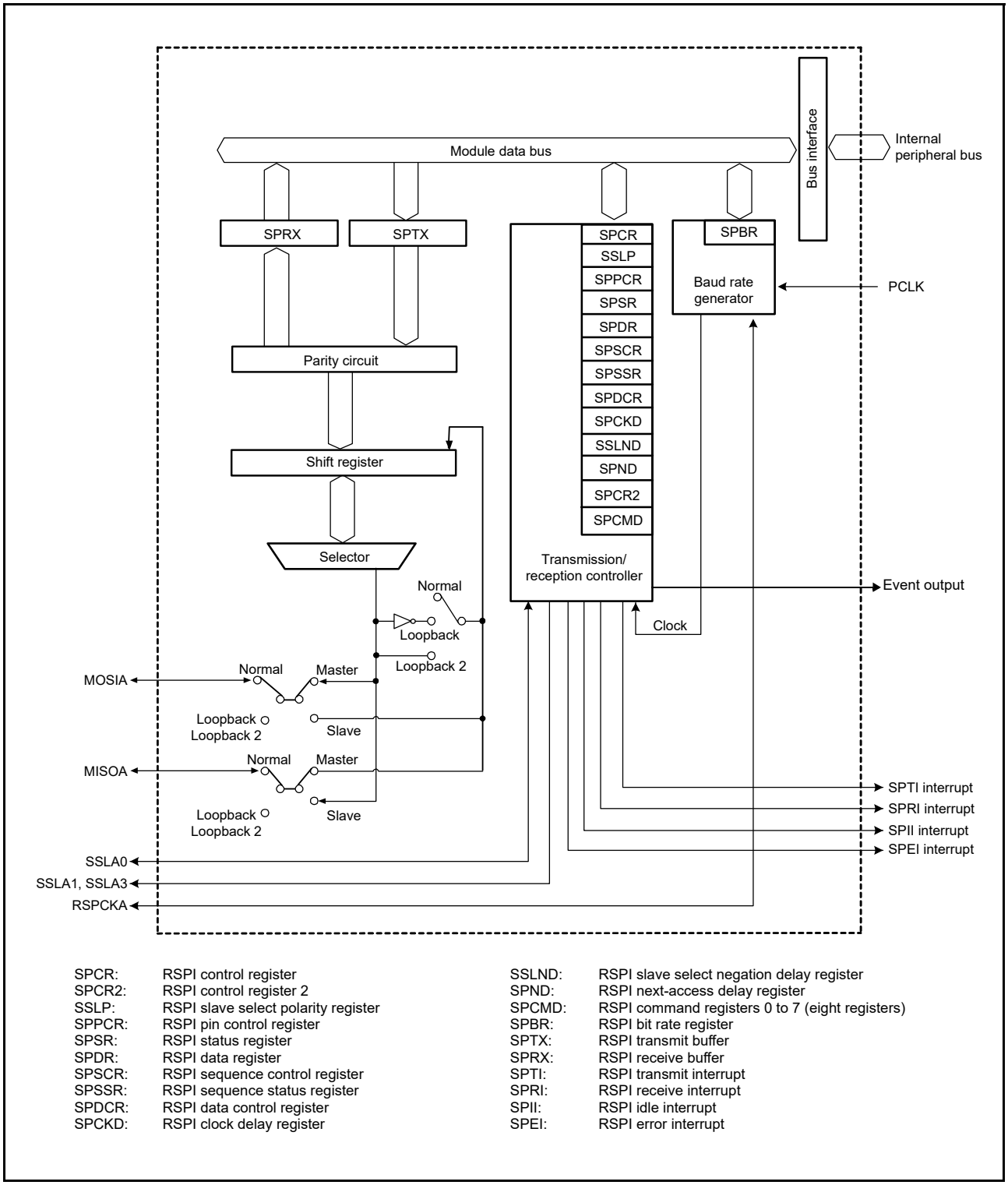
Downloaded from Arrow.com.

Table 38.1 RSPI Specifications (2/2)

Item	Description
Interrupt sources	• Interrupt sources Receive buffer full interrupt Transmit buffer empty interrupt RSPI error interrupt (mode fault, overrun, or parity error) RSPI idle interrupt (RSPI idle)
Event link function (output)	• The following events can be output to the event link controller. (RSPI0) Receive buffer full signal Transmit buffer empty signal Mode fault, overrun, or parity error signal RSPI idle signal Transmission-completed signal
Others	• Function for switching between CMOS output and open-drain output • Function for initializing the RSPI • Loopback mode
Low power consumption function	Module stop state can be set.

Note 1. In master reception and when the RSPCK auto-stop function is enabled, an overrun error does not occur because the transfer clock is stopped at the timing of overrun error detection.

Downloaded from Arrow.com.



SPCR:	RSPI control register	SSLND:	RSPI slave select negation delay register
SPCR2:	RSPI control register 2	SPND:	RSPI next-access delay register
SSLP:	RSPI slave select polarity register	SPCMD:	RSPI command registers 0 to 7 (eight registers)
SPPCR:	RSPI pin control register	SPBR:	RSPI bit rate register
SPSR:	RSPI status register	SPTX:	RSPI transmit buffer
SPDR:	RSPI data register	SPRX:	RSPI receive buffer
SPSCR:	RSPI sequence control register	SPTI:	RSPI transmit interrupt
SPSSR:	RSPI sequence status register	SPRI:	RSPI receive interrupt
SPDCR:	RSPI data control register	SPII:	RSPI idle interrupt
SPCKD:	RSPI clock delay register	SPEI:	RSPI error interrupt

Figure 38.1 RSPI Block Diagram

Downloaded from Arrow.com.

Table 38.2 lists the I/O pins used in the RSPI.
The RSPI automatically switches the I/O direction of the SSLA0 pin. SSLA0 is set as an output when the RSPI is a single master and as an input when the RSPI is a multi-master or a slave. Pins RSPCKA, MOSIA, and MISOA are automatically set as inputs or outputs according to the setting of master or slave and the level input on the SSLA0 pin. Refer to section 38.3.2, Controlling RSPI Pins for details.

Table 38.2 RSPI Pin Configuration

Channel	Pin Name	I/O	Function
RSPI0	RSPCKA	I/O	Clock I/O
	MOSIA	I/O	Master transmit data I/O
	MISOA	I/O	Slave transmit data I/O
	SSLA0	I/O	Slave selection I/O
	SSLA1	Output	Slave selection output
	SSLA3	Output	Slave selection output

Downloaded from Arrow.com.

38.2 Register Descriptions

38.2.1 RSPI Control Register (SPCR)

Address(es): RSPI0.SPCR 0008 8380h

	b7	b6	b5	b4	b3	b2	b1	b0
	SPRIE	SPE	SPTIE	SPEIE	MSTR	MODFEN	TXMD	SPMS
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SPMS	RSPI Mode Select	0: SPI operation (4-wire method) 1: Clock synchronous operation (3-wire method)	R/W
b1	TXMD	Communications Operating Mode Select	0: Full-duplex synchronous serial communications 1: Serial communications consisting of only transmit operations	R/W
b2	MODFEN	Mode Fault Error Detection Enable	0: Disables the detection of mode fault error 1: Enables the detection of mode fault error	R/W
b3	MSTR	RSPI Master/Slave Mode Select	0: Slave mode 1: Master mode	R/W
b4	SPEIE	RSPI Error Interrupt Enable	0: Disables the generation of RSPI error interrupt requests 1: Enables the generation of RSPI error interrupt requests	R/W
b5	SPTIE	Transmit Buffer Empty Interrupt Enable	0: Disables the generation of transmit buffer empty interrupt requests 1: Enables the generation of transmit buffer empty interrupt requests	R/W
b6	SPE	RSPI Function Enable	0: Disables the RSPI function 1: Enables the RSPI function	R/W
b7	SPRIE	RSPI Receive Buffer Full Interrupt Enable	0: Disables the generation of RSPI receive buffer full interrupt requests 1: Enables the generation of RSPI receive buffer full interrupt requests	R/W

Do not change the SPCR.MSTR, SPCR.MODFEN, or SPCR.TXMD bit while the SPCR.SPE bit is 1.

SPMS Bit (RSPI Mode Select)

The SPMS bit selects SPI operation (4-wire method) or clock synchronous operation (3-wire method).
The SSLA0, SSLA1, and SSLA3 pins are not used in clock synchronous operation. The RSPCKA, MOSIA, and MISOA pins handle communications. If clock synchronous operation is to proceed in master mode (SPCR.MSTR = 1), the SPCMDm.CPHA bit can be set to either 0 or 1. Set the CPHA bit to 1 if clock synchronous operation is to proceed in slave mode (SPCR.MSTR = 0). Do not set the CPHA bit to 0 when clock synchronous operation is to proceed in slave mode (SPCR.MSTR = 0).

TXMD Bit (Communications Operating Mode Select)

The TXMD bit selects full-duplex synchronous serial communications or transmit operations only.
When performing communications with the TXMD bit set to 1, the RSPI performs only transmit operations and not receive operations (refer to section 38.3.6, Communications Operating Mode).
When the TXMD bit is set to 1, receive buffer full interrupt requests cannot be used.

Downloaded from Arrow.com.

MODFEN Bit (Mode Fault Error Detection Enable)

The MODFEN bit enables or disables the detection of mode fault error (refer to section 38.3.8, Error Detection). In addition, the RSPI determines the I/O direction of the SSLA0, SSLA1, and SSLA3 pins based on combinations of the MODFEN and MSTR bits (refer to section 38.3.2, Controlling RSPI Pins).

MSTR Bit (RSPI Master/Slave Mode Select)

The MSTR bit selects master/slave mode of the RSPI. According to MSTR bit settings, the RSPI determines the direction of pins RSPCKA, MOSIA, MISOA, SSLA0, SSLA1, and SSLA3.

SPEIE Bit (RSPI Error Interrupt Enable)

The SPEIE bit enables or disables the generation of RSPI error interrupt requests when the RSPI detects a mode fault error and sets the SPSR.MODF flag to 1, when the RSPI detects an overrun error and sets the SPSR.OVRF flag to 1, or when the RSPI detects a parity error and sets the SPSR.PERF flag to 1 (refer to section 38.3.8, Error Detection).

SPTIE Bit (Transmit Buffer Empty Interrupt Enable)

The SPTIE bit enables or disables the generation of transmit buffer empty interrupt requests when the RSPI detects when the transmit buffer is empty.

A transmit buffer empty interrupt request when transmission starts is generated by setting the SPE and SPTIE bits to 1 at the same time or by setting the SPE bit to 1 after setting the SPTIE bit to 1.

Note that a transmit buffer interrupt is generated when the SPTIE bit is 1 even if the RSPI function is disabled (the SPTIE bit is changed to 0).

SPE Bit (RSPI Function Enable)

The SPE bit enables or disables the RSPI function.

When the SPSR.MODF flag is 1, the SPE bit cannot be set to 1. For details, refer to section 38.3.8, Error Detection.

Setting the SPE bit to 0 disables the RSPI function, and initializes a part of the module function. For details, refer to section 38.3.9, Initializing RSPI. Furthermore, a transmit buffer empty interrupt request is generated by the state of the SPE bit changing from 0 to 1 or from 1 to 0.

SPRIE Bit (RSPI Receive Buffer Full Interrupt Enable)

If the RSPI has detected a receive buffer full write after completion of a serial transfer, the SPRIE bit enables or disables the generation of an RSPI receive buffer full interrupt request.

Downloaded from Arrow.com.

38.2.2 RSPI Slave Select Polarity Register (SSLP)

Address(es): RSPI0.SSLP 0008 8381h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	SSL3P	—	SSL1P	SSL0P
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SSL0P	SSL0 Signal Polarity Setting	0: SSL0 signal is active low 1: SSL0 signal is active high	R/W
b1	SSL1P	SSL1 Signal Polarity Setting	0: SSL1 signal is active low 1: SSL1 signal is active high	R/W
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3	SSL3P	SSL3 Signal Polarity Setting	0: SSL3 signal is active low 1: SSL3 signal is active high	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Do not change the SSLP register while the SPCR.SPE bit is 1.

Downloaded from Arrow.com.

38.2.3 RSPI Pin Control Register (SPPCR)

Address(es): RSPI0.SPPCR 0008 8382h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	MOIFE	MOIFV	—	—	SPLP2	SPLP
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SPLP	RSPI Loopback	0: Normal mode 1: Loopback mode (data is inverted for transmission)	R/W
b1	SPLP2	RSPI Loopback 2	0: Normal mode 1: Loopback mode (data is not inverted for transmission)	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	MOIFV	MOSI Idle Fixed Value	0: The level output on the MOSIA pin during MOSI idling corresponds to low 1: The level output on the MOSIA pin during MOSI idling corresponds to high	R/W
b5	MOIFE	MOSI Idle Value Fixing Enable	0: MOSI output value equals final data from previous transfer 1: MOSI output value equals the value set in the MOIFV bit	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Do not change the SPPCR register while the SPCR.SPE bit is 1.

SPLP Bit (RSPI Loopback)

The SPLP bit selects the mode of the RSPI pins.

When the SPLP bit is set to 1, the RSPI shuts off the path between the MISOA pin and the shift register if the SPCR.MSTR bit is 1, and between the MOSIA pin and the shift register if the SPCR.MSTR bit is 0, and connects (inverts) the input path and output path for the shift register (loopback mode).

SPLP2 Bit (RSPI Loopback 2)

The SPLP2 bit selects the mode of the RSPI pins.

When the SPLP2 bit is set to 1, the RSPI shuts off the path between the MISOA pin and the shift register if the SPCR.MSTR bit is 1, and between the MOSIA pin and the shift register if the SPCR.MSTR bit is 0, and connects the input path and output path for the shift register (loopback mode).

MOIFV Bit (MOSI Idle Fixed Value)

If the MOIFE bit is 1 in master mode, the MOIFV bit determines the MOSIA pin output value during the SSL negation period (including the SSL retention period during a burst transfer).

MOIFE Bit (MOSI Idle Value Fixing Enable)

The MOIFE bit fixes the MOSIA output value when the RSPI in master mode is in an SSL negation period (including the SSL retention period during a burst transfer). When the MOIFE bit is 0, the RSPI outputs the last data from the previous serial transfer during the SSL negation period to the MOSIA pin. When the MOIFE bit is 1, the RSPI outputs the fixed value set in the MOIFV bit to the MOSIA pin.

Downloaded from Arrow.com.

38.2.4 RSPI Status Register (SPSR)

Address(es): RSPI0.SPSR 0008 8383h

	b7	b6	b5	b4	b3	b2	b1	b0
	SPRF	—	SPTEF	—	PERF	MODF	IDLNF	OVRF
Value after reset:	0	0	1	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	OVRF	Overrun Error Flag	0: No overrun error occurs 1: An overrun error occurs	R/(W) *1
b1	IDLNF	RSPI Idle Flag	0: RSPI is in the idle state 1: RSPI is in the transfer state	R
b2	MODF	Mode Fault Error Flag	0: No mode fault error occurs 1: A mode fault error occurs	R/(W) *1
b3	PERF	Parity Error Flag	0: No parity error occurs 1: A parity error occurs	R/(W) *1
b4	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5	SPTEF	Transmit Buffer Empty Flag	0: Transmit buffer has valid data 1: Transmit buffer has no valid data	R*2
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7	SPRF	Receive Buffer Full Flag	0: Receive buffer has no valid data 1: Receive buffer has valid data	R*2

Note 1. Only 0 can be written to clear the flag after reading 1.
Note 2. The write value should be 1.

OVRF Flag (Overrun Error Flag)

The OVRF flag indicates the occurrence of an overrun error. In master mode (when the SPCR.MSTR bit is 1) and when the RSPCK clock auto-stop function is enabled (the SPCR2.SCKASE bit is 1), an overrun error does not occur; accordingly this flag does not become 1. For details, refer to section 38.3.8.1, Overrun Error.

[Setting condition]

- When the next serial transfer ends while the SPCR.TXMD bit is 0 and the receive buffer is full.

[Clearing condition]

- When SPSR is read while the OVRF flag is 1, and then 0 is written to the OVRF flag.

IDLNF Flag (RSPI Idle Flag)

The IDLNF flag indicates the transfer status of the RSPI.

[Setting condition]

Master mode

- Condition 1 and condition 2 are not satisfied in master mode under the [Clearing condition] below.

Slave mode

- The SPCR.SPE bit is 1 (enables the RSPI function)

[Clearing condition]

Master mode

- The following 1 is satisfied (condition 1) or all of the following 2 to 4 are satisfied (condition 2).

1. The SPCR.SPE bit is 0 (disables the RSPI function)
2. The transmit buffer (SPTX) is empty (data for the next transfer is not set)
3. The SPSSR.SPCP[2:0] bits are 000b (beginning of sequence control)

Downloaded from Arrow.com.

4. The RSPI internal sequencer has entered the idle state (status in which operations up to the next-access delay have finished)

Slave mode

- The SPCR.SPE bit is 0 (disables the RSPI function)

MODF Flag (Mode Fault Error Flag)

Indicates the occurrence of a mode fault error.

[Setting condition]

Multi-master mode

- When the input level of the SSLAi pin changes to the active level while the SPCR.MSTR bit is 1 (master mode) and the SPCR.MODFEN bit is 1 (mode fault error detection is enabled), the RSPI detects a mode fault error

Slave mode

- When the SSLAi pin is negated before the RSPCK cycle necessary for data transfer ends while the SPCR.MSTR bit is 0 (slave mode) and the SPCR.MODFEN bit is 1 (mode fault error detection is enabled), the RSPI detects a mode fault error

The active level of the SSLAi signal is determined by the SSLP.SSLiP bit (SSLi signal polarity setting bit).

[Clearing condition]

- When SPSR is read while the MODF flag is 1, and then 0 is written to the MODF flag

PERF Flag (Parity Error Flag)

Indicates the occurrence of a parity error.

[Setting condition]

- When a serial transfer ends while the SPCR.TXMD bit is 0 and the SPCR2.SPPE bit is 1, the RSPI detects a parity error

[Clearing condition]

- When SPSR is read while the PERF flag is 1, and then 0 is written to the PERF flag

SPTEF Flag (Transmit Buffer Empty Flag)

Indicates whether the transmit buffer (SPTX) in the RSPI data register has valid data.

[Setting condition]

- When the SPCR.SPE bit is 0 (disables the RSPI function)
- When data is transferred from the transmit buffer to the shift register

[Clearing condition]

- When the number of frames of transmit data specified by the SPDCR.SPFC[1:0] bits is written to the SPDR register

The SPDR register can be set only when the SPTEF flag is 1. The data in the transmit buffer is not updated when the SPDR register is set while the SPTEF flag is 0.

SPRF Flag (Receive Buffer Full Flag)

Indicates whether the receive buffer (SPRX) in the RSPI data register has valid data.

[Setting condition]

- When the number of frames of receive data specified by the SPDCR.SPFC[1:0] bits is transferred from shift register to the receive buffer (SPRX) while the SPCR.TXMD bit is 0 (full duplex) and the SPRF flag is 0.
  Note that the SPRF flag does not become 1 when the OVRF flag is 1.

Downloaded from Arrow.com.

[Clearing condition]

- When all of the received data are read from the SPDR register

Downloaded from Arrow.com.

38.2.5 RSPI Data Register (SPDR)

- When accessing in longword size

Address(es): RSPI0.SPDR 0008 8384h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

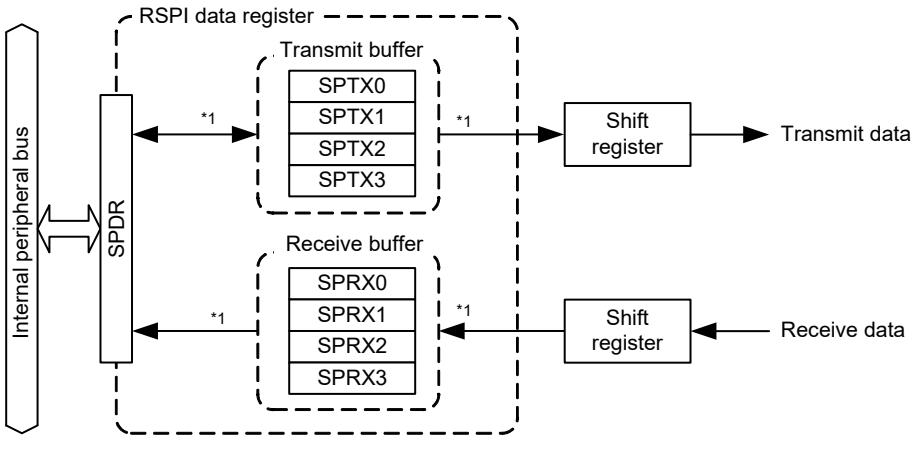
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

- When accessing in word size

Address(es): RSPI0.SPDR.H 0008 8384h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	H															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SPDR is the interface with the buffers that hold data for transmission and reception by the RSPI.
When accessing in longwords (the SPLW bit is 1), access SPDR in 32-bit units.
When accessing in words (the SPLW bit is 0), access SPDR.H in 16-bit units.
The transmit buffer (SPTX) and receive buffer (SPRX) are independent but are both mapped to SPDR. Figure 38.2 shows the Configuration of SPDR.



Note 1. The destination buffer and stage for access is automatically switched by hardware.

Figure 38.2 Configuration of SPDR

The transmit and receive buffers each have four stages. The number of stages to be used is selectable by the number of frames specification bits in the RSPI data control register (SPDCR.SPFC[1:0]). The eight stages of the buffer are all

Downloaded from Arrow.com.

mapped to the single address of SPDR.
Data written to SPDR are written to a transmit-buffer stage (SPTXn) (n = 0 to 3) and then transmitted from the buffer. The receive buffer holds received data on completion of reception. The receive buffer is not updated if an overrun is generated.
Furthermore, if the data length is other than 32 bits, bits not referred to in SPTXn (n = 0 to 3) are stored in the corresponding bits in SPRXn. For example, if the data length is 9 bits, received data are stored in the SPRXn[8:0] bits and the SPTXn[31:9] bits are stored in the SPRXn[31:9] bits.

(1) Bus Interface

SPDR is the interface with 32-bit wide transmit and receive buffers, each of which has four stages, for a total of 32 bytes. In other words, the 32 bytes are mapped to the 4-byte address space for SPDR. Furthermore, the unit of access for SPDR is selected by the SPDCR.SPLW bit.
Data for transmission should be flush with the LSB end of the register. Received data are stored flush with the LSB end. Operations involved in writing to and reading from SPDR are described below.

(a) Writing

Data written to SPDR are written to a transmit buffer (SPTXn). This is not influenced by the value of the SPDCR.SPRDTD bit unlike when reading from SPDR.
The transmit buffer includes a transmit buffer write pointer which is automatically updated to indicate the next stage each time data are written to SPDR.
Figure 38.3 shows the configuration of the bus interface with the transmit buffer in the case of writing to SPDR.

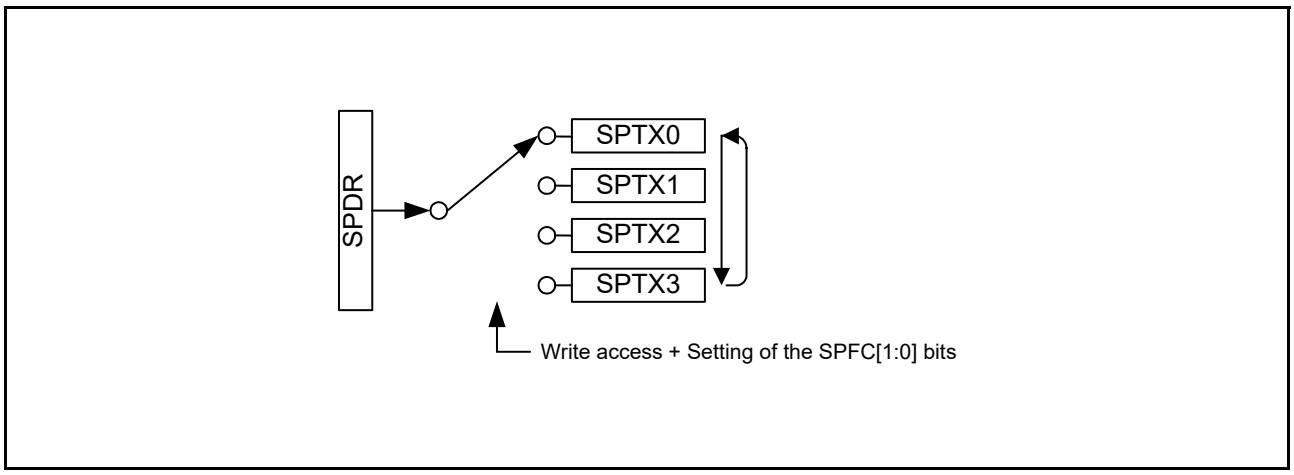


Figure 38.3 Configuration of SPDR (Writing)

The sequence for switching the transmit buffer write pointer differs with the setting of the number of frames specification bits in the RSPI data control register (SPDCR.SPFC[1:0]).

- Settings of the SPFC[1:0] bits and sequence of switching the pointer among SPTX0 to SPTX3.
  When the SPFC[1:0] bits are 00b: SPTX0 → SPTX0 → SPTX0 → …
  When the SPFC[1:0] bits are 01b: SPTX0 → SPTX1 → SPTX0 → SPTX1 → …
  When the SPFC[1:0] bits are 10b: SPTX0 → SPTX1 → SPTX2 → SPTX0 → SPTX1 → …
  When the SPFC[1:0] bits are 11b: SPTX0 → SPTX1 → SPTX2 → SPTX3 → SPTX0 → SPTX1 → …

When 1 is written to the RSPI function enable bit in the RSPI control register (SPCR.SPE) while the bit's current value is 0, SPTX0 will be the destination the next time writing proceeds.
When writing to the transmit buffer (SPTXn) after generation of the transmit buffer empty interrupt (after the SPSR.SPTEF flag becomes 1), write the number of frames set by the number of frames specification bits (SPFC[1:0]) in

Downloaded from Arrow.com.

the RSPI data control register (SPDCR). Even if the number of frames is written to the transmit buffer (SPTXn), the value of the buffer is not updated after completion of the writing and before generation of the next transmit buffer empty interrupt (while the SPSR.SPTEF flag is 0).

(b) Reading

SPDR can be read to read the value of a receive buffer (SPRXn) or a transmit buffer (SPTXn). The setting of the RSPI receive/transmit data select bit in the RSPI data control register (SPDCR.SPRDTD) selects whether reading is of the receive or transmit buffer.

The sequence of reading the SPDR register is controlled by independent pointers, receive buffer read pointer and transmit buffer read pointer.

Figure 38.4 shows the configuration of the bus interface with the receive and transmit buffers in the case of reading from SPDR.

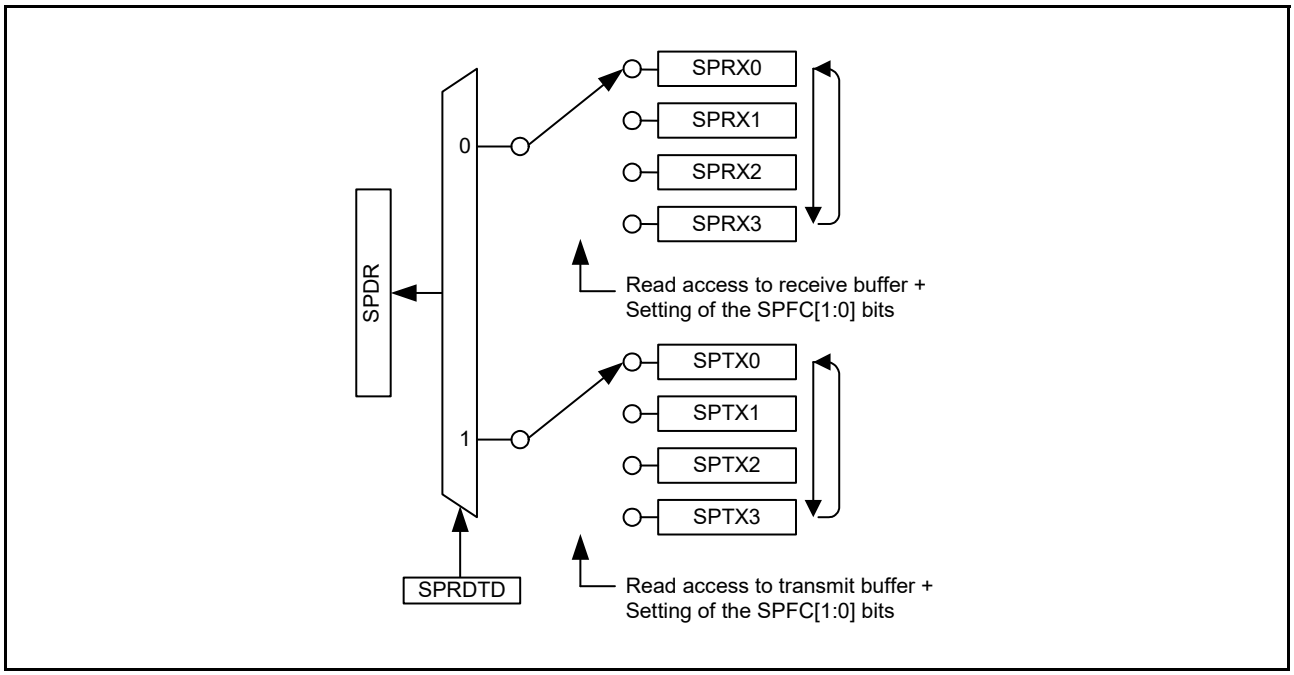


Figure 38.4 Configuration of SPDR (Reading)

Reading the receive buffer switches the receive buffer read pointer to the next buffer automatically.

The sequence of switching the receive buffer read pointer is the same as that for the transmit buffer write pointer. However, when 1 is written to the RSPI function enable bit in the RSPI control register (SPCR.SPE) while the bit's current value is 0, SPRX0 will be indicated by the buffer read pointer the next time reading proceeds.

The transmit buffer read pointer is updated when writing to SPDR, and not updated when reading from the transmit buffer. When reading from the transmit buffer, the value most recently written to SPDR is read. However, after generation of the transmit buffer empty interrupt, the values read from the transmit buffer are all 0 in the interval after completion of writing the number of frames of data specified in the number of frames specification bits (SPDCR.SPFC[1:0]) and before generation of the next buffer empty interrupt (while the SPSR.SPTEF flag is 0).

Downloaded from Arrow.com.

38.2.6 RSPI Sequence Control Register (SPSCR)

Address(es): RSPI0.SPSCR 0008 8388h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	SPSLN[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	SPSLN[2:0]	RSPI Sequence Length Specification	b2 b0 Sequence Length — Referenced SPCMD0 to SPCMD7 (No.) 0 0 0: 1 — 0→0→… 0 0 1: 2 — 0→1→0→… 0 1 0: 3 — 0→1→2→0→… 0 1 1: 4 — 0→1→2→3→0→… 1 0 0: 5 — 0→1→2→3→4→0→… 1 0 1: 6 — 0→1→2→3→4→5→0→… 1 1 0: 7 — 0→1→2→3→4→5→6→0→… 1 1 1: 8 — 0→1→2→3→4→5→6→7→0→… The order in which the SPCMD0 to SPCMD7 registers are to be referenced is changed according to the sequence length that is set in these bits. The relationship among the setting of these bits, sequence length, and SPCMD0 to SPCMD7 registers referenced by the RSPI is shown above. However, the RSPI in slave mode references SPCMD0.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

SPSCR sets the sequence length when the RSPI operates in master mode. When changing the SPSCR.SPSLN[2:0] bits while both the SPCR.MSTR and SPCR.SPE bits are 1, the bits should be changed while the SPSR.IDLNF flag is 0.

SPSLN[2:0] Bits (RSPI Sequence Length Specification)

The SPSLN[2:0] bits specify a sequence length when the RSPI in master mode performs sequential operations. The RSPI in master mode changes SPCMD0 to SPCMD7 registers to be referenced and the order in which they are referenced according to the sequence length that is set in the SPSLN[2:0] bits.
In slave mode, SPCMD0 is referred.

Downloaded from Arrow.com.

38.2.7 RSPI Sequence Status Register (SPSSR)

Address(es): RSPI0.SPSSR 0008 8389h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	SPECM[2:0]			—	SPCP[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	SPCP[2:0]	RSPI Command Pointer	b2 b0 0 0 0: SPCMD0 0 0 1: SPCMD1 0 1 0: SPCMD2 0 1 1: SPCMD3 1 0 0: SPCMD4 1 0 1: SPCMD5 1 1 0: SPCMD6 1 1 1: SPCMD7	R
b3	—	Reserved	This bit is read as 0.	R
b6 to b4	SPECM[2:0]	RSPI Error Command	b6 b4 0 0 0: SPCMD0 0 0 1: SPCMD1 0 1 0: SPCMD2 0 1 1: SPCMD3 1 0 0: SPCMD4 1 0 1: SPCMD5 1 1 0: SPCMD6 1 1 1: SPCMD7	R
b7	—	Reserved	This bit is read as 0.	R

SPSSR indicates the sequence control status when the RSPI operates in master mode.
Any writing to SPSSR is ignored.

SPCP[2:0] Bits (RSPI Command Pointer)

The SPCP[2:0] bits indicate SPCMDm that is currently pointed to by the pointer during sequence control by the RSPI. For the RSPI's sequence control, refer to section 38.3.10.1, Master Mode Operation.

SPECM[2:0] Bits (RSPI Error Command)

The SPECM[2:0] bits indicate SPCMDm that is specified by the SPCP[2:0] bits when an error is detected during sequence control by the RSPI. The RSPI updates the SPECM[2:0] bits only when an error is detected. If both the SPSR.OVRF and SPSR.MODF flags are 0 and there is no error, the values of the SPECM[2:0] bits have no meaning. For the RSPI's error detection function, refer to section 38.3.8, Error Detection. For the RSPI's sequence control, refer to section 38.3.10.1, Master Mode Operation.

Downloaded from Arrow.com.

38.2.8 RSPI Bit Rate Register (SPBR)

Address(es): RSPI0.SPBR 0008 838Ah



SPBR sets the bit rate in master mode. Do not change the SPBR register while both the SPCR.MSTR and SPCR.SPE bits are 1.

When the RSPI is used in slave mode, the bit rate depends on the bit rate of the input clock (bit rate satisfying the electrical characteristics should be used) regardless of the settings of SPBR and the SPCMDm.BRDV[1:0] bits (bit rate division setting bits).
The bit rate is determined by combinations of the SPBR setting and the SPCMDm.BRDV[1:0] bit setting. The equation for calculating the bit rate is given below. In the equation, n denotes an SPBR setting (0, 1, 2, …, 255), and N denotes a BRDV[1:0] bit setting (0, 1, 2, 3).

$$\text{Bit rate} = \frac{f(PCLK)}{2 \times (n+1) \times 2^N}$$

Table 38.3 lists examples of the relationship among the SPBR settings, the BRDV[1:0] settings, and bit rates.
Use the bit rate that meets electrical characteristics based on the AC specifications of the target device.

Table 38.3 Relationship among SPBR Settings, BRDV[1:0] Settings, and Bit Rates

SPBR (n)	BRDV[1:0] Bits (N)	Division Ratio	Bit Rate
			PCLK = 32 MHz
0	0	2	16.0 Mbps
1	0	4	8.00 Mbps
2	0	6	5.33 Mbps
3	0	8	4.00 Mbps
4	0	10	3.20 Mbps
5	0	12	2.67 Mbps
5	1	24	1.33 Mbps
5	2	48	667 kbps
5	3	96	333 kbps
255	3	4096	7.81 kbps

Downloaded from Arrow.com.

38.2.9 RSPI Data Control Register (SPDCR)

Address(es): RSPI0.SPDCR 0008 838Bh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	SPLW	SPRDTD	—	—	SPFC[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	SPFC[1:0]	Number of Frames Specification	b1 b0 0 0: 1 frame 0 1: 2 frames 1 0: 3 frames 1 1: 4 frames	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	SPRDTD	RSPI Receive/Transmit Data Select	0: SPDR values are read from the receive buffer 1: SPDR values are read from the transmit buffer (but only if the transmit buffer is empty)	R/W
b5	SPLW	RSPI Longword Access/ Word Access Specification	0: SPDR is accessed in words 1: SPDR is accessed in longwords	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Up to four frames can be transmitted or received in one round of transmission or reception activation. The amount of data in each transfer is controlled by the combination of the SPCMDm.SPB[3:0] bits, the SPSCR.SPSLN[2:0] bits, and the SPDCR.SPFC[1:0] bits.
When changing the SPDCR.SPFC[1:0] bits while the SPCR.SPE bit is 1, the bits should be changed while the SPSR.IDLNF flag is 0.

SPFC[1:0] Bits (Number of Frames Specification)

The SPFC[1:0] bits specify the number of frames that can be stored in SPDR (per transfer activation). Up to four frames can be transmitted or received in one round of transmission or reception, and the amount of data is determined by the combination of the SPSCR.SPSLN[2:0] bits, and the SPDCR.SPFC[1:0] bits. Furthermore, the setting of the SPFC[1:0] bits adjusts the number of frames for generation of RSPI receive buffer full interrupt, and start of transmission or generation of transmit buffer empty interrupts.
When the number of frames of transmit data specified by SPFC[1:0] bits is written to the SPDR register, the SPSR.SPTEF flag becomes 0 and transmission starts. Then, when the specified number of frames of transmit data has been transferred to the shift register, the SPTEF flag becomes 1 and the RSPI transmit buffer empty interrupt is generated.
When the number of frames specified by the SPFC[1:0] bits are received, the SPSR.SPRF flag becomes 1 and the RSPI receive buffer full interrupt is generated.
Table 38.4 lists the frame configurations that can be stored in SPDR and examples of combinations of settings for transmission and reception. Do not select the combinations of settings other than those shown in the examples.

Downloaded from Arrow.com.

Table 38.4 Settable Combinations of SPSLN[2:0] Bits and SPFC[1:0] Bits

Setting	SPSLN[2:0]	SPFC[1:0]	Number of Frames in a Single Sequence	Number of Frames at which Transmit Buffer or Receive Buffer Status Becomes “Has Valid Data”
1-1	000b	00b	1	1
1-2	000b	01b	2	2
1-3	000b	10b	3	3
1-4	000b	11b	4	4
2-1	001b	01b	2	2
2-2	001b	11b	4	4
3	010b	10b	3	3
4	011b	11b	4	4
5	100b	00b	5	1
6	101b	00b	6	1
7	110b	00b	7	1
8	111b	00b	8	1

SPRDTD Bit (RSPI Receive/Transmit Data Select)

The SPRDTD bit selects whether the SPDR reads values from the receive buffer or from the transmit buffer.
If reading is from the transmit buffer, the value written to SPDR register immediately beforehand is read.
When reading the transmit buffer, do so before writing of the number of frames set in the SPFC[1:0] bits is finished and after generation of the transmit buffer empty interrupt (While the SPSR.SPTEF flag is 1).
For details, refer to section 38.2.5, RSPI Data Register (SPDR).

SPLW Bit (RSPI Longword Access/Word Access Specification)

The SPLW bit specifies the access width for SPDR. Access to the SPDR register in words when the SPLW bit is 0 and in longwords when the SPLW bit is 1.
Also, when the SPLW bit is 0, set the SPCMDm.SPB[3:0] bits (RSPI data length setting bits) to 8 to 16 bits. Do not select 20, 24, or 32 bits.

Downloaded from Arrow.com.

38.2.10 RSPI Clock Delay Register (SPCKD)

Address(es): RSPI0.SPCKD 0008 838Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	SCKDL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	SCKDL[2:0]	RSPCK Delay Setting	b2 b0 0 0 0: 1 RSPCK 0 0 1: 2 RSPCK 0 1 0: 3 RSPCK 0 1 1: 4 RSPCK 1 0 0: 5 RSPCK 1 0 1: 6 RSPCK 1 1 0: 7 RSPCK 1 1 1: 8 RSPCK	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

SPCKD sets a period from the beginning of SSLAi signal assertion to RSPCK oscillation (RSPCK delay) when the SPCMDm.SCKDEN bit is 1. Do not change the SPCKD register while both the SPCR.MSTR and SPCR.SPE bits are 1.

SCKDL[2:0] Bits (RSPCK Delay Setting)

The SCKDL[2:0] bits set an RSPCK delay value when the SPCMDm.SCKDEN bit is 1.
When using the RSPI in slave mode, set the SCKDL[2:0] bits to 000b.

Downloaded from Arrow.com.

38.2.11 RSPI Slave Select Negation Delay Register (SSLND)

Address(es): RSPI0.SSLND 0008 838Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	SLNDL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	SLNDL[2:0]	SSL Negation Delay Setting	b2 b0 0 0 0: 1 RSPCK 0 0 1: 2 RSPCK 0 1 0: 3 RSPCK 0 1 1: 4 RSPCK 1 0 0: 5 RSPCK 1 0 1: 6 RSPCK 1 1 0: 7 RSPCK 1 1 1: 8 RSPCK	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

SSLND sets a period (SSL negation delay) from the transmission of a final RSPCK edge to the negation of the SSLAi signal during a serial transfer by the RSPI in master mode. Do not change the SSLND register while both the SPCR.MSTR and SPCR.SPE bits are 1.

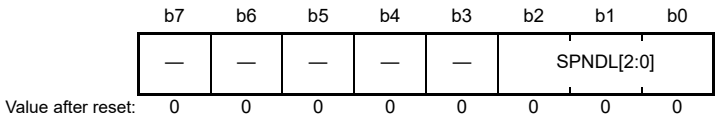
SLNDL[2:0] Bits (SSL Negation Delay Setting)

The SLNDL[2:0] bits set an SSL negation delay value when the RSPI is in master mode.
When using the RSPI in slave mode, set the SLNDL[2:0] bits to 000b.

Downloaded from Arrow.com.

38.2.12 RSPI Next-Access Delay Register (SPND)

Address(es): RSPI0.SPND 0008 838Eh



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	SPNDL[2:0]	RSPI Next-Access Delay Setting	b2 b0 0 0 0: 1 RSPCK + 2 PCLK 0 0 1: 2 RSPCK + 2 PCLK 0 1 0: 3 RSPCK + 2 PCLK 0 1 1: 4 RSPCK + 2 PCLK 1 0 0: 5 RSPCK + 2 PCLK 1 0 1: 6 RSPCK + 2 PCLK 1 1 0: 7 RSPCK + 2 PCLK 1 1 1: 8 RSPCK + 2 PCLK	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

SPND sets a non-active period (next-access delay) of the SSLAi signal after termination of a serial transfer when the SPCMDm.SPNDEN bit is 1. Do not change the SPND register while both the SPCR.MSTR and SPCR.SPE bits are 1.

SPNDL[2:0] Bits (RSPI Next-Access Delay Setting)

The SPNDL[2:0] bits set a next-access delay when the SPCMDm.SPNDEN bit is 1.
When using the RSPI in slave mode, set the SPNDL[2:0] bits to 000b.

Downloaded from Arrow.com.

38.2.13 RSPI Control Register 2 (SPCR2)

Address(es): RSPI0.SPCR2 0008 838Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	SCKASE	PTE	SPIIE	SPOE	SPPE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SPPE	Parity Enable	0: Does not add the parity bit to transmit data and does not check the parity bit of receive data 1: Adds the parity bit to transmit data and checks the parity bit of receive data (when SPCR.TXMD = 0) Adds the parity bit to transmit data but does not check the parity bit of receive data (when SPCR.TXMD = 1)	R/W
b1	SPOE	Parity Mode	0: Selects even parity for use in transmission and reception 1: Selects odd parity for use in transmission and reception	R/W
b2	SPIIE	RSPI Idle Interrupt Enable	0: Disables the generation of idle interrupt requests 1: Enables the generation of idle interrupt requests	R/W
b3	PTE	Parity Self-Diagnosis	0: Disables the self-diagnosis function of the parity circuit 1: Enables the self-diagnosis function of the parity circuit	R/W
b4	SCKASE	RSPCK Auto-Stop Function Enable	0: Disables the RSPCK auto-stop function 1: Enables the RSPCK auto-stop function	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Do not change the SPPE, SPOE, and SCKASE bits in the SPCR2 register while the SPCR.SPE bit is 1.

SPPE Bit (Parity Enable)

The SPPE bit enables or disables the parity function.

The parity bit is added to transmit data and parity checking is performed for receive data when the SPCR.TXMD bit is 0 and the SPCR2.SPPE bit is 1.

The parity bit is added to transmit data but parity checking is not performed for receive data when the SPCR.TXMD bit is 1 and the SPCR2.SPPE bit is 1.

SPOE Bit (Parity Mode)

The SPOE bit specifies odd or even parity.

When even parity is set, parity bit addition is performed so that the total number of 1-bits in the transmit/receive character plus the parity bit is even. Similarly, when odd parity is set, parity bit addition is performed so that the total number of 1-bits in the transmit/receive character plus the parity bit is odd.

The SPOE bit is valid only when the SPPE bit is 1.

SPIIE Bit (RSPI Idle Interrupt Enable)

The SPIIE bit enables or disables the generation of RSPI idle interrupt requests when the RSPI being in the idle state is detected and the SPSR.IDLNF flag is set to 0.

PTE Bit (Parity Self-Diagnosis)

The PTE bit enables the self-diagnosis function of the parity circuit in order to check whether the parity function is operating correctly.

Downloaded from Arrow.com.

SCKASE Bit (RSPCK Auto-Stop Function Enable)

The SCKASE bit enables or disables the RSPCK auto-stop function. When this function is enabled, the RSPCK clock is stopped before an overrun error occurs when data is received in master mode. For details, refer to section 38.3.8.1, Overrun Error.

Downloaded from Arrow.com.

38.2.14 RSPI Command Register m (SPCMDm) (m = 0 to 7)

Address(es): RSPI0.SPCMD0 0008 8390h, RSPI0.SPCMD1 0008 8392h, RSPI0.SPCMD2 0008 8394h, RSPI0.SPCMD3 0008 8396h, RSPI0.SPCMD4 0008 8398h, RSPI0.SPCMD5 0008 839Ah, RSPI0.SPCMD6 0008 839Ch, RSPI0.SPCMD7 0008 839Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB[3:0]				SSLKP	SSLA[2:0]			BRDV[1:0]		CPOL	CPHA
Value after reset:	0	0	0	0	0	1	1	1	0	0	0	0	1	1	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	CPHA	RSPCK Phase Setting	0: Data sampling on odd edge, data variation on even edge 1: Data variation on odd edge, data sampling on even edge	R/W
b1	CPOL	RSPCK Polarity Setting	0: RSPCK is low when idle 1: RSPCK is high when idle	R/W
b3, b2	BRDV[1:0]	Bit Rate Division Setting	b3 b2 0 0: These bits select the base bit rate 0 1: These bits select the base bit rate divided by 2 1 0: These bits select the base bit rate divided by 4 1 1: These bits select the base bit rate divided by 8	R/W
b6 to b4	SSLA[2:0]	SSL Signal Assertion Setting	b6 b4 0 0 0: SSL0 0 0 1: SSL1 0 1 1: SSL3 1 x x: Setting prohibited	R/W
b7	SSLKP	SSL Signal Level Keeping	0: Negates all SSL signals upon completion of transfer 1: Keeps the SSL signal level from the end of transfer until the beginning of the next access	R/W
b11 to b8	SPB[3:0]	RSPI Data Length Setting	b11 b8 0100 to 0111: 8 bits 1 0 0 0: 9 bits 1 0 0 1: 10 bits 1 0 1 0: 11 bits 1 0 1 1: 12 bits 1 1 0 0: 13 bits 1 1 0 1: 14 bits 1 1 1 0: 15 bits 1 1 1 1: 16 bits 0 0 0 0: 20 bits 0 0 0 1: 24 bits 0010, 0011: 32 bits	R/W
b12	LSBF	RSPI LSB First	0: MSB first 1: LSB first	R/W
b13	SPNDEN	RSPI Next-Access Delay Enable	0: A next-access delay of 1 RSPCK + 2 PCLK 1: A next-access delay is equal to the setting of the RSPI next-access delay register (SPND)	R/W
b14	SLNDEN	SSL Negation Delay Setting Enable	0: An SSL negation delay of 1 RSPCK 1: An SSL negation delay is equal to the setting of the RSPI slave select negation delay register (SSLND)	R/W
b15	SCKDEN	RSPCK Delay Setting Enable	0: An RSPCK delay of 1 RSPCK 1: An RSPCK delay is equal to the setting of the RSPI clock delay register (SPCKD)	R/W

x: Don’t care

Downloaded from Arrow.com.

SPCMDm register is used to set a transfer format for the RSPI in master mode. Each channel has eight RSPI command registers (SPCMD0 to SPCMD7). Some of the bits in SPCMD0 register is used to set a transfer mode for the RSPI in slave mode. The RSPI in master mode sequentially references SPCMDm register according to the settings in the SPSCR.SPSLN[2:0] bits, and executes the serial transfer that is set in the referenced SPCMDm register.
SPCMDm register should be set while the transmit buffer is empty (data for the next transfer is not set) and before setting of the data that is to be transmitted when that SPCMDm register is referenced.
SPCMDm that is referenced by the RSPI in master mode can be checked by means of the SPSSR.SPCP[2:0] bits. Do not change the SPCMDm register while the SPCR.MSTR bit is 0 and the SPCR.SPE bit is 1.

CPHA Bit (RSPCK Phase Setting)

The CPHA bit sets an RSPCK phase of the RSPI in master mode or slave mode. Data communications between RSPI modules require the same RSPCK phase setting between the modules.

CPOL Bit (RSPCK Polarity Setting)

The CPOL bit sets an RSPCK polarity of the RSPI in master mode or slave mode. Data communications between RSPI modules require the same RSPCK polarity setting between the modules.

BRDV[1:0] Bits (Bit Rate Division Setting)

The BRDV[1:0] bits are used to determine the bit rate. A bit rate is determined by combinations of the settings in the BRDV[1:0] bits and SPBR (refer to section 38.2.8, RSPI Bit Rate Register (SPBR)). The settings in SPBR determine the base bit rate. The settings in the BRDV[1:0] bits are used to select a bit rate which is obtained by dividing the base bit rate by 1, 2, 4, or 8. In SPCMDm register, different BRDV[1:0] bit settings can be specified. This enables execution of serial transfers at a different bit rate for each command.

SSLA[2:0] Bits (SSL Signal Assertion Setting)

The SSLA[2:0] bits control the SSLAi signal assertion when the RSPI performs serial transfers in master mode.
Setting the SSLA[2:0] bits controls the assertion for the SSLAi signal. When an SSLAi signal is asserted, its polarity is determined by the set value in the corresponding SSLP. When the SSLA[2:0] bits are set to 000b in multi-master mode, serial transfers are performed with all the SSL signals in the negated state (as the SSLA0 pin acts as input).
When using the RSPI in slave mode, set the SSLA[2:0] bits to 000b.

SSLKP Bit (SSL Signal Level Keeping)

When the RSPI in master mode performs a serial transfer, the SSLKP bit specifies whether the SSLAi signal level for the current command is to be kept or negated between the SSL negation timing associated with the current command and the SSL assertion timing associated with the next command.
Setting the SSLKP bit to 1 enables a burst transfer. For details, refer to section 38.3.10.1, Master Mode Operation (4) Burst Transfer.
When using the RSPI in slave mode, the SSLKP bit should be set to 0.

SPB[3:0] Bits (RSPI Data Length Setting)

The SPB[3:0] bits set a transfer data length for the RSPI in master mode or slave mode. When the SPDCR.SPLW bit is 0, set the SPB[3:0] bits to 0100b (8 bits) to 1111b (16 bits).

LSBF Bit (RSPI LSB First)

The LSBF bit sets the data format of the RSPI in master mode or slave mode to MSB first or LSB first.

Downloaded from Arrow.com.

SPNDEN Bit (RSPI Next-Access Delay Enable)

The SPNDEN bit sets the period from the time the RSPI in master mode terminates a serial transfer and sets the SSLAi signal inactive until the RSPI enables the SSLAi signal assertion for the next access (next-access delay). If the SPNDEN bit is 0, the RSPI sets the next-access delay to 1 RSPCK + 2 PCLK. If the SPNDEN bit is 1, the RSPI inserts a next-access delay in compliance with the SPND setting.

When using the RSPI in slave mode, the SPNDEN bit should be set to 0.

SLNDEN Bit (SSL Negation Delay Setting Enable)

The SLNDEN bit sets the period from the time the RSPI in master mode stops RSPCK oscillation until the RSPI sets the SSLAi signal inactive (SSL negation delay). If the SLNDEN bit is 0, the RSPI sets the SSL negation delay to 1 RSPCK. If the SLNDEN bit is 1, the RSPI negates the SSL signal at an SSL negation delay in compliance with the SSLND setting.

When using the RSPI in slave mode, the SLNDEN bit should be set to 0.

SCKDEN Bit (RSPCK Delay Setting Enable)

The SCKDEN bit sets the period from the point when the RSPI in master mode activates the SSLAi signal until the RSPCK starts oscillation (RSPI clock delay). If the SCKDEN bit is 0, the RSPI sets the RSPCK delay to 1 RSPCK. If the SCKDEN bit is 1, the RSPI starts the oscillation of RSPCK at an RSPCK delay in compliance with the SPCKD setting.

When using the RSPI in slave mode, the SCKDEN bit should be set to 0.

Downloaded from Arrow.com.

38.3 Operation

In this section, the serial transfer period means a period from the beginning of driving valid data to the fetching of the final valid data.

38.3.1 Overview of RSPI Operations

The RSPI is capable of synchronous serial transfers in slave mode (SPI operation), single-master mode (SPI operation), multi-master mode (SPI operation), slave mode (clock synchronous operation), and master mode (clock synchronous operation). A particular mode of the RSPI can be selected by using the MSTR, MODFEN, and SPMS bits in SPCR. Table 38.5 lists the relationship between RSPI modes and SPCR settings, and a description of each mode.

Table 38.5 Relationship between RSPI Modes and SPCR Settings and Description of Each Mode

	SPI Operation			Clock Synchronous Operation	
Mode	**Slave**	**Single-Master**	**Multi-Master**	**Slave**	**Master**
MSTR bit setting	0	1	1	0	1
MODFEN bit setting	0 or 1	0	1	0	0
SPMS bit setting	0	0	0	1	1
RSPCKA signal	Input	Output	Output/Hi-Z	Input	Output
MOSIA signal	Input	Output	Output/Hi-Z	Input	Output
MISOA signal	Output/Hi-Z	Input	Input	Output	Input
SSLA0 signal	Input	Output	Input	Hi-Z*1	Hi-Z*1
SSLA1, SSLA3 signals	Hi-Z*1	Output	Output/Hi-Z	Hi-Z*1	Hi-Z*1
SSL polarity change function	Supported	Supported	Supported	—	—
Transfer rate	Up to PCLK/8	Up to PCLK/2	Up to PCLK/2	Up to PCLK/8	Up to PCLK/2
Clock source	RSPCK input	On-chip baud rate generator	On-chip baud rate generator	RSPCK input	On-chip baud rate generator
Clock polarity	Two				
Clock phase	Two	Two	Two	One (CPHA = 1)	Two
First transfer bit	MSB/LSB				
Transfer data length	8 to 16, 20, 24, 32 bits				
Burst transfer	Possible (CPHA = 1)	Possible (CPHA = 0,1)	Possible (CPHA = 0,1)	—	—
RSPCK delay control	Not supported	Supported	Supported	Not supported	Supported
SSL negation delay control	Not supported	Supported	Supported	Not supported	Supported
Next-access delay control	Not supported	Supported	Supported	Not supported	Supported
Transfer activation method	SSL input active or RSPCK oscillation	Transmit buffer is written to when a transmit buffer empty interrupt request is generated or the SPTEF flag is 1	Transmit buffer is written to when a transmit buffer empty interrupt request is generated or the SPTEF flag is 1	RSPCK oscillation	Transmit buffer is written to when a transmit buffer empty interrupt request is generated or the SPTEF flag is 1
Sequence control	Not supported	Supported	Supported	Not supported	Supported
Transmit buffer empty detection	Supported				
Receive buffer full detection	Supported*2				
Overrun error detection	Supported*2	Supported*2, *4	Supported*2, *4	Supported*2	Supported*2
Parity error detection	Supported*2, *3				
Mode fault error detection	Supported (MODFEN = 1)	Not supported	Supported	Not supported	Not supported

Note 1. This function is not supported in this mode.
Note 2. When the SPCR.TXMD bit is 1, receiver buffer full detection, overrun error detection, and parity error detection are not performed.
Note 3. When the SPCR2.SPPE bit is 0, parity error detection is not performed.
Note 4. When the SPCR2.SCKASE bit is 1, overrun error detection does not proceed.

Downloaded from Arrow.com.

38.3.2 Controlling RSPI Pins

According to the MSTR, MODFEN, and SPMS bits in SPCR and the ODRn.Bi bit for I/O ports, the RSPI can switch pin states. Table 38.6 lists the relationship between pin states and bit settings. Setting the ODRn.Bi bit for an I/O port to 0 selects CMOS output; setting it to 1 selects open-drain output. The I/O port settings should follow this relationship.

Table 38.6 Relationship between Pin States and Bit Settings

Mode	Pin	Pin State*2	
		ODRn.Bi Bit for I/O Ports = 0	ODRn.Bi Bit for I/O Ports = 1
Single-master mode (SPI operation) (MSTR = 1, MODFEN = 0, SPMS = 0)	RSPCKA	CMOS output	Open-drain output
	SSLA0, SSLA1, SSLA3	CMOS output	Open-drain output
	MOSIA	CMOS output	Open-drain output
	MISOA	Input	Input
Multi-master mode (SPI operation) (MSTR = 1, MODFEN = 1, SPMS = 0)	RSPCKA*3	CMOS output/Hi-Z	Open-drain output/Hi-Z
	SSLA0	Input	Input
	SSLA1, SSLA3*3	CMOS output/Hi-Z	Open-drain output/Hi-Z
	MOSIA*3	CMOS output/Hi-Z	Open-drain output/Hi-Z
	MISOA	Input	Input
Slave mode (SPI operation) (MSTR = 0, SPMS = 0)	RSPCKA	Input	Input
	SSLA0	Input	Input
	SSLA1, SSLA3*5	Hi-Z*1	Hi-Z*1
	MOSIA	Input	Input
	MISOA*4	CMOS output/Hi-Z	Open-drain output/Hi-Z
Master mode (Clock synchronous operation) (MSTR = 1, MODFEN = 0, SPMS = 1)	RSPCKA	CMOS output	Open-drain output
	SSLA0, SSLA1, SSLA3*5	Hi-Z*1	Hi-Z*1
	MOSIA	CMOS output	Open-drain output
	MISOA	Input	Input
Slave mode (Clock synchronous operation) (MSTR = 0, SPMS = 1)	RSPCKA	Input	Input
	SSLA0, SSLA1, SSLA3*5	Hi-Z*1	Hi-Z*1
	MOSIA	Input	Input
	MISOA	CMOS output	Open-drain output

Note 1. This function is not supported in this mode.
Note 2. RSPI settings are not reflected in the multiplex pins for which the RSPI function is not selected.
Note 3. When SSLA0 is at the active level, the pin state is Hi-Z.
Note 4. When SSLA0 is at the non-active level or the SPCR.SPE bit is 0, the pin state is Hi-Z.
Note 5. These pins are available for use as I/O port pins.

The RSPI in single-master mode (SPI operation) or multi-master mode (SPI operation) determines MOSI signal values during the SSL negation period (including the SSL retention period during a burst transfer) according to MOIFE and MOIFV bit settings in SPPCR, as listed in Table 38.7.

Table 38.7 MOSI Signal Value Determination during SSL Negation Period

MOIFE Bit	MOIFV Bit	MOSIA Signal Value during SSL Negation Period
0	0, 1	Final data from previous transfer
1	0	Low
1	1	High

Downloaded from Arrow.com.

38.3.3 RSPI System Configuration Examples

38.3.3.1 Single Master/Single Slave (with This MCU Acting as Master)

Figure 38.5 shows a single-master/single-slave RSPI system configuration example when this MCU is used as a master. In the single-master/single-slave configuration, the SSLA0, SSLA1, and SSLA3 output of this MCU (master) are not used. The SSL input of the SPI slave is fixed to the low level, and the SPI slave is maintained in a select state.*1
This MCU (master) drives the RSPCKA and MOSIA. The SPI slave drives the MISO.

Note 1. In the transfer format corresponding to the case where the SPCMDm.CPHA bit is 0, there are slave devices for which the SSL signal cannot be fixed to the active level. In situations where the SSL signal cannot be fixed, the SSLAi output of this MCU should be connected to the SSL input of the slave device.

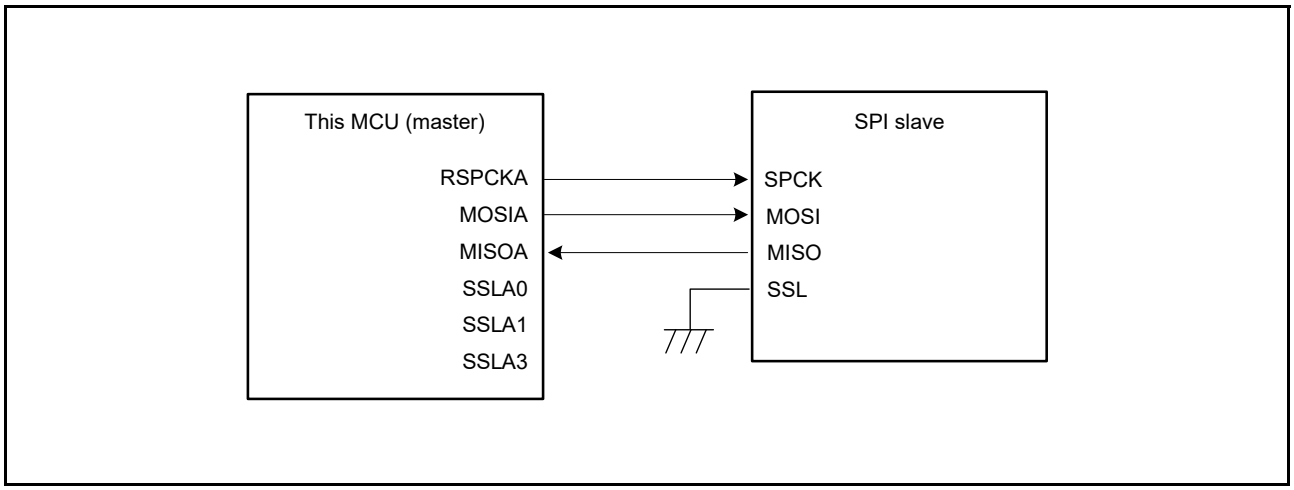


Figure 38.5 Single-Master/Single-Slave Configuration Example (This MCU = Master)

Downloaded from Arrow.com.

38.3.3.2 Single Master/Single Slave (with This MCU Acting as Slave)

Figure 38.6 shows a single-master/single-slave RSPI system configuration example when this MCU is used as a slave. When this MCU is to operate as a slave, the SSLA0 pin is used as SSL input. The SPI master drives the RSPCK and MOSI. This MCU (slave) drives the MISOA.*1

In the single-slave configuration in which the SPCMDm.CPHA bit is set to 1, the SSLA0 input of this MCU (slave) is fixed to the low level, this MCU (slave) is maintained in a select state, and in this manner it is possible to execute serial transfer (Figure 38.7).

Note 1. When SSLA0 is at the non-active level, the pin state is Hi-Z.

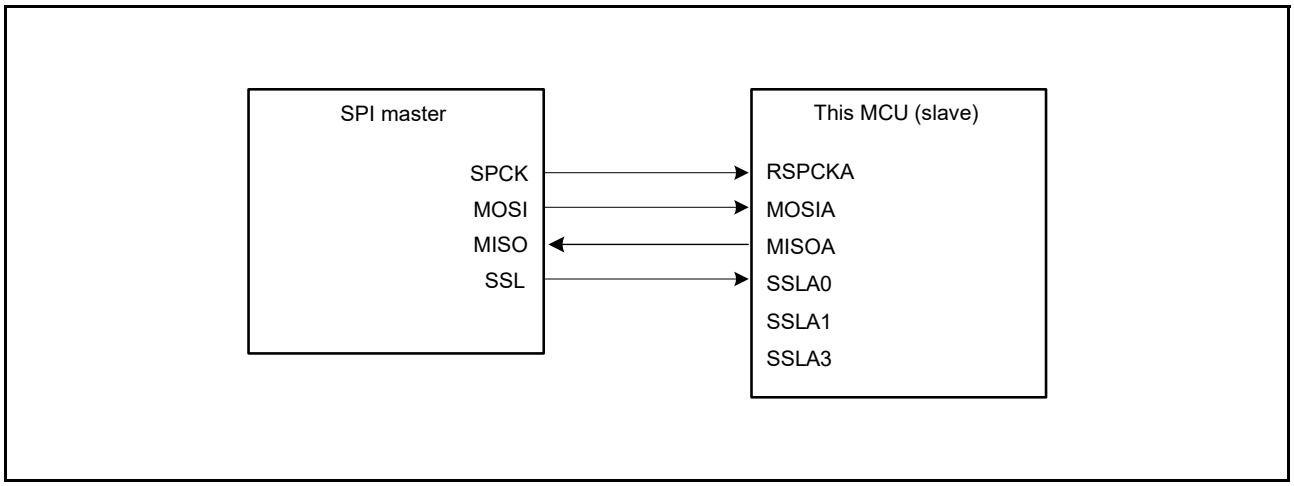


Figure 38.6 Single-Master/Single-Slave Configuration Example (This MCU = Slave, CPHA = 0)

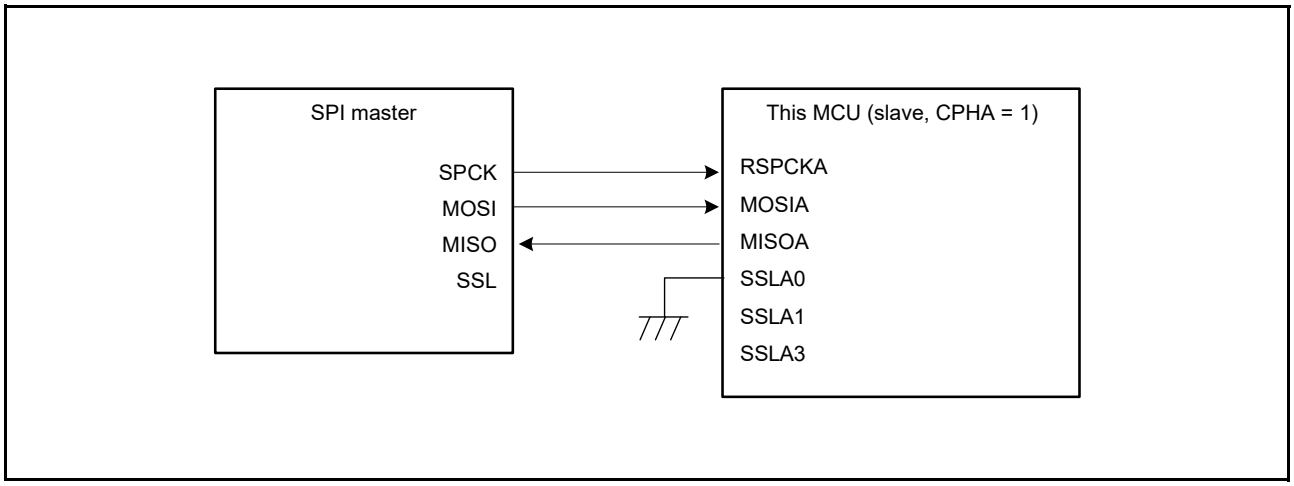


Figure 38.7 Single-Master/Single-Slave Configuration Example (This MCU = Slave, CPHA = 1)

Downloaded from Arrow.com.

38.3.3.3 Single Master/Multi-Slave (with This MCU Acting as Master)

Figure 38.8 shows a single-master/multi-slave RSPI system configuration example when this MCU is used as a master. In the example of Figure 38.8, the RSPI system is comprised of this MCU (master) and three slaves (SPI slave 0 to SPI slave 2).

The RSPCKA and MOSIA outputs of this MCU (master) are connected to the RSPCK and MOSI inputs of SPI slave 0 to SPI slave 2. The MISO outputs of SPI slave 0 to SPI slave 3 are all connected to the MISOA input of this MCU (master). SSLA0, SSLA1, and SSLA3 outputs of this MCU (master) are connected to the SSL inputs of SPI slave 0 to SPI slave 2, respectively.

This MCU (master) drives RSPCKA, MOSIA, SSLA0, SSLA1, and SSLA3. Of the SPI slave 0 to SPI slave 2, the slave that receives low-level input into the SSL input drives MISO.

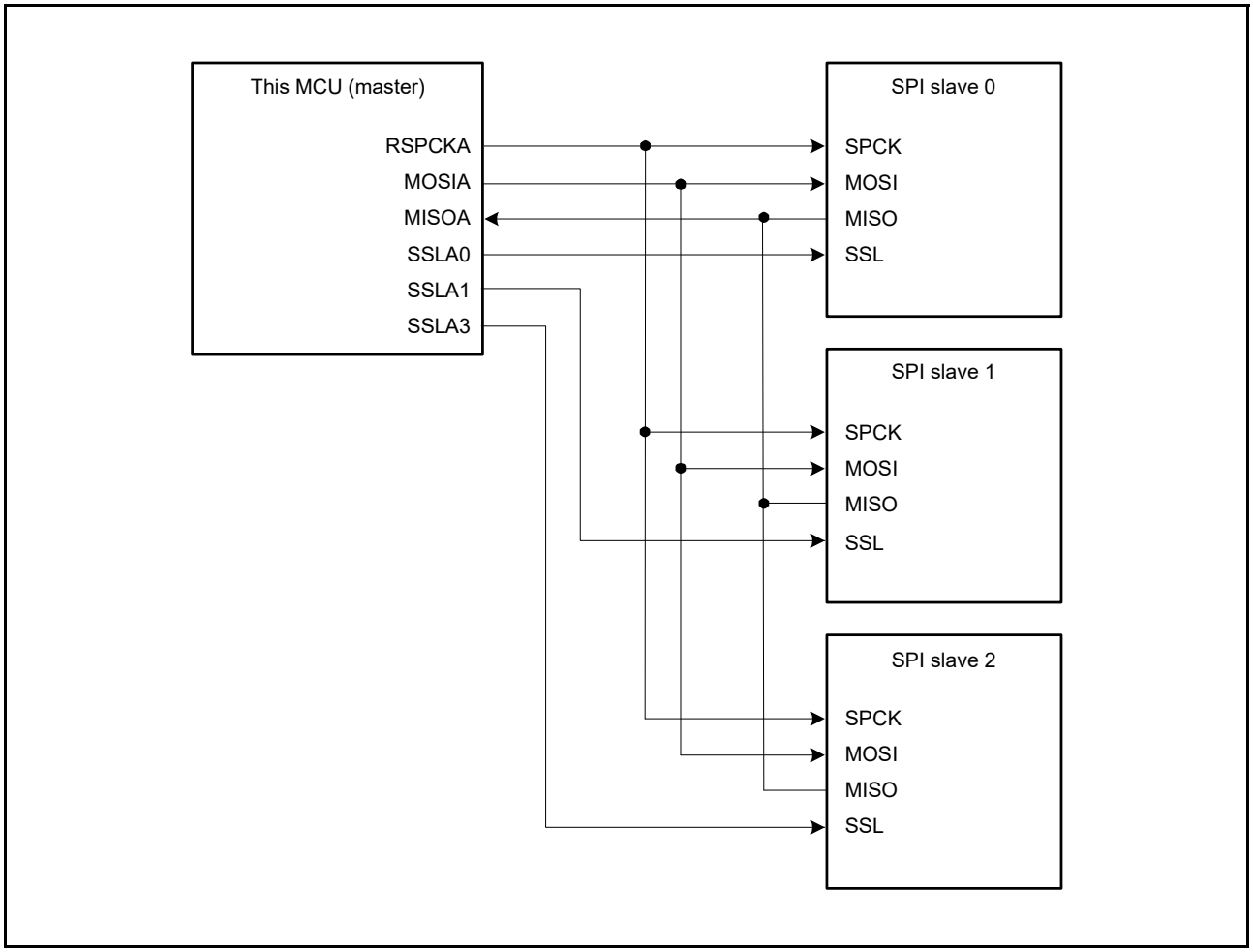


Figure 38.8 Single-Master/Multi-Slave Configuration Example (This MCU = Master)

Downloaded from Arrow.com.

38.3.3.4 Single Master/Multi-Slave (with This MCU Acting as Slave)

Figure 38.9 shows a single-master/multi-slave RSPI system configuration example when this MCU is used as a slave. In the example of Figure 38.9, the RSPI system is comprised of an SPI master and two MCUs (slave X and slave Y).
The SPCK and MOSI outputs of the SPI master are connected to the RSPCKA and MOSIA inputs of the MCUs (slave X and slave Y). The MISOA outputs of the MCUs (slave X and slave Y) are all connected to the MISO input of the SPI master. SSLX and SSLY outputs of the SPI master are connected to the SSLA0 inputs of the MCUs (slave X and slave Y), respectively.
The SPI master drives SPCK, MOSI, SSLX, and SSLY. Of the MCUs (slave X and slave Y), the slave that receives low-level input into the SSLA0 input drives MISOA.

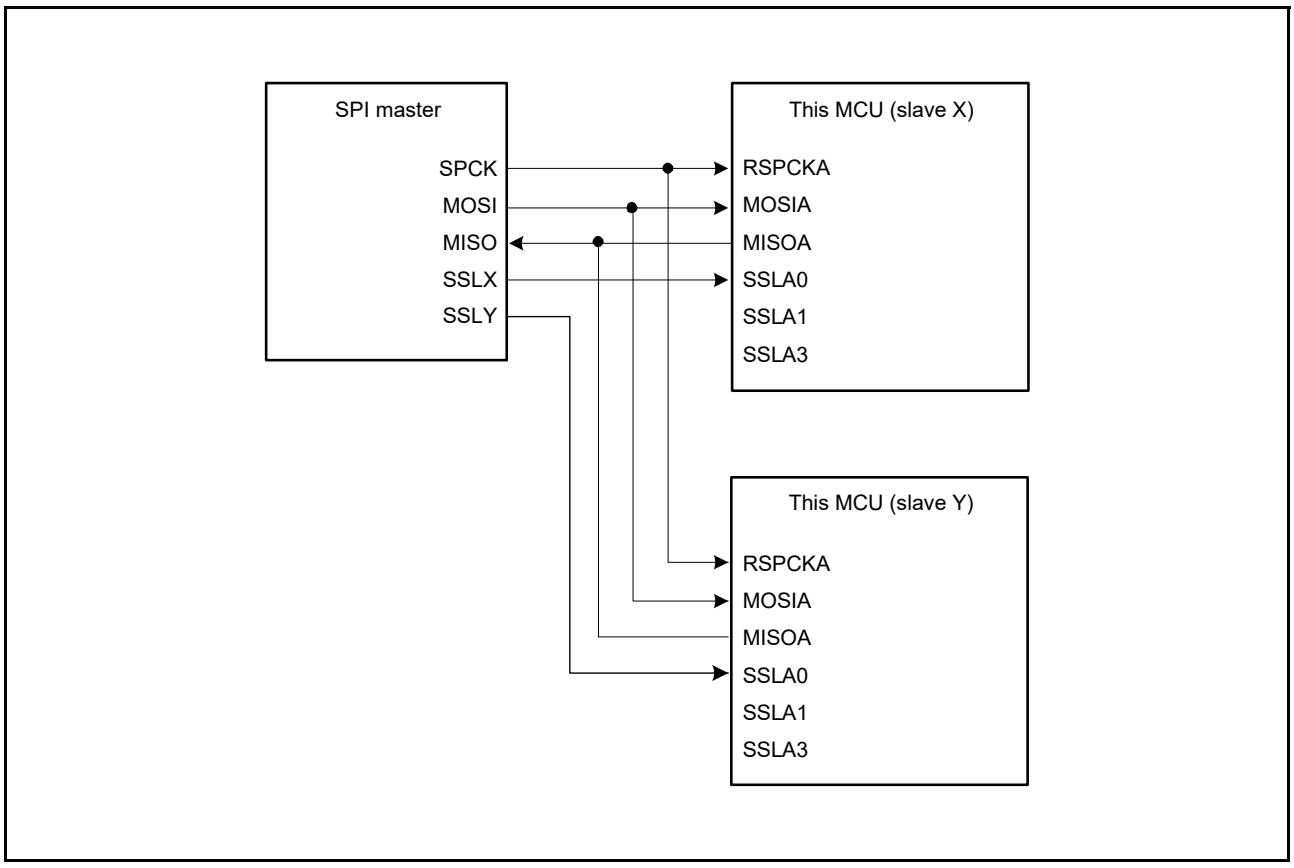


Figure 38.9 Single-Master/Multi-Slave Configuration Example (This MCU = Slave)

Downloaded from Arrow.com.

38.3.3.5 Multi-Master/Multi-Slave (with This MCU Acting as Master)

Figure 38.10 shows a multi-master/multi-slave RSPI system configuration example when this MCU is used as a master. In the example of Figure 38.10, the RSPI system is comprised of two MCUs (master X and master Y) and two SPI slaves (SPI slave 1 and SPI slave 2).

The RSPCKA and MOSIA outputs of the MCUs (master X and master Y) are connected to the RSPCK and MOSI inputs of SPI slaves 1 and 2. The MISO outputs of SPI slaves 1 and 2 are connected to the MISOA inputs of the MCUs (master X and master Y). Any generic port Y output from this MCU (master X) is connected to the SSLA0 input of this MCU (master Y). Any generic port X output of this MCU (master Y) is connected to the SSLA0 input of this MCU (master X). The SSLA1 and SSLA3 outputs of the MCUs (master X and master Y) are connected to the SSL inputs of the SPI slaves 1 and 2.

This MCU drives RSPCKA, MOSIA, SSLA1, and SSLA3 when the SSLA0 input level is high. When the SSLA0 input level is low, this MCU detects a mode fault error, sets RSPCKA, MOSIA, SSLA1, and SSLA3 to Hi-Z, and releases the RSPI bus right to the other master. Of the SPI slaves 1 and 2, the slave that receives low-level input into the SSL input drives MISO.

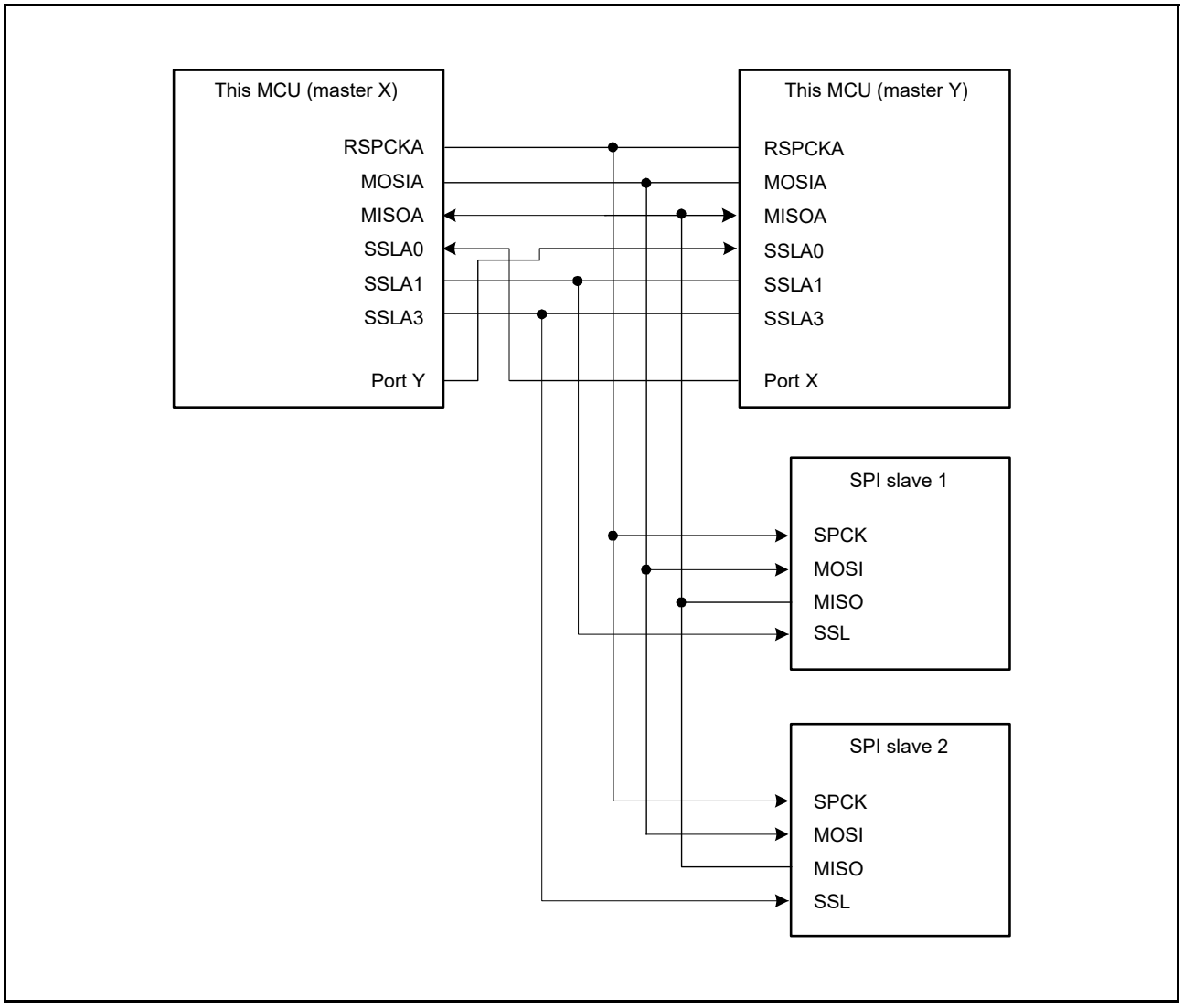


Figure 38.10 Multi-Master/Multi-Slave Configuration Example (This MCU = Master)

Downloaded from Arrow.com.

38.3.3.6 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) (with This MCU Acting as Master)

Figure 38.11 shows a master (clock synchronous operation)/slave (clock synchronous operation) RSPI system configuration example when this MCU is used as a master. In the master (clock synchronous operation)/slave (clock synchronous operation) configuration, SSLA0, SSLA1, and SSLA3 of this MCU (master) are not used.
This MCU (master) drives the RSPCKA and MOSIA. The SPI slave drives the MISO.

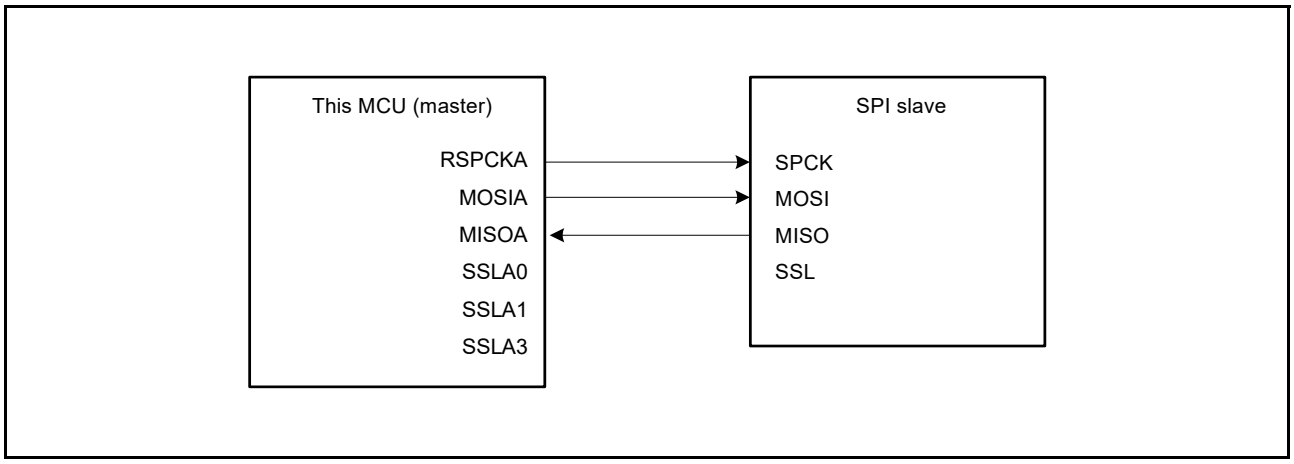


Figure 38.11 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) Configuration Example (This MCU = Master)

38.3.3.7 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) (with This MCU Acting as Slave)

Figure 38.12 shows a master (clock synchronous operation)/slave (clock synchronous operation) RSPI system configuration example when this MCU is used as a slave. When this MCU is to operate as a slave (clock synchronous operation), this MCU (slave) drives the MISOA and the SPI master drives the SPCK and MOSI. In addition, SSLA0, SSLA1, and SSLA3 of this MCU (slave) are not used.
Only in the single-slave configuration in which the SPCMDm.CPHA bit is set to 1, this MCU (slave) can execute serial transfer.

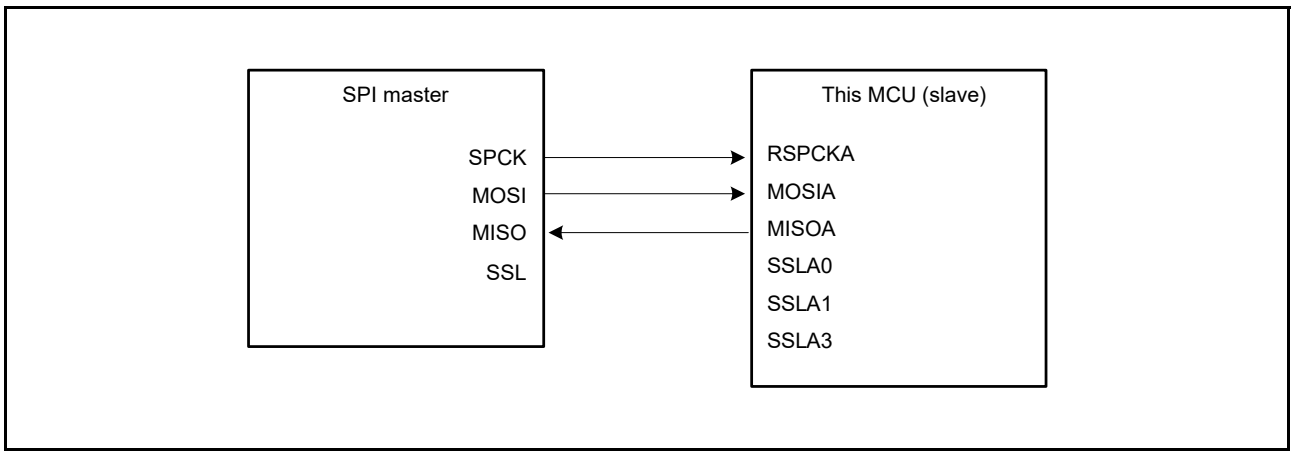


Figure 38.12 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) Configuration Example (This MCU = Slave, CPHA = 1)

Downloaded from Arrow.com.

38.3.4 Data Format

The RSPI's data format depends on the settings in RSPI command register m (SPCMDm) (m = 0 to 7) and the parity enable bit in RSPI control register 2 (SPCR2.SPPE). Regardless of whether the MSB or LSB is first, the RSPI treats the range from the LSB bit in the RSPI data register (SPDR) to the selected data length as transfer data.
The format of one frame of data before or after transfer is shown below.

(a) With Parity Disabled

When parity is disabled, transmission or reception of data proceeds with the length in bits selected in the RSPI data length setting bits in RSPI command register m (SPCMDm.SPB[3:0]).

(b) With Parity Enabled

When parity is enabled, transmission or reception of data proceeds with the length in bits selected in the RSPI data length setting bits in RSPI command register m (SPCMDm.SPB[3:0]). In this case, however, the last bit is a parity bit.

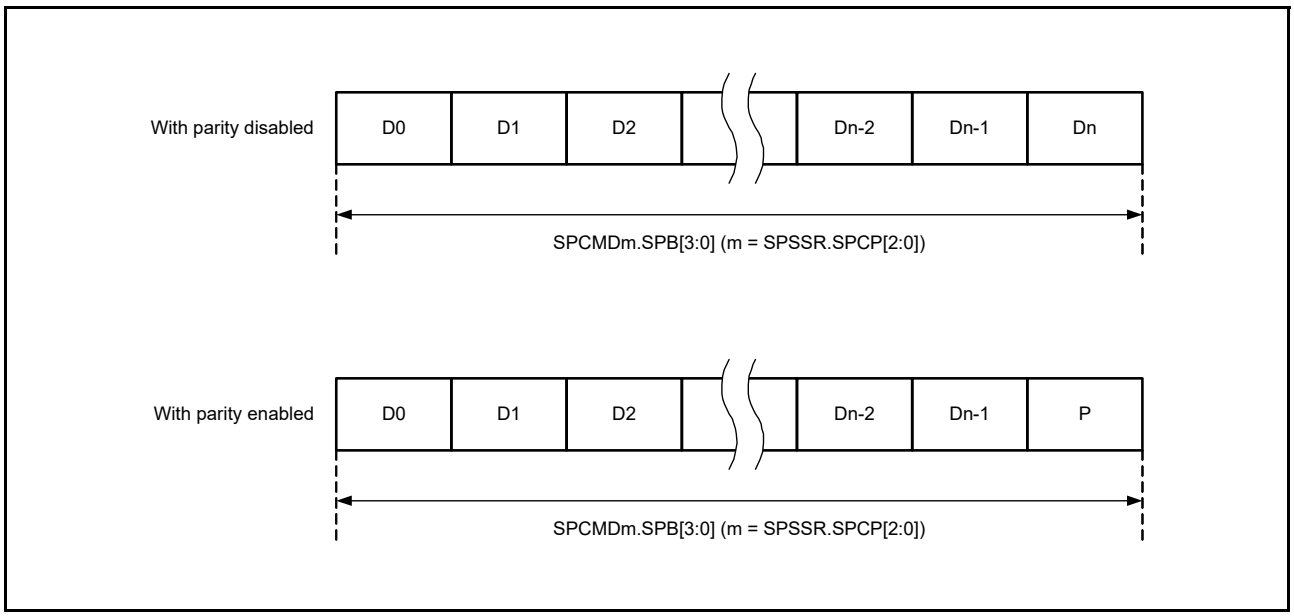


Figure 38.13 Outline of the Data Format (with Parity Disabled/Enabled)

Downloaded from Arrow.com.

38.3.4.1 When Parity is Disabled (SPCR2.SPPE = 0)

When parity is disabled, data for transmission are copied to the shift register with no prior processing. A description of the connection between the RSPI data register (SPDR) and the shift register in terms of the combination of MSB or LSB first and data length is given below.

(1) MSB First Transfer (32-Bit Data)

Figure 38.14 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity disabled, an RSPI data length of 32 bits, and MSB first selected.
In transmission, bits T31 to T00 from the current stage of the transmit buffer are copied to the shift register. Data for transmission are shifted out from the shift register in order from T31, through T30, and so on to T00.
In reception, received data are shifted in bit by bit through bit 0 of the shift register. When bits R31 to R00 have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer.

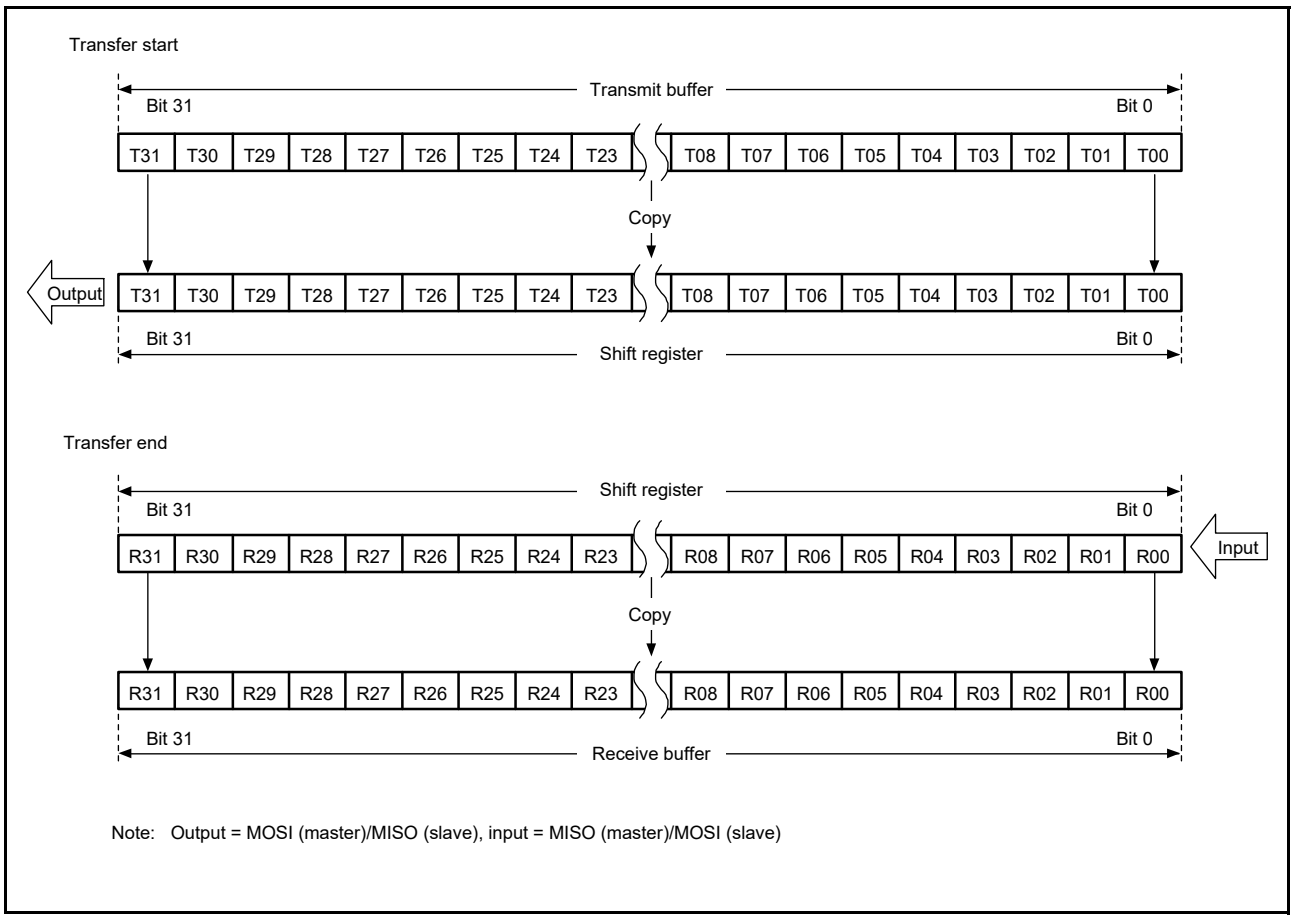


Figure 38.14 MSB First Transfer (32-Bit Data, Parity Disabled)

Downloaded from Arrow.com.

(2) MSB First Transfer (24-Bit Data)

Figure 38.15 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity disabled, 24 bits as the RSPI data length for an example that is not 32 bits, and MSB first selected.

In transmission, the lower-order 24 bits (T23 to T00) from the current stage of the transmit buffer are copied to the shift register. Data for transmission are shifted out from the shift register in order from T23, through T22, and so on to T00.

In reception, received data are shifted in bit by bit through bit 0 of the shift register. When bits R23 to R00 have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer. At this time, the higher-order 8 bits of the transmit buffer are stored in the higher-order 8 bits of the receive buffer.

Writing 0 to bits T31 to T24 at the time of transmission leads to 0 being inserted in the higher-order 8 bits of the receive buffer.

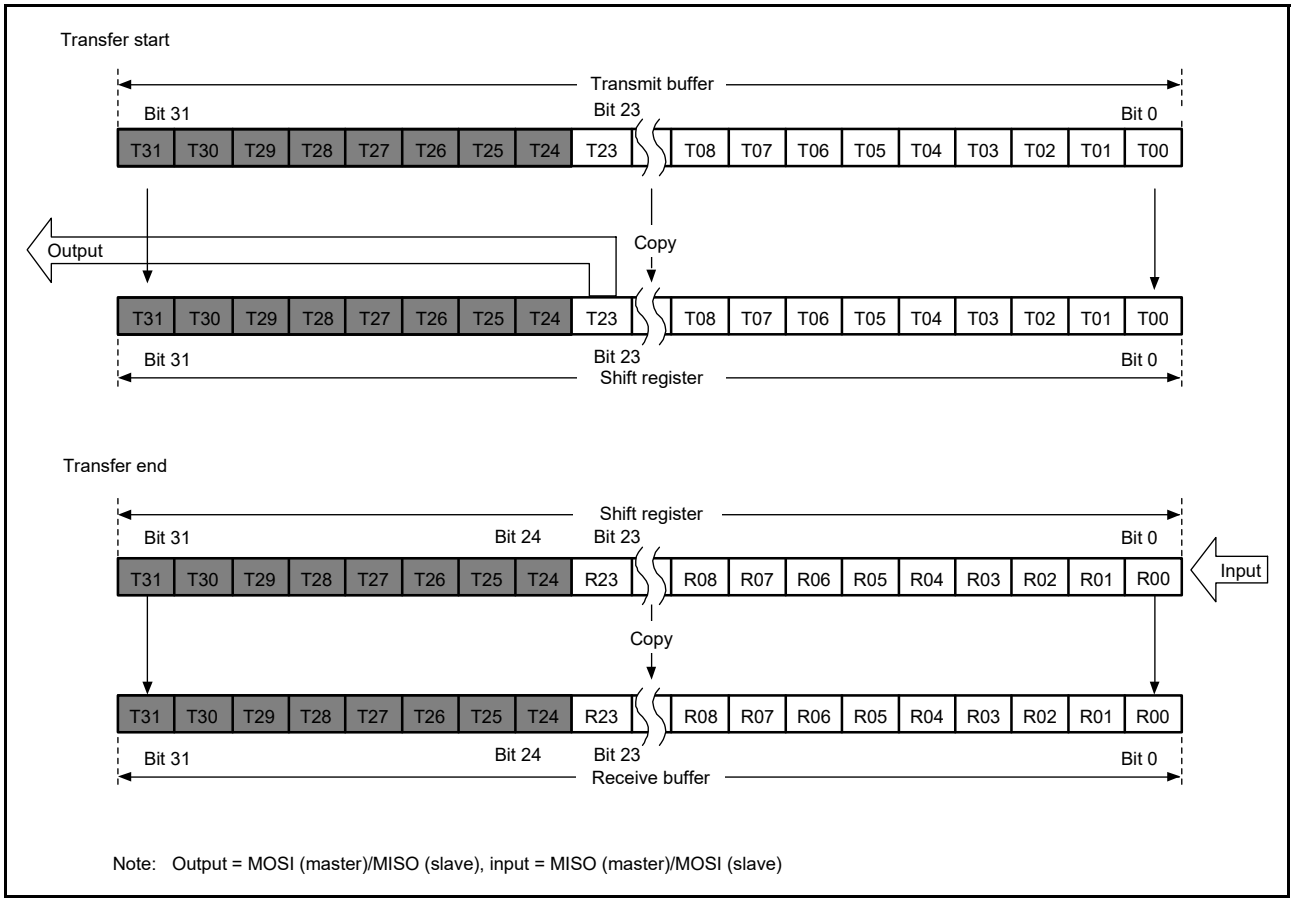


Figure 38.15 MSB First Transfer (24-Bit Data, Parity Disabled)

Downloaded from Arrow.com.

(3) LSB First Transfer (32-Bit Data)

Figure 38.16 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity disabled, an RSPI data length of 32 bits, and LSB first selected.

In transmission, bits T31 to T00 from the current stage of the transmit buffer are reordered bit by bit to obtain the order T00 to T31 for copying to the shift register. Data for transmission are shifted out from the shift register in order from T00, through T01, and so on to T31.

In reception, received data are shifted in bit by bit through bit 0 of the shift register. When bits R00 to R31 have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer.

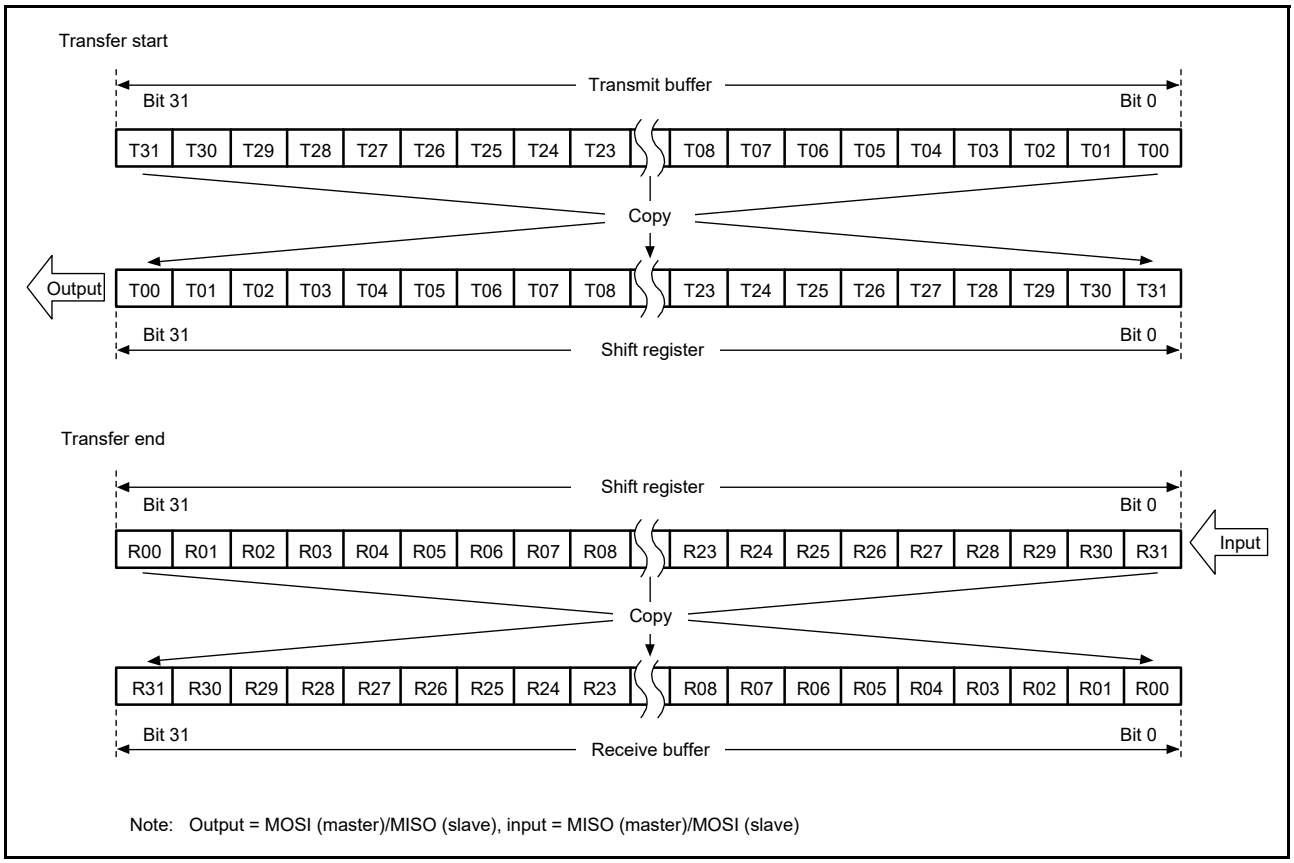


Figure 38.16 LSB First Transfer (32-Bit Data, Parity Disabled)

Downloaded from Arrow.com.

(4) LSB First Transfer (24-Bit Data)

Figure 38.17 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity disabled, 24 bits as the RSPI data length for an example that is not 32 bits, and LSB first selected.

In transmission, the lower-order 24 bits (T23 to T00) from the current stage of the transmit buffer are reordered bit by bit to obtain the order T00 to T23 for copying to the shift register. Data for transmission are shifted out from the shift register in order from T00, through T01, and so on to T23.

In reception, received data are shifted in bit by bit through bit 8 of the shift register. When bits R00 to R23 have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer.

At this time, the higher-order 8 bits of the transmit buffer are stored in the higher-order 8 bits of the receive buffer.

Writing 0 to bits T31 to T24 at the time of transmission leads to 0 being inserted in the higher-order 8 bits of the receive buffer.

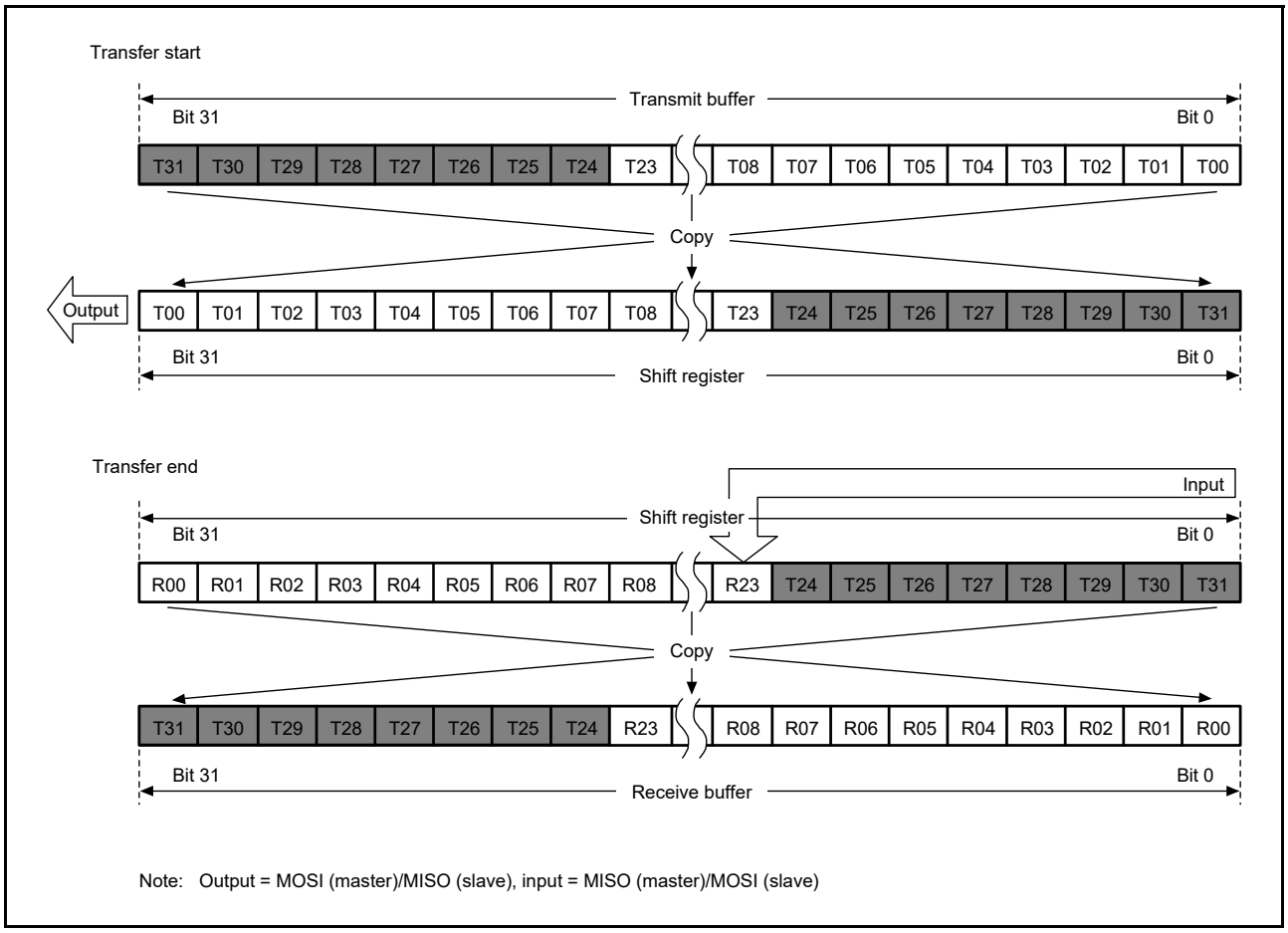


Figure 38.17 LSB First Transfer (24-Bit Data, Parity Disabled)

Downloaded from Arrow.com.

38.3.4.2 When Parity is Enabled (SPCR2.SPPE = 1)

When parity is enabled, the lowest-order bit of the data for transmission becomes a parity bit. Hardware calculates the value of the parity bit.

(1) MSB First Transfer (32-Bit Data)

Figure 38.18 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity enabled, an RSPI data length of 32 bits, and MSB first selected.
In transmission, the value of the parity bit (P) is calculated from bits T31 to T01. This replaces the final bit, T00, and the whole is copied to the shift register. Data are transmitted in the order T31, T30, …, T01, and P.
In reception, received data are shifted in bit by bit through bit 0 of the shift register. When bits R31 to P have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer. On copying of data to the shift register, the data from R31 to P are checked by judging the parity.

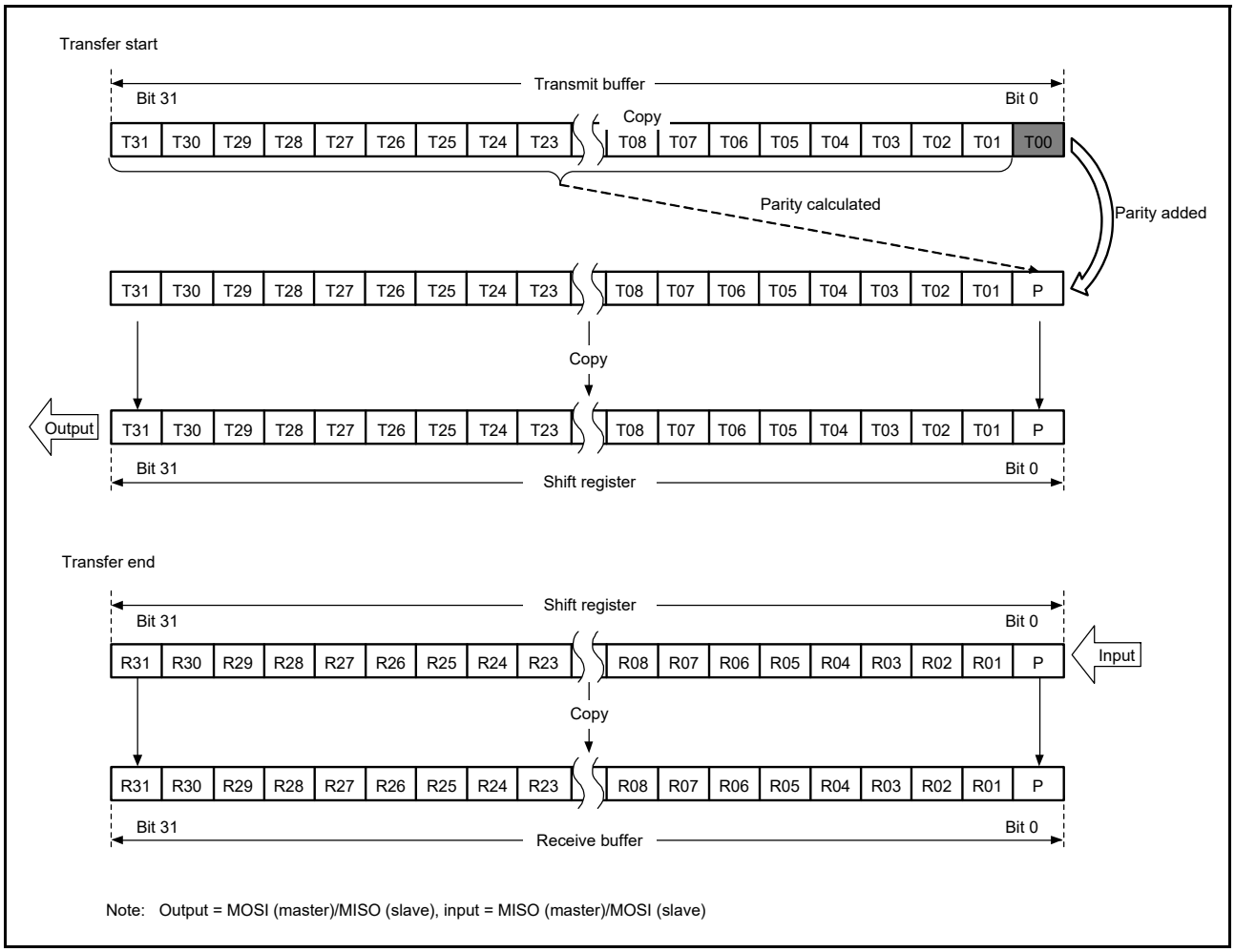


Figure 38.18 MSB First Transfer (32-Bit Data, Parity Enabled)

Downloaded from Arrow.com.

(2) MSB First Transfer (24-Bit Data)

Figure 38.19 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity enabled, 24 bits as the RSPI data length for an example that is not 32 bits, and MSB first selected.
In transmission, the value of the parity bit (P) is calculated from bits T23 to T01. This replaces the final bit, T00, and the whole is copied to the shift register. Data are transmitted in the order T23, T22, …, T01, and P.
In reception, received data are shifted in bit by bit through bit 0 of the shift register. When bits R23 to P have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer. On copying of data to the shift register, the data from R23 to P are checked by judging the parity. At this time, the higher-order 8 bits of the transmit buffer are stored in the higher-order 8 bits of the receive buffer. Writing 0 to bits T31 to T24 at the time of transmission leads to 0 being inserted in the higher-order 8 bits of the receive buffer.

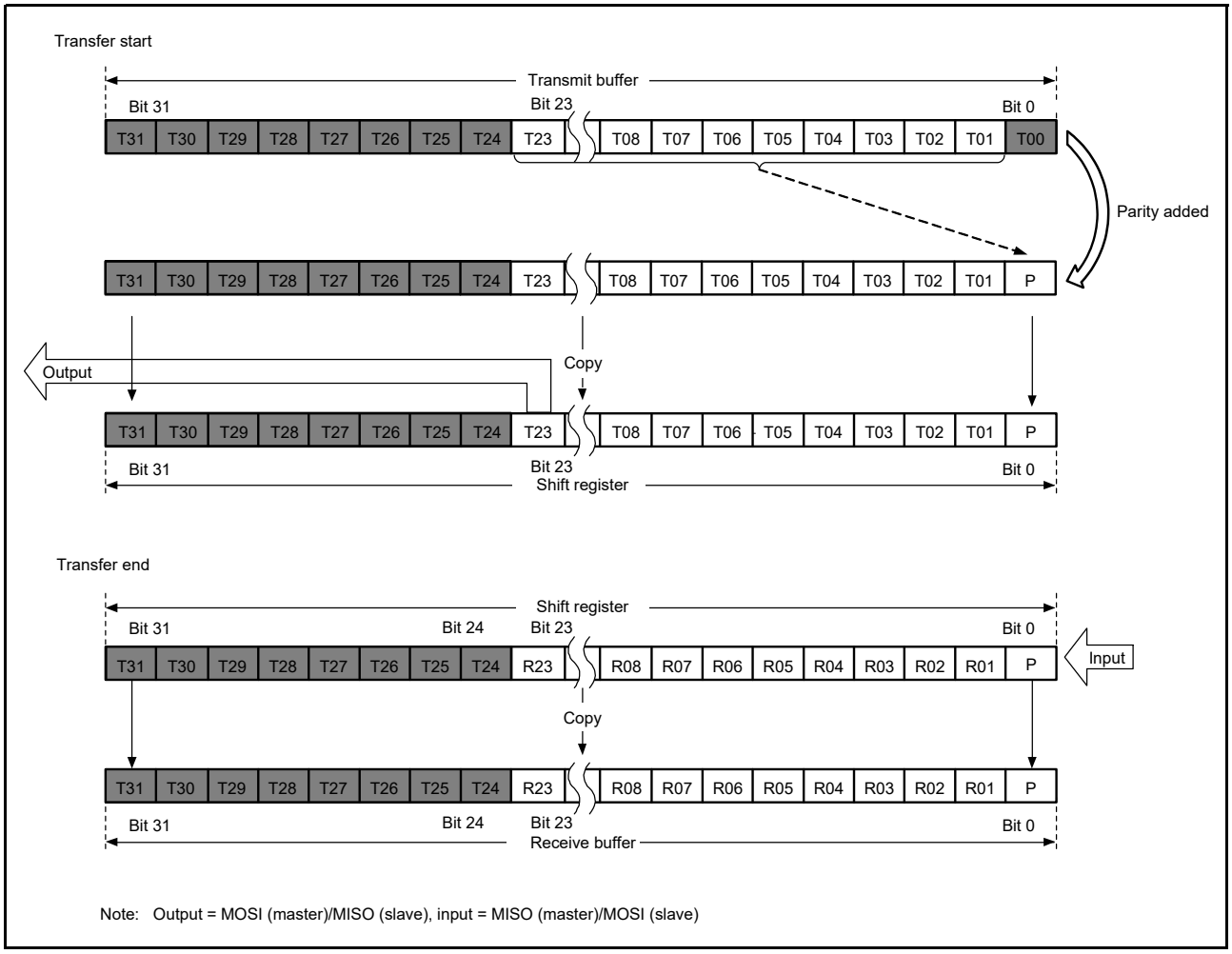


Figure 38.19 MSB First Transfer (24-Bit Data, Parity Enabled)

Downloaded from Arrow.com.

(3) LSB First Transfer (32-Bit Data)

Figure 38.20 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity enabled, an RSPI data length of 32 bits, and LSB first selected.

In transmission, the value of the parity bit (P) is calculated from bits T30 to T00. This replaces the final bit, T31, and the whole is copied to the shift register. Data are transmitted in the order T00, T01, …, T30, and P.

In reception, received data are shifted in bit by bit through bit 0 of the shift register. When bits R00 to P have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer. On copying of data to the shift register, the data from R00 to P are checked by judging the parity.

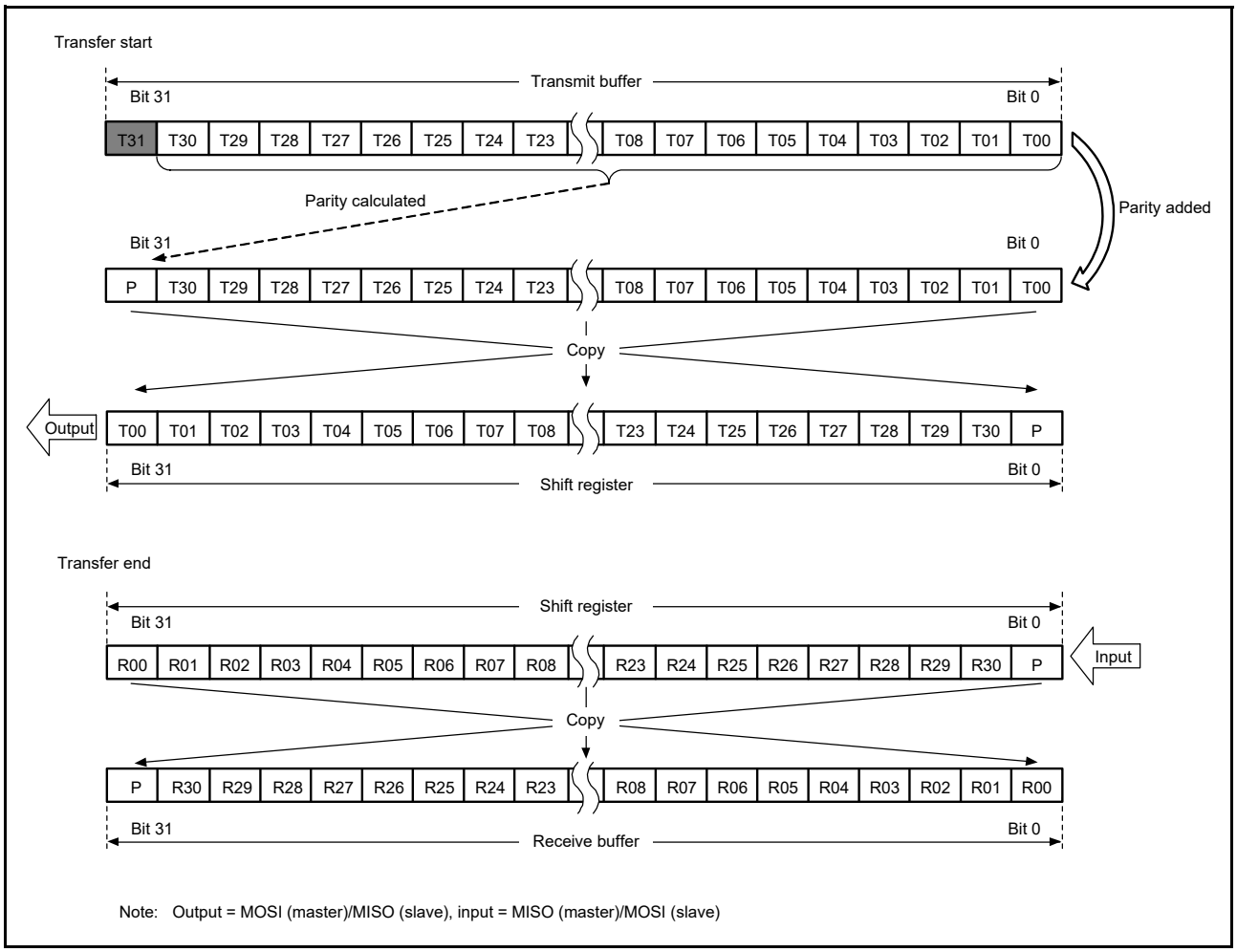


Figure 38.20 LSB First Transfer (32-Bit Data, Parity Enabled)

Downloaded from Arrow.com.

(4) LSB First Transfer (24-Bit Data)

Figure 38.21 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity enabled, 24 bits as the RSPI data length for an example that is not 32 bits, and LSB first selected.
In transmission, the value of the parity bit (P) is calculated from bits T22 to T00. This replaces the final bit, T23, and the whole is copied to the shift register. Data are transmitted in the order T00, T01, …, T22, and P.
In reception, received data are shifted in bit by bit through bit 8 of the shift register. When bits R00 to P have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer. On copying of data to the shift register, the data from R00 to P are checked by judging the parity. At this time, the higher-order 8 bits of the transmit buffer are stored in the higher-order 8 bits of the receive buffer. Writing 0 to bits T31 to T24 at the time of transmission leads to 0 being inserted in the higher-order 8 bits of the receive buffer.

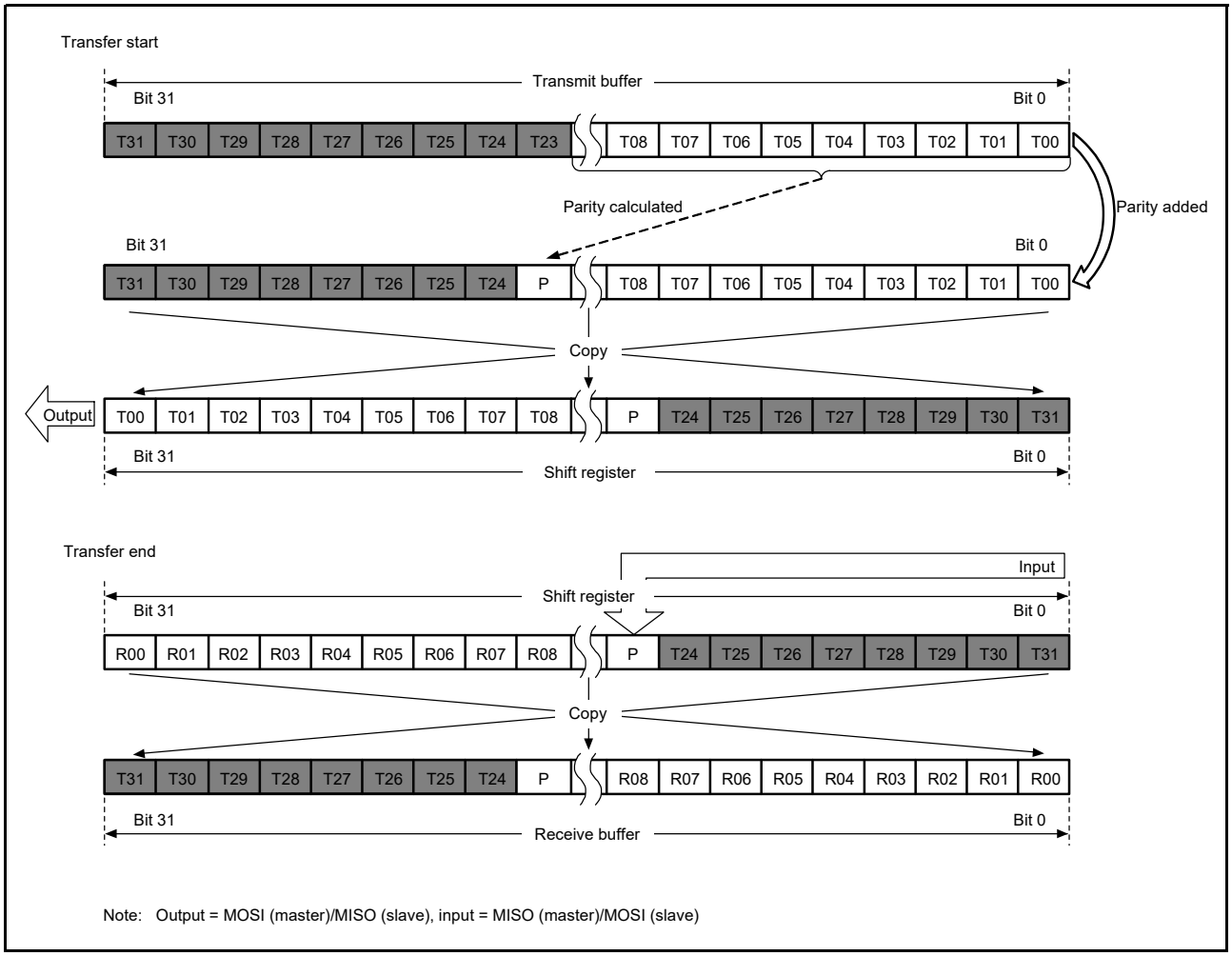


Figure 38.21 LSB First Transfer (24-Bit Data, Parity Enabled)

Downloaded from Arrow.com.

38.3.5 Transfer Format

38.3.5.1 CPHA = 0

Figure 38.22 shows a sample transfer format for the serial transfer of 8-bit data when the SPCMDm.CPHA bit is 0. Note that clock synchronous operation (the SPCR.SPMS bit is 1) should not performed when the RSPI operates in slave mode (SPCR.MSTR = 0) and the CPHA bit is 0. In Figure 38.22, RSPCKA (CPOL = 0) indicates the RSPCKA signal waveform when the SPCMDm.CPOL bit is 0; RSPCKA (CPOL = 1) indicates the RSPCKA signal waveform when the CPOL bit is 1. The sampling timing represents the timing at which the RSPI fetches serial transfer data into the shift register. The I/O directions of the signals depend on the RSPI settings. For details, refer to section 38.3.2, Controlling RSPI Pins.

When the SPCMDm.CPHA bit is 0, the driving of valid data to the MOSIA and MISOA signals commences at an SSLAi signal assertion timing. The first RSPCKA signal change timing that occurs after the SSLAi signal assertion becomes the first transfer data fetch timing. After this timing, data is sampled at every 1 RSPCK cycle. The change timing for MOSIA and MISOA signals is 1/2 RSPCK cycles after the transfer data fetch timing. The CPOL bit setting does not affect the RSPCKA signal operation timing; it only affects the signal polarity.

t1 denotes a period from an SSLAi signal assertion to RSPCKA oscillation (RSPCK delay). t2 denotes a period from the termination of RSPCKA oscillation to an SSLAi signal negation (SSL negation delay). t3 denotes a period in which SSLAi signal assertion is suppressed for the next transfer after the end of serial transfer (next-access delay). t1, t2, and t3 are controlled by a master device running on the RSPI system. For a description of t1, t2, and t3 when the RSPI of this MCU is in master mode, refer to section 38.3.10.1, Master Mode Operation.

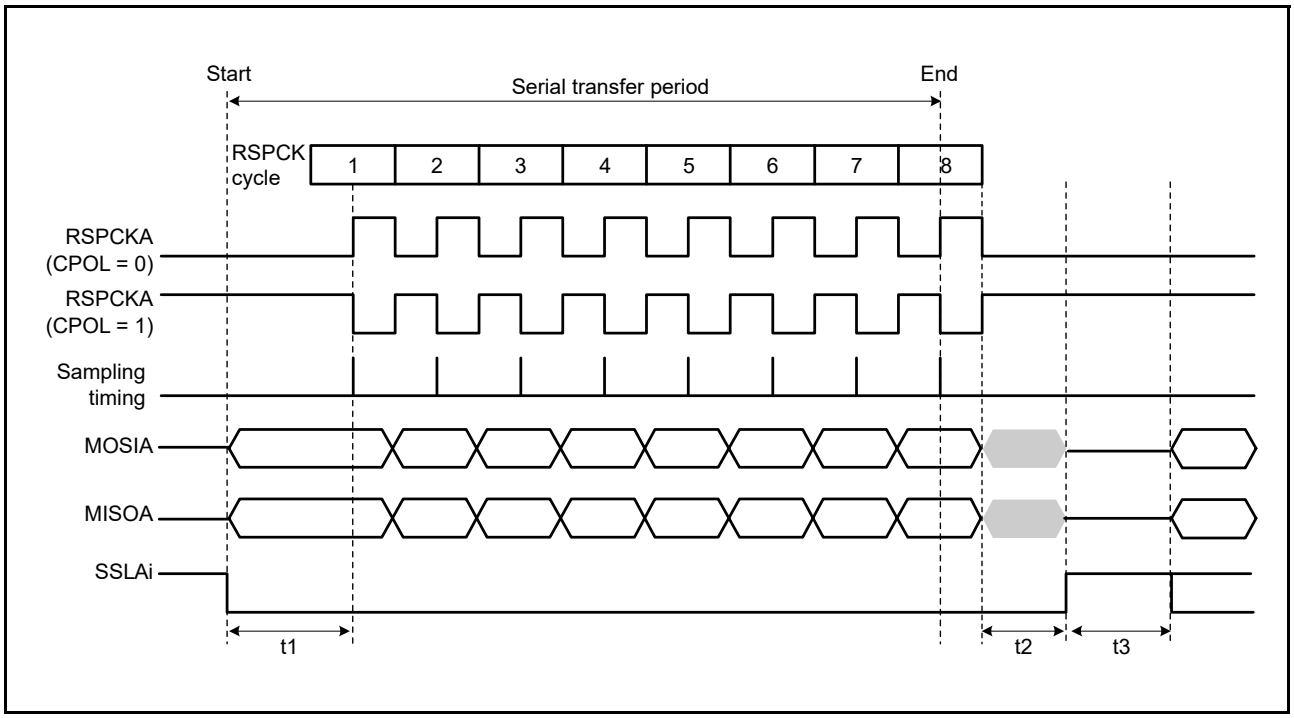


Figure 38.22 RSPI Transfer Format (CPHA = 0)

Downloaded from Arrow.com.

38.3.5.2 CPHA = 1

Figure 38.23 shows a sample transfer format for the serial transfer of 8-bit data when the SPCMDm.CPHA bit is 1. However, when the SPCR.SPMS bit is 1, the SSLAi signals are not used, and only the three signals RSPCKA, MOSIA, and MISOA handle communications. In Figure 38.23, RSPCK (CPOL = 0) indicates the RSPCKA signal waveform when the SPCMDm.CPOL bit is 0; RSPCKA (CPOL = 1) indicates the RSPCKA signal waveform when the CPOL bit is 1. The sampling timing represents the timing at which the RSPI fetches serial transfer data into the shift register. The I/O directions of the signals depend on the RSPI mode (master or slave). For details, refer to section 38.3.2, Controlling RSPI Pins.

When the SPCMDm.CPHA bit is 1, the driving of invalid data to the MISOA signal commences at an SSLAi signal assertion timing. The output of valid data to the MOSIA and MISOA signals commences at the first RSPCKA signal change timing that occurs after the SSLAi signal assertion. After this timing, data is updated at every 1 RSPCK cycle. The transfer data fetch timing is 1/2 RSPCK cycles after the data update timing. The SPCMDm.CPOL bit setting does not affect the RSPCKA signal operation timing; it only affects the signal polarity.

t1, t2, and t3 are the same as those in the case of CPHA = 0. For a description of t1, t2, and t3 when the RSPI of this MCU is in master mode, refer to section 38.3.10.1, Master Mode Operation.

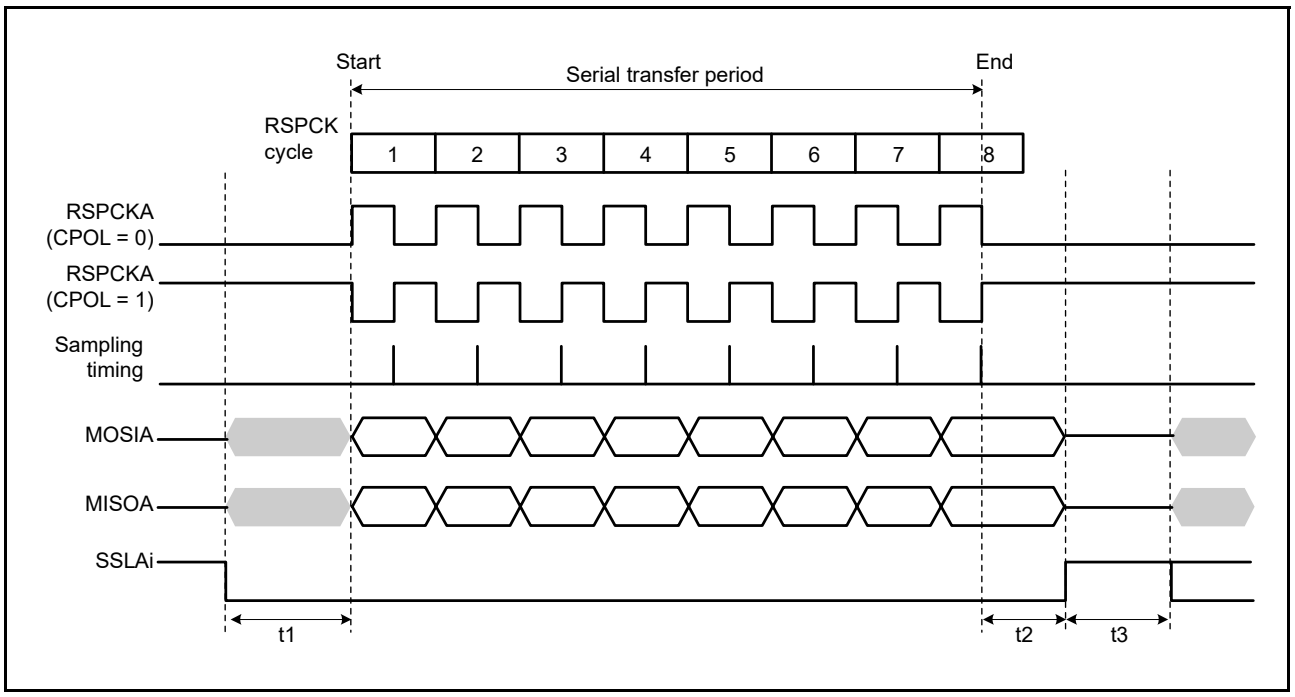


Figure 38.23 RSPI Transfer Format (CPHA = 1)

Downloaded from Arrow.com.

38.3.6 Communications Operating Mode

Full-duplex synchronous serial communications or transmit operations only can be selected by the communications operating mode select bit (SPCR.TXMD). The SPDR access shown in Figure 38.24 and Figure 38.25 indicate the condition of access to the SPDR register, where W denotes a write cycle.

38.3.6.1 Full-Duplex Synchronous Serial Communications (SPCR.TXMD = 0)

Figure 38.24 shows an example of operation when the communications operating mode select bit (SPCR.TXMD) is set to 0. In the example in Figure 38.24, the RSPI performs an 8-bit serial transfer in which the SPDCR.SPFC[1:0] bits are 00b, the SPCMDm.CPHA bit is 1, and the SPCMDm.CPOL bit is 0. The numbers given under the RSPCKA waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

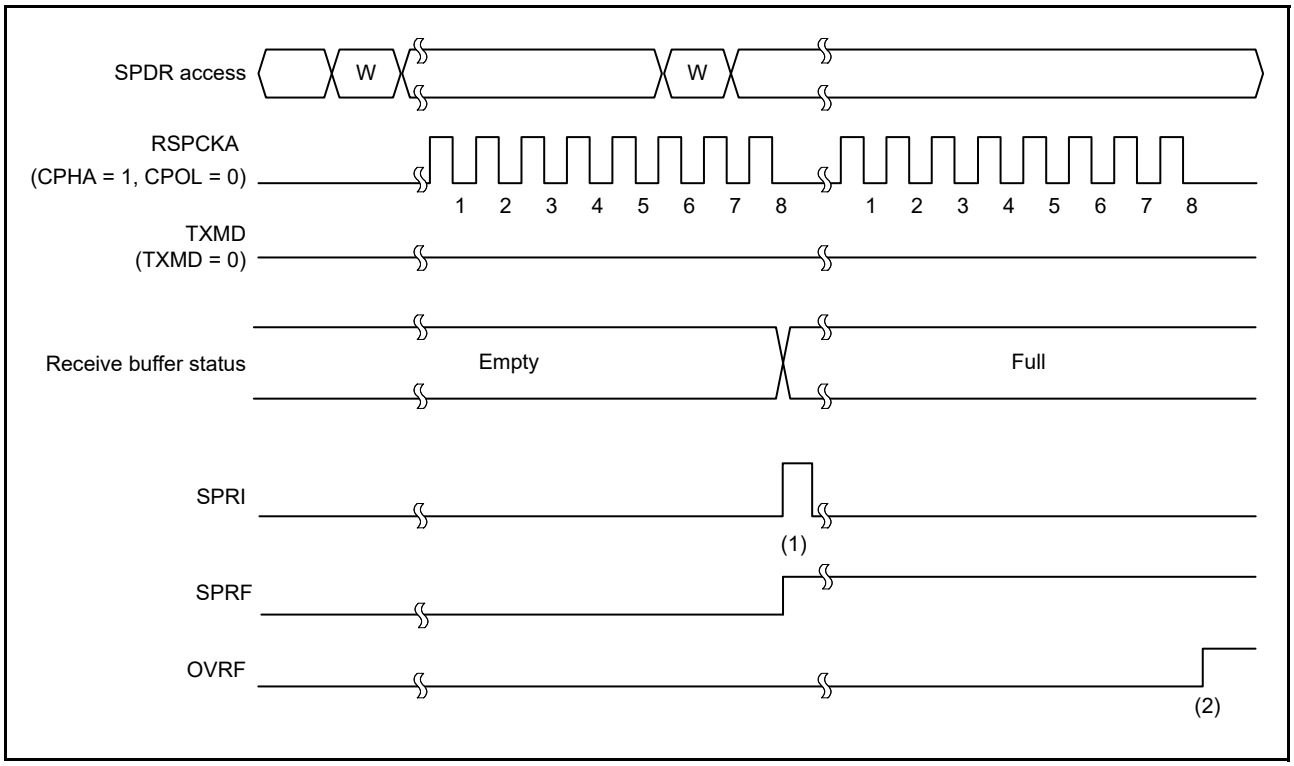


Figure 38.24 Operation Example of SPCR.TXMD = 0

The operation of the flags at timings shown in steps (1) and (2) in the figure is described below.

(1) When a serial transfer ends with the receive buffer of SPDR empty, the RSPI generates a receive buffer full interrupt request (SPRI) (sets the SPSR.SPRF flag to 1) and copies the received data in the shift register to the receive buffer.

(2) When a serial transfer ends with the receive buffer of SPDR holding data that was received in the previous serial transfer, the RSPI sets the SPSR.OVRF flag to 1 and discards the received data in the shift register.

When full-duplex synchronous serial communications (SPCR.TXMD = 0) is selected, reception occurs simultaneously with transmit operations. As such, the SPRF and OVRF flags in the SPSR register become 1 at the timing described in (1) and (2), respectively, according to the state of the receive buffer.

Downloaded from Arrow.com.

38.3.6.2 Transmit Operations Only (SPCR.TXMD = 1)

Figure 38.25 shows an example of operation when the communications operating mode select bit (SPCR.TXMD) is set to 1. In the example in Figure 38.25, the RSPI performs an 8-bit serial transfer in which the SPDCR.SPFC[1:0] bits are 00b, the SPCMDm.CPHA bit is 1, and the SPCMDm.CPOL bit is 0. The numbers given under the RSPCKA waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

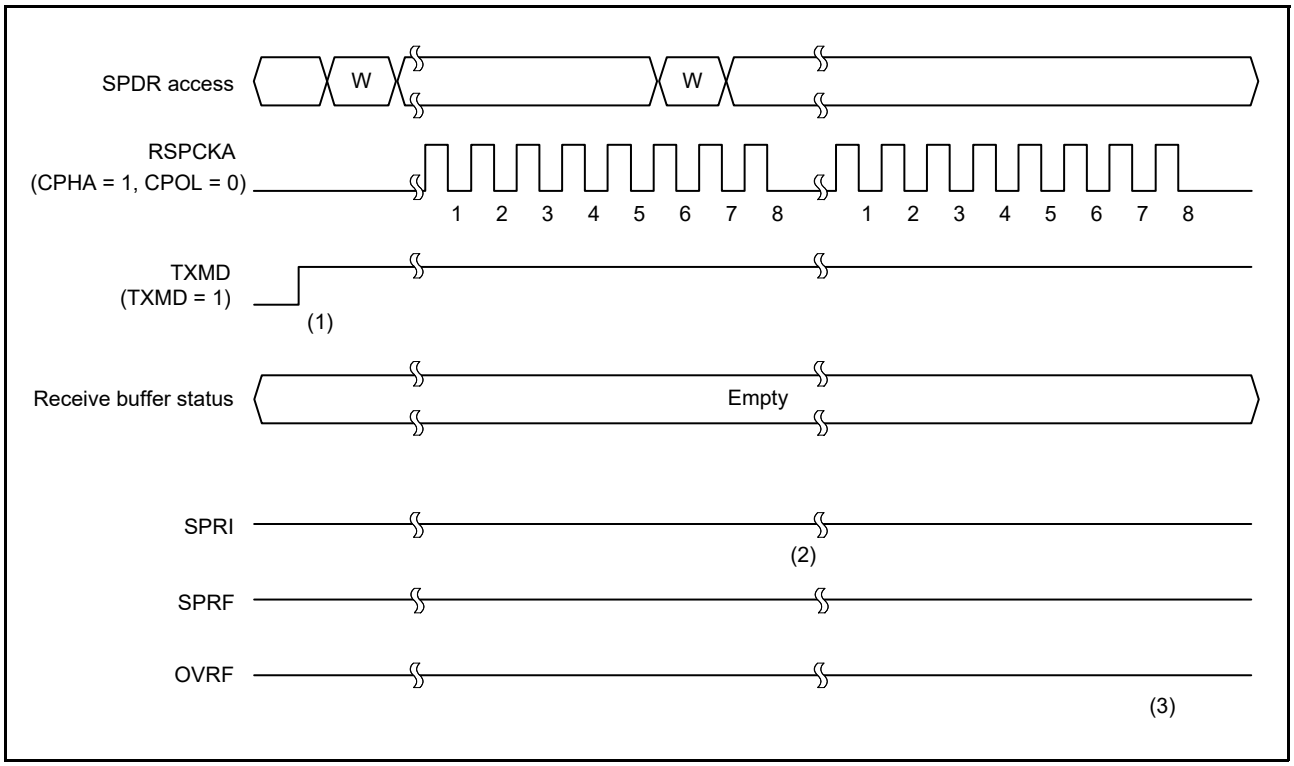


Figure 38.25 Operation Example of SPCR.TXMD = 1

The operation of the flags at timings shown in steps (1) to (3) in the figure is described below.

(1) Make sure there is no data left in the receive buffer and the SPSR.SPRF, OVRF flags are 0 before entering the mode of transmit operations only (SPCR.TXMD = 1).

(2) When a serial transfer ends with the receive buffer of SPDR empty, if the mode of transmit operations only is selected (SPCR.TXMD = 1), the SPRF flag remains 0 and the RSPI does not copy the data from the shift register to the receive buffer.

(3) Since the receive buffer of SPDR does not hold data that was received in the previous serial transfer, even when a serial transfer ends, the SPSR.OVRF flag retains the value of 0, and the data in the shift register is not copied to the receive buffer.

When performing transmit operations only (SPCR.TXMD = 1), the RSPI transmits data but does not receive data. Therefore, the SPSR.SPRF, OVRF flags remain 0 at the timings of (1) to (3).

Downloaded from Arrow.com.

38.3.7 Transmit Buffer Empty/Receive Buffer Full Interrupts

Figure 38.26 shows an example of operation of the transmit buffer empty interrupt (SPTI) and the receive buffer full interrupt (SPRI). The SPDR register access shown in Figure 38.26 indicates the condition of access to the SPDR register, where W denotes a write cycle, and R a read cycle. In the example in Figure 38.26, the RSPI performs an 8-bit serial transfer in which the SPCR.TXMD bit is 0, the SPDCR.SPFC[1:0] bits are 00b, the SPCMDm.CPHA bit is 1, and the SPCMDm.CPOL bit is 0. The numbers given under the RSPCKA waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

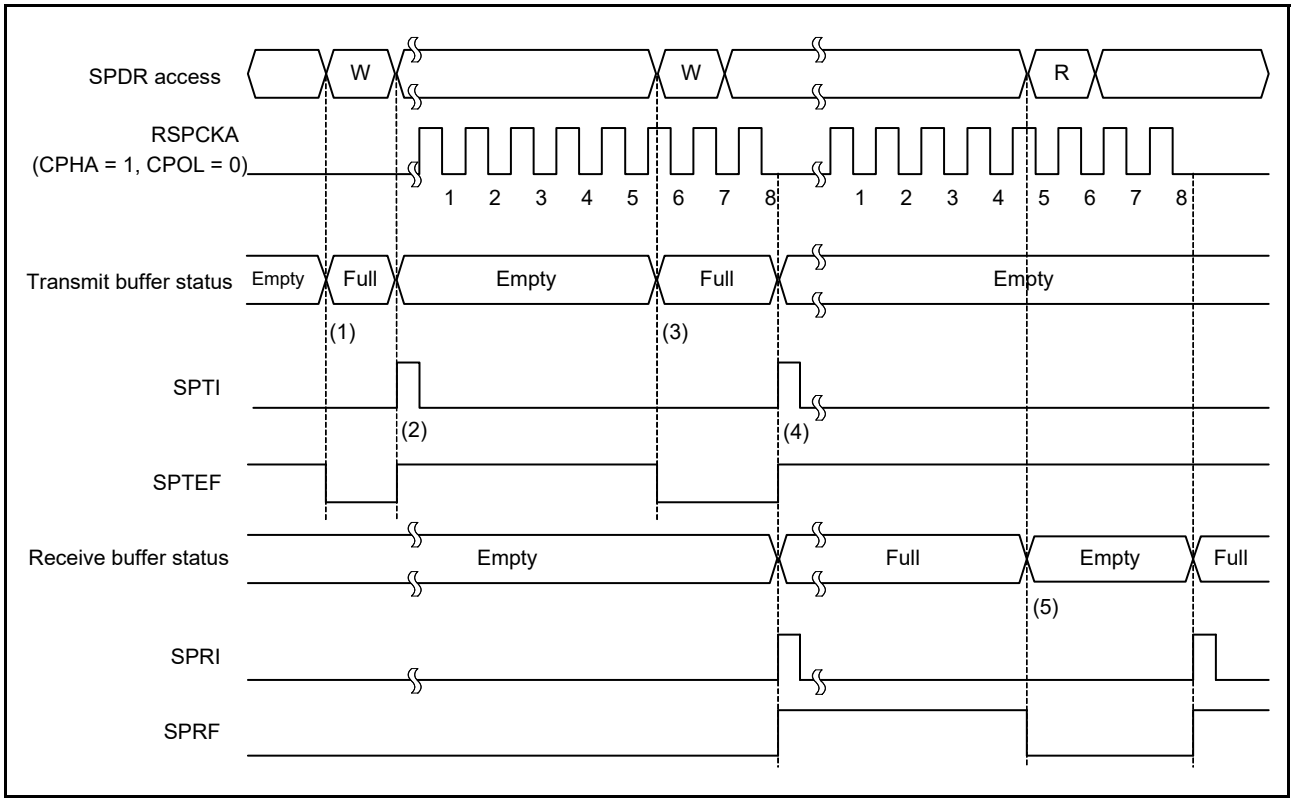


Figure 38.26 Operation Example of SPTI and SPRI Interrupts

The operation of the interrupts at timings shown in steps (1) to (5) in the figure is described below.

(1) When transmit data is written to SPDR when the transmit buffer of SPDR is empty (data for the next transfer is not set), the RSPI writes data to the transmit buffer and sets the SPSR.SPTEF flag to 0.

(2) If the shift register is empty, the RSPI copies the data from the transmit buffer to the shift register and generates a transmit buffer empty interrupt request (SPTI) and sets the SPSR.SPTEF flag to 1. How a serial transfer is started depends on the mode of the RSPI. For details, refer to section 38.3.10, SPI Operation, and section 38.3.11, Clock Synchronous Operation.

(3) When transmit data is written to SPDR in the transmit buffer empty interrupt routine or in the transmit buffer empty detecting process by polling the SPTEF flag, the data is transferred to the transmit buffer and the SPSR.SPTEF flag becomes 0. Because the data being transmitted is stored in the shift register, the RSPI does not copy the data from the transmit buffer to the shift register.

(4) When the serial transfer ends with the receive buffer of SPDR being empty, the RSPI copies the receive data from the shift register to the receive buffer, generates a receive buffer full interrupt request (SPRI), and sets the SPSR.SPRF flag to 1. Since the shift register becomes empty upon completion of serial transfer, when the transmit buffer had been full before the serial transfer ended, the RSPI sets the SPSR.SPTEF flag to 1 and copies the data from the transmit buffer to the shift register. Even when received data is not copied from the shift register to the receive buffer in an overrun error status, upon completion of the serial transfer, the RSPI determines that the shift register is empty, thus data transfer from the transmit buffer to the shift register is enabled.

Downloaded from Arrow.com.

(5) When SPDR is read in the receive buffer full interrupt routine or in the receive buffer full detecting process by polling the SPRF flag, the receive data can be read. When the receive data is read, the SPRF flag becomes 0.

If transmit data is written to SPDR while the transmit buffer holds data that has not yet been transmitted (the SPTEF flag is 0), the RSPI does not update the data in the transmit buffer. Transmit data should be written to SPDR in the transmit buffer empty interrupt request routine or in the transmit buffer empty detecting process by polling the SPTEF flag. To use a transmit buffer empty interrupt, set the SPTIE bit in SPCR to 1.
When setting the SPCR.SPE bit to 0 (RSPI disabled), the SPCR.SPTIE bit should also be set to 0. Otherwise (if the SPCR.SPE bit is 0 and the SPCR.SPTIE is 1), a transmit buffer empty interrupt request will occur.
When serial transfer ends with the receive buffer being full (the SPRF flag is 1), the RSPI does not copy data from the shift register to the receive buffer, and detects an overrun error (refer to section 38.3.8, Error Detection). To prevent a receive data overrun error, read the received data using a receive buffer full interrupt request before the next serial transfer ends. To use an RSPI receive buffer full interrupt, set the SPCR.SPRIE bit to 1.
Transmit and receive interrupts or the corresponding IRn.IR flags (where n is the interrupt vector number) in the ICU can be used to confirm the states of the transmit and receive buffers. Refer to section 15, Interrupt Controller (ICUb), for the interrupt vector numbers. The status of the transmit and receive buffers can be also confirmed by the SPTEF and SPRF flags.

Downloaded from Arrow.com.

38.3.8 Error Detection

In the normal RSPI serial transfer, the data written to the transmit buffer of SPDR is transmitted, and the received data can be read from the receive buffer of SPDR. If access is made to SPDR, depending on the status of the transmit/receive buffer or the status of the RSPI at the beginning or end of serial transfer, in some cases non-normal transfers can be executed.

If a non-normal transfer operation occurs, the RSPI detects the event as an overrun error, parity error, or mode fault error. Table 38.8 lists the relationship between non-normal transfer operations and the RSPI's error detection function.

Table 38.8 Relationship between Non-Normal Transfer Operations and RSPI Error Detection Function

	Occurrence Condition	RSPI Operation	Error Detection
1	SPDR is written when the transmit buffer is full.	• The contents of the transmit buffer are kept. • Missing write data.	None
2	SPDR is read when the receive buffer is empty.	Data received previously is output to the bus.	None
3	Serial transfer is started in slave mode when transmit data is still not loaded on the shift register.	Data received in previous serial transfer is transmitted.	None
4	Serial transfer terminates when the receive buffer is full.	• The contents of the receive buffer are kept. • Missing receive data.	Overrun error
5	An incorrect parity bit is received when performing full-duplex synchronous serial communications with the parity function enabled.	The parity error flag is asserted.	Parity error
6	The SSLA0 input signal is asserted when the serial transfer is idle in multi-master mode.	• Driving of the RSPCKA, MOSIA, SSLA1, SSLA3 output signals is stopped. • RSPI function is disabled.	Mode fault error
7	The SSLA0 input signal is asserted during serial transfer in multi-master mode.	• Serial transfer is suspended. • Missing transmit/receive data. • Driving of the RSPCKA, MOSIA, SSLA1, SSLA3 output signals is stopped. • RSPI function is disabled.	Mode fault error
8	The SSLA0 input signal is negated during serial transfer in slave mode.	• Serial transfer is suspended. • Missing transmit/receive data. • Driving of the MISOA output signal is stopped. • RSPI function is disabled.	Mode fault error

On operation 1 described in Table 38.8, the RSPI does not detect an error. To prevent data omission during the writing to SPDR, the SPDR register should be written when a transmit buffer empty interrupt request occurs or while the SPSR.SPTEF flag is 1.

Likewise, the RSPI does not detect an error on operation 2. To prevent extraneous data from being read, the SPDR register should be read when an RSPI receive buffer full interrupt request occurs or while the SPSR.SPRF flag is 1.

Similarly, the RSPI does not detect an error on operation 3. In a serial transfer that was started before the shift register was updated, the RSPI sends the data that was received in the previous serial transfer, and does not treat the operation indicated in 3 as an error. Note that the received data from the previous serial transfer is retained in the receive buffer of SPDR, thus it can be correctly read (if SPDR is not read before the end of the serial transfer, an overrun error may occur).

An overrun error shown in 4 is described in section 38.3.8.1, Overrun Error. A parity error shown in 5 is described in section 38.3.8.2, Parity Error. A mode fault error shown in 6 to 8 is described in section 38.3.8.3, Mode Fault Error. For the transmit and receive interrupts, refer to section 38.3.7, Transmit Buffer Empty/Receive Buffer Full Interrupts.

Downloaded from Arrow.com.

38.3.8.1 Overrun Error

If a serial transfer ends when the receive buffer of SPDR is full, the RSPI detects an overrun error, and sets the SPSR.OVRF flag to 1. When the OVRF flag is 1, the RSPI does not copy data from the shift register to the receive buffer so that the data prior to the occurrence of the error is retained in the receive buffer. To set the OVRF flag to 0, write 0 to the OVRF flag after the CPU has read SPSR with the OVRF flag set to 1.
Figure 38.27 shows an example of operations of the SPRF and OVRF flags. The SPSR and SPDR accesses shown in Figure 38.27 indicate the condition of accesses to SPSR and SPDR, respectively, where W denotes a write cycle, and R a read cycle. In the example in Figure 38.27, the RSPI performs an 8-bit serial transfer in which the SPCMDm.CPHA bit is 1 and the SPCMDm.CPOL bit is 0. The numbers given under the RSPCKA waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

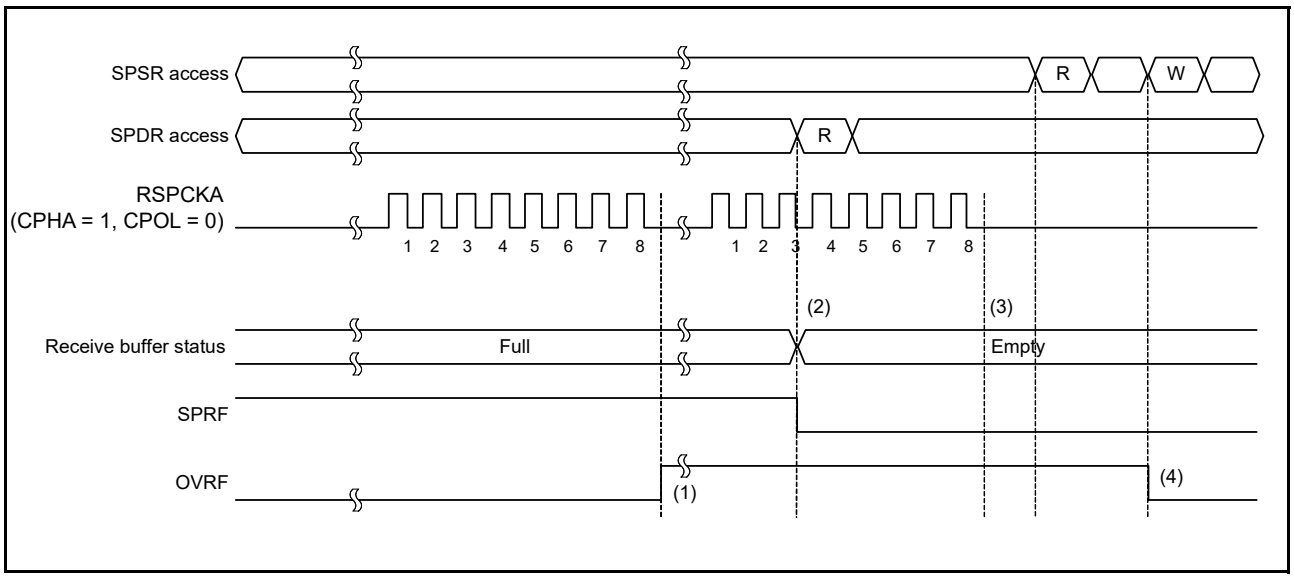


Figure 38.27 Operation Example of SPRF and OVRF Flags

The operation of the flags at the timing shown in steps (1) to (4) in the figure is described below.

(1) If a serial transfer terminates with the receive buffer full (the SPRF flag is 1), the RSPI detects an overrun error, and sets the OVRF flag to 1. The RSPI does not copy the data in the shift register to the receive buffer. Even if the SPPE bit is 1, parity errors are not detected. In master mode, the RSPI copies the pointer value to SPCMDm register to the SPSSR.SPECM[2:0] bits.

(2) When SPDR is read, the RSPI outputs the data in the receive buffer. At this time the SPRF flag becomes 0. Even if the receive buffer becomes empty, the OVRF flag does not become 0.

(3) If the serial transfer ends with the OVRF flag being 1 (an overrun error occurs), the RSPI does not copy the data in the shift register to the receive buffer (the SPRF flag remains 0). A receive buffer full interrupt is not generated. Even if the SPPE bit is 1, parity errors are not detected. When in master mode, the RSPI does not update the SPSSR.SPECM[2:0] bits. When in an overrun error state and the RSPI does not copy the received data from the shift register to the receive buffer, upon termination of the serial transfer, the RSPI determines that the shift register is empty; in this manner, data transfer from the transmit buffer to the shift register is enabled.

(4) If 0 is written to the OVRF flag after SPSR is read when the OVRF flag is 1, the OVRF flag is set to 0.

The occurrence of an overrun can be checked either by reading SPSR or by using an RSPI error interrupt and reading SPSR. When executing a serial transfer, measures should be taken to ensure the early detection of overrun errors, such as reading SPSR immediately after SPDR is read. When the RSPI is used in master mode, the pointer value to SPCMDm register at the occurrence of the error can be checked by reading the SPSSR.SPECM[2:0] bits.
If an overrun error occurs and the OVRF flag is set to 1, normal reception operations cannot be performed until the OVRF flag is set to 0.

Downloaded from Arrow.com.

When the RSPCK auto-stop function is enabled in master mode, an overrun error does not occur. Figure 38.28 and Figure 38.29 show the clock stop waveform when a serial transfer continues while the receive buffer is full in master mode.

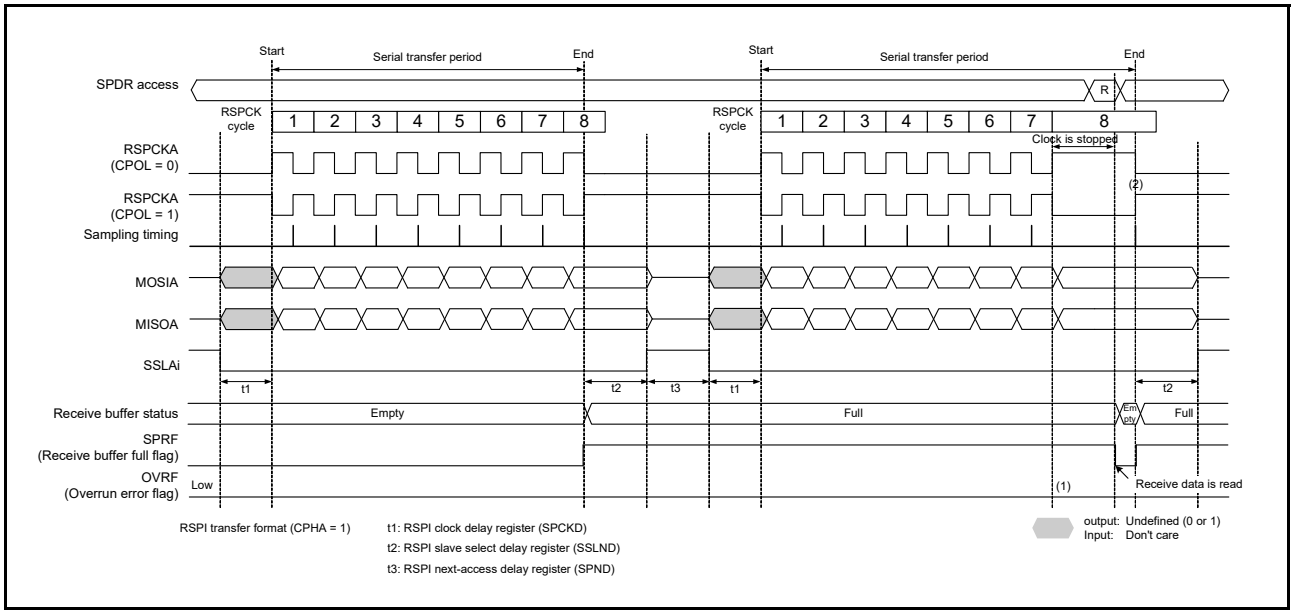


Figure 38.28 Clock Stop Waveform When a Serial Transfer Continues While the Receive Buffer is Full in Master Mode (CPHA = 1)

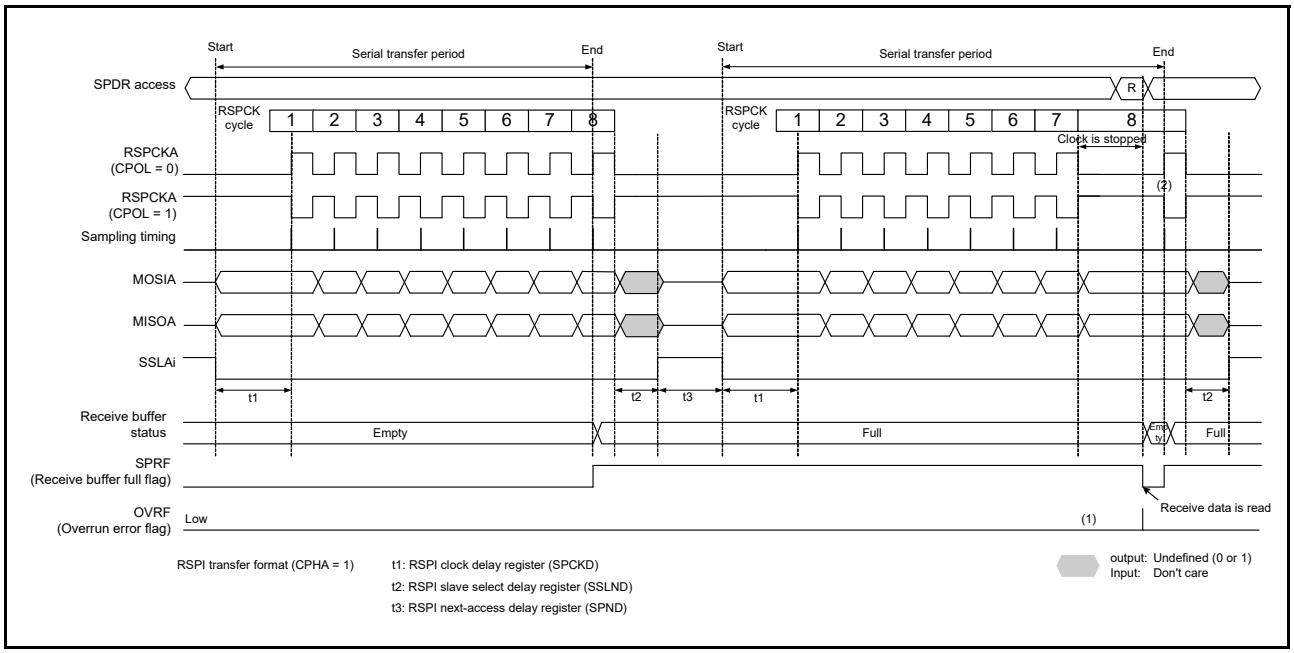


Figure 38.29 Clock Stop Waveform When a Serial Transfer Continues While the Receive Buffer is Full in Master Mode (CPHA = 0)

The operation of the flags at the timings shown in steps (1) and (2) in the figure is described below.

(1) When the receive buffer is full, an overrun error does not occur because the RSPCK clock is stopped.

(2) If SPDR is read while the clock is stopped, data in the receive buffer can be read. The RSPCK clock restarts after reading the receive buffer (after the SPRF flag becomes 0).

Downloaded from Arrow.com.

38.3.8.2 Parity Error

If full-duplex synchronous serial communications is performed with the SPCR.TXMD bit set to 0 and the SPCR2.SPPE bit set to 1, when serial transfer ends, the RSPI checks whether there are parity errors. Upon detecting a parity error in the received data, the RSPI sets the SPSR.PERF flag to 1. Since the RSPI does not copy the data in the shift register to the receive buffer when the SPSR.OVRF flag is set to 1, parity error detection is not performed for the received data. To set the PERF flag to 0, write 0 to the PERF flag after SPSR register is read with the PERF flag set to 1.
Figure 38.30 shows an example of operation of the OVRF and PERF flags. The SPSR access shown in Figure 38.30 indicates the condition of access to SPSR register, where W denotes a write cycle, and R a read cycle. In the example of Figure 38.30, full-duplex synchronous serial communications is performed while the SPCR.TXMD bit is 0 and the SPCR2.SPPE bit is 1. The RSPI performs an 8-bit serial transfer in which the SPCMDm.CPHA bit is 1 and the SPCMDm.CPOL bit is 0. The numbers given under the RSPCKA waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

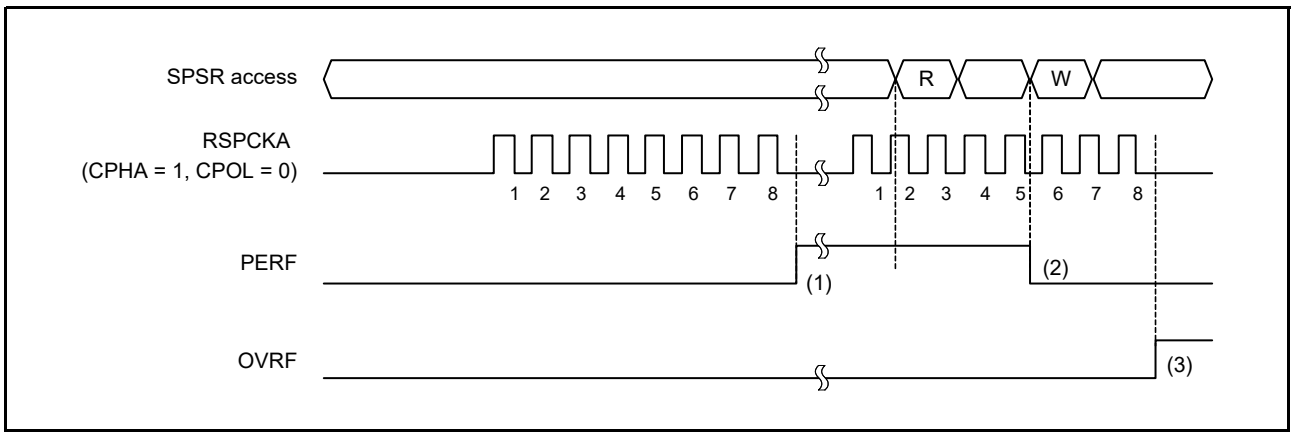


Figure 38.30 Operation Example of PERF Flag

The operation of the flags at the timing shown in steps (1) to (3) in the figure is described below.

(1) If a serial transfer terminates with the RSPI not detecting an overrun error, the RSPI copies the data in the shift register to the receive buffer. The RSPI judges the received data at this timing, and sets the PERF flag to 1 if a parity error is detected. In master mode, the RSPI copies the pointer value to SPCMDm register to the SPSSR.SPECM[2:0] bits.

(2) If 0 is written to the PERF flag after SPSR register is read when the PERF flag is 1, the PERF flag is set to 0.

(3) When the RSPI detects an overrun error and serial transfer is terminated, the data in the shift register is not copied to the receive buffer. The RSPI does not perform parity error detection at this timing.

The occurrence of a parity error can be checked either by reading the SPSR register or by using an RSPI error interrupt and reading the SPSR register. When executing a serial transfer, measures should be taken to ensure the early detection of parity errors, such as reading SPSR. When the RSPI is used in master mode, the pointer value to SPCMDm register at the occurrence of the error can be checked by reading the SPSSR.SPECM[2:0] bits.

Downloaded from Arrow.com.

38.3.8.3 Mode Fault Error

The RSPI operates in multi-master mode when the SPCR.MSTR bit is 1, the SPCR.SPMS bit is 0, and the SPCR.MODFEN bit is 1. If the active level is input with respect to the SSLA0 input signal of the RSPI in multi-master mode, the RSPI detects a mode fault error irrespective of the status of the serial transfer, and sets the SPSR.MODF flag to 1. Upon detecting the mode fault error, the RSPI copies the value of the pointer to SPCMDm to the SPSSR.SPECM[2:0] bits. The active level of the SSLA0 signal is determined by the SSLP.SSL0P bit.

When the MSTR bit is 0, the RSPI operates in slave mode. The RSPI detects a mode fault error if the MODFEN bit of the RSPI in slave mode is 1, and the SPMS bit is 0, and if the SSLA0 input signal is negated during the serial transfer period (from the time the driving of valid data is started to the time the final valid data is fetched).

Upon detecting a mode fault error, the RSPI stops driving of the output signals and clears the SPCR.SPE bit to 0 (refer to section 38.3.9, Initializing RSPI). In the case of multi-master configuration, detection of a mode fault error is used to stop driving of the output signals and the RSPI function, which allows the master right to be released.

The occurrence of a mode fault error can be checked either by reading SPSR or by using an RSPI error interrupt and reading SPSR. Detecting mode fault errors without utilizing the RSPI error interrupt requires polling of SPSR. When using the RSPI in master mode, the pointer value to SPCMDm register at the occurrence of the error can be checked by reading the SPSSR.SPECM[2:0] bits.

When the MODF flag is 1, writing of the value 1 to the SPE bit is ignored by the RSPI. To enable the RSPI function after the detection of a mode fault error, set the MODF flag to 0.

Downloaded from Arrow.com.

38.3.9 Initializing RSPI

If 0 is written to the SPCR.SPE bit or the RSPI sets the SPE bit to 0 because of the detection of a mode fault error, the RSPI disables the RSPI function, and initializes some of the module functions. When a system reset is generated, the RSPI initializes all of the module functions. The following describes initialization by the clearing of the SPCR.SPE bit and initialization by a system reset.

38.3.9.1 Initialization by Clearing the SPE Bit

When the SPCR.SPE bit is set to 0, the RSPI performs the following initialization:

- Suspending any serial transfer that is being executed
- Stopping the driving of output signals (Hi-Z) in slave mode
- Initializing the internal state of the RSPI
- Initializing the transmit buffer of the RSPI (Set the SPTEF flag to 1)

Initialization by the clearing of the SPE bit does not initialize the control bits of the RSPI. For this reason, the RSPI can be started in the same transfer mode as prior to the initialization if the SPE bit is set to 1 again.
The SPSR.SPRF, SPSR.OVRF, SPSR.MODF, and SPSR.PERF flags are not initialized, nor is the value of the RSPI sequence status register (SPSSR) initialized. For this reason, even after the RSPI is initialized, data from the receive buffer can be read in order to check the status of error occurrence during an RSPI transfer.
The transmit buffer is initialized to an empty state (the SPTEF flag is 1). Therefore, if the SPCR.SPTIE bit is set to 1 after RSPI initialization, a transmit buffer empty interrupt is generated. When the RSPI is initialized, in order to disable any transmit buffer empty interrupt, 0 should be written to the SPTIE bit simultaneously with the writing of 0 to the SPE bit.

38.3.9.2 System Reset

The initialization by a system reset completely initializes the RSPI through the initialization of all bits for controlling the RSPI, initialization of the status bits, and initialization of data registers, in addition to the requirements described in section 38.3.9.1, Initialization by Clearing the SPE Bit.

Downloaded from Arrow.com.

38.3.10 SPI Operation

38.3.10.1 Master Mode Operation

The only difference between single-master mode operation and multi-master mode operation lies in mode fault error detection (refer to section 38.3.8, Error Detection). When operating in single-master mode, the RSPI does not detect mode fault errors whereas the RSPI running in multi-master mode does detect mode fault errors. This section explains operations that are common to single-master mode and multi-master mode.

(1) Starting a Serial Transfer

The RSPI updates the data in the transmit buffer (SPTX) when data is written to the RSPI data register (SPDR) with the RSPI transmit buffer being empty (the SPTEF flag is 1 and data for the next transfer is not set). When the shift register is empty after the number of frames set in the SPDCR.SPFC[1:0] bits are written to the SPDR, the RSPI copies data from the transmit buffer to the shift register and starts serial transfer. Upon copying transmit data to the shift register, the RSPI changes the status of the shift register to “full”, and upon termination of serial transfer, it changes the status of the shift register to “empty”. The status of the shift register cannot be referenced.
For details on the RSPI transfer format, refer to section 38.3.5, Transfer Format. The polarity of the SSLAi output pins depends on the SSLP register settings.

(2) Terminating a Serial Transfer

Irrespective of the SPCMDm.CPHA bit, the RSPI terminates the serial transfer after transmitting an RSPCKA edge corresponding to the final sampling timing. If free space is available in the receive buffer (SPRX) (the SPRF flag is 0), upon termination of serial transfer, the RSPI copies data from the shift register to the receive buffer of the SPDR register. It should be noted that the final sampling timing varies depending on the bit length of transfer data. In master mode, the RSPI data length depends on the SPCMDm.SPB[3:0] bit setting. The polarity of the SSLAi output pin depends on the SSLP register settings.
For details on the RSPI transfer format, refer to section 38.3.5, Transfer Format.

Downloaded from Arrow.com.

(3) Sequence Control

The transfer format that is employed in master mode is determined by SPSCR, SPCMDm, SPBR, SPCKD, SSLND, and SPND registers.

SPSCR is a register used to determine the sequence configuration for serial transfers that are executed by the RSPI in master mode. The following items are set in SPCMDm register: SSLAi pin output signal value, MSB/LSB first, data length, some of the bit rate settings, RSPCK polarity/phase, whether SPCKD is to be referenced, whether SSLND is to be referenced, and whether SPND is to be referenced. SPBR holds some of the bit rate settings; SPCKD, an RSPI clock delay value; SSLND, an SSL negation delay; and SPND, a next-access delay value.

According to the sequence length that is assigned to SPSCR, the RSPI makes up a sequence comprised of a part or all of SPCMDm register. The RSPI contains a pointer to the SPCMDm register that makes up the sequence. The value of this pointer can be checked by reading the SPSSR.SPCP[2:0] bits. When the SPCR.SPE bit is set to 1 and the RSPI function is enabled, the RSPI loads the pointer to the commands in SPCMD0, and incorporates the SPCMD0 settings into the transfer format at the beginning of serial transfer. The RSPI increments the pointer each time the next-access delay period for a data transfer ends. Upon completion of the serial transfer that corresponds to the final command comprising the sequence, the RSPI sets the pointer in SPCMD0, and in this manner the sequence is executed repeatedly.

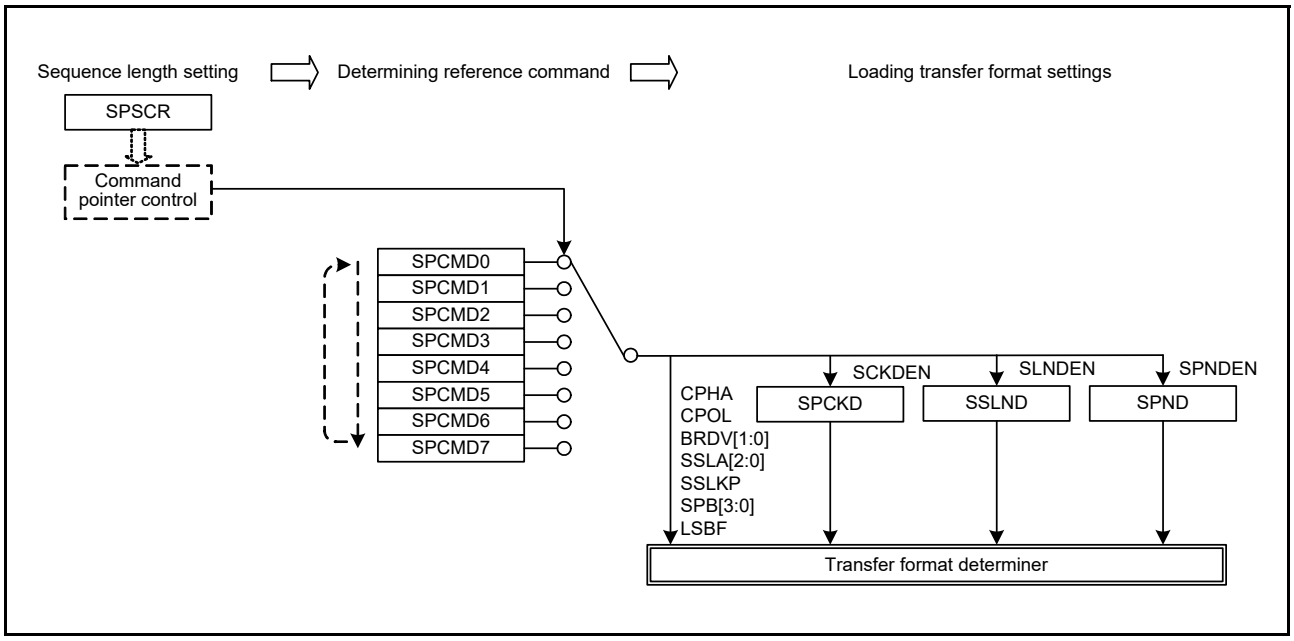


Figure 38.31 Procedure for Determining the Form of Serial Transfer in Master Mode

In this section, a frame is the combination of the data (SPDR) and the settings (SPCMDm).

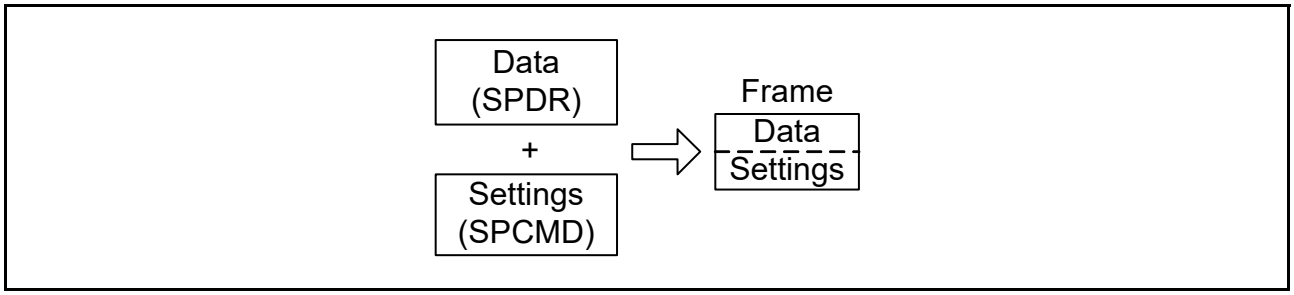


Figure 38.32 Concept of a Frame

Downloaded from Arrow.com.

Figure 38.33 shows the relationship between the command and the transmit and receive buffers in the sequence of operations specified by the settings in Table 38.4.

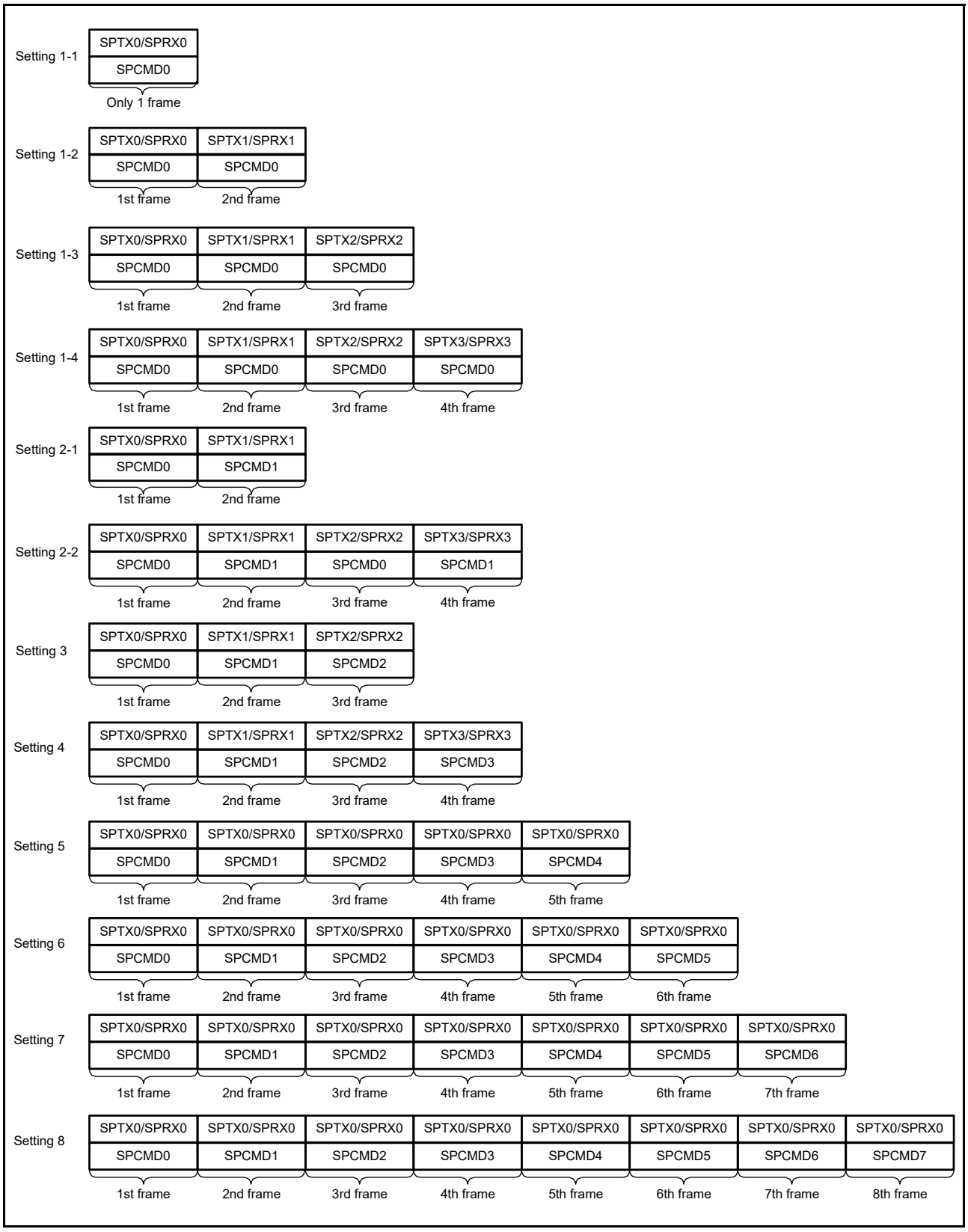


Figure 38.33 Correspondence between the RSPI Command Register and Transmit/Receive Buffers in Sequence Operations

Downloaded from Arrow.com.

(4) Burst Transfer

If the SPCMDm.SSLKP bit that the RSPI references during the current serial transfer is 1, the RSPI keeps the SSLAi signal level during the serial transfer until the beginning of the SSLAi signal assertion for the next serial transfer. If the SSLAi signal level for the next serial transfer is the same as the SSLAi signal level for the current serial transfer, the RSPI can execute continuous serial transfers while keeping the SSLAi signal assertion status (burst transfer).
Figure 38.34 shows an example of an SSLAi signal operation for the case where a burst transfer is implemented using SPCMD0 and SPCMD1 register settings. The text below explains the RSPI operations (1) to (7) as shown in Figure 38.34. It should be noted that the polarity of the SSLAi output signal depends on the SSLP register settings.

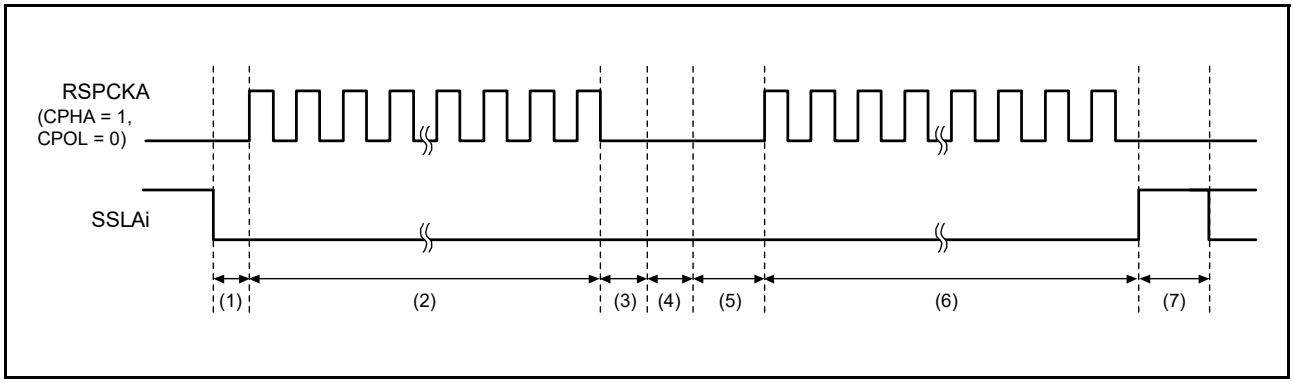


Figure 38.34 Example of Burst Transfer Operation Using SSLKP Bit

(1) Based on SPCMD0, the RSPI asserts the SSLAi signal and inserts RSPCK delays.
(2) The RSPI executes serial transfers according to SPCMD0.
(3) The RSPI inserts SSL negation delays.
(4) Since the SPCMD0.SSLKP bit is 1, the RSPI keeps the SSLAi signal value on SPCMD0. This period is sustained, at the shortest, for a period equal to the next-access delay of SPCMD0. If the shift register is empty after the passage of a minimum period, this period is sustained until the transmit data is stored in the shift register for the next transfer.
(5) Based on SPCMD1, the RSPI asserts the SSLAi signal and inserts RSPCK delays.
(6) The RSPI executes serial transfers according to SPCMD1.
(7) Because the SPCMD1.SSLKP bit is 0, the RSPI negates the SSLAi signal. In addition, a next-access delay is inserted according to SPCMD1.

If the SSLAi signal output settings in the SPCMDm register in which 1 is assigned to the SSLKP bit are different from the SSLAi signal output settings in the SPCMDm register to be used in the next transfer, the RSPI switches the SSLAi signal status to SSLAi signal assertion ((5) in Figure 38.34) corresponding to the command for the next transfer. Note that if such an SSLAi signal switching occurs, the slaves that drive the MISOA signal compete, and collision of signal levels may occur.
The RSPI in master mode references the SSLAi signal operation within the module for the case where the SSLKP bit is not used. Even when the SPCMDm.CPHA bit is 0, the RSPI can accurately start serial transfers by using the SSLAi signal assertion for the next transfer that is detected internally.

Downloaded from Arrow.com.

(5) RSPCK Delay (t1)

The RSPCK delay value of the RSPI in master mode depends on the SPCMDm.SCKDEN bit setting and the SPCKD register setting. The RSPI determines the SPCMDm register to be referenced during serial transfer by pointer control, and determines an RSPCK delay value during serial transfer by using the SPCMDm.SCKDEN bit and SPCKD, as listed in Table 38.9. For a definition of RSPCK delay, refer to section 38.3.5, Transfer Format.

Table 38.9 Relationship among SCKDEN Bit, SPCKD, and RSPCK Delay Value

SPCMDm.SCKDEN Bit	SPCKD.SCKDL[2:0] Bits	RSPCK Delay Value
0	000b to 111b	1 RSPCK
1	000b	1 RSPCK
	001b	2 RSPCK
	010b	3 RSPCK
	011b	4 RSPCK
	100b	5 RSPCK
	101b	6 RSPCK
	110b	7 RSPCK
	111b	8 RSPCK

(6) SSL Negation Delay (t2)

The SSL negation delay value of the RSPI in master mode depends on the SPCMDm.SLNDEN bit setting and the SSLND register setting. The RSPI determines the SPCMDm register to be referenced during serial transfer by pointer control, and determines an SSL negation delay value during serial transfer by using the SPCMDm.SLNDEN bit and SSLND, as listed in Table 38.10. For a definition of SSL negation delay, refer to section 38.3.5, Transfer Format.

Table 38.10 Relationship among SLNDEN Bit, SSLND, and SSL Negation Delay Value

SPCMDm.SLNDEN Bit	SSLND.SLNDL[2:0] Bits	SSL Negation Delay Value
0	000b to 111b	1 RSPCK
1	000b	1 RSPCK
	001b	2 RSPCK
	010b	3 RSPCK
	011b	4 RSPCK
	100b	5 RSPCK
	101b	6 RSPCK
	110b	7 RSPCK
	111b	8 RSPCK

Downloaded from Arrow.com.

(7) Next-Access Delay (t3)

The next-access delay value of the RSPI in master mode depends on the SPCMDm.SPNDEN bit setting and the SPND setting. The RSPI determines the SPCMDm register to be referenced during serial transfer by pointer control, and determines a next-access delay value during serial transfer by using the SPCMDm.SPNDEN bit and SPND, as listed in Table 38.11. For a definition of next-access delay, refer to section 38.3.5, Transfer Format.

Table 38.11 Relationship among SPNDEN Bit, SPND, and Next-Access Delay Value

SPCMDm.SPNDEN Bit	SPND.SPNDL[2:0] Bits	Next-Access Delay Value
0	000b to 111b	1 RSPCK + 2 PCLK
1	000b	1 RSPCK + 2 PCLK
	001b	2 RSPCK + 2 PCLK
	010b	3 RSPCK + 2 PCLK
	011b	4 RSPCK + 2 PCLK
	100b	5 RSPCK + 2 PCLK
	101b	6 RSPCK + 2 PCLK
	110b	7 RSPCK + 2 PCLK
	111b	8 RSPCK + 2 PCLK

Downloaded from Arrow.com.

(8) Initialization Flowchart

Figure 38.35 is a flowchart illustrating an example of initialization in SPI operation when the RSPI is used in master mode. For a description of how to set up the interrupt controller, DMAC, and I/O ports, refer to the descriptions given in the individual blocks.

