User's Manual

16

RL78/G12

User's Manual: Hardware

16-Bit Single-Chip Microcontrollers

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.2.00 Aug 2013

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 of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
3. Renesas Electronics does not assume any liability for infringement of 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. 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 should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from such alteration, modification, copy or otherwise misappropriation of Renesas Electronics product.
5. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The recommended 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; and industrial robots etc.

   “High Quality”: Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; and safety equipment etc.

   Renesas Electronics products are neither intended nor 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 damages (nuclear reactor control systems, military equipment etc.). You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application for which it is not intended. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for which the product is not intended by Renesas Electronics.
6. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, 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, please evaluate the safety of the final products or systems manufactured by you.
8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technology may 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 should not use Renesas Electronics products or technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. When exporting the Renesas Electronics products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, who distributes, disposes of, or otherwise places the product with a third party, to notify such third party in advance of the contents and conditions set forth in this document, Renesas Electronics assumes no responsibility for any losses incurred by you or third parties as a result of unauthorized use of Renesas Electronics products.
11. This document may not be reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(2012.4)

Downloaded from Arrow.com.

NOTES FOR CMOS DEVICES

(1) 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 VIL (MAX) and VIH (MIN) due to noise, etc., 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 VIL (MAX) and VIH (MIN).

(2) HANDLING OF UNUSED INPUT PINS: Unconnected CMOS device inputs can be cause of malfunction. If an input pin is unconnected, it is possible that an internal input level may be generated due to noise, etc., causing malfunction. CMOS devices behave differently than Bipolar or NMOS devices. Input levels of CMOS devices must be fixed high or low by using pull-up or pull-down circuitry. Each unused pin should be connected to VDD or GND via a resistor if there is a possibility that it will be an output pin. All handling related to unused pins must be judged separately for each device and according to related specifications governing the device.

(3) PRECAUTION AGAINST ESD: A strong electric field, when exposed to a MOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop generation of static electricity as much as possible, and quickly dissipate it when it has occurred. Environmental control must be adequate. When it is dry, a humidifier should be used. It is recommended to avoid using insulators that 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 should be grounded. The operator should be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions need to be taken for PW boards with mounted semiconductor devices.

(4) STATUS BEFORE INITIALIZATION: Power-on does not necessarily define the initial status of a MOS device. Immediately after the power source is turned ON, devices with reset functions have not yet been initialized. Hence, power-on does not guarantee output pin levels, I/O settings or contents of registers. A device is not initialized until the reset signal is received. A reset operation must be executed immediately after power-on for devices with reset functions.

(5) POWER ON/OFF SEQUENCE: In the case of a device that uses different power supplies for the internal operation and external interface, as a rule, switch on the external power supply after switching on the internal power supply. When switching the power supply off, as a rule, switch off the external power supply and then the internal power supply. Use of the reverse power on/off sequences may result in the application of an overvoltage to the internal elements of the device, causing malfunction and degradation of internal elements due to the passage of an abnormal current. The correct power on/off sequence must be judged separately for each device and according to related specifications governing the device.

(6) INPUT OF SIGNAL DURING POWER OFF STATE : Do not input signals or an I/O pull-up power supply while the device is not powered. 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. Input of signals during the power off state must be judged separately for each device and according to related specifications governing the device.

Downloaded from Arrow.com.

How to Use This Manual

Readers

This manual is intended for user engineers who wish to understand the functions of the RL78/G12 and design and develop application systems and programs for these devices.

Purpose

This manual is intended to give users an understanding of the functions described in the **Organization** below.

Organization

The RL78/G12 manual is separated into two parts: this manual and the software edition (common to the RL78/G12 family).

RL78/G12 **User's Manual** **Hardware**	**RL78 family** **User's Manual** **Software**
• Pin functions • Internal block functions • Interrupts • Other on-chip peripheral functions • Electrical specifications	• CPU functions • Instruction set • Explanation of each instruction

How to Read This Manual

It is assumed that the readers of this manual have general knowledge of electrical engineering, logic circuits, and microcontrollers.

- To gain a general understanding of functions:
  → Read this manual in the order of the **CONTENTS**. The mark "<R>" shows major revised points. The revised points can be easily searched by copying an "<R>" in the PDF file and specifying it in the "Find what:" field.
- How to interpret the register format:
  → For a bit number enclosed in angle brackets, the bit name is defined as a reserved word in the assembler, and is defined as an sfr variable using the #pragma sfr directive in the compiler.
- To know details of the RL78/G12 Microcontroller instructions:
  → Refer to the separate document **RL78 Family User's Manual: Software (R01US0015E)**.

Conventions

Data significance:	Higher digits on the left and lower digits on the right	
Active low representations:	$\overline{\times\times\times}$ (overscore over pin and signal name)	
Note:	Footnote for item marked with **Note** in the text	
Caution:	Information requiring particular attention	
Remark:	Supplementary information	
Numerical representations:	Binary	···×××× or ××××B
	Decimal	···××××
	Hexadecimal	···××××H

Related Documents

The related documents indicated in this publication may include preliminary versions. However, preliminary versions are not marked as such.

Downloaded from Arrow.com.

Documents Related to Devices

Document Name	Document No.
RL78/G12 User's Manual: Hardware	R01UH0200E
RL78 Family User's Manual: Software	R01US0015E

Documents Related to Flash Memory Programming

Document Name	Document No.
PG-FP5 Flash Memory Programmer User's Manual	R20UT0008E

Caution The related documents listed above are subject to change without notice. Be sure to use the latest version of each document when designing.

Other Documents

Document Name	Document No.
Renesas MPUs & MCUs RL78 Family	R01CP0003E
Semiconductor Package Mount Manual	**Note**
Quality Grades on NEC Semiconductor Devices	C11531E
Guide to Prevent Damage for Semiconductor Devices by Electrostatic Discharge (ESD)	C11892E
Semiconductor Reliability Handbook	R51ZZ0001E

Note See the "Semiconductor Package Mount Manual" website (http://www.renesas.com/products/package/manual/index.jsp).

Caution The related documents listed above are subject to change without notice. Be sure to use the latest version of each document when designing.

All trademarks and registered trademarks are the property of their respective owners.
EEPROM is a trademark of Renesas Electronics Corporation.
Windows, Windows NT and Windows XP are registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.
PC/AT is a trademark of International Business Machines Corporation.
SuperFlash is a registered trademark of Silicon Storage Technology, Inc. in several countries including the United States and Japan.

Caution: This product uses SuperFlash® technology licensed from Silicon Storage Technology, Inc.

Downloaded from Arrow.com.

CONTENTS

CHAPTER 1 OUTLINE 1

1.1 Differences between the R5F102 Products and the R5F103 Products 1

1.1.1 Data Flash 1

1.1.2 On-chip oscillator characteristics 2

1.1.3 Peripheral Functions 2

1.2 Features 3

1.3 List of Part Numbers 6

1.4 Pin Configuration (Top View) 8

1.4.1 20-pin products 8

1.4.2 24-pin products 9

1.4.3 30-pin products 10

1.5 Pin Identification 11

1.6 Block Diagram 12

1.6.1 20-pin products 12

1.6.2 24-pin products 13

1.6.3 30-pin products 14

1.7 Outline of Functions 15

CHAPTER 2 PIN FUNCTIONS 17

2.1 Port Functions 17

2.1.1 20-pin products 17

2.1.2 24-pin products 18

2.1.3 30-pin products 20

2.2 Functions other than port pins 22

2.2.1 Functions for each product 22

2.2.2 Description of Functions 23

2.3 Connection of Unused Pins 24

2.4 Block Diagrams of Pins 25

CHAPTER 3 CPU ARCHITECTURE 37

3.1 Memory Space 37

3.1.1 Internal program memory space 45

3.1.2 Mirror area 49

3.1.3 Internal data memory space 50

3.1.4 Special function register (SFR) area 52

3.1.5 Extended special function register (2nd SFR: 2nd Special Function Register) area 52

3.1.6 Data memory addressing 53

Downloaded from Arrow.com.

3.2 Processor Registers 54
3.2.1 Control registers 54
3.2.2 General-purpose registers 56
3.2.3 ES and CS registers 57
3.2.4 Special function registers (SFRs) 58
3.2.5 Extended special function registers (2nd SFRs: 2nd Special Function Registers) 63
3.3 Instruction Address Addressing 69
3.3.1 Relative addressing 69
3.3.2 Immediate addressing 69
3.3.3 Table indirect addressing 70
3.3.4 Register direct addressing 70
3.4 Addressing for Processing Data Addresses 71
3.4.1 Implied addressing 71
3.4.2 Register addressing 71
3.4.3 Direct addressing 72
3.4.4 Short direct addressing 73
3.4.5 SFR addressing 74
3.4.6 Register indirect addressing 75
3.4.7 Based addressing 76
3.4.8 Based indexed addressing 79
3.4.9 Stack addressing 80

CHAPTER 4 PORT FUNCTIONS 83

4.1 Port Functions 83
4.2 Port Configuration 83
4.2.1 20, 24-pin products 84
4.2.1.1 Port 0 84
4.2.1.2 Port 1 84
4.2.1.3 Port 2 84
4.2.1.4 Port 4 85
4.2.1.5 Port 6 85
4.2.1.6 Port 12 85
4.2.1.7 Port 13 85
4.2.2 30-pin products 85
4.2.2.1 Port 0 85
4.2.2.2 Port 1 86
4.2.2.3 Port 2 86
4.2.2.4 Port 3 86
4.2.2.5 Port 4 86
4.2.2.6 Port 5 87
4.2.2.7 Port 6 87
4.2.2.8 Port 12 87
4.2.2.9 Port 13 87
4.2.2.10 Port 14 87

Downloaded from Arrow.com.

4.3 Registers Controlling Port Function 88
4.3.1 Port mode registers (PMxx) 90
4.3.2 Port registers (Pxx) 91
4.3.3 Pull-up resistor option registers (PUxx) 93
4.3.4 Port input mode register (PIMx) 94
4.3.5 Port output mode registers (POMx) 95
4.3.6 Port mode control registers (PMCxx) 96
4.3.7 A/D port configuration register (ADPC) 97
4.3.8 Peripheral I/O redirection register (PIOR) 97
4.4 Port Function Operations 99
4.4.1 Writing to I/O port 99
4.4.2 Reading from I/O port 99
4.4.3 Operations on I/O port 99
4.4.4 Handling different potentials (1.8 V, 2.5 V, and 3 V) by using I/O buffers 100
4.5 Register Settings When Using Alternate Function 102
4.5.1 Basic concept when using alternate function 102
4.5.2 Register settings for alternate function whose output function is not used 103
4.5.3 Register setting examples for using the port and alternate functions 104
4.6 Cautions When Using Port Function 112
4.6.1 Cautions on 1-Bit Manipulation Instruction for Port Register n (Pn) 112
4.6.2 Notes on specifying the pin settings 113

CHAPTER 5 CLOCK GENERATOR 114

5.1 Functions of Clock Generator 114
5.2 Configuration of Clock Generator 115
5.3 Registers Controlling Clock Generator 117
5.3.1 Clock operation mode control register (CMC) 118
5.3.2 System clock control register (CKC) 119
5.3.3 Clock operation status control register (CSC) 120
5.3.4 Oscillation stabilization time counter status register (OSTC) 121
5.3.5 Oscillation stabilization time select register (OSTS) 123
5.3.6 Peripheral enable register 0 (PER0) 125
5.3.7 Operation speed mode control register (OSMC) 126
5.3.8 High-speed on-chip oscillator frequency selection register (HOCODIV) 127
5.3.9 High-speed on-chip oscillator trimming register (HIOTRM) 128
5.4 System Clock Oscillator 129
5.4.1 X1 oscillator 129
5.4.2 High-speed on-chip oscillator 132
5.4.3 Low-speed on-chip oscillator 132
5.5 Clock Generator Operation 132
5.6 Controlling Clock 134

Downloaded from Arrow.com.

5.6.1 Example of setting high-speed on-chip oscillator 134
5.6.2 Example of setting X1 oscillation clock 135
5.6.3 CPU clock status transition diagram 136
5.6.4 Condition before changing CPU clock and processing after changing CPU clock 139
5.6.5 Time required for switchover of CPU clock and main system clock 140
5.6.6 Conditions before clock oscillation is stopped 140
5.7 Resonator and Oscillator Constants 141

CHAPTER 6 TIMER ARRAY UNIT 143

6.1 Functions of Timer Array Unit 145
6.1.1 Independent channel operation function 145
6.1.2 Simultaneous channel operation function 146
6.1.3 8-bit timer operation function (channels 1 and 3 only) 147
6.2 Configuration of Timer Array Unit 148
6.2.1 Timer/counter register 0n (TCR0n) 154
6.2.2 Timer data register 0n (TDR0n) 156
6.3 Registers Controlling Timer Array Unit 157
6.3.1 Peripheral enable register 0 (PER0) 158
6.3.2 Timer clock select register 0 (TPS0) 159
6.3.3 Timer mode register 0n (TMR0n) 161
6.3.4 Timer status register 0n (TSR0n) 166
6.3.5 Timer channel enable status register 0 (TE0) 167
6.3.6 Timer channel start register 0 (TS0) 168
6.3.7 Timer channel stop register 0 (TT0) 169
6.3.8 Timer input select register 0 (TIS0) 170
6.3.9 Timer output enable register 0 (TOE0) 171
6.3.10 Timer output register 0 (TO0) 172
6.3.11 Timer output level register 0 (TOL0) 173
6.3.12 Timer output mode register 0 (TOM0) 174
6.3.13 Noise filter enable register 1 (NFEN1) 175
6.3.14 Registers controlling port functions of pins to be used for timer I/O 176
6.4 Basic Rules of Timer Array Unit 177
6.4.1 Basic Rules of Simultaneous Channel Operation Function 177
6.4.2 Basic rules of 8-bit timer operation function (Only Channels 1 and 3) 179
6.5 Operation of Counter 180
6.5.1 Count clock (f_{TCLK}) 180
6.5.2 Start timing of counter 182
6.5.3 Counter Operation 183
6.6 Channel Output (TO0n pin) Control 188
6.6.1 TO0n pin output circuit configuration 188
6.6.2 TO0n Pin Output Setting 189

Downloaded from Arrow.com.

6.6.3 Cautions on Channel Output Operation 190
6.6.4 Collective manipulation of TO0n bit 194
6.6.5 Timer Interrupt and TO0n Pin Output at Operation Start 195
6.7 Timer Input (TI0n) Control 196
6.7.1 TI0n input circuit configuration 196
6.7.2 Noise filter 196
6.7.3 Cautions on channel input operation 197
6.8 Independent Channel Operation Function of Timer Array Unit 198
6.8.1 Operation as interval timer/square wave output 198
6.8.2 Operation as external event counter 203
6.8.3 Operation as frequency divider (channel 0 of 30-pin products only) 208
6.8.4 Operation as input pulse interval measurement 212
6.8.5 Operation as input signal high-/low-level width measurement 217
6.8.6 Operation as delay counter 221
6.9 Simultaneous Channel Operation Function of Timer Array Unit 226
6.9.1 Operation as one-shot pulse output function 226
6.9.2 Operation as PWM function 233
6.9.3 Operation as multiple PWM output function 240
6.10 Cautions When Using Timer Array Unit 247
6.10.1 Cautions When Using Timer output 247

CHAPTER 7 12-BIT INTERVAL TIMER 248

7.1 Functions of 12-bit Interval Timer 248
7.2 Configuration of 12-bit Interval Timer 248
7.3 Registers Controlling 12-bit Interval Timer 248
7.3.1 Peripheral enable register 0 (PER0) 249
7.3.2 Operation speed mode control register (OSMC) 249
7.3.3 Interval timer control register (ITMC) 250
7.4 12-bit Interval Timer Operation 251
7.4.1 12-bit interval timer operation timing 251
7.4.2 Start of count operation and re-enter to HALT/STOP mode after returned from HALT/STOP mode 252

CHAPTER 8 CLOCK OUTPUT/BUZZER OUTPUT CONTROLLER 253

8.1 Functions of Clock Output/Buzzer Output Controller 253
8.2 Configuration of Clock Output/Buzzer Output Controller 254
8.3 Registers Controlling Clock Output/Buzzer Output Controller 254
8.3.1 Clock output select register n (CKSn) 255
8.3.2 Registers controlling port functions of pins to be used for clock or buzzer output 256
8.4 Operations of Clock Output/Buzzer Output Controller 257
8.4.1 Operation as output pin 257

Downloaded from Arrow.com.

8.5 Cautions of Clock Output/Buzzer Output Controller 257

CHAPTER 9 WATCHDOG TIMER 258

9.1 Functions of Watchdog Timer 258
9.2 Configuration of Watchdog Timer 259
9.3 Register Controlling Watchdog Timer 260
9.3.1 Watchdog timer enable register (WDTE) 260
9.4 Operation of Watchdog Timer 261
9.4.1 Controlling operation of watchdog timer 261
9.4.2 Setting overflow time of watchdog timer 262
9.4.3 Setting window open period of watchdog timer 263
9.4.4 Setting watchdog timer interval interrupt 264

CHAPTER 10 A/D CONVERTER 265

10.1 Function of A/D Converter 265
10.2 Configuration of A/D Converter 267
10.3 Registers Controlling A/D Converter 269
10.3.1 Peripheral enable register 0 (PER0) 270
10.3.2 A/D converter mode register 0 (ADM0) 271
10.3.3 A/D converter mode register 1 (ADM1) 280
10.3.4 A/D converter mode register 2 (ADM2) 281
10.3.5 10-bit A/D conversion result register (ADCR) 284
10.3.6 8-bit A/D conversion result register (ADCRH) 284
10.3.7 Analog input channel specification register (ADS) 285
10.3.8 Conversion result comparison upper limit setting register (ADUL) 287
10.3.9 Conversion result comparison lower limit setting register (ADLL) 287
10.3.10 A/D test register (ADTES) 288
10.3.11 Registers controlling port function of analog input pins 288
10.4 A/D Converter Conversion Operations 289
10.5 Input Voltage and Conversion Results 291
10.6 A/D Converter Operation Modes 292
10.6.1 Software trigger mode (select mode, sequential conversion mode) 292
10.6.2 Software trigger mode (select mode, one-shot conversion mode) 293
10.6.3 Software trigger mode (scan mode, sequential conversion mode) 294
10.6.4 Software trigger mode (scan mode, one-shot conversion mode) 295
10.6.5 Hardware trigger no-wait mode (select mode, sequential conversion mode) 296
10.6.6 Hardware trigger no-wait mode (select mode, one-shot conversion mode) 297
10.6.7 Hardware trigger no-wait mode (scan mode, sequential conversion mode) 298
10.6.8 Hardware trigger no-wait mode (scan mode, one-shot conversion mode) 299
10.6.9 Hardware trigger wait mode (select mode, sequential conversion mode) 300
10.6.10 Hardware trigger wait mode (select mode, one-shot conversion mode) 301

Downloaded from Arrow.com.

10.6.11 Hardware trigger wait mode (scan mode, sequential conversion mode) 302
10.6.12 Hardware trigger wait mode (scan mode, one-shot conversion mode) 303
10.7 A/D Converter Setup Flowchart 304
10.7.1 Setting up software trigger mode 304
10.7.2 Setting up hardware trigger no-wait mode 305
10.7.3 Setting up hardware trigger wait mode 306
10.7.4 Setup when temperature sensor output voltage/internal reference voltage is selected (example for software trigger mode and one-shot conversion mode) 307
10.7.5 Setting up test mode 308
10.8 SNOOZE mode function 309
10.9 How to Read A/D Converter Characteristics Table 313
10.10 Cautions for A/D Converter 315

CHAPTER 11 SERIAL ARRAY UNIT 319

11.1 Functions of Serial Array Unit 320
11.1.1 3-wire serial I/O (CSI00, CSI01, CSI11, CSI20) 320
11.1.2 UART (UART0 to UART2) 321
11.1.3 Simplified I^2C (IIC00, IIC01, IIC11, IIC20) 322
11.2 Configuration of Serial Array Unit 323
11.2.1 Shift register 327
11.2.2 Lower 8/9 bits of the serial data register mn (SDRmn) 327
11.3 Registers Controlling Serial Array Unit 329
11.3.1 Peripheral enable register 0 (PER0) 329
11.3.2 Serial clock select register m (SPSm) 331
11.3.3 Serial mode register mn (SMRmn) 332
11.3.4 Serial communication operation setting register mn (SCRmn) 333
11.3.5 Serial data register mn (SDRmn) 336
11.3.6 Serial flag clear trigger register mn (SIRmn) 338
11.3.7 Serial status register mn (SSRmn) 339
11.3.8 Serial channel start register m (SSm) 341
11.3.9 Serial channel stop register m (STm) 342
11.3.10 Serial channel enable status register m (SEm) 343
11.3.11 Serial output enable register m (SOEm) 344
11.3.12 Serial output register m (SOm) 345
11.3.13 Serial output level register m (SOLm) 346
11.3.14 Serial standby control register 0 (SSC0) 348
11.3.15 Noise filter enable register 0 (NFEN0) 349
11.3.16 Registers controlling port functions of serial input/output pins 350
11.4 Operation Stop Mode 351
11.4.1 Stopping the operation by units 351
11.4.2 Stopping the operation by channels 352

Downloaded from Arrow.com.

11.5 Operation of 3-Wire Serial I/O (CSI00, CSI01, CSI11, CSI20) Communication 353
11.5.1 Master transmission 354
11.5.2 Master reception 363
11.5.3 Master transmission/reception 371
11.5.4 Slave transmission 380
11.5.5 Slave reception 389
11.5.6 Slave transmission/reception 395
11.5.7 SNOOZE mode function 404
11.5.8 Calculating transfer clock frequency 408
11.5.9 Procedure for processing errors that occurred during 3-wire serial I/O (CSI00, CSI01, CSI11, CSI20) communication 410
11.6 Operation of UART (UART0 to UART2) Communication 411
11.6.1 UART transmission 412
11.6.2 UART reception 422
11.6.3 SNOOZE mode function 429
11.6.4 Calculating baud rate 437
11.6.5 Procedure for processing errors that occurred during UART (UART0 to UART2) communication 441
11.7 Operation of Simplified I^2C (IIC00, IIC01, IIC11, IIC20) Communication 442
11.7.1 Address field transmission 444
11.7.2 Data transmission 448
11.7.3 Data reception 451
11.7.4 Stop condition generation 455
11.7.5 Calculating transfer rate 456
11.7.6 Procedure for processing errors that occurred during simplified I^2C (IIC00, IIC01, IIC11, IIC20) communication 458

CHAPTER 12 SERIAL INTERFACE IICA 459

12.1 Functions of Serial Interface IICA 459
12.2 Configuration of Serial Interface IICA 462
12.3 Registers Controlling Serial Interface IICA 465
12.3.1 Peripheral enable register 0 (PER0) 465
12.3.2 IICA control register 00 (IICCTL00) 466
12.3.3 IICA status register 0 (IICS0) 470
12.3.4 IICA flag register 0 (IICF0) 472
12.3.5 IICA control register 01 (IICCTL01) 474
12.3.6 IICA low-level width setting register 0 (IICWL0) 476
12.3.7 IICA high-level width setting register 0 (IICWH0) 476
12.3.8 Port mode register 6 (PM6) 477
12.4 I^2C Bus Mode Functions 478
12.4.1 Pin configuration 478

Downloaded from Arrow.com.

12.4.2 Setting transfer clock by using IICWL0 and IICWH0 registers ... 479
12.5 I^2C Bus Definitions and Control Methods ... 480
12.5.1 Start conditions ... 480
12.5.2 Addresses ... 481
12.5.3 Transfer direction specification ... 481
12.5.4 Acknowledge ($\overline{ACK}$) ... 482
12.5.5 Stop condition ... 483
12.5.6 Wait ... 484
12.5.7 Canceling wait ... 486
12.5.8 Interrupt request (INTIICA0) generation timing and wait control ... 487
12.5.9 Address match detection method ... 488
12.5.10 Error detection ... 488
12.5.11 Extension code ... 488
12.5.12 Arbitration ... 489
12.5.13 Wakeup function ... 491
12.5.14 Communication reservation ... 494
12.5.15 Cautions ... 498
12.5.16 Communication operations ... 499
12.5.17 Timing of I^2C interrupt request (INTIICA0) occurrence ... 507
12.6 Timing Charts ... 528

CHAPTER 13 MULTIPLIER AND DIVIDER/MULTIPLY-ACCUMULATOR ... 543

13.1 Functions of Multiplier and Divider/Multiply-Accumulator ... 543
13.2 Configuration of Multiplier and Divider/Multiply-Accumulator ... 543
13.2.1 Multiplication/division data register A (MDAH, MDAL) ... 545
13.2.2 Multiplication/division data register B (MDBL, MDBH) ... 546
13.2.3 Multiplication/division data register C (MDCL, MDCH) ... 547
13.3 Register Controlling Multiplier and Divider/Multiply-Accumulator ... 549
13.3.1 Multiplication/division control register (MDUC) ... 549
13.4 Operations of Multiplier and Divider/Multiply-Accumulator ... 551
13.4.1 Multiplication (unsigned) operation ... 551
13.4.2 Multiplication (signed) operation ... 552
13.4.3 Multiply-accumulation (unsigned) operation ... 553
13.4.4 Multiply-accumulation (signed) operation ... 555
13.4.5 Division operation ... 557

CHAPTER 14 DMA CONTROLLER ... 559

14.1 Functions of DMA Controller ... 559
14.2 Configuration of DMA Controller ... 560
14.2.1 DMA SFR address register n (DSAn) ... 560
14.2.2 DMA RAM address register n (DRAn) ... 561

Downloaded from Arrow.com.

14.2.3 DMA byte count register n (DBCn) ... 562
14.3 Registers Controlling DMA Controller ... 563
14.3.1 DMA mode control register n (DMCn) ... 563
14.3.2 DMA operation control register n (DRCn) ... 565
14.4 Operation of DMA Controller ... 566
14.4.1 Operation procedure ... 566
14.4.2 Transfer mode ... 567
14.4.3 Termination of DMA transfer ... 567
14.5 Example of Setting of DMA Controller ... 567
14.5.1 CSI consecutive transmission ... 567
14.5.2 Consecutive capturing of A/D conversion results ... 569
14.5.3 UART consecutive reception + ACK transmission ... 570
14.5.4 Holding DMA transfer pending by DWAITn bit ... 571
14.5.5 Forced termination by software ... 572
14.6 Cautions on Using DMA Controller ... 574

CHAPTER 15 INTERRUPT FUNCTIONS ... 577

15.1 Interrupt Function Types ... 577
15.2 Interrupt Sources and Configuration ... 577
15.3 Registers Controlling Interrupt Functions ... 584
15.3.1 Interrupt request flag registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H) ... 588
15.3.2 Interrupt mask flag registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H) ... 590
15.3.3 Priority specification flag registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H) ... 592
15.3.4 External interrupt rising edge enable register (EGP0), external interrupt falling edge enable register (EGN0) ... 596
15.3.5 Program status word (PSW) ... 597
15.4 Interrupt Servicing Operations ... 598
15.4.1 Maskable interrupt request acknowledgment ... 598
15.4.2 Software interrupt request acknowledgment ... 600
15.4.3 Multiple interrupt servicing ... 600
15.4.4 Interrupt request hold ... 604

CHAPTER 16 KEY INTERRUPT FUNCTION ... 605

16.1 Functions of Key Interrupt ... 605
16.2 Configuration of Key Interrupt ... 606
16.3 Register Controlling Key Interrupt ... 608
16.3.1 Key return control register (KRCTL) ... 608
16.3.2 Key return mode control registers (KRM0, KRM1) ... 609
16.3.3 Key return flag register (KRF) ... 610
16.3.4 Port mode registers 0, 4, 6 (PM0, PM4, PM6) ... 610

Downloaded from Arrow.com.

16.4 Key Interrupt Operation 611
16.4.1 When not using the key interrupt flag (KRMD = 0) 611
16.4.2 When using the key interrupt flag (KRMD = 1) 612

CHAPTER 17 STANDBY FUNCTION 615

17.1 Standby Function 615
17.2 Registers controlling standby function 616
17.3 Standby Function Operation 616
17.3.1 HALT mode 616
17.3.2 STOP mode 620
17.3.3 SNOOZE mode 625

CHAPTER 18 RESET FUNCTION 628

18.1 Timing of Reset Operation 630
18.2 States of Operation During Reset Periods 632
18.3 Register for Confirming Reset Source 634
18.3.1 Reset Control Flag Register (RESF) 634

CHAPTER 19 POWER-ON-RESET CIRCUIT 637

19.1 Functions of Power-on-reset Circuit 637
19.2 Configuration of Power-on-reset Circuit 638
19.3 Operation of Power-on-reset Circuit 638

CHAPTER 20 VOLTAGE DETECTOR 642

20.1 Functions of Voltage Detector 642
20.2 Configuration of Voltage Detector 643
20.3 Registers Controlling Voltage Detector 643
20.3.1 Voltage detection register (LVIM) 644
20.3.2 Voltage detection level register (LVIS) 645
20.4 Operation of Voltage Detector 648
20.4.1 When used as reset mode 648
20.4.2 When used as interrupt mode 650
20.4.3 When used as interrupt and reset mode 652
20.5 Cautions for Voltage Detector 658

CHAPTER 21 SAFETY FUNCTIONS 660

21.1 Overview of Safety Functions 660
21.2 Registers Used by Safety Functions 661
21.3 Operation of Safety Functions 661
21.3.1 CRC operation function (general-purpose CRC) 661

Downloaded from Arrow.com.

21.3.1.1 CRC input register (CRCIN) 662
21.3.1.2 CRC data register (CRCD) 662
21.3.2 RAM parity error detection function 664
21.3.2.1 RAM parity error control register (RPECTL) 664
21.3.3 RAM guard function 666
21.3.3.1 Invalid memory access detection control register (IAWCTL) 666
21.3.4 SFR guard function 667
21.3.4.1 Invalid memory access detection control register (IAWCTL) 667
21.3.5 Invalid memory access detection function 668
21.3.5.1 Invalid memory access detection control register (IAWCTL) 669
21.3.6 Frequency detection function 669
21.3.6.1 Timer input select register 0 (TIS0) 670
21.3.7 A/D test function 671
21.3.7.1 A/D test register (ADTES) 672
21.3.7.2 Analog input channel specification register (ADS) 673

CHAPTER 22 REGULATOR 675

22.1 Overview of Regulators 675

CHAPTER 23 OPTION BYTE 676

23.1 Functions of Option Bytes 676
23.1.1 User option byte (000C0H to 000C2H) 676
23.1.2 On-chip debug option byte (000C3H) 677
23.2 Format of User Option Byte 678
23.3 Format of On-chip Debug Option Byte 682
23.4 Setting of Option Byte 683

CHAPTER 24 FLASH MEMORY 684

24.1 Serial Programming Using Flash Memory Programmer 686
24.1.1 Programming environment 687
24.1.2 Communication mode 687
24.2 Serial Programming Using External Device (that Incorporates UART) 688
24.2.1 Programming environment 688
24.2.2 Communication mode 689
24.3 Connection of Pins on Board 690
24.3.1 P40/TOOL0 pin 690
24.3.2 $\overline{\text{RESET}}$ pin 690
24.3.3 Port pins 691
24.3.4 REGC pins 691
24.3.5 X1 and X2 pins 691
24.3.6 Power supply 691
24.4 Serial Programming Method 692

Downloaded from Arrow.com.

24.4.1 Serial programming procedure ... 692
24.4.2 Flash memory programming mode ... 692
24.4.3 Selecting communication mode ... 694
24.4.4 Communication commands ... 694
24.5 Processing Time for Each Command When PG-FP5 Is in Use (Reference Value) ... 696
24.6 Self-Programming ... 697
24.6.1 Self-programming procedure ... 698
24.6.2 Flash shield window function ... 699
24.7 Security Settings ... 700
24.8 Data Flash ... 702
24.8.1 Data flash overview ... 702
24.8.2 Register controlling data flash memory ... 702
24.8.2.1 Data flash control register (DFLCTL) ... 702
24.8.3 Procedure for accessing data flash memory ... 703

CHAPTER 25 ON-CHIP DEBUG FUNCTION ... 704

25.1 Connecting E1 On-chip Debugging Emulator ... 704
25.2 On-Chip Debug Security ID ... 706
25.3 Securing of User Resources ... 706

CHAPTER 26 BCD CORRECTION CIRCUIT ... 708

26.1 BCD Correction Circuit Function ... 708
26.2 Registers Used by BCD Correction Circuit ... 708
26.2.1 BCD correction result register (BCDADJ) ... 708
26.3 BCD Correction Circuit Operation ... 709

CHAPTER 27 INSTRUCTION SET ... 711

27.1 Conventions Used in Operation List ... 712
27.1.1 Operand identifiers and specification methods ... 712
27.1.2 Description of operation column ... 713
27.1.3 Description of flag operation column ... 714
27.1.4 PREFIX instruction ... 714
27.2 Operation List ... 715

CHAPTER 28 ELECTRICAL SPECIFICATIONS (A, D: T_A = –40 to +85°C) ... 732

28.1 Absolute Maximum Ratings ... 733
28.2 Oscillator Characteristics ... 734
28.2.1 X1 oscillator characteristics ... 734
28.2.2 On-chip oscillator characteristics ... 734
28.3 DC Characteristics ... 735

Downloaded from Arrow.com.

28.3.1 Pin characteristics 735
28.3.2 Supply current characteristics 739
28.4 AC Characteristics 744
28.5 Peripheral Functions Characteristics 747
28.5.1 Serial array unit 747
28.5.2 Serial interface IICA 768
28.6 Analog Characteristics 769
28.6.1 A/D converter characteristics 769
28.6.2 Temperature sensor/internal reference voltage characteristics 773
28.6.3 POR circuit characteristics 773
28.6.4 LVD circuit characteristics 774
28.6.5 Power supply voltage rising slope characteristics 775
28.7 Data Memory STOP Mode Low Supply Voltage Data Retention Characteristics 776
28.8 Flash Memory Programming Characteristics 776
28.9 Dedicated Flash Memory Programmer Communication (UART) 777
28.10 Timing of Entry to Flash Memory Programming Modes 777

CHAPTER 29 ELECTRICAL SPECIFICATIONS (G: T_A = –40 to +105°C) 778

29.1 Absolute Maximum Ratings 779
29.2 Oscillator Characteristics 780
29.2.1 X1 oscillator characteristics 780
29.2.2 On-chip oscillator characteristics 780
29.3 DC Characteristics 781
29.3.1 Pin characteristics 781
29.3.2 Supply current characteristics 785
29.4 AC Characteristics 790
29.5 Peripheral Functions Characteristics 793
29.5.1 Serial array unit 793
29.5.2 Serial interface IICA 810
29.6 Analog Characteristics 811
29.6.1 A/D converter characteristics 811
29.6.2 Temperature sensor/internal reference voltage characteristics 815
29.6.3 POR circuit characteristics 815
29.6.4 LVD circuit characteristics 816
29.6.5 Power supply voltage rising slope characteristics 817
29.7 Data Memory STOP Mode Low Supply Voltage Data Retention Characteristics 818
29.8 Flash Memory Programming Characteristics 818
29.9 Dedicated Flash Memory Programmer Communication (UART) 819
29.10 Timing of Entry to Flash Memory Programming Modes 819

Downloaded from Arrow.com.

CHAPTER 30 PACKAGE DRAWINGS 820

30.1 20-pin products 820
30.2 24-pin products 821
30.3 30-pin products 822

APPENDIX A REVISION HISTORY 823

A.1 Major Revisions in This Edition 823
A.2 Revision History of Preceding Editions 832

Downloaded from Arrow.com.

CHAPTER 1 OUTLINE

1.1 Differences between the R5F102 Products and the R5F103 Products

The following are differences between the R5F102 products and the R5F103 products.

- Whether the data flash memory is mounted or not
- High-speed on-chip oscillator oscillation frequency accuracy
- Number of channels in serial interface
- Whether the DMA function is mounted or not
- Whether a part of the safety functions are mounted or not

1.1.1 Data Flash

The data flash memory of 2 KB is mounted on the R5F102 products, but not on the R5F103 products.

Product	Data Flash
R5F102 products R5F1026A, R5F1027A, R5F102AA, R5F10269, R5F10279, R5F102A9, R5F10268, R5F10278, R5F102A8, R5F10267, R5F10277, R5F102A7, R5F10266 Note	2KB
R5F103 products R5F1036A, R5F1037A, R5F103AA, R5F10369, R5F10379, R5F103A9, R5F10368, R5F10378 R5F103A8, R5F10367, R5F10377, R5F103A7, R5F10366	Not mounted

Note The RAM in the R5F10266 has capacity as small as 256 bytes. Depending on the customer's program specification, the stack area to execute the data flash library may not be kept and data may not be written to or erased from the data flash memory.

Caution When the flash memory is rewritten via a user program, the code flash area and RAM area are used because each library is used. When using the library, refer to RL78 Family Flash Self Programming Library Type01 User's Manual and RL78 Family Data Flash Library Type04 User's Manual.

Downloaded from Arrow.com.

1.1.2 On-chip oscillator characteristics

(1) High-speed on-chip oscillator oscillation frequency of the R5F102 products

Oscillator	Condition	MIN	MAX	Unit
High-speed on-chip oscillator oscillation frequency accuracy	T_A = -20 to +85 °C	-1.0	+1.0	%
	T_A = -40 to -20 °C	-1.5	+1.5	
	T_A = +85 to +105 °C	-2.0	+2.0	

(2) High-speed on-chip oscillator oscillation frequency of the R5F103 products

Oscillator	Condition	MIN	MAX	Unit
High-speed on-chip oscillator oscillation frequency accuracy	T_A = -40 to + 85 °C	-5.0	+5.0	%

1.1.3 Peripheral Functions

The following are differences in peripheral functions between the R5F102 products and the R5F103 products.

RL78/G12		R5F102 product		R5F103 product	
		20, 24 pin product	30 pin product	20, 24 pin product	30 pin product
Serial interface	UART	1 channel	3 channels	1 channel	
	CSI	2 channels	3 channels	1 channel	
	Simplified I^2C	2 channels	3 channels	None	
DMA function		2 channels		None	
Safety function	CRC operation	Yes		None	
	RAM guard	Yes		None	
	SFR guard	Yes		None	

Downloaded from Arrow.com.

<R> 1.2 Features

Ultra-low power consumption technology

- V_{DD} = single power supply voltage of 1.8 to 5.5 V which can operate at a low voltage
- HALT mode
- STOP mode
- SNOOZE mode

RL78 CPU core

- CISC architecture with 3-stage pipeline
- Minimum instruction execution time: Can be changed from high speed (0.04167 μs: @ 24 MHz operation with high-speed on-chip oscillator) to ultra-low speed (1 μs: @ 1 MHz operation)
- Address space: 1 MB
- General-purpose registers: (8-bit register $\times$ 8) $\times$ 4 banks
- On-chip RAM: 256 B to 2 KB

Code flash memory

- Code flash memory: 2 to 16 KB
- Block size: 1 KB
- Prohibition of block erase and rewriting (security function)
- On-chip debug function
- Self-programming (with flash shield window function)

Data flash memory **Note**

- Data flash memory: 2 KB
- Back ground operation (BGO): Instructions are executed from the program memory while rewriting the data flash memory.
- Number of rewrites: 1,000,000 times (TYP.)
- Voltage of rewrites: V_{DD} = 1.8 to 5.5 V

High-speed on-chip oscillator

- Select from 24 MHz, 16 MHz, 12 MHz, 8 MHz, 6 MHz, 4 MHz, 3 MHz, 2 MHz, and 1 MHz
- High accuracy: ±1.0 % (V_{DD} = 1.8 to 5.5 V, T_A = -20 to +85°C)

Operating ambient temperature

- T_A = -40 to +85°C (A: Consumer applications, D: Industrial applications)
- T_A = -40 to +105°C (G: Industrial applications) **Note**

Power management and reset function

- On-chip power-on-reset (POR) circuit
- On-chip voltage detector (LVD) (Select interrupt and reset from 12 levels)

DMA (Direct Memory Access) controller **Note**

- 2 channels
- Number of clocks during transfer between 8/16-bit SFR and internal RAM: 2 clocks

Note Provided in the R5F102 products only.

Downloaded from Arrow.com.

Multiplier and divider/multiply-accumulator

- 16 bits × 16 bits = 32 bits (Unsigned or signed)
- 32 bits × 32 bits = 32 bits (Unsigned)
- 16 bits × 16 bits + 32 bits = 32 bits (Unsigned or signed)

Serial interface

- CSI : 1 to 3 channels
- UART : 1 to 3 channels
- Simplified I^2C communication : 0 to 3 channels
- I^2C communication : 1 channel

Timer

- 16-bit timer : 4 to 8 channels
- 12-bit interval timer : 1 channel
- Watchdog timer : 1 channel (operable with the dedicated low-speed on-chip oscillator)

A/D converter

- 8/10-bit resolution A/D converter (V_{DD} = 1.8 to 5.5 V)
- 8 to 11 channels, internal reference voltage (1.45 V), and temperature sensor Note

I/O port

- I/O port: 18 to 26 (N-ch open drain I/O [withstand voltage of 6 V]: 2,
  N-ch open drain I/O [V_{DD} withstand voltage]: 4 to 9)
- Can be set to N-ch open drain, TTL input buffer, and on-chip pull-up resistor
- Different potential interface: Can connect to a 1.8/2.5/3 V device
- On-chip key interrupt function
- On-chip clock output/buzzer output controller

Others

- On-chip BCD (binary-coded decimal) correction circuit

Note Can be selected only in HS (high-speed main) mode.

Remark The functions mounted depend on the product. See **1.7 Outline of Functions**.

Downloaded from Arrow.com.

○ ROM, RAM capacities

Code flash	Data flash	RAM	20 pins	24 pins	30 pins
16 KB	2 KB	2 KB	—	—	R5F102AA
	—		—	—	R5F103AA
	2 KB	1.5 KB	R5F1026A Note 1	R5F1027A Note 1	—
	—		R5F1036A Note 1	R5F1037A Note 1	—
12 KB	2KB	1 KB	R5F10269 Note 1	R5F10279 Note 1	R5F102A9
	—		R5F10369 Note 1	R5F10379 Note 1	R5F103A9
8 KB	2 KB	768 B	R5F10268 Note 1	R5F10278 Note 1	R5F102A8
	—		R5F10368 Note 1	R5F10378 Note 1	R5F103A8
4 KB	2KB	512 B	R5F10267	R5F10277	R5F102A7
	—		R5F10367	R5F10377	R5F103A7
2 KB	2 KB	256 B	R5F10266 Note 2	—	—
	—		R5F10366 Note 2	—	—

Notes 1. This is 640 bytes when the self-programming function or data flash function is used. (For details, see **CHAPTER 3 CPU ARCHITECTURE**.)

2. The self-programming function cannot be used for R5F10266 and R5F10366.

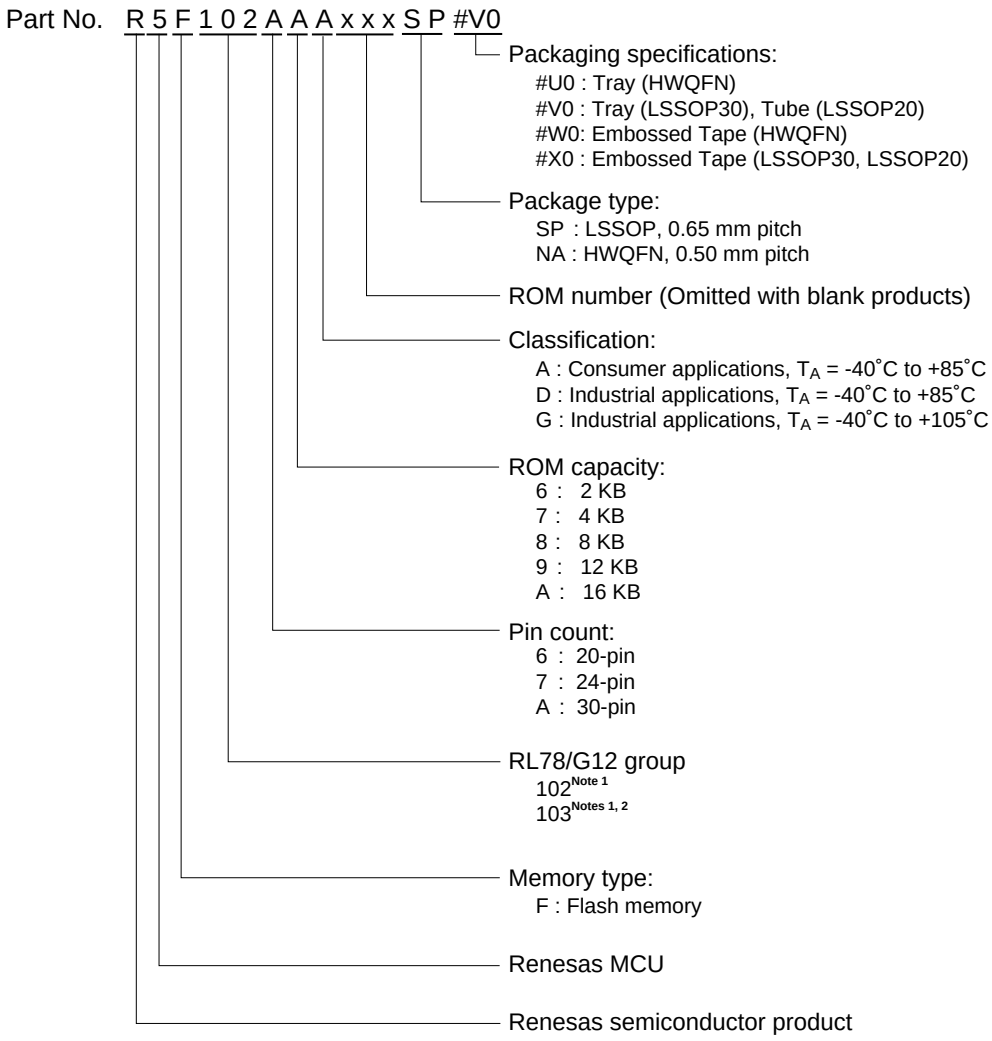
Caution When the flash memory is rewritten via a user program, the code flash area and RAM area are used because each library is used. When using the library, refer to RL78 Family Flash Self Programming Library Type01 User's Manual and RL78 Family Data Flash Library Type04 User's Manual.

Downloaded from Arrow.com.

1.3 List of Part Numbers

Figure 1-1. Part Number, Memory Size, and Package of RL78/G12

<R>



Notes 1. For details about the differences between the R5F102 products and the R5F103 products of RL78/G12, see **1.1 Differences between the R5F102 Products and the R5F103 Products**.

2. Products only for "A: Consumer applications (T_A = -40 to +85°C)" and "D: Industrial applications (T_A = -40 to +85°C)"

Downloaded from Arrow.com.

Table 1-1. List of Ordering Part Numbers

<R>

Pin count	Package	Data flash	Fields of Application	Part Number
20 pins	20-pin plastic LSSOP (4.4 × 6.5 mm, 0.65 mm pitch)	Mounted	A	R5F1026AASP#V0, R5F10269ASP#V0, R5F10268ASP#V0, R5F10267ASP#V0, R5F10266ASP#V0 R5F1026AASP#X0, R5F10269ASP#X0, R5F10268ASP#X0, R5F10267ASP#X0, R5F10266ASP#X0
			D	R5F1026ADSP#V0, R5F10269DSP#V0, R5F10268DSP#V0, R5F10267DSP#V0, R5F10266DSP#V0 R5F1026ADSP#X0, R5F10269DSP#X0, R5F10268DSP#X0, R5F10267DSP#X0, R5F10266DSP#X0
			G	R5F1026AGSP#V0, R5F10269GSP#V0, R5F10268GSP#V0, R5F10267GSP#V0, R5F10266GSP#V0 R5F1026AGSP#X0, R5F10269GSP#X0, R5F10268GSP#X0, R5F10267GSP#X0, R5F10266GSP#X0
		Not mounted	A	R5F1036AASP#V0, R5F10369ASP#V0, R5F10368ASP#V0, R5F10367ASP#V0, R5F10366ASP#V0 R5F1036AASP#X0, R5F10369ASP#X0, R5F10368ASP#X0, R5F10367ASP#X0, R5F10366ASP#X0
			D	R5F1036ADSP#V0, R5F10369DSP#V0, R5F10368DSP#V0, R5F10367DSP#V0, R5F10366DSP#V0 R5F1036ADSP#X0, R5F10369DSP#X0, R5F10368DSP#X0, R5F10367DSP#X0, R5F10366DSP#X0
24 pins	24-pin plastic HWQFN (4 × 4 mm, 0.5 mm pitch)	Mounted	A	R5F1027AANA#U0, R5F10279ANA#U0, R5F10278ANA#U0, R5F10277ANA#U0 R5F1027AANA#W0, R5F10279ANA#W0, R5F10278ANA#W0, R5F10277ANA#W0
			D	R5F1027ADNA#U0, R5F10279DNA#U0, R5F10278DNA#U0, R5F10277DNA#U0 R5F1027ADNA#W0, R5F10279DNA#W0, R5F10278DNA#W0, R5F10277DNA#W0
			G	R5F1027AGNA#U0, R5F10279GNA#U0, R5F10278GNA#U0, R5F10277GNA#U0 R5F1027AGNA#W0, R5F10279GNA#W0, R5F10278GNA#W0, R5F10277GNA#W0
		Not mounted	A	R5F1037AANA#V0, R5F10379ANA#V0, R5F10378ANA#V0, R5F10377ANA#V0 R5F1037AANA#X0, R5F10379ANA#X0, R5F10378ANA#X0, R5F10377ANA#X0
			D	R5F1037ADNA#V0, R5F10379DNA#V0, R5F10378DNA#V0, R5F10377DNA#V0 R5F1037ADNA#X0, R5F10379DNA#X0, R5F10378DNA#X0, R5F10377DNA#X0
30 pins	30-pin plastic LSSOP (7.62 mm (300), 0.65 mm pitch)	Mounted	A	R5F102AAASP#V0, R5F102A9ASP#V0, R5F102A8ASP#V0, R5F102A7ASP#V0 R5F102AAASP#X0, R5F102A9ASP#X0, R5F102A8ASP#X0, R5F102A7ASP#X0
			D	R5F102AADSP#V0, R5F102A9DSP#V0, R5F102A8DSP#V0, R5F102A7DSP#V0 R5F102AADSP#X0, R5F102A9DSP#X0, R5F102A8DSP#X0, R5F102A7DSP#X0
			G	R5F102AAGSP#V0, R5F102A9GSP#V0, R5F102A8GSP#V0, R5F102A7GSP#V0 R5F102AAGSP#X0, R5F102A9GSP#X0, R5F102A8GSP#X0, R5F102A7GSP#X0
		Not mounted	A	R5F103AAASP#V0, R5F103A9ASP#V0, R5F103A8ASP#V0, R5F103A7ASP#V0 R5F103AAASP#X0, R5F103A9ASP#X0, R5F103A8ASP#X0, R5F103A7ASP#X0
			D	R5F103AADSP#V0, R5F103A9DSP#V0, R5F103A8DSP#V0, R5F103A7DSP#V0 R5F103AADSP#X0, R5F103A9DSP#X0, R5F103A8DSP#X0, R5F103A7DSP#X0

Note For fields of application, see **Figure 1-1 Part Number, Memory Size, and Package of RL78/G12**.

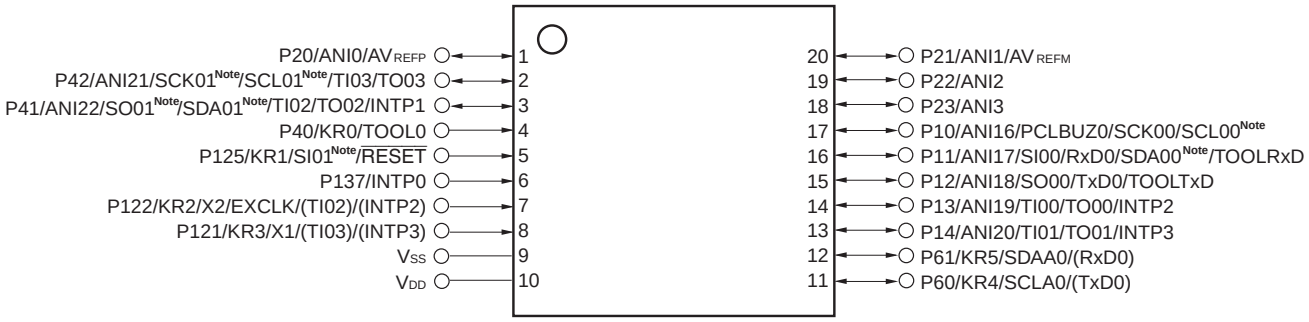
<R> **Caution The ordering part numbers represent the numbers at the time of publication. For the latest ordering part numbers, refer to the target product page of the Renesas Electronics website.**

Downloaded from Arrow.com.

1.4 Pin Configuration (Top View)

1.4.1 20-pin products

<R> • 20-pin plastic LSSOP (4.4 × 6.5 mm, 0.65 mm pitch)



Note Provided only in the R5F102 products.

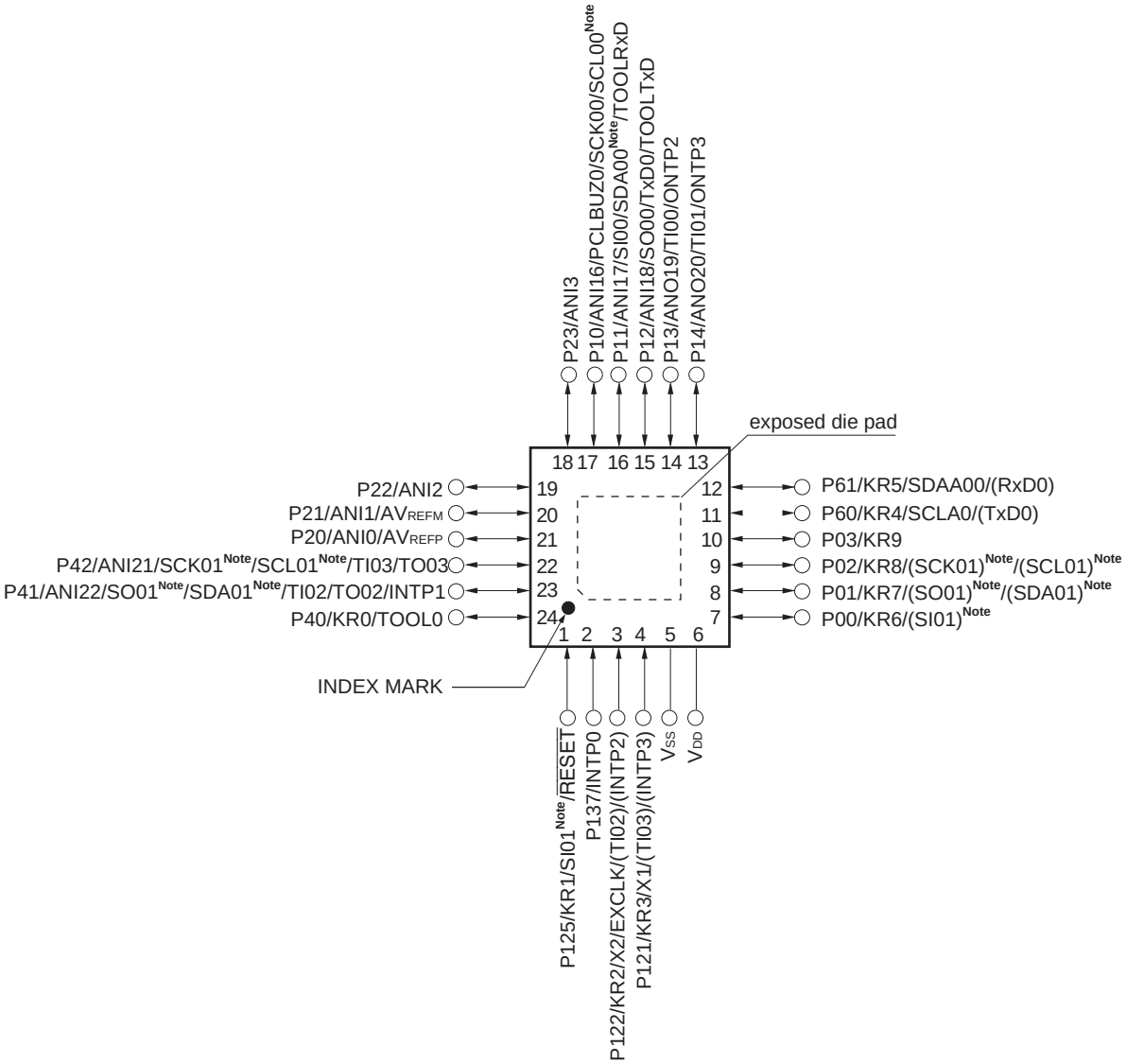
Remarks 1. For pin identification, see **1.5 Pin Identification**.

2. Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR). See **Figure 4-8 Format of Peripheral I/O Redirection Register (PIOR)**.

Downloaded from Arrow.com.

1.4.2 24-pin products

<R> • 24-pin plastic HWQFN (4 × 4 mm, 0.5 mm pitch)



Note Provided only in the R5F102 products.

Remarks 1. For pin identification, see **1.5 Pin Identification**.

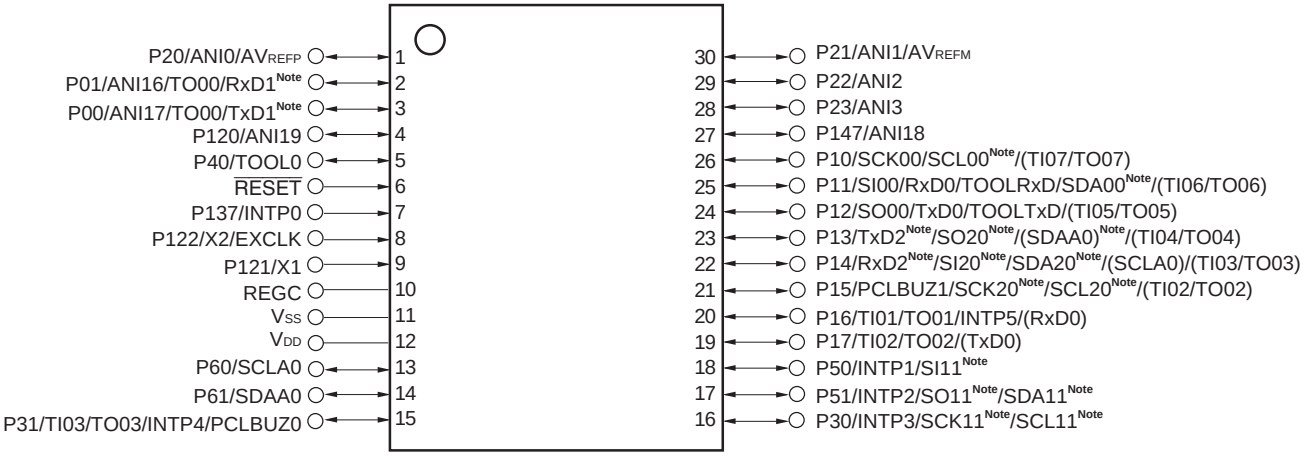
2. Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR). See **Figure 4-8 Format of Peripheral I/O Redirection Register (PIOR)**.

<R> **3.** It is recommended to connect an exposed die pad to Vss.

Downloaded from Arrow.com.

1.4.3 30-pin products

<R> • 30-pin plastic LSSOP (7.62 mm (300), 0.65 mm pitch)



Note Provided only in the R5F102 products.

Caution Connect the REGC pin to VSS via capacitor (0.47 to 1 μF).

Remarks 1. For pin identification, see **1.5 Pin Identification**.

2. Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR). See **Figure 4-8 Format of Peripheral I/O Redirection Register (PIOR).**

Downloaded from Arrow.com.

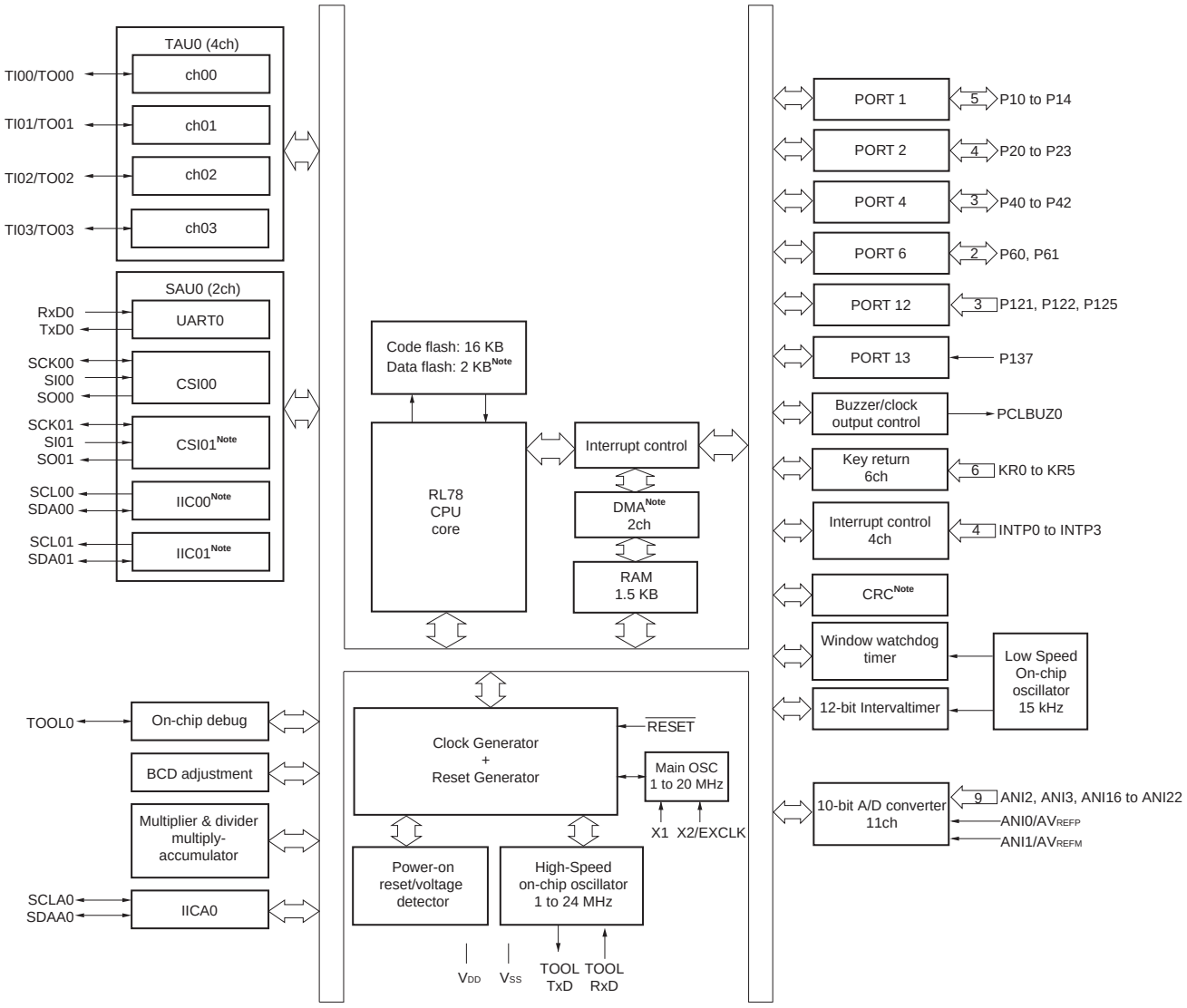
1.5 Pin Identification

ANI0 to ANI3, ANI16 to ANI22:	Analog input
AV_{REFM}:	Analog Reference Voltage Minus
AV_{REFP}:	Analog reference voltage plus
EXCLK:	External Clock Input (Main System Clock)
INTP0 to INTP5	Interrupt Request From Peripheral
KR0 to KR9:	Key Return
P00 to P03:	Port 0
P10 to P17:	Port 1
P20 to P23:	Port 2
P30 to P31:	Port 3
P40 to P42:	Port 4
P50, P51:	Port 5
P60, P61:	Port 6
P120 to P122, P125:	Port 12
P137:	Port 13
P147:	Port 14
PCLBUZ0, PCLBUZ1:	Programmable Clock Output/ Buzzer Output
REGC:	Regulator Capacitance
$\overline{\text{RESET}}$:	Reset
RxD0 to RxD2:	Receive Data
SCK00, SCK01, SCK11, SCK20:	Serial Clock Input/Output
SCL00, SCL01, SCL11, SCL20, SCLA0:	Serial Clock Input/Output
SDA00, SDA01, SDA11, SDA20, SDAA0:	Serial Data Input/Output
SI00, SI01, SI11, SI20:	Serial Data Input
SO00, SO01, SO11, SO20:	Serial Data Output
TI00 to TI07:	Timer Input
TO00 to TO07:	Timer Output
TOOL0:	Data Input/Output for Tool
TOOLRxD, TOOLTxD:	Data Input/Output for External Device
TxD0 to TxD2:	Transmit Data
V_{DD}:	Power supply
V_{SS}:	Ground
X1, X2:	Crystal Oscillator (Main System Clock)

Downloaded from Arrow.com.

1.6 Block Diagram

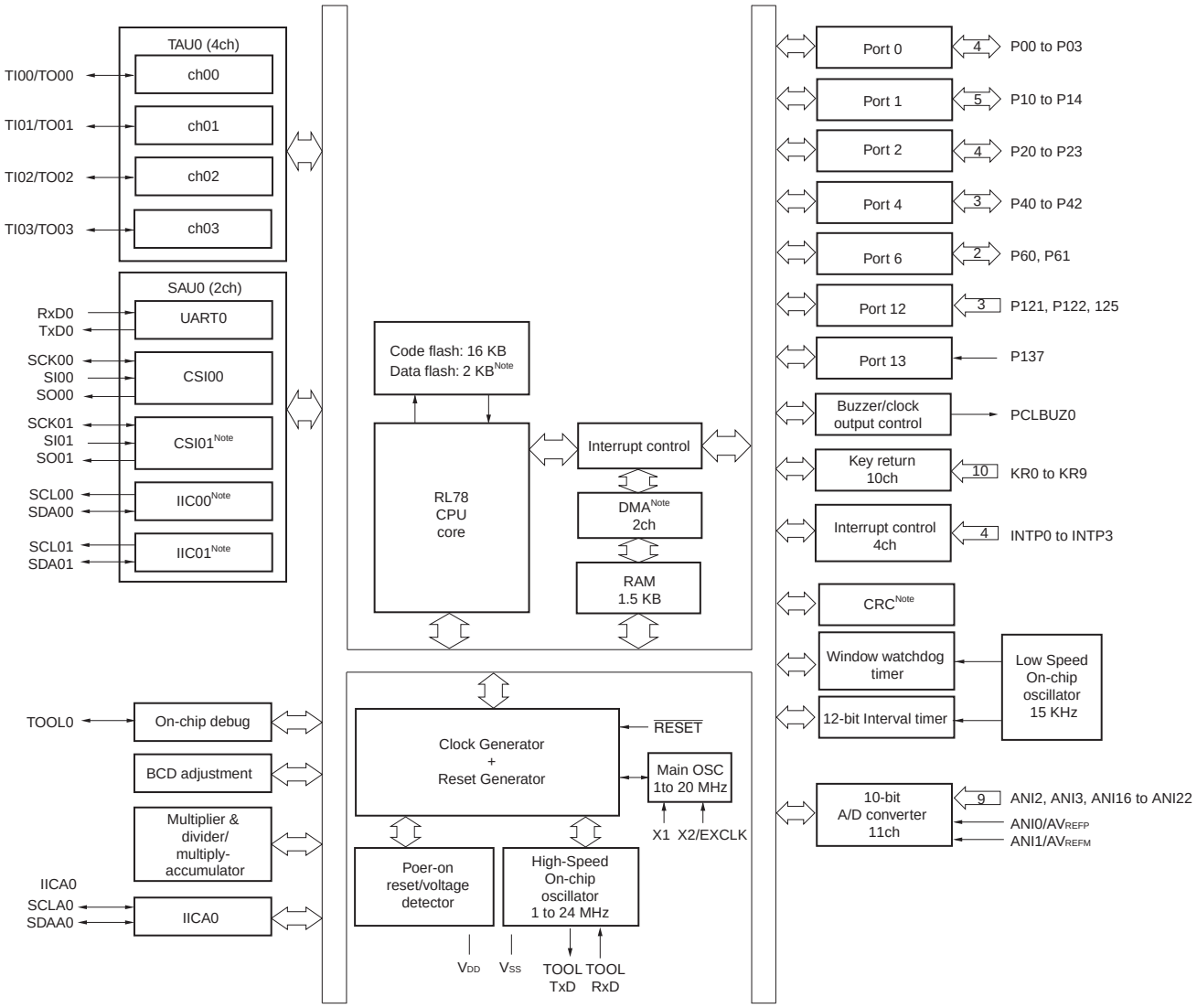
1.6.1 20-pin products



Note Provided only in the R5F102 products.

Downloaded from Arrow.com.

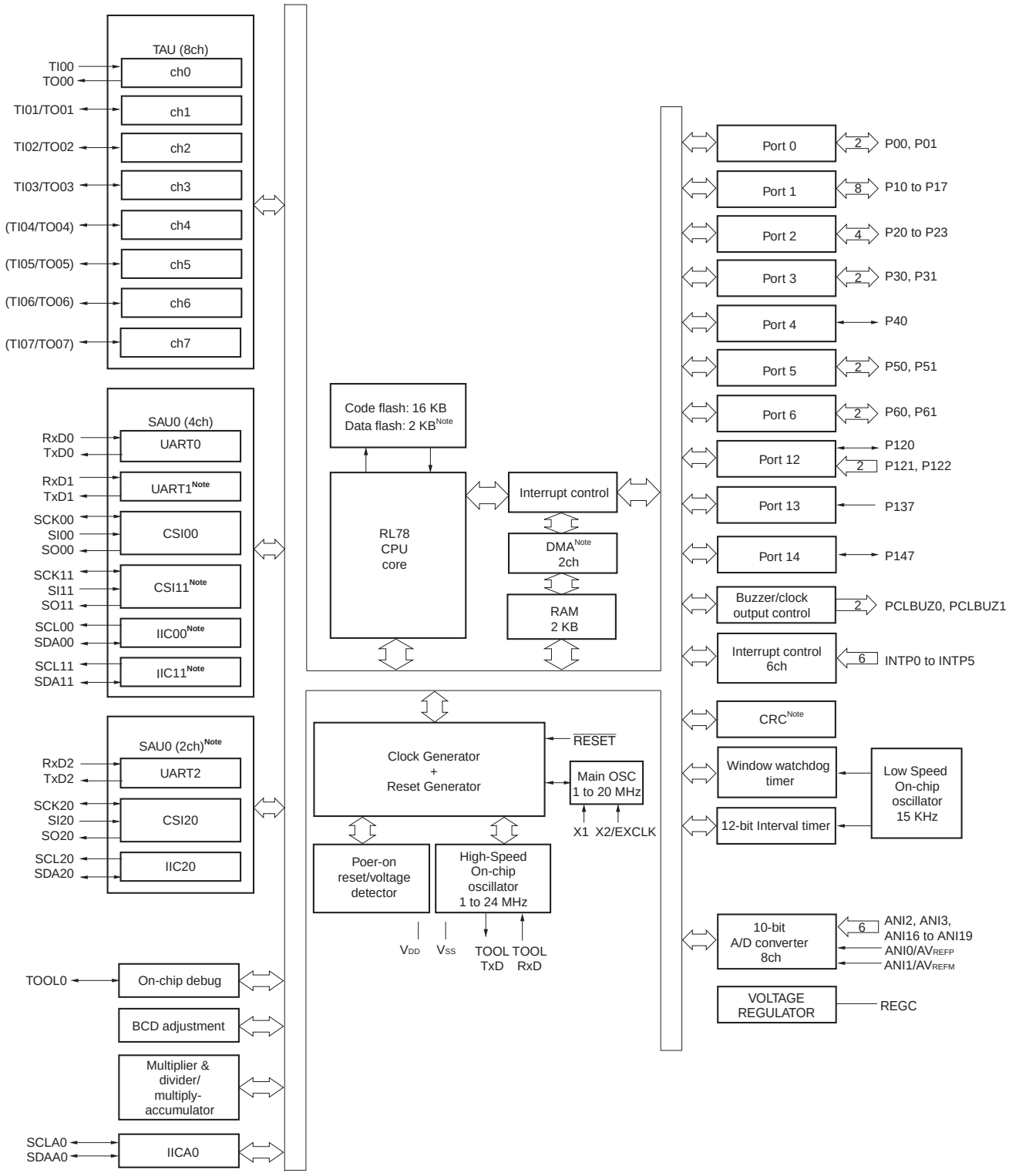
1.6.2 24-pin products



Note Provided only in the R5F102 products.

Downloaded from Arrow.com.

1.6.3 30-pin products



Note Provided only in the R5F102 products.

Remark Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR). See **Figure 4-8 Format of Peripheral I/O Redirection Register (PIOR)**.

Downloaded from Arrow.com.

1.7 Outline of Functions

This outline describes the function at the time when Peripheral I/O redirection register (PIOR) is set to 00H.

(1/2)

Item		20-pin		24-pin		30-pin	
		R5F1026x	R5F1036x	R5F1027x	R5F1037x	R5F102Ax	R5F103Ax
Code flash memory		2 to 16 KB Note 1		4 to 16 KB			
Data flash memory		2 KB	–	2 KB	–	2 KB	–
RAM		256 B to 1.5 KB		512 B to 1.5 KB		512 B to 2KB	
Address space		1 MB					
Main system clock	High-speed system clock	X1, X2 (crystal/ceramic) oscillation, external main system clock input (EXCLK) 1 to 20 MHz: V_{DD} = 2.7 to 5.5 V, 1 to 8 MHz: V_{DD} = 1.8 to 5.5 V					
	High-speed on-chip oscillator clock	HS (High-speed main) mode : 1 to 24 MHz (V_{DD} = 2.7 to 5.5 V), 1 to 16 MHz (V_{DD} = 2.4 to 5.5 V), LS (Low-speed main) mode : 1 to 8 MHz (V_{DD} = 1.8 to 5.5 V)					
Low-speed on-chip oscillator clock		15 kHz (TYP)					
General-purpose register		(8-bit register × 8) × 4 banks					
Minimum instruction execution time		0.04167 μs (High-speed on-chip oscillator clock: f_{IH} = 24 MHz operation)					
		0.05 μs (High-speed system clock: f_{MX} = 20 MHz operation)					
Instruction set		・ Data transfer (8/16 bits) ・ Adder and subtractor/logical operation (8/16 bits) ・ Multiplication (8 bits × 8 bits) ・ Rotate, barrel shift, and bit manipulation (set, reset, test, and Boolean operation), etc.					
I/O port	Total	18		22		26	
<R>	CMOS I/O	12 (N-ch O.D. I/O [V_{DD} withstand voltage]: 4)		16 (N-ch O.D. I/O [V_{DD} withstand voltage]: 5)		21 (N-ch O.D. I/O [V_{DD} withstand voltage]: 9)	
	CMOS input	4		4		3	
	N-ch open-drain I/O (6 V tolerance)	2					
Timer	16-bit timer	4 channels				8 channels	
	Watchdog timer	1 channel					
	12-bit Interval timer	1 channel					
	Timer output	4 channels (PWM outputs: 3 Note 3)				8 channels (PWM outputs: 7 Note 3, Note 2)	

Notes 1. The self-programming function cannot be used in the R5F10266 and R5F10366.

2. The maximum number of channels when PIOR0 is set to 1.

3. The number of PWM outputs varies depending on the setting of channels in use (the number of masters and slaves). (See **6.9.3 Operation as multiple PWM output function**.)

Caution When the flash memory is rewritten via a user program, the code flash area and RAM area are used because each library is used. When using the library, refer to RL78 Family Flash Self Programming Library Type01 User's Manual and RL78 Family Data Flash Library Type04 User's Manual.

Downloaded from Arrow.com.

(2/2)

Item		20-pin		24-pin		30-pin	
		R5F1026x	R5F1036x	R5F1027x	R5F1037x	R5F102Ax	R5F103Ax
Clock output/buzzer output		1				2	
		2.44 kHz to 10 MHz: (Peripheral hardware clock: f_{MAIN} = 20 MHz operation)					
8/10-bit resolution A/D converter		11 channels				8 channels	
Serial interface		[R5F1026x (20-pin), R5F1027x (24-pin)] ・ CSI: 2 channels/Simplified I^2C: 2 channels/UART: 1 channel [R5F102Ax (30-pin)] ・ CSI: 1 channel/Simplified I^2C: 1 channel/UART: 1 channel ・ CSI: 1 channel/Simplified I^2C: 1 channel/UART: 1 channel ・ CSI: 1 channel/Simplified I^2C: 1 channel/UART: 1 channel [R5F1036x (20-pin), R5F1037x (24-pin)] ・ CSI: 1 channel/Simplified I^2C: 0 channel/UART: 1 channel [R5F103Ax (30-pin)] ・ CSI: 1 channel/Simplified I^2C: 0 channel/UART: 1 channel					
	I^2C bus	1 channel					
Multiplier and divider/multiply-accumulator		• 16 bits × 16 bits = 32 bits (unsigned or signed) • 32 bits × 32 bits = 32 bits (unsigned) • 16 bits × 16 bits + 32 bits = 32 bits (unsigned or signed)					
DMA controller		2 channels	—	2 channels	—	2 channels	—
Vectored interrupt sources	Internal	18	16	18	16	26	19
	External	5				6	
Key interrupt		6		10		—	
Reset		• Reset by $\overline{\text{RESET}}$ pin • Internal reset by watchdog timer • Internal reset by power-on-reset • Internal reset by voltage detector • Internal reset by illegal instruction execution Note • Internal reset by RAM parity error • Internal reset by illegal-memory access					
Power-on-reset circuit		• Power-on-reset: 1.51 V (TYP) • Power-down-reset: 1.50 V (TYP)					
Voltage detector		• Rising edge : 1.88 to 4.06 V (12 stages) • Falling edge : 1.84 to 3.98 V (12 stages)					
On-chip debug function		Provided					
Power supply voltage		V_{DD} = 1.8 to 5.5 V					
Operating ambient temperature		T_A = –40 to +85°C (A: Consumer applications, D: Industrial applications), T_A = –40 to +105°C (G: Industrial applications)					

Note The illegal instruction is generated when instruction code FFH is executed.
Reset by the illegal instruction execution not issued by emulation with the in-circuit emulator or on-chip debug emulator.

Downloaded from Arrow.com.

CHAPTER 2 PIN FUNCTIONS

<R> 2.1 Port Functions

V_{DD} is the I/O buffer power supply for pins.

Set in each port I/O, buffer, pull-up resistor is also valid for alternate functions.

2.1.1 20-pin products

(1/2)

<R>

Function Name	Pin Type	I/O	After Reset	Alternate Function	Function
P10	8-3-2	I/O	Analog input port	ANI16/PCLBUZ0/ SCK00/ SCL00 Note 3	Port 1. 5-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port (in 1-bit units). Input of P10 and P11 can be set to TTL input buffer. Output of P10 to P12 can be set to N-ch open-drain output (V_{DD} tolerance). Can be set to analog input Note 1.
P11				ANI17/SI00/RxD0/ SDA00 Note 3/ TOOLRxD	
P12	7-3-2			ANI18/SO00/TxD0/ TOOLTxD	
P13	7-3-1			ANI19/TI00/TO00/ INTP2	
P14				ANI20/TI01/TO01/ INTP3	
P20	4-3-1	I/O	Analog input port	ANI0/AV_{REFP}	Port 2. 4-bit I/O port. Input/output can be specified in 1-bit units. Can be set to analog input Note 2.
P21				ANI1/AV_{REFM}	
P22				ANI2	
P23				ANI3	
P40	7-1-1	I/O	Input port	KR0/TOOL0	Port 4. 3-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port (in 1-bit units). Output of P41 can be set to N-ch open-drain output (V_{DD} tolerance). P41 and P42 can be set to analog input Note 1.
P41	7-3-2		Analog input port	ANI22/SO01 Note 3/ SDA01 Note 3/TI02/ TO02/INTP1	
P42	7-3-1			ANI21/SCK01 Note 3/ SCL01 Note 3/TI03/ TO03	
P60	12-1-1	I/O	Input port	KR4/SCLA0/(TxD0)	Port 6 2-bit I/O port. Input/output can be specified in 1-bit units. N-ch open-drain output (6-V tolerance).
P61				KR5/SDAA0/(RxD0)	

Notes 1. Digital or analog for each pin can be selected with the port mode control register x (PMCx) (can be set in 1-bit unit).

<R> **2.** Digital or analog for each pin can be selected with the A/D port configuration register (ADPC).

3. Provided only in the R5F102 products.

Remark Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR). For details, see **Figure 4-8 Format of Peripheral I/O Redirection Register (PIOR)**.

Downloaded from Arrow.com.

(2/2)

Function Name	Pin Type	I/O	After Reset	Alternate Function	Function
P121	2-2-1	Input	Input port	KR3/X1/ (TI03)/(INTP3)	Port 12 3-bit input port. P125 is also used for the input pin for external reset ($\overline{\text{RESET}}$). To use the pin for external reset, set the PORTSELB bit in the option byte (000C1H) to 1. For P125, use of an on-chip pull-up resistor can be specified by a software setting.
P122				KR2/X2/EXCLK/ (TI02)/(INTP2)	
P125	7-1-1			KR1/SI01/$\overline{\text{RESET}}$	
P137	2-1-2	Input	Input port	INTP0	Port 13 1-bit input port

Remark Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR). For details, see **Figure 4-8 Format of Peripheral I/O Redirection Register (PIOR)**.

2.1.2 24-pin products

(1/2)

<R>

Function Name	Pin Type	I/O	After Reset	Alternate Function	Function
P00	7-1-1	I/O	Input port	KR6/(SI01) Note 3	Port 0. 4-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port (in 1-bit units). Output of P01 can be set to N-ch open-drain output (V_{DD} tolerance).
P01	7-1-2			KR7/(SO01) Note 3/ (SDA01) Note 3	
P02	7-1-1			KR8/(SCK01) Note 3/ (SCL01) Note 3	
P03				KR9	
P10	8-3-2	I/O	Analog input port	ANI16/PCLBUZ0/ SCK00/SCL00 Note 3	Port 1. 5-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port (in 1-bit units). Input of P10 and P11 can be set to TTL input buffer. Output of P10 to P12 can be set to N-ch open-drain output (V_{DD} tolerance). Can be set to analog input Note 1.
P11				ANI17/SI00/RxD0/ SDA00 Note 3/ TOOLRxD	
P12	7-3-2			ANI18/SO00/TxD0/ TOOLTxD	
P13	7-3-1			ANI19/TI00/TO00/ INTP2	
P14				ANI20/TI01/TO01/ INTP3	
P20	4-3-1	I/O	Analog input port	ANI0/AV_{REFP}	Port 2. 4-bit I/O port. Input/output can be specified in 1-bit units. Can be set to analog input Note 2.
P21				ANI1/AV_{REFM}	
P22				ANI2	
P23				ANI3	

<R>

Notes 1. Digital or analog for each pin can be selected with the port mode control register x (PMCx) (can be set in 1-bit unit).

2. Digital or analog for each pin can be selected with the A/D port configuration register (ADPC).

3. Provided only in the R5F102 products.

Remark Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR). For details, see **Figure 4-8 Format of Peripheral I/O Redirection Register (PIOR)**.

Downloaded from Arrow.com.

(2/2)

Function Name	Pin Type	I/O	After Reset	Alternate Function	Function
P40	7-1-1	I/O	Input port	KR0/TOOL0	Port 4. 3-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port (in 1-bit units). Output of P41 can be set to N-ch open-drain output (V_{DD} tolerance). P41 and P42 can be set to analog input Note 1.
P41	7-3-2		Analog input port	ANI22/SO01 Note 2/ SDA01 Note 2/TI02/ TO02/INTP1	
P42	7-3-1			ANI21/SCK01 Note 2/ SCL01 Note 2/TI03/ TO03	
P60	12-1-1	I/O	Input port	KR4/SCLA0/ (TxD0)	Port 6. 2-bit I/O port. Input/output can be specified in 1-bit units. N-ch open-drain output (6-V tolerance).
P61				KR5/SDAA0/ (RxD0)	
P121	2-1-1	Input	Input port	KR3/X1/ (TI03)/(INTP3)	Port 12. 3-bit input port. P125 is also used for the input pin for external reset ($\overline{\text{RESET}}$). To use the pin for external reset, set the PORTSELB bit in the option byte (000C1H) to 1. For P125, use of an on-chip pull-up resistor can be specified by a software setting.
P122				KR2/X2/EXCLK/ (TI02)/(INTP2)	
P125	7-1-1			KR1/SI01 Note 2/ $\overline{\text{RESET}}$	
P137	2-1-2	Input	Input port	INTP0	Port 13. 1-bit input port.

Notes 1. Digital or analog for each pin can be selected with the port mode control register x (PMCx) (can be set in 1-bit unit).

2. Provided only in the R5F102 products.

<R> **Remark** Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR). For details, see **Figure 4-8 Format of Peripheral I/O Redirection Register (PIOR)**.

Downloaded from Arrow.com.

2.1.3 30-pin products

(1/2)

<R>

Function Name	Pin Type	I/O	After Reset	Alternate Function	Function
P00	7-3-2	I/O	Analog input port	ANI17/TI00/TxD1 Note 3	Port 0. 2-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port (in 1-bit units). Input of P01 can be set to TTL input buffer. Output of P00 can be set to N-ch open-drain output (V_{DD} tolerance). Can be set to analog input Note 1.
P01	8-3-1			ANI16/TO00/RxD1 Note 3	
P10	8-1-2	I/O	Input port	SCK00/SCL00 Note 3/(TI07)/(TO07)	Port 1. 8-bit I/O port. Input/output can be specified in 1-bit units. Input of P10, P11, and P13 to P17 can be set to TTL input buffer. Output of P10 to P15 and P17 can be set to N-ch open-drain output (V_{DD} tolerance). Use of an on-chip pull-up resistor can be specified by a software setting at input port (in 1-bit units).
P11				SI00/RxD0/TOOLRxD/SDA00 Note 3/(TI06)/(TO06)	
P12	7-1-2			SO00/TxD0/TOOLTXD/(TI05)/(TO05)	
P13	8-1-2			TxD2 Note 3/SO20 Note 3/(SDAA0) Note 3/(TI04)/(TO04)	
P14				RxD2 Note 3/SI20 Note 3/SDA20 Note 3/(SCLA0) Note 3/(TI03)/(TO03)	
P15				PCLBUZ1/SCK20 Note 3/SCL20 Note 3/(TI02)/(TO02)	
P16	8-1-1			TI01/TO01/INTP5/(RxD0)	
P17	8-1-2			TI02/TO02/(TxD0)	
P20	4-3-1	I/O	Analog input port	ANI0/AV_{REFP}	Port 2. 4-bit I/O port. Input/output can be specified in 1-bit units. Can be set to analog input Note 2.
P21				ANI1/AV_{REFM}	
P22				ANI2	
P23				ANI3	

Notes 1. Digital or analog for each pin can be selected with the port mode control register x (PMCx) (can be set in 1-bit unit).

2. Digital or analog for each pin can be selected with the A/D port configuration register (ADPC).

3. Provided only in the R5F102 products.

<R> **Remark** Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR). For details, see **Figure 4-8 Format of Peripheral I/O Redirection Register (PIOR)**.

Downloaded from Arrow.com.

(2/2)

Function Name	Pin Type	I/O	After Reset	Alternate Function	Function
P30	7-1-1	I/O	Input port	INTP3/SCK11 Note 2/ SCL11 Note 2	Port 3. 2-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port (in 1-bit units).
P31				TI03/TO03/INTP4/ PCLBUZ0	
P40	7-1-1	I/O	Input port	TOOL0	Port 4. 1-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port .
P50	7-1-2	I/O	Input port	INTP1/SI11 Note 2/ SDA11 Note 2	Port 5. 2-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port (in 1-bit units). Output of P50 can be set to N-ch open-drain output (V_{DD} tolerance).
P51	7-1-1			INTP2/SO11 Note 2	
P60	12-1-1	I/O	Input port	SCLA0	Port 6. 2-bit I/O port. Input/output can be specified in 1-bit units. N-ch open-drain output (6V tolerance).
P61				SDAA0	
P120	7-3-1	I/O	Analog input port	ANI19	Port 12. 1-bit I/O port and 2-bit input port. Only for P120, input/output can be specified. Only for P120, use of an on-chip pull-up resistor can be specified by a software setting. P120 can be set to analog input Note 1.
P121	2-2-1	Input	Input port	X1	
P122				X2/EXCLK	
P137	2-1-2	Input	Input port	INTP0	Port 13. Dedicated 1-bit input port.
P147	7-3-1	I/O	Analog input port	ANI18	Port 14. 1-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. Can be set to analog input Note 1.
$\overline{\text{RESET}}$	2-1-1	Input	–	–	Dedicated input pin for external reset. When external reset is not used, connect this pin to V_{DD} directly or via a resistor.

Notes 1. Digital or analog for each pin can be selected with the port mode control register x (PMCx) (can be set in 1-bit unit).

2. Provided only in the R5F102 products.

<R> **Remark** Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR). For details, see **Figure 4-8 Format of Peripheral I/O Redirection Register (PIOR)**.

Downloaded from Arrow.com.

2.2 Functions other than port pins

2.2.1 Functions for each product

Function Name	20-pin products	24-pin products	30-pin products
ANI0	√	√	√
ANI1	√	√	√
ANI2	√	√	√
ANI3	√	√	√
ANI16	√	√	√
ANI17	√	√	√
ANI18	√	√	√
ANI19	√	√	√
ANI20	√	√	–
ANI21	√	√	–
ANI22	√	√	–
INTP0	√	√	√
INTP1	√	√	√
INTP2	√	√	√
INTP3	√	√	√
INTP4	–	–	√
INTP5	–	–	√
KR0	√	√	–
KR1	√	√	–
KR2	√	√	–
KR3	√	√	–
KR4	√	√	–
KR5	√	√	–
KR6	–	√	–
KR7	–	√	–
KR8	–	√	–
KR9	–	√	–
PCLBUZ0	√	√	√
PCLBUZ1	–	–	√
REGC	–	–	√
$\overline{\text{RESET}}$	√	√	√
RxD0	√	√	√
RxD1	–	–	√Note
RxD2	–	–	√Note
TxD0	√	√	√
TxD1	–	–	√Note
TxD2	–	–	√Note
SCK00	√	√	√
SCK01	√Note	√Note	–
SCK11	–	–	√Note
SCK20	–	–	√Note
SCLA0	√	√	√
SDAA0	√	√	√

Function Name	20-pin products	24-pin products	30-pin products
SCL00	√Note	√Note	√Note
SCL01	√Note	√Note	–
SCL11	–	–	√Note
SCL20	–	–	√Note
SDA00	√Note	√Note	√Note
SDA01	√Note	√Note	–
SDA11	–	–	√Note
SDA20	–	–	√Note
SI00	√	√	√
SI01	√Note	√Note	–
SI11	–	–	√Note
SI20	–	–	√Note
SO00	√	√	√
SO01	√Note	√Note	–
SO11	–	–	√Note
SO20	–	–	√Note
TI00	√	√	√
TI01	√	√	√
TI02	√	√	√
TI03	√	√	√
TI04	–	–	(√)
TI05	–	–	(√)
TI06	–	–	(√)
TI07	–	–	(√)
TO00	√	√	√
TO01	√	√	√
TO02	√	√	√
TO03	√	√	√
TO04	–	–	(√)
TO05	–	–	(√)
TO06	–	–	(√)
TO07	–	–	(√)
X1	√	√	√
X2	√	√	√
EXCLK	√	√	√
VDD	√	√	√
AVREFP	√	√	√
AVREFM	√	√	√
VSS	√	√	√
TOOLRxD	√	√	√
TOOLTxD	√	√	√
TOOL0	√	√	√

Note Provided only in the R5F102 products.

Remark Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR). For details, see **Figure 4-8 Format of Peripheral I/O Redirection Register (PIOR)**.

Downloaded from Arrow.com.

2.2.2 Description of Functions

Function Name	I/O	Functions
ANI0 to ANI3, ANI16 to ANI22	input	Analog input pins of A/D converter (see **Figure 10-44 Analog Input Pin Connection**)
AV_{REFP}	input	Inputs the A/D converter reference potential (+ side)
AV_{REFM}	input	Inputs the A/D converter reference potential (– side)
INTP0 to INTP5	input	External interrupt request input Specified available edge : rising edge, falling edge, or both rising and falling edges
KR0 to KR9	input	Key interrupt input
PCLBUZ0, PCLBUZ1	output	Clock/buzzer output
REGC	–	Connecting regulator output stabilization capacitance for internal operation. Connect this pin to V_{SS} via a capacitor (0.47 to 1 μF)
$\overline{\text{RESET}}$	input	External reset input for low level active When the external reset pin is not used, connect this pin directly or via a resistor to V_{DD}.
RxD0 to RxD2	input	Serial data input for serial interfaces UART0, UART1, and UART2
TxD0 to TxD2	output	Serial data output for serial interfaces UART0, UART1, and UART2
SCK00, SCK01, SCK11, SCK20	I/O	Serial clock I/O for serial interfaces CSI00, CSI01, CSI11, and CSI20
SI00, SI01, SI11, SI20	input	Serial data input for serial interfaces CSI00, CSI01, CSI11, and CSI20
SO00, SO01, SO11, SO20	output	Serial data output for serial interfaces CSI00, CSI01, CSI11, and CSI20
SCLA0	I/O	Serial clock I/O for serial interface IICA
SDAA0	I/O	Serial data I/O for serial interface IICA
SCL00, SCL01, SCL11, SCL20	output	Clock output for simplified I^2C serial interfaces IIC00, IIC01, IIC11, IIC20
SDA00, SDA01, SDA11, SDA20	I/O	Serial data I/O for simplified I^2C serial interfaces IIC00, IIC01, IIC11, IIC20
TI00 to TI07	input	Inputting an external count clock/capture trigger to 16-bit timers 00 to 07
TO00 to TO07	output	Timer output pins of 16-bit timers 00 to 07
X1, X2	–	Connecting a resonator for main system clock
EXCLK	input	External clock input pin for main system clock
V_{DD}	–	Positive power supply
V_{SS}	–	Ground potential
TOOLRxD	input	This UART serial data input pin for an external device connection is used during flash memory programming
TOOLTxD	output	This UART serial data output pin for an external device connection is used during flash memory programming
TOOL0	I/O	Data I/O pin for a flash memory programmer/debugger

Caution The following shows the relationship between P40/TOOL0 and the operation mode when reset is released.

Table 2-1. Relationship between P40/TOOL0 and the Operation Mode When Reset Is Released

P40/TOOL0	Operation mode
V_{DD}	Normal operation mode
0 V	Flash memory programming mode

For details, see **24.4 Serial Programming Method**.

Remark Use bypass capacitors (about 0.1 µF) as noise and latch up countermeasures with relatively thick wires at the shortest distance to V_{DD} to V_{SS} lines.

Downloaded from Arrow.com.

2.3 Connection of Unused Pins

Tables 2-2 and 2-3 show the connections of unused pins.

The pins mounted depend on the product. Refer to **1.4 Pin Configuration (Top View)** and **2.1 Port Functions**.

Table 2-2. Connection of Unused Pins (20-, 24-pin products)

Pin Name	I/O	Recommended Connection of Unused Pins
P00 to P03 Note	I/O	Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P10 to P14		
P20 to P23		
P40/TOOL0		Input: Leave open or independently connect to V_{DD} via a resistor. Output: Leave open.
P41, P42		Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P60, P61		Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Set the port's output latch to 0 and leave the pins open, or set the port's output latch to 1 and independently connect the pins to V_{DD} or V_{SS} via a resistor.
P121, P122	input	Independently connect to V_{DD} or V_{SS} via a resistor.
P125/$\overline{\text{RESET}}$		PORTSELB = 0: Independently connect to V_{DD} or V_{SS} via a resistor. PORTSELB = 1: Leave open or connect to V_{DD}.
P137		Independently connect to V_{DD} or V_{SS} via a resistor.

Note Provided only in 24-pin products.

Table 2-3. Connection of Unused Pins (30-pin products)

Pin Name	I/O	Recommended Connection of Unused Pins
P00, P01	I/O	Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P10 to P17		
P20 to P23		
P30, P31		
P40/TOOL0		
P50, P51		
P60, P61		Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Set the port's output latch to 0 and leave the pins open, or set the port's output latch to 1 and independently connect the pins to V_{DD} or V_{SS} via a resistor.
P120		Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P121, P122	input	Input: Independently connect to V_{DD} or V_{SS} via a resistor.
P137		
P147	I/O	Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
$\overline{\text{RESET}}$	input	Connect to V_{DD} directly or via a resistor.
REGC	–	Connect to V_{SS} via a capacitor (0.47 to 1 F).

Downloaded from Arrow.com.

<R> 2.4 Block Diagrams of Pins

Figures 2-1 to 2-13 show the block diagrams of the pins described in **2.1.1 20-pin products** to **2.1.3 30-pin products**.

Figure 2-1. Pin Block Diagram for Pin Type 2-1-1

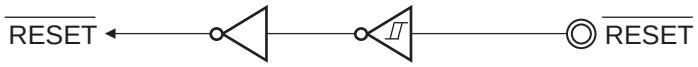
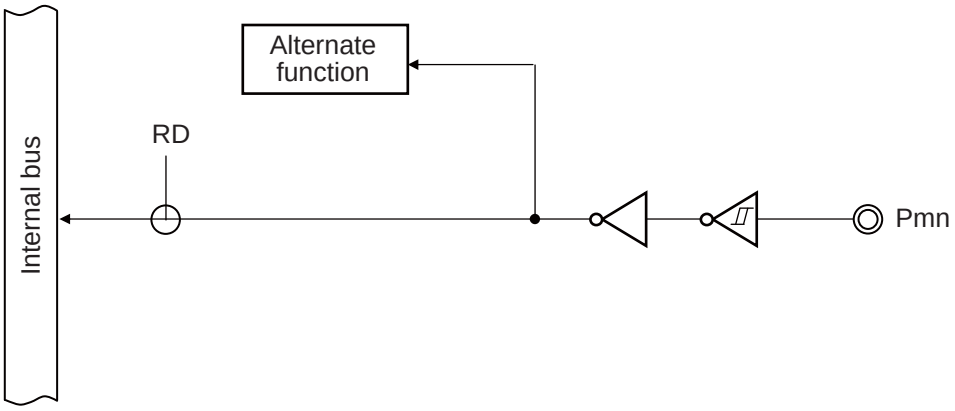


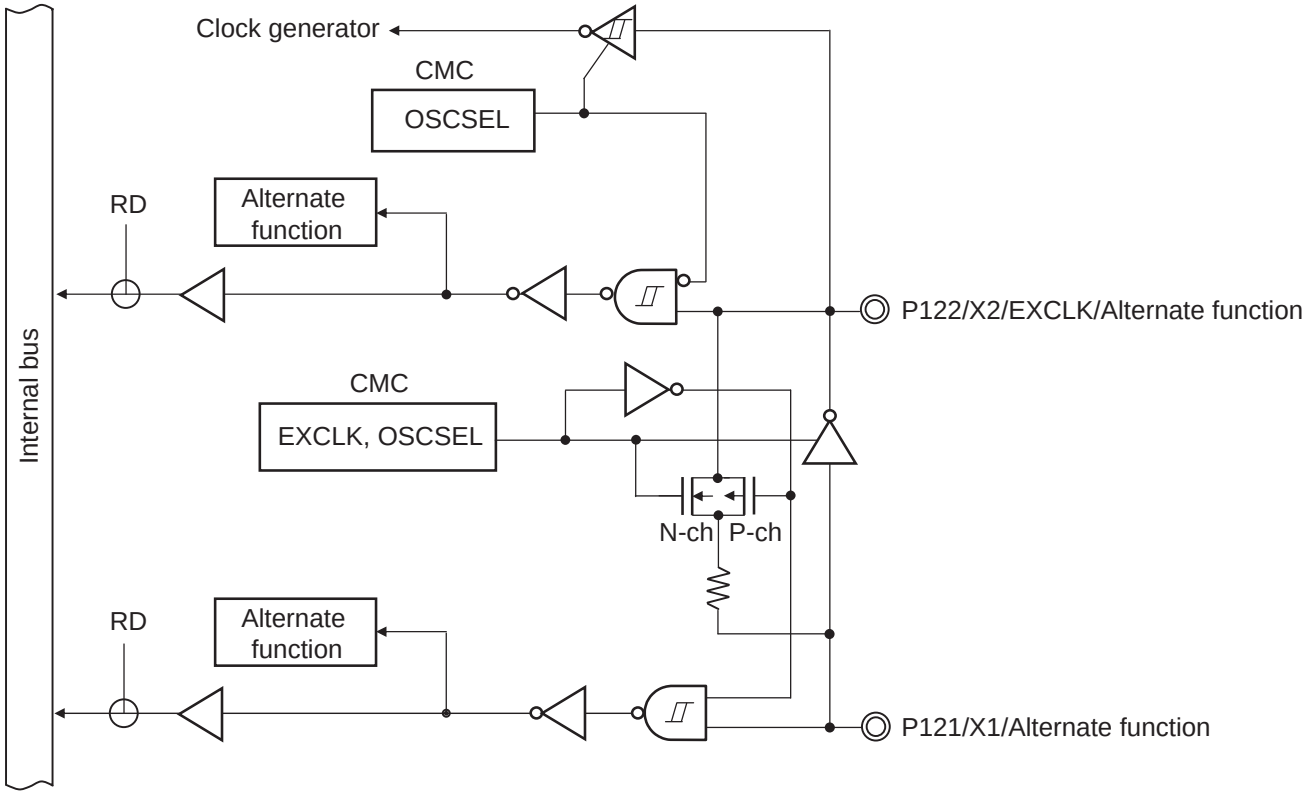
Figure 2-2. Pin Block Diagram for Pin Type 2-1-2



Remark For alternate functions, see **2.1 Port Functions**.

Downloaded from Arrow.com.

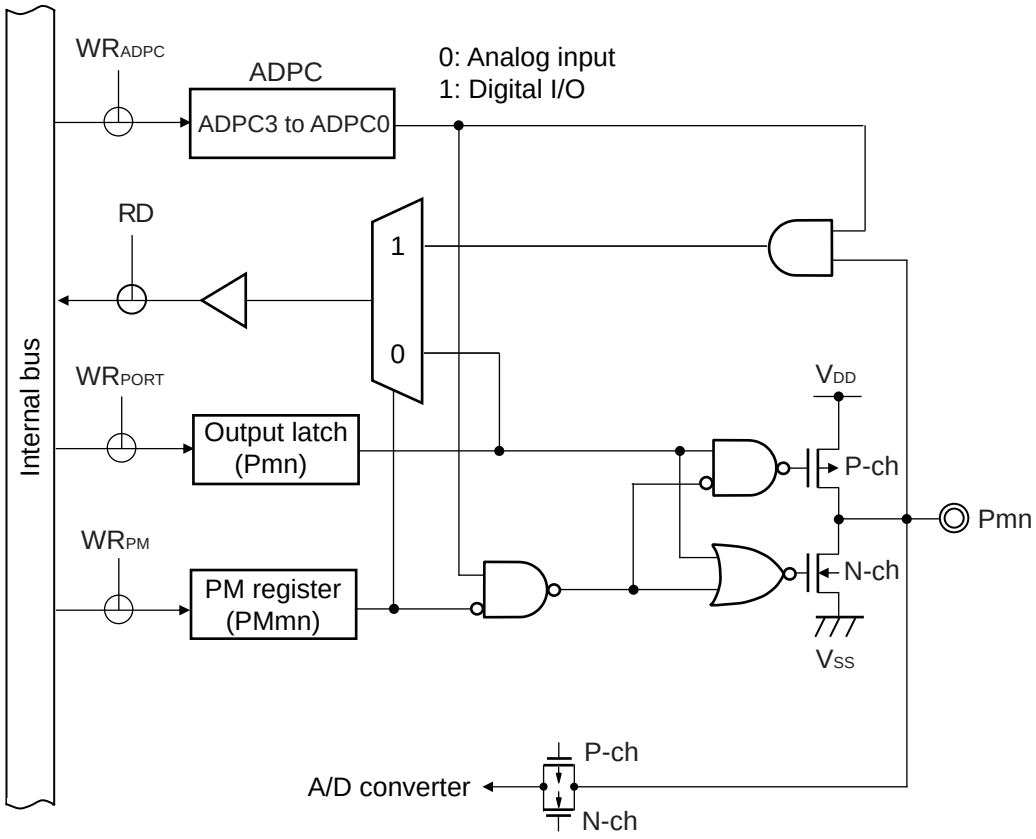
Figure 2-3. Pin Block Diagram for Pin Type 2-2-1



Remark For alternate functions, see **2.1 Port Functions**.

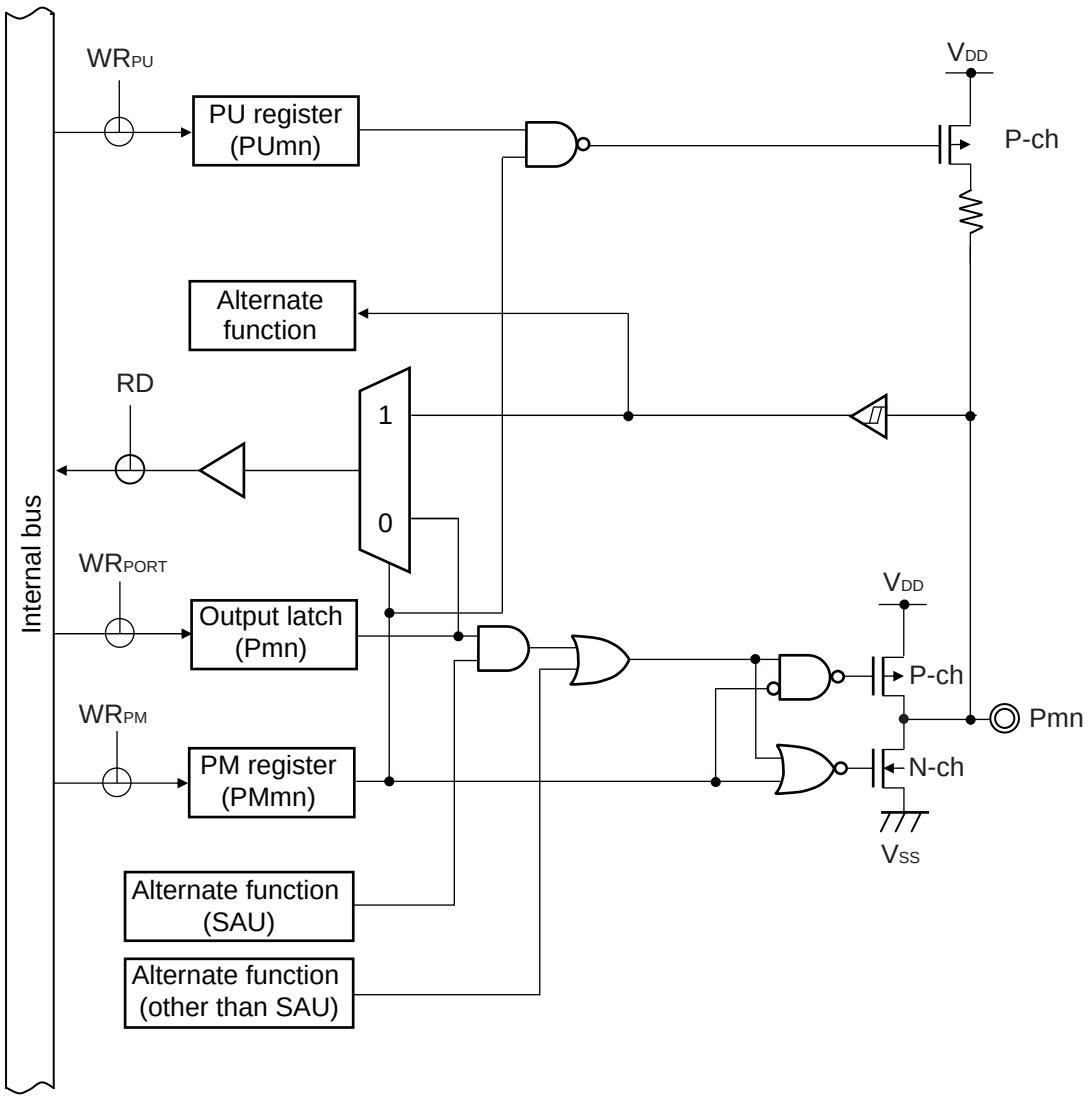
Downloaded from Arrow.com.

Figure 2-4. Pin Block Diagram for Pin Type 4-3-1



Downloaded from Arrow.com.

Figure 2-5. Pin Block Diagram for Pin Type 7-1-1

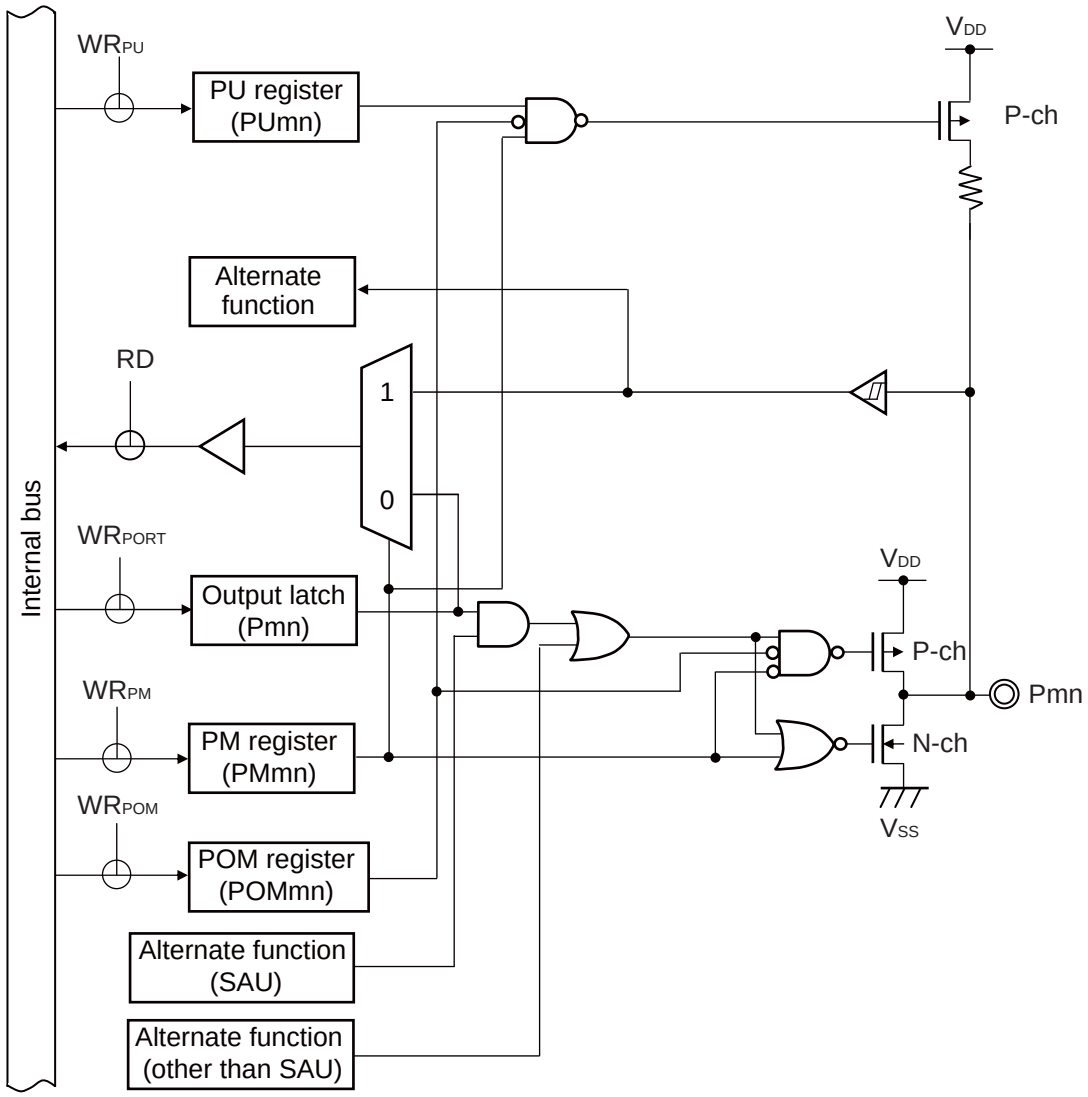


Remarks 1. For alternate functions, see **2.1 Port Functions**.

2. SAU: Serial array unit

Downloaded from Arrow.com.

Figure 2-6. Pin Block Diagram for Pin Type 7-1-2

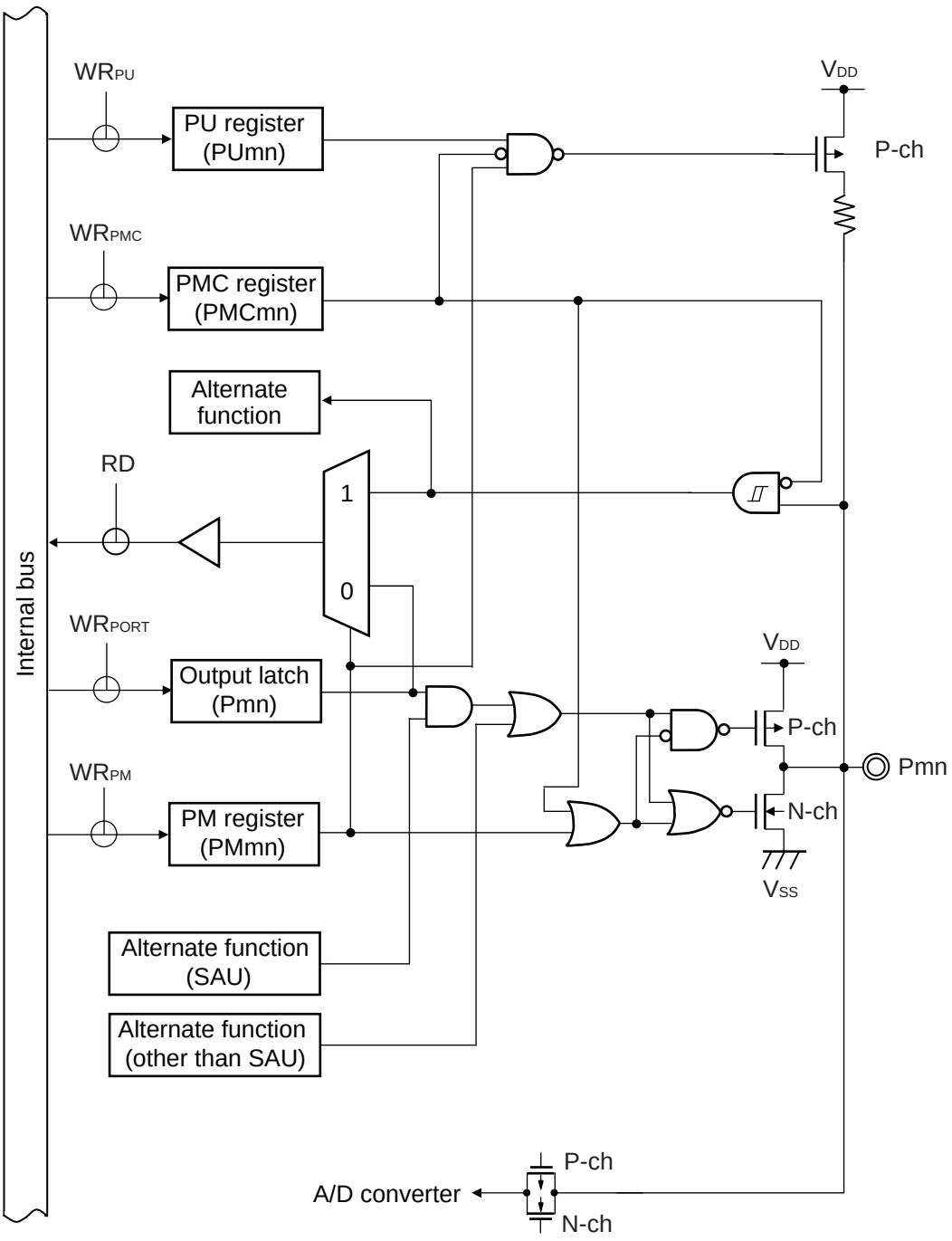


Remarks 1. For alternate functions, see **2.1 Port Functions**.

2. SAU: Serial array unit

Downloaded from Arrow.com.

Figure 2-7. Pin Block Diagram for Pin Type 7-3-1

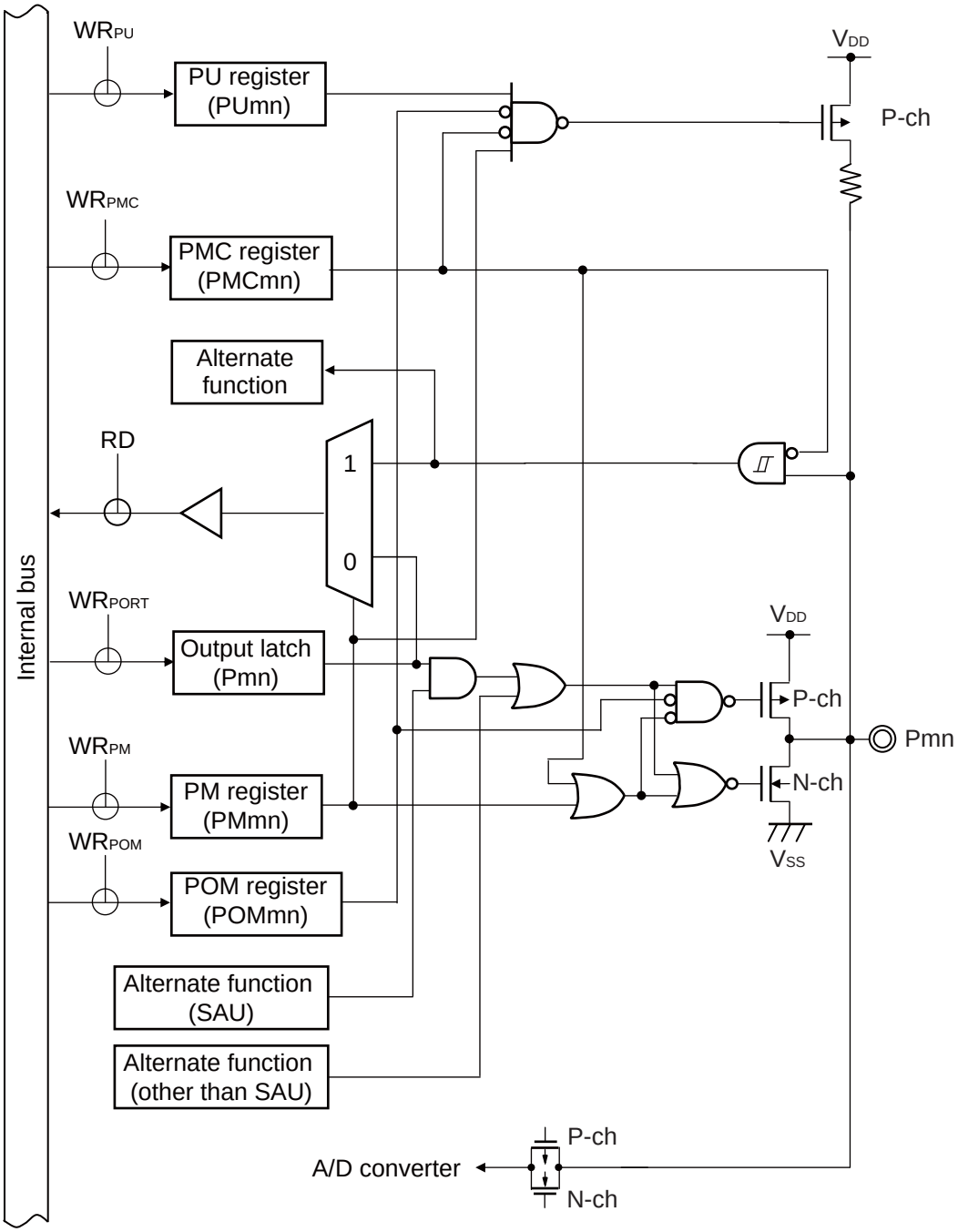


Remarks 1. For alternate functions, see **2.1 Port Functions**.

2. SAU: Serial array unit

Downloaded from Arrow.com.

Figure 2-8. Pin Block Diagram for Pin Type 7-3-2

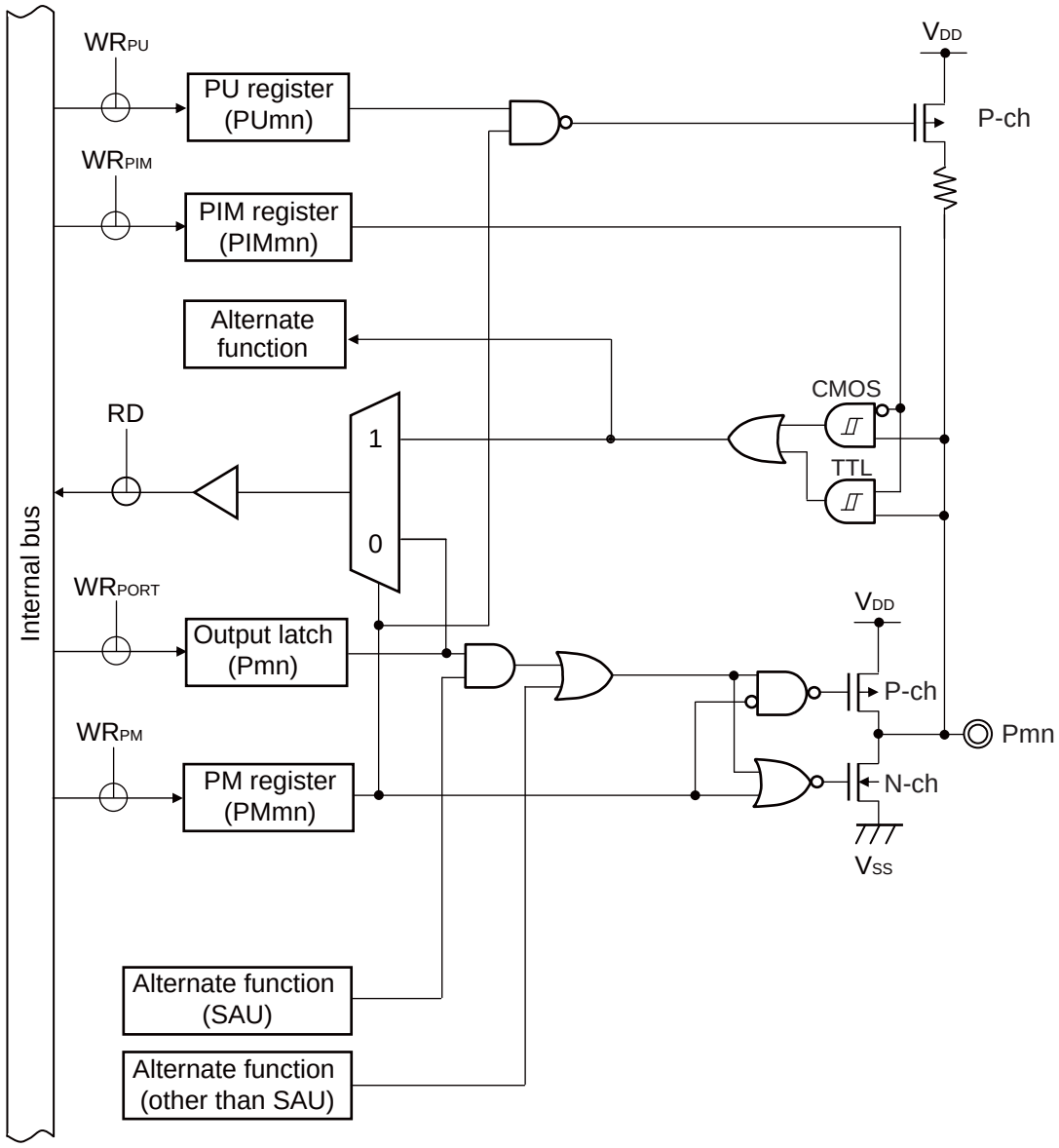


Remarks 1. For alternate functions, see **2.1 Port Functions**.

2. SAU: Serial array unit

Downloaded from Arrow.com.

Figure 2-9. Pin Block Diagram for Pin Type 8-1-1

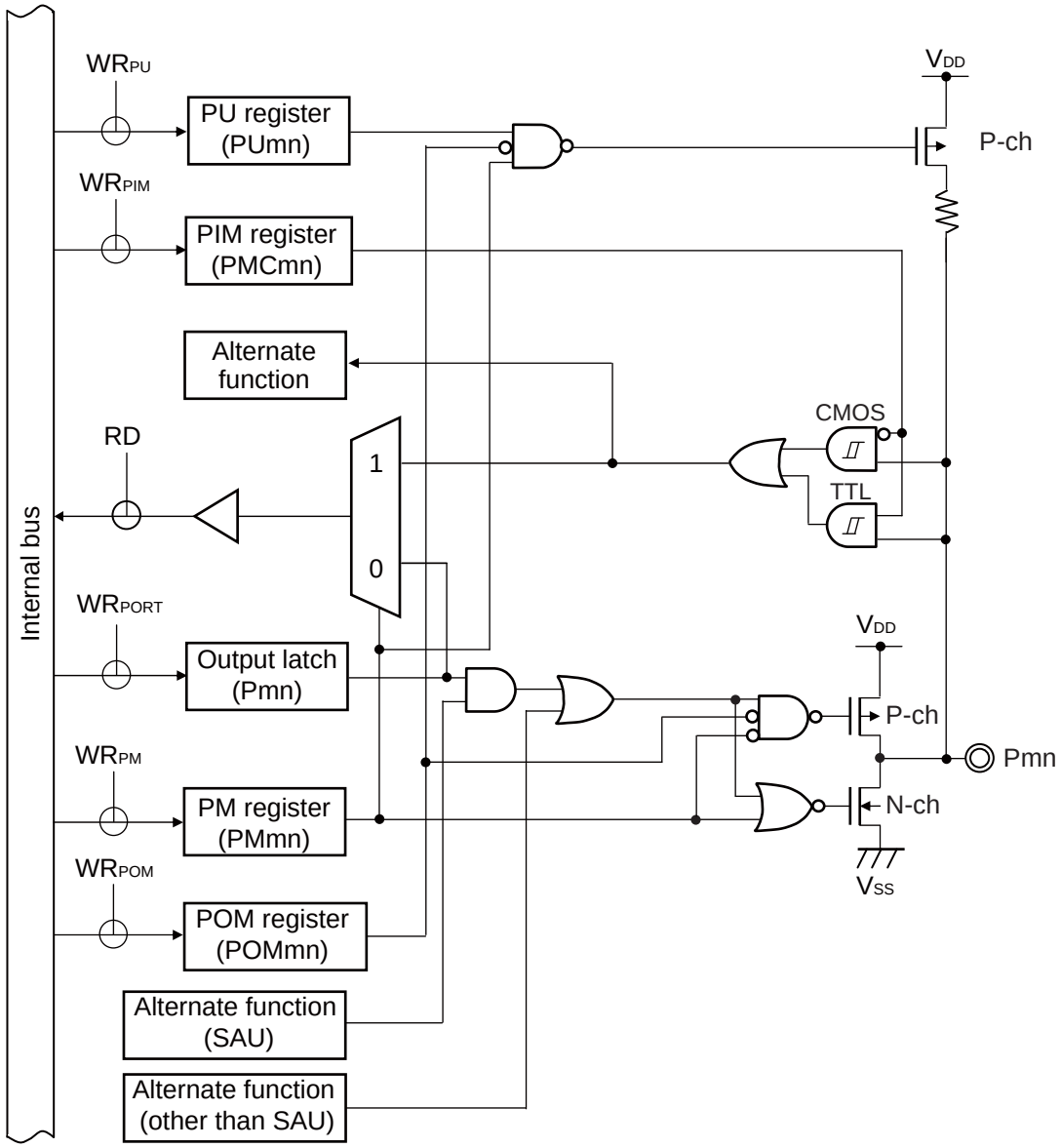


Remarks 1. For alternate functions, see **2.1 Port Functions**.

2. SAU: Serial array unit

Downloaded from Arrow.com.

Figure 2-10. Pin Block Diagram for Pin Type 8-1-2

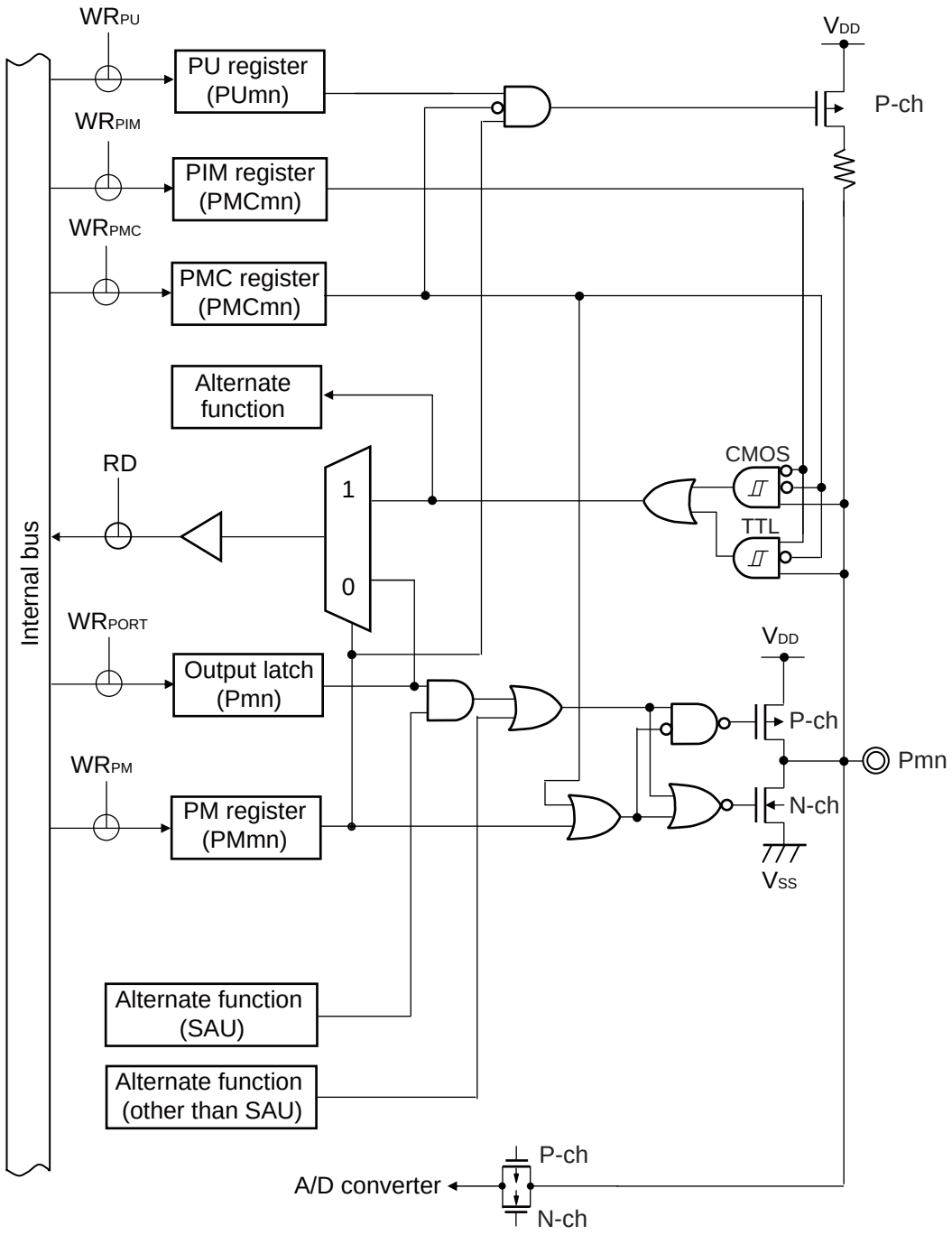


Remarks 1. For alternate functions, see **2.1 Port Functions**.

2. SAU: Serial array unit

Downloaded from Arrow.com.

Figure 2-11. Pin Block Diagram for Pin Type 8-3-1

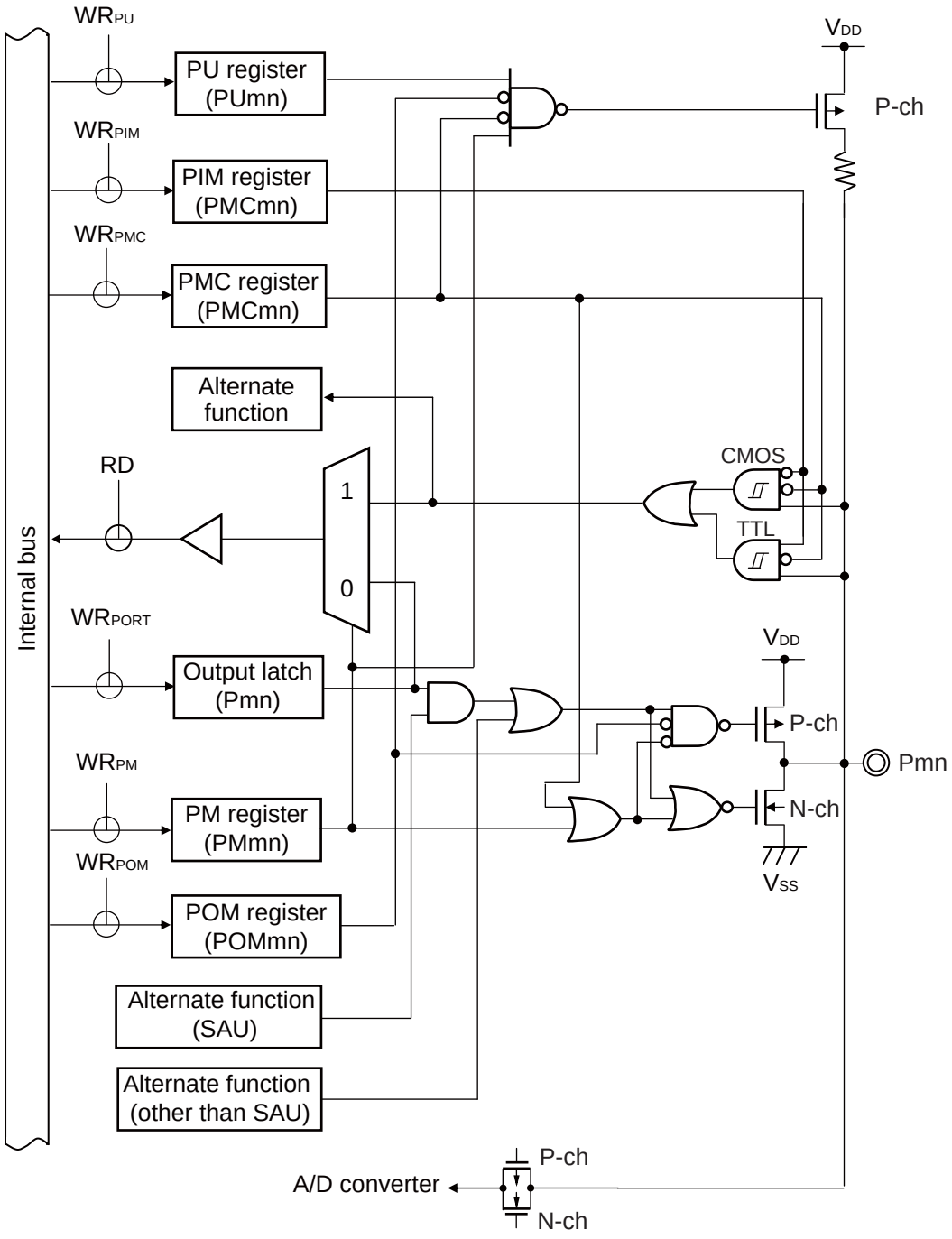


Remarks 1. For alternate functions, see **2.1 Port Functions**.

2. SAU: Serial array unit

Downloaded from Arrow.com.

Figure 2-12. Pin Block Diagram for Pin Type 8-3-2

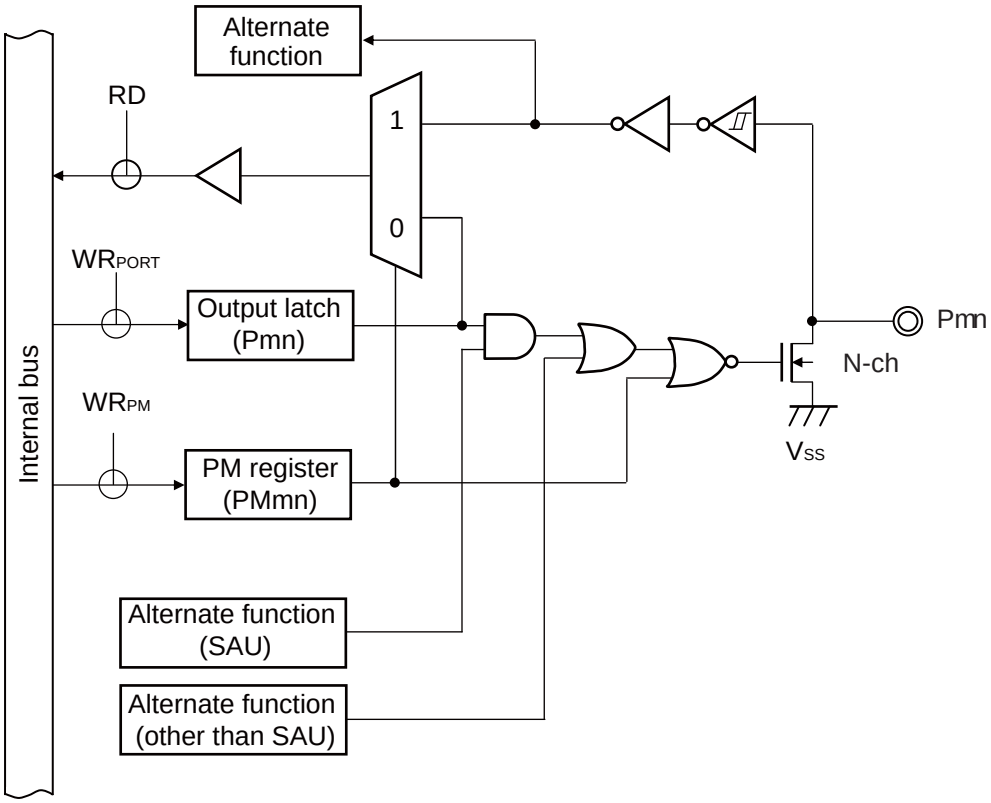


Remarks 1. For alternate functions, see **2.1 Port Functions**.

2. SAU: Serial array unit

Downloaded from Arrow.com.

Figure 2-13. Pin Block Diagram for Pin Type 12-1-1



Remarks 1. For alternate functions, see **2.1 Port Functions**.

2. SAU: Serial array unit

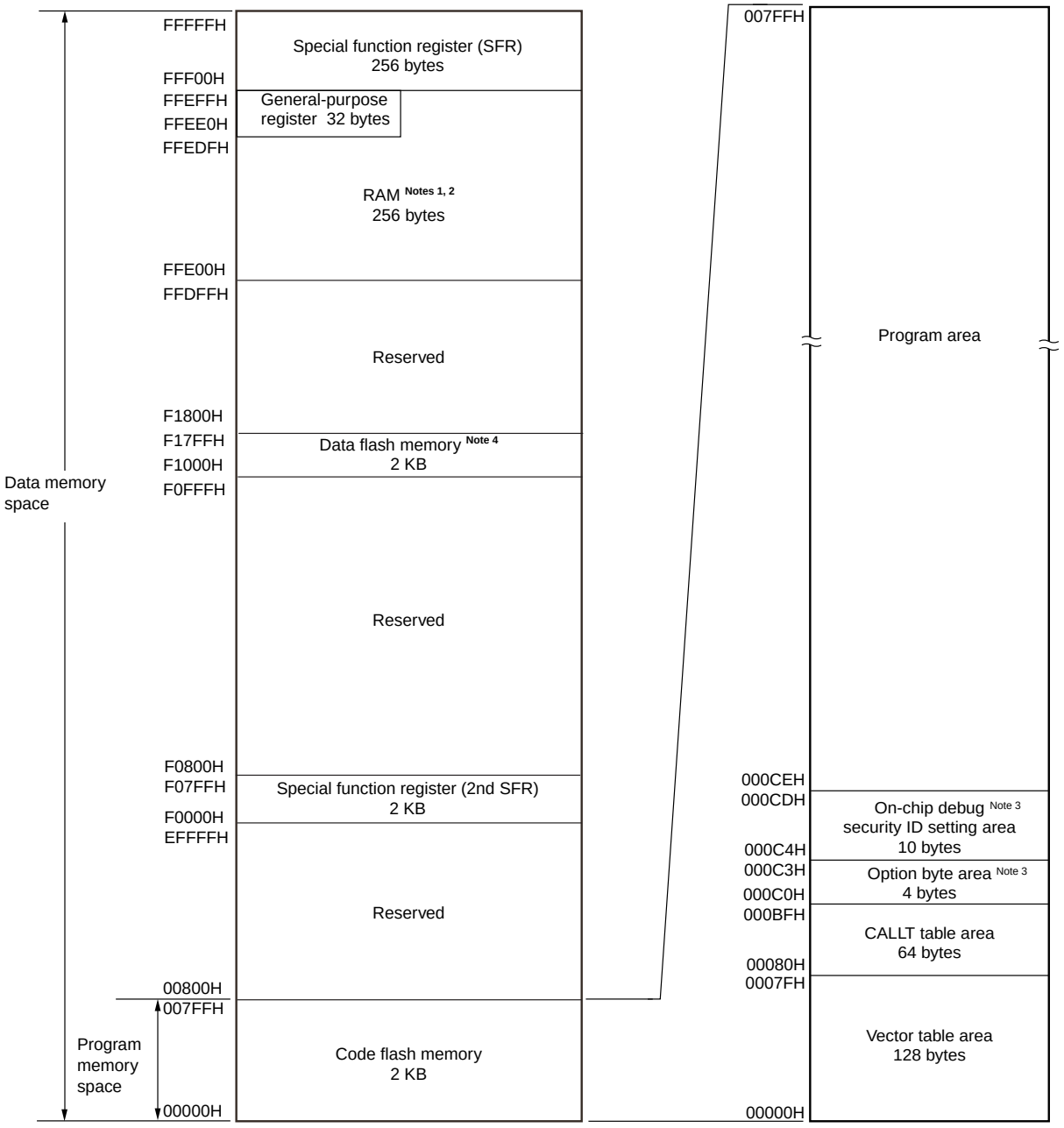
Downloaded from Arrow.com.

CHAPTER 3 CPU ARCHITECTURE

3.1 Memory Space

Products in the RL78/G12 can access a 1-MB memory space. Figures 3-1 to 3-6 show the memory maps.

Figure 3-1. Memory Map for the R5F10266 and R5F10366



Notes 1. Allocate the stack used for the data flash library to FFEA2H to FFEDFH and the RAM address used for the data buffer and DMA transfer to FFE00H to FFE1FH when rewriting the data flash memory. For details, refer to RL78 Family Data Flash Library Type04 User's Manual.

2. Instructions can be executed from the RAM area excluding the general-purpose register area.

3. Set the option bytes to 000C0H to 000C3H, and the on-chip debug security IDs to 000C4H to 000CDH.

<R> **4.** The areas are reserved in the R5F10366.

Downloaded from Arrow.com.

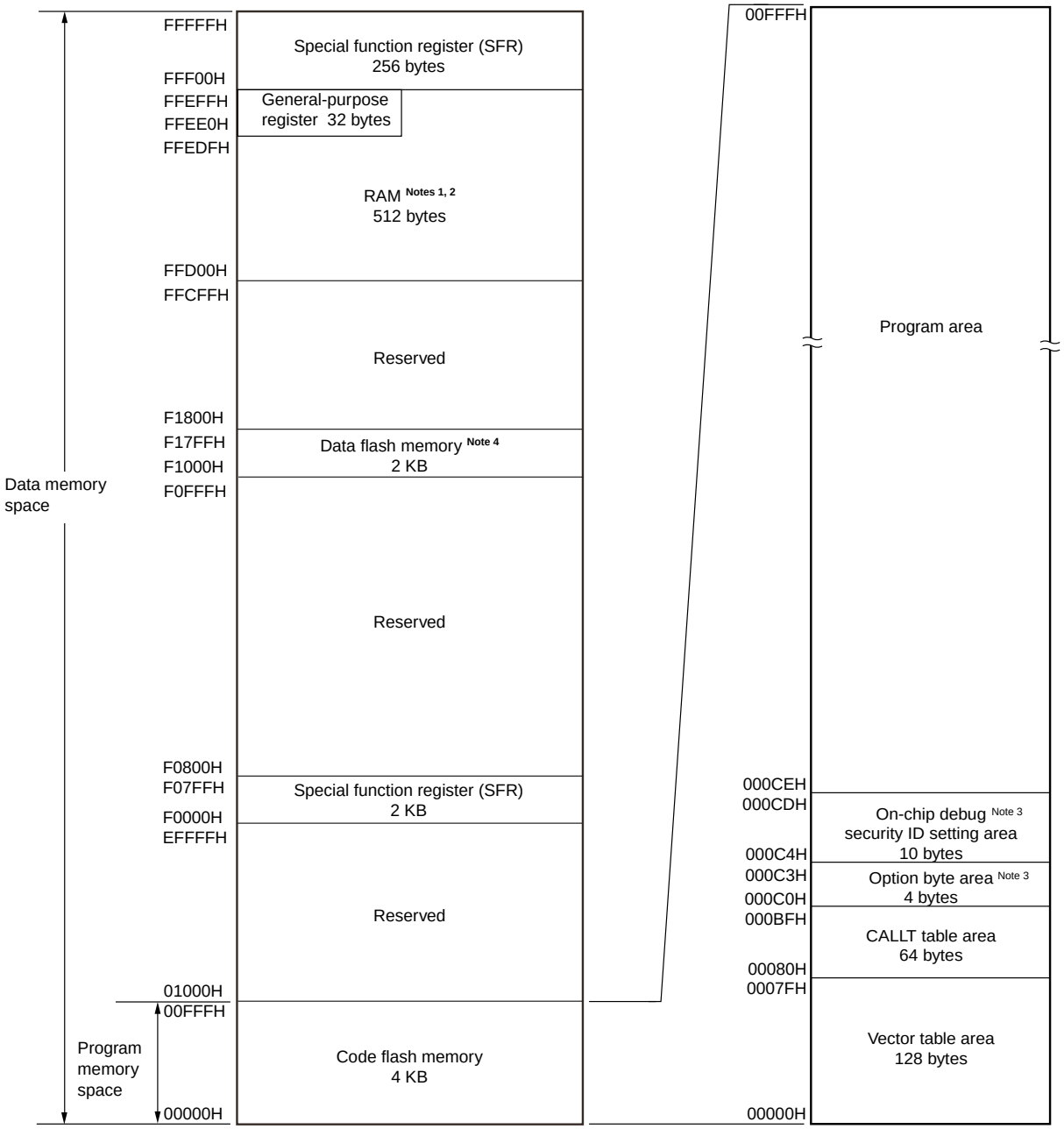
Cautions 1. While RAM parity error resets are enabled (RPERDIS = 0), be sure to initialize RAM areas where data access is to proceed and the RAM area +10 bytes when instructions are fetched from RAM areas, respectively. Reset signal generation sets RAM parity error resets to enabled (RPERDIS = 0). For details, see 21.3.2 RAM parity error detection function.

2. The RAM in the R5F10266 has capacity as small as 256 bytes. Depending on the customer's program specification, the stack area to execute the data flash library may not be kept and data may not be written to or erased from the data flash memory. For details, refer to RL78 Family Data Flash Library Type04 User's Manual.

3. The self-programming function cannot be used in the R5F10266 and R5F10366

Downloaded from Arrow.com.

Figure 3-2. Memory Map for the R5F10x67, and R5F10x77, and R5F10xA7 (x = 2 or 3)



<R> **Notes 1.** Do not allocate RAM addresses which are used as a stack area, a data buffer, and a DMA transfer destination/transfer source to the area FFE20H to FFEFFH when performing self-programming and rewriting the data flash memory.

2. Instructions can be executed from the RAM area excluding the general-purpose register area.

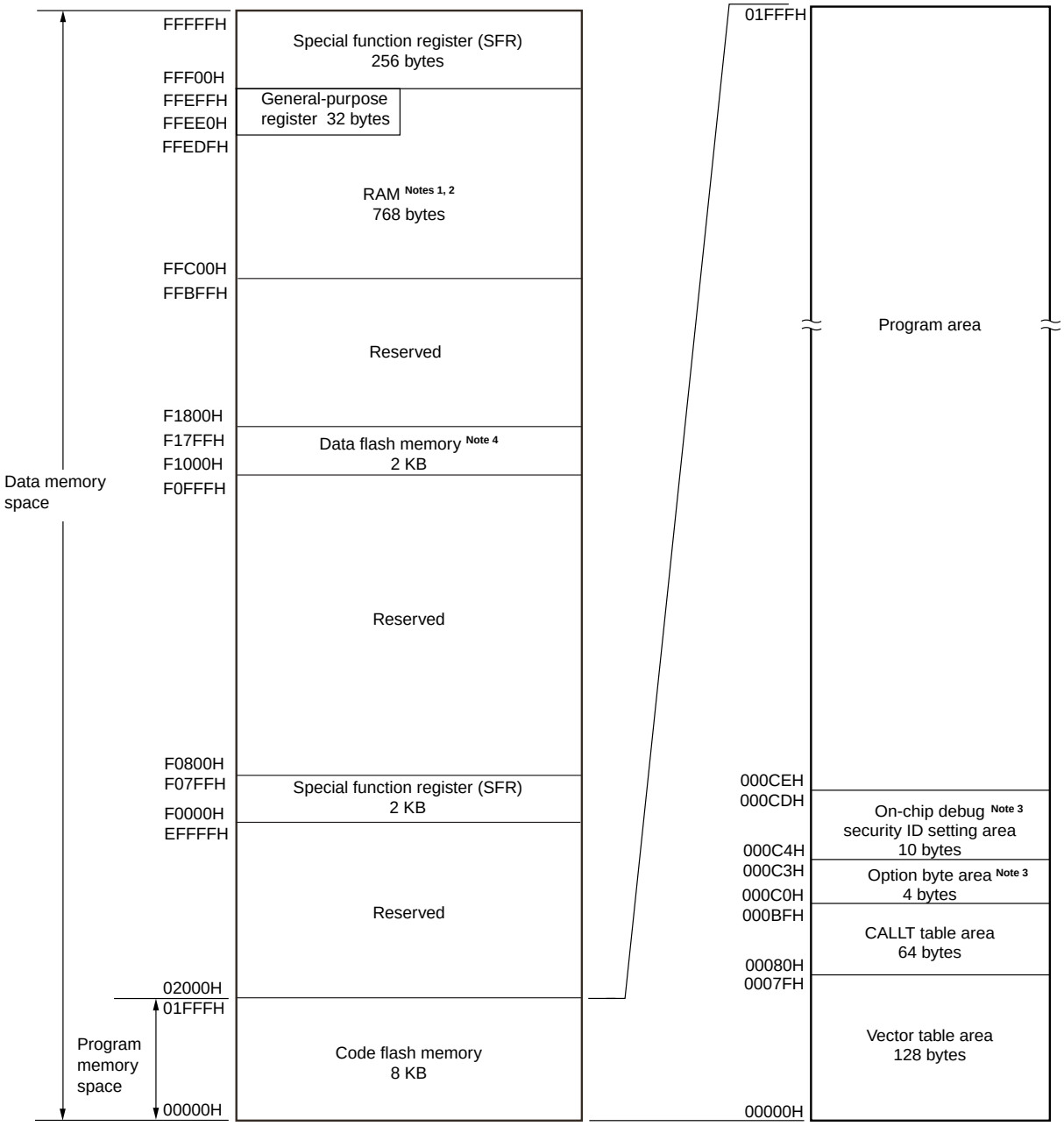
3. Set the option bytes to 000C0H to 000C3H, and the on-chip debug security IDs to 000C4H to 000CDH.

<R> **4.** The areas are reserved in the R5F10367, R5F10377, and R5F103A7.

Caution While RAM parity error resets are enabled (RPERDIS = 0), be sure to initialize RAM areas where data access is to proceed and the RAM area +10 bytes when instructions are fetched from RAM areas, respectively. Reset signal generation sets RAM parity error resets to enabled (RPERDIS = 0). For details, see 21.3.2 RAM parity error detection function.

Downloaded from Arrow.com.

Figure 3-3. Memory Map for the R5F10x68, R5F10x78, and R5F10xA8 (x = 2 or 3)



<R> **Notes 1.** Do not allocate RAM addresses which are used as a stack area, a data buffer, and a DMA transfer destination/transfer source to the area FFE20H to FFEFFH when performing self-programming and rewriting the data flash memory. Also, use of the area FFC00H to FFC89H in the R5F10x68 and R5F10x78 is prohibited, because this area is used for each library.

2. Instructions can be executed from the RAM area excluding the general-purpose register area.

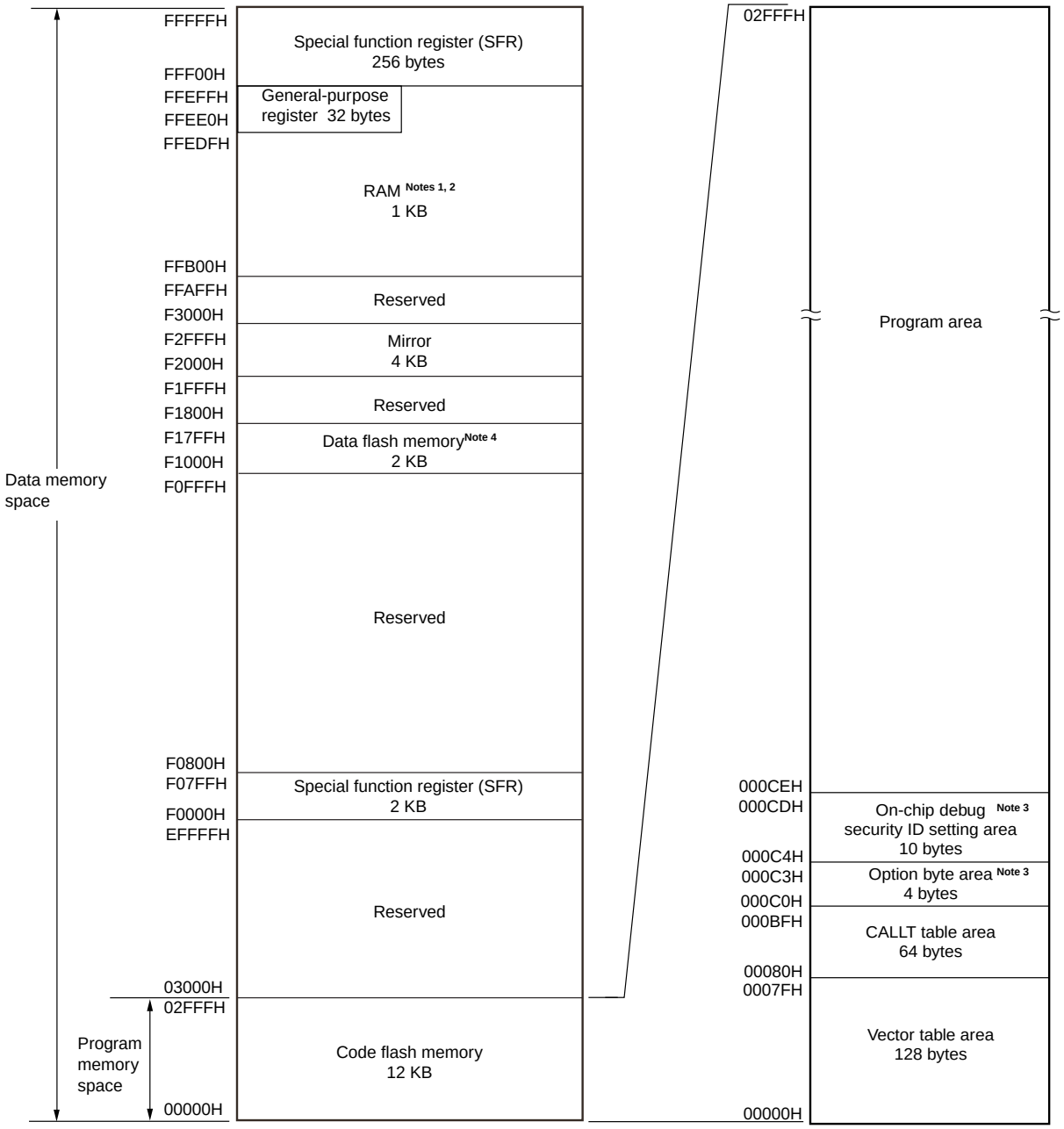
3. Set the option bytes to 000C0H to 000C3H, and the on-chip debug security IDs to 000C4H to 000CDH.

<R> **4.** The areas are reserved in the R5F10368, R5F10378, and R5103A8.

Caution While RAM parity error resets are enabled (RPERDIS = 0), be sure to initialize RAM areas where data access is to proceed and the RAM area +10 bytes when instructions are fetched from RAM areas, respectively. Reset signal generation sets RAM parity error resets to enabled (RPERDIS = 0). For details, see 21.3.2 RAM parity error detection function.

Downloaded from Arrow.com.

Figure 3-4. Memory Map for the R5F10x69, R5F10x79, and R5F10xA9 (x = 2 or 3)



<R> **Notes 1.** Do not allocate RAM addresses which are used as a stack area, a data buffer, and a DMA transfer destination/transfer source to the area FFE20H to FFEFFH when performing self-programming and rewriting the data flash memory. Also, use of the area FFB00H to FFC89H in the R5F10x69 and R5F10x79 is prohibited, because this area is used for each library.

2. Instructions can be executed from the RAM area excluding the general-purpose register area.

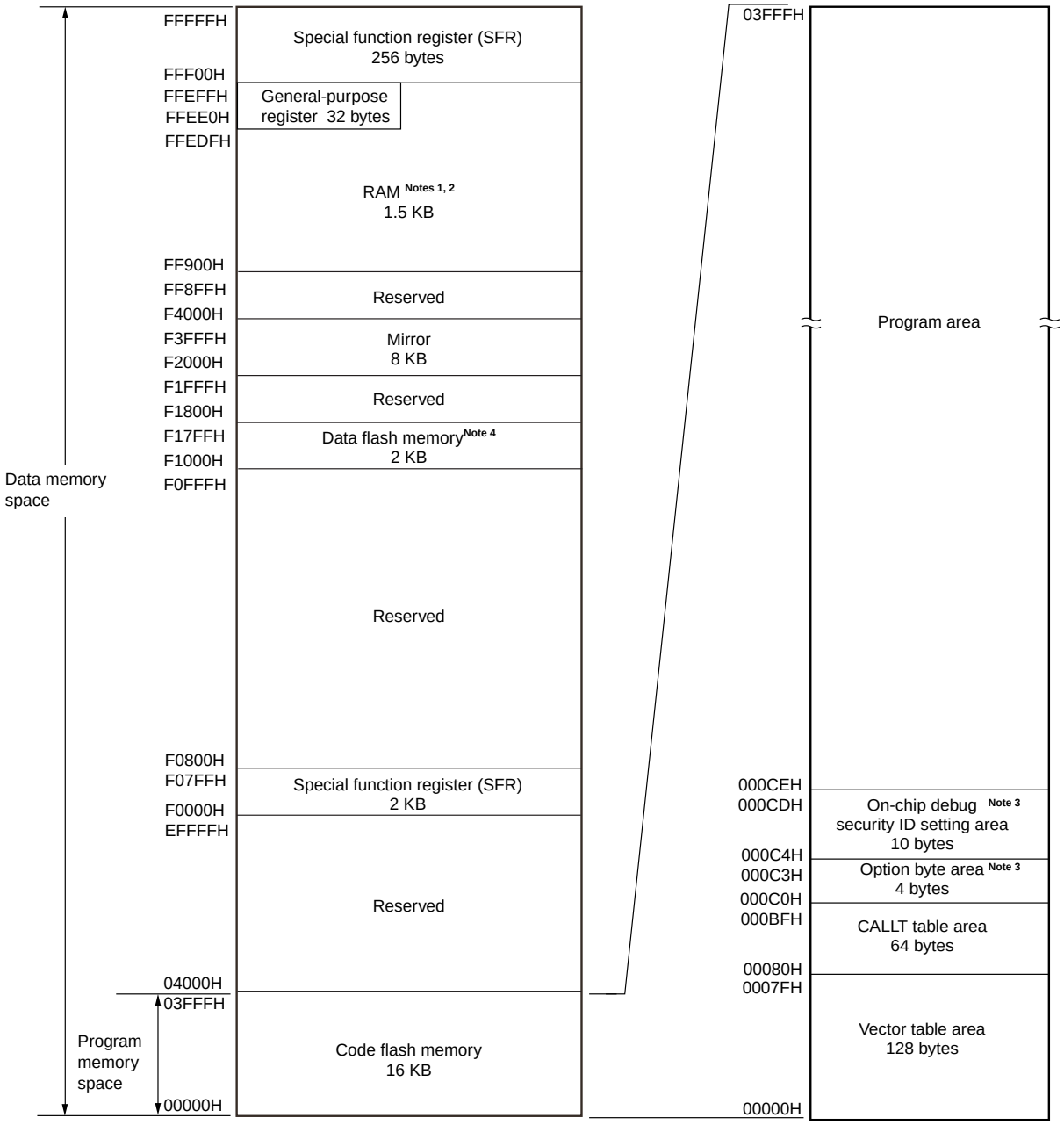
3. Set the option bytes to 000C0H to 000C3H, and the on-chip debug security IDs to 000C4H to 000CDH.

<R> **4.** The areas are reserved in the R5F10369, R5F10379, and R5F103A9.

Caution While RAM parity error resets are enabled (RPERDIS = 0), be sure to initialize RAM areas where data access is to proceed and the RAM area +10 bytes when instructions are fetched from RAM areas, respectively. Reset signal generation sets RAM parity error resets to enabled (RPERDIS = 0). For details, see 21.3.2 RAM parity error detection.

Downloaded from Arrow.com.

Figure 3-5. Memory Map for the R5F10x6A, R5F10x7A (x = 2 or 3)



<R> **Notes 1.** Do not allocate RAM addresses which are used as a stack area, a data buffer, and a DMA transfer destination/transfer source to the area FFE20H to FFEFFH when performing self-programming and rewriting the data flash memory. Also, use of the area FF900H to FFC89H in the R5F10x6A and R5F10x7A is prohibited, because this area is used for each library.

2. Instructions can be executed from the RAM area excluding the general-purpose register area.

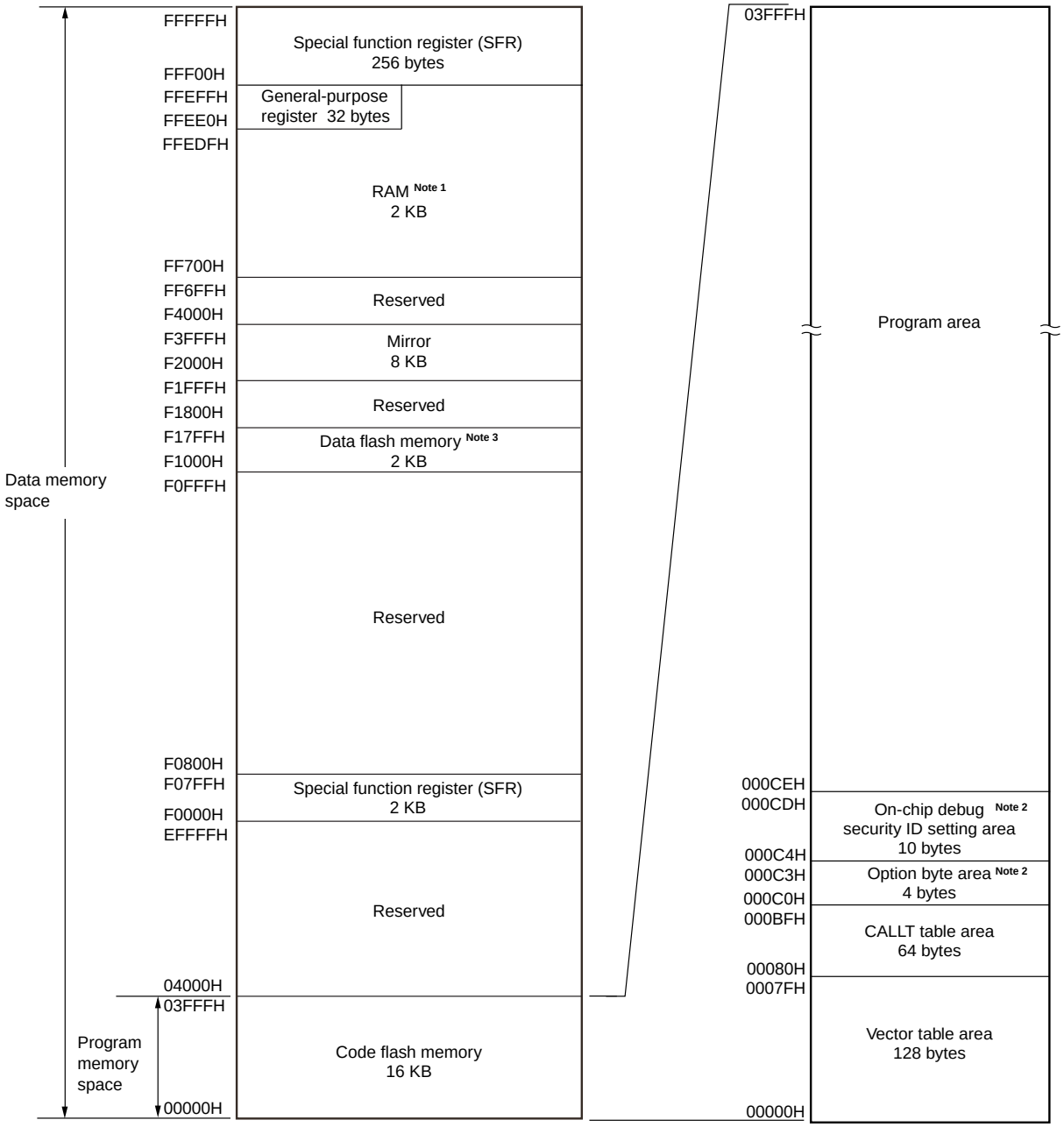
3. Set the option bytes to 000C0H to 000C3H, and the on-chip debug security IDs to 000C4H to 000CDH.

<R> **4.** The areas are reserved in the R5F1036A and R5F1037A.

Caution While RAM parity error resets are enabled (RPERDIS = 0), be sure to initialize RAM areas where data access is to proceed and the RAM area +10 bytes when instructions are fetched from RAM areas, respectively. Reset signal generation sets RAM parity error resets to enabled (RPERDIS = 0). For details, see 21.3.2 RAM parity error detection function.

Downloaded from Arrow.com.

Figure 3-6. Memory Map for the R5F10xAA (x = 2 or 3)



<R> **Notes 1.** Do not allocate RAM addresses which are used as a stack area, a data buffer, and a DMA transfer destination/transfer source to the area FFE20H to FFEFFH when performing self-programming and rewriting the data flash memory.

2. Instructions can be executed from the RAM area excluding the general-purpose register area.

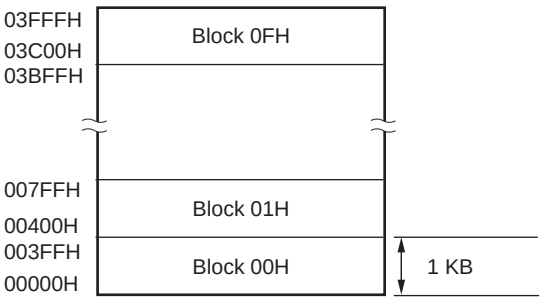
3. Set the option bytes to 000C0H to 000C3H, and the on-chip debug security IDs to 000C4H to 000CDH.

<R> **4.** The areas are reserved in the R5F103AA.

Caution While RAM parity error resets are enabled (RPERDIS = 0), be sure to initialize RAM areas where data access is to proceed and the RAM area +10 bytes when instructions are fetched from RAM areas, respectively. Reset signal generation sets RAM parity error resets to enabled (RPERDIS = 0). For details, see 21.3.2 RAM parity error detection.

Downloaded from Arrow.com.

Remark The flash memory is divided into blocks (one block = 1 KB). For the address values and block numbers, see **Table 3-1 Correspondence Between Address Values and Block Numbers in Flash Memory**.



(For the R5F1026A and R5F1027A)

Correspondence between the address values and block numbers in the flash memory are shown below.

Table 3-1. Correspondence Between Address Values and Block Numbers in Flash Memory

Address Value	Block Number	product
00000H to 003FFH	00H	R5F10x66
00400H to 007FFH	01H	
00800H to 00BFFH	02H	R5F10x67
00C00H to 00FFFH	03H	R5F10x77
01000H to 013FFH	04H	R5F10x68
01400H to 017FFH	05H	R5F10x78
01800H to 01BFFH	06H	
01C00H to 01FFFH	07H	
02000H to 023FFH	08H	R5F10x69
02400H to 027FFH	09H	R5F10x79
02800H to 02BFFH	0AH	
02C00H to 02FFFH	0BH	
03000H to 033FFH	0CH	R5F10x6A
03400H to 037FFH	0DH	R5F10x7A
03800H to 03BFFH	0EH	R5F10xAA
03C00H to 03FFFH	0FH	

(x = 2, 3)

Downloaded from Arrow.com.

3.1.1 Internal program memory space

The internal program memory space stores the program and table data.

The RL78/G12 products incorporate internal ROM (flash memory), as shown below.

Table 3-2. Internal ROM Capacity

Part Number	Internal ROM	
	Structure	Capacity
R5F10x66	Flash memory	2048 × 8 bits (00000H to 07FFFH)
R5F10x67, R5F10x77, R5F10xA7		4096 × 8 bits (00000H to 00FFFH)
R5F10x68, R5F10x78, R5F10xA8		8192 × 8 bits (00000H to 01FFFH)
R5F10x69, R5F10x79, R5F10xA9		12288 × 8 bits (00000H to 02FFFH)
R5F10x6A, R5F10x7A, R5F10xAA		16384 × 8 bits (00000H to 03FFFH)

(x = 2 or 3)

The internal program memory space is divided into the following areas.

(1) Vector table area

The 128-byte area of 00000H to 0007FH is reserved as a vector table area. The program start addresses for branch upon reset or generation of each interrupt request are stored in the vector table area. Furthermore, the interrupt jump addresses are assigned to a 64 KB address area of 00000H to 0FFFFH, because the vector code is 2 bytes.

Of 16-bit addresses, the lower 8 bits are stored at even addresses and the higher 8 bits are stored at odd addresses.

Downloaded from Arrow.com.

Table 3-3. Vector Table (20-, 24-pin products)

Vector Table Address	Interrupt Source
0000H	$\overline{\text{RESET}}$, POR, LVD, WDT, TRAP, IAW, RPE
0004H	INTWDTI
0006H	INTLVI
0008H	INTP0
000AH	INTP1
000CH	INTP2
000EH	INTP3
0010H	INTDMA0Note
0012H	INTDMA1Note
0014H	INTST0/INTCSI00/INTIIC00Note
0016H	INTSR0/INTCSI01Note/INTIIC01Note
0018H	INTSRE0
001AH	INTTM01H
001CH	INTTM03H
001EH	INTIICA0
0020H	INTTM00
0022H	INTTM01
0024H	INTTM02
0026H	INTTM03
0028H	INTAD
002AH	INTIT
002CH	INTKR
002EH	INTMD
0030H	INTFL
007EH	BRK

Note Provided only in the R5F102 products.

Downloaded from Arrow.com.

Table 3-4. Vector Table (30-pin products)

Vector Table Address	Interrupt Source
0000H	$\overline{\text{RESET}}$, POR, LVD, WDT, TRAP, IAW, RPE
0004H	INTWDTI
0006H	INTLVI
0008H	INTP0
000AH	INTP1
000CH	INTP2
000EH	INTP3
0010H	INTP4
0012H	INTP5
0014H	INTST2/INTCSI20Note/INTIIC20Note
0016H	INTSR2Note
0018H	INTSRE2Note
001AH	INTDMA0Note
001CH	INTDMA1Note
001EH	INTST0/INTCSI00/INTIIC00Note
0020H	INTSR0
0022H	INTSRE0/INTTM01H
0024H	INTST1Note
0026H	INTSR1Note/INTCSI11Note/INTIIC11Note
0028H	INTSRE1Note/INTTM03H
002AH	INTIICA0
002CH	INTTM00
002EH	INTTM01
0030H	INTTM02
0032H	INTTM03
0034H	INTAD
0038H	INTIT
0042H	INTTM04
0044H	INTTM05
0046H	INTTM06
0048H	INTTM07
005EH	INTMD
0062H	INTFL
007EH	BRK

Note Provided only in the R5F102 products.

Downloaded from Arrow.com.

(2) CALLT instruction table area

The 64-byte area of 00080H to 000BFH can store the subroutine entry address of a 2-byte call instruction (CALLT). Set the subroutine entry address to a value in a range of 00000H to 0FFFFH (because an address code is 2 bytes).

(3) Option byte area

The 4-byte area of 000C0H to 000C3H can be used as an option byte area. For details, see **CHAPTER 23 OPTION BYTE**.

(4) On-chip debug security ID setting area

The 10-byte areas of 000C4H to 000CDH and 010C4H to 010CDH can be used as an on-chip debug security ID setting area. For details, see **CHAPTER 25 ON-CHIP DEBUG FUNCTION**.

Downloaded from Arrow.com.

3.1.2 Mirror area

The RL78/G12 mirrors the code flash area of 00000H to 0FFFFH, to F0000H to FFFFFH (the code flash area to be mirrored is set by the processor mode control register (PMC)).

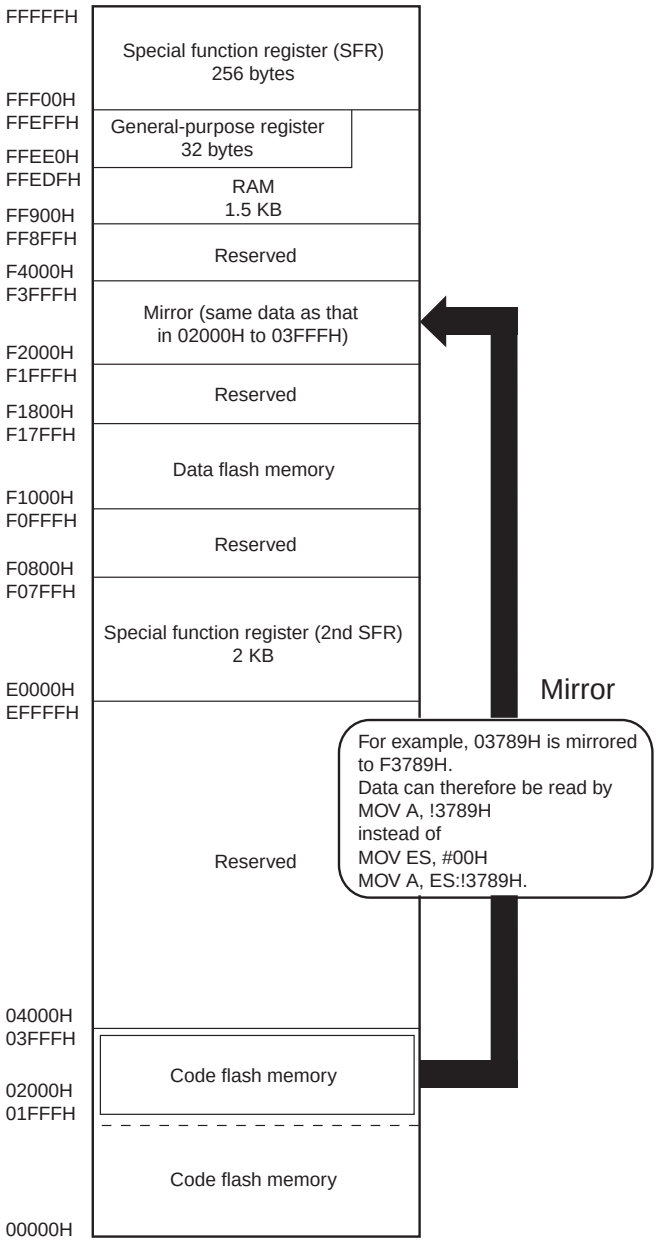
By reading data from F0000H to FFFFFH, an instruction that does not have the ES register as an operand can be used, and thus the contents of the code flash can be read with the shorter code. However, the code flash area is not mirrored to the SFR, extended SFR, RAM, and use prohibited areas.

See **3.1 Memory Space** for the mirror area of each product.

The mirror area can only be read and no instruction can be fetched from this area.

The following shows examples.

Example R5F1026A and RF5F1027A (Flash memory: 16 KB, RAM: 1.5 KB)



The PMC register is described below.

Downloaded from Arrow.com.

- **Processor mode control register (PMC)**

This register sets the flash memory space for mirroring to the area from F0000H to FFFFFH.

The PMC register can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation sets this register to 00H.

Figure 3-7. Format of Configuration of Processor Mode Control Register (PMC)

Address: FFFFEH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	<0>
PMC	0	0	0	0	0	0	0	MAA

MAA	Selection of flash memory space for mirroring to the area from F0000H to FFFFFH
0	00000H to 0FFFFH is mirrored to F0000H to FFFFFH
1	Setting prohibited

Cautions 1. Be sure to clear bit 0 (MAA) of this register to 0 (default value).
2. After setting the PMC register, wait for at least one instruction and access the mirror area.

3.1.3 Internal data memory space

The RL78/G12 products incorporate the following RAMs.

Table 3-5. Internal RAM Capacity

Part Number	Internal RAM
R5F10x66	256 × 8 bits (FFE00H to FFEFFH)
R5F10x67, R5F10x77, R5F10xA7	512 × 8 bits (FFD00H to FFEFFH)
R5F10x68, R5F10x78, R5F10xA8	768 × 8 bits (FFC00H to FFEFFH)
R5F10x69, R5F10x79, R5F10xA9	1024 × 8 bits (FFB00H to FFEFFH)
R5F10x6A, R5F10x7A	1536 × 8 bits (FF900H to FFEFFH)
R5F10xAA	2048 × 8 bits (FF700H to FFEFFH)

(x = 2 or 3)

The internal RAM can be used as a data area and a program area where instructions are fetched (it is prohibited to use the general-purpose register area for fetching instructions). Four general-purpose register, banks consisting of eight 8-bit registers registers per bank are assigned to the 32-byte area of FFEE0H to FFEFFH of the internal RAM area.

The internal RAM is used as a stack memory.

Downloaded from Arrow.com.

Cautions 1. It is prohibited to use the general-purpose register space (FFEE0H to FFEFFH) for fetching instructions or as a stack area.

<R> **2. Do not allocate RAM addresses which are used as a stack area, a data buffer, and a DMA transfer destination/transfer source to the RAM areas of the following products when performing self-programming and rewriting the data flash memory. For details, refer to RL78 Family Flash Self Programming Library Type01 User's Manual and RL78 Family Data Flash Library Type04 User's Manual.**

R5F10266 : FFE20H to FFEA1H, FFEE0H to FFEFFH
(Allocate the stack used for the data flash library to FFEA2H to FFEDFH and the RAM address used for the data buffer and DMA transfer to FFE00H to FFE1FH.)

R5F102mn, R5F103mn : FFE20H to FFEFFH
m: Pin count (m = 6, 7, A), n: ROM capacitance (n = 7, 8, 9, A)

3. Use of the RAM areas of the following products is prohibited when performing self-programming and rewriting the data flash memory, because these areas are used for each library. (Refer to Figures 3-3 to 3-5 Memory Map)

R5F102m8, R5F103m8: FFC00H to FFC89H
R5F102m9, R5F103m9: FFB00H to FFC89H
R5F102mA, R5F103mA: FF900H to FFC89H
m: Pin count (m = 6, 7)

Downloaded from Arrow.com.

3.1.4 Special function register (SFR) area

On-chip peripheral hardware special function registers (SFRs) are allocated in the area of FFF00H to FFFFFH (see **Table 3-6** in **3.2.4 Special function registers (SFRs)**).

Caution Do not access addresses to which SFRs are not assigned.

3.1.5 Extended special function register (2nd SFR: 2nd Special Function Register) area

On-chip peripheral hardware special function registers (2nd SFRs) are allocated in the area of F0000H to F07FFH (see **Table 3-7** in **3.2.5 Extended Special function registers (2nd SFRs: 2nd Special Function Registers)**).

SFRs other than those in the SFR area (FFF00H to FFFFFH) are allocated to this area. An instruction that accesses the extended SFR area, however, is 1 byte longer than an instruction that accesses the SFR area.

Caution Do not access addresses to which extended SFRs are not assigned.

Downloaded from Arrow.com.

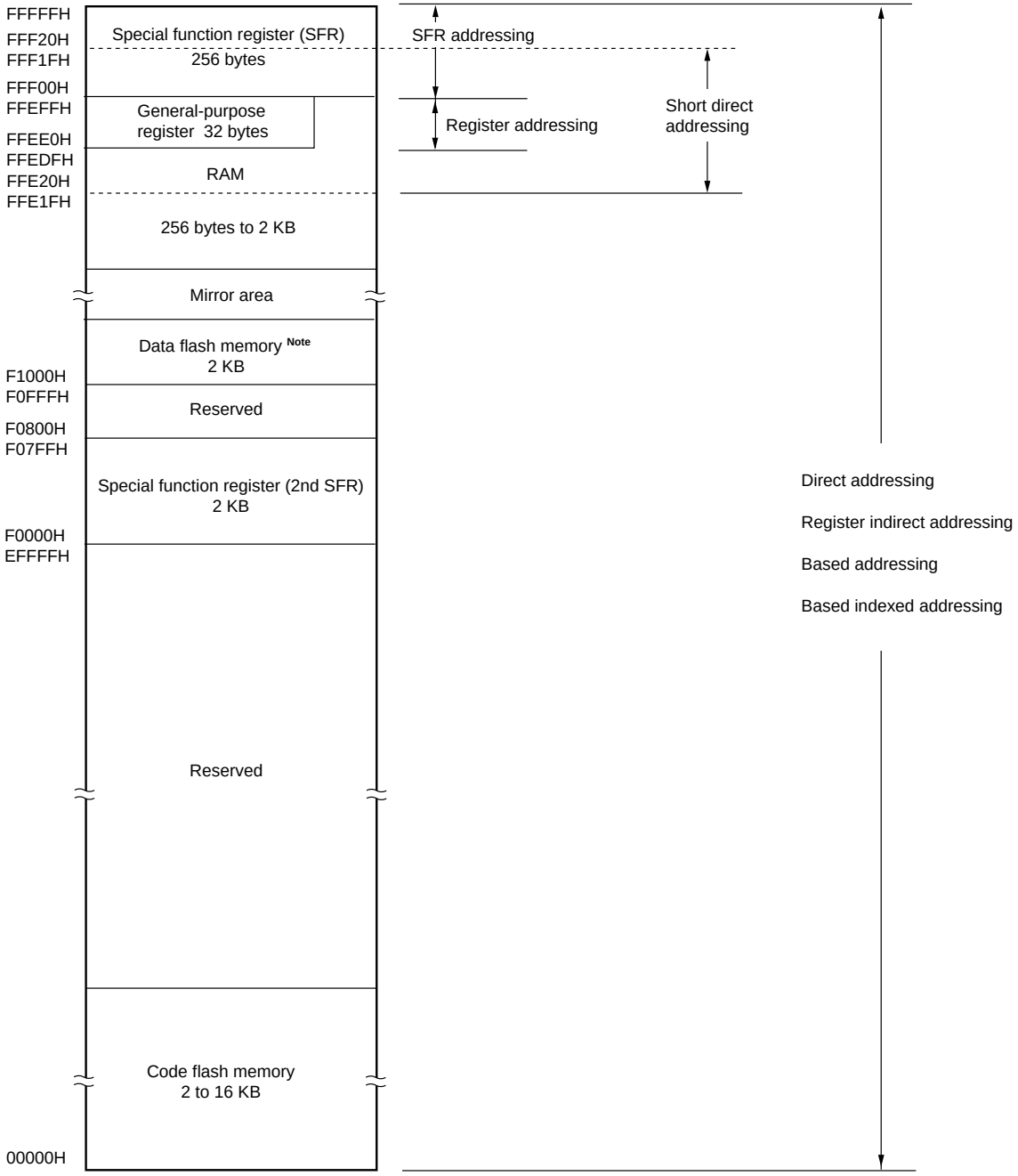
<R> 3.1.6 Data memory addressing

Addressing refers to the method of specifying the address of the instruction to be executed next or the address of the register or memory relevant to the execution of instructions.

Several addressing modes are provided for addressing the memory relevant to the execution of instructions for the RL78 microcontroller, based on operability and other considerations. In particular, special addressing methods designed for the functions of the special function registers (SFR) and general-purpose registers are available for use. Figure 3-8 shows correspondence between data memory and addressing.

For details of each addressing, see **3.4 Addressing for Processing Data Addresses**.

Figure 3-8. Correspondence Between Data Memory and Addressing



<R> **Note** The areas are reserved in the R5F103 products.

Downloaded from Arrow.com.

3.2 Processor Registers

The RL78/G12 products incorporate the following processor registers.

3.2.1 Control registers

The control registers control the program sequence, statuses and stack memory. The control registers consist of a program counter (PC), a program status word (PSW) and a stack pointer (SP).

(1) Program counter (PC)

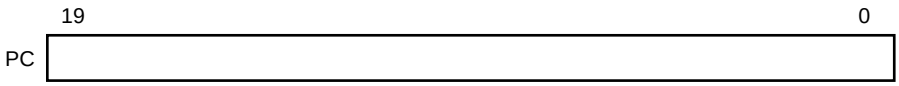
The program counter is a 20-bit register that holds the address information of the next program to be executed.

In normal operation, PC is automatically incremented according to the number of bytes of the instruction to be fetched.

When a branch instruction is executed, immediate data and register contents are set.

Reset signal generation sets the reset vector table values at addresses 0000H and 0001H to the 16 lower-order bits of the program counter. The four higher-order bits of the program counter are cleared to 0000.

Figure 3-9. Format of Program Counter



(2) Program status word (PSW)

The program status word is an 8-bit register consisting of various flags set/reset by instruction execution.

Program status word contents are stored in the stack area upon acknowledgment of a vectored interrupt request or PUSH PSW instruction execution, and are restored upon execution of the RETB, RETI and POP PSW instructions.

Reset signal generation sets the PSW register to 06H.

Figure 3-10. Format of Program Status Word

	7							0
PSW	IE	Z	RBS1	AC	RBS0	ISP1	ISP0	CY

(a) Interrupt enable flag (IE)

This flag controls the interrupt request acknowledge operations of the CPU.

When 0, the IE flag is set to the interrupt disabled (DI) state, and all maskable interrupt requests are disabled.

When 1, the IE flag is set to the interrupt enabled (EI) state, and interrupt request acknowledgment is controlled with an in-service priority flag (ISP1, ISP0), an interrupt mask flag for various interrupt sources, and a priority specification flag.

The IE flag is reset (0) upon DI instruction execution or interrupt acknowledgment and is set (1) upon EI instruction execution.

(b) Zero flag (Z)

When the operation or comparison result is zero or equal, this flag is set (1). It is reset (0) in all other cases.

(c) Register bank select flags (RBS0, RBS1)

These are 2-bit flags to select one of the four register banks.

In these flags, the 2-bit information that indicates the register bank selected by SEL RBn instruction execution is stored.

Downloaded from Arrow.com.

(d) Auxiliary carry flag (AC)

If the operation result has a carry from bit 3 or a borrow at bit 3, this flag is set (1). It is reset (0) in all other cases.

(e) In-service priority flags (ISP1, ISP0)

These flags manage the priority of acknowledgeable maskable vectored interrupts. Vectored interrupt requests specified lower than the value of ISP0 and ISP1 flags by the priority specification flag registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H) (see **15.3 (3)**) can not be acknowledged. Actual vectored interrupt request acknowledgment is controlled by the interrupt enable flag (IE).

Remark n = 0, 1

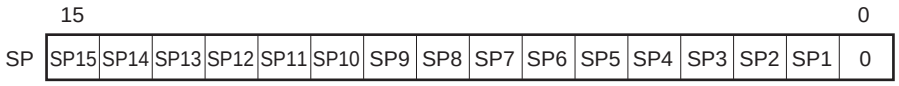
(f) Carry flag (CY)

This flag stores overflow and underflow upon add/subtract instruction execution. It stores the shift-out value upon rotate instruction execution and functions as a bit accumulator during bit operation instruction execution.

(3) Stack pointer (SP)

This is a 16-bit register to hold the start address of the memory stack area. Only the internal RAM area can be set as the stack area.

<R> **Figure 3-11. Format of Stack Pointer**



In stack addressing through a stack pointer, the SP is decremented ahead of write (save) to the stack memory and is incremented after read (restore) from the stack memory. Each stack operation saves data as shown in Figure 3-12.

Cautions 1. Since reset signal generation makes the SP contents undefined, be sure to initialize the SP before using the stack.

2. It is prohibited to use the general-purpose register space (FFEE0H to FFEFFH) for fetching instructions or as a stack area.

<R> **3. Do not allocate RAM addresses which are used as a stack area, a data buffer, and a DMA transfer destination/transfer source to the RAM areas of the following products when performing self-programming and rewriting the data flash memory. For details, refer to RL78 Family Flash Self Programming Library Type01 User's Manual and RL78 Family Data Flash Library Type04 User's Manual.**

R5F10266 : FFE20H to FFEA1H, FFEE0H to FFEFFH
(Allocate the stack used for the data flash library to FFEA2H to FFEDFH and the RAM address used for the data buffer and DMA transfer to FFE00H to FFE1FH.)

R5F102mn, R5F103mn : FFE20H to FFEFFH
m: Pin count (m = 6, 7, A), n: ROM capacitance (n = 7, 8, 9, A)

4. Use of the RAM areas of the following products is prohibited when performing self-programming and rewriting the data flash memory, because these areas are used for each library. (Refer to Figures 3-3 to 3-5 Memory Map)

R5F102m8, R5F103m8: FFC00H to FFC89H
R5F102m9, R5F103m9: FFB00H to FFC89H
R5F102mA, R5F103mA: FF900H to FFC89H
m: Pin count (m = 6, 7)

Downloaded from Arrow.com.

3.2.2 General-purpose registers

General-purpose registers are mapped at particular addresses (FFEE0H to FFEFFH) of the data memory. The general-purpose registers consists of 4 banks, each bank consisting of eight 8-bit registers (X, A, C, B, E, D, L, and H).

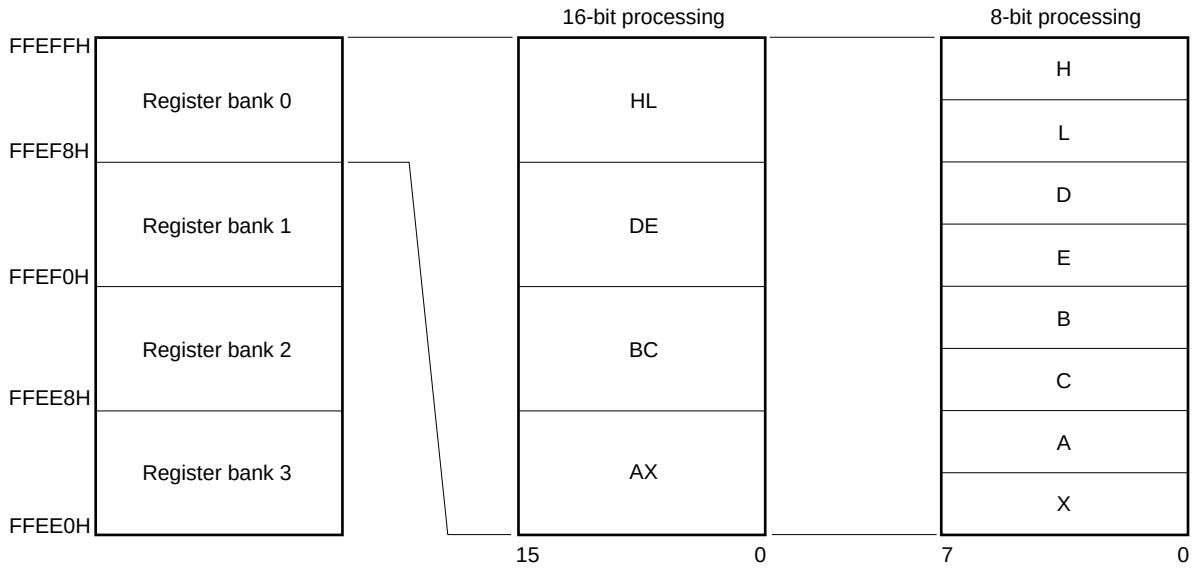
Each register can be used as an 8-bit register, and two 8-bit registers can also be used in a pair as a 16-bit register (AX, BC, DE, and HL).

Register banks to be used for instruction execution are set by the CPU control instruction (SEL RBn). Because of the 4-register bank configuration, an efficient program can be created by switching between a register for normal processing and a register for interrupts for each bank.

Caution It is prohibited to use the general-purpose register space (FFEE0H to FFEFFH) for fetching instructions or as a stack area.

Figure 3-12. Configuration of General-Purpose Registers

(a) Function name



Downloaded from Arrow.com.

3.2.3 ES and CS registers

The ES register is used for data access and the CS register is used to specify the higher address when a branch instruction is executed.

The default value of the ES register after reset is 0FH, and that of the CS register is 00H.

Figure 3-13. Configuration of ES and CS Registers

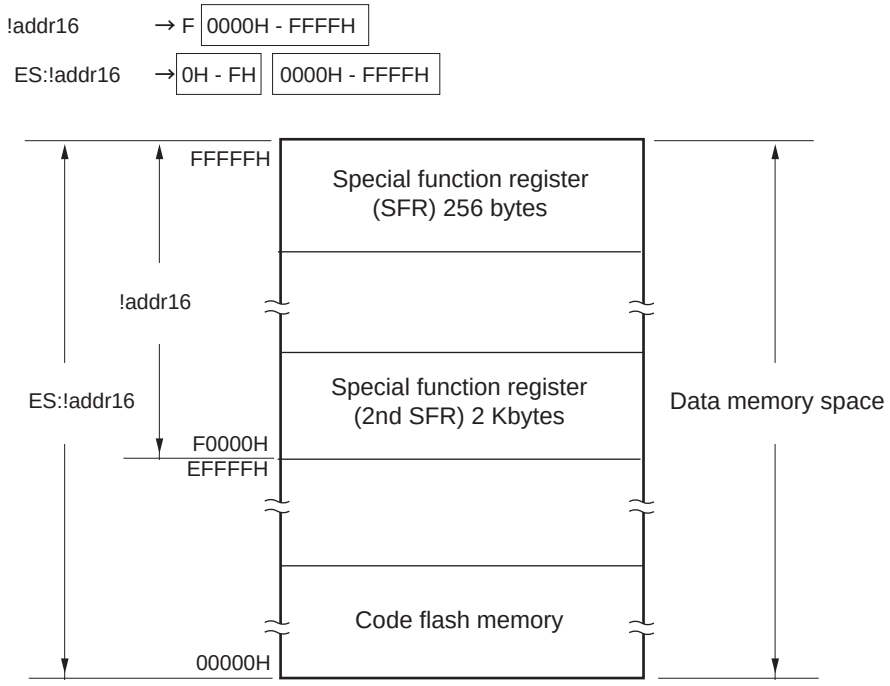
	7	6	5	4	3	2	1	0
ES	0	0	0	0	ES3	ES2	ES1	ES0

<R>

	7	6	5	4	3	2	1	0
CS	0	0	0	0	CS3	CS2	CS1	CS0

Though the data area which can be accessed with 16-bit addresses is the 64 Kbytes from F0000H to FFFFFH, using the ES register as well extends this to the 1 Mbyte from 00000H to FFFFFH.

Figure 3-14 Extension of Data Area Which Can Be Accessed



Downloaded from Arrow.com.

3.2.4 Special function registers (SFRs)

Unlike a general-purpose register, each SFR has a special function.

SFRs are allocated to the FFF00H to FFFFFH area.

SFRs can be manipulated like general-purpose registers, using operation, transfer, and bit manipulation instructions. The manipulable bit units, 1, 8, and 16, depend on the SFR type.

Each manipulation bit unit can be specified as follows.