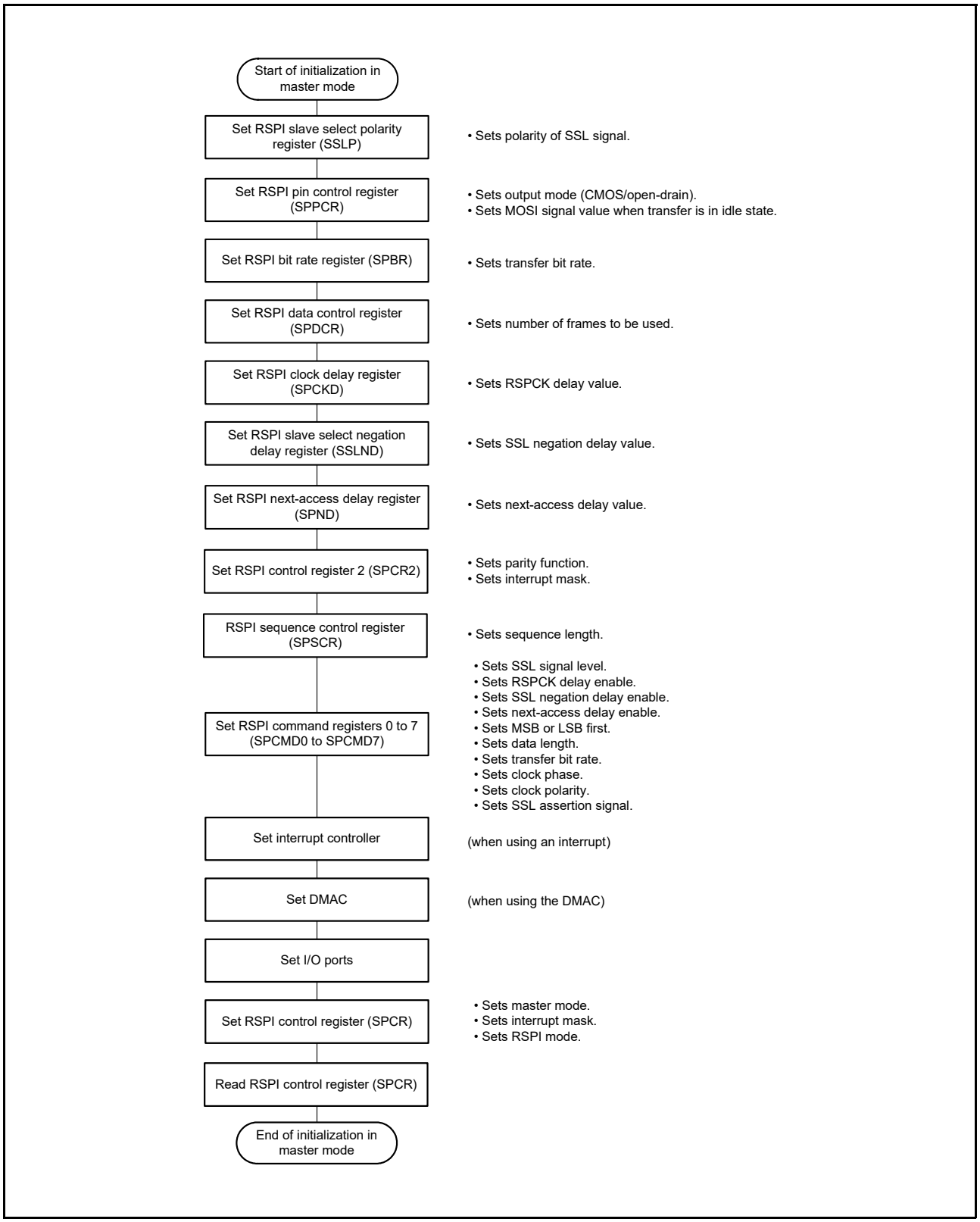


Figure 38.35 Example of Initialization Flowchart in Master Mode (SPI Operation)

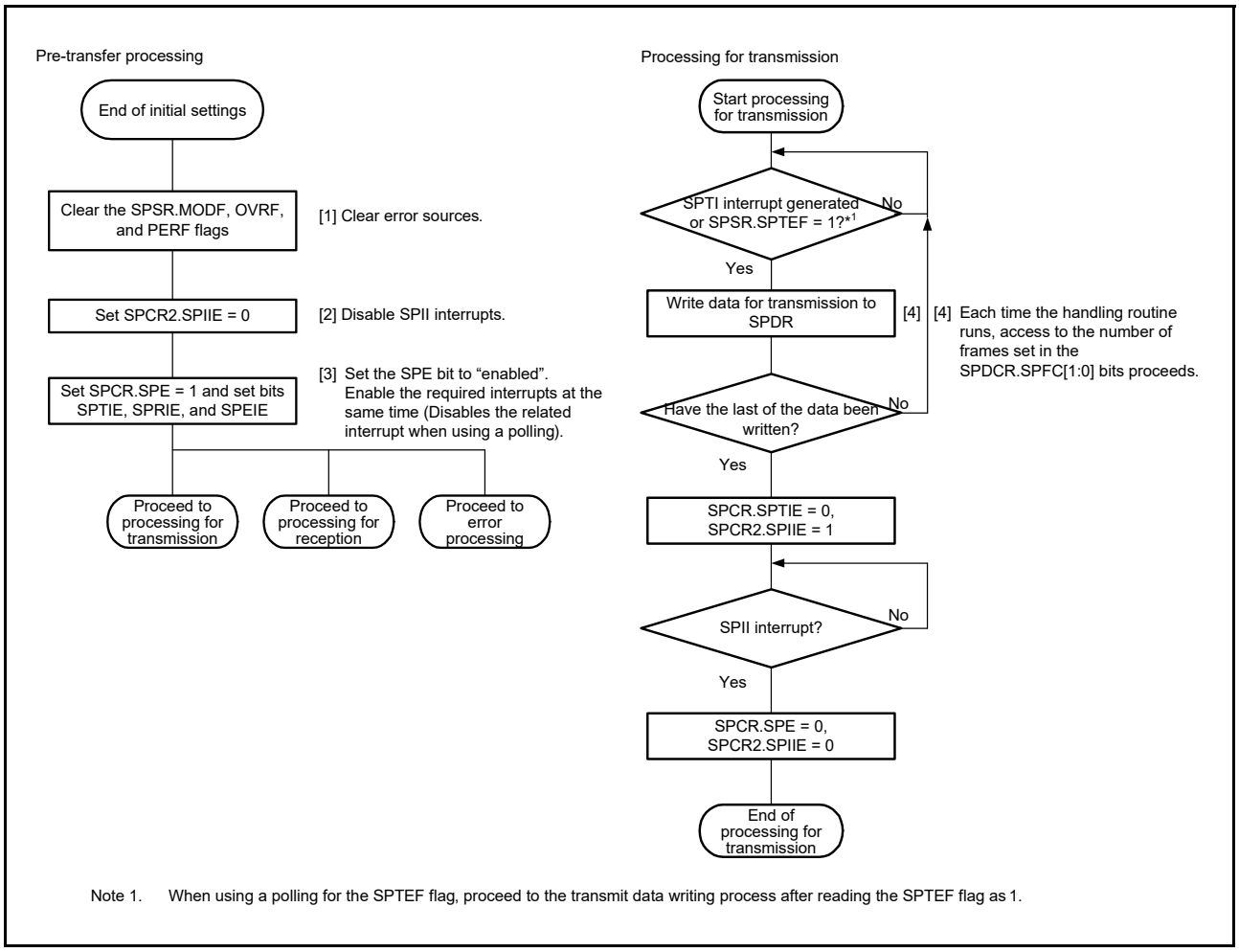
Downloaded from Arrow.com.

(9) Software Processing Flow

Figure 38.36 to Figure 38.38 show examples of the flow of software processing.

(a) Transmit Processing Flow

When transmitting data, the CPU will be notified of the completion of data transmission by enabling the SPI interrupt after the last writing of data for transmission.
The completion of data transmission can also be checked by polling to see if the SPSR.IDLNF flag has become 0, instead of using the SPII interrupt. However, one cycle of PCLK is required for the time from when data for transmission is written in the SPDR register to when the IDLNF flag becomes 1. After the last data is written in the SPDR register, discard the value of the SPSR register once not to judge the condition with the IDLNF flag which has not yet become 1, and read and use the value of the SPSR.IDLNF flag to confirm the completion of data transmission.



Note 1. When using a polling for the SPTEF flag, proceed to the transmit data writing process after reading the SPTEF flag as 1.

Figure 38.36 Flowchart in Master Mode (Transmission)

Downloaded from Arrow.com.

(b) Receive Processing Flow

The RSPI does not handle receive-only operation, so processing for transmission is required.

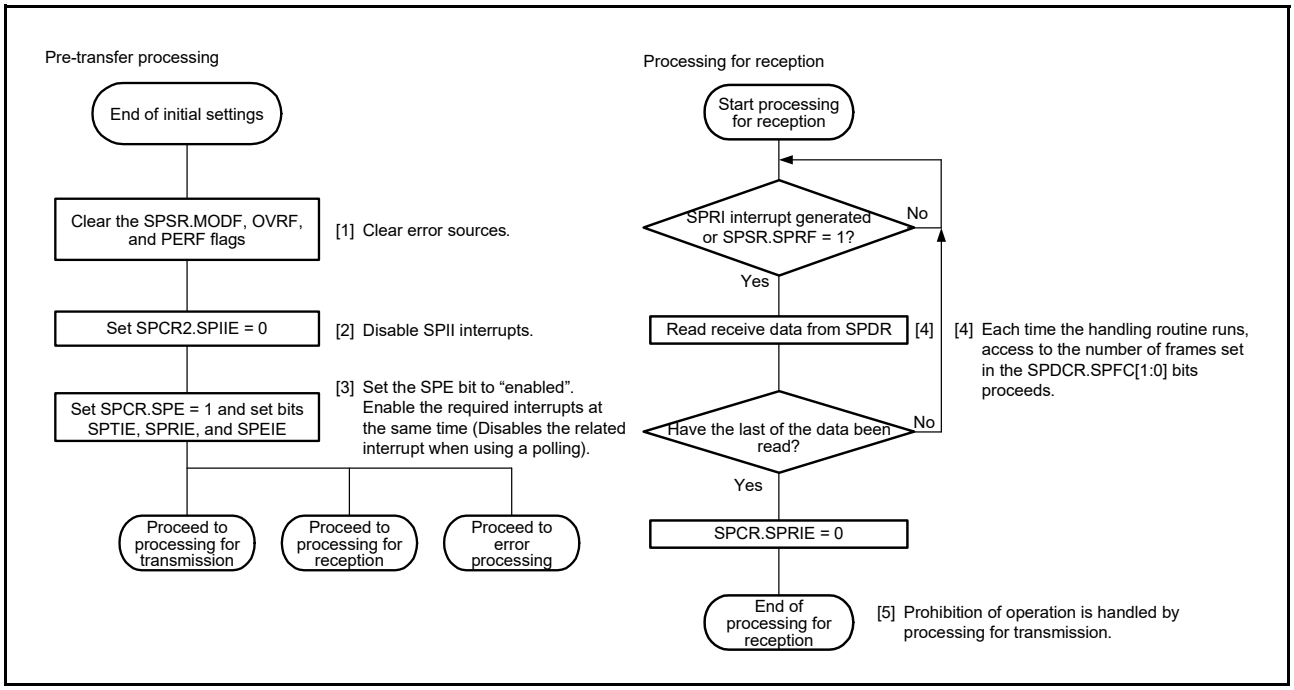


Figure 38.37 Flowchart in Master Mode (Reception)

Downloaded from Arrow.com.

(c) Flow of Error Processing

The RSPI has three types of error. When a mode fault error is generated, the SPCR.SPE bit is automatically cleared, stopping operations for transmission and reception. For errors from other sources, however, the SPCR.SPE bit is not cleared and operations for transmission and reception continue; accordingly, we recommend clearing of the SPCR.SPE bit to stop operations in the case of errors other than mode fault errors. Not doing so will lead to updating of the SPSSR.SPECM[2:0] bits.

When interrupts are used and an error occurs, if the ICU.IRn.IR flag for the SPTI or SPRI interrupt request is set to 1, clear the ICU.IRn.IR flag in the error processing routine. If the SPRI interrupt request is indicated, read the receive buffer and initialize the sequencer in the RSPI.

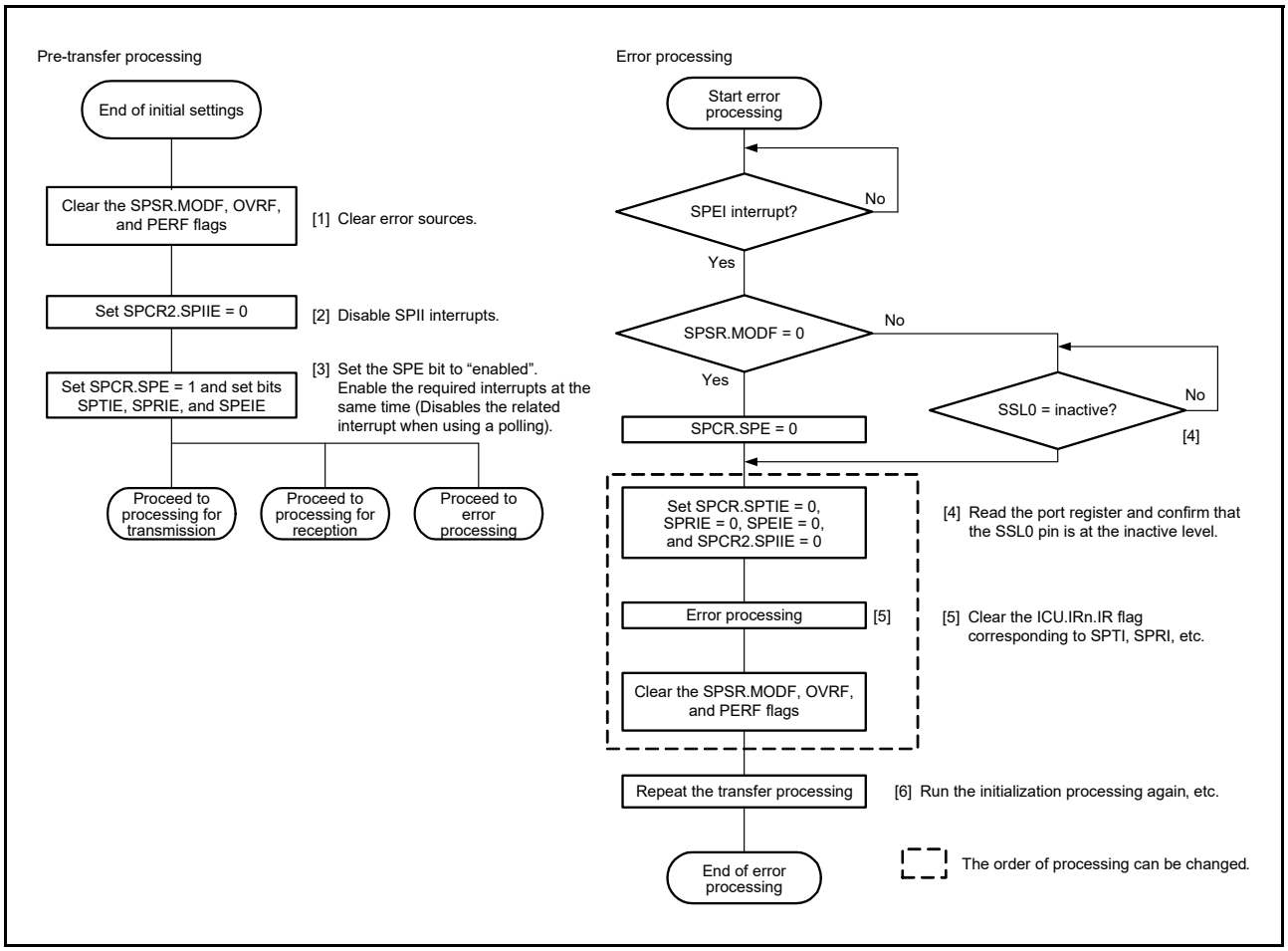


Figure 38.38 Flowchart for Master Mode (Error Processing)

Downloaded from Arrow.com.

38.3.10.2 Slave Mode Operation

(1) Starting a Serial Transfer

If the SPCMD0.CPHA bit is 0, when detecting an SSLA0 input signal assertion, the RSPI needs to start driving valid data to the MISOA output signal. For this reason, when the CPHA bit is 0, the assertion of the SSLA0 input signal triggers the start of a serial transfer.

If the CPHA bit is 1, when detecting the first RSPCKA edge in an SSLA0 signal asserted condition, the RSPI needs to start driving valid data to the MISOA output signal. For this reason, when the CPHA bit is 1, the first RSPCKA edge in an SSLA0 signal asserted condition triggers the start of a serial transfer.

When detecting the start of a serial transfer in a condition in which the shift register is empty, the RSPI changes the status of the shift register to “full”, so that data cannot be copied from the transmit buffer to the shift register when serial transfer is in progress. If the shift register was full before the serial transfer started, the RSPI leaves the status of the shift register unchanged, in the full state.

Irrespective of the CPHA bit setting, the timing at which the RSPI starts driving of the MISOA output signal is the SSLA0 signal assertion timing. The data which is output by the RSPI is either valid or invalid, depending on the CPHA bit setting.

For details on the RSPI transfer format, refer to section 38.3.5, Transfer Format. The polarity of the SSLA0 input signal depends on the setting of the SSLP.SSL0P bit.

(2) Terminating a Serial Transfer

Irrespective of the SPCMD0.CPHA bit, the RSPI terminates the serial transfer after detecting an RSPCKA edge corresponding to the final sampling timing. When free space is available in the receive buffer (the SPRF flag is 0), upon termination of serial transfer the RSPI copies received data from the shift register to the receive buffer of the SPDR register. Upon termination of a serial transfer the RSPI changes the status of the shift register to “empty”, regardless of the receive buffer state. A mode fault error occurs if the RSPI detects an SSLA0 input signal negation from the beginning of serial transfer to the end of serial transfer (refer to section 38.3.8, Error Detection).

The final sampling timing changes depending on the bit length of transfer data. In slave mode, the RSPI data length depends on the SPCMD0.SPB[3:0] bit setting. The polarity of the SSLA0 input signal depends on the SSLP.SSL0P bit setting.

For details on the RSPI transfer format, refer to section 38.3.5, Transfer Format.

(3) Notes on Single-Slave Operations

If the SPCMD0.CPHA bit is 0, the RSPI starts serial transfers when it detects the assertion edge for an SSLA0 input signal. In the type of configuration shown in Figure 38.7 as an example, if the RSPI is used in single-slave mode, the SSLA0 signal is fixed at the active state. Therefore, when the CPHA bit is set to 0, the RSPI cannot correctly start a serial transfer. To correctly execute transmit/receive operations by the RSPI in slave mode in a configuration in which the SSLA0 input signal is fixed at the active state, the CPHA bit should be set to 1. If there is a need for setting the CPHA bit to 0, the SSLA0 input signal should not be fixed.

Downloaded from Arrow.com.

(4) Burst Transfer

If the SPCMD0.CPHA bit is 1, continuous serial transfer (burst transfer) can be executed while retaining the assertion state for the SSLA0 input signal. If the CPHA bit is 1, the period from the first RSPCKA edge to the sampling timing for the reception of the final bit in an SSLA0 signal active state corresponds to a serial transfer period. Even when the SSLA0 input signal remains at the active level, the RSPI can accommodate burst transfers because it can detect the start of an access.
If the CPHA bit is 0, the second and subsequent serial transfers during burst transfer cannot be executed correctly.

(5) Initialization Flowchart

Figure 38.39 is a flowchart illustrating an example of initialization in SPI operation when the RSPI is used in slave mode. For a description of how to set up the interrupt controller, DMAC, and I/O ports, refer to the descriptions given in the individual blocks.

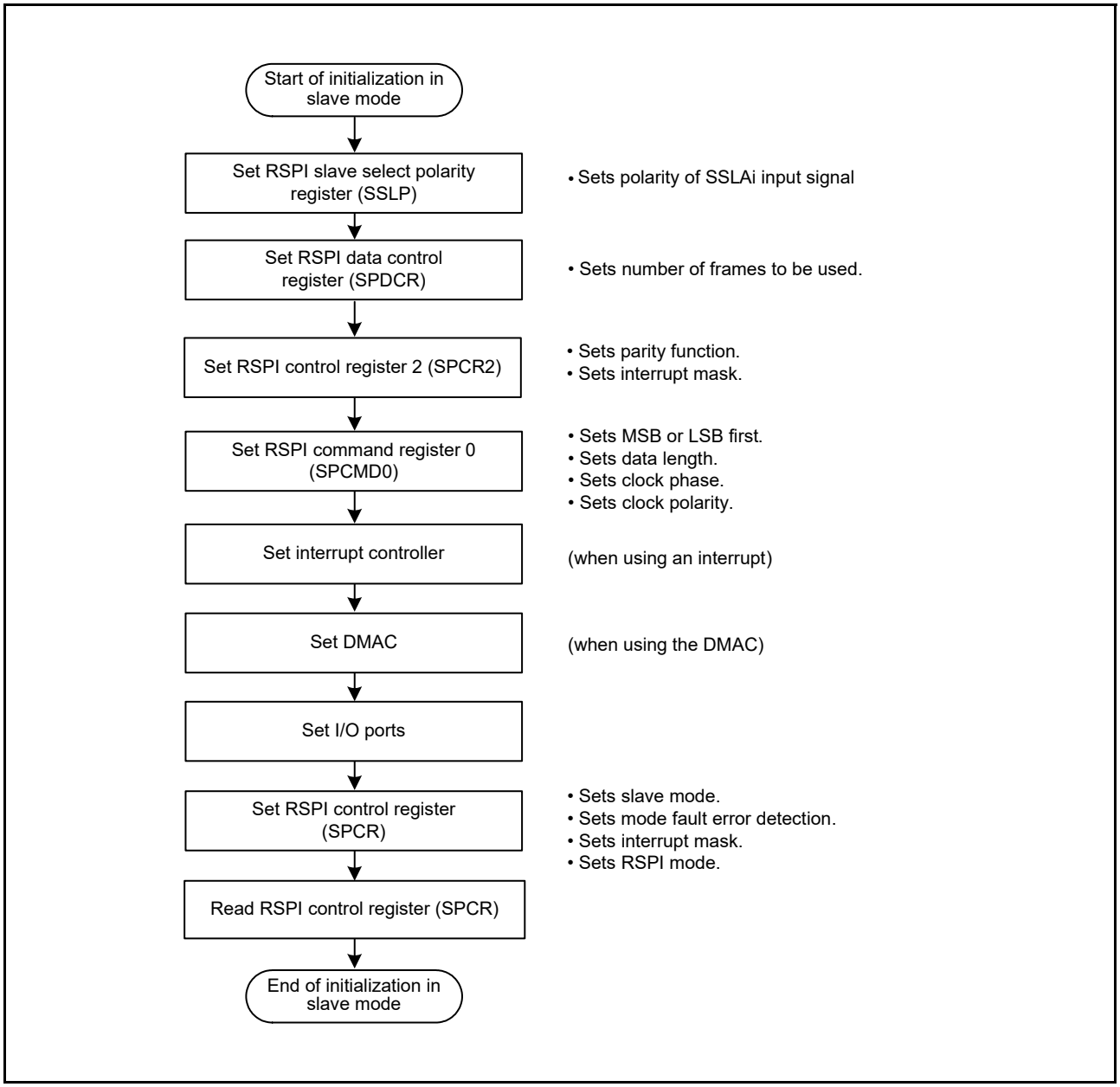


Figure 38.39 Example of Initialization Flowchart in Slave Mode (SPI Operation)

Downloaded from Arrow.com.

(6) Software Processing Flow

Figure 38.40 to Figure 38.42 show examples of the flow of software processing.

(a) Transmit Processing Flow

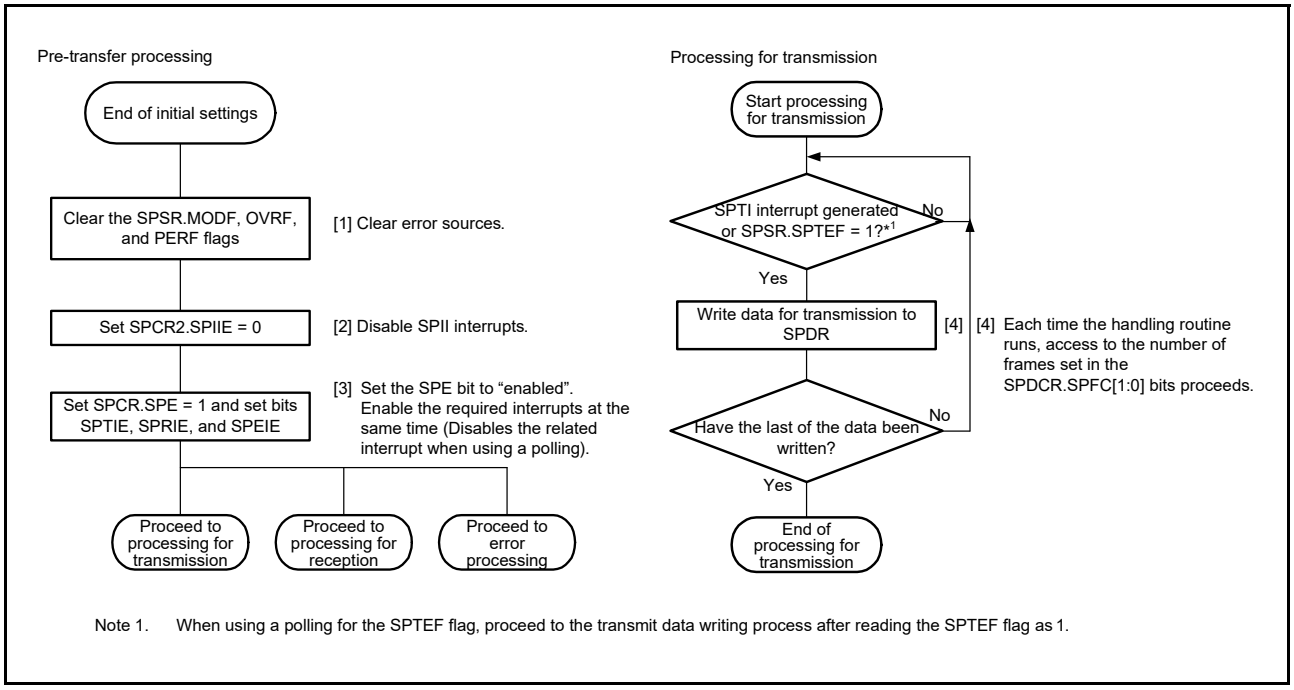


Figure 38.40 Flowchart in Slave Mode (Transmission)

(b) Receive Processing Flow

The RSPI does not handle receive-only operation, so processing for transmission is required.

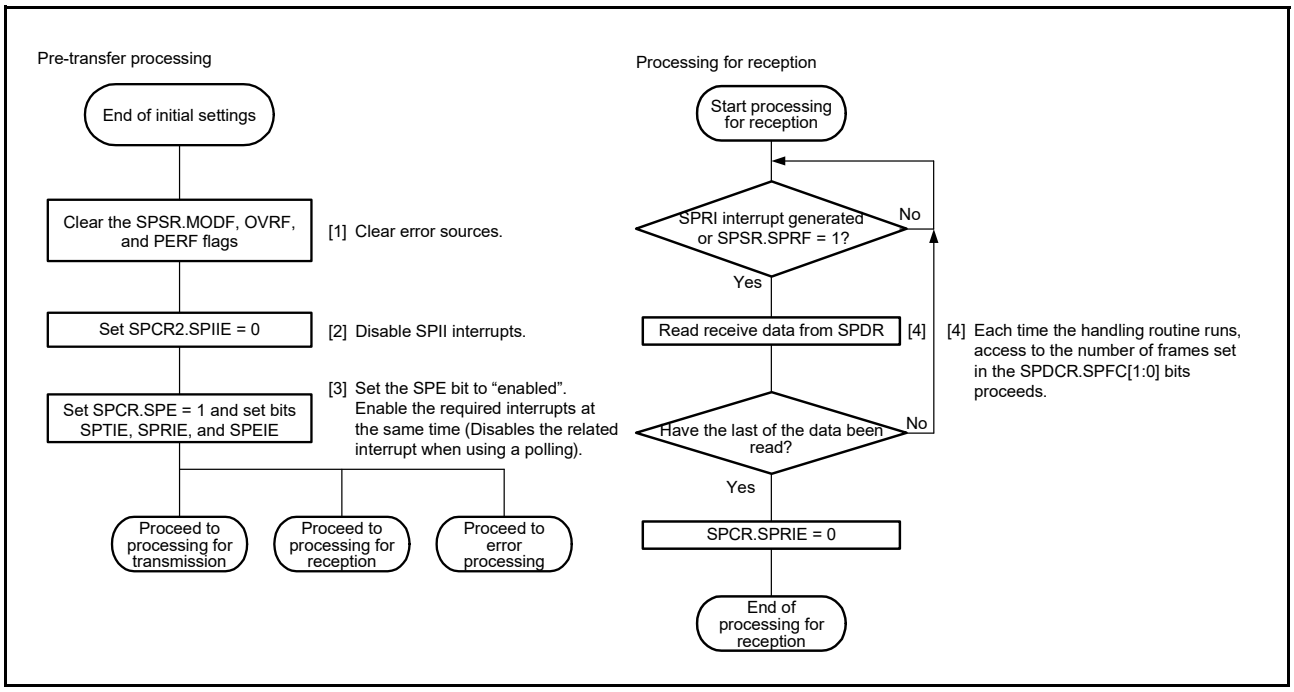


Figure 38.41 Flowchart in Slave Mode (Reception)

Downloaded from Arrow.com.

(c) Flow of Error Processing

In slave operation, even when a mode fault error is generated, the SPSR.MODF flag can be cleared regardless of the status of the SSLA0 pin.

When interrupts are used and an error occurs, if the ICU.IRn.IR flag for the SPTI or SPRI interrupt request is set to 1, clear the ICU.IRn.IR flag in the error processing routine. If the SPRI interrupt request is indicated, read the receive buffer and initialize the sequencer in the RSPI.

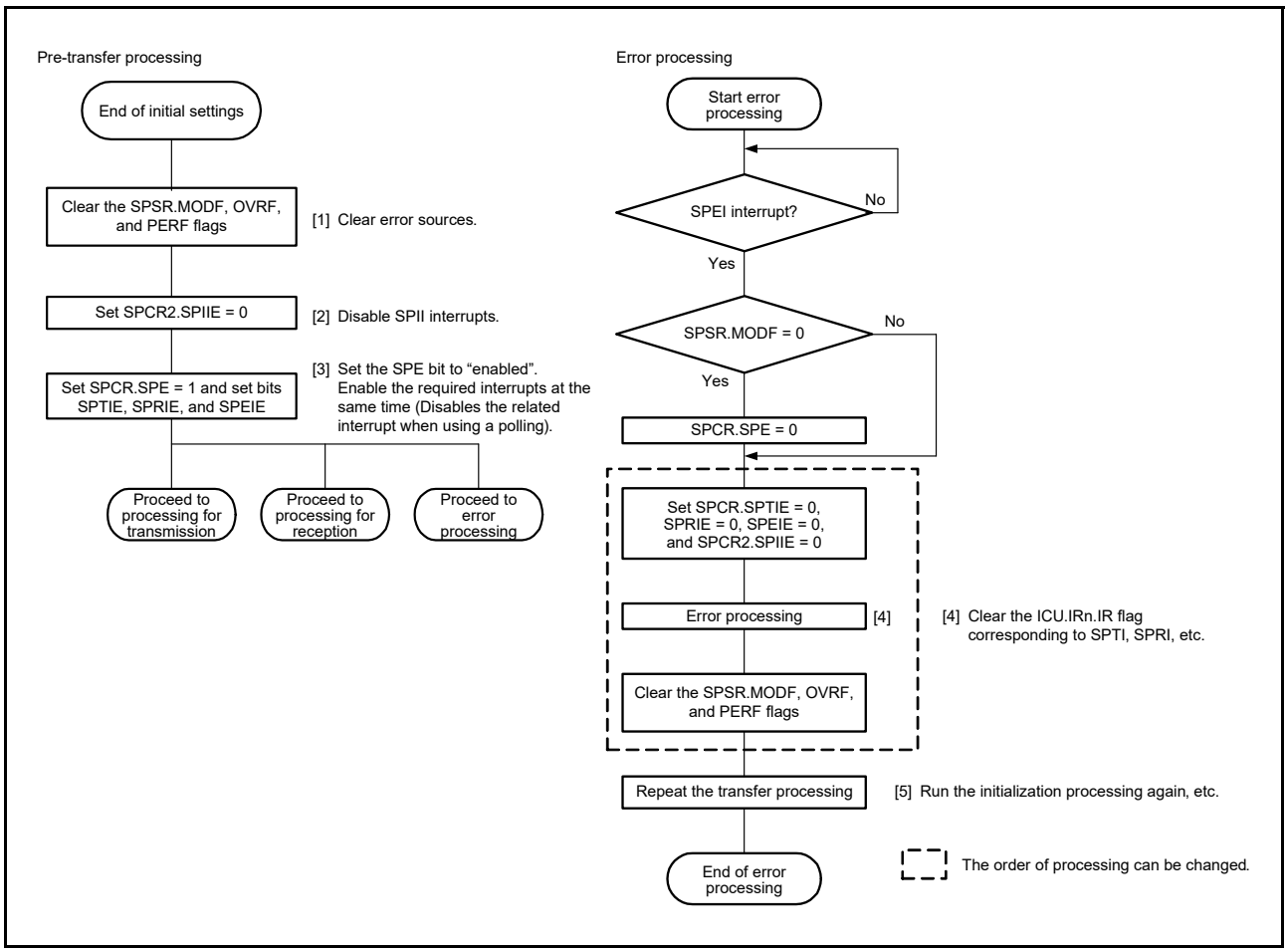


Figure 38.42 Flowchart for Slave Mode (Error Processing)

Downloaded from Arrow.com.

38.3.11 Clock Synchronous Operation

Setting the SPCR.SPMS bit to 1 selects clock synchronous operation of the RSPI. In clock synchronous operation, the SSLAi pin is not used, and the three pins of RSPCKA, MOSIA, and MISOA handle communications. The SSLAi pin is available as I/O port pins.
Although clock synchronous operation does not require use of the SSLAi pin, operation of the module is the same as in SPI operation. That is, in both master and slave operations, communications can be performed with the same flow as in SPI operation. However, mode fault errors are not detected because the SSLAi pin is not used.
Furthermore, do not set the SPCMDm.CPHA bit to 0 if clock synchronous operation is to proceed in slave mode (SPCR.MSTR = 0).

38.3.11.1 Master Mode Operation

(1) Starting a Serial Transfer

The RSPI updates the data in the transmit buffer (SPTX) of SPDR when data is written to the SPDR register with the transmit buffer being empty (the SPTEF flag is 1 and data for the next transfer is not set). When the shift register is empty after the number of frames set in the SPDCR.SPFC[1:0] bits are written to the SPDR, the RSPI copies data from the transmit buffer to the shift register and starts serial transmission. Upon copying transmit data to the shift register, the RSPI changes the status of the shift register to “full”, and upon termination of serial transfer, it changes the status of the shift register to “empty”. The status of the shift register cannot be referenced.
For details on the RSPI transfer format, refer to section 38.3.5, Transfer Format.
However, transfer in clock synchronous operation is conducted without the SSLA0 output signal.

(2) Terminating a Serial Transfer

The RSPI terminates the serial transfer after transmitting an RSPCKA edge corresponding to the sampling timing. If free space is available in the receive buffer (SPRX) (the SPRF flag is 0), upon termination of serial transfer, the RSPI copies data from the shift register to the receive buffer of the RSPI data register (SPDR).
It should be noted that the final sampling timing varies depending on the bit length of transfer data. In master mode, the RSPI data length depends on the SPCMDm.SPB[3:0] bit setting.
For details on the RSPI transfer format, refer to section 38.3.5, Transfer Format.
However, transfer in clock synchronous operation is conducted without the SSLA0 output signal.

(3) Sequence Control

The transfer format employed in master mode is determined by SPSCR, SPCMDm, SPBR, SPCKD, SSLND, and SPND registers. Although the SSLAi signals are not output in clock synchronous operation, these settings are valid.
SPSCR is a register used to determine the sequence configuration for serial transfers that are executed by the RSPI in master mode. The following items are set in SPCMDm register: SSLAi output signal value, MSB/LSB first, data length, some of the bit rate settings, RSPCKA polarity/phase, whether SPCKD is to be referenced, whether SSLND is to be referenced, and whether SPND is to be referenced. SPBR holds some of the bit rate settings; SPCKD, an RSPI clock delay value; SSLND, an SSL negation delay; and SPND, a next-access delay value.
According to the sequence length that is assigned to SPSCR, the RSPI makes up a sequence comprised of a part or all of SPCMDm register. The RSPI contains a pointer to the SPCMDm register that makes up the sequence. The value of this pointer can be checked by reading the SPSSR.SPCP[2:0] bits. When the SPCR.SPE bit is set to 1 and the RSPI function is enabled, the RSPI loads the pointer to the commands in SPCMD0 register, and incorporates the SPCMD0 register setting into the transfer format at the beginning of serial transfer. The RSPI increments the pointer each time the next-access delay period for a data transfer ends. Upon completion of the serial transfer that corresponds to the final command comprising the sequence, the RSPI sets the pointer in SPCMD0 register, and in this manner the sequence is executed repeatedly.

Downloaded from Arrow.com.

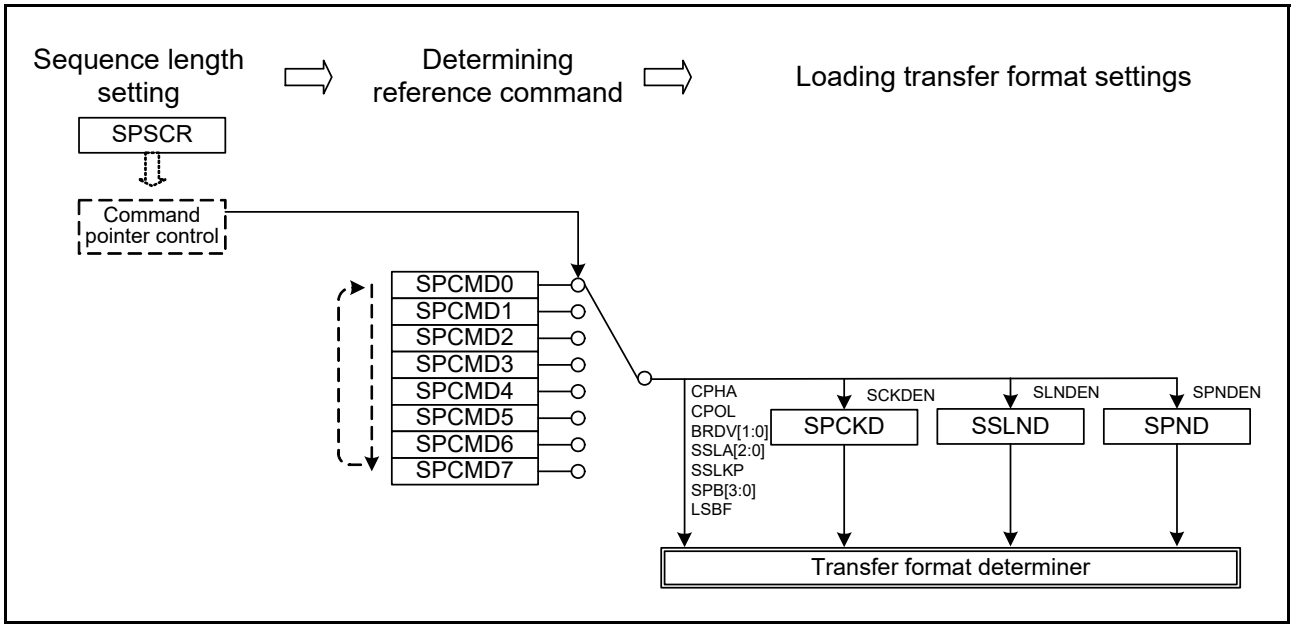


Figure 38.43 Procedure for Determining the Form of Serial Transmission in Master Mode

In this section, a frame is the combination of the data (SPDR) and the settings (SPCMDm).

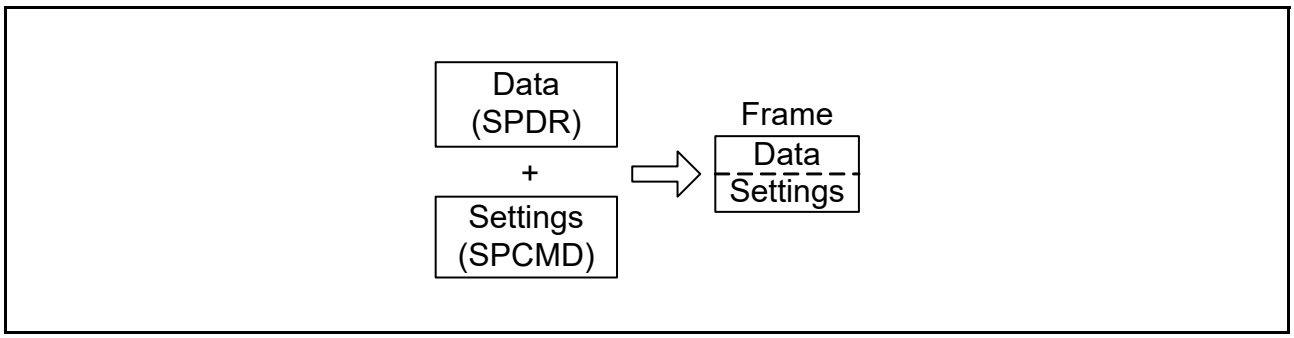


Figure 38.44 Concept of a Frame

Downloaded from Arrow.com.

Figure 38.45 shows the relationship between the command and the transmit and receive buffers in the sequence of operations specified by the settings in Table 38.4.

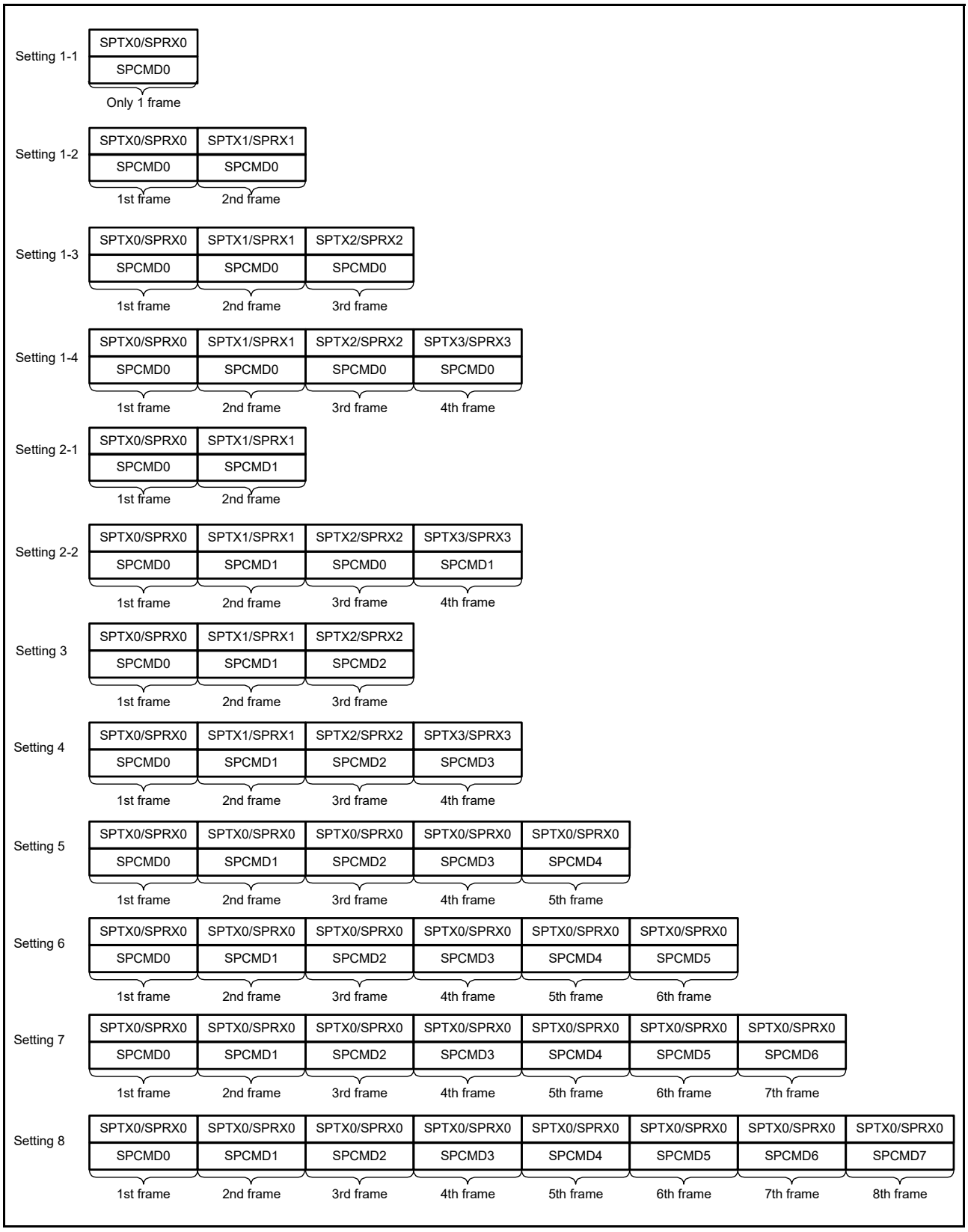


Figure 38.45 Correspondence between the RSPI Command Register and Transmit/Receive Buffers in Sequence Operations

Downloaded from Arrow.com.

(4) Initialization Flowchart

Figure 38.46 is a flowchart illustrating an example of initialization in clock synchronous operation when the RSPI is used in master mode. For a description of how to set up the interrupt controller, DMAC, and I/O ports, refer to the descriptions given in the individual blocks.

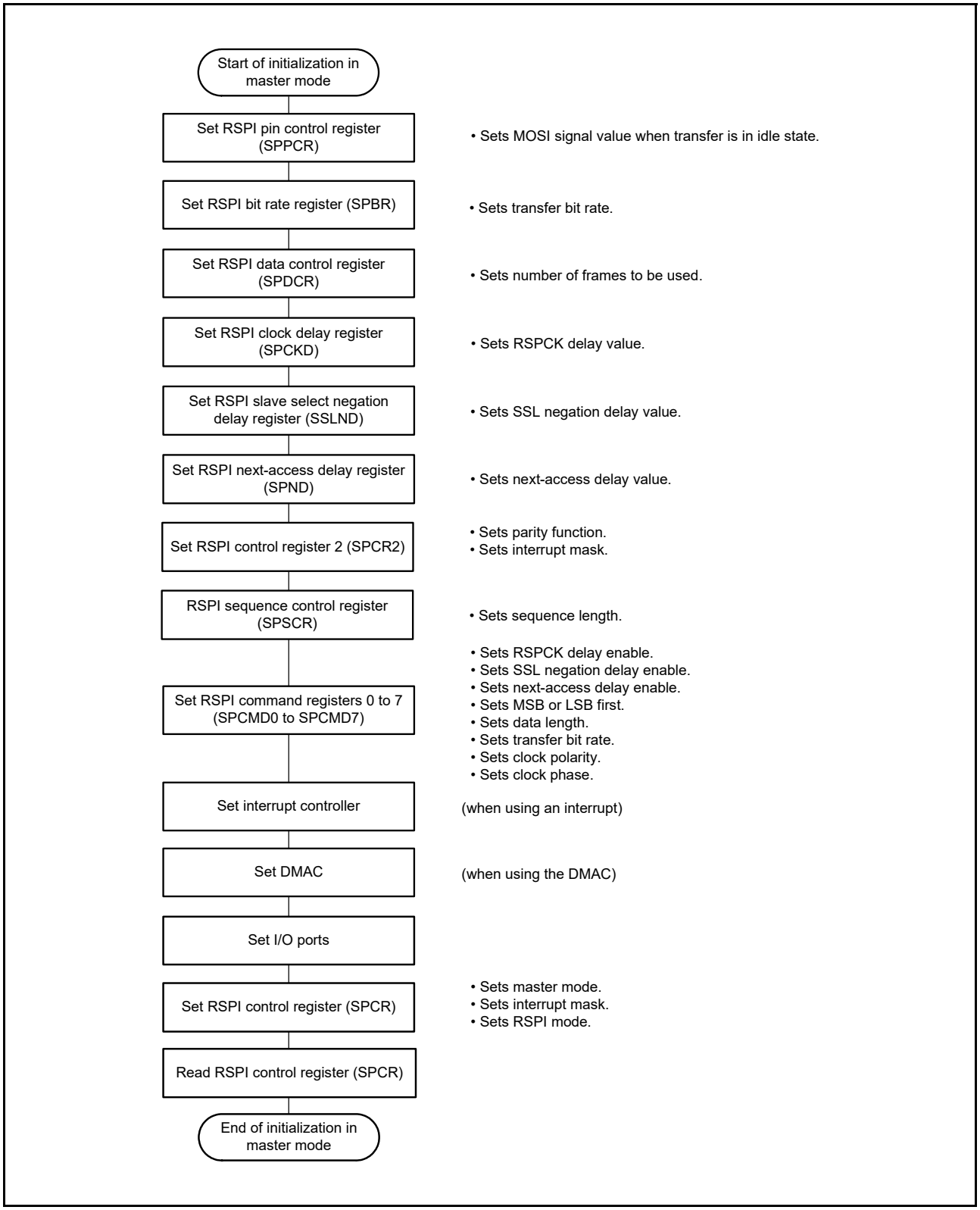


Figure 38.46 Example of Initialization Flowchart in Master Mode (Clock Synchronous Operation)

Downloaded from Arrow.com.

(5) Flow of Software Processing

Software processing during clock-synchronous master operation is the same as that for SPI master operation. For details, refer to section 38.3.10.1, (9) Software Processing Flow. Note that mode fault errors will not occur.

38.3.11.2 Slave Mode Operation

(1) Starting a Serial Transfer

When the SPCR.SPMS bit is 1, the first RSPCKA edge triggers the start of a serial transfer in the RSPI.
When detecting the start of a serial transfer in a condition in which the shift register is empty, the RSPI changes the status of the shift register to “full”, so that data cannot be copied from the transmit buffer to the shift register when serial transfer is in progress. If the shift register was full before the serial transfer started, the RSPI keeps the status of the shift register unchanged, in the full state.
When the SPMS bit is 1, the RSPI drives the MISOA output signal.
For details on the RSPI transfer format, refer to section 38.3.5, Transfer Format.
It should be noted that the SSLA0 input signal is not used in clock synchronous operation.

(2) Terminating a Serial Transfer

The RSPI terminates the serial transfer after detecting an RSPCKA edge corresponding to the final sampling timing. When free space is available in the receive buffer (the SPRF flag is 0), upon termination of serial transfer the RSPI copies received data from the shift register to the receive buffer of the SPDR register. Upon termination of a serial transfer the RSPI changes the status of the shift register to “empty” regardless of the receive buffer status. The final sampling timing changes depending on the bit length of transfer data. In slave mode, the RSPI data length depends on the SPCMD0.SPB[3:0] bit setting.
For details on the RSPI transfer format, refer to section 38.3.5, Transfer Format.

Downloaded from Arrow.com.

(3) Initialization Flowchart

Figure 38.47 is a flowchart illustrating an example of initialization in clock synchronous operation when the RSPI is used in slave mode. For a description of how to set up the interrupt controller, DMAC, and I/O ports, refer to the descriptions given in the individual blocks.

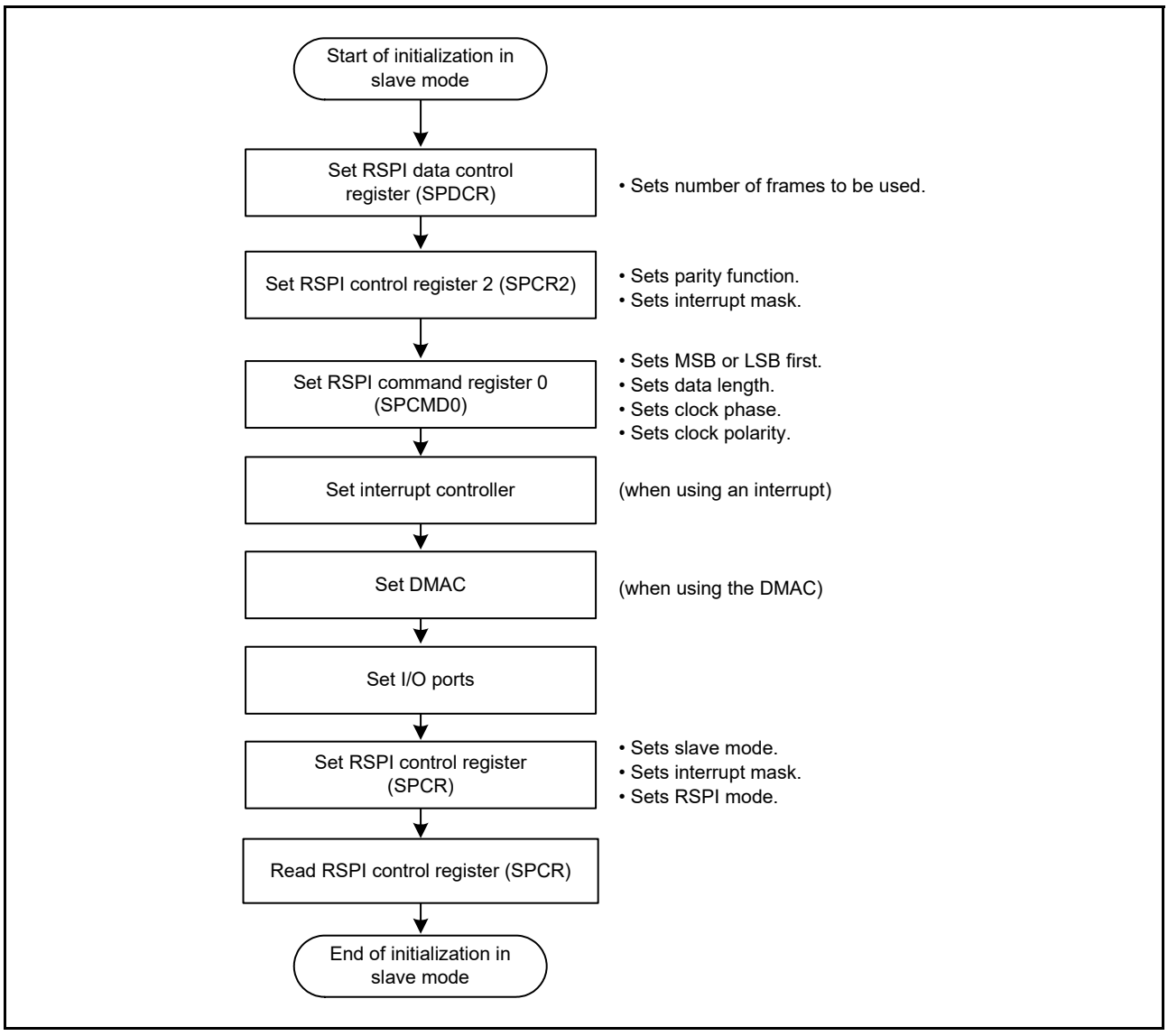


Figure 38.47 Example of Initialization Flowchart in Slave Mode (Clock Synchronous Operation)

(4) Flow of Software Processing

Software processing during clock-synchronous slave operation is the same as that for SPI slave operation. For details, refer to section 38.3.10.2, (6) Software Processing Flow. Note that mode fault errors will not occur.

Downloaded from Arrow.com.

38.3.12 Loopback Mode

When 1 is written to the SPPCR.SPLP2 bit or SPPCR.SPLP bit, the RSPI shuts off the path between the MISOA pin and the shift register if the SPCR.MSTR bit is 1, and between the MOSIA pin and the shift register if the SPCR.MSTR bit is 0, and connects the input path and output path of the shift register. The RSPI does not shut off the path between the MOSIA pin and the shift register if the SPCR.MSTR bit is 1, and between the MISOA pin and the shift register if the SPCR.MSTR bit is 0. This is called loopback mode. When a serial transfer is executed in loopback mode, the transmit data for the RSPI or the reversed transmit data becomes the received data for the RSPI.
Table 38.12 lists the relationship among the SPLP2 and SPLP bits and the received data. Figure 38.48 shows the configuration of the shift register I/O paths for the case where the RSPI in master mode is set in loopback mode (SPPCR.SPLP2 = 0, SPPCR.SPLP = 1).

Table 38.12 SPLP2 and SPLP Bit Settings and Received Data

SPPCR.SPLP2 Bit	SPPCR.SPLP Bit	Received Data
0	0	Input data from the MOSIA pin or MISOA pin
0	1	Inverted transmit data
1	0	Transmit data
1	1	Transmit data

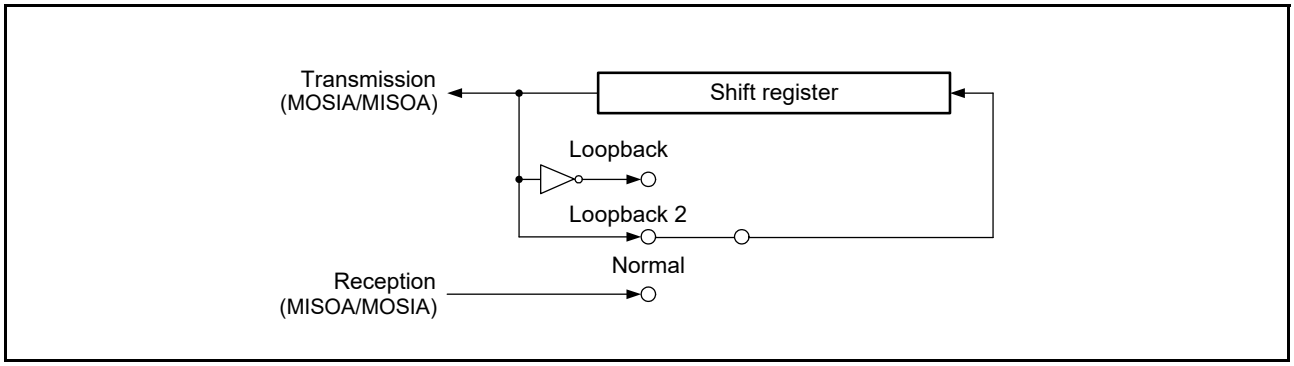


Figure 38.48 Configuration of Shift Register I/O Paths in Loopback Mode (Master Mode)

Downloaded from Arrow.com.

38.3.13 Self-Diagnosis of Parity Bit Function

The parity circuit consists of a parity bit adding unit used for transmit data and an error detecting unit used for received data. In order to detect defects in the parity bit adding unit and error detecting unit of the parity circuit, self-diagnosis is executed for the parity circuit following the flowchart shown in Figure 38.49.

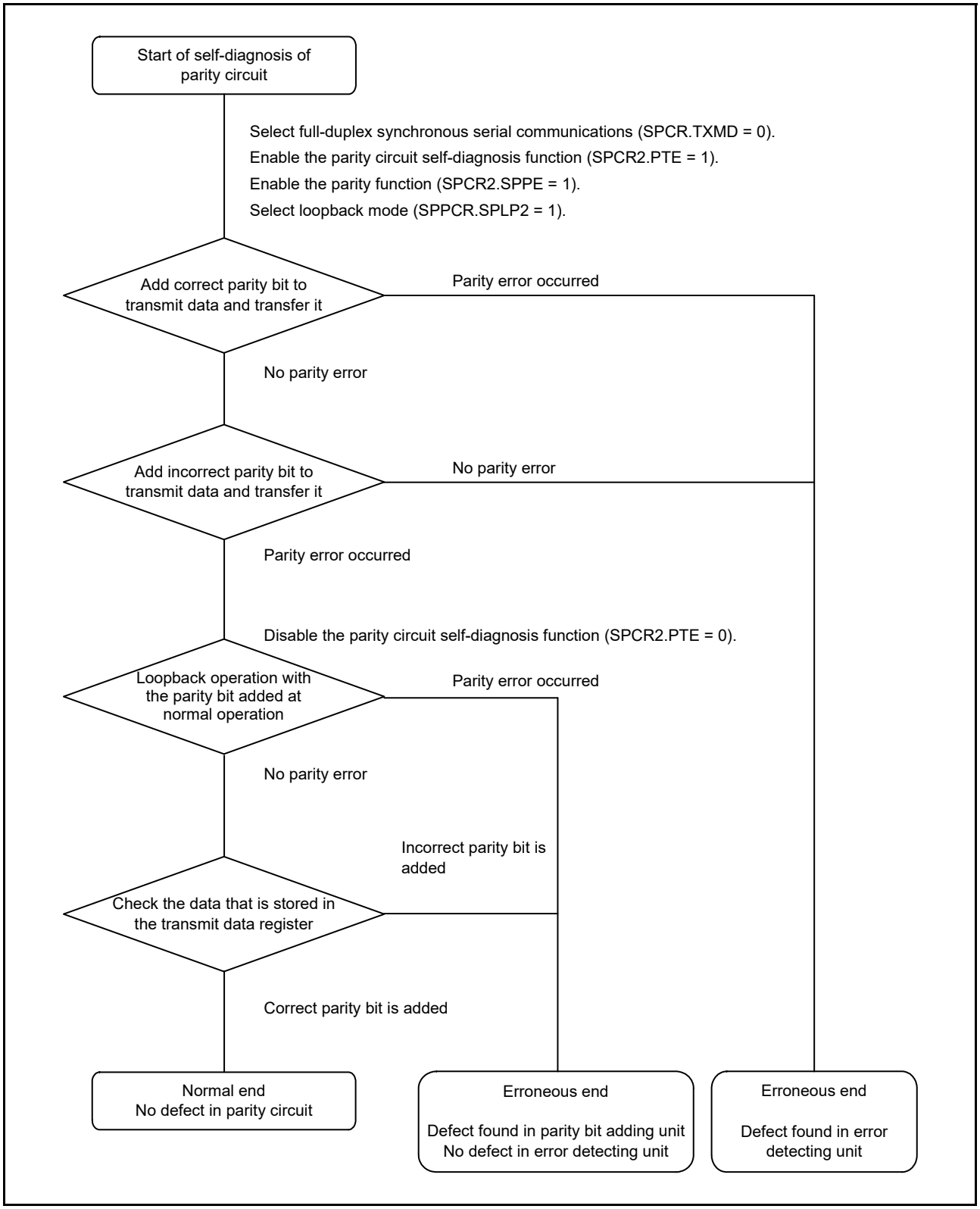


Figure 38.49 Flowchart for Self-Diagnosis of Parity Circuit

Downloaded from Arrow.com.

38.3.14 Interrupt Sources

The RSPI has interrupt sources of receive buffer full, transmit buffer empty, mode fault, overrun, parity error, and RSPI idle. In addition, the DTC or DMAC can be activated by the receive buffer full or transmit buffer empty interrupt to perform data transfer.

Since the vector address for SPEI is allocated to interrupt requests due to mode fault, overrun, and parity errors, the actual interrupt source must be determined from the flags. Interrupt sources for the RSPI are listed in Table 38.13. An interrupt is generated on satisfaction of an interrupt condition in Table 38.13. Clear the receive buffer full and transmit buffer empty sources through data transfer.

When using the DTC or DMAC to perform data transmission/reception, the DTC or DMAC must be set up first to be in a status in which transfer is enabled before making the RSPI settings. For the method for setting the DTC or DMAC, refer to section 18, DMA Controller (DMACA), or section 19, Data Transfer Controller (DTCa).

If the conditions for generating a transmit buffer empty or receive buffer full interrupt are generated while the ICU.IRn.IR flag is 1, the interrupt is not output as a request for ICU but is retained internally (the capacity for retention is one request per source). A retained interrupt request is output when the ICU.IRn.IR flag becomes 0. A retained interrupt request is automatically discarded once it is output as an actual interrupt request. The interrupt enable bit (the SPCR.SPTIE or SPCR.SPRIE bit) for an internally retained interrupt request can also be cleared to 0.

Table 38.13 Interrupt Sources of RSPI

Interrupt Source	Symbol	Interrupt Condition	DMAC/DTC Activation
Receive buffer full	SPRI	The receive buffer becomes full (the SPRF flag becomes 1) while the SPCR.SPRIE bit is 1.	Possible
Transmit buffer empty	SPTI	The transmit buffer becomes empty (the SPTEF flag becomes 1) while the SPCR.SPTIE bit is 1.	Possible
RSPI errors (mode fault, overrun and parity error)	SPEI	The SPSR.MODF, OVRF, or PERF flag is set to 1 while the SPCR.SPEIE bit is 1.	Impossible
RSPI idle	SPII	The SPSR.IDLNF flag is set to 0 while the SPCR2.SPIIE bit is 1.	Impossible

Downloaded from Arrow.com.

38.4 Link Operation by Event Linking

The RSPI0 supports the following event output for the event link controller (ELC). The event link output signal is output regardless of the interrupt enable bit setting.

38.4.1 Receive Buffer Full Event Output

This event signal is output when received data have been transferred from the shift register to the SPDR on completion of serial transfer.

38.4.2 Transmit Buffer Empty Event Output

This event signal is output when data for transmission have been transferred from the transmit buffer to the shift register and when the value of the SPE bit has changed from 0 to 1.

38.4.3 Mode Fault, Overrun, or Parity Error Event Output

(1) Mode Fault

Table 38.14 lists the occurrence conditions of a mode fault event.

Table 38.14 Occurrence Conditions of Mode Fault Event

	SPCR.MODFEN Bit	SSLA0 Pin	Remarks
Master (SPCR.MSTR bit = 1)	1	Active	Under the condition (the SPCR.MSTR bit is 1 and the MODFEN bit is 1), if the SPCR.SPMS bit is 0, mode fault error, overrun error, and parity error event output cannot be used. Do not set the ELSRn register to 52h.
Slave (SPCR.MSTR bit = 0)	1	Not active	Event is output only when the pin is deactivated during transmission.

(2) Overrun

The condition for this event signal being output in response to an overrun is completion of serial transfer while the receive buffer contains data that have not been read and the value of the SPCR.TXMD bit is 0, in which case the OVRF flag is set to 1.

(3) Parity Error

The condition for this event signal being output in response to a parity error is detection of a parity error on completion of serial transfer while the value of the TXMD bit in SPCR is 0 and the value of the SPPE bit in SPCR2 is 1.

Downloaded from Arrow.com.

38.4.4 RSPI Idle Event Output

(1) In Master Mode

In master mode, an event is output when the condition for setting the IDLNF flag (RSPI idle flag) to 0 is satisfied.

(2) In Slave Mode

In slave mode, an event is output when the SPCR.SPE bit is set to 0 (RSPI is initialized).

38.4.5 Transmission-Completed Event Output

During both SPI operation and clock synchronous operation in master mode, an event is output under the condition for setting the IDLNF flag (RSPI idle flag) from 1 to 0.

Table 38.15 Conditions for Generation of a Transmission-Completed Event (Slave)

	Transmit Buffer State	Shift Register State	Others
SPI operation (SPMS = 0)	Empty	Empty	Negation of SSLA0 input
Clock synchronous operation (SPMS = 1)	Empty	Empty	Edge detection of the last RSPCKA

Whether the operation is in master mode or slave mode, an event is not output if 0 is written to the SPCR.SPE bit in transmission or the SPCR.SPE bit is cleared by the mode fault error.

Downloaded from Arrow.com.

38.5 Usage Notes

38.5.1 Setting Module Stop Function

Module stop control register B (MSTPCRB) can be used to enable or disable the RSPI. Immediately after a reset, operation of the RSPI is disabled. Register access is enabled by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

38.5.2 Note on Low Power Consumption Functions

When using the module stop function and entering a low power consumption mode other than sleep mode, set the SPCR.SPE bit to 0 before completing communication.

38.5.3 Notes on Starting Transfer

If the ICU.IRn.IR flag is 1 at the time transfer is to be started, an interrupt request is internally retained after transfer starts, and this can lead to unanticipated behavior of the ICU.IRn.IR flag.
When the ICU.IRn.IR flag is 1 at the time transfer is to start, follow the procedure below to clear interrupt requests before enabling operations (by setting the SPCR.SPE bit to 1).

1. Confirm that transfer has stopped (i.e. that the SPCR.SPE bit is 0).
2. Set the relevant interrupt enable bit (the SPCR.SPTIE or SPCR.SPRIE bit) to 0.
3. Read the relevant interrupt enable bit (the SPCR.SPTIE or SPCR.SPRIE bit) and confirm that its value is 0.
4. Set the ICU.IRn.IR flag to 0.

38.5.4 Notes on the SPRF and SPTEF flags

When polling the SPSR.SPRF flag and/or SPSR.SPTEF flag, set the SPCR.SPRIE bit and/or SPCR.SPTIE bit to 0.

Downloaded from Arrow.com.

39. CRC Calculator (CRC)

The CRC (Cyclic Redundancy Check) calculator generates CRC codes.

39.1 Overview

Table 39.1 lists the specifications of the CRC calculator, and Figure 39.1 shows a block diagram of the CRC calculator.

Table 39.1 CRC Specifications

Item	Description
Data for CRC calculation*1	CRC codes are generated for any desired data in 8n-bit units (where n is a whole number)
CRC processor unit	8-bit parallel processing
CRC generating polynomial	One of three generating polynomials is selectable • 8-bit CRC $X^8 + X^2 + X + 1$ • 16-bit CRC $X^{16} + X^{15} + X^2 + 1$ $X^{16} + X^{12} + X^5 + 1$
CRC calculation switching	The bit order of CRC calculation results can be switched for LSB first or MSB first communication
Low power consumption function	Module stop state can be set.

Note 1. The circuit does not have a function to divide data for calculation into CRC calculation units. Write data in 8-bit units.

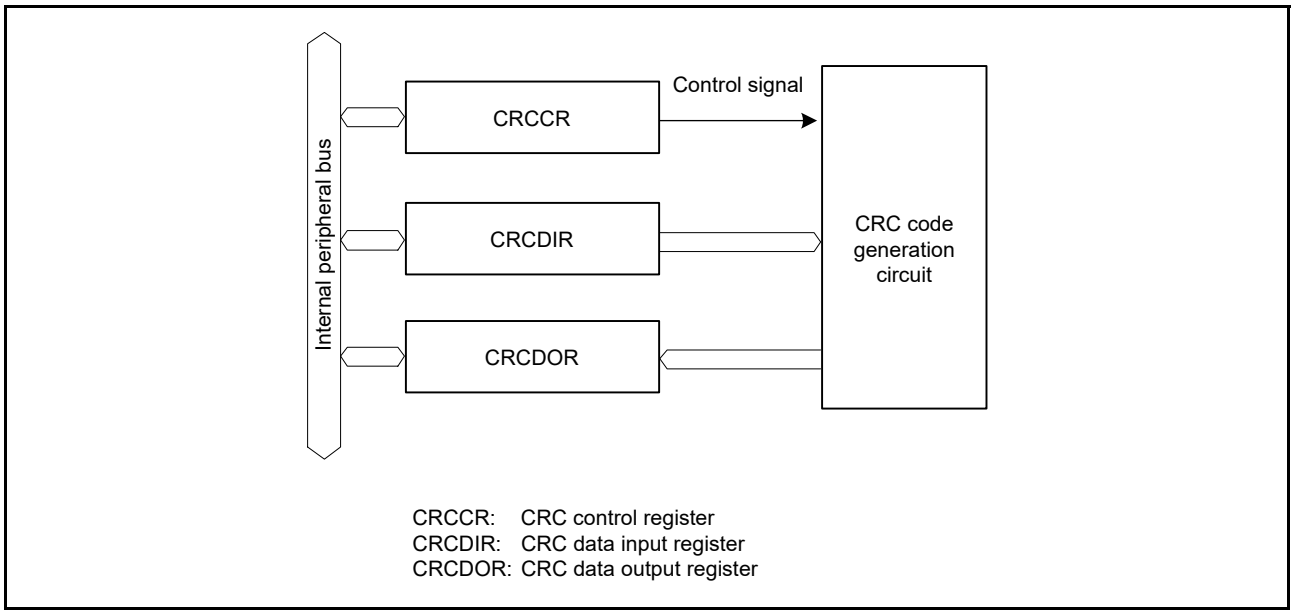


Figure 39.1 CRC Block Diagram

Downloaded from Arrow.com.

39.2 Register Descriptions

39.2.1 CRC Control Register (CRCCR)

Address(es): 0008 8280h

	b7	b6	b5	b4	b3	b2	b1	b0
	DORCLR	—	—	—	—	LMS	GPS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	GPS[1:0]	CRC Generating Polynomial Switching	b1 b0 0 0: No calculation is executed. 0 1: 8-bit CRC ($X^8 + X^2 + X + 1$) 1 0: 16-bit CRC ($X^{16} + X^{15} + X^2 + 1$) 1 1: 16-bit CRC ($X^{16} + X^{12} + X^5 + 1$)	R/W
b2	LMS	CRC Calculation Switching	0: Generates CRC for LSB first communication. 1: Generates CRC for MSB first communication.	R/W
b6 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	DORCLR	CRCDOR Register Clear	1: Clears the CRCDOR register. This bit is read as 0.	R/W*1

Note 1. Only 1 can be written.

LMS Bit (CRC Calculation Switching)

This bit selects the bit order of generated 16-bit CRC code. Transmit the lower-order byte (b7 to b0) of the CRC code first for LSB first communication and the higher-order byte (b15 to b8) first for MSB first communication. For details on the transmission and reception of CRC codes, refer to section 39.3, Operation.

DORCLR Bit (CRCDOR Register Clear)

Write 1 to this bit so that the CRCDOR register is set to 0000h.
This bit is read as 0. Only 1 can be written.

39.2.2 CRC Data Input Register (CRCDIR)

Address(es): 0008 8281h

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0

CRCDIR is a readable and writable register. Write data for CRC calculation to this register.

Downloaded from Arrow.com.

39.2.3 CRC Data Output Register (CRCDOR)

Address(es): 0008 8282h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CRCDOR is a readable and writable register.
Since its initial value is 0000h, rewrite the CRCDOR register to perform calculation using a value other than the initial value.
Data written to the CRCDIR register is CRC calculated and the result is stored in the CRCDOR register. If the CRC code is calculated following the transferred data and the result is 0000h, there is no CRC error.
When an 8-bit CRC ($X^8 + X^2 + X + 1$ polynomial) is in use, the valid CRC code is obtained in the low-order byte (b7 to b0). The high-order byte (b15 to b8) is not updated.

Downloaded from Arrow.com.

39.3 Operation

The CRC calculator generates CRC codes for use in LSB first or MSB first transfer.
The following shows examples of generating the CRC code for input data (F0h) using the 16-bit CRC generating polynomial ($X^{16} + X^{12} + X^5 + 1$). In these examples, the value of the CRC data output register (CRCDOR) is cleared before CRC calculation.
When an 8-bit CRC (with the polynomial $X^8 + X^2 + X + 1$) is in use, the valid bits of the CRC code are obtained in the lower-order byte of the CRCDOR register.

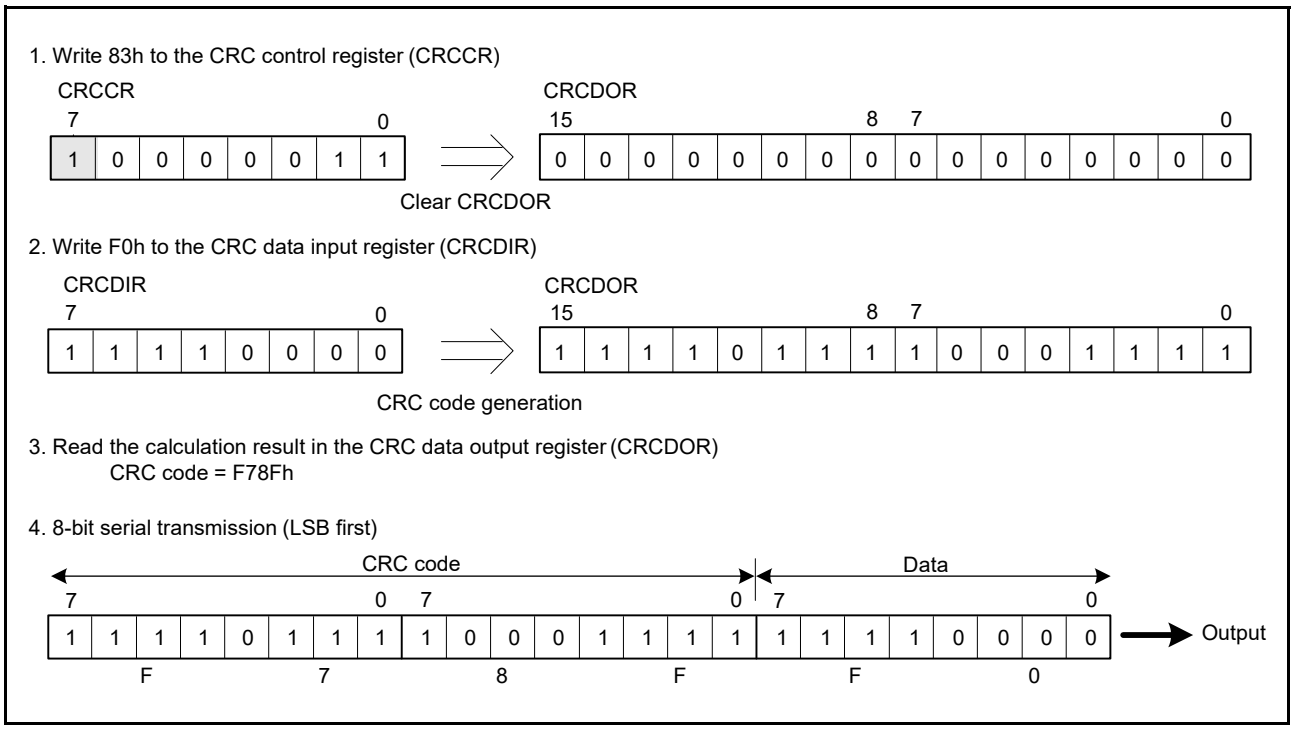


Figure 39.2 LSB First Data Transmission

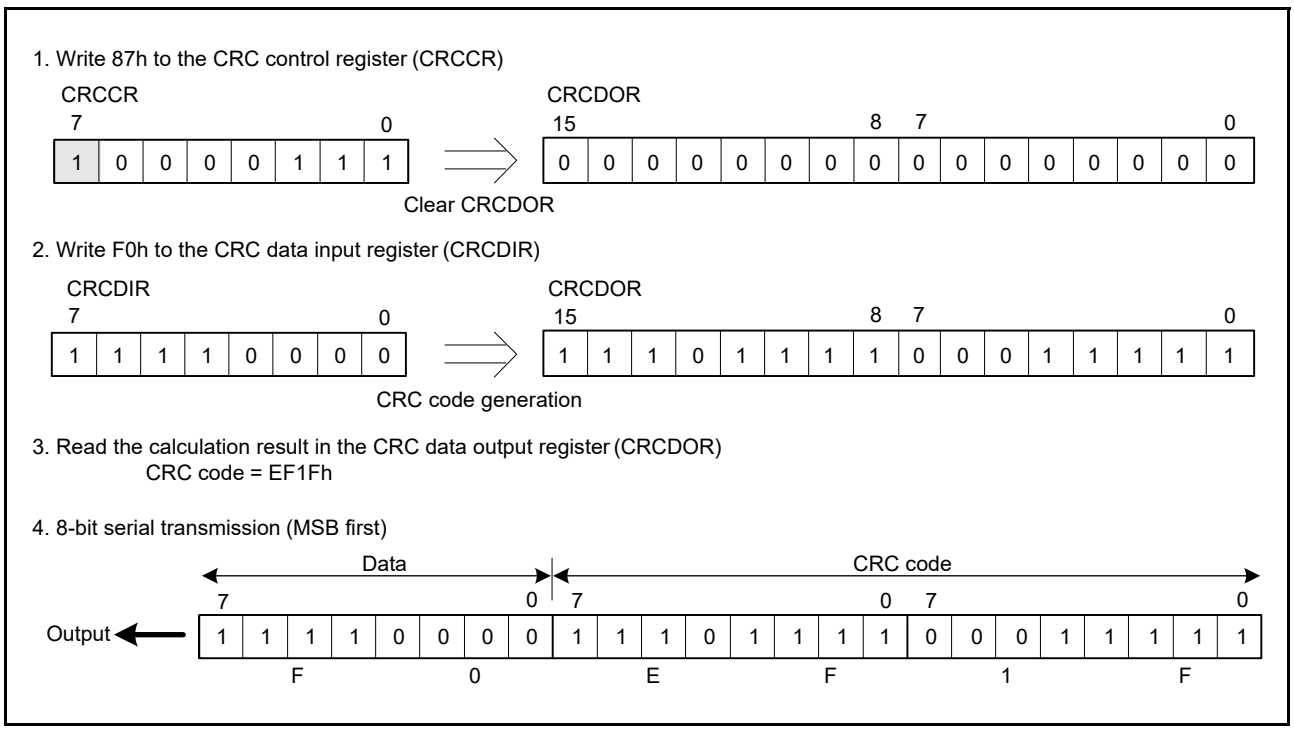


Figure 39.3 MSB First Data Transmission

Downloaded from Arrow.com.

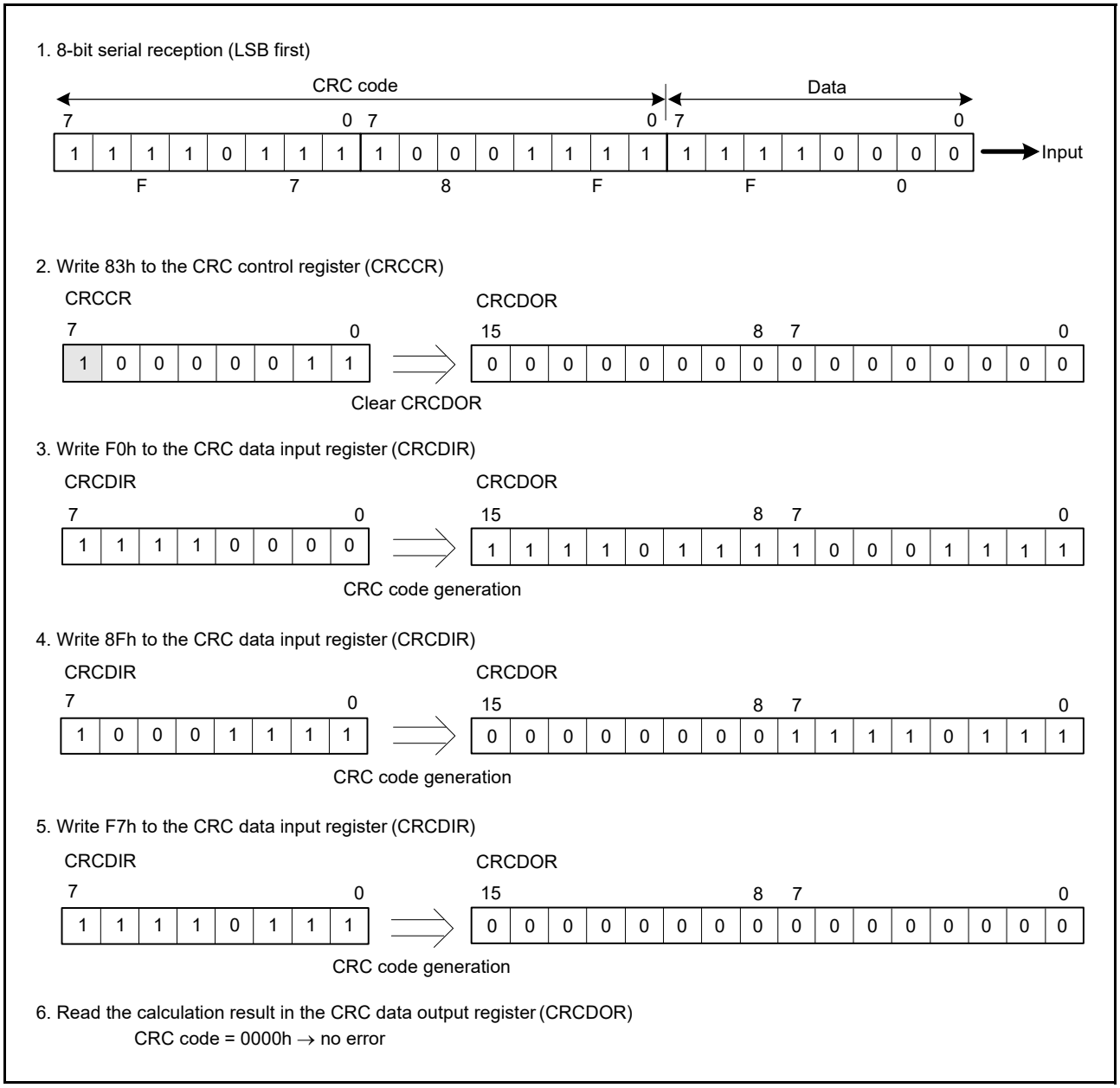


Figure 39.4 LSB First Data Reception

Downloaded from Arrow.com.

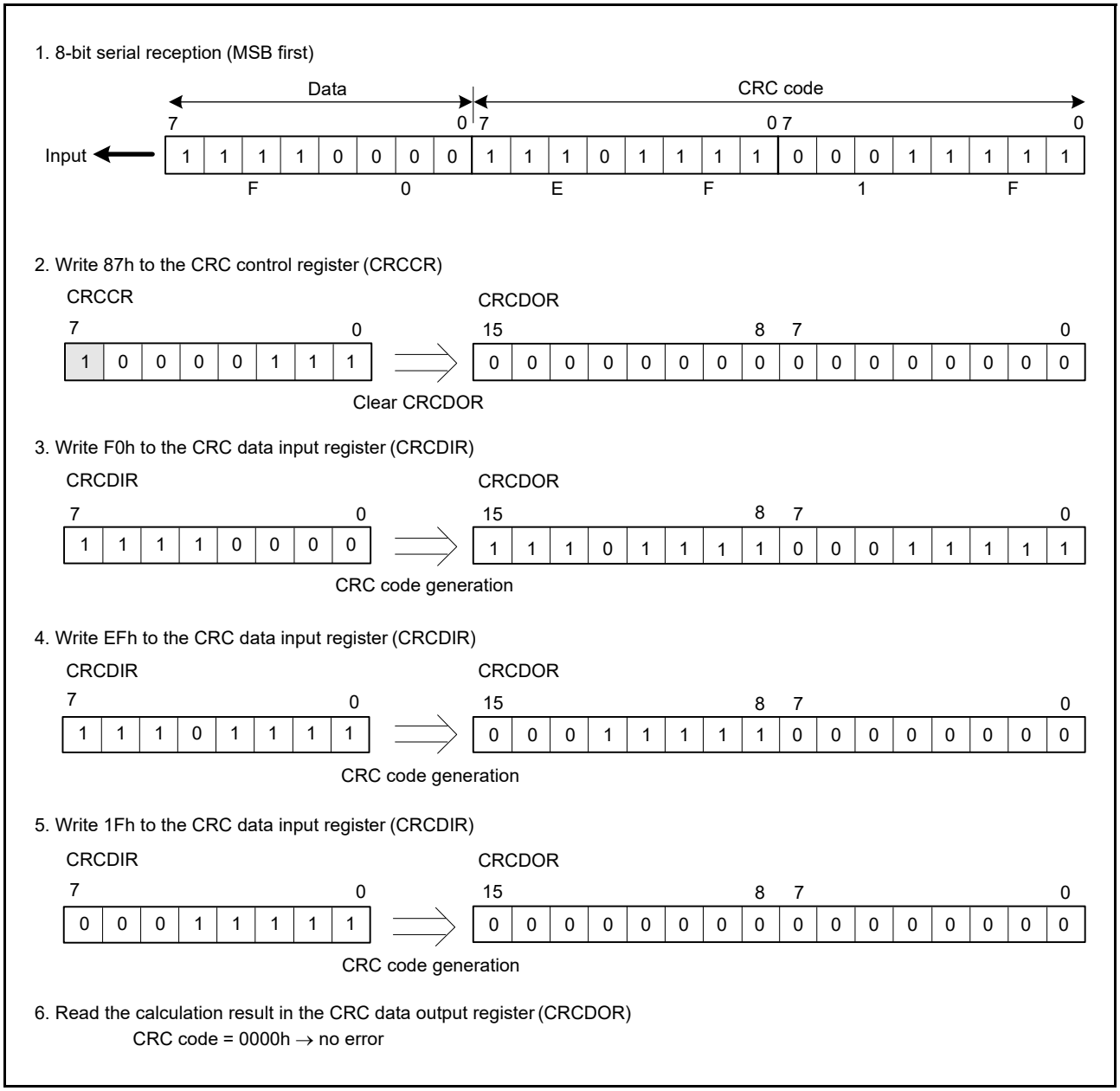


Figure 39.5 MSB First Data Reception

Downloaded from Arrow.com.

39.4 Usage Notes

39.4.1 Module Stop Function Setting

Operation of the CRC calculator can be disabled or enabled using the module stop control register B (MSTPCRB). After a reset, the CRC is in the module stop state. Register access is enabled by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

39.4.2 Note on Transmission

Note that the sequence of transmission for the CRC code differs according to whether transmission is LSB first or MSB first.

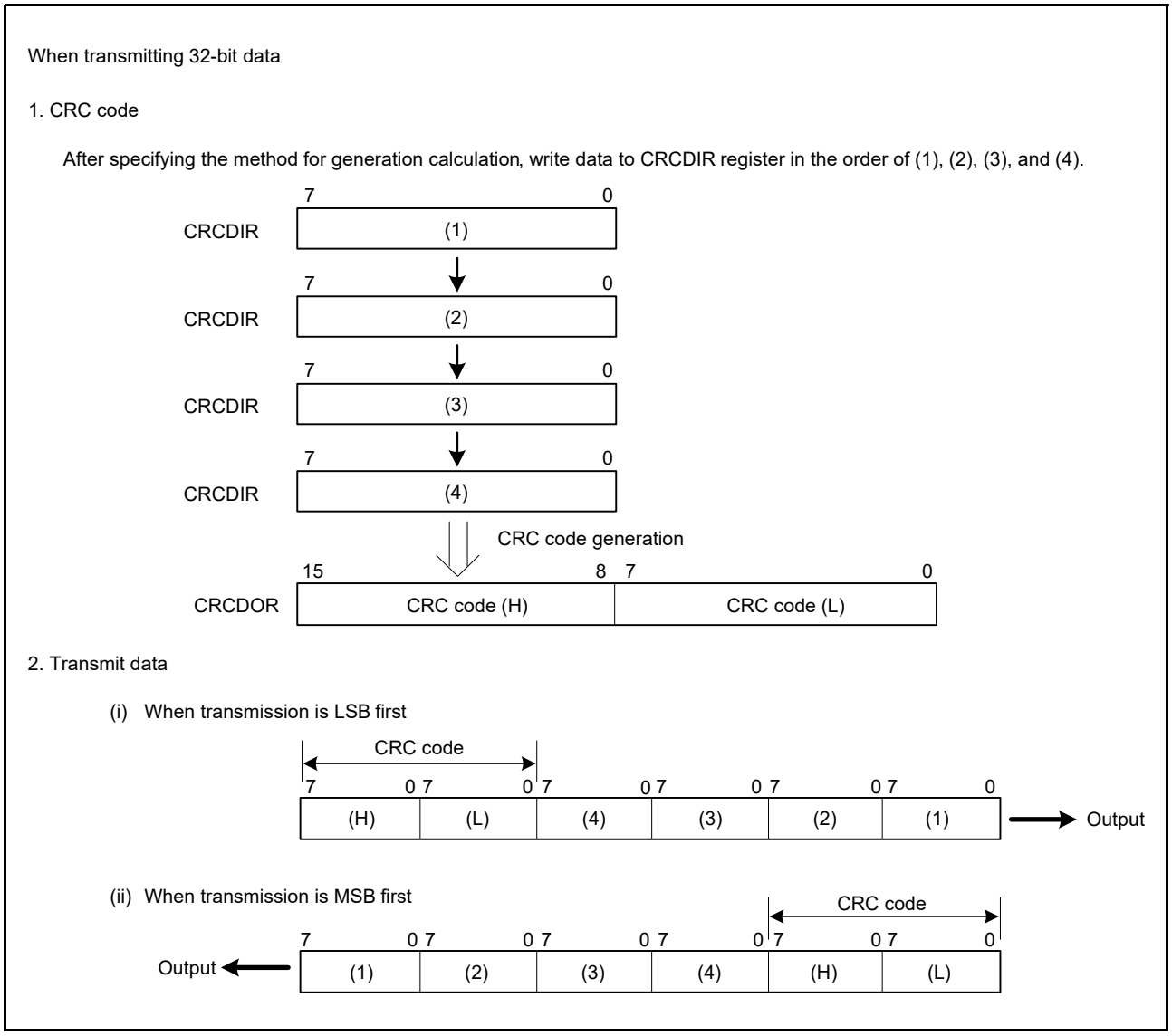


Figure 39.6 LSB First and MSB First Data Transmission

Downloaded from Arrow.com.

40. SD Host Interface (SDHIa)

This MCU incorporates an SD host interface (SDHI) which is compliant with the SD Specifications. When developing host devices that are compliant with the SD Specifications, the user must enter into the SD Host/Ancillary Product License Agreement (SD HALA).

40.1 Overview

Table 40.1 lists the SDHI specifications.

Table 40.1 SDHI Specifications

Item	Description
SD bus interface	• Compatible with SD memory card and SDIO card (**NOT** compatible with the SPI bus interface, embedded SDIO shared bus, 8-bit SD bus, or SDIO suspend/resume functions) • Transfer bus mode selectable from 4-bit wide bus mode or 1-bit default bus mode • Compatible with SD, SDHC, and SDXC formats
Transfer modes	Supports default speed mode
SDHI clock	The SDHI clock is generated by dividing peripheral module clock B (PCLKB) by *n*, where *n* = 1, 2, 4, 8, 16, 32, 64, 128, 256, or 512
Error check functions	• CRC7 (command/response) • CRC16 (transfer data)
Interrupt sources	• Card access interrupt (CACI) • SDIO access interrupt (SDACI) • Card detection interrupt (CDETI) • SD buffer access interrupt (SBFAI)
DMA transfer sources	• DMAC and DTC triggerable by the SBFAI interrupt • SD buffer is read and write accessible using the DMAC and DTC
Other functions	• Card detection • Write protection

Figure 40.1 shows a block diagram of the SDHI.

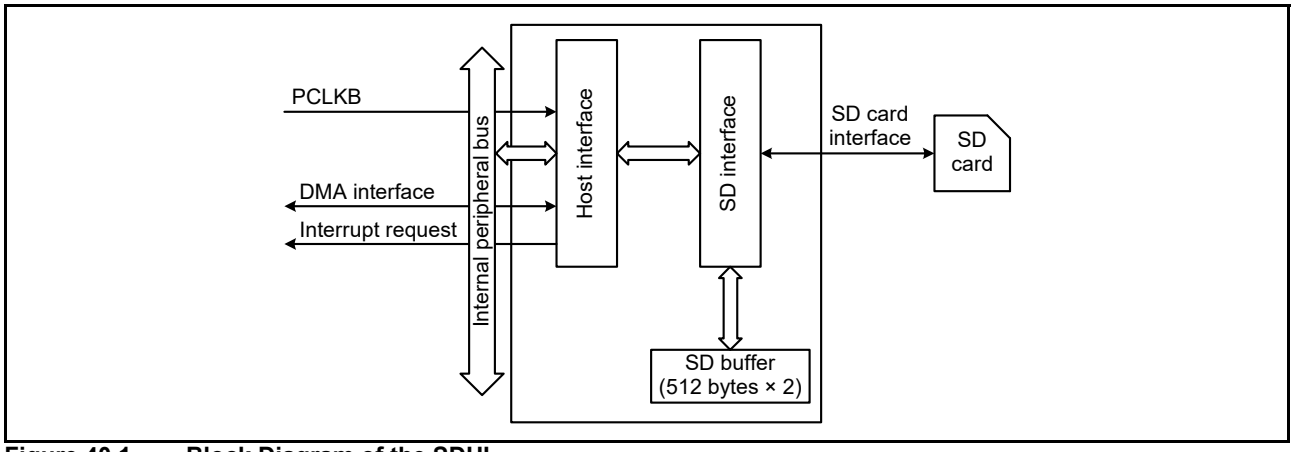


Figure 40.1 Block Diagram of the SDHI

Table 40.2 lists the pin configuration of the SDHI.

Table 40.2 Pin Configuration of the SDHI

Pin Name	I/O	Description
SDHI_CLK	Output	SDHI clock
SDHI_CMD	I/O	Command output, response input
SDHI_D0	I/O	Data 0 (DAT0)
SDHI_D1	I/O	Data 1 (DAT1), SDIO access interrupt
SDHI_D2	I/O	Data 2 (DAT2), read wait
SDHI_D3	I/O	Data 3 (DAT3), SD card detection
SDHI_CD	Input	SD card detection
SDHI_WP	Input	SD card write protection

Downloaded from Arrow.com.

40.2 Register Details

40.2.1 Command Register (SDCMD)

Address(es): SDHI.SDCMD 0008 AC00h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CMD12AT[1:0]		TRSTP	CMDRW	CMDTP	RSPTP[2:0]			ACMD[1:0]		CMDIDX[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	CMDIDX[5:0]	Command Index Field Value Select	These bits configure the command index field value. The examples below include the bit values for the ACMD[1:0] bits. b7 b0 0 0 0 0 0 1 1 0: CMD6 0 0 0 1 0 0 1 0: CMD18 0 1 0 0 1 1 0 1: ACMD13	R/W
b7, b6	ACMD[1:0]	Command Type Select	b7 b6 0 0: CMD 0 1: ACMD Only set the values listed above.	R/W
b10 to b8	RSPTP[2:0]	Response Type Select *1	b10 b8 0 0 0: Normal mode. Depending on the command, the response type and transfer method are selected by setting the ACMD[1:0] bits and CMDIDX[5:0] bits. At this time, the values for b15 to b11 in this register are invalid. 0 1 1: Expansion mode and no response 1 0 0: Expansion mode and R1, R5, R6, or R7 response 1 0 1: Expansion mode and R1b response 1 1 0: Expansion mode and R2 response 1 1 1: Expansion mode and R3 or R4 response Only set the values listed above.	R/W
b11	CMDTP	Data Transfer Select *2	0: Command does not include data transfer (bc, bcr, or ac) 1: Command includes data transfer (adtc)	R/W
b12	CMDRW	Data Transfer Direction Select *3	0: Write data to the SD card 1: Read data from the SD card	R/W
b13	TRSTP	Block Transfer Select *3	0: Single block transferred 1: Multiple blocks transferred	R/W
b15, b14	CMD12AT[1:0]	CMD12 Automatic Issue Select *4	b15 b14 0 0: CMD12 is automatically issued during multi-block transfer 0 1: CMD12 is not automatically issued during multi-block transfer Only set the values listed above.	R/W
b31 to b16	—	Reserved	These bits are 0 when read and cannot be modified.	R

Note 1. Some commands cannot be used in normal mode. Refer to Table 40.3 and set the RSPTP[2:0] bits.
Note 2. The CMDTP bit is valid only when the RSPTP[2:0] bits are 011b, 100b, 101b, 110b, or 111b.
Note 3. Bits CMDRW and TRSTP are valid only when the RSPTP[2:0] bits are 011b, 100b, 101b, 110b, or 111b, and the CMDTP bit is 1.
Note 4. The CMD12AT[1:0] bits are valid only when the RSPTP[2:0] bits are 011b, 100b, 101b, 110b, or 111b, and the TRSTP bit is 1.

The command type and response type are set in the SDCMD register. The command type and transfer mode must be set when the RSPTP[2:0] bits are 011b, 100b, 101b, 110b, or 111b. The sequence starts when a value is written to this register. Refer to Table 40.3 for setting examples. Do not write to the SDCMD register when the SDSTS2.CBSY flag is 1.

Downloaded from Arrow.com.

Table 40.3 lists examples of SDCMD register settings.

Table 40.3 Examples of SDCMD Register Settings

Type	Command Symbol	SDCMD Register Setting	Remarks
CMD	CMD0	0000 0000h	
	CMD2	0000 0002h	
	CMD3	0000 0003h	
	CMD4	0000 0004h	
	CMD5	0000 0705h or 0000 0005h	
	CMD6	0000 1C06h or 0000 0006h	
	CMD7	0000 0007h	When the card is deselected, the SD card does not return a response, so the SDSTS2.RSPTO flag becomes 1.
	CMD8	0000 0408h or 0000 0008h	
	CMD9	0000 0009h	
	CMD10	0000 000Ah	
	CMD11	0000 040Bh or 0000 000Bh	
	CMD12	0000 000Ch	
	CMD13	0000 000Dh	
	CMD15	0000 000Fh	
	CMD16	0000 0010h	
	CMD17	0000 0011h	
	CMD18	0000 0012h	
	CMD20	0000 0514h or 0000 0014h	
	CMD24	0000 0018h	
	CMD25	0000 0019h	
	CMD27	0000 001Bh	
	CMD28	0000 001Ch	
	CMD29	0000 001Dh	
	CMD30	0000 001Eh	
	CMD32	0000 0020h	
	CMD33	0000 0021h	
	CMD38	0000 0026h	
	CMD42	0000 002Ah	
	CMD52	0000 0434h or 0000 0034h	
	CMD53	0000 1C35h	Single block read
		0000 0C35h	Single block write
		0000 7C35h	Multi-block read
		0000 6C35h	Multi-block write
		0000 0035h	The setting to the left can be used regardless of the transfer being single block or multi-block. However, the MSB in the SDARG register (RW flag) must be set to 0 when reading and 1 when writing.
	CMD55	0000 0037h	
	CMD56	0000 0038h	
ACMD	ACMD6	0000 0046h	
	ACMD13	0000 004Dh	
	ACMD22	0000 0056h	
	ACMD23	0000 0057h	
	ACMD41	0000 0069h	
	ACMD42	0000 006Ah	
	ACMD51	0000 0073h	

Downloaded from Arrow.com.

40.2.2 Argument Register (SDARG)

Address(es): SDHI.SDARG 0008 AC08h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The SDARG register is used for setting the argument field value. Set the SDARG register before setting the SDCMD register. The argument field value of the automatically issued CMD12 is 0000 0000h regardless of the SDARG register value.

40.2.3 Data Stop Register (SDSTOP)

Address(es): SDSTOP 0008 AC10h

	b31	b30	b29	b28	b27	b26	b25	b24
	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8
	—	—	—	—	—	—	—	SDBLKCNTEN
Value after reset:	0	0	0	0	0	0	0	0

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	STP
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	STP	Transfer Stop	Data transfer stops when this bit is set to 1.	R/W
b7 to b1	—	Reserved	These bits are 0 when read and cannot be modified.	R
b8	SDBLKCNTEN	Block Count Register Value Select *1	0: SDBLKCNT register value is invalid 1: SDBLKCNT register value is valid	R/W
b31 to b9	—	Reserved	These bits are 0 when read and cannot be modified.	R

Note 1. Do not rewrite this bit when the SDSTS2.CBSY flag is 1.

The SDSTOP register stops data transfer. During a multi-block transfer sequence, the SDBLKCNT register value (number of blocks to be transferred) can be set to valid or invalid by setting the SDSTOP register.

STP Bit (Transfer Stop)

When setting the STP bit to 1, set it after the SDSTS1.RSPEND flag becomes 1; when setting the STP bit to 0, set it after the SDSTS1.ACEND flag becomes 1. After a command sequence is complete, the SDHI does not issue CMD12 and the SDSTS1.ACEND flag does not become 1 even if the STP bit is set to 1. When the SDHI is in the busy state after receiving the R1b response, the SDHI does not issue CMD12 even if the STP bit is 1, and after the SDHI is released from

Downloaded from Arrow.com.

the busy state, the SDSTS1.ACEND flag becomes 1.

- Performing a multi-block transfer
  When the STP bit is set to 1, the SDHI issues CMD12, and the command sequence is stopped. The SD buffer can be accessed even after the STP bit is set to 1, but a buffer access error occurs and the SDSTS2.ILW flag or SDSTS2.ILR flag becomes 1. If the command sequence stops due to a communication error or a timeout, the SDHI does not issue CMD12.
- Performing a single block transfer
  When the STP bit is set to 1 during a single block write access, if there is no data in the SD buffer, the SDSTS1.ACEND flag becomes 1. If there is data in the SD buffer, after the SDHI is released from the busy state, the SDSTS1.ACEND flag becomes 1. When the STP bit is set to 1 during a single block read access, the SDSTS1.ACEND flag becomes 1. Also, CMD12 is not issued even if the STP bit is set to 1 during a single block read access or single block write access.

SDBLKCNTEN Bit (Block Count Register Value Select)

If the SDBLKCNTEN bit is 1 during a multi-block transfer sequence, the SDHI automatically issues CMD12. When the SDCMD.RSPTP[2:0] bits are set to 000b and CMD18 or CMD25 is issued, or if the SDCMD.RSPTP[2:0] bits are set to 011b, 100b, 101b, 110b, or 111b and the SDCMD.TRSTP bit is 1 (multiple blocks transferred), if the SDCMD.CMD12AT[1:0] bits are 00b (CMD12 is automatically issued during multi-block transfer), and the number of transfer blocks reaches the value set in the SDBLKCNT register, the SDHI automatically issues CMD12.
If the command sequence is stopped by a communication error or a timeout, CMD12 is not automatically issued.

40.2.4 Block Count Register (SDBLKCNT)

Address(es): SDHI.SDBLKCNT 0008 AC14h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

When performing a multi-block transfer, SDBLKCNT is a readable/writable register used to set the number of blocks to be transferred. For example, when the register value is 0000 0001h, 1 block is transferred; when the register value is 0000 FFFFh, 65,535 blocks are transferred; and when the register value is FFFF FFFFh, 4,294,967,295 blocks are transferred. Do not set this register to 0000 0000h. Do not rewrite the SDBLKCNT register when the SDSTS2.CBSY flag is 1.

Downloaded from Arrow.com.

40.2.5 Response Register 10 (SDRSP10), Response Register 32 (SDRSP32), Response Register 54 (SDRSP54), Response Register 76 (SDRSP76)

- SDRSP10, SDRSP32, SDRSP54

Address(es): SDHI.SDRSP10 0008 AC18h, SDHI.SDRSP32 0008 AC20h, SDHI.SDRSP54 0008 AC28h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

- SDRSP76

Address(es): SDHI.SDRSP76 0008 AC30h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—																								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b23 to b0	—	—	This register stores the response from the SD card.	R
b31 to b24	—	Reserved	These bits are 0 when read.	R

Registers SDRSP10, SDRSP32, SDRSP54, and SDRSP76 are read-only registers that store the response from the SD card. Depending on the type of response from the SD card, the SDHI divides and stores the response among the four registers.

Table 40.4 lists the correspondence between the response type and its storage destination.

Table 40.4 Correspondence Between the Response Type and Its Storage Destination

Response Type	SDRSP10 Register	SDRSP32 Register	SDRSP54 Register	SDRSP76 Register
R1	[39:8]	—	[39:8] *1	—
R1b	[39:8]	—	[39:8] *1	—
R2	[39:8]	[71:40]	[103:72]	[127:104]
R3	[39:8]	—	—	—
R4	[39:8]	—	—	—
R5	[39:8]	—	—	—
R6	[39:8]	—	—	—
R7	[39:8]	—	—	—

Note 1. The response for CMD18 and CMD25 is stored in registers SDRSP10 and SDRSP54. Therefore, even if the SDRSP10 register is overwritten with the response for the automatically transmitted CMD12, the response for CMD18 or CMD25 can be confirmed by reading the SDRSP54 register.

Downloaded from Arrow.com.

40.2.6 SD Status Register 1 (SDSTS1)

Address(es): SDHI.SDSTS1 0008 AC38h

	b31	b30	b29	b28	b27	b26	b25	b24
	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8
	—	—	—	—	—	SDD3MON	SDD3IN	SDD3RM
Value after reset:	0	0	0	0	0	x	0	0

	b7	b6	b5	b4	b3	b2	b1	b0
	SDWPMON	—	SDCDMON	SDCDIN	SDCDRM	ACEND	—	RSPEND
Value after reset:	x	0	x	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RSPEND	Response End Detection Flag	0: Response end is not detected 1: Response end is detected	R/(W) *1
b1	—	Reserved	This bit is 0 when read and cannot be modified.	R
b2	ACEND	Access End Detection Flag	0: Access end is not detected 1: Access end is detected	R/(W) *1
b3	SDCDRM	SDHI_CD Removal Flag	0: SD card removal not detected by the SDHI_CD pin 1: SD card removal detected by the SDHI_CD pin	R/(W) *1
b4	SDCDIN	SDHI_CD Insertion Flag	0: SD card insertion not detected by the SDHI_CD pin 1: SD card insertion detected by the SDHI_CD pin	R/(W) *1
b5	SDCDMON	SDHI_CD Pin Monitor Flag	0: SDHI_CD pin level is high *2 1: SDHI_CD pin level is low *2	R
b6	—	Reserved	This bit is 0 when read and cannot be modified.	R
b7	SDWPMON	SDHI_WP Pin Monitor Flag	0: SDHI_WP pin level is high 1: SDHI_WP pin level is low	R
b8	SDD3RM	SDHI_D3 Removal Flag	0: SD card removal not detected by the SDHI_D3 pin 1: SD card removal detected by the SDHI_D3 pin	R/(W) *1
b9	SDD3IN	SDHI_D3 Insertion Flag	0: SD card insertion not detected by the SDHI_D3 pin 1: SD card insertion detected by the SDHI_D3 pin	R/(W) *1
b10	SDD3MON	SDHI_D3 Pin Monitor Flag	0: SDHI_D3 pin level is low 1: SDHI_D3 pin level is high	R
b31 to b11	—	Reserved	These bits are 0 when read and cannot be modified.	R

Note 1. The flag does not change even if set to 1. Writing 0 changes the flag value to 0.
Note 2. The flag changes when the pin level continues for the period set in the SDOPT.CTOP[3:0] bits or longer.

The SDSTS1 register indicates the detection of a response end or access end for a command sequence. The SDSTS1 register also indicates the detection SD card insertion/removal, and indicates the write protection status.
During a multi-block transfer sequence, if CMD12 or CMD52 (SDIO abort) is issued, the ACEND flag becomes 1, but the RSPEND flag remains set to 0.
If the command sequence is stopped due to a communication error or timeout, the ACEND flag or RSPEND flag becomes 1.
After a reset is released, the SDD3MON flag, SDD3IN flag, and SDD3RM flag values are changed according to the status of the SDHI_D3 pin, and their values are changed when data is being transferred in wide bus mode.
Flags to be cleared should be set to 0; flags not being cleared should be set to 1.

Downloaded from Arrow.com.

RSPEND Flag (Response End Detection Flag)

— This flag becomes 1 under any of the following conditions:

- A response is received.
- A command that does not have a response is issued.
- After the R1b response is received, the SDHI is released from the busy state.
- During a multi-block transmission, after the SDIOMD.C52PUB bit is set to 1, the CMD52 response is received.
- A communication error or timeout causes the command sequence to abort.

— This flag becomes 0 under the following condition:

- The flag is set to 0.

Note: When a command is issued that is absent of data transfer, the RSPEND flag becomes 1 after the command sequence ends.

ACEND Flag (Access End Detection Flag)

— This flag becomes 1 under any of the following conditions:

- During a single block read sequence, the SD buffer read access is completed.
- During a multi-block read sequence, the last block is read from the SD buffer.
- During a multi-block read sequence, if CMD12 is automatically issued, data is read from the SD buffer, and the response for CMD12 is received.
- During a single block write sequence, after a CRC status token is received, the SDHI is released from the busy state.
- During a multi-block write sequence, after a CRC status token is received for the last block, the SDHI is released from the busy state.
- During a multi-block write sequence, when CMD12 is automatically issued, a response busy of the automatically issued CMD12 is received.
- During a multi-block read sequence, when CMD12 is automatically issued, after setting the SDSTOP.STP bit to 1, a response of the automatically issued CMD12 is received.
- During a multi-block write sequence, when CMD12 is automatically issued, after setting the SDSTOP.STP bit to 1, a response busy of the automatically issued CMD12 is received.
- During a multi-block read sequence, after the SDIOMD.IOABT bit is set to 1, the response for CMD52 is received.
- During a multi-block write sequence, after the SDIOMD.IOABT bit is set to 1, the response for CMD52 is received.
- A communication error or timeout causes the command sequence to abort.

— This flag becomes 0 under the following condition:

- The flag is set to 0.

Note: The ACEND flag becomes 1 after the command sequence ends.

SDCDRM Flag (SDHI_CD Removal Flag)

— This flag becomes 1 under the following condition:

- The SDHI_CD pin changes from low to high, and the high period is the period set in the SDOPT.CTOP[3:0] bits or longer.

— This flag becomes 0 under the following condition:

- The flag is set to 0.

SDCDIN Flag (SDHI_CD Insertion Flag)

— This flag becomes 1 under the following condition:

- The SDHI_CD pin changes from high to low, and the low period is the period specified in the SDOPT.CTOP[3:0] bits or longer.

— This flag becomes 0 under the following condition:

- The flag is set to 0.

Downloaded from Arrow.com.

SDD3RM Flag (SDHI_D3 Removal Flag)

— This flag becomes 1 under the following condition:

- The SDHI_D3 pin changes from high to low, and the low period is at least two PCLKB cycles.

— This flag becomes 0 under the following condition:

- The flag is set to 0.

SDD3IN Flag (SDHI_D3 Insertion Flag)

— This flag becomes 1 under the following condition:

- The SDHI_D3 pin changes from low to high, and the high period is at least two PCLKB cycles.

— This flag becomes 0 under the following condition:

- The flag is set to 0.

Downloaded from Arrow.com.

40.2.7 SD Status Register 2 (SDSTS2)

Address(es): SDHI.SDSTS2 0008 AC3Ch

	b31	b30	b29	b28	b27	b26	b25	b24
	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8
	ILA	CBSY	SDCLKCREN	—	—	—	BWE	BRE
Value after reset:	0	0	1	0	0	0	0	0

	b7	b6	b5	b4	b3	b2	b1	b0
	SDD0MON	RSPTO	ILR	ILW	DTO	ENDE	CRCE	CMDE
Value after reset:	x	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CMDE	Command Error Detection Flag	0: Command error not detected 1: Command error detected	R/(W) *1
b1	CRCE	CRC Error Detection Flag	0: CRC error not detected 1: CRC error detected	R/(W) *1
b2	ENDE	End Bit Error Detection Flag	0: End bit error not detected 1: End bit error detected	R/(W) *1
b3	DTO	Data Timeout Detection Flag	0: Data timeout not detected 1: Data timeout detected	R/(W) *1
b4	ILW	SDBUFR Illegal Write Access Detection Flag	0: Illegal write access to the SDBUFR register not detected 1: Illegal write access to the SDBUFR register detected	R/(W) *1
b5	ILR	SDBUFR Illegal Read Access Detection Flag	0: Illegal read access to the SDBUFR register not detected 1: Illegal read access to the SDBUFR register detected	R/(W) *1
b6	RSPTO	Response Timeout Detection Flag	0: Response timeout not detected 1: Response timeout detected	R/(W) *1
b7	SDD0MON	SDHI_D0 Pin Status Flag	0: SDHI_D0 pin is low 1: SDHI_D0 pin is high	R
b8	BRE	SDBUFR Read Enable Flag	0: Read access to the SDBUFR register disabled 1: Read access to the SDBUFR register enabled	R/(W) *1
b9	BWE	SDBUFR Write Enable Flag	0: Write access to the SDBUFR register disabled 1: Write access to the SDBUFR register enabled	R/(W) *1
b10	—	Reserved	This bit is 0 when read and cannot be modified.	R
b11	—	Reserved	This bit is 0 when read. Set it to 1 when writing.	R/W
b12	—	Reserved	This bit is 0 when read and cannot be modified.	R
b13	SDCLKCREN	SDCLKCR Write Enable Flag	0: SD bus (CMD and DAT lines) is busy, so write access to the SDCLKCR.CLKEN bit and CLKSEL[7:0] bits is disabled. 1: SD bus (CMD and DAT lines) is not busy, so write access to the SDCLKCR.CLKEN bit and CLKSEL[7:0] bits is enabled.	R
b14	CBSY	Command Sequence Status Flag	0: Command sequence completed 1: Command sequence in progress (busy)	R
b15	ILA	Illegal Access Error Detection Flag	0: Illegal access error not detected 1: Illegal access error detected	R/(W) *1
b31 to b16	—	Reserved	These bits are 0 when read and cannot be modified.	R

Note 1. The flag does not change even if set to 1. Writing 0 changes the flag value to 0.

Downloaded from Arrow.com.

The SDSTS2 register indicates the status of the SD buffer and the status of the SD card. Flags to be cleared should be set to 0; flags not being cleared should be set to 1.

CMDE Flag (Command Error Detection Flag)

The command sequence is stopped when a command error occurs. When the SDIOMD.C52PUB bit is set to 1 and CMD52 is automatically issued, if a communication error or response timeout occurs, the command sequence will not be completed. Perform the error processing shown in section 40.3.6.8 or section 40.3.6.9 and complete the command sequence.

— This flag becomes 1 under any of the following conditions:

- The command index field value for the command transmitted differs from the command index field value for the response received.
- The command index field value for the automatically issued CMD12 or CMD52 (which are the commands to stop transfer) differs from the command index field value for the response received.

— This flag becomes 0 under the following condition:

- The flag is set to 0.

CRCE Flag (CRC Error Detection Flag)

The command sequence is stopped when a CRC error occurs. When the SDIOMD.C52PUB bit is set to 1 and CMD52 is automatically issued, if a communication error or response timeout occurs, the command sequence will not be completed. Perform the error processing shown in section 40.3.6.8 or section 40.3.6.9 and complete the command sequence.

— This flag becomes 1 under any of the following conditions:

- The received CRC status token is in error (the value of the CRC status is a value other than 010b).
- The read data contains a CRC error.
- The response contains a CRC error.
- The response for the automatically issued CMD12 or CMD52 (which are the commands to stop transfer) contains a CRC error.

— This flag becomes 0 under the following condition:

- The flag is set to 0.

ENDE Flag (End Bit Error Detection Flag)

The command sequence is stopped when an end bit error occurs. When the SDIOMD.C52PUB bit is set to 1 and CMD52 is automatically issued, if a communication error or response timeout occurs, the command sequence will not be completed. Perform the error processing shown in section 40.3.6.8 or section 40.3.6.9 and complete the command sequence.

— This flag becomes 1 under any of the following conditions:

- The response length is in error (the end bit could not be detected).
- The read data length is in error (the end bit of the enabled bit could not be detected).
- The CRC status token length is in error (the end bit could not be detected).
- The response length for the automatically issued CMD12 or CMD52 (which are the commands to stop transfer) contains an error (the end bit could not be detected).

— This flag becomes 0 under the following condition:

- The flag is set to 0.

Downloaded from Arrow.com.

DTO Flag (Data Timeout Detection Flag)

This flag indicates that the data expected to be received during the period specified (set in the SDOPT.TOP[3:0] bits) was not received. However, response timeouts are excluded. The command sequence stops when a data timeout occurs.

— This flag becomes 1 under any of the following conditions:

- After the R1b response is received, the SDHI is busy for the period specified or longer.
- After the CRC status token is received, the SDHI is busy for the period specified or longer.
- After data is written, the CRC status token is not received even after the period specified elapsed.
- After a read command is issued, the read data is not received even after the period specified elapsed.
- After CMD12 is issued during a command sequence, the SDHI is busy for the period specified or longer.
- After the read data is received, the next read data is not received even after the period specified elapsed.
- After the SDHI exits the read wait state, the next read data is not received even after the period specified elapsed.

— This flag becomes 0 under the following condition:

- The flag is set to 0.

ILW Flag (SDBUFR Illegal Write Access Detection Flag)

— This flag becomes 1 under any of the following conditions:

- A value is written to the SDBUFR register while the SDHI is not in the data read or data write command state.
- A value is written to the SDBUFR register while the SD buffer is full.
- A value is written to the SDBUFR register while the CRC status token or CRC status token length is in error.
- After the CRC status token is received, a value is written to the SDBUFR register if the SDHI is busy for the period set in bits SDOPT.TOP[3:0] or longer.

— This flag becomes 0 under the following condition:

- The flag is set to 0.

ILR Flag (SDBUFR Illegal Read Access Detection Flag)

— This flag becomes 1 under any of the following conditions:

- The SDBUFR register is read while the SD buffer is empty.
- The value read from the SDBUFR register includes a CRC error or end bit error.

— This flag becomes 0 under the following condition:

- The flag is set to 0.

RSPTO Flag (Response Timeout Detection Flag)

The command sequence is stopped when a response timeout occurs. When the SDIOMD.C52PUB bit is set to 1 and CMD52 is automatically issued, if a communication error or response timeout occurs, the command sequence will not be completed. Perform the error processing shown in section 40.3.6.8 or section 40.3.6.9 and complete the command sequence.

— This flag becomes 1 under the following condition:

- A response is not received even after 640 SDHI clock cycles or more have elapsed (including the response for the automatically issued CMD12 or CMD52 (which are the commands to stop transfer)).

— This flag becomes 0 under the following condition:

- The flag is set to 0.

SDD0MON Flag (SDHI_D0 Pin Status Flag)

This flag indicates the status of the SDHI_D0 pin. After an erase command is issued, if the DTO flag is 1 and the RSPTO flag is 0, polling can be used to monitor the SDD0MON flag change from 0 to 1, and check that the erase command sequence is complete. If a communication error or timeout occurs during the write sequence, the SDHI_D0 pin may remain low. When the SDHI clock is stopped, the value before the SDHI clock was stopped is retained.

Downloaded from Arrow.com.

BRE Flag (SDBUFR Read Enable Flag)

— This flag becomes 1 under any of the following conditions:

- During a single block transfer, the data size set in the SDSIZE.LEN[9:0] bits is stored in the SD buffer.
- During a multi-block transfer, the data size set in the SDSIZE.LEN[9:0] bits is stored in one of the two SD buffers.

— This flag becomes 0 under any of the following conditions:

- The bit is set to 0.
- DMA transfer is used to read 1 block of data from the SD buffer.

If the CPU is used to read data from the SDBUFR register, set the BRE flag to 0 before reading the data size *1 set in the SDSIZE.LEN[9:0] bits. Even if the block of data read contains a CRC error or end bit error, the data is stored in the SD buffer and the BRE flag becomes 1.

Note 1. If the transfer data size set in the SDSIZE.LEN[9:0] bits is an odd number, the odd numbered byte is ignored. Refer to section 40.5.2, SDBUFR Register Illegal Write Error for details.

BWE Flag (SDBUFR Write Enable Flag)

— This flag becomes 1 under any of the following conditions:

- During a single block transfer, the SD buffer is empty.
- During a multi-block transfer, bank 1 or bank 2 of the SD buffer is empty.

— This flag becomes 0 under any of the following conditions:

- The flag is set to 0.
- DMA transfer is used to write 1 block of data to the SD buffer.

If the CPU is used to write data to the SDBUFR register, set the BWE flag to 0 before writing the data size *1 set in the SDSIZE.LEN[9:0] bits.

Note 1. If the transfer data size set in the SDSIZE.LEN[9:0] bits is an odd number, the odd numbered byte is ignored. Refer to section 40.5.2, SDBUFR Register Illegal Write Error for details.

SDCLKCREN Flag (SDCLKCR Write Enable Flag)

When a value is written to the SDCMD register, the SDHI starts the command sequence, the SDSTS2.CBSY flag becomes 1, and the SDSTS2.SDCLKCREN flag becomes 0. When the command sequence is complete, after the SDSTS2.CBSY flag becomes 0, eight cycles of the SDHI clock elapse and then the SDSTS2.SDCLKCREN flag becomes 1.

ILA Flag (Illegal Access Error Detection Flag)

— This flag becomes 1 under any of the following conditions:

- A value is written to the SDCMD register when the SDSTS2.CBSY flag is 1.
- The SDCMD.CMDTP bit is set to 1 (command accompanying data transfer), the SDCMD.ACMD[1:0] bits are set to 00b, and the SDCMD.CMDIDX[5:0] bits are set to 001100b (CMD12).

— This flag becomes 0 under the following condition:

- The flag is set to 0.

Downloaded from Arrow.com.

40.2.8 SD Interrupt Mask Register 1 (SDIMSK1)

Address(es): SDHI.SDIMSK1 0008 AC40h

	b31	b30	b29	b28	b27	b26	b25	b24
	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8
	—	—	—	—	—	—	SDD3INM	SDD3RMM
Value after reset:	0	0	0	0	0	0	1	1

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	SDCDINM	SDCDRMM	ACENDM	—	RSPENDM
Value after reset:	0	0	0	1	1	1	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	RSPENDM	Response End Interrupt Request Mask	0: Response end interrupt request is not masked 1: Response end interrupt request is masked	R/W
b1	—	Reserved	This bit is 0 when read and cannot be modified.	R
b2	ACENDM	Access End Interrupt Request Mask	0: Access end interrupt request is not masked 1: Access end interrupt request is masked	R/W
b3	SDCDRMM	SDHI_CD Removal Interrupt Request Mask	0: SD card removal interrupt request by the SDHI_CD pin not masked 1: SD card removal interrupt request by the SDHI_CD pin masked	R/W
b4	SDCDINM	SDHI_CD Insertion Interrupt Request Mask	0: SD card insertion interrupt request by the SDHI_CD pin not masked 1: SD card insertion interrupt request by the SDHI_CD pin masked	R/W
b7 to b5	—	Reserved	These bits are 0 when read and cannot be modified.	R
b8	SDD3RMM	SDHI_D3 Removal Interrupt Request Mask	0: SD card removal interrupt request by the SDHI_D3 pin not masked 1: SD card removal interrupt request by the SDHI_D3 pin masked	R/W
b9	SDD3INM	SDHI_D3 Insertion Interrupt Request Mask	0: SD card insertion interrupt request by the SDHI_D3 pin not masked 1: SD card insertion interrupt request by the SDHI_D3 pin masked	R/W
b31 to b10	—	Reserved	These bits are 0 when read and cannot be modified.	R

The SDIMSK1 register enables and disables the interrupt requests from the status flags in the SDSTS1 register. Refer to Table 40.8, Interrupt Sources for details on the relationship between the status flags and the requested interrupt source.

Downloaded from Arrow.com.

40.2.9 SD Interrupt Mask Register 2 (SDIMSK2)

Address(es): SDHI.SDIMSK2 0008 AC44h

	b31	b30	b29	b28	b27	b26	b25	b24
	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8
	ILAM	—	—	—	—	—	BWEM	BREM
Value after reset:	1	0	0	0	1	0	1	1

	b7	b6	b5	b4	b3	b2	b1	b0
	—	RSPTOM	ILRM	ILWM	DTTOM	ENDEM	CRCEM	CMDEM
Value after reset:	0	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b0	CMDEM	Command Error Interrupt Request Mask	0: Command error interrupt request not masked 1: Command error interrupt request masked	R/W
b1	CRCEM	CRC Error Interrupt Request Mask	0: CRC error interrupt request not masked 1: CRC error interrupt request masked	R/W
b2	ENDEM	End Bit Error Interrupt Request Mask	0: End bit detection error interrupt request not masked 1: End bit detection error interrupt request masked	R/W
b3	DTTOM	Data Timeout Interrupt Request Mask	0: Data timeout interrupt request not masked 1: Data timeout interrupt request masked	R/W
b4	ILWM	SDBUFR Register Illegal Write Interrupt Request Mask	0: Illegal write detection interrupt request for the SDBUFR register not masked 1: Illegal write detection interrupt request for the SDBUFR register masked	R/W
b5	ILRM	SDBUFR Register Illegal Read Interrupt Request Mask	0: Illegal read detection interrupt request for the SDBUFR register not masked 1: Illegal read detection interrupt request for the SDBUFR register masked	R/W
b6	RSPTOM	Response Timeout Interrupt Request Mask	0: Response timeout interrupt request not masked 1: Response timeout interrupt request masked	R/W
b7	—	Reserved	This bit is 0 when read and cannot be modified.	R
b8	BREM	BRE Interrupt Request Mask	0: Read enable interrupt request for the SDBUFR register not masked 1: Read enable interrupt request for the SDBUFR register masked	R/W
b9	BWEM	BWE Interrupt Request Mask	0: Write enable interrupt request for the SDBUFR register not masked 1: Write enable interrupt request for the SDBUFR register masked	R/W
b10	—	Reserved	This bit is 0 when read and cannot be modified.	R
b11	—	Reserved	This bit is 1 when read and cannot be modified.	R
b14 to b12	—	Reserved	These bits are 0 when read and cannot be modified.	R
b15	ILAM	Illegal Access Error Interrupt Request Mask	0: Illegal access error interrupt request not masked 1: Illegal access error interrupt request masked	R/W
b31 to b16	—	Reserved	These bits are 0 when read and cannot be modified.	R

Note 1. When the SDIMSK2.BWEM bit is 0 or the SDIMSK2.BREM bit is 0, set the SDDMAEN.DMAEN bit to 0. When the SDDMAEN.DMAEN bit is 1, set the SDIMSK2.BWEM bit to 1 and the SDIMSK2.BREM bit to 1.

The SDIMSK2 register enables and disables the interrupt requests from the status flags in the SDSTS2 register. Refer to Table 40.8 for details on the relationship between the status flags and the requested interrupt source.

Downloaded from Arrow.com.

40.2.10 SDHI Clock Control Register (SDCLKCR)

Address(es): SDHI.SDCLKCR 0008 AC48h

	b31	b30	b29	b28	b27	b26	b25	b24
	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8
	—	—	—	—	—	—	CLKCTRLEN	CLKEN
Value after reset:	0	0	0	0	0	0	0	0

	b7	b6	b5	b4	b3	b2	b1	b0
	CLKSEL[7:0]							
Value after reset:	0	0	1	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	CLKSEL[7:0]	SDHI Clock Frequency Select *1	b7 b0 0 0 0 0 0 0 0 0: PCLKB divided by 2 0 0 0 0 0 0 0 1: PCLKB divided by 4 0 0 0 0 0 0 1 0: PCLKB divided by 8 0 0 0 0 0 1 0 0: PCLKB divided by 16 0 0 0 0 1 0 0 0: PCLKB divided by 32 0 0 0 1 0 0 0 0: PCLKB divided by 64 0 0 1 0 0 0 0 0: PCLKB divided by 128 0 1 0 0 0 0 0 0: PCLKB divided by 256 1 0 0 0 0 0 0 0: PCLKB divided by 512 1 1 1 1 1 1 1 1: PCLKB *2 Only set the values listed above.	R/W
b8	CLKEN	SDHI Clock Output Control *1	0: SDHI clock output is disabled (SDHI_CLK signal fixed low) 1: SDHI clock output enabled	R/W
b9	CLKCTRLEN	SDHI Clock Output Automatic Control Select	0: Automatic control of SDHI clock output disabled 1: Automatic control of SDHI clock output enabled	R/W
b31 to b10	—	Reserved	These bits are 0 when read and cannot be modified.	R

Note 1. Bits CLKSEL[7:0] and CLKEN cannot be write accessed when the SDSTS2.SDCLKCREN flag is 0.
Note 2. When setting the CLKSEL[7:0] bits to 11111111b or when changing the CLKSEL[7:0] bit values from 11111111b to another value, perform the following steps:
(1) Set the CLKEN bit to 0. Do not change the other bit values.
(2) Change the CLKSEL[7:0] bit values. Do not change the other bit values.
(3) Set the CLKEN bit to 1. Do not change the other bit values.

The SDCLKCR register controls the SDHI clock frequency settings and output. Set the CLKEN bit to 1 before writing to the SDCMD register to start a command sequence. Do not write access the SDCLKCR register when the SDSTS2.SDCLKCREN flag is 0.

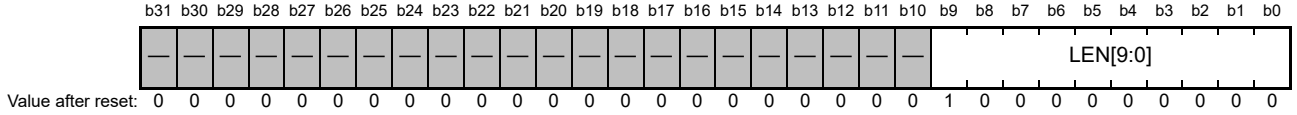
CLKCTRLEN Bit (SDHI Clock Output Automatic Control Select)

The SDHI clock output automatic control is a function for starting and stopping SDHI clock output only during a command sequence. When this function is enabled, the SDHI starts outputting the SDHI clock after a value is set to the SDCMD register. After the command sequence is complete and eight cycles of the SDHI clock elapse, the SDHI stops outputting the SDHI clock. When the SDCLKCR.CLKEN bit is 0, output from the SDHI_CLK pin becomes low regardless of the CLKCTRLEN bit setting.

Downloaded from Arrow.com.

40.2.11 Transfer Data Size Register (SDSIZE)

Address(es): SDHI.SDSIZE 0008 AC4Ch



Bit	Symbol	Bit Name	Description	R/W
b9 to b0	LEN[9:0]	Transfer Data Size Setting	Set the transfer data size. *1	R/W
b11 to b10	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R
b31 to b12	—	Reserved	These bits are 0 when read and cannot be modified.	R

Note 1. Do not rewrite these bits when the SDSTS2.CBSY flag is 1.

The SDSIZE register is used to set the transfer data size.

LEN[9:0] Bits (Transfer Data Size Setting)

When using single block transfer, the transfer data size can be set from 1 byte to 512 bytes. When CMD12 is automatically issued during a multi-block transfer sequence (CMD18 and CMD25), the transfer data size can only be set to 512 bytes. When CMD12 is not automatically issued during a multi-block transfer sequence, the transfer data size can be set to 32, 64, 128, 256, or 512 bytes. However, a 32-, 64-, 128-, or 256-byte multi-block read transfer can only be performed during an SDIO multi-block transfer (CMD53). Do not set these bits to 0 when using a command that includes data transfer.

Downloaded from Arrow.com.

40.2.12 Card Access Option Register (SDOPT)

Address SDHI.SDOPT 0008 AC50h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	WIDTH	—	—	—	—	—	—	—	TOP[3:0]				CTOP[3:0]			
Value after reset	0	1	0	0	0	0	0	0	1	1	1	0	1	1	1	0

Bit	Symbol	Bit Name	Description		R/W
b3 to b0	CTOP[3:0]	Card Detection Time Counter *1	b3 b0 0 0 0 0: PCLKB × 2^{10} 0 0 0 1: PCLKB × 2^{11} 0 0 1 0: PCLKB × 2^{12} 0 0 1 1: PCLKB × 2^{13} 0 1 0 0: PCLKB × 2^{14} 0 1 0 1: PCLKB × 2^{15} 0 1 1 0: PCLKB × 2^{16} 0 1 1 1: PCLKB × 2^{17}	b3 b0 1 0 0 0: PCLKB × 2^{18} 1 0 0 1: PCLKB × 2^{19} 1 0 1 0: PCLKB × 2^{20} 1 0 1 1: PCLKB × 2^{21} 1 1 0 0: PCLKB × 2^{22} 1 1 0 1: PCLKB × 2^{23} 1 1 1 0: PCLKB × 2^{24} 1 1 1 1: Do not set this value.	R/W
b7 to b4	TOP[3:0]	Timeout Counter *1	b7 b4 0 0 0 0: SDHI clock × 2^{13} 0 0 0 1: SDHI clock × 2^{14} 0 0 1 0: SDHI clock × 2^{15} 0 0 1 1: SDHI clock × 2^{16} 0 1 0 0: SDHI clock × 2^{17} 0 1 0 1: SDHI clock × 2^{18} 0 1 1 0: SDHI clock × 2^{19} 0 1 1 1: SDHI clock × 2^{20}	b7 b4 1 0 0 0: SDHI clock × 2^{21} 1 0 0 1: SDHI clock × 2^{22} 1 0 1 0: SDHI clock × 2^{23} 1 0 1 1: SDHI clock × 2^{24} 1 1 0 0: SDHI clock × 2^{25} 1 1 0 1: SDHI clock × 2^{26} 1 1 1 0: SDHI clock × 2^{27} 1 1 1 1: Do not set this value.	R/W
b8	—	Reserved	This bit is 0 when read. Set it to 0 when writing.		R/W
b12 to b9	—	Reserved	These bits are 0 when read and cannot be modified.		R
b13	—	Reserved	This bit is 0 when read. Set it to 0 when writing.		R/W
b14	—	Reserved	This bit is 1 when read and cannot be modified.		R
b15	WIDTH	SD Bus Width Select *1	0: Wide bus mode (4 bits) 1: Default bus mode (1 bit)		R/W
b31 to b16	—	Reserved	These bits are 0 when read and cannot be modified.		R

Note 1. Do not rewrite these bits when the SDSTS2.CBSY flag is 1.

The SD bus width and timeout counter are set in the SDOPT register.

Downloaded from Arrow.com.

40.2.13 SD Error Status Register 1 (SDERSTS1)

Address(es): SDHI.SDERSTS1 0008 AC58h

	b31	b30	b29	b28	b27	b26	b25	b24
	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8
	—	CRCTK[2:0]			CRCTKE	RDCRCE	RSPCRCE1	RSPCRCE0
Value after reset:	0	0	1	0	0	0	0	0

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	CRCLENE	RDLENE	RSPLENE1	RSPLENE0	CMDE1	CMDE0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CMDE0	Command Error Flag 0	0: Command index field value for a command *1 response is error free 1: Command index field value for a command *1 response is in error	R
b1	CMDE1	Command Error Flag 1	0: Command index field value for a command *2 response is error free 1: Command index field value for a command *2 response is in error (by setting the SDCMD.CMDIDX[5:0] bits, the error that occurs by issuing CMD12 is indicated by the CMDE0 flag)	R
b2	RSPLENE0	Response Length Error Flag 0	0: Command *1 response length is error free 1: Command *1 response length is in error	R
b3	RSPLENE1	Response Length Error Flag 1	0: Command *2 response length is error free 1: Command *2 response length is in error (by setting the SDCMD.CMDIDX[5:0] bits, the error that occurs by issuing CMD12 is indicated by the RSPLENE0 flag)	R
b4	RDLENE	Read Data Length Error Flag	0: Read data length error did not occur 1: Read data length error occurred	R
b5	CRCLENE	CRC Status Token Length Error Flag	0: CRC status token length error did not occur 1: CRC status token length error occurred	R
b7, b6	—	Reserved	These bits are 0 when read.	R
b8	RSPCRCE0	Response CRC Error Flag 0	0: No CRC error detected in command *1 response 1: CRC error detected in command *1 response	R
b9	RSPCRCE1	Response CRC Error Flag 1	0: No CRC error detected in command *2 response 1: CRC error detected in command *2 response (the error that occurs for CMD12 with the setting of the SDCMD.CMDIDX[5:0] bits is indicated by the RSPCRCE0 flag)	R
b10	RDCRCE	Read Data CRC Error Flag	0: No CRC error detected in read data 1: CRC error detected in read data	R
b11	CRCTKE	CRC Status Token Error Flag	0: No error detected in CRC status token 1: Error detected in CRC status token	R
b14 to b12	CRCTK[2:0]	CRC Status Token	Store the CRC status token value (normal value is 010b)	R
b15	—	Reserved	This bit is 0 when read.	R
b31 to b16	—	Reserved	These bits are undefined when read.	R

Note 1. Command other than CMD12 or CMD52 which are automatically issued to stop data transfer.
Note 2. CMD12 or CMD52 which are automatically issued to stop data transfer.

The SDERSTS1 register indicates the CRC status token, CRC error, end bit error, and command error.

Downloaded from Arrow.com.

40.2.14 SD Error Status Register 2 (SDERSTS2)

Address(es): SDHI.SDERSTS2 0008 AC5Ch

	b31	b30	b29	b28	b27	b26	b25	b24
	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8
	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

	b7	b6	b5	b4	b3	b2	b1	b0
	—	CRCBSYTO	CRCTO	RDTO	BSYTO1	BSYTO0	RSPTO1	RSPTO0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RSPTO0	Response Timeout Flag 0	0: After a command *1 was issued, a response was received in less than 640 cycles of the SDHI clock. 1: After a command *1 was issued, a response was not received even after 640 cycles or more of the SDHI clock elapsed.	R
b1	RSPTO1	Response Timeout Flag 1	0: After a command *2 was issued, a response was received in less than 640 cycles of the SDHI clock. 1: After a command *2 was issued, a response was not received even after 640 cycles or more of the SDHI clock elapsed (by setting the SDCMD.CMDIDX[5:0] bits, the error that occurs by issuing CMD12 is indicated by the RSPTO0 flag).	R
b2	BSYTO0	Busy Timeout Flag 0	0: After the R1b response was received, the SDHI was released from the busy state during the specified period *3. 1: After the R1b response was received, the SDHI was in the busy state even after the specified period *3 elapsed.	R
b3	BSYTO1	Busy Timeout Flag 1	0: After CMD12 was automatically issued, the SDHI was released from the busy state during the specified period *3. 1: After CMD12 was automatically issued, the SDHI was in the busy state even after the specified period *3 elapsed (by setting the SDCMD.CMDIDX[5:0] bits, the error that occurs by issuing CMD12 is indicated by the BSYTO0 flag).	R
b4	RDTO	Read Data Timeout Flag	After a read command is issued, this flag becomes 1 when read data is not received even after the specified period *3 elapses. After read data is received, this flag becomes 1 when the next block of read data is not received even after the specified period *3 elapses. After the SDHI exits the read wait state, this flag becomes 1 when the next block of read data is not received even after the specified period *3 elapses.	R
b5	CRCTO	CRC Status Token Timeout Flag	0: After data was written to the SD card, a CRC status token was received during the specified period *3. 1: After CRC data was written to the SD card, a CRC status token was not received even after the specified period *3 elapsed.	R
b6	CRCBSYTO	CRC Status Token Busy Timeout Flag	0: After a CRC status token was received, the SDHI was released from the busy state during the specified period *3. 1: After a CRC status token was received, the SDHI is in the busy state even after the specified period *3 elapsed.	R
b31 to b7	—	Reserved	These bits are 0 when read.	R

Note 1. Command other than CMD12 or CMD52 which are automatically issued to stop data transfer.
Note 2. CMD12 or CMD52 which are automatically issued to stop transfer.

Downloaded from Arrow.com.

Note 3. Set the SDOPT.TOP[3:0] bits to select the number of *n* cycles.

The SDERSTS2 register indicates the timeout status.

40.2.15 SD Buffer Register (SDBUFR)

Address(es): SDHI.SDBUFR 0008 AC60h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	Undefined																															

The SDBUFR register is used when writing data to the SD card and when reading data from the SD card. The SDBUFR register is connected to the SDHI's internal SD buffer. Refer to section 40.3.1, Data Block Format of the SD Card for details on the configuration of the SDBUFR register and the SD buffer.

40.2.16 SDIO Mode Control Register (SDIOMD)

Address(es): SDHI.SDIOMD 0008 AC68h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	C52PUB	IOABT	—	—	—	—	—	RWREQ	—	INTEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	INTEN	SDIO Interrupt Acceptance Enable *1	0: SDIO interrupt accept disabled 1: SDIO interrupt accept enabled	R/W
b1	—	Reserved	This bit is 0 when read and cannot be modified.	R
b2	RWREQ	Read Wait Request	0: SDHI exits read wait state 1: Request for SDHI to enter read wait state	R/W
b7 to b3	—	Reserved	These bits are 0 when read and cannot be modified.	R
b8	IOABT	SDIO Abort	If this bit is set to 1 during multi-block transfer triggered by CMD53, CMD52 is immediately issued, and the command sequence is aborted.	R/W
b9	C52PUB	SDIO None Abort	If this bit is set to 1 during multi-block transfer triggered by CMD53, CMD52 is issued before the transfer process is complete, and the command sequence is completed.	R/W
b31 to b10	—	Reserved	These bits are 0 when read and cannot be modified.	R

Note 1. Do not rewrite this bit when the SDSTS2.CBSY flag is 1.

The SDIOMD register controls reception of the SDIO interrupt, controls CMD52 issuance during multi-block transfer, and controls the read wait request. Do not set bits C52PUB and IOABT to 1 at the same time.

Downloaded from Arrow.com.

RWREQ Bit (Read Wait Request)

If the RWREQ bit is set to 1 during a multi-block read sequence triggered by issuing CMD53, when the current block is done being read, the SDHI enters the read wait state. The method for exiting the read wait state is as follows.

- If the RWREQ bit is set to 0 while the SDHI is in the read wait state, the SDHI exits the read wait state.
- If the IOABT bit is set to 1 while the SDHI is in the read wait state, after CMD52 is issued, the RWREQ bit becomes 0 and the SDHI exits the read wait state.
- If bits C52PUB and RWREQ are simultaneously set to 1 during a multi-block read sequence triggered by issuing CMD53 *1, the SDHI does not automatically exit the read wait state, so after receiving the CMD52 response, set the RWREQ bit to 0.

Note 1. Set bits RWREQ and C52PUB to 1 simultaneously.

If the RWREQ bit is set to 1 while the last block is being transferred during a multi-block read sequence triggered by issuing CMD53, the SDHI will not enter the read wait state, the SDSTS1.ACEND flag becomes 1, and the RWREQ bit becomes 0. Set the RWREQ bit to 1 after the SDSTS1.RSPEND flag becomes 1.

IOABT Bit (SDIO Abort)

- If the IOABT bit is set to 1 during a multi-block transfer sequence triggered by issuing CMD53, the SDHI stops the CMD53 command sequence, and CMD52 is issued. If the command sequence is stopped due to a communication error or timeout, the SDHI does not issue CMD52. The SD buffer can be accessed even after the IOABT bit is set to 1, but the SDSTS2.ILR flag or ILW flag becomes 1, and a buffer access error occurs. Write a value to the SDARG register before setting the IOABT bit to 1.
- During a single block write, if there is no data in the SD buffer when the IOABT bit is set to 1, the SDHI does not issue CMD52, and the SDSTS1.ACEND flag becomes 1. If there is data in the SD buffer when the IOABT bit is set to 1, the SDHI does not issue CMD52, and after the SDHI exits the busy state, the SDSTS1.ACEND flag becomes 1.
- If the IOABT bit is set to 1 during a single block read, the SDHI does not issue CMD52, and the SDSTS1.ACEND flag immediately becomes 1.
- If the SDHI is in the busy state after the R1b response is received and the IOABT bit is set to 1, the SDHI does not issue CMD52, and after the SDHI exits the busy state, the SDSTS1.ACEND flag becomes 1.
- If the IOABT bit is set to 1 after the command sequence is completed, the SDHI does not issue CMD52, and the SDSTS1.ACEND flag does not become 1.
- Set the IOABT bit to 1 after the SDSTS1.RSPEND flag becomes 1.
- Set the IOABT bit to 0 after the SDSTS1.ACEND flag becomes 1.

C52PUB Bit (SDIO None Abort)

- If the C52PUB bit is set to 1 during a multi-block write sequence triggered by issuing CMD53, CMD52 is automatically issued when the SD buffer is empty and the current block write access is complete. The C52PUB bit becomes 0 after the response for CMD52 is received. If the C52PUB bit is 1 while the last block is being transferred, the SDHI does not issue CMD52, and after the SDSTS1.RSPEND flag becomes 1, the C52PUB bit is set to 0.
- If the C52PUB bit and RWREQ bit are set to 1 during a multi-block read sequence triggered by issuing CMD53, the SDHI enters the read wait state after the current block read access is complete, and the SDHI automatically issues CMD52. The C52PUB bit becomes 0 after the response for CMD52 is received. If the C52PUB bit is set to 1 while the last block is being transferred, the SDHI does not issue CMD52, and after the SDSTS1.RSPEND flag becomes 1, the C52PUB bit is set to 0.
- During a multi-block read sequence triggered by issuing CMD53, if the C52PUB bit is set to 1, also set the RWREQ bit to 1.
- Write a value to the SDARG register before setting the C52PUB bit to 1.
- Set the C52PUB bit to 1 after the SDSTS1.RSPEND flag becomes 1.

Downloaded from Arrow.com.

40.2.17 SDIO Status Register (SDIOSTS)

Address(es): SDHI.SDIOSTS 0008 AC6Ch

	b31	b30	b29	b28	b27	b26	b25	b24
	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8
	EXWT	EXPUB52	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	IOIRQ
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	IOIRQ	SDIO Interrupt Status Flag	0: SDIO interrupt not accepted 1: SDIO interrupt accepted	R/(W) *1
b2, b1	—	Reserved	These bits are undefined when read. Set them to 1 when writing.	R/W
b13 to b3	—	Reserved	These bits are 0 when read and cannot be modified.	R
b14	EXPUB52	EXPUB52 Status Flag	Indicates the status of the EXPUB52	R/(W) *1
b15	EXWT	EXWT Status Flag	Indicates the status of the EXWT	R/(W) *1
b31 to b16	—	Reserved	These bits are 0 when read and cannot be modified.	R

Note 1. The flag value does not change even when set to 1. If 0 is written to this flag, it becomes 0.

The SDIOSTS register indicates the status of the SDIO card access. When clearing a flag, bits to be cleared should be set to 0; bits not being cleared should be set to 1.

IOIRQ Flag (SDIO Interrupt Status Flag)

— This flag becomes 1 under the following condition:

- The SDIO interrupt from the SDIO card is accepted while the SDIOMD.INTEN bit is 1.

— This flag becomes 0 under the following condition:

- The flag is set to 0. *1

Note 1. Access the SDIO card, negate the SDIO interrupt from the SDIO card, and then set the IOIRQ flag to 0. If the SDIO interrupt from the SDIO card is not negated, the IOIRQ flag might become 1 again.

EXPUB52 Flag (EXPUB52 Status Flag)

— This flag becomes 1 under any of the following conditions:

- When multi-block transfer is triggered by CMD53 being issued, the SDIOMD.C52PUB bit is set to 1 while the last block is being transferred.
- When multi-block write is triggered by CMD53 being issued, the SDIOMD.C52PUB bit remains set to 1 while the last block is being transferred.

— This flag becomes 0 under the following condition:

- The flag is set to 0.

Downloaded from Arrow.com.

EXWT Flag (EXWT Status Flag)

— This flag becomes 1 under the following condition:

- During a multi-block read sequence triggered by CMD53 being issued, the SDIOMD.RWREQ bit is set to 1 while the last block is being transferred.

— This flag becomes 0 under the following condition:

- The flag is set to 0.

40.2.18 SDIO Interrupt Mask Register (SDIOIMSK)

Address(es): SDHI.SDIOIMSK 0008 AC70h

	b31	b30	b29	b28	b27	b26	b25	b24
	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8
	EXWTM	EXPUB52M	—	—	—	—	—	—
Value after reset:	1	1	0	0	0	0	0	0

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	IOIRQM
Value after reset:	0	0	0	0	0	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b0	IOIRQM	IOIRQ Interrupt Mask Control	0: IOIRQ interrupt not masked 1: IOIRQ interrupt masked	R/W
b2, b1	—	Reserved	These bits are 1 when read. Set them to 1 when writing.	R/W
b13 to b3	—	Reserved	These bits are 0 when read and cannot be modified.	R
b14	EXPUB52M	EXPUB52 Interrupt Request Mask Control	0: EXPUB52 interrupt request not masked 1: EXPUB52 interrupt request masked	R/W
b15	EXWTM	EXWT Interrupt Request Mask Control	0: EXWT interrupt request not masked 1: EXWT interrupt request masked	R/W
b31 to b16	—	Reserved	These bits are 0 when read and cannot be modified.	R

The SDIOIMSK register enables and disables the interrupt requests from the status flags in the SDIOSTS register. Refer to Table 40.8, Interrupt Sources for details on the relationship between the status flags and the requested interrupt source.

Downloaded from Arrow.com.

40.2.19 DMA Transfer Enable Register (SDDMAEN)

Address(es): SDHI.SDDMAEN 0008 ADB0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DMAEN	—
Value after reset:	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is 0 when read. Set it to 0 when writing.	R/W
b1	DMAEN	DMA Transfer Enable *1 *2	0: Using DMAC and DTC to access the SDBUFR register is disabled 1: Using DMAC and DTC to access the SDBUFR register is enabled	R/W
b3, b2	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R
b4	—	Reserved	This bit is 1 when read. Set it to 1 when writing.	R
b5	—	Reserved	This bit is 0 when read. Set it to 0 when writing.	R/W
b7, b6	—	Reserved	These bits are 0 when read and cannot be modified.	R
b9, b8	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R/W
b11, b10	—	Reserved	These bits are 0 when read and cannot be modified.	R
b12	—	Reserved	This bit is 1 when read. Set it to 1 when writing.	R
b31 to b13	—	Reserved	These bits are 0 when read and cannot be modified.	R

Note 1. Do not rewrite this bit when the SDSTS2.CBSY bit is 1.
Note 2. When the SDIMSK2.BWEM bit is 0 or the SDIMSK2.BREM bit is 0, set the SDDMAEN.DMAEN bit to 0. When the SDDMAEN.DMAEN bit is 1, set the SDIMSK2.BWEM bit to 1 and the SDIMSK2.BREM bit to 1.

The SDDMAEN register enables and disables DMA transfer.

DMAEN Bit (DMA Transfer Enable)

When using DMA transfer to access the SD buffer, set the DMAEN bit to 1 before setting the SDCMD register.

Downloaded from Arrow.com.

40.2.20 SDHI Software Reset Register (SDRST)

Address(es): SDHI.SDRST 0008 ADC0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SDRST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b0	SDRST	SDHI Software Reset Control	0: SDHI software reset 1: SDHI software reset released	R/W
b2, b1	—	Reserved	These bits are 1 when read. Set them to 1 when writing.	R
b31 to b3	—	Reserved	These bits are 0 when read and cannot be modified.	R

Table 40.5 lists the bits and flags initialized by the SDHI software reset.

Table 40.5 Bits and Flags Initialized by the SDHI Software Reset

Register	Bit/Flag
SDSTOP	SDBLKCNTEN
SDSTS1	RSPEND, ACEND
SDSTS2	CMDE, CRCE, ENDE, DTO, ILW, ILR, RSPTO, SDD0MON, BRE, BWE, SDCLKCREN, ILA
SDCLKCR	CLKEN
SDOPT	CTOP[3:0], TOP[3:0], WIDTH Bits b8 and b13 in the SDOPT register are also initialized by the SDHI software reset.
SDERSTS1	CMDE0, CMDE1, RSPLENE0, RSPLENE1, RDLENE, CRCLENE, RSPCRCE0, RSPCRCE1, RDCRCE, CRCTKE, CRCTK[2:0]
SDERSTS2	RSPTO0, RSPTO1, BSYTO0, BSYTO1, RDTO, CRCTO, CRCBSYTO
SDIOSTS	IOIRQ, EXPUB52, EXWT

Downloaded from Arrow.com.

40.2.21 Swap Control Register (SDSWAP)

Address(es): SDHI.SDSWAP 0008 ADE0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	BRSWP	BWSWP	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is 0 when read and cannot be modified.	R
b1	—	Reserved	This bit is 0 when read. Set it to 0 when writing.	R/W
b2	—	Reserved	This bit is 0 when read and cannot be modified.	R
b4, b3	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R/W
b5	—	Reserved	This bit is 0 when read and cannot be modified.	R
b6	BWSWP	SDBUFR Swap Write *1	0: Normal write operation 1: Swap the byte endian before writing to the SDBUFR register	R/W
b7	BRSWP	SDBUFR Swap Read *1	0: Normal read operation 1: Swap the byte endian before reading the SDBUFR register	R/W
b10 to b8	—	Reserved	These bits are 0 when read and cannot be modified.	R
b12, b11	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R/W
b14, b13	—	Reserved	These bits are 0 when read and cannot be modified.	R
b15	—	Reserved	This bit is 0 when read. Set it to 0 when writing.	R/W
b31 to b16	—	Reserved	These bits are 0 when read and cannot be modified.	R

Note 1. Do not rewrite this bit when the SDSTS2.CBSY flag is 1.

The SDSWAP register is used to select whether or not the byte endian is swapped when accessing the SDBUFR register. Refer to section 40.3.1 for details on the differences in accessing the SDBUFR register based on the SDSWAP register value.

Downloaded from Arrow.com.

40.3 SDHI Operation

40.3.1 Data Block Format of the SD Card

The SDHI has a default bus mode (1-bit width) that uses just the SDHI_D0 pin as a data line and a wide bus mode (4-bit width) that uses pins SDHI_D0 to SDHI_D3. Figure 40.2 shows the transfer format when the SDOPT.WIDTH bit is 1 (default bus mode), and Figure 40.3 shows the transfer format when the SDOPT.WIDTH bit is 0 (wide bus mode).

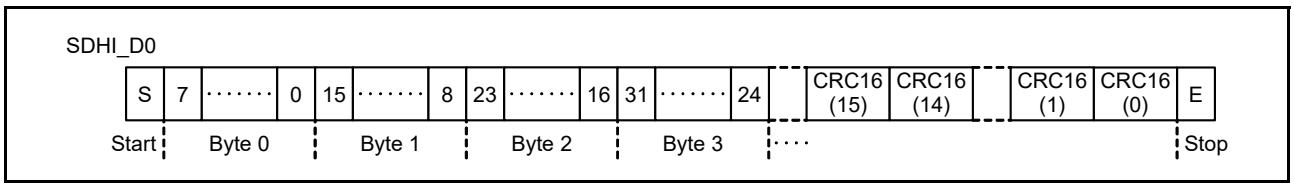


Figure 40.2 Transfer Format in Default Bus Mode

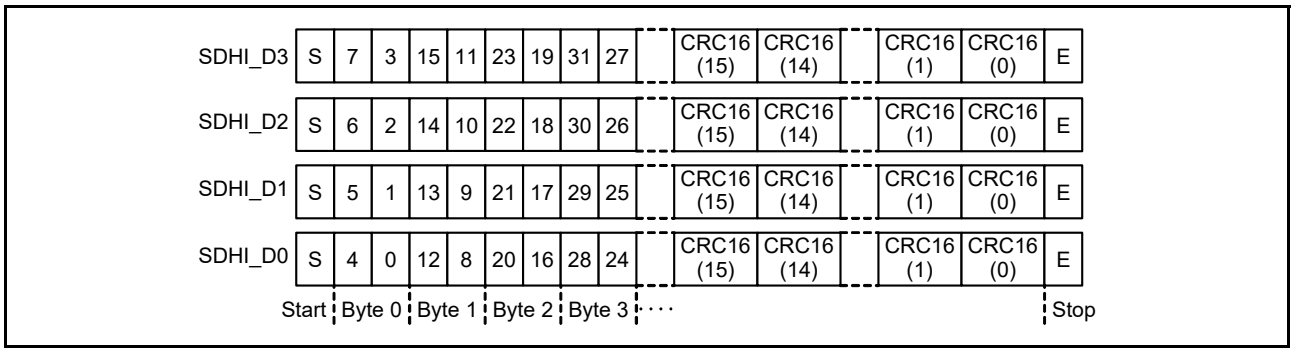


Figure 40.3 Transfer Format in Wide Bus Mode

Downloaded from Arrow.com.

40.3.2 SD Buffer and the SDBUFR Register

The SDHI transfers data to an SD card via its internal SD buffer. The SD buffer is comprised of a double buffer, and each buffer is 512 bytes. Figure 40.4 shows the data configuration of a single buffer of the SD buffer's double buffer.

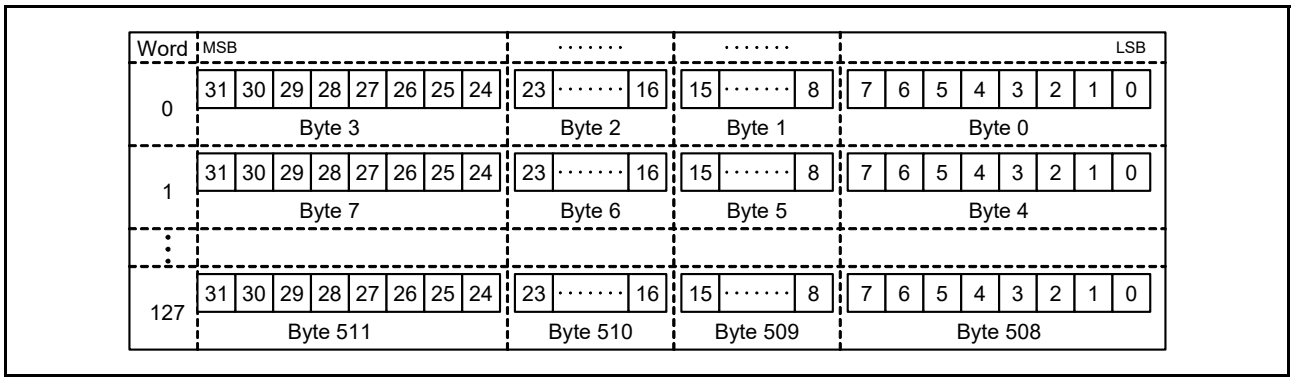


Figure 40.4 Data Configuration of Single Buffer in the SD Buffer

Access to the SD buffer is done via the SDBUFR register. If data is written to the SDBUFR register while the SDSWAP.BWSWP bit is 1, the SDHI swaps the endian for the byte, and stores the data in the SDBUFR register. If data is read from the SDBUFR register while the SDSWAP.BRSWP bit is 1, the data of the endian for the byte that was swapped can be read. Figure 40.5 shows the data alignment when reading the SDBUFR register.

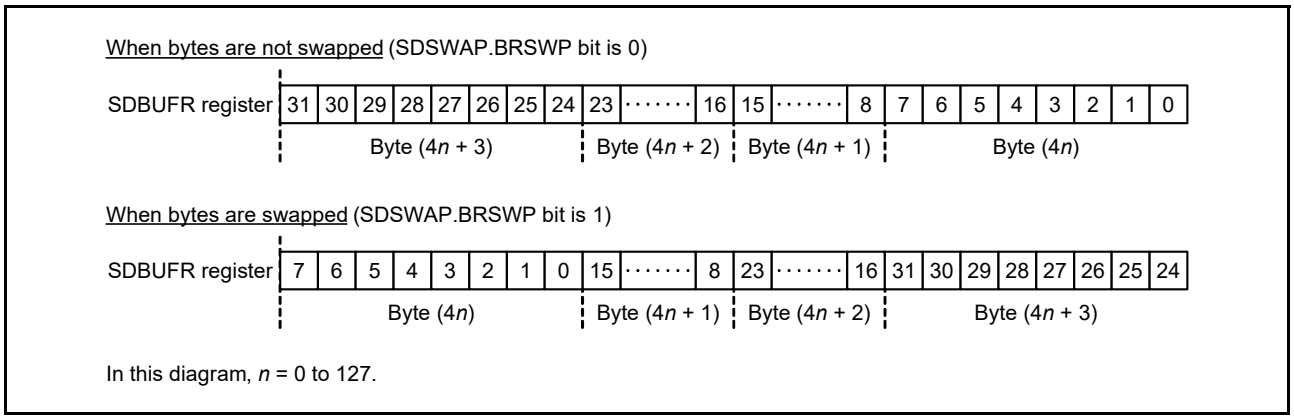


Figure 40.5 Data Alignment When Reading the SDBUFR Register

Downloaded from Arrow.com.

40.3.3 SD Card Detection

The SDHI can detect an SD card using either the SDHI_CD pin or SDHI_D3 pin.

40.3.3.1 Using the SDHI_CD Pin to Detect an SD Card

Figure 40.6 shows the timing chart for SD card detection using the SDHI_CD pin. The SDHI_CD pin is connected to the card detection switch of the SD card connector, and is pulled-up by the MCU. The pull-up resistance value is determined by the specifications of the host device. Note that there are some SD card sockets whose card detection switches become open when the SD card is inserted.

- Detecting SD card insertion
  The signal from the SDHI_CD pin becomes low when an SD card is inserted. This causes the SDSTS1.SDCDIN flag to become 1 if the SDHI_CD pin is low for the number of cycles set in the SDOPT.CTOP[3:0] bits. The SDSTS1.SDCDIN flag is cleared by setting it to 0.
- Detecting SD card removal
  The signal from the SDHI_CD pin becomes high when the SD card is removed. This causes the SDSTS1.SDCDRM flag to become 1 if the SDHI_CD pin is high for the number of cycles set in the SDOPT.CTOP[3:0] bits. The SDSTS1.SDCDRM flag is cleared by setting it to 0.

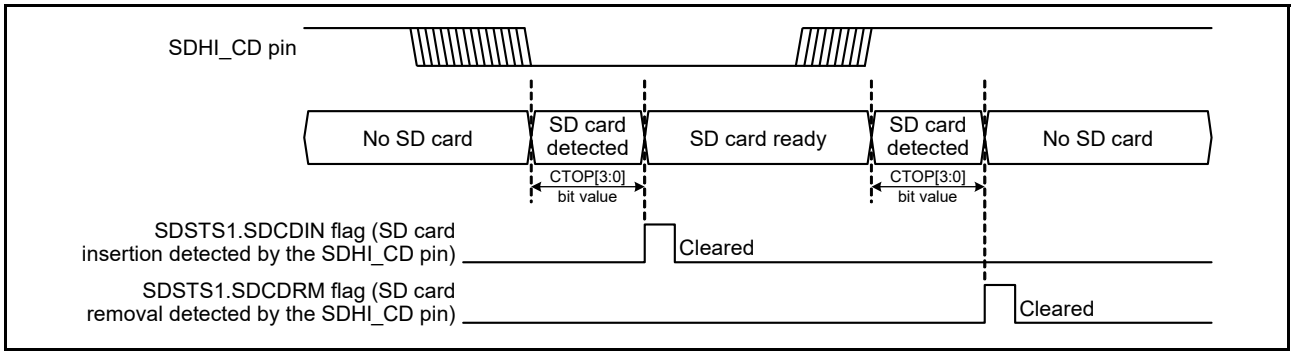


Figure 40.6 SD Card Detection Using the SDHI_CD Pin

40.3.3.2 Using the SDHI_D3 Pin to Detect an SD Card

Figure 40.7 shows the timing chart for SD card detection using the SDHI_D3 pin. The SDHI_D3 pin is pulled-down by the MCU. The pull-down resistance value is determined by the specifications of the host device.

- Detecting SD card insertion
  The signal from the SDHI_D3 pin becomes high when an SD card is inserted. This causes the SDSTS1.SDD3IN flag to become 1. The SDSTS1.SDD3IN flag is cleared by setting it to 0.
- Detecting SD card removal
  The signal from the SDHI_D3 pin becomes low when the SD card is removed. This causes the SDSTS1.SDD3RM flag to become 1. The SDSTS1.SDD3RM flag is cleared by setting it to 0.

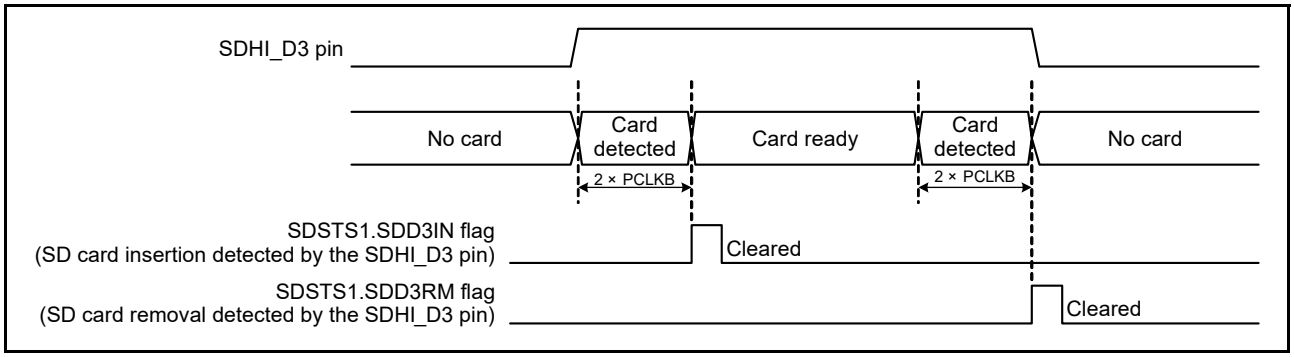


Figure 40.7 SD Card Detection Using the SDHI_D3 Pin

Downloaded from Arrow.com.

40.3.4 SD Card Write Protection

The SDHI can disable writing to an SD card via the SDHI_WP pin or a command.

40.3.4.1 Using the SDHI_WP Pin to Enable Write Protection

The SDHI_WP pin is connected to the WP detection switch of the SD card connector, and the SDHI_WP pin is pulled-down or pulled-up when an SD card is inserted. The resistance value and whether the SDHI_WP pin is pulled-up or pulled-down are determined by the specifications of the host device. The status of the SDHI_WP pin is reflected in the SDSTS1.SDWPMON flag. After an SD card is inserted, read the SDSTS1.SDWPMON flag to check if write protection is enabled or disabled.

40.3.4.2 Using a Command to Enable Write Protection

The SDHI uses the write protect command or the SD card lock command to disable writing to the SD card.

Downloaded from Arrow.com.

40.3.5 Communication Errors and Timeouts

When a communication error or timeout error occurs, depending on the type of error, the corresponding status flag in the SDSTS2 register becomes 1. Also, depending on the source of the error, the corresponding flag in the SDERSTS1 or SDERSTS2 register becomes 1.

The status flags in registers SDERSTS1 and SDERSTS2 become 0 by writing to the SDCMD register, or by setting the SDRST.SDRST bit to 0.

Table 40.6 Communication Errors

Communication Error	Interrupt Flag Register		Error Status Register		This Occurs When...
	Register symbol	Bit symbol	Register symbol	Bit symbol	
End bit error	SDSTS2	ENDE	SDERSTS1	CRCLENE	The CRC status token length is in error
				RDLENE	The read data length is in error
				RSPLENE1	The response length is in error *1
				RSPLENE0	The response length is in error *2
CRC error		CRCE		CRCTKE	The CRC status token is in error
				RDCRCE	There is a CRC error in the read data
				RSPCRCE1	There is a CRC error in the response *1
				RSPCRCE0	There is a CRC error in the response *2
Command error		CMDE		CMDE1	The command index field value for the transmitted command and received response do not match *1
				CMDE0	The command index field value for the transmitted command and received response do not match *2

Note 1. CMD12 or CMD52 which are automatically issued to stop transfer.
Note 2. A command other than CMD12 or CMD52 which are automatically issued to stop transfer.

Table 40.7 Timeouts

Timeout	Interrupt Flag Register		Error Status Register		This Occurs When...
	Register symbol	Bit symbol	Register symbol	Bit symbol	
Response timeout	SDSTS2	RSPTO	SDERSTS2	RSPTO1	A response is not received even after a minimum of 640 SDHI clock cycles elapse *1
				RSPTO0	A response is not received even after a minimum of 640 SDHI clock cycles elapse *2
Data timeout (excluding response timeout)		DTO		CRCBSYTO	After the CRC status token is received, the SDHI is busy for at least the period set *3
				CRCTO	After the write data is transmitted, the CRC status token is not received even after at least the period set *3 elapses
				RDTO	After the read command is issued, the read data is not received even after at least the period set *3 elapses
					After the read data is received, the next block read data is not received even after at least the period set *3 elapses
					After the SDHI exits the read wait state, the next block read data is not received even after at least the period set *3 elapses
				BSYTO1	After CMD12 is issued during the command sequence, the SDHI is busy for at least the period set *3
				BSYTO0	After the R1b response is received, the SDHI is busy for at least the period set *3 (a command other than CMD12 is issued during the command sequence)

Note 1. CMD12 or CMD52 which are automatically issued to stop transfer.
Note 2. A command other than CMD12 or CMD52 which are automatically issued to stop transfer.
Note 3. The period is set in the SDOPT.TOP[3:0] bits.

Downloaded from Arrow.com.

40.3.6 Examples of Issuing a Command

40.3.6.1 Command Absent of Response Reception and Data Transfer

Figure 40.8 shows an example of no response being received and no data being transferred after the SDHI issues a command.

1. Set the flags in registers SDSTS1 and SDSTS2 to 0.
2. Set the SDHI clock in the SDCLKCR register, and set the interrupt requests to be masked in registers SDIMSK1 and SDIMSK2. Refer to section 40.5.5 for details on setting the SDCLKCR register.
3. After setting the argument field value to the SDARG register, write the command information to be sent to the SDCMD register. The SDHI issues a command when a value is written to the SDCMD register.
4. After a command is issued, the SDSTS1.RSPEND flag becomes 1, and a response end interrupt request is generated.
5. Set the SDSTS1.RSPEND flag to 0.

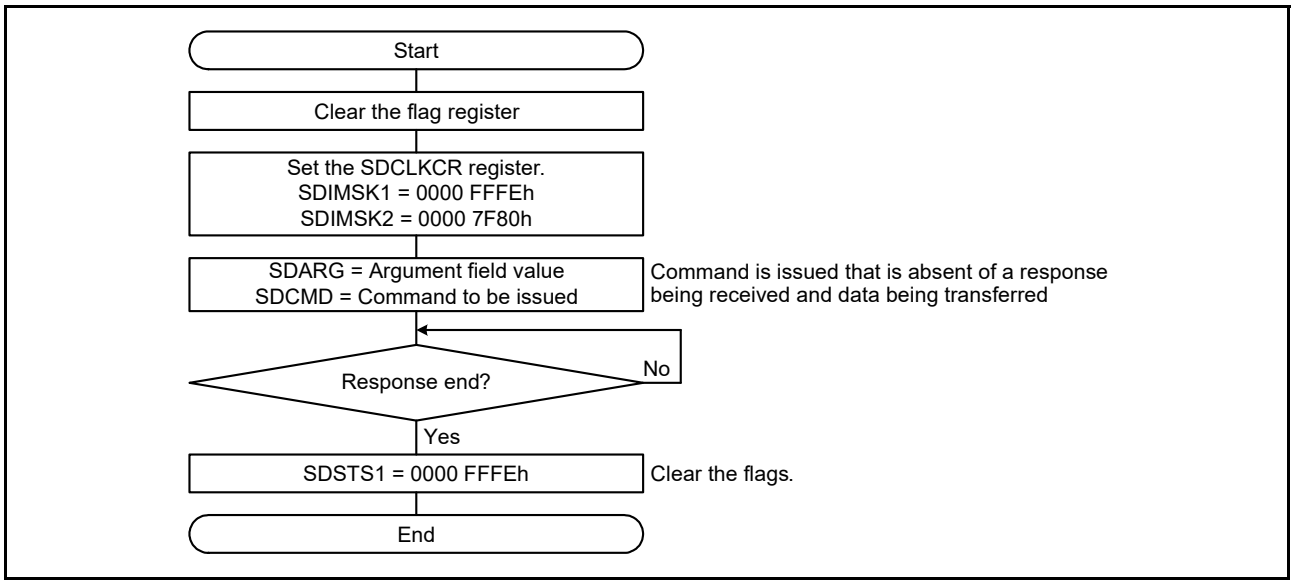


Figure 40.8 Command Issued That Is Absent of Response Reception and Data Transfer

Downloaded from Arrow.com.

40.3.6.2 Command Absent of Data Transfer

Figure 40.9 shows an example of no data being transferred after the SDHI issues a command.

1. Set the flags in registers SDSTS1 and SDSTS2 to 0.
2. Set the SDHI clock in the SDCLKCR register, and set the interrupt requests to be masked in registers SDIMSK1 and SDIMSK2. Refer to section 40.5.5 for details on setting the SDCLKCR register.
3. After setting the argument field value in the SDARG register, write the information of the command to be issued to the SDCMD register. The SDHI issues a command when a value is written to the SDCMD register.
4. After a response is received, the SDSTS1.RSPEND flag becomes 1, and a response end interrupt request is generated.
5. Set the SDSTS1.RSPEND flag to 0, and read the response stored in the SDRSP10 register.

Perform error processing (clear the interrupt flag) if a communication error or timeout occurs.

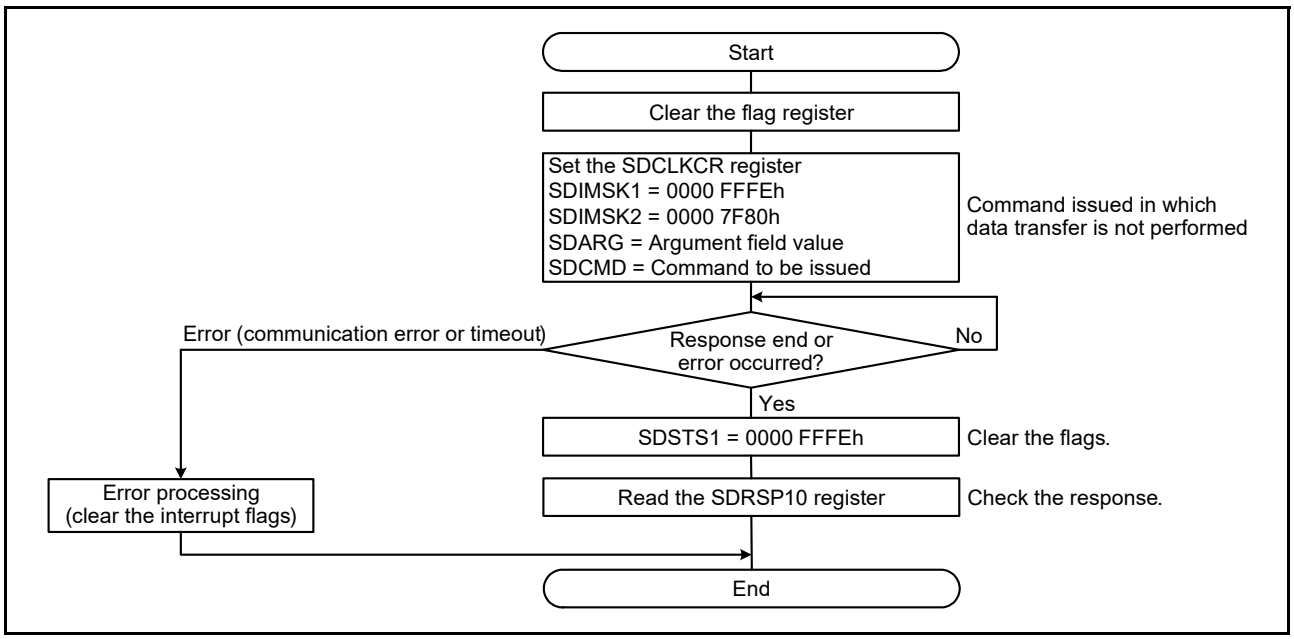


Figure 40.9 Command Issued That Is Absent of Data Transfer

Downloaded from Arrow.com.

40.3.6.3 Single Block Read Command (CMD17)

Figure 40.10 shows an example of issuing the single block read command (CMD17).

1. Set the flags in registers SDSTS1 and SDSTS2 to 0.
2. Set the SDHI clock in the SDCLKCR register, and set the interrupt requests to be masked in registers SDIMSK1 and SDIMSK2. Refer to section 40.5.5 for details on setting the SDCLKCR register.
3. After setting the argument field value for CMD17 to the SDARG register, write 0000 0011h to the SDCMD register. The SDHI issues CMD17 when a value is written to the SDCMD register.
4. When the response is received, the SDSTS1.RSPEND flag becomes 1, and a response end interrupt request is generated.
5. Set the SDSTS1.RSPEND flag to 0, and read the response stored in the SDRSP10 register. If the response read is in error, set the SDSTOP.STP bit or SDIOMD.IOABT bit to 1, and the command sequence can be stopped. When the command sequence is stopped, the SDSTS1.ACEND flag becomes 1. Note that CMD12 and CMD52 are not automatically issued by stopping this command sequence.
6. After the response is received, set the SDIMSK1.ACENDM bit to 0 and set the SDIMSK2.BREM bit to 0.
7. After the amount of data set in the SDSIZE.LEN[9:0] bits is received from the SD card, the SDSTS2.BRE flag becomes 1, and the BRE interrupt request is generated.
8. Set the SDSTS2.BRE flag to 0, and read the amount of data set in the SDSIZE.LEN[9:0] bits from the SDBUFR register.
9. After data has been read from the SDBUFR register, the SDSTS1.ACEND flag becomes 1, and the access end interrupt request is generated.
10. Set the SDSTS1.ACEND flag to 0.

Perform error processing (clear the interrupt flag) if a communication error or timeout occurs.

Downloaded from Arrow.com.

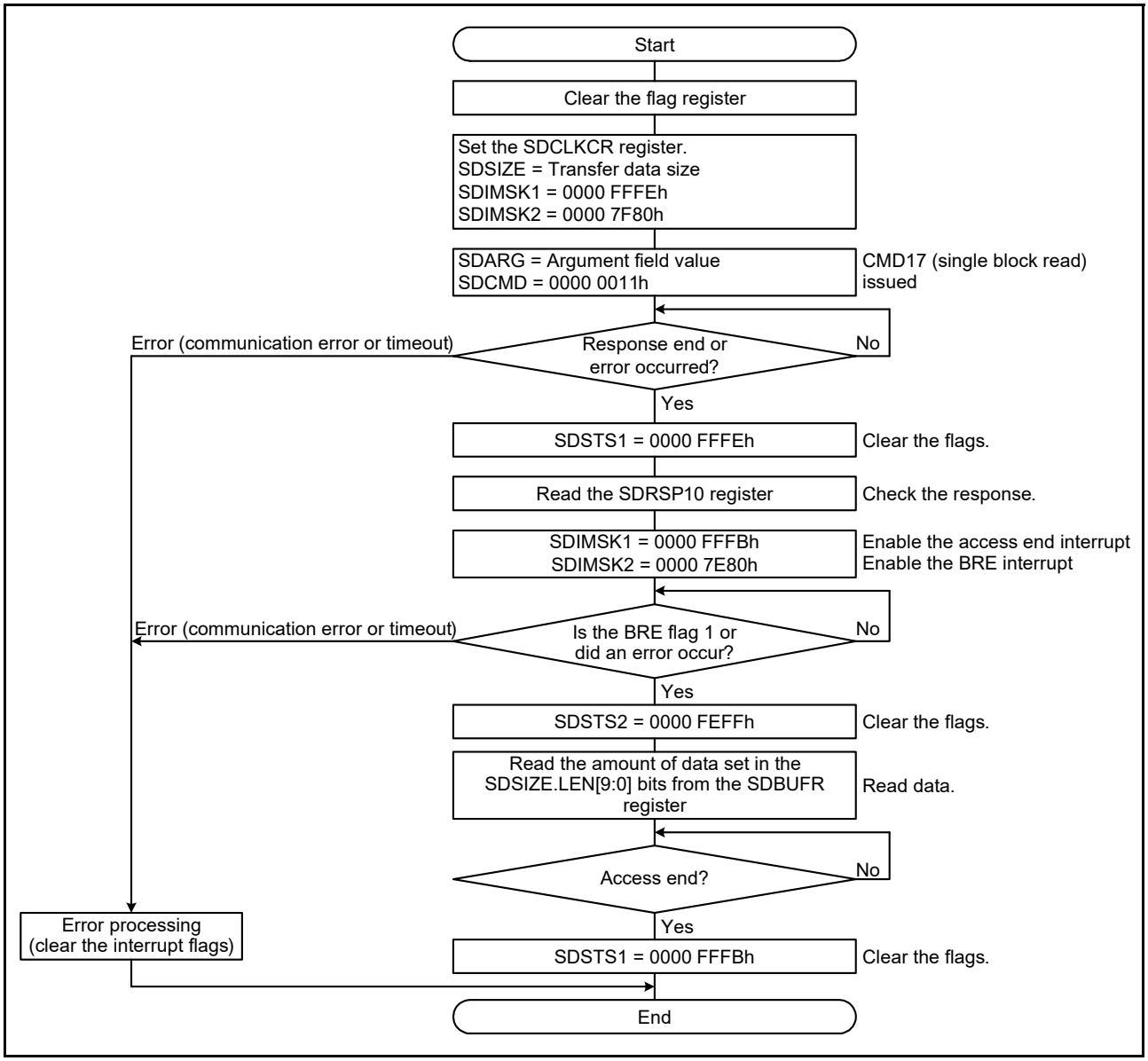


Figure 40.10 Issuing the Single Block Read Command

Downloaded from Arrow.com.

40.3.6.4 Single Block Write Command (CMD24)

Figure 40.11 shows an example of issuing the single block write command (CMD24).

1. Set the flags in registers SDSTS1 and SDSTS2 to 0.
2. Set the SDHI clock in the SDCLKCR register, and set the interrupt requests to be masked in registers SDIMSK1 and SDIMSK2. Refer to section 40.5.5 for details on setting the SDCLKCR register.
3. After setting the argument field value for CMD24 to the SDARG register, write 0000 0018h to the SDCMD register. The SDHI issues CMD24 when a value is written to the SDCMD register.
4. When the response is received, the SDSTS1.RSPEND flag becomes 1, and the response end interrupt request is generated.
5. Set the SDSTS1.RSPEND flag to 0 and read the response stored in the SDRSP10 register. If the read response is in error, set the SDSTOP.STP bit or SDIOMD.IOABT bit to 1, and the command sequence can be stopped. When the command sequence is stopped, the SDSTS1.ACEND flag becomes 1. Note that when this command sequence is stopped, CMD12 and CMD52 are not automatically issued.
6. After the response is received, set the SDIMSK1.ACENDM bit to 0, and set the SDIMSK2.BWEM bit to 0.
7. When the SDBUFR register becomes write accessible, the SDSTS2.BWE flag becomes 1, and the BWE interrupt request is generated.
8. Set the SDSTS2.BWE flag to 0, and write the amount of data set in the SDSIZE.LEN[9:0] bits to the SDBUFR register. After writing to the SDBUFR register, the SDHI transmits write data to the SD card. Also, after writing to the SDBUFR register, data transmission may cause a communication error or timeout to occur.
9. After all data has been written to the SD card, the SDHI receives the CRC status token, and the SDHI_D0 pin line becomes busy (low). Then, when the SDHI exits the busy state, the SDSTS1.ACEND flag becomes 1, and the access end interrupt request is generated.
10. Set the SDSTS1.ACEND flag to 0.

Perform error processing (clear the interrupt flag) if a communication error or timeout occurs.

Downloaded from Arrow.com.

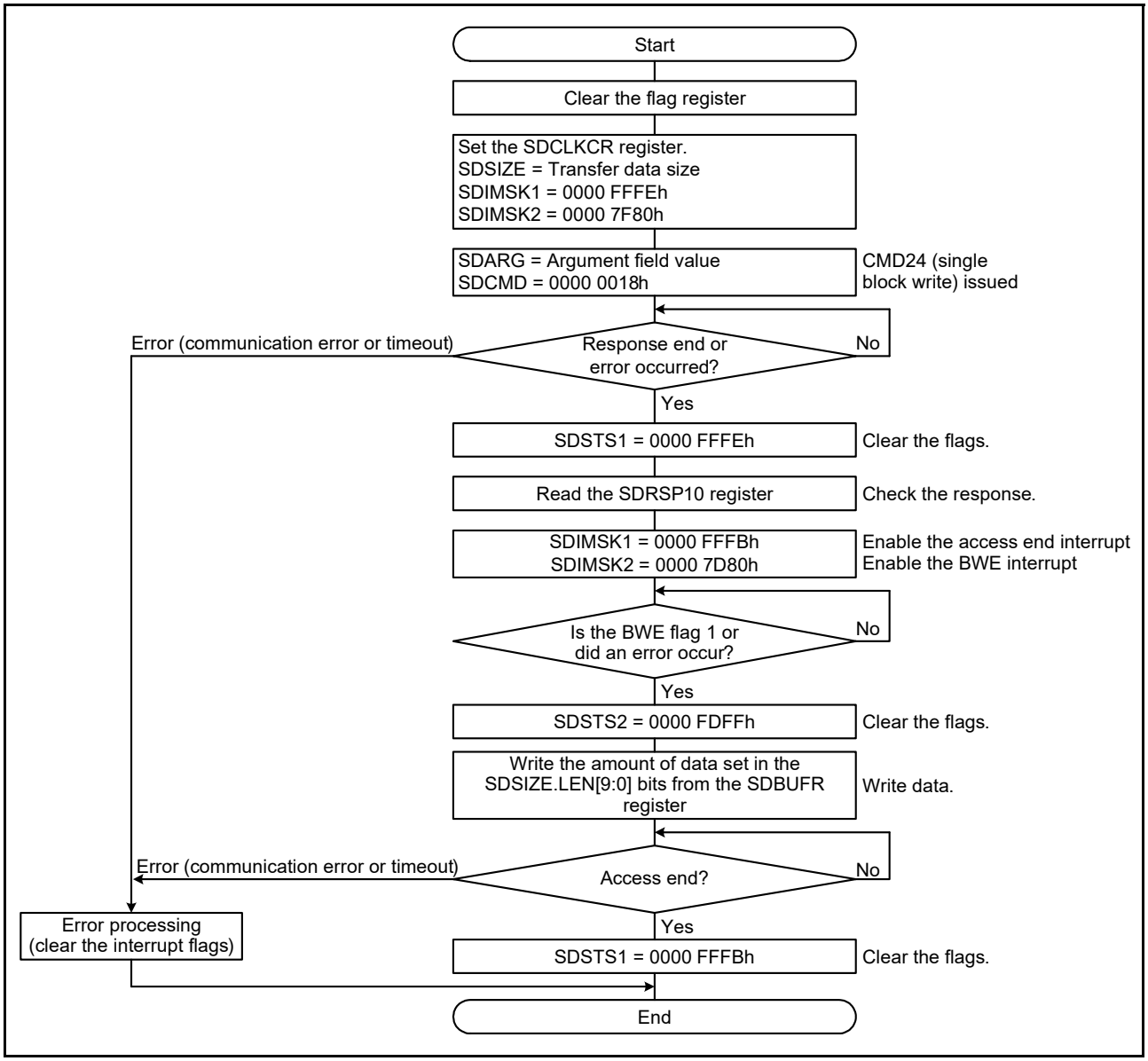


Figure 40.11 Issuing the Single Block Write Command

Downloaded from Arrow.com.

40.3.6.5 Multi-Block Read Command (CMD18)

Figure 40.12 shows an example of issuing the multi-block read command (CMD18).

1. Set the flags in registers SDSTS1 and SDSTS2 to 0.
2. Set the SDHI clock in the SDCLKCR register, and set the interrupt requests to be masked in registers SDIMSK1 and SDIMSK2. Refer to section 40.5.5 for details on setting the SDCLKCR register. Set the SDSTOP.SDBLKCNTEN bit to 1, and set the number of transfer blocks in the SDBLKCNT register.
3. After setting the argument field value for CMD18 to the SDARG register, write 0000 0012h to the SDCMD register. The SDHI issues CMD18 when a value is written to the SDCMD register.
4. When the response is received, the SDSTS1.RSPEND flag becomes 1, and the response end interrupt request is generated.
5. Set the SDSTS1.RSPEND flag to 0 and read the response stored in the SDRSP54 register. If the read response is in error, set the SDSTOP.STP bit to 1, and the command sequence can be stopped. When the SDSTOP.STP bit is set to 1, the SDHI automatically issues CMD12, and the response is received. At this point, the SDSTS1.ACEND flag becomes 1, and if the access end interrupt request is enabled, the access end interrupt request is generated. Next, set the SDSTS1.ACEND flag to 0 and read the response.
6. After the response is received, set the SDIMSK1.ACENDM bit to 0, and set the SDIMSK2.BREM bit to 0.
7. After receiving one block of data from the SD card, the SDSTS2.BRE bit becomes 1, and the BRE interrupt request is generated.
8. Set the SDSTS2.BRE flag to 0, and read the amount of data set in the SDSIZE.LEN[9:0] bits from the SDBUFR register. The read access to the SDBUFR register repeats for the amount of transfer blocks set in the SDBLKCNT register. Also, while reading the SDBUFR register, data reception may cause a communication error or timeout to occur. After the amount of transfer blocks set in the SDBLKCNT register have been read, the SDHI automatically issues CMD12, and the response is received. At this time, the SDHI automatically writes 0000 0000h to the SDARG register.
9. When all blocks have been received and the CMD12 response is received, the SDSTS1.ACEND flag becomes 1 and the access end interrupt request is generated.
10. Set the SDSTS1.ACEND flag to 0 and read the response.

Perform error processing (clear the interrupt flag) if a communication error or timeout occurs.

Downloaded from Arrow.com.

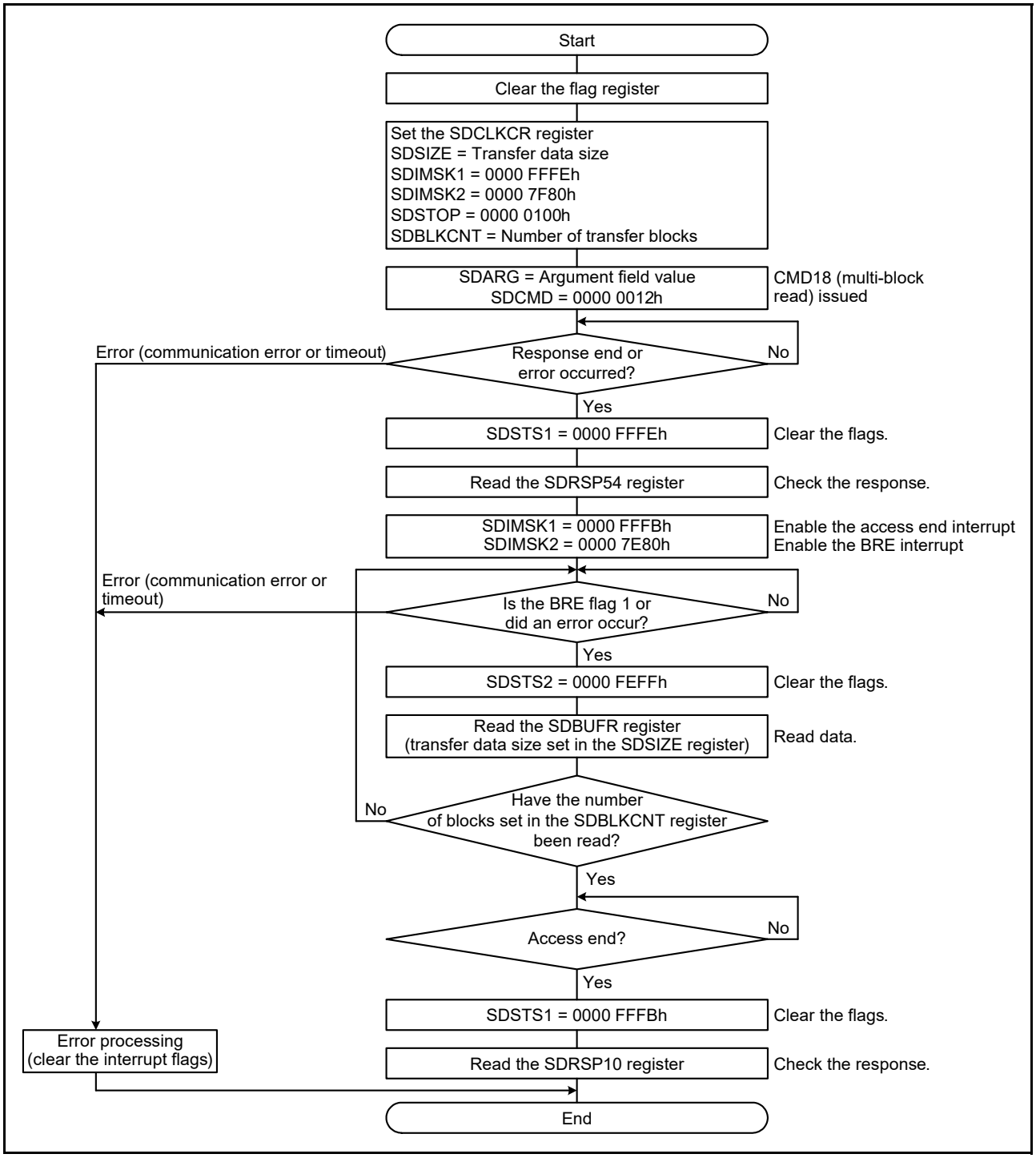


Figure 40.12 Issuing the Multi-Block Read Command

Downloaded from Arrow.com.

40.3.6.6 Multi-Block Write Command (CMD25)

Figure 40.13 shows an example of issuing the multi-block write command (CMD25).