- 1-bit manipulation
  Describe as follows for the 1-bit manipulation instruction operand (sfr.bit).
  When the bit name is defined: <Bit name>
  When the bit name is not defined: <Register name>, <Bit number> or <Address>, <Bit number>
- 8-bit manipulation
  Describe the symbol defined by the assembler for the 8-bit manipulation instruction operand (sfr). This manipulation can also be specified with an address
- 16-bit manipulation
  Describe the symbol defined by the assembler for the 16-bit manipulation instruction operand (sfrp). When specifying an address, describe an even address.

Table 3-6 gives a list of the SFRs. The meanings of items in the table are as follows.

- Symbol
  Symbol indicating the address of a special function register. It is a reserved word in the assembler, and is defined as an sfr variable using the #pragma sfr directive in the compiler. When using the assembler, debugger, and simulator, symbols can be written as an instruction operand.
- R/W
  Indicates whether the corresponding SFR can be read or written.
  R/W: Read/write enable
  R: Read only
  W: Write only
- Manipulable bit units
  "√" indicates the manipulable bit unit (1, 8, or 16). "–" indicates a bit unit for which manipulation is not possible.
- After reset
  Indicates each register status upon reset signal generation.

Caution Do not access addresses to which extended SFRs are not assigned.

Remark For extended SFRs (2nd SFRs), see **3.2.5 Extended special function registers (2nd SFRs: 2nd Special Function Registers)**.

Downloaded from Arrow.com.

Table 3-6. SFR List (1/4)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
FFF00H	Port register 0	P0		R/W	√	√	–	00H
FFF01H	Port register 1	P1		R/W	√	√	–	00H
FFF02H	Port register 2	P2		R/W	√	√	–	00H
FFF03H	Port register 3	P3		R/W	√	√	–	00H
FFF04H	Port register 4	P4		R/W	√	√	–	00H
FFF05H	Port register 5	P5		R/W	√	√	–	00H
FFF06H	Port register 6	P6		R/W	√	√	–	00H
FFF0CH	Port register 12	P12		R/W Note	√	√	–	Undefined
FFF0DH	Port register 13	P13		R	√	√	–	Undefined
FFF0EH	Port register 14	P14		R/W	√	√	–	00H
FFF10H	Serial data register 00	TXD0/ SIO00	SDR00	R/W	–	√	√	0000H
FFF11H		–			–	–		
FFF12H	Serial data register 01	RXD0/ SIO01	SDR01	R/W	–	√	√	0000H
FFF13H		–			–	–		
FFF18H	Timer data register 00	TDR00		R/W	–	–	√	0000H
FFF19H								
FFF1AH	Timer data register 01	TDR01L	TDR01	R/W	–	√	√	00H
FFF1BH		TDR01H			–	√		00H
FFF1EH	10-bit A/D conversion result register	ADCR		R	–	–	√	0000H
FFF1FH	8-bit A/D conversion result register	ADCRH		R	–	√	–	00H
FFF20H	Port mode register 0	PM0		R/W	√	√	–	FFH
FFF21H	Port mode register 1	PM1		R/W	√	√	–	FFH
FFF22H	Port mode register 2	PM2		R/W	√	√	–	FFH
FFF23H	Port mode register 3	PM3		R/W	√	√	–	FFH
FFF24H	Port mode register 4	PM4		R/W	√	√	–	FFH
FFF25H	Port mode register 5	PM5		R/W	√	√	–	FFH
FFF26H	Port mode register 6	PM6		R/W	√	√	–	FFH
FFF2CH	Port mode register 12	PM12		R/W	√	√	–	FFH
FFF2EH	Port mode register 14	PM14		R/W	√	√	–	FFH
FFF30H	A/D converter mode register 0	ADM0		R/W	√	√	–	00H
FFF31H	Analog input channel specification register	ADS		R/W	√	√	–	00H
FFF32H	A/D converter mode register 1	ADM1		R/W	√	√	–	00H
FFF34H	Key return control register	KRCTL		R/W	√	√	–	00H
FFF35H	Key return flag register	KRF		R/W	√	√	–	00H
FFF36H	Key return mode register 1	KRM1		R/W	√	√	–	00H
FFF37H	Key return mode register 0	KRM0		R/W	√	√	–	00H
FFF38H	External interrupt rising edge enable register 0	EGP0		R/W	√	√	–	00H
FFF39H	External interrupt falling edge enable register 0	EGN0		R/W	√	√	–	00H

Note Read only for 30-pin product.

Downloaded from Arrow.com.

<R> **Table 3-6. SFR List (2/4)**

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
FFF44H	Serial data register 02	TXD1/ SIO10	SDR02	R/W	–	√	√	0000H
FFF45H		–			–	–		
FFF46H	Serial data register 03	RXD1/ SIO11	SDR03	R/W	–	√	√	0000H
FFF47H		–			–	–		
FFF48H	Serial data register 10	TXD2/ SIO20	SDR10	R/W	–	√	√	0000H
FFF49H		–			–	–		
FFF4AH	Serial data register 11	RXD2/ SIO21	SDR11	R/W	–	√	√	0000H
FFF4BH		–			–	–		
FFF50H	IICA shift register 0	IICA0		R/W	–	√	–	00H
FFF51H	IICA status register 0	IICS0		R	√	√	–	00H
FFF52H	IICA flag register 0	IICF0		R/W	√	√	–	00H
FFF64H	Timer data register 02	TDR02		R/W	–	–	√	0000H
FFF65H								
FFF66H	Timer data register 03	TDR03L	TDR03	R/W	–	√	√	00H
FFF67H		TDR03H			–	√		00H
FFF68H	Timer data register 04	TDR04		R/W	–	–	√	0000H
FFF69H								
FFF6AH	Timer data register 05	TDR05		R/W	–	–	√	0000H
FFF6BH								
FFF6CH	Timer data register 06	TDR06		R/W	–	–	√	0000H
FFF6DH								
FFF6EH	Timer data register 07	TDR07		R/W	–	–	√	0000H
FFF6FH								
FFF90H	Interval timer control register	ITMC		R/W	–	–	√	0FFFH
FFF91H								
FFFA0H	Clock operation mode control register	CMC		R/W	–	√	–	00H
FFFA1H	Clock operation status control register	CSC		R/W	√	√	–	C0H
FFFA2H	Oscillation stabilization time counter status register	OSTC		R	√	√	–	00H
FFFA3H	Oscillation stabilization time select register	OSTS		R/W	–	√	–	07H
FFFA4H	System clock control register	CKC		R/W	√	√	–	00H
FFFA5H	Clock output select register 0	CKS0		R/W	√	√	–	00H
FFFA6H	Clock output select register 1	CKS1		R/W	√	√	–	00H
FFFA8H	Reset control flag register	RESF		R	–	√	–	Undefined Note 1
FFFA9H	Voltage detection register	LVIM		R/W	√	√	–	00H Note 1
FFFAAH	Voltage detection level register	LVIS		R/W	√	√	–	00H/01H/81H Note 1
FFFABH	Watchdog timer enable register	WDTE		R/W	–	√	–	1A/9A Note 2
FFFACH	CRC input register	CRCIN		R/W	–	√	–	00H

Downloaded from Arrow.com.

<R> **Notes 1.** The reset values of the registers vary depending on the reset source as shown below.

<R>

Register \ Reset Source		RESET Input	Reset by POR	Reset by Execution of Illegal Instruction	Reset by WDT	Reset by RAM parity error	Reset by illegal-memory access	Reset by LVD
RESF	TRAP bit	Cleared (0)		Set (1)	Held			Held
	WDTRF bit			Held	Set (1)	Held		
	RPERF bit			Held		Set (1)	Held	
	IAWRF bit			Held			Set (1)	
	LVIRF bit			Held				Set (1)
LVIM	LVISEN bit	Cleared (0)						Held
	LVIOMSK bit	Held						
	LVIF bit							
LVIS		Cleared (00H/01H/81H)						

2. The reset value of the WDTE register is determined by the setting of the option byte.

Table 3-6. SFR List (3/4)

<R>

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
FFFB0H	DMA SFR address register 0	DSA0		R/W	–	√	–	00H
FFFB1H	DMA SFR address register 1	DSA1		R/W	–	√	–	00H
FFFB2H	DMA RAM address register 0	DRA0L	DRA0	R/W	–	√	√	00H
FFFB3H		DRA0H		R/W	–	√		00H
FFFB4H	DMA RAM address register 1	DRA1L	DRA1	R/W	–	√	√	00H
FFFB5H		DRA1H		R/W	–	√		00H
FFFB6H	DMA byte count register 0	DBC0L	DBC0	R/W	–	√	√	00H
FFFB7H		DBC0H		R/W	–	√		00H
FFFB8H	DMA byte count register 1	DBC1L	DBC1	R/W	–	√	√	00H
FFFB9H		DBC1H		R/W	–	√		00H
FFFBAH	DMA mode control register 0	DMC0		R/W	√	√	–	00H
FFFBBH	DMA mode control register 1	DMC1		R/W	√	√	–	00H
FFFBCH	DMA operation control register 0	DRC0		R/W	√	√	–	00H
FFFBDH	DMA operation control register 1	DRC1		R/W	√	√	–	00H
FFFD0H	Interrupt request flag register 2	IF2L	IF2	R/W	√	√	√	00H
FFFD1H		IF2H		R/W	√	√		00H
FFFD4H	Interrupt mask flag register 2	MK2L	MK2	R/W	√	√	√	FFH
FFFD5H		MK2H		R/W	√	√		FFH
FFFD8H	Priority specification flag register 02	PR02L	PR02	R/W	√	√	√	FFH
FFFD9H		PR02H		R/W	√	√		FFH
FFFDCH	Priority specification flag register 12	PR12L	PR12	R/W	√	√	√	FFH
FFFDDH		PR12H		R/W	√	√		FFH

Downloaded from Arrow.com.

Table 3-6. SFR List (4/4)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
FFFE0H	Interrupt request flag register 0	IF0L	IF0	R/W	√	√	√	00H
FFFE1H		IF0H		R/W	√	√		00H
FFFE2H	Interrupt request flag register 1	IF1L	IF1	R/W	√	√	√	00H
FFFE3H		IF1H		R/W	√	√		00H
FFFE4H	Interrupt mask flag register 0	MK0L	MK0	R/W	√	√	√	FFH
FFFE5H		MK0H		R/W	√	√		FFH
FFFE6H	Interrupt mask flag register 1	MK1L	MK1	R/W	√	√	√	FFH
FFFE7H		MK1H		R/W	√	√		FFH
FFFE8H	Priority specification flag register 00	PR00L	PR00	R/W	√	√	√	FFH
FFFE9H		PR00H		R/W	√	√		FFH
FFFEAH	Priority specification flag register 01	PR01L	PR01	R/W	√	√	√	FFH
FFFEBH		PR01H		R/W	√	√		FFH
FFFECH	Priority specification flag register 10	PR10L	PR10	R/W	√	√	√	FFH
FFFEDH		PR10H		R/W	√	√		FFH
FFFEEH	Priority specification flag register 11	PR11L	PR11	R/W	√	√	√	FFH
FFFEFH		PR11H		R/W	√	√		FFH
FFFF0H	Multiplication/division data register A (L)	MDAL		R/W	–	–	√	0000H
FFFF1H								
FFFF2H	Multiplication/division data register A (H)	MDAH		R/W	–	–	√	0000H
FFFF3H								
FFFF4H	Multiplication/division data register B (H)	MDBH		R/W	–	–	√	0000H
FFFF5H								
FFFF6H	Multiplication/division data register B (L)	MDBL		R/W	–	–	√	0000H
FFFF7H								
FFFFEH	Processor mode control register	PMC		R/W	√	√	–	00H

Remark For extended SFRs (2nd SFRs), see **Table 3-7 Extended SFR (2nd SFR) List**.

Downloaded from Arrow.com.

3.2.5 Extended special function registers (2nd SFRs: 2nd Special Function Registers)

Unlike a general-purpose register, each extended SFR (2nd SFR) has a special function.

Extended SFRs are allocated to the F0000H to F07FFH area. SFRs other than those in the SFR area (FFF00H to FFFFFH) are allocated to this area. An instruction that accesses the extended SFR area, however, is 1 byte longer than an instruction that accesses the SFR area.

Extended SFRs can be manipulated like general-purpose registers, using operation, transfer, and bit manipulation instructions. The manipulable bit units, 1, 8, and 16, depend on the SFR type.

Each manipulation bit unit can be specified as follows.

- 1-bit manipulation
  Describe the symbol reserved by the assembler for the 1-bit manipulation instruction operand (!addr16.bit). This manipulation can also be specified with an address.
- 8-bit manipulation
  Describe the symbol reserved by the assembler for the 8-bit manipulation instruction operand (!addr16). This manipulation can also be specified with an address.
- 16-bit manipulation
  Describe the symbol reserved by the assembler for the 16-bit manipulation instruction operand (!addr16). When specifying an address, describe an even address.

Table 3-7 gives a list of the extended SFRs. The meanings of items in the table are as follows.

- Symbol
  Symbol indicating the address of an extended SFR. It is a reserved word in the assembler, and is defined as an sfr variable using the #pragma sfr directive in the compiler. When using the assembler, debugger, and simulator, symbols can be written as an instruction operand.
- R/W
  Indicates whether the corresponding extended SFR can be read or written.
  R/W: Read/write enable
  R: Read only
  W: Write only
- Manipulable bit units
  "√" indicates the manipulable bit unit (1, 8, or 16). "–" indicates a bit unit for which manipulation is not possible.
- After reset
  Indicates each register status upon reset signal generation.

Caution Do not access addresses to which extended SFRs are not assigned.

Remark For SFRs in the SFR area, see **3.2.4 Special function registers (SFRs)**.

Downloaded from Arrow.com.

<R> **Table 3-7. Extended SFR (2nd SFR) List (1/5)**

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F0010H	A/D converter mode register 2	ADM2	R/W	√	√	–	00H
F0011H	Conversion result comparison upper limit setting register	ADUL	R/W	–	√	–	FFH
F0012H	Conversion result comparison lower limit setting register	ADLL	R/W	–	√	–	00H
F0013H	A/D test register	ADTES	R/W	–	√	–	00H
F0030H	Pull-up resistor option register 0	PU0	R/W	√	√	–	00H
F0031H	Pull-up resistor option register 1	PU1	R/W	√	√	–	00H
F0033H	Pull-up resistor option register 3	PU3	R/W	√	√	–	00H
F0034H	Pull-up resistor option register 4	PU4	R/W	√	√	–	01H
F0035H	Pull-up resistor option register 5	PU5	R/W	√	√	–	00H
F0037H	Pull-up resistor option register 7	PU7	R/W	√	√	–	00H
F003CH	Pull-up resistor option register 12	PU12	R/W	√	√	–	00H/20H Note 3
F003EH	Pull-up resistor option register 14	PU14	R/W	√	√	–	00H
F0040H	Port input mode register 0	PIM0	R/W	√	√	–	00H
F0041H	Port input mode register 1	PIM1	R/W	√	√	–	00H
F0050H	Port output mode register 0	POM0	R/W	√	√	–	00H
F0051H	Port output mode register 1	POM1	R/W	√	√	–	00H
F0054H	Port output mode register 4	POM4	R/W	√	√	–	00H
F0055H	Port output mode register 5	POM5	R/W	√	√	–	00H
F0060H	Port mode control register 0	PMC0	R/W	√	√	–	FFH
F0061H	Port mode control register 1	PMC1	R/W	√	√	–	FFH
F0064H	Port mode control register 4	PMC4	R/W	√	√	–	FFH
F006CH	Port mode control register 12	PMC12	R/W	√	√	–	FFH
F006EH	Port mode control register 14	PMC14	R/W	√	√	–	FFH
F0070H	Noise filter enable register 0	NFEN0	R/W	√	√	–	00H
F0071H	Noise filter enable register 1	NFEN1	R/W	√	√	–	00H
F0074H	Timer input select register 0	TIS0	R/W	–	√	–	00H
F0076H	A/D port configuration register	ADPC	R/W	–	√	–	00H
F0077H	Peripheral I/O redirection register	PIOR	R/W	–	√	–	00H
F0078H	Invalid memory access detection control register	IAWCTL	R/W	–	√	–	00H
F0090H	Data flash control register	DFLCTL	R/W	√	√	–	00H
F00A0H	High-speed on-chip oscillator trimming register	HOTRM	R/W	–	√	–	Undefined Note 1
F00A8H	High-speed on-chip oscillator frequency selecting register	HOCODIV	R/W	–	√	–	Undefined Note 2
F00E0H	Multiplication/division data register C (L)	MDCL	R/W	–	–	√	0000H
F00E2H	Multiplication/division data register C (H)	MDCH	R/W	–	–	√	0000H
F00E8H	Multiplication/division control register	MDUC	R/W	√	√	–	00H

Notes 1. The value after a reset is adjusted at the time of shipment.

<R> **2.** The value after a reset is a value set in FRQSEL2 to FRQSEL0 of the option byte (000C2H).

<R> **3.** The value after a reset is 20H in 20- and 24-pin products.

Downloaded from Arrow.com.

Table 3-7. Extended SFR (2nd SFR) List (2/5)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F00F0H	Peripheral enable register 0	PER0		R/W	√	√	–	00H
F00F3H	Operation speed mode control register	OSMC		R/W	–	√	–	00H
F00F5H	RAM parity error control register	RPECTL		R/W	√	√	–	00H
F00FEH	BCD adjust result register	BCDADJ		R	–	√	–	Undefined
F0100H	Serial status register 00	SSR00L	SSR00	R	–	√	√	0000H
F0101H		–			–	–		
F0102H	Serial status register 01	SSR01L	SSR01	R	–	√	√	0000H
F0103H		–			–	–		
F0104H	Serial status register 02	SSR02L	SSR02	R	–	√	√	0000H
F0105H		–			–	–		0000H
F0106H	Serial status register 03	SSR03L	SSR03	R	–	√	√	0000H
F0107H		–			–	–		0000H
F0108H	Serial flag clear trigger register 00	SIR00L	SIR00	R/W	–	√	√	0000H
F0109H		–			–	–		
F010AH	Serial flag clear trigger register 01	SIR01L	SIR01	R/W	–	√	√	0000H
F010BH		–			–	–		
F010CH	Serial flag clear trigger register 02	SIR02L	SIR02	R/W	–	√	√	0000H
F010DH		–			–	–		
F010EH	Serial flag clear trigger register 03	SIR03L	SIR03	R/W	–	√	√	0000H
F010FH		–			–	–		
F0110H	Serial mode register 00	SMR00		R/W	–	–	√	0020H
F0111H								
F0112H	Serial mode register 01	SMR01		R/W	–	–	√	0020H
F0113H								
F0114H	Serial mode register 02	SMR02		R/W	–	–	√	0020H
F0115H								
F0116H	Serial mode register 03	SMR03		R/W	–	–	√	0020H
F0117H								
F0118H	Serial communication operation setting register 00	SCR00		R/W	–	–	√	0087H
F0119H								
F011AH	Serial communication operation setting register 01	SCR01		R/W	–	–	√	0087H
F011BH								
F011CH	Serial communication operation setting register 02	SCR02		R/W	–	–	√	0087H
F011DH								
F011EH	Serial communication operation setting register 03	SCR03		R/W	–	–	√	0087H
F011FH								

Downloaded from Arrow.com.

Table 3-7. Extended SFR (2nd SFR) List (3/5)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F0120H	Serial channel enable status register 0	SE0L	SE0	R	√	√	√	0000H
F0121H		–			–	–		
F0122H	Serial channel start register 0	SS0L	SS0	R/W	√	√	√	0000H
F0123H		–			–	–		
F0124H	Serial channel stop register 0	ST0L	ST0	R/W	√	√	√	0000H
F0125H		–			–	–		
F0126H	Serial clock select register 0	SPS0L	SPS0	R/W	–	√	√	0000H
F0127H		–			–	–		
F0128H	Serial output register 0	SO0		R/W	–	–	√	0F0FH
F0129H								
F012AH	Serial output enable register 0	SOE0L	SOE0	R/W	√	√	√	0000H
F012BH		–			–	–		
F0134H	Serial output level register 0	SOL0L	SOL0	R/W	–	√	√	0000H
F0135H		–			–	–		
F0138H	Serial standby control register 0	SSC0L	SSC0	R/W	–	√	√	0000H
		–			–	–		
F0140H	Serial status register 10	SSR10L	SSR10	R	–	√	√	0000H
F0141H		–			–	–		
F0142H	Serial status register 11	SSR11L	SSR11	R	–	√	√	0000H
F0143H		–			–	–		
F0148H	Serial flag clear trigger register 10	SIR10L	SIR10	R/W	–	√	√	0000H
F0149H		–			–	–		
F014AH	Serial flag clear trigger register 11	SIR11L	SIR11	R/W	–	√	√	0000H
F014BH		–			–	–		
F0150H	Serial mode register 10	SMR10		R/W	–	–	√	0020H
F0151H								
F0152H	Serial mode register 11	SMR11		R/W	–	–	√	0020H
F0153H								
F0158H	Serial communication operation setting register 10	SCR10		R/W	–	–	√	0087H
F0159H								
F015AH	Serial communication operation setting register 11	SCR11		R/W	–	–	√	0087H
F015BH								
F0160H	Serial channel enable status register 1	SE1L	SE1	R	√	√	√	0000H
F0161H		–			–	–		
F0162H	Serial channel start register 1	SS1L	SS1	R/W	√	√	√	0000H
F0163H		–			–	–		
F0164H	Serial channel stop register 1	ST1L	ST1	R/W	√	√	√	0000H
F0165H		–			–	–		
F0166H	Serial clock select register 1	SPS1L	SPS1	R/W	–	√	√	0000H
F0167H		–			–	–		
F0168H	Serial output register 1	SO1		R/W	–	–	√	0F0FH
F0169H								
F016AH	Serial output enable register 1	SOE1L	SOE1	R/W	√	√	√	0000H
F016BH		–			–	–		
F0174H	Serial output level register 1	SOL1L	SOL1	R/W	–	√	√	0000H
F0175H		–			–	–		

Downloaded from Arrow.com.

Table 3-7. Extended SFR (2nd SFR) List (4/5)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F0180H	Timer counter register 00	TCR00		R	–	–	√	FFFFH
F0181H								
F0182H	Timer counter register 01	TCR01		R	–	–	√	FFFFH
F0183H								
F0184H	Timer counter register 02	TCR02		R	–	–	√	FFFFH
F0185H								
F0186H	Timer counter register 03	TCR03		R	–	–	√	FFFFH
F0187H								
F0188H	Timer counter register 04	TCR04		R	–	–	√	FFFFH
F0189H								
F018AH	Timer counter register 05	TCR05		R	–	–	√	FFFFH
F018BH								
F018CH	Timer counter register 06	TCR06		R	–	–	√	FFFFH
F018DH								
F018EH	Timer counter register 07	TCR07		R	–	–	√	FFFFH
F018FH								
F0190H	Timer mode register 00	TMR00		R/W	–	–	√	0000H
F0191H								
F0192H	Timer mode register 01	TMR01		R/W	–	–	√	0000H
F0193H								
F0194H	Timer mode register 02	TMR02		R/W	–	–	√	0000H
F0195H								
F0196H	Timer mode register 03	TMR03		R/W	–	–	√	0000H
F0197H								
F0198H	Timer mode register 04	TMR04		R/W	–	–	√	0000H
F0199H								
F019AH	Timer mode register 05	TMR05		R/W	–	–	√	0000H
F019BH								
F019CH	Timer mode register 06	TMR06		R/W	–	–	√	0000H
F019DH								
F019EH	Timer mode register 07	TMR07		R/W	–	–	√	0000H
F019FH								
F01A0H	Timer status register 00	TSR00L	TSR00	R	–	√	√	0000H
F01A1H		–			–	–		
F01A2H	Timer status register 01	TSR01L	TSR01	R	–	√	√	0000H
F01A3H		–			–	–		
F01A4H	Timer status register 02	TSR02L	TSR02	R	–	√	√	0000H
F01A5H		–			–	–		
F01A6H	Timer status register 03	TSR03L	TSR03	R	–	√	√	0000H
F01A7H		–			–	–		
F01A8H	Timer status register 04	TSR04L	TSR04	R	–	√	√	0000H
F01A9H		–			–	–		
F01AAH	Timer status register 05	TSR05L	TSR05	R	–	√	√	0000H
F01ABH		–			–	–		
F01ACH	Timer status register 06	TSR06L	TSR06	R	–	√	√	0000H
F01ADH		–			–	–		
F01AEH	Timer status register 07	TSR07L	TSR07	R	–	√	√	0000H
F01AFH		–			–	–		

Downloaded from Arrow.com.

Table 3-7. Extended SFR (2nd SFR) List (5/5)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F01B0H	Timer channel enable status register 0	TE0L	TE0	R	√	√	√	0000H
F01B1H		–			–	–		
F01B2H	Timer channel start register 0	TS0L	TS0	R/W	√	√	√	0000H
F01B3H		–			–	–		
F01B4H	Timer channel stop register 0	TT0L	TT0	R/W	√	√	√	0000H
F01B5H		–			–	–		
F01B6H	Timer clock select register 0	TPS0		R/W	–	–	√	0000H
F01B7H								
F01B8H	Timer output register 0	TO0L	TO0	R/W	–	√	√	0000H
F01B9H		–			–	–		
F01BAH	Timer output enable register 0	TOE0L	TOE0	R/W	√	√	√	0000H
F01BBH		–			–	–		
F01BCH	Timer output level register 0	TOL0L	TOL0	R/W	–	√	√	0000H
F01BDH		–			–	–		
F01BEH	Timer output mode register 0	TOM0L	TOM0	R/W	–	√	√	0000H
F01BFH		–			–	–		
F0230H	IICA control register 00	IICCTL00		R/W	√	√	–	00H
F0231H	IICA control register 01	IICCTL01		R/W	√	√	–	00H
F0232H	IICA low-level width setting register 0	IICWL0		R/W	–	√	–	FFH
F0233H	IICA high-level width setting register 0	IICWH0		R/W	–	√	–	FFH
F0234H	Slave address register 0	SVA0		R/W	–	√	–	00H
F02FAH	CRC data register	CRCD		R/W	–	–	√	0000H

Remark For SFRs in the SFR area, see **Table 3-6 SFR List**.

Downloaded from Arrow.com.

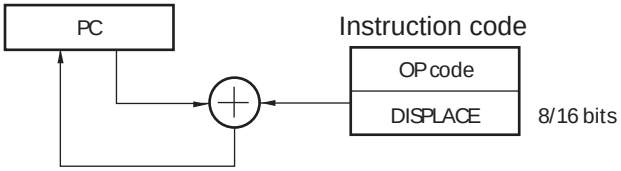
3.3 Instruction Address Addressing

3.3.1 Relative addressing

[Function]

Relative addressing stores in the program counter (PC) the result of adding a displacement value included in the instruction word (signed complement data: –128 to +127 or –32768 to +32767) to the program counter (PC)'s value (the start address of the next instruction), and specifies the program address to be used as the branch destination. Relative addressing is applied only to branch instructions.

Figure 3-15. Outline of Relative Addressing



3.3.2 Immediate addressing

[Function]

Immediate addressing stores immediate data of the instruction word in the program counter, and specifies the program address to be used as the branch destination.

For immediate addressing, CALL !!addr20 or BR !!addr20 is used to specify 20-bit addresses and CALL !addr16 or BR !addr16 is used to specify 16-bit addresses. 0000 is set to the higher 4 bits when specifying 16-bit addresses.

Figure 3-16. Example of CALL !!addr20/BR !!addr20

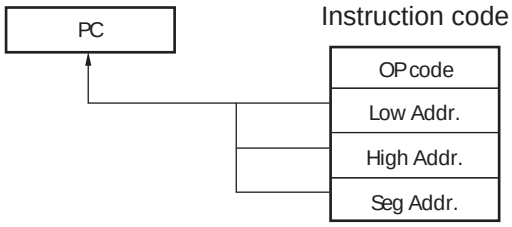
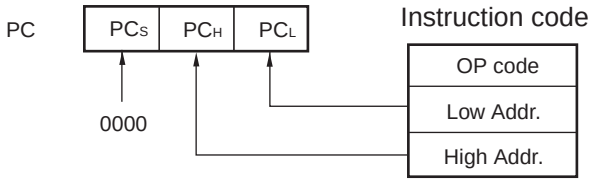


Figure 3-17. Example of CALL !addr16/BR !addr16



Downloaded from Arrow.com.

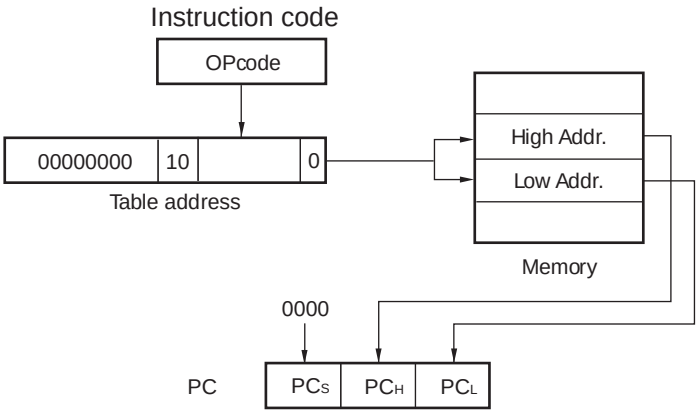
3.3.3 Table indirect addressing

[Function]

Table indirect addressing specifies a table address in the CALLT table area (0080H to 00BFH) with the 5-bit immediate data in the instruction word, stores the contents at that table address and the next address in the program counter (PC) as 16-bit data, and specifies the program address. Table indirect addressing is applied only for CALLT instructions.

In the RL78 microcontrollers, branching is enabled only to the 64 KB space from 00000H to 0FFFFH.

Figure 3-18. Outline of Table Indirect Addressing

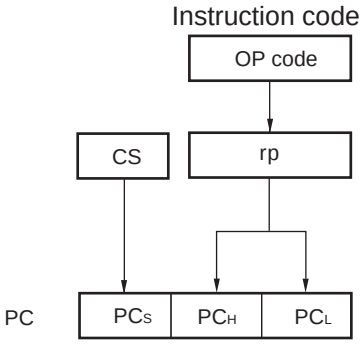


3.3.4 Register direct addressing

[Function]

Register direct addressing stores in the program counter (PC) the contents of a general-purpose register pair (AX/BC/DE/HL) and CS register of the current register bank specified with the instruction word as 20-bit data, and specifies the program address. Register direct addressing can be applied only to the CALL AX, BC, DE, HL, and BR AX instructions.

Figure 3-19. Outline of Register Direct Addressing



Downloaded from Arrow.com.

3.4 Addressing for Processing Data Addresses

3.4.1 Implied addressing

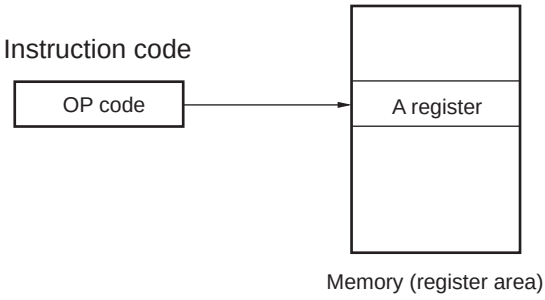
[Function]

Instructions for accessing registers (such as accumulators) that have special functions are directly specified with the instruction word, without using any register specification field in the instruction word.

[Operand format]

Implied addressing can be applied only to MULU X.

Figure 3-20. Outline of Implied Addressing



3.4.2 Register addressing

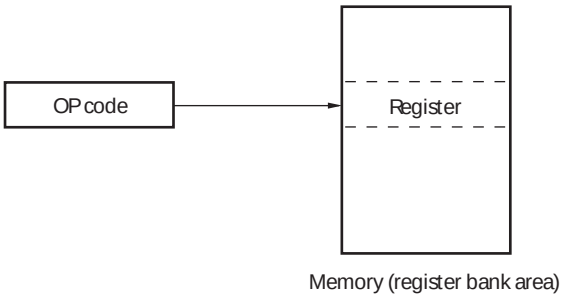
[Function]

Register addressing accesses a general-purpose register as an operand. The instruction word of 3-bit long is used to select an 8-bit register and the instruction word of 2-bit long is used to select a 16-bit register.

[Operand format]

Identifier	Description
r	X, A, C, B, E, D, L, H
rp	AX, BC, DE, HL

Figure 3-21. Outline of Register Addressing



Downloaded from Arrow.com.

3.4.3 Direct addressing

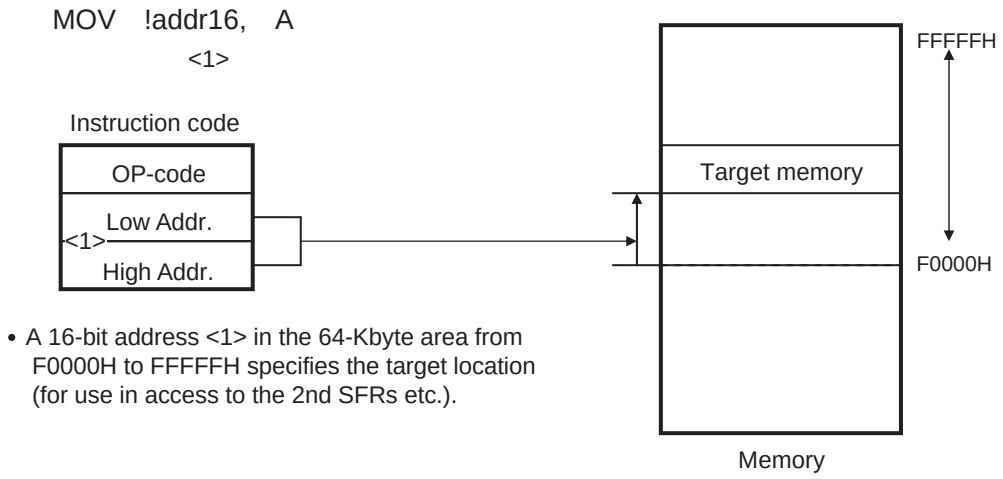
[Function]

Direct addressing uses immediate data in the instruction word as an operand address to directly specify the target address.

[Operand format]

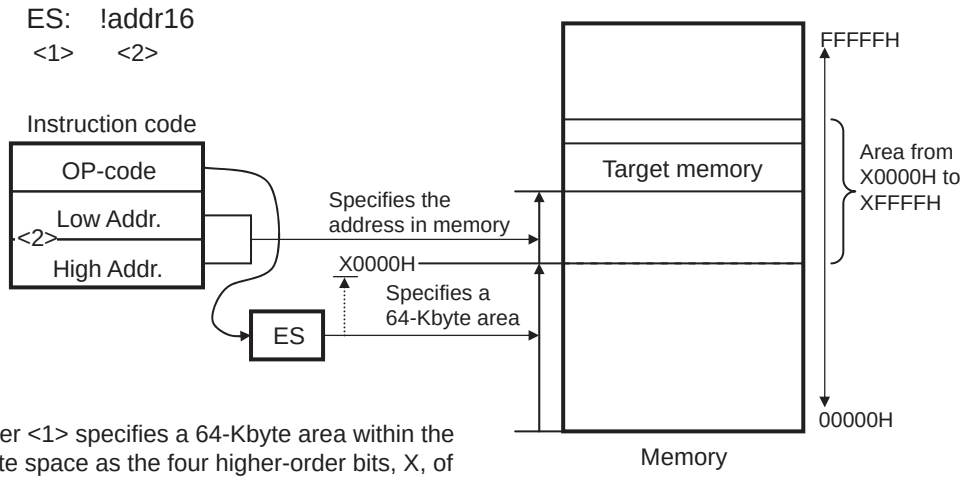
Identifier	Description
ADDR16	Label or 16-bit immediate data (only the space from F0000H to FFFFFH is specifiable: automatically added F of higher 4-bit addresses)
ES: ADDR16	Label or 16-bit immediate data (higher 4-bit addresses are specified by the ES register)

Figure 3-22. Example of ADDR16



- A 16-bit address <1> in the 64-Kbyte area from F0000H to FFFFFH specifies the target location (for use in access to the 2nd SFRs etc.).

Figure 3-23. Example of ES:ADDR16



- The ES register <1> specifies a 64-Kbyte area within the overall 1-Mbyte space as the four higher-order bits, X, of the address range.
- A 16-bit address <2> in the area from X0000H to XFFFFH and the ES register <1> specify the target location; this is used for access to fixed data other than that in mirrored areas.

Downloaded from Arrow.com.

3.4.4 Short direct addressing

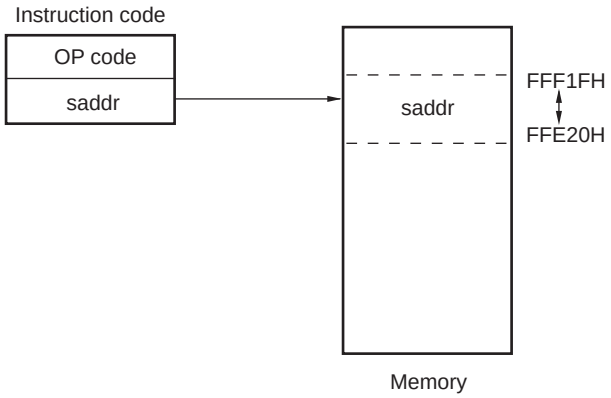
[Function]

Short direct addressing directly specifies the target addresses using 8-bit data in the instruction word. This type of addressing is applied only to the space from FFE20H to FFF1FH.

[Operand format]

Identifier	Description
SADDR	Label or FFE20H to FFF1FH immediate data
SADDRP	Label or FFE20H to FFF1FH immediate data (only even address is specifiable.)

Figure 3-24. Outline of Short Direct Addressing



Remark SADDR and SADDRP are used to describe the values of addresses FE20H to FF1FH with 16-bit immediate data (higher 4 bits of actual address are omitted), and the values of addresses FFE20H to FFF1FH with 20-bit immediate data.

Regardless of whether SADDR or SADDRP is used, addresses within the space from FFE20H to FFF1FH are specified for the memory.

Downloaded from Arrow.com.

3.4.5 SFR addressing

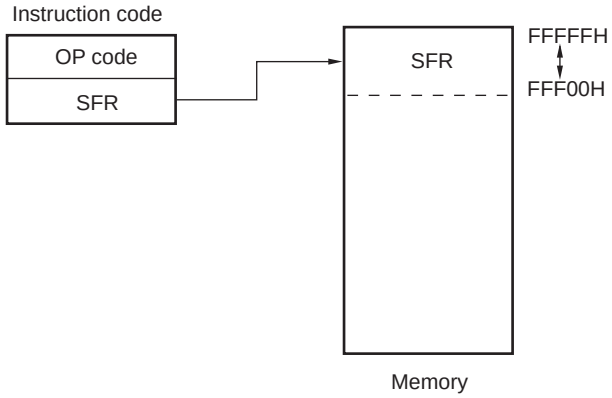
[Function]

SFR addressing directly specifies the target SFR addresses using 8-bit data in the instruction word. This type of addressing is applied only to the space from FFF00H to FFFFFH.

[Operand format]

Identifier	Description
SFR	SFR name
SFRP	16-bit-manipulatable SFR name (even address)

Figure 3-25. Outline of SFR Addressing



Downloaded from Arrow.com.

3.4.6 Register indirect addressing

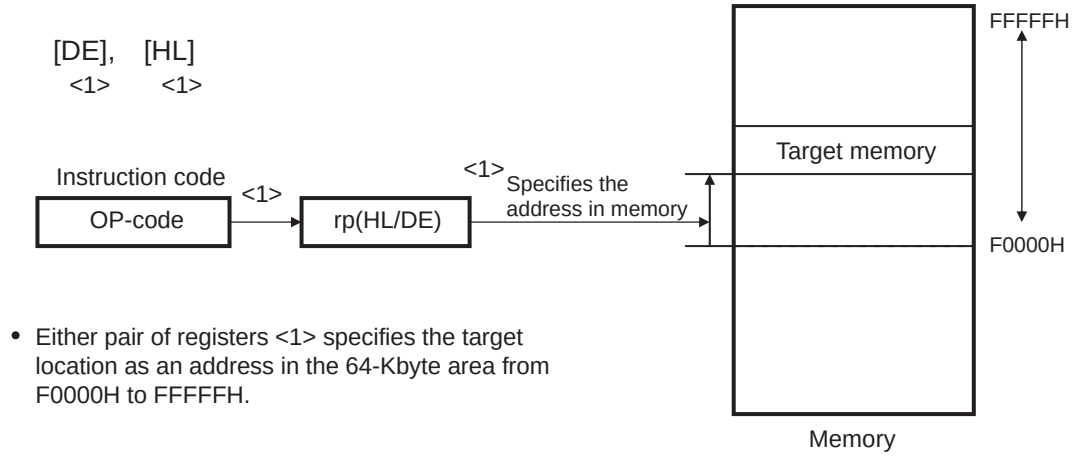
[Function]

Register indirect addressing directly specifies the target addresses using the contents of the register pair specified with the instruction word as an operand address.

[Operand format]

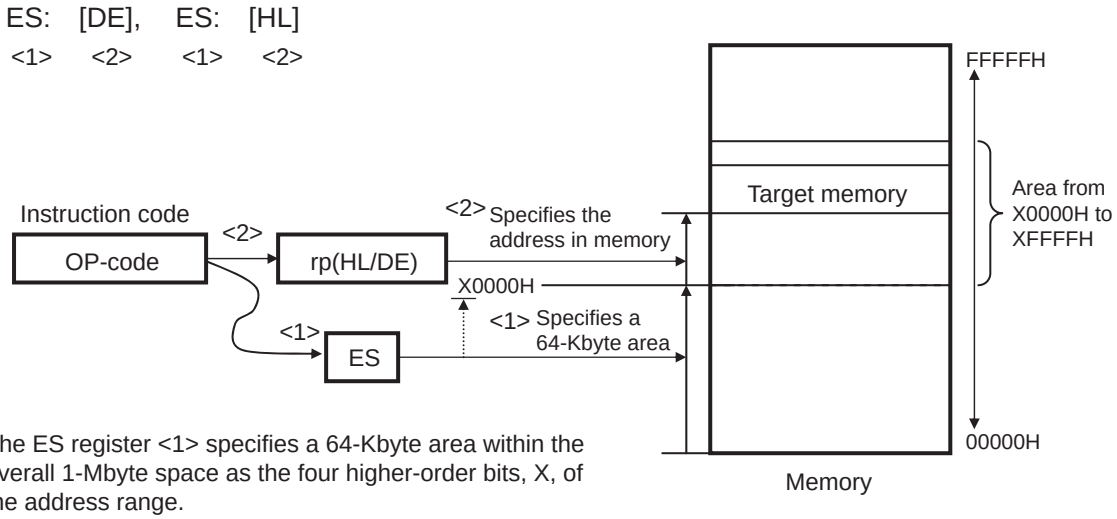
Identifier	Description
–	[DE], [HL] (only the space from F0000H to FFFFFH is specifiable)
–	ES:[DE], ES:[HL] (higher 4-bit addresses are specified by the ES register)

Figure 3-26. Example of [DE], [HL]



- Either pair of registers <1> specifies the target location as an address in the 64-Kbyte area from F0000H to FFFFFH.

Figure 3-27. Example of ES:[DE], ES:[HL]



- The ES register <1> specifies a 64-Kbyte area within the overall 1-Mbyte space as the four higher-order bits, X, of the address range.
- Either pair of registers <2> and the ES register <1> specify the target location in the area from X0000H to XFFFFH.

Downloaded from Arrow.com.

3.4.7 Based addressing

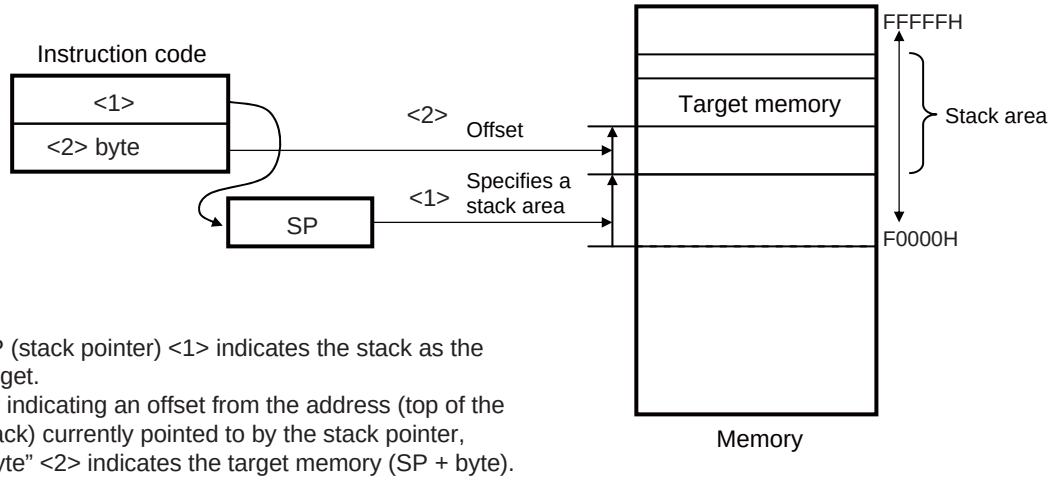
[Function]

Based addressing uses the contents of a register pair specified with the instruction word as a base address, and 8-bit immediate data or 16-bit immediate data as offset data. The sum of these values is used to specify the target address.

[Operand format]

Identifier	Description
–	[HL + byte], [DE + byte], [SP + byte] (only the space from F0000H to FFFFFH is specifiable)
–	word[B], word[C] (only the space from F0000H to FFFFFH is specifiable)
–	word[BC] (only the space from F0000H to FFFFFH is specifiable)
–	ES:[HL + byte], ES:[DE + byte] (higher 4-bit addresses are specified by the ES register)
–	ES:word[B], ES:word[C] (higher 4-bit addresses are specified by the ES register)
–	ES:word[BC] (higher 4-bit addresses are specified by the ES register)

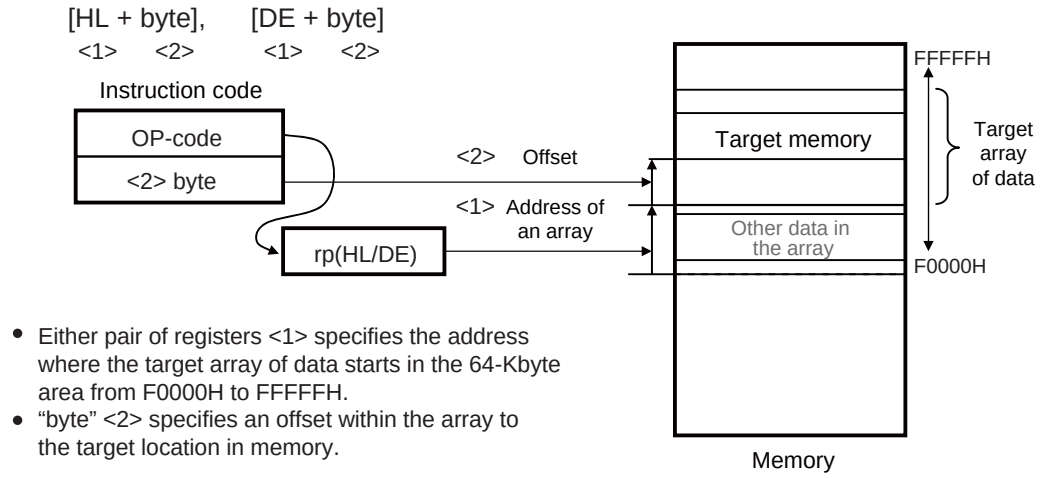
Figure 3-28. Example of [SP+byte]



- SP (stack pointer) <1> indicates the stack as the target.
- By indicating an offset from the address (top of the stack) currently pointed to by the stack pointer, “byte” <2> indicates the target memory (SP + byte).

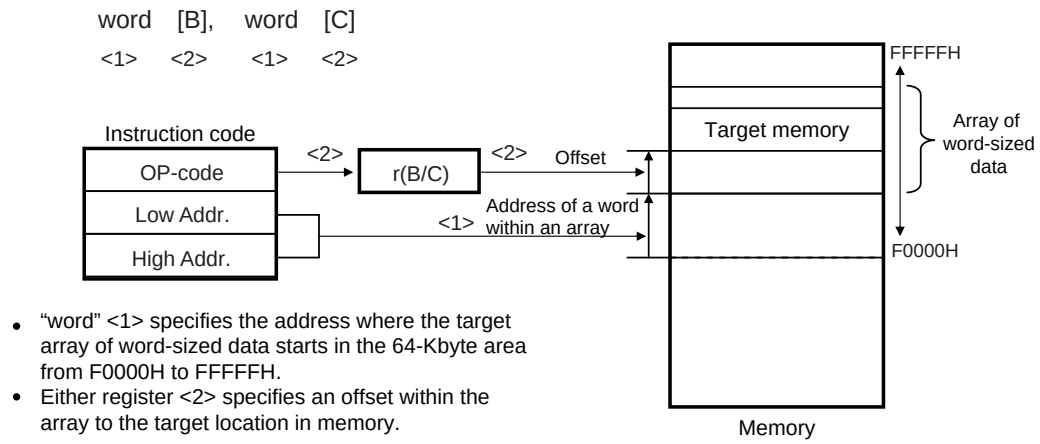
Downloaded from Arrow.com.

Figure 3-29. Example of [HL + byte], [DE + byte]



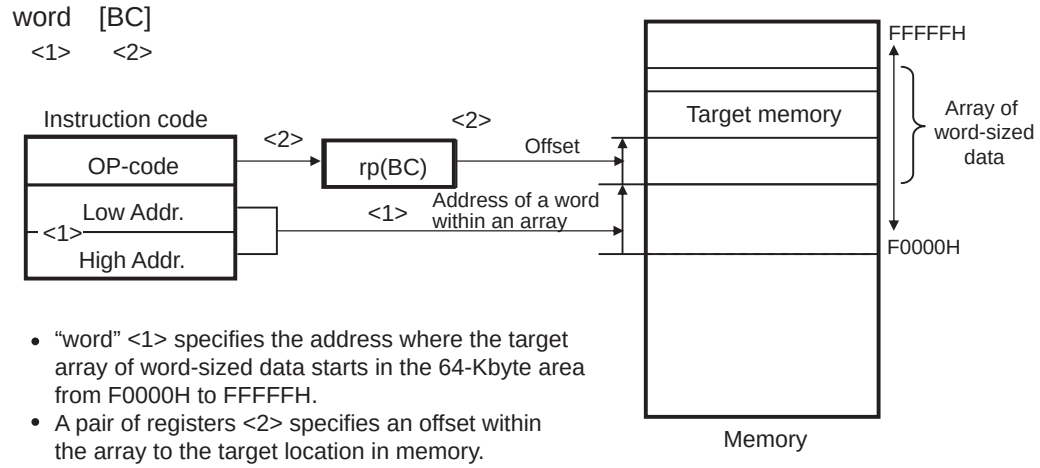
- Either pair of registers <1> specifies the address where the target array of data starts in the 64-Kbyte area from F0000H to FFFFFH.
- "byte" <2> specifies an offset within the array to the target location in memory.

Figure 3-30. Example of word[B], word[C]



- "word" <1> specifies the address where the target array of word-sized data starts in the 64-Kbyte area from F0000H to FFFFFH.
- Either register <2> specifies an offset within the array to the target location in memory.

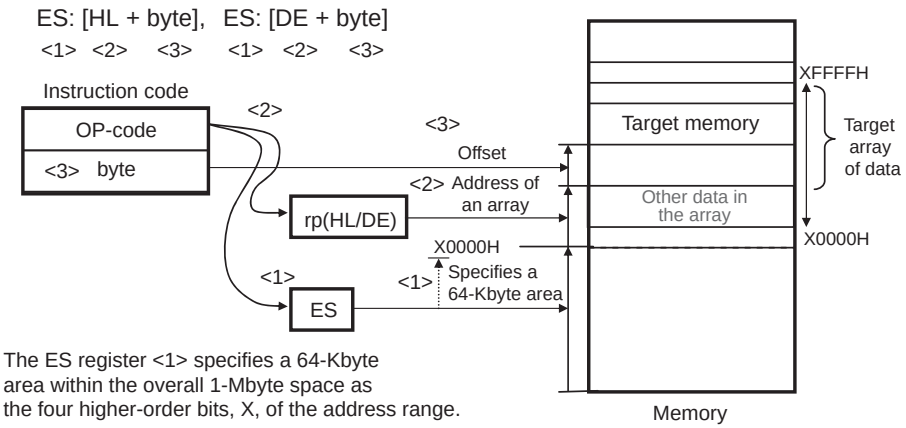
Figure 3-31. Example of word[BC]



- "word" <1> specifies the address where the target array of word-sized data starts in the 64-Kbyte area from F0000H to FFFFFH.
- A pair of registers <2> specifies an offset within the array to the target location in memory.

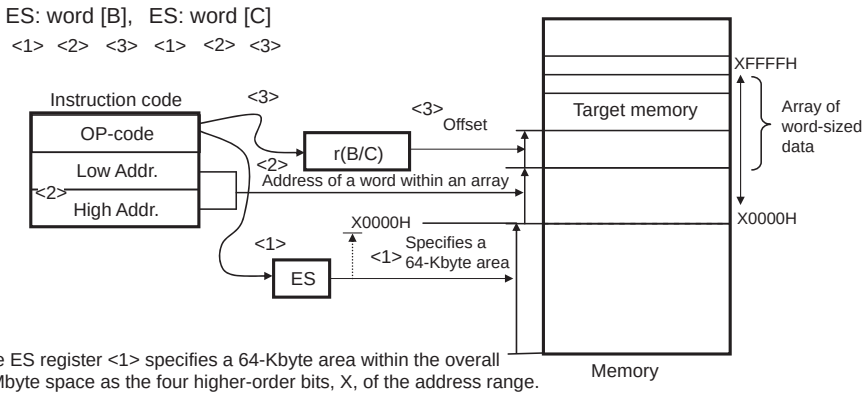
Downloaded from Arrow.com.

Figure 3-32. Example of ES:[HL + byte], ES:[DE + byte]



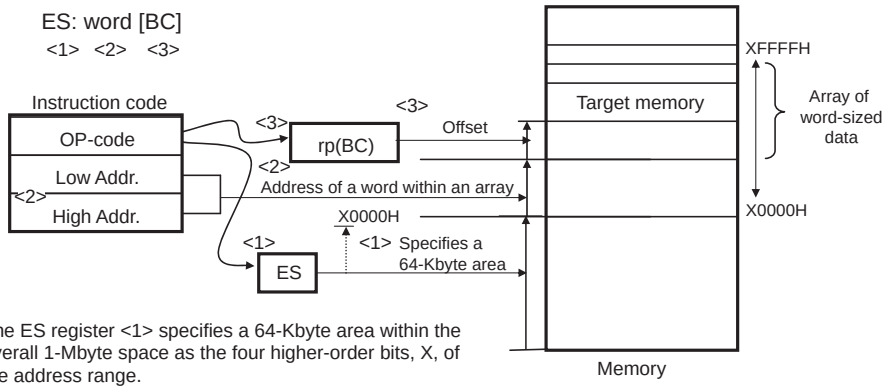
- The ES register <1> specifies a 64-Kbyte area within the overall 1-Mbyte space as the four higher-order bits, X, of the address range.
- Either pair of registers <2> specifies the address where the target array of data starts in the 64-Kbyte area specified in the ES register <1>.
- "byte" <3> specifies an offset within the array to the target location in memory.

Figure 3-33. Example of ES:word[B], ES:word[C]



- The ES register <1> specifies a 64-Kbyte area within the overall 1-Mbyte space as the four higher-order bits, X, of the address range.
- "word" <2> specifies the address where the target array of word-sizeddata starts in the 64-Kbyte area specified in the ES register <1>.
- Either register <3> specifies an offset within the array tothe target location in memory.

Figure 3-34. Example of ES:word[BC]



- The ES register <1> specifies a 64-Kbyte area within the overall 1-Mbyte space as the four higher-order bits, X, of the address range.
- "word" <2> specifies the address where the target array of word-sized data starts in the 64-Kbyte area specified in the ES register <1>.
- A pair of registers <3> specifies an offset within the array to the target location in memory.

Downloaded from Arrow.com.

3.4.8 Based indexed addressing

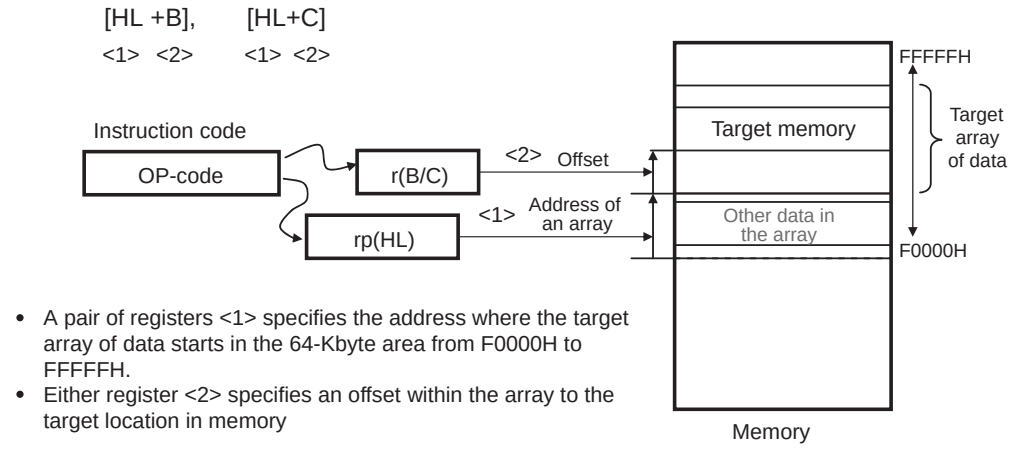
[Function]

Based indexed addressing uses the contents of a register pair specified with the instruction word as the base address, and the content of the B register or C register similarly specified with the instruction word as offset address. The sum of these values is used to specify the target address.

[Operand format]

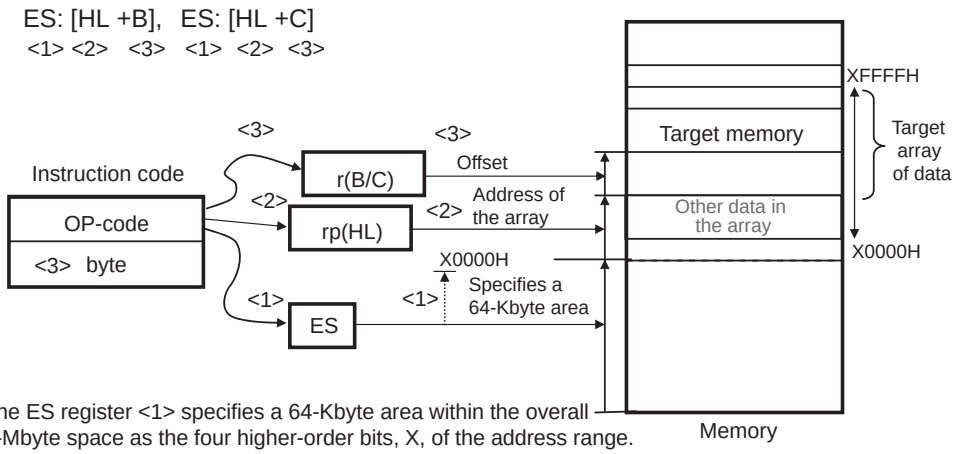
Identifier	Description
–	[HL+B], [HL+C] (only the space from F0000H to FFFFFH is specifiable)
–	ES:[HL+B], ES:[HL+C] (higher 4-bit addresses are specified by the ES register)

Figure 3-35 Example of [HL+B], [HL+C]



- A pair of registers <1> specifies the address where the target array of data starts in the 64-Kbyte area from F0000H to FFFFFH.
- Either register <2> specifies an offset within the array to the target location in memory

Figure 3-36. Example of ES:[HL+B], ES:[HL+C]



- The ES register <1> specifies a 64-Kbyte area within the overall 1-Mbyte space as the four higher-order bits, X, of the address range.
- A pair of registers <2> specifies the address where the target array of data starts in the 64-Kbyte area specified in the ES register <1>.
- Either register <3> specifies an offset within the array to the target location in memory.

Downloaded from Arrow.com.

3.4.9 Stack addressing

[Function]

The stack area is indirectly addressed with the stack pointer (SP) contents. This addressing is automatically employed when the PUSH, POP, subroutine call, and return instructions are executed or the register is saved/restored upon generation of an interrupt request.

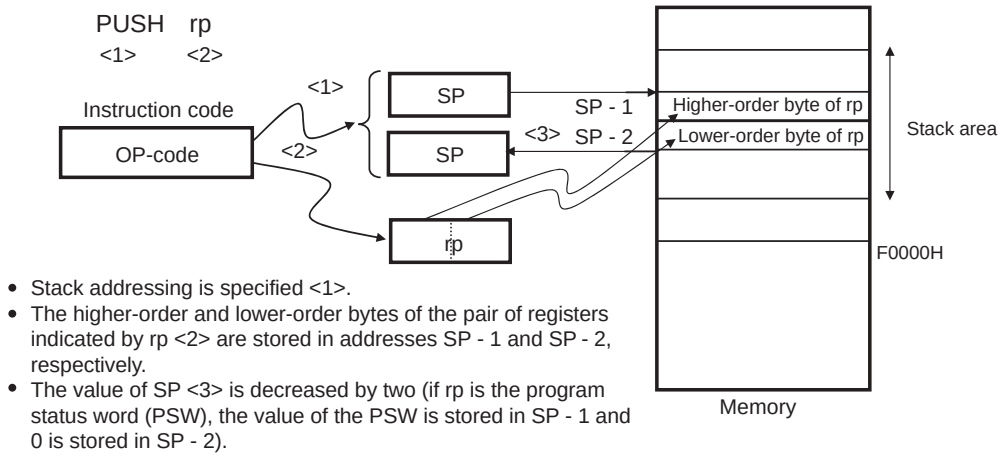
Stack addressing is applied only to the internal RAM area.

[Operand format]

Identifier	Description
–	PUSH AX/BC/DE/HL POP AX/BC/DE/HL CALL/CALLT RET BRK RETB (Interrupt request generated) RETI

Each stack operation saves or restores data as shown in **Figures 3-37** to **3-42**.

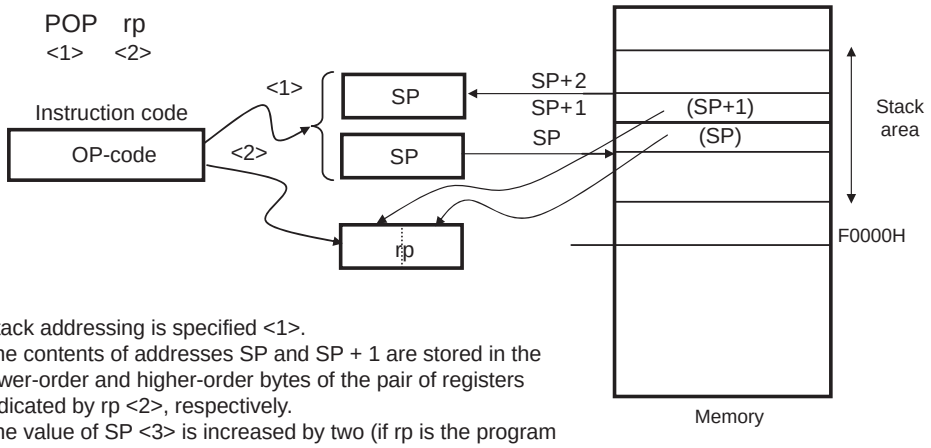
Figure 3-37. Example of PUSH rp



- Stack addressing is specified <1>.
- The higher-order and lower-order bytes of the pair of registers indicated by rp <2> are stored in addresses SP - 1 and SP - 2, respectively.
- The value of SP <3> is decreased by two (if rp is the program status word (PSW), the value of the PSW is stored in SP - 1 and 0 is stored in SP - 2).

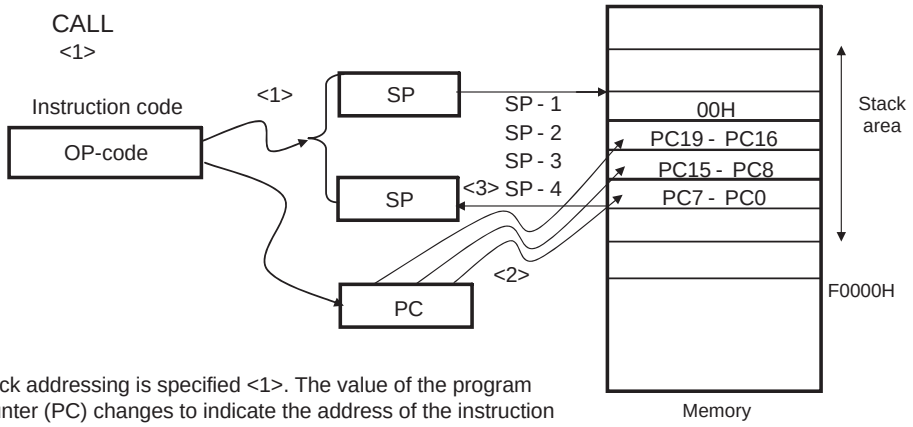
Downloaded from Arrow.com.

Figure 3-38. Example of POP



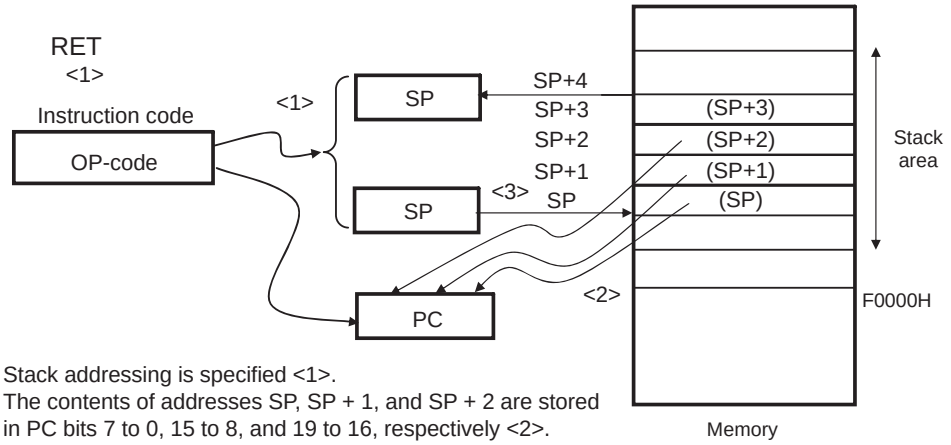
- Stack addressing is specified <1>.
- The contents of addresses SP and SP + 1 are stored in the lower-order and higher-order bytes of the pair of registers indicated by rp <2>, respectively.
- The value of SP <3> is increased by two (if rp is the program status word (PSW), the content of address SP + 1 is stored in the PSW).

Figure 3-39. Example of CALL, CALLT



- Stack addressing is specified <1>. The value of the program counter (PC) changes to indicate the address of the instruction following the CALL instruction.
- 00H, the values of PC bits 19 to 16, 15 to 8, and 7 to 0 are stored in addresses SP - 1, SP - 2, SP - 3, and SP - 4, respectively <2>.
- The value of the SP <3> is decreased by 4.

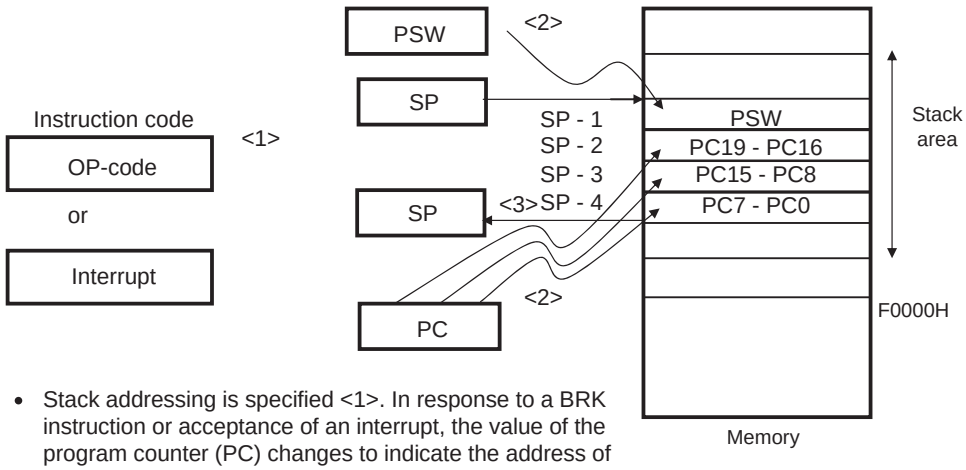
Figure 3-40. Example of RET



- Stack addressing is specified <1>.
- The contents of addresses SP, SP + 1, and SP + 2 are stored in PC bits 7 to 0, 15 to 8, and 19 to 16, respectively <2>.
- The value of SP <3> is increased by four.

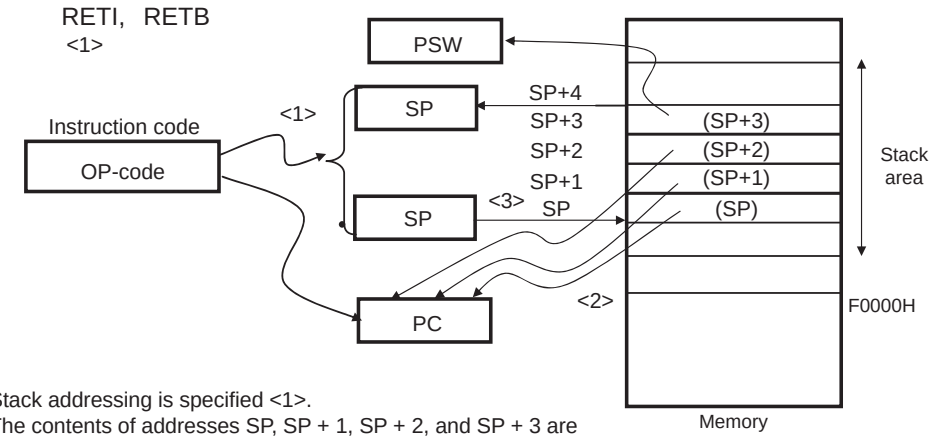
Downloaded from Arrow.com.

Figure 3-41. Example of Interrupt, BRK



- Stack addressing is specified <1>. In response to a BRK instruction or acceptance of an interrupt, the value of the program counter (PC) changes to indicate the address of the next instruction.
- The values of the PSW, PC bits 19 to 16, 15 to 8, and 7 to 0 are stored in addresses SP - 1, SP - 2, SP - 3, and SP - 4, respectively <2>.
- The value of the SP <3> is decreased by 4.

Figure 3-42. Example of RETI, RETB



- Stack addressing is specified <1>.
- The contents of addresses SP, SP + 1, SP + 2, and SP + 3 are stored in PC bits 7 to 0, 15 to 8, 19 to 16, and the PSW, respectively <2>.
- The value of SP <3> is increased by four.

Downloaded from Arrow.com.

CHAPTER 4 PORT FUNCTIONS

4.1 Port Functions

The RL78 microcontrollers are provided with digital I/O ports, which enable variety of control operations.

In addition to the function as digital I/O ports, these ports have several alternate functions. For details of the alternate functions, see **CHAPTER 2 PIN FUNCTIONS**.

4.2 Port Configuration

Ports include the following hardware.

Table 4-1. Port Configuration

Item	Configuration
Control registers	Port mode registers (PM0 to PM6, PM12, PM14) Port registers (P0 to P06, P12 to P14) Pull-up resistor option registers (PU0, PU1, PU3 to PU5, PU12, PU14) Port input mode register (PIM0, PIM1) Port output mode registers (POM0, POM1, POM4, POM5) Port mode control registers (PMC0, PMC1, PMC4, PMC12, PMC14) A/D port configuration register (ADPC) Peripheral I/O redirection register (PIOR)
Port	• 20-pin products <R> Total: 18 (CMOS I/O: 12, (N-ch open drain I/O [V_{DD} tolerance]: 4), CMOS input: 4, N-ch open drain I/O [6V tolerance]: 2) • 24-pin products <R> Total: 22 (CMOS I/O: 16, (N-ch open drain I/O [V_{DD} tolerance]: 5), CMOS input: 4, N-ch open drain I/O [6V tolerance]: 2) • 30-pin products <R> Total: 26 (CMOS I/O: 21, (N-ch open drain I/O [V_{DD} tolerance]: 9), CMOS input: 3, N-ch open drain I/O [6V tolerance]: 2)
Pull-up resistor	• 20-pin products Total: 9 • 24-pin products Total: 13 • 30-pin products Total: 17

Downloaded from Arrow.com.

Caution Most of the following descriptions in this chapter use the R5F102 products.

4.2.1 20, 24-pin products

4.2.1.1 Port 0

Port 0 is an I/O port with an output latch (with 24-pin products). Port 0 can be set to the input mode or output mode in 1-bit units by using port mode register 0 (PM0). When the P00 to P03 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 0 (PU0).

Output from the P01 pin can be specified as N-ch open-drain output (V_{DD} tolerance) in 1-bit units by using port output mode register 0 (POM0).

This port can also be used for key return input, serial interface data I/O, and clock I/O.

Reset signal generation sets port 0 to input mode.

4.2.1.2 Port 1

Port 1 is an I/O port with an output latch. Port 1 can be set to the input mode or output mode in 1-bit units by using port mode register 1 (PM1). When the P10 to P14 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 1 (PU1).

Input to the P10 and P11 pins can be specified through a normal input buffer or a TTL input buffer in 1-bit units by using port input mode register 1 (PIM1).

Output from the P10 to P12 pins can be specified as N-ch open-drain output (V_{DD} tolerance) in 1-bit units by using port output mode register 1 (POM1).

The P10 to P14 pins can be specified for digital I/O or analog input in 1-bit units by using the port mode control register 1 (PMC1).

This port can also be used for analog input, clock/buzzer output, serial interface data I/O, clock I/O, transmission/reception of programming UART, timer I/O, and external interrupt request input.

Reset signal generation sets port 1 to analog input.

4.2.1.3 Port 2

Port 2 is an I/O port with an output latch. Port 2 can be set to the input mode or output mode in 1-bit units by using port mode register 2 (PM2).

This port can also be used for A/D converter analog input and reference voltage input (+ side and - side).

To use the P20/ANI0 to P23/ANI3 pins as a digital I/O port, set them in the digital I/O mode by using the A/D port configuration register (ADPC). Use these pins starting from the upper bit.

To use the P20/ANI0 to P23/ANI3 pins as an analog input port, set them in the analog input mode by using the A/D port configuration register (ADPC) and in the input mode by using the PM2 register. Use these pins starting from the lower bit.

Table 4-2. Setting Functions of P20/ANI0 to P23/ANI3 Pins

ADPC Register	PM2 Register	ADS Register	P20/ANI0 to P23/ANI3 Pins
Digital I/O selection	Input mode	–	Digital input
	Output mode	–	Digital output
Analog input selection	Input mode	Selects ANI.	Analog input (to be converted)
		Does not select ANI.	Analog input (not to be converted)
	Output mode	Selects ANI.	Setting prohibited
		Does not select ANI.	

Reset signal generation sets all the P20/ANI0 to P23/ANI3 pins to analog input.

Downloaded from Arrow.com.

4.2.1.4 Port 4

Port 4 is an I/O port with an output latch. Port 4 can be set to the input mode or output mode in 1-bit units by using port mode register 4 (PM4). When the P40 to P42 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 4 (PU4).

Output from the P41 pin can be specified as N-ch open-drain output (V_{DD} tolerance) by using port output mode register 4 (POM4).

The P41 and P42 pins can be specified for digital I/O or analog input in 1-bit units by using the port mode control register 4 (PMC4).

This port can also be used for key return input, data I/O for a flash memory programmer/debugger, analog input, serial interface data I/O, clock I/O, timer I/O, and external interrupt request input.

Reset signal generation sets port 4 to input mode (the P41 and P42 pins are analog input).

4.2.1.5 Port 6

Port 6 is an I/O port with an output latch. Port 6 can be set to the input mode or output mode in 1-bit units by using port mode register 6 (PM6).

The output of the P60 and P61 pins is N-ch open-drain output (6-V tolerance).

This port can also be used for key return input, serial interface data I/O, and clock I/O.

Reset signal generation sets port 6 to input mode.

4.2.1.6 Port 12

Port 12 is a 3-bit input-only port. Use of an on-chip pull-up resistor can be specified for the P125 pin by using pull-up resistor option register 12 (PU12) (the on-chip pull-up resistor is valid after the $\overline{\text{RESET}}$ pin (PORTSELB = 1) is selected).

This port can also be used for key return input, connecting resonator for main system clock, external clock input for main system clock, and reset input.

Caution After the power is turned on, P125 functions as $\overline{\text{RESET}}$ input. Even if an internal reset signal is released by power-on-reset (POR), the reset status continues as long as the low level is output to this pin.
To use P125/KR1/SI01, select the port function (PORTSELB = 0) by the option byte (000C1H) and clear all reset sources.

4.2.1.7 Port 13

Port 13 is a 1-bit input-only port.

This port can also be used for external interrupt request input.

4.2.2 30-pin products

4.2.2.1 Port 0

Port 0 is an I/O port with an output latch. Port 0 can be set to the input mode or output mode in 1-bit units by using the port mode register 0 (PM0). When the P00 and P01 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 0 (PU0).

Input to the P01 pin can be specified through a normal input buffer or a TTL input buffer by using the port input mode register 0 (PIM0).

Output from the P00 pin can be specified as N-ch open-drain output (V_{DD} tolerance) in 1-bit units by using the port output mode register 1 (POM1).

The P00 and P01 pins can be specified for digital I/O or analog input in 1-bit units by using the port mode control register 1 (PMC1).

This port can also be used for timer I/O, analog input of A/D converter, and serial interface data I/O.

Reset signal generation sets port 0 to analog input.

Downloaded from Arrow.com.

4.2.2.2 Port 1

Port 1 is an I/O port with an output latch. Port 1 can be set to the input mode or output mode in 1-bit units by using port mode register 1 (PM1). When the P10 to P17 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 1 (PU1).

Input to the P10, P11, P13 to P17 pins can be specified through a normal input buffer or a TTL input buffer in 1-bit units by using port input mode register 1 (PIM1).

Output from the P10 to P15, P17 pins can be specified as N-ch open-drain output (V_{DD} tolerance) in 1-bit units by using port output mode register 1 (POM1).

This port can also be used for serial interface data I/O, clock I/O, transmission/reception of programming UART, timer I/O, clock/buzzer output, and external interrupt request input.

Reset signal generation sets port 1 to input mode.

4.2.2.3 Port 2

Port 2 is an I/O port with an output latch. Port 2 can be set to the input mode or output mode in 1-bit units by using port mode register 2 (PM2).

This port can also be used for A/D converter analog input and reference voltage input (+ side and - side).

To use the P20/ANI0 to P23/ANI3 pins as a digital I/O port, set them in the digital I/O mode by using the A/D port configuration register (ADPC). Use these pins starting from the upper bit.

To use the P20/ANI0 to P23/ANI3 pins as an analog input port, set them in the analog input mode by using the ADPC register and in the input mode by using the PM2 register. Use these pins starting from the lower bit.

Table 4-3. Setting Functions of P20/ANI0 to P23/ANI3 Pins

ADPC Register	PM2 Register	ADS Register	P20/ANI0 to P23/ANI3 Pins
Digital I/O selection	Input mode	–	Digital input
	Output mode	–	Digital output
Analog input selection	Input mode	Selects ANI.	Analog input (to be converted)
		Does not select ANI.	Analog input (not to be converted)
	Output mode	Selects ANI.	Setting prohibited
		Does not select ANI.	

Reset signal generation sets port 2 to analog input.

4.2.2.4 Port 3

Port 3 is an I/O port with an output latch. Port 3 can be set to the input mode or output mode in 1-bit units by using port mode register 3 (PM3). When this port is used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 3 (PU3).

This port can also be used for external interrupt request input, serial interface data I/O, clock I/O, clock/buzzer output, and timer I/O.

Reset signal generation sets port 3 to input mode.

4.2.2.5 Port 4

Port 4 is an I/O port with an output latch. Port 4 can be set to the input mode or output mode using port mode register 4 (PM4). When this port is used as an input port, use of an on-chip pull-up resistor can be specified by pull-up resistor option register 4 (PU4).

This port can also be used for data I/O for a flash memory programmer/debugger.

Reset signal generation sets port 4 to input mode.

Downloaded from Arrow.com.

4.2.2.6 Port 5

Port 5 is an I/O port with an output latch. Port 5 can be set to the input mode or output mode in 1-bit units by using port mode register 5 (PM5). When the P50 and P51 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 5 (PU5).

Output from the P50 pin can be specified as N-ch open-drain output (V_{DD} tolerance) using port output mode register 5 (POM5).

This port can also be used for external interrupt request input and serial interface data I/O.

Reset signal generation sets port 5 to input mode.

4.2.2.7 Port 6

Port 6 is an I/O port with an output latch. Port 6 can be set to the input mode or output mode in 1-bit units by using port mode register 6 (PM6).

The output of the P60 and P61 pins is N-ch open-drain output (6-V tolerance).

This port can also be used for serial interface data I/O and clock I/O.

Reset signal generation sets port 6 to input mode.

4.2.2.8 Port 12

The P120 pin is a 1-bit I/O port with an output latch. Port 12 can be set to the input mode or output mode using port mode register 12 (PM12). When the P120 pin is used as an input port, use of an on-chip pull-up resistor can be specified by pull-up resistor option register 12 (PU12).

The P120 pin can be specified for digital I/O or analog input by using the port mode control register 12 (PMC12).

The P121 and P122 pins are 2-bit input-only ports.

This port can also be used for A/D converter analog input, connecting resonator for main system clock, and external clock input for main system clock.

Reset signal generation sets P120 to analog input, and sets P121 and P122 to input mode.

4.2.2.9 Port 13

Port 13 is a 1-bit input-only port.

This port can also be used for external interrupt request input.

4.2.2.10 Port 14

Port 14 is an I/O port with an output latch. Port 14 can be set to the input mode or output mode using port mode register 14 (PM14). When this port is used as an input port, use of an on-chip pull-up resistor can be specified by pull-up resistor option register 14 (PU14).

The P147 pin can be specified for digital I/O or analog input by using the port mode control register 14 (PMC14).

This port can also be used for A/D converter analog input.

Reset signal generation sets port to analog input.

Downloaded from Arrow.com.

4.3 Registers Controlling Port Function

Port functions are controlled by the following registers.

- Port mode registers (PMxx)
- Port registers (Pxx)
- Pull-up resistor option registers (PUxx)
- Port input mode registers (PIMx)
- Port output mode registers (POMx)
- Port mode control registers (PMCxx)
- A/D port configuration register (ADPC)
- Peripheral I/O redirection register (PIOR)

Caution Which registers and bits are included depends on the product. For details about registers and bits mounted on each product, see Tables 4-4 and 4-5. Be sure to set bits that are not mounted to their initial values.

Table 4-4. PMxx, Pxx, PUxx, PIMx, POMx, and PMCxx Registers and the Bits (20-, 24-pin Products)

Port		Bit name					
		PMxx register	Pxx register	PUxx register	PIMx register	POMx register	PMCxx register
Port 0[Note]	0	PM00	P00	PU00	–	–	–
	1	PM01	P01	PU01	–	POM01	–
	2	PM02	P02	PU02	–	–	–
	3	PM03	P03	PU03	–	–	–
Port 1	0	PM10	P10	PU10	PIM10	POM10	PMC10
	1	PM11	P11	PU11	PIM11	POM11	PMC11
	2	PM12	P12	PU12	–	POM12	PMC12
	3	PM13	P13	PU13	–	–	PMC13
	4	PM14	P14	PU14	–	–	PMC14
Port 2	0	PM20	P20	–	–	–	–
	1	PM21	P21	–	–	–	–
	2	PM22	P22	–	–	–	–
	3	PM23	P23	–	–	–	–
Port 4	0	PM40	P40	PU40	–	–	–
	1	PM41	P41	PU41	–	POM41	PMC41
	2	PM42	P42	PU42	–	–	PMC42
Port 6	0	PM60	P60	–	–	–	–
	1	PM61	P61	–	–	–	–
Port 12	1	–	P121	–	–	–	–
	2	–	P122	–	–	–	–
	5	–	P125	PU125	–	–	–
Port 13	7	–	P137	–	–	–	–

Note Provided only in 24-pin products.

Downloaded from Arrow.com.

Table 4-5. PMxx, Pxx, PUxx, PIMx, POMx, PMCxx Registers and the Bits (30-pin Products)

Port		Bit name					
		PMxx register	Pxx register	PUxx register	PIMx register	POMx register	PMCxx register
Port 0	0	PM00	P00	PU00	–	POM00	PMC00
	1	PM01	P01	PU01	PIM01	–	PMC01
Port 1	0	PM10	P10	PU10	PIM10	POM10	–
	1	PM11	P11	PU11	PIM11	POM11	–
	2	PM12	P12	PU12	–	POM12	–
	3	PM13	P13	PU13	PIM13	POM13	–
	4	PM14	P14	PU14	PIM14	POM14	–
	5	PM15	P15	PU15	PIM15	POM15	–
	6	PM16	P16	PU16	PIM16	–	–
	7	PM17	P17	PU17	PIM17	POM17	–
Port 2	0	PM20	P20	–	–	–	–
	1	PM21	P21	–	–	–	–
	2	PM22	P22	–	–	–	–
	3	PM23	P23	–	–	–	–
Port 3	0	PM30	P30	PU30	–	–	–
	1	PM31	P31	PU31	–	–	–
Port 4	0	PM40	P40	PU40	–	–	–
Port 5	0	PM50	P50	PU50	–	POM50	–
	1	PM51	P51	PU51	–	–	–
Port 6	0	PM60	P60	–	–	–	–
	1	PM61	P61	–	–	–	–
Port 12	0	PM120	P120	PU120	–	–	PMC120
	1	–	P121	–	–	–	–
	2	–	P122	–	–	–	–
Port 13	7	–	P137	–	–	–	–
Port 14	7	PM147	P147	PU147	–	–	PMC147

Downloaded from Arrow.com.

4.3.1 Port mode registers (PMxx)

These registers specify input or output mode for the port in 1-bit units.

These registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

When port pins are used as alternate-function pins, set the port mode register by referencing **4.5 Register Settings When Using Alternate Function**.

Figure 4-1. Format of Port Mode Register

20-, 24-pin products

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM0[Note]	1	1	1	1	PM03	PM02	PM01	PM00	FFF20H	FFH	R/W
PM1	1	1	1	PM14	PM13	PM12	PM11	PM10	FFF21H	FFH	R/W
PM2	1	1	1	1	PM23	PM22	PM21	PM20	FFF22H	FFH	R/W
PM4	1	1	1	1	1	PM42	PM41	PM40	FFF24H	FFH	R/W
PM6	1	1	1	1	1	1	PM61	PM60	FFF26H	FFH	R/W

Note Provided only in 24-pin products.

30-pin products

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM0	1	1	1	1	1	1	PM01	PM00	FFF20H	FFH	R/W
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10	FFF21H	FFH	R/W
PM2	1	1	1	1	PM23	PM22	PM21	PM20	FFF22H	FFH	R/W
PM3	1	1	1	1	1	1	PM31	PM30	FFF23H	FFH	R/W
PM4	1	1	1	1	1	1	1	PM40	FFF24H	FFH	R/W
PM5	1	1	1	1	1	1	PM51	PM50	FFF25H	FFH	R/W
PM6	1	1	1	1	1	1	PM61	PM60	FFF26H	FFH	R/W
PM12	1	1	1	1	1	1	1	PM120	FFF2CH	FFH	R/W
PM14	PM147	1	1	1	1	1	1	1	FFF2EH	FFH	R/W

PMmn	Pmn pin I/O mode selection (m = 0 to 6, 12, 14; n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

<R> **Caution Be sure to set bits that are not mounted to their initial values.**

Downloaded from Arrow.com.

4.3.2 Port registers (Pxx)

These registers set the output latch value of a port.

If the data is read in the input mode, the pin level is read. If it is read in the output mode, the output latch value is read[Note].

These registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Note If the input data is read from a port for which analog input of the A/D converter is selected, 0 is always returned rather than the pin level.

In addition, if the input data is read from P125 when the $\overline{\text{RESET}}$ pin (PORTSELB = 1) is selected, 1 is always read.

Downloaded from Arrow.com.

Figure 4-2. Format of Port Register

20-, 24-pin products

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
P0	0	0	0	0	P03	P02	P01	P00	FFF00H	00H (output latch)	R/W
P1	0	0	0	P14	P13	P12	P11	P10	FFF01H	00H (output latch)	R/W
P2	0	0	0	0	P23	P22	P21	P20	FFF02H	00H (output latch)	R/W
P4	0	0	0	0	0	P42	P41	P40	FFF04H	00H (output latch)	R/W
P6	0	0	0	0	0	0	P61	P60	FFF06H	00H (output latch)	R/W
P12	0	0	P125	0	0	P122	P121	0	FFF0CH	Undefined	R
P13	P137	0	0	0	0	0	0	0	FFF0DH	Undefined	R

30-pin products

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
P0	0	0	0	0	0	0	P01	P00	FFF00H	00H (output latch)	R/W
P1	P17	P16	P15	P14	P13	P12	P11	P10	FFF01H	00H (output latch)	R/W
P2	0	0	0	0	P23	P22	P21	P20	FFF02H	00H (output latch)	R/W
P3	0	0	0	0	0	0	P31	P30	FFF03H	00H (output latch)	R/W
P4	0	0	0	0	0	0	0	P40	FFF04H	00H (output latch)	R/W
P5	0	0	0	0	0	0	P51	P50	FFF05H	00H (output latch)	R/W
P6	0	0	0	0	0	0	P61	P60	FFF06H	00H (output latch)	R/W
P12	0	0	0	0	0	P122	P121	P120	FFF0CH	Undefined	R/W[Note]
P13	P137	0	0	0	0	0	0	0	FFF0DH	Undefined	R
P14	P147	0	0	0	0	0	0	0	FFF0EH	00H (output latch)	R/W

Pmn	Output data control (in output mode)	Input data read (in input mode)
0	Output 0	Input low level
1	Output 1	Input high level

m = 0 to 6, 12, 13, 14; n = 0 to 7

Note P121 and P122 are read-only.

<R> **Caution Be sure to set bits that are not mounted to their initial values.**

Downloaded from Arrow.com.

<R> **4.3.3 Pull-up resistor option registers (PUxx)**

These registers specify whether the on-chip pull-up resistors are to be used or not. On-chip pull-up resistors can be used in 1-bit units only for the bits set to both normal output mode (POMmn = 0) and input mode (PMmn = 1) for the pins for which the use of an on-chip pull-up resistor is specified in these registers. On-chip pull-up resistors cannot be connected to bits set to output mode and bits used as alternate-function output pins and analog setting (PMC = 1, ADPC = 1), regardless of the settings of these registers.

These registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets PU4 to 01H, PU12 to 20H (20-, 24-pin products), and others to 00H.

Caution When a port with the PIMn register is input from a different-potential device to the TTL buffer, pull up to the power supply of the different-potential device via an external pull-up resistor by setting PUmn = 0.

Figure 4-3. Format of Pull-up Resistor Option Register

20-, 24-pin products

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PU0	0	0	0	0	PU03	PU02	PU01	PU00	F0030H	00H	R/W
PU1	0	0	0	PU14	PU13	PU12	PU11	PU10	F0031H	00H	R/W
PU4	0	0	0	0	0	PU42	PU41	PU40	F0034H	01H	R/W
PU12	0	0	PU125Note	0	0	0	0	0	F003CH	20H	R/W

30-pin products

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PU0	0	0	0	0	0	0	PU01	PU00	F0030H	00H	R/W
PU1	PU17	PU16	PU15	PU14	PU13	PU12	PU11	PU10	F0031H	00H	R/W
PU3	0	0	0	0	0	0	PU31	PU30	F0033H	00H	R/W
PU4	0	0	0	0	0	0	0	PU40	F0034H	01H	R/W
PU5	0	0	0	0	0	0	PU51	PU50	F0035H	00H	R/W
PU12	0	0	0	0	0	0	0	PU120	F003CH	00H	R/W
PU14	PU147	0	0	0	0	0	0	0	F003EH	00H	R/W

PUmn	Pmn pin on-chip pull-up resistor selection (m = 0, 1, 3 to 5, 12, 14; n = 0 to 7)
0	On-chip pull-up resistor not connected.
1	On-chip pull-up resistor connected.

Note PU125 can be selected only when P125/KR1/SI01 (PORTSELB = 0) is selected.
If the $\overline{\text{RESET}}$ pin (PORTSELB = 1) is selected, the on-chip pull-up resistor is always valid.

<R> **Caution Be sure to set bits that are not mounted to their initial values.**

Downloaded from Arrow.com.

4.3.4 Port input mode register (PIMx)

These registers set an input buffer in 1-bit units.

TTL input buffer can be selected during serial communication with an external device of a different potential.

This register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 4-4. Format of Port Input Mode Register

20-, 24-pin products

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PIM1	0	0	0	0	0	0	PIM11	PIM10	F0041H	00H	R/W

30-pin products

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PIM0	0	0	0	0	0	0	PIM11	0	F0040H	00H	R/W
PIM1	PIM17	PIM16	PIM15	PIM14	PIM13	0	PIM11	PIM10	F0041H	00H	R/W

PIMmn	Pmn pin input buffer selection (m = 0, 1; n = 0, 1, 3 to 7)
0	Normal input buffer
1	TTL input buffer

<R> **Caution Be sure to set bits that are not mounted to their initial values.**

Downloaded from Arrow.com.

4.3.5 Port output mode registers (POMx)

These registers set the output mode in 1-bit units.

N-ch open drain output (V_{DD} tolerance) mode can be selected for the SDAxx pin during serial communication with an external device of a different potential or during simplified I^2C communication with an external device of the same potential.

In addition, the POMx and PUx registers specify whether to use the on-chip pull-up resistor.

The POMx registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

<R> **Caution An on-chip pull-up resistor is not connected to a bit for which N-ch open drain output (V_{DD} tolerance) mode (POMmn = 1) is set.**

Figure 4-5. Format of Port Output Mode Register

20-, 24-pin products

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
POM0	0	0	0	0	0	0	POM01	0	F0050H	00H	R/W
POM1	0	0	0	0	0	POM12	POM11	POM10	F0051H	00H	R/W
POM4	0	0	0	0	0	0	POM41	0	F0054H	00H	R/W

30-pin products

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
POM0	0	0	0	0	0	0	0	POM01	F0050H	00H	R/W
POM1	POM17	0	POM15	POM14	POM13	POM12	POM11	POM10	F0051H	00H	R/W
POM5	0	0	0	0	0	0	0	POM50	F0055H	00H	R/W

POMmn	Pmn pin output mode selection (m = 0, 1, 4, 5; n = 0 to 7)
0	Normal output mode
1	N-ch open-drain output (V_{DD} tolerance) mode

<R> **Caution Be sure to set bits that are not mounted to their initial values.**

Downloaded from Arrow.com.

4.3.6 Port mode control registers (PMCxx)

These registers set the digital I/O or analog input in 1-bit units.

These registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Figure 4-6. Format of Port Mode Control Register

20-, 24-pin products

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PMC1	1	1	1	PMC14	PMC13	PMC12	PMC11	PMC10	F0061H	FFH	R/W
PMC4	1	1	1	1	1	PMC42	PMC41	1	F0064H	FFH	R/W

30-pin products

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PMC0	1	1	1	1	1	1	PMC01	PMC00	F0060H	FFH	R/W
PMC12	1	1	1	1	1	1	1	PMC120	F006CH	FFH	R/W
PMC14	PMC147	1	1	1	1	1	1	1	F006EH	FFH	R/W

PMCmn	Pmn pin digital I/O/analog input selection (m = 1, 4, 12, 14; n = 0 to 4, 7)
0	Digital I/O (alternate function other than analog input)
1	Analog input

<R> **Cautions 1. Use the port mode register m (PMm) to select the input mode for the ports that are set to analog input by using the PMCxx register.**

2. Do not use the analog input channel specification register (ADS) to set the pins that will be set to digital I/O by using the PMCxx register.

<R> **3. Be sure to set bits that are not mounted to their initial values.**

Downloaded from Arrow.com.

4.3.7 A/D port configuration register (ADPC)

This register switches the P20/ANI0 to P23/ANI3 pins to digital I/O of port or analog input of A/D converter.

This register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 4-7. Format of A/D Port Configuration Register (ADPC)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
ADPC	0	0	0	0	0	ADPC2	ADPC1	ADPC0	F0076H	00H	R/W

ADPC2	ADPC1	ADPC0	Analog input (A)/digital I/O (D) switching			
			ANI3/P23	ANI2/P22	ANI1/P21	ANI0/P20
0	0	0	A	A	A	A
0	0	1	D	D	D	D
0	1	0	D	D	D	A
0	1	1	D	D	A	A
1	0	0	D	A	A	A
Other than the above			Setting prohibited			

Cautions 1. Use the port mode register 2 (PM2) to select the input mode for the ports that are set to analog input by using the ADPC register.

2. Do not use the analog input channel specification register (ADS) to set the pins that will be set to digital I/O by using the ADPC register.

3. To use AV_{REFP} and AV_{REFM}, set ANI0 and ANI1 to analog input and set the port mode register to the input mode.

4.3.8 Peripheral I/O redirection register (PIOR)

This register is used to specify whether to enable or disable the peripheral I/O redirect function.

This function is used to switch ports to which alternate functions are assigned.

Use the PIOR register to assign a port to the function to redirect and enable the function.

In addition, the settings for redirection can be changed only until operation of the function is enabled.

The PIOR register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Downloaded from Arrow.com.

Figure 4-8. Format of Peripheral I/O Redirection Register (PIOR)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PIOR	0	0	0	0	PIOR3 Notes 1, 2	PIOR2	PIOR1	PIOR0	F0077H	00H	R/W

20-, 24-pin products

Bit	Alternate function	Setting value	
		0	1
PIOR3 Notes 1, 2	SCK01	P42	P02
	SI01	P125	P00
	SO01	P41	P01
	SCL01	P42	P02
	SDA01	P41	P01
PIOR2 Note 3	TI02	P41	P122
	TI03	P42	P121
PIOR1	RxD0	P11	P61
	TxD0	P12	P60
PIOR0	INTP2	P13	P122
	INTP3	P14	P121

30-pin products

Bit	Alternate function	Setting value	
		0	1
PIOR3	–	(fixed)	Setting prohibited
PIOR2	SCLA0	P60	P14
	SDAA0	P61	P13
PIOR1	TxD2 Note 1	P13	–
	RxD2 Note 1	P14	–
	SCL20 Note 1	P15	–
	SDA20 Note 1	P14	–
	SI20 Note 1	P14	–
	SO20 Note 1	P13	–
	SCK20 Note 1	P15	–
	TxD0	P12	P17
	RxD0	P11	P16
	SCL00 Note 1	P10	–
	SDA00 Note 1	P11	–
	SI00	P11	–
	SO00	P12	–
	SCK00	P10	–
PIOR0	TI02/TO02	P17	P15
	TI03/TO03	P31	P14
	TI04/TO04	–	P13
	TI05/TO05	–	P12
	TI06/TO06	–	P11
	TI07/TO07	–	P10

Notes 1. Provided only in the R5F102 products.

2. Provided only in 24-pin products.

<R> **3.** When the PIOR2 bit is set to 1 in the 20- or 24-pin products, using TO02 and TO03 for timer output is prohibited.

Remark –: The function cannot be used.

Downloaded from Arrow.com.

4.4 Port Function Operations

Port operations differ depending on whether the input or output mode is set, as shown below.

4.4.1 Writing to I/O port

(1) Output mode

A value is written to the output latch by a transfer instruction, and the output latch contents are output from the pin.
Once data is written to the output latch, it is retained until data is written to the output latch again.
The data of the output latch is cleared when a reset signal is generated.

(2) Input mode

A value is written to the output latch by a transfer instruction, but since the output buffer is off, the pin status does not change. Therefore, byte data can be written to the ports used for both input and output.
Once data is written to the output latch, it is retained until data is written to the output latch again.
The data of the output latch is cleared when a reset signal is generated.

4.4.2 Reading from I/O port

(1) Output mode

The output latch contents are read by a transfer instruction. The output latch contents do not change.

(2) Input mode

The pin status is read by a transfer instruction. The output latch contents do not change.

4.4.3 Operations on I/O port

(1) Output mode

An operation is performed on the output latch contents, and the result is written to the output latch. The output latch contents are output from the pins.
Once data is written to the output latch, it is retained until data is written to the output latch again.
The data of the output latch is cleared when a reset signal is generated.

(2) Input mode

The pin level is read and an operation is performed on its contents. The result of the operation is written to the output latch, but since the output buffer is off, the pin status does not change. Therefore, byte data can be written to the ports used for both input and output.
The data of the output latch is cleared when a reset signal is generated.

Downloaded from Arrow.com.

4.4.4 Handling different potentials (1.8 V, 2.5 V, and 3 V) by using I/O buffers

Connection with an external device operating on a different potential (1.8 V, 2.5 V or 3 V) is possible by switching I/O buffers by using the port input mode register (PIMxx) and port output mode register (POMxx).

To receive input from an external device with a different potential (1.8 V, 2.5 V or 3 V), set the port input mode registers 0 and 1 (PIM0 and PIM1)[Note] on a bit-by-bit basis to enable switching between normal input (CMOS) and TTL input buffer.

To output data to an external device with a different potential (1.8 V, 2.5 V or 3 V), set the port output mode registers 0 and 1 (POM0 and POM1) on a bit-by-bit basis to enable switching between normal output (CMOS) and N-ch open drain (V_{DD} tolerance).

The following describes the connection of a serial interface.

Note For 20- and 24-pin products, only port 1 can be used.

(1) Setting procedure when using input ports of UART0 to UART2, CSI00 and CSI20 functions for the TTL input buffer

Interface	20, 24-pin product	30-pin product[Note]
UART0	P11	P11 (P16)
UART1	–	P01
UART2	–	P14
CSI00	P10	P10
	P11	P11
CSI20	–	P14
		P15

Note The pin in parentheses can be assigned by setting the peripheral I/O redirection register (PIOR).

<1> Use an external register to pull up the pin to be used to the power supply of the target device (on-chip pull-up resistor cannot be used).

<2> Set the corresponding bit of the PIM0 and PIM1 registers to 1 to switch to the TTL input buffer. For V_{IH} and V_{IL}, see the DC characteristics when the TTL input buffer is selected.

<3> Enable the operation of the serial array unit and set the mode to the UART/CSI mode.

Downloaded from Arrow.com.

(2) Setting procedure when using output ports of UART0 to UART2, CSI00 and CSI20 functions in the N-ch open drain output mode

Interface	20, 24-pin product	30-pin product[Note]
UART0	P12	P12 (P17)
UART1	–	P00
UART2	–	P13
CSI00	P10	P10
	P12	P12
CSI20	–	P15
		P13

Note The pin in parentheses can be assigned by setting the peripheral I/O redirection register (PIOR).

<1> Use an external register to pull up the output pin to be used to the power supply of the target device (on-chip pull-up resistor cannot be used).
<2> After reset release, the port mode is the input mode (Hi-Z).
<3> Set the output latch of the corresponding port to 1.
<4> Set the corresponding bit of the POM1 register to 1 to set the N-ch open drain output (V_{DD} tolerance) mode.
<5> Enable the operation of the serial array unit and set the mode to the UART/CSI mode.
<6> Set the output mode by manipulating the PM0 and PM1 registers. At this time, the output data is high level, so the pin is in the Hi-Z state.

(3) Setting procedure when using I/O ports of IIC00 and IIC20 functions for a different potential (1.8 V, 2.5 V or 3 V)

Interface	20, 24-pin product	30-pin product
IIC00	P10	P10
	P11	P11
IIC20	–	P14
		P15

<1> Use an external register to pull up the input pin to be used to the power supply of the target device (on-chip pull-up resistor cannot be used).
<2> After reset release, the port mode is the input mode (Hi-Z).
<3> Set the output latch of the corresponding port to 1.
<4> Set the corresponding bit of the POM0 and POM1 registers to 1 to set the N-ch open drain output (V_{DD} tolerance) mode.
<5> Set the corresponding bit of the POM0 and PIM1 registers to 1 to switch to the TTL input buffer. For V_{IH} and V_{IL}, see the DC characteristics when the TTL input buffer is selected.
<6> Enable the operation of the serial array unit and set the mode to the simplified I^2C mode.
<7> Set the output mode by manipulating the PM0 and PM1 registers (data I/O is possible in the output mode). At this time, the output data is high level, so the pin is in the Hi-Z state.

Downloaded from Arrow.com.

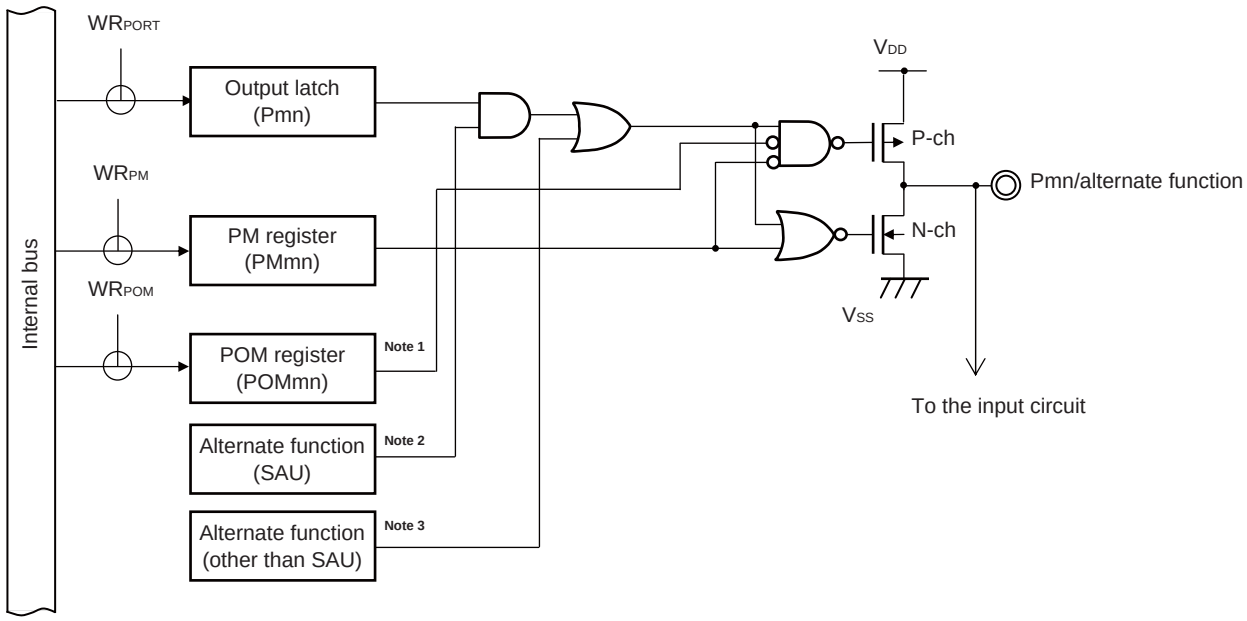
<R> 4.5 Register Settings When Using Alternate Function

<R> 4.5.1 Basic concept when using alternate function

First, for a pin that is also assigned to be used for analog input, use the ADPC register or port mode control register (PMCxx) to specify whether to use the pin for analog input or digital I/O.

Figure 4-9 shows the basic configuration of an output circuit for pins used for digital I/O. The output of the output latch for the port and the output of the alternate SAU function are input to an AND gate. The output of the AND gate is input to an OR gate. The output of an alternate function other than SAU (such as TAU, clock/buzzer output, or IICA) is connected to the other input pin of the OR gate. When such kind of pins are used for the port function or alternate function, the unused alternate function must not hinder the output of the function to be used. Table 4-6 shows the concept of basic settings for this configuration.

Figure 4-9. Basic Configuration of Output Circuit for Pins



Notes 1. If there is no POM register, this signal should be considered to be low level (0).

2. If there is no alternate function, this signal should be considered to be high level (1).

3. If there is no alternate function, this signal should be considered to be low level (0).

Remark m: Port number (m = 0 to 14), n: Bit number (n = 0 to 7)

Downloaded from Arrow.com.

Table 4-6. Concept of Basic Settings

Output Function of Used Pin	Output Function of Unused Alternate Function		
	Port output function	SAU output function	Output function for other than SAU
Port output function	–	Output is high (1)	Output is low (0)
Output function for SAU	High (1)	–	Output is low (0)
Output function for other than SAU	Low (0)	Output is high (1)	Output is low (0)Note

Note Because more than one output function other than SAU might be assigned to a single pin, the output of unused alternate functions must be set to low level (0). For details about the setting method, see **4.5.2 Register settings for alternate function whose output function is not used**.

<R> **4.5.2 Register settings for alternate function whose output function is not used**

If the output of an alternate function is not used, specify the settings as described below. If the output of the peripheral function is still the target of the peripheral I/O redirection function, the output can be switched to another pin by setting the peripheral I/O redirection register (PIOR). This allows usage of the port function or other alternate functions assigned to the target pin.

(1) SOp = 1 and TxDq = 1 (settings when the serial output (SOp/TxDq) of SAU is not used)

If the serial output (SOp/TxDq) is not used, such as a case in which only the serial input of SAU is used, set the bit in the serial output enable register m (SOEm) that corresponds to the unused output to 0 (output disabled) and set the SOmn bit in the serial output register m (SOm) to 1 (high). These settings are the same as the initial state.

(2) SCKp = 1, SDAr = 1, and SCLr = 1 (settings when channel n in SAU is not used)

If SAU is not used, set the bit n (SEmn) in the serial channel enable status register m (SEm) to 0 (operation stopped state), set the bit in the serial output enable register m (SOEm) that corresponds to the unused output to 0 (output disabled), and set the SOmn bit and CKOmn bit in the serial output register m (SOm) to 1 (high). These settings are the same as the initial state.

(3) TOmn = 0 (settings when the output of channel n in TAU is not used)

If the TOmn output of TAU is not used, set the bit in the timer output enable register 0 (TOE0) that corresponds to the unused output to 0 (output disabled) and set the bit in the timer output register 0 (TO0) to 0 (low). These settings are the same as the initial state.

(4) SDAAn = 0 and SCLAn = 0 (setting when IICA is not used)

If IICA is not used, set the IICEn bit in the IICA control register n0 (IICCTLn0) to 0 (operation stopped). This setting is the same as the initial state.

(5) PCLBUZn = 0 (setting when clock output or buzzer output is not used)

If the clock output or buzzer output is not used, set the PCLOEn bit in the clock output selection register n (CKSn) to 0 (output disabled). This setting is the same as the initial state.

Downloaded from Arrow.com.

<R> 4.5.3 Register setting examples for using the port and alternate functions

Table 4-7 shows register setting examples for used port and alternate functions. Set the registers used to control the port function as shown in Table 4-7. See the following remark for legends used in Table 4-7.

Remark –: Not applicable
×: don't care
PIORx: Peripheral I/O redirection register
POMxx: Port output mode register
PMCxx: Port mode control register
PMxx: Port mode register
Pxx: Port output latch
Functions in parentheses can be assigned via settings in the peripheral I/O redirection register (PIOR).

Table 4-7. Setting Examples of Registers and Output Latches When Using Alternate Function (20-, 24-Pin Products) (1/5)

<R>

Pin Name	Used Function		PIORx	POMxx	PMCxx	PMxx	Pxx	Alternate Function Output	
	Function Name	I/O						SAU	Other than SAU
P00[Note 1]	P00	Input	–	–	–	1	×	–	–
		Output	–	–	–	0	0/1	–	–
	KR6	Input	–	–	–	1	×	–	–
	(SI01)[Note 2]	Input	PIOR3 = 1	–	–	1	×	–	–
P01[Note 1]	P01	Input	–	×	–	1	×	×	–
		Output	–	0	–	0	0/1	(SO01)/(SDA01) = 1	–
		N-ch open drain output	–	1	–	0	0/1		–
	KR7	Input	–	×	–	1	×	×	–
	(SO01)[Note 2]	Output	PIOR3 = 1	0/1	–	0	1	×	–
	(SDA01)[Note 2]	I/O	PIOR3 = 1	1	–	0	1	×	–
P02[Note 1]	P02	Input	–	–	–	1	×	×	–
		Output	–	–	–	0	0/1	(SCK01)/(SCL01) = 1	–
	KR8	Input	–	–	–	1	×	×	–
	(SCK01)[Note 2]	Input	PIOR3 = 1	–	–	1	×	×	–
		Output	PIOR3 = 1	–	–	0	1	×	–
	(SCL01)[Note 2]	Output	PIOR3 = 1	–	–	0	1	×	–
P03[Note 1]	P03	Input	–	–	–	1	×	–	–
		Output	–	–	–	0	0/1	–	–
	KR9	Input	–	–	–	1	×	–	–
P10	P10	Input	–	×	0	1	×	×	×
		Output	–	0	0	0	0/1	SCK00/ SCL00 = 1	PCLBZ0 = 0
		N-ch open drain output	–	1	0	0	0/1		
	ANI16	Input	–	×	1	1	×	×	×
	PCLBZ0	Output	–	0	0	0	0	SCK00/SCL00 = 1	×
	SCK00	Input	–	×	0	1	×	×	PCLBZ0 = 0
		Output	–	0/1	0	0	1	×	PCLBZ0 = 0
	SCL00[Note 2]	Output	–	0/1	0	0	1	×	PCLBZ0 = 0

Notes 1. Provided only in 24-pin products.
2. Provided only in the R5F102 products.

Downloaded from Arrow.com.

Table 4-7. Setting Examples of Registers and Output Latches When Using Alternate Function (20-, 24-Pin Products) (2/5)

<R>

Pin Name	Used Function		PIORx	POMxx	PMCxx	PMxx	Pxx	Alternate Function Output	
	Function Name	I/O						SAU	Other than SAU
P11	P11	Input	–	×	0	1	×	×	–
		Output	–	0	0	0	0/1	SDA00 = 1	–
		N-ch open drain output	–	1	0	0	0/1		
	ANI17	Input	–	×	1	1	×	×	–
	SI00	Input	–	×	0	1	×	×	–
	RxD0	Input	PIOR1 = 0	×	0	1	×	×	–
	SDA00Note	I/O	–	1	0	0	1	×	–
P12	P12	Input	–	×	0	1	×	×	–
		Output	–	0	0	0	0/1	SO00/TxD0 = 1	–
		N-ch open drain output	–	1	0	0	0/1		
	ANI18	Input	–	×	1	1	×	×	–
	SO00	Output	–	0/1	0	0	1	×	–
	TxD0	Output	PIOR1 = 0	0/1	0	0	1	×	–
P13	P13	Input	–	–	0	1	×	×	×
		Output	–	–	0	0	0/1	–	TO00 = 0
	ANI19	Input	–	–	1	1	×	–	×
	TI00	Input	–	–	0	1	×	–	×
	TO00	Output	–	–	0	0	0	–	×
	INTP2	Input	PIOR0 = 0	–	0	1	×	–	×
P14	P14	Input	–	–	0	1	×	–	×
		Output	–	–	0	0	0/1	–	TO01 = 0
	ANI20	Input	–	–	1	1	×	–	×
	TI01	Input	–	–	0	1	×	–	×
	TO01	Output	–	–	0	0	0	–	×
	INTP3	Input	PIOR0 = 0	–	0	1	×	×	×

Note Provided only in the R5F102 products.

Table 4-7. Setting Examples of Registers and Output Latches When Using Alternate Function (20-, 24-Pin Products) (3/5)

<R>

Pin Name	Used Function		ADPC	ADM2	PMxx	Pxx
	Function Name	I/O				
P20	P20	Input	ADPC = 01H	×	1	×
		Output	ADPC = 01H	×	0	0/1
	ANI0	Analog input	ADPC = 00H/02H to 04H	00x0xx0x, 10x0xx0x	1	×
	AV_{REFP}	Reference voltage	ADPC = 00H/02H to 04H	01x0xx0x	1	×
P21		Input	ADPC = 01H/02H	×	1	×
		Output	ADPC = 01H/02H	×	0	0/1
	ANI1	Analog input	ADPC = 00H/03H/04H	xx00xx0x	1	×
	AV_{REFM}	Reference voltage	ADPC = 00H/03H/04H	xx10xx0x	1	×
P22	P22	Input	ADPC = 01H to 03H	×	1	×
		Output	ADPC = 01H to 03H	×	0	0/1
	ANI2	Analog input	ADPC = 00H/04H	×	1	×
P23	P23	Input	ADPC = 01H to 04H	×	1	×
		Output	ADPC = 01H to 04H	×	0	0/1
	AN3	Analog input	ADPC = 00H	×	1	×

Downloaded from Arrow.com.

Table 4-7. Setting Examples of Registers and Output Latches When Using Alternate Function (20-, 24-Pin Products (4/5)

<R>

Pin Name	Used Function		PIORx	POMxx	PMCxx	PMxx	Pxx	Alternate Function Output	
	Function Name	I/O						SAU	Other than SAU
P40	P40	Input	–	–	–	1	×	–	–
		Output	–	–	–	0	0/1	–	–
	KR0	Input	–	–	–	1	×	–	–
P41	P41	Input	–	×	0	1	×	×	×
		Output	–	0	0	0	0/1	SO01/SDA01 = 1	TO02 = 0
		N-ch open drain output	–	1	0	0	0/1		
	ANI22	Input	–	×	1	1	×	×	
	SO01[Note 2]	Output	PIOR3 = 0[Note 1]	0/1	0	0	1	×	TO02 = 0
	SDA01[Note 2]	I/O	PIOR3 = 0[Note 1]	1	0	0	1	×	TO02 = 0
	TI02	Input	PIOR2 = 0	×	0	1	×	×	×
	TO02	Output	–	0	0	0	0	SO01/SDA01 = 1	×
	INTP1	Input	–	×	0	1	×	×	×
P42	P42	Input	–	–	0	1	×	×	×
		Output	–	–	0	0	0/1	SCK01/SCL01 = 1	TO03 = 0
	ANI21	Input	–	–	1	1	×	×	×
	SCK01[Note 2]	Input	PIOR3 = 0[Note 1]	–	0	1	×	×	×
		Output	PIOR3 = 0[Note 1]	–	0	0	1	×	TO03 = 0
	SCL01[Note 2]	Output	PIOR3 = 0[Note 1]	–	0	0	1	×	TO03 = 0
	TI03	Input	PIOR2 = 0	–	0	1	×	×	×
	TO03	Output	–	–	0	0	0	SCK01/SCL01 = 1	×
P60	P60	Input	–	–	–	1	×	×	–
		N-ch open drain output (6-V tolerance)	–	–	–	0	0/1	SCLA0/(TxD0) = 1	–
	KR4	Input	–	–	–	1	×	×	–
	SCLA0	I/O	–	–	–	0	0	×	–
	(TxD0)	Output	PIOR1 = 1	–	–	0	1	×	–
P61	P61	Input	–	–	–	1	×	×	–
		N-ch open drain output (6-V tolerance)	–	–	–	0	0/1	SDAA0 = 1	–
	KR5	Input	–	–	–	1	×	×	–
	SDAA0	I/O	–	–	–	0	0	×	–
	(RxD0)	Input	PIOR1 = 1	–	–	1	×	×	–

Notes 1. Provided only in 24-pin products.

2. Provided only in the R5F102 products.

Downloaded from Arrow.com.

Table 4-7. Setting Examples of Registers and Output Latches When Using Alternate Function (20-, 24-Pin Products (5/5)

<R>

Pin Name	Used Function		PIORx	Pxx	Others
	Function Name	I/O			
P121	P121	Input	–	×	EXCLK, OSCSEL = 00/10/11
	KR3	Input	–	×	EXCLK, OSCSEL = 00/10/11
	(TI03)	Input	PIOR2 = 1	×	EXCLK, OSCSEL = 00/10/11
	(INTP3)	Input	PIOR0 = 1	×	EXCLK, OSCSEL = 00/10/11
	X1	–	–	×	EXCLK, OSCSEL = 01
P122	P122	Input	–	×	EXCLK, OSCSEL = 00/10
	KR2	Input	–	×	EXCLK, OSCSEL = 00/10
	(TI02)	Input	PIOR2 = 1	×	EXCLK, OSCSEL = 00/10
	(INTP2)	Input	PIOR0 = 1	×	EXCLK, OSCSEL = 00/10
	X2	–	–	×	EXCLK, OSCSEL = 01
	EXCLK	Input	–	×	EXCLK, OSCSEL = 11
P125	P125	Input	–	×	PORTSELB = 0
	KR1	Input	–	×	PORTSELB = 0
	SI01 Note 2	Input	PIOR3 = 0 Note 1	×	PORTSELB = 0
	$\overline{\text{RESET}}$	Input	–	×	PORTSELB = 1
P137	P137	Input	–	×	–
	INTP0	Input	–	×	–

Notes 1. Provided only in 24-pin products.

2. Provided only in the R5F102 products.

Downloaded from Arrow.com.

Table 4-8. Setting Examples of Registers and Output Latches When Using Alternate Function (30-Pin Products) (1/6)

<R>

Pin Name	Used Function		PIORx	POMxx	PMCxx	PMxx	Pxx	Alternate Function Output	
	Function Name	I/O						SAU	Other than SAU
P00	P00	Input	–	×	0	1	×	×	–
		Output	–	0	0	0	0/1	TxD1 = 1	–
		N-ch open drain output	–	1	0	0	0/1		
	ANI17	Input	–	×	1	1	×	×	–
	TI00	Input	–	×	0	1	×	×	–
	TxD1[Note]	Output	–	0/1	0	0	1	×	–
P01	P01	Input	–	–	0	1	×	–	×
		Output	–	–	0	0	0/1	–	TO00 = 0
	ANI16	Input	–	–	1	1	×	–	×
	TO00	Output	–	–	0	0	0	–	×
	RxD1[Note]	Input	–	–	0	1	×	–	×
P10	P10	Input	–	×	–	1	×	×	×
		Output	–	0	–	0	0/1	SCK00/SCL00 = 1	(TO07) = 0
		N-ch open drain output	–	1	–	0	0/1		
	SCK00	Input	PIOR1 = 0	×	–	1	×	×	×
		Output	PIOR1 = 0	0/1	–	0	1	×	(TO07) = 0
	SCL00[Note]	Output	PIOR1 = 0	0/1	–	0	1	×	(TO07) = 0
	(TI07)	Input	PIOR0 = 1	×	–	1	×	×	×
	(TO07)	Output	PIOR0 = 1	0	–	0	0	SCK00/SCL00 = 1	×
P11	P11	Input	–	×	–	1	×	×	×
		Output	–	0	–	0	0/1	SDA00 = 1	(TO06) = 0
		N-ch open drain output	–	1	–	0	0/1		
	SI00	Input	PIOR1 = 0	×	–	1	×	×	×
	RxD0	Input	PIOR1 = 0	×	–	1	×	×	×
	SDA00[Note]	I/O	PIOR1 = 0	1	–	0	1	×	(TO06) = 0
	(TI06)	Input	PIOR0 = 1	×	–	1	×	×	×
	(TO06)	Output	PIOR0 = 1	0	–	0	0	SDA00 = 1	×
P12	P12	Input	–	×	–	1	×	×	×
		Output	–	0	–	0	0/1	SO00/TxD0 = 1	(TO05) = 0
		N-ch open drain output	–	1	–	0	0/1		
	SO00	Output	PIOR1 = 0	0/1	–	0	1	×	(TO05) = 0
	TxD0	Output	PIOR1 = 0	0/1	–	0	1	×	(TO05) = 0
	(TI05)	Input	PIOR0 = 1	×	–	1	×	×	×
	(TO05)	Output	PIOR0 = 1	0	–	0	0	SO00/TxD0 = 1	×
P13	P13	Input	–	×	–	1	×	×	×
		Output	–	0	–	0	0/1	TxD2/SO20 = 1	(SDAA0)/(TO04) = 0
		N-ch open drain output	–	1	–	0	0/1		
	TxD2[Note]	Output	PIOR1 = 0	0/1	–	0	1	×	(SDAA0)/(TO04) = 0
	SO20[Note]	Output	PIOR1 = 0	0/1	–	0	1	×	(SDAA0)/(TO04) = 0
	(SDAA0)	I/O	PIOR2 = 1	1	–	0	0	TxD2/SO20 = 1	(TO04) = 0
	(TI04)	Input	PIOR0 = 1	×	–	1	×	×	×
	(TO04)	Output	PIOR0 = 1	0	–	0	0	TxD2/SO20 = 1	(SDAA0) = 0

Note Provided only in the R5F102 products.

Downloaded from Arrow.com.

Table 4-8. Setting Examples of Registers and Output Latches When Using Alternate Function (30-Pin Products) (2/6)

<R>

Pin Name	Used Function		PIORx	POMxx	PMCxx	PMxx	Pxx	Alternate Function Output	
	Function Name	I/O						SAU	Other than SAU
P14	P14	Input	–	×	–	1	×		
		Output	–	0	–	0	0/1	SDA20 = 1	(SCLA0)/(TO03) = 0
		N-ch open drain output	–	1	–	0	0/1		
	RxD2Note	Input	PIOR1 = 0	×	–	1	×	×	×
	SI20Note	Input	PIOR1 = 0	×	–	1	×	×	×
	SDA20Note	I/O	PIOR1 = 0	1	–	0	1	×	(SCLA0)/(TO03) = 0
	(SCLA0)	I/O	PIOR2 = 1	1	–	0	0	SDA20 = 1	(TO03) = 0
	(TI03)	Input	PIOR0 = 1	×	–	1	×	×	×
	(TO03)	Output	PIOR0 = 1	0	–	0	0	SDA20 = 1	(SCLA0) = 0
P15	P15	Input	–	×	–	1	×	×	×
		Output	–	0	–	0	0/1	SCK20/SCL20 = 1	PCLBUZ1/(TO02) = 0
		N-ch open drain output	–	1	–	0	0/1		
	PCLBUZ1	Output	–	0	–	0	0	SCK20/SCL20 = 1	(TO02) = 0
	SCK20Note	Input	PIOR1 = 0	×	–	1	×	×	×
		Output	PIOR1 = 0	0/1	–	0	1	×	PCLBUZ1/(TO02) = 0
	SCL20Note	Output	PIOR1 = 0	0/1	–	0	1	×	PCLBUZ1/(TO02) = 0
	(TI02)	Input	PIOR0 = 1	×	–	1	×	×	×
	(TO02)	Output	PIOR0 = 1	0	–	0	0	SCK20/SCL20 = 1	PCLBUZ1 = 0
P16	P16	Input	–	–	–	1	×	–	×
		Output	–	–	–	0	0/1	–	TO01 = 0
	TI01	Input	–	–	–	1	×	–	×
	TO01	Output	–	–	–	0	0	–	×
	INTP5	Input	–	–	–	1	×	–	×
	(RxD0)	Input	PIOR1 = 1	–	–	1	×	–	×
P17	P15	Input	–	×	–	1	×	×	×
		Output	–	0	–	0	0/1	(TxD0) = 1	TO02 = 0
		N-ch open drain output	–	1	–	0	0/1		
	TI02	Input	PIOR0 = 0	×	–	1	×	×	×
	TO02	Output	PIOR0 = 0	0	–	0	0	(TxD0) = 1	×
	(TxD0)	Output	PIOR1 = 1	0/1	–	0	1	×	TO02 = 0

Note Provided only in the R5F102 products.

Downloaded from Arrow.com.

Table 4-8. Setting Examples of Registers and Output Latches When Using Alternate Function (30-Pin Products) (3/6)

<R>

Pin Name	Used Function		ADPC	ADM2	PMxx	Pxx
	Function Name	I/O				
P20	P20	Input	ADPC = 01H	×	1	×
		Output	ADPC = 01H	×	0	0/1
	ANI0	Analog input	ADPC = 00H/02H to 04H	00x0xx0x, 10x0xx0x	1	×
	AV_{REFP}	Reference voltage	ADPC = 00H/02H to 04H	01x0xx0x	1	×
P21	P21	Input	ADPC = 01H/02H	×	1	×
		Output	ADPC = 01H/02H	×	0	0/1
	ANI1	Analog input	ADPC = 00H/03H/04H	xx00xx0x	1	×
	AV_{REFM}	Reference voltage	ADPC = 00H/03H/04H	xx10xx0x	1	×
P22	P22	Input	ADPC = 01H to 03H	×	1	×
		Output	ADPC = 01H to 03H	×	0	0/1
	ANI2	Analog input	ADPC = 00H/04H	×	1	×
P23	P23	Input	ADPC = 01H to 04H	×	1	×
		Output	ADPC = 01H to 04H	×	0	0/1
	AN3	Analog input	ADPC = 00H	×	1	×

Table 4-8. Setting Examples of Registers and Output Latches When Using Alternate Function (30-Pin Products) (4/6)

<R>

Pin Name	Used Function		PIORx	POMxx	PMCxx	PMxx	Pxx	Alternate Function Output	
	Function Name	I/O						SAU	Other than SAU
P30	P30	Input	–	–	–	1	×	×	–
		Output	–	–	–	0	0/1	SCK11/SCL11 = 1	–
	INTP3	Input	–	–	–	1	×	×	–
	SCK11[Note]	Input	–	–	–	1	×	×	–
		Output	–	–	–	0	1	×	–
	SCL11[Note]	Output	–	–	–	0	1	×	–
P31	P31	Input	–	–	–	1	×	–	×
		Output	–	–	–	0	0/1	–	TO03/PCLBUZ0 = 0
	TI03	Input	PIOR0 = 0	–	–	1	×	–	×
	TO03	Output	PIOR0 = 0	–	–	0	0	–	PCLBUZ0 = 0
	INTP4	Input	–	–	–	1	×	–	×
	PCLBUZ0	Output	–	–	–	0	0	–	TO03 = 0
P40	P40	Input	–	–	–	1	×	–	–
		Output	–	–	–	0	0/1	–	–
P50	P50	Input	–	x	–	1	×	×	–
		Output	–	0	–	0	0/1	SDA11 = 1	–
		N-ch open drain output	–	1	–	0	0/1		
	INTP1	Input	–	×	–	1	×	×	–
	SI11[Note]	Input	–	×	–	1	×	×	–
	SDA11[Note]	I/O	–	1	–	0	1	×	–
P51	P51	Input	–	–	–	1	×	×	–
		Output	–	–	–	0	0/1	SO11 = 1	–
	INTP2	Input	–	–	–	1	×	×	–
	SO11[Note]	Output	–	–	–	0	1	×	–

Note Provided only in the R5F102 products.

Downloaded from Arrow.com.

Table 4-8. Setting Examples of Registers and Output Latches When Using Alternate Function (30-Pin Products) (5/6)

<R>

Pin Name	Used Function		PIORx	POMxx	PMCxx	PMxx	Pxx	Alternate Function Output	
	Function Name	I/O						SAU	Other than SAU
P60	P60	Input	–	–	–	1	×	×	–
		N-ch open drain output (6-V tolerance)	–	–	–	0	0/1	SCLA0 = 1	–
	SCLA0	I/O	PIOR2 = 0	–	–	0	0	×	–
P61	P61	Input	–	–	–	1	×	×	–
		N-ch open drain output (6-V tolerance)	–	–	–	0	0/1	SDAA0 = 1	–
	SDAA0	I/O	PIOR2 = 0	–	–	0	0	×	–
P120	P120	Input	–	–	0	1	×	–	–
		Output	–	–	0	0	0/1	–	–
	ANI19	Analog input	–	–	1	1	×	–	–
P137	P137	Input	–	–	–	–	×	–	–
	INTP0	Input	–	–	–	–	×	–	–
P147	P147	Input	–	–	0	1	×	–	–
		Output	–	–	0	0	0/1	–	–
	ANI18	Analog input	–	–	1	1	×	–	–

Table 4-8. Setting Examples of Registers and Output Latches When Using Alternate Function (30-Pin Products) (6/6)

<R>

Pin Name	Used Function		PIORx	Pxx	Others
	Function Name	I/O			
P121	P121	Input	–	×	EXCLK, OSCSEL = 00/10/11
	X1	–	–	×	EXCLK, OSCSEL = 01
P122	P122	Input	–	×	EXCLK, OSCSEL = 00/10
	X2	–	–	×	EXCLK, OSCSEL = 01
	EXCLK	Input	–	×	EXCLK, OSCSEL = 11

Downloaded from Arrow.com.

4.6 Cautions When Using Port Function

4.6.1 Cautions on 1-Bit Manipulation Instruction for Port Register n (Pn)

When a 1-bit manipulation instruction is executed on a port that provides both input and output functions, the output latch value of an input port that is not subject to manipulation may be written in addition to the targeted bit.

Therefore, it is recommended to rewrite the output latch when switching a port from input mode to output mode.

Example When P00 is an output port, P01 to P03 are input ports (all pin statuses are high level), and the output latch value of port 0 is 00H, if the output of output port P00 is changed from low level to high level via a 1-bit manipulation instruction, the output latch value of port 0 is 0FH.

Explanation: The targets of writing to and reading from the Pn register of a port whose PMmn bit is 1 are the output latch and pin status, respectively.

A 1-bit manipulation instruction is executed in the following order in the RL78/G12.

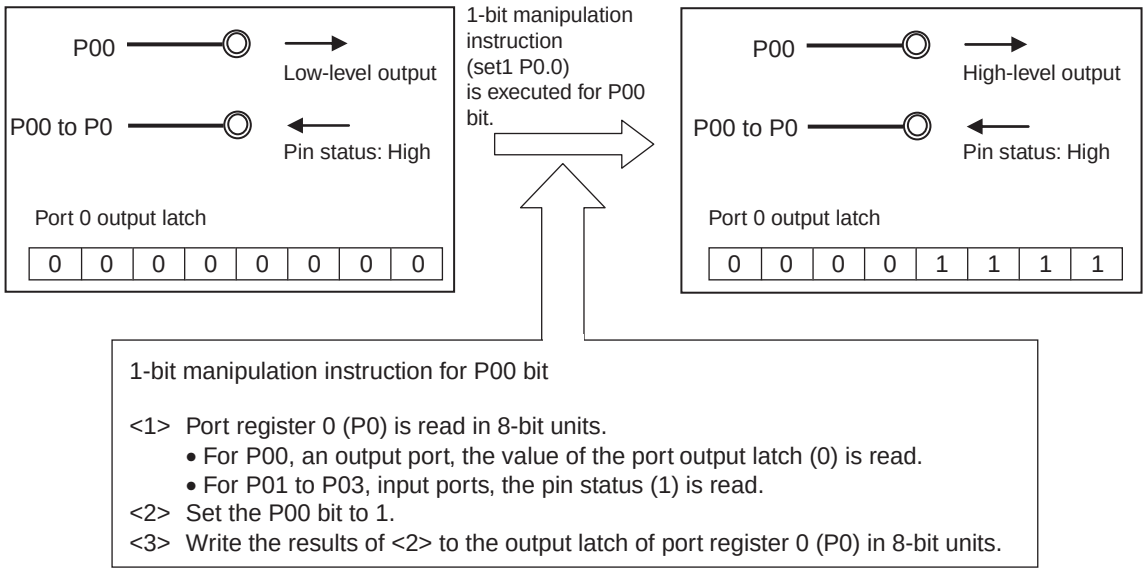
<1> The Pn register is read in 8-bit units.
<2> The targeted one bit is manipulated.
<3> The Pn register is written in 8-bit units.

In step <1>, the output latch value (0) of P00, which is an output port, is read, while the pin statuses of P01 to P03, which are input ports, are read. If the pin statuses of P01 to P03 are high level at this time, the read value is 0EH.

The value is changed to 0FH by the manipulation in <2>.

0FH is written to the output latch by the manipulation in <3>.

Figure 4-10. Bit Manipulation Instruction (P00)



Downloaded from Arrow.com.

4.6.2 Notes on specifying the pin settings

For an output pin to which multiple alternate functions are assigned, the output of the unused alternate function must be set to its initial state so as to prevent conflicting outputs. This also applies to the functions assigned by using the peripheral I/O redirection register (PIOR). For details about the alternate output function, see **4.5 Register Settings When Using Alternate Function**.

No specific setting is required for input pins because the output of their alternate functions is disabled (the buffer output is Hi-Z).

Disabling the unused functions, including blocks that are only used for input or do not have I/O, is recommended to lower power consumption.

Downloaded from Arrow.com.

CHAPTER 5 CLOCK GENERATOR

5.1 Functions of Clock Generator

The clock generator generates the clock to be supplied to the CPU and peripheral hardware.
The following three kinds of system clocks and clock oscillators are selectable.

(1) Main system clock

<1> X1 oscillator

This circuit oscillates a clock of f_X = 1 to 20 MHz by connecting a resonator to X1 and X2.
Oscillation can be stopped by executing the STOP instruction or setting of the MSTOP bit (bit 7 of the clock operation status control register (CSC)).

<2> High-speed on-chip oscillator

<R> The frequency at which to oscillate can be selected from among f_{IH} = 24, 16, 12, 8, 6, 4, 3, 2, or 1 MHz (TYP.) by using the option byte (000C2H). After a reset release, the CPU always starts operating with this high-speed on-chip oscillator clock. Oscillation can be stopped by executing the STOP instruction or setting the HIOSTOP bit (bit 0 of the CSC register).
The frequency specified by using an option byte can be changed by using the high-speed on-chip oscillator frequency select register (HOCODIV). For details about the frequency, see **Figure 5-9 Format of High-speed On-chip Oscillator Frequency Select Register (HOCODIV)**.
The frequencies that can be specified for the high-speed on-chip oscillator by using the option byte and the high-speed on-chip oscillator frequency select register (HOCODIV) are shown below.

Power Supply Voltage	Oscillation Frequency (MHz)								
	1	2	3	4	6	8	12	16	24
2.7 V ≤ V_{DD} ≤ 5.5 V	√	√	√	√	√	√	√	√	√
<R> 2.4 V ≤ V_{DD} ≤ 5.5 V	√	√	√	√	√	√	√	√	–
<R> 1.8 V ≤ V_{DD} ≤ 5.5 V	√	√	√	√	√	√	–	–	–

Remark √: Can be specified, –: Cannot be specified

An external main system clock (f_{EX} = 1 to 20 MHz) can also be supplied from the EXCLK/X2/P122 pin. An external main system clock input can be disabled by executing the STOP instruction or setting of the MSTOP bit.
As the main system clock, a high-speed system clock (X1 clock or external main system clock) or high-speed on-chip oscillator clock can be selected by setting of the MCM0 bit (bit 4 of the system clock control register (CKC)).

(2) Low-speed on-chip oscillator clock (Low-speed on-chip oscillator)

This circuit oscillates a clock of f_{IL} = 15 kHz (TYP.).
The low speed on-chip oscillator clock cannot be used as the CPU clock.
Only the following peripheral hardware runs on the low speed on-chip oscillator clock.

- Watchdog timer
- 12-bit Interval timer

This clock operates when bit 4 (WDTON) of the option byte (000C0H), bit 4 (WUTMMCK0) of the operation speed mode control register (OSMC), or both are set to 1.
However, when WDTON = 1, WUTMMCK0 = 0, and bit 0 (WDSTBYON) of the option byte (000C0H) is 0, oscillation of the LOCO stops if the HALT or STOP instruction is executed.

Downloaded from Arrow.com.

Remark f_X: X1 clock oscillation frequency
f_{IH}: High-speed on-chip oscillator clock frequency
f_{EX}: External main system clock frequency
f_{IL}: Low speed on-chip oscillator clock frequency

5.2 Configuration of Clock Generator

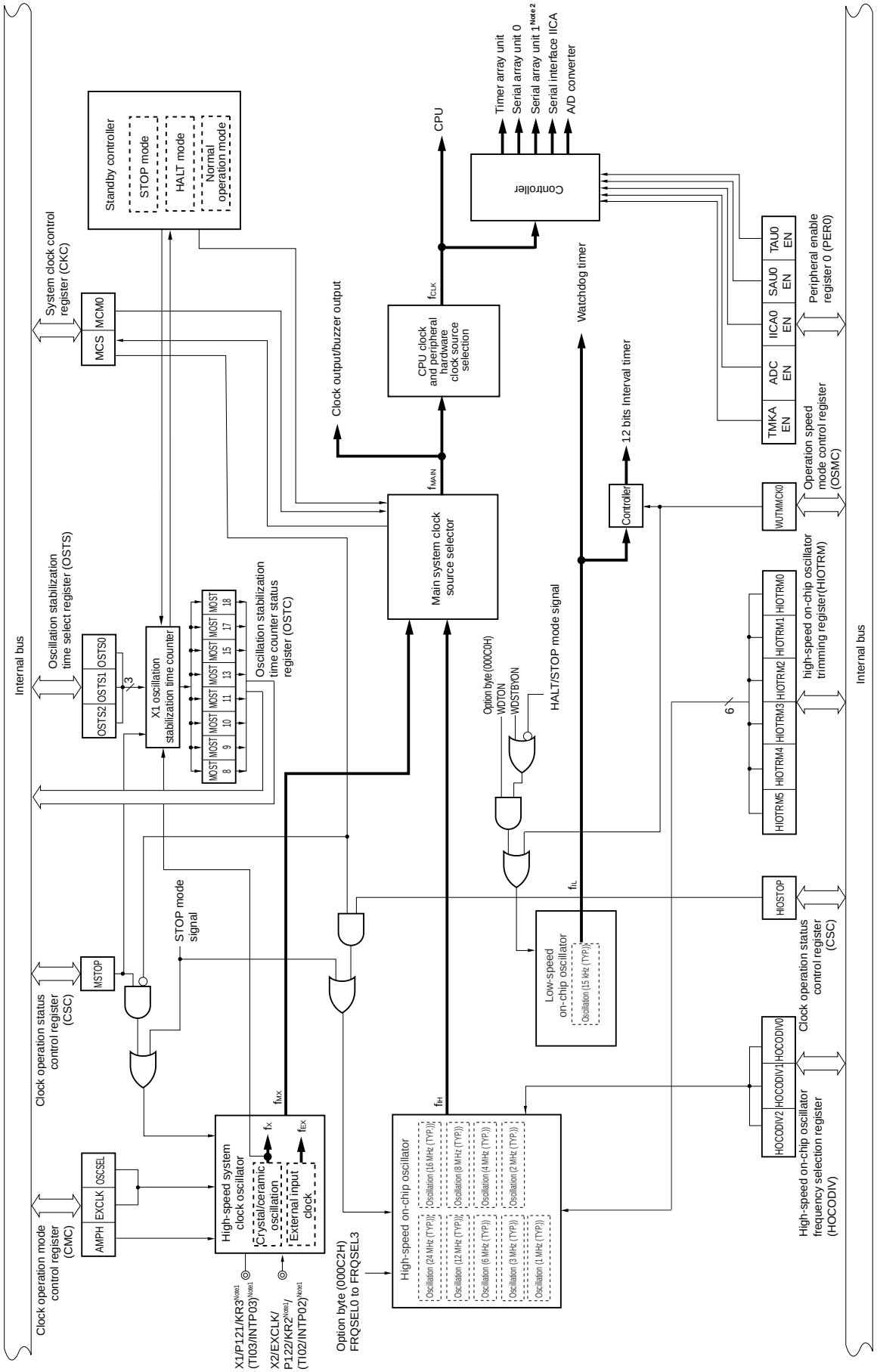
The clock generator includes the following hardware.

Table 5-1. Configuration of Clock Generator

Item	Configuration
Control registers	Clock operation mode control register (CMC) System clock control register (CKC) Clock operation status control register (CSC) Oscillation stabilization time counter status register (OSTC) Oscillation stabilization time select register (OSTS) Peripheral enable register 0 (PER0) Operation speed mode control register (OSMC) High-speed on-chip oscillator frequency selection register (HOCODIV) High-speed on-chip oscillator trimming register (HIOTRM)
Oscillators	X1 oscillator High-speed on-chip oscillator Low-speed on-chip oscillator

Downloaded from Arrow.com.

Figure 5-1. Block Diagram of Clock Generator



<R>

Notes 1. 20-, 24-pin products only.

2. 30-pin products only.

Downloaded from Arrow.com.

Remark f_X: X1 clock oscillation frequency
f_{IH}: High-speed on-chip oscillator clock frequency
f_{EX}: External main system clock frequency
f_{MX}: High-speed system clock frequency
f_{MAIN}: Main system clock frequency
f_{CLK}: CPU/peripheral hardware clock frequency
f_{IL}: Low-speed on-chip oscillator clock frequency

5.3 Registers Controlling Clock Generator

The following nine registers are used to control the clock generator.

- Clock operation mode control register (CMC)
- System clock control register (CKC)
- Clock operation status control register (CSC)
- Oscillation stabilization time counter status register (OSTC)
- Oscillation stabilization time select register (OSTS)
- Peripheral enable register 0 (PER0)
- Operation speed mode control register (OSMC)
- High-speed on-chip oscillator frequency selection register (HOCODIV)
- High-speed on-chip oscillator trimming register (HIOTRM)

Caution Which registers and bits are included depends on the product. Be sure to set registers and bits that are not mounted in a product to their initial values.

Downloaded from Arrow.com.

5.3.1 Clock operation mode control register (CMC)

This register is used to set the operation mode of the X1/P121 and X2/EXCLK/P122 pins, and to select a gain of the oscillator.

The CMC register can be written only once by an 8-bit memory manipulation instruction after reset release. This
<R> register can be read by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 5-2. Format of Clock Operation Mode Control Register (CMC)

Address: FFFA0H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CMC	EXCLK	OSCSEL	0	0	0	0	0	AMPH

EXCLK	OSCSEL	High-speed system clock pin operation mode	X1/P121/KR3 pin	X2/EXCLK/P122/KR2 pin
0	0	Input port mode	Input port	
0	1	X1 oscillation mode	Crystal/ceramic resonator connection	
1	0	Input port mode	Input port	
1	1	External clock input mode	Input port	External clock input

AMPH	Control of X1 clock oscillation frequency
0	1 MHz ≤ f_X ≤ 10 MHz
1	10 MHz < f_X ≤ 20 MHz

Cautions 1. The CMC register can be written only once after reset release, by an 8-bit memory manipulation instruction. When using the CMC register with its initial value (00H), be sure to set the register to 00H after a reset ends in order to prevent malfunction due to a program loop. Such a malfunction becomes unrecoverable when a value other than 00H is mistakenly written.

2. After reset release, set the CMC register before X1 oscillation is started as set by the clock operation status control register (CSC).

<R> **3. Be sure to set the AMPH bit to 1 if the X1 clock oscillation frequency exceeds 10 MHz.**

<R> **4. Specify the setting for the AMPH bit while f_{IH} is selected as f_{CLK} after a reset ends (before f_{CLK} is switched to f_{MX}).**

5. Although the maximum system clock frequency is 24 MHz, the maximum frequency of the X1 oscillator is 20 MHz.

Remark f_X: X1 clock frequency

Downloaded from Arrow.com.

5.3.2 System clock control register (CKC)

This register is used to select a CPU/peripheral hardware clock and a main system clock.

The CKC register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to 00H.

Figure 5-3. Format of System Clock Control Register (CKC)

Address: FFFA4H After reset: 00H R/W [Note]

Symbol	7	6	<5>	<4>	3	2	1	0
CKC	0	0	MCS	MCM0	0	0	0	0

MCS	Status of Main system clock (f_{MAIN})
0	High-speed on-chip oscillator clock (f_{IH})
1	High-speed system clock (f_{MX})

MCM0	Main system clock (f_{MAIN}) operation control
0	Selects the high-speed on-chip oscillator clock (f_{IH}) as the main system clock (f_{MAIN})
1	Selects the high-speed system clock (f_{MX}) as the main system clock (f_{MAIN})

Note Bit 5 is read-only.

Remark f_{IH}: High-speed on-chip oscillator clock frequency
f_{MX}: High-speed system clock frequency
f_{MAIN}: Main system clock frequency

<R> **Caution Be sure to set bits 7, 6, and 3 to 0 to 0.**

Downloaded from Arrow.com.

5.3.3 Clock operation status control register (CSC)

This register is used to control the operations of the high-speed system clock and high-speed on-chip oscillator clock, (except the low-speed on-chip oscillator clock).

The CSC register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to C0H.

Figure 5-4. Format of Clock Operation Status Control Register (CSC)

Address: FFFA1H After reset: C0H R/W

Symbol	<7>	6	5	4	3	2	1	<0>
CSC	MSTOP	1	0	0	0	0	0	HIOSTOP

MSTOP	High-speed system clock operation control		
	X1 oscillation mode	External clock input mode	Input port mode
0	X1 oscillator operating	External clock from EXCLK pin is valid	Input port
1	X1 oscillator stopped	External clock from EXCLK pin is invalid	

HIOSTOP	High-speed on-chip oscillator clock operation control
0	High-speed on-chip oscillator operating
1	High-speed on-chip oscillator stopped

Cautions

1. **After reset release, set the clock operation mode control register (CMC) before setting the CSC register.**
2. **Set the oscillation stabilization time select register (OSTS) before setting the MSTOP bit to 0 after releasing reset. Note that if the OSTS register is being used with its default settings, the OSTS register is not required to be set here.**
3. **To start X1 oscillation as set by the MSTOP bit, check the oscillation stabilization time of the X1 clock by using the oscillation stabilization time counter status register (OSTC).**
4. **Do not stop the clock selected for the CPU peripheral hardware clock (f_{CLK}) with the CSC register.**
5. **The setting of the flags of the register to stop clock oscillation (invalidate the external clock input) and the condition before clock oscillation is to be stopped are as Table 5-2.**

<R>

Table 5-2. Stopping Clock Method

Clock	Condition Before Stopping Clock (Invalidating External Clock Input)	Setting of CSC Register Flags
X1 clock External main system clock	CPU and peripheral hardware clocks operate with a high-speed on-chip oscillator clock. (MCS = 0)	MSTOP = 1
High-speed on-chip oscillator clock	CPU and peripheral hardware clocks operate with a high-speed system clock.(MCS = 1)	HIOSTOP = 1

Downloaded from Arrow.com.

5.3.4 Oscillation stabilization time counter status register (OSTC)

This is the register that indicates the count status of the X1 clock oscillation stabilization time counter.

The X1 clock oscillation stabilization time can be checked in the following case:

- If the X1 clock starts oscillation while the high-speed on-chip oscillator clock is being used as the CPU clock.
- If the STOP mode is entered and then released while the high-speed on-chip oscillator clock is being used as the CPU clock with the X1 clock oscillating.

The OSTC register can be read by a 1-bit or 8-bit memory manipulation instruction.

When reset signal is generated, the STOP instruction and MSTOP (bit 7 of clock operation status control register (CSC)) = 1 clear the OSTC register to 00H.

Remark The oscillation stabilization time counter starts counting in the following cases.

- When oscillation of the X1 clock starts (EXCLK, OSCSEL = 0, 1 → MSTOP = 0)
- When the STOP mode is released

Downloaded from Arrow.com.

Figure 5-5. Format of Oscillation Stabilization Time Counter Status Register (OSTC)

Address: FFFA2H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
OSTC	MOST 8	MOST 9	MOST 10	MOST 11	MOST 13	MOST 15	MOST 17	MOST 18

<R>

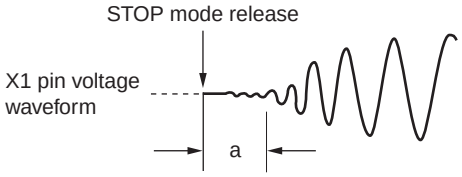
MOST 8	MOST 9	MOST 10	MOST 11	MOST 13	MOST 15	MOST 17	MOST 18	Oscillation stabilization time status		
									f_X = 10 MHz	f_X = 20 MHz
0	0	0	0	0	0	0	0	$2^8/f_X$ max.	25.6 μs max.	12.8 μs max.
1	0	0	0	0	0	0	0	$2^8/f_X$ min.	25.6 μs min.	12.8 μs min.
1	1	0	0	0	0	0	0	$2^9/f_X$ min.	51.2 μs min.	25.6 μs min.
1	1	1	0	0	0	0	0	$2^{10}/f_X$ min.	102 μs min.	51.2 μs min.
1	1	1	1	0	0	0	0	$2^{11}/f_X$ min.	204 μs min.	102 μs min.
1	1	1	1	1	0	0	0	$2^{13}/f_X$ min.	819 μs min.	409 μs min.
1	1	1	1	1	1	0	0	$2^{15}/f_X$ min.	3.27 ms min.	1.63 ms min.
1	1	1	1	1	1	1	0	$2^{17}/f_X$ min.	13.1 ms min.	6.55 ms min.
1	1	1	1	1	1	1	1	$2^{18}/f_X$ min.	26.2 ms min.	13.1 ms min.

Cautions 1. After the above time has elapsed, the bits are set to 1 in order from the MOST8 bit and remain 1.

2. The oscillation stabilization time counter counts up to the oscillation stabilization time set by the oscillation stabilization time select register (OSTS).
In the following cases, set the oscillation stabilization time of the OSTS register to the value greater than the count value which is to be checked by the OSTC register.

- **If the X1 clock starts oscillation while the high-speed on-chip oscillator clock is being used as the CPU clock.**
- **If the STOP mode is entered and then released while the high-speed on-chip oscillator clock is being used as the CPU clock with the X1 clock oscillating.**
(Note, therefore, that only the status up to the oscillation stabilization time set by the OSTS register is set to the OSTC register after the STOP mode is released.)

3. The X1 clock oscillation stabilization wait time does not include the time until clock oscillation starts (“a” below).



Remark f_X: X1 clock oscillation frequency

Downloaded from Arrow.com.

5.3.5 Oscillation stabilization time select register (OSTS)

<R> This register is used to select the X1 clock oscillation stabilization wait time.

<R> When the X1 clock is made to oscillate by clearing the MSTOP bit to start the X1 oscillation circuit operating, actual operation is automatically delayed for the time set in the OSTS register.

<R> When switching the CPU clock from the high-speed on-chip oscillator clock to the X1 clock, and when using the high-speed on-chip oscillator clock for switching the X1 clock from the oscillating state to STOP mode, use the oscillation stabilization time counter status register (OSTC) to confirm that the desired oscillation stabilization time has elapsed after release from the STOP mode. That is, use the OSTC register to check that the oscillation stabilization time corresponding to its setting has been reached.

The OSTS register can be set by an 8-bit memory manipulation instruction.

Reset signal generation sets the OSTS register to 07H.

Figure 5-6. Format of Oscillation Stabilization Time Select Register (OSTS)

Address: FFFA3H After reset: 07H R/W

Symbol	7	6	5	4	3	2	1	0
OSTS	0	0	0	0	0	OSTS2	OSTS1	OSTS0

<R>

OSTS2	OSTS1	OSTS0	Oscillation stabilization time selection		
				f_X = 10 MHz	f_X = 20 MHz
0	0	0	$2^8/f_X$	25.6 μs	12.8 μs
0	0	1	$2^9/f_X$	51.2 μs	25.6 μs
0	1	0	$2^{10}/f_X$	102 μs	51.2 μs
0	1	1	$2^{11}/f_X$	204 μs	102 μs
1	0	0	$2^{13}/f_X$	819 μs	409 μs
1	0	1	$2^{15}/f_X$	3.27 ms	1.63 ms
1	1	0	$2^{17}/f_X$	13.1 ms	6.55 ms
1	1	1	$2^{18}/f_X$	26.2 ms	13.1 ms

Cautions 1. To set the STOP mode when the X1 clock is used as the CPU clock, set the OSTS register before executing the STOP instruction.

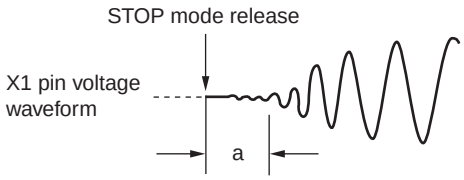
2. The oscillation stabilization time counter counts up to the oscillation stabilization time set by the OSTS register.

In the following cases, set the oscillation stabilization time of the OSTS register to the value greater than the count value which is to be checked by the OSTC register after the oscillation starts.

- **If the X1 clock starts oscillation while the high-speed on-chip oscillator clock is being used as the CPU clock.**
- **If the STOP mode is entered and then released while the high-speed on-chip oscillator clock is being used as the CPU clock with the X1 clock oscillating. (Note, therefore, that only the status up to the oscillation stabilization time set by the OSTS register is set to the OSTC register after the STOP mode is released.)**

Downloaded from Arrow.com.

3. The X1 clock oscillation stabilization wait time does not include the time until clock oscillation starts (“a” below).



Remark fx: X1 clock oscillation frequency

Downloaded from Arrow.com.

5.3.6 Peripheral enable register 0 (PER0)

This register is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to the hardware that is not used is also stopped so as to decrease the power consumption and noise.

To use the peripheral functions below, which are controlled by this register, set (1) the bit corresponding to each function before specifying the initial settings of the peripheral functions.

- 12-bit Interval timer
- A/D converter
- Serial interface IICA
- Serial array unit 1
- Serial array unit 0
- Timer array unit 0

The PER0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

<R> **Figure 5-7. Format of Peripheral Enable Register 0 (PER0) (1/2)**

Address: F00F0H After reset: 00H R/W

Symbol	<7>	6	<5>	<4>	<3>	<2>	1	<0>
PER0	TMKAEN	0	ADCEN	IICA0EN	SAU1EN Note	SAU0EN	0	TAU0EN

TMKAE	Control of 12-bit interval timer clock supply
0	Stops clock supply. • SFR used by the 12-bit interval timer cannot be written. • The 12-bit interval timer is in the reset status.
1	Enables clock supply. • SFR used by the 12-bit interval timer can be read and written.

ADCEN	Control of A/D converter clock supply
0	Stops clock supply. • SFR used by the A/D converter cannot be written. • The A/D converter is in the reset status.
1	Enables clock supply. • SFR used by the A/D converter can be read and written.

IICA0EN	Control of serial interface IICA clock supply
0	Stops clock supply. • SFR used by the serial interface IICA cannot be written. • The serial interface IICA is in the reset status.
1	Enables clock supply. • SFR used by the serial interface IICA can be read and written.

<R> **Note** 30-pin products only.

<R> **Caution Be sure to set the following bits to 0.**
20, 24-pin products: bits 1, 3, 6
30-pin products: bits 1, 6

Downloaded from Arrow.com.

<R> **Figure 5-7. Format of Peripheral Enable Register 0 (PER0) (2/2)**

Address: F00F0H After reset: 00H R/W

Symbol	<7>	6	<5>	<4>	<3>	<2>	1	<0>
PER0	TMKAEN	0	ADCEN	IICA0EN	SAU1EN Note	SAU0EN	0	TAU0EN

SAU1EN	Control of serial array unit 1 clock supply
0	Stops clock supply. • SFR used by the serial array unit 1 cannot be written. • The serial array unit 1 is in the reset status.
1	Enables clock supply. • SFR used by the serial array unit 1 can be read and written.

SAU0EN	Control of serial array unit 0 clock supply
0	Stops clock supply. • SFR used by the serial array unit 0 cannot be written. • The serial array unit 0 is in the reset status.
1	Enables clock supply. • SFR used by the serial array unit 0 can be read and written.

TAU0EN	Control of timer array unit clock supply
0	Stops clock supply. • SFR used by timer array unit cannot be written. • Timer array unit is in the reset status.
1	Enables clock supply. • SFR used by timer array unit can be read and written.

<R> **Note** 30-pin products only.

<R> **Caution Be sure to set the following bits to 0.**
20, 24-pin products: bits 1, 3, 6
30-pin products: bits 1, 6

5.3.7 Operation speed mode control register (OSMC)

<R> The OSMC register is used to control supply of the operation clock for the 12-bit interval timer.

When operating the 12-bit interval timer, set WUTMMCK0 = 1 beforehand and do not set WUTMMCK0 = 0 until the timer is stopped.

The OSMC register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 5-8. Format of Operation Speed Mode Control Register (OSMC)

Address: F00F3H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
OSMC	0	0	0	WUTMMCK0	0	0	0	0

WUTMMCK0	Supply of operation clock for 12-bit interval timer
0	Stops Clock supply
1	Low-speed on-chip oscillator clock (f_{IL}) supply

Downloaded from Arrow.com.

5.3.8 High-speed on-chip oscillator frequency selection register (HOCODIV)

This register is used to change the frequency of the high-speed on-chip oscillator set with the option byte (000C2H). The available frequency varies depending on the value of the FRQSEL3 bit of the option byte (000C2H).

The HOCODIV register can be set by an 8-bit memory manipulation instruction.

<R> Reset signal generation clears this register to the value set by FRQSEL2 to FRQSEL0 of the option byte (000C2H).

Figure 5-9 Format of High-Speed On-Chip Oscillator Frequency Selection Register (HOCODIV)

<R> Address: F00A8H After reset: the value set by FRQSEL2 to FRQSEL0 of the option byte (000C2H) R/W

Symbol	7	6	5	4	3	2	1	0
HOCODIV	0	0	0	0	0	HOCODIV 2	HOCODIV 1	HOCODIV 0

HOCODIV 2	HOCODIV 1	HOCODIV 0	High-speed on-chip oscillator clock frequency selection	
			FRQSEL3 bit is 0	FRQSEL3 bit is 1
0	0	0	24 MHz	Setting prohibited
0	0	1	12 MHz	16 MHz
0	1	0	6 MHz	8 MHz
0	1	1	3 MHz	4 MHz
1	0	0	Setting prohibited	2 MHz
1	0	1	Setting prohibited	1 MHz
Other than above			Setting prohibited	

Cautions 1. Set the HOCODIV register within the operable voltage range of the flash operation mode set in the option byte (000C2H) before and after the frequency change.

Option byte (000C2H) value		Flash operation mode	Operating frequency range	Operating voltage range
CMODE1	CMODE0			
1	0	LS (low-speed main) mode	1 MHz to 8 MHz	1.8 V to 5.5 V
1	1	HS (high-speed main) mode	1 MHz to 16 MHz	2.4 V to 5.5 V
			1 MHz to 24 MHz	2.7 V to 5.5 V

2. Set the HOCODIV register with the high-speed on-chip oscillator clock (f_{IH}) selected as the CPU/peripheral hardware clock (f_{CLK}).

3. After settings are changed with the HOCODIV register, the frequency is switched after the following transition time has elapsed.

- **Operation for up to three clocks at the pre-change frequency**
- **CPU/peripheral hardware clock wait at the post-change frequency for up to three clocks**

Downloaded from Arrow.com.

5.3.9 High-speed on-chip oscillator trimming register (HIOTRM)

This register is used to adjust the accuracy of the high-speed on-chip oscillator.

With self-measurement of the high-speed on-chip oscillator frequency via a timer using high-accuracy external clock input (timer array unit), and so on, the accuracy can be adjusted.

The HIOTRM register can be set by an 8-bit memory manipulation instruction.

Caution The frequency will vary if the temperature and V_{DD} pin voltage change after accuracy adjustment. When the temperature and V_{DD} voltage change, accuracy adjustment must be executed regularly or before the frequency accuracy is required.

Figure 5-10. Format of High-Speed On-chip Oscillator Trimming Register (HIOTRM)

<R> Address: F00A0H After reset: undefined Note R/W

Symbol	7	6	5	4	3	2	1	0
HIOTRM	0	0	HIOTRM5	HIOTRM4	HIOTRM3	HIOTRM2	HIOTRM1	HIOTRM0

HIOTRM5	HIOTRM4	HIOTRM3	HIOTRM2	HIOTRM1	HIOTRM0	High-speed on-chip oscillator
0	0	0	0	0	0	Minimum speed
0	0	0	0	0	1	▲
0	0	0	0	1	0	
0	0	0	0	1	1	
0	0	0	1	0	0	
•	•	•	•	•	•	
1	1	1	1	1	0	▼
1	1	1	1	1	1	Maximum speed

<R> **Note** The value after reset is the value adjusted at shipment.

<R> **Remarks 1.** The HIOTRM register can be used to adjust the high-speed on-chip oscillator clock to an accuracy within about 0.05%.

<R> **2.** For the usage example of the HIOTRM register, see the application note for RL78 MCU series High-speed On-chip Oscillator (HOCO) Clock Frequency Correction (R01AN0464).

Downloaded from Arrow.com.

5.4 System Clock Oscillator

5.4.1 X1 oscillator

The X1 oscillator oscillates with a crystal resonator or ceramic resonator (1 to 20 MHz) connected to the X1 and X2 pins.

An external clock can also be input. In this case, input the clock signal to the EXCLK pin.

To use the X1 oscillator, set bits 7 and 6 (EXCLK, OSCSEL) of the clock operation mode control register (CMC) as follows.

- Crystal or ceramic oscillation: EXCLK, OSCSEL = 0, 1
- External clock input: EXCLK, OSCSEL = 1, 1

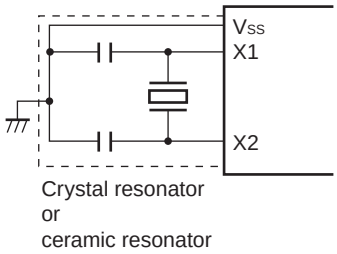
When the X1 oscillator is not used, set the input port mode (EXCLK, OSCSEL = 0, 0).

When the pins are not used as input port pins, either, see **2.3 Connection of Unused Pins**.

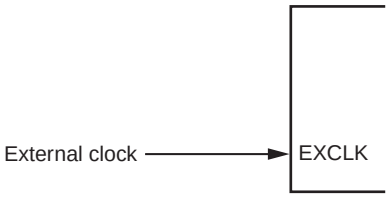
Figure 5-11 shows an example of the external circuit of the X1 oscillator.

Figure 5-11. Example of External Circuit of X1 Oscillator

(a) Crystal or ceramic oscillation



(b) External clock



Caution When using the X1 oscillator, wire as follows in the area enclosed by the broken lines in the Figure 5-11 to avoid an adverse effect from wiring capacitance.

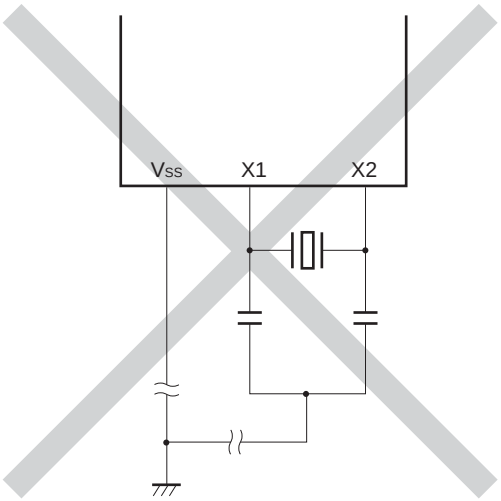
- **Keep the wiring length as short as possible.**
- **Do not cross the wiring with the other signal lines. Do not route the wiring near a signal line through which a high fluctuating current flows.**
- **Always make the ground point of the oscillator capacitor the same potential as Vss. Do not ground the capacitor to a ground pattern through which a high current flows.**
- **Do not fetch signals from the oscillator.**

Downloaded from Arrow.com.

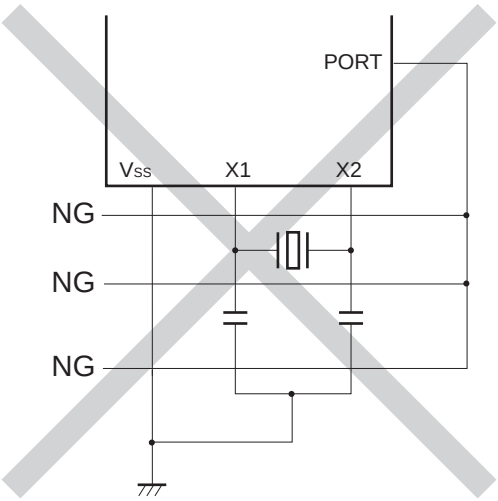
Figure 5-12 shows examples of incorrect resonator connection.

Figure 5-12. Examples of Incorrect Resonator Connection (1/2)

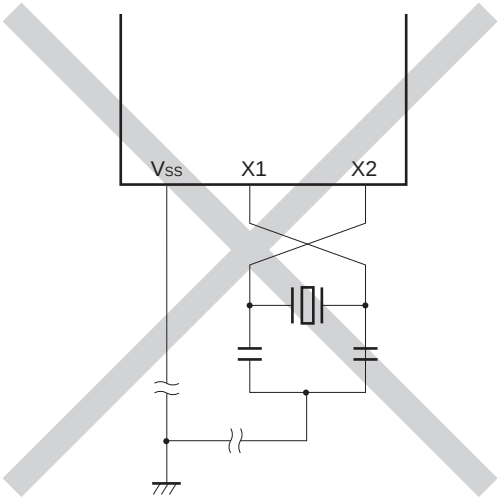
(a) Too long wiring



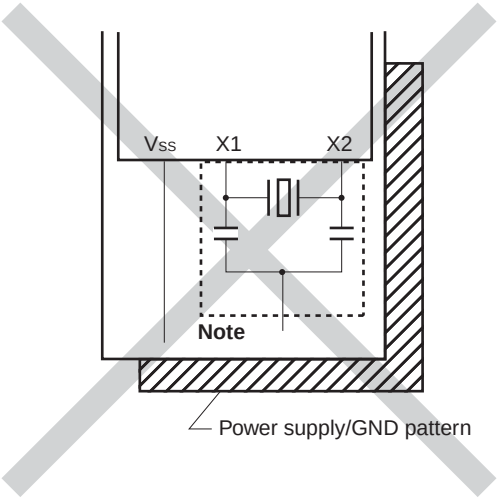
(b) Crossed signal line



(c) The X1 and X2 signal line wires cross.



(d) A power supply/GND pattern exists under the X1 and X2 wires.

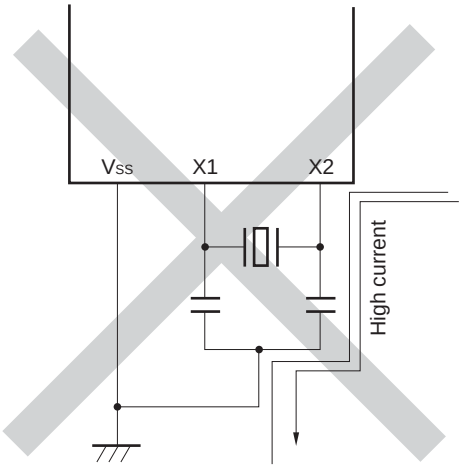


Note Do not place a power supply/GND pattern under the wiring section (section indicated by a broken line in the figure) of the X1 and X2 pins and the resonators in a multi-layer board or double-sided board.
Do not configure a layout that will cause capacitance elements and affect the oscillation characteristics.

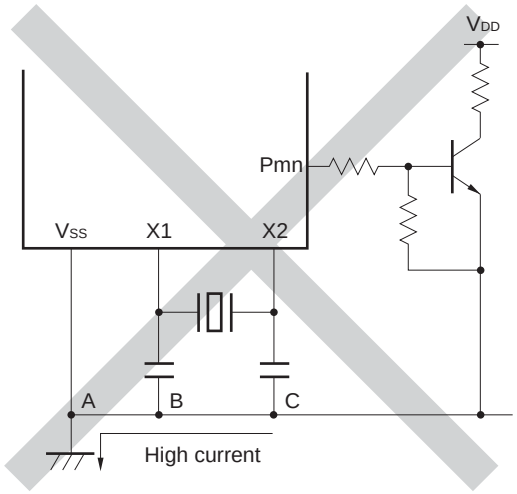
Downloaded from Arrow.com.

Figure 5-12. Examples of Incorrect Resonator Connection (2/2)

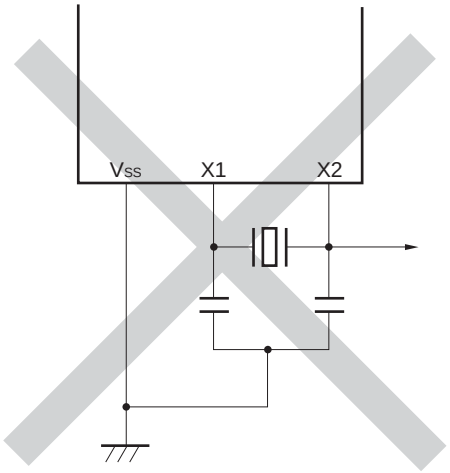
(e) Wiring near high alternating current



(f) Current flowing through ground line of oscillator (potential at points A, B, and C fluctuates)



(g) Signals are fetched



Downloaded from Arrow.com.

5.4.2 High-speed on-chip oscillator

<R> The high-speed on-chip oscillator is incorporated in the RL78/G12. The frequency can be selected from among 24, 16, 12, 8, 6, 4, 3, 2, or 1 MHz by using the option byte (000C2H). Oscillation can be controlled by bit 0 (HIOSTOP) of the clock operation status control register (CSC). The high-speed on-chip oscillator automatically starts oscillating after reset release.

5.4.3 Low-speed on-chip oscillator

The low-speed on-chip oscillator is incorporated in the RL78/G12.

The low-speed on-chip oscillator clock is used only as the watchdog timer, and 12-bit interval timer clock. The low-speed on-chip oscillator clock cannot be used as the CPU clock.

This clock operates when bit 4 (WDTON) of the option byte (000C0H), bit 4 (WUTMMCK0) of the operation speed mode control register (OSMC), or both are set to 1.

Unless the watchdog timer is stopped and WUTMMCK0 is a value other than zero, oscillation of the low-speed on-chip
<R> oscillator continues. Note that only when the watchdog timer is operating and the WUTMMCK0 bit is 0, oscillation of the low-speed on-chip oscillator will stop while the WDSTBYON bit is 0 and operation is in the HALT, STOP, or SNOOZE mode. While the watchdog timer operates, the low-speed on-chip oscillator clock does not stop even if the program freezes.

5.5 Clock Generator Operation

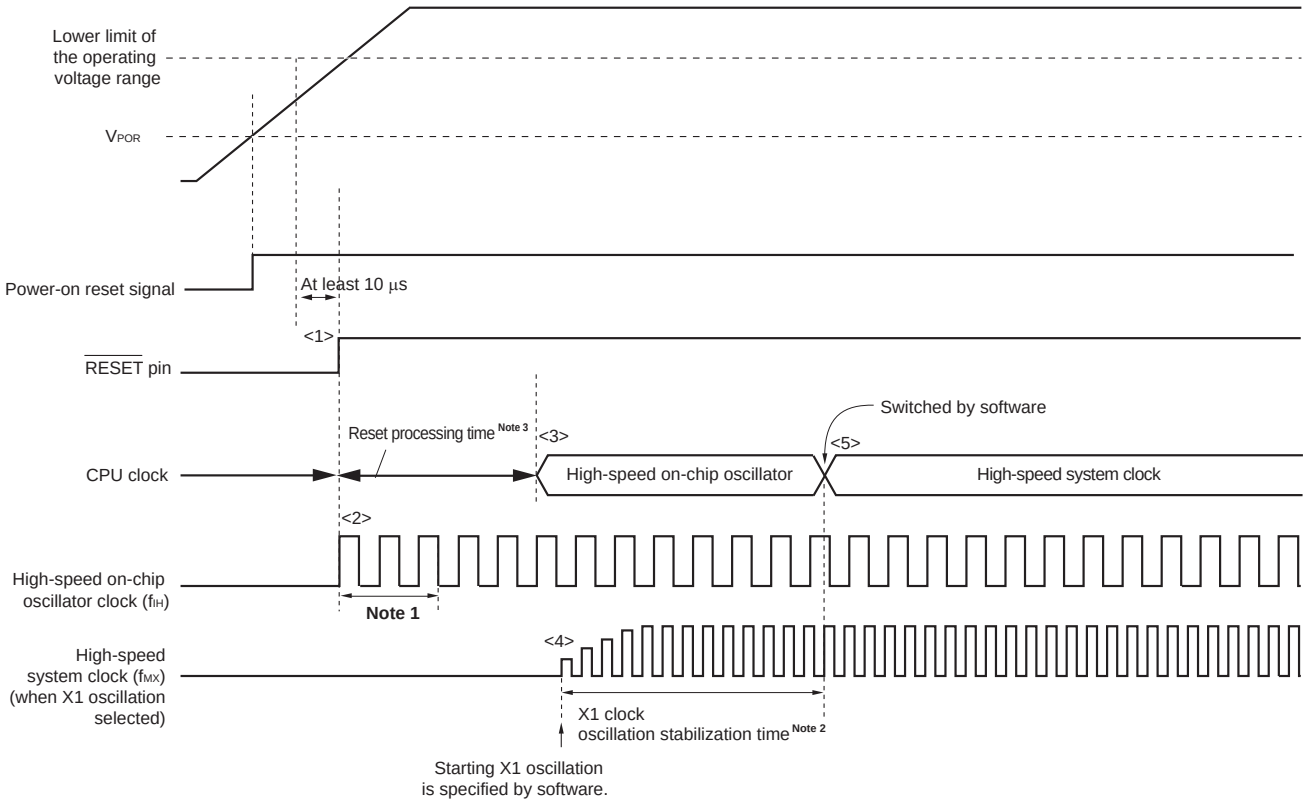
The clock generator generates the following clocks and controls the operation modes of the CPU, such as standby mode (see **Figure 5-1**).

- Main system clock f_{MAIN}
  - High-speed system clock f_{MX}
    X1 clock f_X
    External main system clock f_{EX}
  - High-speed on-chip oscillator clock f_{IH}
- Low-speed on-chip oscillator clock f_{IL}
- CPU/peripheral hardware clock f_{CLK}

The CPU starts operation when the high-speed on-chip oscillator starts outputting after a reset release in the RL78/G12. When the power supply voltage is turned on, the clock generator operation is shown in Figure 5-13.

Downloaded from Arrow.com.

<R> **Figure 5-13. Clock Generator Operation When Power Supply Voltage Is Turned On**



<1> When the power is turned on, an internal reset signal is generated by the power-on-reset (POR) circuit. Note that the reset state is maintained after a reset by the voltage detection circuit or an external reset until the voltage reaches the range of operating voltage described in **28.4 AC Characteristics** and **29.4 AC Characteristics** (the above figure is an example when the external reset is in use).

<2> When the reset is released, the high-speed on-chip oscillator automatically starts oscillation.

<3> The CPU starts operation on the high-speed on-chip oscillator clock after waiting for the voltage to stabilize and a reset processing have been performed after reset release.

<4> Set the start of oscillation of the X1 clock via software (see **5.6.2 Example of setting X1 oscillation clock**).

<5> When switching the CPU clock to the X1 clock, wait for the clock oscillation to stabilize, and then switch the clock via software.

Notes 1. The internal reset processing time includes the oscillation accuracy stabilization time of the high-speed on-chip oscillator clock.

2. When releasing a reset, confirm the oscillation stabilization time for the X1 clock using the oscillation stabilization time counter status register (OSTC).

3. For the reset processing time, see **CHAPTER 19 POWER-ON-RESET CIRCUIT**.

Caution It is not necessary to wait for the oscillation stabilization time when an external clock input from the EXCLK pin is used.

Downloaded from Arrow.com.

5.6 Controlling Clock

5.6.1 Example of setting high-speed on-chip oscillator

<R> After a reset release, the CPU/peripheral hardware clock (f_{CLK}) always starts operating with the high-speed on-chip oscillator clock. The frequency of the high-speed on-chip oscillator can be selected from 24, 16, 12, 8, 6, 4, 3, 2, and 1 MHz by using FRQSEL0 to FRQSEL3 of the option byte (000C2H). The frequency can also be changed by the high-speed on-chip oscillator frequency select register (HOCODIV).

[Option byte setting]

Address: 000C2H

Option byte (000C2H)

7	6	5	4	3	2	1	0
CMODE1	CMODE0			FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0
0/1	0/1	1	0	0/1	0/1	0/1	0/1

CMODE1	CMODE0	Setting of flash operation mode
1	0	LS (low speed main) mode V_{DD} = 1.8 V to 5.5 V @ 1 MHz to 8 MHz
1	1	HS (high speed main) mode V_{DD} = 2.4 V to 5.5 V @ 1 MHz to 16 MHz V_{DD} = 2.7 V to 5.5 V @ 1 MHz to 24 MHz
Other than above		Setting prohibited

FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0	Frequency of the high-speed on-chip oscillator
0	0	0	0	24 MHz
1	0	0	1	16 MHz
0	0	0	1	12 MHz
1	0	1	0	8 MHz
0	0	1	0	6 MHz
1	0	1	1	4 MHz
0	0	1	1	3 MHz
1	1	0	0	2 MHz
1	1	0	1	1 MHz
Other than above				Setting prohibited

<R>
<R>
<R>

[High-speed on-chip oscillator frequency selection register (HOCODIV) setting]

Address: F00A8H

HOCODIV

7	6	5	4	3	2	1	0
0	0	0	0	0	HOCODIV 2	HOCODIV 1	HOCODIV 0

HOCODIV 2	HOCODIV 1	HOCODIV 0	High-speed on-chip oscillator clock frequency selection	
			FRQSEL3 bit is 0	FRQSEL3 bit is 1
0	0	0	24 MHz	Setting prohibited
0	0	1	12 MHz	16 MHz
0	1	0	6 MHz	8 MHz
0	1	1	3 MHz	4 MHz
1	0	0	Setting prohibited	2 MHz
1	0	1	Setting prohibited	1 MHz
Other than above			Setting prohibited	

Downloaded from Arrow.com.

5.6.2 Example of setting X1 oscillation clock

After a reset release, the CPU/peripheral hardware clock (f_{CLK}) always starts operating with the high-speed on-chip oscillator clock. To subsequently change the clock to the X1 oscillation clock, set the oscillator and start oscillation by using the oscillation stabilization time select register (OSTS) and clock operation mode control register (CMC) and clock operation status control register (CSC) and wait for oscillation to stabilize by using the oscillation stabilization time select register (OSTC). After the oscillation stabilizes, set the X1 oscillation clock to f_{CLK} by using the system clock control register (CKC).

[Register settings] Set the register in the order of <1> to <5> below.

<1> Set (1) the OSCSEL bit of the CMC register, except for the cases f_X > 10 MHz, in such cases set (1) the AMPH bit, to operate the X1 oscillator.

<R>

	7	6	5	4	3	2	1	0
CMC	EXCLK	OSCSEL						AMPH
	0	1	0	0	0	0	0	0/1

<2> Using the OSTS register, select the oscillation stabilization time of the X1 oscillator at releasing of the STOP mode.

<R> Example: Setting values when a wait of at least 102 μs is set based on a 10 MHz resonator.

	7	6	5	4	3	2	1	0
OSTS						OSTS2	OSTS1	OSTS0
	0	0	0	0	0	0	1	0

<3> Clear (0) the MSTOP bit of the CSC register to start oscillating the X1 oscillator.

	7	6	5	4	3	2	1	0
CSC	MSTOP							HIOSTOP
	0	0	0	0	0	0	0	0

<4> Use the OSTC register to wait for oscillation of the X1 oscillator to stabilize.

<R> Example: Wait until the bits reach the following values when a wait of at least 102 μs is set based on a 10 MHz resonator.

	7	6	5	4	3	2	1	0
OSTC	MOST8	MOST9	MOST10	MOST11	MOST13	MOST15	MOST17	MOST18
	1	1	1	0	0	0	0	0

<5> Use the MCM0 bit of the CKC register to specify the X1 oscillation clock as the CPU/peripheral hardware clock.

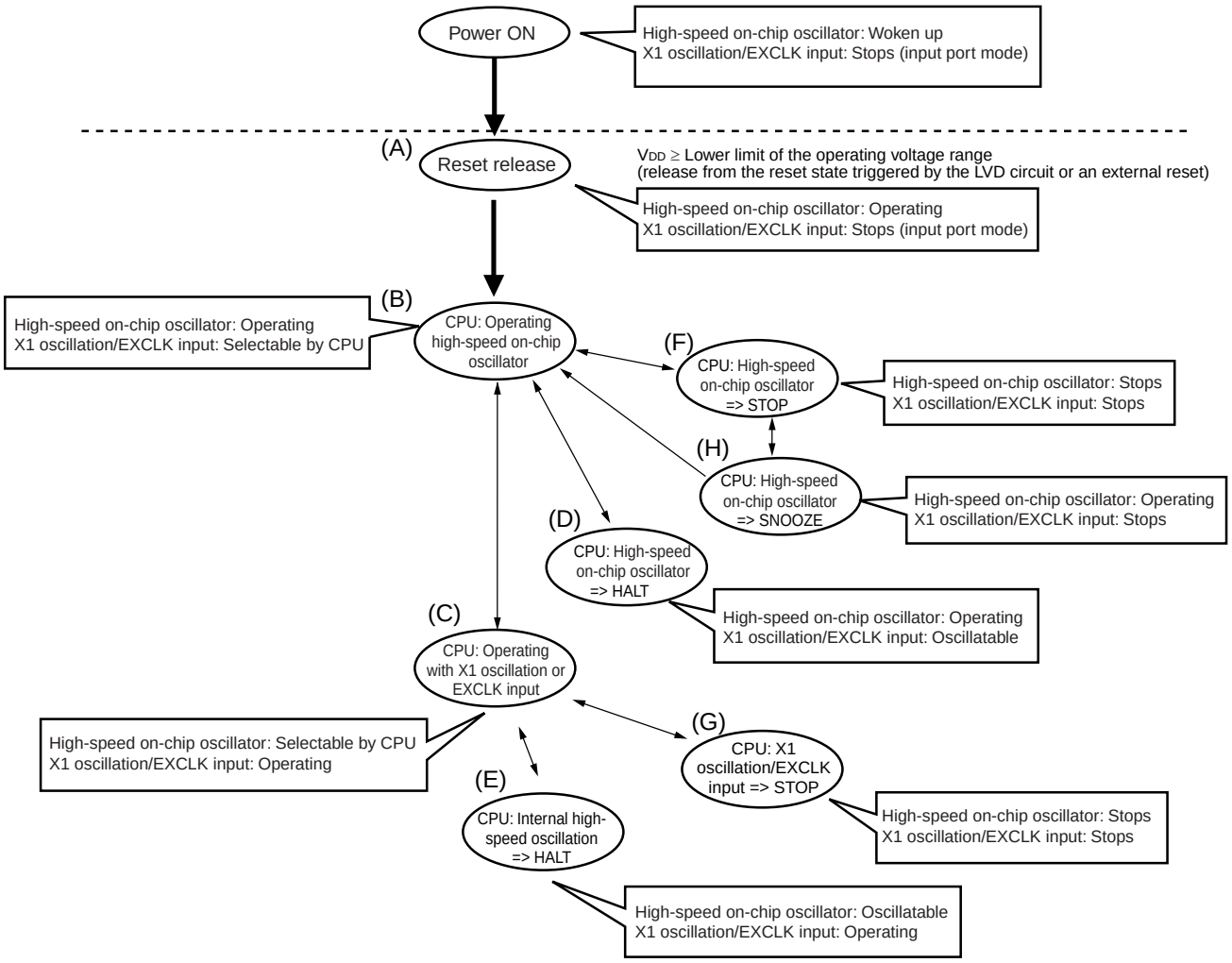
	7	6	5	4	3	2	1	0
CKC			MCS	MCM0				
	0	0	0	1	0	0	0	0

Downloaded from Arrow.com.

5.6.3 CPU clock status transition diagram

Figure 5-14 shows the CPU clock status transition diagram of this product.

<R> **Figure 5-14. CPU Clock Status Transition Diagram**



Downloaded from Arrow.com.

Table 5-3 shows transition of the CPU clock and examples of setting the SFR registers.

Table 5-3. CPU Clock Transition and SFR Register Setting Examples (1/3)

(1) CPU operating with high-speed on-chip oscillator clock (B) after reset release (A)

Status Transition	SFR Register Setting
(A) → (B)	SFR registers do not have to be set (default status after reset release).

(2) CPU operating with high-speed system clock (C) after reset release (A)

(The CPU operates with the high-speed on-chip oscillator clock immediately after a reset release (B).)

(Setting sequence of SFR registers) ⟶

Setting Flag of SFR Register / Status Transition	CMC Register Note1			OSTS Register	CSC Register	OSTC Register	CKC Register
	EXCLK	OSCSEL	AMPH		MSTOP		MCM0
(A) → (B) → (C) (X1 clock: 1 MHz ≤ f_X ≤ 10 MHz)	0	1	0	**Note 2**	0	Must be checked	1
(A) → (B) → (C) (X1 clock: 10 MHz < f_X ≤ 20 MHz)	0	1	1	**Note 2**	0	Must be checked	1
(A) → (B) → (C) (external main clock)	1	1	×	**Note 2**	0	Must not be checked	1

Notes 1. The clock operation mode control register (CMC) can be written only once by an 8-bit memory manipulation instruction after reset release.

2. Set the oscillation stabilization time as follows.

- Desired the oscillation stabilization time counter status register (OSTC) oscillation stabilization time ≤ Oscillation stabilization time set by the oscillation stabilization time select register (OSTS)

Caution Set the clock after the supply voltage has reached the operable voltage of the clock to be set (see CHAPTER 28 or 29 ELECTRICAL SPECIFICATIONS).

Remarks 1. ×: don't care

2. (A) to (H) in Table 5-3 correspond to (A) to (H) in Figure 5-14.

Downloaded from Arrow.com.

Table 5-3. CPU Clock Transition and SFR Register Setting Examples (2/3)

(3) CPU clock changing from high-speed on-chip oscillator clock (B) to high-speed system clock (C)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	CMC Register[Note 1]			OSTS Register	CSC Register	OSTC Register	CKC Register
	EXCLK	OSCSEL	AMPH		MSTOP		MCM0
(B) → (C) (X1 clock: 1 MHz ≤ fx ≤ 10 MHz)	0	1	0	**Note 2**	0	Must be checked	1
(B) → (C) (X1 clock: 10 MHz < fx ≤ 20 MHz)	0	1	1	**Note 2**	0	Must be checked	1
(B) → (C) (external main clock)	1	1	×	**Note 2**	0	Must not be checked	1

Unnecessary if these registers are already set

Unnecessary if the CPU is operating with the high-speed system clock

Notes 1. The clock operation mode control register (CMC) can be written only once by an 8-bit memory manipulation instruction after reset release. This setting is not necessary if it has already been set.

2. Set the oscillation stabilization time as follows.

- Desired the oscillation stabilization time counter status register (OSTC) oscillation stabilization time ≤ Oscillation stabilization time set by the oscillation stabilization time select register (OSTS)

Caution Set the clock after the supply voltage has reached the operable voltage of the clock to be set (see CHAPTER 28 or 29 ELECTRICAL SPECIFICATIONS).

(4) CPU clock changing from high-speed system clock (C) to high-speed on-chip oscillator clock (B)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	CSC Register	Oscillation accuracy stabilization time	CKC Register
	HIOSTOP		MCM0
<R> (C) → (B)	0	18 μs to 65 μs	0

Unnecessary if the CPU is operating with the high-speed on-chip oscillator clock

Remark The oscillation accuracy stabilization time changes according to the temperature conditions and the STOP mode period.

(5) • HALT mode (D) set while CPU is operating with high-speed on-chip oscillator clock (B)
• HALT mode (E) set while CPU is operating with high-speed system clock (C)

Status Transition	Setting
(B) → (D) (C) → (E)	Executing HALT instruction

Remarks 1. ×: don't care

2. (A) to (H) in Table 5-3 correspond to (A) to (H) in Figure 5-14.

Downloaded from Arrow.com.

Table 5-3. CPU Clock Transition and SFR Register Setting Examples (3/3)

(6) • STOP mode (F) set while CPU is operating with high-speed on-chip oscillator clock (B)
• STOP mode (G) set while CPU is operating with high-speed system clock (C)

(Setting sequence) →

Status Transition		Setting		
(B) → (F)		Stopping peripheral functions that are disabled in STOP mode	–	Executing STOP instruction
(C) → (G)	In X1 oscillation		Sets the OSTS register	
	External main system clock		–	

(7) CPU changing from STOP mode (F) to SNOOZE mode (H)

For details about the setting for switching from the STOP mode to the SNOOZE mode, see **10.8 SNOOZE Mode Function**, **11.5.7 SNOOZE mode function** and **11.6.3 SNOOZE mode function**.

Remark (A) to (H) in Table 5-3 correspond to (A) to (H) in Figure 5-14.

5.6.4 Condition before changing CPU clock and processing after changing CPU clock

Condition before changing the CPU clock and processing after changing the CPU clock are shown below.

Table 5-4. Changing CPU Clock

CPU Clock		Condition Before Change	Processing After Change
Before Change	After Change		
High-speed on-chip oscillator clock	X1 clock	Stabilization of X1 oscillation • OSCSEL = 1, EXCLK = 0, MSTOP = 0 • After elapse of oscillation stabilization time	Operating current can be reduced by stopping high-speed on-chip oscillator (HIOSTOP = 1).
	External main system clock	Enabling external clock input from the EXCLK pin • OSCSEL = 1, EXCLK = 1, MSTOP = 0	
X1 clock	High-speed on-chip oscillator clock	Enabling oscillation of high-speed on-chip oscillator • HIOSTOP = 0 • After elapse of oscillation accuracy stabilization time	X1 oscillation can be stopped (MSTOP = 1).
	External main system clock	Transition not possible	–
External main system clock	High-speed on-chip oscillator clock	Oscillation of high-speed on-chip oscillator • HIOSTOP = 0 • After elapse of oscillation accuracy stabilization time	External main system clock input can be disabled (MSTOP = 1).
	X1 clock	Transition not possible	–

Downloaded from Arrow.com.

5.6.5 Time required for switchover of CPU clock and main system clock

The main system clock can be switched between the high-speed on-chip oscillator clock and the high-speed system clock by specifying bit 4 (MCM0) of the system clock control register (CKC).

The actual switchover operation is not performed immediately after rewriting to the CKC register; operation continues on the pre-switchover clock for several clocks (see **Table 5-5**).

Whether the main system clock is operating on the high-speed system clock or high-speed on-chip oscillator clock can be ascertained using bit 5 (MCS) of the CKC register.

When the CPU clock is switched, the peripheral hardware is also switched.

Table 5-5. Maximum Number of Clocks Required for $f_{IH} \leftrightarrow f_{MX}$

Set Value Before Switchover		Set Value After Switchover	
MCM0		MCM0	
		0 ($f_{MAIN} = f_{IH}$)	1 ($f_{MAIN} = f_{MX}$)
0 ($f_{MAIN} = f_{IH}$)	$f_{MX} \geq f_{IH}$		2 clock
	$f_{MX} < f_{IH}$		$2f_{IH}/f_{MX}$ clock
1 ($f_{MAIN} = f_{MX}$)	$f_{MX} \geq f_{IH}$	$2f_{MX}/f_{IH}$ clock	
	$f_{MX} < f_{IH}$	2 clock	

Remarks 1. Number of CPU clocks before switchover.

2. Calculate the number of clocks by rounding to the nearest whole number.

Example When switching the main system clock from the high-speed system clock to the high-speed on-chip oscillator clock (@ oscillation with f_{IH} = 8 MHz selected, f_{MX} = 10 MHz)

$2\ f_{IH}/f_{MX} = 2(10/8) = 2.5 \rightarrow 3$ clocks

5.6.6 Conditions before clock oscillation is stopped

The following lists the register flag settings for stopping the clock oscillation (disabling external clock input) and conditions before the clock oscillation is stopped.

Table 5-6. Conditions Before the Clock Oscillation Is Stopped and Flag Settings

Clock	Conditions Before Clock Oscillation Is Stopped (External Clock Input Disabled)	Flag Settings of SFR
High-speed on-chip oscillator clock	MCS = 1 (The CPU is operating on the high-speed system clock.)	HIOSTOP = 1
X1 clock	MCS = 0 (The CPU is operating on the high-speed on-chip oscillator clock.)	MSTOP = 1
External main system clock		

Downloaded from Arrow.com.

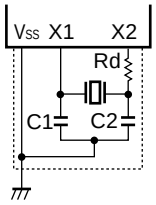
5.7 Resonator and Oscillator Constants

The resonators for which the operation is verified and their oscillator constants are shown below.

Cautions 1. The constants for these oscillator circuits are reference values based on specific environments set up for evaluation by the manufacturers. For actual applications, request evaluation by the manufacturer of the oscillator circuit mounted on a board. Furthermore, if you are switching from a different product to this microcontroller, and whenever you change the board, again request evaluation by the manufacturer of the oscillator circuit mounted on the new board.

2. The oscillation voltage and oscillation frequency only indicate the oscillator characteristic. Use the RL78 microcontroller so that the internal operation conditions are within the specifications of the DC and AC characteristics.

Figure 5-15. External Oscillation Circuit Example (X1 Oscillation)



Downloaded from Arrow.com.