1. Set the flags in registers SDSTS1 and SDSTS2 to 0.
2. Set the SDHI clock in the SDCLKCR register, and set the interrupt requests to be masked in registers SDIMSK1 and SDIMSK2. Refer to section 40.5.5 for details on setting the SDCLKCR register. Set the SDSTOP.SDBLKCNTEN bit to 1, and set the number of transfer blocks in the SDBLKCNT register.
3. After setting the argument field value for CMD25 to the SDARG register, write 0000 0019h to the SDCMD register. The SDHI issues CMD25 when a value is written to the SDCMD register.
4. When the response is received, the SDSTS1.RSPEND flag becomes 1, and the response end interrupt request is generated.
5. Set the SDSTS1.RSPEND flag to 0 and read the response stored in the SDRSP54 register. If the read response is in error, set the SDSTOP.STP bit to 1, and the command sequence can be stopped. When the SDSTOP.STP bit is set to 1, the SDHI automatically issues CMD12, and the response is received. At this point, the SDSTS1.ACEND flag becomes 1, and if the access end interrupt request is enabled, the access end interrupt request is generated. Next, set the SDSTS1.ACEND flag to 0 and read the response.
6. After the response is received, configure the SDIMSK1 register to enable the access end interrupt request, and configure the SDIMSK2 register to enable the BWE interrupt request.
7. When the SDBUFR register becomes write accessible, the SDSTS2.BWE flag becomes 1, and the BWE interrupt request is generated.
8. Set the SDSTS2.BWE flag to 0, and write the amount of data set in the SDSIZE.LEN[9:0] bits to the SDBUFR register. After writing to the SDBUFR register, and after the SDHI transmits write data to the SD card, the CRC status token is received, and the SDHI_D0 pin line becomes busy (low). The write access to the SDBUFR register and CRC status token reception repeat for the amount of transfer blocks set in the SDBLKCNT register. Also, after writing to the SDBUFR register, data transmission may cause a communication error or timeout to occur. After the amount of transfer blocks set in the SDBLKCNT register have been written, the SDHI automatically issues CMD12, and the response is received. At that time, the SDHI automatically writes 0000 0000h to the SDARG register.
9. When all blocks have been transmitted and the CRC status token is received, the SDHI exits the busy state, the SDSTS1.ACEND flag becomes 1, and the access end interrupt request is generated.
10. Set the SDSTS1.ACEND flag to 0 and read the response.

Perform error processing (clear the interrupt flag) if a communication error or timeout occurs.

Downloaded from Arrow.com.

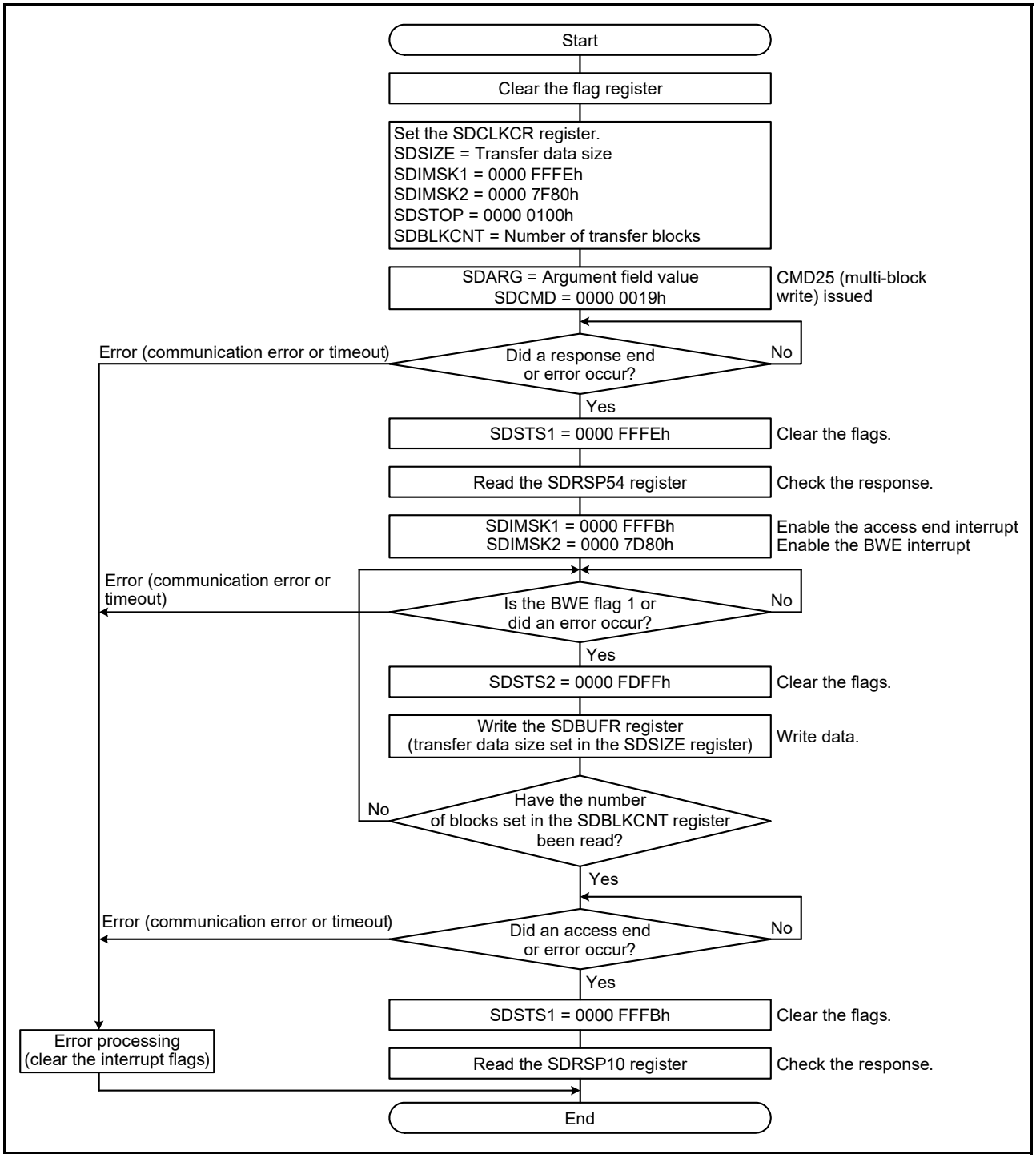


Figure 40.13 Issuing the Multi-Block Write Command

Downloaded from Arrow.com.

40.3.6.7 IO_RW_DIRECT Command (CMD52)

Figure 40.14 shows an example of issuing the IO_RW_DIRECT command (CMD52).

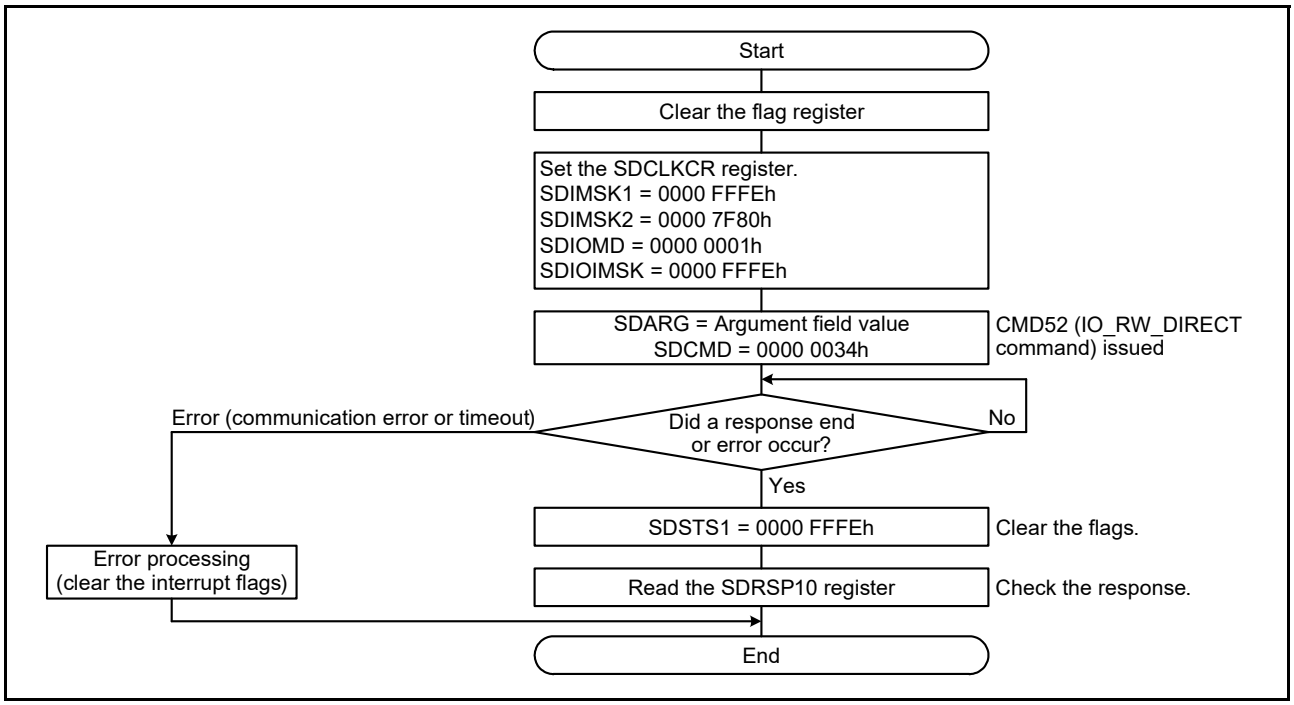


Figure 40.14 Issuing the IO_RW_DIRECT Command

Downloaded from Arrow.com.

40.3.6.8 IO_RW_EXTENDED Command (CMD53 (Multi-Block Read))

Figure 40.15 shows an example of issuing the IO_RW_EXTENDED command (CMD53/Multi-block read).

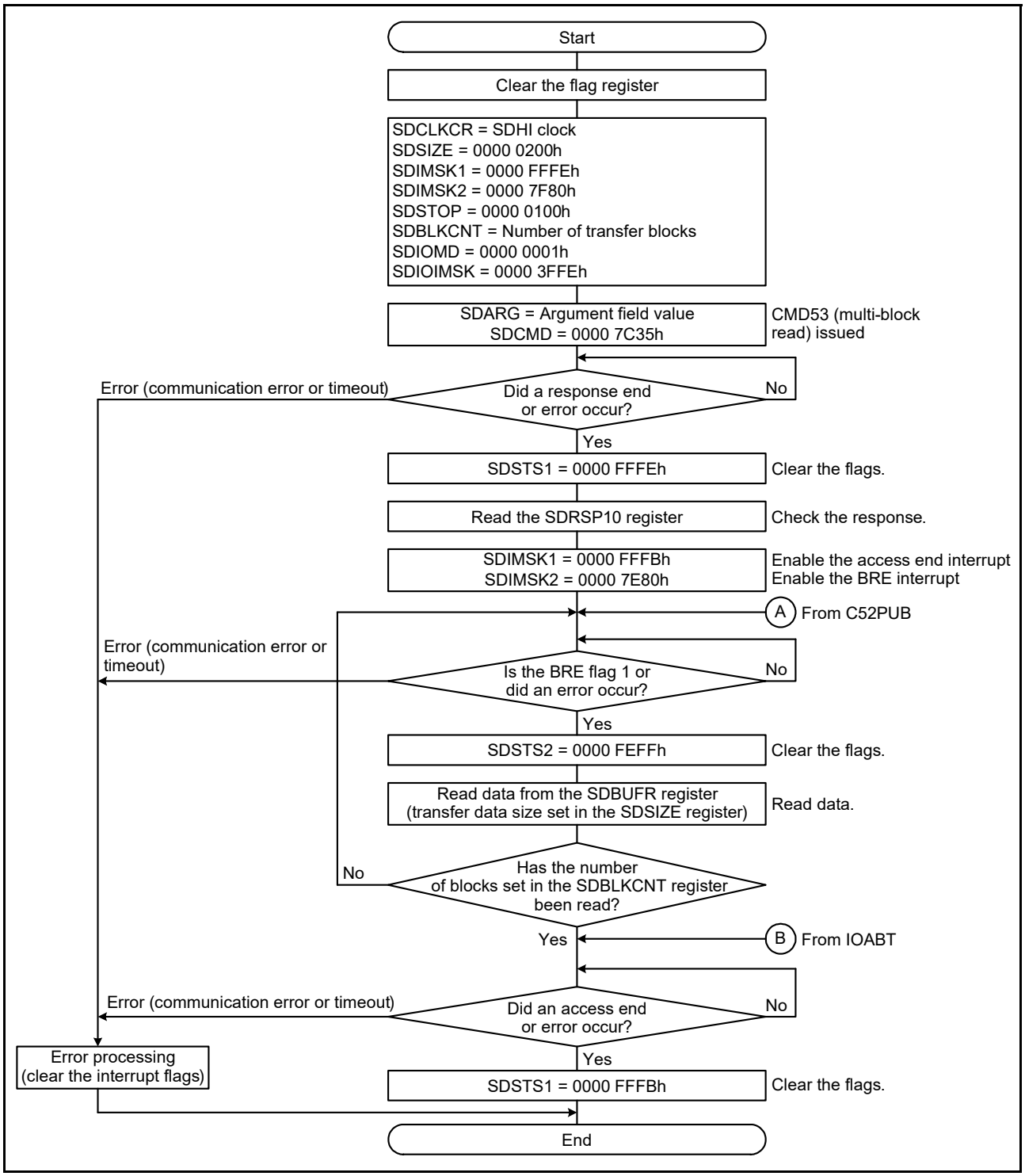


Figure 40.15 Issuing the IO_RW_EXTENDED Command (CMD53/Multi-Block Read)

Downloaded from Arrow.com.

Figure 40.16 shows an example of entering the read wait state and then issuing the SDIO none abort command (CMD52) during the IO_RW_EXTENDED command (CMD53/Multi-block read).

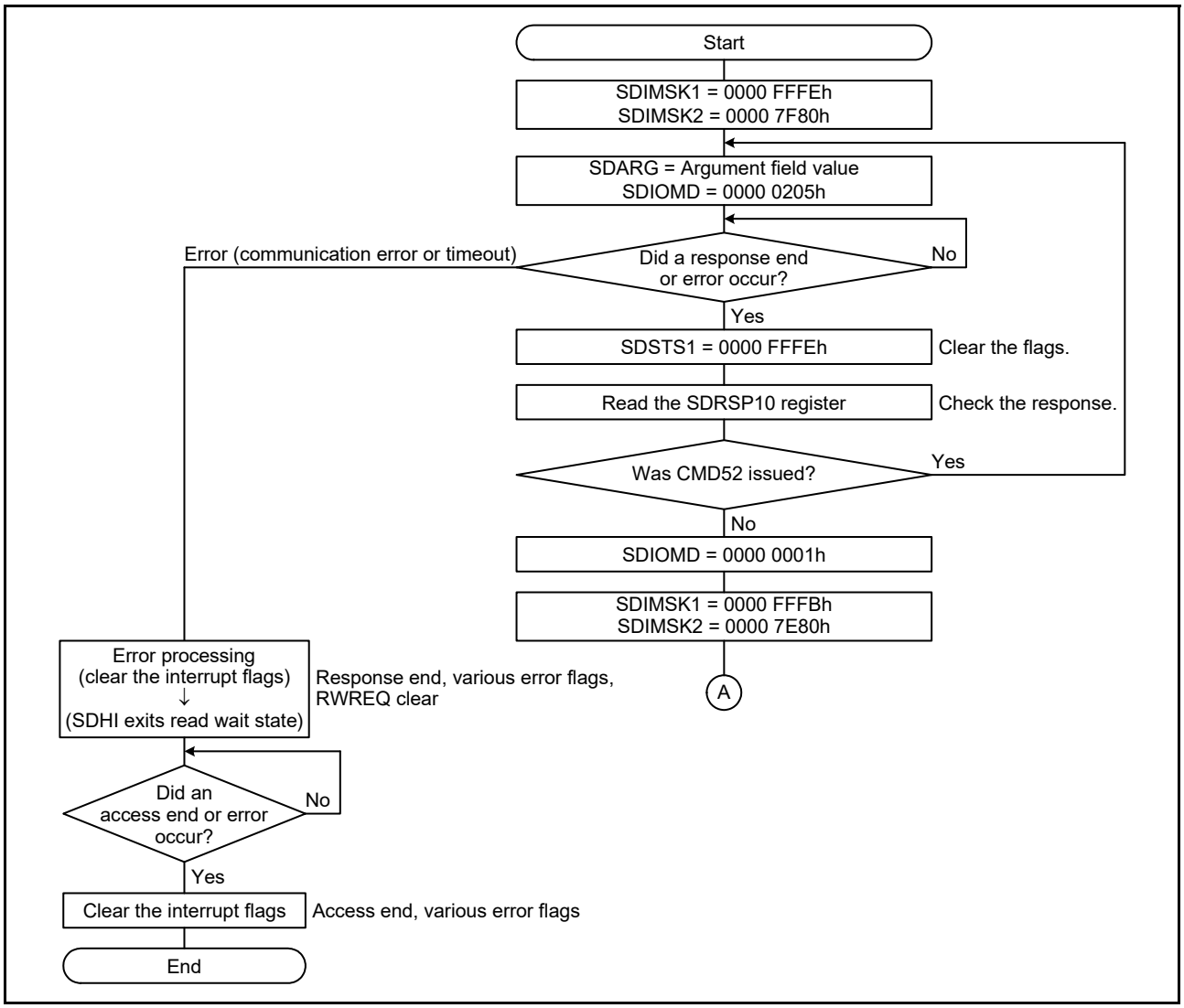


Figure 40.16 Entering the Read Wait State and Then Issuing the SDIO None Abort Command (CMD52) During the IO_RW_EXTENDED Command (CMD53/Multi-Block Read)

Figure 40.17 shows an example of SDIO abort (CMD52) issued during IO_RW_EXTENDED command (CMD53/ Multi-block read) sequence.

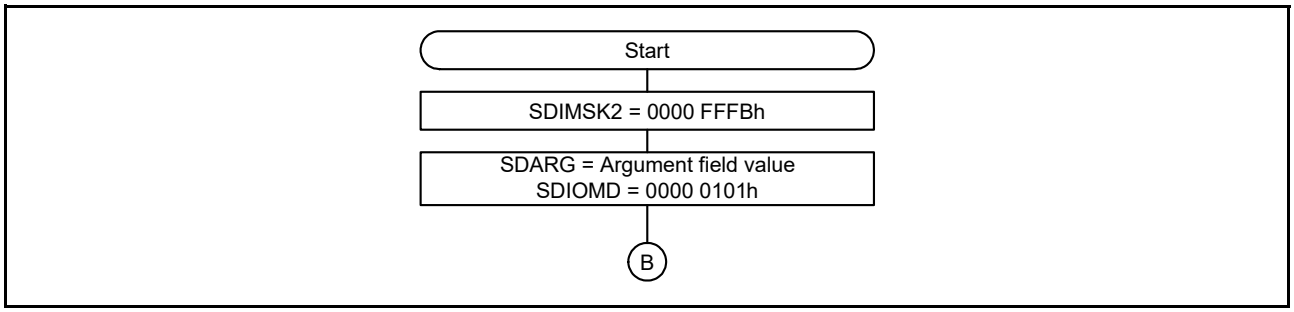


Figure 40.17 SDIO Abort (CMD52) Issued During IO_RW_EXTENDED Command (CMD53/Multi-Block Read) Sequence

Downloaded from Arrow.com.

40.3.6.9 IO_RW_EXTENDED (CMD53 Multi-Block Write)

Figure 40.18 shows an example of issuing the IO_RW_EXTENDED command (CMD53/Multi-block write).

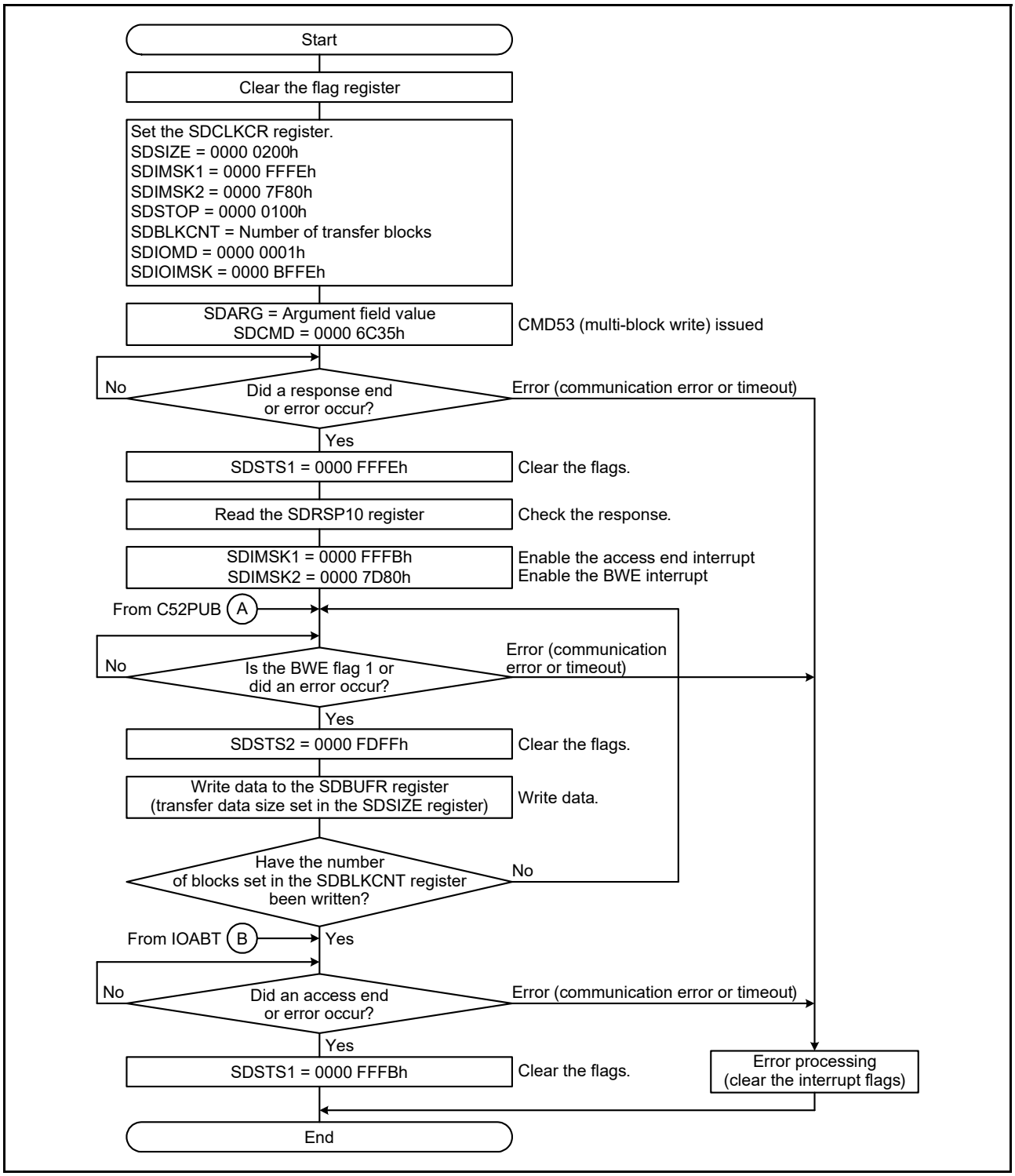


Figure 40.18 Issuing the IO_RW_EXTENDED Command (CMD53/Multi-Block Write)

Downloaded from Arrow.com.

Figure 40.19 shows SDIO none abort (CMD52) issued during IO_RW_EXTENDED command (CMD53/Multi-block write) sequence.

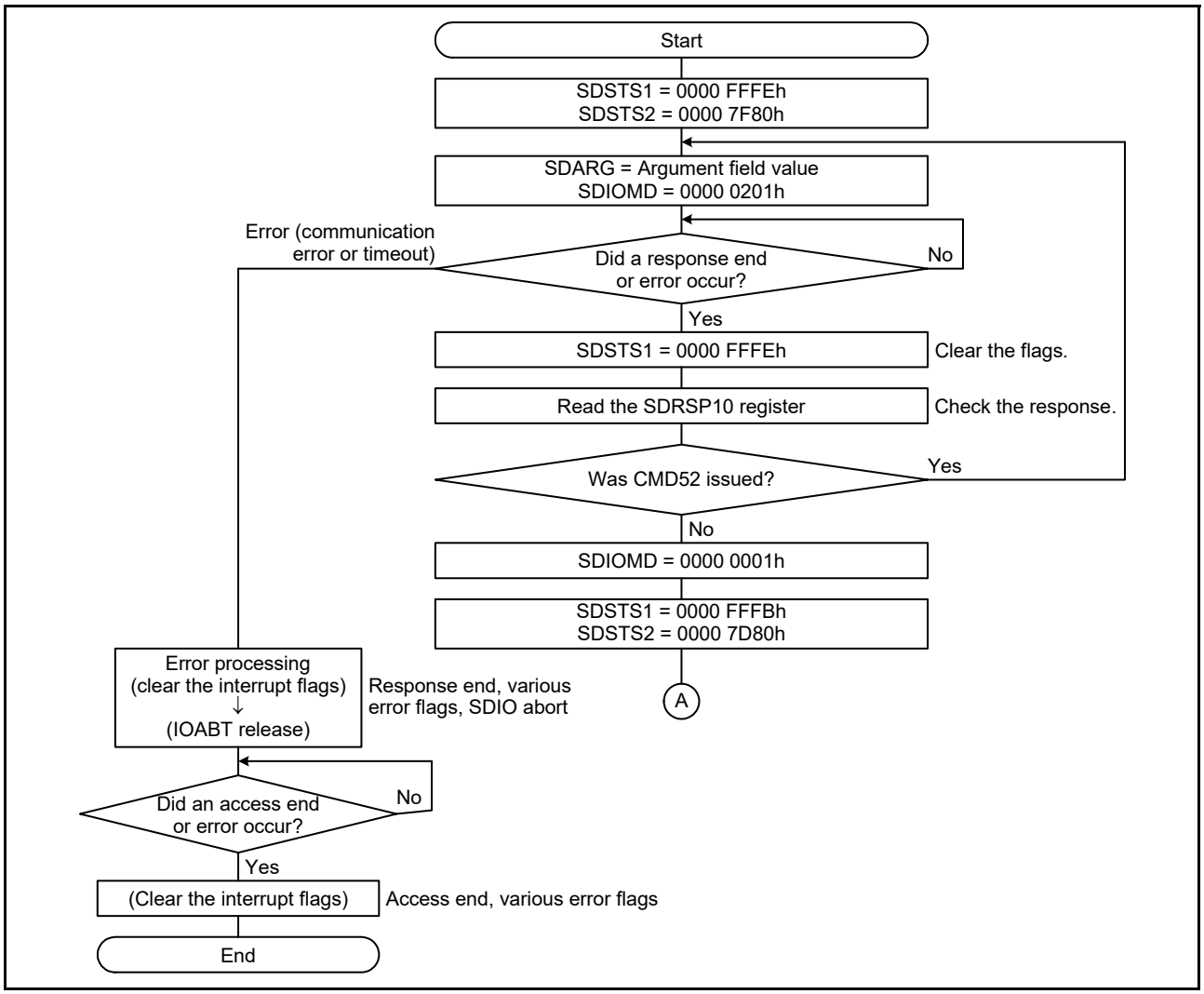


Figure 40.19 SDIO None Abort (CMD52) Issued During IO_RW_EXTENDED Command (CMD53/Multi-Block Write) Sequence

Figure 40.20 shows SDIO abort (CMD52) issued during IO_RW_EXTENDED command (CMD53) sequence.

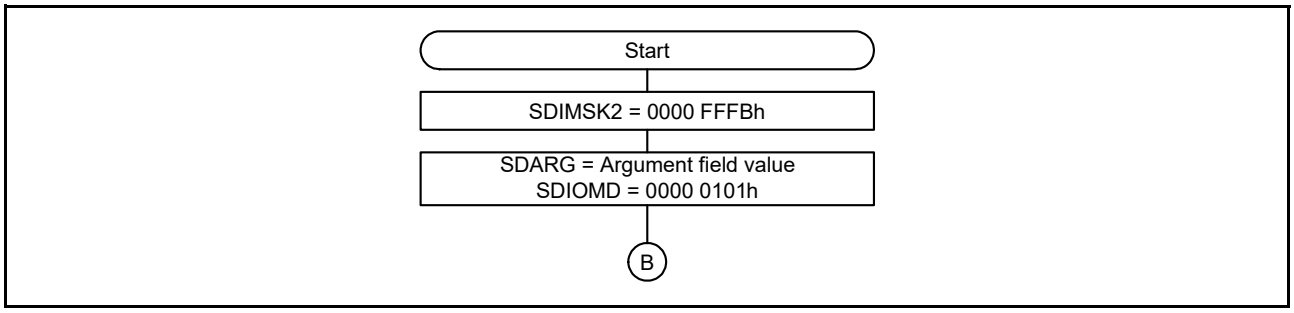


Figure 40.20 SDIO Abort (CMD52) Issued During IO_RW_EXTENDED Command (CMD53) Sequence

Downloaded from Arrow.com.

40.3.6.10 DMA Transfer

Figure 40.21 shows an example of data being transferred from the SDBUFR register after the CMD18 multi-block read command is issued.

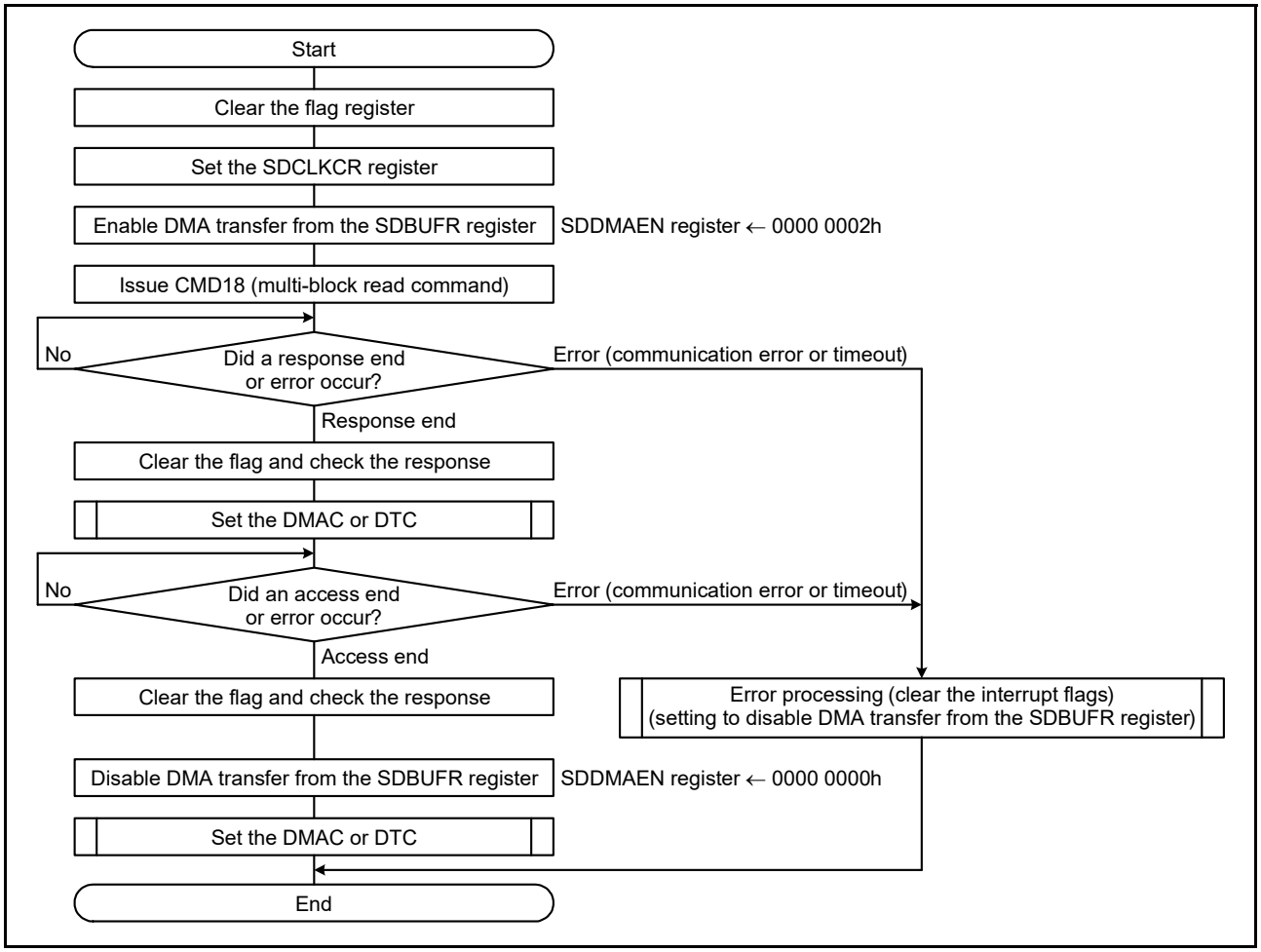


Figure 40.21 DMA Transfer After CMD18 is Issued

Downloaded from Arrow.com.

Figure 40.22 shows an example of data being DMA transferred to the SDBUFR register after the CMD25 multi-block write command is issued.

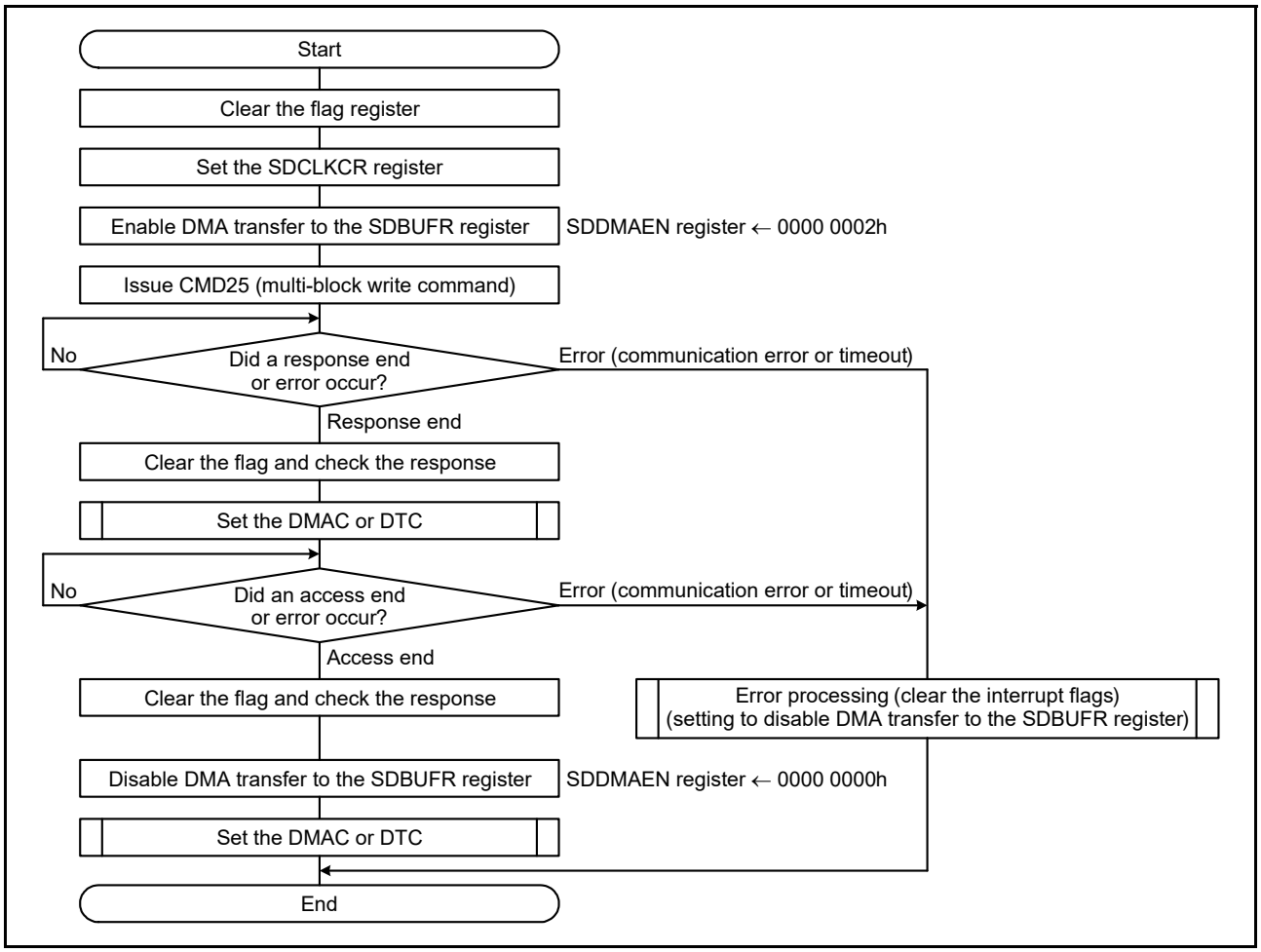


Figure 40.22 DMA Transfer After CMD25 is Issued

Downloaded from Arrow.com.

40.4 Interrupts

Table 40.8 lists the SDHI interrupt sources. When the status flags in registers SDSTS1, SDSTS2, and SDIOSTS become 1, if the corresponding bits in registers SDIMSK1, SDIMSK2, and SDIOIMSK are 0, the SDHI requests an interrupt. When clearing the status flags in registers SDSTS1, SDSTS2, and SDIOSTS, write 0 to the status flags to be cleared and write 1 to the states flags not being cleared.

Table 40.8 Interrupt Sources

Interrupt Source	Status Flag Register		Interrupt Mask/Enable Register		DMAC/DTC Triggerable
	Register symbol	Bit symbol	Register symbol	Bit symbol	
CACI	SDSTS1	ACEND	SDIMSK1	ACENDM	No
		RSPEND		RSPENDM	
	SDSTS2	ILA	SDIMSK2	ILAM	
		BWE		BWEM	
		BRE		BREM	
		RSPTO		RSPTOM	
		ILR		ILRM	
		ILW		ILWM	
		DTO		DTTOM	
		ENDE		ENDEM	
		CRCE		CRCEM	
		CMDE		CMDEM	
SDACI	SDIOSTS	EXWT	SDIOIMSK	EXWTM	No
		EXPUB52		EXPUB52M	
		IOIRQ		IOIRQM	
CDETI	SDSTS1	SDD3IN	SDIMSK1	SDD3INM	No
		SDD3RM		SDD3RMM	
		SDCDIN		SDCDINM	
		SDCDRM		SDCDRMM	
SBFAI	SDSTS2	BWE	SDDMAEN	DMAEN	Yes
		BRE			

Downloaded from Arrow.com.

40.4.1 DMA Transfer Triggered by Interrupt Requests

When the SBFAI interrupt is requested, DMA/DTC transfer can be used to write to or read the SDBUFR register. When using the SBFAI interrupt, set the SDDMAEN.DMAEN bit to 1, the SDIMSK2.BWEM bit to 1, and SDIMSK2.BREM bit to 1.

When the SDDMAEN.DMAEN bit is 1, if a write command is issued, the SDSTS2.BWE flag becomes 1; if a read command is issued, the SDSTS2.BRE flag becomes 1. At this point, the SBFAI interrupt request is output. When the last data of a block is transferred (one block is the transfer data size set in the SDSIZE.LEN[9:0] bits), the SBFAI interrupt request is cleared, and the SDSTS2.BWE flag or the SDSTS2.BRE flag becomes 0.

The SBFAI interrupt request is also cleared by following:

- The SDRST.SDRST bit is set to 0 (SDHI software reset).
- The SDSTOP.STP bit is set to 1.
- The SDIOMD.IOABT bit is set to 1.
- The SDDMAEN.DMAEN bit is set to 0.
  However, if the DMAEN bit is set to 1 again before the next command is written to the SDCMD register, the SBFAI interrupt request is output again.

The SBFAI interrupt request will not be cleared when a communication error or timeout occurs during DMA transfer. Perform error processing by software.

The SDSTS2.BWE flag and BRE flag will not become 0 when a communication error or timeout occurs, nor will they become 0 when the SDSTOP.STP bit and SDIOMD.IOABT bit are set to 1. If the SDSTS2.BWE flag or BRE flag remain set to 1, even if a write command or read command is issued, the SBFAI interrupt request will not be output, and the SDSTS2.BWE flag and BRE flag must be set to 0 before issuing the next command.

Table 40.9 lists the DMAC and DTC settings when performing DMA transfer.

Table 40.9 DMAC and DTC Settings When Performing DMA Transfer

Item		Setting Description
Transfer mode		Block transfer mode
Transfer data	1 data	32 bits
	Block size	Size set in the SDSIZE.LEN[9:0] bits divided by 4
Number of block transfers		Number of transfers set in the SDBLKCNT register

Downloaded from Arrow.com.

40.5 Notes on Using the SDHI

40.5.1 Illegal Read Access During a Multi-Block Read and How To Avoid It

When the multi-block read command (CMD18) is issued to read one or two blocks, if the response to CMD18 stored in the SDRSP10 register is read, the timing of the read access may cause the response to be read incorrectly.
Figure 40.23 shows examples of a normal read and an illegal read when using CMD18 to read two blocks.

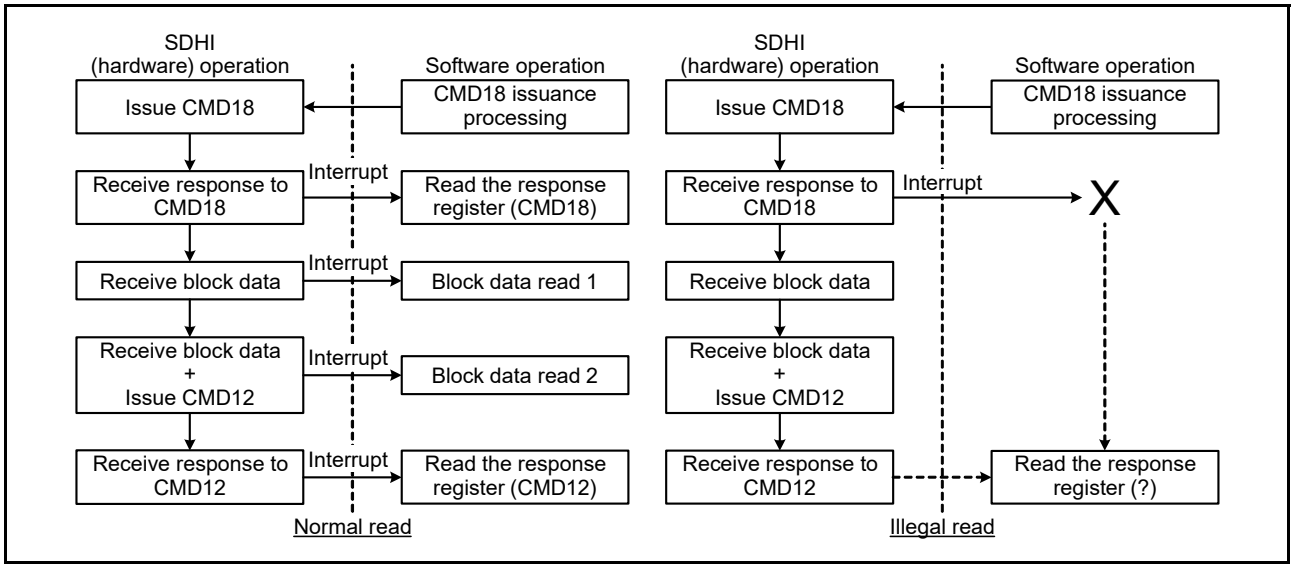


Figure 40.23 Multi-Block Read Processing When Reading Two Blocks

In the example of the illegal read, when the interrupt is generated by receiving the response to CMD18, the timing for reading the SDRSP10 register during the interrupt handling is delayed, and instead of the response for CMD18 being read, the response for CMD12 or the data during the response for CMD12 may be read.
This problem can be avoided by performing either of the following:

- When reading one block or two blocks, use a single block read command instead of a multi-block read command.
- When reading the response for CMD18, read the SDRSP54 register instead of the SDRSP10 register.

When using a multi-block read command to read three or more blocks, CMD12 will not be issued if no block data is read, which will prevent this problem from happening. Also, when using a multi-block write command, the above problem can be avoided by transmitting block data after reading the response to CMD25.

40.5.2 SDBUFR Register Illegal Write Error

When writing to the SDBUFR register after issuing a single block write command or a multi-block write command, write data for the size set by the SDSIZE.LEN[9:0] bits. If the amount of data written to the SDBUFR register is greater than the size set by the SDSIZE.LEN[9:0] bits, an illegal write error occurs in the SDBUFR register, and the SDSTS2.ILW flag becomes 1. However, the padding data included in the data being written to the SDBUFR register is ignored, so this error does not occur. For example, if the data size set by the SDSIZE.LEN[9:0] bits is an odd number, of the data written to the SDBUFR register, there is a remainder of 1 byte or 3 bytes. Although the extra data is written to the register, no error occurs. If the data size set by the SDSIZE.LEN[9:0] bits is an even number, and there is a remainder of 2 bytes, no error occurs even if these 2 bytes are written to the SDBUFR register.
If data written to the SDBUFR register is not transmitted, the SDSTS2.SDCLKCREN flag may remain set to 0. In this case, in order to set the SDSTS2.SDCLKCREN flag to 1, the SDRST.SDRST bit must be set to 0 and then set back to 1.

Downloaded from Arrow.com.

40.5.3 Automatic Control of the SDHI Clock Output

As per the SD card specifications, after MCU power-on, 74 cycles of the SDHI clock must be output from the host to the SD card before the card initialization command (CMD0) can be issued. Therefore, 74 cycles of the SDHI clock should be output from the SDHI to the SD card before enabling automatic control of the SDHI clock output.
When automatic control of the SDHI clock output is enabled, SDHI clock output stops if the command sequence is ended by a communication error or timeout. Therefore, if it is necessary to change the internal status of the SD card even after the command sequence ends, disable automatic control of the SDHI clock output and output the SDHI clock to the SD card.

40.5.4 Restrictions on Setting the C52PUB Bit During a Multi-Block Write Sequence

During a CMD53 multi-block write sequence, if the SDIOMD.C52PUB bit is set to 1, the SDHI issues CMD52 after the SD buffer becomes empty. To immediately issue CMD52, perform one of the procedures below to suspend writing to the SD buffer, and set the C52PUB bit to 1.

Procedure to suspend writing to the SD buffer when not performing DMA transfer (interrupt used)

1. Set the SDIMSK2.BWEM bit to 1 to disable the interrupt, and suspend writing to the SDBUFR register.
2. Set the SDIOMD.C52PUB bit to 1. Then, when the SD buffer becomes empty, the SDHI issues CMD52.
3. After receiving the response for CMD52, set the SDIMSK2.BWEM bit to 0 to enable the interrupt, and resume writing to the SDBUFR register.

Procedure to suspend writing to the SD buffer when performing DMA transfer

1. Configure settings to perform DMA transfer every [SDSIZE register setting value × *n* blocks], and suspend writing to the SDBUFR register before setting the SDIOMD.C52PUB bit (*n* = 1, 2, ...).
2. Set the SDIOMD.C52PUB bit to 1. Then, when the SD buffer becomes empty, the SDHI issues CMD52.
3. After receiving the response for CMD52, resume DMA transfer to the SDBUFR register.

40.5.5 Note on Setting the SDCLKCR Register

The SDCLKCR register cannot be written when the SDSTS2.SDCLKCREN flag is 0. Set the SDSTS2.SDCLKCREN flag to 1 before writing to the SDCLKCR register.

40.5.6 Writing to the SDSTOP Register During a Multi-Block Read Sequence

When the SDSTOP.SDBLKCNTEN bit is 1 during a multi-block read sequence, if the SDSTOP.STP bit is set to 1 and the command sequence is stopped, the command sequence may not be completed depending on when the SDSTOP.STP bit is set to 1. To avoid this problem, set the SDSTOP.STP bit and the SDSTOP.SDBLKCNTEN bit to 0 simultaneously. Note that at this time, the SDSTOP.SDBLKCNTEN bit should be set to 0 even if the SDSTS2.SDCLKCREN bit is 0.
If the SDSTOP.SDBLKCNTEN bit is not set to 0, the command sequence can be completed by setting the SDRST.SDRST bit to 0.
During a CMD53 multi-block transfer, when stopping data transfer by setting the SDIOMD.IOABT bit to 1, the SDSTOP.SDBLKCNTEN bit should remain set to 1.

40.5.7 Controlling Module Operation

SDHI operation is controlled by setting the MSTPCRD.MSTPD19 bit. Setting the MSTPD19 bit to 0 enables the SDHI; setting the MSTPD19 bit to 1 disables the SDHI. The SDHI is disabled after a reset. Registers in the SDHI can be accessed by setting the MSTPD19 bit to 0. Refer to section 11, Low Power Consumption for details.

Downloaded from Arrow.com.

41. Bluetooth Low Energy (BLE)

This MCU has a Bluetooth Low Energy (BLE), which consists of an RF transceiver compliant with Bluetooth 5.0 Low Energy (single mode), a link layer, and an RF transceiver power-supply.
The BLE is controlled by a Bluetooth middleware available from Renesas Electronics Corporation.

41.1 Overview

Table 41.1 lists the Specifications of the BLE. Figure 41.1 is a BLE Block Diagram.

Table 41.1 Specifications of the BLE

Item	Description		
Features	• RF transceiver compliant with the Bluetooth 5.0 Low Energy specification and a link layer • An on-chip matching circuit allows reducing the number of external parts • On-chip BLE-dedicated high-precision low-speed oscillator (32.768 kHz) • Transmission power is selectable as 0 dBm or +4 dBm. • The minimum receiving sensitivity is selectable from among –105 dBm (125 kbps), –100 dBm (500 kbps), –95 dBm (1 Mbps), or –92 dBm (2 Mbps).		
Bluetooth 5.0 functions	Classification	Function	Remark
	Device Address	Public or random address	The address can be set as a desired address.
	Advertising	Extended or periodic	
		Multiple advertising	Maximum number of sets: 4
		Advertising or Scan Response Data	Maximum data length = 1650 bytes
	Scanning	Passive, active, or periodic	Number of units for concurrent synchronization with periodic advertising = 2
		Whitelist or periodic advertiser list	Number of units registered in the whitelist: 4 Number of units registered in the periodic advertiser list: 4
	Master or slave	Data transmission or reception	Maximum payload length = 251 bytes MoreData function is supported. Master/slave multi-role function is supported.
	Other	Bit rates	125 kbps, 500 kbps, 1 Mbps, 2 Mbps Bit-rate combinations for transmission and reception can be set as desired.
		Frequency hopping	Channel Selection Algorithm #2
		Encryption circuit for Bluetooth	On-chip Bluetooth-dedicated AES-CCM (128 bits) circuit
Other functions	RF transceiver power-supply (DC-to-DC converter, and linear regulator) (Only the linear regulator is available for the 83-pin LGA product.)		
Other features	In addition to the functions listed above, the 83-pin LGA product also has the following features. • Small PCB trace antenna • Bluetooth-dedicated clock oscillator • Certified as compliant with radio-related laws (technical standards, FCC, ISED, and CE)		

Downloaded from Arrow.com.

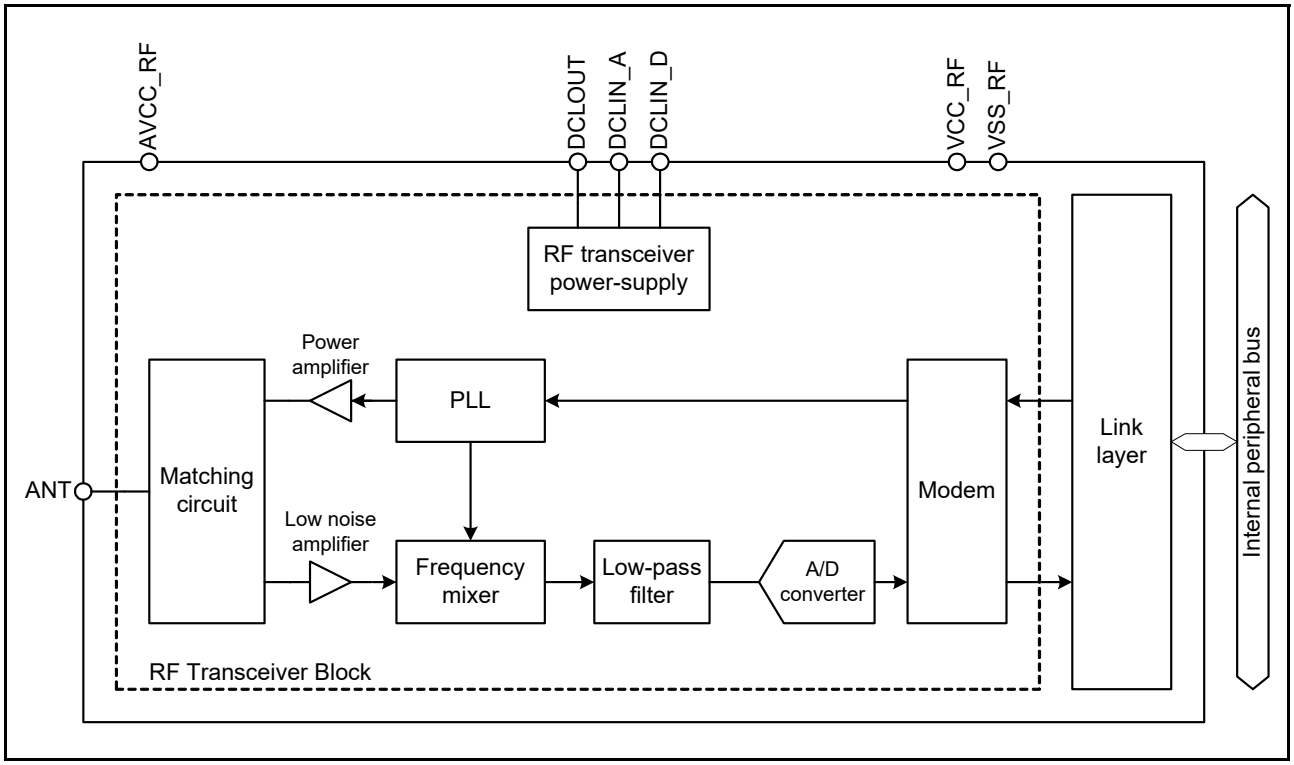


Figure 41.1 BLE Block Diagram

Table 41.2 lists the BLE I/O Pins.

Table 41.2 BLE I/O Pins

Pin name	I/O	Description
ANT	I/O	RF single I/O pin for RF transceiver Set the impedance of the signal line to 50 Ω.
DCLOUT	Output	Output pin for the RF transceiver power-supply
DCLIN_A, DCLIN_D	Input	If the DC-to-DC converter is to be selected as the power supply for the RF transceiver, connect an inductor and capacitor between the DCLOUT pin and the converter. If the linear regulator is to be selected as the power supply for the RF transceiver, connect a capacitor between the DCLOUT pin and the converter.
VCC_RF	Input	RF transceiver power-supply pin
AVCC_RF	Input	RF transceiver power-supply pin
VSS_RF	Input	RF transceiver ground pin

Downloaded from Arrow.com.

Figure 41.2 shows an example of the external connection circuit for the 83-pin LGA version of this MCU.

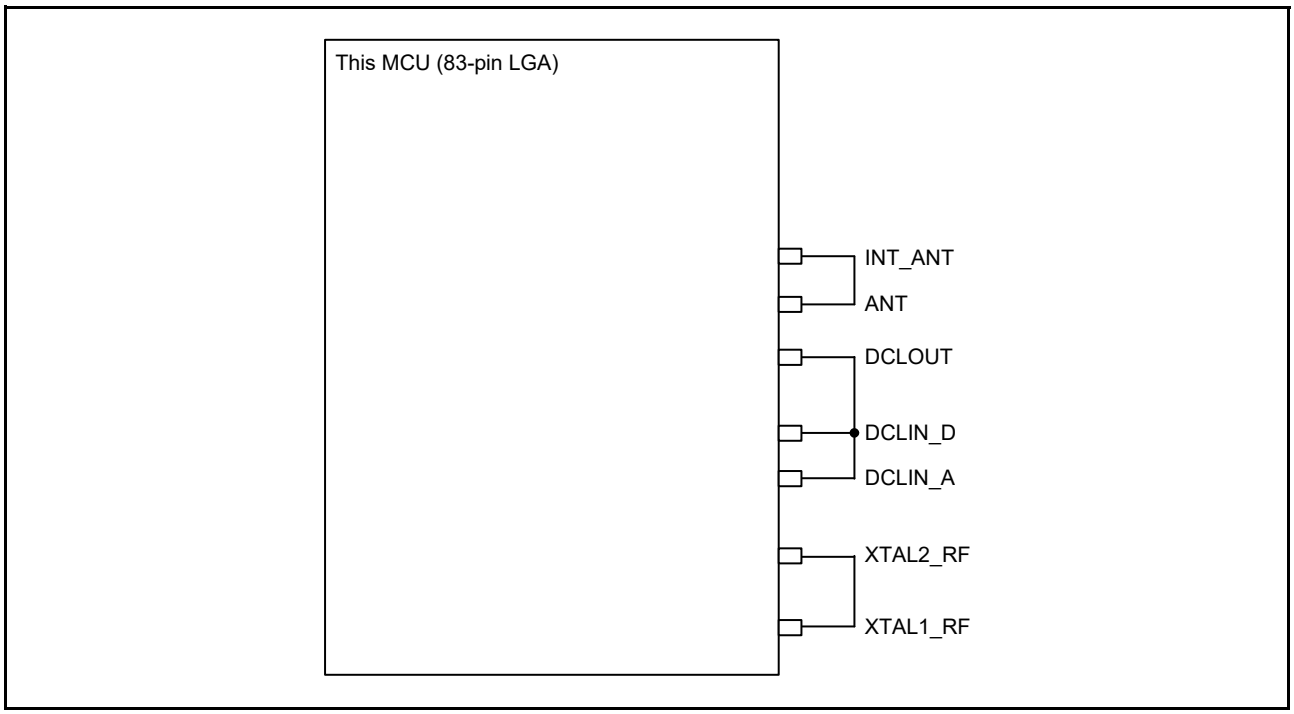


Figure 41.2 Example of the External Connection Circuit (83-Pin LGA)

Downloaded from Arrow.com.

Table 41.3 lists the Bluetooth Low Energy functions that this MCU supports. The use of these functions requires the Bluetooth middleware.

Table 41.3 List of Supported Bluetooth Low Energy Functions

Function	Bluetooth® core spec
Low Energy Controller (PHY and LL)	v4.0
Low Energy Host (L2CAP and Security Manager)	
Attribute Protocol and Generic Attribute Profile	
Appearance Data Type	v4.1
Low Duty Cycle Directed Advertising	
32-bit UUID Support in LE	
LE L2CAP Connection Oriented Channel Support	
LE Link Layer Topology	
LE Ping	
LE Data Packet Length Extension	v4.2
LE Secure Connections	
Link Layer Privacy	
Link Layer Extended Filter Policies	
LE 2M PHY	v5.0
LE Coded PHY	
High Duty Cycle Non-Connectable Advertising	
LE Advertising Extensions	
LE Channel Selection Algorithm #2	

Downloaded from Arrow.com.

41.2 Operation

41.2.1 State Transitions

Figure 41.3 is the State Transition Diagram of the BLE.

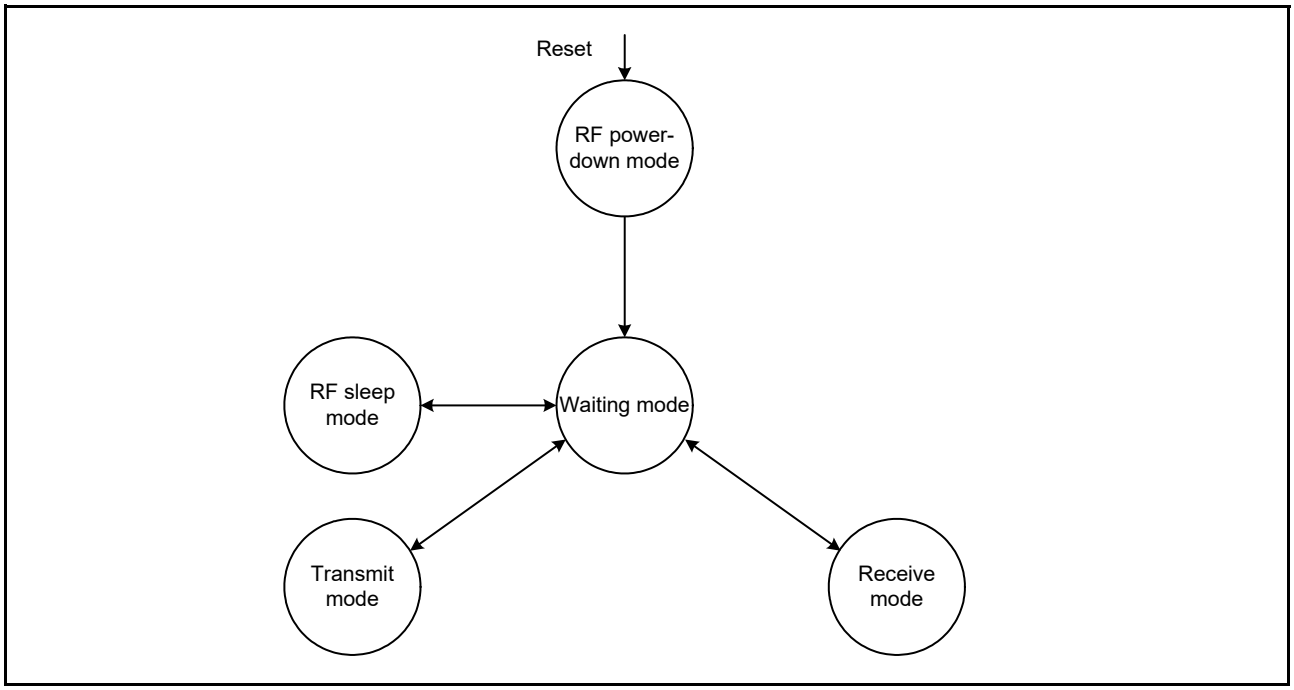


Figure 41.3 State Transition Diagram of the BLE

RF Power-Down Mode

The RF power-down mode is the initial mode of the BLE after release from the reset state.
Power is supplied to the MCU, but not to the RF transceiver.

Waiting Mode

In waiting mode, the BLE waits until the transmission or reception of data or a transition to sleep mode is requested.

RF Sleep Mode

The RF sleep mode is a low-power operating mode and the power supply, except to part of the link-layer circuit, is stopped.

Transmit Mode

The transmit mode is for the transmission of data.
The mode shifts to waiting mode after the data have been transmitted.

Receive Mode

The receive mode is for the reception of data.
The mode shifts to waiting mode after the data have been received.

Downloaded from Arrow.com.

41.3 Interrupts

Table 41.4 shows the interrupt sources.
The Bluetooth middleware executes processing in response to these interrupts.
Do not set the ICU.IERm.IENj bits for these interrupts to 0.

Table 41.4 BLE Interrupt Sources

Name	DTC Activation	DMAC Activation
BLEIRQ	Not possible	Not possible
ERI	Not possible	Not possible
RXI	Possible	Possible
TXI	Possible	Possible
TEI	Not possible	Not possible

Downloaded from Arrow.com.

41.4 Certificates of Compliance

41.4.1 Radio-Related Laws

The 83-pin LGA version of this MCU complies with the radio laws listed below.

- Japan: Certificate of construction type

Model number	Construction type certificate number*1
R5F523W8CDLN	R 006-000937
R5F523W8DDLN	R 006-000940

- North America: FCC (FCC ID: 2AEMXRX23W8DLN), ISED (IC ID: 20194-RX23W8DLN)*1
- Europe: CE (RE)*2

Note 1. This device is too small to allow indication of the construction type certificate number, FCC ID, or ISED ID on its package.

Note 2. This is based on testing of the radio operation for compliance with the EU Radio Equipment Directive. The user will need to test end products for compliance before affixing the CE mark to them.

41.4.2 Bluetooth SIG Certification

Declaration ID: D044683
QDID: 134349 - End Product
Name: RX23W Module
Model number: R5F523W8xDLN

41.4.3 Labeling and Information RX23W Users Should Provide to End Users of Their Products

Note the following statements when incorporating the 83-pin LGA products into the end products.

This module should be installed in the host product according the installation manual.
The following statements must be described on the user manual of host device of this module;

Note to users in the United States of America

Caution:

Changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment.

Declaration of Conformity

This device complies with part 15 of FCC Rules. Operation is subject to the following two conditions: (1) this device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

Note to users

The following items must be followed when using this certification module:

List of applicable FCC rules

This module has been tested for compliance to FCC Part 15

Downloaded from Arrow.com.

Summarize the specific operational use conditions

The moduleis installed in end productformobile RF exposure use condition. Any other usage conditions such as co-location with other transmitter(s) or being used in a portable condition will need a separate reassessment through a class II permissive change application or new certification.

Limited module procedures

Any other usage conditions such as co location with other transmitter(s) will need a separate reassessment through a class II permissive change applica tion or new certification.

Trace antenna designs

It is strictly forbidden to use antenna except designated.

RF exposure considerations

This equipment complies with FCC mobile radiation exposure limits set forth for an uncontrolled environment. This equipment should be installed and operated with a minimum distance of 20cm between the radiator & your body. If the module is installed in a portable host, a separate SAR evaluation is required to confirm compliance with relevant FCC portable RF exposure rules.

Antennas

The antenna of this module is PCB type.

This module is certified only with the antenna on the board. No other antenna can be used.

Label and compliance information

The final end product must be labeled in a visible area with the following: “Contains FCC ID: 2AEMXRX23W8DLN”. The grantee's FCC ID can be used only when all FCC compliance requirements are met.

Information on test modes and additional testing requirements

This transmitter isinstalled in end product formobile RF exposure condition and any co-located or simultaneous transmission with other transmitter(s) use will require a separate class II permissive change re-evaluation or new certification.

Additional testing, Part 15 Subpart B disclaimer

This transmitter module is tested as a subsystem and its certification does not cover the FCC Part 15 Subpart B (unintentional radiator) rule requirement applicable to the final host. The final host will still need to be reassessed for compliance to this portion of rule requirements if applicable.

Note to users in the United States of America and Canada

Note to users

It is strictly forbidden to use antenna except designated.

This equipment must not be co-located or operated in conjunction with any other antenna or transmitter.

[For mobile equipment]

This equipment complies with FCC/IC radiation exposure limits set forth for an uncontrolled environment and meets the FCC radio frequency (RF) Exposure Guidelines and RSS-102 of the IC radio frequency (RF) Exposure rules. This equipment has very low levels of RF energy that is deemed to comply without maximum permissive exposure evaluation (MPE).

Downloaded from Arrow.com.

Note to users in Canada

Cet équipement est conforme aux limites d’exposition aux rayonnements énoncées pour un environnement non contrôlé et respecte les règles les radioélectriques (RF) de la FCC lignes directrices d'exposition et d’exposition aux fréquences radioélectriques (RF) CNR-102 de l’IC. Cet équipement émet une énergie RF très faible qui est considérée comme conforme sans évaluation de l’exposition maximale autorisée (MPE).

[For portable equipment]

This equipment complies with FCC/IC radiation exposure limits set forth for an uncontrolled environment and meets the FCC radio frequency (RF) Exposure Guidelines and RSS-102 of the IC radio frequency (RF) Exposure rules. This equipment has very low levels of RF energy that is deemed to comply without testing of specific absorption rate (SAR).

Note to users in Canada

Cet équipement est conforme aux limites d’exposition aux rayonnements énoncées pour un environnement non contrôlé et respecte les règles les radioélectriques (RF) de la FCC lignes directrices d'exposition et d’exposition aux fréquences radioélectriques (RF) CNR-102 de l’IC. Cet équipement émet une énergie RF très faible qui est considérée comme conforme sans évaluation du débit d’absorption spécifique (DAS).

Note to users in Canada

This device complies with Industry Canada’s licence-exempt RSSs. Operation is subject to the following two conditions: (1) this device may not cause interference; and (2) this device must accept any interference, including interference that may cause undesired operation of the device.

Remarque concermant les utilisateurs au Canada

Le présent appareil est conforme aux CNR d'Industrie Canada applicables aux appareils radio exempts de licence.

L'exploitation est autorisée aux deux conditions suivantes: (1) l'appareil ne doit pas produire de brouillage, et (2) l'utilisateur de l'appareil doit accepter tout brouillage radioélectrique subi, même si le brouillage est susceptible d'en compromettre le fonctionnement.

The following information must be indicated on the host device of this module;

[for FCC]

Contains Transmitter Module FCC ID: 2AEMXRX23W8DLN.

This device complies with part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) this device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

[for ISED]

Contains IC: 20194-RX23W8DLN

European Community Compliance Statement

Note:

Hereby, Renesas Electronics Corporation, declares that this R5F523W8xDLN is in compliance with the essential requirements and other relevant provisions of Directive 2014/53/EU.

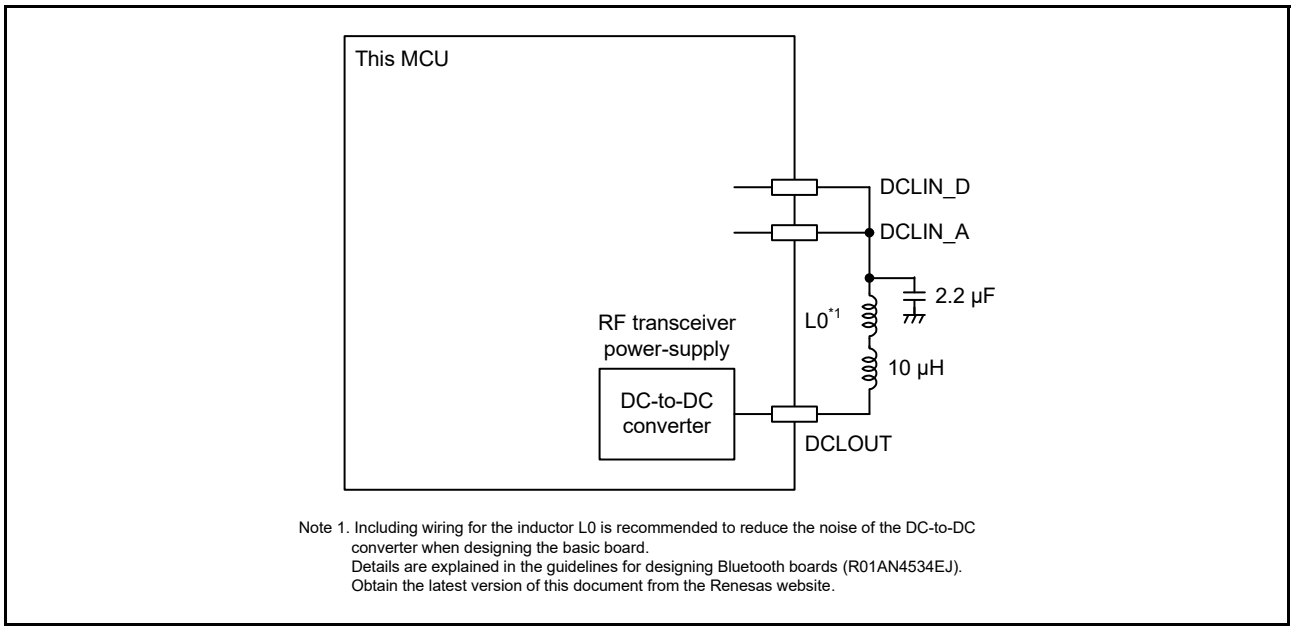
CE

Downloaded from Arrow.com.

41.5 Usage Notes

41.5.1 RF Transceiver Power-Supply

In the case of the 85-pin BGA and 56-pin QFN products, the RF transceiver power supply is selectable as either the DC-to-DC converter or linear regulator. Only the linear regulator is available for the 83-pin LGA product.
Figure 41.4 shows an example of the externally connected circuit for BLE when the DC-to-DC converter is selected (only available for the 85-pin BGA and 56-pin QFN products). Figure 41.5 and Figure 41.6 respectively show examples of the externally connected circuit for BLE when the linear regulator is selected for an 85-pin BGA or 56-pin QFN product and for an 83-pin LGA product.



Note 1. Including wiring for the inductor L0 is recommended to reduce the noise of the DC-to-DC converter when designing the basic board.
Details are explained in the guidelines for designing Bluetooth boards (R01AN4534EJ).
Obtain the latest version of this document from the Renesas website.

Figure 41.4 Example of the BLE Externally Connected Circuit When the DC-to-DC Converter is Selected (85-pin BGA, 56-pin QFN)

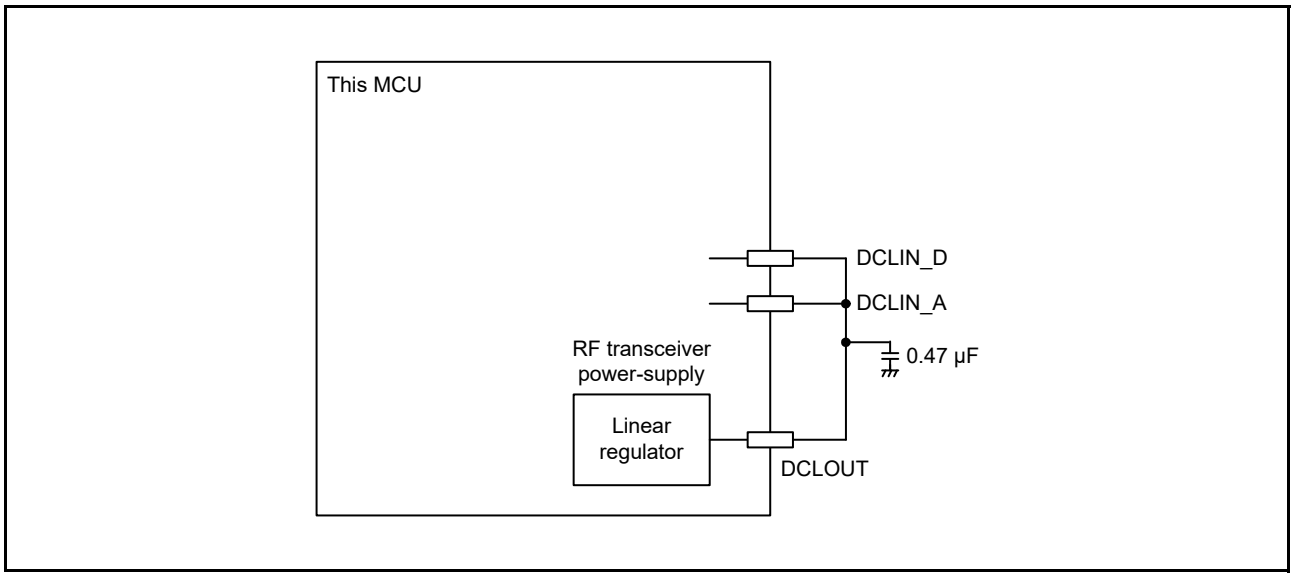


Figure 41.5 Example of the BLE Externally Connected Circuit When the Linear Regulator is Selected (85-pin BGA, 56-pin QFN)

Downloaded from Arrow.com.

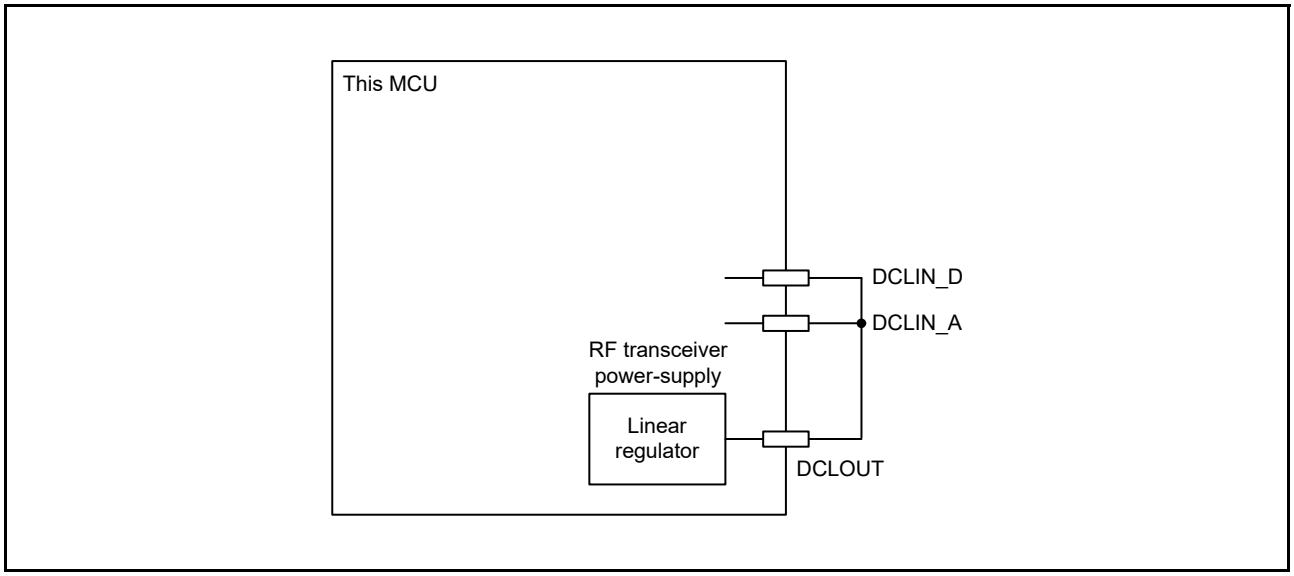


Figure 41.6 Example of the BLE Externally Connected Circuit (83-pin LGA)

Do not connect other pins or external circuits to the I/O pins (DCLOUT, DCLIN_D, or DCLIN_A) of the RF transceiver power-supply whether the DC to DC converter or linear regulator is selected.

41.5.2 Wireless Standards

International standards and domestic regulations stipulate the use of wireless receivers and transmitters.
Use the device in compliance with the standards or regulations of the country in which the device is being used.

Main standards applicable to the 2.4 GHz band:

- Japan: ARIB STD-T66
- U.S.A.: FCC CFR Title 47 parts 15.247 and 15.249
- Europe: EN 300 328

41.5.3 Notes on Board Design

The applicability of the notes on board design differs with the wireless standard to be employed.
Notes on board design are explained in more detail in the guidelines for designing Bluetooth boards (R01AN4534).
Obtain the latest version of this document from the Renesas website.

Downloaded from Arrow.com.

42. Trusted Secure IP (TSIP-Lite)

This MCU incorporates a Trusted Secure IP Lite (TSIP-Lite) module to provide security functions. The module consists of an access management circuit, encryption engine, and random number generator. In combination with the TSIP-Lite library, the TSIP-Lite can prevent eavesdropping (confidentiality), falsification of information (integrity), and impersonation (authenticity).
Key information to be used in encrypting and decrypting data is only stored within the TSIP-Lite, and any external access can be shut out to obtain a system with strong security.

42.1 Overview

Table 42.1 summarizes the specifications of the TSIP-Lite. Figure 42.1 shows a block diagram of the TSIP-Lite.

Table 42.1 Specifications of TSIP-Lite

Item	Description
Access control	Access management circuit • In case of irregular access to the TSIP-Lite due to a falsified program or runaway execution of a program, this circuit blocks all subsequent access and stops the output of data from the TSIP-Lite.
Encryption engine	AES: Compliant with NIST FIPS PUB 197 algorithm • Key sizes: 128 or 256 bits • Block sizes: 128 bits • Block cipher mode of operation ECB, CBC, CTR: Compliant with NIST SP 800-38A CMAC: Compliant with NIST SP 800-38B CCM: Compliant with NIST SP 800-38C GCM: Compliant with NIST SP 800-38D XTS: Compliant with NIST SP 800-38E GCTR • Number of cycles for execution*1 ECB, CBC, CTR, CMAC, GCTR, XTS: 44 cycles of PCLKB for 128-bit keys, 61 cycles of PCLKB for 256-bit keys CCM: 88 cycles of PCLKB for 128-bit keys AES-GCM • AES-GCM is realized by combining AES-GCTR and GHASH. Key management • Keys are only valid within the TSIP-Lite. • Only key generation information is output from the TSIP-Lite. • Keys can be regenerated by the input of key generation information to the TSIP-Lite. Endian • Big or little
Generation of random numbers	32-bit true random number generator • The TSIP-Lite library can assemble 32-bit true random numbers to generate 128- or 256-bit true random numbers. • The generated 128-bit and 256-bit true random numbers are used as keys in encrypting and decrypting data.
Protection against illicit key copying	• An ID unique to the MCU (unique ID) is accessible from the access management circuit through the dedicated bus. • Combining the unique ID with the key generation information prevents the illicit copying of the key to another MCU.
Supervisor mode	• The supervisor mode signal is connected to the access management circuit and is used to limit control of the TSIP-Lite module to supervisor mode only.
Interrupt sources	Three These can be used as triggers for data transfer by the DMAC or DTC.
Low power consumption	Setting of the module stop state is possible.

Note 1. This does not include the overhead for calling functions of the TSIP-Lite library.

Downloaded from Arrow.com.

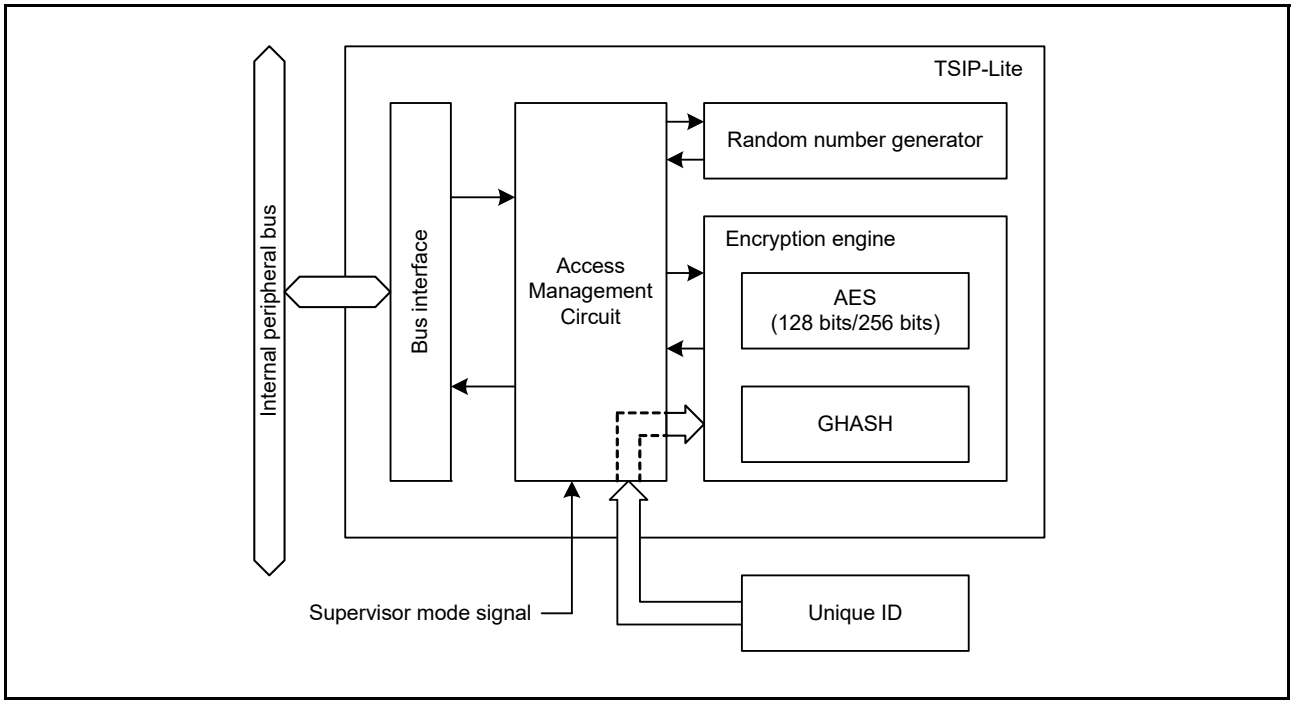


Figure 42.1 TSIP-Lite Block Diagram

Downloaded from Arrow.com.

42.2 Operation

42.2.1 Operating Modes and State Transitions

Figure 42.2 shows the state transitions of the TSIP-Lite.
Use of the TSIP-Lite security functions is only possible through use of the TSIP-Lite library provided by Renesas Electronics, in accordance with the state transitions as shown in the figure below.
When irregular access to the TSIP-Lite (access that violates the defined procedure) due to a falsified program or a program entering runaway execution, etc. is attempted, the access management circuit does not accept any subsequent access and stops the output of any data from the TSIP-Lite.

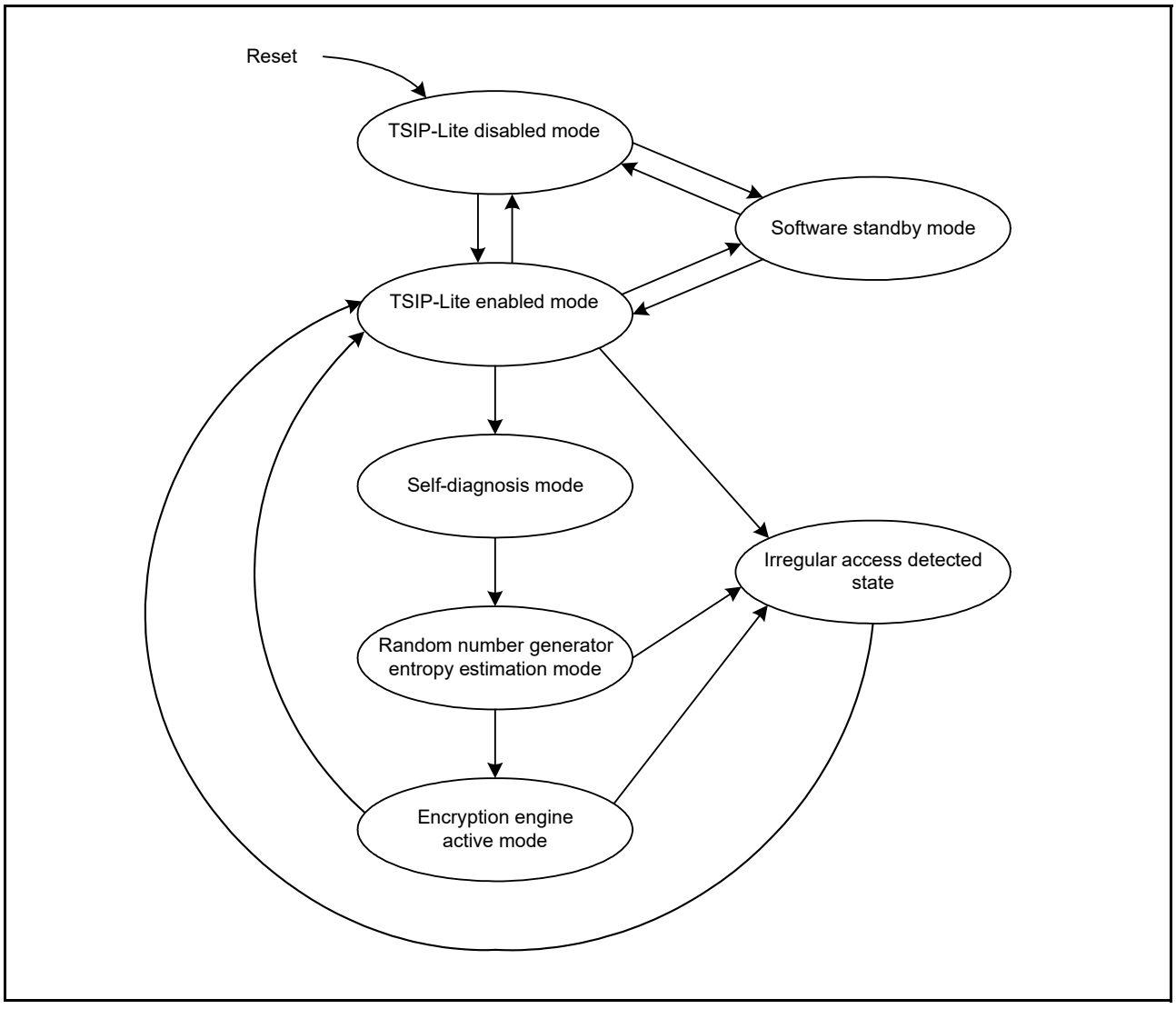


Figure 42.2 TSIP-Lite Operating Modes and State Transitions

Many of the security functions that the TSIP-Lite offers are applicable only in the encryption engine active mode. The operations that can be performed in this mode are given below.

(1) Key Installation
(2) Encryption and decryption
(3) Key generation
(4) Random number generation

Downloaded from Arrow.com.

42.2.2 Encryption Engine

Figure 42.3 shows processes of the encryption engine integrated in the TSIP-Lite.
The encryption engine, using the key generation information, performs plaintext to ciphertext encryption and ciphertext to plaintext decryption by hardware.
In no part of the encryption or decryption process, is key data or intermediate data ever exposed outside of the TSIP-Lite.

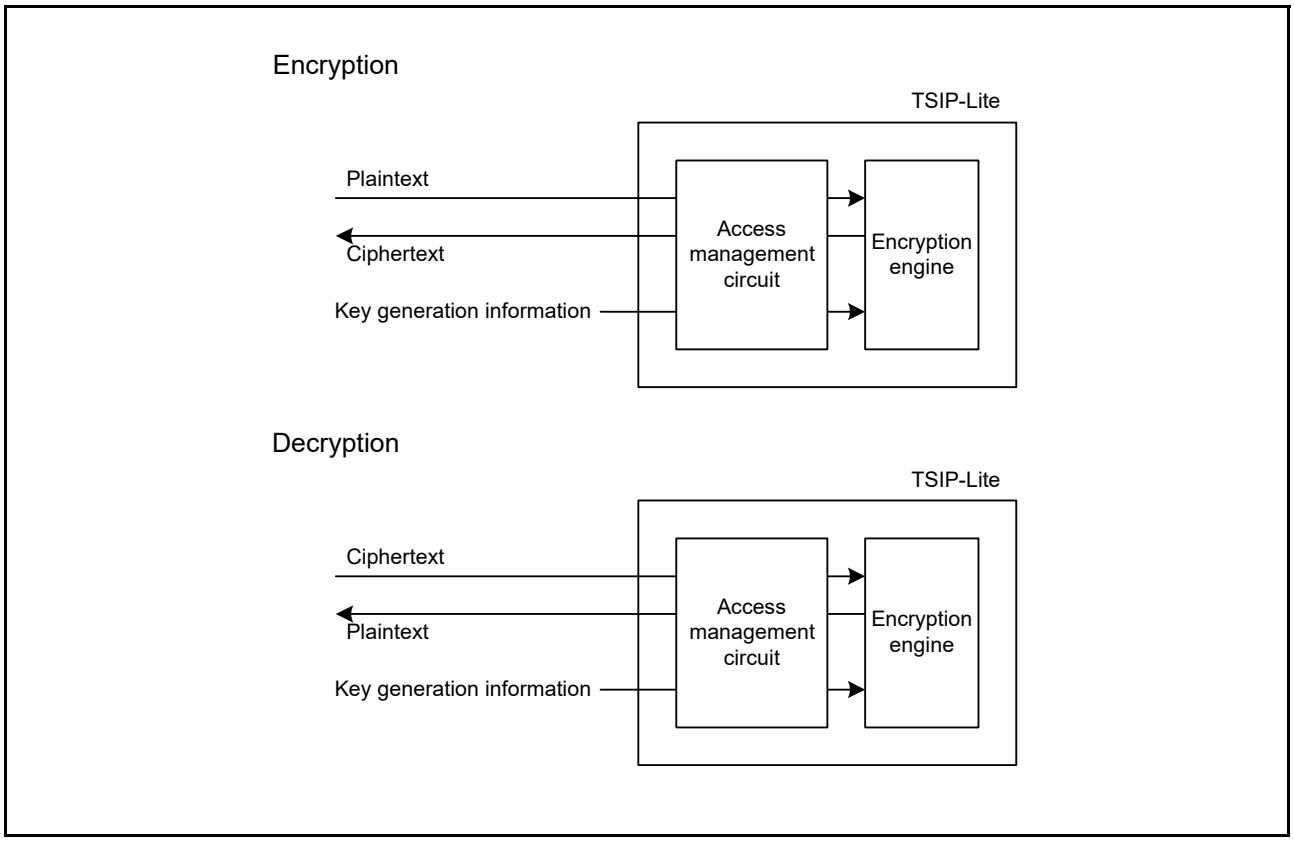


Figure 42.3 Encryption and Decryption processes by Encryption Engine

Downloaded from Arrow.com.

42.2.3 Key Installation

The key installation is the operation that safely converts the user key to the key generation information and stores it in flash memory. The procedure for installing the key data are given below.

(1) The user uses the key (Key-2) used for encrypting the user key to encrypt the user key (Key-1) producing eKey-1.
(2) The user sends the encrypted user key (eKey-1) to the TSIP-Lite over the serial interface.
(3) The key generation information of the Key-2 (Index-2) contained in the TSIP-Lite library is used to recover the Key-2, which is then used to decrypt the user key.
(4) The user key is converted to user key generation information (Index-1) using the unique ID and a random number, and stored in flash memory.

The installation process and flow chart are given in Figure 42.4 and Figure 42.5, respectively.
Once the key data is installed, the user key generation information (Index-1) can then be used to perform encryption or decryption.

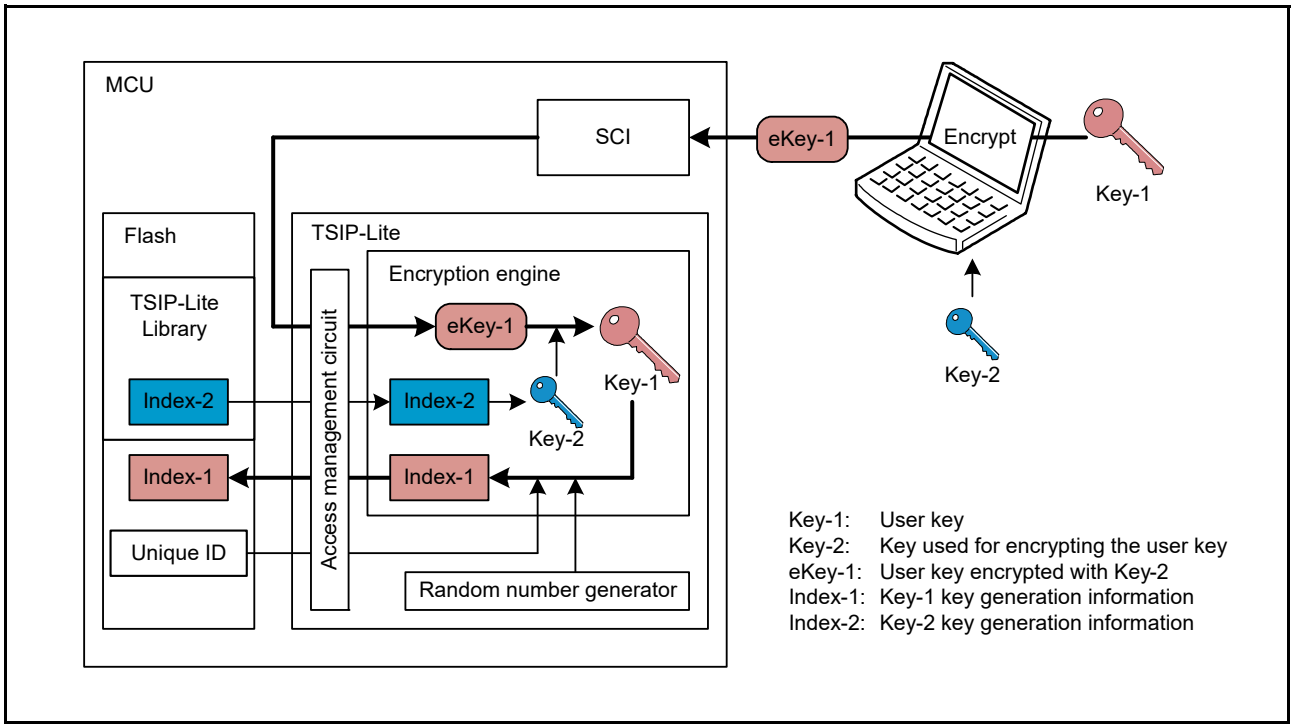


Figure 42.4 Key Installation Process

Downloaded from Arrow.com.

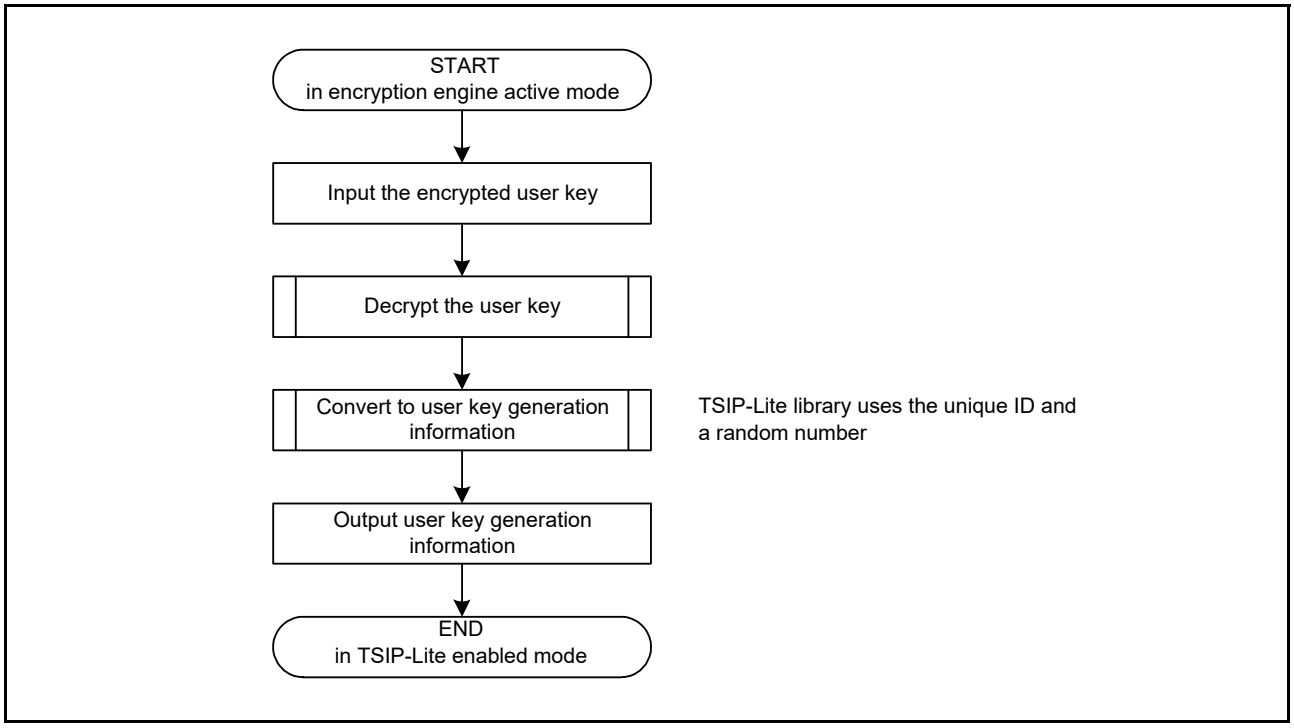


Figure 42.5 Key Installation Flow Chart

42.2.4 Encryption and Decryption

The procedures for encrypting and decrypting data are given below.

(1) Input the key generation information into the TSIP-Lite, and recover the key data.
(2) Input the data to encrypt or decrypt into the TSIP-Lite. This converts plaintext into ciphertext, and ciphertext into plaintext.
(3) Read the converted data.

The encryption engine has an input buffer and an output buffer, enabling encryption/decryption to proceed in parallel with data input/output.
Figure 42.6, Figure 42.7, and Figure 42.8 show the timing diagram, encryption flow, and decryption flow, respectively.

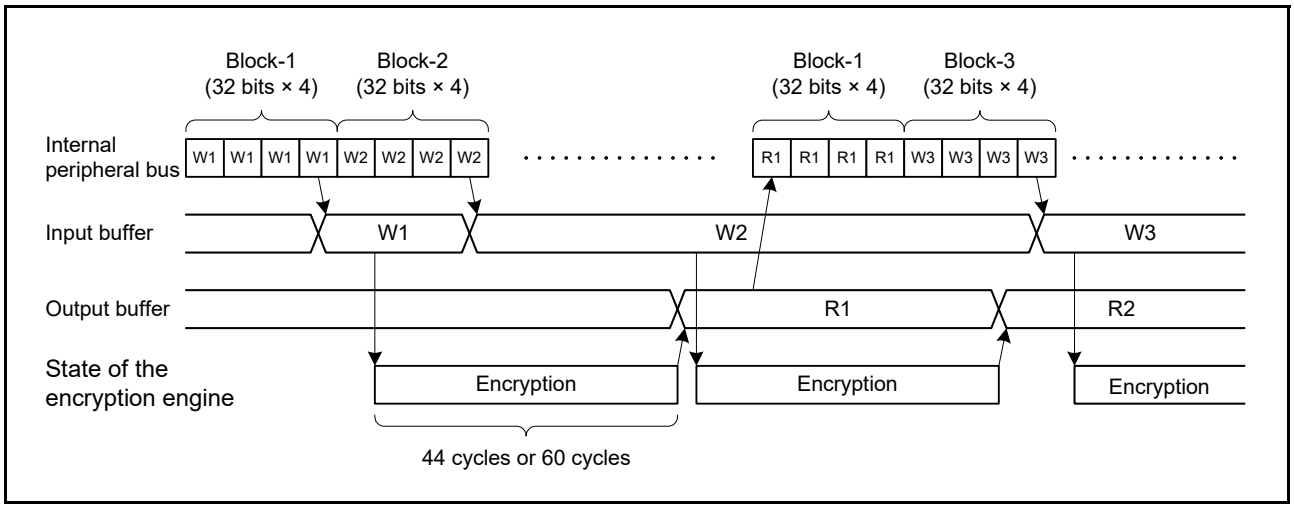


Figure 42.6 Encryption and Decryption Timing Diagram

Downloaded from Arrow.com.

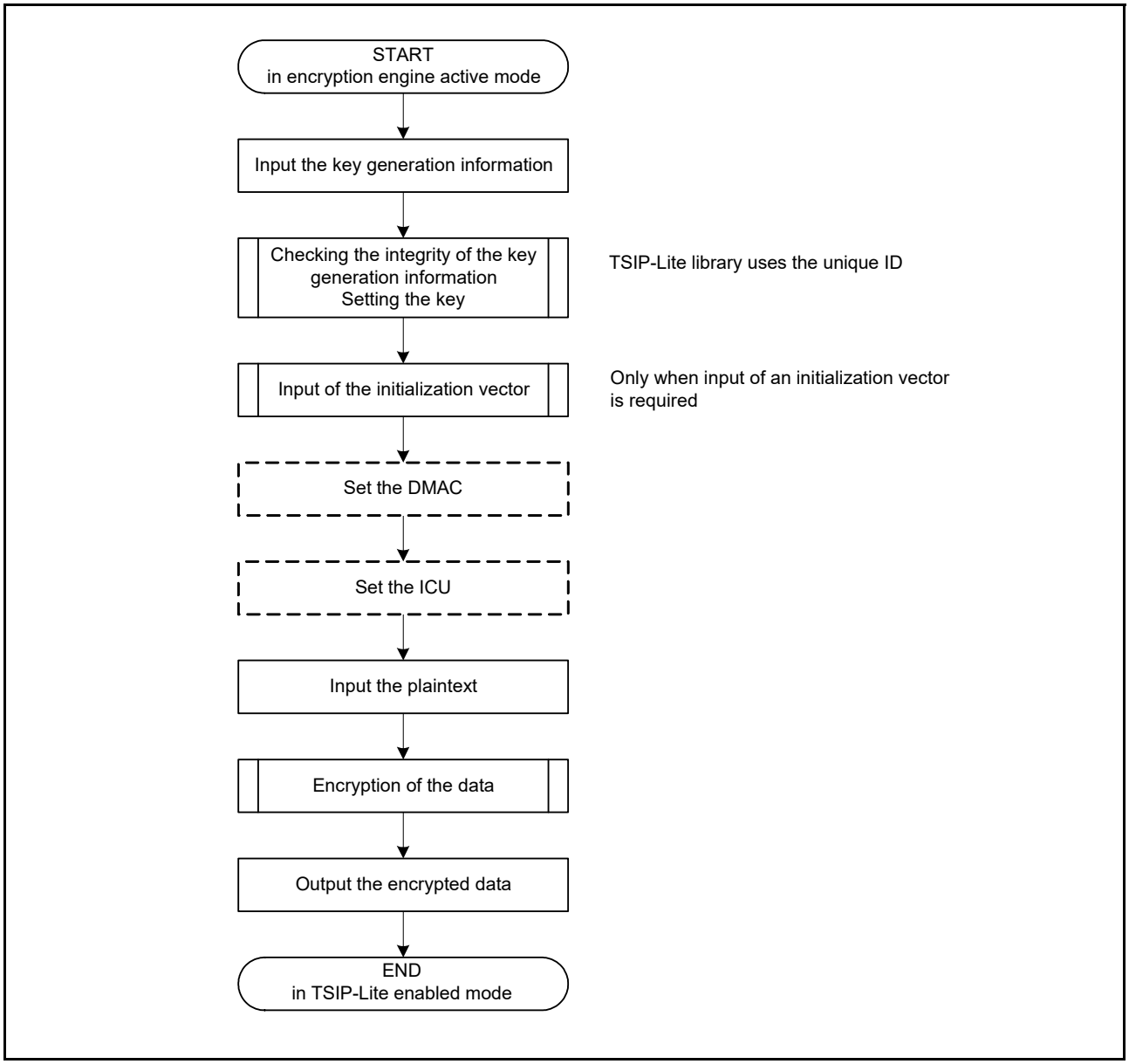


Figure 42.7 Encryption Flow Chart

Downloaded from Arrow.com.

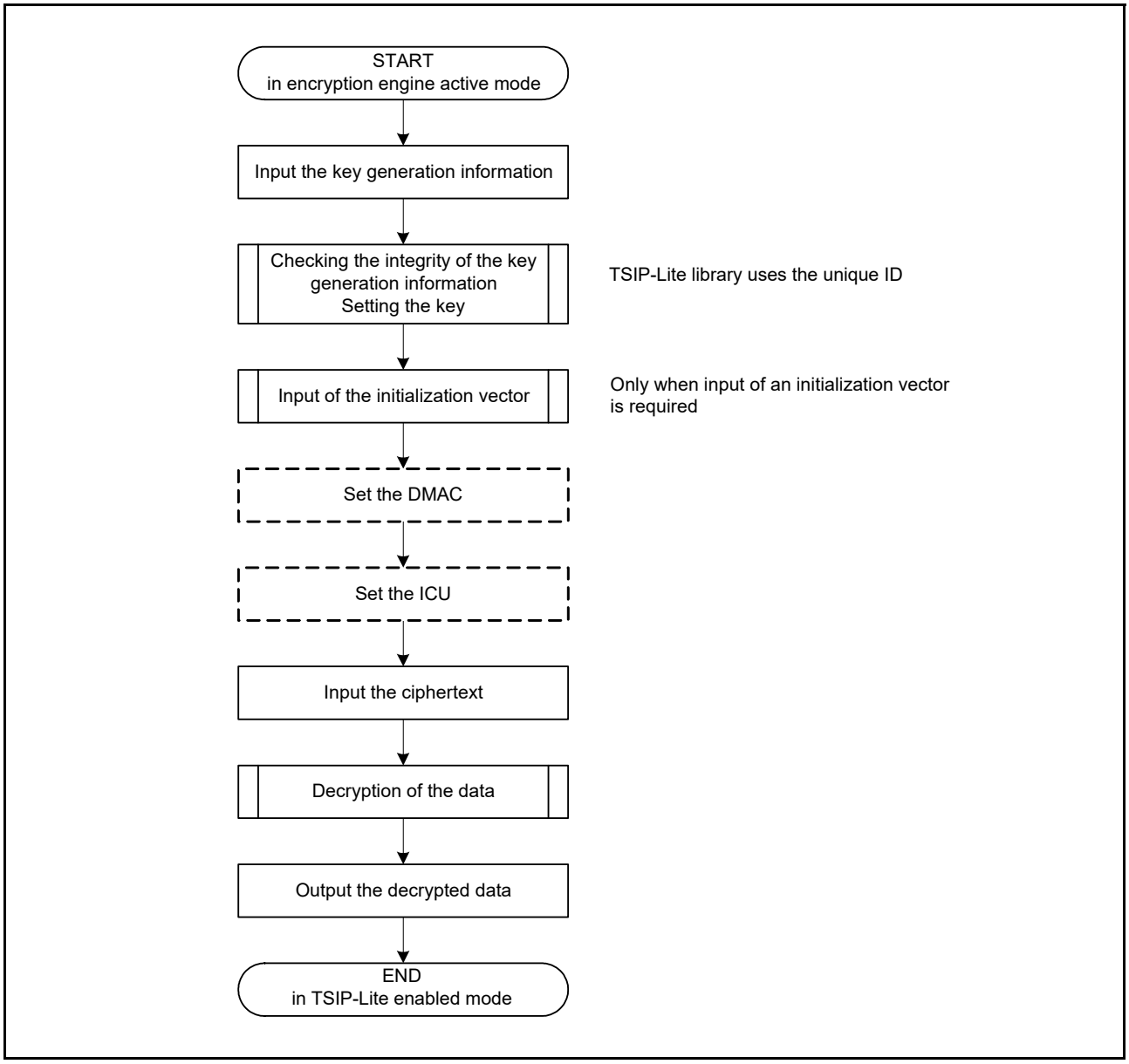


Figure 42.8 Decryption Flow Chart

Downloaded from Arrow.com.

42.2.5 Generating Key Generation Information (by Using Random Numbers)

Figure 42.9 shows the generating flow for the key generation information by using random numbers.

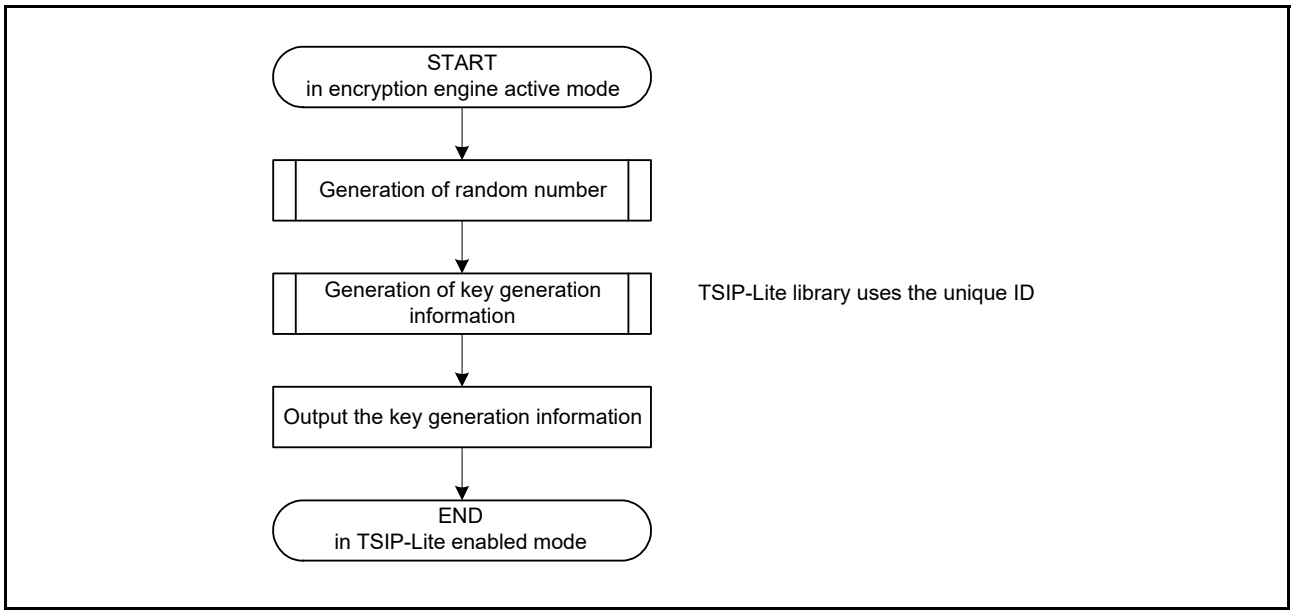


Figure 42.9 Key Generation Information Generating Flow Chart (Using Random Numbers)

42.2.6 Random Number Generation

Figure 42.10 shows the random number generation flow.

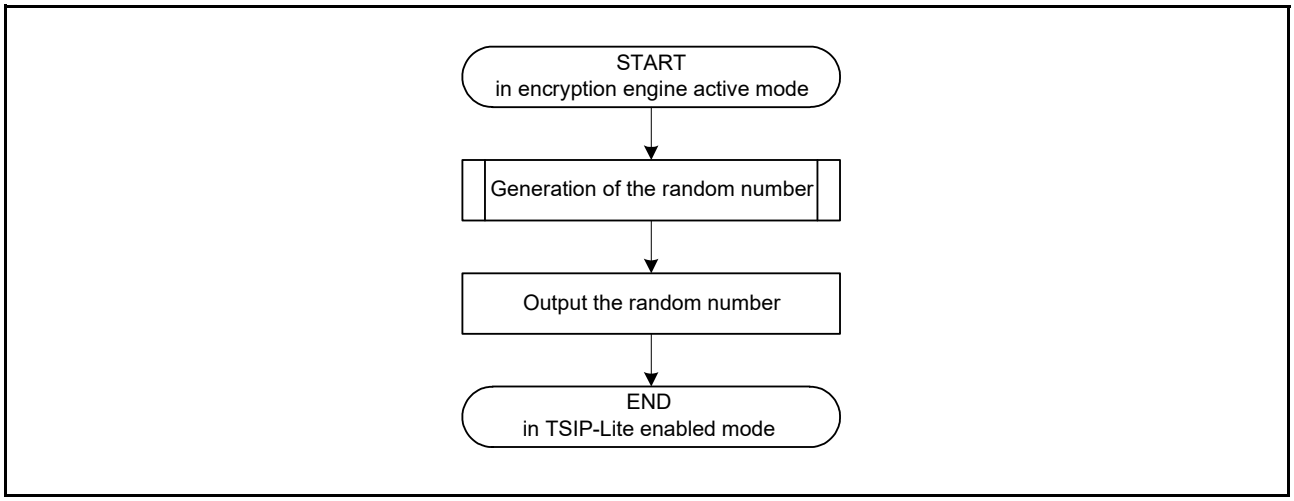


Figure 42.10 Random Number Generation Flow Chart

Downloaded from Arrow.com.

42.3 Interrupt

Table 42.2 lists the interrupt sources.
TSIP-Lite library uses interrupts caused by these interrupt sources. Do not set the ICU.IERm.IENj bits corresponding to these interrupt sources to 0.

Table 42.2 TSIP-Lite Interrupt Sources

Name	Interrupt Source	DTC Triggerable	DMAC Triggerable
RD	Data read ready	Yes	Yes
WR	Data write ready	Yes	Yes
Error	Illegal access detected	No	No

42.4 Usage Notes

42.4.1 Standby Mode

When standby mode is entered while the encryption engine is in processing, proper processing cannot be resumed after standby mode is exited. Standby mode should therefore be entered only after first entering TSIP-Lite disabled mode or TSIP-Lite enabled mode.

42.4.2 Setting the Module Stop Function

The module stop control register D (MSTPCRD) enables or disables operation of the TSIP-Lite. After a reset, the TSIP-Lite is stopped. After exiting the module stop state, the TSIP-Lite can be accessed. Refer to section 11, Low Power Consumption for details.

42.4.3 TSIP-Lite Library

Use of the TSIP-Lite requires the TSIP-Lite library provided by Renesas Electronics. Please contact our sales office for information regarding the TSIP-Lite library.

Downloaded from Arrow.com.

43. Capacitive Touch Sensing Unit (CTSU)

The capacitive touch sensing unit (CTSU) measures the electrostatic capacitance of the touch sensor. Changes in the electrostatic capacitance are determined by software, which enables the CTSU to detect whether a finger is in contact with the touch sensor. The electrode surface of the touch sensor is usually enclosed with a dielectric so that a finger does not come into contact with the electrode.

As shown in Figure 43.1, electrostatic capacitance (parasitic capacitance) exists between the electrode and the surrounding conductors. Because the human body is an electrical conductor, when a finger is placed close to the electrode, the value of electrostatic capacitance increases.

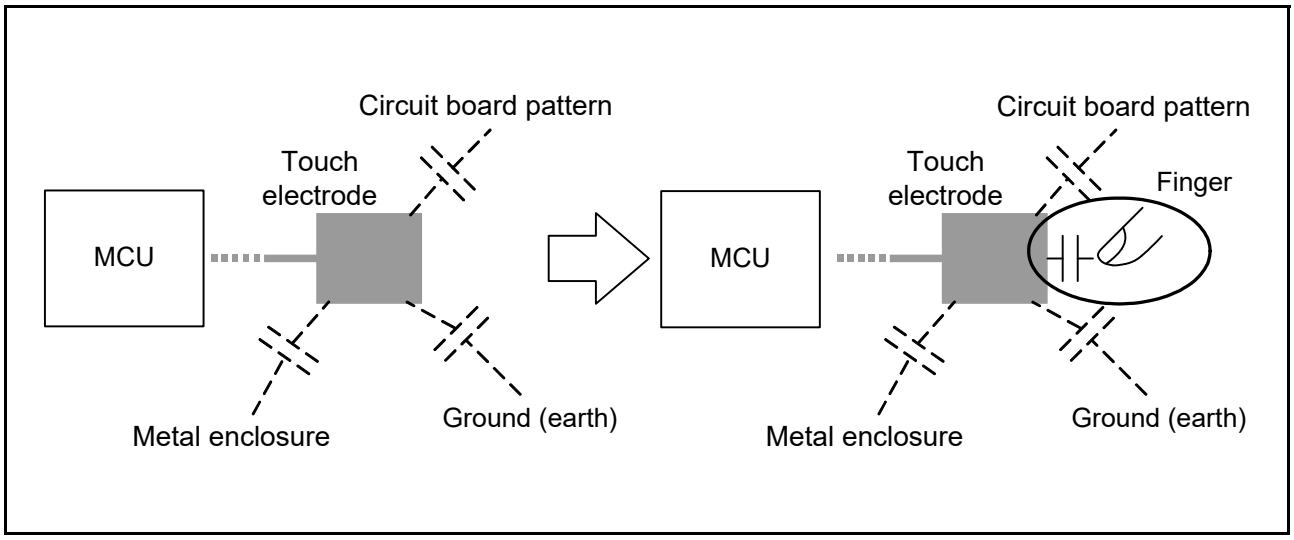


Figure 43.1 Increased Electrostatic Capacitance Due to Presence of Finger

Electrostatic capacitance is detected by the following methods: Self-capacitance and mutual capacitance.

In the self-capacitance method, the CTSU detects electrostatic capacitance generated between a finger and a single electrode. In the mutual capacitance method, two electrodes are used as a transmit electrode and a receive electrode, and the CTSU detects the change in the electrostatic capacitance generated between the two when a finger is placed close to them.

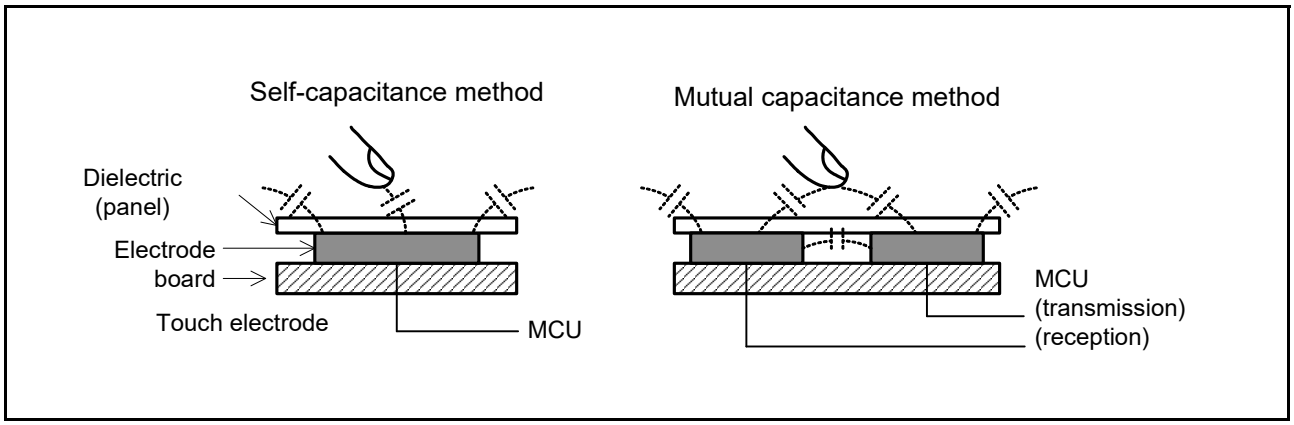


Figure 43.2 Self-Capacitance Method and Mutual Capacitance Method

Electrostatic capacitance is measured by counting a clock signal whose frequency changes according to the amount of charged/discharged current, for a specified period.

For details on the measurement principles of the CTSU, refer to section 43.3.1, Principles of Measurement Operation.

In this section, “PCLK” is used to refer to PCLKB.

Downloaded from Arrow.com.

43.1 Overview

Table 43.1 lists the specifications of the CTSU, and Figure 43.3 shows a block diagram of the CTSU.

Table 43.1 CTSU Specifications

Item		Description
Operating clock		PCLK, PCLK/2, or PCLK/4
Pins	TS2, TS3, TS4, TS7, TS8, TS12, TS13, TS22, TS23, TS27, TS30, TS35	Electrostatic capacitance measurement pins (12 channels)
	TSCAP	LPF (low-pass filter) connection pin
Measurement modes	Self-capacitance single scan mode	Electrostatic capacitance on a channel is measured by the self-capacitance method.
	Self-capacitance multi-scan mode	Electrostatic capacitance on multiple channels is measured successively by the self-capacitance method.
	Mutual capacitance full scan mode	Electrostatic capacitance on multiple channels is measured successively by mutual capacitance.
Noise prevention		Synchronous noise prevention, high-pass noise prevention
Measurement start conditions		• Software trigger • External trigger (event input from the event link controller (ELC))

As shown in Figure 43.3, the CTSU consists of the status control block, trigger control block, clock control block, channel control block, port control block, sensor drive pulse generator, measurement block, interrupt block, and control registers.

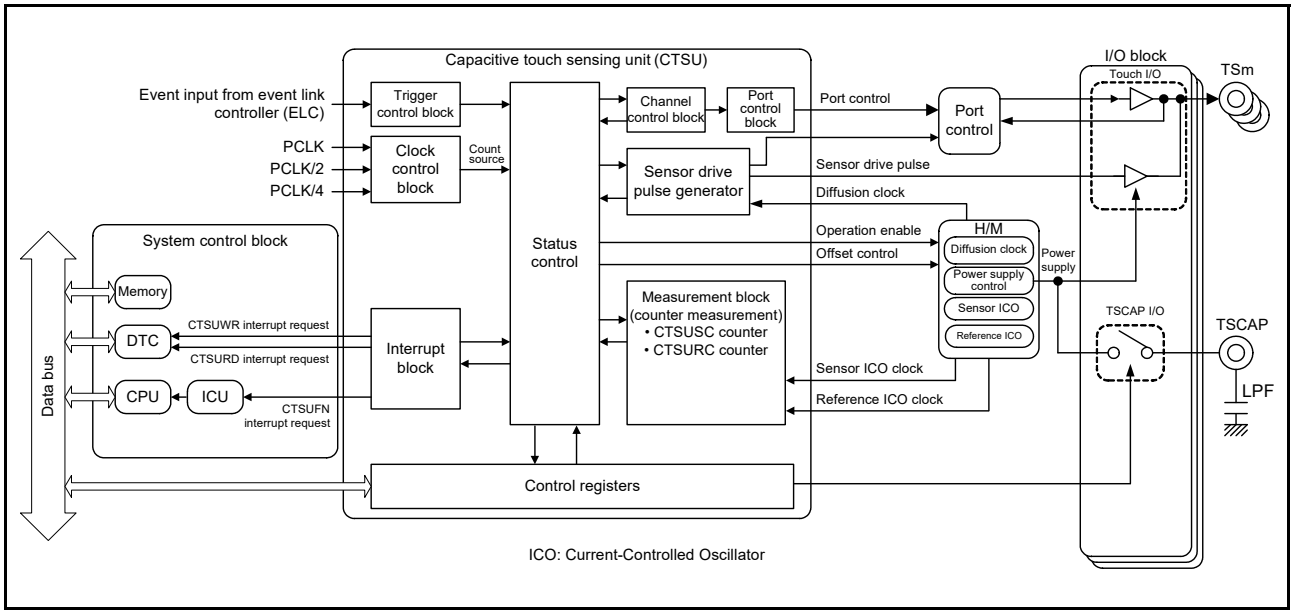


Figure 43.3 CTSU Block Diagram (m = 2, 3, 4, 7, 8, 12, 13, 22, 23, 27, 30, 35)

Table 43.2 CTSU Pin Configuration

Pin Name	I/O	Function
TS2, TS3, TS4, TS7, TS8, TS12, TS13, TS22, TS23, TS27, TS30, TS35	I/O	Electrostatic capacitive measurement pins (touch pins)
TSCAP	—	LPF connection pin

Downloaded from Arrow.com.

43.2 Register Descriptions

43.2.1 CTSU Control Register 0 (CTSUCR0)

Address(es): CTSU.CTSUCR0 000A 0900h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	CTSUINIT	CTSUIOC	CTSUSNZ	CTSUCAP	CTSUSTRT
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CTSUSTRT	CTSU Measurement Operation Start	0: Measurement operation stops 1: Measurement operation starts	R/W
b1	CTSUCAP	CTSU Measurement Operation Start Trigger Select	0: Software trigger 1: External trigger	R/W
b2	CTSUSNZ	CTSU Wait State Power-Saving Enable	0: Power-saving function during wait state is disabled 1: Power-saving function during wait state is enabled	R/W
b3	CTSUIOC	CTSU Transmit Pin Control	0: The TS pins are driven low 1: The TS pins are driven high	R/W
b4	CTSUINIT	CTSU Control Block Initialization	Writing 1 to this bit initializes the CTSU control block and registers*1. This bit is read as 0.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. The CTSUSC, CTSURC, CTSUMCH0, CTSUMCH1, and CTSUST registers are initialized.

The CTSUCAP and CTSUSNZ bits should be set when the CTSUSTRT bit is 0. These bits can be set at the same time as starting measurement operation.

CTSUSTRT Bit (CTSU Measurement Operation Start)

This bit specifies whether CTSU operation starts or stops.
When the CTSUCAP bit is 0 (software trigger), measurement is started by writing 1 to the CTSUSTRT bit, and the CTSUSTRT bit becomes 0 when measurement is finished.
When the CTSUCAP bit is 1 (external trigger), the CTSU waits for an external trigger by writing 1 to the CTSUSTRT bit, and measurement is started at the rising edge of the external trigger. When measurement is finished, the CTSU waits for the next external trigger and operation is continued.
Table 43.3 lists the CTSU states.

Table 43.3 CTSU States

CTSUSTRT Bit	CTSUCAP Bit	CTSU State
0	0	Stopped
0	1	Stopped
1	0	During measurement
1	1	During measurement/wait for an external trigger*1

Note 1. The state can be read from the CTSUST.CTSUSTC[2:0] flags.
During measurement: CTSUST.CTSUSTC[2:0] flags ≠ 000b
Wait for an external trigger: CTSUST.CTSUSTC[2:0] flags = 000b

If the CTSUSTRT bit is set to 1 when the CTSUSTRT bit is 1, writing is ignored and operation is continued.
To forcibly stop operation (forced stop) when the CTSUSTRT bit is 1, set the CTSUSTRT bit to 0 and the CTSUINIT bit to 1 simultaneously.

Downloaded from Arrow.com.

CTSUCAP Bit (CTSU Measurement Operation Start Trigger Select)

This bit specifies the measurement start condition. For details, see the description of the CTSUSTRT bit.

CTSUSNZ Bit (CTSU Wait State Power-Saving Enable)

This bit enables or disables power-saving operation during a wait state. This bit can also be used to suspend the CTSU power supply, which decreases power consumption during the wait state.
In the suspended state, the CTSU power supply is turned off while the external TSCAP is still charged after the CTSU power supply has been turned on and the TSCAP has been charged.

Table 43.4 CTSU Power Supply State Control

CTSUCR1.CTSUPON Bit	CTSUSNZ Bit	CTSUCAP Bit	CTSUSTRT Bit	CTSU Power Supply State
0	0	0	0	Stopped
1	0	—	—	Operating
1	1	0	0	Suspended

Note: Settings other than the above are prohibited.

To start measurement from the suspended state, set the CTSUSNZ bit to 0 and wait for 16 µs before setting the CTSUSTRT bit to 1. To set the suspended state after measurement is finished, set the CTSUSNZ bit to 1.

CTSUIOC Bit (CTSU Transmit Pin Control)

This bit selects the logic level of the TS pin when the CTSUERRS.CTSUTSOD bit is set to 1.
This bit setting is ignored when the CTSUTSOD bit is 0.

CTSUINIT Bit (CTSU Control Block Initialization)

The internal control registers can be initialized by writing 1 to this bit. To forcibly stop the current operation, set the CTSUSTRT bit to 0 and the CTSUINIT bit to 1 simultaneously. In this case, the operation is stopped and the internal control registers are initialized.
Do not write 1 to the CTSUINIT bit at the same time as setting the CTSUSTRT bit to 1 (CTSU operation starts).

Downloaded from Arrow.com.

43.2.2 CTSU Control Register 1 (CTSUCR1)

Address(es): CTSU.CTSUCR1 000A 0901h

	b7	b6	b5	b4	b3	b2	b1	b0
	CTSUMD[1:0]		CTSUCLK[1:0]		CTSUATUNE1	CTSUATUNE0	CTSUCSW	CTSUPON
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CTSUPON	CTSU Power Supply Enable	0: Powered off 1: Powered on	R/W
b1	CTSUCSW	CTSU LPF Capacitance Charging Control	0: Capacitance switch turned off 1: Capacitance switch turned on	R/W
b2	CTSUATUNE0	CTSU Power Supply Operating Mode Setting	VCC ≥ 2.4 V 0: Normal operating mode 1: Low-voltage operating mode VCC < 2.4 V 0: Setting prohibited 1: Low-voltage operating mode	R/W
b3	CTSUATUNE1	CTSU Power Supply Capacity Adjustment	0: Normal output 1: High-current output	R/W
b5, b4	CTSUCLK[1:0]	CTSU Operating Clock Select	b5 b4 0 0: PCLK 0 1: PCLK/2 (PCLK divided by 2) 1 0: PCLK/4 (PCLK divided by 4) 1 1: Setting prohibited	R/W
b7, b6	CTSUMD[1:0]	CTSU Measurement Mode Select	b7 b6 0 0: Self-capacitance single scan mode 0 1: Self-capacitance multi-scan mode 1 0: Setting prohibited 1 1: Mutual capacitance full scan mode	R/W

The CTSUCR1 register should be set when the CTSUCR0.CTSUSTRT bit is 0.

CTSUPON Bit (CTSU Power Supply Enable)

This bit controls power supply to the CTSU. Set the CTSUPON and CTSUCSW bits to the same value at the same time.

CTSUCSW Bit (CTSU LPF Capacitance Charging Control)

This bit controls charging of the LPF capacitor connected to the TSCAP pin (turning on/off of the capacitance switch). After the capacitance switch is turned on, wait until the capacitance connected to the TSCAP pin is charged for the specified time before starting measurement (CTSUCR0.CTSUSTRT = 1). Prior to measurement, use an I/O port to output a low level to the TSCAP pin, and discharge the LPF capacitance that has been already charged. Set the CTSUPON and CTSUCSW bits to the same value at the same time.

CTSUATUNE0 Bit (CTSU Power Supply Operating Mode Setting)

This bit sets the power supply operating mode. Set this bit according to the lower limit of the VCC for operating the CTSU. As an example, when performing touch measurement in a system (the VCC voltage range is 2 to 3 V) where the VCC varies depending on battery operation, set this bit to 1 regardless of the initial VCC voltage.

CTSUATUNE1 Bit (CTSU Power Supply Capacity Adjustment)

This bit sets the capacity of the CTSU power supply. Normally, the value of this bit should be set to 0.

Downloaded from Arrow.com.

CTSUCLK[1:0] Bits (CTSU Operating Clock Select)

These bits select the operating clock.

CTSUMD[1:0] Bits (CTSU Measurement Mode Select)

These bits set the measurement mode. For details, refer to section 43.3.2, Measurement Modes.

43.2.3 CTSU Synchronous Noise Reduction Setting Register (CTSUSDPRS)

Address(es): CTSU.CTSUSDPRS 000A 0902h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	CTSUSOFF	CTSUPRMODE[1:0]		CTSUPRRATIO[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	CTSUPRRATIO[3:0]	CTSU Measurement Time and Pulse Count Adjustment	Recommended setting value: 3 (0011b)	R/W
b5, b4	CTSUPRMODE[1:0]	CTSU Base Period and Pulse Count Setting	b5 b4 0 0: 510 pulses 0 1: 126 pulses 1 0: 62 pulses (recommended setting value) 1 1: Setting prohibited	R/W
b6	CTSUSOFF	CTSU High-Pass Noise Reduction Function Off Setting	0: High-pass noise reduction function turned on 1: High-pass noise reduction function turned off	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

The CTSUSDPRS register should be set when the CTSUCR0.CTSUSTRT bit is 0.

CTSUPRRATIO[3:0] Bits (CTSU Measurement Time and Pulse Count Adjustment)

These bits are used to determine the measurement time and the number of measurement pulses. These are calculated by the following formula. The number of base pulses is determined by setting the CTSUPRMODE[1:0] bits.
Number of measurement pulses = number of base pulses × (CTSUPRRATIO[3:0] bits + 1)
Measurement time = (number of base pulses × (CTSUPRRATIO[3:0] bits + 1) + (number of base pulses – 2) × 0.25) × base clock cycle

Note: For details on the base clock cycle, refer to section 43.2.21, CTSU Sensor Offset Register 1 (CTSUSO1).

CTSUPRMODE[1:0] Bits (CTSU Base Period and Pulse Count Setting)

These bits select the number of base pulses during measurement.

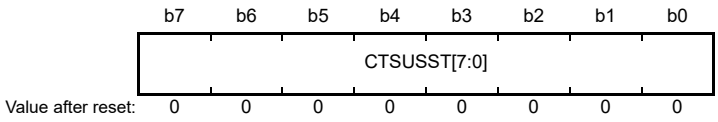
CTSUSOFF Bit (CTSU High-Pass Noise Reduction Function Off Setting)

This bit turns on or off the function for reducing high-pass noise. Set this bit to 1 when turning off the high-pass noise reduction function.

Downloaded from Arrow.com.

43.2.4 CTSU Sensor Stabilization Wait Control Register (CTSUSST)

Address(es): CTSU.CTSUSST 000A 0903h



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	CTSUSST[7:0]	CTSU Sensor Stabilization Wait Control	The value of these bits should be fixed to 00010000b.	R/W

The CTSUSST register should be set when the CTSUCR0.CTSUSTRT bit is 0.

CTSUSST[7:0] Bits (CTSU Sensor Stabilization Wait Control)

These bits set the stabilization wait time for the TSCAP pin voltage. The value of these bits should be fixed to 00010000b. If these bits are not set, the TSCAP voltage becomes unstable at the start of measurement, and the CTSU is unable to obtain correct touch measurement results.

Downloaded from Arrow.com.

43.2.5 CTSU Measurement Channel Register 0 (CTSUMCH0)

Address(es): CTSU.CTSUMCH0 000A 0904h



Bit	Symbol	Bit Name	Description	R/W
b5 to b0	CTSUMCH0[5:0]	CTSU Measurement Channel 0	• In self-capacitance single scan b5 b0 0 0 0 0 1 0: TS2 0 0 0 0 1 1: TS3 0 0 0 1 0 0: TS4 0 0 0 1 1 1: TS7 0 0 1 0 0 0: TS8 0 0 1 1 0 0: TS12 0 0 1 1 0 1: TS13 0 1 0 1 1 0: TS22 0 1 0 1 1 1: TS23 0 1 1 0 1 1: TS27 0 1 1 1 1 0: TS30 1 0 0 0 1 1: TS35 Other than above: Starting measurement operation (CTSUCR0.CTSUSTRT bit = 1) is prohibited after these bits are set. • In other measurement modes b5 b0 0 0 0 0 1 0: TS2 0 0 0 0 1 1: TS3 0 0 0 1 0 0: TS4 0 0 0 1 1 1: TS7 0 0 1 0 0 0: TS8 0 0 1 1 0 0: TS12 0 0 1 1 0 1: TS13 0 1 0 1 1 0: TS22 0 1 0 1 1 1: TS23 0 1 1 0 1 1: TS27 0 1 1 1 1 0: TS30 1 0 0 0 1 1: TS35 1 1 1 1 1 1: Measurement is stopped	R/W*1
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Writing to these bits is enabled only in self-capacitance single scan mode (CTSUCR1.CTSUMD[1:0] bits = 00b).

The CTSUMCH0 register should be set when the CTSUCR0.CTSUSTRT bit is 0.

CTSUMCH0[5:0] Bits (CTSU Measurement Channel 0)

These bits set the channel to be measured in self-capacitance single scan mode, and indicate the receive channel that is being measured in other modes.

Set only enabled channels (000010b, 000011b, 000100b, 000111b, 001000b, 001100b, 001101b, 010110b, 010111b, 011011b, 011110b, 100011b) when setting channels in self-capacitance single scan mode. In other modes, writing to these bits has no effect.

Downloaded from Arrow.com.

43.2.6 CTSU Measurement Channel Register 1 (CTSUMCH1)

Address(es): CTSU.CTSUMCH1 000A 0905h



Bit	Symbol	Bit Name	Description	R/W
b5 to b0	CTSUMCH1[5:0]	CTSU Measurement Channel 1	b5 b0 0 0 0 0 1 0: TS2 0 0 0 0 1 1: TS3 0 0 0 1 0 0: TS4 0 0 0 1 1 1: TS7 0 0 1 0 0 0: TS8 0 0 1 1 0 0: TS12 0 0 1 1 0 1: TS13 0 1 0 1 1 0: TS22 0 1 0 1 1 1: TS23 0 1 1 0 1 1: TS27 0 1 1 1 1 0: TS30 1 0 0 0 1 1: TS35 1 1 1 1 1 1: Measurement is stopped	R
b7, b6	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R

CTSUMCH1[5:0] Bits (CTSU Measurement Channel 1)

These bits indicate the transmit channel that is being measured in full scan mode. The value of these bits is 111111b while measurement is stopped or in self-capacitance single scan mode and multi-scan mode.

Downloaded from Arrow.com.

43.2.7 CTSU Channel Enable Control Register 0 (CTSUCHAC0)

Address(es): CTSU.CTSUCHAC0 000A 0906h

	b7	b6	b5	b4	b3	b2	b1	b0
	CTSUCHAC07	—	—	CTSUCHAC04	CTSUCHAC03	CTSUCHAC02	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b2	CTSUCHAC02	CTSU Channel 2 Enable Control	0: Not measurement target 1: Measurement target	R/W
b3	CTSUCHAC03	CTSU Channel 3 Enable Control		R/W
b4	CTSUCHAC04	CTSU Channel 4 Enable Control		R/W
b6, b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	CTSUCHAC07	CTSU Channel 7 Enable Control	0: Not measurement target 1: Measurement target	R/W

The CTSUCHAC0 register should be set when the CTSUCR0.CTSUSTRT bit is 0.

CTSUCHAC0j Bit (CTSU Channel m Enable Control) (j = 2 to 4, 7; m = 2 to 4, 7)

This bit sets the pin (for receive and transmit) whose electrostatic capacitance is to be measured.

Downloaded from Arrow.com.

43.2.8 CTSU Channel Enable Control Register 1 (CTSUCHAC1)

Address(es): CTSU.CTSUCHAC1 000A 0907h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	CTSUCHAC15	CTSUCHAC14	—	—	—	CTSUCHAC10
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CTSUCHAC10	CTSU Channel 8 Enable Control	0: Not measurement target 1: Measurement target	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	CTSUCHAC14	CTSU Channel 12 Enable Control	0: Not measurement target 1: Measurement target	R/W
b5	CTSUCHAC15	CTSU Channel 13 Enable Control		R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The CTSUCHAC1 register should be set when the CTSUCR0.CTSUSTRT bit is 0.

CTSUCHAC1j Bit (CTSU Channel m Enable Control) (j = 0, 4, 5; m = 8, 12, 13)

This bit sets the pin (for receive and transmit) whose electrostatic capacitance is to be measured.

Downloaded from Arrow.com.

43.2.9 CTSU Channel Enable Control Register 2 (CTSUCHAC2)

Address(es): CTSU.CTSUCHAC2 000A 0908h

	b7	b6	b5	b4	b3	b2	b1	b0
	CTSUCHAC27	CTSUCHAC26	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	CTSUCHAC26	CTSU Channel 22 Enable Control	0: Not measurement target 1: Measurement target	R/W
b7	CTSUCHAC27	CTSU Channel 23 Enable Control		R/W

The CTSUCHAC2 register should be set when the CTSUCR0.CTSUSTRT bit is 0.

CTSUCHAC2j Bit (CTSU Channel m Enable Control) (j = 6, 7; m = 22, 23)

This bit sets the pin (for receive and transmit) whose electrostatic capacitance is to be measured.

Downloaded from Arrow.com.

43.2.10 CTSU Channel Enable Control Register 3 (CTSUCHAC3)

Address(es): CTSU.CTSUCHAC3 000A 0909h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	CTSUCHAC36	—	—	CTSUCHAC33	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b3	CTSUCHAC33	CTSU Channel 27 Enable Control	0: Not measurement target 1: Measurement target	R/W
b5, b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	CTSUCHAC36	CTSU Channel 30 Enable Control	0: Not measurement target 1: Measurement target	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

The CTSUCHAC3 register should be set when the CTSUCR0.CTSUSTRT bit is 0.

CTSUCHAC3j Bit (CTSU Channel m Enable Control) (j = 3, 6; m = 27, 30)

This bit sets the pin (for receive and transmit) whose electrostatic capacitance is to be measured.

Downloaded from Arrow.com.

43.2.11 CTSU Channel Enable Control Register 4 (CTSUCHAC4)

Address(es): CTSU.CTSUCHAC4 000A 090Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	CTSUCHAC43	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b3	CTSUCHAC43	CTSU Channel 35 Enable Control	0: Not measurement target 1: Measurement target	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The CTSUCHAC4 register should be set when the CTSUCR0.CTSUSTRT bit is 0.

CTSUCHAC43 Bit (CTSU Channel 35 Enable Control)

This bit sets the pin (for receive and transmit) whose electrostatic capacitance is to be measured.

Downloaded from Arrow.com.

43.2.12 CTSU Channel Transmit/Receive Control Register 0 (CTSUCHTRC0)

Address(es): CTSU.CTSUCHTRC0 000A 090Bh

	b7	b6	b5	b4	b3	b2	b1	b0
	CTSUCHTRC07	—	—	CTSUCHTRC04	CTSUCHTRC03	CTSUCHTRC02	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b2	CTSUCHTRC02	CTSU Channel 2 Transmit/Receive Control	0: Reception 1: Transmission	R/W
b3	CTSUCHTRC03	CTSU Channel 3 Transmit/Receive Control		R/W
b4	CTSUCHTRC04	CTSU Channel 4 Transmit/Receive Control		R/W
b6, b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	CTSUCHTRC07	CTSU Channel 7 Transmit/Receive Control	0: Reception 1: Transmission	R/W

The CTSUCHTRC0 register should be set when the CTSUCR0.CTSUSTRT bit is 0.

CTSUCHTRC0j Bit (CTSU Channel m Transmit/Receive Control) (j = 2 to 4, 7; m = 2 to 4, 7)

This bit allocates reception or transmission to the corresponding TSm pin in full scan mode. The setting of this bit is ignored in self-capacitance single scan mode and multi-scan mode.

Downloaded from Arrow.com.

43.2.13 CTSU Channel Transmit/Receive Control Register 1 (CTSUCHTRC1)

Address(es): CTSU.CTSUCHTRC1 000A 090Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	CTSUCHTRC15	CTSUCHTRC14	—	—	—	CTSUCHTRC10
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CTSUCHTRC10	CTSU Channel 8 Transmit/Receive Control	0: Reception 1: Transmission	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	CTSUCHTRC14	CTSU Channel 12 Transmit/Receive Control	0: Reception 1: Transmission	R/W
b5	CTSUCHTRC15	CTSU Channel 13 Transmit/Receive Control		R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The CTSUCHTRC1 register should be set when the CTSUCR0.CTSUSTRT bit is 0.

CTSUCHTRC1j Bit (CTSU Channel m Transmit/Receive Control) (j = 0, 4, 5; m = 8, 12, 13)

This bit allocates reception or transmission to the corresponding TSm pin in full scan mode. The setting of this bit is ignored in self-capacitance single scan mode and multi-scan mode.

Downloaded from Arrow.com.

43.2.14 CTSU Channel Transmit/Receive Control Register 2 (CTSUCHTRC2)

Address(es): CTSU.CTSUCHTRC2 000A 090Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	CTSUCHTRC27	CTSUCHTRC26	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	CTSUCHTRC26	CTSU Channel 22 Transmit/Receive Control	0: Reception 1: Transmission	R/W
b7	CTSUCHTRC27	CTSU Channel 23 Transmit/Receive Control		R/W

The CTSUCHTRC2 register should be set when the CTSUCR0.CTSUSTRT bit is 0.

CTSUCHTRC2j Bit (CTSU Channel m Transmit/Receive Control) (j = 6, 7; m = 22, 23)

This bit allocates reception or transmission to the corresponding TSm pin in full scan mode. The setting of this bit is ignored in self-capacitance single scan mode and multi-scan mode.

Downloaded from Arrow.com.

43.2.15 CTSU Channel Transmit/Receive Control Register 3 (CTSUCHTRC3)

Address(es): CTSU.CTSUCHTRC3 000A 090Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	CTSUCHTRC36	—	—	CTSUCHTRC33	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b3	CTSUCHTRC33	CTSU Channel 27 Transmit/Receive Control	0: Reception 1: Transmission	R/W
b5, b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	CTSUCHTRC36	CTSU Channel 30 Transmit/Receive Control	0: Reception 1: Transmission	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

The CTSUCHTRC3 register should be set when the CTSUCR0.CTSUSTRT bit is 0.

CTSUCHTRC3j Bit (CTSU Channel m Transmit/Receive Control) (j = 3, 6; m = 27, 30)

This bit allocates reception or transmission to the corresponding TSm pin in full scan mode. The setting of this bit is ignored in self-capacitance single scan mode and multi-scan mode.

Downloaded from Arrow.com.

43.2.16 CTSU Channel Transmit/Receive Control Register 4 (CTSUCHTRC4)

Address(es): CTSU.CTSUCHTRC4 000A 090Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	CTSUCHTRC43	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b3	CTSUCHTRC43	CTSU Channel 35 Transmit/Receive Control	0: Reception 1: Transmission	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The CTSUCHTRC4 register should be set when the CTSUCR0.CTSUSTRT bit is 0.

CTSUCHTRC43 Bit (CTSU Channel 35 Transmit/Receive Control)

This bit allocates reception or transmission to the corresponding TS35 pin in full scan mode. The setting of this bit is ignored in self-capacitance single scan mode and multi-scan mode.

Downloaded from Arrow.com.

43.2.17 CTSU High-Pass Noise Reduction Control Register (CTSUDCLKC)

Address(es): CTSU.CTSUDCLKC 000A 0910h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	CTSUSSCNT[1:0]		—	—	CTSUSSMOD[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	CTSUSSMOD[1:0]	CTSU Diffusion Clock Mode Select	These bits should be set to 00b.	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5, b4	CTSUSSCNT[1:0]	CTSU Diffusion Clock Control	These bits should be set to 11b.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The CTSUDCLKC register should be set when the CTSUCR0.CTSUSTRT bit is 0.

CTSUSSMOD[1:0] Bits (CTSU Diffusion Clock Mode Select)

These bits set the mode of the spectrum diffusion clock for high-pass noise reduction. When using the high-pass function, the value of these bits should be fixed to 00b. If these bits are not set, the effect of high-pass noise reduction cannot be correctly obtained.

CTSUSSCNT[1:0] Bits (CTSU Diffusion Clock Control)

These bits adjust the spectrum diffusion amount to reduce high-pass noise. When using the high-pass noise reduction function, the value of these bits should be fixed to 11b. If these bits are not set, touch measurement may not be correctly performed.

Downloaded from Arrow.com.

43.2.18 CTSU Status Register (CTSUST)

Address(es): CTSU.CTSUST 000A 0911h

	b7	b6	b5	b4	b3	b2	b1	b0
	CTSUPS	CTSUROVF	CTSUSOVF	CTSUDTSR	—	CTSUSTC[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	CTSUSTC[2:0]	CTSU Measurement Status Counter	b2 b0 0 0 0: Status 0 0 0 1: Status 1 0 1 0: Status 2 0 1 1: Status 3 1 0 0: Status 4 1 0 1: Status 5	R
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	CTSUDTSR	CTSU Data Transfer Status Flag	0: Measurement result has been read 1: Measurement result has not been read	R
b5	CTSUSOVF	CTSU Sensor Counter Overflow Flag	0: No overflow 1: An overflow	R/W
b6	CTSUROVF	CTSU Reference Counter Overflow Flag	0: No overflow 1: An overflow	R/W
b7	CTSUPS	CTSU Mutual Capacitance Status Flag	0: First measurement 1: Second measurement	R

When using the CTSUCR0.CTSUINIT bit to clear an overflow flag, make sure that the CTSUCR0.CTSUSTRT bit is 0.

CTSUSTC[2:0] Flags (CTSU Measurement Status Counter)

These counters indicate the current measurement status. For details on each status, refer to section 43.3.2.2, Status Counter.

CTSUDTSR Flag (CTSU Data Transfer Status Flag)

This flag indicates whether the measurement result stored in the sensor counter and the reference counter has been read.
This flag is set to 1 when measurement is completed; 0 when the reference counter is read by software or the DTC.
This flag is also cleared using the CTSUCR0.CTSUINIT bit.

CTSUSOVF Flag (CTSU Sensor Counter Overflow Flag)

This flag indicates whether the sensor counter has overflowed. FFFFh can be read as the measurement result (CTSUSC counter) when an overflow has occurred.
Even if an overflow occurs, measurement processing is continued until the set period.
No interrupt is generated even when an overflow occurs. To determine the channel on which the overflow has occurred, read the measurement result of each channel after measurement is completed (after a measurement end interrupt is generated).
This flag is cleared when 0 is written after 1 is read by software. This flag is also cleared using the CTSUCR0.CTSUINIT bit.

CTSUROVF Flag (CTSU Reference Counter Overflow Flag)

This flag indicates whether the reference counter has overflowed. FFFFh can be read as the measurement result (CTSURC counter) when an overflow has occurred.
Even if an overflow occurs, measurement processing is continued until the set period.

Downloaded from Arrow.com.

No interrupt is generated even when an overflow occurs. To determine the channel on which the overflow has occurred, read the measurement result of each channel after measurement is completed (after a measurement end interrupt is generated).
This flag is cleared when 0 is written after 1 is read by software. This flag is also cleared using the CTSUCR0.CTSUINIT bit.

CTSUPS Flag (CTSU Mutual Capacitance Status Flag)

This flag indicates whether the measurement is the first or second of two measurements for each channel in mutual capacitance full scan mode (CTSUCR1.CTSUMD[1:0] bits = 11b).
This flag indicates 0 while measurement is stopped or in other measurement modes.

Downloaded from Arrow.com.

43.2.19 CTSU High-Pass Noise Reduction Spectrum Diffusion Control Register (CTSUSSC)

Address(es): CTSU.CTSUSSC 000A 0912h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	CTSUSSDIV[3:0]				—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11 to b8	CTSUSSDIV[3:0]	CTSU Spectrum Diffusion Frequency Division Setting	These bits specify the spectrum diffusion frequency division setting according to the base clock frequency division setting.	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

CTSUSSDIV[3:0] Bits (CTSU Spectrum Diffusion Frequency Division Setting)

These bits specify the spectrum diffusion frequency division setting according to the base clock frequency division setting. See the relationship between base clock frequencies and the CTSUSSDIV[3:0] bits settings in Table 43.5, for setting the value of these bits.

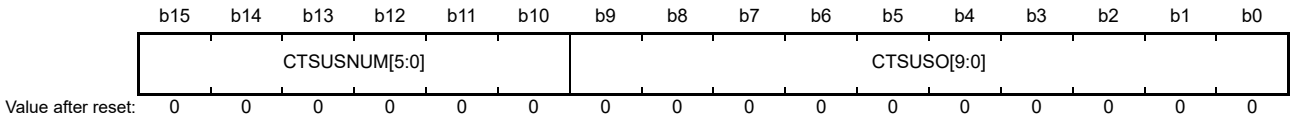
Table 43.5 Relationship between Base Clock Frequencies and the CTSUSSDIV[3:0] Bits Settings

Base Clock Frequency fb (MHz)	CTSUSSDIV[3:0] Bits Setting
4.00 ≤ fb	0000b
2.00 ≤ fb < 4.00	0001b
1.33 ≤ fb < 2.00	0010b
1.00 ≤ fb < 1.33	0011b
0.80 ≤ fb < 1.00	0100b
0.67 ≤ fb < 0.80	0101b
0.57 ≤ fb < 0.67	0110b
0.50 ≤ fb < 0.57	0111b
0.44 ≤ fb < 0.50	1000b
0.40 ≤ fb < 0.44	1001b
0.36 ≤ fb < 0.40	1010b
0.33 ≤ fb < 0.36	1011b
0.31 ≤ fb < 0.33	1100b
0.29 ≤ fb < 0.31	1101b
0.27 ≤ fb < 0.29	1110b
fb < 0.27	1111b

Downloaded from Arrow.com.

43.2.20 CTSU Sensor Offset Register 0 (CTSUSO0)

Address(es): CTSU.CTSUSO0 000A 0914h



Bit	Symbol	Bit Name	Description	R/W
b9 to b0	CTSUSO[9:0]	CTSU Sensor Offset Adjustment	b9 b0 0 0 0 0 0 0 0 0 0 0: Current offset amount is 0 0 0 0 0 0 0 0 0 0 1: Current offset amount is 1 0 0 0 0 0 0 0 0 1 0: Current offset amount is 2 : : 1 1 1 1 1 1 1 1 1 0: Current offset amount is 1022 1 1 1 1 1 1 1 1 1 1: Current offset amount is maximum	R/W
b15 to b10	CTSUSNUM[5:0]	CTSU Measurement Count Setting	These bits set the number of measurements.	R/W

CTSUSO[9:0] Bits (CTSU Sensor Offset Adjustment)

These control bits adjust the input current offset of the sensor ICO. These bits are used to offset the sensor ICO input current generated from electrostatic capacitance while the electrode is not being touched during touch measurement, thus preventing overflow of the CTSU sensor counter.
Make settings for the TS pin that is to be measured next after a CTSUWR interrupt is generated.

CTSUSNUM[5:0] Bits (CTSU Measurement Count Setting)

These bits set how many times the number of measurement pulses specified by the CTSUSDPRS.CTSUPRRATIO[3:0] and CTSUSDPRS.CTSUPRMODE[1:0] bits is repeated in the measurement time. The number of measurement pulses is repeated (CTSUSNUM[5:0] bits + 1) times.
Make settings for the TS pin that is to be measured next after a CTSUWR interrupt is generated.

Downloaded from Arrow.com.

43.2.21 CTSU Sensor Offset Register 1 (CTSUSO1)

Address(es): CTSU.CTSUSO1 000A 0916h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	CTSUICOG[1:0]		CTSUSDPA[4:0]					CTSURICOA[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	CTSURICOA[7:0]	CTSU Reference ICO Current Adjustment	b7 b0 0 0 0 0 0 0 0 0: Input current amount 0 0 0 0 0 0 0 0 1: Input current amount 1 0 0 0 0 0 0 1 0: Input current amount 2 : : 1 1 1 1 1 1 1 0: Input current amount 254 1 1 1 1 1 1 1 1: Input current amount maximum	R/W
b12 to b8	CTSUSDPA[4:0]	CTSU Base Clock Setting	b12 b8 0 0 0 0 0: Operating clock divided by 2*1 0 0 0 0 1: Operating clock divided by 4 0 0 0 1 0: Operating clock divided by 6 0 0 0 1 1: Operating clock divided by 8 0 0 1 0 0: Operating clock divided by 10 0 0 1 0 1: Operating clock divided by 12 0 0 1 1 0: Operating clock divided by 14 0 0 1 1 1: Operating clock divided by 16 0 1 0 0 0: Operating clock divided by 18 0 1 0 0 1: Operating clock divided by 20 0 1 0 1 0: Operating clock divided by 22 0 1 0 1 1: Operating clock divided by 24 0 1 1 0 0: Operating clock divided by 26 0 1 1 0 1: Operating clock divided by 28 0 1 1 1 0: Operating clock divided by 30 0 1 1 1 1: Operating clock divided by 32 1 0 0 0 0: Operating clock divided by 34 1 0 0 0 1: Operating clock divided by 36 1 0 0 1 0: Operating clock divided by 38 1 0 0 1 1: Operating clock divided by 40 1 0 1 0 0: Operating clock divided by 42 1 0 1 0 1: Operating clock divided by 44 1 0 1 1 0: Operating clock divided by 46 1 0 1 1 1: Operating clock divided by 48 1 1 0 0 0: Operating clock divided by 50 1 1 0 0 1: Operating clock divided by 52 1 1 0 1 0: Operating clock divided by 54 1 1 0 1 1: Operating clock divided by 56 1 1 1 0 0: Operating clock divided by 58 1 1 1 0 1: Operating clock divided by 60 1 1 1 1 0: Operating clock divided by 62 1 1 1 1 1: Operating clock divided by 64	R/W
b14, b13	CTSUICOG[1:0]	CTSU ICO Gain Adjustment	b14 b13 0 0: 100% gain 0 1: 66% gain 1 0: 50% gain 1 1: 40% gain	R/W
b15	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note 1. The CTSUSDPA[4:0] bits should not be set to 00000b while the high-pass noise reduction function is turned off (CTSUSDPRS.CTSUSOFF bit = 1) in mutual capacitance full scan mode (CTSUCR1.CTSUMD[1:0] bits = 11b).

Write first to the CTSUSSC register, then CTSUSO0 register, and then CTSUSO1 register after a CTSUWR interrupt is generated. Write operation to the CTSUSO1 register causes a transition to Status 3. Thus, set all the bits in a single setting when writing to the CTSUSO1 register.

Downloaded from Arrow.com.

CTSURICOA[7:0] Bits (CTSU Reference ICO Current Adjustment)

These bits adjust the oscillation frequency using the input current of the reference ICO.

CTSUSDPA[4:0] Bits (CTSU Base Clock Setting)

These bits are used to generate a base clock used as the source for the sensor drive pulse by dividing the operating clock. For details on the setting procedure, refer to section 43.3.2.1, Initial Setting Flowchart.

CTSUICOG[1:0] Bits (CTSU ICO Gain Adjustment)

These bits adjust the output frequency gain of the sensor ICO and the reference ICO. If changes in the capacitance between when the electrode is touched and when it is not touched greatly exceed the dynamic range of the sensor ICO, set the gain adjustment bits to adjust the gain appropriately.

43.2.22 CTSU Sensor Counter (CTSUSC)

Address(es): CTSU.CTSUSC 000A 0918h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CTSUSC[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	CTSUSC[15:0]	CTSU Sensor Counter	These bits indicate FFFFh when an overflow occurs.	R

Read first from the CTSUSC counter and then the CTSURC counter after a CTSURD interrupt is generated.

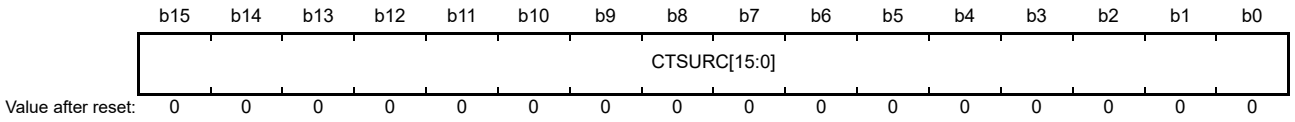
CTSUSC[15:0] Bits (CTSU Sensor Counter)

These bits are configured as an increment counter that counts the sensor ICO clock.
Read these bits after a CTSURD interrupt is generated. After the CTSURC counter is read, these bits are cleared immediately before the CTSU measurement status counter value changes to Status 4 (the CTSUST.CTSUSTC[2:0] flags changes to 100b) in the next measurement. These bits are also cleared using the CTSUCR0.CTSUINIT bit.

Downloaded from Arrow.com.

43.2.23 CTSU Reference Counter (CTSURC)

Address(es): CTSU.CTSURC 000A 091Ah



Bit	Symbol	Bit Name	Description	R/W
b15 to b0	CTSURC[15:0]	CTSU Reference Counter	These bits indicate FFFFh when an overflow occurs.	R

Read first from the CTSUSC counter and then the CTSURC counter after a CTSURD interrupt is generated.
Even when the stabilization time specified for Status 3 has elapsed, if the CTSURC counter is not read, Status 3 continues until the counter is read.

CTSURC[15:0] Bits (CTSU Reference Counter)

These bits are configured as an increment counter that counts the reference ICO clock.
The reference ICO is used to optimize touch measurement performed using the sensor ICO. There is some deviation depending on the internal sensor ICO and the reference ICO in the CTSU, but both ICOs have almost the same characteristics, and the dynamic range and the current to frequency characteristics are almost the same. The range of current amount that can be set by the reference ICO current adjustment bits is about the same as the range of both ICOs, and the current amount input to the sensor ICO must be within this dynamic range. First, use the reference ICO to check the differences between the ICOs and measure the current to oscillation frequency characteristics. Since the reference ICO oscillation frequency can be obtained from the reference ICO counter, the ICO oscillation frequency (counter value/ measurement time) for the input current amount can be measured by setting the value in the reference ICO current adjustment bits and measuring the reference ICO counter. The reference ICO counter value measured using the maximum value of the reference ICO current adjustment bits is the maximum value of the ICO dynamic range. Therefore, the current amount of the sensor ICO needs to be offset by setting the offset adjustment bits so that the sensor ICO counter value does not exceed this value.
Read the CTSURC[15:0] bits after a CTSURD interrupt is generated. After these bits are read, they are cleared immediately before the CTSU measurement status counter value changes to Status 4 (the CTSUST.CTSUSTC[2:0] flags changes to 100b) in the next measurement. These bits are also cleared using the CTSUCR0.CTSUINIT bit.

Downloaded from Arrow.com.

43.2.24 CTSU Error Status Register (CTSUERRS)

Address(es): CTSU.CTSUERRS 000A 091Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CTSUICOMP	—	—	—	—	—	—	—	CTSUTSOC	CTSUCLKSEL1	—	—	CTSUDRV	CTSUTSOD	CTSUSPMD[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	CTSUSPMD[1:0]	Calibration Mode	b1 b0 0 0: Capacitance measurement mode 0 1: Setting prohibited 1 0: Calibration mode 1 1: Setting prohibited	R/W
b2	CTSUTSOD	TS Pin Fixed Output	0: Capacitance measurement mode 1: TS pins are forced to be high or low	R/W
b3	CTSUDRV	Calibration Setting 1	0: Capacitance measurement mode 1: Calibration setting 1	R/W
b5, b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	CTSUCLKSEL1	Calibration Setting 3	0: Capacitance measurement mode 1: Calibration setting 3	R/W
b7	CTSUTSOC	Calibration Setting 2	0: Capacitance measurement mode 1: Calibration setting 2	R/W
b14 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	CTSUICOMP	TSCAP Voltage Error Monitor	0: Normal TSCAP voltage 1: Abnormal TSCAP voltage	R

CTSUSPMD[1:0] Bit (Calibration Mode)

These bits are used to calibrate the CTSU.

When measuring the capacitance, set these bits to 00b.

CTSUTSOD Bit (TS Pin Fixed Output)

This bit is used to calibrate the CTSU. When setting this bit to 1, the TS pins are forced to the logic level specified by the CTSUCR0.CTSUIOC bit.

When measuring the capacitance, set this bit to 0.

CTSUDRV Bit (Calibration Setting 1)

This bit is used to calibrate the CTSU.

When measuring the capacitance, set this bit to 0.

CTSUCLKSEL1 Bit (Calibration Setting 3)

This bit is used to calibrate the CTSU.

When measuring the capacitance, set this bit to 0.

CTSUTSOC Bit (Calibration Setting 2)

This bit is used to calibrate the CTSU.

When measuring the capacitance, set this bit to 0.

CTSUICOMP Bit (TSCAP Voltage Error Monitor)

If the offset current amount set by the CTSUSO0 register exceeds the sensor ICO input current during touch

Downloaded from Arrow.com.

measurement, the TSCAP voltage becomes abnormal and touch measurement cannot be correctly performed. This bit monitors the TSCAP voltage and it is set to 1 if the voltage becomes abnormal. If the TSCAP voltage becomes abnormal, the sensor ICO counter value will be undefined, but touch measurement is normally completed, so it difficult to detect an abnormality by reading the sensor ICO counter value. If the CTSU reference ICO current adjustment bits (CTSURICOA[7:0]) in the CTSUSO1 register are set to a value other than 0, check this bit when touch measurement is completed.
This bit is cleared by writing 0 to the CTSUCR1.CTSUPON bit and turning off the power supply.

Downloaded from Arrow.com.

43.3 Operation

43.3.1 Principles of Measurement Operation

Figure 43.4 shows the measurement circuit.

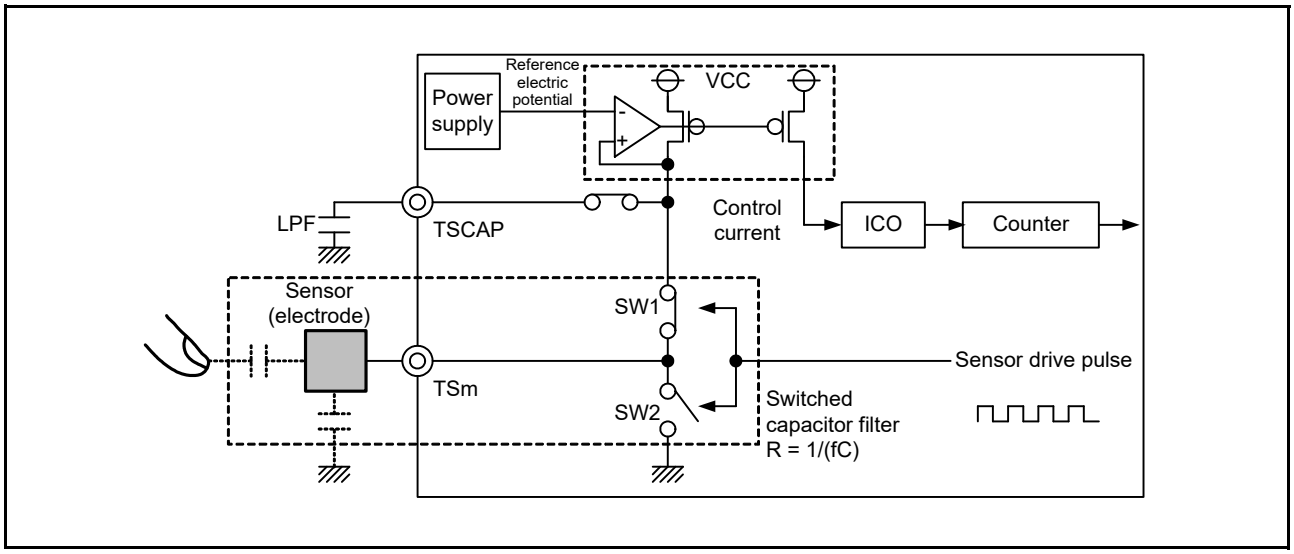


Figure 43.4 Measurement Circuit (m = 2, 3, 4, 7, 8, 12, 13, 22, 23, 27, 30, 35)

The electrostatic capacitance measurement operation principles of the CTSU current frequency conversion method are explained using Figure 43.5 to Figure 43.7.

(1) The electrostatic capacitance of the electrode is charged by turning SW1 on and SW2 off (Figure 43.5).

(2) The charged capacitance is discharged by turning SW1 off and SW2 on (Figure 43.6).

Current flows to the switched capacitor filter by switching between charging and discharging in steps (1) and (2). At this time, the value of electrostatic capacitance varies depending on whether a finger is in close proximity, so the flowing current changes. A clock is generated by supplying the control current, which is proportional to the amount of the current flowing through the switched capacitor filter, from the circuit that generates the TSCAP power supply to the ICO. The counter is used to measure the clock frequency which changes depending on whether a finger is in close proximity, and the value read from the counter is used by software to determine contact with a finger (Figure 43.7).

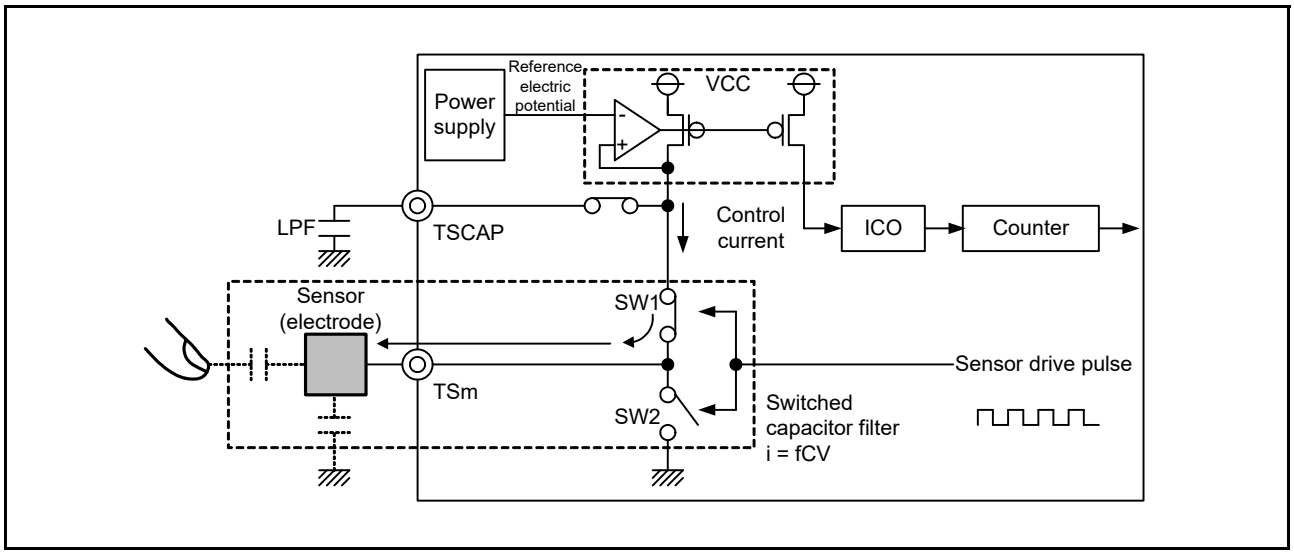


Figure 43.5 Charging Operation (m = 2, 3, 4, 7, 8, 12, 13, 22, 23, 27, 30, 35)

Downloaded from Arrow.com.

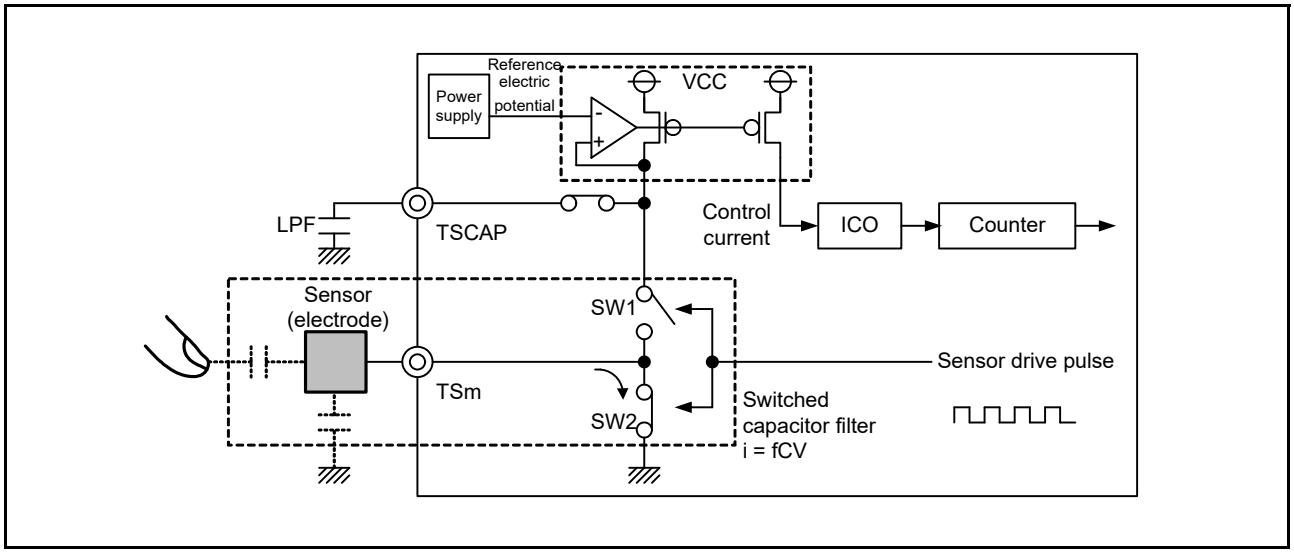


Figure 43.6 Discharging Operation (m = 2, 3, 4, 7, 8, 12, 13, 22, 23, 27, 30, 35)

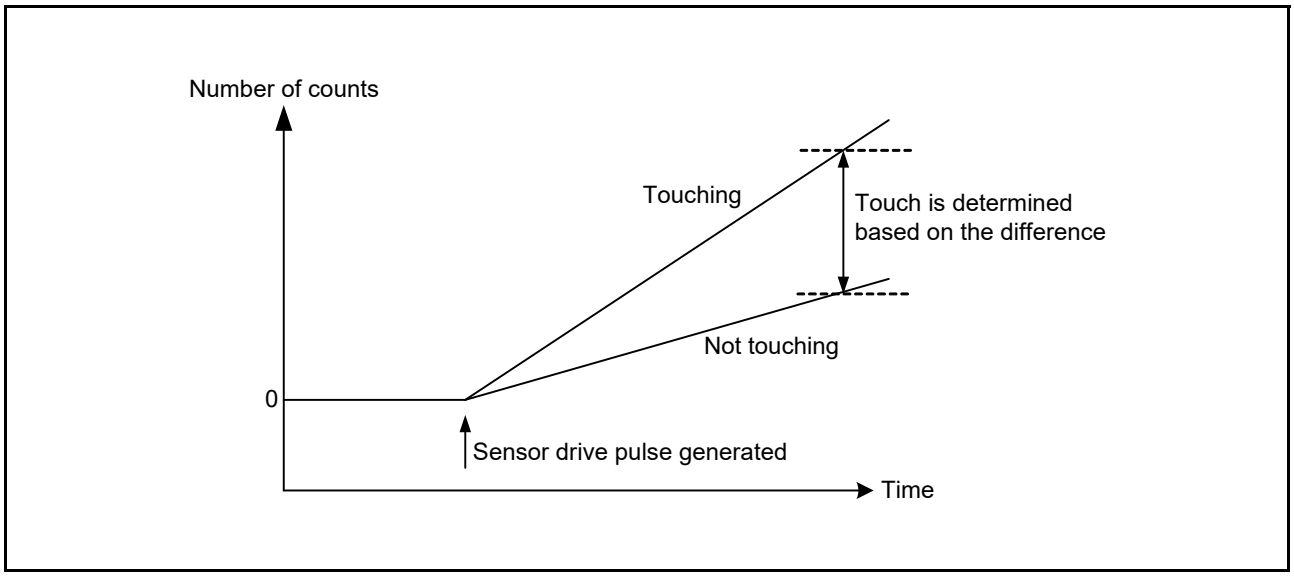


Figure 43.7 Change in Measured Value When Finger is Touching and Not Touching

Downloaded from Arrow.com.

43.3.2 Measurement Modes

The CTSU supports self-capacitance and mutual capacitance methods. Figure 43.8 illustrates these methods.

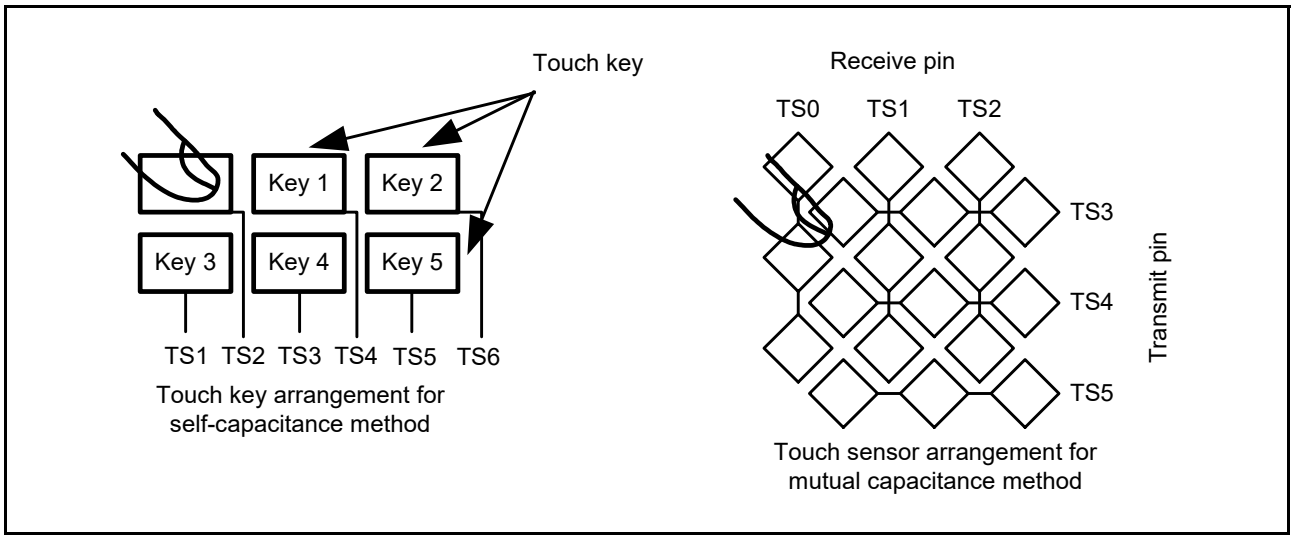


Figure 43.8 Overview of Self-Capacitance Method and Mutual Capacitance Method

In the self-capacitance method, a single touch pin is allocated to a single touch key to measure individual electrostatic capacitance when a finger is in close proximity. In this method, single scan and multi-scan can be used as measurement modes.

In the mutual capacitance method, the capacitance between two opposite electrodes (transmit and receive pins) is measured.

Downloaded from Arrow.com.

43.3.2.1 Initial Setting Flowchart

Figure 43.9 shows the flowchart for CTSU initial setting.

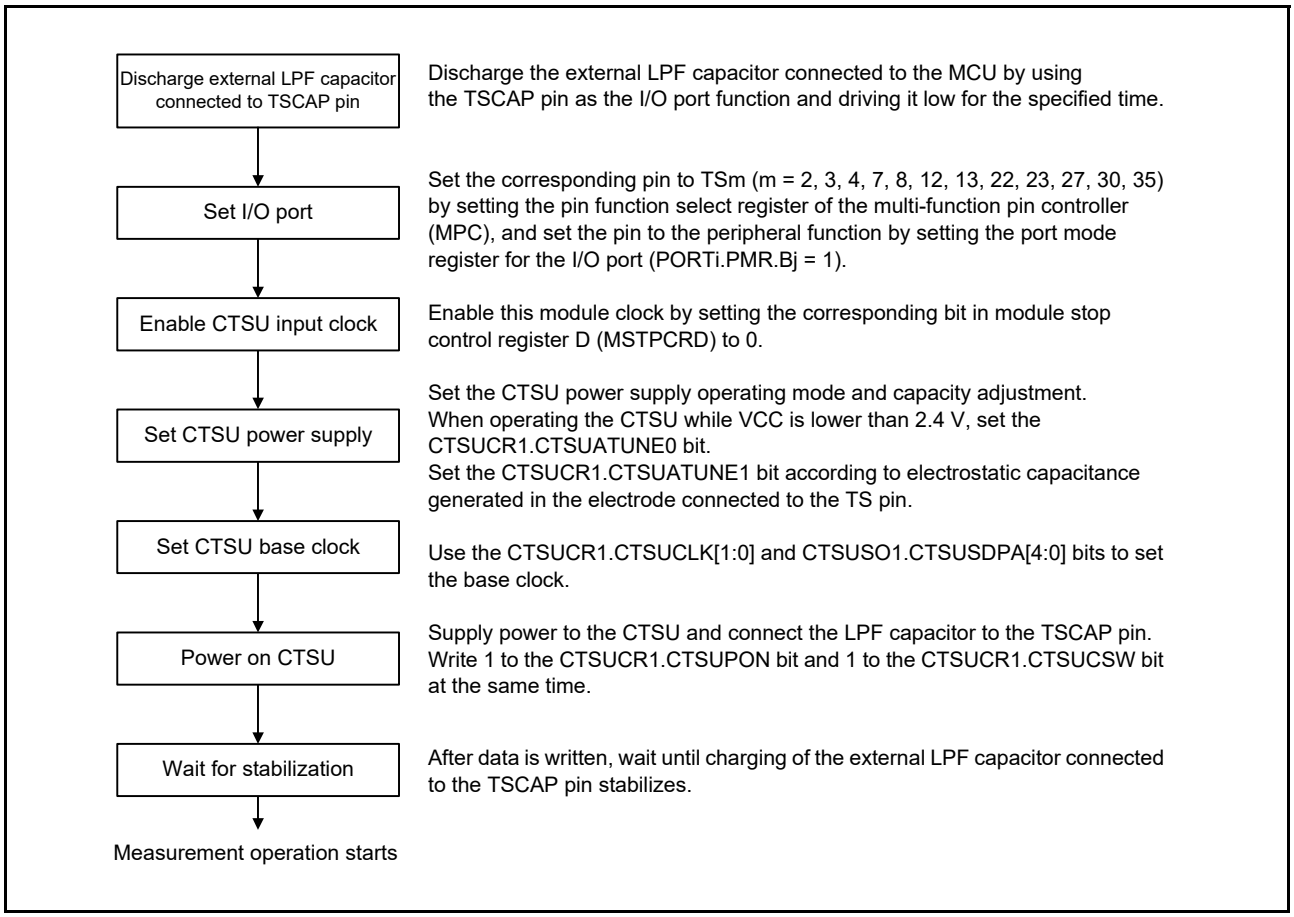


Figure 43.9 CTSU Initial Setting Flowchart

Figure 43.10 shows the flowchart for stopping CTSU operation and setting to the standby state.

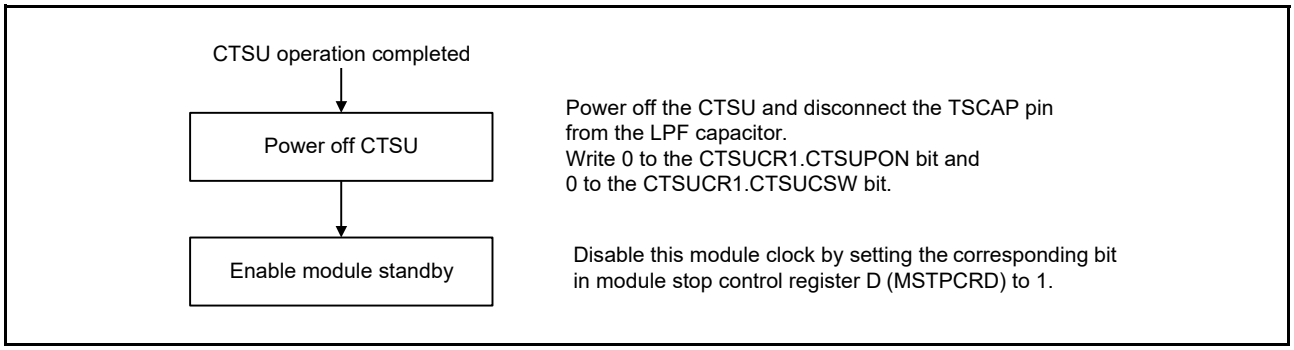


Figure 43.10 CTSU Stopping Flowchart

When restarting operation after it has been stopped, follow the initial setting flowchart shown in Figure 43.9.

Downloaded from Arrow.com.

43.3.2.2 Status Counter

The measurement status counter of the CTSU status register (CTSUST) indicates the current measurement status. The measurement status is common to all four modes. Figure 43.11 shows status operation transitions.

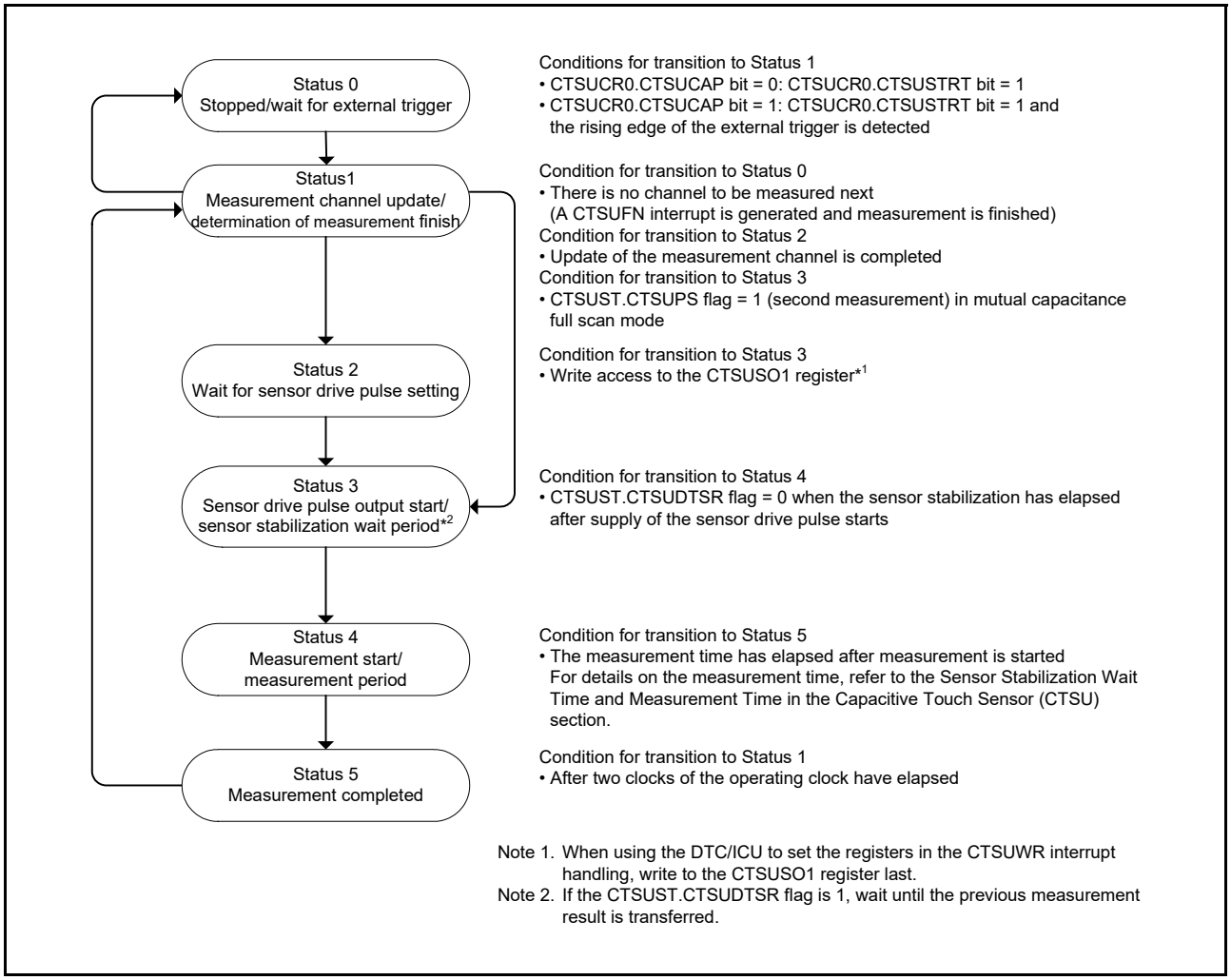


Figure 43.11 Status Operation Transitions

The status of the status counter transitions to Status 0 when all of the specified measurement channels are measured. The CTSUCR0.CTSUSTRT bit is cleared to 0 by hardware when a software trigger is used. When an external trigger is used, the value 1 is retained, and the CTSU waits for the next trigger.

When operation is forcibly stopped (by writing 0 to the CTSUCR0.CTSUSTRT bit and 1 to the CTSUCR0.CTSUINIT bit at the same time) during measurement or the wait state for the trigger, the status transitions to Status 0 and measurement is stopped forcibly.

If there is no channel to be measured by setting the CTSUMCH0, CTSUCHACn, and CTSUCHTRCn registers (n = 0 to 4), a CTSUFN interrupt is generated immediately after a transition to Status 1, and then the status transitions to Status 0. The following are the cases when there is no channel to be measured.

- A measurement target channel is not specified by the CTSUCHACn registers.
- In self-capacitance single scan mode, the channel specified in the CTSUMCH0 register is not a measurement target in the CTSUCHACn registers.
- In full scan mode, there is no transmit channel or receive channel to be measured by combining the CTSUCHACn and CTSUCHTRCn registers.

Downloaded from Arrow.com.

43.3.2.3 Self-Capacitance Single Scan Mode Operation

In self-capacitance single scan mode, electrostatic capacitance on a channel is measured. Figure 43.12 shows the software flowchart and an operation example, and Figure 43.13 shows the timing chart.

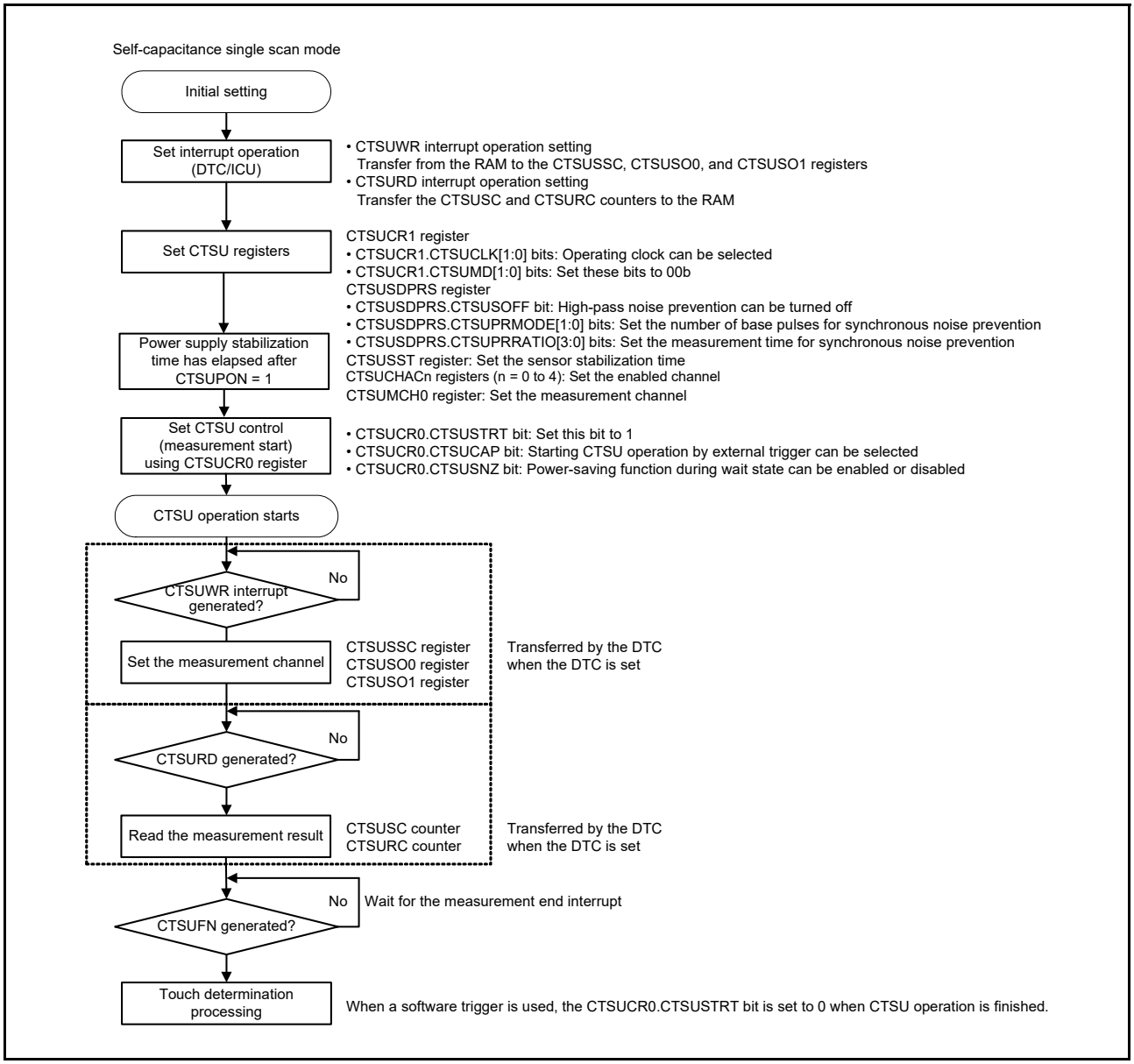


Figure 43.12 Software Flowchart and Operation Example of Self-Capacitance Single Scan Mode

Downloaded from Arrow.com.