<R> As of March, 2013

Manufacturer	Resonator	Part Number Note 3	SMD/ Lead	Frequency (MHz)	Flash operation mode Note 1	Recommended Circuit Constants Note 2 (reference)			Oscillation Voltage Range (V)	
						C1 (pF)	C2 (pF)	Rd (kΩ)	MIN.	MAX.
Murata Manufacturing Co., Ltd.	Ceramic resonator	CSTCC2M00G56-R0	SMD	2.0	LS	(47)	(47)	0	1.8	5.5
		CSTCR4M00G55-R0	SMD	4.0		(39)	(39)	0		
		CSTLS4M00G53-B0	Lead			(15)	(15)	0		
		CSTCR4M19G55-R0	SMD	4.194		(39)	(39)	0		
		CSTLS4M19G53-B0	Lead			(15)	(15)	0		
		CSTCR4M91G53-R0	SMD	4.915		(15)	(15)	0		
		CSTLS4M91G53-B0	Lead			(15)	(15)	0		
		CSTCR5M00G53-R0	SMD	5.0		(15)	(15)	0		
		CSTLS5M00G53-B0	Lead			(15)	(15)	0		
		CSTCR6M00G53-R0	SMD	6.0		(15)	(15)	0		
		CSTLS6M00G53-B0	Lead			(15)	(15)	0		
		CSTCE8M00G52-R0	SMD	8.0		(10)	(10)	0		
		CSTLS8M00G53-B0	Lead			(15)	(15)	0		
		CSTCE8M38G52-R0	SMD	8.388	HS	(10)	(10)	0	2.4	5.5
		CSTLS8M38G53-B0	Lead			(15)	(15)	0		
		CSTCE10M0G52-R0	SMD	10.0		(10)	(10)	0		
		CSTLS10M0G53-B0	Lead			(15)	(15)	0		
		CSTCE12M0G52-R0	SMD	12.0		(10)	(10)	0		
		CSTCE16M0V53-R0	SMD	16.0		(15)	(15)	0		
		CSTLS16M0X51-B0	Lead			(5)	(5)	0		
		CSTCE20M0V51-R0	SMD	20.0		(5)	(5)	0	2.7	5.5
		CSTLS20M0X51-B0	Lead			(5)	(5)	0		

Notes 1. Set the flash operation mode by using CMODE1 and CMODE0 bits of the option byte (000C2H).

2. Values in parentheses in the C1, C2 columns indicate an internal capacitance.

<R> **3.** Products supporting 105°C operation have different part numbers. For details, contact Murata Manufacturing Co., Ltd. (http://www.murata.com)

Remark Relationship between operation voltage width, operation frequency of CPU and operation mode is as below.

HS (High speed main) mode: 2.7 V ≤ V_{DD} ≤ 5.5 V@1 MHz to 24 MHz

2.4 V ≤ V_{DD} ≤ 5.5 V@1 MHz to 16 MHz

LS (Low speed main) mode: 1.8 V ≤ V_{DD} ≤ 5.5 V@1 MHz to 8 MHz

Downloaded from Arrow.com.

CHAPTER 6 TIMER ARRAY UNIT

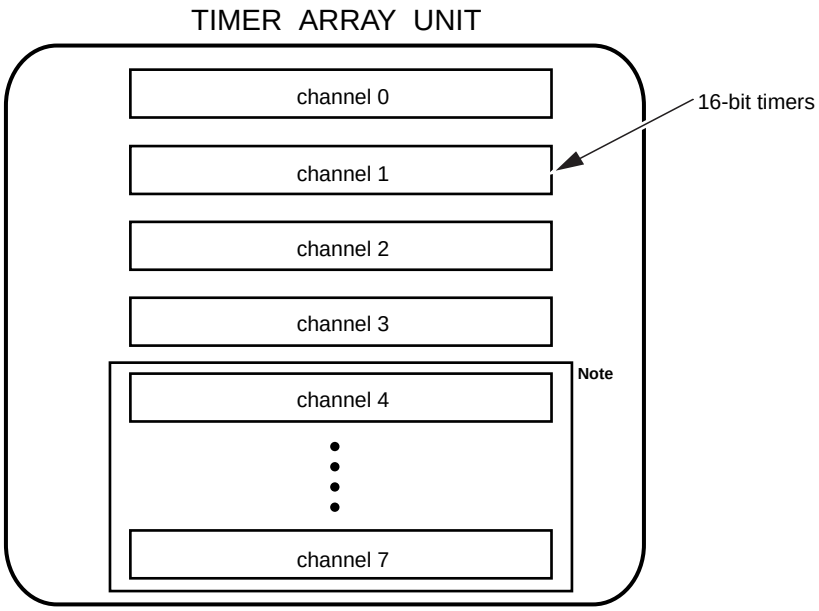
<R> The number of channels of the timer array unit differs depending on the product.

Channels	20-, 24-pin	30-pin
Channel 0	√	√
Channel 1	√	√
Channel 2	√	√
Channel 3	√	√
Channel 4	–	√
Channel 5	–	√
Channel 6	–	√
Channel 7	–	√

Caution The presence or absence of timer I/O pins depends on the product. For details, see Table 6-2 Timer I/O Pins in the Product.

The timer array unit has four/eight 16-bit timers.

Each 16-bit timer is called a channel and can be used as an independent timer. In addition, two or more “channels” can be used to create a high-accuracy timer.



Note Provided only in 30-pin products

Downloaded from Arrow.com.

For details about each function, see the table below.

Independent channel operation function	Simultaneous channel operation function
• Interval timer (→ refer to **6.8.1**) • Square wave output (→ refer to **6.8.1**) • External event counter (→ refer to **6.8.2**) • Divider[Note] (→ refer to **6.8.3**) • Input pulse interval measurement (→ refer to **6.8.4**) • Measurement of high-/low-level width of input signal (→ refer to **6.8.5**) • Delay counter (→ refer to **6.8.6**)	• One-shot pulse output (→ refer to **6.9.1**) • PWM output (→ refer to **6.9.2**) • Multiple PWM output (→ refer to **6.9.3**)

Note Only channel 0 of the 30-pin products.

It is possible to use the 16-bit timer of channels 1 and 3 as two 8-bit timers (higher and lower). The functions that can use channels 1 and 3 as 8-bit timers are as follows:

- Interval timer (higher and lower 8-bit timers)/square wave output (lower 8-bit timer only)
- External event counter (lower 8-bit timer only)
- Delay counter (lower 8-bit timer only)

Downloaded from Arrow.com.

6.1 Functions of Timer Array Unit

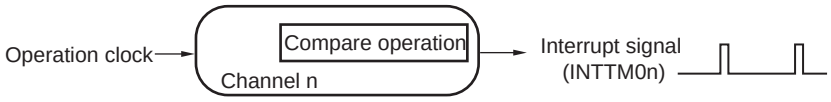
Timer array unit has the following functions.

6.1.1 Independent channel operation function

By operating a channel independently, it can be used for the following purposes without being affected by the operation mode of other channels.

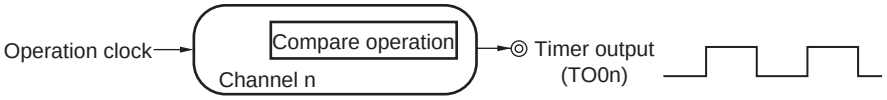
(1) Interval timer

Each timer of a unit can be used as a reference timer that generates an interrupt (INTTM0n) at fixed intervals.



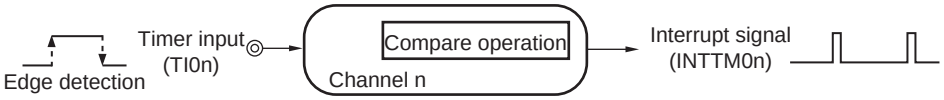
(2) Square wave output

A toggle operation is performed each time INTTM0n interrupt is generated and a square wave with a duty factor of 50% is output from a timer output pin (TO0n).



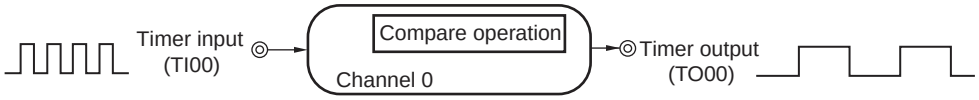
(3) External event counter

Each timer of a unit can be used as an event counter that generates an interrupt when the number of the valid edges of a signal input to the timer input pin (TI0n) has reached a specific value.



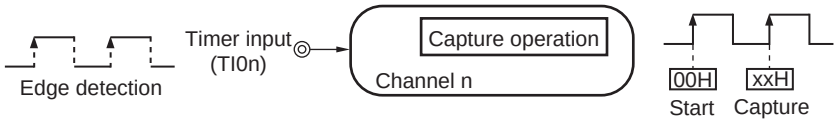
<R> **(4) Divider** [Note]

A clock input from a timer input pin (TI00) is divided and output from an output pin (TO00).



(5) Input pulse interval measurement

Counting is started by the valid edge of a pulse signal input to a timer input pin (TI0n). The count value of the timer is captured at the valid edge of the next pulse. In this way, the interval of the input pulse can be measured.

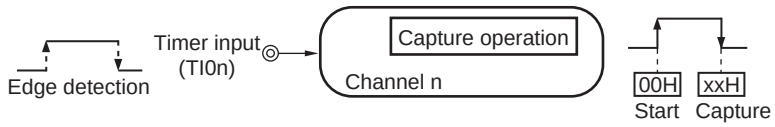


<R> **Note** Only channel 0 of the 30-pin products.

Downloaded from Arrow.com.

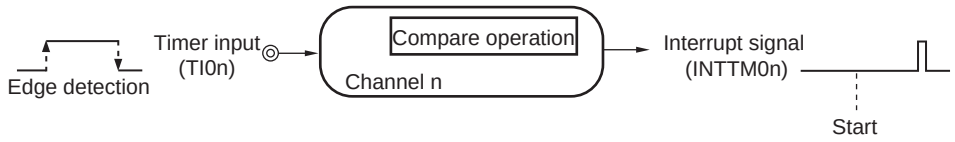
(6) Measurement of high-/low-level width of input signal

Counting is started by a single edge of the signal input to the timer input pin (TI0n), and the count value is captured at the other edge. In this way, the high-level or low-level width of the input signal can be measured.



(7) Delay counter

Counting is started at the valid edge of the signal input to the timer input pin (TI0n), and an interrupt is generated after any delay period.



Remark n: Channel number (n = 0 to 7)

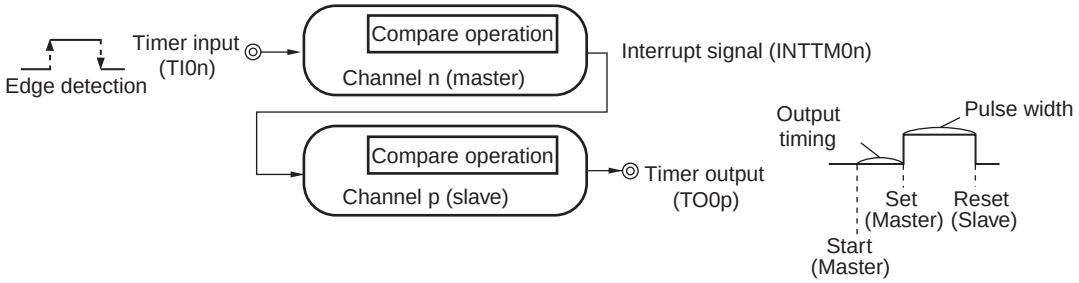
6.1.2 Simultaneous channel operation function

By using the combination of a master channel (a reference timer mainly controlling the cycle) and slave channels (timers operating according to the master channel), channels can be used for the following purposes.

(1) One-shot pulse output

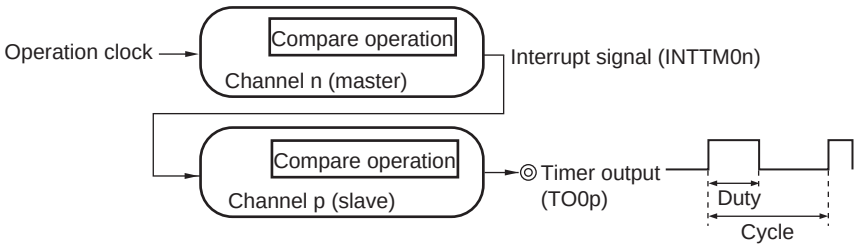
Two channels are used as a set to generate a one-shot pulse with a specified output timing and a specified pulse width.

<R>



(2) PWM (Pulse Width Modulation) output

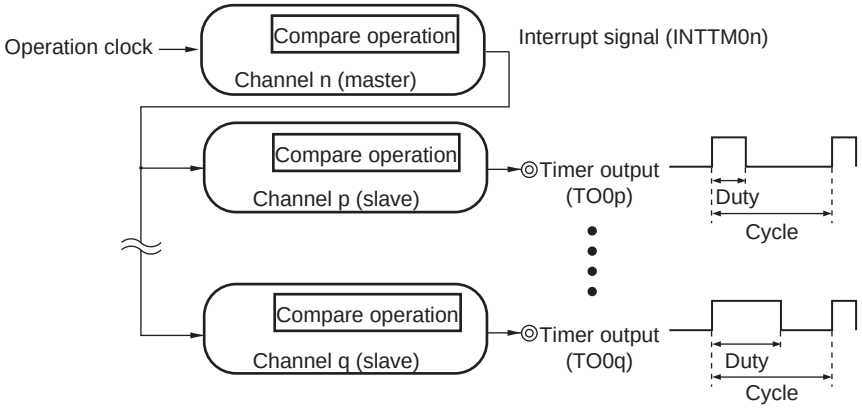
Two channels are used as a set to generate a pulse with a specified period and a specified duty factor.



Downloaded from Arrow.com.

(3) Multiple PWM (Pulse Width Modulation) output

By extending the PWM function and using one master channel and two or more slave channels, up to three types of PWM signals that have a specific period and a specified duty factor can be generated.



Caution For details about the rules of simultaneous channel operation function, see 6.4.1 Basic Rules of Simultaneous Channel Operation Function.

Remark n: Channel number (n = 0 to 7)
p, q: Slave channel number (n < p < q ≤ 7)

6.1.3 8-bit timer operation function (channels 1 and 3 only)

The 8-bit timer operation function makes it possible to use a 16-bit timer channel in a configuration consisting of two 8-bit timer channels. This function can only be used for channels 1 and 3.

Caution There are several rules for using 8-bit timer operation function.
For details, see 6.4.2 Basic rules of 8-bit timer operation function (Only Channels 1 and 3).

Downloaded from Arrow.com.

6.2 Configuration of Timer Array Unit

Timer array unit includes the following hardware.

Table 6-1. Configuration of Timer Array Unit

Item	Configuration
Timer/counter	Timer/counter register 0n (TCR0n)
Register	Timer data register 0n (TDR0n)
Timer input	TI00 to TI07
Timer output	TO00 to TO07 pins, output controller
Control registers	<Registers of unit setting block> • Peripheral enable register 0 (PER0) • Timer clock select register 0 (TPS0) • Timer channel enable status register 0 (TE0) • Timer channel start register 0 (TS0) • Timer channel stop register 0 (TT0) • Timer input select register 0 (TIS0) • Timer output enable register 0 (TOE0) • Timer output register 0 (TO0) • Timer output level register 0 (TOL0) • Timer output mode register 0 (TOM0)
	<Registers of each channel> • Timer mode register 0n (TMR0n) • Timer status register 0n (TSR0n) • Noise filter enable register 1 (NFEN1) • Port mode control register (PMCxx)Note • Port mode register (PMxx)Note • Port register (Pxx)Note

<R> **Note** The Port mode control register (PMCxx), port mode registers (PMxx), and port registers (Pxx) to be set differ depending on the product. For details, see **4.5.3 Register setting examples for using the port and alternate functions**.

Remark n: Channel number (n = 0 to 7)

Alternate port for timer I/O of the timer array unit channels varies depending on products.

<R> **Table 6-2. Timer I/O Pins in the Products**

Timer array unit channel	30-pin products	20, 24-pin products
Channel 0	TI00/TO00	TI00/TO00
Channel 1	TI01/TO01	TI01/TO01
Channel 2	TI02/TO02	TI02/TO02
Channel 3	TI03/TO03	TI03/TO03
Channel 4	(TI04/TO04)	×
Channel 5	(TI05/TO05)	×
Channel 6	(TI06/TO06)	×
Channel 7	(TI07/TO07)	×

Remarks 1. If a pin is to be used for both timer input and timer output, it can be used only for timer input or timer output.

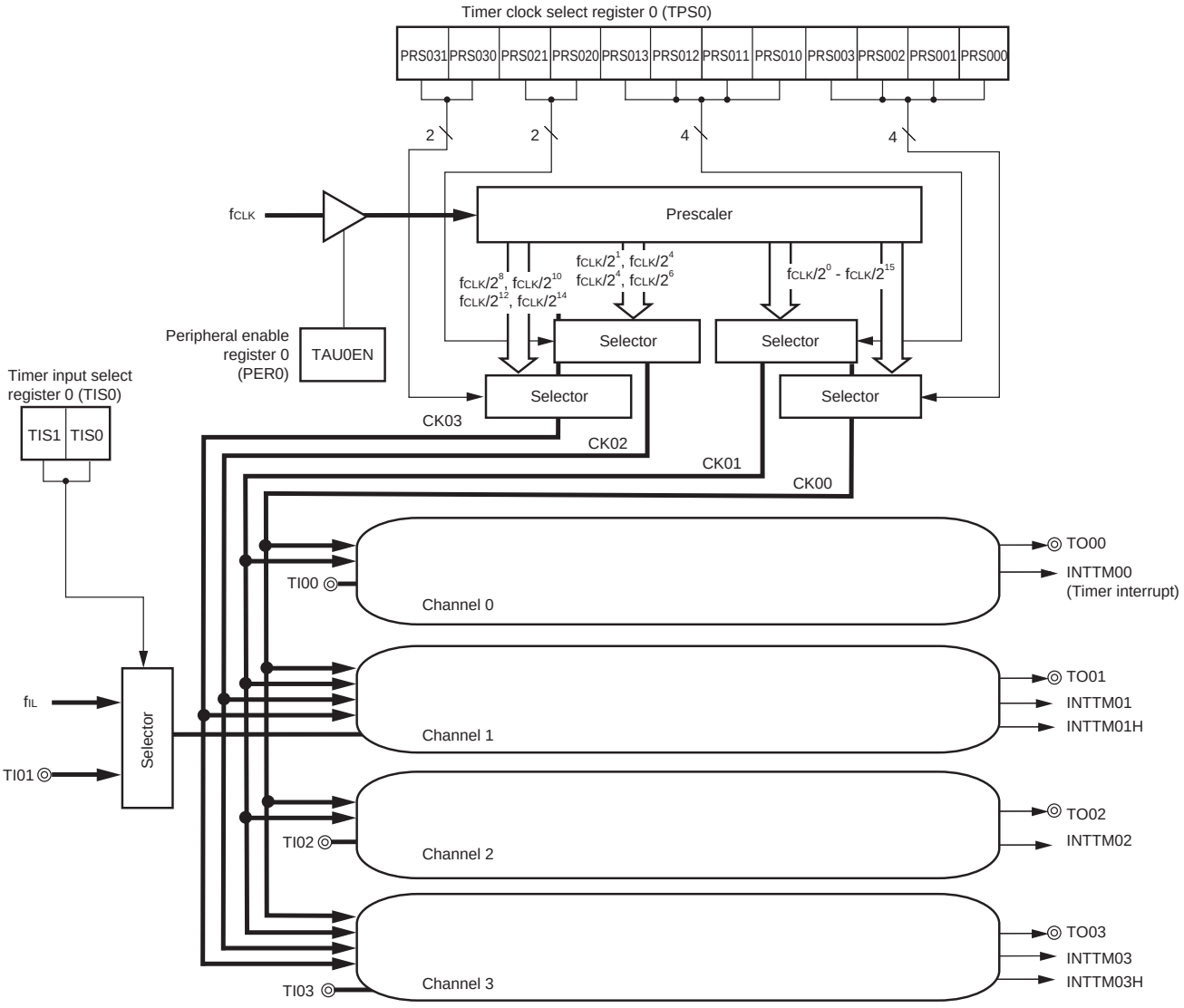
2. ×: The channel is not available

3. Pins in the parentheses indicate an alternate port when the bit 0 (PIOR0) of the peripheral I/O redirection register (PIOR) is set to "1" in 30-pin products.

Downloaded from Arrow.com.

Figures 6-1 to 6-3 show the block diagrams of the timer array unit.

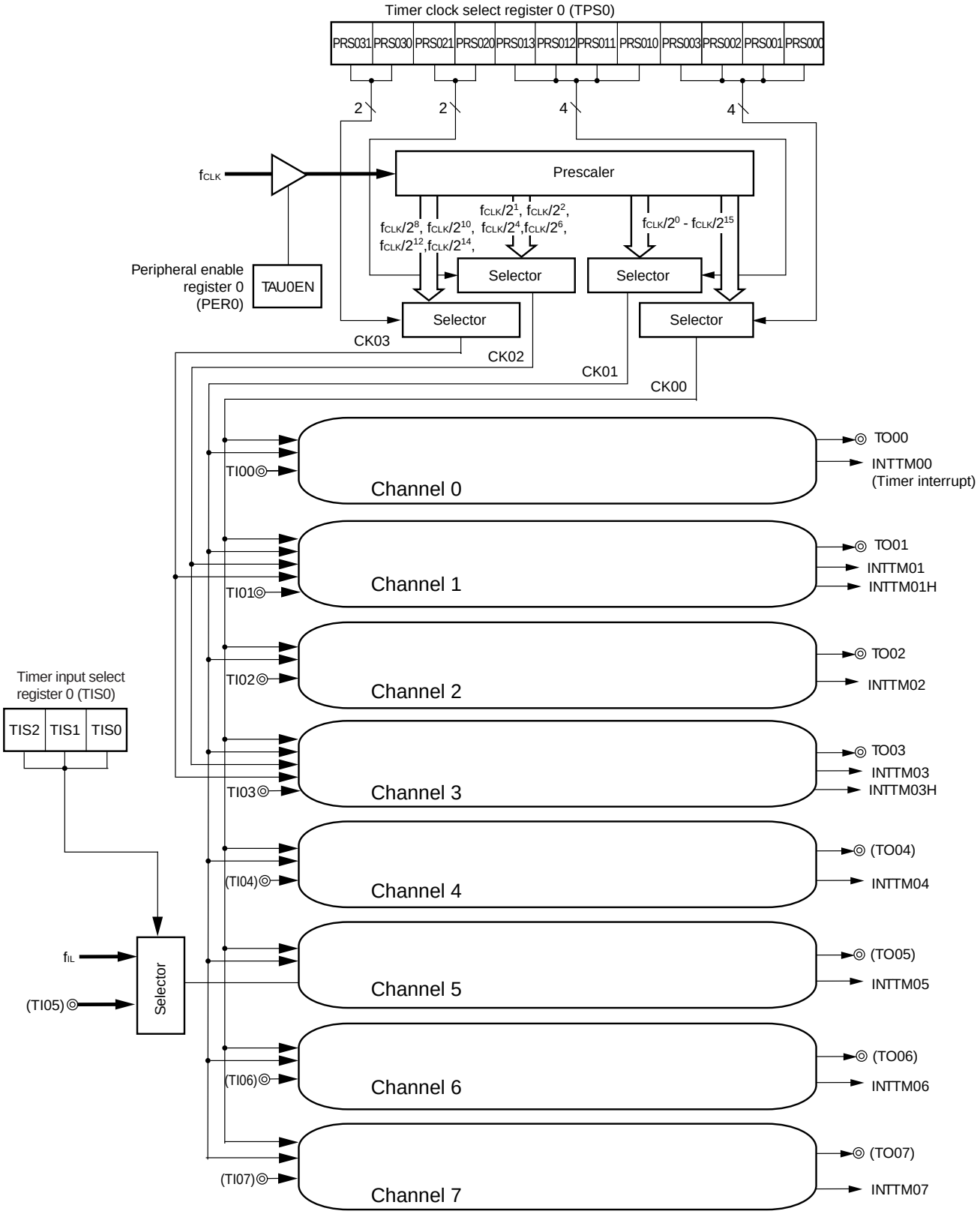
<R> **Figure 6-1. Entire Configuration of Timer Array Unit (20-, and 24-pin products)**



Remark f_{IL}: Low-speed on-chip oscillator clock frequency

Downloaded from Arrow.com.

<R> **Figure 6-2. Entire Configuration of Timer Array Unit (30-pin products)**



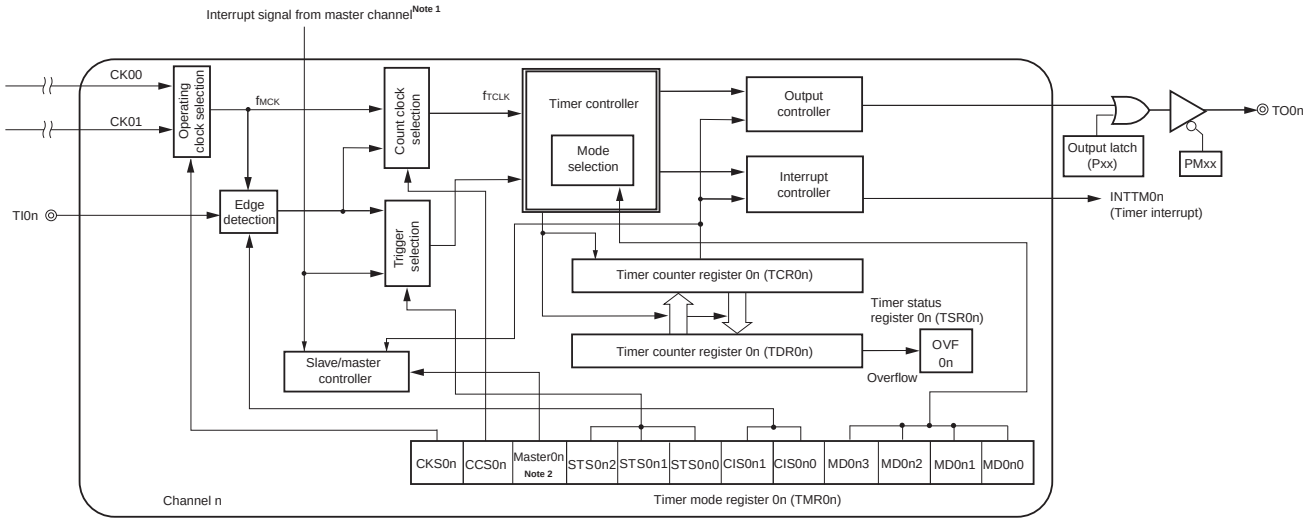
Remark Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR).

Downloaded from Arrow.com.

Figure 6-3. Internal Block Diagram of Channel of Timer Array Unit

(a) Channel 0, 2, 4, 6

<R>



Notes 1. Channels 2, 4, and 6 only

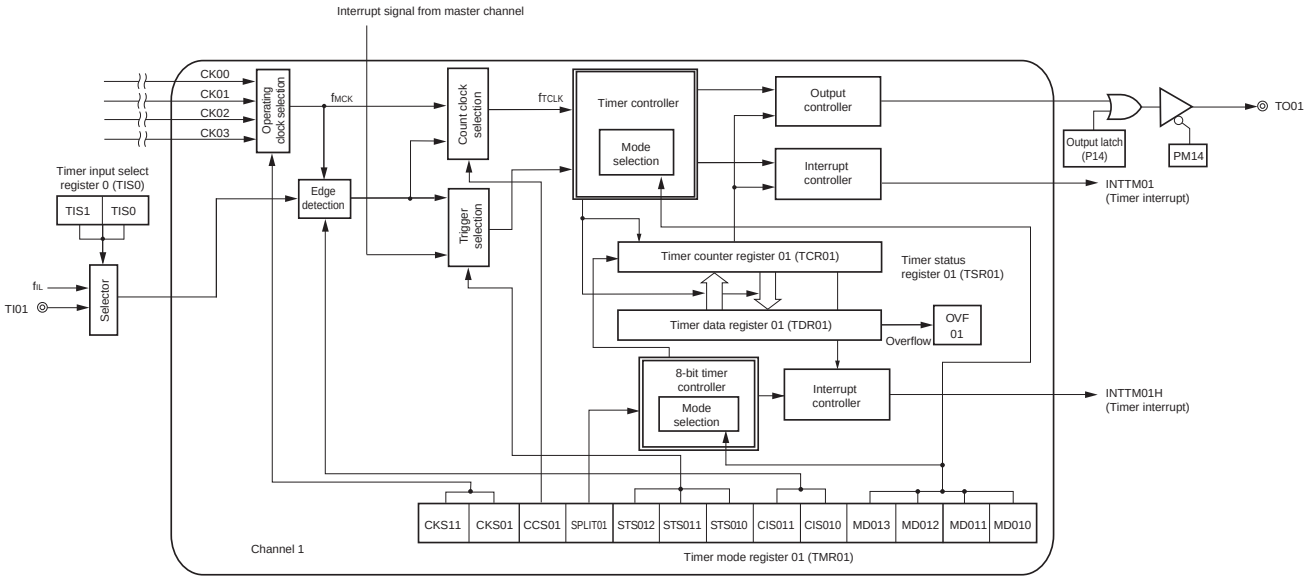
2. n = 2, 4, 6 only

Remarks 1. n = 0, 2, 4, or 6

2. Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR).

(b) Channel 1 for 20-pin and 24-pin product

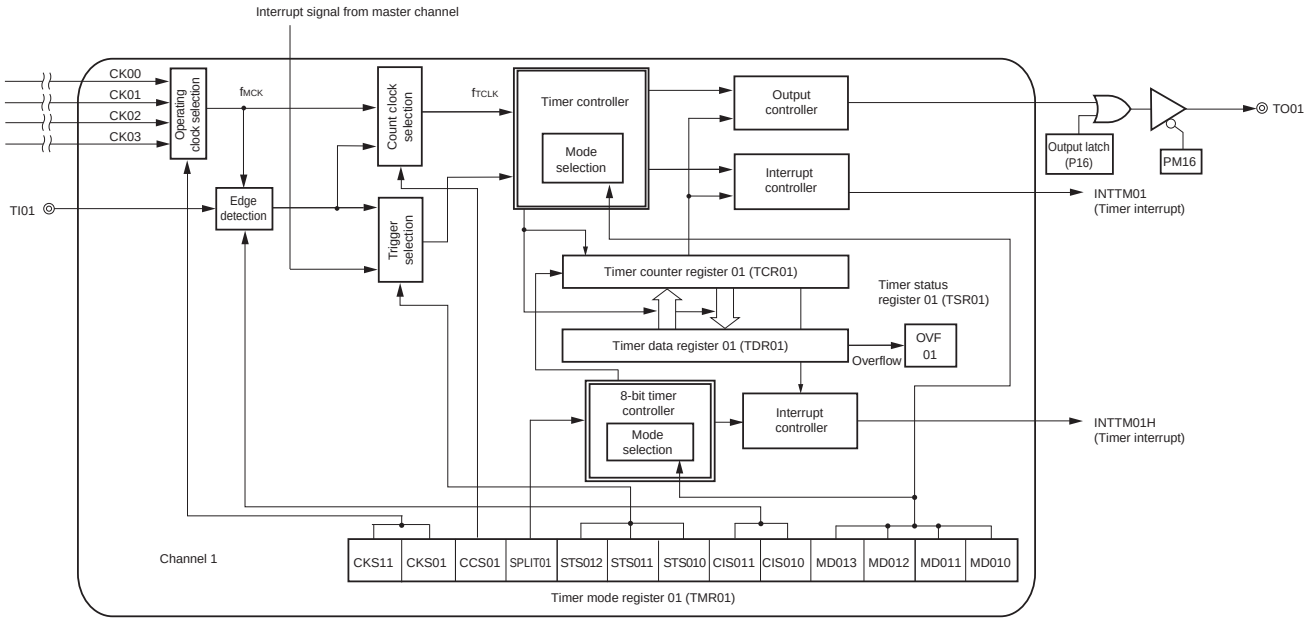
<R>



Downloaded from Arrow.com.

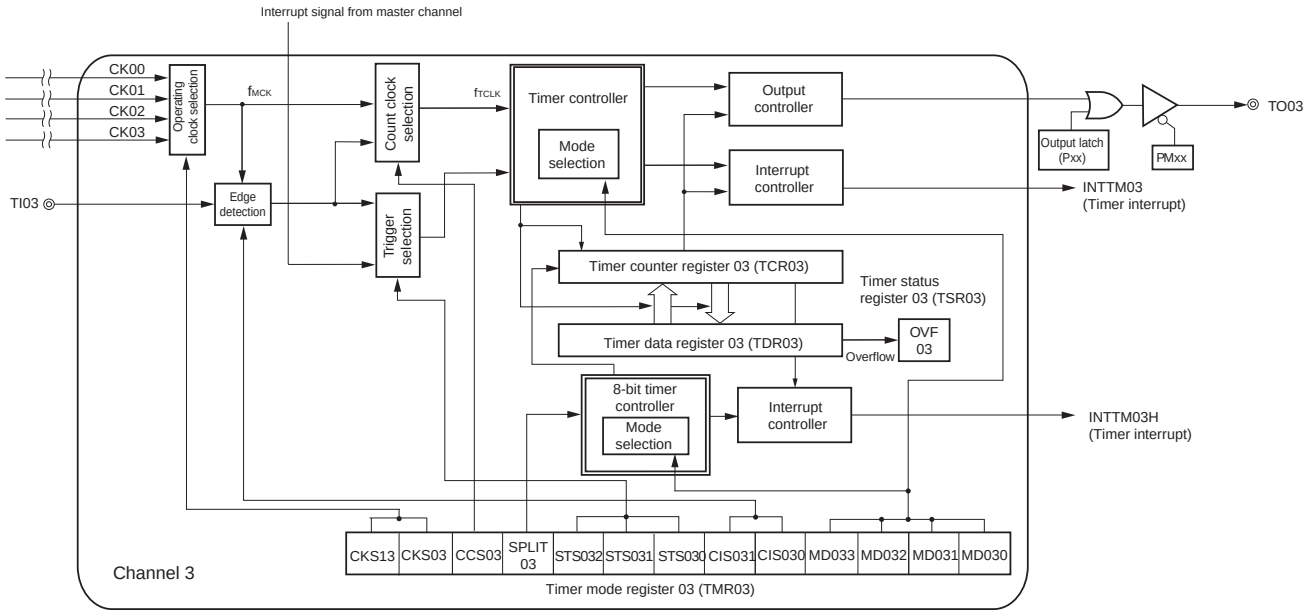
(c) Channel 1 for 30-pin product

<R>



(d) Channel 3

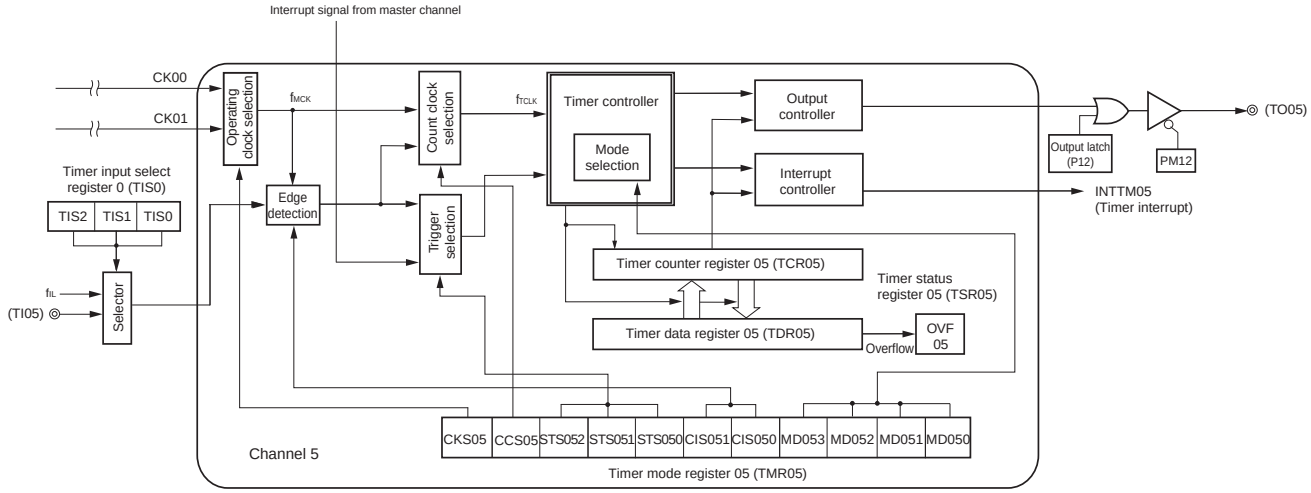
<R>



Downloaded from Arrow.com.

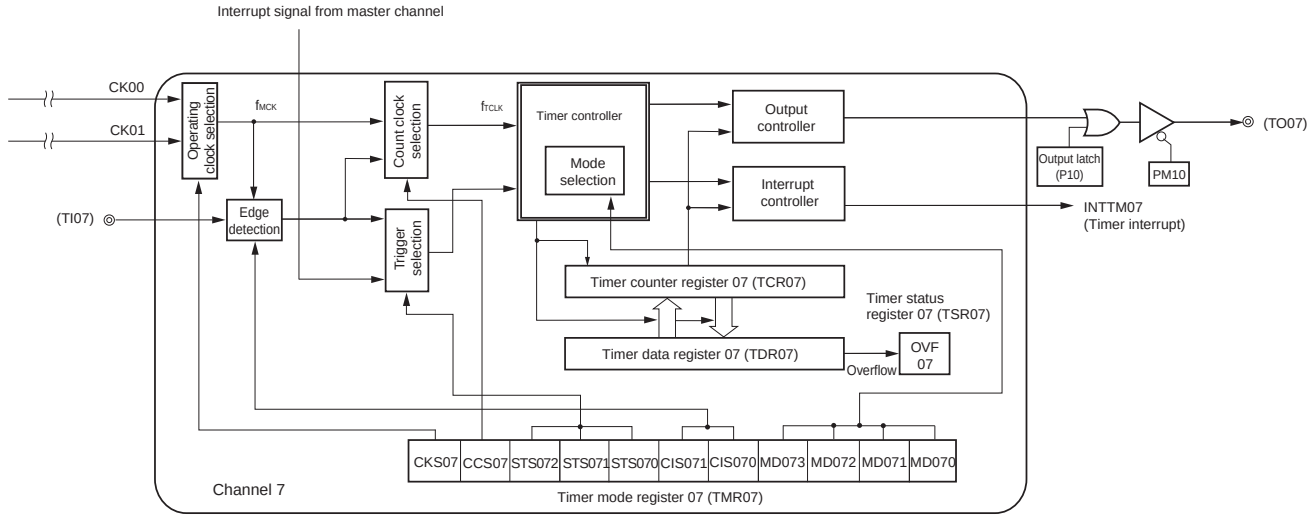
(e) Channel 5 for 30-Pin product

<R>



(f) Channel 7 for 30-Pin product

<R>



Remark Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR).

Downloaded from Arrow.com.

<R> 6.2.1 Timer/counter register 0n (TCR0n)

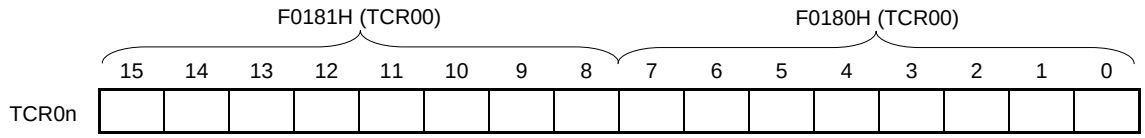
The TCR0n register is a 16-bit read-only register and is used to count clocks.

The value of this counter is incremented or decremented in synchronization with the rising edge of a count clock.

Whether the counter is incremented or decremented depends on the operation mode that is selected by the MD0n3 to MD0n0 bits of timer mode register 0n (TMR0n) (refer to **6.3.3 Timer mode register 0n (TMR0n)**).

Figure 6-4. Format of Timer/Counter Register 0n (TCR0n)

Address: F0180H, F0181H (TCR00) to F0186H, F0187H (TCR03) After reset: FFFFH R



Remark n: Channel number (n = 0 to 7)

The count value can be read by reading timer counter register 0n (TCR0n).

The count value is set to FFFFH in the following cases.

- When the reset signal is generated
- When the TAU0EN bit of peripheral enable register 0 (PER0) is cleared
- When counting of the slave channel has been completed in the PWM output mode
- When counting of the slave channel has been completed in the delay count mode
- When counting of the master/slave channel has been completed in the one-shot pulse output mode
- When counting of the slave channel has been completed in the multiple PWM output mode

The count value is cleared to 0000H in the following cases.

- When the start trigger is input in the capture mode
- When capturing has been completed in the capture mode

Caution The count value is not captured to timer data register 0n (TDR0n) even when the TCR0n register is read.

The TCR0n register read value differs as follows according to operation mode changes and the operating status.

Downloaded from Arrow.com.

Table 6-3. Timer/counter Register 0n (TCR0n) Read Value in Various Operation Modes

Operation Mode	Count Mode	Timer/counter register 0n (TCR0n) Read Value[Note]			
		Value if the operation mode was changed after releasing reset	Value if the count operation paused (TT0n = 1)	Value if the operation mode was changed after count operation paused (TT0n = 1)	Value when waiting for a start trigger after one count
Interval timer mode	Count down	FFFFH	Value if stop	Undefined	–
Capture mode	Count up	0000H	Value if stop	Undefined	–
Event counter mode	Count down	FFFFH	Value if stop	Undefined	–
One-count mode	Count down	FFFFH	Value if stop	Undefined	FFFFH
Capture & one-count mode	Count up	0000H	Value if stop	Undefined	Capture value of TDR0n register + 1

Note This indicates the value read from the TCR0n register when channel n has stopped operating as a timer (TE0n = 0) and has been enabled to operate as a counter (TS0n = 1). The read value is held in the TCR0n register until the count operation starts.

Remark n: Channel number (n = 0 to 7)

Downloaded from Arrow.com.

<R> 6.2.2 Timer data register 0n (TDR0n)

This is a 16-bit register from which a capture function and a compare function can be selected.

The capture or compare function can be switched by selecting an operation mode by using the MD0n3 to MD0n0 bits of timer mode register 0n (TMR0n).

The value of the TDR0n register can be changed at any time.

This register can be read or written in 16-bit units.

In addition, for the TDR01, TDR03 registers, while in the 8-bit timer mode (when the SPLIT bit of timer mode register m1, m3 (TMR01, TMR03) is 1), it is possible to read and write the data in 8-bit units, with TDR01H, TDR03H used as the higher 8 bits, and TDR01L, TDR03L used as the lower 8 bits.

Reset signal generation clears this register to 0000H.

Figure 6-5. Format of Timer Data Register 0n (TDR0n) (n = 0, 2, 4 to 7)

Address: FFF18H, FFF19H (TDR00), FFF64H, FFF65H (TDR02), FFF68H, FFF69H (TDR04) to FFF6EH, FFF6FH (TDR07) After reset: 0000H R/W

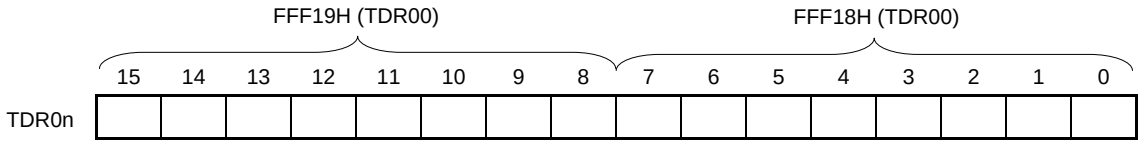
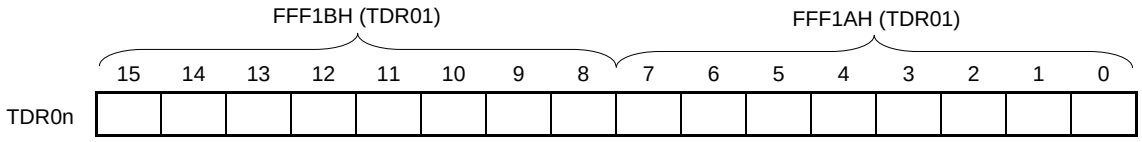


Figure 6-6. Format of Timer Data Register 01, 03 (TDR01, TDR03)

Address: FFF1AH, FFF1BH (TDR01), FFF66H, FFF67H (TDR03) After reset: 0000H R/W



(i) When timer data register 0n (TDR0n) is used as compare register

Counting down is started from the value set to the TDR0n register. When the count value reaches 0000H, an interrupt signal (INTTM0n) is generated. The TDR0n register holds its value until it is rewritten.

Caution The TDR0n register does not perform a capture operation even if a capture trigger is input, when it is set to the compare function.

(ii) When timer data register 0n (TDR0n) is used as capture register

The count value of timer/counter register 0n (TCR0n) is captured to the TDR0n register when the capture trigger is input.

A valid edge of the TI0n pin can be selected as the capture trigger. This selection is made by timer mode register 0n (TMR0n).

Remark n: Channel number (n = 0 to 7)

Downloaded from Arrow.com.

6.3 Registers Controlling Timer Array Unit

Timer array unit is controlled by the following registers.

- Peripheral enable register 0 (PER0)
- Timer clock select register 0 (TPS0)
- Timer mode register 0n (TMR0n)
- Timer status register 0n (TSR0n)
- Timer channel enable status register 0 (TE0)
- Timer channel start register 0 (TS0)
- Timer channel stop register 0 (TT0)
- Timer input select register 0 (TIS0)
- Timer output enable register 0 (TOE0)
- Timer output register 0 (TO0)
- Timer output level register 0 (TOL0)
- Timer output mode register 0 (TOM0)
- Noise filter enable register 1 (NFEN1)

<R> • Port mode control register (PMCxx)

<R> • Port mode register (PMxx)

<R> • Port register (Pxx)

Remark n: Channel number (n = 0 to 7)

<R> **Caution Which registers and bits are included depends on the product. Be sure to set bits that are not mounted to their initial values.**

Downloaded from Arrow.com.

6.3.1 Peripheral enable register 0 (PER0)

This registers is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

When the timer array unit is used, be sure to set bit 0 (TAU0EN) of this register to 1.

The PER0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 6-7. Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H After reset: 00H R/W

Symbol	<7>	6	<5>	<4>	<3>	<2>	1	<0>
PER0	TMKAEN	0	ADCEN	IICA0EN	SAU1EN[Note]	SAU0EN	0	TAU0EN

TAU0EN	Control of timer array unit clock
0	Stops supply of clock. • SFR used by the timer array unit cannot be written. • The timer array unit is in the reset status.
1	Supplies clock. • SFR used by the timer array unit can be read/written.

<R> **Note** 30-pin products only.

<R> **Cautions 1. When setting the timer array unit, be sure to set the following registers first while the TAU0EN bit is set to 1. If TAU0EN = 0, the values of the registers which control the timer array unit are cleared to their initial values and writing to them is ignored (except for the timer input select register 0 (TIS0), input switch control register (ISC), noise filter enable register 1 (NFEN1), port mode registers 0, 1, 3, 4 (PM0, PM1, PM3, PM4), port registers 0, 1, 3, 4 (P0, P1, P3, P4), and Port mode control register 0, 1, 4 (PMC0, PMC1, PMC4)).**

- **Timer clock select register 0 (TPS0)**
- **Timer mode register 0n (TMR0n)**
- **Timer status register 0n (TSR0n)**
- **Timer channel enable status register 0 (TE0)**
- **Timer channel start register 0 (TS0)**
- **Timer channel stop register 0 (TT0)**
- **Timer output enable register 0 (TOE0)**
- **Timer output register 0 (TO0)**
- **Timer output level register 0 (TOL0)**
- **Timer output mode register 0 (TOM0)**

2. Be sure to clear the following bits to 0:

- **20-, 24-pin products: Bits 1, 3, and 6**
- **30-pin products: Bits 1 and 6**

Downloaded from Arrow.com.

6.3.2 Timer clock select register 0 (TPS0)

The TPS0 register is a 16-bit register that is used to select four types of operation clocks (CK00 to CK03) that are commonly supplied to each channel from external prescaler.

<R> The TPS0 register is a 16-bit register that is used to select two types or four types of operation clocks (CK00, CK01, CK02, CK03) that are commonly supplied to each channel. CK00 is selected by using bits 3 to 0 of the TPS0 register, and CK01 is selected by using bits 7 to 4 of the TPS0 register. In addition, only for channels 1 and 3, CK02 and CK03 can be also selected. CK02 is selected by using bits 9 and 8 of the TPS0 register, and CK03 is selected by using bits 13 and 12 of the TPS0 register. Rewriting of the TPS0 register during timer operation is possible only in the following cases.

If the PRS000 to PRS003 bits can be rewritten (n = 0 to 7):
All channels for which CK00 is selected as the operation clock (CKS0n1, CKS0n0 = 0, 0) are stopped (TE0n = 0).
If the PRS010 to PRS013 bits can be rewritten (n = 0 to 7):
All channels for which CK01 is selected as the operation clock (CKS0n1, CKS0n0 = 0, 1) are stopped (TE0n = 0).
If the PRS020 and PRS021 bits can be rewritten (n = 1, 3):
All channels for which CK02 is selected as the operation clock (CKS0n1, CKS0n0 = 1, 0) are stopped (TE0n = 0).
If the PRS030 and PRS031 bits can be rewritten (n = 1, 3):
All channels for which CK03 is selected as the operation clock (CKS0n1, CKS0n0 = 1, 1) are stopped (TE0n = 0).

The TPS0 register can be set by a 16-bit memory manipulation instruction.
Reset signal generation clears this register to 0000H.

Downloaded from Arrow.com.

Figure 6-8. Format of Timer Clock Select register 0 (TPS0)

Address: F01B6H, F01B7H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPS0	0	0	PRS 031	PRS 030	0	0	PRS 021	PRS 020	PRS 013	PRS 012	PRS 011	PRS 010	PRS 003	PRS 002	PRS 001	PRS 000

<R>

PRS 0k3	PRS 0k2	PRS 0k1	PRS 0k0	Selection of operation clock (CK0k)[Note] (k = 0, 1)						
					f_{CLK} = 2 MHz	f_{CLK} = 4 MHz	f_{CLK} = 8 MHz	f_{CLK} = 16 MHz	f_{CLK} = 20 MHz	f_{CLK} = 24 MHz
0	0	0	0	f_{CLK}	2 MHz	4 MHz	8 MHz	16 MHz	20 MHz	24 MHz
0	0	0	1	$f_{CLK}/2$	1 MHz	2 MHz	4 MHz	8 MHz	10 MHz	12 MHz
0	0	1	0	$f_{CLK}/2^2$	500 kHz	1 MHz	2 MHz	4 MHz	5 MHz	6 MHz
0	0	1	1	$f_{CLK}/2^3$	250 kHz	500 kHz	1 MHz	2 MHz	2.5 MHz	3 MHz
0	1	0	0	$f_{CLK}/2^4$	125 kHz	250 kHz	500 kHz	1 MHz	1.25 MHz	1.5 MHz
0	1	0	1	$f_{CLK}/2^5$	62.5 kHz	125 kHz	250 kHz	500 kHz	625 kHz	750 kHz
0	1	1	0	$f_{CLK}/2^6$	31.3 kHz	62.5 kHz	125 kHz	250 kHz	313 kHz	375 kHz
0	1	1	1	$f_{CLK}/2^7$	15.6 kHz	31.3 kHz	62.5 kHz	125 kHz	156 kHz	188 kHz
1	0	0	0	$f_{CLK}/2^8$	7.81 kHz	15.6 kHz	31.3 kHz	62.5 kHz	78.1 kHz	93.8 kHz
1	0	0	1	$f_{CLK}/2^9$	3.91 kHz	7.81 kHz	15.6 kHz	31.3 kHz	39.1 kHz	46.9 kHz
1	0	1	0	$f_{CLK}/2^{10}$	1.95 kHz	3.91 kHz	7.81 kHz	15.6 kHz	19.5 kHz	23.4 kHz
1	0	1	1	$f_{CLK}/2^{11}$	977 Hz	1.95 kHz	3.91 kHz	7.81 kHz	9.77 kHz	11.7 kHz
1	1	0	0	$f_{CLK}/2^{12}$	488 Hz	997 Hz	1.95 kHz	3.91 kHz	4.88 kHz	5.86 kHz
1	1	0	1	$f_{CLK}/2^{13}$	244 Hz	488 Hz	977 Hz	1.95 kHz	2.44 kHz	2.93 kHz
1	1	1	0	$f_{CLK}/2^{14}$	122 Hz	244 Hz	488 Hz	977 Hz	1.22 kHz	1.46 kHz
1	1	1	1	$f_{CLK}/2^{15}$	61 Hz	122 Hz	244 Hz	488 Hz	610 Hz	732 Hz

<R>

PRS 021	PRS 020	Selection of operation clock (CK02)[Note]						
			f_{CLK} = 2 MHz	f_{CLK} = 4 MHz	f_{CLK} = 8 MHz	f_{CLK} = 16 MHz	f_{CLK} = 20 MHz	f_{CLK} = 24 MHz
0	0	$f_{CLK}/2$	1 MHz	2 MHz	4 MHz	8 MHz	10 MHz	12 MHz
0	1	$f_{CLK}/2^2$	500 kHz	1 MHz	2 MHz	4 MHz	5 MHz	6 MHz
1	0	$f_{CLK}/2^4$	125 kHz	250 kHz	500 kHz	1 MHz	1.25 MHz	1.5 MHz
1	1	$f_{CLK}/2^6$	31.3 kHz	62.5 kHz	125 kHz	250 kHz	313 kHz	375 kHz

<R>

PRS 031	PRS 030	Selection of operation clock (CK03)[Note]						
			f_{CLK} = 2 MHz	f_{CLK} = 4 MHz	f_{CLK} = 8 MHz	f_{CLK} = 16 MHz	f_{CLK} = 20 MHz	f_{CLK} = 24 MHz
0	0	$f_{CLK}/2^8$	7.81 kHz	15.6 kHz	31.3 kHz	62.5 kHz	78.1 kHz	93.8 kHz
0	1	$f_{CLK}/2^{10}$	1.95 kHz	3.91 kHz	7.81 kHz	15.6 kHz	19.5 kHz	23.4 kHz
1	0	$f_{CLK}/2^{12}$	488 Hz	977 Hz	1.95 kHz	3.91 kHz	4.88 kHz	5.86 kHz
1	1	$f_{CLK}/2^{14}$	122 Hz	244 Hz	488 Hz	977 Hz	1.22 kHz	1.46 kHz

Note When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), stop timer array unit (TT0 = 00FFH).

Downloaded from Arrow.com.

Cautions 1. Be sure to clear bits 15, 14, 11, 10 to "0".

2. If f_{CLK} (undivided) is selected as the operation clock (CKmk) and TDRnm is set to 0000H (n = 0 or 1, m = 0 to 7), interrupt requests output from timer array units cannot be used.

Remarks 1. f_{CLK}: CPU/peripheral hardware clock frequency

2. The above selected clock , but a signal which becomes high level for one period of f_{CLK} from its rising edge (m = 2 to 15). For details, see **6.5.1 Count clock (f_{TCLK}).**

By using channels 1 and 3 in the 8-bit timer mode and specifying CK02 or CK03 as the operation clock, the interval times shown in **Table 6-4** can be achieved by using the interval timer function.

Table 6-4. Interval Times Available for Operation Clock CKS02 or CKS03

Clock		Interval time (f_{CLK} = 20 MHz)[Note]			
		10 μs	100 μs	1 ms	10 ms
CK02	$f_{CLK}/2$	√	–	–	–
	$f_{CLK}/2^2$	√	–	–	–
	$f_{CLK}/2^4$	√	√	–	–
	$f_{CLK}/2^6$	–	√	√	–
CK03	$f_{CLK}/2^8$	–	√	√	–
	$f_{CLK}/2^{10}$	–	–	√	–
	$f_{CLK}/2^{12}$	–	–	–	√
	$f_{CLK}/2^{14}$	–	–	–	–

Note The margin is within 4 %.

Remarks 1. f_{CLK}: CPU/peripheral hardware clock frequency

2. For details of a signal of $f_{CLK}/2^n$ selected with the TPSm register, see **6.5.1 Count clock (f_{TCLK})**.

6.3.3 Timer mode register 0n (TMR0n)

The TMR0n register sets an operation mode of channel n. This register is used to select the operation clock (f_{MCK}), select the count clock, select the master/slave, select the 16 or 8-bit timer (only for channels 1 and 3), specify the start trigger and capture trigger, select the valid edge of the timer input, and specify the operation mode (interval, capture, event counter, one-count, or capture and one-count).

Rewriting the TMR0n register is prohibited when the register is in operation (when TE0n = 1). However, bits 7 and 6 (CIS0n1, CIS0n0) can be rewritten even while the register is operating with some functions (when TE0n = 1) (for details, see **6.8 Independent Channel Operation Function of Timer Array Unit** and **6.9 Simultaneous Channel Operation Function of Timer Array Unit)**.

The TMR0n register can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Caution The bits mounted depend on the channels in the bit 11 of TMR0n register.

TMR02, TMR04, TMR06: MASTER0n bit (n = 2, 4, 6)

TMR01, TMR03: SPLIT0n bit (n = 1, 3)

TMR00, TMR05, TMR07: Fixed to 0

Downloaded from Arrow.com.

Figure 6-9. Format of Timer Mode Register 0n (TMR0n) (1/4)

Address: : F0190H, F0191H (TMR00) to F019EH, F019FH (TMR07) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR0n (n = 2, 4, 6)	CKS 0n1	CKS 0n0	0	CCS 0n	MASTER0n	STS 0n2	STS 0n1	STS 0n0	CIS 0n1	CIS 0n0	0	0	MD 0n3	MD 0n2	MD 0n1	MD 0n0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR0n (n = 1, 3)	CKS 0n1	CKS 0n0	0	CCS 0n	SPLIT 0n	STS 0n2	STS 0n1	STS 0n0	CIS 0n1	CIS 0n0	0	0	MD 0n3	MD 0n2	MD 0n1	MD 0n0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR0n (n = 0, 5, 7)	CKS 0n1	CKS 0n0	0	CCS 0n	0Note	STS 0n2	STS 0n1	STS 0n0	CIS 0n1	CIS 0n0	0	0	MD 0n3	MD 0n2	MD 0n1	MD 0n0

CKS0n1	CKS0n0	Selection of operation clock (f_{MCK}) of channel n
0	0	Operation clock CK00 set by timer clock select register 0 (TPS0)
0	1	Operation clock CK02 set by timer clock select register 0 (TPS0)
1	0	Operation clock CK01 set by timer clock select register 0 (TPS0)
1	1	Operation clock CK03 set by timer clock select register 0 (TPS0)
Operation clock (f_{MCK}) is used by the edge detector. A count clock (f_{TCLK}) and a sampling clock are generated depending on the setting of the CCS0n bit. The operation clocks CK02 and CK03 can only be selected for channels 1 and 3.		

CCS0n	Selection of count clock (f_{TCLK}) of channel n
0	Operation clock (f_{MCK}) specified by the CKS0n0 and CKS0n1 bits
1	Valid edge of input signal input from the TI0n pin In channel 1 for 20-, 24-pin product and channel 5 for 30-pin product, Valid edge of input signal selected by TIS0
Count clock (f_{TCLK}) is used for the timer/counter, output controller, and interrupt controller.	

Note Bit 11 is a read-only bit and fixed to 0. Writing to this bit is ignored.

Cautions 1. Be sure to clear bits 13, 5, and 4 to "0".

2. The timer array unit must be stopped (TT0 = 00FFH) if the clock selected for f_{CLK} is changed (by changing the value of the system clock control register (CKC)), even if the operating clock specified by using the CKS0n0 and CKS0n1 bits (f_{MCK}) or the valid edge of the signal input from the TI0n pin is selected as the count clock (f_{TCLK}).

Remark n: Channel number (n = 0 to 7)

Downloaded from Arrow.com.

Figure 6-9. Format of Timer Mode Register 0n (TMR0n) (2/4)

Address: : F0190H, F0191H (TMR00) to F019EH, F019FH (TMR07) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR0n (n =2, 4, 6)	CKS 0n1	CKS 0n0	0	CCS 0n	MASTER0n	STS 0n2	STS 0n1	STS 0n0	CIS 0n1	CIS 0n0	0	0	MD 0n3	MD 0n2	MD 0n1	MD 0n0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR0n (n = 1, 3)	CKS 0n1	CKS 0n0	0	CCS 0n	SPLIT 0n	STS 0n2	STS 0n1	STS 0n0	CIS 0n1	CIS 0n0	0	0	MD 0n3	MD 0n2	MD 0n1	MD 0n0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR0n (n = 0, 5, 7)	CKS 0n1	CKS 0n0	0	CCS 0n	0[Note]	STS 0n2	STS 0n1	STS 0n0	CIS 0n1	CIS 0n0	0	0	MD 0n3	MD 0n2	MD 0n1	MD 0n0

(Bit 11 of TMR0n (n = 2, 4, 6))

MASTER0n	Selection between using channel n independently or simultaneously with another channel (as a slave or master)
0	Operates in independent channel operation function or as slave channel in simultaneous channel operation function.
1	Operates as master channel in simultaneous channel operation function.
Only the 2, 4, 6 channel can be set as a master channel (MASTER0n = 1). Be sure to use channel 0, 5, 7 are fixed to 0 (Regardless of the bit setting, channel 0 operates as master, because it is the highest channel). Clear the MASTER0n bit to 0 for a channel that is used with the independent channel operation function.	

(Bit 11 of TMR0n (n = 1, 3))

SPLIT0n	Selection of 8 or 16-bit timer operation for channels 1 and 3
0	Operates as 16-bit timer. (Operates in independent channel operation function or as slave channel in simultaneous channel operation function.)
1	Operates as 8-bit timer.

STS0n2	STS0n1	STS0n0	Setting of start trigger or capture trigger of channel n
0	0	0	Only software trigger start is valid (other trigger sources are unselected).
0	0	1	Valid edge of the TI0n pin input is used as both the start trigger and capture trigger.
0	1	0	Both the edges of the TI0n pin input are used as a start trigger and a capture trigger.
1	0	0	Interrupt signal of the master channel is used (when the channel is used as a slave channel with the simultaneous channel operation function).
Other than above			Setting prohibited

Note Bit 11 is a read-only bit and fixed to 0. Writing to this bit is ignored.

Remark n: Channel number (n = 0 to 7)

Downloaded from Arrow.com.

Figure 6-9. Format of Timer Mode Register 0n (TMR0n) (3/4)

Address: : F0190H, F0191H (TMR00) to F019EH, F019FH (TMR07) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR0n (n = 2, 4, 6)	CKS 0n1	CKS 0n0	0	CCS 0n	MASTER0n	STS 0n2	STS 0n1	STS 0n0	CIS 0n1	CIS 0n0	0	0	MD 0n3	MD 0n2	MD 0n1	MD 0n0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR0n (n = 1, 3)	CKS 0n1	CKS 0n0	0	CCS 0n	SPLIT 0n	STS 0n2	STS 0n1	STS 0n0	CIS 0n1	CIS 0n0	0	0	MD 0n3	MD 0n2	MD 0n1	MD 0n0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR0n (n=0, 5, 7)	CKS 0n1	CKS 0n0	0	CCS 0n	0[Note]	STS 0n2	STS 0n1	STS 0n0	CIS 0n1	CIS 0n0	0	0	MD 0n3	MD 0n2	MD 0n1	MD 0n0

CIS0n1	CIS0n0	Selection of TI0n pin input valid edge
0	0	Falling edge
0	1	Rising edge
1	0	Both edges (when low-level width is measured) Start trigger: Falling edge, Capture trigger: Rising edge
1	1	Both edges (when high-level width is measured) Start trigger: Rising edge, Capture trigger: Falling edge
If both the edges are specified when the value of the STS0n2 to STS0n0 bits is other than 010B, set the CIS0n1 to CIS0n0 bits to 10B.		

Note Bit 11 is a read-only bit and fixed to 0. Writing to this bit is ignored.

Remark n: Channel number (n = 0 to 7)

Downloaded from Arrow.com.

Figure 6-9. Format of Timer Mode Register 0n (TMR0n) (4/4)

Address: : F0190H, F0191H (TMR00) to F019EH, F019FH (TMR07) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR0n (n = 2, 4, 6)	CKS 0n1	CKS 0n0	0	CCS 0n	MASTER0n	STS 0n2	STS 0n1	STS 0n0	CIS 0n1	CIS 0n0	0	0	MD 0n3	MD 0n2	MD 0n1	MD 0n0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR0n (n = 1, 3)	CKS 0n1	CKS 0n0	0	CCS 0n	SPLIT 0n	STS 0n2	STS 0n1	STS 0n0	CIS 0n1	CIS 0n0	0	0	MD 0n3	MD 0n2	MD 0n1	MD 0n0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
<R> TMR0n (n=0, 5, 7)	CKS 0n1	CKS 0n0	0	CCS 0n	0 Note 1	STS 0n2	STS 0n1	STS 0n0	CIS 0n1	CIS 0n0	0	0	MD 0n3	MD 0n2	MD 0n1	MD 0n0

MD 0n3	MD 0n2	MD 0n1	Setting of operation mode of channel n	Corresponding function	Count operation of TCR
0	0	0	Interval timer mode	Interval timer / Square wave output / Divider function / PWM output (master)	Down count
0	1	0	Capture mode	Input pulse interval measurement	Up count
0	1	1	Event counter mode	External event counter	Down count
1	0	0	One-count mode	Delay counter / One-shot pulse output / PWM output (slave)	Down count
1	1	0	Capture & one-count mode	Measurement of high-/low-level width of input signal	Up count
Other than above			Setting prohibited		
The operation of MD0n0 bit changes depending on the operation of each mode (refer to the table bellow)					

Operation mode (Value set by the MD0n3 to MD0n1 bits (see table above))	MD 0n0	Setting of starting counting and interrupt
• Interval timer mode (0, 0, 0)	0	Timer interrupt is not generated when counting is started (timer output does not change, either).
• Capture mode (0, 1, 0)	1	Timer interrupt is generated when counting is started (timer output also changes).
• Event counter mode (0, 1, 1)	0	Timer interrupt is not generated when counting is started (timer output does not change, either).
<R> • One-count mode Note 2 (1, 0, 0)	0	Start trigger is invalid during counting operation. At that time, interrupt is not generated, either.
<R>	1	Start trigger is valid during counting operation Note 3. At that time, interrupt is also generated.
• Capture & one-count mode (1, 1, 0)	0	Timer interrupt is not generated when counting is started (timer output does not change, either). Start trigger is invalid during counting operation. At that time interrupt is not generated, either.
Other than above		Setting prohibited

Downloaded from Arrow.com.

<R> **Notes 1.** Bit 11 is a read-only bit and fixed to 0. Writing to this bit is ignored.

2. In one-count mode, interrupt output (INTTM0n) when starting a count operation and TO0n output are not controlled.

3. If the start trigger (TS0n = 1) is issued during operation, the counter is initialized, and recounting is started (interrupt request is not generated).

Remark n: Channel number (n = 0 to 7)

6.3.4 Timer status register 0n (TSR0n)

The TSR0n register indicates the overflow status of the counter of channel n.

The TSR0n register is valid only in the capture mode (MD0n3 to MD0n1 = 010B) and capture & one-count mode (MD0n3 to MD0n1 = 110B). It will not be set in any other mode. See Table 6-4 for the operation of the OVF bit in each operation mode and set/clear conditions.

The TSR0n register can be read by a 16-bit memory manipulation instruction.

The lower 8 bits of the TSR0n register can be set with an 8-bit memory manipulation instruction with TSR0nL.

Reset signal generation clears this register to 0000H.

Figure 6-10. Format of Timer Status Register 0n (TSR0n)

Address: F01A0H, F01A1H (TSR00) to F01AEH, F01AFH (TSR07) After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSR0n	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	OVF

OVF	Counter overflow status of channel n
0	Overflow does not occur.
1	Overflow occurs.
When OVF = 1, this flag is cleared (OVF = 0) when the next value is captured without overflow.	

Remark n: Channel number (n = 0 to 7)

Table 6-5. OVF Bit Operation and Set/Clear Conditions in Each Operation Mode

Timer operation mode	OVF bit	Set/clear conditions
• Capture mode	clear	When no overflow has occurred upon capturing
• Capture & one-count mode	set	When an overflow has occurred upon capturing
• Interval timer mode • Event counter mode • One-count mode	clear	– (Use prohibited)
	set	

Remark The OVF bit does not change immediately after the counter has overflowed, but changes upon the subsequent capture.

Downloaded from Arrow.com.

6.3.5 Timer channel enable status register 0 (TE0)

The TE0 register is used to enable or stop the timer operation of each channel.

Each bit of the TE0 register corresponds to each bit of the timer channel start register 0 (TS0) and the timer channel stop register 0 (TT0). When a bit of the TSm register is set to 1, the corresponding bit of this register is set to 1. When a bit of the TT0 register is set to 1, the corresponding bit of this register is cleared to 0.

The TE0 register can be read by a 16-bit memory manipulation instruction.

The lower 8 bits of the TE0 register can be set with a 1-bit or 8-bit memory manipulation instruction with TE0L.

Reset signal generation clears this register to 0000H.

Figure 6-11. Format of Timer Channel Enable Status register 0 (TE0)

Address: F01B0H, F01B1H After reset: 0000H R

20- and 24-pin products

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TE0	0	0	0	0	TEH03	0	TEH01	0	0	0	0	0	TE03	TE02	TE01	TE00

30-pin products

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TE0	0	0	0	0	TEH03	0	TEH01	0	TE07	TE06	TE05	TE04	TE03	TE02	TE01	TE00

TEH03	Indication of whether operation of the higher 8-bit timer is enabled or stopped when channel 3 is in the 8-bit timer mode
0	Operation is stopped.
1	Operation is enabled.

TEH01	Indication of whether operation of the higher 8-bit timer is enabled or stopped when channel 1 is in the 8-bit timer mode
0	Operation is stopped.
1	Operation is enabled.

TE0n	Indication of operation enable/stop status of channel n
0	Operation is stopped.
1	Operation is enabled.
This bit displays whether operation of the lower 8-bit timer for TE01, TE03 is enabled or stopped when channel 1, 3 is in the 8-bit timer mode.	

Remark n: Channel number (n = 0 to 7)

Downloaded from Arrow.com.

6.3.6 Timer channel start register 0 (TS0)

The TS0 register is a trigger register that is used to initialize timer/counter register 0n (TCR0n) and start the counting operation of each channel.

When a bit of this register is set to 1, the corresponding bit of timer channel enable status register 0 (TE0) is set to 1. The TS0n, TSH01, TSH03 bits are immediately cleared when operation is enabled (TE0n, TEH01, TEH03 = 1), because they are trigger bits.

The TS0 register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the TS0 register can be set with a 1-bit or 8-bit memory manipulation instruction with TS0L.

Reset signal generation clears this register to 0000H.

Figure 6-12. Format of Timer Channel Start register 0 (TS0)

Address: F01B2H, F01B3H After reset: 0000H R/W

20- and 24-pin products

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TS0	0	0	0	0	TSH03	0	TSH01	0	0	0	0	0	TS03	TS02	TS01	TS00

30-pin products

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TS0	0	0	0	0	TSH03	0	TSH01	0	TS07	TS06	TS05	TS04	TS03	TS02	TS01	TS00

TSH03	Trigger to enable operation (start operation) of the higher 8-bit timer when channel 3 is in the 8-bit timer mode
0	No trigger operation
1	The TEH03 bit is set to 1 and the count operation becomes enabled. The TCR03 register count operation start in the interval timer mode in the count operation enabled state (see Table 6-6).

TSH01	Trigger to enable operation (start operation) of the higher 8-bit timer when channel 1 is in the 8-bit timer mode
0	No trigger operation
1	The TEH01 bit is set to 1 and the count operation becomes enabled. The TCR01 register count operation start in the interval timer mode in the count operation enabled state (see Table 6-6).

TS0n	Operation enable (start) trigger of channel n
0	No trigger operation
1	The TE0n bit is set to 1 and the count operation becomes enabled. The TCR0n register count operation start in the count operation enabled state varies depending on each operation mode (see Table 6-6). This bit is the trigger to enable operation (start operation) of the lower 8-bit timer for TS01 and TS03 when channel 1 or 3 is in the 8-bit timer mode.

Cautions 1. Be sure to clear undefined bits to "0".

2. When switching from a function that does not use TI0n pin input to one that does, the following wait period is required from when timer mode register 0n (TMR0n) is set until the TS0n (TSH01, TSH03) bit is set to 1.

When the TI0n pin noise filter is enabled (TNFEN = 1): Four cycles of the operation clock (f_{MCK})

Downloaded from Arrow.com.

When the TI0n pin noise filter is disabled (TNFEN = 0): Two cycles of the operation clock (f_{MCK})

Remarks 1. When the TS0 register is read, 0 is always read.

2. n: Channel number (n = 0 to 7)

6.3.7 Timer channel stop register 0 (TT0)

The TT0 register is a trigger register that is used to stop the counting operation of each channel.

When a bit of this register is set to 1, the corresponding bit of timer channel enable status register 0 (TE0) is cleared to 0. The TT0n, TTH01, TTH03 bits are immediately cleared when operation is stopped (TE0n, TEH01, TEH03 = 0), because they are trigger bits.

The TT0 register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the TT0 register can be set with a 1-bit or 8-bit memory manipulation instruction with TT0L.

Reset signal generation clears this register to 0000H.

Figure 6-13. Format of Timer Channel Stop register 0 (TT0)

Address: F01B4H, F01B5H After reset: 0000H R/W

20- and 24-pin products

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TT0	0	0	0	0	TTH03	0	TTH01	0	0	0	0	0	TT03	TT02	TT01	TT00

30-pin products

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TT0	0	0	0	0	TTH03	0	TTH01	0	TT07	TT06	TT05	TT04	TT03	TT02	TT01	TT00

TTH03	Trigger to stop operation of the higher 8-bit timer when channel 3 is in the 8-bit timer mode
0	No trigger operation
1	TEH03 is cleared to 0 and the count operation is stopped.

TTH01	Trigger to stop operation of the higher 8-bit timer when channel 1 is in the 8-bit timer mode
0	No trigger operation
1	TEH01 is cleared to 0 and the count operation is stopped.

TT0n	Operation stop trigger of channel n
0	No trigger operation
1	TE0n is cleared to 0 and the count operation is stopped. This bit is the trigger to stop operation of the lower 8-bit timer for TT01 and TT03 when channel 1 or 3 is in the 8-bit timer mode.

Caution Be sure to clear undefined bits to "0".

Remarks 1. When the TT0 register is read, 0 is always read.

2. n: Channel number (n = 0 to 7)

Downloaded from Arrow.com.

6.3.8 Timer input select register 0 (TIS0)

The TIS0 register is used to select the channel 1 for 20- or 24-pin product, channel 5 for 30-pin product timer input.

The TIS0 register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 6-14. Format of Timer Input Select register 0 (TIS0)

Address: F0074H After reset: 00H R/W

20- and 24-pin products

Symbol	7	6	5	4	3	2	1	0
TIS0	0	0	0	0	0	0	TIS01	TIS00

TIS01	TIS00	Selection of timer input used with channel 1
×	0	Input signal of timer input pin (TI01)
0	1	Low-speed on-chip oscillator clock (f_{IL})
1	1	Setting prohibited

30-pin products

Symbol	7	6	5	4	3	2	1	0
TIS0	0	0	0	0	0	TIS2	TIS01	TIS00

TIS2	TIS01	TIS00	Selection of timer input used with channel 5
0	×	×	Input signal of timer input pin (TI05)
1	0	0	Low-speed on-chip oscillator clock (f_{IL})
Other than above			Setting prohibited

×: don't care

Downloaded from Arrow.com.

6.3.9 Timer output enable register 0 (TOE0)

The TOE0 register is used to enable or disable timer output of each channel.

Channel n for which timer output has been enabled becomes unable to rewrite the value of the TO0n bit of timer output register 0 (TO0) described later by software, and the value reflecting the setting of the timer output function through the count operation is output from the timer output pin (TO0n).

The TOE0 register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the TOE0 register can be set with a 1-bit or 8-bit memory manipulation instruction with TOE0L.

Reset signal generation clears this register to 0000H.

Figure 6-15. Format of Timer Output Enable register 0 (TOE0)

Address: F01BAH, F01BBH After reset: 0000H R/W

20- and 24-pin products

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOE0	0	0	0	0	0	0	0	0	0	0	0	0	TOE03	TOE02	TOE01	TOE00

30-pin products

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOE0	0	0	0	0	0	0	0	0	TOE07	TOE06	TOE05	TOE04	TOE03	TOE02	TOE01	TOE00

<R>

TOE0n	Timer output enable/disable of channel n
0	Disable output of timer. Without reflecting on TO0n bit timer operation, to fixed the output. Writing to the TO0n bit is enabled and the level set in the TO0n bit is output from the TO0n pin.
1	Enable output of timer. Reflected in the TO0n bit timer operation, to generate the output waveform. Writing to the TO0n bit is ignored.

Caution Be sure to clear undefined bits to "0".

Remark n: Channel number (n = 0 to 7)

Downloaded from Arrow.com.

6.3.10 Timer output register 0 (TO0)

The TO0 register is a buffer register of timer output of each channel.

The value of each bit in this register is output from the timer output pin (TO0n) of each channel.

The TO0n bit oh this register can be rewritten by software only when timer output is disabled (TOE0n = 0). When timer output is enabled (TOE0n = 1), rewriting this register by software is ignored, and the value is changed only by the timer operation.

<R> To use the TO00, TO01, TO02, TO03, (TO04), (TO05), (TO06), or (TO07) pin as a port function pin, set the corresponding TO0n bit to 0.

The TO0 register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the TO0 register can be set with an 8-bit memory manipulation instruction with TO0L.

Reset signal generation clears this register to 0000H.

Figure 6-16. Format of Timer Output register 0 (TO0)

Address: F01B8H, F01B9H After reset: 0000H R/W

20- and 24-pin products

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TO0	0	0	0	0	0	0	0	0	0	0	0	0	TO03	TO02	TO01	TO00

30-pin products

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TO0	0	0	0	0	0	0	0	0	TO07	TO06	TO05	TO04	TO03	TO02	TO01	TO00

TO0n	Timer output of channel n
0	Timer output value is "0".
1	Timer output value is "1".

Caution Be sure to clear undefined bits to "0".

Remark n: Channel number (n = 0 to 7)

Downloaded from Arrow.com.

6.3.11 Timer output level register 0 (TOL0)

The TOL0 register is a register that controls the timer output level of each channel.

The setting of the inverted output of channel n by this register is reflected at the timing of set or reset of the timer output signal while the timer output is enabled (TOE0n = 1) in the Slave channel output mode (TOM0n = 1). In the master channel output mode (TOM0n = 0), this register setting is invalid.

The TOL0 register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the TOL0 register can be set with an 8-bit memory manipulation instruction with TOL0L.

Reset signal generation clears this register to 0000H.

Figure 6-17. Format of Timer Output Level register 0 (TOL0)

Address: F01BCH, F01BDH After reset: 0000H R/W

20- and 24-pin products

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOL0	0	0	0	0	0	0	0	0	0	0	0	0	TOL03	TOL02	TOL01	0

30-pin products

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOL0	0	0	0	0	0	0	0	0	TOL07	TOL06	TOL05	TOL04	TOL03	TOL02	TOL01	0

TOL0n	Control of timer output level of channel n
0	Positive logic output (active-high)
1	Negative logic output (active-low)

Caution Be sure to clear undefined bits to "0".

Remarks 1. If the value of this register is rewritten during timer operation, the timer output logic is inverted when the timer output signal changes next, instead of immediately after the register value is rewritten.

2. n: Channel number (n = 1 to 7)

Downloaded from Arrow.com.

6.3.12 Timer output mode register 0 (TOM0)

The TOM0 register is used to control the timer output mode of each channel.

When a channel is used for the independent channel operation function, set the corresponding bit of the channel to be used to 0.

When a channel is used for the simultaneous channel operation function (PWM output, one-shot pulse output, or multiple PWM output), set the corresponding bit of the master channel to 0 and the corresponding bit of the slave channel to 1.

The setting of each channel n by this register is reflected at the timing when the timer output signal is set or reset while the timer output is enabled (TOE0n = 1).

The TOM0 register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the TOM0 register can be set with an 8-bit memory manipulation instruction with TOM0L.

Reset signal generation clears this register to 0000H.

Figure 6-18. Format of Timer Output Mode register 0 (TOM0)

Address: F01BEH, F01BFH After reset: 0000H R/W

20- and 24-pin products

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOM0	0	0	0	0	0	0	0	0	0	0	0	0	TOM03	TOM02	TOM01	0

30-pin products

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOM0	0	0	0	0	0	0	0	0	TOM07	TOM06	TOM05	TOM04	TOM03	TOM02	TOM01	0

TOM0n	Control of timer output mode of channel n
0	Master channel output mode (to produce toggle output by timer interrupt request signal (INTTM0n))
1	Slave channel output mode (output is set by the timer interrupt request signal (INTTM0n) of the master channel, and reset by the timer interrupt request signal (INTTM0p) of the slave channel)

Caution Be sure to clear undefined bits to "0".

Remark n: Channel number
n = 1 to 7 (n = 0, 2, 4, or 6 for master channel)
p: Slave channel number
n<p≤7
(For details of the relation between the master channel and slave channel, refer to **6.4.1 Basic Rules of Simultaneous Channel Operation Function**.)

Downloaded from Arrow.com.

6.3.13 Noise filter enable register 1 (NFEN1)

The NFEN1 register is used to set whether the noise filter can be used for the timer input signal to each channel.

Enable the noise filter by setting the corresponding bits to 1 on the pins in need of noise removal.

<R> When the noise filter is enabled, after synchronization with the operating clock (f_{MCK}) for the target channel, whether the signal keeps the same value for two clock cycles is detected.

When the noise filter is disabled, the input signal is only synchronized with the operating clock (f_{MCK}) for the target channel[Note].

The NFEN1 registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

<R> **Note** For details, see **6.5.1 (2) When valid edge of input signal via the TI0n pin is selected (CCS0n = 1)**, **6.5.2 Start timing of counter**, and **6.7 Timer Input (TI0n) Control**.

Figure 6-19. Format of Noise Filter Enable Register 1 (NFEN1)

Address: F0071H After reset: 00H R/W

20- and 24-pin products

Symbol	7	6	5	4	3	2	1	0
NFEN1	0	0	0	0	TNFEN03	TNFEN02	TNFEN01	TNFEN00

30-pin products

Symbol	7	6	5	4	3	2	1	0
NFEN1	TNFEN07	TNFEN06	TNFEN05	TNFEN04	TNFEN03	TNFEN02	TNFEN01	TNFEN00

TNFEN0n	Enable/disable using noise filter of TI0n pin input signal
0	Noise filter OFF
1	Noise filter ON

Remark The presence or absence of timer I/O pins of channel 0 to 7 depends on the product. See **Table 6-2 Timer I/O Pins in the Products** for details.

Downloaded from Arrow.com.

<R> **6.3.14 Registers controlling port functions of pins to be used for timer I/O**

Using port pins for the timer array unit functions requires setting of the registers that control the port functions multiplexed on the target pins (port mode register (PMxx), port register (Pxx), and port mode control register (PMCxx)). For details, see **4.3.1 Port mode registers (PMxx)**, **4.3.2 Port registers (Pxx)**, and **4.3.6 Port mode control registers (PMCxx)**.

The port mode register (PMxx), port register (Pxx), and port mode control register (PMCxx) to be set depend on the product. For details, see **4.5.3 Register setting examples for using the port and alternate functions**.

When using the ports that shares the pin with the timer output (such as TI00/TO00/P13) for timer output, set the port mode control register (PMCxx) bit, port mode register (PMxx) bit, and port register (Pxx) bit corresponding to each port to 0.

Example: When using P13/TI00/TO00 for timer output in 20-, 24-pin product
Set the PMC13 bit of port mode control register 1 to 0.
Set the PM00 bit of port mode register 1 to 0.
Set the P00 bit of port register 1 to 0.

When using the ports (such as TI00/TO00/P13) to be shared with the timer output pin for timer input, set the port mode register (PMxx) bit corresponding to each port to 1. Also set the port mode control register (PMCxx) bit corresponding to each port to 0. At this time, the port register (Pxx) bit may be 0 or 1.

Example: When using P13/TO00/TI00 for timer input in 20-, 24-pin product
Set the PMC13 bit of port mode control register 1 to 0.
Set the PM00 bit of port mode register 1 to 1.
Set the P00 bit of port register 1 to 0 or 1.

Downloaded from Arrow.com.

6.4 Basic Rules of Timer Array Unit

6.4.1 Basic Rules of Simultaneous Channel Operation Function

When simultaneously using multiple channels, namely, a combination of a master channel (a reference timer mainly counting the cycle) and slave channels (timers operating according to the master channel), the following rules apply.

(1) Only an even channel (channel 0, 2, 4, 6) can be set as a master channel.
(2) Any channel, except channel 0, can be set as a slave channel.
(3) The slave channel must be lower than the master channel.
Example: If channel 2 is set as a master channel, channel 3 or those that follow (Channel 3 to 7) can be set as a slave channel.

(4) Two or more slave channels can be set for one master channel.
(5) When two or more master channels are to be used, slave channels with a master channel between them may not be set.
Example: If channels 0 and 4 are set as master channels, channels 1 to 3 can be set as the slave channels of master channel 0. Channels 5 to 7 cannot be set as the slave channels of master channel 0.

(6) The operating clock for a slave channel in combination with a master channel must be the same as that of the master channel. The CKS0n0 and CKS0n1 bits (bits 15 and 14 of timer mode register 0n (TMR0n)) of the slave channel that operate in combination with the master channel must be the same value as that of the master channel.
(7) A master channel can transmit INTTM0n (interrupt), start software trigger, and count clock to the lower channels.
(8) A slave channel can use INTTM0n (interrupt), a start software trigger, or the count clock of the master channel as a source clock, but cannot transmit its own INTTM0n (interrupt), start software trigger, or count clock to channels with lower channel numbers.
(9) A master channel cannot use INTTM0n (interrupt), a start software trigger, or the count clock from the other higher master channel as a source clock.
(10) To simultaneously start channels that operate in combination, the channel start trigger bit (TS0n) of the channels in combination must be set at the same time.
(11) During the counting operation, a TS0n bit of a master channel or TS0n bits of all channels which are operating simultaneously can be set. It cannot be applied to TS0n bits of slave channels alone.
(12) To stop the channels in combination simultaneously, the channel stop trigger bit (TT0n) of the channels in combination must be set at the same time.
(13) CK02/CK03 cannot be selected while channels are operating simultaneously, because the operating clocks of master channels and slave channels have to be synchronized.
(14) Timer mode register 0n (TMR0n) has no master bit (it is fixed as “0”). However, as channel 0 is the highest channel, it can be used as a master channel during simultaneous operation.

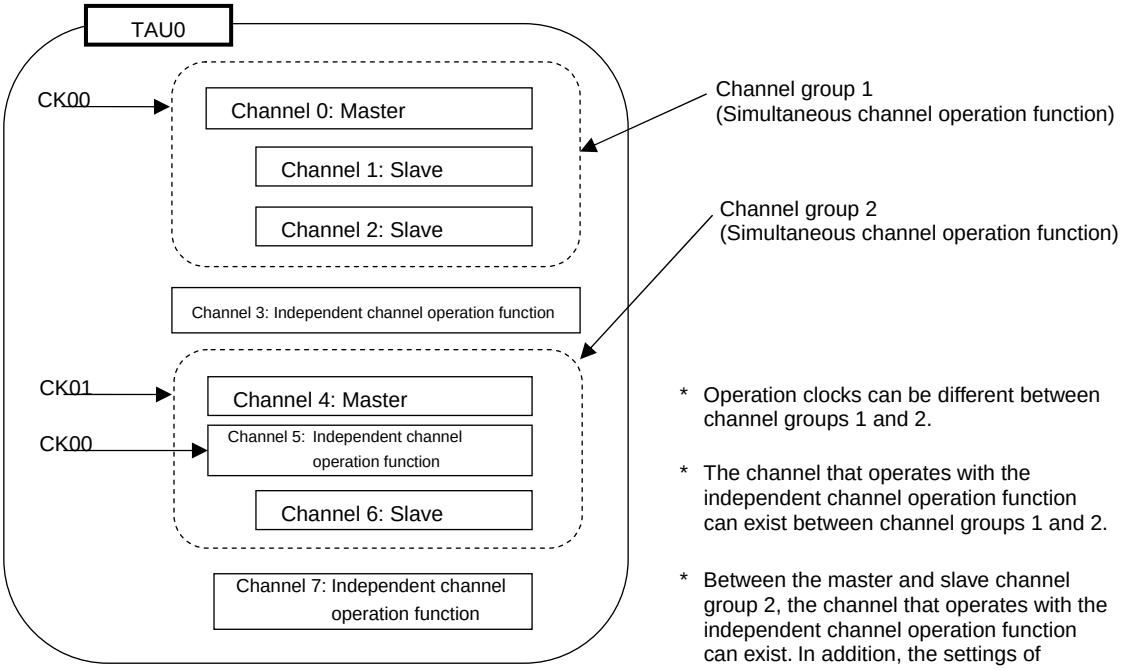
The rules of the simultaneous channel operation function are applied in a channel group (a master channel and slave channels forming one simultaneous channel operation function).

If two or more channel groups that do not operate in combination are specified, the basic rules of the simultaneous channel operation function in **6.4.1 Basic rules of simultaneous channel operation function** do not apply to the channel groups.

Remark n: Channel number (n = 0 to 7)

Downloaded from Arrow.com.

Example



* Operation clocks can be different between channel groups 1 and 2.
* The channel that operates with the independent channel operation function can exist between channel groups 1 and 2.
* Between the master and slave channel group 2, the channel that operates with the independent channel operation function can exist. In addition, the settings of operation clocks can be performed individually.

Downloaded from Arrow.com.

6.4.2 Basic rules of 8-bit timer operation function (Only Channels 1 and 3)

The 8-bit timer operation function makes it possible to use a 16-bit timer channel in a configuration consisting of two 8-bit timer channels.

This function can only be used for channels 1 and 3, and there are several rules for using it.

The basic rules for this function are as follows:

(1) The 8-bit timer operation function applies only to channels 1 and 3.
(2) When using 8-bit timers, set the SPLIT bit of timer mode register 0n (TMR0n) to 1.
(3) The higher 8 bits can be operated as the interval timer function.
(4) At the start of operation, the higher 8 bits output INTTM01H/INTTM03H (an interrupt) (which is the same operation performed when MD0n0 is set to 1).
(5) The operation clock of the higher 8 bits is selected according to the CKS0n1 and CKS0n0 bits of the lower-bit TMR0n register.
(6) For the higher 8 bits, the TSH01/TSH03 bit is manipulated to start channel operation and the TTH01/TTH03 bit is manipulated to stop channel operation. The channel status can be checked using the TEH01/TEH03 bit.
(7) The lower 8 bits operate according to the TMR0n register settings. The following three functions support operation of the lower 8 bits:
- Interval timer function
- External event counter function
- Delay count function

(8) For the lower 8 bits, the TS01/TS03 bit is manipulated to start channel operation and the TT01/TT03 bit is manipulated to stop channel operation. The channel status can be checked using the TE01/TE03 bit.
(9) During 16-bit operation, manipulating the TSH01/TSH03/TTH01/TTH03 bits is invalid. The TS01/TS03/TT01/TT03 bits are manipulated to operate channels 1 and 3. The TEH03 and TEH01 bits are not changed.
(10) For the 8-bit timer function, the simultaneous operation functions (one-shot pulse, PWM, and multiple PWM) cannot be used.

Remark n: Channel number (n = 1, 3)

Downloaded from Arrow.com.

6.5 Operation of Counter

6.5.1 Count clock (f_{TCLK})

The count clock (f_{TCLK}) of the timer array unit can be selected between following by CCS0n bit of timer mode register 0n (TMR0n).

- Operation clock (f_{MCK}) specified by the CKS0n0 and CKS0n1 bits
- Valid edge of input signal input from the TI0n pin

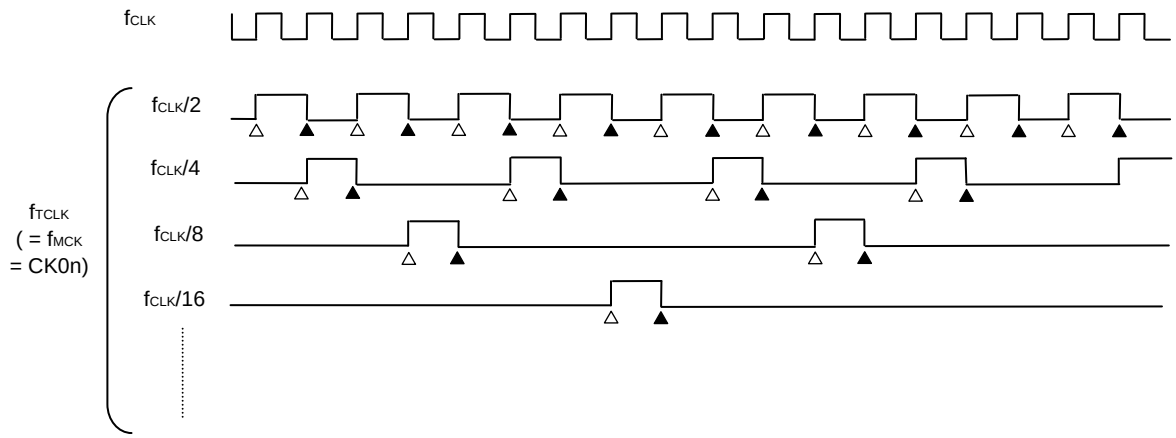
Because the timer array unit is designed to operate in synchronization with f_{CLK}, the timings of the count clock (f_{TCLK}) are shown below.

(1) When operation clock (f_{MCK}) specified by the CKS0n0 and CKS0n1 bits is selected (CCS0n = 0)

The count clock (f_{TCLK}) is between f_{CLK} to $f_{CLK}/2^{15}$ by setting of timer clock select register 0 (TPS0). When a divided f_{CLK} is selected, however, the clock selected in TPSmn register, but a signal which becomes high level for one period of f_{CLK} from its rising edge. When a f_{CLK} is selected, fixed to high level

Counting of timer count register 0n (TCR0n) delayed by one period of f_{CLK} from rising edge of the count clock, because of synchronization with f_{CLK}. But, this is described as "counting at rising edge of the count clock", as a matter of convenience.

Figure 6-20. Timing of f_{CLK} and count clock (f_{TCLK}) (When CCS0n = 0)



Remarks 1. △: Rising edge of the count clock
▲: Synchronization, increment/decrement of counter

2. f_{CLK}: CPU/peripheral hardware clock

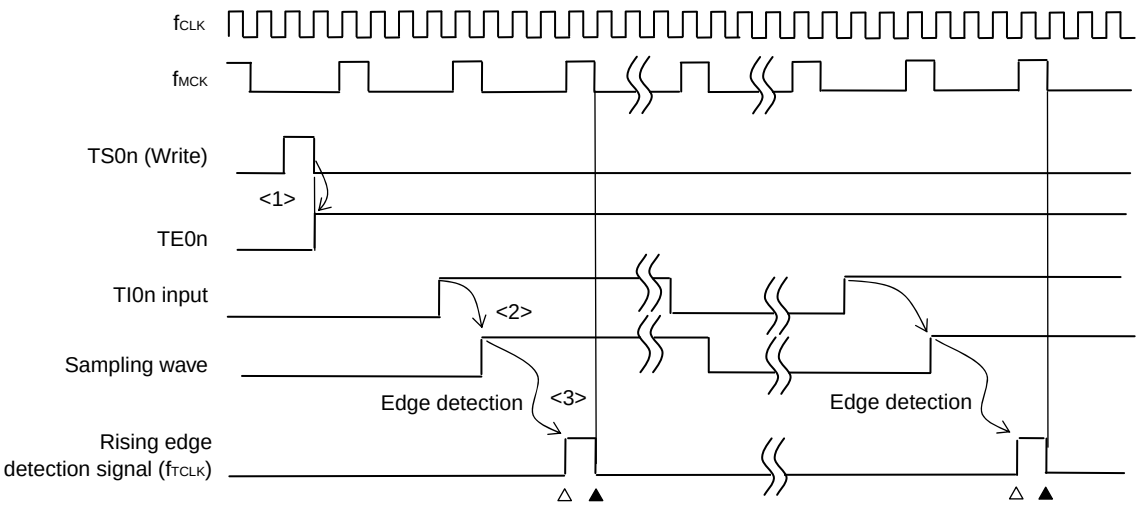
Downloaded from Arrow.com.

(2) When valid edge of input signal via the TI0n pin is selected (CCS0n = 1)

The count clock (f_{TCLK}) becomes the signal that detects valid edge of input signal via the TI0n pin and synchronizes next rising f_{MCK}. The count clock (f_{TCLK}) is delayed for 1 to 2 period of f_{MCK} from the input signal via the TI0n pin (when a noise filter is used, the delay becomes 3 to 4 clock).

Counting of timer count register 0n (TCR0n) delayed by one period of f_{CLK} from rising edge of the count clock, because of synchronization with f_{CLK}. But, this is described as "counting at valid edge of input signal via the TI0n pin", as a matter of convenience.

Figure 6-21. Timing of f_{CLK} and count clock (f_{TCLK}) (When CCS0n = 1, noise filter unused)



<1> Setting TS0n bit to 1 enables the timer to be started and to become wait state for valid edge of input signal via the TI0n pin.

<2> The rise of input signal via the TI0n pin is sampled by f_{MCK}.

<3> The edge is detected by the rising of the sampled signal and the detection signal (count clock) is output.

Remarks 1. △ : Rising edge of the count clock
▲ : Synchronization, increment/decrement of counter

2. f_{CLK}: CPU/peripheral hardware clock
f_{MCK}: Operation clock of channel n

3. The waveform of the input signal via TI0n pin of the input pulse interval measurement, the measurement of high/low width of input signal, and the delay counter, the one-shot pulse output are the same as that shown in **Figure 6-21**.

Downloaded from Arrow.com.

6.5.2 Start timing of counter

Timer count register 0n (TCR0n) becomes enabled to operation by setting of TS0n bit of timer channel start register 0 (TS0).

Operations from count operation enabled state to timer count Register 0n (TCR0n) count start is shown in **Table 6-6**.

Table 6-6. Operations from Count Operation Enabled State to Timer count Register 0n (TCR0n) Count Start

Timer operation mode	Operation when TS0n = 1 is set
• Interval timer mode	No operation is carried out from start trigger detection (TS0n = 1) until count clock generation. The first count clock loads the value of the TDR0n register to the TCR0n register and the subsequent count clock performs count down operation (see **6.5.3 (1) Interval timer mode operation**).
• Event counter mode	Writing 1 to the TS0n bit loads the value of the TDR0n register to the TCR0n register. Detection TI0n input edge, the subsequent count clock performs count down operation. (see **6.5.3 (2) Event counter mode operation**).
• Capture mode	No operation is carried out from start trigger (TS0n = 1) detection until count clock generation. The first count clock loads 0000H to the TCR0n register and the subsequent count clock performs count up operation (see **6.5.3 (3) Capture mode operation (input pulse interval measurement)**).
• One-count mode	The waiting-for-start-trigger state is entered by writing 1 to the TS0n bit while the timer is stopped (TE0n = 0). No operation is carried out from start trigger detection until count clock generation. The first count clock loads the value of the TDR0n register to the TCR0n register and the subsequent count clock performs count down operation (see **6.5.3 (4) One-count mode operation**).
• Capture & one-count mode	The waiting-for-start-trigger state is entered by writing 1 to the TS0n bit while the timer is stopped (TE0n = 0). No operation is carried out from start trigger detection until count clock generation. The first count clock loads 0000H to the TCR0n register and the subsequent count clock performs count up operation (see **6.5.3 (5) Capture & one-count mode operation (high-level width is measured)**).

Downloaded from Arrow.com.

6.5.3 Counter Operation

Here, the counter operation in each mode is explained.

(1) Interval timer mode operation

<1> Operation is enabled (TE0n = 1) by writing 1 to the TS0n bit. Timer count register 0n (TCR0n) holds the initial value until count clock generation.

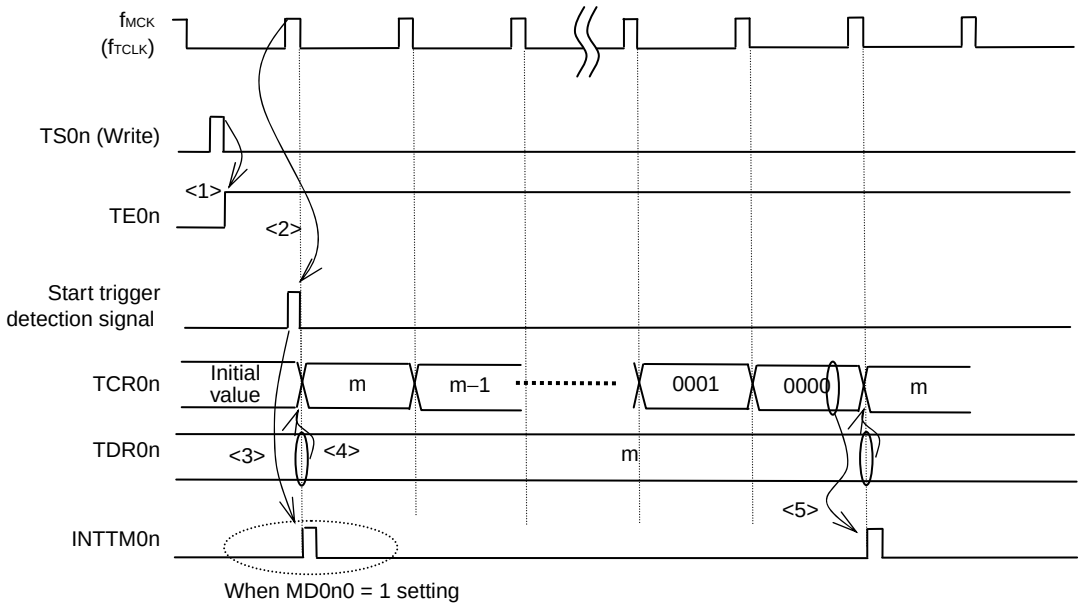
<2> A start trigger is generated at the first count clock (f_{MCK}) after operation is enabled.

<3> When the MD0n0 bit is set to 1, INTTM0n is generated by the start trigger.

<4> By the first count clock after the operation enable, the value of timer data register 0n (TDR0n) is loaded to the TCR0n register and counting starts in the interval timer mode.

<5> When the TCR0n register counts down and its count value is 0000H, INTTM0n is generated in the next count clock (f_{MCK}) and the value of timer data register 0n (TDR0n) is loaded to the TCR0n register and counting keeps on.

Figure 6-22. Operation Timing (In Interval Timer Mode)



Caution In the first cycle operation of count clock after writing the TS0n bit, an error at a maximum of one clock is generated since count start delays until count clock has been generated. When the information on count start timing is necessary, an interrupt can be generated at count start by setting MD0n0 = 1.

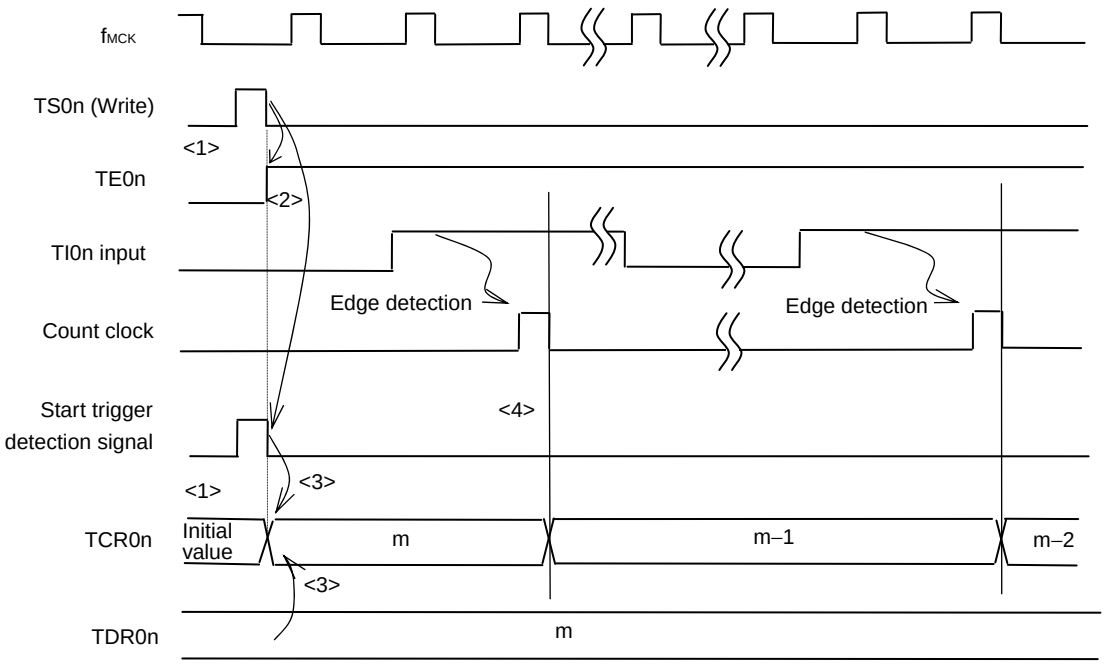
Remark f_{MCK}, the start trigger detection signal, and INTTM0n become active between one clock in synchronization with f_{CLK}.

Downloaded from Arrow.com.

(2) Event counter mode operation

<1> Timer count register 0n (TCR0n) holds its initial value while operation is stopped (TE0n = 0).
<2> Operation is enabled (TE0n = 1) by writing 1 to the TS0n bit.
<3> As soon as 1 has been written to the TS0n bit and 1 has been set to the TE0n bit, the value of timer data register 0n (TDR0n) is loaded to the TCR0n register to start counting.
<4> After that, the TCR0n register value is counted down according to the count clock of the valid edge of the TI0n input.

Figure 6-23. Operation Timing (In Event Counter Mode)



Remark The timing is shown in **Figure 6-23** indicates while the noise filter is not used. By making the noise filter on-state, the edge detection becomes 2 f_{MCK} cycles (it sums up to 3 to 4 cycles) later than the normal cycle of TI0n input.
The error per one period occurs be the asynchronous between the period of the TI0n input and that of the count clock (f_{MCK}).

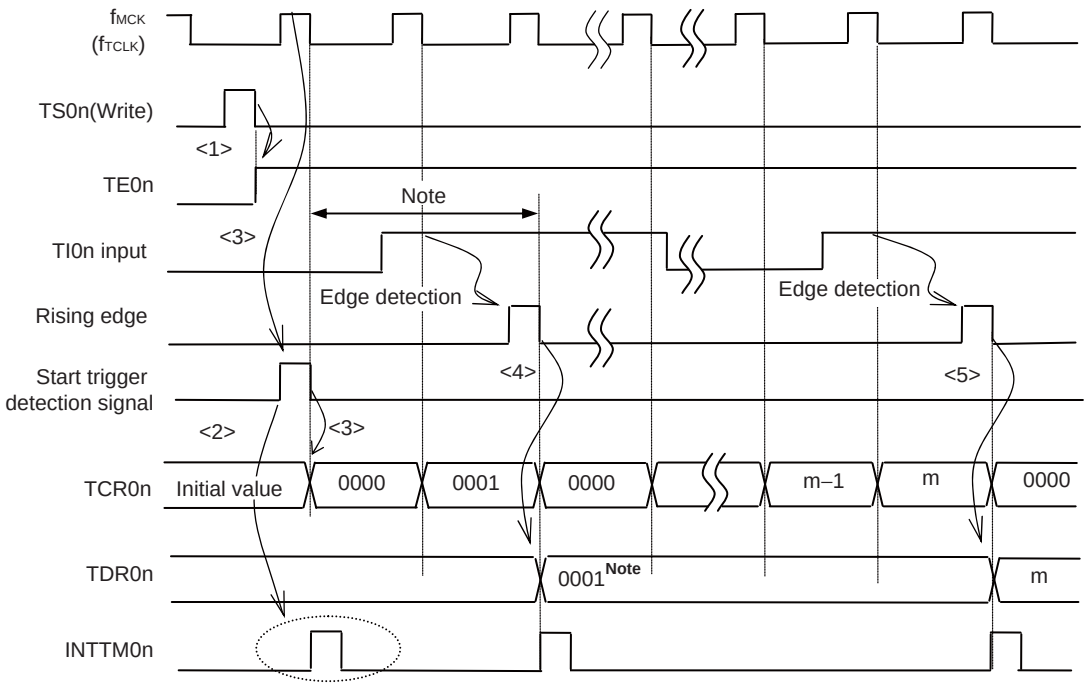
Downloaded from Arrow.com.

(3) Capture mode operation (input pulse interval measurement)

<1> Operation is enabled (TE0n = 1) by writing 1 to the TS0n bit.
<2> Timer count register 0n (TCR0n) holds the initial value until count clock generation.
<3> A start trigger is generated at the first count clock after operation is enabled. And the value of 0000H is loaded to the TCR0n register and counting starts in the capture mode. (When the MD0n0 bit is set to 1, INTTM0n is generated by the start trigger.)
<4> On detection of the valid edge of the TI0n input, the value of the TCR0n register is captured to timer data register 0n (TDR0n) and INTTM0n is generated. However, this capture value is nomeaning. The TCR0n register keeps on counting from 0000H.
<5> On next detection of the valid edge of the TI0n input, the value of the TCR0n register is captured to timer data register 0n (TDR0n) and INTTM0n is generated.

Figure 6-24. Operation Timing (In Capture Mode: Input Pulse Interval Measurement)

<R>



When MD0n0 = 1 setting

Note If a clock has been input to TI0n (the trigger exists) when capturing starts, counting starts when a trigger is detected, even if no edge is detected. Therefore, the first captured value (<4>) does not determine a pulse interval (in the above figure, 0001 just indicates two clock cycles but does not determine the pulse interval) and so the user can ignore it.

Caution In the first cycle operation of count clock after writing the TS0n bit, an error at a maximum of one clock is generated since count start delays until count clock has been generated. When the information on count start timing is necessary, an interrupt can be generated at count start by setting MD0n0 = 1.

Remark The timing is shown in **Figure 6-24** indicates while the noise filter is not used. By making the noise filter on-state, the edge detection becomes 2 fMCK cycles (it sums up to 3 to 4 cycles) later than the normal cycle of TI0n input.
The error per one period occurs be the asynchronous between the period of the TI0n input and that of the count clock (fMCK).

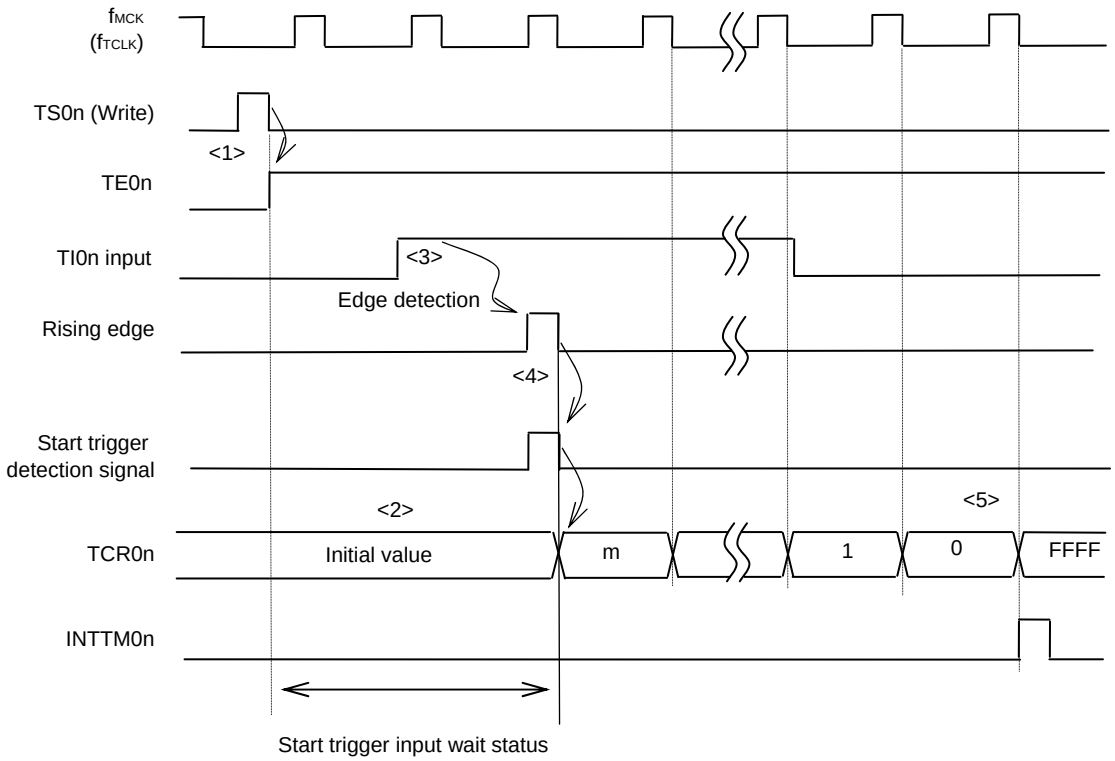
Downloaded from Arrow.com.

(4) One-count mode operation

<1> Operation is enabled (TE0n = 1) by writing 1 to the TS0n bit.
<2> Timer count register 0n (TCR0n) holds the initial value until start trigger generation.
<3> Rising edge of the TI0n input is detected.
<4> On start trigger detection, the value of timer data register 0n (TDR0n) is loaded to the TCR0n register and count starts.
<5> When the TCR0n register counts down and its count value is 0000H, INTTM0n is generated and the value of the TCR0n register becomes FFFFH and counting stops

.

Figure 6-25. Operation Timing (In One-count Mode)



<R>

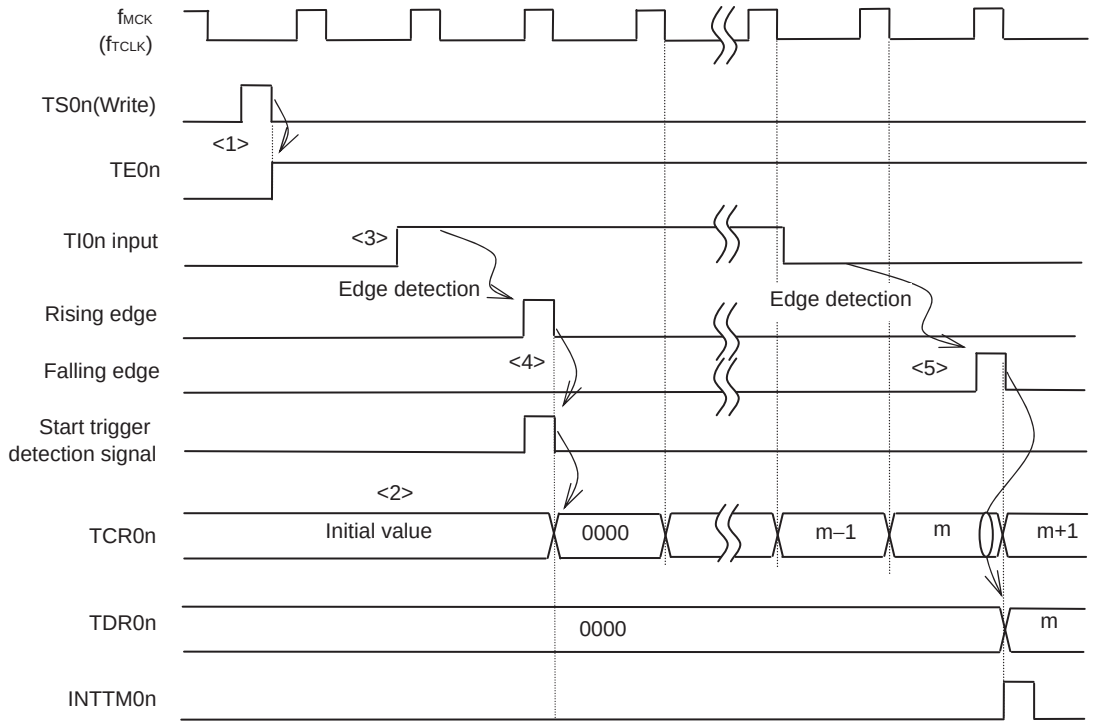
Remark The timing is shown in **Figure 6-25** indicates while the noise filter is not used. By making the noise filter on-state, the edge detection becomes 2 fMCK cycles (it sums up to 3 to 4 cycles) later than the normal cycle of TI0n input. The error per one period occurs be the asynchronous between the period of the TI0n input and that of the count clock (fMCK).

Downloaded from Arrow.com.

(5) Capture & one-count mode operation (high-level width is measured)

<1> Operation is enabled (TE0n = 1) by writing 1 to the TS0n bit of timer channel start register 0 (TS0).
<2> Timer count register 0n (TCR0n) holds the initial value until start trigger generation.
<3> Rising edge of the TI0n input is detected.
<4> On start trigger detection, the value of 0000H is loaded to the TCR0n register and count starts.
<5> On detection of the falling edge of the TI0n input, the value of the TCR0n register is captured to timer data register 0n (TDR0n) and INTTM0n is generated.

Figure 6-26. Operation Timing (In Capture & One-count Mode: High-level Width Measurement)



<R>

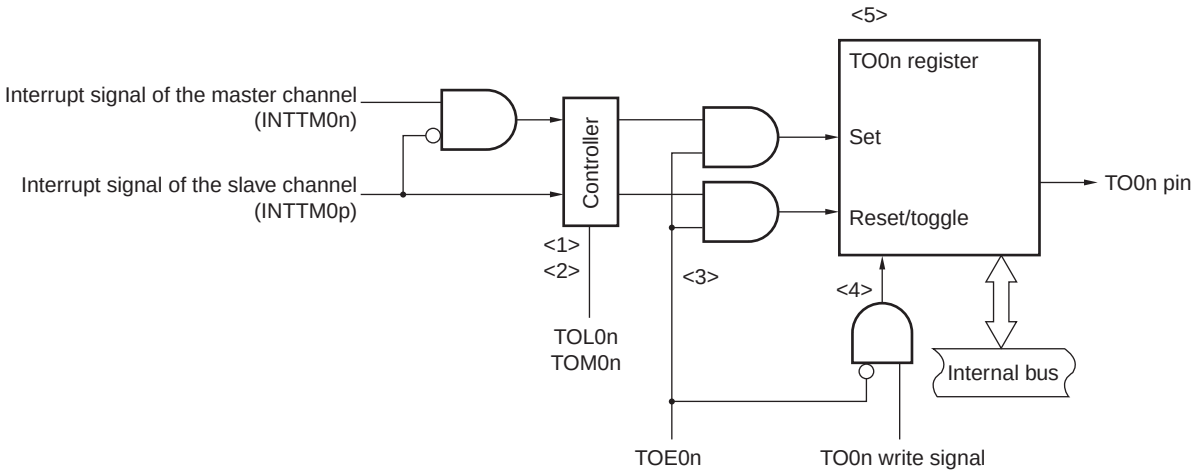
Remark The timing is shown in **Figure 6-28** indicates while the noise filter is not used. By making the noise filter on-state, the edge detection becomes 2 f_{MCK} cycles (it sums up to 3 to 4 cycles) later than the normal cycle of TI0n input. The error per one period occurs be the asynchronous between the period of the TI0n input and that of the count clock (f_{MCK}).

Downloaded from Arrow.com.

6.6 Channel Output (TO0n pin) Control

6.6.1 TO0n pin output circuit configuration

Figure 6-27. Output Circuit Configuration



The following describes the TO0n pin output circuit.

<1> When TOM0n = 0 (master channel output mode), the set value of timer output level register 0 (TOL0) is ignored and only INTTM0p (slave channel timer interrupt) is transmitted to timer output register 0 (TO0).

<2> When TOM0n = 1 (slave channel output mode), both INTTM0n (master channel timer interrupt) and INTTM0p (slave channel timer interrupt) are transmitted to the TO0 register.
At this time, the TOL0 register becomes valid and the signals are controlled as follows:

When TOL0n = 0: Positive logic output (INTTM0n → set, INTTM0p → reset)
When TOL0n = 1: Negative logic output (INTTM0n → reset, INTTM0p → set)

When INTTM0n and INTTM0p are simultaneously generated, (0% output of PWM), INTTM0p (reset signal) takes priority, and INTTM0n (set signal) is masked.

<3> While timer output is enabled (TOE0n = 1), INTTM0n (master channel timer interrupt) and INTTM0p (slave channel timer interrupt) are transmitted to the TO0 register. Writing to the TO0 register (TO0n write signal) becomes invalid.
When TOE0n = 1, the TO0n pin output never changes with signals other than interrupt signals.
To initialize the TO0n pin output level, it is necessary to set timer operation is stopped (TOE0n = 0) and to write a value to the TO0 register.

<4> While timer output is disabled (TOE0n = 0), writing to the TO0n bit to the target channel (TO0n write signal) becomes valid. When timer output is disabled (TOE0n = 0), neither INTTM0n (master channel timer interrupt) nor INTTM0p (slave channel timer interrupt) is transmitted to the TO0 register.

<5> The TO0 register can always be read, and the TO0n pin output level can be checked.

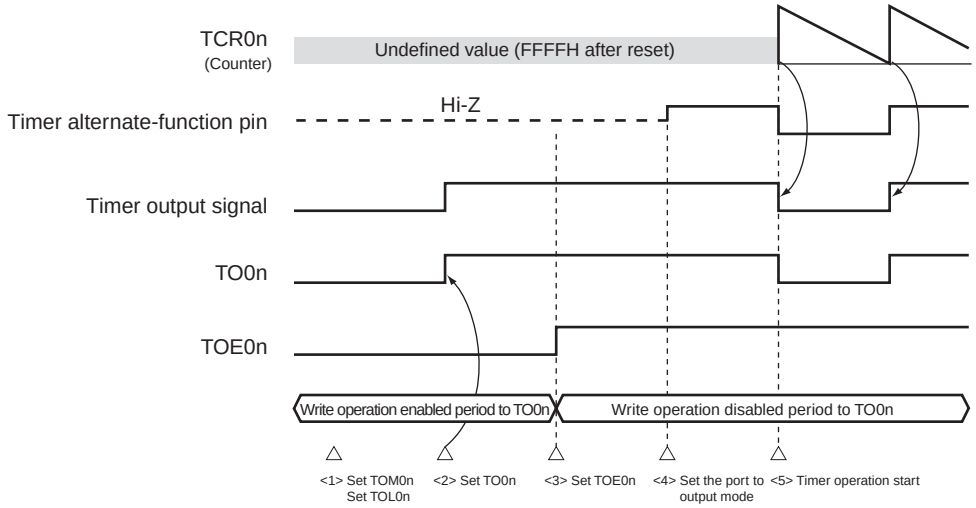
Remark n: Channel number
n = 0 to 7 (n = 0, 2, 4, or 6 for master channel)
p: Slave channel number
n<p≤7

Downloaded from Arrow.com.

6.6.2 TO0n Pin Output Setting

The following figure shows the procedure and status transition of the TO0n output pin from initial setting to timer operation start.

Figure 6-28. Status Transition from Timer Output Setting to Operation Start



<1> The operation mode of timer output is set.

- TOM0n bit (0: Master channel output mode, 1: Slave channel output mode)
- TOL0n bit (0: Positive logic output, 1: Negative logic output)

<2> The timer output signal is set to the initial status by setting timer output register 0 (TO0).
<3> The timer output operation is enabled by writing 1 to the TOE0n bit (writing to the TO0 register is disabled).
<4> The port is set to digital I/O by port mode control register (PMCxx) (see **6.3.14 Registers controlling port functions of pins to be used for timer I/O**).
<5> The port I/O setting is set to output (see **6.3.14 Registers controlling port functions of pins to be used for timer I/O**).
<6> The timer operation is enabled (TS0n = 1).

Remark n: Channel number (n = 0 to 7)

Downloaded from Arrow.com.

6.6.3 Cautions on Channel Output Operation

(1) Changing values set in the registers TO0, TOE0, and TOL0 during timer operation

Since the timer operations (operations of timer count register 0n (TCR0n) and timer data register 0n (TDR0n)) are independent of the TO0n output circuit and changing the values set in timer output register 0 (TO0), timer output enable register 0 (TOE0), timer output level register 0 (TOL0) does not affect the timer operation, the values can be changed during timer operation. To output an expected waveform from the TO0n pin by timer operation, however, set the TO0, TOE0, TOL0, and TOM0 registers to the values stated in the register setting example of each operation.
When the values set to the TOE0 and TOL0 registers (but not the TO0 register) are changed close to the occurrence of the timer interrupt (INTTM0n) of each channel, the waveform output to the TO0n pin might differ, depending on whether the values are changed immediately before or immediately after the timer interrupt (INTTM0n) occurs.

Remark n: Channel number (n = 0 to 7)

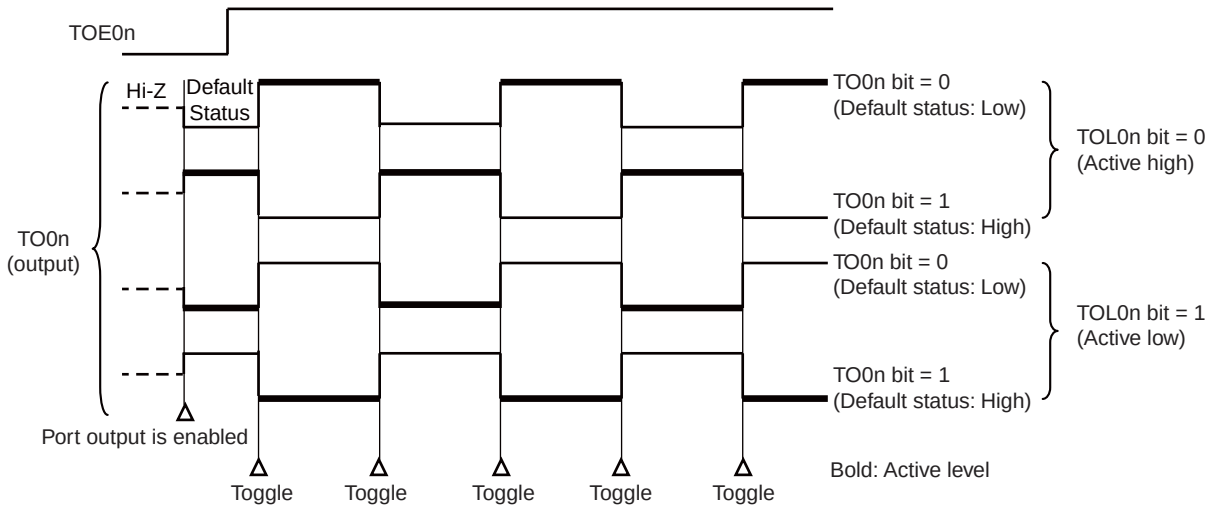
(2) Default level of TO0n pin and output level after timer operation start

The change in the output level of the TO0n pin when timer output register 0 (TO0) is written while timer output is disabled (TOE0n = 0), the initial level is changed, and then timer output is enabled (TOE0n = 1) before port output is enabled, is shown below.

(a) When operation starts with master channel output mode (TOM0n = 0) setting

The setting of timer output level register 0 (TOL0) is invalid when master channel output mode (TOM0n = 0).
When the timer operation starts after setting the default level, the toggle signal is generated and the output level of the TO0n pin is reversed.

Figure 6-29. TO0n Pin Output Status at Toggle Output (TOM0n = 0)



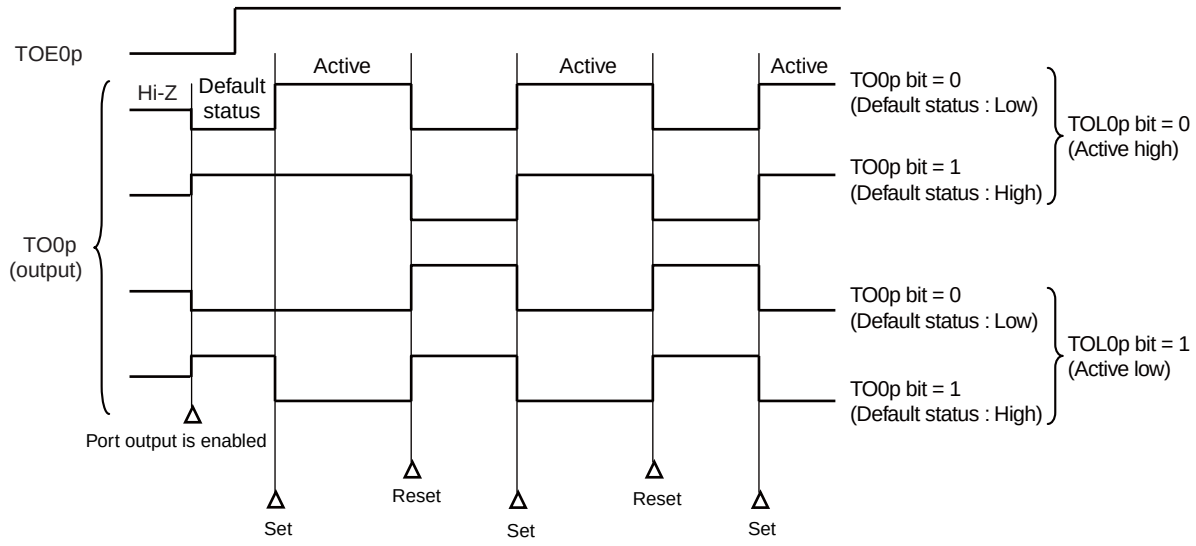
Remarks **1.** Toggle: Reverse TO0n pin output status
2. n: Channel number (n = 0 to 7)

Downloaded from Arrow.com.

(b) When operation starts with slave channel output mode (TOM0p = 1) setting (PWM output))

When slave channel output mode (TOM0p = 1), the active level is determined by timer output level register 0 (TOL0p) setting.

Figure 6-30. TOM0p Pin Output Status at PWM Output (TO0p = 1)



Remarks 1. Set: The output signal of the TO0p pin changes from inactive level to active level.
Reset: The output signal of the TO0p pin changes from active level to inactive level.

2. p: Channel number (n < p ≤ 7)

Downloaded from Arrow.com.

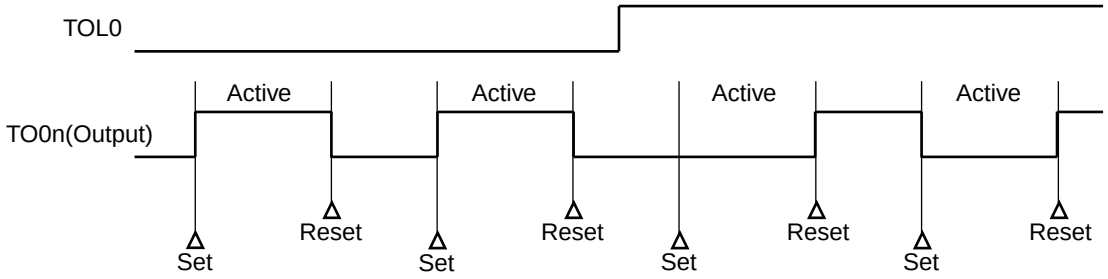
(3) Operation of TO0n pin in slave channel output mode (TOM0n = 1)

(a) When timer output level register 0 (TOL0) setting has been changed during timer operation

When the TOL0 register setting has been changed during timer operation, the setting becomes valid at the generation timing of the TO0n pin change condition. Rewriting the TOL0 register does not change the output level of the TO0n pin.

The operation when TOM0n is set to 1 and the value of the TOL0 register is changed while the timer is operating (TE0n = 1) is shown below.

Figure 6-31. Operation when TOL0 Register Has Been Changed during Timer Operation



Remarks 1. Set: The output signal of the TO0p pin changes from inactive level to active level.
Reset: The output signal of the TO0p pin changes from active level to inactive level.

2. n: Channel number (n = 0 to 7)

(b) Set/reset timing

To realize 0%/100% output at PWM output, the TO0n pin/TO0n bit set timing at master channel timer interrupt (INTTM0n) generation is delayed by 1 count clock by the slave channel.

If the set condition and reset condition are generated at the same time, a higher priority is given to the latter.

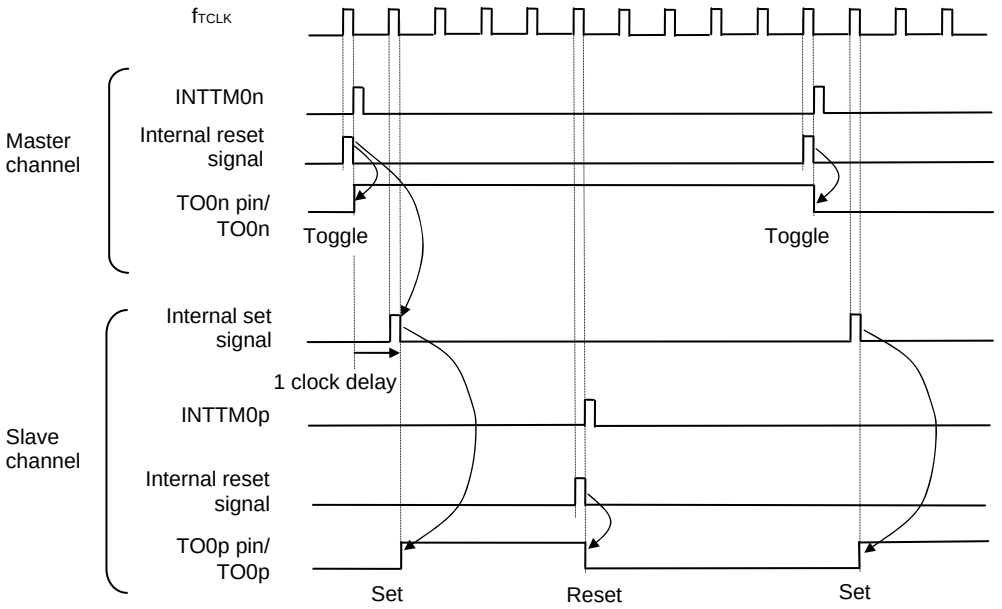
Figure 6-32 shows the set/reset operating statuses where the master/slave channels are set as follows.

Master channel: TOE0n = 1, TOM0n = 0, TOL0n = 0
Slave channel: TOE0p = 1, TOM0p = 1, TOL0p = 0

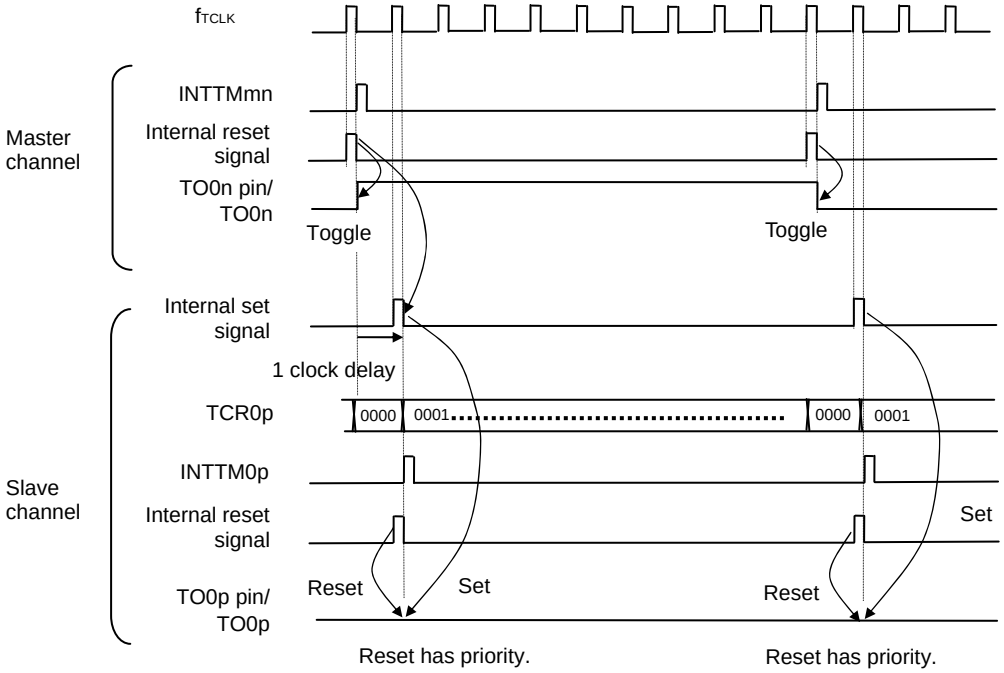
Downloaded from Arrow.com.

Figure 6-32. Set/Reset Timing Operating Statuses

(a) Basic operation timing



(b) Operation timing when 0 % duty



Remarks 1. Internal reset signal: TO0n pin reset/toggle signal
Internal set signal: TO0n pin set signal

2. n: Channel number (n = 0 to 7)
n = 0 to 7 (n = 0, 2, 4, 6 for master channel)
p: Slave channel number
n < p ≤ 7

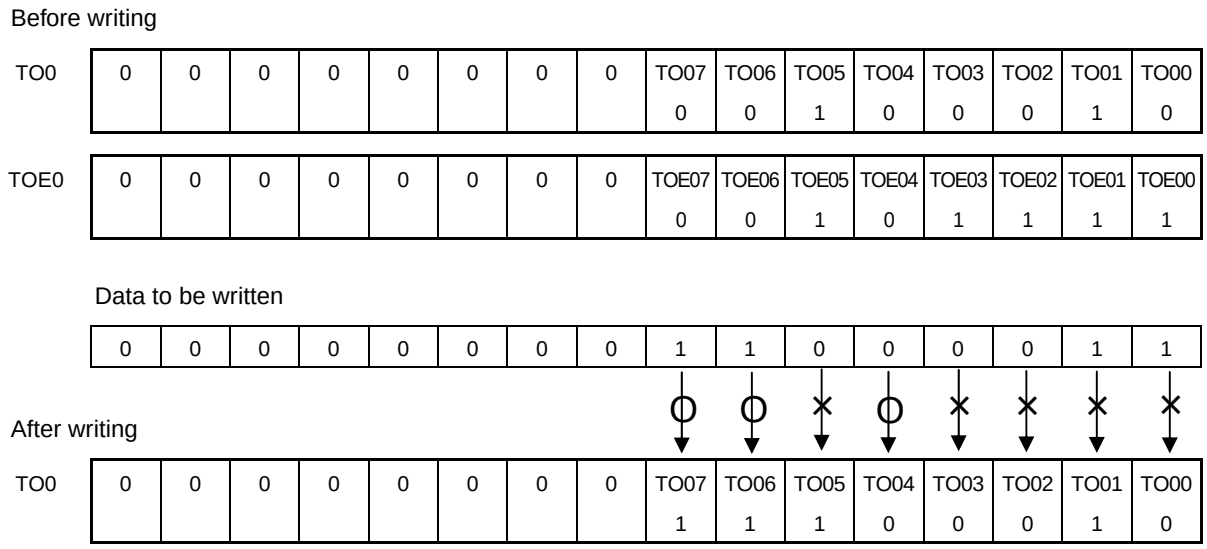
Downloaded from Arrow.com.

6.6.4 Collective manipulation of TO0n bit

In timer output register 0 (TO0), the setting bits for all the channels are located in one register in the same way as timer channel start register 0 (TS0). Therefore, the TO0n bit of all the channels can be manipulated collectively.

Only the desired bits can also be manipulated by enabling writing only to the TO0n bits (TOE0n = 0) that correspond to the relevant bits of the channel used to perfor0 output (TO0n).

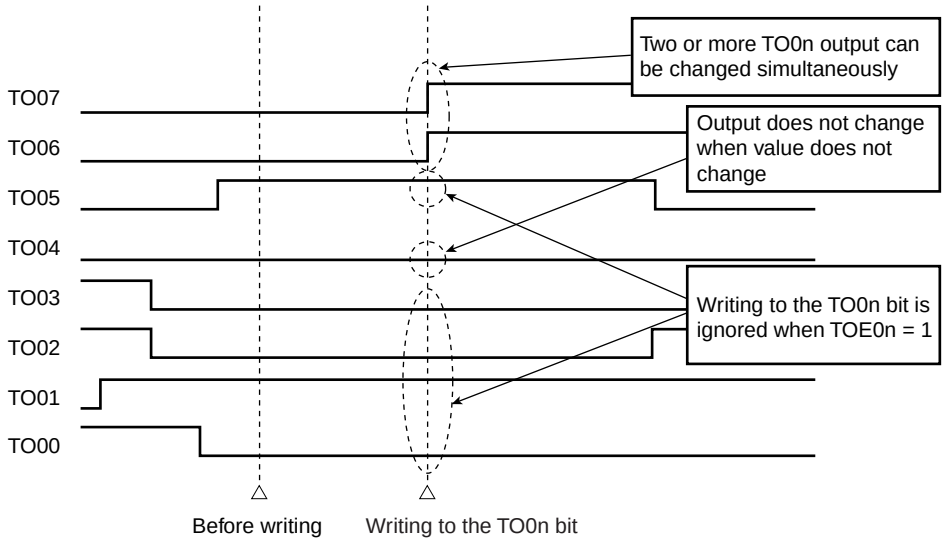
Figure 6-33. Example of TO0n Bit Collective Manipulation



Writing is done only to the TO0n bit with TOE0n = 0, and writing to the TO0n bit with TOE0n = 1 is ignored.

TO0n (channel output) to which TOE0n = 1 is set is not affected by the write operation. Even if the write operation is done to the TO0n bit, it is ignored and the output change by timer operation is normally done.

Figure 6-34. TO0n Pin Statuses by Collective Manipulation of TO0n Bit



Caution While timer output is enabled (TOE0n = 1), even if the output by timer interrupt of each timer (INTTM0n) contends with writing to the TO0n bit, output is normally done to the TO0n pin.

Remark n: Channel number (n = 0 to 7)

Downloaded from Arrow.com.

6.6.5 Timer Interrupt and TO0n Pin Output at Operation Start

In the interval timer mode or capture mode, the MD0n0 bit in timer mode register 0n (TMR0n) sets whether or not to generate a timer interrupt at count start.

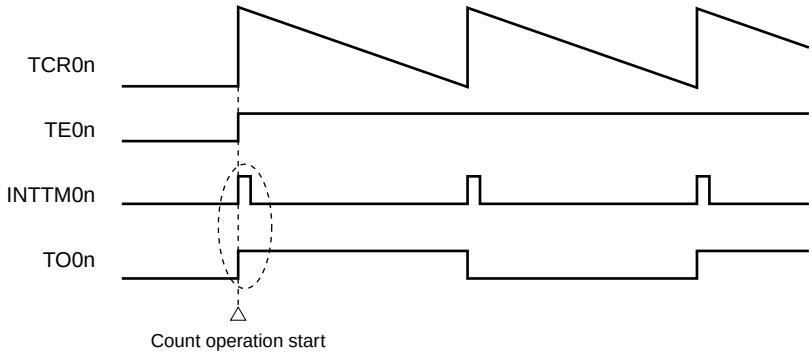
When MD0n0 is set to 1, the count operation start timing can be known by the timer interrupt (INTTM0n) generation.

In the other modes, neither timer interrupt at count operation start nor TO0n output is controlled.

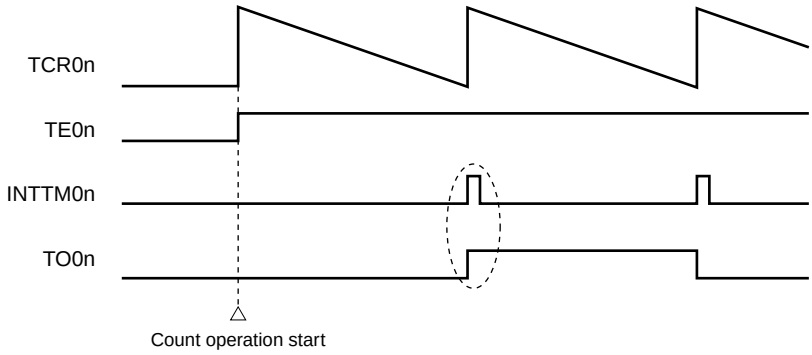
Figure 6-37 shows operation examples when the interval timer mode (TOE0n = 1, TOM0n = 0) is set.

Figure 6-35. Operation examples of timer interrupt at count operation start and TO0n output

(a) When MD0n0 is set to 1



(b) When MD0n0 is set to 0



When MD0n0 is set to 1, a timer interrupt (INTTM0n) is output at count operation start, and TO0n performs a toggle operation.

When MD0n0 is set to 0, a timer interrupt (INTTM0n) is not output at count operation start, and TO0n does not change either. After counting one cycle, INTTM0n is output and TO0n performs a toggle operation.

Remark n: Channel number (n = 0 to 7)

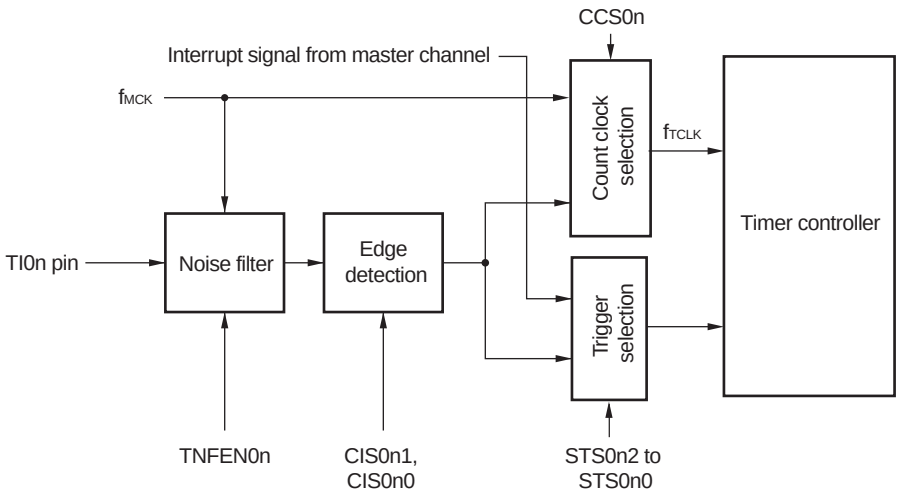
Downloaded from Arrow.com.

<R> 6.7 Timer Input (TI0n) Control

6.7.1 TI0n input circuit configuration

A signal is input from a timer input pin, goes through a noise filter and an edge detector, and is sent to a timer controller. Enable the noise filter for the pin in need of noise removal. The following shows the configuration of the input circuit.

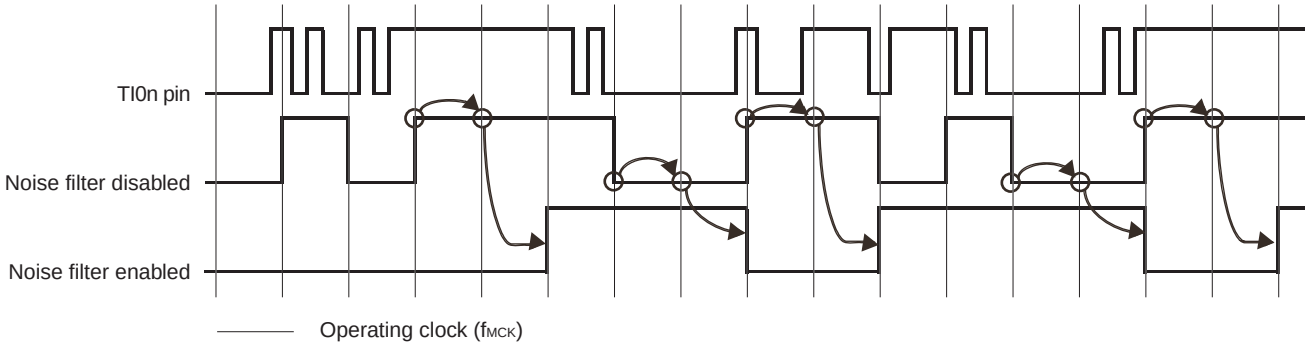
Figure 6-36. Input Circuit Configuration



6.7.2 Noise filter

When the noise filter is disabled, the input signal is only synchronized with the operating clock (f_{MCK}) for channel n. When the noise filter is enabled, after synchronization with the operating clock (f_{MCK}) for channel n, whether the signal keeps the same value for two clock cycles is detected. The following shows differences in waveforms output from the noise filter between when the noise filter is enabled and disabled.

Figure 6-37. Sampling Waveforms through TI0n Input Pin with Noise Filter Enabled and Disabled



Downloaded from Arrow.com.

6.7.3 Cautions on channel input operation

When a timer input pin is set as unused, the operating clock is not supplied to the noise filter. Therefore, after settings are made to use the timer input pin, the following wait time is necessary before a trigger is specified to enable operation of the channel corresponding to the timer input pin.

(1) Noise filter is disabled

When bits 12 (CCS0n), 9 (STS0n1), and 8 (STS0n0) in the timer mode register 0n (TMR0n) are all 0 and then one of them is set to 1, wait for at least two cycles of the operating clock (f_{MCK}), and then set the operation enable trigger bit in the timer channel start register (TS0).

(2) Noise filter is enabled

When bits 12 (CCS0n), 9 (STS0n1), and 8 (STS0n0) in the timer mode register 0n (TMR0n) are all 0 and then one of them is set to 1, wait for at least four cycles of the operating clock (f_{MCK}), and then set the operation enable trigger bit in the timer channel start register (TS0).

Downloaded from Arrow.com.

6.8 Independent Channel Operation Function of Timer Array Unit

6.8.1 Operation as interval timer/square wave output

(1) Interval timer

The timer array unit can be used as a reference timer that generates INTTM0n (timer interrupt) at fixed intervals. The interrupt generation period can be calculated by the following expression.

Generation period of INTTM0n (timer interrupt) = Period of count clock × (Set value of TDR0n + 1)

(2) Operation as square wave output

TO0n performs a toggle operation as soon as INTTM0n has been generated, and outputs a square wave with a duty factor of 50%.

The period and frequency for outputting a square wave from TO0n can be calculated by the following expressions.

- Period of square wave output from TO0n = Period of count clock × (Set value of TDR0n + 1) × 2
- Frequency of square wave output from TO0n = Frequency of count clock/{(Set value of TDR0n + 1) × 2}

Timer count register 0n (TCR0n) operates as a down counter in the interval timer mode.

The TCR0n register loads the value of timer data register 0n (TDR0n) at the first count clock after the channel start trigger bit (TS0n, TSH01, TSH03) of timer channel start register 0 (TS0) is set to 1. If the MD0n0 bit of timer mode register 0n (TMR0n) is 0 at this time, INTTM0n is not output and TO0n is not toggled. If the MD0n0 bit of the TMR0n register is 1, INTTM0n is output and TO0n is toggled.

After that, the TCR0n register count down in synchronization with the count clock.

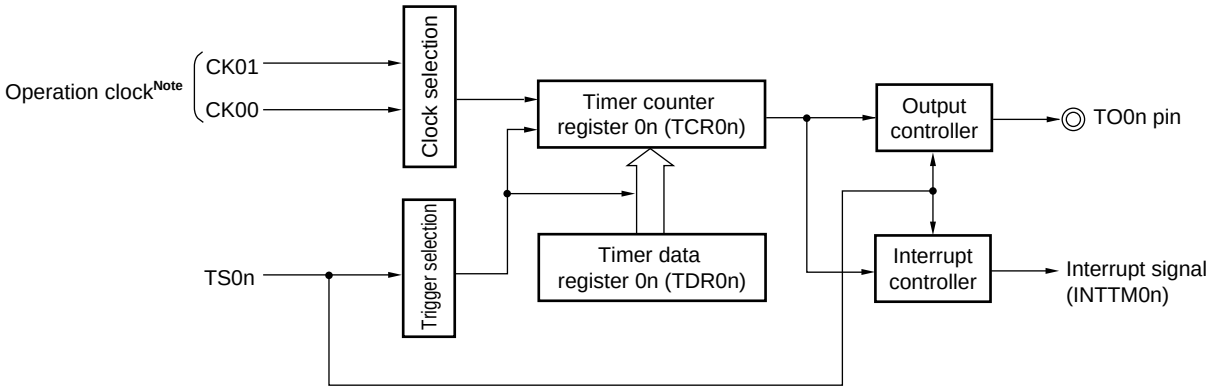
When TCR0n = 0000H, INTTM0n is output and TO0n is toggled at the next count clock. At the same time, the TCR0n register loads the value of the TDR0n register again. After that, the same operation is repeated.

The TDR0n register can be rewritten at any time. The new value of the TDR0n register becomes valid from the next period.

Remark n: Channel number (n = 0 to 7)

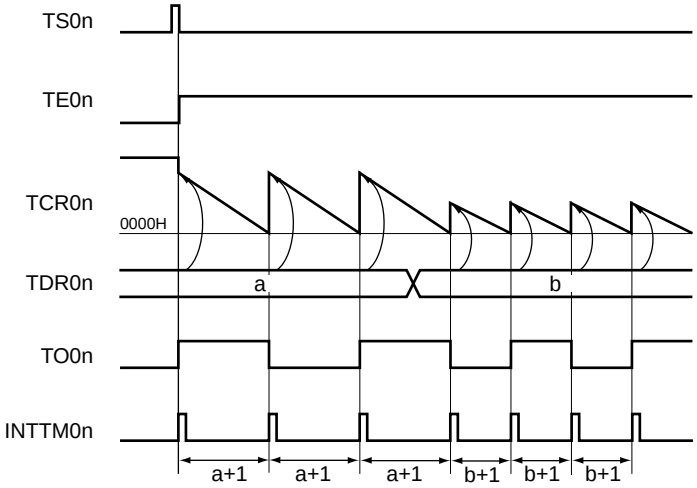
Downloaded from Arrow.com.

Figure 6-38. Block Diagram of Operation as Interval Timer/Square Wave Output



Note When channels 1 and 3, the clock can be selected from CK00 to CK03.

Figure 6-39. Example of Basic Timing of Operation as Interval Timer/Square Wave Output (MD0n0 = 1)



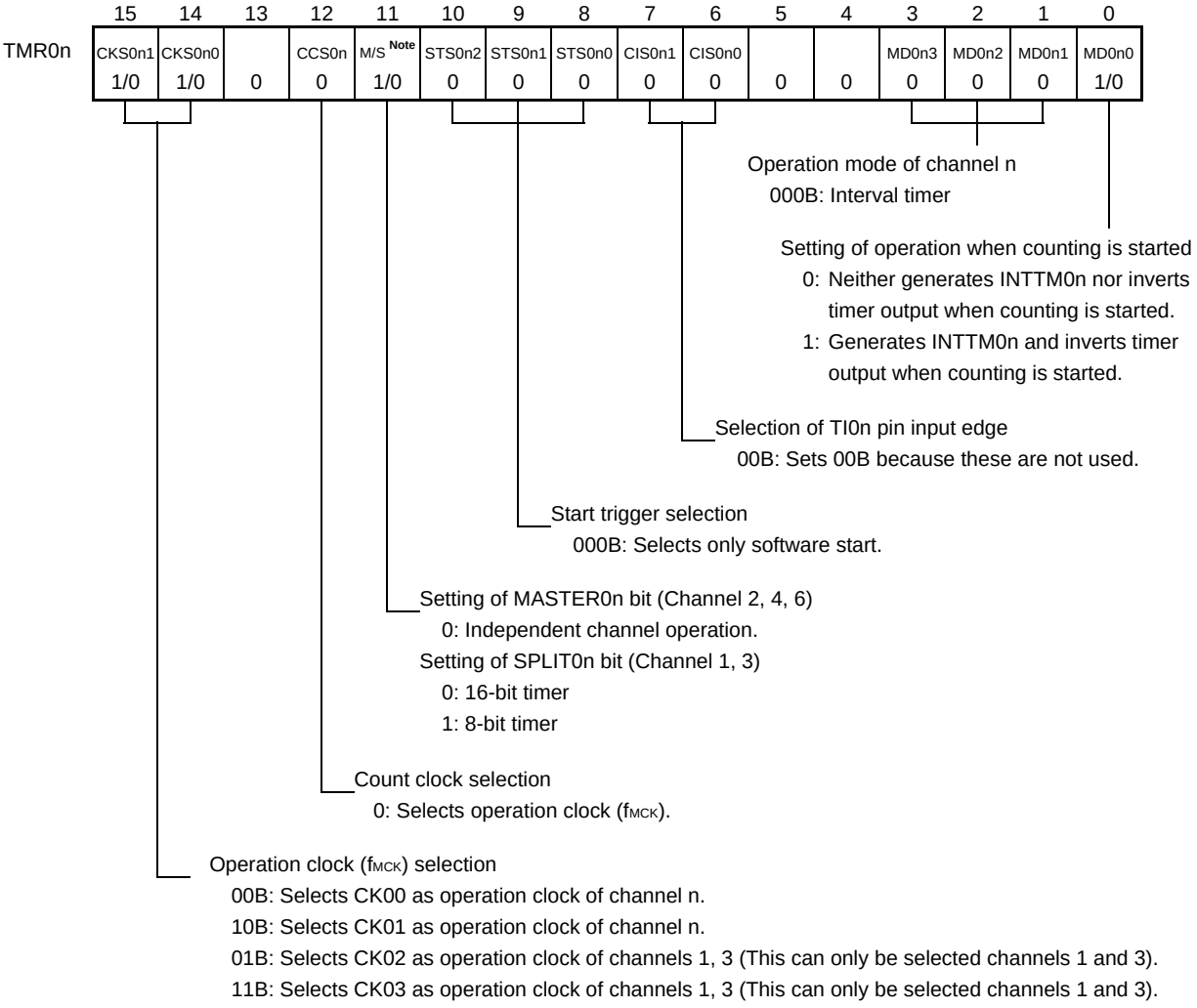
Remarks 1. n: Channel number (n = 0 to 7)

2. TS0n: Bit n of timer channel start register 0 (TS0)
TE0n: Bit n of timer channel enable status register 0 (TE0)
TCR0n: Timer count register 0n (TCR0n)
TDR0n: Timer data register 0n (TDR0n)
TO0n: TO0n pin output signal

Downloaded from Arrow.com.

Figure 6-40. Example of Set Contents of Registers During Operation as Interval Timer/Square Wave Output (1/2)

(a) Timer mode register 0n (TMR0n)



<R>

(b) Timer output register 0 (TO0)

	Bit n	
TO0	TO0n 1/0	0: Outputs 0 from TO0n. 1: Outputs 1 from TO0n.

(c) Timer output enable register 0 (TOE0)

	Bit n	
TOE0	TOE0n 1/0	0: Stops the TO0n output operation by counting operation. 1: Enables the TO0n output operation by counting operation.

Note TMR02, TMR04, TMR06: MASTER0n bit
TMR01, TMR03: SPLIT0n bit
TMR00, TMR05, TMR07: 0 fixed

Remark n: Channel number (n = 0 to 7)

Downloaded from Arrow.com.

Figure 6-40. Example of Set Contents of Registers During Operation as Interval Timer/Square Wave Output (2/2)

(d) Timer output level register 0 (TOL0)

	Bit n	
TOL0	TOL0n 0	0: Cleared to 0 when master channel output mode (TOM0n = 0)

(e) Timer output mode register 0 (TOM0)

	Bit n	
TOM0	TOM0n 0	0: Sets master channel output mode.

Remark n: Channel number (n = 0 to 7)

Downloaded from Arrow.com.

Figure 6-41. Operation Procedure of Interval Timer/Square Wave Output Function (1/2)

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN bit of peripheral enable register 0 (PER0) to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register 0 (TPS0). Determines clock frequencies of CK00 and CK01 (or CK02 and CK03 when using the 8-bit timer mode).	
Channel default setting	Sets timer mode register 0n (TMR0n) (determines operation mode of channel). Sets interval (period) value to timer data register 0n (TDR0n).	Channel stops operating. (Clock is supplied and some power is consumed.)
	To use the TO0n output Clears the TOM0n bit of timer output mode register 0 (TOM0) to 0 (master channel output mode). Clears the TOL0n bit to 0. Sets the TO0n bit and determines default level of the TO0n output. →	The TO0n pin goes into Hi-Z output state. The TO0n default setting level is output when the port mode register is in the output mode and the port register is 0.
	Sets the TOE0n bit to 1 and enables operation of TO0n. → Clears the port register and port mode register to 0. →	TO0n does not change because channel stops operating. The TO0n pin outputs the TO0n set level.
Operation start	(Sets the TOE0n bit to 1 only if using TO0n output and resuming operation.). Sets the TS0n (TSH01, TSH03) bit to 1. → The TS0n (TSH01, TSH03) bit automatically returns to 0 because it is a trigger bit.	TE0n (TEH01, TEH03) = 1, and count operation starts. Value of the TDR0n register is loaded to timer count register 0n (TCR0n) at the count clock input. INTTM0n is generated and TO0n performs toggle operation if the MD0n0 bit of the TMR0n register is 1.
During operation	Set value of the TDR0n register can be changed. The TCR0n register can always be read. The TSR0n register is not used. Set values of the TMR0n register, TOM0n, and TOL0n bits cannot be changed.	Counter (TCR0n) counts down. When count value reaches 0000H, the value of the TDR0n register is loaded to the TCR0n register again and the count operation is continued. By detecting TCR0n = 0000H, INTTM0n is generated and TO0n performs toggle operation. After that, the above operation is repeated.
Operation stop	The TT0n (TTH01, TTH03) bit is set to 1. → The TT0n (TTH01, TTH03) bit automatically returns to 0 because it is a trigger bit.	TE0n (TEH01, TEH03), and count operation stops. The TCR0n register holds count value and stops. The TO0n output is not initialized but holds current status.
	The TOE0n bit is cleared to 0 and value is set to the TO0n bit. →	The TO0n pin outputs the TO0n bit set level.
TAU stop	To hold the TO0n pin output level Clears the TO0n bit to 0 after the value to be held is set to the port register. → When holding the TO0n pin output level is not necessary Setting not required.	The TO0n pin output level is held by port function.
	The TAU0EN bit of the PER0 register is cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TO0n bit is cleared to 0 and the TO0n pin is set to port mode.)

<R> Operation is resumed.

Remark n: Channel number (n = 0 to 7)

Downloaded from Arrow.com.

6.8.2 Operation as external event counter

The timer array unit can be used as an external event counter that counts the number of times the valid input edge (external event) is detected in the TI0n pin. When a specified count value is reached, the event counter generates an interrupt. The specified number of counts can be calculated by the following expression.

Specified number of counts = Set value of TDR0n + 1

Timer count register 0n (TCR0n) operates as a down counter in the event counter mode.

The TCR0n register loads the value of timer data register 0n (TDR0n) by setting any channel start trigger bit (TS0n, TSH01, TSH03) of timer channel start register 0 (TS0) to 1.

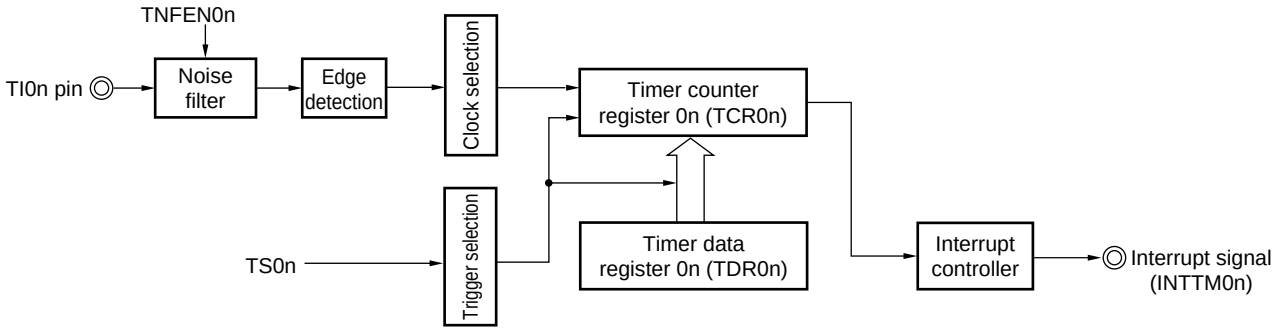
The TCR0n register counts down each time the valid input edge of the TI0n pin has been detected. When TCR0n = 0000H, the TCR0n register loads the value of the TDR0n register again, and outputs INTTM0n.

After that, the above operation is repeated.

An irregular waveform that depends on external events is output from the TO0n pin. Stop the output by setting the TOE0n bit of timer output enable register 0 (TOE0) to 0.

The TDR0n register can be rewritten at any time. The new value of the TDR0n register becomes valid during the next count period.

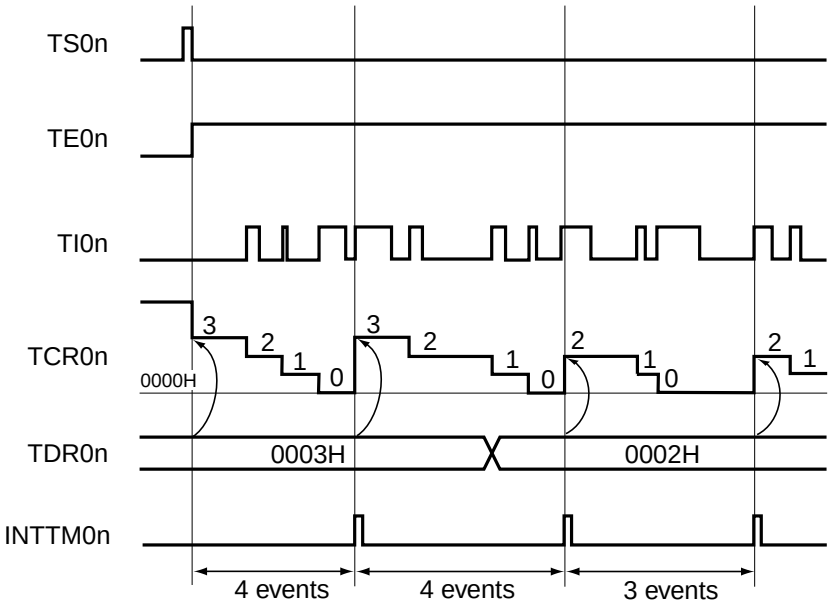
Figure 6-42. Block Diagram of Operation as External Event Counter



Remark n: Channel number (n = 0 to 7)

Downloaded from Arrow.com.

Figure 6-43. Example of Basic Timing of Operation as External Event Counter



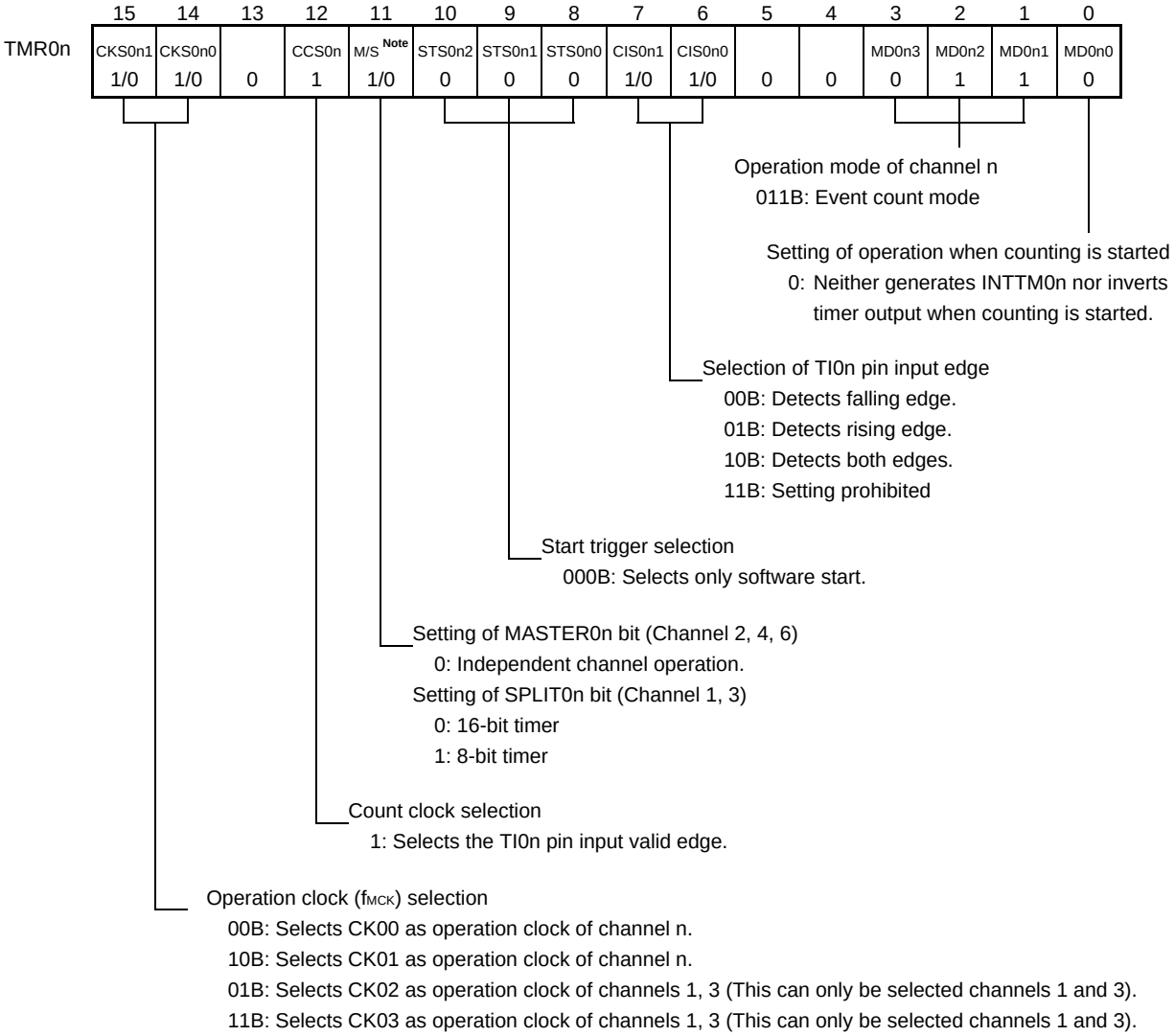
Remarks 1. n: Channel number (n = 0 to 7)

2. TS0n: Bit n of timer channel start register 0 (TS0)
TE0n: Bit n of timer channel enable status register 0 (TE0)
TI0n: TI0n pin input signal
TCR0n: Timer count register 0n (TCR0n)
TDR0n: Timer data register 0n (TDR0n)

Downloaded from Arrow.com.

Figure 6-44. Example of Set Contents of Registers in External Event Counter Mode (1/2)

(a) Timer mode register 0n (TMR0n)



<R>

(b) Timer output register 0 (TO0)

	Bit n	
TO0	TO0n	0: Outputs 0 from TO0n.
	0	

(c) Timer output enable register 0 (TOE0)

	Bit n	
TOE0	TOE0n	0: Stops the TO0n output operation by counting operation.
	0	

Note TMR02, TMR04, TMR06: MASTER0n bit
TMR01, TMR03: SPLIT0n bit
TMR00, TMR05, TMR07: 0 fixed

Remark n: Channel number (n = 0 to 7)

Downloaded from Arrow.com.

Figure 6-44. Example of Set Contents of Registers in External Event Counter Mode (2/2)

(d) Timer output level register 0 (TOL0)

	Bit n	
TOL0	TOL0n 0	0: Cleared to 0 when master channel output mode (TOM0n = 0).

(e) Timer output mode register 0 (TOM0)

	Bit n	
TOM0	TOM0n 0	0: Sets master channel output mode.

Remark n: Channel number (n = 0 to 7)

Downloaded from Arrow.com.

Figure 6-45. Operation Procedure When External Event Counter Function Is Used

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN bit of peripheral enable register 0 (PER0) to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register 0 (TPS0). Determines clock frequencies of CK00 and CK01 (or CK02 and CK03 when using the 8-bit timer mode).	
<R> Channel default setting	Sets the corresponding bit of the noise filter enable registers 1 (NFEN1) to 0 (off) or 1 (on). Sets timer mode register 0n (TMR0n) (determines operation mode of channel). Sets number of counts to timer data register 0n (TDR0n). Sets noise filter enable register 1 (NFEN1) Clears the TOE0n bit of timer output enable register 0 (TOE0) to 0.	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	Sets the TS0n bit to 1. → The TS0n bit automatically returns to 0 because it is a trigger bit.	TE0n = 1, and count operation starts. Value of the TDR0n register is loaded to timer count register 0n (TCR0n) and detection of the TI0n pin input edge is awaited.
During operation	Set value of the TDR0n register can be changed. The TCR0n register can always be read. The TSR0n register is not used. Set values of the TMR0n register, TOM0n, TOL0n, TO0n, and TOE0n bits cannot be changed.	Counter (TCR0n) counts down each time input edge of the TI0n pin has been detected. When count value reaches 0000H, the value of the TDR0n register is loaded to the TCR0n register again, and the count operation is continued. By detecting TCR0n = 0000H, the INTTM0n output is generated. After that, the above operation is repeated.
Operation stop	The TT0n bit is set to 1. → The TT0n bit automatically returns to 0 because it is a trigger bit.	TE0n = 0, and count operation stops. The TCR0n register holds count value and stops.
TAU stop	The TAU0EN bit of the PER0 register is cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized.

Operation is resumed.

Remark n: Channel number (n = 0 to 7)

Downloaded from Arrow.com.

6.8.3 Operation as frequency divider (channel 0 of 30-pin products only)

The timer array unit can be used as a frequency divider that divides a clock input to the TI00 pin and outputs the result from the TO00 pin.

The divided clock frequency output from TO00 can be calculated by the following expression.

- When rising edge/falling edge is selected:
  Divided clock frequency = Input clock frequency/{(Set value of TDR00 + 1) × 2}
- When both edges are selected:
  Divided clock frequency ≅ Input clock frequency/(Set value of TDR00 + 1)

Timer count register 00 (TCR00) operates as a down counter in the interval timer mode.

After the channel start trigger bit (TS00) of timer channel start register 0 (TS0) is set to 1, the TCR00 register loads the value of timer data register 00 (TDR00) when the TI00 valid edge is detected.

If the MD000 bit of timer mode register 00 (TMR00) is 0 at this time, INTTM00 is not output and TO00 is not toggled. If the MD000 bit of timer mode register 00 (TMR00) is 1, INTTM00 is output and TO00 is toggled.

After that, the TCR00 register counts down at the valid edge of the TI00 pin. When TCR00 = 0000H, it toggles TO00. At the same time, the TCR00 register loads the value of the TDR00 register again, and continues counting.

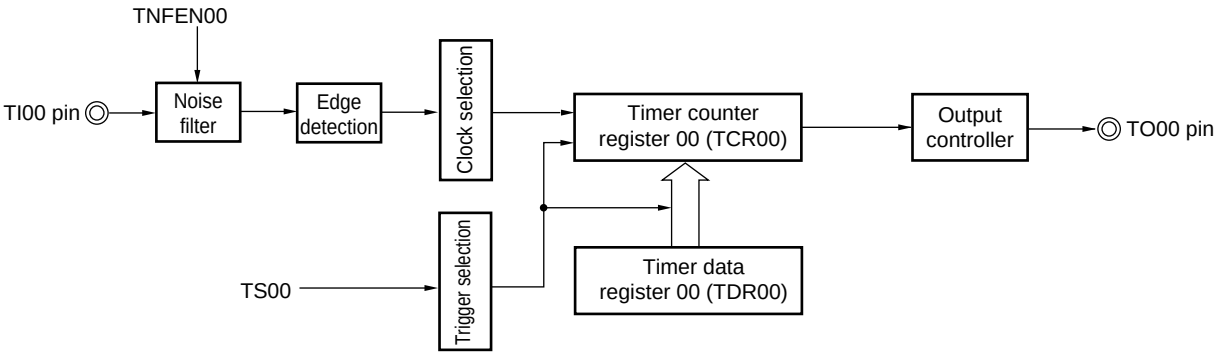
If detection of both the edges of the TI00 pin is selected, the duty factor error of the input clock affects the divided clock period of the TO00 output.

The period of the TO00 output clock includes a sampling error of one period of the operation clock.

Clock period of TO00 output = Ideal TO00 output clock period ± Operation clock period (error)

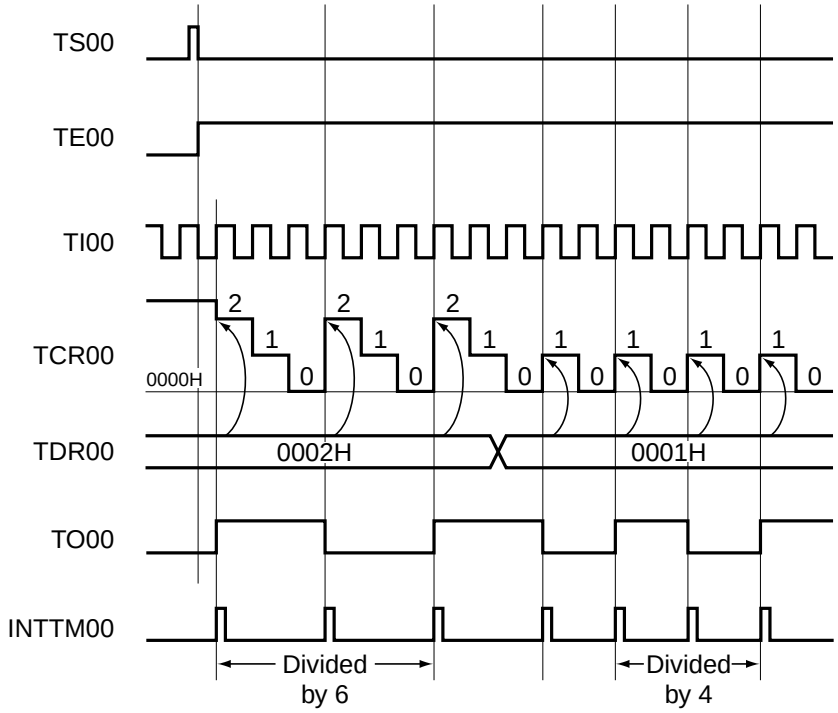
The TDR00 register can be rewritten at any time. The new value of the TDR00 register becomes valid during the next count period.

Figure 6-46. Block Diagram of Operation as Frequency Divider



Downloaded from Arrow.com.

Figure 6-47. Example of Basic Timing of Operation as Frequency Divider (MD000 = 1)

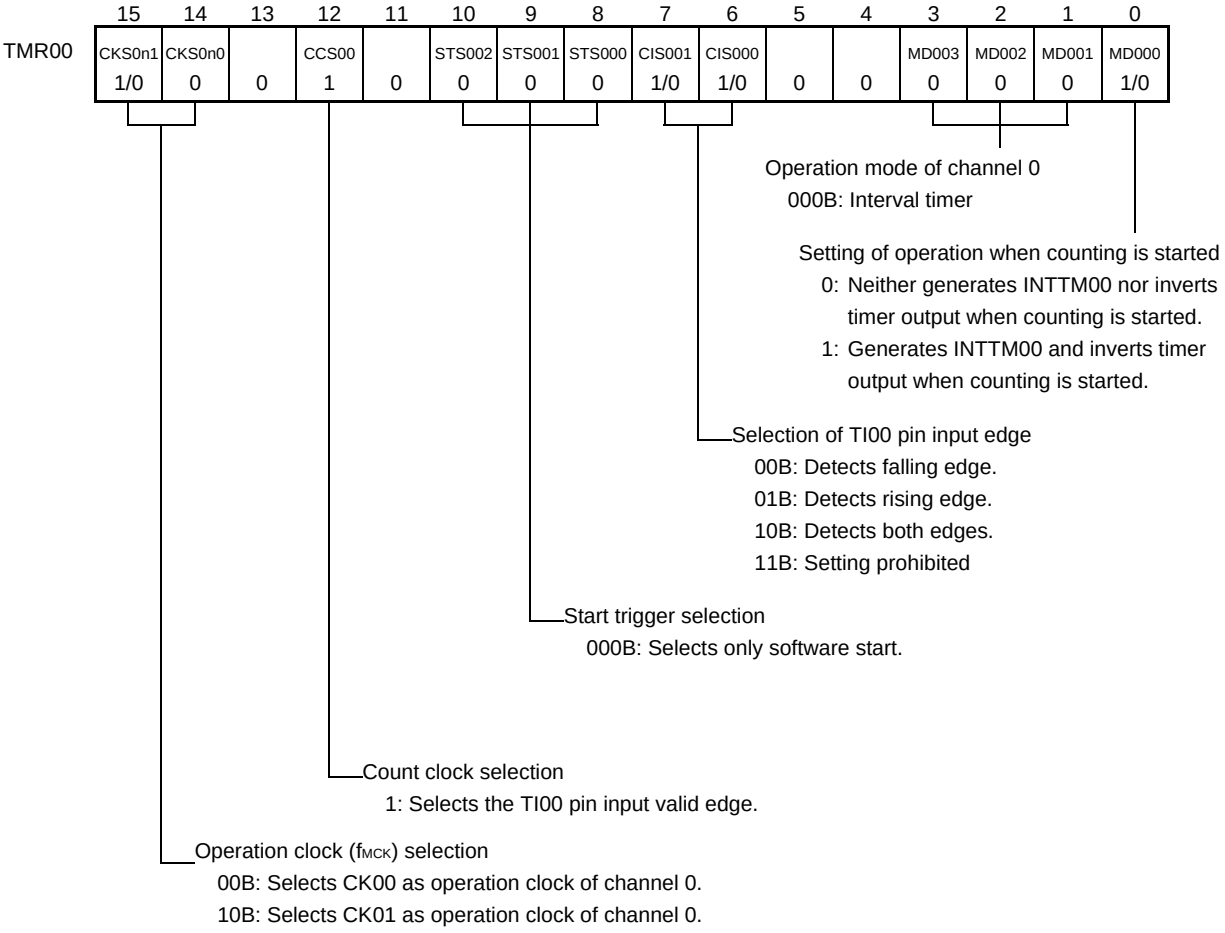


Remark TS00: Bit n of timer channel start register 0 (TS0)
TE00: Bit n of timer channel enable status register 0 (TE0)
TI00: TI00 pin input signal
TCR00: Timer count register 00 (TCR00)
TDR00: Timer data register 00 (TDR00)
TO00: TO00 pin output signal

Downloaded from Arrow.com.

Figure 6-48. Example of Set Contents of Registers During Operation as Frequency Divider

(a) Timer mode register 00 (TMR00)



(b) Timer output register 0 (TO0)

	Bit 0	
TO0	TO00	0: Outputs 0 from TO00. 1: Outputs 1 from TO00.
	1/0	

(c) Timer output enable register 0 (TOE0)

	Bit 0	
TOE0	TOE00	0: Stops the TO00 output operation by counting operation. 1: Enables the TO00 output operation by counting operation.
	1/0	

(d) Timer output level register 0 (TOL0)

	Bit 0	
TOL0	TOL00	0: Cleared to 0 when master channel output mode (TOM00 = 0)
	0	

(e) Timer output mode register 0 (TOM0)

	Bit 0	
TOM0	TOM00	0: Sets master channel output mode.
	0	

Downloaded from Arrow.com.

Figure 6-49. Operation Procedure When Frequency Divider Function Is Used

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN bit of peripheral enable register 0 (PER0) to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register 0 (TPS0). Determines clock frequencies of CK00 and CK01.	
<R> Channel default setting	Sets the corresponding bit of the noise filter enable registers 1 (NFEN1) to 0 (off) or 1 (on). Sets timer mode register 00 (TMR00) (determines operation mode of channel and selects the detection edge). Sets interval (period) value to timer data register 00 (TDR00).	Channel stops operating. (Clock is supplied and some power is consumed.)
	Clears the TOM00 bit of timer output mode register 0 (TOM0) to 0 (master channel output mode). Clears the TOL0n bit to 0. Sets the TO00 bit and determines default level of the TO00 output. →	The TO00 pin goes into Hi-Z output state. The TO00 default setting level is output when the port mode register is in output mode and the port register is 0.
	Sets the TOE00 bit to 1 and enables operation of TO00. →	TO00 does not change because channel stops operating.
	Clears the port register and port mode register to 0. →	The TO00 pin outputs the TO00 set level.
Operation start	Sets the TOE00 bit to 1 (only when operation is resumed). → Sets the TS00 bit to 1. The TS00 bit automatically returns to 0 because it is a trigger bit.	TE00 = 1, and count operation starts. Value of the TDR00 register is loaded to timer count register 00 (TCR00). INTTM0n is generated and TO00 performs toggle operation if the MD00n bit of the TMR00 register is 1.
During operation	Set value of the TDR00 register can be changed. The TCR00 register can always be read. The TSR00 register is not used. Set values of the TO0 and TOE0 registers can be changed. Set values of the TMR00 register, TOM00, and TOL00 bits cannot be changed.	Counter (TCR00) counts down. When count value reaches 0000H, the value of the TDR00 register is loaded to the TCR00 register again, and the count operation is continued. By detecting TCR00 = 0000H, INTTM00 is generated and TO00 performs toggle operation. After that, the above operation is repeated.
Operation stop	The TT00 bit is set to 1. → The TT00 bit automatically returns to 0 because it is a trigger bit.	TE00 = 0, and count operation stops. The TCR00 register holds count value and stops. The TO00 output is not initialized but holds current status.
	The TOE00 bit is cleared to 0 and value is set to the TO00 bit. →	The TO00 pin outputs the TO00 set level.
TAU stop	To hold the TO00 pin output level Clears the TO00 bit to 0 after the value to be held is set to the port register. → When holding the TO00 pin output level is not necessary Setting not required.	The TO00 pin output level is held by port function.
	The TAU0EN bit of the PER0 register is cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TO00 bit is cleared to 0 and the TO00 pin is set to port mode).

Operation is resumed.

Downloaded from Arrow.com.

6.8.4 Operation as input pulse interval measurement

The count value can be captured at the TI0n valid edge and the interval of the pulse input to TI0n can be measured. In addition, the count value can be captured by using software operation (TS0n = 1) as a capture trigger while the TE0n bit is set to 1.

The pulse interval can be calculated by the following expression.

TI0n input pulse interval = Period of count clock × ((10000H × TSR0n: OVF) + (Capture value of TDR0n + 1))

Caution The TI0n pin input is sampled using the operating clock selected with the CKS0n bit of timer mode register 0n (TMR0n), so an error of up to one operating clock cycle occurs.

Timer count register 0n (TCR0n) operates as an up counter in the capture mode.

When the channel start trigger bit (TS0n) of timer channel start register 0 (TS0) is set to 1, the TCR0n register counts up from 0000H in synchronization with the count clock.

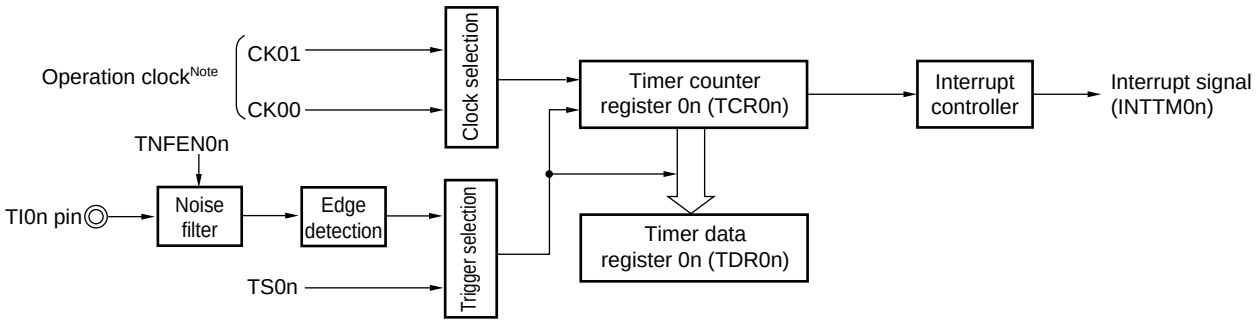
When the TI0n pin input valid edge is detected, the count value of the TCR0n register is transferred (captured) to timer data register 0n (TDR0n) and, at the same time, the TCR0n register is cleared to 0000H, and the INTTM0n is output. If the counter overflows at this time, the OVF bit of timer status register 0n (TSR0n) is set to 1. If the counter does not overflow, the OVF bit is cleared. After that, the above operation is repeated.

As soon as the count value has been captured to the TDR0n register, the OVF bit of the TSR0n register is updated depending on whether the counter overflows during the measurement period. Therefore, the overflow status of the captured value can be checked.

If the counter reaches a full count for two or more periods, it is judged to be an overflow occurrence, and the OVF bit of the TSR0n register is set to 1. However, a normal interval value cannot be measured for the OVF bit, if two or more overflows occur.

Set the STS0n2 to STS0n0 bits of the TMR0n register to 001B to use the valid edges of TI0n as a start trigger and a capture trigger.

Figure 6-50. Block Diagram of Operation as Input Pulse Interval Measurement

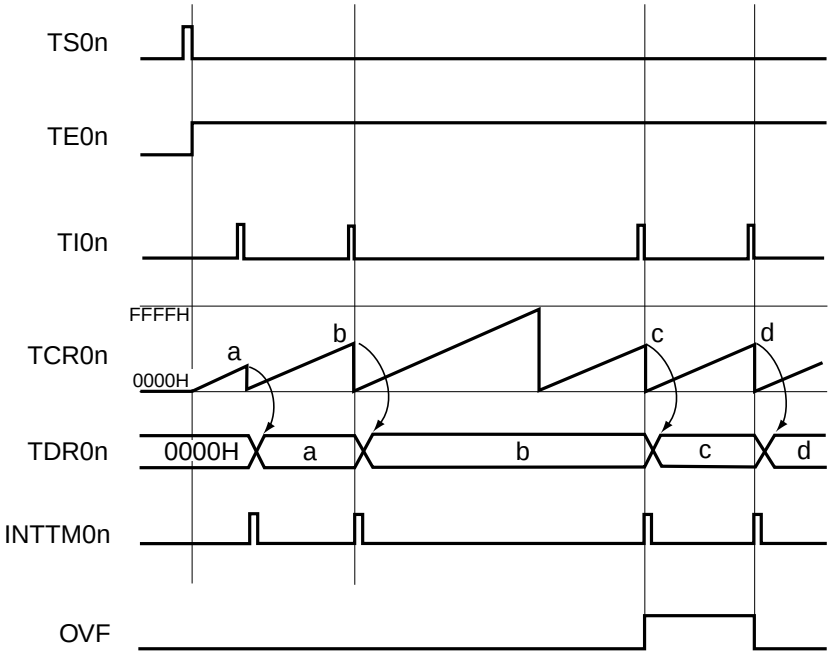


<R> **Note** When channels 1 and 3, the clock can be selected from CK00, CK01, CK02, and CK03.

Remark n: Channel number (n = 0 to 7)

Downloaded from Arrow.com.

Figure 6-51. Example of Basic Timing of Operation as Input Pulse Interval Measurement (MD0n0 = 0)



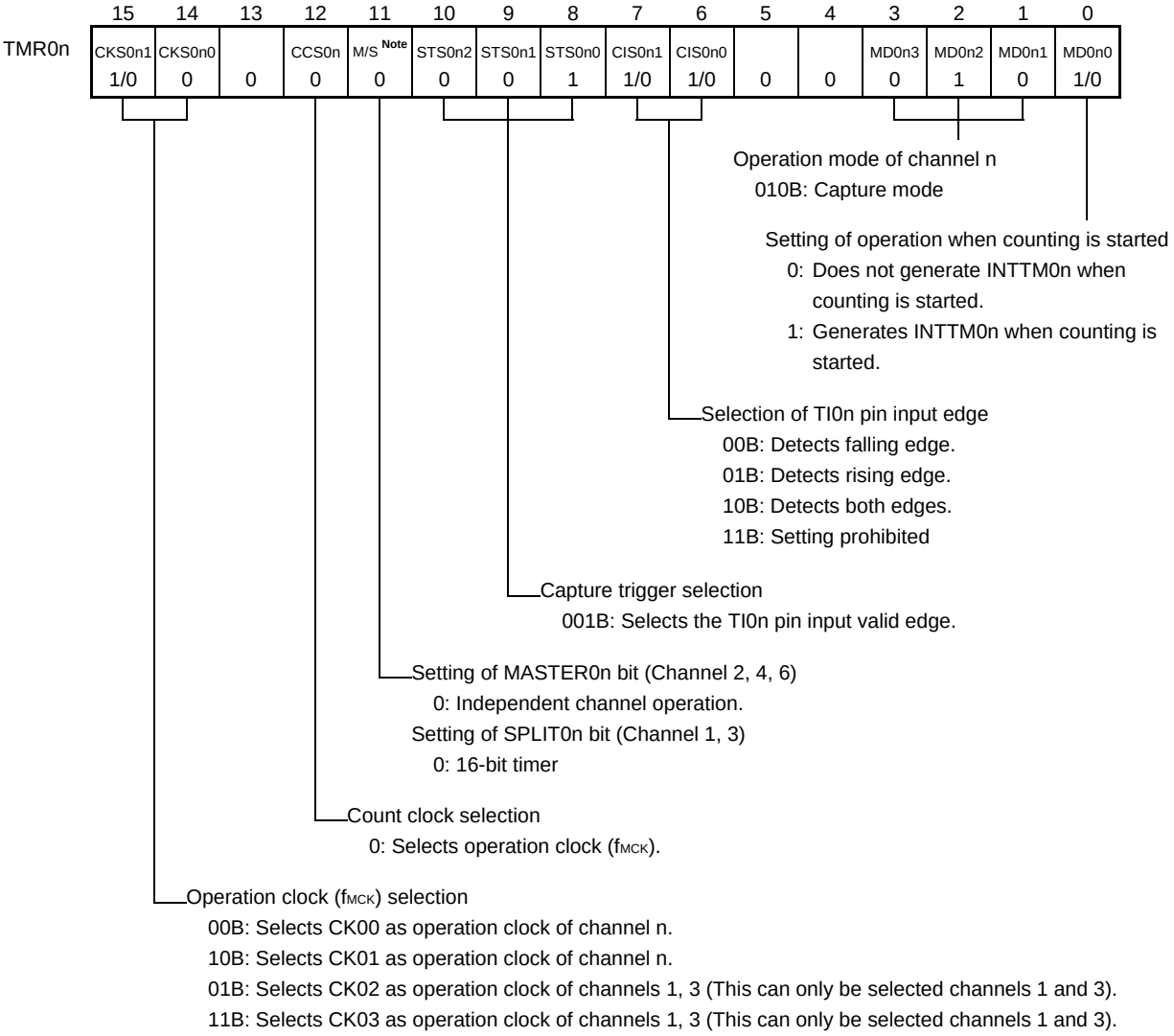
Remarks 1. n: Channel number (n = 0 to 7)

2. TS0n: Bit n of timer channel start register 0 (TS0)
TE0n: Bit n of timer channel enable status register 0 (TE0)
TI0n: TI0n pin input signal
TCR0n: Timer count register 0n (TCR0n)
TDR0n: Timer data register 0n (TDR0n)
OVF: Bit 0 of timer status register 0n (TSR0n)

Downloaded from Arrow.com.

Figure 6-52. Example of Set Contents of Registers to Measure Input Pulse Interval (1/2)

(a) Timer mode register 0n (TMR0n)



(b) Timer output register 0 (TO0)

	Bit n	
TO0	TO0n 0	0: Outputs 0 from TO0n.

(c) Timer output enable register 0 (TOE0)

	Bit n	
TOE0	TOE0n 0	0: Stops TO0n output operation by counting operation.

Note TMR02, TMR04, TMR06: MASTER0n bit
TMR01, TMR03: SPLIT0n bit
TMR00, TMR05, TMR07: 0 fixed

Remark n: Channel number (n = 0 to 7)

Downloaded from Arrow.com.

Figure 6-52. Example of Set Contents of Registers to Measure Input Pulse Interval (2/2)

(d) Timer output level register 0 (TOL0)

	Bit n	
TOL0	TOL0n 0	0: Cleared to 0 when master channel output mode (TOM0n = 0).

(e) Timer output mode register 0 (TOM0)

	Bit n	
TOM0	TOM0n 0	0: Sets master channel output mode.

Note TMR02, TMR04, TMR06: MASTER0n bit
TMR01, TMR03: SPLIT0n bit
TMR00, TMR05, TMR07: 0 fixed

Remark n: Channel number (n = 0 to 7)

Downloaded from Arrow.com.

Figure 6-53. Operation Procedure When Input Pulse Interval Measurement Function Is Used

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN bit of peripheral enable register 0 (PER0) to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register 0 (TPS0). Determines clock frequencies of CK00 to CK03.	
<R> Channel default setting	Sets the corresponding bit of the noise filter enable registers 1 (NFEN1) to 0 (off) or 1 (on). Sets timer mode register 0n (TMR0n) (determines operation mode of channel). Sets Noise filter enable register 1 (NFEN1).	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	Sets TS0n bit to 1. → The TS0n bit automatically returns to 0 because it is a trigger bit.	TE0n = 1, and count operation starts. Timer count register 0n (TCR0n) is cleared to 0000H at the count clock input. When the MD0n0 bit of the TMR0n register is 1, INTTM0n is generated.
<R> During operation	Set values of only the CIS0n1 and CIS0n0 bits of the TMR0n register can be changed. The TDR0n register can always be read. The TCR0n register can always be read. The TSR0n register can always be read. Set values of the TOM0n, TOL0n, TO0n, and TOE0n bits cannot be changed.	Counter (TCR0n) counts up from 0000H. When the valid edge of the TI0n pin input is detected or the TS0n bit is set to 1, the count value is transferred (captured) to timer data register 0n (TDR0n). At the same time, the TCR0n register is cleared to 0000H, and the INTTM0n signal is generated. If an overflow occurs at this time, the OVF bit of timer status register 0n (TSR0n) is set; if an overflow does not occur, the OVF bit is cleared. After that, the above operation is repeated.
Operation stop	The TT0n bit is set to 1. → The TT0n bit automatically returns to 0 because it is a trigger bit.	TE0n = 0, and count operation stops. The TCR0n register holds count value and stops. The OVF bit of the TSR0n register is also held.
TAU stop	The TAU0EN bit of the PER0 register is cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized.

Operation is resumed.

Remark n: Channel number (n = 0 to 7)

Downloaded from Arrow.com.

6.8.5 Operation as input signal high-/low-level width measurement

By starting counting at one edge of the TI0n pin input and capturing the number of counts at another edge, the signal width (high-level width/low-level width) of TI0n can be measured. The signal width of TI0n can be calculated by the following expression.

Signal width of TI0n input = Period of count clock × ((10000H × TSR0n: OVF) + (Capture value of TDR0n + 1))

Caution The TI0n pin input is sampled using the operating clock selected with the CKS0n bit of timer mode register 0n (TMR0n), so an error equivalent to one operation clock occurs.

Timer count register 0n (TCR0n) operates as an up counter in the capture & one-count mode.

When the channel start trigger bit (TS0n) of timer channel start register 0 (TS0) is set to 1, the TE0n bit is set to 1 and the TI0n pin start edge detection wait status is set.

When the TI0n pin input start edge (rising edge of the TI0n pin input when the high-level width is to be measured) is detected, the counter counts up from 0000H in synchronization with the count clock. When the valid capture edge (falling edge of the TI0n pin input when the high-level width is to be measured) is detected later, the count value is transferred to timer data register 0n (TDR0n) and, at the same time, INTTM0n is output. If the counter overflows at this time, the OVF bit of timer status register 0n (TSR0n) is set to 1. If the counter does not overflow, the OVF bit is cleared. The TCR0n register stops at the value "value transferred to the TDR0n register + 1", and the TI0n pin start edge detection wait status is set. After that, the above operation is repeated.

As soon as the count value has been captured to the TDR0n register, the OVF bit of the TSR0n register is updated depending on whether the counter overflows during the measurement period. Therefore, the overflow status of the captured value can be checked.

If the counter reaches a full count for two or more periods, it is judged to be an overflow occurrence, and the OVF bit of the TSR0n register is set to 1. However, a normal interval value cannot be measured for the OVF bit, if two or more overflows occur.

Whether the high-level width or low-level width of the TI0n pin is to be measured can be selected by using the CIS0n1 and CIS0n0 bits of the TMR0n register.

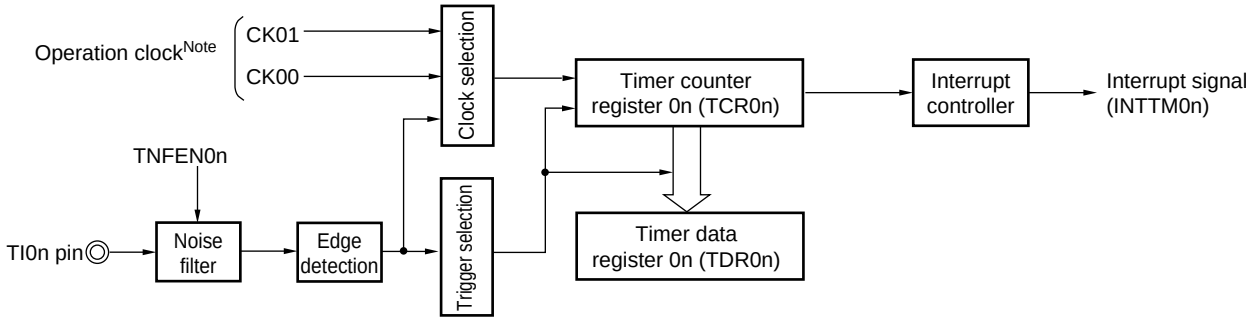
Because this function is used to measure the signal width of the TI0n pin input, the TS0n bit cannot be set to 1 while the TE0n bit is 1.

CIS0n1, CIS0n0 of TMR0n register = 10B: Low-level width is measured.

CIS0n1, CIS0n0 of TMR0n register = 11B: High-level width is measured.

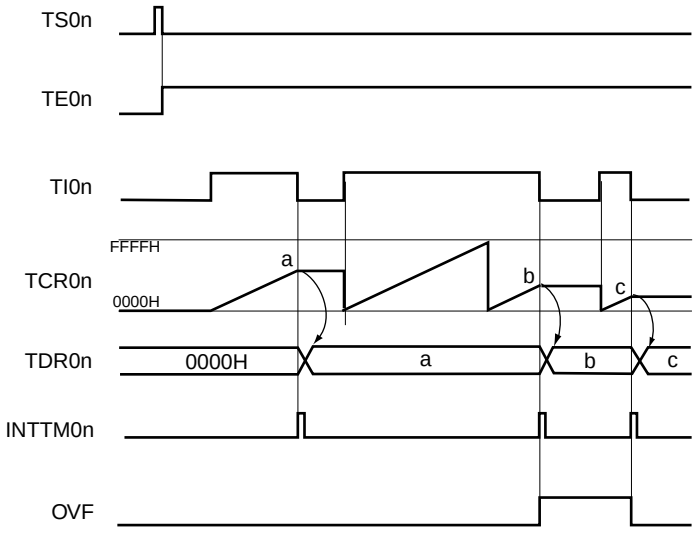
Downloaded from Arrow.com.

Figure 6-54. Block Diagram of Operation as Input Signal High-/Low-Level Width Measurement



Note For channels 1 and 3, the clock can be selected from CK00, CK01, CK02, and CK03.

Figure 6-55. Example of Basic Timing of Operation as Input Signal High-/Low-Level Width Measurement



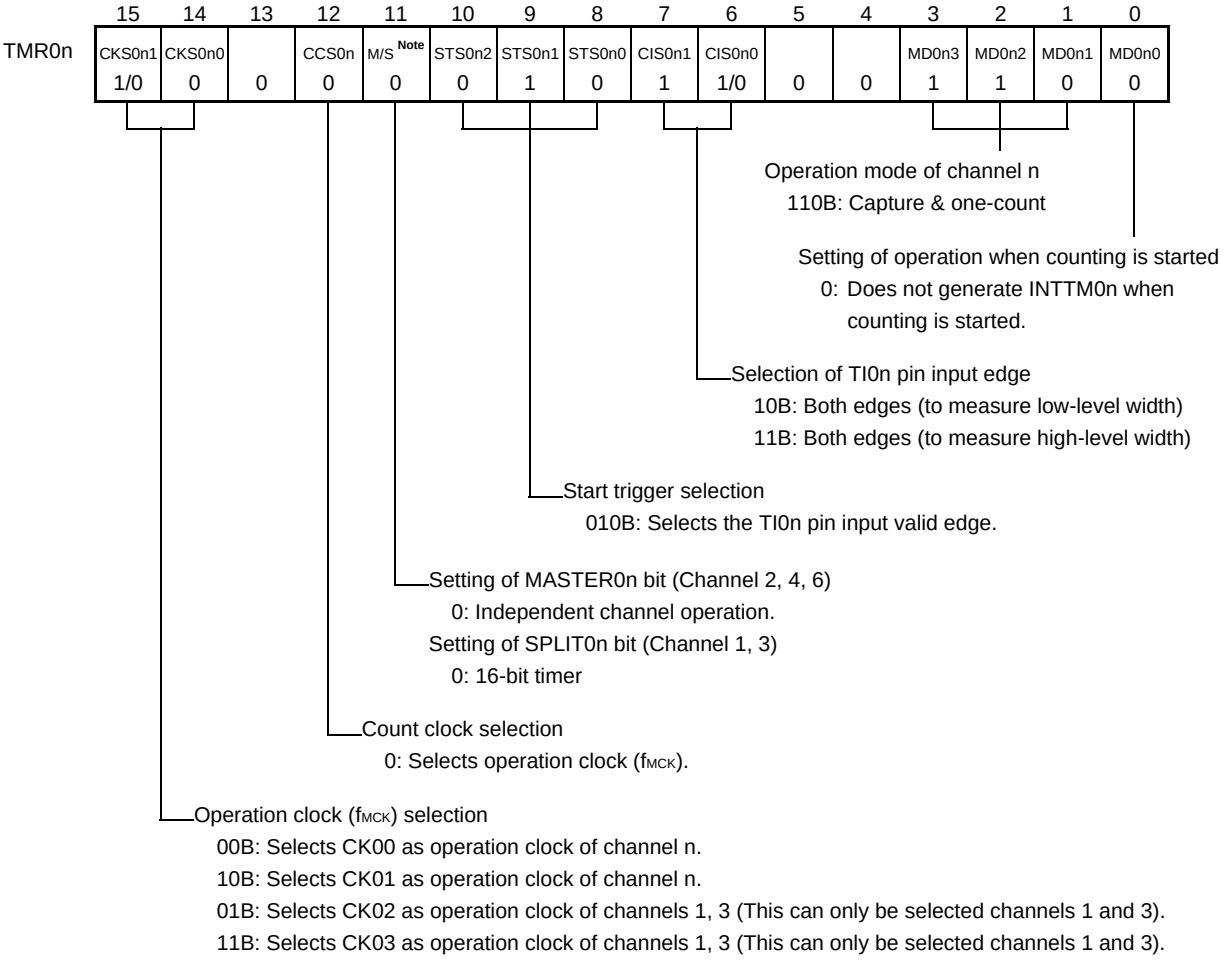
Remarks 1. n: Channel number (n = 0 to 7)

2. TS0n: Bit n of timer channel start register 0 (TS0)
TE0n: Bit n of timer channel enable status register 0 (TE0)
TI0n: TI0n pin input signal
TCR0n: Timer count register 0n (TCR0n)
TDR0n: Timer data register 0n (TDR0n)
OVF: Bit 0 of timer status register 0n (TSR0n)

Downloaded from Arrow.com.

Figure 6-56. Example of Set Contents of Registers to Measure Input Signal High-/Low-Level Width

(a) Timer mode register 0n (TMR0n)



(b) Timer output register 0 (TO0)

	Bit n	
TO0	TO0n 0	0: Outputs 0 from TO0n.

(c) Timer output enable register 0 (TOE0)

	Bit n	
TOE0	TOE0n 0	0: Stops the TO0n output operation by counting operation.

(d) Timer output level register 0 (TOL0)

	Bit n	
TOL0	TOL0n 0	0: Cleared to 0 when master channel output mode (TOM0n = 0).

(e) Timer output mode register 0 (TOM0)

	Bit n	
TOM0	TOM0n 0	0: Sets master channel output mode.

Note TMR02, TMR04, TMR06: MASTER0n bit
TMR01, TMR03: SPLIT0n bit
TMR00, TMR05, TMR07: 0 fixed

Remark n: Channel number (n = 0 to 7)

Downloaded from Arrow.com.

Figure 6-57. Operation Procedure When Input Signal High-/Low-Level Width Measurement Function Is Used

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN bit of peripheral enable register 0 (PER0) to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register 0 (TPS0). Determines clock frequencies of CK00 to CK03.	
<R> Channel default setting	Sets the corresponding bit of the noise filter enable registers 1 (NFEN1) to 0 (off) or 1 (on). Sets timer mode register 0n (TMR0n) (determines operation mode of channel). Sets noise filter enable register 1 (NFEN1) Clears the TOE0n bit to 0 and stops operation of TO0n.	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	Sets the TS0n bit to 1. → The TS0n bit automatically returns to 0 because it is a trigger bit.	TE0n = 1, and the TI0n pin start edge detection wait status is set.
	Detects the TI0n pin input count start valid edge. →	Clears timer count register 0n (TCR0n) to 0000H and starts counting up.
During operation	Set value of the TDR0n register can be changed. The TCR0n register can always be read. The TSR0n register is not used. Set values of the TMR0n register, TOM0n, TOL0n, TO0n, and TOE0n bits cannot be changed.	When the TI0n pin start edge is detected, the counter (TCR0n) counts up from 0000H. If a capture edge of the TI0n pin is detected, the count value is transferred to timer data register 0n (TDR0n) and INTTM0n is generated. If an overflow occurs at this time, the OVF bit of timer status register 0n (TSR0n) is set; if an overflow does not occur, the OVF bit is cleared. The TCR0n register stops the count operation until the next TI0n pin start edge is detected.
Operation stop	The TT0n bit is set to 1. → The TT0n bit automatically returns to 0 because it is a trigger bit.	TE0n = 0, and count operation stops. The TCR0n register holds count value and stops. The OVF bit of the TSR0n register is also held.
TAU stop	The TAU0EN bit of the PER0 register is cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized.

Operation is resumed.

Remark n: Channel number (n = 0 to 7)

Downloaded from Arrow.com.

6.8.6 Operation as delay counter

It is possible to start counting down when the valid edge of the TI0n pin input is detected (an external event), and then generate INTTM0n (a timer interrupt) after any specified interval.

It can also start counting down and generate INTTM0n (timer interrupt) at any interval by setting TS0n to 1 by software during the period of TE0n = 1.

The interrupt generation period can be calculated by the following expression.

Generation period of INTTM0n (timer interrupt) = Period of count clock × (Set value of TDR0n + 1)

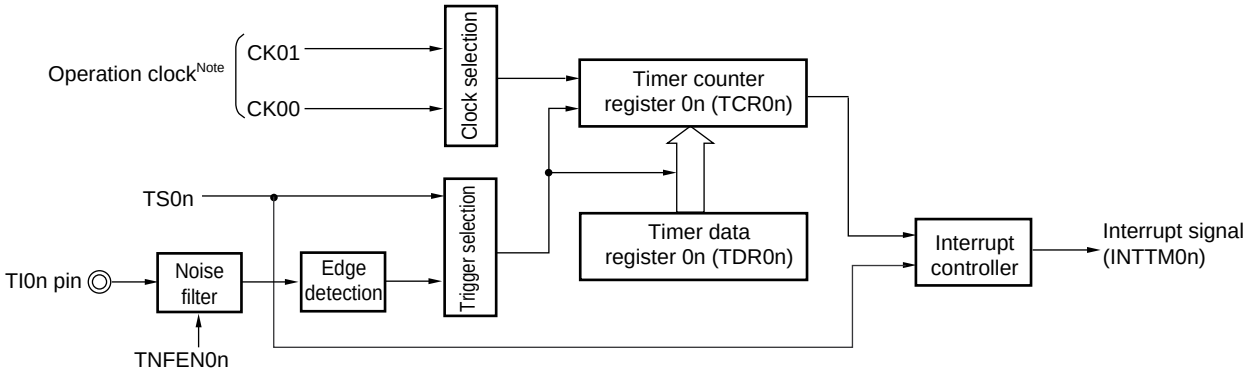
Timer count register 0n (TCR0n) operates as a down counter in the one-count mode.

When the channel start trigger bit (TS0n, TSHm1, TSHm3) of timer channel start register 0 (TS0) is set to 1, the TE0n, TEHm1, TEHm3 bits are set to 1 and the TI0n pin input valid edge detection wait status is set.

Timer count register 0n (TCR0n) starts operating upon TI0n pin input valid edge detection and loads the value of timer data register 0n (TDR0n). The TCR0n register counts down from the value of the TDR0n register it has loaded, in synchronization with the count clock. When TCR0n = 0000H, it outputs INTTM0n and stops counting until the next TI0n pin input valid edge is detected.

The TDR0n register can be rewritten at any time. The new value of the TDR0n register becomes valid from the next period.

Figure 6-58. Block Diagram of Operation as Delay Counter

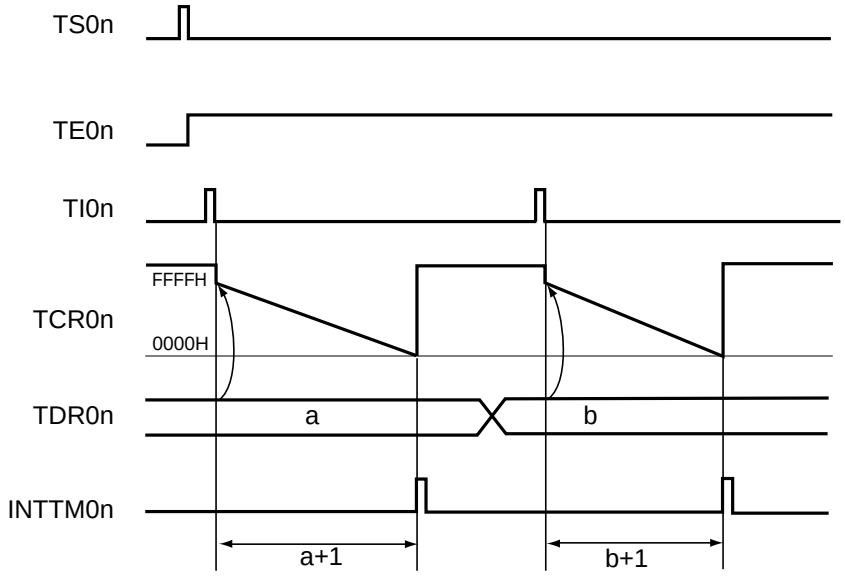


Note For using channels 1 and 3 in 8-bit timer mode, the clock can be selected from CK00, CK01, CK02, and CK03.

Remark n: Channel number (n = 0 to 7)

Downloaded from Arrow.com.

Figure 6-59. Example of Basic Timing of Operation as Delay Counter



Remarks 1. n: Channel number (n = 0 to 7)

2. TS0n: Bit n of timer channel start register 0 (TS0)

TE0n: Bit n of timer channel enable status register 0 (TE0)

TI0n: TI0n pin input signal

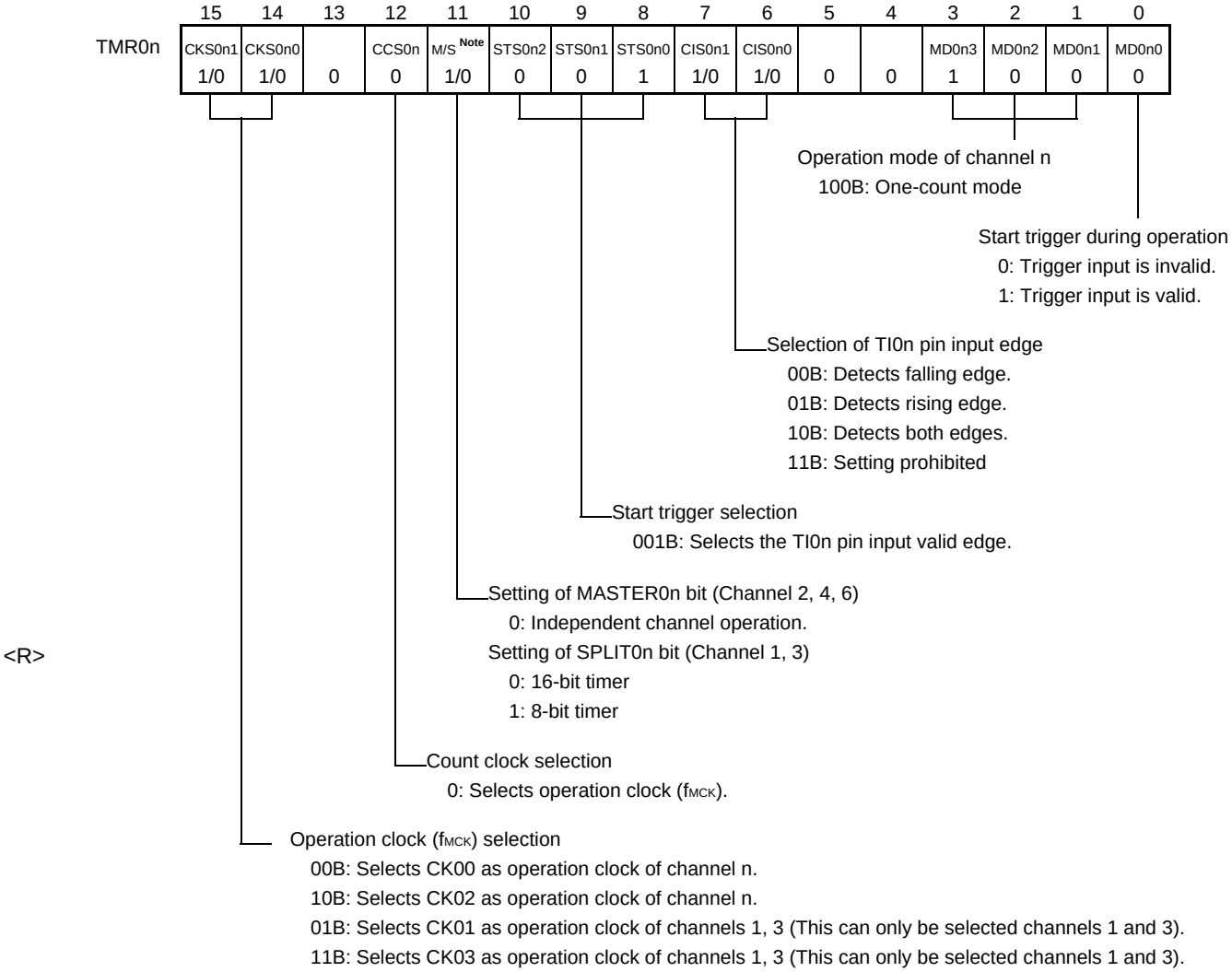
TCR0n: Timer count register 0n (TCR0n)

TDR0n: Timer data register 0n (TDR0n)

Downloaded from Arrow.com.

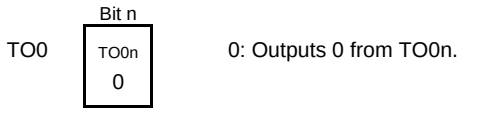
Figure 6-60. Example of Set Contents of Registers to Delay Counter (1/2)

(a) Timer mode register 0n (TMR0n)

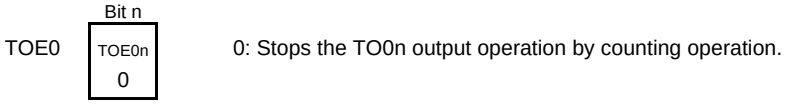


<R>

(b) Timer output register 0 (TO0)



(c) Timer output enable register 0 (TOE0)



Note TMR02, TMR04, TMR06: MASTER0n bit
TMR01, TMR03: SPLIT0n bit
TMR00, TMR05, TMR07: 0 fixed

Remark n: Channel number (n = 0 to 7)

Downloaded from Arrow.com.

Figure 6-60. Example of Set Contents of Registers to Delay Counter (2/2)

(d) Timer output level register 0 (TOL0)

	Bit n	
TOL0	TOL0n 0	0: Cleared to 0 when master channel output mode (TOM0n = 0).

(e) Timer output mode register 0 (TOM0)

	Bit n	
TOM0	TOM0n 0	0: Sets master channel output mode.

Remark n: Channel number (n = 0 to 7)

Downloaded from Arrow.com.

Figure 6-61. Operation Procedure When Delay Counter Function Is Used

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN bit of peripheral enable register 0 (PER0) to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register 0 (TPS0). Determines clock frequencies of CK00 and CK01 (or CK02 and CK03 when using the 8-bit timer mode).	
<R> Channel default setting	Sets the corresponding bit of the noise filter enable registers 1 (NFEN1) to 0 (off) or 1 (on). Sets timer mode register 0n (TMR0n) (determines operation mode of channel). INTTM0n output delay is set to timer data register 0n (TDR0n). Sets noise filter enable register 1 (NFEN1). Clears the TOE0n bit to 0 and stops operation of TO0n.	Channel stops operating. (Clock is supplied and some power is consumed.)
<R> Operation start	Sets the TS0n bit to 1. → The TS0n bit automatically returns to 0 because it is a trigger bit.	TE0n = 1, and the start trigger detection (the valid edge of the TI0n pin input is detected or the TS0n bit is set to 1) wait status is set.
	The counter starts counting down by the next start trigger detection. • Detects the valid edge of the TI0n pin input. → • Sets the TS0n bit to 1 by the software.	Value of the TDR0n register is loaded to the timer count register 0n (TCR0n).
<R> During operation	Set value of the TDR0n register can be changed. The TCR0n register can always be read. The TSR0n register is not used.	The counter (TCR0n) counts down. When the count value of TCR0n reaches 0000H, INTTM0n output is generated, and count operation stops until the next start trigger detection (the valid edge of the TI0n pin input is detected or the TS0n bit is set to 1).
Operation stop	The TT0n bit is set to 1. → The TT0n bit automatically returns to 0 because it is a trigger bit.	TE0n = 0, and count operation stops. The TCR0n register holds count value and stops.
TAU stop	The TAU0EN bit of the PER0 register is cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized.

Operation is resumed.

Remark n: Channel number (n = 0 to 7)

Downloaded from Arrow.com.

6.9 Simultaneous Channel Operation Function of Timer Array Unit

6.9.1 Operation as one-shot pulse output function

By using two channels as a set, a one-shot pulse having any delay pulse width can be generated from the signal input to the TI0n pin.

The delay time and pulse width can be calculated by the following expressions.

Delay time = {Set value of TDR0n (master) + 2} × Count clock period
Pulse width = {Set value of TDR0p (slave)} × Count clock period

The master channel operates in the one-count mode and counts the delays. Timer count register 0n (TCR0n) of the master channel starts operating upon start trigger detection and loads the value of timer data register 0n (TDR0n).

The TCR0n register counts down from the value of the TDR0n register it has loaded, in synchronization with the count clock. When TCR0n = 0000H, it outputs INTTM0n and stops counting until the next start trigger is detected.

The slave channel operates in the one-count mode and counts the pulse width. The TCR0p register of the slave channel starts operation using INTTM0n of the master channel as a start trigger, and loads the value of the TDR0p register. The TCR0p register counts down from the value of The TDR0p register it has loaded, in synchronization with the count value. When count value = 0000H, it outputs INTTM0p and stops counting until the next start trigger (INTTM0n of the master channel) is detected. The output level of TO0p becomes active one count clock after generation of INTTM0n from the master channel, and inactive when TCR0p = 0000H.

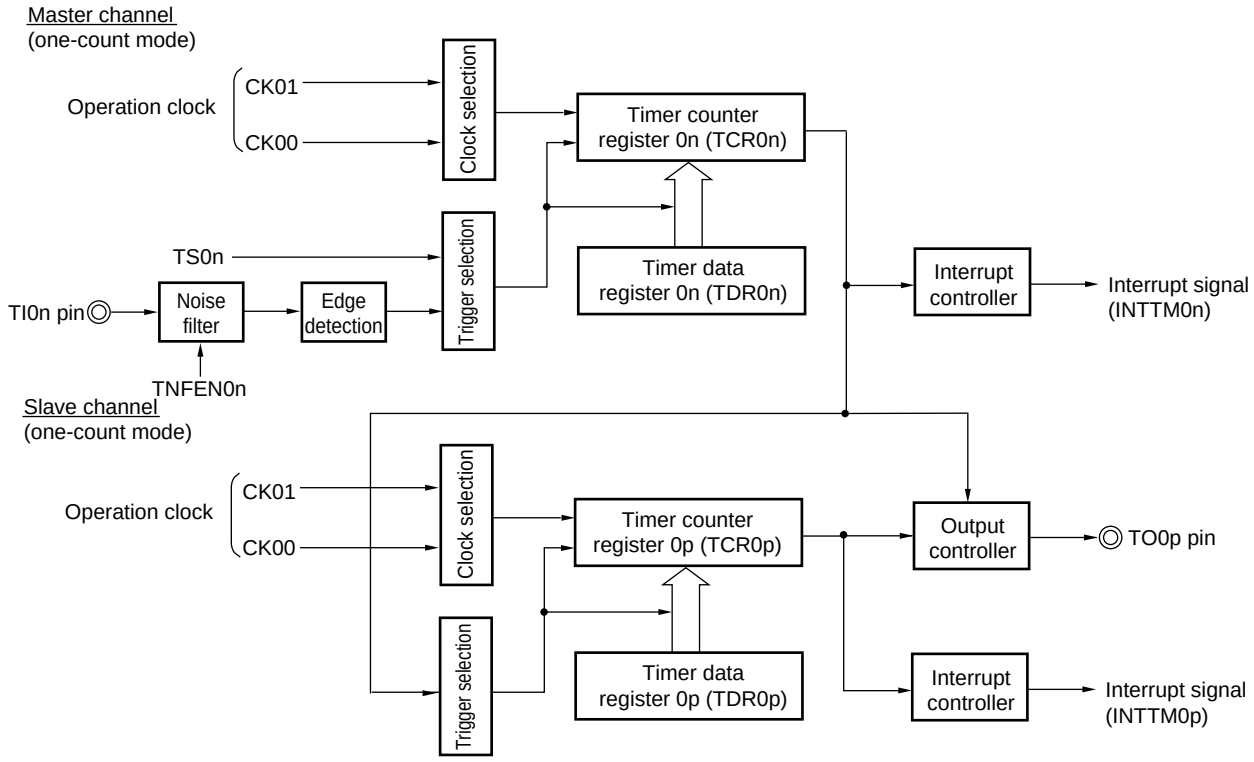
Instead of using the TI0n pin input, a one-shot pulse can also be output using the software operation (TS0n = 1) as a start trigger.

Caution The timing of loading of timer data register 0n (TDR0n) of the master channel is different from that of the TDR0p register of the slave channel. If the TDR0n and TDR0p registers are rewritten during operation, therefore, an illegal waveform is output. Rewrite the TDR0n register after INTTM0n is generated and the TDR0p register after INTTM0p is generated.

Remark n: Channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

Downloaded from Arrow.com.

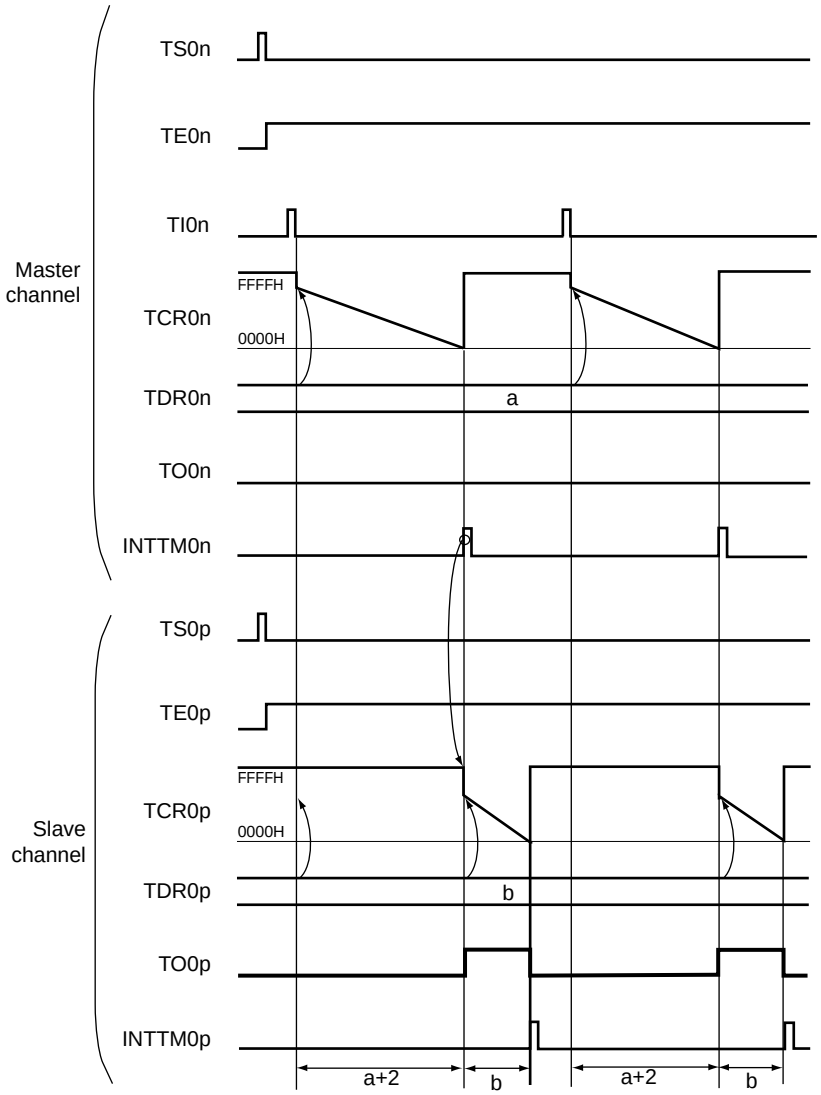
Figure 6-62. Block Diagram of Operation as One-Shot Pulse Output Function



Remark n: Channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

Downloaded from Arrow.com.

Figure 6-63. Example of Basic Timing of Operation as One-Shot Pulse Output Function



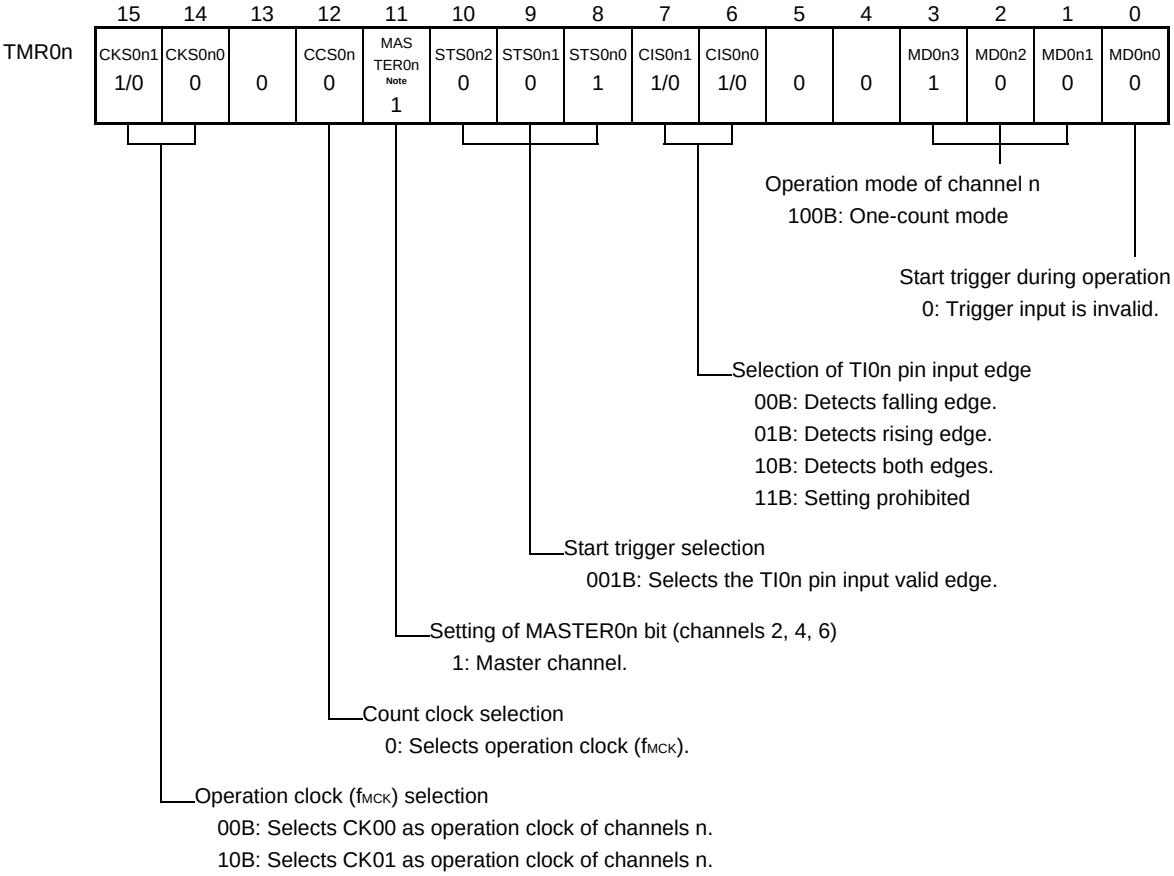
Remarks 1. n: Channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

2. TS0n, TS0p: Bit n, p of timer channel start register 0 (TS0)
TE0n, TE0p: Bit n, p of timer channel enable status register 0 (TE0)
TI0n, TI0p: TI0n and TI0p pins input signal
TCR0n, TCR0p: Timer count registers mn, mp (TCR0n, TCR0p)
TDR0n, TDR0p: Timer data registers mn, mp (TDR0n, TDR0p)
TO0n, TO0p: TO0n and TO0p pins output signal

Downloaded from Arrow.com.

Figure 6-64. Example of Set Contents of Registers When One-Shot Pulse Output Function Is Used (Master Channel)

(a) Timer mode register 0n (TMR0n)



<R>

Note TMR02, TMR04, TMR06: MASTER0n = 1
TMR00: Fixed to 0

(b) Timer output register 0 (TO0)

	Bit n	
TO0	TO0n 0	0: Outputs 0 from TO0n.

(c) Timer output enable register 0 (TOE0)

	Bit n	
TOE0	TOE0n 0	0: Stops the TO0n output operation by counting operation.

(d) Timer output level register 0 (TOL0)

	Bit n	
TOL0	TOL0n 0	0: Cleared to 0 when TOM0n = 0 (master channel output mode).

(e) Timer output mode register 0 (TOM0)

	Bit n	
TOM0	TOM0n 0	0: Sets master channel output mode.

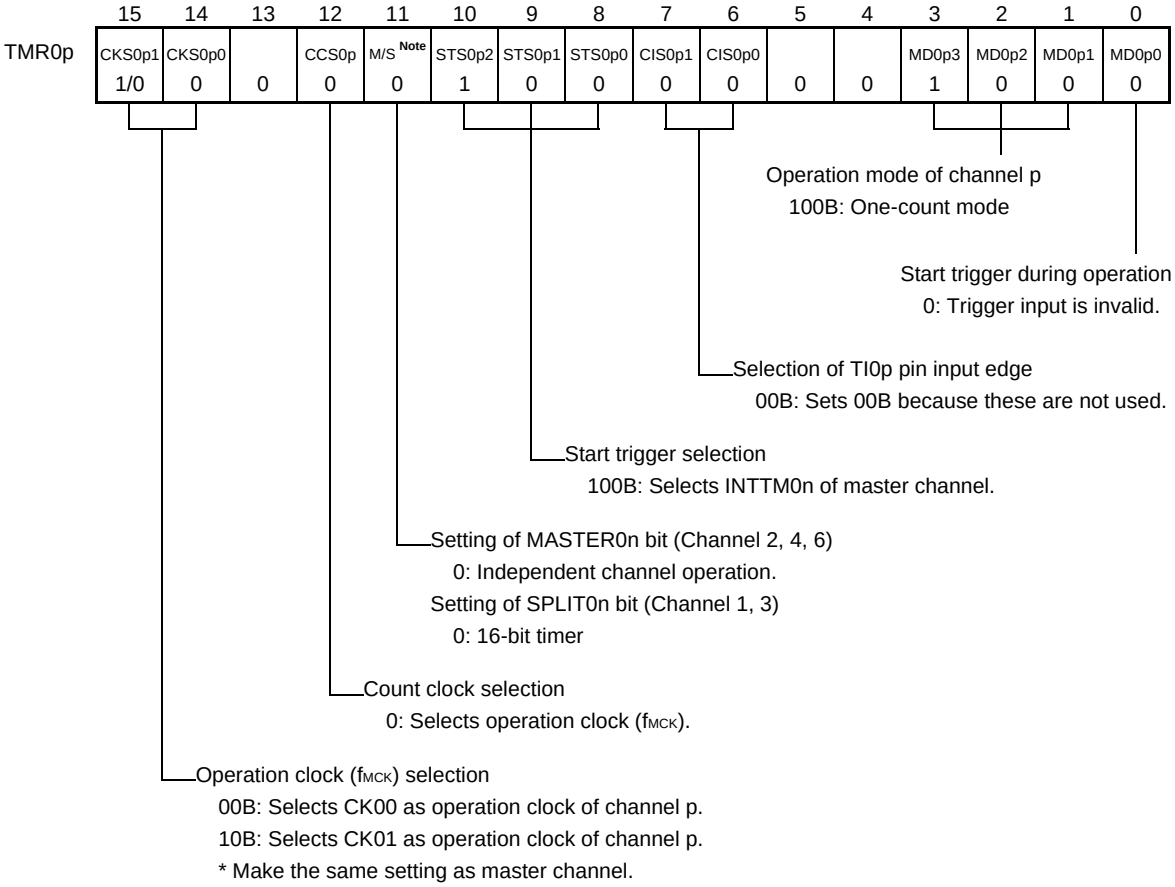
<R> **Note** TMR02, TMR04, TMR06: MASTER0n = 1
TMR00: Fixed to 0

Remark n: Channel number (n = 0, 2, 4, 6)

Downloaded from Arrow.com.

Figure 6-65. Example of Set Contents of Registers When One-Shot Pulse Output Function Is Used (Slave Channel)

(a) Timer mode register 0p (TMR0p)



(b) Timer output register 0 (TO0)

	Bit p	
TO0	TO0p 1/0	0: Outputs 0 from TO0p. 1: Outputs 1 from TO0p.

(c) Timer output enable register 0 (TOE0)

	Bit p	
TOE0	TOE0p 1/0	0: Stops the TO0p output operation by counting operation. 1: Enables the TO0p output operation by counting operation.

(d) Timer output level register 0 (TOL0)

	Bit p	
TOL0	TOL0p 1/0	0: Positive logic output (active-high) 1: Negative logic output (active-low)

(e) Timer output mode register 0 (TOM0)

	Bit p	
TOM0	TOM0p 1	1: Sets the slave channel output mode.

Note TMR02, TMR04, TMR06: MASTER0n bit
TMR01, TMR03: SPLIT0n bit
TMR00, TMR05, TMR07: 0 fixed

Remark n: Master channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

Downloaded from Arrow.com.

Figure 6-66. Operation Procedure of One-Shot Pulse Output Function (1/2)

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN bit of peripheral enable registers 0 (PER0) to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register 0 (TPS0). Determines clock frequencies of CK00 and CK01.	
<R> Channel default setting	Sets the corresponding bit of the noise filter enable registers 1 (NFEN1) to 0 (off) or 1 (on). Sets timer mode register 0n, mp (TMR0n, TMR0p) of two channels to be used (determines operation mode of channels). An output delay is set to timer data register 0n (TDR0n) of the master channel, and a pulse width is set to the TDR0p register of the slave channel.	Channel stops operating. (Clock is supplied and some power is consumed.)
	Sets slave channel. The TOM0p bit of timer output mode register 0 (TOM0) is set to 1 (slave channel output mode). Sets the TOL0p bit. Sets the TO0p bit and determines default level of the TO0p output. →	The TO0p pin goes into Hi-Z output state. The TO0p default setting level is output when the port mode register is in output mode and the port register is 0.
	Sets the TOE0p bit to 1 and enables operation of TO0p. →	TO0p does not change because channel stops operating.
	Clears the port register and port mode register to 0. →	The TO0p pin outputs the TO0p set level.

Remark n: Channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

Downloaded from Arrow.com.

Figure 6-66. Operation Procedure of One-Shot Pulse Output Function (2/2)

	Software Operation	Hardware Status
Operation start	Sets the TOE0p bit (slave) to 1 (only when operation is resumed). The TS0n (master) and TS0p (slave) bits of timer channel start register 0 (TS0) are set to 1 at the same time. → The TS0n and TS0p bits automatically return to 0 because they are trigger bits.	→ The TE0n and TE0p bits are set to 1 and the master channel enters the start trigger detection (the valid edge of the TI0n pin input is detected or the TS0n bit of the master channel is set to 1) wait status. Counter stops operating.
<R> Operation is resumed.	Count operation of the master channel is started by start trigger detection of the master channel. • Detects the TI0n pin input valid edge. → • Sets the TS0n bit of the master channel to 1 by software[Note]. **Note** Do not set the TS0n bit of the slave channel to 1.	Master channel starts counting.
<R> During operation	Set values of only the CIS0n1 and CIS0n0 bits of the TMR0n register can be changed. Set values of the TMR0p, TDR0n, TDR0p registers, TOM0n, TOM0p, TOL0n, and TOL0p bits cannot be changed. The TCR0n and TCR0p registers can always be read. The TSR0n and TSR0p registers are not used. Set values of the TO0 and TOE0 registers of slave channel can be changed.	Master channel loads the value of the TDR0n register to timer count register 0n (TCR0n) by the start trigger detection (the valid edge of the TI0n pin input is detected or the TS0n bit of the master channel is set to 1), and the counter starts counting down. When the count value reaches TCR0n = 0000H, the INTTM0n output is generated, and the counter stops until the next start trigger detection. The slave channel, triggered by INTTM0n of the master channel, loads the value of the TDR0p register to the TCR0p register, and the counter starts counting down. The output level of TO0p becomes active one count clock after generation of INTTM0n from the master channel. It becomes inactive when TCR0p = 0000H, and the counting operation is stopped. After that, the above operation is repeated.
Operation stop	The TT0n (master) and TT0p (slave) bits are set to 1 at the same time. → The TT0n and TT0p bits automatically return to 0 because they are trigger bits.	→ TE0n, TE0p = 0, and count operation stops. The TCR0n and TCR0p registers hold count value and stop. The TO0p output is not initialized but holds current status.
	The TOE0p bit of slave channel is cleared to 0 and value is set to the TO0p bit. →	The TO0p pin outputs the TO0p set level.
TAU stop	To hold the TO0p pin output level Clears the TO0p bit to 0 after the value to be held is set to the port register. →	The TO0p pin output level is held by port function.
	When holding the TO0p pin output level is not necessary Setting not required.	
	The TAU0EN bit of the PER0 register is cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TO0p bit is cleared to 0 and the TO0p pin is set to port mode.)

Remark n: Channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

Downloaded from Arrow.com.

6.9.2 Operation as PWM function

Two channels can be used as a set to generate a pulse of any period and duty factor.

The period and duty factor of the output pulse can be calculated by the following expressions.

Pulse period = {Set value of TDR0n (master) + 1} × Count clock period
Duty factor [%] = {Set value of TDR0p (slave)}/{Set value of TDR0n (master) + 1} × 100
0% output: Set value of TDR0p (slave) = 0000H
100% output: Set value of TDR0p (slave) ≥ {Set value of TDR0n (master) + 1}

Remark Although the duty factor exceeds 100% if the set value of TDR0p (slave) > (set value of TDR0n (master) + 1), it summarizes to 100% output.

The master channel operates in the interval timer mode. If the channel start trigger bit (TS0n) of timer channel start register 0 (TS0) is set to 1, an interrupt (INTTM0n) is output, the value set to timer data register 0n (TDR0n) is loaded to timer count register 0n (TCR0n), and the counter counts down in synchronization with the count clock. When the counter reaches 0000H, INTTM0n is output, the value of the TDR0n register is loaded again to the TCR0n register, and the counter counts down. This operation is repeated until the channel stop trigger bit (TT0n) of timer channel stop register 0 (TT0) is set to 1.

If two channels are used to output a PWM waveform, the period until the master channel counts down to 0000H is the PWM output (TO0p) cycle.

The slave channel operates in one-count mode. By using INTTM0n from the master channel as a start trigger, the TCR0p register loads the value of the TDR0p register and the counter counts down to 0000H. When the counter reaches 0000H, it outputs INTTM0p and waits until the next start trigger (INTTM0n from the master channel) is generated.

If two channels are used to output a PWM waveform, the period until the slave channel counts down to 0000H is the PWM output (TO0p) duty.

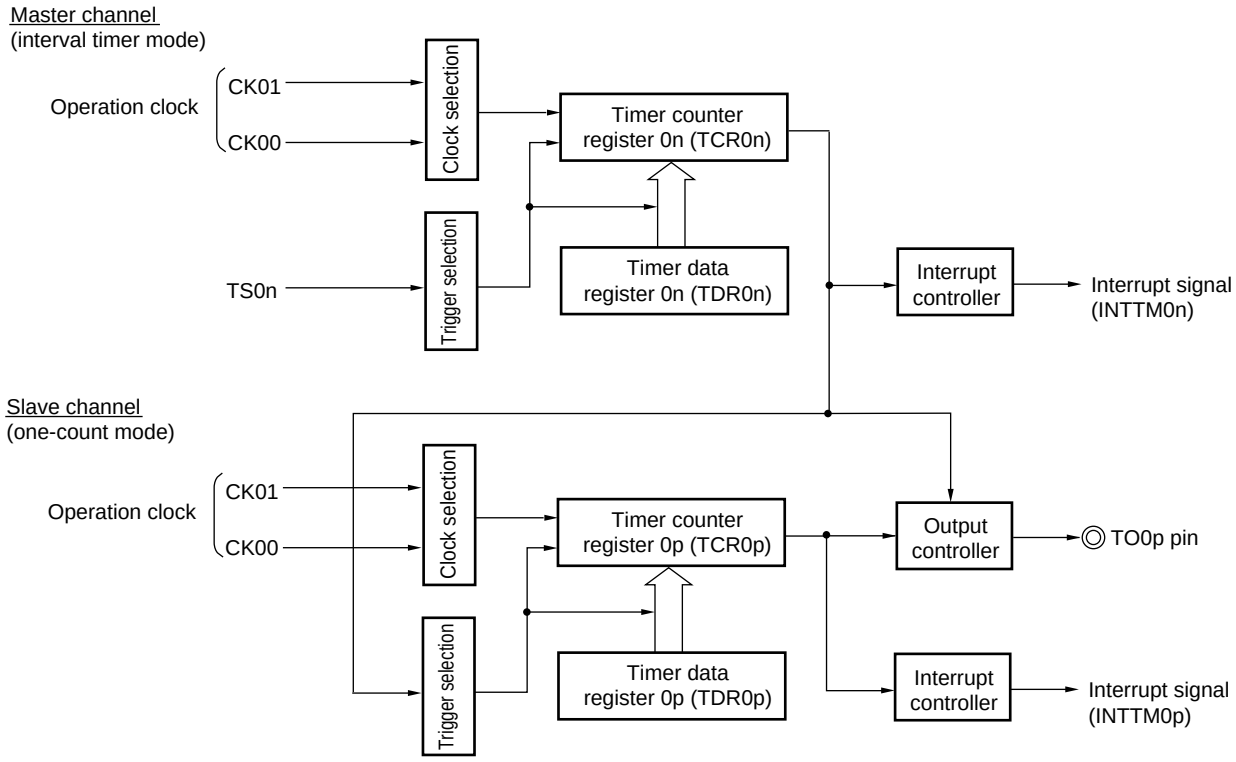
PWM output (TO0p) goes to the active level one clock after the master channel generates INTTM0n and goes to the inactive level when the TCR0p register of the slave channel becomes 0000H.

Caution To rewrite both timer data register 0n (TDR0n) of the master channel and the TDR0p register of the slave channel, a write access is necessary two times. The timing at which the values of the TDR0n and TDR0p registers are loaded to the TCR0n and TCR0p registers is upon occurrence of INTTM0n of the master channel. Thus, when rewriting is performed split before and after occurrence of INTTM0n of the master channel, the TO0p pin cannot output the expected waveform. To rewrite both the TDR0n register of the master and the TDR0p register of the slave, therefore, be sure to rewrite both the registers immediately after INTTM0n is generated from the master channel.

Remark n: Channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

Downloaded from Arrow.com.

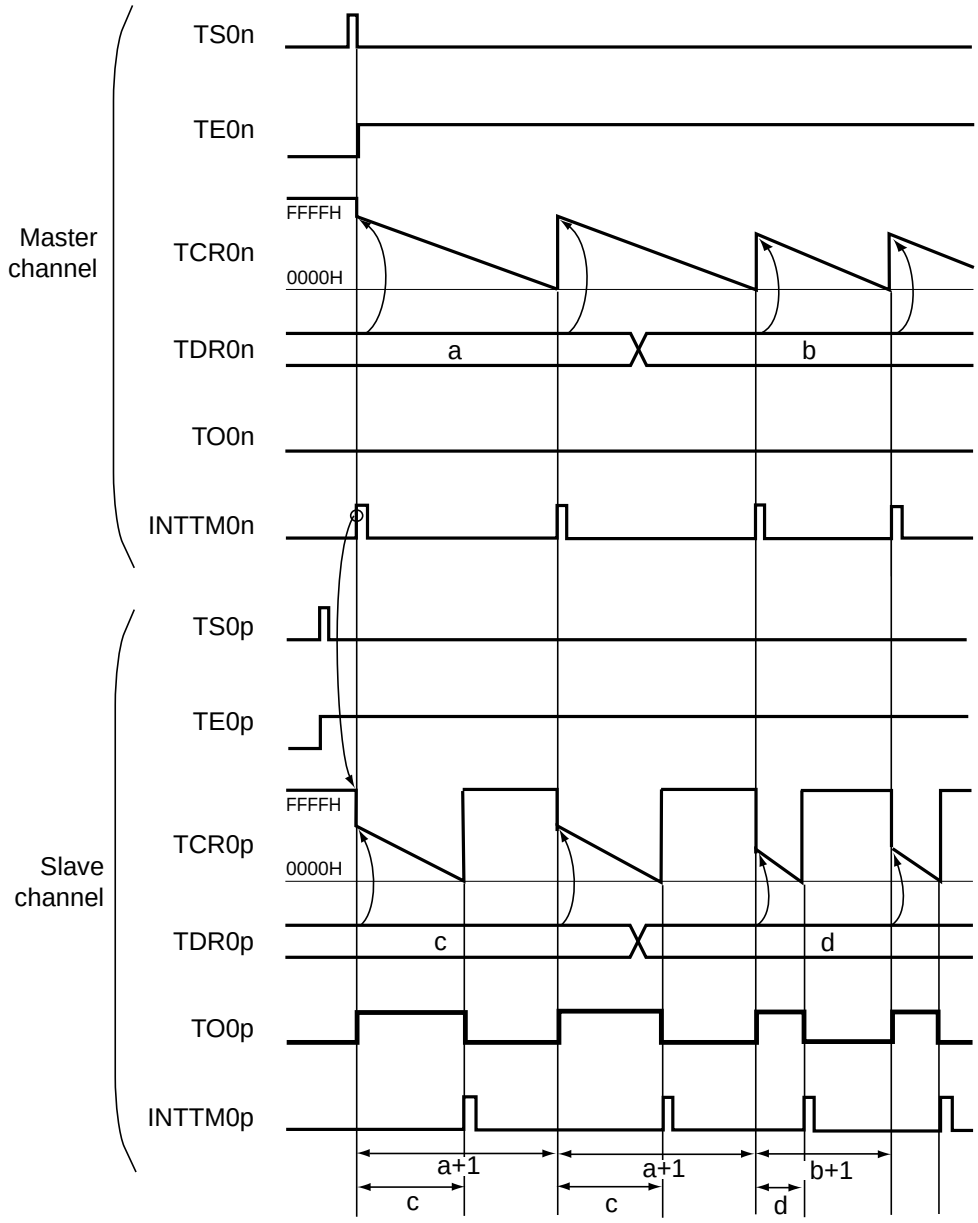
Figure 6-67. Block Diagram of Operation as PWM Function



Remark n: Channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

Downloaded from Arrow.com.

Figure 6-68. Example of Basic Timing of Operation as PWM Function



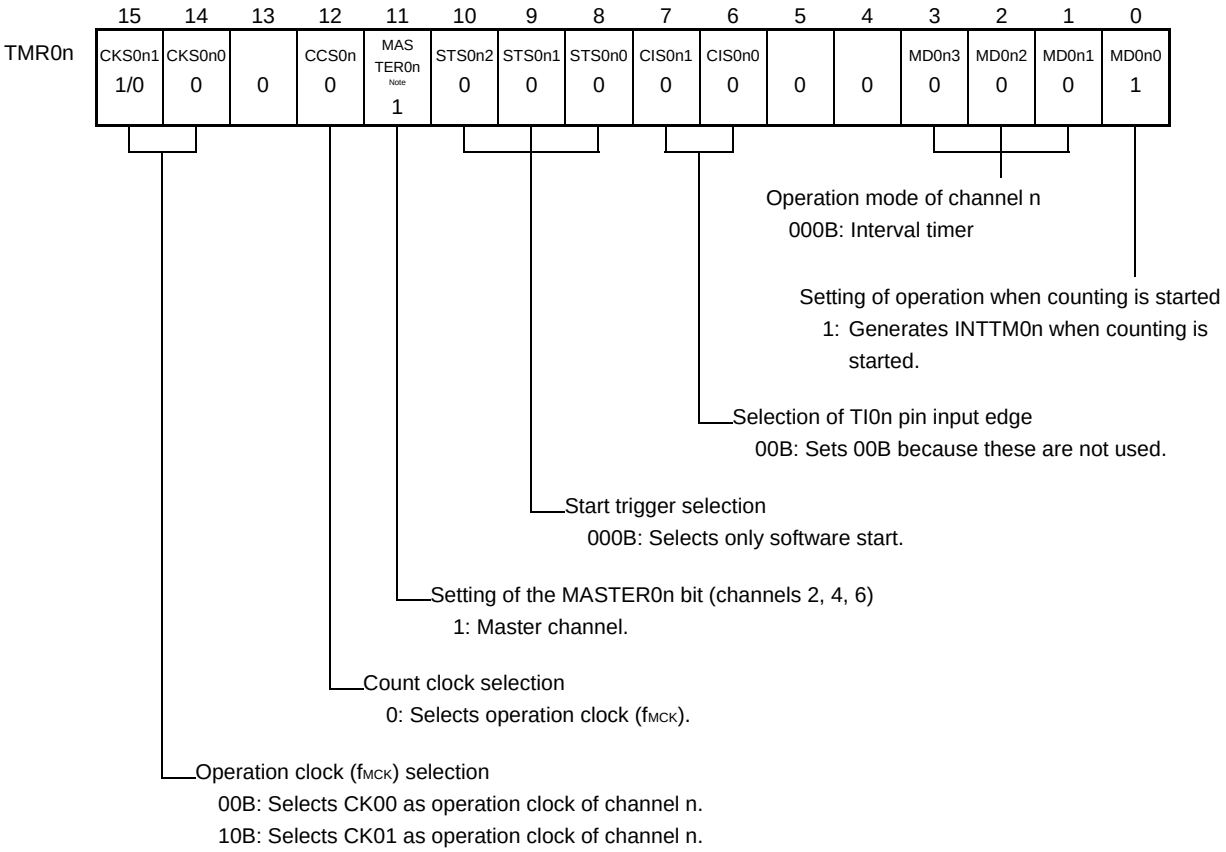
Remarks 1. n: Channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

2. TS0n, TS0p: Bit n, p of timer channel start register 0 (TS0)
TE0n, TE0p: Bit n, p of timer channel enable status register 0 (TE0)
TCR0n, TCR0p: Timer count registers mn, mp (TCR0n, TCR0p)
TDR0n, TDR0p: Timer data registers mn, mp (TDR0n, TDR0p)
TO0n, TO0p: TO0n and TO0p pins output signal

Downloaded from Arrow.com.

Figure 6-69. Example of Set Contents of Registers When PWM Function (Master Channel) Is Used

(a) Timer mode register 0n (TMR0n)



<R>

(b) Timer output register 0 (TO0)

	Bit n	
TO0	TO0n 0	0: Outputs 0 from TO0n.

(c) Timer output enable register 0 (TOE0)

	Bit n	
TOE0	TOE0n 0	0: Stops the TO0n output operation by counting operation.

(d) Timer output level register 0 (TOL0)

	Bit n	
TOL0	TOL0n 0	0: Cleared to 0 when TOM0n = 0 (master channel output mode).

(e) Timer output mode register 0 (TOM0)

	Bit n	
TOM0	TOM0n 0	0: Sets master channel output mode.

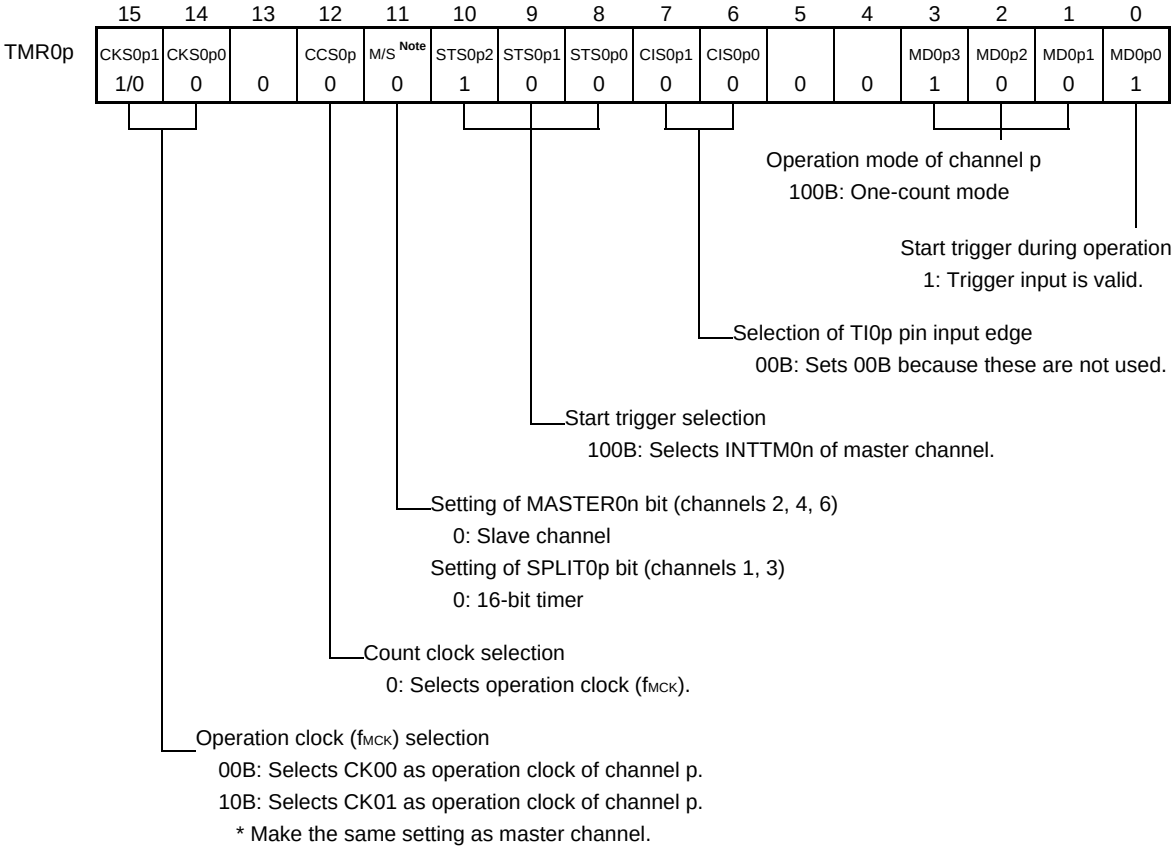
Note TMR02, TMR04, TMR06: MASTER0n = 1
TMR00: Fixed to 0

<R> **Remark** n: Channel number (n = 0, 2, 4, 6)

Downloaded from Arrow.com.

Figure 6-70. Example of Set Contents of Registers When PWM Function (Slave Channel) Is Used

(a) Timer mode register 0p (TMR0p)



<R>

(b) Timer output register 0 (TO0)

	Bit p	
TO0	TO0p 1/0	0: Outputs 0 from TO0p. 1: Outputs 1 from TO0p.

(c) Timer output enable register 0 (TOE0)

	Bit p	
TOE0	TOE0p 1/0	0: Stops the TO0p output operation by counting operation. 1: Enables the TO0p output operation by counting operation.

(d) Timer output level register 0 (TOL0)

	Bit p	
TOL0	TOL0p 1/0	0: Positive logic output (active-high) 1: Negative logic output (active-low)

(e) Timer output mode register 0 (TOM0)

	Bit p	
TOM0	TOM0p 1	1: Sets the slave channel output mode.

<R> **Note** TMR02, TMR04, TMR06: MASTER0n = 1
TMR01, TMR03: SPLIT0p bit
TMR05, TMR07: 0 fixed

Remark n: Channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

Downloaded from Arrow.com.

Figure 6-71. Operation Procedure When PWM Function Is Used (1/2)

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN bit of peripheral enable register 0 (PER0) to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register 0 (TPS0). Determines clock frequencies of CK00 and CK01.	
Channel default setting	Sets timer mode registers mn, mp (TMR0n, TMR0p) of two channels to be used (determines operation mode of channels). An interval (period) value is set to timer data register 0n (TDR0n) of the master channel, and a duty factor is set to the TDR0p register of the slave channel.	Channel stops operating. (Clock is supplied and some power is consumed.)
	Sets slave channel. The TOM0p bit of timer output mode register 0 (TOM0) is set to 1 (slave channel output mode). Sets the TOL0p bit. Sets the TO0p bit and determines default level of the TO0p output. →	The TO0p pin goes into Hi-Z output state. The TO0p default setting level is output when the port mode register is in output mode and the port register is 0.
	Sets the TOE0p bit to 1 and enables operation of TO0p. →	TO0p does not change because channel stops operating.
	Clears the port register and port mode register to 0. →	The TO0p pin outputs the TO0p set level.

Remark n: Channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

Downloaded from Arrow.com.

Figure 6-71. Operation Procedure When PWM Function Is Used (2/2)

	Software Operation	Hardware Status
Operation start	Sets the TOE0p bit (slave) to 1 (only when operation is resumed). The TS0n (master) and TS0p (slave) bits of timer channel start register 0 (TS0) are set to 1 at the same time. →	TE0n = 1, TE0p = 1 → When the master channel starts counting, INTTM0n is generated. Triggered by this interrupt, the slave channel also starts counting.
	The TS0n and TS0p bits automatically return to 0 because they are trigger bits.	
During operation	Set values of the TMR0n and TMR0p registers, TOM0n, TOM0p, TOL0n, and TOL0p bits cannot be changed. Set values of the TDR0n and TDR0p registers can be changed after INTTM0n of the master channel is generated. The TCR0n and TCR0p registers can always be read. The TSR0n and TSR0p registers are not used.	The counter of the master channel loads the TDR0n register value to timer count register 0n (TCR0n), and counts down. When the count value reaches TCR0n = 0000H, INTTM0n output is generated. At the same time, the value of the TDR0n register is loaded to the TCR0n register, and the counter starts counting down again. At the slave channel, the value of the TDR0p register is loaded to the TCR0p register, triggered by INTTM0n of the master channel, and the counter starts counting down. The output level of TO0p becomes active one count clock after generation of the INTTM0n output from the master channel. It becomes inactive when TCR0p = 0000H, and the counting operation is stopped. After that, the above operation is repeated.
Operation stop	The TT0n (master) and TT0p (slave) bits are set to 1 at the same time. →	TE0n, TE0p = 0, and count operation stops. The TCR0n and TCR0p registers hold count value and stop. The TO0p output is not initialized but holds current status.
	The TOE0p bit of slave channel is cleared to 0 and value is set to the TO0p bit. →	The TO0p pin outputs the TO0p set level.
TAU stop	To hold the TO0p pin output level Clears the TO0p bit to 0 after the value to be held is set to the port register. →	The TO0p pin output level is held by port function.
	When holding the TO0p pin output level is not necessary Setting not required.	
	The TAU0EN bit of the PER0 register is cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TO0p bit is cleared to 0 and the TO0p pin is set to port mode.)

Operation is resumed.

Remark n: Channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

Downloaded from Arrow.com.

6.9.3 Operation as multiple PWM output function

By extending the PWM function and using multiple slave channels, many PWM waveforms with different duty values can be output.

For example, when using two slave channels, the period and duty factor of an output pulse can be calculated by the following expressions.

Pulse period = {Set value of TDR0n (master) + 1} × Count clock period
Duty factor 1 [%] = {Set value of TDR0p (slave 1)}/{Set value of TDR0n (master) + 1} × 100
Duty factor 2 [%] = {Set value of TDR0q (slave 2)}/{Set value of TDR0n (master) + 1} × 100

<R> **Remark** Although the duty factor exceeds 100% if the set value of TDR0p (slave 1) > {set value of TDR0n (master) + 1} or if the {set value of TDR0q (slave 2)} > {set value of TDR0n (master) + 1}, it is summarized into 100% output.

Timer count register 0n (TCR0n) of the master channel operates in the interval timer mode and counts the periods.

The TCR0p register of the slave channel 1 operates in one-count mode, counts the duty factor, and outputs a PWM waveform from the TO0p pin. The TCR0p register loads the value of timer data register 0p (TDR0p), using INTTM0n of the master channel as a start trigger, and starts counting down. When TCR0p = 0000H, TCR0p outputs INTTM0p and stops counting until the next start trigger (INTTM0n of the master channel) has been input. The output level of TO0p becomes active one count clock after generation of INTTM0n from the master channel, and inactive when TCR0p = 0000H.

In the same way as the TCR0p register of the slave channel 1, the TCR0q register of the slave channel 2 operates in one-count mode, counts the duty factor, and outputs a PWM waveform from the TO0q pin. The TCR0q register loads the value of the TDR0q register, using INTTM0n of the master channel as a start trigger, and starts counting down. When TCR0q = 0000H, the TCR0q register outputs INTTM0q and stops counting until the next start trigger (INTTM0n of the master channel) has been input. The output level of TO0q becomes active one count clock after generation of INTTM0n from the master channel, and inactive when TCR0q = 0000H.

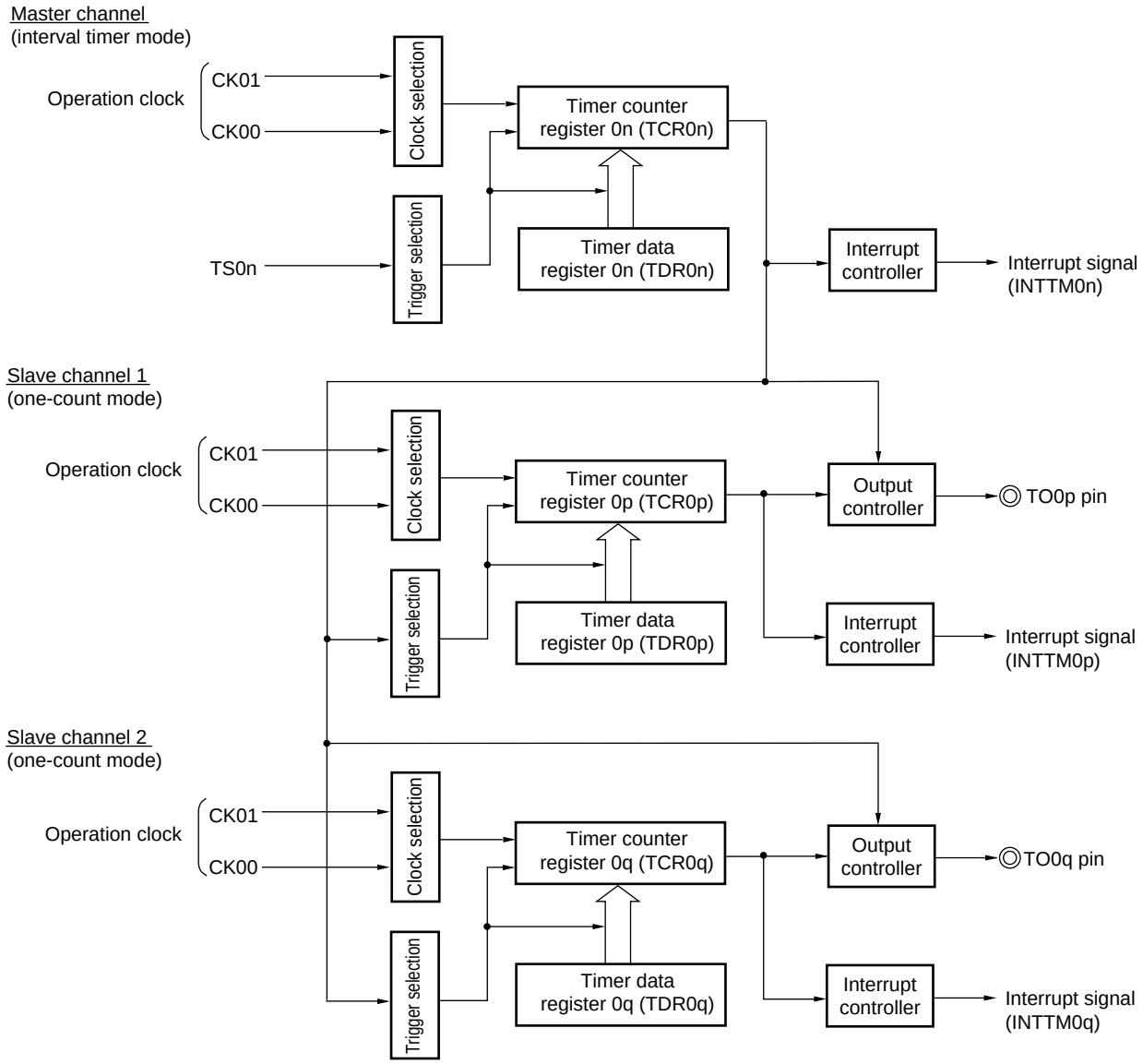
When channel 0 is used as the master channel as above, up to seven types of PWM signals can be output at the same time.

Caution To rewrite both timer data register 0n (TDR0n) of the master channel and the TDR0p register of the slave channel 1, write access is necessary at least twice. Since the values of the TDR0n and TDR0p registers are loaded to the TCR0n and TCR0p registers after INTTM0n is generated from the master channel, if rewriting is performed separately before and after generation of INTTM0n from the master channel, the TO0p pin cannot output the expected waveform. To rewrite both the TDR0n register of the master and the TDR0p register of the slave, be sure to rewrite both the registers immediately after INTTM0n is generated from the master channel (This applies also to the TDR0q register of the slave channel 2).

Remark n: Channel number (n = 0, 2, 4)
p: Slave channel number 1, q: Slave channel number 2
$n < p < q \leq 7$ (Where p and q are consecutive integers greater than n)

Downloaded from Arrow.com.

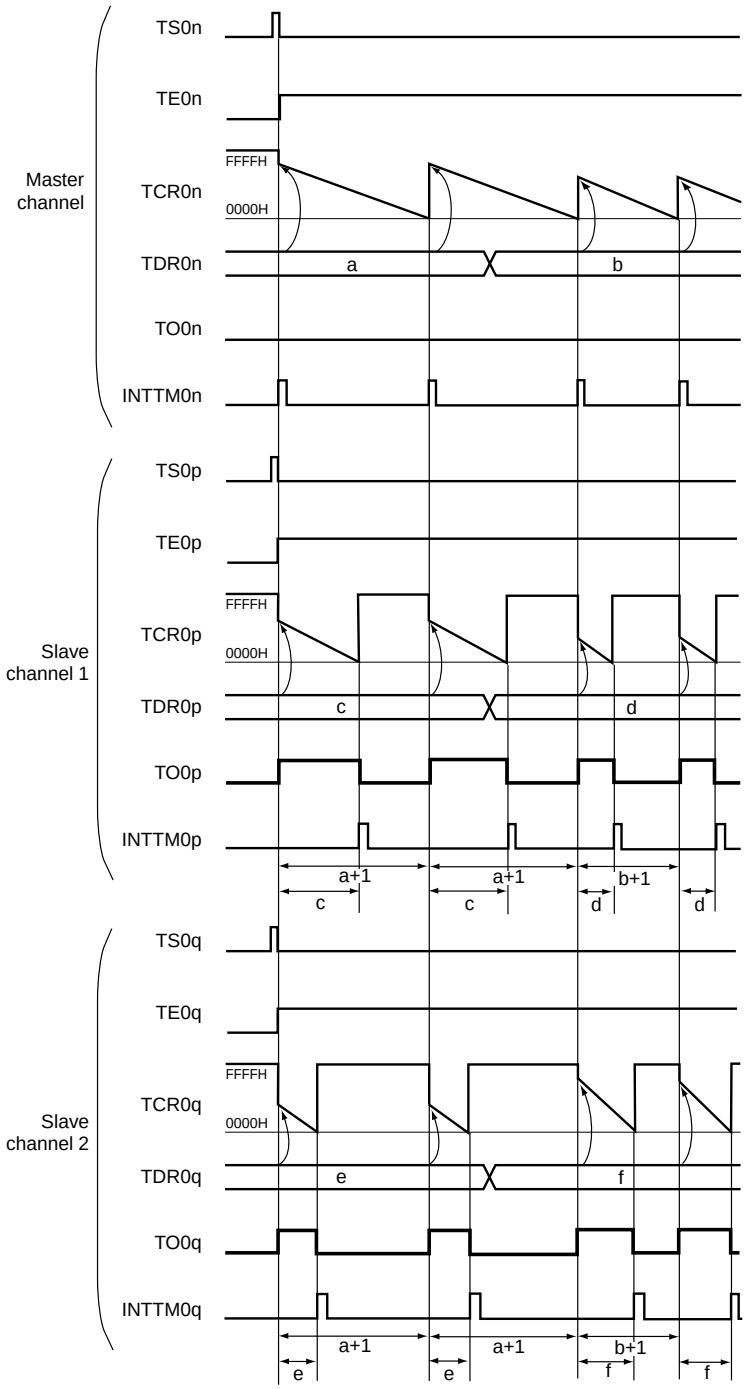
Figure 6-72. Block Diagram of Operation as Multiple PWM Output Function (output two types of PWMs)



Remark n: Channel number (n = 0, 2, 4)
p: Slave channel number 1, q: Slave channel number 2
n < p < q ≤ 7 (Where p and q are consecutive integers greater than n)

Downloaded from Arrow.com.

Figure 6-73. Example of Basic Timing of Operation as Multiple PWM Output Function (Output two types of PWMs) (1/2)



Remarks 1. n: Channel number (n = 0, 2, 4)

p: Slave channel number 1, q: Slave channel number 2

n < p < q ≤ 7 (Where p and q are consecutive integers greater than n)

2. TS0n, TS0p, TS0q: Bit n, p, q of timer channel start register 0 (TS0)

TE0n, TE0p, TE0q: Bit n, p, q of timer channel enable status register 0 (TE0)

TCR0n, TCR0p, TCR0q: Timer count registers mn, mp, mq (TCR0n, TCR0p, TCR0q)

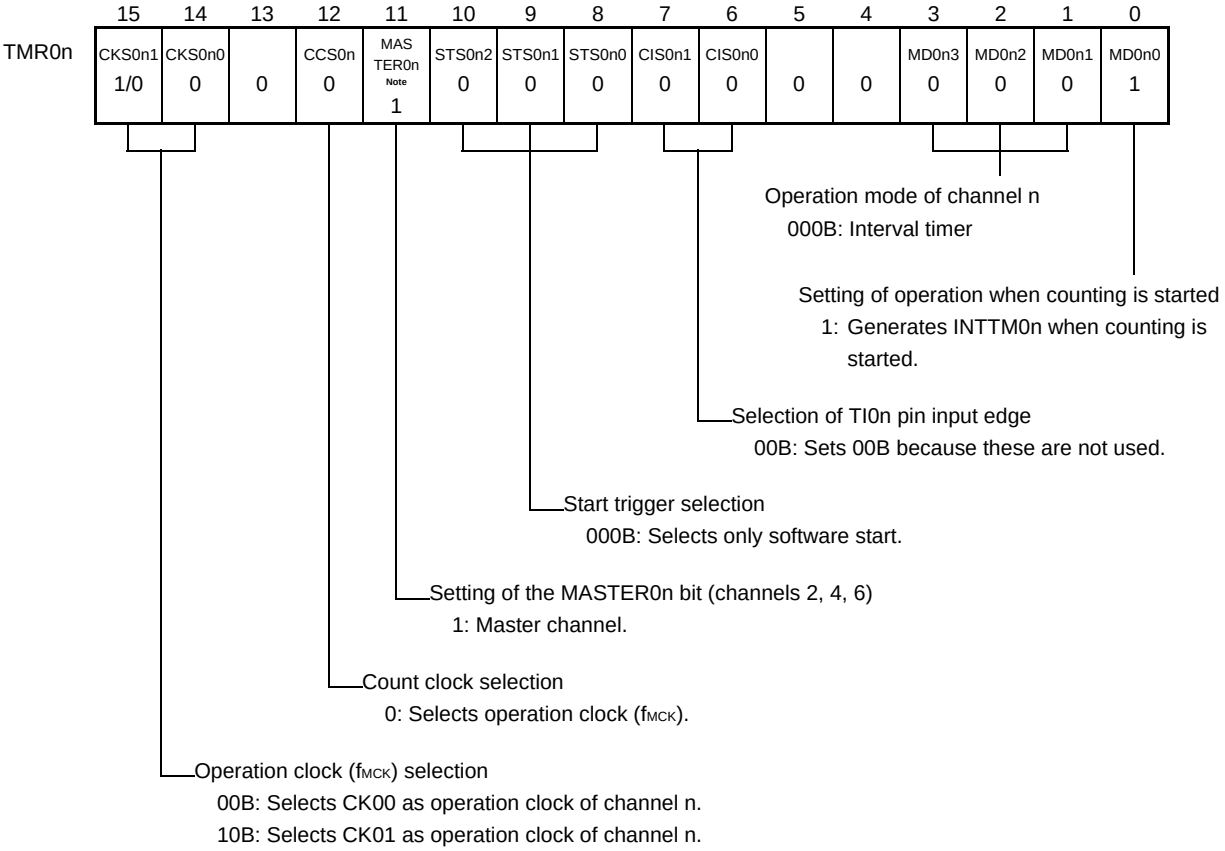
TDR0n, TDR0p, TDR0q: Timer data registers mn, mp, mq (TDR0n, TDR0p, TDR0q)

TO0n, TO0p, TO0q: TO0n, TO0p, and TO0q pins output signal

Downloaded from Arrow.com.

Figure 6-74. Example of Set Contents of Registers When Multiple PWM Output Function (Master Channel) Is Used

(a) Timer mode register 0n (TMR0n)



<R>

(b) Timer output register 0 (TO0)

	Bit n	
TO0	TO0n 0	0: Outputs 0 from TO0n.

(c) Timer output enable register 0 (TOE0)

	Bit n	
TOE0	TOE0n 0	0: Stops the TO0n output operation by counting operation.

(d) Timer output level register 0 (TOL0)

	Bit n	
TOL0	TOL0n 0	0: Cleared to 0 when TOM0n = 0 (master channel output mode).

(e) Timer output mode register 0 (TOM0)

	Bit n	
TOM0	TOM0n 0	0: Sets master channel output mode.

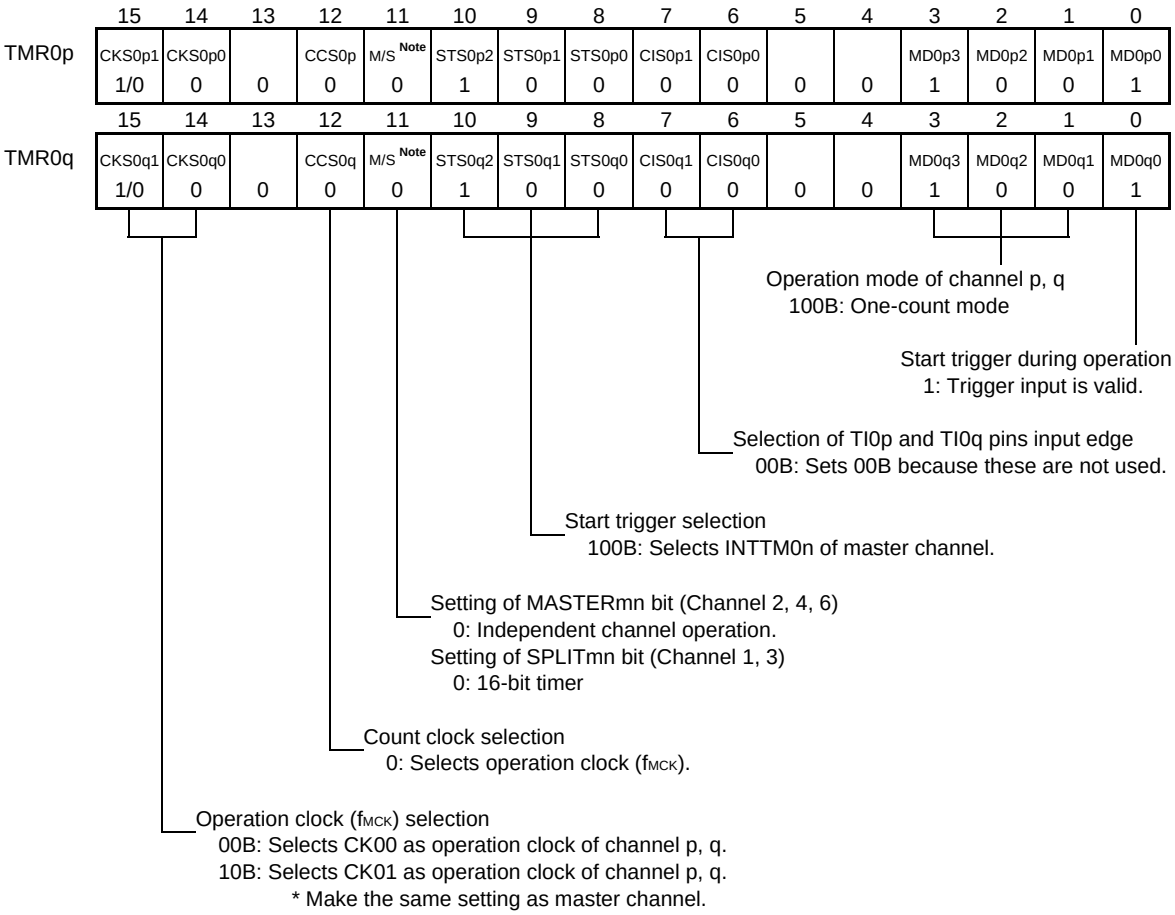
<R> **Note** TMR02, TMR04, TMR06: MASTER0n = 1
TMR00: Fixed to 0

Remark n: Channel number (n = 0, 2, 4)

Downloaded from Arrow.com.

Figure 6-75. Example of Set Contents of Registers
When Multiple PWM Output Function (Slave Channel) Is Used (output two types of PWMs)

(a) Timer mode register 0p, mq (TMR0p, TMR0q)



(b) Timer output register 0 (TO0)

	Bit q	Bit p	
TO0	TO0q	TO0p	0: Outputs 0 from TO0p or TO0q. 1: Outputs 1 from TO0p or TO0q.
	1/0	1/0	

(c) Timer output enable register 0 (TOE0)

	Bit q	Bit p	
TOE0	TOE0q	TOE0p	0: Stops the TO0p or TO0q output operation by counting operation. 1: Enables the TO0p or TO0q output operation by counting operation.
	1/0	1/0	

(d) Timer output level register 0 (TOL0)

	Bit q	Bit p	
TOL0	TOL0q	TOL0p	0: Positive logic output (active-high) 1: Negative logic output (active-low)
	1/0	1/0	

(e) Timer output mode register 0 (TOM0)

	Bit q	Bit p	
TOM0	TOM0q	TOM0p	1: Sets the slave channel output mode.
	1	1	

Note TMR02, TMR04, TMR06: MASTER0n bit
TMR01, TMR03: SPLIT0n bit
TMR05, TMR07: 0 fixed

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2, 4)
p: Slave channel number 1, q: Slave channel number 2
n < p < q ≤ 7 (Where p and q are consecutive integers greater than n)

Downloaded from Arrow.com.

Figure 6-76. Operation Procedure When Multiple PWM Output Function Is Used (1/2)

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN bit of peripheral enable register 0 (PER0) to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register 0 (TPS0). Determines clock frequencies of CK00 and CK01.	
Channel default setting	Sets timer mode registers mn, mp, 0q (TMR0n, TMR0p, TMR0q) of each channel to be used (determines operation mode of channels). An interval (period) value is set to timer data register 0n (TDR0n) of the master channel, and a duty factor is set to the TDR0p and TDR0q registers of the slave channels.	Channel stops operating. (Clock is supplied and some power is consumed.)
	Sets slave channels. The TOM0p and TOM0q bits of timer output mode register 0 (TOM0) are set to 1 (slave channel output mode). Sets the TOL0p and TOL0q bits. Sets the TO0p and TO0q bits and determines default level of the TO0p and TO0q outputs. →	The TO0p and TO0q pins go into Hi-Z output state. The TO0p and TO0q default setting levels are output when the port mode register is in output mode and the port register is 0.
	Sets the TOE0p and TOE0q bits to 1 and enables operation of TO0p and TO0q. →	TO0p and TO0q do not change because channels stop operating.
	Clears the port register and port mode register to 0. →	The TO0p and TO0q pins output the TO0p and TO0q set levels.

Remark n: Channel number (n = 0, 2, 4)
p: Slave channel number 1, q: Slave channel number 2
n < p < q ≤ 7 (Where p and q are a consecutive integer greater than n)

Downloaded from Arrow.com.

Figure 6-77. Operation Procedure When Multiple PWM Output Function Is Used (2/2)

	Software Operation	Hardware Status
Operation start	(Sets the TOE0p and TOE0q (slave) bits to 1 only when resuming operation.) The TS0n bit (master), and TS0p and TS0q (slave) bits of timer channel start register 0 (TS0) are set to 1 at the same time. The TS0n, TS0p, and TS0q bits automatically return to 0 because they are trigger bits.	→ TE0n = 1, TE0p, TE0q = 1 When the master channel starts counting, INTTM0n is generated. Triggered by this interrupt, the slave channel also starts counting.
During operation	Set values of the TMR0n, TMR0p, TMR0q registers, TOM0n, TOM0p, TOM0q, TOL0n, TOL0p, and TOL0q bits cannot be changed. Set values of the TDR0n, TDR0p, and TDR0q registers can be changed after INTTM0n of the master channel is generated. The TCR0n, TCR0p, and TCR0q registers can always be read. The TSR0n, TSR0p, and TSR0q registers are not used.	The counter of the master channel loads the TDR0n register value to timer count register 0n (TCR0n) and counts down. When the count value reaches TCR0n = 0000H, INTTM0n output is generated. At the same time, the value of the TDR0n register is loaded to the TCR0n register, and the counter starts counting down again. At the slave channel 1, the values of the TDR0p register are transferred to the TCR0p register, triggered by INTTM0n of the master channel, and the counter starts counting down. The output levels of TO0p become active one count clock after generation of the INTTM0n output from the master channel. It becomes inactive when TCR0p = 0000H, and the counting operation is stopped. At the slave channel 2, the values of the TDR0q register are transferred to TCR0q register, triggered by INTTM0n of the master channel, and the counter starts counting down. The output levels of TO0q become active one count clock after generation of the INTTM0n output from the master channel. It becomes inactive when TCR0q = 0000H, and the counting operation is stopped. After that, the above operation is repeated.
Operation stop	The TT0n bit (master), TT0p, and TT0q (slave) bits are set to 1 at the same time. The TT0n, TT0p, and TT0q bits automatically return to 0 because they are trigger bits.	→ TE0n, TE0p, TE0q = 0, and count operation stops. The TCR0n, TCR0p, and TCR0q registers hold count value and stop. The TO0p and TO0q output are not initialized but hold current status.
	The TOE0p and TOE0q bits of slave channels are cleared to 0 and value is set to the TO0p and TO0q bits.	→ The TO0p and TO0q pins output the TO0p and TO0q set levels.
TAU stop	To hold the TO0p and TO0q pin output levels Clears the TO0p and TO0q bits to 0 after the value to be held is set to the port register. When holding the TO0p and TO0q pin output levels are not necessary Setting not required	→ The TO0p and TO0q pin output levels are held by port function.
	The TAU0EN bit of the PER0 register is cleared to 0.	→ Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TO0p and TO0q bits are cleared to 0 and the TO0p and TO0q pins are set to port mode.)

Operation is resumed.

Remark n: Channel number (n = 0, 2, 4)
p: Slave channel number 1, q: Slave channel number 2
n < p < q ≤ 7 (Where p and q are a consecutive integer greater than n)

Downloaded from Arrow.com.

6.10 Cautions When Using Timer Array Unit

6.10.1 Cautions When Using Timer output

Pins may be assigned multiplexed timer output and other alternate functions. If you intend to use a timer output in such a case, set the outputs from all other multiplexed pin functions to their initial values.

<R> For details, see **4.5 Register Settings When Using Alternate Function**.

Downloaded from Arrow.com.

CHAPTER 7 12-BIT INTERVAL TIMER

7.1 Functions of 12-bit Interval Timer

An interrupt (INTIT) is generated at any previously specified time interval. It can be utilized for wakeup from STOP mode and triggering an A/D converter's SNOOZE mode.

7.2 Configuration of 12-bit Interval Timer

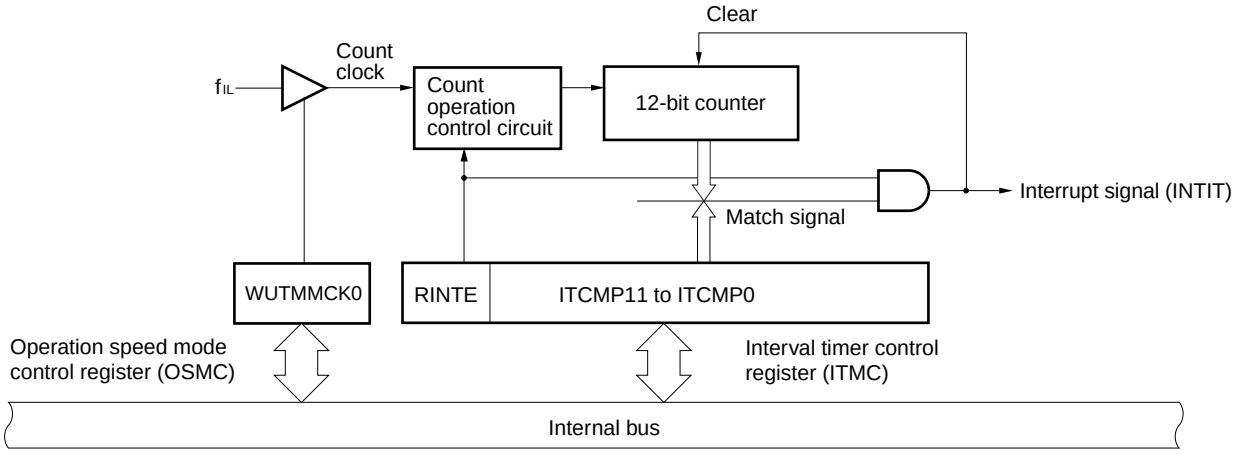
The 12-bit interval timer includes the following hardware.

Table 7-1. Configuration of 12-bit Interval Timer

Item	Configuration
Counter	12-bit counter
Control registers	Peripheral enable register 0 (PER0)
	Operation speed mode control register (OSMC)
	Interval timer control register (ITMC)

<R>

Figure 7-1. Block Diagram of 12-bit Interval Timer



7.3 Registers Controlling 12-bit Interval Timer

The 12-bit interval timer is controlled by the following registers.

- Peripheral enable register 0 (PER0)
- Operation speed mode control register (OSMC)
- Interval timer control register (ITMC)

Downloaded from Arrow.com.

7.3.1 Peripheral enable register 0 (PER0)

This register is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to the hardware that is not used is also stopped so as to decrease the power consumption and noise.

When using the 12-bit interval timer, be sure to set bit 7 (TMKAEN) to 1 at first.

The PER0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 7-2. Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H After reset: 00H R/W

Symbol	<7>	6	<5>	<4>	<3>	<2>	1	<0>
PER0	TMKAEN	0	ADCEN	IICA0EN	SAU1EN Note	SAU0EN	0	TAU0EN

TMKAEN	Control of clock supply for 12-bit interval timer
0	Stops clock supply. • SFR used by the 12-bit interval timer cannot be written. • The 12-bit interval timer is in the reset status.
1	Enables clock supply. • SFR used by the 12-bit interval timer can be read and written.

Note Provided only in 30-pin products.

<R> **Cautions 1. When using the 12-bit interval timer, be sure to first set the TMKAEN bit to 1 after starting supply of the low-speed on-chip oscillator clock (WUTMMCK0 = 1) and then set the interval timer control register (ITMC). If TMKAEN bit = 0, writing to the registers controlling the 12-bit interval timer is ignored, and, even if the register is read, only the default value is read (except the operation speed mode control register (OSMC)).**

2. Be sure to clear the following bits to 0.
20- or 24-pin products: Bits 1, 3, and 6
30-pin products: Bits 1 and 6

7.3.2 Operation speed mode control register (OSMC)

The WUTMMCK0 bit can be used to control supply of the 12-bit interval timer operation clock.

The OSMC register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 7-3. Format of Operation Speed Mode Control Register (OSMC)

Address: F00F3H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
OSMC	0	0	0	WUTMMCK0	0	0	0	0

WUTMMCK0	Supply of operation clock for 12-bit interval timer
0	Clock supply stop.
1	Low-speed on-chip oscillator clock (f_{IL}) supply

Downloaded from Arrow.com.

7.3.3 Interval timer control register (ITMC)

This register is used to set up the starting and stopping of the 12-bit interval timer operation and to specify the timer compare value.

The ITMC register can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0FFFH.

Figure 7-4. Format of Interval Timer Control Register (ITMC)

Address: FFF90H After reset: 0FFFH R/W

Symbol	15	14	13	12	11 to 0
ITMC	RINTE	0	0	0	ITCMP11 to ITCMP0

RINTE	12-bit interval timer operation control
0	Count operation stopped (count clear)
1	Count operation started

ITCMP11 to ITCMP0	Specification of the 12-bit interval timer compare value
001H	These bits generate an interrupt at the fixed cycle (count clock cycles x (ITCMP setting + 1)).
•	
•	
•	
FFFH1	
Example interrupt cycles when 001H or FFFH is specified for ITCMP11 to ITCMP0 • ITCMP11 to ITCMP0 = 001H, count clock: when f_{IL} = 15 kHz 1/15 [kHz] × (1 + 1) ÷ 0.1333 [ms] = 133.3 [μs] • ITCMP11 to ITCMP0 = FFFH, count clock: when f_{IL} = 15 kHz 1/15 [kHz] × (4095 + 1) ÷ 273 [ms]	

Cautions 1. When RINTE bit is changed from 0 to 1, set WUTMMCK0 bit of OSMC register to 1 before the change so that the operation clock is established.

2. Before changing the RINTE bit from 1 to 0, use the interrupt mask flag register to disable the INTIT interrupt servicing. When operation starts (0 to 1) again, clear the TMKAIF flag, and then enable the interrupt servicing.

3. The value read from the RINTE bit is applied one count clock cycle after setting the RINTE bit.

4. When setting the RINTE bit after returned from standby mode and entering standby mode again, confirm that the written value of the RINTE bit is reflected, or wait that more than one clock of the count clock has elapsed after returned from standby mode. Then enter standby mode.

5. Only change the setting of the ITCMP11 to ITCMP0 bits when RINTE = 0.
However, it is possible to change the settings of the ITCMP11 to ITCMP0 bits at the same time as when changing RINTE from 0 to 1 or 1 to 0.

Downloaded from Arrow.com.

7.4 12-bit Interval Timer Operation

7.4.1 12-bit interval timer operation timing

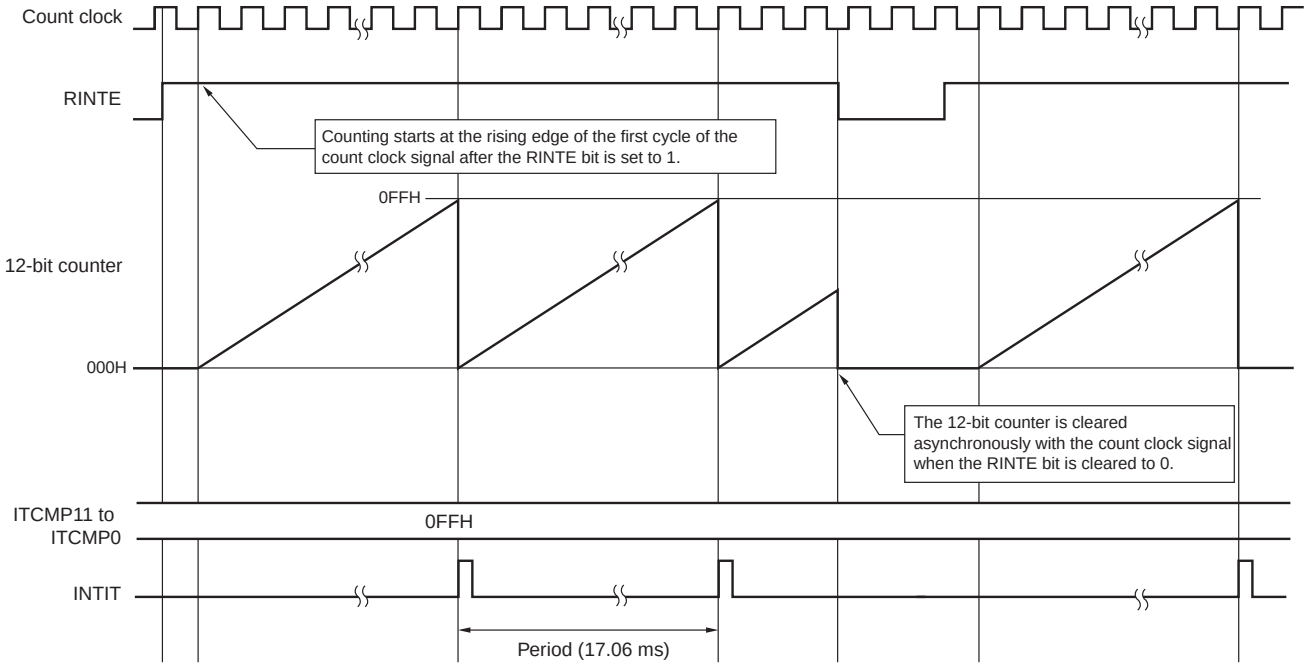
The count value specified for the ITCMP11 to ITCMP0 bits is used as an interval to operate an 12-bit interval timer that repeatedly generates interrupt requests (INTIT).

When the RINTE bit is set to 1, the 12-bit counter starts counting.

When the 12-bit counter value matches the value specified for the ITCMP11 to ITCMP0 bits, the 12-bit counter value is cleared to 0, counting continues, and an interrupt request signal (INTIT) is generated at the same time.

The basic operation of the 12-bit interval timer is shown in Figure 7-5.

Figure 7-5. 12-bit Interval Timer Operation Timing
(ITCMP11 to ITCMP0 = 0FFH, count clock: f_{IL} = 15 kHz)



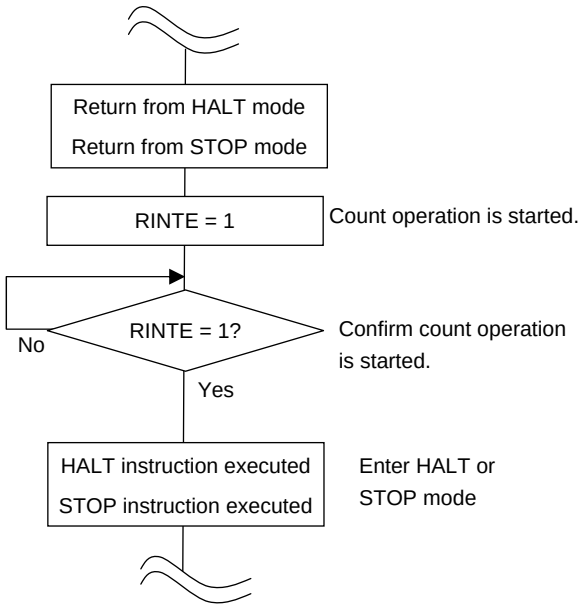
Downloaded from Arrow.com.

<R> 7.4.2 Start of count operation and re-enter to HALT/STOP mode after returned from HALT/STOP mode

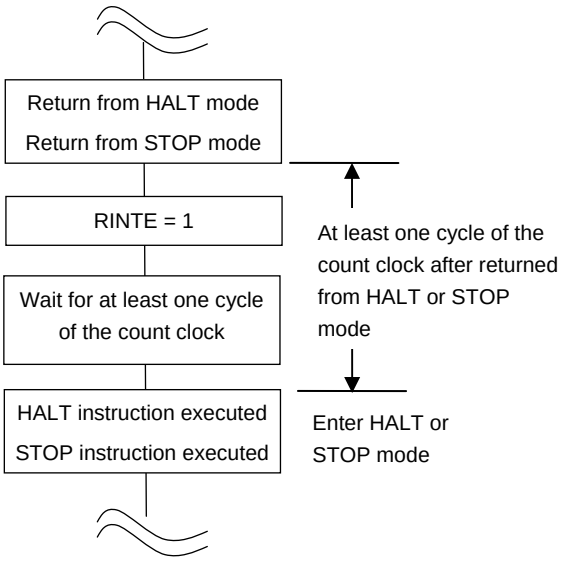
When setting the RINTE bit after returned from HALT or STOP mode and entering HALT or STOP mode again, write 1 to the RINTE bit, and confirm the written value of the RINTE bit is reflected or wait for at least one cycle of the count clock. Then, enter HALT or STOP mode.

- After setting RINTE to 1, confirm by polling that the RINTE bit has become 1, and then enter HALT or STOP mode (see Example 1 in Figure 7-6).
- After setting RINTE to 1, wait for at least one cycle of the count clock and then enter HALT or STOP mode (see Example 2 in Figure 7-6).

Figure 7-6. Procedure of entering to HALT or STOP mode after setting RINTE to 1



Example 1



Example 2

Downloaded from Arrow.com.

CHAPTER 8 CLOCK OUTPUT/BUZZER OUTPUT CONTROLLER

<R> The number of output pins of the clock output and buzzer output controllers differs, depending on the product.

Output pin	20- and 24-pin	30-pin
PCLBUZ0	√	√
PCLBUZ1	–	√

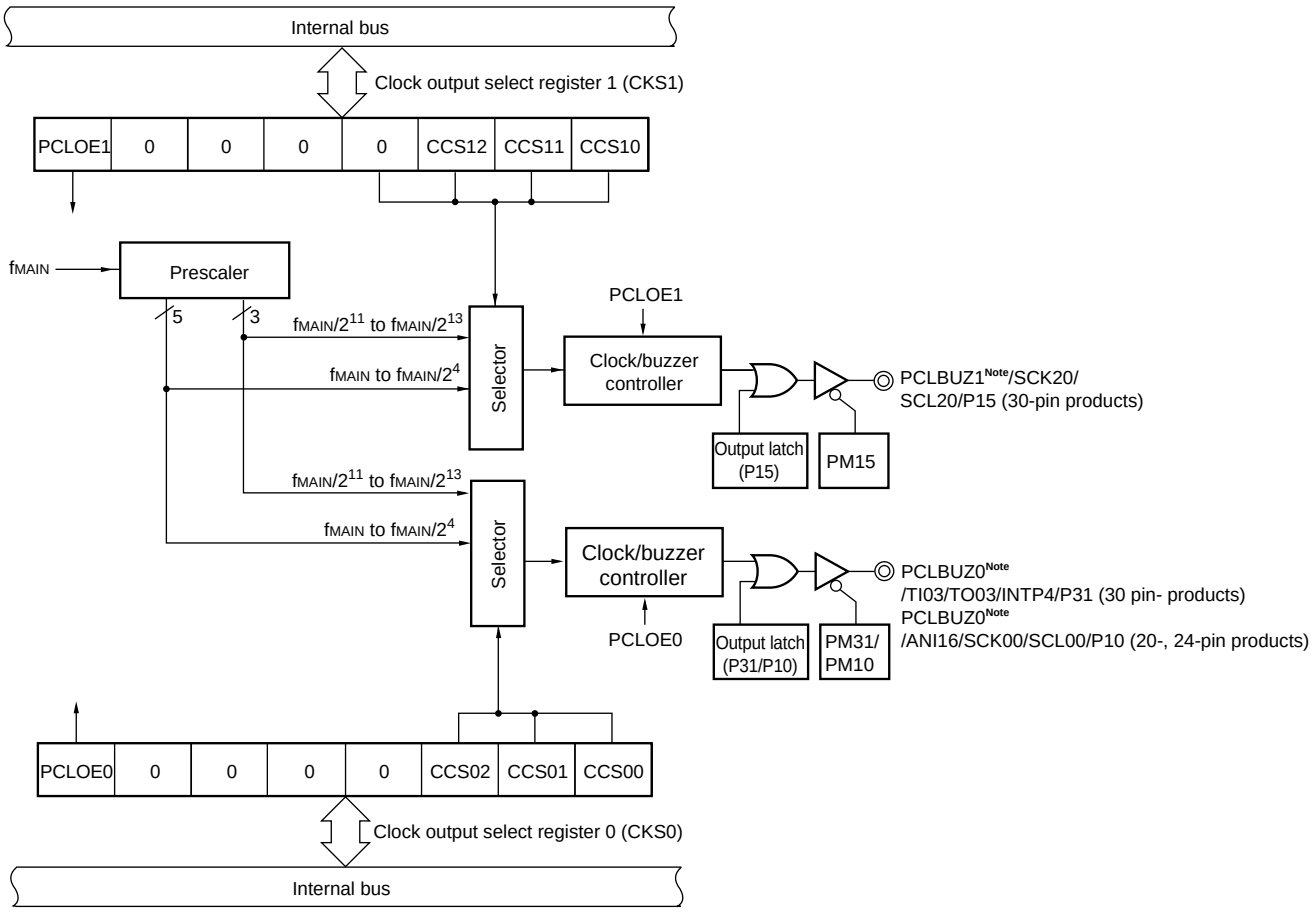
Caution Most of the following descriptions in this chapter use the 30-pin as an example.

8.1 Functions of Clock Output/Buzzer Output Controller

<R> The clock output controller is intended for clock output for supply to peripheral ICs.
Buzzer output is a function to output a square wave of buzzer frequency.
One pin can be used to output a clock or buzzer sound.
The PCLBUZn pin outputs a clock selected by clock output select register n (CKSn).
Figure 8-1 shows the block diagram of clock output/buzzer output controller.

Remark n = 0, 1

Figure 8-1. Block Diagram of Clock Output/Buzzer Output Controller



Note For output frequencies available from PCLBUZ0 and PCLBUZ1, refer to **28.4** or **29.4 AC Characteristics**.
PCLBUZ1 output function is available only in 30-pin products.

Downloaded from Arrow.com.

8.2 Configuration of Clock Output/Buzzer Output Controller

The clock output/buzzer output controller includes the following hardware.

Table 8-1. Configuration of Clock Output/Buzzer Output Controller

Item	Configuration
Control registers	Clock output select register n (CKSn) Port mode register 1, 3 (PM1, PM3) Port register 1, 3 (P1, P3)

8.3 Registers Controlling Clock Output/Buzzer Output Controller

The following registers are used to control the clock output/buzzer output controller.

- Clock output select register n (CKSn)
- Port mode register 1, 3 (PM1, PM3)
- <R> Port register 1, 3 (P1, P3)

Downloaded from Arrow.com.

8.3.1 Clock output select register n (CKSn)

These registers set output enable/disable for clock output or for the buzzer frequency output pin (PCLBUZn), and set the output clock.

Select the clock to be output from the PCLBUZn pin by using the CKSn register.

The CKSn register are set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Figure 8-2. Format of Clock Output Select Register n (CKSn)

Address: FFFA5H (CKS0), FFFA6 (CKS1) Note 1 After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	0
CKSn	PCLOEn	0	0	0	0	CCSn2	CCSn1	CCSn0

PCLOEn	PCLBUZn pin output enable/disable specification
0	Output disable (default)
1	Output enable

<R>

CCSn2	CCSn1	CCSn0	PCLBUZn pin output clock selection					
				f_{MAIN} (MHz)				
				5	10	16	20	24
0	0	0	f_{MAIN}	5 MHz	10 MHz Note 2	16 MHz Note 2	Setting prohibited Note 2	Setting prohibited Note 2
0	0	1	$f_{MAIN}/2$	2.5 MHz	5 MHz	8 MHz Note 2	10 MHz Note 2	12 MHz Note 2
0	1	0	$f_{MAIN}/2^2$	1.25 MHz	2.5 MHz	4 MHz	5 MHz	6 MHz
0	1	1	$f_{MAIN}/2^3$	625 kHz	1.25 MHz	2 MHz	2.5 MHz	3 MHz
1	0	0	$f_{MAIN}/2^4$	313 kHz	625 kHz	1 MHz	1.25 MHz	1.5 MHz
1	0	1	$f_{MAIN}/2^{11}$	2.44 kHz	4.88 kHz	7.81 kHz	9.77 kHz	11.7 kHz
1	1	0	$f_{MAIN}/2^{12}$	1.22 kHz	2.44 kHz	3.91 kHz	4.88 kHz	5.86 kHz
1	1	1	$f_{MAIN}/2^{13}$	610 Hz	1.22 kHz	1.95 kHz	2.44 kHz	2.93 kHz

Notes 1. 30-pin products only.

<R> **2.** Use the output clock within a range of 16 MHz. For detail, refer to **28.4** or **29.4 AC characteristics**.

Cautions 1. Change the output clock after disabling clock output (PCLOEn = 0).

<R> **2. To shift to STOP mode, set PCLOEn = 0 before executing the STOP instruction.**

Remarks 1. n = 0, 1

2. f_{MAIN}: Main system clock frequency

Downloaded from Arrow.com.

<R> **8.3.2 Registers controlling port functions of pins to be used for clock or buzzer output**

Using a port pin for clock or buzzer output requires setting of the registers that control the port functions multiplexed on the target pin (port mode register (PMxx), port register (Pxx)). For details, **see 4.3.1 Port mode registers (PMxx)** and **4.3.2 Port registers (Pxx)**.

Specifically, using a port pin with a multiplexed clock or buzzer output function (P10/PCLBUZ0 (20- and 24-pin products), P15/PCLBUZ1, P31/PCLBUZ0 (30-pin products)) for clock or buzzer output, requires setting the corresponding bits in the port mode register (PMxx) and port register (Pxx) to 0.

Example: When P15/PCLBUZ1 is to be used for clock or buzzer output
Set the PM15 bit of port mode register 1 to 0.
Set the P15 bit of port register 1 to 0.

Downloaded from Arrow.com.

8.4 Operations of Clock Output/Buzzer Output Controller

One pin can be used to output a clock or buzzer sound.

The PCLBUZ0 pin outputs a clock/buzzer selected by the clock output select register 0 (CKS0).

The PCLBUZ1 pin outputs a clock/buzzer selected by the clock output select register 1 (CKS1).

8.4.1 Operation as output pin

The PCLBUZn pin is output as the following procedure.

<R> <1> Set 0 in the bit of the port mode register (PMxx) and port register (Px) which correspond to the port which has a pin used as the PCLBUZn pin.

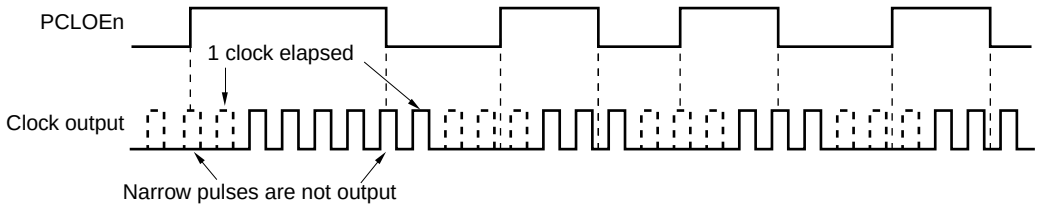
<2> Select the output frequency with bits 0 to 2 (CCSn0 to CCSn2) of the clock output select register (CKSn) of the PCLBUZn pin (output in disabled).

<3> Set bit 7 (PCLOEn) of the CKSn register to 1 to enable clock/buzzer output.

Remarks 1. The controller used for outputting the clock starts or stops outputting the clock one clock after enabling or disabling clock output (PCLOEn bit) is switched. At this time, pulses with a narrow width are not output. Figure 8-3 shows enabling or stopping output using the PCLOEn bit and the timing of outputting the clock.

2. n = 0 or 1

<R> **Figure 8-3. Timing of Outputting Clock from PCLBUZn Pin**



<R> 8.5 Cautions of Clock Output/Buzzer Output Controller

If STOP mode is entered within 1.5 clock cycles output from the PCLBUZn pin after the output is disabled (PCLOEn = 0), the PCLBUZn output width becomes shorter.

Downloaded from Arrow.com.

CHAPTER 9 WATCHDOG TIMER

9.1 Functions of Watchdog Timer

The counting operation of the watchdog timer is set by the option byte (000C0H).

<R> The watchdog timer operates on the low-speed on-chip oscillator clock (f_{IL}).

The watchdog timer is used to detect an inadvertent program loop. If a program loop is detected, an internal reset signal is generated.

Program loop is detected in the following cases.

- If the watchdog timer counter overflows
- If a 1-bit manipulation instruction is executed on the watchdog timer enable register (WDTE)
- If data other than “ACH” is written to the WDTE register
- If data is written to the WDTE register during a window close period

When a reset occurs due to the watchdog timer, bit 4 (WDTRF) of the reset control flag register (RESF) is set to 1. For details of the RESF register, see **CHAPTER 18 RESET FUNCTION**.

When 75%+1/2f_{IL} of the overflow time is reached, an interval interrupt can be generated.

Downloaded from Arrow.com.

9.2 Configuration of Watchdog Timer

The watchdog timer includes the following hardware.

Table 9-1. Configuration of Watchdog Timer

<R>

Item	Configuration
Counter	Internal counter (17 bits)
Control register	Watchdog timer enable register (WDTE)

How the counter operation is controlled, overflow time, window open period, and interval interrupt are set by the option byte.

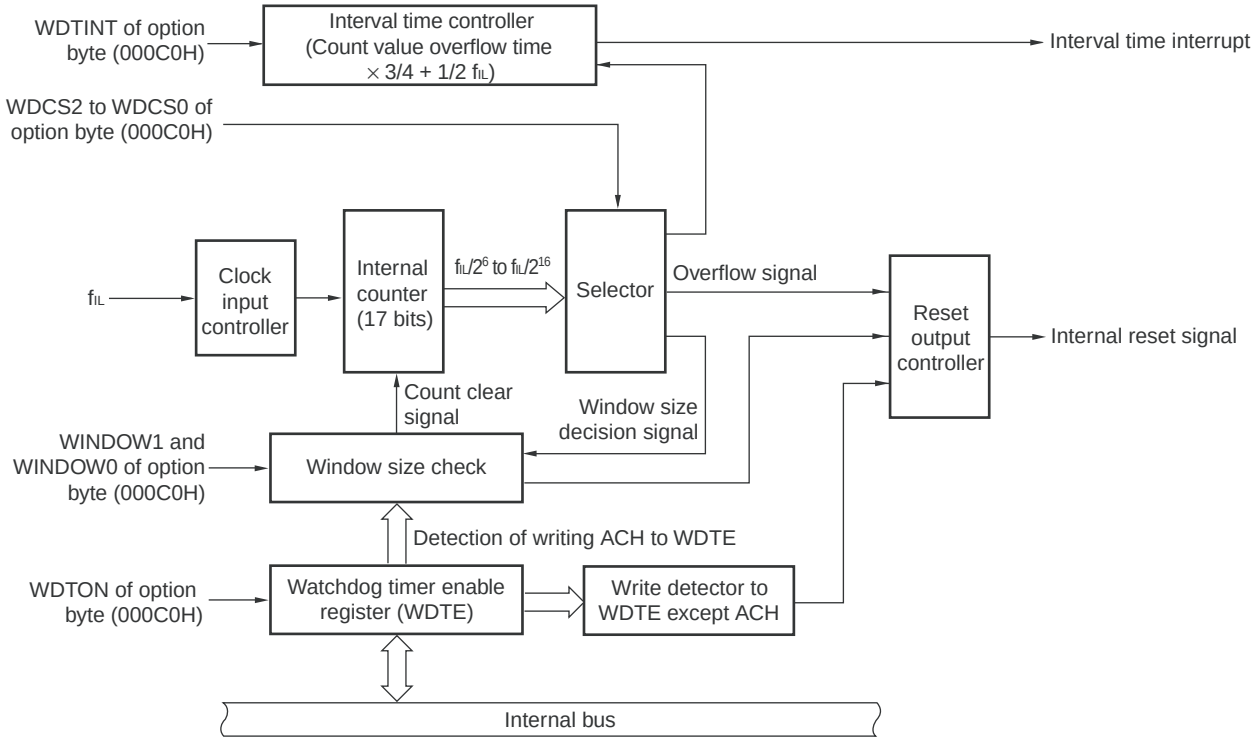
Table 9-2. Setting of Option Bytes and Watchdog Timer

Setting of Watchdog Timer	Option Byte (000C0H)
Watchdog timer interval interrupt	Bit 7 (WDTINT)
Window open period	Bits 6 and 5 (WINDOW1, WINDOW0)
Controlling counter operation of watchdog timer	Bit 4 (WDTON)
Overflow time of watchdog timer	Bits 3 to 1 (WDCS2 to WDCS0)
Controlling counter operation of watchdog timer (in HALT/STOP mode)	Bit 0 (WDSTBYON)

Remark For the option byte, see **CHAPTER 23 OPTION BYTE**.

<R>

Figure 9-1. Block Diagram of Watchdog Timer



Remark f_{IL}: Low-speed on-chip oscillator clock

Downloaded from Arrow.com.

9.3 Register Controlling Watchdog Timer

The watchdog timer is controlled by the watchdog timer enable register (WDTE).

9.3.1 Watchdog timer enable register (WDTE)

Writing “ACH” to the WDTE register clears the watchdog timer counter and starts counting again.

This register can be set by an 8-bit memory manipulation instruction.

Reset signal generation sets this register to 9AH or 1AH[Note].

Figure 9-2. Format of Watchdog Timer Enable Register (WDTE)

Address: FFFABH After reset: 9AH/1AH[Note] R/W

Symbol	7	6	5	4	3	2	1	0
WDTE								

Note The WDTE register reset value differs depending on the WDTON bit setting value of the option byte (000C0H). To operate watchdog timer, set the WDTON bit to 1.

WDTON Bit Setting Value	WDTE Register Reset Value
0 (watchdog timer count operation disabled)	1AH
1 (watchdog timer count operation enabled)	9AH

Cautions
1. **If a value other than “ACH” is written to the WDTE register, an internal reset signal is generated.**
2. **If a 1-bit memory manipulation instruction is executed for the WDTE register, an internal reset signal is generated.**
3. **The value read from the WDTE register is 9AH/1AH (this differs from the written value (ACH)).**

Downloaded from Arrow.com.

9.4 Operation of Watchdog Timer

9.4.1 Controlling operation of watchdog timer

1. When the watchdog timer is used, its operation is specified by the option byte (000C0H).

   - Enable counting operation of the watchdog timer by setting bit 4 (WDTON) of the option byte (000C0H) to 1 (the counter starts operating after a reset release) (for details, see **CHAPTER 23**).

WDTON	Watchdog Timer Counter
0	Counter operation disabled (counting stopped after reset)
1	Counter operation enabled (counting started after reset)

   - Set an overflow time by using bits 3 to 1 (WDCS2 to WDCS0) of the option byte (000C0H) (for details, see **9.4.2** and **CHAPTER 23**).
   - Set a window open period by using bits 6 and 5 (WINDOW1 and WINDOW0) of the option byte (000C0H) (for details, see **9.4.3** and **CHAPTER 23**).

2. After a reset release, the watchdog timer starts counting.
3. By writing “ACH” to the watchdog timer enable register (WDTE) after the watchdog timer starts counting and before the overflow time set by the option byte, the watchdog timer is cleared and starts counting again.
4. After that, write the WDTE register the second time or later after a reset release during the window open period. If the WDTE register is written during a window close period, an internal reset signal is generated.
5. If the overflow time expires without “ACH” written to the WDTE register, an internal reset signal is generated.
   An internal reset signal is generated in the following cases.

   - If a 1-bit manipulation instruction is executed on the WDTE register
   - If data other than “ACH” is written to the WDTE register

Cautions 1. When data is written to the watchdog timer enable register (WDTE) for the first time after reset release, the watchdog timer is cleared in any timing regardless of the window open time, as long as the register is written before the overflow time, and the watchdog timer starts counting again.

<R> **2. After "ACH" is written to the WDTE register, an error of up to 2 clocks (f_{IL}) may occur before the watchdog timer is cleared.**

3. The watchdog timer can be cleared immediately before the count value overflows.

(**Cautions 4** is listed on the next page.)

Downloaded from Arrow.com.

Cautions 4. The operation of the watchdog timer in the HALT, STOP, and SNOOZE modes differs as follows depending on the set value of bit 0 (WDSTBYON) of the option byte (000C0H).

	WDSTBYON = 0	WDSTBYON = 1
In HALT mode	Watchdog timer operation stops.	Watchdog timer operation continues.
In STOP mode		
In SNOOZE mode		

If WDSTBYON = 0, the watchdog timer resumes counting after the HALT or STOP mode is released. At this time, the counter is cleared to 0 and counting starts.
When operating with the X1 oscillation clock after releasing the STOP mode, the CPU starts operating after the oscillation stabilization time has elapsed.
Therefore, if the period between the STOP mode release and the watchdog timer overflow is short, an overflow occurs during the oscillation stabilization time, causing a reset.
Consequently, set the overflow time in consideration of the oscillation stabilization time when operating with the X1 oscillation clock and when the watchdog timer is to be cleared after the STOP mode release by an interval interrupt.

9.4.2 Setting overflow time of watchdog timer

Set the overflow time of the watchdog timer by using bits 3 to 1 (WDCS2 to WDCS0) of the option byte (000C0H).

If an overflow occurs, an internal reset signal is generated. The present count is cleared and the watchdog timer starts counting again by writing “ACH” to the watchdog timer enable register (WDTE) during the window open period before the overflow time.

The following overflow times can be set.

Table 9-3. Setting of Overflow Time of Watchdog Timer

WDCS2	WDCS1	WDCS0	Overflow Time of Watchdog Timer (f_{IL} = 17.25 kHz (MAX.))
0	0	0	$2^6/f_{IL}$ (3.71 ms)
0	0	1	$2^7/f_{IL}$ (7.42 ms)
0	1	0	$2^8/f_{IL}$ (14.84 ms)
0	1	1	$2^9/f_{IL}$ (29.68 ms)
1	0	0	$2^{11}/f_{IL}$ (118.72 ms)
1	0	1	$2^{13}/f_{IL}$ (474.89 ms)
1	1	0	$2^{14}/f_{IL}$ (949.79 ms)
1	1	1	$2^{16}/f_{IL}$ (3799.18 ms)

Remark f_{IL}: Low-speed on-chip oscillator clock frequency

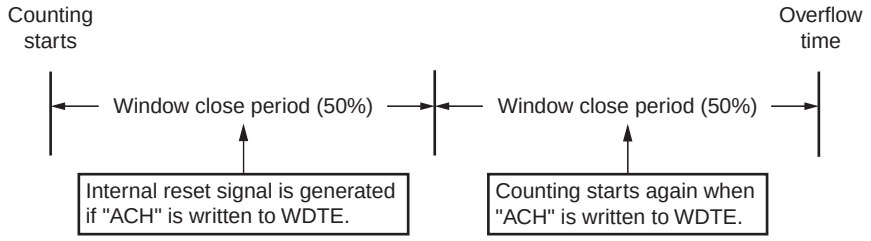
Downloaded from Arrow.com.

9.4.3 Setting window open period of watchdog timer

Set the window open period of the watchdog timer by using bits 6 and 5 (WINDOW1, WINDOW0) of the option byte (000C0H). The outline of the window is as follows.

- If “ACH” is written to the watchdog timer enable register (WDTE) during the window open period, the watchdog timer is cleared and starts counting again.
- Even if “ACH” is written to the WDTE register during the window close period, an abnormality is detected and an internal reset signal is generated.

Example: If the window open period is 50%



Caution When data is written to the WDTE register for the first time after reset release, the watchdog timer is cleared in any timing regardless of the window open time, as long as the register is written before the overflow time, and the watchdog timer starts counting again.

The window open period can be set is as follows.

Table 9-4. Setting Window Open Period of Watchdog Timer

WINDOW1	WINDOW0	Window Open Period of Watchdog Timer
0	0	Setting prohibited
0	1	50%
1	0	75%
1	1	100%

Caution When bit 0 (WDSTBYON) of the option byte (000C0H) = 0, the window open period is 100% regardless of the values of the WINDOW1 and WINDOW0 bits.

Remark If the overflow time is set to $2^9/f_{IL}$, the window close time and open time are as follows.

	Setting of Window Open Period		
	50%	75%	100%
Window close time	0 to 20.08 ms	0 to 10.04 ms	None
Window open time	20.08 to 29.68 ms	10.04 to 29.68 ms	0 to 29.68 ms

<When window open period is 50%>

- Overflow time:
  $2^9/f_{IL}$ (MAX.) = 2^9/17.25 kHz (MAX.) = 29.68 ms
- Window close time:
  0 to $2^9/f_{IL}$ (MIN.) × (1 – 0.5) = 0 to 2^9/12.75 kHz × 0.5 = 0 to 20.08 ms
- Window open time:
  $2^9/f_{IL}$ (MIN.) × (1 – 0.5) to $2^9/f_{IL}$ (MAX.) = 2^9/12.75 kHz × 0.5 to 2^9/17.25 kHz = 20.08 to 29.68 ms

Downloaded from Arrow.com.

9.4.4 Setting watchdog timer interval interrupt

Depending on the setting of bit 7 (WDTINT) of an option byte (000C0H), an interval interrupt (INTWDTI) can be generated when 75% + 1/2 f_{IL} of the overflow time is reached.

Table 9-5. Setting of Watchdog Timer Interval Interrupt

WDTINT	Use of Watchdog Timer Interval Interrupt
0	Interval interrupt is used.
1	Interval interrupt is generated when 75% + 1/2 f_{IL} of overflow time is reached.

Caution When operating with the X1 oscillation clock after releasing the STOP mode, the CPU starts operating after the oscillation stabilization time has elapsed.
Therefore, if the period between the STOP mode release and the watchdog timer overflow is short, an overflow occurs during the oscillation stabilization time, causing a reset.
Consequently, set the overflow time in consideration of the oscillation stabilization time when operating with the X1 oscillation clock and when the watchdog timer is to be cleared after the STOP mode release by an interval interrupt.

Remark The watchdog timer continues counting even after INTWDTI is generated (until ACH is written to the watchdog timer enable register (WDTE)). If ACH is not written to the WDTE register before the overflow time, an internal reset signal is generated.

Downloaded from Arrow.com.

CHAPTER 10 A/D CONVERTER

The number of analog input channels of the A/D converter differs depending on the product.

	20- and 24-pin	30-pin
Analog input channels	11 ch (ANI0 to ANI3, ANI6 to ANI22)	8 ch (ANI0 to ANI3, ANI16 to ANI19)

10.1 Function of A/D Converter

The A/D converter converts analog input signals into digital values, and is configured to control analog inputs, including up to 11 channels of A/D converter analog inputs (ANI0 to ANI3 and ANI16 to ANI22). 10-bit or 8-bit resolution can be selected by using the ADTYP bit of the A/D converter mode register 2 (ADM2).

The A/D converter has the following function.

<R> • **10-bit/8-bit resolution A/D conversion**

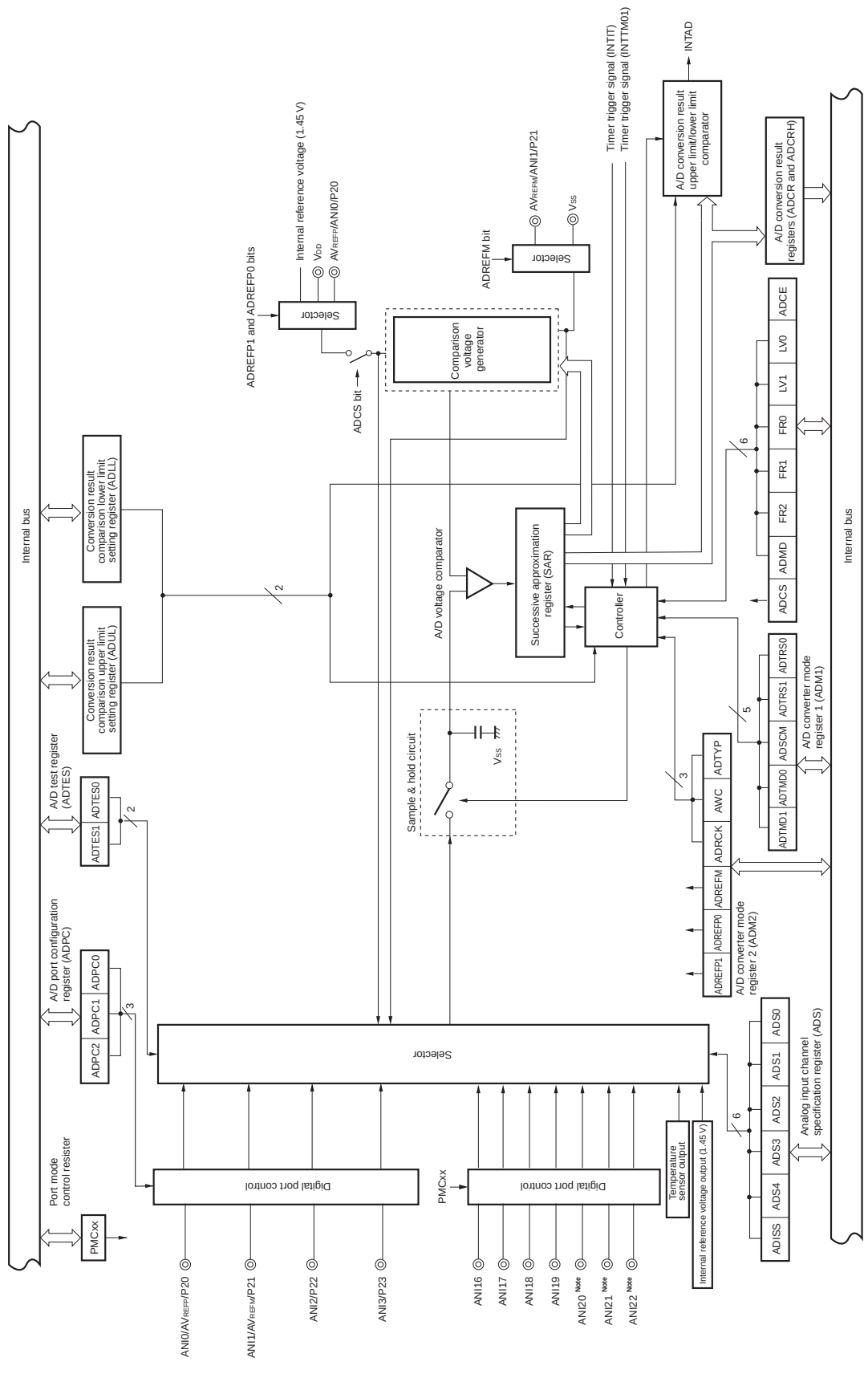
10-bit or 8-bit resolution A/D conversion is carried out repeatedly for one analog input channel selected from ANI0 to ANI3 and ANI16 to ANI22 (ANI0 to ANI3 and ANI16 to ANI19 for 30-pin products). Each time an A/D conversion operation ends, an interrupt request (INTAD) is generated (when in the select mode).

Various A/D conversion modes can be specified by using the mode combinations below.

Trigger mode	Software trigger	Conversion is started by software.
	Hardware trigger no-wait mode	Conversion is started by detecting a hardware trigger.
	Hardware trigger wait mode	The power is turned on by detecting a hardware trigger while the system is off and in the conversion standby state, and conversion is then started automatically after the A/D power supply stabilization wait time passes. Select hardware trigger wait mode when using the SNOOZE mode function.
Channel selection mode	Select mode	A/D conversion is performed on the analog input of one channel.
	Scan mode	A/D conversion is performed on the analog input of four channels in order. ANI0 to ANI3 can be selected as analog input.
Conversion operation mode	One-shot conversion mode	A/D conversion is performed on the selected channel once.
	Sequential conversion mode	A/D conversion is sequentially performed on the selected channels until it is stopped by software.
Operation voltage mode	Normal 1 or normal 2 mode	Conversion is done in the operation voltage range of $2.7\ V \leq V_{DD} \leq 5.5\ V$.
	Low-voltage 1 or low-voltage 2 mode	Conversion is done in the operation voltage range of $1.8\ V \leq V_{DD} \leq 5.5\ V$. Select this mode for conversion at a low voltage. Because the operation voltage is low, it is internally boosted during conversion.
Sapling time selection	Sampling clock cycles: 7 f_{AD}	The sampling time in normal 1 or low-voltage 1 mode is seven cycles of the conversion clock (f_{AD}). Select this mode when the output impedance of the analog input source is high and a longer sampling time is required.
	Sampling clock cycles: 5 f_{AD}	The sampling time in normal 2 or low-voltage 2 mode is five cycles of the conversion clock (f_{AD}). Select this mode when enough sampling time is ensured (for example, when the output impedance of the analog input source is low).

Downloaded from Arrow.com.

Figure 10-1. Block Diagram of A/D Converter



Note Provided only in 20- or 24-pin products.

<R>

Downloaded from Arrow.com.

10.2 Configuration of A/D Converter

The A/D converter includes the following hardware.

(1) ANI0 to ANI3 and ANI16 to ANI22 pins

These are the analog input pins of the 11 channels of the A/D converter (for 30-pin products, these are analog input pins of the 8 channels of ANI0 to ANI3 and ANI16 to ANI19 pins). They input analog signals to be converted into digital signals. Pins other than the one selected as the analog input pin can be used as I/O port pins.

(2) Sample & hold circuit

The sample & hold circuit samples each of the analog input voltages sequentially sent from the input circuit, and sends them to the A/D voltage comparator. This circuit also holds the sampled analog input voltage during A/D conversion.

(3) A/D voltage comparator

This A/D voltage comparator compares the voltage generated from the voltage tap of the comparison voltage generator with the analog input voltage. If the analog input voltage is found to be greater than the reference voltage (1/2 AV_{REF}) as a result of the comparison, the most significant bit (MSB) of the successive approximation register (SAR) is set. If the analog input voltage is less than the reference voltage (1/2 AV_{REF}), the MSB bit of the SAR is reset.

After that, bit 8 of the SAR register is automatically set, and the next comparison is made. The voltage tap of the comparison voltage generator is selected by the value of bit 9, to which the result has been already set.

Bit 9 = 0: (1/4 AV_{REF})
Bit 9 = 1: (3/4 AV_{REF})

The voltage tap of the comparison voltage generator and the analog input voltage are compared and bit 8 of the SAR register is manipulated according to the result of the comparison.

Analog input voltage ≥ Voltage tap of comparison voltage generator: Bit 8 = 1
Analog input voltage ≤ Voltage tap of comparison voltage generator: Bit 8 = 0

Comparison is continued like this to bit 0 of the SAR register.

When performing A/D conversion at a resolution of 8 bits, the comparison continues until bit 2 of the SAR register.

Remark AV_{REF}: The + side reference voltage of the A/D converter. This can be selected from AV_{REFP}, the internal reference voltage (1.45 V), and V_{DD}.

(4) Comparison voltage generator

The comparison voltage generator generates the comparison voltage input from an analog input pin.

Downloaded from Arrow.com.

(5) Successive approximation register (SAR)

The SAR register is a register that sets voltage tap data whose values from the comparison voltage generator match the voltage values of the analog input pins, 1 bit at a time starting from the most significant bit (MSB).

If data is set in the SAR register all the way to the least significant bit (LSB) (end of A/D conversion), the contents of the SAR register (conversion results) are held in the A/D conversion result register (ADCR). When all the specified A/D conversion operations have ended, an A/D conversion end interrupt request signal (INTAD) is generated.

(6) 10-bit A/D conversion result register (ADCR)

The A/D conversion result is loaded from the successive approximation register to this register each time A/D conversion is completed, and the ADCR register holds the A/D conversion result in its higher 10 bits (the lower 6 bits are fixed to 0).

(7) 8-bit A/D conversion result register (ADCRH)

The A/D conversion result is loaded from the successive approximation register to this register each time A/D conversion is completed, and the ADCRH register stores the higher 8 bits of the A/D conversion result.

(8) Controller

This circuit controls the conversion time of an input analog signal that is to be converted into a digital signal, as well as starting and stopping of the conversion operation. When A/D conversion has been completed, this controller generates INTAD through the A/D conversion result upper limit/lower limit comparator.

(9) AV_{REFP} pin

This pin inputs an external reference voltage (AV_{REFP}).

If using AV_{REFP} as the + side reference voltage of the A/D converter, set the ADREFP1 and ADREFP0 bits of the A/D converter mode register 2 (ADM2) to 0 and 1, respectively.

The analog signals input to ANI3 to ANI10 and ANI16 to ANI22 are converted to digital signals based on the voltage applied between AV_{REFP} and the – side reference voltage (AV_{REFM}/V_{SS}).

In addition to AV_{REFP}, it is possible to select V_{DD} or the internal reference voltage (1.45 V) as the + side reference voltage of the A/D converter.

(10) AV_{REFM} pin

This pin inputs an external reference voltage (AV_{REFM}). If using AV_{REFM} as the – side reference voltage of the A/D converter, set the ADREFM bit of the ADM2 register to 1.

In addition to AV_{REFM}, it is possible to select V_{SS} as the – side reference voltage of the A/D converter.

Downloaded from Arrow.com.

<R> 10.3 Registers Controlling A/D Converter

The A/D converter is controlled by the following registers.

- Peripheral enable register 0 (PER0)
- A/D converter mode register 0 (ADM0)
- A/D converter mode register 1 (ADM1)
- A/D converter mode register 2 (ADM2)
- 10-bit A/D conversion result register (ADCR)
- 8-bit A/D conversion result register (ADCRH)
- Analog input channel specification register (ADS)
- Conversion result comparison upper limit setting register (ADUL)
- Conversion result comparison lower limit setting register (ADLL)
- A/D test register (ADTES)
- A/D port configuration register (ADPC)
- Port mode control registers 0, 1, 4, 12, 14 (PMC0, PMC1, PMC4, PMC12, PMC14)
- Port mode registers 0, 1, 2, 4, 12, 14 (PM0, PM1, PM2, PM4, PM12, PM14)

Downloaded from Arrow.com.

10.3.1 Peripheral enable register 0 (PER0)

This register is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

When the A/D converter is used, be sure to set bit 5 (ADCEN) of this register to 1.

The PER0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 10-2. Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H After reset: 00H R/W

Symbol	<7>	6	<5>	<4>	<3>	<2>	1	<0>
PER0	TMKAEN	0	ADCEN	IICA0EN	SAU1EN[Note]	SAU0EN	0	TAU0EN

ADCEN	Control of clock supply to the A/D converter
0	Stops clock supply. • SFR used by the A/D converter cannot be written. • The A/D converter is in the reset status.
1	Enables clock supply. • SFR used by the A/D converter can be read/written.

Note 30-pin products only.

<R> **Cautions 1. When setting the A/D converter, be sure to set the following registers first while the ADCEN bit is set to 1. If ADCEN = 0, the values of the A/D converter control registers are cleared to their initial values, and writing to them is ignored (except for port mode registers 0, 1, 2, 4, 12, and 14 (PM0, PM1, PM2, PM4, PM12, and PM14), port mode control registers 0, 1, 4, 12, and 14 (PMC0, PMC1, PMC4, PMC12, and PMC14), and A/D port configuration register (ADPC)).**

- **A/D converter mode register 0 (ADM0)**
- **A/D converter mode register 1 (ADM1)**
- **A/D converter mode register 2 (ADM2)**
- **10-bit A/D conversion result register (ADCR)**
- **8-bit A/D conversion result register (ADCRH)**
- **Analog input channel specification register (ADS)**
- **Conversion result comparison upper limit setting register (ADUL)**
- **Conversion result comparison lower limit setting register (ADLL)**
- **A/D test register (ADTES).**

2. Be sure to clear the following bits to 0.

20- or 24-pin products: Bits 1, 3, and 6

30-pin products: Bits 1 and 6

Downloaded from Arrow.com.

10.3.2 A/D converter mode register 0 (ADM0)

This register sets the conversion time for analog input to be A/D converted, and starts/stops conversion.

The ADM0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 10-3. Format of A/D Converter Mode Register 0 (ADM0)

Address: FFF30H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	<0>
ADM0	ADCS	ADMD	FR2 Note 1	FR1 Note 1	FR0 Note 1	LV1 Note 1	LV0 Note 1	ADCE

<R>

ADCS	A/D conversion operation control
0	Stops conversion operation [When read] Conversion stopped/standby status
1	Enables conversion operation [When read] While in the software trigger mode: Conversion operation status While in the hardware trigger wait mode: A/D power supply stabilization wait status + conversion operation status

ADMD	Specification of the A/D conversion channel selection mode
0	Select mode
1	Scan mode

ADCE	A/D voltage comparator operation control Note 2
0	Stops A/D voltage comparator operation
1	Enables A/D voltage comparator operation

Notes 1. For details of the FR2 to FR0, LV1, LV0 bits, and A/D conversion, see **Table 10-3 A/D Conversion Time Selection**.

2. While in the software trigger mode or hardware trigger no-wait mode, the operation of the A/D voltage comparator is controlled by the ADCS and ADCE bits, and it takes 1 μs from the start of operation for the operation to stabilize. Therefore, when the ADCS bit is set to 1 after 1 μs or more has elapsed from the time ADCE bit is set to 1, the conversion result at that time has priority over the first conversion result. Otherwise, ignore data of the first conversion.

Cautions 1. Change the ADMD, FR2 to FR0, LV1 and LV0 bits while conversion is stopped (ADCS = 0, ADCE = 0).

2. The setting combination of the ADCS bit to 1 and the ADCE bit to 0 is prohibited.

3. Do not change the ADCS and ADCE bits from 0 to 1 at the same time by using an 8-bit manipulation instruction. Be sure to set these bits in the order described in 10.7 A/D Converter Setup Flowchart.

Downloaded from Arrow.com.

Table 10-1. Settings of ADCS and ADCE Bits

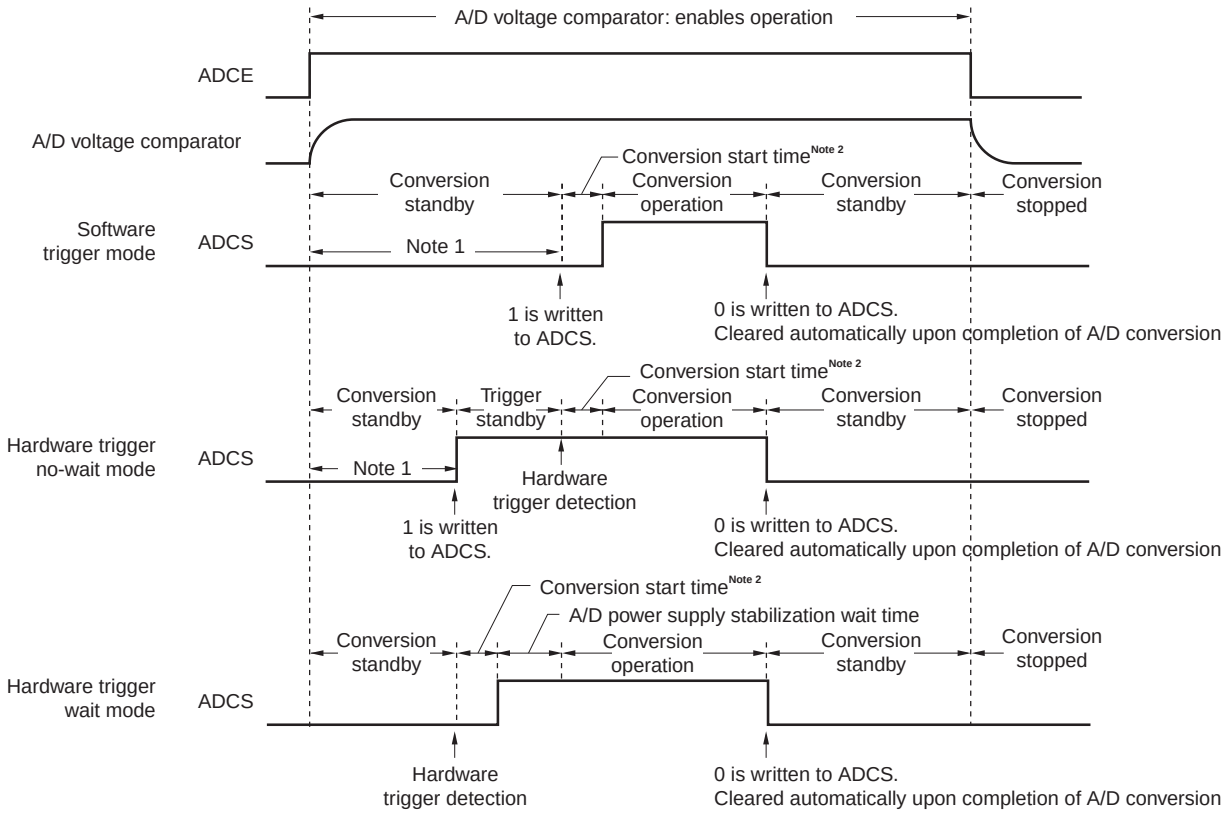
ADCS	ADCE	A/D Conversion Operation
0	0	Conversion stopped state
0	1	Conversion standby state
1	0	Setting prohibited
1	1	Conversion-in-progress state

Table 10-2. Setting and Clearing Conditions for ADCS Bit

A/D Conversion Mode			Set Conditions	Clear Conditions
Software trigger	Select mode	Sequential conversion mode	When 1 is written to ADCS	When 0 is written to ADCS
		One-shot conversion mode		• When 0 is written to ADCS • The bit is automatically cleared to 0 when A/D conversion ends.
	Scan mode	Sequential conversion mode		When 0 is written to ADCS
		One-shot conversion mode		• When 0 is written to ADCS • The bit is automatically cleared to 0 when conversion ends on the specified four channels.
Hardware trigger no-wait mode	Select mode	Sequential conversion mode		When 0 is written to ADCS
		One-shot conversion mode		When 0 is written to ADCS
	Scan mode	Sequential conversion mode		When 0 is written to ADCS
		One-shot conversion mode		When 0 is written to ADCS
Hardware trigger wait mode	Select mode	Sequential conversion mode	When a hardware trigger is input	When 0 is written to ADCS
		One-shot conversion mode		• When 0 is written to ADCS • The bit is automatically cleared to 0 when A/D conversion ends.
	Scan mode	Sequential conversion mode		When 0 is written to ADCS
		One-shot conversion mode		• When 0 is written to ADCS • The bit is automatically cleared to 0 when conversion ends on the specified four channels.

Downloaded from Arrow.com.

Figure 10-4. Timing Chart When A/D Voltage Comparator Is Used



Notes 1. While in the software trigger mode or hardware trigger no-wait mode, the time from the rising of the ADCE bit to the falling of the ADCS bit must be 1 μs or longer to stabilize the internal circuit.

2. In starting conversion, the longer will take up to following time

ADM0			Conversion Clock (f_{AD})	Conversion Start Time (Number of f_{CLK} Clock)	
FR2	FR1	FR0		Software Trigger Mode/ Hardware Trigger No-wait Mode	Hardware Trigger Wait Mode
0	0	0	f_{CLK}/64	63	1
0	0	1	f_{CLK}/32	31	
0	1	0	f_{CLK}/16	15	
0	1	1	f_{CLK}/8	7	
1	0	0	f_{CLK}/6	5	
1	0	1	f_{CLK}/5	4	
1	1	0	f_{CLK}/4	3	
1	1	1	f_{CLK}/2	1	

However, for the second and subsequent conversion in sequential conversion mode and for conversion of the channel specified by scan 1, 2, and 3 in scan mode, the conversion start time and stabilization wait time for A/D power supply do not occur after a hardware trigger is detected.

Downloaded from Arrow.com.

Cautions 1. If using the hardware trigger wait mode, setting the ADCS bit to 1 is prohibited (but the bit is automatically switched to 1 when the hardware trigger signal is detected). However, it is possible to clear the ADCS bit to 0 to specify the A/D conversion standby status.

2. While in the one-shot conversion mode of the hardware trigger no-wait mode, the ADCS flag is not automatically cleared to 0 when A/D conversion ends. Instead, 1 is retained.

3. Only rewrite the value of the ADCE bit when ADCS = 0 (while in the conversion stopped/conversion standby status).

4. To complete A/D conversion, specify at least the following time as the hardware trigger interval:
Hardware trigger no wait mode: 2 f_{CLK} clock + conversion start time + A/D conversion time
Hardware trigger wait mode: 2 f_{CLK} clock + conversion start time + A/D power supply stabilization wait time + A/D conversion time

Remark f_{CLK}: CPU/peripheral hardware clock frequency

Downloaded from Arrow.com.

Table 10-3. A/D Conversion Time Selection (1/4)

<R> **(1) When there is no A/D power supply stabilization wait time**
Normal mode 1 or 2 (software trigger mode/hardware trigger no-wait mode)

A/D Converter Mode Register 0 (ADM0)					Mode	Conversion Clock (f_{AD})	Number of Conversion Clock Note	Conversion Time	Conversion Time at 10-Bit Resolution					
									2.7 V ≤ V_{DD} ≤ 5.5 V					
FR2	FR1	FR0	LV1	LV0					f_{CLK} = 1 MHz	f_{CLK} = 2 MHz	f_{CLK} = 4 MHz	f_{CLK} = 8 MHz	f_{CLK} = 16 MHz	f_{CLK} = 24 MHz
0	0	0	0	0	Normal 1	f_{CLK}/64	19 f_{AD} (number of sampling clock: 7 f_{AD})	1216/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	76 μs	50.67 μs
0	0	1				f_{CLK}/32		608/f_{CLK}				76 μs	38 μs	25.33 μs
0	1	0				f_{CLK}/16		304/f_{CLK}			76 μs	38 μs	19 μs	12.67 μs
0	1	1				f_{CLK}/8		152/f_{CLK}		76 μs	38 μs	19 μs	9.5 μs	6.33 μs
1	0	0				f_{CLK}/6		114/f_{CLK}		57 μs	28.5 μs	14.25 μs	7.125 μs	4.75 μs
1	0	1				f_{CLK}/5		95/f_{CLK}	95 μs	47.5 μs	23.75 μs	11.875 μs	5.938 μs	3.96 μs
1	1	0				f_{CLK}/4		76/f_{CLK}	76 μs	38 μs	19 μs	9.5 μs	4.75 μs	3.17 μs
1	1	1				f_{CLK}/2		38/f_{CLK}	38 μs	19 μs	9.5 μs	4.75 μs	2.375 μs	Setting prohibited
0	0	0	0	1	Normal 2	f_{CLK}/64	17 f_{AD} (number of sampling clock: 5 f_{AD})	1088/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	68 μs	45.33 μs
0	0	1				f_{CLK}/32		544/f_{CLK}				68 μs	34 μs	22.67 μs
0	1	0				f_{CLK}/16		272/f_{CLK}			68 μs	34 μs	17 μs	11.33 μs
0	1	1				f_{CLK}/8		136/f_{CLK}		68 μs	34 μs	17 μs	8.5 μs	5.67 μs
1	0	0				f_{CLK}/6		102/f_{CLK}		51 μs	25.5 μs	12.75 μs	6.375 μs	4.25 μs
1	0	1				f_{CLK}/5		85/f_{CLK}	85 μs	42.5 μs	21.25 μs	10.625 μs	5.3125 μs	3.54 μs
1	1	0				f_{CLK}/4		68/f_{CLK}	68 μs	34 μs	17 μs	8.5 μs	4.25 μs	2.83 μs
1	1	1				f_{CLK}/2		34/f_{CLK}	34 μs	17 μs	8.5 μs	4.25 μs	2.125 μs	Setting prohibited
Other than the above						–	–	–	Setting prohibited					

<R> **Note** These are the numbers of clock cycles when conversion is with 10-bit resolution. When eight-bit resolution is selected, the values are shorter by two cycles of the conversion clock (f_{AD}).

<R> **Cautions**
1. **The A/D conversion time must be within the range of conversion times (t_{CONV}) described in 28.6.1 A/D converter characteristics or 29.6.1 A/D converter characteristics.**
2. **When rewriting the FR2 to FR0, LV1, and LV0 bits to other than the current data, stop A/D conversion once (ADCS = 0, ADCE = 0) beforehand.**
3. **The above conversion time does not include conversion state time. Conversion state time add in the first conversion. Select conversion time, taking clock frequency errors into consideration.**

Remark f_{CLK}: CPU/peripheral hardware clock frequency

Downloaded from Arrow.com.

Table 10-3. A/D Conversion Time Selection (2/4)

(2) When there is no A/D power supply stabilization wait time

<R> **Low-voltage mode 1 or 2 (software trigger mode/hardware trigger no-wait mode)**

A/D Converter Mode Register 0 (ADM0)					Mode	Conversion Clock (f_{AD})	Number of Conversion Clock Note 3	Conversion Time	Conversion Time at 10-Bit Resolution					
									1.8 V ≤ V_{DD} ≤ 5.5 V				**Note 1**	**Note 2**
FR2	FR1	FR0	LV1	LV0					f_{CLK} = 1 MHz	f_{CLK} = 2 MHz	f_{CLK} = 4 MHz	f_{CLK} = 8 MHz	f_{CLK} = 16 MHz	f_{CLK} = 24 MHz
0	0	0	1	0	Low voltage 1	f_{CLK}/64	19 f_{AD} (number of sampling clock : 7 f_{AD})	1216/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	76 μs	50.67 μs
0	0	1				f_{CLK}/32		608/f_{CLK}				76 μs	38 μs	25.33 μs
0	1	0				f_{CLK}/16		304/f_{CLK}			76 μs	38 μs	19 μs	12.67 μs
0	1	1				f_{CLK}/8		152/f_{CLK}		76 μs	38 μs	19 μs	9.5 μs	6.33 μs
1	0	0				f_{CLK}/6		114/f_{CLK}		57 μs	28.5 μs	14.25 μs	7.125 μs	4.75 μs
1	0	1				f_{CLK}/5		95/f_{CLK}	95 μs	47.5 μs	23.75 μs	11.875 μs	5.938 μs	3.96 μs
1	1	0				f_{CLK}/4		76/f_{CLK}	76 μs	38 μs	19 μs	9.5 μs	4.75 μs	3.17 μs
1	1	1				f_{CLK}/2		38/f_{CLK}	38 μs	19 μs	9.5 μs	4.75 μs	2.375 μs	Setting prohibited
0	0	0	1	1	Low voltage 2	f_{CLK}/64	17 f_{AD} (number of sampling clock : 5 f_{AD})	1088/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	68 μs	45.33 μs
0	0	1				f_{CLK}/32		544/f_{CLK}				68 μs	34 μs	22.667 μs
0	1	0				f_{CLK}/16		272/f_{CLK}			68 μs	34 μs	17 μs	11.333 μs
0	1	1				f_{CLK}/8		136/f_{CLK}		68 μs	34 μs	17 μs	8.5 μs	5.667 μs
1	0	0				f_{CLK}/6		102/f_{CLK}		51 μs	25.5 μs	12.75 μs	6.375 μs	4.25 μs
1	0	1				f_{CLK}/5		85/f_{CLK}	85 μs	42.5 μs	21.25 μs	10.625 μs	5.313 μs	3.542 μs
1	1	0				f_{CLK}/4		68/f_{CLK}	68 μs	34 μs	17 μs	8.5 μs	4.25 μs	2.833 μs
1	1	1				f_{CLK}/2		34/f_{CLK}	34 μs	17 μs	8.5 μs	4.25 μs	2.125 μs	Setting prohibited
Other than the above						–	–	–	Setting prohibited					

Notes 1. 2.4 V ≤ V_{DD} ≤ 5.5 V

2. 2.7 V ≤ V_{DD} ≤ 5.5 V

<R> **3.** These are the numbers of clock cycles when conversion is with 10-bit resolution. When eight-bit resolution is selected, the values are shorter by two cycles of the conversion clock (f_{AD}).

<R> **Cautions 1. The A/D conversion time must be within the range of conversion times (t_{CONV}) described in 28.6.1 A/D converter characteristics or 29.6.1 A/D converter characteristics.**

2. When rewriting the FR2 to FR0, LV1, and LV0 bits to other than the current data, stop A/D conversion once (ADCS = 0, ADCE = 0) beforehand.

3. The above conversion time does not include conversion state time. Conversion state time add in the first conversion. Select conversion time, taking clock frequency errors into consideration.

Remark f_{CLK}: CPU/peripheral hardware clock frequency

Downloaded from Arrow.com.

Table 10-3. A/D Conversion Time Selection (3/4)

(3) When there is A/D power supply stabilization wait time

<R> **Normal mode 1 or 2 (hardware trigger wait mode [Note1])**

A/D Converter Mode Register 0 (ADM0)					Mode	Conversion Clock (f_{AD})	Number of A/D Power Supply Stabilization Wait Clock	Number of Conversion Clock [Note 2]	A/D Power Supply Stabilization Wait Time + Conversion Time	A/D Power Supply Stabilization Wait Time + Conversion Time					
										2.7 V ≤ V_{DD} ≤ 5.5 V					
FR2	FR1	FR0	LV1	LV0						f_{CLK} = 1 MHz	f_{CLK} = 2 MHz	f_{CLK} = 4 MHz	f_{CLK} = 8 MHz	f_{CLK} = 16 MHz	f_{CLK} = 24 MHz
0	0	0	0	0	Normal 1	f_{CLK}/64	8 f_{AD}	19 f_{AD} (number of sampling clock : 7 f_{AD})	1728/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	108 μs	72 μs
0	0	1				f_{CLK}/32			864/f_{CLK}				108 μs	54 μs	36 μs
0	1	0				f_{CLK}/16			432/f_{CLK}			108 μs	54 μs	27 μs	18 μs
0	1	1				f_{CLK}/8			216/f_{CLK}		108 μs	54 μs	27 μs	13.5 μs	9 μs
1	0	0				f_{CLK}/6			162/f_{CLK}		81 μs	40.5 μs	20.25 μs	10.125 μs	6.75 μs
1	0	1				f_{CLK}/5			135/f_{CLK}	135 μs	67.5 μs	33.75 μs	16.875 μs	8.438 μs	5.625 μs
1	1	0				f_{CLK}/4			108/f_{CLK}	108 μs	54 μs	27 μs	13.5 μs	6.75 μs	4.5 μs
1	1	1				f_{CLK}/2			54/f_{CLK}	54 μs	27 μs	13.5 μs	6.75 μs	3.375 μs	Setting prohibited
0	0	0	0	1	Normal 2	f_{CLK}/64		17 f_{AD} (number of sampling clock : 5 f_{AD})	1600/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	100 μs	66.67 μs
0	0	1				f_{CLK}/32			800/f_{CLK}				100 μs	50 μs	33.33 μs
0	1	0				f_{CLK}/16			400/f_{CLK}			100 μs	50 μs	25 μs	16.67 μs
0	1	1				f_{CLK}/8			200/f_{CLK}		100 μs	50 μs	25 μs	12.5 μs	8.33 μs
1	0	0				f_{CLK}/6			150/f_{CLK}		75 μs	37.5 μs	18.75 μs	9.375 μs	6.25 μs
1	0	1				f_{CLK}/5			125/f_{CLK}	125 μs	62.5 μs	31.25 μs	15.625 μs	7.8125 μs	5.21 μs
1	1	0				f_{CLK}/4			100/f_{CLK}	100 μs	50 μs	25 μs	12.5 μs	6.25 μs	4.17 μs
1	1	1				f_{CLK}/2			50/f_{CLK}	50 μs	25 μs	12.5 μs	6.25 μs	3.125 μs	Setting prohibited
Other than the above						–	–	–	–	Setting prohibited					

<R> **Notes 1.** For the second and subsequent conversion in sequential conversion mode and for conversion of the channel specified by scan 1, 2, and 3 in scan mode, the conversion start time and stabilization wait time for A/D power supply do not occur after a hardware trigger is detected (see **Table 10-3 (1/4)**).

2. These are the numbers of clock cycles when conversion is with 10-bit resolution. When eight-bit resolution is selected, the values are shorter by two cycles of the conversion clock (f_{AD}).

<R> **Cautions 1. The A/D conversion time must be within the range of conversion times (t_{CONV}) indicated in 28.6.1 A/D converter characteristics or 29.6.1 A/D converter characteristics. Note that the conversion time (t_{CONV}) does not include the A/D power supply stabilization wait time.**

2. When rewriting the FR2 to FR0, LV1, and LV0 bits to other than the current data, stop A/D conversion once (ADCS = 0, ADCE = 0) beforehand.

3. The above conversion time does not include conversion state time. Conversion state time add in the first conversion. Select conversion time, taking clock frequency errors into consideration.

4. While in the hardware trigger wait mode, the conversion time includes the A/D power supply stabilization wait time after the hardware trigger is detected.

Remark f_{CLK}: CPU/peripheral hardware clock frequency

Downloaded from Arrow.com.

Table 10-3. A/D Conversion Time Selection (4/4)

(4) When there is A/D power supply stabilization wait time
Low-voltage mode 1 or 2 (hardware trigger wait mode Note 1)

A/D Converter Mode Register 0 (ADM0)					Mode	Conversion Clock (f_{AD})	Number of A/D Power Supply Stabilization Wait Clock	Number of Conversion Clock Note 4	A/D Power Supply Stabilization Wait Time + Conversion Time	A/D Power Supply Stabilization Wait Time + Conversion Time					
										1.8 V ≤ V_{DD} ≤ 5.5 V				**Note 2**	**Note 3**
FR2	FR1	FR0	LV1	LV0						f_{CLK} = 1 MHz	f_{CLK} = 2 MHz	f_{CLK} = 4 MHz	f_{CLK} = 8 MHz	f_{CLK} = 16 MHz	f_{CLK} = 24 MHz
0	0	0	1	0	Low voltage 1	f_{CLK}/64	2 f_{AD}	19 f_{AD} (number of sampling clock :7 f_{AD})	1344/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	84 μs	56 μs
0	0	1				f_{CLK}/32			672/f_{CLK}				84 μs	42 μs	28 μs
0	1	0				f_{CLK}/16			336/f_{CLK}			84 μs	42 μs	21 μs	14 μs
0	1	1				f_{CLK}/8			168/f_{CLK}		84 μs	42 μs	21 μs	10.5 μs	7 μs
1	0	0				f_{CLK}/6			126/f_{CLK}		63 μs	31.25 μs	15.75 μs	7.875 μs	5.25 μs
1	0	1				f_{CLK}/5			105/f_{CLK}	105 μs	52.5 μs	26.25 μs	13.125 μs	6.563 μs	4.38 μs
1	1	0				f_{CLK}/4			84/f_{CLK}	84 μs	42 μs	21 μs	10.5 μs	5.25 μs	3.5 μs
1	1	1				f_{CLK}/2			42/f_{CLK}	42 μs	21 μs	10.5 μs	5.25 μs	2.625 μs	Setting prohibited
0	0	0	1	1	Low voltage 2	f_{CLK}/64		17 f_{AD} (number of sampling clock :5 f_{AD})	1216/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	76 μs	50.67 μs
0	0	1				f_{CLK}/32			608/f_{CLK}				76 μs	38 μs	25.33 μs
0	1	0				f_{CLK}/16			304/f_{CLK}			76 μs	38 μs	19 μs	12.67 μs
0	1	1				f_{CLK}/8			152/f_{CLK}		76 μs	38 μs	19 μs	9.5 μs	6.33 μs
1	0	0				f_{CLK}/6			114/f_{CLK}		57 μs	28.5 μs	14.25 μs	7.125 μs	4.75 μs
1	0	1				f_{CLK}/5			95/f_{CLK}	95 μs	47.5 μs	23.75 μs	11.875 μs	5.938 μs	3.96 μs
1	1	0				f_{CLK}/4			76/f_{CLK}	76 μs	38 μs	19 μs	9.5 μs	4.75 μs	3.17 μs
1	1	1				f_{CLK}/2			38/f_{CLK}	38 μs	19 μs	9.5 μs	4.75 μs	2.375 μs	Setting prohibited
Other than the above						–	–	–	–	Setting prohibited					

<R> **Notes 1.** For the second and subsequent conversion in sequential conversion mode and for conversion of the channel specified by scan 1, 2, and 3 in scan mode, the conversion start time and stabilization wait time for A/D power supply do not occur after a hardware trigger is detected (see **Table 10-3 (2/4)**).

2. 2.4 V ≤ V_{DD} ≤ 5.5 V.

3. 2.7 V ≤ V_{DD} ≤ 5.5 V.

4. These are the numbers of clock cycles when conversion is with 10-bit resolution. When eight-bit resolution is selected, the values are shorter by two cycles of the conversion clock (f_{AD}).

<R> **Cautions 1. The A/D conversion time must be within the range of conversion times (t_{CONV}) described in 28.6.1 A/D converter characteristics or 29.6.1 A/D converter characteristics. Note that the conversion time (t_{CONV}) does not include the A/D power supply stabilization wait time.**

2. When rewriting the FR2 to FR0, LV1 and LV0 bits to other than the current data, stop A/D conversion once conversion (ADCS = 0, ADCE = 0) beforehand .

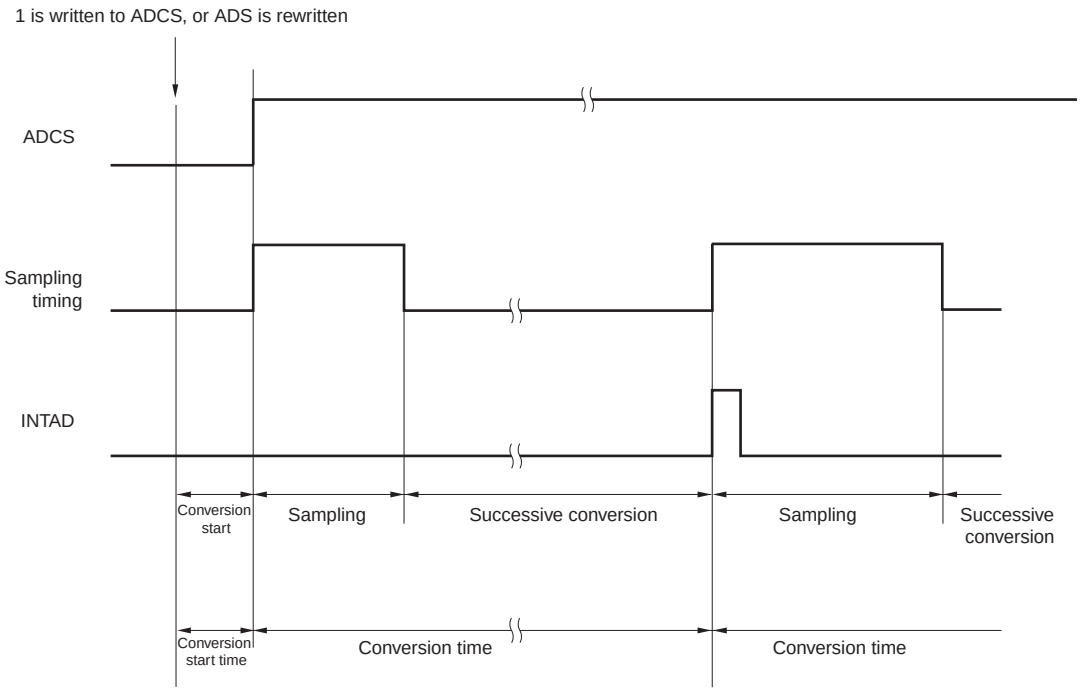
3. The above conversion time does not include conversion state time. Conversion state time add in the first conversion. Select conversion time, taking clock frequency errors into consideration.

4. While in the hardware trigger wait mode, the conversion time includes the A/D power supply stabilization wait time after the hardware trigger is detected.

Remark f_{CLK}: CPU/peripheral hardware clock frequency

Downloaded from Arrow.com.

Figure 10-5. A/D Converter Sampling and A/D Conversion Timing (Example for Software Trigger Mode)



Downloaded from Arrow.com.

10.3.3 A/D converter mode register 1 (ADM1)

This register is used to specify the A/D conversion trigger, conversion mode, and hardware trigger signal.

The ADM1 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 10-6. Format of A/D Converter Mode Register 1 (ADM1)

Address: FFF32H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADM1	ADTMD1	ADTMD0	ADCSM	0	0	0	ADTRS1	ADTRS0

ADTMD1	ADTMD0	Selection of the A/D conversion trigger mode
0	×	Software trigger mode
1	0	Hardware trigger no- wait mode
1	1	Hardware trigger wait mode

ADSCM	Specification of the A/D conversion mode
0	Sequential conversion mode
1	One-shot conversion mode

ADTRS1	ADTRS0	Selection of the hardware trigger signal
0	0	Count completion of timer channel 01 or capture completion interrupt signal (INTTM01)
1	1	12-bit interval timer interrupt signal (INTIT)
Other than above		Setting prohibited

Cautions 1. Only rewrite the value of the ADM1 register while conversion is stopped (ADCS = 0, ADCE = 0).

<R> **2. To complete A/D conversion, specify at least the following values as the hardware trigger interval:**

Hardware trigger no-wait mode: 2 f_{CLK} clock + conversion start time + A/D conversion time

Hardware trigger wait mode: 2 f_{CLK} clock + conversion start time + A/D power supply stabilization wait time + A/D conversion time

3. In modes other than SNOOZE mode, input of the next INTIT will not be recognized as a valid hardware trigger for up to four f_{CLK} cycles after the first INTIT is input.

Remarks 1. ×: don't care

2. f_{CLK}: CPU/peripheral hardware clock frequency

Downloaded from Arrow.com.

10.3.4 A/D converter mode register 2 (ADM2)

This register is used to select the + side or - side reference voltage of the A/D converter, check the upper limit and lower limit of A/D conversion result values, select the resolution, and specify whether to use the SNOOZE mode.

The ADM2 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

<R> **Figure 10-7. Format of A/D Converter Mode Register 2 (ADM2) (1/2)**

Address: F0010H After reset: 00H R/W

Symbol	7	6	5	4	<3>	<2>	1	<0>
ADM2	ADREFP1	ADREFP0	ADREFM	0	ADRCK	AWC	0	ADTYP

ADREFP1	ADREFP0	Selection of the + side reference voltage of the A/D converter
0	0	Supplied from V_{DD}
0	1	Supplied from P20/AV_{REFP}/ANI0
1	0	Supplied from the internal reference voltage (1.45 V) Note
1	1	Setting prohibited

- When ADREFP1 or ADREFP0 bit is rewritten, this must be configured in accordance with the following procedures.

(1) Set ADCE = 0

(2) Change the values of ADREFP1 and ADREFP0

(3) Reference voltage stabilization wait time (A)

(4) Set ADCE = 1

(5) Reference voltage stabilization wait time (B)

When ADREFP1 and ADREFP0 are set to 1 and 0, the setting is changed to A = 5 μs, B = 1 μs.

When ADREFP1 and ADREFP0 are set to 0 and 0 or 0 and 1, A needs no wait and B = 1 μs.

After wait of (5), start A/D conversion

- When ADREFP1 and ADREFP0 are set to 1 and 0, respectively, A/D conversion cannot be performed on the temperature sensor output voltage and internal reference voltage (1.45 V).
  Be sure to perform A/D conversion while ADISS = 0.

ADREFM	Selection of the – side reference voltage of the A/D converter
0	Supplied from V_{SS}
1	Supplied from P21/AV_{REFM}/ANI1

Note This setting can be used only in HS (high-speed main) mode.

Cautions 1. Only rewrite the value of the ADM2 register while conversion is stopped (ADCS = 0, ADCE = 0).

2. Do not set the ADREFP1 bit to 1 when shifting to STOP mode. If the internal reference voltage (ADREFP1, ADREFP0 = 1, 0) is selected, the A/D converter reference voltage current (I_{ADREF}) indicated in 28.3.2 or 29.3.2 Supply current characteristics will be added.

3. When using AV_{REFP} and AV_{REFM}, specify ANI0 and ANI1 as the analog input channels and specify input mode by using the port mode register.

Downloaded from Arrow.com.

Figure 10-7. Format of A/D Converter Mode Register 2 (ADM2) (2/2)

Address: F0010H After reset: 00H R/W

Symbol	7	6	5	4	<3>	<2>	1	<0>
ADM2	ADREFP1	ADREFP0	ADREFM	0	ADRCK	AWC	0	ADTYP

ADRCK	Checking the upper limit and lower limit conversion result values
0	The interrupt signal (INTAD) is output when the ADLL register ≤ the ADCR register ≤ the ADUL register (AREA1).
1	The interrupt signal (INTAD) is output when the ADCR register < the ADLL register (AREA2) or the ADUL register < the ADCR register (AREA3).
Figure 10-8 shows the generation range of the interrupt signal (INTAD) for AREA 1 to AREA 3.	

AWC	Specification of the SNOOZE mode
0	Do not use the SNOOZE mode function.
1	Use the SNOOZE mode function.

When there is a hardware trigger signal in the STOP mode, the STOP mode is exited, and A/D conversion is performed without operating the CPU (the SNOOZE mode).

- The SNOOZE mode function can be specified only when the high-speed on-chip oscillator clock is selected for the CPU/peripheral hardware clock (f_{CLK}). If any other clock is selected, specifying this mode is prohibited.
- Using the SNOOZE mode function in the software trigger mode or hardware trigger no-wait mode is prohibited.
- Using the SNOOZE mode function in the sequential conversion mode is prohibited.
- When using the SNOOZE mode function, specify a hardware trigger interval of at least “shift time to SNOOZE mode **Note** + conversion start time + A/D power supply stabilization wait time + A/D conversion time+ 2 f_{CLK} clock” (refer to **Table 10-3**).
- Even when using the SNOOZE mode, be sure to set the AWC bit to 0 in the normal operation and change it to 1 just before shifting to the STOP mode.

  Also, be sure to change the AWC bit to 0 after returning from the STOP mode to normal operation. If the AWC bit is left set to 1, A/D conversion will not start normally in spite of the subsequent SNOOZE or normal operation mode.

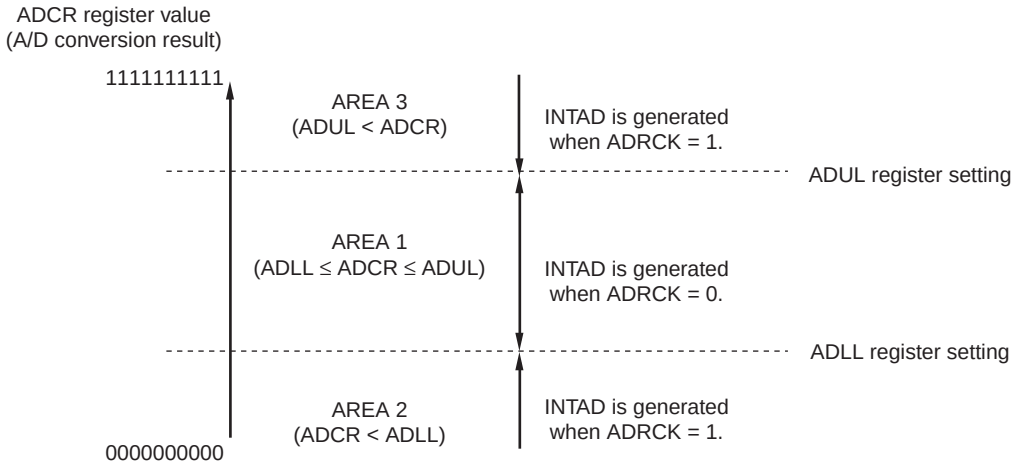
ADTYP	Selection of the A/D conversion resolution
0	10-bit resolution
1	8-bit resolution

Note Refer to "Transition time from STOP mode to SNOOZE mode" in **17.3.3 SNOOZE mode**.

Caution Only rewrite the value of the ADM2 register while conversion is stopped (ADCS = 0, ADCE = 0).

Downloaded from Arrow.com.

Figure 10-8. ADRCK Bit Interrupt Signal Generation Range



Remark If INTAD does not occur, the A/D conversion result is not stored in the ADCR or ADCRH register.

Downloaded from Arrow.com.

10.3.5 10-bit A/D conversion result register (ADCR)

This register is a 16-bit register that stores the A/D conversion result. The lower 6 bits are fixed to 0. Each time A/D conversion ends, the conversion result is loaded from the successive approximation register (SAR). The higher 8 bits of the conversion result are stored in FFF1FH and the lower 2 bits are stored in the higher 2 bits of FFF1EH[Note].

The ADCR register can be read by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Note If the A/D conversion result is outside the range specified by using the A/D conversion comparison function (the value specified by the ADRCK bit of the ADUL/ADLL registers; see Figure 10-8), the result is not stored.

Figure 10-9. Format of 10-bit A/D Conversion Result Register (ADCR)

Address: FFF1FH, FFF1EH After reset: 0000H R

Symbol	FFF1FH								FFF1EH							
ADCR											0	0	0	0	0	0

Cautions 1. When 8-bit resolution A/D conversion is selected (when the ADTYP bit of A/D converter mode register 2 (ADM2) is 1) and the ADCR register is read, 0 is read from the lower two bits (bits 7 and 6 of the ADCR register).

2. When the ADCR register is accessed in 16-bit units, the higher 10 bits of the conversion result are read in order starting at bit 15 of the ADCR register.

10.3.6 8-bit A/D conversion result register (ADCRH)

This register is an 8-bit register that stores the A/D conversion result. The higher 8 bits of 10-bit resolution are stored[Note].

The ADCRH register can be read by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Note If the A/D conversion result is outside the range specified by using the A/D conversion comparison function (the value specified by the ADRCK bit of the ADUL/ADLL registers; see Figure 10-8), the result is not stored.

Figure 10-10. Format of 8-bit A/D Conversion Result Register (ADCRH)

Address: FFF1FH After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ADCRH								

Caution When writing to the A/D converter mode register 0 (ADM0), analog input channel specification register (ADS), and A/D port configuration register (ADPC), the contents of the ADCRH register may become undefined. Read the conversion result following conversion completion before writing to the ADM0, ADS, and ADPC registers. Using timing other than the above may cause an incorrect conversion result to be read.

Downloaded from Arrow.com.

10.3.7 Analog input channel specification register (ADS)

This register specifies the input channel of the analog voltage to be A/D converted.

The ADS register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 10-11. Format of Analog Input Channel Specification Register (ADS)

Address: FFF31H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADS	ADISS	0	0	ADS4	ADS3	ADS2	ADS1	ADS0

○ Select mode (ADMD = 0) (20- or 24-pin products)

ADISS	ADS4	ADS3	ADS2	ADS1	ADS0	Analog input channel	Input source
0	0	0	0	0	0	ANI0	P20/ANI0/AV_{REFP} pin
0	0	0	0	0	1	ANI1	P21/ANI1/AV_{REFM} pin
0	0	0	0	1	0	ANI2	P22/ANI2 pin
0	0	0	0	1	1	ANI3	P23/ANI3 pin
0	1	0	0	0	0	ANI16	P10/ANI16 pin
0	1	0	0	0	1	ANI17	P11/ANI17 pin
0	1	0	0	1	0	ANI18	P12/ANI18 pin
0	1	0	0	1	1	ANI19	P13/ANI19 pin
0	1	0	1	0	0	ANI20	P14/ANI20 pin
0	1	0	1	0	1	ANI21	P42/ANI21 pin
0	1	0	1	1	0	ANI22	P41/ANI22 pin
1	0	0	0	0	0	–	Temperature sensor output Note
1	0	0	0	0	1	–	Internal reference voltage output (1.45 V) Note
Other than the above						Setting prohibited	

Note This setting can be used only in HS (high-speed main) mode.

○ Select mode (ADMD = 0) (30-pin products)

ADISS	ADS4	ADS3	ADS2	ADS1	ADS0	Analog input channel	Input source
0	0	0	0	0	0	ANI0	P20/ANI0/AV_{REFP} pin
0	0	0	0	0	1	ANI1	P21/ANI1/AV_{REFM} pin
0	0	0	0	1	0	ANI2	P22/ANI2 pin
0	0	0	0	1	1	ANI3	P23/ANI3 pin
0	1	0	0	0	0	ANI16	P01/ANI16 pin
0	1	0	0	0	1	ANI17	P00/ANI17 pin
0	1	0	0	1	0	ANI18	P147/ANI18 pin
0	1	0	0	1	1	ANI19	P120/ANI19 pin
1	0	0	0	0	0	–	Temperature sensor output Note
1	0	0	0	0	1	–	Internal reference voltage output (1.45 V) Note
Other than the above						Setting prohibited	

Note This setting can be used only in HS (high-speed main) mode.

Downloaded from Arrow.com.

○ Scan mode (ADMD = 1)

ADS4	ADS3	ADS2	ADS1	ADS0	Analog input channel			
					Scan 0	Scan 1	Scan 2	Scan 3
0	0	0	0	0	ANI0	ANI1	ANI2	ANI3
0	0	0	0	1	ANI1	ANI2	ANI3	–
0	0	0	1	0	ANI2	ANI3	–	–
0	0	0	1	1	ANI3	–	–	–
Other than the above					Setting prohibited			

Remark – : Ignore the conversion result because it is undefined.

Cautions
1. **Be sure to clear bits 5 and 6 to 0.**
2. **Set a channel to be set the analog input by ADPC and PMCx registers in the input mode by using port mode registers 0, 1, 2, 4, 12, or 14 (PM0, PM1, PM2, PM4 PM12, PM14).**
3. **Do not set the pin that is set by the A/D port configuration register (ADPC) as digital I/O by the ADS register.**
4. **Do not set the pin that is set by port mode control registers 0, 1, 4, 12, 14 (PMC0, PMC1, PMC4, PMC12, PMC14) as digital I/O by the ADS register.**
5. **Only rewrite the value of the ADISS bit while A/D conversion comparator operation is stopped(ADCS = 0, ADCE = 0)**
6. **If using AV_{REFP} as the + side reference voltage of the A/D converter, do not select ANI0 as an A/D conversion channel.**
7. **If using AV_{REFM} as the – side reference voltage of the A/D converter, do not select ANI1 as an A/D conversion channel.**
8. **If the ADISS bit is set to 1, the internal reference voltage (1.45 V) cannot be used for the + side reference voltage. After the ADISS bit is set to 1, the initial conversion result cannot be used. For details about the setting flow, see 10.7.4 Setup when temperature sensor output voltage/internal reference voltage is selected.**
9. **Do not set the ADISS bit to 1 when shifting to the STOP mode or HALT mode. If the ADISS bit is set to 1, the A/D converter reference voltage current (I_{ADREF}) indicated in 28.3.2 or 29.3.2 Supply current characteristics will be added.**

Downloaded from Arrow.com.

10.3.8 Conversion result comparison upper limit setting register (ADUL)

This register is used to specify the setting for checking the upper limit of the A/D conversion results.

The A/D conversion results and ADUL register value are compared, and interrupt signal (INTAD) generation is controlled in the range specified for the ADRCK bit of A/D converter mode register 2 (ADM2) (shown in **Figure 10-8**).

The ADUL register can be set by an 8-bit memory manipulation instruction.

Reset signal generation sets this register to FFH.

Figure 10-12. Format of Conversion Result Comparison Upper Limit Setting Register (ADUL)

Address: F0011H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
ADUL	ADUL7	ADUL6	ADUL5	ADUL4	ADUL3	ADUL2	ADUL1	ADUL0

10.3.9 Conversion result comparison lower limit setting register (ADLL)

This register is used to specify the setting for checking the lower limit of the A/D conversion results.

The A/D conversion results and ADLL register value are compared, and interrupt signal (INTAD) generation is controlled in the range specified for the ADRCK bit of A/D converter mode register 2 (ADM2) (shown in **Figure 10-8**).

The ADLL register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 10-13. Format of Conversion Result Comparison Lower Limit Setting Register (ADLL)

Address: F0012H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADLL	ADLL7	ADLL6	ADLL5	ADLL4	ADLL3	ADLL2	ADLL1	ADLL0

Cautions

1. When 10-bit resolution A/D conversion is selected, the higher 8 bits of the 10-bit A/D conversion result register (ADCR) are compared with the value in the ADUL and ADLL registers.

<R> **2. Only rewrite the values of the ADUL and ADLL registers while conversion is stopped (ADCS = 0, ADCE = 0).**

<R> **3. The value of the ADUL register must be greater than that of the ADLL register.**

Downloaded from Arrow.com.

10.3.10 A/D test register (ADTES)

This register is used to select the + side or - side reference voltage of the A/D converter, an analog input channel (ANIxx), the temperature sensor output voltage, and the internal reference voltage (1.45 V) as the target for A/D conversion. For details, see **21.3.7 A/D test function**.

When using this register to test the A/D converter, set as follows.

- For zero-scale measurement, select the - side reference voltage as the target for conversion.
- For full-scale measurement, select the + side reference voltage as the target for conversion.

The ADTES register is set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 10-14. Format of A/D Test Register (ADTES)

Address: F0013H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADTES	0	0	0	0	0	0	ADTES1	ADTES0

ADTES1	ADTES0	A/D conversion target
0	0	ANIxx/temperature sensor output voltage [Note]/internal reference voltage (1.45 V) [Note] (This is specified using the analog input channel specification register (ADS).)
1	0	The - side reference voltage (selected by the ADREFM bit of the ADM2 register)
1	1	The + side reference voltage (selected by the ADREFP1 and ADREFP0 bits of the ADM2 register)
Other than the above		Setting prohibited

Note The temperature sensor output voltage and internal reference voltage (1.45 V) can be selected only in HS (high-speed main) mode.

10.3.11 Registers controlling port function of analog input pins

Set up the port mode registers (PMxx), port mode control registers (PMCxx), and A/D port configuration register (ADPC), which are used for controlling the functions of the ports shared with the analog input pins of the A/D converter. For details, see **4.3.1 Port mode registers (PMxx)**, **4.3.6 Port mode control registers (PMCxx)**, and **4.3.7 A/D port configuration register**.

When using the ANI0 to ANI3 pins for analog input of the A/D converter, set the port mode register (PMxx) bit corresponding to each port to 1, and specify analog input by using the A/D port configuration register (ADPC).

When using the ANI16 to ANI22 pins for analog input of the A/D converter, set the port mode register (PMxx) bit and the port mode control register (PMCxx) bit to 1.

Downloaded from Arrow.com.

10.4 A/D Converter Conversion Operations

The A/D converter conversion operations are described below.

<1> The voltage input to the selected analog input channel is sampled by the sample & hold circuit.

<2> When sampling has been done for a certain time, the sample & hold circuit is placed in the hold state and the sampled voltage is held until the A/D conversion operation has ended.

<3> Bit 9 of the successive approximation register (SAR) is set. The series resistor string voltage tap is set to (1/2) AV_{REF} by the tap selector.

<4> The voltage difference between the series resistor string voltage tap and sampled voltage is compared by the voltage comparator. If the analog input is greater than (1/2) AV_{REF}, the MSB bit of the SAR register remains set to 1. If the analog input is smaller than (1/2) AV_{REF}, the MSB bit is reset to 0.

<5> Next, bit 8 of the SAR register is automatically set to 1, and the operation proceeds to the next comparison. The series resistor string voltage tap is selected according to the preset value of bit 9, as described below.

- Bit 9 = 1: (3/4) AV_{REF}
- Bit 9 = 0: (1/4) AV_{REF}

The voltage tap and sampled voltage are compared and bit 8 of the SAR register is manipulated as follows.

- Sampled voltage ≥ Voltage tap: Bit 8 = 1
- Sampled voltage < Voltage tap: Bit 8 = 0

<6> Comparison is continued in this way up to bit 0 of the SAR register.

<7> Upon completion of the comparison of 10 bits, an effective digital result value remains in the SAR register, and the result value is transferred to the A/D conversion result register (ADCR, ADCRH) and then latched **Note 1**.
At the same time, the A/D conversion end interrupt request (INTAD) can also be generated **Note 1**.

<8> Repeat steps <1> to <7>, until the ADCS bit is cleared to 0 **Note 2**.
To stop the A/D converter, clear the ADCS bit to 0.

Notes 1. If the A/D conversion result is outside the A/D conversion result range specified by the ADRCK bit and the ADUL and ADLL registers (see **Figure 10-8**), the A/D conversion result interrupt request signal is not generated and no A/D conversion results are stored in the ADCR and ADCRH registers.

2. While in the sequential conversion mode, the ADCS flag is not automatically cleared to 0. This flag is not automatically cleared to 0 while in the one-shot conversion mode of the hardware trigger no-wait mode, either. Instead, 1 is retained.

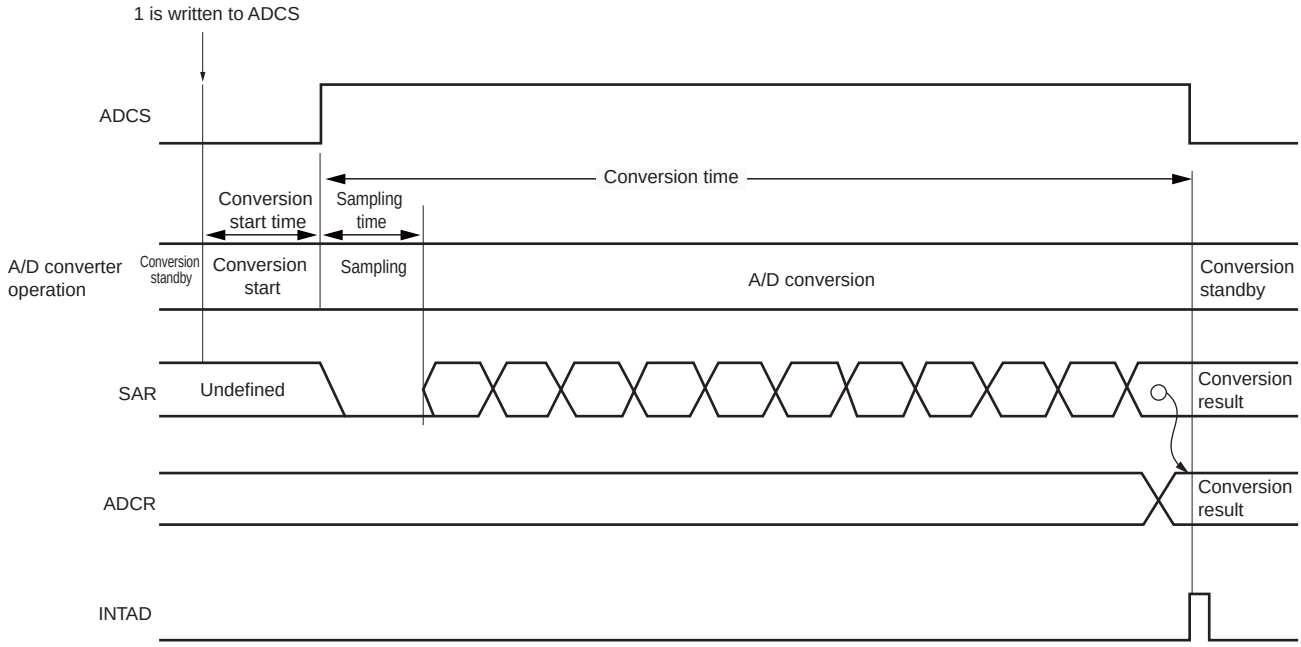
Remarks 1. Two types of the A/D conversion result registers are available.

- ADCR register (16 bits): Store 10-bit A/D conversion value
- ADCRH register (8 bits): Store 8-bit A/D conversion value

2. AV_{REF}: The + side reference voltage of the A/D converter. This can be selected from AV_{REFP}, the internal reference voltage (1.45 V), and V_{DD}.

Downloaded from Arrow.com.

Figure 10-15. Conversion Operation of A/D Converter (Software Trigger Mode)



<R> In one-shot conversion mode, the ADCS bit is automatically cleared to 0 after completion of A/D conversion.

In sequential conversion mode, A/D conversion operations proceed continuously until the software clears bit 7 (ADCS) of the A/D converter mode register 0 (ADM0) to 0.

If the analog input channel specification register (ADS) is overwritten during A/D conversion operation, the current conversion is interrupted, and A/D conversion of the analog input specified by the ADS register is performed. Data from the A/D conversion that was in progress are discarded.

Reset signal generation clears the A/D conversion result register (ADCR, ADCRH) to 0000H or 00H.

Downloaded from Arrow.com.