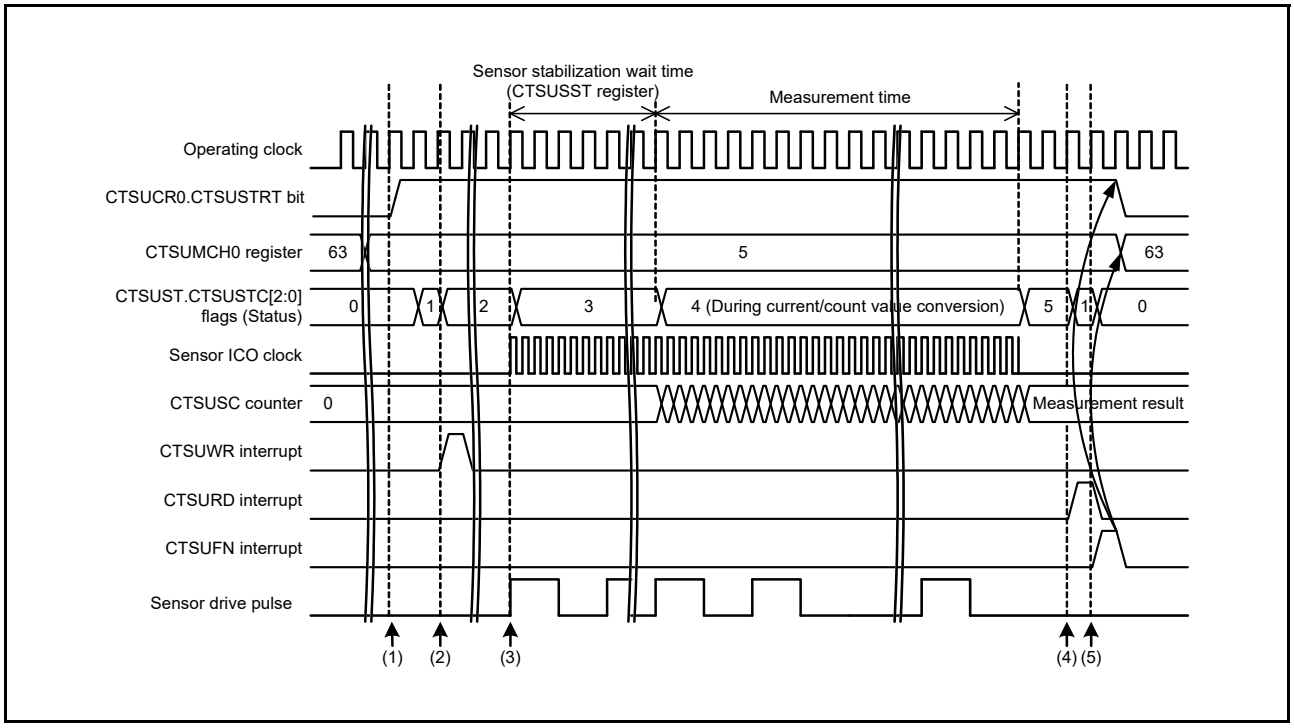


Figure 43.13 Timing Chart of Self-Capacitance Single Scan Mode (Measurement Start Condition is Software Trigger)

The following describes operation shown in the timing chart in Figure 43.13.

(1) After various settings are made, operation is started by writing 1 to the CTSUCR0.CTSUSTRT bit.

(2) After a channel to be measured is determined according to the preset conditions, a request for setting the corresponding channel (CTSUWR) is output.

(3) Upon completion of writing the measurement channel settings (CTSUSSC, CTSUSO0, and CTSUSO1 registers), the sensor drive pulse is output and the sensor ICO clock and the reference ICO clock operate.

(4) After the sensor stabilization wait time and the measurement time have elapsed and measurement is finished, a measurement result read request (CTSURD) is output.

(5) A measurement end interrupt (CTSUFN) is output and measurement is finished (transition to Status 0).

Table 43.6 lists the touch pin states in self-capacitance single scan mode.

Table 43.6 Touch Pin States in Self-Capacitance Single Scan Mode

	Touch Pin	
Status	**Measurement Channel**	**Non-Measurement Channel**
0	Low	Low
1	Low	Low
2	Low	Low
3	Pulse	Low
4	Pulse	Low
5	Low	Low

Downloaded from Arrow.com.

43.3.2.4 Self-Capacitance Multi-Scan Mode Operation

In self-capacitance multi-scan mode, electrostatic capacitance on all channels that are specified as measurement targets by setting the CTSUCHACn registers (n = 0 to 4) are measured sequentially in ascending order. Figure 43.14 shows the software flowchart and an operation example, and Figure 43.15 shows the timing chart.

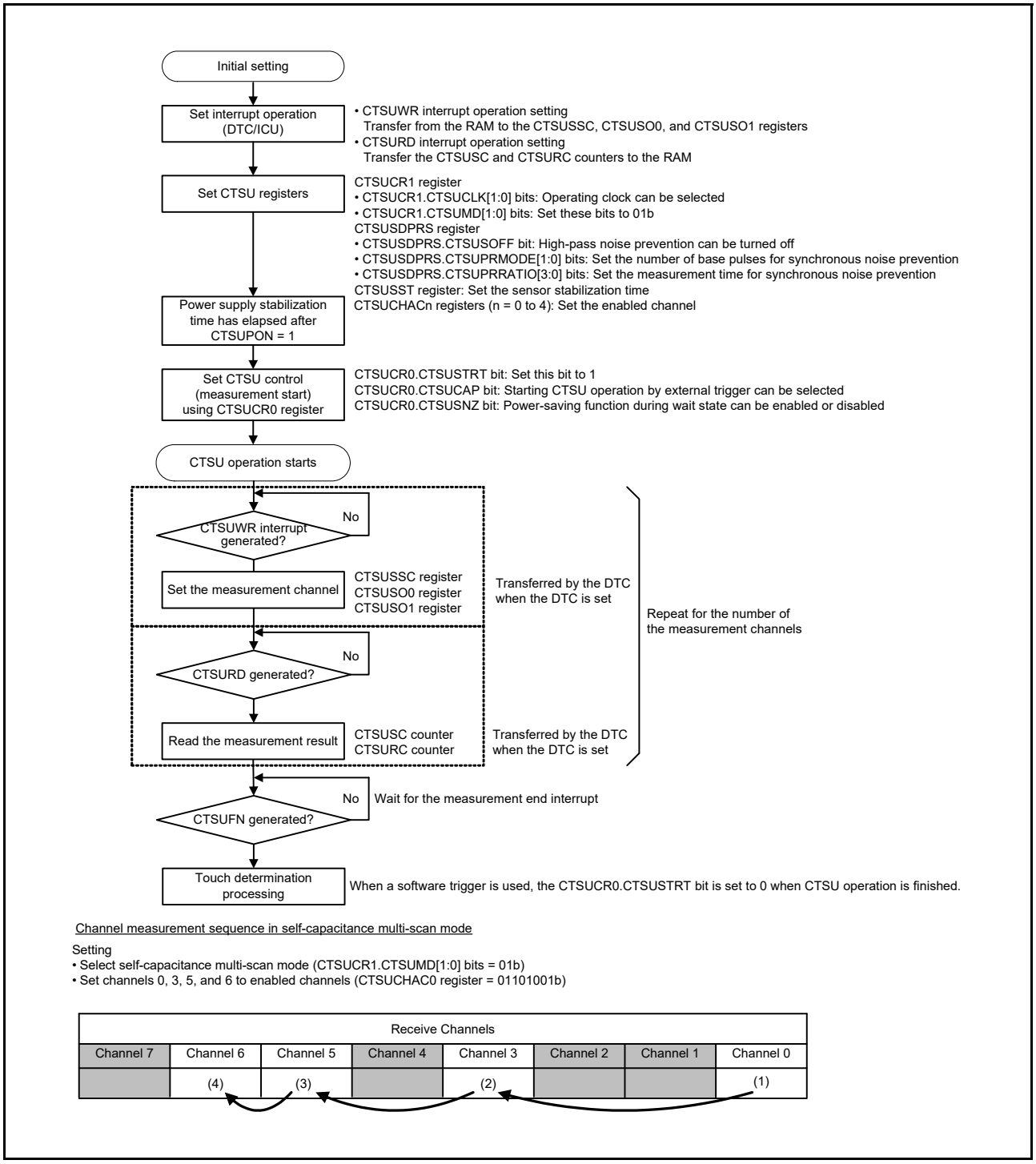


Figure 43.14 Software Flow and Operation Example of Self-Capacitance Multi-Scan Mode

Downloaded from Arrow.com.

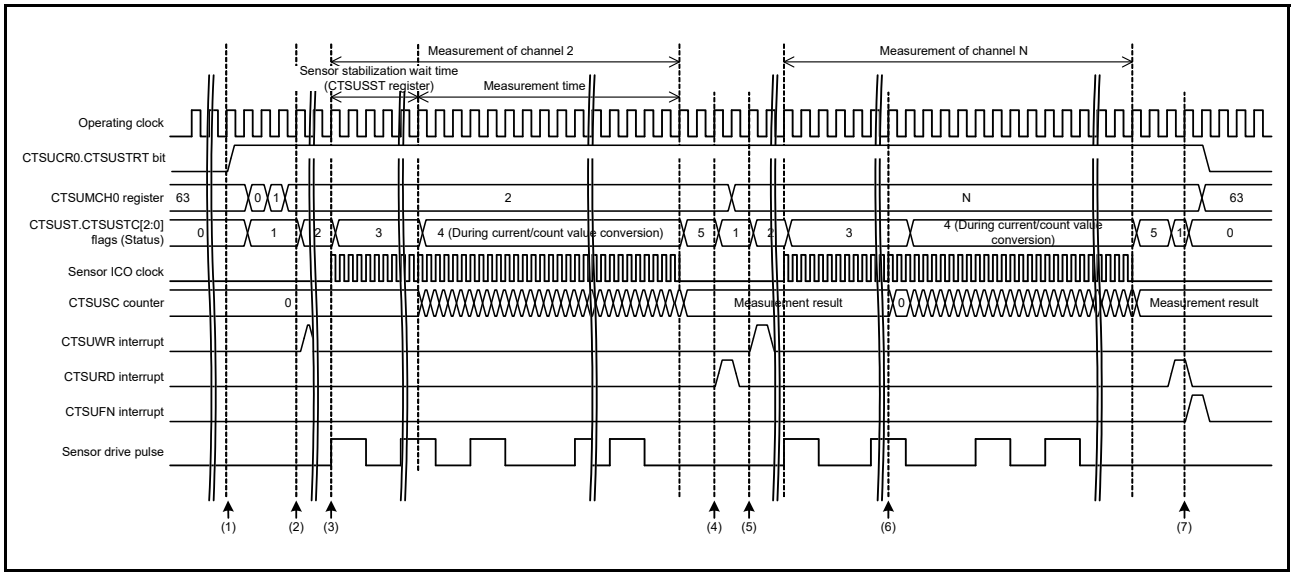


Figure 43.15 Timing Chart of Self-Capacitance Multi-Scan Mode (Measurement Start Condition is Software Trigger)

The following describes operation shown in the timing chart in Figure 43.15.

(1) After various settings are made, operation is started by writing 1 to the CTSUCR0.CTSUSTRT bit.

(2) After a channel to be measured is determined according to the preset conditions, a request for setting the corresponding channel (CTSUWR) is output.

(3) Upon completion of writing the measurement channel settings (CTSUSSC, CTSUSO0, and CTSUSO1 registers), the sensor drive pulse is output and the sensor ICO clock and the reference ICO clock operate.

(4) After the sensor stabilization wait time and the measurement time have elapsed and measurement is finished, a measurement result read request (CTSURD) is output.

(5) After a channel to be measured next is determined, a measurement channel setting request (CTSUWR) is output.

(6) After the stabilization wait time has elapsed and when the previous measurement is read, the result is cleared and measurement is started.

(7) Upon completion of all measurement channels, a measurement end interrupt (CTSUFN) is output and measurement is finished (transition to Status 0).

Table 43.7 lists the touch pin states in self-capacitance multi-scan mode.

Table 43.7 Touch Pin States in Self-Capacitance Multi-Scan Mode

Status	Touch Pin	
	Measurement Channel	Non-Measurement Channel
0	Low	Low
1	Low	Low
2	Low	Low
3	Pulse	Low
4	Pulse	Low
5	Low	Low

Downloaded from Arrow.com.

43.3.2.5 Mutual Capacitance Full Scan Mode Operation

In mutual capacitance full scan mode, measurement is performed during the high-level period of the sensor drive pulse on the receive channel by applying the edge to the target transmit channel to be measured. A single measurement target is measured twice, at the rising and falling edges. The difference between the data of these two measurements is used to determine whether or not the electrode is touched, thus achieving higher touch sensitivity.
Electrostatic capacitance is measured sequentially on channels set to transmission or reception specified by the CTSUCHTRCn registers (n = 0 to 4), and measurement targets specified by the CTSUCHACn registers. Electrostatic capacitance is measured by combining signals from the measurement target pins that are allocated to transmission or reception. Figure 43.16 shows the software flowchart and an operation example, and Figure 43.17 shows the timing chart.

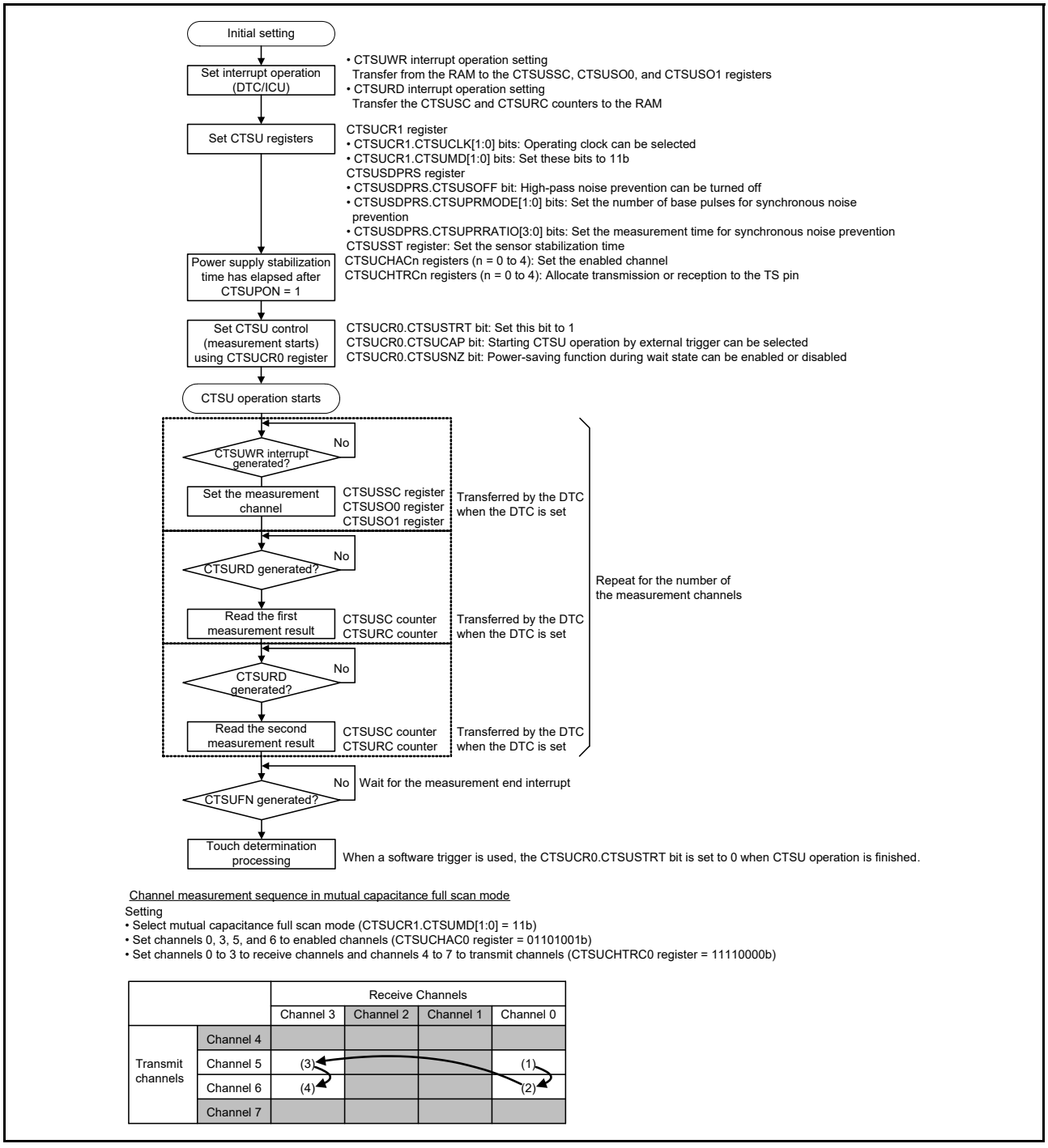


Figure 43.16 Software Flowchart and Operation Example of Mutual Capacitance Full Scan Mode

Downloaded from Arrow.com.

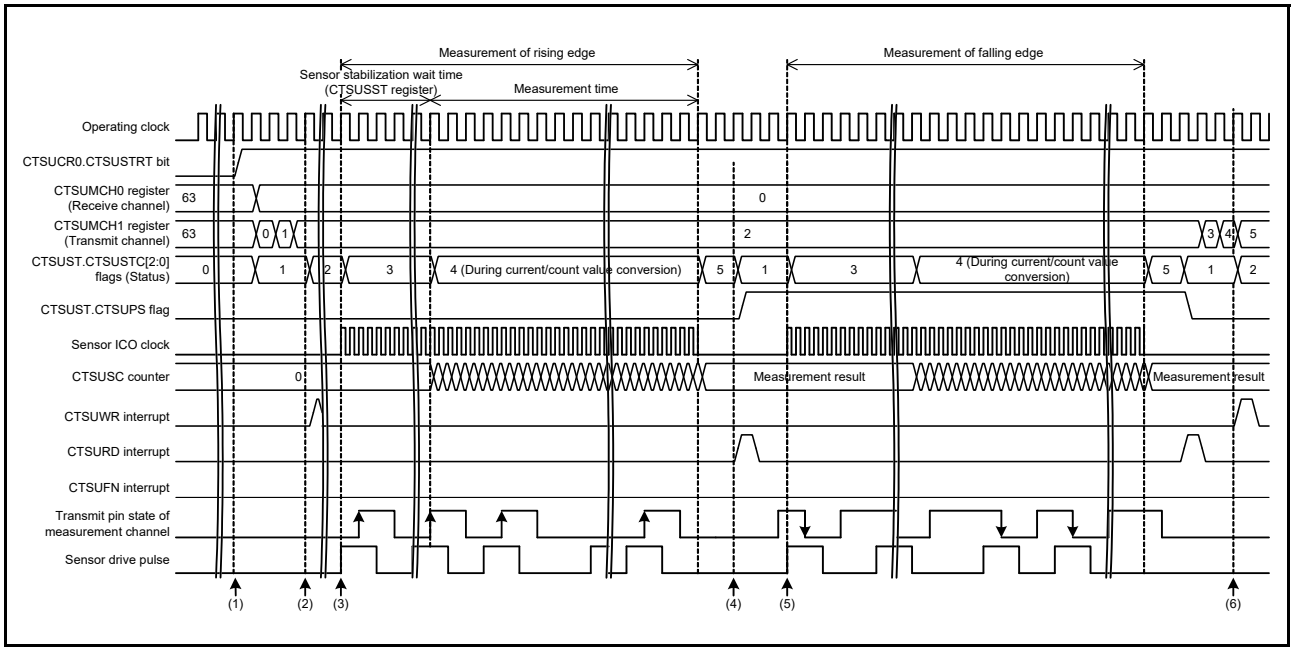


Figure 43.17 Timing Chart of Mutual Capacitance Full Scan Mode (Measurement Start Condition is Software Trigger)

The following describes operation shown in the timing chart in Figure 43.17.

(1) After various settings are made, operation is started by writing 1 to the CTSUCR0.CTSUSTRT bit.

(2) After a channel to be measured is determined according to the preset conditions, a request for setting the corresponding channel (CTSUWR) is output.

(3) Upon completion of writing the measurement channel settings (CTSUSSC, CTSUSO0, and CTSUSO1 registers), the sensor drive pulse is output and the sensor ICO clock and the reference ICO clock operate.
At the same time, a pulse which is handled as the rising edge is output to the transmit pin on the measurement channel during the high-level period of the sensor drive pulse.

(4) After the sensor stabilization wait time and the measurement time have elapsed and measurement is finished, a measurement result read request (CTSURD) is output.

(5) The same channel is measured by outputting a pulse that is handled as the falling edge during the high-level period of the sensor drive pulse.

(6) After the same channel is measured twice, a channel to be measured next is determined and measured in the similar way.

(7) Upon completion of all measurement channels, a measurement end interrupt (CTSUFN) is output and measurement is finished (transition to Status 0).

The mutual capacitance measurement status flag (CTSUST.CTSUPS flag) is changed when Status 5 transitions to Status 1.

Downloaded from Arrow.com.

Table 43.8 lists the touch pin states in mutual capacitance full scan mode.

Table 43.8 Touch Pin States in Mutual Capacitance Full Scan Mode

Status	Touch Pin of Receive Channel		Touch Pin of Transmit Channel		Remarks
	Measurement Channel	Non-Measurement Channel	Measurement Channel	Non-Measurement Channel	
0	Low	Low	Low	Low	—
1	Low	Low	Low/High	Low	—
2	Low	Low	Low	Low	—
3	Pulse	Low	Pulse	Low	Pulse of the phase same as that of the receive channel at the first measurement Pulse of the phase opposite to that of the receive channel at the second measurement
4	Pulse	Low	Pulse	Low	—
5	Low	Low	Low	Low	—

Downloaded from Arrow.com.

43.3.3 Items Common to Multiple Modes

43.3.3.1 Sensor Stabilization Wait Time and Measurement Time

Figure 43.18 shows the timing chart of the sensor stabilization wait time and measurement time.

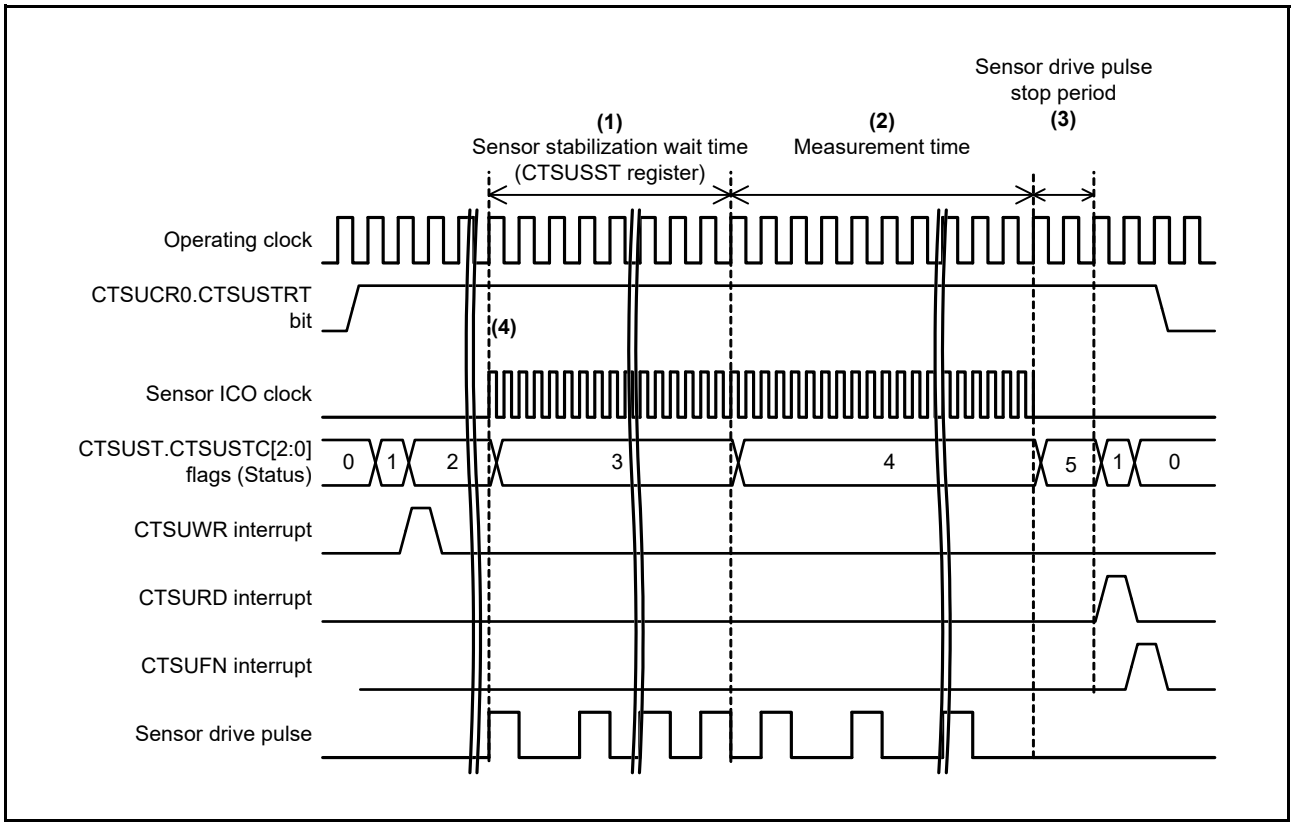


Figure 43.18 Sensor Stabilization Wait Time and Measurement Time

(1) In response to the CTSUWR interrupt request, output of the sensor drive pulse is started by write access to the CTSUSO1 register. Then, wait for the stabilization time set in the CTSUSST register.

(2) When the sensor stabilization time has elapsed and the CTSUST.CTSUDTSR flag is set to 0, measurement is started at transition to Status 4. The measurement time is determined by setting the base clock cycle and the CTSUSDPRS.CTSUPRMODE[1:0], CTSUPRRATIO[3:0], and CTSUSO0.CTSUSNUM[5:0] bits. When the measurement time has elapsed, measurement of the corresponding channel is finished.

(3) After the measurement time has elapsed, the status transitions to Status 1 after two operating clock cycles and a CTSURD interrupt is generated, so read the data from the CTSUSC and CTSURC counters.
At this time, the sensor drive pulse is output at the low level. When measurement of all specified channels is completed, the CTSUCR0.CTSUSTRT bit becomes 0.

(4) The sensor ICO clock oscillates while the CTSUSTC[2:0] flags are 011b (Status 3) or 100b (Status 4).

Downloaded from Arrow.com.

43.3.3.2 Interrupts

There are three types of interrupts for the CTSU:

- Write request interrupt for setting registers for each channel (CTSUWR)
- Measurement data transfer request interrupt (CTSURD)
- Measurement end interrupt (CTSUFN)

(1) Write request interrupt for setting registers for each channel (CTSUWR)

Store the setting data for each measurement channel in the RAM, and set the DTC or ICU transfer corresponding to the CTSUWR interrupt in advance. The CTSUWR interrupt is output when Status 1 transitions to Status 2. Write the setting data of the corresponding channel from the RAM to the CTSUSSC, CTSUSO0, and CTSUSO1 registers (Figure 43.19). Since write access to the CTSUSO1 register controls a transition to the next status, be sure to set this register last.

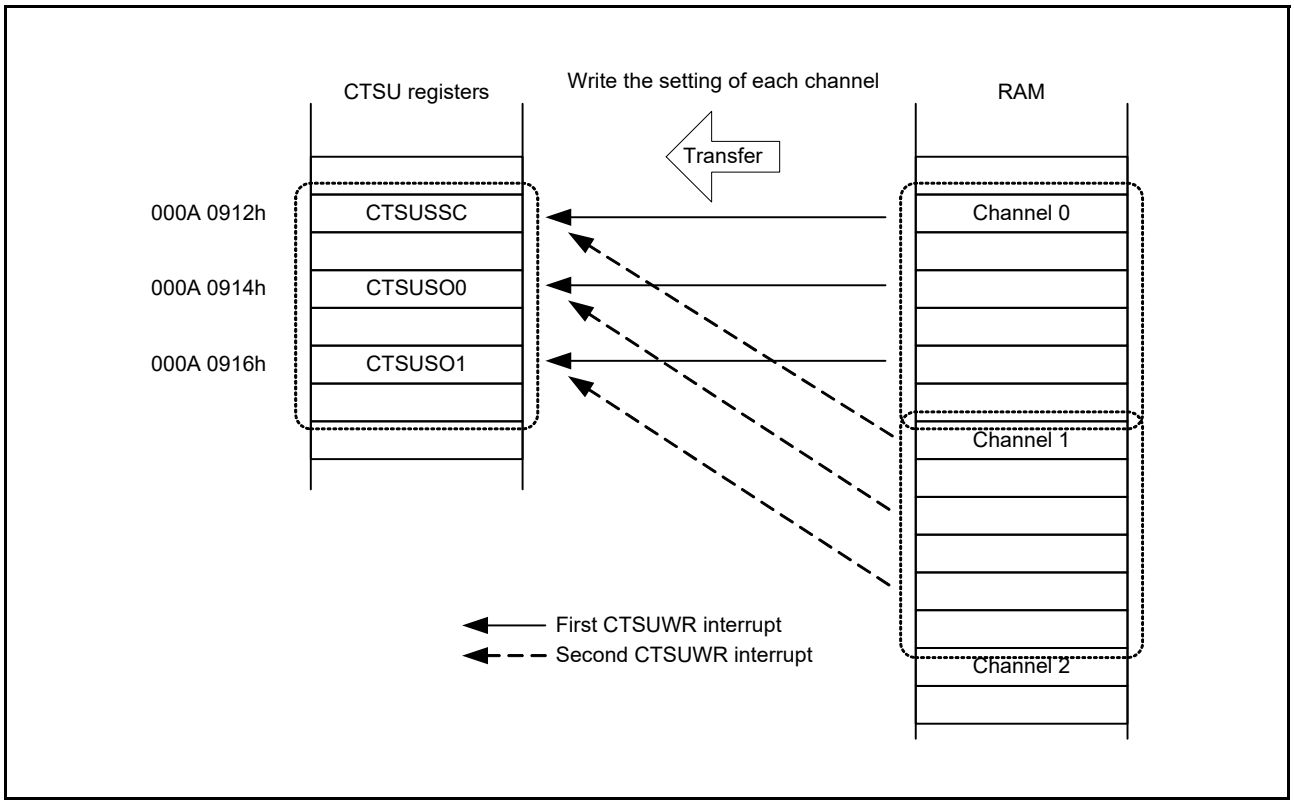


Figure 43.19 Example of DTC Transfer Operation Using CTSUWR Interrupt

The registers (CTSUSSC, CTSUSO0, and CTSUSO1 registers) to be set are allocated at sequential addresses. Set the operation at interrupt generation as shown below:

- Transfer destination address: Address of the CTSUSSC register
- Handling at the transfer destination address: Transfer 2-byte data three times by a single interrupt. (The address of the start byte is fixed.)
- Transfer source address: CTSUSSC register data storage address for the minimum channel in the setting data stored in the RAM
- Handling at the transfer source address: Transfer 2-byte data three times by a single interrupt. (The address of the first byte is continued from the previous interrupt handling.)
- Number of transfers by an interrupt: Specify the number of measurements.

Downloaded from Arrow.com.

(2) Measurement data transfer request interrupt (CTSURD)

Set DTC or ICU transfer corresponding to the CTSURD interrupt in advance. The CTSURD interrupt is output when Status 5 transitions to Status 1. Read the measurement result from the CTSUSC and CTSURC counters (Figure 43.20).

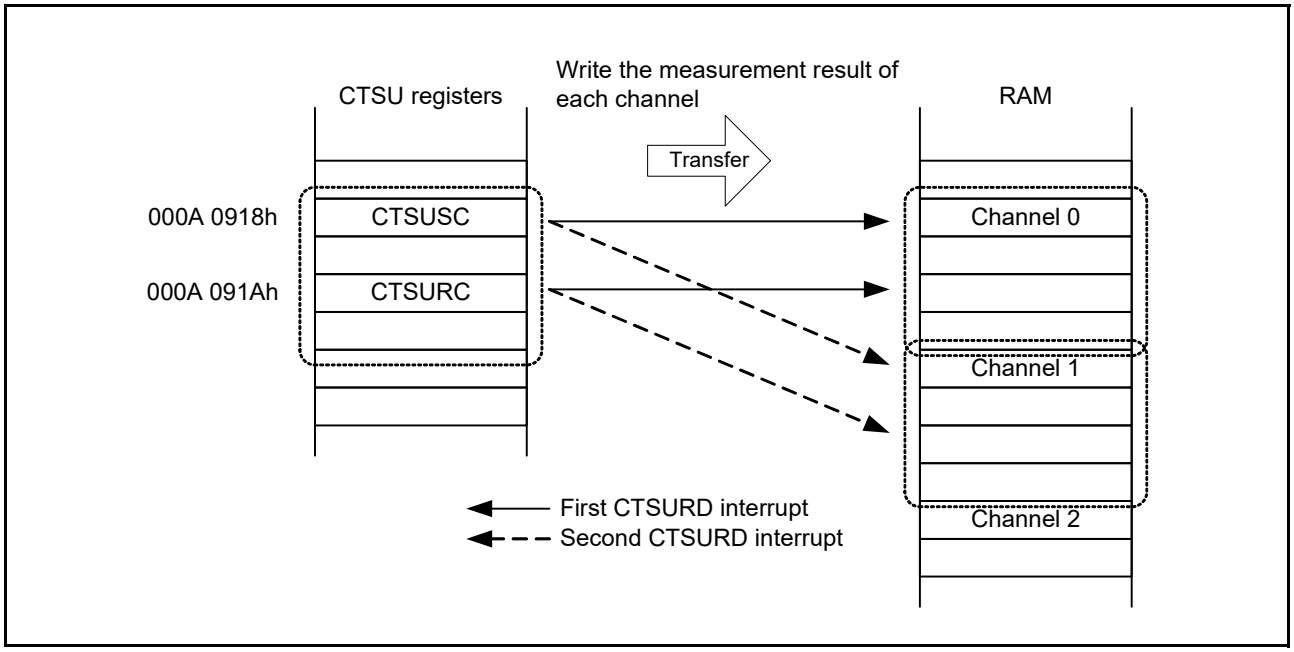


Figure 43.20 Example of DTC Transfer Operation Using CTSURD Interrupt

The measurement result registers (CTSUSC and CTSURC counters) used as transfer sources are allocated at sequential addresses. Set the operation at interrupt generation as shown below:

- Transfer source address: Address of the CTSUSC counter
- Handling at the transfer source address: Transfer 2-byte data twice by a single interrupt. (The start address is fixed.)
- Transfer destination address: CTSUSC counter data storage address for the minimum channel in the setting data stored in the RAM.
- Handling at the transfer destination address: Transfer 2-byte data twice by a single interrupt. (The start address is continued from the previous interrupt handling.)
- Number of transfers by an interrupt: Specify the number of measurements.

(3) Measurement end interrupt (CTSUFN)

When all channels are measured, an interrupt is generated when Status 1 transitions to Status 0. Use software to confirm the overflow flags (CTSUST.CTSUSOVF and CTSUROVF flags) and read the measurement results to determine whether or not the electrode is touched.

Interrupt requests are accepted or disabled in the interrupt control block.

Downloaded from Arrow.com.

43.4 Usage Notes

43.4.1 Measurement Result Data (CTSUSC and CTSURC Counters)

Read access during measurement is prohibited. If the measurement result data is accessed, an incorrect value may be read due to asynchronous operation.

43.4.2 Software Trigger

When 10b (PCLK/4) is selected by the CTSUCR1.CTSUCLK[1:0] bits, to restart measurement by writing 1 to the CTSUCR0.CTSUSTRT bit after measurement has been completed, wait for at least three cycles to elapse after an interrupt is generated, and then write to the CTSUCR0.CTSUSTRT bit.

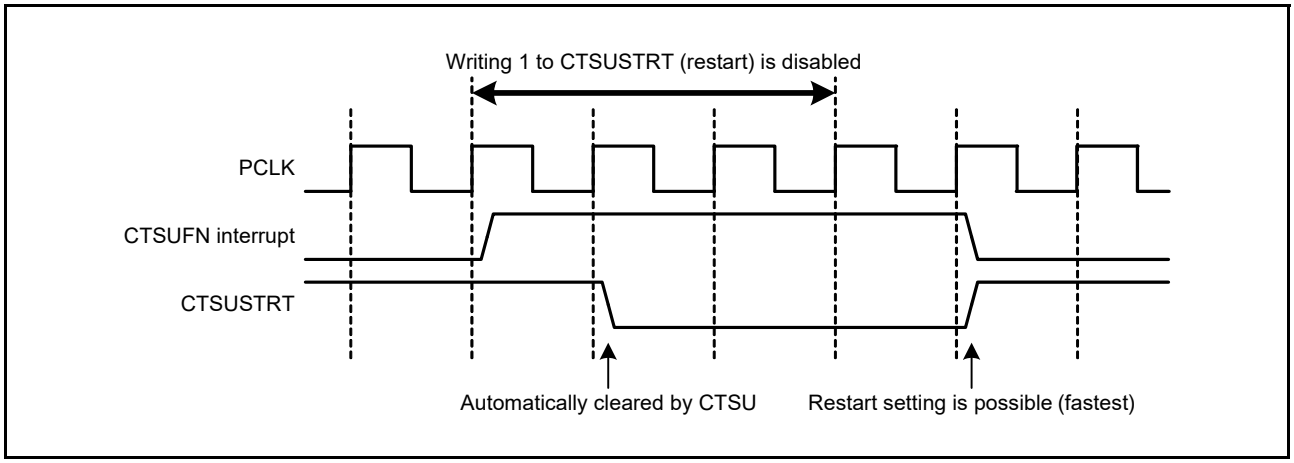


Figure 43.21 Notes on Restarting Measurement

43.4.3 External Trigger

- If an external trigger is input during the measurement time, measurement is not started. The next external event is enabled after one cycle of the operating clock when a CTSUFN interrupt is generated.
- To stop external trigger mode, write 0 to the CTSUCR0.CTSUSTRT bit and 1 to the CTSUCR0.CTSUINIT bit at the same time (forced stop).

Downloaded from Arrow.com.

43.4.4 Notes on Forcibly Stopping Operation

To forcibly stop the current operation, write 0 to the CTSUCR0.CTSUSTRT bit and 1 to the CTSUCR0.CTSUINIT bit at the same time. After this setting, the operation is stopped and the internal control registers are initialized.
When the CTSUCR0.CTSUINIT bit is used for initialization, the following registers are initialized in addition to the initialization of the internal measurement state.

- CTSUMCH0 register
- CTSUMCH1 register
- CTSUST register
- CTSUSC counter
- CTSURC counter

If operation is forcibly stopped, an interrupt request may be generated depending on the internal state. After operation is forcibly stopped, perform the processing for stopping/disabling the DTC or ICU.

If DTC transfer is stopped in the mounted system for some reason, also perform the processing for forcibly stopping and initializing the CTSU.

43.4.5 TSCAP Pin

The TSCAP pin requires an external decoupling capacitor to stabilize CTSU internal voltage. The traces between the TSCAP pin and the capacitor, and the capacitor and ground should be as short and wide as physically possible.
The capacitor connected to the TSCAP pin should be fully discharged using I/O port control to output a low level, before turning on the switch (CTSUCR1.CTSUCSW bit = 1) to establish a connection.

43.4.6 Notes during Measurement Operation (CTSUCR0.CTSUSTRT Bit = 1)

During measurement operation (CTSUCR0.CTSUSTRT bit = 1), do not use settings such as “stop the peripheral module clock” or “change the port settings related to the touch pins (TS and TSCAP pins)” in the higher layers of the system.
If control settings non-compliant to these restrictions are made, after operation is forcibly stopped (CTSUCR0.CTSUSTRT bit = 0 and CTSUCR0.CTSUINIT bit = 1), write 0 to the CTSUCR1.CTSUPON bit and 0 to the CTSUCR1.CTSUCSW bit at the same time, and set the CTSUCR0.CTSUSNZ bit to 0. Then, restart from the initial setting flow shown in Figure 43.9.

Downloaded from Arrow.com.

44. 12-Bit A/D Converter (S12ADE)

44.1 Overview

This MCU incorporates one unit of a 12-bit successive approximation A/D converter. Up to 14 channel analog inputs, temperature sensor output, and internal reference voltage are selectable for conversion.

The 12-bit A/D converter converts a maximum of 14 selected channels of analog inputs, temperature sensor output, and internal reference voltage, which have been selected, into a 12-bit digital value through successive approximation.

The A/D converter has three operating modes: single scan mode in which the analog inputs of up to 14 arbitrarily selected channels are converted only once in ascending channel order; and continuous scan mode in which the analog inputs of up to 14 arbitrarily selected channels are continuously converted in ascending channel order; and group scan mode in which up to 14 channels of the analog inputs are arbitrarily divided into two groups (group A and group B) and converted in ascending channel order in each group.

In group scan mode, the conditions for scanning start of group A and group B (synchronous trigger) can be independently selected, thus allowing A/D conversion of group A and group B to be started independently. When group-A priority control is selected along with operation as described above, if a request to start scanning for group A is received during A/D conversion for group B, the conversion operation for group B is discontinued and the conversion for group A starts, which is given priority.

In double trigger mode, one analog input channel arbitrarily selected is converted in single scan mode or group scan mode (group A), and the resulting data of A/D conversion started by the first and second synchronous triggers are stored into different registers (duplication of A/D conversion data).

Self-diagnosis is executed once at the beginning of each scan, and one of the three voltages internally generated in the 12-bit A/D converter is converted.

It is prohibited to simultaneously select both temperature sensor output and internal reference voltage. Perform A/D conversion independently for the temperature sensor output or the internal reference voltage.

The external pin input (VREFH0) or the analog reference voltage (AVCC0) is selectable as the reference voltage on the high-potential side. The external pin input (VREFL0) or the analog reference voltage (AVSS0) is selectable as the reference voltage on the low-potential side.

This IP has a compare function (window A and window B). This function is used to specify the high-side reference value and low-side reference value for window A and window B, respectively. When the A/D-converted value of the selected channel meets the comparison conditions, the ELC event (S12ADWMELC/S12ADWUMELC) is output according to the event conditions (A or B, A and B, A exor B). Furthermore, the comparator operation to compare the A/D-converted value with the low-side reference value is also enabled.

The A/D data storage buffer is a ring buffer consisting of 16 buffers to sequentially store A/D converted data.

Table 44.1 lists the specifications of the 12-bit A/D converter and Table 44.2 lists the functions of the 12-bit A/D converter. Figure 44.1 shows a block diagram of the 12-bit A/D converter.

Downloaded from Arrow.com.

Table 44.1 Specifications of 12-Bit A/D Converter (1/2)

Item	Description
Number of units	One unit
Input channels	Up to 14 channels
Extended analog function	Temperature sensor output, internal reference voltage
A/D conversion method	Successive approximation method
Resolution	12 bits
Conversion time	0.83 µs per channel (when A/D conversion clock ADCLK = 54 MHz)
A/D conversion clock	Peripheral module clock PCLKB*1 and A/D conversion clock ADCLK*1 can be set so that the frequency ratio should be one of the following. PCLKB to ADCLK frequency ratio = 1:1, 1:2, 2:1, 4:1, 8:1 ADCLK is set using the clock generation circuit.
Data registers	• 14 registers for analog input, 1 for A/D-converted data duplication in double trigger mode • One register for temperature sensor output • One register for internal reference voltage • One register for self-diagnosis • The results of A/D conversion are stored in 12-bit A/D data registers. • 12-bit accuracy output for the results of A/D conversion • The value obtained by adding up A/D-converted results is stored as a value in the number of bit for conversion accuracy + 2 bits/4 bits*2 in the A/D data registers in A/D-converted value addition mode. • Double trigger mode (selectable in single scan and group scan modes): The first piece of A/D-converted analog-input data on one selected channel is stored in the data register for the channel, and the second piece is stored in the duplication register.
Operating modes	• Single scan mode: A/D conversion is performed only once on the analog inputs of up to 14 channels arbitrarily selected. A/D conversion is performed only once on the temperature sensor output. A/D conversion is performed only once on the internal reference voltage. • Continuous scan mode: A/D conversion is performed repeatedly on the analog inputs of up to 14 channels arbitrarily selected. • Group scan mode: Analog inputs of up to 14 channels arbitrarily selected, are divided into group A and group B, and A/D conversion of the analog input selected on a group basis is performed only once. The conditions for scanning start of group A and group B (synchronous trigger) can be independently selected, thus allowing A/D conversion of group A and group B to be started independently. • Group scan mode (when group A is given priority): If a group A trigger is input during A/D conversion on group B, the A/D conversion on group B is stopped and A/D conversion is performed on group A. Restart (rescan) of A/D conversion on group B after completion of A/D conversion on group A can be set.
Conditions for A/D conversion start	• Software trigger • Synchronous trigger Trigger by the multi-function timer pulse unit (MTU), the event link controller (ELC), or the 16-bit timer pulse unit (TPU). • Asynchronous trigger A/D conversion can be triggered by the external trigger ADTRG0# pin.
Functions	• Variable sampling state count • Self-diagnosis of 12-bit A/D converter • Selectable A/D-converted value addition mode or average mode • Analog input disconnection detection function (discharge function/precharge function) • Double trigger mode (duplication of A/D conversion data) • Automatic clear function of A/D data registers • Compare function (window A and window B) • 16 ring buffers when the compare function is used

Downloaded from Arrow.com.

Table 44.1 Specifications of 12-Bit A/D Converter (2/2)

Item	Description
Interrupt sources	• In the modes except double trigger mode and group scan mode, A/D scan end interrupt request (S12ADI0) can be generated on completion of single scan. • In double trigger mode, A/D scan end interrupt request (S12ADI0) can be generated on completion of double scan. • In group scan mode, an A/D scan end interrupt request (S12ADI0) can be generated on completion of group A scan, whereas an A/D scan end interrupt request (GBADI) for group B can be generated on completion of group B scan. • When double trigger mode is selected in group scan mode, A/D scan end interrupt request (S12ADI0) can be generated on completion of double scan of group A, whereas A/D scan end interrupt request (GBADI) specially for group B can be generated on completion of group B scan. • The S12ADI0 and GBADI interrupts can activate the DMA controller (DMAC) and the data transfer controller (DTC).
Event link function	• An ELC event is generated on completion of scans other than group B scan in group scan mode. • An ELC event is generated on completion of group B scan in group scan mode. • An ELC event is generated on completion of all scans. • Scan can be started by a trigger output by the ELC. • An ELC event is generated according to the event conditions of the window compare function in single scan mode.
Low power consumption function	• Module stop state can be set.*3, *4

Note 1. The peripheral module clock PCLKB frequency is set according to the setting of the SCKCR.PCKB[3:0] bits and the A/D conversion clock ADCLK frequency is set according to the setting of the SCKCR.PCKD[3:0] bits.
Note 2. The number of extended bits during addition differs depending on the addition count.
2-bit extension: 1-time to 4-time conversion (addition zero to three times)
4-bit extension: 16-time conversion (addition 15 times)
Note 3. See section 11, Low Power Consumption for details.
Note 4. Wait for 1 µs or longer to start A/D conversion after release from the module stop state.

Table 44.2 Functions of 12-Bit A/D Converter

Item			Pin Name, Abbreviation
Analog input channels			AN000 to AN007, AN016 to AN020, AN027, temperature sensor output, internal reference voltage
Conditions for A/D conversion start	Software	Software trigger	Enabled
	Asynchronous trigger	ADTRG0#	Enabled
	Synchronous trigger	Compare match/input capture from MTU0.TGRA	TRG0AN
		Compare match/input capture from MTU0.TGRB	TRG0BN
		Compare match/input capture from MTU0 to MTU4.TGRA or underflow (trough) of MTU4.TCNT in complementary PWM mode	TRGAN
		Compare match from MTU0.TGRE	TRG0EN
		Compare match from MTU0.TGRF	TRG0FN
		Compare match between MTU4.TADCORA and MTU4.TCNT (interrupt skipping function)	TRG4AN
		Compare match between MTU4.TADCORB and MTU4.TCNT (interrupt skipping function)	TRG4BN
		Compare match between MTU4.TADCORA and MTU4.TCNT, or compare match between MTU4.TADCORB and MTU4.TCNT (interrupt skipping function)	TRG4ABN
		TGRA compare match/input capture from TPU0 to TPU4 or TGRA compare match/input capture from TPU0	TRGAN1 TRG4ABN1
		ELC trigger	Enabled
Interrupt			S12ADI0, GBADI interrupt
Setting of the module stop function*1			MSTPCRA.MSTPA17 bit

Note 1. See section 11, Low Power Consumption for details.

Downloaded from Arrow.com.

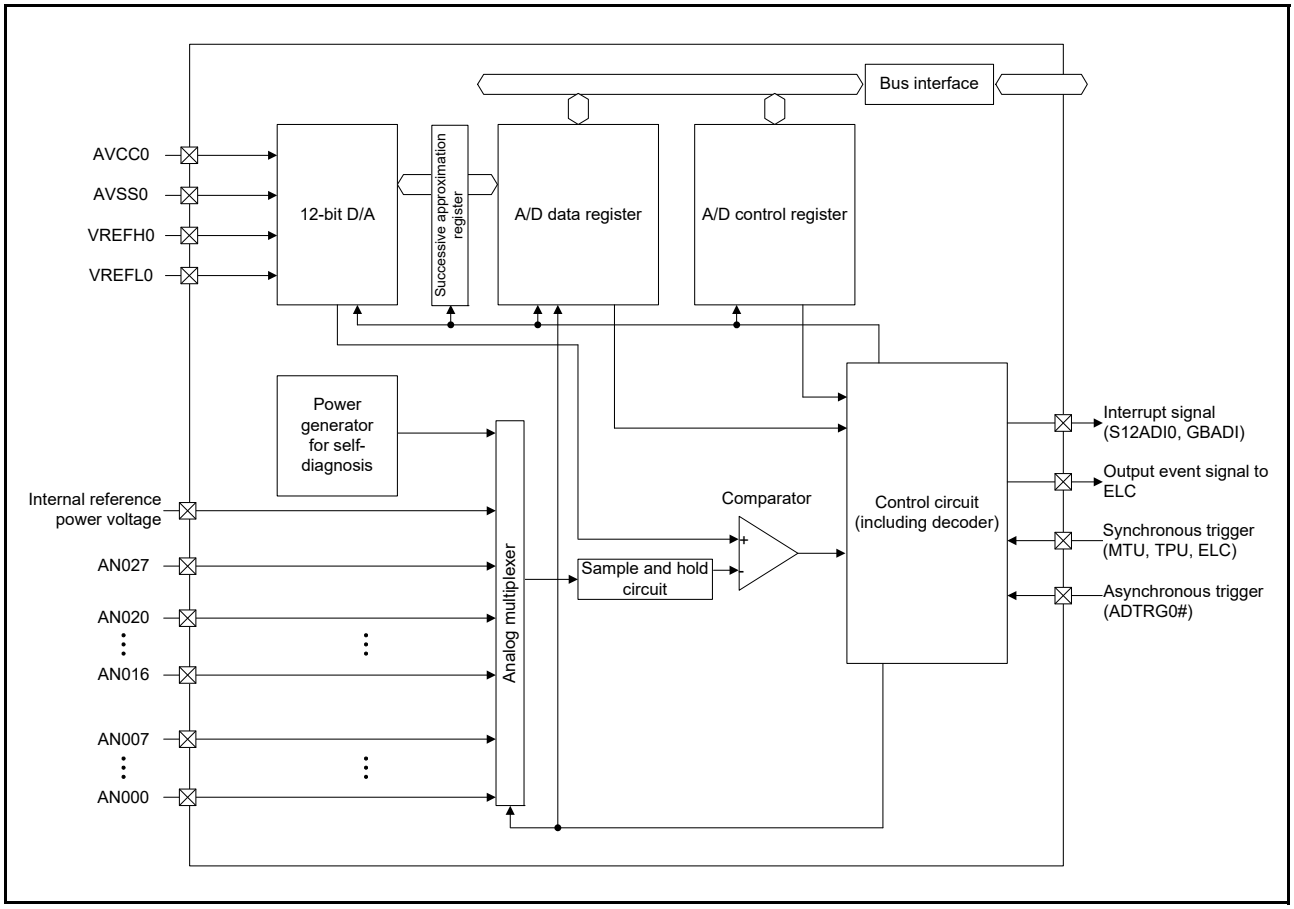


Figure 44.1 Block Diagram of 12-Bit A/D Converter

Table 44.3 lists the input pins of the 12-bit A/D converter.

Table 44.3 Pin Configuration of 12-Bit A/D Converter

Pin Name	I/O	Function
AVCC0	Input	Analog block power supply pin
AVSS0	Input	Analog block ground pin
VREFH0	Input	Reference power supply pin
VREFL0	Input	Reference power supply ground pin
AN000 to AN007, AN016 to AN020, AN027	Input	Analog input pins 0 to 7, analog input pins 16 to 20, analog input pin 27
ADTRG0#	Input	External trigger input pin for starting A/D conversion

Downloaded from Arrow.com.

44.2 Register Descriptions

44.2.1 A/D Data Registers y (ADDRy) (y = 0 to 7, 16 to 20, 27), A/D Data Duplication Register (ADDBLDR), A/D Temperature Sensor Data Register (ADTSDR), A/D Internal Reference Voltage Data Register (ADOCDR)

Address(es): S12AD.ADDR0 0008 9020h, S12AD.ADDR1 0008 9022h, S12AD.ADDR2 0008 9024h, S12AD.ADDR3 0008 9026h, S12AD.ADDR4 0008 9028h, S12AD.ADDR5 0008 902Ah, S12AD.ADDR6 0008 902Ch, S12AD.ADDR7 0008 902Eh, S12AD.ADDR16 0008 9040h, S12AD.ADDR17 0008 9042h, S12AD.ADDR18 0008 9044h, S12AD.ADDR19 0008 9046h, S12AD.ADDR20 0008 9048h, S12AD.ADDR27 0008 9056h, S12AD.ADDBLDR 0008 9018h, S12AD.ADTSDR 0008 901Ah, S12AD.ADOCDR 0008 901Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The ADDRy registers (y = 0 to 7, 16 to 20, 27) are 16-bit read-only registers which store the A/D conversion results.
The ADDBLDR register is a 16-bit read-only register used in double trigger mode. The ADDBLDR register stores the results of A/D conversion when the conversion is started by the second trigger.
The ADTSDR register is a 16-bit read-only register that stores the A/D conversion results of the temperature sensor output.
The ADOCDR register is a 16-bit read-only register that stores the A/D conversion results of the internal reference voltage.
The format of each register differs depending on the conditions below.

- Settings of the A/D data register format select bit (ADCER.ADRFMT) (flush-right or flush-left)
- Settings of the addition count select bits (ADADC.ADC[2:0]) (addition once, twice, three, or 15 times)
- Settings of the average mode enable bit (ADADC.AVEE) (addition or average)

The data formats for each given condition are shown below.

(1) When A/D-Converted Value Addition/Average Mode is Not Selected

- Flush-right format
  The A/D-converted value is stored in bits 11 to 0. Bits 15 to 12 are read as 0.
- Flush-left format
  The A/D-converted value is stored in bits 15 to 4. Bits 3 to 0 are read as 0.

(2) When A/D-Converted Average Mode is Selected

- Flush-right format
  The mean value of the A/D-converted results of the same channel is stored in bits 11 to 0.
  Bits 15 to 12 are read as 0.
- Flush-left format
  The mean value of the A/D-converted results of the same channel is stored in bits 15 to 4.
  Bits 3 to 0 are read as 0.

A/D-converted value average mode can be set only when twice or four times is selected in A/D-converted value addition mode.

(3) When A/D-Converted Value Addition Mode is Selected

- Flush-right format (A/D-converted value addition mode and 1-time to 4-time conversion selected)
  The value added by the A/D-converted value of the same channel is stored in bits 13 to 0.
  Bits 15 and 14 are read as 0.
- Flush-right format (A/D-converted value addition mode and 16-time conversion selected)
  The value added by the A/D-converted value of the same channel is stored in bits 15 to 0.

Downloaded from Arrow.com.

- Flush-left format (A/D-converted value addition mode and 1-time to 4-time conversion selected)
  The value added by the A/D-converted value of the same channel is stored in bits 15 to 2.
  Bits 1 and 0 are read as 0.
- Flush-left format (A/D-converted value addition mode and 16-time conversion selected)
  The value added by the A/D-converted value of the same channel is stored in bits 15 to 0.

When A/D-converted addition mode is selected, the value added by the A/D-converted value of the same channel is indicated. The number of A/D conversions can be set to 1, 2, 3, 4, or 16 times. If A/D-converted addition mode is selected, when the conversion count is set to 1 to 4 times, the value added by the A/D conversion result is retained in the A/D data register as 2-bit extended data of the conversion accuracy bits; when the conversion count is set to 16 times, the value added by the A/D conversion result is retained in the A/D data register as 4-bit extended data of the conversion accuracy bits. Even if A/D-converted value addition mode is selected, the value is stored in the A/D data register according to the settings of the A/D data register format select bits.

Downloaded from Arrow.com.

44.2.2 A/D Self-Diagnosis Data Register (ADRD)

Address(es): S12AD.ADRD 0008 901Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The ADRD register is a 16-bit read-only register that stores the A/D conversion results based on the 12-bit A/D converter's self-diagnosis. In addition to the A/D-converted value, the self-diagnosis status is included in. In the ADRD register, the different formats are used depending on the conditions below.

- Settings of the A/D data register format select bit (ADCER.ADRFMT) (flush-right or flush-left)

The A/D-converted value addition mode and A/D-converted value average mode cannot be applied to the A/D self-diagnosis function. For details of self-diagnosis, see section 44.2.11, A/D Control Extended Register (ADCER).

The data formats for each given condition are shown below.

- Flush-right format
  The A/D-converted value is stored in bits 11 to 0. The self-diagnosis status is stored in bits 15 and 14.
  Bits 13 and 12 are read as 0.
- Flush-left format
  The A/D-converted value is stored in bits 15 to 4. The self-diagnosis status is stored in bits 1 and 0.
  Bits 3 and 2 are read as 0.

Table 44.4 Self-Diagnosis Status Description

Bits 15 and 14 for flush-right format setting Bits 1 and 0 for flush-left format setting	Self-diagnosis status
00b	Self-diagnosis has never been executed since power-on.
01b	Self-diagnosis using the voltage of 0 V has been executed.
10b	Self-diagnosis using the reference voltage × 1/2 has been executed.
11b	Self-diagnosis using the reference voltage has been executed.

Note: For details of self-diagnosis, see section 44.2.11, A/D Control Extended Register (ADCER).

Downloaded from Arrow.com.

44.2.3 A/D Control Register (ADCSR)

Address(es): S12AD.ADCSR 0008 9000h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ADST	ADCS[1:0]		ADIE	—	ADHSC	TRGE	EXTRG	DBLE	GBADIE	—	DBLANS[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	DBLANS[4:0]	Double Trigger Channel Select	These bits select one analog input channel for double triggered operation. The setting is only effective while double trigger mode is selected.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	GBADIE	Group B Scan End Interrupt Enable	0: Disables GBADI interrupt generation upon group B scan completion. 1: Enables GBADI interrupt generation upon group B scan completion.	R/W
b7	DBLE	Double Trigger Mode Select	0: Deselects double trigger mode. 1: Selects double trigger mode.	R/W
b8	EXTRG	Trigger Select *1	0: A/D conversion is started by synchronous trigger. 1: A/D conversion is started by asynchronous trigger.	R/W
b9	TRGE	Trigger Start Enable	0: Disables A/D conversion to be started by synchronous or asynchronous trigger. 1: Enables A/D conversion to be started by synchronous or asynchronous trigger.	R/W
b10	ADHSC	A/D Conversion Select	0: High-speed conversion 1: Low-current conversion	R/W
b11	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b12	ADIE	Scan End Interrupt Enable	0: Disables S12ADI0 interrupt generation upon scan completion. 1: Enables S12ADI0 interrupt generation upon scan completion.	R/W
b14, b13	ADCS[1:0]	Scan Mode Select	b14 b13 0 0: Single scan mode 0 1: Group scan mode 1 0: Continuous scan mode 1 1: Setting prohibited	R/W
b15	ADST	A/D Conversion Start	0: Stops A/D conversion process. 1: Starts A/D conversion process.	R/W

Note 1. Starting A/D conversion using an external pin (asynchronous trigger)
After a high-level signal is input to the external pin (ADTRG0#), write 1 to both the TRGE and EXTRG bits in ADCSR and change the signals of ADTRG0# to low. Thus the falling edge of ADTRG0# is detected and the scan conversion process is started. In this case, the pulse width of the low-level input must be at least 1.5 clock cycles of PCLKB.

The ADCSR register sets double trigger mode, A/D conversion start trigger; enables/disables scan end interrupt; selects the scan mode; and starts or stops A/D conversion.

DBLANS[4:0] Bits (Double Trigger Channel Select)

The DBLANS[4:0] bits select one of the channels for A/D conversion data duplication in double trigger mode. The A/D conversion results of the analog input of the channel selected by the DBLANS[4:0] bits are stored into the A/D data register y when conversion is started by the first trigger, and into the A/D data duplication register when started by the second trigger. Table 44.5 shows selection of the channel for double triggered operation.
When double trigger mode is selected, channel selection using the ADANSA0 and ADANSA1 registers is invalid, and the channel selected by the DBLANS[4:0] bits is subjected to A/D conversion instead.
When double trigger mode is used, do not select A/D conversion for the self-diagnosis function, temperature sensor output, and internal reference voltage (temperature sensor output and internal reference voltage can be selected for A/D conversion for group B in group scan mode). The DBLANS[4:0] bits should be set while the ADST bit is 0. They should

Downloaded from Arrow.com.

not be set simultaneously when 1 is written to the ADST bit.
To enter A/D-converted value addition/average mode while double trigger mode is set, the channel selected by the DBLANS[4:0] bits should be selected in the ADANSA0 and ADANSA1 registers.

Table 44.5 Relationship between DBLANS[4:0] Bits Settings and Double Trigger Enabled Channels

DBLANS[4:0]	Duplication Channel	DBLANS[4:0]	Duplication Channel
00000b	AN000	10000b	AN016
00001b	AN001	10001b	AN017
00010b	AN002	10010b	AN018
00011b	AN003	10011b	AN019
00100b	AN004	10100b	AN020
00101b	AN005	11011b	AN027
00110b	AN006		
00111b	AN007		

GBADIE Bit (Group B Scan End Interrupt Enable)

The GBADIE bit enables or disables group B scan end interrupt (GBADI) in group scan mode.

DBLE Bit (Double Trigger Mode Select)

Double trigger mode has a function to store the resulting data of A/D conversion started by the first and second synchronous triggers into separate registers.
When double trigger mode is selected, channel selection using the ADANSA0 and ADANSA1 registers is invalid and the channel selected by the DBLANS[4:0] bits is effective instead. Double trigger mode can be only operated by the synchronous trigger selected by the ADSTRGR.TRSA[5:0] bits. Do not generate an asynchronous or software trigger.
The A/D conversion results started by the first trigger are stored into the A/D data register y and those started by the second trigger are stored into the A/D data duplication register. In this case, if the ADIE bit is set to 1, the interrupt is generated not upon completion of the first conversion but upon completion of the second conversion.
In continuous scan mode, double trigger mode should not be selected.
The DBLE bit should be set after the ADST bit has been set to 0.

EXTRG Bit (Trigger Select)

The EXTRG bit selects the synchronous trigger or the asynchronous trigger as the trigger for starting A/D conversion.

TRGE Bit (Trigger Start Enable)

The TRGE bit enables or disables A/D conversion by the synchronous trigger and the asynchronous trigger.
This bit should be set to 1 in group scan mode.

ADHSC Bit (A/D Conversion Select)

The ADHSC bit sets the operating mode of A/D conversion. When modifying this bit, set the 12-bit converter to the standby state. For the procedure for modifying the ADHSC bit, see section 44.8.10, ADHSC Bit Rewriting Procedure.

ADIE Bit (Scan End Interrupt Enable)

The ADIE bit enables or disables the A/D scan end interrupt (S12ADI0) in scans except for group B scan in group scan mode.
With double trigger mode deselected, the S12ADI0 interrupt is generated after the first scan is completed if the ADIE bit is set to 1.

Downloaded from Arrow.com.

With double trigger mode selected, the S12ADI0 interrupt is generated after the second scan is completed if the ADIE bit is set to 1 as long as the scan is started by the synchronous trigger selected by the ADSTRGR.TRSA[5:0] bits.

ADCS[1:0] Bits (Scan Mode Select)

The ADCS[1:0] bits select the scan mode.
In single scan mode, A/D conversion is performed for the analog inputs of a maximum of 14 channels selected with the ADANSA0 and ADANSA1 registers in the ascending order of the channel number, and when one cycle of A/D conversion is completed for all the selected channels, the scan conversion is stopped.
In continuous scan mode, while the ADCSR.ADST bit is 1, A/D conversion is performed for the analog inputs of a maximum of 14 channels selected with the ADANSA0 and ADANSA1 registers in the ascending order of the channel number, and when one cycle of A/D conversion is completed for all the selected channels, A/D conversion is repeated from the first channel. If the ADCSR.ADST bit is set to 0 during continuous scan, A/D conversion is stopped even if scanning is in progress.
In group scan mode, A/D conversion is performed for the analog inputs (group A) of 14 channels selected with the ADANSA0 and ADANSA1 registers in the ascending order of the channel number after scanning is started by the synchronous trigger selected by the ADSTRGR.TRSA[5:0] bits, and when one cycle of A/D conversion is completed for all the selected channels, A/D conversion is stopped. A/D conversion is also performed for the analog inputs (group B) of a maximum of 14 channels selected with the ADANSB0 and ADANSB1 registers in the ascending order of the channel number after scanning is started by the synchronous trigger selected by the ADSTRGR.TRSB[5:0] bits, and when one cycle of A/D conversion is completed for all the selected channels, A/D conversion is stopped.
When selecting group scan mode, different channels and triggers should be selected for group A and group B.
When selecting the temperature sensor output or internal reference voltage, select single scan mode, and deselect all the channels selected with the ADANSA0 and ADANSA1 registers before performing A/D conversion. When A/D conversion of the selected temperature sensor output or internal reference voltage is completed, A/D conversion is stopped.
The ADCS[1:0] bits should be set while the ADST bit is 0. They should not be set simultaneously when 1 is written to the ADST bit.

ADST Bit (A/D Conversion Start)

The ADST bit starts or stops A/D conversion process.
Before the ADST bit is set to 1, set the A/D conversion clock, the conversion mode, and conversion target analog input.
[Setting conditions]

- 1 is written by software.
- The synchronous trigger selected by the ADSTRGR.TRSA[5:0] bits is detected with ADCSR.EXTRG and ADCSR.TRGE bits being set to 0 and 1, respectively.
- The synchronous trigger selected by the ADSTRGR.TRSB[5:0] bits is detected with the ADCSR.TRGE bit being set to 1 in group scan mode.
- The asynchronous trigger is detected with the ADCSR.TRGE and ADCSR.EXTRG bits being set to 1 and the ADSTRGR.TRSA[5:0] bits being set to 000000b.
- With group-A priority control operation mode enabled (ADCSR.ADCS[1:0] bits = 01b and ADGSPCR.PGS bit = 1), a group B trigger is detected and A/D conversion of group B is started.
- With group-A priority control operation mode enabled (ADCSR.ADCS[1:0] bits = 01b and ADGSPCR.PGS bit = 1), the ADGSPCR.GBRSCN bit is set to 1 and A/D conversion of group B is restarted.
- With group-A priority control operation mode enabled (ADCSR.ADCS[1:0] bits = 01b and ADGSPCR.PGS bit = 1), the ADGSPCR.GBRP bit is set to 1 and A/D conversion of group B is started.

[Clearing conditions]

- 0 is written by software.
- The A/D conversion of all the selected channels, the temperature sensor output, or the internal reference voltage is

Downloaded from Arrow.com.

completed in single scan mode.

- Group A scan is completed in group scan mode.
- Group B scan is completed in group scan mode.
- With group-A priority control operation mode enabled (ADCSR.ADCS[1:0] bits = 01b and ADGSPCR.PGS bit = 1), a group A trigger is detected during group B A/D conversion and the scanning of group B is stopped.
- With group-A priority control operation mode enabled (ADCSR.ADCS[1:0] bits = 01b and ADGSPCR.PGS bit = 1), the ADGSPCR.GBRSCN bit is set to 1 and the scanning of group B started by a resumption trigger is completed.
- With group-A priority control operation mode enabled (ADCSR.ADCS[1:0] bits = 01b and ADGSPCR.PGS bit = 1) the ADGSPCR.GBRP bit is set to 1 and the scanning of group B by a trigger is completed.

Note: When group-A priority control operation mode has been enabled (ADCSR.ADCS[1:0] bits = 01b and ADGSPCR.PGS bit = 1), do not set the ADST bit to 1.

Note: When group-A priority control operation mode has been enabled (ADCSR.ADCS[1:0] bits = 01b and ADGSPCR.PGS bit = 1) and ADGSPCR.GBRP = 1, do not set the ADST bit to 0. When forcibly terminating A/D conversion, follow the procedure for clearing the ADST bit.

44.2.4 A/D Channel Select Register A0 (ADANSA0)

Address(es): S12AD.ADANSA0 0008 9004h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	ANSA007	ANSA006	ANSA005	ANSA004	ANSA003	ANSA002	ANSA001	ANSA000
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ANSA000	A/D Conversion Channel Select	0: AN000 to AN007 are not subjected to conversion. 1: AN000 to AN007 are subjected to conversion.	R/W
b1	ANSA001			R/W
b2	ANSA002			R/W
b3	ANSA003			R/W
b4	ANSA004			R/W
b5	ANSA005			R/W
b6	ANSA006			R/W
b7	ANSA007			R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADANSA0 register selects analog input channels for A/D conversion among AN000 to AN007. In group scan mode, this register selects group A channels.

ANSA0n Bit (n = 00 to 07) (A/D Conversion Channel Select)

The ANSA0n bit selects analog input channels for A/D conversion among AN000 to AN007. The channels to be selected and the number of channels can be arbitrarily set. The ANSA000 bit corresponds to AN000 and the ANSA007 bit corresponds to AN007.

When performing A/D conversion of the temperature sensor output or internal reference voltage, do not select analog input channels. The setting value of this register should be 0000h.

When double trigger mode is selected, the channel selected by the ANSA0n bit is invalid, and the channel selected by the ADCSR.DBLANS[4:0] bits is selected in group A instead.

The ANSA0n bit should be set while the ADCSR.ADST bit is 0.

Downloaded from Arrow.com.

44.2.5 A/D Channel Select Register A1 (ADANSA1)

Address(es): S12AD.ADANSA1 0008 9006h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	ANSA111	—	—	—	—	—	—	ANSA104	ANSA103	ANSA102	ANSA101	ANSA100
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ANSA100	A/D Conversion Channel Select	0: AN016 to AN020 are not subjected to conversion. 1: AN016 to AN020 are subjected to conversion.	R/W
b1	ANSA101			R/W
b2	ANSA102			R/W
b3	ANSA103			R/W
b4	ANSA104			R/W
b10 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11	ANSA111	A/D Conversion Channel Select	0: AN027 is not subjected to conversion. 1: AN027 is subjected to conversion.	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADANSA1 register selects analog input channels for A/D conversion among AN016 to AN020 and AN027. In group scan mode, group A channels are to be selected.

ANSA1n Bit (n = 00 to 04, 11) (A/D Conversion Channel Select)

The ANSA1n bit select analog input channels for A/D conversion among AN016 to AN020 and AN027. The channels to be selected and the number of channels can be arbitrarily set. The ANSA100 bit corresponds to AN016 and the ANSA111 bit corresponds to AN027.

When performing A/D conversion of the temperature sensor output or internal reference voltage, do not select analog input channels. The setting value of this register should be 0000h.

When double trigger mode is selected, the channel selected by the ANSA1n bit is invalid, and the channel selected by the ADCSR.DBLANS[4:0] bits is selected in group A instead.

The ANSA1n bit should be set while the ADCSR.ADST bit is 0.

Downloaded from Arrow.com.

44.2.6 A/D Channel Select Register B0 (ADANSB0)

Address(es): S12AD.ADANSB0 0008 9014h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	ANSB007	ANSB006	ANSB005	ANSB004	ANSB003	ANSB002	ANSB001	ANSB000
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ANSB000	A/D Conversion Channel Select	0: AN000 to AN007 are not subjected to conversion. 1: AN000 to AN007 are subjected to conversion.	R/W
b1	ANSB001			R/W
b2	ANSB002			R/W
b3	ANSB003			R/W
b4	ANSB004			R/W
b5	ANSB005			R/W
b6	ANSB006			R/W
b7	ANSB007			R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADANSB0 register selects analog input channels for A/D conversion among AN000 to AN007 in group B when group scan mode is selected. The ADANSB0 register is not used in any scan mode other than group scan mode.

ANSB0n Bit (n = 00 to 07) (A/D Conversion Channel Select)

The ANSB0n bit selects analog input channels for A/D conversion among AN000 to AN007 in group B when group scan mode is selected. The ADANSB0 register is used for group scan mode only; not used for any other modes. The channels specified in group A (the channels corresponding to group A, selected with the ADANSA0 and ADANSA1 registers and the ADCSR.DBLANS[4:0] bits in double trigger mode) should be excluded as the channels to be selected and the number of channels to be set.

The ANSB000 bit corresponds to AN000 and the ANSB007 bit corresponds to AN007.

When performing A/D conversion of the temperature sensor output or internal reference voltage, do not select analog input channels. The setting value of this register should be 0000h.

The ANSB0n bit should be set while the ADCSR.ADST bit is 0.

Downloaded from Arrow.com.

44.2.7 A/D Channel Select Register B1 (ADANSB1)

Address(es): S12AD.ADANSB1 0008 9016h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	ANSB111	—	—	—	—	—	—	ANSB104	ANSB103	ANSB102	ANSB101	ANSB100
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ANSB100	A/D Conversion Channel Select	0: AN016 to AN020 are not subjected to conversion. 1: AN016 to AN020 are subjected to conversion.	R/W
b1	ANSB101			R/W
b2	ANSB102			R/W
b3	ANSB103			R/W
b4	ANSB104			R/W
b10 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11	ANSB111	A/D Conversion Channel Select	0: AN027 is not subjected to conversion. 1: AN027 is subjected to conversion.	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADANSB1 register selects analog input channels for A/D conversion among AN016 to AN020 and AN027 in group B when group scan mode is selected. The ADANSB1 register is not used in any scan mode other than group scan mode.

ANSB1n Bit (n = 00 to 04, 11) (A/D Conversion Channel Select)

The ANSB1n bit selects analog input channels for A/D conversion among AN016 to AN020 and AN027 in group B when group scan mode is selected. The ADANSB1 register is used for group scan mode only; not used for any other modes. The channels specified in group A (the channels corresponding to group A, selected with the ADANSA0 and ADANSA1 registers and the ADCSR.DBLANS[4:0] bits in double trigger mode) should be excluded as the channels to be selected and the number of channels to be set.

The ANSB100 bit corresponds to AN016 and the ANSB111 bit corresponds to AN027.

When performing A/D conversion of the temperature sensor output or internal reference voltage, do not select analog input channels. The setting value of this register should be 0000h.

The ANSB1n bit should be set while the ADCSR.ADST bit is 0.

Downloaded from Arrow.com.

44.2.8 A/D-Converted Value Addition/Average Function Select Register 0 (ADADS0)

Address(es): S12AD.ADADS0 0008 9008h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	ADS007	ADS006	ADS005	ADS004	ADS003	ADS002	ADS001	ADS000
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ADS000	A/D-Converted Value Addition/ Average Channel Select	0: A/D-converted value addition/average mode for AN000 to AN007 is not selected. 1: A/D-converted value addition/average mode for AN000 to AN007 is selected.	R/W
b1	ADS001			R/W
b2	ADS002			R/W
b3	ADS003			R/W
b4	ADS004			R/W
b5	ADS005			R/W
b6	ADS006			R/W
b7	ADS007			R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADADS0 register selects the channels 00 to 07 on which A/D conversion is performed successively 2, 3, 4, or 16 times and then converted values are added (integrated) or averaged.

ADS0n Bit (n = 00 to 07) (A/D-Converted Value Addition/Average Channel Select)

When the ADS0n bit of the number that is the same as that of A/D-converted channel selected by the ADANSA0.ANSA0n bit or ADCSR.DBLANS[4:0] bits and ADANSB0.ANSB0n bit is set to 1, A/D conversion of analog input of the selected channels is performed successively 2, 3, 4, or 16 times that is set with the ADADC.ADC[2:0] bits. When the ADADC.AVEE bit is 0, the value obtained by addition (integration) is stored in the A/D data register. When the ADADC.AVEE bit is 1, the mean value of the results obtained by addition (integration) is stored in the A/D data register. As for the channel on which the A/D conversion is performed and addition/average mode is not selected, a normal one-time conversion is executed and the conversion result is stored to the A/D data register. The ADS0n bit should be set while the ADCSR.ADST bit is 0.

Downloaded from Arrow.com.

44.2.9 A/D-Converted Value Addition/Average Function Select Register 1 (ADADS1)

Address(es): S12AD.ADADS1 0008 900Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	ADS111	—	—	—	—	—	—	ADS104	ADS103	ADS102	ADS101	ADS100
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ADS100	A/D-Converted Value Addition/ Average Channel Select	0: A/D-converted value addition/average mode for AN016 to AN020 is not selected. 1: A/D-converted value addition/average mode for AN016 to AN020 is selected.	R/W
b1	ADS101			R/W
b2	ADS102			R/W
b3	ADS103			R/W
b4	ADS104			R/W
b10 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11	ADS111	A/D-Converted Value Addition/ Average Channel Select	0: A/D-converted value addition/average mode for AN027 is not selected. 1: A/D-converted value addition/average mode for AN027 is selected.	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADADS1 register selects the channels 16 to 20 and 27 on which A/D conversion is performed successively 2, 3, 4, or 16 times and then converted values are added (integrated) or averaged.

ADS1n Bit (n = 00 to 04, 11) (A/D-Converted Value Addition/Average Channel Select)

When the ADS1n bit of the number that is the same as that of A/D-converted channel selected by the ADANSA1.ANSA1n bit or ADCSR.DBLANS[4:0] bits and ADANSB1.ANSB1n bit is set to 1, A/D conversion of analog input of the selected channels is performed successively 2, 3, 4, or 16 times that is set with the ADADC.ADC[2:0] bits. When the ADADC.AVEE bit is 0, the value obtained by addition (integration) is stored in the A/D data register. When the ADADC.AVEE bit is 1, the mean value of the results obtained by addition (integration) is stored in the A/D data register. As for the channel on which the A/D conversion is performed and addition/average mode is not selected, a normal one-time conversion is executed and the conversion result is stored to the A/D data register. The ADS1n bit should be set while the ADCSR.ADST bit is 0.

Figure 44.2 shows a scanning operation sequence in which both the ADS002 and ADS006 bits are set to 1. It is assumed that addition mode is selected (ADADC.AVEE = 0), the addition count is set to three times (ADADC.ADC[2:0] = 011b), and the channels AN000 to AN007 are selected (ADANSA0.ANSA0n = FFh) in continuous scan mode (ADCSR.ADCS[1:0] = 10b). The conversion process begins with AN000. The AN002 conversion is performed successively four times (addition three times), and the added (integrated) value is stored in A/D data register 2. After that the AN003 conversion is started. The AN006 conversion is performed successively 4 times and the added (integrated) value is stored in A/D data register 6. After conversion of AN007, the conversion operation is once again performed in the same sequence from AN000.

Downloaded from Arrow.com.

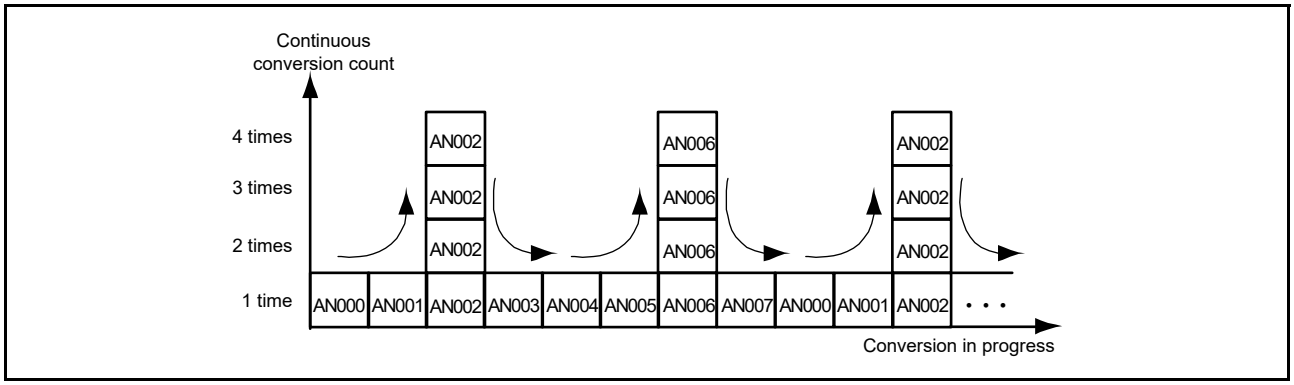


Figure 44.2 Scan Conversion Sequence with ADADC.ADC[2:0] = 011b, ADS002 = 1, and ADS006 = 1

Downloaded from Arrow.com.

44.2.10 A/D-Converted Value Addition/Average Count Select Register (ADADC)

Address(es): S12AD.ADADC 0008 900Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	AVEE	—	—	—	—	ADC[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	ADC[2:0]	Addition Count Select	b2 b0 0 0 0: 1-time conversion (no addition; same as normal conversion) 0 0 1: 2-time conversion (addition once) 0 1 0: 3-time conversion (addition twice)*1 0 1 1: 4-time conversion (addition three times) 1 0 1: 16-time conversion (addition 15 times)*1 Settings other than above are prohibited.	R/W
b6 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	AVEE	Average Mode Enable	0: Addition mode is selected. 1: Average mode is selected.	R/W

Note 1. The AVEE bit is enabled only when 2-time or 4-time conversion is selected. When average mode is selected (ADADC.AVEE bit = 1), do not set 3-time conversion (ADADC.ADC[2:0] = 010b) nor 16-time conversion (ADADC.ADC[2:0] = 101b).

The ADADC register sets the number of times addition is to proceed for the temperature sensor output, the internal reference voltage, and channels selected as being in A/D-converted value addition or average mode, and to select either addition or average mode for them.

ADC[2:0] Bits (Addition Count Select)

The ADC[2:0] bits set the number of times addition is to proceed as a common value for the temperature sensor output, the internal reference voltage, and channels for which A/D-converted value addition or average mode is selected, including those channels selected in double trigger mode (by the ADCSR.DBLANS[4:0] bits).
When average mode is selected by setting the ADADC.AVEE bit to 1, do not set the addition count to one time (ADADC.ADC[2:0] = 000b), three times (ADADC.ADC[2:0] = 010b), or 16 times (ADADC.ADC[2:0] = 101b).
The ADC[2:0] bits should be set while the ADCSR.ADST bit is 0.

AVEE Bit (Average Mode Enable)

The AVEE bit selects addition or average mode for the temperature sensor output, the internal reference voltage, and the channels selected for which the addition or average mode of A/D conversion is selected, including those channels selected in double trigger mode (by ADCSR.DBLANS[4:0] bits).
When average mode is selected by setting the ADADC.AVEE bit to 1, do not set the addition count to one time (ADADC.ADC[2:0] = 000b), three times (ADADC.ADC[2:0] = 010b), or 16 times (ADADC.ADC[2:0] = 101b). The mean value of 1-time, 3-time, and 16-time conversion cannot be obtained.
The AVEE bit should be set while the ADCSR.ADST bit is 0.

Downloaded from Arrow.com.

44.2.11 A/D Control Extended Register (ADCER)

Address(es): S12AD.ADCER 0008 900Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ADRFMT	—	—	—	DIAGM	DIAGLD	DIAGVAL[1:0]		—	—	ACE	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5	ACE	A/D Data Register Automatic Clearing Enable	0: Disables automatic clearing. 1: Enables automatic clearing.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b9, b8	DIAGVAL[1:0]	Self-Diagnosis Conversion Voltage Select	b9 b8 0 0: Setting prohibited in self-diagnosis voltage fixed mode 0 1: Uses the voltage of 0 V for self-diagnosis. 1 0: Uses the reference voltage x 1/2 for self-diagnosis.*1 1 1: Uses the reference voltage for self-diagnosis.*1	R/W
b10	DIAGLD	Self-Diagnosis Mode Select	0: Rotation mode for self-diagnosis voltage 1: Fixed mode for self-diagnosis voltage	R/W
b11	DIAGM	Self-Diagnosis Enable	0: Disables self-diagnosis of 12-bit A/D converter. 1: Enables self-diagnosis of 12-bit A/D converter.	R/W
b14 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	ADRFMT	A/D Data Register Format Select	0: Flush-right is selected for the A/D data register format. 1: Flush-left is selected for the A/D data register format.	R/W

Note 1. The reference voltage refers to the voltage on the pin selected in the ADHVREFCNT register.

The ADCER register sets self-diagnosis mode, format of A/D data registers y (ADDRy), and automatic clearing of A/D data registers.

ACE Bit (A/D Data Register Automatic Clearing Enable)

The ACE bit enables or disables automatic clearing (all “0”) of the ADDRy, ADRD, ADDBLDR, ADTSDR, or ADOCDR register after any of these registers have been read by the CPU, DTC, or DMACA. Automatic clearing of the A/D data register is enabled to detect a failure which has not been updated in the A/D data register.

DIAGVAL[1:0] Bits (Self-Diagnosis Conversion Voltage Select)

These bits select the voltage value used in self-diagnosis voltage fixed mode. For details, refer to the descriptions of the ADCER.DIAGLD bit.

Self-diagnosis should not be executed by setting the ADCER.DIAGLD bit to 1 when the ADCER.DIAGVAL[1:0] bits are set to 00b.

DIAGLD Bit (Self-Diagnosis Mode Select)

The DIAGLD bit selects whether the three voltage values are rotated or the fixed voltage is used in self-diagnosis. Setting this bit (ADCER.DIAGLD) to 0 allows conversion of the voltages in rotation mode where 0, the reference voltage × 1/2, and the reference voltage are converted in this order. When self-diagnosis rotation mode is selected after a reset, self-diagnosis is performed from 0 V. When self-diagnosis voltage fixed mode is selected, the fixed voltage specified by the ADCER.DIAGVAL[1:0] bits is converted. In self-diagnosis voltage rotation mode, the self-diagnosis voltage value does not return to 0 when scan conversion is completed. When scan conversion is restarted, therefore, rotation starts at the voltage value following the previous value. If fixed mode is switched to rotation mode, rotation starts at the fixed voltage value.

Downloaded from Arrow.com.

The DIAGLD bit should be set while the ADCSR.ADST bit is 0.

DIAGM Bit (Self-Diagnosis Enable)

The DIAGM bit enables or disables self-diagnosis.
Self-diagnosis is used to detect a failure of the 12-bit A/D converter. Specifically, one of the internally generated voltage values 0, the reference voltage × 1/2, and the reference voltage is converted. When conversion is completed, information on the converted voltage and the conversion result is stored into the self-diagnosis data register (ADRD). The ADRD register can then be read out by software to determine whether the conversion result falls within the normal range (normal) or not (abnormal). Self-diagnosis is executed once at the beginning of each scan, and one of the three voltages is converted. When self-diagnosis is selected in group scan mode, self-diagnosis is separately executed in groups A and B.
The DIAGM bit should be set while the ADCSR.ADST bit is 0.

ADRFMT Bit (A/D Data Register Format Select)

The ADRFMT bit specifies flush-right or flush-left for the data to be stored in the ADDRy, ADRD, ADTSDR, ADOCDR, ADDBLDR, ADCMPDR0, ADCMPDR1, ADWINLLB, or ADWINULB register.
The ADRFMT bit should be set while the ADCSR.ADST bit is 0.

For details on the format of each data register, see section 44.2.1, A/D Data Registers y (ADDRy) (y = 0 to 7, 16 to 20, 27), A/D Data Duplication Register (ADDBLDR), A/D Temperature Sensor Data Register (ADTSDR), A/D Internal Reference Voltage Data Register (ADOCDR), section 44.2.2, A/D Self-Diagnosis Data Register (ADRD), section 44.2.25, A/D Compare Function Window A Lower-Side Level Setting Register (ADCMPDR0), section 44.2.26, A/D Compare Function Window A Upper-Side Level Setting Register (ADCMPDR1), section 44.2.33, A/D Compare Function Window B Lower-Side Level Setting Register (ADWINLLB), and section 44.2.34, A/D Compare Function Window B Upper-Side Level Setting Register (ADWINULB).

Downloaded from Arrow.com.

44.2.12 A/D Conversion Start Trigger Select Register (ADSTRGR)

Address(es): S12AD.ADSTRGR 0008 9010h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	TRSA[5:0]						—	—	TRSB[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	TRSB[5:0]	A/D Conversion Start Trigger Select for Group B	Select the A/D conversion start trigger for group B in group scan mode.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b13 to b8	TRSA[5:0]	A/D Conversion Start Trigger Select	Select the A/D conversion start trigger in single scan mode and continuous mode. In group scan mode, the A/D conversion start trigger for group A is selected.	R/W
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADSTRGR register selects the A/D conversion start trigger.

TRSB[5:0] Bits (A/D Conversion Start Trigger Select for Group B)

The TRSB[5:0] bits select the trigger to start scanning of the analog input selected in group B. The TRSB[5:0] bits require to be set only in group scan mode and are not used in any other scan mode. For the scan conversion start trigger for group B, setting a software trigger or an asynchronous trigger is prohibited. Therefore, the TRSB[5:0] bits should be set to the value other than 000000b and the ADCSR.TRGE bit should be set to 1 in group scan mode.
When group A is given priority in group scan mode, setting the ADGSPCR.GBRP bit to 1 allows group B to continuously operate in single scan mode. When setting the ADGSPCR.GBRP bit to 1, set the TRSB[5:0] bits to 3Fh.
Note that the issuance period of trigger for A/D conversion must be more than or equal to the actual scan conversion time (t_{SCAN}). If the issuance period is less than t_{SCAN}, A/D conversion by the trigger may have no effect.
When the trigger from the module operated in PCLKA (MTU) is selected as an A/D conversion start trigger, a delay of the period for synchronization processing occurs. See section 44.3.6, Analog Input Sampling Time and Scan Conversion Time for details.
Table 44.6 lists the A/D conversion startup sources selected by the TRSB[5:0] bits.

TRSA[5:0] Bits (A/D Conversion Start Trigger Select)

The TRSA[5:0] bits select the trigger to start A/D conversion in single scan mode and continuous scan mode. In group scan mode, the trigger to start scanning of the analog input selected in group A is selected. When scanning is executed in group scan mode or double trigger mode, software trigger and asynchronous trigger cannot be used.

- When using the A/D conversion startup source of a synchronous trigger, set the ADCSR.TRGE bit to 1 and set the ADCSR.EXTRG bit to 0.
- When using the asynchronous trigger, set the ADCSR.TRGE bit to 1 and set the ADCSR.EXTRG bit to 1.
- Software trigger (ADCSR.ADST) is enabled regardless of the settings of the ADCSR.TRGE bit, the ADCSR.EXTRG bit, and the TRSA[5:0] bits.

Note that the issuance period of trigger for A/D conversion must be more than or equal to the actual scan conversion time (t_{SCAN}). If the issuance period is less than t_{SCAN}, A/D conversion by a trigger may have no effect. When the trigger from the module operated in PCLKA (MTU) is selected as an A/D conversion start trigger, a delay of the period for synchronization processing occurs. See section 44.3.6, Analog Input Sampling Time and Scan Conversion Time for details.

Downloaded from Arrow.com.

Table 44.7 lists the selection of A/D conversion start sources selected by the TRSA[5:0] bits.

Table 44.6 Selection of A/D Activation Sources by the TRSB[5:0] Bits

Module	Source	Remarks	TRSB [5]	TRSB [4]	TRSB [3]	TRSB [2]	TRSB [1]	TRSB [0]
Trigger source deselection state			1	1	1	1	1	1
MTU	TRG0AN	Compare match/input capture from MTU0.TGRA	0	0	0	0	0	1
	TRG0BN	Compare match/input capture from MTU0.TGRB	0	0	0	0	1	0
	TRGAN	Compare match/input capture from MTU0 to MTU4.TGRA or underflow (trough) of MTU4.TCNT in complementary PWM mode	0	0	0	0	1	1
	TRG0EN	Compare match from MTU0.TGRE	0	0	0	1	0	0
	TRG0FN	Compare match from MTU0.TGRF	0	0	0	1	0	1
	TRG4AN	Compare match between MTU4.TADCORA and MTU4.TCNT (interrupt skipping function)	0	0	0	1	1	0
	TRG4BN	Compare match between MTU4.TADCORB and MTU4.TCNT (interrupt skipping function)	0	0	0	1	1	1
	TRG4ABN	Compare match between MTU4.TADCORA and MTU4.TCNT or compare match between MTU4.TADCORB and MTU4.TCNT (interrupt skipping function)	0	0	1	0	0	0
TPU	TRGAN1	TGRA compare match/input capture from TPU0 to TPU4	0	0	1	1	0	1
	TRG4ABN1	TGRA compare match/input capture from TPU0	0	0	1	1	1	0
ELC	ELCTRG0		0	0	1	0	0	1

Table 44.7 Selection of A/D Activation Sources by the TRSA[5:0] Bits

Module	Source	Remarks	TRSA [5]	TRSA [4]	TRSA [3]	TRSA [2]	TRSA [1]	TRSA [0]
Trigger source deselection state			1	1	1	1	1	1
External pin	ADTRG0#	Input pin for the trigger	0	0	0	0	0	0
MTU	TRG0AN	Compare match/input capture from MTU0.TGRA	0	0	0	0	0	1
	TRG0BN	Compare match/input capture from MTU0.TGRB	0	0	0	0	1	0
	TRGAN	Compare match/input capture from MTU0 to MTU4.TGRA or underflow (trough) of MTU4.TCNT in complementary PWM mode	0	0	0	0	1	1
	TRG0EN	Compare match from MTU0.TGRE	0	0	0	1	0	0
	TRG0FN	Compare match from MTU0.TGRF	0	0	0	1	0	1
	TRG4AN	Compare match between MTU4.TADCORA and MTU4.TCNT (interrupt skipping function)	0	0	0	1	1	0
	TRG4BN	Compare match between MTU4.TADCORB and MTU4.TCNT (interrupt skipping function)	0	0	0	1	1	1
	TRG4ABN	Compare match between MTU4.TADCORA and MTU4.TCNT or compare match between MTU4.TADCORB and MTU4.TCNT (interrupt skipping function)	0	0	1	0	0	0
TPU	TRGAN1	TGRA compare match/input capture from TPU0 to TPU4	0	0	1	1	0	1
	TRG4ABN1	TGRA compare match/input capture from TPU0	0	0	1	1	1	0
ELC	ELCTRG0		0	0	1	0	0	1

Downloaded from Arrow.com.

44.2.13 A/D Conversion Extended Input Control Register (ADEXICR)

Address(es): S12AD.ADEXICR 0008 9012h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	OCSA	TSSA	—	—	—	—	—	—	OCSAD	TSSAD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TSSAD	Temperature Sensor Output A/D-Converted Value Addition/Average Mode Select	0: Temperature sensor output A/D-converted value addition/average mode is not selected. 1: Temperature sensor output A/D-converted value addition/average mode is selected.	R/W
b1	OCSAD	Internal Reference Voltage A/D-Converted Value Addition/Average Mode Select	0: Internal reference voltage A/D-converted value addition/average mode is not selected. 1: Internal reference voltage A/D-converted value addition/average mode is selected.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	TSSA	Temperature Sensor Output A/D Conversion Select	0: A/D conversion of temperature sensor output is not performed. 1: A/D conversion of temperature sensor output is performed.	R/W
b9	OCSA	Internal Reference Voltage A/D Conversion Select	0: A/D conversion of internal reference voltage is not performed. 1: A/D conversion of internal reference voltage is performed.	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADEXICR register specifies the settings of A/D conversion of the temperature sensor output and internal reference voltage.

TSSAD Bit (Temperature Sensor Output A/D-Converted Value Addition/Average Mode Select)

When the TSSAD bit is set to 1, A/D conversion of the temperature sensor output is selected and performed successively 2, 3, 4, or 16 times that is set with the ADADC.ADC[2:0] bits. When the ADADC.AVEE bit is 0, the value obtained by addition (integration) is stored in the A/D temperature sensor data register (ADTSDR). When the ADADC.AVEE bit is 1, the mean value is stored in the A/D temperature sensor data register (ADTSDR).
The TSSAD bit should be set while the ADCSR.ADST bit is 0.

OCSAD Bit (Internal Reference Voltage A/D-Converted Value Addition/Average Mode Select)

When the OCSAD bit is set to 1, A/D conversion of the internal reference voltage is selected and performed successively 2, 3, 4, or 16 times that is set with the ADADC.ADC[2:0] bits. When the ADADC.AVEE bit is 0, the value obtained by addition (integration) is stored in the A/D internal reference voltage data register (ADOCDR). When the ADADC.AVEE bit is 1, the mean value is stored in ADOCDR.
The OCSAD bit should be set while the ADCSR.ADST bit is 0.

TSSA Bit (Temperature Sensor Output A/D Conversion Select)

This bit selects A/D conversion of the temperature sensor output in single scan mode. When A/D conversion of the temperature sensor output is to be performed, all the bits in the ADANSA0, ADANSA1, ADANSB0, and ADANSB1 registers and the ADCSR.DBLE and OCSA bits should all be set to 0 in single scan mode.
The TSSA bit should be set while the ADCSR.ADST bit is 0. For A/D conversion of the temperature sensor output, the ADDISCR.ADNDIS[4:0] bits should be automatically set to 0Fh to discharge the A/D converter before sampling. The

Downloaded from Arrow.com.

sampling time should be 5 µs or longer.
Sampling starts after discharging is completed during A/D conversion of the temperature sensor output, an auto-discharging period of 15 ADCLK cycles is inserted before sampling.

OCSA Bit (Internal Reference Voltage A/D Conversion Select)

This bit selects A/D conversion of the internal reference voltage in single scan mode. When A/D conversion of the internal reference voltage is to be performed, set all the bits in the ADANSA0, ADANSA1, ADANSB0, and ADANSB1 registers and the ADCSR.DBLE bit and TSSA bit should be set to all 0 in single scan mode.
The OCSA bit should be set while the ADCSR.ADST bit is 0. For A/D conversion of the internal reference voltage, the ADDISCR.ADNDIS[4:0] bits should be automatically set to 0Fh to discharge the A/D converter before sampling. The sampling time should be 5 µs or longer.
Sampling starts after discharging is completed during A/D conversion of the internal reference voltage, so an auto-discharging period of 15 ADCLK cycles is inserted before sampling.

Downloaded from Arrow.com.

44.2.14 A/D Sampling State Register n (ADSSTRn) (n = 0 to 7, L, T, O)

Address(es): S12AD.ADSSTR0 0008 90E0h, S12AD.ADSSTR1 0008 90E1h, S12AD.ADSSTR2 0008 90E2h, S12AD.ADSSTR3 0008 90E3h, S12AD.ADSSTR4 0008 90E4h, S12AD.ADSSTR5 0008 90E5h, S12AD.ADSSTR6 0008 90E6h, S12AD.ADSSTR7 0008 90E7h, S12AD.ADSSTRL 0008 90DDh, S12AD.ADSSTRT 0008 90DEh, S12AD.ADSSTRO 0008 90DFh

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	1	1	0	1

The ADSSTRn register sets the sampling time for analog input.
If one state is one ADCLK (A/D conversion clock) cycle and the ADCLK clock is 54 MHz, one state is 18.5 ns. The initial value is 13 states. If the impedance of analog input signal source is too high to secure sufficient sampling time or if the ADCLK clock is slow, the sampling time can be adjusted. The ADSSTRn register should be set while the ADCSR.ADST bit is 0. The lower-limit value for sampling time differs depending on the PCLKB to ADCLK frequency ratio.
Set a value that is 5 states or more when PCLKB to ADCLK frequency ratio = 1:1, 2:1, 4:1, or 8:1.
Set a value that is 6 states or more when PCLKB to ADCLK frequency ratio = 1:2.

Table 44.8 shows the relationship between the A/D sampling state register and the relevant channels.
For details, refer to section 44.3.6, Analog Input Sampling Time and Scan Conversion Time.

Table 44.8 Relationship between A/D Sampling State Register and Relevant Channels

Register Name	Channels
ADSSTR0	AN000
ADSSTR1	AN001
ADSSTR2	AN002
ADSSTR3	AN003
ADSSTR4	AN004
ADSSTR5	AN005
ADSSTR6	AN006
ADSSTR7	AN007
ADSSTRL	AN016 to AN020, AN027
ADSSTRT	Temperature sensor output*1
ADSSTRO	Internal reference voltage*1

Note 1. When performing A/D conversion of the temperature sensor output or internal reference voltage, the sampling time should be 5 µs or longer. Since the maximum number of states that can be set by this register is 255, take note of the ADCLK frequency. For example, when ADCLK = 54 MHz, the sampling time does not reach 5 µs even if 255 states is set.

Downloaded from Arrow.com.

44.2.15 A/D Disconnection Detection Control Register (ADDISCR)

Address(es): S12AD.ADDISCR 0008 907Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	ADNDIS[4:0]				
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	ADNDIS[4:0]	A/D Disconnection Detection Assist Setting	b4 ADNDIS[4]: Discharge/precharge selected 0: Discharge 1: Precharge b3 to b0 ADNDIS[3:0]: Discharge/precharge period	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADDISCR register sets the disconnection detection assist function.

ADNDIS[4:0] Bits (A/D Disconnection Detection Assist Setting)

These bits select either precharge or discharge and the period of precharge/discharge for the A/D disconnection detection assist function. Setting the ADNDIS[4] bit = 1 allows to select precharge and setting the ADNDIS[4] bit = 0 allows to select discharge. The period of precharge/discharge can be set with the ADNDIS[3:0] bits. When the ADNDIS[3:0] bits = 0000b, the disconnection detection assist function is not effective. Setting of the ADNDIS[3:0] bits to 0001b is prohibited. Except for the case of ADNDIS[3:0] = 0000b or 0001b, the specified value indicates the number of states for the period of precharge/discharge.

When the ADEXICR.OCSA or TSSA bit is set to 1 to perform A/D conversion of the temperature sensor output or internal reference voltage, ADNDIS[4:0] are automatically fixed to 0Fh, and discharging is executed prior to A/D conversion (auto-discharging). An auto-discharge period of 15 ADCLK cycles is inserted before sampling each time the temperature sensor output or internal reference voltage is A/D-converted.

Downloaded from Arrow.com.

44.2.16 A/D Event Link Control Register (ADELCCR)

Address(es): S12AD.ADELCCR 0008 907Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	ELCC[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	ELCC[1:0]	Event Link Control	b1 b0 0 0: Event is generated on completion of the scan other than group B in group scan mode 0 1: Event is generated on completion of the scan of group B in group scan mode 1 x: Event is generated on completion of all scans	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

x: Don't care

The ADELCCR register sets the generation conditions of the ELC scan end event (S12ADELC).

ELCC[1:0] Bits (Event Link Control)

These bits select the generation conditions of the scan end event (S12ADELC) for the ELC.

Downloaded from Arrow.com.

44.2.17 A/D Group Scan Priority Control Register (ADGSPCR)

Address(es): S12AD.ADGSPCR 0008 9080h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	GBRP	—	—	—	—	—	—	—	—	—	—	—	—	—	GBRSCN	PGS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PGS	Group-A Priority Control Setting *1	0: Operation is without group-A priority control 1: Operation is with group-A priority control	R/W
b1	GBRSCN	Group B Restart Setting *2	(Enabled only when PGS = 1. Reserved when PGS = 0.) 0: Scanning for group B is not restarted after having been discontinued due to group-A priority control. 1: Scanning for group B is restarted after having been discontinued due to group-A priority control.	R/W
b14 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	GBRP	Group B Single Scan Continuous Start *3	(Enabled only when PGS = 1. Reserved when PGS = 0.) 0: Single scan for group B is not continuously activated. 1: Single scan for group B is continuously activated.	R/W

Note 1. When the PGS bit is to be set to 1, the ADCSR.ADCS[1:0] bits must be set to 01b (group scan mode). If the bits are set to any other values, proper operation is not guaranteed.
Note 2. When the GBRSCN bit is to be set to 1, the frequency ratio of peripheral module clock PCLKB to A/D conversion clock ADCLK should be set to 1:1.
Note 3. When the GBRP bit has been set to 1, single scan is performed continuously for group B regardless of the setting of the GBRSCN bit.

The ADGSPCR register is used to make settings for priority control of A/D conversion for group A in group scan mode.

PGS Bit (Group-A Priority Control Setting)

This bit sets the priority of operation on group A. Set this bit to 1 when giving priority to operation on group A.
When the PGS bit is to be set to 1, the ADCSR.ADCS[1:0] bits must be set to 01b (group scan mode).
When setting the PGS bit to 0, clearing should be performed by software according to section 44.8.2, Notes on Stopping A/D Conversion. When setting the PGS bit to 1, follow the procedure described in section 44.3.4.3, Operation under Group-A Priority Control.

GBRSCN Bit (Group B Restart Setting)

This bit controls the restarting of scan operation on group B when operation on group A is given priority.
If a scan operation on group B has been stopped by a group A trigger input with the GBRSCN bit set to 1, the scan operation is restarted on completion of the A/D conversion on group A. Also, if a group B trigger is input during A/D conversion on group A, the scan operation on group B is restarted on completion of the A/D conversion on group A.
If the GBRSCN bit has been set to 0, triggers that are input during A/D conversion are ignored. Also, the ADCSR.ADST bit must be 0 when the GBRSCN bit is to be set.
The setting of the GBRSCN bit is enabled when the PGS bit is set to 1.

GBRP Bit (Group B Single Scan Continuous Start)

This bit is set when a single scan operation is to be performed continuously on group B.
Setting the GBRP bit to 1 starts a single scan on group B. On completion of the scan, another single scan on group B is automatically started. If an A/D conversion on group B has been stopped due to an operation on group A that takes priority, single scan on group B is automatically restarted on completion of the A/D conversion on group A.
Disable group B trigger input before setting the GBRP bit to 1. Setting the GBRP bit to 1 invalidates the setting of the

Downloaded from Arrow.com.

GBRSCN bit. The ADCSR.ADST bit must be 0 when the GBRP bit is to be set.
The setting of the GBRP bit is enabled when the PGS bit is 1.

44.2.18 A/D Compare Function Control Register (ADCMPCR)

Address(es): S12AD.ADCMPCR 0008 9090h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	WCMPE	—	—	CMPAE	—	CMPBE	—	—	—	—	—	—	—	CMPAB[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	CMPAB[1:0]	Window A/B Composite Condition Setting	b1 b0 0 0: S12ADWMELC is output when window A comparison conditions are met OR window B comparison conditions are met. S12ADWUMELC is output in other cases. 0 1: S12ADWMELC is output when window A comparison conditions are met EXOR window B comparison conditions are met. S12ADWUMELC is output in other cases. 1 0: S12ADWMELC is output when window A comparison conditions are met AND window B comparison conditions are met. S12ADWUMELC is output in other cases. 1 1: Setting prohibited.	R/W
b8 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b9	CMPBE	Compare Window B Operation Enable	0: Compare window B operation is disabled. S12ADWMELC and S12ADWUMELC outputs are disabled. 1: Compare window B operation is enabled.	R/W
b10	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b11	CMPAE	Compare Window A Operation Enable	0: Compare window A operation is disabled. S12ADWMELC and S12ADWUMELC outputs are disabled. 1: Compare window A operation is enabled.	R/W
b13, 12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b14	WCMPE	Window Function Setting	0: Window function is disabled. Window A and window B operate as a comparator to compare the single value on the lower side with the A/D conversion result. 1: Window function is enabled. Window A and window B operate as a comparator to compare the two values on the upper and lower sides with the A/D conversion result.	R/W
b15	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

The ADCMPCR register sets the compare window A and window B functions.

CMPAB[1:0] Bits (Window A/B Composite Condition Setting)

These bits are valid when both window A and window B are enabled (CMPAE = 1 and CMPBE = 1) in single scan mode. These bits are used to select compare function match/mismatch event output conditions for the ELC or monitoring conditions of the ADWINMON.MONCOMB flag. The CMPAB[1:0] bits should be set while the ADCSR.ADST bit is 0.

CMPBE Bit (Compare Window B Operation Enable)

This bit enables or disables the compare window B operation. The CMPBE bit should be set while the ADCSR.ADST bit is 0.
Set this bit to 0 before setting the following registers.

Downloaded from Arrow.com.

- A/D channel select registers A0/A1/B0/B1 (ADANSA0, ADANSA1, ADANSB0, ADANSB1)
- The OCSA or TSSA bit in the A/D conversion extended input control register (ADEXICR.OCSA, TSSA)
- The CMPCHB[5:0] bits in the window B channel select register (ADCMPBNSR.CMPCHB[5:0])

CMPAE Bit (Compare Window A Operation Enable)

This bit enables or disables the compare window A operation. The CMPAE bit should be set while the ADCSR.ADST bit is 0.

Set this bit to 0 before setting the following registers.

- A/D channel select registers A0/A1/B0/B1 (ADANSA0, ADANSA1, ADANSB0, ADANSB1)
- The OCSA or TSSA bit in the A/D conversion extended input control register (ADEXICR.OCSA, TSSA)
- Window A channel select registers 0/1 (ADCMPANSR0, ADCMPANSR1)
- Window A extended input select register (ADCMPANSER)

WCMPE Bit (Window Function Setting)

This bit enables or disables the window function. The WCMPE bit should be set while the ADCSR.ADST bit is 0.

Downloaded from Arrow.com.

44.2.19 A/D Compare Function Window A Channel Select Register 0 (ADCMPANSR0)

Address(es): S12AD.ADCMPANSR0 0008 9094h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	CMPCHA007	CMPCHA006	CMPCHA005	CMPCHA004	CMPCHA003	CMPCHA002	CMPCHA001	CMPCHA000
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CMPCHA000	Compare Window A Channel Select	0: The corresponding channel from among AN000 to AN007 is not a target for compare window A. 1: The corresponding channel from among AN000 to AN007 is a target for compare window A.	R/W
b1	CMPCHA001			R/W
b2	CMPCHA002			R/W
b3	CMPCHA003			R/W
b4	CMPCHA004			R/W
b5	CMPCHA005			R/W
b6	CMPCHA006			R/W
b7	CMPCHA007			R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADCMPANSR0 register is used to select analog input channels for comparison under compare window A conditions from among AN000 to AN007.

CMPCHA0n Bit (n = 00 to 07) (Compare Window A Channel Select)

Setting the CMPCHA0n bit which has the same number as the A/D channel selected by the ADANSA0.ANSA0n or ADANSB0.ANSB0n bit to 1 enables the compare function.

The CMPCHA0n bit should be set while ADCSR.ADST bit is 0.

Downloaded from Arrow.com.

44.2.20 A/D Compare Function Window A Channel Select Register 1 (ADCMPANSR1)

Address(es): S12AD.ADCMPANSR1 0008 9096h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	CMPCHA111	—	—	—	—	—	—	CMPCHA104	CMPCHA103	CMPCHA102	CMPCHA101	CMPCHA100
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CMPCHA100	Compare Window A Channel Select	0: The corresponding channel from among AN016 to AN020 is not a target for compare window A. 1: The corresponding channel from among AN016 to AN020 is a target for compare window A.	R/W
b1	CMPCHA101			R/W
b2	CMPCHA102			R/W
b3	CMPCHA103			R/W
b4	CMPCHA104			R/W
b10 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11	CMPCHA111	Compare Window A Channel Select	0: AN027 is not a target for compare window A. 1: AN027 is a target for compare window A.	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADCMPANSR1 register is used to select analog input channels for comparison under compare window A conditions from among AN016 to AN020 and AN027.

CMPCHA1n Bit (n = 00 to 04, 11) (Compare Window A Channel Select)

Setting the CMPCHA1n bit which has the same number as the A/D channel selected by the ADANSA1.ANSA1n or ADANSB1.ANSB1n bit to 1 enables the compare function.
The CMPCHA1n bit should be set while ADCSR.ADST bit is 0.

Downloaded from Arrow.com.

44.2.21 A/D Compare Function Window A Extended Input Select Register (ADCMPANSER)

Address(es): S12AD.ADCMPANSER 0008 9092h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	CMPOCA	CMPTSA
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CMPTSA	Temperature Sensor Output Compare Select	0: Temperature sensor output is not a target for compare window A. 1: Temperature sensor output is a target for compare window A.	R/W
b1	CMPOCA	Internal Reference Voltage Compare Select	0: Internal reference voltage is not a target for compare window A. 1: Internal reference voltage is a target for compare window A.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADCMPANSER register is used to select whether the temperature sensor output or internal reference voltage is compared under compare window A conditions.

CMPTSA Bit (Temperature Sensor Output Compare Select)

Setting the CMPTSA bit to 1 while the ADEXICR.TSSA bit is 1 enables the compare window A function. This bit should be set while the ADCSR.ADST bit is 0.

CMPOCA Bit (Internal Reference Voltage Compare Select)

Setting the CMPOCA bit to 1 while ADEXICR.OCSA bit is 1 enables the compare window A function. This bit should be set while the ADCSR.ADST bit is 0.

Downloaded from Arrow.com.

44.2.22 A/D Compare Function Window A Comparison Condition Setting Register 0 (ADCMPLR0)

Address(es): S12AD.ADCMPLR0 0008 9098h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	CMPLCHA007	CMPLCHA006	CMPLCHA005	CMPLCHA004	CMPLCHA003	CMPLCHA002	CMPLCHA001	CMPLCHA000
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CMPLCHA000	Compare Window A Comparison Condition Select	When the window function is disabled (ADCMPCR.WCMPE bit = 0): 0: ADCMPDR0 register value > A/D-converted value 1: ADCMPDR0 register value < A/D-converted value When the window function is enabled (ADCMPCR.WCMPE bit = 1): 0: A/D-converted value < ADCMPDR0 register value or A/D-converted value > ADCMPDR1 register value 1: ADCMPDR0 register value < A/D-converted value < ADCMPDR1 register value	R/W
b1	CMPLCHA001			R/W
b2	CMPLCHA002			R/W
b3	CMPLCHA003			R/W
b4	CMPLCHA004			R/W
b5	CMPLCHA005			R/W
b6	CMPLCHA006			R/W
b7	CMPLCHA007			R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADCMPLR0 register sets the condition for use in comparing the values of the ADCMPDR0 and ADCMPDR1 registers with results of A/D conversion. The ADCMPLR0 register should be set while ADCSR.ADST bit is 0.

CMPLCHA0n Bit (n = 00 to 07) (Compare Window A Comparison Condition Select)

This bit sets the condition for use in comparison with the selected channel from among AN000 to AN007 to which compare window A conditions are applied. A condition can be set for individual comparison of each analog input.
The CMPLCHA000 bit is used for AN000, the CMPLCHA007 bit is used for AN007.
When the result of comparison matches the set condition, the ADCMPSR0.CMPSTCHA0n flag is set to 1.
Figure 44.3 shows the comparison conditions.

Downloaded from Arrow.com.

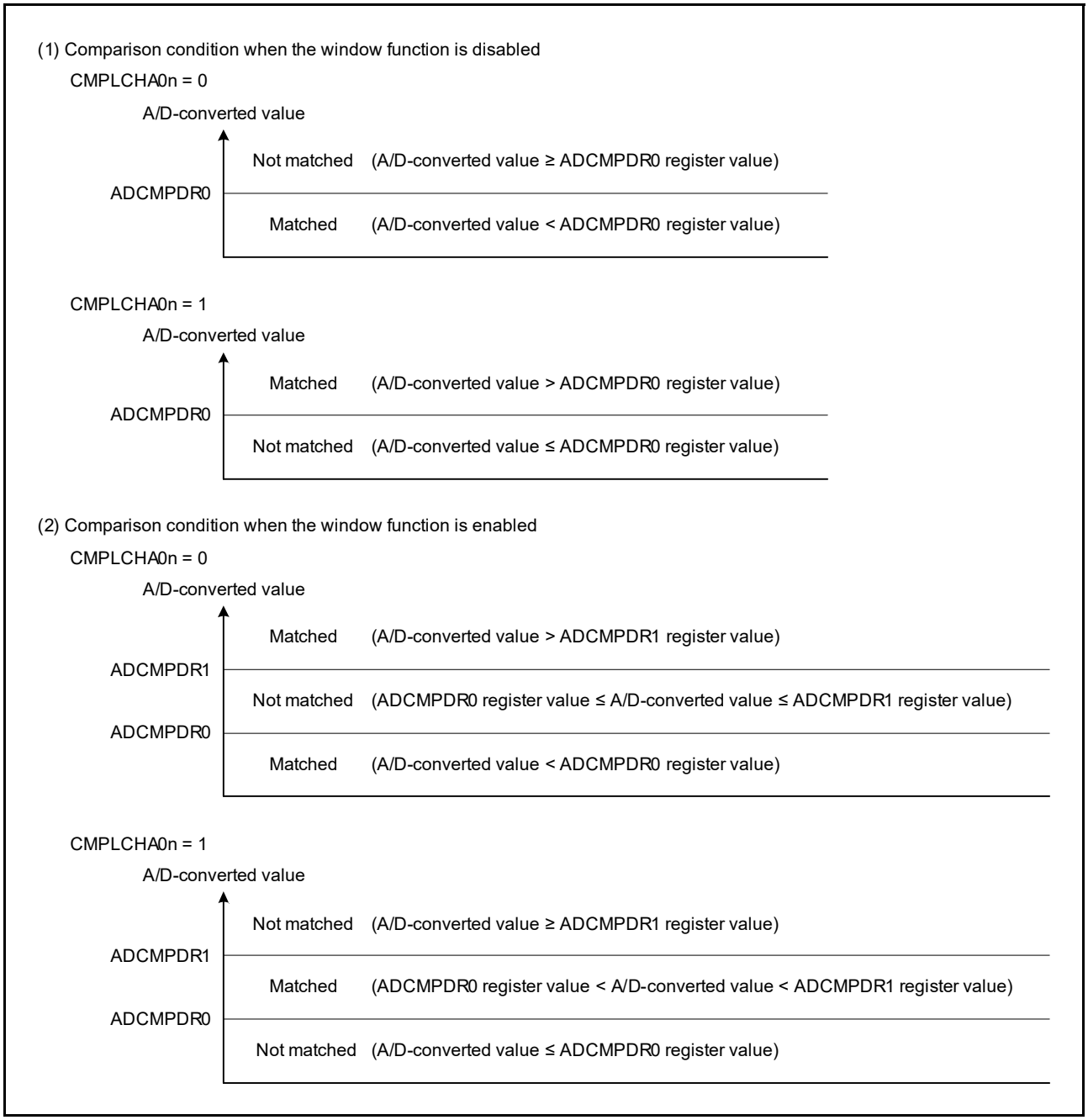


Figure 44.3 Explanation of Compare Function Window A Comparison Conditions

Downloaded from Arrow.com.

44.2.23 A/D Compare Function Window A Comparison Condition Setting Register 1 (ADCMPLR1)

Address(es): S12AD.ADCMPLR1 0008 909Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	CMPLCHA111	—	—	—	—	—	—	CMPLCHA104	CMPLCHA103	CMPLCHA102	CMPLCHA101	CMPLCHA100
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CMPLCHA100	Compare Window A Comparison Condition Select	When the window function is disabled (ADCMPCR.WCMPE bit = 0): 0: ADCMPDR0 register value > A/D-converted value 1: ADCMPDR0 register value < A/D-converted value When the window function is enabled (ADCMPCR.WCMPE bit = 1): 0: A/D-converted value < ADCMPDR0 register value or A/D-converted value > ADCMPDR1 register value 1: ADCMPDR0 register value < A/D-converted value < ADCMPDR1 register value	R/W
b1	CMPLCHA101			R/W
b2	CMPLCHA102			R/W
b3	CMPLCHA103			R/W
b4	CMPLCHA104			R/W
b10 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11	CMPLCHA111	Compare Window A Comparison Condition Select	When the window function is disabled (ADCMPCR.WCMPE bit = 0): 0: ADCMPDR0 register value > A/D-converted value 1: ADCMPDR0 register value < A/D-converted value When the window function is enabled (ADCMPCR.WCMPE bit = 1): 0: A/D-converted value < ADCMPDR0 register value or A/D-converted value > ADCMPDR1 register value 1: ADCMPDR0 register value < A/D-converted value < ADCMPDR1 register value	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADCMPLR1 register sets the condition for use in comparing the values of the ADCMPDR0 and ADCMPDR1 registers with results of A/D conversion.
The ADCMPLR1 register should be set while ADCSR.ADST bit is 0.

CMPLCHA1n Bit (n = 00 to 04, 11) (Compare Window A Comparison Condition Select)

This bit sets the condition for use in comparison with the selected channel from among AN016 to AN020 and AN027 to which compare window A conditions are applied. A condition can be set for individual comparison of each analog input. The CMPLCHA100 bit is used for AN016 and the CMPLCHA111 bit is used for AN027.
When the result of comparison matches the set condition, the ADCMPSR1.CMPSTCHA1n flag is set to 1.
Figure 44.3 shows the comparison conditions.

Downloaded from Arrow.com.

44.2.24 A/D Compare Function Window A Extended Input Comparison Condition Setting Register (ADCMPLER)

Address(es): S12AD.ADCMPLER 0008 9093h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	CMPLOCA	CMPLTSA
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CMPLTSA	Compare Window A Temperature Sensor Output Comparison Condition Select	When the window function is disabled (ADCMPCR.WCMPE bit = 0): 0: ADCMPDR0 register value > A/D-converted value 1: ADCMPDR0 register value < A/D-converted value When the window function is enabled (ADCMPCR.WCMPE bit = 1): 0: A/D-converted value < ADCMPDR0 register value or A/D-converted value > ADCMPDR1 register value 1: ADCMPDR0 register value < A/D-converted value < ADCMPDR1 register value	R/W
b1	CMPLOCA	Internal Reference Voltage Comparison Condition Select	When the window function is disabled (ADCMPCR.WCMPE bit = 0): 0: ADCMPDR0 register value > A/D-converted value 1: ADCMPDR0 register value < A/D-converted value When the window function is enabled (ADCMPCR.WCMPE bit = 1): 0: A/D-converted value < ADCMPDR0 register value or A/D-converted value > ADCMPDR1 register value 1: ADCMPDR0 register value < A/D-converted value < ADCMPDR1 register value	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADCMPLER register sets the condition for use in comparing the values of ADCMPDR0 and ADCMPDR1 registers with results of A/D conversion.
The ADCMPLER register should be set while ADCSR.ADST is 0.

CMPLTSA Bit (Compare Window A Temperature Sensor Output Comparison Condition Select)

This bit sets the condition for use in comparison with temperature sensor output to which compare window A conditions are applied.
When the result of comparison matches the set condition, the ADCMPSER.CMPSTTSA flag is set to 1.
Figure 44.3 shows the comparison conditions.

CMPLOCA Bit (Internal Reference Voltage Comparison Condition Select)

This bit sets conditions for use in comparison with internal reference voltage to which compare window A conditions are applied.
When the result of comparison matches the set condition, the ADCMPSER.CMPSTOCA flag is set to 1.
Figure 44.3 shows the comparison conditions.

Downloaded from Arrow.com.

44.2.25 A/D Compare Function Window A Lower-Side Level Setting Register (ADCMPDR0)

Address(es): S12AD.ADCMPDR0 0008 909Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The ADCMPDR0 register is a readable/writable register that sets the reference data when the compare window A function is used. The ADCMPDR0 register sets the lower-side level of window A.
The ADCMPDR0 register is writable even during A/D conversion. The reference data can be dynamically modified by rewriting register values during A/D conversion.
Set the registers so that the upper-side level is not less than the lower-side level (ADCMPDR1 setting value ≥ ADCMPDR0 setting value).

The ADCMPDR0 register uses different formats depending on the following conditions.

- Settings of the A/D data register format select bit (flush-right or flush-left)
- Settings of the A/D-converted value addition/average function select register (A/D-converted value average mode selected or not selected)
- Settings of the A/D-converted value addition/average count select register (addition/average mode selected, addition count selected)

Note: If a format different from the format setting of A/D data register y is used to set the compare value, a correct comparison result will not be obtained.

(1) When A/D-Converted Value Addition/Average Mode is Not Selected

- Flush-right format
  Set bits 11 to 0 to the lower-side comparison level. Write 0 to bits 15 to 12.
- Flush-left format
  Set bits 15 to 4 to the lower-side comparison level. Write 0 to bits 3 to 0.

(2) When A/D-Converted Value Average Mode is Selected

- Flush-right format
  Set bits 11 to 0 to the lower-side comparison level for comparison with the A/D-converted value of the same channel. Write 0 to bits 15 to 12.
- Flush-left format
  Set bits 15 to 4 to the lower-side comparison level for comparison with the A/D-converted value of the same channel. Write 0 to bits 3 to 0.

A/D-converted value average mode can be set only when two or four times is selected in A/D-converted value addition mode.

(3) When A/D-Converted Value Addition Mode is Selected

- Flush-right format (A/D-converted value addition mode and 1-time to 4-time conversion selected)
  Set bits 13 to 0 to the lower-side comparison level for comparison with the A/D-converted value of the same channel. Write 0 to bits 15 and 14.
- Flush-right format (A/D-converted value addition mode and 16-time conversion selected)
  Set bits 15 to 0 to the lower-side comparison level for comparison with the A/D-converted value of the same channel.
- Flush-left format (A/D-converted value addition mode and 1-time to 4-time conversion selected)

Downloaded from Arrow.com.

Set bits 15 to 2 to the lower-side comparison level for comparison with the A/D-converted value of the same channel. Write 0 to bits 1 and 0.

- Flush-left format (A/D-converted value addition mode and 16-time conversion selected)
  Set bits 15 to 0 to the lower-side comparison level for comparison with the A/D-converted value of the same channel.

When A/D-converted addition mode is selected, set the value added by the A/D-converted value of the same channel. The number of A/D conversions can be set to 1, 2, 3, 4, or 16 times. If A/D-converted addition mode is selected, when the A/D conversion count is set to 1 to 4 times, set the number of conversion accuracy bits extended by 2 bits in the ADCMPDR0 register; when the A/D conversion count is set to 16 times, set the number of conversion accuracy bits extended by 4 bits in the ADCMPDR0 register.
Even if A/D converted value addition mode is selected, set the reference data in the A/D data register according to the settings of the A/D data register format select bits.

Downloaded from Arrow.com.

44.2.26 A/D Compare Function Window A Upper-Side Level Setting Register (ADCMPDR1)

Address(es): S12AD.ADCMPDR1 0008 909Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The ADCMPDR1 register is a readable/writable register that sets the reference data when the compare window A function is used. The ADCMPDR1 register sets the upper-side level of window A.
The ADCMPDR1 register is writable even during A/D conversion. The reference data can be dynamically modified by rewriting register values during A/D conversion.
Set the registers so that the upper-side level is not less than the lower-side level (ADCMPDR1 setting value ≥ ADCMPDR0 setting value).
The ADCMPDR1 register is not used when the window function is disabled.

The ADCMPDR1 register uses different formats depending on the following conditions.

- Settings of the A/D data register format select bit (flush-right or flush-left)
- Settings of the A/D-converted value addition/average function select register (A/D-converted value average mode selected or not selected)
- Settings of the A/D-converted value addition/average count select register (addition/average mode selected, addition count selected)

Note: If a format different from the format setting of A/D data register y is used to set the compare value, a correct comparison result will not be obtained.

(1) When A/D-Converted Value Addition/Average Mode is Not Selected

- Flush-right format
  Set bits 11 to 0 to the upper-side comparison level. Write 0 to bits 15 to 12.
- Flush-left format
  Set bits 15 to 4 to the upper-side comparison level. Write 0 to bits 3 to 0.

(2) When A/D-Converted Value Average Mode is Selected

- Flush-right format
  Set bits 11 to 0 to the upper-side comparison level for comparison with the A/D-converted value of the same channel. Write 0 to bits 15 to 12.
- Flush-left format
  Set bits 15 to 4 to the upper-side comparison level for comparison with the A/D-converted value of the same channel. Write 0 to bits 3 to 0.

A/D-converted value average mode can be set only when two or four times is selected in A/D-converted value addition mode.

(3) When A/D-Converted Value Addition Mode is Selected

- Flush-right format (A/D-converted value addition mode and 1-time to 4-time conversion selected)
  Set bits 13 to 0 to the upper-side comparison level for comparison with the A/D-converted value of the same channel. Write 0 to bits 15 and 14.
- Flush-right format (A/D-converted value addition mode and 16-time conversion selected)
  Set bits 15 to 0 to the upper-side comparison level for comparison with the A/D-converted value of the same channel.

Downloaded from Arrow.com.

- Flush-left format (A/D-converted value addition mode and 1-time to 4-time conversion selected)
- Set bits 15 to 2 to the upper-side comparison level for comparison with the A/D-converted value of the same channel. Write 0 to bits 1 and 0.
- Flush-left format (A/D-converted value addition mode and 16-time conversion selected)
  Set bits 15 to 0 to the upper-side comparison level for comparison with the A/D-converted value of the same channel.

When A/D-converted addition mode is selected, set the value added by the A/D-converted value of the same channel. The number of A/D conversions can be set to 1, 2, 3, 4, or 16 times. If A/D-converted addition mode is selected, when the A/D conversion count is set to 1 to 4 times, set the number of conversion accuracy bits extended by 2 bits in the ADCMPDR1 register; when the A/D conversion count is set to 16 times, set the number of conversion accuracy bits extended by 4 bits in the ADCMPDR1 register.
Even if A/D converted value addition mode is selected, set the reference data in the A/D data register according to the settings of the A/D data register format select bits.

Downloaded from Arrow.com.

44.2.27 A/D Compare Function Window A Channel Status Register 0 (ADCMPSR0)

Address(es): S12AD.ADCMPSR0 0008 90A0h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	CMPST CHA007	CMPST CHA006	CMPST CHA005	CMPST CHA004	CMPST CHA003	CMPST CHA002	CMPST CHA001	CMPST CHA000
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CMPSTCHA000	Compare Window A Flag	When window A operation is enabled (ADCMPCR.CMPAE = 1), these flags indicate the comparison result of channels (AN000 to AN007 to which window A comparison conditions are applied. 0: Comparison conditions are not met. 1: Comparison conditions are met.	R/W
b1	CMPSTCHA001			R/W
b2	CMPSTCHA002			R/W
b3	CMPSTCHA003			R/W
b4	CMPSTCHA004			R/W
b5	CMPSTCHA005			R/W
b6	CMPSTCHA006			R/W
b7	CMPSTCHA007			R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADCMPSR0 register stores the comparison results of the compare window A function.

CMPSTCHA0n Flag (n = 00 to 07) (Compare Window A Flag)

This flag is comparison result status flag of channel (AN000 to AN007) to which window A comparison conditions are applied. When the comparison condition set by the ADCMPLR0.CMPLCHAn bit is met at the end of A/D conversion, each of these flags is set to 1. The CMPSTCHA000 and CMPSTCHA007 flags correspond to AN000 and AN007, respectively.

Writing 1 to the CMPSTCHA0n flag is disabled.

[Setting condition]

- The condition set by the ADCMPLR0.CMPLCHA0n bit is met when ADCMPCR.CMPAE = 1

[Clearing condition]

- 0 is written after reading 1

Downloaded from Arrow.com.

44.2.28 A/D Compare Function Window A Channel Status Register 1 (ADCMPSR1)

Address(es): S12AD.ADCMPSR1 0008 90A2h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	CMPST CHA111	—	—	—	—	—	—	CMPST CHA104	CMPST CHA103	CMPST CHA102	CMPST CHA101	CMPST CHA100
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CMPSTCHA100	Compare Window A Flag	When window A operation is enabled (ADCMPCR.CMPAE = 1), these flags indicate the comparison result of channels (AN016 to AN020) to which window A comparison conditions are applied. 0: Comparison conditions are not met. 1: Comparison conditions are met.	R/W
b1	CMPSTCHA101			R/W
b2	CMPSTCHA102			R/W
b3	CMPSTCHA103			R/W
b4	CMPSTCHA104			R/W
b10 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11	CMPSTCHA111	Compare Window A Flag	When window A operation is enabled (ADCMPCR.CMPAE = 1), this flag indicates the comparison result of channel (AN027) to which window A comparison condition is applied. 0: Comparison conditions are not met. 1: Comparison conditions are met.	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADCMPSR1 register stores the comparison results of the compare window A function.

CMPSTCHA1n Flag (n = 00 to 04, 11) (Compare Window A Flag)

This flag is comparison result status flag of channel (AN016 to AN020, AN027) to which window A comparison conditions are applied. When the comparison condition set by the ADCMPLR1.CMPLCHA1n bit is met at the end of A/D conversion, each of these flags is set to 1. The CMPSTCHA100 flag corresponds to AN016, the CMPSTCHA104 flag corresponds to AN020, and the CMPSTCHA111 flag corresponds to AN027.

Writing 1 to the CMPSTCHA1n flag is disabled.

[Setting condition]

- The condition set by the ADCMPLR1.CMPLCHA1n bit is met when ADCMPCR.CMPAE = 1

[Clearing condition]

- 0 is written after reading 1

Downloaded from Arrow.com.

44.2.29 A/D Compare Function Window A Extended Input Channel Status Register (ADCMPSER)

Address(es): S12AD.ADCMPSER 0008 90A4h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	CMPSTOCA	CMPSTTSA
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CMPSTTSA	Compare Window A Temperature Sensor Output Compare Flag	When window A operation is enabled (ADCMPCR.CMPAE = 1), this flag indicates the temperature sensor output comparison result. 0: Comparison conditions are not met. 1: Comparison conditions are met.	R/W
b1	CMPSTOCA	Compare Window A Internal Reference Voltage Compare Flag	When window A operation is enabled (ADCMPCR.CMPAE = 1), this flag indicates the internal reference voltage comparison result. 0: Comparison conditions are not met. 1: Comparison conditions are met.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADCMPSER register stores the comparison result of the compare window A function.

CMPSTTSA Flag (Compare Window A Temperature Sensor Output Compare Flag)

This flag is a status flag that indicates the temperature sensor output comparison result. When the comparison condition set by the ADCMPLER.CMPLTSA bit is met at the end of A/D conversion, this flag is set to 1.
Writing 1 to the CMPSTTSA flag is disabled.
[Setting condition]

- The condition set by the ADCMPLER.CMPLTSA bit is met when ADCMPCR.CMPAE = 1

[Clearing condition]

- 0 is written after reading 1

CMPSTOCA Flag (Compare Window A Internal Reference Voltage Compare Flag)

This flag is a status flag that indicates the internal reference voltage comparison result. When the comparison condition set by the ADCMPLER.CMPLOCA bit is met at the end of A/D conversion, this flag is set to 1.
Writing 1 to the CMPSTOCA flag is disabled.
The value 1 cannot be written to the CMPSTOCA flag.
[Setting condition]

- The condition set in by the ADCMPLER.CMPLOCA bit is met when ADCMPCR.CMPAE = 1

[Clearing condition]

- 0 is written after reading 1

Downloaded from Arrow.com.

44.2.30 A/D High-Potential/Low-Potential Reference Voltage Control Register (ADHVREFCNT)

Address(es): S12AD.ADHVREFCNT 0008 908Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	ADSLP	—	—	LVSEL	—	—	HVSEL[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	HVSEL[1:0]	High-Potential Reference Voltage Select	b1 b0 0 0: AVCC0 is selected as the high-potential reference voltage. 0 1: VREFH0 is selected as the high-potential reference voltage. Settings other than above are prohibited.	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	LVSEL	Low-Potential Reference Voltage Select	0: AVSS0 is selected as the low-potential reference voltage. 1: VREFL0 is selected as the low-potential reference voltage.	R/W
b6, b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	ADSLP	Sleep	0: Normal operation 1: Standby state	R/W

The ADHVREFCNT register specifies the high-potential and low-potential reference voltages. Set this register before performing A/D conversion.

HVSEL[1:0] Bits (High-Potential Reference Voltage Select)

These bits are used to set the high-potential reference voltage. AVCC0 or VREFH0 is selectable as the high-potential reference voltage.

LVSEL Bit (Low-Potential Reference Voltage Select)

This bit is used to set the low-potential reference voltage. AVSS0 or VREFL0 is selectable as the low-potential reference voltage.

ADSLP Bit (Sleep)

This bit is used to transition the 12-bit A/D converter to the standby state. Set the ADSLP bit to 1 only when modifying the ADCSR.ADHSC bit. In other cases, setting the ADSLP bit to 1 is prohibited.

After the ADSLP bit is set to 1, wait at least 5 µs before clearing this bit to 0. Furthermore, after the ADSLP bit is cleared to 0, wait at least 1 µs and then start the A/D conversion.

For the ADHSC bit rewriting procedure, see section 44.8.10, ADHSC Bit Rewriting Procedure.

Downloaded from Arrow.com.

44.2.31 A/D Compare Function Window A/B Status Monitor Register (ADWINMON)

Address(es): S12AD.ADWINMON 0008 908Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	MONCMPB	MONCMPA	—	—	—	MONCOMB
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MONCOMB	Combination Result Monitor Flag	This flag indicates the combination result. This flag is valid when both window A operation and window B operation are enabled. 0: Window A/window B composite conditions are not met. 1: Window A/window B composite conditions are met.	R
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R
b4	MONCMPA	Comparison Result Monitor A Flag	0: Window A comparison conditions are not met. 1: Window A comparison conditions are met.	R
b5	MONCMPB	Comparison Result Monitor B Flag	0: Window B comparison conditions are not met. 1: Window B comparison conditions are met.	R
b7, b6	—	Reserved	These bits are read as 0.	R

The ADWINMON register can monitor the comparison result and the combination result.

MONCOMB Flag (Combination Result Monitor Flag)

This read-only flag indicates the result in combination of comparison condition result A and comparison result condition B with the combination condition set by the ADCMPCR.CMPAB[1:0] bits.

[Setting condition]

- The combined result meets the combination condition set by the ADCMPCR.CMPAB[1:0] bits when ADCMPCR.CMPAE = 1 and ADCMPCR.CMPBE = 1.

[Clearing conditions]

- The combined result does not meet the combination condition set by the ADCMPCR.CMPAB[1:0] bits.
- ADCMPCR.CMPAE = 0 or ADCMPCR.CMPBE = 0.

MONCMPA Flag (Comparison Result Monitor A Flag)

This read-only flag is read as 1 when the A/D-converted value of the window A target channel meets the condition set in the ADCMPLR0, ADCMPLR1, and ADCMPLER registers, and is read as 0 otherwise.

[Setting condition]

- The A/D-converted value meets the condition set by the ADCMPLR0.CMPLCHA0n bit when ADCMPCR.CMPAE = 1.

[Clearing conditions]

- The A/D-converted value does not meet the condition set by the ADCMPLR0.CMPLCHA0n bit when ADCMPCR.CMPAE = 1.
- ADCMPCR.CMPAE = 0 (Automatically cleared when the ADCMPCR.CMPAE bit value changes from 1 to 0.)

MONCMPB Flag (Comparison Result Monitor B Flag)

This read-only flag is read as 1 when the A/D converted value of the window B target channel meets the condition set by the ADCMPBNSR.CMPLB bit, and is read as 0 in other cases.

[Setting condition]

- The A/D-converted value meets the condition set by ADCMPBNSR.CMPLB bit when ADCMPCR.CMPBE = 1.

Downloaded from Arrow.com.

[Clearing conditions]

- The A/D-converted value does not meet the condition set by ADCMPBNSR.CMPLB bit when ADCMPCR.CMPBE = 1.
- ADCMPCR.CMPBE = 0 (Automatically cleared when the ADCMPCR.CMPBE bit value changes from 1 to 0.)

44.2.32 A/D Compare Function Window B Channel Select Register (ADCMPBNSR)

Address(es): S12AD.ADCMPBNSR 0008 90A6h

	b7	b6	b5	b4	b3	b2	b1	b0
	CMPLB	—	CMPCHB[5:0]					
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	CMPCHB[5:0]	Compare Window B Channel Select	These bits select channels to be compared with the compare window B conditions. b5 b0 0 0 0 0 0 0: AN000 0 0 0 0 0 1: AN001 0 0 0 0 1 0: AN002 : : 0 0 0 1 1 0: AN006 0 0 0 1 1 1: AN007 0 1 0 0 0 0: AN016 0 1 0 0 0 1: AN017 : : 0 1 0 1 0 0: AN020 0 1 1 0 1 1: AN027 1 0 0 0 0 0: Temperature sensor 1 0 0 0 0 1: Internal reference voltage Settings other than above are prohibited.	R/W
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7	CMPLB	Compare Window B Comparison Condition Setting	When the window function is disabled (ADCMPCR.WCMPE bit = 0) 0: ADWINLLB register value > A/D-converted value 1: ADWINLLB register value < A/D-converted value When the window function is enabled (ADCMPCR.WCMPE bit = 1) 0: A/D-converted value < ADWINLLB register value or ADWINULB register value < A/D-converted value 1: ADWINLLB register value < A/D-converted value < ADWINULB register value	R/W

The ADCMPBNSR register is used to set the compare window B function.

CMPCHB[5:0] Bits (Compare Window B Channel Select)

These bits are used to select channels to be compared from among AN000 to AN007, AN016 to AN020, AN027, temperature sensor, and internal reference voltage against the conditions of comparison window B.
The compare window B function is enabled by specifying the hexadecimal number of the A/D conversion channel selected by the ADANSA0, ADANSA1, ADANSB0, and ADANSB1 registers.
The CMPCHB[5:0] bits should be set while the ADCSR.ADST bit is 0.

CMPLB Bit (Compare Window B Comparison Condition Setting)

This bit is used to set comparison conditions of channels for window B. When the comparison result of each analog input meets the set condition, the ADCMPBSR.CMPSTB flag is set to 1.

Downloaded from Arrow.com.

Figure 44.4 shows the comparison conditions.

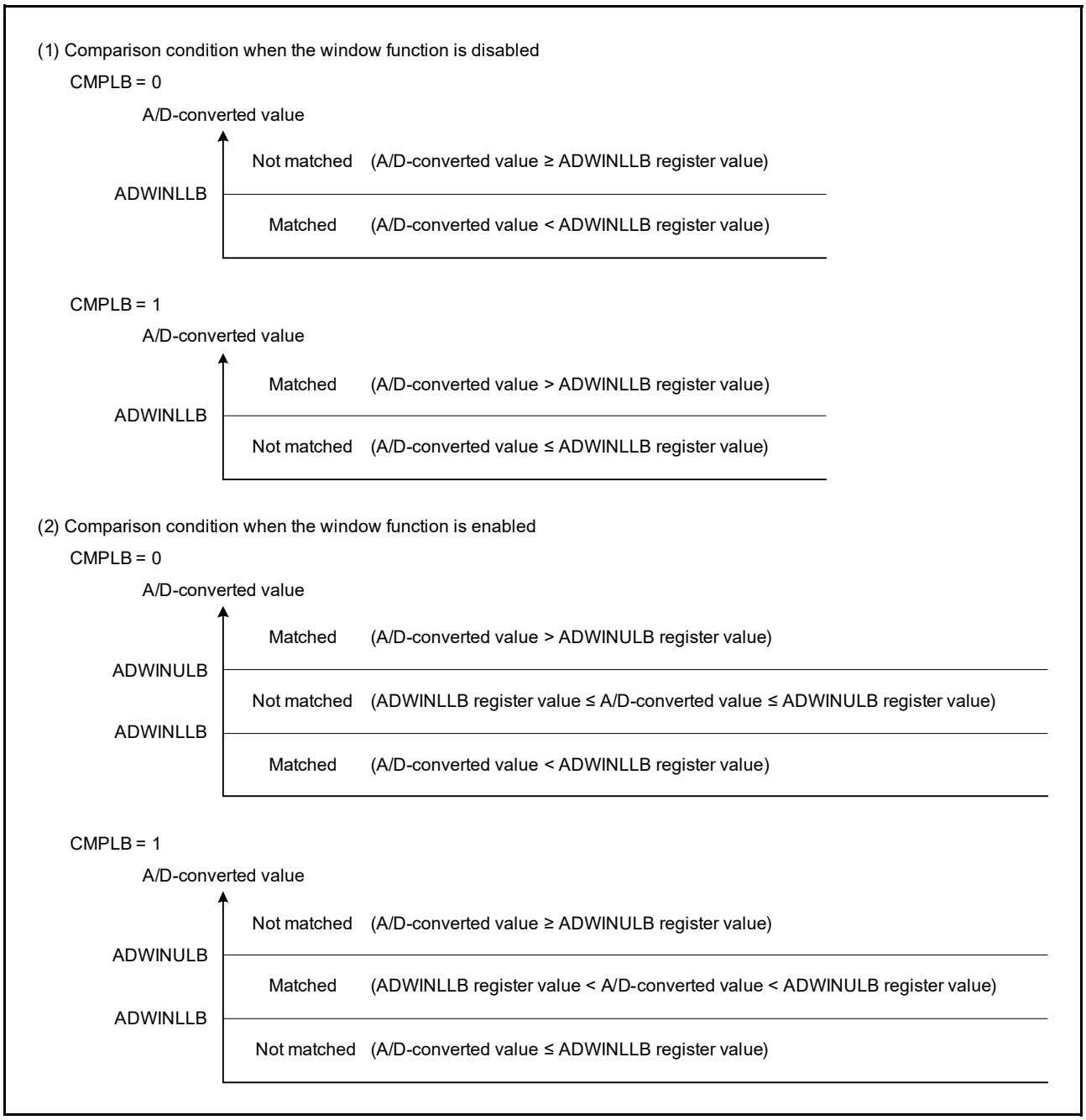


Figure 44.4 Explanation of Compare Function Window B Compare Conditions

Downloaded from Arrow.com.

44.2.33 A/D Compare Function Window B Lower-Side Level Setting Register (ADWINLLB)

Address(es): S12AD.ADWINLLB 0008 90A8h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The ADWINLLB register is a readable/writable register that sets the reference data when the compare window B function is used. The ADWINLLB register sets the lower-side level of window B.
The ADWINLLB register is writable even during A/D conversion. The reference data can be dynamically modified by rewriting register values during A/D conversion.
Set the registers so that the upper-side level is not less than the lower-side level (ADWINULB setting value ≥ ADWINLLB setting value).

The ADWINLLB register uses different formats depending on the following conditions.

- Settings of the A/D data register format select bit (flush-right or flush-left)
- Settings of the A/D-converted value addition/average function select register (A/D-converted value average mode selected or not selected)
- Settings of the A/D-converted value addition/average count select register (addition/average mode selected, addition count selected)

Note: If a format different from the format setting of A/D data register y (ADDRy) is used to set the compare value, a correct comparison result will not be obtained.

(1) When A/D-Converted Value Addition/Average Mode is Not Selected

- Flush-right format
  Set bits 11 to 0 to the lower-side comparison level. Write 0 to bits 15 to 12.
- Flush-left format
  Set bits 15 to 4 to the lower-side comparison level. Write 0 to bits 3 to 0.

(2) When A/D-Converted Value Average Mode is Selected

- Flush-right format
  Set bits 11 to 0 to the lower-side comparison level for comparison with the A/D-converted value of the same channel. Write 0 to bits 15 to 12.
- Flush-left format
  Set bits 15 to 4 to the lower-side comparison level for comparison with the A/D-converted value of the same channel. Write 0 to bits 3 to 0.

A/D-converted value average mode can be set only when two or four times is selected in A/D-converted value addition mode.

(3) When A/D-Converted Value Addition Mode is Selected

- Flush-right format (A/D-converted value addition mode and 1-time to 4-time conversion selected)
  Set bits 13 to 0 to the lower-side comparison level for comparison with the A/D-converted value of the same channel. Write 0 to bits 15 and 14.
- Flush-right format (A/D-converted value addition mode and 16-time conversion selected)
  Set bits 15 to 0 to the lower-side comparison level for comparison with the A/D-converted value of the same channel.
- Flush-left format (A/D-converted value addition mode and 1-time to 4-time conversion selected)

Downloaded from Arrow.com.

Set bits 15 to 2 to the lower-side comparison level for comparison with the A/D-converted value of the same channel. Write 0 to bits 1 and 0.

- Flush-left format (A/D-converted value addition mode and 16-time conversion selected)
  Set bits 15 to 0 to the lower-side comparison level for comparison with the A/D-converted value of the same channel.

When A/D-converted addition mode is selected, set the value added by the A/D-converted value of the same channel. The number of A/D conversions can be set to 1, 2, 3, 4, or 16 times. If A/D-converted addition mode is selected, when the A/D conversion count is set to 1 to 4 times, set the number of conversion accuracy bits extended by 2 bits in the ADWINLLB register; when the A/D conversion count is set to 16 times, set the number of conversion accuracy bits extended by 4 bits in the ADWINLLB register.
Even if A/D converted value addition mode is selected, set the reference data in the A/D data register according to the settings of the A/D data register format select bits.

Downloaded from Arrow.com.

44.2.34 A/D Compare Function Window B Upper-Side Level Setting Register (ADWINULB)

Address(es): S12AD.ADWINULB 0008 90AAh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The ADWINULB register is a readable/writable register that sets the reference data when the compare window B function is used. The ADWINULB register sets the upper-side level of window B.
The ADWINULB register is writable even during A/D conversion. The reference data can be dynamically modified by rewriting register values during A/D conversion.
Set the registers so that the upper-side level is not less than the lower-side level (ADWINULB setting value ≥ ADWINLLB setting value)
The ADWINULB register is not used when the window function is disabled.

The ADWINULB register uses different formats depending on the following conditions.

- Settings of the A/D data register format select bit (flush-right or flush-left)
- Settings of the A/D-converted value addition/average function select register (A/D-converted value average mode selected or not selected)
- Settings of the A/D-converted value addition/average count select register (addition/average mode selected, addition count selected)

Note: If a format different from the format setting of A/D data register y (ADDRy) is used to set the compare value, a correct comparison result will not be obtained.

(1) When A/D-Converted Value Addition/Average Mode is Not Selected

- Flush-right format
  Set bits 11 to 0 to the upper-side comparison level. Write 0 to bits 15 to 12.
- Flush-left format
  Set bits 15 to 4 to the upper-side comparison level. Write 0 to bits 3 to 0.

(2) When A/D-Converted Value Average Mode is Selected

- Flush-right format
  Set bits 11 to 0 to the upper-side comparison level for comparison with the A/D-converted value of the same channel. Write 0 to bits 15 to 12.
- Flush-left format
  Set bits 15 to 4 to the upper-side comparison level for comparison with the A/D-converted value of the same channel. Write 0 to bits 3 to 0.

A/D-converted value average mode can be set only when two or four times is selected in A/D-converted value addition mode.

(3) When A/D-Converted Value Addition Mode is Selected

- Flush-right format (A/D-converted value addition mode and 1-time to 4-time conversion selected)
  Set bits 13 to 0 to the upper-side comparison level for comparison with the A/D-converted value of the same channel. Write 0 to bits 15 and 14.
- Flush-right format (A/D-converted value addition mode and 16-time conversion selected)
  Set bits 15 to 0 to the upper-side comparison level for comparison with the A/D-converted value of the same channel.

Downloaded from Arrow.com.

- Flush-left format (A/D-converted value addition mode and 1-time to 4-time conversion selected)
  Set bits 15 to 2 to the upper-side comparison level for comparison with the A/D-converted value of the same channel. Write 0 to bits 1 and 0.
- Flush-left format (A/D-converted value addition mode and 16-time conversion selected)
  Set bits 15 to 0 to the upper-side comparison level for comparison with the A/D-converted value of the same channel.

When A/D-converted addition mode is selected, set the value added by the A/D-converted value of the same channel. The number of A/D conversions can be set to 1, 2, 3, 4, or 16 times. If A/D-converted addition mode is selected, when the A/D conversion count is set to 1 to 4 times, set the number of conversion accuracy bits extended by 2 bits in the ADWINULB register; when the A/D conversion count is set to 16 times, set the number of conversion accuracy bits extended by 4 bits in the ADWINULB register.
Even if A/D converted value addition mode is selected, set the reference data in the A/D data register according to the settings of the A/D data register format select bits.

44.2.35 A/D Compare Function Window B Channel Status Register (ADCMPBSR)

Address(es): S12AD.ADCMPBSR 0008 90ACh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	CMPSTB
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CMPSTB	Compare Window B Flag	0: Comparison conditions are not met. 1: Comparison conditions are met.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADCMPBSR register stores the comparison result of the compare window B function.

CMPSTB Flag (Compare Window B Flag)

This flag is a status flag that indicates the results of comparison for the channels (AN000 to AN007, AN016 to AN020, AN027, temperature sensor, and internal reference voltage) to which the window B comparison conditions are applied. When the comparison condition set by ADCMPBNSR.CMPCHB[5:0] bits is met at the end of A/D conversion, this flag is set to 1.
Writing 1 to the CMPSTB flag is disabled.
[Setting condition]

- The condition set by ADCMPBNSR.CMPLB bit is met when ADCMPCR.CMPAE = 1

[Clearing condition]

- 0 is written after reading 1

Downloaded from Arrow.com.

44.2.36 A/D Data Storage Buffer Register n (ADBUFn) (n = 0 to 15)

Address(es): S12AD.ADBUF0 0008 90B0h, S12AD.ADBUF1 0008 90B2h, S12AD.ADBUF2 0008 90B4h, S12AD.ADBUF3 0008 90B6h, S12AD.ADBUF4 0008 90B8h, S12AD.ADBUF5 0008 90BAh, S12AD.ADBUF6 0008 90BCh, S12AD.ADBUF7 0008 90BEh, S12AD.ADBUF8 0008 90C0h, S12AD.ADBUF9 0008 90C2h, S12AD.ADBUF10 0008 90C4h, S12AD.ADBUF11 0008 90C6h, S12AD.ADBUF12 0008 90C8h, S12AD.ADBUF13 0008 90CAh, S12AD.ADBUF14 0008 90CCh, S12AD.ADBUF15 0008 90CEh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

A/D data storage buffer registers n (ADBUFn) are 16-bit read-only registers that sequentially store all A/D converted values. The automatic clear function is not applied to these registers.

The ADBUFn register uses different formats depending on the following conditions.

- Settings of the A/D data register format select bit (flush-right or flush-left)
- Settings of the A/D-converted value addition/average function select register (A/D-converted value average mode selected or not selected)
- Settings of the A/D-converted value addition/average count select register (addition/average mode selected, addition count selected)

(1) When A/D-Converted Value Addition/Average Mode is Not Selected

- Flush-right format
  The A/D-converted value is stored in bits 11 to 0. Bits 15 to 12 are read as 0.
- Flush-left format
  The A/D-converted value is stored in bits 15 to 4. Bits 3 to 0 are read as 0.

(2) When A/D-Converted Value Average Mode is Selected

- Flush-right format
  The mean value of the A/D converted results of the same channel is stored in bits 11 to 0. Bits 15 to 12 are read as 0.
- Flush-left format
  The mean value of the A/D converted results of the same channel is stored in bits 15 to 4. Bits 3 to 0 are read as 0.

A/D-converted value average mode can be set only when two or four times is selected in A/D-converted value addition mode.

(3) When A/D-Converted Value Addition Mode is Selected

- Flush-right format (A/D-converted value addition mode and 1-time to 4-time conversion selected)
  The value added by the A/D-converted value of the same channel is stored in bits 13 to 0.
  Bits 15 and 14 are read as 0.
- Flush-right format (A/D-converted value addition mode and 16-time conversion selected)
  The value added by the A/D-converted value of the same channel is stored in bits 15 to 0.
- Flush-left format (A/D-converted value addition mode and 1-time to 4-time conversion selected)
  The value added by the A/D-converted value of the same channel is stored in bits 15 to 2.
  Bits 1 and 0 are read as 0.
- Flush-left format (A/D-converted value addition mode and 16-time conversion selected)
  The value added by the A/D-converted value of the same channel is stored in bits 15 to 0.

When A/D-converted addition mode is selected, the value added by the A/D-converted value of the same channel is indicated. The number of A/D conversion can be set to 1, 2, 3, 4, or 16 times. If A/D-converted addition mode is selected, when the conversion count is set to 1 to 4 times, the value added by the A/D conversion result is retained in the ADBUFn register as 2-bit extended data of the conversion accuracy bits: when the conversion count is set to 16 times, the value

Downloaded from Arrow.com.

added by the A/D conversion result is retained in the ADBUFn register as 4-bit extended data of the conversion accuracy bits.
Even if A/D-converted value addition mode is selected, the extended A/D-converted value is stored in the ADBUFn register according to the settings of the A/D data register format select bits.

44.2.37 A/D Data Storage Buffer Enable Register (ADBUFEN)

Address(es): S12AD.ADBUFEN 0008 90D0h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	BUFEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	BUFEN	Data Storage Buffer Enable	0: The data storage buffer is not used. 1: The data storage buffer is used.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADBUFEN register is used to enable the data storage buffer.

BUFEN Bit (Data Storage Buffer Enable)

This bit enables the use of the data storage buffer when using the compare function.
When BUFEN = 1, A/D conversion result (addition result) other than self-diagnosis result is stored in ADBUFn.
Disable the data storage operation (BUFEN = 0) before reading ADBUFn and ADBUFPTR.
Do not use the data storage buffer for data duplexing, continuous scan, or group scan.

Downloaded from Arrow.com.

44.2.38 A/D Data Storage Buffer Pointer Register (ADBUFPTR)

Address(es): S12AD.ADBUFPTR 0008 90D2h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	PTROVF	BUFPTR[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	BUFPTR[3:0]	Data Storage Buffer Pointer	These bits indicate the number of data storage buffer to which the next A/D conversion data is transferred.	R/W
b4	PTROVF	Pointer Overflow Flag	0: The data storage buffer pointer has not overflowed. 1: The data storage buffer pointer has overflowed.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADBUFPTR register is used for the data storage buffer pointer.

BUFPTR[3:0] Bits (Data Storage Buffer Pointer)

These read-only bits indicate the number of data storage buffer to which the next A/D conversion data is transferred. When data has been transferred to data storage buffer 15, the pointer value becomes 0000b and the PTROVF flag is set to 1. When the next data has been transferred, the data in data storage buffer 0 is overwritten.
Writing 00h to this register clears the value of these bits. Writing a value other than 00h is disabled.

PTROVF Flag (Pointer Overflow Flag)

This read-only flag indicates whether the data storage buffer pointer has overflowed. This flag is set to 1 when the pointer value becomes 0000b (overflow).
Writing 00h to this register clears this flag value. Writing a value other than 00h is disabled.

Downloaded from Arrow.com.

44.3 Operation

44.3.1 Scanning Operation

In scanning, A/D conversion is performed sequentially on the analog inputs of the specified channels.
A scan conversion is performed in three operating modes: single scan mode, continuous scan mode, and group scan mode. Also, conversion modes are divided into high-speed conversion mode and normal conversion mode. In single scan mode, one or more specified channels are scanned once. In continuous scan mode, one or more specified channels are scanned repeatedly until the ADCSR.ADST bit is cleared to 0 from 1 by software. In group scan mode, the selected channels of group A and the selected channels of group B are scanned once after starting to be scanned according to the respective synchronous trigger.
In single scan mode and continuous scan mode, A/D conversion is performed for ANn channels selected by the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n. In group scan mode, A/D conversion is performed for ANn channels of group A and group B selected by the ADANSA0, ADANSA1, ADANSB0, and ADANSB1 registers, respectively, starting from the channel with the smallest number n.
When self-diagnosis is selected, it is executed once at the beginning of each scan and one of the three voltages internally generated in the 12-bit A/D converter is converted.
When performing A/D conversion of the temperature sensor output or internal reference voltage, execute scanning individually.
Double trigger mode is to be used with single scan mode or group scan mode. With double trigger mode being enabled, A/D conversion data of a channel selected by the ADCSR.DBLANS[4:0] bits is duplicated only if the conversion is started by the synchronous trigger selected by the ADSTRGR.TRSA[5:0] bits.

Downloaded from Arrow.com.

44.3.2 Single Scan Mode

44.3.2.1 Basic Operation

In basic operation of single scan mode, A/D conversion is performed once on the analog input of the specified channels as below.

(1) When the ADCSR.ADST bit is set to 1 (A/D conversion start) by software, synchronous trigger, or asynchronous trigger input, A/D conversion is performed for ANn channels selected by the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
(2) Each time A/D conversion of a single channel is completed, the A/D conversion result is stored into the corresponding A/D data register (ADDRy).
(3) When A/D conversion of all the selected channels is completed, an S12ADI0 interrupt request is generated if the ADCSR.ADIE bit is 1 (S12ADI0 interrupt upon scanning completion enabled).
(4) The ADST bit remains 1 (A/D conversion start) during A/D conversion, and is automatically cleared to 0 when A/D conversion of all the selected channels is completed. Then the 12-bit A/D converter enters a wait state.

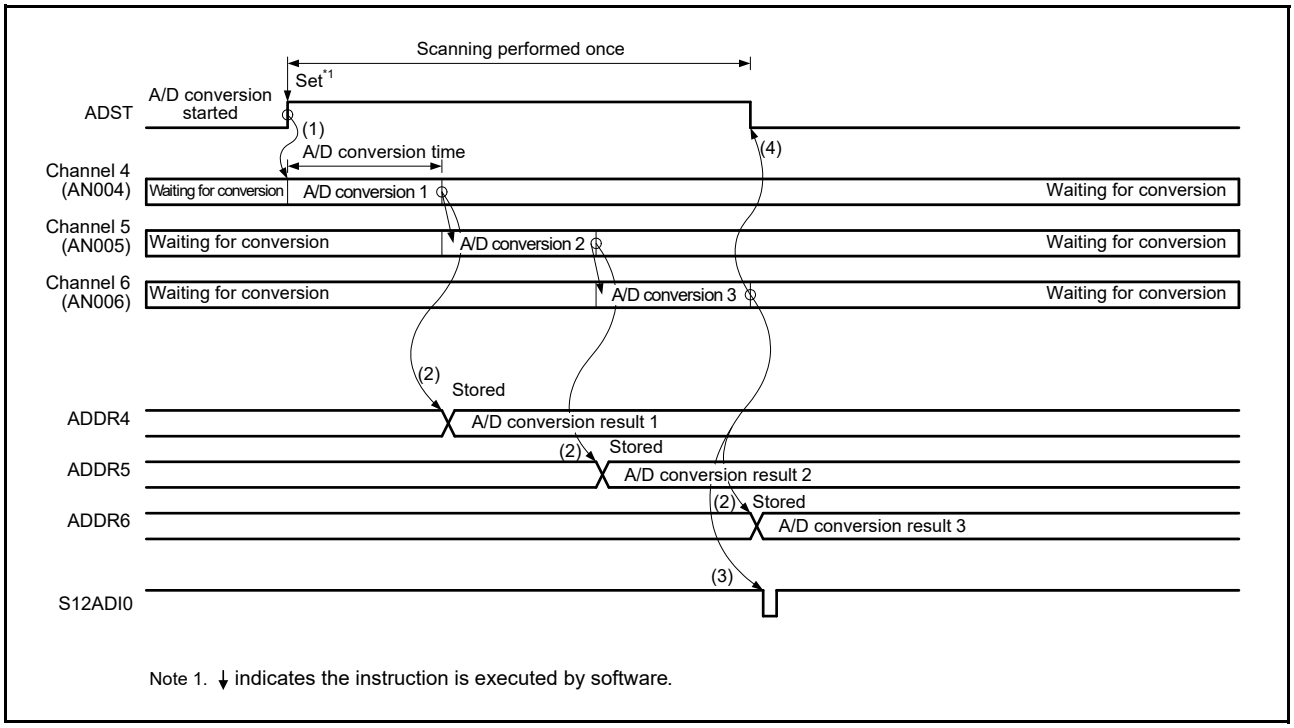


Figure 44.5 Example of Operation in Single Scan Mode (Basic Operation: AN004, AN005, AN006 Selected)

Downloaded from Arrow.com.

44.3.2.2 Channel Selection and Self-Diagnosis

When channels and self-diagnosis are selected, A/D conversion is performed once for the reference voltage VREFH0 supplied to the 12-bit A/D converter as below. After that, A/D conversion is performed only once on the analog input of the selected channels.

(1) A/D conversion for self-diagnosis is started when the ADCSR.ADST bit is set to 1 (A/D conversion start) by software, synchronous trigger, or asynchronous trigger input.
(2) When A/D conversion for self-diagnosis is completed, A/D conversion result is stored into the A/D self-diagnosis data register (ADRD), and A/D conversion is performed for ANn channels selected by the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
(3) Each time A/D conversion of a single channel is completed, the A/D conversion result is stored into the corresponding A/D data register (ADDRy).
(4) When A/D conversion of all the selected channels is completed, an S12ADI0 interrupt request is generated if the ADCSR.ADIE bit is 1 (S12ADI0 interrupt upon scanning completion enabled).
(5) The ADST bit remains 1 (A/D conversion start) during A/D conversion, and is automatically cleared to 0 when A/D conversion of all the selected channels is completed. Then the 12-bit A/D converter enters a wait state.

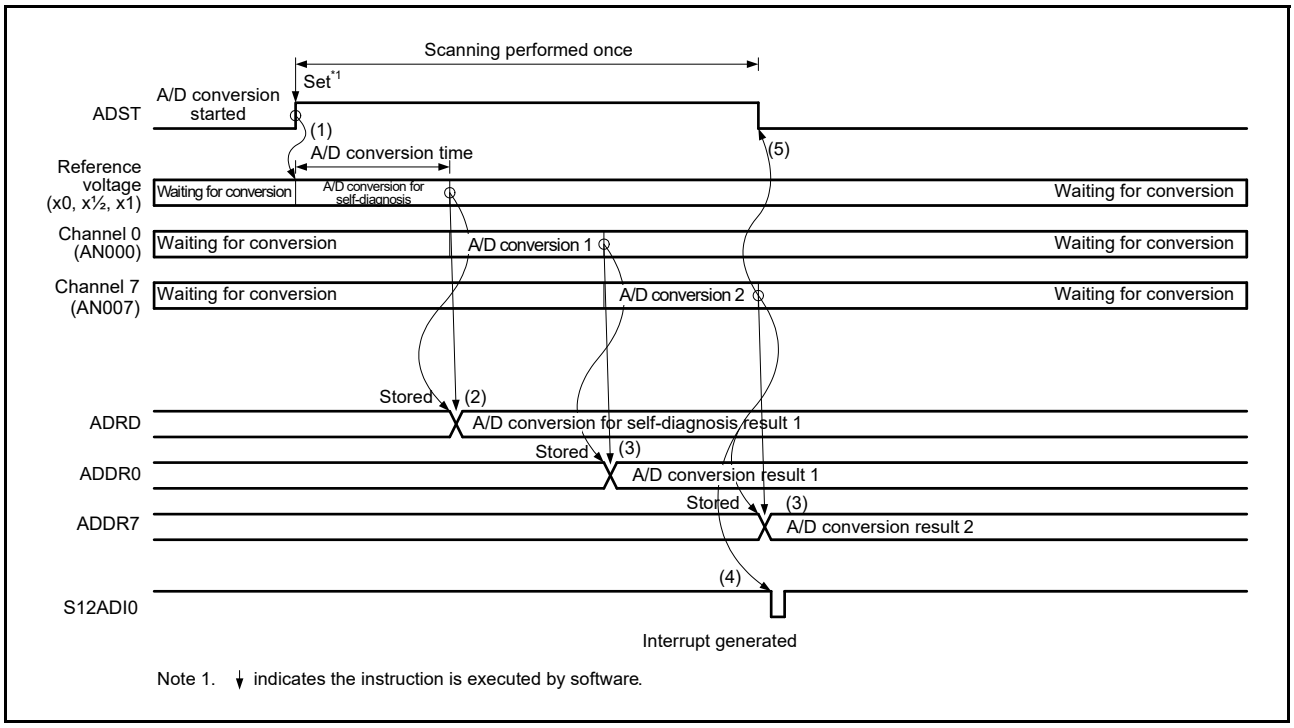


Figure 44.6 Example of Operation in Single Scan Mode (Basic Operation: AN000, AN007 Selected + Self-Diagnosis)

Downloaded from Arrow.com.

44.3.2.3 A/D Conversion of Temperature Sensor Output/Internal Reference Voltage

A/D conversion of the temperature sensor output and internal reference voltage is performed in single scan mode as below.

All channels should be deselected (by setting the ADANSA0 and ADANSA1 register bits to all 0 and the ADCSR.DBLE bit to 0). When selecting A/D conversion of the temperature sensor output, the A/D conversion select bit for the internal reference voltage (ADEXICR.OCSA) should be set to 0 (deselected). When selecting A/D conversion of the internal reference voltage, the A/D conversion select bit for the temperature sensor output (ADEXICR.TSSA) should be set to 0 (deselected).

(1) Set the sampling time to 5 µs or longer.
(2) After switching to A/D conversion of the internal reference voltage or the temperature sensor output, start A/D conversion by setting the ADST bit to 1.
(3) When A/D conversion is completed, the conversion result is stored into the corresponding A/D temperature sensor data register (ADTSDR) or A/D internal reference voltage data register (ADOCDR). If the ADCSR.ADIE bit is 1 (S12ADI0 interrupt upon scanning completion enabled), an S12ADI0 interrupt request is generated.
(4) The ADST bit remains 1 during A/D conversion, and is automatically cleared to 0 upon completion of A/D conversion. Then the 12-bit A/D converter enters a wait state.

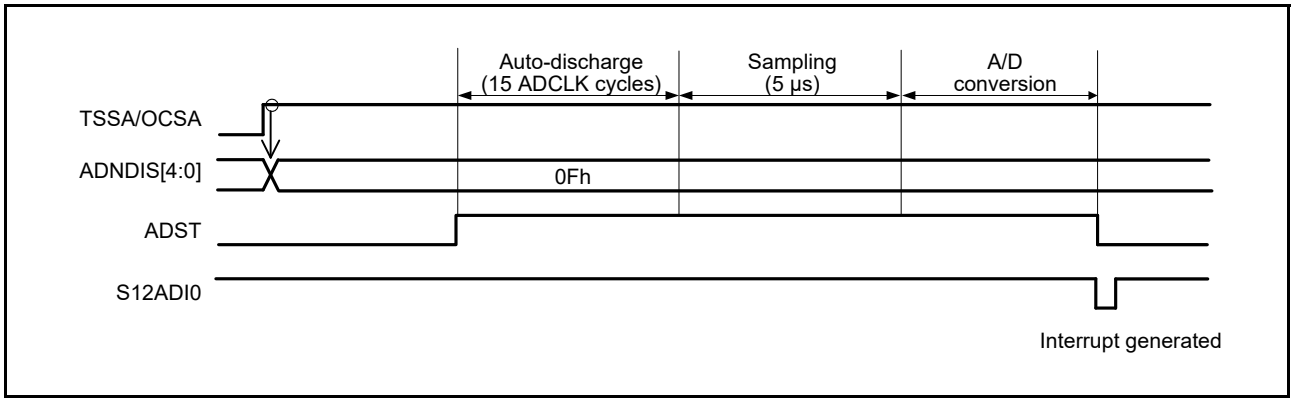


Figure 44.7 Example of Operation in Single Scan Mode (Temperature Sensor Output or Internal Reference Voltage Selected)

Downloaded from Arrow.com.

44.3.2.4 A/D Conversion in Double Trigger Mode

In single scan mode with double trigger mode, single scan operation started by synchronous trigger is performed twice as below.

Self-diagnosis should be deselected, and the temperature sensor output A/D conversion select bit (ADEXICR.TSSA) and the internal reference voltage A/D conversion select bit (ADEXICR.OCSA) should be set to 0.

Duplication of A/D conversion data is enabled by setting the channel numbers to be duplicated to the ADCSR.DBLANS[4:0] bits and setting the ADCSR.DBLE bit to 1. When the DBLE bit in ADCSR is set to 1, channel selection using the ADANSA0 and ADANSA1 registers is invalid. In double trigger mode, synchronous triggers should be selected using the ADSTRGR.TRSA[5:0] bits, the ADCSR.EXTRG bit should be set to 0, and the ADCSR.TRGE bit should be set to 1. Software trigger should not be used.

(1) When the ADCSR.ADST bit is set to 1 (A/D conversion start) by synchronous trigger input, A/D conversion is started on the single channel selected by the ADCSR.DBLANS[4:0] bits.
(2) When A/D conversion is completed, the A/D conversion result is stored into the corresponding A/D data register (ADDRy).
(3) The ADST bit is automatically cleared to 0 and the 12-bit A/D converter enters a wait state. Here, an S12ADI0 interrupt request is not generated irrespective of the ADCSR.ADIE bit setting (S12ADI0 interrupt upon scanning completion enabled).
(4) When the ADCSR.ADST bit is set to 1 (A/D conversion start) by the second trigger input, A/D conversion is started on the single channel selected by the ADCSR.DBLANS[4:0] bits.
(5) When A/D conversion is completed, the A/D conversion result is stored into the A/D data duplication register (ADDBLDR), which is exclusively used in double trigger mode.
(6) If the ADCSR.ADIE bit is 1 (S12ADI0 interrupt upon scanning completion enabled), an S12ADI0 interrupt request is generated.
(7) The ADST bit remains 1 (A/D conversion start) during A/D conversion, and is automatically cleared to 0 when A/D conversion is completed. Then the 12-bit A/D converter enters a wait state.

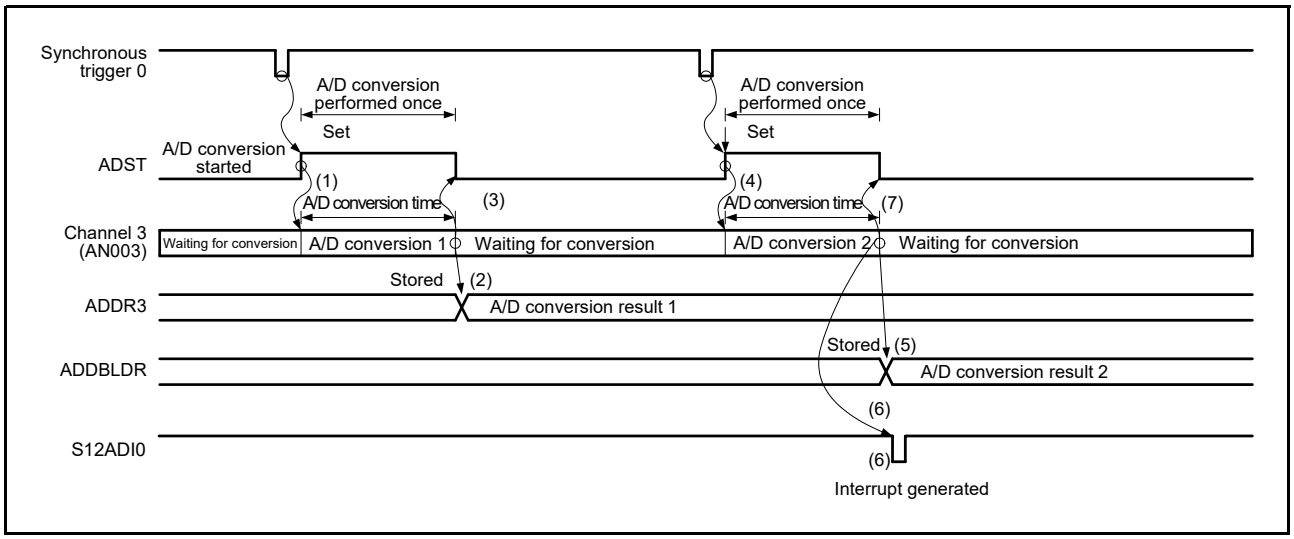


Figure 44.8 Example of Operation in Single Scan Mode (Double Trigger Mode Selected; AN003 Duplicated)

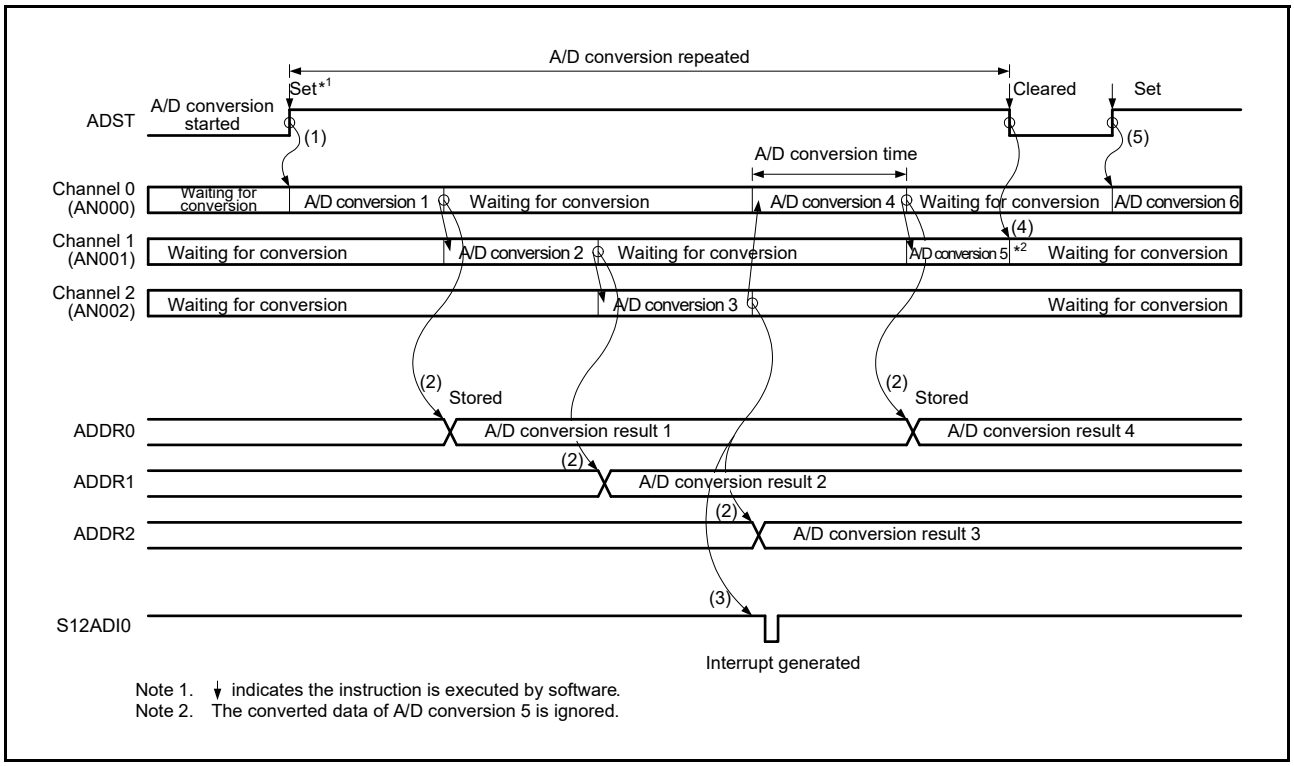
Downloaded from Arrow.com.

44.3.3 Continuous Scan Mode

44.3.3.1 Basic Operation

In basic operation of continuous scan mode, A/D conversion is performed repeatedly on the analog input of the specified channels as below.
In continuous scan mode, the temperature sensor output A/D conversion select bit (ADEXICR.TSSA) and the internal reference voltage A/D conversion select bit (ADEXICR.OCSA) should be set to 0 (deselected).

(1) When the ADCSR.ADST bit is set to 1 (A/D conversion start) by software, synchronous trigger, or asynchronous trigger input, A/D conversion is performed for ANn channels selected by the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
(2) Each time A/D conversion of a single channel is completed, the A/D conversion result is stored into the corresponding A/D data register (ADDRy).
(3) When A/D conversion of all the selected channels is completed, an S12ADI0 interrupt request is generated if the ADCSR.ADIE bit is 1 (S12ADI0 interrupt upon scanning completion enabled).
The 12-bit A/D converter sequentially starts A/D conversion for ANn channels selected by the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
(4) The ADCSR.ADST bit is not automatically cleared to 0 and steps 2 and 3 are repeated as long as the bit remains 1 (A/D conversion start). When the ADCSR.ADST bit is set to 0 (A/D conversion stop), A/D conversion stops and the 12-bit A/D converter enters a wait state.
(5) When the ADST bit is later set to 1 (A/D conversion start), A/D conversion is started again for ANn channels selected by the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.



Note 1. ↓ indicates the instruction is executed by software.
Note 2. The converted data of A/D conversion 5 is ignored.

Figure 44.9 Example of Operation in Continuous Scan Mode (Basic Operation: AN000 to AN002 Selected)

Downloaded from Arrow.com.

44.3.3.2 Channel Selection and Self-Diagnosis

When channels and self-diagnosis are selected at the same time, A/D conversion is first performed for the reference voltage VREFH0 supplied to the 12-bit A/D converter, and then A/D conversion is performed on the analog input of the selected channels, which sequence is repeated as below. In continuous scan mode, the temperature sensor output A/D conversion select bit (ADEXICR.TSSA) and the internal reference voltage A/D conversion select bit (ADEXICR.OCSA) should be set to 0 (deselected).

(1) When the ADCSR.ADST bit is set to 1 (A/D conversion start) by software, synchronous trigger, or asynchronous trigger input, A/D conversion for self-diagnosis is started first.
(2) When A/D conversion for self-diagnosis is completed, the A/D conversion result is stored into the A/D self-diagnosis data register (ADRD). A/D conversion is then performed for ANn channels selected by the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
(3) Each time A/D conversion of a single channel is completed, the A/D conversion result is stored into the corresponding A/D data register (ADDRy).
(4) When A/D conversion of all the selected channels is completed, an S12ADI0 interrupt request is generated if the ADCSR.ADIE bit is 1 (S12ADI0 interrupt upon scanning completion enabled). At the same time, the 12-bit A/D converter starts A/D conversion for self-diagnosis and then starts A/D conversion on ANn channels selected by the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
(5) The ADST bit is not automatically cleared and steps 2 to 4 are repeated as long as the bit remains 1. When the ADST bit is set to 0 (A/D conversion stop), A/D conversion stops and the 12-bit A/D converter enters a wait state.
(6) When the ADST bit is later set to 1 (A/D conversion start), the A/D conversion for self-diagnosis is started again.

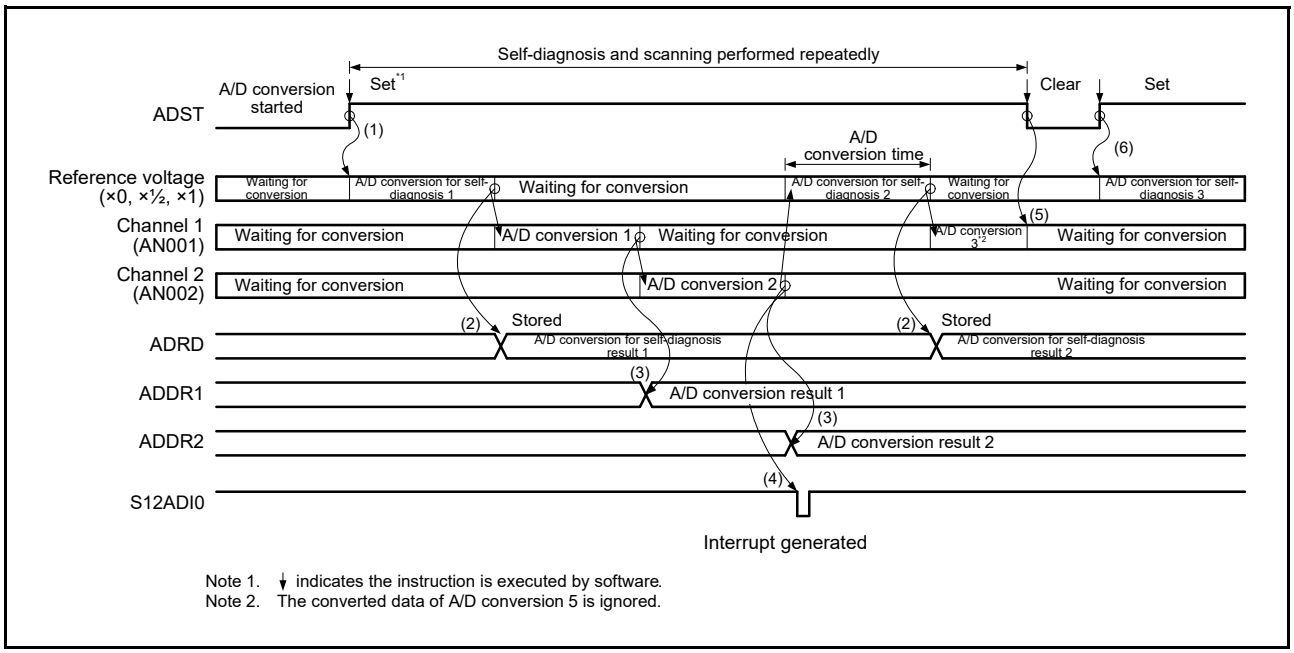


Figure 44.10 Example of Operation in Continuous Scan Mode (Basic Operation; AN001 and AN002 Selected + Self-Diagnosis)

Downloaded from Arrow.com.

44.3.4 Group Scan Mode

44.3.4.1 Basic Operation

In basic operation of group scan mode, A/D conversion is performed once on the analog inputs of all the specified channels in group A and group B after scanning is started by a synchronous trigger as below. Scan operation of each group is similar to the scan operation in single scan mode.
The synchronous triggers of group A and B can be selected using the TRSA[5:0] and TRSB[5:0] bits in ADSTRGR, respectively. The different triggers should be used for group A and group B to prevent simultaneous A/D conversion of group A and group B. Software trigger should not be used.
The group A channels to be A/D-converted are selected using the ADANSA0 and ADANSA1 registers while the group B channels to be A/D-converted are selected using the ADANSB0 and ADANSB1 registers. The same channels cannot be selected for both groups.
In group scan mode, the temperature sensor output A/D conversion select bit (ADEXICR.TSSA) and the internal reference voltage A/D conversion select bit (ADEXICR.OCSA) should be set to 0 (deselected).
When self-diagnosis is selected in group scan mode, self-diagnosis is separately executed for group A and group B.
The following describes operation in group scan mode using a trigger from the MTU. The TRG4AN and TRG4BN triggers from the MTU are assumed to be used to start conversion of group A and group B, respectively.

(1) Scanning of group A is started by the TRG4AN trigger from the MTU.
(2) When group A scanning is completed, an S12ADI0 interrupt is generated if the ADCSR.ADIE bit is 1 (S12ADI0 interrupt upon scanning completion enabled).
(3) Scanning of group B is started by the TRG4BN trigger from the MTU.
(4) When group B scanning is completed, a GBADI interrupt is generated if the ADCSR.GBADIE bit is 1 (GBADI interrupt upon scanning completion enabled).

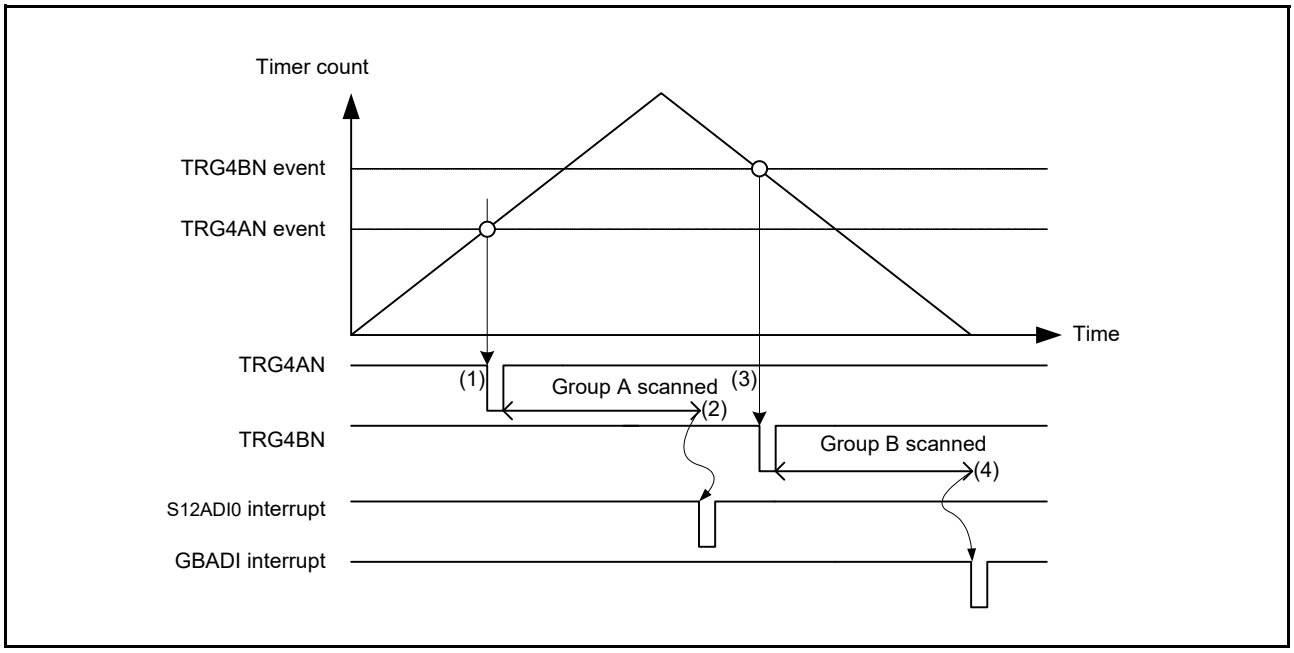


Figure 44.11 Example of Operation in Group Scan Mode (Basic Operation: Synchronous Triggers from MTU Used)

Downloaded from Arrow.com.

44.3.4.2 A/D Conversion in Double Trigger Mode

When double trigger mode is selected in group scan mode, two rounds of single scan operation started by a synchronous trigger are performed as a sequence for group A. For group B, single scan operation started by a synchronous trigger is performed once.
In group scan mode, the synchronous triggers of group A and B can be selected using the TRSA[5:0] and TRSB[5:0] bits in ADSTRGR, respectively. The different triggers should be used for group A and group B to prevent simultaneous A/D conversion of group A and group B. Software trigger and asynchronous trigger should not be used.
The group A and group B channels to be A/D-converted are selected using the ADCSR.DBLANS[4:0] bits and the ADANSB0 and ADANSB1 registers, respectively. The same channels cannot be selected for both groups.
In group scan mode, the temperature sensor output A/D conversion select bit (ADEXICR.TSSA) and the internal reference voltage A/D conversion select bit (ADEXICR.OCSA) should be set to 0 (deselected).
When double trigger mode is selected in group scan mode, self-diagnosis cannot be selected.
Duplication of A/D conversion data is enabled by setting the channel numbers to be duplicated to the ADCSR.DBLANS[4:0] bits and setting the ADCSR.DBLE bit to 1.
The following describes operation in group scan mode with double trigger mode using a synchronous trigger from the MTU. The TRG4ABN and TRG0AN triggers from the MTU are assumed to be used to start conversion of group A and group B, respectively.

(1) Scanning of group B is started by the TRG0AN trigger from the MTU.
(2) When group B scanning is completed, a GBADI interrupt is generated if the ADCSR.GBADIE bit is 1 (GBADI interrupt upon scanning completion enabled).
(3) The first scanning of group A is started by the first TRG4ABN trigger from the MTU.
(4) When the first scanning of group A is completed, the conversion result is stored into the corresponding A/D data register (ADDRy); an S12ADI0 interrupt request is not generated irrespective of the ADIE bit setting in ADCSR.
(5) The second scanning of group A is started by the second TRG4ABN trigger from the MTU.
(6) When the second scanning of group A is completed, the conversion result is stored into ADDBLDR. An S12ADI0 interrupt is generated if the ADCSR.ADIE bit is 1 (S12ADI0 interrupt upon scanning completion enabled).

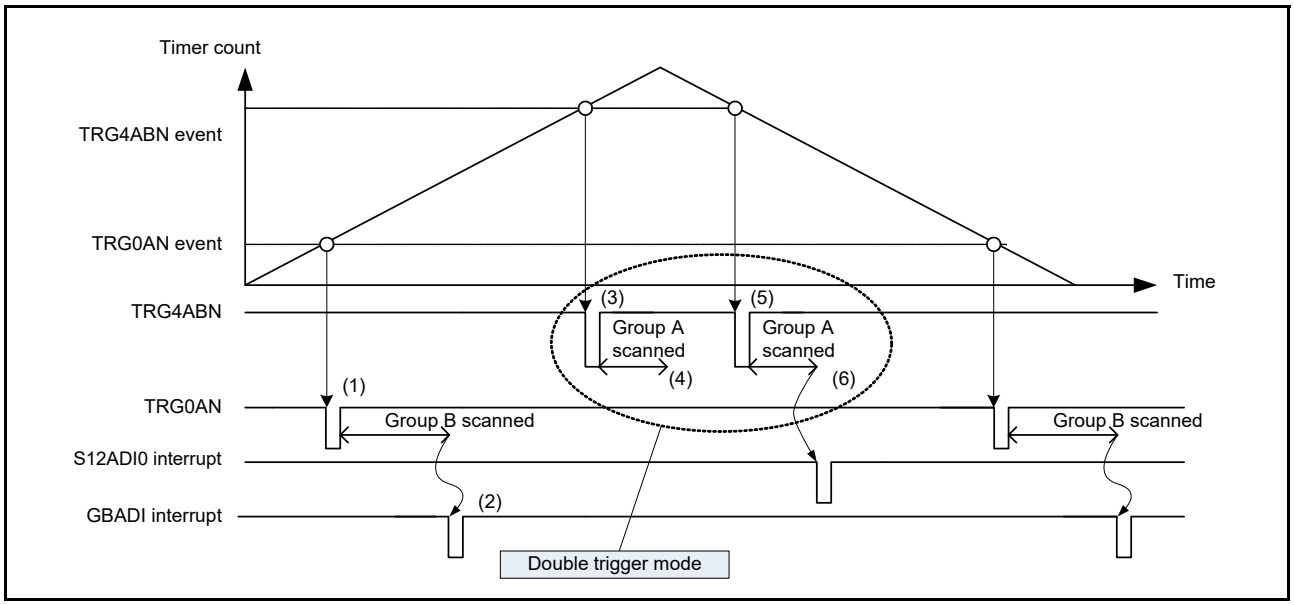


Figure 44.12 Example of Operation in Group Scan Mode with Double Trigger Mode (Basic Operation: Synchronous Triggers from MTU Used)

Downloaded from Arrow.com.

44.3.4.3 Operation under Group-A Priority Control

Setting the PGS bit in the A/D group scan priority control register (ADGSPCR) to 1 in group scan mode makes operation proceed under group-A priority control. When setting the PGS bit in the ADGSPCR register to 1, follow the procedure described in Figure 44.13. If the procedure is not followed, A/D conversion operation and stored data are not guaranteed.
In operation in basic group scan mode, input of the trigger for the other group during operation for A/D conversion in group A or group B is ignored. Under group-A priority control, if a group-A trigger is input during A/D conversion for group B, A/D conversion for group B is discontinued and A/D conversion for group A proceeds. If the setting of the ADGSPCR.GBRSCN bit is 0, the converter enters a wait state on completion of the A/D conversion for group A. If the setting of the ADGSPCR.GBRSCN bit is 1, the converter automatically restarts scanning for group B from the head of the group after completion of the A/D conversion for group A. Table 44.9 summarizes operations in response to the input of a trigger during A/D conversion with the settings of the ADGSPCR.GBRSCN bit.

Scan operations in group A or group B are the same in single scan mode. Furthermore, single scanning continues to proceed if the ADGSPCR.GBRP bit is set to 1 during scanning operations for group B.

For the trigger settings in group scan mode, select a synchronous trigger for group A using the ADSTRGR.TRSA[5:0] bits and select a synchronous trigger different from that of group A for group B using the ADSTRGR.TRSB[5:0] bits. Set the ADSTRGR.TRSB[5:0] bits to 3Fh when setting the ADGSPCR.GBRP bit to 1. Furthermore, as targets for A/D conversion, select channels for group A using the ADANSA0 and ADANSA1 registers, and for group B, select channels different from those for group A using the ADANSB0 and ADANSB1 registers.

Downloaded from Arrow.com.

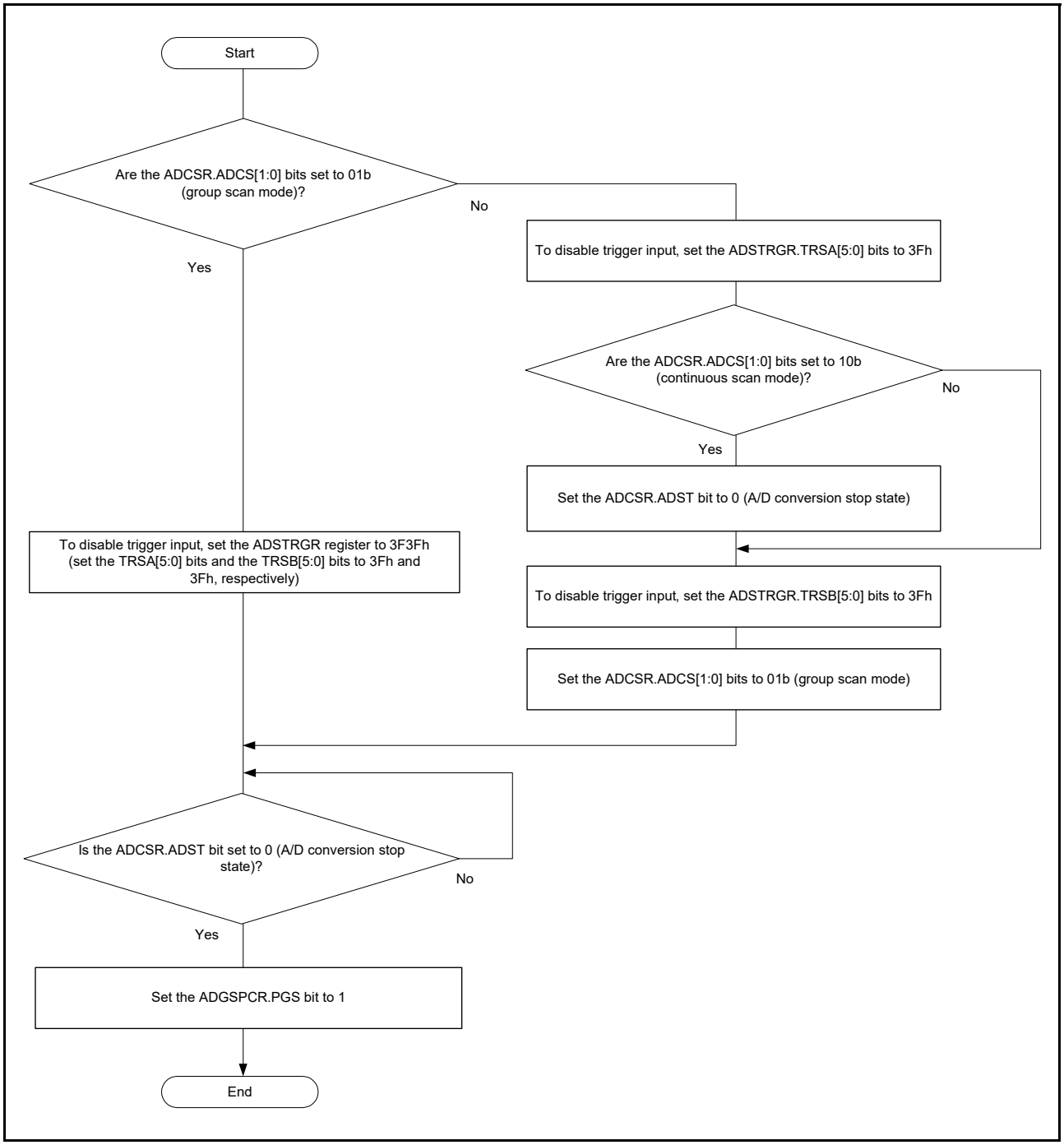


Figure 44.13 Flow of Setting the ADGSPCR.PGS Bit

Downloaded from Arrow.com.

Table 44.9 Control of A/D Conversion Operations According to the Settings of the ADGSPCR.GBRSCN Bit

A/D Conversion Operation	Trigger Input	ADGSPCR.GBRSCN = 0	ADGSPCR.GBRSCN = 1
When A/D conversion for group A is in progress	Input of trigger for group A	Trigger input is ineffective.	Trigger input is ineffective.
	Input of trigger for group B	Trigger input is ineffective.	A/D conversion is performed on group B after A/D conversion on group A is completed.
When A/D conversion for group B is in progress	Input of trigger for group A	Conversion for group B that is in progress is discontinued and conversion for group A starts.	• Conversion in progress for group B is discontinued and conversion for group A starts. • Conversion for group B starts after conversion for group A is completed.
	Input of trigger for group B	Trigger input is ineffective.	Trigger input is ineffective.

The following describes the operations in group scan mode under group-A priority control (i.e. ADGSPCR.GBRSCN = 1 and ADGSPCR.GBRP = 0) when channel 0 is selected for group A and channels 1 to 3 are selected for group B.

(1) When input of a trigger for group B sets the ADCSR.ADST bit to 1 (A/D conversion start), conversion for the ANn channels selected in the ADANSB0 and ADANSB1 registers, starting from the channel with the smallest number n.
(2) On completion of A/D conversion, the result is stored in the corresponding A/D data register (ADDRy).
(3) The ADCSR.ADST bit is cleared on the input of a trigger for group A while operation for A/D conversion in group B is in progress, and the latter is discontinued. After that, the ADCSR.ADST bit is set to 1 (A/D conversion start), and conversion for the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
(4) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).
(5) An S12ADI0 interrupt request is generated if the setting of the ADCSR.ADIE bit is 1 (S12ADI0 interrupt upon scanning completion enabled).
(6) After the ADST bit is automatically cleared, again, the bit is automatically set to 1 (A/D conversion start) and conversion for the ANn channels of group B selected in the ADANSB0 and ADANSB1 registers, starting from the channel with the smallest number n.
(7) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).
(8) A GBADI interrupt request is generated if the setting of the ADCSR.GBADIE bit is 1 (GBADI interrupt upon group B scanning completion enabled).
(9) The ADST bit remains 1 (A/D conversion start) during A/D conversion and is automatically cleared on completion of conversion, after which the A/D converter enters a wait state.

Downloaded from Arrow.com.

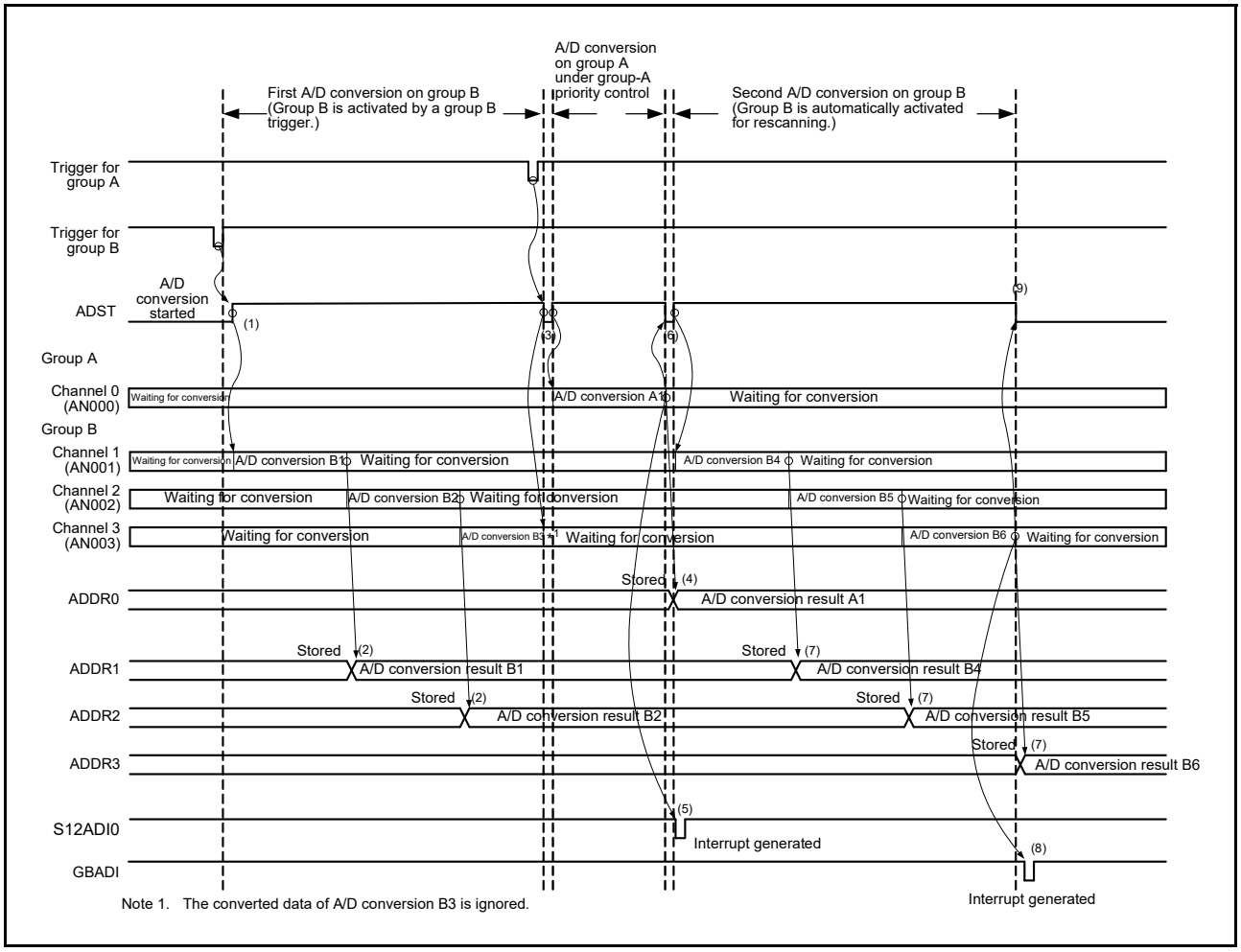


Figure 44.14 Example of Operations under Group-A Priority Control (1) (when ADGSPCR.GBRSCN = 1 and ADGSPCR.GBRP = 0)

The following is an example when a group A trigger is input again during rescanning operation on group B. In this example, channel 0 is selected for group A and channels 1 to 3 are selected for group B when operation on group A is given priority (ADGSPCR.GBRSCN = 1, ADGSPCR.GBRP = 0).

(1) When a group B trigger input sets the ADCSR.ADST bit to 1 (A/D conversion start), conversion for the ANn channels of group B selected in the ADANSB0 and ADANSB1 registers starts in order from the channel with the lowest number n.
(2) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).
(3) The ADCSR.ADST bit is cleared to 0 (A/D conversion stop) on the input of a trigger for group A while operation for A/D conversion in group B is in progress, and the latter is discontinued.
(4) After that, the ADCSR.ADST bit is set to 1 automatically and A/D conversion for the ANn group A channels selected in the ADANSA0 and ADANSA1 registers starts in order from the channel with the lowest number n.
(5) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).
(6) An S12ADI0 interrupt request is generated if the setting of the ADCSR.ADIE bit is 1 (S12ADI0 interrupt upon scanning completion enabled).

Downloaded from Arrow.com.

(7) On completion of A/D conversion on the group A, rescanning operation on group B sets the ADCSR.ADST bit to 1 automatically if the setting of the ADGSPCR.GBRSCN bit is 1 (rescanning operation enabled). After that, A/D conversion for the ANn group B channels selected in the ADANSB0 and ADANSB1 registers starts again in order from the channel with the lowest number n.

(8) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).

(9) If a group A trigger is input during A/D conversion on group B for rescanning, the ADCSR.ADST bit is cleared to 0 (A/D conversion stop) and the ongoing A/D conversion on group B is stopped.

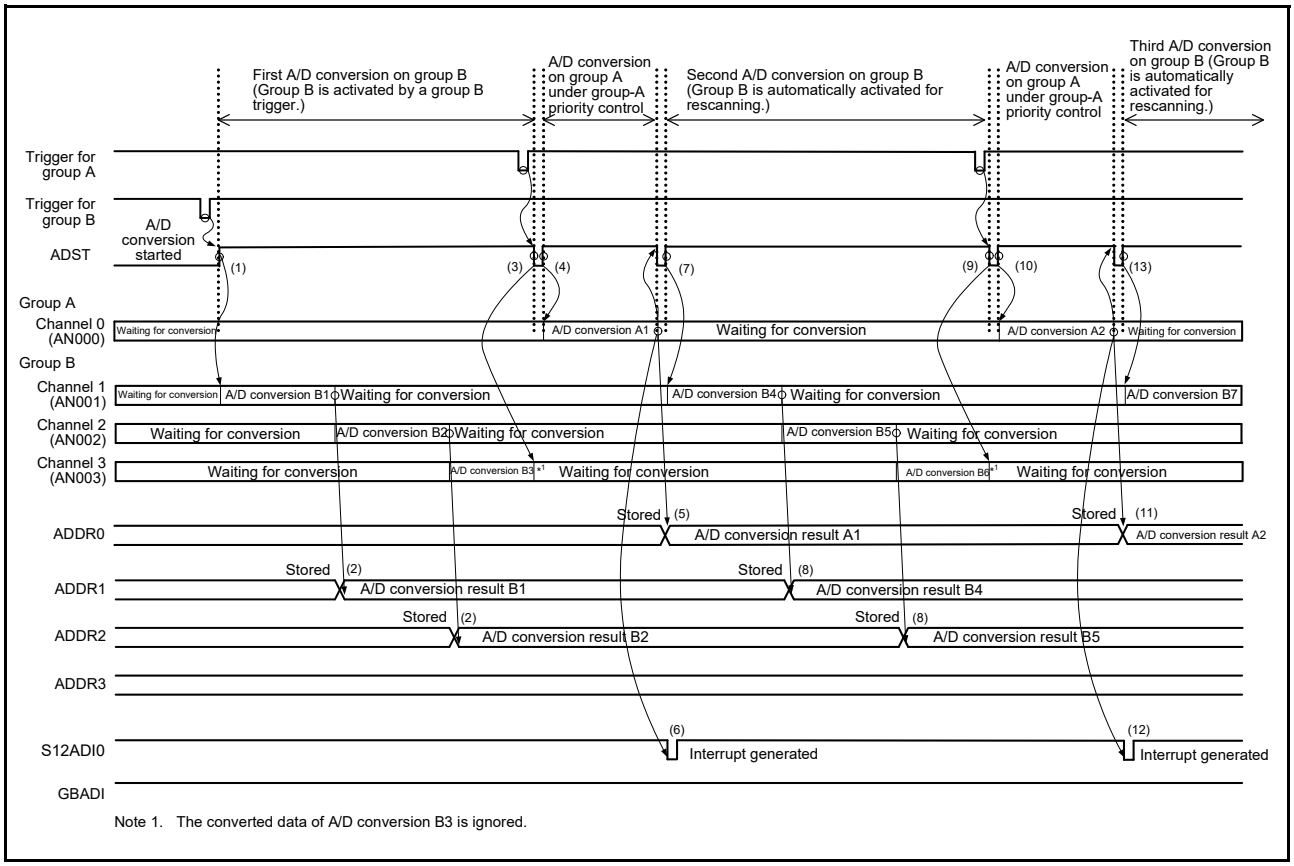
(10) After that, the ADCSR.ADST bit is set to 1 automatically and A/D conversion for the ANn group A channels selected in the ADANSA0 and ADANSA1 registers starts in order from the channel with the lowest number n.

(11) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).

(12) An S12ADI0 interrupt request is generated if the setting of the ADCSR.ADIE bit is 1 (S12ADI0 interrupt upon scanning completion enabled).

(13) On completion of A/D conversion on group A, rescanning operation on group B sets the ADCSR.ADST bit to 1 automatically if the setting of the ADGSPCR.GBRSCN bit is 1 (rescanning operation enabled). After that, A/D conversion for the ANn group B channels selected in the ADANSB0 and ADANSB1 registers starts again in order from the channel with the lowest number n.

(14) If a group A trigger is input during A/D conversion on group B for rescanning, steps 9 to 13 are repeated. If a group A trigger is not input, the ADCSR.ADST bit is cleared automatically on completion of A/D conversion on group B and the 12-bit A/D converter enters a wait state.



Note 1. The converted data of A/D conversion B3 is ignored.

Figure 44.15 Example of Operations under Group-A Priority Control (2) (when ADGSPCR.GBRSCN = 1 and ADGSPCR.GBRP = 0)

Downloaded from Arrow.com.

The following is an example of a rescanning operation in which a group B trigger is input during A/D conversion on group A. In this example, channels 1 to 3 are selected for group A and channel 0 is selected for group B when operation on group A is given priority (ADGSPCR.GBRSCN = 1, ADGSPCR.GBRP = 0).

(1) When input of a trigger for group A sets the ADCSR.ADST bit to 1 (A/D conversion start), conversion for the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
(2) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).
(3) If a group B trigger is input during A/D conversion on group A, A/D conversion on group B can be performed after the A/D conversion on group A is completed. (However, if group A triggers are input continuously, the scan operation on group B is canceled by group A and is not performed.)
(4) On completion of the A/D conversion on the group A, an S12ADI0 interrupt request is generated if the setting of the ADCSR.ADIE bit is 1 (S12ADI0 interrupt upon scanning completion enabled).
(5) On completion of the A/D conversion on the group A, activation of group B for rescanning sets the ADCSR.ADST bit to 1 automatically.
After that, conversion for the ANn channels of group B selected in the ADANSB0 and ADANSB1 registers, starting from the channel with the smallest number n.
(6) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).
(7) On completion of the rescanning operation on the group B, a GBADI interrupt request is generated if the setting of the ADCSR.GBADIE bit is 1 (GBADI interrupt upon scanning completion enabled).
(8) The ADST bit retains the value 1 (A/D conversion start) during A/D conversion and is automatically cleared on completion of conversion, after which the A/D converter enters a wait state.

Downloaded from Arrow.com.

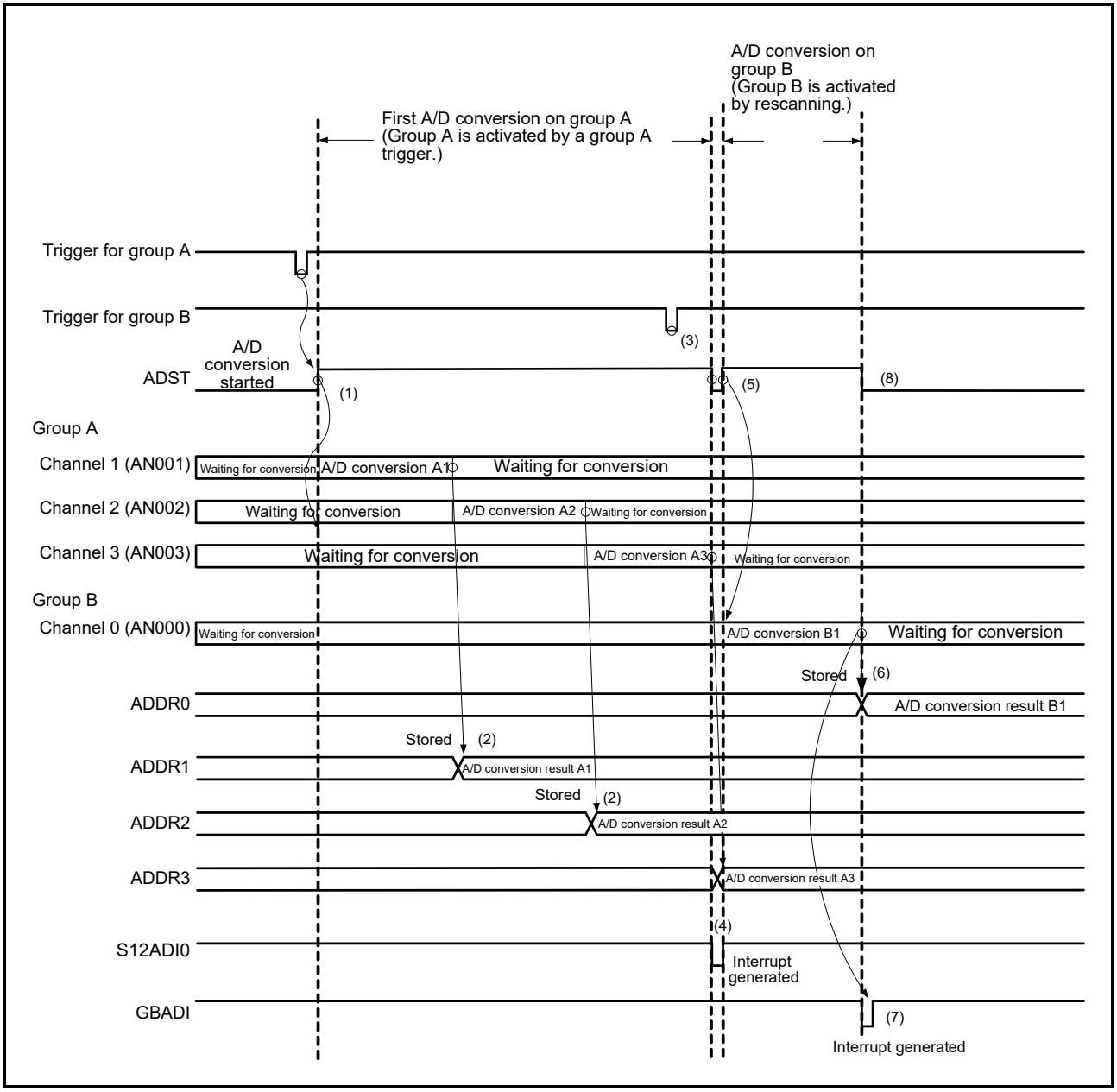
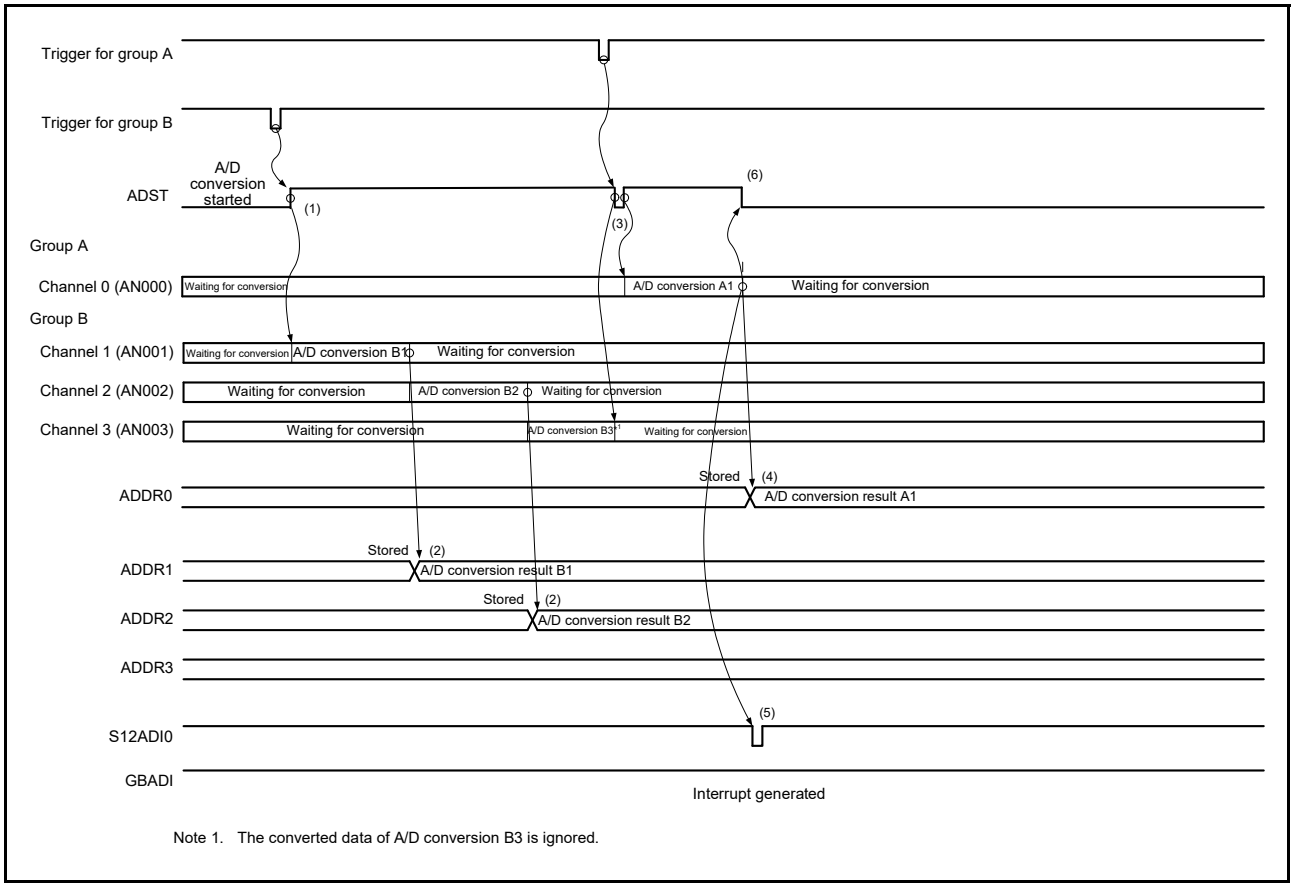


Figure 44.16 Example of Operations under Group-A Priority Control (3) (when ADGSPCR.GBRSCN = 1 and ADGSPCR.GBRP = 0)

Downloaded from Arrow.com.

The following is an example of operation under group-A priority control in which channel 0 is selected for group A and channels 1 to 3 are selected for group B (ADGSPCR.GBRSCN = 0, ADGSPCR.GBRP = 0).

(1) When input of a trigger for group B sets the ADCSR.ADST bit to 1 (A/D conversion start), conversion for the ANn channels selected in the ADANSB0 and ADANSB1 registers, starting from the channel with the smallest number n.
(2) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).
(3) If a group A trigger is input during A/D conversion on group B, the ADCSR.ADST bit is cleared to 0 and the ongoing A/D conversion on group B is stopped. After that, the ADCSR.ADST bit is set to 1 (A/D conversion start) and conversion for the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
(4) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).
(5) An S12ADI0 interrupt request is generated if the setting of the ADCSR.ADIE bit is 1 (S12ADI0 interrupt upon scanning completion enabled).
(6) The ADCSR.ADST bit retains the value 1 (A/D conversion start) during A/D conversion and is cleared on completion of conversion, after which the A/D converter enters a wait state.



Note 1. The converted data of A/D conversion B3 is ignored.

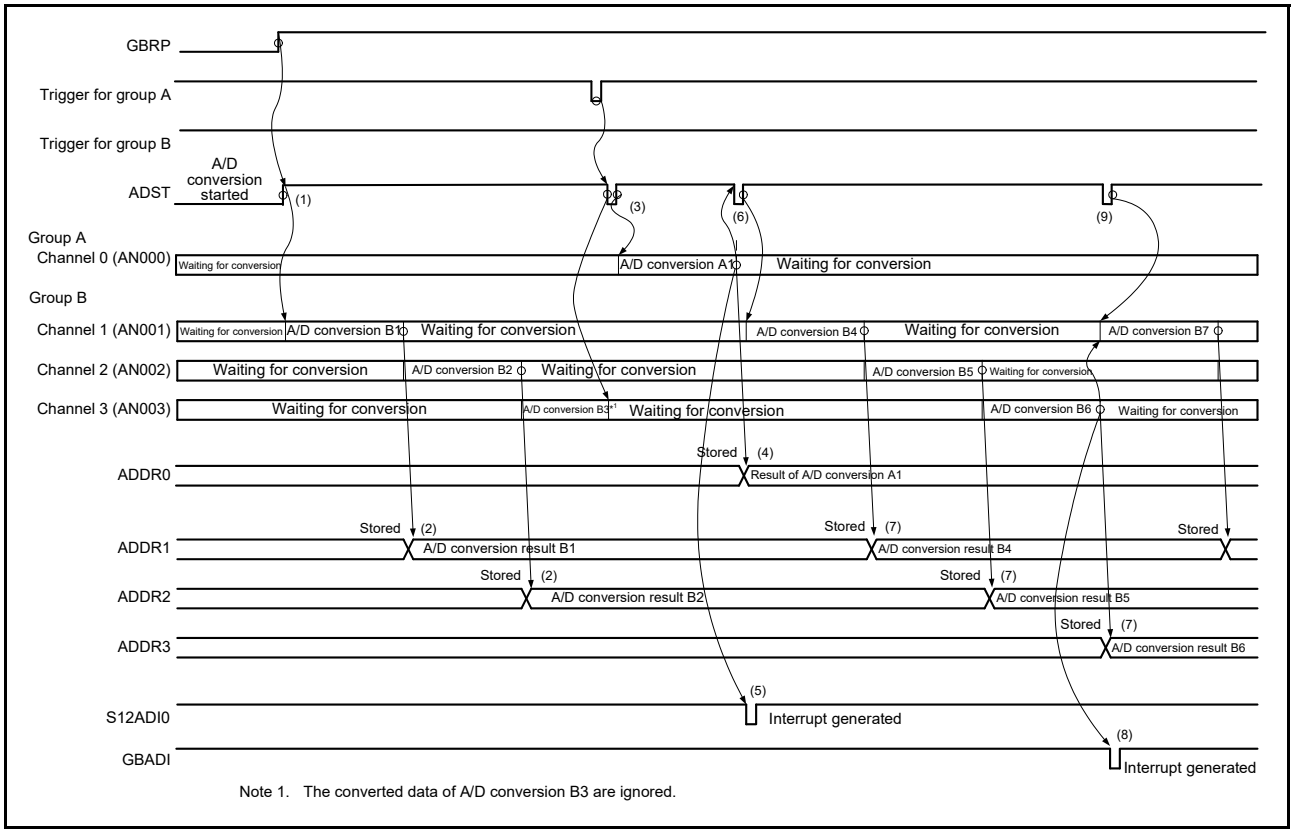
Figure 44.17 Example of Operation under Group-A Priority Control (4) (when ADGSPCR.GBRSCN = 0 and ADGSPCR.GBRP = 0)

Downloaded from Arrow.com.

The following is an example of operation under group-A priority control in which channel 0 is selected for group A and channels 1 to 3 are selected for group B (ADGSPCR.GBRP = 1).

(1) The ADCSR.ADST bit is set to 1 (A/D conversion start) when ADGSPCR.GBRP is set to 1, and conversion for the ANn channels selected in the ADANSB0 and ADANSB1 registers, starting from the channel with the smallest number n.
(2) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).
(3) If a group A trigger is input during A/D conversion on group B, the ADCSR.ADST bit is cleared to 0 and the ongoing A/D conversion on group B is stopped. After that, the ADCSR.ADST bit is set to 1 (A/D conversion start) and conversion for the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
(4) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).
(5) An S12ADI0 interrupt request is generated if the setting of the ADCSR.ADIE bit is 1 (S12ADI0 interrupt upon scanning completion enabled).
(6) After the ADST bit is automatically cleared, again, the ADCSR.ADST bit is automatically set to 1 (A/D conversion start) and conversion for the ANn channels selected in the ADANSB0 and ADANSB1 registers, starting from the channel with the smallest number n.
(7) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).
(8) A GBADI interrupt request is generated if the setting of the ADCSR. GBADIE bit is 1.
(9) After the ADST bit is automatically cleared, again, the bit is automatically set to 1 (A/D conversion start) and conversion for the ANn channels selected in the ADANSB0 and ADANSB1 registers, starting from the channel with the smallest number n. Steps 6 to 9 are repeated as long as the ADGSPCR.GBRP bit remains 1.
Clearing of the ADCSR.ADST bit to 0 is prohibited while the ADGSPCR.GBRP bit is set to 1. To forcibly stop A/D conversion when ADGSPCR.GBRP = 1, follow the procedures for clear operation by software through the ADCSR.ADST bit shown in section 44.8.2, Notes on Stopping A/D Conversion.

Downloaded from Arrow.com.



Note 1. The converted data of A/D conversion B3 are ignored.

Figure 44.18 Example of Operation under Group-A Priority Control (5) (when ADGSPCR.GBRP = 1)

Downloaded from Arrow.com.

44.3.5 Compare Function (Window A, Window B)

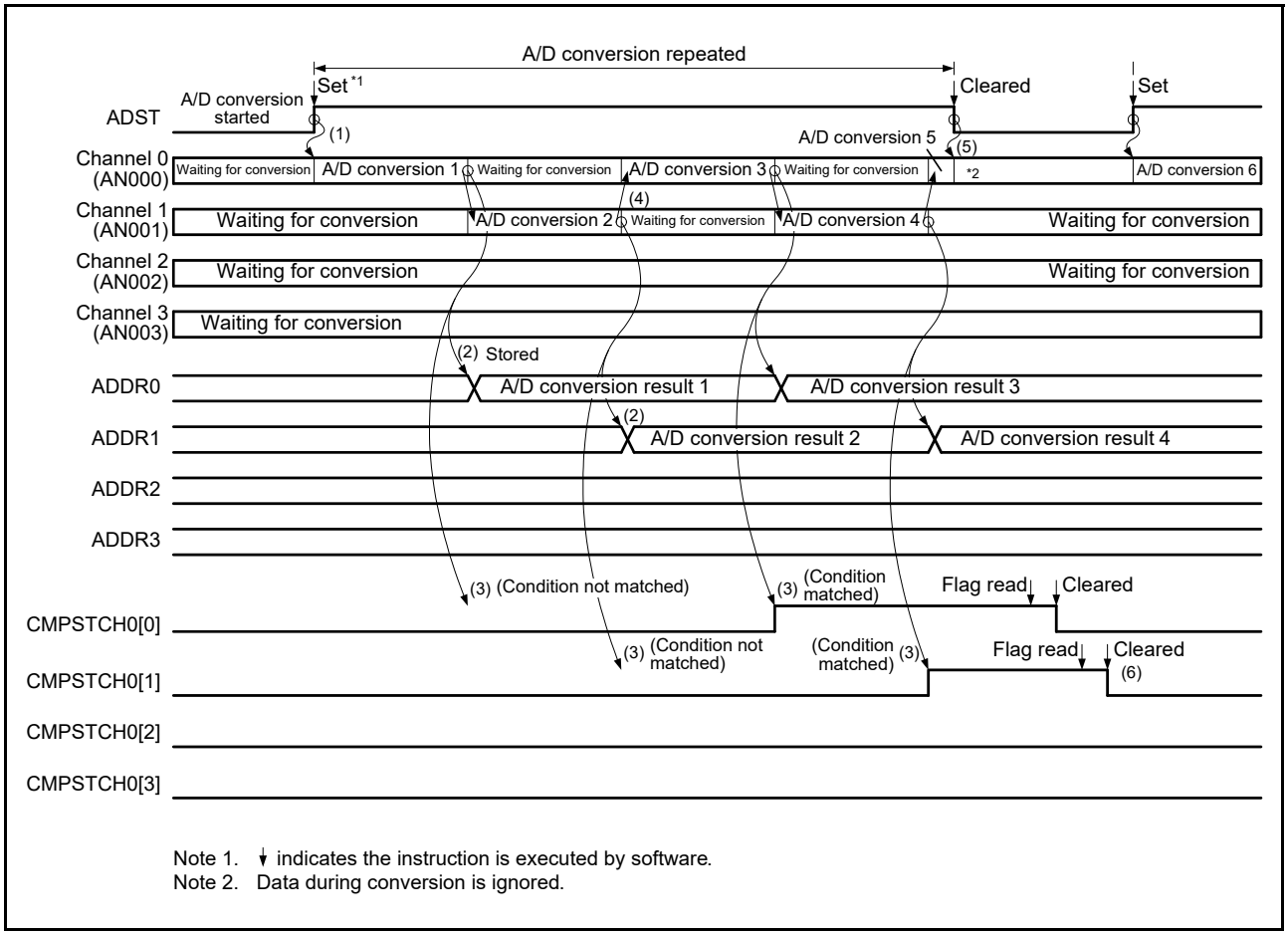
44.3.5.1 Compare Function Window A/B

The compare function compares the reference value set in the register with the A/D conversion result. The reference value can be set for window A and window B independently. When the compare function is in use, the self-diagnosis function and double trigger mode cannot be used. Big differences between window A and window B are different interrupt output signals and that window B can select only one channel.

The following describes operations in combination of continuous scan mode and the compare function.

(1) When the ADCSR.ADST bit is set to 1 (A/D conversion start) by software, synchronous trigger, or asynchronous trigger input A/D conversion starts in the order of the selected channel.
(2) Upon completion of A/D conversion, the result is stored in the corresponding A/D data register (ADDRy, ADTSDR, or ADOCDR). When ADCMPCR.CMPAE is 1, if the ADCMPANSR0, ADCMPANSR1, or ADCMPANSER register is set for window A, the results of A/D conversion are to be compared with the values set in the ADCMPDR0 and ADCMPDR1 registers. When ADCMPCR.CMPBE is 1, if the ADCMPBNSR register is set for window B, the results of A/D conversion are to be compared with the values set in the ADWINULB and ADWINLLB registers.
(3) As a result of the comparison, when window A meets the condition set in ADCMPDR0, ADCMPDR1 or ADCMPLER, the compare window A flag (ADCMPSR0.CMPSTCHA0n, ADCMPSR1.CMPSTCHA1n, ADCMPSER.CMPSTTSA, or ADCMPSER.CMPSTOCA) is set to 1. In the same way, when window B meets the condition set in ADCMPBNSR.CMPLB, the compare window B flag (ADCMPBSR.CMPSTB) is set to 1.
(4) Upon completion of all selected A/D conversions and comparisons, scan restarts.
(5) Set the ADCSR.ADST bit to 0 (A/D conversion stop), and execute processing for the channel with the compare flag set to 1.
(6) Clear all compare flags after processing is competed. To perform comparison again, restart A/D conversion.

Downloaded from Arrow.com.



Note 1. ↓ indicates the instruction is executed by software.
Note 2. Data during conversion is ignored.

Figure 44.19 Operation Example of Comparison (AN000 to AN003 Compared)

Downloaded from Arrow.com.

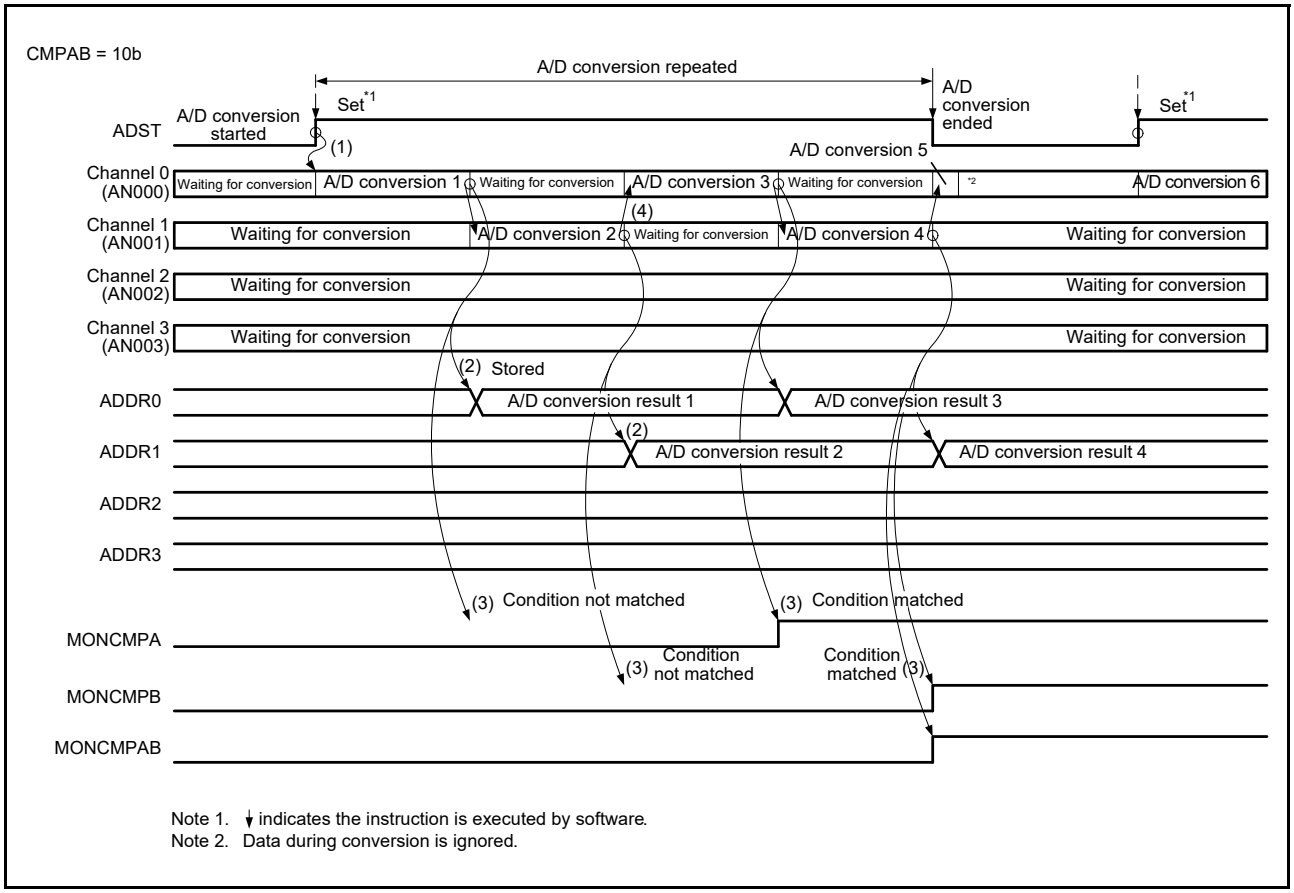
44.3.5.2 ELC Output of Compare Function

The ELC output of the compare function is used to specify the high-side reference value and the low-side reference value for window A and window B respectively, and to compare the A/D converted value of the selected channel with the high/low-side reference value. Depending on whether the comparison conditions for window A and window B are met or not met, the ELC event (S12ADWMELC/S12ADWUMELC) is output according to the event conditions (A or B, A and B, A exor B).
If multiple channels are selected for window A, when the comparison conditions for any of the channels are met, it is recognized that the comparison conditions for window A are met.

When using this function, perform A/D conversion in single scan mode.
Any channels from AN000 to AN007, AN016 to AN020, AN027, temperature sensor output, and internal reference voltage are selectable for window A.
However, when selecting the internal reference voltage or the temperature sensor output, it cannot be selected together with any other channel. Any channels from AN000 to AN007, AN016 to AN020, AN027, temperature sensor output, and internal reference voltage are selectable for window B.
The setting procedure is as follows when this function is to be used. The setting procedure required for normal A/D conversion in single scan mode is omitted.

(1) Confirm that the value of the ADCSR.ADCS[1:0] bits is 00b (single scan mode).
(2) Select channels (from among AN000 to AN007, AN016 to AN020, AN027, temperature sensor, and internal reference voltage) in the ADCMPANSR0, ADCMPANSR1, or ADCMPANSER register (for window A) and in the ADCMPBNSR register (for window B).
(3) Set window comparison conditions in the ADCMPLR0, ADCMPLR1, ADCMPLER, and ADCMPBNSR registers, and set the upper-limit and lower-limit reference values in the ADCMPDR0, ADCMPDR1, ADWINULB, and ADWINLLB registers.
(4) Set composite conditions for window A/B, window A/B operation enable, and interrupt output enable in the ADCMPCR register. A scan end event (S12ADELC) is output to the ELC at the end of each single scan. In addition, a match or mismatch event (S12ADWMELC or S12ADWUMELC) is output with a delay of one PCLKB cycle depending on the ADCMPCR.CMPAB[1:0] setting.
Since match and mismatch events are mutually exclusive, these are not output at the same time.

Downloaded from Arrow.com.



Note 1. ↓ indicates the instruction is executed by software.
Note 2. Data during conversion is ignored.

Figure 44.20 Example of Window Compare Function Operation (AN000 to AN003 Compared)

Downloaded from Arrow.com.

44.3.5.3 Using Data Buffers

This S12ADE is provided with a ring buffer function consisting of 16 A/D data buffers. This function sequentially stores A/D conversion results other than self-diagnosis result (including addition/average results) in data buffers (ADBUFn, n = 0 to 15) when the compare function is used.

Each conversion result is stored at the timing when the A/D conversion result is stored in the data register, and most recent 16 conversion result data are retained.

The following shows the schematic of data buffers, pointer, and overflow flag operations. When the BUFEN bit is set to1, the A/D conversion result is transferred at each end of A/D conversion. The pointer indicates the number of data buffer to which the next transferred data is to be written. When data is written to up to buffer 15, the pointer is reset to 0000b and the overflow flag is set to 1. Subsequently transferred data overwrites the previously written data. The pointer and overflow flag are reset to the initial value by writing 00h to the ADBUFPTR register.

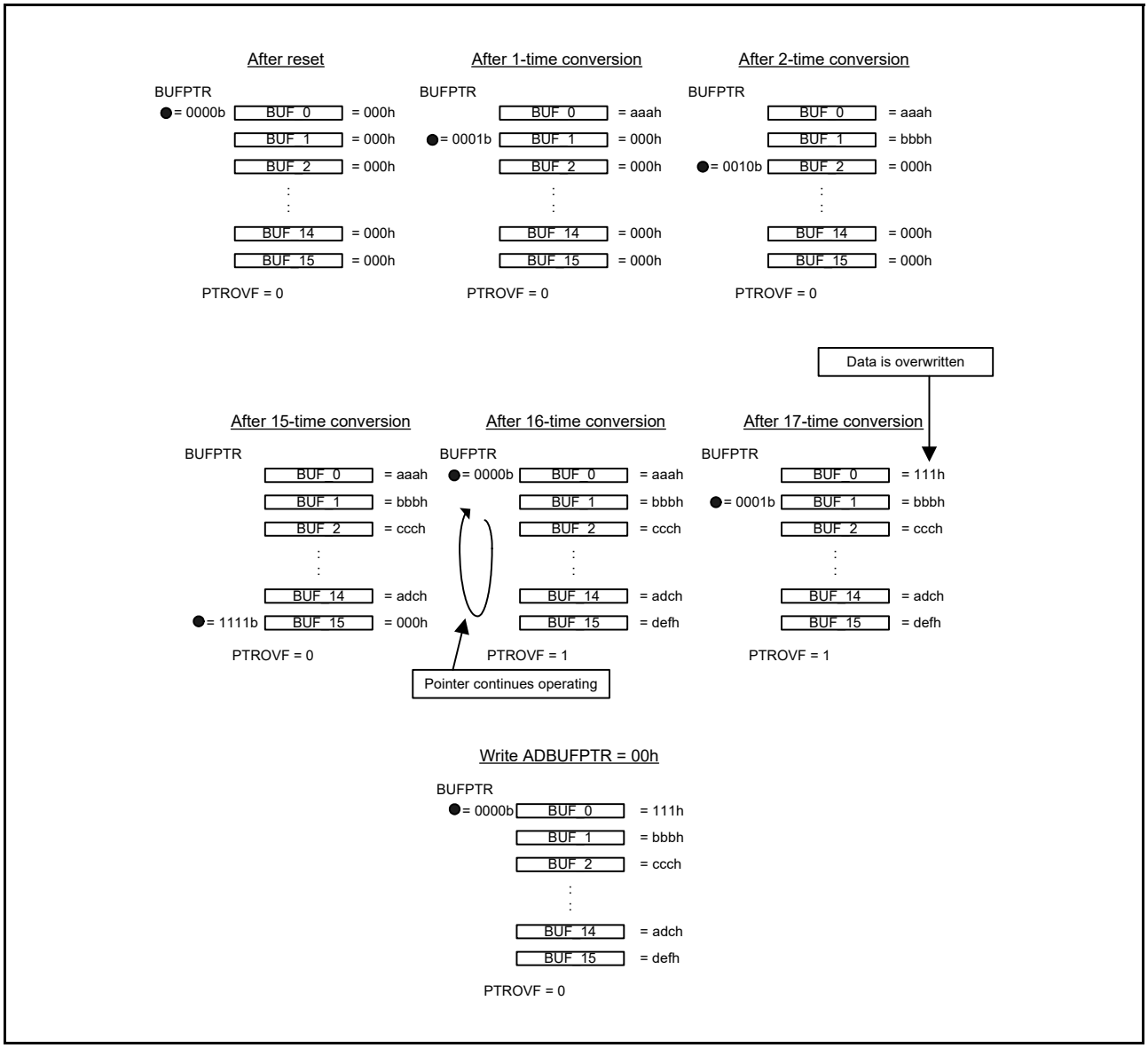


Figure 44.21 Schematic of Data Buffers, Pointer, and Overflow Flag Operations

Downloaded from Arrow.com.

44.3.5.4 Restrictions for Compare Function

The following restrictions are provided for the compare function.

1. The compare function must not be used together the self-diagnosis function or double trigger mode. (The compare function is not available for the ADRD register and the ADDBLDR register.)
2. Specify single scan mode when using match/mismatch event outputs.
3. When temperature sensor or internal reference voltage is selected for window A, window B operations are disabled.
4. When temperature sensor or internal reference voltage is selected for window B, window A operations are disabled.
5. It is prohibited to set the same channel for window A and window B.
6. When using the buffer function, specify single scan mode. (It is also prohibited to use double trigger mode together.)
7. Set the reference voltage values so that the high-side reference voltage value is equal to or larger than the low-side reference voltage value.

44.3.6 Analog Input Sampling Time and Scan Conversion Time

Scan conversion can be activated either by software, synchronous trigger, or asynchronous trigger input. After the start-of-scanning-delay time (t_D) has elapsed, processing for disconnection detection assistance and processing of conversion for self-diagnosis proceed, and this is followed by processing for A/D conversion.
Figure 44.22 shows the scan conversion timing in single scan mode, in which scan conversion is activated by software or a synchronous trigger. Figure 44.23 shows the scan conversion timing in single scan mode, in which scan conversion is activated by an asynchronous trigger. The scan conversion time (t_{SCAN}) includes the start-of-scanning-delay time (t_D), disconnection detection assistance processing time (t_{DIS})*1, self-diagnosis A/D conversion processing time (t_{DIAG})*2, A/D conversion processing time (t_{CONV}), and end-of-scanning-delay time (t_{ED}).
The A/D conversion processing time (t_{CONV}) consists of sampling time (t_{SPL}) and time for conversion by successive approximation (t_{SAM}). The sampling time (t_{SPL}) is used to charge sample-and-hold circuits in the A/D converter. If there is not sufficient sampling time due to the high impedance of an analog input signal source, or if the A/D conversion clock (ADCLK) is slow, sampling time can be adjusted using the ADSSTRn register.
The time for conversion by successive approximation (t_{SAM}) is at 32 ADCLK states during high-speed conversion operation, and 41 ADCLK states during low-current conversion operation. Table 44.10 shows the scan conversion time.

The scan conversion time (t_{SCAN}) in single scan mode for which the number of selected channels is n can be determined as follows:

$$t_{SCAN} = t_D + (t_{DIS} \times n) + t_{DIAG} + (t_{CONV} \times n)^{*3} + t_{ED}$$

The scan conversion time for the first cycle in continuous scan mode is t_{SCAN} for single scan minus t_{ED}.
The scan conversion time for the second and subsequent cycles in continuous scan mode is fixed to $(t_{DIS} \times n) + t_{DIAG} + t_{DSD} + (t_{CONV} \times n)$.

Note 1. When disconnection detection assistance is not selected, $t_{DIS} = 0$. The auto-discharge period of 15 ADCLK states is inserted only when the temperature sensor or internal reference voltage is A/D-converted.
Note 2. When the self-diagnosis function is not used, $t_{DIAG} = 0$, $t_{DSD} = 0$.
Note 3. $t_{CONV} \times n$ when the sampling time (t_{SPL}) of selected channels is the same, but it is the total of the sampling time of each channel and time for conversion by successive approximation (t_{SAM}).

Downloaded from Arrow.com.

Table 44.10 Times for Conversion during Scanning (in Numbers of Cycles of ADCLK and PCLKB)

Item			Symbol		Type/Conditions: Synchronous Trigger *5	Type/Conditions: Asynchronous Trigger	Type/Conditions: Software Trigger	Unit
Scan start processing time*1, *2	A/D conversion on group A under group-A priority control.	Group B is to be stopped. (Group A is activated after group B is stopped due to an A/D conversion source of group A.)	t_D		3 PCLKB + 6 ADCLK	—	—	Cycle
		Group B is not to be stopped. (Activation by an A/D conversion source of group A.)			2 PCLKB + 4 ADCLK	—	—	
	A/D conversion when self-diagnosis is enabled	A/D conversion for self-diagnosis is to be started.			2 PCLKB + 6 ADCLK	4 PCLKB + 6 ADCLK	6 ADCLK	
	Other than above				2 PCLKB + 4 ADCLK	4 PCLKB + 4 ADCLK	4 ADCLK	
Disconnection detection assistance processing time			t_{DIS}		The setting of ADNDIS[3:0] (initial value = 00h) × ADCLK*3			
Self-diagnosis conversion processing time*1	Sampling time		t_{DIAG}	t_{SPL}	The setting of ADSSTR0 (initial value = 0Dh) × ADCLK*4			
	Time for conversion by successive approximation	12-bit conversion accuracy		t_{SAM}	32 ADCLK (during high-speed conversion operation)			
					41 ADCLK (during low-current conversion operation)			
	Normal A/D conversion is to be started after completion of self-diagnosis conversion.			t_{DED}	2 ADCLK			
	A/D conversion for self-diagnosis is to be started after completion of conversion for continuous scan on the last channel specified.			t_{DSD}	2 ADCLK			
A/D conversion processing time*1	Sampling time		t_{CONV}	t_{SPL}	The setting of ADSSTRn (n = 0 to 7, L, T, O) (initial value = 0Dh) × ADCLK*4			
	Time for conversion by successive approximation	12-bit conversion accuracy		t_{SAM}	32 ADCLK (during high-speed conversion operation)			
					41 ADCLK (during low-current conversion operation)			
Scan end processing time*1			t_{ED}		1 PCLKB + 3 ADCLK*6			

Note 1. For t_D, t_{DIAG}, t_{CONV}, and t_{ED}, see Figure 44.22 and Figure 44.23.
Note 2. This is the maximum time required from software writing or trigger input to A/D conversion start.
Note 3. The value is fixed to 0Fh (15 ADCLK) when the temperature sensor output or internal reference voltage is A/D-converted.
Note 4. The required sampling time (ns) is specified according to the voltage conditions. See section 51.5, A/D Conversion Characteristics.
Note 5. This does not include the time consumed in the path from timer output to trigger input.
Note 6. 2 PCLKB + 2 ADCLK when ADCLK is faster than PCLKB (PCLKB to ADCLK frequency ratio = 1:2).

Downloaded from Arrow.com.

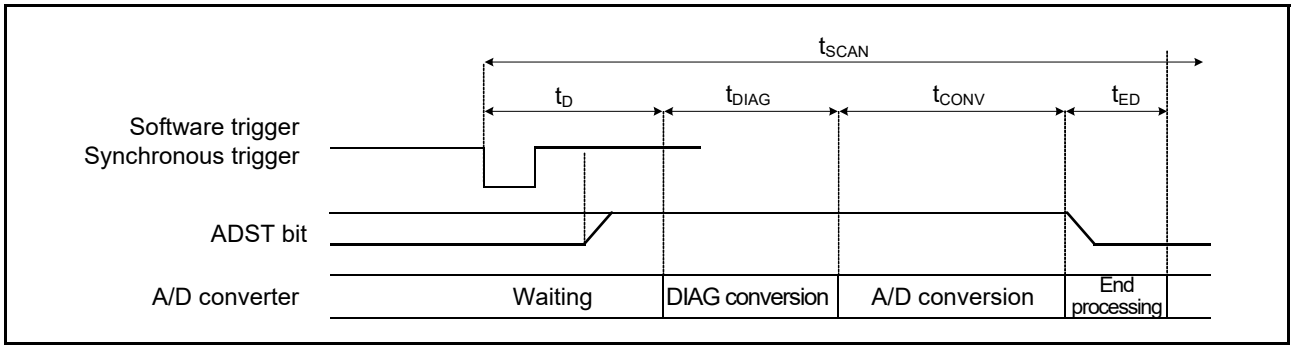


Figure 44.22 Scan Conversion Timing (Activated by Software or Synchronous Trigger)

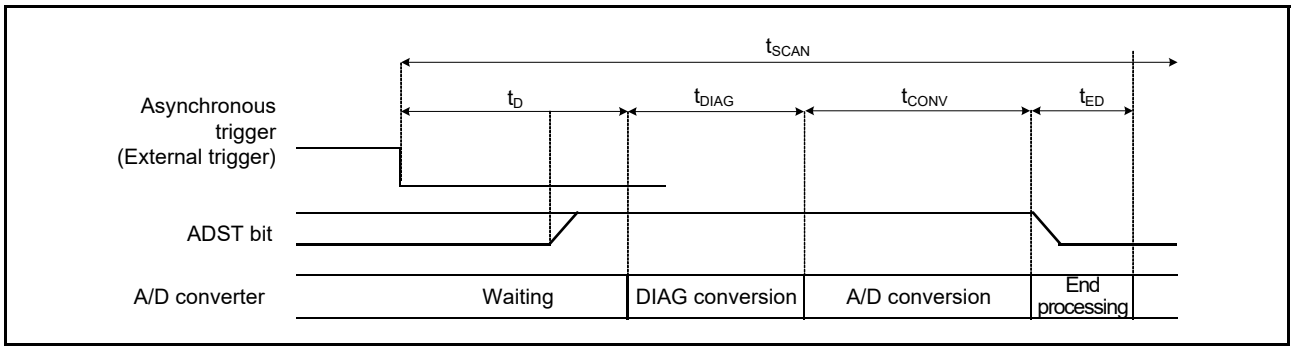


Figure 44.23 Scan Conversion Timing (Activated by Asynchronous Trigger)

44.3.7 Usage Example of A/D Data Register Automatic Clearing Function

Setting the ADCER.ACE bit to 1 automatically clears the A/D data registers (ADDRy, ADRD, ADTSDR, ADOCDR, ADDBLDR) to 0000h when the A/D data registers are read by the CPU, DTC, or DMACA.

The ring buffer (ADBUFn: n = 0 to 15) is not subject to auto-clearing.

This function enables detection of update failures of the A/D data registers (ADDRy, ADRD, ADTSDR, ADOCDR, ADDBLDR). The following describes the examples in which the function to automatically clear the ADDRy register is enabled and disabled.

In a case where the ADCER.ACE bit is 0 (automatic clearing disabled), if the A/D conversion result (0222h) is not written to the ADDRy register for some reason, the old data (0111h) will be the ADDRy value. Furthermore, if this ADDRy value is read into a general register using an A/D scan end interrupt, the old data (0111h) can be saved in the general register. When checking whether there is an update failure, it is necessary to frequently save the old data in the RAM or a general register.

In a case where the ADCER.ACE bit is 1 (automatic clearing enabled), when ADDRy = 0111h is read by the CPU, DTC, or DMACA, ADDRy is automatically cleared to 0000h. After that, if the A/D conversion result 0222h cannot be transferred to ADDRy for some reason, the cleared data (0000h) remains as the ADDRy value. If this ADDRy value is read into a general register using an A/D scan end interrupt at this point, 0000h will be saved in the general register. Occurrence of an ADDRy update failure can be determined by simply checking that the read data value is 0000h.

Downloaded from Arrow.com.

44.3.8 A/D-Converted Value Addition/Average Mode

In A/D-converted value addition mode, the same channel is A/D-converted 2, 3, 4, or 16 consecutive times and the sum of the converted values is stored in the data register. In A/D-converted value average mode, the same channel is A/D-converted two or four consecutive times and the mean of the converted values is stored in the data register. The use of the average of these results can improve the accuracy of A/D conversion, depending on the types of noise components that are present. This function, however, cannot always guarantee an improvement in A/D conversion accuracy.
The A/D-converted value addition or average mode can be specified when A/D conversion of the channel select analog input, temperature sensor output, or internal reference voltage is selected.

44.3.9 Disconnection Detection Assist Function

This converter incorporates the function to fix the charge for sampling capacitance to the specified state (reference voltage selected by the A/D high-potential/low-potential reference voltage control register) before start of A/D conversion. This function enables disconnection detection in wiring of analog inputs.
Figure 44.24 illustrates the A/D conversion operation when the disconnection detection assist function is used. Figure 44.25 shows an example of disconnection detection when precharge is selected. Figure 44.26 shows an example of disconnection detection when discharge is selected.

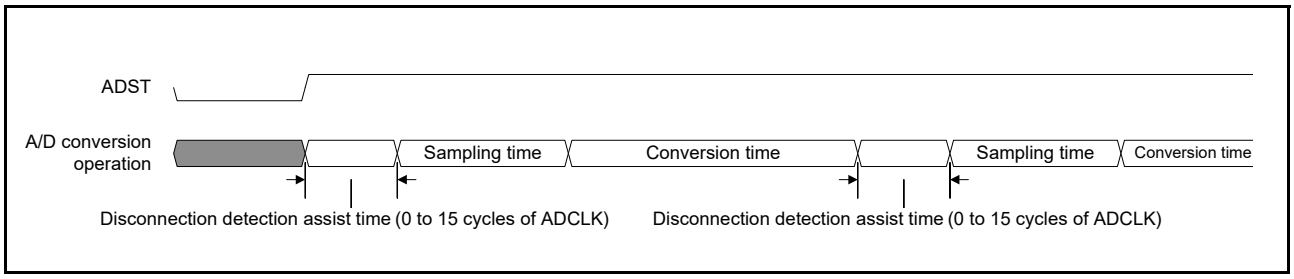
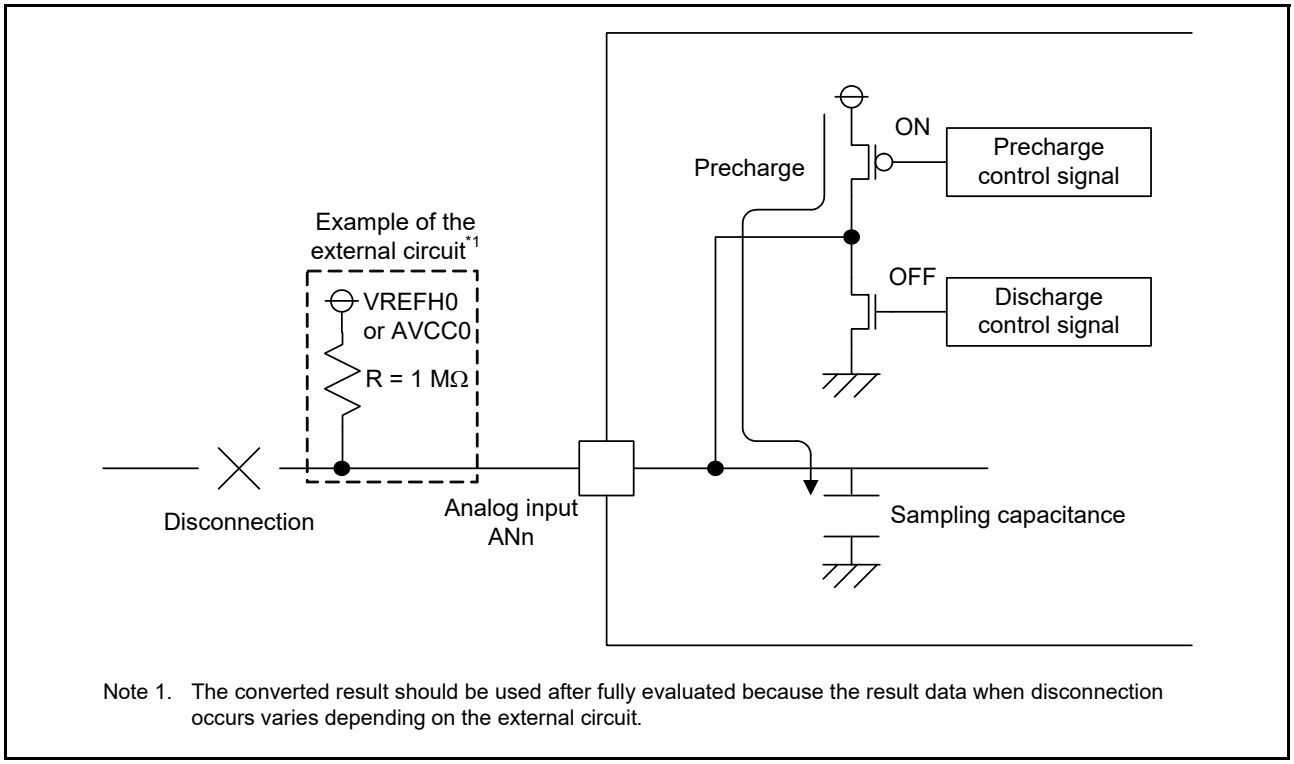


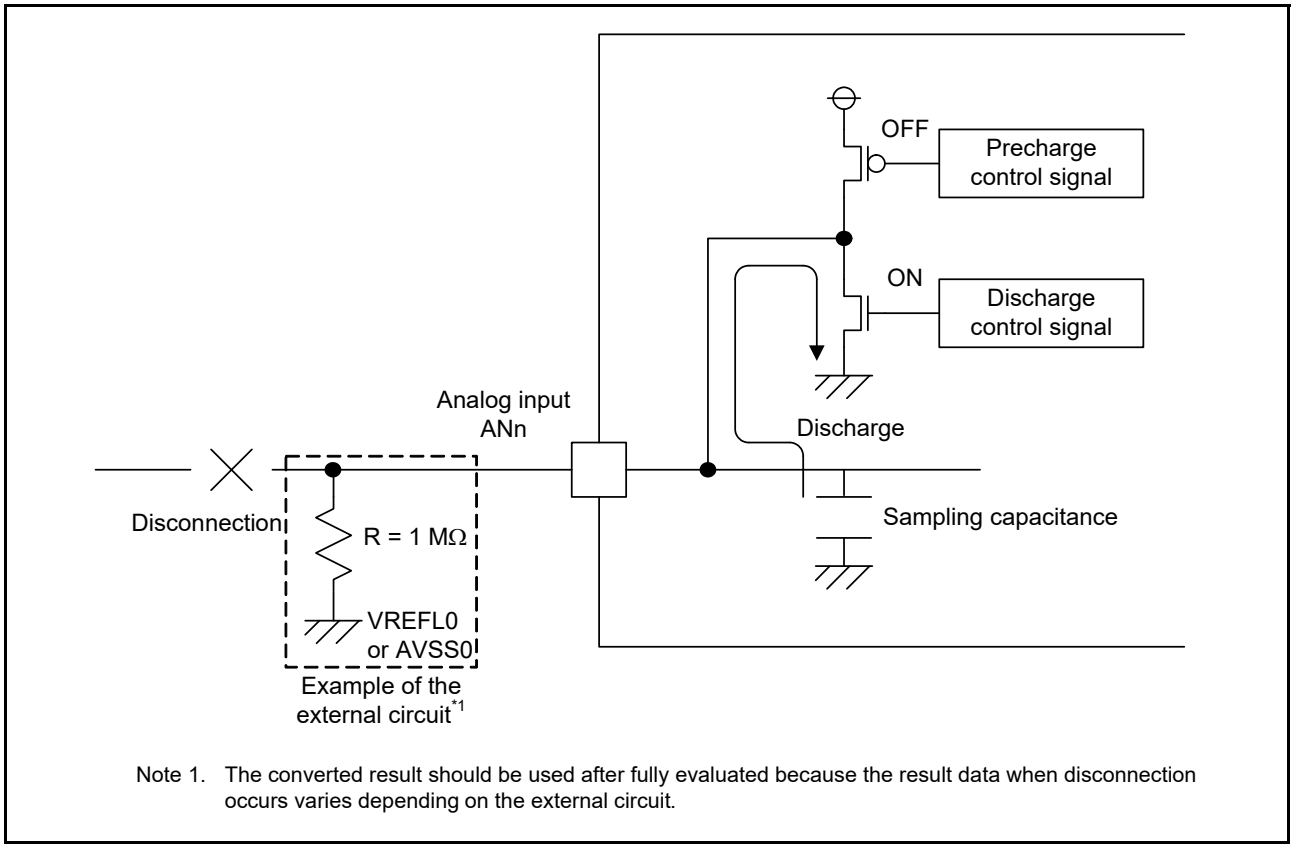
Figure 44.24 Operation of A/D Conversion When the Disconnection Detection Assist Function is Used



Note 1. The converted result should be used after fully evaluated because the result data when disconnection occurs varies depending on the external circuit.

Figure 44.25 Example of Disconnection Detection When Precharge is Selected

Downloaded from Arrow.com.



Note 1. The converted result should be used after fully evaluated because the result data when disconnection occurs varies depending on the external circuit.

Figure 44.26 Example of Disconnection Detection When Discharge is Selected

Downloaded from Arrow.com.

44.3.10 Starting A/D Conversion with Asynchronous Trigger

The A/D conversion can be started by the input of an asynchronous trigger. To start up the A/D converter by an asynchronous trigger, the A/D conversion start trigger select bits (ADSTRGR.TRSA[5:0]) should be set to 000000b, and a high-level signal should be input to the asynchronous trigger (ADTRG0# pin). Then, the ADCSR.TRGE and ADCSR.EXTRG bits should be set to 1. Figure 44.27 shows a timing of the asynchronous trigger input.
For the time from when the ADST bit is set to 1 until conversion starts, refer to section 44.8.3, A/D Conversion Restarting Timing and Termination Timing. An asynchronous trigger cannot be selected for group B used in group scan mode.

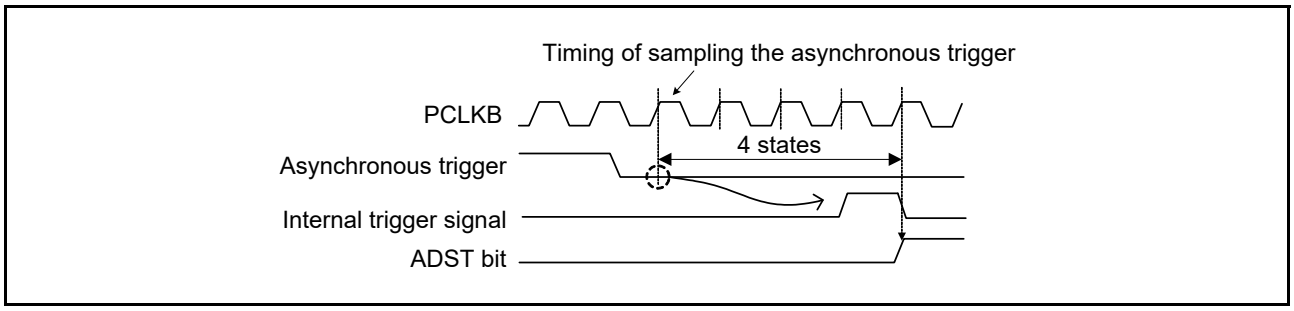


Figure 44.27 Timing of Sampling Asynchronous Trigger

44.3.11 Starting A/D Conversion with Synchronous Trigger from Peripheral Module

The A/D conversion can be started by a synchronous trigger. To start the A/D conversion by a synchronous trigger, the ADCSR.TRGE bit should be set to 1, the ADCSR.EXTRG bit should be cleared to 0, and the relevant sources should be selected by the ADSTRGR.TRSA[5:0] and ADSTRGR.TRSB[5:0] bits.

44.4 Interrupt Sources and DTC/DMAC Transfer Requests

44.4.1 Interrupt Requests

The 12-bit A/D converter can send scan end interrupt requests S12ADI0 and GBADI to the CPU.
Setting the ADCSR.ADIE bit to 1 and 0 enables and disables an S12ADI0 interrupt, respectively; similarly, setting the ADCSR.GBADIE bit to 1 and 0 enables and disables a GBADI interrupt, respectively.
In addition, the DTC or DMACA can be activated when an S12ADI0 or a GBADI interrupt is generated. Using an S12ADI0 or a GBADI interrupt to allow the DTC or DMACA to read the converted data enables continuous conversion without burden on software.
For details on DTC settings, see section 19, Data Transfer Controller (DTCa), and for details on DMACA settings, see section 18, DMA Controller (DMACA).

Downloaded from Arrow.com.

44.5 Event Link Function

44.5.1 Event Output to the ELC

The ELC uses the S12ADI0 interrupt request signal as an event signal (S12ADELC), enabling link operation for the preset module. An event signal is generated under the conditions set by the event link control bits (ADELCCR.ELCC[1:0] bits).
The event signal can be output regardless of the setting of the corresponding interrupt request enable bit.
The 12-bit A/D converter outputs the A/D conversion end event (S12ADELC), window function compare match event (S12ADWMELC), and mismatch event (S12ADWUMELC).
The scan end event (S12ADELC) is output to the ELC at the same time as the interrupt output (S12ADI0) regardless of the ADCSR.ADIE setting.
The compare match/mismatch event (S12ADWMELC/S12ADWUMELC) is output to the ELC with a delay of one PCLKB cycle from the interrupt output (S12ADI0) regardless of the ADCSR.ADIE setting.
When using compare match/mismatch events (S12ADWMELC/S12ADWUMELC) to the ELC, specify single scan mode.

44.5.2 12-Bit A/D Converter Operation by Event from the ELC

The 12-bit A/D converter can be started by the predetermined event by setting the ELSRn register of the ELC.

44.5.3 Note on 12-Bit A/D Converter When an Event Is Input from the ELC

If an event occurs during A/D conversion, the event is disabled.

44.6 Selecting Reference Voltage

For the A/D converter, the high-potential reference voltage can be selected from VREFH0 and AVCC0, and the low-potential reference voltage can be selected from VREFL0 and AVSS0, respectively. Set these before starting A/D conversion. For details of this setting, refer to section 44.2.30, A/D High-Potential/Low-Potential Reference Voltage Control Register (ADHVREFCNT).

Downloaded from Arrow.com.

44.7 Allowable Impedance of Signal Source

Figure 44.28 shows an equivalent circuit of an analog input pin and an external sensor.
To perform A/D conversion accurately, the internal capacitor (Cs) must be fully charged within the sampling time. If the impedance (R0) of the signal source is high and it takes time to charge Cs, extend the sampling time with the ADSSTRn register. Conversely, if R0 is small, the sampling time can be shortened. Refer to the electrical characteristics for the permissible signal source impedance under various operating conditions.
When converting only a single pin input in single scan mode, the influence of R0 can be ignored because the input load becomes practically only the internal input resistor (Rs) by connecting an external high-capacity capacitor (C). However, because a low-pass filter is formed by R0 and C, it may not possible to follow the analog signal that changes at high speed. Insert a low-impedance buffer when converting high speed analog signals or when converting multiple pins in scan mode.

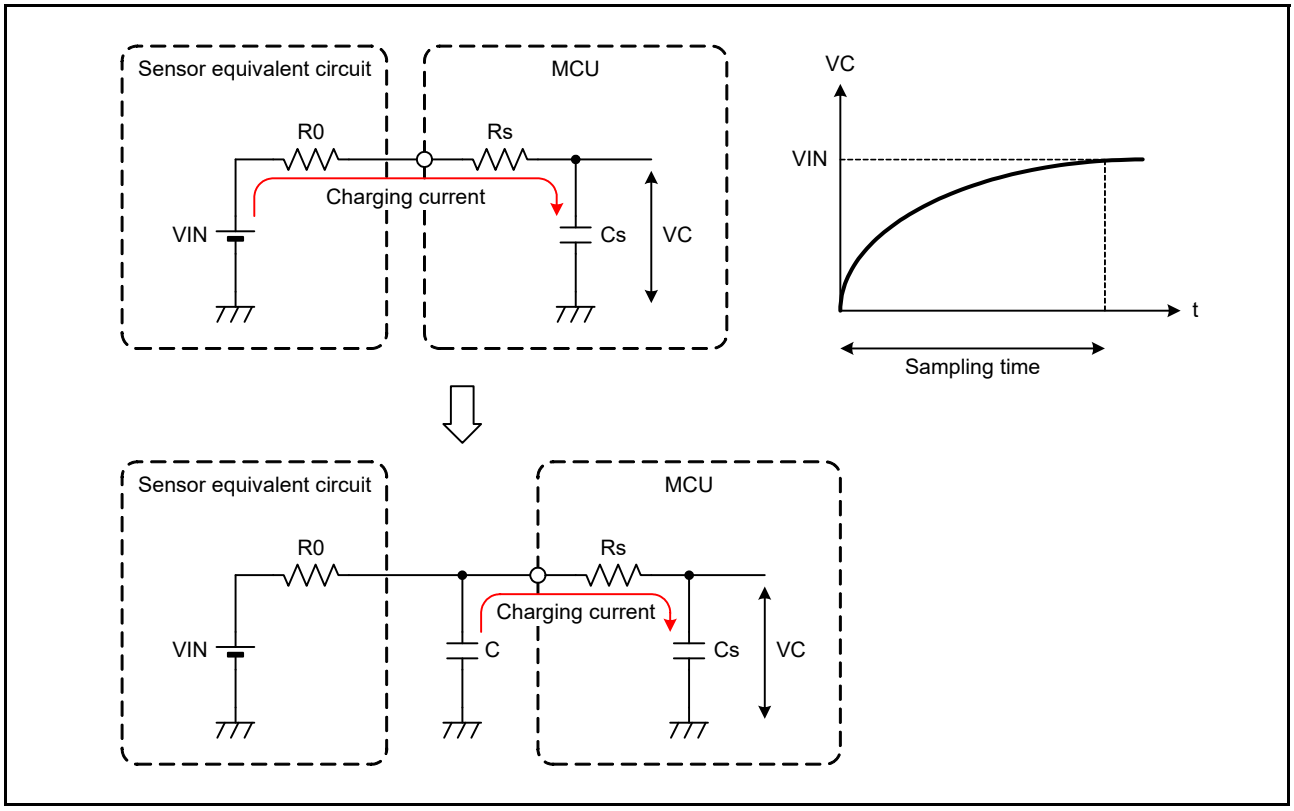


Figure 44.28 Equivalent Circuit of Analog Input Pin and External Sensor

Downloaded from Arrow.com.

44.8 Usage Notes

44.8.1 Notes on Reading Data Registers

Read the A/D data registers, A/D data duplication register, A/D data duplication register A, A/D data duplication register B, A/D temperature sensor data register, A/D internal reference voltage data register, and A/D self-diagnosis data register in word units. If a register is read twice in byte units, that is, the higher-order byte and lower-order byte are separately read, the A/D-converted value having been read first may disagree with the A/D-converted value having been read for the second time. To prevent this, the data registers should never be read in byte units.

44.8.2 Notes on Stopping A/D Conversion

To stop A/D conversion when an asynchronous trigger or a synchronous trigger has been selected as the condition for starting A/D conversion, follow the procedure in Figure 44.29.

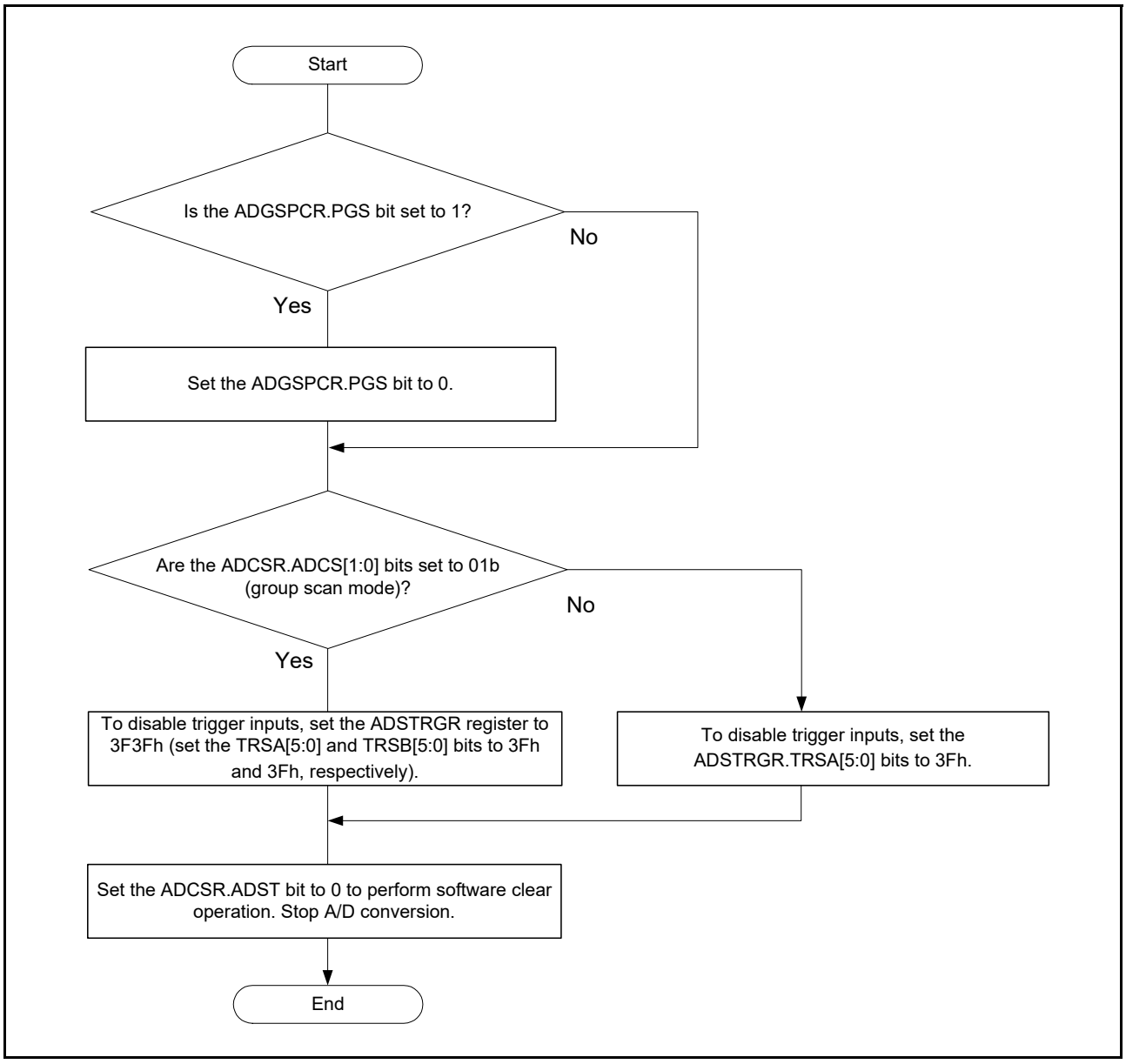


Figure 44.29 Procedure for Clear Operation by Software through the ADCSR.ADST Bit

Downloaded from Arrow.com.

44.8.3 A/D Conversion Restarting Timing and Termination Timing

It takes a maximum of six ADCLK cycles for the idle analog unit of the 12-bit A/D converter to be restarted by setting the ADCSR.ADST bit to 1. It takes a maximum of three ADCLK cycles (1 PCLKB + 2 ADCLK when ADCLK is faster than PCLKB (PCLKB to ADCLK frequency ratio = 1:2)) for the operating analog unit of the 12-bit A/D converter to be terminated by setting the ADCSR.ADST bit to 0.

44.8.4 Notes on Scan End Interrupt Handling

When scanning the same analog input twice using any trigger, the first A/D-converted data is overwritten with the second A/D-converted data in the case that the CPU does not complete reading the A/D-converted data by the time the A/D conversion of the first analog input for the second scan ends after the first scan end interrupt is generated.

44.8.5 Module Stop Function Setting

Operation of the 12-bit A/D converter can be disabled or enabled by setting module stop control register A (MSTPCRA). The initial setting is for operation of the 12-bit A/D converter to be halted. Register access is enabled by releasing the module stop state.
After the module stop state is released, wait for 1 µs to start A/D conversion. For details, refer to section 11, Low Power Consumption.

44.8.6 Notes on Entering Low Power Consumption States

Before entering the module stop state or software standby mode, make sure to stop A/D conversion. Here, set the ADCSR.ADST bit to 0, and secure certain period of time until the analog unit of the 12-bit A/D converter is stopped. Follow the procedure given below to secure this time.
Follow the procedure for clear operation by software through the ADCSR.ADST bit, shown in Figure 44.29. After that, wait for two clock cycles of ADCLK before entering the peripheral module stop state or software standby mode.

44.8.7 Notes on Canceling Software Standby Mode

After software standby mode is canceled, wait until the crystal oscillation stabilization time or the PLL circuit stabilization time elapses, and then wait for 1 µs before starting A/D conversion. For details, refer to section 11, Low Power Consumption.

44.8.8 Pin Setting when the 12-bit A/D Converter is Used

When the 12-bit A/D converter is used, do not set any pin of port 4 as output.
Output from any of the pins may affect on A/D conversion accuracy because analog power supply is used in the part of the port 4 circuit.

44.8.9 Error in Absolute Accuracy When Disconnection Detection Assistance is in Use

Using disconnection detection assistance leads to an error in absolute accuracy of the A/D converter. This is because an error voltage is input to the analog input pins due to the resistive voltage division between the pull-up or pull-down resistor (Rp) and the resistance of the signal source (Rs). This error in absolute accuracy is calculated from the following formula. Only use disconnection detection assistance after thorough evaluation.

Maximum error in absolute accuracy (LSB) = 4095 × Rs / Rp

Downloaded from Arrow.com.

44.8.10 ADHSC Bit Rewriting Procedure

Before rewriting the A/D conversion select bit (ADCSR.ADHSC) from 0 to 1 or from 1 to 0, the 12-bit A/D converter must be in the standby state. Carry out steps 1 to 3 below to modify the ADCSR.ADHSC bit. After the sleep bit (ADHVREFCNT.ADSLP) is cleared to 0, wait for at least 1 µs and then start A/D conversion.

ADHSC Bit Rewriting Procedure:

1. Set the sleep bit (ADHVREFCNT.ADSLP) to 1.
2. Wait for at least 0.2 µs, and then modify the A/D conversion select bit (ADCSR.ADHSC).
3. Wait for at least 4.8 µs, and then clear the sleep bit (ADHVREFCNT.ADSLP) to 0.

Note: It is prohibited to set the ADHVREFCNT.ADSLP bit to 1 except for modifying the A/D conversion select bit (ADCSR.ADHSC).

Downloaded from Arrow.com.

44.8.11 Voltage Range of Analog Power Supply Pins

If this MCU is used with the voltages outside the following ranges, the reliability of the MCU may be affected.

- Analog input voltage range
  Voltage applied to analog input pins ANn: AVSS0 ≤ VAN ≤ AVCC0
  Reference voltage range applied to pins VREFH0 and VREFL0: VREFH0 ≤ AVCC0, VREFL0 = AVSS0
  Conversion will not succeed if the voltage applied to analog input pins ANn is greater than VREFH0 (see Figure 44.30).

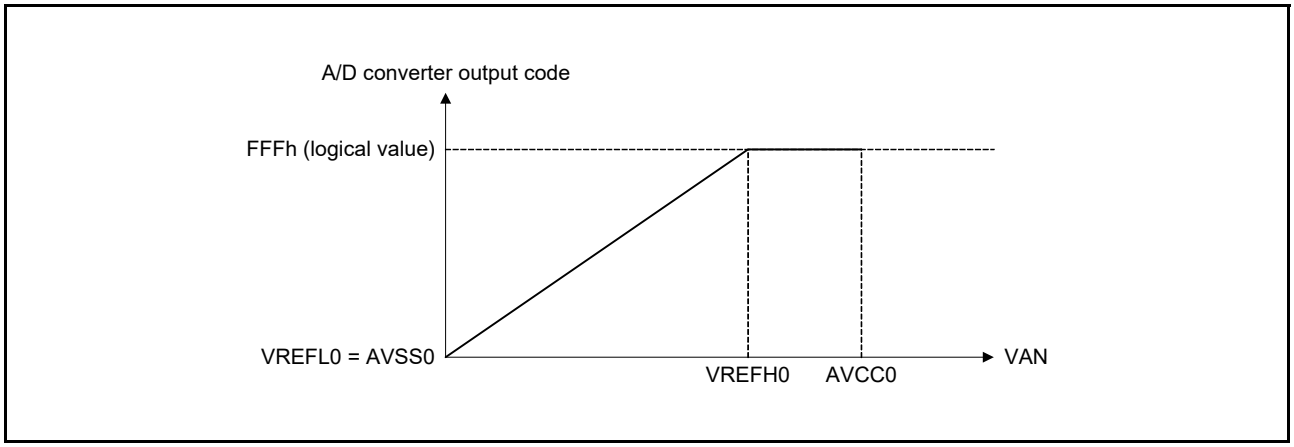


Figure 44.30 Relationship Between Voltage Applied to Analog Input Pins and Output Code

- Relationship between power supply pin pairs (AVCC0–AVSS0, VREFH0–VREFL0, VCC–VSS)
  The following condition should be satisfied: AVSS0 = VSS. When performing A/D conversion of analog input pins AN016 to AN020 and AN027, the following condition should be satisfied: AVCC0 = VCC. A 0.1-µF capacitor should be connected between each pair of power supply pins to create a closed loop with the shortest route possible as shown in Figure 44.31, and connection should be made so that the following conditions are satisfied at the supply side.
  VREFL0 = AVSS0 = VSS
  When the 12-bit A/D converter is not used, the following conditions should be satisfied.
  VREFH0 = AVCC0 = VCC and VREFL0 = AVSS0 = VSS

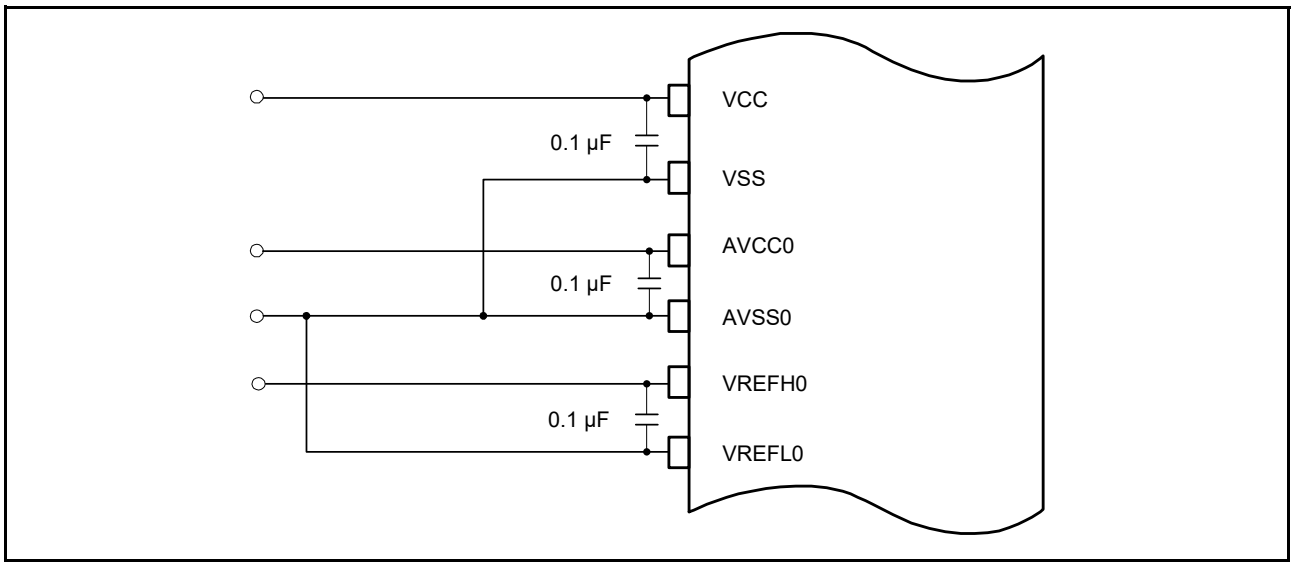


Figure 44.31 Power Supply Pin Connection Example

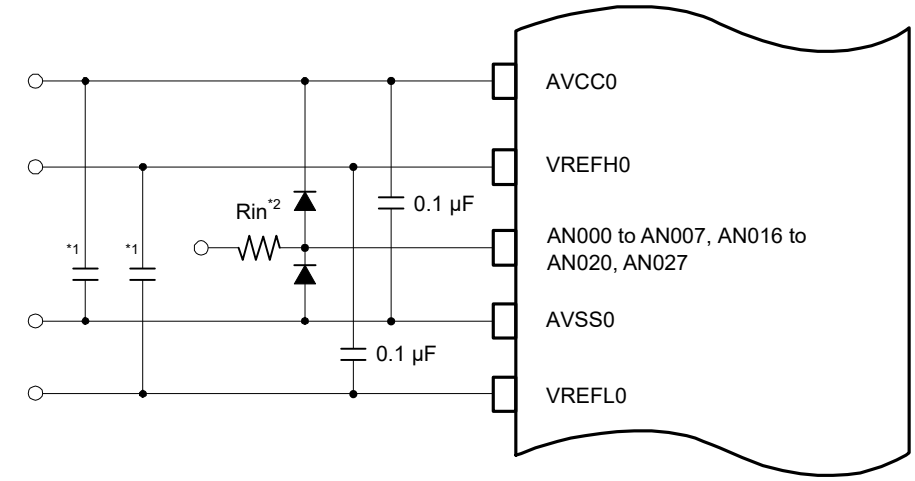
Downloaded from Arrow.com.

44.8.12 Notes on Board Design

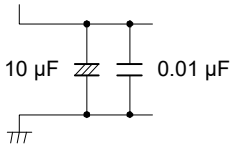
The board should be designed so that digital circuits and analog circuits are separated from each other as far as possible. In addition, digital circuit signal lines and analog circuit signal lines should not intersect or placed near each other. If these rules are not followed, noise will be produced on analog signals and A/D conversion accuracy will be affected. The analog input pins (AN000 to AN007, AN016 to AN020, AN027), reference power supply pin (VREFH0), reference ground pin (VREFL0), and analog power supply (AVCC0) should be separated from digital circuits using the analog ground (AVSS0). The analog ground (AVSS0) should be connected to a stable digital ground (VSS) on the board (single-point ground plane connection).

44.8.13 Notes on Noise Prevention

To prevent the analog input pins (AN000 to AN007, AN016 to AN020, AN027) from being destroyed by abnormal voltage such as excessive surge, a capacitor should be inserted between AVCC0 and AVSS0 and between VREFH0 and VREFL0, and a protection circuit should be connected to protect the analog input pins (AN000 to AN007, AN016 to AN020, AN027) as shown Figure 44.32.



Note 1. The values shown here are reference values.



Note 2. Rin: Signal source impedance

Figure 44.32 Sample Protection Circuit for Analog Inputs

Downloaded from Arrow.com.

45. 12-Bit D/A Converter (R12DAA)

45.1 Overview

This MCU includes two channels of 12-bit D/A converter.
Table 45.1 lists the specifications of the 12-bit D/A converter and Figure 45.1 shows a block diagram of the 12-bit D/A converter.

Table 45.1 Specifications of 12-Bit D/A Converter

Item	Specifications
Resolution	12 bits
Output channels	Two channels
Measure against mutual interference between analog modules	Measure against interference between D/A and A/D conversion D/A converted data update timing is controlled by the 12-bit A/D converter synchronous D/A conversion enable signal from the 12-bit A/D converter. Therefore, the degradation of A/D conversion accuracy due to interference is reduced by controlling the timing in which the 12-bit D/A converter inrush current occurs, with the enable signal.
Low power consumption function	Module stop state can be set.
Event link function (input)	DA0 conversion can be started when an event signal is input.

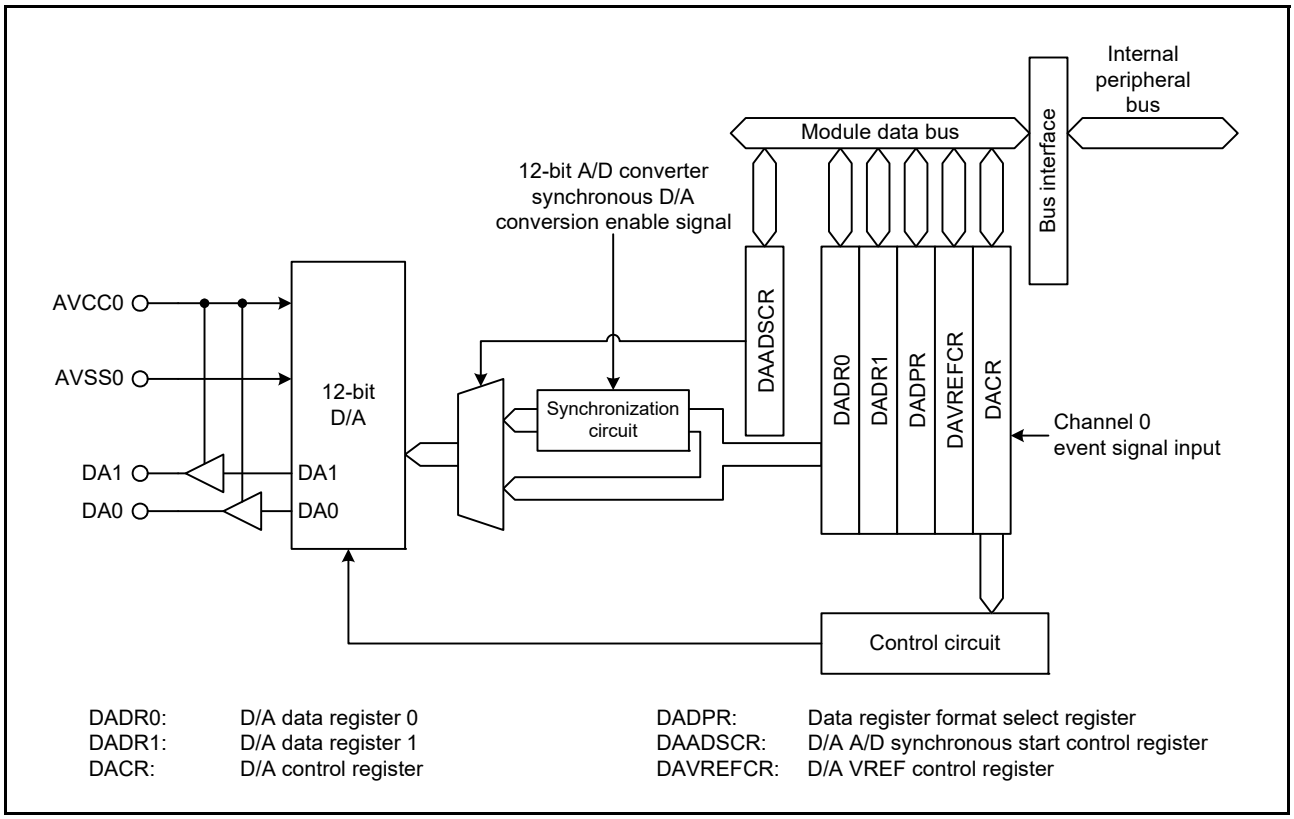


Figure 45.1 Block Diagram of 12-Bit D/A Converter

Downloaded from Arrow.com.

Table 45.2 lists the pin configuration of the 12-bit D/A converter.

Table 45.2 Pin Configuration of 12-Bit D/A Converter

Pin Name	I/O	Function
AVCC0	Input	Analog power supply pin
AVSS0	Input	Analog ground pin
DA0	Output	Channel 0 analog output pin
DA1	Output	Channel 1 analog output pin

45.2 Register Descriptions

45.2.1 D/A Data Register m (DADRm) (m = 0, 1)

Address(es): DA.DADR0 0008 8040h, DA.DADR1 0008 8042h

• DADPR.DPSEL bit = 0 (data is right-justified)

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

• DADPR.DPSEL bit = 1 (data is left-justified)

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
													—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The DADRm register is a 16-bit readable/writable register, which stores data to which D/A conversion is to be performed. Whenever an analog output is enabled, the values in DADRm are converted and output from the D/A converter.

12-bit data can be relocated by setting the DADPR.DPSEL bit.

Bits "—" are read as 0. The write value should be 0.

Downloaded from Arrow.com.

45.2.2 D/A Control Register (DACR)

Address(es): DA.DACR 0008 8044h

	b7	b6	b5	b4	b3	b2	b1	b0
	DAOE1	DAOE0	—	—	—	—	—	—
Value after reset:	0	0	0	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R
b6	DAOE0	D/A Output Enable 0	0: Analog output of channel 0 (DA0) is disabled. 1: D/A conversion of channel 0 is enabled. Analog output of channel 0 (DA0) is enabled.	R/W
b7	DAOE1	D/A Output Enable 1	0: Analog output of channel 1 (DA1) is disabled. 1: D/A conversion of channel 1 is enabled. Analog output of channel 1 (DA1) is enabled.	R/W

This register should be set when the DAADSCR.DAADST bit is 1 (measure against interference between D/A and A/D conversion is enabled) while the 12-bit A/D converter is halted (the ADCSR.ADST bit is 0). At that time, the software trigger should be selected for the 12-bit A/D converter trigger to securely stop the 12-bit A/D converter.

DAOE0 Bit (D/A Output Enable 0)

The DAOE0 bit controls the D/A conversion and analog output.

The event link function can be used to set the DAOE0 bit to 1. The DAOE0 bit becomes 1 when the event specified by setting the ELSR16 register of the ELC occurs, and output of the D/A conversion results starts.

DAOE1 Bit (D/A Output Enable 1)

The DAOE1 bit controls the D/A conversion and analog output.

45.2.3 Data Register Format Select Register (DADPR)

Address(es): DA.DADPR 0008 8045h

	b7	b6	b5	b4	b3	b2	b1	b0
	DPSEL	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	DPSEL	Format Select	0: Data is right-justified. 1: Data is left-justified.	R/W

Downloaded from Arrow.com.

45.2.4 D/A A/D Synchronous Start Control Register (DAADSCR)

Address(es): DA.DAADSCR 0008 8046h

	b7	b6	b5	b4	b3	b2	b1	b0
	DAADST	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	DAADST	D/A A/D Synchronous Conversion	0: 12-bit D/A converter operation does not synchronize with 12-bit A/D converter operation. (measure against interference between D/A and A/D conversion is disabled) 1: 12-bit D/A converter operation synchronizes with 12-bit A/D converter operation. (measure against interference between D/A and A/D conversion is enabled)	R/W

As a measure against interference between D/A and A/D conversion, the DAADSCR register selects whether or not the timing for starting 12-bit D/A conversion is synchronized with the 12-bit A/D converter synchronous D/A conversion enable signal from the 12-bit A/D converter.
This register should be set while the 12-bit A/D converter is halted (while the ADCSR.ADST bit is 0 after selecting software trigger as the 12-bit A/D converter trigger).

DAADST Bit (D/A A/D Synchronous Conversion)

Setting the DAADST bit to 0 allows the DADRm register value (m = 0, 1) to be converted into analog data at any time.
Setting the DAADST bit to 1 allows synchronous D/A conversion with the 12-bit A/D converter synchronous D/A conversion enable signal from the 12-bit A/D converter. Therefore, even if the DADRm register value is modified, D/A conversion does not start until the 12-bit A/D converter completes A/D conversion.
Set this bit while the ADCSR.ADST bit is set to 0. At this time, the software trigger should be selected for the 12-bit A/D converter trigger to securely stop the 12-bit A/D converter.
The event link function cannot be used when the DAADST bit is set to 1. Stop the event link function by setting the ELSR16 register of the ELC. The setting of the DAADST bit is common to channels 0 and 1 of the 12-bit D/A converter.

Downloaded from Arrow.com.

45.2.5 D/A VREF Control Register (DAVREFCR)

Address(es): DA.DAVREFCR 0008 8047h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	REF[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	REF[2:0]	D/A Reference Voltage Select	b2 b0 0 0 0: Not selected 0 0 1: AVCC0/AVSS0 0 1 1: Internal reference voltage/AVSS0 Settings other than above are prohibited.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The DAVREFCR register selects the reference voltage of the 12-bit D/A converter.

REF[2:0] Bits (D/A Reference Voltage Select)

The REF[2:0] bits select the reference voltage of the 12-bit D/A converter 0 and 1. When changing the value of these bits, write 000b to the DAVREFCR.REF[2:0] bits in advance. Read the REF[2:0] bits after changing their value, and confirm that it has been changed. When selecting the internal reference voltage, set the DADR0 and DADR1 registers to 0000h and discharge the VREF path before switching the voltage. As the path remains discharged after the reset is released, the internal reference voltage can be selected. For details on discharging, refer to section 45.3.2, Notes on Using the Internal Reference Voltage as the Reference Voltage. Do not rewrite this register during A/D conversion using the 12-bit A/D converter. If this register is rewritten, the accuracy of A/D conversion is not guaranteed.
When the internal reference voltage is selected, the voltage generation circuit operates and current increases by about 75 µA. This circuit is not automatically turned off even if the MCU enters software standby mode with the internal reference voltage selected.

Downloaded from Arrow.com.

45.3 Operation

The 12-bit D/A converter includes D/A conversion circuits for two channels, each of which can operate independently. When the DACR.DAOEm bit (m = 0, 1) is set to 1, D/A converter is enabled and the conversion result is output. An operation example of D/A conversion on channel 0 is shown below. Figure 45.2 shows the timing of this operation.

(1) Set the data for D/A conversion in the DADPR.DPSEL bit and the DADR0 register.

(2) Set the DACR.DAOE0 bit to 1 to start D/A conversion. The DA0 output settles to the voltage corresponding to the setting value after the conversion time tDCONV has elapsed. The DA0 output voltage is held at this level until the DADR0 register is updated or the DAOE0 bit is set to 0. The output voltage (reference) is expressed by the following formula:

$$\frac{\text{Value of DADRm register}}{4096} \times \text{Reference voltage}$$

Because the D/A converter output is buffered with an amplifier, the output voltage does not reach AVSS0 or AVCC0. Refer to section 51, Electrical Characteristics for the output voltage range.

(3) When the DADR0 register is updated, the conversion starts. The DA0 output settles at the new output voltage after the conversion time tDCONV has elapsed.
When the DAADSCR.DAADST bit is 1 (measure against interference between D/A and A/D conversion is enabled), it takes a maximum of one A/D conversion time for D/A conversion to start. When ADCLK is faster than the peripheral module clock, it may take longer than one A/D conversion time.

(4) When the DAOE0 bit is set to 0, analog output is disabled.

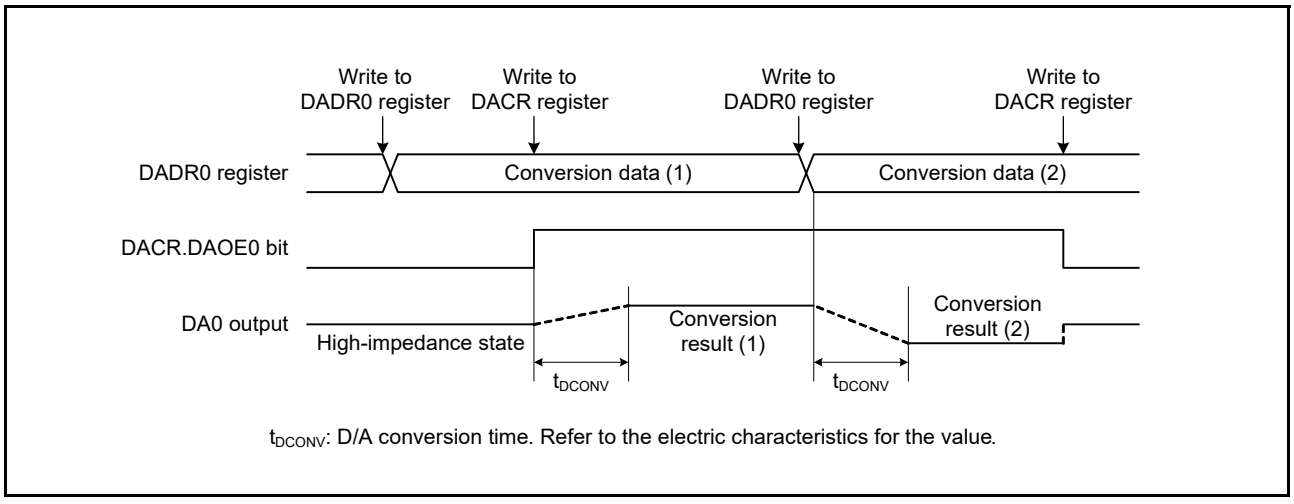


Figure 45.2 Example of 12-Bit D/A Converter Operation

Downloaded from Arrow.com.

45.3.1 Measure against Interference between D/A and A/D Conversion

When D/A conversion starts, an inrush current occurs to the 12-bit D/A converter. Since the same analog power supply is shared by the 12-bit D/A converter and 12-bit A/D converter, the inrush current may interfere with the proper operation of the 12-bit A/D converter.

With the DAADSCR.DAADST bit being 1, even if the DADRm register data (m = 0, 1) is modified during 12-bit A/D converter operation, D/A conversion does not start immediately but starts synchronously with A/D conversion completion. It takes a maximum of one A/D conversion time for the DADRm register data update to be reflected as the D/A conversion circuit input. Before reflection, the DADRm register value does not correspond to the analog output value.

When this function is enabled, it is impossible to check by any software means whether the DADRm register value has been D/A converted or not.

Even with the DAADSCR.DAADST bit being 1, when the DADRm register data is modified while the 12-bit A/D converter is halted, D/A conversion starts in one PCLKB cycle.

Figure 45.3 shows an example of channel 0 D/A conversion, in which the 12-bit D/A converter operates synchronously with the 12-bit A/D converter.

(1) Confirm that the 12-bit A/D converter is halted. Set the DAADSCR.DAADST bit to 1.
(2) Confirm that the 12-bit A/D converter is halted. Set the DACR.DAOE0 bit to 1.
(3) Set the DADR0 register. When ADCLK is faster than the peripheral module clock, it may take longer than one A/D conversion time.

- If the 12-bit A/D conversion is halted (ADCSR.ADST bit = 0) when the DADR0 register is modified, D/A conversion starts in one PCLKB cycle.
- If the 12-bit A/D conversion is in progress (ADCSR.ADST bit = 1) when the DADR0 register is modified, D/A conversion starts upon A/D conversion completion. If the DADR0 register is modified twice during A/D conversion, the first update may not be converted.

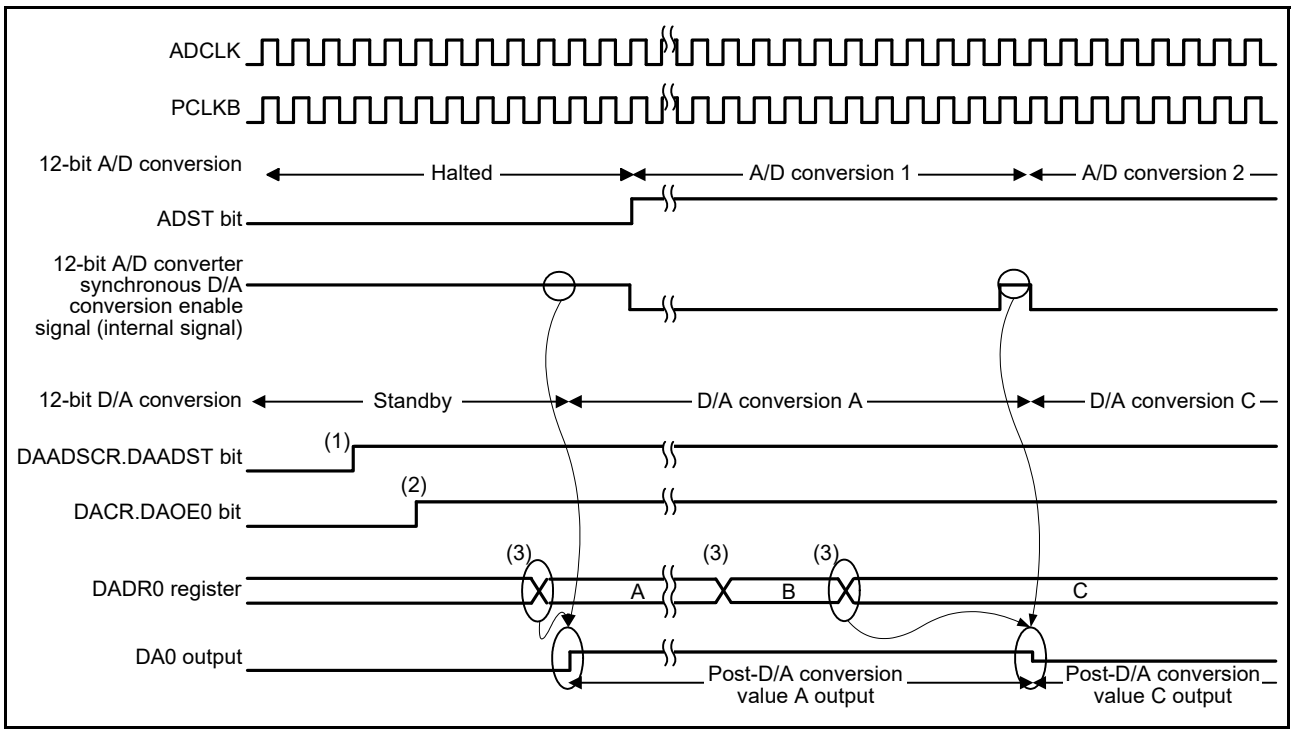


Figure 45.3 Example of Conversion When the 12-Bit D/A Converter is Synchronized with the 12-Bit A/D Converter

Downloaded from Arrow.com.

When ADCLK is faster than PCLKB, the 12-bit D/A converter may not be able to capture a 12-bit A/D converter synchronous D/A conversion enable signal for one ADCLK cycle which is output between A/D conversion 1 and A/D conversion 2. Figure 45.4 shows example when the 12-bit D/A converter cannot capture the 12-bit A/D converter synchronous D/A conversion enable signal. In this case, the DA0 output is held at the level of the post-D/A conversion value A.

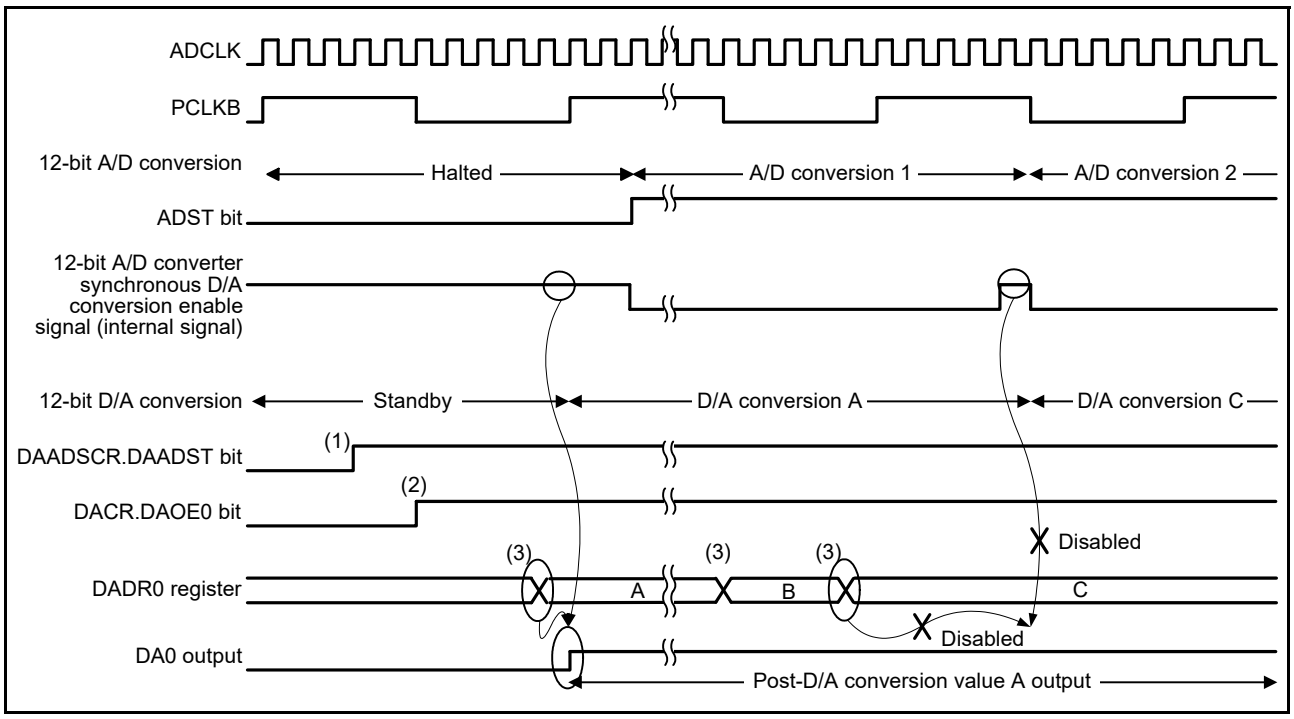


Figure 45.4 Example When the 12-Bit D/A Converter Cannot Capture the 12-Bit A/D Converter Synchronous D/A Conversion Enable Signal

Downloaded from Arrow.com.

45.3.2 Notes on Using the Internal Reference Voltage as the Reference Voltage

When setting the DAVREFCR.REF[2:0] bits to 011b to use the internal reference voltage/AVSS0 as the reference voltage, the VREF path needs to be discharged before selecting the voltage. The following shows the discharging procedure.

(1) Write 000b to the REF[2:0] bits.
(2) Set the DADR0 register to 0000h and the DADR1 register to 0000h.
(3) Keep the state of step (2) for 10 µs (discharging).
(4) After discharging is completed, write 011b to the REF[2:0] bits and select the internal reference voltage/AVSS0.
(5) Set the DACR.DAOEm bit to 1 (m = 0, 1) and wait for the stabilization wait time (5 µs) of the internal reference voltage.
(6) Write data to the DADRm register and start D/A conversion.

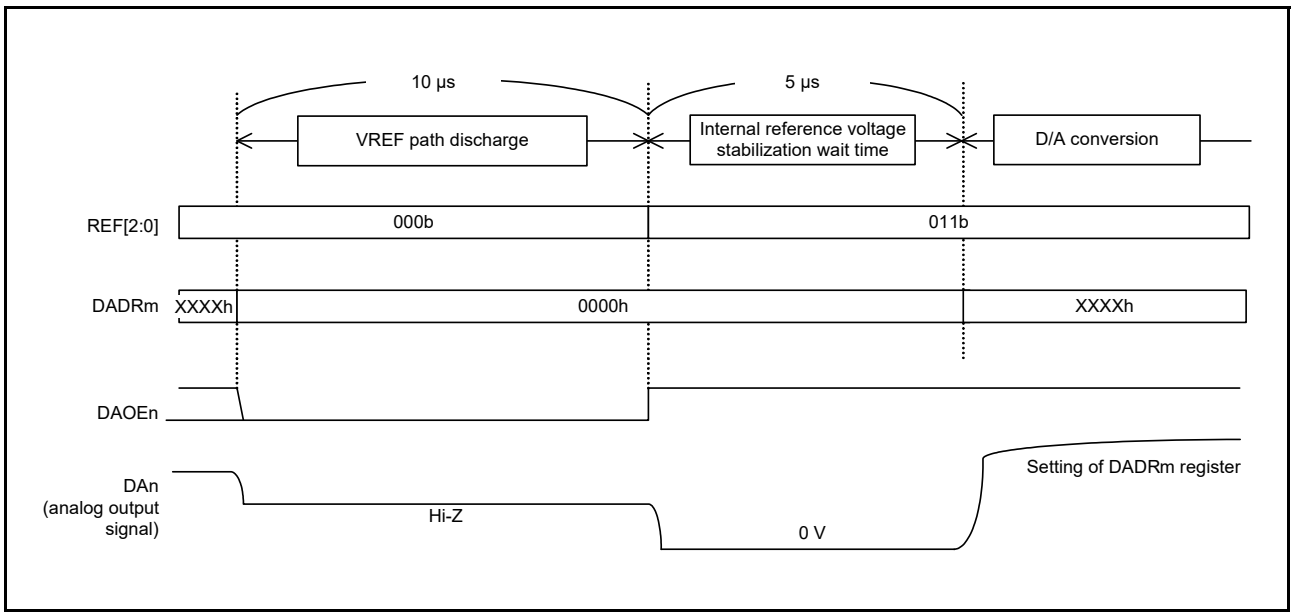


Figure 45.5 Procedure for Selecting the Internal Reference Voltage as the Reference Voltage

Downloaded from Arrow.com.

45.4 Event Link Operation Setting Procedure

The event link operation procedure is described below.

(1) Set the DADPR.DPSEL bit and set the data for D/A conversion in the DADR0 register.
(2) Set the bit value of the ELSR16 setting event signal to link the ELSR16 register of the ELC.
(3) Set the ELCR.ELCON bit to 1. This procedure enables event link operation for all modules with the event link function selected.
(4) Set the event output source module to activate the event link. After the event is output from the module, the DACR.DAOE0 bit becomes 1, and D/A conversion on channel 0 starts.
(5) Set the ELSR16.ELS[7:0] bits to 0000 0000b to stop event link operation of 12-bit D/A converter channel 0. All event link operation is stopped when the ELCR.ELCON bit is set to 0.

45.5 Usage Notes on Event Link Operation

(1) When the event specified by the ELSR16 register is generated while the write cycle is performed to the DACR.DAOE0 bit, the write cycle is stopped, and the setting to 1 by the generated event takes precedence.
(2) Use of the event link function is prohibited when the DAADSCR.DAADST bit is set to 1 as the measure against an interfere between D/A and A/D conversions.

Downloaded from Arrow.com.

45.6 Usage Notes

45.6.1 Module Stop Function Setting

Operation of the 12-bit D/A converter can be disabled or enabled using the module stop control register. The initial setting is for operation of the 12-bit D/A converter to be stopped. Register access is enabled by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

45.6.2 Operation of the D/A Converter in Module Stop State

When the MCU enters the module stop state with D/A conversion enabled, the D/A converter outputs are retained, and the analog power supply current is the same as during D/A conversion. If the analog power supply current has to be reduced in the module stop state, disable D/A conversion by setting the DACR.DAOE1, and DAOE0 bits to 0.

45.6.3 Operation of the D/A Converter in Software Standby Mode

When the MCU enters software standby mode with D/A conversion enabled, the D/A converter outputs are retained, and the analog power supply current is the same as during D/A conversion. If the analog power supply current has to be reduced in software standby mode, disable D/A conversion by setting the DACR.DAOE1, and DAOE0 bits to 0.

45.6.4 Note on Usage When Measure against Interference between D/A and A/D Conversion is Enabled

When the DAADSCR.DAADST bit is 1 (measure against interference between D/A and A/D conversion is enabled), do not place the 12-bit A/D converter in the module stop state. It may halt D/A conversion in addition to A/D conversion.

Downloaded from Arrow.com.

46. Temperature Sensor (TEMPSA)

46.1 Overview

This MCU has a built-in temperature sensor. The temperature sensor outputs a voltage proportional to temperature. By converting the output voltage of the temperature sensor into a digital value with a 12-bit A/D converter and converting it into temperature, the internal temperature of the MCU can be obtained.
Table 46.1 lists the specifications of the temperature sensor. Figure 46.1 shows a overall block diagram of the temperature sensor system.

Table 46.1 Temperature Sensor Specifications

Item	Description
Temperature sensor voltage output	The temperature sensor voltage is output to the 12-bit A/D converter.

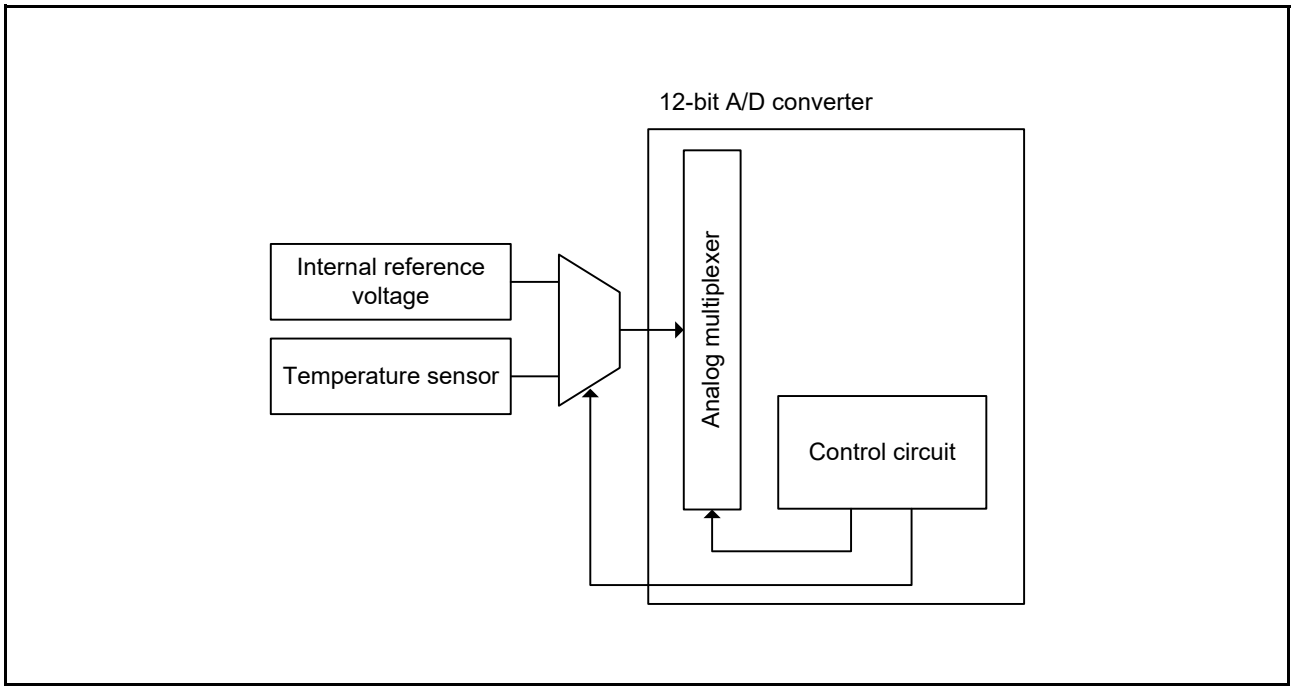


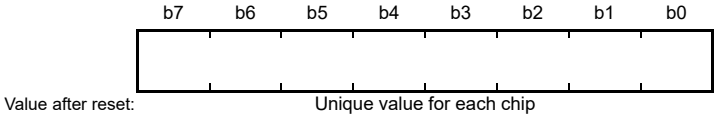
Figure 46.1 Block Diagram of Temperature Sensor System

Downloaded from Arrow.com.

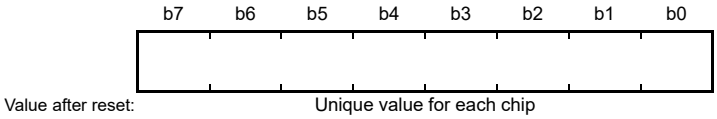
46.2 Register Descriptions

46.2.1 Temperature Sensor Calibration Data Register (TSCDRH, TSCDRL)

Address(es): TEMPSCONST.TSCDRL 007F C0ACh



Address(es): TEMPSCONST.TSCDRH 007F C0ADh



The TSCDRH and TSCDRL registers store temperature sensor calibration data measured for each chip at factory shipment.
Temperature sensor calibration data consists of the high and low bytes of the overall value (CAL_{88}) obtained by using the 12-bit A/D converter to convert the voltage output by the temperature sensor under the condition of Tj = 88°C and AVCC0 = VREFH0 = 3.3 V. The TSCDRH register stores the higher 4 bits of the converted value, and the TSCDRL register stores the lower 8 bits.

The voltage V1 output by the temperature sensor under the condition of Tj = 88°C can be calculated from the converted value CAL_{88} according to the formula below.

$V1 = 3.3 \times CAL_{88}/4096 \quad (V)$

Note that voltage V1 is independent of both the AVCC0 and VREFH0 voltages.

Downloaded from Arrow.com.

46.3 Using the Temperature Sensor

The temperature sensor outputs a voltage proportional to temperature. By converting this voltage into a digital value with a 12-bit A/D converter and converting it into temperature, the internal temperature of the MCU can be obtained.

46.3.1 Preparation for Using the Temperature Sensor

Perform a calibration of the temperature sensor as shown below. The voltage output by the temperature sensor is proportional to temperature, which can be calculated according to the following formula.

Formula for the temperature characteristic:
T = (Vs – V1)/Slope + T1

T: Measured temperature (°C)
Vs: Voltage output by the temperature sensor when the temperature is measured (V)
T1: Sample temperature measurement at first point (°C)
V1: Voltage output by the temperature sensor when T1 is measured (V)
T2: Sample temperature measurement at second point (°C)
V2: Voltage output by the temperature sensor when T2 is measured (V)
Slope: Temperature slope of the temperature sensor (V/°C); Slope = (V2 – V1)/(T2 – T1)

Since there are individual differences in temperature sensors, it is recommended to prepare a temperature slope by performing trial measurements at two different temperatures as shown below.
Use the 12-bit A/D converter to measure the voltage V1 output by the temperature sensor at temperature T1.
Again, using the 12-bit A/D converter, measure the voltage V2 output by the temperature sensor at a different temperature T2. Obtain the temperature slope (Slope = (V2 – V1)/(T2 – T1)) from these results.
Subsequently, obtain temperatures by substituting the slope into the formula for the temperature characteristic (T = (Vs – V1)/Slope + T1).

If you are using the temperature slope given in section 51, Electrical Characteristics, use the A/D converter to measure the voltage V1 output by the temperature sensor at temperature T1, and then calculate the temperature characteristic by using the formula below.
However, this method produces less accurate temperatures than measurement at two points.

T = (Vs – V1)/Slope + T1

In this MCU, the TSCDRH and TSCDRL registers store the temperature value (CAL_{88}) of the temperature sensor measured under the condition of Tj = 88°C and AVCC0 = VREFH0 = 3.3 V. By using this value as the sample measurement result at the first point, preparation before using the temperature sensor can be omitted.
This measured value CAL_{88} can be calculated as follows:

CAL_{88} = (TSCDRH register value << 8) + TSCDRL register value

If V1 is calculated from CAL_{88},

$V1 = 3.3 \times CAL_{88}/4096$ (V)

Using this, the measured temperature can be calculated according to the formula below.

T = (Vs – V1)/Slope + 88 (°C)

Downloaded from Arrow.com.

T: Measured temperature (internal temperature of the MCU) (°C)
Vs: Voltage output by the temperature sensor when the temperature is measured (V)
V1: Voltage output by the temperature sensor when Tj = 88°C and AVCC0 = VREFH0 = 3.3 V (V)
Slope: Temperature slope listed in Table 51.56 ÷ 1000 (V/°C)

Error in the measured temperature (the range of variation is 3σ) is shown in Figure 46.2.

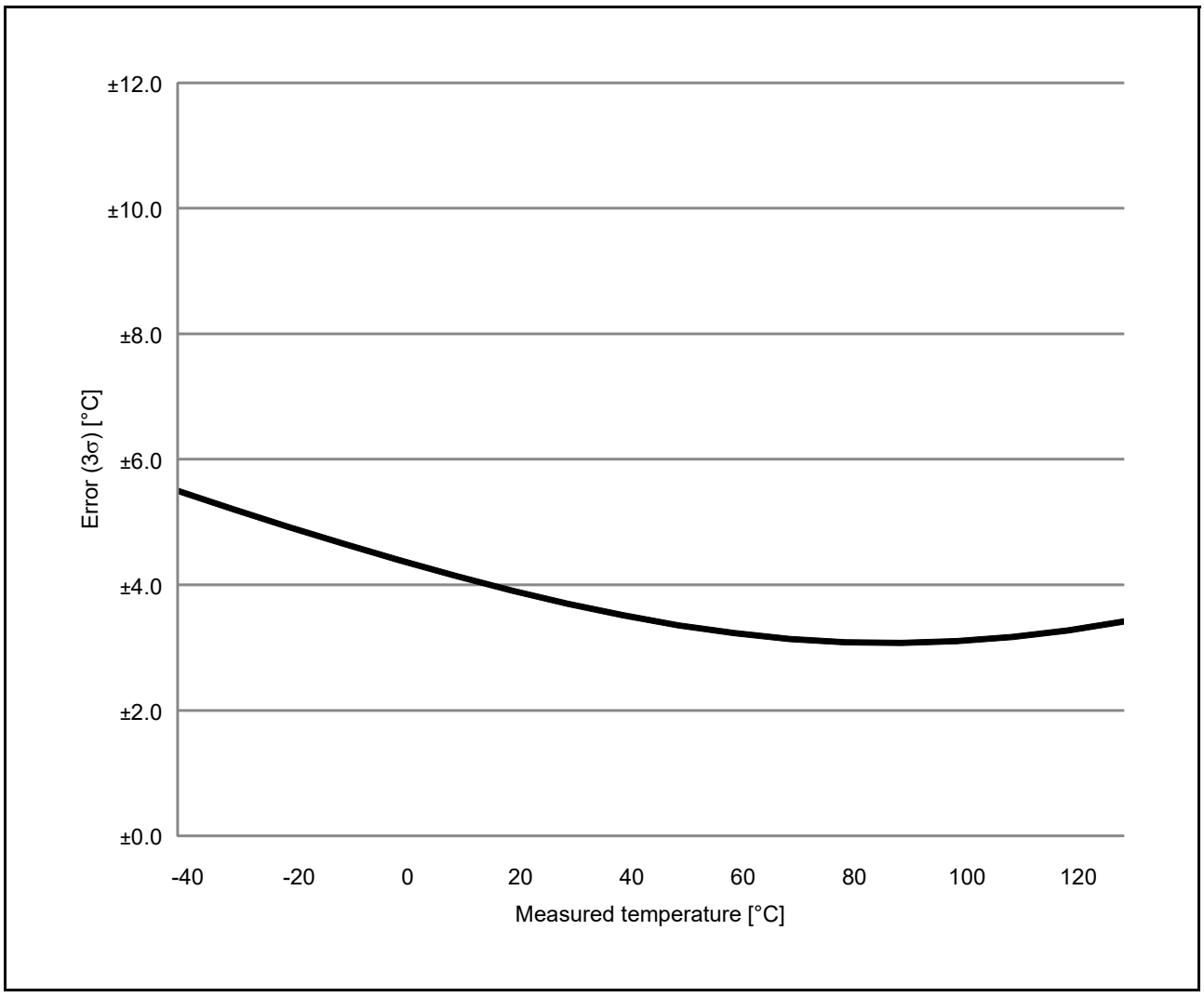


Figure 46.2 Error in the Measured Temperature (Designed Value)

46.3.2 Setting the 12-Bit A/D Converter

For details, refer to section 44, 12-Bit A/D Converter (S12ADE).

Downloaded from Arrow.com.

47. Comparator B (CMPBa)

Comparator B compares a reference input voltage and an analog input voltage. comparator B2 and comparator B3 operate independently.
In this section, “PCLK” is used to refer to PCLKB.

47.1 Overview

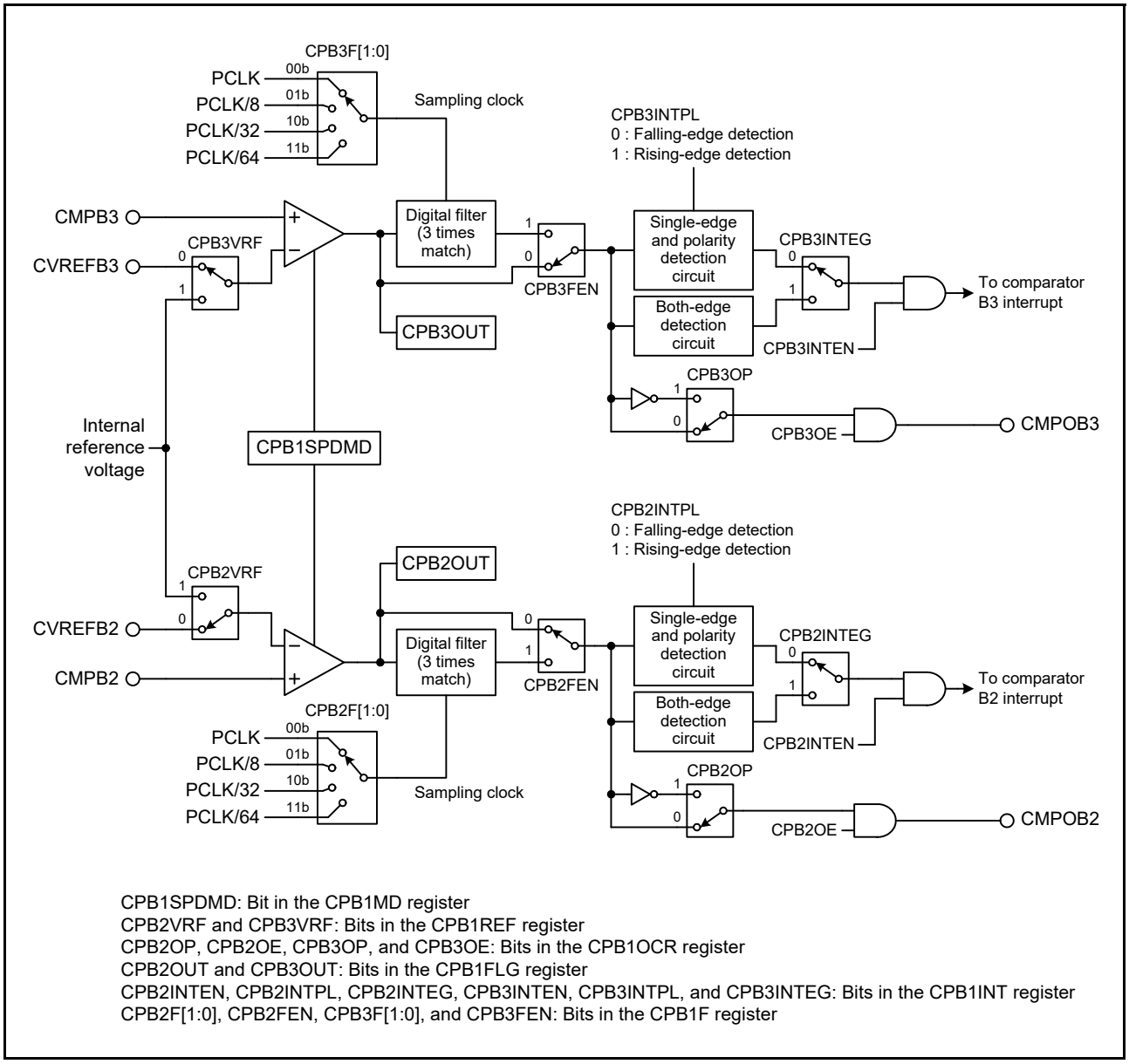
The comparison result of the reference input voltage and analog input voltage can be read by software. The comparison result can also be output externally. The reference input voltage can be selected from either an input to the CVREFBn pin (n = 2, 3) or the internal reference voltage generated internally in the MCU.
The comparator B response speed can be set before starting an operation. Setting high-speed mode decreases the response delay time, but increases current consumption. Setting low-speed mode increases the response delay time, but decreases current consumption.
Table 47.1 lists the specifications of comparator B, Figure 47.1 shows a block diagram of comparators B2 and B3 when the window function is disabled, and Figure 47.2 shows a block diagram of comparators B2 and B3 when the window function is enabled. Table 47.2 lists the I/O pins of comparator B.

Table 47.1 Comparator B Specifications (n = 2, 3)

Item	Specification
Analog input voltage	Input voltage to the CMPBn pin
Reference input voltage	Input voltage to the CVREFBn pin or internal reference voltage
Comparison result	Read from the CPBFLG.CPBnOUT flag The comparison result can be output to the CMPOBn pin.
Interrupt request generation timing	When comparator B2 comparison result changes When comparator B3 comparison result changes
Selectable function	• Digital filter function Whether the digital filter is applied or not, and the sampling frequency can be selected. • Window function Whether the window function is enabled or disabled (VRFL < CMPBn < VRFH)*1 can be selected. • Reference input voltage CVREFBn pin input or internal reference voltage (generated internally) can be selected. • Comparator B response speed High-speed mode/low-speed mode can be selected.
Low power consumption function	Module stop state can be set.

Note 1. VRFL: low-side reference voltage, VRFH: high-side reference voltage

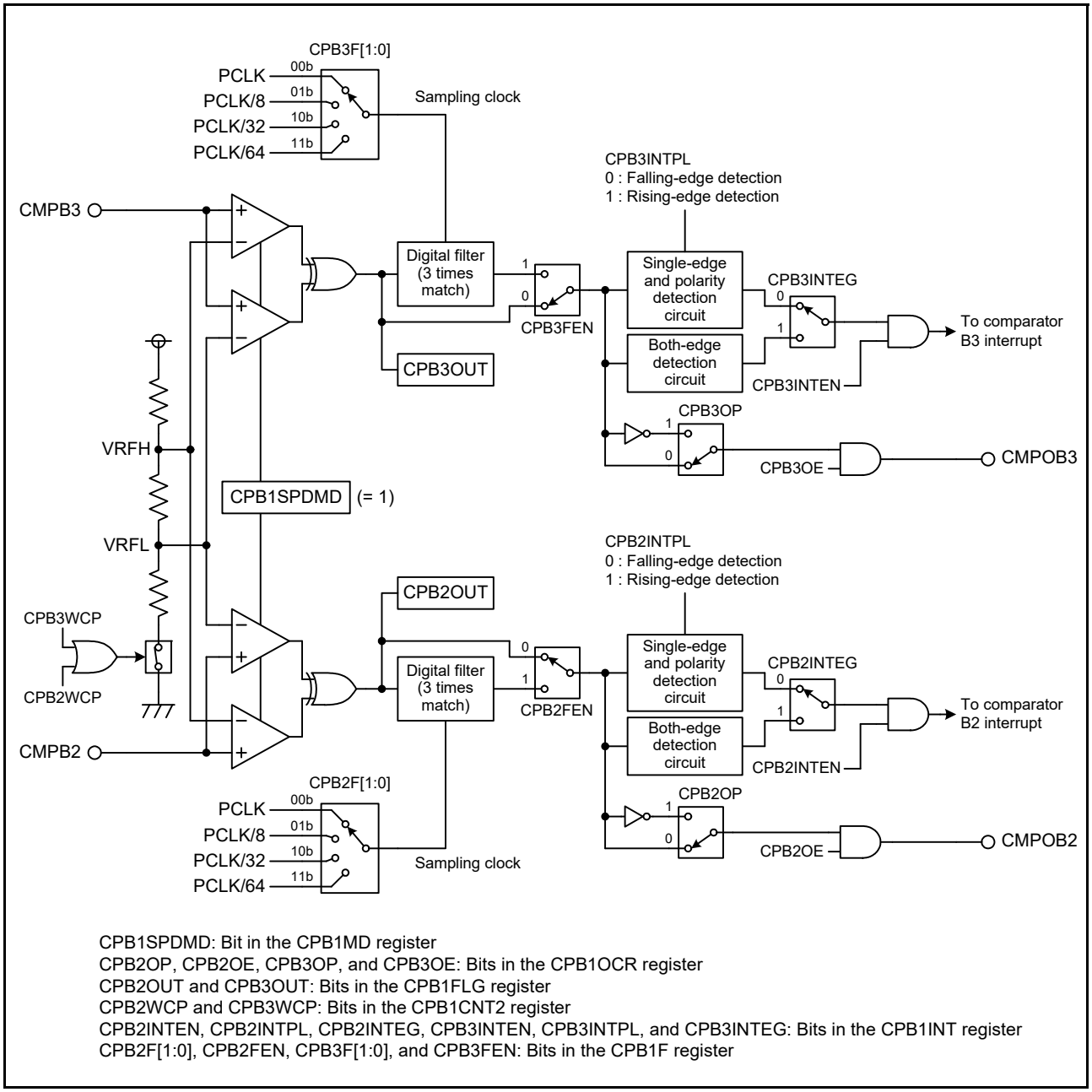
Downloaded from Arrow.com.



CPB1SPDMD: Bit in the CPB1MD register
CPB2VRF and CPB3VRF: Bits in the CPB1REF register
CPB2OP, CPB2OE, CPB3OP, and CPB3OE: Bits in the CPB1OCR register
CPB2OUT and CPB3OUT: Bits in the CPB1FLG register
CPB2INTEN, CPB2INTPL, CPB2INTEG, CPB3INTEN, CPB3INTPL, and CPB3INTEG: Bits in the CPB1INT register
CPB2F[1:0], CPB2FEN, CPB3F[1:0], and CPB3FEN: Bits in the CPB1F register

Figure 47.1 Block Diagram of Comparators B2 and B3 When Window Function is Disabled

Downloaded from Arrow.com.



CPB1SPDMD: Bit in the CPB1MD register
CPB2OP, CPB2OE, CPB3OP, and CPB3OE: Bits in the CPB1OCR register
CPB2OUT and CPB3OUT: Bits in the CPB1FLG register
CPB2WCP and CPB3WCP: Bits in the CPB1CNT2 register
CPB2INTEN, CPB2INTPL, CPB2INTEG, CPB3INTEN, CPB3INTPL, and CPB3INTEG: Bits in the CPB1INT register
CPB2F[1:0], CPB2FEN, CPB3F[1:0], and CPB3FEN: Bits in the CPB1F register

Figure 47.2 Block Diagram of Comparators B2 and B3 When Window Function is Enabled

Table 47.2 I/O Pins of Comparator B

Pin Name	I/O	Function
CMPB2	Input	Comparator B2 analog pin
CVREFB2	Input	Comparator B2 reference input voltage pin
CMPB3	Input	Comparator B3 analog pin
CVREFB3	Input	Comparator B3 reference input voltage pin
CMPOB2	Output	Comparator B2 output
CMPOB3	Output	Comparator B3 output

Downloaded from Arrow.com.

47.2 Register Descriptions

47.2.1 Comparator B1 Control Register 1 (CPB1CNT1)

Address: 0008 C5A0h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	CPB3INI	—	—	—	CPB2INI
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CPB2INI	Comparator B2 Power Enable	0: Disabled 1: Enabled (comparator powered on)	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	CPB3INI	Comparator B3 Power Enable	0: Disabled 1: Enabled (comparator powered on)	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

47.2.2 Comparator B1 Control Register 2 (CPB1CNT2)

Address: 0008 C5A1h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	CPB3WCP	—	—	—	CPB2WCP
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CPB2WCP	Comparator B2 Window Function Enable	0: Disabled 1: Enabled	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	CPB3WCP	Comparator B3 Window Function Enable	0: Disabled 1: Enabled	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Downloaded from Arrow.com.

47.2.3 Comparator B1 Flag Register (CPB1FLG)

Address: 0008 C5A2h

	b7	b6	b5	b4	b3	b2	b1	b0
	CPB3OUT	—	—	—	CPB2OUT	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b3	CPB2OUT	Comparator B2 Monitor Flag	When the window function is disabled 0: CMPB2 < CVREFB2, CMPB2 < internal reference voltage, or comparator B2 operation disabled 1: CMPB2 > CVREFB2, or CMPB2 > internal reference voltage When the window function is enabled 0: CMPB2 < VRFL*1, CMPB2 > VRFH*1, or comparator B2 operation disabled 1: VRFL < CMPB2 < VRFH	R
b6 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	CPB3OUT	Comparator B3 Monitor Flag	When the window function is disabled 0: CMPB3 < CVREFB3, CMPB3 < internal reference voltage, or comparator B3 operation disabled 1: CMPB3 > CVREFB3, or CMPB3 > internal reference voltage When the window function is enabled 0: CMPB3 < VRFL*1, CMPB3 > VRFH*1, or comparator B3 operation disabled 1: VRFL < CMPB3 < VRFH	R

Note 1. VRFL: low-side reference voltage, VRFH: high-side reference voltage

Downloaded from Arrow.com.

47.2.4 Comparator B1 Interrupt Control Register (CPB1INT)

Address: 0008 C5A3h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	CPB3INTPL	CPB3INTEG	CPB3INTEN	—	CPB2INTPL	CPB2INTEG	CPB2INTEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CPB2INTEN	Comparator B2 Interrupt Enable	0: Disabled 1: Enabled	R/W
b1	CPB2INTEG	Comparator B2 Interrupt Edge Select*1	0: Single edge 1: Both edges	R/W
b2	CPB2INTPL	Comparator B2 Interrupt Edge Polarity Select*2	0: Falling edge 1: Rising edge	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	CPB3INTEN	Comparator B3 Interrupt Enable	0: Disabled 1: Enabled	R/W
b5	CPB3INTEG	Comparator B3 Interrupt Edge Select*1	0: Single edge 1: Both edges	R/W
b6	CPB3INTPL	Comparator B3 Interrupt Edge Polarity Select*2	0: Falling edge 1: Rising edge	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note 1. The IR104.IR flag may become 1 (interrupt request is generated) when the CPB0INTPL bit is modified, and the IR105.IR flag may become 1 (interrupt request is generated) when the CPB1INTPL bit is modified. For details, refer to section 15, Interrupt Controller (ICUb).

Note 2. The CPBnINTPL bit setting is valid only when the CPBnINTEG bit is 0 (single edge is selected as the comparator interrupt edge).

Downloaded from Arrow.com.

47.2.5 Comparator B1 Filter Select Register (CPB1F)

Address: 0008 C5A4h

	b7	b6	b5	b4	b3	b2	b1	b0
	CPB3F[1:0]		—	CPB3FEN	CPB2F[1:0]		—	CPB2FEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CPB2FEN	Comparator B2 Filter Enable/Disable Select*1	0: Filter is disabled. 1: Filter is enabled.	R/W
b1	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3, b2	CPB2F[1:0]	Comparator B2 Filter Select*1	b3 b2 0 0: Sampling at PCLK 0 1: Sampling at PCLK/8 1 0: Sampling at PCLK/32 1 1: Sampling at PCLK/64	R/W
b4	CPB3FEN	Comparator B3 Filter Enable/Disable Select*1	0: Filter is disabled. 1: Filter is enabled.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7, b6	CPB3F[1:0]	Comparator B3 Filter Select*1	b7 b6 0 0: Sampling at PCLK 0 1: Sampling at PCLK/8 1 0: Sampling at PCLK/32 1 1: Sampling at PCLK/64	R/W

Note 1. The CPBnF[1:0] bits are enabled only when the CPBnFEN bit = 1 (filter is enabled).

47.2.6 Comparator B1 Mode Select Register (CPB1MD)

Address: 0008 C5A5h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	CPB1SPDMD
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CPB1SPDMD	Comparator B Speed Select*1	0: High-speed mode 1: Low-speed mode	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. When rewriting the CPB1SPDMD bit, be sure to set the CPBnINI bit (n = 2, 3) in the CPB1CNT1 register to 0 in advance.

Downloaded from Arrow.com.

47.2.7 Comparator B1 Reference Input Voltage Select Register (CPB1REF)

Address: 0008 C5A6h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	CPB3VRF	—	—	—	CPB2VRF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CPB2VRF	Comparator B2 Reference Input Voltage Select	0: Comparator B2 reference input voltage is CVREFB2 input 1: Comparator B2 reference input voltage is internal reference voltage*1, *2, *3	R/W*4
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	CPB3VRF	Comparator B3 Reference Input Voltage Select	0: Comparator B3 reference input voltage is CVREFB3 input 1: Comparator B3 reference input voltage is internal reference voltage*1, *2, *3	R/W*4
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Enabled only when the window function is disabled. When the window function is enabled, the internal reference voltage of comparator B is selected regardless of the setting of this bit.
Note 2. When the internal reference voltage is selected, the temperature sensor output cannot be selected for the A/D converter.
Note 3. When the internal reference voltage is selected, the voltage generation circuit operates and current increases by about 75 µA. This circuit is not automatically turned off even if the MCU enters software standby mode with the internal reference voltage selected.
Note 4. Do not rewrite the CPBnVRF bit when CPB1CNT2.CPBnWCP = 0.

Notes on changing the reference input voltage

- When changing the reference input voltage from CVREFBn (n = 2, 3) to the internal reference voltage, use the following procedure.

1. Set the CPB1CNT1.CPBnINI bit to 1.
2. Set the CPB1CNT2.CPBnWCP bit to 1.
3. Set the CPB1REF.CPBnVRF bit to 1 to select the internal reference voltage.
4. Set the analog select bit (ASEL) in the pin function control register of the port that is used as the CVREFBn pin to 0.
5. Wait for the comparator stabilization time (min. 100 µs).
6. Set the CPB1CNT2.CPBnWCP bit to 0.

- When changing the reference input voltage from the internal reference voltage to CVREFBn (n = 2, 3), use the following procedure.

1. Set the CPB1CNT1.CPBnINI bit to 1.
2. Set the CPB1CNT2.CPBnWCP bit to 1.
3. Set the CPB1REF.CPBnVRF bit to 0 to select the CVREFBn pin input.
4. Set the analog select bit (ASEL) in the pin function control register of the port that is used as the CVREFBn pin to 1.
5. Wait for the comparator stabilization time (min. 100 µs).
6. Set the CPB1CNT2.CPBnWCP bit to 0.

Downloaded from Arrow.com.

47.2.8 Comparator B1 Output Control Register (CPB1OCR)

Address: 0008 C5A7h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	CPB3OP	CPB3OE	—	—	CPB2OP	CPB2OE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CPB2OE	CMPOB2 Pin Output Enable	0: Comparator B2 CMPOB2 pin output disabled*1 1: Comparator B2 CMPOB2 pin output enabled	R/W
b1	CPB2OP	CMPOB2 Output Polarity Select	0: Comparator B2 output is output to CMPOB2 1: Inverted comparator B2 output is output to CMPOB2	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	CPB3OE	CMPOB3 Pin Output Enable	0: Comparator B3 CMPOB3 pin output disabled*1 1: Comparator B3 CMPOB3 pin output enabled	R/W
b5	CPB3OP	CMPOB3 Output Polarity Select	0: Comparator B3 output is output to CMPOB3 1: Inverted comparator B3 output is output to CMPOB3	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. When the CPBnOE bit (n = 2, 3) is set to 0 to disable the CMPOBn pin output, 0 is output to CMPOBn regardless of the value of the CPBnOP bit.

Downloaded from Arrow.com.

47.3 Operation

Comparator B2 and comparator B3 operate independently, and their operations are the same. Operation is not guaranteed when the values of registers are changed during comparator operation. Table 47.3 shows the procedure of setting comparator B associated registers when the window function is disabled. Table 47.4 shows the procedure of setting comparator B associated registers when the window function is enabled.

Table 47.3 Procedure for Setting Registers Associated with Comparator B When Window Function is Disabled (n = 2, 3)

Step No.	Register	Bit	Setting	
1	PijPFS of the port to which the CMPBn pin is assigned	ASEL	1	
2	CPB1MD	CPBSPDMD	Select the comparator response speed (0: High-speed mode/1: Low-speed mode)	
3	CPB1CNT1	CPBnINI	Powered on: 1	
4	CPB1CNT2	CPBnWCP	1*1	
5	CPB1REF	CPBnVRF	0: Reference input voltage = CVREFBn input*1	1: Reference input voltage = Internal reference voltage
6	PijPFS of the port to which the CVREFBn pin is assigned	ASEL	1	0
7	Waiting for the comparator stabilization time (min. 100 µs)*1			
8	CPB1CNT2	CPBnWCP	0*1	
9	CPB1F	Select whether to enable or disable the filter and select the sampling clock.		
10	Waiting for the comparator stabilization time (min. 100 µs)			
11	CPB1OCR	CPBnOP, CPBnOE	Set the CMPOBn output (select the polarity and set output enabled or disabled).	
12	CPB1INT	CPBnINTEN	When using an interrupt: 1 (interrupt enabled)	
		CPBnINTEG	When using an interrupt or the ELC: Select the input edge (1 = both edges or 0 = single edge).	
		CPBnINTPL	When using an interrupt or the ELC: For CPBnINTEG = 0 (single edge selected), select the input polarity (1 = rising edge or 0 = falling edge).	
13	IPR104 (comparator B2), IPR105 (comparator B3)	IPR[3:0]	When using an interrupt: Select the interrupt priority level.	
	IR104 (comparator B2), IR105 (comparator B3)	IR	When using an interrupt: 0 (no interrupt requested: initialization)	
	IER07	IEN0 (comparator B2), IEN1 (comparator B3)	When using an interrupt: 1 (interrupt is enabled on the interrupt controller (ICU) side)	

Note 1. This setting is necessary when changing the reference input voltage from the CVREFBn input to the internal reference voltage or from the internal reference voltage to the CVREFBn input. When selecting the CVREFBn input after the reset is released, steps 4, 5, 7, and 8 are not necessary because the initial value of the CPB1REF.CPBnVRF bit is 0.

Downloaded from Arrow.com.

Table 47.4 Procedure for Setting Registers Associated with Comparator B When Window Function is Enabled (n = 2, 3)

Step No.	Register	Bit	Setting
1	PijPFS of the port to which the CMPBn pin is assigned	ASEL	1
2	CPB1MD	CPBSPDMD	0 (always specify high-speed mode)
3	CPB1CNT1	CPBnINI	Powered on: 1
4	CPB1F	Select whether to enable or disable the filter and select the sampling clock.	
5	CPB1CNT2	CPBnWCP	1 (operation enabled)
6	Waiting for the comparator stabilization time (min. 100 µs)		
7	CPB1OCR	CPBnOP, CPBnOE	Set the CMPOBn output (select the polarity and set output enabled or disabled).
8	CPB1INT	CPBnINTEN	When using an interrupt: 1 (interrupt enabled)
		CPBnINTEG	When using an interrupt or the ELC: Select the input edge (1 = both edges or 0 = single edge).
		CPBnINTPL	When using an interrupt or the ELC: For CPBnINTEG = 0 (single edge selected), select the input polarity (1 = rising edge or 0 = falling edge).
9	IPR104 (comparator B2), IPR105 (comparator B3)	IPR[3:0]	When using an interrupt: Select the interrupt priority level.
	IR104 (comparator B2), IR105 (comparator B3)	IR	When using an interrupt: 0 (no interrupt requested: initialization)
	IER07	IEN0 (comparator B2), IEN1 (comparator B3)	When using an interrupt: 1 (interrupt enabled)

Downloaded from Arrow.com.

Figure 47.3 shows an operating example of comparator Bn (n = 2, 3) when window function is disabled.
The reference input voltage (CVREFB2/CVREFB3 or internal reference voltage) and the analog input voltage are compared. If the analog input voltage is higher than the reference input voltage, the CPB1FLG.CPBnOUT bit is set to 1. If the analog input voltage is lower than the reference input voltage, the CPBnOUT bit is set to 0.
To use the comparator Bn interrupt, set the CPB1INT.CPBnINTEN bit to 1 (interrupt enabled). If the comparison result changes at this time, a comparator Bn interrupt request is generated. For details on interrupts, refer to section 47.4, Comparator B2 and Comparator B3 Interrupts.
The values of the registers should not be changed during comparison.

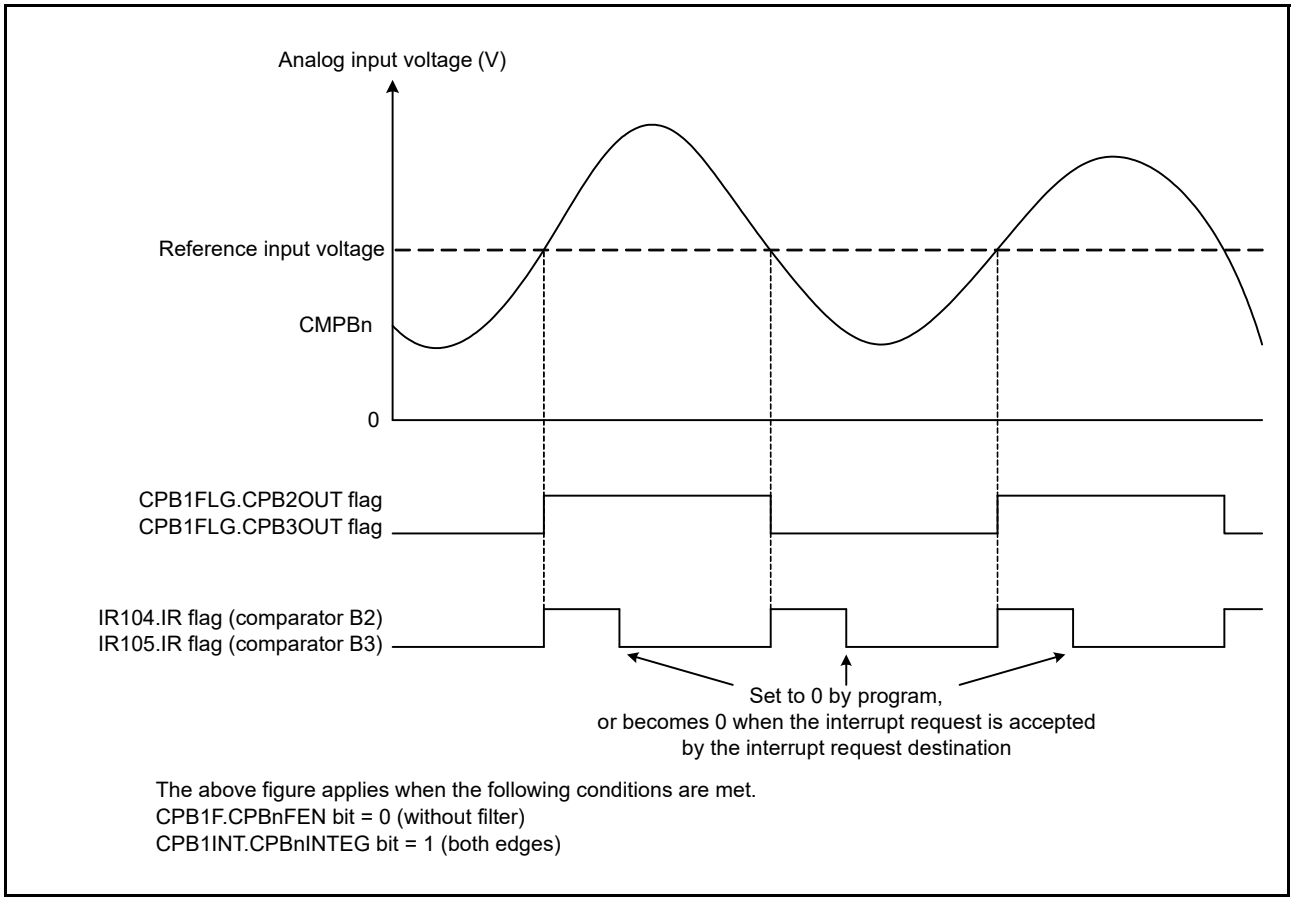


Figure 47.3 Operating Example of Comparator Bn When Window Function is Disabled (n = 2, 3)

Downloaded from Arrow.com.

Figure 47.4 shows an operation example of comparator Bn (n = 2, 3) when the window function is enabled.
The internal reference voltage (VRFH/VRFL) for the window function and the analog input voltage are compared. The CPBnOUT bit is set to 1 when VRFL < the analog input voltage < VRFH, and the CPBnOUT bit is set to 0 when the analog input voltage < VRFL, or VRFH < the analog input voltage.
To use the comparator Bn interrupt, set the CPB1INT.CPBnINTEN bit to 1 (interrupt enabled). If the comparison result changes at this time, a comparator Bn interrupt request is generated. For details on interrupts, refer to section 47.4, Comparator B2 and Comparator B3 Interrupts.
The values of the registers should not be changed during comparison.

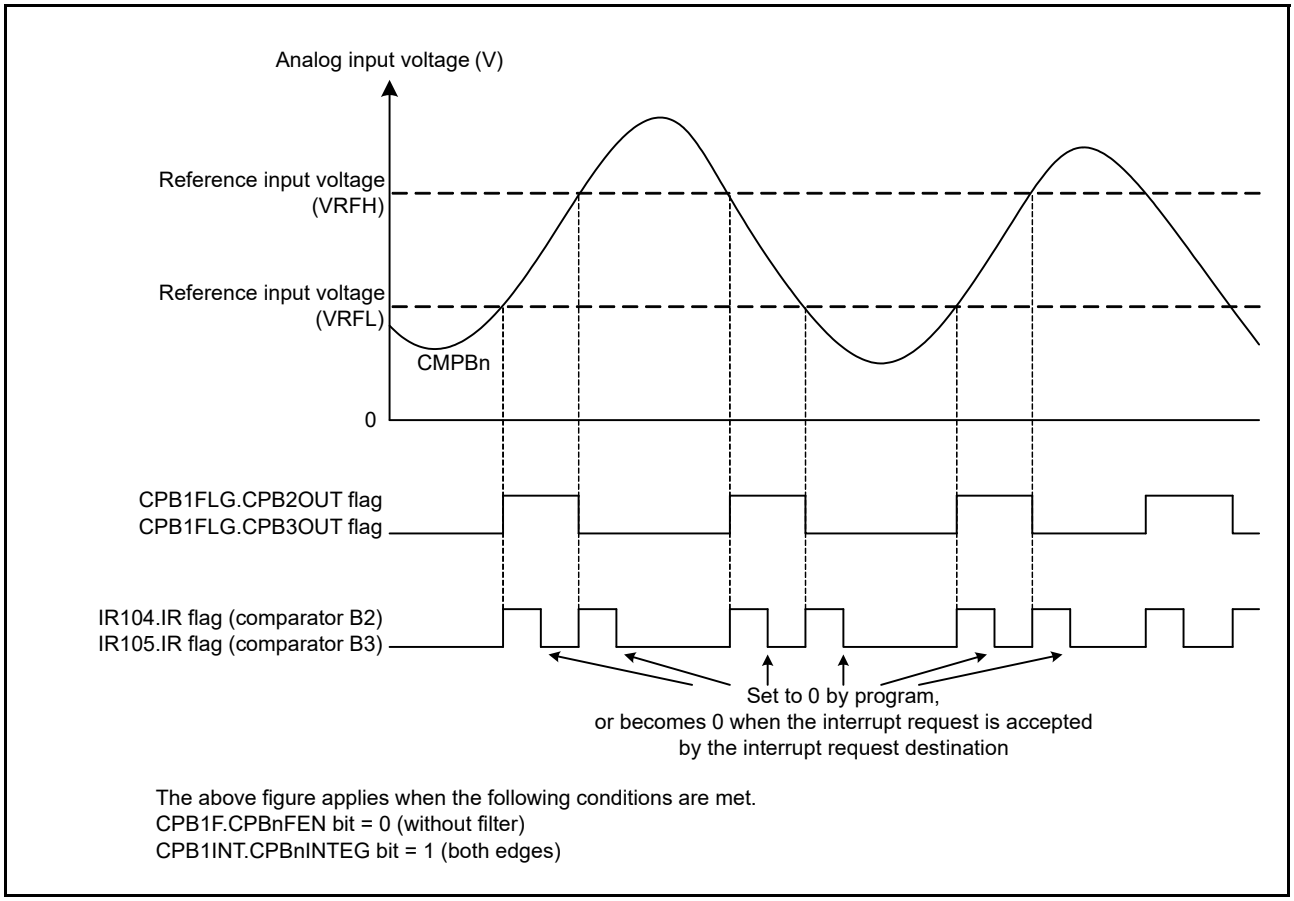


Figure 47.4 Operating Example of Comparator Bn When Window Function is Enabled (n = 2, 3)

Downloaded from Arrow.com.

47.3.1 Comparator Bn Digital Filter (n = 2, 3)

The sampling clock can be selected by the CPB1F.CPBnF[1:0] bits. The CPBnOUT signal (internal signal) output from comparator Bn is sampled at every sampling clock cycle. At the next clock timing after the level matches three times, the IR104.IR flag (when comparator B2 selected) or IR105.IR flag (when comparator B3 selected) is set to 1 (interrupt requested).

Figure 47.5 shows the configuration of the comparator Bn digital filter, and Figure 47.6 shows an operating example of the comparator Bn digital filter.

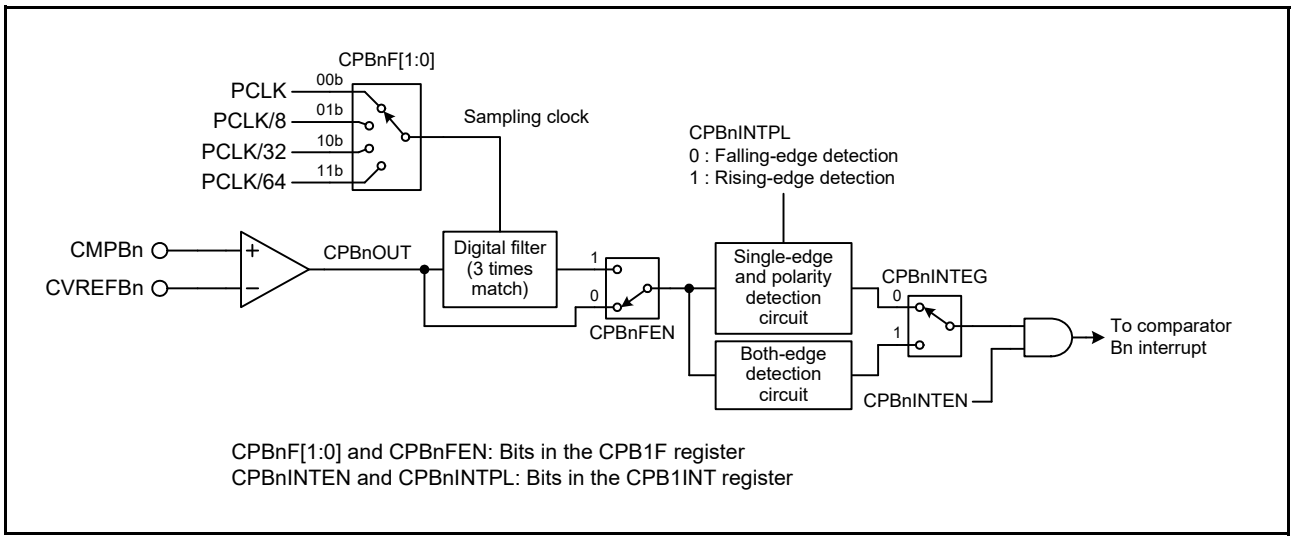


Figure 47.5 Configuration of Comparator Bn Digital Filter (n = 2, 3)

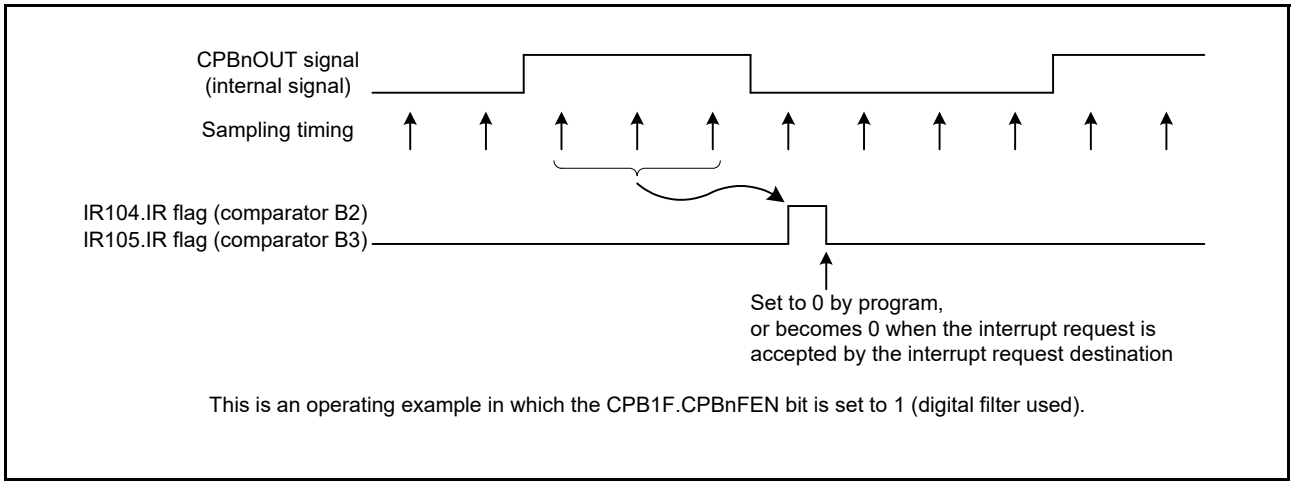


Figure 47.6 Operating Example of Comparator Bn Digital Filter (n = 2, 3)

Downloaded from Arrow.com.

47.3.2 Comparator Bn Output Function (n = 2, 3)

The comparison result from comparator B can be output to external pins. The CPB1OCR.CPBnOP and CPB1OCR.CPBnOE bits can be used to set the output polarity (non-inverted output or inverted output) and output enabled or disabled. For the register settings and corresponding comparator output, refer to section 47.2.8, Comparator B1 Output Control Register (CPB1OCR).

To output the comparator B comparison result to the CMPOB2 or CMPOB3 output pin, use the following procedure to make port settings. Note that the ports are set to input after a reset.

(1) Set the mode and input for comparator B (steps 1 to 10 listed in Table 47.3 and steps 1 to 6 listed in Table 47.4).
(2) Select the polarity of the CMPOB2 or CMPOB3 output and enable the output (set the CPB1OCR.CPBnOP and CPB1OCR.CPBnOE bits).
(3) Set the port mode register and pin function control register corresponding to the CMPOB2 or CMPOB3 output pin (start outputting from the pin).

47.3.3 Example of Using Comparator B to Exit Software Standby Mode

The following shows an example of using comparator B2 output to exit software standby mode.
In this example, it is assumed that the reference input voltage (CVREFB2) > analog input voltage (CMPB2).

Set the following steps (1) to (3) before entering software standby mode.

(1) Set the registers associated with comparator B2 according to section 47.3, Operation.
However, set the CPB1F.CPB2FEN bit to ‘filter is disabled’, the CPB1OCR.CPB2OE bit to ‘output enabled’, and the CPB1OCR.CPB2OP bit to ‘comparator B2 output is output to CMPOB2’.
(2) Make the IRQ7 interrupt settings according to section 15.4.8, External Pin Interrupts.
However, set the IRQFLTE0.FLTEN7 bit to 0 (digital filter disabled) and set the IRQCR7.IRQMD[1:0] bits to the same polarity as that of comparator B2 output.
In this example, a rising edge is selected.
(3) Set the multi-function pin controller to select the CMPOB2 function and enable IRQ7.

When exiting software standby mode, input a voltage from the comparator B2 analog pin (CMPB2) so that the reference input voltage (CVREFB2) is less than the analog input voltage (CMPB2). This allows the IRQ7 interrupt to be generated through the comparator B2 output pin (CMPOB2) and the MCU exits software standby mode.

Downloaded from Arrow.com.

47.4 Comparator B2 and Comparator B3 Interrupts

Comparator B generates two interrupt requests from sources, comparator B2 and . The comparator Bn interrupt (n = 2, 3) uses the IR104.IR flag, IR105.IR flag, IPR104.IPR[3:0] bits, IPR105.IPR[3:0] bits, and the respective single interrupt vector.

To use the comparator Bn interrupt, set the CPB1INT.CPBnINTEN bit to 1 (interrupt enabled). In addition, select either single-edge detection or both-edge detection using the CPB1INT.CPBnINTEG bit. When single-edge detection is selected, select the polarity using the CPB1INT.CPBnINTPL bit.

Inputs can also be passed through the digital filter with four different sampling clocks.

47.5 Usage Note

47.5.1 Module Stop Function Setting

Operation of comparator B can be enabled or disabled by setting a bit in the module stop control register B (MSTPCRB). Comparator B is initially disabled after a reset. Registers only become accessible after it has been released from the module stop state. For details, refer to section 11, Low Power Consumption.

Downloaded from Arrow.com.

48. Data Operation Circuit (DOC)

48.1 Overview

The data operation circuit (DOC) is used to compare, add, or subtract 16-bit values.
Table 48.1 lists the specifications of the DOC and Figure 48.1 is a block diagram of the DOC.
An interrupt can be generated if the result of 16-bit comparison meets one of the set interrupt conditions.

Table 48.1 DOC Specifications

Item	Description
Data operation function	16-bit data comparison, addition, and subtraction
Lower power consumption function	The DOC can be placed in a module-stop state.
Interrupts	• The result of data comparison meets the detection condition. • The result of data addition is greater than FFFFh, which is an overflow. • The result of data subtraction is less than 0000h, which is an underflow.
Event link function (output)	• The result of data comparison meets the detection condition. • The result of data addition is greater than FFFFh, which is an overflow. • The result of data subtraction is less than 0000h, which is an underflow.

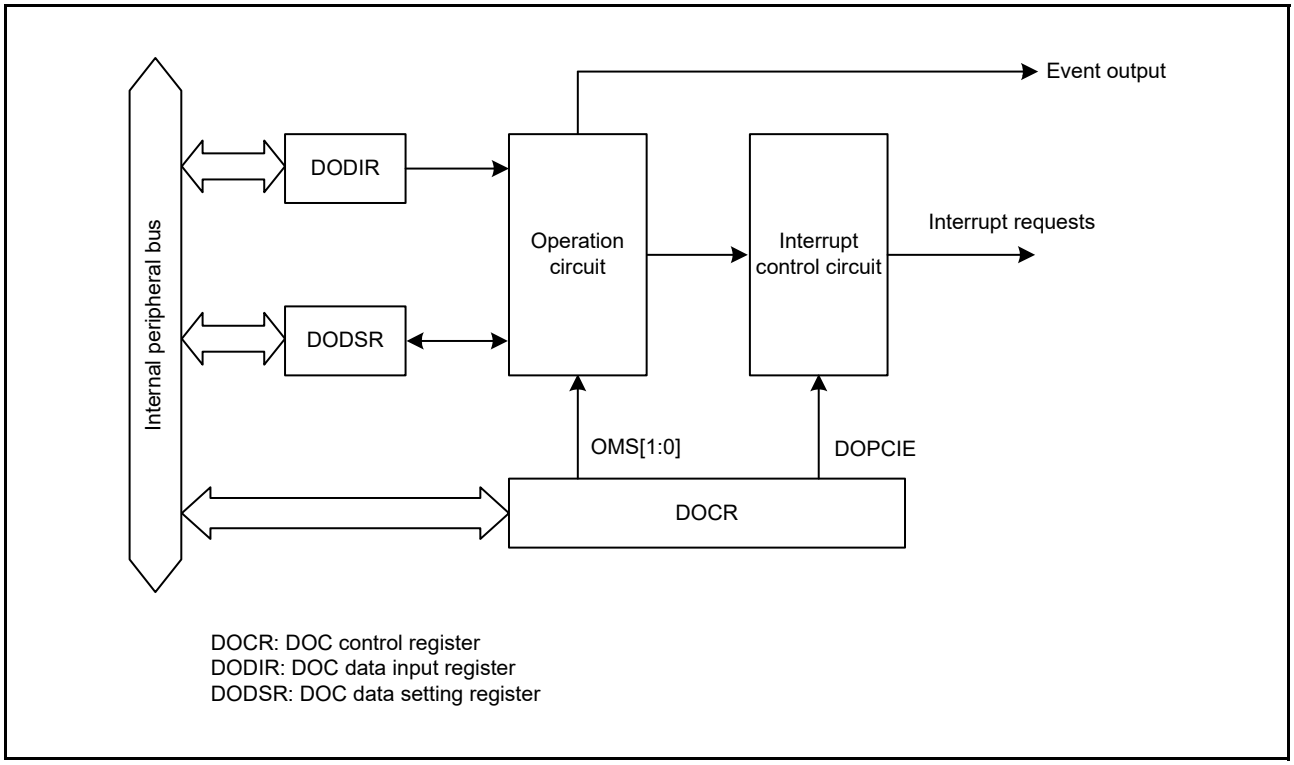


Figure 48.1 DOC Block Diagram

Downloaded from Arrow.com.

48.2 Register Descriptions

48.2.1 DOC Control Register (DOCR)

Address(es): DOC.DOCR 0008 B080h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	DOPCFCL	DOPCF	DOPCIE	—	DCSEL	OMS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	OMS[1:0]	Operating Mode Select	b1 b0 0 0: Data comparison mode 0 1: Data addition mode 1 0: Data subtraction mode 1 1: Setting prohibited	R/W
b2	DCSEL	Detection Condition Select*1	0: ‘Not equal to’ is to be detected. 1: ‘Equal to’ is to be detected.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	DOPCIE	Data Operation Circuit Interrupt Enable	0: Interrupt disabled 1: Interrupt enabled	R/W
b5	DOPCF	Data Operation Result Flag	Indicates the result of an operation.	R
b6	DOPCFCL	Data Operation Result Clear	0: Retain the value of the DOPCF flag. 1: Clears the DOPCF flag.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note 1. Valid only when data comparison mode is selected.

The DOCR register specifies the operation of DOC, or enabling or disabling of the interrupt.

OMS[1:0] Bits (Operating Mode Select)

These bits select the operating mode of the DOC.

DCSEL Bit (Detection Condition Select)

This bit is valid only when data comparison mode is selected.

This bit selects the condition for detection in data comparison mode.

DOPCIE Bit (Data Operation Circuit Interrupt Enable)

Setting this bit to 1 enables interrupts from the DOC.

DOPCF Flag (Data Operation Result Flag)

[Setting conditions]

- The condition selected by the DCSEL bit is met
- A result of data addition is greater than FFFFh
- A result of data subtraction is less than 0000h

[Clearing condition]

- Writing 1 to the DOPCFCL bit

Downloaded from Arrow.com.

DOPCFCL Bit (Data Operation Result Clear)

Writing 1 to this bit clears the DOPCF flag.

This bit is read as 0.

48.2.2 DOC Data Input Register (DODIR)

Address(es): DOC.DODIR 0008 B082h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The DODIR register is a readable and writable register that holds values for use in operations.

48.2.3 DOC Data Setting Register (DODSR)

Address(es): DOC.DODSR 0008 B084h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The DODSR register is a readable and writable register that holds values for use in comparison or the results of other operations.

In data comparison mode, store the standard value for use in comparison in this register.

In data addition or data subtraction mode, this register holds the results of operations.

Downloaded from Arrow.com.

48.3 Operation

48.3.1 Data Comparison Mode

Figure 48.2 shows an example of the steps involved in data comparison mode operation by the DOC.
An example of operation when DCSEL is set to 0 (‘not equal to’ is to be detected as the result of data comparison) is shown below.

(1) Writing 00b to the DOCR.OMS[1:0] bits places the DOC in the data comparison mode.
(2) Specify the standard value for comparison in the DODSR register.
(3) Write the value for comparison in the DODIR register.
(4) Write all values for use in comparison to the DODIR register.
(5) If the value written to the DODIR register is not equal to the value set in the DODSR register, the DOCR.DOPCF flag becomes 1. If the DOCR.DOPCIE bit is 1, a data operation circuit interrupt is also issued.

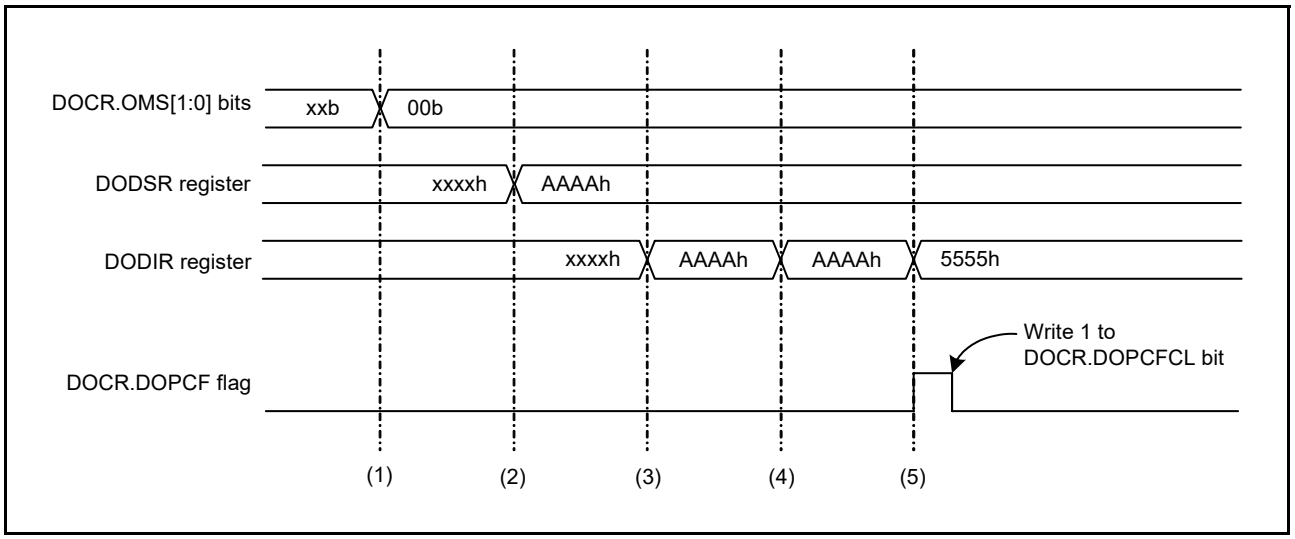


Figure 48.2 Example of Operation in Data Comparison Mode

Downloaded from Arrow.com.

48.3.2 Data Addition Mode

Figure 48.3 shows an example of the steps involved in data addition mode operation by the DOC.

(1) Writing 01b to the DOCR.OMS[1:0] bits selects data addition mode.
(2) Set the initial value in the DODSR register.
(3) Write the value for addition in the DODIR register. The result of the operation is stored in DODSR.
(4) Write all values for use in addition to the DODIR register.
(5) If the result of the operation is greater than FFFFh, the DOCR.DOPCF flag becomes 1. If the DOCR.DOPCIE bit is 1, a data operation circuit interrupt is also issued.

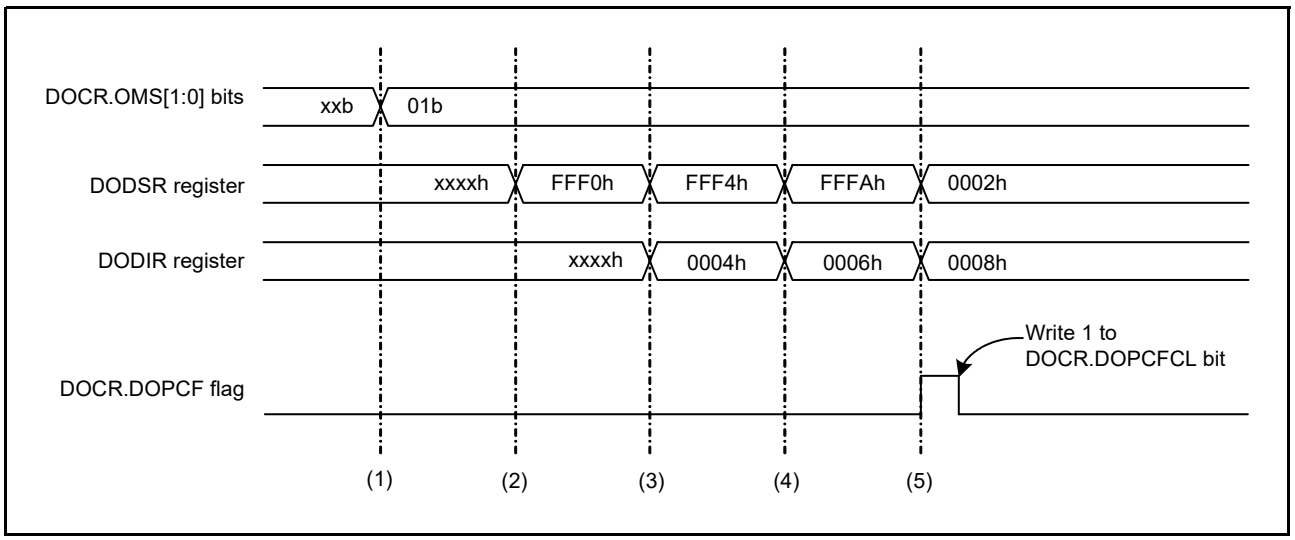


Figure 48.3 Example of Operation in Data Addition Mode

Downloaded from Arrow.com.

48.3.3 Data Subtraction Mode

Figure 48.4 shows an example of the steps involved in data subtraction mode operation by the DOC.

(1) Writing 10b to the DOCR.OMS[1:0] bits selects data subtraction mode.
(2) Set the initial value in the DODSR register.
(3) Write the value for subtraction in the DODIR register. The result of the operation is stored in DODSR.
(4) Write all values for use in subtraction to the DODIR register.
(5) If the result of the operation is less than 0000h, the DOCR.DOPCF flag becomes 1. If the DOCR.DOPCIE bit is 1, a data operation circuit interrupt is also issued.

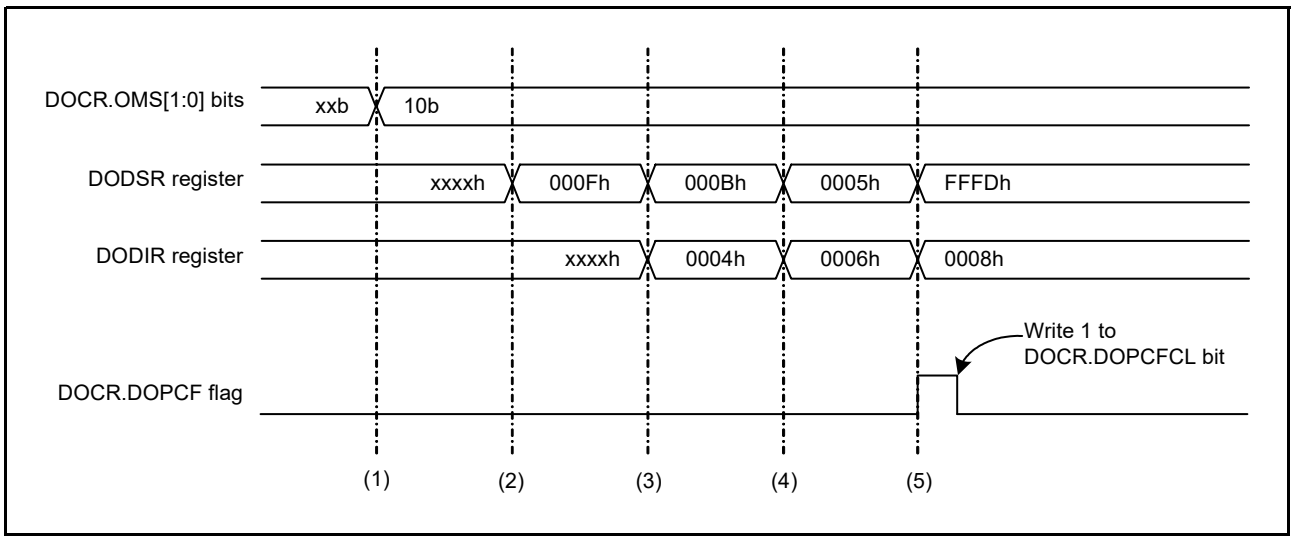


Figure 48.4 Example of Operation in Data Subtraction Mode

48.4 Interrupt Requests

The data operation circuit interrupt (DOPCI) is the interrupt request generated by the DOC. The DOCR.DOPCF flag becomes 1 when the interrupt source condition is satisfied.
Table 48.2 lists the details of the interrupt request.

Table 48.2 Interrupt Request from DOC

Interrupt Request	Data Operation Result Flag	Interrupt Generation Timing
Data operation circuit interrupt (DOPCI)	DOPCF	• The result of data comparison meets the detection condition. • The result of data addition is greater than FFFFh. • The result of data subtraction is less than 0000h.

Downloaded from Arrow.com.

48.5 Event Link Output

The DOC outputs event signals for the event link controller (ELC) under the following conditions, and these can be used to initiate operations by other modules selected in advance.

- The result of data comparison meets the detection condition.
- The result of data addition is greater than FFFFh.
- The result of data subtraction is less than 0000h.

48.5.1 Interrupt Handling and Event Linking

The DOC has a bit to enable or disable interrupts. When an interrupt source condition is satisfied while the interrupt is enabled, the interrupt request signal is issued to the CPU.
In contrast, an event link output signal is sent to other modules as an event signal via the ELC when an interrupt source is generated, regardless of the setting of the corresponding interrupt enable bit.

48.6 Usage Note

48.6.1 Module Stop Function Setting

Operation of the DOC can be enabled or disabled by setting the MSTPB6 bit in module stop control register B (MSTPCRB). The DOC is initially disabled after a reset. Register access is enabled by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

Downloaded from Arrow.com.

49. RAM

This MCU has an on-chip high-speed static RAM.

49.1 Overview

Table 49.1 lists the specifications of the RAM.

Table 49.1 Specifications of RAM

Item	Description
Capacity	64 Kbytes (0000 0000h to 0000 FFFFh)
Access	• Single-cycle access is possible for both reading and writing. • On-chip RAM can be enabled or disabled.*1
Low power consumption function	Transitions to the module stopped state are possible.

Note 1. Selectable by the RAME bit in SYSCR1. For details on SYSCR1, see section 3.2.2, System Control Register 1 (SYSCR1).

49.2 Operation

49.2.1 Low Power Consumption Function

Power consumption can be reduced by setting module stop control register C (MSTPCRC) to stop supply of the clock signal to the RAM.
Setting the MSTPCRC.MSTPC0 bit to 1 stops supply of the clock signal to RAM.
Stopping supply of the clock signal places the RAM in the module stop state. The RAM operates after initialization by a reset.
The RAM is not accessible in the module stop state. Do not allow transitions to the module stop state while access to RAM is in progress.
For details on the MSTPCRC register, see section 11, Low Power Consumption.

49.2.2 Notes on Self-Diagnosis of the RAM

A write buffer is mounted for the RAM. When the same address is read after a write operation, data in the write buffer, rather than in the memory cell of the RAM may be read. When the RAM is self-diagnosed, confirm that the data have been written by following the procedure below so that data will not be read from the write buffer.

(1) Write data to the address targeted for diagnosis.
(2) Write data to an address which is at least 4 addresses away from the that in (1).
(3) Read the data from the address in (1).

Downloaded from Arrow.com.

50. Flash Memory (FLASH)

This MCU has packages with 384 and 512 Kbyte flash memory (ROM) for storing code and 8-Kbyte flash memory (E2 DataFlash) for storing data.
In this section, “PCLK” is used to refer to PCLKB.

50.1 Overview

Table 50.1 lists the flash memory specifications.
Table 50.7 lists the I/O pins used in boot mode.