10.5 Input Voltage and Conversion Results

The relationship between the analog input voltage input to the analog input pins (ANI0 to ANI3, ANI16 to ANI22) and the theoretical A/D conversion result (stored in the 10-bit A/D conversion result register (ADCR)) is shown by the following expression.

$$SAR = INT\left(\frac{V_{AIN}}{AV_{REF}} \times 1024 + 0.5\right)$$

$$ADCR = SAR \times 64$$

or

$$\left(\frac{ADCR}{64} - 0.5\right) \times \frac{AV_{REF}}{1024} \leq V_{AIN} < \left(\frac{ADCR}{64} + 0.5\right) \times \frac{AV_{REF}}{1024}$$

where, INT( ): Function which returns integer part of value in parentheses

V_{AIN}: Analog input voltage

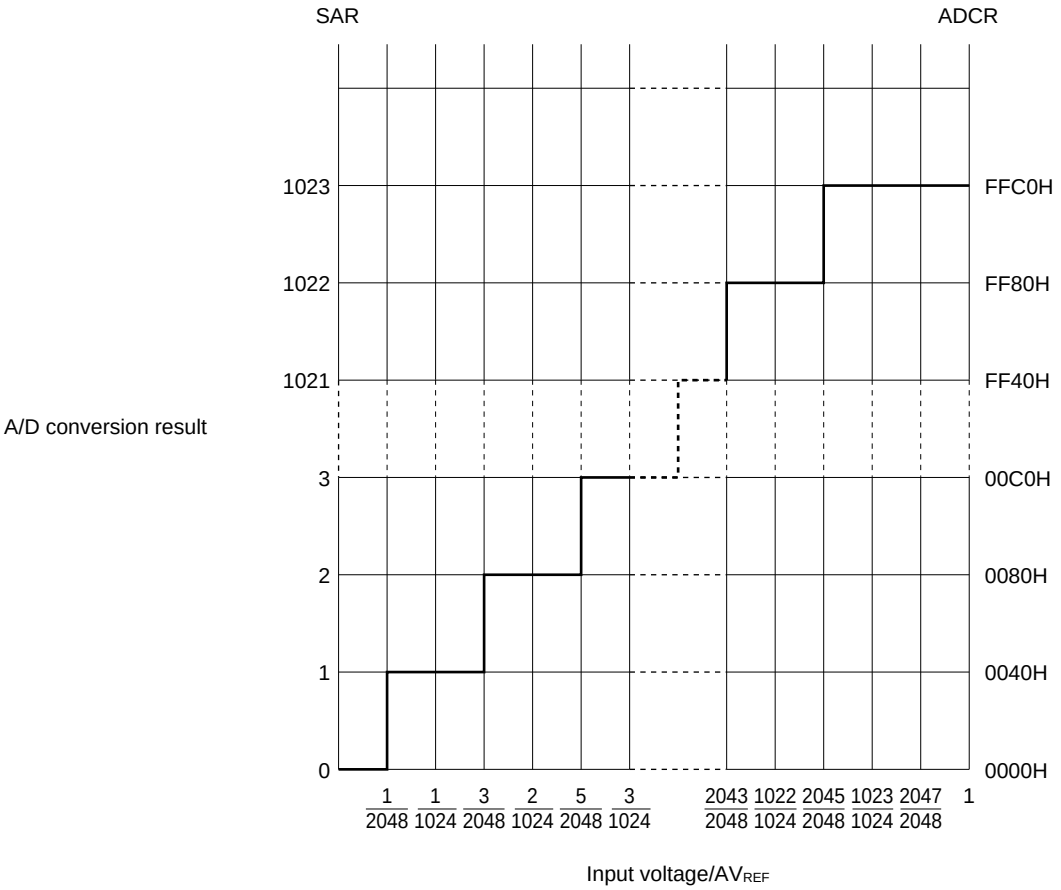
AV_{REF}: AV_{REF} pin voltage

ADCR: A/D conversion result register (ADCR) value

SAR: Successive approximation register

Figure 10-16 shows the relationship between the analog input voltage and the A/D conversion result.

Figure 10-16. Relationship Between Analog Input Voltage and A/D Conversion Result



Remark AV_{REF}: The + side reference voltage of the A/D converter. This can be selected from AV_{REFP}, the internal reference voltage (1.45 V), and V_{DD}.

Downloaded from Arrow.com.

10.6 A/D Converter Operation Modes

The operation of each A/D converter mode is described below. In addition, the procedure for specifying each mode is described in **10.7 A/D Converter Setup Flowchart**.

10.6.1 Software trigger mode (select mode, sequential conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1 μs), the ADCS bit of the ADM0 register is set to 1 to perform the A/D conversion of the analog input specified by the analog input channel specification register (ADS).

<3> When A/D conversion ends, the conversion result is stored in the A/D conversion result register (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) is generated. After A/D conversion ends, the next A/D conversion immediately starts.

<4> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

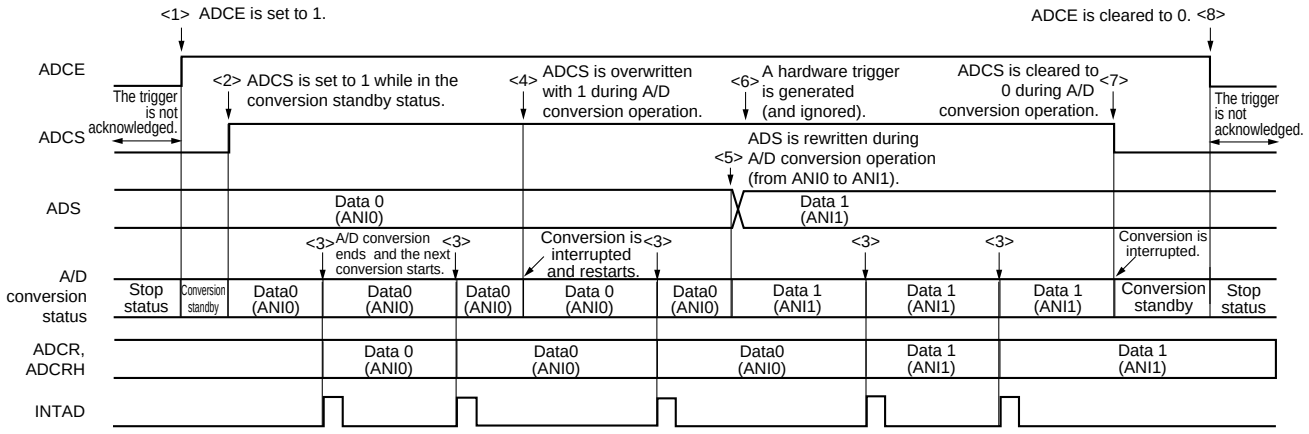
<5> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.

<6> Even if a hardware trigger is input during conversion operation, A/D conversion does not start.

<7> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status.

<8> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCE = 0, specifying 1 for ADCS is ignored and A/D conversion does not start.

Figure 10-17. Example of Software Trigger Mode (Select Mode, Sequential Conversion Mode) Operation Timing



Downloaded from Arrow.com.

10.6.2 Software trigger mode (select mode, one-shot conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1 μs), the ADCS bit of the ADM0 register is set to 1 to perform the A/D conversion of the analog input specified by the analog input channel specification register (ADS).

<3> When A/D conversion ends, the conversion result is stored in the A/D conversion result register (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) is generated.

<4> After A/D conversion ends, the ADCS bit is automatically cleared to 0, and the system enters the A/D conversion standby status.

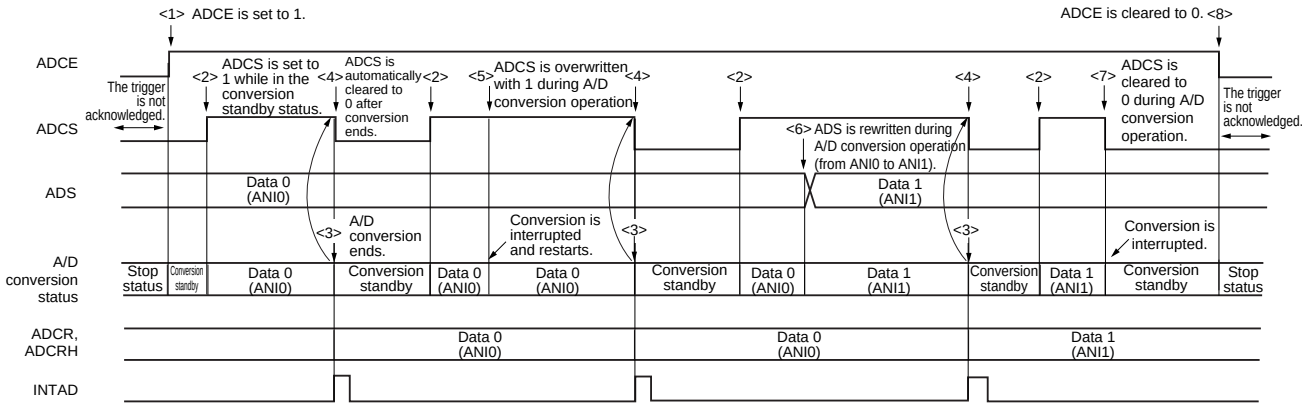
<5> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<6> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.

<7> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status.

<8> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCE = 0, specifying 1 for ADCS is ignored and A/D conversion does not start. In addition, A/D conversion does not start even if a hardware trigger is input while in the A/D conversion standby status.

Figure 10-18. Example of Software Trigger Mode (Select Mode, One-Shot Conversion Mode) Operation Timing



Downloaded from Arrow.com.

10.6.3 Software trigger mode (scan mode, sequential conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1 μs), the ADCS bit of the ADM0 register is set to 1 to perform A/D conversion on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS). A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.

<3> A/D conversion is sequentially performed on the four analog input channels, the conversion results are stored in the A/D conversion result register (ADCR, ADCRH) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD) is generated. After A/D conversion of the four channels ends, the A/D conversion of the channel following the specified channel automatically starts (until all four channels are finished).

<4> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

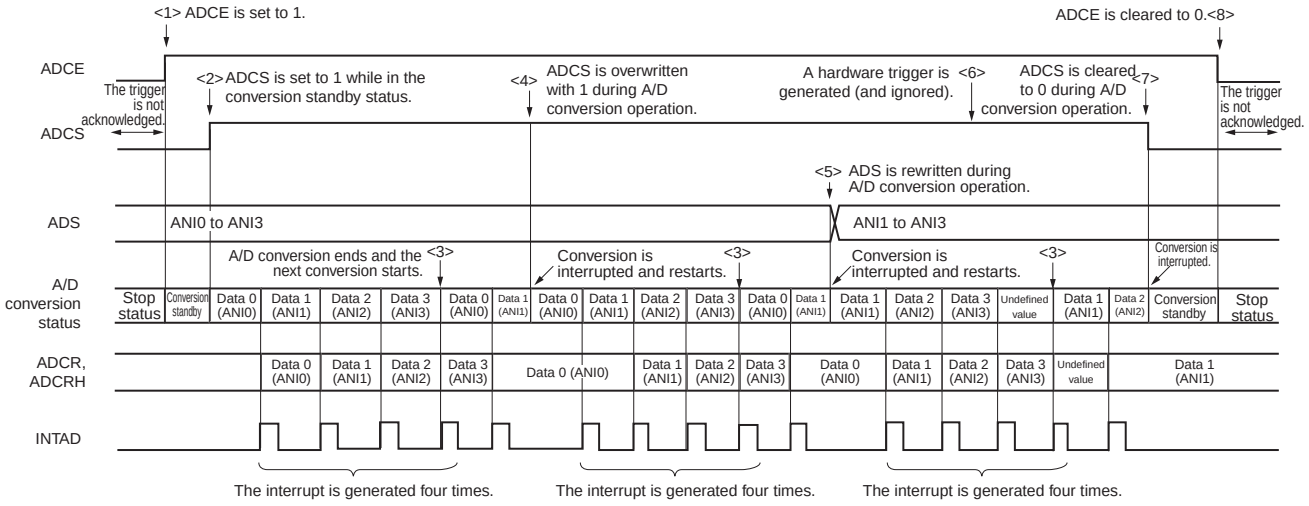
<5> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.

<6> Even if a hardware trigger is input during conversion operation, A/D conversion does not start.

<7> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status.

<8> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCE = 0, specifying 1 for ADCS is ignored and A/D conversion does not start.

Figure 10-19. Example of Software Trigger Mode (Scan Mode, Sequential Conversion Mode) Operation Timing



Downloaded from Arrow.com.

10.6.4 Software trigger mode (scan mode, one-shot conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1 μs), the ADCS bit of the ADM0 register is set to 1 to perform A/D conversion on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS). A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.

<3> A/D conversion is sequentially performed on the four analog input channels, the conversion results are stored in the A/D conversion result register (ADCR, ADCRH) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD) is generated.

<4> After A/D conversion of the four channels ends, the ADCS bit is automatically cleared to 0, and the system enters the A/D conversion standby status.

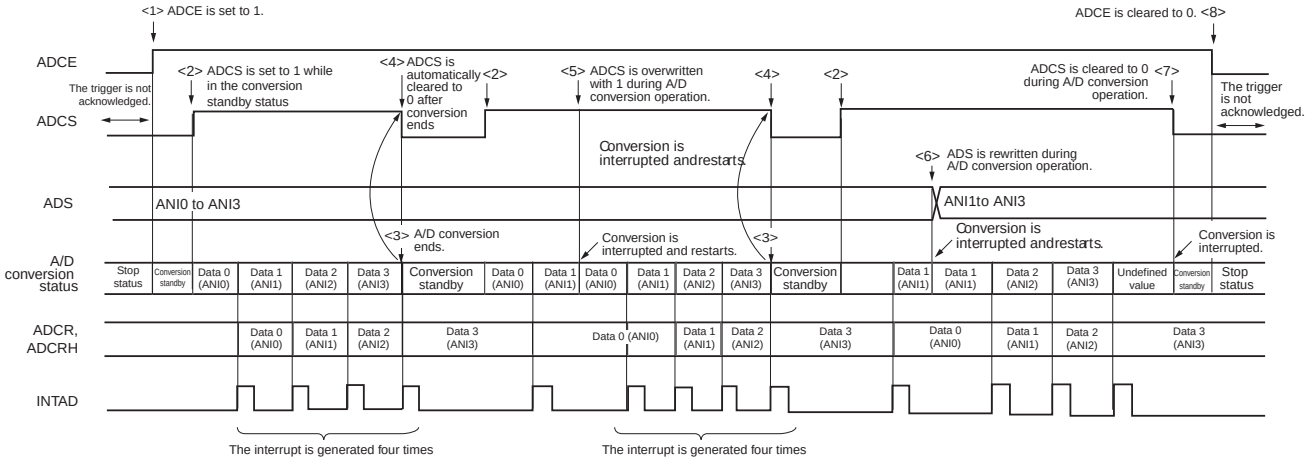
<5> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

<6> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.

<7> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status.

<8> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCE = 0, specifying 1 for ADCS is ignored and A/D conversion does not start. In addition, A/D conversion does not start even if a hardware trigger is input while in the A/D conversion standby status.

Figure 10-20. Example of Software Trigger Mode (Scan Mode, One-Shot Conversion Mode) Operation Timing



Downloaded from Arrow.com.

10.6.5 Hardware trigger no-wait mode (select mode, sequential conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1 μs), the ADCS bit of the ADM0 register is set to 1 to place the system in the hardware trigger standby status (and conversion does not start at this stage). Note that, while in this status, A/D conversion does not start even if ADCS is set to 1.

<3> If a hardware trigger is input while ADCS = 1, A/D conversion is performed on the analog input specified by the analog input channel specification register (ADS).

<4> When A/D conversion ends, the conversion result is stored in the A/D conversion result register (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) is generated. After A/D conversion ends, the next A/D conversion immediately starts.

<5> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

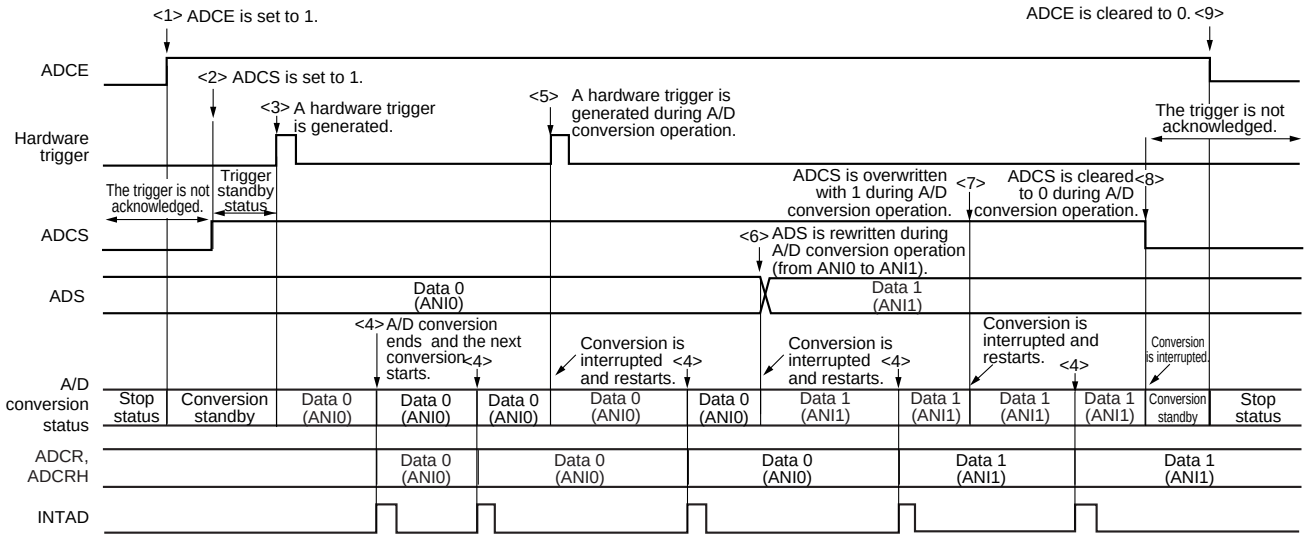
<6> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.

<7> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<8> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status. However, the A/D converter does not power down in this status.

<9> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCS = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 10-21. Example of Hardware Trigger No-Wait Mode (Select Mode, Sequential Conversion Mode) Operation Timing



Downloaded from Arrow.com.

10.6.6 Hardware trigger no-wait mode (select mode, one-shot conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1 μs), the ADCS bit of the ADM0 register is set to 1 to place the system in the hardware trigger standby status (and conversion does not start at this stage). Note that, while in this status, A/D conversion does not start even if ADCS is set to 1.

<3> If a hardware trigger is input while ADCS = 1, A/D conversion is performed on the analog input specified by the analog input channel specification register (ADS).

<4> When A/D conversion ends, the conversion result is stored in the A/D conversion result register (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) is generated.

<5> After A/D conversion ends, the ADCS bit remains set to 1, and the system enters the A/D conversion standby status.

<6> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

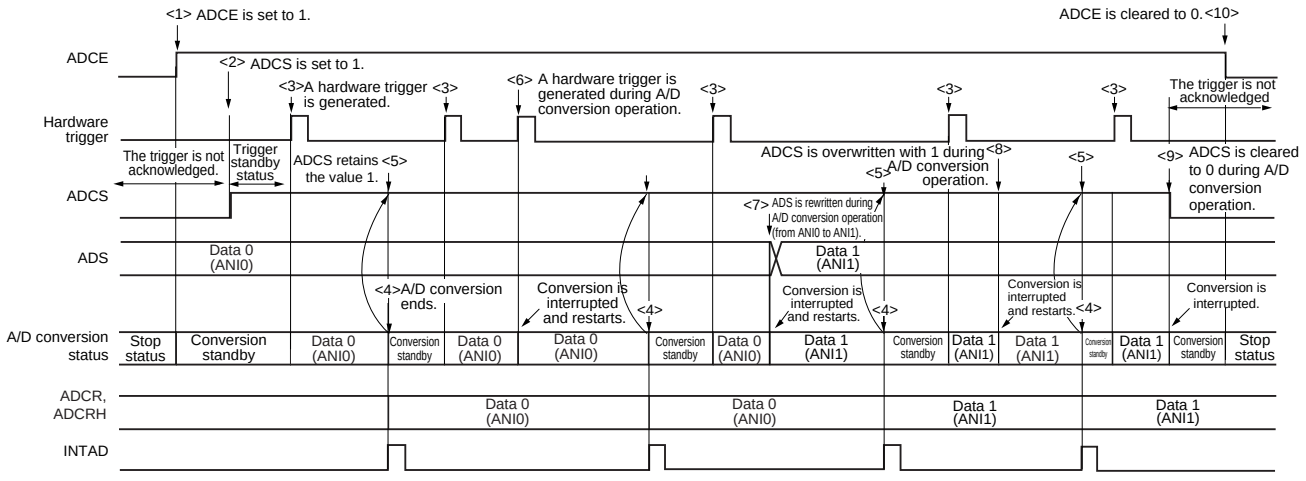
<7> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.

<8> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and A/D conversion restarts. The partially converted data is discarded.

<9> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status. However, the A/D converter does not power down in this status.

<10> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCS = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 10-22. Example of Hardware Trigger No-Wait Mode (Select Mode, One-Shot Conversion Mode) Operation Timing



Downloaded from Arrow.com.

10.6.7 Hardware trigger no-wait mode (scan mode, sequential conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1 μs), the ADCS bit of the ADM0 register is set to 1 to place the system in the hardware trigger standby status (and conversion does not start at this stage). Note that, while in this status, A/D conversion does not start even if ADCS is set to 1.

<3> If a hardware trigger is input while ADCS = 1, A/D conversion is performed on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS). A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.

<4> A/D conversion is sequentially performed on the four analog input channels, the conversion results are stored in the A/D conversion result register (ADCR, ADCRH) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD) is generated. After A/D conversion of the four channels ends, the A/D conversion of the channel following the specified channel automatically starts.

<5> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

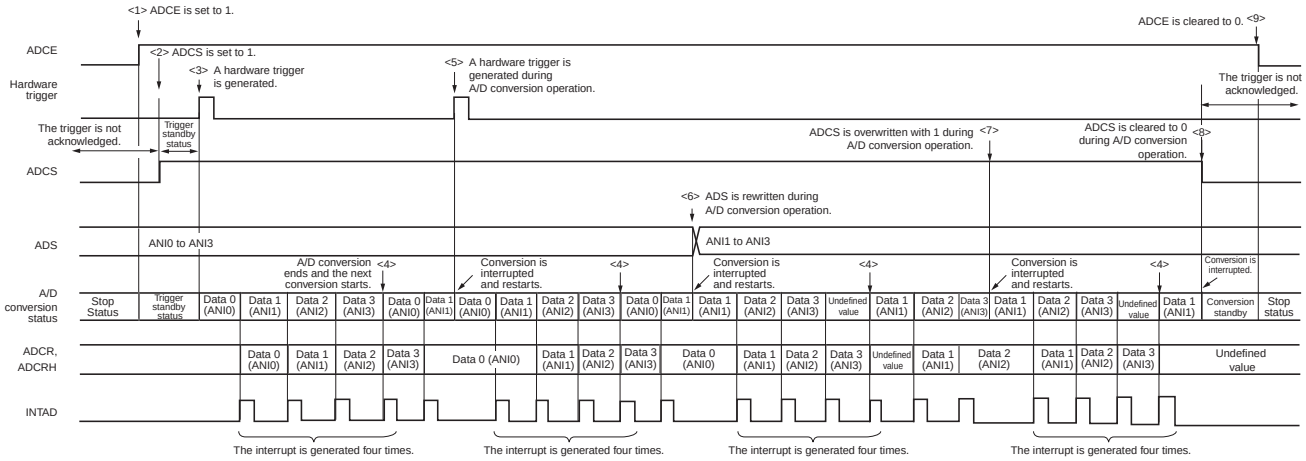
<6> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.

<7> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<8> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status. However, the A/D converter does not power down in this status.

<9> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCE = 0, specifying 1 for ADCS is ignored and A/D conversion does not start.

Figure 10-23. Example of Hardware Trigger No-Wait Mode (Scan Mode, Sequential Conversion Mode) Operation Timing



Downloaded from Arrow.com.

10.6.8 Hardware trigger no-wait mode (scan mode, one-shot conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1 μs), the ADCS bit of the ADM0 register is set to 1 to place the system in the hardware trigger standby status (and conversion does not start at this stage). Note that, while in this status, A/D conversion does not start even if ADCS is set to 1.

<3> If a hardware trigger is input while ADCS = 1, A/D conversion is performed on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS). A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.

<4> A/D conversion is sequentially performed on the four analog input channels, the conversion results are stored in the A/D conversion result register (ADCR, ADCRH) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD) is generated.

<5> After A/D conversion of the four channels ends, the ADCS bit remains set to 1, and the system enters the A/D conversion standby status.

<6> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

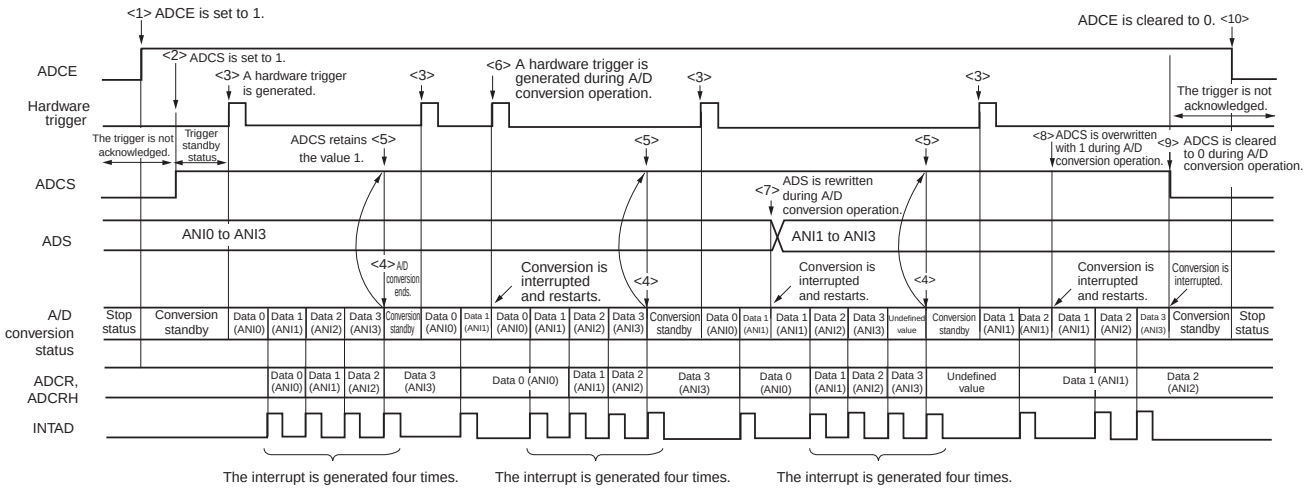
<7> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.

<8> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel. The partially converted data is discarded.

<9> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status. However, the A/D converter does not power down in this status.

<10> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCS = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 10-24. Example of Hardware Trigger No-Wait Mode (Scan Mode, One-Shot Conversion Mode) Operation Timing



Downloaded from Arrow.com.

10.6.9 Hardware trigger wait mode (select mode, sequential conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the hardware trigger standby status.

<2> If a hardware trigger is input while in the hardware trigger standby status, A/D conversion is performed on the analog input specified by the analog input channel specification register (ADS). The ADCS bit of the ADM0 register is automatically set to 1 according to the hardware trigger input.

<3> When A/D conversion ends, the conversion result is stored in the A/D conversion result register (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) is generated. After A/D conversion ends, the next A/D conversion immediately starts. (At this time, no hardware trigger is necessary.)

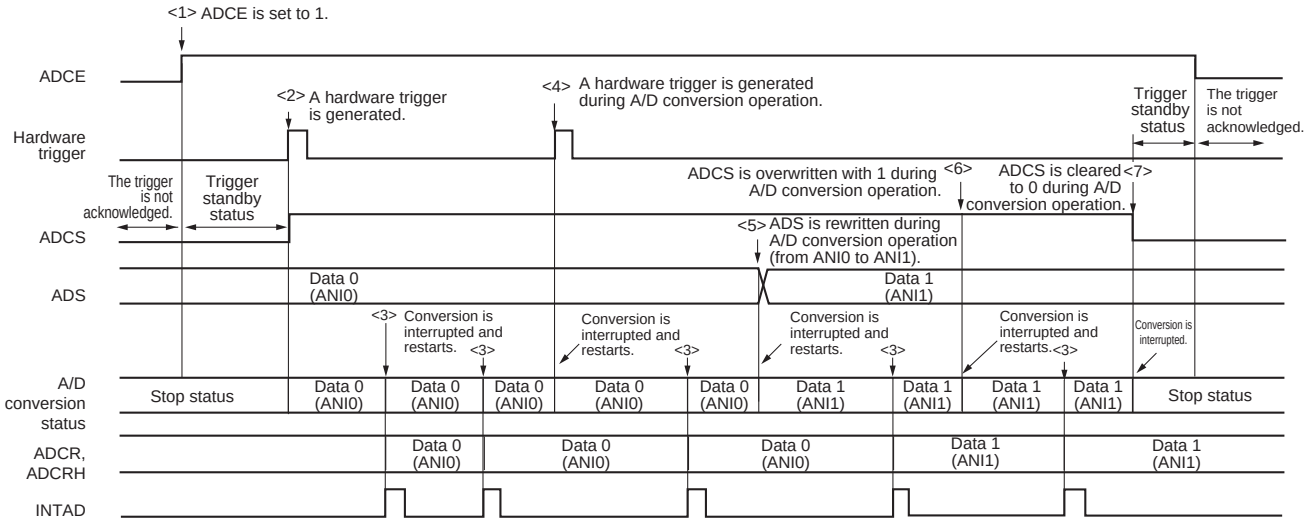
<4> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<5> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.

<6> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<7> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, the system enters the hardware trigger standby status, and the A/D converter enters the stop status. When ADCE = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 10-25. Example of Hardware Trigger Wait Mode (Select Mode, Sequential Conversion Mode) Operation Timing



Downloaded from Arrow.com.

10.6.10 Hardware trigger wait mode (select mode, one-shot conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the hardware trigger standby status.

<2> If a hardware trigger is input while in the hardware trigger standby status, A/D conversion is performed on the analog input specified by the analog input channel specification register (ADS). The ADCS bit of the ADM0 register is automatically set to 1 according to the hardware trigger input.

<3> When A/D conversion ends, the conversion result is stored in the A/D conversion result register (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) is generated.

<4> After A/D conversion ends, the ADCS bit is automatically cleared to 0, and the A/D converter enters the stop status.

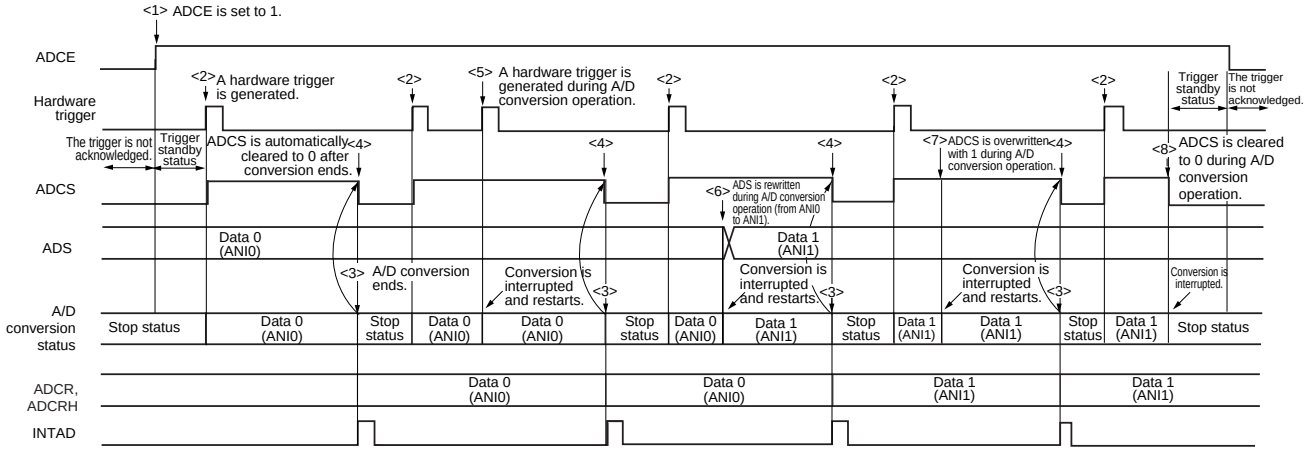
<5> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<6> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.

<7> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is initialized.

<8> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, the system enters the hardware trigger standby status, and the A/D converter enters the stop status. When ADCE = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 10-26. Example of Hardware Trigger Wait Mode (Select Mode, One-Shot Conversion Mode) Operation Timing



Downloaded from Arrow.com.

10.6.11 Hardware trigger wait mode (scan mode, sequential conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> If a hardware trigger is input while in the hardware trigger standby status, A/D conversion is performed on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS). The ADCS bit of the ADM0 register is automatically set to 1 according to the hardware trigger input. A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.

<3> A/D conversion is sequentially performed on the four analog input channels, the conversion results are stored in the A/D conversion result register (ADCR, ADCRH) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD) is generated. After A/D conversion of the four channels ends, the A/D conversion of the channel following the specified channel automatically starts.

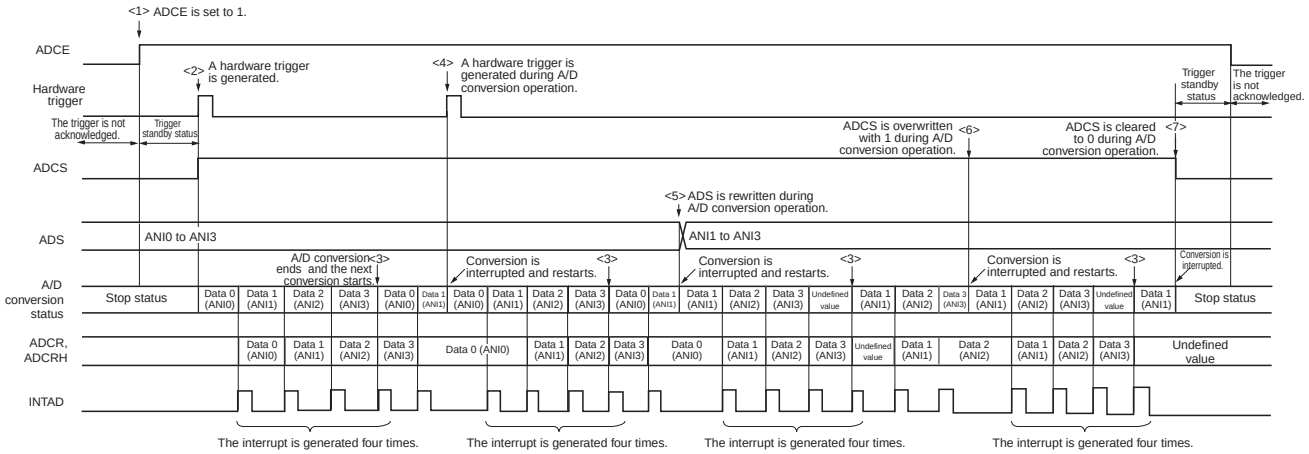
<4> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

<5> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.

<6> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<7> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, the system enters the hardware trigger standby status, and the A/D converter enters the stop status. When ADCE = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 10-27. Example of Hardware Trigger Wait Mode (Scan Mode, Sequential Conversion Mode) Operation Timing



Downloaded from Arrow.com.

10.6.12 Hardware trigger wait mode (scan mode, one-shot conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> If a hardware trigger is input while in the hardware trigger standby status, A/D conversion is performed on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS). The ADCS bit of the ADM0 register is automatically set to 1 according to the hardware trigger input. A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.

<3> A/D conversion is sequentially performed on the four analog input channels, the conversion results are stored in the A/D conversion result register (ADCR, ADCRH) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD) is generated.

<4> After A/D conversion ends, the ADCS bit is automatically cleared to 0, and the A/D converter enters the stop status.

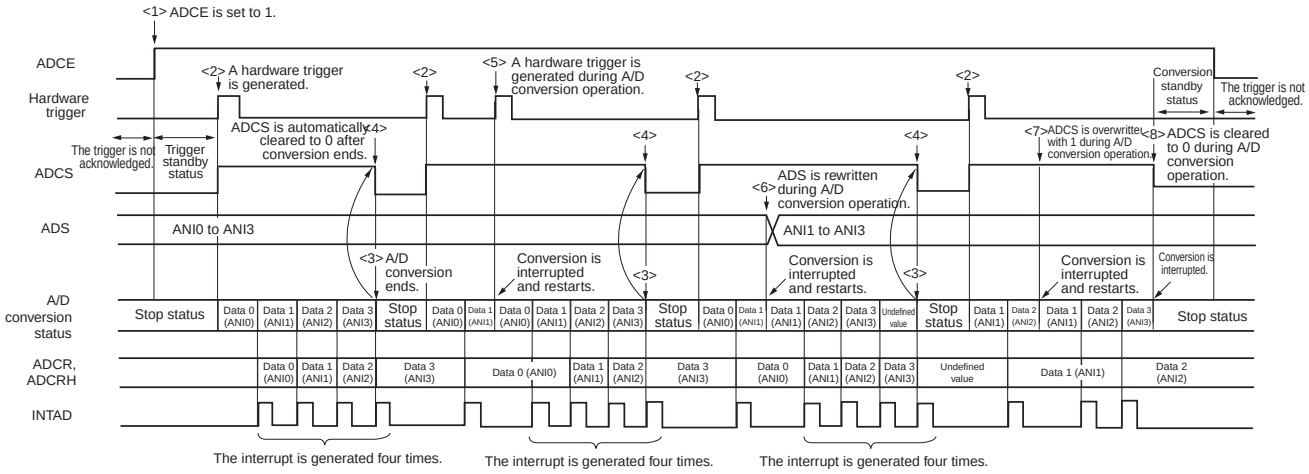
<5> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

<6> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.

<7> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<8> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, the system enters the hardware trigger standby status, and the A/D converter enters the stop status. When ADCE = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 10-28. Example of Hardware Trigger Wait Mode (Scan Mode, One-Shot Conversion Mode) Operation Timing



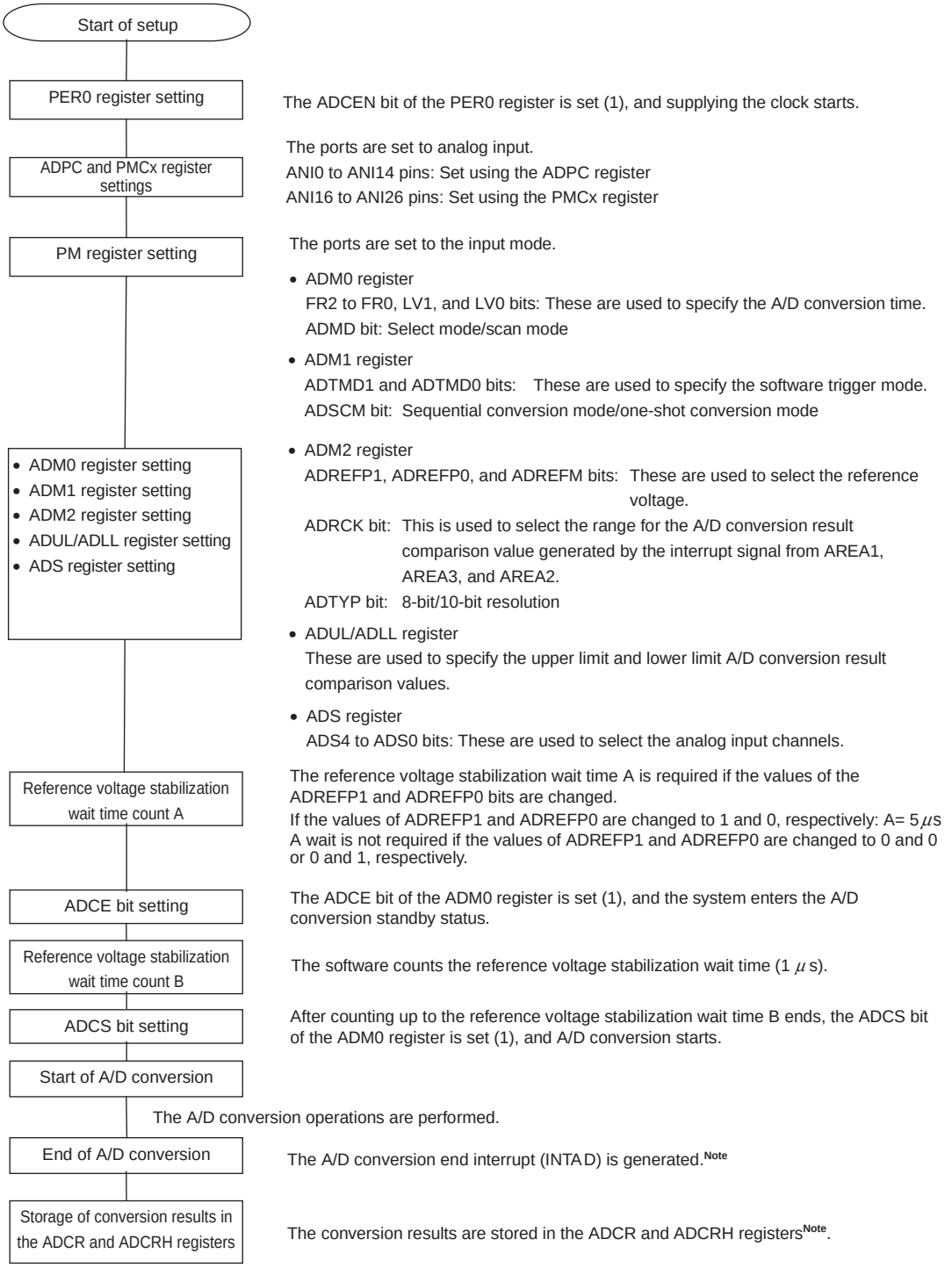
Downloaded from Arrow.com.

10.7 A/D Converter Setup Flowchart

The A/D converter setup flowchart in each operation mode is described below.

10.7.1 Setting up software trigger mode

Figure 10-29. Setting up Software Trigger Mode

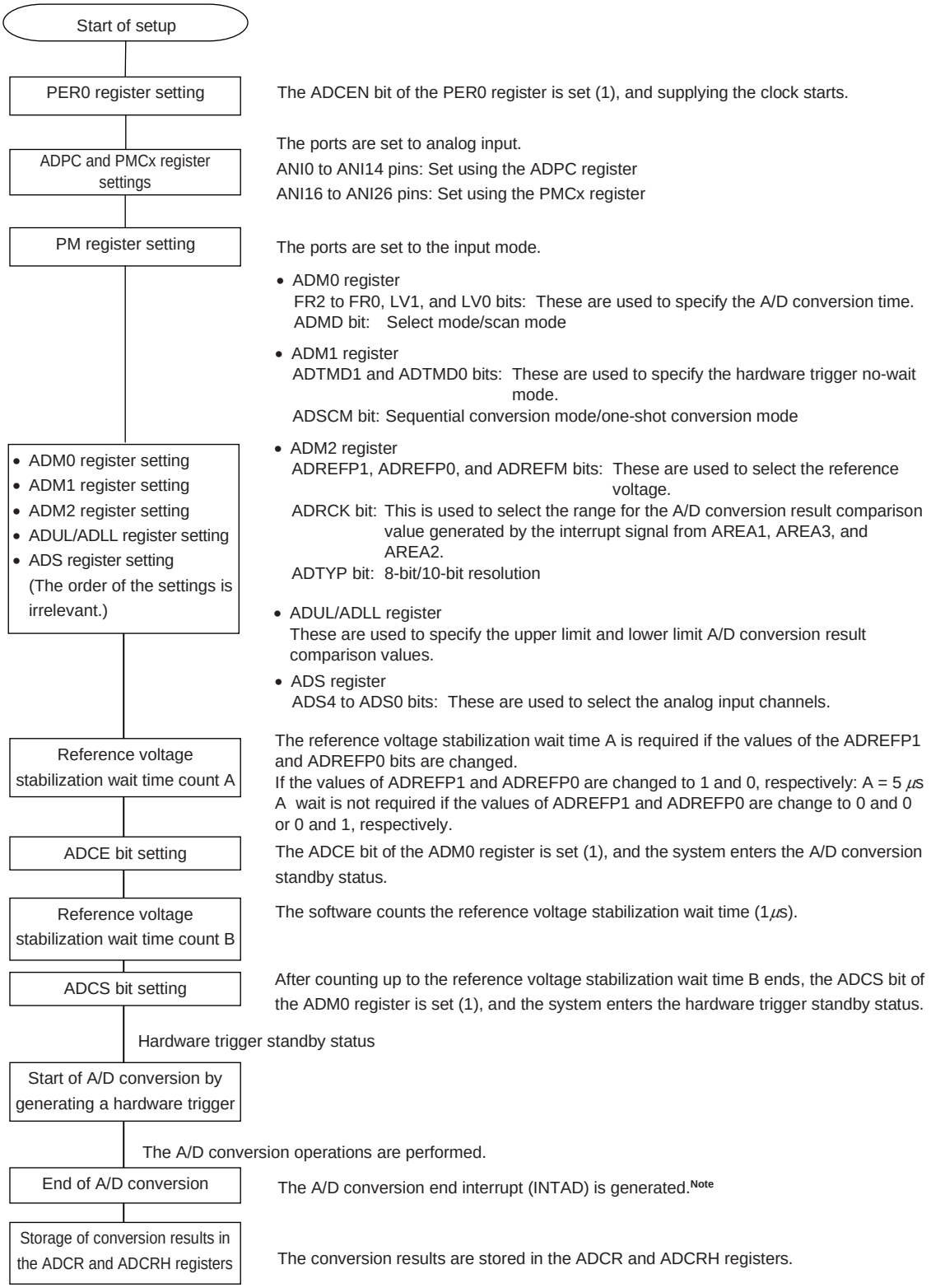


Note Depending on the settings of the ADRCK bit and ADUL/ADLL register, there is a possibility of no interrupt signal being generated. In this case, the results are not stored in the ADCR and ADCRH registers.

Downloaded from Arrow.com.

10.7.2 Setting up hardware trigger no-wait mode

Figure 10-30. Setting up Hardware Trigger No-Wait Mode

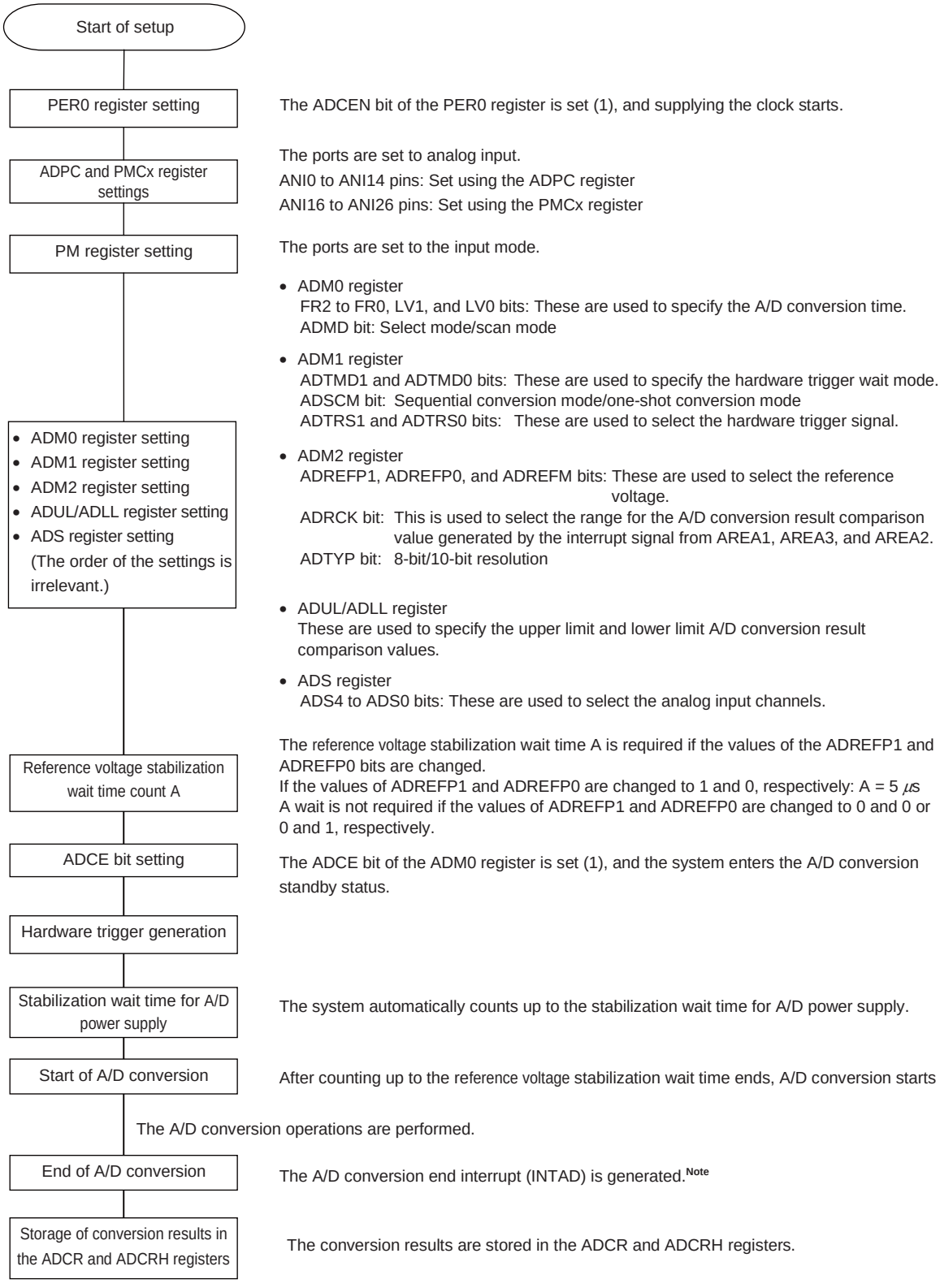


Note Depending on the settings of the ADRCK bit and ADUL/ADLL register, there is a possibility of no interrupt signal being generated. In this case, the results are not stored in the ADCR and ADCRH registers.

Downloaded from Arrow.com.

10.7.3 Setting up hardware trigger wait mode

Figure 10-31. Setting up Hardware Trigger Wait Mode

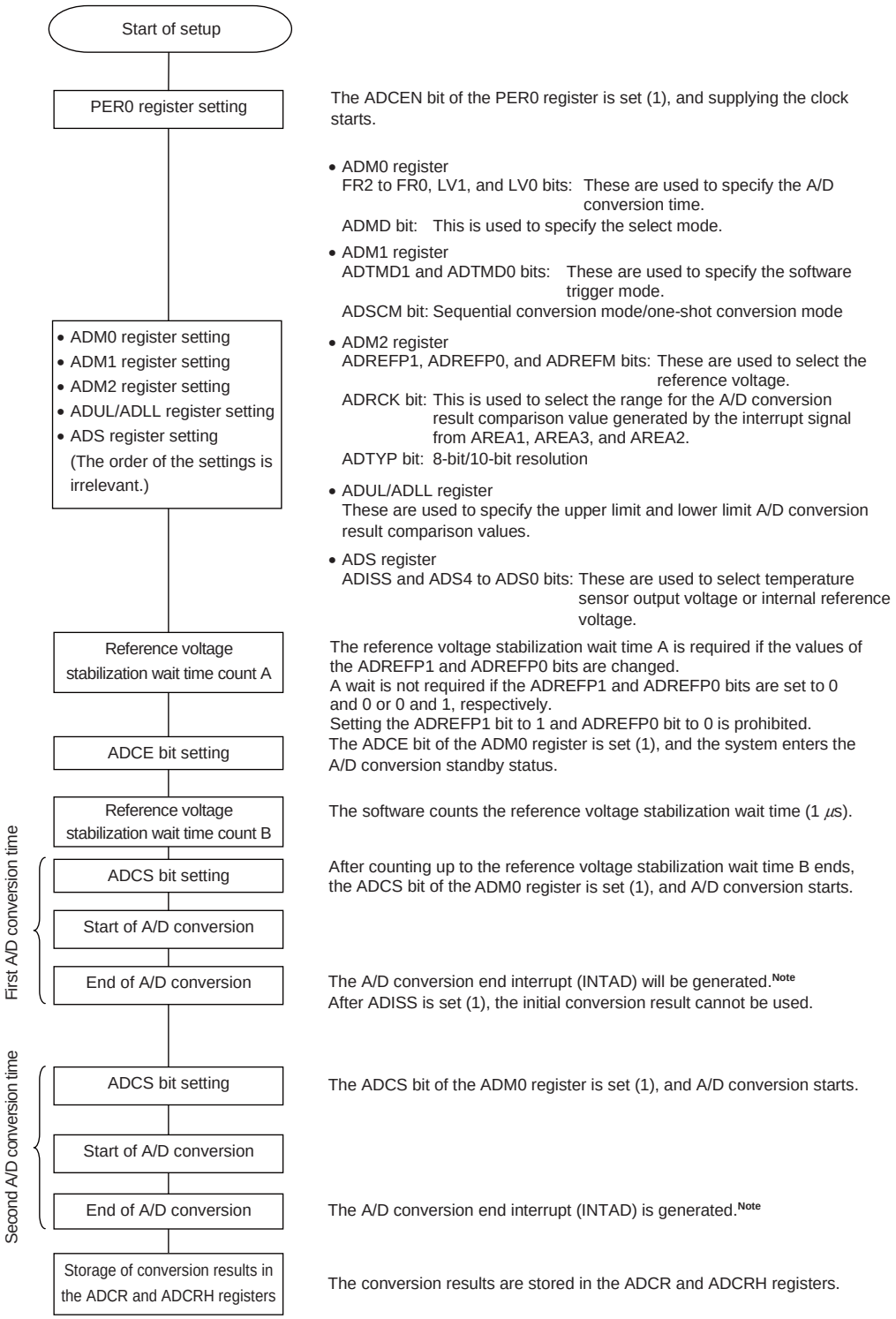


Note Depending on the settings of the ADRCK bit and ADUL/ADLL register, there is a possibility of no interrupt signal being generated. In this case, the results are not stored in the ADCR and ADCRH registers.

Downloaded from Arrow.com.

10.7.4 Setup when temperature sensor output voltage/internal reference voltage is selected (example for software trigger mode and one-shot conversion mode)

Figure 10-32. Setup when Temperature Sensor Output Voltage/Internal Reference Voltage is Selected



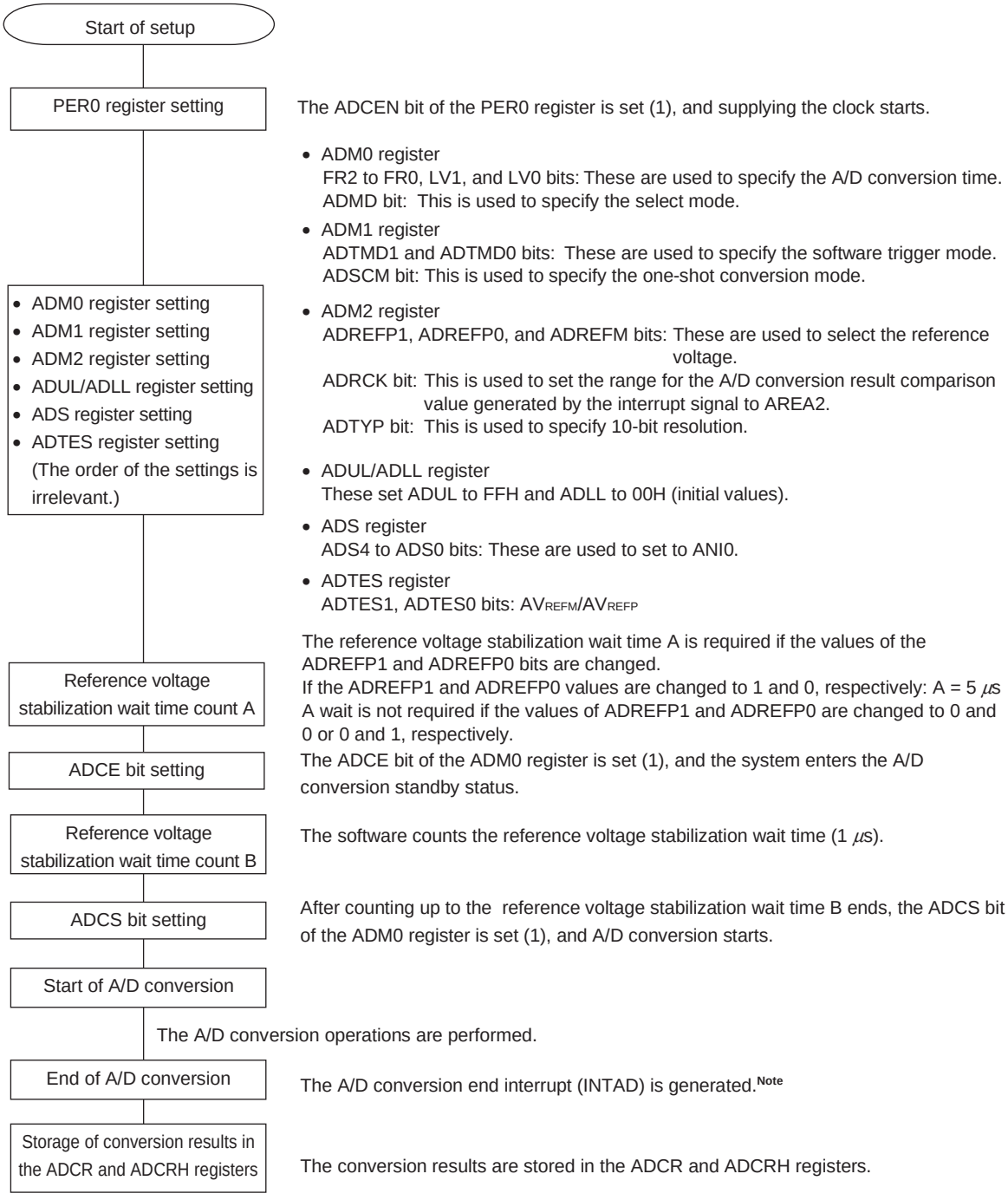
Note Depending on the settings of the ADRCK bit and ADUL/ADLL register, there is a possibility of no interrupt signal being generated. In this case, the results are not stored in the ADCR and ADCRH registers.

Caution This setting can be used only in HS (high-speed main) mode.

Downloaded from Arrow.com.

10.7.5 Setting up test mode

Figure 10-33. Setting up Test Mode



Note Depending on the settings of the ADRCK bit and ADUL/ADLL register, there is a possibility of no interrupt signal being generated. In this case, the results are not stored in the ADCR and ADCRH registers.

Caution For details about how to test the A/D converter, see 21.3.7 A/D test function.

Downloaded from Arrow.com.

10.8 SNOOZE mode function

In the SNOOZE mode, A/D conversion is triggered by inputting a hardware trigger in the STOP mode. Normally, A/D conversion is stopped while in the STOP mode, but, by using the SNOOZE mode function, A/D conversion can be performed without operating the CPU by inputting a hardware trigger. This is effective for reducing the operation current.

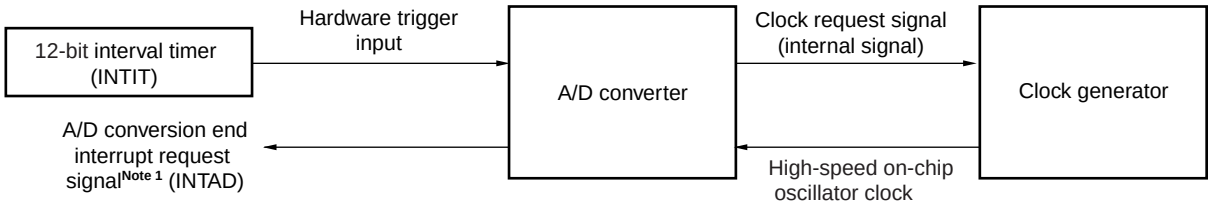
If the A/D conversion result range is specified using the ADUL and ADLL registers, A/D conversion results can be judged at a certain interval of time in SNOOZE mode. Using this function enables power supply voltage monitoring and input key judgment based on A/D inputs.

In the SNOOZE mode, only the following two conversion modes can be used:

- Hardware trigger wait mode (select mode, one-shot conversion mode)
- Hardware trigger wait mode (scan mode, one-shot conversion mode)

Caution SNOOZE mode can only be specified when the high-speed on-chip oscillator clock is selected for f_{CLK}.

Figure 10-34. Block Diagram When Using SNOOZE Mode Function



When using the SNOOZE mode function, the initial setting of each register is specified before switching to the STOP mode (for details about these settings, see **10.7.3 Setting up hardware trigger wait mode**Note 2). Just before switching to the STOP mode, bit 2 (AWC) of the A/D converter mode register 2 (ADM2) is set to 1. After the initial settings are specified, bit 0 (ADCE) of the A/D converter mode register 0 (ADM0) is set to 1.

If a hardware trigger is input after switching to the STOP mode, the high-speed on-chip oscillator clock is supplied to the A/D converter. After supplying this clock, the system automatically counts up to the A/D power supply stabilization wait time, and then A/D conversion starts.

The SNOOZE mode operation after A/D conversion ends differs depending on whether an interrupt signal is generated Note 1.

Notes 1. Depending on the setting of the A/D conversion result comparison function (ADRCK bit, ADUL/ADLL register), there is a possibility of no interrupt signal being generated.

2. Be sure to set the ADM1 register to E2H or E3H.

Remarks 1. The hardware trigger is INTIT.

2. Specify the hardware trigger by using the A/D Converter Mode Register 1 (ADM1)

Downloaded from Arrow.com.

(1) If an interrupt request is generated after A/D conversion ends

If the A/D conversion result value is inside the range of values specified by the A/D conversion result comparison function (which is set up by using the ADRCK bit and ADUL/ADLL register), the A/D conversion end interrupt request signal (INTAD) is generated.

- While in the select mode

  After A/D conversion ends and the A/D conversion end interrupt request signal (INTAD) is generated, the clock request signal remains at the high level, and the A/D converter switches from the SNOOZE mode to the normal operation mode.

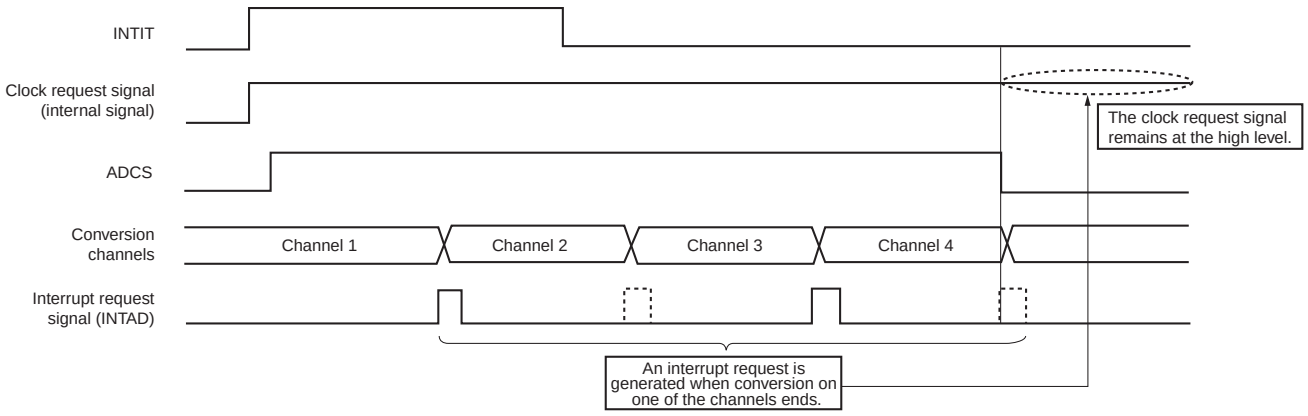
  At this time, be sure to clear bit 2 (AWC = 0: SNOOZE mode release) of the A/D converter mode register 2 (ADM2). If the AWC bit is left set to 1, A/D conversion will not start normally in the subsequent SNOOZE or normal operation mode.

- While in the scan mode

  If even one A/D conversion end interrupt request signal (INTAD) is generated during A/D conversion of the four channels, the clock request signal remains at the high level, and the A/D converter switches from the SNOOZE mode to the normal operation mode.

  At this time, be sure to clear bit 2 (AWC = 0: SNOOZE mode release) of A/D converter mode register 2 (ADM2) to 0. If the AWC bit is left set to 1, A/D conversion will not start normally in the subsequent SNOOZE or normal operation mode.

Figure 10-35. Operation Example When Interrupt Request Is Generated After A/D Conversion Ends (While in Scan Mode)



Downloaded from Arrow.com.

(2) If no interrupt request is generated after A/D conversion ends

If the A/D conversion result value is outside the range of values specified by the A/D conversion result comparison function (which is set up by using the ARDCK bit and ADUL/ADLL register), the A/D conversion end interrupt request signal (INTAD) is not generated.

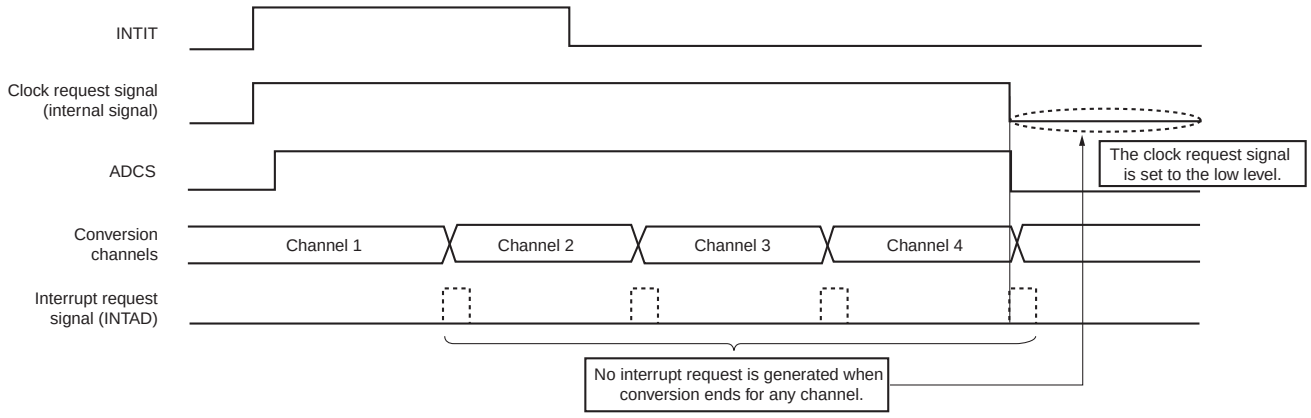
- While in the select mode

  If the A/D conversion end interrupt request signal (INTAD) is not generated after A/D conversion ends, the clock request signal (an internal signal) is automatically set to the low level, and supplying the high-speed on-chip oscillator clock stops. If a hardware trigger is input later, A/D conversion work is again performed in the SNOOZE mode.

- While in the scan mode

  If the A/D conversion end interrupt request signal (INTAD) is not generated even once during A/D conversion of the four channels, the clock request signal (an internal signal) is automatically set to the low level after A/D conversion of the four channels ends, and supplying the high-speed on-chip oscillator clock stops. If a hardware trigger is input later, A/D conversion work is again performed in the SNOOZE mode.

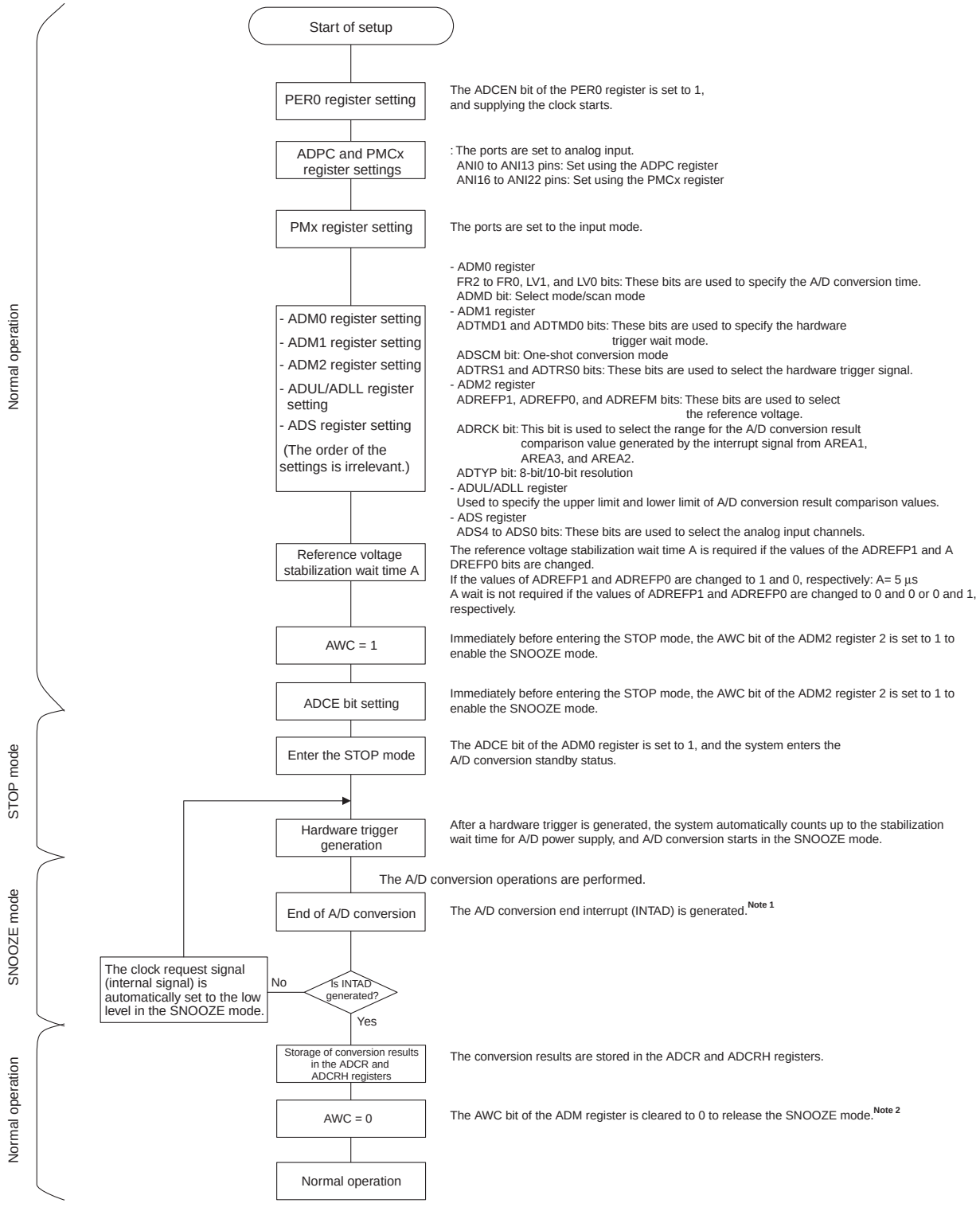
Figure 10-36. Operation Example When No Interrupt Request Is Generated After A/D Conversion Ends (While in Scan Mode)



Downloaded from Arrow.com.

<R>

Figure 10-37. Flowchart for Setting Up SNOOZE Mode



Notes 1. If the A/D conversion end interrupt request signal (INTAD) is not generated due to the setting of the ADRCK bit and ADUL/ADLL register, conversion results are not stored in the ADCR and ADCRH registers. The system enters the STOP mode again. If a hardware trigger is input later, A/D conversion is performed again in the SNOOZE mode.

2. If the AWC bit is left set to 1, A/D conversion will not start normally in spite of the subsequent SNOOZE or normal operation mode. Be sure to clear the AWC bit to 0.

Downloaded from Arrow.com.

10.9 How to Read A/D Converter Characteristics Table

Here, special terms unique to the A/D converter are explained.

(1) Resolution

This is the minimum analog input voltage that can be identified. That is, the percentage of the analog input voltage per bit of digital output is called 1LSB (Least Significant Bit). The percentage of 1LSB with respect to the full scale is expressed by %FSR (Full Scale Range).

1LSB is as follows when the resolution is 10 bits.

$1LSB = 1/2^{10} = 1/1024$
$= 0.098\%FSR$

Accuracy has no relation to resolution, but is determined by overall error.

(2) Overall error

This shows the maximum error value between the actual measured value and the theoretical value.
Zero-scale error, full-scale error, integral linearity error, and differential linearity errors that are combinations of these express the overall error.
Note that the quantization error is not included in the overall error in the characteristics table.

(3) Quantization error

When analog values are converted to digital values, a ±1/2LSB error naturally occurs. In an A/D converter, an analog input voltage in a range of ±1/2LSB is converted to the same digital code, so a quantization error cannot be avoided.
Note that the quantization error is not included in the overall error, zero-scale error, full-scale error, integral linearity error, and differential linearity error in the characteristics table.

Figure 10-38. Overall Error

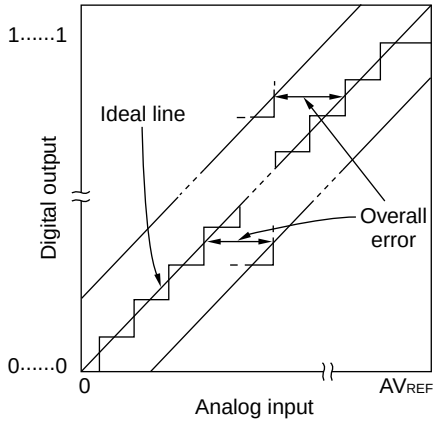
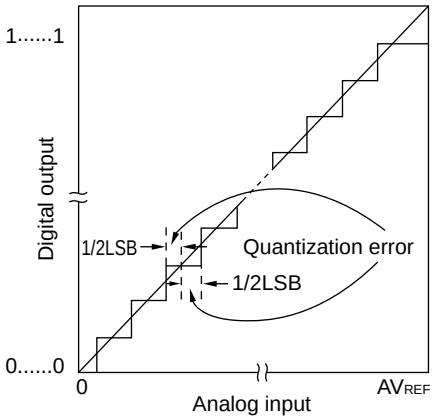


Figure 10-39. Quantization Error



(4) Zero-scale error

This shows the difference between the actual measurement value of the analog input voltage and the theoretical value (1/2LSB) when the digital output changes from 0.....000 to 0.....001.
If the actual measurement value is greater than the theoretical value, it shows the difference between the actual measurement value of the analog input voltage and the theoretical value (3/2LSB) when the digital output changes from 0.....001 to 0.....010.

Downloaded from Arrow.com.

(5) Full-scale error

This shows the difference between the actual measurement value of the analog input voltage and the theoretical value (full-scale – 3/2LSB) when the digital output changes from 1.....110 to 1.....111.

(6) Integral linearity error

This shows the degree to which the conversion characteristics deviate from the ideal linear relationship. It expresses the maximum value of the difference between the actual measurement value and the ideal straight line when the zero-scale error and full-scale error are 0.

(7) Differential linearity error

While the ideal width of code output is 1LSB, this indicates the difference between the actual measurement value and the ideal value.

Figure 10-40. Zero-Scale Error

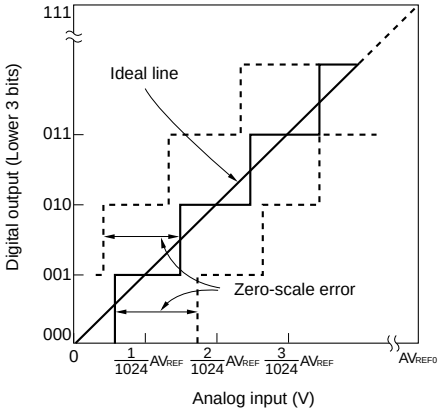


Figure 10-41. Full-Scale Error

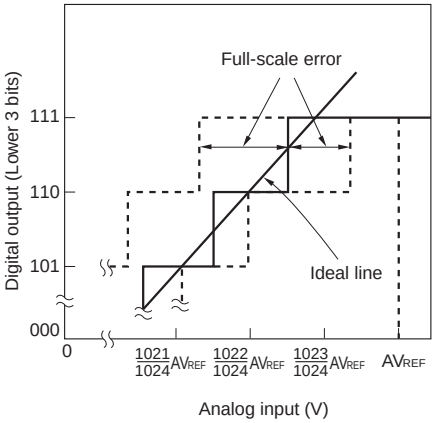


Figure 10-42. Integral Linearity Error

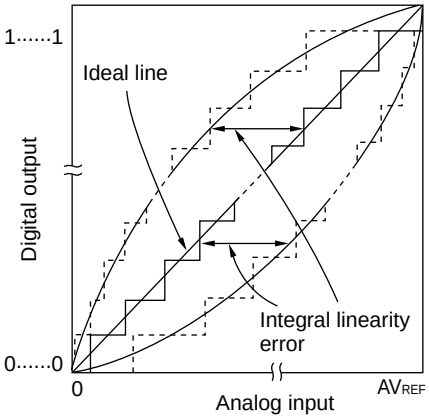
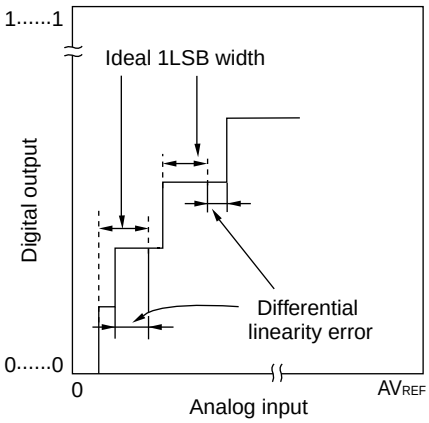


Figure 10-43. Differential Linearity Error



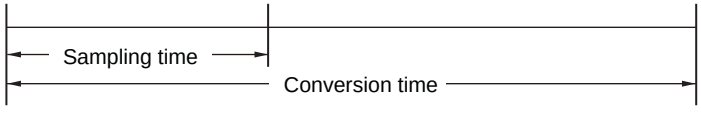
(8) Conversion time

This expresses the time from the start of sampling to when the digital output is obtained.

The sampling time is included in the conversion time in the characteristics table.

(9) Sampling time

This is the time the analog switch is turned on for the analog voltage to be sampled by the sample & hold circuit.



Downloaded from Arrow.com.

10.10 Cautions for A/D Converter

(1) Operating current in STOP mode

Shift to STOP mode after stopping the A/D converter (by setting bit 7 (ADCS) of A/D converter mode register 0 (ADM0) to 0). The operating current can be reduced by setting bit 0 (ADCE) of the ADM0 register to 0 at the same time.

To restart from the standby status, clear bit 0 (ADIF) of interrupt request flag register 1H (IF1H) to 0 and start operation.

(2) Input range of ANI0 to ANI3 and ANI16 to ANI22 pins

Observe the rated range of the ANI0 to ANI3 and ANI16 to ANI22 pins input voltage. If a voltage of V_{DD} and AV_{REFP} or higher and V_{SS} and AV_{REFM} or lower (even in the range of absolute maximum ratings) is input to an analog input channel, the converted value of that channel becomes undefined. In addition, the converted values of the other channels may also be affected.

When internal reference voltage (1.45 V) is selected reference voltage for the + side of the A/D converter, do not input internal reference voltage (1.45 V) or higher voltage to a pin selected by the ADS register. However, it is no problem that a pin not selected by the ADS register is greater than the internal reference voltage (1.45 V).

Caution Internal reference voltage (1.45 V) can be used only in HS (high-speed main) mode.

(3) Conflicting operations

<1> Conflict between the A/D conversion result register (ADCR, ADCRH) write and the ADCR or ADCRH register read by instruction upon the end of conversion

The ADCR or ADCRH register read has priority. After the read operation, the new conversion result is written to the ADCR or ADCRH registers.

<2> Conflict between the ADCR or ADCRH register write and the A/D converter mode register 0 (ADM0) write, the analog input channel specification register (ADS), or A/D port configuration register (ADPC) write upon the end of conversion

The ADM0, ADS, or ADPC registers write has priority. The ADCR or ADCRH register write is not performed, nor is the conversion end interrupt signal (INTAD) generated.

(4) Noise countermeasures

To maintain the 10-bit resolution, attention must be paid to noise input to the AV_{REFP}, V_{DD}, ANI0 to ANI3, and ANI16 to ANI22 pins.

<1> Connect a capacitor with a low equivalent resistance and a good frequency response (capacitance of about 0.01 μF) via the shortest possible run of relatively thick wiring to the power supply.

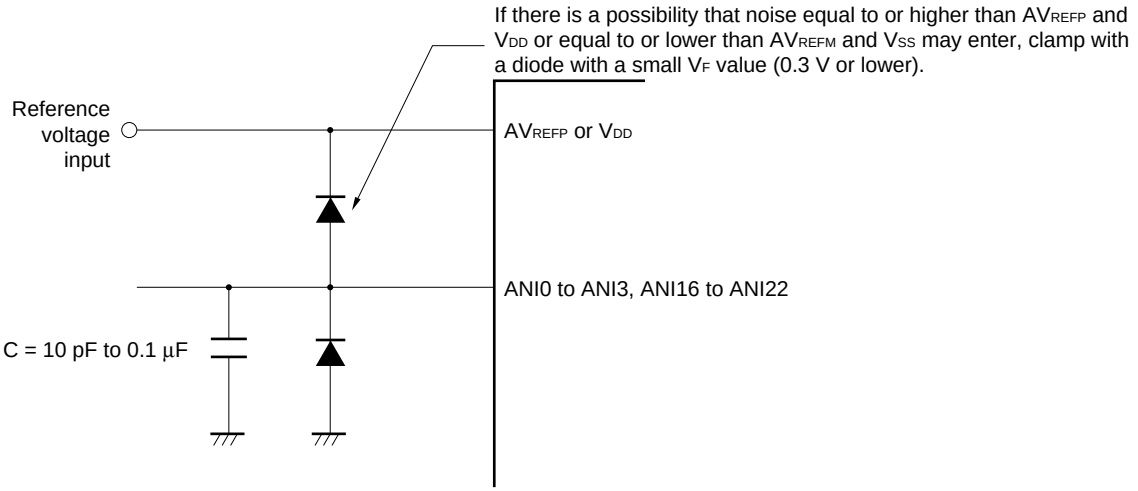
<2> The higher the output impedance of the analog input source, the greater the influence. To reduce the noise, connecting external capacitor as shown in Figure 10-44 is recommended.

<3> Do not switch these pins with other pins during conversion.

<4> The accuracy is improved if the HALT mode is set immediately after the start of conversion.

Downloaded from Arrow.com.

Figure 10-44. Analog Input Pin Connection



(5) Analog input (ANIn) pins

<1> The analog input pins (ANI0 to ANI3) are also used as input port pins (P20 to P23).
When A/D conversion is performed with any of the ANI0 to ANI3 pins selected, do not change output value to alternate port P20 to P23 while conversion is in progress; otherwise the conversion resolution may be degraded.

<2> If a pin adjacent to a pin that is being A/D converted is used as a digital I/O port pin, the A/D conversion result might differ from the expected value due to a coupling noise. Be sure to prevent a pulse with sharp transitions such as a digital signal from being input or output during A/D conversion.

(6) Input impedance of analog input (ANIn) pins

This A/D converter charges a sampling capacitor for sampling during sampling time.
Therefore, only a leakage current flows when sampling is not in progress, and a current that charges the capacitor flows during sampling. Consequently, the input impedance fluctuates depending on whether sampling is in progress, and on the other states.
To make sure that sampling is effective, however, we recommend using the converter with analog input sources that have output impedances no greater than 1 kΩ. If a source has a higher output impedance, lengthen the sampling time or connect a capacitor (with a value of about 100 μF) to the ANI0 to ANI3 and ANI16 to ANI22 pins (see **Figure 10-44**). The voltage which is charged in the sampling capacitor become undefined when the ADCS bit is set to 0 or conversion is restarted during conversion operations. Accordingly, the state of conversion is undefined when charging starts in the next round of conversion after the ADCS bit is set to 0 or when conversion is repeated. Thus, to secure full charging regardless of the size of fluctuations in the analog signal, ensure that the output impedances of the sources of analog inputs are low or secure sufficient time for the completion of conversion.

Downloaded from Arrow.com.

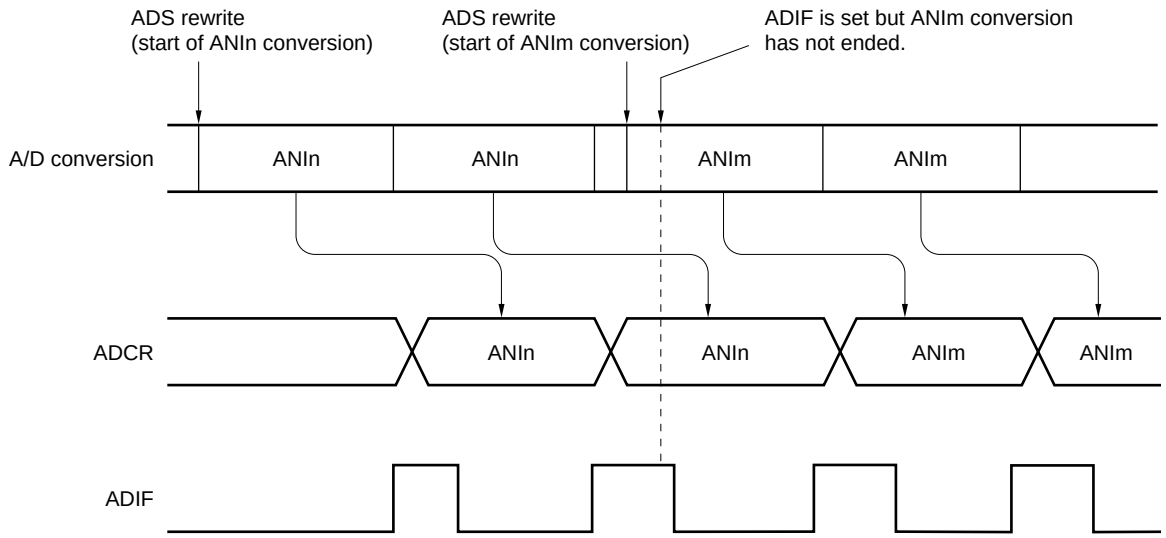
(7) Interrupt request flag (ADIF)

The interrupt request flag (ADIF) is not cleared even if the analog input channel specification register (ADS) is changed.

Therefore, if an analog input pin is changed during A/D conversion, the A/D conversion result and ADIF flag for the pre-change analog input may be set just before the ADS register rewrite. Caution is therefore required since, at this time, when ADIF flag is read immediately after the ADS register rewrite, ADIF flag is set despite the fact A/D conversion for the post-change analog input has not ended.

When A/D conversion is stopped and then resumed, clear ADIF flag before the A/D conversion operation is resumed.

Figure 10-45. Timing of A/D Conversion End Interrupt Request Generation



(8) Conversion results just after A/D conversion start

While in the software trigger mode or hardware trigger no-wait mode, the first A/D conversion value immediately after A/D conversion starts may not fall within the rating range if the ADCS bit is set to 1 within 1 μs after the ADCE bit was set to 1. Take measures such as polling the A/D conversion end interrupt request (INTAD) and removing the first conversion result.

(9) A/D conversion result register (ADCR, ADCRH) read operation

When a write operation is performed to the A/D converter mode register 0 (ADM0), analog input channel specification register (ADS), A/D port configuration register (ADPC), or port mode control register (PMCx), the contents of the ADCR and ADCRH registers may become undefined. Read the conversion result following conversion completion before writing to the ADM0, ADS, ADPC, or PMCx register. Using a timing other than the above may cause an incorrect conversion result to be read.

Downloaded from Arrow.com.

(10) Internal equivalent circuit

The equivalent circuit of the analog input block is shown below.

Figure 10-46. Internal Equivalent Circuit of ANIn Pin

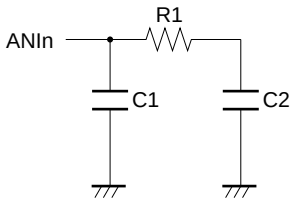


Table 10-4. Resistance and Capacitance Values of Equivalent Circuit (Reference Values)

AV_{REFP}, V_{DD}	ANIn Pins	R1 [kΩ]	C1 [pF	C2 [pF]
3.6 V ≤ V_{DD} ≤ 5.5 V	ANI0 to ANI3	14	8	2.5
	ANI16 to ANI22	18		7.0
2.7 V ≤ V_{DD} < 3.6 V	ANI0 to ANI3	39		2.5
	ANI16 to ANI22	53		7.0
1.8 V ≤ V_{DD} < 2.7 V	ANI0 to ANI3	231		2.5
	ANI16 to ANI22	321		7.0

Remark The resistance and capacitance values shown in Table 10-6 are not guaranteed values.

(11) Starting the A/D converter

Start the A/D converter after the AV_{REFP} and V_{DD} voltages stabilize.

Downloaded from Arrow.com.

CHAPTER 11 SERIAL ARRAY UNIT

Serial array unit 0 has two serial channels in 20- and 24-pinproducts and four serial channels in 30-pin products, and serial array unit 1 mounted 30-pin products, has two serial channels.

Each channel can achieve 3-wire serial (CSI), UART, and simplified I^2C communication.

Function assignment of each channel supported by the RL78/G12 is as shown below.

20- or 24-pin products

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I^2C
0	0	CSI00	UART0	IIC00Note
	1	CSI01Note		IIC01Note

30-pin products

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I^2C
0	0	CSI00	UART0	IIC00Note
	1	–		–
	2	–	UART1Note	–
	3	CSI11Note		IIC11Note
1	0	CSI20Note	UART2Note	IIC20Note
	1	–		–

Note Provided in the R5F102 products only.

A single channel cannot be used under multiple communication methods. When a different communication method is to be configured, use another channel.

When using CSI00, CSI20, IIC00, IIC20, UART0, UART1, or UART2, communication between devices with different voltages (1.8, 2.5, or 3 V) is possible, except when the serial I/O for UART0 of a 20- or 24-pin product is assigned to P6 by the setting of the I/O redirection register (PIOR1 = 1). For details about the settings, see **4.4.4 Handling different potentials (1.8 V, 2.5 V, and 3 V) by using I/O buffers.**

Downloaded from Arrow.com.

11.1 Functions of Serial Array Unit

Each serial interface supported by the RL78/G12 has the following features.

11.1.1 3-wire serial I/O (CSI00, CSI01, CSI11, CSI20)

Data is transmitted or received in synchronization with the serial clock (SCK) output from the master channel.

3-wire serial communication is clocked communication performed by using three communication lines: one for the serial clock (SCK), one for transmitting serial data (SO), one for receiving serial data (SI).

For details about the settings, see **11.5 Operation of 3-Wire Serial I/O (CSI00, CSI01, CSI11, CSI20) Communication.**

[Data transmission/reception]

- Data length of 7 or 8 bits
- Phase control of transmit/receive data
- MSB/LSB first selectable

[Clock control]

- Master/slave selection
- Phase control of I/O clock
- Setting of transfer period by prescaler and internal counter of each channel

<R> - Maximum transfer rate[Note]

During master communication: Max. f_{CLK}/2 (CSI00 only)
Max. f_{CLK}/4

During slave communication: Max. f_{MCK}/6

[Interrupt function]

- Transfer end interrupt/buffer empty interrupt

[Error detection flag]

- Overrun error

In addition, CSI00 (channel 0 of unit 0) supports the SNOOZE mode. When SCK00 pin input is detected while in the STOP mode, the SNOOZE mode makes data reception that does not require the CPU possible. Only CSI00, which supports asynchronous receive operations, can be specified.

<R> **Note** Use the clocks within a range satisfying the SCK cycle time (t_{KCY}) characteristics. For details, see **CHAPTER 28 ELECTRICAL SPECIFICATIONS (A, D: T_A = -40 to +85°C)** or **CHAPTER 29 ELECTRICAL SPECIFICATIONS (G: T_A = -40 to +105°C)**.

Downloaded from Arrow.com.

11.1.2 UART (UART0 to UART2)

This is a start-stop synchronization function using two lines: serial data transmission (TxD) and serial data reception (RxD) lines. By using these two communication lines, each data frame, which consist of a start bit, data, parity bit, and stop bit, is transferred asynchronously (using the internal baud rate) between the microcontroller and the other communication party. Full-duplex UART communication can be performed by using a channel dedicated to transmission (even-numbered channel) and a channel dedicated to reception (odd-numbered channel).

For details about the settings, see **11.6 Operation of UART (UART0 to UART2) Communication**.

[Data transmission/reception]

- Data length of 7, 8, or 9 bits[Note]
- Select the MSB/LSB first
- Level setting of transmit/receive data and select of reverse
- Parity bit appending and parity check functions
- Stop bit appending

[Interrupt function]

- Transfer end interrupt/buffer empty interrupt
- Error interrupt in case of framing error, parity error, or overrun error

[Error detection flag]

- Framing error, parity error, or overrun error

Note Only UART0 can be specified for the 9-bit data length.

In addition, UART0 reception (channel 1 of unit 0) supports the SNOOZE mode. When RxD0 pin input is detected while in the STOP mode, the SNOOZE mode makes data reception that does not require the CPU possible.

Downloaded from Arrow.com.

11.1.3 Simplified I²C (IIC00, IIC01, IIC11, IIC20)

This is a clocked communication function to communicate with two or more devices by using two lines: serial clock (SCL) and serial data (SDA). This simplified I²C is designed for single communication with a device such as EEPROM, flash memory, or A/D converter, and therefore, it functions only as a master.

Make sure by using software, as well as operating the control registers, that the AC specifications of the start and stop conditions are observed.

For details about the settings, see **11.7 Operation of Simplified I²C (IIC00, IIC01, IIC11, IIC20) Communication**.

[Data transmission/reception]

- Master transmission, master reception (only master function with a single master)
- ACK output function[Note] and ACK detection function
- Data length of 8 bits (When an address is transmitted, the address is specified by the higher 7 bits, and the least significant bit is used for R/W control.)
- Manual generation of start condition and stop condition

[Interrupt function]

- Transfer end interrupt

[Error detection flag]

- ACK error, or overrun error

* [Functions not supported by simplified I²C]

- Slave transmission, slave reception
- Arbitration loss detection function
- Wait detection functions

Note When receiving the last data, 0 is written to the SOEmn bit of the serial output enable register m (SOEm) and serial communication data output is stopped, disabling ACK output. See the processing flow in **11.7.3 (2)** for details.

Remarks 1. To use a fully functional I²C bus, see **CHAPTER 12 SERIAL INTERFACE IICA**.

2. m: Unit number, n: Channel number (mn = 00, 01, 03, 10)

Downloaded from Arrow.com.

11.2 Configuration of Serial Array Unit

The serial array unit includes the following hardware.

Table 11-1. Configuration of Serial Array Unit

Item	Configuration
Shift register	8 or 9 bits[Note 1]
Buffer register	Lower 8 or 9 bits of serial data register mn (SDRmn)[Notes 1, 2]
Serial clock I/O	SCK00, SCK01, SCK11, and SCK20 pins (for 3-wire serial I/O), SCL00, SCL01, SCL11, and SCL20 pins (for simplified I^2C)
Serial data input	SI00, SI01, SI11, and SI20 pins (for 3-wire serial I/O), RxD0, RxD1, and RxD2 pins (for UART)
<R> Serial data output	SO00, SO01, SO11, and SO20 pins (for 3-wire serial I/O), TxD0, TxD1, and TxD2 pins (for UART)
Serial data I/O	SDA00, SDA01, SDA11 and SDA20 pins (for simplified I^2C)
Control registers	<Registers of unit setting block> • Peripheral enable register 0 (PER0) • Serial clock select register m (SPSm) • Serial channel enable status register m (SEm) • Serial channel start register m (SSm) • Serial channel stop register m (STm) • Serial output enable register m (SOEm) • Serial output register m (SOm) • Serial output level register m (SOLm) • Serial standby control register m (SSCm) • Noise filter enable register 0 (NFEN0)
	<Registers of each channel> • Serial data register mn (SDRmn) • Serial mode register mn (SMRmn) • Serial communication operation setting register mn (SCRmn) • Serial status register mn (SSRmn) • Serial flag clear trigger register mn (SIRmn)
	• Port input mode register 0, 1 (PIM0, PIM1) • Port output mode registers 1, 4, 5 (POM1, POM4, POM5) • Port mode control registers 0, 1, 4 (PMC0, PMC1, PMC4) • Port mode registers 0, 1, 3 to 6 (PM0, PM1, PM3 to PM6) • Port register 0, 1, 3 to 6 (P0, P1, P3 to P6)

Notes 1. The number of bits used as shift register or buffer register varies depending on the unit or channel.
mn = 00, 01: lower 9 bits, mn = 02, 03, 10, 11: lower 8 bits

2. The lower 8 bits of serial data register mn (SDRmn) can be read or written as the following SFR, depending on the communication mode.

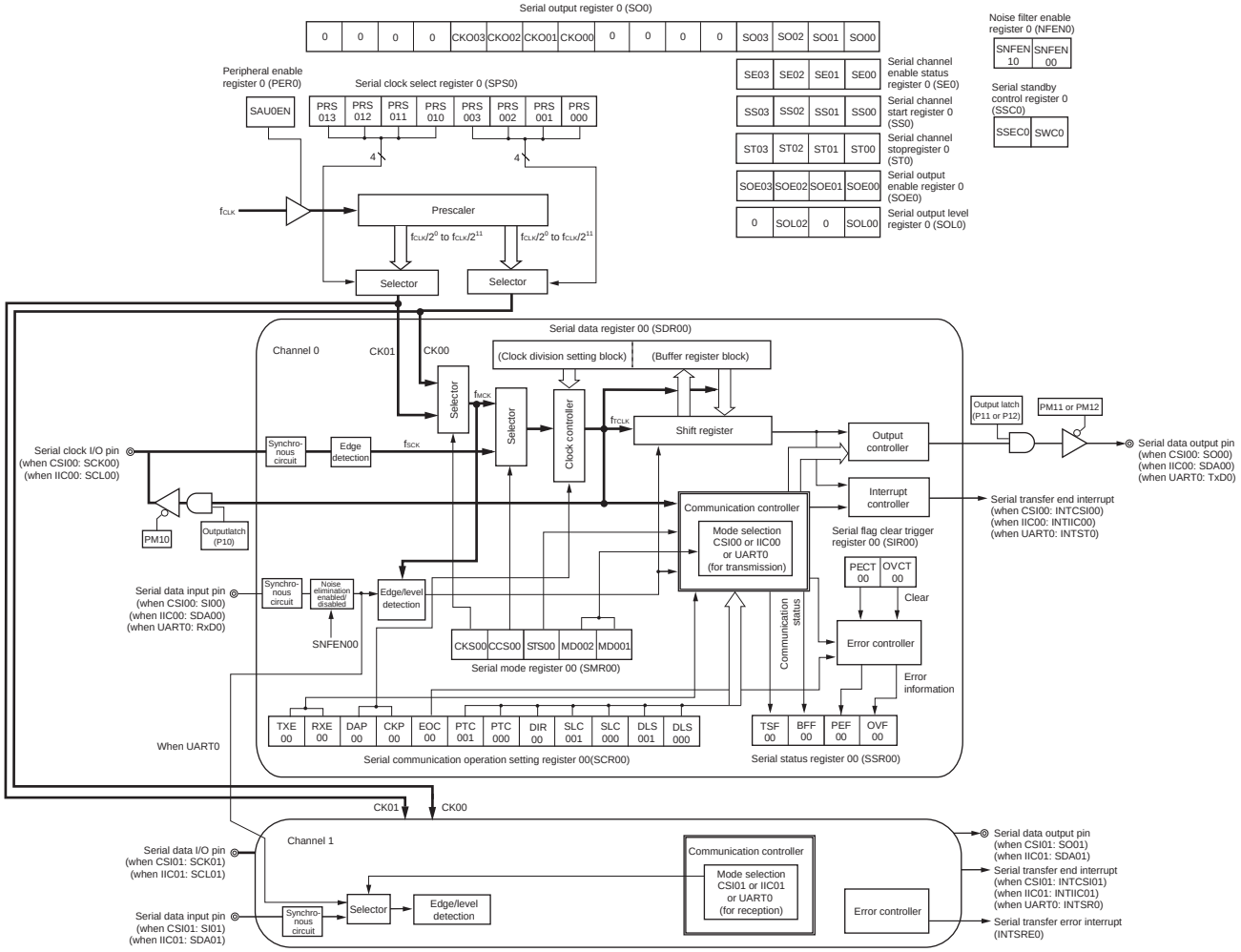
- During CSIp communication: SIOp (CSIp data register)
- During UARTq reception: RXDq (UART0 receive data register)
- During UARTq transmission: TXDq (UART0 transmit data register)
- During IICr communication: SIOr (IICr data register)

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 11, 20), q: UART number (q = 0 to 2), r: IIC number (r = 00, 01, 11, 20)

Downloaded from Arrow.com.

Figure 11-1 or 11-2 shows the block diagram of the serial array unit 0.

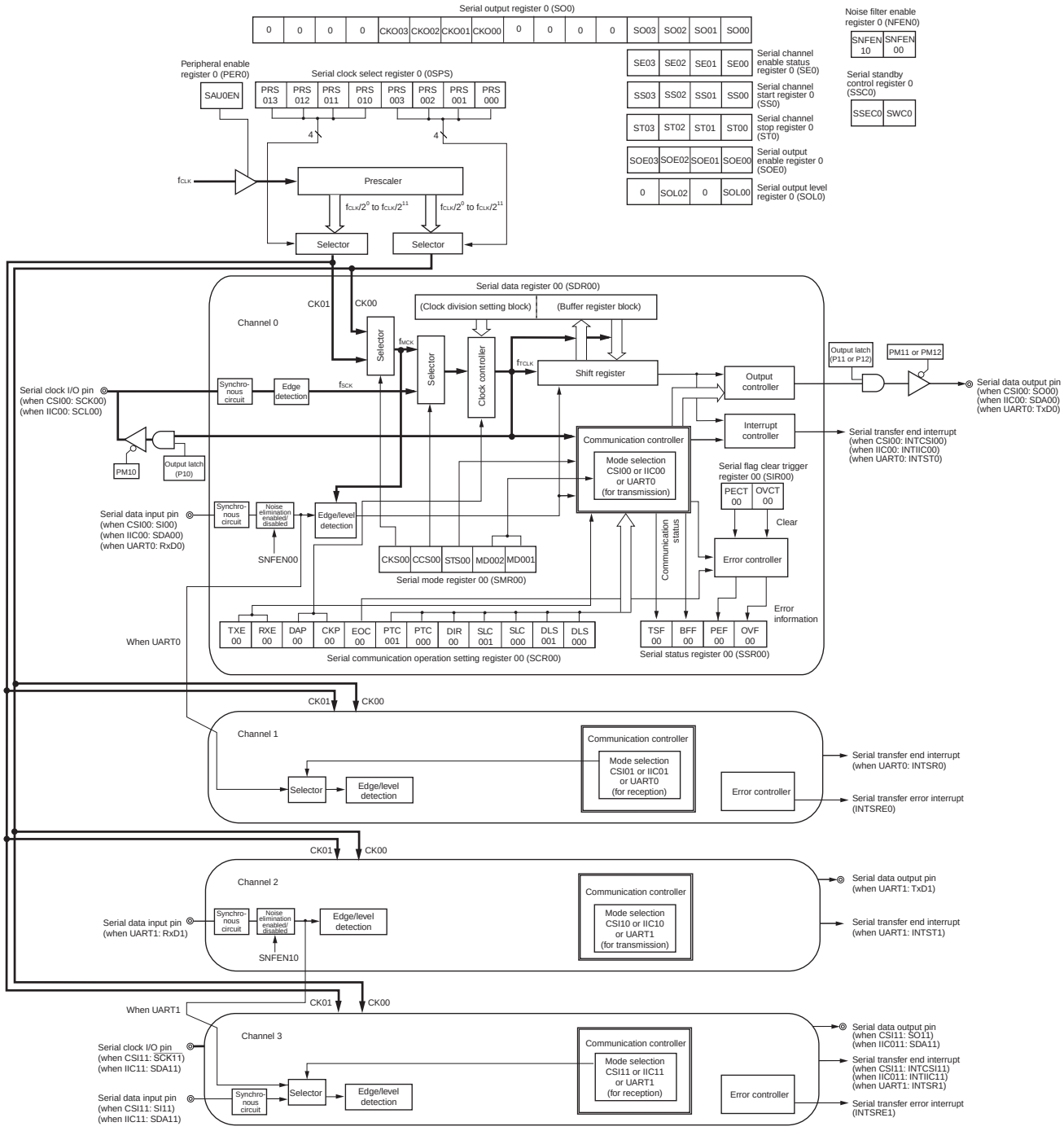
<R> **Figure 11-1. Block Diagram of Serial Array Unit 0 (20- or 24-pin products)**



Downloaded from Arrow.com.

<R>

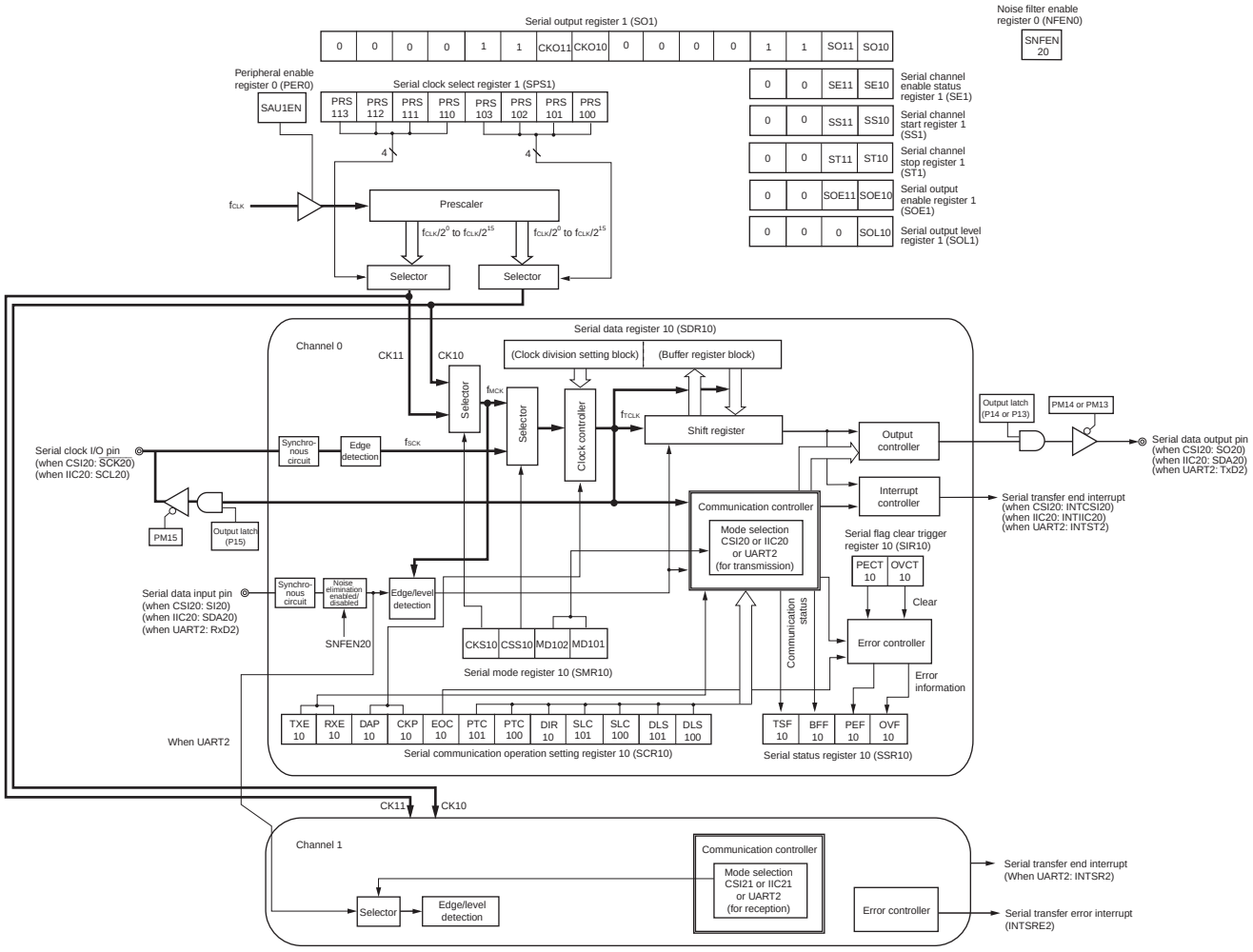
Figure 11-2. Block Diagram of Serial Array Unit 0 (30-pin products)



Downloaded from Arrow.com.

Figure 11-3 shows the block diagram of the serial array unit 0.

<R> **Figure 11-3. Block Diagram of Serial Array Unit 1 (30-pin products)**



The serial array unit 1 is available only in the 30-pin R5F102 products.

Downloaded from Arrow.com.

<R> 11.2.1 Shift register

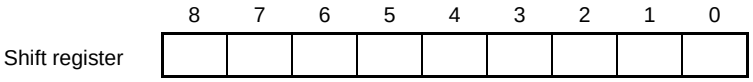
This is an 8-bit register that converts parallel data into serial data or vice versa.

In case of the UART communication of nine bits of data, nine bits (bits 0 to 8) are used[Note 1].

During reception, it converts data input to the serial pin into parallel data. When data is transmitted, the value set to this register is output as serial data from the serial output pin.

The shift register cannot be directly manipulated by program.

To read or write to the shift register, use the lower 8/9 bits of serial data register mn (SDRmn).



<R> 11.2.2 Lower 8/9 bits of the serial data register mn (SDRmn)

The SDRmn register is the transmit/receive data register (16 bits) of channel n. Bits 8 to 0 (lower 9 bits)[Note 1] or bits 7 to 0 (lower 8 bits) function as a transmit/receive buffer register, and bits 15 to 9 are used as a register that sets the division ratio of the operation clock (f_{MCK}).

When data is received, parallel data converted by the shift register is stored in the lower 8/9 bits. When data is to be transmitted, set transmit data to be transferred to the shift register to the lower 8/9 bits.

The length of data stored in the lower 8/9 bits of this register is as follows, depending on the setting of bits 0 and 1 (DLSmn0, DLS0m1) of serial communication operation setting register mn (SCRmn), regardless of the output sequence of the data.

- 7-bit data length (stored in bits 0 to 6 of SDRmn register)
- 8-bit data length (stored in bits 0 to 7 of SDRmn register)
- 9-bit data length (stored in bits 0 to 8 of SDR00 and SDR01 registers)[Note 1]

The SDRmn register can be read or written in 16-bit units.

The lower 8/9 bits of the SDRmn register can be read or written in 8-bit units as the following SFR, depending on the communication mode[Note 2].

- During CSIp communication: SIOp (CSIp data register)
- During UARTq reception: RXDq (UARTq receive data register)
- During UARTq transmission: TXDq (UARTq transmit data register)
- During IICr communication: SIOr (IICr data register)

Reset signal generation clears the SDRmn register to 0000H.

<R> **Notes 1.** Only UART0 can be specified for the 9-bit data length.

<R> **2.** Writing in 8-bit units is prohibited when the operation is stopped (SEmn = 0).

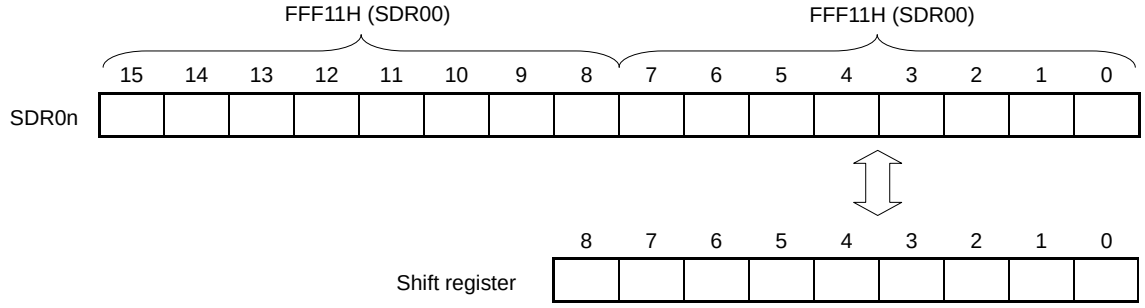
Remarks 1. After data is received, "0" is applied to some bits of bits 0 to 8 to make up the specified data length.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 11, 20), q: UART number (q = 0 to 2), r: IIC number (r = 00, 01, 11, 20)

Downloaded from Arrow.com.

Figure 11-4. Format of Serial Data Register mn (SDRmn) (mn = 00, 01)

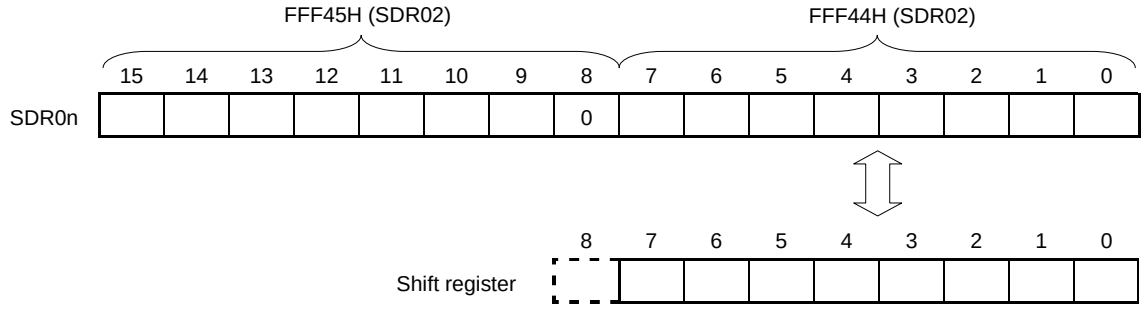
Address: FFF10H, FFF11H (SDR00), FFF12H, FFF13H (SDR01) After reset: 0000H R/W



Remark For the function of the higher 7 bits of the SDRmn register, see **11.3 Registers Controlling Serial Array Unit**.

Figure 11-5. Format of Serial Data Register mn (SDRmn) (mn = 02, 03, 10, 11)

Address: FFF44H, FFF45H (SDR02), FFF46H, FFF47H (SDR03) After reset: 0000H R/W
FFF48H, FFF49H (SDR10), FFF4AH, FFF4BH (SDR11)



Caution Be sure to clear bit 8 to "0".

Remark For the function of the higher 7 bits of the SDRmn register, see **11.3 Registers Controlling Serial Array Unit**.

Downloaded from Arrow.com.

11.3 Registers Controlling Serial Array Unit

Serial array unit is controlled by the following registers.

- Peripheral enable register 0 (PER0)
- Serial clock select register m (SPSm)
- Serial mode register mn (SMRmn)
- Serial communication operation setting register mn (SCRmn)
- Serial data register mn (SDRmm)
- Serial flag clear trigger register mn (SIRmn)
- Serial status register mn (SSRmn)
- Serial channel start register m (SSm)
- Serial channel stop register m (STm)
- Serial channel enable status register m (SEm)
- Serial output enable register m (SOEm)
- Serial output level register m (SOLm)
- Serial output register m (SOm)
- Serial standby control register m (SSCm)
- Noise filter enable register 0 (NFEN0)
- Port input mode registers 0, 1 (PIM0, PIM1)
- Port output mode registers 0, 1, 4, 5 (POM0, POM1, POM4, POM5)
- Port mode control registers 0, 1, 4 (PMC0, PMC1, PMC4)
- Port mode registers 0, 1, 3 to 6 (PM0, PM1, PM3 to PM6)
- Port registers 0, 1, 3 to 6 (P0, P1, P3 to P6)

Remark m: Unit number (m = 0, 1) n: Channel number (n = 0 to 3)

11.3.1 Peripheral enable register 0 (PER0)

PER0 is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

When serial array unit 0 is used, be sure to set bit 2 (SAU0EN) of this register to 1.

When serial array unit 1 is used, be sure to set bit 3 (SAU1EN) of this register to 1.

The PER0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears the PER0 register to 00H.

Downloaded from Arrow.com.

Figure 11-6. Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H After reset: 00H R/W

Symbol	<7>	6	<5>	<4>	<3>	<2>	1	<0>
PER0	TMKAEN	0	ADCEN	IICA0EN	SAU1EN[Note]	SAU0EN	0	TAU0EN

SAU1EN	Control of serial array unit 1 clock supply
0	Stops clock supply (fixed as "0" in 20- or 24-pin products). • SFR used by serial array unit 1 cannot be written. • Serial array unit 1 is in the reset status.
1	Enables clock supply. • SFR used by serial array unit 1 can be read/written.

SAU0EN	Control of serial array unit 0 clock supply
0	Stops clock supply. • SFR used by serial array unit 0 cannot be written. • Serial array unit 0 is in the reset status.
1	Enables clock supply. • SFR used by serial array unit 0 can be read/written.

<R> **Note** 30-pin products only.

<R> **Cautions 1. When setting serial array unit m, be sure to first set the following registers with the SAUmEN bit set to 1. If SAUmEN = 0, control registers of serial array unit m become default values and writing to them is ignored (except for the noise filter enable register 0 (NFEN0), port input mode registers 0, 1 (PIM0, PIM1), port output mode registers 0, 1, 4, 5 (POM0, POM1, POM4, POM5), port mode control registers 0, 1, 4 (PMC0, PMC1, PMC4), port mode registers 0, 1, 3 to 6 (PM0, PM1, PM3 to PM6), and port registers 0, 1, 3 to 6 (P0, P1, P3 to P6)).**

- **Serial clock select register m (SPSm)**
- **Serial mode register mn (SMRmn)**
- **Serial communication operation setting register mn (SCRmn)**
- **Serial data register mn (SDRmn)**
- **Serial flag clear trigger register mn (SIRmn)**
- **Serial status register mn (SSRmn)**
- **Serial channel start register m (SSm)**
- **Serial channel stop register m (STm)**
- **Serial channel enable status register m (SEm)**
- **Serial output enable register m (SOEm)**
- **Serial output level register m (SOLm)**
- **Serial output register m (SOm)**
- **Serial standby control register m (SSCm)**

2. Be sure to clear the following bits to 0.
20, 24-pin products: bits 1, 3, 6
30-pin products: bits 1, 6

Downloaded from Arrow.com.

11.3.2 Serial clock select register m (SPSm)

The SPSm register is a 16-bit register that is used to select two types of operation clocks (CKm0, CKm1) that are commonly supplied to each channel. CKm1 is selected by bits 7 to 4 of the SPSm register, and CKm0 is selected by bits 3 to 0.

Rewriting the SPSm register is prohibited when the register is in operation (when SEmn = 1).

The SPSm register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the SPSm register can be set with an 8-bit memory manipulation instruction with SPSmL.

Reset signal generation clears the SPSm register to 0000H.

Figure 11-7. Format of Serial Clock Select Register m (SPSm)

Address: F0126H, F0127H (SPS0), F0166H, F0167H (SPS1) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPSm	0	0	0	0	0	0	0	0	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00

<R>

PRS mk3	PRS mk2	PRS mk1	PRS mk0	Section of operation clock (CKmk)[Note]						
					f_{CLK} = 2 MHz	f_{CLK} = 4 MHz	f_{CLK} = 8 MHz	f_{CLK} = 16 MHz	f_{CLK} = 20 MHz	f_{CLK} = 24 MHz
0	0	0	0	f_{CLK}	2 MHz	4 MHz	8 MHz	16 MHz	20 MHz	24 MHz
0	0	0	1	$f_{CLK}/2$	1 MHz	2 MHz	4 MHz	8 MHz	10 MHz	12 MHz
0	0	1	0	$f_{CLK}/2^2$	500 kHz	1 MHz	2 MHz	4 MHz	5 MHz	6 MHz
0	0	1	1	$f_{CLK}/2^3$	250 kHz	500 kHz	1 MHz	2 MHz	2.5 MHz	3 MHz
0	1	0	0	$f_{CLK}/2^4$	125 kHz	250 kHz	500 kHz	1 MHz	1.25 MHz	1.5 MHz
0	1	0	1	$f_{CLK}/2^5$	62.5 kHz	125 kHz	250 kHz	500 kHz	625 kHz	750 kHz
0	1	1	0	$f_{CLK}/2^6$	31.3 kHz	62.5 kHz	125 kHz	250 kHz	313 kHz	375 kHz
0	1	1	1	$f_{CLK}/2^7$	15.6 kHz	31.3 kHz	62.5 kHz	125 kHz	156 kHz	188 kHz
1	0	0	0	$f_{CLK}/2^8$	7.81 kHz	15.6 kHz	31.3 kHz	62.5 kHz	78.1 kHz	93.8 kHz
1	0	0	1	$f_{CLK}/2^9$	3.91 kHz	7.81 kHz	15.6 kHz	31.3 kHz	39.1 kHz	46.9 kHz
1	0	1	0	$f_{CLK}/2^{10}$	1.95 kHz	3.91 kHz	7.81 kHz	15.6 kHz	19.5 kHz	23.4 kHz
1	0	1	1	$f_{CLK}/2^{11}$	977 Hz	1.95 kHz	3.91 kHz	7.81 kHz	9.77 kHz	11.7 kHz
1	1	0	0	$f_{CLK}/2^{12}$	488 Hz	977 Hz	1.95 kHz	3.91 kHz	4.88 kHz	5.86 kHz
1	1	0	1	$f_{CLK}/2^{13}$	244 Hz	488 Hz	977 Hz	1.95 kHz	2.44 kHz	2.93 kHz
1	1	1	0	$f_{CLK}/2^{14}$	122 Hz	244 Hz	488 Hz	977 Hz	1.22 kHz	1.46 kHz
1	1	1	1	$f_{CLK}/2^{15}$	61 Hz	122 Hz	244 Hz	488 Hz	610 Hz	732 Hz

<R> **Note** When changing the clock selected for f_{CLK} while the serial array unit (SAU) is operating (by changing the system clock control register (CKC) value), do so after having stopped (serial channel stop register m (STm) = 000FH) the operation of SAU.

Caution Be sure to clear bits 15 to 8 to "0".

Remarks 1. f_{CLK}: CPU/peripheral hardware clock frequency

2. m: Unit number (m = 0, 1) k = 0, 1

Downloaded from Arrow.com.

11.3.3 Serial mode register mn (SMRmn)

The SMRmn register is a register that sets an operation mode of channel n. It is also used to select an operation clock (f_{MCK}), specify whether the serial clock (f_{SCK}) may be input or not, set a start trigger, an operation mode (CSI, UART, or simplified I^2C), and an interrupt source. This register is also used to invert the level of the receive data only in the UART mode.

Rewriting the SMRmn register is prohibited when the register is in operation (when SEmn = 1). However, the MDmn0 bit can be rewritten during operation.

The SMRmn register can be set by a 16-bit memory manipulation instruction.

Reset signal generation sets the SMRmn register to 0020H.

Figure 11-8. Format of Serial Mode Register mn (SMRmn) (1/2)

Address: F0110H, F0111H (SMR00), F0116H, F0117H (SMR03) After reset: 0020H R/W
F0150H, F0151H (SMR10), F0152H, F0153H (SMR11)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKS mn	CCS mn	0	0	0	0	0	STS mn Note	0	SIS mn0 Note	1	0	0	MD mn2	MD mn1	MD mn0

CKSmn	Selection of operation clock (f_{MCK}) of channel n
0	Operation clock CK00 set by the SPSm register
1	Operation clock CK01 set by the SPSm register
Operation clock (f_{MCK}) is used by the edge detector. In addition, depending on the setting of the CCSmn bit and the higher 7 bits of the SDRmn register, a transfer clock (f_{TCLK}) is generated.	

CCSmn	Selection of transfer clock (f_{TCLK}) of channel n
0	Divided operation clock f_{MCK} specified by the CKSmn bit
1	Clock input f_{SCK} from the SCKp pin (slave transfer in CSI mode)
Transfer clock f_{TCLK} is used for the shift register, communication controller, output controller, interrupt controller, and error controller. When CCSmn = 0, the division ratio of operation clock (f_{MCK}) is set by the higher 7 bits of the SDRmn register.	

STSmn Note	Selection of start trigger source
0	Only software trigger is valid (selected for CSI, UART transmission, and simplified I^2C).
1	Valid edge of the RxDq pin (selected for UART reception)
Transfer is started when the above source is satisfied after 1 is set to the SSm register.	

Note Provided in the SMR01, SMR03, and SMR11 registers only.

<R> **Caution Be sure to clear bits 13 to 9, 7, 4, and 3 (or bits 13 to 6, 4, and 3 for the SMR00, SMR02, SMR10, or SMR12 register) to "0". Be sure to set bit 5 to "1".**

(**Remark** is listed on the next page.)

Downloaded from Arrow.com.

Figure 11-8. Format of Serial Mode Register mn (SMRmn) (2/2)

Address: F0110H, F0111H (SMR00) to F0116H, F0117H (SMR03), F0150H, F0151H (SMR10), F0152H, F0153H (SMR11) After reset: 0020H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKS mn	CCS mn	0	0	0	0	0	STS mn[Note]	0	SIS mn0[Note]	1	0	0	MD mn2	MD mn1	MD mn0

SISmn0[Note]	Controls inversion of level of receive data of channel n in UART mode
0	Falling edge is detected as the start bit. The input communication data is captured as is.
1	Rising edge is detected as the start bit. The input communication data is inverted and captured.

MDmn2	MDmn1	Setting of operation mode of channel n
0	0	CSI mode
0	1	UART mode
1	0	Simplified I^2C mode
1	1	Setting prohibited

MDmn0	Selection of interrupt source of channel n
0	Transfer end interrupt
1	Buffer empty interrupt (Occurs when data is transferred from the SDRmn register to the shift register.)
For successive transmission, the next transmit data is written by setting the MDmn0 bit to 1 when SDRmn data has run out.	

Note Provided in the SMR01, SMR03, and SMR11 registers only.

<R> **Caution Be sure to clear bits 13 to 9, 7, 4, and 3 (or bits 13 to 6, 4, and 3 for the SMR00, SMR02, SMR10, or SMR12 register) to "0". Be sure to set bit 5 to "1".**

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 11, 20), q: UART number (q = 0 to 2), r: IIC number (r = 00, 01, 11, 20)

11.3.4 Serial communication operation setting register mn (SCRmn)

The SCRmn register is a communication operation setting register of channel n. It is used to set a data transmission/reception mode, phase of data and clock, whether an error signal is to be masked or not, parity bit, start bit, stop bit, and data length.

Rewriting the SCRmn register is prohibited when the register is in operation (when SEmn = 1).

The SCRmn register can be set by a 16-bit memory manipulation instruction.

Reset signal generation sets the SCRmn register to 0087H.

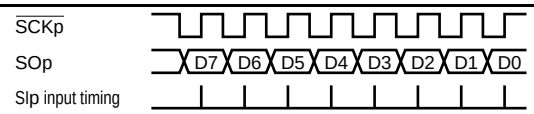
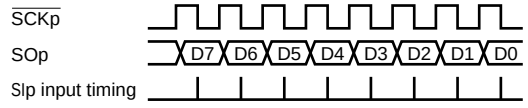
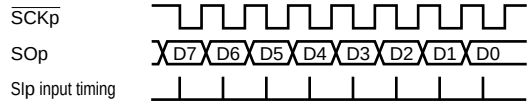
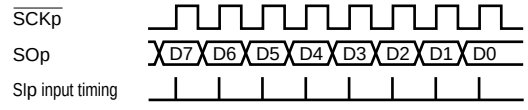
Downloaded from Arrow.com.

Figure 11-9. Format of Serial Communication Operation Setting Register mn (SCRmn) (1/2)

Address: F0118H, F0119H (SCR00) to F011EH, F011FH (SCR03) After reset: 0087H R/W
F0158H, F0159H (SCR10), F015AH, F015BH (SCR11)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn	RXEmn	DAPmn	CKPmn	0	EOCmn	PTCmn1	PTCmn0	DIRmn	0	SLCmn1 Note 1	SLCmn0	0	1	DLSmn1 Note 2	DLSmn0

TXEmn	RXEmn	Setting of operation mode of channel n
0	0	Disable communication.
0	1	Reception only
1	0	Transmission only
1	1	Transmission/reception

DAPmn	CKPmn	Selection of data and clock phase in CSI mode	Type
0	0		1
0	1		2
1	0		3
1	1		4
Be sure to set DAPmn, CKPmn = 0, 0 in the UART mode and simplified I^2C mode.			

EOCmn	Mask control of error interrupt signal (INTSREx (x = 0 to 3))
0	Disables generation of error interrupt INTSREx (INTSRx is generated).
1	Enables generation of error interrupt INTSREx (INTSRx is not generated if an error occurs).
Set EOCmn = 0 in the CSI mode, simplified I^2C mode, and during UART transmission Note 3.	

Notes 1. Provided in the SCR00, SCR02, and SCR10 registers only.
2. Provided in the SCR00 and SCR01 registers only. Others are fixed to 1.
3. If EOCmn is not cleared for CSImn, error interrupt INTSREn may be generated.

<R> **Caution Be sure to clear bits 3, 6, and 11 to "0" (also clear bit 5 of the SCR01, SCR03, or SCR11 register to 0). Be sure to set bit 2 to "1".**

(**Remark** is listed on the next page.)

Downloaded from Arrow.com.

Figure 11-9. Format of Serial Communication Operation Setting Register mn (SCRmn) (2/2)

Address: F0118H, F0119H (SCR00), F011EH, F011FH (SCR03) After reset: 0087H R/W
F0158H, F0159H (SCR10), F015AH, F015BH (SCR11)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn	RXEmn	DAPmn	CKPmn	0	EOCmn	PTCmn1	PTCmn0	DIRmn	0	SLCmn1 Note 1	SLCmn0	0	1	DLSmn1 Note 2	DLSmn0

PTCmn1	PTCmn0	Setting of parity bit in UART mode	
		Transmission	Reception
0	0	Does not output the parity bit.	Receives without parity
0	1	Outputs 0 parity Note 3.	No parity judgment
1	0	Outputs even parity.	Judged as even parity.
1	1	Outputs odd parity.	Judges as odd parity.
Be sure to set PTCmn1, PTCmn0 = 0, 0 in the CSI mode and simplified I^2C mode.			

DIRmn	Selection of data transfer sequence in CSI and UART modes
0	Inputs/outputs data with MSB first.
1	Inputs/outputs data with LSB first.
Be sure to clear DIRmn = 0 in the simplified I^2C mode.	

SLCmn1 Note 1	SLCmn0	Setting of stop bit in UART mode
0	0	No stop bit
0	1	Stop bit length = 1 bit
1	0	Stop bit length = 2 bits (mn = 00, 02, 10 only)
1	1	Setting prohibited
When the transfer end interrupt is selected, the interrupt is generated when all stop bits have been completely transferred. Set the stop bit length to 1 bit (SLCmn1, SLCmn0 = 0, 1) during UART reception and in the simplified I^2C mode. <R> Set no stop bit (SLCmn1, SLCmn0 = 0, 0) in the CSI mode. Set the stop bit length to 1 bit (SLCmn1, SLCmn0 = 0, 1) or 2 bits (SLCmn1, SLCmn0 = 1, 0) during UART transmission.		

DLSmn1 Note 2	DLSmn0	Setting of data length in CSI and UART modes
0	1	9-bit data length (stored in bits 0 to 8 of the SDR00 and SDR01 registers) (settable in UART0 mode only)
1	0	7-bit data length (stored in bits 0 to 6 of the SDRmn register)
1	1	8-bit data length (stored in bits 0 to 7 of the SDRmn register)
Other than above		Setting prohibited
Be sure to set DLSmn1, DLSmn0 = 1, 1 in the simplified I^2C mode.		

Notes 1. Provided in the SCR00, SCR02, and SCR10 registers only.
2. Provided in the SCR00 and SCR01 registers only. Others are fixed to 1.
3. 0 is always added regardless of the data contents.

<R> **Caution Be sure to clear bits 3, 6, and 11 to "0" (also clear bit 5 of the SCR01, SCR03, or SCR11 register to 0). Be sure to set bit 2 to "1".**

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 11, 20)

Downloaded from Arrow.com.

<R> 11.3.5 Serial data register mn (SDRmn)

The SDRmn register is the transmit/receive data register of channel n (16 bits). Bits 8 to 0 (lower 9 bits) of SDR00 and SDR01 or bits 7 to 0 (lower 8 bits) of SDR02, SDR03, SDR10, and SDR11 function as a transmit/receive buffer register, and bits 15 to 9 (higher 7 bits) are used as a register that sets the division ratio of the operation clock (f_{MCK}).

If the CCSmn bit of serial mode register mn (SMRmn) is cleared to 0, the operating clock divided by the division ratios specified by bits 15 to 9 (higher 7 bits) of the SDRmn register is used as the transfer clock.

<R> If the CCSmn bit of SMRmn is set to 1, set bits 15 to 9 (higher 7 bits) of SDR00, SDR01, SDR10, and SDR11 to 0000000B. The input clock f_{SCK} (slave transfer in CSI mode) from the SCKp pin is used as the transfer clock.

The lower 8/9 bits of the SDRmn register function as a transmit/receive buffer register. During reception, the parallel data converted by the shift register is stored in the lower 8/9 bits, and during transmission, the data to be transmitted to the shift register is set to the lower 8/9 bits.

The SDRmn register can be read or written in 16-bit units.

However, the higher 7 bits can be written or read only when the operation is stopped (SEmn = 0). During operation (SEmn = 1), a value is written only to the lower 9 bits of the SDRmn register. When the SDRmn register is read during operation, the higher 7 bits are always read as 0.

Reset signal generation clears the SDRmn register to 0000H.

Figure 11-10. Format of Serial Data Register mn (SDRmn)

Address: FFF10H, FFF11H (SDR00), FFF12H, FFF13H (SDR01) After reset: 0000H R/W

	FFF11H (SDR00)								FFF10H (SDR00)							
Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn																

Address: FFF44H, FFF45H (SDR02), FFF46H, FFF47H (SDR03) After reset: 0000H R/W
FFF48H, FFF49H (SDR10), FFF4AH, FFF4BH (SDR11)

	FFF45H (SDR02)								FFF44H (SDR02)							
Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn								0								

<R>

SDRmn[15:9]							Transfer clock setting by dividing the operating clock (f_{MCK})
0	0	0	0	0	0	0	f_{MCK}/2
0	0	0	0	0	0	1	f_{MCK}/4
0	0	0	0	0	1	0	f_{MCK}/6
0	0	0	0	0	1	1	f_{MCK}/8
•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•
1	1	1	1	1	1	0	f_{MCK}/254
1	1	1	1	1	1	1	f_{MCK}/256

(**Cautions** and **Remarks** are listed on the next page.)

Downloaded from Arrow.com.

Cautions 1. Be sure to clear bit 8 of SDR02, SDR03, SDR10, and SDR11 registers to 0.

2. Setting SDRmn[15:9] = (0000000B, 0000001B) is prohibited when UART is used.

3. Setting SDRmn[15:9] = 0000000B is prohibited when simplified I^2C is used. Set SDRmn[15:9] to 0000001B or greater.

4. Do not write 8-bit data to the lower 8 bits if operation is stopped (SEmn = 0). Otherwise, the higher 7 bits are cleared to 0.

Remarks 1. For the function of the lower 8/9 bits of the SDRmn register, see **11.2 Configuration of Serial Array Unit**.

2. m: Unit number (m = 0, 1) n: Channel number (n = 0 to 3)

Downloaded from Arrow.com.

11.3.6 Serial flag clear trigger register mn (SIRmn)

The SIRmn register is a trigger register that is used to clear each error flag of channel n.

When each bit (FECTmn, PECTmn, OVCTmn) of this register is set to 1, the corresponding bit (FEFmn, PEFmn, OVFmn) of serial status register mn (SSRmn) is cleared to 0. Because the SIRmn register is a trigger register, it is cleared immediately when the corresponding bit of the SSRmn register is cleared.

The SIRmn register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the SIRmn register can be set with an 8-bit memory manipulation instruction with SIRmnL.

Reset signal generation clears the SIRmn register to 0000H.

Figure 11-11. Format of Serial Flag Clear Trigger Register mn (SIRmn)

Address: F0108H, F0109H (SIR00), F010EH, F010FH (SIR03), F0148H, F0149H (SIR10), F014AH, F014BH (SIR11) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIRmn	0	0	0	0	0	0	0	0	0	0	0	0	0	FECTmn[Note]	PECTmn	OVCTmn

FECTmn[Note]	Clear trigger of framing error of channel n
0	Not cleared
1	Clears the FEFmn bit of the SSRmn register to 0.

PECTmn	Clear trigger of parity error flag of channel n
0	Not cleared
1	Clears the PEFmn bit of the SSRmn register to 0.

OVCTmn	Clear trigger of overrun error flag of channel n
0	Not cleared
1	Clears the OVFmn bit of the SSRmn register to 0.

Note Provided in the SIR01, SIR03, SIR11 registers only.

Caution Be sure to clear bits 15 to 3 (or bits 15 to 2 for the SIR00, SIR02, or SIR10 register) to "0".

<R> **Remarks 1.** m: Unit number (m = 0, 1) n: Channel number (n = 0 to 3)

2. When the SIRmn register is read, 0000H is always read.

Downloaded from Arrow.com.

11.3.7 Serial status register mn (SSRmn)

The SSRmn register indicates the communication status and error occurrence status of channel n. The errors indicated by this register are framing errors, parity errors, and overrun errors.

The SSRmn register can be read by a 16-bit memory manipulation instruction.

The lower 8 bits of the SSRmn register can be read with an 8-bit memory manipulation instruction as SSRmnL.

Reset signal generation clears the SSRmn register to 0000H.

Figure 11-12. Format of Serial Status Register mn (SSRmn) (1/2)

Address: F0100H, F0101H (SSR00) to F0106H, F0107H (SSR03) After reset: 0000H R
F0140H, F0141H (SSR10), F0142H, F0143H (SSR11)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSRmn	0	0	0	0	0	0	0	0	0	TSF mn	BFF mn	0	0	FEF mn[Note]	PEF mn	OVF mn

TSFmn	Communication status indication flag of channel n
0	Communication is stopped or suspended.
1	Communication is in progress.
<Clear conditions> • The STmn bit of the STm register is set to 1 (communication is stopped) or the SSmn bit of the SSm register is set to 1 (communication is suspended). • Communication ends. <Set condition> • Communication starts.	

BFFmn	Buffer register status indication flag of channel n
0	Valid data is not stored in the SDRmn register.
1	Valid data is stored in the SDRmn register.
<Clear conditions> • Transferring transmit data from the SDRmn register to the shift register ends during transmission. • Reading receive data from the SDRmn register ends during reception. • The STmn bit of the STm register is set to 1 (communication is stopped) or the SSmn bit of the SSm register is set to 1 (communication is enabled). <Set conditions> • Transmit data is written to the SDRmn register while the TXEmn bit of the SCRmn register is set to 1 (transmission or transmission and reception mode in each communication mode). • Receive data is stored in the SDRmn register while the RXEmn bit of the SCRmn register is set to 1 (reception or transmission and reception mode in each communication mode). • A reception error occurs.	

Note Provided in the SSR01, SSR03, SSR11 registers only.

<R> **Caution When the CSI is performing reception operations in the SNOOZE mode (SWC0 = 1), the BFF01 flag will not change.**

Remark m: Unit number (m = 0, 1) n: Channel number (n = 0 to 3)

Downloaded from Arrow.com.

Figure 11-12. Format of Serial Status Register mn (SSRmn) (2/2)

Address: F0100H, F0101H (SSR00) to F0106H, F0107H (SSR03) After reset: 0000H R
F0140H, F0141H (SSR10), F0142H, F0143H (SSR11)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSRmn	0	0	0	0	0	0	0	0	0	TSF mn	BFF mn	0	0	FEF mn[Note]	PEF mn	OVF mn

FEFmn[Note]	Framing error detection flag of channel n
0	No error occurs.
1	An error occurs (during UART reception).
<Clear condition> • 1 is written to the FECTmn bit of the SIRmn register. <Set condition> • A stop bit is not detected when UART reception ends.	

PEFmn	Parity / ACK error detection flag of channel n
0	No error occurs.
1	Parity error occurs (during UART reception) or ACK is not detected (during simplified I^2C transmission).
<Clear condition> • 1 is written to the PECTmn bit of the SIRmn register. <Set condition> • The parity of the transmit data and the parity bit do not match when UART reception ends (parity error). • No ACK signal is returned from the slave channel at the ACK reception timing during simplified I^2C transmission (ACK is not detected).	

OVFmn	Overrun error detection flag of channel n
0	No error occurs.
1	An error occurs
<Clear condition> • 1 is written to the OVCTmn bit of the SIRmn register. <Set condition> • Even though receive data is stored in the SDRmn register, that data is not read and transmit data or the next receive data is written while the RXEmn bit of the SCRmn register is set to 1 (reception or transmission and reception mode in each communication mode). • Transmit data is not ready for slave transmission or transmission and reception in CSI mode.	

Note Provided in the SSR01, SSR03, SSR11 registers only.

<R> **Cautions 1. If data is written to the SDRmn register when BFFmn = 1, the transmit/receive data stored in the register is discarded and an overrun error (OVEmn = 1) is detected.**

<R> **2. When the CSI is performing reception operations in the SNOOZE mode (SWC0 = 1), the OVF01 flag will not change.**

Remark m: Unit number (m = 0, 1) n: Channel number (n = 0 to 3)

Downloaded from Arrow.com.

11.3.8 Serial channel start register m (SSm)

The SSm register is a trigger register that is used to enable communication/count for each channel.

When 1 is written to a bit of this register (SSmn), the corresponding bit (SEmn) of serial channel enable status register m (SEm) is set to 1 (operation is enabled). Because the SSmn bit is a trigger bit, it is cleared immediately when SEmn = 1.

The SSm register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the SSm register can be set with a 1-bit or 8-bit memory manipulation instruction with SSmL.

Reset signal generation clears the SSm register to 0000H.

Figure 11-13. Format of Serial Channel Start Register m (SSm)

Address: F0122H, F0123H (SS0) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SS0	0	0	0	0	0	0	0	0	0	0	0	0	SS03 Note 1	SS02 Note 1	SS01	SS00

Address: F0162H, F0163H (SS1) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SS1 Note 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SS11	SS10

SSmn	Operation start trigger of channel n
0	No trigger operation
1	Sets the SEmn bit to 1 and enters the communication wait status Note2.

Notes 1. 30-pin product only.

2. If set the SSmn = 1 to during a communication operation, will wait status to stop the communication. At this time, holding status value of control register and shift register, SCKmn and SOmn pins, and FEFmn, PEFmn, OVFmn flags.

<R> **Cautions 1. Be sure to clear bits 15 to 2 of the SS0 register for 20- or 24-pin products, bits 15 to 4 of the SS0 register for 30-pin products, and bits 15 to 2 of the SS1 register to "0".**

2. For the UART reception, set the RXEmn bit of SCRmn register to 1, and then be sure to set SSmn to 1 after 4 or more f_{MCK} clocks have elapsed.

Remarks 1. m: Unit number (m = 0, 1) n: Channel number (n = 0 to 3)

2. When the SSm register is read, 0000H is always read.

Downloaded from Arrow.com.

11.3.9 Serial channel stop register m (STm)

The STm register is a trigger register that is used to enable stopping communication/count for each channel.

When 1 is written to a bit of this register (STmn), the corresponding bit (SEmn) of serial channel enable status register m (SEm) is cleared to 0 (operation is stopped). Because the STmn bit is a trigger bit, it is cleared immediately when SEmn = 0.

The STm register is set by a 16-bit memory manipulation instruction.

The lower 8 bits of the STm register can be set with a 1-bit or 8-bit memory manipulation instruction as STmL.

Reset signal generation clears the STm register to 0000H.

Figure 11-14. Format of Serial Channel Stop Register m (STm)

Address: F0124H, F0125H (ST0) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ST0	0	0	0	0	0	0	0	0	0	0	0	0	ST03 Note 1	ST02 Note 1	ST01	ST00

Address: F0164H, F0165H (ST1) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ST1 Note 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	ST11	ST10

STmn	Operation stop trigger of channel n
0	No trigger operation
1	Clears the SEmn bit to 0 and stops the communication operation Note2

Notes 1. 30-pin product only.

2. While holding the value of the control register and shift register, and the status of the, SCKmn, SOmn pins, FEFmn, PEFmn, OVFmn flag.

<R> **Caution Be sure to clear bits 15 to 2 of the ST0 register for 20- or 24-pin products, bits 15 to 4 of the ST0 register for 30-pin products, and bits 15 to 2 of the ST1 register to "0".**

Remarks 1. m: Unit number (m = 0, 1) n: Channel number (n = 0 to 3)

2. When the STm register is read, 0000H is always read.

Downloaded from Arrow.com.

11.3.10 Serial channel enable status register m (SEm)

The SEm register indicates whether the data transmission/reception operation of each channel is enabled or disabled.

When 1 is written to a bit of serial channel start register m (SSm), the corresponding bit of this register is set to 1. When 1 is written to a bit of serial channel stop register m (STm), the corresponding bit of this register is cleared to 0.

If the operation of channel n is enabled, the value of the CKOmn bit (serial clock output of channel n) of serial output register m (SOm) cannot be rewritten by software, and a value is output from the serial clock pin according to the communication operation.

If the operation of channel n is disabled, the value of the CKOmn bit of the SOm register can be set by software and its value is output from the serial clock pin. In this way, any waveform, such as that of a start condition/stop condition, can be created by software.

The SEm register can be read by a 16-bit memory manipulation instruction.

The lower 8 bits of the SEm register can be set with a 1-bit or 8-bit memory manipulation instruction as SEmL.

Reset signal generation clears the SEm register to 0000H.

Figure 11-15. Format of Serial Channel Enable Status Register m (SEm)

Address: F0120H, F0121H (SE0) After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SE0	0	0	0	0	0	0	0	0	0	0	0	0	SE03 Note	SE02 Note	SE01	SE00

Address: F0160H, F0161H (SE1) After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SE1 Note 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SE11	SE11

SEmn	Indication of operation enable/disable status of channel n
0	Operation is disabled (stopped)
1	Operation is enabled.

Note 30-pin product only.

Caution Be sure to clear bits 15 to 2 of the SE0 register for 20- or 24-pin products, bits 15 to 4 of the SE0 register for 30-pin products, and bits 15 to 2 of the SE1 register to "0".

Remark m: Unit number (m = 0, 1) n: Channel number (n = 0 to 3)

Downloaded from Arrow.com.

11.3.11 Serial output enable register m (SOEm)

The SOEm register is used to enable or disable output of the serial communication operation of each channel.

If serial output is enabled for channel n, the value of the SOmn bit of serial output register m (SOm) cannot be rewritten by software, and a value is output from the serial data output pin according to the communication operation.

If serial output is disabled for channel n, the SOmn bit value of the SOm register can be set by software, and its value is output from the serial data output pin. In this way, any waveform, such as that of a start condition/stop condition, can be created by software.

The SOEm register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the SOEm register can be set with a 1-bit or 8-bit memory manipulation instruction as SOEmL.

Reset signal generation clears the SOEm register to 0000H.

Figure 11-16. Format of Serial Output Enable Register m (SOEm)

Address: F012AH, F012BH(SOE0) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOE0	0	0	0	0	0	0	0	0	0	0	0	0	SOE03 Note 1	SOE02 Note 1	SOE01 Note 2	SOE00

Address: F016AH, F016BH(SOE1) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOE1 Note 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE10

SOEmn	Serial output enable/disable of channel n
0	Disables output by serial communication operation.
1	Enables output by serial communication operation.

Notes 1. 30-pin product only.

2. 20-, 24-pin product only.

<R> **Caution Be sure to clear bits 15 to 2 of the SOE register for 20- or 24-pin products, bits 15 to 4, and 1 of the SOE0 register for 30-pin products, and bits 15 to 1 of the SOE1 register to "0".**

Remark m: Unit number (m = 0, 1) n: Channel number (n = 0 to 3)

Downloaded from Arrow.com.

11.3.12 Serial output register m (SOm)

The SOm register is a buffer register for serial output of each channel.

The value of the SOmn bit of this register is output from the serial data output pin of channel n.

The value of the CKOmn bit of this register is output from the serial clock output pin of channel n.

The SOmn bit of this register can be rewritten by software only when serial output is disabled (SOEmn = 0). When serial output is enabled (SOEmn = 1), rewriting by software is ignored, and the value of the register can be changed only by a serial communication operation.

The CKOmn bit of this register can be rewritten by software only when the channel operation is stopped (SEmn = 0). While channel operation is enabled (SEmn = 1), rewriting by software is ignored, and the value of the CKOmn bit can be changed only by a serial communication operation.

<R> To use the pin for serial interface as a port function pin, set the corresponding CKOmn and SOmn bits to 1.

The SOm register can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears the SOm register to 0F0FH.

Figure 11-17. Format of Serial Output Register m (SOm)

Address: F0128H, F0129H(SO0) After reset: 0F0FH R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO0	0	0	0	0	CKO03	1	CKO01	CKO00	0	0	0	0	SO03 Note 1	SO02 Note 1	SO01 Note 2	SO00

Address: F0168H, F0169H(SO1) After reset: 0F0FH R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO1 Note 1	0	0	0	0	1	1	1	CKO10	0	0	0	0	1	1	1	SO10

CKOmn	Serial clock output of channel n
0	Serial clock output value is "0".
1	Serial clock output value is "1".

SOmn	Serial data output of channel n
0	Serial data output value is "0".
1	Serial data output value is "1".

Notes 1. 30-pin product only.

2. 20-, 24-pin product only.

<R> **Caution Be sure to clear bits 15 to 12, and 7 to 4 of the SOm register to "0".**
Be sure to set bits 10, and 3 to 2 of the SO0 register for 20- or 24-pin products, and bits 10 and 1 of the SO0 register for 30-pin products to "1".
Be sure to set bits 11 to 9, and 3 to 1 of the SO1 register to "1".

Remark m: Unit number (m = 0, 1) n: Channel number (n = 0 to 3)

Downloaded from Arrow.com.

11.3.13 Serial output level register m (SOLm)

The SOLm register is used to set inversion of the data output level of each channel.

This register can be set only in the UART mode. Be sure to set 0 to corresponding bit in the CSI mode and simplified I^2C mode.

Inverting channel n by using this register is reflected on pin output only when serial output is enabled (SOEmn = 1). When serial output is disabled (SOEmn = 0), the value of the SOmn bit is output as is.

Rewriting the SOLm register is prohibited during operation (SEmn = 1).

The SOLm register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the SOLm register can be set with an 8-bit memory manipulation instruction as SOLmL.

Reset signal generation clears the SOLm register to 0000H.

Figure 11-18. Format of Serial Output Level Register m (SOLm)

Address: F0134H, F0135H (SOL0) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOL0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOL02[Note]	0	SOL00

Address: F0174H, F0175H (SOL1) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOL1[Note]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOL10

SOLmn	Selects inversion of the level of the transmit data of channel n in UART mode
0	Communication data is output as is.
1	Communication data is inverted and output.

Note 30-pin product only.

<R> **Caution Be sure to clear bits 15 to 1 of the SOL0 register for 20- or 24-pin products, bits 15 to 3, and 1 of the SOL0 register for 30-pin products, and bits 15 to 1 of the SOL1 register to "0".**

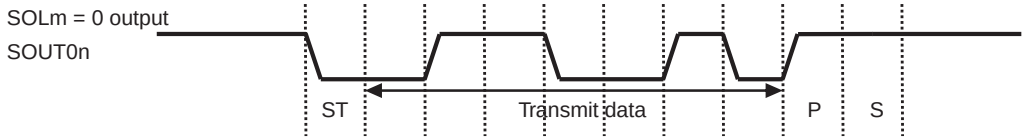
(**Remark** is listed on the next page.)

<R> Figure 11-19 shows examples in which the level of transmit data is reversed during UART transmission.

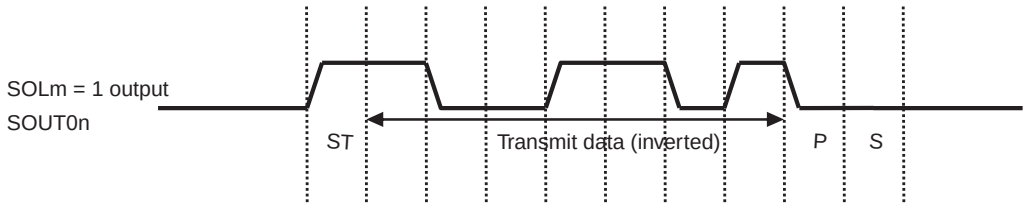
Downloaded from Arrow.com.

<R> **Figure 11-19. Examples of Reverse Transmit Data**

(a) Non-reverse Output (SOLmn = 0)



(b) Reverse Output (SOLmn = 1)



Remark m: Unit number (m = 0, 1) n: Channel number (n = 0, 2)

Downloaded from Arrow.com.

11.3.14 Serial standby control register 0 (SSC0)

The SSC0 register is used to control the startup of reception (the SNOOZE mode) while in the STOP mode when receiving CSI00 or UART0 serial data.

The SSC0 register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the SSC0 register can be set with an 8-bit memory manipulation instruction as SSC0L.

Reset signal generation clears the SSC0 register to 0000H.

<R> **Caution The maximum transfer rate in the SNOOZE mode is as follows.**

- **When using CSI00: Up to 1 Mbps**
- **When using UART0: 4800 bps only**

Figure 11-20. Format of Serial Standby Control Register 0 (SSC0)

Address: F0138H, F0139H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSCm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SSEC0	SWC0

<R>

SSEC0	Selection of whether to enable or disable the generation of communication error interrupts in the SNOOZE mode
0	Enable the generation of error interrupts (INTSRE0).
1	Stop the generation of error interrupts (INTSRE0).
• The SSEC0 bit can be set to 1 or 0 only when both the SWC0 and EOC01 bits are set to 1 during UART reception in the SNOOZE mode. In other cases, clear the SSEC0 bit to 0. • Setting SSEC0, SWC0 = 1, 0 is prohibited.	

SWC0	Setting the SNOOZE mode
0	Do not use the SNOOZE mode function.
1	Use the SNOOZE mode function.
• When there is a hardware trigger signal in the STOP mode, the STOP mode is exited, and A/D conversion is performed without operating the CPU (the SNOOZE mode). • The SNOOZE mode function can only be specified when the high-speed on-chip oscillator clock is selected for the CPU/peripheral hardware clock (f_{CLK}). If any other clock is selected, specifying this mode is prohibited. • Even when using SNOOZE mode, be sure to set the SWC0 bit to 0 in normal operation and change it to 1 just before shifting to STOP mode. Also, be sure to change the SWC0 bit to 0 after returning from STOP mode to normal operation.	

<R> **Figure 11-21. Interrupt in UART Reception Operation in SNOOZE Mode**

EOC01 Bit	SSEC0 Bit	Reception Ended Successfully	Reception Ended in an Error
0	0	INTSR0 is generated.	INTSR0 is generated.
0	1	INTSR0 is generated.	INTSR0 is generated.
1	0	INTSR0 is generated.	INTSRE0 is generated.
1	1	INTSR0 is generated.	No interrupt is generated.

Downloaded from Arrow.com.

11.3.15 Noise filter enable register 0 (NFEN0)

The NFEN0 register is used to set whether the noise filter can be used for the input signal from the serial data input pin to each channel.

Disable the noise filter of the pin used for CSI or simplified I^2C communication, by clearing the corresponding bit of this register to 0.

Enable the noise filter of the pin used for UART communication, by setting the corresponding bit of this register to 1.

<R> When the noise filter is enabled, after synchronization is performed with the operation clock (f_{MCK}) of the target channel, 2-clock match detection is performed. When the noise filter is disabled, only synchronization is performed with the operation clock (f_{MCK}) of the target channel.

The NFEN0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears the NFEN0 register to 00H.

Figure 11-22. Format of Noise Filter Enable Register 0 (NFEN0)

Address: F0070H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
NFEN0	0	0	0	SNFEN20Note	0	SNFEN10Note	0	SNFEN00

SNFEN20	Use of noise filter of RxD2 pin
0	Noise filter OFF
1	Noise filter ON
Set the SNFEN20 bit to 1 to use the RxD2 pin. Clear the SNFEN20 bit to 0 to use other than the RxD2 pin.	

SNFEN10	Use of noise filter of RxD1 pin
0	Noise filter OFF
1	Noise filter ON
Set the SNFEN10 bit to 1 to use the RxD1 pin. Clear the SNFEN10 bit to 0 to use the other than RxD1 pin.	

SNFEN00	Use of noise filter of RxD0 pin
0	Noise filter OFF
1	Noise filter ON
Set the SNFEN00 bit to 1 to use the RxD0 pin. Clear the SNFEN00 bit to 0 to use the other than RxD0 pin.	

Note 30-pin product only.

<R> **Caution Be sure to clear bits 7 to 1 for 20- or 24-pin products, and bits 7 to 5, 3, and 1 for 30-pin products to "0".**

Downloaded from Arrow.com.

<R> 11.3.16 Registers controlling port functions of serial input/output pins

Using the serial array unit requires setting of the registers that control the port functions multiplexed on the target channel (port mode register (PMxx), port register (Pxx), port input mode register (PIMxx), port output mode register (POMxx), and port mode control register (PMCxx)).
For details, see **4.3.1 Port mode registers (PMxx)**, **4.3.2 Port registers (Pxx)**, **4.3.4 Port input mode registers (PIMxx)**, **4.3.5 Port output mode registers (POMxx)**, and **4.3.6 Port mode control registers (PMCxx)**.

Specifically, using a port pin with a multiplexed serial data or serial clock output function (e.g. P10/ANI16/PCLBUZ0/SCK00/SCL00) for serial data or serial clock output, requires setting the corresponding bits in the port mode control register (PMCxx) and port mode register (PMxx) to 0, and the corresponding bits in the port register (Pxx) to 1.
When using the port pin in N-ch open-drain output (V_{DD} tolerance) mode, set the corresponding bit in the port output mode register (POMxx) to 1. When connecting an external device operating on a different potential (1.8 V, 2.5 V or 3 V), see **4.4.4 Handling different potentials (1.8 V, 2.5 V, and 3 V) by using I/O buffers**.

Example: When P10/ANI16/PCLBUZ0/SCK00/SCL00 for 20 or 24-pin products is to be used for serial clock output:
Set the PMC10 bit of port mode control register 1 to 0.
Set the PM10 bit of port mode register 1 to 0.
Set the P10 bit of port register 1 to 1.

Specifically, using a port pin with a multiplexed serial data or serial clock input function (e.g. P10/ANI16/PCLBUZ0/SCK00/ SCL00) for serial data or serial clock input, requires setting the corresponding bit in the port mode register (PMxx) to 1, and the corresponding bit in the port mode control register (PMCxx) to 0. In this case, the corresponding bit in the port register (Pxx) can be set to 0 or 1.
When the TTL input buffer is selected, set the corresponding bit in the port input mode register (PIMxx) to 1. When connecting an external device operating on a different potential (1.8 V, 2.5 V or 3 V), see **4.4.4 Handling different potentials (1.8 V, 2.5 V, and 3 V) by using I/O buffers**.

Example: When P10/ANI16/PCLBUZ0/SCK00/SCL00 for 20 or 24-pin products is to be used for serial clock input:
Set the PMC10 bit of port mode control register 1 to 0.
Set the PM10 bit of port mode register 1 to 0.
Set the P10 bit of port register 1 to 1.

Downloaded from Arrow.com.

11.4 Operation Stop Mode

Each serial interface of serial array unit has the operation stop mode.

In this mode, serial communication cannot be executed, thus reducing the power consumption.

In addition, the pin for serial interface can be used as port function pins in this mode.

11.4.1 Stopping the operation by units

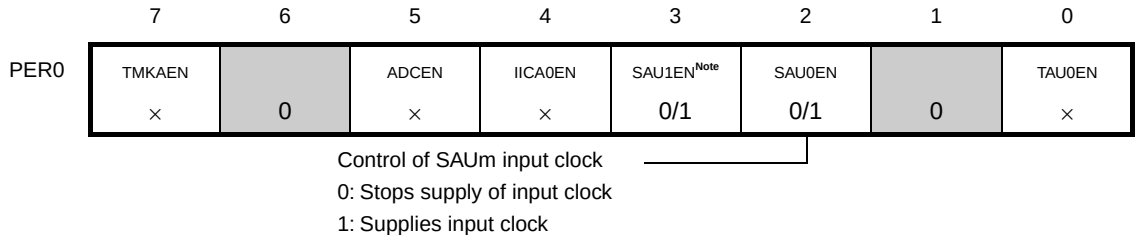
The stopping of the operation by units is set by using peripheral enable register 0 (PER0).

The PER0 register is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

To stop the operation of serial array unit 0, set bit 2 (SAU0EN) to 0. To stop the operation of serial array unit 1, set bit 3 (SAU1EN) to 0.

Figure 11-23. Peripheral Enable Register 0 (PER0) Setting When Stopping Operation by Units

(a) Peripheral enable register 0 (PER0) ... Set only the bit of SAUm to be stopped to 0.



Note Provided only in 30-pin products.

Cautions 1. If SAUmEN = 0, writing to a control register of serial array unit m is ignored, and, even if the register is read, only the default value is read
Note that this does not apply to the following registers.
- **Noise filter enable register 0 (NFEN0)**
- **Port input mode register 0, 1 (PIM0, PIM1)**
- **Port output mode registers 0, 1, 4, 5 (POM0, POM1, POM4, POM5)**
- **Port mode registers 0, 1, 3 to 6 (PM0, PM1, PM3 to PM6)**
- **Port registers 0, 1, 3 to 6 (P0, P1, P3 to P6)**
- **Port mode control registers 0, 1, 4 (PMC0, PMC1, PMC4)**

2. Be sure to clear the following bits to 0.
20, 24-pin products: bits 1, 3, 6
30-pin products: bits 1, 6

Remark ☐: Setting disabled (fixed by hardware)
×: Bits not used with serial array units (depending on the settings of other peripheral functions)
0/1: Set to 0 or 1 depending on the usage of the user.

Downloaded from Arrow.com.

11.4.2 Stopping the operation by channels

The stopping of the operation by channels is set using each of the following registers.

Figure 11-24. Each Register Setting When Stopping Operation by Channels

(a) Serial channel stop register m (STm) ... This register is a trigger register that is used to enable stopping communication/count by each channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STm	0	0	0	0	0	0	0	0	0	0	0	0	ST03 Note 1 0/1	ST02 Note 1 0/1	STm1 0/1	STm0 0/1

1: Clears the SEmn bit to 0 and stops the communication operation

* Because the STmn bit is a trigger bit, it is cleared immediately when SEmn = 0.

(b) Serial Channel Enable Status Register m (SEm) ... This register indicates whether data transmission/reception operation of each channel is enabled or stopped.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEm	0	0	0	0	0	0	0	0	0	0	0	0	SE03 Note 1 0/1	SE02 Note 1 0/1	SEm1 0/1	SEm0 0/1

0: Operation stops

* The SEm register is a read-only status register. Operation is stopped by using the STm register. For a channel whose operation is disabled, the value of the CKOmn bit of the SOm register can be set by software.

(c) Serial output enable register m (SOEm) ... This register is a register that is used to enable or stop output of the serial communication operation of each channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	SOE03 Note 1 0/1	SOE02 Note 1 0/1	SOE01 Note 2 0/1	SOEm0 0/1

0: Stops output by serial communication operation

* For channel n whose serial output is stopped, the SOmn bit value of the SOm register can be set by software.

(d) Serial output register m (SOm) ...This register is a buffer register for serial output of each channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	1	1	CKOm1 0/1	CKOm0 0/1	0	0	0	0	SO03 Note 1 0/1	SO02 Note 1 0/1	SO01 Note 2 0/1	SOm0 0/1

1: Serial clock output value is "1" 1: Serial data output value is "1"

* When using pins corresponding to each channel as port function pins, set the corresponding CKOmn, SOmn bits to "1".

Notes 1. Provided in the serial array unit 0 of 30-pin products only.

2. 20-, 24-pin products only.

Remarks 1. m: Unit number (m = 0, 1) n: Channel number (n = 0 to 3)

2. ☐ : Setting disabled (fixed by hardware), 0/1: Set to 0 or 1 depending on the usage of the user

Downloaded from Arrow.com.

11.5 Operation of 3-Wire Serial I/O (CSI00, CSI01, CSI11, CSI20) Communication

This is a clocked communication function that uses three lines: serial clock (SCK) and serial data (SI and SO) lines.

[Data transmission/reception]

- Data length of 7 or 8 bits
- Phase control of transmit/receive data
- MSB/LSB first selectable

[Clock control]

- Master/slave selection
- Phase control of I/O clock
- Setting of transfer period by prescaler and internal counter of each channel

<R>
- Maximum transfer rate**Note**

  During master communication: Max. f_{CLK}/2 (CSI00 only)
  Max. f_{CLK}/4

  During slave communication: Max. f_{MCK}/6

[Interrupt function]

- Transfer end interrupt/buffer empty interrupt

[Error detection flag]

- Overrun error

In addition, CSI00 (channel 0 of unit 0) supports the SNOOZE mode. When SCK00 pin input is detected while in the STOP mode, the SNOOZE mode makes data reception that does not require the CPU possible.

<R> **Note** Use the clocks within a range satisfying the SCK cycle time (t_{KCY}) characteristics. For details, see **CHAPTER 28 ELECTRICAL SPECIFICATIONS (A, D: T_A = -40 to +85°C)** or **CHAPTER 29 ELECTRICAL SPECIFICATIONS (G: T_A = -40 to +105°C)**.

The channels supporting 3-wire serial I/O (CSI00, CSI01, CSI11, CSI20) are channels 0, 1, 3 of SAU0 and channel 0 of SAU1.

20- or 24-pin products

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I^2C
0	0	CSI00	UART0	IIC00**Note**
	1	CSI01**Note**		IIC01**Note**

30-pin products

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I^2C
0	0	CSI00	UART0	IIC00
	1	–		–
	2	–	UART1**Note**	–
	3	CSI11**Note**		IIC11**Note**
1	0	CSI20**Note**	UART2**Note**	IIC20**Note**
	1	–		–

Note Provided in the R5F102 products only.

Downloaded from Arrow.com.

3-wire serial I/O (CSI00, CSI01, CIS10, CIS20) performs the following seven types of communication operations.

- Master transmission (See **11.5.1**.)
- Master reception (See **11.5.2**.)
- Master transmission/reception (See **11.5.3**.)
- Slave transmission (See **11.5.4**.)
- Slave reception (See **11.5.5**.)
- Slave transmission/reception (See **11.5.6**.)
- SNOOZE mode function (for CSI00 only) (See **11.5.7**.)

11.5.1 Master transmission

<R> Master transmission is that the RL78 microcontroller outputs a transfer clock and transmits data to another device.

3-Wire Serial I/O	CSI00	CSI01	CSI11	CSI20
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1
Pins used	SCK00, SO00	SCK01, SO01	SCK11, SO11	SCK20, SO20
Interrupt	INTCSI00	INTCSI01	INTCSI11	INTCSI20
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.			
Error detection flag	None			
Transfer data length	7 or 8 bits			
<R> Transfer rate**Note**	Max. $f_{CLK}/2$ [Hz] (CSI00 only), $f_{CLK}/4$ [Hz] Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [Hz] f_{CLK}: System clock frequency			
Data phase	Selectable by the DAPmn bit of the SCRmn register • DAPmn = 0: Data output starts at the start of the operation of the serial clock. • DAPmn = 1: Data output starts half a clock before the start of the serial clock operation.			
Clock phase	Selectable by the CKPmn bit of the SCR0n register • CKPmn = 0: Non-reverse (data output at the falling edge and data input at the rising edge of SCK) • CKPmn = 1: Reverse (data output at the rising edge and data input at the falling edge of SCK)			
Data direction	MSB or LSB first			

Note Use this operation within a range that satisfies the conditions above and the peripheral function characteristics in the electrical specifications (see **CHAPTER 28 ELECTRICAL SPECIFICATIONS (A, D: T_A = -40 to +85°C)** or **CHAPTER 29 ELECTRICAL SPECIFICATIONS (G: T_A = -40 to +105°C)**).

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00, 01, 03, 10

<R>

Downloaded from Arrow.com.

(1) Register setting

Figure 11-25. Example of Contents of Registers for Master Transmission of 3-Wire Serial I/O (CSI00, CSI01, CSI11, CSI20)

(a) Serial mode register mn (SMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn	CCSmn						STSmn		SISmn0				MDmn2	MDmn1	MDmn0
	0/1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0/1

Operation clock (f_{MCK}) of channel n
0: Prescaler output clock CKm0 set by the SPSm register
1: Prescaler output clock CKm1 set by the SPSm register

Interrupt source of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(b) Serial communication operation setting register mn (SCRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn	RXEmn	DAPmn	CKPmn		EOCmn	PTCmn1	PTCmn0	DIRmn		SLCmn1	SLCmn0			DLSmn1	DLSmn0
	1	0	0/1	0/1	0	0	0	0	0/1	0	0	0	0	1	1	0/1

Selection of the data and clock phase (For details about the setting, see **11.3 Registers Controlling Serial Array Unit**.)

Selection of data transfer sequence
0: Inputs/outputs data with MSB first
1: Inputs/outputs data with LSB first.

Setting of data length
0: 7-bit data length
1: 8-bit data length

(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Baud rate setting (Operation clock (f_{MCK}) division setting)								0	Transmit data (Transmit data setting)						

SIOp

(d) Serial output register m (SOm) ... Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm							CKOm1	CKOm0					SO03 Note 1	SO02 Note 1	SO01 Note 2	SOm0
	0	0	0	0	1	1	0/1	0/1	0	0	0	0	0/1	×	0/1	0/1

Communication starts when these bits are 1 if the clock phase is non-inversion (the CKPmn bit of the SCRmn = 0). If the clock phase is inverted (CKPmn = 1), communication starts when these bits are 0.

(e) Serial output enable register m (SOEm) ... Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm													SOE03 Note 1	SOE02 Note 1	SOE01 Note 2	SOEm0
	0	0	0	0	0	0	0	0	0	0	0	0	0/1	×	0/1	0/1

(f) Serial channel start register m (SSm) ... Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm													SS03 Note 1	SS02 Note 1	SS01	SSm0
	0	0	0	0	0	0	0	0	0	0	0	0	0/1	×	0/1	0/1

Notes 1. Provided in the serial array unit 0 of 30-pin products.

2. 20-, 24-pin products only.

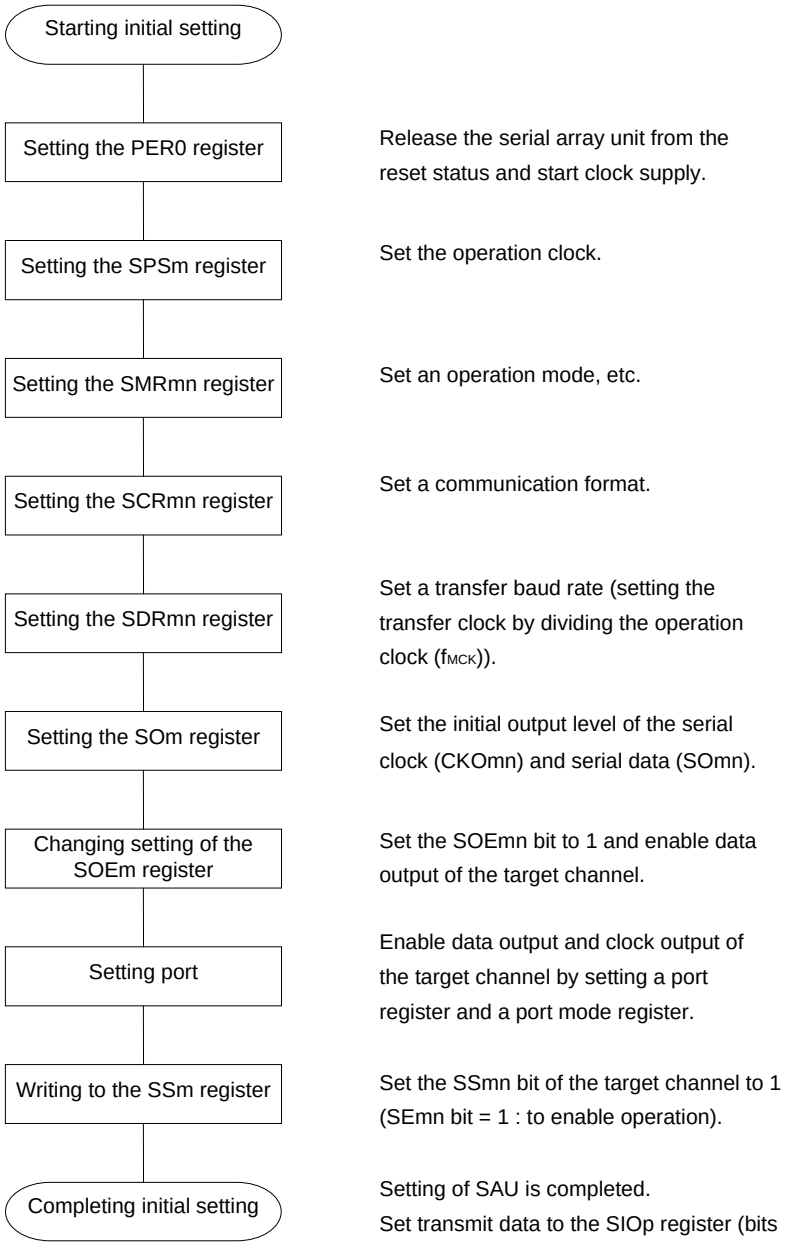
Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00, 01, 03, 10, p: CSI number (p = 00, 01, 11, 20)

2. ☐: Setting is fixed in the CSI master transmission mode, ■: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user.

Downloaded from Arrow.com.

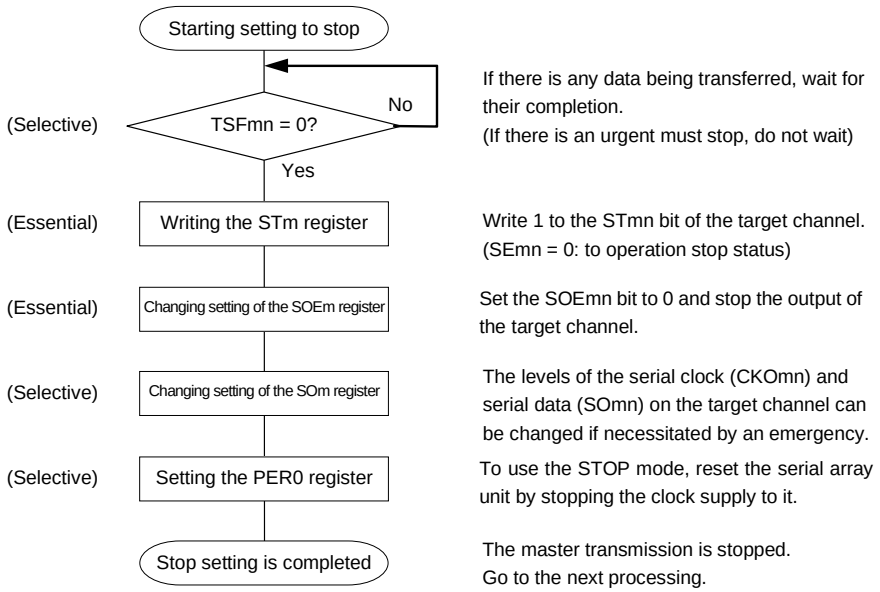
(2) Operation procedure

Figure 11-26. Initial Setting Procedure for Master Transmission



Downloaded from Arrow.com.

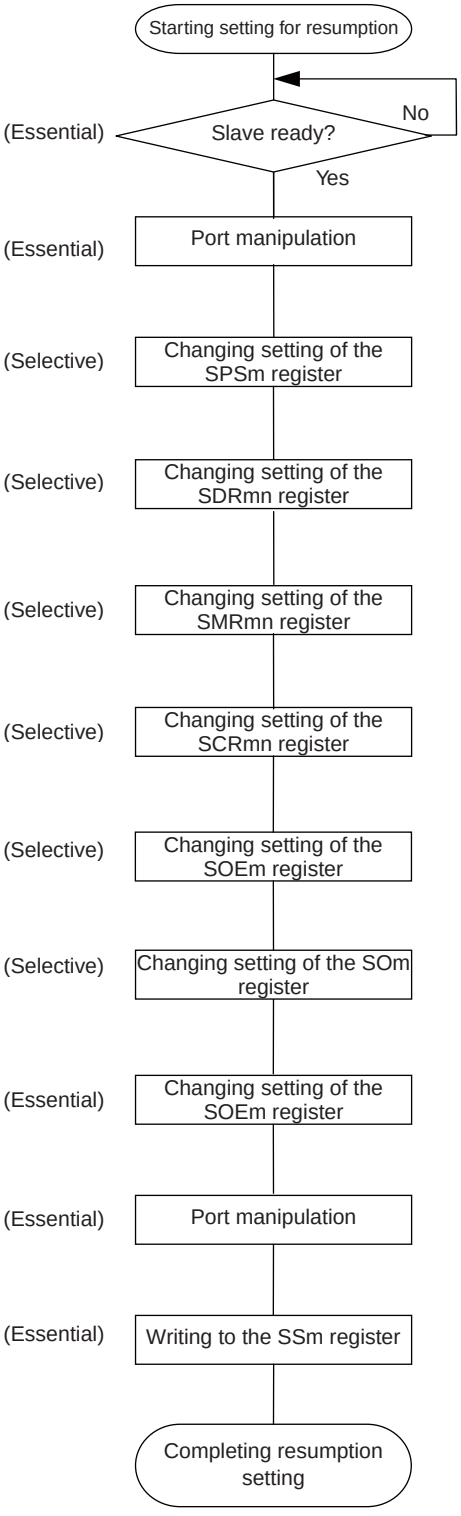
Figure 11-27. Procedure for Stopping Master Transmission



Downloaded from Arrow.com.

Figure 11-28. Procedure for Resuming Master Transmission

<R>



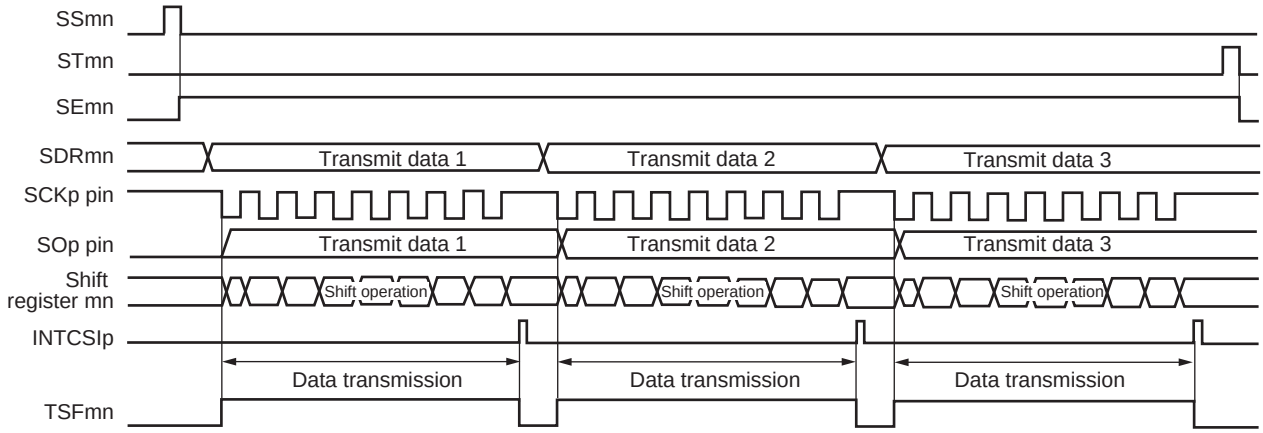
Remark If PER0 is rewritten while stopping the master transmission and the clock supply is stopped, wait until the transmission target (slave) stops or transmission finishes, and then perform initialization instead of restarting the transmission.

Downloaded from Arrow.com.

(3) Processing flow (in single-transmission mode)

Figure 11-29. Timing Chart of Master Transmission (in Single-Transmission Mode)
(Type 1: DAPmn = 0, CKPmn = 0)

<R>

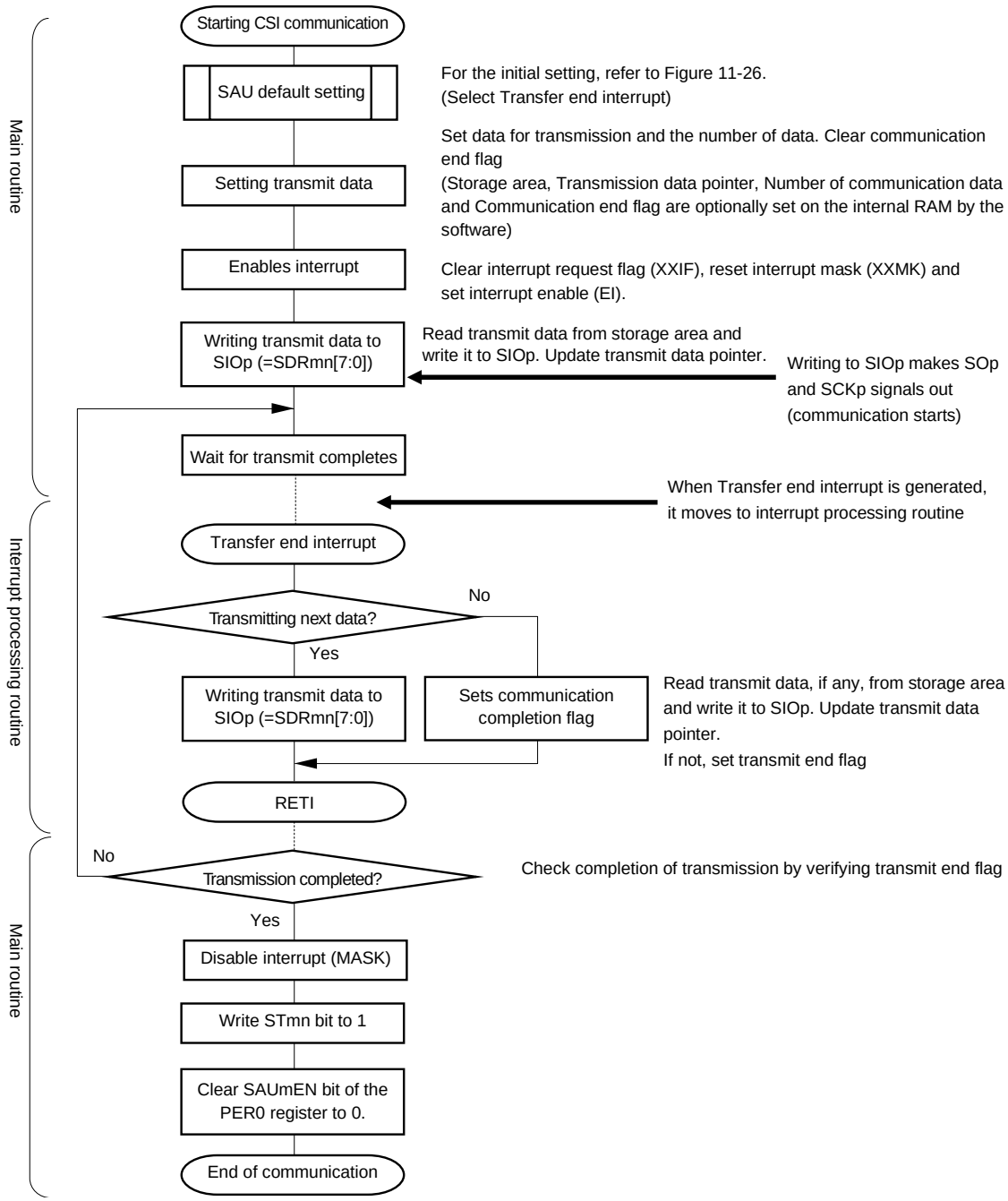


Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00, 01, 03, 10, p: CSI number (p = 00, 01, 11, 20)

Downloaded from Arrow.com.

Figure 11-30. Flowchart of Master Transmission (in Single-Transmission Mode)

<R>

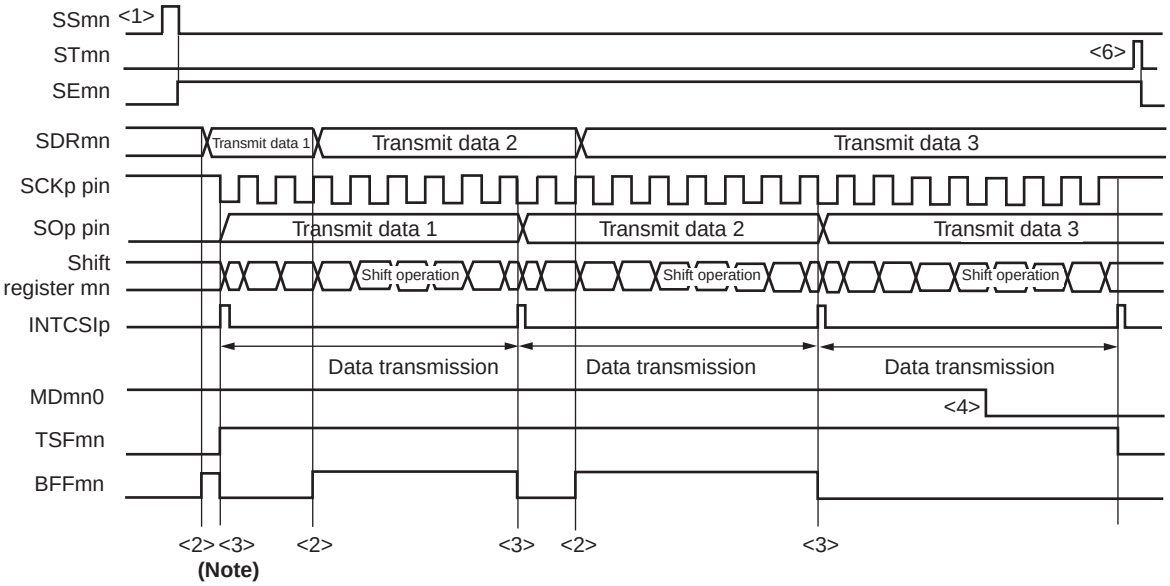


Downloaded from Arrow.com.

(4) Processing flow (in continuous transmission mode)

Figure 11-31. Timing Chart of Master Transmission (in Continuous Transmission Mode)
(Type 1: DAPmn = 0, CKPmn = 0)

<R>



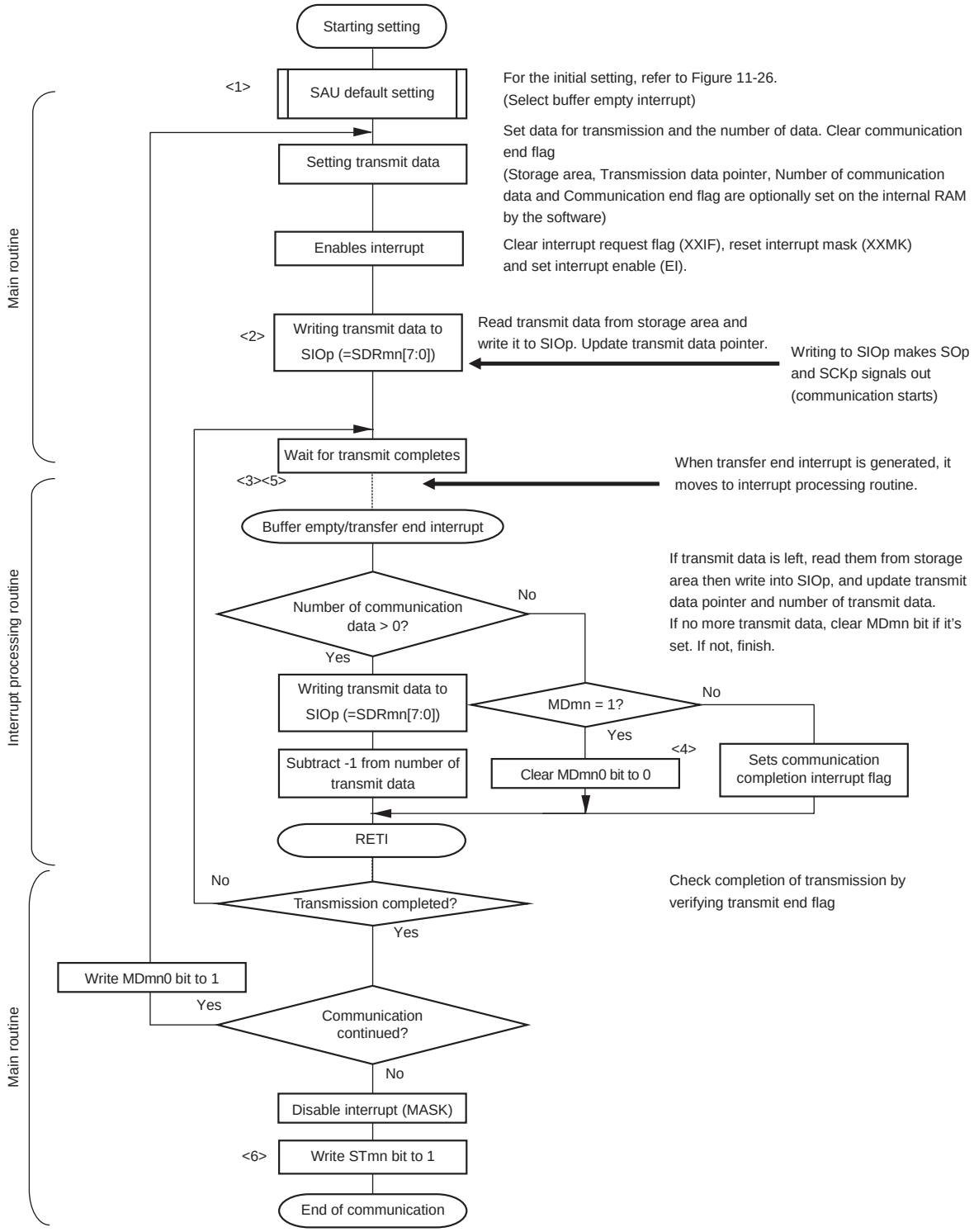
Note If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

Caution The MDmn0 bit of serial mode register mn (SMRmn) can be rewritten even during operation. However, rewrite it before transfer of the last bit is started, so that it will be rewritten before the transfer end interrupt of the last transmit data.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 11, 20), mn = 00, 01, 03, 10

Downloaded from Arrow.com.

<R> **Figure 11-32. Flowchart of Master Transmission (in Continuous Transmission Mode)**



Remark <1> to <6> in the figure correspond to <1> to <6> in **Figure 11-31 Timing Chart of Master Transmission (in Continuous Transmission Mode)**.

Downloaded from Arrow.com.

11.5.2 Master reception

<R> Master reception is that the RL78 microcontroller outputs a transfer clock and receives data from other device.

3-Wire Serial I/O	CSI00	CSI01	CSI11	CSI20
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1
Pins used	SCK00, SI00	SCK01, SI01	SCK11, SI11	SCK20, SI20
Interrupt	INTCSI00	INTCSI01	INTCSI11	INTCSI20
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.			
Error detection flag	Overrun error detection flag (OVFmn) only			
Transfer data length	7 or 8 bits			
<R> Transfer rate**Note**	Max. $f_{CLK}/2$ [Hz] (CSI00 only), $f_{CLK}/4$ [Hz] Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [Hz] f_{CLK}: System clock frequency			
Data phase	Selectable by the DAPmn bit of the SCRmn register • DAPmn = 0: Data input starts at the start of the operation of the serial clock. • DAPmn = 1: Data input starts half a clock before the start of the serial clock operation.			
Clock phase	Selectable by the CKPmn bit of the SCRmn register • CKPmn = 0: Non-inversion • CKPmn = 1: Inverted			
Data direction	MSB or LSB first			

Note Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 28 ELECTRICAL SPECIFICATIONS (A, D: T_A = -40 to +85°C)** or **CHAPTER 29 ELECTRICAL SPECIFICATIONS (G: T_A = -40 to +105°C)**).

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 13

<R>

Downloaded from Arrow.com.

(1) Register setting

Figure 11-33. Example of Contents of Registers for Master Reception of 3-Wire Serial I/O (CSI00, CSI01, CSI11, CSI20)

(a) Serial mode register mn (SMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn	CCSmn	0	0	0	0	0	STSmn	0	SISmn0	1	0	0	MDmn2	MDmn1	MDmn0
	0/1	0						0		0				0	0	0/1

Operation clock (f_{MCK}) of channel n
0: Prescaler output clock CKm0 set by the SPSm register
1: Prescaler output clock CKm1 set by the SPSm register

Interrupt source of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(b) Serial communication operation setting register mn (SCRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn	RXEmn	DAPmn	CKPmn		EOCmn	PTCmn1	PTCmn0	DIRmn		SLCmn1	SLCmn0			DLSmn1	DLSmn0
	0	1	0/1	0/1	0	0	0	0	0/1	0	0	0	0	1	1	0/1

Selection of the data and clock phase (For details about the setting, see **11.3 Registers Controlling Serial Array Unit**.)

Selection of data transfer sequence
0: Inputs/outputs data with MSB first
1: Inputs/outputs data with LSB first.

Setting of data length
0: 7-bit data length
1: 8-bit data length

(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Baud rate setting (Operation clock (f_{MCK}) division setting)							0	Receive data (Write FFH as dummy data.)							

SIOp

(d) Serial output register m (SOm) ... Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	1	1	CKOm1	CKOm0	0	0	0	0	SO03 Note 1	SO02 Note 1	SO01 Note 2	SOm0
							0/1	0/1					×	×	×	×

Communication starts when these bits are 1 if the clock phase is non-inversion (the CKPmn bit of the SCRmn = 0). If the clock phase is inverted (CKPmn = 1), communication starts when these bits are 0.

(e) Serial output enable register m (SOEm) ... The register that not used in this mode.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	SOE03 Note 1	SOE02 Note 1	SOE01 Note 2	SOEm0
													×	×	×	×

(f) Serial channel start register m (SSm) ... Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SS03 Note 1	SS02 Note 1	SSm1	SSm0
													0/1	×	0/1	0/1

Notes 1. Provided in the serial array unit 0 of 30-pin products.

2. 20-, 24-pin products only.

Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 11, 20), mn = 00, 01, 03, 10,

2. ☐: Setting is fixed in the CSI master transmission mode, ▒: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

Downloaded from Arrow.com.

(2) Operation procedure

Figure 11-34. Initial Setting Procedure for Master Reception

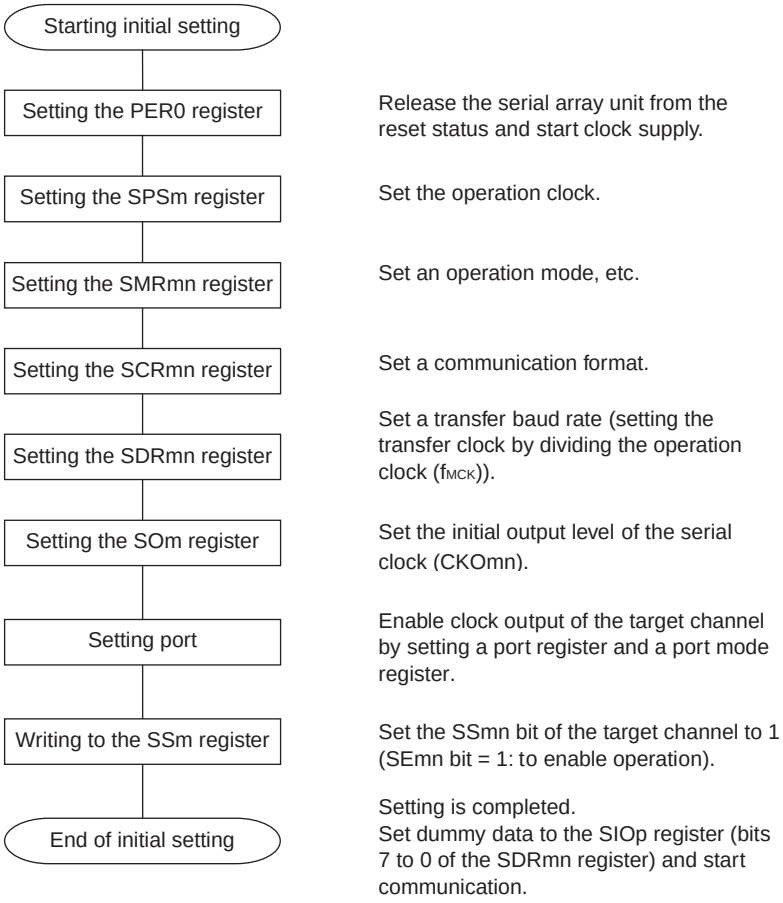
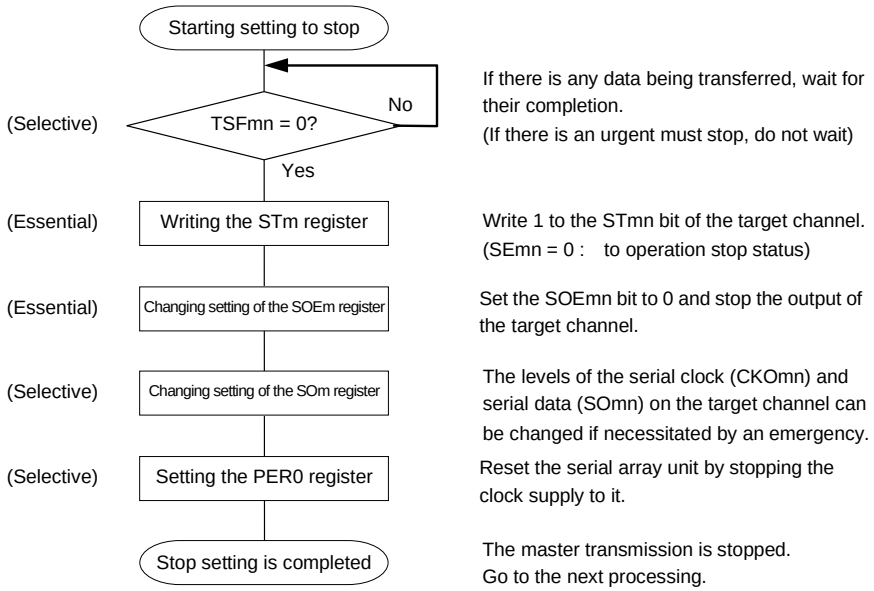


Figure 11-35. Procedure for Stopping Master Reception

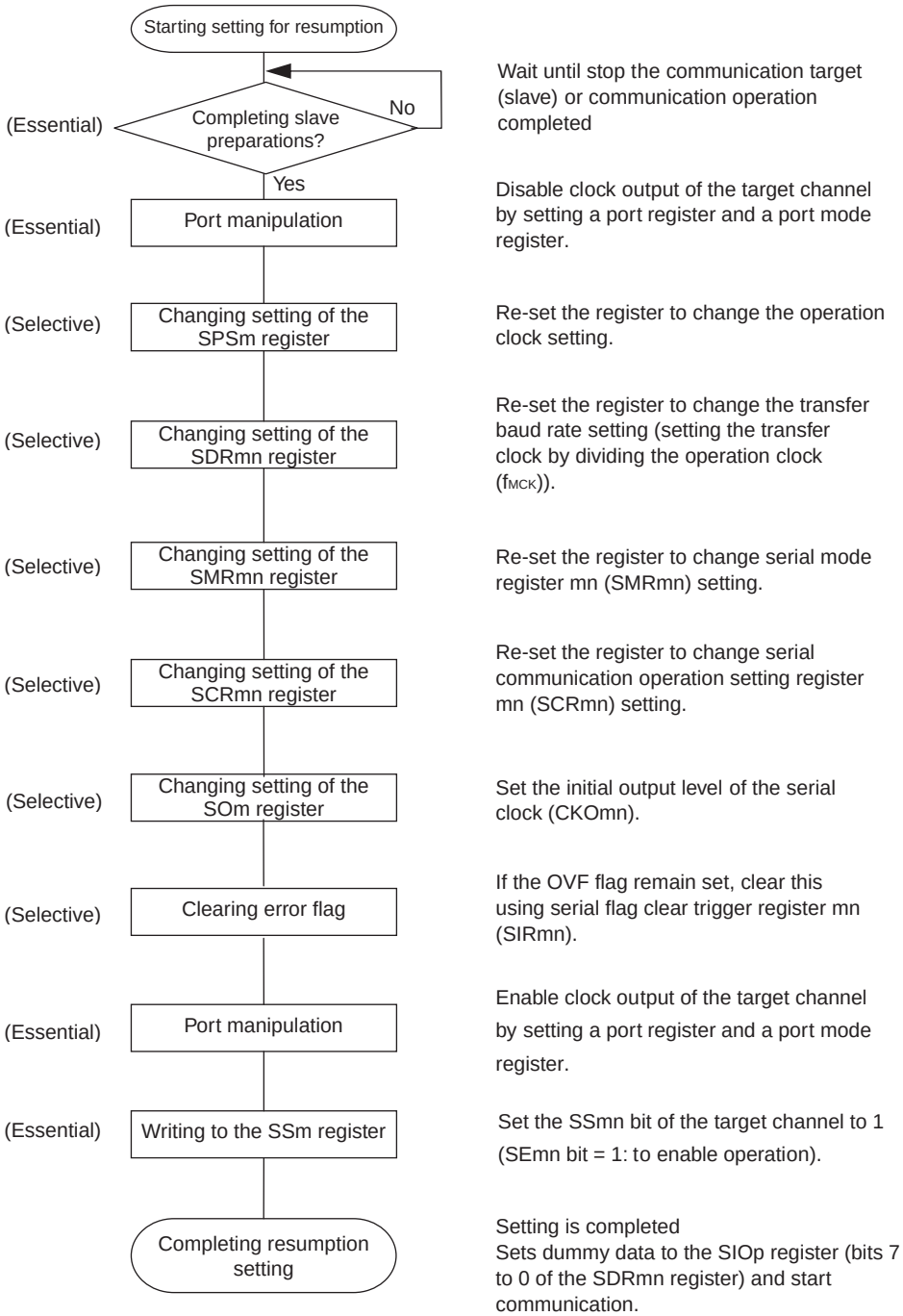
<R>



Downloaded from Arrow.com.

Figure 11-36. Procedure for Resuming Master Reception

<R>



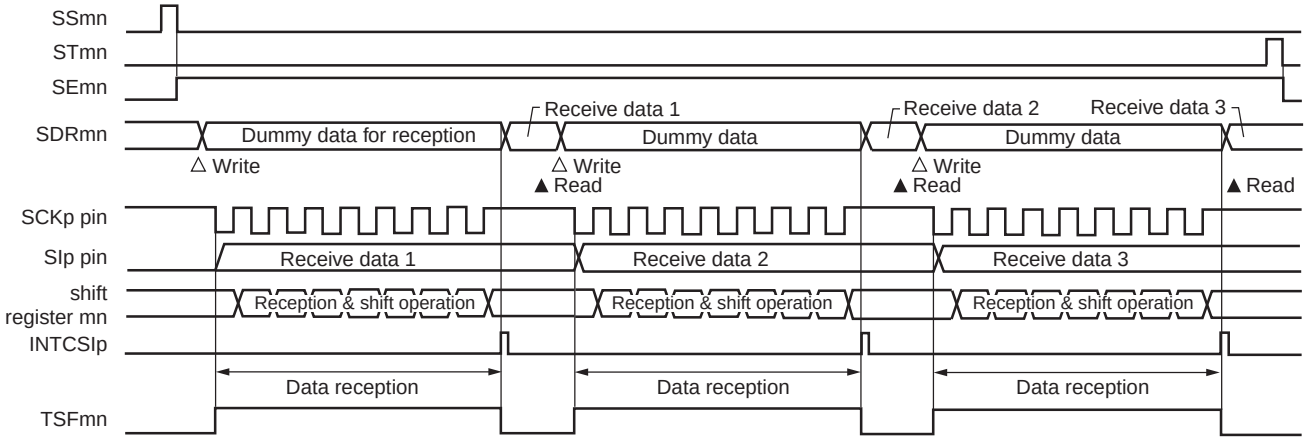
Remark If PER0 is rewritten while stopping the master transmission and the clock supply is stopped, wait until the transmission target (slave) stops or transmission finishes, and then perform initialization instead of restarting the transmission.

Downloaded from Arrow.com.

(3) Processing flow (in single-reception mode)

Figure 11-37. Timing Chart of Master Reception (in Single-Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)

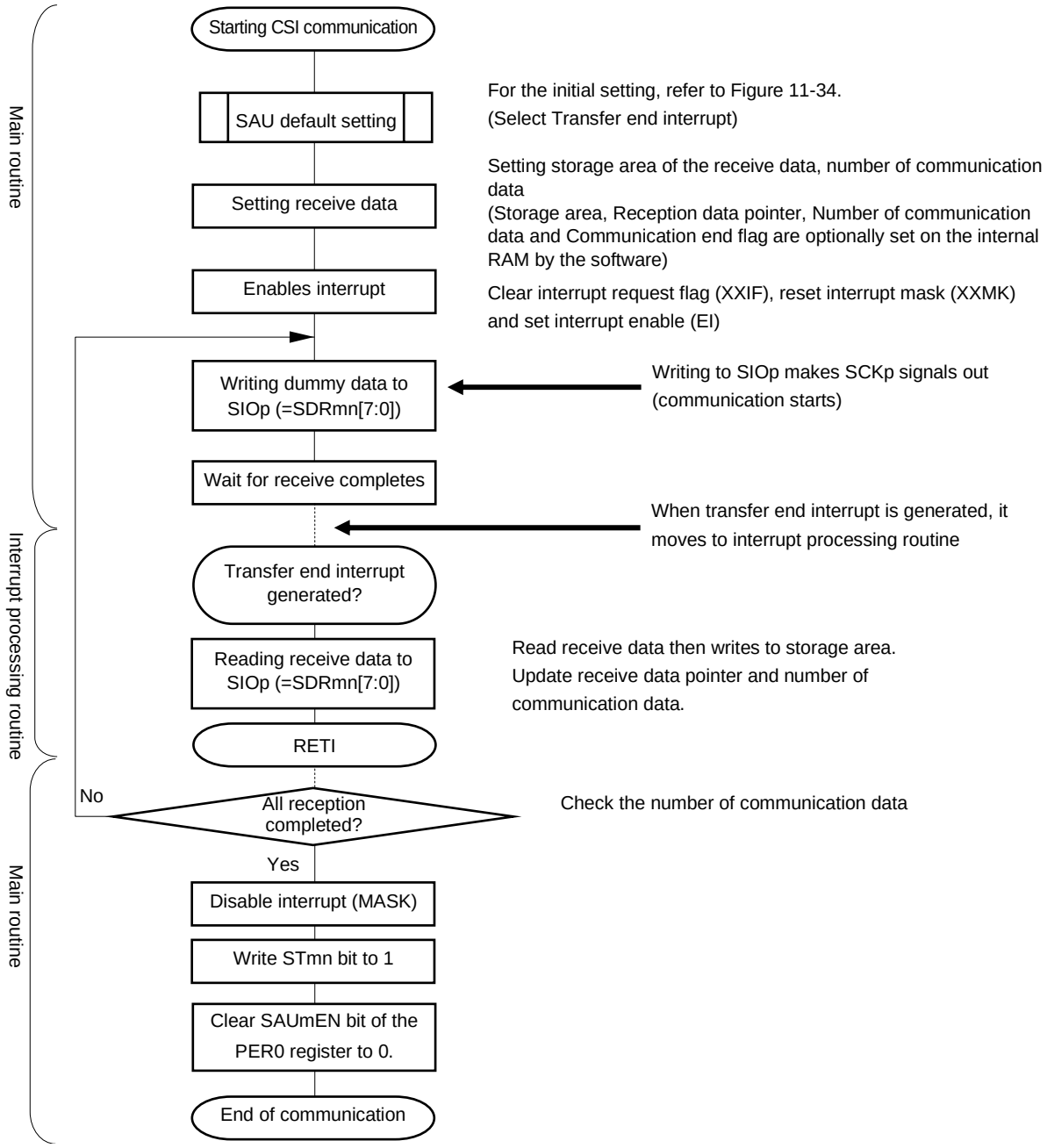
<R>



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 11, 20), mn = 00, 01, 03, 10,

Downloaded from Arrow.com.

Figure 11-38. Flowchart of Master Reception (in Single-Reception Mode)

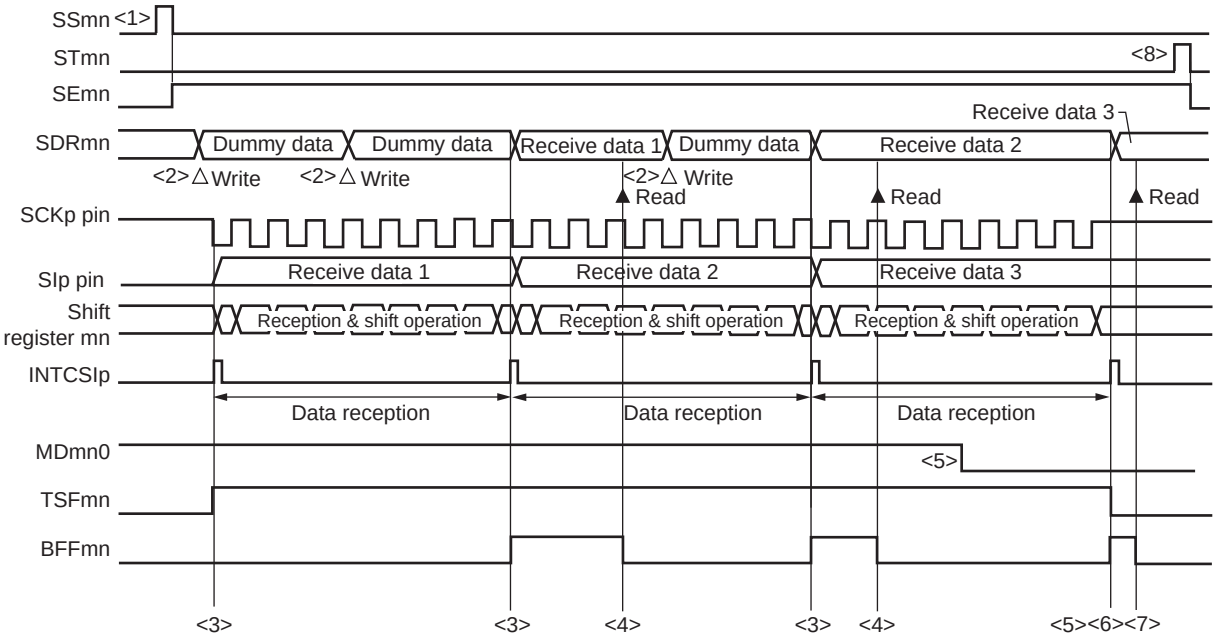


Downloaded from Arrow.com.

(4) Processing flow (in continuous reception mode)

Figure 11-39. Timing Chart of Master Reception (in Continuous Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)

<R>



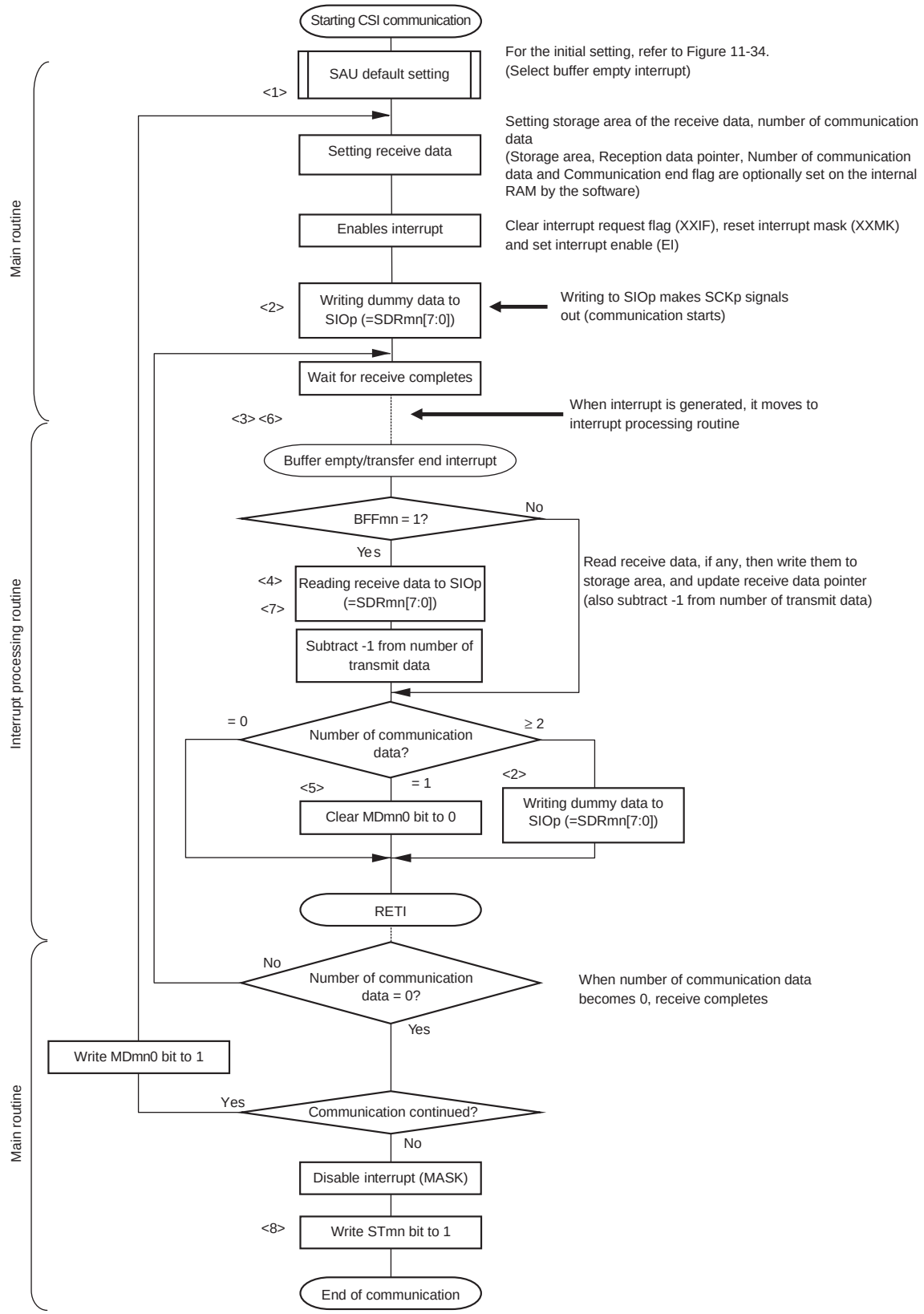
Caution The MDmn0 bit can be rewritten even during operation.
However, rewrite it before receive of the last bit is started, so that it has been rewritten before the transfer end interrupt of the last receive data.

Remarks 1. <1> to <8> in the figure correspond to <1> to <8> in **Figure 11-40 Flowchart of Master Reception (in Continuous Reception Mode)**.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 11, 20), mn = 00, 01, 03, 10

Downloaded from Arrow.com.

<R> **Figure 11-40. Flowchart of Master Reception (in Continuous Reception Mode)**



Remark <1> to <8> in the figure correspond to <1> to <8> in **Figure 11-39 Timing Chart of Master Reception (in Continuous Reception Mode)**.

Downloaded from Arrow.com.

11.5.3 Master transmission/reception

<R> Master transmission/reception is that the RL78 microcontroller outputs a transfer clock and transmits/receives data to/from other device.

3-Wire Serial I/O	CSI00	CSI01	CSI11	CSI20
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1
Pins used	SCK00, SI00, SO00	SCK01, SI01, SO01	SCK11, SI11, SO11	SCK20, SI20, SO20
Interrupt	INTCSI00	INTCSI01	INTCSI11	INTCSI20
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.			
Error detection flag	Overrun error detection flag (OVFmn) only			
Transfer data length	7 or 8 bits			
<R> Transfer rate[Note]	Max. $f_{CLK}/2$ [Hz] (CSI00 only), $f_{CLK}/4$ [Hz] Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [Hz] f_{CLK}: System clock frequency			
Data phase	Selectable by the DAPmn bit of the SCRmn register • DAPmn = 0: Data I/O starts at the start of the operation of the serial clock. • DAPmn = 1: Data I/O starts half a clock before the start of the serial clock operation.			
Clock phase	Selectable by the CKPmn bit of the SCRmn register • CKPmn = 0: Non-inversion • CKPmn = 1: Inverted			
Data direction	MSB or LSB first			

Note Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 28 ELECTRICAL SPECIFICATIONS (A, D: T_A = -40 to +85°C)** or **CHAPTER 29 ELECTRICAL SPECIFICATIONS (G: T_A = -40 to +105°C)**).

Remarks m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 13

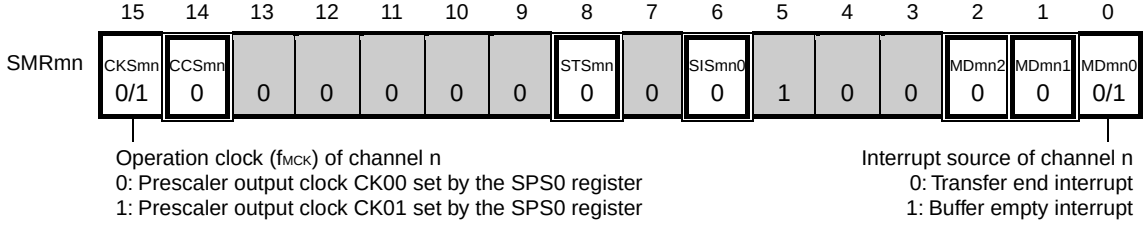
<R>

Downloaded from Arrow.com.

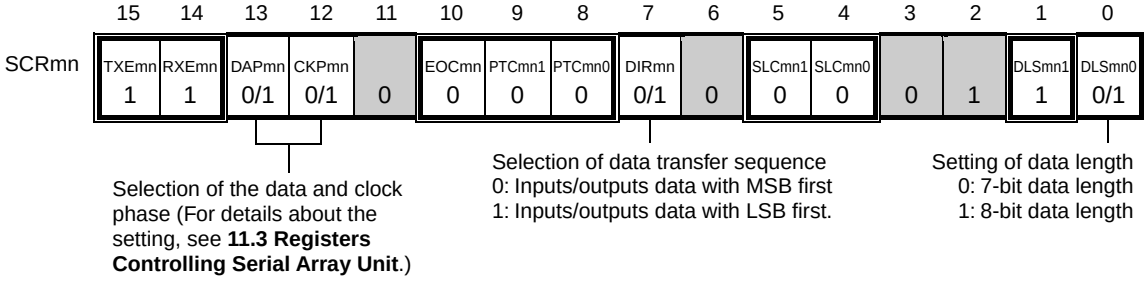
(1) Register setting

Figure 11-41. Example of Contents of Registers for Master Transmission/Reception of 3-Wire Serial I/O (CSI00, CSI01, CSI11, CSI20) (1/2)

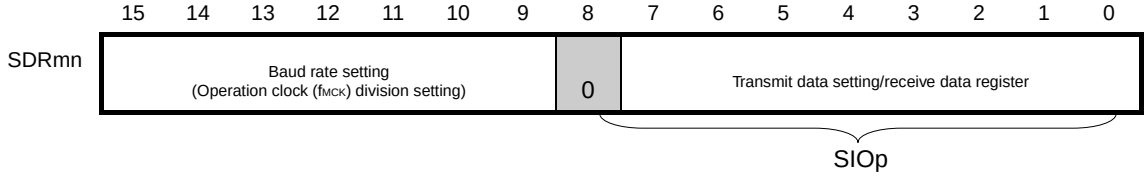
(a) Serial mode register mn (SMRmn)



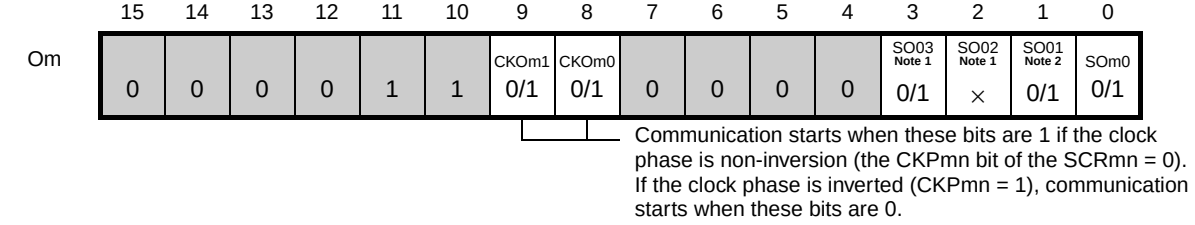
(b) Serial communication operation setting register mn (SCRmn)



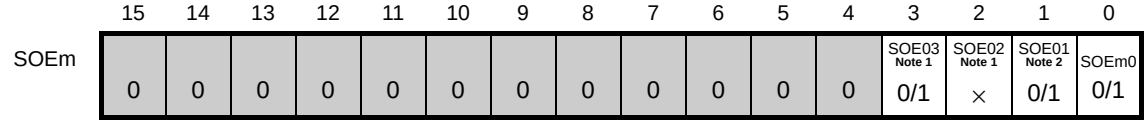
(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)



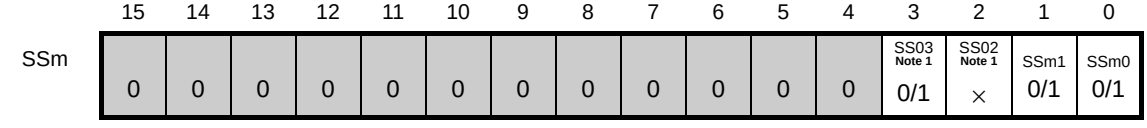
(d) Serial output register m (SOm) ... Sets only the bits of the target channel.



(e) Serial output enable register m (SOEm) ... Sets only the bits of the target channel to 1.



(f) Serial channel start register m (SSm) ... Sets only the bits of the target channel to 1.



Notes 1. Provided in the serial array unit 0 of 30-pin products.

2. 20-, 24-pin products only.

Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 11, 20), mn = 00, 01, 03, 10

2. ☐: Setting is fixed in the CSI master transmission/reception mode

▒: Setting disabled (set to the initial value)

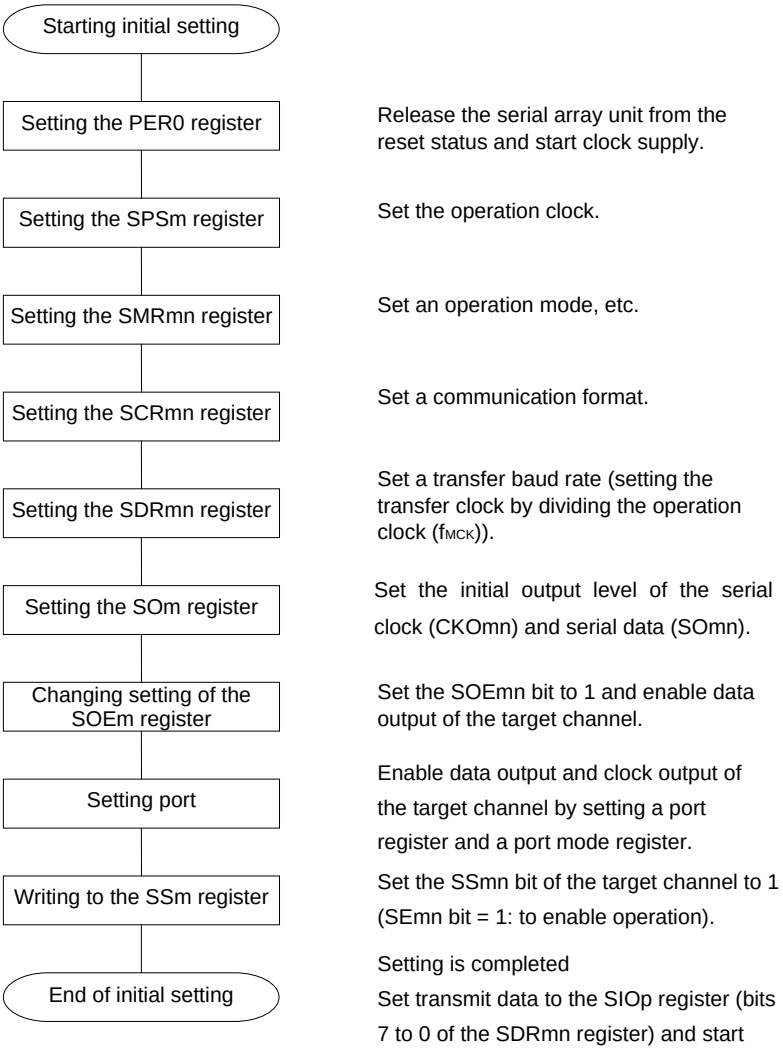
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)

0/1: Set to 0 or 1 depending on the usage of the user

Downloaded from Arrow.com.

(2) Operation procedure

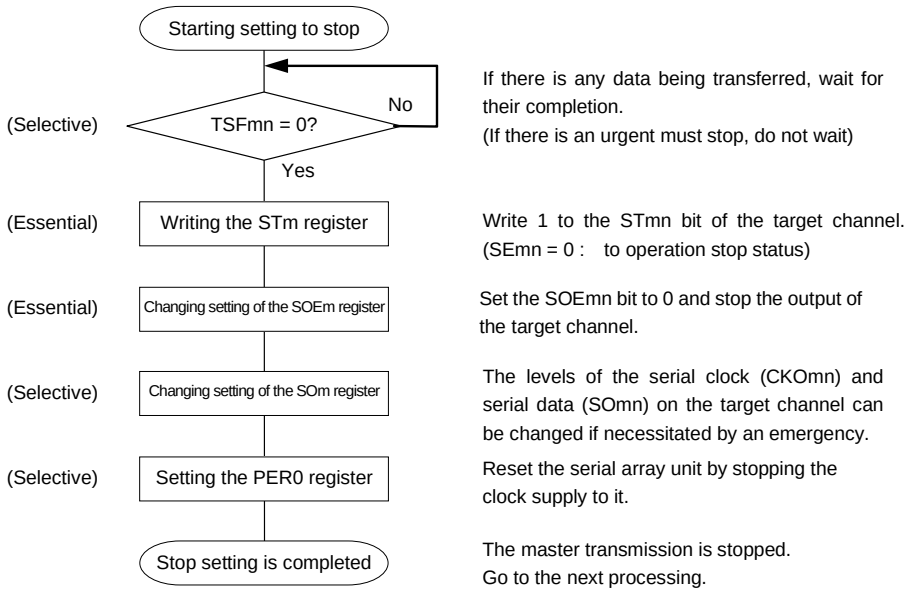
Figure 11-42. Initial Setting Procedure for Master Transmission/Reception



Downloaded from Arrow.com.

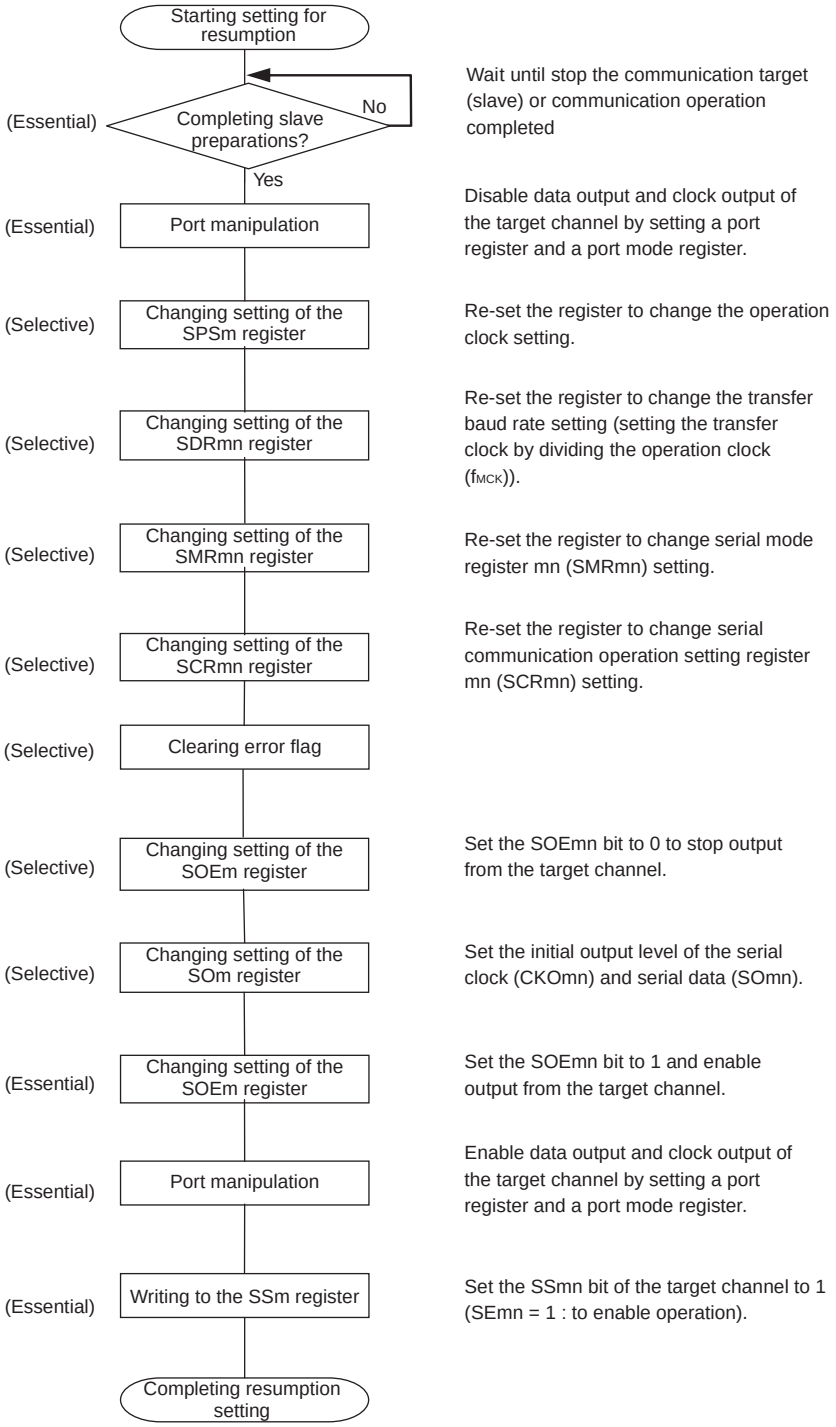
Figure 11-43. Procedure for Stopping Master Transmission/Reception

<R>



Downloaded from Arrow.com.

Figure 11-44. Procedure for Resuming Master Transmission/Reception

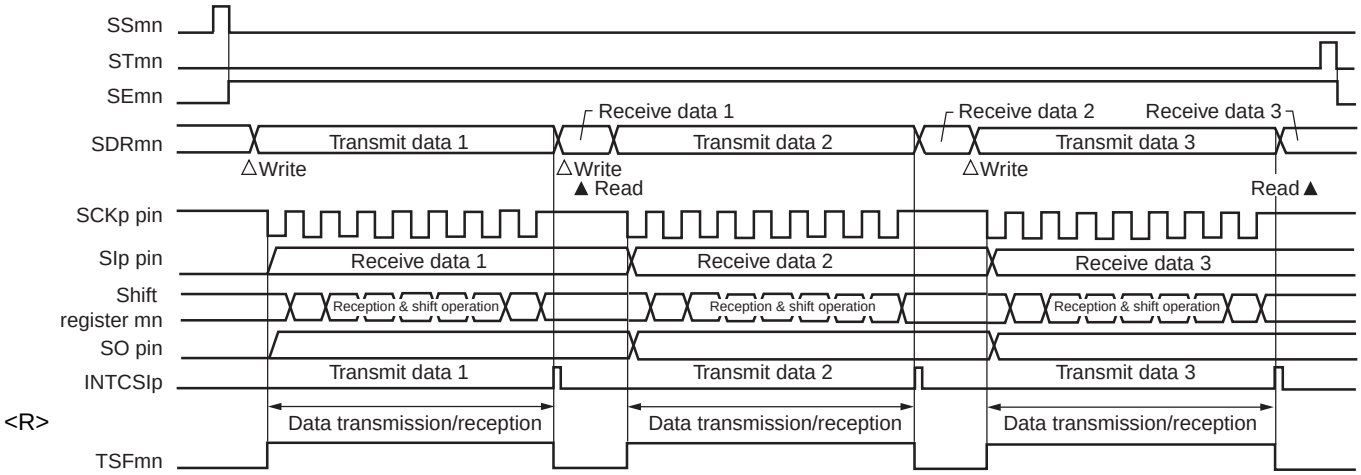


Remark If PER0 is rewritten while stopping the master transmission and the clock supply is stopped, wait until the transmission target (slave) stops or transmission finishes, and then perform initialization instead of restarting the transmission.

Downloaded from Arrow.com.

(3) Processing flow (in single-transmission/reception mode)

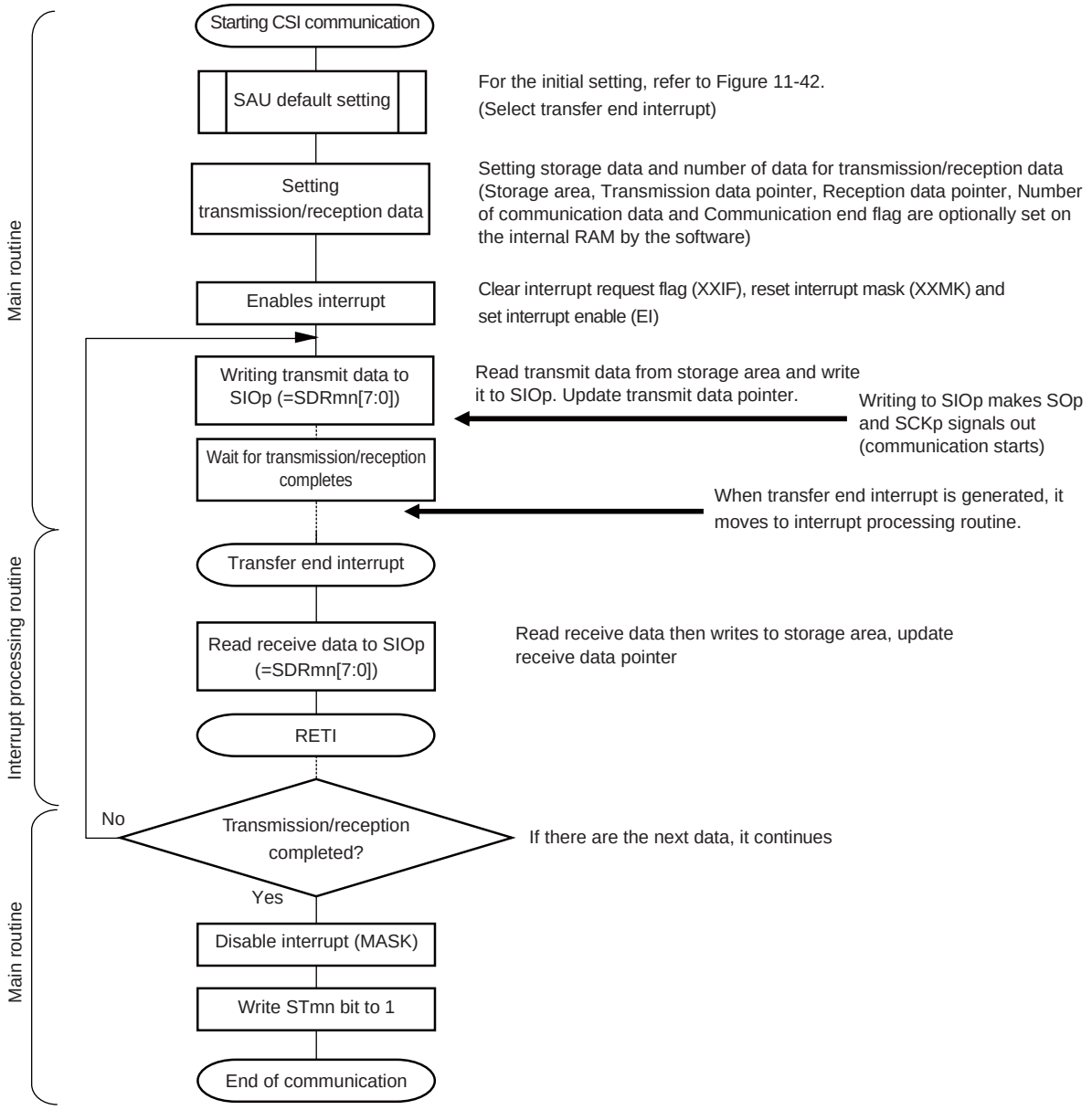
Figure 11-45. Timing Chart of Master Transmission/Reception (in Single-Transmission/Reception Mode) (Type 1: DAPmn =0, CKPmn = 0)



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 11, 20), mn = 00, 01, 03, 10

Downloaded from Arrow.com.

Figure 11-46. Flowchart of Master Transmission/Reception (in Single-Transmission/Reception Mode)

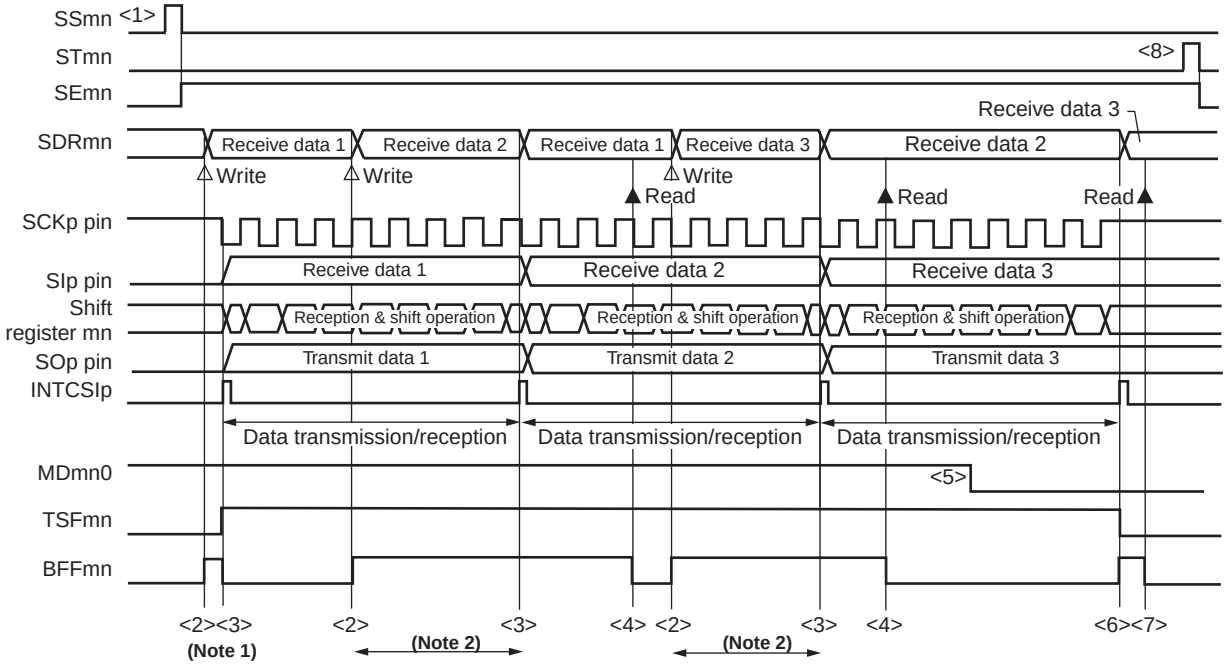


Downloaded from Arrow.com.

(4) Processing flow (in continuous transmission/reception mode)

Figure 11-47. Timing Chart of Master Transmission/Reception (in Continuous Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)

<R>



Notes 1. If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

2. The transmit data can be read by reading the SDRmn register during this period. At this time, the transfer operation is not affected.

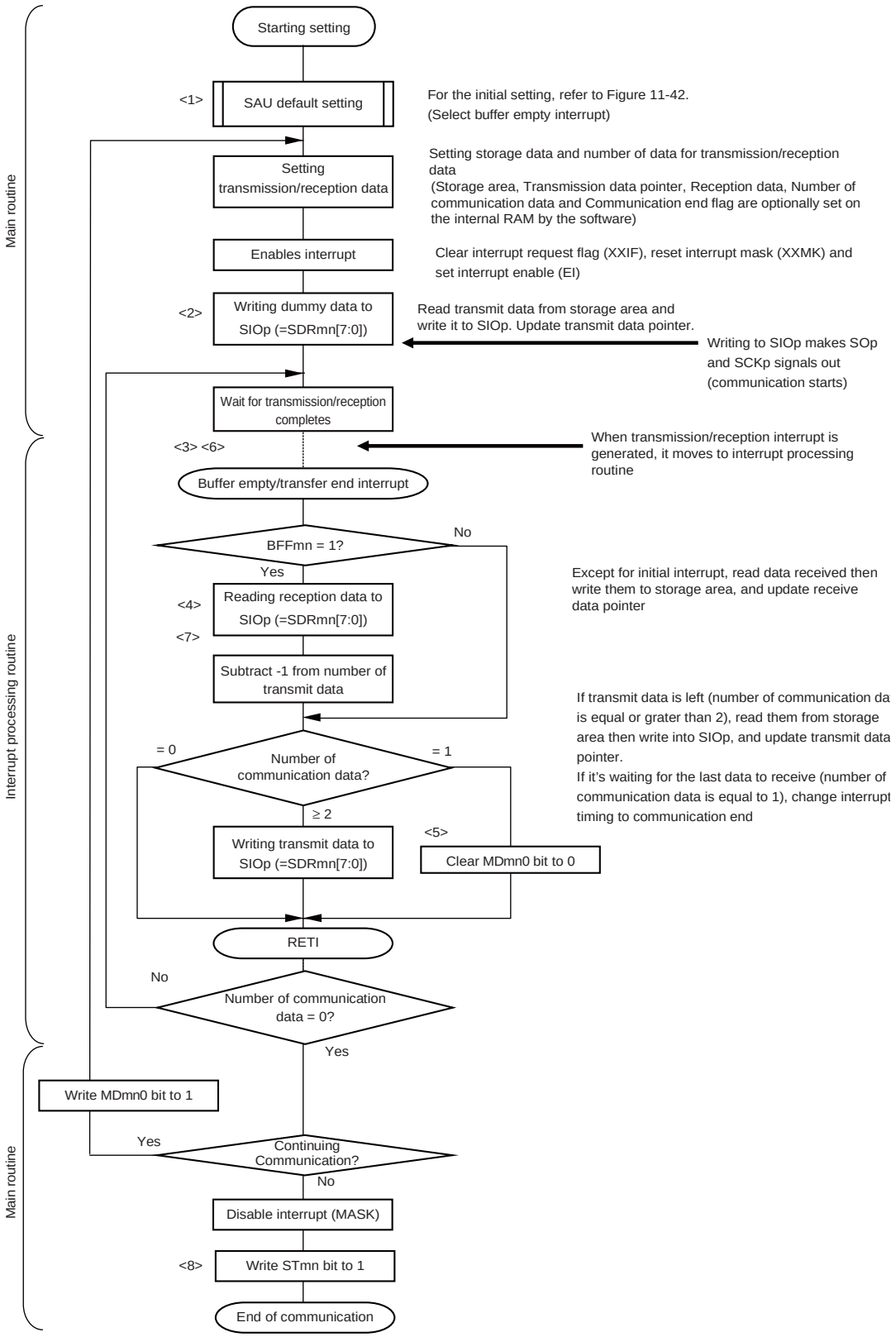
Caution The MDmn0 bit of serial mode register mn (SMRmn) can be rewritten even during operation. However, rewrite it before transfer of the last bit is started, so that it has been rewritten before the transfer end interrupt of the last transmit data.

Remarks 1. <1> to <8> in the figure correspond to <1> to <8> in **Figure 11-48 Flowchart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)**.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00, 01, 03, 10, p: CSI number (p = 00, 01, 11, 20)

Downloaded from Arrow.com.

<R> **Figure 11-48. Flowchart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)**



Remark <1> to <8> in the figure correspond to <1> to <8> in **Figure 11-47 Timing Chart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)**.

Downloaded from Arrow.com.

11.5.4 Slave transmission

<R> Slave transmission is that the RL78 microcontroller transmits data to another device in the state of a transfer clock being input from another device.

3-Wire Serial I/O	CSI00	CSI01	CSI11	CSI20
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1
Pins used	SCK00, SO00	SCK01, SO01	SCK11, SO11	SCK20, SO20
Interrupt	INTCSI00	INTCSI01	INTCSI11	INTCSI20
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.			
Error detection flag	Overrun error detection flag (OVFmn) only			
Transfer data length	7 or 8 bits			
Transfer rate	Max. f_{MCK}/6 [Hz]Notes 1, 2			
Data phase	Selectable by the DAPmn bit of the SCRmn register • DAPmn = 0: Data output starts at the start of the operation of the serial clock. • DAPmn = 1: Data output starts half a clock before the start of the serial clock operation.			
Clock phase	Selectable by the CKPmn bit of the SCRmn register • CKPmn = 0: Non-inversion • CKPmn = 1: Inverted			
Data direction	MSB or LSB first			

<R> **Notes 1.** Because the external serial clock input to the SCK00, SCK01, SCK11, and SCK20 pins is sampled internally and used, the fastest transfer rate is f_{MCK}/6 [Hz].

<R> **2.** Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 28 ELECTRICAL SPECIFICATIONS (A, D: T_A = -40 to +85°C)** or **CHAPTER 29 ELECTRICAL SPECIFICATIONS (G: T_A = -40 to +105°C)**).

Remarks 1. f_{MCK}: Operation clock frequency of target channel
f_{SCK}: Serial clock frequency

2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00, 01, 03, 10

<R>

Downloaded from Arrow.com.

(1) Register setting

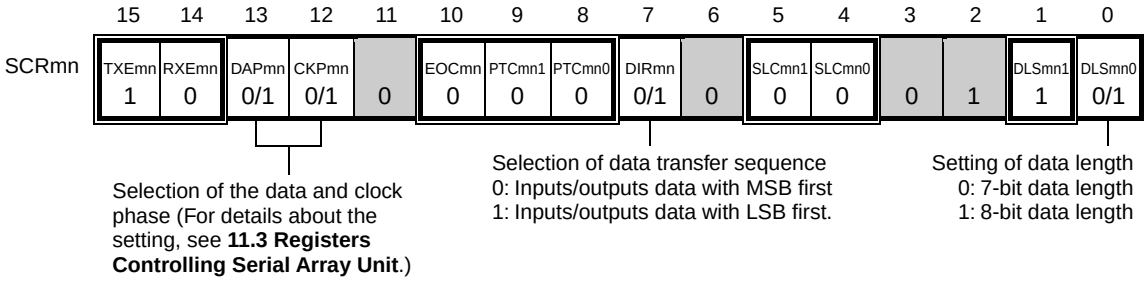
Figure 11-49. Example of Contents of Registers for Slave Transmission of 3-Wire Serial I/O (CSI00, CSI01, CSI11, CSI20)

(a) Serial mode register mn (SMRmn)

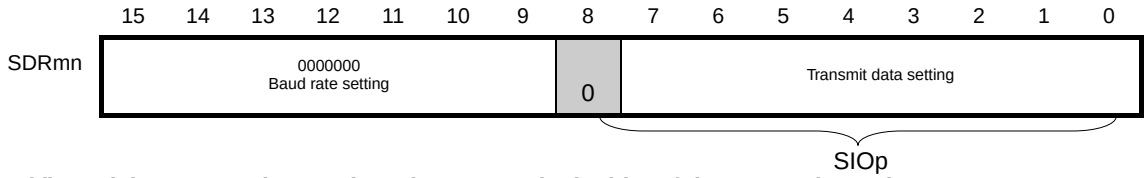
<R>



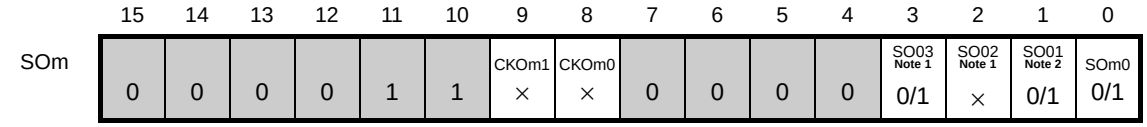
(b) Serial communication operation setting register mn (SCRmn)



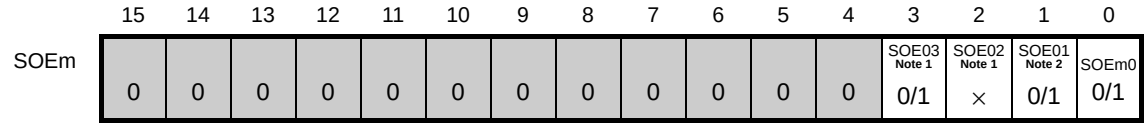
(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)



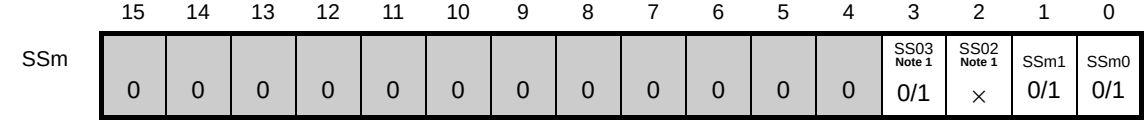
(d) Serial output register m (SOm) ... Sets only the bits of the target channel.



(e) Serial output enable register m (SOEm) ... Sets only the bits of the target channel to 1.



(f) Serial channel start register m (SSm) ... Sets only the bits of the target channel to 1.



Notes 1. Provided in the serial array unit 0 of 30-pin products.

2. 20-, 24-pin products only.

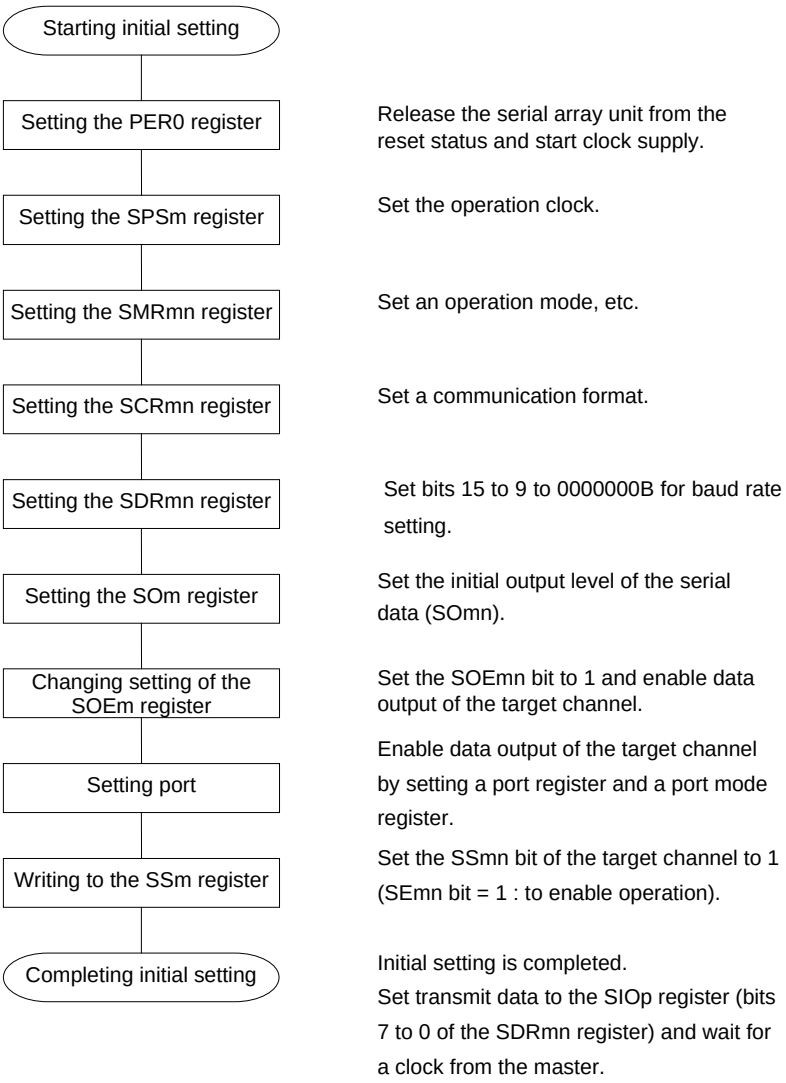
Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 11, 20), mn = 00, 01, 03, 10

2. ☐: Setting is fixed in the CSI master transmission mode, ▒: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

Downloaded from Arrow.com.

(2) Operation procedure

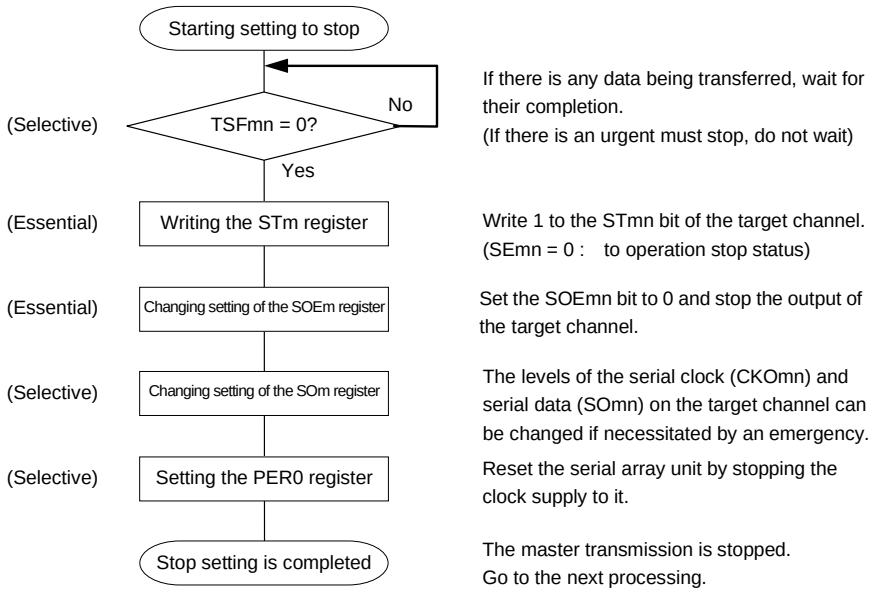
Figure 11-50. Initial Setting Procedure for Slave Transmission



Downloaded from Arrow.com.

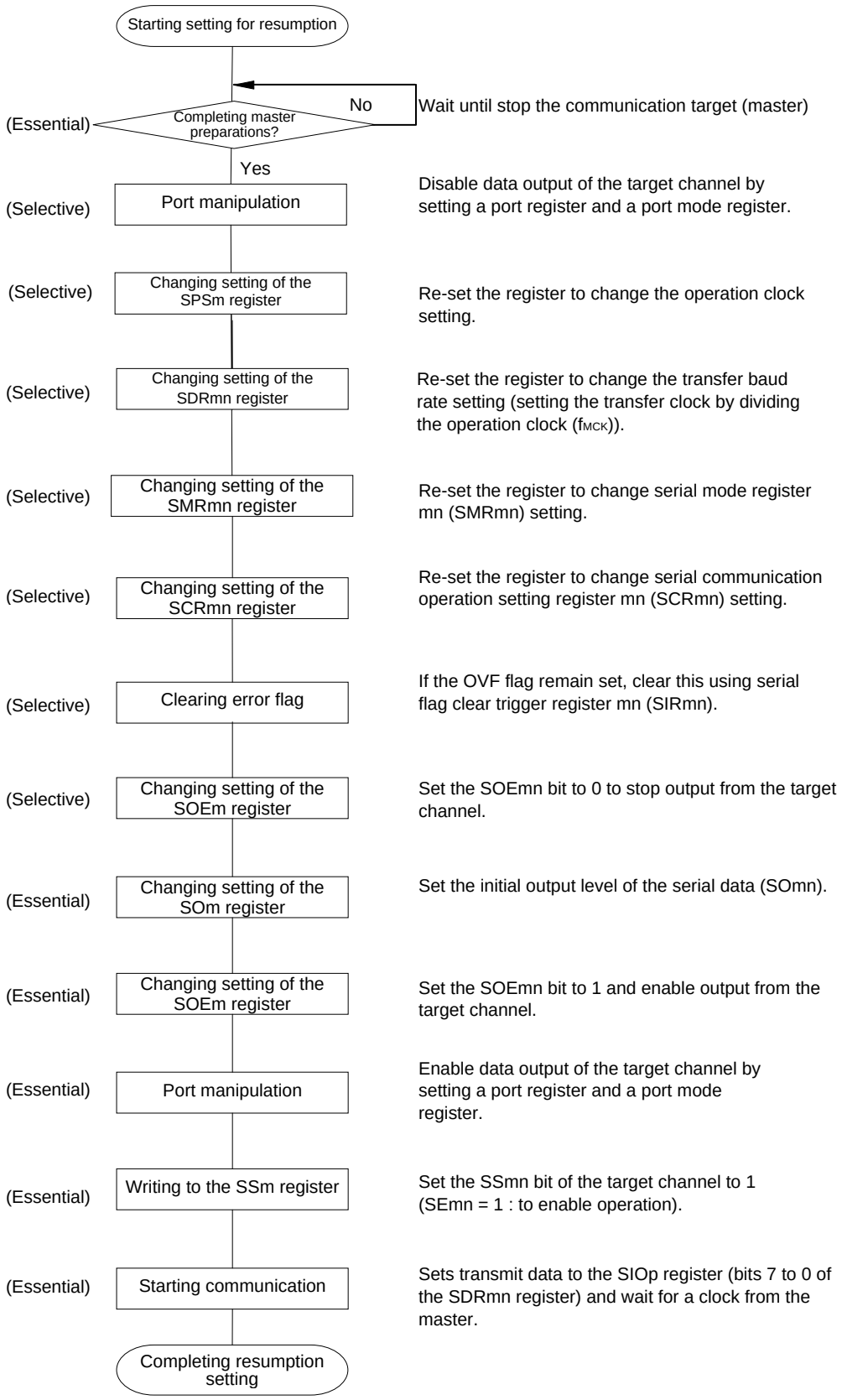
Figure 11-51. Procedure for Stopping Slave Transmission

<R>



Downloaded from Arrow.com.

Figure 11-52. Procedure for Resuming Slave Transmission



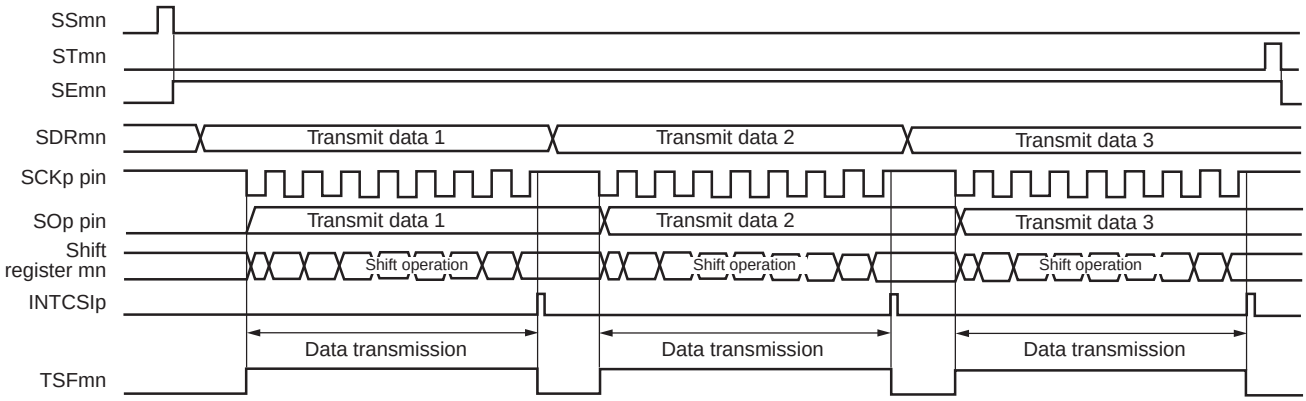
Remark If PER0 is rewritten while stopping the master transmission and the clock supply is stopped, wait until the transmission target (master) stops or transmission finishes, and then perform initialization instead of restarting the transmission.

Downloaded from Arrow.com.

(3) Processing flow (in single-transmission mode)

Figure 11-53. Timing Chart of Slave Transmission (in Single-Transmission Mode)
(Type 1: DAPmn = 0, CKPmn = 0)

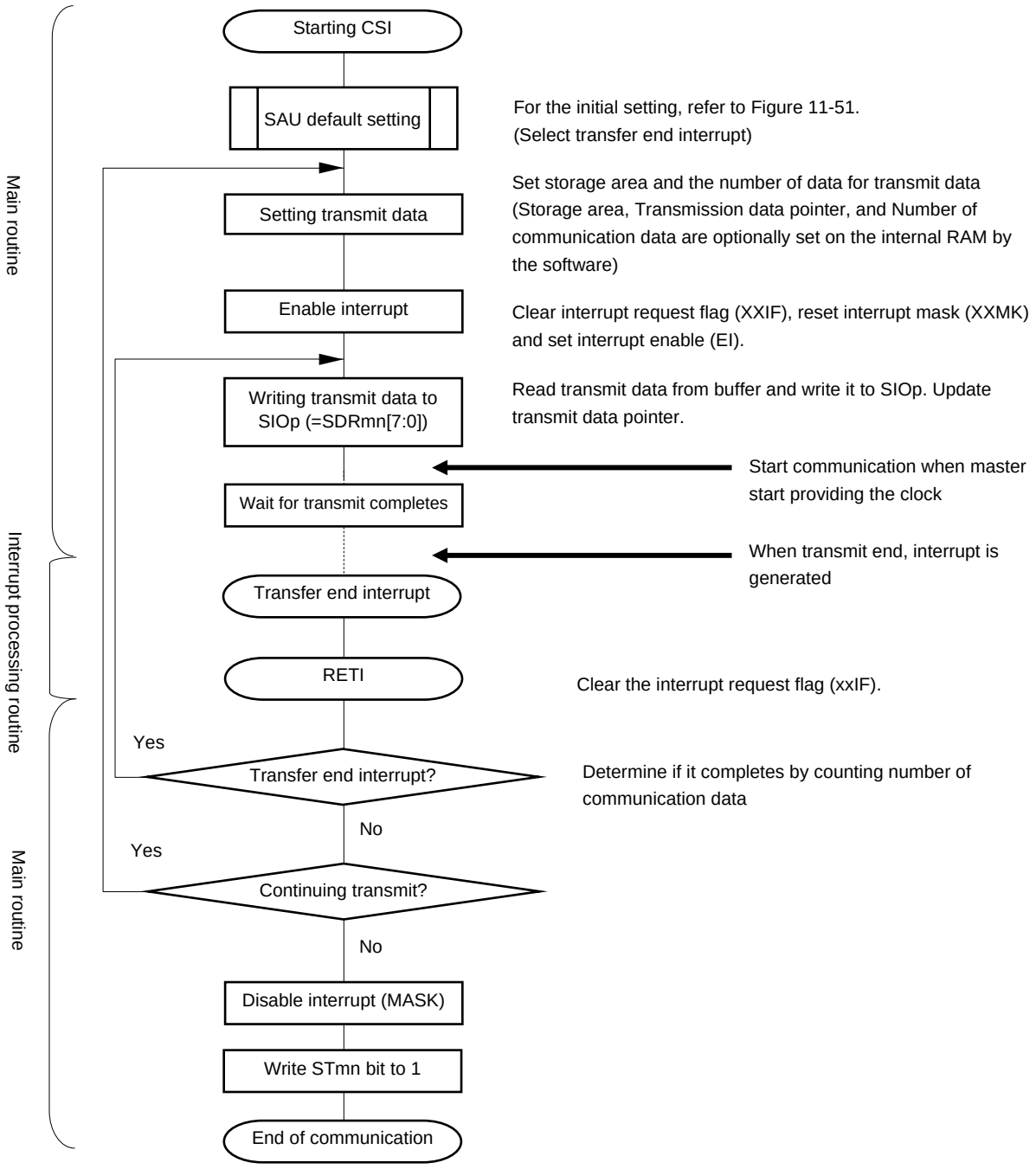
<R>



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 11, 20), mn = 00, 01, 03, 10

Downloaded from Arrow.com.

<R> **Figure 11-54. Flowchart of Slave Transmission (in Single-Transmission Mode)**

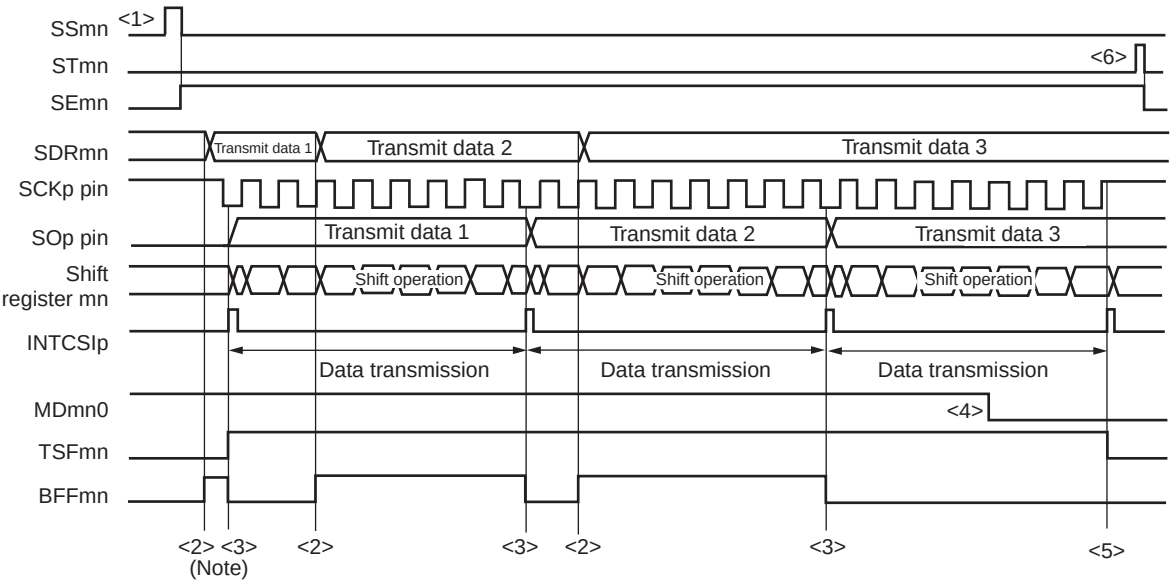


Downloaded from Arrow.com.

(4) Processing flow (in continuous transmission mode)

Figure 11-55. Timing Chart of Slave Transmission (in Continuous Transmission Mode)
(Type 1: DAPmn = 0, CKPmn = 0)

<R>



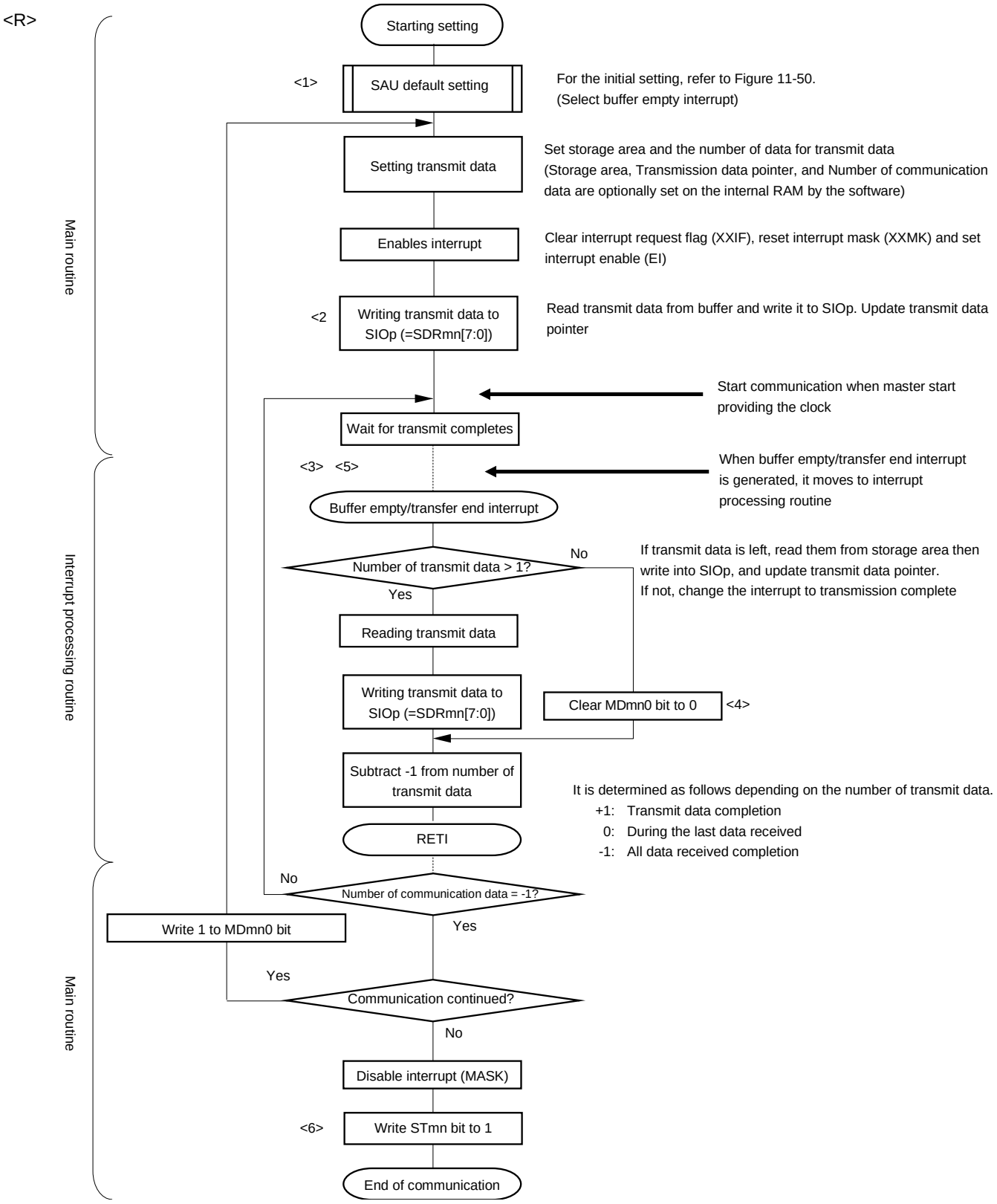
Note If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

Caution The MDmn0 bit of serial mode register mn (SMRmn) can be rewritten even during operation. However, rewrite it before transfer of the last bit is started.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 11, 20), mn = 00, 01, 03, 10

Downloaded from Arrow.com.

Figure 11-56. Flowchart of Slave Transmission (in Continuous Transmission Mode)



Remark <1> to <6> in the figure correspond to <1> to <6> in **Figure 11-55 Timing Chart of Slave Transmission (in Continuous Transmission Mode)**.

Downloaded from Arrow.com.

11.5.5 Slave reception

<R> Slave reception is that the RL78 microcontroller receives data from another device in the state of a transfer clock being input from another device.

3-Wire Serial I/O	CSI00	CSI01	CSI11	CSI20
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1
Pins used	SCK00, SI00	SCK01, SI01	SCK11, SI11	SCK20, SI20
Interrupt	INTCSI00	INTCSI01	INTCSI11	INTCSI20
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.			
Error detection flag	Overrun error detection flag (OVFmn) only			
Transfer data length	7 or 8 bits			
Transfer rate	Max. f_{MCK}/6 [Hz]Notes 1, 2.			
Data phase	Selectable by the DAPmn bit of the SCRmn register • DAPmn = 0: Data output starts at the start of the operation of the serial clock. • DAPmn = 1: Data output starts half a clock before the start of the serial clock operation.			
Clock phase	Selectable by the CKPmn bit of the SCRmn register • CKPmn = 0: Non-inversion • CKPmn = 1: Inverted			
Data direction	MSB or LSB first			

<R> **Notes 1.** Because the external serial clock input to the SCK00, SCK01, SCK11, and SCK20 pins is sampled internally and used, the fastest transfer rate is f_{MCK}/6 [Hz].

<R> **2.** Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 28 ELECTRICAL SPECIFICATIONS (A, D: T_A = -40 to +85°C)** or **CHAPTER 29 ELECTRICAL SPECIFICATIONS (G: T_A = -40 to +105°C)**).

Remarks 1. f_{MCK}: Operation clock frequency of target channel
f_{SCK}: Serial clock frequency

2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00, 01, 03, 10

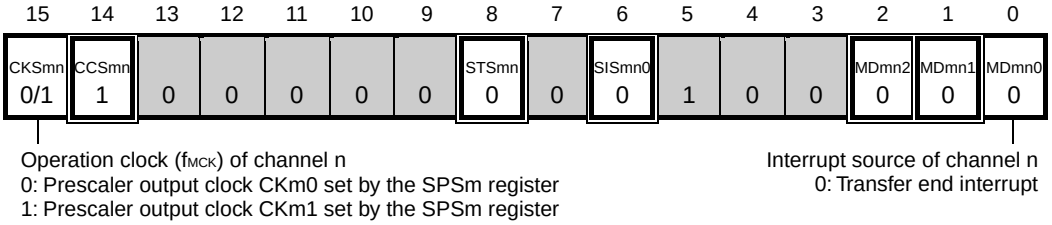
Downloaded from Arrow.com.

(1) Register setting

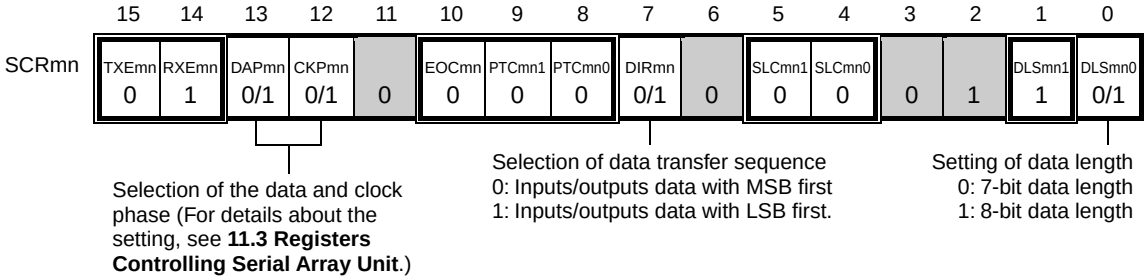
Figure 11-57. Example of Contents of Registers for Slave Reception of 3-Wire Serial I/O (CSI00, CSI01, CSI11, CSI20)

(a) Serial mode register mn (SMRmn)

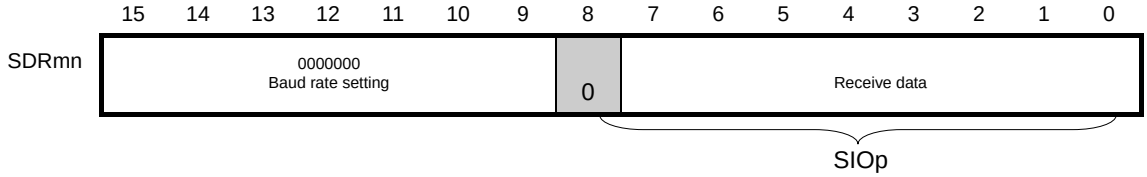
<R>



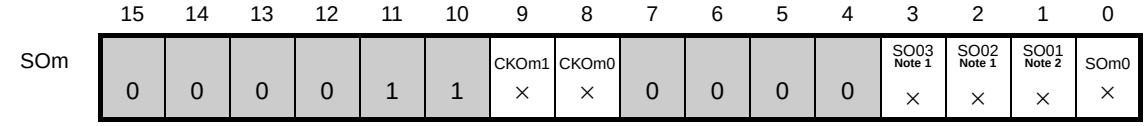
(b) Serial communication operation setting register mn (SCRmn)



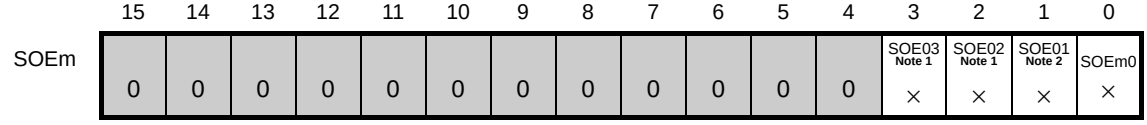
(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)



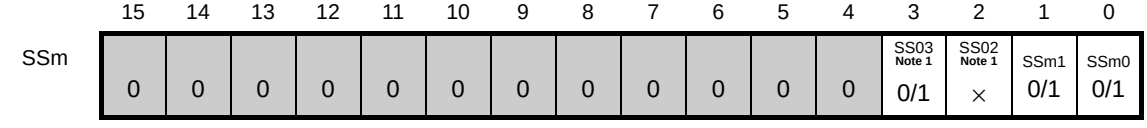
(d) Serial output register m (SOm) ... The Register that not used in this mode.



(e) Serial output enable register m (SOEm) ... The Register that not used in this mode.



(f) Serial channel start register m (SSm) ... Sets only the bits of the target channel to 1.



Notes 1. Provided in the serial array unit 0 of 30-pin products.

2. 20-, 24-pin products only.

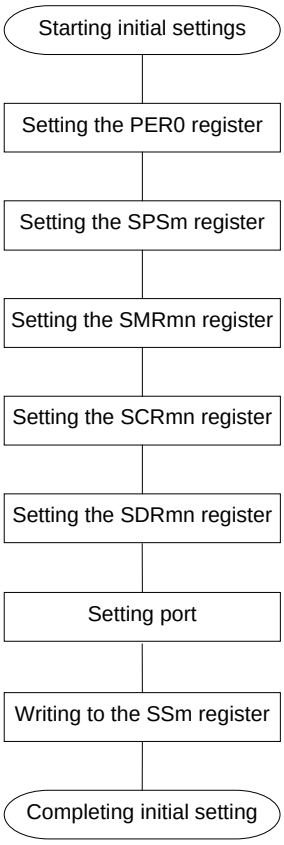
Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 11, 20), mn = 00, 01, 03, 10

2. ☐: Setting is fixed in the CSI master transmission mode, ▒: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

Downloaded from Arrow.com.

(2) Operation procedure

Figure 11-58. Initial Setting Procedure for Slave Reception



<R>

Release the serial array unit from the reset status and start clock supply.

Set the operation clock.

Set an operation mode, etc.

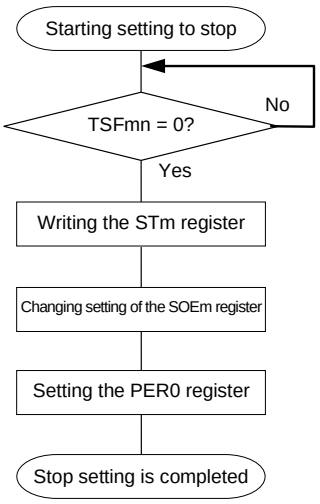
Set a communication format.

Set baud rate setting (bits 15 to 9) to 0000000B.

Enable data input and clock input of the target channel by setting a port register and a port mode register.

Set the SSmn bit of the target channel to 1 (SEmn bit = 1: to enable operation). Wait for a clock from the master.

Figure 11-59. Procedure for Stopping Slave Reception



If there is any data being transferred, wait for their completion.
(If there is an urgent must stop, do not wait)

Write 1 to the STmn bit of the target channel.
(SEmn = 0 : to operation stop status)

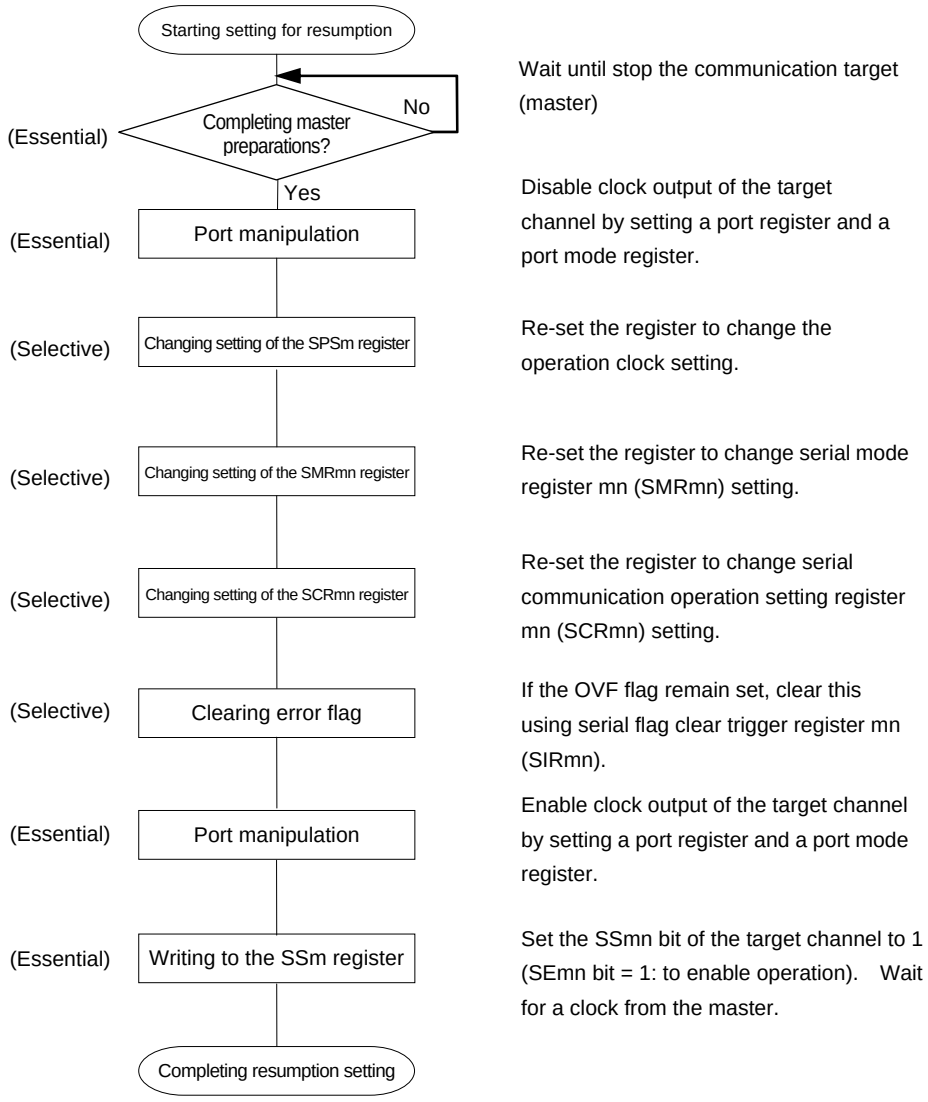
Set the SOEmn bit to 0 and stop the output of the target channel.

Reset the serial array unit by stopping the clock supply to it.

The master transmission is stopped.
Go to the next processing.

Downloaded from Arrow.com.

Figure 11-60. Procedure for Resuming Slave Reception

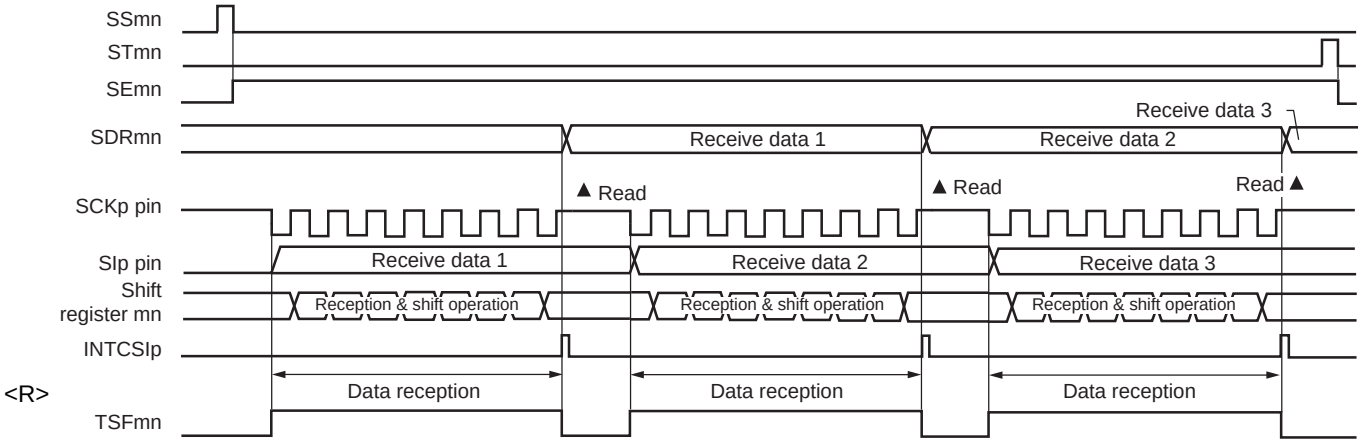


Remark If PER0 is rewritten while stopping the master transmission and the clock supply is stopped, wait until the transmission target (master) stops or transmission finishes, and then perform initialization instead of restarting the transmission.

Downloaded from Arrow.com.

(3) Processing flow (in single-reception mode)

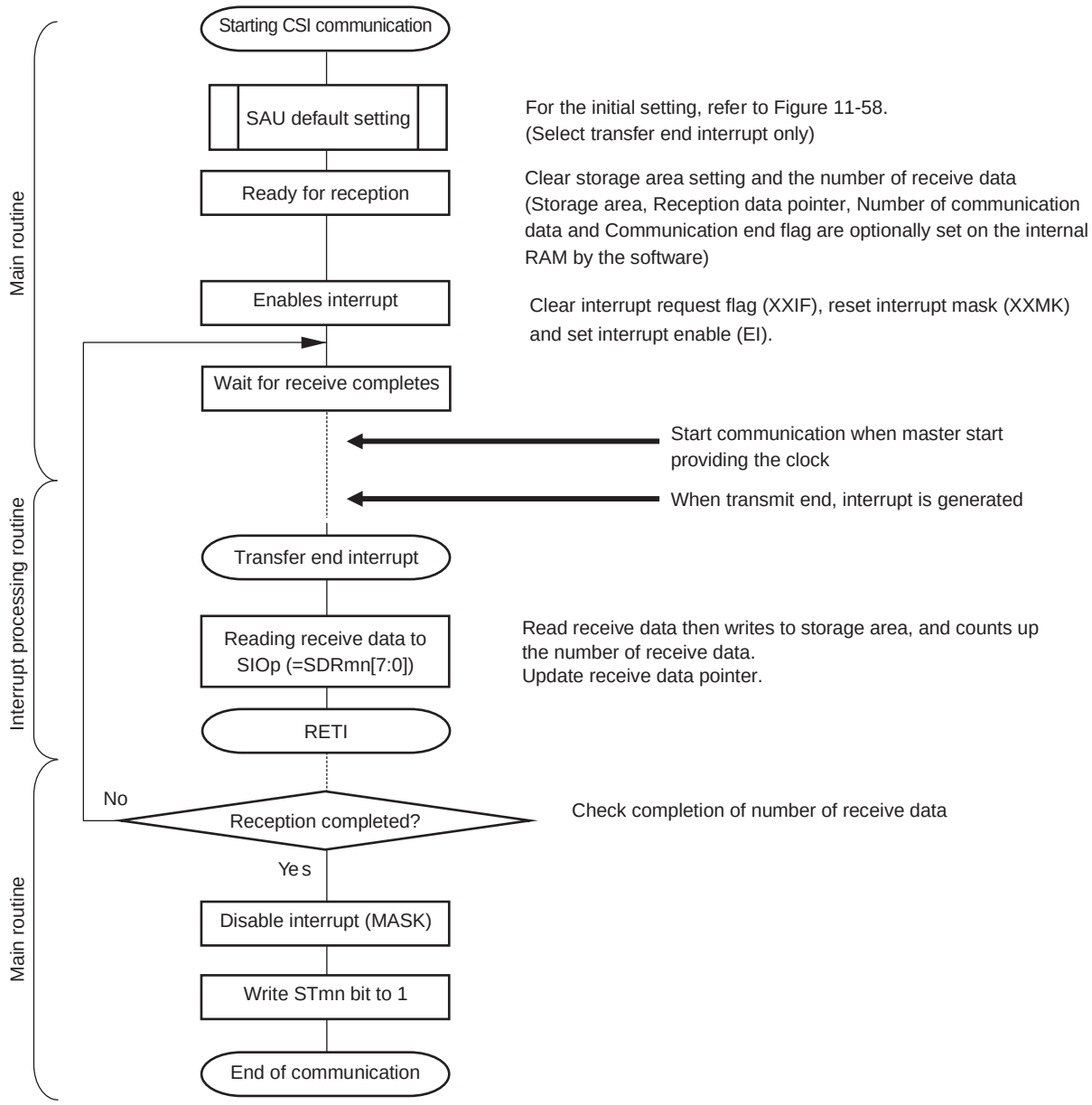
Figure 11-61. Timing Chart of Slave Reception (in Single-Reception Mode)
(Type 1: DAPmn = 0, CKPmn = 0)



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 11, 20), mn = 00, 01, 03, 10

Downloaded from Arrow.com.

<R> **Figure 11-62. Flowchart of Slave Reception (in Single-Reception Mode)**



Downloaded from Arrow.com.

11.5.6 Slave transmission/reception

<R> Slave transmission/reception is that the RL78 microcontroller transmits/receives data to/from another device in the state of a transfer clock being input from another device.

3-Wire Serial I/O	CSI00	CSI01	CSI11	CSI20
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1
Pins used	SCK00, SI00, SO00	SCK01, SI01, SO01	SCK11, SI11, SO11	SCK20, SI20, SO20
Interrupt	INTCSI00	INTCSI01	INTCSI11	INTCSI20
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.			
Error detection flag	Overrun error detection flag (OVFmn) only			
Transfer data length	7 or 8 bits			
Transfer rate	Max. f_{MCK}/6 [Hz]Notes 1, 2			
Data phase	Selectable by the DAPmn bit of the SCRmn register • DAPmn = 0: Data output starts at the start of the operation of the serial clock. • DAPmn = 1: Data output starts half a clock before the start of the serial clock operation.			
Clock phase	Selectable by the CKPmn bit of the SCRmn register • CKPmn = 0: Non-inversion • CKPmn = 1: Inverted			
Data direction	MSB or LSB first			

<R> **Notes 1.** Because the external serial clock input to the SCK00, SCK01, SCK11, SCK20 pins is sampled internally and used, the fastest transfer rate is f_{MCK}/6 [Hz].

<R> **2.** Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 28 ELECTRICAL SPECIFICATIONS (A, D: T_A = -40 to +85°C)** or **CHAPTER 29 ELECTRICAL SPECIFICATIONS (G: T_A = -40 to +105°C)**).

Remarks 1. f_{MCK}: Operation clock frequency of target channel
f_{SCK}: Serial clock frequency

2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00, 01, 03, 10

Downloaded from Arrow.com.

(1) Register setting

Figure 11-63. Example of Contents of Registers for Slave Transmission/Reception of 3-Wire Serial I/O (CSI00, CSI01, CSI11, CSI20)

(a) Serial mode register mn (SMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn	CCSmn	0	0	0	0	0	STSmn	0	SISmn0	1	0	0	MDmn2	MDmn1	MDmn0
	0/1	1						0		0				0	0	0/1

Operation clock (f_{MCK}) of channel n
0: Prescaler output clock CKm0 set by the SPSm register
1: Prescaler output clock CKm1 set by the SPSm register

Interrupt source of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(b) Serial communication operation setting register mn (SCRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn	RXEmn	DAPmn	CKPmn		EOCmn	PTCmn1	PTCmn0	DIRmn		SLCmn1	SLCmn0			DLSmn1	DLSmn0
	1	1	0/1	0/1	0	0	0	0	0/1	0	0	0	0	1	1	0/1

Selection of the data and clock phase (For details about the setting, see **11.3 Registers Controlling Serial Array Unit**.)

Selection of data transfer sequence
0: Inputs/outputs data with MSB first
1: Inputs/outputs data with LSB first.

Setting of data length
0: 7-bit data length
1: 8-bit data length

(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	0000000 Baud rate setting							0	Transmit data setting/receive data register							

SIOp

(d) Serial output register m (SOm) ... Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	1	1	CKOm1	CKOm0	0	0	0	0	SO03 Note 1	SO02 Note 1	SO01 Note 2	SOm0
							×	×					0/1	×	0/1	0/1

(e) Serial output enable register m (SOEm) ... Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	SOE03 Note 1	SOE02 Note 1	SOE01 Note 2	SOEm0
													0/1	×	0/1	0/1

(f) Serial channel start register m (SSm) ... Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SS03 Note 1	SS02 Note 1	SSm1	SSm0
													0/1	×	0/1	0/1

Notes 1. Provided in the serial array unit 0 of 30-pin products only.

2. 20-, 24-pin products only.

Caution Be sure to set transmit data to the SIOp register before the clock from the master is started.

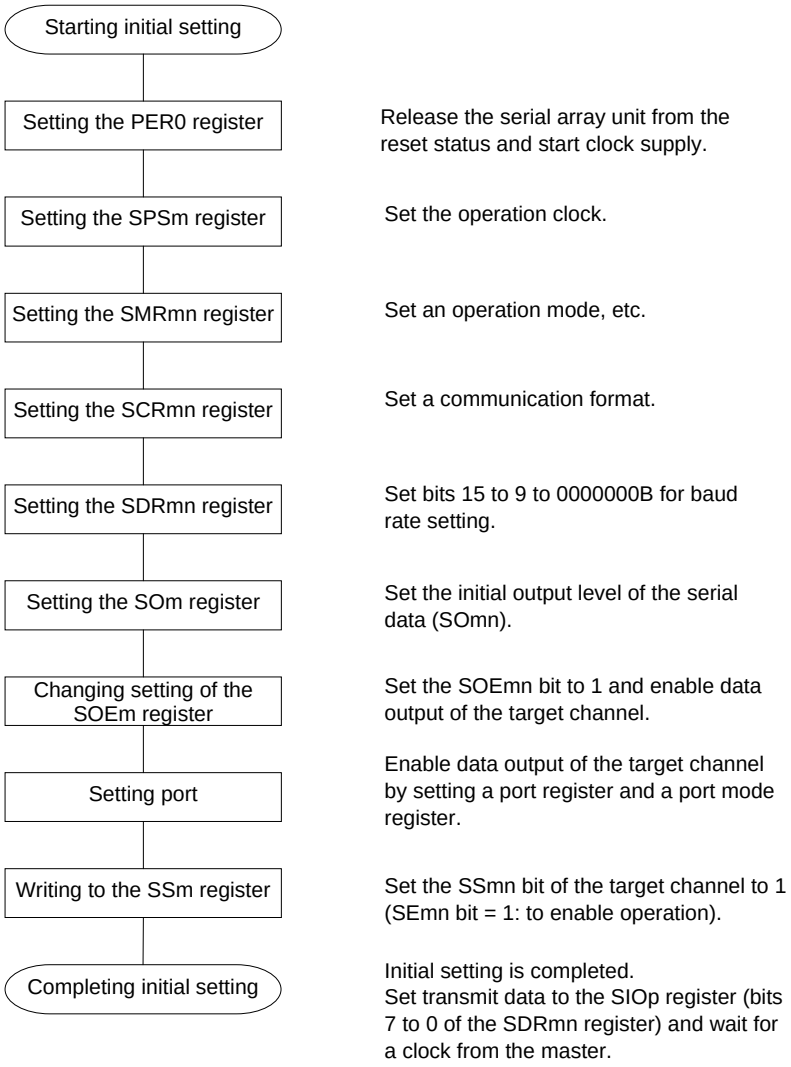
Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 11, 20), mn = 00, 01, 03, 10

2. ☐: Setting is fixed in the CSI master transmission mode, ▒: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

Downloaded from Arrow.com.

(2) Operation procedure

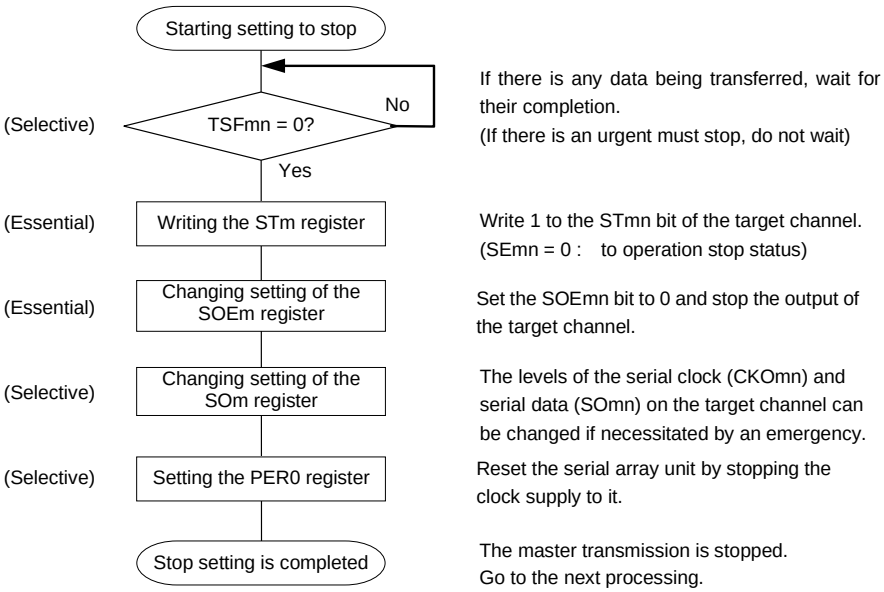
Figure 11-64. Initial Setting Procedure for Slave Transmission/Reception



Caution Be sure to set transmit data to the SIOp register before the clock from the master is started.

Downloaded from Arrow.com.

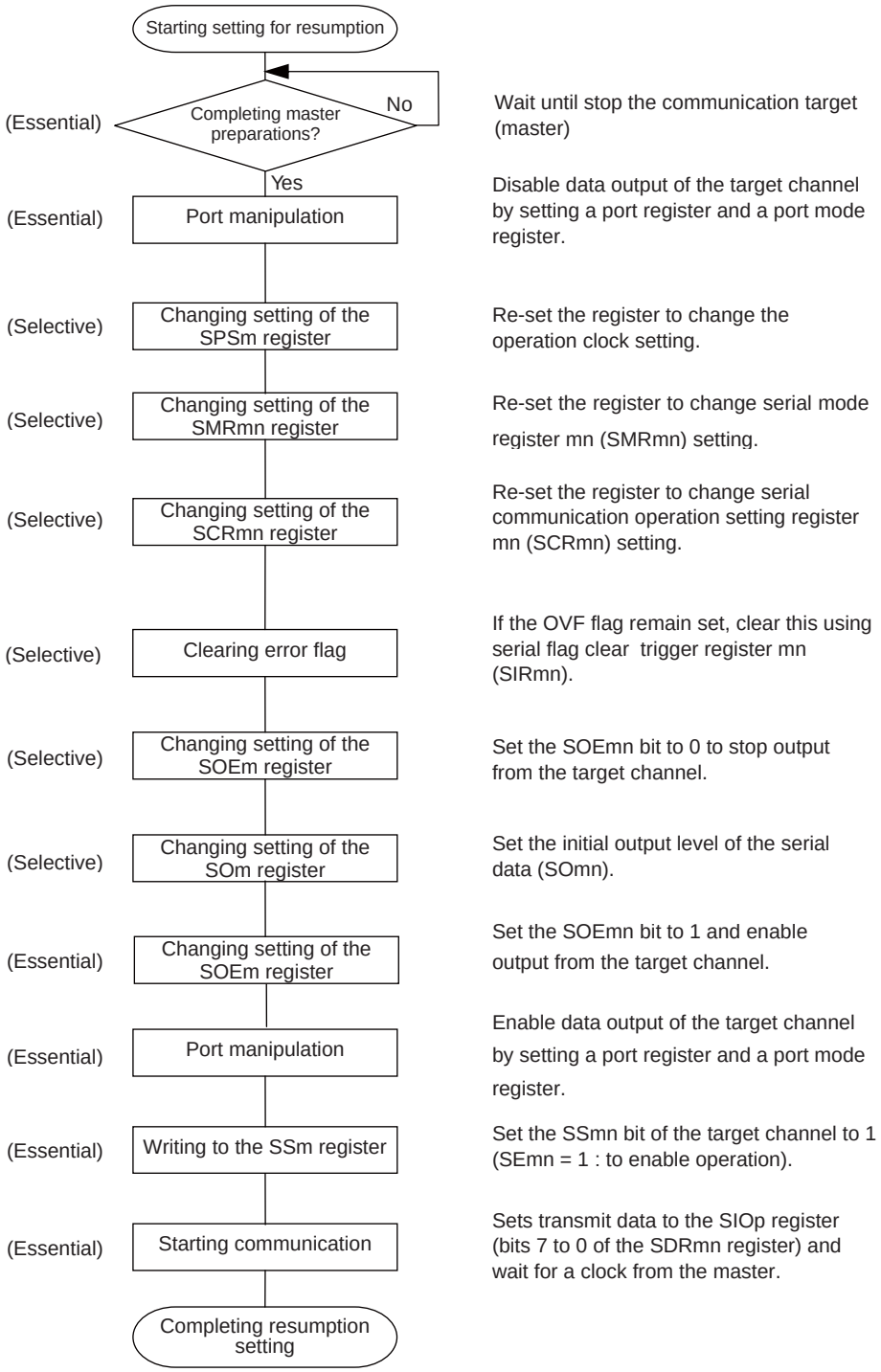
Figure 11-65. Procedure for Stopping Slave Transmission/Reception



<R>

Downloaded from Arrow.com.

Figure 11-66. Procedure for Resuming Slave Transmission/Reception



Cautions 1. Be sure to set transmit data to the SIOp register before the clock from the master is started.

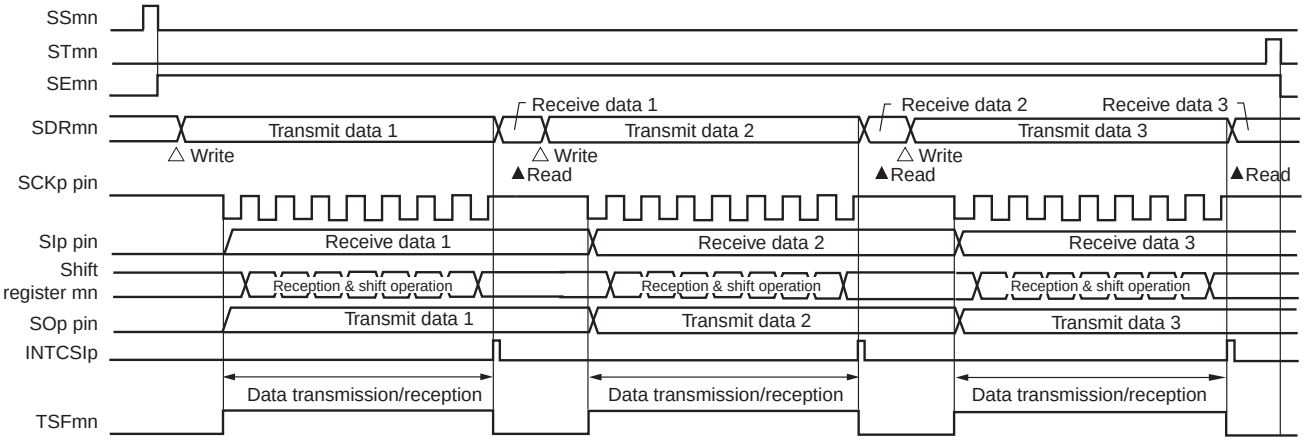
2. If PER0 is rewritten while stopping the master transmission and the clock supply is stopped, wait until the transmission target (master) stops or transmission finishes, and then perform initialization instead of restarting the transmission.

Downloaded from Arrow.com.

(3) Processing flow (in single-transmission/reception mode)

Figure 11-67. Timing Chart of Slave Transmission/Reception (in Single-Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)

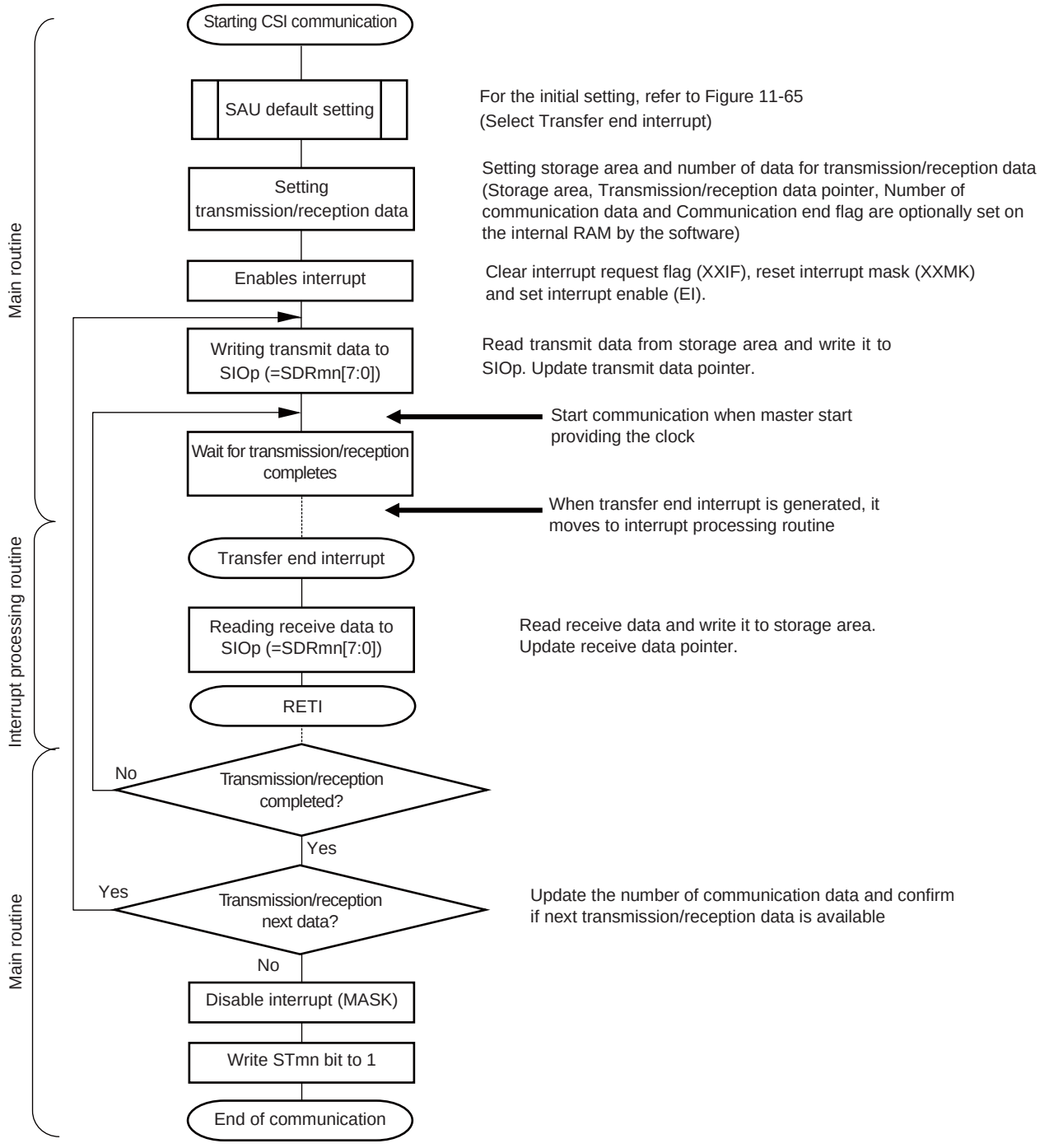
<R>



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 11, 20), mn = 00, 01, 03, 10

Downloaded from Arrow.com.

<R> **Figure 11-68. Flowchart of Slave Transmission/Reception (in Single- Transmission/Reception Mode)**



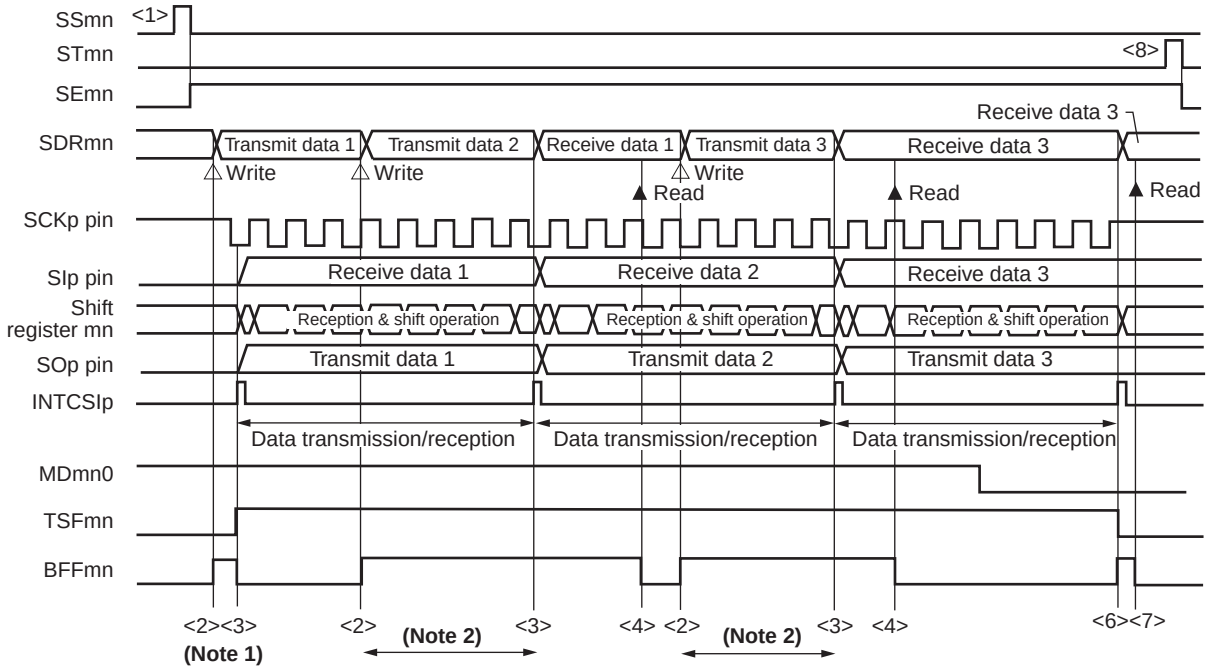
Caution Be sure to set transmit data to the SIOp register before the clock from the master is started.

Downloaded from Arrow.com.

(4) Processing flow (in continuous transmission/reception mode)

Figure 11-69. Timing Chart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)

<R>



Notes 1. If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

2. The transmit data can be read by reading the SDRmn register during this period. At this time, the transfer operation is not affected.

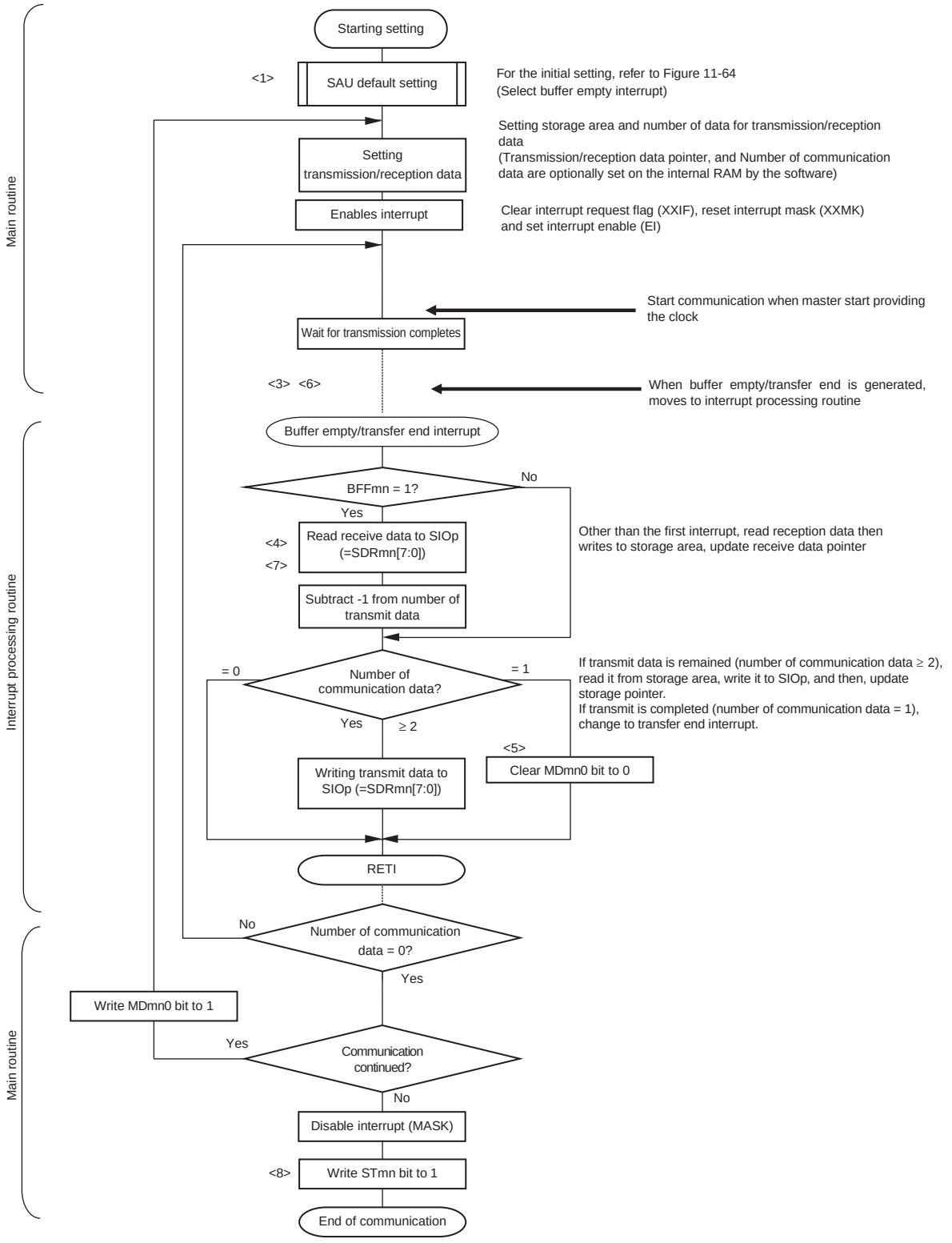
Caution The MDmn0 bit of serial mode register mn (SMRmn) can be rewritten even during operation. However, rewrite it before transfer of the last bit is started, so that it has been rewritten before the transfer end interrupt of the last transmit data.

Remarks 1. <1> to <8> in the figure correspond to <1> to <8> in **Figure 11-70 Flowchart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)**.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 11, 20), mn = 00, 01, 03, 10

Downloaded from Arrow.com.

<R> **Figure 11-70. Flowchart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)**



Caution Be sure to set transmit data to the SIOp register before the clock from the master is started.

Remark <1> to <8> in the figure correspond to <1> to <8> in **Figure 11-69 Timing Chart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)**.

Downloaded from Arrow.com.

11.5.7 SNOOZE mode function

<R> The SNOOZE mode makes the CSI perform reception operations upon SCK00 pin input detection while in the STOP mode. Normally the CSI stops communication in the STOP mode. However, using the SNOOZE mode enables the CSI to perform reception operations without CPU operation upon detection of the SCK00 pin input. Only CSI00 can be set to the SNOOZE mode.

When using the CSI in SNOOZE mode, make the following setting before switching to the STOP mode (see **Figure 11-72 Flowchart of SNOOZE Mode Operation (once startup)** and **Figure 11-74 Flowchart of SNOOZE Mode Operation (continuous startup)**).

• When using the SNOOZE mode function, set the SWC0 bit of serial standby control register 0 (SSC0) to 1 just before switching to the STOP mode. After the initial setting has been completed, set the SS01 bit of serial channel start register 0 (SS0) to 1.

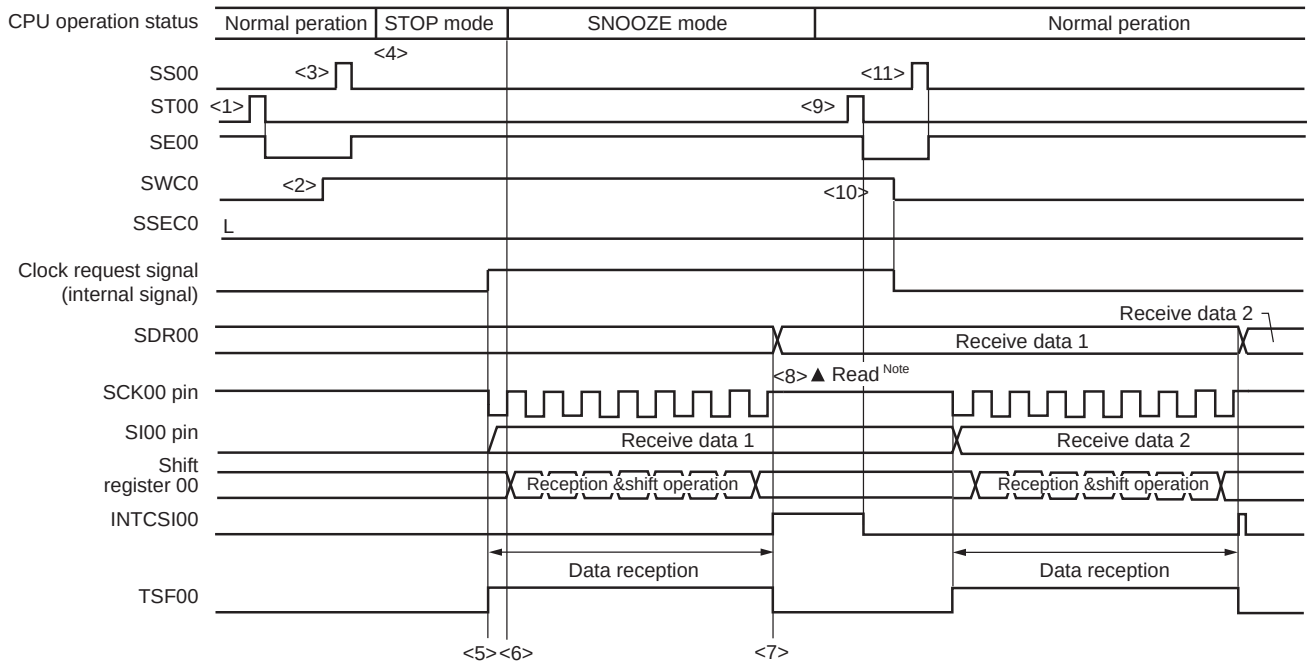
After a transition to the STOP mode, the CSI starts reception operations upon detection of an edge of the SCK00 pin.

Cautions 1. The SNOOZE mode can only be specified when the high-speed on-chip oscillator clock is selected for f_{CLK}.

2. The maximum transfer rate when using CSI00 in the SNOOZE mode is 1 Mbps.

(1) SNOOZE mode operation (once startup)

Figure 11-71. Timing Chart of SNOOZE Mode Operation (once startup) (Type 1: DAP00 = 0, CKP00 = 0)



<R> **Note** Only read received data while SWC0 = 1 and before the next edge of the SCK00 pin input is detected.

<R> **Cautions 1. Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the ST00 bit to 1 (clear the SE00 bit, and stop the operation).**
And after completion the receive operation, also clearing SWC0 bit to 0 (SNOOZE mode release).

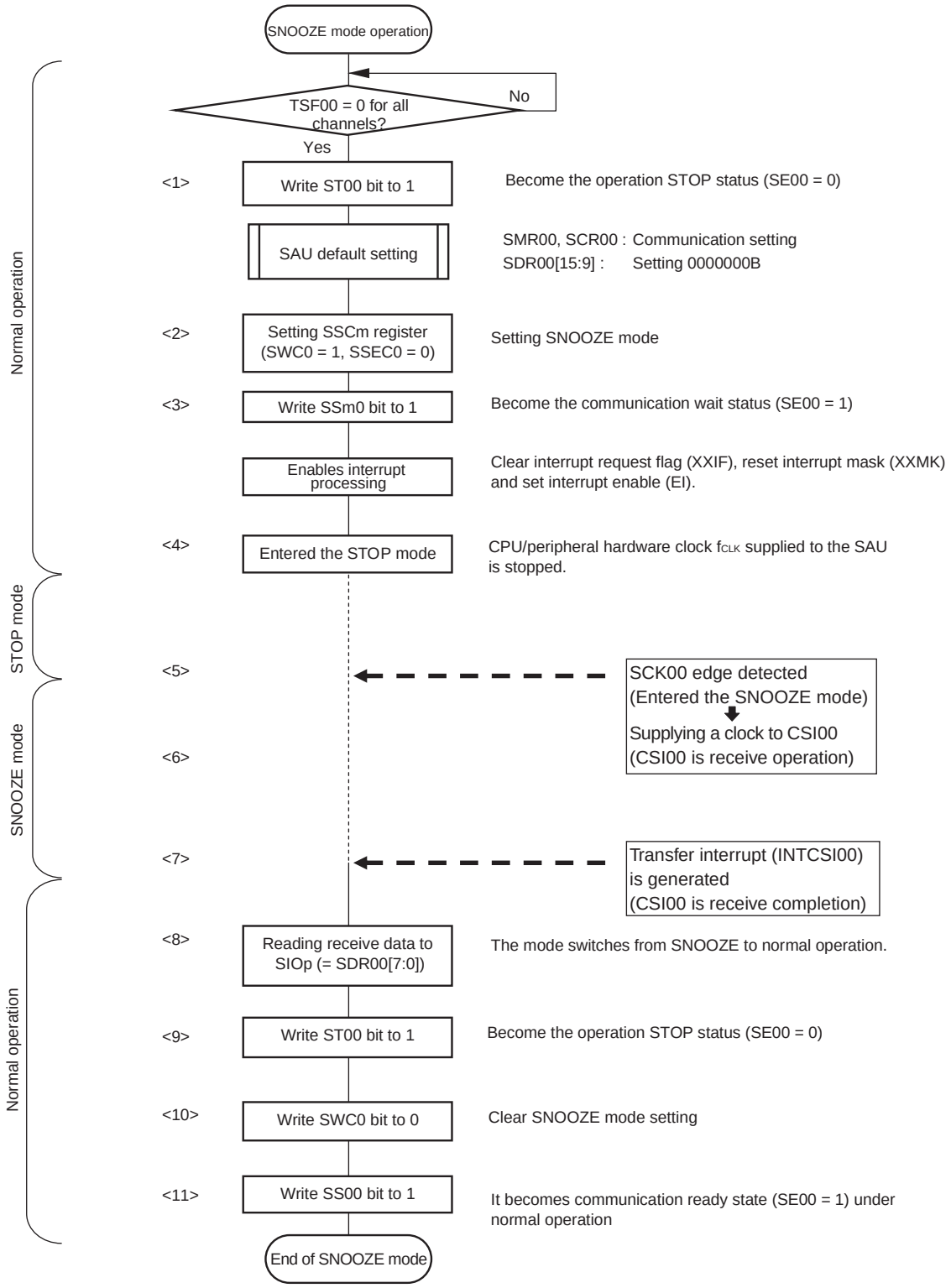
<R> **2. When SWC0 = 1, the BFF01 and OVF01 flags will not change.**

Remark <1> to <11> in the figure correspond to <1> to <11> in **Figure 11-72 Flowchart of SNOOZE Mode Operation (once startup)**.

Downloaded from Arrow.com.

Figure 11-72. Flowchart of SNOOZE Mode Operation (once startup)

<R>

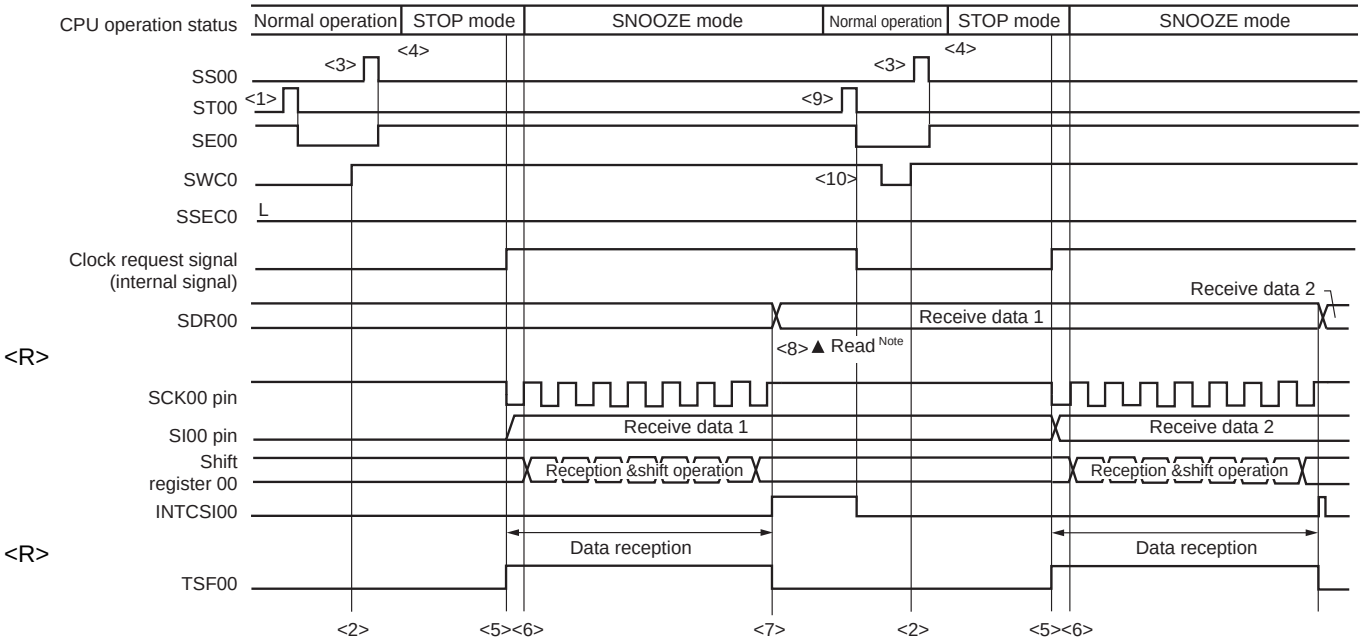


Remark <1> to <11> in the figure correspond to <1> to <11> in **Figure 11-71 Timing Chart of SNOOZE Mode Operation (once startup)**.

Downloaded from Arrow.com.

(2) SNOOZE mode operation (continuous startup)

Figure 11-73. Timing Chart of SNOOZE Mode Operation (continuous startup) (Type 1: DAP00 = 0, CKP00 = 0)



<R> **Note** Only read received data while SWC0 = 1 and before the next edge of the SCK00 pin input is detected.

<R> **Cautions 1. Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the ST00 bit to 1 (clear the SE00 bit, and stop the operation).**
And after completion the receive operation, also clearing SWC0 bit to 0 (SNOOZE release).

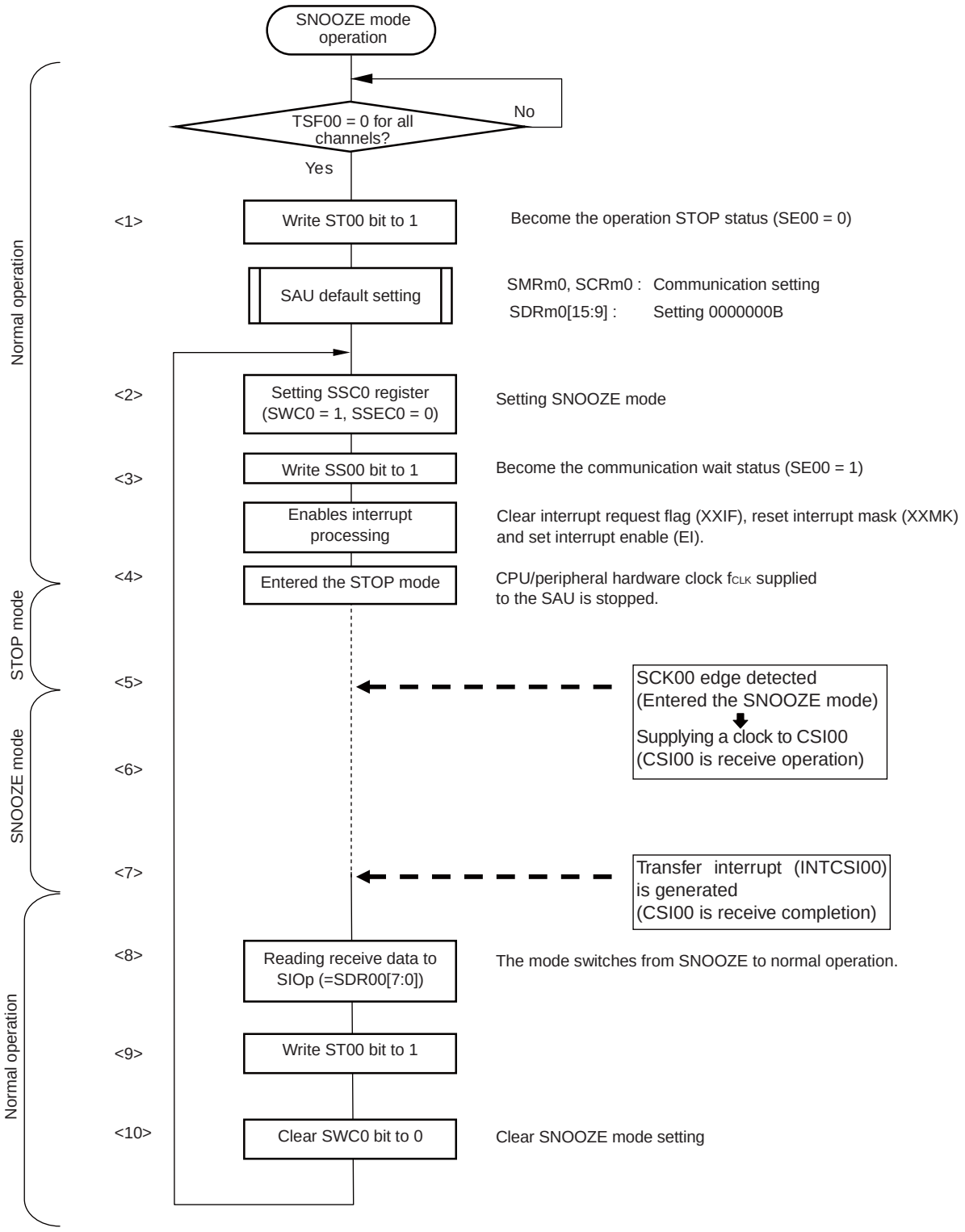
<R> **2. When SWC0 = 1, the BFF01 and OVF01 flags will not change.**

Remark <1> to <10> in the figure correspond to <1> to <10> in **Figure 11-74 Flowchart of SNOOZE Mode Operation (continuous startup)**.

Downloaded from Arrow.com.

Figure 11-74. Flowchart of SNOOZE Mode Operation (continuous startup)

<R>



Remark <1> to <10> in the figure correspond to <1> to <10> in **Figure 11-73 Timing Chart of SNOOZE Mode Operation (continuous startup)**.

Downloaded from Arrow.com.

11.5.8 Calculating transfer clock frequency

The transfer clock frequency for 3-wire serial I/O (CSI00, CSI01, CSI11, CSI20) communication can be calculated by the following expressions.

(1) Master

(Transfer clock frequency) = {Operation clock (f_{MCK}) frequency of target channel} ÷ (SDRmn[15:9] + 1) ÷ 2 [Hz]

(2) Slave

(Transfer clock frequency) = {Frequency of serial clock (SCK) supplied by master}[Note] [Hz]

Note The permissible maximum transfer clock frequency is f_{MCK}/6.

Remark The value of SDRmn[15:9] is the value of bits 15 to 9 of serial data register mn (SDRmn) (0000000B to 1111111B) and therefore is 0 to 127.

The operation clock (f_{MCK}) is determined by serial clock select register m (SPSm) and bit 15 (CKSmn) of serial mode register mn (SMRmn).

Downloaded from Arrow.com.

Table 11-2. Selection of Operation Clock For 3-Wire Serial I/O

SMRmn Register	SPS0 Register								Operation Clock (f_{MCK})[Note]	
CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		f_{CLK} = 20 MHz
0	X	X	X	X	0	0	0	0	f_{CLK}	20 MHz
	X	X	X	X	0	0	0	1	$f_{CLK}/2$	10 MHz
	X	X	X	X	0	0	1	0	$f_{CLK}/2^2$	5 MHz
	X	X	X	X	0	0	1	1	$f_{CLK}/2^3$	2.5 MHz
	X	X	X	X	0	1	0	0	$f_{CLK}/2^4$	1.25 MHz
	X	X	X	X	0	1	0	1	$f_{CLK}/2^5$	625 kHz
	X	X	X	X	0	1	1	0	$f_{CLK}/2^6$	312.5 kHz
	X	X	X	X	0	1	1	1	$f_{CLK}/2^7$	156.2 kHz
	X	X	X	X	1	0	0	0	$f_{CLK}/2^8$	78.1 kHz
	X	X	X	X	1	0	0	1	$f_{CLK}/2^9$	39.1 kHz
	X	X	X	X	1	0	1	0	$f_{CLK}/2^{10}$	19.5kHz
	X	X	X	X	1	0	1	1	$f_{CLK}/2^{11}$	9.77 kHz
	X	X	X	X	1	1	0	0	$f_{CLK}/2^{12}$	4.88 kHz
	X	X	X	X	1	1	0	1	$f_{CLK}/2^{13}$	2.44 kHz
	X	X	X	X	1	1	1	0	$f_{CLK}/2^{14}$	1.22 kHz
	X	X	X	X	1	1	1	1	$f_{CLK}/2^{15}$	610 Hz
1	0	0	0	0	X	X	X	X	f_{CLK}	20 MHz
	0	0	0	1	X	X	X	X	$f_{CLK}/2$	10 MHz
	0	0	1	0	X	X	X	X	$f_{CLK}/2^2$	5 MHz
	0	0	1	1	X	X	X	X	$f_{CLK}/2^3$	2.5 MHz
	0	1	0	0	X	X	X	X	$f_{CLK}/2^4$	1.25 MHz
	0	1	0	1	X	X	X	X	$f_{CLK}/2^5$	625 kHz
	0	1	1	0	X	X	X	X	$f_{CLK}/2^6$	312.5 kHz
	0	1	1	1	X	X	X	X	$f_{CLK}/2^7$	156.2 kHz
	1	0	0	0	X	X	X	X	$f_{CLK}/2^8$	78.1 kHz
	1	0	0	1	X	X	X	X	$f_{CLK}/2^9$	39.1 kHz
	1	0	1	0	X	X	X	X	$f_{CLK}/2^{10}$	19.5 kHz
	1	0	1	1	X	X	X	X	$f_{CLK}/2^{11}$	9.77 kHz
	1	1	0	0	X	X	X	X	$f_{CLK}/2^{12}$	4.88 kHz
	1	1	0	1	X	X	X	X	$f_{CLK}/2^{13}$	2.44 kHz
	1	1	1	0	X	X	X	X	$f_{CLK}/2^{14}$	1.22 kHz
	1	1	1	1	X	X	X	X	$f_{CLK}/2^{15}$	610 Hz

Note When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), do so after having stopped (serial channel stop register m (STm) = 000FH) the operation of the serial array unit (SAU).

Remarks 1. X: don't care

2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00, 01, 03, 10

Downloaded from Arrow.com.

11.5.9 Procedure for processing errors that occurred during 3-wire serial I/O (CSI00, CSI01, CSI11, CSI20) communication

The procedure for processing errors that occurred during 3-wire serial I/O (CSI00, CSI01, CSI11, CSI20) communication is described in Figure 11-75.

Figure 11-75. Processing Procedure in Case of Overrun Error

Software Manipulation	Hardware Status	Remark
Reads serial data register mn (SDRmn). →	The BFFmn bit of the SSRmn register is set to 0 and channel n is enabled to receive data.	This is to prevent an overrun error if the next reception is completed during error processing.
Reads serial status register mn (SSRmn).		Error type is identified and the read value is used to clear error flag.
Writes 1 to serial flag clear trigger register mn (SIRmn). →	Error flag is cleared.	Error can be cleared only during reading, by writing the value read from the SSRmn register to the SIRmn register without modification.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00, 01, 03, 10

Downloaded from Arrow.com.

11.6 Operation of UART (UART0 to UART2) Communication

This is a start-stop synchronization function using two lines: serial data transmission (TXD) and serial data reception (RXD) lines. By using these two communication lines, each data frame, which consists of a start bit, data, parity bit, and stop bit, is transferred asynchronously (using the internal baud rate) between the microcontroller and the other communication party. Full-duplex asynchronous UART communication can be performed by using a channel dedicated to transmission (an even-numbered channel) and a channel dedicated to reception (an odd-numbered channel).

[Data transmission/reception]

- Data length of 7, 8, or 9 bits[Note 1]
- Select the MSB/LSB first
- Level setting of transmit/receive data and select of reverse (selecting whether to reverse the level)
- Parity bit appending and parity check functions
- Stop bit appending and stop bit check functions

[Interrupt function]

- Transfer end interrupt/buffer empty interrupt
- Error interrupt in case of framing error, parity error, or overrun error

[Error detection flag]

- Framing error, parity error, or overrun error

UART0 is compatible with SNOOZE mode. When RxD0 pin input is detected while in the STOP mode, the SNOOZE mode makes data reception that does not require the CPU possible.

<R> **Note 1.** Only UART0 can be specified for the 9-bit data length.

20- or 24-pin products

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I^2C
0	0	CSI00	UART0	IIC00[Note 2]
	1	CSI01[Note 2]		IIC01[Note 2]

30-pin products

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I^2C
0	0	CSI00	UART0	IIC00[Note 2]
	1	–		–
	2	–	UART1[Note 2]	–
	3	CSI11[Note 2]		IIC11[Note 2]
1	0	CSI20[Note 2]	UART2[Note 2]	IIC20[Note 2]
	1	–		–

Note 2. Provided in the R5F102 products only.

<R> Select any function for each channel. Only the selected function is possible. If UART0 is selected for channels 0 and 1 of unit 0, for example, these channels cannot be used for CSI00 and CSI01. At this time, however, channel 2, 3, or other channels of the same unit can be used for a function other than UART0, such as CSI10, UART1, and IIC10.

(**Caution** is listed on the next page.)

Downloaded from Arrow.com.

Caution When using a serial array unit for UART, both the transmitter side (even-numbered channel) and the receiver side (odd-numbered channel) can only be used for UART.

UART performs the following two types of communication operations.

- UART transmission (See **11.6.1**.)
- UART reception (See **11.6.2**.)

11.6.1 UART transmission

<R> UART transmission is an operation to transmit data from the RL78 microcontroller to another device asynchronously (start-stop synchronization).

Of the two channels used for UART, the even-numbered channel is used for UART transmission.

UART	UART0	UART1	UART2
Target channel	Channel 0 of SAU0	Channel 2 of SAU0	Channel 0 of SAU1
Pins used	TxD0	TxD1	TxD2
Interrupt	INTST0	INTST1	INTST2
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.		
Error detection flag	None		
Transfer data length	7, 8, or 9 bits Note 1		
Transfer rate Note 2	Max. $f_{MCK}/6$ [bps] (SDRmn[15:9] = 2 or greater), Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [bps] Note		
Data phase	Non-inverted output (default: high level) Inverted output (default: low level)		
Parity bit	The following selectable • No parity bit • Appending 0 parity • Appending even parity • Appending odd parity		
Stop bit	The following selectable • Appending 1 bit • Appending 2 bits		
Data direction	MSB or LSB first		

<R> **Notes 1.** Only UART0 can be specified for the 9-bit data length.

<R> **2.** Use this operation within a range that satisfies the conditions above and the peripheral function characteristics in the electrical specifications (see **CHAPTER 28 ELECTRICAL SPECIFICATIONS (A, D: T_A = -40 to +85°C)** or **CHAPTER 29 ELECTRICAL SPECIFICATIONS (G: T_A = -40 to +105°C)**).

Remarks 1. f_{MCK}: Operation clock frequency of target channel
f_{CLK}: System clock frequency

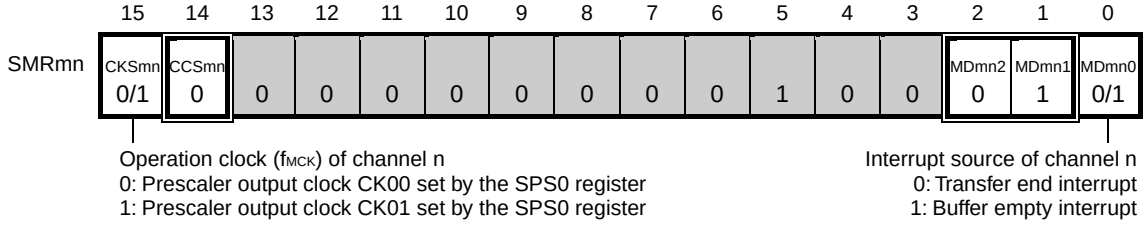
2. m: Unit number (m = 0, 1) n: Channel number (n = 0, 2), mn = 00, 02, 10

Downloaded from Arrow.com.

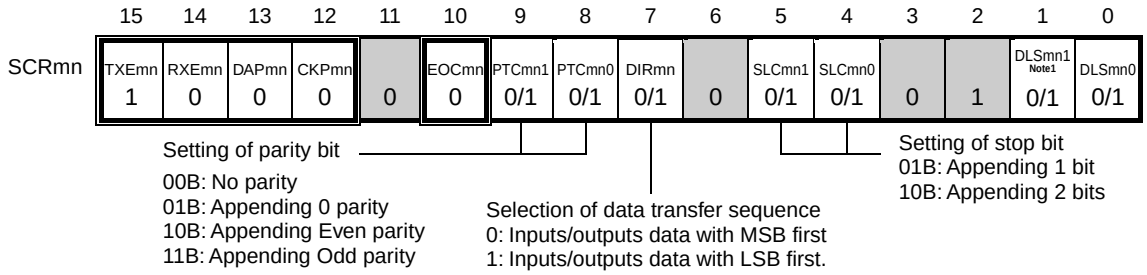
(1) Register setting

Figure 11-76. Example of Contents of Registers for UART Transmission (UART0 to UART2) (1/2)

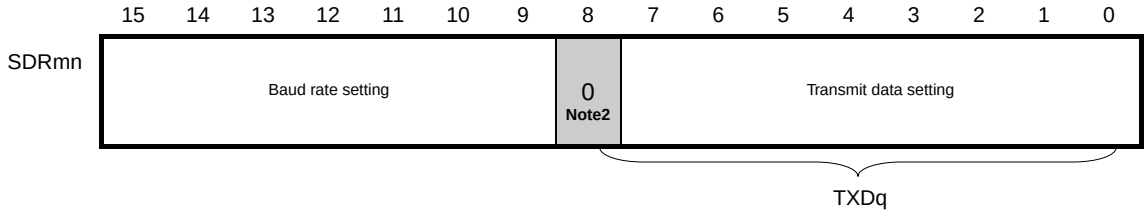
(a) Serial mode register mn (SMRmn)



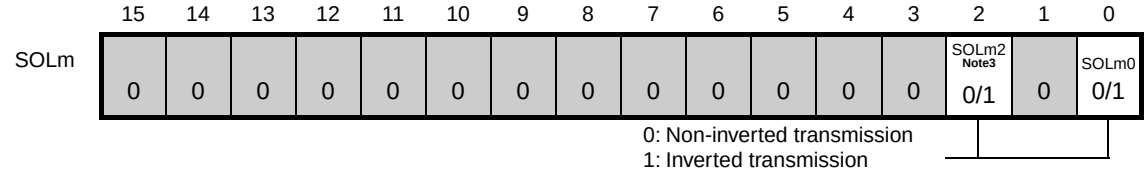
(b) Serial communication operation setting register mn (SCRmn)



(c) Serial data register mn (SDRmn) (lower 8 bits: TXDq)



(d) Serial output level register m (SOLm) ... Sets only the bits of the target channel.



Notes 1. Provided only in SCR00 register (UART0) only. For SCR02 and SCR10 registers, this bit is fixed to 1.

2. When performs 9-bit communication (by setting the DLS001 and DLS000 bits of the SMR00 register to 1), bits 0 to 8 of the SDR00 register are used as the transmission data specification area. 9-bit communication is available only in UART0.

3. Provided only in 30-pin product serial array unit 0.

Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), q: UART number (q = 0 to 2), mn = 00, 02, 10

2. □ : Setting is fixed in the CSI master transmission mode, ▒ : Setting disabled (set to the initial value)
0/1: Set to 0 or 1 depending on the usage of the user

Downloaded from Arrow.com.

Figure 11-76. Example of Contents of Registers for UART Transmission (UART0 to UART2) (2/2)

(e) Serial output register m (SOm) ... Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	1	1	CKOm1 ×	CKOm0 ×	0	0	0	0	SO03 Note 1 ×	SO02 Note 1 0/1 Note 2	SO01 ×	SOm0 0/1 Note 2

0: Serial data output value is "0"
1: Serial data output value is "1"

(f) Serial output enable register m (SOEm) ... Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	SOE03 Note 1 ×	SOE02 Note 1 0/1	SOE01 ×	SOEm0 0/1

(g) Serial channel start register m (SSm) ... Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SS03 Note 1 ×	SS02 Note 1 0/1	SSm1 ×	SSm0 0/1

Notes 1. Provided only in 30-pin product serial array unit 0.

2. Before transmission is started, be sure to set to 1 when the SOL00 bit of the target channel is set to 0, and set to 0 when the SOL00 bit of the target channel is set to 1. The value varies depending on the communication data during communication operation.

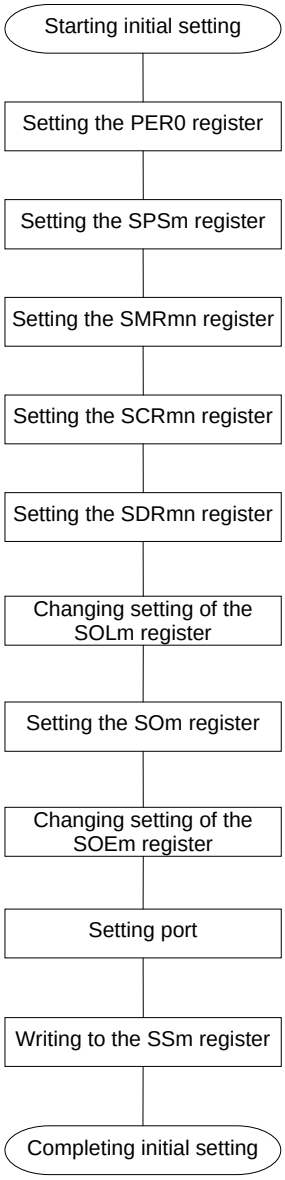
Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 10

2. ☐ : Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

Downloaded from Arrow.com.

(2) Operation procedure

Figure 11-77. Initial Setting Procedure for UART Transmission



Release the serial array unit from the reset status and start clock supply.

Set the operation clock.

Set an operation mode, etc.

Set a communication format.

Set a transfer baud rate (setting the transfer clock by dividing the operation clock (f_{MCK})).

Set an output data level.

Set the initial output level of the serial data (SOm0).

Set the SOEm0 bit to 1 and enable data output of the target channel.

Enable data output of the target channel by setting a port register and a port mode register.

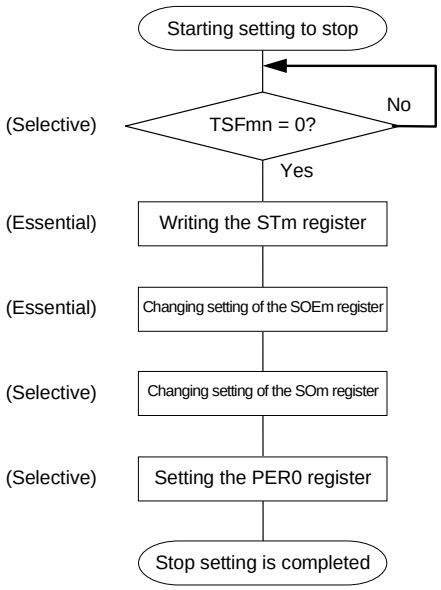
Set the SSm0 bit of the target channel to 1 and set the SEmn bit to 1 (to enable operation).

<R> Initial setting is completed.
Set transmit data to the SDRmn[7:0] bits (TXDq register) (8 bits) or the SDR00[8:0] bits (9 bits) and start communication.

Downloaded from Arrow.com.

Figure 11-78. Procedure for Stopping UART Transmission

<R>



If there is any data being transferred, wait for their completion.
(If there is an urgent must stop, do not wait)

Write 1 to the STmn bit of the target channel.
(SEmn = 0 : to operation stop status)

Set the SOEmn bit to 0 and stop the output of the target channel.

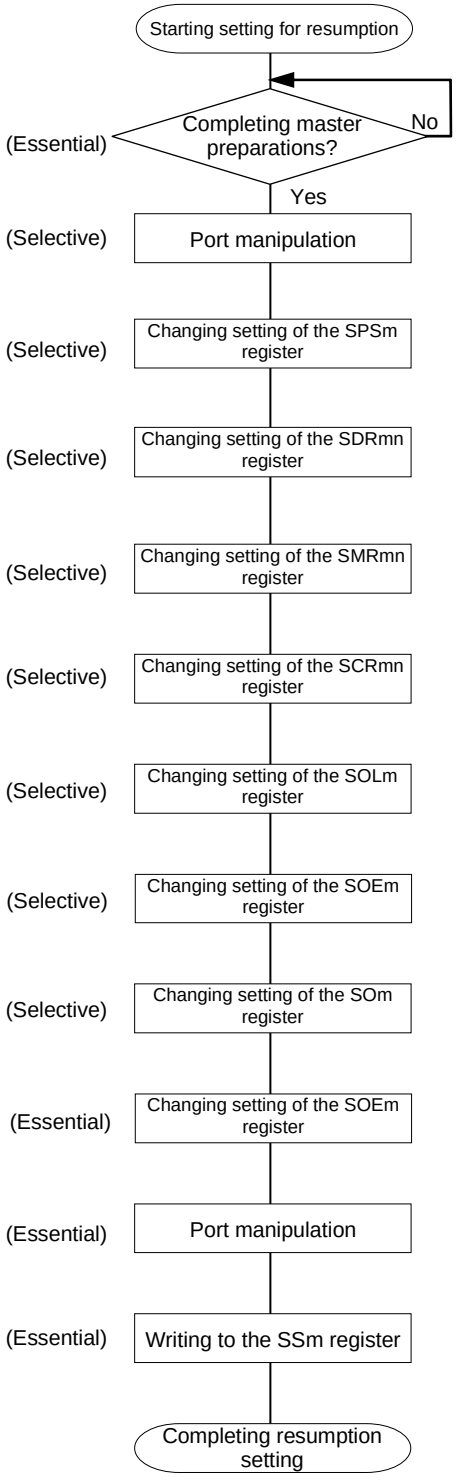
The levels of the serial clock (CKOmn) and serial data (SOmn) on the target channel can be changed if necessitated by an emergency.

Reset the serial array unit by stopping the clock supply to it.

The master transmission is stopped.
Go to the next processing.

Downloaded from Arrow.com.

Figure 11-79. Procedure for Resuming UART Transmission



Wait until stop the communication target or communication operation completed

Disable data output of the target channel by setting a port register and a port mode register.

Re-set the register to change the operation clock setting.

Re-set the register to change the transfer baud rate setting (setting the transfer clock by dividing the operation clock (f_{MCK})).

Re-set the register to change serial mode register mn (SMRmn) setting.

Re-set the register to change the serial communication operation setting register mn (SCRmn) setting.

Re-set the register to change serial output level register m (SOLm) setting.

Clear the SOEmn bit to 0 and stop output.

Set the initial output level of the serial data (SOmn).

Set the SOEmn bit to 1 and enable output.

Enable data output of the target channel by setting a port register and a port mode register.

Set the SSmn bit of the target channel to 1 and set the SEmn bit to 1 (to enable operation).

<R> Setting is completed
Set transmit data to the SDRmn[7:0] bits (TXDq register) (8 bits) or the SDR00[8:0] bits (9 bits) and start communication.

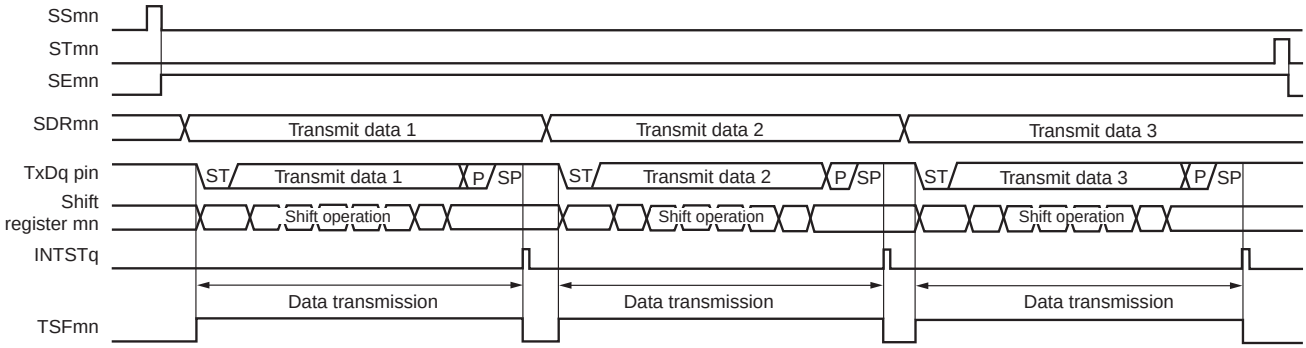
Remark If PER0 is rewritten while stopping the master transmission and the clock supply is stopped, wait until the transmission target stops or transmission finishes, and then perform initialization instead of restarting the transmission.

Downloaded from Arrow.com.

(3) Processing flow (in single-transmission mode)

Figure 11-80. Timing Chart of UART Transmission (in Single-Transmission Mode)

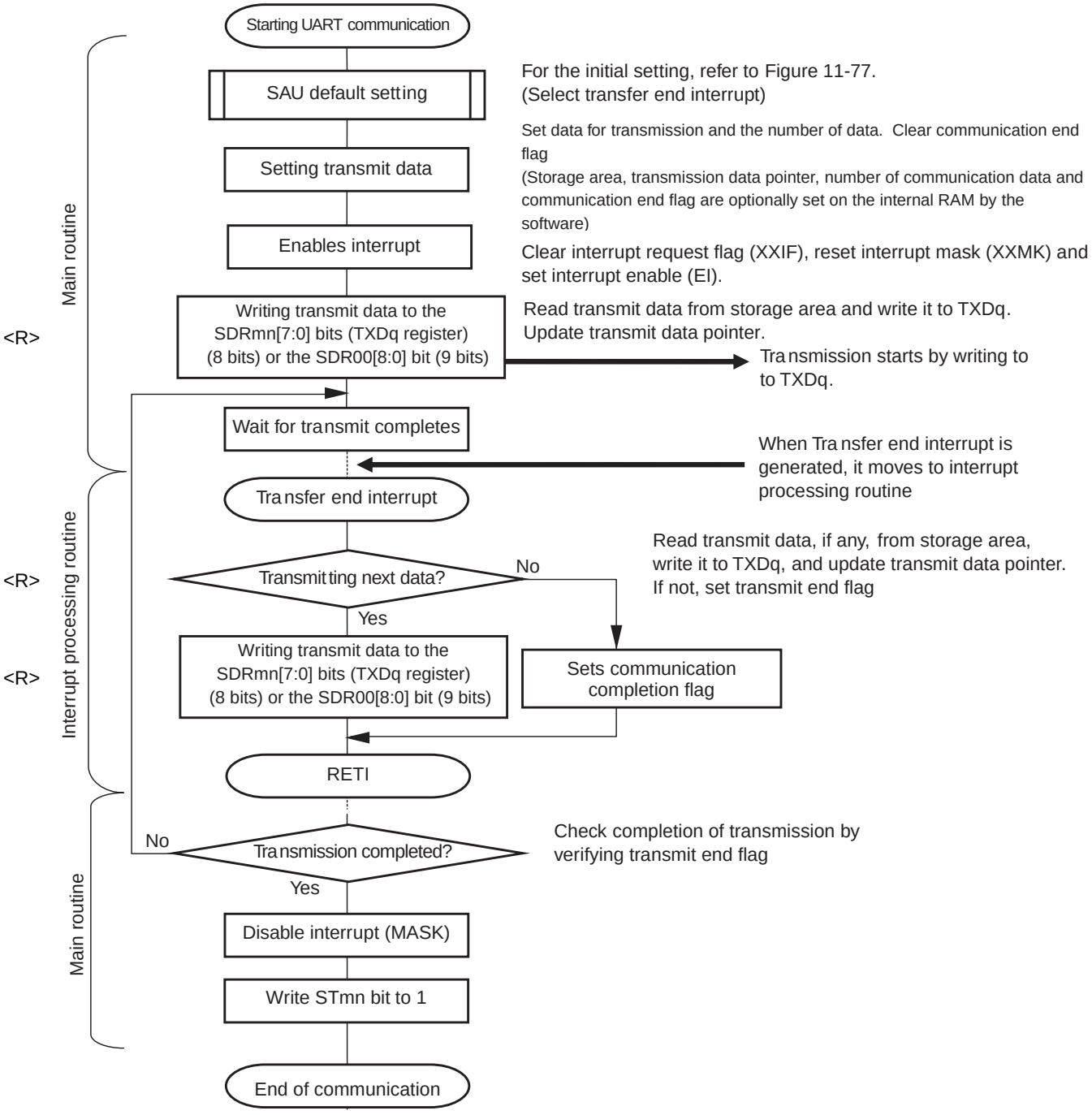
<R>



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), q: UART number (q = 0 to 2), mn = 00, 02, 10

Downloaded from Arrow.com.

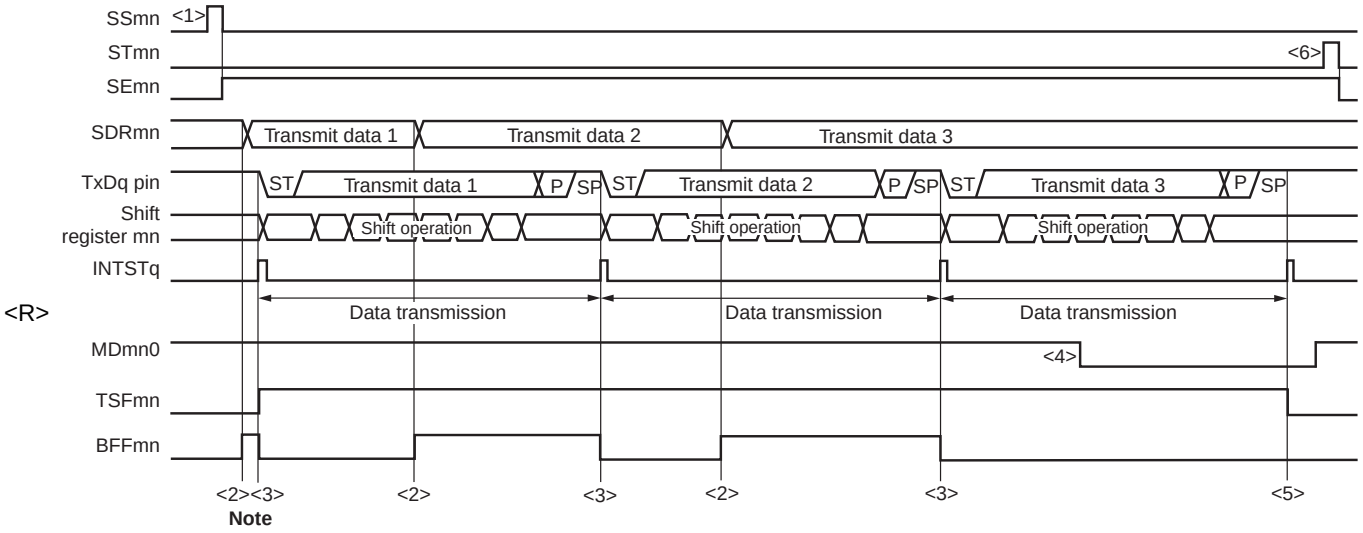
Figure 11-81. Flowchart of UART Transmission (in Single-Transmission Mode)



Downloaded from Arrow.com.

(4) Processing flow (in continuous transmission mode)

Figure 11-82. Timing Chart of UART Transmission (in Continuous Transmission Mode)



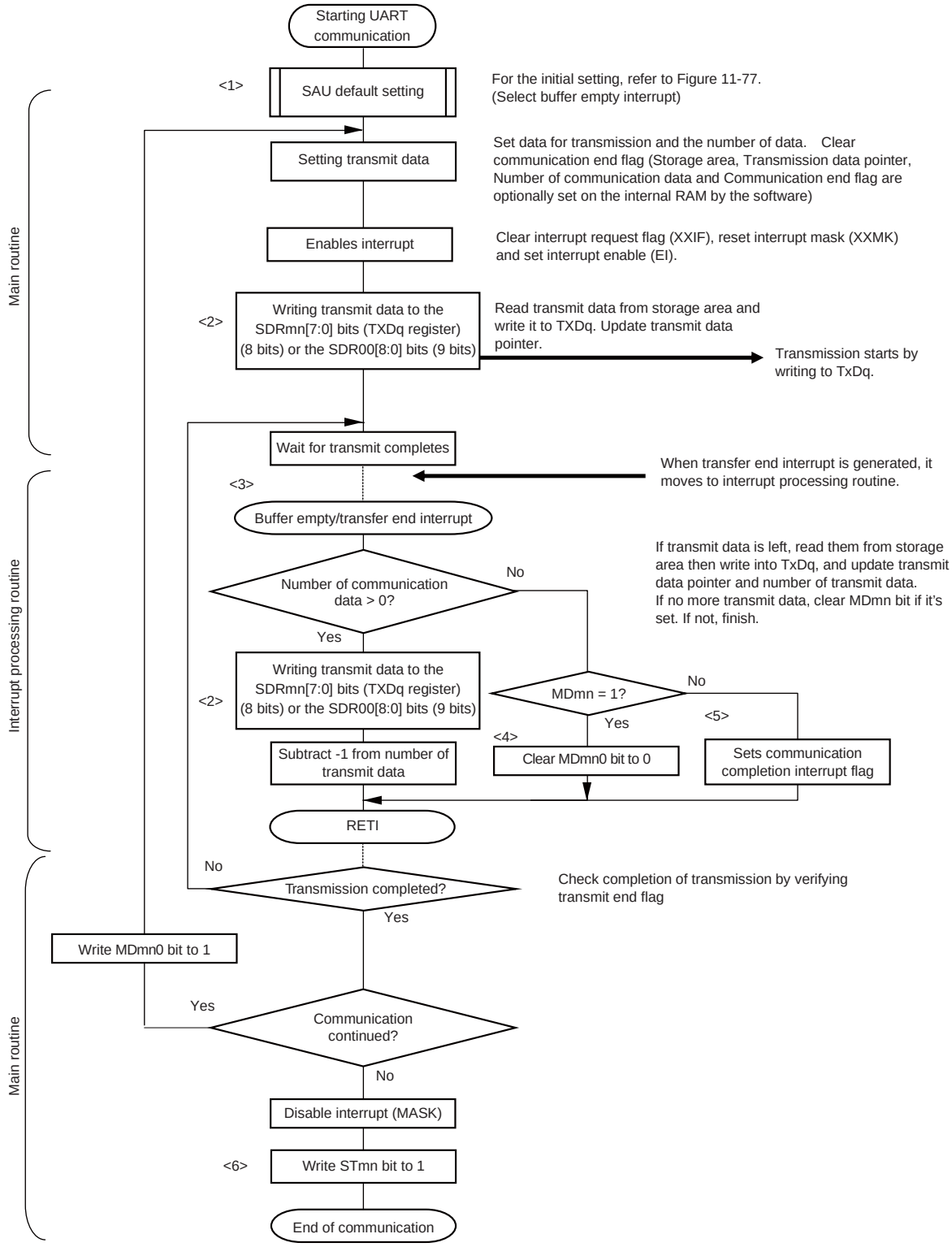
Note If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

Caution The MDmn0 bit of serial mode register mn (SMRmn) can be rewritten even during operation. However, rewrite it before transfer of the last bit is started, so that it will be rewritten before the transfer end interrupt of the last transmit data.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), q: UART number (q = 0 to 2), mn = 00, 02, 10

Downloaded from Arrow.com.

<R> **Figure 11-83. Flowchart of UART Transmission (in Continuous Transmission Mode)**



Remark <1> to <6> in the figure correspond to <1> to <6> in **Figure 11-82 Timing Chart of UART Transmission (in Continuous Transmission Mode)**.

Downloaded from Arrow.com.

11.6.2 UART reception

<R> UART reception is an operation wherein the RL78 microcontroller asynchronously receives data from another device (start-stop synchronization).

For UART reception, the odd-number channel of the two channels used for UART is used. The SMR register of both the odd- and even-numbered channels must be set.

UART	UART0	UART1	UART2
Target channel	Channel 1 of SAU0	Channel 3 of SAU0	Channel 1 of SAU1
Pins used	RxD0	RxD1	RxD2
Interrupt	INTSR0	INTSR1	INTSR2
	Transfer end interrupt only (setting the buffer empty interrupt is prohibited)		
Error interrupt	INTSRE0	INTSRE1	INTSRE2
Error detection flag	• Framing error detection flag (FEFmn) • Parity error detection flag (PEFmn) • Overrun error detection flag (OVFmn)		
Transfer data length	7, 8 or 9 bits[Note 1]		
Transfer rate[Note 2]	Max. $f_{MCK}/6$ [bps] (SDRmn [15:9] = 2 or more), Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [bps][Note]		
Data phase	Non-inverted output (default: high level) Inverted output (default: low level)		
Parity bit	The following selectable • No parity bit (no parity check) • No parity specified (0 parity) • Appending even parity • Appending odd parity		
<R> Stop Bit	Appending 1 bit		
Data direction	MSB or LSB first		

<R> **Notes 1.** Only UART0 can be specified for the 9-bit data length.

<R> **2.** Use this operation within a range that satisfies the conditions above and the peripheral characteristics in the electrical specifications (see **CHAPTER 28 ELECTRICAL SPECIFICATIONS (A, D: T_A = -40 to +85°C)** or **CHAPTER 29 ELECTRICAL SPECIFICATIONS (G: T_A = -40 to +105°C)**).

Remarks 1. f_{MCK}: Operation clock frequency of target channel
f_{CLK}: System clock frequency

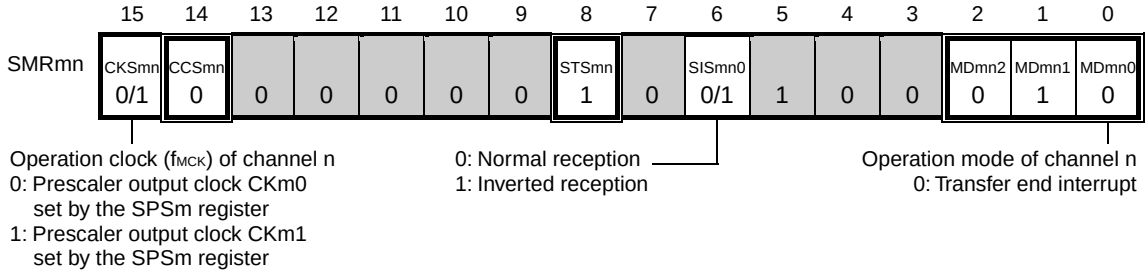
2. m: Unit number (m = 0, 1) n: Channel number (n = 1, 3), mn = 01, 03, 11

Downloaded from Arrow.com.

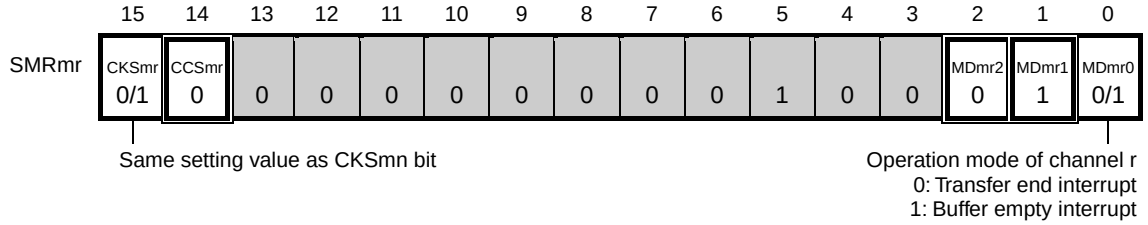
(1) Register setting

Figure 11-84. Example of Contents of Registers for UART Reception (UART0 to UART2) (1/2)

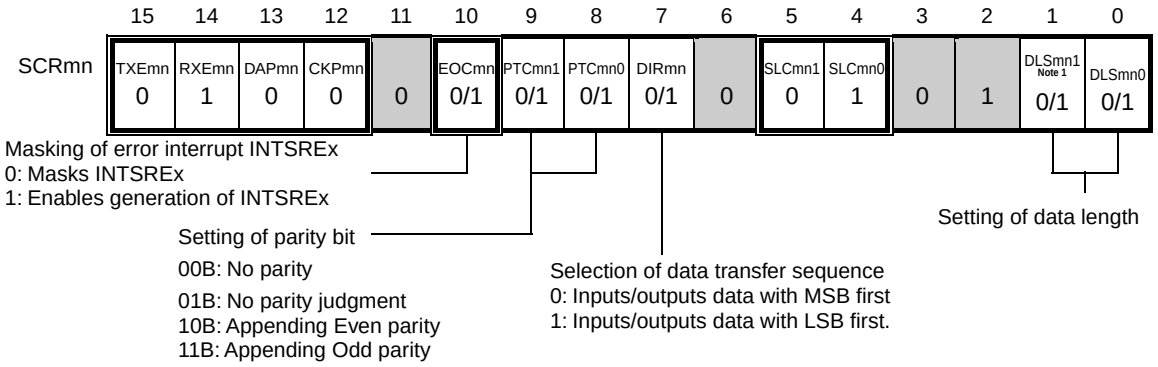
(a) Serial mode register mn (SMRmn)



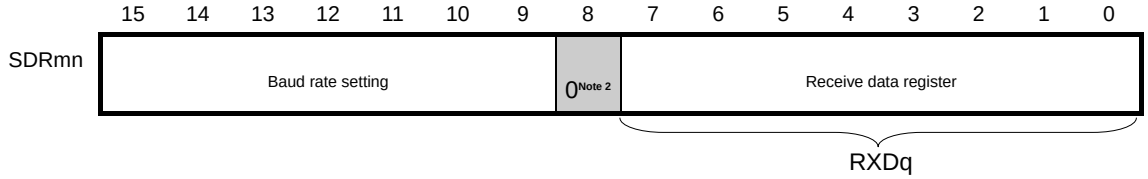
(b) Serial mode register mr (SMRmr)



(c) Serial communication operation setting register mn (SCRmn)



(d) Serial data register mn (SDRmn) (lower 8 bits: RXDq)



Notes 1. Provided only in SCR01 register (UART0) only. For SCR03 and SCR11 registers, this bit is fixed to 1.

2. When UART performs 9-bit communication, bits 0 to 8 of the SDR01 register are used as the transmission data specification area. 9-bit communication is available only in UART0.

Caution For the UART reception, be sure to set the SMRmr register of channel r to UART transmission mode that is to be paired with channel n.

Remarks 1. m: Unit number (m = 0, 1) n: Channel number (n = 1, 3), mn = 01, 03, 11
r: Channel number (r = n - 1) q: UART number (q = 0 to 2)

2. ☐: Setting is fixed in the UART master transmission mode, ▒: Setting disabled (set to the initial value)
0/1: Set to 0 or 1 depending on the usage of the user

Downloaded from Arrow.com.

Figure 11-84. Example of Contents of Registers for UART Reception (UART0 to UART2) (2/2)

(e) Serial output register m (SOm) ... The register that not used in this mode.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	1	1	CKOm1 ×	CKOm0 ×	0	0	0	0	SO03 Note ×	SO02 Note ×	SO01 ×	SOm0 ×

(f) Serial output enable register m (SOEm) ... The register that not used in this mode.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	SOE03 Note ×	SOE02 Note ×	SOE01 ×	SOEm0 ×

(g) Serial channel start register m (SSm) ... Sets only the bits of the target channel is 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SS03 Note 0/1	SS02 Note ×	SSm1 0/1	SSm0 ×

Note Provided only in 30-pin product serial array unit 0.

Remarks 1. m: Unit number (m = 0, 1)

2. ☐ : Setting disabled (set to the initial value)

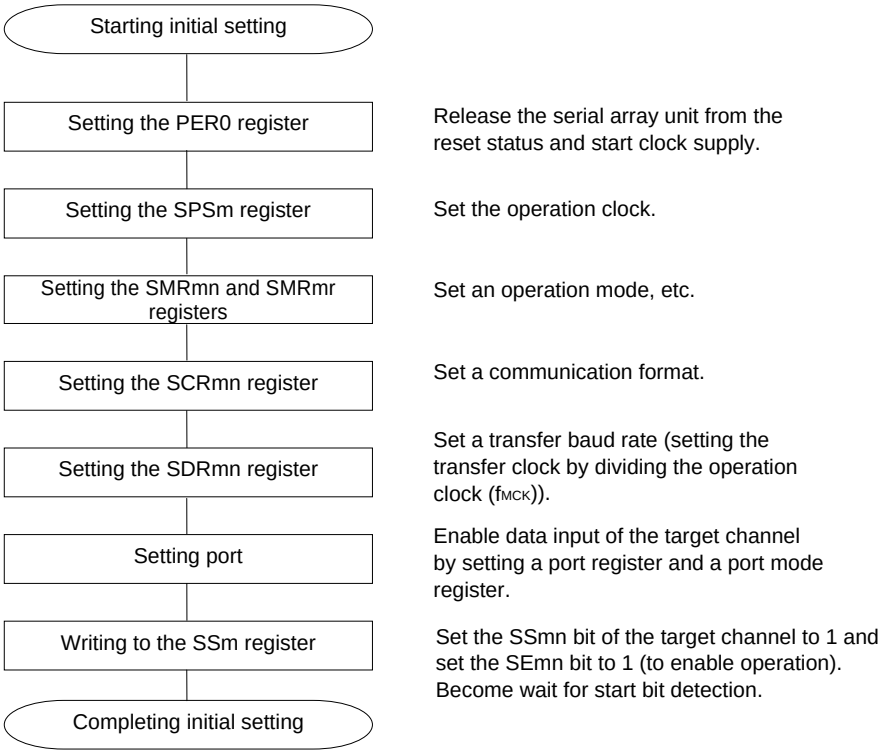
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)

0/1: Set to 0 or 1 depending on the usage of the user

Downloaded from Arrow.com.

(2) Operation procedure

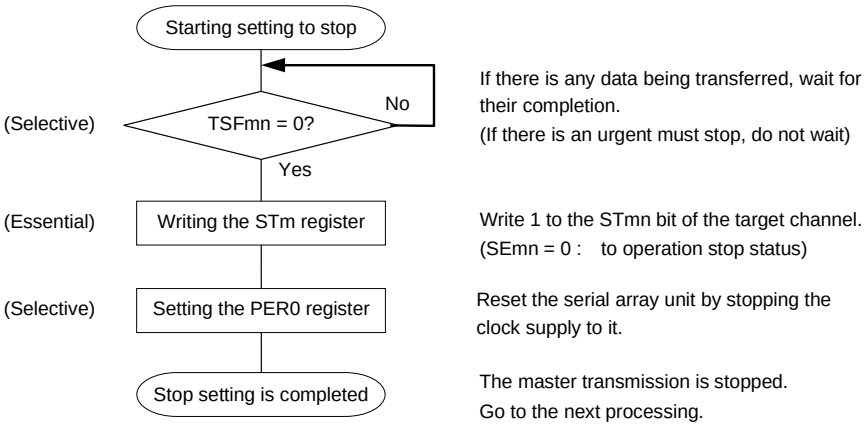
Figure 11-85. Initial Setting Procedure for UART Reception



Caution After setting the RXEmn bit of SCRmn register to 1, be sure to set SSmn to 0 after 4 or more f_{CLK} clocks have elapsed.

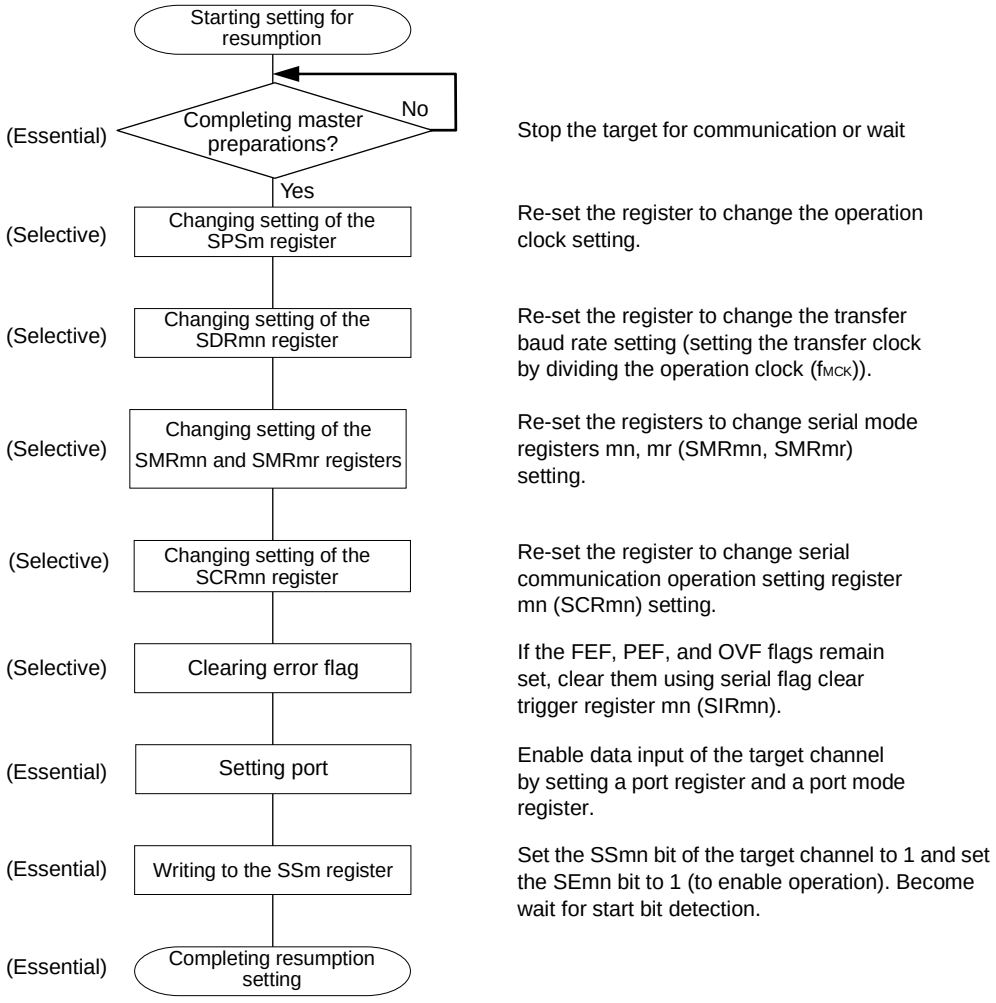
Figure 11-86. Procedure for Stopping UART Reception

<R>



Downloaded from Arrow.com.

Figure 11-87. Procedure for Resuming UART Reception



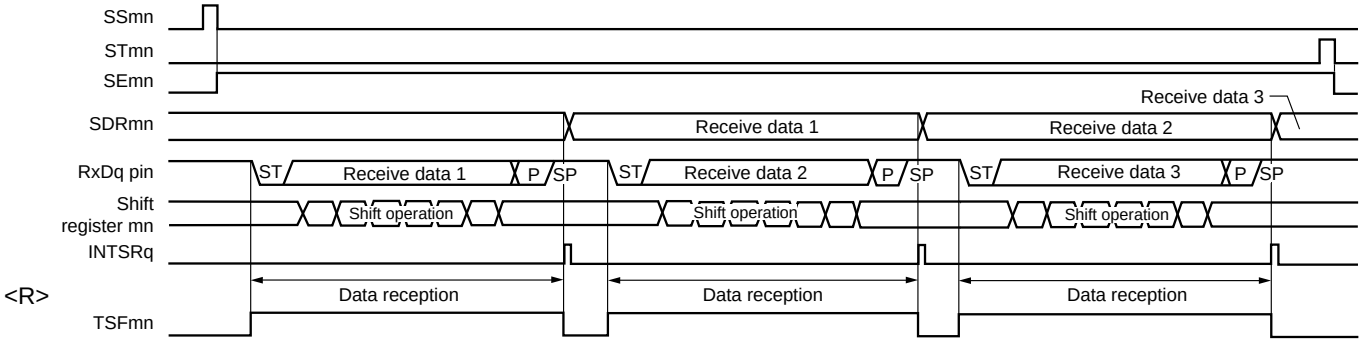
Caution After setting the RXEmn bit of SCRmn register to 1, be sure to set SSmn to 0 after 4 or more f_{MCK} clocks have elapsed.

Remark If PER0 is rewritten while stopping the master transmission and the clock supply is stopped, wait until the transmission target (slave) stops or transmission finishes, and then perform initialization instead of restarting the transmission.

Downloaded from Arrow.com.

(3) Processing flow

Figure 11-88. UART Reception Timing Chart

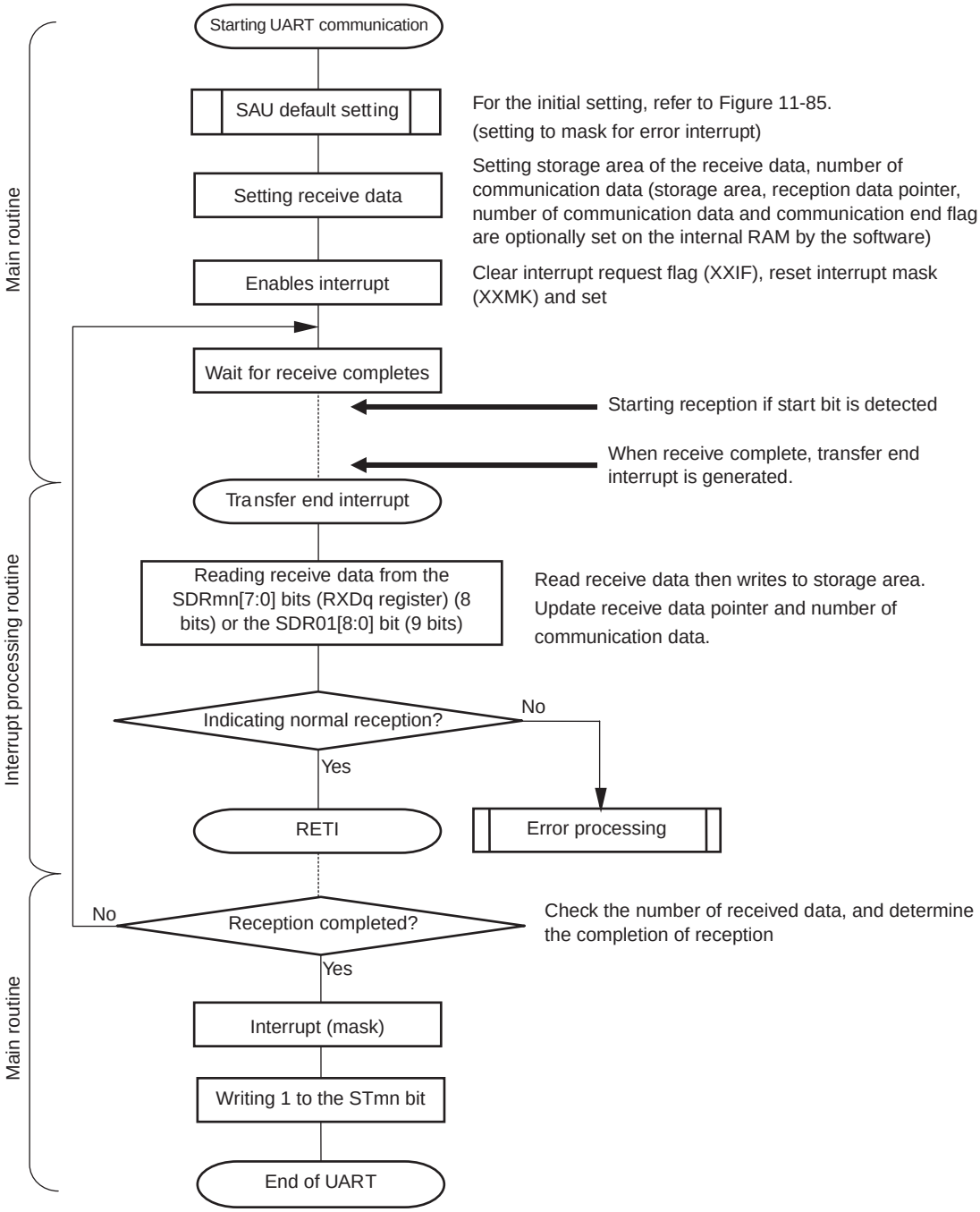


Remark m: Unit number (m = 0, 1) n: Channel number (n = 1, 3), mn = 01, 03, 11
r: Channel number (r = n - 1) q: UART number (q = 0 to 2)

Downloaded from Arrow.com.

<R>

Figure 11-89. Flowchart of UART Reception



Downloaded from Arrow.com.

11.6.3 SNOOZE mode function

SNOOZE mode makes UART operate reception by RxD0 pin input detection while the STOP mode. Normally the UART stops communication in the STOP mode. However, using the SNOOZE mode enables the UART to perform reception operations without CPU operation. Only UART0 can be set to the SNOOZE mode.

<R> When using UART0 in the SNOOZE mode, make the following settings before entering the STOP mode. (See **Figure 11-92 Flowchart of SNOOZE Mode Operation (EOC01 = 0, SSEC0 = 0/1 or EOC01 = 1, SSEC0 = 0)** and **Figure 11-94 Flowchart of SNOOZE Mode Operation (EOC01 = 1, SSEC0 = 1).**)

- In the SNOOZE mode, the baud rate setting for UART reception needs to be changed to a value different from that in normal operation. Set the SPS0 register and bits 15 to 9 of the SDR01 register with reference to Table 11-3.
- Set the EOC01 and SSEC0 bits. This is for enabling or stopping generation of an error interrupt (INTSRE0) when a communication error occurs.
- Set the SWC0 bit of serial standby control register 0 (SSC0) to 1 just before switching to the STOP mode. After the initial setting has completed, set the SS01 bit of serial channel start register 0 (SS0) to 1.

Upon detecting the edge of RxD0 (start bit input) after a transition was made to the STOP mode, UART reception is started.

Cautions 1. The SNOOZE mode can only be used when the high-speed on-chip oscillator clock (f_{IH}) is selected for f_{CLK}.

<R> **2. The transfer rate in the SNOOZE mode is only 4800 bps.**

<R> **3. When SWC0 = 1, UART0 can be used only when the reception operation is started in the STOP mode. When used simultaneously with another SNOOZE mode function or interrupt, if the reception operation is started in a state other than the STOP mode, such as those given below, data may not be received correctly and a framing error or parity error may be generated.**

- **When after the SWC0 bit has been set to 1, the reception operation is started before the STOP mode is entered**
- **When the reception operation is started while another function is in the SNOOZE mode**
- **When after returning from the STOP mode to normal operation due to an interrupt or other cause, the reception operation is started before the SWC0 bit is returned to 0**

<R> **4. If a parity error, framing error, or overrun error occurs while the SSEC0 bit is set to 1, the PEF01, FEF01, or OVF01 flag is not set and an error interrupt (INTSRE0) is not generated. Therefore, when the setting of SSEC0 = 1 is made, clear the PEF01, FEF01, or OVF01 flag before setting the SWC0 bit to 1 and read the value in bits 7 to 0 (RxD0 register) of the SDR01 register.**

Downloaded from Arrow.com.

<R>

Table 11-3. Baud Rate Setting for UART Reception in SNOOZE Mode

High-speed On-chip Oscillator (f_{IH})	Baud Rate for UART Reception in SNOOZE Mode			
	Baud Rate of 4800 bps			
	Operation Clock (f_{MCK})	SDRmn[15:9]	Maximum Permissible Value	Minimum Permissible Value
24 MHz ± 1.0%Note	$f_{CLK}/2^5$	79	1.60%	–2.18%
16 MHz ± 1.0%Note	$f_{CLK}/2^4$	105	2.27%	–1.53%
12 MHz ± 1.0%Note	$f_{CLK}/2^4$	79	1.60%	–2.19%
8 MHz ± 1.0%Note	$f_{CLK}/2^3$	105	2.27%	–1.53%
6 MHz ± 1.0%Note	$f_{CLK}/2^3$	79	1.60%	–2.19%
4 MHz ± 1.0%Note	$f_{CLK}/2^2$	105	2.27%	–1.53%
3 MHz ± 1.0%Note	$f_{CLK}/2^2$	79	1.60%	–2.19%
2 MHz ± 1.0%Note	$f_{CLK}/2$	105	2.27%	–1.54%
1 MHz ± 1.0%Note	f_{CLK}	105	2.27%	–1.57%

Note When the accuracy of the clock frequency of the high-speed on-chip oscillator is ±1.5% or ±2.0%, the permissible range becomes smaller as shown below.

- In the case of f_{IH} ± 1.5%, perform (Maximum permissible value – 0.5%) and (Minimum permissible value + 0.5%) to the values in the above table.
- In the case of f_{IH} ± 2.0%, perform (Maximum permissible value – 1.0%) and (Minimum permissible value + 1.0%) to the values in the above table.

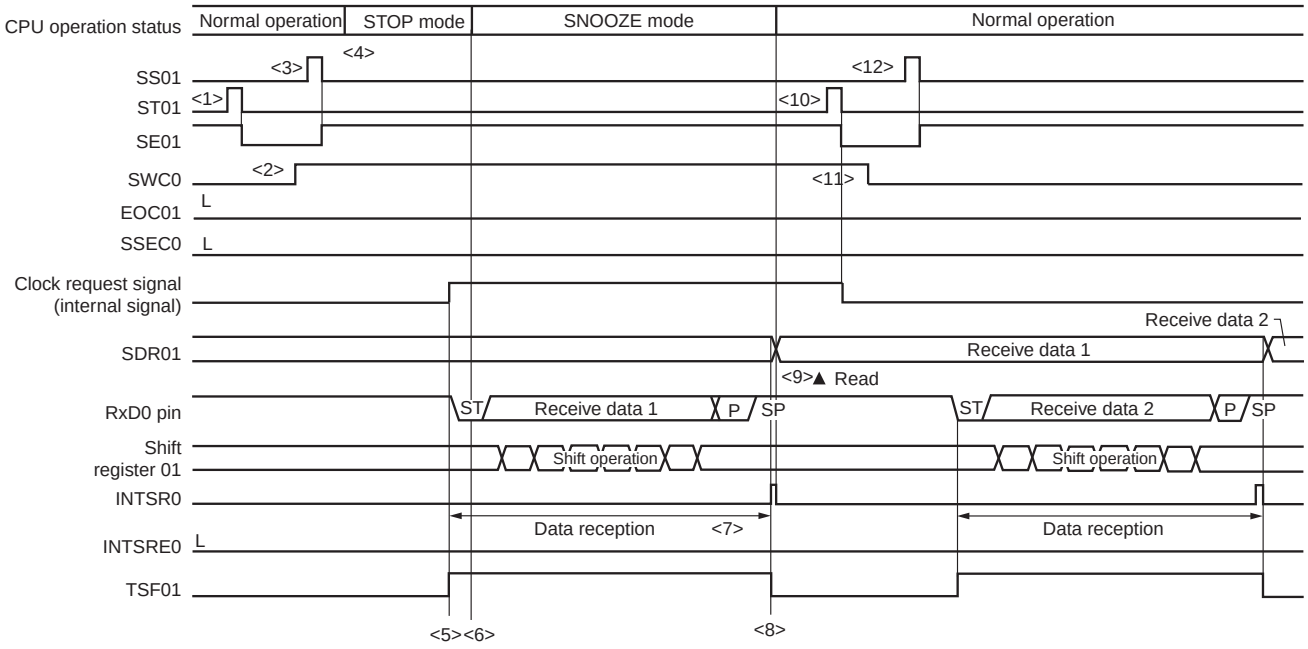
Remark The maximum permissible value and minimum permissible value are permissible values for the baud rate in UART reception. The baud rate on the transmitting side should be set to fall inside this range.

Downloaded from Arrow.com.

<R> **(1) SNOOZE mode operation (EOC01 = 0, SSEC0 = 0/1)**

Because of the setting of EOC01 = 0, even though a communication error occurs, an error interrupt (INTSRE0) is not generated, regardless of the setting of the SSEC0 bit. A transfer end interrupt (INTSR0) will be generated.

Figure 11-90. Timing Chart of SNOOZE Mode Operation (EOC01 = 0, SSEC0 = 0/1)



Note Read the received data when SWC0 = 1.

Caution Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, be sure to set the ST00 bit to 1 (clear the SE00 bit to stop the operation). And after completion the receive operation, also clearing SWC0 bit to 0 (SNOOZE mode release).

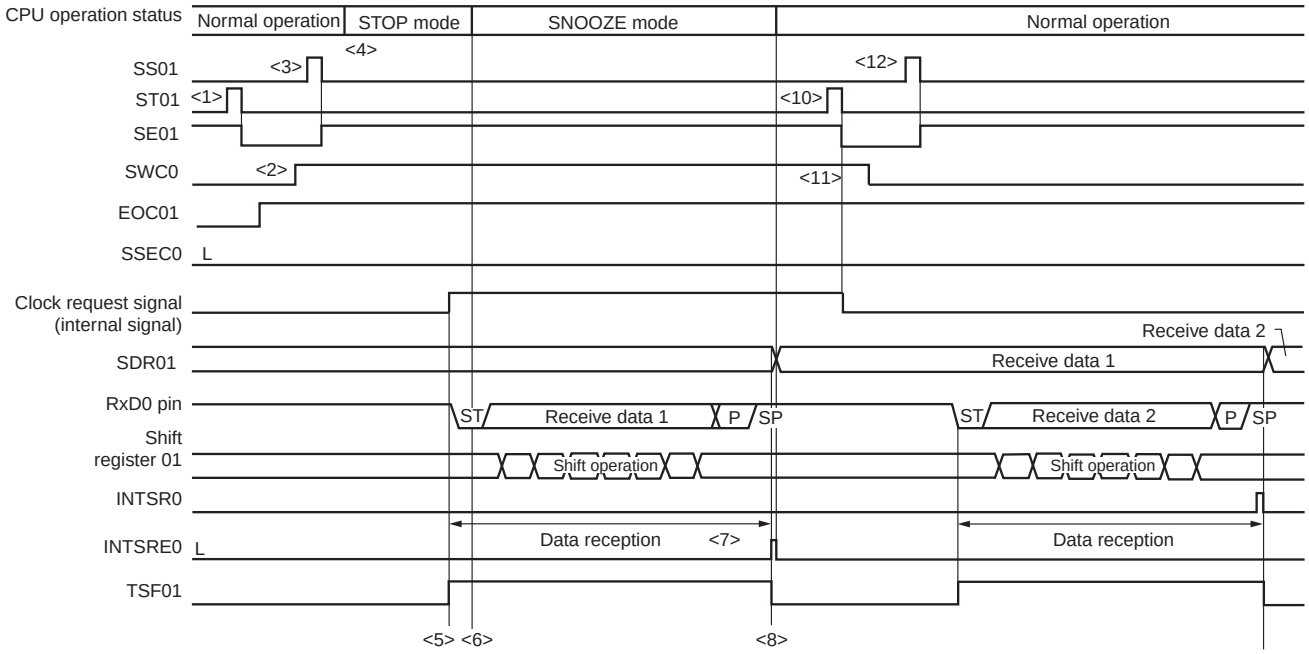
Remark <1> to <12> in the figure correspond to <1> to <12> in **Figure 11-92 Flowchart of SNOOZE Mode Operation (EOC01 = 0, SSEC0 = 0/1 or EOC01 = 1, SSEC0 = 0)**.