Table 50.1 Flash Memory Specifications

Item	Description
Memory space	• User area: Up to 512 Kbytes • Data area: 8 Kbytes • Extra area: Stores the start-up area information, access window information, and unique ID
Software commands	• The following commands are implemented: Program, blank check, block erase, and all-block erase • The following commands are implemented for programming the extra area: Start-up area information program and access window information program
Value after erasure	• ROM: FFh • E2 DataFlash: FFh
Interrupt	An interrupt (FRDYI) is generated upon completion of software command processing or forced stop processing.
On-board programming	Boot mode (SCI Interface)*1 • Channel 1 of the serial communications interface (SCI1) is used for asynchronous serial communication. • The user area and data area are rewritable. Boot mode (FINE interface)*1 • The FINE is used. • The user area and data area are rewritable. Boot mode (USB interface)*1 • Channel 0 of the USB 2.0 function (USB0) module is used. • The user area and data area are rewritable. • The flash memory can be rewritable in self-powered or bus-powered mode. • A personal computer can be connected using only a USB cable. Self-programming in single-chip mode • The user area and data area are rewritable using the flash rewrite routine in the user program.
Off-board programming	The user area and data area are rewritable using a flash programmer compatible with this MCU.
ID code protection	• Connection with the serial programmer can be enabled or disabled using ID codes in boot mode. • Connection with the on-chip debugging emulator can be enabled or disabled using ID codes.
Start-up program protection	This function is used to safely rewrite block 0 to block 7.
Area protection	This function enables rewriting only the selected blocks in the user area and disables the other blocks during self-programming.
Background Operation (BGO)	Programs on the ROM can be executed while rewriting the E2 DataFlash.

Note 1. Refer to the manual of each serial programmer and “Renesas Flash Programmer Flash memory programming software User’s Manual” for more details.

Downloaded from Arrow.com.

50.2 ROM Area and Block Configuration

The maximum ROM size of this MCU is 512 Kbytes. The ROM area is divided into blocks. A block is 2-Kbyte area. When executing the block erase command, the memory is erased by the block. Figure 50.1 shows the ROM area and block configuration.

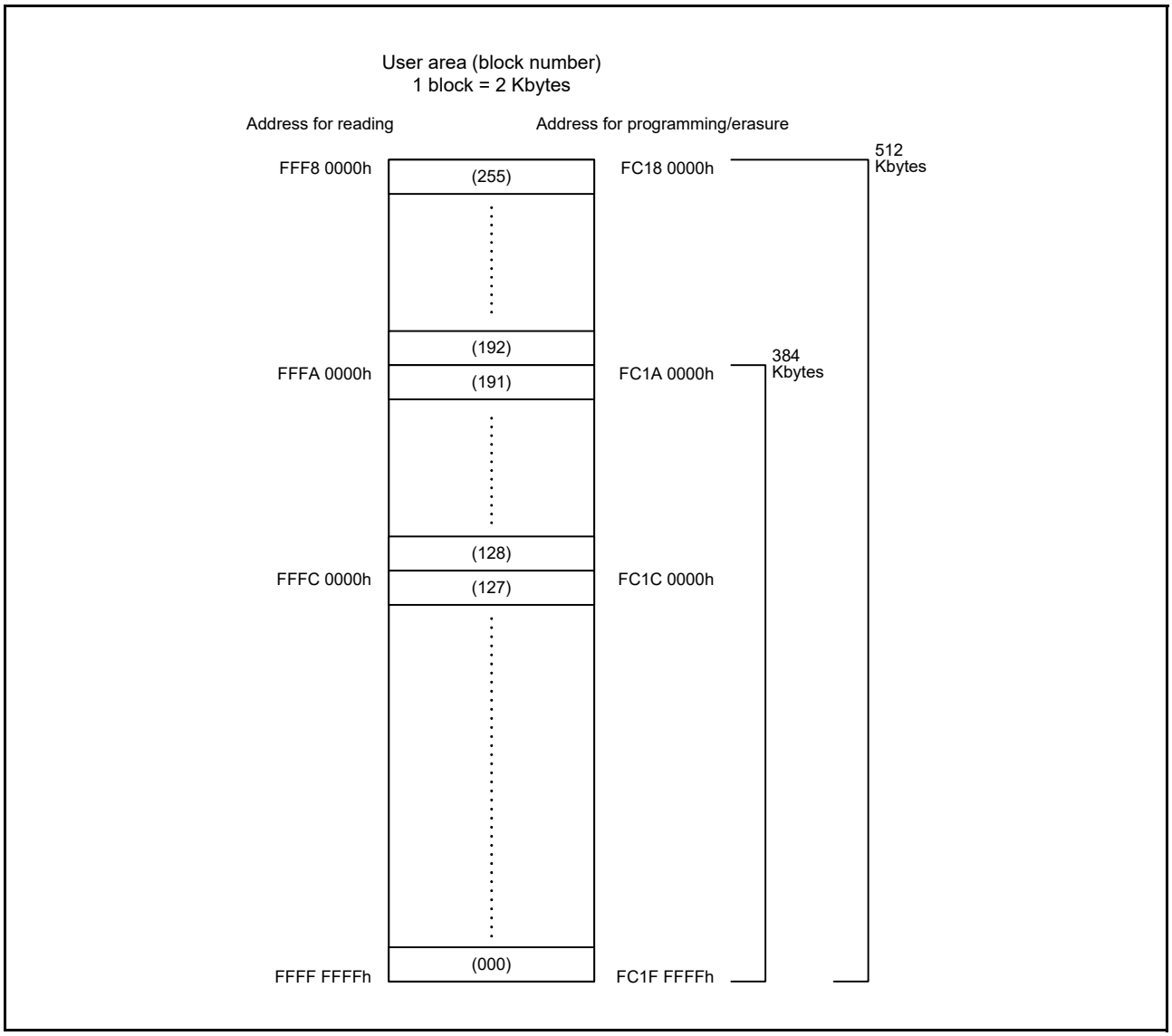


Figure 50.1 ROM Area and Block Configuration

Table 50.2 Correspondence Between ROM Capacity and Addresses for Reading

ROM Capacity	Addresses for Reading
512 Kbytes	FFF8 0000h to FFFF FFFFh
384 Kbytes	FFFA 0000h to FFFF FFFFh

Downloaded from Arrow.com.

50.3 E2 DataFlash Area and Block Configuration

The E2 DataFlash is 8 Kbytes in the MCU. The E2 DataFlash is divided into blocks and erased in block units. Figure 50.2 shows the E2 DataFlash area and the block configuration.

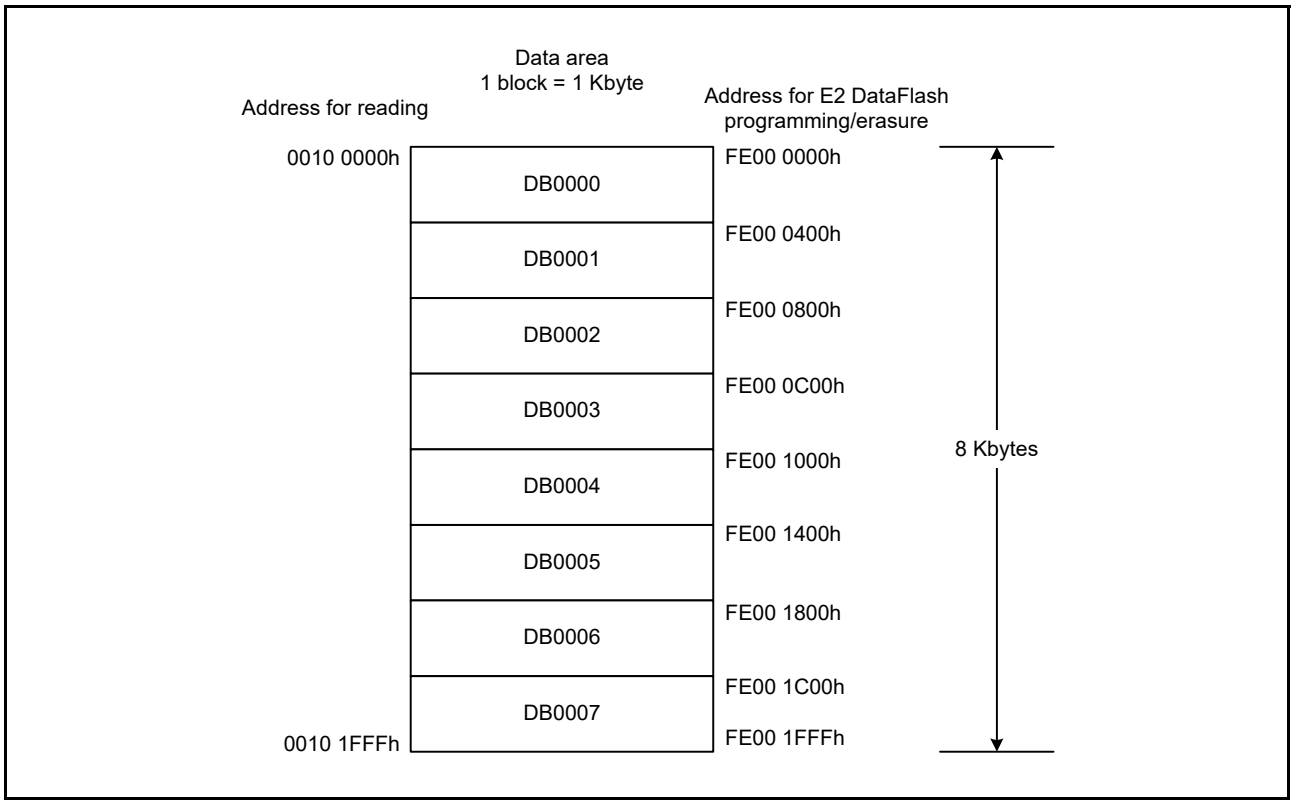


Figure 50.2 E2 DataFlash Area and Block Configuration

Downloaded from Arrow.com.

50.4 Register Descriptions

50.4.1 E2 DataFlash Control Register (DFLCTL)

Address(es): FLASH.DFLCTL 007F C090h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	DFLEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DFLEN	E2 DataFlash Access Enable	0: Access to E2 DataFlash and access to the extra area in P/E mode*1 disabled 1: Access to E2 DataFlash and access to the extra area in P/E mode*1 enabled	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Start-up area information program and access window information program

The DFLCTL register is used to enable or disable access (read, program, and erase) to the E2 DataFlash and access (start-up area information program and access window information program) to the extra area in P/E mode.
When reading, programming, and erasing the E2 DataFlash, set the DFLCTL.DFLEN bit to 1 and wait for the E2 DataFlash STOP recovery time (tDSTOP) to elapse before reading the E2 DataFlash and entering E2 DataFlash P/E mode. Do not read the E2 DataFlash or enter E2 DataFlash P/E mode until tDSTOP has elapsed.
Refer to section 50.7.1, Sequencer Modes for details on E2 DataFlash P/E mode. Refer to section 51, Electrical Characteristics for E2 DataFlash STOP recovery time (tDSTOP).

Downloaded from Arrow.com.

50.4.2 Flash P/E Mode Entry Register (FENTRYR)

Address(es): FLASH.FENTRYR 007F FFB2h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	FEKEY[7:0]								FENTRYD	—	—	—	—	—	—	FENTRY0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	FENTRY0	ROM P/E Mode Entry 0	0: ROM is in read mode. 1: ROM can be placed in P/E mode.	R/W
b6 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	FENTRYD	E2 DataFlash P/E Mode Entry	0: E2 DataFlash is in read mode. 1: E2 DataFlash can be placed in P/E mode.	R/W
b15 to b8	FEKEY[7:0]	Key Code	The FEKEY[7:0] bits are used to control rewriting of the FENTRYR register. When rewriting the value of the lower 8 bits, set the FEKEY[7:0] bits to AAh at the same time (write this register in 16 bits). The FEKEY[7:0] bits are read as 00h.	R/W

To rewrite the ROM or E2 DataFlash, the FENTRYD or FENTRY0 bit must be set to 1 to place the ROM or E2 DataFlash in P/E mode.
When returning to read mode, set the FENTRYR register and confirm that its value has been rewritten before reading the ROM or E2 DataFlash.
Refer to section 50.7.1, Sequencer Modes for details on P/E mode and read mode.

FENTRY0 Bit (ROM P/E Mode Entry 0)

This bit is used to place the ROM in P/E mode.
[Setting condition]

- AA01h is written to the FENTRYR register when the FENTRYR register is 0000h.

Note: When entering ROM P/E mode, the instruction fetch address must be transferred to an area other than the ROM so that instruction fetching is not executed to the ROM. Copy necessary instruction code to the internal RAM and jump to the RAM. Note that E2 DataFlash can be rewritten by a program in the ROM.

[Clearing condition]

- AA00h is written to the FENTRYR register.

FENTRYD Bit (E2 DataFlash P/E Mode Entry)

This bit is used to place the E2 DataFlash in P/E mode.
[Setting condition]

- AA80h is written to the FENTRYR register when the FENTRYR register is 0000h.

[Clearing condition]

- AA00h is written to the FENTRYR register.

Downloaded from Arrow.com.

50.4.3 Protection Unlock Register (FPR)

Address(es): FLASH.FPR 007F C180h

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

This write-only register is used to protect the FPMCR register from being rewritten inadvertently when the CPU runs out of control. Writing to the FPMCR register is enabled only when the following procedure is used to access the register.

Procedure to unlock protection

(1) Write A5h to the FPR register.
(2) Write a set value to the FPMCR register.
(3) Write the inverted set value to the FPMCR register.
(4) Write a set value to the FPMCR register again.

When a procedure other than the above is used to write data, the FPSR.PERR flag is set to 1.

50.4.4 Protection Unlock Status Register (FPSR)

Address(es): FLASH.FPSR 007F C184h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	PERR
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PERR	Protect Error Flag	0: No error 1: An error occurs.	R
b7 to b1	—	Reserved	These bits are read as 0.	R

PERR Flag (Protect Error Flag)

When the FPMCR register is not accessed as described in the procedure to unlock protection, data is not written to the register and this flag is set to 1.

[Setting condition]

- The FPMCR register is not accessed as described in the procedure to unlock protection.

[Clearing condition]

- The FPMCR register is accessed according to the procedure to unlock protection described in section 50.4.3, Protection Unlock Register (FPR).

Downloaded from Arrow.com.

50.4.5 Flash P/E Mode Control Register (FPMCR)

Address(es): FLASH.FPMCR 007F C100h

	b7	b6	b5	b4	b3	b2	b1	b0
	FMS2	LVPE	—	FMS1	RPDIS	—	FMS0	—
Value after reset:	0	0	0	0	1	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b1	FMS0	Flash Operating Mode Select 0	FMS2 FMS1 FMS0 0 0 0: ROM/E2 DataFlash read mode 0 1 0: E2 DataFlash P/E mode 0 1 1: Discharge mode 1 1 0 1: ROM P/E mode 1 1 1: Discharge mode 2 Settings other than above are prohibited.	R/W
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3	RPDIS	ROM P/E Disable	0: ROM programming/erasure enabled 1: ROM programming/erasure disabled	R/W
b4	FMS1	Flash Operating Mode Select 1	See the FMS0 bit.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	LVPE	Low-Voltage P/E Mode Enable	0: Low-voltage P/E mode disabled 1: Low-voltage P/E mode enabled	R/W
b7	FMS2	Flash Operating Mode Select 2	See the FMS0 bit.	R/W

The FPMCR register is used to set the operating mode of the flash memory.
This register is protected. Set its value using the procedure to unlock protection. For details, refer to section 50.4.3, Protection Unlock Register (FPR).
When entering discharge mode 2 or ROM P/E mode, or during either of these modes, an instruction must be executed on the RAM.

FMS0, FMS1, and FMS2 Bits (Flash Operating Mode Select 0 to Flash Operating Mode Select 2)

These bits are used to set the operating mode of the flash memory.

- Transition from read mode to ROM P/E mode
  Set the FMS2 bit = 0, the FMS1 bit = 1, the FMS0 bit = 1, and the RPDIS bit = 0.
  Wait for ROM mode transition wait time 1 (tDIS, refer to section 51, Electrical Characteristics).
  Set the FMS2 bit = 1, the FMS1 bit = 1, the FMS0 bit = 1, and the RPDIS bit = 0.
  Set the FMS2 bit = 1, the FMS1 bit = 0, the FMS0 bit = 1, and the RPDIS bit = 0.
  Wait for ROM mode transition wait time 2 (tMS, refer to section 51, Electrical Characteristics).
- Transition from ROM P/E mode to read mode
  Set the FMS2 bit = 1, the FMS1 bit = 1, the FMS0 bit = 1, and the RPDIS bit = 0.
  Wait for ROM mode transition wait time 1 (tDIS, refer to section 51, Electrical Characteristics).
  Set the FMS2 bit = 0, the FMS1 bit = 1, the FMS0 bit = 1, and the RPDIS bit = 0.
  Set the FMS2 bit = 0, the FMS1 bit = 0, the FMS0 bit = 0, and the RPDIS bit = 1.
  Wait for ROM mode transition wait time 2 (tMS, refer to section 51, Electrical Characteristics).
- Transition from read mode to E2 DataFlash P/E mode
  Set the FMS2 bit = 0, the FMS1 bit = 1, the FMS0 bit = 0, and the RPDIS bit = 0.
- Transition from E2 DataFlash P/E mode to read mode
  Set the FMS2 bit = 0, the FMS1 bit = 0, the FMS0 bit = 0, and the RPDIS bit = 1.

Downloaded from Arrow.com.

Wait for ROM mode transition wait time 2 (tMS, refer to section 51, Electrical Characteristics).

RPDIS Bit (ROM P/E Disable)

This bit is used to disable the execution of ROM programming/erasure with software.

LVPE Bit (Low-Voltage P/E Mode Enable)

Set this bit to 0 for programming/erasure in high-speed mode, and set this bit to 1 for programming/erasure in middle-speed mode.

50.4.6 Flash Initial Setting Register (FISR)

Address(es): FLASH.FISR 007F C1D8h



Bit	Symbol	Bit Name	Description	R/W
b4 to b0	PCKA[4:0]	Peripheral Clock Notification	These bits are used to set the frequency of the FlashIF clock (FCLK).	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7, b6	SAS[1:0]	Start-Up Area Select	b7 b6 0 x: The start-up area is selected according to the start-up area settings of the extra area. 1 0: The start-up area is switched to the default area temporarily. 1 1: The start-up area is switched to the alternate area temporarily.	R/W

x: Don't care

Data can be written to the FISR register in ROM P/E mode or E2 DataFlash P/E mode.

PCKA[4:0] Bits (Peripheral Clock Notification)

These bits are used to set the frequency of the FlashIF clock (FCLK) when programming/erasing the ROM/E2 DataFlash.

Set the FCLK frequency in the PCKA[4:0] bits before programming/erasure. Do not change the frequency during programming/erasure of the ROM/E2 DataFlash.

- When FCLK is higher than 4 MHz
  Set a rounded-up value for a non-integer frequency.
  For example, set 32 MHz (PCKA[4:0] bits = 11111b) when the frequency is 31.5 MHz.
- When FCLK is 4 MHz or lower
  Do not use a non-integer frequency.
  Use the FCLK at a frequency of 1, 2, 3, or 4 MHz.

Note: When the PCKA[4:0] bits are set to a frequency different from the FCLK, the data in the ROM/E2 DataFlash may be damaged.

Downloaded from Arrow.com.

Table 50.3 Example of FlashIF Clock Frequency Settings

FlashIF Clock Frequency (MHz)	PCKA[4:0] Bit Setting	FlashIF Clock Frequency (MHz)	PCKA[4:0] Bit Setting	FlashIF Clock Frequency (MHz)	PCKA[4:0] Bit Setting
32	11111b	31	11110b	30	11101b
29	11100b	28	11011b	27	11010b
26	11001b	25	11000b	24	10111b
23	10110b	22	10101b	21	10100b
20	10011b	19	10010b	18	10001b
17	10000b	16	01111b	15	01110b
14	01101b	13	01100b	12	01011b
11	01010b	10	01001b	9	01000b
8	00111b	7	00110b	6	00101b
5	00100b	4	00011b	3	00010b
2	00001b	1	00000b	—	—

SAS[1:0] Bits (Start-Up Area Select)

These bits are used to select the start-up area. To change the start-up area, the following three methods can be used.

- When selecting the start-up area according to the start-up area settings of the extra area
  With the SAS[1:0] bits set to 00b or 01b, the start-up area is selected according to the start-up area settings of the extra area. The settings are enabled after a reset is released.
- When switching the start-up area to the default area temporarily
  When 10b is written to the SAS[1:0] bits, the start-up area is switched to the default area immediately after data is written to the register, regardless of the start-up area settings of the extra area.
  When a reset is generated after this, the area is selected according to the start-up area settings of the extra area.
- When switching the start-up area to the alternative area temporarily
  When 11b is written to the SAS[1:0] bits, the start-up area is switched to the alternative area, regardless of the start-up area settings of the extra area.
  When a reset is generated after this, the area is selected according to the start-up area settings of the extra area.

Downloaded from Arrow.com.

50.4.7 Flash Reset Register (FRESETR)

Address(es): FLASH.FRESETR 007F C124h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	FRESET
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	FRESET	Flash Reset	0: Flash control circuit reset is released. 1: Flash control circuit is reset.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

FRESET Bit (Flash Reset)

When this bit is set to 1, registers FASR, FSARH, FSARL, FEARH, FEARL, FWB0, FWB1, FWB2, FWB3, FCR, and FEXCR are reset. Also, the values of registers FEAMH and FEAML are undefined. Do not access these registers during a reset. To release the reset, set this bit to 0.

Do not write to this register while executing a software command or rewriting the extra area.

50.4.8 Flash Area Select Register (FASR)

Address(es): FLASH.FASR 007F C104h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	EXS
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	EXS	Extra Area Select	0: User area or data area 1: Extra area	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Data can be written to the FASR register in ROM P/E mode or E2 DataFlash P/E mode.

This register is initialized by a reset or setting the FRESETR.FRESET bit to 1.

Data cannot be written to this register while the FRESETR.FRESET bit is 1.

EXS Bit (Extra Area Select)

Set this bit to 1 before issuing a software command (start-up area information program or access window information program) for the extra area. Set this bit to 0 before issuing a software command (program, blank check, block erase, or all-block erase) for the user area.

After issuing a software command, do not change the value until changing it for issuing the next software command.

Downloaded from Arrow.com.

50.4.9 Flash Control Register (FCR)

Address(es): FLASH.FCR 007F C114h

	b7	b6	b5	b4	b3	b2	b1	b0
	OPST	STOP	—	—	CMD[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	CMD[3:0]	Software Command Setting	b3 b0 0 0 0 1: Program 0 0 1 1: Blank check 0 1 0 0: Block erase 0 1 1 0: All-block erase Settings other than above are prohibited.*1	R/W
b5, b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	STOP	Forced Processing Stop	When this bit is set to 1, the processing being executed can be forcibly stopped.	R/W
b7	OPST	Processing Start	0: Processing stops. 1: Processing starts.	R/W

Note 1. This does not include set the FCR register to 00h when the FSTATR1.FRDY flag is 1.

Data can be written to the FCR register when in ROM P/E mode and the ROM can be programmed/erased or in E2 DataFlash P/E mode.
This register is initialized by a reset or setting the FRESETR.FRESET bit to 1. Data cannot be written to this register while the FRESETR.FRESET bit is 1.
Note that this register cannot be initialized by the FRESETR.FRESET bit while a software command is being executed.

CMD[3:0] Bits (Software Command Setting)

These bits are used to set a software command (program, blank check, block erase, or all-block erase).
The function of each command is described below.

- Program
  Write the value set in registers FWB0, FWB1, FWB2, and FWB3 to the address set in registers FSARH and FSARL.
- Blank check
  Check whether there is data in the area from the address set in registers FSARH and FSARL to the address set in registers FEARH and FEARL. Confirm that data is not programmed in the area. This command does not guarantee whether the area remains erased.
- Block erase
  Erase consecutive areas specified in the flash memory by the blocks. Set the beginning address of the block in registers FSARH and FSARL and the end address in registers FEARH and FEARL.
- All-block erase
  Erase all blocks in the ROM or E2 DataFlash.
  All-block erase requires less time to erase the memory compared to block erase. When erasing the whole of the ROM area, set the beginning address of the ROM area in registers FSARH and FSARL, and the end address in registers FEARH and FEARL. Table 50.4 lists the setting address for all-block erase.

Downloaded from Arrow.com.

Table 50.4 Setting Address for All-Block Erase

Target	Memory Size	FSARH/FSARL	FEARH/FEARL
ROM	512 Kbytes	FC18 0000h	FC1F FFF8h
	384 Kbytes	FC1A 0000h	FC1F FFF8h
E2 DataFlash	8 Kbytes	FE00 0000h	FE00 1FFFh

STOP Bit (Forced Processing Stop)

This bit is used to forcibly stop the processing (blank check, block erase, or all-block erase) being executed.
After setting this bit to 1, wait until the FSTATR1.FRDY flag is 1 (processing completed) before setting the OPST bit to 0.

OPST Bit (Processing Start)

This bit is used to execute the command set in the CMD[3:0] bits.
This bit is not set to 0 again even when the processing is completed. Confirm that the FSTATR1.FRDY flag is 1 (processing completed) before setting the OPST bit to 0 again. After that, confirm that the FRDY flag is 0 before executing the next processing.

Downloaded from Arrow.com.

50.4.10 Flash Extra Area Control Register (FEXCR)

Address(es): FLASH.FEXCR 007F C1DCh

	b7	b6	b5	b4	b3	b2	b1	b0
	OPST	—	—	—	—	CMD[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	CMD[2:0]	Software Command Setting	b2 b0 0 0 1: Start-up area information program 0 1 0: Access window information program Settings other than above are prohibited.*1	R/W
b6 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	OPST	Processing Start	0: Processing stops. 1: Processing starts.	R/W

Note 1. This does not include set the FEXCR register to 00h when the FSTATR1.EXRDY flag is 1.

Data can be written to the FEXCR register when in ROM P/E mode and the ROM can be programmed/erased.
This register is initialized by a reset or setting the FRESETR.FRESET bit to 1. Data cannot be written to this register while the FRESETR.FRESET bit is 1.
Note that this register cannot be initialized by the FRESETR.FRESET bit while a software command is being executed.

CMD[2:0] Bits (Software Command Setting)

These bits are used to set a software command (start-up area information program or access window information program).
The details of each command are described below.

- Start-up area information program
  This command is used to switch the start-up area used for start-up program protection.
  When setting the start-up area to the default area, set the FWB1 and FWB0 registers to FFFFh, and execute this command.
  When setting the start-up area to the alternative area, set the FWB1 register to FFFFh, set the FWB0 register to FEFFh, and execute this command.
  When the FWB1 and FWB0 registers are set to values other than the above, do not execute the start-up area information program.
- Access window information program
  This command is used to set the access window used for area protection.
  Set the access window in block units.
  Specify the access window start address, which is the beginning address of the access window in the FWB0 register, specify the access window end address, which is the next address of the last address of the access window in the FWB1 register, and issue this command. Set bit 21 to bit 10 of the address for programming/erasure in each register.
  If the same value is set as the start address and end address, all areas can be accessed. Do not set the start address to a value larger than the value of the end address.

OPST Bit (Processing Start)

This bit is used to execute the command set in the CMD[2:0] bits.
This bit is not set to 0 again even when the processing is completed. Confirm that the FSTATR1.EXRDY flag is 1 (processing completed) before setting the OPST bit to 0 again. After that, confirm that the FSTATR1.EXRDY flag is 0 before executing the next processing.

Downloaded from Arrow.com.

Writing to the extra area is started by writing 1 to the OPST bit. Do not write to the CMD[2:0] bits while a software command is being executed.

50.4.11 Flash Processing Start Address Register H (FSARH)

Address(es): FLASH.FSARH 007F C110h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
								—	—	—	—					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The FSARH register is used to set the target processing address or the start address of the target processing range in the flash memory when a software command is executed.
Set bit 31 to bit 25 and bit 20 to bit 16 of the flash memory address for programming/erasure in this register.
Data can be written to this register in ROM P/E mode or E2 DataFlash P/E mode.
This register is initialized by a reset or setting the FRESETR.FRESET bit to 1. Data cannot be written to this register while the FRESETR.FRESET bit is 1.
If this register is read while executing a software command set by the FEXCR register, an undefined value is read.
Refer to Figure 50.1 and Figure 50.2 for details on the addresses of the flash memory.

50.4.12 Flash Processing Start Address Register L (FSARL)

Address(es): FLASH.FSARL 007F C108h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The FSARL register is used to set the target processing address or the start address of the target processing range in the flash memory when a software command is executed.
Set bit 15 to bit 0 of the flash memory address for programming/erasure in this register.
When the target is the ROM, set bit 2 to bit 0 to 000b.
Data can be written to this register in ROM P/E mode or E2 DataFlash P/E mode.
This register is initialized by a reset or setting the FRESETR.FRESET bit to 1. Data cannot be written to this register while the FRESETR.FRESET bit is 1.
This register is incremented by 8h if the ROM is specified and 1h if the E2 DataFlash is specified after a program command is executed. Therefore, it is not necessary to set the target address to be written to this register when executing a program command sequentially.
If this register is read while executing a software command set by the FEXCR register, an undefined value is read.
Refer to Figure 50.1 and Figure 50.2 for details on the addresses of the flash memory.

Downloaded from Arrow.com.

50.4.13 Flash Processing End Address Register H (FEARH)

Address(es): FLASH.FEARH 007F C120h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
								—	—	—	—					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The FEARH register is used to set the end address of the target processing range in the flash memory when a software command is executed.
Set bit 31 to bit 25 and bit 20 to bit 16 of the flash memory address for programming/erasure in this register.
Data can be written to this register in ROM P/E mode or E2 DataFlash P/E mode.
This register is initialized by a reset or setting the FRESETR.FRESET bit to 1. Data cannot be written to this register while the FRESETR.FRESET bit is 1.
If this register is read while executing a software command set by the FEXCR register, an undefined value is read.
Refer to Figure 50.1 and Figure 50.2 for details on the addresses of the flash memory.

50.4.14 Flash Processing End Address Register L (FEARL)

Address(es): FLASH.FEARL 007F C118h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The FEARH register is used to set the end address of the target range for processing when a software command is executed.
Set bit 15 to bit 0 of the flash memory address for programming/erasure in this register.
When the target is the ROM, set bit 2 to bit 0 to 000b.
Data can be written to this register in ROM P/E mode or E2 DataFlash P/E mode.
This register is initialized by a reset or setting the FRESETR.FRESET bit to 1. Data cannot be written to this register while the FRESETR.FRESET bit is 1.
If this register is read while executing a software command set by the FEXCR register, an undefined value is read.
Refer to Figure 50.1 and Figure 50.2 for details on the addresses of the flash memory.

Downloaded from Arrow.com.

50.4.15 Flash Write Buffer Register n (FWBn) (n = 0 to 3)

Address(es): FLASH.FWB0 007F C130h, FLASH.FWB1 007F C138h, FLASH.FWB2 007F C140h, FLASH.FWB3 007F C144h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is used to set the data for programming the ROM, E2 DataFlash, or extra area. The data can be written in ROM P/E mode or E2 DataFlash P/E mode.
This register is initialized by a reset or setting the FRESETR.FRESET bit to 1. Data cannot be written to this register while the FRESETR.FRESET bit is 1.
The read value of this register is undefined while executing a software command set by the FCR register or the FEXCR register.
When programming the extra area, set the 4-byte data for programming in registers FWB0 and FWB1.
When programming the E2 DataFlash, set the data for programming in the lower 8 bits in the FWB0 register.
When programming the ROM, set the 8-byte data for programming in registers FWB0 to FWB3. Figure 50.3 shows the relationship between the addresses indicated by registers FSARH and FSARL and the data set in the FWBn register.

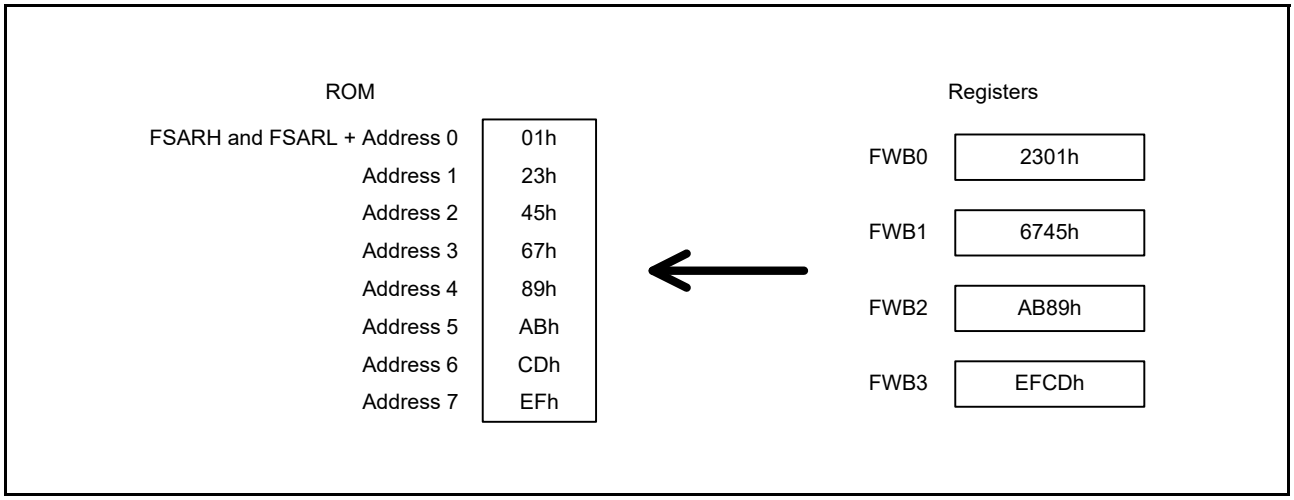


Figure 50.3 FWBn Register Setting Values and Data Allocation in the ROM

Downloaded from Arrow.com.

50.4.16 Flash Status Register 0 (FSTATR0)

Address(es): FLASH.FSTATR0 007F C1F0h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	EILGLERR	ILGLERR	BCERR	—	PRGERR	ERERR
Value after reset:	x	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ERERR	Erase Error Flag	0: Erasure terminates normally. 1: An error occurs during erasure.	R
b1	PRGERR	Program Error Flag	0: Programming terminates normally. 1: An error occurs during programming.	R
b2	—	Reserved	The read value is undefined.	R
b3	BCERR	Blank Check Error Flag	0: Blank checking terminates normally. 1: An error occurs during blank checking.	R
b4	ILGLERR	Illegal Command Error Flag	0: No illegal software command or illegal access is detected. 1: An illegal command or illegal access is detected.	R
b5	EILGLERR	Extra Area Illegal Command Error Flag	0: No illegal command or illegal access to the extra area is detected. 1: An illegal command or illegal access to the extra area is detected.	R
b7, b6	—	Reserved	The read value is undefined.	R

This register is a status register used to confirm the result of executing a software command. Each error flag is set to 0 when the next software command is executed.

ERERR Flag (Erase Error Flag)

This flag indicates the result of the erase processing for the ROM/E2 DataFlash.
[Setting condition]

- An error occurs during erasure.

[Clearing condition]

- The next software command is executed.
  The value read from this flag is undefined when the FCR.STOP bit is set to 1 (processing is forcibly stopped) during erasure.

PRGERR Flag (Program Error Flag)

This flag indicates the result of the program processing for the ROM/E2 DataFlash.
[Setting condition]

- An error occurs during programming.

[Clearing condition]

- The next software command is executed.

BCERR Flag (Blank Check Error Flag)

This flag indicates the result of the blank check processing for the ROM/E2 DataFlash.
[Setting condition]

- An error occurs during blank checking.

Downloaded from Arrow.com.

[Clearing condition]

- The next software command is executed.
  The value read from this flag is undefined when the FCR.STOP bit is set to 1 (processing is forcibly stopped) during blank checking.

ILGLERR Flag (Illegal Command Error Flag)

This flag indicates the result of executing a software command.
[Setting conditions]

- Programming/erasure is executed to an area other than the access window range.
- A blank check or block erase command is executed when the set value of registers FSARH and FSARL is larger than the set value of registers FEARH and FEARL.
- Program and block erase commands are executed when the FASR.EXS bit is 1.
- An all-block erase command is executed while the access window is set.
- An all-block erase command is executed without setting registers FSARH and FSARL and registers FEARH and FEARL properly.
- The E2 DataFlash address is set in registers FSARH and FSARL and a software command is executed when the ROM is in P/E mode.
- The ROM address is set in registers FSARH and FSARL and a software command is executed when the E2 DataFlash is in P/E mode.
- The ROM and E2 DataFlash are set to P/E mode and a software command is executed.

[Clearing condition]

- The next software command is executed.

EILGLERR Flag (Extra Area Illegal Command Error Flag)

This flag indicates the result of executing a software command for the extra area.
[Setting condition]

- A software command for the extra area is executed when the FASR.EXS bit is 0.

[Clearing condition]

- The next software command is executed.

Downloaded from Arrow.com.

50.4.17 Flash Status Register 1 (FSTATR1)

Address(es): FLASH.FSTATR1 007F C12Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	EXRDY	FRDY	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	1	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	—	Reserved	These bits are read as 0.	R
b2	—	Reserved	This bit is read as 1.	R
b5 to b3	—	Reserved	These bits are read as 0.	R
b6	FRDY	Flash Ready Flag	0: Other than below 1: 00h can be written to the FCR register (processing to complete the software command).	R
b7	EXRDY	Extra Area Ready Flag	0: Other than below 1: 00h can be written to the FEXCR register (processing to complete the software command).	R

This register is a status register used to confirm the result of executing a software command. Each flag is set to 0 when the next software command is executed.

FRDY Flag (Flash Ready Flag)

This flag is used to confirm whether a software command is executed.
This flag becomes 1 when processing of the executed software command or the forced stop processing is completed, and this flag becomes 0 when setting the FCR.OPST bit to 0.
Also, an interrupt (FRDYI) is generated when this flag becomes 1.

EXRDY Flag (Extra Area Ready Flag)

This flag is used to confirm whether a software command for the extra area is executed.
This flag is set to 1 when processing of the executed software command is completed, and 0 when the FEXCR.OPST bit is set to 0.
Also, an interrupt (FRDYI) is generated when this flag becomes 1.

Downloaded from Arrow.com.

50.4.18 Flash Error Address Monitor Register H (FEAMH)

Address(es): FLASH.FEAMH 007F C1E8h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
								—	—	—	—					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is used to check the address where the error has occurred if an error occurs during processing of a software command. This register stores bit 31 to bit 25 and bit 20 to bit 16 of the address where the error has occurred for the program command or blank check command, or it stores bit 31 to bit 25 and bit 20 to bit 16 of the beginning address of the area where the error has occurred for the block erase command or all-block erase command.
Since this register value becomes undefined if setting the FRESETR.FRESET bit to 1, read the value before error processing.
If the software command terminates normally, this register stores bit 31 to bit 25 and bit 20 to bit 16 of the end address at execution of the command.
Refer to Figure 50.1 and Figure 50.2 for details on the addresses of the flash memory.

50.4.19 Flash Error Address Monitor Register L (FEAML)

Address(es): FLASH.FEAML 007F C1E0h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is used to check the address where the error has occurred if an error occurs during processing of a software command. This register stores bit 15 to bit 0 of the address where the error has occurred for the program command or blank check command, or it stores bit 15 to bit 0 of the beginning address of the area where the error has occurred for the block erase command or all-block erase command.
Since this register value becomes undefined if setting the FRESETR.FRESET bit to 1, read the value before error processing.
When the software command is normally completed, this register stores bit 15 to bit 0 of the last address at execution of the command.
When executing a software command for the ROM, lower 3 bits become 000b.
Refer to Figure 50.1 and Figure 50.2 for details on the addresses of the flash memory.

Downloaded from Arrow.com.

50.4.20 Flash Start-Up Setting Monitor Register (FSCMR)

Address(es): FLASH.FSCMR 007F C1C0h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	SASMF	—	—	—	—	—	—	—	—
Value after reset:	0	1	1	1	0	1	1	Value set by user*1	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	Reserved	These bits are read as 0.	R
b8	SASMF	Start-Up Area Setting Monitor Flag	0: Setting to start up using the alternative area 1: Setting to start up using the default area	R
b10, b9	—	Reserved	These bits are read as 1. Writing to these bits has no effect.	R
b11	—	Reserved	This bit is read as 0. Writing to this bit has no effect.	R
b14 to b12	—	Reserved	These bits are read as 1. Writing to these bits has no effect.	R
b15	—	Reserved	This bit is read as 0. Writing to this bit has no effect.	R

Note 1. The value of the blank product is 1. It is set to the same value set in bit 8 in the FWB0 register after the start-up area information program command is executed.

SASMF Flag (Start-Up Area Setting Monitor Flag)

This flag is used to confirm the settings of the start-up area.

When this flag is 0, the user program is set to start up using the alternative area.

When this flag is 1, the user program is set to start up using the default area.

50.4.21 Flash Access Window Start Address Monitor Register (FAWSMR)

Address(es): FLASH.FAWSMR 007F C1C8h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—												
Value after reset:	0	0	0	0	The value set by the user*1											

Note 1. The value of the blank product is 1. It is set to the same value set in bit 11 to bit 0 the FWB0 register after the access window information program command is executed.

This register is used to confirm the set value of the access window start address used for area protection.

Downloaded from Arrow.com.

50.4.22 Flash Access Window End Address Monitor Register (FAWEMR)

Address(es): FLASH.FAWEMR 007F C1D0h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—												
Value after reset:	0	0	0	0	The value set by the user*1											

Note 1. The value of the blank product is 1. It is set to the same value set in bit 11 to bit 0 in the FWB1 register after the access window information program command is executed.

This register is used to confirm the set value of the access window end address used for area protection.

50.4.23 Unique ID Register n (UIDRn) (n = 0 to 3)

Address(es): FLASHCONST.UIDR0 007F C350h, FLASHCONST.UIDR1 007F C354h, FLASHCONST.UIDR2 007F C358h, FLASHCONST.UIDR3 007F C35Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	Unique value for each chip															

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	Unique value for each chip															

The UIDRn register stores a 16-byte ID code (unique ID) for identifying the individual MCU.
The unique ID is stored in the extra area of the flash memory and cannot be rewritten by the user.

Downloaded from Arrow.com.

50.5 Start-Up Program Protection

When rewriting the start-up program*1 by self-programming, if the rewrite operation is interrupted due to temporary blackout, the start-up program may not be successfully programmed and the user program may not start properly. This problem can be avoided by rewriting the start-up program without erasing the existing start-up program using the start-up program protection. This function is available in products with a 32-Kbyte or larger ROM.
Figure 50.4 shows the overview of the start-up program protection. In this figure, the default area indicates block 0 to block 7, and the alternate area indicates block 8 to block 15.

Note 1. Program to perform operation to start the user program. It includes the fixed vector table.

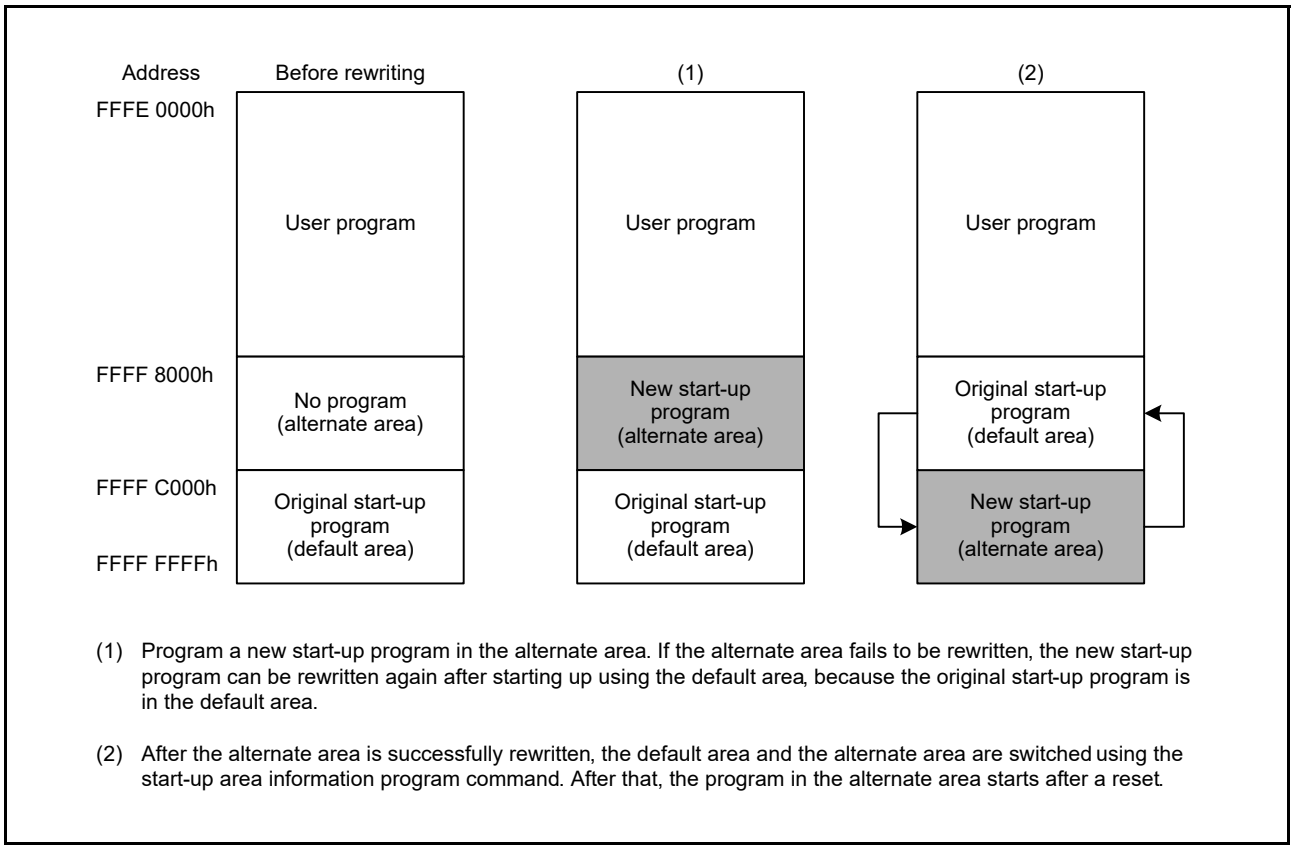


Figure 50.4 Overview of the Start-Up Program Protection

Downloaded from Arrow.com.

50.6 Area Protection

Area protection enables rewriting only the selected blocks (access window) in the user area and disables rewriting the other blocks during self-programming. The access window cannot be set in the data area.
Specify the start address and end address to set the access window. While the access window can be set in boot mode or by self-programming, area protection is enabled only during self-programming in single-chip mode.
Figure 50.5 shows the overview of the area protection.

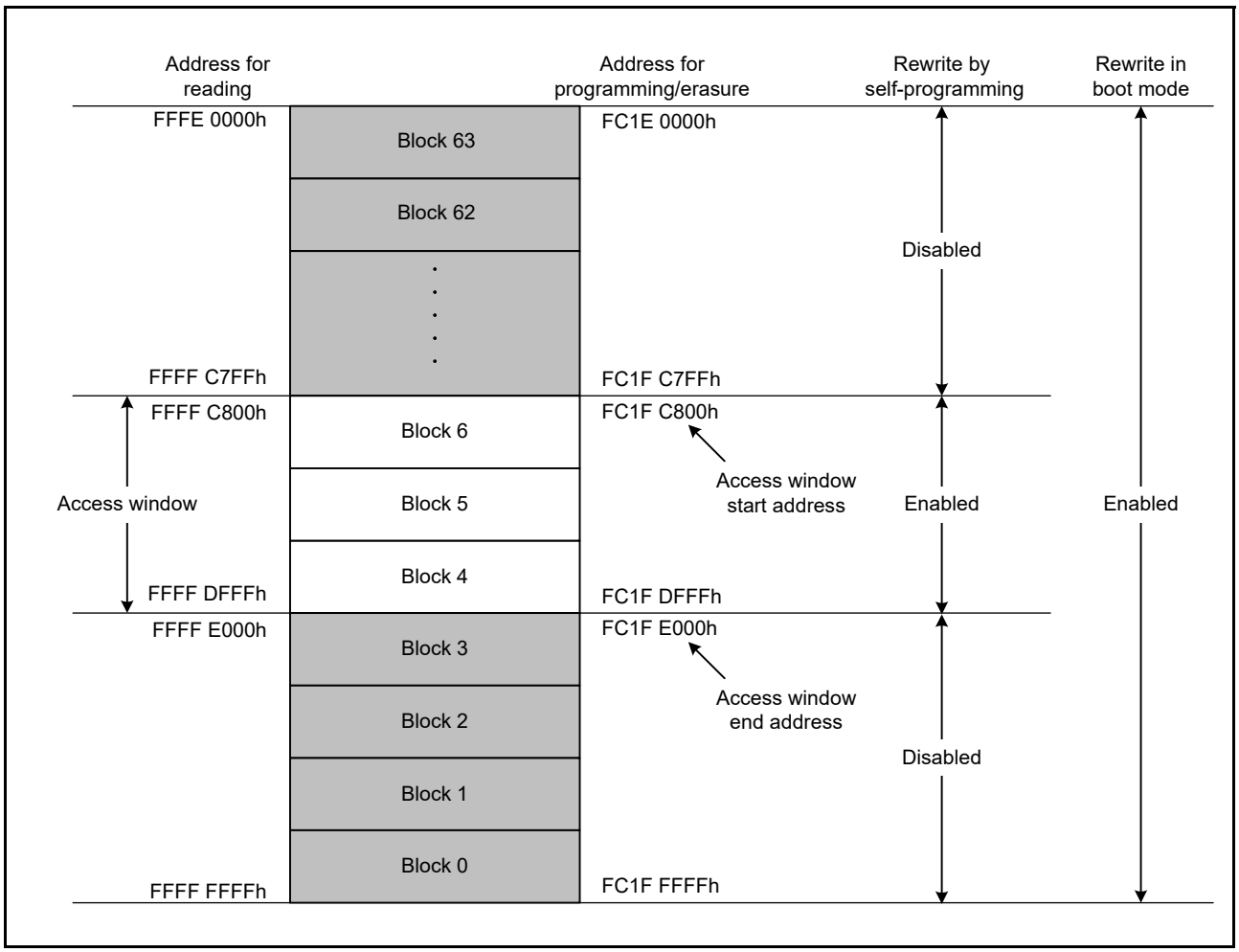


Figure 50.5 Area Protection Overview (When Blocks 4 to 6 are Set as the Access Window in Products with 128-Kbyte ROM)

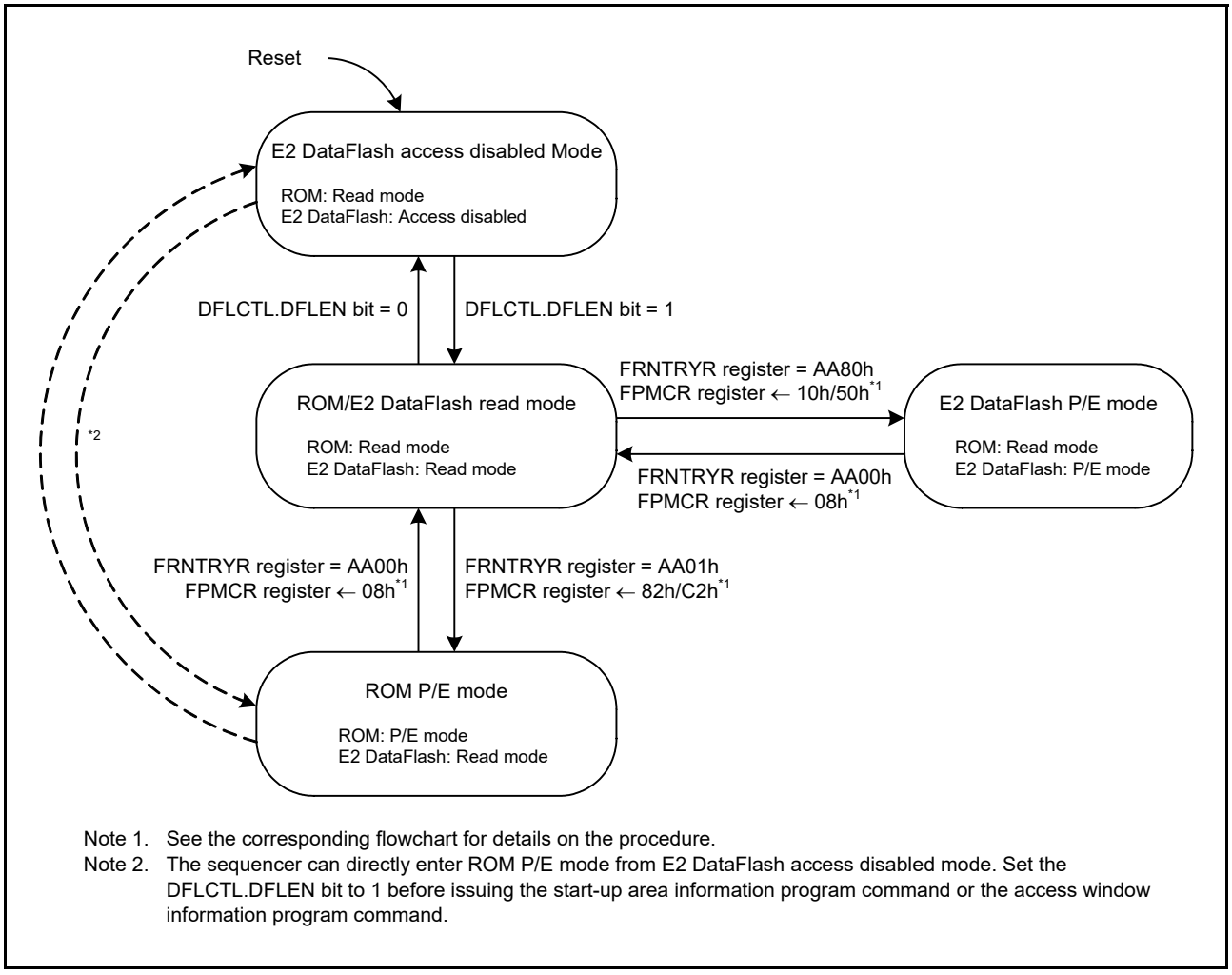
Downloaded from Arrow.com.

50.7 Programming and Erasure

The ROM and E2 DataFlash can be programmed and erased by changing the mode of the dedicated sequencer for programming and erasure, and by issuing commands for programming and erasure.
The mode transitions and commands required to program or erase the ROM and E2 DataFlash are described below. The descriptions apply in common to boot mode and single-chip mode.

50.7.1 Sequencer Modes

The sequencer has four modes. Transitions between modes are caused by writing to the DFLCTL and FENTRYR registers and setting the FPMCR register. Figure 50.6 is a diagram of mode transitions of the flash memory.



Note 1. See the corresponding flowchart for details on the procedure.
Note 2. The sequencer can directly enter ROM P/E mode from E2 DataFlash access disabled mode. Set the DFLCTL.DFLEN bit to 1 before issuing the start-up area information program command or the access window information program command.

Figure 50.6 Mode Transitions of the Flash Memory

50.7.1.1 E2 DataFlash Access Disabled Mode

In E2 DataFlash access disabled mode, access to the E2 DataFlash is disabled. After a reset, the sequencer enters this mode.
When setting the DFLCTL.DFLEN bit to 1, the E2 DataFlash is placed in read mode.

Downloaded from Arrow.com.

50.7.1.2 Read Mode

Read mode is for high-speed reading of the ROM/E2 DataFlash. Reading from a ROM address for reading can be accomplished in one ICLK clock.

(1) ROM/E2 DataFlash Read Mode

In this mode, both the ROM and E2 DataFlash are in read mode. The sequencer enters this mode from P/E mode when setting the FPMCR register to 08h, setting the FENTRYR.FENTRYD bit to 0, and setting the FENTRYR.FENTRY0 bit to 0.

50.7.1.3 P/E Modes

The P/E mode is for programming and erasure of the ROM/E2 DataFlash.

(1) ROM P/E Mode

In this mode, the ROM is in P/E mode, and the E2 DataFlash is in read mode. The sequencer enters this mode when setting the FENTRYR.FENTRYD to 0, setting the FENTRYR.FENTRY0 bit to 1, and setting the FPMCR register 82h or C2h.

(2) E2 DataFlash P/E Mode

In this mode, the ROM is in read mode, and the E2 DataFlash is in P/E mode. The sequencer enters this mode when the setting the FENTRYR.FENTRYD to 1, setting the FENTRYR.FENTRY0 bit to 0, and setting the FPMCR register 10h or 50h.

50.7.2 Mode Transitions

50.7.2.1 Transition from E2 DataFlash Access Disable Mode to Read Mode

Reading of the E2 DataFlash requires switching from E2 DataFlash access disabled mode to ROM/E2 DataFlash read mode.

Set the DFLCTL.DFLEN bit to 1 to switch to ROM/E2 DataFlash read mode.

Figure 50.7 shows the procedure for transition from E2 DataFlash access disabled mode to ROM/E2 DataFlash read mode.

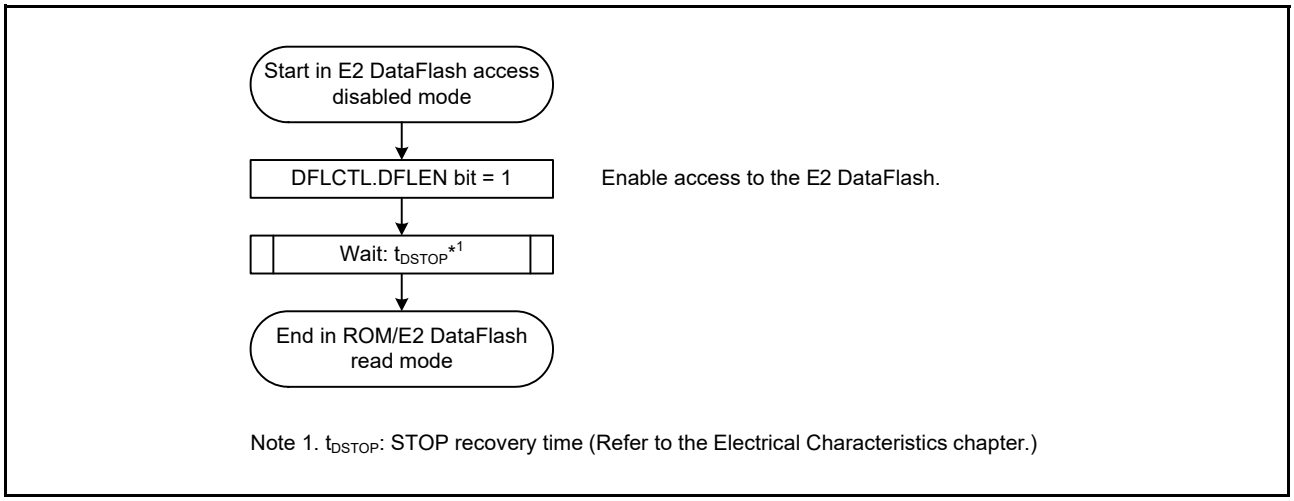
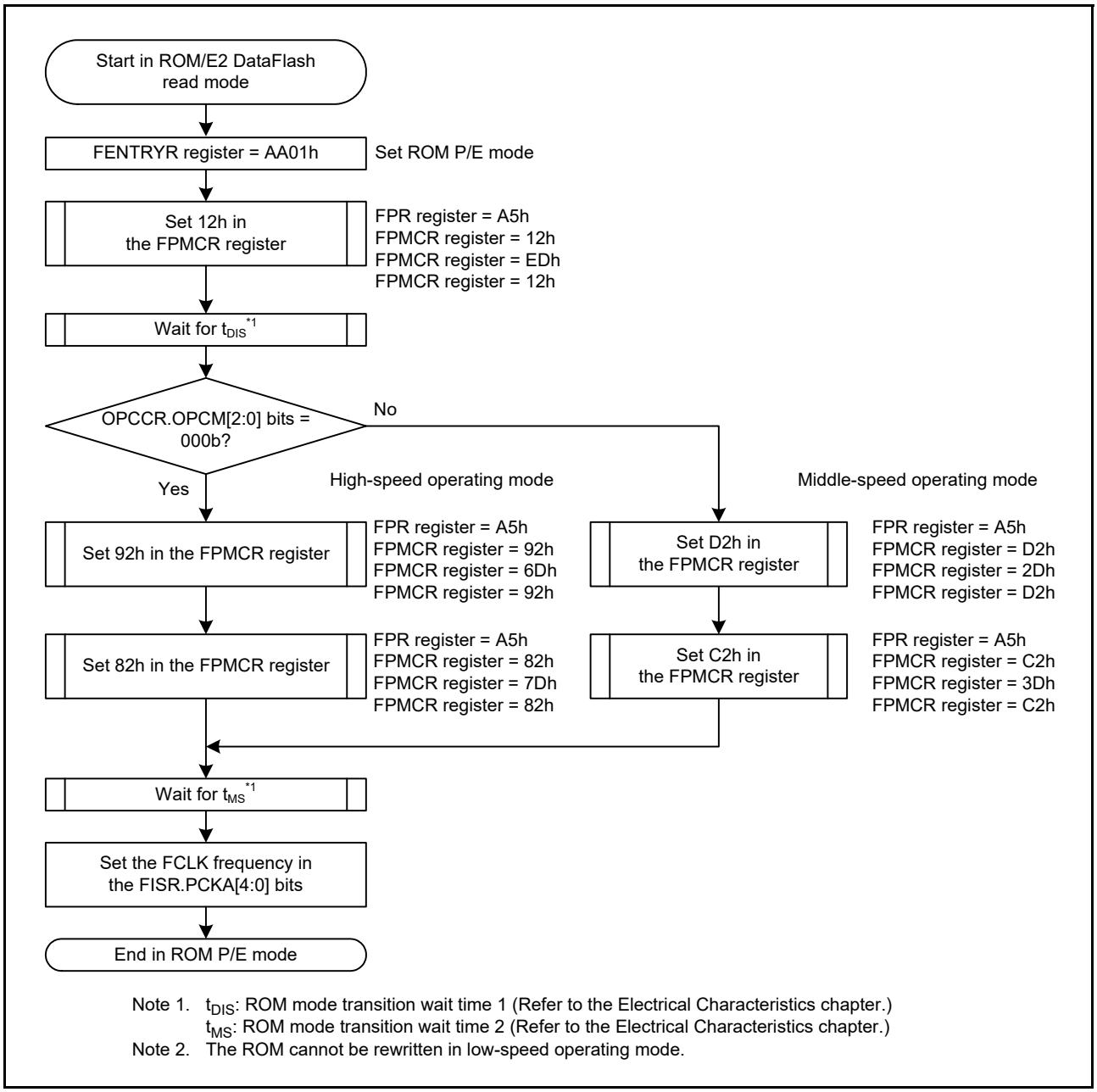


Figure 50.7 Procedure for Transition from E2 DataFlash Access Disabled Mode to ROM/E2 DataFlash Read Mode

Downloaded from Arrow.com.

50.7.2.2 Transition from Read Mode to P/E Mode

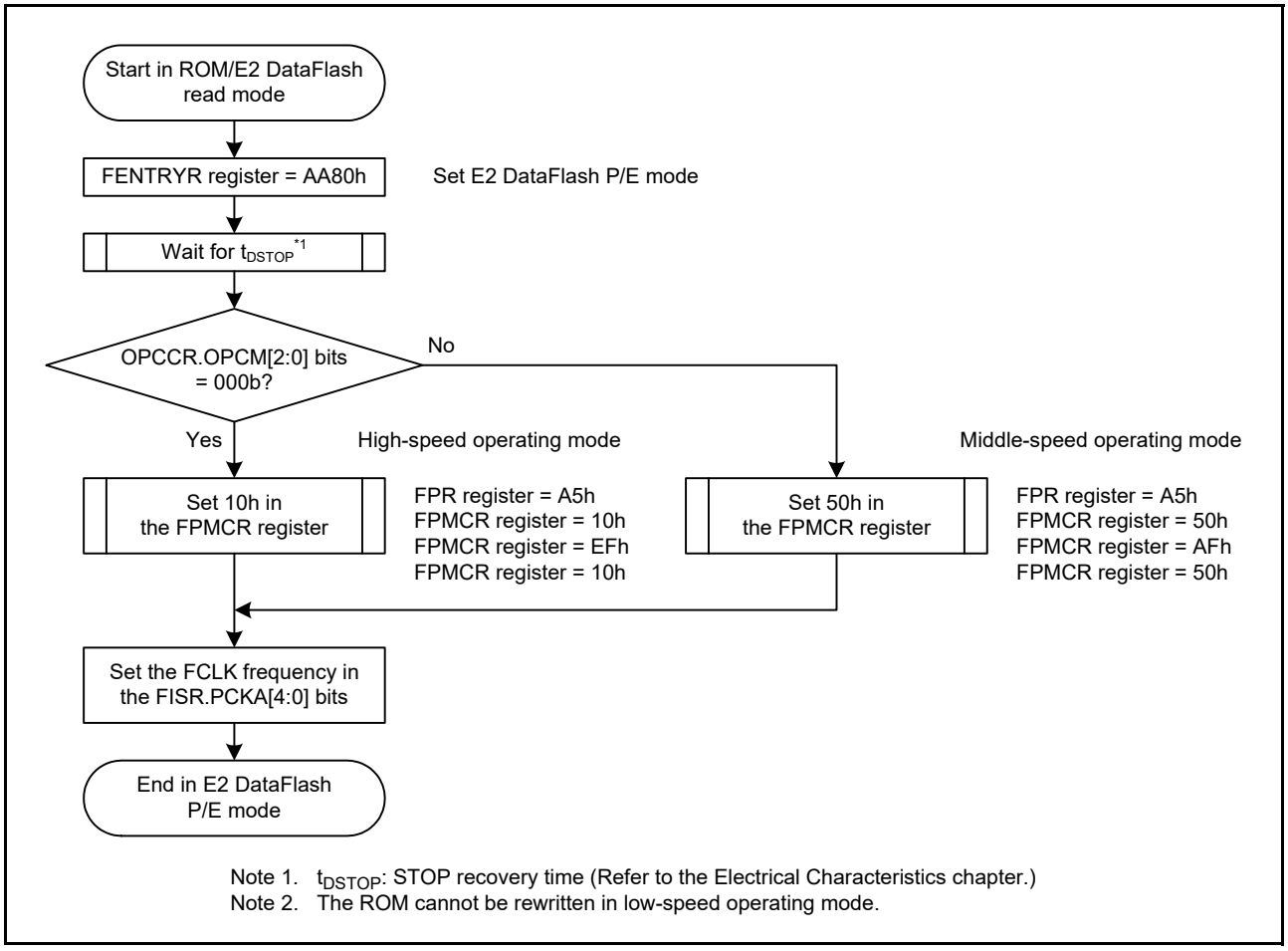
Switching to ROM P/E mode is required before executing a software command for the ROM.
Figure 50.8 shows the procedure for transition from ROM/E2 DataFlash read mode to ROM P/E mode. Figure 50.9 shows the procedure for transition from ROM/E2 DataFlash read mode to E2 DataFlash P/E mode.



Note 1. t_{DIS}: ROM mode transition wait time 1 (Refer to the Electrical Characteristics chapter.)
t_{MS}: ROM mode transition wait time 2 (Refer to the Electrical Characteristics chapter.)
Note 2. The ROM cannot be rewritten in low-speed operating mode.

Figure 50.8 Procedure for Transition from ROM/E2 DataFlash Read Mode to ROM P/E Mode

Downloaded from Arrow.com.



Note 1. t_{DSTOP}: STOP recovery time (Refer to the Electrical Characteristics chapter.)
Note 2. The ROM cannot be rewritten in low-speed operating mode.

Figure 50.9 Procedure for Transition from ROM/E2 DataFlash Read Mode to E2 DataFlash P/E Mode

Downloaded from Arrow.com.

50.7.2.3 Transition from P/E Mode to Read Mode

High-speed reading of the ROM requires switching to ROM/E2 DataFlash read mode.
Figure 50.10 shows the procedure for transition from ROM P/E mode to ROM/E2 DataFlash read mode. Figure 50.11 shows the procedure for transition from E2 DataFlash P/E mode to ROM/E2 DataFlash read mode.

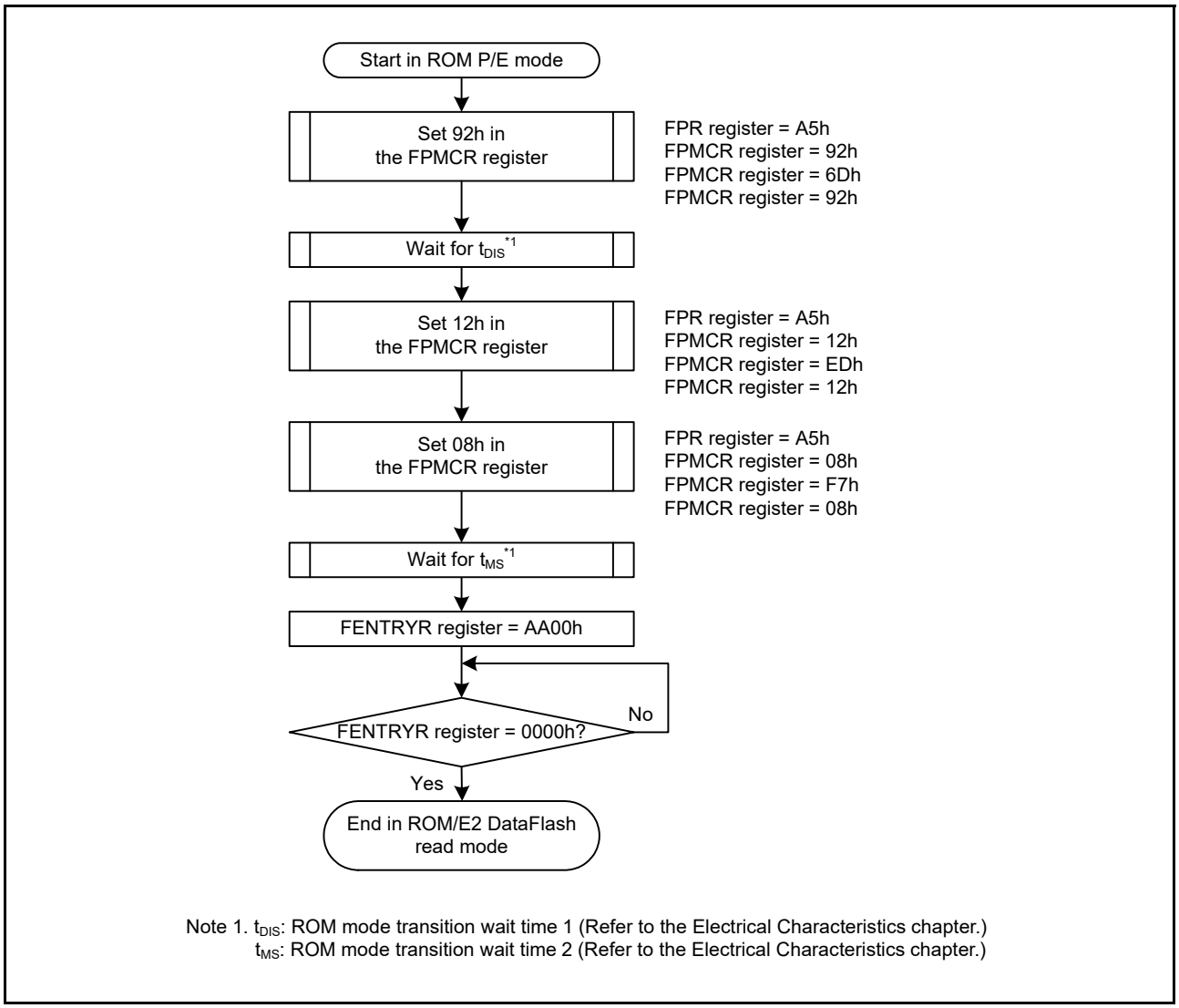
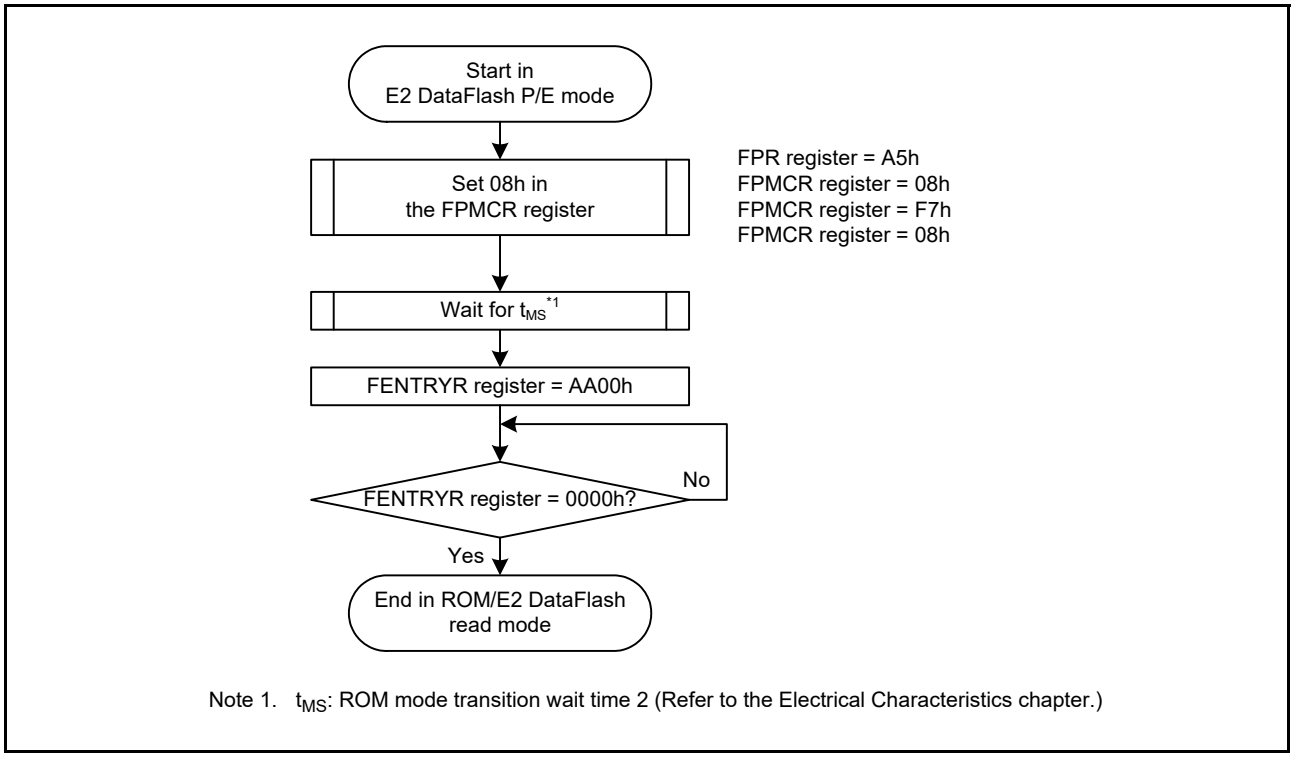


Figure 50.10 Procedure for Transition from ROM P/E Mode to ROM/E2 DataFlash Read Mode

Downloaded from Arrow.com.



Note 1. t_{MS}: ROM mode transition wait time 2 (Refer to the Electrical Characteristics chapter.)

Figure 50.11 Procedure for Transition from E2 DataFlash P/E Mode to ROM/E2 DataFlash Read Mode

Downloaded from Arrow.com.

50.7.3 Software Commands

Software commands consist of commands for programming and erasure and commands for programming start-up program area information and access window information. Table 50.5 lists the software commands for use with the flash memory.

Table 50.5 Software Commands

Command	Function
Program	• ROM programming (8 bytes) • E2 DataFlash programming (1 byte)
Block erase	ROM/E2 DataFlash erasure
All-block erase	Erasure of all blocks in the ROM/E2 DataFlash
Blank check	Check whether the specified area is blank. Confirm that data is not programmed in the area. This command does not guarantee whether the area remains erased.
Start-up area information program	Rewrite the start-up area switching information used for start-up program protection.
Access window information program	Set the access window used for area protection.

Downloaded from Arrow.com.

50.7.4 Software Command Usage

This section describes how to use each software command, using flowcharts.

50.7.4.1 Program

Figure 50.12 and Figure 50.13 show the procedure to issue the program command.

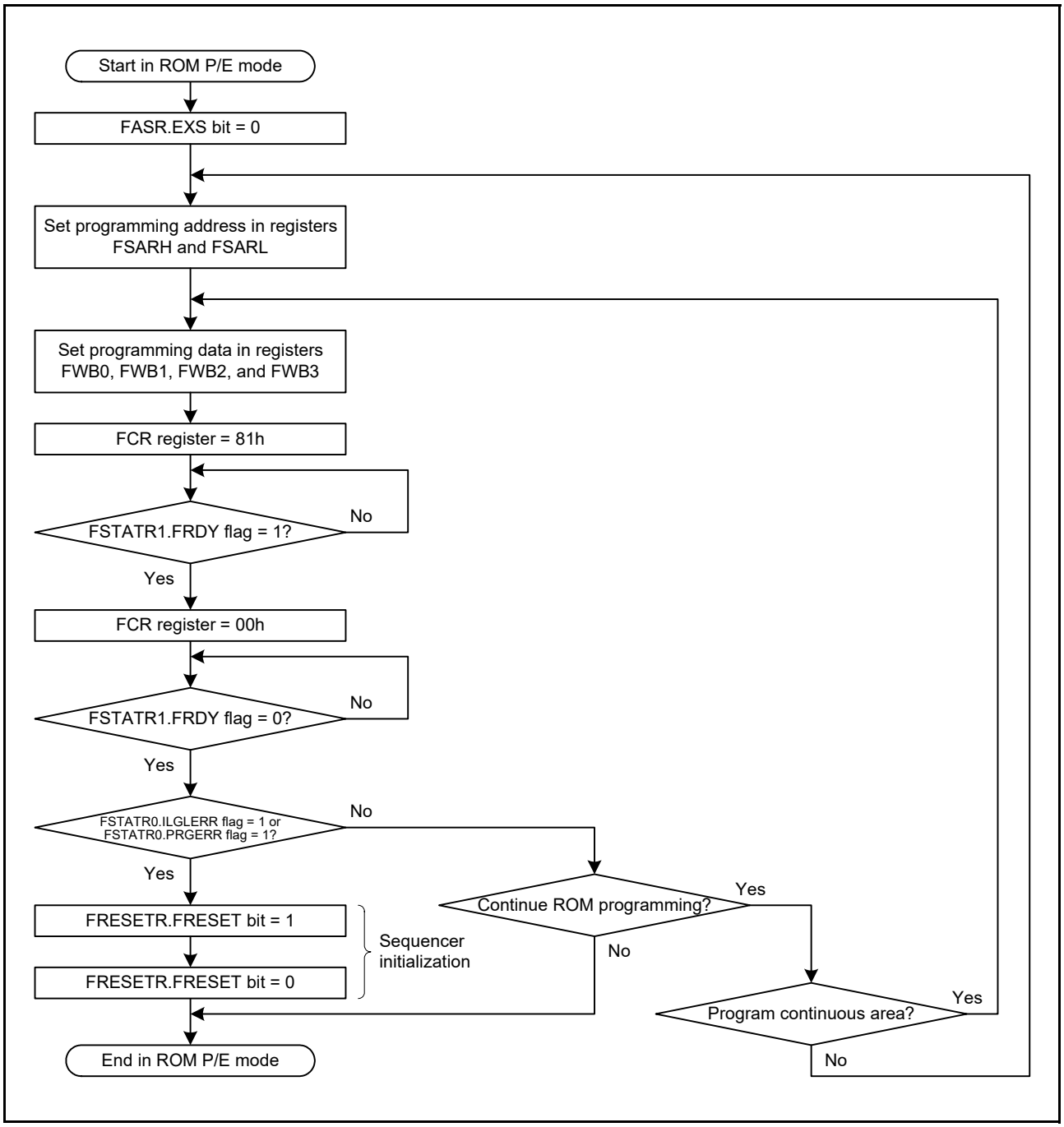


Figure 50.12 Procedure to Issue the Program Command for the ROM

Downloaded from Arrow.com.

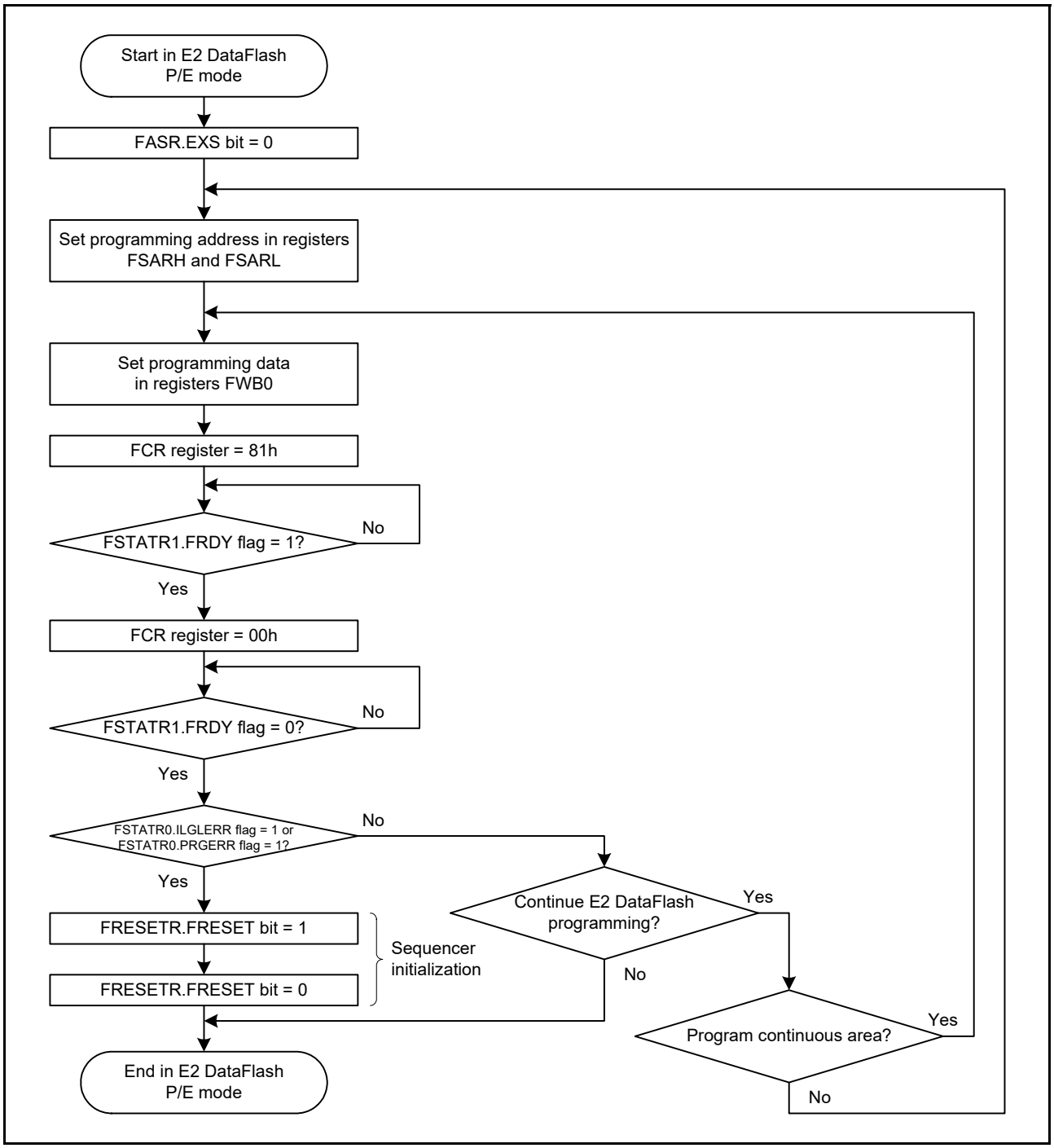


Figure 50.13 Procedure to Issue the Program Command for the E2 DataFlash

Downloaded from Arrow.com.

50.7.4.2 Block Erase

Figure 50.14 and Figure 50.15 show the procedure to issue the block erase command.

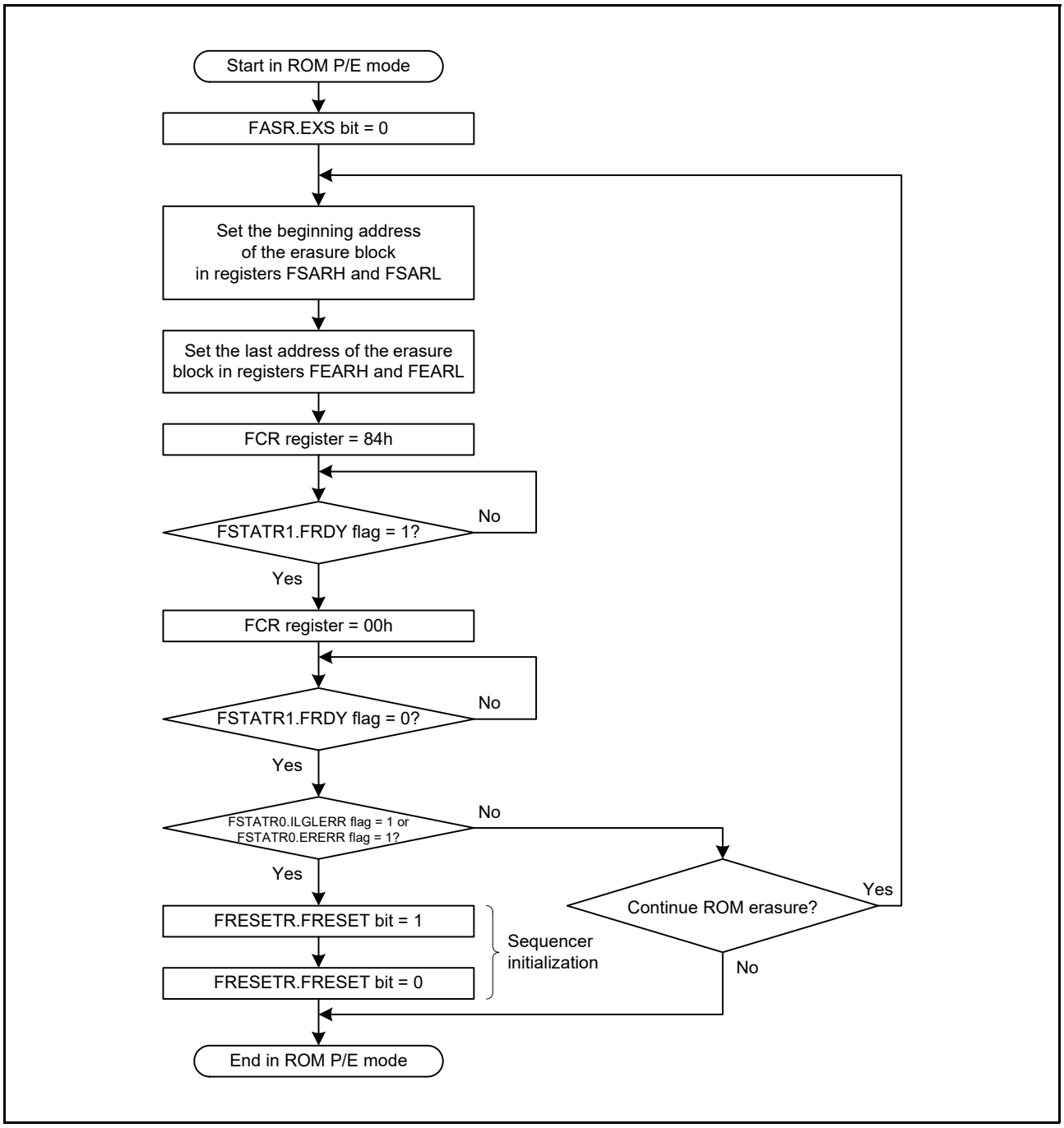


Figure 50.14 Procedure to Issue the Block Erase Command for the ROM

Downloaded from Arrow.com.

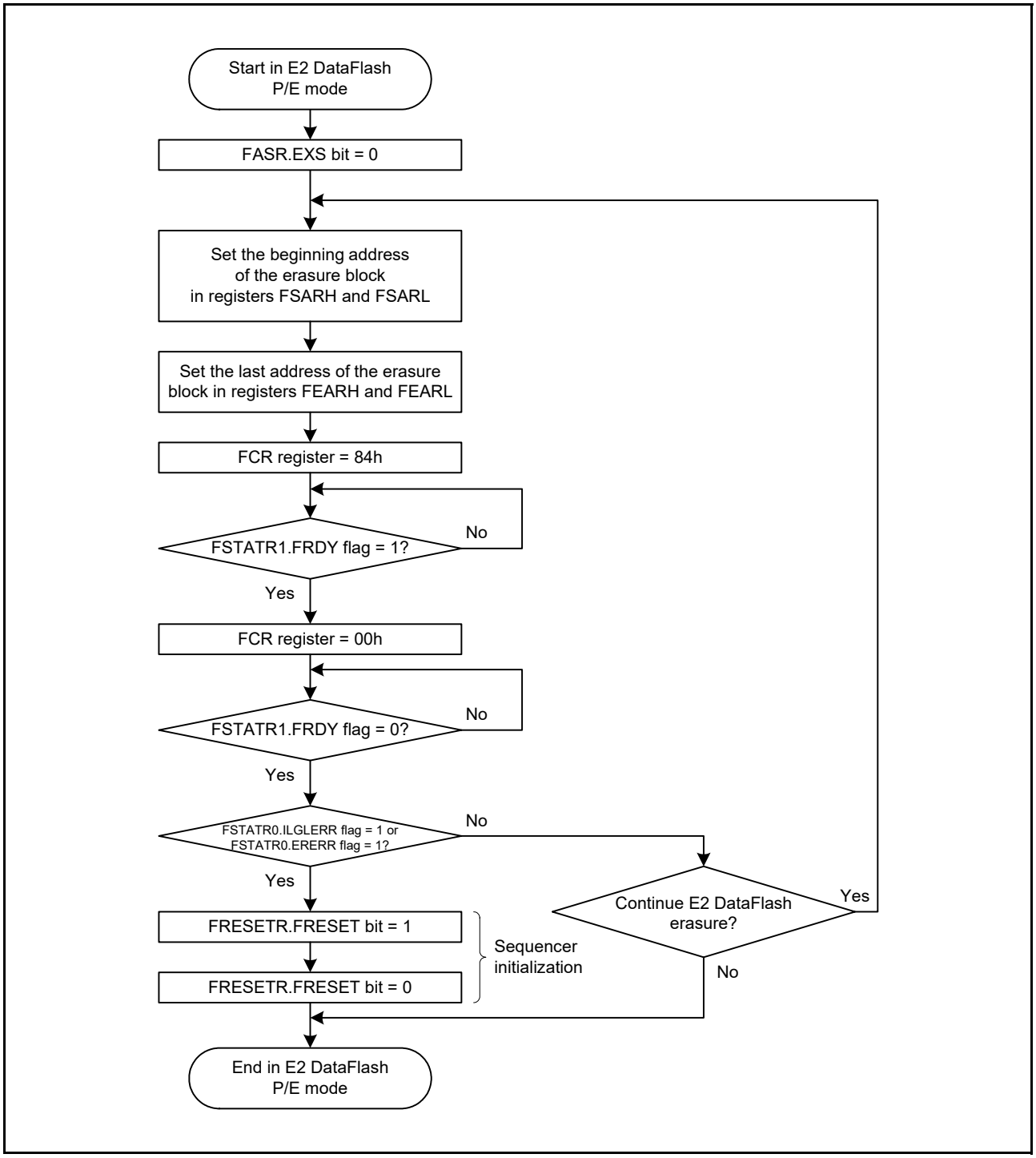


Figure 50.15 Procedure to Issue the Block Erase Command for the E2 DataFlash

Downloaded from Arrow.com.

50.7.4.3 All-Block Erase

Figure 50.16 and Figure 50.17 show the procedure to issue the all-block erase command.

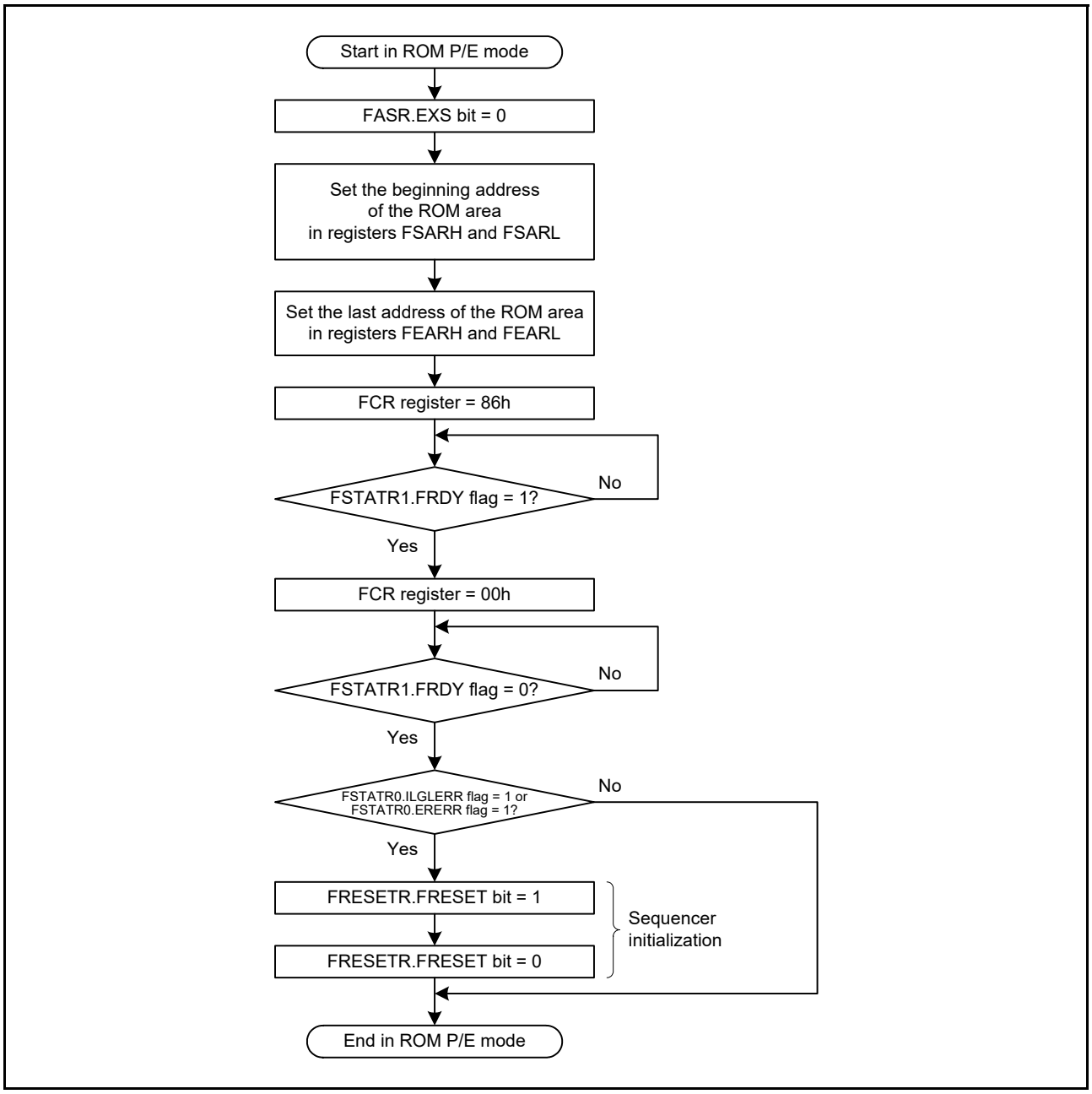


Figure 50.16 Procedure to Issue the All-Block Erase Command for the ROM

Downloaded from Arrow.com.

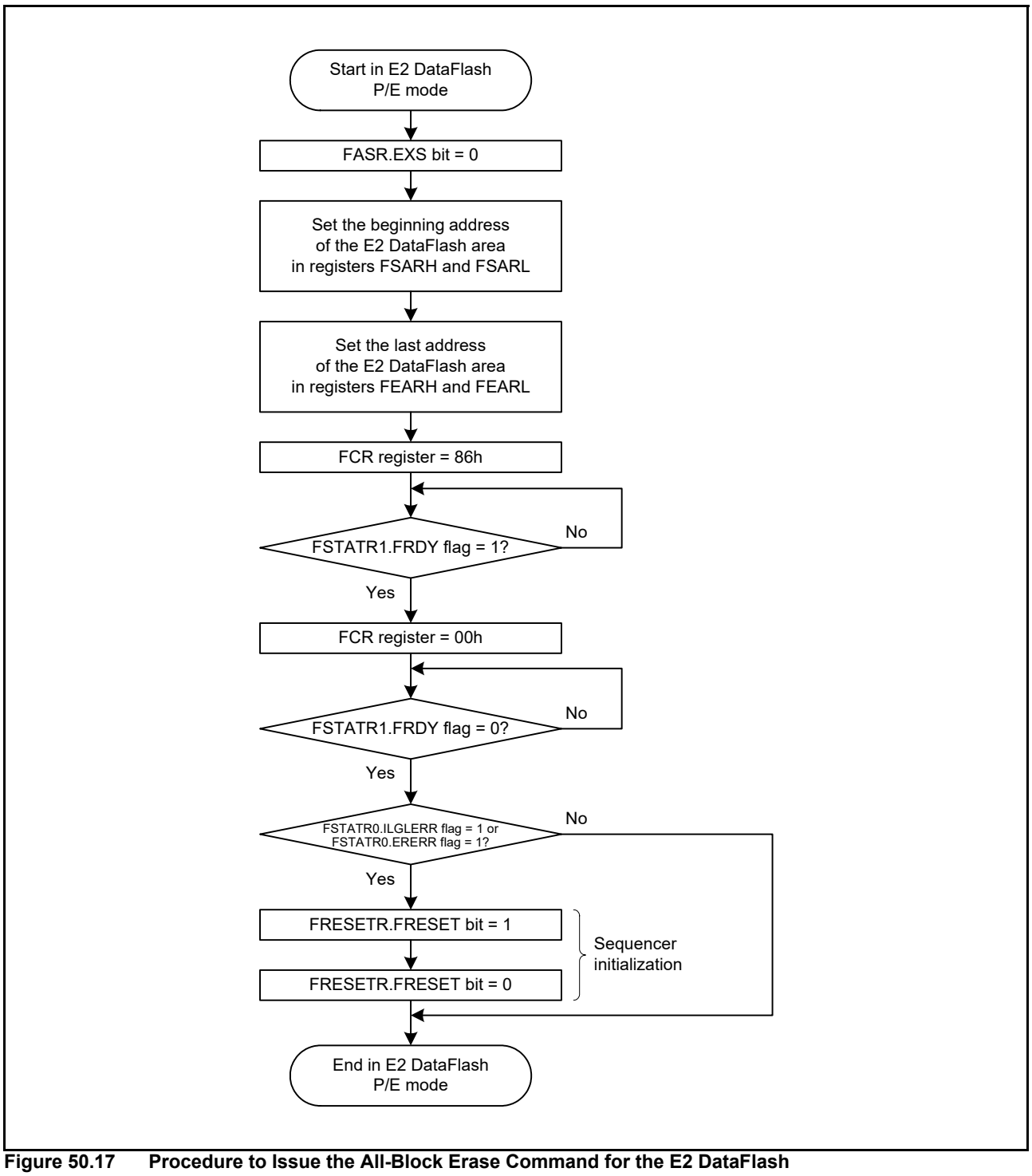


Figure 50.17 Procedure to Issue the All-Block Erase Command for the E2 DataFlash

Downloaded from Arrow.com.

50.7.4.4 Blank Check

Figure 50.18 and Figure 50.19 show the procedure to issue the blank check command.

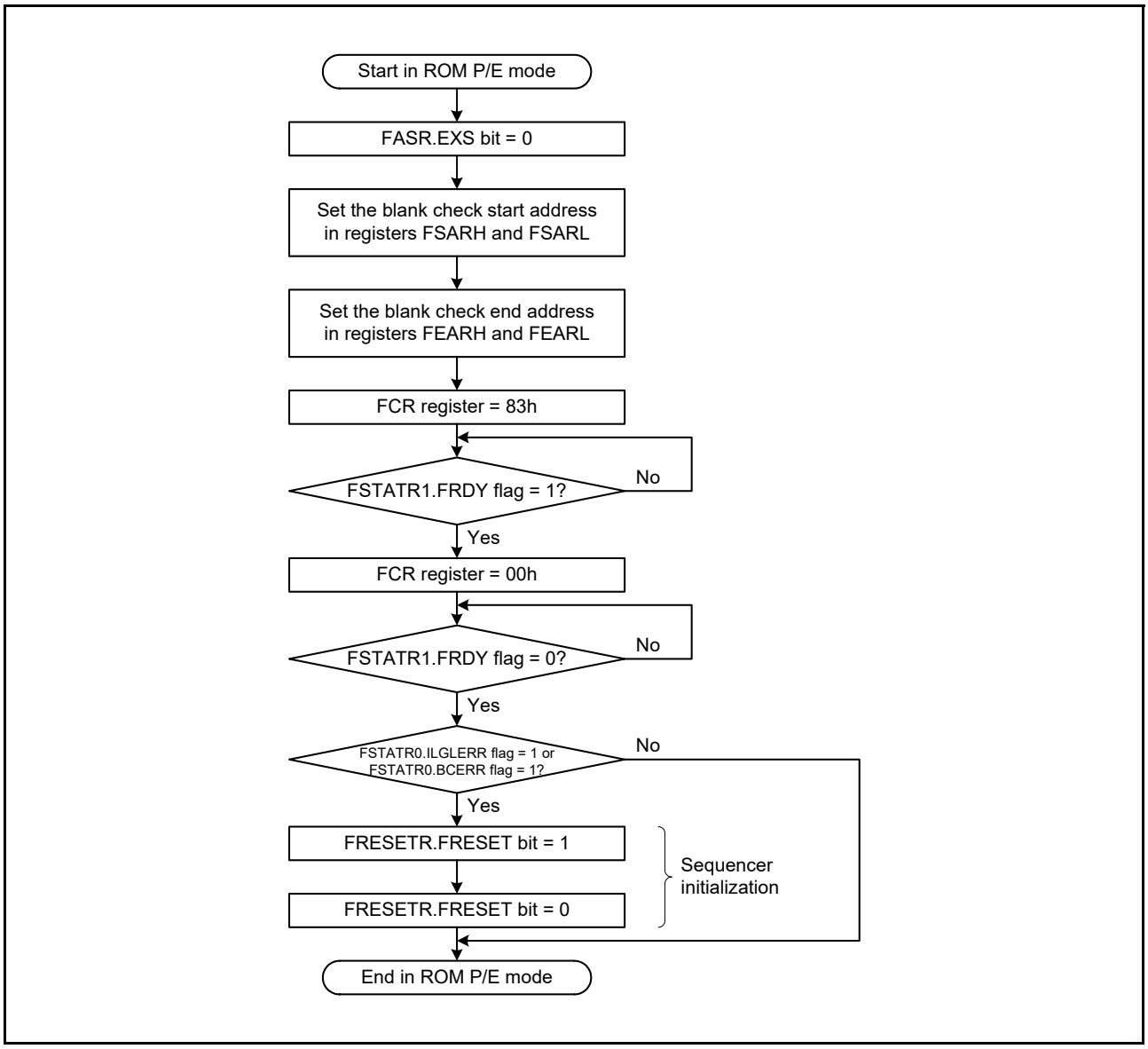


Figure 50.18 Procedure to Issue the Blank Check Command for the ROM

Downloaded from Arrow.com.

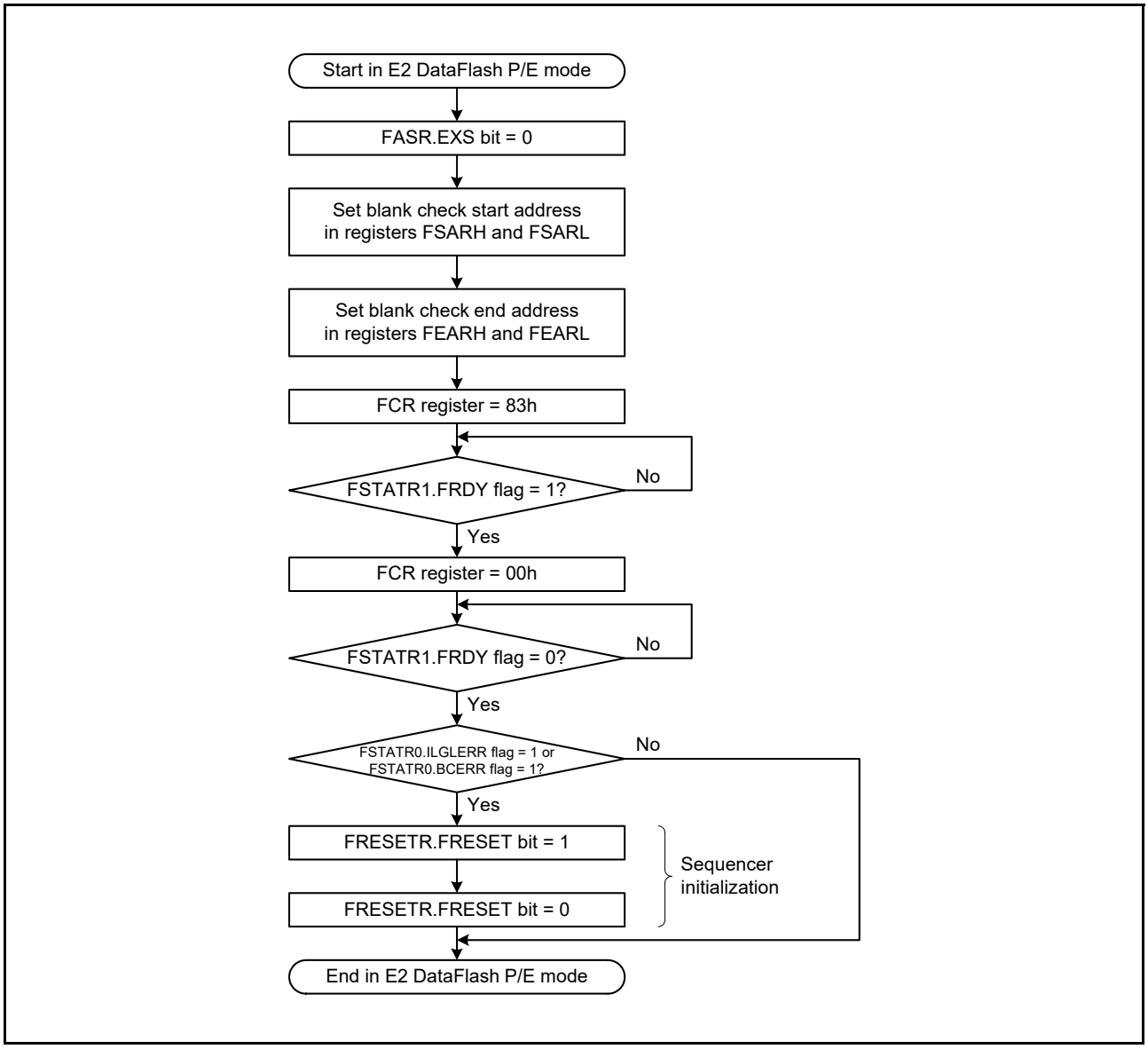


Figure 50.19 Procedure to Issue the Blank Check Command for the E2 DataFlash

Downloaded from Arrow.com.

50.7.4.5 Start-Up Area Information Program/Access Window Information Program

Figure 50.20 shows the procedure to issue the start-up area information program command and access window information program command.

When the sequencer has directly entered ROM/PE mode from E2 DataFlash access disabled mode, set the DFLCTL.DFLEN bit to 1 at the beginning of the procedure.

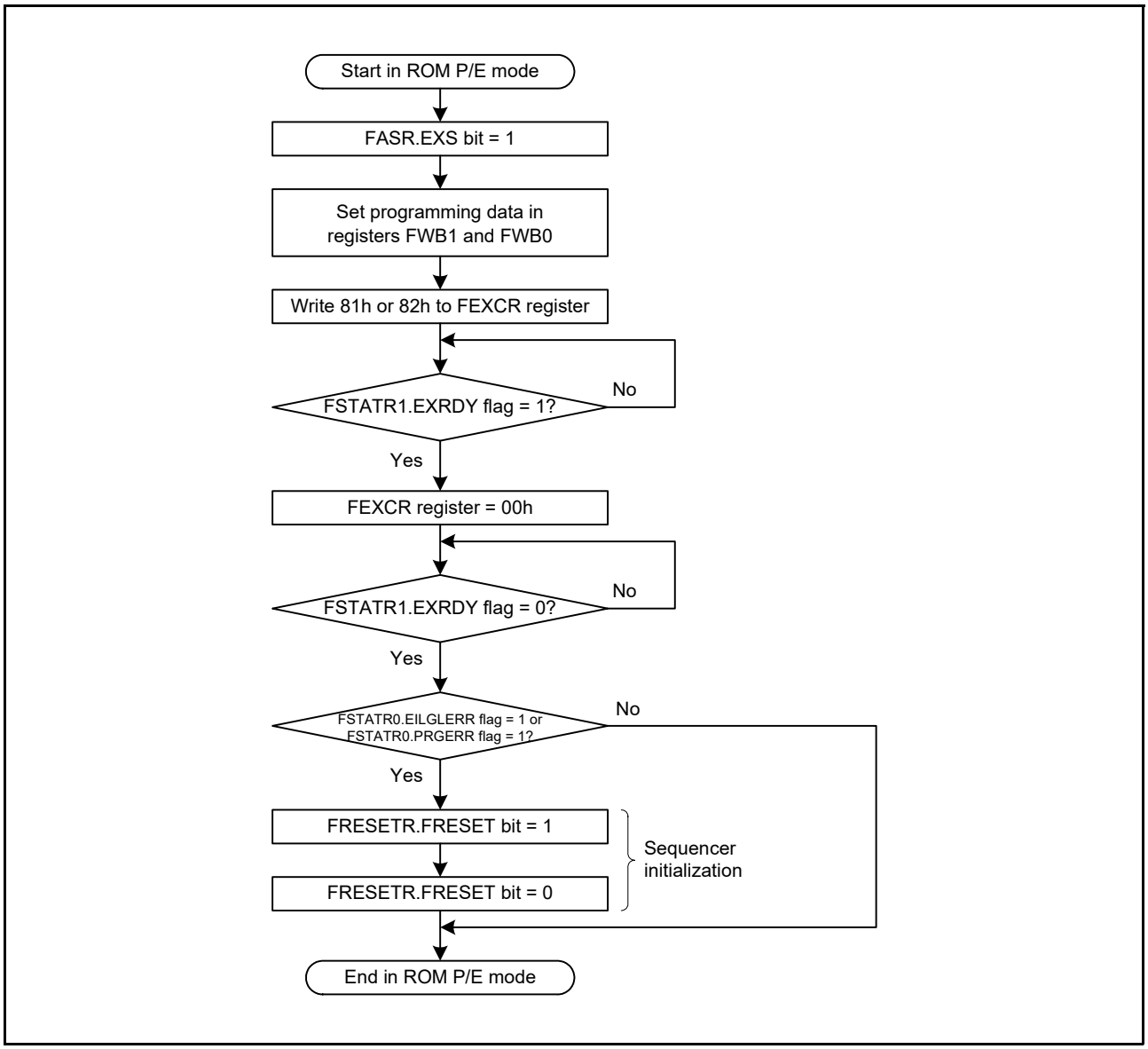


Figure 50.20 Procedure to Issue the Start-Up Area Information Program Command/Access Window Information Program Command

Downloaded from Arrow.com.

50.7.4.6 Forced Stop of Software Commands

Perform the procedure shown in Figure 50.21 to forcibly stop the blank check command or block erase command. When the command processing is forcibly stopped, registers FEAMH and FEAML store the address at the time of the forced stop. For blank check, the stopped processing can be continued by copying the FEAMH and FEAML register values to registers FSARH and FSARL.

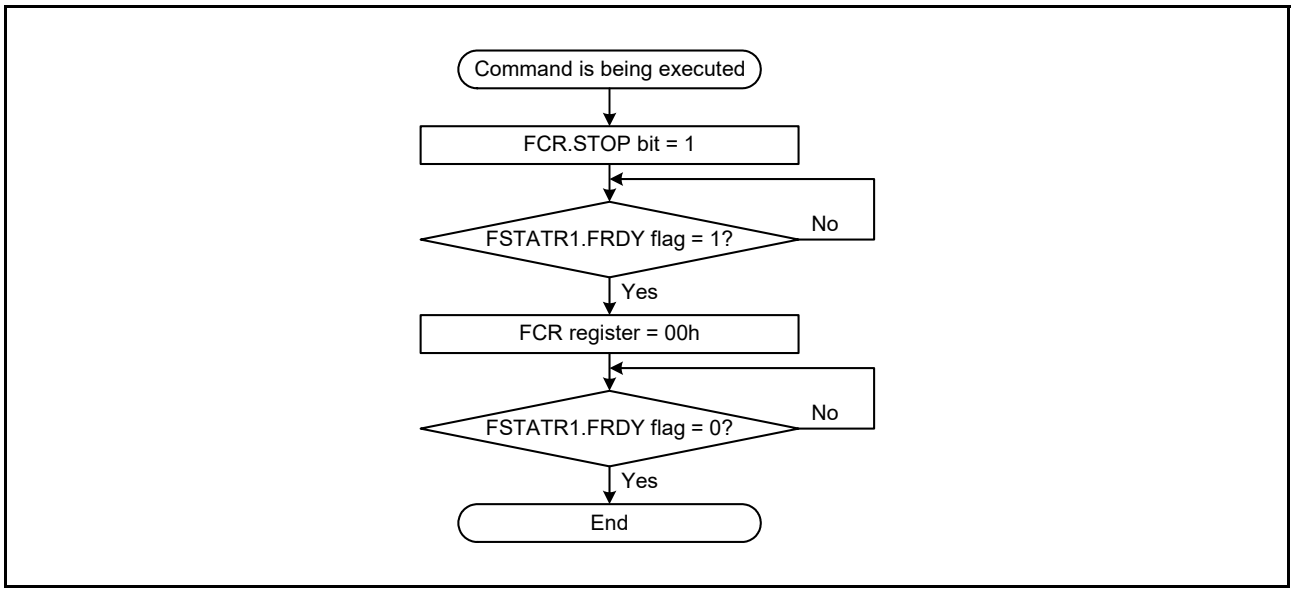


Figure 50.21 Procedure for Forced Stop of Software Commands

50.7.5 Interrupt

When software command processing or forced stop processing is completed, an interrupt (FRDYI) is generated.
When the FSTATR1.FRDY flag becomes 0 by setting the FCR.OPST bit to 0 and the FSTATR1.EXRDY flag becomes 0 by setting the FEXCR.OPST bit to 0, the next interrupt (FRDYI) can be accepted.
Clear the IRn.IR flag before setting the IERm.IEN bit of the ICU corresponding to this interrupt.

Downloaded from Arrow.com.

50.8 Boot Mode

The USB interface, SCI, or FINE interface is used in boot mode.
Table 50.6 lists the programmable and erasable areas and peripheral modules used in boot mode. Table 50.7 lists the I/O pins used in boot mode.

Table 50.6 Programmable and Erasable Areas and Peripheral Modules Used in Boot Mode

	Boot Mode		
Item	**USB Interface**	**SCI Interface**	**FINE Interface**
Programmable and erasable areas	User area Data area	User area Data area	User area Data area
Peripheral module	USB0	SCI1 (asynchronous serial communication)	FINE

Table 50.7 I/O Pins Used in Boot Mode

Pin Name	I/O	Mode	Description
PC7/UB	Input	Boot mode	Select operating mode (refer to section 3, Operating Modes).
MD	Input		Select operating mode (refer to section 3, Operating Modes).
MD/FINED	I/O	Boot mode (FINE interface)	Select operating mode/FINE data I/O
USB0_DP, USB0_DM	I/O	Boot mode (USB interface)	USB data I/O
P16/USB0_VBUS	Input		Detect USB cable connection/disconnection
P35/UPSEL	Input		Set bus-powered mode or self-powered mode
P30/RXD1	Input	Boot mode (SCI interface)	Receive data*1
P26/TXD1	Output		Transmit data*1

Note 1. When using the SCI, connect (pull up) this pin to VCC via a resistor.

Downloaded from Arrow.com.

50.8.1 Boot Mode (USB Interface)

The flash memory can be programmed and erased using the USB interface in boot mode (USB interface). The user area and data area can be rewritten.

When a reset is released while the MD pin is low and the UB pin is high, the MCU starts in boot mode (USB interface). Self-powered or bus-powered can be selected in accordance with the state of the UPSEL pin. When a reset is released while the UPSEL pin is low, self-powered mode is selected. When a reset is released while the UPSEL pin is high, bus-powered mode is selected.

Contact the manufacturer for details on the serial programmer (USB programmer).

50.8.1.1 Operating Conditions in Boot Mode (USB Interface)

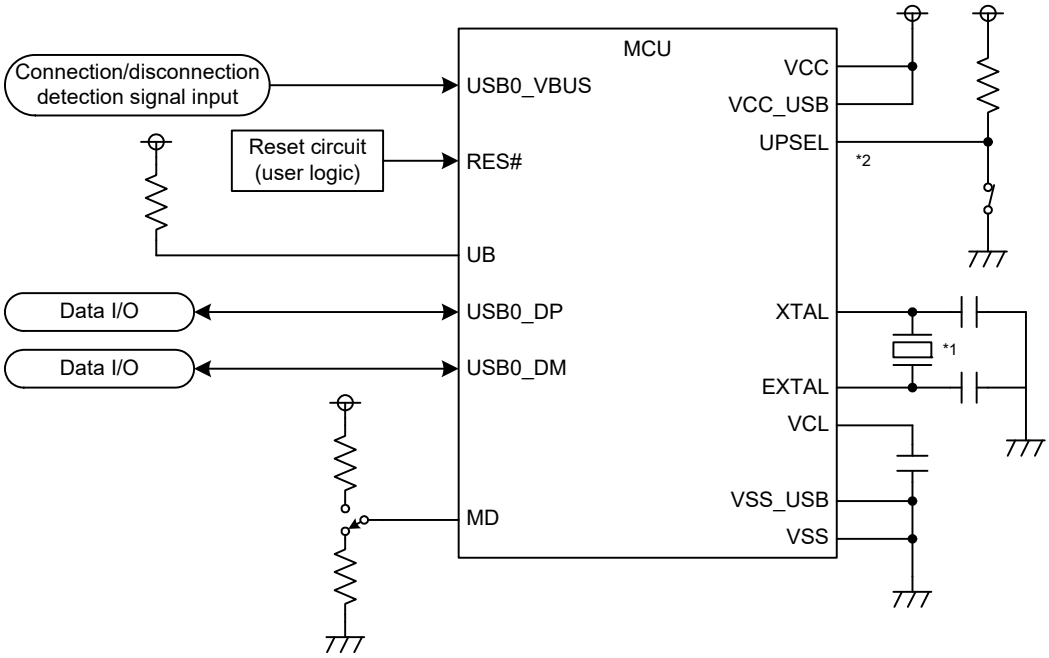
USB0 is used for communication with the serial programmer in boot mode (USB interface).

4, 6, 8, 12, or 16 MHz can be used as the frequency input to the main clock oscillator. The operating voltage range is between 3.0 V and 3.6 V.

Connect the UB pin to VCC directly or VCC via a resistor (pull up).

Figure 50.22 shows an example of pin connections in boot mode (USB interface) when self-powered. Table 50.8 lists pin handling in boot mode (USB interface) when self-powered. Figure 50.23 shows an example of pin connections in boot mode (USB interface) when bus-powered. Table 50.9 lists pin handling in boot mode (USB interface) in bus-powered.

Examples of pin connections shown in Figure 50.22 and Figure 50.23 are simplified circuits. Operations are not guaranteed in all systems.



Note 1. Connect a 4, 6, 8, 12, or 16 MHz crystal, ceramic resonator or oscillator to the main clock oscillator.
Note 2. UPSEL is assigned to the same pin as NMI. This is an example when the falling edge is selected by the NMI detection set bit.

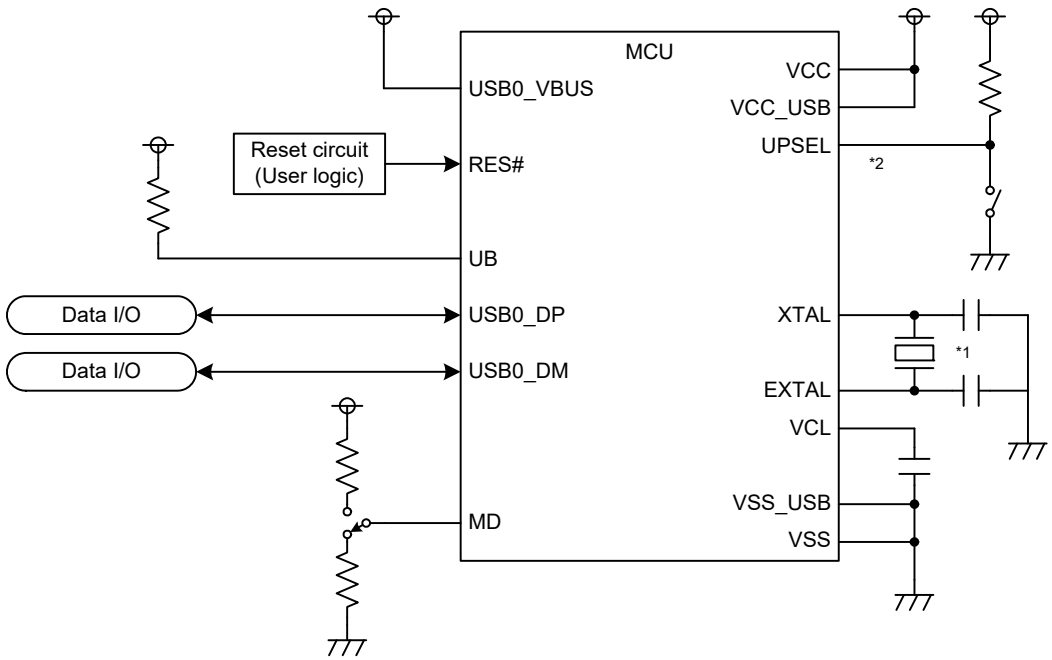
Figure 50.22 Example of Pin Connections in Boot Mode (USB Interface) When Self-Powered

Downloaded from Arrow.com.

Table 50.8 Pin Handling in Boot Mode (USB Interface) When Self-Powered

Pin Name	Name	I/O	Function
VCC, VSS	Power supply	—	Input the voltage between 3.0 V and 3.6 V to the VCC pin. Input 0 V to the VSS pin.
VCC_USB, VSS_USB	USB power supply	—	Connect the VCC_USB pin to the VCC pin. Connect the VSS_USB pin to the VSS pin.
AVCC0, AVSS0	12-bit A/D converter power supply	—	Connect the AVCC0 pin to the VCC pin. Connect the AVSS0 pin to the VSS pin.
VCL	Decoupling capacitor connect pin	—	Connect to the VSS pin via a decoupling capacitor for stabilizing the internal voltage.
XTAL, EXTAL	Main clock I/O pin	I/O	Connect a 4, 6, 8, 12, or 16 MHz crystal or ceramic resonator or oscillator.
MD	Operating mode control	Input	Input low.
PC7/UB	Operating mode control	Input	Input high.*1
P35/UPSEL	USB power mode control	Input	Input low.
RES#	Reset input	Input	Reset pin. Connect to the reset circuit.
USB0_DP	USB on-chip transceiver D+ I/O pin	I/O	Connect to the circuit described in section 32, USB 2.0 Host/Function Module (USBc).
USB0_DM	USB on-chip transceiver D– I/O pin	I/O	Connect to the circuit described in section 32, USB 2.0 Host/Function Module (USBc).
P16/USB0_VBUS	USB cable connection monitor pin	Input	Connect to the circuit described in section 32, USB 2.0 Host/Function Module (USBc).

Note 1. Maintain the input level for 2 ms or longer after a reset is released.



Note 1. Connect a 4, 6, 8, 12, or 16 MHz crystal, ceramic resonator or oscillator to the main clock oscillator.
Note 2. UPSEL is assigned to the same pin as NMI. This is an example when the falling edge is selected by the NMI detection set bit.

Figure 50.23 Example of Pin Connections in Boot Mode (USB Interface) When Bus-Powered

Downloaded from Arrow.com.

Table 50.9 Pin Handling in Boot Mode (USB Interface) in Bus-Powered

Pin Name	Name	I/O	Function
VCC, VSS	Power supply	—	Input the voltage between 3.0 V and 3.6 V to the VCC pin. Input 0 V to the VSS pin.
VCC_USB, VSS_USB	USB power supply	—	Connect the VCC_USB pin to the VCC pin. Connect the VSS_USB pin to the VSS pin.
AVCC0, AVSS0	12-bit A/D converter power supply	—	Connect the AVCC0 pin to the VCC pin. Connect the AVSS0 pin to the VSS pin.
VCL	Decoupling capacitor connect pin	—	Connect to the VSS pin via a decoupling capacitor for stabilizing the internal voltage.
XTAL, EXTAL	Main clock I/O pin	I/O	Connect a 4, 6, 8, 12, or 16 MHz crystal or ceramic resonator or oscillator.
MD	Operating mode control	Input	Input low.
PC7/UB	Operating mode control	Input	Input high.*1
P35/UPSEL	USB power mode control	Input	Input high.
RES#	Reset input	Input	Reset pin. Connect to the reset circuit.
USB0_DP	USB on-chip transceiver D+ I/O pin	I/O	Connect to the circuit described in section 32, USB 2.0 Host/Function Module (USBc).
USB0_DM	USB on-chip transceiver D– I/O pin	I/O	Connect to the circuit described in section 32, USB 2.0 Host/Function Module (USBc).
P16/USB0_VBUS	USB cable connection monitor pin	Input	Connect to the USB0_VBUS pin to the VCC pin.

Note 1. Maintain the input level for 2 ms or longer after a reset is released.

Downloaded from Arrow.com.

50.8.2 Boot Mode (SCI Interface)

The flash memory can be programmed and erased using asynchronous serial communication in boot mode (SCI interface). The user area and data area can be rewritten.
When a reset is released while the MD pin and the UB pin are low, the MCU starts in boot mode (SCI interface).
Contact the manufacturer for details on the serial programmer.

50.8.2.1 Operating Conditions in Boot Mode (SCI Interface)

SCI1 is used to communicate with the serial programmer in boot mode (SCI interface).
Figure 50.24 shows an example of pin connections in boot mode (SCI interface). Table 50.10 lists pin handling in boot mode (SCI interface).
An example of pin connections shown in Figure 50.24 is a simplified circuit. Operations are not guaranteed in all systems.

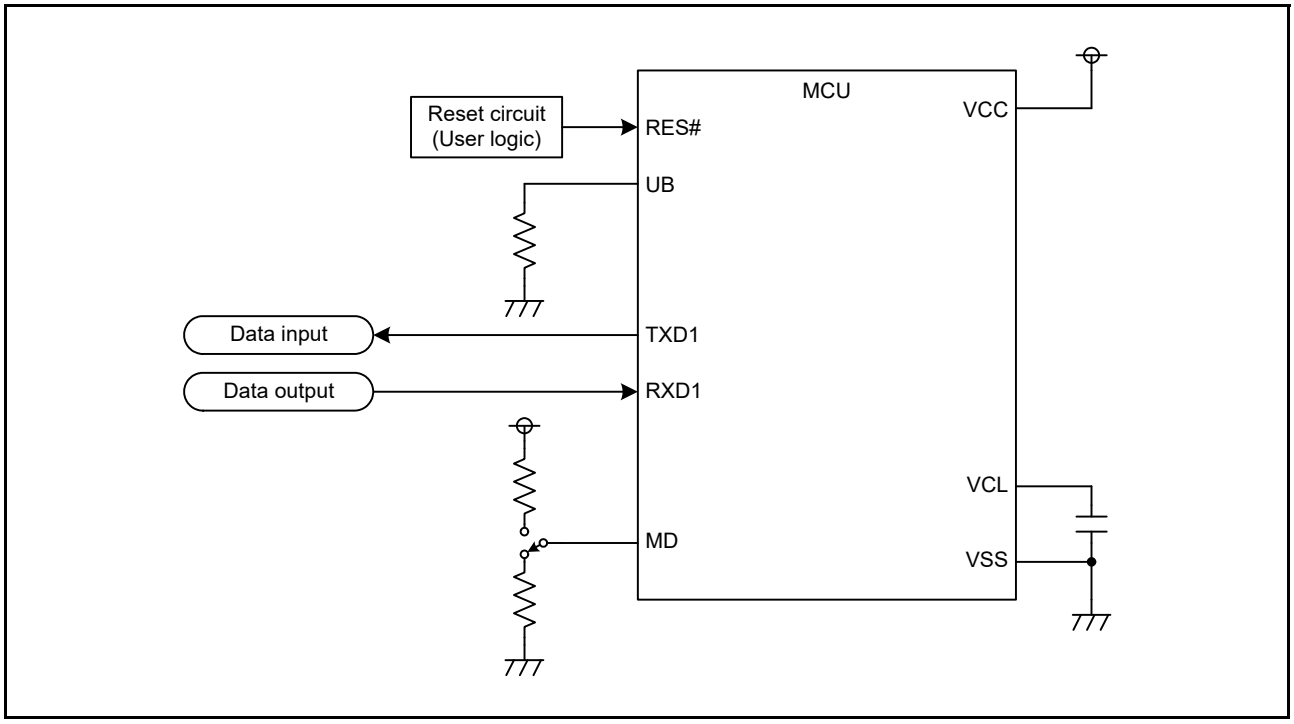


Figure 50.24 Example of Pin Connections in Boot Mode (SCI Interface)

Table 50.10 Pin Handling in Boot Mode (SCI Interface)

Pin Name	Name	I/O	Function
VCC, VSS	Power supply	—	Input 1.8 V or higher to the VCC pin. Input 0 V to the VSS pin.
VCL	Decoupling capacitor connect pin	—	Connect to the VSS pin via a decoupling capacitor for stabilizing the internal voltage.
MD	Operating mode control	Input	Input low.
PC7/UB	Operating mode control	Input	Input low.*1
RES#	Reset input	Input	Reset pin. Connect to the reset circuit.
P30/RXD1	Data input RXD	Input	Input pin for serial data
P26/TXD1	Data output TXD	Output	Output pin for serial data

Note 1. Maintain the input level for 2 ms or longer after a reset is released.

Downloaded from Arrow.com.

As shown in Figure 50.25, set the format to 8-bit data, 1 stop bit, no parity, and LSB first to communicate with the serial programmer.

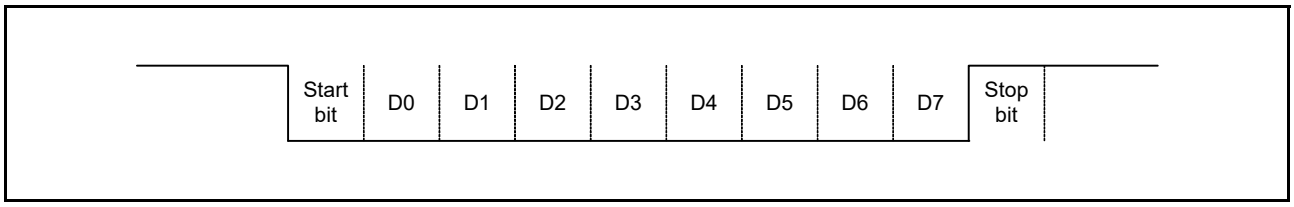


Figure 50.25 Communication Format

Initial communication with the programmer is performed at 9,600 or 19,200 bps. The communication bit rate can be changed after the MCU is connected with the programmer.
Table 50.11 lists the maximum communication bit rates for communication in boot mode (SCI interface).

Table 50.11 Conditions for Communication

Operating Voltage	Maximum Communication Bit Rate
Lower than 3.0 V	500 kbps
3.0 V or higher	2 Mbps

50.8.2.2 Starting Up in Boot Mode (SCI Interface)

To start the MCU in boot mode (SCI interface), a reset must be released by changing the RES# pin from low to high while both of the MD pin and UB pin are low. After starting up in boot mode (SCI interface), wait at least 400 ms until communication with the MCU is enabled in boot mode (SCI interface).
As shown in Figure 50.26, keep the signal of each pin unchanged for 400 ms after the reset is released. Use resets according to the range described in section 51.3.2, Reset Timing.

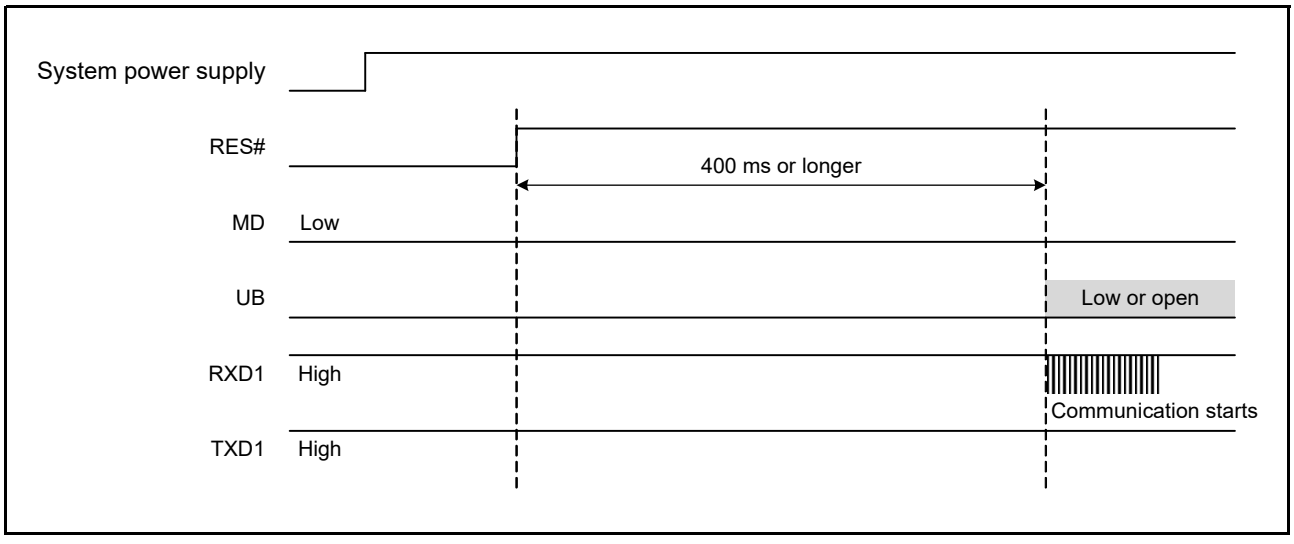


Figure 50.26 Wait Time until Communication Becomes Possible in Boot Mode (SCI Interface)

Downloaded from Arrow.com.

50.8.3 Boot Mode (FINE Interface)

The flash memory can be programmed and erased using the FINE in boot mode (FINE interface). The user area and data area can be rewritten.
Contact the manufacturer for details on the serial programmer.

50.8.3.1 Operating Conditions in Boot Mode (FINE Interface)

FINE is used to communicate with the serial programmer in boot mode (FINE interface).
Figure 50.27 shows an example of pin connections in boot mode (FINE interface). Table 50.12 lists pin handling in boot mode (FINE interface).
An example of pin connections shown in Figure 50.27 is a simplified circuit. Operations are not guaranteed in all systems.

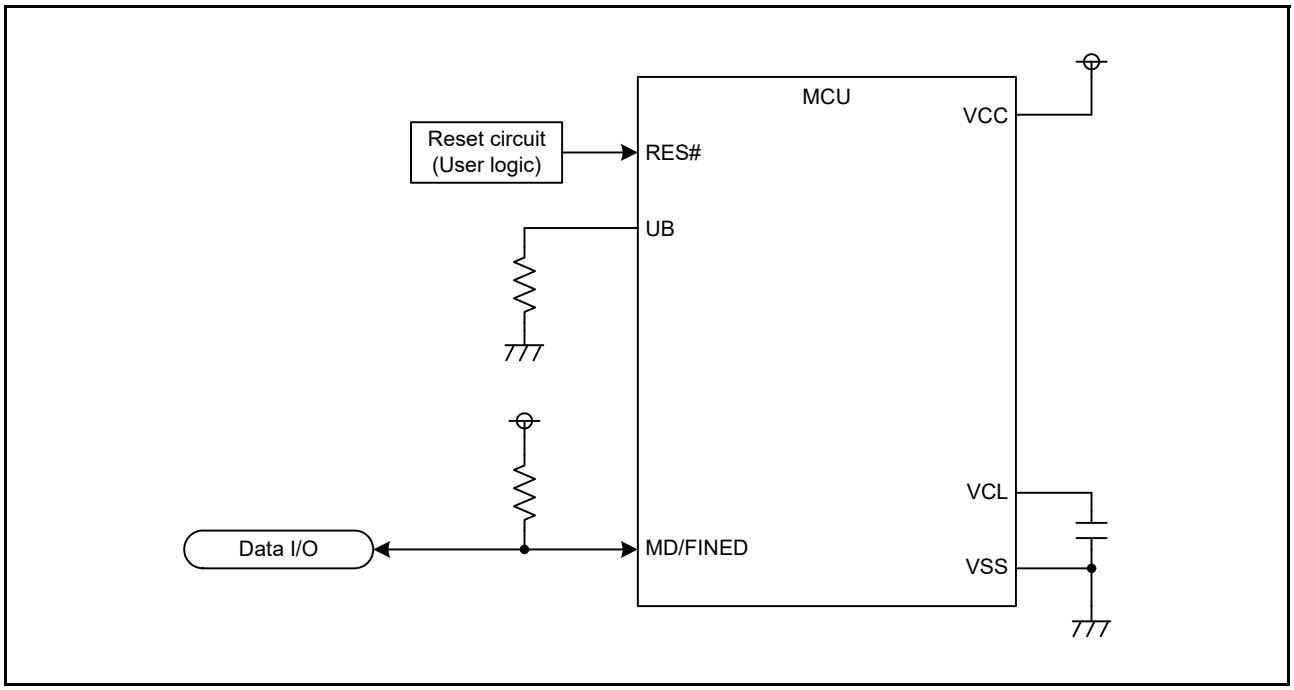


Figure 50.27 Example of Pin Connections in Boot Mode (FINE Interface)

Table 50.12 Pin Handling in Boot Mode (FINE Interface)

Pin Name	Name	I/O	Function
VCC, VSS	Power supply	—	Input 1.8 V or higher to the VCC pin. Input 0 V to the VSS pin.
VCL	Decoupling capacitor connect pin	—	Connect to the VSS pin via a decoupling capacitor for stabilizing the internal voltage.
MD/FINED	Operating mode control/data I/O	I/O	Connect to the VCC pin via a resistor (pull up).
PC7/UB	Operating mode control	Input	Input low.*1
RES#	Reset input	Input	Reset pin. Connect to the reset circuit.

Note 1. Maintain the input level for 2 ms or longer after a reset is released.

Downloaded from Arrow.com.

50.9 Flash Memory Protection

Flash memory protection prevents the flash memory from being read or rewritten by the third party.
The boot mode ID code protection is for connecting the serial programmer, and the on-chip debugging emulator ID code protection is for connecting the on-chip debugging emulator.

50.9.1 ID Code Protection

There are two types of ID code protection: Boot mode ID code protection for connecting the serial programmer and on-chip debugging emulator ID code protection is for connecting the on-chip debugging emulator. The same ID codes are used for both functions, but operations differ.
ID codes consist of the control code and ID code 1 to ID code 15. Set ID codes to four 32-bit data in 32-bit units. Figure 50.28 shows the ID code configuration.

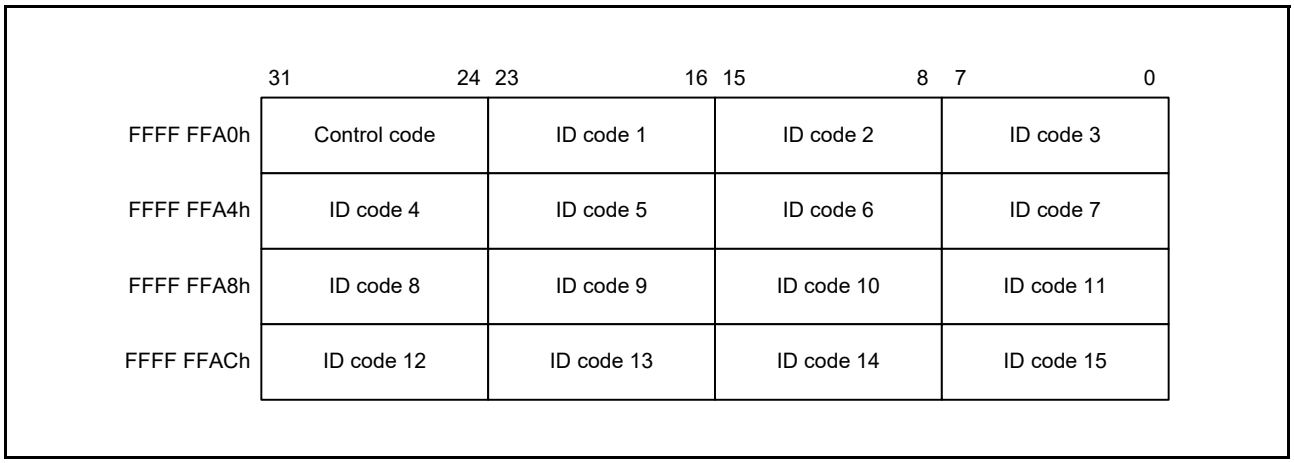


Figure 50.28 ID Code Configuration

The following shows a program example for setting ID codes.
This is an example when setting the control code to 45h and setting ID codes to 01h, 02h, 03h, 04h, 05h, 06h, 07h, 08h, 09h, 0Ah, 0Bh, 0Ch, 0Dh, 0Eh, and 0Fh (from the ID code 1 field to the ID code 15 field).

C language:

```
#pragma address ID_CODE = 0xFFFFFFA0
const unsigned long ID_CODE [4] = {0x45010203, 0x04050607,0x08090A0B, 0x0C0D0E0F};
```

Assembly language:

```
.SECTION ID_CODE,CODE
.ORG 0FFFFFFA0h
.LWORD 45010203h
.LWORD 04050607h
.LWORD 08090A0Bh
.LWORD 0C0D0E0Fh
```


Downloaded from Arrow.com.

50.9.1.1 Boot Mode ID Code Protection

Boot mode ID code protection disables reading and programming of the user area and data area when the serial programmer is connected by the third party.
When the control code indicates 45h or 52h (boot mode ID code protection is enabled), the MCU compares 16-byte ID code sent from the serial programmer with the ID code in the user area. According to the comparison result, reading and programming the user area and data area are enabled.
When the control code indicates a value other than 45h and 52h (boot mode ID code protection is disabled), all blocks in the user area and data area are erased, and reading and programming the user area and data area are enabled.

The control code is used to enable or disable protection. Table 50.13 lists the specifications of boot mode ID code protection, and Figure 50.29 shows the authentication flow of boot mode ID code protection.
ID code 1 to ID code 15 can be set to any desired value.
However, only when disabling connection with the serial programmer, the ID codes must be set to 50h, 72h, 6Fh, 74h, 65h, 63h, 74h, FFh, FFh, FFh, FFh, FFh, FFh, FFh, and FFh (from the ID code 1 field to the ID code 15 field).

Table 50.13 Boot Mode ID Code Protection Specifications

ID Code				
Control Code	**ID Code 1 to ID Code 15**	**Protection**	**ID Code Matching Result**	**Operation**
45h	Any desired value	Enabled	Matched	Exit the boot mode ID code authentication state and enter the program/erase host command wait state.
			Not matched	Continue the boot mode ID code authentication state.
			Not matched three times consecutively	Erase all blocks in the user area and data area, and continue boot mode ID code authentication state.
52h	50h, 72h, 6Fh, 74h, 65h, 63h, 74h, FFh, ..., and FFh (8 bytes are all FFh)	Enabled	N/A	Disable reading or rewriting of the flash memory, regardless of the codes sent from the serial programmer.
	Other than above		Matched	Exit the boot mode ID code authentication state and enter the program/erase state.
			Not matched	Continue the boot mode ID code authentication state.
Other than above	Any desired value	Disabled	N/A	Erase all blocks in the user area and data area.

Downloaded from Arrow.com.

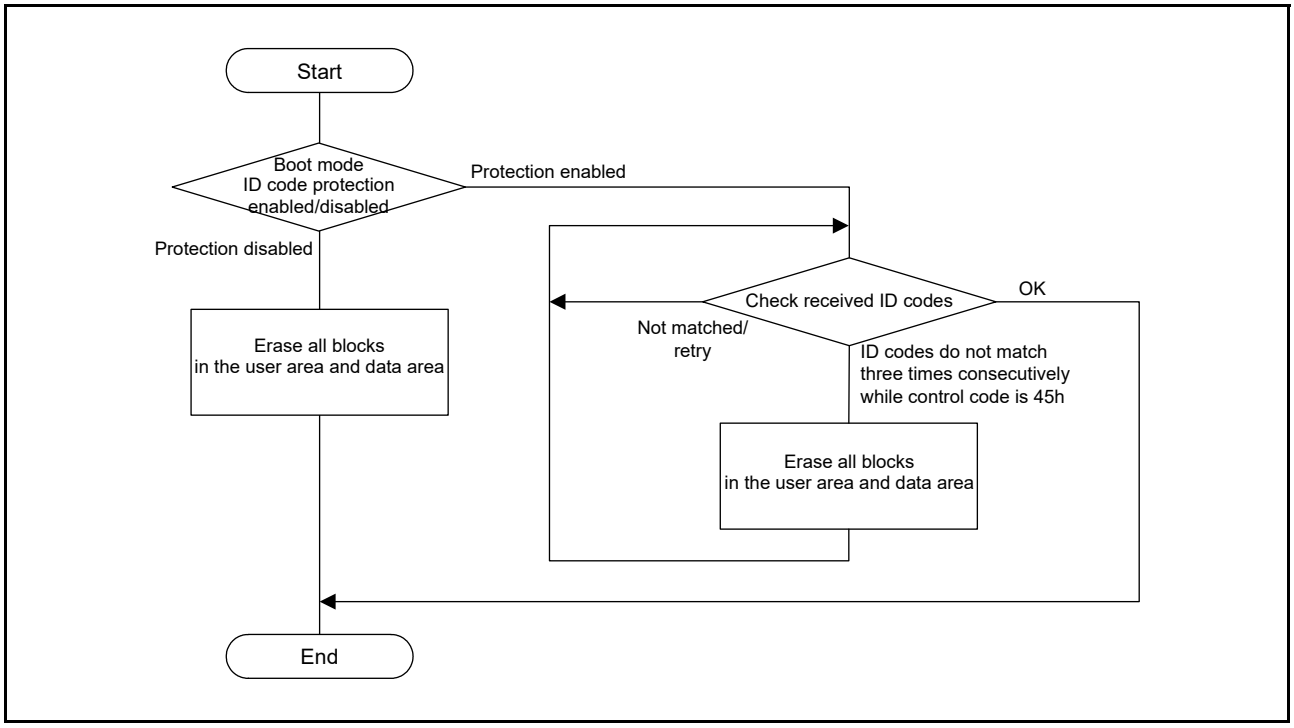


Figure 50.29 Authentication for Boot Mode ID Code Protection

50.9.1.2 On-Chip Debugging Emulator ID Code Protection

On-chip debugging emulator ID code protection enables or disables connection with the on-chip debugging emulator. When the on-chip debugging emulator ID code protection is disabled, connection with the on-chip debugging emulator is enabled. When 16-byte ID codes sent from the on-chip debugging emulator and ID codes in the user area match while on-chip debugging emulator ID code protection is enabled, connection with the on-chip debugging emulator is also enabled.

Table 50.14 lists the specifications of on-chip debugging emulator ID code protection.

Table 50.14 On-Chip Debugging Emulator ID Code Protection Specifications

ID Code				
Control Code	ID Code 1 to ID Code 15	Protection	ID Code Matching Result	Operation
FFh	FFh,…, and FFh (15 bytes are all FFh)	Disabled	N/A	Enable connection with the on-chip debugging emulator.
52h	50h, 72h, 6Fh, 74h, 65h, 63h, and 74h + any 8 bytes	Enabled	N/A	Disable connection with the on-chip debugging emulator, regardless of the codes sent from the on-chip debugging emulator.
Other than above	Other than above	Enabled	Matched	Enable connection with the on-chip debugging emulator.
			Not matched	Continue the ID code wait state.

Downloaded from Arrow.com.

50.10 Communication Protocol

This section describes the protocol used in boot mode. When developing a serial programmer, control with this communication protocol.

50.10.1 State Transition in Boot Mode (SCI Interface)

Figure 50.30 shows the state transition in boot mode (SCI interface).

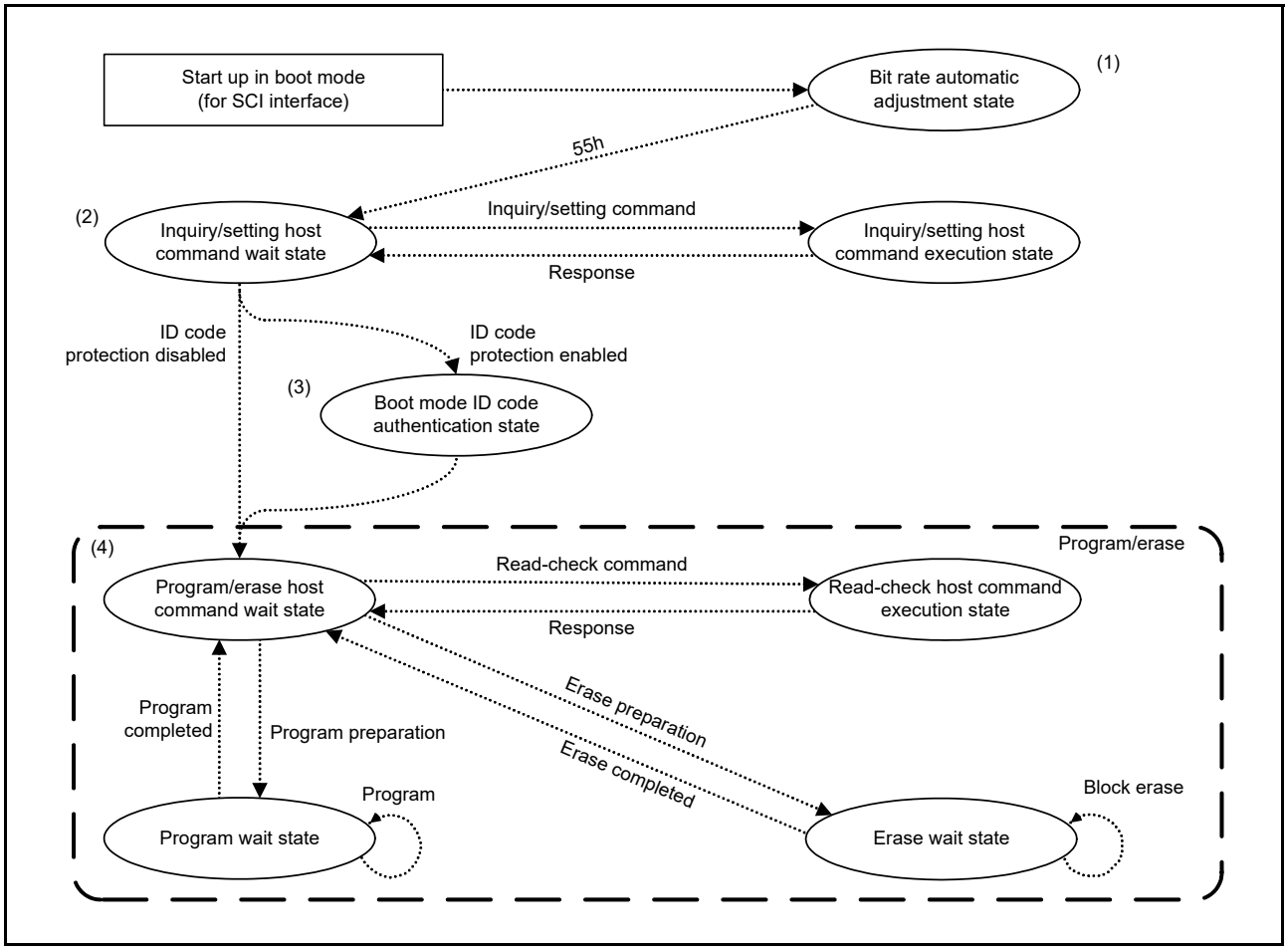


Figure 50.30 Boot Mode (SCI Interface) State Transition

(1) Bit rate automatic adjustment state

In this state, the bit rate is automatically adjusted to 9,600 or 19,200 bps for communication with the host.

When the bit rate adjustment is completed, the MCU sends 00h to the host. After that, when the MCU receives 55h sent from the host, the MCU sends E6h to the host, and enters the inquiry/setting host command wait state.

The host must not send data until 400 ms elapse after a reset of the MCU is released.

(2) Inquiry/setting host command wait state

In this state, the host can make inquiries for the MCU information including block configuration, size, and addresses where the user area and data area are allocated, and select the endian of data and a bit rate.

When the MCU receives the program/erase host state transition command from the host, it determines whether boot mode ID code protection is enabled or disabled. If boot mode ID code protection is disabled, the MCU enters the inquiry/setting host command wait state. If boot mode ID code protection is enabled, the MCU enters the boot mode ID code authentication state.

Refer to section 50.10.5, Inquiry Commands and section 50.10.6, Setting Commands for details on inquiry/setting commands.

Downloaded from Arrow.com.

(3) Boot mode ID code authentication state
In this state, the MCU accepts the ID code authentication command.
When boot mode ID codes do not match, the MCU remains in the boot mode ID code authentication state.
Refer to section 50.9.1.1, Boot Mode ID Code Protection for details on boot mode ID code protection. Refer to section 50.10.7, ID Code Authentication Command for details on the ID code authentication command.

(4) Program/erase state
In this state, the MCU executes program/erase or read-check commands according to commands sent from the host.
Refer to section 50.10.8, Program/Erase Commands for details on program/erase commands. Refer to section 50.10.9, Read-Check Commands for details on read-check commands.

50.10.2 Command and Response Configuration

The communication protocol is composed of a “Command” sent from the host to the MCU and a “Response” sent from the MCU to the host. Commands include 1-byte commands and multiple-byte commands. Responses include 1-byte responses, multiple-byte responses, and error responses.
A multiple-byte command and multiple-byte response have “Size” for informing the number of transmit/receive data bytes and “SUM” for detecting communication errors.
“Size” indicates the number of transmit/receive data bytes excluding Command code (the first byte), Size, and SUM.
“SUM” indicates byte data that is calculated so the total bytes of Command or Response becomes 00h.
The flash memory addresses for reading are used as the following addresses: the program address specified in the program command, the block start address specified in the block erase command, the AW start and end addresses specified in the access window information program command, and the AW start and end addresses received in the access window read command.

50.10.3 Response to Undefined Commands

When the MCU receives an undefined command, it sends a command error as a response. The contents of the response are shown below. “Command code” in the error response stores the first byte of the command sent from the MCU.

Error response	80h	Command code

Downloaded from Arrow.com.

50.10.4 Boot Mode Status Inquiry

This command is used to check the current state and which type of an error occurred immediately after a command issued in the boot program.
Table 50.15 and Table 50.16 list a state or error that the MCU responds to.
The boot mode status inquiry command can be used in the inquiry/setting host command wait state and program/erase host command wait state.

Command	4Fh				
Response	5Fh	Size	State	Error	SUM

Size (1 byte): Total bytes of “State” and “Error” (the value is always 02h)
State (1 byte): MCU’s current state (see Table 50.15)
Error (1 byte): Information about the error occurred in response to a command issued immediately before (see Table 50.16)
SUM (1 byte): Value that is calculated so the sum of response data is 00h

Table 50.15 Information Regarding the States

Code	State*1	Description
11h	Inquiry/setting host command wait state	Device selection wait state
12h/13h		Operating frequency selection wait state
1Fh		Program/erase host command wait state transition command wait state
31h	Boot mode ID code authentication state	The user area and data area are being erased
3Fh	Program/erase host command wait state	Program/erase command wait state
4Fh		Program data reception wait state
5Fh		Block erase specification wait state

Note 1. Refer to Figure 50.30 for details on the states.

Table 50.16 Error Information

Code	Description
00h	No error
11h	SUM error
21h	Device code error
24h	Bit rate selection error
29h	Block start address error
2Ah	Address error
2Bh	Data length error
51h	Erase error
52h	Not blank (blank check error)
53h	Program error
61h	ID code do not match
63h	ID code do not match and erase error
80h	Command error
FFh	Bit rate automatic adjustment error

Downloaded from Arrow.com.

50.10.5 Inquiry Commands

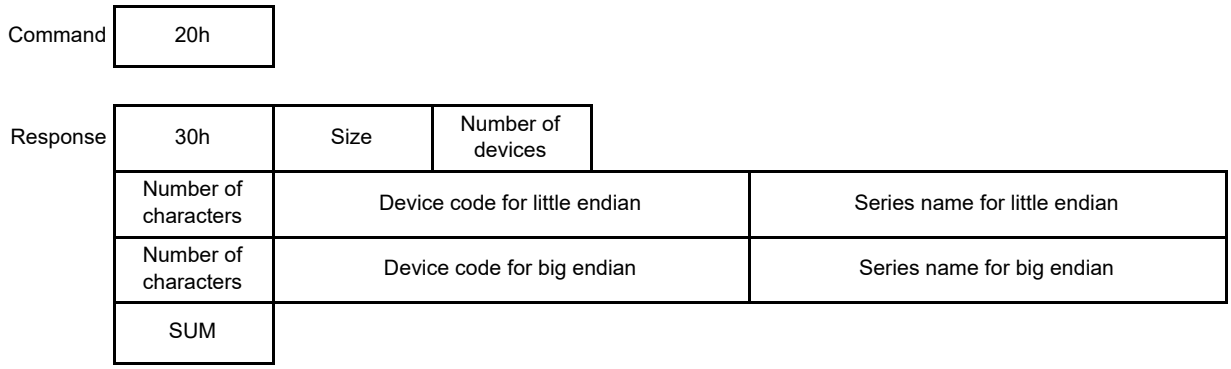
Inquiry commands are used to obtain necessary information for sending setting commands, program/erase commands, and read-check commands. Table 50.17 lists the inquiry commands. These commands can only be used in the inquiry/setting host command wait state.

Table 50.17 Inquiry Commands

Command	Description
Supported device inquiry	Inquiry for the device code and series name
Data area availability inquiry	Inquiry for the availability of the data area
User area information inquiry	Inquiry for the number of user areas, and the start and end addresses of the user area
Data area information inquiry	Inquiry for the number of data areas, and the start and end addresses of the data area
Block information inquiry	Inquiry for the start address, the block size, and the number of blocks of each of the user and data areas

50.10.5.1 Supported Device Inquiry

This command is used to obtain the device information for identifying the endian of developed software.
When the MCU receives this command, it sends the device information when developed software uses little endian data and the device information when developed software uses big endian data in this order.



Size (1 byte): Total bytes of Number of Devices, Characters, Device code, and Series name
Number of devices (1 byte): Number of endian types that the MCU supports (the value is always 02h)
Number of characters (1 byte): Number of characters for the device code and device name
Device code (4 bytes): Identification code indicating the endian of developed software
Series name (n bytes): The series name of the MCU (ASCII code) and the classification of little endian/big endian
SUM (1 byte): Value that is calculated so the sum of response data is 00h

Downloaded from Arrow.com.

50.10.5.2 Data Area Availability Inquiry

When the MCU receives this command, it sends the result indicating that the data area is available, area protection can be used, and the data area program command is available.

Command	2Ah			
Response	3Ah	Size	Availability	SUM

Size (1 byte): Number of characters of Availability (the value is always 01h)
Availability (1 byte): Availability of the data area (the value is always 1Dh)
1Dh represents the data area is available, area protection can be used, and data area program command is available.
SUM (1 byte): Value that is calculated so the sum of response data is 00h (the value is always A8h)

50.10.5.3 User Area Information Inquiry

When the MCU receives this command, it sends the number of user areas and addresses.

Command	25h		
Response	35h	Size	Number of areas
	Area start address		
	Area end address		
	SUM		

Size (1 byte): Total bytes of Number of areas, Area start address, and Area end address (the value is always 09h)
Number of areas (1 byte): Number of user areas (the value is always 01h)
Area start address (4 bytes): Start address of the user area
Area end address (4 bytes): End address of the user area
SUM (1 byte): Value that is calculated so the sum of the response data is 00h

Downloaded from Arrow.com.

50.10.5.4 Data Area Information Inquiry

When the MCU receives this command, it sends the number of data areas and addresses.

Command	2Bh		
Response	3Bh	Size	Number of areas
	Area start address		
	Area end address		
	SUM		

Size (1 byte): Total bytes of data of Number of areas, Area start address, and Area end address (the value is always 09h)
Number of areas (1 byte): Number of areas in the data area (the value is always 01h)
Area start address (4 bytes): Start address of the data area (the value is always 0010 0000h)
Area end address (4 bytes): End address of the data area (the value is always 0010 1FFFh)
SUM (1 byte): Value that is calculated so the sum of the response data is 00h (the value is always 7Dh)

50.10.5.5 Block Information Inquiry

When the MCU receives this command, it sends the start address, the size of one block, and the number of blocks in the user area and data area.

Command	26h		
Response	36h	Size	DDh
	Start address of the user area		
	Block size of one block for the user area		
	Number of blocks of the user area		
	Start address of the data area		
	Block size of one block for the data area		
	Number of blocks of the data area		
	SUM		

Size (2 bytes): Total bytes of data from DDh to Number of blocks of the data area (the value is always 00 19h)
Start address of the user area (4 bytes): Start address of the user area
Block size of one block for the user area (4 bytes): Memory size of one block (the value is always 00 00 08 00h)
Number of blocks of the user area (4 bytes): Number of blocks in the user area
Start address of the data area (4 bytes): Start address of the data area (the value is always 00 10 00 00h)
Block size of one block for the data area (4 bytes): Memory size of one block (the value is always 00 00 04 00h)
Number of blocks of the data area (4 bytes): Number of blocks in the data area (the value is always 00 00 00 08h)
SUM (1 byte): Value that is calculated so the sum of response data is 00h

Downloaded from Arrow.com.

50.10.6 Setting Commands

Setting commands are used to configure the settings necessary to execute program/erase commands in the MCU. Table 50.18 lists setting commands. These commands can be used only in the inquiry/setting host command wait state.

Table 50.18 Setting Commands

Command	Function
Device select	Select a device code.
Operating frequency select	Change the bit rate for communication.
Program/erase host command wait state transition	Enter the program/erase host command wait state or boot mode ID code authentication state.

50.10.6.1 Device Select

This command is used to specify the endian of developed software. Select a device code from among the device codes obtained in the response to the support device inquiry command.
If the received device code matches the supported device, the MCU sends a response (46h).
If the device is not supported or the SUM of the received command does not match, the MCU sends an error response.

Command	10h	Size	Device code	SUM

Size (1 byte): Number of characters of the device code (the value is always 04h)
Device code (4 bytes): Identification code to identify an endian of the developed software
(code in the response to the support device inquiry command)
SUM (1 byte): Value that is calculated so the sum of command data is 00h

Response	46h

Error response	90h	Error

Error (1 byte): Error code
11h: SUM error
21h: Device code error

Downloaded from Arrow.com.

50.10.6.2 Operating Frequency Select

This command is used to specify the operating frequency of the MCU and a bit rate for communication with the flash memory programmer. The bit rate selected in this command should be set to a value with error of less than 4% compared to the bit rate obtained by dividing 32 or 8 MHz that corresponds to the operating voltage.
If the specified settings can be supported, the MCU sends a response (06h). If the bit rate error is 4% or more or the SUM of the received command does not match, the MCU sends an error response.

After the host receives a response, wait for at least a 1-bit period at the old bit rate, and send communication confirmation data at the new bit rate.
If the MCU successfully receives communication confirmation data, the MCU sends a response (06h). If the MCU fails to receive the communication confirmation data, the MCU sends an error response.

Command	3Fh	Size	Bit rate	Dummy data
	Number of clocks	Multiplier 1	Multiplier 2	
	SUM			

Size (1 byte): Total bytes of data of Bit rate, Dummy data, Number of clocks, and Multiplier (the value is always 07h)
Bit rate (2 bytes): New bit rate
The value is calculated by dividing the bit rate by 100 (Example: Set 00C0h for 19200 bps)
Dummy data (2 bytes): The value should always be set to 0000h
Number of clocks (1 byte): Types of clocks for multiplier setting (the value is always 02h)
Multiplier 1 (1 byte): Multiplier of the system clock (ICLK) (the value is always 01h)
Multiplier 2 (1 byte): Multiplier of the peripheral module clock (PCLK) (the value is always 01h)
SUM (1 byte): Value that is calculated so the sum of command data including dummy data is 00h

Response	06h	
Error response	BFh	Error

Error (1 byte): Error code
11h: SUM error
24h: Bit rate selection error

Communication confirmation	06h
Response	06h
Error response	FFh

Downloaded from Arrow.com.

- Bit rate selection error

A bit rate selection error occurs when the bit rate specified with the operating frequency select command cannot be set to a value with error of less than 4%. When the new bit rate specified with the operating frequency select command is B, and 32 (MHz) or 8 (MHz) corresponding to the operating voltage is Pφ, the bit rate error is calculated by the following formula:

$$\text{Error}(\%) = \left(\frac{P\varphi \times 10^6}{B \times 32 \times N} - 1\right) \times 100$$

$$N = \text{INT}\left(\frac{P\varphi \times 10^6}{B \times 32}\right)$$

Pφ: 32 (MHz) when the operating voltage is 3.0 V or above
8 (MHz) when the operating voltage is below 3.0 V
B: New bit rate (bps)
N: Ratio between Pφ and the new bit rate multiplied by 32 (however, 1 ≤ N ≤ 256)

50.10.6.3 Program/Erase Host Command Wait State Transition

This command is used for the transition from the inquiry/setting host command wait state to the program/erase host command wait state.
When the MCU receives this command, it determines whether boot mode ID code protection is enabled or disabled.

When boot mode ID code protection is disabled, all blocks in the user area and data area are erased.
When all blocks are successfully erased, the MCU sends a response (06h) and enters the program/erase host command wait state. If not all blocks are successfully erased, the MCU sends an error response.
When boot mode ID code protection is enabled, the MCU sends a response (16h) and enters boot mode ID code authentication state.

Command	40h

Response	ACK

ACK (1 byte): ACK code
06h: ID code protection is disabled.
16h: ID code protection is enabled.

Error response	C0h	Error

Error (1 byte): Error code
51h: Erase error

Downloaded from Arrow.com.

50.10.7 ID Code Authentication Command

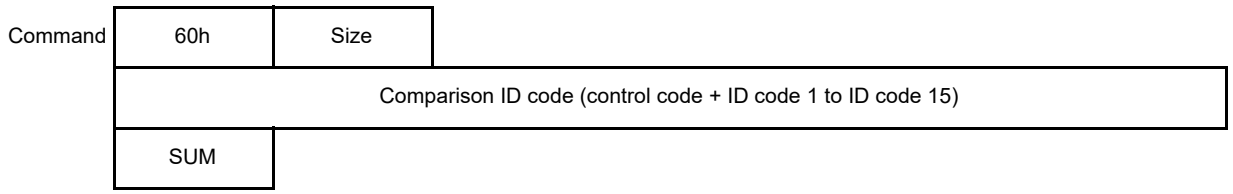
This command is used for ID code authentication when boot mode ID code protection is enabled.
Table 50.19 lists ID code authentication command. This command can be used only in the boot mode ID code authentication state.

Table 50.19 ID Code Authentication Command

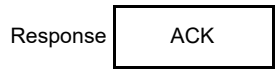
Command	Function
ID code check	Compare the 16-byte code sent from the host and ID code.

50.10.7.1 ID Code Check

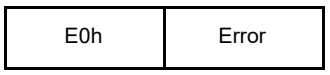
This command is used to unlock boot mode ID code protection.
The comparison ID code specified with the command should be set to the same value as the control code and ID code 1 to ID code 15.
If the comparison ID code sent from the host matches the ID code programmed in the user area, the MCU sends a response (06h) and enters program/erase host command wait state.
If the codes do not match or the SUM of the received command does not match, the MCU sends an error response.
When the ID codes do not match three times consecutively while the control code is 45h, all blocks in the user area and data area are erased. If an error occurs during erasure, the MCU sends an error response.
Also, even if all blocks are successfully erased, the MCU sends an error response and continues the boot mode ID code state. Reset the MCU to enter the program/erase host command wait state.



Size (1 byte): Number of bytes of ID codes (the value is always 10h)
ID code (16 bytes): Control code (1 byte) + ID code 1 to ID code 15 (15 bytes)
SUM (1 byte): Value that is calculated so the sum of the command data is 00h



ACK (1 byte): ACK code
06h: The MCU enters the program/erase host command wait state.



Error (1 byte): Error code
11h: SUM error
61h: ID codes do not match
63h: ID codes do not match and erase error

Downloaded from Arrow.com.

50.10.8 Program/Erase Commands

Program/erase commands are used to program or erase the user area or data area based on the response to inquiry commands. Table 50.20 lists commands used in each of the program/erase host command wait state, program wait state, and erase wait state. Table 50.21 lists commands that can be accepted in each state.
When a command that is not listed in Table 50.21 is received in each state, the MCU sends a command error response.

Table 50.20 Program/Erase Commands

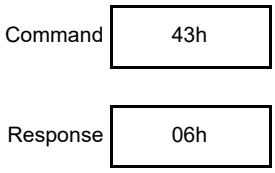
Command	Function
User/data area program preparation	Select the user area or data area to program, and enter the program wait state.
Program	Program the specified data to the selected area in the user area or data area. Or enter the program/erase host command wait state (end of program).
Data area program	Program the specified-size data to the selected area in the data area. Or enter the program/erase host command wait state (end of program).
Erase preparation	Enter the erase wait state.
Block erase	Erase the selected block, or enter the program/erase host command wait state (end of erase).

Table 50.21 Acceptable Commands for Each State

State	Acceptable Command
Program/erase host command wait state	User/data area program preparation command, and erase preparation command
Program wait state	Program command and data area program command
Erase wait state	Block erase command

50.10.8.1 User/Data Area Program Preparation

This command is used to prepare for accepting the program command and the data area program command.
When the MCU receives this command, it recognizes that an instruction to prepare for the program command is issued from the host. Then, the MCU enters the program wait state, where only the program command and the data area program command can be accepted, and sends a response (06h).

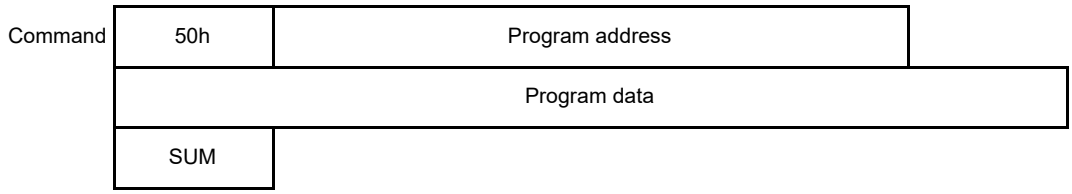


Downloaded from Arrow.com.

50.10.8.2 Program

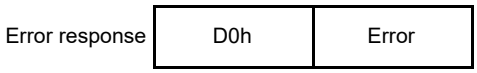
This command is used to program the specified data to the user area or data area. Set the lower 8 bits to 0 for the program address selected in this command. When the data length is shorter than 256 bytes, the data cannot be programmed. Fill the gaps with FFh.
When the program from the selected address is successfully completed, the MCU sends a response (06h). If the SUM of the received command does not match or an error occurs during a program operation, the MCU sends an error response.

To enter the program/erase host command wait state after the program operation ends, send 50h FFh FFh FFh FFh B4h from the host. The MCU sends a response (06h), and enters the program/erase host command wait state.



Program address (4 bytes): Address for program destination
Set the lower 8 bits to 0
Set FFFF FFFFh for end of program
Program data (n bytes): Program data (n = 256, 0 for end of program)
When the program data is less than n bytes, set FFh for the missing data.
No program data for the end of program
SUM (1 byte): Value that is calculated so the sum of command data is 00h





Error (1 byte): Error code
11h: SUM error
2Ah: Address error (the address is not in the selected area.)
53h: Program error (the data cannot be programmed.)

Downloaded from Arrow.com.

50.10.8.3 Data Area Program

This command is used to program the specified data to the data area. Set the lower 2 bits to 0 for the program address selected in this command. When the data length is shorter than 4 bytes, the data cannot be programmed. Fill the gaps with FFh.
When the program from the selected address is successfully completed, the MCU sends a response (06h). If the SUM of the received command does not match or an error occurs during a program operation, the MCU sends an error response.

To enter the program/erase host command wait state after the program operation ends, send 51h FFh FFh FFh FFh 00h B3h from the host. The MCU sends a response (06h), and enters the program/erase host command wait state.

Command	51h	Program address	Program data length
	Program data		
	SUM		

Program address (4 bytes): Address for program destination
Set the lower 2 bits of the selected address to 0
Set FFFF FFFFh for end of data area program
Program data length (1 byte): Size of program data
Set 4-byte data
Set 00h for end of data area program
Program data (n bytes): Program data for the data area (n = program data length, 0 for end of program)
Set data of the program data length
When the program is less than n bytes, set FFh for the missing data.
No program data for the end of data area program
SUM (1 byte): Value that is calculated so the sum of command data is 00h

Response	06h

Error response	D1h	Error

Error (1 byte): Error code
11h: SUM error
2Ah: Address error (the address is not in the selected area.)
2Bh: Program data length error
53h: Program error (the data or program data cannot be programmed.)

Downloaded from Arrow.com.

50.10.8.4 Erase Preparation

This command is used to prepare for accepting the block erase command.
When the MCU receives this command, it recognizes that an instruction to prepare for the erase command is issued from the host. Then, the MCU enters the erase wait state, where only the block erase command can be accepted, and sends a response (06h).

Command	48h
Response	06h

50.10.8.5 Block Erase

This command is used to erase the selected block in the user area or data area.
Specify the block start address selected in the command by calculating the address based on the response to the block information inquiry command.
When the block selected in the block start address is successfully erased, the MCU sends a response (06h). If the SUM of the received command does not match or an error occurs during an erase operation, the MCU sends an error response.

To enter the program/erase host command wait state after the erase operation ends, send 59h 04h FFh FFh FFh FFh A7h from the host. The MCU enters the program/erase host command wait state and sends a response (06h).

Command	59h	Size
	Block start address	
Response	SUM	

Size (1 byte): Total bytes of Block start address (the value is always 04h)
Block start address (4 bytes): Start address of the block that is erased
Set FFFF FFFFh for end of erase
SUM (1 byte): Value that is calculated so the sum of response data is 00h

Response	06h	
Error response	D9h	Error

Error (1 byte): Error code
11h: SUM error
29h: Block start address error
51h: Erase error (the selected block cannot be erased)

Downloaded from Arrow.com.

50.10.9 Read-Check Commands

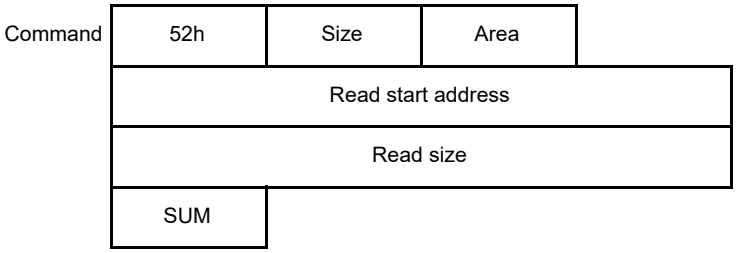
Read-check commands are used to read data or check whether data is programmed in the user area or data area in the MCU based on the response to inquiry commands.
Table 50.22 lists read-check commands used in the program/erase host command wait state.

Table 50.22 Read-Check Commands

Command	Function
Memory read	Read data from the user area or data area.
User area checksum	Obtain the checksum of the entire user area.
Data area checksum	Obtain the checksum of the entire data area.
User area blank check	Check whether data is programmed in the user area.
Data area blank check	Check whether data is programmed in the data area.
Access window information program	Set the access window.
Access window read	Read the settings of the access window.

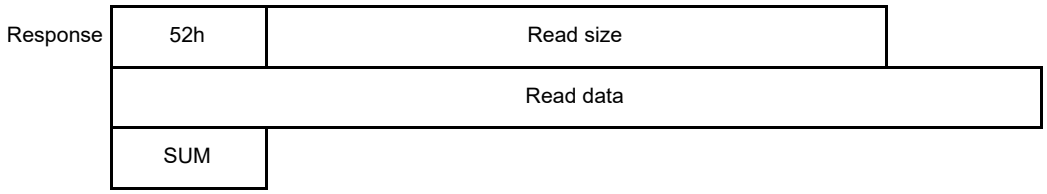
50.10.9.1 Memory Read

This command is used to read data programmed in the user area or data area.
For a read start address selected in the command, set a value within the range from the area start address to the area end address received in the response to the user area information inquiry command or the data area information inquiry command.
For a read size selected in the command, set a value so the sum of the read start address and the read size is within the range from the area start address to the area end address received in the response to the user area information inquiry command or the data area information inquiry command.
When the MCU performs a read successfully, it sends data of the specified range. If the SUM of the received command does not match or the MCU fails to perform a read successfully, it sends an error response.

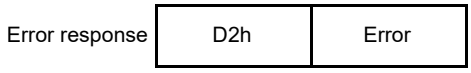


Size (1 byte): Total bytes for Read start address and Read size
Area (1 byte): Area that is read
01h: User area or data area
Read start address (4 bytes): Start address of the area that is read
Read size (4 bytes): Size of data that is read (in bytes)
SUM (1 byte): Value that is calculated so the sum of response data is 00h

Downloaded from Arrow.com.



Read size (4 bytes): Size of Data that is read (in bytes)
Read data (n bytes): Data read from the specified range (n = read size)
SUM (1 byte): Value that is calculated so the sum of response data is 00h

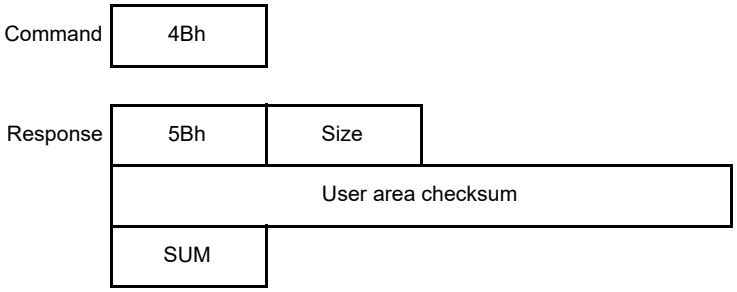


Error (1 byte): Error code
11h: SUM error
2Ah: Address error
- A value other than 01h is set for the “Area” field.
- The read start address is not in the selected area.

2Bh: Size error
- The read size is set to 0000 0000h.
- The read size exceeds the area size.
- The address calculated from the read start address and read size is not in the selected area.

50.10.9.2 User Area Checksum

This command used to obtain the checksum of the entire user area.
When the MCU receives this command, it adds data from the start address to the end address in bytes in the user area, and sends the calculated result (checksum) as a response.

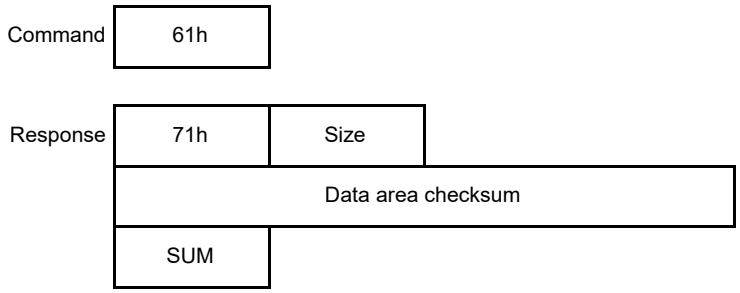


Size (1 byte): Number of bytes for checksum of the user area (the value is always 04h)
User area checksum (4 bytes): Calculated result of the data in the user area in bytes
SUM (1 byte): Value that is calculated so the sum of response data is 00h

Downloaded from Arrow.com.

50.10.9.3 Data Area Checksum

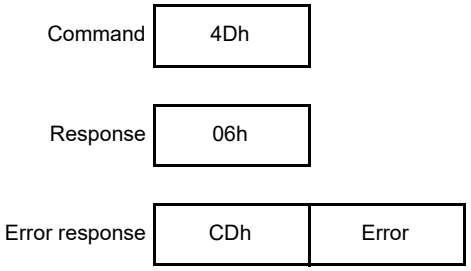
This command used to obtain the checksum of the entire data area.
When the MCU receives this command, it adds data from the start address to the end address in bytes in the data area, and sends the calculated result (checksum) as a response.



Size (1 byte): Number of bytes for checksum of the data area (the value is always 04h)
Data area checksum (4 bytes): Calculated result of the data in the data area in bytes
SUM (1 byte): Value that is calculated so the sum of response data is 00h

50.10.9.4 User Area Blank Check

This command is used to check whether data is programmed in the user area.
When the MCU receives this command, it checks whether there is data in the entire user area. If there is no programmed data, the MCU sends a response (06h). If there is at least 1 byte of programmed data, the MCU sends an error response.



Error (1 byte): Error code
52h: Not blank

Downloaded from Arrow.com.

50.10.9.5 Data Area Blank Check

This command is used to check whether data is programmed in the user area.
When the MCU receives this command, it checks whether there is programmed data in the entire user area. If there is no programmed data, the MCU sends a response (06h). If there is at least 1 byte of programmed data, the MCU sends an error response.

Command	62h	
Response	06h	
Error response	E2h	Error

Error (1 byte): Error code
52h: Not blank

50.10.9.6 Access Window Information Program

This command is used to set the access window used for area protection.
For the access window start address selected in the command, set the start address of the start block. For the access window end address, set the end address of the end block.
When the specified access window settings are successfully completed, the MCU sends a response (06h). If the SUM of the received command does not match or an error occurs during the access window settings, the MCU sends an error response.
For details on the access window, see section 50.6, Area Protection.

Command	74h	05h	Access window	
	Access window start address LH	Access window start address HL	Access window end address LH	Access window end address HL
	SUM			

Access window (1 byte): Set the access window or clear the access window settings
Set 00h to set the access window
Set FFh to clear the access window settings
Access window start address LH (1 byte): Start address of the access window (A15 to A8)
Set A15 to A8 of the start address of the start block.
Set FFh to clear the access window settings
Access window start address HL (1 byte): Start address of the access window (A23 to A16)
Set A23 to A16 of the start address of the start block.
Set FFh to clear the access window settings
Access window end address LH (1 byte): End address of the access window (A15 to A8)
Set A15 to A8 of the end address of the end block.
Set FFh to clear the access window settings
Access window end address HL (1 byte): End address of the access window (A23 to A16)
Set A23 to A16 of the end address of the end block.
Set FFh to clear the access window settings
SUM (1 byte): Value that is calculated so the sum of response data is 00h

Downloaded from Arrow.com.

Response	06h	
Error response	F4h	Error

Error (1 byte): Error code
11h: SUM error
2Ah: Address error (specified address is not in the area)
53h: Program error (access window cannot be set)

50.10.9.7 Access Window Read

This command is used to check the set range of the access window.
When the MCU successfully obtains the access window range, the MCU sends the access window start address and end address that it read. If the SUM of the received command does not match, the MCU sends an error response.

Command	73h	01h	FFh	8Dh

Response	73h	05h		
	Access window start address LH	Access window start address HL	Access window end address LH	Access window end address HL
	FFh			
	SUM			

Access window start address LH (1 byte): Start address of the access window range (A15 to A8)
Access window start address HL (1 byte): Start address of the access window range (A23 to A16)
Access window end address LH (1 byte): End address of the access window range (A15 to A8)
Access window end address HL (1 byte): End address of the access window range (A23 to A16)
SUM (1 byte): Value that is calculated so the sum of response data is 00h

Error response	F3h	Error

Error (1 byte): Error code
11h: SUM error

Downloaded from Arrow.com.

50.11 Serial Programmer Operation in Boot Mode (SCI Interface)

The following describes the procedure for the serial programmer to program/erase the user area and data area in boot mode (SCI Interface).

1. Automatically adjust the bit rate
2. Receive the MCU information*1
3. Select the device and change the bit rate
4. Enter the program/erase host command wait state
5. Unlock boot mode ID code protection
6. Erase the user area and data area*2, *3
7. Program the user area and data area*2, *3
8. Check data in the user area*2
9. Check data in the data area*2
10. Set the access window in the user area*2
11. Reset the MCU

Note 1. If the necessary information has been already received, step 2 can be skipped.
Note 2. Processing steps from 6 to 10 can be proceeded as necessary, and their order can be changed.
Note 3. When a timeout occurs or invalid response data is received, stop the operation and perform step 11 (reset the MCU).

Refer to section 50.10.5, Inquiry Commands, section 50.10.6, Setting Commands, section 50.10.7, ID Code Authentication Command, section 50.10.8, Program/Erase Commands, and section 50.10.9, Read-Check Commands for details on the commands used in the above steps 2 to 10.

Downloaded from Arrow.com.

50.11.1 Bit Rate Automatic Adjustment

The MCU measures the low width of data 00h that is sent from the serial programmer at 9,600 or 19,200 bps to automatically adjust the bit rate.

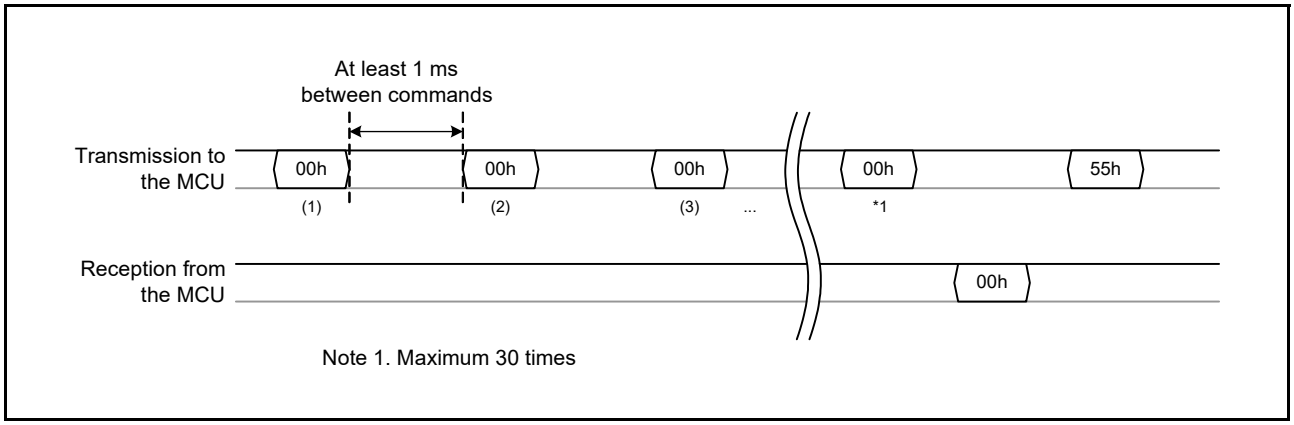


Figure 50.31 Transmit/Receive Data for Bit Rate Automatic Adjustment

After starting up in boot mode, wait for at least 400 ms and then send 00h to the MCU from the serial programmer. When the bit rate adjustment is completed, the MCU sends 00h to the programmer. When the programmer receives 00h, send 55h to the MCU from the programmer. When the programmer can not receive 00h, wait for at least 1 ms and send 00h to the MCU again. When the programmer fails to receive 00h even if it send 00h 30 times, restart the MCU in boot mode and perform the automatic adjustment for the bit rate again.

When the MCU receives 55h, the MCU sends E6h and enters the inquiry/setting command wait state. If the MCU fails to receive 55h, the MCU sends FFh. When the programmer receives FFh, restart the MCU in boot mode, and perform the automatic adjustment for the bit rate again.

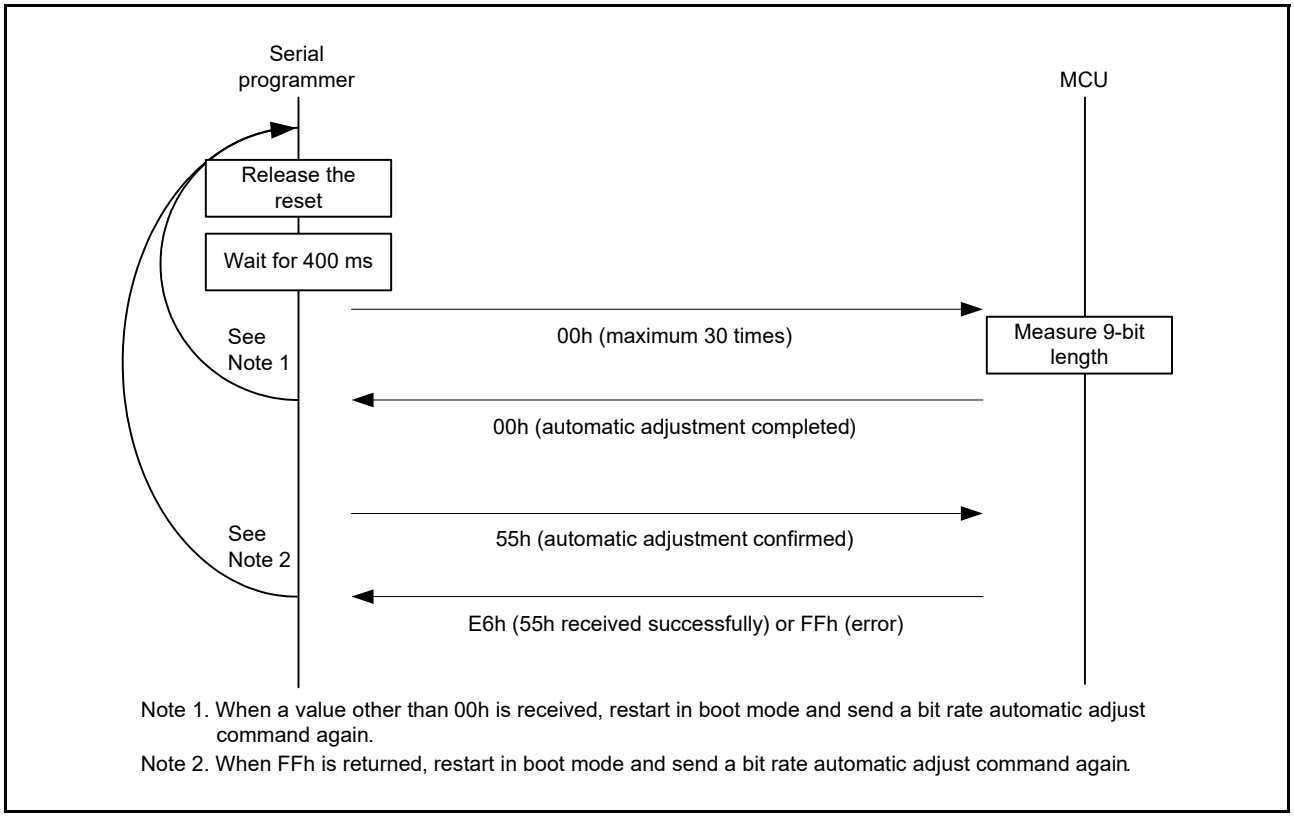


Figure 50.32 Bit Rate Automatic Adjustment Procedure

Downloaded from Arrow.com.

50.11.2 Receive the MCU Information

Procedure to send inquiry commands, and receive the information necessary to send setting commands, program/erase commands, and read-check commands is as follows.

(1) Send a support device inquiry command (20h) to check what type of endianness the MCU supports. The MCU returns all device codes and series names that it supports.
(2) Send a user area information inquiry command (25h) to check the start and end addresses of the user area. The MCU returns the start and end addresses of the user area.
(3) Send a block information inquiry command (26h) to check the block configuration. The MCU returns the start address, the size of one block, and the number of blocks for the user area and data area.
(4) Send a data area information inquiry command (2Bh) to check the start and end addresses of the data area. The MCU returns the start and end addresses of the data area.

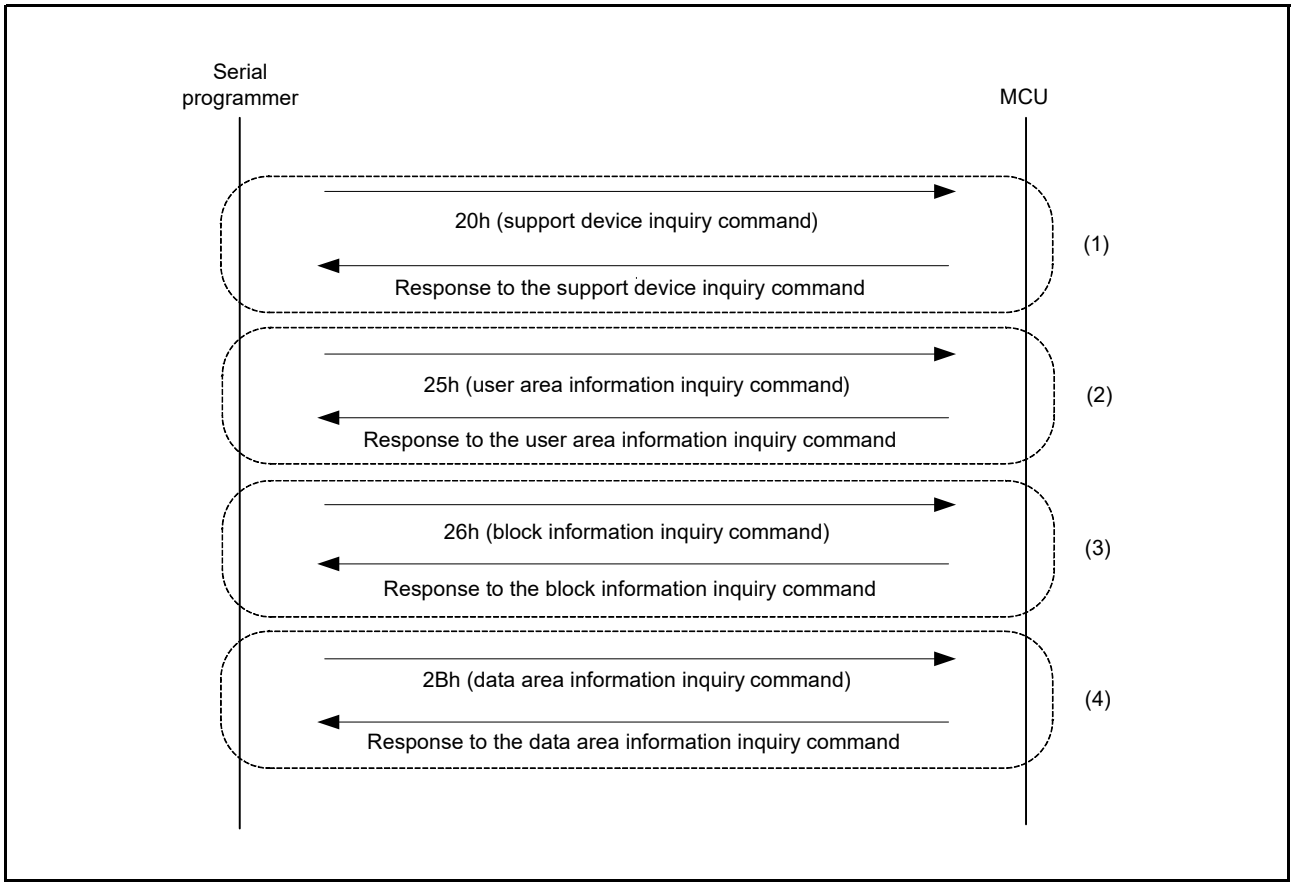


Figure 50.33 Procedure to Receive the MCU Information

Downloaded from Arrow.com.