Downloaded from Arrow.com.

<R> **(2) SNOOZE mode operation (EOC01 = 1, SSEC0 = 0: Error interrupt (INTSRE0) generation is enabled)**

Because EOC01 = 0 and SSEC0 = 0, an error interrupt (INTSRE0) is generated when a communication error occurs.

Figure 11-91. Timing Chart of SNOOZE Mode Operation (EOC01 = 1, SSEC0 = 0)

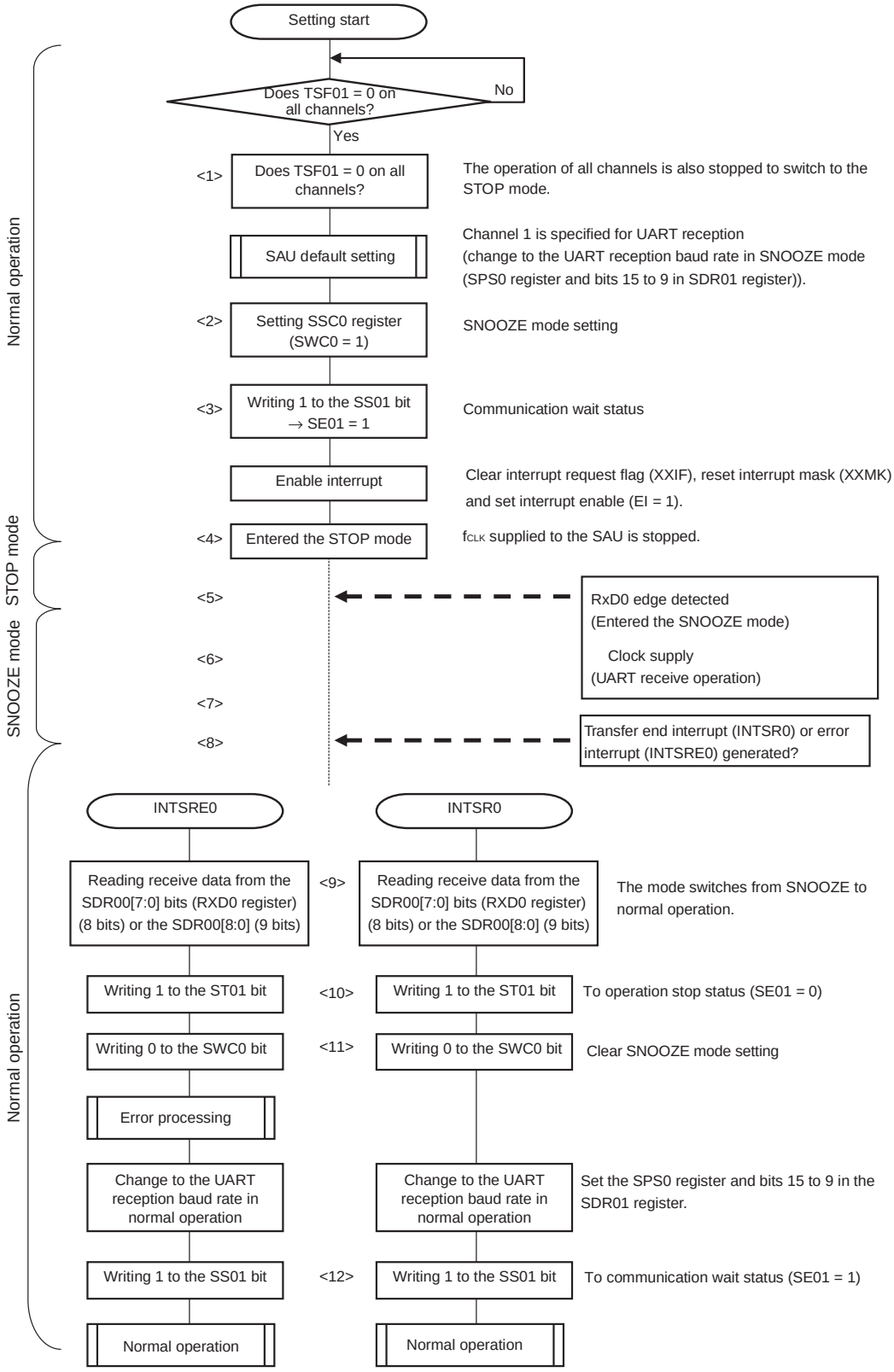


Caution Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, be sure to set the ST01 bit to 1 (clear the SE01 bit to stop the operation). And after completion of the receive operation, also clearing SWC0 bit to 0 (SNOOZE mode release).

Remark <1> to <12> in the figure correspond to <1> to <12> in **Figure 11-92 Flowchart of SNOOZE Mode Operation (EOC01 = 0, SSEC0 = 0/1 or EOC01 = 1, SSEC0 = 0)**.

Downloaded from Arrow.com.

<R> **Figure 11-92. Flowchart of SNOOZE Mode Operation (EOC01 = 0, SSEC0 = 0/1 or EOC01 = 1, SSEC0 = 0)**



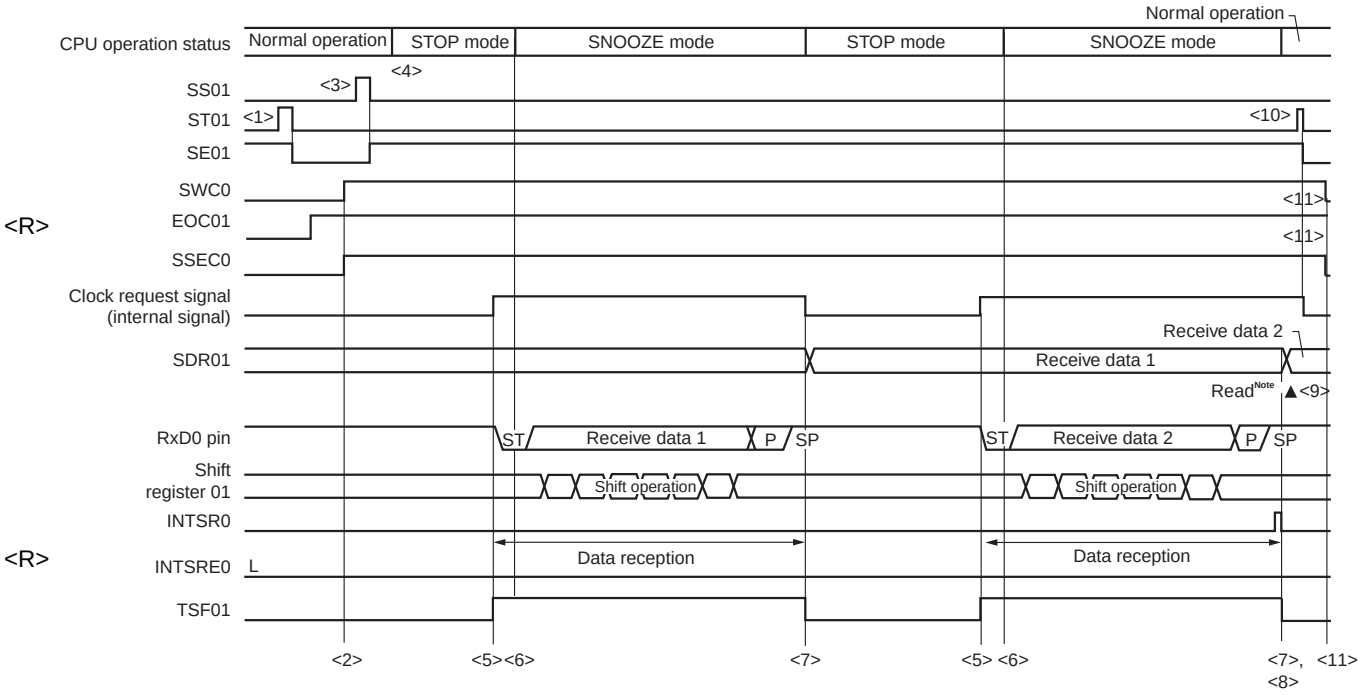
Remark <1> to <11> in the figure correspond to <1> to <11> in **Figure 11-90 Timing Chart of SNOOZE Mode Operation (EOC01 = 0, SSEC0 = 0/1)** and **Figure 11-91 Timing Chart of SNOOZE Mode Operation (EOC01 = 1, SSEC0 = 0)**.

Downloaded from Arrow.com.

<R> **(3) SNOOZE mode operation (EOC01 = 1, SSEC0 = 1: Error interrupt (INTSRE0) generation is stopped)**

Because EOC01 = 1 and SSEC0 = 1, an error interrupt (INTSRE0) is not generated when a communication error occurs.

Figure 11-93. Timing Chart of SNOOZE Mode Operation (EOC01 = 1, SSEC0 = 1: Error interrupt (INTSRE0) generation is stopped)



Note Read the received data when SWC0 = 1.

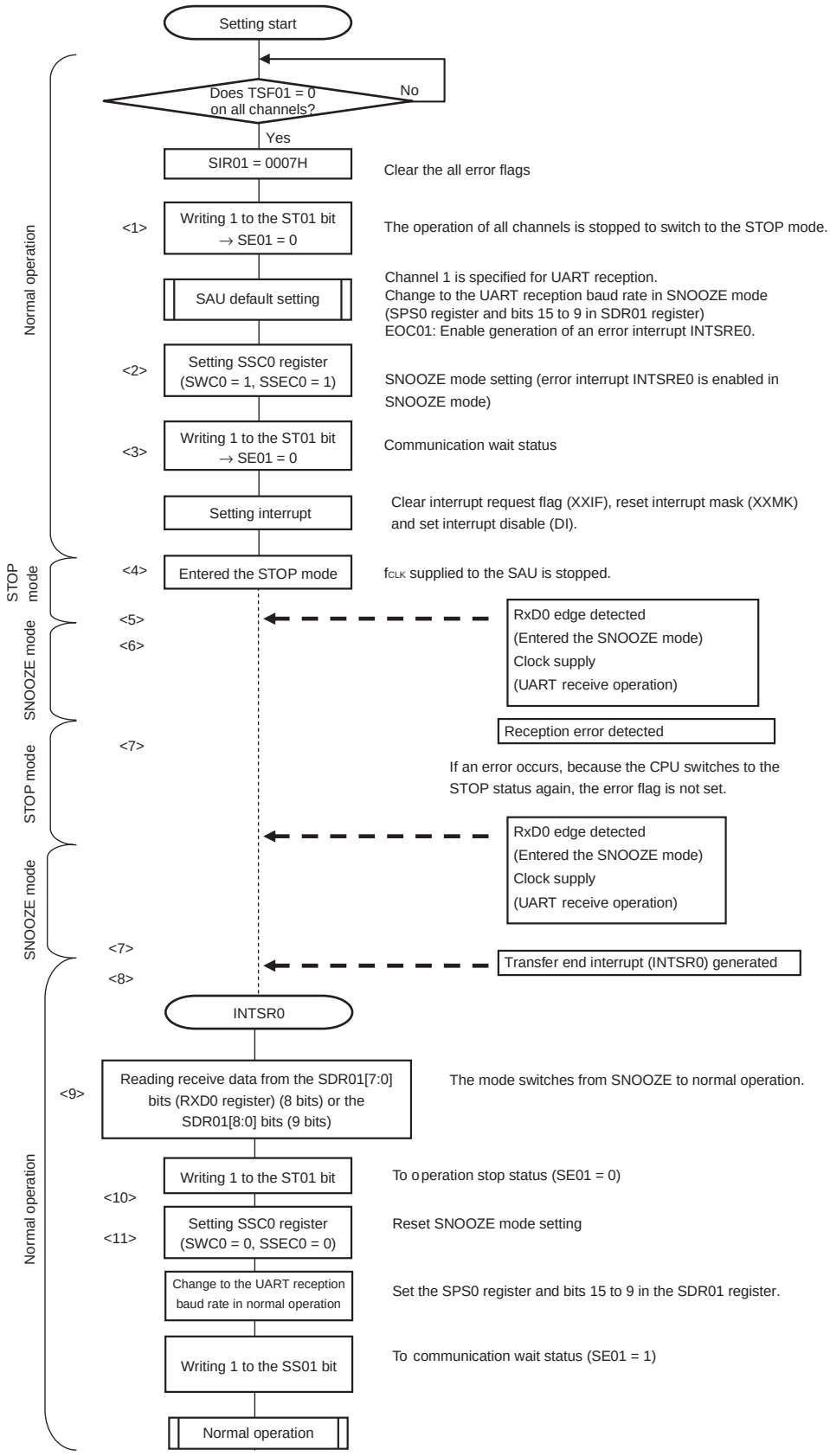
Cautions 1. Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the ST01 bit to 1 (clear the SE01 bit and stop the operation).
After the receive operation completes, also clear the SWC0 bit to 0 (SNOOZE mode release).
2. If a parity error, framing error, or overrun error occurs while the SSEC0 bit is set to 1, the PEF01, FEF01, or OVF01 flag is not set and an error interrupt (INTSRE0) is not generated. Therefore, when the setting of SSEC0 = 1 is made, clear the PEF01, FEF01, or OVF01 flag before setting the SWC0 bit to 1 and read the value in SDR01[7:0] (RxD0 register) (8 bits) or SDR01[8:0] (9 bits).

Remark <1> to <11> in the figure correspond to <1> to <11> in **Figure 11-94 Flowchart of SNOOZE Mode Operation (EOC01 = 1, SSEC0 = 1)**.

Downloaded from Arrow.com.

<R>

Figure 11-94. Flowchart of SNOOZE Mode Operation (EOC01 = 1, SSEC0 = 1)



Downloaded from Arrow.com.

Caution **If a parity error, framing error, or overrun error occurs while the SSEC0 bit is set to 1, the PEF01, FEF01, or OVF01 flag is not set and an error interrupt (INTSRE0) is not generated. Therefore, when the setting of SSEC0 = 1 is made, clear the PEF01, FEF01, or OVF01 flag before setting the SWC0 bit to 1 and read the value in SDR01[7:0] (RxD0 register) (8 bits) or SDR01[8:0] (9 bits).**

Remark <1> to <11> in the figure correspond to <1> to <11> in **Figure 11-93 Timing Chart of SNOOZE Mode Operation (EOC01 = 1, SSEC0 = 1: Error interrupt (INTSRE0) generation is stopped)**.

Downloaded from Arrow.com.

11.6.4 Calculating baud rate

(1) Baud rate calculation expression

The baud rate for UART (UART0 to UART2) communication can be calculated by the following expressions.

(Baud rate) = {Operation clock (f_{MCK}) frequency of target channel} ÷ (SDR0n[15:9] + 1) ÷ 2 [bps]

Caution Setting serial data register mn (SDRmn) SDRmn[15:9] = (0000000B, 0000001B) is prohibited.

Remarks 1. When UART is used, the value of SDRmn[15:9] is the value of bits 15 to 9 of the SDRmn register (0000010B, 1111111B) and therefore is 2 to 127.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

The operation clock (f_{MCK}) is determined by serial clock select register m (SPSm) and bit 15 (CKSmn) of serial mode register mn (SMRmn).

Downloaded from Arrow.com.

Table 11-4. Selection of Operation Clock For UART

SMRmn Register	SPSm Register								Operation Clock (f_{MCK})[Note]	
CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		f_{CLK} = 20 MHz
0	X	X	X	X	0	0	0	0	f_{CLK}	20 MHz
	X	X	X	X	0	0	0	1	$f_{CLK}/2$	10 MHz
	X	X	X	X	0	0	1	0	$f_{CLK}/2^2$	5 MHz
	X	X	X	X	0	0	1	1	$f_{CLK}/2^3$	2.5 MHz
	X	X	X	X	0	1	0	0	$f_{CLK}/2^4$	1.25 MHz
	X	X	X	X	0	1	0	1	$f_{CLK}/2^5$	625 kHz
	X	X	X	X	0	1	1	0	$f_{CLK}/2^6$	312.5 kHz
	X	X	X	X	0	1	1	1	$f_{CLK}/2^7$	156.2 kHz
	X	X	X	X	1	0	0	0	$f_{CLK}/2^8$	78.1 kHz
	X	X	X	X	1	0	0	1	$f_{CLK}/2^9$	39.1 kHz
	X	X	X	X	1	0	1	0	$f_{CLK}/2^{10}$	19.5 kHz
	X	X	X	X	1	0	1	1	$f_{CLK}/2^{11}$	9.77 kHz
	X	X	X	X	1	1	0	0	$f_{CLK}/2^{12}$	4.88 kHz
	X	X	X	X	1	1	0	1	$f_{CLK}/2^{13}$	2.44 kHz
	X	X	X	X	1	1	1	0	$f_{CLK}/2^{14}$	1.22 kHz
	X	X	X	X	1	1	1	1	$f_{CLK}/2^{15}$	610 Hz
1	0	0	0	0	X	X	X	X	f_{CLK}	20 MHz
	0	0	0	1	X	X	X	X	$f_{CLK}/2$	10 MHz
	0	0	1	0	X	X	X	X	$f_{CLK}/2^2$	5 MHz
	0	0	1	1	X	X	X	X	$f_{CLK}/2^3$	2.5 MHz
	0	1	0	0	X	X	X	X	$f_{CLK}/2^4$	1.25 MHz
	0	1	0	1	X	X	X	X	$f_{CLK}/2^5$	625 MHz
	0	1	1	0	X	X	X	X	$f_{CLK}/2^6$	312.5 kHz
	0	1	1	1	X	X	X	X	$f_{CLK}/2^7$	156.2 kHz
	1	0	0	0	X	X	X	X	$f_{CLK}/2^8$	78.1 kHz
	1	0	0	1	X	X	X	X	$f_{CLK}/2^9$	39.1 kHz
	1	0	1	0	X	X	X	X	$f_{CLK}/2^{10}$	19.5 kHz
	1	0	1	1	X	X	X	X	$f_{CLK}/2^{11}$	9.77 kHz
	1	1	0	0	X	X	X	X	$f_{CLK}/2^{12}$	4.88 kHz
	1	1	0	1	X	X	X	X	$f_{CLK}/2^{13}$	2.44 kHz
	1	1	1	0	X	X	X	X	$f_{CLK}/2^{14}$	1.22 kHz
	1	1	1	1	X	X	X	X	$f_{CLK}/2^{15}$	610 Hz

Note When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), do so after having stopped (serial channel stop register m (STm) = 000FH) the operation of the serial array unit (SAU).

Remarks 1. X: don't care

2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

Downloaded from Arrow.com.

(2) Baud rate error during transmission

The baud rate error of UART (UART0 to UART2) communication during transmission can be calculated by the following expression. Make sure that the baud rate at the transmission side is within the permissible baud rate range at the reception side.

(Baud rate error) = (Calculated baud rate value) ÷ (Target baud rate) × 100 – 100 [%]

Here is an example of setting a UART baud rate at f_{CLK} = 20 MHz.

UART Baud Rate (Target Baud Rate)	f_{CLK} = 20 MHz			
	Operation Clock (f_{MCK})	SDRmn[15:9]	Calculated Baud Rate	Error from Target Baud Rate
300 bps	$f_{CLK}/2^9$	64	300.48 bps	+0.16 %
600 bps	$f_{CLK}/2^8$	64	600.96 bps	+0.16 %
1200 bps	$f_{CLK}/2^7$	64	1201.92 bps	+0.16 %
2400 bps	$f_{CLK}/2^6$	64	2403.85 bps	+0.16 %
4800 bps	$f_{CLK}/2^5$	64	4807.69 bps	+0.16 %
9600 bps	$f_{CLK}/2^4$	64	9615.38 bps	+0.16 %
19200 bps	$f_{CLK}/2^3$	64	19230.8 bps	+0.16 %
31250 bps	$f_{CLK}/2^3$	39	31250.0 bps	±0.0 %
38400 bps	$f_{CLK}/2^2$	64	38461.5 bps	+0.16 %
76800 bps	$f_{CLK}/2$	64	76923.1 bps	+0.16 %
153600 bps	f_{CLK}	64	153846 bps	+0.16 %
312500 bps	f_{CLK}	31	312500 bps	±0.0 %

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 10

Downloaded from Arrow.com.

(3) Permissible baud rate range for reception

The permissible baud rate range for reception during UART (UART0 to UART2) communication can be calculated by the following expression. Make sure that the baud rate at the transmission side is within the permissible baud rate range at the reception side.

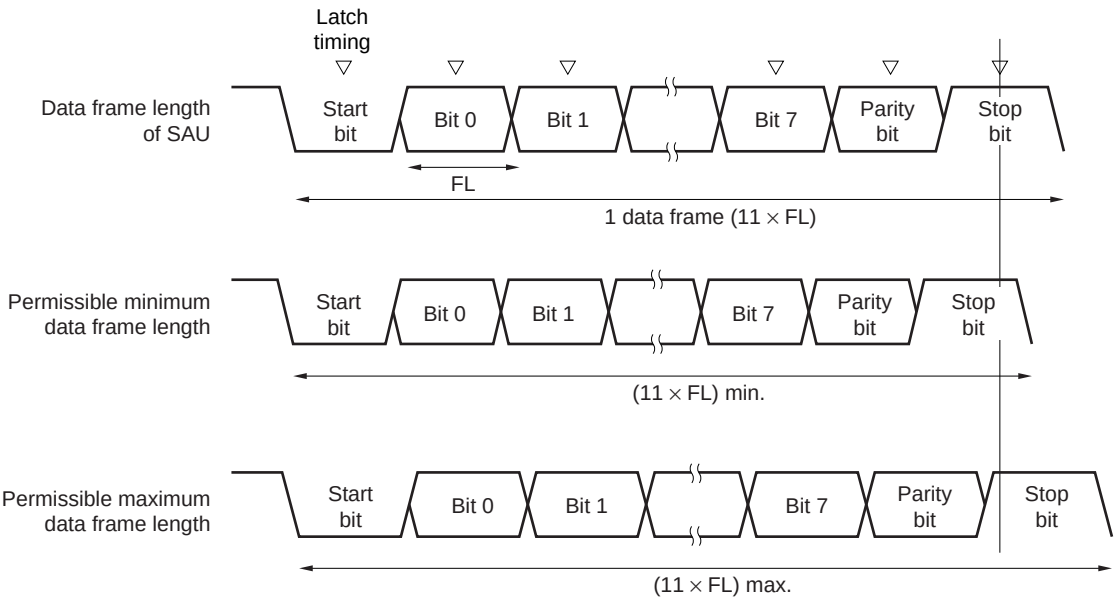
$$\text{(Maximum receivable baud rate)} = \frac{2 \times k \times Nfr}{2 \times k \times Nfr - k + 2} \times Brate$$

$$\text{(Minimum receivable baud rate)} = \frac{2 \times k \times (Nfr - 1)}{2 \times k \times Nfr - k - 2} \times Brate$$

Brate: Calculated baud rate value at the reception side (See **11.6.4 (1) Baud rate calculation expression**.)
k: SDRmn[15:9] + 1
Nfr: 1 data frame length [bits]
= (Start bit) + (Data length) + (Parity bit) + (Stop bit)

Remark m: Unit number (m = 0, 1), n: Channel number (n = 1, 3), mn = 01, 03, 11

Figure 11-95. Permissible Baud Rate Range for Reception (1 Data Frame Length = 11 Bits)



As shown in Figure 11-95 the timing of latching receive data is determined by the division ratio set by bits 15 to 9 of serial data register mn (SDRmn) after the start bit is detected. If the last data (stop bit) is received before this latch timing, the data can be correctly received.

Downloaded from Arrow.com.

11.6.5 Procedure for processing errors that occurred during UART (UART0 to UART2) communication

The procedure for processing errors that occurred during UART (UART0 to UART2) communication is described in Figures 11-96 and 11-97.

Figure 11-96. Processing Procedure in Case of Parity Error or Overrun Error

Software Manipulation	Hardware Status	Remark
Reads serial data register mn (SDRmn). →	The BFFmn bit of the SSRmn register is set to 0 and channel n is enabled to receive data.	This is to prevent an overrun error if the next reception is completed during error processing.
Reads serial status register mn (SSRmn).		Error type is identified and the read value is used to clear error flag.
Writes 1 to serial flag clear trigger register mn (SIRmn). →	Error flag is cleared.	Error can be cleared only during reading, by writing the value read from the SSRmn register to the SIRmn register without modification.

Figure 11-97. Processing Procedure in Case of Framing Error

Software Manipulation	Hardware Status	Remark
Reads serial data register mn (SDRmn). →	The BFFmn bit of the SSRmn register is set to 0 and channel n is enabled to receive data.	This is to prevent an overrun error if the next reception is completed during error processing.
Reads serial status register mn (SSRmn).		Error type is identified and the read value is used to clear error flag.
Writes serial flag clear trigger register mn (SIRmn). →	Error flag is cleared.	Error can be cleared only during reading, by writing the value read from the SSRmn register to the SIRmn register without modification.
Sets the STmn bit of serial channel stop register m (STm) to 1. →	The SEmn bit of serial channel enable status register m (SEm) is set to 0 and channel n stops operating.	
Synchronization with other party of communication		Synchronization with the other party of communication is re-established and communication is resumed because it is considered that a framing error has occurred because the start bit has been shifted.
Sets the SSmn bit of serial channel start register m (SSm) to 1. →	The SEmn bit of serial channel enable status register m (SEm) is set to 1 and channel n is enabled to operate.	

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

Downloaded from Arrow.com.

11.7 Operation of Simplified I^2C (IIC00, IIC01, IIC11, IIC20) Communication

This is a clocked communication function to communicate with two or more devices by using two lines: serial clock (SCL) and serial data (SDA). This communication function is designed to execute single communication with devices such as EEPROM, flash memory, and A/D converter, and therefore, can be used only by the master.

Operate the control registers by software for setting the start and stop conditions while observing the specifications of the I^2C bus line.

[Data transmission/reception]

- Master transmission, master reception (only master function with a single master)
- ACK output function**Note** and ACK detection function
- Data length of 8 bits
  (When an address is transmitted, the address is specified by the higher 7 bits, and the least significant bit is used for R/W control.)
- Generation of start condition and stop condition for software

[Interrupt function]

- Transfer end interrupt

[Error detection flag]

<R>
- Overrun error
- ACK error

* [Functions not supported by simplified I^2C]

- Slave transmission, slave reception
- Multi master function (Arbitration loss detection function)
- Wait detection function

Note When receiving the last data, ACK will not be output if 0 is written to the SOEmn (SOEm register) bit and serial communication data output is stopped. See the processing flow in **11.7.3 (2)** for details.

Remark m: Unit number, n: Channel number (mn = 00, 01, 03, 10)

Downloaded from Arrow.com.

The channels supporting simplified I²C (IIC00, IIC01, IIC11, IIC20) are channels 0, 1, 3 of SAU0 and channel 0 of SAU1.

20- or 24-pin products

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I²C
0	0	CSI00	UART0	IIC00Note
	1	CSI01Note		IIC01Note

30-pin products

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I²C
0	0	CSI00	UART0	IIC00Note
	1	–		–
	2	–	UART1Note	–
	3	CSI11Note		IIC11Note
1	0	CSI20Note	UART 2Note	IIC20Note
	1	–		–

Note Provided in the R5F102 products only.

Simplified I²C (IIC00, IIC01, IIC11, IIC20) performs the following four types of communication operations.

- Address field transmission (See **11.7.1**.)
- Data transmission (See **11.7.2**.)
- Data reception (See **11.7.3**.)
- Stop condition generation (See **11.7.4**.)

Downloaded from Arrow.com.

11.7.1 Address field transmission

Address field transmission is a transmission operation that first executes in I^2C communication to identify the target for transfer (slave). After a start condition is generated, an address (7 bits) and a transfer direction (1 bit) are transmitted in one frame.

Simplified I^2C	IIC00	IIC01	IIC11	IIC20
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1
<R> Pins used	SCL00, SDA00Note 1	SCL01, SDA01Note 1	SCL11, SDA11Note 1	SCL20, SDA20Note 1
Interrupt	INTIIC00	INTIIC01	INTIIC11	INTIIC20
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)			
Error detection flag	ACK error detection flag (PEFmn)			
Transfer data length	8 bits (transmitted with specifying the higher 7 bits as address and the least significant bit as R/W control)			
<R> Transfer rateNote 2	Max. $f_{MCK}/4$ [Hz] (SDRmn[15:9] = 1 or more) f_{MCK}: Operation clock frequency of target channel However, the following condition must be satisfied in each mode of I^2C. • Max. 400 kHz (fast mode) • Max. 100 kHz (standard mode)			
Data level	Non-inversion output (default: high level)			
Parity bit	No parity bit			
Stop bit	Appending 1 bit (for ACK transmission/reception timing)			
Data direction	MSB first			

<R> **Notes 1.** To perform communication via simplified I^2C, set the N-ch open-drain output (V_{DD} tolerance) mode (POMxx = 1 (POM11, POM41 = 1 for 20- or 24-pin products, POM11, POM14, POM50 = 1 for 30-pin products)) for the port output mode registers (POMxx). For details, see **4.3 Registers Controlling Port Function** and **4.5 Register Settings When Using Alternate Function**.
When IIC00 and IIC20 are communicating with an external device with a different potential, set the N-ch open-drain output (V_{DD} tolerance) mode (POMxx = 1 (POM10 = 1 for 20- or 24-pin products, POM10, POM15 = 1 for 30-pin products)) also for the clock input/output pins (SCL00, SCL20). For details, see **4.4.4 Handling different potentials (1.8 V, 2.5 V, and 3 V) by using I/O buffers**.

<R> **2.** Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 28 ELECTRICAL SPECIFICATIONS (A, D: T_A = -40 to +85°C)** or **CHAPTER 29 ELECTRICAL SPECIFICATIONS (G: T_A = -40 to +105°C)**).

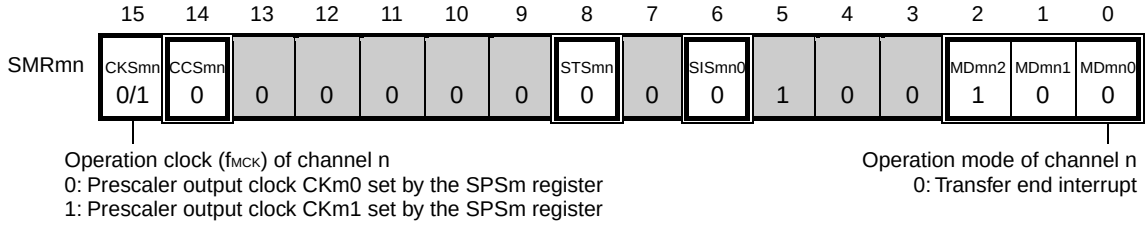
Remark m: Unit number, n: Channel number (mn = 00, 01, 03, 10)

Downloaded from Arrow.com.

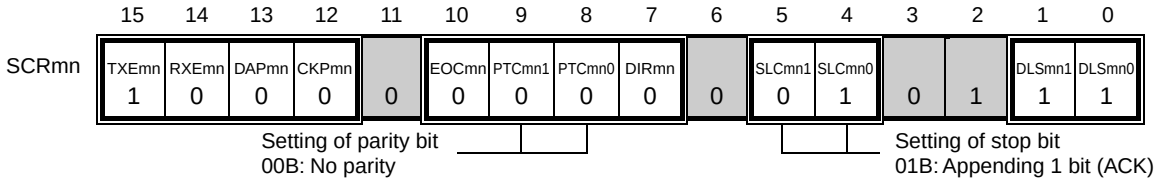
(1) Register setting

Figure 11-98. Example of Contents of Registers for Address Field Transmission of Simplified I^2C (IIC00, IIC01, IIC11, IIC20)

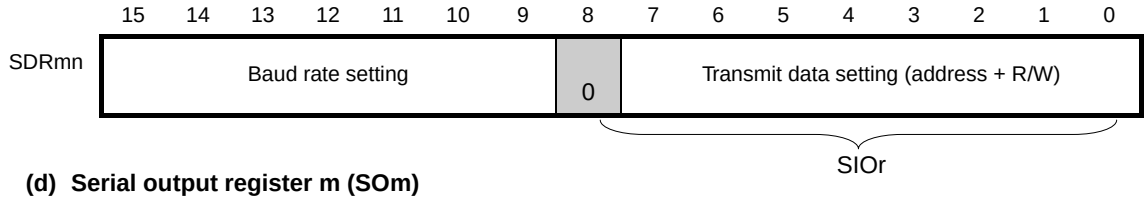
(a) Serial mode register 0n (SMR0n)



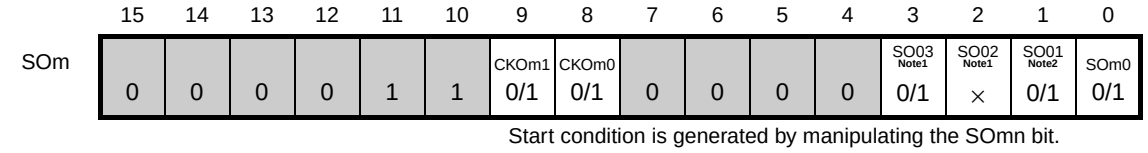
(b) Serial communication operation setting register mn (SCRmn)



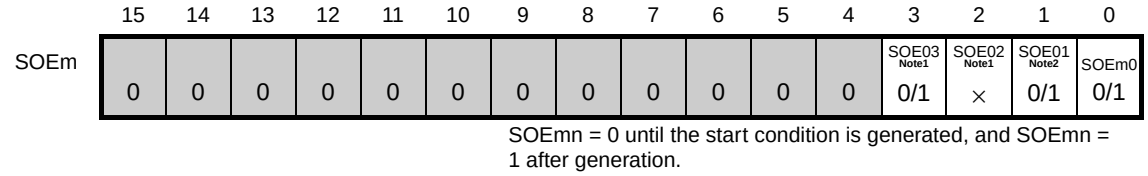
(c) Serial data register mn (SDRmn) (lower 8 bits: SIOr)



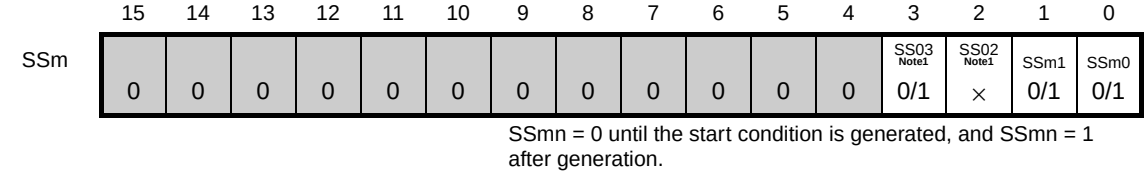
(d) Serial output register m (SOm)



(e) Serial output enable register m (SOEm)



(f) Serial channel start register m (SSm) ... Sets only the bits of the target channel is 1.



Notes 1. Provided only in 30-pin product serial array unit 0.

2. Only for 20, 24-pin product

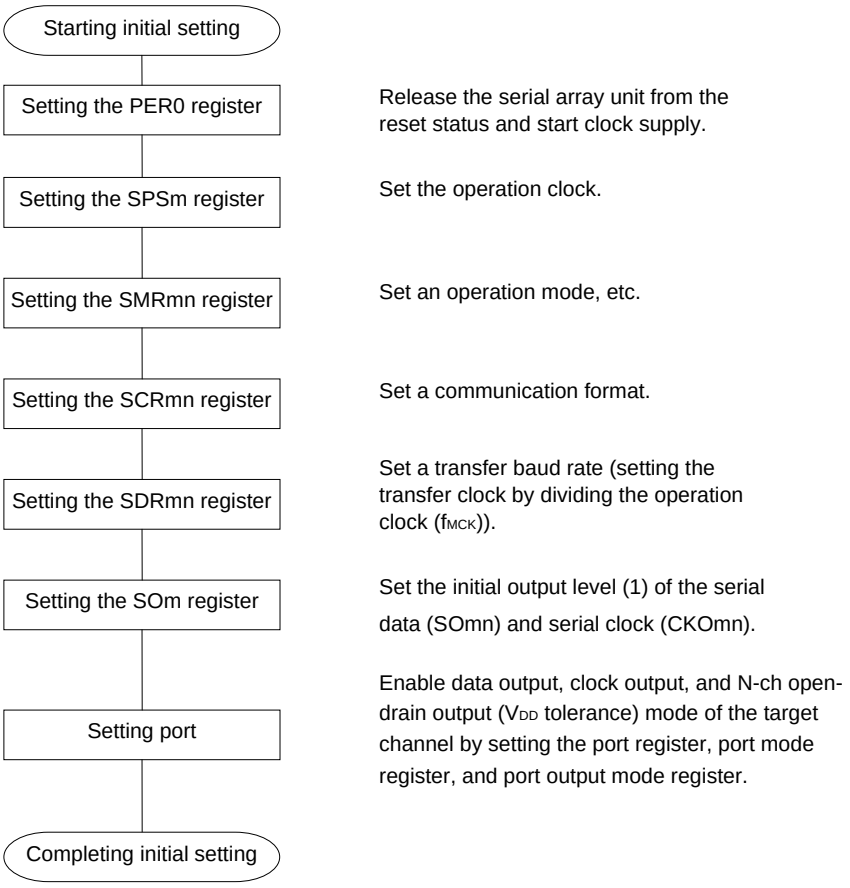
Remarks 1. m: Unit number, n: Channel number (mn = 00, 01, 03, 10), r: IIC number (r = 00, 01, 11, 20)

2. ☐: Setting is fixed in the CSI master transmission mode, ▒: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

Downloaded from Arrow.com.

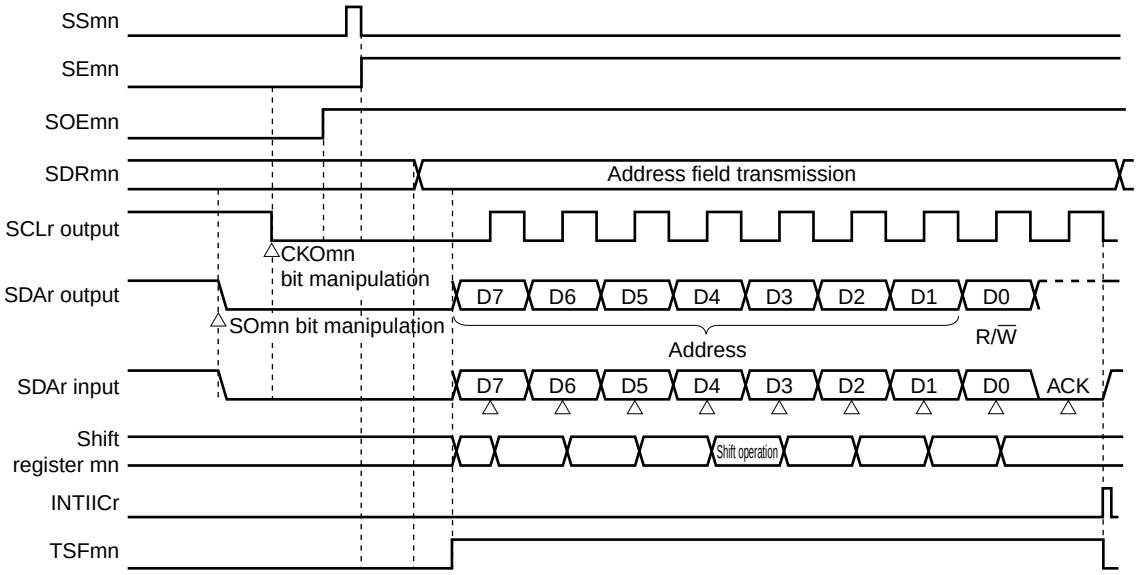
(2) Operation procedure

Figure 11-99. Initial Setting Procedure for simplified I^2C Address Field Transmission



(3) Processing flow

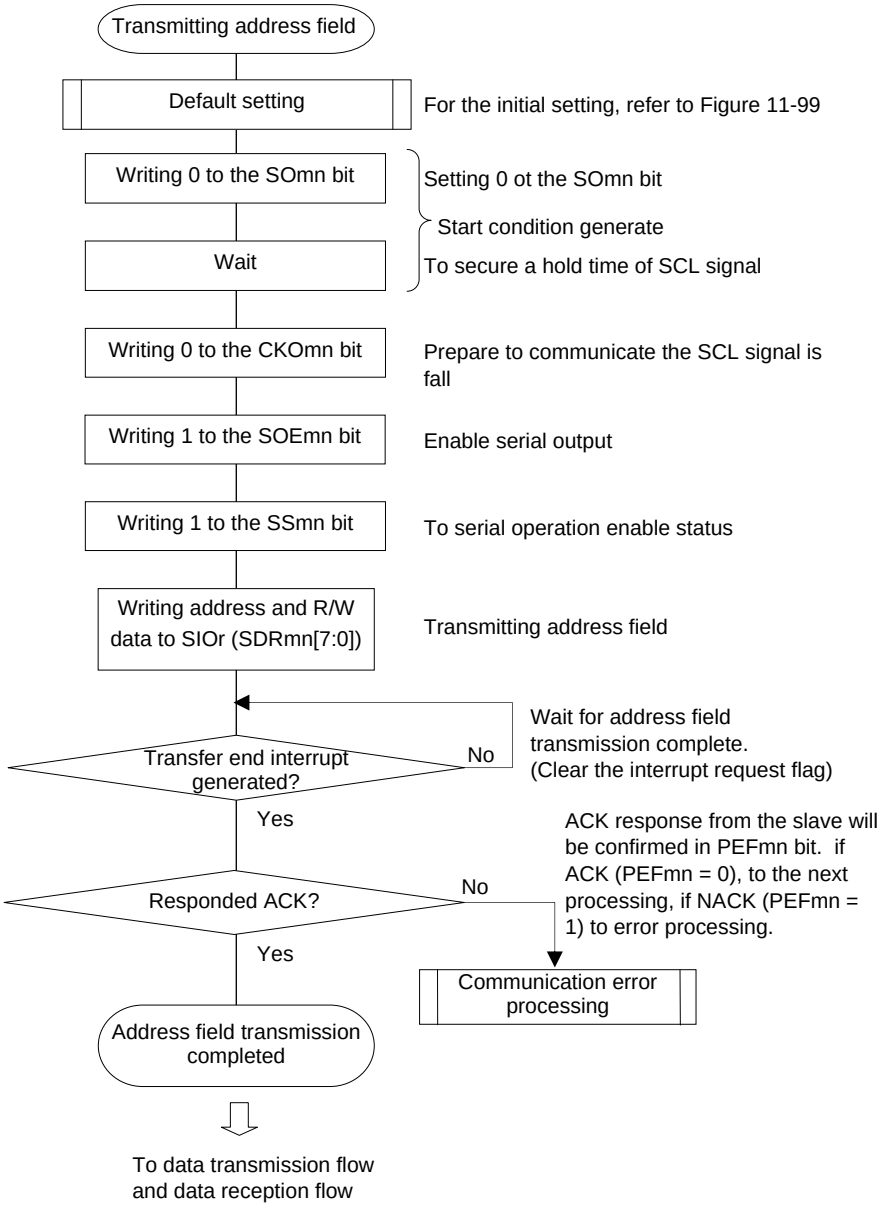
Figure 11-100. Timing Chart of Address Field Transmission



Remark m: Unit number, n: Channel number (mn = 00, 01, 03, 10), r: IIC number (r = 00, 01, 11, 20)

Downloaded from Arrow.com.

Figure 11-101. Flowchart of simplified I²C Address Field Transmission



Downloaded from Arrow.com.

11.7.2 Data transmission

Data transmission is an operation to transmit data to the target for transfer (slave) after transmission of an address field. After all data are transmitted to the slave, a stop condition is generated and the bus is released.

Simplified I²C	IIC00	IIC01	IIC11	IIC20
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1
<R> Pins used	SCL00, SDA00Note1	SCL01, SDA01Note1	SCL11, SDA11Note1	SCL20, SDA20Note1
Interrupt	INTIIC00	INTIIC01	INTIIC11	INTIIC20
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)			
Error detection flag	ACK error detection flag (PEFmn)			
Transfer data length	8 bits			
<R> Transfer rateNote2	Max. f_{MCK}/4 [Hz] (SDRmn[15:9] = 1 or more) f_{MCK}: Operation clock frequency of target channel However, the following condition must be satisfied in each mode of I²C. • Max. 400 kHz (fast mode) • Max. 100 kHz (standard mode)			
Data level	Non-inversion output (default: high level)			
Parity bit	No parity bit			
Stop bit	Appending 1 bit (for ACK reception timing)			
Data direction	MSB first			

<R> **Notes 1.** To perform communication via simplified I²C, set the N-ch open-drain output (V_{DD} tolerance) mode (POMxx = 1 (POM11, POM41 = 1 for 20- or 24-pin products, POM11, POM14, POM50 = 1 for 30-pin products)) for the port output mode registers (POMxx). For details, see **4.3 Registers Controlling Port Function**.
When IIC00 and IIC20 are communicating with an external device with a different potential, set the N-ch open-drain output (V_{DD} tolerance) mode (POMxx = 1 (POM10 = 1 for 20- or 24-pin products, POM10, POM15 = 1 for 30-pin products)) also for the clock input/output pins (SCL00, SCL20). For details, see **4.4.4 Handling different potentials (1.8 V, 2.5 V, and 3 V) by using I/O buffers**.

<R> **2.** Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 28 ELECTRICAL SPECIFICATIONS (A, D: T_A = -40 to +85°C)** or **CHAPTER 29 ELECTRICAL SPECIFICATIONS (G: T_A = -40 to +105°C)**).

Remark m: Unit number, n: Channel number (mn = 00, 01, 03, 10)

Downloaded from Arrow.com.

(1) Register setting

Figure 11-102. Example of Contents of Registers for Data Transmission of Simplified I^2C (IIC00, IIC01, IIC11, IIC20)

(a) Serial mode register mn (SMRmn) ... Do not manipulate this register during data transmission/reception.

SMRmn

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CKSmn	CCSmn						STSmn		SISmn0				MDmn2	MDmn1	MDmn0
0/1	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0

(b) Serial communication operation setting register mn (SCRmn) ... Do not manipulate the bits of this register, except the TXEmn and RXEmn bits, during data transmission/reception.

SCRmn

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXEmn	RXEmn	DAPmn	CKPmn		EOCmn	PTCmn1	PTCmn0	DIRmn		SLCmn1	SLCmn0			DLSmn1	DLSmn0
1	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1

(c) Serial data register mn (SDRmn) …During data transmission/reception, valid only lower 8-bits (SIOr)

SDRmn

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Baud rate setting[Note4]							0	Transmit data setting							

SIOr

(d) Serial output register m (SOm) ... Do not manipulate this register during data transmission/reception.

SOm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	1	1	CKOm1	CKOm0	0	0	0	0	SOm3 Note 1	SOm2 Note 1	SOm1 Note 2	SOm0
						0/1 Note 2	0/1 Note 2					0/1 Note 3	×	0/1 Note 3	0/1 Note 3

(e) Serial output enable register m (SOEm) ... Do not manipulate this register during data transmission/reception.

SOEm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	SOEm3 Note 1	SOEm2 Note 1	SOEm1 Note 2	SOEm0
												0/1	×	1	1

(f) Serial channel start register m (SSm) ... Do not manipulate this register during data transmission/reception.

SSm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	SSm3 Note 1	SSm2 Note 1	SSm1 Note 2	SSm0
												0/1	×	0/1	0/1

Notes 1. Provided only in 30-pin product serial array unit 0.

2. Only for 20, 24-pin product

3. The values may change during operation, depending on the communication data.

4. Because the setting is completed by address field transmission, setting is not required.

Remarks 1. m: Unit number, n: Channel number (mn = 00, 01, 03, 10), r: IIC number (r = 00, 01, 11, 20)

2. □: Setting is fixed in the IIC master transmission mode, ■: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

Downloaded from Arrow.com.

(2) Processing flow

Figure 11-103. Timing Chart of Data Transmission

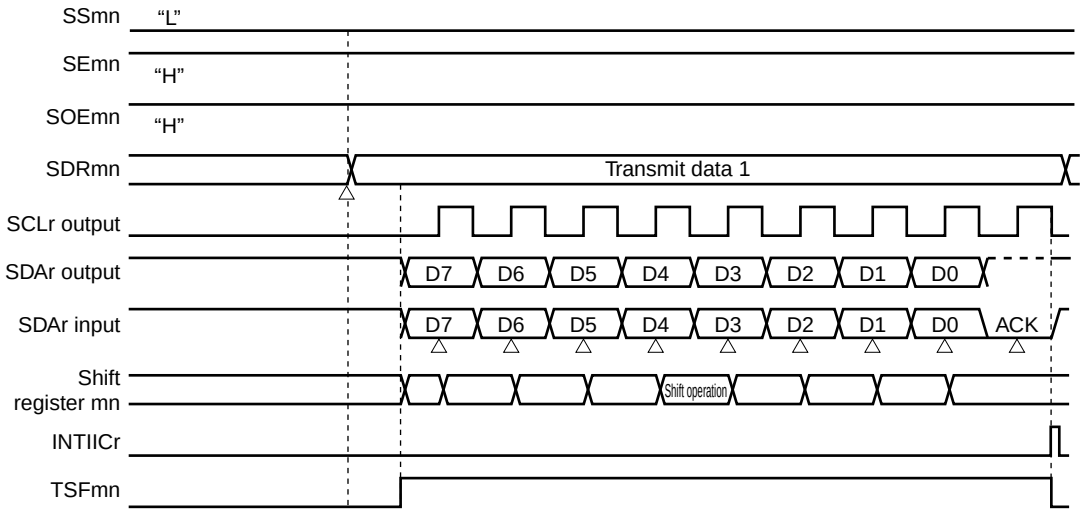
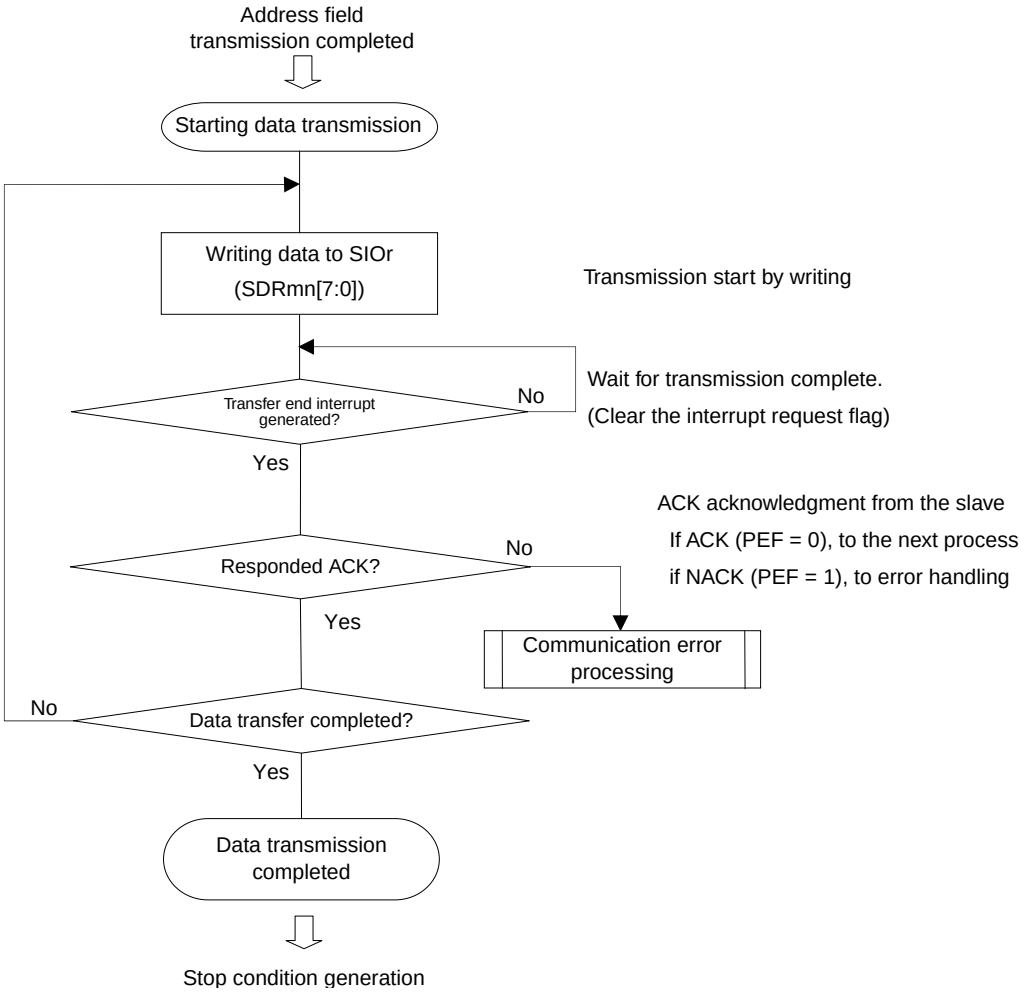


Figure 11-104. Flowchart of Simplified I^2C Data Transmission



Downloaded from Arrow.com.

11.7.3 Data reception

Data reception is an operation to receive data to the target for transfer (slave) after transmission of an address field. After all data are received to the slave, a stop condition is generated and the bus is released.

Simplified I^2C	IIC00	IIC01	IIC11	IIC20
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1
<R> Pins used	SCL00, SDA00Note 1	SCL01, SDA01Note 1	SCL11, SDA11Note 1	SCL20, SDA20Note 1
Interrupt	INTIIC00	INTIIC01	INTIIC11	INTIIC20
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)			
Error detection flag	ACK error detection flag (OVFmn)			
Transfer data length	8 bits			
<R> Transfer rateNote 2	Max. f_{MCK}/4 [Hz] (SDRmn[15:9] = 1 or more) f_{MCK}: Operation clock frequency of target channel However, the following condition must be satisfied in each mode of I^2C. • Max. 400 kHz (fast mode) • Max. 100 kHz (standard mode)			
Data level	Non-inversion output (default: high level)			
Parity bit	No parity bit			
Stop bit	Appending 1 bit (ACK transmission)			
Data direction	MSB first			

<R> **Notes 1.** To perform communication via simplified I^2C, set the N-ch open-drain output (V_{DD} tolerance) mode (POMxx = 1 (POM11, POM41 = 1 for 20- or 24- pin products, POM11, POM14, POM50 = 1 for 30-pin products)) for the port output mode registers (POMxx). For details, see **4.3 Registers Controlling Port Function** and **4.5 Register Settings When Using Alternate Function**.

When IIC00 and IIC20 is communicating with an external device with a different potential, set the N-ch open-drain output (V_{DD} tolerance) mode (POMxx = 1 (POM10 = 1 for 20- or 24- pin products, POM10, POM15 = 1 for 30-pin products)) also for the clock input/output pins (SCL00, SCL20). For details, see **4.4.4 Handling different potentials (1.8 V, 2.5 V, and 3 V) by using I/O buffers**.

<R> **2.** Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 28 ELECTRICAL SPECIFICATIONS (A, D: T_A = -40 to +85°C)** or **CHAPTER 29 ELECTRICAL SPECIFICATIONS (G: T_A = -40 to +105°C)**).

Remark m: Unit number, n: Channel number (mn = 00, 01, 03, 10)

Downloaded from Arrow.com.

(1) Register setting

Figure 11-105. Example of Contents of Registers for Data Reception of Simplified I^2C (IIC00, IIC01, IIC11, IIC20)

(a) Serial mode register mn (SMRmn) ... Do not manipulate this register during data transmission/reception.

SMRmn

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CKSmn	CCSmn						STSmn		SISmn0				MDmn2	MDmn1	MDmn0
0/1	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0

(b) Serial communication operation setting register mn (SCRmn) ... Do not manipulate the bits of this register, except the TXEmn and RXEmn bits, during data transmission/reception.

SCRmn

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXEmn	RXEmn	DAPmn	CKPmn		EOCmn	PTCmn1	PTCmn0	DIRmn		SLCmn1	SLCmn0			DLSmn1	DLSmn0
0	1	0	0	0	0	0	0	0	0	0	1	0	1	1	1

(c) Serial data register mn (SDRmn) (lower 8 bits: SIOr)

SDRmn

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Baud rate setting[Note4]							0	Dummy transmit data setting (FFH)							

SIOr

(d) Serial output register m (SOm) ... Do not manipulate this register during data transmission/reception.

SOm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	1	1	CKOm1	CKOm0	0	0	0	0	SOm3 Note 1	SOm2 Note 1	SOm1 Note 2	SOm0
						0/1 Note 2	0/1 Note 2					0/1 Note 3	×	0/1 Note 3	0/1 Note 3

(e) Serial output enable register m (SOEm) ... Do not manipulate this register during data transmission/reception.

SOEm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	SOEm3 Note 1	SOEm2 Note 1	SOEm1 Note 2	SOEm0
												0/1	×	0/1	0/1

(f) Serial channel start register m (SSm) ... Do not manipulate this register during data transmission/reception.

SSm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	SSm3 Note 1	SSm2 Note 1	SSm1	SSm0
												0/1	×	0/1	0/1

Notes 1. Provided only in 30-pin products serial array unit 0.

2. Only for 20, 24-pin products.

3. The values may change during operation, depending on the communication data.

4. Because the setting is completed by address field transmission, setting is not required.

Remarks 1. m: Unit number, n: Channel number (mn = 00, 01, 03, 10), r: IIC number (r = 00, 01, 11, 20)

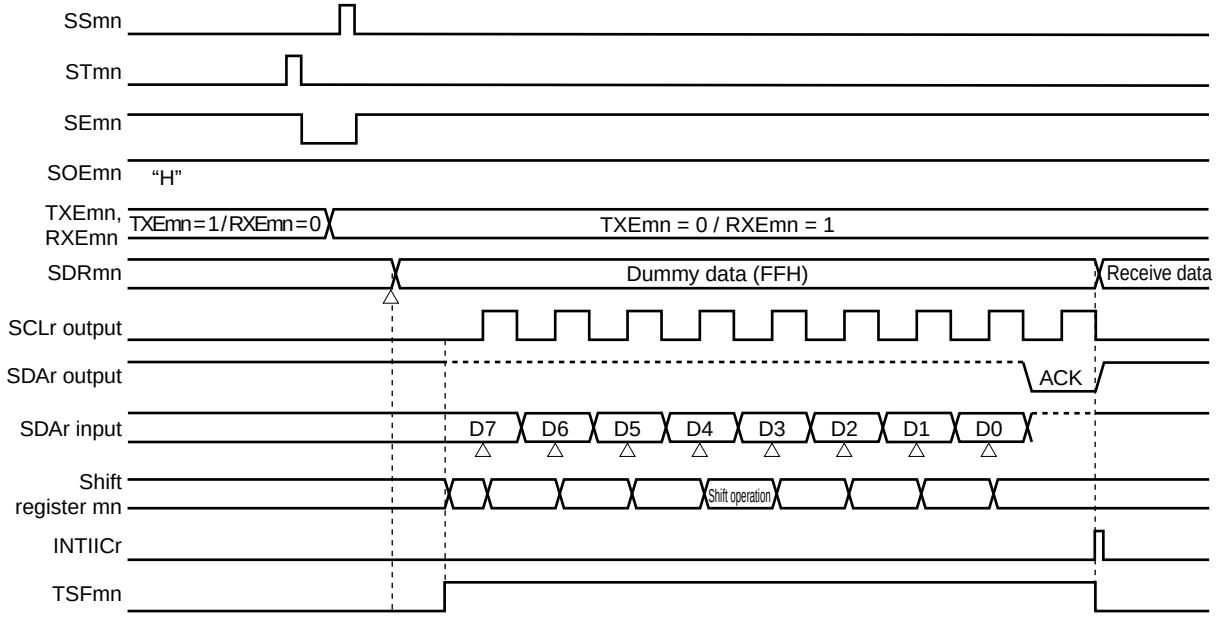
2. □: Setting is fixed in the IIC master transmission mode, ■ Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

Downloaded from Arrow.com.

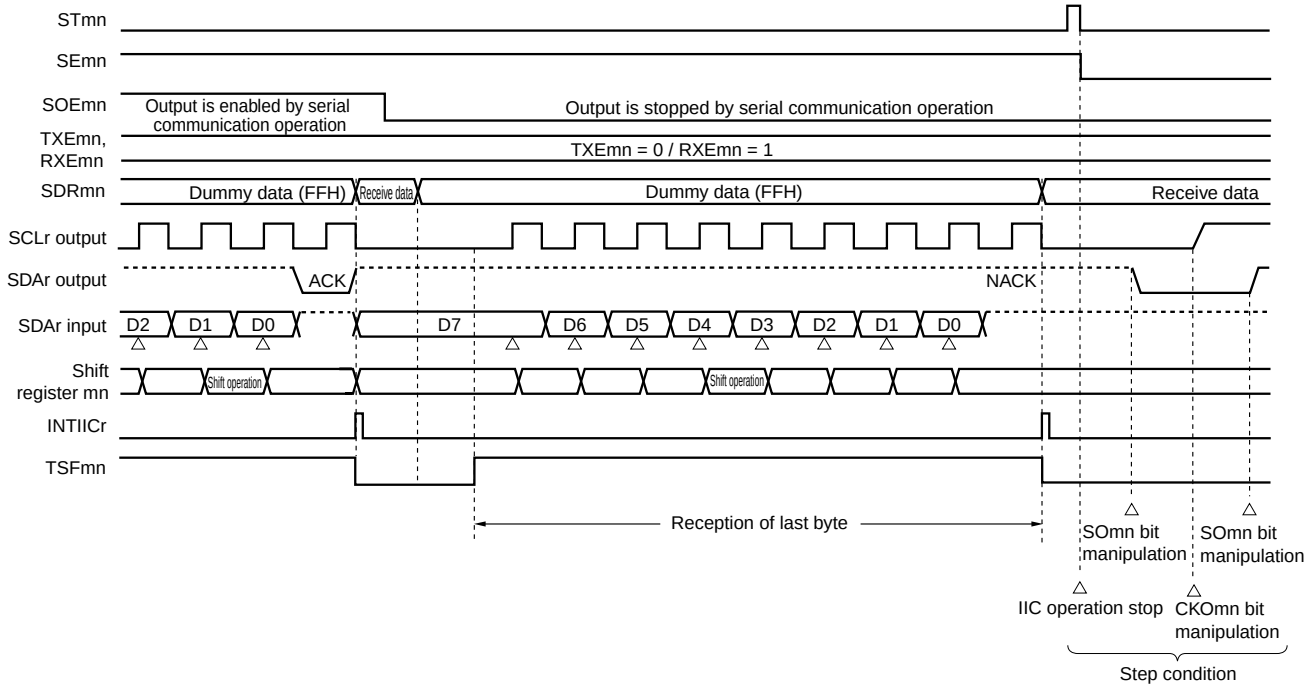
(2) Processing flow

Figure 11-106. Timing Chart of Data Reception

(a) When starting data reception



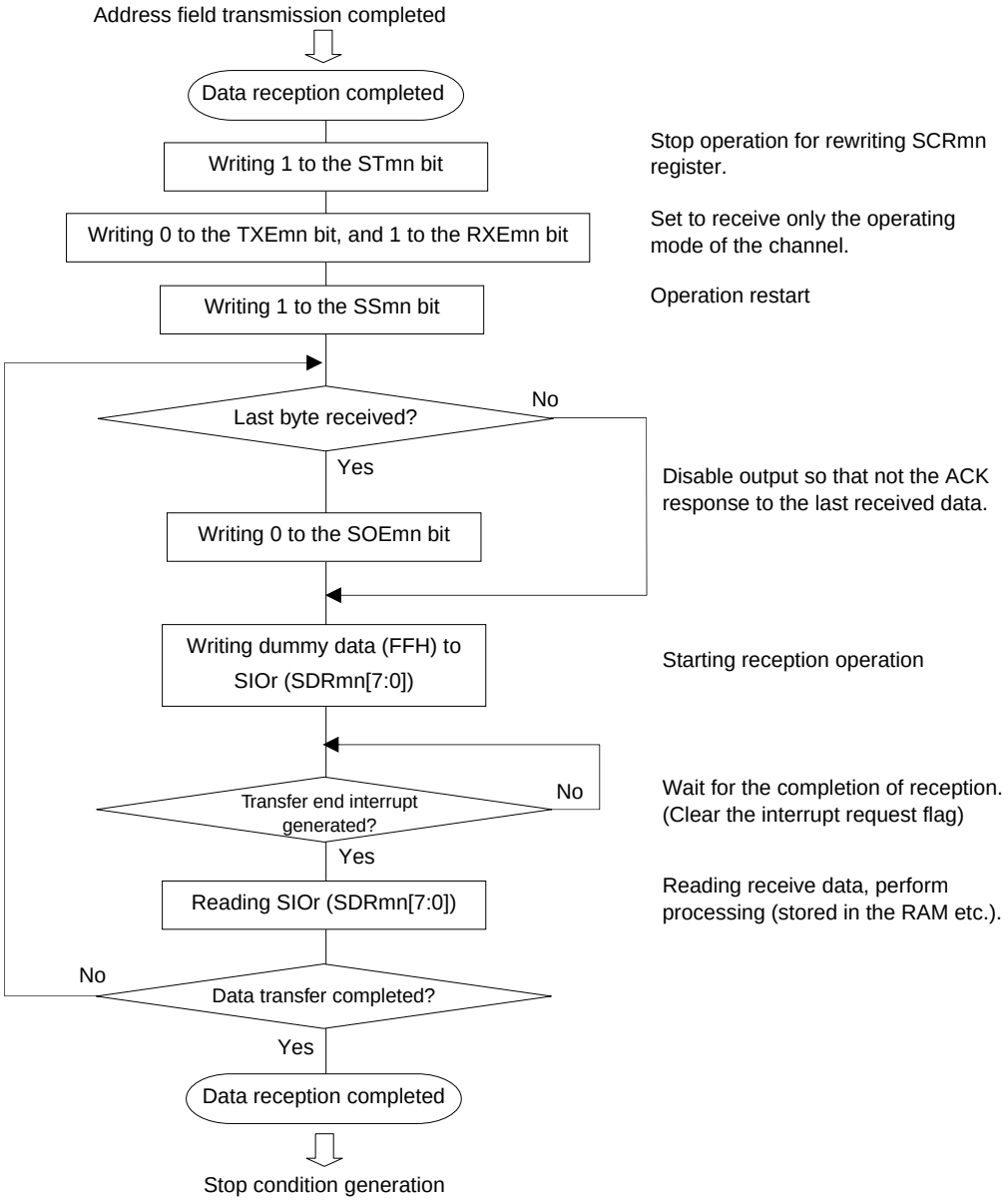
(b) When receiving last data



Remark m: Unit number, n: Channel number (mn = 00, 01, 03, 10), r: IIC number (r = 00, 01, 11, 20)

Downloaded from Arrow.com.

Figure 11-107. Flowchart of Simplified I²C Data Reception



Caution ACK is not output when the last data is received (NACK). Communication is then completed by setting "1" to the STmn bit of serial channel stop register m (STm) to stop operation and generating a stop condition.

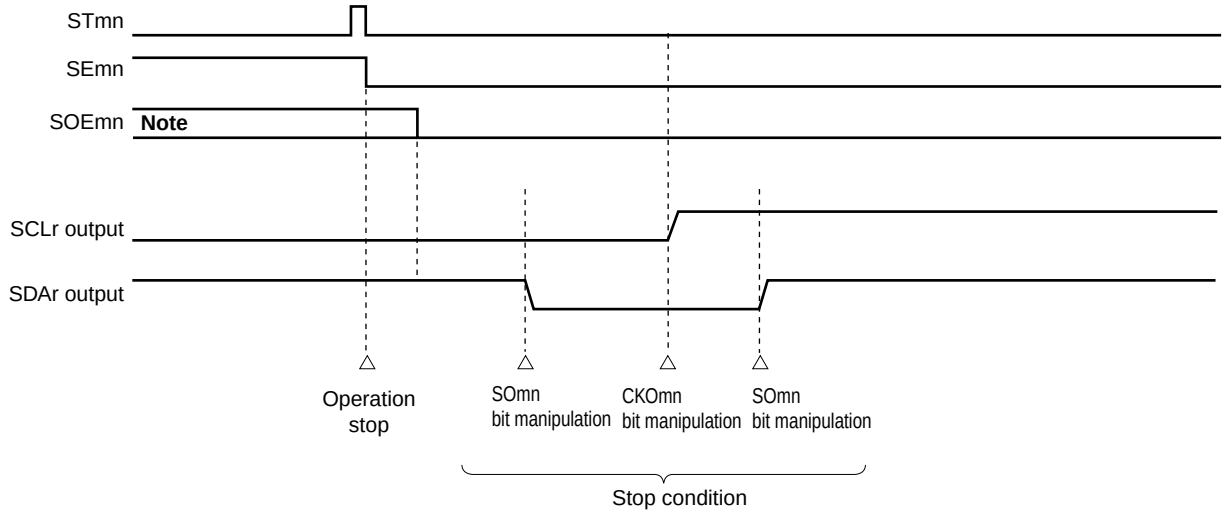
Downloaded from Arrow.com.

11.7.4 Stop condition generation

After all data are transmitted to or received from the target slave, a stop condition is generated and the bus is released.

(1) Processing flow

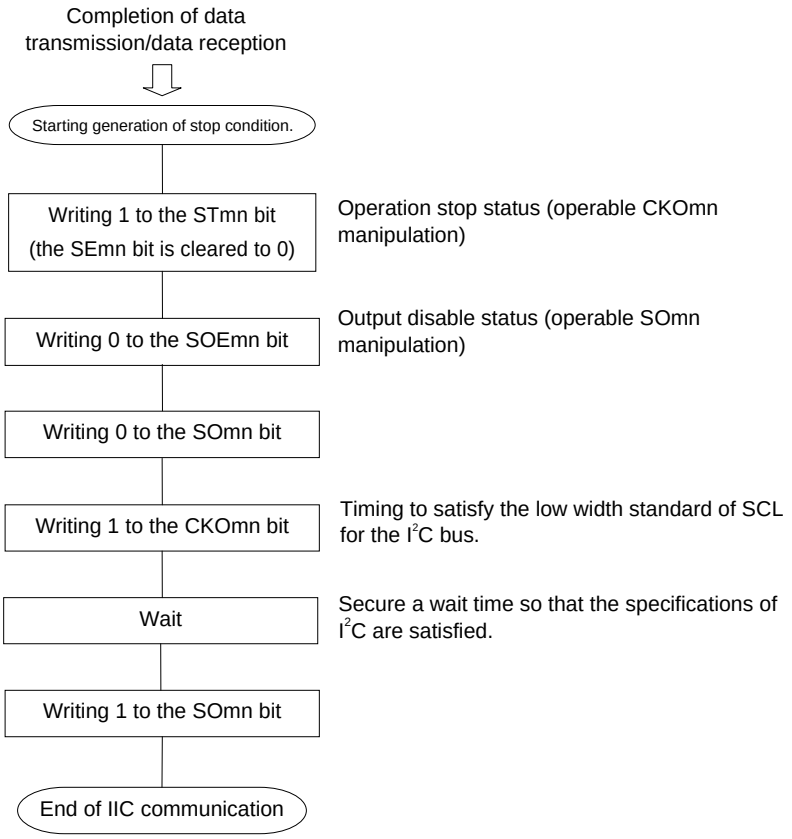
Figure 11-108. Timing Chart of Stop Condition Generation



Note During a receive operation, the SOEmn bit of serial output enable register m (SOEm) is cleared to 0 before receiving the last data.

<R>

Figure 11-109. Flowchart of Stop Condition Generation



Downloaded from Arrow.com.

11.7.5 Calculating transfer rate

The transfer rate for simplified I^2C (IIC00, IIC01, IIC11, IIC20) communication can be calculated by the following expressions.

(Transfer rate) = {Operation clock (f_{MCK}) frequency of target channel} ÷ (SDRmn[15:9] + 1) ÷ 2

Caution SDRmn[15:9] must not be set to 00000000B. Be sure to set SDRmn[15:9] to a value of at least 00000001B. The duty cycle of the SCL signal output by the simplified I^2C interface is 50%. The I^2C bus specifications stipulate that the width at low level of the SCL signal is longer than the width at high level. If 400 kbps (fast mode) is specified, therefore, the width at low level of the SCL output signal will become shorter than the value specified in the I^2C bus specifications. Make sure that the SDRmn[15:9] value satisfies the I^2C bus specifications.

Remarks 1. The value of SDRmn[15:9] is the value of bits 15 to 9 of the SDRmn register (0000001B to 1111111B) and therefore is 1 to 127.

2. m: Unit number, n: Channel number (mn = 00, 01, 03, 10)

The operation clock (f_{MCK}) is determined by serial clock select register m (SPSm) and bit 15 (CKSmn) of serial mode register mn (SMRmn).

Downloaded from Arrow.com.

Table 11-5. Selection of Operation Clock for Simplified I^2C

SMRmn Register	SPS0 Register								Operation Clock (f_{MCK})[Note]	
CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		f_{CLK} = 20 MHz
0	X	X	X	X	0	0	0	0	f_{CLK}	20 MHz
	X	X	X	X	0	0	0	1	$f_{CLK}/2$	10 MHz
	X	X	X	X	0	0	1	0	$f_{CLK}/2^2$	5 MHz
	X	X	X	X	0	0	1	1	$f_{CLK}/2^3$	2.5 MHz
	X	X	X	X	0	1	0	0	$f_{CLK}/2^4$	1.25 MHz
	X	X	X	X	0	1	0	1	$f_{CLK}/2^5$	625 KHz
	X	X	X	X	0	1	1	0	$f_{CLK}/2^6$	312.5 kHz
	X	X	X	X	0	1	1	1	$f_{CLK}/2^7$	156.2 kHz
	X	X	X	X	1	0	0	0	$f_{CLK}/2^8$	78.1 kHz
	X	X	X	X	1	0	0	1	$f_{CLK}/2^9$	39.1 kHz
	X	X	X	X	1	0	1	0	$f_{CLK}/2^{10}$	19.5 kHz
	X	X	X	X	1	0	1	1	$f_{CLK}/2^{11}$	9.77 kHz
	X	X	X	X	1	1	0	0	$f_{CLK}/2^{12}$	4.87 kHz
	X	X	X	X	1	1	0	1	$f_{CLK}/2^{13}$	2.44 kHz
	X	X	X	X	1	1	1	0	$f_{CLK}/2^{14}$	1.22 kHz
	X	X	X	X	1	1	1	1	$f_{CLK}/2^{15}$	610 Hz
1	0	0	0	0	X	X	X	X	f_{CLK}	20 MHz
	0	0	0	1	X	X	X	X	$f_{CLK}/2$	10 MHz
	0	0	1	0	X	X	X	X	$f_{CLK}/2^2$	5 MHz
	0	0	1	1	X	X	X	X	$f_{CLK}/2^3$	2.5 MHz
	0	1	0	0	X	X	X	X	$f_{CLK}/2^4$	1.25 MHz
	0	1	0	1	X	X	X	X	$f_{CLK}/2^5$	625 KHz
	0	1	1	0	X	X	X	X	$f_{CLK}/2^6$	312.5 KHz
	0	1	1	1	X	X	X	X	$f_{CLK}/2^7$	156.2 kHz
	1	0	0	0	X	X	X	X	$f_{CLK}/2^8$	78.1 kHz
	1	0	0	1	X	X	X	X	$f_{CLK}/2^9$	39.1 kHz
	1	0	1	0	X	X	X	X	$f_{CLK}/2^{10}$	19.5 kHz
	1	0	1	1	X	X	X	X	$f_{CLK}/2^{11}$	9.76 kHz
	1	1	0	0	X	X	X	X	$f_{CLK}/2^{12}$	4.87 kHz
	1	1	0	1	X	X	X	X	$f_{CLK}/2^{13}$	2.44 kHz
	1	1	1	0	X	X	X	X	$f_{CLK}/2^{14}$	1.22 kHz
	1	1	1	1	X	X	X	X	$f_{CLK}/2^{15}$	610 Hz

Note When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), do so after having stopped (serial channel stop register m(STm) = 000FH) the operation of the serial array unit (SAU).

Remarks 1. X: don't care

2. m: Unit number, n: Channel number (mn = 00, 01, 03, 10)

Here is an example of setting an IIC transfer rate where $f_{MCK} = f_{CLK}$ = 20 MHz.

IIC Transfer Mode (Desired Transfer Rate)	f_{CLK} = 20 MHz			
	Operation Clock (f_{MCK})	SDRmn[15:9]	Calculated Transfer Rate	Error from Desired Transfer Rate
100 kHz	$f_{CLK}/2$	49	100 kHz	0.0%
400 kHz	f_{CLK}	25	384.6 kHz	3.8%[Note]

Note The error cannot be set to about 0% because the duty ratio of the SCL signal is 50%.

Downloaded from Arrow.com.

11.7.6 Procedure for processing errors that occurred during simplified I²C (IIC00, IIC01, IIC11, IIC20) communication

The procedure for processing errors that occurred during simplified I²C (IIC00, IIC01, IIC11, IIC20) communication is described in Figure 11-110 and Figure 11-111.

<R> **Figure 11-110. Processing Procedure in Case of Overrun Error**

Software Manipulation	Hardware Status	Remark
Reads serial data register mn (SDRmn). →	The BFFmn bit of the SSRmn register is set to 0 and channel n is enabled to receive data.	This is to prevent an overrun error if the next reception is completed during error processing.
Reads serial status register mn (SSRmn).		The error type is identified and the read value is used to clear the error flag.
Writes 1 to serial flag clear trigger register mn (SIRmn). →	The error flag is cleared.	The error only during reading can be cleared, by writing the value read from the SSRmn register to the SIRmn register without modification.

Figure 11-111. Processing Procedure in Case of ACK error in Simplified I²C Mode

Software Manipulation	Hardware Status	Remark
Reads serial status register mn (SSRmn).		Error type is identified and the read value is used to clear error flag.
Writes serial flag clear trigger register mn (SIRmn). →	Error flag is cleared.	Error can be cleared only during reading, by writing the value read from the SSRmn register to the SIRmn register without modification.
Sets the STmn bit of serial channel stop register m (STm) to 1. →	The SEmn bit of serial channel enable status register m (SEm) is set to 0 and channel n stops operation.	Slave is not ready for reception because ACK is not returned. Therefore, a stop condition is created, the bus is released, and communication is started again from the start condition. Or, a restart condition is generated and transmission can be redone from address transmission.
Creates stop condition.		
Creates start condition.		
Sets the SSmn bit of serial channel start register m (SSm) to 1. →	The SEmn bit of serial channel enable status register m (SEm) is set to 1 and channel n is enabled to operate.	

Remark m: Unit number, n: Channel number (mn = 00, 01, 03, 10), r: IIC number (r = 00, 01, 11, 20)

Downloaded from Arrow.com.

CHAPTER 12 SERIAL INTERFACE IICA

12.1 Functions of Serial Interface IICA

Serial interface IICA has the following three modes.

(1) Operation stop mode

This mode is used when serial transfers are not performed. It can therefore be used to reduce power consumption.

(2) I^2C bus mode (multimaster supported)

This mode is used for 8-bit data transfers with several devices via two lines: a serial clock (SCLA0) line and a serial data bus (SDAA0) line.

This mode complies with the I^2C bus format and the master device can generated “start condition”, “address”, “transfer direction specification”, “data”, and “stop condition” data to the slave device, via the serial data bus. The slave device automatically detects these received status and data by hardware. This function can simplify the part of application program that controls the I^2C bus.

Since the SCLA0 and SDAA0 pins are used for open drain outputs, serial interface IICA requires pull-up resistors for the serial clock line and the serial data bus line.

(3) Wakeup mode

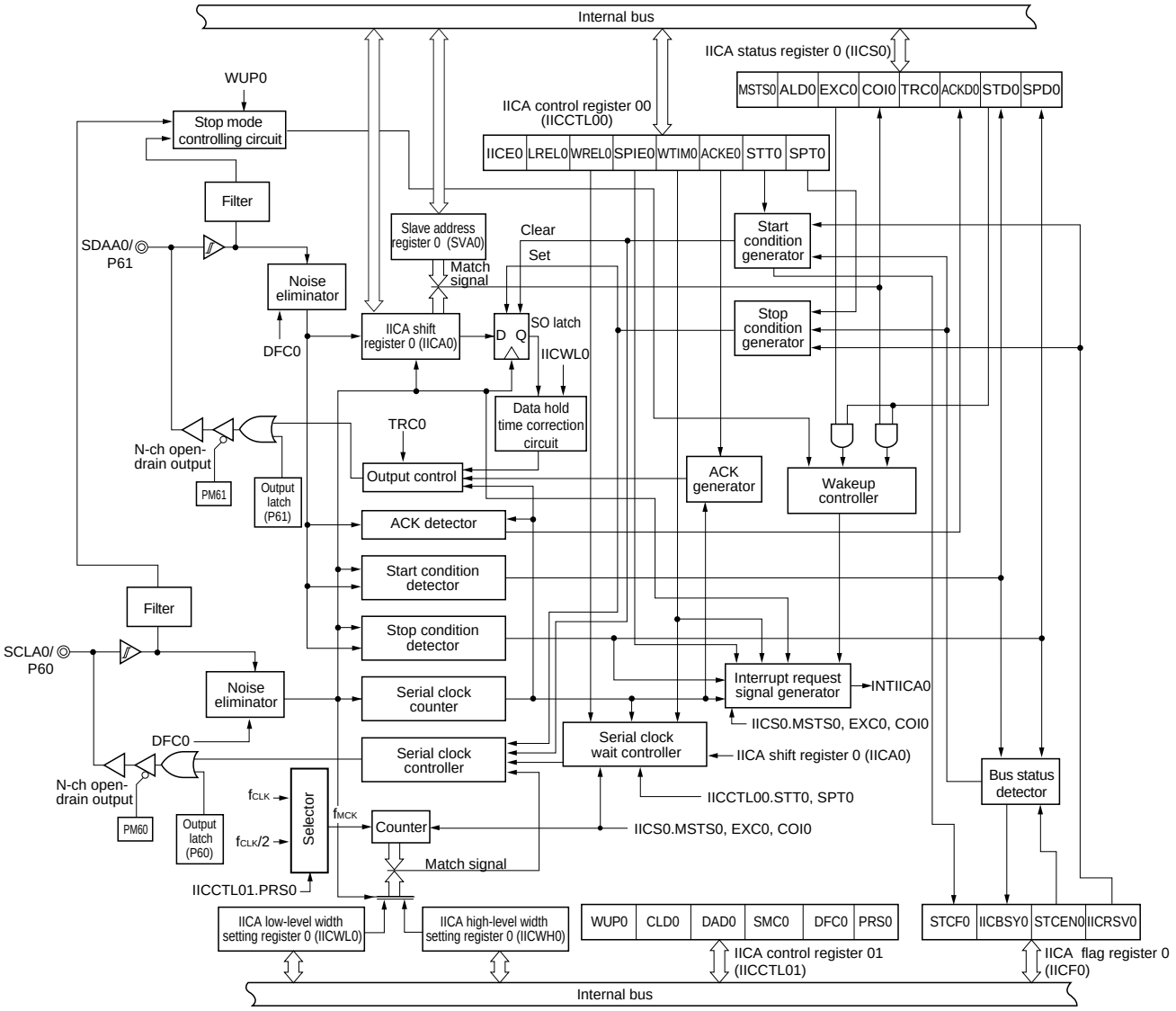
The STOP mode can be released by generating an interrupt request signal (INTIICA0) when an extension code from the master device or a local address has been received while in STOP mode. This can be set by using the WUP0 bit of IICA control register 01 (IICCTL01).

Figure 12-1 shows a block diagram of serial interface IICA.

Downloaded from Arrow.com.

<R>

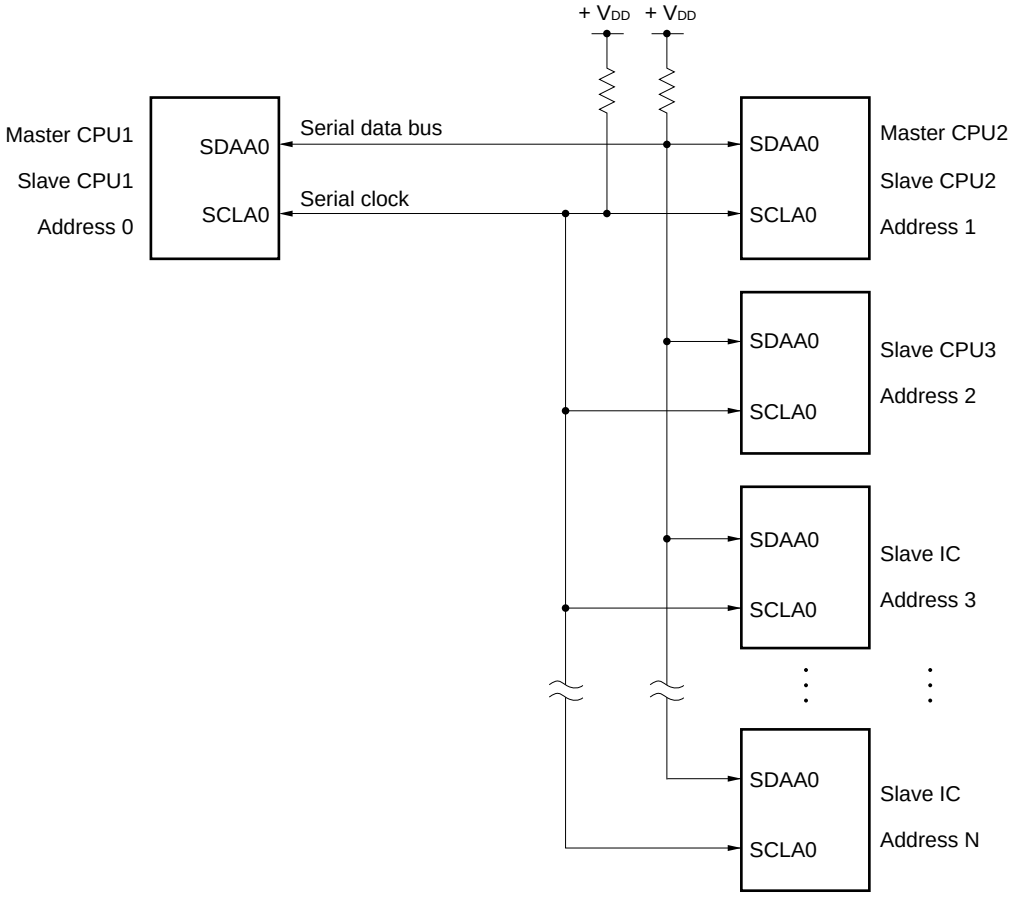
Figure 12-1. Block Diagram of Serial Interface IICA



Downloaded from Arrow.com.

Figure 12-2 shows a serial bus configuration example.

Figure 12-2. Serial Bus Configuration Example Using I²C Bus



Downloaded from Arrow.com.

12.2 Configuration of Serial Interface IICA

Serial interface IICA includes the following hardware.

Table 12-1. Configuration of Serial Interface IICA

Item	Configuration
Registers	IICA shift register 0 (IICA0) Slave address register 0 (SVA0)
Control registers	Peripheral enable register 0 (PER0) IICA control register 00 (IICCTL00) IICA status register 0 (IICS0) IICA flag register 0 (IICF0) IICA control register 01 (IICCTL01) IICA low-level width setting register 0 (IICWL0) IICA high-level width setting register 0 (IICWH0) Port mode register 6 (PM6) Port register 6 (P6)

(1) IICA shift register 0 (IICA0)

The IICA0 register is used to convert 8-bit serial data to 8-bit parallel data and vice versa in synchronization with the serial clock. The IICA0 register can be used for both transmission and reception.

The actual transmit and receive operations can be controlled by writing and reading operations to the IICA0 register.

Cancel the wait state and start data transfer by writing data to the IICA0 register during the wait period.

The IICA0 register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears IICA0 to 00H.

Figure 12-3. Format of IICA Shift Register 0 (IICA0)

Address: FFF50H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IICA0								

Cautions 1. Do not write data to the IICA0 register during data transfer.

2. Write or read the IICA0 register only during the wait period. Accessing the IICA0 register in a communication state other than during the wait period is prohibited. When the device serves as the master, however, the IICA0 register can be written only once after the communication trigger bit (STT0) is set to 1.

3. When communication is reserved, write data to the IICA0 register after the interrupt triggered by a stop condition is detected.

Downloaded from Arrow.com.

(2) Slave address register 0 (SVA0)

This register stores seven bits of local addresses {A6, A5, A4, A3, A2, A1, A0} when in slave mode.

The SVA0 register can be set by an 8-bit memory manipulation instruction.

However, rewriting to this register is prohibited while STD0 = 1 (while the start condition is detected).

Reset signal generation clears the SVA0 register to 00H.

Figure 12-4. Format of Slave Address Register 0 (SVA0)

Address: F0234H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SVA0	A6	A5	A4	A3	A2	A1	A0	0Note

Note Bit 0 is fixed to 0.

(3) SO latch

The SO latch is used to retain the SDAA0 pin's output level.

(4) Wakeup controller

This circuit generates an interrupt request (INTIICA0) when the address received by this register matches the address value set to the slave address register 0 (SVA0) or when an extension code is received.

(5) Serial clock counter

This counter counts the serial clocks that are output or input during transmit/receive operations and is used to verify that 8-bit data was transmitted or received.

(6) Interrupt request signal generator

This circuit controls the generation of interrupt request signals (INTIICA0).

An I^2C interrupt request is generated by the following two triggers.

- Falling edge of eighth or ninth clock of the serial clock (set by the WTIM0 bit)
- Interrupt request generated when a stop condition is detected (set by the SPIE0 bit)

Remark WTIM0 bit: Bit 3 of IICA control register 00 (IICCTL00)
SPIE0 bit: Bit 4 of IICA control register 00 (IICCTL00)

(7) Serial clock controller

In master mode, this circuit generates the clock output via the SCLA0 pin from a sampling clock.

(8) Serial clock wait controller

This circuit controls the wait timing.

(9) ACK generator, stop condition detector, start condition detector, and ACK detector

These circuits generate and detect each status.

(10) Data hold time correction circuit

This circuit generates the hold time for data corresponding to the falling edge of the serial clock.

Downloaded from Arrow.com.

(11) Start condition generator

This circuit generates a start condition when the STT0 bit is set to 1.

However, in the communication reservation disabled status (IICRSV0 bit = 1), when the bus is not released (IICBSY0 bit = 1), start condition requests are ignored and the STCF bit is set to 1.

(12) Stop condition generator

This circuit generates a stop condition when the SPT0 bit is set to 1.

(13) Bus status detector

This circuit detects whether or not the bus is released by detecting start conditions and stop conditions.

However, as the bus status cannot be detected immediately following operation, the initial status is set by the STCEN bit.

Remark STT0 bit: Bit 1 of IICA control register 00 (IICCTL00)
SPT0 bit: Bit 0 of IICA control register 00 (IICCTL00)
IICRSV0 bit: Bit 0 of IICA flag register 0 (IICF0)
IICBSY0 bit: Bit 6 of IICA flag register 0 (IICF0)
STCF0 bit: Bit 7 of IICA flag register 0 (IICF0)
STCEN0 bit: Bit 1 of IICA flag register 0 (IICF0)

Downloaded from Arrow.com.

12.3 Registers Controlling Serial Interface IICA

Serial interface IICA is controlled by the following eight registers.

- Peripheral enable register 0 (PER0)
- IICA control register 00 (IICCTL00)
- IICA flag register 0 (IICF0)
- IICA status register 0 (IICS0)
- IICA control register 01 (IICCTL01)
- IICA low-level width setting register 0 (IICWL0)
- IICA high-level width setting register 0 (IICWH0)
- Port mode register 6 (PM6)
- Port register 6 (P6)

12.3.1 Peripheral enable register 0 (PER0)

This register is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

When serial interface IICA is used, be sure to set bit 4 (IICA0EN) of this register to 1.

The PER0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

<R> **Figure 12-5. Format of Peripheral Enable Register 0 (PER0)**

Address: F00F0H After reset: 00H R/W

Symbol	<7>	6	<5>	<4>	<3>	<2>	1	<0>
PER0	TMKAEN	0	ADCEN	IICA0EN	SAU1EN Note	SAU0EN	0	TAU0EN

IICA0EN	Control of serial interface IICA clock supply
0	Stops clock supply. • SFR used by serial interface IICA cannot be written. • Serial interface IICA is in the reset status.
1	Enables clock supply. • SFR used by serial interface IICA can be read/written.

<R> **Note** 30-pin products only

Cautions 1. When setting serial interface IICA, be sure to set the following registers first while the IICAnEN bit is set to 1. If IICAnEN = 0, the control registers of serial interface IICA are set to their initial values, and writing to them is ignored (except for port mode register 6 (PM6) and port register 6 (P6)).

- **IICA control register 00 (IICCTL00)**
- **IICA flag register 0 (IICF0)**
- **IICA status register 0 (IICS0)**
- **IICA control register 01 (IICCTL01)**
- **IICA low-level width setting register 0 (IICWL0)**
- **IICA high-level width setting register 0 (IICWH0)**

<R> **2. Be sure to clear the following bits to 0.**
20- or 24-pin products: Bits 1, 3, and 6
30-pin products: Bits 1 and 6

Downloaded from Arrow.com.

12.3.2 IICA control register 00 (IICCTL00)

This register is used to enable/stop I^2C operations, set wait timing, and set other I^2C operations.

The IICCTL00 register can be set by a 1-bit or 8-bit memory manipulation instruction. However, set the SPIE0, WTIM0, and ACKE0 bits while IICE0 = 0 or during the wait period. These bits can be set at the same time when the IICE0 bit is set from "0" to "1".

Reset signal generation clears this register to 00H.

Figure 12-6. Format of IICA Control Register 00 (IICCTL00) (1/4)

Address: F0230H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICCTL00	IICE0	LREL0	WREL0	SPIE0	WTIM0	ACKE0	STT0	SPT0

IICE0	I^2C operation enable
0	Stop operation. Reset the IICA status register 0 (IICS0) Note 1. Stop internal operation.
1	Enable operation.
Be sure to set this bit (1) while the SCLA0 and SDAA0 lines are at high level.	
Condition for clearing (IICE0 = 0)	Condition for setting (IICE0 = 1)
• Cleared by instruction • Reset	• Set by instruction

LREL0 Notes 2, 3	Exit from communications
0	Normal operation
1	This exits from the current communications and sets standby mode. This setting is automatically cleared to 0 after being executed. Its uses include cases in which a locally irrelevant extension code has been received. The SCLA0 and SDAA0 lines are set to high impedance. The following flags of IICA control register 00 (IICCTL00) and the IICA status register 0 (IICS0) are cleared to 0. • STT0 • SPT0 • MSTS0 • EXC0 • COI0 • TRC0 • ACKD0 • STD0
The standby mode following exit from communications remains in effect until the following communications entry conditions are met. • After a stop condition is detected, restart is in master mode. • An address match or extension code reception occurs after the start condition.	
Condition for clearing (LREL0 = 0)	Condition for setting (LREL0 = 1)
• Automatically cleared after execution • Reset	• Set by instruction

WREL0 Notes 2, 3	Wait cancellation
0	Do not cancel wait
1	Cancel wait. This setting is automatically cleared after wait is canceled.
When the WREL0 bit is set (wait canceled) during the wait period at the ninth clock pulse in the transmission status (TRC0 = 1), the SDAA0 line goes into the high impedance state (TRC0 = 0).	
Condition for clearing (WREL0 = 0)	Condition for setting (WREL0 = 1)
• Automatically cleared after execution • Reset	• Set by instruction

(**Notes** and **Caution** are listed on the next page.)

Downloaded from Arrow.com.

Notes 1. The IICA status register 0 (IICS0), the STCF and IICBSY bits of the IICA flag register 0 (IICF0), and the CLD0 and DAD0 bits of IICA control register 01 (IICCTL01) are reset.

2. The signal of this bit is invalid while IICE0 is 0.

3. When the LREL0 and WREL0 bits are read, 0 is always read.

Caution If the operation of I^2C is enabled (IICE0 = 1) when the SCLA0 line is high level, the SDAA0 line is low level, and the digital filter is turned on (DFC0 bit of IICCTL01 register = 1), a start condition will be inadvertently detected immediately. In this case, set (1) the LREL0 bit by using a 1-bit memory manipulation instruction immediately after enabling operation of I^2C (IICE0 = 1).

Figure 12-6. Format of IICA Control Register 00 (IICCTL00) (2/4)

SPIE0[Note 1]	Enable/disable generation of interrupt request when stop condition is detected
0	Disable
1	Enable
If the WUP0 bit of IICA control register 01 (IICCTL01) is 1, no stop condition interrupt will be generated even if SPIE0 = 1.	
Condition for clearing (SPIE0 = 0)	Condition for setting (SPIE0 = 1)
• Cleared by instruction • Reset	• Set by instruction

WTIM0[Note 1]	Control of wait and interrupt request generation
0	Interrupt request is generated at the eighth clock's falling edge. Master mode: After output of eight clocks, clock output is set to low level and wait is set. Slave mode: After input of eight clocks, the clock is set to low level and wait is set for master device.
1	Interrupt request is generated at the ninth clock's falling edge. Master mode: After output of nine clocks, clock output is set to low level and wait is set. Slave mode: After input of nine clocks, the clock is set to low level and wait is set for master device.
An interrupt is generated at the falling edge of the ninth clock during address transfer independently of the setting of this bit. The setting of this bit is valid when the address transfer is completed. When in master mode, a wait is inserted at the falling edge of the ninth clock during address transfers. For a slave device that has received a local address, a wait is inserted at the falling edge of the ninth clock after an acknowledge ($\overline{ACK}$) signal is issued. However, when the slave device has received an extension code, a wait is inserted at the falling edge of the eighth clock.	
Condition for clearing (WTIM0 = 0)	Condition for setting (WTIM0 = 1)
• Cleared by instruction • Reset	• Set by instruction

ACKE0 Notes 1, 2	Acknowledgment control
0	Disable acknowledgment.
1	Enable acknowledgment. During the ninth clock period, the SDAA0 line is set to low level.
Condition for clearing (ACKE0 = 0)	Condition for setting (ACKE0 = 1)
• Cleared by instruction • Reset	• Set by instruction

Notes 1. The signal of this bit is invalid while IICE0 is 0. Set this bit during that period.

2. The set value is invalid during address transfer and if the code is not an extension code.
When the device serves as a slave and the addresses match, an acknowledgment is generated regardless of the set value.

Downloaded from Arrow.com.

Figure 12-6. Format of IICA Control Register 00 (IICCTL00) (3/4)

<R>

STT0 Notes 1, 2	Start condition trigger
0	Do not generate a start condition.
1	When bus is released (in standby state, when IICBSY = 0): If this bit is set (1), a start condition is generated (startup as the master). When a third party is communicating: • When communication reservation function is enabled (IICRSV = 0) Functions as the start condition reservation flag. When set to 1, automatically generates a start condition after the bus is released. • When communication reservation function is disabled (IICRSV = 1) Even if this bit is set (1), the STT0 bit is cleared and the STT0 clear flag (STCF) is set (1). No start condition is generated. In the wait state (when master device): Generates a restart condition after releasing the wait.

Cautions concerning set timing
- For master reception: Cannot be set to 1 during transfer. Can be set to 1 only in the waiting period when the ACKE0 bit has been cleared to 0 and slave has been notified of final reception.
- For master transmission: A start condition cannot be generated normally during the acknowledge period. Set to 1 during the wait period that follows output of the ninth clock.
- Cannot be set to 1 at the same time as stop condition trigger (SPT0).
- Setting the STT0 bit to 1 and then setting it again before it is cleared condition is prohibited.

Condition for clearing (STT0 = 0)	Condition for setting (STT0 = 1)
• Cleared by setting the STT0 bit to 1 while communication reservation is prohibited. • Cleared by loss in arbitration • Cleared after start condition is generated by master device • Cleared by LREL0 = 1 (exit from communications) • When IICE0 = 0 (operation stop) • Reset	• Set by instruction

Notes 1. The signal of this bit is invalid while IICE0 is 0.

<R> **2.** The value read from the STT0 bit is always 0.

Remark IICRSV0: Bit 0 of IIC flag register 0 (IICF0)
STCF0: Bit 7 of IIC flag register 0 (IICF0)

Downloaded from Arrow.com.

Figure 12-6. Format of IICA Control Register 00 (IICCTL00) (4/4)

<R>

SPT0 Note	Stop condition trigger
0	Stop condition is not generated.
1	Stop condition is generated (termination of master device's transfer).

Cautions concerning set timing
- For master reception: Cannot be set to 1 during transfer. Can be set to 1 only in the waiting period when the ACKE0 bit has been cleared to 0 and slave has been notified of final reception.
- For master transmission: A stop condition cannot be generated normally during the acknowledge period. Therefore, set it during the wait period that follows output of the ninth clock.
- Cannot be set to 1 at the same time as start condition trigger (STT0).
- The SPT0 bit can be set to 1 only when in master mode.
- When the WTIM0 bit has been cleared to 0, if the SPT0 bit is set to 1 during the wait period that follows output of eight clocks, note that a stop condition will be generated during the high-level period of the ninth clock. The WTIM0 bit should be changed from 0 to 1 during the wait period following the output of eight clocks, and the SPT0 bit should be set to 1 during the wait period that follows the output of the ninth clock.
- Setting the SPT0 bit to 1 and then setting it again before it is cleared condition is prohibited.

Condition for clearing (SPT0 = 0)	Condition for setting (SPT0 = 1)
• Cleared by loss in arbitration • Automatically cleared after stop condition is detected • Cleared by LREL0 = 1 (exit from communications) • When IICE0 = 0 (operation stop) • Reset	• Set by instruction

<R> **Note** The value read from the STT0 bit is always 0.

Caution When bit 3 (TRC0) of the IICA status register 0 (IICS0) is set to 1 (transmission status), bit 5 (WREL0) of IICA control register 00 (IICCTL00) is set to 1 during the ninth clock and wait is canceled, after which the TRC0 bit is cleared (reception status) and the SDAA0 line is set to high impedance. Release the wait performed while the TRC0 bit is 1 (transmission status) by writing to the IICA shift register 0.

Remark Bit 0 (SPT0) becomes 0 when it is read after data setting.

Downloaded from Arrow.com.

12.3.3 IICA status register 0 (IICS0)

This register indicates the status of I^2C.

The IICS0 register is read by a 1-bit or 8-bit memory manipulation instruction only when STT0 = 1 and during the wait period.

Reset signal generation clears this register to 00H.

Caution Reading the IICS0 register while the address match wakeup function is enabled (WUP0 = 1) in STOP mode is prohibited. When the WUP0 bit is changed from 1 to 0 (wakeup operation is stopped), regardless of the INTIICA0 interrupt request, the change in status is not reflected until the next start condition or stop condition is detected. To use the wakeup function, therefore, enable (SPIE0 = 1) the interrupt generated by detecting a stop condition and read the IICS0 register after the interrupt has been detected.

Remark STT0: bit 1 of IICA control register 00 (IICCTL00)
WUP0: bit 7 of IICA control register 01 (IICCTL01)

Figure 12-7. Format of IICA Status Register 0 (IICS0) (1/3)

Address: FFF51H After reset: 00H R

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICS0	MSTS0	ALD0	EXC0	COI0	TRC0	ACKD0	STD0	SPD0

MSTS0	Master status check flag
0	Slave device status or communication standby status
1	Master device communication status

Condition for clearing (MSTS0 = 0)	Condition for setting (MSTS0 = 1)
• When a stop condition is detected • When ALD0 = 1 (arbitration loss) • Cleared by LREL0 = 1 (exit from communications) • When the IICE0 bit changes from 1 to 0 (operation stop) • Reset	• When a start condition is generated

ALD0	Detection of arbitration loss
0	This status means either that there was no arbitration or that the arbitration result was a “win”.
1	This status indicates the arbitration result was a “loss”. The MSTS0 bit is cleared.

Condition for clearing (ALD0 = 0)	Condition for setting (ALD0 = 1)
• Automatically cleared after the IICS0 register is read[Note] • When the IICE0 bit changes from 1 to 0 (operation stop) • Reset	• When the arbitration result is a “loss”.

Note This register is also cleared when a 1-bit memory manipulation instruction is executed for bits other than the IICS0 register. Therefore, when using the ALD0 bit, read the data of this bit before the data of the other bits.

Remark LREL0: Bit 6 of IICA control register 00 (IICCTL00)
IICE0: Bit 7 of IICA control register 00 (IICCTL00)

Downloaded from Arrow.com.

Figure 12-7. Format of IICA Status Register 0 (IICS0) (2/3)

EXC0	Detection of extension code reception
0	Extension code was not received.
1	Extension code was received.

Condition for clearing (EXC0 = 0)	Condition for setting (EXC0 = 1)
• When a start condition is detected • When a stop condition is detected • Cleared by LREL0 = 1 (exit from communications) • When the IICE0 bit changes from 1 to 0 (operation stop) • Reset	• When the higher four bits of the received address data is either “0000” or “1111” (set at the rising edge of the eighth clock).

COI0	Detection of matching addresses
0	Addresses do not match.
1	Addresses match.

Condition for clearing (COI0 = 0)	Condition for setting (COI0 = 1)
• When a start condition is detected • When a stop condition is detected • Cleared by LREL0 = 1 (exit from communications) • When the IICE0 bit changes from 1 to 0 (operation stop) • Reset	• When the received address matches the local address (slave address register 0 (SVA0)) (set at the rising edge of the eighth clock).

TRC0	Detection of transmit/receive status
0	Receive status (other than transmit status). The SDAA0 line is set for high impedance.
1	Transmit status. The value in the SO0 latch is enabled for output to the SDAA0 line (valid starting at the falling edge of the first byte’s ninth clock).

Condition for clearing (TRC0 = 0)	Condition for setting (TRC0 = 1)
<Both master and slave> • When a stop condition is detected • Cleared by LREL0 = 1 (exit from communications) • When the IICE0 bit changes from 1 to 0 (operation stop) • Cleared by WREL0 = 1[Note] (wait cancel) • When the ALD0 bit changes from 0 to 1 (arbitration loss) • Reset • When not used for communication (MSTS0, EXC0, COI0 = 0) <Master> • When “1” is output to the first byte’s LSB (transfer direction specification bit) <Slave> • When a start condition is detected • When “0” is input to the first byte’s LSB (transfer direction specification bit)	<Master> • When a start condition is generated • When 0 (master transmission) is output to the LSB (transfer direction specification bit) of the first byte (during address transfer) <Slave> • When 1 (slave transmission) is input to the LSB (transfer direction specification bit) of the first byte from the master (during address transfer)

Note When bit 3 (TRC0) of the IICA status register 0 (IICS0) is set to 1 (transmission status), bit 5 (WREL0) of IICA control register 00 (IICCTL00) is set to 1 during the ninth clock and wait is canceled, after which the TRC0 bit is cleared (reception status) and the SDAA0 line is set to high impedance. Release the wait performed while the TRC0 bit is 1 (transmission status) by writing to the IICA shift register 0.

Remark LREL0: Bit 6 of IICA control register 00 (IICCTL00)
IICE0: Bit 7 of IICA control register 00 (IICCTL00)

Downloaded from Arrow.com.

Figure 12-7. Format of IICA Status Register 0 (IICS0) (3/3)

ACKD0	Detection of acknowledge ($\overline{ACK}$)
0	Acknowledge was not detected.
1	Acknowledge was detected.

Condition for clearing (ACKD0 = 0)	Condition for setting (ACKD0 = 1)
• When a stop condition is detected • At the rising edge of the next byte's first clock • Cleared by LREL0 = 1 (exit from communications) • When the IICE0 bit changes from 1 to 0 (operation stop) • Reset	• After the SDAA0 line is set to low level at the rising edge of SCLA0 line's ninth clock

STD0	Detection of start condition
0	Start condition was not detected.
1	Start condition was detected. This indicates that the address transfer period is in effect.

Condition for clearing (STD0 = 0)	Condition for setting (STD0 = 1)
• When a stop condition is detected • At the rising edge of the next byte's first clock following address transfer • Cleared by LREL0 = 1 (exit from communications) • When the IICE0 bit changes from 1 to 0 (operation stop) • Reset	• When a start condition is detected

SPD0	Detection of stop condition
0	Stop condition was not detected.
1	Stop condition was detected. The master device's communication is terminated and the bus is released.

Condition for clearing (SPD0 = 0)	Condition for setting (SPD0 = 1)
• At the rising edge of the address transfer byte's first clock following setting of this bit and detection of a start condition • When the WUP0 bit changes from 1 to 0 • When the IICE0 bit changes from 1 to 0 (operation stop) • Reset	• When a stop condition is detected

Remark LREL0: Bit 6 of IICA control register 00 (IICCTL00)
IICE0: Bit 7 of IICA control register 00 (IICCTL00)

12.3.4 IICA flag register 0 (IICF0)

This register sets the operation mode of I^2C and indicates the status of the I^2C bus.

The IICF0 register can be set by a 1-bit or 8-bit memory manipulation instruction. However, the STT0 clear flag (STCF) and I^2C bus status flag (IICBSY) bits are read-only.

The IICRSV bit can be used to enable/disable the communication reservation function.

The STCEN bit can be used to set the initial value of the IICBSY bit.

The IICRSV and STCEN bits can be written only when the operation of I^2C is disabled (bit 7 (IICE0) of IICA control register 00 (IICCTL00) = 0). When operation is enabled, the IICF0 register can be read.

Reset signal generation clears this register to 00H.

Downloaded from Arrow.com.

Figure 12-8. Format of IICA Flag Register 0 (IICF0)

Address: FFF52H After reset: 00H R/W[Note]

Symbol	<7>	<6>	5	4	3	2	<1>	<0>
IICF0	STCF0	IICBSY0	0	0	0	0	STCEN0	IICRSV0

STCF0	STT0 clear flag
0	Generate start condition
1	Start condition generation unsuccessful: clear the STT0 flag

Condition for clearing (STCF0 = 0)	Condition for setting (STCF0 = 1)
- Cleared by STT0 = 1 - When IICE0 = 0 (operation stop) - Reset	- Generating start condition unsuccessful and the STT0 bit cleared to 0 when communication reservation is disabled (IICRSV0 = 1).

IICBSY0	I^2C bus status flag
0	Bus release status (communication initial status when STCEN0 = 1)
1	Bus communication status (communication initial status when STCEN0 = 0)

Condition for clearing (IICBSY0 = 0)	Condition for setting (IICBSY0 = 1)
- Detection of stop condition - When IICE0 = 0 (operation stop) - Reset	- Detection of start condition - Setting of the IICE0 bit when STCEN0 = 0

STCEN0	Initial start enable trigger
0	After operation is enabled (IICE0 = 1), enable generation of a start condition upon detection of a stop condition.
1	After operation is enabled (IICE0 = 1), enable generation of a start condition without detecting a stop condition.

Condition for clearing (STCEN0 = 0)	Condition for setting (STCEN0 = 1)
- Cleared by instruction - Detection of start condition - Reset	- Set by instruction

IICRSV0	Communication reservation function disable bit
0	Enable communication reservation
1	Disable communication reservation

Condition for clearing (IICRSV0 = 0)	Condition for setting (IICRSV0 = 1)
- Cleared by instruction - Reset	- Set by instruction

Note Bits 6 and 7 are read-only.

Cautions 1. Write to the STCEN bit only when the operation is stopped (IICE0 = 0).

2. As the bus release status (IICBSY = 0) is recognized regardless of the actual bus status when STCEN = 1, when generating the first start condition (STT0 = 1), it is necessary to verify that no third party communications are in progress in order to prevent such communications from being destroyed.

3. Write to IICRSV only when the operation is stopped (IICE0 = 0).

Remarks 1. STT0: Bit 1 of IICA control register 00 (IICCTL00)

2. IICE0: Bit 7 of IICA control register 00 (IICCTL00)

Downloaded from Arrow.com.

12.3.5 IICA control register 01 (IICCTL01)

This register is used to set the operation mode of I^2C and detect the statuses of the SCLA0 and SDAA0 pins.

The IICCTL01 register can be set by a 1-bit or 8-bit memory manipulation instruction. However, the CLD0 and DAD0 bits are read-only.

Set the IICCTL01 register, except the WUP0 bit, while operation of I^2C is disabled (bit 7 (IICE0) of IICA control register 00 (IICCTL00) is 0).

Reset signal generation clears this register to 00H.

Figure 12-9. Format of IICA Control Register 01 (IICCTL01) (1/2)

Address: F0231H After reset: 00H R/W[Note 1]

Symbol	7	6	<5>	<4>	<3>	<2>	1	<0>
IICCTL01	WUP0	0	CLD0	DAD0	SMC0	DFC0	0	PRS0

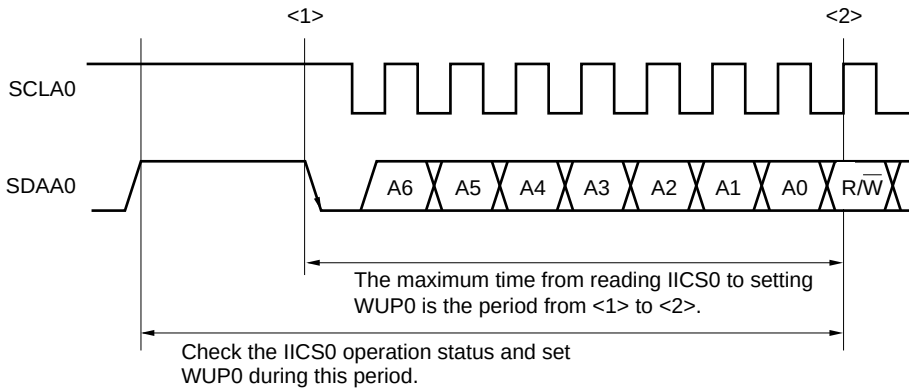
WUP0	Control of address match wakeup
0	Stops operation of address match wakeup function in STOP mode.
1	Enables operation of address match wakeup function in STOP mode.

To shift to STOP mode when WUP0 = 1, execute the STOP instruction at least three cycles of the f_{MCK} clocks after setting (1) the WUP0 bit (see **Figure 12-22 Flow When Setting WUP0 = 1**).
Clear (0) the WUP0 bit after the address has matched or an extension code has been received. The subsequent communication can be entered by the clearing (0) WUP0 bit. (The wait must be released and transmit data must be written after the WUP0 bit has been cleared (0).)
The interrupt timing when the address has matched or when an extension code has been received, while WUP0 = 1, is identical to the interrupt timing when WUP0 = 0. (A delay of the difference of sampling by the clock will occur.) Furthermore, when WUP0 = 1, a stop condition interrupt is not generated even if the SPIE0 bit is set to 1.
When WUP0 = 0 is set by a source other than an interrupt from serial interface IICA, operation as the master device cannot be performed until the subsequent start condition or stop condition is detected. Do not output a start condition by setting (1) the STT0 bit, without waiting for the detection of the subsequent start condition or stop condition.

Condition for clearing (WUP0 = 0)	Condition for setting (WUP0 = 1)
• Cleared by instruction (after address match or extension code reception)	• Set by instruction (when the MSTS0, EXC0, and COI0 bits are “0”, and the STD0 bit also “0” (communication not entered))[Note 2]

Notes 1. Bits 4 and 5 are read-only.

2. The status of the IICA status register 0 (IICS0) must be checked and the WUP0 bit must be set during the period shown below.



Downloaded from Arrow.com.

Figure 12-9. Format of IICA Control Register 01 (IICCTL01) (2/2)

CLD0	Detection of SCLA0 pin level (valid only when IICE0 = 1)
0	The SCLA0 pin was detected at low level.
1	The SCLA0 pin was detected at high level.

Condition for clearing (CLD0 = 0)	Condition for setting (CLD0 = 1)
• When the SCLA0 pin is at low level • When IICE0 = 0 (operation stop) • Reset	• When the SCLA0 pin is at high level

DAD0	Detection of SDAA0 pin level (valid only when IICE0 = 1)
0	The SDAA0 pin was detected at low level.
1	The SDAA0 pin was detected at high level.

Condition for clearing (DAD0 = 0)	Condition for setting (DAD0 = 1)
• When the SDAA0 pin is at low level • When IICE0 = 0 (operation stop) • Reset	• When the SDAA0 pin is at high level

SMC0	Operation mode switching
0	Operates in standard mode (fastest transfer rate: 100 kbps).
1	Operates in fast mode (fastest transfer rate: 400 kbps).

DFC0	Digital filter operation control
0	Digital filter off.
1	Digital filter on.
Use the digital filter in fast mode. In fast mode, the transfer clock does not vary, regardless of the DFC0 bit being set (1) or cleared (0). The digital filter is used for noise elimination in fast mode.	

<R>

PRS0	Control of the IICA operation clock (f_{MCK})
0	Selects f_{CLK} (1 MHz ≤ f_{CLK} ≤ 20 MHz)
1	Selects $f_{CLK}/2$ (20 MHz < f_{CLK})

<R> **Cautions 1. The fastest operation frequency of IICA operation clock (f_{MCK}) is 20 MHz (Max.). When only the f_{MCK} exceeds 20 MHz, set bit 0 (PRSn) of the IICA control register n1 (IICCTLn1) to 1.**

2. Note the minimum f_{CLK} operation frequency when setting the transfer clock.
The minimum f_{CLK} operation frequency for serial interface IICA is determined according to the mode.
Fast mode: f_{CLK} = 3.5 MHz (min.)
Normal mode: f_{CLK} = 1 MHz (min.)

Remark IICE0: Bit 7 of IICA control register 00 (IICCTL00)

Downloaded from Arrow.com.

12.3.6 IICA low-level width setting register 0 (IICWL0)

<R> This register is used to set the low-level width (t_{LOW}) of the SCLA0 pin signal and the SDAAn pin signal that are output by serial interface IICA. The IICWL0 register can be set by an 8-bit memory manipulation instruction.

Set the IICWL0 register while operation of I^2C is disabled (bit 7 (IICE0) of IICA control register 00 (IICCTL00) is 0).

Reset signal generation sets this register to FFH.

For details about setting the IICWL0 register, see **12.4.2 Setting transfer clock by using IICWL0 and IICWH0 registers**.

Figure 12-10. Format of IICA Low-Level Width Setting Register 0 (IICWL0)

Address: F0232H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
IICWL0								

12.3.7 IICA high-level width setting register 0 (IICWH0)

<R> This register is used to set the high-level width (t_{HIGH}) of the SCLA0 pin signal and the SDAAn pin signal that are output by serial interface IICA.

The IICWH0 register can be set by an 8-bit memory manipulation instruction.

Set the IICWH0 register while operation of I^2C is disabled (bit 7 (IICE0) of IICA control register 00 (IICCTL00) is 0).

Reset signal generation sets this register to FFH.

Figure 12-11. Format of IICA High-Level Width Setting Register 0 (IICWH0)

Address: F0233H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
IICWH0								

Remark For how to set the transfer clock on the master side, see **12.4.2 (1)**.

For how to set the transfer clock by using the IICWL0 and IICWH0 registers, see **12.4.2 (2)**.

Downloaded from Arrow.com.

12.3.8 Port mode register 6 (PM6)

This register sets the input/output of port 6 in 1-bit units.

When using the P60/SCLA0 pin as clock I/O and the P61/SDAA0 pin as serial data I/O, clear PM60 and PM61, and the output latches of P60 and P61 to 0.

Set the IICE0 bit (bit 7 of IICA control register 00 (IICCTL00)) to 1 before setting the output mode because the P60/SCLA0 and P61/SDAA0 pins output a low level (fixed) when the IICE0 bit is 0.

The PM6 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to FFH.

Figure 12-12. Format of Port Mode Register 6 (PM6)

Address: FFF26H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM6	1	1	1	1	1	1	PM61	PM60

PM6n	P6n pin I/O mode selection (n = 0, 1)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Downloaded from Arrow.com.

12.4 I^2C Bus Mode Functions

12.4.1 Pin configuration

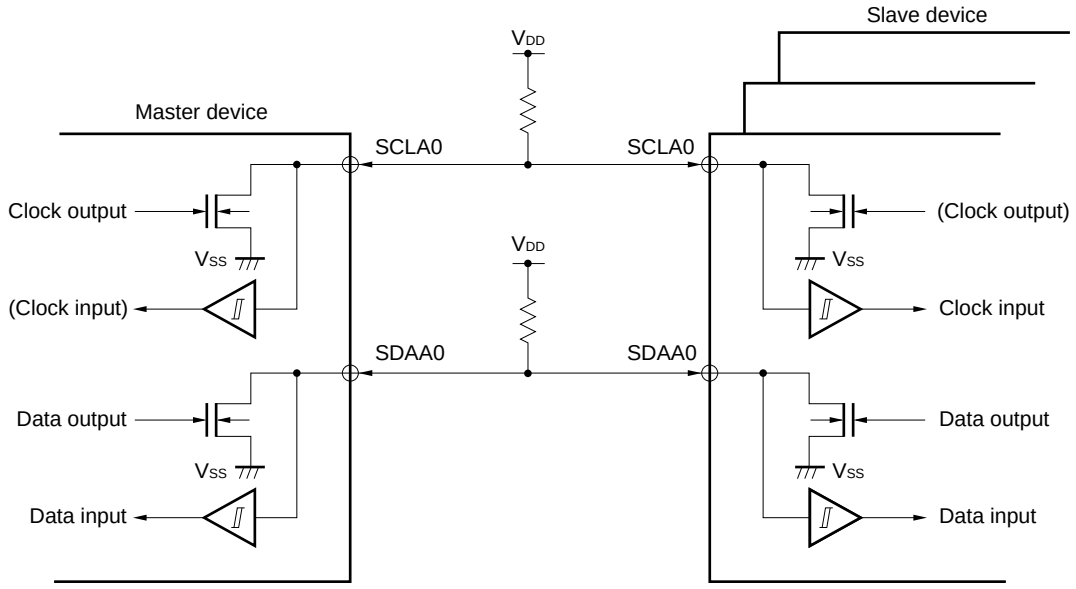
The serial clock pin (SCLA0) and the serial data bus pin (SDAA0) are configured as follows.

(1) SCLA0 .... This pin is used for serial clock input and output.
This pin is an N-ch open-drain output for both master and slave devices. Input is Schmitt input.

(2) SDAA0 .... This pin is used for serial data input and output.
This pin is an N-ch open-drain output for both master and slave devices. Input is Schmitt input.

Since outputs from the serial clock line and the serial data bus line are N-ch open-drain outputs, an external pull-up resistor is required.

Figure 12-13. Pin Configuration Diagram



Downloaded from Arrow.com.

<R> 12.4.2 Setting transfer clock by using IICWL0 and IICWH0 registers

(1) Setting transfer clock on master side

$$\text{Transfer clock} = \frac{f_{MCK}}{IICWL + IICWH + f_{MCK}(t_R + t_F)}$$

At this time, the optimal setting values of the IICWL0 and IICWH0 registers are as follows.
(The fractional parts of all setting values are rounded up.)

- When the fast mode

$$IICWL0 = \frac{0.52}{\text{Transfer clock}} \times f_{MCK}$$
$$IICWH0 = \left(\frac{0.48}{\text{Transfer clock}} - t_R - t_F\right) \times f_{MCK}$$

- When the normal mode

$$IICWL0 = \frac{0.47}{\text{Transfer clock}} \times f_{MCK}$$
$$IICWH0 = \left(\frac{0.53}{\text{Transfer clock}} - t_R - t_F\right) \times f_{MCK}$$

(2) Setting IICWL0 and IICWH0 registers on slave side

(The fractional parts of all setting values are truncated.)

- When the fast mode

$$IICWL0 = 1.3\ \mu s \times f_{MCK}$$
$$IICWH0 = (1.2\ \mu s - t_R - t_F) \times f_{MCK}$$

- When the normal mode

$$IICWL0 = 4.7\ \mu s \times f_{MCK}$$
$$IICWH0 = (5.3\ \mu s - t_R - t_F) \times f_{MCK}$$

Cautions 1. The fastest operation frequency of IICA operation clock (f_{MCK}) is 20 MHz (max.). When only the f_{CLK} exceeds 20 MHz, set bit 0 (PRSn) of the IICA control register n1 (IICCTLn1) to 1.

2. Note the minimum f_{CLK} operation frequency when setting the transfer clock. The minimum f_{CLK} operation frequency for serial interface IICA is determined according to the mode.

Fast mode: f_{CLK} = 3.5 MHz (min.)

Normal mode: f_{CLK} = 1 MHz (min.)

Remarks 1. Calculate the rise time (t_R) and fall time (t_F) of the SDAA0 and SCLA0 signals separately, because they differ depending on the pull-up resistance and wire load.

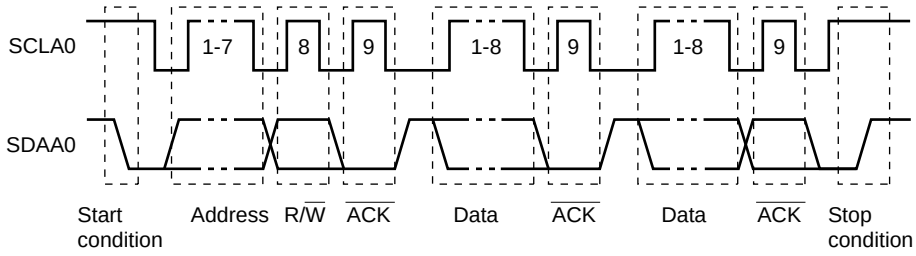
2. IICWL0: IICA low-level width setting register 0
IICWH0: IICA high-level width setting register 0
t_F: SDAA0 and SCLA0 signal falling times
t_R: SDAA0 and SCLA0 signal rising times
f_{MCK}: Serial interface IICA operation clock

Downloaded from Arrow.com.

12.5 I²C Bus Definitions and Control Methods

The following section describes the I²C bus's serial data communication format and the signals used by the I²C bus. Figure 12-14 shows the transfer timing for the "start condition", "address", "data", and "stop condition" output via the I²C bus's serial data bus.

Figure 12-14. I²C Bus Serial Data Transfer Timing



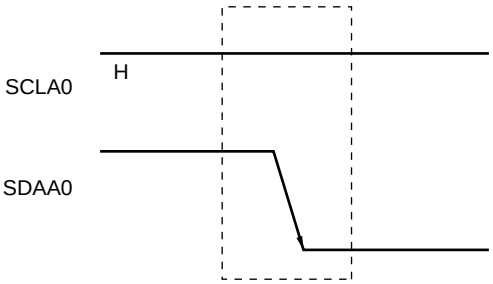
The master device generates the start condition, slave address, and stop condition.
The acknowledge ($\overline{ACK}$) can be generated by either the master or slave device (normally, it is output by the device that receives 8-bit data).

The serial clock (SCLA0) is continuously output by the master device. However, in the slave device, the SCLA0 pin low level period can be extended and a wait can be inserted.

12.5.1 Start conditions

A start condition is met when the SCLA0 pin is at high level and the SDAA0 pin changes from high level to low level. The start conditions for the SCLA0 pin and SDAA0 pin are signals that the master device generates to the slave device when starting a serial transfer. When the device is used as a slave, start conditions can be detected.

Figure 12-15. Start Conditions



A start condition is output when bit 1 (STT0) of IICA control register 00 (IICCTL00) is set (1) after a stop condition has been detected (SPD0: Bit 0 of the IICA status register 0 (IICS0) = 1). When a start condition is detected, bit 1 (STD0) of the IICS0 register is set (1).

Downloaded from Arrow.com.

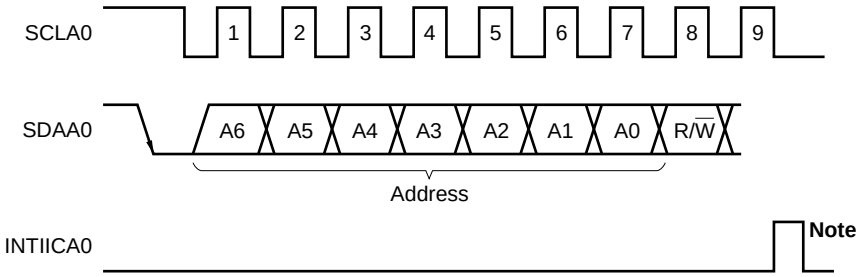
12.5.2 Addresses

The address is defined by the 7 bits of data that follow the start condition.

An address is a 7-bit data segment that is output in order to select one of the slave devices that are connected to the master device via the bus lines. Therefore, each slave device connected via the bus lines must have a unique address.

The slave devices include hardware that detects the start condition and checks whether or not the 7-bit address data matches the data values stored in the slave address register 0 (SVA0). If the address data matches the SVA0 register values, the slave device is selected and communicates with the master device until the master device generates a start condition or stop condition.

Figure 12-16. Address



Note INTIICA0 is not issued if data other than a local address or extension code is received during slave device operation.

Addresses are output when a total of 8 bits consisting of the slave address and the transfer direction described in **12.5.3 Transfer direction specification** are written to the IICA shift register 0 (IICA0). The received addresses are written to the IICA0 register.

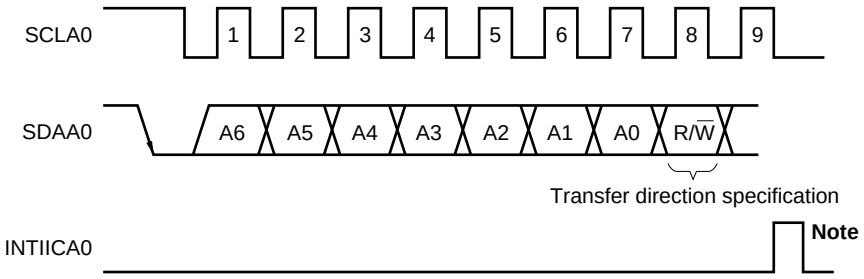
The slave address is assigned to the higher 7 bits of the IICA0 register.

12.5.3 Transfer direction specification

In addition to the 7-bit address data, the master device sends 1 bit that specifies the transfer direction.

When this transfer direction specification bit has a value of “0”, it indicates that the master device is transmitting data to a slave device. When the transfer direction specification bit has a value of “1”, it indicates that the master device is receiving data from a slave device.

Figure 12-17. Transfer Direction Specification



Note INTIICA0 is not issued if data other than a local address or extension code is received during slave device operation.

Downloaded from Arrow.com.

12.5.4 Acknowledge ($\overline{ACK}$)

$\overline{ACK}$ is used to check the status of serial data at the transmission and reception sides.

The reception side returns $\overline{ACK}$ each time it has received 8-bit data.

The transmission side usually receives $\overline{ACK}$ after transmitting 8-bit data. When $\overline{ACK}$ is returned from the reception side, it is assumed that reception has been correctly performed and processing is continued. Whether $\overline{ACK}$ has been detected can be checked by using bit 2 (ACKD0) of the IICA status register 0 (IICS0).

When the master receives the last data item, it does not return $\overline{ACK}$ and instead generates a stop condition. If a slave does not return $\overline{ACK}$ after receiving data, the master outputs a stop condition or restart condition and stops transmission. If $\overline{ACK}$ is not returned, the possible causes are as follows.

<1> Reception was not performed normally.
<2> The final data item was received.
<3> The reception side specified by the address does not exist.

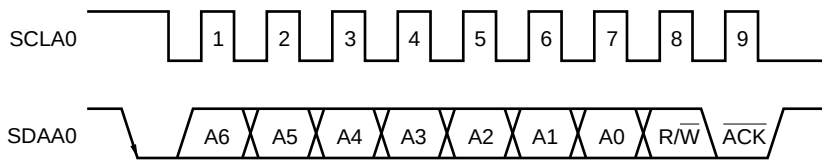
To generate $\overline{ACK}$, the reception side makes the SDAA0 line low at the ninth clock (indicating normal reception).

Automatic generation of $\overline{ACK}$ is enabled by setting bit 2 (ACKE0) of IICA control register 00 (IICCTL00) to 1. Bit 3 (TRC0) of the IICS0 register is set by the data of the eighth bit that follows 7-bit address information. Usually, set the ACKE0 bit to 1 for reception (TRC0 = 0).

If a slave can receive no more data during reception (TRC0 = 0) or does not require the next data item, then the slave must inform the master, by clearing the ACKE0 bit to 0, that it will not receive any more data.

When the master does not require the next data item during reception (TRC0 = 0), it must clear the ACKE0 bit to 0 so that $\overline{ACK}$ is not generated. In this way, the master informs a slave at the transmission side that it does not require any more data (transmission will be stopped).

Figure 12-18. $\overline{ACK}$



When the local address is received, $\overline{ACK}$ is automatically generated, regardless of the value of the ACKE0 bit. When an address other than that of the local address is received, $\overline{ACK}$ is not generated (NACK).

When an extension code is received, $\overline{ACK}$ is generated if the ACKE0 bit is set to 1 in advance.

How $\overline{ACK}$ is generated when data is received differs as follows depending on the setting of the wait timing.

- When 8-clock wait state is selected (bit 3 (WTIM0) of IICCTL00 register = 0):
  By setting the ACKE0 bit to 1 before releasing the wait state, $\overline{ACK}$ is generated at the falling edge of the eighth clock of the SCLA0 pin.
- When 9-clock wait state is selected (bit 3 (WTIM0) of IICCTL00 register = 1):
  $\overline{ACK}$ is generated by setting the ACKE0 bit to 1 in advance.

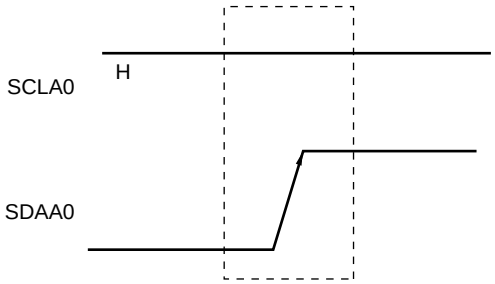
Downloaded from Arrow.com.

12.5.5 Stop condition

When the SCLA0 pin is at high level, changing the SDAA0 pin from low level to high level generates a stop condition.

A stop condition is a signal that the master device generates to the slave device when serial transfer has been completed. When the device is used as a slave, stop conditions can be detected.

Figure 12-19. Stop Condition



A stop condition is generated when bit 0 (SPT0) of IICA control register 00 (IICCTL00) is set to 1. When the stop condition is detected, bit 0 (SPD0) of the IICA status register 0 (IICS0) is set to 1 and INTIICA0 is generated when bit 4 (SPIE0) of the IICCTL00 register is set to 1.

Downloaded from Arrow.com.

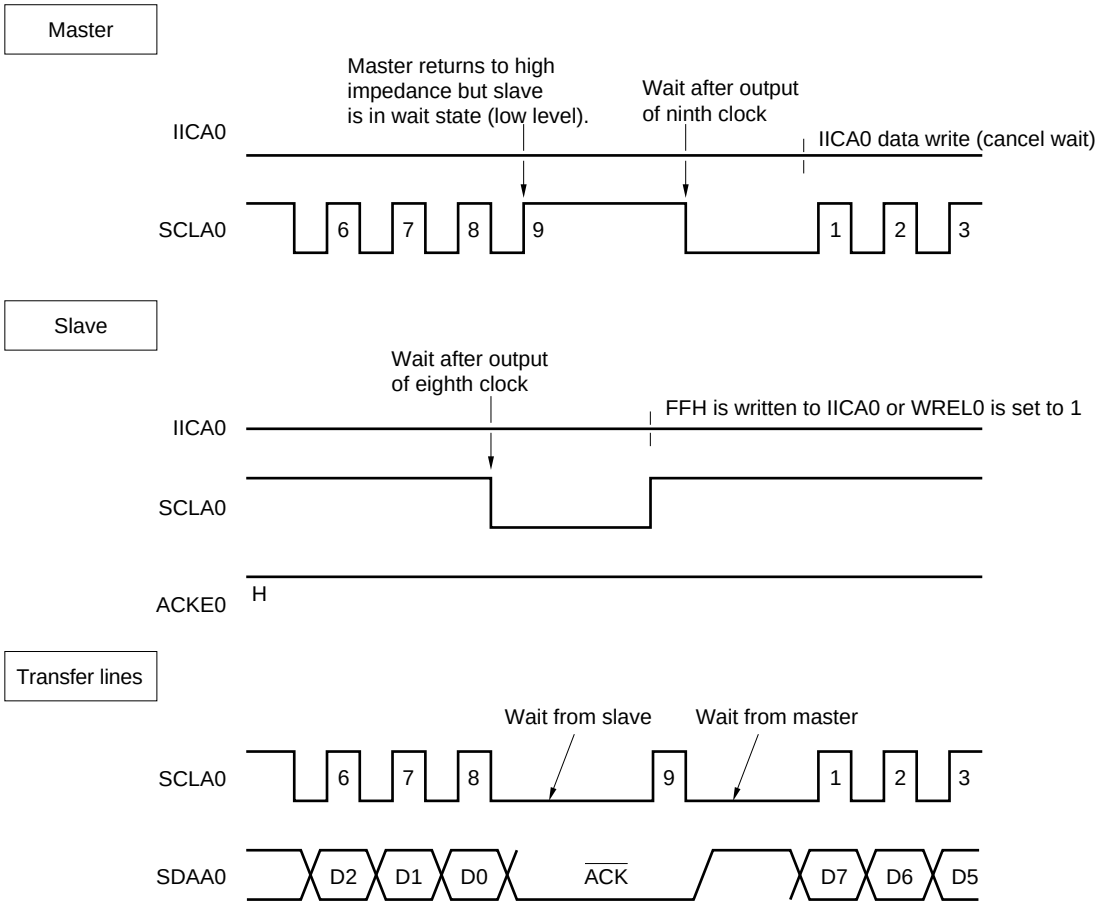
12.5.6 Wait

The wait is used to notify the communication partner that a device (master or slave) is preparing to transmit or receive data (i.e., is in a wait state).

Setting the SCLA0 pin to low level notifies the communication partner of the wait state. When wait state has been canceled for both the master and slave devices, the next data transfer can begin.

Figure 12-20. Wait (1/2)

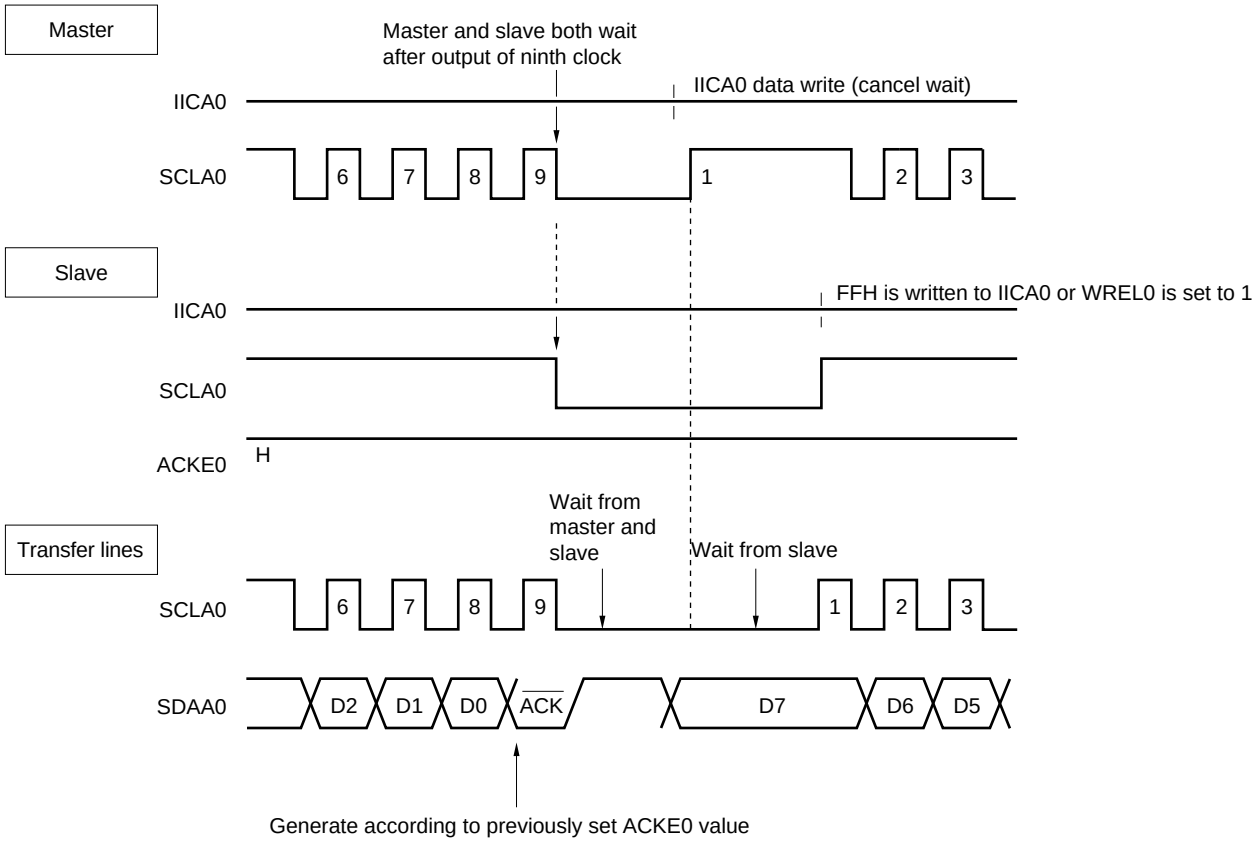
(1) When master device has a nine-clock wait and slave device has an eight-clock wait (master transmits, slave receives, and ACKE0 = 1)



Downloaded from Arrow.com.

Figure 12-20. Wait (2/2)

(2) When master and slave devices both have a nine-clock wait (master transmits, slave receives, and ACKE0 = 1)



Remark ACKE0: Bit 2 of IICA control register 00 (IICCTL00)
WREL0: Bit 5 of IICA control register 00 (IICCTL00)

A wait may be automatically generated depending on the setting of bit 3 (WTIM0) of IICA control register 00 (IICCTL00).

Normally, the receiving side cancels the wait state when bit 5 (WREL0) of the IICCTL00 register is set to 1 or when FFH is written to the IICA shift register 0 (IICA0), and the transmitting side cancels the wait state when data is written to the IICA0 register.

The master device can also cancel the wait state via either of the following methods.

- By setting bit 1 (STT0) of the IICCTL00 register to 1
- By setting bit 0 (SPT0) of the IICCTL00 register to 1

Downloaded from Arrow.com.

12.5.7 Canceling wait

The I^2C usually cancels a wait state by the following processing.

- Writing data to the IICA shift register 0 (IICA0)
- Setting bit 5 (WREL0) of IICA control register 00 (IICCTL00) (canceling wait)
- Setting bit 1 (STT0) of the IICCTL00 register (generating start condition)Note
- Setting bit 0 (SPT0) of the IICCTL00 register (generating stop condition)Note

Note Master only

When the above wait canceling processing is executed, the I^2C cancels the wait state and communication is resumed.

To cancel a wait state and transmit data (including addresses), write the data to the IICA0 register.

To receive data after canceling a wait state, or to complete data transmission, set bit 5 (WREL0) of the IICCTL00 register to 1.

To generate a restart condition after canceling a wait state, set bit 1 (STT0) of the IICCTL00 register to 1.

To generate a stop condition after canceling a wait state, set bit 0 (SPT0) of the IICCTL00 register to 1.

Execute the canceling processing only once for one wait state.

If, for example, data is written to the IICA0 register after canceling a wait state by setting the WREL0 bit to 1, an incorrect value may be output to SDAA0 line because the timing for changing the SDAA0 line conflicts with the timing for writing the IICA0 register.

In addition to the above, communication is stopped if the IICE0 bit is cleared to 0 when communication has been aborted, so that the wait state can be canceled.

If the I^2C bus has deadlocked due to noise, processing is saved from communication by setting bit 6 (LREL0) of the IICCTL00 register, so that the wait state can be canceled.

Caution If a processing to cancel a wait state is executed when WUP0 = 1, the wait state will not be canceled.

Downloaded from Arrow.com.

12.5.8 Interrupt request (INTIICA0) generation timing and wait control

The setting of bit 3 (WTIM0) of IICA control register 00 (IICCTL00) determines the timing by which INTIICA0 is generated and the corresponding wait control, as shown in Table 12-2.

Table 12-2. INTIICA0 Generation Timing and Wait Control

WTIM0	During Slave Device Operation			During Master Device Operation		
	Address	Data Reception	Data Transmission	Address	Data Reception	Data Transmission
0	9 Notes 1, 2	8 Note 2	8 Note 2	9	8	8
1	9 Notes 1, 2	9 Note 2	9 Note 2	9	9	9

Notes 1. The slave device's INTIICA0 signal and wait period occurs at the falling edge of the ninth clock only when there is a match with the address set to the slave address register 0 (SVA0).
At this point, $\overline{ACK}$ is generated regardless of the value set to the IICCTL00 register's bit 2 (ACKE0). For a slave device that has received an extension code, INTIICA0 occurs at the falling edge of the eighth clock.
However, if the address does not match after restart, INTIICA0 is generated at the falling edge of the 9th clock, but wait does not occur.

2. If the received address does not match the contents of the slave address register 0 (SVA0) and extension code is not received, neither INTIICA0 nor a wait occurs.

Remark The numbers in the table indicate the number of the serial clock's clock signals. Interrupt requests and wait control are both synchronized with the falling edge of these clock signals.

(1) During address transmission/reception

- Slave device operation: Interrupt and wait timing are determined depending on the conditions described in Notes 1 and 2 above, regardless of the WTIM0 bit.
- Master device operation: Interrupt and wait timing occur at the falling edge of the ninth clock regardless of the WTIM0 bit.

(2) During data reception

- Master/slave device operation: Interrupt and wait timing are determined according to the WTIM0 bit.

(3) During data transmission

- Master/slave device operation: Interrupt and wait timing are determined according to the WTIM0 bit.

(4) Wait cancellation method

The four wait cancellation methods are as follows.

- Writing data to the IICA shift register 0 (IICA0)
- Setting bit 5 (WREL0) of IICA control register 00 (IICCTL00) (canceling wait)
- Setting bit 1 (STT0) of IICCTL00 register (generating start condition)Note
- Setting bit 0 (SPT0) of IICCTL00 register (generating stop condition)Note

Note Master only.

When an 8-clock wait has been selected (WTIM0 = 0), the presence/absence of $\overline{ACK}$ generation must be determined prior to wait cancellation.

(5) Stop condition detection

INTIICA0 is generated when a stop condition is detected (only when SPIE0 = 1).

Downloaded from Arrow.com.

12.5.9 Address match detection method

In I²C bus mode, the master device can select a particular slave device by transmitting the corresponding slave address.

Address match can be detected automatically by hardware. An interrupt request (INTIICA0) occurs when the address set to the slave address register 0 (SVA0) matches the slave address sent by the master device, or when an extension code has been received.

12.5.10 Error detection

In I²C bus mode, the status of the serial data bus (SDAA0) during data transmission is captured by the IICA shift register 0 (IICA0) of the transmitting device, so the IICA data prior to transmission can be compared with the transmitted IICA data to enable detection of transmission errors. A transmission error is judged as having occurred when the compared data values do not match.

12.5.11 Extension code

(1) When the higher 4 bits of the receive address are either “0000” or “1111”, the extension code reception flag (EXC0) is set to 1 for extension code reception and an interrupt request (INTIICA0) is issued at the falling edge of the eighth clock. The local address stored in the slave address register 0 (SVA0) is not affected.

(2) The settings below are specified if 11110xx0 is transferred from the master by using a 10-bit address transfer when the SVA0 register is set to 11110xx0. Note that INTIICA0 occurs at the falling edge of the eighth clock.

- Higher four bits of data match: EXC0 = 1
- Seven bits of data match: COI0 = 1

Remark EXC0: Bit 5 of IICA status register 0 (IICS0)
COI0: Bit 4 of IICA status register 0 (IICS0)

(3) Since the processing after the interrupt request occurs differs according to the data that follows the extension code, such processing is performed by software.
If the extension code is received while a slave device is operating, then the slave device is participating in communication even if its address does not match.
For example, after the extension code is received, if you do not wish to operate the target device as a slave device, set bit 6 (LREL0) of IICA control register 00 (IICCTL00) to 1 to set the standby mode for the next communication operation.

Table 12-3. Bit Definitions of Major Extension Codes

Slave Address	R/W Bit	Description
0 0 0 0 0 0 0	0	General call address
1 1 1 1 0 x x	0	10-bit slave address specification (during address authentication)
1 1 1 1 0 x x	1	10-bit slave address specification (after address match, when read command is issued)

Remark See the I²C bus specifications issued by NXP Semiconductors for details of extension codes other than those described above.

Downloaded from Arrow.com.

12.5.12 Arbitration

When several master devices simultaneously generate a start condition (when the STT0 bit is set to 1 before the STD0 bit is set to 1), communication among the master devices is performed as the number of clocks are adjusted until the data differs. This kind of operation is called arbitration.

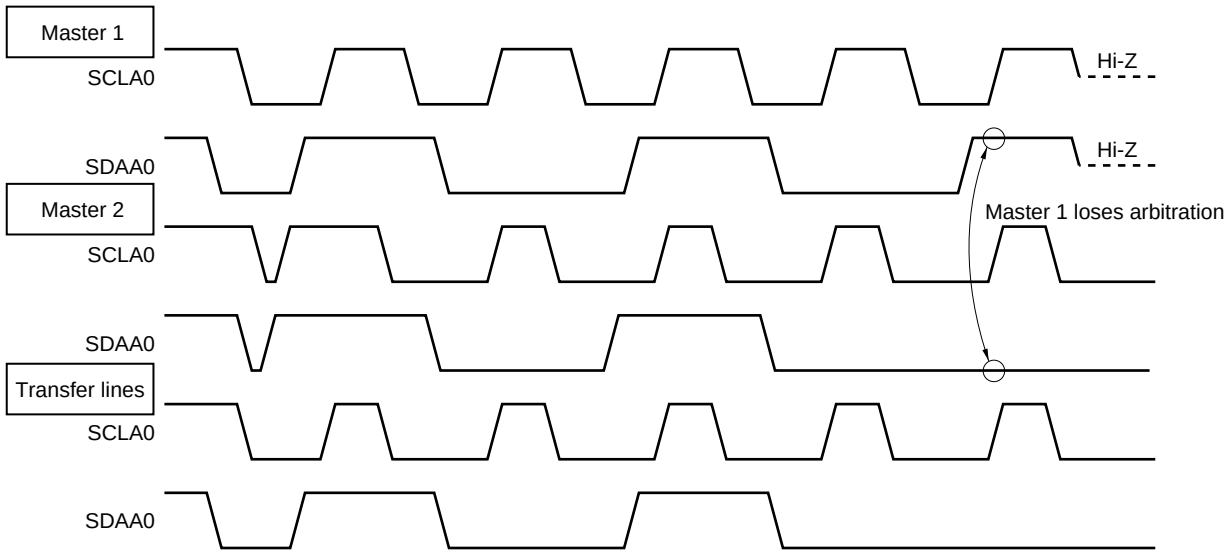
When one of the master devices loses in arbitration, an arbitration loss flag (ALD0) in the IICA status register 0 (IICS0) is set (1) via the timing by which the arbitration loss occurred, and the SCLA0 and SDAA0 lines are both set to high impedance, which releases the bus.

The arbitration loss is detected based on the timing of the next interrupt request (the eighth or ninth clock, when a stop condition is detected, etc.) and the ALD0 = 1 setting that has been made by software.

For details of interrupt request timing, see **12.5.8 Interrupt request (INTIICA0) generation timing and wait control**.

Remark STD0: Bit 1 of IICA status register 0 (IICS0)
STT0: Bit 1 of IICA control register 00 (IICCTL00)

Figure 12-21. Arbitration Timing Example



Downloaded from Arrow.com.

Table 12-4. Status During Arbitration and Interrupt Request Generation Timing

Status During Arbitration	Interrupt Request Generation Timing
During address transmission	At falling edge of eighth or ninth clock following byte transfer Note 1
Read/write data after address transmission	
During extension code transmission	
Read/write data after extension code transmission	
During data transmission	
During $\overline{\text{ACK}}$ transfer period after data transmission	
When restart condition is detected during data transfer	
When stop condition is detected during data transfer	When stop condition is generated (when SPIE0 = 1) Note 2
When data is at low level while attempting to generate a restart condition	At falling edge of eighth or ninth clock following byte transfer Note 1
When stop condition is detected while attempting to generate a restart condition	When stop condition is generated (when SPIE0 = 1) Note 2
When data is at low level while attempting to generate a stop condition	At falling edge of eighth or ninth clock following byte transfer Note 1
When SCLA0 is at low level while attempting to generate a restart condition	

Notes 1. When the WTIM0 bit (bit 3 of IICA control register 00 (IICCTL00)) = 1, an interrupt request occurs at the falling edge of the ninth clock. When WTIM0 = 0 and the extension code's slave address is received, an interrupt request occurs at the falling edge of the eighth clock.

2. When there is a chance that arbitration will occur, set SPIE0 = 1 for master device operation.

Remark SPIE0: Bit 4 of IICA control register 00 (IICCTL00)

Downloaded from Arrow.com.

12.5.13 Wakeup function

The I^2C bus slave function is a function that generates an interrupt request signal (INTIICA0) when a local address and extension code have been received.

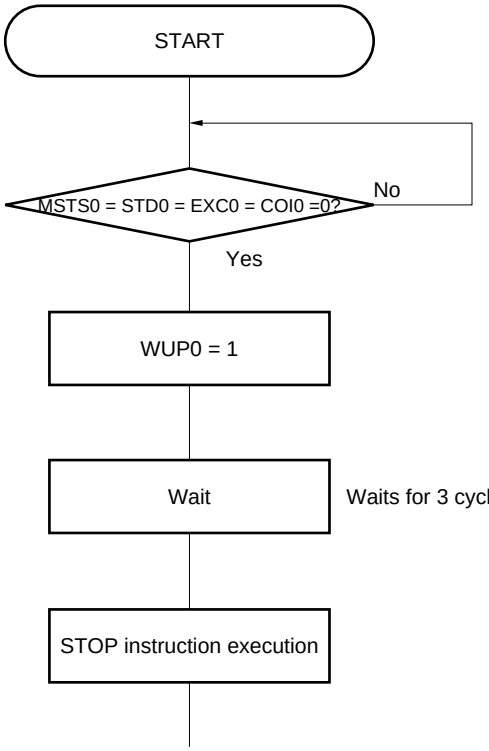
This function makes processing more efficient by preventing unnecessary INTIICA0 signal from occurring when addresses do not match.

When a start condition is detected, wakeup standby mode is set. This wakeup standby mode is in effect while addresses are transmitted due to the possibility that an arbitration loss may change the master device (which has generated a start condition) to a slave device.

To use the wakeup function in the STOP mode, set the WUP0 bit to 1. Addresses can be received regardless of the operation clock. An interrupt request signal (INTIICA0) is also generated when a local address and extension code have been received. Operation returns to normal operation by using an instruction to clear (0) the WUP0 bit after this interrupt has been generated.

Figure 12-22 shows the flow for setting WUP0 = 1 and Figure 12-23 shows the flow for setting WUP0 = 0 upon an address match.

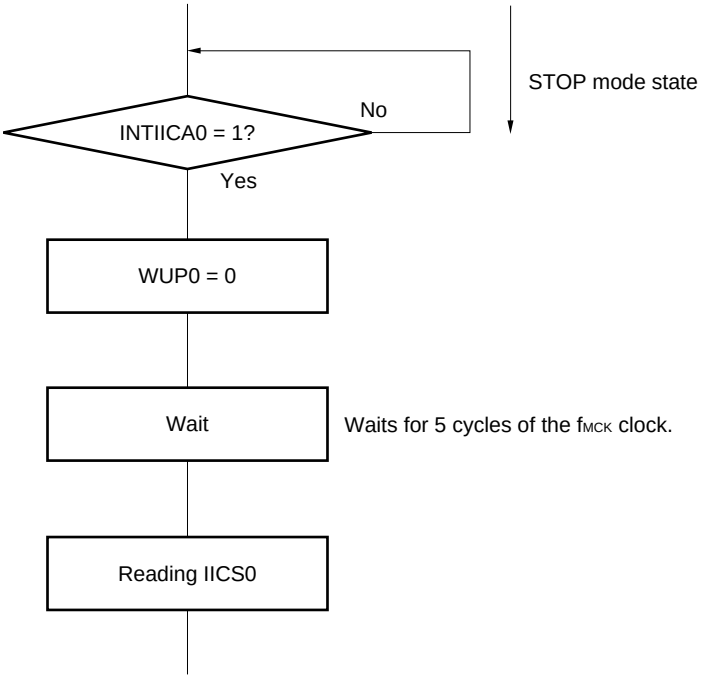
Figure 12-22. Flow When Setting WUP0 = 1



Downloaded from Arrow.com.

Figure 12-23. Flow When Setting WUP0 = 0 upon Address Match (Including Extension Code Reception)

<R>



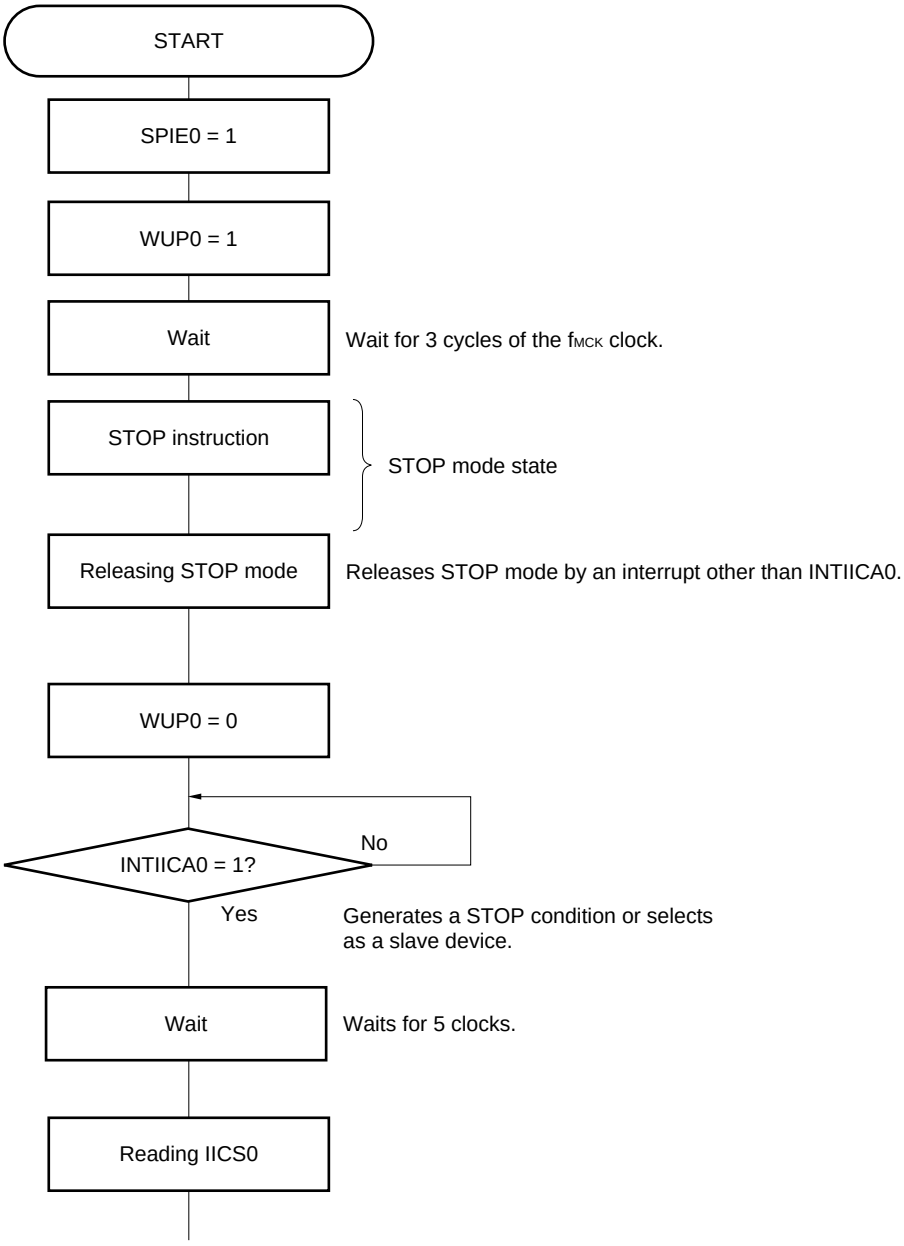
Use the following flows to perform the processing to release the STOP mode other than by an interrupt request (INTIICA0) generated from serial interface IICA.

- Master device operation: Flow shown in Figure 12-24
- Slave device operation: Same as the flow in Figure 12-23

Downloaded from Arrow.com.

Figure 12-24. When Operating as Master Device after Releasing STOP Mode other than by INTIICA0

<R>



Downloaded from Arrow.com.

12.5.14 Communication reservation

(1) When communication reservation function is enabled (bit 0 (IICRSV) of IICA flag register 0 (IICF0) = 0)

To start master device communications when not currently using a bus, a communication reservation can be made to enable transmission of a start condition when the bus is released. There are two modes under which the bus is not used.

- When arbitration results in neither master nor slave operation
- When an extension code is received and slave operation is disabled ($\overline{ACK}$ is not returned and the bus was released by setting bit 6 (LREL0) of IICA control register 00 (IICCTL00) to 1 and saving communication).

If bit 1 (STT0) of the IICCTL00 register is set to 1 while the bus is not used (after a stop condition is detected), a start condition is automatically generated and wait state is set.
If an address is written to the IICA shift register 0 (IICA0) after bit 4 (SPIE0) of the IICCTL00 register was set to 1, and it was detected by generation of an interrupt request signal (INTIICA0) that the bus was released (detection of the stop condition), then the device automatically starts communication as the master. Data written to the IICA0 register before the stop condition is detected is invalid.
When the STT0 bit has been set to 1, the operation mode (as start condition or as communication reservation) is determined according to the bus status.

- If the bus has been released ..... a start condition is generated
- If the bus has not been released (standby mode)..... communication reservation

Check whether the communication reservation operates or not by using the MSTS0 bit (bit 7 of the IICA status register 0 (IICS0)) after the STT0 bit is set to 1 and the wait time elapses.
Use software to secure the wait time calculated by the following expression.

<R>

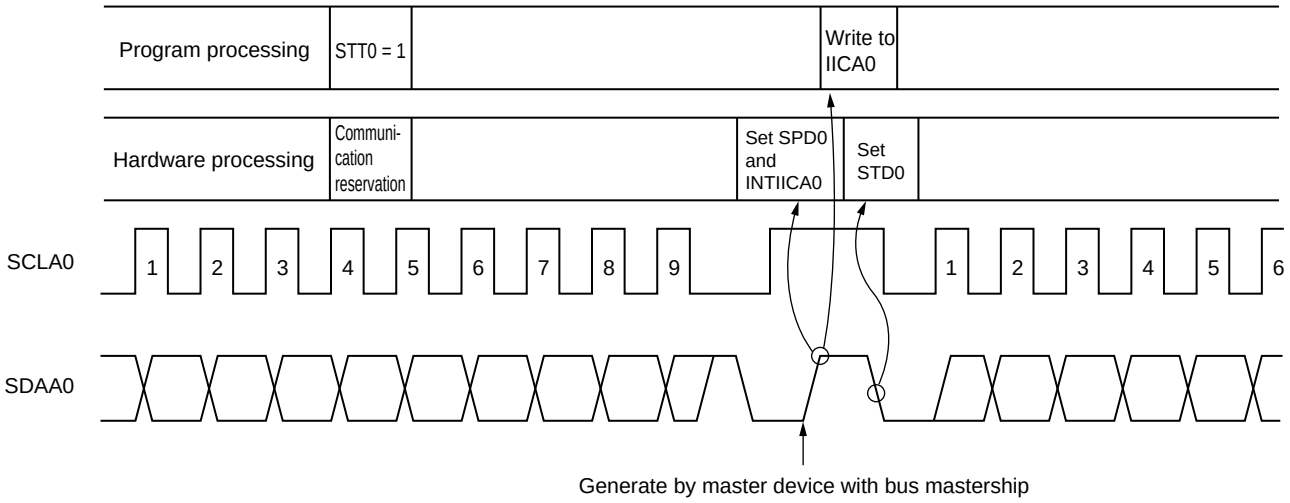
Wait time from setting STT0 = 1 to checking the MSTS0 flag (the number of cycles of the f_{MCK} clock): (IICWL0 setting value + IICWH0 setting value + 4) + $t_F \times 2 \times f_{MCK}$ [clocks]

Remark IICWL0: IICA low-level width setting register 0
IICWH0: IICA high-level width setting register 0
t_F: SDAA0 and SCLA0 signal falling times
f_{MCK}: Serial interface IICA operation clock

Downloaded from Arrow.com.

Figure 12-25 shows the communication reservation timing.

Figure 12-25. Communication Reservation Timing



Remark IICA0: IICA shift register 0
STT0: Bit 1 of IICA control register 00 (IICCTL00)
STD0: Bit 1 of IICA status register 0 (IICS0)
SPD0: Bit 0 of IICA status register 0 (IICS0)

Communication reservations are accepted via the timing shown in Figure 12-26. After bit 1 (STD0) of the IICA status register 0 (IICS0) is set to 1, a communication reservation can be made by setting bit 1 (STT0) of IICA control register 00 (IICCTL00) to 1 before a stop condition is detected.

Figure 12-26. Timing for Accepting Communication Reservations

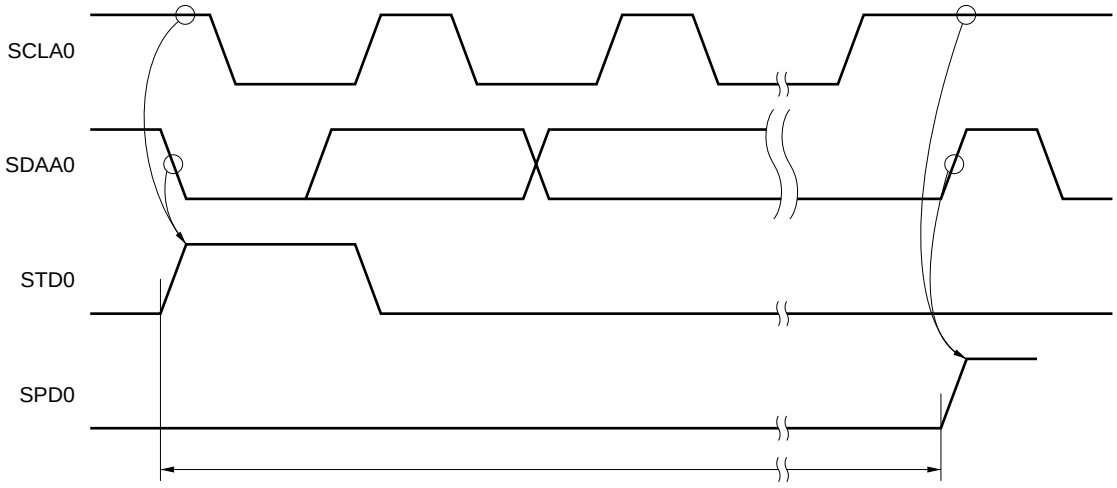
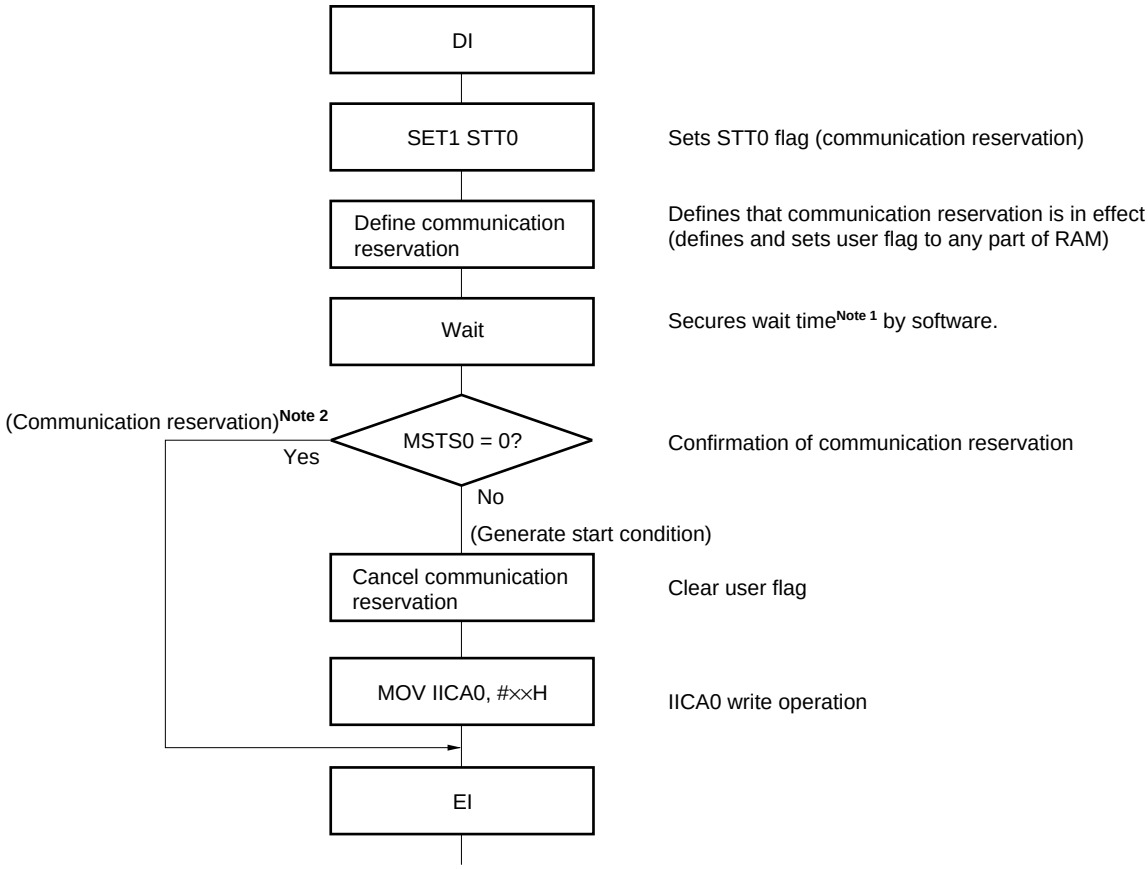


Figure 12-27 shows the communication reservation protocol.

Downloaded from Arrow.com.

Figure 12-27. Communication Reservation Protocol



<R> **Notes 1.** The wait time (the number of cycles of the f_{MCK} clock) is calculated as follows.
(IICWL0 setting value + IICWH0 setting value + 4) + $t_F \times 2 \times f_{MCK}$ [clocks]

2. The communication reservation operation executes a write to the IICA shift register 0 (IICA0) when a stop condition interrupt request occurs.

Remark STT0: Bit 1 of IICA control register 00 (IICCTL00)
MSTS0: Bit 7 of IICA status register 0 (IICS0)
IICA0: IICA shift register 0
IICWL0: IICA low-level width setting register 0
IICWH0: IICA high-level width setting register 0
t_F: SDAA0 and SCLA0 signal falling times
f_{MCK}: Serial interface IICA operation clock

Downloaded from Arrow.com.

(2) When communication reservation function is disabled (bit 0 (IICRSV) of IICA flag register 0 (IICF0) = 1)

When bit 1 (STT0) of IICA control register 00 (IICCTL00) is set to 1 when the bus is not used in a communication during bus communication, this request is rejected and a start condition is not generated. The following two statuses are included in the status where bus is not used.

- When arbitration results in neither master nor slave operation
- When an extension code is received and slave operation is disabled ($\overline{ACK}$ is not returned and the bus was released by setting bit 6 (LREL0) of the IICCTL00 register to 1 and saving communication)

<R> To confirm whether the start condition was generated or request was rejected, check STCF (bit 7 of the IICF0 register). It takes up to 5 cycles of the f_{MCK} clock until the STCF bit is set to 1 after setting STT0 = 1. Therefore, secure the time by software.

Downloaded from Arrow.com.

12.5.15 Cautions

(1) When STCEN = 0

Immediately after I^2C operation is enabled (IICE0 = 1), the bus communication status (IICBSY = 1) is recognized regardless of the actual bus status. When changing from a mode in which no stop condition has been detected to a master device communication mode, first generate a stop condition to release the bus, then perform master device communication.

When using multiple masters, it is not possible to perform master device communication when the bus has not been released (when a stop condition has not been detected).

Use the following sequence for generating a stop condition.

<1> Set IICA control register 01 (IICCTL01).
<2> Set bit 7 (IICE0) of IICA control register 00 (IICCTL00) to 1.
<3> Set bit 0 (SPT0) of the IICCTL00 register to 1.

(2) When STCEN = 1

Immediately after I^2C operation is enabled (IICE0 = 1), the bus released status (IICBSY = 0) is recognized regardless of the actual bus status. To generate the first start condition (STT0 = 1), it is necessary to confirm that the bus has been released, so as to not disturb other communications.

(3) If other I^2C communications are already in progress

If I^2C operation is enabled and the device participates in communication already in progress when the SDAA0 pin is low and the SCLA0 pin is high, the macro of I^2C recognizes that the SDAA0 pin has gone low (detects a start condition). If the value on the bus at this time can be recognized as an extension code, $\overline{ACK}$ is returned, but this interferes with other I^2C communications. To avoid this, start I^2C in the following sequence.

<1> Clear bit 4 (SPIE0) of the IICCTL00 register to 0 to disable generation of an interrupt request signal (INTIICA0) when the stop condition is detected.
<2> Set bit 7 (IICE0) of the IICCTL00 register to 1 to enable the operation of I^2C.
<3> Wait for detection of the start condition.
<R> <4> Set bit 6 (LREL0) of the IICCTL00 register to 1 before $\overline{ACK}$ is returned (4 to 72 cycles of the f_{MCK} clocks after setting the IICE0 bit to 1), to forcibly disable detection.

(4) Setting the STT0 and SPT0 bits (bits 1 and 0 of the IICCTL00 register) again after they are set and before they are cleared to 0 is prohibited.

(5) When transmission is reserved, set the SPIE0 bit (bit 4 of the IICTL0 register) to 1 so that an interrupt request is generated when the stop condition is detected. Transfer is started when communication data is written to the IICA shift register 0 (IICA0) after the interrupt request is generated. Unless the interrupt is generated when the stop condition is detected, the device stops in the wait state because the interrupt request is not generated when communication is started. However, it is not necessary to set the SPIE0 bit to 1 when the MSTS0 bit (bit 7 of the IICA status register 0 (IICS0)) is detected by software.

Downloaded from Arrow.com.

12.5.16 Communication operations

The following shows three operation procedures with the flowchart.

(1) Master operation in single master system

The flowchart when using the RL78/G12 as the master in a single master system is shown below.

This flowchart is broadly divided into the initial settings and communication processing. Execute the initial settings at startup. If communication with the slave is required, prepare the communication and then execute communication processing.

(2) Master operation in multimaster system

In the I^2C bus multimaster system, whether the bus is released or used cannot be judged by the I^2C bus specifications when the bus takes part in a communication. Here, when data and clock are at a high level for a certain period (1 frame), the RL78/G12 takes part in a communication with bus released state.

This flowchart is broadly divided into the initial settings, communication waiting, and communication processing. The processing when the RL78/G12 looses in arbitration and is specified as the slave is omitted here, and only the processing as the master is shown. Execute the initial settings at startup to take part in a communication. Then, wait for the communication request as the master or wait for the specification as the slave. The actual communication is performed in the communication processing, and it supports the transmission/reception with the slave and the arbitration with other masters.

(3) Slave operation

An example of when the RL78/G12 is used as the I^2C bus slave is shown below.

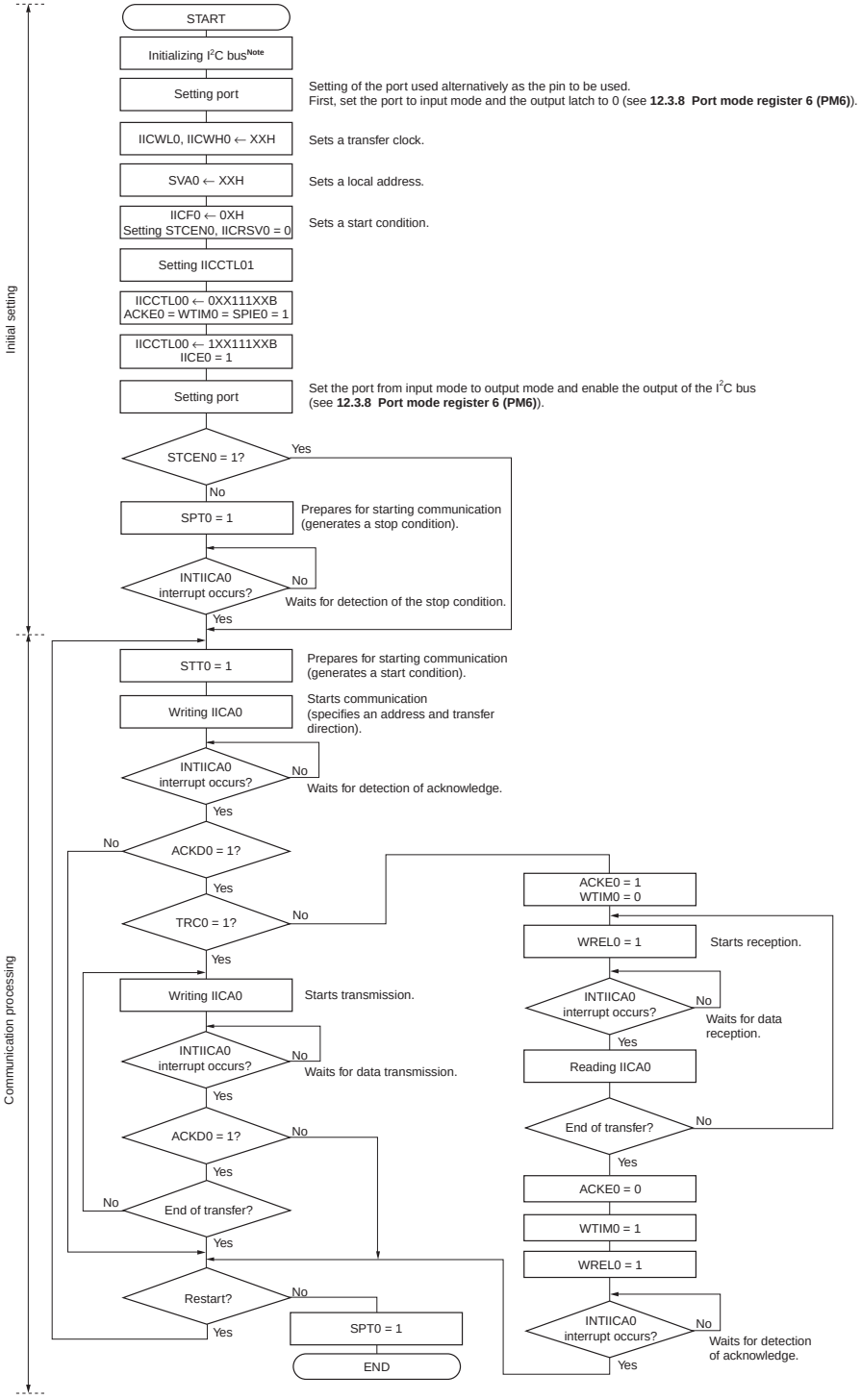
When used as the slave, operation is started by an interrupt. Execute the initial settings at startup, then wait for the INTIICA0 interrupt occurrence (communication waiting). When an INTIICA0 interrupt occurs, the communication status is judged and its result is passed as a flag over to the main processing.

By checking the flags, necessary communication processing is performed.

Downloaded from Arrow.com.

(1) Master operation in single-master system

Figure 12-28. Master Operation in Single-Master System



<R>

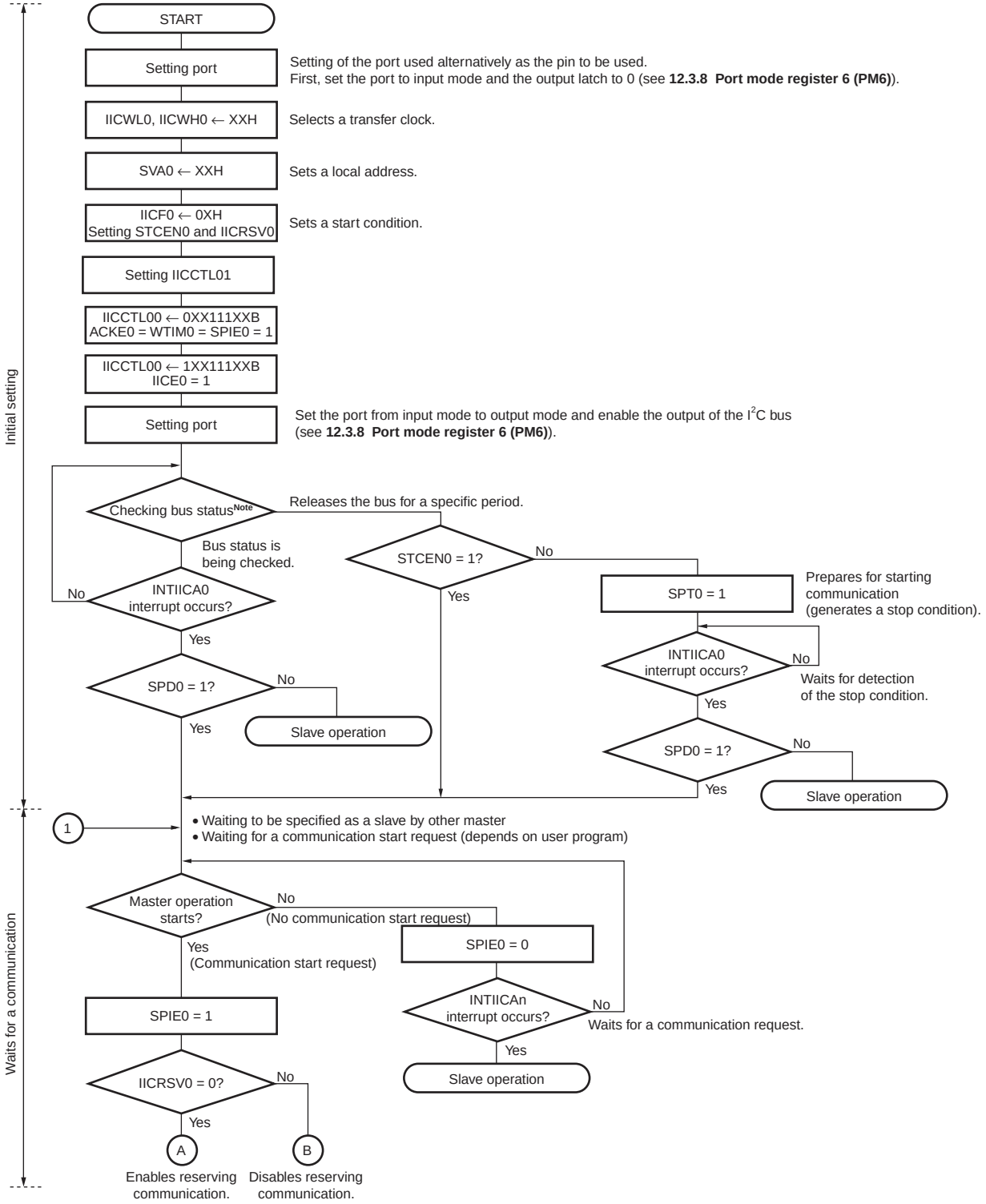
Note Release (SCLA0 and SDAA0 pins = high level) the I²C bus in conformance with the specifications of the product that is communicating. If EEPROM is outputting a low level to the SDAA0 pin, for example, set the SCLA0 pin in the output port mode, and output a clock pulse from the output port until the SDAA0 pin is constantly at high level.

Remark Conform to the specifications of the product that is communicating, with respect to the transmission and reception formats.

Downloaded from Arrow.com.

(2) Master operation in multi-master system

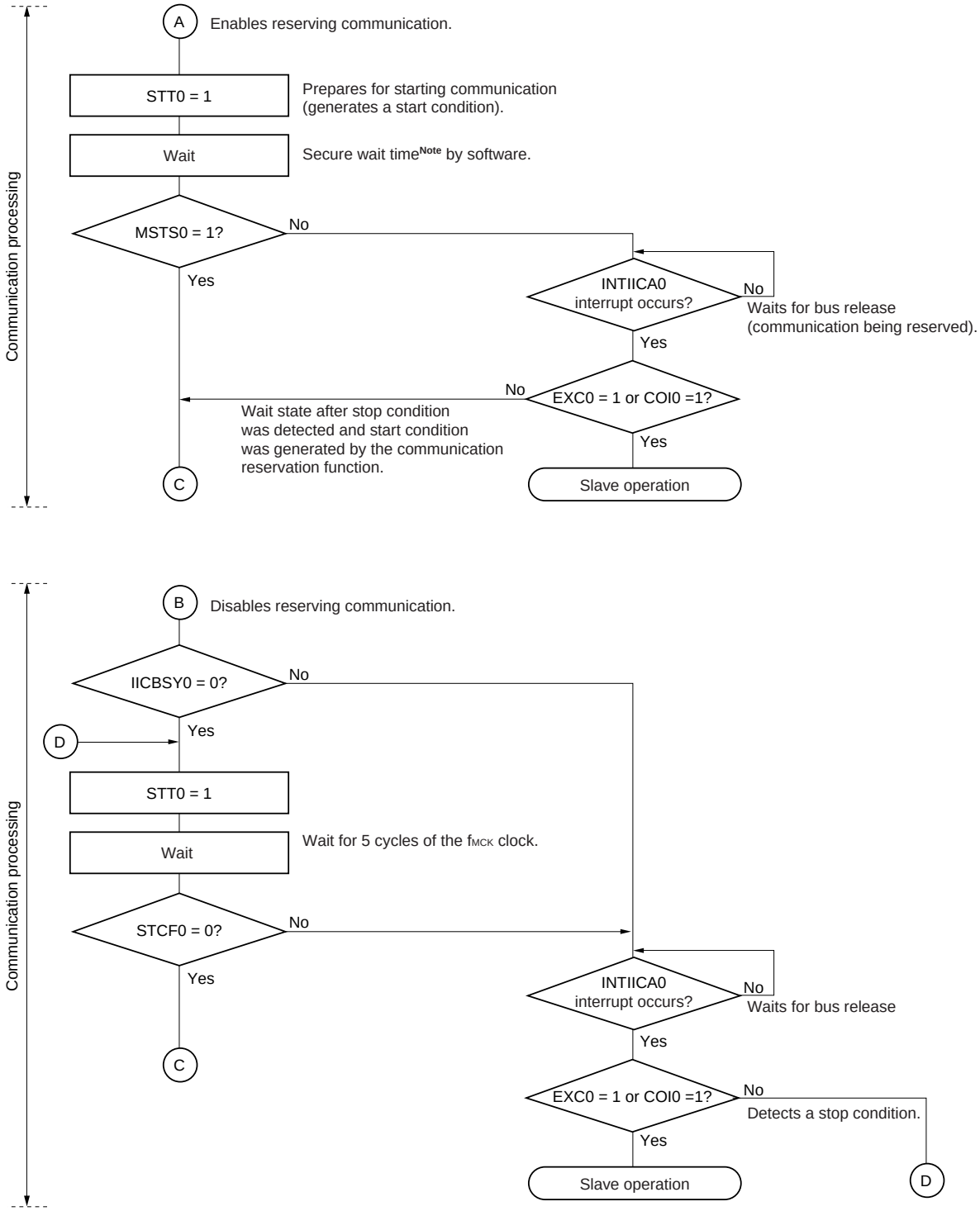
Figure 12-29. Master Operation in Multi-Master System (1/3)



Note Confirm that the bus is released (CLD0 bit = 1, DAD0 bit = 1) for a specific period (for example, for a period of one frame). If the SDAA0 pin is constantly at low level, decide whether to release the I²C bus (SCLA0 and SDAA0 pins = high level) in conformance with the specifications of the product that is communicating.

Downloaded from Arrow.com.

Figure 12-29. Master Operation in Multi-Master System (2/3)



Note The wait time is calculated as follows.

(IICWL0 setting value + IICWH0 setting value + 4) × f_{CLK} + t_F × 2 [clocks]

Remark IICWL0: IICA low-level width setting register 0

IICWH0: IICA high-level width setting register 0

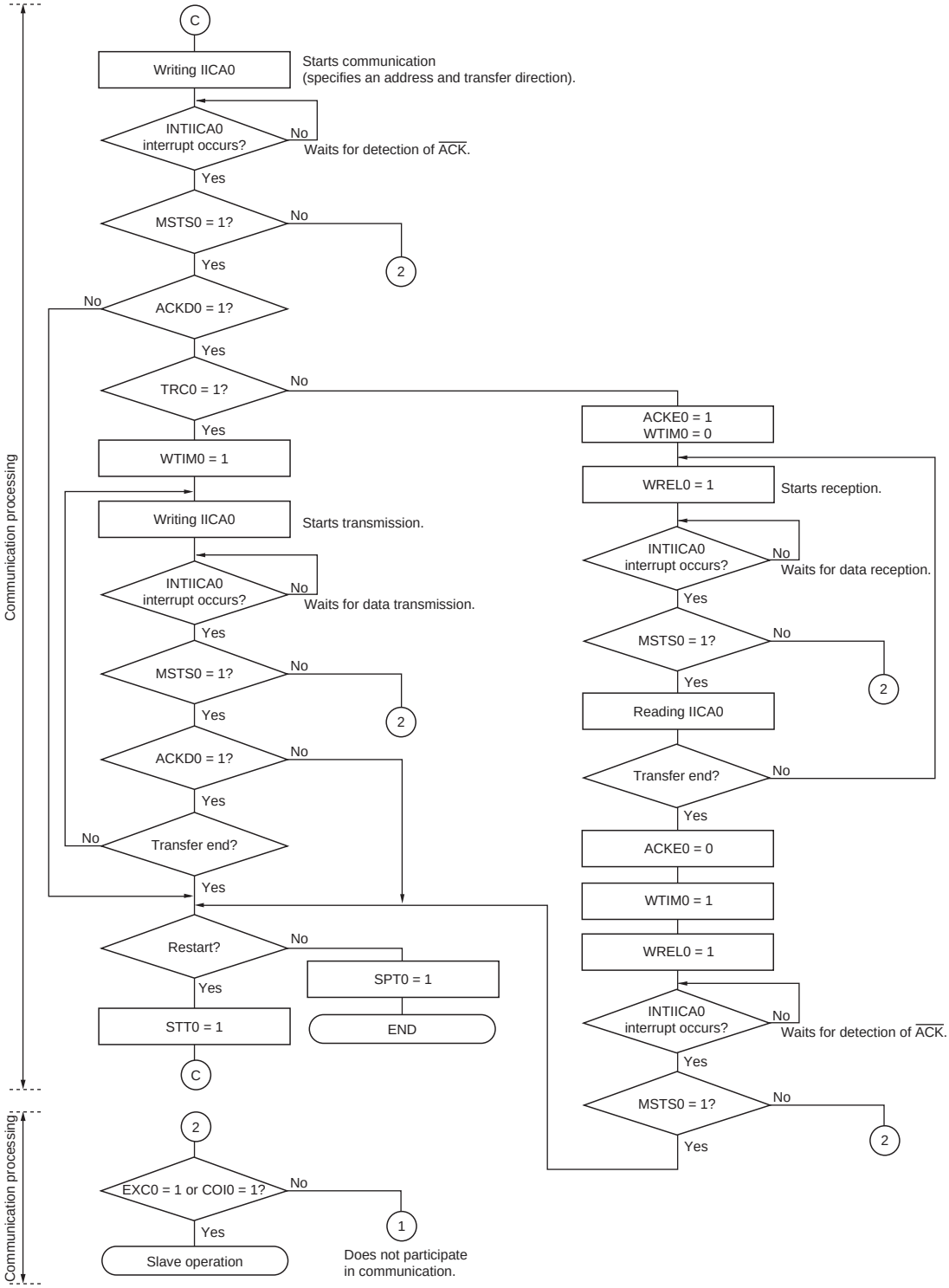
t_F: SDAA0 and SCLA0 signal falling times

f_{CLK}: CPU/peripheral hardware clock frequency

f_{MCK}: Serial interface IICA operation clock

Downloaded from Arrow.com.

Figure 12-29. Master Operation in Multi-Master System (3/3)



<R>

Remarks 1. Conform to the specifications of the product that is communicating, with respect to the transmission and reception formats.

2. To use the device as a master in a multi-master system, read the MSTS0 bit each time interrupt INTIICA0 has occurred to check the arbitration result.

3. To use the device as a slave in a multi-master system, check the status by using the IICA status register 0 (IICS0) and IICA flag register 0 (IICF0) each time interrupt INTIICA0 has occurred, and determine the processing to be performed next.

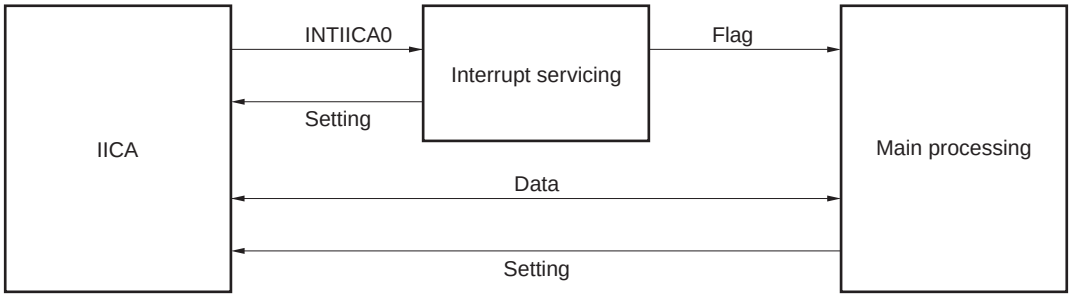
Downloaded from Arrow.com.

(3) Slave operation

The processing procedure of the slave operation is as follows.

Basically, the slave operation is event-driven. Therefore, processing by the INTIICA0 interrupt (processing that must substantially change the operation status such as detection of a stop condition during communication) is necessary.

In the following explanation, it is assumed that the extension code is not supported for data communication. It is also assumed that the INTIICA0 interrupt servicing only performs status transition processing, and that actual data communication is performed by the main processing.



Therefore, data communication processing is performed by preparing the following three flags and passing them to the main processing instead of INTIICA0.

<1> Communication mode flag

This flag indicates the following two communication statuses.

- Clear mode: Status in which data communication is not performed
- Communication mode: Status in which data communication is performed (from valid address detection to stop condition detection, no detection of $\overline{\text{ACK}}$ from master, address mismatch)

<2> Ready flag

This flag indicates that data communication is enabled. Its function is the same as the INTIICA0 interrupt for ordinary data communication. This flag is set by interrupt servicing and cleared by the main processing. Clear this flag by interrupt servicing when communication is started. However, the ready flag is not set by interrupt servicing when the first data is transmitted. Therefore, the first data is transmitted without the flag being cleared (an address match is interpreted as a request for the next data).

<3> Communication direction flag

This flag indicates the direction of communication. Its value is the same as the TRC0 bit.

The main processing of the slave operation is explained next.