50.11.3 Select the Device and Change the Bit Rate

Procedure to select the device to connect with the serial programmer and to change the bit rate for communication is as follows.

(1) Send the device select command (10h). Select the device code according to the endian of developed software.
(2) Send the operating frequency select command (3Fh) to change the communication bit rate from 9,600 or 19,200 bps.

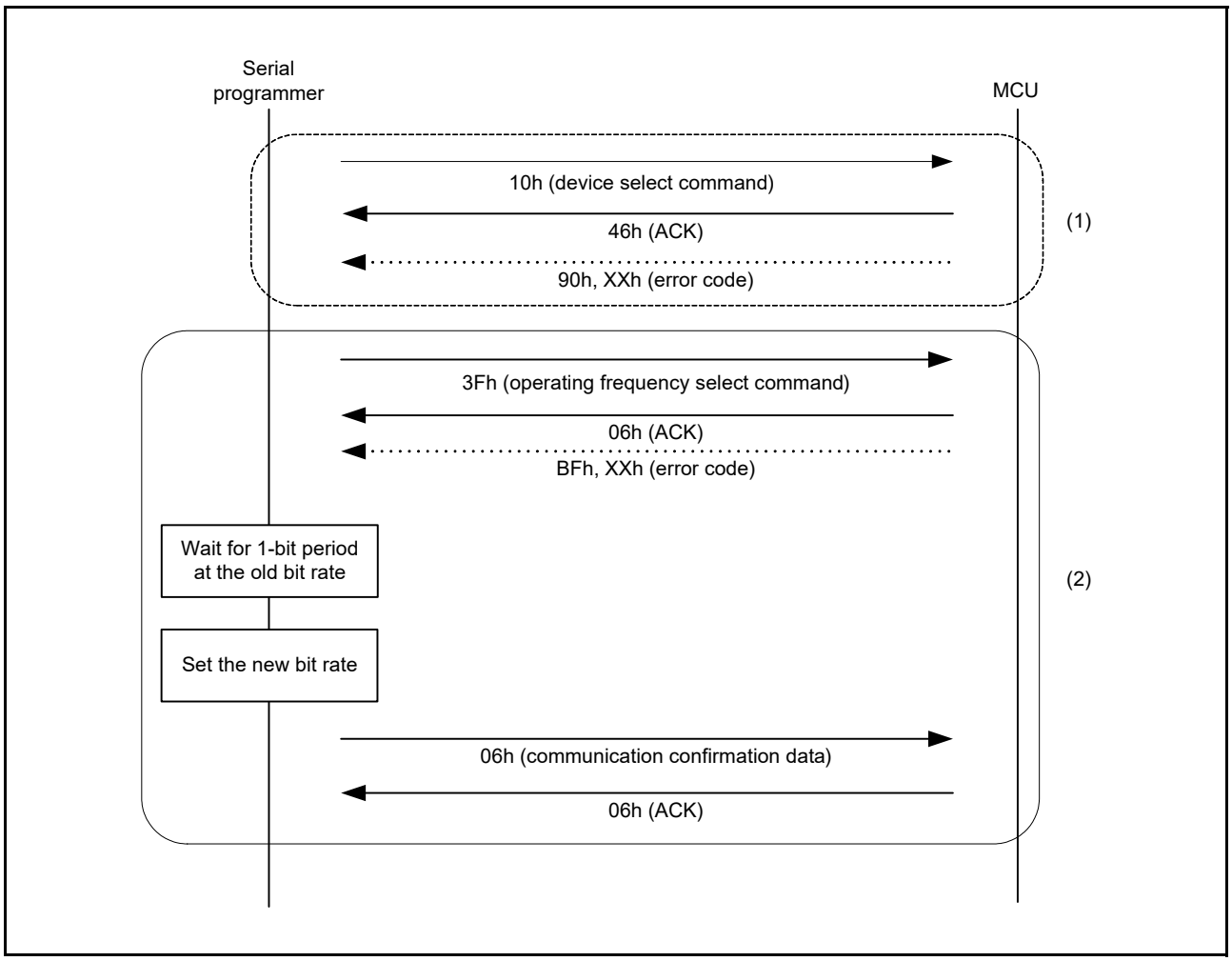


Figure 50.34 Procedure to Select the Device and Change the Bit Rate

Downloaded from Arrow.com.

50.11.4 Enter the Program/Erase Host Command Wait State

Send the program/erase host command wait state transition command to perform program/erase operations. The MCU sends a response according to whether boot mode ID code protection is enabled or disabled.

(1) When boot mode ID code protection is disabled, the MCU sends a response (06h), and enters the program/erase host command wait state. Use the serial programmer to start from the operation described in section 50.11.6, Erase the User Area and Data Area.

(2) When the boot mode ID code protection is enabled, the MCU sends a response (16h), and enters the ID code authentication wait state. Use the serial programmer to start from the operation described in section 50.11.5, Unlock Boot Mode ID Code Protection.

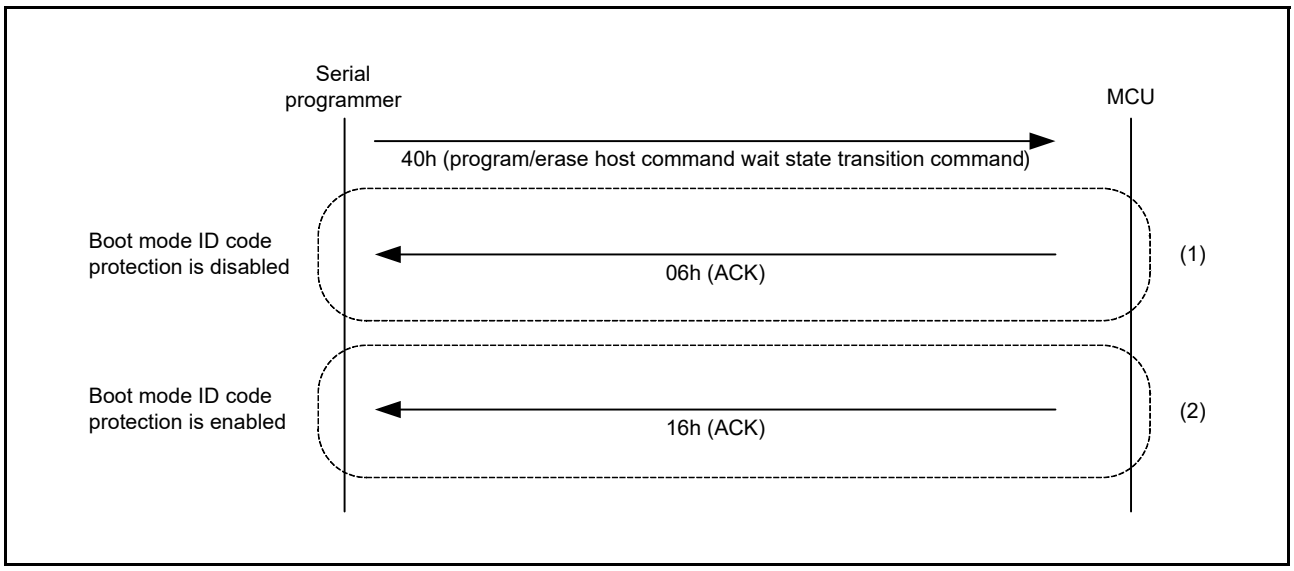


Figure 50.35 Procedure to Transition to the Program/Erase Host Command Wait State

Downloaded from Arrow.com.

50.11.5 Unlock Boot Mode ID Code Protection

Send the ID code check command to unlock boot mode ID code protection.

(1) When ID codes match, the MCU enters the program/erase host command wait state. Data in the user area and data area are not erased. Use the serial programmer to start from the operation described in section 50.11.6, Erase the User Area and Data Area.

(2) If ID codes do not match consecutively, the MCU remains in the boot mode ID code authentication state. Reset the MCU, and then use the serial programmer to start again from section 50.11.1, Bit Rate Automatic Adjustment.

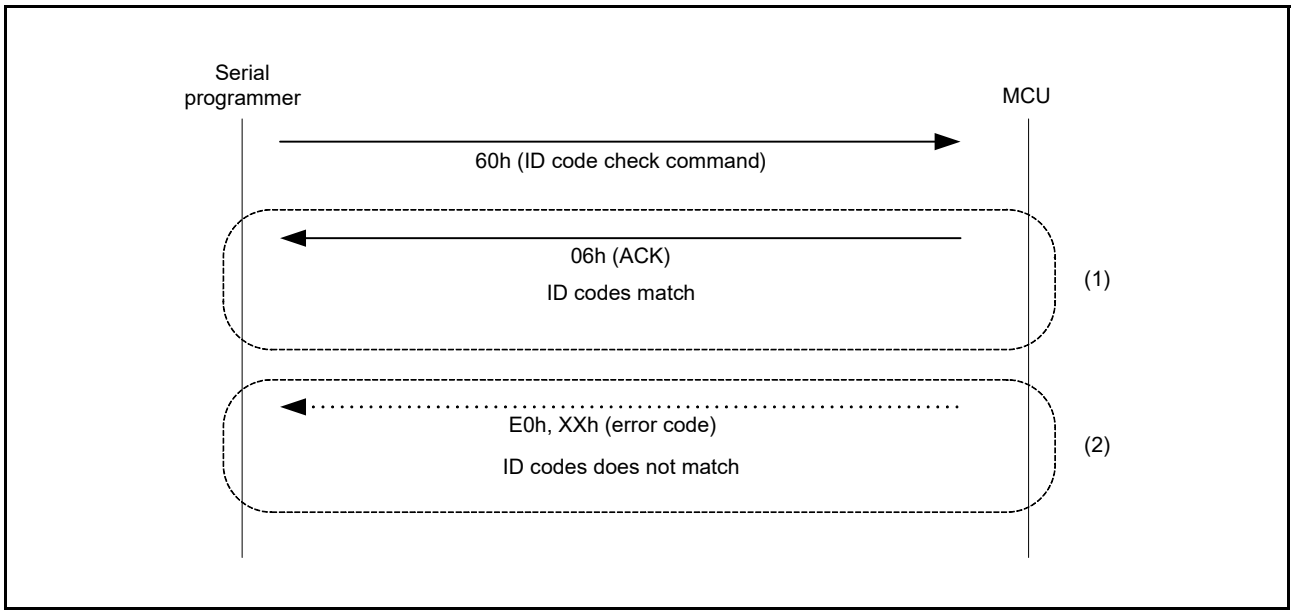


Figure 50.36 Procedure to Unlock ID Code Protection

Downloaded from Arrow.com.

50.11.6 Erase the User Area and Data Area

Procedure to erase blocks that are programmed in the user area and data area to program a user program and data is as follows.

(1) Send an erase preparation command (48h).
(2) Send a block erase command (59h).
(3) To place the MCU in the program/erase host command wait state, send a block erase command for ending the erasure (59h 04h FFh FFh FFh FFh A7h).

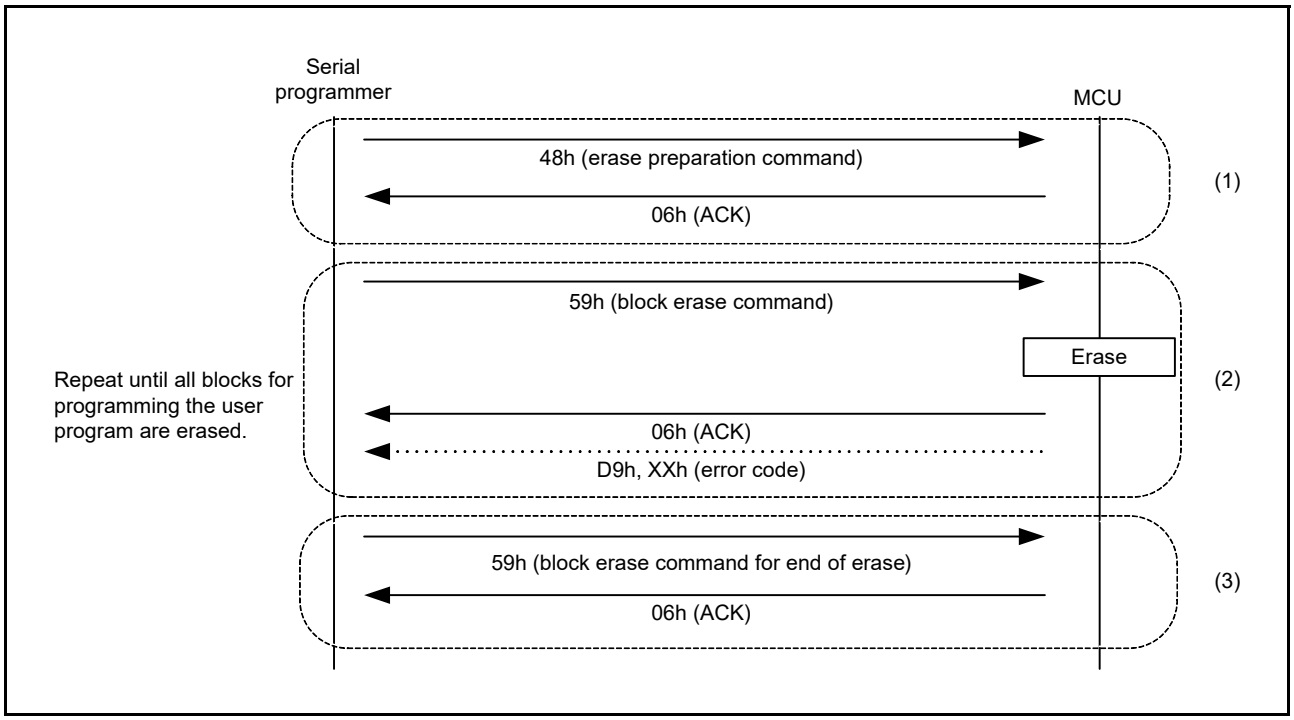


Figure 50.37 Procedure to Erase the User Area and Data Area

Downloaded from Arrow.com.

50.11.7 Program the User Area and Data Area

Procedure to program a user program and data in the user area and data area is as follows.

(1) Send the user/data area program preparation command (43h).
(2) Send the program command (50h) or the data area program command (51h).
(3) To place the MCU in the program/erase host command wait state, send the program command (50h FFh FFh FFh FFh B4h) or the data area program command (51h FFh FFh FFh FFh 00h B3h) for ending the programming.

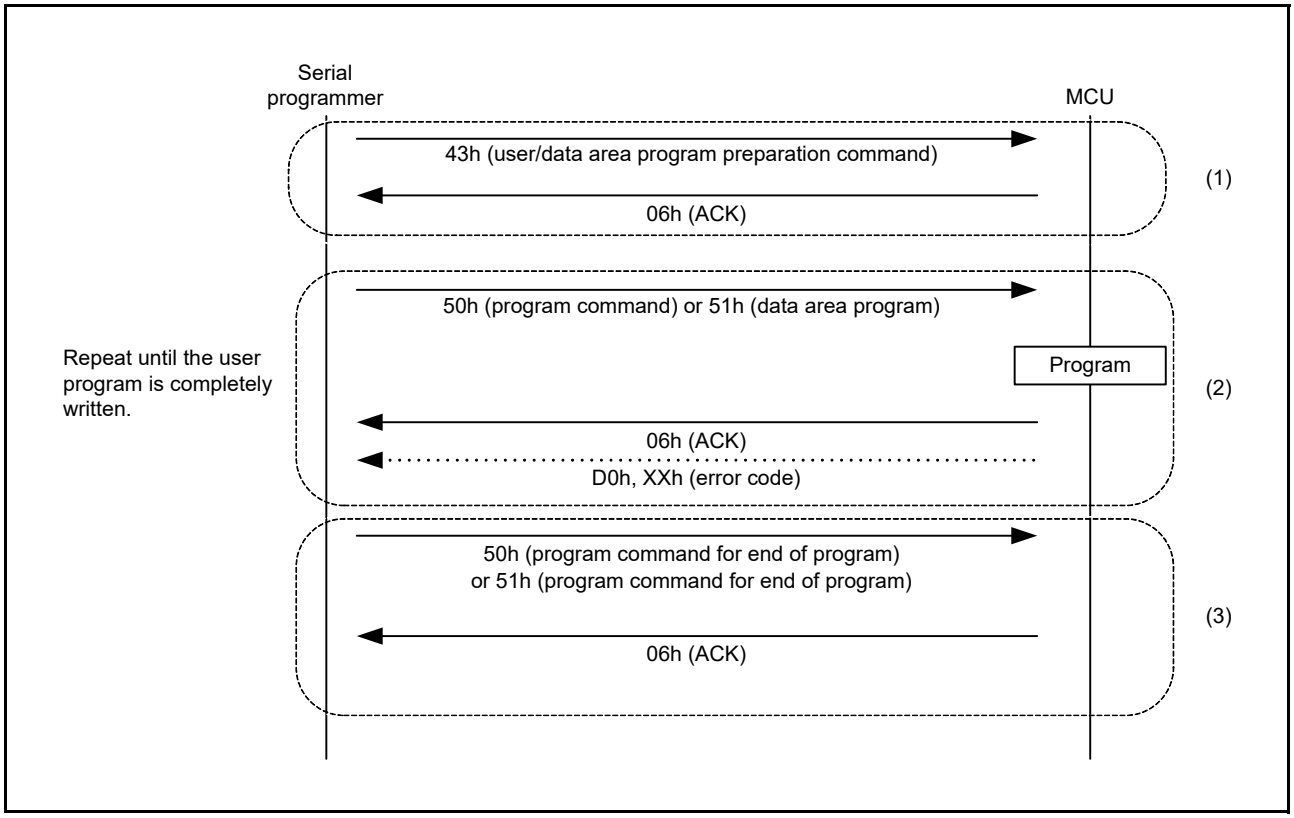


Figure 50.38 Procedure to Program the User Area and Data Area

Downloaded from Arrow.com.

50.11.8 Check Data in the User Area

Procedure to read and check, checksum, and blank check the user area to check the programmed data in the user area is as follows.

(1) The read and check operation is used to read data in the user area and compare the read data with the programmed data to check if the program operation is performed successfully. Send a memory read command (52h) to read data in the user area.
(2) Send the user area checksum command (4Bh) to check program data using the checksum of user area.
(3) Send a user area blank check command (4Dh) to check if the user area has data.

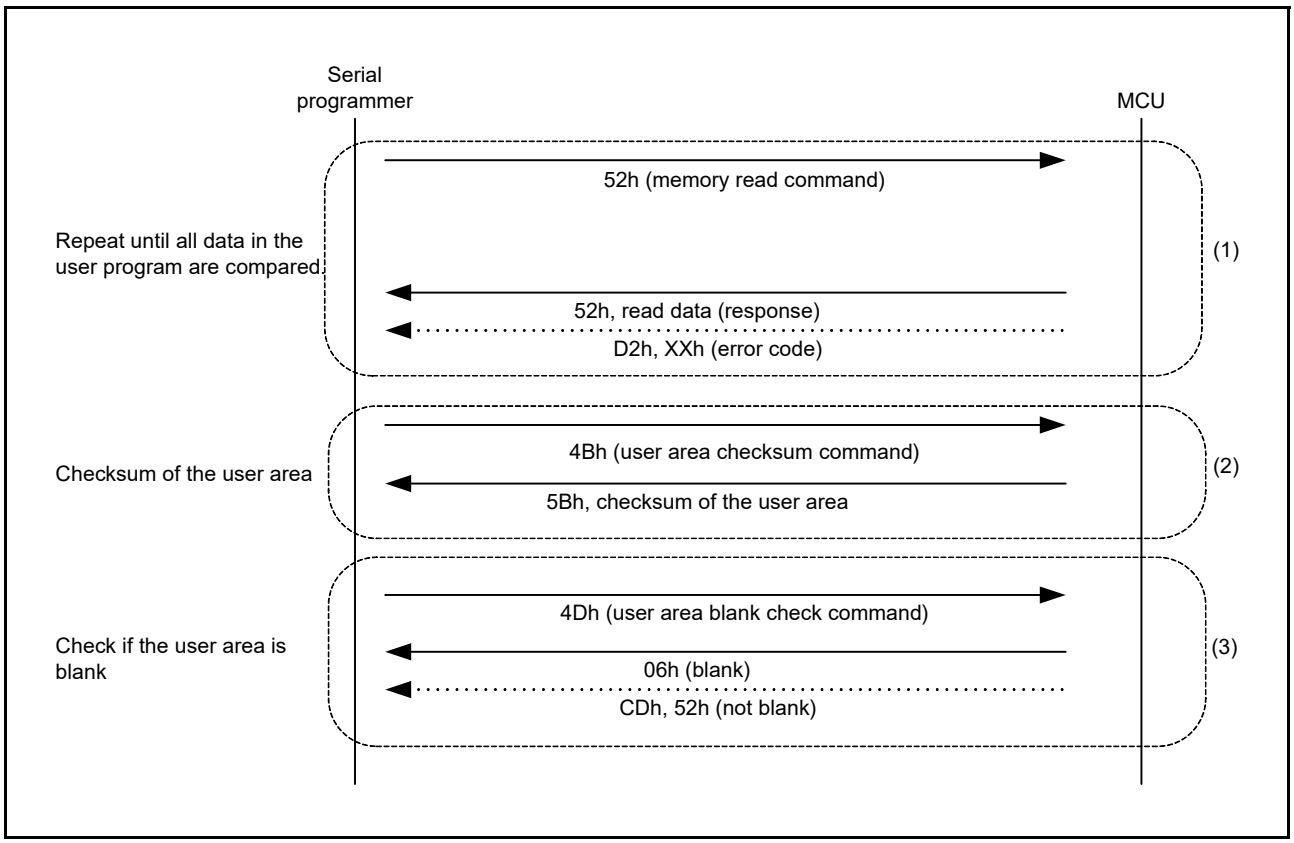


Figure 50.39 Procedure to Check Data in the User Area

Downloaded from Arrow.com.

50.11.9 Check Data in the Data Area

Procedure to read and check, checksum, and blank check the data area to check the programmed data in the data area is as follows.

(1) The read and check operation is used to read data in the data area and compare the read data with the programmed data to check if the program operation is performed successfully. Send a memory read command (52h) to read data in the data area.
(2) Send the data area checksum command (61h) to check program data using the checksum of data area.
(3) Send the data area blank check command (62h) to check if the data area has data.

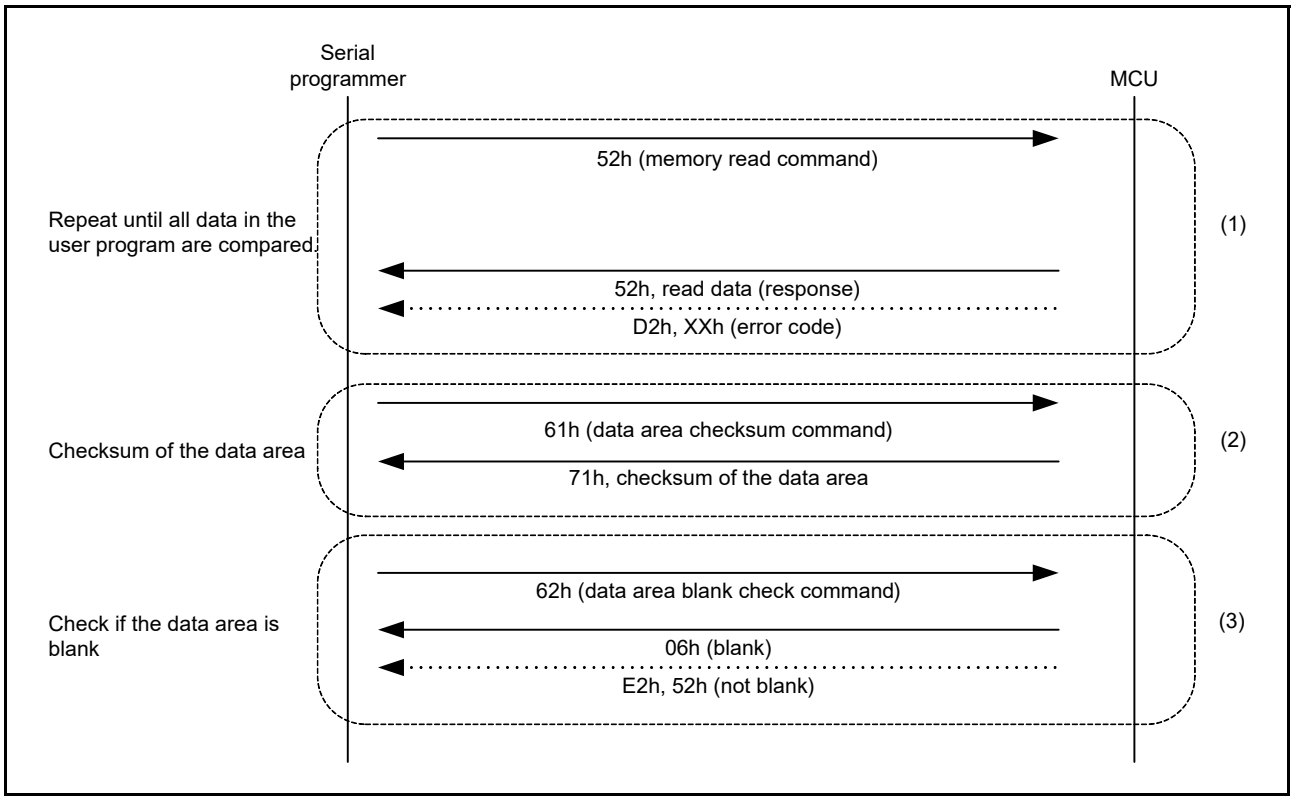


Figure 50.40 Procedure to Check Data in the Data Area

Downloaded from Arrow.com.

50.11.10 Set the Access Window in the User Area

Procedure to set the access window to avoid unintentionally rewriting the user area during the self-programming is as follows.

(1) Send the access window program command (74h) to set the access window settings.
(2) Send the access window read command (73h) to confirm the access window settings.

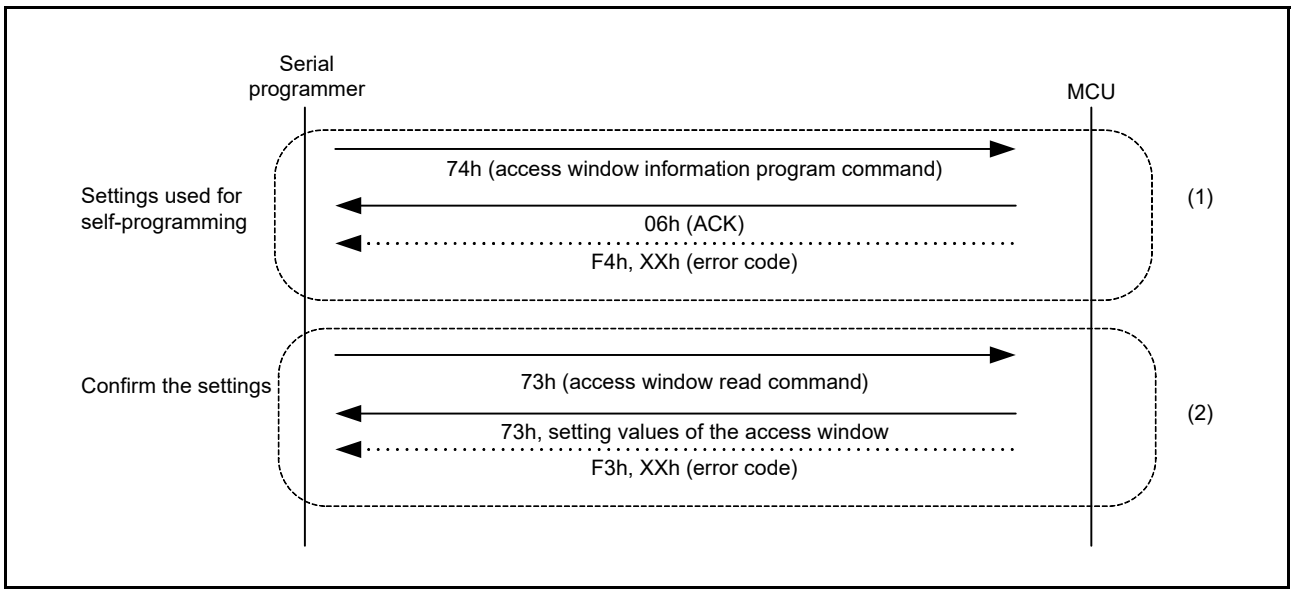


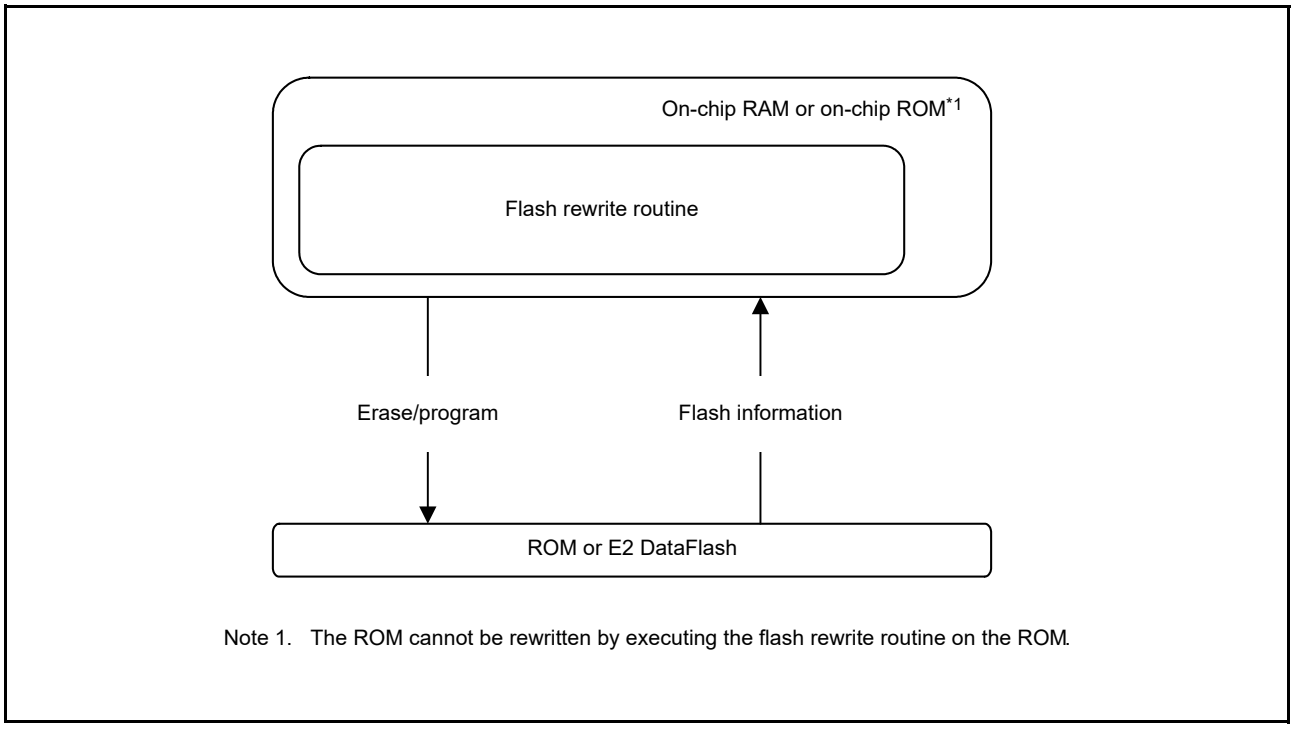
Figure 50.41 Procedure to Set the Access Window in the User Area

Downloaded from Arrow.com.

50.12 Rewriting by Self-Programming

50.12.1 Overview

The MCU supports rewriting of the flash memory by the user program. The ROM and E2 DataFlash can be rewritten by preparing a routine to rewrite the flash memory (flash rewrite routine) in the user program.
When rewriting the E2 DataFlash, the BGO can be used to execute the flash rewrite routine on the ROM. The E2 DataFlash can also be rewritten by executing the flash rewrite routine that is transferred on the RAM in advance.



Note 1. The ROM cannot be rewritten by executing the flash rewrite routine on the ROM.

Figure 50.42 Self-Programming Overview

Downloaded from Arrow.com.

50.13 Usage Notes

(1) Access the Block Where Erase Operation is Forcibly Stopped
When forcibly stopping an erase operation, data in the block where the erase operation is aborted is undefined. To avoid malfunctions caused by reading undefined data, do not execute instructions or read data in the block where an erase operation is forcibly stopped.

(2) Processing After Forced Stop of Erase Operation
When an erase operation is forcibly stopped, issue a block erase command again to the same block.

(3) Additional Programming Disabled
The same address cannot be programmed more than once. When programming an area that has been already programmed, erase the area first.

(4) Reset during Program/Erase
If inputting a reset from the RES# pin, release the reset after reset input time of at least tRESW (refer to section 51, Electrical Characteristics) within the range of the operating voltage defined in the electrical characteristics.
The IWDT reset and software reset can be used regardless of tRESW.

(5) Location of Interrupt Vectors and Exception Vectors during Program/Erase Operation
When an interrupt or an exception occurs during a program/erase operation, the vector may be fetched from the ROM. To avoid fetching the vector from the ROM, allocate the interrupt vector table and exception vector table to the area other than the ROM with the INTB and EXTB registers in the CPU.

(6) Program/Erase in Low-Speed Operating Mode
Do not program or erase the flash memory when low-speed operating mode is selected by the SOPCCR register for low-power consumption functions.

(7) Abnormal Termination during Program/Erase
When the voltage exceeds the range of the operating voltage during a program/erase operation or when a program/erase operation is not completed successfully due to a reset or prohibited actions described in (8), erase the area again.

(8) Actions Prohibited during Program/Erase
To prevent the damage to the flash memory, comply with the following instructions.

- Do not use the MCU power supply that is outside the operating voltage range.
- Do not update the value of the OPCCR.OPCM[2:0] bits.
- Do not update the value of the SOPCCR.SOPCM bit.
- Do not change the clock source select bit in the SCKCR3 register.
- Do not enable switching clock sources by setting the RSTCKCR.RSTCKEN bit when exiting sleep mode.
- Do not change the division ratio of the flash interface clock (FCLK).
- Do not place the MCU in deep sleep mode or software standby mode.
- Do not access the E2 DataFlash during a program/erase operation to the ROM.
- Do not change the DFLCTL.DFLEN bit value during a program/erase operation to the E2 DataFlash.

(9) FCLK during Program/Erase
For programming/erasure by self-programming, set the frequency of the FlashIF clock (FCLK), and specify an integer FCLK frequency (MHz) in FISR.PCKA[4:0] bits. Note that when the FCLK is 4 to 32 MHz, a rounded-up value should be set for a non-integer frequency such as 12.5 MHz (i.e. 12.5 MHz should be set rounded up to 13 MHz). If the FCLK is equal to or less than 4 MHz, only 1, 2, 3, or 4 MHz can be used.

Downloaded from Arrow.com.

50.14 Usage Notes in Boot Mode

(1) Notes on Communication Errors in Boot Mode
When communication with the MCU cannot be performed properly, reset and start up in boot mode again.

(2) Notes on Power Supply Voltage in Boot Mode (SCI Interface)
When the bit rate exceeds 500 kbps in boot mode (SCI Interface), use a voltage that is 3.0 V or higher.

(3) Notes on Option-Setting Memory in Boot Mode
The settings of option function select register 0 (OFS0), option function select register 1 (OFS1), and endian select register (MDE) are disabled in boot mode.

(4) Notes on Clocks in Boot Mode (USB Interface)
When USB interface mode is selected, externally input a clock to the EXTAL or XTAL pin, or connect a crystal or ceramic resonator to supply a clock.
Use a 4, 6, 8, 12, or 16 MHz external clock in boot mode (USB interface). An clock other than a 4, 6, 8, 12, or 16 MHz external clock cannot be used.

(5) Notes on Power Supply Voltage in Boot Mode (USB Interface)
Use a voltage between 3.0 V and 3.6 V in boot mode (USB interface). A voltage that is lower than 3.0 V cannot be used.

(6) Notes on Switching the Start-Up Area
Switch the start-up area by self-programming.

Downloaded from Arrow.com.

51. Electrical Characteristics

51.1 Absolute Maximum Ratings

Table 51.1 Absolute Maximum Ratings

Conditions: VSS = AVSS0 = VREFL0 = VSS_USB = VSS_RF = 0 V

Item		Symbol	Value	Unit
Power supply voltage		VCC, VCC_USB	–0.3 to +4.0	V
VBATT power supply voltage		VBATT	–0.3 to +4.0	V
Input voltage	Ports for 5 V tolerant*1	V_{in}	–0.3 to +6.5	V
	P03, P05, P07, P40 to P47		–0.3 to AVCC0 + 0.3	
	ANT		–1.0 to +1.4	
	XTAL1_RF, XTAL2_RF		–0.3 to +1.4	
	DCLIN_A, DCLIN_D		–0.3 to +2.2	
	Ports other than above*2		–0.3 to VCC + 0.3	
Reference power supply voltage		VREFH0	–0.3 to AVCC0 + 0.3	V
Analog power supply voltage		AVCC0	–0.3 to +4.0	V
		VCC_RF	–0.3 to +4.0	V
		AVCC_RF	–0.3 to +4.0	V
Analog input voltage	When AN000 to AN007 are used	V_{AN}	–0.3 to AVCC0 + 0.3	V
	When AN016 to AN020, AN027 are used		–0.3 to VCC + 0.3	
Operating temperature		T_{opr}	–40 to +85	°C
Storage temperature		T_{stg}	–55 to +125	°C

Caution: Permanent damage to the MCU may be caused if absolute maximum ratings are exceeded.
To preclude any malfunctions due to noise interference, insert capacitors with high frequency characteristics between the VCC and VSS pins, between the AVCC0 and AVSS0 pins, between the VCC_USB and VSS_USB pins, between the VREFH0 and VREFL0 pins, between the VCC_RF and VSS_RF pins, and between the AVCC_RF and VSS_RF pins. Place capacitors with values of about 2.2 µF in the case of the VCC_RF pin and about 0.1 µF otherwise as close as possible to every power supply pin, and use the shortest and thickest possible traces for the connections.
Connect the VCL pin to a VSS pin via a 4.7 µF capacitor. The capacitor must be placed close to the pin. For details, refer to section 51.16.1, Connecting VCL Capacitor and Bypass Capacitors.
Do not input signals or an I/O pull-up power supply to ports other than 5-V tolerant ports while the device is not powered.
The current injection that results from input of such a signal or I/O pull-up may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Even if –0.3 to +6.5 V is input to 5-V tolerant ports, it will not cause problems such as damage to the MCU.
Note 1. P16, P17, and PB5 are 5 V tolerant.
Note 2. When the VBATT power supply is selected, P30, P31 are rated from –0.3 V to VBATT + 0.3 V.

Downloaded from Arrow.com.

Table 51.2 Recommended Operating Voltage Conditions

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Power supply voltages	VCC*1, *2, *3	When USB is not used	1.8	—	3.6	V
		When USB is used	3.0	—	3.6	
	VSS		—	0	—	
USB power supply voltages	VCC_USB	When USB regulator is not used	—	VCC	—	V
	VSS_USB		—	0	—	
VBATT power supply voltage	VBATT		1.8	—	3.6	V
Analog power supply voltages	AVCC0*1, *2		1.8	—	3.6	V
	AVSS0		—	0	—	
	VREFH0		1.8	—	AVCC0	
	VREFL0		—	0	—	
BLE power supply voltages	VCC_RF*3		1.8	—	3.6	V
	AVCC_RF*3		1.8	—	3.6	
	VSS_RF		—	0	—	

Note 1. P41 and P47: Set AVCC0 to the same voltage as VCC.
If conditions other than those above are applicable, those listed below apply.
While VCC > 2.4 V: AVCC and VCC can be set independently when AVCC0 ≥ 2.4 V
While VCC ≤ 2.4 V: AVCC and VCC can be set independently when AVCC0 ≥ VCC
Note 2. When powering on the VCC and AVCC0 pins, power them on at the same time or the VCC pin first and then the AVCC0 pin.
Note 3. Set VCC_RF and AVCC_RF to the same voltage as VCC.

Downloaded from Arrow.com.

51.2 DC Characteristics

Table 51.3 DC Characteristics (1) (1/2)

Conditions: 2.7 V ≤ VCC = VCC_USB = VCC_RF = AVCC_RF ≤ 3.6 V, 2.7 V ≤ AVCC0 ≤ 3.6 V,
VSS = AVSS0 = VSS_USB = VSS_RF = 0 V, T_a = –40 to +85°C

Item			Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Schmitt trigger input voltage	RIIC input pin (except for SMBus, 5 V tolerant)		V_{IH}	VCC × 0.7	—	5.8	V	
	P16, P17, PB5 (5 V tolerant)			VCC × 0.8	—	5.8		
	P14, P15, P21, P22, P25 to P27, P35 to P37, PB0, PB1, PB3, PB7, PC0, PC2 to PC7, PD3, PE0 to PE4, PJ3, P30, P31 (other than RTCICn pin), RES#			VCC × 0.8	—	VCC + 0.3		
	P03, P05, P07, P40 to P47			AVCC0 × 0.8	—	AVCC0 + 0.3		
	P30, P31 (RTCICn pin)	When VCC is supplied		VCC × 0.8	—	VCC + 0.3		
		When VBATT is supplied		VBATT × 0.8	—	VBATT + 0.3		
	RIIC input pin (except for SMBus)		V_{IL}	–0.3	—	VCC × 0.3		
	P14 to P17, P21, P22, P25 to P27, P35 to P37, PB0, PB1, PB3, PB5, PB7, PC0, PC2 to PC7, PD3, PE0 to PE4, PJ3, P30, P31 (other than RTCICn pin), RES#			–0.3	—	VCC × 0.2		
	P03, P05, P07, P40 to P47			–0.3	—	AVCC0 × 0.2		
	P30, P31 (RTCICn pin)	When VCC is supplied		–0.3	—	VCC × 0.3		
		When VBATT is supplied		–0.3	—	VBATT × 0.3		
Schmitt trigger input voltage	RIIC input pin (except for SMBus), P16, P17, PB5		ΔV_T	VCC × 0.05	—	—	V	
	P14, P15, P21, P22, P25 to P27, P35 to P37, PB0 , PB1, PB3, PB7, PC0, PC2 to PC7, PD3, PE0 to PE4, PJ3, P30, P31 (other than RTCICn pin), RES#			VCC × 0.1	—	—		
	P03, P05, P07, P40 to P47			AVCC0 × 0.1	—	—		

Downloaded from Arrow.com.

Table 51.3 DC Characteristics (1) (2/2)

Conditions: 2.7 V ≤ VCC = VCC_USB = VCC_RF = AVCC_RF ≤ 3.6 V, 2.7 V ≤ AVCC0 ≤ 3.6 V,
VSS = AVSS0 = VSS_USB = VSS_RF = 0 V, T_a = –40 to +85°C

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Input level voltage (except for Schmitt trigger input pins)	MD	V_{IH}	VCC × 0.9	—	VCC + 0.3	V	
	EXTAL (external clock input)		VCC × 0.8	—	VCC + 0.3		
	RIIC input pin (SMBus)		2.1	—	VCC + 0.3		
	MD	V_{IL}	–0.3	—	VCC × 0.1		
	EXTAL (external clock input)		–0.3	—	VCC × 0.2		
	RIIC input pin (SMBus)		–0.3	—	0.8		

Downloaded from Arrow.com.

Table 51.4 DC Characteristics (2) (1/2)

Conditions: 1.8 V ≤ VCC = VCC_USB = VCC_RF = AVCC_RF ≤ 2.7 V, 1.8 V ≤ AVCC0 < 2.7 V,
VSS = AVSS0 = VSS_USB = VSS_RF = 0 V, T_a = –40 to +85°C

Item			Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Schmitt trigger input voltage	P16, P17, PB5 (5 V tolerant)		V_{IH}	VCC × 0.8	—	5.8	V	
	P14, P15, P21, P22, P25 to P27, P35 to P37, PB0, PB1, PB3, PB7, PC0, PC2 to PC7, PD3, PE0 to PE4, PJ3, P30, P31 (other than RTCICn pin), RES#			VCC × 0.8	—	VCC + 0.3		
	P03, P05, P07, P40 to P47			AVCC0 × 0.8	—	AVCC0 + 0.3		
	P30, P31 (RTCICn pin)	When VCC is supplied		VCC × 0.8	—	VCC + 0.3		
		When VBATT is supplied		VBATT × 0.8	—	VBATT + 0.3		
	P14 to P17, P21, P22, P25 to P27, P35 to P37, PB0 , PB1, PB3, PB5, PB7, PC0, PC2 to PC7, PD3, PE0 to PE4, PJ3, P30, P31 (other than RTCICn pin), RES#		V_{IL}	–0.3	—	VCC × 0.2		
	P03, P05, P07, P40 to P47			–0.3	—	AVCC0 × 0.2		
	P30, P31 (RTCICn pin)	When VCC is supplied		–0.3	—	VCC × 0.2		
		When VBATT is supplied		–0.3	—	VBATT × 0.2		

Downloaded from Arrow.com.

Table 51.4 DC Characteristics (2) (2/2)

Conditions: 1.8 V ≤ VCC = VCC_USB = VCC_RF = AVCC_RF ≤ 2.7 V, 1.8 V ≤ AVCC0 < 2.7 V, VSS = AVSS0 = VSS_USB = VSS_RF = 0 V, T_a = –40 to +85°C

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Schmitt trigger input voltage	P14 to P17, P21, P22, P25 to P27, P35 to P37, PB0, PB1, PB3, PB5, PB7, PC0, PC2 to PC7, PD3, PE0 to PE4, PJ3, P30, P31 (other than RTCICn pin), RES#	ΔV_T	VCC × 0.1	—	—	V	
	P03, P05, P07, P40 to P47		AVCC0 × 0.1	—	—		
Input level voltage (except for Schmitt trigger input pins)	MD	V_{IH}	VCC × 0.9	—	VCC + 0.3	V	
	EXTAL (external clock input)		VCC × 0.8	—	VCC + 0.3		
	MD	V_{IL}	–0.3	—	VCC × 0.1		
	EXTAL (external clock input)		–0.3	—	VCC × 0.2		

Table 51.5 DC Characteristics (3)

Conditions: 1.8 V ≤ VCC = VCC_USB = AVCC0 = VCC_RF = AVCC_RF ≤ 3.6 V, VSS = AVSS0 = VSS_USB = VSS_RF = 0 V, T_a = –40 to +85°C

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Input leakage current	RES#, MD, P35	$\lvert I_{in} \rvert$	—	—	1.0	μA	V_{in} = 0 V, VCC
Three-state leakage current (off-state)	Ports for 5 V tolerant	$\lvert I_{TSI} \rvert$	—	—	1.0	μA	V_{in} = 0 V, 5.8 V
	Ports except for 5 V tolerant		—	—	0.2	μA	V_{in} = 0 V, VCC
Input capacitance	All input pins (except for P35, USB0_DM, USB0_DP)	C_{in}	—	—	15	pF	V_{in} = 0 mV, f = 1 MHz, T_a = 25°C
	P35, USB0_DM, USB0_DP		—	—	30		

Table 51.6 DC Characteristics (4)

Conditions: 1.8 V ≤ VCC = VCC_USB = AVCC0 = VCC_RF = AVCC_RF ≤ 3.6 V, VSS = AVSS0 = VSS_USB = VSS_RF = 0 V, T_a = –40 to +85°C

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Input pull-up resistor	All ports (except for P35)	R_U	10	20	50	kΩ	V_{in} = 0 V

Downloaded from Arrow.com.

Table 51.7 DC Characteristics (5)

Conditions: 1.8 V ≤ VCC = VCC_USB = AVCC0 = VCC_RF = AVCC_RF ≤ 3.6 V, VSS = AVSS0 = VSS_USB = VSS_RF = 0 V, T_a = –40 to +85°C

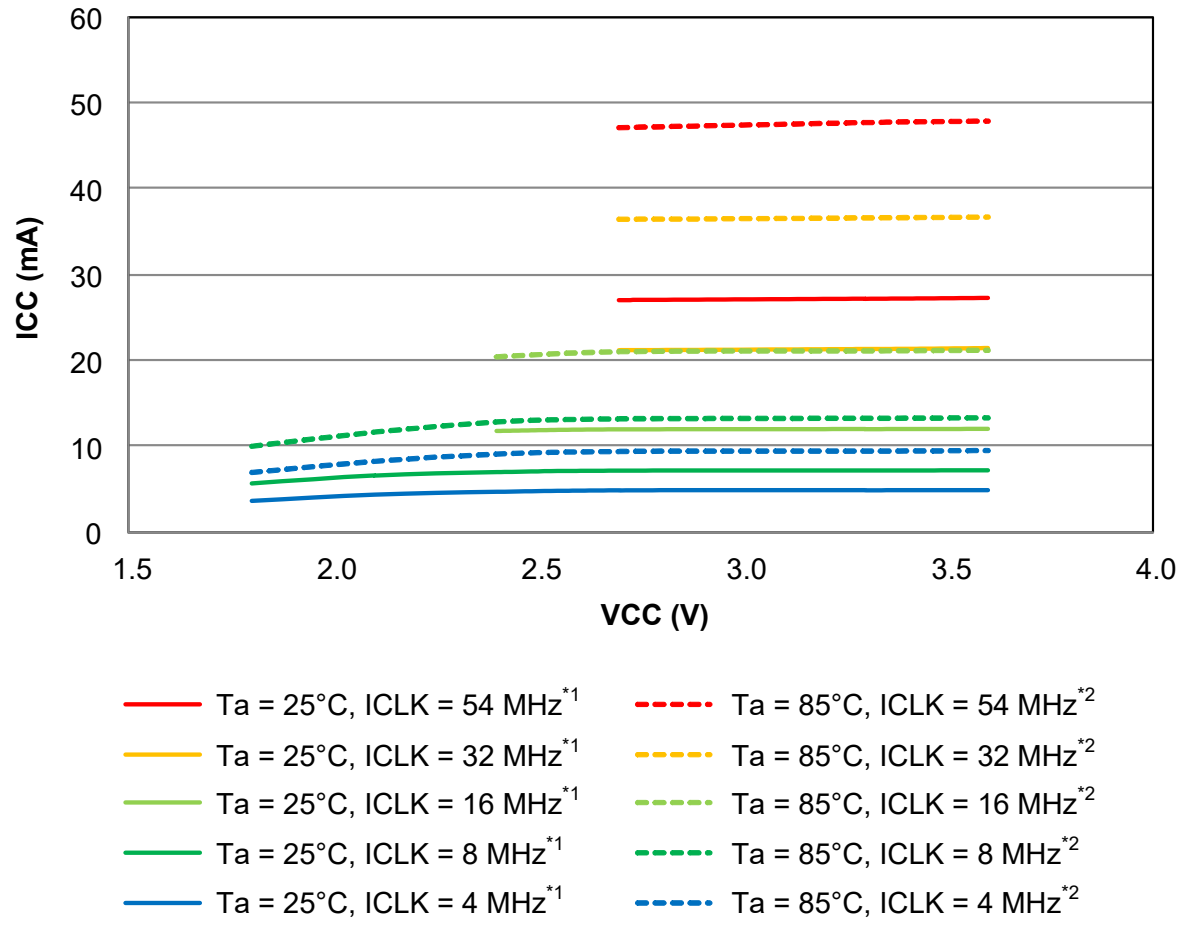
Item					Symbol	Typ. *4	Max.	Unit	Test Conditions
Supply current *1	High-speed operating mode	Normal operating mode	No peripheral operation*2	ICLK = 54 MHz	I_{CC}	6.5	—	mA	
				ICLK = 32 MHz		4.1	—		
				ICLK = 16 MHz		2.9	—		
				ICLK = 8 MHz		2.2	—		
				ICLK = 4 MHz		1.9	—		
			All peripheral operation: Normal	ICLK = 54 MHz*11		26.5	—		
				ICLK = 32 MHz*3		21.0	—		
				ICLK = 16 MHz*3		11.8	—		
				ICLK = 8 MHz*3		6.6	—		
				ICLK = 4 MHz*3		4.2	—		
			All peripheral operation: Max.	ICLK = 54 MHz*11		—	53.3		
				ICLK = 32 MHz*3		—	40.8		
			Increase due to operation of the Trusted Secure IP	PCLKB = 32 MHz		—	2		
		Sleep mode	No peripheral operation*2	ICLK = 54 MHz		3.5	—		
				ICLK = 32 MHz		2.4	—		
				ICLK = 16 MHz		1.9	—		
				ICLK = 8 MHz		1.6	—		
				ICLK = 4 MHz		1.5	—		
			All peripheral operation: Normal	ICLK = 54 MHz*11		13.4	—		
				ICLK = 32 MHz*3		12.5	—		
				ICLK = 16 MHz*3		7.3	—		
				ICLK = 8 MHz*3		4.6	—		
				ICLK = 4 MHz*3		3.3	—		
		Deep sleep mode	No peripheral operation*2	ICLK = 54 MHz		2.3	—		
				ICLK = 32 MHz		1.5	—		
				ICLK = 16 MHz		1.3	—		
				ICLK = 8 MHz		1.2	—		
				ICLK = 4 MHz		1.1	—		
			All peripheral operation: Normal	ICLK = 54 MHz*11		10.6	—		
				ICLK = 32 MHz*3		9.9	—		
				ICLK = 16 MHz*3		5.9	—		
				ICLK = 8 MHz*3		3.8	—		
				ICLK = 4 MHz*3		2.7	—		
		Increase during BGO operation*5				2.5	—		
	Middle-speed operating mode	Normal operating mode	No peripheral operation*6	ICLK = 12 MHz	I_{CC}	2.7	—	mA	
				ICLK = 8 MHz		1.8	—		
				ICLK = 4 MHz		1.4	—		
				ICLK = 1 MHz		1.1	—		
			All peripheral operation: Normal*7	ICLK = 12 MHz		9.6	—		
				ICLK = 8 MHz		6.2	—		
				ICLK = 4 MHz		3.8	—		
				ICLK = 1 MHz		2.3	—		

Downloaded from Arrow.com.

Item					Symbol	Typ. *4	Max.	Unit	Test Conditions
Supply current *1	Middle-speed operating mode	Normal operating mode	All peripheral operation: Max.*7	ICLK = 12 MHz	I_{CC}	—	16.7	mA	
		Sleep mode	No peripheral operation*6	ICLK = 12 MHz		1.9	—		
				ICLK = 8 MHz		1.2	—		
				ICLK = 4 MHz		1.1	—		
				ICLK = 1 MHz		1.0	—		
			All peripheral operation: Normal*7	ICLK = 12 MHz		6.1	—		
				ICLK = 8 MHz		4.4	—		
				ICLK = 4 MHz		3.0	—		
				ICLK = 1 MHz		2.0	—		
		Deep sleep mode	No peripheral operation*6	ICLK = 12 MHz		1.6	—		
				ICLK = 8 MHz		1.0	—		
				ICLK = 4 MHz		0.9	—		
				ICLK = 1 MHz		0.8	—		
			All peripheral operation: Normal*7	ICLK = 12 MHz		5.1	—		
				ICLK = 8 MHz		3.7	—		
				ICLK = 4 MHz		2.6	—		
				ICLK = 1 MHz		1.8	—		
		Increase during BGO operation*5				2.5	—		
	Low-speed operating mode	Normal operating mode	No peripheral operation*8	ICLK = 32 kHz	I_{CC}	5.2	—	μA	
			All peripheral operation: Normal *9, *10	ICLK = 32 kHz		22.3	—		
			All peripheral operation: Max.*9, *10	ICLK = 32 kHz		—	74.4		
		Sleep mode	No peripheral operation*8	ICLK = 32 kHz		3.0	—		
			All peripheral operation: Normal*9	ICLK = 32 kHz		13.1	—		
		Deep sleep mode	No peripheral operation*8	ICLK = 32 kHz		2.4	—		
			All peripheral operation: Normal*9	ICLK = 32 kHz		10.5	—		

Note 1. Supply current values do not include the output charge/discharge current from all pins. The values apply when internal pull-up MOSs are in the off state.
Note 2. Clock supply to the peripheral functions is stopped. This does not include BGO operation. The clock source is PLL. FCLK, and PCLK are set to divided by 64.
Note 3. Clocks are supplied to the peripheral functions. This does not include BGO operation. The clock source is PLL. FCLK, and PCLK are the same frequency as that of ICLK.
Note 4. Values when VCC is 3.3 V.
Note 5. This is the increase when data is programmed to or erased from the ROM or E2 DataFlash during program execution.
Note 6. Clock supply to the peripheral functions is stopped. The clock source is PLL when ICLK is 12 MHz and HOCO for other cases. FCLK, and PCLK are set to divided by 64.
Note 7. Clocks are supplied to the peripheral functions. The clock source is PLL when ICLK is 12 MHz and HOCO for other cases. FCLK, and PCLK are the same frequency of that of the ICLK.
Note 8. Clock supply to the peripheral functions is stopped. The clock source is the sub oscillation circuit. FCLK, and PCLK are set to divided by 64.
Note 9. Clocks are supplied to the peripheral functions. The clock source is the sub oscillation circuit. FCLK, and PCLK are the same frequency as that of ICLK.
Note 10. This is the value when the MSTPCRA.MSTPA17 (12-bit A/D converter module stop bit) is in the module stop state.
Note 11. Clocks are supplied to the peripheral functions. This does not include BGO operation. The clock source is PLL. FCLK, and PCLKB are set to divided by 2 and PCLKA and PCLKD are the same frequency as that of ICLK.

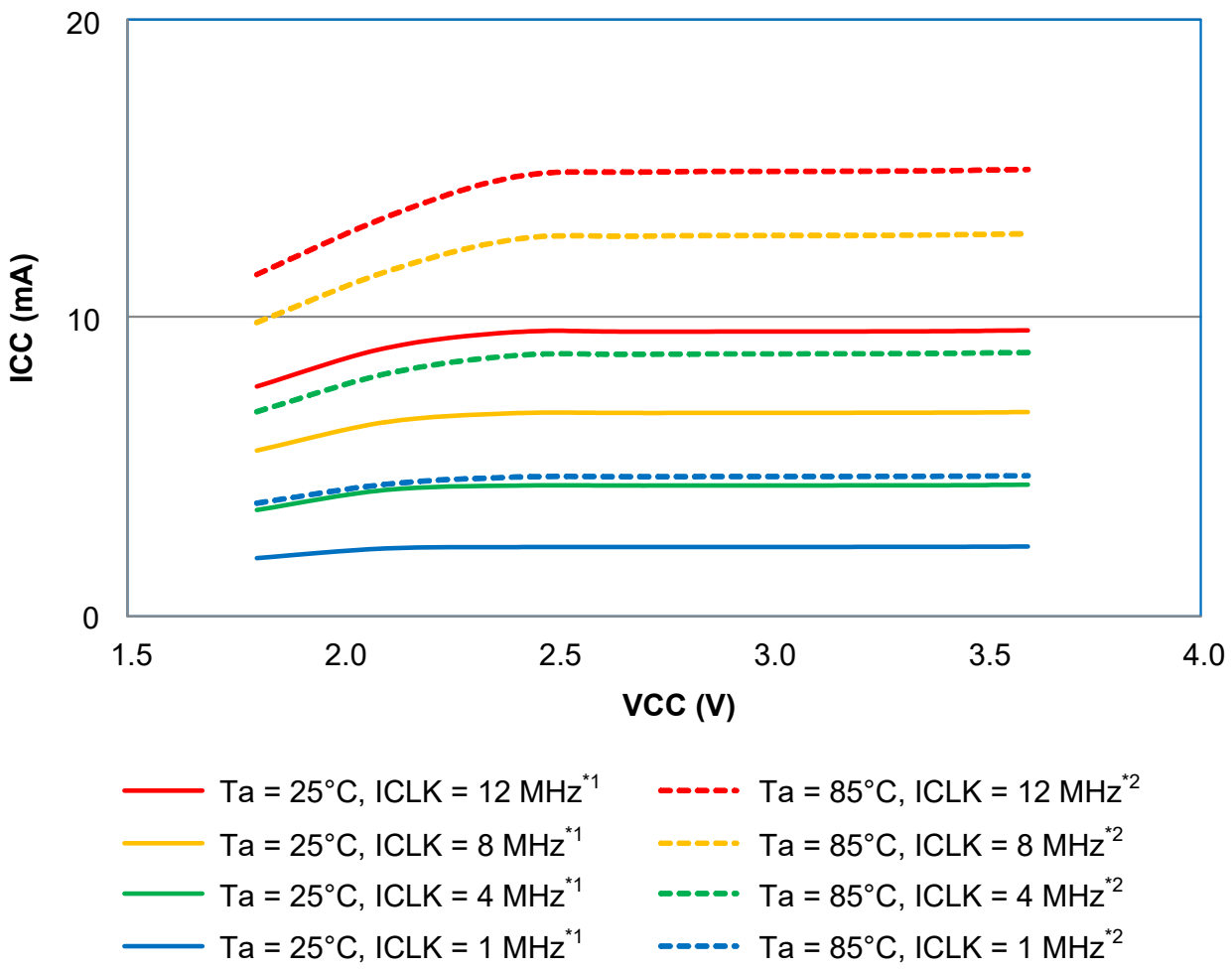
Downloaded from Arrow.com.



Note 1. All peripheral operations except any BGO operation are operating normally. Indicates the average of the typical samples through actual measurement during product evaluation.
Note 2. All peripheral operations except any BGO operation are operating at maximum. Indicates the average of the upper-limit samples through actual measurement during product evaluation.

Figure 51.1 Voltage Dependency in High-Speed Operating Mode (Reference Data)

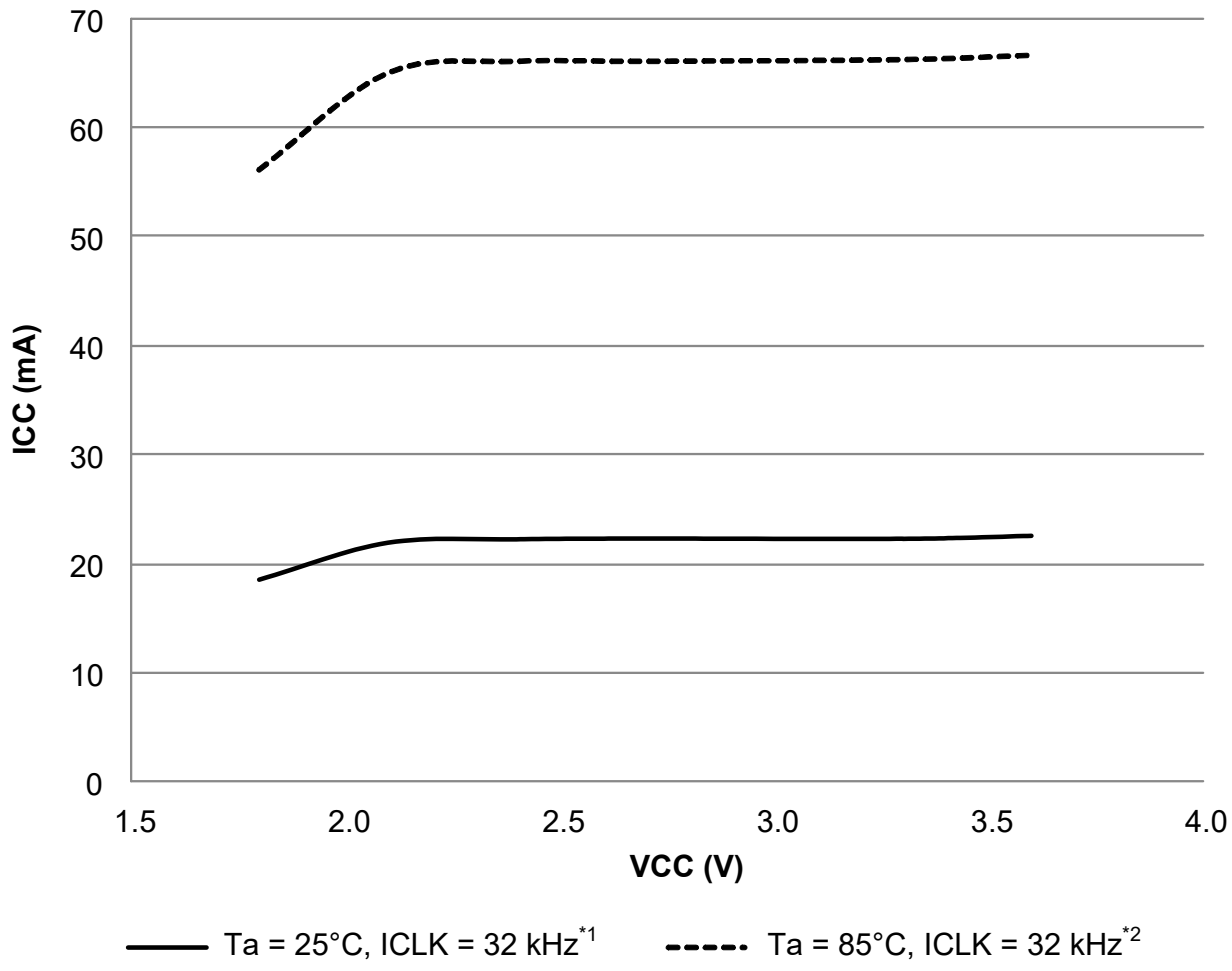
Downloaded from Arrow.com.



Note 1. All peripheral operations except any BGO operation are operating normally. Indicates the average of the typical samples through actual measurement during product evaluation.
Note 2. All peripheral operations except any BGO operation are operating at maximum. Indicates the average of the upper-limit samples through actual measurement during product evaluation.

Figure 51.2 Voltage Dependency in Middle-Speed Operating Mode (Reference Data)

Downloaded from Arrow.com.



Note 1. All peripheral operations except any BGO operation are operating normally. Indicates the average of the typical samples through actual measurement during product evaluation.
Note 2. All peripheral operations except any BGO operation are operating at maximum. Indicates the average of the upper-limit samples through actual measurement during product evaluation.

Figure 51.3 Voltage Dependency in Low-Speed Operating Mode (Reference Data)

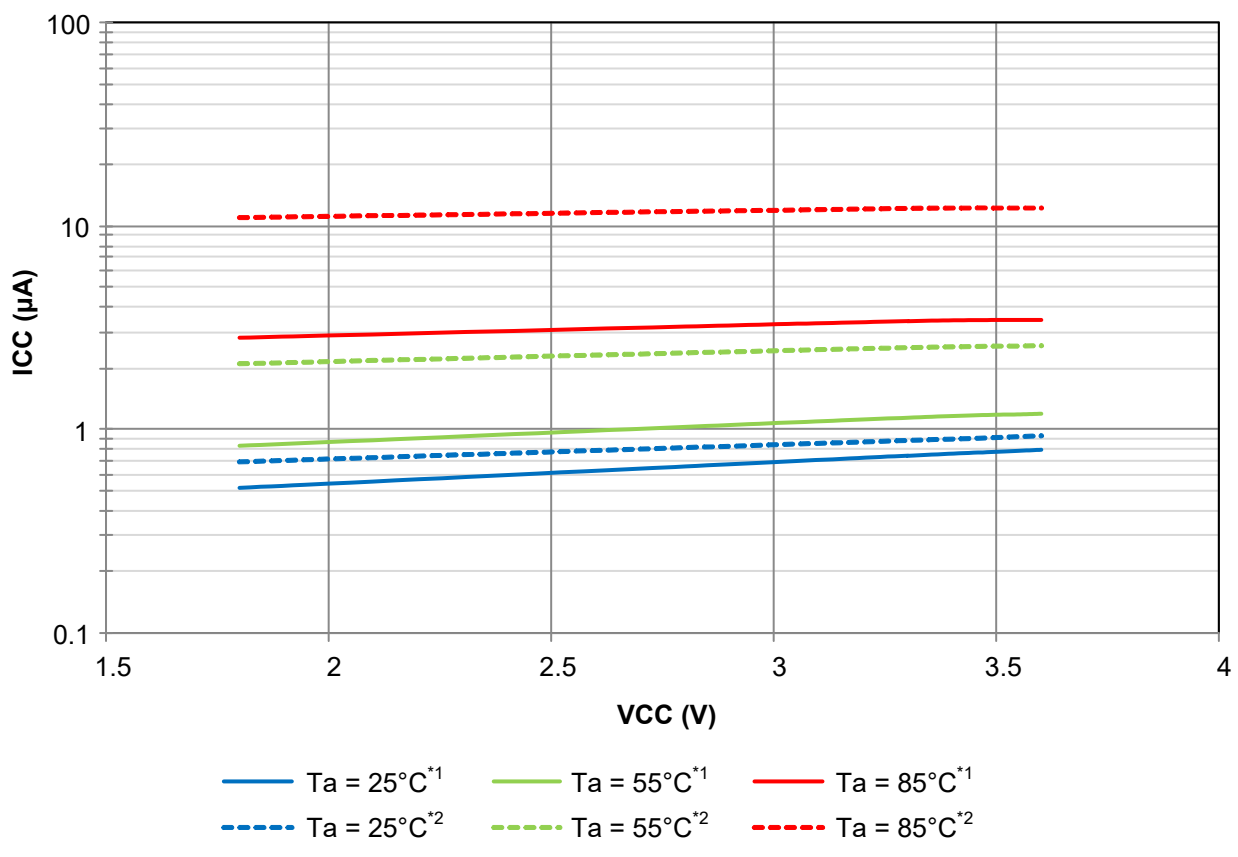
Downloaded from Arrow.com.

Table 51.8 DC Characteristics (6)

Conditions: 1.8 V ≤ VCC = VCC_USB = AVCC0 = VCC_RF = AVCC_RF ≤ 3.6 V, VSS = AVSS0 = VSS_USB = VSS_RF = 0 V, T_a = –40 to +85°C

Item			Symbol	Typ.*3	Max.	Unit	Test Conditions
Supply current*1	Software standby mode*2	T_a = 25°C	I_{CC}	0.8	3.7	μA	
		T_a = 55°C		1.2	4.3		
		T_a = 85°C		3.5	18.6		
	Increment for IWDT operation			0.4	—		
	Increment for LPT operation			0.4	—		Use IWDT-Dedicated On-Chip Oscillator for clock source
	Increment for RTC operation*4			0.4	—		RCR3.RTCDV[2:0] set to low drive capacity
				1.2	—		RCR3.RTCDV[2:0] set to normal drive capacity

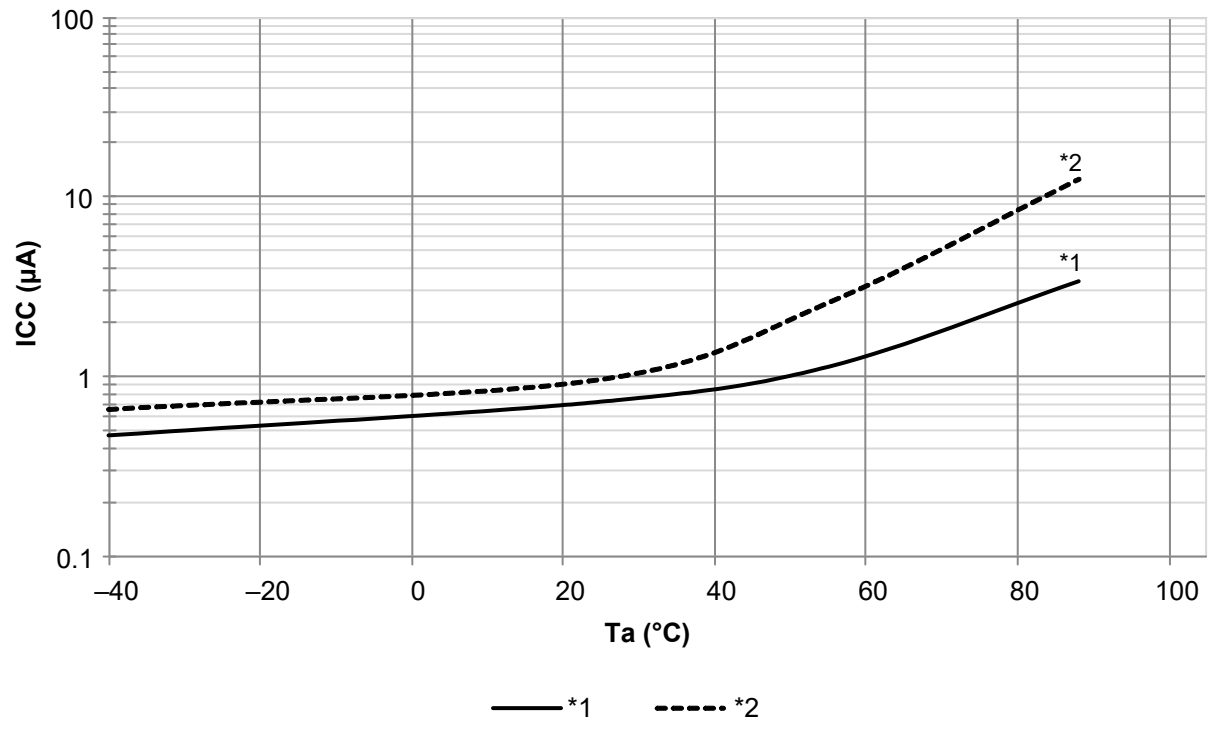
Note 1. Supply current values are with all output pins unloaded and all input pull-up MOSs in the off state.
Note 2. The IWDT, LVD, and CMPB are stopped.
Note 3. When VCC is 3.3 V.
Note 4. This increment includes the oscillation circuit.



Note 1. Indicates the average of the typical samples through actual measurement during product evaluation.
Note 2. Indicates the average of the upper-limit samples through actual measurement during product evaluation.

Figure 51.4 Voltage Dependency in Software Standby Mode (Reference Data)

Downloaded from Arrow.com.



Note 1. Average value of the tested middle samples during product evaluation.
Note 2. Average value of the tested upper-limit samples during product evaluation.

Figure 51.5 Temperature Dependency in Software Standby Mode (Reference Data)

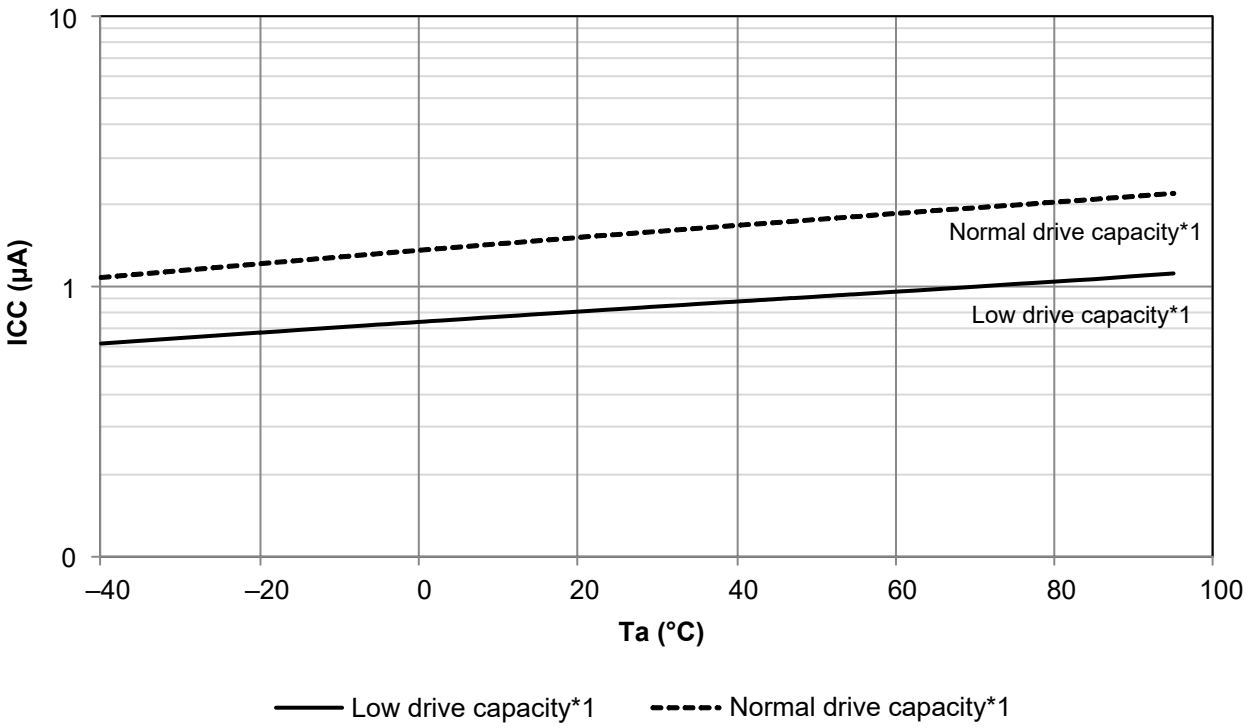
Table 51.9 DC Characteristics (7)

Conditions: 1.8 V ≤ VCC = VCC_USB = AVCC0 = VCC_RF = AVCC_RF ≤ 3.6 V, VSS = AVSS0 = VSS_USB = VSS_RF = 0 V, T_a = –40 to +85°C

Item			Symbol	Typ.	Max.	Unit	Test Conditions
Supply current*1	RTC operation when VCC is off	T_a = 25°C	I_{CC}	0.8	—	µA	VBATT = 2.0 V RCR3.RTCDV[2:0] set to low drive capacity
		T_a = 55°C		0.9	—		
		T_a = 85°C		1.0	—		
		T_a = 25°C		0.9	—		VBATT = 3.3 V RCR3.RTCDV[2:0] set to low drive capacity
		T_a = 55°C		1.0	—		
		T_a = 85°C		1.1	—		
		T_a = 25°C		1.5	—		VBATT = 2.0 V RCR3.RTCDV[2:0] set to normal drive capacity
		T_a = 55°C		1.8	—		
		T_a = 85°C		2.1	—		
		T_a = 25°C		1.6	—		VBATT = 3.3 V RCR3.RTCDV[2:0] set to normal drive capacity
		T_a = 55°C		1.9	—		
		T_a = 85°C		2.2	—		

Note 1. Supply current values do not include output charge/discharge current from all pins. The values apply when internal pull-up MOSs are in the off state.

Downloaded from Arrow.com.



Note 1. Indicates the average of the typical samples through actual measurement during product evaluation.

Figure 51.6 Temperature Dependency of RTC Operation with VCC Off (Reference Data)

Table 51.10 DC Characteristics (8)

Conditions: 1.8 V ≤ VCC = VCC_USB = AVCC0 = VCC_RF = AVCC_RF ≤ 3.6 V, VSS = AVSS0 = VSS_USB = VSS_RF = 0 V

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Permissible total power consumption*1	Pd	—	—	350	mW	D-version product

Note 1. Total power dissipated by the entire chip (including output currents)

Downloaded from Arrow.com.

Table 51.11 DC Characteristics (9)

Conditions: 1.8 V ≤ VCC = VCC_USB = AVCC0 = VCC_RF = AVCC_RF ≤ 3.6 V, VSS = AVSS0 = VSS_USB = VSS_RF = 0 V, T_a = –40 to +85°C

Item		Symbol	Min.	Typ.*7	Max.	Unit	Test Conditions
Analog power supply current	During A/D conversion (at high-speed conversion)	I_{AVCC}	—	0.7	1.7	mA	
	During A/D conversion (in low-current mode)		—	0.6	1.0		
	During D/A conversion (per channel)*1		—	0.4	0.8		
	Waiting for A/D and D/A conversion (all units)		—	—	0.4	µA	
Reference power supply current	During A/D conversion (at high-speed conversion)	I_{REFH0}	—	25	150	µA	
	Waiting for A/D conversion (all units)		—	—	60	nA	
	During D/A conversion (per channel)	I_{REFH}	—	50	100	µA	
	Waiting for D/A conversion (all units)		—	—	100	nA	
LVD1	—	I_{LVD}	—	0.15	—	µA	
Temperature sensor*6	—	I_{TEMP}	—	75	—	µA	
Comparator B operating current*6	Window mode	I_{CMP}*5	—	12.5	28.6	µA	
	Comparator high-speed mode (per channel)		—	3.2	16.2	µA	
	Comparator low-speed mode (per channel)		—	1.7	4.4	µA	
CTSU operating current	• When sleep mode Base clock frequency: 2MHz Pin capacitance: 50pF	I_{CTSU}	—	150	—	µA	
USB operating current*4	During USB communication operation under the following settings and conditions • Host controller operation is set to full-speed mode Bulk OUT transfer (64 bytes) × 1, bulk IN transfer (64 bytes) × 1 • Connect peripheral devices via a 1-meter USB cable from the USB port.	I_{USBH}*2	—	4.3 (VCC) 0.9 (VCC_USB)	—	mA	
	During USB communication operation under the following settings and conditions • Function controller operation is set to full-speed mode Bulk OUT transfer (64 bytes) × 1, bulk IN transfer (64 bytes) × 1 • Connect the host device via a 1-meter USB cable from the USB port.	I_{USBF}*2	—	3.6 (VCC) 1.1 (VCC_USB)	—	mA	
	During suspended state under the following setting and conditions • Function controller operation is set to full-speed mode (pull up the USB0_DP pin) • Software standby mode • Connect the host device via a 1-meter USB cable from the USB port.	I_{SUSP}*3	—	0.35 (VCC) 170 (VCC_USB)	—	µA	

Note 1. The value of the D/A converter is the value of the power supply current including the reference current.
Note 2. Current consumed only by the USB module.
Note 3. Includes the current supplied from the pull-up resistor of the USB0_DP pin to the pull-down resistor of the host device, in addition to the current consumed by this MCU during the suspended state.
Note 4. Current consumed by the power supplies (VCC and VCC_USB).
Note 5. Current consumed only by the comparator B module.
Note 6. Current consumed by the power supply (VCC).
Note 7. When VCC = AVCC0 = VCC_USB = 3.3 V.

Downloaded from Arrow.com.

Table 51.12 DC Characteristics (10)

Conditions: VCC = VCC_USB = AVCC0 = VCC_RF = AVCC_RF = 3.3 V, VSS = AVSS0 = VSS_USB = VSS_RF = 0 V,
T_a = +25°C

Item	Symbol	Typ. Transmit output power 0 dBm	Typ. Transmit output power 4 dBm	Unit	Test Conditions
BLE operating current (when the DC-to-DC converter is selected)	Idd_tx	4.3	8.7	mA	Transmit mode, 2Mbps
					Transmit mode, 1Mbps
		4.5	8.7		Transmit mode, 500kbps
					Transmit mode, 125kbps
	Idd_rx	3.0	3.5	mA	Receive mode, 2Mbps Prf = –67dBm
		3.0	3.4		Receive mode, 1Mbps Prf = –67dBm
		3.2	3.5		Receive mode, 500kbps Prf = –72dBm
		3.3	3.5		Receive mode, 125kbps Prf = –79dBm
	Idd_idle	0.5		mA	Idle mode
	Idd_slp	1.5		µA	Deep sleep mode
	Idd_down	0.1		µA	Power down mode
BLE operating current (when the linear regulator is selected)	Idd_tx	10.2	18.1	mA	Transmit mode, 2Mbps
					Transmit mode, 1Mbps
					Transmit mode, 500kbps
					Transmit mode, 125kbps
	Idd_rx	6.9		mA	Receive mode, 2Mbps Prf = –67dBm
		6.9			Receive mode, 1Mbps Prf = –67dBm
		6.9			Receive mode, 500kbps Prf = –72dBm
		7.1			Receive mode, 125kbps Prf = –79dBm
	Idd_idle	0.7		mA	Idle mode
	Idd_slp	1.5		µA	Deep sleep mode
	Idd_down	0.1		µA	Power down mode

Table 51.13 DC Characteristics (11)

Conditions: 1.8 V ≤ VCC = VCC_USB = AVCC0 = VCC_RF = AVCC_RF ≤ 3.6 V, VSS = AVSS0 = VSS_USB = VSS_RF = 0 V,
T_a = –40 to +85°C

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
RAM standby voltage	V_{RAM}	1.8	—	—	V	

Downloaded from Arrow.com.

Table 51.14 DC Characteristics (12)

Conditions: 0 V ≤ VCC = VCC_USB = AVCC0 = VCC_RF = AVCC_RF ≤ 3.6 V, VSS = AVSS0 = VSS_USB = VSS_RF = 0 V, T_a = –40 to +85°C

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Power-on VCC rising gradient	At normal startup*1	SrVCC	0.02	—	20	ms/V	
	During fast startup time*2		0.02	—	2		
	Voltage monitoring 0 reset enabled at startup*3, *4		0.02	—	—		

Note 1. When OFS1.(FASTSTUP, LVDAS) bits are 11b.
Note 2. When OFS1.(FASTSTUP, LVDAS) bits are 01b.
Note 3. When OFS1.LVDAS bit is 0.
Note 4. Turn on the power supply voltage according to the normal startup rising gradient because the settings in the OFS1 register are not read in boot mode.

Table 51.15 DC Characteristics (13)

Conditions: 1.8 V ≤ VCC = VCC_USB = AVCC0 = VCC_RF = AVCC_RF ≤ 3.6 V, VSS = AVSS0 = VSS_USB = VSS_RF = 0 V, T_a = –40 to +85°C

The ripple voltage must meet the allowable ripple frequency $f_{r\,(VCC)}$ within the range between the VCC upper limit and lower limit. When VCC change exceeds VCC ±10%, the allowable voltage change rising/falling gradient dt/dVCC must be met.

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Allowable ripple frequency	$f_{r\,(VCC)}$	—	—	10	kHz	Figure 51.7 $V_{r\,(VCC)}$ ≤ VCC × 0.2
		—	—	1	MHz	Figure 51.7 $V_{r\,(VCC)}$ ≤ VCC × 0.08
		—	—	10	MHz	Figure 51.7 $V_{r\,(VCC)}$ ≤ VCC × 0.06
Allowable voltage change rising/falling gradient	dt/dVCC	1.0	—	—	ms/V	When VCC change exceeds VCC ±10%

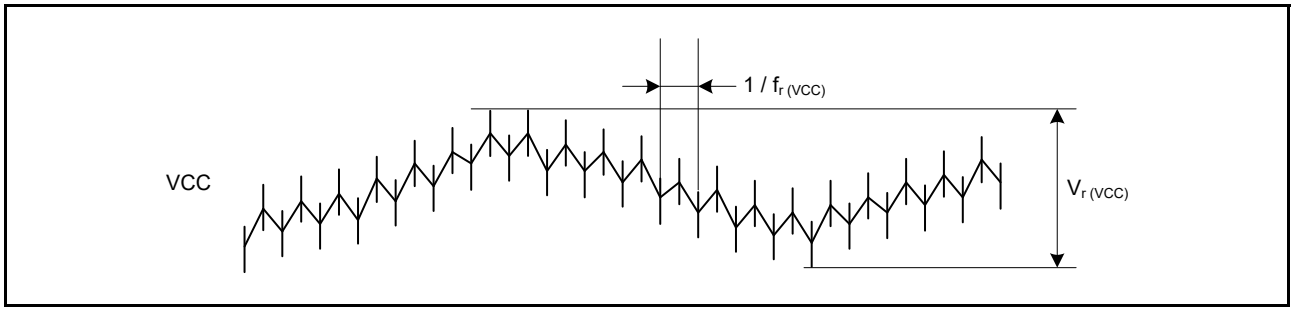


Figure 51.7 Ripple Waveform

Table 51.16 DC Characteristics (14)

Conditions: 1.8 V ≤ VCC = VCC_USB = AVCC0 = VCC_RF = AVCC_RF ≤ 3.6 V, VSS = AVSS0 = VSS_USB = VSS_RF = 0 V, T_a = –40 to +85°C

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Permissible error of VCL pin external capacitance	C_{VCL}	1.4	4.7	7.0	µF	

Note: The recommended capacitance is 4.7 µF. Variations in connected capacitors should be within the above range.

Downloaded from Arrow.com.

Table 51.17 Permissible Output Currents

Conditions: 1.8 V ≤ VCC = VCC_USB = AVCC0 = VCC_RF = AVCC_RF ≤ 3.6 V, VSS = AVSS0 = VSS_USB = VSS_RF = 0 V, T_a = –40 to +85°C

Item			Symbol	Max.	Unit
Permissible output low current (average value per pin)	P03, P05, P07, P36, P37, P40 to P47		I_{OL}	4.0	mA
	Ports other than above	Normal output mode		4.0	
		High-drive output mode		8.0	
Permissible output low current (maximum value per pin)	P03, P05, P07, P36, P37, P40 to P47			4.0	
	Ports other than above	Normal output mode		4.0	
		High-drive output mode		8.0	
Permissible output low current	Total of P03, P05, P07, P40 to P47		ΣI_{OL}	40	
	Total of P14 to P17, P21, P22, P25 to P27, P30, P31, P35 to P37, PJ3			40	
	Total of PB0, PB1, PB3, PB5, PB7, PC0, PC2 to PC7			40	
	Total of PD3, PE0 to PE4			40	
	Total of all output pins			80	
Permissible output high current (average value per pin)	P03, P05, P07, P36, P37, P40 to P47		I_{OH}	–4.0	
	Ports other than above	Normal output mode		–4.0	
		High-drive output mode		–8.0	
Permissible output high current (maximum value per pin)	P03, P05, P07, P36, P37, P40 to P47			–4.0	
	Ports other than above	Normal output mode		–4.0	
		High-drive output mode		–8.0	
Permissible output high current	Total of P03, P05, P07, P40 to P47		ΣI_{OH}	–40	
	Total of P14 to P17, P21, P22, P25 to P27, P30, P31, P35 to P37, PJ3			–40	
	Total of PB0, PB1, PB3, PB5, PB7, PC0, PC2 to PC7			–40	
	Total of PD3, PE0 to PE4			–40	
	Total of all output pins			–80	

Note: Do not exceed the permissible total supply current.

Downloaded from Arrow.com.

Table 51.18 Output Values of Voltage (1)

Conditions: 1.8 V ≤ VCC = VCC_USB = AVCC0 = VCC_RF = AVCC_RF ≤ 2.7 V, VSS = AVSS0 = VSS_USB = VSS_RF = 0 V, T_a = –40 to +85°C

Item				Symbol	Min.	Max.	Unit	Test Conditions
Output low	All output ports*1	Normal output mode		V_{OL}	—	0.8	V	I_{OL} = 0.5 mA
		High-drive output mode			—	0.8		I_{OL} = 1.0 mA
Output high	All output ports*1	Normal output mode	P03, P05, P07, P40 to P47	V_{OH}	AVCC0 – 0.5	—	V	I_{OH} = –0.5 mA
			Ports other than above		VCC – 0.5	—		
		High-drive output mode			VCC – 0.5	—		I_{OH} = –1.0 mA

Note 1. This excludes the CLKOUT_RF pin.

Table 51.19 Output Values of Voltage (2)

Conditions: 2.7 V ≤ VCC = VCC_USB = AVCC0 = VCC_RF = AVCC_RF ≤ 3.6 V, VSS = AVSS0 = VSS_USB = VSS_RF = 0 V, T_a = –40 to +85°C

Item				Symbol	Min.	Max.	Unit	Test Conditions
Output low	All output ports (except for RIIC) *1	Normal output mode		V_{OL}	—	0.8	V	I_{OL} = 1.0 mA
		High-drive output mode			—	0.8		I_{OL} = 2.0 mA
	RIIC pins	Standard mode (Normal output mode)			—	0.4		I_{OL} = 3.0 mA
		Fast mode (High-drive output mode)			—	0.6		I_{OL} = 6.0 mA
Output high	All output ports*1	Normal output mode	P03, P05, P07, P40 to P47	V_{OH}	AVCC0 – 0.8	—	V	I_{OH} = –1.0 mA
			Ports other than above		VCC – 0.8	—		
		High-drive output mode			VCC – 0.8	—		I_{OH} = –2.0 mA

Note 1. This excludes the CLKOUT_RF pin.

Table 51.20 Output Values of Voltage (3)

Conditions: 3.0V ≤ VCC = VCC_USB = AVCC0 = VCC_RF = AVCC_RF ≤ 3.6V, VSS = AVSS0 = VSS_USB = VSS_RF = 0V, Ta = –40 to +85°C

Item		Symbol	Min.	Max.	Unit	Test Conditions
Output low	CLKOUT_RF	V_{OL}	—	0.3	V	I_{OL} = 0.5 mA
Output high	CLKOUT_RF	V_{OH}	VCC_RF – 0.3	—	V	I_{OH} = –0.5 mA

Downloaded from Arrow.com.

51.2.1 Normal I/O Pin Output Characteristics (1)

Figure 51.8 to Figure 51.11 show the characteristics when normal output is selected by the drive capacity control register.

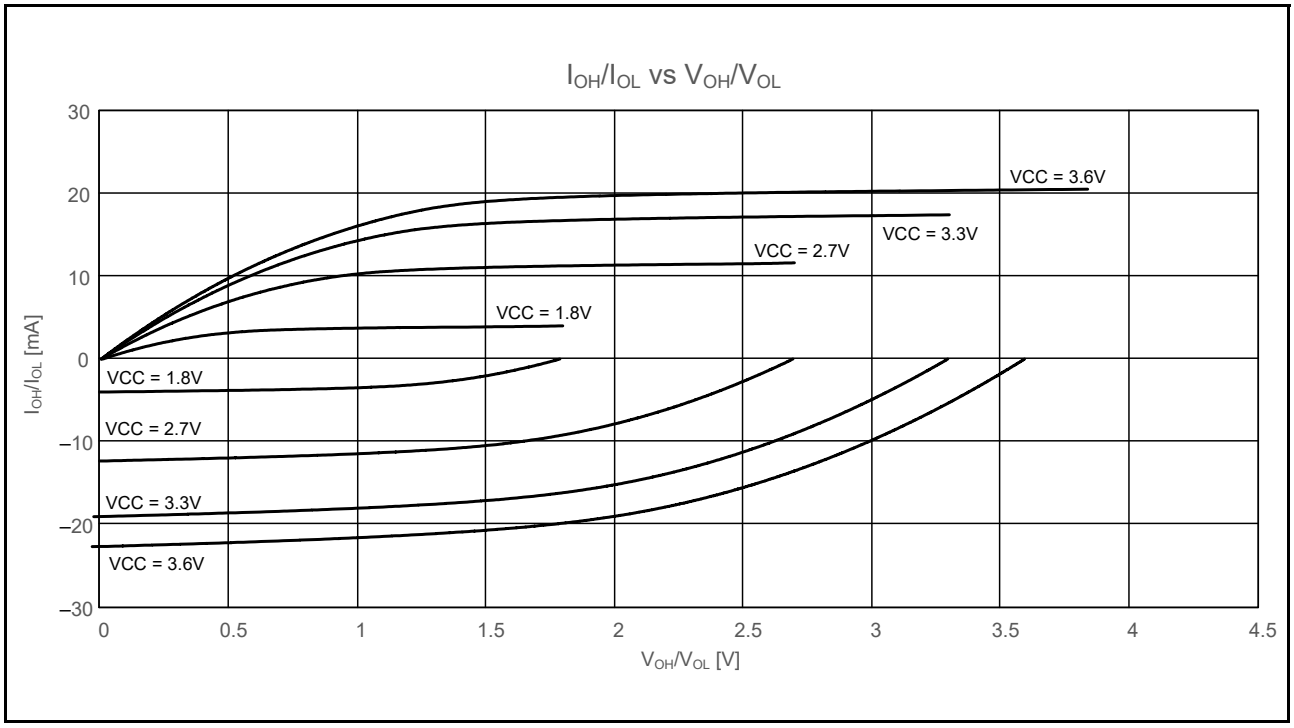


Figure 51.8 V_{OH}/V_{OL} and I_{OH}/I_{OL} Voltage Characteristics at T_a = 25°C When Normal Output is Selected (Reference Data)

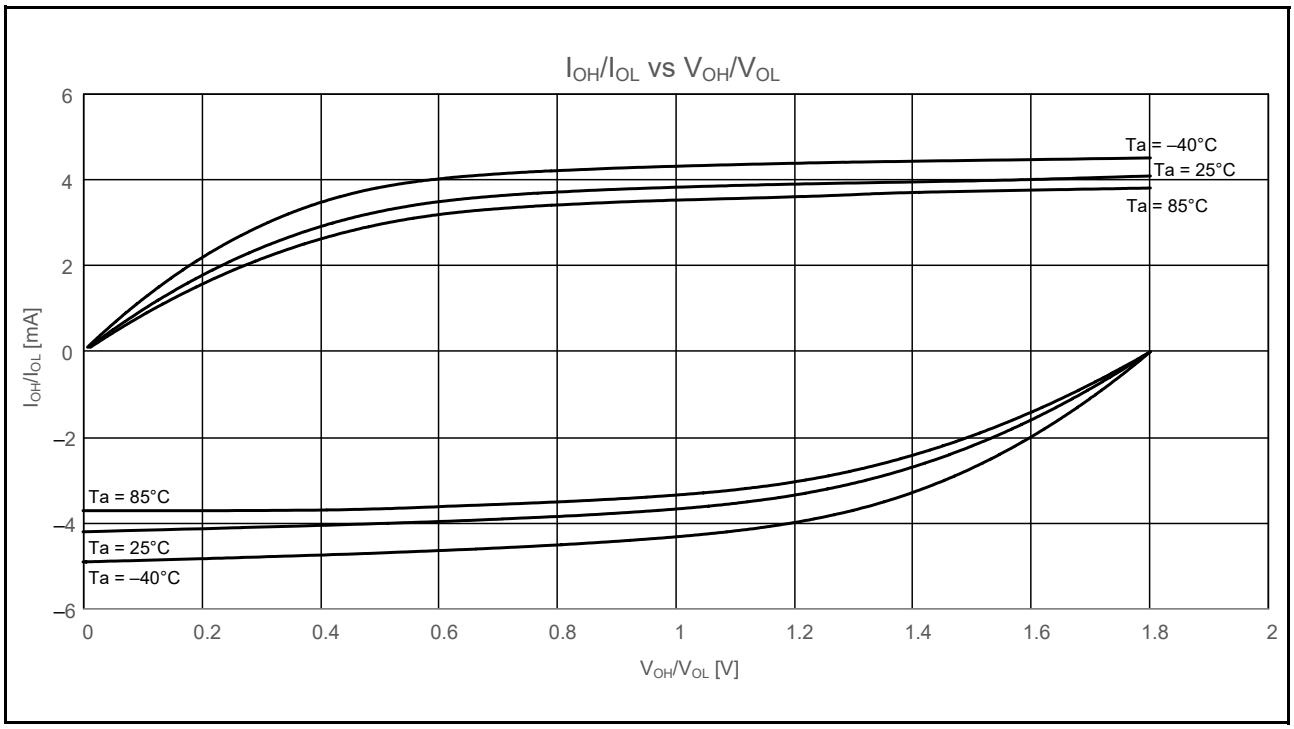


Figure 51.9 V_{OH}/V_{OL} and I_{OH}/I_{OL} Temperature Characteristics at VCC = 1.8 V When Normal Output is Selected (Reference Data)

Downloaded from Arrow.com.

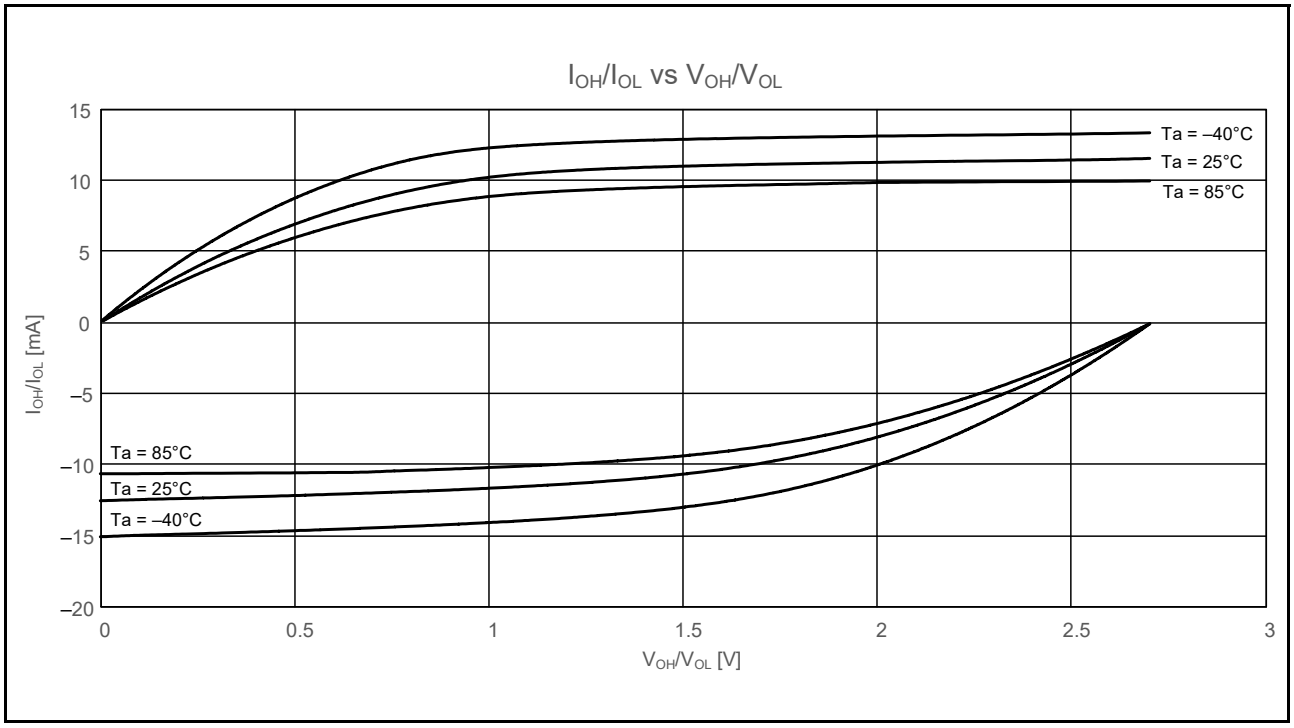


Figure 51.10 V_{OH}/V_{OL} and I_{OH}/I_{OL} Temperature Characteristics at VCC = 2.7 V When Normal Output is Selected (Reference Data)

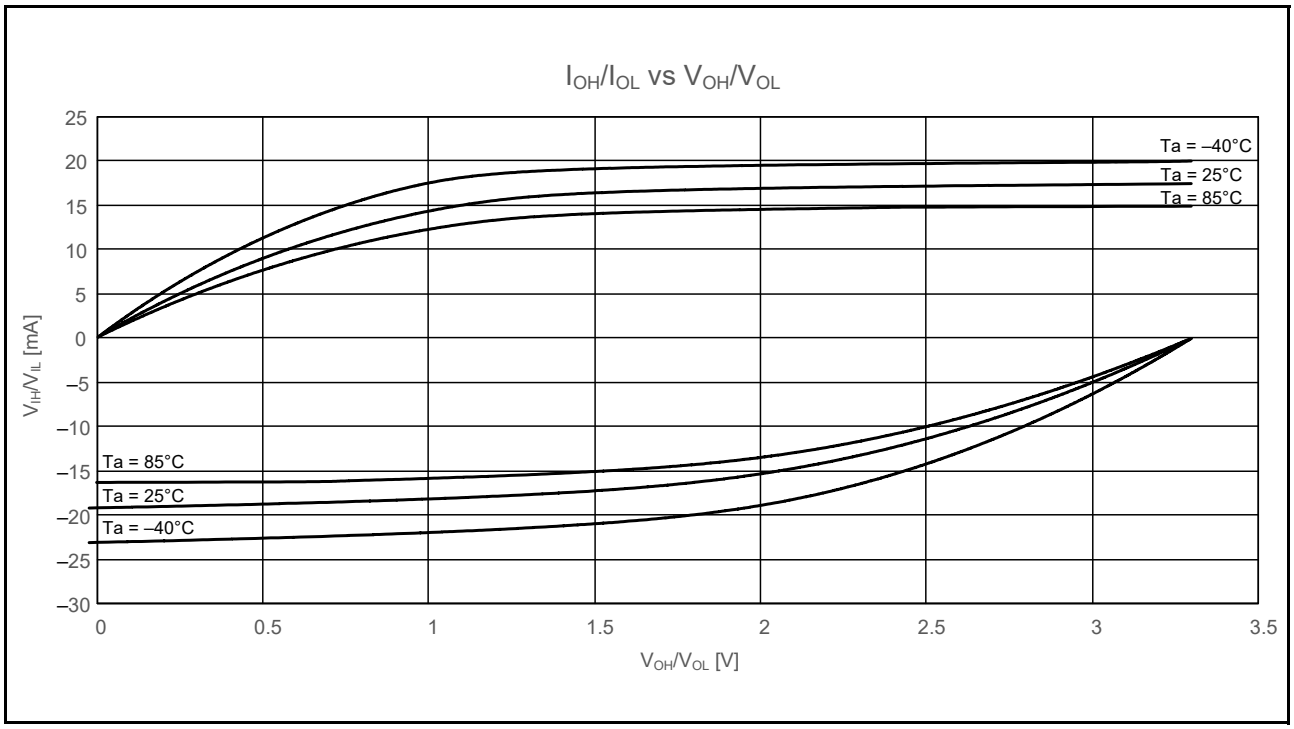


Figure 51.11 V_{OH}/V_{OL} and I_{OH}/I_{OL} Temperature Characteristics at VCC = 3.3 V When Normal Output is Selected (Reference Data)

Downloaded from Arrow.com.

51.2.2 Normal I/O Pin Output Characteristics (2)

Figure 51.12 to Figure 51.15 show the characteristics when high-drive output is selected by the drive capacity control register.

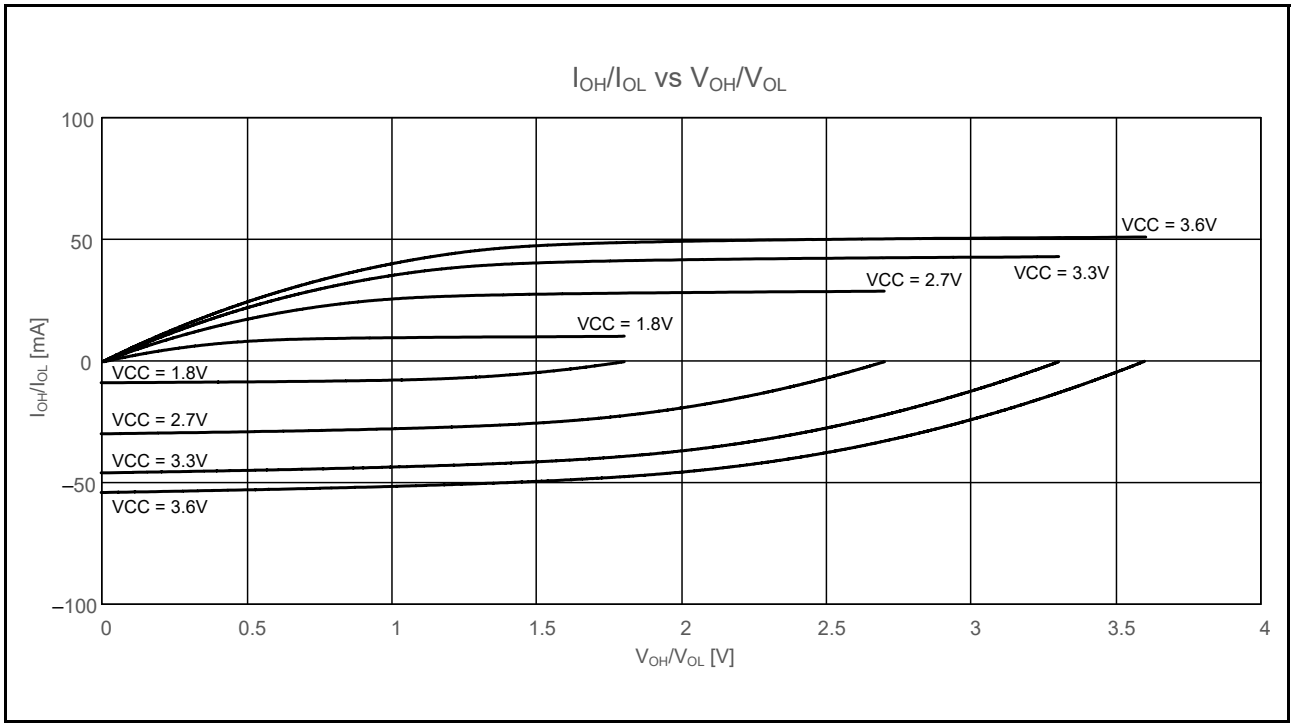


Figure 51.12 V_{OH}/V_{OL} and I_{OH}/I_{OL} Voltage Characteristics at T_a = 25°C When High-Drive Output is Selected (Reference Data)

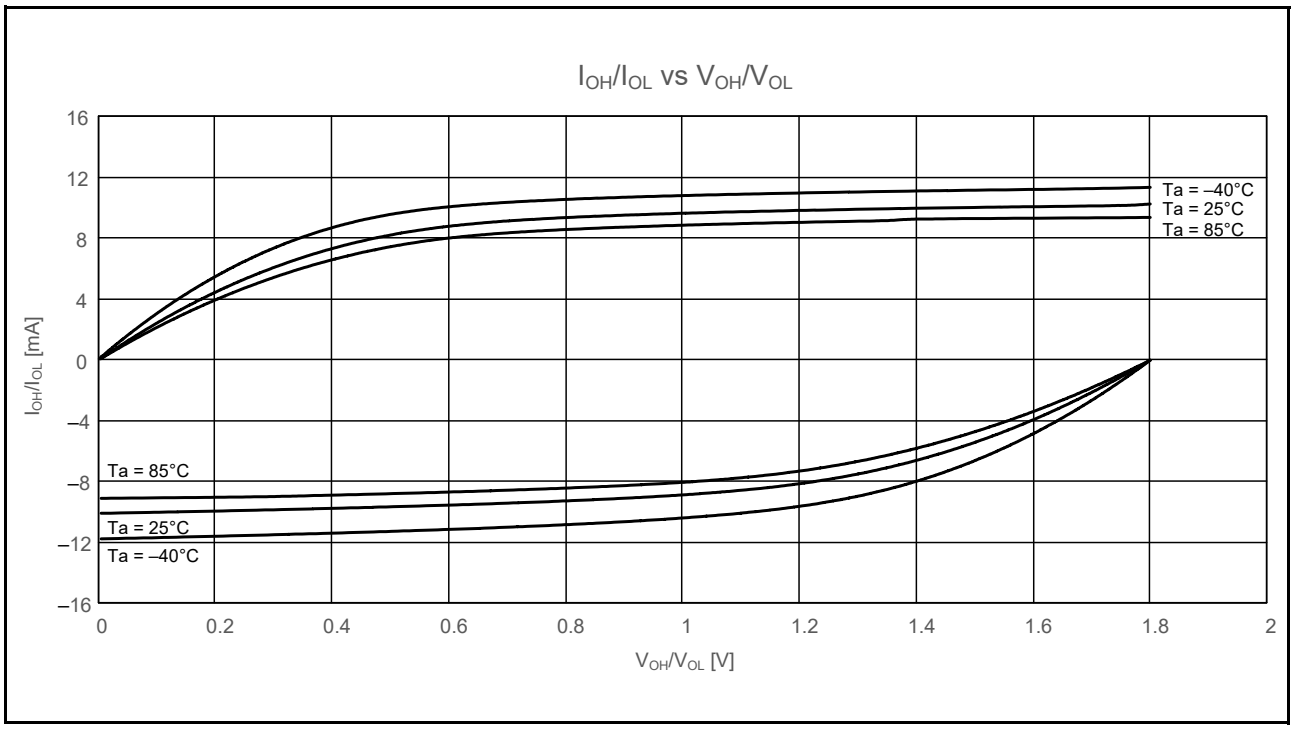


Figure 51.13 V_{OH}/V_{OL} and I_{OH}/I_{OL} Temperature Characteristics at VCC = 1.8 V When High-Drive Output is Selected (Reference Data)

Downloaded from Arrow.com.

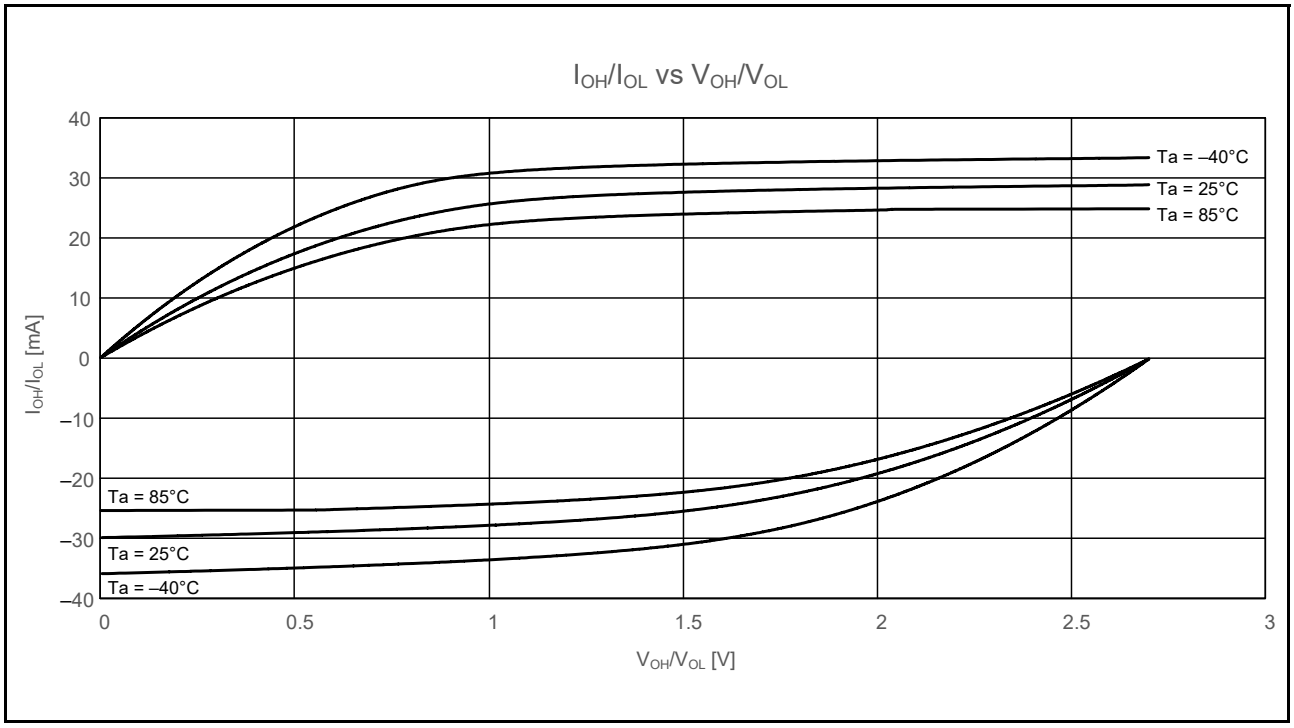


Figure 51.14 V_{OH}/V_{OL} and I_{OH}/I_{OL} Temperature Characteristics at VCC = 2.7 V When High-Drive Output is Selected (Reference Data)

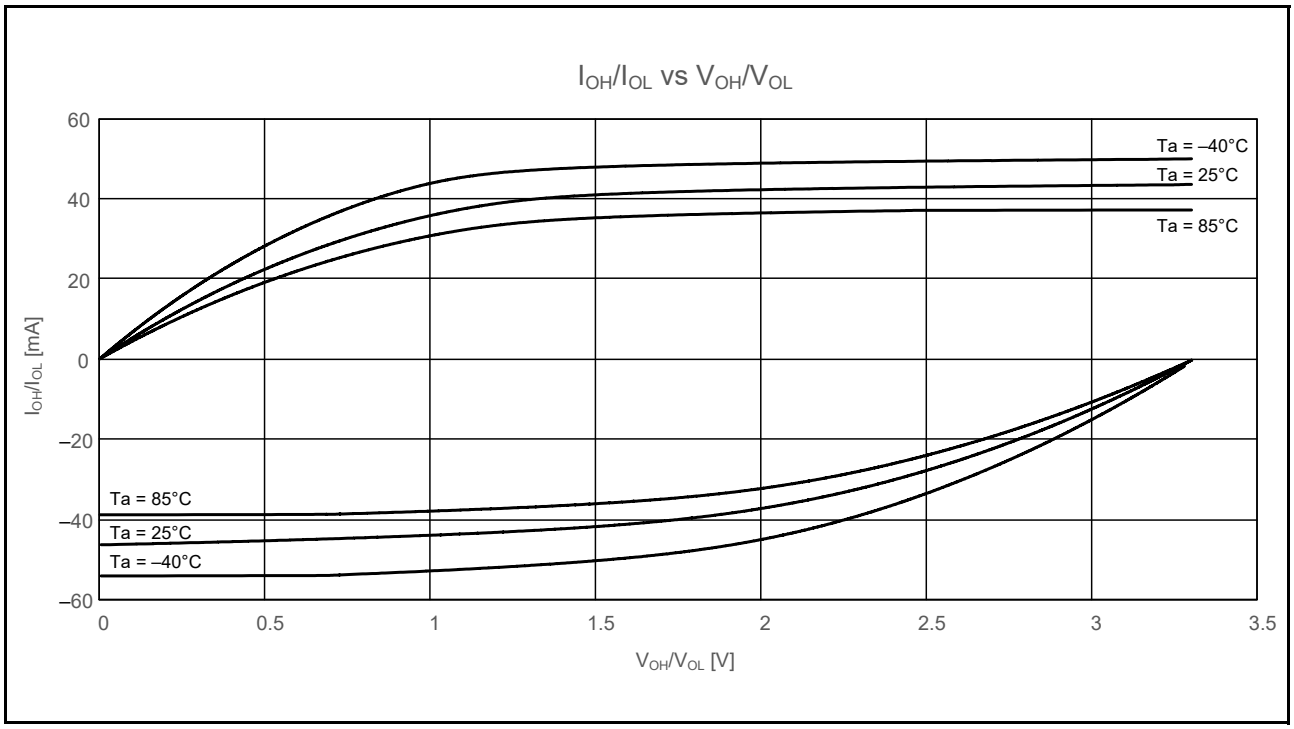


Figure 51.15 V_{OH}/V_{OL} and I_{OH}/I_{OL} Temperature Characteristics at VCC = 3.3 V When High-Drive Output is Selected (Reference Data)

Downloaded from Arrow.com.

51.2.3 Normal I/O Pin Output Characteristics (3)

Figure 51.16 to Figure 51.18 show the characteristics of the RIIC output pin.

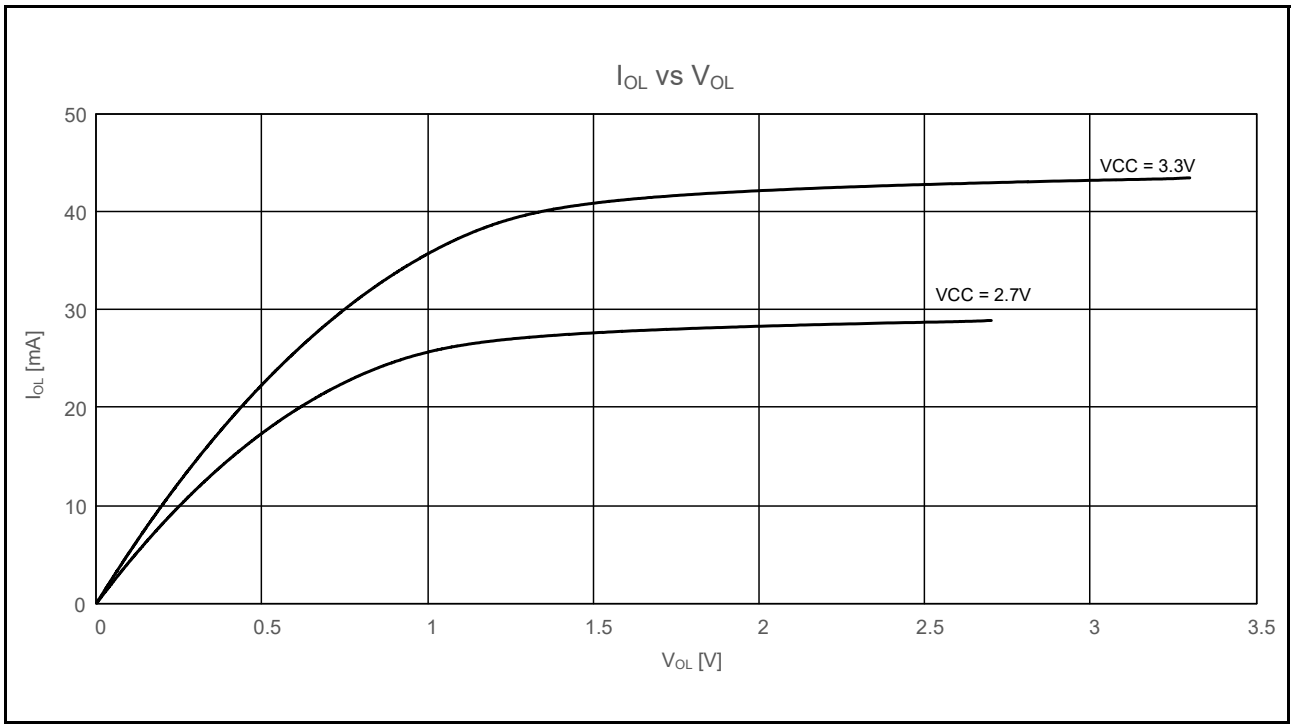


Figure 51.16 V_{OL} and I_{OL} Voltage Characteristics of RIIC Output Pin at T_a = 25°C (Reference Data)

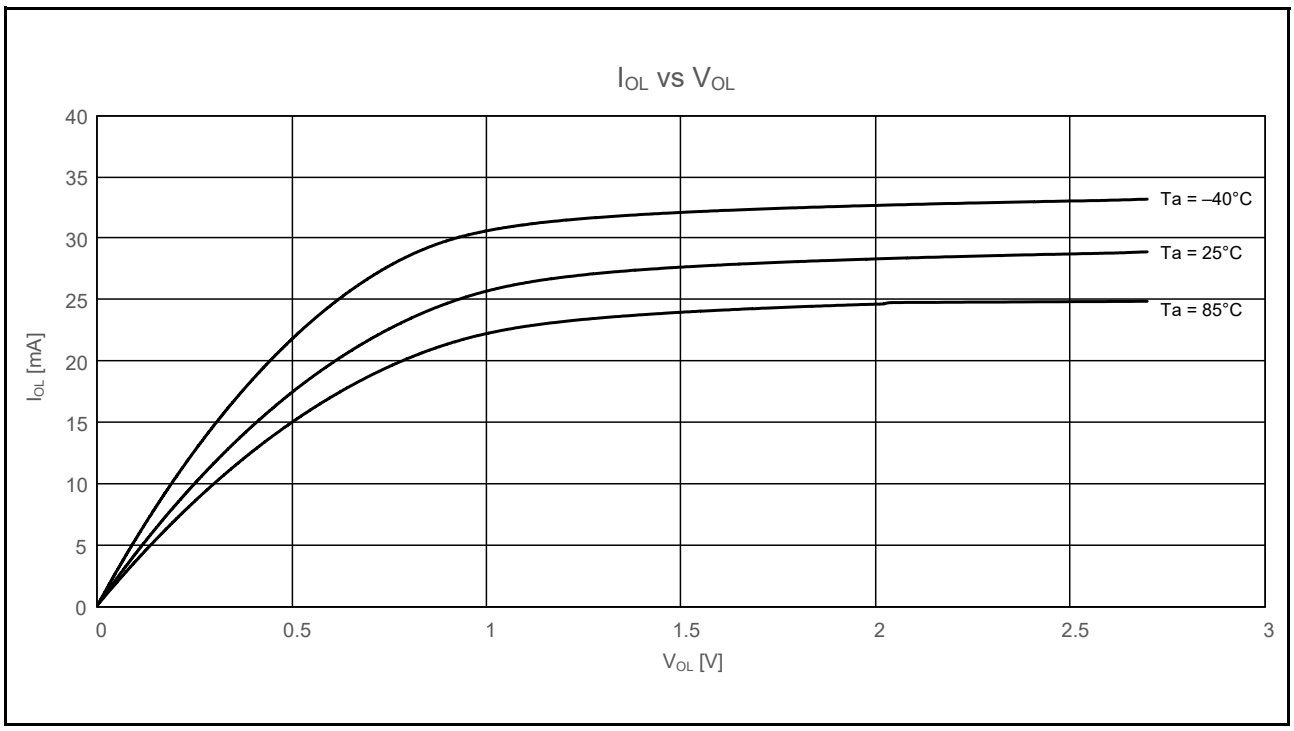


Figure 51.17 V_{OL} and I_{OL} Temperature Characteristics of RIIC Output Pin at VCC = 2.7 V (Reference Data)

Downloaded from Arrow.com.

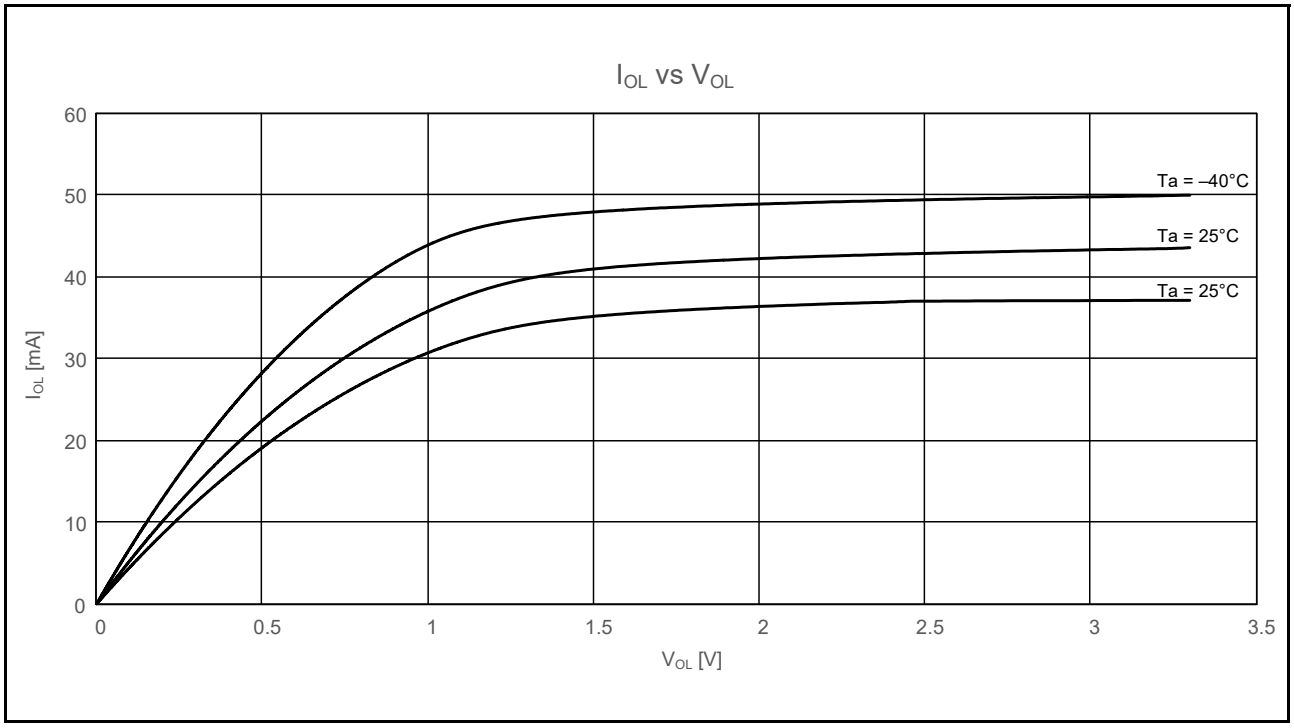


Figure 51.18 V_{OL} and I_{OL} Temperature Characteristics of RIIC Output Pin at VCC = 3.3 V (Reference Data)

Downloaded from Arrow.com.

51.3 AC Characteristics

51.3.1 Clock Timing

Table 51.21 Operating Frequency Value (High-Speed Operating Mode)

Conditions: 1.8 V ≤ VCC = VCC_USB = AVCC0 = VCC_RF = AVCC_RF ≤ 3.6 V, VSS = AVSS0 = VSS_USB = VSS_RF = 0 V, T_a = –40 to +85°C

Item		Symbol	VCC				Unit
			1.8 V ≤ VCC < 2.4 V	2.4 V ≤ VCC < 2.7 V	2.7 V ≤ VCC ≤ 3.6 V	When USB is in Use*3	
Maximum operating frequency*4	System clock (ICLK)	f_{max}	8	16	54	54	MHz
	FlashIF clock (FCLK)*1, *2		8	16	32	32	
	Peripheral module clock (PCLKA)		8	16	54	54	
	Peripheral module clock (PCLKB)		8	16	32	32	
	Peripheral module clock (PCLKD)		8	32	54	54	
	USB clock (UCLK)	f_{usb}	—	—	—	48	

Note 1. The lower-limit frequency of FCLK is 1 MHz during programming or erasing of the flash memory. When FCLK is in use at below 4 MHz, the frequency can be set to 1 MHz, 2 MHz, or 3 MHz. A non-integer frequency such as 1.5 MHz cannot be set.
Note 2. The frequency accuracy of FCLK must be within ±3.5%.
Note 3. The VCC_USB range is 3.0 to 3.6 V when the USB clock is in use.
Note 4. The maximum operating frequency listed above does not include errors of the external oscillator and internal oscillator. For details on the range for the guaranteed operation, see Table 51.24, Clock Timing.

Table 51.22 Operating Frequency Value (Middle-Speed Operating Mode)

Conditions: 1.8 V ≤ VCC = VCC_USB = AVCC0 = VCC_RF = AVCC_RF ≤ 3.6 V, VSS = AVSS0 = VSS_USB = VSS_RF = 0 V, T_a = –40 to +85°C

Item		Symbol	VCC				Unit
			1.8 V ≤ VCC < 2.4 V	2.4 V ≤ VCC < 2.7 V	2.7 V ≤ VCC ≤ 3.6 V	When USB is in Use*3	
Maximum operating frequency*4	System clock (ICLK)	f_{max}	8	12	12	12	MHz
	FlashIF clock (FCLK)*1, *2		8	12	12	12	
	Peripheral module clock (PCLKA)		8	12	12	12	
	Peripheral module clock (PCLKB)		8	12	12	12	
	Peripheral module clock (PCLKD)		8	12	12	12	
	USB clock (UCLK)	f_{usb}	—	—	—	48	

Note 1. The lower-limit frequency of FCLK is 1 MHz during programming or erasing of the flash memory. When using FCLK at below 4 MHz, the frequency can be set to 1 MHz, 2 MHz, or 3 MHz. A non-integer frequency such as 1.5 MHz cannot be set.
Note 2. The frequency accuracy of FCLK must be within ±3.5%.
Note 3. The VCC_USB range is 3.0 to 3.6 V when the USB clock is in use.
Note 4. The maximum operating frequency listed above does not include errors of the external oscillator and internal oscillator. For details on the range for the guaranteed operation, see Table 51.24, Clock Timing.

Downloaded from Arrow.com.

Table 51.23 Operating Frequency Value (Low-Speed Operating Mode)

Conditions: 1.8 V ≤ VCC = VCC_USB = AVCC0 = VCC_RF = AVCC_RF ≤ 3.6 V, VSS = AVSS0 = VSS_USB = VSS_RF = 0 V, T_a = –40 to +85°C

Item		Symbol	VCC			Unit
			1.8 V ≤ VCC < 2.4 V	2.4 V ≤ VCC < 2.7 V	2.7 V ≤ VCC ≤ 3.6 V	
Maximum operating frequency*3	System clock (ICLK)	f_{max}	32.768			kHz
	FlashIF clock (FCLK)*1		32.768			
	Peripheral module clock (PCLKA)		32.768			
	Peripheral module clock (PCLKB)		32.768			
	Peripheral module clock (PCLKD)*2		32.768			

Note 1. Programming and erasing the flash memory is impossible.
Note 2. The A/D converter cannot be used.
Note 3. The maximum operating frequency listed above does not include errors of the external oscillator. For details on the range for the guaranteed operation, see Table 51.24, Clock Timing.

Downloaded from Arrow.com.

Table 51.24 Clock Timing

Conditions: 1.8 V ≤ VCC = VCC_USB = AVCC0 = VCC_RF = AVCC_RF ≤ 3.6 V, VSS = AVSS0= VSS_USB = VSS_RF = 0 V, T_a = –40 to +85°C

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
EXTAL external clock input cycle time		t_{Xcyc}	50	—	—	ns	Figure 51.19
EXTAL external clock input high pulse width		t_{XH}	20	—	—	ns	
EXTAL external clock input low pulse width		t_{XL}	20	—	—	ns	
EXTAL external clock rise time		t_{Xr}	—	—	5	ns	
EXTAL external clock fall time		t_{Xf}	—	—	5	ns	
EXTAL external clock input wait time*1		t_{XWT}	0.5	—	—	μs	
Main clock oscillator oscillation frequency*2	2.4 ≤ VCC ≤ 3.6	f_{MAIN}	1	—	20	MHz	
	1.8 ≤ VCC < 2.4		1	—	8		
Main clock oscillation stabilization time (crystal)*2		$t_{MAINOSC}$	—	3	—	ms	Figure 51.20
Main clock oscillation stabilization time (ceramic resonator)*2		$t_{MAINOSC}$	—	50	—	μs	
LOCO clock oscillation frequency		f_{LOCO}	3.44	4.0	4.56	MHz	
LOCO clock oscillation stabilization time		t_{LOCO}	—	—	0.5	μs	Figure 51.21
IWDT-dedicated clock oscillation frequency		f_{ILOCO}	12.75	15	17.25	kHz	
IWDT-dedicated clock oscillation stabilization time		t_{ILOCO}	—	—	50	μs	Figure 51.22
Bluetooth-dedicated clock oscillation frequency		f_{BLECK}	—	32	—	MHz	
Bluetooth-dedicated low-speed on-chip oscillator oscillation frequency		$f_{BLELOCO}$	—	32.768	—	kHz	
HOCO clock oscillation frequency		f_{HOCO} (32 MHz)	31.36	32	32.64	MHz	T_a = 0 to +85°C
			31.04	32	32.96		T_a = –40 to +85°C
		f_{HOCO} (54 MHz)	52.96	54	55.08	MHz	T_a = 0 to +85°C
			52.38	54	55.62		T_a = –40 to +85°C
HOCO clock oscillation stabilization time		t_{HOCO}	—	—	30	μs	Figure 51.24
PLL input frequency*3		f_{PLLIN}	4	—	12.5	MHz	
PLL circuit oscillation frequency*3		f_{PLL}	24	—	54	MHz	
PLL clock oscillation stabilization time		t_{PLL}	—	—	50	μs	Figure 51.25
PLL free-running oscillation frequency		f_{PLLFR}	—	8	—	MHz	
USBPLL input frequency*5		f_{PLLIN}	—	4, 6, 8, 12	—	MHz	
USBPLL circuit oscillation frequency*5		f_{PLL}	—	48*6	—	MHz	
USBPLL clock oscillation stabilization time		t_{PLL}	—	—	50	μs	Figure 51.25
Sub-clock oscillator oscillation frequency*7		f_{SUB}	—	32.768	—	kHz	
Sub-clock oscillation stabilization time*4		t_{SUBOSC}	—	0.5	—	s	Figure 51.26

Note 1. Time until the clock can be used after the main clock oscillator stop bit (MOSCCR.MOSTP) is set to 0 (operating).

Note 2. Reference values when an 8-MHz resonator is used.
When specifying the main clock oscillator stabilization time, set the MOSCWTCR register with a stabilization time value that is equal to or greater than the resonator-manufacturer-recommended value.
After the MOSCCR.MOSTP bit is changed to enable the main clock oscillator, confirm that the OSCOVFSR.MOOVF flag has become 1, and then start using the main clock.

Note 3. The VCC range should be 2.4 to 3.6 V when the PLL is used.

Note 4. Reference values when a 32.768-kHz resonator is used.
After the setting of the SOSCCR.SOSTP bit or RCR3.RTCEN bit is changed to operate the sub-clock oscillator, only start using the sub-clock after the sub-clock oscillation stabilization wait time that is equal to or greater than the oscillator-manufacturer-recommended value has elapsed.

Note 5. The VCC range should be 3.0 to 3.6 V when the USBPLL is used.

Note 6. The oscillation frequency can be set to 48 MHz only.

Note 7. Only 32.768 kHz can be used.

Downloaded from Arrow.com.

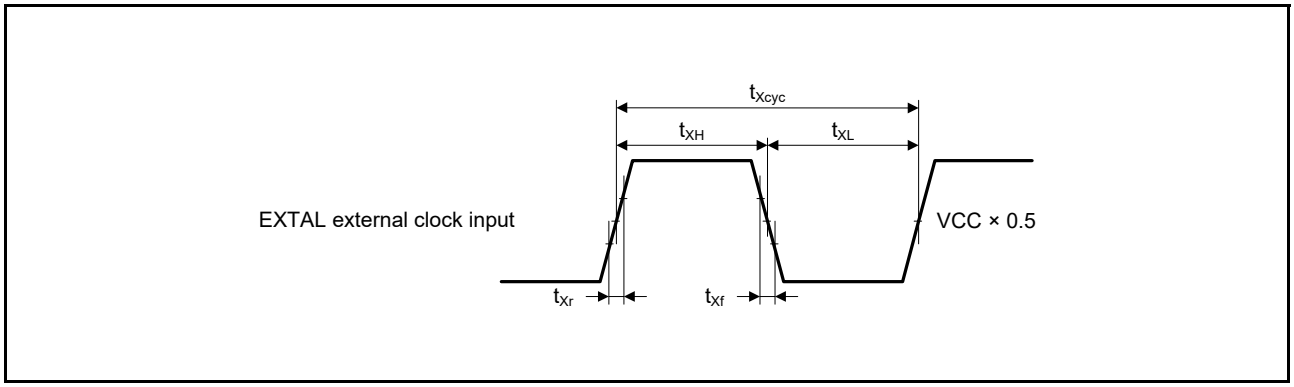


Figure 51.19 EXTAL External Clock Input Timing

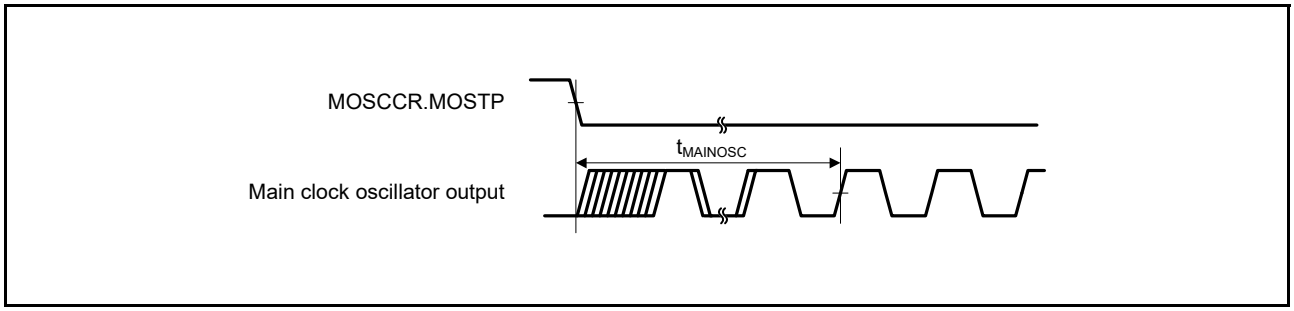


Figure 51.20 Main Clock Oscillation Start Timing

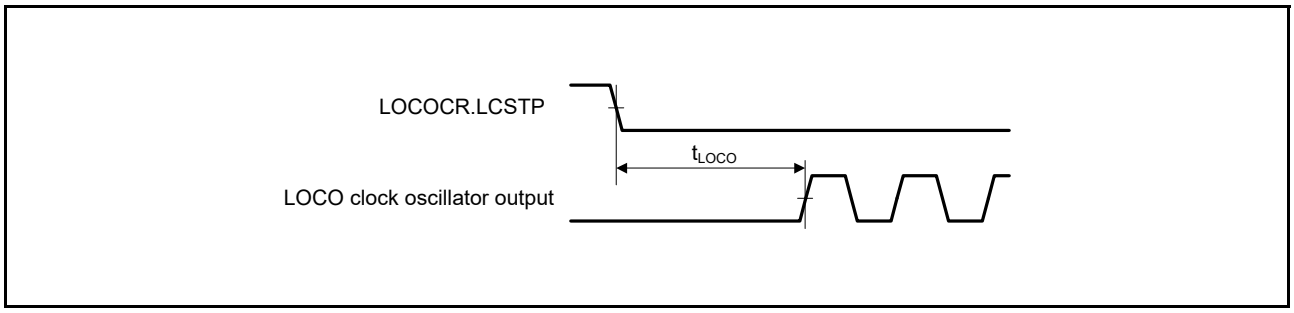


Figure 51.21 LOCO Clock Oscillation Start Timing

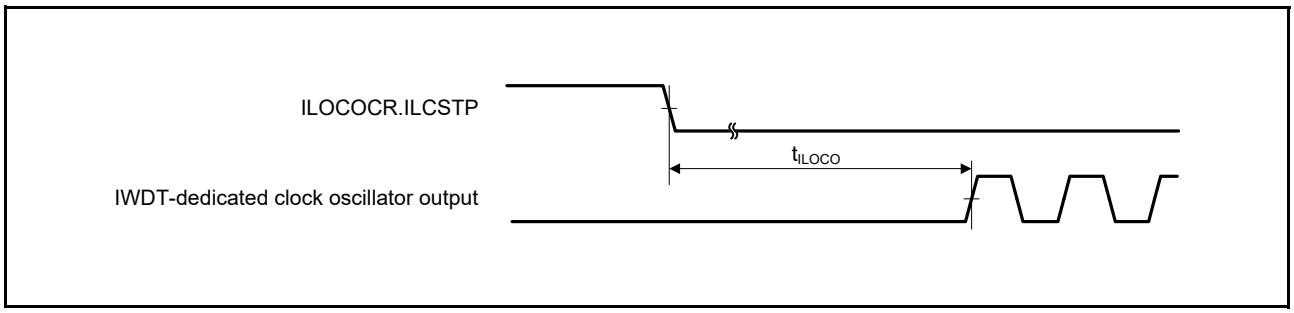


Figure 51.22 IWDT-Dedicated Clock Oscillation Start Timing

Downloaded from Arrow.com.

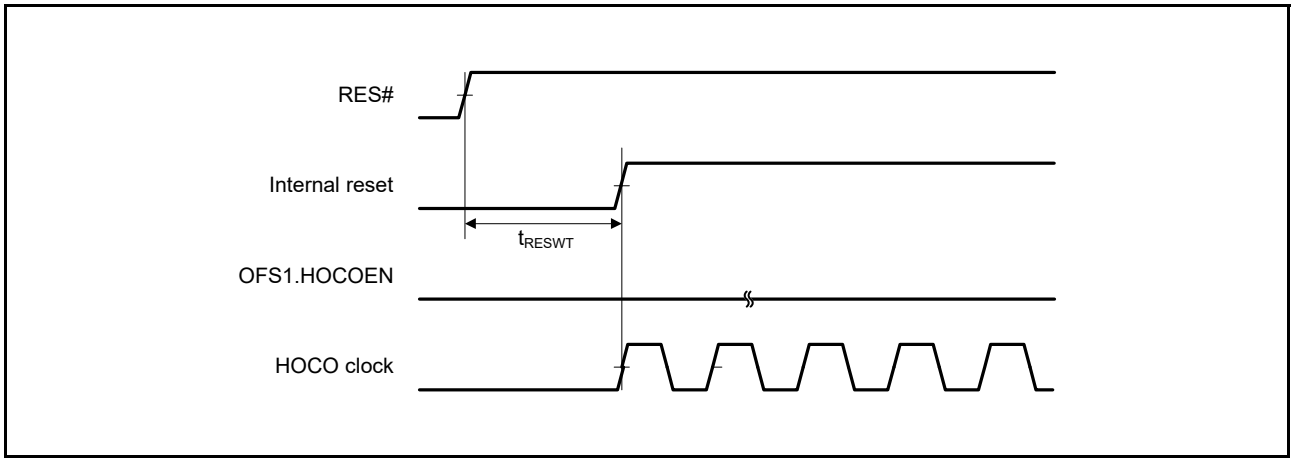


Figure 51.23 HOCO Clock Oscillation Start Timing (After Reset is Canceled by Setting OFS1.HOCOEN Bit to 0)

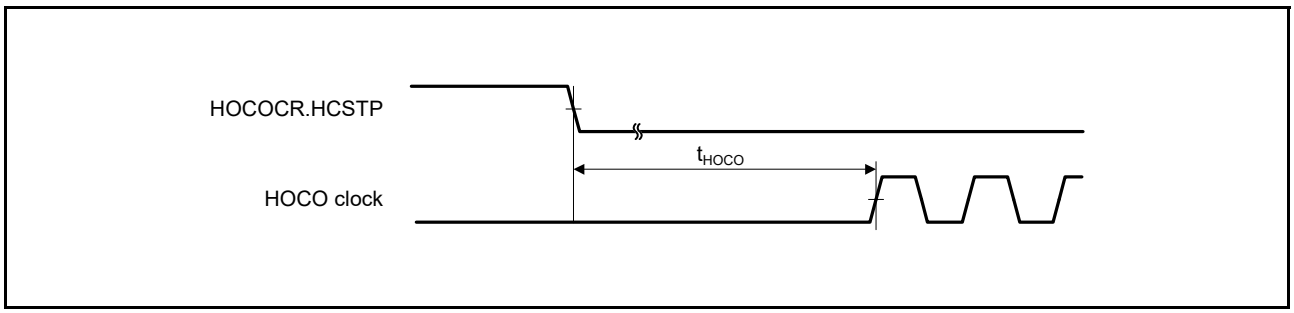


Figure 51.24 HOCO Clock Oscillation Start Timing (Oscillation is Started by Setting HOCOCR.HCSTP Bit)

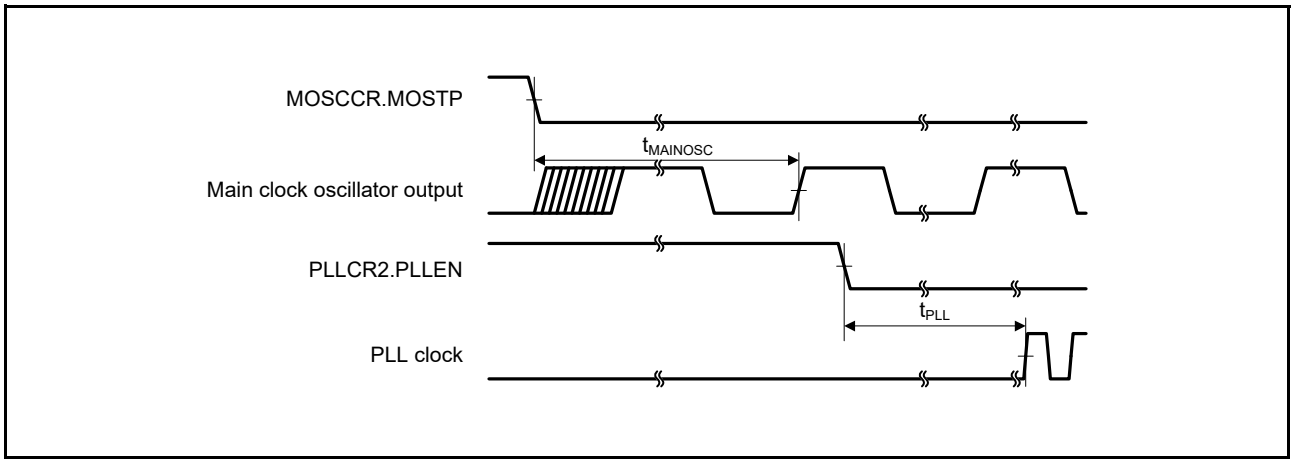


Figure 51.25 PLL Clock Oscillation Start Timing (PLL is Operated after Main Clock Oscillation Has Been Stabled)

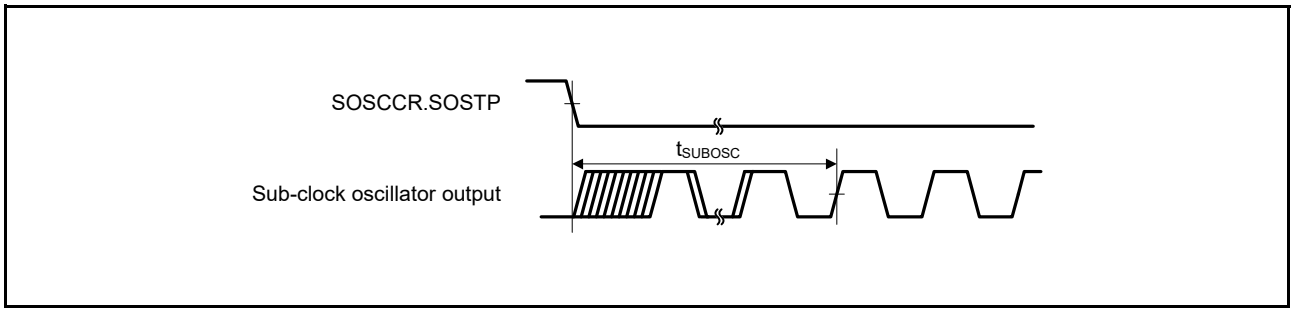


Figure 51.26 Sub-Clock Oscillation Start Timing

Downloaded from Arrow.com.

51.3.2 Reset Timing

Table 51.25 Reset Timing

Conditions: 1.8 V ≤ VCC = VCC_USB = AVCC0 = VCC_RF = AVCC_RF ≤ 3.6 V, VSS = AVSS0 = VSS_USB = VSS_RF = 0 V, T_a = –40 to +85°C

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
RES# pulse width	At power-on	t_{RESWP}	3	—	—	ms	Figure 51.27
	Other than above	t_{RESW}	30	—	—	µs	Figure 51.28
Wait time after RES# cancellation (at power-on)	At normal startup*1	t_{RESWT}	—	8.5	—	ms	Figure 51.27
	During fast startup time*2	t_{RESWT}	—	560	—	µs	
Wait time after RES# cancellation (during powered-on state)		t_{RESWT}	—	120	—	µs	Figure 51.28
Independent watchdog timer reset period		t_{RESWIW}	—	1	—	IWDT clock cycle	Figure 51.29
Watchdog timer reset period		t_{RESWWW}	—	4	—	PCLKB cycle	
Software reset period		t_{RESWSW}	—	1	—	ICLK cycle	
Wait time after independent watchdog timer reset cancellation*3		t_{RESWT2}	—	300	—	µs	
Wait time after watchdog timer reset cancellation*4		t_{RESWT2}	—	300	—	µs	
Wait time after software reset cancellation		t_{RESWT2}	—	170	—	µs	

Note 1. When OFS1.(LVDAS, FASTSTUP) bits are 11b.
Note 2. When OFS1.(LVDAS, FASTSTUP) bits are a value other than 11b.
Note 3. When IWDTCR.CKS[3:0] bits are 0000b.
Note 4. When WDTCR.CKS[3:0] bits are 0001b.

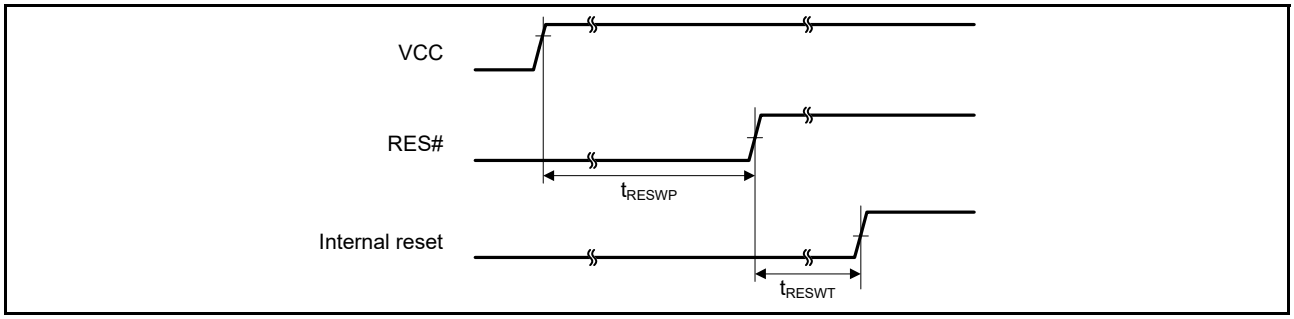


Figure 51.27 Reset Input Timing at Power-On

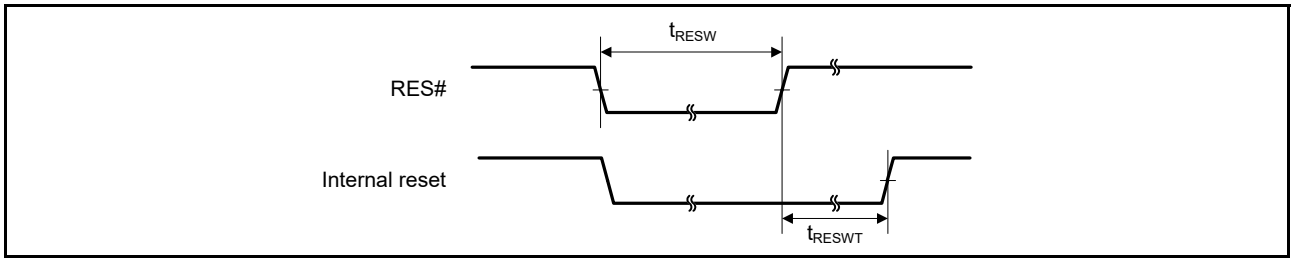


Figure 51.28 Reset Input Timing (1)

Downloaded from Arrow.com.

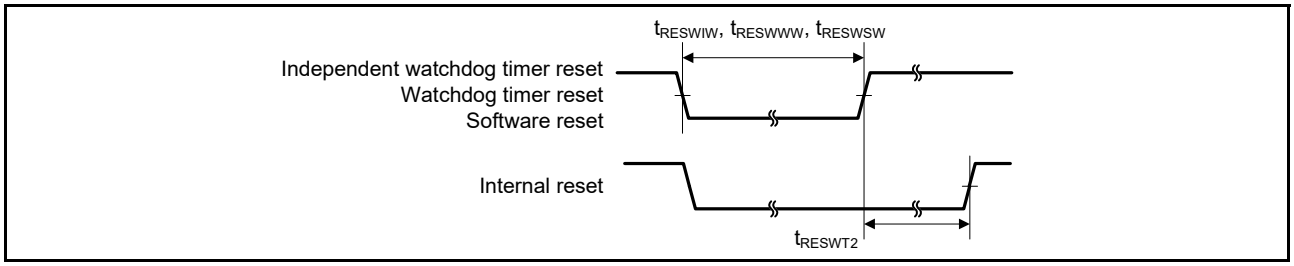


Figure 51.29 Reset Input Timing (2)

Downloaded from Arrow.com.

51.3.3 Timing of Recovery from Low Power Consumption Modes

Table 51.26 Timing of Recovery from Low Power Consumption Modes (1)

Conditions: 1.8 V ≤ VCC = VCC_USB = AVCC0 = VCC_RF = AVCC_RF ≤ 3.6 V, VSS = AVSS0 = VSS_USB = VSS_RF = 0 V, T_a = –40 to +85°C

Item				Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Recovery time from software standby mode*1	High-speed mode	Crystal connected to main clock oscillator	Main clock oscillator operating*2	t_{SBYMC}	—	2	3	ms	Figure 51.30
		External clock input to main clock oscillator	Main clock oscillator operating*3	t_{SBYEX}	—	35	50	µs	
		Sub-clock oscillator operating		t_{SBYSC}	—	650	800	µs	
		HOCO clock oscillator operating		t_{SBYHO}	—	40	55	µs	
		LOCO clock oscillator operating		t_{SBYLO}	—	40	55	µs	

Note 1. The recovery time varies depending on the state of each oscillator when the WAIT instruction is executed. When multiple oscillators are operating, the recovery time varies depending on the operating state of the oscillators that are not selected as the system clock source. The above table applies when only the corresponding clock is operating.
Note 2. When the frequency of the crystal is 20 MHz.
When the main clock oscillator wait control register (MOSCWTCR) is set to 04h.
Note 3. When the frequency of the external clock is 20 MHz.
When the main clock oscillator wait control register (MOSCWTCR) is set to 00h.

Table 51.27 Timing of Recovery from Low Power Consumption Modes (2)

Conditions: 1.8 V ≤ VCC = VCC_USB = AVCC0 = VCC_RF = AVCC_RF ≤ 3.6 V, VSS = AVSS0 = VSS_USB = VSS_RF = 0 V, T_a = –40 to +85°C

Item				Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Recovery time from software standby mode*1	Middle-speed mode	Crystal connected to main clock oscillator	Main clock oscillator operating*2	t_{SBYMC}	—	2	3	ms	Figure 51.30
			Main clock oscillator and PLL circuit operating*3	t_{SBYPC}	—	2	3	ms	
		External clock input to main clock oscillator	Main clock oscillator operating*4	t_{SBYEX}	—	3	4	µs	
			Main clock oscillator and PLL circuit operating*5	t_{SBYPE}	—	65	85	µs	
		Sub-clock oscillator operating		t_{SBYSC}	—	600	750	µs	
		HOCO clock oscillator operating*6		t_{SBYHO}	—	40	50	µs	
		LOCO clock oscillator operating		t_{SBYLO}	—	5	7	µs	

Note 1. The recovery time varies depending on the state of each oscillator when the WAIT instruction is executed. When multiple oscillators are operating, the recovery time varies depending on the operating state of the oscillators that are not selected as the system clock source. The above table applies when only the corresponding clock is operating.
Note 2. When the frequency of the crystal is 12 MHz.
When the main clock oscillator wait control register (MOSCWTCR) is set to 04h.
Note 3. When the frequency of PLL is 12 MHz.
When the main clock oscillator wait control register (MOSCWTCR) is set to 04h.
Note 4. When the frequency of the external clock is 12 MHz.
When the main clock oscillator wait control register (MOSCWTCR) is set to 00h.
Note 5. When the frequency of PLL is 12 MHz.
When the main clock oscillator wait control register (MOSCWTCR) is set to 00h.
Note 6. This is the case when HOCO is selected as the system clock and its frequency division is set to be 8 MHz.

Downloaded from Arrow.com.

Table 51.28 Timing of Recovery from Low Power Consumption Modes (3)

Conditions: 1.8 V ≤ VCC = VCC_USB = AVCC0 = VCC_RF = AVCC_RF ≤ 3.6 V, VSS = AVSS0 = VSS_USB = VSS_RF = 0 V, T_a = –40 to +85°C

Item			Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Recovery time from software standby mode*1	Low-speed mode	Sub-clock oscillator operating	t_{SBYSC}	—	600	750	µs	Figure 51.30

Note 1. The sub-clock continues oscillating in software standby mode during low-speed mode.

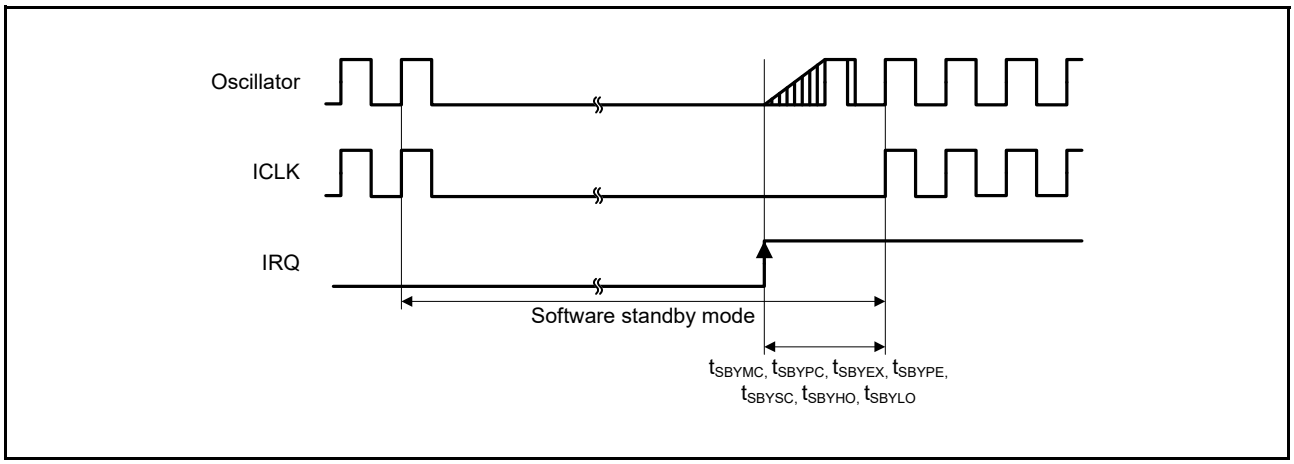


Figure 51.30 Software Standby Mode Recovery Timing

Table 51.29 Timing of Recovery from Low Power Consumption Modes (4)

Conditions: 1.8 V ≤ VCC = VCC_USB = AVCC0 = VCC_RF = AVCC_RF ≤ 3.6 V, VSS = AVSS0 = VSS_USB = VSS_RF = 0 V, T_a = –40 to +85°C

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Recovery time from deep sleep mode*1	High-speed mode*2	t_{DSLP}	—	2	3.5	µs	Figure 51.31
	Middle-speed mode*3	t_{DSLP}	—	3	4	µs	
	Low-speed mode*4	t_{DSLP}	—	400	500	µs	

Note 1. Oscillators continue oscillating in deep sleep mode.
Note 2. When the frequency of the system clock is 32 MHz.
Note 3. When the frequency of the system clock is 12 MHz.
Note 4. When the frequency of the system clock is 32 kHz.

Downloaded from Arrow.com.

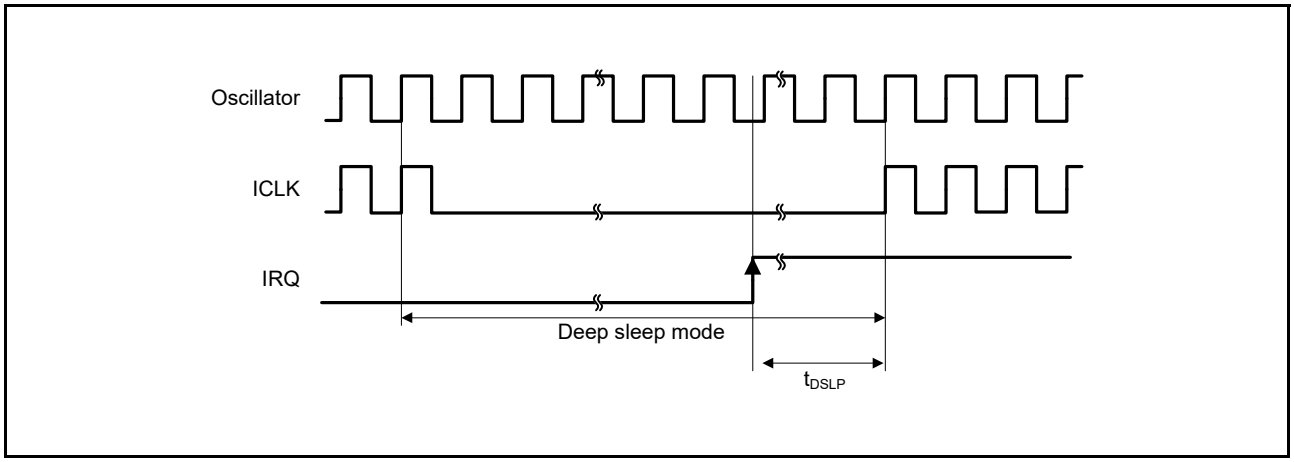


Figure 51.31 Deep Sleep Mode Recovery Timing

Table 51.30 Operating Mode Transition Time

Conditions: 1.8 V ≤ VCC = VCC_USB = AVCC0 = VCC_RF = AVCC_RF ≤ 3.6 V, VSS = AVSS0 = VSS_USB = VSS_RF = 0 V, T_a = –40 to +85°C

Mode before Transition	Mode after Transition	ICLK Frequency	Transition Time			Unit
			Min.	Typ.	Max.	
High-speed operating mode	Middle-speed operating modes	8 MHz	—	10	—	µs
Middle-speed operating modes	High-speed operating mode	8 MHz	—	37.5	—	µs
Low-speed operating mode	Middle-speed operating mode, high-speed operating mode	32.768 kHz	—	215	—	µs
Middle-speed operating mode, high-speed operating mode	Low-speed operating mode	32.768 kHz	—	185	—	µs

Note: Values when the frequencies of PCLKA, PCLKB, PCLKD, and FCLK, are not divided.

Downloaded from Arrow.com.

51.3.4 Control Signal Timing

Table 51.31 Control Signal Timing

Conditions: 1.8 V ≤ VCC = VCC_USB = AVCC0 = VCC_RF = AVCC_RF ≤ 3.6 V, VSS = AVSS0 = VSS_USB = VSS_RF = 0 V, T_a = –40 to +85°C

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions	
NMI pulse width	t_{NMIW}	200	—	—	ns	NMI digital filter is disabled (NMIFLTE.NFLTEN = 0)	t_{Pcyc} × 2 ≤ 200 ns
		t_{Pcyc} × 2*1	—	—			t_{Pcyc} × 2 > 200 ns
		200	—	—		NMI digital filter is enabled (NMIFLTE.NFLTEN = 1)	t_{NMICK} × 3 ≤ 200 ns
		t_{NMICK} × 3.5*2	—	—			t_{NMICK} × 3 > 200 ns
IRQ pulse width	t_{IRQW}	200	—	—	ns	IRQ digital filter is disabled (IRQFLTE0.FLTENi = 0)	t_{Pcyc} × 2 ≤ 200 ns
		t_{Pcyc} × 2*1	—	—			t_{Pcyc} × 2 > 200 ns
		200	—	—		IRQ digital filter is enabled (IRQFLTE0.FLTENi = 1)	t_{IRQCK} × 3 ≤ 200 ns
		t_{IRQCK} × 3.5*3	—	—			t_{IRQCK} × 3 > 200 ns

Note: 200 ns minimum in software standby mode.
Note 1. t_{Pcyc} indicates the cycle of PCLKB.
Note 2. t_{NMICK} indicates the cycle of the NMI digital filter sampling clock.
Note 3. t_{IRQCK} indicates the cycle of the IRQi digital filter sampling clock (i = 0 to 7).

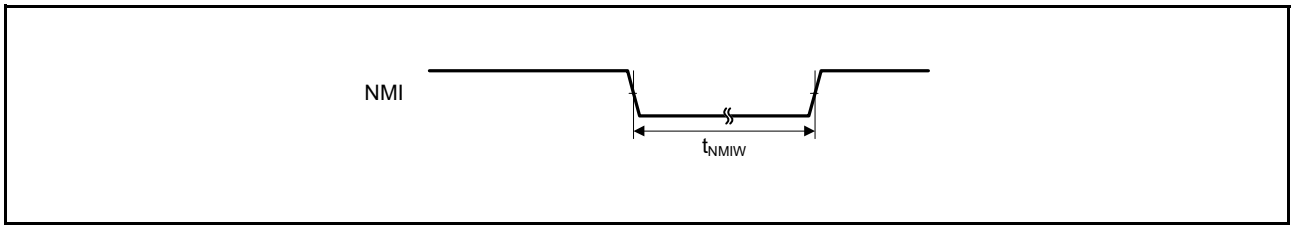


Figure 51.32 NMI Interrupt Input Timing

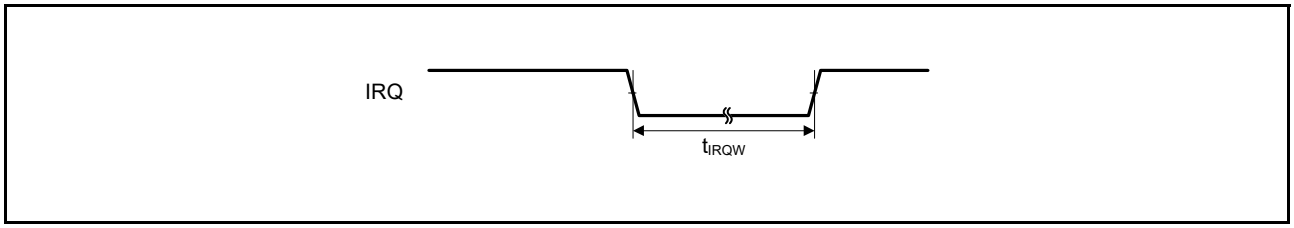


Figure 51.33 IRQ Interrupt Input Timing

Downloaded from Arrow.com.

51.3.5 Timing of On-Chip Peripheral Modules

51.3.5.1 Timing of I/O Ports

Table 51.32 Timing of I/O Ports

Conditions: 1.8 V ≤ VCC = VCC_USB = AVCC0 = VCC_RF = AVCC_RF ≤ 3.6 V, VSS = AVSS0 = VSS_USB = VSS_RF = 0 V,
T_a = –40 to +85°C

Item		Symbol	Min.	Max.	Unit *1	Test Conditions
I/O ports	Input data pulse width	t_{PRW}	1.5	—	t_{Pcyc}	Figure 51.34

Note 1. t_{Pcyc}: PCLK cycle

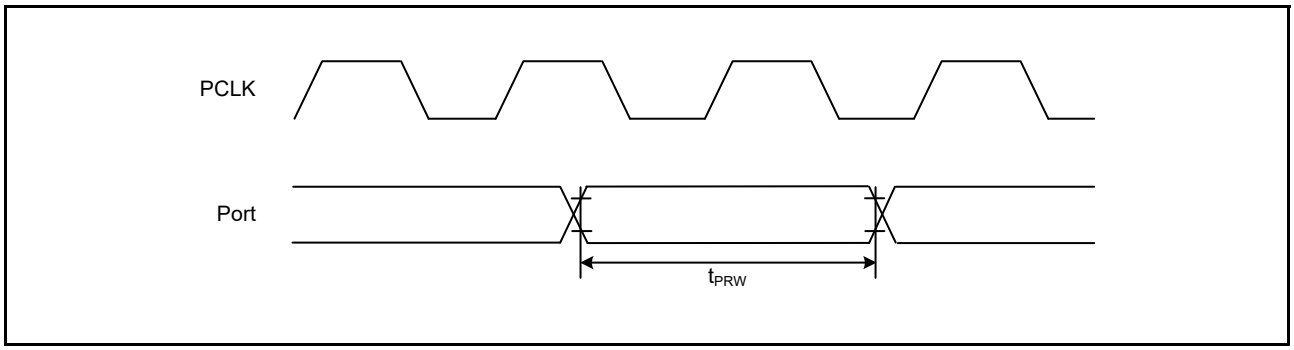


Figure 51.34 I/O Port Input Timing

Downloaded from Arrow.com.

51.3.5.2 Timing of MTU/TPU

Table 51.33 Timing of MTU/TPU

Conditions: 1.8 V ≤ VCC = VCC_USB = AVCC0 = VCC_RF = AVCC_RF ≤ 3.6 V, VSS = AVSS0 = VSS_USB = VSS_RF = 0 V,
T_a = –40 to +85°C

Item			Symbol	Min.	Max.	Unit *1	Test Conditions
MTU/TPU	Input capture input pulse width	Single-edge setting	t_{TICW}	1.5	—	t_{Pcyc}	Figure 51.35
		Both-edge setting		2.5	—		
	Timer clock pulse width	Single-edge setting	t_{TCKWH}, t_{TCKWL}	1.5	—	t_{Pcyc}	Figure 51.36
		Both-edge setting		2.5	—		
		Phase counting mode		2.5	—		

Note 1. t_{Pcyc}: PCLK cycle

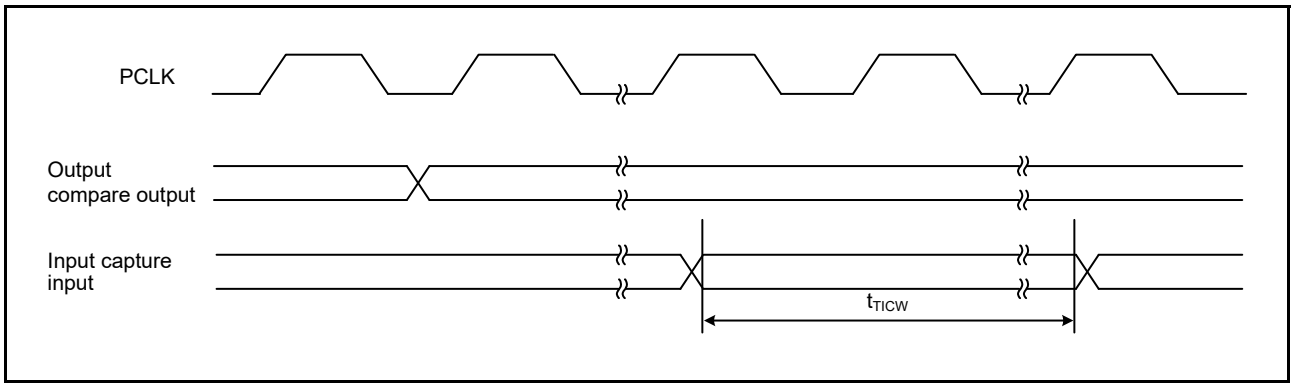


Figure 51.35 MTU Input/Output Timing

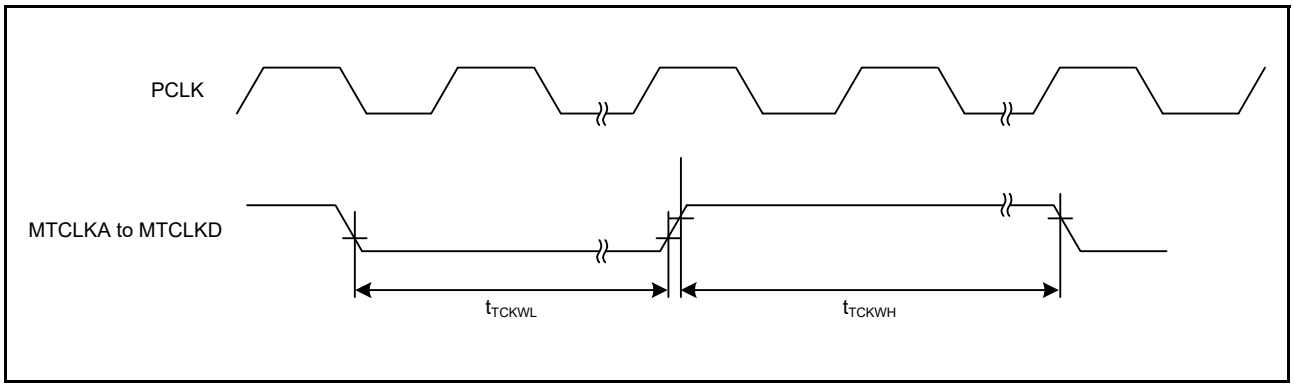


Figure 51.36 MTU Clock Input Timing

Downloaded from Arrow.com.

51.3.5.3 Timing of POE

Table 51.34 Timing of POE

Conditions: 1.8 V ≤ VCC = VCC_USB = AVCC0 = VCC_RF = AVCC_RF ≤ 3.6 V, VSS = AVSS0 = VSS_USB = VSS_RF = 0 V,
T_a = –40 to +85°C

Item		Symbol	Min.	Max.	Unit *1	Test Conditions
POE	POE# input pulse width	t_{POEW}	1.5	—	t_{Pcyc}	Figure 51.37

Note 1. t_{Pcyc}: PCLK cycle

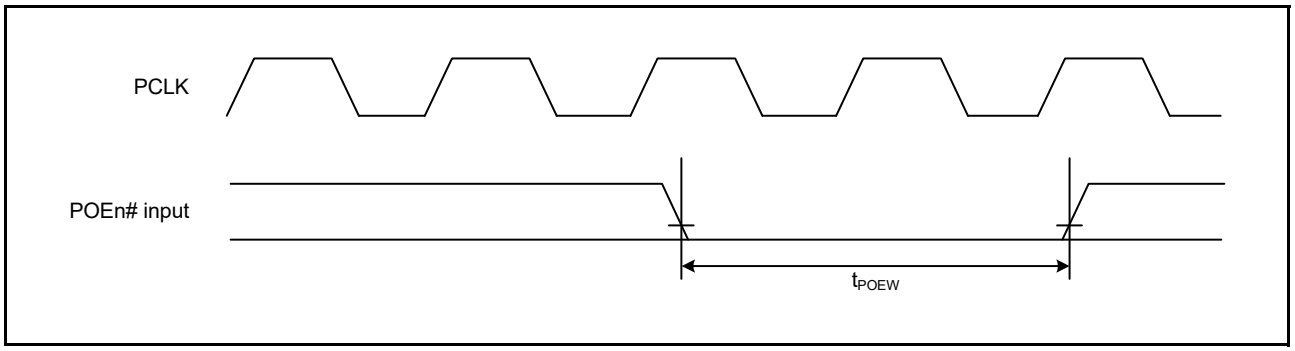


Figure 51.37 POE# Input Timing

51.3.5.4 Timing of TMR

Table 51.35 Timing of TMR

Conditions: 1.8 V ≤ VCC = VCC_USB = AVCC0 = VCC_RF = AVCC_RF ≤ 3.6 V, VSS = AVSS0 = VSS_USB = VSS_RF = 0 V,
T_a = –40 to +85°C

Item			Symbol	Min.	Max.	Unit *1	Test Conditions
TMR	Timer clock pulse width	Single-edge setting	t_{TMCWH}, t_{TMCWL}	1.5	—	t_{Pcyc}	Figure 51.38
		Both-edge setting		2.5	—		

Note 1. t_{Pcyc}: PCLK cycle

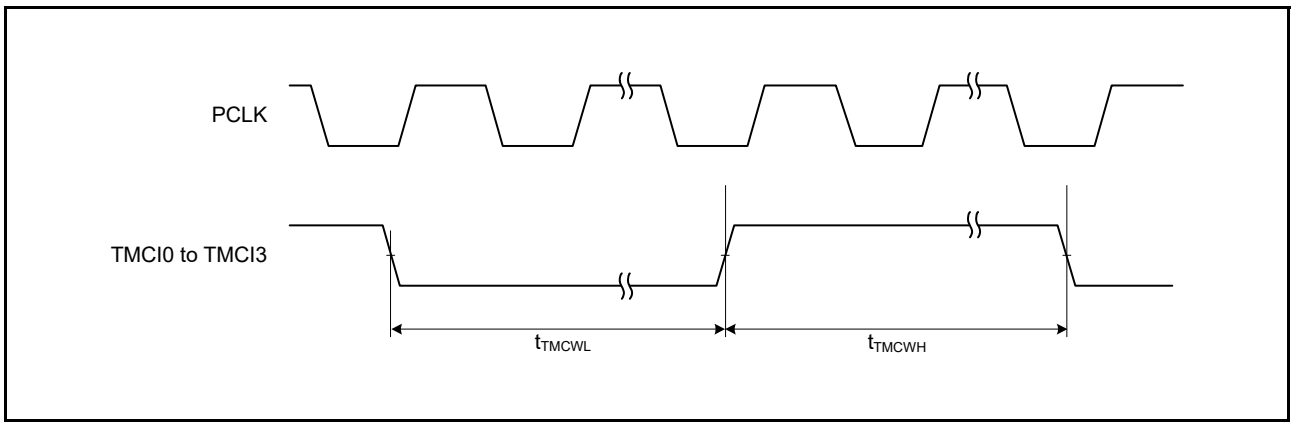


Figure 51.38 TMR Clock Input Timing

Downloaded from Arrow.com.

51.3.5.5 Timing of SCI

Table 51.36 Timing of SCI

Conditions: 1.8 V ≤ VCC = VCC_USB = AVCC0 = VCC_RF = AVCC_RF ≤ 3.6 V, VSS = AVSS0 = VSS_USB = VSS_RF = 0 V,
T_a = –40 to +85°C

Item				Symbol	Min.	Max.	Unit *1	Test Conditions
SCI	Input clock cycle time		Asynchronous	t_{Scyc}	4	—	t_{Pcyc}	Figure 51.39
			Clock synchronous		6	—		
	Input clock pulse width			t_{SCKW}	0.4	0.6	t_{Scyc}	
	Input clock rise time			t_{SCKr}	—	20	ns	
	Input clock fall time			t_{SCKf}	—	20	ns	
	Output clock cycle time		Asynchronous	t_{Scyc}	16	—	t_{Pcyc}	Figure 51.40
			Clock synchronous		4	—		
	Output clock pulse width			t_{SCKW}	0.4	0.6	t_{Scyc}	
	Output clock rise time			t_{SCKr}	—	20	ns	
	Output clock fall time			t_{SCKf}	—	20	ns	
	Transmit data delay time (master)	Clock synchronous		t_{TXD}	—	40	ns	
	Transmit data delay time (slave)	Clock synchronous	2.7 V or above		—	65	ns	
			1.8 V or above		—	100	ns	
	Receive data setup time (master)	Clock synchronous	2.7 V or above	t_{RXS}	65	—	ns	
			1.8 V or above		90	—	ns	
	Receive data setup time (slave)	Clock synchronous			40	—	ns	
	Receive data hold time	Clock synchronous		t_{RXH}	40	—	ns	

Note 1. t_{Pcyc}: PCLK cycle

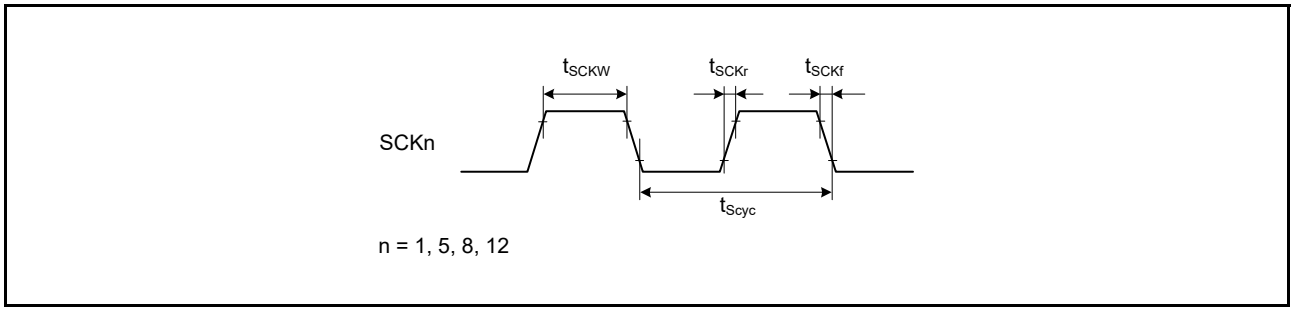


Figure 51.39 SCK Clock Input Timing

Downloaded from Arrow.com.

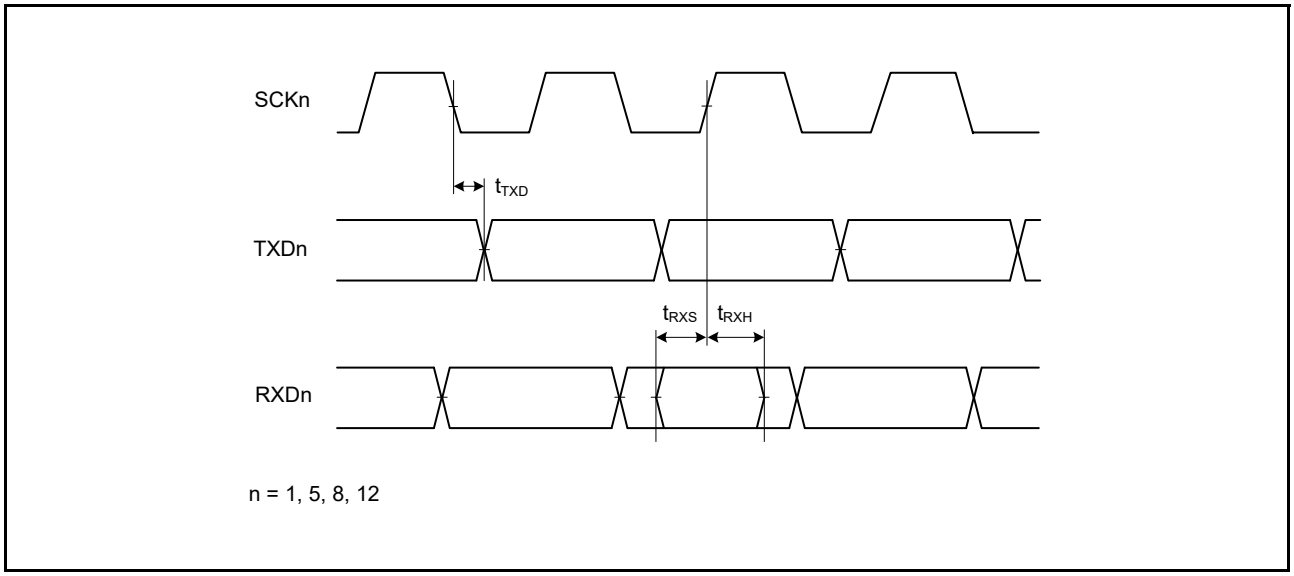


Figure 51.40 SCI Input/Output Timing: Clock Synchronous Mode

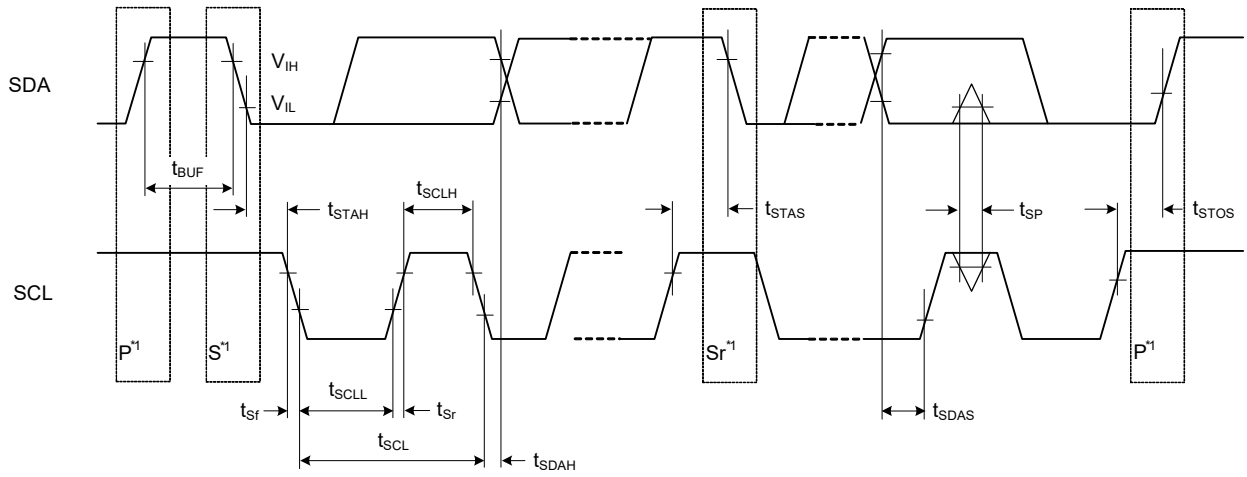
Table 51.37 Timing of Simple I²C

Conditions: 2.7 V ≤ VCC = VCC_USB = AVCC0 = VCC_RF = AVCC_RF ≤ 3.6 V, VSS = AVSS0 = VSS_USB = VSS_RF = 0 V, fPCLKB ≤ 32 MHz, T_a = –40 to +85°C

Item		Symbol	Min.*1	Max.	Unit	Test Conditions
Simple I²C (Standard mode)	SSDA rise time	t_{Sr}	—	1000	ns	Figure 51.41
	SSDA fall time	t_{Sf}	—	300	ns	
	SSDA spike pulse removal time	t_{SP}	0	4 × t_{Pcyc}	ns	
	Data setup time	t_{SDAS}	250	—	ns	
	Data hold time	t_{SDAH}	0	—	ns	
	SSCL, SSDA capacitive load	C_b	—	400	pF	
Simple I²C (Fast mode)	SSDA rise time	t_{Sr}	—	300	ns	Figure 51.41
	SSDA fall time	t_{Sf}	—	300	ns	
	SSDA spike pulse removal time	t_{SP}	0	4 × t_{Pcyc}	ns	
	Data setup time	t_{SDAS}	100	—	ns	
	Data hold time	t_{SDAH}	0	—	ns	
	SSCL, SSDA capacitive load	C_b	—	400	pF	

Note: t_{Pcyc}: PCLK cycle
Note 1. C_b is the total capacitance of the bus lines.

Downloaded from Arrow.com.



Note 1. S, P, and Sr indicate the following conditions, respectively.
S: START condition
P: STOP condition
Sr: Repeated START condition

Test conditions
V_{IH} = VCC × 0.7, V_{IL} = VCC × 0.3

Figure 51.41 RIIC Bus Interface Input/Output Timing and Simple I²C Bus Interface Input/Output Timing

Table 51.38 Timing of Simple SPI

Conditions: 1.8 V ≤ VCC = VCC_USB = AVCC0 = VCC_RF = AVCC_RF ≤ 3.6 V, VSS = AVSS0 = VSS_USB = VSS_RF = 0 V,
T_a = –40 to +85°C

Item			Symbol	Min.	Max.	Unit*1	Test Conditions
Simple SPI	SCK clock cycle output (master)		t_{SPcyc}	4	65536	t_{Pcyc}	Figure 51.42
	SCK clock cycle input (slave)			6	65536	t_{Pcyc}	
	SCK clock high pulse width		t_{SPCKWH}	0.4	0.6	t_{SPcyc}	
	SCK clock low pulse width		t_{SPCKWL}	0.4	0.6	t_{SPcyc}	
	SCK clock rise/fall time		t_{SPCKr}, t_{SPCKf}	—	20	ns	
	Data input setup time (master)	2.7 V or above	t_{SU}	65	—	ns	Figure 51.43, Figure 51.44
		1.8 V or above		95	—		
	Data input setup time (slave)			40	—		
	Data input hold time		t_H	40	—	ns	
	SSL input setup time		t_{LEAD}	3	—	t_{SPcyc}	
	SSL input hold time		t_{LAG}	3	—	t_{SPcyc}	
	Data output delay time (master)		t_{OD}	—	40	ns	
	Data output delay time (slave)	2.7 V or above		—	65		
		1.8 V or above		—	100		
	Data output hold time (master)	2.7 V or above	t_{OH}	–10	—	ns	
		1.8 V or above		–20	—		
	Data output hold time (slave)			–10	—		
	Data rise/fall time		t_{Dr}, t_{Df}	—	20	ns	
	SSL input rise/fall time		t_{SSLr}, t_{SSLf}	—	20	ns	
	Slave access time		t_{SA}	—	6	t_{Pcyc}	Figure 51.45, Figure 51.46
	Slave output release time		t_{REL}	—	6	t_{Pcyc}	

Note 1. t_{Pcyc}: PCLK cycle

Downloaded from Arrow.com.

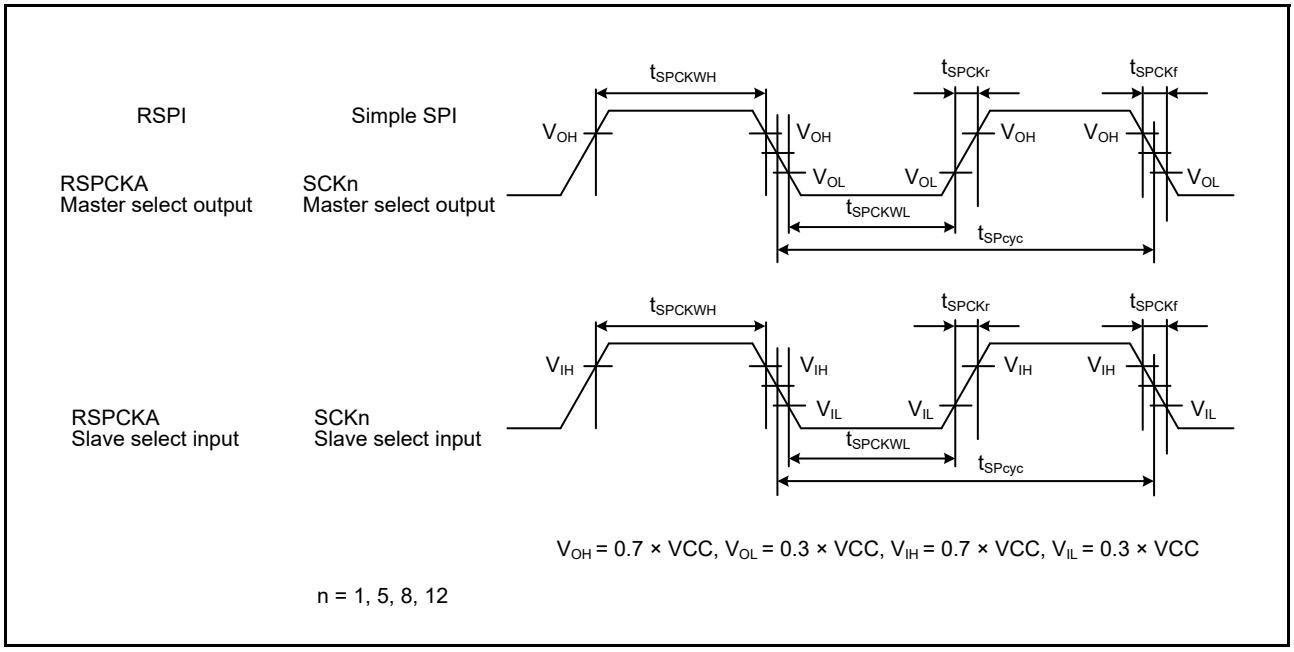


Figure 51.42 RSPI Clock Timing and Simple SPI Clock Timing

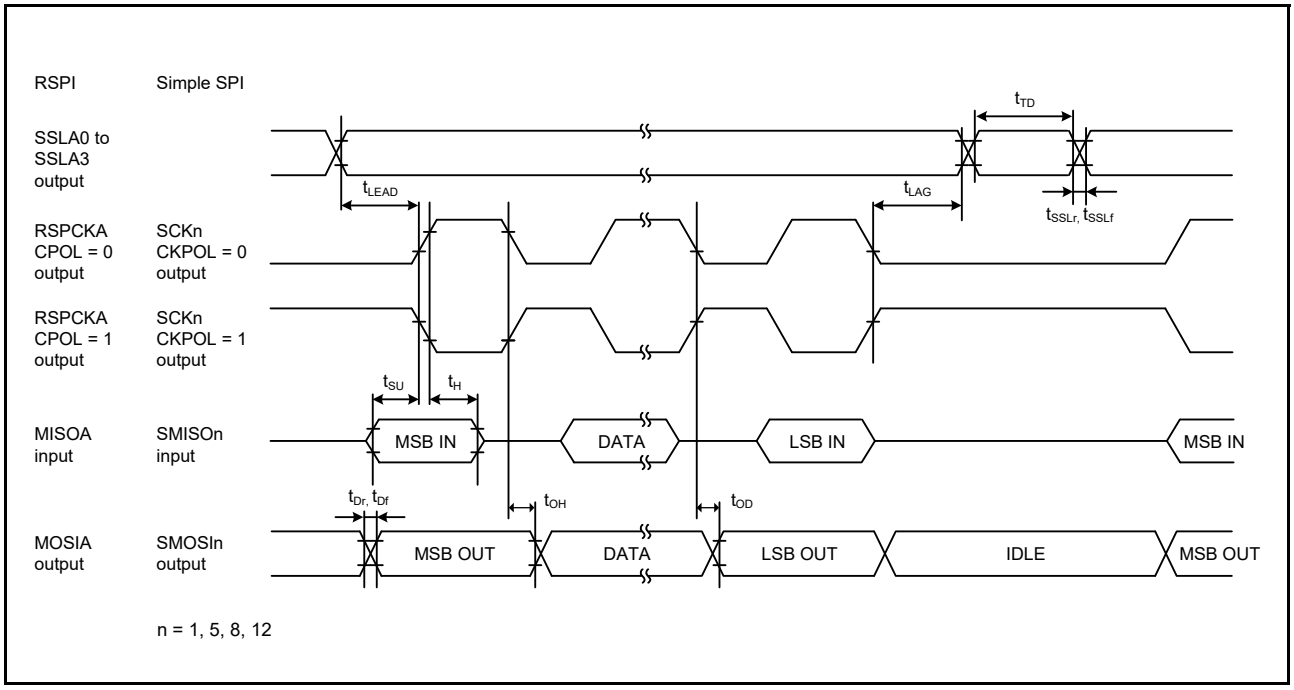


Figure 51.43 RSPI Timing (Master, CPHA = 0) and Simple SPI Clock Timing (Master, CKPH = 1)

Downloaded from Arrow.com.

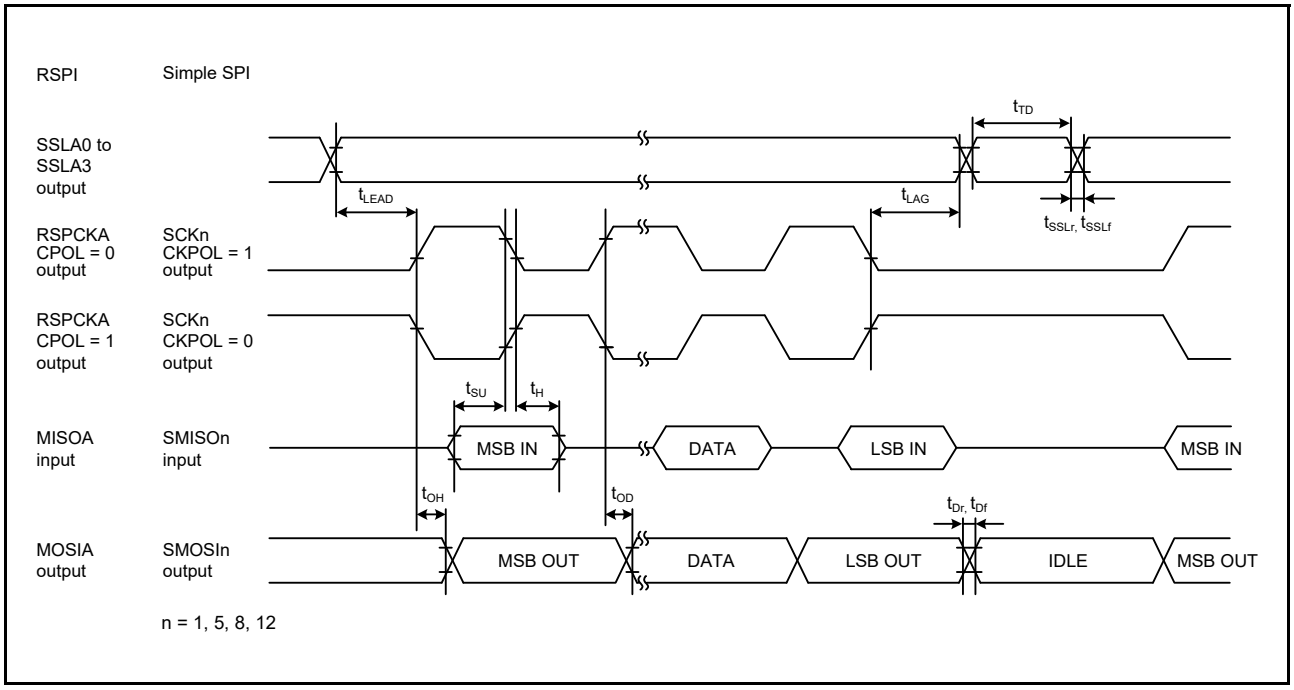


Figure 51.44 RSPI Timing (Master, CPHA = 1) and Simple SPI Clock Timing (Master, CKPH = 0)

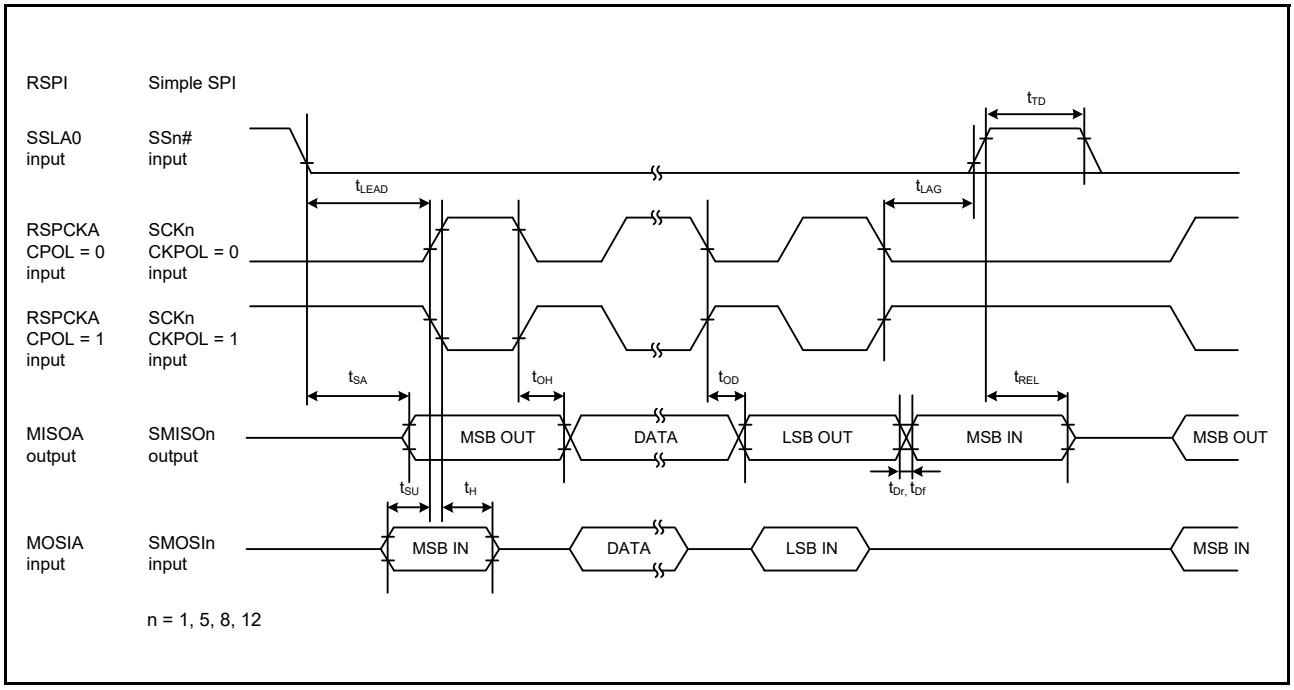


Figure 51.45 RSPI Timing (Slave, CPHA = 0) and Simple SPI Clock Timing (Slave, CKPH = 1)

Downloaded from Arrow.com.

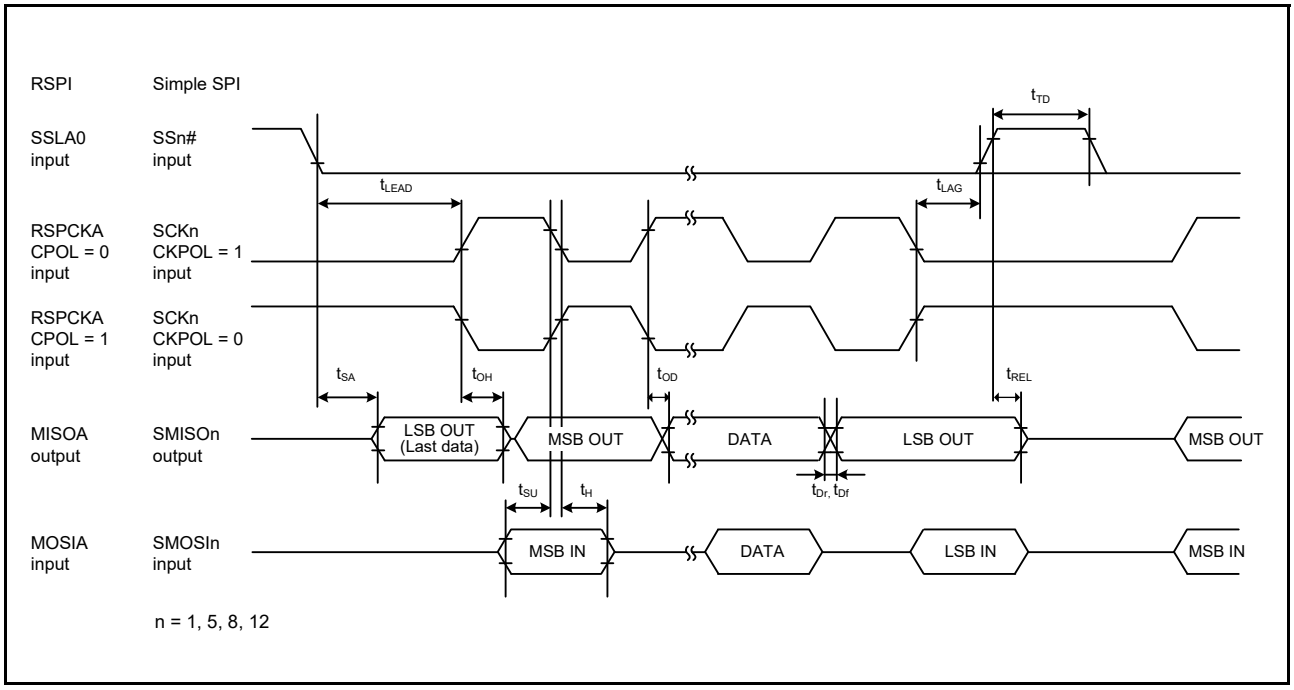


Figure 51.46 RSPI Timing (Slave, CPHA = 1) and Simple SPI Clock Timing (Slave, CKPH = 0)

Downloaded from Arrow.com.

51.3.5.6 Timing of RIIC

Table 51.39 Timing of RIIC

Conditions: 2.7 V ≤ VCC = VCC_USB = AVCC0 = VCC_RF = AVCC_RF ≤ 3.6 V, VSS = AVSS0 = VSS_USB = VSS_RF = 0 V, fPCLKB ≤ 32 MHz, T_a = –40 to +85°C

Item		Symbol	Min.*1, *2	Max.	Unit	Test Conditions
RIIC (Standard mode, SMBus)	SCL cycle time	t_{SCL}	6 (12) × t_{IICcyc} + 1300	—	ns	Figure 51.47
	SCL high pulse width	t_{SCLH}	3 (6) × t_{IICcyc} + 300	—	ns	
	SCL low pulse width	t_{SCLL}	3 (6) × t_{IICcyc} + 300	—	ns	
	SCL, SDA rise time	t_{Sr}	—	1000	ns	
	SCL, SDA fall time	t_{Sf}	—	300	ns	
	SCL, SDA spike pulse removal time	t_{SP}	0	1 (4) × t_{IICcyc}	ns	
	SDA bus free time	t_{BUF}	3 (6) × t_{IICcyc} + 300	—	ns	
	START condition hold time	t_{STAH}	t_{IICcyc} + 300	—	ns	
	Repeated START condition setup time	t_{STAS}	1000	—	ns	
	STOP condition setup time	t_{STOS}	1000	—	ns	
	Data setup time	t_{SDAS}	t_{IICcyc} + 50	—	ns	
	Data hold time	t_{SDAH}	0	—	ns	
	SCL, SDA capacitive load	C_b	—	400	pF	
RIIC (Fast mode)	SCL cycle time	t_{SCL}	6 (12) × t_{IICcyc} + 600	—	ns	Figure 51.47
	SCL high pulse width	t_{SCLH}	3 (6) × t_{IICcyc} + 300	—	ns	
	SCL low pulse width	t_{SCLL}	3 (6) × t_{IICcyc} + 300	—	ns	
	SCL, SDA rise time	t_{Sr}	—	300	ns	
	SCL, SDA fall time	t_{Sf}	—	300	ns	
	SCL, SDA spike pulse removal time	t_{SP}	0	1 (4) × t_{IICcyc}	ns	
	SDA bus free time	t_{BUF}	3 (6) × t_{IICcyc} + 300	—	ns	
	START condition hold time	t_{STAH}	t_{IICcyc} + 300	—	ns	
	Repeated START condition setup time	t_{STAS}	300	—	ns	
	STOP condition setup time	t_{STOS}	300	—	ns	
	Data setup time	t_{SDAS}	t_{IICcyc} + 50	—	ns	
	Data hold time	t_{SDAH}	0	—	ns	
	SCL, SDA capacitive load	C_b	—	400	pF	

Note: t_{IICcyc}: RIIC internal reference clock (IICφ) cycle

Note 1. The value in parentheses is used when the ICMR3.NF[1:0] bits are set to 11b while a digital filter is enabled with the ICFER.NFE bit = 1.

Note 2. C_b is the total capacitance of the bus lines.

Downloaded from Arrow.com.

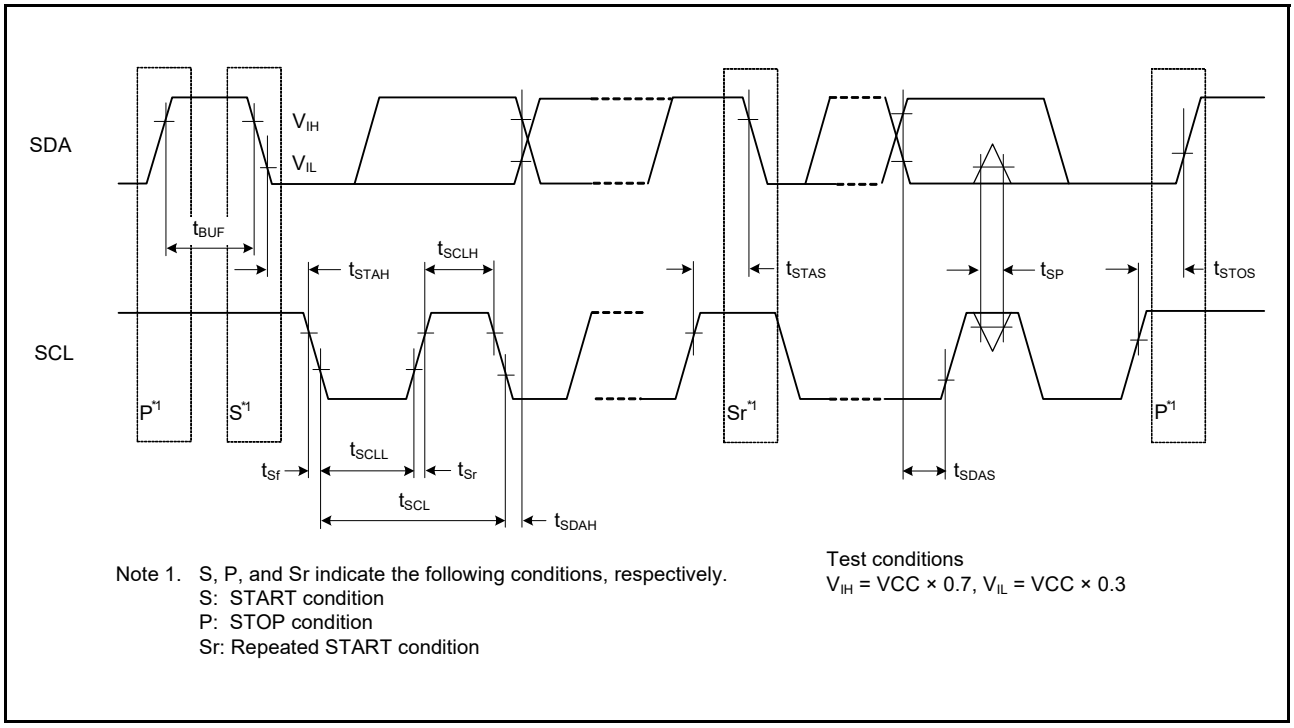


Figure 51.47 RIIC Bus Interface Input/Output Timing and Simple I²C Bus Interface Input/Output Timing

Downloaded from Arrow.com.

51.3.5.7 Timing of RSPI

Table 51.40 Timing of RSPI

Conditions: 1.8 V ≤ VCC = VCC_USB = AVCC0 = VCC_RF = AVCC_RF ≤ 3.6 V, VSS = AVSS0 = VSS_USB = VSS_RF = 0 V, T_a = –40 to +85°C, C = 30 pF, when high-drive output is selected by the drive capacity control register

Item				Symbol	Min.	Max.	Unit	Test Conditions
RSPI	RSPCK clock cycle	Master		t_{SPcyc}	2	4096	t_{Pcyc}*1	Figure 51.48
		Slave			8	4096		
	RSPCK clock high pulse width	Master		t_{SPCKWH}	$(t_{SPcyc} - t_{SPCKr} - t_{SPCKf})/2 - 3$	—	ns	
		Slave			$(t_{SPcyc} - t_{SPCKr} - t_{SPCKf})/2$	—		
	RSPCK clock low pulse width	Master		t_{SPCKWL}	$(t_{SPcyc} - t_{SPCKr} - t_{SPCKf})/2 - 3$	—	ns	
		Slave			$(t_{SPcyc} - t_{SPCKr} - t_{SPCKf})/2$	—		
	RSPCK clock rise/fall time	Output	2.7 V or above	t_{SPCKr}, t_{SPCKf}	—	10	ns	
			1.8 V or above		—	15		
		Input			—	1	µs	
	Data input setup time	Master	2.7 V or above	t_{SU}	10	—	ns	Figure 51.49 to Figure 51.52
			1.8 V or above		30	—		
		Slave			$25 - t_{Pcyc}$	—		
	Data input hold time	Master	RSPCK set to a division ratio other than PCLKB divided by 2	t_H	t_{Pcyc}	—	ns	
			RSPCK set to PCLKB divided by 2	t_{HF}	0	—		
		Slave		t_H	$20 + 2 \times t_{Pcyc}$	—		
	SSL setup time	Master		t_{LEAD}	$–30 + N^{*2} \times t_{SPcyc}$	—	ns	
		Slave			2	—	t_{Pcyc}	
	SSL hold time	Master		t_{LAG}	$–30 + N^{*3} \times t_{SPcyc}$	—	ns	
		Slave			2	—	t_{Pcyc}	
	Data output delay time	Master	2.7 V or above	t_{OD}	—	14	ns	
			1.8 V or above		—	30		
		Slave	2.7 V or above		—	$3 \times t_{Pcyc} + 65$		
			1.8 V or above		—	$3 \times t_{Pcyc} + 105$		
	Data output hold time	Master		t_{OH}	0	—	ns	
		Slave			0	—		
	Successive transmission delay time	Master		t_{TD}	$t_{SPcyc} + 2 \times t_{Pcyc}$	$8 \times t_{SPcyc} + 2 \times t_{Pcyc}$	ns	
		Slave			$4 \times t_{Pcyc}$	—		
	MOSI and MISO rise/fall time	Output	2.7 V or above	t_{Dr}, t_{Df}	—	10	ns	
			1.8 V or above		—	15		
		Input			—	1	µs	
	SSL rise/fall time	Output	2.7 V or above	t_{SSLr}, t_{SSLf}	—	10	ns	
			1.8 V or above		—	15	ns	
		Input			—	1	µs	
	Slave access time		2.7 V or above	t_{SA}	—	6	t_{Pcyc}	Figure 51.51, Figure 51.52
			1.8 V or above		—	7		
	Slave output release time		2.7 V or above	t_{REL}	—	5	t_{Pcyc}	
			1.8 V or above		—	6		

Downloaded from Arrow.com.

Note 1. t_{Pcyc}: PCLK cycle
Note 2. N: An integer from 1 to 8 that can be set by the RSPI clock delay register (SPCKD)
Note 3. N: An integer from 1 to 8 that can be set by the RSPI slave select negation delay register (SSLND)

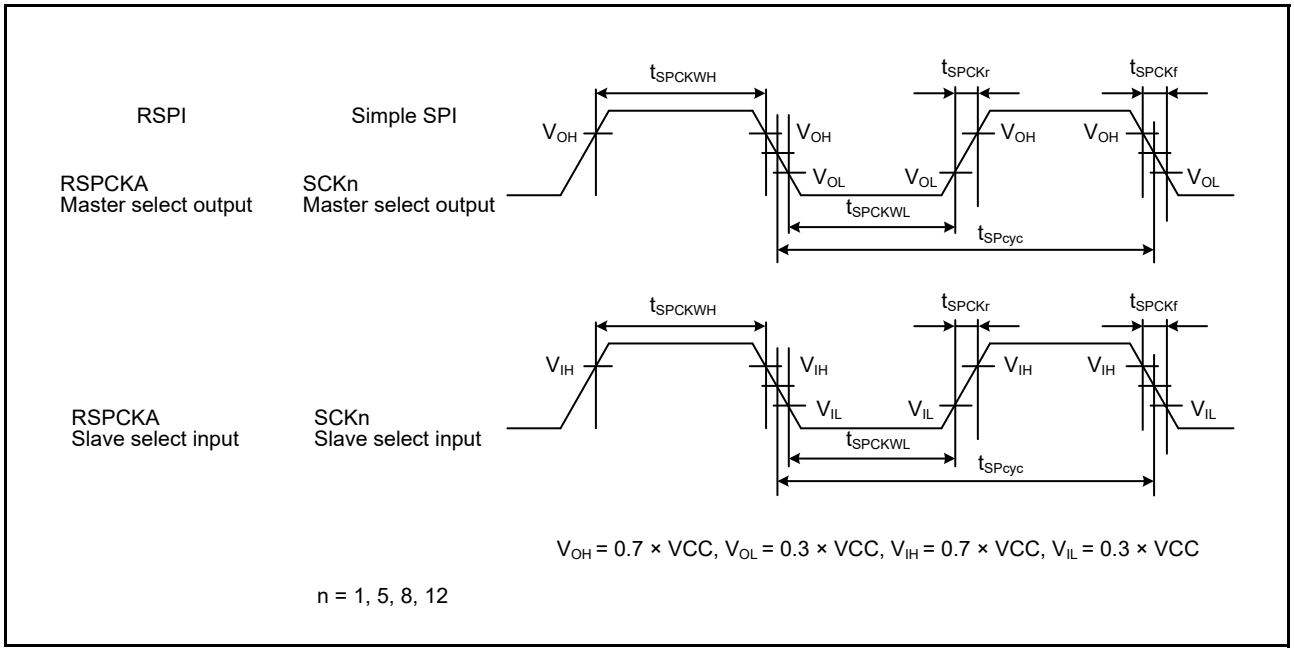


Figure 51.48 RSPI Clock Timing and Simple SPI Clock Timing

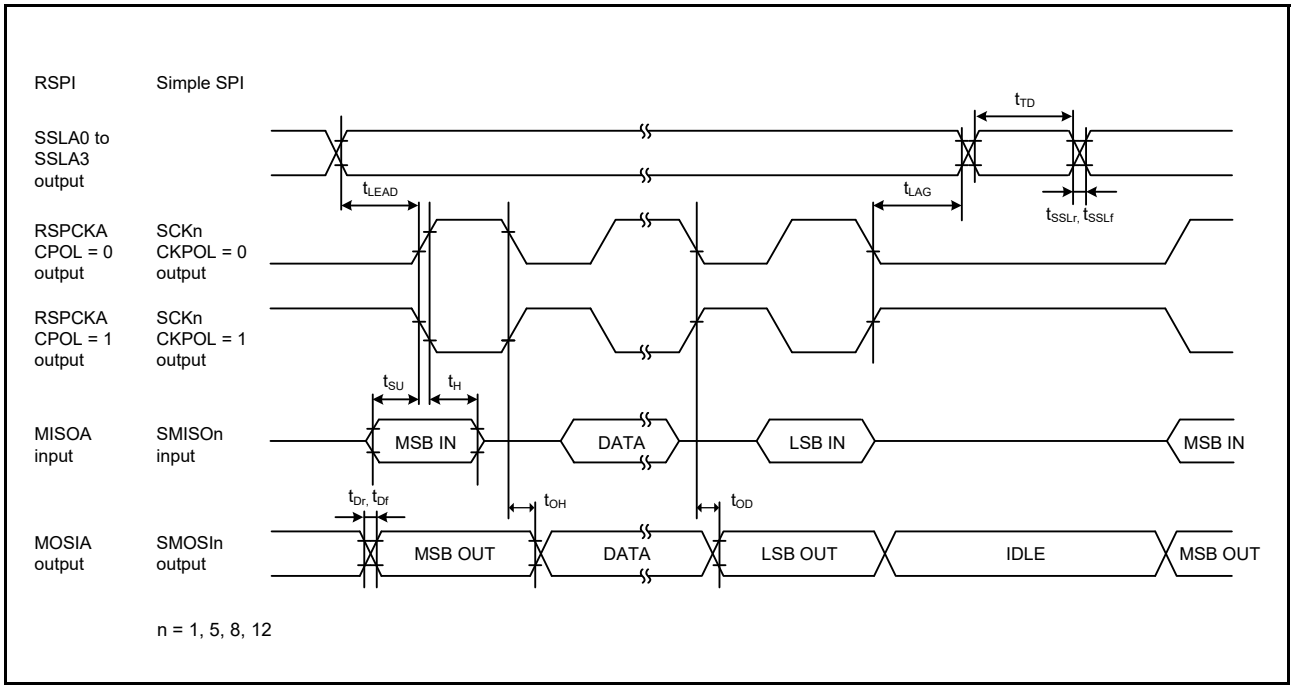


Figure 51.49 RSPI Timing (Master, CPHA = 0) and Simple SPI Clock Timing (Master, CKPH = 1)

Downloaded from Arrow.com.

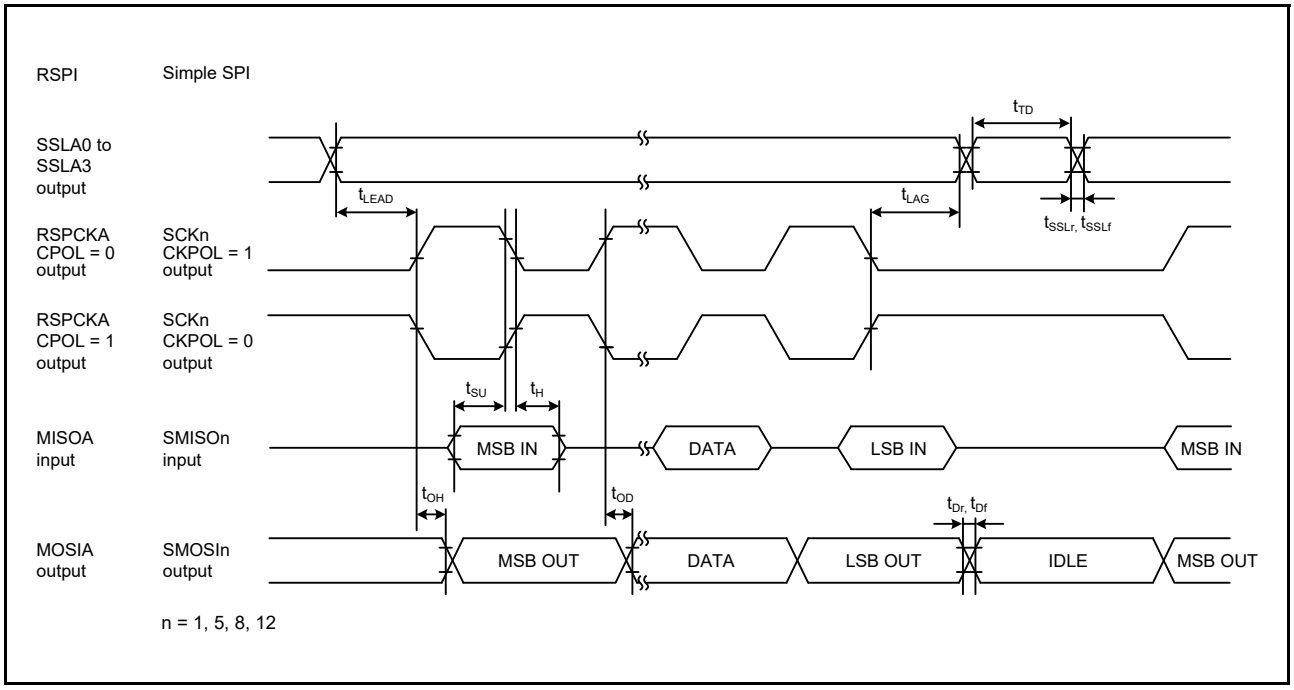


Figure 51.50 RSPI Timing (Master, CPHA = 1) and Simple SPI Clock Timing (Master, CKPH = 0)

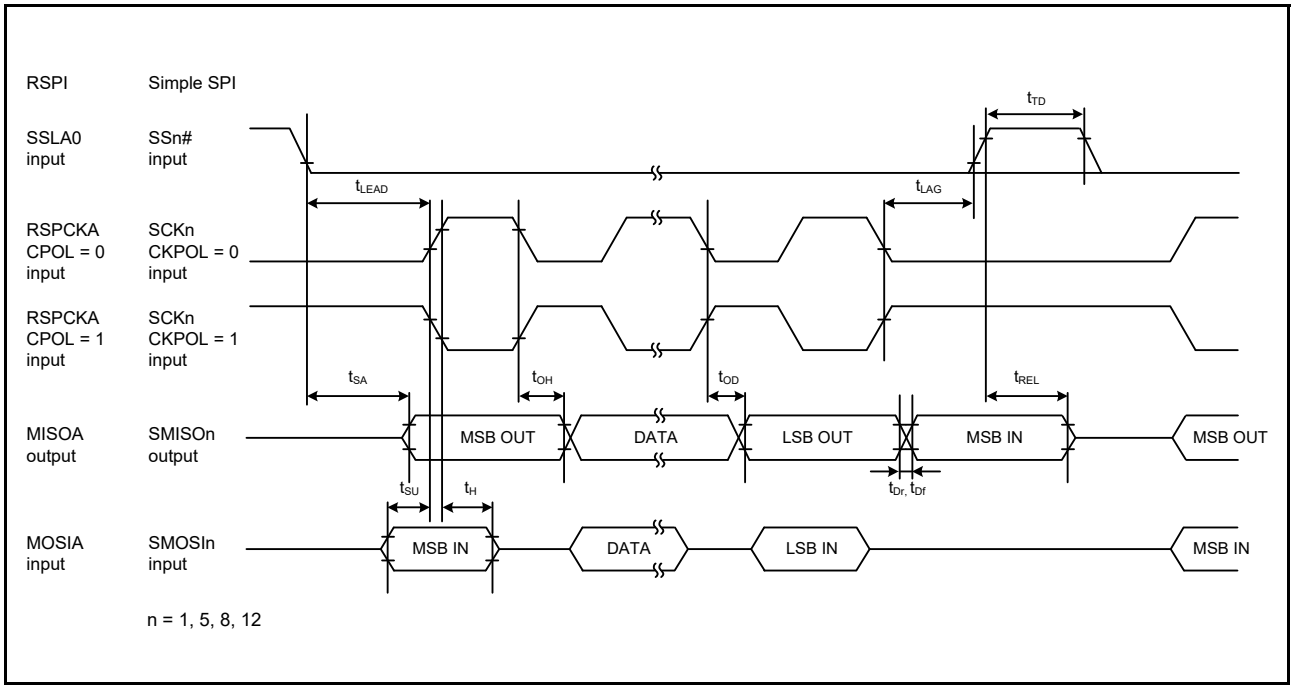


Figure 51.51 RSPI Timing (Slave, CPHA = 0) and Simple SPI Clock Timing (Slave, CKPH = 1)

Downloaded from Arrow.com.

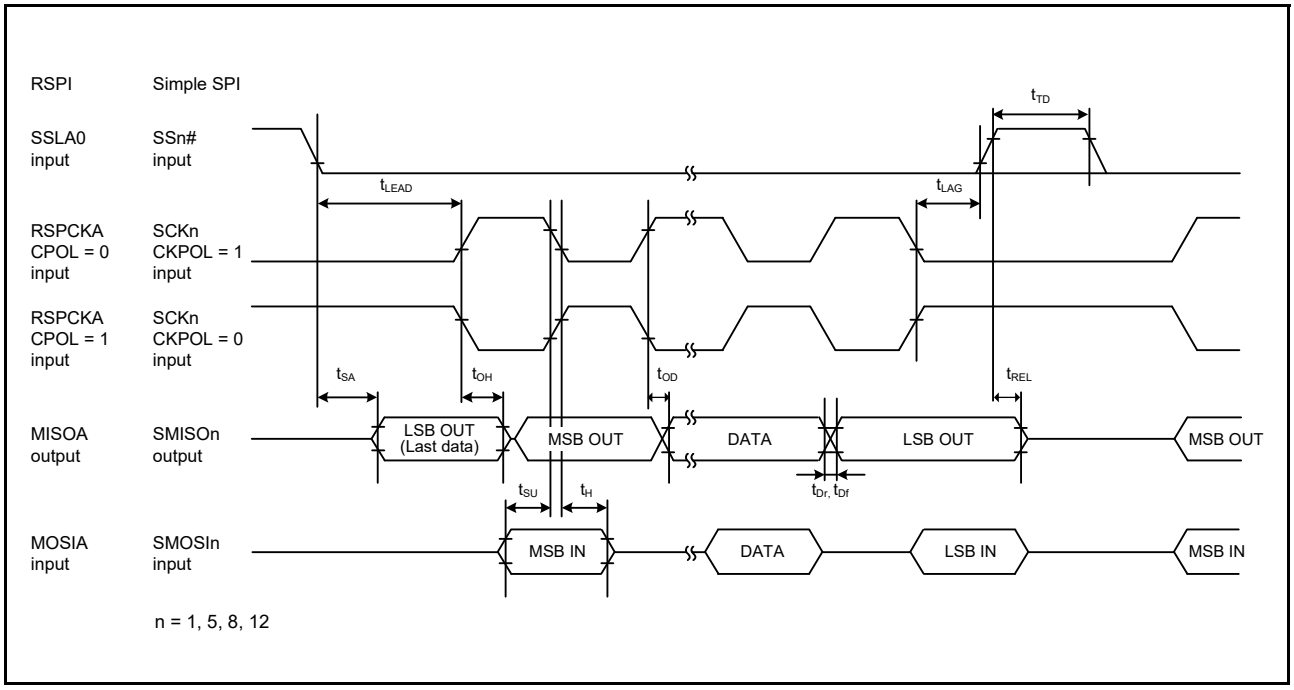


Figure 51.52 RSPI Timing (Slave, CPHA = 1) and Simple SPI Clock Timing (Slave, CKPH = 0)

Downloaded from Arrow.com.

51.3.5.8 Timing of SSI

Table 51.41 Timing of SSI

Conditions: 1.8 V ≤ VCC = VCC_USB = AVCC0 = VCC_RF = AVCC_RF ≤ 3.6 V, VSS = AVSS0 = VSS_USB = VSS_RF = 0 V, fPCLKB ≤ 32 MHz, T_a = –40 to +85°C

Item			Symbol	Min.	Max.	Unit	Test Conditions
SSI	AUDIO_MCLK input frequency	2.7 V or above	t_{AUDIO}	1	25	MHz	
		1.8 V or above		1	4		
	Output clock cycle		t_O	250	—	ns	Figure 51.53
	Input clock cycle		t_I	250	—	ns	
	Clock high level		t_{HC}	0.4	0.6	to, ti	
	Clock low level		t_{LC}	0.4	0.6	to, ti	
	Clock rise time		t_{RC}	—	20	ns	
	Data delay time	2.7 V or above	t_{DTR}	—	65	ns	Figure 51.54 Figure 51.55
		1.8 V or above		—	105		
	Setup time	2.7 V or above	t_{SR}	65	—	ns	
		1.8 V or above		90	—		
	Hold time		t_{HTR}	40	—	ns	
	WS changing edge SSIDATA output delay		t_{DTRW}	—	105	ns	Figure 51.56

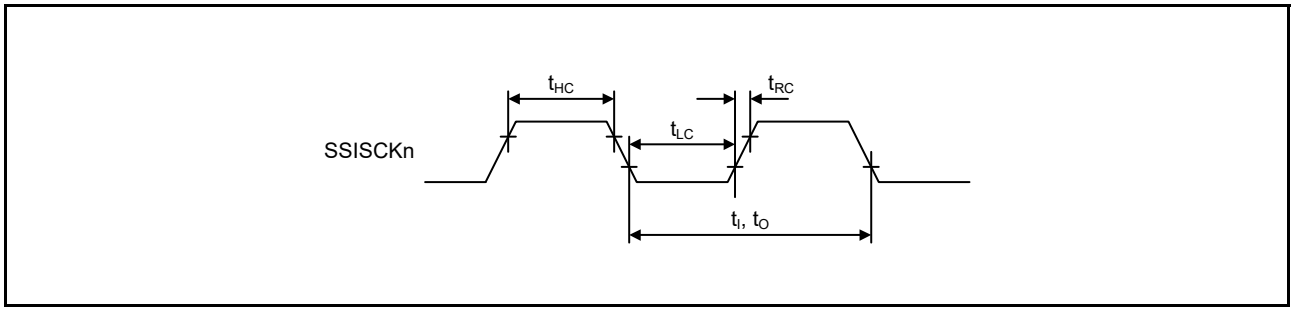


Figure 51.53 SSI Clock Input/Output Timing

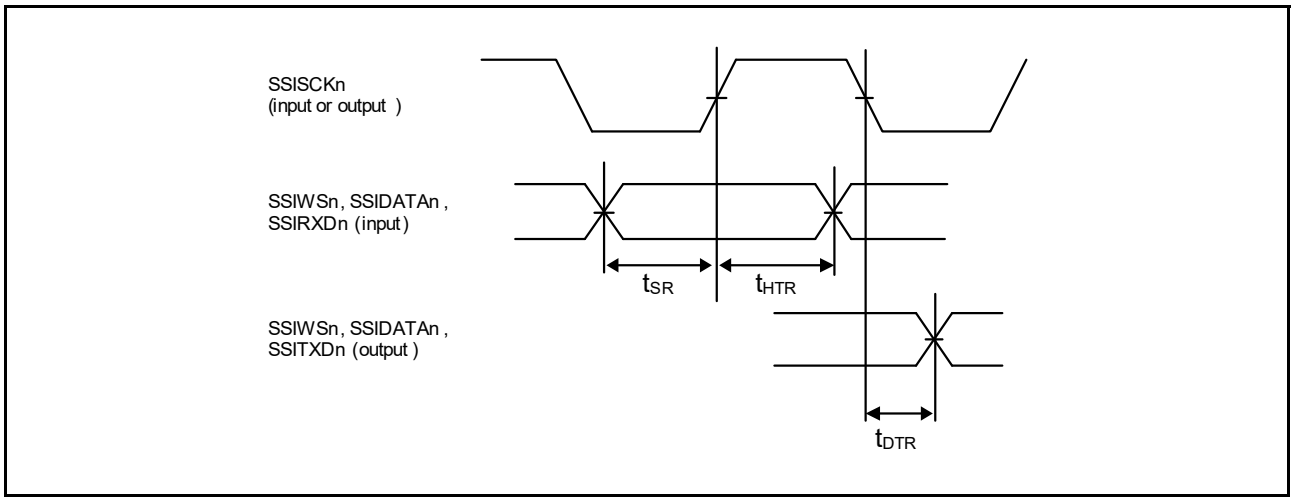


Figure 51.54 SSI Transmission/Reception Timing (SSICR.SCKP = 0)

Downloaded from Arrow.com.

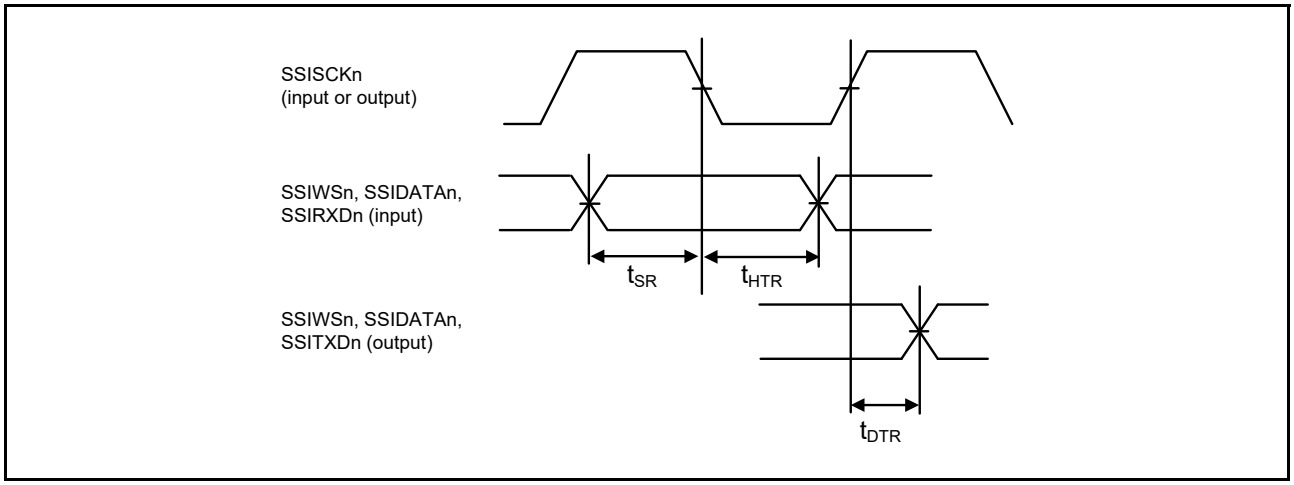
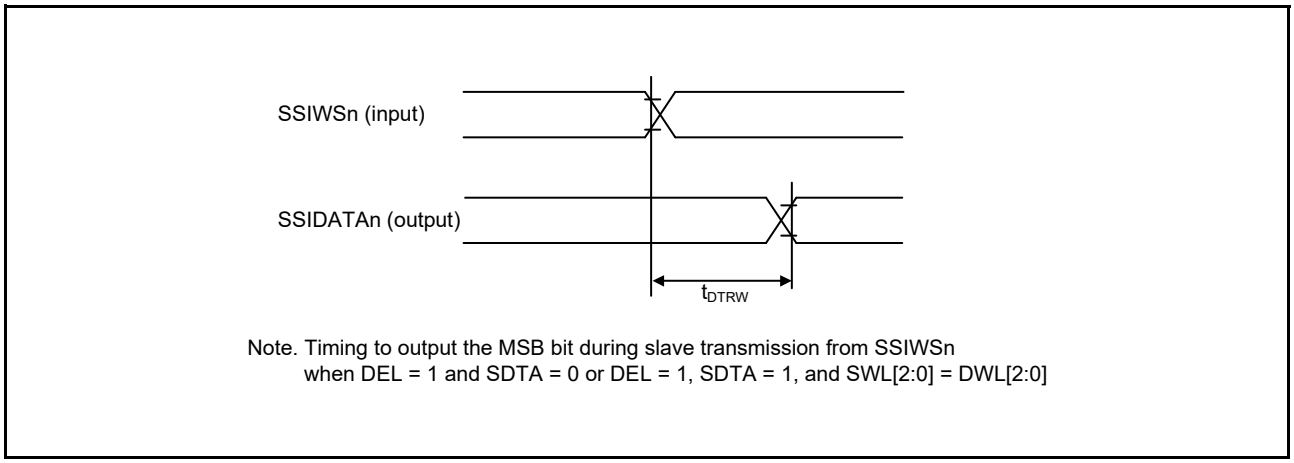


Figure 51.55 SSI Transmission/Reception Timing (SSICR.SCKP = 1)



Note. Timing to output the MSB bit during slave transmission from SSIWSn
when DEL = 1 and SDTA = 0 or DEL = 1, SDTA = 1, and SWL[2:0] = DWL[2:0]

Figure 51.56 SSIDATA Output Delay After SSIWSn Changing Edge

Downloaded from Arrow.com.

51.3.5.9 Timing of SDHI

Table 51.42 Timing of SDHI

Conditions: 2.7 V ≤ VCC = VCC_USB = AVCC0 = VCC_RF = AVCC_RF ≤ 3.6 V, VSS = AVSS0 = VSS_USB = VSS_RF = 0 V, fPCLKB ≤ 32 MHz, T_a = –40 to +85°C,
when high-drive output is selected by the drive capacity control register

Item		Symbol	Min.	Max.	Unit	Test Conditions
SDHI	SDHI_CLK pin output cycle time	$t_{PP(SD)}$	62.5	—	ns	Figure 51.57
	SDHI_CLK pin output high pulse width	$t_{WH(SD)}$	18.25	—	ns	
	SDHI_CLK pin output low pulse width	$t_{WL(SD)}$	18.25	—	ns	
	SDHI_CLK pin output rise time	$t_{TLH(SD)}$	—	10	ns	
	SDHI_CLK pin output fall time	$t_{THL(SD)}$	—	10	ns	
	Output data delay time (data transfer mode) for SDHI_CMD and SDHI_D0 to SDHI_D3 pins	$t_{ODLY(SD)}$	–18.25	18.25	ns	
	Input data setup time for SDHI_CMD and SDHI_D0 to SDHI_D3 pins	$t_{ISU(SD)}$	9.25	—	ns	
	Input data hold time for SDHI_CMD and SDHI_D0 to SDHI_D3 pins	$t_{IH(SD)}$	8.3	—	ns	

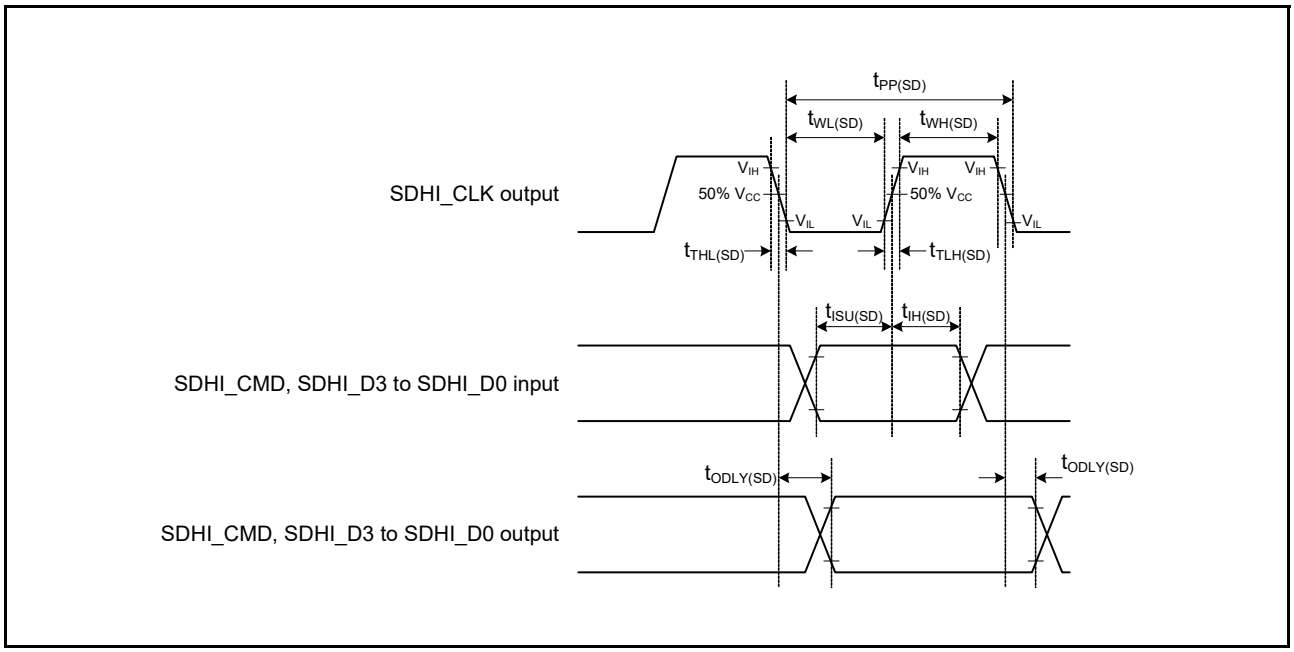


Figure 51.57 SD Host Interface Input/Output Signal Timing

Downloaded from Arrow.com.

51.3.5.10 Timing of A/D Converter Trigger

Table 51.43 Timing of A/D Converter Trigger

Conditions: 1.8 V ≤ VCC = VCC_USB = AVCC0 = VCC_RF = AVCC_RF ≤ 3.6 V, VSS = AVSS0 = VSS_USB = VSS_RF = 0 V,
T_a = –40 to +85°C

Item		Symbol	Min.	Max.	Unit *1	Test Conditions
A/D converter	Trigger input pulse width	t_{TRGW}	1.5	—	t_{Pcyc}	Figure 51.58

Note 1. t_{Pcyc}: PCLK cycle

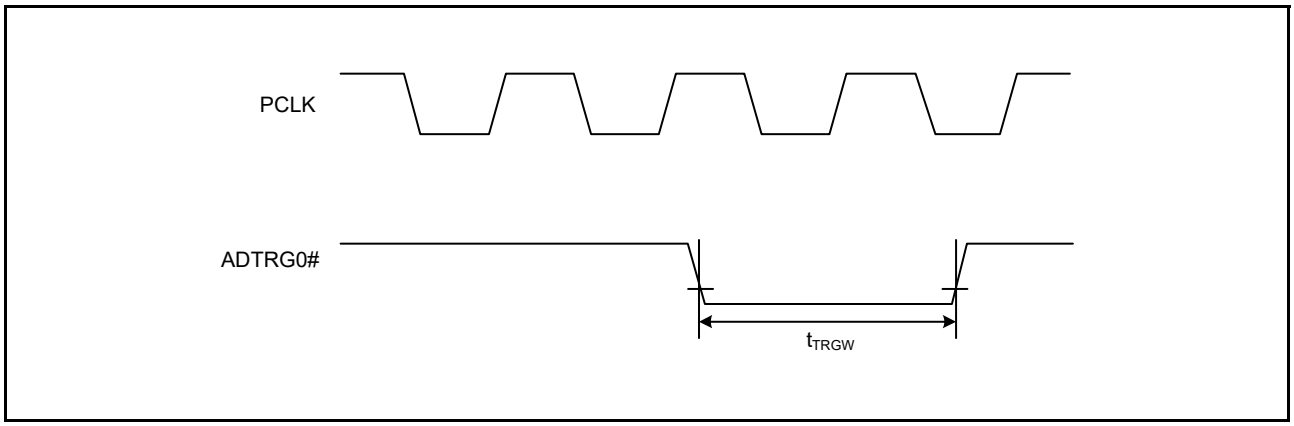


Figure 51.58 A/D Converter External Trigger Input Timing

51.3.5.11 Timing of CAC

Table 51.44 Timing of CAC

Conditions: 1.8 V ≤ VCC = VCC_USB = AVCC0 = VCC_RF = AVCC_RF ≤ 3.6 V, VSS = AVSS0 = VSS_USB = VSS_RF = 0 V,
T_a = –40 to +85°C

Item			Symbol	Min.	Max.	Unit *1	Test Conditions
CAC	CACREF input pulse width	$t_{Pcyc} \le t_{cac}$ *2	t_{CACREF}	4.5 t_{cac} + 3 t_{Pcyc}	—	ns	
		$t_{Pcyc} > t_{cac}$ *2		5 t_{cac} + 6.5 t_{Pcyc}			

Note 1. t_{Pcyc}: PCLK cycle
Note 2. t_{cac}: CAC count clock source cycle

Downloaded from Arrow.com.

51.3.5.12 Timing of CLKOUT

Table 51.45 Timing of CLKOUT

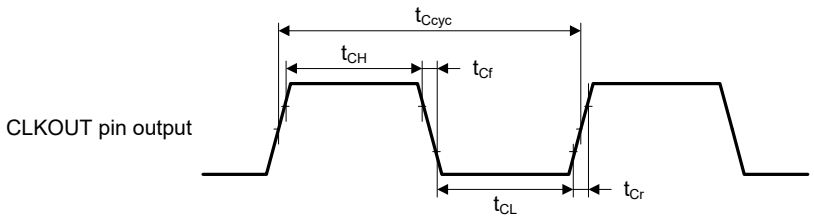
Conditions: 1.8 V ≤ VCC = VCC_USB = AVCC0 = VCC_RF = AVCC_RF ≤ 3.6 V, VSS = AVSS0 = VSS_USB = VSS_RF = 0 V, T_a = –40 to +85°C

Item			Symbol	Min.	Max.	Unit *1	Test Conditions
CLKOUT	CLKOUT pin output cycle*3	VCC = 2.7 V or above	t_{Ccyc}	62.5	—	ns	Figure 51.59
		VCC = 1.8 V or above		125			
	CLKOUT pin high pulse width*2	VCC = 2.7 V or above	t_{CH}	15	—	ns	
		VCC = 1.8 V or above		30			
	CLKOUT pin low pulse width*2	VCC = 2.7 V or above	t_{CL}	15	—	ns	
		VCC = 1.8 V or above		30			
	CLKOUT pin output rise time	VCC = 2.7 V or above	t_{Cr}	—	12	ns	
		VCC = 1.8 V or above			25		
	CLKOUT pin output fall time	VCC = 2.7 V or above	t_{Cf}	—	12	ns	
		VCC = 1.8 V or above			25		

Note 1. t_{Pcyc}: PCLK cycle

Note 2. When the LOCO is selected as the clock output source (the CKOCR.CKOSEL[2:0] bits are 000b), set the clock output division ratio selection to divided by 2 (the CKOCR.CKODIV[2:0] bits are 001b).

Note 3. When the EXTAL external clock input or an oscillator is used with divided by 1 (the CKOCR.CKOSEL[2:0] bits are 010b and the CKOCR.CKODIV[2:0] bits are 000b) to output from CLKOUT, the above should be satisfied with an input duty cycle of 45 to 55%.



Test conditions: V_{OH} = VCC × 0.7, V_{OL} = VCC × 0.3, I_{OH} = -1.0 mA, I_{OL} = 1.0 mA, C = 30 pF

Figure 51.59 CLKOUT Output Timing

Downloaded from Arrow.com.

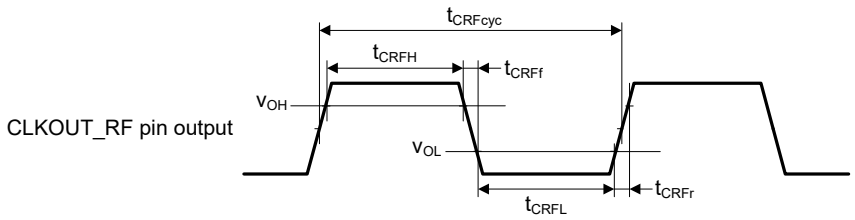
51.3.5.13 Timing of CLKOUT_RF

Table 51.46 Timing of CLKOUT_RF

Conditions: 1.8 V ≤ VCC = VCC_USB = AVCC0 = VCC_RF = AVCC_RF ≤ 3.6 V, VSS = AVSS0 = VSS_USB = VSS_RF = 0 V,
T_a = –40 to +85°C

Item		Symbol	Min.	Max.	Unit *1	Test Conditions
CLKOUT_RF *2	CLKOUT_RF pin output cycle	t_{CRFcyc}	250	—	ns	Figure 51.60
	CLKOUT_RF pin high pulse width	t_{CRFH}	100	—	ns	
	CLKOUT_RF pin low pulse width	t_{CRFL}	100	—	ns	
	CLKOUT_RF pin output rise time	t_{CRFr}	—	5	ns	
	CLKOUT_RF pin output fall time	t_{CRFf}	—	5	ns	

Note 1. t_{Pcyc}: PCLK cycle
Note 2. The voltage for VCC_RF when CLKOUT_RF pin is to be used is between 3.0 V and 3.6 V.



Test conditions: V_{OH} = VCC_RF × 0.8 , V_{OL} = VCC_RF × 0.2
Note. The voltage for VCC_RF pin when CLKOUT_RF is to be used is between 3.0 V and 3.6 V.

Figure 51.60 CLKOUT_RF Output Timing

Downloaded from Arrow.com.

51.4 USB Characteristics

Table 51.47 USB Characteristics (USB0_DP and USB0_DM Pin Characteristics)

Conditions: 3.0 V ≤ VCC = VCC_USB = AVCC = VCC_RF = AVCC_RF < 3.6 V, VSS = AVSS0 = VSS_USB = VSS_RF = 0 V, T_a = –40 to +85°C

Item			Symbol	Min.	Max.	Unit	Test Conditions
Input characteristics	Input high level voltage		V_{IH}	2.0	—	V	
	Input low level voltage		V_{IL}	—	0.8	V	
	Differential input sensitivity		V_{DI}	0.2	—	V	| USB0_DP – USB0_DM |
	Differential common mode range		V_{CM}	0.8	2.5	V	
Output characteristics	Output high level voltage		V_{OH}	2.8	VCC_USB	V	I_{OH} = –200 µA
	Output low level voltage		V_{OL}	0.0	0.3	V	I_{OL} = 2 mA
	Cross-over voltage		V_{CRS}	1.3	2.0	V	Figure 51.61, Figure 51.62
	Rise time	FS	t_r	4	20	ns	
		LS		75	300		
	Fall time	FS	t_f	4	20	ns	
		LS		75	300		
	Rise/fall time ratio	FS	t_r/t_f	90	111.11	%	t_r/t_f
		LS		80	125		
	Output resistance		Z_{DRV}	28	44	Ω	(Adjusting the resistance by external elements is not necessary.)
VBUS characteristics	VBUS input voltage		V_{IH}	VCC × 0.8	—	V	
			V_{IL}	—	VCC × 0.2	V	
Pull-up, pull-down	Pull-down resistor		R_{PD}	14.25	24.80	kΩ	
	Pull-up resistor		R_{PUI}	0.9	1.575	kΩ	During idle state
			R_{PUA}	1.425	3.09	kΩ	During reception
Battery Charging Specification Ver 1.2	D+ sink current		I_{DP_SINK}	25	175	µA	
	D- sink current		I_{DM_SINK}	25	175	µA	
	DCD source current		I_{DP_SRC}	7	13	µA	
	Data detection voltage		V_{DAT_REF}	0.25	0.4	V	
	D+ source current		V_{DP_SRC}	0.5	0.7	V	Output current = 250 µA
	D- source current		V_{DM_SRC}	0.5	0.7	V	Output current = 250 µA

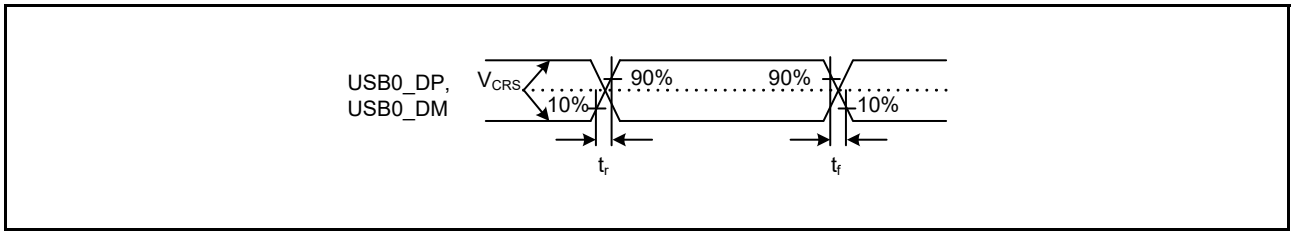


Figure 51.61 USB0_DP and USB0_DM Output Timing

Downloaded from Arrow.com.

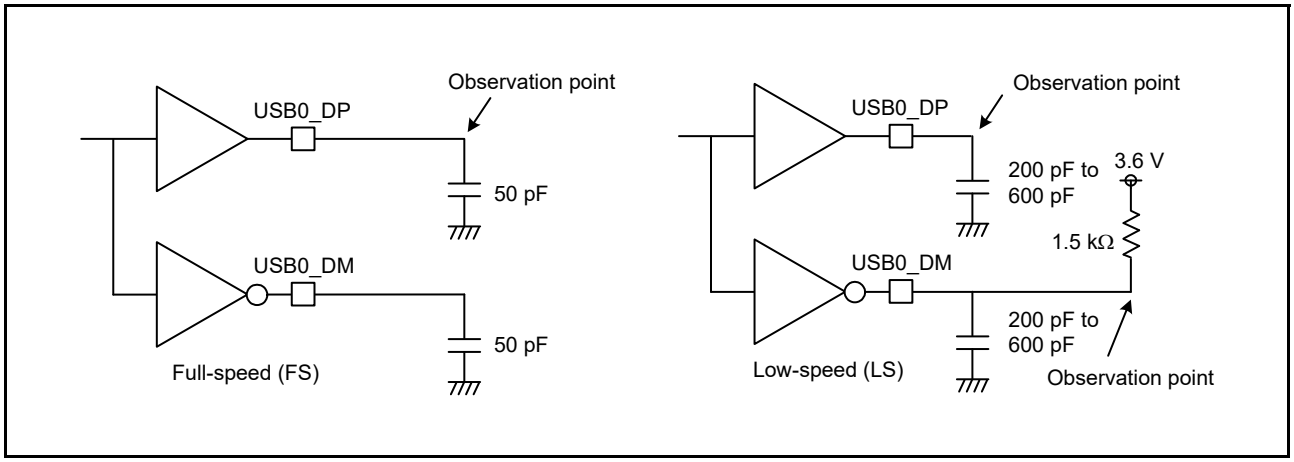


Figure 51.62 Test Circuit

Downloaded from Arrow.com.

51.5 A/D Conversion Characteristics

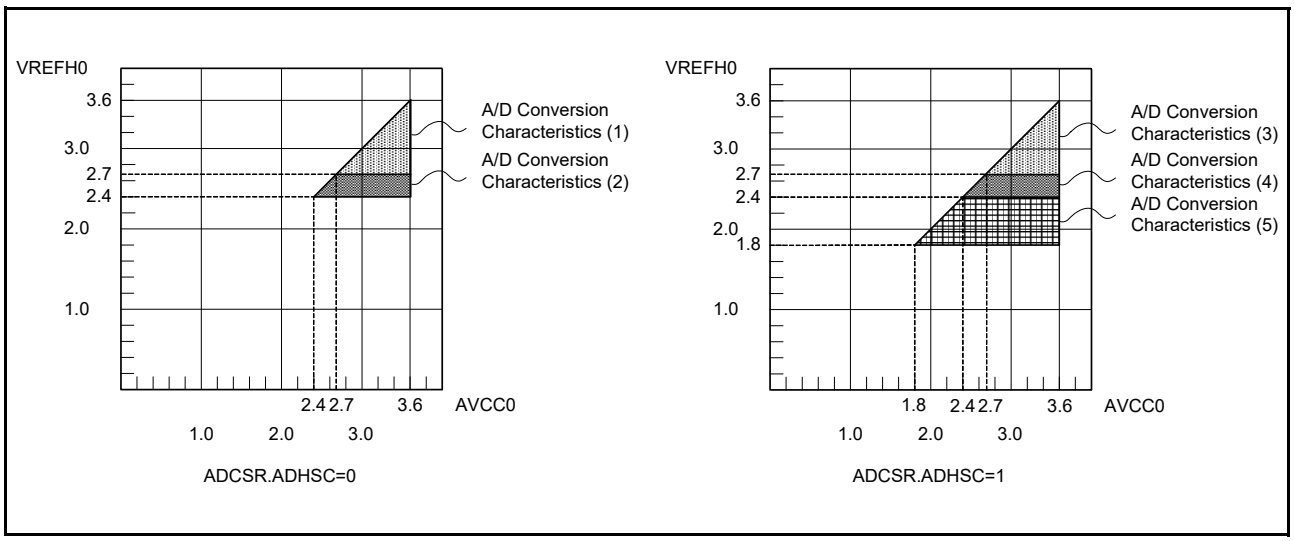


Figure 51.63 VREFH0 Voltage Range vs. AVCC0

Table 51.48 A/D Conversion Characteristics (1)

Conditions: 2.7 V ≤ VCC = VCC_USB = AVCC0 = VCC_RF = AVCC_RF ≤ 3.6 V, 2.7 V ≤ VREFH0 ≤ AVCC0, reference voltage = VREFH0 selected, VSS = AVSS0 = VREFL0 = VSS_USB = VSS_RF = 0 V, T_a = –40 to +85°C

Item		Min.	Typ.	Max.	Unit	Test Conditions
Frequency		1	—	54	MHz	
Resolution		—	—	12	Bit	
Conversion time*1 (Operation at PCLKD = 54 MHz)	Permissible signal source impedance (Max.) = 0.3 kΩ	0.83	—	—	μs	High-precision channel The ADCSR.ADHSC bit is 0 The ADSSTRn register is 0Dh
		1.33	—	—		Normal-precision channel The ADCSR.ADHSC bit is 0 The ADSSTRn register is 28h
Analog input capacitance	Cs	—	—	15	pF	Pin capacitance included Figure 51.64
Analog input resistance	Rs	—	—	2.5	kΩ	Figure 51.64
Analog input voltage range	Ain	0	—	VREFH0	V	
Offset error		—	±0.5	±4.5	LSB	High-precision channel
				±6.0	LSB	Other than above
Full-scale error		—	±0.75	±4.5	LSB	High-precision channel
				±6.0	LSB	Other than above
Quantization error		—	±0.5	—	LSB	
Absolute accuracy		—	±1.25	±5.0	LSB	High-precision channel
				±8.0	LSB	Other than above
DNL differential non-linearity error		—	±1.0	—	LSB	
INL integral non-linearity error		—	±1.0	±3.0	LSB	

Note: The characteristics apply when no pin functions other than A/D converter input are used. Absolute accuracy includes quantization errors. Offset error, full-scale error, DNL differential non-linearity error, and INL integral non-linearity error do not include quantization errors.

Note 1. The conversion time is the sum of the sampling time and the comparison time. As the test conditions, the number of sampling states is indicated.

Downloaded from Arrow.com.

Table 51.49 A/D Conversion Characteristics (2)

Conditions: 2.4 V ≤ VCC = VCC_USB = AVCC0 = VCC_RF = AVCC_RF ≤ 3.6 V, 2.4 V ≤ VREFH0 ≤ AVCC0, reference voltage = VREFH0 selected, VSS = AVSS0 = VREFL0 = VSS_USB = VSS_RF = 0 V, T_a = –40 to +85°C

Item		Min.	Typ.	Max.	Unit	Test Conditions
Frequency		1	—	32	MHz	
Resolution		—	—	12	Bit	
Conversion time*1 (Operation at PCLKD = 32 MHz)	Permissible signal source impedance (Max.) = 1.3 kΩ	1.41	—	—	μs	High-precision channel The ADCSR.ADHSC bit is 0 The ADSSTRn register is 0Dh
		2.25	—	—		Normal-precision channel The ADCSR.ADHSC bit is 0 The ADSSTRn register is 28h
Analog input capacitance	Cs	—	—	15	pF	Pin capacitance included Figure 51.64
Analog input resistance	Rs	—	—	2.5	kΩ	Figure 51.64
Offset error		—	±0.5	±4.5	LSB	
Full-scale error		—	±0.75	±4.5	LSB	
Quantization error		—	±0.5	—	LSB	
Absolute accuracy		—	±1.25	±5.0	LSB	High-precision channel
				±8.0	LSB	Other than above
DNL differential non-linearity error		—	±1.0	—	LSB	
INL integral non-linearity error		—	±1.0	±4.5	LSB	

Note: The characteristics apply when no pin functions other than A/D converter input are used. Absolute accuracy includes quantization errors. Offset error, full-scale error, DNL differential non-linearity error, and INL integral non-linearity error do not include quantization errors.

Note 1. The conversion time is the sum of the sampling time and the comparison time. As the test conditions, the number of sampling states is indicated.

Downloaded from Arrow.com.

Table 51.50 A/D Conversion Characteristics (3)

Conditions: 2.7 V ≤ VCC = VCC_USB = AVCC0 = VCC_RF = AVCC_RF ≤ 3.6 V, 2.7 V ≤ VREFH0 ≤ AVCC0, reference voltage = VREFH0 selected, VSS = AVSS0 = VREFL0 = VSS_USB = VSS_RF = 0 V, T_a = –40 to +85°C

Item		Min.	Typ.	Max.	Unit	Test Conditions
Frequency		1	—	27	MHz	
Resolution		—	—	12	Bit	
Conversion time*1 (Operation at PCLKD = 27 MHz)	Permissible signal source impedance (Max.) = 1.1 kΩ	2	—	—	μs	High-precision channel The ADCSR.ADHSC bit is 1 The ADSSTRn.SST[7:0] bits are 0Dh
		3	—	—		Normal-precision channel The ADCSR.ADHSC bit is 1 The ADSSTRn.SST[7:0] bits are 28h
Analog input capacitance	Cs	—	—	15	pF	Pin capacitance included Figure 51.64
Analog input resistance	Rs	—	—	2.5	kΩ	Figure 51.64
Offset error		—	±0.5	±4.5	LSB	
Full-scale error		—	±0.75	±4.5	LSB	
Quantization error		—	±0.5	—	LSB	
Absolute accuracy		—	±1.25	±5.0	LSB	High-precision channel
				±8.0	LSB	Other than above
DNL differential non-linearity error		—	±1.0	—	LSB	
INL integral non-linearity error		—	±1.0	±3.0	LSB	

Note: The characteristics apply when no pin functions other than A/D converter input are used. Absolute accuracy includes quantization errors. Offset error, full-scale error, DNL differential non-linearity error, and INL integral non-linearity error do not include quantization errors.

Note 1. The conversion time is the sum of the sampling time and the comparison time. As the test conditions, the number of sampling states is indicated.

Downloaded from Arrow.com.

Table 51.51 A/D Conversion Characteristics (4)

Conditions: 2.4 V ≤ VCC = VCC_USB = AVCC0 = VCC_RF = AVCC_RF ≤ 3.6 V, 2.4 V ≤ VREFH0 ≤ AVCC0,
VSS = AVSS0 = VSS_USB = 0 V, reference voltage = VREFH0 selected, T_a = –40 to +85°C

Item		Min.	Typ.	Max.	Unit	Test Conditions
Frequency		1	—	16	MHz	
Resolution		—	—	12	Bit	
Conversion time*1 (Operation at PCLKD = 16 MHz)	Permissible signal source impedance (Max.) = 2.2 kΩ	3.38	—	—	μs	High-precision channel The ADCSR.ADHSC bit is 1 The ADSSTRn register is 0Dh
		5.06	—	—		Normal-precision channel The ADCSR.ADHSC bit is 1 The ADSSTRn register is 28h
Analog input capacitance	Cs	—	—	15	pF	Pin capacitance included Figure 51.64
Analog input resistance	Rs	—	—	2.5	kΩ	Figure 51.64
Offset error		—	±0.5	±4.5	LSB	
Full-scale error		—	±0.75	±4.5	LSB	
Quantization error		—	±0.5	—	LSB	
Absolute accuracy		—	±1.25	±5.0	LSB	High-precision channel
				±8.0	LSB	Other than above
DNL differential non-linearity error		—	±1.0	—	LSB	
INL integral non-linearity error		—	±1.0	±3.0	LSB	

Note: The characteristics apply when no pin functions other than A/D converter input are used. Absolute accuracy includes quantization errors. Offset error, full-scale error, DNL differential non-linearity error, and INL integral non-linearity error do not include quantization errors.

Note 1. The conversion time is the sum of the sampling time and the comparison time. As the test conditions, the number of sampling states is indicated.

Downloaded from Arrow.com.

Table 51.52 A/D Conversion Characteristics (5)

Conditions: 1.8 V ≤ VCC = VCC_USB = AVCC0 = VCC_RF = AVCC_RF ≤ 3.6 V, 1.8 V ≤ VREFH0 ≤ AVCC0,
VSS = AVSS0 = VSS_USB = 0 V, reference voltage = VREFH0 selected, T_a = –40 to +85°C

Item		Min.	Typ.	Max.	Unit	Test Conditions
Frequency		1	—	8	MHz	
Resolution		—	—	12	Bit	
Conversion time*1 (Operation at PCLKD = 8 MHz)	Permissible signal source impedance (Max.) = 5 kΩ	6.75	—	—	µs	High-precision channel The ADCSR.ADHSC bit is 1 The ADSSTRn register is 0Dh
		10.13	—	—		Normal-precision channel The ADCSR.ADHSC bit is 1 The ADSSTRn register is 28h
Analog input capacitance	Cs	—	—	15	pF	Pin capacitance included Figure 51.64
Analog input resistance	Rs	—	—	2.5	kΩ	Figure 51.64
Offset error		—	±1	±7.5	LSB	
Full-scale error		—	±1.5	±7.5	LSB	
Quantization error		—	±0.5	—	LSB	
Absolute accuracy		—	±3.0	±8.0	LSB	
DNL differential non-linearity error		—	±1.0	—	LSB	
INL integral non-linearity error		—	±1.25	±3.0	LSB	

Note: The characteristics apply when no pin functions other than A/D converter input are used. Absolute accuracy includes quantization errors. Offset error, full-scale error, DNL differential non-linearity error, and INL integral non-linearity error do not include quantization errors.

Note 1. The conversion time is the sum of the sampling time and the comparison time. As the test conditions, the number of sampling states is indicated.

Table 51.53 A/D Converter Channel Classification

Classification	Channel	Conditions	Remarks
High-precision channel	AN000 to AN007	AVCC0 = 1.8 to 3.6 V	Pins AN000 to AN007 cannot be used as digital outputs when the A/D converter is in use.
Normal-precision channel	AN016 to AN020, AN027		
Internal reference voltage input channel	Internal reference voltage	AVCC0 = 2.0 to 3.6 V	
Temperature sensor input channel	Temperature sensor output	AVCC0 = 2.0 to 3.6 V	

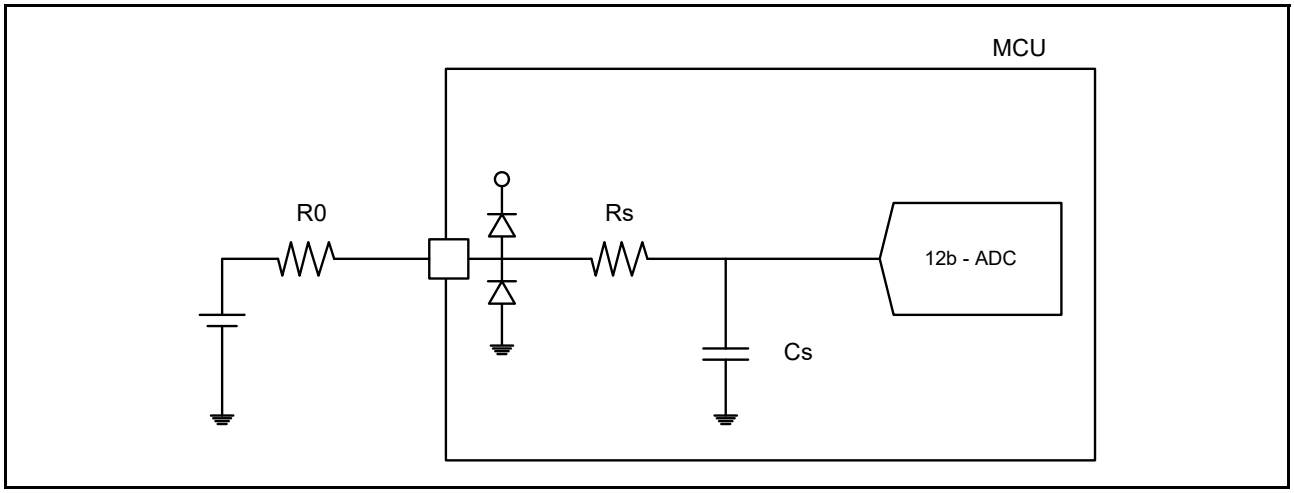


Figure 51.64 Equivalent Circuit

Downloaded from Arrow.com.

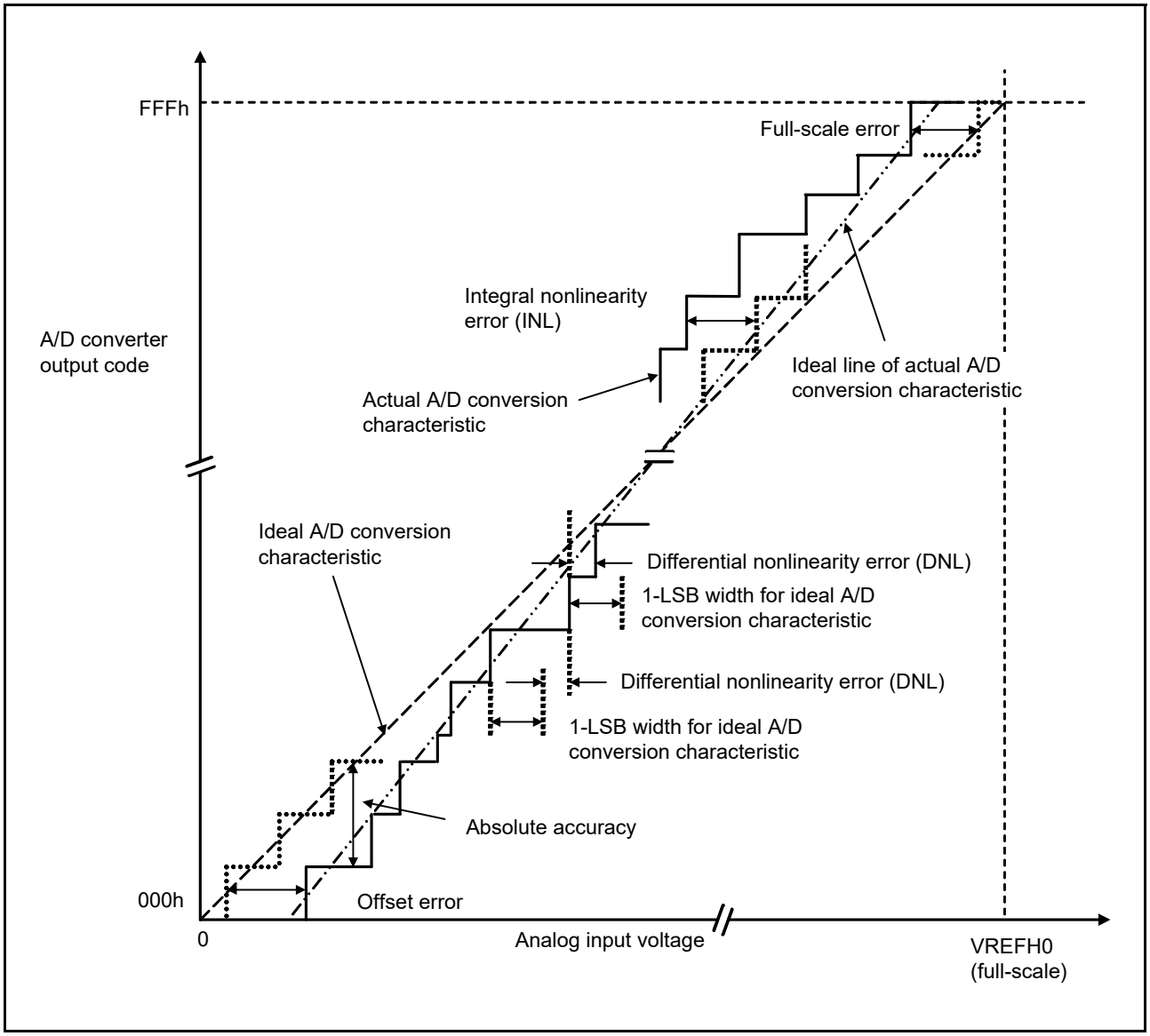


Figure 51.65 Illustration of A/D Converter Characteristic Terms

Absolute accuracy

Absolute accuracy is the difference between output code based on the theoretical A/D conversion characteristics and the actual A/D conversion result. When measuring absolute accuracy, the voltage at the midpoint of the width of analog input voltage (1-LSB width), that can meet the expectation of outputting an equal code based on the theoretical A/D conversion characteristics, is used as an analog input voltage. For example, if 12-bit resolution is used and if reference voltage (VREFH0 = 3.072 V), then 1-LSB width becomes 0.75 mV, and 0 mV, 0.75 mV, 1.5 mV, ... are used as analog input voltages.

If analog input voltage is 6 mV, absolute accuracy = ±5 LSB means that the actual A/D conversion result is in the range of 003h to 00Dh, although an output code, 008h, can be expected from the theoretical A/D conversion characteristics.

Integral non-linearity error (INL)

The integral non-linearity error is the maximum deviation between the ideal line when the measured offset and full-scale errors are zeroed, and the actual output code.

Downloaded from Arrow.com.

Differential non-linearity error (DNL)

The differential non-linearity error is the difference between 1-LSB width based on the ideal A/D conversion characteristics and the width of the actual output code.

Offset error

An offset error is the difference between a transition point of the ideal first output code and the actual first output code.

Full-scale error

A full-scale error is the difference between a transition point of the ideal last output code and the actual last output code.

Downloaded from Arrow.com.

51.6 D/A Conversion Characteristics

Table 51.54 D/A Conversion Characteristics (1)

Conditions: 1.8 V ≤ VCC = VCC_USB = AVCC0 = VCC_RF = AVCC_RF ≤ 3.6 V, VSS = AVSS0 = VSS_USB = VSS_RF = 0 V, T_a = –40 to +85°C, Reference voltage = AVCC0 or AVSS0 selected

Item	Min.	Typ.	Max.	Unit	Test Conditions
Resolution	—	—	12	Bit	
Resistive load	30	—	—	kΩ	
Capacitive load	—	—	50	pF	
Output voltage range	0.35	—	AVCC0 - 0.47	V	
DNL differential non-linearity error	—	±0.5	±2.0	LSB	
INL integral non-linearity error	—	±2.0	±8.0	LSB	
Offset error	—	—	±30	mV	
Full-scale error	—	—	±30	mV	
Output resistance	—	5	—	Ω	
Conversion time	—	—	30	µs	

Table 51.55 D/A Conversion Characteristics (2)

Conditions: 1.8 V ≤ VCC = VCC_USB = AVCC0 = VCC_RF = AVCC_RF ≤ 3.6 V, VSS = AVSS0 = VSS_USB = VSS_RF = 0 V, T_a = –40 to +85°C, Reference voltage = internal reference voltage selected

Item	Min.	Typ.	Max.	Unit	Test Conditions
Resolution	—	—	12	Bit	
Internal reference voltage (Vbgr)	1.36	1.43	1.50	V	
Resistive load	30	—	—	kΩ	
Capacitive load	—	—	50	pF	
Output voltage range	0.35	—	Vbgr	V	
DNL differential non-linearity error	—	±2.0	±16.0	LSB	
INL integral non-linearity error	—	±8.0	±16.0	LSB	
Offset error	—	—	30	mV	
Output resistance	—	5	—	Ω	
Conversion time	—	—	30	µs	

Downloaded from Arrow.com.

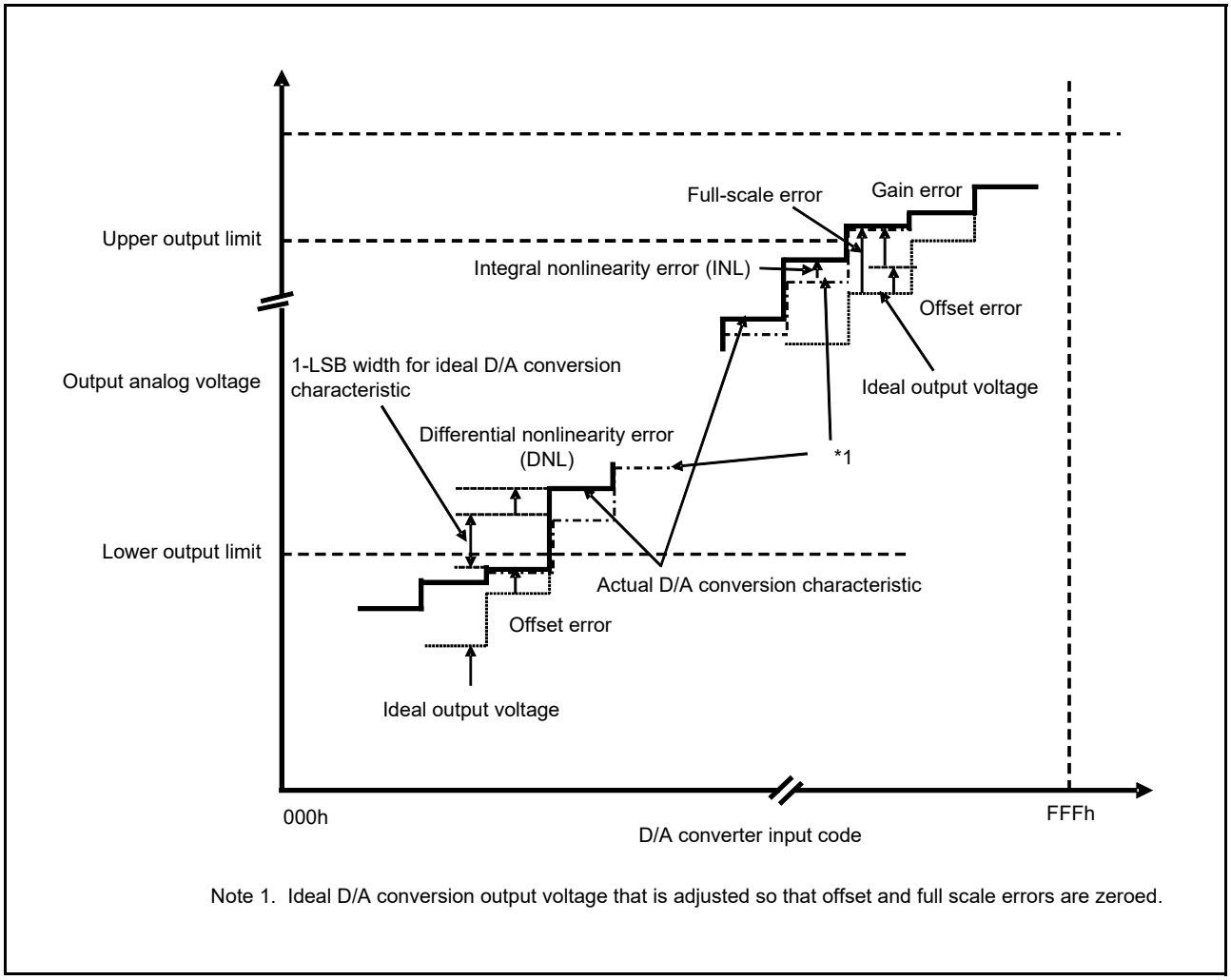


Figure 51.66 Illustration of D/A Converter Characteristic Terms

Integral non-linearity error (INL)

The integral non-linearity error is the maximum deviation between the ideal line when the measured offset and full-scale errors are zeroed, and the actual output code.

Differential non-linearity error (DNL)

The differential non-linearity error is the difference between 1-LSB width based on the ideal D/A conversion characteristics and the width of the actually output code.

Offset error

An offset error is the difference between a transition point of the ideal first output code and the actual first output code.

Full-scale error

A full-scale error is the difference between a transition point of the ideal last output code and the actual last output code.

Downloaded from Arrow.com.

51.7 Temperature Sensor Characteristics

Table 51.56 Temperature Sensor Characteristics

Conditions: 2.0 V ≤ VCC = VCC_USB = AVCC0 = VCC_RF = AVCC_RF ≤ 3.6 V, VSS = AVSS0 = VSS_USB = VSS_RF = 0 V, T_a = –40 to +85°C

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Relative accuracy	—	—	±1.5	—	°C	2.4 V or above
		—	±2.0	—		Below 2.4 V
Temperature slope	—	—	–3.65	—	mV/°C	
Output voltage (25°C)	—	—	1.05	—	V	VCC = 3.3 V
Temperature sensor start time	t_{START}	—	—	5	µs	
Sampling time	—	5	—	—	µs	

51.8 Comparator Characteristics

Table 51.57 Comparator Characteristics

Conditions: 1.8 V ≤ VCC = VCC_USB = AVCC0 = VCC_RF = AVCC_RF ≤ 3.6 V, VSS = AVSS0 = VSS_USB = VSS_RF = 0 V, T_a = –40 to +85°C

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
CVREFB2, CVREFB3 input reference voltage		VREF	0	—	VCC - 1.4	V	
CMPB2, CMPB3 input voltage		VI	–0.3	—	VCC + 0.3	V	
Offset	Comparator high-speed mode	—	—	—	50	mV	
	Comparator high-speed mode Window function enabled	—	—	—	60	mV	
	Comparator low-speed mode	—	—	—	40	mV	
Comparator output delay time	Comparator high-speed mode	Td	—	—	1.2	µs	VCC = 3 V, input slew rate ≥ 50 mV/µs
	Comparator high-speed mode Window function enabled	Tdw	—	—	2.0	µs	
	Comparator low-speed mode	Td	—	—	5.0	µs	
High-side reference voltage (comparator high-speed mode, window function enabled)		VRFH	—	0.76 VCC	—	V	
Low-side reference voltage (comparator high-speed mode, window function enabled)		VRFL	—	0.24 VCC	—	V	
Operation stabilization wait time		Tcmp	100	—	—	µs	

Downloaded from Arrow.com.

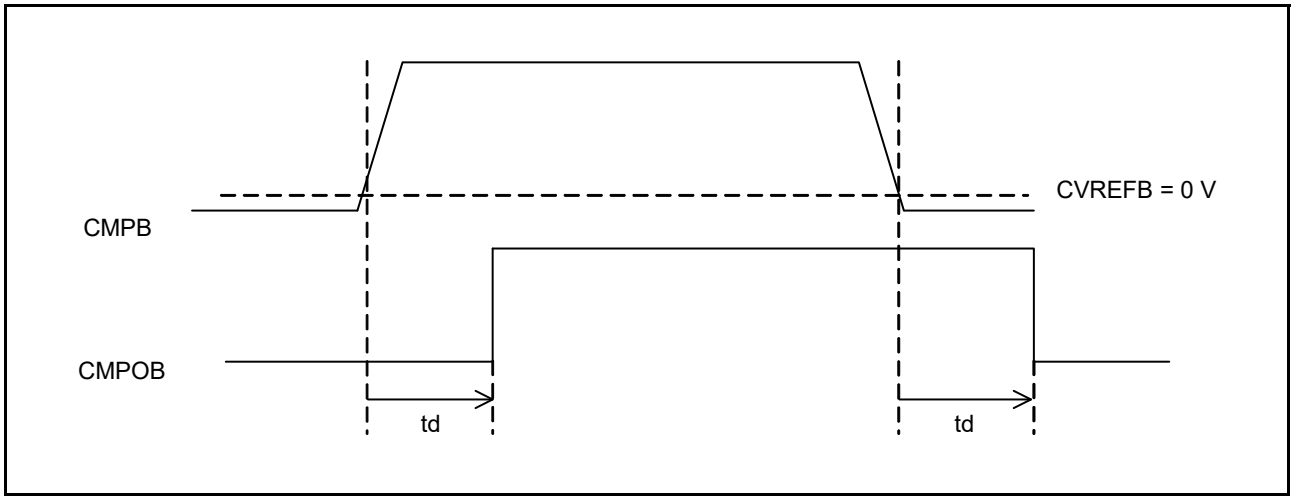


Figure 51.67 Comparator Output Delay Time in Comparator High-Speed Mode and Low-Speed Mode

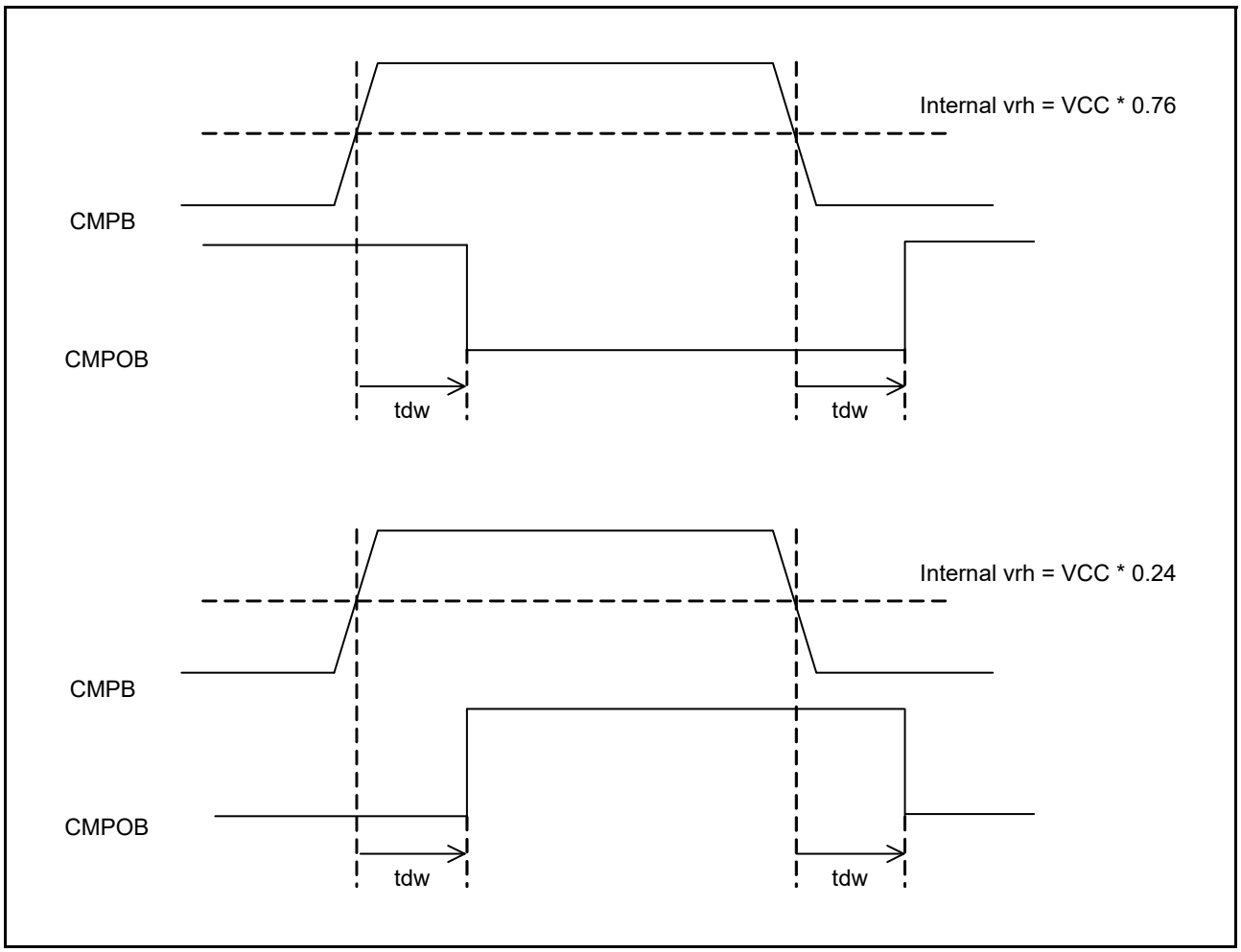


Figure 51.68 Comparator Output Delay Time in High-Speed Mode with Window Function Enabled

Downloaded from Arrow.com.

51.9 CTSU Characteristics

Table 51.58 CTSU Characteristics

Conditions: 1.8 V ≤ VCC = VCC_USB = AVCC0 = VCC_RF = AVCC_RF ≤ 3.6 V, VSS = AVSS0 = VSS_USB = VSS_RF = 0 V, T_a = –40 to +85°C

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
External capacitance connected to TSCAP pin	C_{tscap}	9	10	11	nF	
TS pin capacitive load	C_{base}	—	—	50	pF	
Permissible output high current	ΣI_{OH}	—	—	–24	mA	When the mutual capacitance method is applied

51.10 Characteristics of Power-On Reset Circuit and Voltage Detection Circuit

Table 51.59 Characteristics of Power-On Reset Circuit and Voltage Detection Circuit (1)

Conditions: 1.8 V ≤ VCC = VCC_USB = AVCC0 = VCC_RF = AVCC_RF ≤ 3.6 V, VSS = AVSS0 = VSS_USB = VSS_RF = 0 V, T_a = –40 to +85°C

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Voltage detection level	Power-on reset (POR)	V_{POR}	1.35	1.50	1.65	V	Figure 51.69, Figure 51.70
	Voltage detection circuit (LVD0)*1	V_{det0_1}	2.70	2.82	3.00	V	Figure 51.71 At falling edge VCC
		V_{det0_2}	2.37	2.51	2.67		
		V_{det0_3}	1.80	1.90	1.99		
	Voltage detection circuit (LVD1)*2	V_{det1_4}	2.99	3.10	3.29	V	Figure 51.72 At falling edge VCC
		V_{det1_5}	2.89	3.00	3.19		
		V_{det1_6}	2.79	2.90	3.09		
		V_{det1_7}	2.68	2.79	2.98		
		V_{det1_8}	2.57	2.68	2.87		
		V_{det1_9}	2.47	2.58	2.67		
		V_{det1_A}	2.37	2.48	2.57		
		V_{det1_B}	2.10	2.20	2.30		
		V_{det1_C}	1.86	1.96	2.06		
		V_{det1_D}	1.80	1.86	1.96		

Note: These characteristics apply when noise is not superimposed on the power supply.
Note 1. n in the symbol Vdet0_n denotes the value of the OFS1.VDSEL[1:0] bits.
Note 2. n in the symbol Vdet1_n denotes the value of the LVDLVLR.LVD1LVL[3:0] bits.

Downloaded from Arrow.com.

Table 51.60 Characteristics of Power-On Reset Circuit and Voltage Detection Circuit (2)

Conditions: 1.8 V ≤ VCC0 = VCC_USB = AVCC0 = VCC_RF = AVCC_RF ≤ 3.6 V, VSS = AVSS0 = VSS_USB = VSS_RF = 0 V, T_a = –40 to +85°C

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Wait time after power-on reset cancellation	At normal startup*1	t_{POR}	—	9.1	—	ms	Figure 51.70
	During fast startup time*2	t_{POR}	—	1.6	—		
Wait time after voltage monitoring 0 reset cancellation	Power-on voltage monitoring 0 reset disabled*1	t_{LVD0}	—	568	—	µs	Figure 51.71
	Power-on voltage monitoring 0 reset enabled*2		—	100	—		
Wait time after voltage monitoring 1 reset cancellation		t_{LVD1}	—	100	—	µs	Figure 51.72
Response delay time		t_{det}	—	—	350	µs	Figure 51.69
Minimum VCC down time*3		t_{VOFF}	350	—	—	µs	Figure 51.69, VCC = 1.0 V or above
Power-on reset enable time		$t_{W(POR)}$	1	—	—	ms	Figure 51.70, VCC = below 1.0 V
LVD operation stabilization time (after LVD is enabled)		$Td_{(E-A)}$	—	—	300	µs	Figure 51.72
Hysteresis width (power-on rest (POR))		V_{PORH}	—	110	—	mV	
Hysteresis width (voltage detection circuit: LVD1)		V_{LVH}	—	70	—	mV	When Vdet1_4 is selected
			—	60	—		When Vdet1_5 to Vdet1_9 is selected
			—	50	—		When Vdet1_A or Vdet1_B is selected
			—	40	—		When Vdet1_C or Vdet1_D is selected

Note: These characteristics apply when noise is not superimposed on the power supply.
Note 1. When OFS1.(LVDAS, FASTSTUP) = 11b.
Note 2. When OFS1.(LVDAS, FASTSTUP) ≠ 11b.
Note 3. The minimum VCC down time indicates the time when VCC is below the minimum value of voltage detection levels V_{POR}, V_{det0}, and V_{det1} for the POR/LVD.

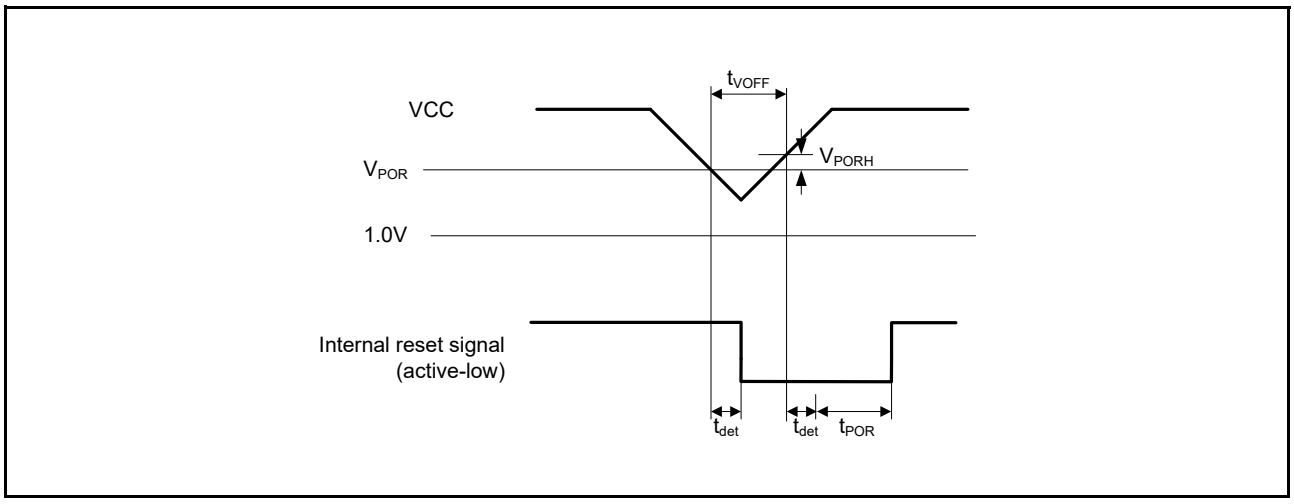
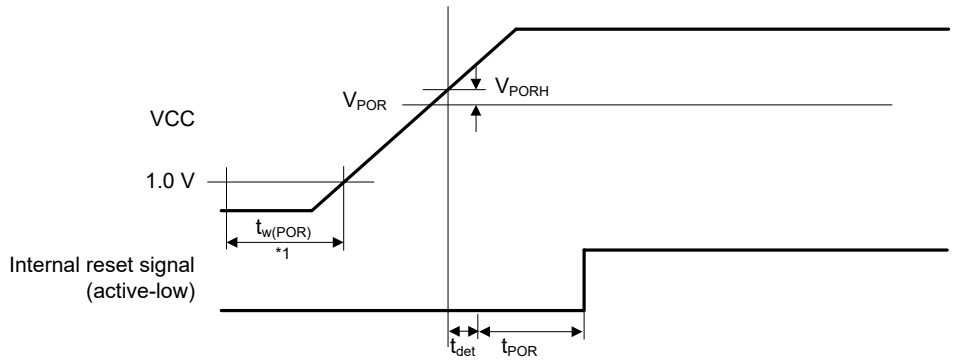


Figure 51.69 Voltage Detection Reset Timing

Downloaded from Arrow.com.



Note 1. $t_{w(POR)}$ is the time required for a power-on reset to be enabled while the external power VCC is being held below the valid voltage (1.0 V).
When turning the VCC on, maintain a voltage below 1.0V for at least 1.0ms.

Figure 51.70 Power-On Reset Timing

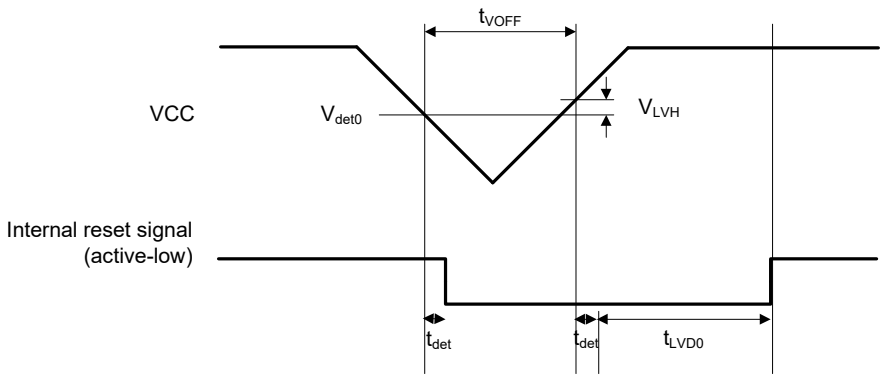


Figure 51.71 Voltage Detection Circuit Timing (Vdet0)

Downloaded from Arrow.com.

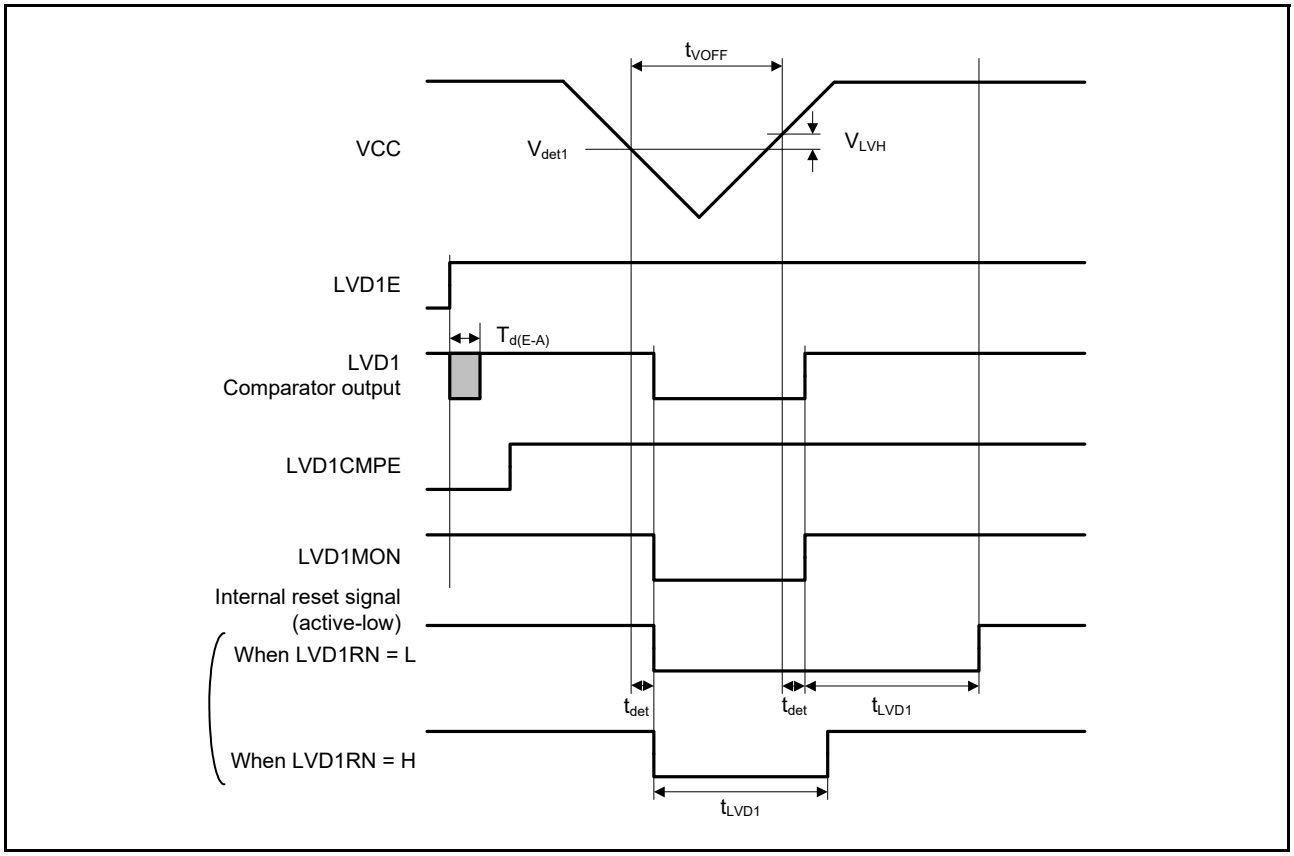


Figure 51.72 Voltage Detection Circuit Timing (V_{det1})

51.11 Oscillation Stop Detection Timing

Table 51.61 Oscillation Stop Detection Timing

Conditions: 1.8 V ≤ VCC = VCC_USB = AVCC0 = VCC_RF = AVCC_RF ≤ 3.6 V,
VSS = AVSS0 = VREFL0 = VSS_USB = VSS_RF = 0 V, T_a = –40 to +85°C

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Detection time	t_{dr}	—	—	1	ms	Figure 51.73

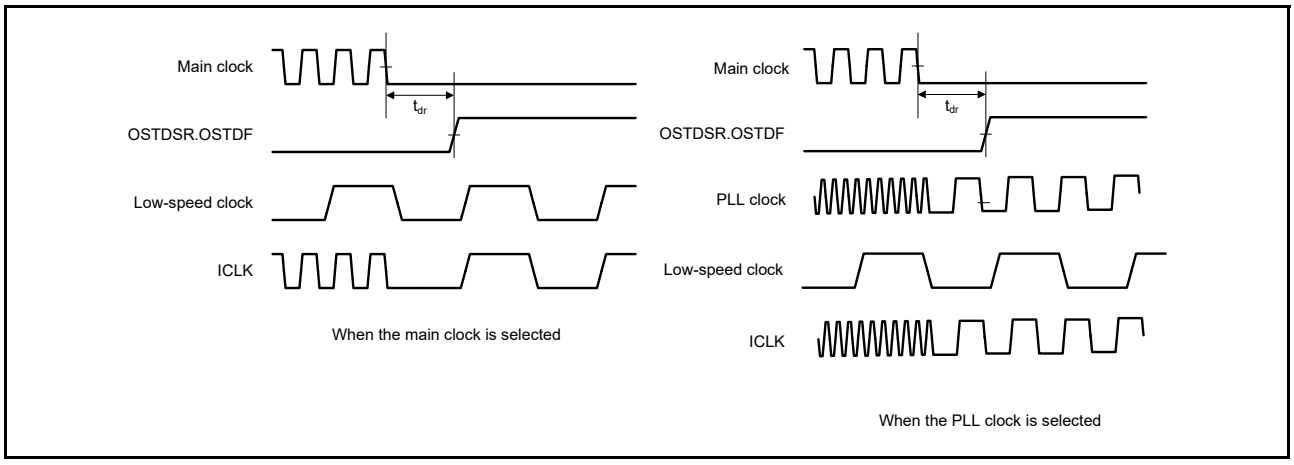


Figure 51.73 Oscillation Stop Detection Timing

Downloaded from Arrow.com.

51.12 Battery Backup Function Characteristics

Table 51.62 Battery Backup Function Characteristics

Conditions: 1.8 V ≤ VCC = VCC_USB = AVCC0 = VCC_RF = AVCC_RF ≤ 3.6 V, 1.8 V ≤ VBATT ≤ 3.6 V,
VSS = AVSS0 = VREFL0 = VSS_USB = VSS_RF = 0 V, T_a = –40 to +85°C

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Voltage level for switching to battery backup (falling)		$V_{DETBATT}$	1.99	2.09	2.19	V	Figure 51.74
Hysteresis width		V_{VBATTH}	—	100	—	mV	
VCC-off period for starting power supply switching		$t_{VOFFBATT}$	—	—	350	µs	
Allowable voltage change rising/falling gradient		dt/dVCC	1.0	—	—	ms/V	Figure 51.7
Level for detection of voltage drop on the VBATT pin (falling)	VBTLVDLVL[1:0] = 10b	$V_{DETBATLVD}$	2.11	2.20	2.29	V	Figure 51.74
	VBTLVDLVL[1:0] = 11b		1.87	2.00	2.13	V	
Hysteresis width for detection of voltage drop on the VBATT pin		$V_{BATLVDH}$	—	50	—	mV	

Note: The VCC-off period for starting power supply switching indicates the period in which VCC is below the minimum value of the voltage level for switching to battery backup ($V_{DETBATT}$).

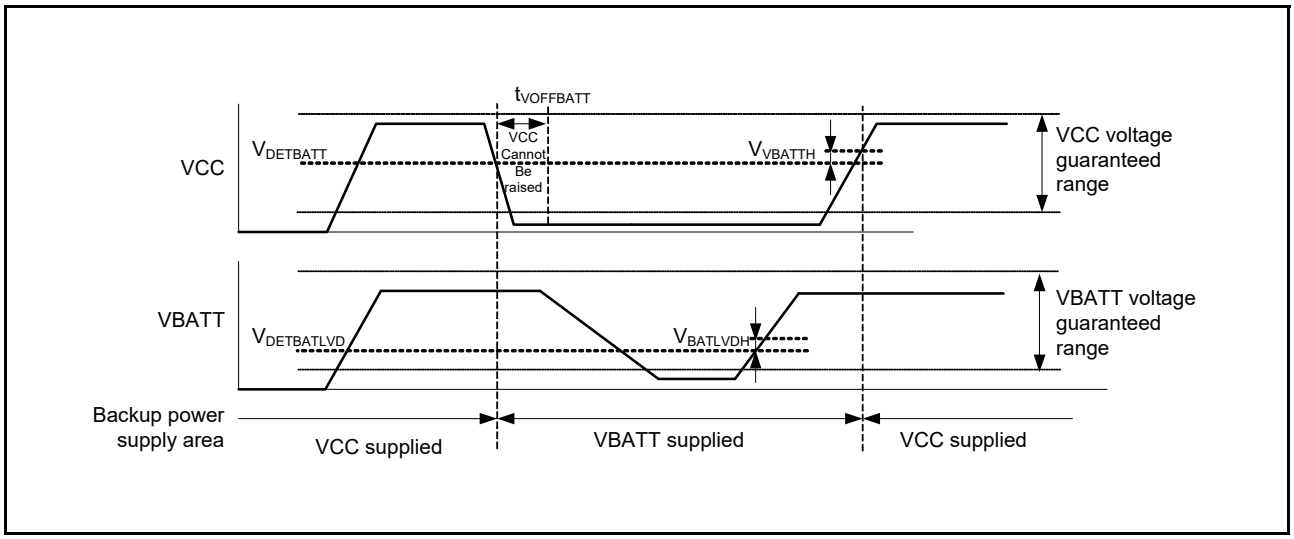


Figure 51.74 Battery Backup Function Characteristics

Downloaded from Arrow.com.

51.13 ROM (Flash Memory for Code Storage) Characteristics

Table 51.63 ROM (Flash Memory for Code Storage) Characteristics (1)

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Reprogramming/erasure cycle*1		N_{PEC}	1000	—	—	Times	
Data hold time	After 1000 times of N_{PEC}	t_{DRP}	20*2, *3	—	—	Year	T_a = +85°C

Note 1. Definition of reprogram/erase cycle: The reprogram/erase cycle is the number of erasing for each block. When the reprogram/erase cycle is n times (n = 1000), erasing can be performed n times for each block. For instance, when 4-byte programming is performed 256 times for different addresses in a 1-Kbyte block and then the entire block is erased, the reprogram/erase cycle is counted as one. However, programming the same address for several times as one erasing is not enabled (overwriting is prohibited).
Note 2. Characteristic when using the flash memory programmer and the self-programming library provided from Renesas Electronics.
Note 3. This result is obtained from reliability testing.

Table 51.64 ROM (Flash Memory for Code Storage) Characteristics (2) High-Speed Operating Mode

Conditions: 2.7 V ≤ VCC = VCC_USB = AVCC0 = VCC_RF = AVCC_RF ≤ 3.6 V, VSS = AVSS0 = VSS_USB = 0 V
Temperature range for the programming/erasure operation: T_a = –40 to +85°C

Item		Symbol	FCLK = 1 MHz			FCLK = 32 MHz			Unit
			Min.	Typ.	Max.	Min.	Typ.	Max.	
Programming time	8-byte	t_{P8}	—	112	967	—	52.3	491	µs
Erasure time	2-Kbyte	t_{E2K}	—	8.75	278	—	5.50	215	ms
	512-Kbyte (when block erase command is used)	t_{E512K}	—	928	19218	—	72.0	1679	ms
	512-Kbyte (when all-block erase command is used)	t_{EA512K}	—	923	19013	—	66.7	1469	ms
Blank check time	8-byte	t_{BC8}	—	—	55.0	—	—	16.1	µs
	2-Kbyte	t_{BC2K}	—	—	1840	—	—	136	ms
Erase operation forced stop time		t_{SED}	—	—	18.0	—	—	10.7	µs
Start-up area switching setting time		t_{SAS}	—	12.3	566.5	—	6.2	434	ms
Access window time		t_{AWS}	—	12.3	566.5	—	6.2	434	ms
ROM mode transition wait time 1		t_{DIS}	2.0	—	—	2.0	—	—	µs
ROM mode transition wait time 2		t_{MS}	5.0	—	—	5.0	—	—	µs

Note: The time until each operation of the flash memory is started after instructions are executed by software is not included.
Note: The lower-limit frequency of FCLK is 1 MHz during programming or erasing of the flash memory. When using FCLK at below 4 MHz, the frequency can be set to 1 MHz, 2 MHz, or 3 MHz. A non-integer frequency such as 1.5 MHz cannot be set.
Note: The frequency accuracy of FCLK must be within ±3.5%.

Downloaded from Arrow.com.

Table 51.65 ROM (Flash Memory for Code Storage) Characteristics (3) Middle-Speed Operating Mode

Conditions: 1.8 V ≤ VCC = VCC_USB = AVCC0 = VCC_RF = AVCC_RF ≤ 3.6 V, VSS = AVSS0 = VSS_USB= VSS_RF = 0 V
Temperature range for the programming/erasure operation: T_a = –40 to +85°C

Item		Symbol	FCLK = 1 MHz			FCLK = 8 MHz			Unit
			Min.	Typ.	Max.	Min.	Typ.	Max.	
Programming time	8-byte	t_{P8}	—	152	1367	—	97.9	936	µs
Erasure time	2-Kbyte	t_{E2K}	—	8.8	279.7	—	5.9	221	ms
	512-Kbyte (when block erase command is used)	t_{E512K}	—	928	19221	—	191	4108	ms
	512-Kbyte (when all-block erase command is used)	t_{EA512K}	—	923	19015	—	185	3901	ms
Blank check time	8-byte	t_{BC8}	—	—	85.0	—	—	50.88	µs
	2-Kbyte	t_{BC2K}	—	—	1870	—	—	402	µs
Erase operation forced stop time		t_{SED}	—	—	28.0	—	—	21.3	µs
Start-up area switching setting time		t_{SAS}	—	13.0	573.3	—	7.7	451	ms
Access window time		t_{AWS}	—	13.0	573.3	—	7.7	451	ms
ROM mode transition wait time 1		t_{DIS}	2.0	—	—	2.0	—	—	µs
ROM mode transition wait time 2		t_{MS}	3.0	—	—	3.0	—	—	µs

Note: The time until each operation of the flash memory is started after instructions are executed by software is not included.
Note: The lower-limit frequency of FCLK is 1 MHz during programming or erasing of the flash memory. When using FCLK at below 4 MHz, the frequency can be set to 1 MHz, 2 MHz, or 3 MHz. A non-integer frequency such as 1.5 MHz cannot be set.
Note: The frequency accuracy of FCLK must be within ±3.5%.

Downloaded from Arrow.com.

51.14 E2 DataFlash Characteristics (Flash Memory for Data Storage)

Table 51.66 E2 DataFlash Characteristics (1)

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Reprogramming/erasure cycle*1		N_{DPEC}	100000	1000000	—	Times	
Data hold time	After 10000 times of N_{DPEC}	t_{DDRP}	20*2, *3	—	—	Year	T_a = +85°C
	After 100000 times of N_{DPEC}		5*2, *3	—	—	Year	
	After 1000000 times of N_{DPEC}		—	1*2, *3	—	Year	T_a = +25°C

Note 1. The reprogram/erase cycle is the number of erasing for each block. When the reprogram/erase cycle is n times (n = 100000), erasing can be performed n times for each block. For instance, when 1-byte programming is performed 1000 times for different addresses in a 1-Kbyte block and then the entire block is erased, the reprogram/erase cycle is counted as one. However, programming the same address for several times as one erasing is not enabled (overwriting is prohibited).
Note 2. Characteristic when the flash memory programmer is used and the self-programming library is provided from Renesas Electronics.
Note 3. These results are obtained from reliability testing.

Table 51.67 E2 DataFlash Characteristics (2): high-speed operating mode

Conditions: 2.7 V ≤ VCC = VCC_USB = AVCC0 = VCC_RF = AVCC_RF ≤ 3.6 V, VSS = AVSS0 = VSS_USB = VSS_RF = 0 V
Temperature range for the programming/erasure operation: T_a = –40 to +85°C

Item		Symbol	FCLK = 1 MHz			FCLK = 32 MHz			Unit
			Min.	Typ.	Max.	Min.	Typ.	Max.	
Programming time	1 byte	t_{DP1}	—	95.0	797	—	40.8	376	µs
Erasure time	1 Kbyte	t_{DE1K}	—	19.5	498	—	6.2	230	ms
	8 Kbyte	t_{DE8K}	—	119.8	2556	—	12.9	368	ms
Blank check time	1 byte	t_{DBC1}	—	—	55.00	—	—	16.1	µs
	1 Kbyte	t_{DBC1K}	—	—	0.72	—	—	0.50	ms
Erase operation forced stop time		t_{DSED}	—	—	16.0	—	—	10.7	µs
DataFlash STOP recovery time		t_{DSTOP}	5.0	—	—	5.0	—	—	µs

Note: The time until each operation of the flash memory is started after instructions are executed by software is not included.
Note: The lower-limit frequency of FCLK is 1 MHz during programming or erasing of the flash memory. When using FCLK at below 4 MHz, the frequency can be set to 1 MHz, 2 MHz, or 3 MHz. A non-integer frequency such as 1.5 MHz cannot be set.
Note: The frequency accuracy of FCLK must be within ±3.5%.

Table 51.68 E2 DataFlash Characteristics (3): middle-speed operating mode

Conditions: 1.8 V ≤ VCC0 = VCC_USB = AVCC0 = VCC_RF = AVCC_RF ≤ 3.6 V, VSS = AVSS0 = VSS_USB = VSS_RF = 0 V
Temperature range for the programming/erasure operation: T_a = –40 to +85°C

Item		Symbol	FCLK = 1 MHz			FCLK = 8 MHz			Unit
			Min.	Typ.	Max.	Min.	Typ.	Max.	
Programming time	1 byte	t_{DP1}	—	135	1197	—	86.5	823	µs
Erasure time	1 Kbyte	t_{DE1K}	—	19.6	501	—	8.0	265	ms
	8 Kbyte	t_{DE8K}	—	120	2558	—	27.7	669	ms
Blank check time	1 byte	t_{DBC1}	—	—	85.0	—	—	50.9	µs
	1 Kbyte	t_{DBC1K}	—	—	0.72	—	—	1.45	ms
Erase operation forced stop time		t_{DSED}	—	—	28.0	—	—	21.3	µs
DataFlash STOP recovery time		t_{DSTOP}	0.72	—	—	0.72	—	—	µs

Note: The time until each operation of the flash memory is started after instructions are executed by software is not included.
Note: The lower-limit frequency of FCLK is 1 MHz during programming or erasing of the flash memory. When using FCLK at below 4 MHz, the frequency can be set to 1 MHz, 2 MHz, or 3 MHz. A non-integer frequency such as 1.5 MHz cannot be set.
Note: The frequency accuracy of FCLK must be within ±3.5%.

Downloaded from Arrow.com.

51.15 BLE Characteristics

51.15.1 Transmission Characteristics

Table 51.69 Transmission Characteristics

Conditions: VCC = VCC_RF = AVCC_RF = 3.3 V, VSS = VSS_RF = 0 V, T_a = +25°C

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Range of frequency		RF_{CF}	2402	—	2480	MHz	
Data rate		RF_{DATA_2M}	—	2	—	Mbps	
		RF_{DATA_1M}	—	1	—	Mbps	
		RF_{DATA_500k}	—	500	—	kbps	
		RF_{DATA_125k}	—	125	—	kbps	
Maximum transmitted output power		RF_{POWER}	—	0	2	dBm	0 dBm output mode
			—	4	6	dBm	4 dBm output mode
Output frequency error	85-pin BGA, 56-pin QFN	RF_{TXFERR}	–10	—	10	ppm	*1
	83-pin LGA	$RF_{MTXFERR}$	–50	—	50	ppm	T_a: –40 to +85°C

Note: The characteristics are based on pins and functions other than those for the BLE interface not being in use.
Note 1. This does not take frequency errors due to manufacturing irregularities, drift with temperature, or deterioration of the crystal over time into account.

51.15.2 Reception Characteristics (2 Mbps)

Table 51.70 Reception Characteristics

Conditions: VCC = VCC_RF = AVCC_RF = 3.3 V, VSS = VSS_RF = 0 V, T_a = +25°C

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions	
Input frequency	RF_{RXFIN_2M}	2402	—	2480	MHz		
Maximum input level	RF_{LEVL_2M}	–10	4	—	dBm	*1	
Receiver sensitivity	RF_{STY_2M}	—	–92	—	dBm	*1	
Secondary emission strength	RF_{RXSP_2M}	—	–72	–57	dBm	30 MHz to 1 GHz	
		—	–54	–47	dBm	1 GHz to 12 GHz	
Co-channel rejection ratio	RF_{CCR_2M}	—	–8	—	dB	Prf = –67 dBm*1	
Adjacent channel rejection ratio	RF_{ADCR_2M}	—	2	—	dB	Prf = –67 dBm*1	±2 MHz
		—	35	—	dB		±4 MHz
		—	39	—	dB		±6 MHz
Blocking	RF_{BLK_2M}	—	–1	—	dBm	Prf = –67 dBm*1	30 MHz to 2000 MHz
		—	–25	—	dBm		2000 MHz to 2399 MHz
		—	–21	—	dBm		2484 MHz to 3000 MHz
		—	–10	—	dBm		> 3000 MHz
Allowable frequency deviation*2	RF_{RXFER_2M}	–120	—	120	ppm	*1	
RSSI accuracy	RF_{RSSIS_2M}	—	±4	—	dB	–70 dBm ≤ Prf ≤ –10 dBm	

Note: The characteristics are based on pins and functions other than those for the BLE interface not being in use.
Note 1. PER ≤ 30.8%, and a 37-byte payload
Note 2. Allowable range of difference between the center frequency for the RF input signals and the carrier frequency generated within the chip

Downloaded from Arrow.com.

51.15.3 Reception Characteristics (1 Mbps)

Table 51.71 Reception Characteristics

Conditions: VCC = VCC_RF = AVCC_RF = 3.3 V, VSS = VSS_RF = 0 V, T_a = +25°C

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions	
Input frequency	RF_{RXFIN_1M}	2402	—	2480	MHz		
Maximum input level	RF_{LEVL_1M}	–10	4	—	dBm	*1	
Receiver sensitivity	RF_{STY_1M}	—	–95	—	dBm	*1	
Secondary emission strength	RF_{RXSP_1M}	—	–72	–57	dBm	30MHz to 1GHz	
		—	–54	–47	dBm	1GHz to 12GHz	
Co-channel rejection ratio	RF_{CCR_1M}	—	–7	—	dB	Prf = –67dBm*1	
Adjacent channel rejection ratio	RF_{ADCR_1M}	—	–1	—	dB	Prf = –67dBm*1	±1MHz
		—	34	—	dB		±2MHz
		—	35	—	dB		±3MHz
Blocking	RF_{BLK_1M}	—	0	—	dBm	Prf = –67dBm*1	30MHz to 2000MHz
		—	–24	—	dBm		2000MHz to 2399MHz
		—	–20	—	dBm		2484MHz to 3000MHz
		—	–4	—	dBm		> 3000MHz
Allowable frequency deviation*2	RF_{RXFER_1M}	–120	—	120	ppm	*1	
RSSI accuracy	RF_{RSSIS_1M}	—	±4	—	dB	–70dBm ≤ Prf ≤ –10dBm	

Note: The characteristics are based on pins and functions other than those for the BLE interface not being in use.
Note 1. PER ≤ 30.8%, and a 37-byte payload
Note 2. Allowable range of difference between the center frequency for the RF input signals and the carrier frequency generated within the chip

51.15.4 Reception Characteristics (500 kbps)

Table 51.72 Reception Characteristics

Conditions: VCC = VCC_RF = AVCC_RF = 3.3 V, VSS = VSS_RF = 0 V, T_a = +25°C

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions	
Input frequency	RF_{RXFIN_500k}	2402	—	2480	MHz		
Maximum input level	RF_{LEVL_500k}	–10	4	—	dBm	*1	
Receiver sensitivity	RF_{STY_500k}	—	–100	—	dBm	*1	
Secondary emission strength	RF_{RXSP_500k}	—	–72	–57	dBm	30MHz to 1GHz	
		—	–54	–47	dBm	1GHz to 12GHz	
Co-channel rejection ratio	RF_{CCR_500k}	—	–4	—	dB	Prf = –72dBm*1	
Adjacent channel rejection ratio	RF_{ADCR_500k}	—	6	—	dB	Prf = –72dBm*1	±1MHz
		—	36	—	dB		±2MHz
		—	42	—	dB		±3MHz
Blocking	RF_{BLK_500k}	—	0	—	dBm	Prf = –72dBm*1	30MHz to 2000MHz
		—	–23	—	dBm		2000MHz to 2399MHz
		—	–20	—	dBm		2484MHz to 3000MHz
		—	–7	—	dBm		> 3000MHz
Allowable frequency deviation*2	RF_{RXFER_500k}	–120	—	120	ppm	*1	
RSSI accuracy	RF_{RSSIS_500k}	—	±4	—	dB	–70dBm ≤ Prf ≤ –10dBm	

Note: The characteristics are based on pins and functions other than those for the BLE interface not being in use.
Note 1. PER ≤ 30.8%, and a 37-byte payload
Note 2. Allowable range of difference between the center frequency for the RF input signals and the carrier frequency generated within the chip

Downloaded from Arrow.com.

51.15.5 Reception Characteristics (125 kbps)

Table 51.73 Reception Characteristics

Conditions: VCC = VCC_RF = AVCC_RF = 3.3 V, VSS = VSS_RF = 0 V, T_a = +25°C

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions	
Input frequency	RF_{RXFIN_125k}	2402	—	2480	MHz		
Maximum input level	RF_{LEVL_125k}	–10	4	—	dBm	*1	
Receiver sensitivity	RF_{STY_125k}	—	–105	—	dBm	*1	
Secondary emission strength	RF_{RXSP_125k}	—	–72	–57	dBm	30 MHz to 1 GHz	
		—	–54	–47	dBm	1 GHz to 12 GHz	
Co-channel rejection ratio	RF_{CCR_125k}	—	–2	—	dB	Prf = –79 dBm*1	
Adjacent channel rejection ratio	RF_{ADCR_125k}	—	12	—	dB	Prf = –79 dBm*1	±1 MHz
		—	39	—	dB		±2 MHz
		—	45	—	dB		±3 MHz
Blocking	RF_{BLK_125k}	—	0	—	dBm	Prf = –79 dBm*1	30 MHz to 2000 MHz
		—	–23	—	dBm		2000 MHz to 2399 MHz
		—	–20	—	dBm		2484 MHz to 3000 MHz
		—	–1	—	dBm		> 3000MHz
Allowable frequency deviation*2	RF_{RXFER_125k}	–120	—	120	ppm	*1	
RSSI accuracy	RF_{RSSIS_125k}	—	±4	—	dB	–70 dBm ≤ Prf ≤ –10 dBm	

Note: The characteristics are based on pins and functions other than those for the BLE interface not being in use.
Note 1. PER ≤ 30.8%, and a 37-byte payload
Note 2. Allowable range of difference between the center frequency for the RF input signals and the carrier frequency generated within the chip

Downloaded from Arrow.com.

51.16 Usage Notes

51.16.1 Connecting VCL Capacitor and Bypass Capacitors

This MCU integrates an internal voltage-down circuit, which is used for lowering the power supply voltage in the internal MCU automatically to the optimum level. A 4.7-µF capacitor needs to be connected between this internal voltage-down power supply (VCL pin) and the VSS pin. Place an external capacitor close to the pins. Do not apply the power supply voltage to the VCL pin.

Insert a multilayer ceramic capacitor as a bypass capacitor between each pair of the power supply pins. Implement a bypass capacitor as closer to the MCU power supply pins as possible. We recommend capacitors with a value of 2.2 µF for that connected to the VCC_RF pin and 0.1 µF for the others. For the capacitors related to crystal oscillation, see section 9, Clock Generation Circuit. For the capacitors related to analog modules, also see section 44, 12-Bit A/D Converter (S12ADE).

For notes on designing the printed circuit board, see the descriptions of the application note, the Hardware Design Guide (R01AN1411EJ). The latest version can be downloaded from the Renesas Electronics website.

Downloaded from Arrow.com.

Appendix 1. Port States in Each Processing Mode

Table 1.1 Port States in Each Processing Mode

Port Name (Pin Name)	Operating Mode According to Registers Setting		Reset	Software Standby Mode	
				OPE = 1	OPE = 0
P03 (DA0)	All	DA0 output (DAOE0 = 1)	Hi-Z	DA output retained	
		Other than the above (DAOE0 = 0)		Keep-O	
P05 (DA1)	All	DA1 output (DAOE1 = 1)	Hi-Z	DA output retained	
		Other than the above (DAOE1 = 0)		Keep-O	
P07	All		Hi-Z	Keep-O	
P14 (USB0_OVRCURA/ IRQ4)	All		Hi-Z	Keep-O*1, *2	
P15 (IRQ5)	All		Hi-Z	Keep-O*1	
P16 (USB0_VBUS/ USB0_OVRCURB/ IRQ6/RTCOUT)	All		Hi-Z	[RTCOUT output] RTCOUT output [Other than the above] Keep-O*1, *2	
P17 (CMPOB2/IRQ7)	All		Hi-Z	[CMPOB2 output] CMPOB2 output [Other than the above] Keep-O*1	
P21	All		Hi-Z	Keep-O	
P22 (USB0_OVRCURB)	All		Hi-Z	Keep-O*2	
P25 to P27	All		Hi-Z	Keep-O	
P30 (CMPOB3/IRQ0)	All		Hi-Z	[CMPOB3 output] CMPOB3 output [Other than the above] Keep-O*1	
P31 (IRQ1)	All		Hi-Z	Keep-O*1	
P35 (NMI)	All		Hi-Z	Keep-O*1	
P36, P37	All		Hi-Z	Keep-O	
P40 to P47	All		Hi-Z	Keep-O	
PB0	All		Hi-Z	Keep-O	
PB1 (IRQ4)	All		Hi-Z	Keep-O*1	
PB3	All		Hi-Z	Keep-O	
PB5 (USB0_VBUS)	All		Hi-Z	Keep-O*2	
PB7	All		Hi-Z	Keep-O	
PC0, PC2 to PC7	All		Hi-Z	Keep-O	
PD3	All		Hi-Z	Keep-O	
PE0, PE1	All		Hi-Z	Keep-O	
PE2 (IRQ7)	All		Hi-Z	Keep-O*1	
PE3, PE4 (CLKOUT, CLKOUT)	All		Hi-Z	[CLKOUT output] CLKOUT output [Other than the above] Keep-O	
PJ3	All		Hi-Z	Keep-O	

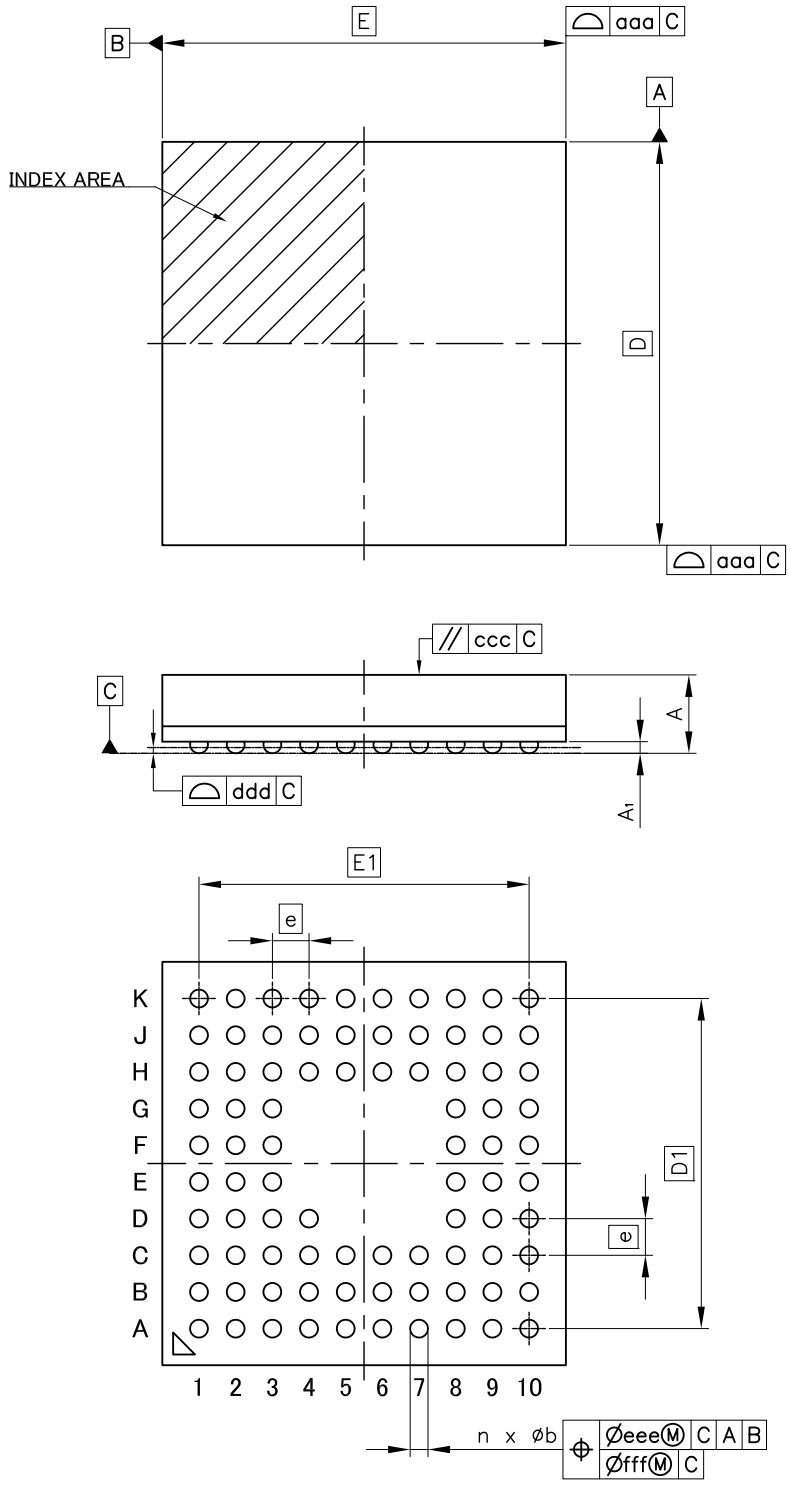
Keep-O: Output pins retain their previous values, and input pins become high-impedance.
Hi-Z: High-impedance
Note 1. Input is enabled if the pin is specified as the software standby mode canceling source while it is used as an external interrupt pin.
Note 2. Input is enabled if the pin is used a USB pin (USB0_VBUS/USB0_OVRCURA/USB0_OVRCURB).

Downloaded from Arrow.com.

Appendix 2. Package Dimensions

Information on the latest version of the package dimensions or mountings has been displayed in “Packages” on Renesas Electronics Corporation website.

JEITA Package Code	RENESAS Code	MASS(Typ.)[g]
P-TFBGA85-5.5×5.5-0.50	PTBG0085KB-A	0.07

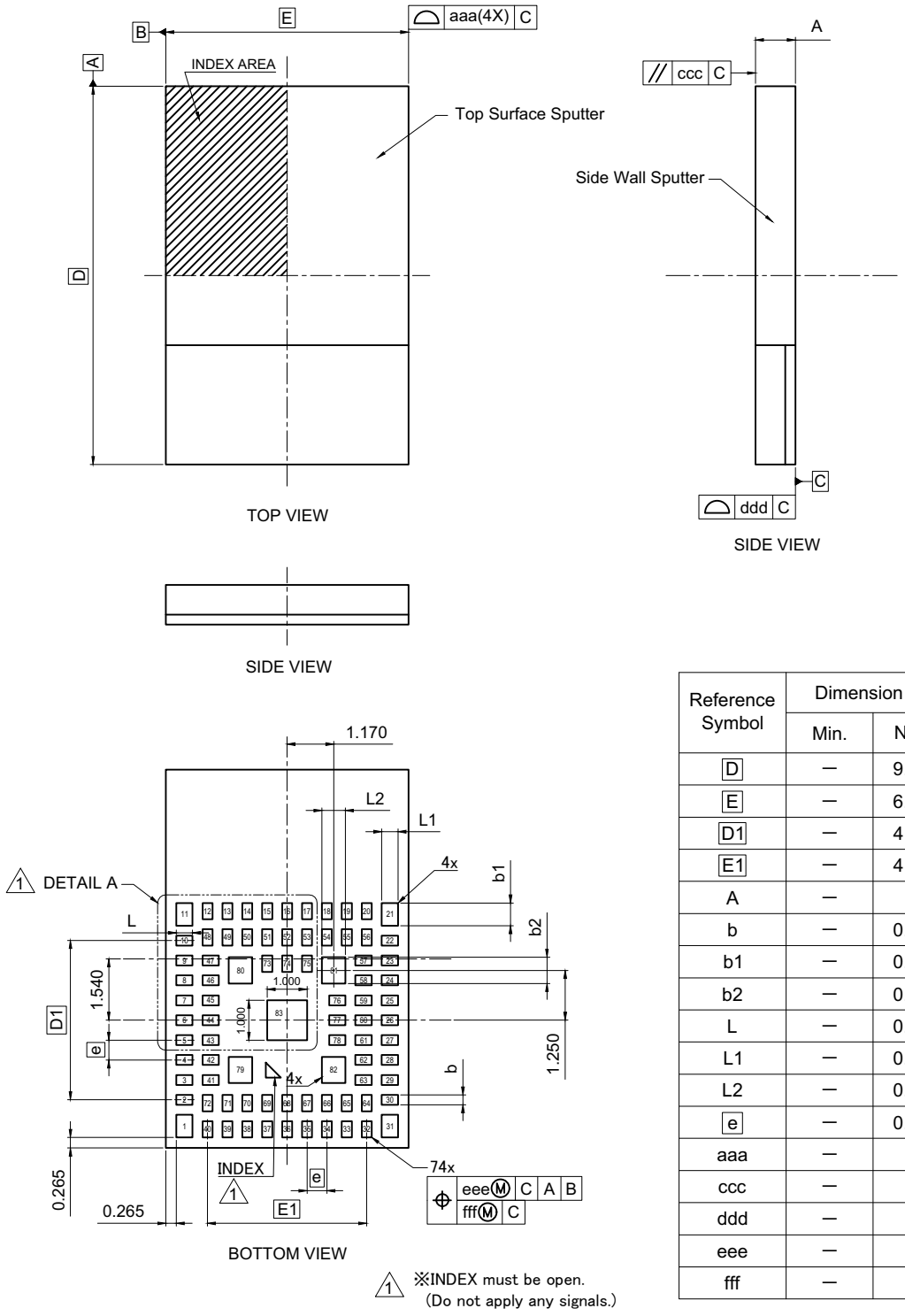


Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
D	—	5.500	—
E	—	5.500	—
D1	—	4.500	—
E1	—	4.500	—
A	—	—	1.200
A_1	0.110	—	0.210
b	0.200	—	0.300
e	—	0.500	—
aaa	—	—	0.100
ccc	—	—	0.100
ddd	—	—	0.080
eee	—	—	0.150
fff	—	—	0.080
n	—	85	—

Figure A 85-Pin BGA (PTBG0085KB-A)

Downloaded from Arrow.com.

JEITA Package code	RENESAS code	MASS(TYP.)[g]
P-TFLGA83-6.1x9.5-0.50	PTLG0083KA-A	0.12



Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
D	—	9.500	—
E	—	6.100	—
D1	—	4.000	—
E1	—	4.000	—
A	—	—	1.009
b	—	0.250	—
b1	—	0.570	—
b2	—	0.660	—
L	—	0.410	—
L1	—	0.410	—
L2	—	0.590	—
e	—	0.500	—
aaa	—	—	0.150
ccc	—	—	0.200
ddd	—	—	0.120
eee	—	—	0.150
fff	—	—	0.080

Figure B 83-Pin LGA (PTLG0083KA-A) (1/2)

Downloaded from Arrow.com.

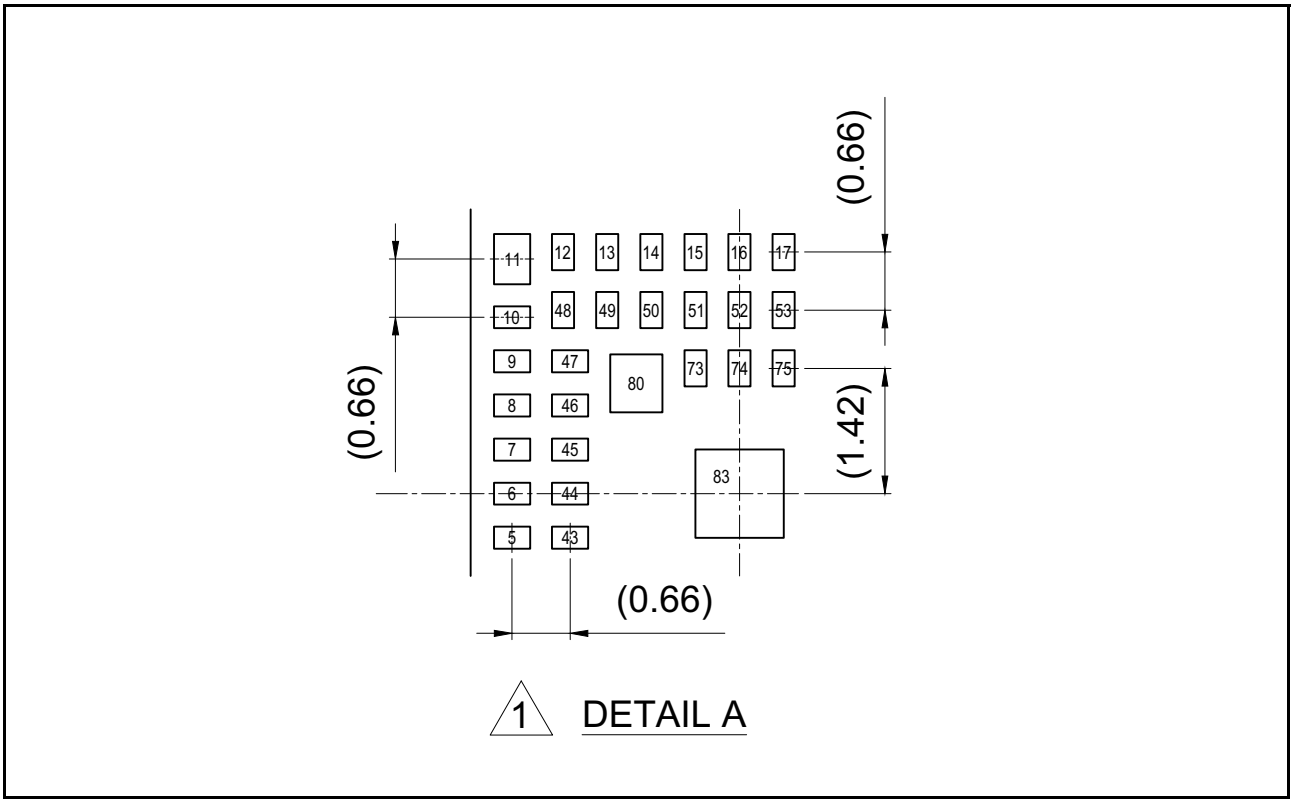
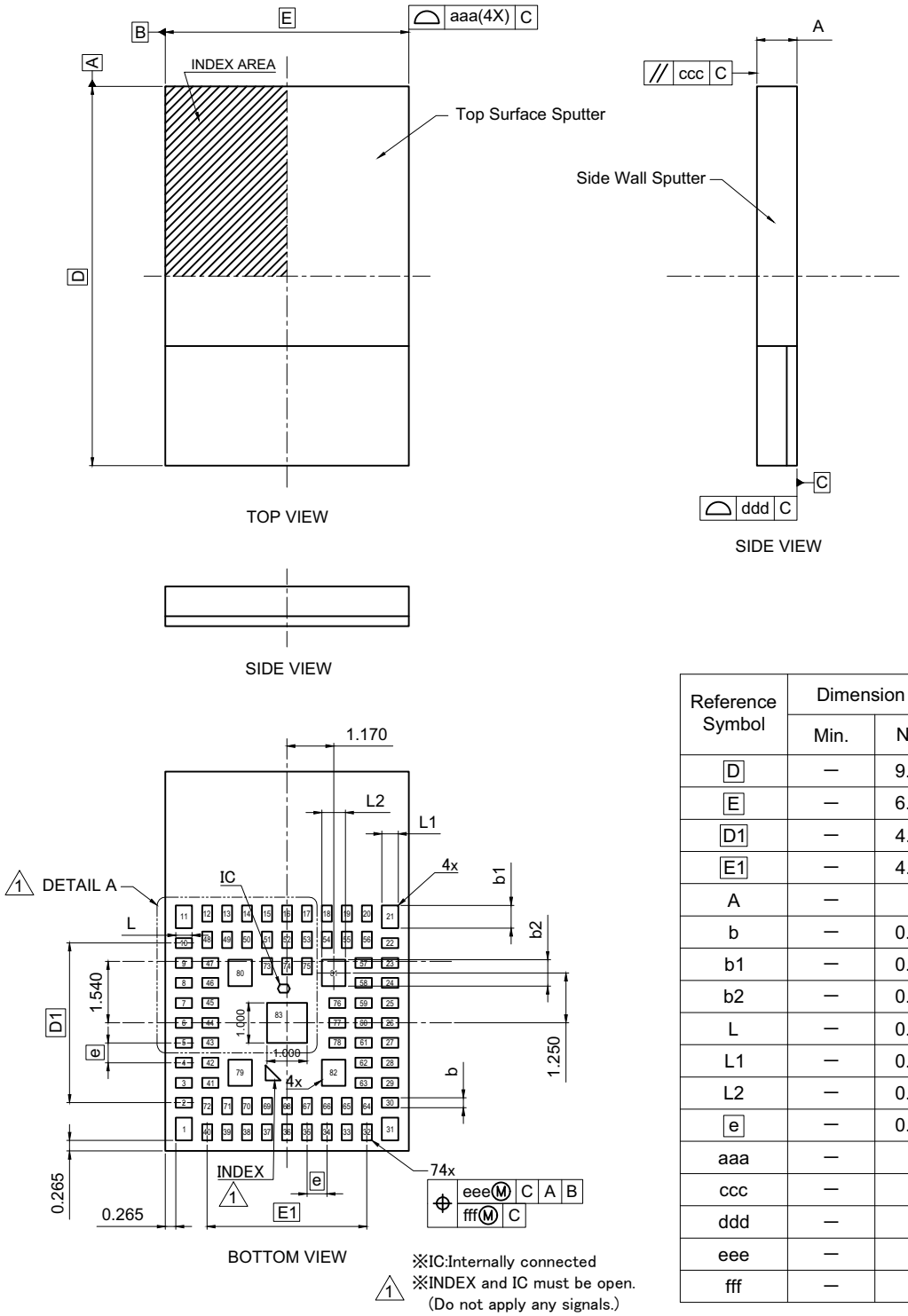


Figure B 83-Pin LGA (PTLG0083KA-A) (2/2)

Downloaded from Arrow.com.

JEITA Package code	RENESAS code	MASS(TYP.)[g]
P-TFLGA83-6.1x9.5-0.50	PTLG0083KA-B	0.12



Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
D	—	9.500	—
E	—	6.100	—
D1	—	4.000	—
E1	—	4.000	—
A	—	—	1.009
b	—	0.250	—
b1	—	0.570	—
b2	—	0.660	—
L	—	0.410	—
L1	—	0.410	—
L2	—	0.590	—
e	—	0.500	—
aaa	—	—	0.150
ccc	—	—	0.200
ddd	—	—	0.120
eee	—	—	0.150
fff	—	—	0.080

Figure C 83-Pin LGA (PTLG0083KA-B) (1/2)

Downloaded from Arrow.com.

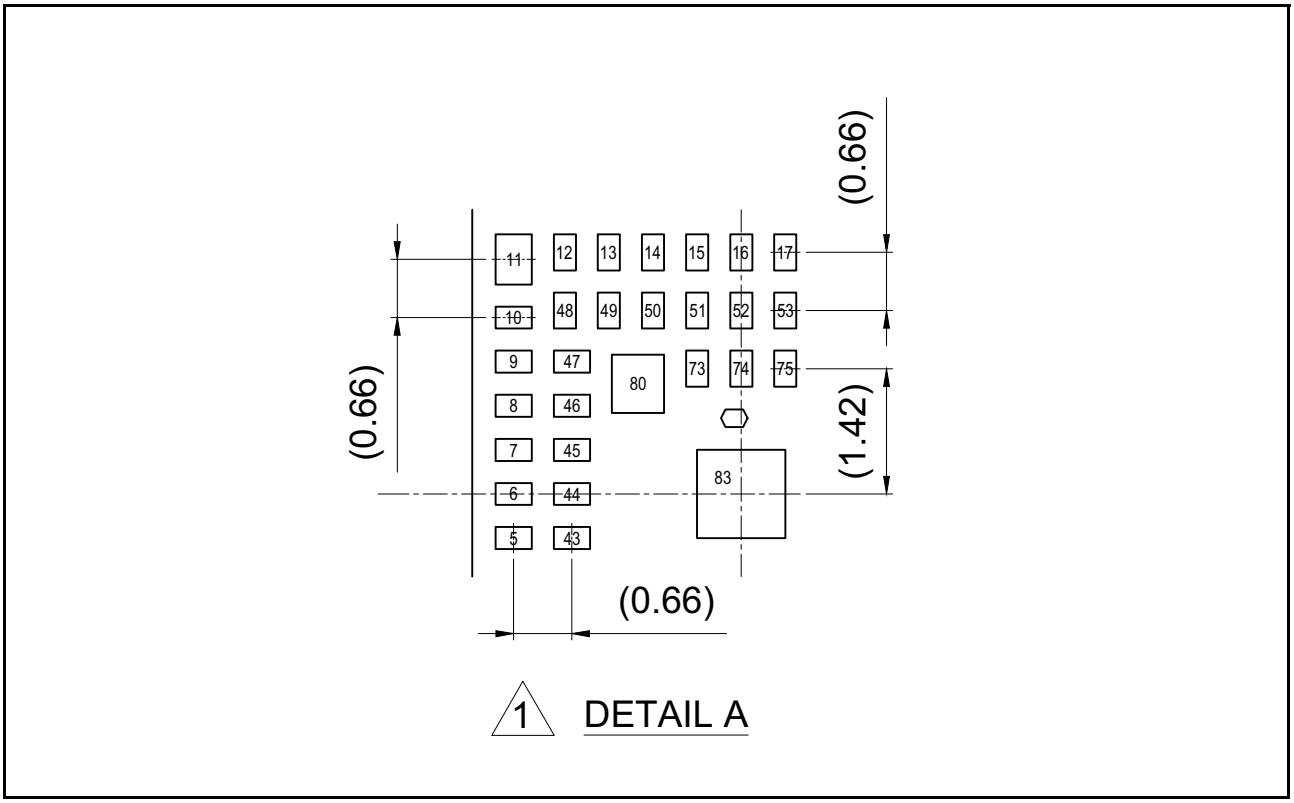
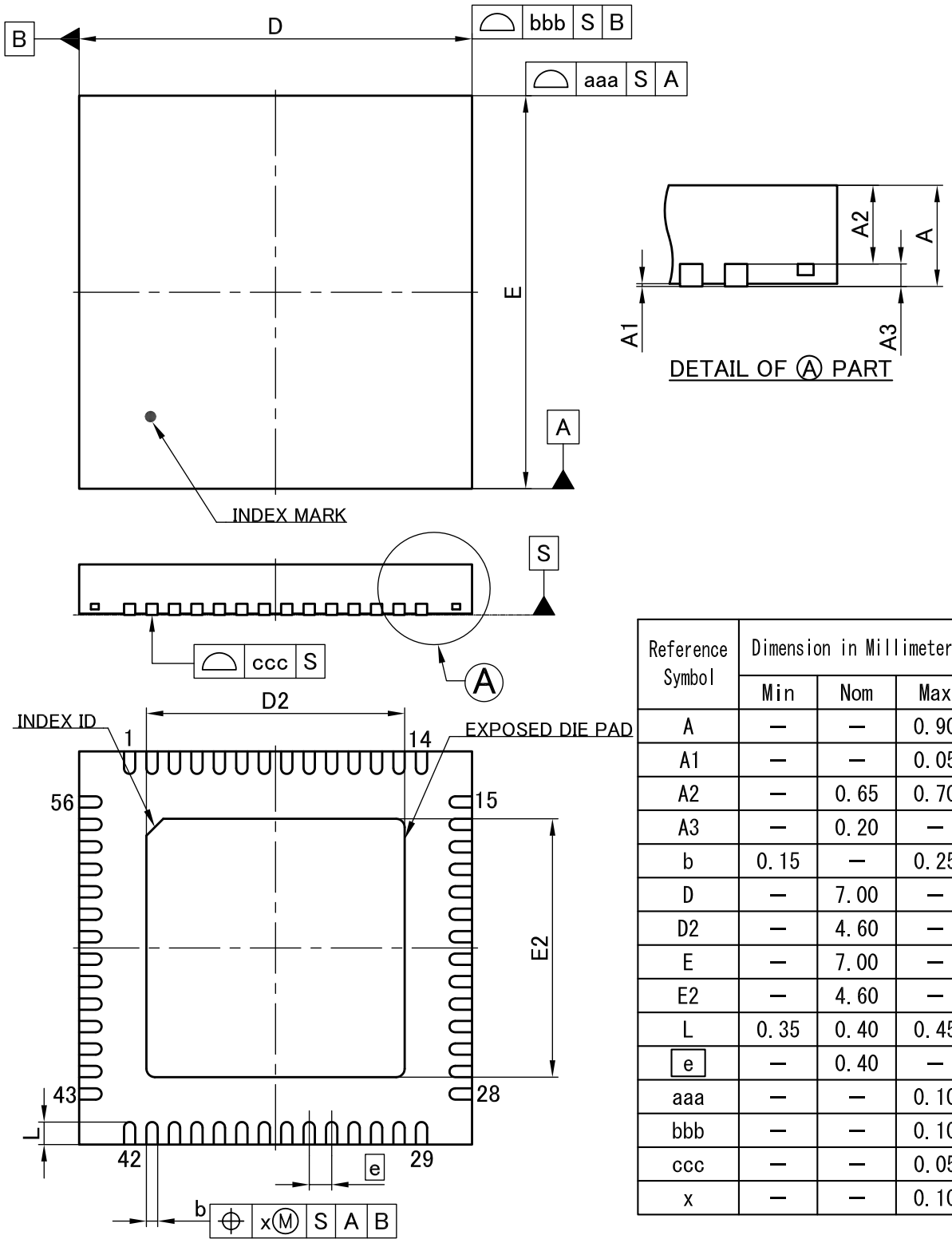


Figure C 83-Pin LGA (PTLG0083KA-B) (2/2)

Downloaded from Arrow.com.

JEITA Package Code	RENESAS Code	MASS(Typ.) [g]
P-HVQFN56-7×7-0.40	PVQN0056LA-A	0.13



Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
A	—	—	0.90
A1	—	—	0.05
A2	—	0.65	0.70
A3	—	0.20	—
b	0.15	—	0.25
D	—	7.00	—
D2	—	4.60	—
E	—	7.00	—
E2	—	4.60	—
L	0.35	0.40	0.45
e	—	0.40	—
aaa	—	—	0.10
bbb	—	—	0.10
ccc	—	—	0.05
x	—	—	0.10

Figure D 56-Pin QFN (PVQN0056LA-A)

Downloaded from Arrow.com.

REVISION HISTORY	RX23W Group User's Manual: Hardware

Classifications

- Items with Technical Update document number: Changes according to the corresponding issued Technical Update
- Items without Technical Update document number: Minor changes that do not require Technical Update to be issued

Rev.	Date	Description		Classification
		Page	Summary	
1.00	Jul 31, 2019	—	First edition, issued	
1.10	Mar 30, 2021	Features		
		51	83-pin LGA specifications, added	
		1. Overview		
		All	83-pin LGA specifications, added	
		9. Clock Generation Circuit		
		173, 174	Table 9.1 Specifications of Clock Generation Circuit, changed	
		176	Figure 9.2 Block Diagram of Clock Generation Circuit (83-Pin LGA), added	
		206	9.5 Dedicated Clock Oscillator for Bluetooth, changed	
		206	9.5.1 Connecting the Oscillator, added	
		206	Figure 9.11 Example of the Connection of a 32-MHz Crystal Resonator, changed	
		207	Figure 9.12 Example of Connection of the Bluetooth-dedicated clock output pin, added	
		212	9.9.2 Note on Rewriting the SCKCR3 Register, added	TN-RX*-A0224B/E
		19. Data Transfer Controller (DTCa)		
		394	19.2.8 DTC Vector Base Register (DTCVBR), changed	
		20. Event Link Controller (ELC)		
		421	Table 20.3 Correspondence between Values Set in ELSRn.ELS[7:0] Bits and Event Signals (1/2), changed	
		424	20.2.5 Event Link Option Setting Register C (ELOPC), changed	
		431	Table 20.5 Operations of Peripheral Modules When Event Signal is Input, changed	
		432	20.3.4 Operation of CTSU When Event Signal is Input, added	
		22. Multi-Function Pin Controller (MPC)		
		All	83-pin LGA specifications, added	
		28. Realtime Clock (RTCe)		
		789, 790	28.2.18 RTC Control Register 2 (RCR2), changed	TN-RX*-A0237B/E
		793, 794	28.2.21 Time Capture Control Register n (RTCCRn) (n = 0, 1), changed	
		801	Figure 28.3 Clock and Count Mode Setting Procedure, changed	TN-RX*-A0237B/E
		802	Figure 28.4 Setting the Time, changed	
		802	Figure 28.5 30-Second Adjustment, changed	
		808	28.3.9 Time Capture Function, title changed	
		813	28.6.5 Notes on Writing to and Reading from Registers, title changed	
		814	Figure 28.14 Initialization Procedure, changed	TN-RX*-A0237B/E
		29. Low-Power Timer (LPT)		
		815	Table 29.1 LPT Specifications, changed	
		815	Figure 29.1 LPT Block Diagram, changed	
		816, 817	29.2.1 Low-Power Timer Control Register 1 (LPTCR1), changed	
		818	29.2.3 Low-Power Timer Control Register 3 (LPTCR3), changed	
		819	29.2.4 Low-Power Timer Period Setting Register (LPTPRD), changed	
		820	Table 29.2 Example of Low-Power Timer Period Settings for IWDTCLK, changed	
		821	Table 29.3 Example of Low-Power Timer Period Settings for Sub-Clock, changed	
		822	29.2.5 Low-Power Timer Compare Register 0 (LPCMR0), changed	
		823	29.2.6 Low-Power Timer Standby Wakeup Enable Register (LPWUCR), changed	
		824, 825	29.3.1 Periodic Counting Operation, changed	
		827	29.4 Wakeup from Software Standby Mode by an Interrupt through the Event Link Controller (ELC), changed	

Downloaded from Arrow.com.

Rev.	Date	Description		Classification
		Page	Summary	
1.10	Mar 30, 2021	32. USB 2.0 Host/Function Module (USBc)		
		861, 862	32.1 Overview, changed	
		861	Table 32.1 USB Specifications, changed	
		865	32.2.2 System Configuration Status Register 0 (SYSSTS0), changed	
		866 to 868	32.2.3 Device State Control Register 0 (DVSTCTR0), changed	
		869, 870	32.2.4 CFIFO Port Register (CFIFO), D0FIFO Port Register (D0FIFO), D1FIFO Port Register (D1FIFO), changed	
		873, 874	32.2.5 CFIFO Port Select Register (CFIFOSEL), D0FIFO Port Select Register (D0FIFOSEL), D1FIFO Port Select Register (D1FIFOSEL) • D0FIFOSEL, D1FIFOSEL, changed	
		875, 876	32.2.6 CFIFO Port Control Register (CFIFOCTR), D0FIFO Port Control Register (D0FIFOCTR), D1FIFO Port Control Register (D1FIFOCTR), changed	
		877	32.2.7 Interrupt Enable Register 0 (INTENB0), changed	
		878	32.2.8 Interrupt Enable Register 1 (INTENB1), changed	
		879	32.2.9 BRDY Interrupt Enable Register (BRDYENB), changed	
		880	32.2.10 NRDY Interrupt Enable Register (NRDYENB), changed	
		881	32.2.11 BEMP Interrupt Enable Register (BEMPENB), changed	
		886 to 888	32.2.14 Interrupt Status Register 1 (INTSTS1), changed	
		889	32.2.15 BRDY Interrupt Status Register (BRDYSTS), changed	
		890	32.2.16 NRDY Interrupt Status Register (NRDYSTS), changed	
		891	32.2.17 BEMP Interrupt Status Register (BEMPSTS), changed	
		901	32.2.26 Pipe Window Select Register (PIPESEL), changed	
		902, 903	32.2.27 Pipe Configuration Register (PIPECFG), changed	
		904	32.2.28 Pipe Maximum Packet Size Register (PIPEMAXP), changed	
		905	32.2.29 Pipe Cycle Control Register (PIPEPERI), changed	
		915	32.2.32 Pipe n Transaction Counter Register (PIPEnTRN) (n = 1 to 5), changed	
		916	32.2.33 Device Address n Configuration Register (DEVADDn) (n = 0 to 5), changed	
		917	32.2.34 USB Module Control Register (USBMC), changed	
		920	32.3.1.2 Controller Function Selection, changed	
		920	32.3.1.3 Controlling USB Data Bus Resistors, changed	
		921 to 926	32.3.1.4 Example of USB External Connection Circuit, changed	
		923	Figure 32.3 Functional Connection of USB Connector in Self-Powered State, changed	
		924	Figure 32.4 Functional Connection Sample of USB Connector with Battery Charging Specification Rev.1.2 Supported, changed	
		926	Figure 32.6 Functional Connection Sample of USB Connector in Bus Powered State, changed	
		927	Table 32.12 Interrupt Sources, changed	
		—	Figure 32.7 Functional Connection Sample of USB Connector in Bus Powered State (2), deleted	
		928	Figure 32.7 Circuits Related to Interrupts in USB, changed	
		930	32.3.3.1 BRDY Interrupt (2) When the SOFCFG.BRDYM Bit = 0 and the PIPECFG.BFRE Bit = 1, changed	
		941	Table 32.15 Pipe Settings, changed	
		942	32.3.4.2 Transfer , changed	
		942	32.3.4.3 Endpoint Number, changed	
		943	32.3.4.4 Maximum Packet Size Setting, changed	
		943	32.3.4.5 Transaction Counter (For Pipes 1 to 5 in Reading Direction), title changed	
		945	32.3.4.9 Auto Response Mode, changed	
		946, 947	32.3.5.1 FIFO Buffer Memory, changed	
		947	32.3.5.2 FIFO Buffer Clearing, changed	
		949	32.3.5.4 DMA Transfers (D0FIFO and D1FIFO Ports) (1) Overview of DMA Transfers, changed	

Downloaded from Arrow.com.

Rev.	Date	Description		Classification
		Page	Summary	
1.10	Mar 30, 2021	952	32.3.7 Bulk Transfers (Pipes 1 to 5), changed	
		952	32.3.8 Interrupt Transfers (Pipes 6 to 9), title changed	
		953	32.3.9 Isochronous Transfers (Pipes 1 and 2), title changed	
		962	32.3.11.2 Transfer Schedule, changed	
		963	32.4.1 Setting the Module-Stop Function, changed	
		33. Serial Communications Interface (SCIg, SCIh)		
		971	Table 33.2 SCIh Specifications (2/2), changed	
		985, 986	33.2.8 Serial Control Register (SCR) (1) Non-Smart Card Interface Mode (SCMR.SMIF = 0), changed	
		994, 995	33.2.10 Smart Card Mode Register (SCMR), changed	
		1028	33.3.2 Receive Data Sampling Timing and Reception Margin in Asynchronous Mode, Note 1, changed	
		1031, 1032	33.3.6 SCI Initialization (Asynchronous Mode), changed	
		1031	Figure 33.8 Sample SCI Initialization Flowchart (Asynchronous Mode), changed	
		1037 to 1040	33.3.8 Serial Data Reception (Asynchronous Mode), changed	
		1038	Table 33.28 Status Flags in the SSR Register and Receive Data Handling, changed	
		1047	33.5.2 CTS and RTS Functions, changed	
		1048	33.5.3 SCI Initialization (Clock Synchronous Mode), changed	
		1048	Figure 33.24 Example of SCI Initialization Flowchart (Clock Synchronous Mode), changed	
		1053 to 1055	33.5.5 Serial Data Reception (Clock Synchronous Mode), changed	
		1055	Figure 33.31 Example Flowchart of Serial Reception in Clock Synchronous Mode, changed	
		1056	Figure 33.32 Example Flowchart of Simultaneous Serial Transmission and Reception in Clock Synchronous Mode, changed	
		1059	33.6.3 Block Transfer Mode, changed	
		1061	Figure 33.38 Example of SCI Initialization Flowchart (Smart Card Interface Mode), changed	
		1063 to 1065	33.6.6 Serial Data Transmission (Except in Block Transfer Mode), changed	
		1066, 1067	33.6.7 Serial Data Reception (Except in Block Transfer Mode), changed	
		1074	Figure 33.51 Example of the Flowchart of SCI Initialization (for Simple I2C Mode), changed	
		1076	Figure 33.54 Example of the Procedure for Master Transmission Operations in Simple I2C Mode (with Transmission Interrupts and Reception Interrupts in Use), changed	
		1083	33.8.5 SCI Initialization (Simple SPI Mode), changed	
		1100	Figure 33.75 Block Diagram of Digital Noise Filter, title changed	
		1101	33.12.2 Interrupts in Asynchronous Mode, Clock Synchronous Mode, and Simple SPI Mode, changed	
		1113	Figure 33.82 Example of Flowchart for Receive Error Handling (during Reception of the Start Frame), changed	
		35. I2C-bus Interface (RIICa)		
		1121 to 1123	35.1 Overview, changed	
		1121	Table 35.1 RIIC Specifications (1/2), changed	
		1122	Table 35.1 RIIC Specifications (2/2), changed	
		1122	Figure 35.1 RIIC Block Diagram, changed	
		1123	Figure 35.2 I/O Pin Connection to the External Circuit (I2C-bus Configuration Example), changed	
		1124, 1125	35.2.1 I2C-bus Control Register 1 (ICCR1), changed	
		1126 to 1129	35.2.2 I2C-bus Control Register 2 (ICCR2), changed	
		1130	35.2.3 I2C-bus Mode Register 1 (ICMR1), changed	
		1131, 1132	35.2.4 I2C-bus Mode Register 2 (ICMR2), changed	
		1133, 1134	35.2.5 I2C-bus Mode Register 3 (ICMR3), changed	
		1135, 1136	35.2.6 I2C-bus Function Enable Register (ICFER), changed	TN-RX*-A0227A/E
		1141 to 1143	35.2.9 I2C-bus Status Register 1 (ICSR1), changed	
		1144 to 1146	35.2.10 I2C-bus Status Register 2 (ICSR2), changed	TN-RX*-A0227A/E
		1145	Table 35.4 Relationship between Arbitration-Lost Generation Sources and Arbitration-Lost Enable Functions, changed	

Downloaded from Arrow.com.

Rev.	Date	Description		Classification
		Page	Summary	
1.10	Mar 30, 2021	1147	35.2.11 Slave Address Register Ly (SARLy) (y = 0 to 2), changed	
		1148	35.2.12 Slave Address Register Uy (SARUy) (y = 0 to 2), changed	
		1149	35.2.13 I2C-bus Bit Rate Low-Level Register (ICBRL), changed	
		1150, 1151	35.2.14 I2C-bus Bit Rate High-Level Register (ICBRH), changed	
		1151	Table 35.5 Examples of ICBRH/ICBRL Settings for Transfer Rate Note, changed	
		1152	35.2.16 I2C-bus Receive Data Register (ICDRR), changed	
		1153	35.3.1 Communication Data Format, changed	
		1153	Figure 35.4 I2C-bus Timing (SLA = 7 Bits), changed	
		1155 to 1158	35.3.3 Master Transmit Operation, changed	
		1156	Figure 35.6 Example of Master Transmission Flowchart, changed	
		1158 to 1163	35.3.4 Master Receive Operation, changed	
		1160	Figure 35.10 Example of Master Reception (7-Bit Address Format, 1 or 2 bytes), changed	
		1161	Figure 35.11 Example of Master Reception (7-Bit Address Format, 3 Bytes or More), changed	
		1164 to 1166	35.3.5 Slave Transmit Operation, changed	
		1165	Figure 35.15 Example of Slave Transmission, changed	
		1167, 1168	35.3.6 Slave Receive Operation, changed	
		1169	35.4 SCL Synchronization Circuit, changed	
		1169	Figure 35.21 Generation and Synchronization of the SCL Signal from the RIIC, changed	
		1170	35.5 SDA Output Delay Function, changed	
		1170	Figure 35.22 SDA Output Delay Function, changed	
		1171	35.6 Digital Noise Filters, changed	
		1171	Figure 35.23 Block Diagram of the Digital Noise Filter, changed	
		1172, 1173	35.7.1 Slave-Address Match , changed	
		1173	Figure 35.25 AASy Flag Set Timing with 10-Bit Address Format Selected, changed	
		1174	35.7.2 Detection of the General Call Address, changed	
		1175, 1176	35.7.3 Device-ID Address Detection, changed	TN-RX*-A0227A/E
		1176	Figure 35.28 Set/Clear Timing of the AASy and DID Flags during Reception of Device-ID Address, changed	
		1177	35.7.4 Host Address Detection, changed	
		1178	35.8.1 Function to Prevent Wrong Transmission of Transmit Data, changed	
		1179	35.8.2 NACK Reception Transfer Suspension Function, changed	TN-RX*-A0227A/E
		1179	Figure 35.31 Suspension of Data Transmission When NACK is Received (NACKE = 1), changed	
		1180, 1181	35.8.3 Function to Prevent Failure to Receive Data, changed	
		1182	35.9 Arbitration-Lost Detection Functions, changed	
		1182, 1183	35.9.1 Master Arbitration-Lost Detection (MALE Bit), changed	
		1183	Figure 35.34 Arbitration-Lost When a Start Condition is Generated (MALE = 1), title changed	
		1184, 1185	35.9.2 Function to Detect Loss of Arbitration during NACK Transmission (NALE Bit), changed	
		1185	35.9.3 Slave Arbitration-Lost Detection (SALE Bit), changed	
		1186, 1187	35.10 Start Condition/Restart Condition/Stop Condition Generating Function, changed	
		1188	35.11 Bus Hanging, changed	
		1188, 1189	35.11.1 Timeout Function, changed	
		1189	Figure 35.39 Timeout Function, changed	
		1190	35.11.2 Additional SCL Output Function, changed	TN-RX*-A0227A/E
		1190	Conditions for using the ICCR1.CLO bit, changed	
		1190	Figure 35.40 Additional SCL Output Function (CLO Bit), changed	TN-RX*-A0227A/E
		1191	35.11.3 RIIC Reset and Internal Reset, changed	
		1192	35.12 SMBus Operation, changed	
		1192, 1193	35.12.1 SMBus Timeout Measurement, changed	
		1193	Figure 35.41 SMBus Timeout Measurement, changed	
		1193	35.12.2 Packet Error Code (PEC), changed	

Downloaded from Arrow.com.

Rev.	Date	Description		Classification
		Page	Summary	
1.10	Mar 30, 2021	1194	35.13 Interrupt Sources, changed	
		1195	35.14 Initialization of Registers and Functions When a Reset is Applied or a Condition is Detectedhe, changed	
		1195	Table 35.7 Reset States of Registers and Functions When a Reset is Applied or a Condition is Detected, changed	TN-RX*-A0227A/E
		38. Serial Peripheral Interface (RSPIa)		
		1383	Figure 38.22 RSPI Transfer Format (CPHA = 0), changed	
		1391	Figure 38.28 Clock Stop Waveform When a Serial Transfer Continues While the Receive Buffer is Full in Master Mode (CPHA = 1), changed	
		1391	Figure 38.29 Clock Stop Waveform When a Serial Transfer Continues While the Receive Buffer is Full in Master Mode (CPHA = 0), changed	
		1412	Figure 38.46 Example of Initialization Flowchart in Master Mode (Clock Synchronous Operation), changed	
		41. Bluetooth Low Energy (BLE)		
		1481 to 1484	41.1 Overview, changed	
		1481	Table 41.1 Specifications of the BLE, changed	
		1483	Figure 41.2 Example of the External Connection Circuit (83-Pin LGA), added	
		1487	41.4 Certificates of Compliance, added	
		1490	41.5.1 RF Transceiver Power-Supply, changed	
		1491	Figure 41.6 Example of the BLE Externally Connected Circuit (83-pin LGA), changed	
		1491	41.5.2 Wireless Standards, changed	
		43. Capacitive Touch Sensing Unit (CTSU)		
		1511	43.2.7 CTSU Channel Enable Control Register 0 (CTSUCHAC0), changed	
		1512	43.2.8 CTSU Channel Enable Control Register 1 (CTSUCHAC1), changed	
		1513	43.2.9 CTSU Channel Enable Control Register 2 (CTSUCHAC2), changed	
		1514	43.2.10 CTSU Channel Enable Control Register 3 (CTSUCHAC3), changed	
		1515	43.2.11 CTSU Channel Enable Control Register 4 (CTSUCHAC4), changed	
		1516	43.2.12 CTSU Channel Transmit/Receive Control Register 0 (CTSUCHTRC0), changed	
		1517	43.2.13 CTSU Channel Transmit/Receive Control Register 1 (CTSUCHTRC1), changed	
		1518	43.2.14 CTSU Channel Transmit/Receive Control Register 2 (CTSUCHTRC2), changed	
		1519	43.2.15 CTSU Channel Transmit/Receive Control Register 3 (CTSUCHTRC3), changed	
		1520	43.2.16 CTSU Channel Transmit/Receive Control Register 4 (CTSUCHTRC4), changed	
		1526, 1527	43.2.21 CTSU Sensor Offset Register 1 (CTSUSO1), changed	
		1534	Figure 43.9 CTSU Initial Setting Flowchart, changed	
		1534	Figure 43.10 CTSU Stopping Flowchart, changed	
		1535	43.3.2.2 Status Counter, changed	
		1536	Figure 43.12 Software Flowchart and Operation Example of Self-Capacitance Single Scan Mode, changed	
		1538, 1539	43.3.2.4 Self-Capacitance Multi-Scan Mode Operation, changed	
		1538	Figure 43.14 Software Flow and Operation Example of Self-Capacitance Multi-Scan Mode, changed	
		1540 to 1542	43.3.2.5 Mutual Capacitance Full Scan Mode Operation, changed	
		1540	Figure 43.16 Software Flowchart and Operation Example of Mutual Capacitance Full Scan Mode, changed	
		46. Temperature Sensor (TEMPSA)		
		1650	46.2.1 Temperature Sensor Calibration Data Register (TSCDRH, TSCDRL), changed	
		1651, 1652	46.3.1 Before Using the Temperature Sensor, changed	
		48. Data Operation Circuit (DOC)		
		All	Full review	
		50. Flash Memory (FLASH)		
		1677	Table 50.1 Flash Memory Specifications, changed	TN-RX*-A0245A/E
		1692	50.4.15 Flash Write Buffer Register n (FWBn) (n = 0 to 3), changed	

Downloaded from Arrow.com.

Rev.	Date	Description		Classification
		Page	Summary	
1.10	Mar 30, 2021	—	50.9.2 ROM Code Protection, deleted	TN-RX*-A0245A/E
		51. Electrical Characteristics		
		1786	Table 51.24 Clock Timing Note 6, changed	TN-RX*-A0245A/E
		1795 to 1815	51.3.5 Timing of On-Chip Peripheral Modules, Layout changed	
		Appendix 2. Package		
		1843	Figure B 83-Pin HWQFN (PTLG0083KA-A), added	
1.20	Apr 20, 2023	Features		
		51	Package, added	
		1. Overview		
		53	Table 1.1 Outline of Specifications (2/5), changed	TN-RX*-A0265A/E
		56	Table 1.1 Outline of Specifications (5/5), changed	
		58	Table 1.3 List of Products: D Version (T_a = –40 to +85°C), changed	
		66	Figure 1.5 Pin Assignments of the 83-Pin LGA, changed	
		8. Voltage Detection Circuit (LVDAb)		
		161	Table 8.1 LVD Specifications, changed	
		15. Interrupt Controller (ICUb)		
		276	Table 15.1 Specifications of Interrupt Controller, changed	
		295	15.3.1 Interrupt Vector Table, changed	
		296 to 301	Table 15.3 Interrupt Vector Table, changed	
		306	15.4.3 Selecting Interrupt Request Destinations (2) DTC Trigger, changed	
		311	15.6 Return from Power-Down States 15.6.1 Return from Sleep Mode or Deep Sleep Mode, changed	
		311	15.6.2 Return from Software Standby Mode, changed	
		20. Event Link Controller (ELC)		
		420	20.2.2 Event Link Setting Register n (ELSRn) (n = 1 to 4, 7, 8, 10, 12, 14 to 16, 18 to 29), changed	
		425	Table 20.4 Registers Related to Port Groups and Corresponding Port Numbers, changed	
		432	20.3.6 I/O Port Operation When Event Signal is Input and Event Generation (1) Single Ports and Port Groups, changed	
		436	20.3.6 I/O Port Operation When Event Signal is Input and Event Generation (8) Restrictions on Writing to PODR and PDBF Registers, changed	
		437	20.4.1 Setting ELSRn Register (1) Setting ELSR8 Register, changed	
		21. I/O Ports		
		439	Table 21.2 Port Functions, changed	TN-RX*-A0265A/E
		28. Realtime Clock (RTCe)		
		771	Figure 28.1 Block Diagram of RTC (n = 0, 1), changed	
		804	Figure 28.7 Using Alarm Function, changed	
		29. Low-Power Timer (LPT)		
		819	29.2.4 Low-Power Timer Period Setting Register (LPTPRD), changed	
		827	29.4 Wakeup from Software Standby Mode by an Interrupt through the Event Link Controller (ELC), changed	
		30. Watchdog Timer (WDTA)		
		828	Table 30.1 WDT Specifications, changed	
		31. Independent Watchdog Timer (IWDTa)		
		842	Table 31.1 IWDT Specifications, changed	
		848	31.2.3 IWDT Status Register (IWDTSR), changed	
		32. USB 2.0 Host/Function Module (USBc)		
		861	Table 32.1 USB Specifications, changed	
		871, 872	32.2.5 CFIFO Port Select Register (CFIFOSEL), D0FIFO Port Select Register (D0FIFOSEL), D1FIFO Port Select Register (D1FIFOSEL) • CFIFOSEL, changed	
		873, 874	• D0FIFOSEL, D1FIFOSEL, changed	
		904	32.2.28 Pipe Maximum Packet Size Register (PIPEMAXP), changed	
		921	32.3.1.4 Example of USB External Connection Circuit, changed	
		921	Figure 32.2 Example of Connection between Self-Powered OTG Device and USB Connector, changed	

Downloaded from Arrow.com.

Rev.	Date	Description		Classification
		Page	Summary	
1.20	Apr 20, 2023	922	Figure 32.3 Example of Connection between Self-Powered Function and USB Connector, changed	
		923	Figure 32.4 Example of Connection between Function and USB Connector When Battery Charging Specification Rev.1.2 is Supported, changed	
		924	Figure 32.5 Example of Connection between Host and USB Connector, changed	
		925	Figure 32.6 Example of Connection between Bus-Powered Function and USB Connector, changed	
		947	Table 32.19 FIFO Port Function Settings, changed	
		33. Serial Communications Interface (SCIg, SCIh)		
		1026	Table 33.27 Serial Transfer Formats (Asynchronous Mode), changed	
		1047	Figure 33.24 Example of SCI Initialization Flowchart (Clock Synchronous Mode), changed	
		1049	Figure 33.25 Example of Serial Data Transmission in Clock Synchronous Mode When the CTS Function is Not Used at the Beginning of Transmission, changed	
		1049	Figure 33.26 Example of Serial Data Transmission in Clock Synchronous Mode When the CTS Function is Used at the Beginning of Transmission, changed	
		1082	33.9 Bit Rate Modulation Function, changed	
		1113	33.14.14 Note on Transmit Enable Bit (TE Bit), changed	
		35. I^2C-bus Interface (RIICa)		
		1120	35.1 Overview	
		1120, 1121	Table 35.1 RIIC Specifications, changed	
		—	Figure 35.2 I/O Pin Connection to the External Circuit (I^2C-bus Configuration Example), deleted	
		1123, 1124	35.2.1 I^2C-bus Control Register 1 (ICCR1), changed	
		1138, 1139	35.2.8 I^2C-bus Interrupt Enable Register (ICIER), changed	
		1143 to 1145	35.2.10 I^2C-bus Status Register 2 (ICSR2), changed	
		1151	35.2.17 I^2C-bus Shift Register (ICDRS), changed	
		1154	35.3.3 Master Transmit Operation, changed	
		1155	Figure 35.5 Example of Master Transmission Flowchart, changed	
		1156	Figure 35.7 Master Transmit Operation Timing (2) (10-Bit Address Format), changed	
		1161	Figure 35.12 Master Receive Operation Timing (2) (10-Bit Address Format, When RDRFS bit is 0), changed	
		1162	Figure 35.13 Master Receive Operation Timing (3) (When RDRFS bit is 0), changed	
		1166	35.3.6 Slave Receive Operation, changed	
		1169	Figure 35.21 SDA Output Delay Function, changed	
		1174	35.7.3 Device-ID Address Detection, changed	
		1181	35.9.1 Master Arbitration-Lost Detection (MALE Bit), changed	
		1183, 1184	35.9.2 NACK Transmission Arbitration-Lost Detection Function (NALE Bit), changed	
		1184	35.9.3 Slave Arbitration-Lost Detection (SALE Bit), changed	
		1191	35.12.1 SMBus Timeout Measurement (2) Measuring timeout of master device, changed	
		1193	35.13 Interrupt Sources, changed	
		1193	Table 35.6 Interrupt Sources, changed	
		1193	35.13.1 Buffer Operation for TXI and RXI Interrupts, changed	
		1195	35.15 Event Link Function (Output), changed	
		1195	35.15.1 Interrupt Handling and Event Linking, changed	
		43. Capacitive Touch Sensing Unit (CTSU)		
		1528, 1529	43.2.24 CTSU Error Status Register (CTSUERRS), changed	TN-RX*-A0262A/E
		44. 12-Bit A/D Converter (S12ADE)		
		All	PCLK → PCLKB	
		1549	Table 44.2 Functions of 12-Bit A/D Converter, changed	
		1550	Table 44.3 Pin Configuration of 12-Bit A/D Converter, changed	
		1561	44.2.8 A/D-Converted Value Addition/Average Function Select Register 0 (ADADS0), changed	

Downloaded from Arrow.com.

Rev.	Date	Description		Classification
		Page	Summary	
1.20	Apr 20, 2023	1562	44.2.9 A/D-Converted Value Addition/Average Function Select Register 1 (ADADS1), changed	
		1567	44.2.12 A/D Conversion Start Trigger Select Register (ADSTRGR, changed	
		1571	44.2.14 A/D Sampling State Register n (ADSSTRn) (n = 0 to 7, L, T, O), changed	
		1577	44.2.19 A/D Compare Function Window A Channel Select Register 0 (ADCMPANSR0), changed	
		1578	44.2.20 A/D Compare Function Window A Channel Select Register 1 (ADCMPANSR1), changed	
		1581	Figure 44.3 Explanation of Compare Function Window A Comparison Conditions, changed	
		1589	44.2.28 A/D Compare Function Window A Channel Status Register 1 (ADCMPSR1), changed	
		1594	Figure 44.4 Explanation of Compare Function Window B Compare Conditions, changed	
		1605	44.3.2.3 A/D Conversion of Temperature Sensor Output/Internal Reference Voltage, changed	
		1621	44.3.5.1 Compare Function Window A/B, changed	
		1623	44.3.5.2 ELC Output of Compare Function, changed	
		1627	Table 44.10 Times for Conversion during Scanning (in Numbers of Cycles of ADCLK and PCLKB), changed	
		1632	44.6 Selecting Reference Voltage, changed	
		1633	44.7 Allowable Impedance of Signal Source, changed	
		1633	Figure 44.28 Equivalent Circuit of Analog Input Pin and External Sensor, changed	
		1635	44.8.3 A/D Conversion Restarting Timing and Termination Timing, changed	
		1637	44.8.11 Voltage Range of Analog Power Supply Pins, changed	
		45. 12-Bit D/A Converter (R12DAA)		
		1640	Table 45.2 Pin Configuration of 12-Bit D/A Converter, changed	
		46. Temperature Sensor (TEMPSA)		
		1650	46.1 Overview, changed	
		1651	46.2.1 Temperature Sensor Calibration Data Register (TSCDRH, TSCDRL), changed	
		1652	46.3 Using the Temperature Sensor, changed	
		1652, 1653	46.3.1 Preparation for Using the Temperature Sensor, changed	
		47. Comparator B (CMPBa)		
		1658	47.2.3 Comparator B1 Flag Register (CPB1FLG), changed	
		1660	47.2.5 Comparator B1 Filter Select Register (CPB1F), changed	
		1660	47.2.6 Comparator B1 Mode Select Register (CPB1MD), changed	
		1661	47.2.7 Comparator B1 Reference Input Voltage Select Register (CPB1REF), changed	
		1662	47.2.8 Comparator B1 Output Control Register (CPB1OCR), changed	
		50. Flash Memory (FLASH)		
		1678	50.1 Overview, changed	
		1679	50.2 ROM Area and Block Configuration, changed	
		1680	50.3 E2 DataFlash Area and Block Configuration, changed	
		1681	50.4.1 E2 DataFlash Control Register (DFLCTL), changed	
		1688, 1689	50.4.9 Flash Control Register (FCR), changed	
		1690, 1691	50.4.10 Flash Extra Area Control Register (FEXCR), changed	
		1698	50.4.20 Flash Start-Up Setting Monitor Register (FSCMR), changed	
		1700	50.5 Start-Up Program Protection, changed	
		1701	50.6 Area Protection, changed	
		1703	50.7.2.1 Transition from E2 DataFlash Access Disable Mode to Read Mode, changed	
		1703	Figure 50.7 Procedure for Transition from E2 DataFlash Access Disabled Mode to ROM/E2 DataFlash Read Mode, changed	
		1704	50.7.2.2 Transition from Read Mode to P/E Mode, changed	
		1706	50.7.2.3 Transition from P/E Mode to Read Mode, changed	
		1717	Figure 50.20 Procedure to Issue the Start-Up Area Information Program Command/Access Window Information Program Command, changed	

Downloaded from Arrow.com.

Rev.	Date	Description		Classification
		Page	Summary	
1.20	Apr 20, 2023	1719	50.8 Boot Mode, changed	
		1720	50.8.1.1 Operating Conditions in Boot Mode (USB Interface), changed	
		1722	Table 50.9 Pin Handling in Boot Mode (USB Interface) in Bus-Powered, changed	
		1723	50.8.2.1 Operating Conditions in Boot Mode (SCI Interface), changed	
		1725	50.8.3.1 Operating Conditions in Boot Mode (FINE Interface), changed	
		1726	50.9.1 ID Code Protection, changed	
		1729, 1730	50.10.1 State Transition in Boot Mode (SCI Interface), changed	
		1731	Table 50.15 Information Regarding the States, changed	
		1732	50.10.5.1 Supported Device Inquiry, changed	
		1735	50.10.6 Setting Commands, changed	
		1739	50.10.8.1 User/Data Area Program Preparation, changed	
		1741	50.10.8.3 Data Area Program, changed	
		1742	50.10.8.5 Block Erase, changed	
		1743, 1744	50.10.9.1 Memory Read, changed	
		1748 to 1758	50.11 Serial Programmer Operation in Boot Mode (SCI Interface), changed	
		51. Electrical Characteristics		
		1762	Table 51.1 Absolute Maximum Ratings, changed	TN-RX*-A0265A/E
		1764, 1765	Table 51.3 DC Characteristics (1), changed	TN-RX*-A0265A/E
		1766, 1767	Table 51.4 DC Characteristics (2), changed	TN-RX*-A0265A/E
		Appendix 2. Package Dimensions		
		1846, 1847	Figure B 83-Pin LGA (PTLG0083KA-A), changed	
		1848, 1849	Figure C 83-Pin LGA (PTLG0083KA-B), added	

Downloaded from Arrow.com.

RX23W Group User's Manual: Hardware

Publication Date: Rev.1.00 Jul 31, 2019
Rev.1.20 Apr 20, 2023

Published by: Renesas Electronics Corporation

Downloaded from Arrow.com.

RX23W Group

Renesas Electronics Corporation

R01UH0823EJ0120

Downloaded from Arrow.com.