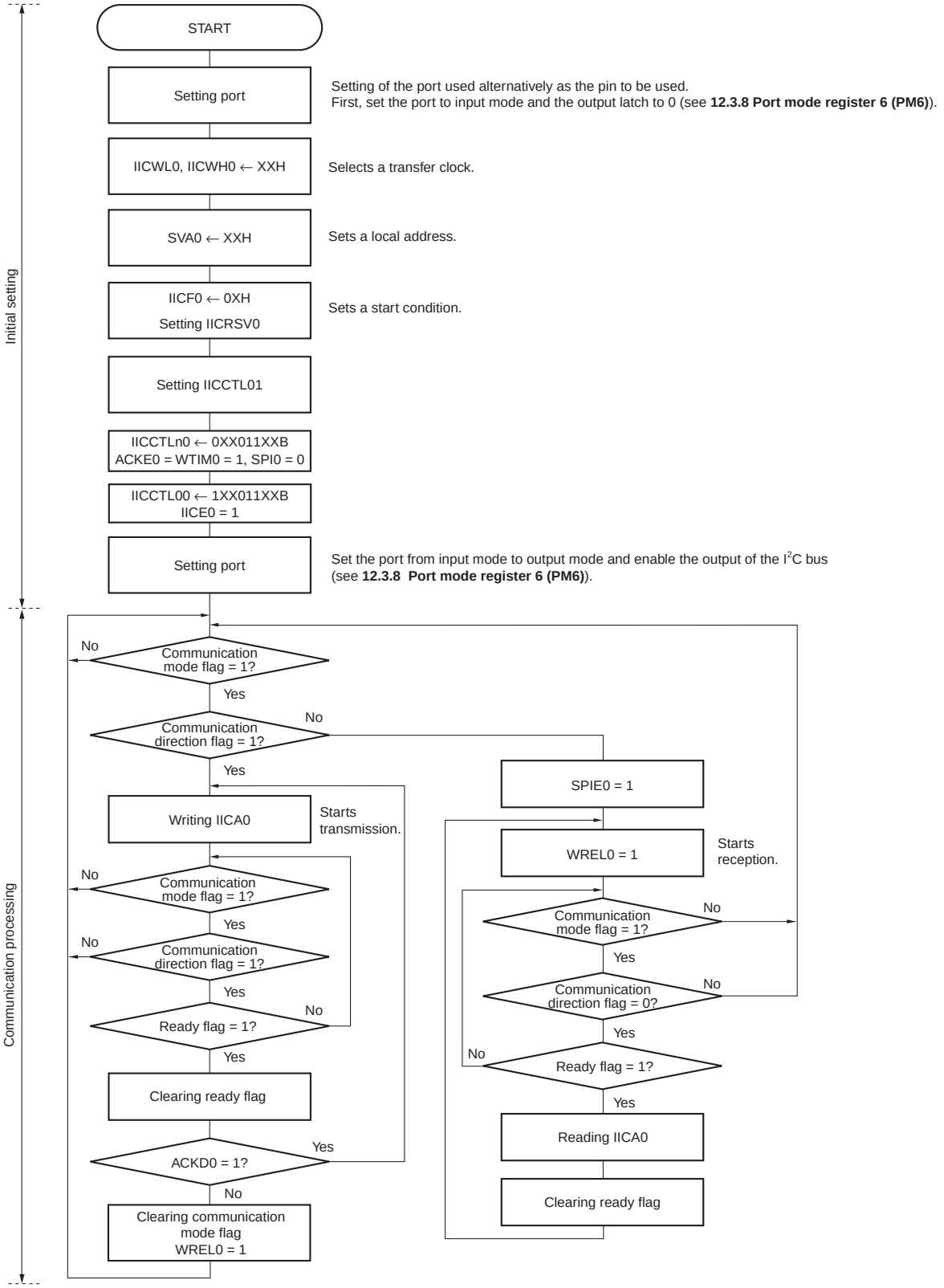
Start serial interface IICA and wait until communication is enabled. When communication is enabled, execute communication by using the communication mode flag and ready flag (processing of the stop condition and start condition is performed by an interrupt. Here, check the status by using the flags).

The transmission operation is repeated until the master no longer returns $\overline{\text{ACK}}$. If $\overline{\text{ACK}}$ is not returned from the master, communication is completed.

For reception, the necessary amount of data is received. When communication is completed, $\overline{\text{ACK}}$ is not returned as the next data. After that, the master generates a stop condition or restart condition. Exit from the communication status occurs in this way.

Downloaded from Arrow.com.

Figure 12-30. Slave Operation Flowchart (1)



<R>

Remark Conform to the specifications of the product that is in communication, regarding the transmission and reception formats.

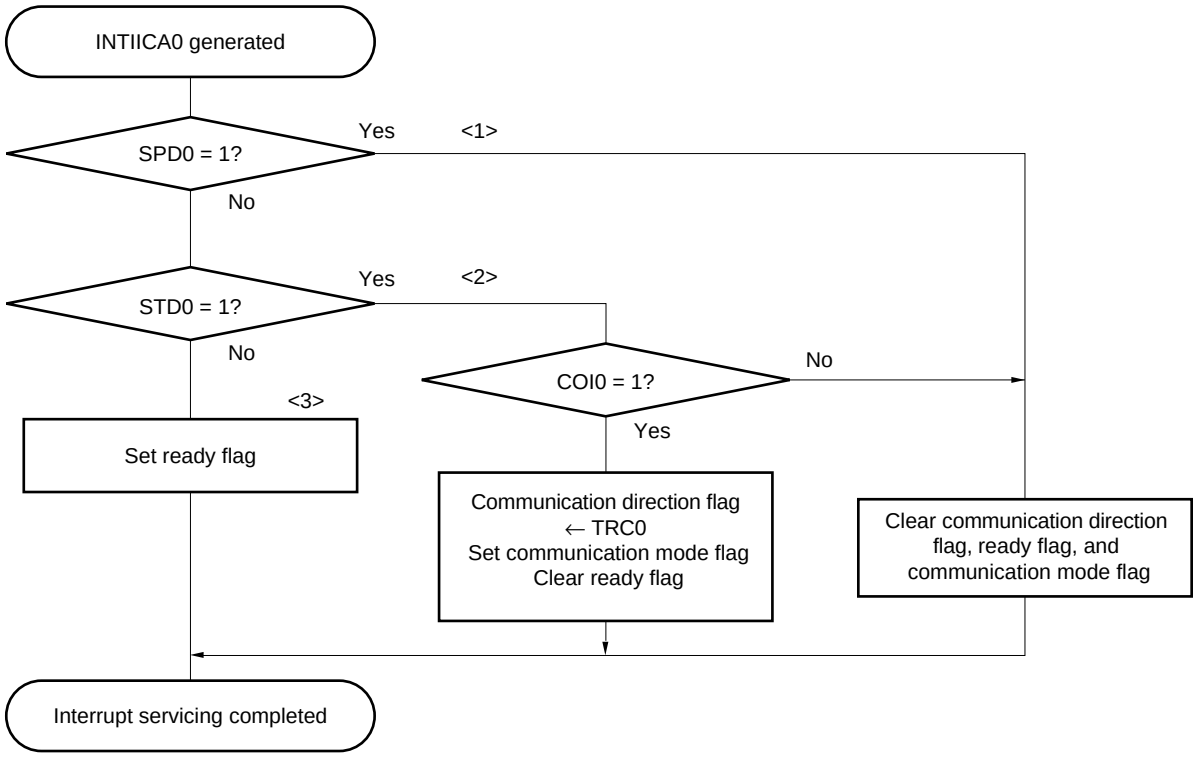
Downloaded from Arrow.com.

An example of the processing procedure of the slave with the INTIICA0 interrupt is explained below (processing is performed assuming that no extension code is used). The INTIICA0 interrupt checks the status, and the following operations are performed.

<1> Communication is stopped if the stop condition is issued.
<2> If the start condition is issued, the address is checked and communication is completed if the address does not match. If the address matches, the communication mode is set, wait is cancelled, and processing returns from the interrupt (the ready flag is cleared).
<3> For data transmit/receive, only the ready flag is set. Processing returns from the interrupt with the I^2C bus remaining in the wait state.

Remark <1> to <3> above correspond to <1> to <3> in Figure 12-31 Slave Operation Flowchart (2).

Figure 12-31. Slave Operation Flowchart (2)



Downloaded from Arrow.com.

12.5.17 Timing of I^2C interrupt request (INTIICA0) occurrence

The timing of transmitting or receiving data and generation of interrupt request signal INTIICA0, and the value of the IICA status register 0 (IICS0) when the INTIICA0 signal is generated are shown below.

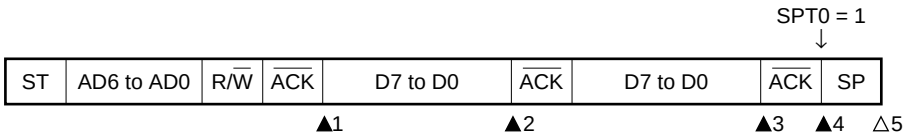
Remark ST: Start condition
AD6 to AD0: Address
R/$\overline{W}$: Transfer direction specification
$\overline{ACK}$: Acknowledge
D7 to D0: Data
SP: Stop condition

Downloaded from Arrow.com.

(1) Master device operation

(a) Start ~ Address ~ Data ~ Data ~ Stop (transmission/reception)

(i) When WTIM0 = 0

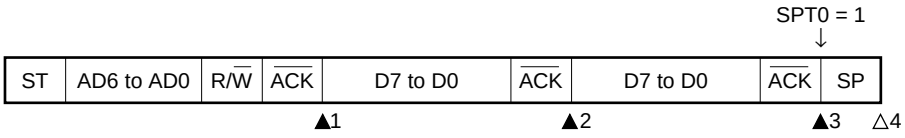


▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000×000B
▲3: IICS0 = 1000×000B (Sets the WTIM0 bit to 1)[Note]
▲4: IICS0 = 1000××00B (Sets the SPT0 bit to 1)[Note]
△5: IICS0 = 00000001B

Note To generate a stop condition, set the WTIM0 bit to 1 and change the timing for generating the INTIICA0 interrupt request signal.

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1



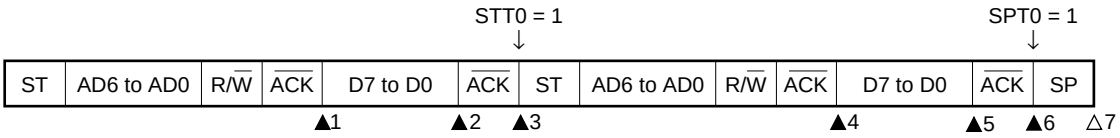
▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000×100B
▲3: IICS0 = 1000××00B (Sets the SPT0 bit to 1)
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

Downloaded from Arrow.com.

(b) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop (restart)

(i) When WTIM0 = 0

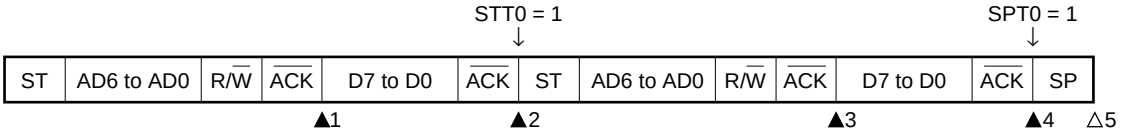


▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000×000B (Sets the WTIM0 bit to 1)Note 1
▲3: IICS0 = 1000××00B (Clears the WTIM0 bit to 0Note 2, sets the STT0 bit to 1)
▲4: IICS0 = 1000×110B
▲5: IICS0 = 1000×000B (Sets the WTIM0 bit to 1)Note 3
▲6: IICS0 = 1000××00B (Sets the SPT0 bit to 1)
△7: IICS0 = 00000001B

Notes 1. To generate a start condition, set the WTIM0 bit to 1 and change the timing for generating the INTIICA0 interrupt request signal.
2. Clear the WTIM0 bit to 0 to restore the original setting.
3. To generate a stop condition, set the WTIM0 bit to 1 and change the timing for generating the INTIICA0 interrupt request signal.

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1



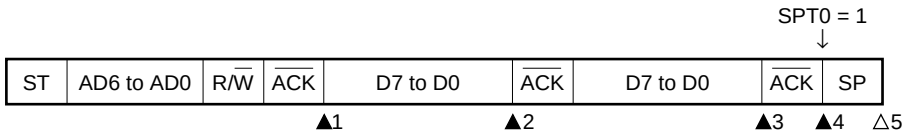
▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000××00B (Sets the STT0 bit to 1)
▲3: IICS0 = 1000×110B
▲4: IICS0 = 1000××00B (Sets the SPT0 bit to 1)
△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

Downloaded from Arrow.com.

(c) Start ~ Code ~ Data ~ Data ~ Stop (extension code transmission)

(i) When WTIM0 = 0

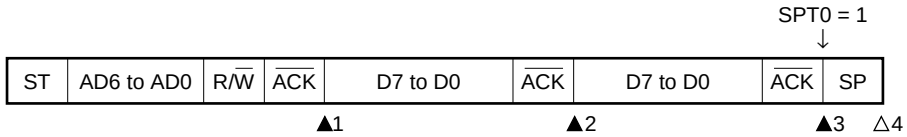


▲1: IICS0 = 1010×110B
▲2: IICS0 = 1010×000B
▲3: IICS0 = 1010×000B (Sets the WTIM0 bit to 1)[Note]
▲4: IICS0 = 1010××00B (Sets the SPT0 bit to 1)
△5: IICS0 = 00000001B

Note To generate a stop condition, set the WTIM0 bit to 1 and change the timing for generating the INTIICA0 interrupt request signal.

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1



▲1: IICS0 = 1010×110B
▲2: IICS0 = 1010×100B
▲3: IICS0 = 1010××00B (Sets the SPT0 bit to 1)
△4: IICS0 = 00001001B

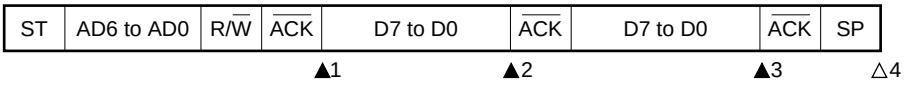
Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

Downloaded from Arrow.com.

(2) Slave device operation (slave address data reception)

(a) Start ~ Address ~ Data ~ Data ~ Stop

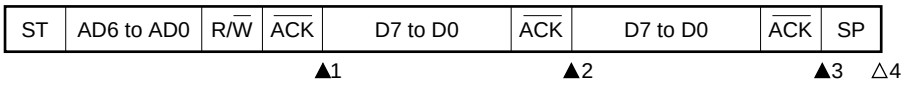
(i) When WTIM0 = 0



▲1: IICS0 = 0001×110B
▲2: IICS0 = 0001×000B
▲3: IICS0 = 0001×000B
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1



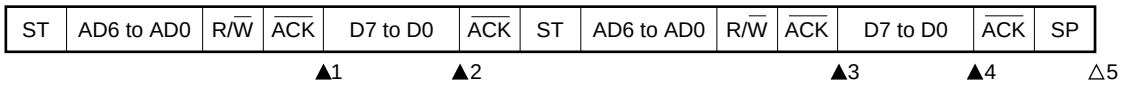
▲1: IICS0 = 0001×110B
▲2: IICS0 = 0001×100B
▲3: IICS0 = 0001××00B
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

Downloaded from Arrow.com.

(b) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

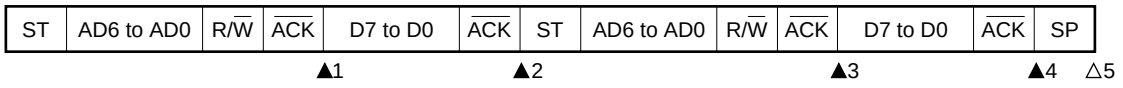
(i) When WTIM0 = 0 (after restart, matches with SVA0)



▲1: IICS0 = 0001×110B
▲2: IICS0 = 0001×000B
▲3: IICS0 = 0001×110B
▲4: IICS0 = 0001×000B
△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1 (after restart, matches with SVA0)



▲1: IICS0 = 0001×110B
▲2: IICS0 = 0001××00B
▲3: IICS0 = 0001×110B
▲4: IICS0 = 0001××00B
△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

Downloaded from Arrow.com.

(c) Start ~ Address ~ Data ~ Start ~ Code ~ Data ~ Stop

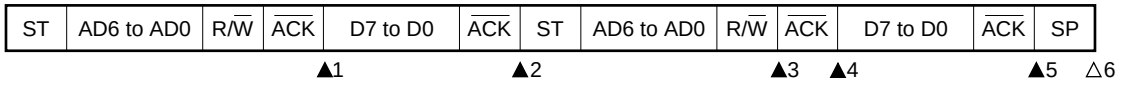
(i) When WTIM0 = 0 (after restart, does not match address (= extension code))



▲1: IICS0 = 0001×110B
▲2: IICS0 = 0001×000B
▲3: IICS0 = 0010×010B
▲4: IICS0 = 0010×000B
△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1 (after restart, does not match address (= extension code))



▲1: IICS0 = 0001×110B
▲2: IICS0 = 0001××00B
▲3: IICS0 = 0010×010B
▲4: IICS0 = 0010×110B
▲5: IICS0 = 0010××00B
△6: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

Downloaded from Arrow.com.

(d) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

(i) When WTIM0 = 0 (after restart, does not match address (= not extension code))

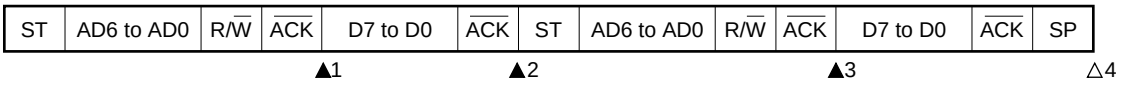


▲1 ▲2 ▲3 △4

▲1: IICS0 = 0001×110B
▲2: IICS0 = 0001×000B
▲3: IICS0 = 00000×10B
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1 (after restart, does not match address (= not extension code))



▲1 ▲2 ▲3 △4

▲1: IICS0 = 0001×110B
▲2: IICS0 = 0001××00B
▲3: IICS0 = 00000×10B
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

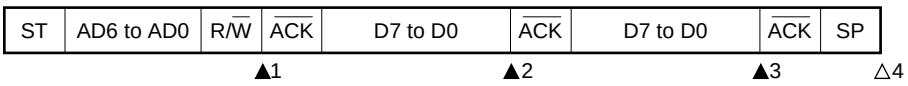
Downloaded from Arrow.com.

(3) Slave device operation (when receiving extension code)

The device is always participating in communication when it receives an extension code.

(a) Start ~ Code ~ Data ~ Data ~ Stop

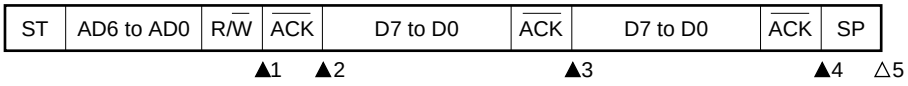
(i) When WTIM0 = 0



▲1: IICS0 = 0010×010B
▲2: IICS0 = 0010×000B
▲3: IICS0 = 0010×000B
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1



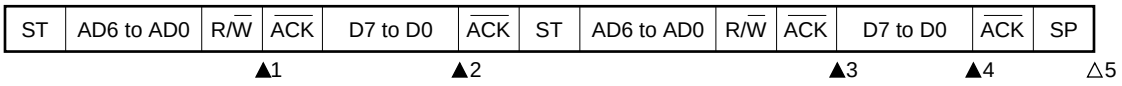
▲1: IICS0 = 0010×010B
▲2: IICS0 = 0010×110B
▲3: IICS0 = 0010×100B
▲4: IICS0 = 0010××00B
△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

Downloaded from Arrow.com.

(b) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

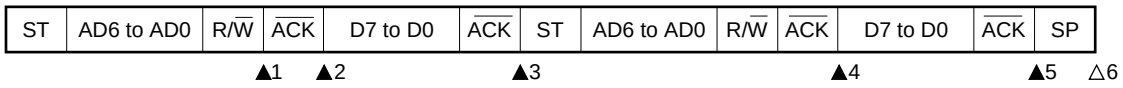
(i) When WTIM0 = 0 (after restart, matches SVA0)



▲1: IICS0 = 0010×010B
▲2: IICS0 = 0010×000B
▲3: IICS0 = 0001×110B
▲4: IICS0 = 0001×000B
△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1 (after restart, matches SVA0)



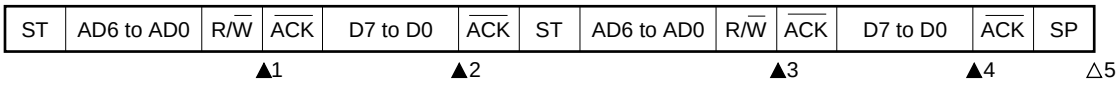
▲1: IICS0 = 0010×010B
▲2: IICS0 = 0010×110B
▲3: IICS0 = 0010××00B
▲4: IICS0 = 0001×110B
▲5: IICS0 = 0001××00B
△6: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

Downloaded from Arrow.com.

(c) Start ~ Code ~ Data ~ Start ~ Code ~ Data ~ Stop

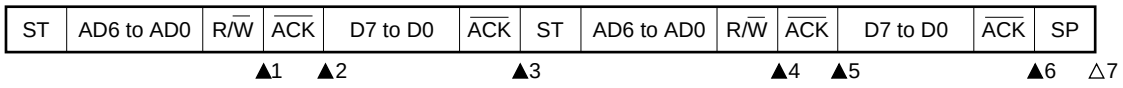
(i) When WTIM0 = 0 (after restart, extension code reception)



▲1: IICS0 = 0010×010B
▲2: IICS0 = 0010×000B
▲3: IICS0 = 0010×010B
▲4: IICS0 = 0010×000B
△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1 (after restart, extension code reception)



▲1: IICS0 = 0010×010B
▲2: IICS0 = 0010×110B
▲3: IICS0 = 0010××00B
▲4: IICS0 = 0010×010B
▲5: IICS0 = 0010×110B
▲6: IICS0 = 0010××00B
△7: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

Downloaded from Arrow.com.

(d) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

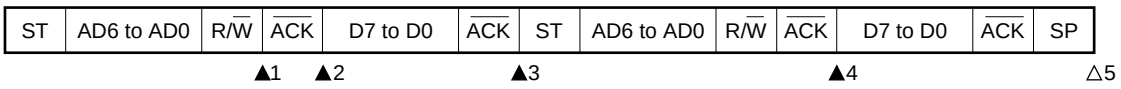
(i) When WTIM0 = 0 (after restart, does not match address (= not extension code))



▲1: IICS0 = 0010×010B
▲2: IICS0 = 0010×000B
▲3: IICS0 = 00000×10B
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1 (after restart, does not match address (= not extension code))



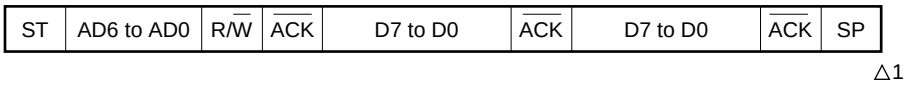
▲1: IICS0 = 0010×010B
▲2: IICS0 = 0010×110B
▲3: IICS0 = 0010××00B
▲4: IICS0 = 00000×10B
△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

Downloaded from Arrow.com.

(4) Operation without communication

(a) Start ~ Code ~ Data ~ Data ~ Stop



△1: IICS0 = 00000001B

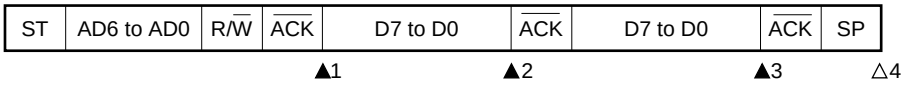
Remark △: Generated only when SPIE0 = 1

(5) Arbitration loss operation (operation as slave after arbitration loss)

When the device is used as a master in a multi-master system, read the MSTS0 bit each time interrupt request signal INTIICA0 has occurred to check the arbitration result.

(a) When arbitration loss occurs during transmission of slave address data

(i) When WTIM0 = 0

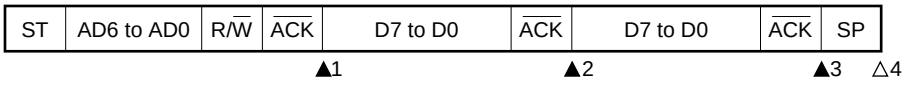


▲1: IICS0 = 0101×110B
▲2: IICS0 = 0001×000B
▲3: IICS0 = 0001×000B
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

Downloaded from Arrow.com.

(ii) When WTIM0 = 1

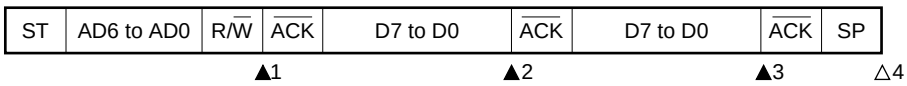


▲1: IICS0 = 0101×110B
▲2: IICS0 = 0001×100B
▲3: IICS0 = 0001××00B
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(b) When arbitration loss occurs during transmission of extension code

(i) When WTIM0 = 0

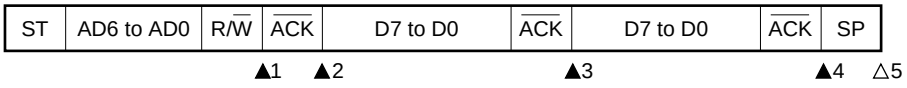


▲1: IICS0 = 0110×010B
▲2: IICS0 = 0010×000B
▲3: IICS0 = 0010×000B
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

Downloaded from Arrow.com.

(ii) When WTIM0 = 1



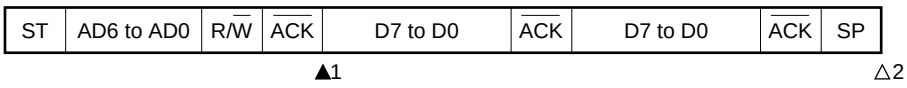
▲1: IICS0 = 0110×010B
▲2: IICS0 = 0010×110B
▲3: IICS0 = 0010×100B
▲4: IICS0 = 0010××00B
△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(6) Operation when arbitration loss occurs (no communication after arbitration loss)

When the device is used as a master in a multi-master system, read the MSTS0 bit each time interrupt request signal INTIICA0 has occurred to check the arbitration result.

(a) When arbitration loss occurs during transmission of slave address data (when WTIM0 = 1)

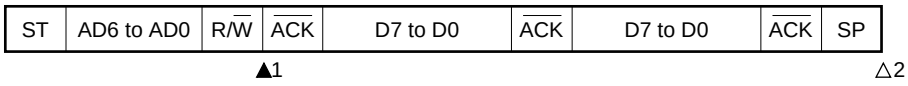


▲1: IICS0 = 01000110B
△2: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1

Downloaded from Arrow.com.

(b) When arbitration loss occurs during transmission of extension code

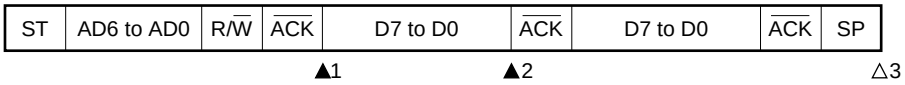


▲1: IICS0 = 0110×010B
Sets LREL0 = 1 by software
△2: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(c) When arbitration loss occurs during transmission of data

(i) When WTIM0 = 0

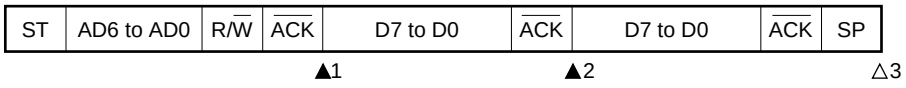


▲1: IICS0 = 10001110B
▲2: IICS0 = 01000000B
△3: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1

Downloaded from Arrow.com.

(ii) When WTIM0 = 1

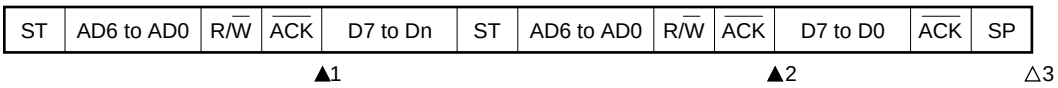


▲1: IICS0 = 10001110B
▲2: IICS0 = 01000100B
△3: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1

(d) When loss occurs due to restart condition during data transfer

(i) Not extension code (Example: unmatches with SVA0)

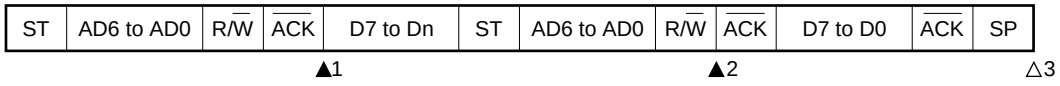


▲1: IICS0 = 1000×110B
▲2: IICS0 = 01000110B
△3: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care
n = 6 to 0

Downloaded from Arrow.com.

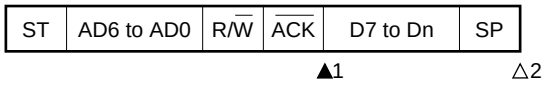
(ii) Extension code



▲1: IICS0 = 1000×110B
▲2: IICS0 = 01100010B
Sets LREL0 = 1 by software
△3: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care
n = 6 to 0

(e) When loss occurs due to stop condition during data transfer



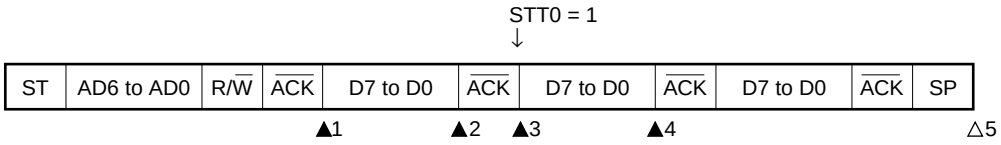
▲1: IICS0 = 10000110B
△2: IICS0 = 01000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care
n = 6 to 0

Downloaded from Arrow.com.

(f) When arbitration loss occurs due to low-level data when attempting to generate a restart condition

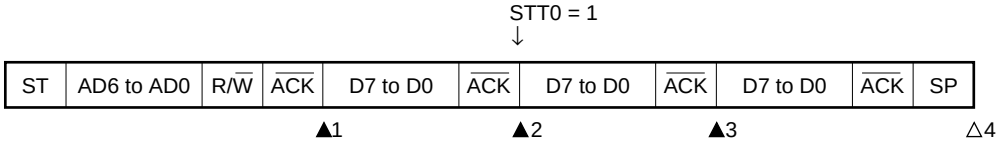
(i) When WTIM0 = 0



▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000×000B (Sets the WTIM0 bit to 1)
▲3: IICS0 = 1000×100B (Clears the WTIM0 bit to 0)
▲4: IICS0 = 01000000B
△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1



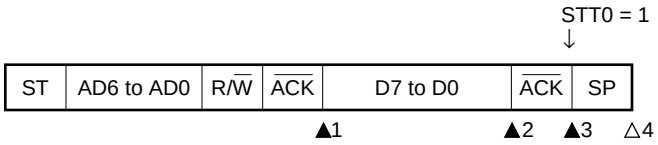
▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000×100B (Sets the STT0 bit to 1)
▲3: IICS0 = 01000100B
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

Downloaded from Arrow.com.

(g) When arbitration loss occurs due to a stop condition when attempting to generate a restart condition

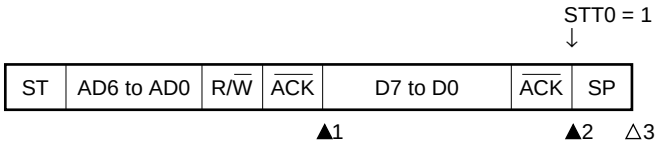
(i) When WTIM0 = 0



▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000×000B (Sets the WTIM0 bit to 1)
▲3: IICS0 = 1000××00B (Sets the STT0 bit to 1)
△4: IICS0 = 01000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1



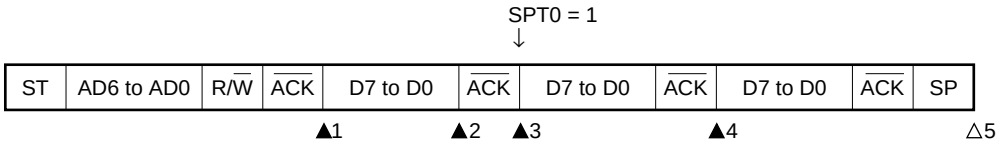
▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000××00B (Sets the STT0 bit to 1)
△3: IICS0 = 01000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

Downloaded from Arrow.com.

(h) When arbitration loss occurs due to low-level data when attempting to generate a stop condition

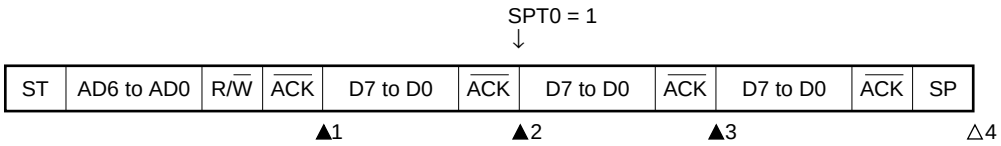
(i) When WTIM0 = 0



▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000×000B (Sets the WTIM0 bit to 1)
▲3: IICS0 = 1000×100B (Clears the WTIM0 bit to 0)
▲4: IICS0 = 01000100B
△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1



▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000×100B (Sets the SPT0 bit to 1)
▲3: IICS0 = 01000100B
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

Downloaded from Arrow.com.

12.6 Timing Charts

When using the I^2C bus mode, the master device outputs an address via the serial bus to select one of several slave devices as its communication partner.

After outputting the slave address, the master device transmits the TRC0 bit (bit 3 of the IICA status register 0 (IICS0)), which specifies the data transfer direction, and then starts serial communication with the slave device.

Figures 12-32 and 12-33 show timing charts of the data communication.

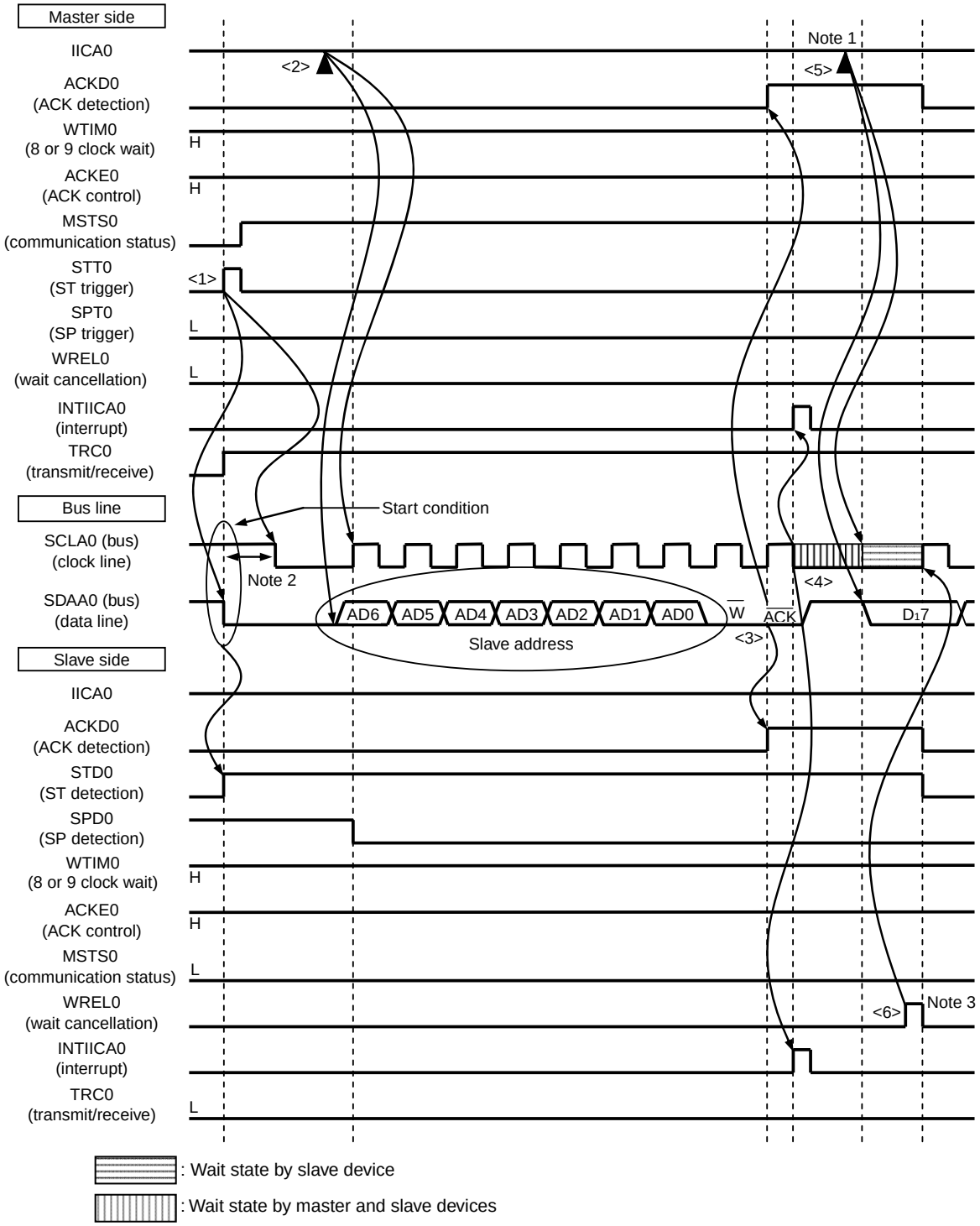
The IICA shift register 0 (IICA0)'s shift operation is synchronized with the falling edge of the serial clock (SCLA0). The transmit data is transferred to the SO latch and is output (MSB first) via the SDAA0 pin.

Data input via the SDAA0 pin is captured into IICA0 at the rising edge of SCLA0.

Downloaded from Arrow.com.

Figure 12-32. Example of Master to Slave Communication (9-Clock Wait Is Selected for Master, 9-Clock Wait Is Selected for Slave) (1/4)

(1) Start condition ~ address ~ data



Notes 1. Write data to IICA0, not setting the WREL0 bit, in order to cancel a wait state during transmission by a master device.

2. Make sure that the time between the fall of the SDAA0 pin signal and the fall of the SCLA0 pin signal is at least 4.0 μs when specifying standard mode and at least 0.6 μs when specifying fast mode.

3. For releasing wait state during reception of a slave device, write “FFH” to IICA0 or set the WREL0 bit.

Downloaded from Arrow.com.

The meanings of <1> to <6> in (1) Start condition ~ address ~ data in Figure 12-32 are explained below.

<1> The start condition trigger is set by the master device (STT0 = 1) and a start condition (SDAA0 = 0 and SCLA0 = 1) is generated once the bus data line goes low (SDAA0 = 0). When the start condition is subsequently detected, the master device enters the master device communication status (MSTS0 = 1). The master device is ready to communicate once the bus clock line goes low (SCLA0 = 0) after the hold time has elapsed.

<2> The master device writes the address + W (transmission) to the IICA shift register 0 (IICA0) and transmits the slave address.

<3> In the slave device if the address received matches the address (SVA0 value) of a slave device[Note], that slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKD0 = 1) at the rising edge of the 9th clock.

<4> The master device issues an interrupt (INTIICA0: end of address transmission) at the falling edge of the 9th clock. The slave device whose address matched the transmitted slave address sets a wait status (SCLA0 = 0) and issues an interrupt (INTIICA0: address match)[Note].

<5> The master device writes the data to transmit to the IICA0 register and releases the wait status that it set by the master device.

<6> If the slave device releases the wait status (WREL0 = 1), the master device starts transferring data to the slave device.

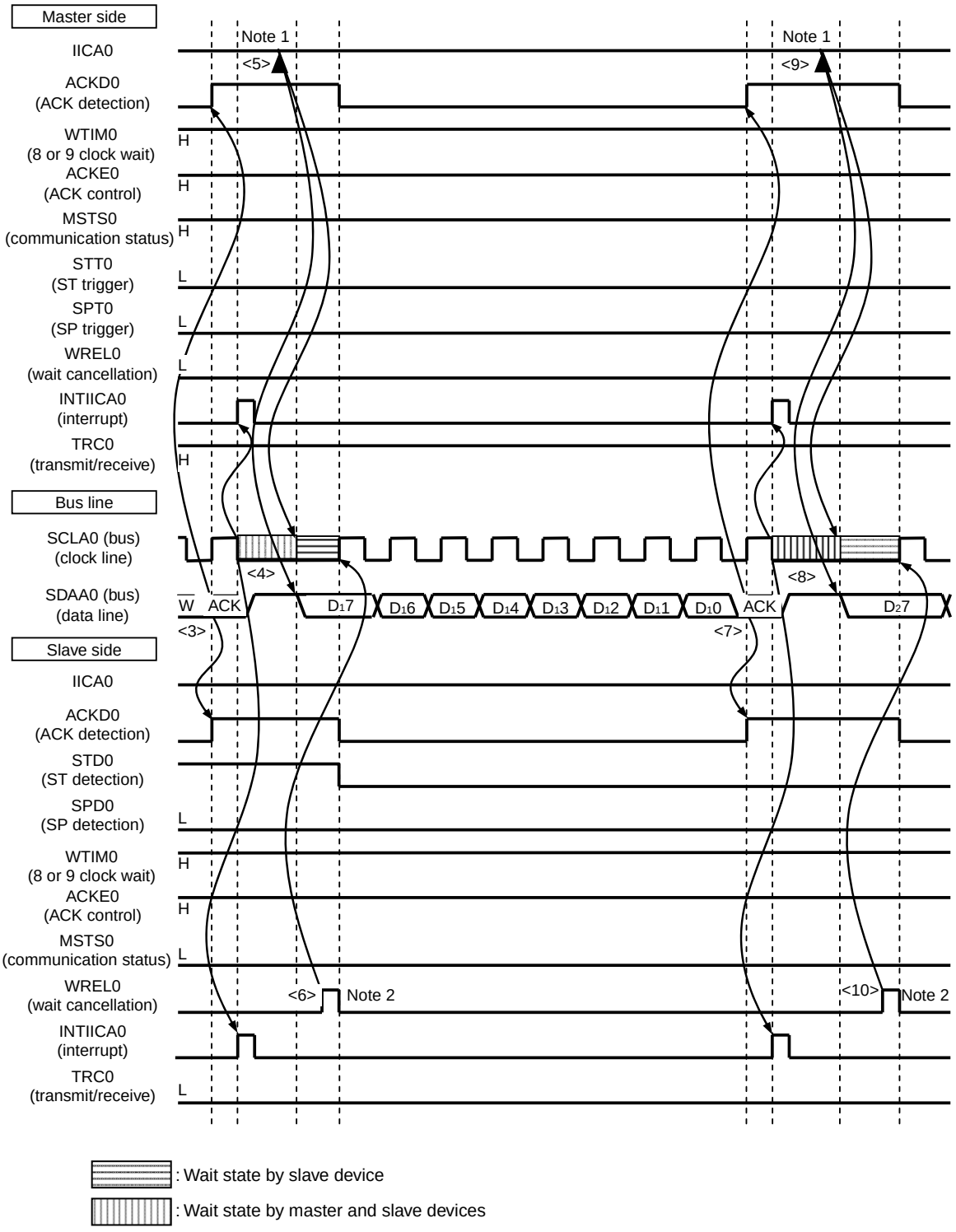
Note If the transmitted address does not match the address of the slave device, the slave device does not return an ACK to the master device (NACK: SDAA0 = 1). The slave device also does not issue the INTIICA0 interrupt (address match) and does not set a wait status. The master device, however, issues the INTIICA0 interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

Remark <1> to <15> in Figure 12-32 show the entire procedure for communicating data using the I^2C bus.
Figure 12-32 (1) Start condition ~ address ~ data shows the processing from <1> to <6>, Figure 12-32 (2) Address ~ data ~ data shows the processing from <3> to <10>, and Figure 12-32 (3) Data ~ data ~ stop condition shows the processing from <7> to <15>.

Downloaded from Arrow.com.

Figure 12-32. Example of Master to Slave Communication
(9-Clock Wait Is Selected for Master, 9-Clock Wait Is Selected for Slave) (2/4)

(2) Address ~ data ~ data



Notes 1. Write data to IICA0, not setting the WREL0 bit, in order to cancel a wait state during transmission by a master device.

2. For releasing wait state during reception of a slave device, write “FFH” to IICA0 or set the WREL0 bit.

Downloaded from Arrow.com.

The meanings of <3> to <10> in (2) Address ~ data ~ data in Figure 12-32 are explained below.

<3> In the slave device if the address received matches the address (SVA0 value) of a slave device[Note], that slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKD0 = 1) at the rising edge of the 9th clock.

<4> The master device issues an interrupt (INTIICA0: end of address transmission) at the falling edge of the 9th clock. The slave device whose address matched the transmitted slave address sets a wait status (SCLA0 = 0) and issues an interrupt (INTIICA0: address match)[Note].

<5> The master device writes the data to transmit to the IICA shift register 0 (IICA0) and releases the wait status that it set by the master device.

<6> If the slave device releases the wait status (WREL0 = 1), the master device starts transferring data to the slave device.

<7> After data transfer is completed, because of ACKE0 = 1, the slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKD0 = 1) at the rising edge of the 9th clock.

<8> The master device and slave device set a wait status (SCLA0 = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICA0: end of transfer).

<9> The master device writes the data to transmit to the IICA0 register and releases the wait status that it set by the master device.

<10> The slave device reads the received data and releases the wait status (WREL0 = 1). The master device then starts transferring data to the slave device.

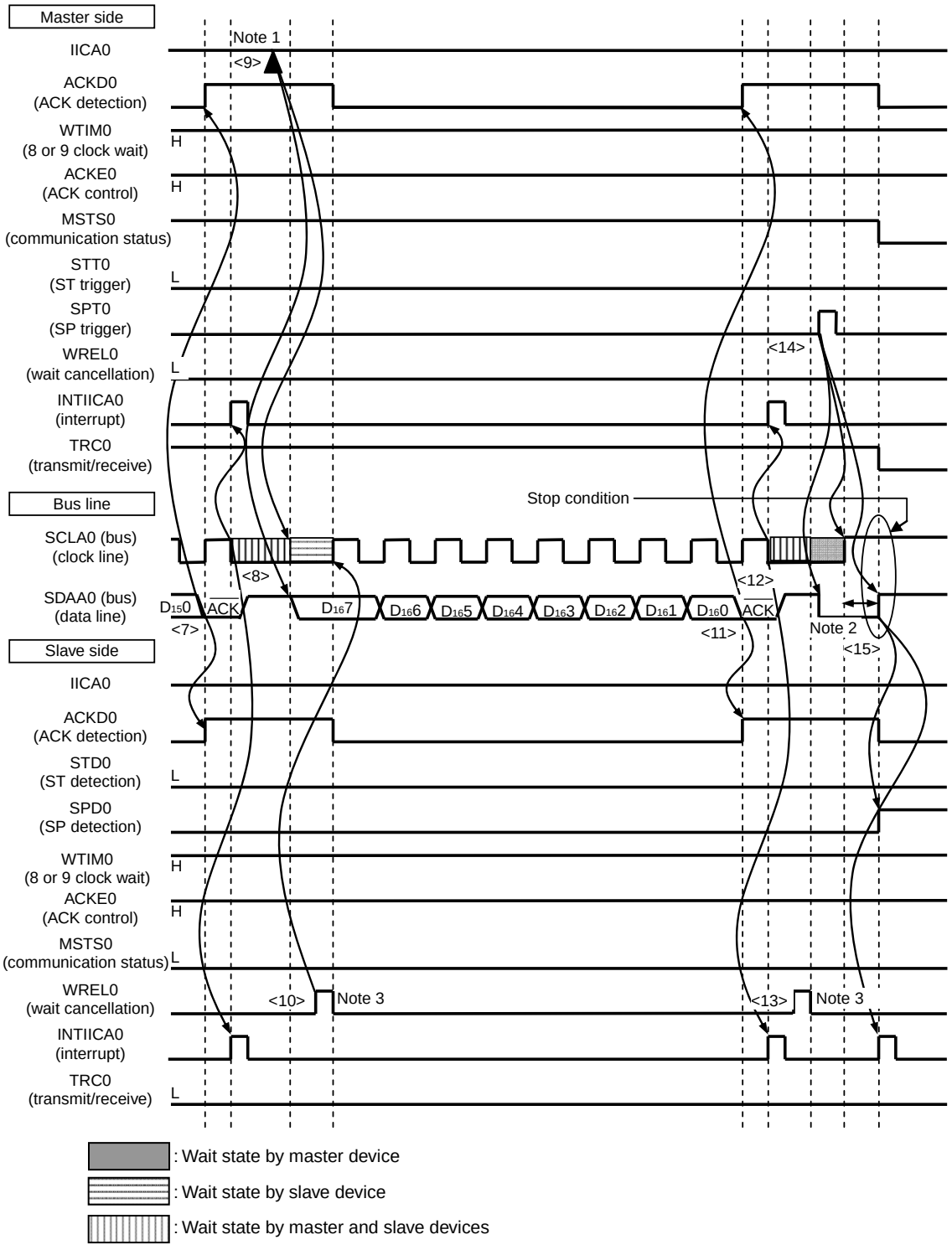
Note If the transmitted address does not match the address of the slave device, the slave device does not return an ACK to the master device (NACK: SDAA0 = 1). The slave device also does not issue the INTIICA0 interrupt (address match) and does not set a wait status. The master device, however, issues the INTIICA0 interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

Remark <1> to <15> in Figure 12-32 show the entire procedure for communicating data using the I^2C bus.
Figure 12-32 (1) Start condition ~ address ~ data shows the processing from <1> to <6>, Figure 12-32 (2) Address ~ data ~ data shows the processing from <3> to <10>, and Figure 12-32 (3) Data ~ data ~ stop condition shows the processing from <7> to <15>.

Downloaded from Arrow.com.

Figure 12-32. Example of Master to Slave Communication (9-Clock Wait Is Selected for Master, 9-Clock Wait Is Selected for Slave) (3/4)

(3) Data ~ data ~ Stop condition



Notes 1. Write data to IICA0, not setting the WREL0 bit, in order to cancel a wait state during transmission by a master device.

2. Make sure that the time between the rise of the SCLA0 pin signal and the generation of the stop condition after a stop condition has been issued is at least 4.0 μs when specifying standard mode and at least 0.6 μs when specifying fast mode.

3. For releasing wait state during reception of a slave device, write “FFH” to IICA0 or set the WREL0 bit.

Downloaded from Arrow.com.

The meanings of <7> to <15> in (3) Data ~ data ~ stop condition in Figure 12-32 are explained below.

<7> After data transfer is completed, because of ACKE0 = 1, the slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKD0 = 1) at the rising edge of the 9th clock.

<8> The master device and slave device set a wait status (SCLA0 = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICA0: end of transfer).

<9> The master device writes the data to transmit to the IICA shift register 0 (IICA0) and releases the wait status that it set by the master device.

<10> The slave device reads the received data and releases the wait status (WREL0 = 1). The master device then starts transferring data to the slave device.

<11> When data transfer is complete, the slave device (ACKE0 =1) sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKD0 = 1) at the rising edge of the 9th clock.

<12> The master device and slave device set a wait status (SCLA0 = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICA0: end of transfer).

<13> The slave device reads the received data and releases the wait status (WREL0 = 1).

<14> By the master device setting a stop condition trigger (SPT0 = 1), the bus data line is cleared (SDAA0 = 0) and the bus clock line is set (SCLA0 = 1). After the stop condition setup time has elapsed, by setting the bus data line (SDAA0 = 1), the stop condition is then generated (i.e. SCLA0 =1 changes SDAA0 from 0 to 1).

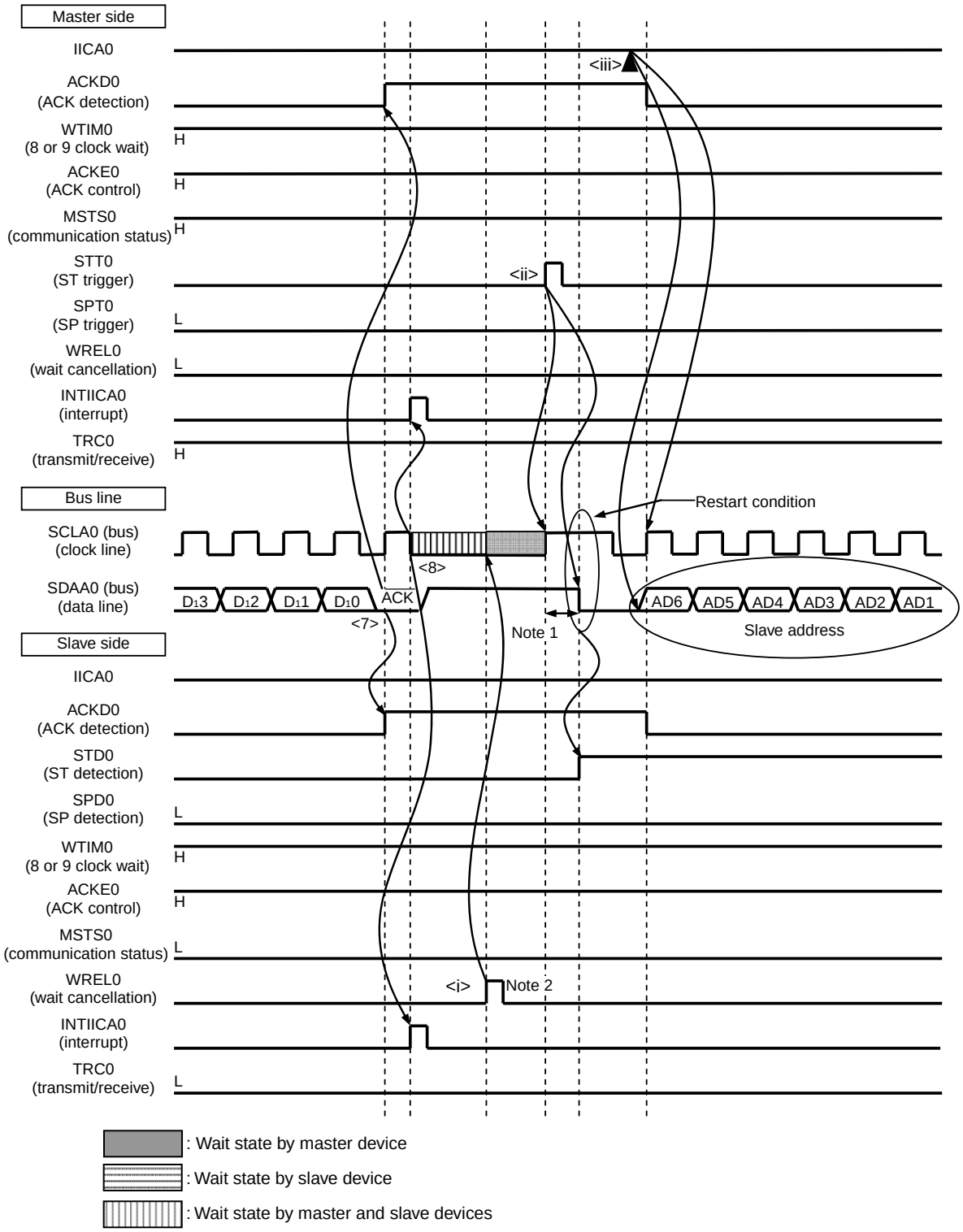
<15> When a stop condition is generated, the slave device detects the stop condition and issues an interrupt (INTIICA0: stop condition).

Remark <1> to <15> in Figure 12-32 show the entire procedure for communicating data using the I^2C bus.
Figure 12-32 (1) Start condition ~ address ~ data shows the processing from <1> to <6>, Figure 12-32 (2) Address ~ data ~ data shows the processing from <3> to <10>, and Figure 12-32 (3) Data ~ data ~ stop condition shows the processing from <7> to <15>.

Downloaded from Arrow.com.

Figure 12-32. Example of Master to Slave Communication (9-Clock Wait Is Selected for Master, 9-Clock Wait Is Selected for Slave) (4/4)

(4) Data ~ restart condition ~ address



Notes 1. Make sure that the time between the rise of the SCLA0 pin signal and the generation of the start condition after a restart condition has been issued is at least 4.7 μs when specifying standard mode and at least 0.6 μs when specifying fast mode.

2. For releasing wait state during reception of a slave device, write “FFH” to IICA0 or set the WREL0 bit.

Downloaded from Arrow.com.

The following describes the operations in Figure 12-32 (4) Data ~ restart condition ~ address. After the operations in steps <7> and <8>, the operations in steps <1> to <3> are performed. These steps return the processing to step <3>, the data transmission step.

<7> After data transfer is completed, because of ACKE0 = 1, the slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKD0 = 1) at the rising edge of the 9th clock.

<8> The master device and slave device set a wait status (SCLA0 = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICA0: end of transfer).

<i> The slave device reads the received data and releases the wait status (WREL0 = 1).

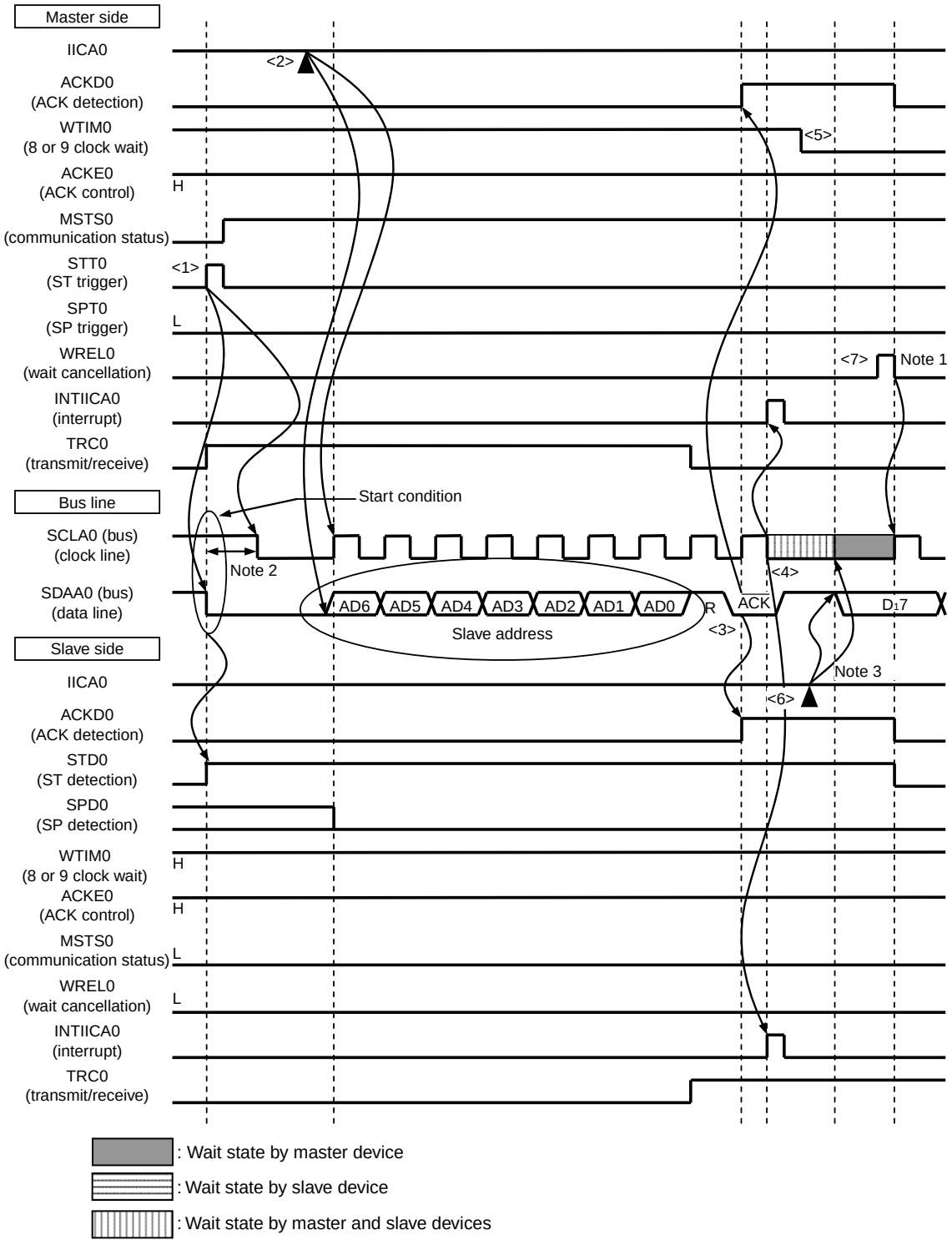
<ii> The start condition trigger is set again by the master device (STT0 = 1) and a start condition (i.e. SCLA0 =1 changes SDAA0 from 1 to 0) is generated once the bus clock line goes high (SCLA0 = 1) and the bus data line goes low (SDAA0 = 0) after the restart condition setup time has elapsed. When the start condition is subsequently detected, the master device is ready to communicate once the bus clock line goes low (SCLA0 = 0) after the hold time has elapsed.

<iii> The master device writing the address + R/W (transmission) to the IICA shift register (IICA0) enables the slave address to be transmitted.

Downloaded from Arrow.com.

Figure 12-33. Example of Slave to Master Communication (8-Clock Wait Is Selected for Master, 9-Clock Wait Is Selected for Slave) (1/3)

(1) Start condition ~ address ~ data



Notes 1. For releasing wait state during reception of a master device, write “FFH” to IICA0 or set the WREL0 bit.

2. Make sure that the time between the fall of the SDAA0 pin signal and the fall of the SCLA0 pin signal is at least 4.0 μs when specifying standard mode and at least 0.6 μs when specifying fast mode.

3. Write data to IICA0, not setting the WREL0 bit, in order to cancel a wait state during transmission by a slave device.

Downloaded from Arrow.com.

The meanings of <1> to <7> in (1) Start condition ~ address ~ data in Figure 12-33 are explained below.

<1> The start condition trigger is set by the master device (STT0 = 1) and a start condition (i.e. SCLA0 =1 changes SDAA0 from 1 to 0) is generated once the bus data line goes low (SDAA0). When the start condition is subsequently detected, the master device enters the master device communication status (MSTS0 = 1). The master device is ready to communicate once the bus clock line goes low (SCLA0 = 0) after the hold time has elapsed.

<2> The master device writes the address + R (reception) to the IICA shift register 0 (IICA0) and transmits the slave address.

<3> In the slave device if the address received matches the address (SVA0 value) of a slave device Note, that slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKD0 = 1) at the rising edge of the 9th clock.

<4> The master device issues an interrupt (INTIICA0: end of address transmission) at the falling edge of the 9th clock. The slave device whose address matched the transmitted slave address sets a wait status (SCLA0 = 0) and issues an interrupt (INTIICA0: address match) Note.

<5> The timing at which the master device sets the wait status changes to the 8th clock (WTIM0 = 0).

<6> The slave device writes the data to transmit to the IICA0 register and releases the wait status that it set by the slave device.

<7> The master device releases the wait status (WREL0 = 1) and starts transferring data from the slave device to the master device.

Note If the transmitted address does not match the address of the slave device, the slave device does not return an ACK to the master device (NACK: SDAA0 = 1). The slave device also does not issue the INTIICA0 interrupt (address match) and does not set a wait status. The master device, however, issues the INTIICA0 interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

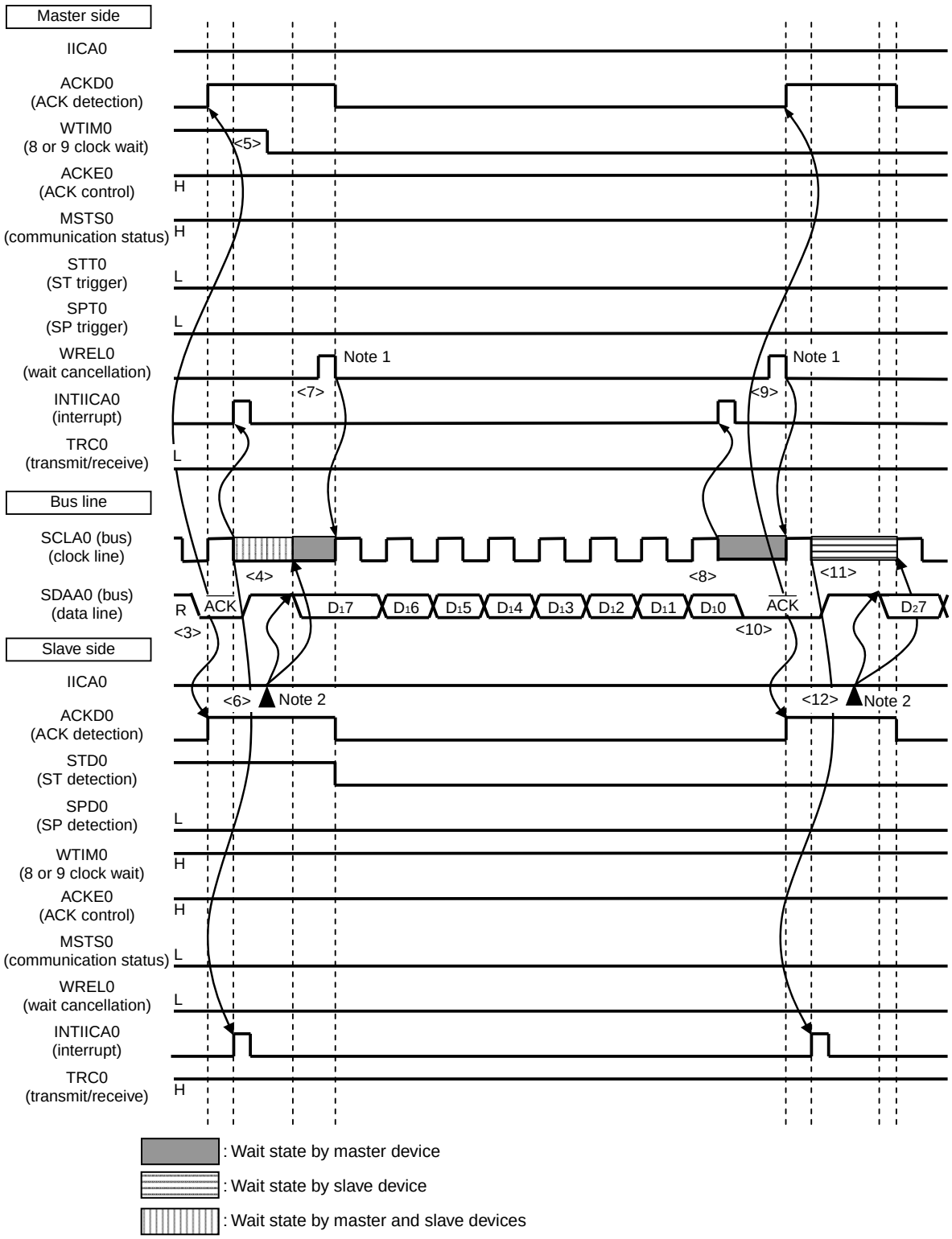
Remark <1> to <19> in Figure 12-33 show descriptions the entire procedure for communicating data using the I^2C bus.

Figure 12-33 (1) Start condition ~ address ~ data shows the processing from <1> to <7>, Figure 12-33 (2) Address ~ data ~ data shows the processing from <3> to <12>, and Figure 12-33 (3) Data ~ data ~ stop condition shows the processing from <8> to <19>.

Downloaded from Arrow.com.

Figure 12-33. Example of Slave to Master Communication (8-Clock Wait Is Selected for Master, 9-Clock Wait Is Selected for Slave) (2/3)

(2) Address ~ data ~ data



Notes 1. For releasing wait state during reception of a master device, write “FFH” to IICA0 or set the WREL0 bit.

2. Write data to IICA0, not setting the WREL0 bit, in order to cancel a wait state during transmission by a slave device.

Downloaded from Arrow.com.

The meanings of <3> to <12> in (2) Address ~ data ~ data in Figure 12-33 are explained below.

<3> In the slave device if the address received matches the address (SVA0 value) of a slave device[Note], that slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKD0 = 1) at the rising edge of the 9th clock.

<4> The master device issues an interrupt (INTIICA0: end of address transmission) at the falling edge of the 9th clock. The slave device whose address matched the transmitted slave address sets a wait status (SCLA0 = 0) and issues an interrupt (INTIICA0: address match)[Note].

<5> The master device changes the timing of the wait status to the 8th clock (WTIM0 = 0).

<6> The slave device writes the data to transmit to the IICA shift register 0 (IICA0) and releases the wait status that it set by the slave device.

<7> The master device releases the wait status (WREL0 = 1) and starts transferring data from the slave device to the master device.

<8> The master device sets a wait status (SCLA0 = 0) at the falling edge of the 8th clock, and issues an interrupt (INTIICA0: end of transfer). Because of ACKE0 = 1 in the master device, the master device then sends an ACK by hardware to the slave device.

<9> The master device reads the received data and releases the wait status (WREL0 = 1).

<10> The ACK is detected by the slave device (ACKD0 = 1) at the rising edge of the 9th clock.

<11> The slave device set a wait status (SCLA0 = 0) at the falling edge of the 9th clock, and the slave device issue an interrupt (INTIICA0: end of transfer).

<12> By the slave device writing the data to transmit to the IICA0 register, the wait status set by the slave device is released. The slave device then starts transferring data to the master device.

Note If the transmitted address does not match the address of the slave device, the slave device does not return an ACK to the master device (NACK: SDAA0 = 1). The slave device also does not issue the INTIICA0 interrupt (address match) and does not set a wait status. The master device, however, issues the INTIICA0 interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

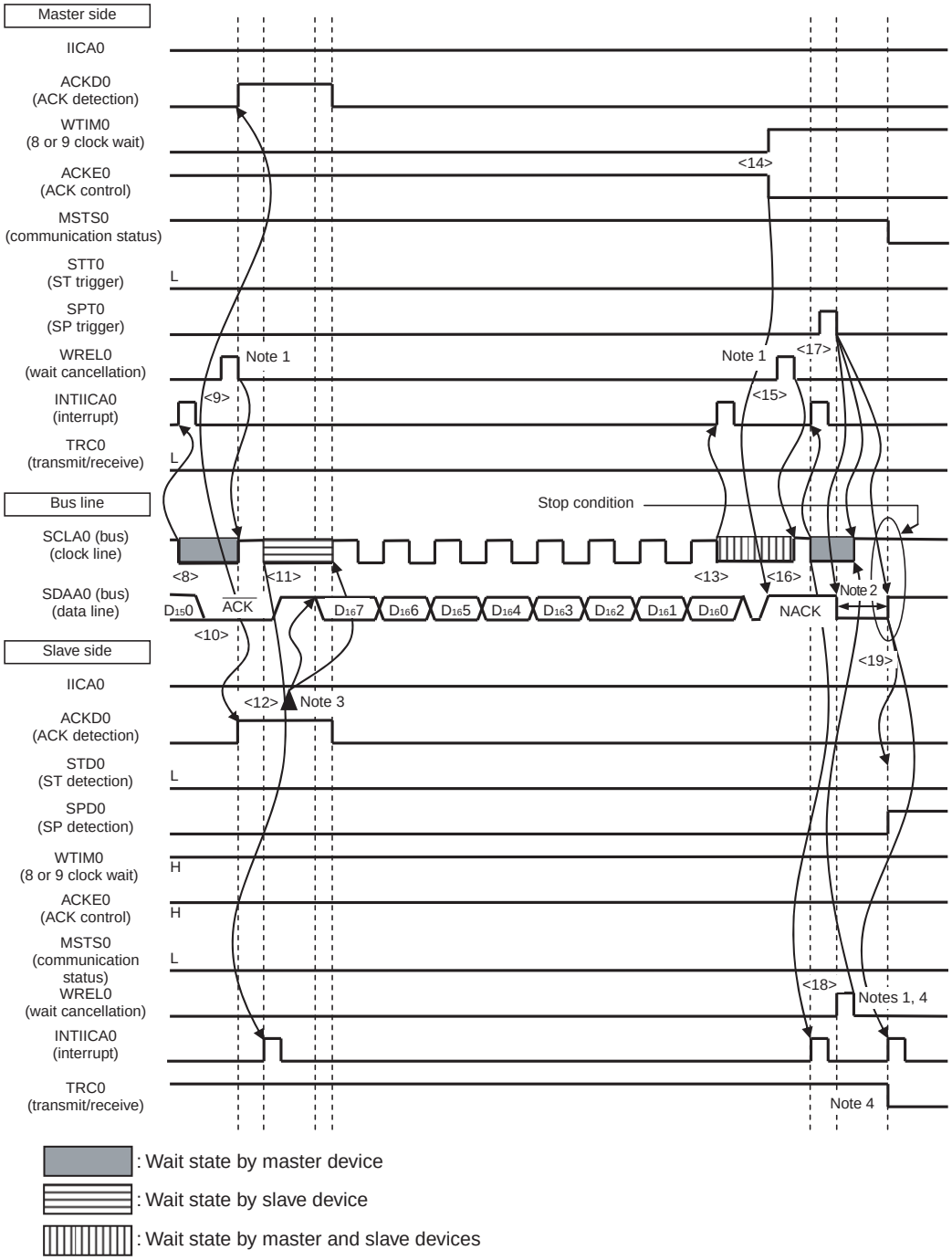
Remark <1> to <19> in Figure 12-33 show descriptions the entire procedure for communicating data using the I^2C bus.

Figure 12-33 (1) Start condition ~ address ~ data shows the processing from <1> to <7>, Figure 12-33 (2) Address ~ data ~ data shows the processing from <3> to <12>, and Figure 12-33 (3) Data ~ data ~ stop condition shows the processing from <8> to <19>.

Downloaded from Arrow.com.

Figure 12-33. Example of Slave to Master Communication (8-Clock and 9-Clock Wait Is Selected for Master, 9-Clock Wait Is Selected for Slave) (3/3)

(3) Data ~ data ~ stop condition



Notes 1. To cancel a wait state, write “FFH” to IICA0 or set the WREL0 bit.

2. Make sure that the time between the rise of the SCLA0 pin signal and the generation of the stop condition after a stop condition has been issued is at least 4.0 μs when specifying standard mode and at least 0.6 μs when specifying fast mode.

3. Write data to IICA0, not setting the WREL0 bit, in order to cancel a wait state during transmission by a slave device.

4. If a wait state during transmission by a slave device is canceled by setting the WREL0 bit, the TRC0 bit will be cleared.

Downloaded from Arrow.com.

The meanings of <8> to <19> in (3) Data ~ data ~ stop condition in Figure 12-33 are explained below.

<8> The master device sets a wait status (SCLA0 = 0) at the falling edge of the 8th clock, and issues an interrupt (INTIICA0: end of transfer). Because of ACKE0 = 0 in the master device, the master device then sends an ACK by hardware to the slave device.

<9> The master device reads the received data and releases the wait status (WREL0 = 1).

<10> The ACK is detected by the slave device (ACKD0 = 1) at the rising edge of the 9th clock.

<11> The slave device set a wait status (SCLA0 = 0) at the falling edge of the 9th clock, and the slave device issue an interrupt (INTIICA0: end of transfer).

<12> By the slave device writing the data to transmit to the IICA register, the wait status set by the slave device is released. The slave device then starts transferring data to the master device.

<13> The master device issues an interrupt (INTIICA0: end of transfer) at the falling edge of the 8th clock, and sets a wait status (SCLA0 = 0). Because ACK control (ACKE0 = 1) is performed, the bus data line is at the low level (SDAA0 = 0) at this stage.

<14> The master device sets NACK as the response (ACKE0 = 0) and changes the timing at which it sets the wait status to the 9th clock (WTIM0 = 1).

<15> If the master device releases the wait status (WREL0 = 1), the slave device detects the NACK (ACK = 0) at the rising edge of the 9th clock.

<16> The master device and slave device set a wait status (SCLA0 = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICA0: end of transfer).

<17> When the master device issues a stop condition (SPT0 = 1), the bus data line is cleared (SDAA0 = 0) and the master device releases the wait status. The master device then waits until the bus clock line is set (SCLA0 = 1).

<18> The slave device acknowledges the NACK, halts transmission, and releases the wait status (WREL0 = 1) to end communication. Once the slave device releases the wait status, the bus clock line is set (SCLA0 = 1).

<19> Once the master device recognizes that the bus clock line is set (SCLA0 = 1) and after the stop condition setup time has elapsed, the master device sets the bus data line (SDAA0 = 1) and issues a stop condition (i.e. SCLA0 =1 changes SDAA0 from 0 to 1). When a stop condition is generated, the slave device detects the stop condition and issues an interrupt (INTIICA0: stop condition).

Remark <1> to <19> in Figure 12-33 show descriptions the entire procedure for communicating data using the I^2C bus.
Figure 12-33 (1) Start condition ~ address ~ data shows the processing from <1> to <7>, Figure 12-33 (2) Address ~ data ~ data shows the processing from <3> to <12>, and Figure 12-33 (3) Data ~ data ~ stop condition shows the processing from <8> to <19>.

Downloaded from Arrow.com.

CHAPTER 13 MULTIPLIER AND DIVIDER/MULTIPLY-ACCUMULATOR

13.1 Functions of Multiplier and Divider/Multiply-Accumulator

The multiplier and divider/multiply-accumulator has the following functions.

- 16 bits × 16 bits = 32 bits (Unsigned)
- 16 bits × 16 bits = 32 bits (Signed)
- 16 bits × 16 bits + 32 bits = 32 bits (Unsigned)
- 16 bits × 16 bits + 32 bits = 32 bits (Signed)
- 32 bits ÷ 32 bits = 32 bits, 32-bits remainder (Unsigned)

13.2 Configuration of Multiplier and Divider/Multiply-Accumulator

The multiplier and divider/multiply-accumulator consists of the following hardware.

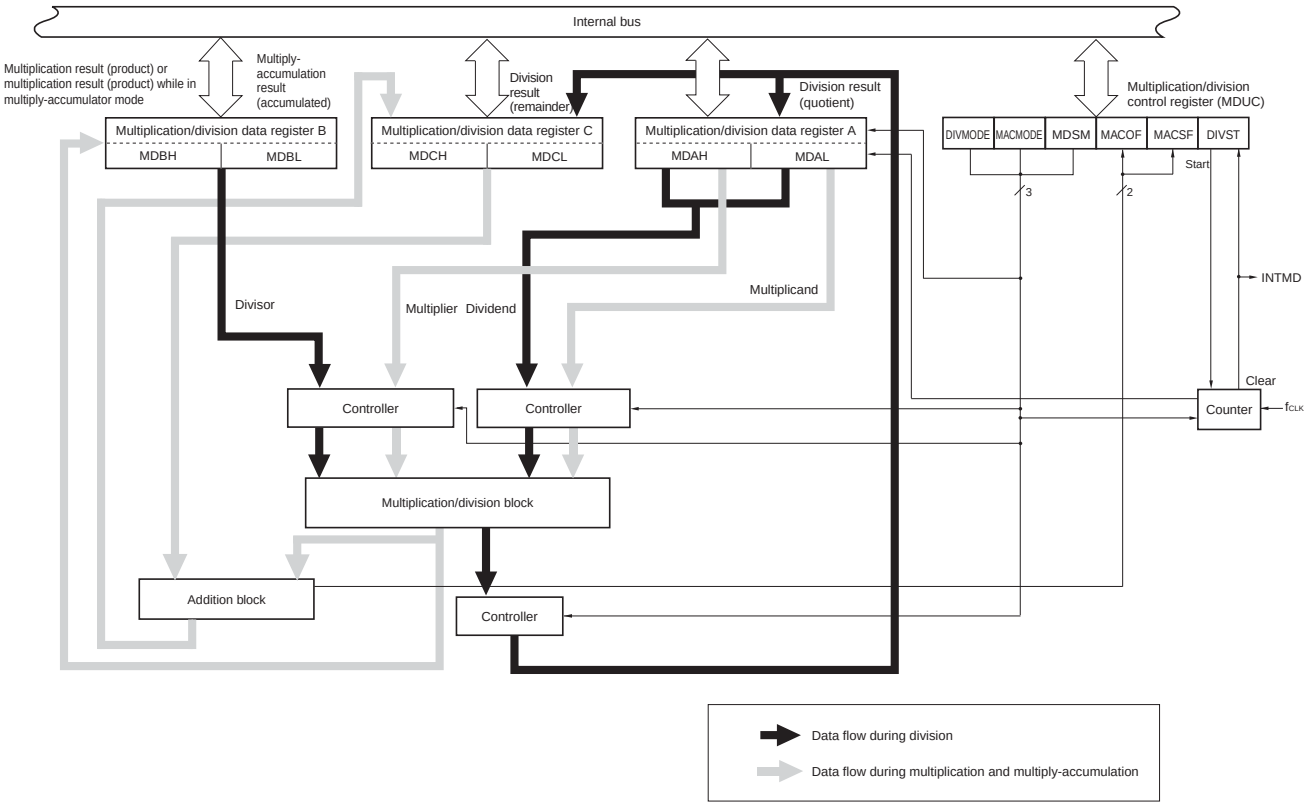
Table 13-1. Configuration of Multiplier and Divider/Multiply-Accumulator

Item	Configuration
Registers	Multiplication/division data register A (L) (MDAL) Multiplication/division data register A (H) (MDAH) Multiplication/division data register B (L) (MDBL) Multiplication/division data register B (H) (MDBH) Multiplication/division data register C (L) (MDCL) Multiplication/division data register C (H) (MDCH)
Control register	Multiplication/division control register (MDUC)

Figure 13-1 shows a block diagram of the multiplier and divider/multiply-accumulator.

Downloaded from Arrow.com.

Figure 13-1. Block Diagram of Multiplier and Divider/Multiply-Accumulator



<R> **Remark** f_{CLK}: CPU/peripheral hardware clock frequency

Downloaded from Arrow.com.

<R> 13.2.1 Multiplication/division data register A (MDAH, MDAL)

The MDAH and MDAL registers set the values that are used for a multiplication or division operation and store the operation result. They set the multiplier and multiplicand data in the multiplication mode or multiply-accumulator mode, and set the dividend data in the division mode. Furthermore, the operation result (quotient) is stored in the MDAH and MDAL registers in the division mode.

The MDAH and MDAL registers can be set by a 16-bit manipulation instruction.

Reset signal generation clears these registers to 0000H.

Figure 13-2. Format of Multiplication/Division Data Register A (MDAH, MDAL)

Address: FFFF0H, FFFF1H, FFFF2H, FFFF3H After reset: 0000H, 0000H R/W

Symbol: FFFF3H | FFFF2H

MDAH	MDAH	MDAH	MDAH	MDAH	MDAH	MDAH	MDAH	MDAH	MDAH	MDAH	MDAH	MDAH	MDAH	MDAH	MDAH
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Symbol: FFFF1H | FFFF0H

MDAL	MDAL	MDAL	MDAL	MDAL	MDAL	MDAL	MDAL	MDAL	MDAL	MDAL	MDAL	MDAL	MDAL	MDAL	MDAL
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Cautions 1. Do not rewrite the MDAH and MDAL registers values during division operation processing (when the multiplication/division control register (MDUC) value is 81H or C1H). The operation will be executed in this case, but the operation result will be an undefined value.

2. The MDAH and MDAL registers values read during division operation processing (when the MDUC register value is 81H or C1H) will not be guaranteed.

3. The data is in the two's complement format in either the multiplication mode (signed) or multiply-accumulator mode (signed).

The following table shows the functions of the MDAH and MDAL registers during operation execution.

<R> **Table 13-2. Functions of MDAH and MDAL Registers During Operation Execution**

Operation Mode	Setting	Operation Result
Multiplication mode (unsigned) Multiply-accumulator mode (unsigned)	MDAH: Multiplier (unsigned) MDAL: Multiplicand (unsigned)	–
Multiplication mode (signed) Multiply-accumulator mode (signed)	MDAH: Multiplier (signed) MDAL: Multiplicand (signed)	–
Division mode (unsigned)	MDAH: Dividend (unsigned) (higher 16 bits) MDAL: Dividend (unsigned) (lower 16 bits)	MDAH: Division result (quotient) (unsigned) Higher 16 bits MDAL: Division result (quotient) (unsigned) Lower 16 bits

Downloaded from Arrow.com.

<R> 13.2.2 Multiplication/division data register B (MDBL, MDBH)

The MDBH and MDBL registers set the values that are used for multiplication or division operation and store the operation result. They store the operation result (product) in the multiplication mode and multiply-accumulator mode, and set the divisor data in the division mode.

The MDBH and MDBL registers can be set by a 16-bit manipulation instruction.

Reset signal generation clears these registers to 0000H.

Figure 13-3. Format of Multiplication/Division Data Register B (MDBH, MDBL)

Address: FFFF4H, FFFF5H, FFFF6H, FFFF7H After reset: 0000H, 0000H R/W

Symbol	FFFF7H								FFFF6H							
MDBH	MDBH 15	MDBH 14	MDBH 13	MDBH 12	MDBH 11	MDBH 10	MDBH 9	MDBH 8	MDBH 7	MDBH 6	MDBH 5	MDBH 4	MDBH 3	MDBH 2	MDBH 1	MDBH 0

Symbol	FFFF5H								FFFF4H							
MDBL	MDBL 15	MDBL 14	MDBL 13	MDBL 12	MDBL 11	MDBL 10	MDBL 9	MDBL 8	MDBL 7	MDBL 6	MDBL 5	MDBL 4	MDBHL 3	MDBL 2	MDBL 1	MDBL 0

Cautions 1. Do not rewrite the MDBH and MDBL registers values during division operation processing (when the multiplication/division control register (MDUC) value is 81H or C1H) or multiply-accumulation processing. The operation result will be an undefined value.

2. Do not set the MDBH and MDBL registers to 0000H in the division mode. If they are set, the operation result will be an undefined value.

3. The data is in the two's complement format in either the multiplication mode (signed) or multiply-accumulator mode (signed).

The following table shows the functions of the MDBH and MDBL registers during operation execution.

<R> **Table 13-3. Functions of MDBH and MDBL Registers During Operation Execution**

Operation Mode	Setting	Operation Result
Multiplication mode (unsigned) Multiply-accumulator mode (unsigned)	–	MDBH: Multiplication result (product) (unsigned) Higher 16 bits MDBL: Multiplication result (product) (unsigned) Lower 16 bits
Multiplication mode (signed) Multiply-accumulator mode (signed)	–	MDBH: Multiplication result (product) (signed) Higher 16 bits MDBL: Multiplication result (product) (signed) Lower 16 bits
Division mode (unsigned)	MDBH: Divisor (unsigned) (higher 16 bits) MDBL: Divisor (unsigned) (lower 16 bits)	–

Downloaded from Arrow.com.

<R> 13.2.3 Multiplication/division data register C (MDCL, MDCH)

The MDCH and MDCL registers are used to store the accumulated result while in the multiply-accumulator mode or the remainder of the operation result while in the division mode. These registers are not used while in the multiplication mode.

The MDCH and MDCL registers can be set by a 16-bit manipulation instruction.

Reset signal generation clears these registers to 0000H.

Figure 13-4. Format of Multiplication/Division Data Register C (MDCH, MDCL)

Address: F00E0H, F00E1H, F00E2H, F00E3H After reset: 0000H, 0000H R/W

Symbol	F00E3H								F00E2H							
MDCH	MDCH 15	MDCH 14	MDCH 13	MDCH 12	MDCH 11	MDCH 10	MDCH 9	MDCH 8	MDCH 7	MDCH 6	MDCH 5	MDCH 4	MDCH 3	MDCH 2	MDCH 1	MDCH 0

Symbol	F00E1H								F00E0H							
MDCL	MDCL 15	MDCL 14	MDCL 13	MDCL 12	MDCL 11	MDCL 10	MDCL 9	MDCL 8	MDCL 7	MDCL 6	MDCL 5	MDCL 4	MDCL 3	MDCL 2	MDCL 1	MDCL 0

Cautions 1. The MDCH and MDCL registers values read during division operation processing (when the multiplication/division control register (MDUC) value is 81H or C1H) will not be guaranteed.

2. During multiply-accumulator processing, do not use software to rewrite the values of the MDCH and MDCL registers. If this is done, the operation result will be undefined.

3. The data is in the two's complement format in the multiply-accumulator mode (signed).

<R> **Table 13-4. Functions of MDCH and MDCL Registers During Operation Execution**

Operation Mode	Setting	Operation Result
Multiplication mode (unsigned or signed)	–	–
Multiply-accumulator mode (unsigned)	MDCH: Initial accumulated value (unsigned) (higher 16 bits) MDCL: Initial accumulated value (unsigned) (lower 16 bits)	MDCH: accumulated value (unsigned) (higher 16 bits) MDCL: accumulated value (unsigned) (lower 16 bits)
Multiply-accumulator mode (signed)	MDCH: Initial accumulated value (signed) (higher 16 bits) MDCL: Initial accumulated value (signed) (lower 16 bits)	MDCH: accumulated value (signed) (higher 16 bits) MDCL: accumulated value (signed) (lower 16 bits)
Division mode (unsigned)	–	MDCH: Remainder (unsigned) (higher 16 bits) MDCL: Remainder (unsigned) (lower 16 bits)

Downloaded from Arrow.com.

The register configuration differs between when multiplication is executed and when division is executed, as follows.

- Register configuration during multiplication

  <Multiplier A> <Multiplier B> <Product>

  MDAL (bits 15 to 0) × MDAH (bits 15 to 0) = [MDBH (bits 15 to 0), MDBL (bits 15 to 0)]

- Register configuration during multiply-accumulation

  < Multiplier A > <Multiplier B> < accumulated value > < accumulated result >

  MDAL (bits 15 to 0) × MDAH (bits 15 to 0) + MDC (bits 31 to 0) = [MDCH (bits 15 to 0), MDCL (bits 15 to 0)]

  (The multiplication result is stored in the MDBH (bits 15 to 0) and MDBL (bits 15 to 0).)

- Register configuration during division

  <Dividend> <Divisor>

  [MDAH (bits 15 to 0), MDAL (bits 15 to 0)] ÷ [MDBH (bits 15 to 0), MDBL (bits 15 to 0)] =

  <Quotient> <Remainder>

  [MDAH (bits 15 to 0), MDAL (bits 15 to 0)] ··· [MDCH (bits 15 to 0), MDCL (bits 15 to 0)]

Downloaded from Arrow.com.

13.3 Register Controlling Multiplier and Divider/Multiply-Accumulator

The multiplier and divider/multiply-accumulator is controlled by using the multiplication/division control register (MDUC).

13.3.1 Multiplication/division control register (MDUC)

The MDUC register is an 8-bit register that controls the operation of the multiplier and divider/multiply-accumulator.

The MDUC register can be set by a 1-bit or 8-bit memory manipulation instruction.

<R> Note that the overflow flag (MACOF) and sign flag (MACSF) of the multiply-accumulation result (accumulated) are read-only flags.

Reset signal generation clears this register to 00H.

Downloaded from Arrow.com.

Figure 13-5. Format of Multiplication/Division Control Register (MDUC)

Address: F00E8H After reset: 00H R/W[Note 1]

Symbol	<7>	<6>	5	4	<3>	<2>	<1>	<0>
MDUC	DIVMODE	MACMODE	0	0	MDSM	MACOF	MACSF	DIVST

<R>

DIVMODE	MACMODE	MDSM	Operation mode selection
0	0	0	Multiplication mode (unsigned) (default)
0	0	1	Multiplication mode (signed)
0	1	0	Multiply-accumulator mode (unsigned)
0	1	1	Multiply-accumulator mode (signed)
1	0	0	Division mode (unsigned), generation of a division completion interrupt (INTMD)
1	1	0	Division mode (unsigned), not generation of a division completion interrupt (INTMD)
Other than above			Setting prohibited

MACOF	Overflow flag of multiply-accumulation result (accumulated value)
0	No overflow
1	With over flow
<Set condition> • For the multiply-accumulator mode (unsigned) The bit is set when the accumulated value goes outside the range from 00000000h to FFFFFFFFh. • For the multiply-accumulator mode (signed) The bit is set when the result of adding a positive product to a positive accumulated value exceeds 7FFFFFFFh and is negative, or when the result of adding a negative product to a negative accumulated value exceeds 80000000h and is positive.	

MACSF	Sign flag of multiply-accumulation result (accumulated value)
0	The accumulated value is positive.
1	The accumulated value is negative.
Multiply-accumulator mode (unsigned):	The bit is always 0.
Multiply-accumulator mode (signed):	The bit indicates the sign bit of the accumulated value.

DIVST[Note 2]	Division operation start/stop
0	Division operation processing complete
1	Starts division operation/division operation processing in progress

Notes
1. Bits 1 and 2 are read-only bits.
2. The DIVST bit can only be set (1) in the division mode. In the division mode, division operation is started by setting (1) the DIVST bit. The DIVST bit is automatically cleared (0) when the operation ends. In the multiplication mode, operation is automatically started by setting the multiplier and multiplicand to multiplication/division data register A (MDAH, MDAL), respectively.

Cautions
1. **Do not rewrite the DIVMODE, MDSM bits during operation processing (while the DIVST bit is 1). If it is rewritten, the operation result will be an undefined value.**
2. **The DIVST bit cannot be cleared (0) by using software during division operation processing (while the DIVST bit is 1).**

Downloaded from Arrow.com.

13.4 Operations of Multiplier and Divider/Multiply-Accumulator

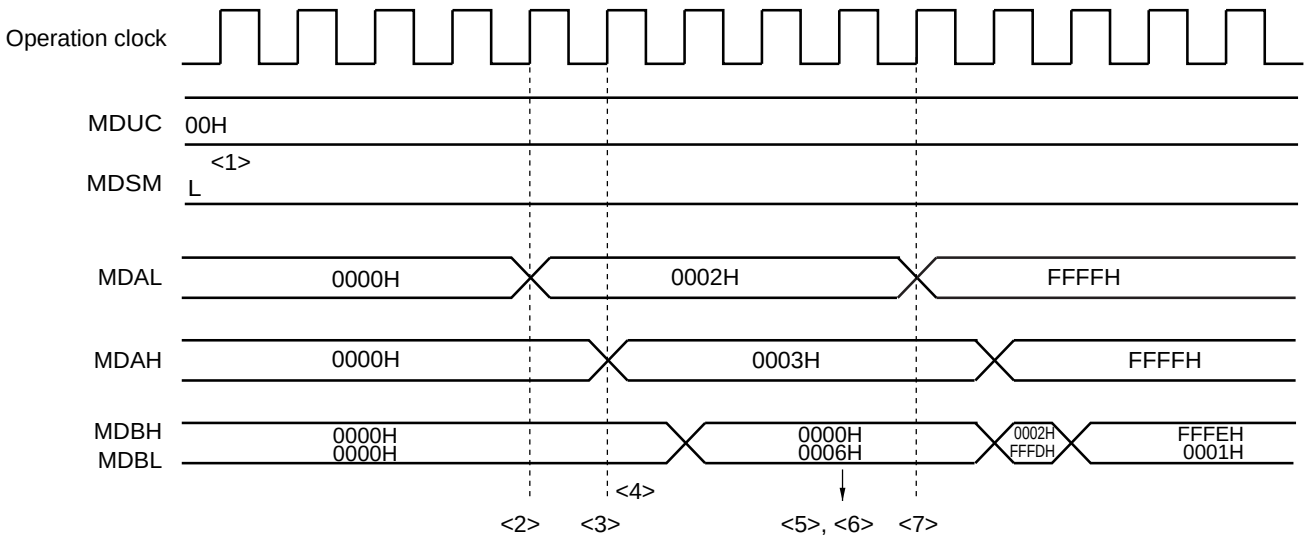
13.4.1 Multiplication (unsigned) operation

- Initial setting
  - <1> Set the multiplication/division control register (MDUC) to 00H.
  - <2> Set the multiplicand to multiplication/division data register A (L) (MDAL).
  - <3> Set the multiplier to multiplication/division data register A (H) (MDAH).
    (There is no preference in the order of executing steps <2> and <3>. Multiplication operation is automatically started when the multiplier and multiplicand are set to the MDAH and MDAL registers, respectively.)
- During operation processing
  - <4> Wait for at least one clock. The operation will end when one clock has been issued.
- Operation end
  - <5> Read the product (lower 16 bits) from multiplication/division data register B (L) (MDBL).
  - <6> Read the product (higher 16 bits) from multiplication/division data register B (H) (MDBH).
    (There is no preference in the order of executing steps <5> and <6>.)
- Next operation

<R> <7> Start with the initial settings of each step to change the operation mode. When the same operation mode is used sequentially, settings <1> and <2> can be omitted.

Remark Steps <1> to <7> correspond to <1> to <7> in Figure 13-6.

Figure 13-6. Timing Diagram of Multiplication (Unsigned) Operation (2 × 3 = 6)



Downloaded from Arrow.com.

13.4.2 Multiplication (signed) operation

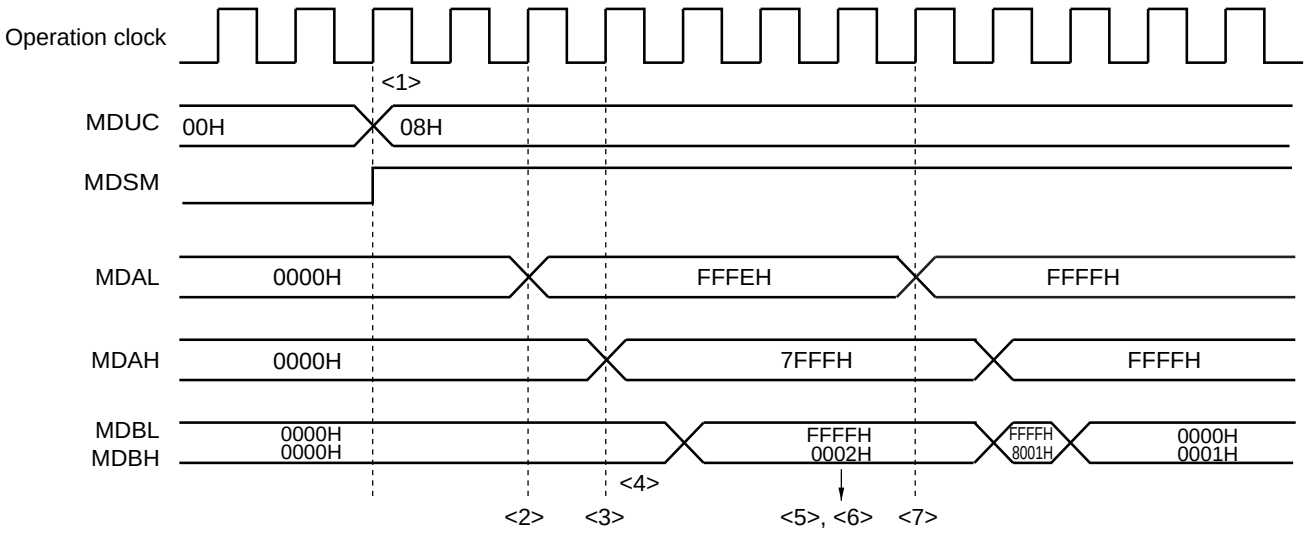
- Initial setting
  - <1> Set the multiplication/division control register (MDUC) to 08H.
  - <2> Set the multiplicand to multiplication/division data register A (L) (MDAL).
  - <3> Set the multiplier to multiplication/division data register A (H) (MDAH).
    (There is no preference in the order of executing steps <2> and <3>. Multiplication operation is automatically started when the multiplier and multiplicand are set to the MDAH and MDAL registers, respectively.)
- During operation processing
  - <4> Wait for at least one clock. The operation will end when one clock has been issued.
- Operation end
  - <5> Read the product (lower 16 bits) from multiplication/division data register B (L) (MDBL).
  - <6> Read the product (higher 16 bits) from multiplication/division data register B (H) (MDBH).
    (There is no preference in the order of executing steps <5> and <6>.)
- Next operation

<R> <7> Start with the initial settings of each step to change the operation mode. When the same operation mode is used sequentially, settings <1> and <2> can be omitted.

Caution The data is in the two's complement format in multiplication mode (signed).

Remark Steps <1> to <7> correspond to <1> to <7> in Figure 13-7.

Figure 13-7. Timing Diagram of Multiplication (Signed) Operation (–2 × 32767 = –65534)



Downloaded from Arrow.com.

13.4.3 Multiply-accumulation (unsigned) operation

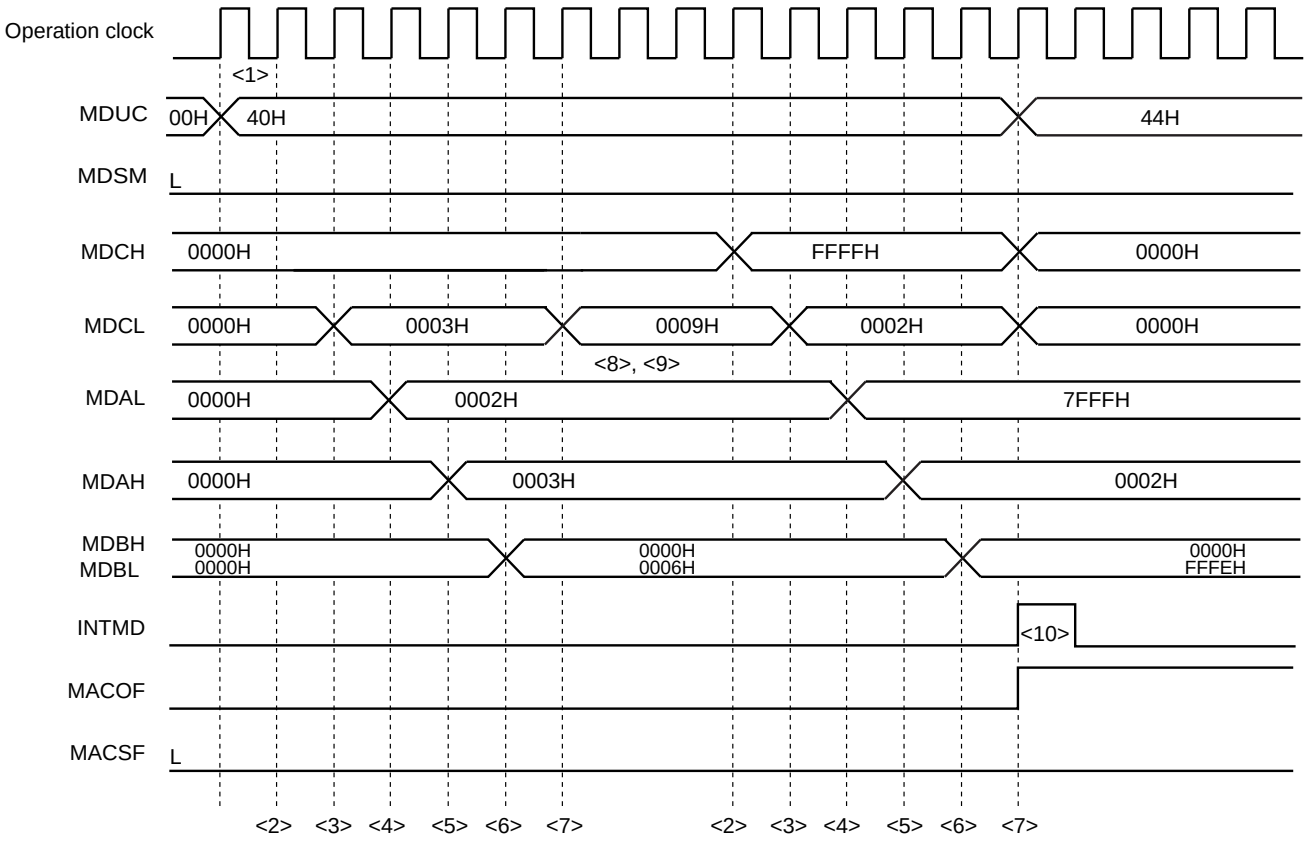
- Initial setting
  - <1> Set the multiplication/division control register (MDUC) to 40H.
  - <2> Set the initial accumulated value of higher 16 bits to multiplication/division data register C (L) (MDCL).
  - <3> Set the initial accumulated value of lower 16 bits to multiplication/division data register C (H) (MDCH).
  - <4> Set the multiplicand to multiplication/division data register A (L) (MDAL).
  - <5> Set the multiplier to multiplication/division data register A (H) (MDAH).
    (There is no preference in the order of executing steps <2>, <3>, and <4>. Multiplication operation is automatically started when the multiplier is set to the MDAH register, respectively.)
- During operation processing
  - <6> The multiplication operation finishes in one clock cycle.
    (The multiplication result is stored in multiplication/division data register B (L) (MDBL) and multiplication/division data register B (H) (MDBH).)
  - <7> After <6>, the multiply-accumulation operation finishes in one additional clock cycle. (There is a wait of at least two clock cycles after specifying the initial settings is finished (<5>).)
- Operation end
  - <8> Read the accumulated value (lower 16 bits) from the MDCL register.
  - <9> Read the accumulated value (higher 16 bits) from the MDCH register.
    (There is no preference in the order of executing steps <8> and <9>.)
  - (<10> If the result of the multiply-accumulation operation causes an overflow, the MACOF bit is set to 1, INTMD signal is occurred.)
- Next operation

<R> <11> Start with the initial settings of each step to change the operation mode. When the same operation mode is used sequentially, settings <1> and <4> can be omitted.

Remark Steps <1> to <10> correspond to <1> to <10> in Figure 13-8.

Downloaded from Arrow.com.

Figure 13-8. Timing Diagram of Multiply-Accumulation (Unsigned) Operation
(2 × 3 + 3 = 9 → 32767 × 2 + 4294901762 = 0 (over flow generated))



Downloaded from Arrow.com.

13.4.4 Multiply-accumulation (signed) operation

- Initial setting
  - <1> Set the multiplication/division control register (MDUC) to 48H.
  - <2> Set the initial accumulated value of higher 16 bits to multiplication/division data register C (H) (MDCH).
    (<3> If the accumulated value in the MDCH register is negative, the MACSF bit is set to 1.)
  - <4> Set the initial accumulated value of lower 16 bits to multiplication/division data register C (L) (MDCL).
  - <5> Set the multiplicand to multiplication/division data register A (L) (MDAL).
  - <6> Set the multiplier to multiplication/division data register A (H) (MDAH).
    (There is no preference in the order of executing steps <2>, <4>, and <5>. Multiplication operation is automatically started when the multiplier is set to the MDAH register of <6>, respectively.)
- During operation processing
  - <7> The multiplication operation finishes in one clock cycle.
    (The multiplication result is stored in multiplication/division data register B (L) (MDBL) and multiplication/division data register B (H) (MDBH).)
  - <8> After <7>, the multiply-accumulation operation finishes in one additional clock cycle. (There is a wait of at least two clock cycles after specifying the initial settings is finished (<6>).)
- Operation end
  - <9> If the accumulated value stored in the MDCL and MDCH registers is positive, the MACSF bit is cleared to 0.
  - <10> Read the accumulated value (lower 16 bits) from the MDCL register.
  - <11> Read the accumulated value (higher 16 bits) from the MDCH register.
    (There is no preference in the order of executing steps <10> and <11>.)
  - (<12> If the result of the multiply-accumulation operation causes an overflow, the MACOF bit is set to 1, INTMD signal is occurred.)
- Next operation

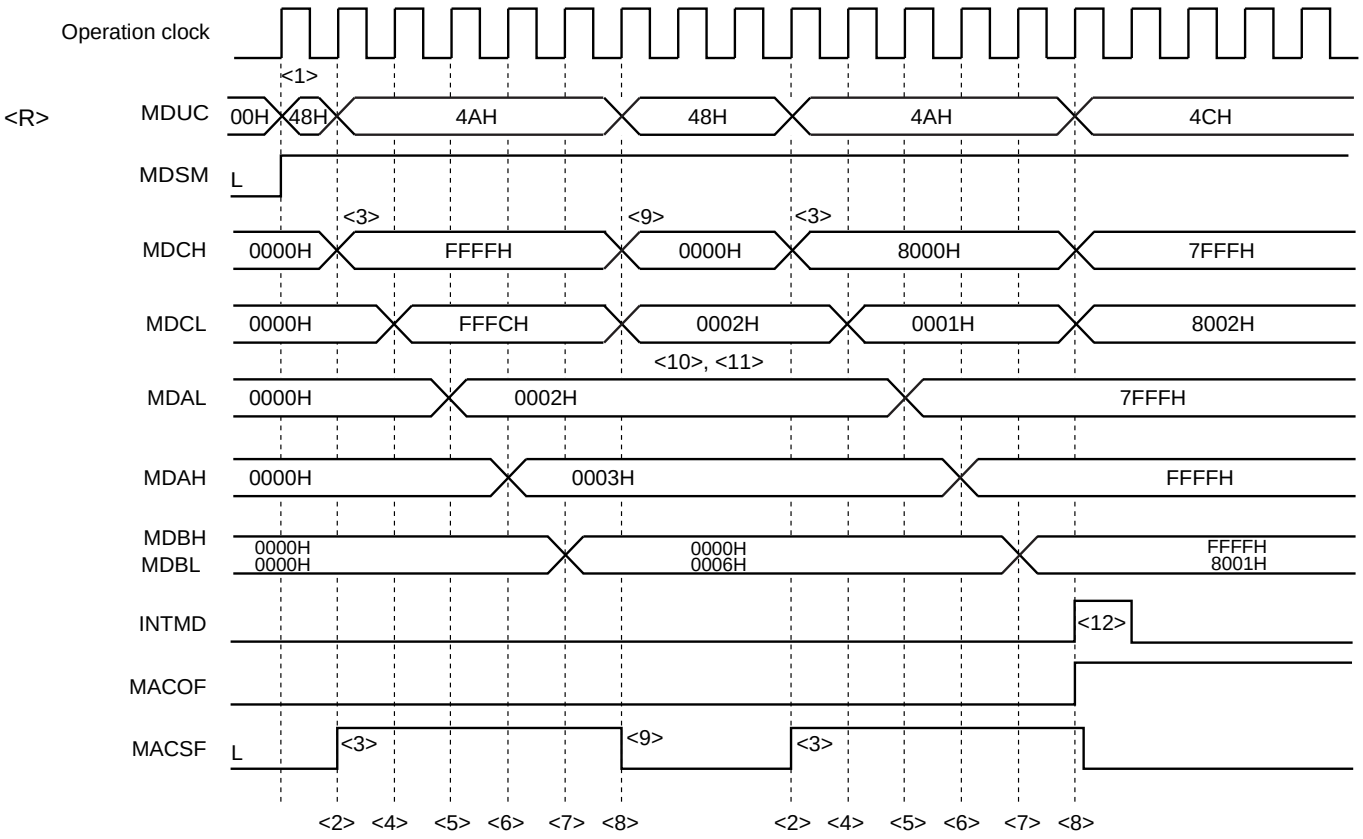
<R> <13> Start with the initial settings of each step to change the operation mode. When the same operation mode is used sequentially, settings <1> and <5> can be omitted.

Caution The data is in the two's complement format in multiply-accumulation (signed) operation.

Remark Steps <1> to <12> correspond to <1> to <12> in Figure 13-9.

Downloaded from Arrow.com.

Figure 13-9. Timing Diagram of Multiply-Accumulation (signed) Operation
(2 × 3 + (–4) = 2 → 32767 × (–1) + (–2147483647) = –2147450882 (overflow occurs.))



Downloaded from Arrow.com.

13.4.5 Division operation

- Initial setting
  - <1> Set the multiplication/division control register (MDUC) to 80H.
  - <2> Set the dividend (higher 16 bits) to multiplication/division data register A (H) (MDAH).
  - <3> Set the dividend (lower 16 bits) to multiplication/division data register A (L) (MDAL).
  - <4> Set the divisor (higher 16 bits) to multiplication/division data register B (H) (MDBH).
  - <5> Set the divisor (lower 16 bits) to multiplication/division data register B (L) (MDBL).
  - <6> Set bit 0 (DIVST) of the MDUC register to 1.
    (There is no preference in the order of executing steps <2> to <5>.)
- During operation processing
  - <7> The operation will end when one of the following processing is completed.
    - A wait of at least 16 clocks (The operation will end when 16 clocks have been issued.)
    - A check whether the DIVST bit has been cleared

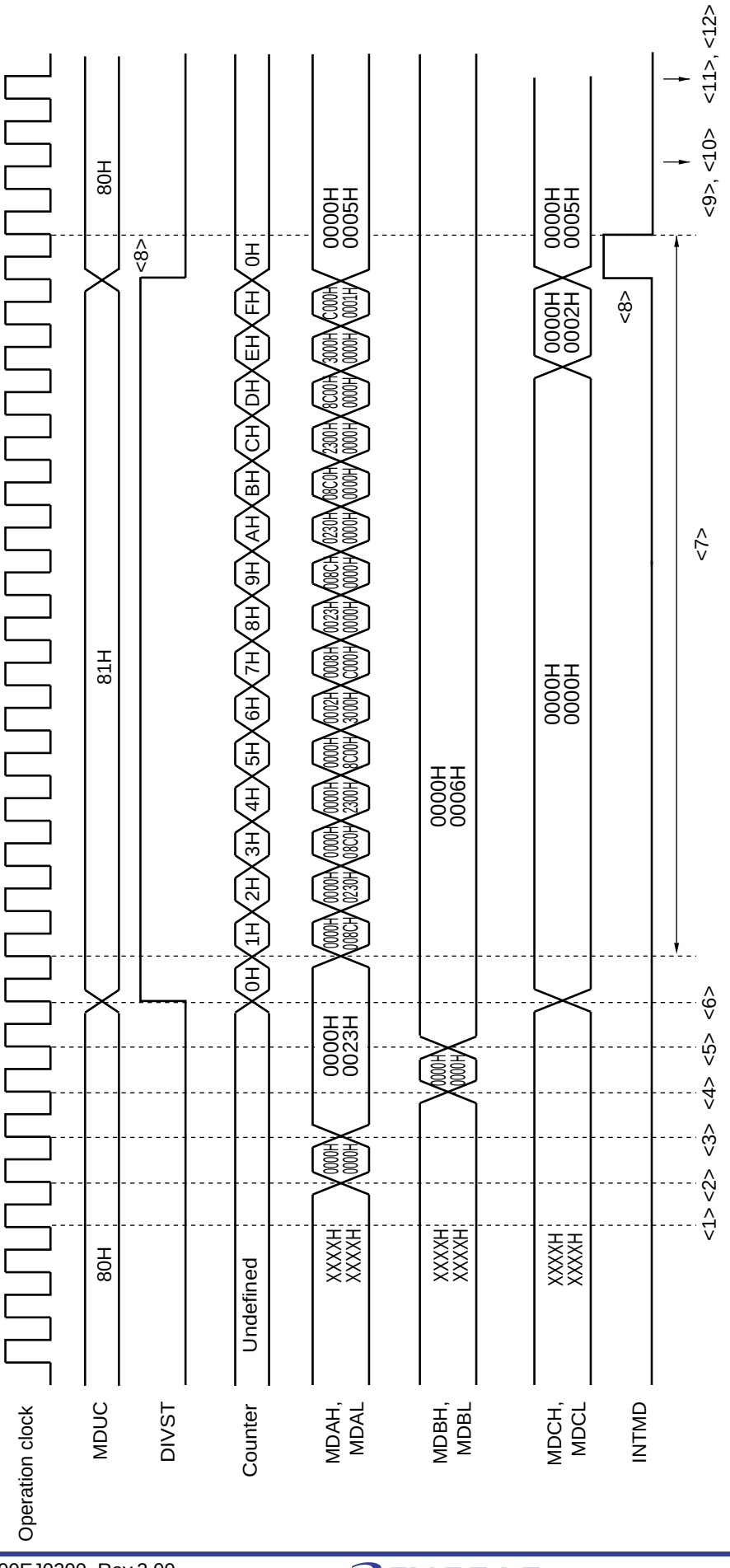
    (The read values of the MDBL, MDBH, MDCL, and MDCH registers during operation processing are not guaranteed.)
- Operation end
  - <8> The DIVST bit is cleared and the operation ends. At this time, an interrupt request signal (INTMD) is generated if the operation was performed with MACMODE = 0.
  - <9> Read the quotient (lower 16 bits) from the MDAL register.
  - <10> Read the quotient (higher 16 bits) from the MDAH register.
  - <11> Read the remainder (lower 16 bits) from multiplication/division data register C (L) (MDCL).
  - <12> Read the remainder (higher 16 bits) from multiplication/division data register C (H) (MDCH).
    (There is no preference in the order of executing steps <9> to <12>.)
- Next operation

<R> <13> Start with the initial settings of each step to change the operation mode. When the same operation mode is used sequentially, settings <1> and <5> can be omitted.

Remark Steps <1> to <12> correspond to <1> to <12> in Figure 13-10.

Downloaded from Arrow.com.

Figure 13-10. Timing Diagram of Division Operation (Example: 35 ÷ 6 = 5, Remainder 5)



Downloaded from Arrow.com.

CHAPTER 14 DMA CONTROLLER

The R5F102 products of the RL78/G12 have an internal DMA (Direct Memory Access) controller.

Data can be automatically transferred between the peripheral hardware supporting DMA, SFRs, and internal RAM without via CPU.

As a result, the normal internal operation of the CPU and data transfer can be executed in parallel with transfer between the SFR and internal RAM, and therefore, a large capacity of data can be processed. In addition, real-time control using communication, timer, and A/D can also be realized.

14.1 Functions of DMA Controller

- Number of DMA channels: 2 channels (R5F102 products)
- Transfer unit: 8 or 16 bits
- Maximum transfer unit: 1024 times
- Transfer type: 2-cycle transfer (One transfer is processed in 2 clocks and the CPU stops during that processing.)
- Transfer mode: Single-transfer mode
- Transfer request: Selectable from the following peripheral hardware interrupts
  - A/D converter
  - Serial interface (CSI00, CSI01, CSI11, CSI20, UART0 to UART2)
  - Timer (channel 0, 1, 2, 3)
- Transfer target: Between SFR and internal RAM

Here are examples of functions using DMA.

- Successive transfer of serial interface
- <R> Consecutive capturing of A/D conversion results
- Capturing A/D conversion result at fixed interval
- Capturing port value at fixed interval

Downloaded from Arrow.com.

14.2 Configuration of DMA Controller

The DMA controller includes the following hardware.

Table 14-1. Configuration of DMA Controller

Item	Configuration
Address registers	• DMA SFR address registers 0, 1 (DSA0, DSA1) • DMA RAM address registers 0, 1 (DRA0, DRA1)
Count register	• DMA byte count registers 0, 1 (DBC0, DBC1)
Control registers	• DMA mode control registers 0, 1 (DMC0, DMC1) • DMA operation control register 0, 1 (DRC0, DRC1)

14.2.1 DMA SFR address register n (DSAn)

This is an 8-bit register that is used to set an SFR address that is the transfer source or destination of DMA channel n.

Set the lower 8 bits of the SFR addresses FFF00H to FFFFFH.

This register is not automatically incremented but fixed to a specific value.

In the 16-bit transfer mode, the least significant bit is ignored and is treated as an even address.

The DSAn register can be read or written in 8-bit units. However, it cannot be written during DMA transfer.

Reset signal generation clears this register to 00H.

Figure 14-1. Format of DMA SFR Address Register n (DSAn)

Address: FFFB0H (DSA0), FFFB1H (DSA1) After reset: 00H R/W

	7	6	5	4	3	2	1	0
DSAn								

Remark n: DMA channel number (n = 0, 1)

Downloaded from Arrow.com.

14.2.2 DMA RAM address register n (DRAn)

This is a 16-bit register that is used to set a RAM address that is the transfer source or destination of DMA channel n.

Addresses of the internal RAM area other than the general-purpose registers (see table 14-2) can be set to this register.

Set the lower 16 bits of the RAM address.

This register is automatically incremented when DMA transfer has been started. It is incremented by +1 in the 8-bit transfer mode and by +2 in the 16-bit transfer mode. DMA transfer is started from the address set to this DRAn register. When the data of the last address has been transferred, the DRAn register stops with the value of the last address +1 in the 8-bit transfer mode, and the last address +2 in the 16-bit transfer mode.

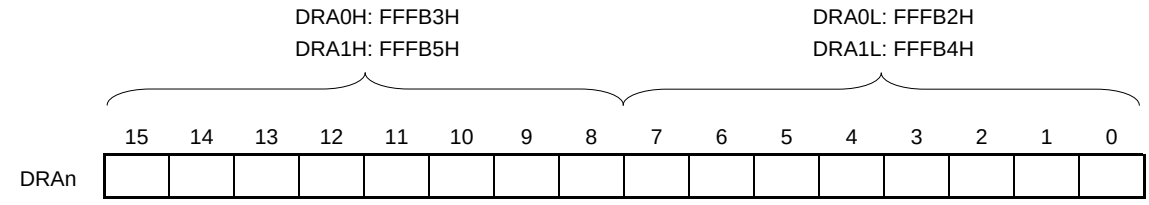
In the 16-bit transfer mode, the least significant bit is ignored and is treated as an even address.

The DRAn register can be read or written in 8-bit or 16-bit units. However, it cannot be written during DMA transfer.

Reset signal generation clears this register to 0000H.

Figure 14-2. Format of DMA RAM Address Register n (DRAn)

Address: FFFB2H, FFFB3H (DRA0), FFFB4H, FFFB5H (DRA1) , After reset: 0000H R/W



Remark n: DMA channel number (n = 0, 1)

Table 14-2 Internal RAM Area other than the General-purpose Registers

Part Number	Internal RAM Area other than the General-purpose Registers
R5F10x66	256 × 8 bits (FFE00H to FFEDFH)
R5F10x67, R5F10x77, R5F10xA7	512 × 8 bits (FFD00H to FFEDFH)
R5F10x68, R5F10x78, R5F10xA8	768 × 8 bits (FFC00H to FFEDFH)
R5F10x69, R5F10x79, R5F10xA9	1024 × 8 bits (FFB00H to FFEDFH)
R5F10x6A, R5F10x7A	1536 × 8 bits (FF900H to FFEDFH)
R5F10xAA	2048 × 8 bits (FF700H to FFEDFH)

(x = 2, 3)

Downloaded from Arrow.com.

14.2.3 DMA byte count register n (DBCn)

This is a 10-bit register that is used to set the number of times DMA channel n executes transfer. Be sure to set the number of times of transfer to this DBCn register before executing DMA transfer (up to 1024 times).

Each time DMA transfer has been executed, this register is automatically decremented. By reading this DBCn register during DMA transfer, the remaining number of times of transfer can be learned.

The DBCn register can be read or written in 8-bit or 16-bit units. However, it cannot be written during DMA transfer.

Reset signal generation clears this register to 0000H.

Figure 14-3. Format of DMA Byte Count Register n (DBCn)

Address: FFFB6H, FFFB7H (DBC0), FFFB8H, FFFB9H (DBC1) After reset: 0000H R/W

DBC0H:FFFB7H
DBC1H:FFFB9H

DBC0L:FFFB6H
DBC1L:FFFB8H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DBCn	0	0	0	0	0	0										

DBCn[9:0]	Number of Times of Transfer (When DBCn is Written)	Remaining Number of Times of Transfer (When DBCn is Read)
000H	1024	Completion of transfer or waiting for 1024 times of DMA transfer
001H	1	Waiting for remaining one time of DMA transfer
002H	2	Waiting for remaining two times of DMA transfer
003H	3	Waiting for remaining three times of DMA transfer
• • •	• • •	• • •
3FEH	1022	Waiting for remaining 1022 times of DMA transfer
3FFH	1023	Waiting for remaining 1023 times of DMA transfer

Cautions 1. Be sure to clear bits 15 to 10 to 0.

2. If the general-purpose register is specified or the internal RAM space is exceeded as a result of continuous transfer, the general-purpose register or SFR space are written or read, resulting in loss of data in these spaces. Be sure to set the number of times of transfer that is within the internal RAM space.

Remark n: DMA channel number (n = 0, 1)

Downloaded from Arrow.com.

14.3 Registers Controlling DMA Controller

DMA controller is controlled by the following registers.

- DMA mode control register n (DMCn)
- DMA operation control register n (DRCn)

Remark n: DMA channel number (n = 0, 1)

14.3.1 DMA mode control register n (DMCn)

The DMCn register is a register that is used to set a transfer mode of DMA channel n. It is used to select a transfer direction, data size, setting of pending, and start source. Bit 7 (STGn) is a software trigger that starts DMA.

Rewriting bits 6, 5, and 3 to 0 of the DMCn register is prohibited during operation (when DSTn = 1).

The DMCn register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 14-4. Format of DMA Mode Control Register n (DMCn) (1/2)

Address: FFFBAH (DMC0), FFFBBH (DMC1) After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	3	2	1	0
DMCn	STGn	DRSn	DSn	DWAITn	IFCn3	IFCn2	IFCn1	IFCn0

STGn[Note 1]	DMA transfer start software trigger
0	No trigger operation
1	DMA transfer is started when DMA operation is enabled (DENn = 1).
DMA transfer is performed once by writing 1 to the STGn bit when DMA operation is enabled (DENn = 1). When this bit is read, 0 is always read.	

DRSn	Selection of DMA transfer direction
0	SFR to internal RAM
1	Internal RAM to SFR

DSn	Specification of transfer data size for DMA transfer
0	8 bits
1	16 bits

DWAITn [Note 2]	Pending of DMA transfer
0	Executes DMA transfer upon DMA start request (not held pending).
1	Holds DMA start request pending if any.
DMA transfer that has been held pending can be started by clearing the value of the DWAITn bit to 0. It takes 2 clocks to actually hold DMA transfer pending when the value of the DWAITn bit is set to 1.	

Notes 1. The software trigger (STGn) can be used regardless of the IFCn0 to IFCn3 bits values.

2. When DMA transfer is held pending while using two or more DMA channels, be sure to hold the DMA transfer pending for all channels (by setting the DWAIT0, DWAIT1, DWAIT2, and DWAIT3 bits to 1).

Remark n: DMA channel number (n = 0, 1)

Downloaded from Arrow.com.

Figure 14-4. Format of DMA Mode Control Register n (DMCn) (2/2)

Address: FFFBAH (DMC0), FFFBBH (DMC1) After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	3	2	1	0
DMCn	STGn	DRSn	DSn	DWAITn	IFCn3	IFCn2	IFCn1	IFCn0

IFCn3	IFCn2	IFCn1	IFCn0	Selection of DMA start source[Note]	
				Trigger signal	Trigger contents
0	0	0	0	–	Disables DMA transfer by interrupt. (Only software trigger is enabled.)
0	0	0	1	INTAD	A/D conversion end interrupt
0	0	1	0	INTTM00	End of timer channel 0 count or capture end interrupt
0	0	1	1	INTTM01	End of timer channel 1 count or capture end interrupt
0	1	0	0	INTTM02	End of timer channel 2 count or capture end interrupt
0	1	0	1	INTTM03	End of timer channel 3 count or capture end interrupt
0	1	1	0	INTST0/INTCSI00	UART0 transmission transfer end or buffer empty interrupt/CSI00 transfer end or buffer empty interrupt
0	1	1	1	INTSR0/INTCSI01	UART0 reception transfer end interrupt/CSI01 transfer end or buffer empty interrupt
1	0	0	0	INTST1	UART1 transmission transfer end or buffer empty interrupt
1	0	0	1	INTSR1/INTCSI11	UART1 reception transfer end interrupt/CSI11 transfer end or buffer empty interrupt
1	0	1	0	INTST2/INTCSI20	UART2 transmission transfer end or buffer empty interrupt/CSI20 transfer end or buffer empty interrupt
1	0	1	1	INTSR2	UART2 reception transfer end interrupt
Other than above				Setting prohibited	

Note The software trigger (STGn) can be used regardless of the IFCn0 to IFCn3 bits values.

Remark n: DMA channel number (n = 0, 1)

Downloaded from Arrow.com.

14.3.2 DMA operation control register n (DRCn)

The DRCn register is a register that is used to enable or disable transfer of DMA channel n.

Rewriting bit 7 (DENn) of this register is prohibited during operation (when DSTn = 1).

The DRCn register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 14-5. Format of DMA Operation Control Register n (DRCn)

Address: FFFBCH (DRC0), FFFBDH (DRC1) After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	<0>
DRCn	DENn	0	0	0	0	0	0	DSTn

DENn	DMA operation enable flag
0	Disables operation of DMA channel n (stops operating cock of DMA).
1	Enables operation of DMA channel n.
DMAC waits for a DMA trigger when DSTn = 1 after DMA operation is enabled (DENn = 1).	

DSTn	DMA transfer mode flag
0	DMA transfer of DMA channel n is completed.
1	DMA transfer of DMA channel n is not completed (still under execution).
DMAC waits for a DMA trigger when DSTn = 1 after DMA operation is enabled (DENn = 1). When a software trigger (STGn) or the start source trigger set by the IFCn3 to IFCn0 bits is input, DMA transfer is started. When DMA transfer is completed after that, this bit is automatically cleared to 0. Write 0 to this bit to forcibly terminate DMA transfer under execution.	

Caution The DSTn flag is automatically cleared to 0 when a DMA transfer is completed. Writing the DENn flag is enabled only when DSTn = 0. When a DMA transfer is terminated without waiting for generation of the interrupt (INTDMAn) of DMAn, therefore, set the DSTn bit to 0 and then the DENn bit to 0 (for details, refer to 14.5.5 Forced termination by software).

Remark n: DMA channel number (n = 0, 1)

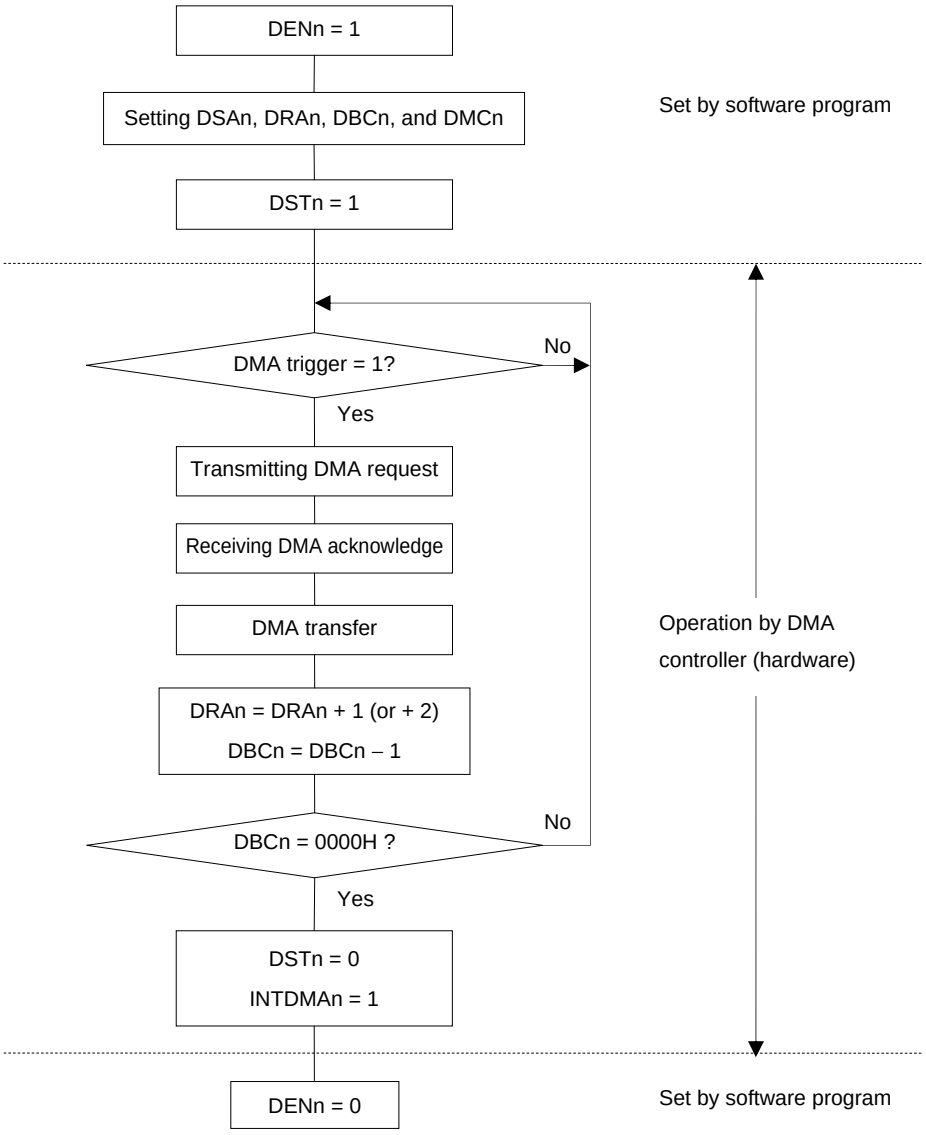
Downloaded from Arrow.com.

14.4 Operation of DMA Controller

14.4.1 Operation procedure

<1> The DMA controller is enabled to operate when DENn = 1. Before writing the other registers, be sure to set the DENn bit to 1. Use 80H to write with an 8-bit manipulation instruction.
<2> Set an SFR address, a RAM address, the number of times of transfer, and a transfer mode of DMA transfer to DMA SFR address register n (DSAn), DMA RAM address register n (DRAn), DMA byte count register n (DBCn), and DMA mode control register n (DMCn).
<3> The DMA controller waits for a DMA trigger when DSTn = 1. Use 81H to write with an 8-bit manipulation instruction.
<4> When a software trigger (STGn) or a start source trigger specified by the IFCn3 to IFCn0 bits is input, a DMA transfer is started.
<5> Transfer is completed when the number of times of transfer set by the DBCn register reaches 0, and transfer is automatically terminated by occurrence of an interrupt (INTDMAn).
<6> Stop the operation of the DMA controller by clearing the DENn bit to 0 when the DMA controller is not used.

Figure 14-6. Operation Procedure



Remark n: DMA channel number (n = 0, 1)

Downloaded from Arrow.com.

14.4.2 Transfer mode

The following four modes can be selected for DMA transfer by using bits 6 and 5 (DRSn and DSn) of DMA mode control register n (DMCn).

DRSn	DSn	DMA Transfer Mode
0	0	Transfer from SFR of 1-byte data (fixed address) to RAM (address is incremented by +1)
0	1	Transfer from SFR of 2-byte data (fixed address) to RAM (address is incremented by +2)
1	0	Transfer from RAM of 1-byte data (address is incremented by +1) to SFR (fixed address)
1	1	Transfer from RAM of 2-byte data (address is incremented by +2) to SFR (fixed address)

By using these transfer modes, up to 1024 bytes of data can be consecutively transferred by using the serial interface, data resulting from A/D conversion can be consecutively transferred, and port data can be scanned at fixed time intervals by using a timer.

14.4.3 Termination of DMA transfer

When DBCn = 00H and DMA transfer is completed, the DSTn bit is automatically cleared to 0. An interrupt request (INTDMAn) is generated and transfer is terminated.

When the DSTn bit is cleared to 0 to forcibly terminate DMA transfer, DMA byte count register n (DBCn) and DMA RAM address register n (DRAn) hold the value when transfer is terminated.

The interrupt request (INTDMAn) is not generated if transfer is forcibly terminated.

Remark n: DMA channel number (n = 0, 1)

14.5 Example of Setting of DMA Controller

14.5.1 CSI consecutive transmission

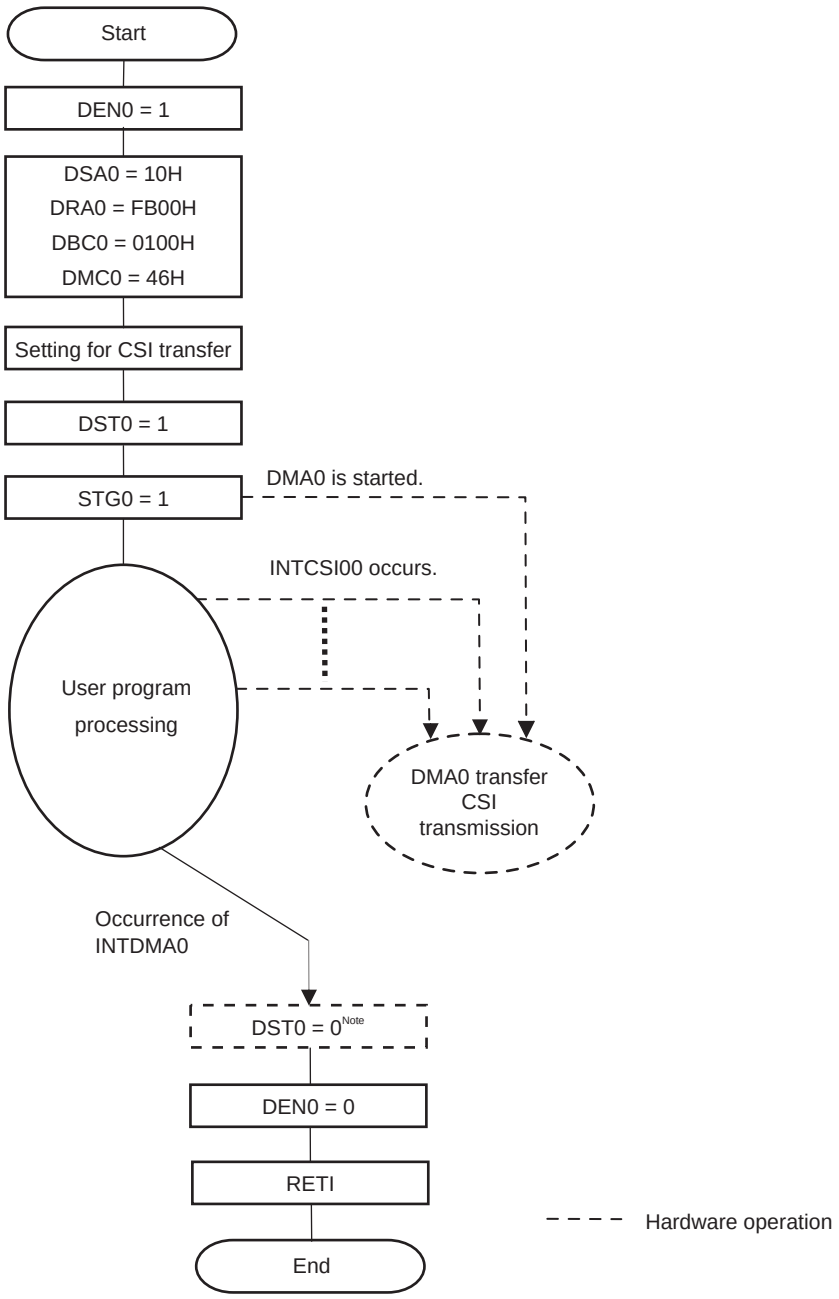
A flowchart showing an example of setting for CSI consecutive transmission is shown below.

- Consecutive transmission of CSI00 (256 bytes)
- DMA channel 0 is used for DMA transfer.
- DMA start source: INTCSI00 (software trigger (STG0) only for the first start source)
- Interrupt of CSI00 is specified by IFC03 to IFC00 = 0110B.
- Transfers FFB00H to FFBFFH (256 bytes) of RAM to FFF10H of the data register (SIO00) of CSI.

Remark IFC03 to IFC00: Bits 3 to 0 of DMA mode control registers 0 (DMC0)

Downloaded from Arrow.com.

Figure 14-7. Example of Setting for CSI Consecutive Transmission



Note The DST0 flag is automatically cleared to 0 when a DMA transfer is completed.
Writing the DEN0 flag is enabled only when DST0 = 0. To terminate a DMA transfer without waiting for occurrence of the interrupt of DMA0 (INTDMA0), set the DST0 bit to 0 and then the DEN0 bit to 0 (for details, refer to **14.5.5 Forced termination by software**).

The first trigger for consecutive transmission is not started by the interrupt of CSI. In this example, it start by a software trigger.

CSI transmission of the second time and onward is automatically executed.

A DMA interrupt (INTDMA0) occurs when the last transmit data has been written to the data register.

Downloaded from Arrow.com.

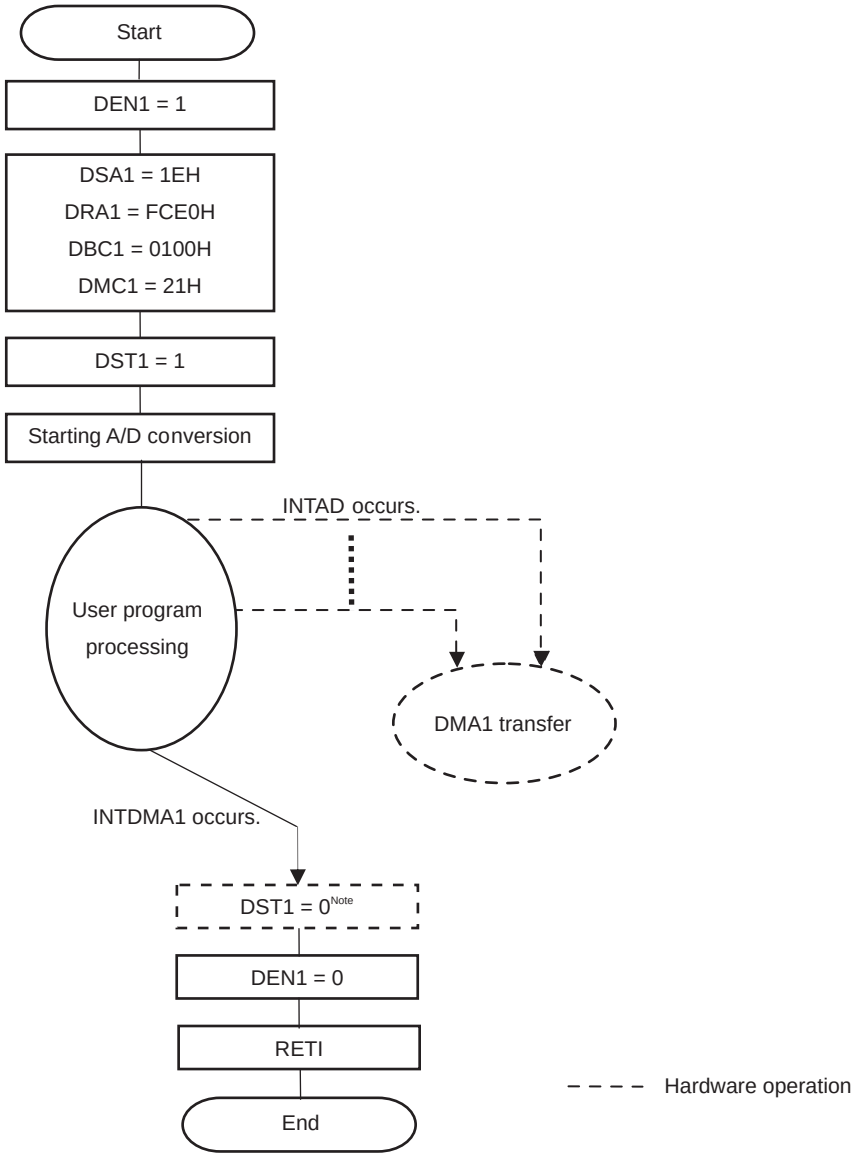
14.5.2 Consecutive capturing of A/D conversion results

A flowchart of an example of setting for consecutively capturing A/D conversion results is shown below.

- Consecutive capturing of A/D conversion results.
- DMA channel 1 is used for DMA transfer.
- DMA start source: INTAD
- Interrupt of A/D is specified by IFC13 to IFC10 = 0001B.
- Transfers FFF1EH and FFF1FH (2 bytes) of the 10-bit A/D conversion result register (ADCR) to 512 bytes of FFCE0H to FFEDFH of RAM.

Remark IFC13 to IFC10: Bits 3 to 0 of DMA mode control registers 1 (DMC1)

Figure 14-8. Example of Setting of Consecutively Capturing A/D Conversion Results



Note The DST1 flag is automatically cleared to 0 when a DMA transfer is completed.
Writing the DEN1 flag is enabled only when DST1 = 0. To terminate a DMA transfer without waiting for occurrence of the interrupt of DMA1 (INTDMA1), set the DST1 bit to 0 and then the DEN1 bit to 0 (for details, refer to **14.5.5 Forced termination by software**).

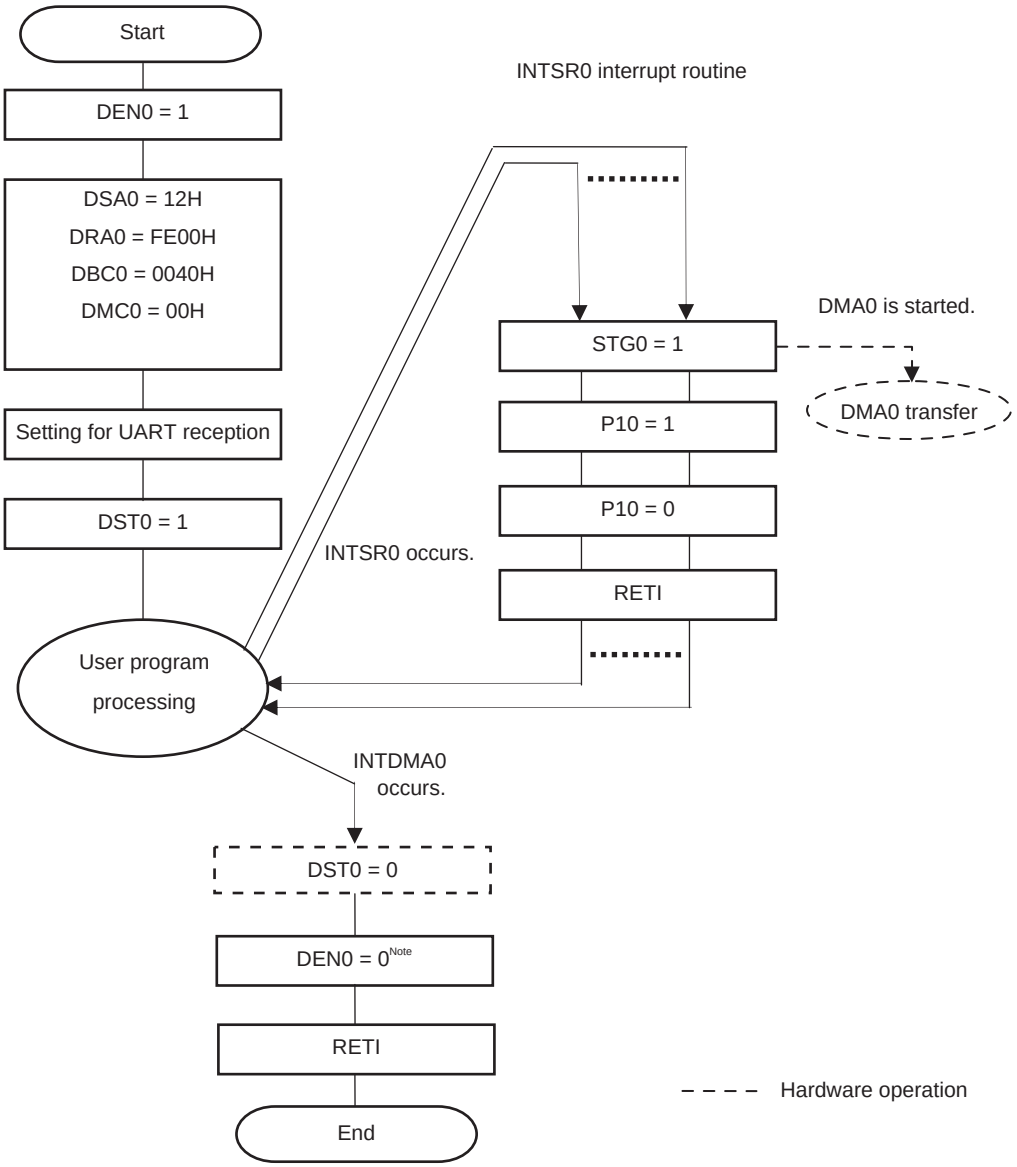
Downloaded from Arrow.com.

14.5.3 UART consecutive reception + ACK transmission

A flowchart illustrating an example of setting for UART consecutive reception + ACK transmission is shown below.

- Consecutively receives data from UART0 and outputs ACK to P10 on completion of reception.
- DMA channel 0 is used for DMA transfer.
- DMA start source: Software trigger (DMA transfer on occurrence of an interrupt is disabled.)
- Transfers FFF12H of UART receive data register 0 (RXD0) to 64 bytes of FFE00H to FFE3FH of RAM.

<R> **Figure 14-9. Example of Setting for UART Consecutive Reception + ACK Transmission**



Note The DST0 flag is automatically cleared to 0 when a DMA transfer is completed.
Writing the DEN0 flag is enabled only when DST0 = 0. To terminate a DMA transfer without waiting for occurrence of the interrupt of DMA0 (INTDMA0), set the DST0 bit to 0 and then the DEN0 bit to 0 (for details, refer to **14.5.5 Forced termination by software**).

Remark This is an example where a software trigger is used as a DMA start source.
If ACK is not transmitted and if only data is consecutively received from UART, the UART reception end interrupt (INTSR0) can be used to start DMA for data reception.

Downloaded from Arrow.com.

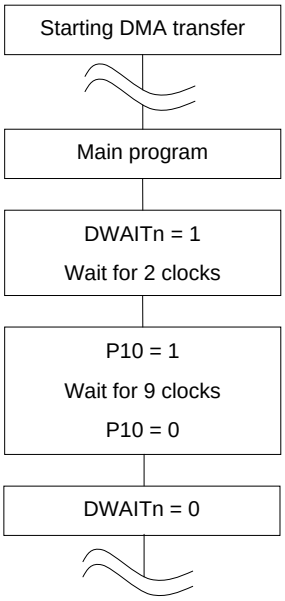
14.5.4 Holding DMA transfer pending by DWAITn bit

When DMA transfer is started, transfer is performed while an instruction is executed. At this time, the operation of the CPU is stopped and delayed for the duration of 2 clocks. If this poses a problem to the operation of the set system, a DMA transfer can be held pending by setting the DWAITn bit to 1. The DMA transfer for a transfer trigger that occurred while DMA transfer was held pending is executed after the pending status is canceled. However, because only one transfer trigger can be held pending for each channel, even if multiple transfer triggers occur for one channel during the pending status, only one DMA transfer is executed after the pending status is canceled.

To output a pulse with a width of 10 clocks of the operating frequency from the P10 pin, for example, the clock width increases to 12 if a DMA transfer is started midway. In this case, the DMA transfer can be held pending by setting the DWAITn bit to 1.

After setting the DWAITn bit to 1, it takes two clocks until a DMA transfer is held pending.

Figure 14-10. Example of Setting for Holding DMA Transfer Pending by DWAITn Bit



Caution When DMA transfer is held pending while using two or more DMA channels, be sure to held the DMA transfer pending for all channels (by setting DWAIT0, DWAIT1, DWAIT2, and DWAIT3 to 1). If the DMA transfer of one channel is executed while that of the other channel is held pending, DMA transfer might not be held pending for the latter channel.

Remarks 1. n: DMA channel number (n = 0, 1)

2. 1 clock: $1/f_{CLK}$ (f_{CLK}: CPU clock)

Downloaded from Arrow.com.

14.5.5 Forced termination by software

After the DSTn bit is set to 0 by software, it takes up to 2 clocks until a DMA transfer is actually stopped and the DSTn bit is set to 0. To forcibly terminate a DMA transfer by software without waiting for occurrence of the interrupt (INTDMAn) of DMAn, therefore, perform either of the following processes.

<When using one DMA channel>

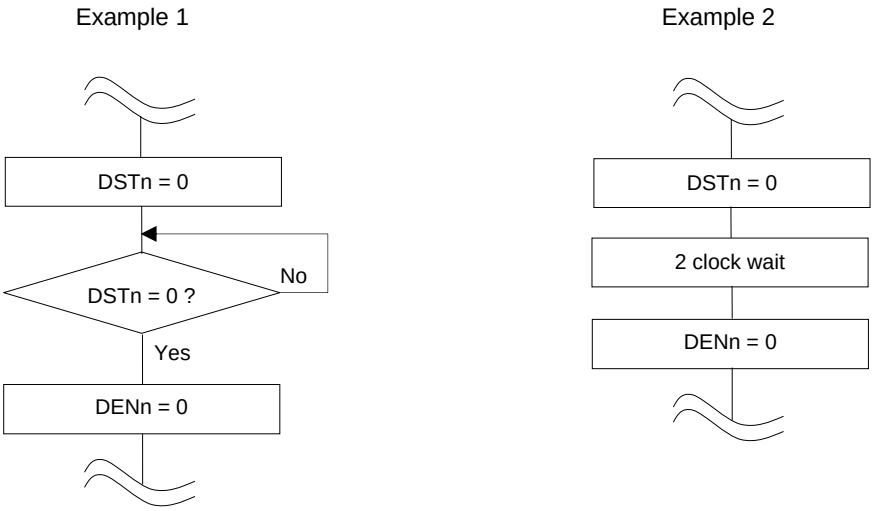
- Set the DSTn bit to 0 (use DRCn = 80H to write with an 8-bit manipulation instruction) by software, confirm by polling that the DSTn bit has actually been cleared to 0, and then set the DENn bit to 0 (use DRCn = 00H to write with an 8-bit manipulation instruction).

- Set the DSTn bit to 0 (use DRCn = 80H to write with an 8-bit manipulation instruction) by software and then set the DENn bit to 0 (use DRCn = 00H to write with an 8-bit manipulation instruction) two or more clocks after.

<When using two DMA channels>

- To forcibly terminate DMA transfer by software when using two DMA channels (by setting DSTn to 0), clear the DSTn bit to 0 after the DMA transfer is held pending by setting the DWAITn bits of both using channels to 1. Next, clear the DWAITn bits of both using channels to 0 to cancel the pending status, and then clear the DENn bit to 0.

Figure 14-11. Forced Termination of DMA Transfer (1/2)



Remarks 1. n: DMA channel number (n = 0, 1)

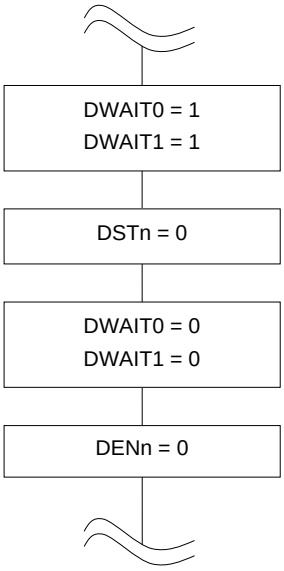
2. 1 clock: $1/f_{CLK}$ (f_{CLK}: CPU clock)

Downloaded from Arrow.com.

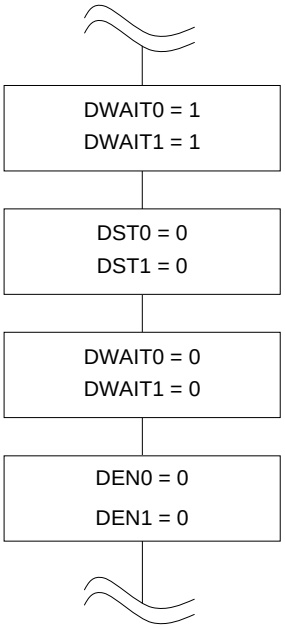
Figure 14-11. Forced Termination of DMA Transfer (2/2)

Example 3

- Procedure for forcibly terminating the DMA transfer for one channel if both channels are used



- Procedure for forcibly terminating the DMA transfer for both channels if both channels are used



Caution In example 3, the system is not required to wait two clock cycles after the DWAITn bit is set to 1. In addition, the system does not have to wait two clock cycles after clearing the DSTn bit to 0, because more than two clock cycles elapse from when the DSTn bit is cleared to 0 to when the DENn bit is cleared to 0.

Remarks 1. n: DMA channel number (n = 0, 1)

2. 1 clock: $1/f_{CLK}$ (f_{CLK}: CPU clock)

Downloaded from Arrow.com.

14.6 Cautions on Using DMA Controller

(1) Priority of DMA

During DMA transfer, a request from the other DMA channel is held pending even if generated. The pending DMA transfer is started after the ongoing DMA transfer is completed. If two or more DMA requests are generated at the same time, however, their priority are DMA channel 0 > DMA channel 1 > DMA channel 2 > DMA channel 3.

If a DMA request and an interrupt request are generated at the same time, the DMA transfer takes precedence, and then interrupt servicing is executed.

(2) DMA response time

The response time of DMA transfer is as follows.

Table 14-2. Response Time of DMA Transfer

	Minimum Time	Maximum Time
Response time	3 clocks	10 clocks[Note]

Note The maximum time necessary to execute an instruction from internal RAM is 16 clock cycles.

Cautions 1. The above response time does not include the two clock cycles required for a DMA transfer.

2. When executing a DMA pending instruction (see 14.6 (4)), the maximum response time is extended by the execution time of that instruction to be held pending.

3. Do not specify successive transfer triggers for a channel within a period equal to the maximum response time plus one clock cycle, because they might be ignored.

Remark 1 clock: $1/f_{CLK}$ (f_{CLK}: CPU clock)

(3) Operation in standby mode

The DMA controller operates as follows in the standby mode.

Table 14-3. DMA Operation in Standby Mode

Status	DMA Operation
HALT mode	Normal operation
STOP mode	Stops operation. If DMA transfer and STOP instruction execution contend, DMA transfer may be damaged. Therefore, stop DMA before executing the STOP instruction.

Downloaded from Arrow.com.

<R> **(4) DMA pending forwarding**

Even if a DMA request is generated, DMA transfer is held pending immediately after the following instructions.

- CALL !addr16
- CALL $!addr20
- CALL !!addr20
- CALL rp
- CALLT [addr5]
- BRK
- MOV PSW, #byte
- MOV PSW, A
- MOV1 PSW. bit, CY
- SET1 PSW. bit
- CLR1 PSW. bit
- POP PSW
- BTCLR PSW. bit, $addr20
- EI
- DI
- Write instructions for registers IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, IF3L, MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, MK3L, PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L, and PSW each.
- Instruction for accessing the data flash memory

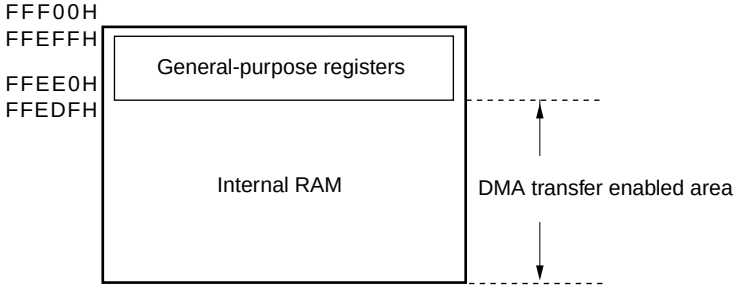
(5) Operation if address in general-purpose register area or other than those of internal RAM area is specified

The address indicated by DMA RAM address register n (DRAn) is incremented during DMA transfer. If the address is incremented to an address in the general-purpose register area or exceeds the area of the internal RAM, the following operation is performed.

- In mode of transfer from SFR to RAM
  The data of that address is lost.

- In mode of transfer from RAM to SFR
  Undefined data is transferred to SFR.

In either case, malfunctioning may occur or damage may be done to the system. Therefore, make sure that the address is within the internal RAM area other than the general-purpose register area.



Downloaded from Arrow.com.

<R> **(6) Operation if instructions for accessing the data flash area**

If the data flash area is accessed after the next instruction execution from start of DMA transfer, a 3-clock wait will be inserted to the next instruction.

Instruction 1
DMA transfer
Instruction 2 ◄– 3-clock wait is inserted.
MOV A, ! DataFlash area

Downloaded from Arrow.com.

CHAPTER 15 INTERRUPT FUNCTIONS

The interrupt function switches the program execution to other processing. When the branch processing is finished, the program returns to the interrupted processing.

The number of interrupt sources differs, depending on the product.

		R5F1026x products R5F1027x products	R5F1036x products R5F1037x products	R5F102Ax products	R5F103Ax products
Maskable interrupts	External	5		6	
	Internal	18	16	26	19

Remark x = 6, 7, 8, 9, A

15.1 Interrupt Function Types

The following two types of interrupt functions are used.

(1) Maskable interrupts

These interrupts undergo mask control. Maskable interrupts can be divided into four priority groups by setting the priority specification flag registers (PR00L, PR00H, PR01L, PR10L, PR10H, PR11L).

Multiple interrupt servicing can be applied to low-priority interrupts when high-priority interrupts are generated. If two or more interrupt requests, each having the same priority, are simultaneously generated, then they are processed according to the default priority of vectored interrupt servicing. For the default priority, see **Tables 15-1** and **15-2**.

A standby release signal is generated and STOP, HALT, and SNOOZE modes are released.

External interrupt requests and internal interrupt requests are provided as maskable interrupts.

(2) Software interrupt

This is a vectored interrupt generated by executing the BRK instruction. It is acknowledged even when interrupts are disabled. The software interrupt does not undergo interrupt priority control.

15.2 Interrupt Sources and Configuration

Interrupt sources include maskable interrupts and software interrupts. In addition, they also have up to seven reset sources (see **Tables 15-1** and **15-2**). The vector codes that store the program start address when branching due to the generation of a reset or various interrupt requests are two bytes each, so interrupts jump to a 64 K address of 00000H to 0FFFFH.

Downloaded from Arrow.com.

Table 15-1. Interrupt Source List (20-, 24-pin products) (1/2)

Interrupt Type	Default Priority Note 1	Interrupt Source		Internal/External	Vector Table Address	Basic Configuration Type Note 2	R5F1026x, R5F1027x products	R5F1036x, R5F1037x products
		Name	Trigger					
Maskable	0	INTWDTI	Watchdog timer interval Note 3 (75%+1/2f_{IL} of overflow time)	Internal	0004H	(A)	✓	✓
	1	INTLVI	Voltage detection Note 4		0006H		✓	✓
	2	INTP0	Pin input edge detection	External	0008H	(B)	✓	✓
	3	INTP1			000AH		✓	✓
	4	INTP2			000CH		✓	✓
	5	INTP3			000EH		✓	✓
	6	INTDMA0	End of DMA0 transfer	Internal	0010H	(A)	✓	–
	7	INTDMA0	End of DMA1 transfer		0012H		✓	–
	8	INTST0/ INTCSI00/ INTIIC00	UART0 transmission transfer end or buffer empty interrupt/CSI00 transfer end or buffer empty interrupt/IIC00 transfer end		0014H		✓	✓ ✓ –
	9	INTSR0/ INTCSI01/ INTIIC01	UART0 reception transfer end/CSI01 transfer end or buffer empty interrupt/IIC01 transfer end		0016H		✓	✓ – –
	10	INTSRE0	UART0 reception communication error occurrence		0018H		✓	✓
	11	INTTM01H	End of timer channel 1 count or capture (at higher 8-bit timer operation)		001AH		✓	✓
	12	INTTM03H	End of timer channel 3 count or capture (at higher 8-bit timer operation)		001CH		✓	✓
	13	INTIICA0	End of IICA0 communication		001EH		✓	✓
	14	INTTM00	End of timer channel 0 count or capture (16 bit/at lower 8-bit timer operation)		0020H		✓	✓
	15	INTTM01	End of timer channel 1 count or capture (16 bit/at lower 8-bit timer operation)		0022H		✓	✓
	16	INTTM02	End of timer channel 2 count or capture (16 bit/at lower 8-bit timer operation)		0024H		✓	✓
	17	INTTM03	End of timer channel 3 count or capture (16 bit/at lower 8-bit timer operation)		0026H		✓	✓
	18	INTAD	End of A/D conversion		0028H		✓	✓
	19	INTIT	Interval signal detection from 12-bit Interval timer		002AH		✓	✓
	20	INTKR	Key return signal detection	External	002CH	(C)	✓	✓
	21	INTMD	End of division operation/ Overflow of multiply-accumulation result occurs	Internal	002EH	(A)	✓	✓
	22	INTFL	Reserved Note 5		0030H		✓	✓

Notes 1. The default priority determines the sequence of interrupts if two or more maskable interrupts occur simultaneously. 0 indicates the highest priority and 22 indicates the lowest priority.

2. Basic configuration types (A) to (D) correspond to (A) to (D) in Figure 15-1.

3. When bit 7 (WDTINT) of the option byte (000C0H) is set to 1.

4. When bit 7 (LVIMD) of the voltage detection level register (LVIS) is cleared to 0.

5. Used for flash self programming library and data flash library.

Downloaded from Arrow.com.

Table 15-1. Interrupt Source List (20-, 24-pin products) (2/2)

Interrupt Type	Default Priority Note 1	Interrupt Source		Internal/External	Vector Table Address	Basic Configuration Type Note 2	R5F1026x, R5F1027x products	R5F1036x, R5F1037x products
		Name	Trigger					
Software	–	BRK	Execution of BRK instruction	–	007EH	(D)	✓	✓
Reset	–	RESET	$\overline{\text{RESET}}$ pin input	–	0000H	–	✓	✓
		POR	Power-on-reset				✓	✓
		LVD	Voltage detection Note 3				✓	✓
		WDT	Overflow of watchdog timer				✓	✓
		TRAP	Execution of illegal instruction Note 4				✓	✓
		IAW	Illegal memory access				✓	✓
		RPE	RAM parity error				✓	✓

Notes 1. The default priority determines the sequence of interrupts if two or more maskable interrupts occur simultaneously. Zero indicates the highest priority and 22 indicates the lowest priority.

2. Basic configuration types (A) to (D) correspond to (A) to (D) in Figure 15-1.

3. When bit 7 (LVIMD) of the voltage detection level register (LVIS) is cleared to 1.

4. When the instruction code in FFH is executed.
No reset is issued even if an illegal instruction is executed during emulation with the in-circuit emulator or on-chip debug emulator.

Downloaded from Arrow.com.

Table 15-2. Interrupt Source List (30-pin products) (1/2)

Interrupt Type	Default Priority Note 1	Interrupt Source		Internal/External	Vector Table Address	Basic Configuration Type Note 2	R5F102Ax products	R5F103Ax products
		Name	Trigger					
Maskable	0	INTWDTI	Watchdog timer interval Note 3 (75% + 1/2f_{IL} of overflow time)	Internal	0004H	(A)	✓	✓
	1	INTLVI	Voltage detection Note 4		0006H		✓	✓
	2	INTP0	Pin input edge detection	External	0008H	(B)	✓	✓
	3	INTP1			000AH		✓	✓
	4	INTP2			000CH		✓	✓
	5	INTP3			000EH		✓	✓
	6	INTP4			0010H		✓	✓
	7	INTP5			0012H		✓	✓
	8	INTST2/ INTCSI20/ INTIIC20	UART2 transmission transfer end, buffer empty interrupt/CSI20 transfer end or buffer empty interrupt/IIC20 transfer end	Internal	0014H	(A)	✓	–
	9	INTSR2	UART2 reception transfer end		0016H		✓	–
	10	INTSRE2	UART2 reception communication error occurrence		0018H		✓	–
	11	INTDMA0	DMA0 transfer end		001AH		✓	–
	12	INTDMA1	DMA1 transfer end		001CH		✓	–
	13	INTST0/ INTCSI00/ INTIIC00	UART0 transmission transfer end, buffer empty interrupt/CSI00 transfer end or buffer empty interrupt/IIC00 transfer end		001EH		✓	✓
								✓
								–
	14	INTSR0	UART0 reception transfer end		0020H		✓	✓
	15	INTSRE0	UART0 reception communication error occurrence		0022H		✓	✓
		INTTM01H	End of timer channel 1 count or capture (at higher 8-bit timer operation)				✓	✓
	16	INTST1	UART1 transmission transfer end		0024H		✓	–
	17	INTSR1/ INTCSI11/ INTIIC11	UART1 reception transfer end/CSI11 transfer end or buffer empty interrupt/IIC11 transfer end		0026H		✓	–
	18	INTSRE1	UART1 reception communication error occurrence		0028H		✓	–
		INTTM03H	End of timer channel 3 count or capture (at higher 8-bit timer operation)				✓	✓
	19	INTIICA0	IICA0 communication end		002AH		✓	✓

Notes 1. The default priority determines the sequence of interrupts if two or more maskable interrupts occur simultaneously. 0 indicates the highest priority and 31 indicates the lowest priority.

2. Basic configuration types (A) to (D) correspond to (A) to (D) in Figure 15-1 respectively.

3. When bit 7 (WDTINT) of the option byte (000C0H) is set to 1.

4. When bit 7 (LVIMD) of the voltage detection level register (LVIS) is cleared to 0.

Downloaded from Arrow.com.

Table 15-2. Interrupt Source List (30-pin products) (2/2)

Interrupt Type	Default Priority Note 1	Interrupt Source		Internal/External	Vector Table Address	Basic Configuration Type Note 2	R5F102Ax products	R5F103Ax products
		Name	Trigger					
Maskable	20	INTTM00	End of timer channel 0 count or capture (16 bit / at lower 8-bit timer operation)	Internal	002CH	(A)	✓	✓
	21	INTTM01	End of timer channel 1 count or capture (16 bit / at lower 8-bit timer operation)		002EH		✓	✓
	22	INTTM02	End of timer channel 2 count or capture (16 bit / at lower 8-bit timer operation)		0030H		✓	✓
	23	INTTM03	End of timer channel 3 count or capture (16 bit / at lower 8-bit timer operation)		0032H		✓	✓
	24	INTAD	End of A/D conversion		0034H		✓	✓
	25	INTIT	Interval signal detection from 12-bit interval timer		0038H		✓	✓
	26	INTTM04	End of timer channel 4 count or capture (16 bit / at lower 8-bit timer operation)		0042H		✓	✓
	27	INTTM05	End of timer channel 5 count or capture (16 bit / at lower 8-bit timer operation)		0044H		✓	✓
	28	INTTM06	End of timer channel 6 count or capture (16 bit / at lower 8-bit timer operation)		0046H		✓	✓
	29	INTTM07	End of timer channel 7 count or capture (16 bit / at lower 8-bit timer operation)		0048H		✓	✓
	30	INTMD	End of division operation/ Overflow of multiply-accumulation result occurs		005EH		✓	✓
	31	INTFL	Reserved Note 3		0062H		✓	✓
Software	–	BRK	Execution of BRK instruction	–	007EH	(D)	✓	✓
Reset	–	RESET	$\overline{\text{RESET}}$ pin input	–	0000H	–	✓	✓
		POR	Power-on-reset				✓	✓
		LVD	Voltage detection Note 4				✓	✓
		WDT	Overflow of watchdog timer				✓	✓
		TRAP	Execution of illegal instruction Note 5				✓	✓
		IAW	Illegal memory access				✓	✓
		RPE	RAM parity error				✓	✓

Notes 1. The default priority determines the sequence of interrupts if two or more maskable interrupts occur simultaneously. 0 indicates the highest priority and 31 indicates the lowest priority.

2. Basic configuration types (A) to (D) correspond to (A) to (D) in Figure 15-1 respectively.

3. Used for flash self programming library and data flash library.

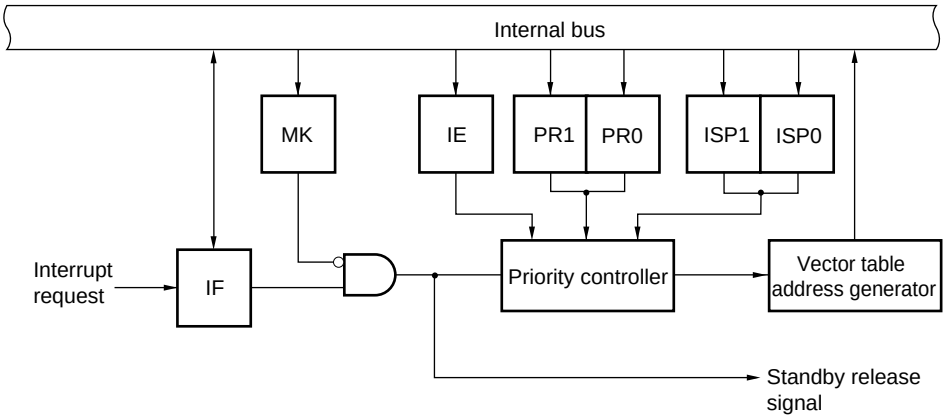
4. When bit 7 (LVIMD) of the voltage detection level register (LVIS) is cleared to 0.

5. When the instruction code in FFH is executed.
No reset is issued even if an illegal instruction is executed during emulation with the in-circuit emulator or on-chip debug emulator.

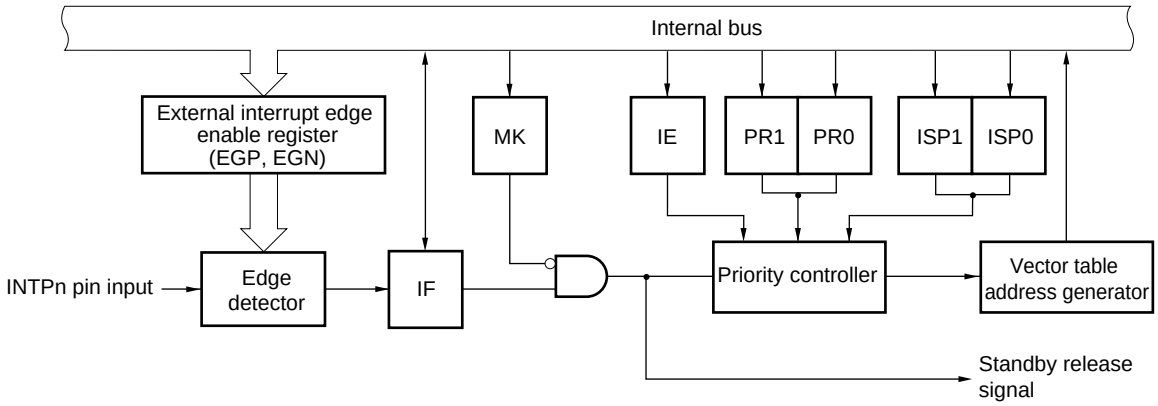
Downloaded from Arrow.com.

Figure 15-1. Basic Configuration of Interrupt Function (1/2)

(a) Internal maskable interrupt



(b) External maskable interrupt (INTPn)



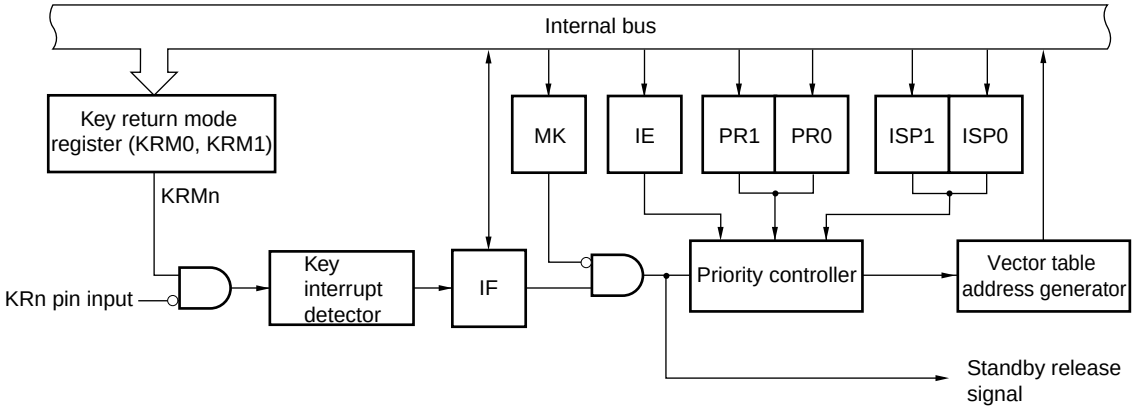
IF: Interrupt request flag
IE: Interrupt enable flag
ISP0: In-service priority flag 0
ISP1: In-service priority flag 1
MK: Interrupt mask flag
PR0: Priority specification flag 0
PR1: Priority specification flag 1

Remark 20, 24-pin product : n = 0 to 3
30-pin product : n = 0 to 5

Downloaded from Arrow.com.

Figure 15-1. Basic Configuration of Interrupt Function (2/2)

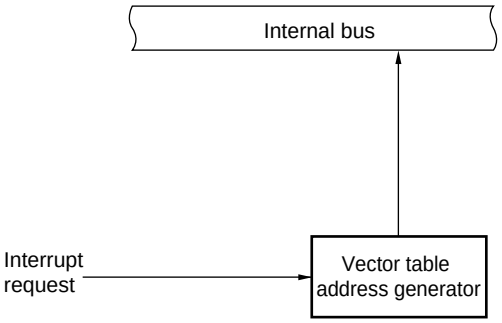
(c) External maskable interrupt (INTKR)



IF: Interrupt request flag
IE: Interrupt enable flag
ISP0: In-service priority flag 0
ISP1: In-service priority flag 1
MK: Interrupt mask flag
PR0: Priority specification flag 0
PR1: Priority specification flag 1

Remark 24-pin product : n = 0 to 9
20-pin product : n = 0 to 5

(d) Software interrupt



Downloaded from Arrow.com.

15.3 Registers Controlling Interrupt Functions

The following 6 types of registers are used to control the interrupt functions.

- Interrupt request flag registers (IF0L, IF0H, IF1L, IF1H, IF2L , IF2H)
- Interrupt mask flag registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H)
- Priority specification flag registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H)
- External interrupt rising edge enable register (EGP0)
- External interrupt falling edge enable register (EGN0)
- Program status word (PSW)

Tables 15-3 and 15-4 show a list of interrupt request flags, interrupt mask flags, and priority specification flags corresponding to interrupt request sources.

Table 15-3. Flags Corresponding to Interrupt Request Sources (20-, 24-pin products) (1/2)

Interrupt Source	Interrupt Request Flag	Register	Interrupt Mask Flag	Register	Priority Specification Flag	Register	R5F1026x, R5F1027x products	R5F1036x, R5F1037x products
INTWDTI	WDTIIF	IF0L	WDTIMK	MK0L	WDTIPR0, WDTIPR1	PR00L, PR10L	✓	✓
INTLVI	LVIIF		LVIMK		LVIPR0, LVIPR1		✓	✓
INTP0	PIF0		PMK0		PPR00, PPR10		✓	✓
INTP1	PIF1		PMK1		PPR01, PPR11		✓	✓
INTP2	PIF2		PMK2		PPR02, PPR12		✓	✓
INTP3	PIF3		PMK3		PPR03, PPR13		✓	✓
INTDMA0	DMAIF0		DMAMK0		DMAPR00, DMAPR10		✓	–
INTDMA1	DMAIF1		DMAMK1		DMAPR01, DMAPR11		✓	–
INTST0 Note 1	STIF0 Note 1	IF0H	STMK0 Note 1	MK0H	STPR00, STPR10 Note 1	PR00H, PR10H	✓	✓
INTCSI00 Note 1	CSIIF00 Note 1		CSIMK00 Note 1		CSIPR000, CSIPR100 Note 1		✓	✓
INTIIC00 Note 1	IICIF00 Note 1		IICMK00 Note 1		IICPR000, IICPR100 Note 1		✓	–
INTSR0 Note 2	SRIF0 Note 2		SRMK0 Note 2		SRPR00, SRPR10 Note 2		✓	✓
INTCSI01 Note 2	CSIIF01 Note 2		CSIMK01 Note 2		CSIPR001, CSIPR101 Note 2		✓	–
INTIIC01 Note 2	IICIF01 Note 2		IICMK01 Note 2		IICPR001, IICPR101 Note 2		✓	–
INTSRE0	SREIF0		SREMK0		SREPR00, SREPR10		✓	✓
INTTM01H	TMIF01H		TMMK01H		TMPR001H, TMPR101H		✓	✓
INTTM03H	TMIF03H		TMMK03H		TMPR003H, TMPR103H		✓	✓
INTIICA0	IICAIF0		IICAMK0		IICAPR00, IICAPR10		✓	✓
INTTM00	TMIF00		TMMK00		TMPR000, TMPR100		✓	✓
INTTM01	TMIF01		TMMK01		TMPR001, TMPR101		✓	✓
INTTM02	TMIF02	IF1L	TMMK02	MK1L	TMPR002, TMPR102	PR01L, PR11L	✓	✓
INTTM03	TMIF03		TMMK03		TMPR003, TMPR103		✓	✓

Downloaded from Arrow.com.

Table 15-3. Flags Corresponding to Interrupt Request Sources (20-, 24-pin products) (2/2)

Interrupt Source	Interrupt Request Flag	Register	Interrupt Mask Flag	Register	Priority Specification Flag	Register	R5F1026x, R5F1027x products	R5F1036x, R5F1037x products
INTAD	ADIF	IF1L	ADMK	MK1L	ADPR0, ADPR1	PR01L, PR11L	✓	✓
INTIT	TMKAIF		TMKAMK		TMKAPR0, TMKAPR1		✓	✓
INTKR	KRIF		KRMK		KRPR0, KRPR1		✓	✓
INTMD	MDIF		MDMK		MDPR0, MDPR1		✓	✓
INTFL	FLIF		FLMK		FLPR0, FLPR1		✓	✓

Notes 1. If interrupt source INTST0, INTCSI00, or INTIIC00 occurs, bit 5 of the IF0H register is set to 1. Bit 5 of the MK0H, PR00H, and PR10H registers corresponds these three interrupt sources.

2. If interrupt source INTSR0, INTCSI01, or INTIIC01 occurs, bit 6 of the IF0H register is set to 1. Bit 6 of the MK0H, PR00H, and PR10H registers corresponds these three interrupt sources.

Downloaded from Arrow.com.

Table 15-4. Flags Corresponding to Interrupt Request Sources (30-pin products) (1/2)

Interrupt Source	Interrupt Request Flag	Register	Interrupt Mask Flag	Register	Priority Specification Flag	Register	R5F102Ax products	R5F103Ax products
INTWDTI	WDTIIF	IF0L	WDTIMK	MK0L	WDTIPR0, WDTIPR1	PR00L, PR10L	✓	✓
INTLVI	LVIIF		LVIMK		LVIPR0, LVIPR1		✓	✓
INTP0	PIF0		PMK0		PPR00, PPR10		✓	✓
INTP1	PIF1		PMK1		PPR01, PPR11		✓	✓
INTP2	PIF2		PMK2		PPR02, PPR12		✓	✓
INTP3	PIF3		PMK3		PPR03, PPR13		✓	✓
INTP4	PIF4		PMK4		PPR04, PPR14		✓	✓
INTP5	PIF5		PMK5		PPR05, PPR15		✓	✓
INTST2[Note1] INTCSI20[Note1] INTIIC20[Note1]	STIF2[Note1] CSIIF20[Note1] IICIF20[Note1]	IF0H	STMK2[Note1] CSIMK20[Note1] IICMK20[Note1]	MK0H	STPR02, STPR12[Note1] CSIPR020, CSIPR120[Note1] IICPR020, IICPR120[Note1]	PR00H, PR10H	✓	–
INTSR2	SRIF2		SRMK2		SRPR02, SRPR12		✓	–
INTSRE2	SREIF2		SREMK2		SREPR02, SREPR12		✓	–
INTDMA0	DMAIF0		DMAMK0		DMAPR00, DMAPR10		✓	–
INTDMA1	DMAIF1		DMAMK1		DMAPR01, DMAPR11		✓	–
INTST0[Note2] INTCSI00[Note2] INTIIC00[Note2]	STIF0[Note2] CSIIF00[Note2] IICIF00[Note2]		STMK0[Note2] CSIMK00[Note2] IICMK00[Note2]		STPR00, STPR10[Note2] CSIPR000, CSIPR100[Note2] IICPR000, IICPR100[Note2]		✓	✓
INTSR0	SRIF0		SRMK0		SRPR00, SRPR10		✓	✓
INTSRE0[Note3] INTTM01H[Note3]	SREIF0[Note3] TMIF01H[Note3]		SREMK0[Note3] TMMK01H[Note3]		SREPR00, SREPR10[Note3] TMPR001H, TMPR101H[Note3]		✓	✓
INTST1	STIF1	IF1L	STMK1	MK1L	STPR01, STPR11	PR01L, PR11L	✓	–
INTSR1[Note4] INTCSI11[Note4] INTIIC11[Note4]	SRIF1[Note4] CSIIF11[Note4] IICIF11[Note4]		SRMK1[Note4] CSIMK11[Note4] IICMK11[Note4]		SRPR01, SRPR11[Note4] CSIPR011, CSIPR111[Note4] IICPR011, IICPR111[Note4]		✓	–
INTSRE1[Note5]	SREIF1[Note5]		SREMK1[Note5]		SREPR01, SREPR11[Note5]		✓	–
INTTM03H[Note5]	TMIF03H[Note5]		TMMK03H[Note5]		TMPR003H, TMPR103H[Note5]			✓
INTIICA0	IICAIF0		IICAMK0		IICAPR00, IICAPR10		✓	✓
INTTM00	TMIF00		TMMK00		TMPR000, TMPR100		✓	✓
INTTM01	TMIF01		TMMK01		TMPR001, TMPR101		✓	✓
INTTM02	TMIF02		TMMK02		TMPR002, TMPR102		✓	✓
INTTM03	TMIF03		TMMK03		TMPR003, TMPR103		✓	✓
INTAD	ADIF	IF1H	ADMK	MK1H	ADPR0, ADPR1	PR01H, PR11H	✓	✓
INTIT	TMKAIF		TMKAMK		TMKAPR0, TMKAPR1		✓	✓
INTTM04	TMIF04		TMMK04		TMPR004, TMPR104		✓	✓

Downloaded from Arrow.com.

Table 15-4. Flags Corresponding to Interrupt Request Sources (30-pin products) (2/2)

Interrupt Source	Interrupt Request Flag	Register	Interrupt Mask Flag	Register	Priority Specification Flag	Register	R5F102Ax products	R5F103Ax products
INTTM05	TMIF05	IF2L	TMMK05	MK2L	TMPR005, TMPR105	PR02L, PR12L	✓	✓
INTTM06	TMIF06		TMMK06		TMPR006, TMPR106		✓	✓
INTTM07	TMIF07		TMMK07		TMPR007, TMPR107		✓	✓
INTMD	MDIF	IF2H	MDMK	MK2H	MDPR0, MDPR1	PR02H, PR12H	✓	✓
INTFL	FLIF		FLMK		FLPR0, FLPR1		✓	✓

Notes 1. If interrupt source INTST2, INTCSI20, or INTIIC20 occurs, bit 0 of the IF0H register is set to 1. In addition, bit 0 of the MK0H, PR00H, and PR10H registers corresponds to these three interrupt sources.

2. If interrupt source INTST0, INTCSI00, or INTIIC00 occurs, bit 5 of the IF0H register is set to 1. In addition, bit 5 of the MK0H, PR00H, and PR10H registers corresponds to these three interrupt sources.

3. Do not use a UART0 reception error interrupt and an interrupt of channel 1 of TAU0 (at higher 8-bit timer operation) at the same time because they share flags for the interrupt request sources. If the UART0 reception error interrupt is not used (EOC01 = 0), UART0 and channel 1 of TAU0 (at higher 8-bit timer operation) can be used at the same time. If one of the interrupt sources INTSRE0 and INTTM01H is generated, bit 7 of the IF0H register is set to 1. Bit 7 of the MK0H, PR00H, and PR10H registers supports these two interrupt sources.

4. If interrupt source INTST1, INTCSI11, or INTIIC11 occurs, bit 1 of the IF1L register is set to 1. Bit 1 of the MK1L, PR01L, and PR11L registers corresponds to these three interrupt sources.

5. Do not use a UART1 reception error interrupt and an interrupt of channel 3 of TAU0 (at higher 8-bit timer operation) at the same time because they share flags for the interrupt request sources. If the UART1 reception error interrupt is not used (EOC01 = 0), UART1 and channel 3 of TAU0 (at higher 8-bit timer operation) can be used at the same time. If interrupt source INTTM03H is generated, bit 2 of the IF1L register is set to 1. Bit 2 of the MK1L, PR01L, and PR11L registers supports these two interrupt sources.

Downloaded from Arrow.com.

15.3.1 Interrupt request flag registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H)

The interrupt request flags are set to 1 when the corresponding interrupt request is generated or an instruction is executed. They are cleared to 0 when the interrupt request is acknowledged, a reset signal is generated, or an instruction is executed.

When an interrupt is acknowledged, the interrupt request flag is automatically cleared and then the interrupt routine is entered.

The IF0L, IF0H, IF1L, IF1H, IF2L, and IF2H registers can be set by a 1-bit or 8-bit memory manipulation instruction. When the IF0L and IF0H registers, the IF1L and IF1H registers, and the IF2L and IF2H registers are combined to form 16-bit registers IF0, IF1, and IF2, they can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Remark If an instruction that writes data to this register is executed, the number of instruction execution clocks increases by 2 clocks.

Figure 15-2. Format of Interrupt Request Flag Registers (IF0L, IF0H, IF1L) (20-, 24-pin product)

Address: FFFE0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF0L	DMAIF1[Note]	DMAIF0[Note]	PIF3	PIF2	PIF1	PIF0	LVIIF	WDTIIF

Address: FFFE1H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF0H	TMIF01	TMIF00	IICAIF0	DMAIF1	MIF01H	SREIF0	SRIF0 CSIIF01[Note] IICIF01[Note]	STIF0 CSIIF00 IICIF00[Note]

Address: FFFE2H After reset: 00H R/W

Symbol	7	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF1L	0	FLIF	MDIF	KRIF	TMKAIF	ADIF	TMIF03	TMIF02

XXIFXX	Interrupt request flag
0	No interrupt request signal is generated
1	Interrupt request is generated, interrupt request status

Note Provided in the R5F102 products only.

(**Cautions** are listed on the next page)

Downloaded from Arrow.com.

Figure 15-3. Format of Interrupt Request Flag Registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H) (30-pin product)

Address: FFFE0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF0L	PIF5	PIF4	PIF3	PIF2	PIF1	PIF0	LVIIF	WDTIIF

Address: FFFE1H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF0H	SREIF0 TMIF01H	SRIF0	STIF0 CSIIF00 IICIF00Note	DMAIF1Note	DMAIF0Note	SREIF2Note	SRIF2Note	STIF2Note CSIIF20Note IICIF20Note

Address: FFFE2H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF1L	TMIF03	TMIF02	TMIF01	TMIF00	IICAIF0	SREIF1Note TMIF03H	SRIF1Note CSIIF11Note IICIF11Note	STIF1Note

Address: FFFE3H After reset: 00H R/W

Symbol	<7>	6	5	4	3	<2>	1	<0>
IF1H	TMIF04	0	0	0	0	TMKAIF	0	ADIF

Address: FFFD0H After reset: 00H R/W

Symbol	7	6	5	4	3	<2>	<1>	<0>
IF2L	0	0	0	0	0	TMIF07	TMIF06	TMIF05

Address: FFFD1H After reset: 00H R/W

Symbol	7	6	<5>	4	3	2	1	0
IF2H	FLIF	0	MDIF	0	0	0	0	0

XXIFXX	Interrupt request flag
0	No interrupt request signal is generated
1	Interrupt request is generated, interrupt request status

Note Provided in the R5F102 products only.

Cautions 1. Be sure to set bits that are not available to the initial value.

2. When manipulating a flag of the interrupt request flag register, use a 1-bit memory manipulation instruction (CLR1). When describing in C language, use a bit manipulation instruction such as IF0L.0 = 0; or _asm("clr1 IF0L, 0"); because the compiled assembler must be a 1-bit memory manipulation instruction (CLR1). If a program is described in C language using an 8-bit memory manipulation instruction such as IF0L &= 0xfe; and compiled, it becomes the assembler of three instructions.

```
mov a, IF0L
and a, #0FEH
mov IF0L, a
```


Downloaded from Arrow.com.

In this case, even if the request flag of the another bit of the same interrupt request flag register (IF0L) is set to 1 at the timing between mov a, IF0L and mov IF0L, a, the flag is cleared to 0 at mov IF0L, a. Therefore, care must be exercised when using an 8-bit memory manipulation instruction in C language.

15.3.2 Interrupt mask flag registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H)

The interrupt mask flags are used to enable/disable the corresponding maskable interrupt.

The MK0L, MK0H, MK1L, MK1H, MK2L, and MK2H registers can be set by a 1-bit or 8-bit memory manipulation instruction. When the MK0L and MK0H registers, the MK1L and MK1H registers, and the MK2L and MK2H registers are combined to form 16-bit registers MK0, MK1, and MK2, they can be set by a 16-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Remark If an instruction that writes data to this register is executed, the number of instruction execution clocks increases by 2 clocks.

Figure 15-4. Format of Interrupt Mask Flag Registers (MK0L, MK0H, MK1L) (20, 24-pin product)

Address: FFFE4H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK0L	DMAMK1[Note]	DMAMK0[Note]	PMK3	PMK2	PMK1	PMK0	LVIMK	WDTIMK

Address: FFFE5H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK0H	TMMK01	TMMK00	CAMK0	TMMK03H	TMMK03H	SREMK0	RMK0 CSIMK01[Note] IICMK01[Note]	TMK0 CSIMK00 IICMK00[Note]

Address: FFFE6H After reset: FFH R/W

Symbol	7	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK1L	1	FLMK	MDMK	KRMK	TMKAMK	ADMK	TMMK03	TMMK02

XXMKXX	Interrupt servicing control
0	Interrupt servicing enabled
1	Interrupt servicing disabled

Note Provided in the R5F102 products only.

Caution Be sure to set bits that are not available to the initial value.

Downloaded from Arrow.com.

Figure 15-5. Format of Interrupt Mask Flag Registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H) (30-pin product)

Address: FFFE4H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK0L	PMK5	PMK4	PMK3	PMK2	PMK1	PMK0	LVIMK	WDTIMK

Address: FFFE5H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK0H	SREMK0 TMMK01H	SRMK0	STMK0 CSIMK01 IICMK01Note	DMAMK1Note	DMAMK0Note	SREMK2Note	SRMK2Note	STMK2Note CSIMK20Note IICMK20Note

Address: FFFE6H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK1L	TMMK03	TMMK02	TMMK01	TMMK00	IICAMK0	SREMK1Note TMMK03H	SRMK1Note CSIMK11Note IICMK11Note	STMK1Note

Address: FFFE7H After reset: FFH R/W

Symbol	<7>	6	5	4	3	<2>	1	<0>
MK1H	TMMK04	1	1	1	1	TMKAMK	1	ADMK

Address: FFFD4H After reset: FFH R/W

Symbol	7	6	5	4	3	<2>	<1>	<0>
MK2L	1	1	1	1	1	TMMK07	TMMK06	TMMK05

Address: FFFD5H After reset: FFH R/W

Symbol	7	6	<5>	4	3	2	1	0
MK2H	FLMK	1	MDMK	1	1	1	1	1

XXMKXX	Interrupt servicing control
0	Interrupt servicing enabled
1	Interrupt servicing disabled

Note Provided in the R5F102 products only.

Caution Be sure to set bits that are not available to the initial value.

Downloaded from Arrow.com.

15.3.3 Priority specification flag registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H)

The priority specification flag registers are used to set the priority level of the corresponding maskable interrupt.

A priority level is set by using the PR0xy and PR1xy registers in combination (xy = 0L, 0H, 1L, 1H, 2L, or 2H).

The PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, and PR12H L registers can be set by a 1-bit or 8-bit memory manipulation instruction. If the PR00L and PR00H registers, the PR01L and PR01H registers, the PR02L and PR02H registers, the PR10L and PR10H registers, the PR11L and PR11H registers, and the PR12L and PR12H registers are combined to form 16-bit registers PR00 and PR10, they can be set by a 16-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Remark If an instruction that writes data to this register is executed, the number of instruction execution clocks increases by 2 clocks.

Downloaded from Arrow.com.

Figure 15-6. Format of Priority Specification Flag Registers (PR00L, PR00H, PR01L, PR10L, PR10H, PR11L) (20-, 24-pin product)

Address: FFFE8H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR00L	DMAPR01[Note]	DMAPR00[Note]	PPR03	PPR02	PPR01	PPR00	LVIPR0	WDTIPR0

Address: FFFECH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR10L	DMAPR11	DMAPR10	PPR13	PPR12	PPR11	PPR10	LVIPR1	WDTIPR1

Address: FFFE9H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR00H	TMPR001	TMPR000	IICAPR00	TMPR003H	TMPR001H	SREPR00	SRPR00 CSIPR001[Note] IICPR001[Note]	STPR00 CSIPR000 IICPR000[Note]

Address: FFFEDH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR10H	TMPR101	TMPR100	IICAPR10	TMPR103H	TMPR101H	SREPR10	SRPR10 CSIPR101 IICPR101	STPR10 CSIPR100 IICPR100

Address: FFFEAH After reset: FFH R/W

Symbol	7	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR01L	1	FLPR0	MDPR0	KRPR0	TMKAPR0	ADPR0	TMPR003	TMPR002

Address: FFFEEH After reset: FFH R/W

Symbol	7	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR11L	1	FLPR1	MDPR1	KRPR1	TMKAPR1	ADPR1	TMPR001	TMPR102

XXPR1X	XXPR0X	Priority Level Selection
0	0	Specifying level 0 (high priority)
0	1	Specifying level 1
1	0	Specifying level 2
1	1	Specifying level 3 (low priority)

Note Provided in the R5F102 products only.

Caution Be sure to set bits that are not available to the initial value.

Downloaded from Arrow.com.

Figure 15-7. Format of Priority Specification Flag Registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H) (30-pin product) (1/2)

Address: FFFE8H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR00L	PPR05	PPR04	PPR03	PPR02	PPR01	PPR00	LVIPR0	WDTIPR0

Address: FFFECH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR10L	PPR15	PPR14	PPR13	PPR12	PPR11	PPR10	LVIPR1	WDTIPR1

Address: FFFE9H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR00H	SREPR00 TMPR001H	SRPR00	STPR00 CSIPR000 IICPR000Note	DMAPR01Note	DMAPR00Note	SREPR02Note	SRPR02Note	STPR02Note CSIPR020Note IICPR020Note

Address: FFFEDH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR10H	SREPR10 TMPR101H	SRPR10	STPR10 CSPR100 IICPR100Note	DMAPR11Note	DMAPR10Note	SREPR12Note	SRPR12Note	STPR12Note CSIPR120Note IICPR120Note

Address: FFFEAH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR01L	TMPR003	TMPR002	TMPR001	TMPR000	IICAPR00	SREPR01Note TMPR003H	SRPR01Note CSIPR011Note IICPR011Note	STPR01Note

Address: FFFEEH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR11L	TMPR103	TMPR102	TMPR101	TMPR100	IICAPR10	SREPR11Note TMPR103H	SRPR11Note CSIPR111Note IICPR111Note	STPR11Note

Address: FFFEBH After reset: FFH R/W

Symbol	<7>	6	5	4	3	<2>	1	<0>
PR01H	TMPR004	1	1	1	1	TMKAPR0	1	ADPR0

Address: FFFEFH After reset: FFH R/W

Symbol	<7>	6	5	4	3	<2>	1	<0>
PR11H	TMPR104	1	1	1	1	TMKAPR1	1	ADPR1

Note Provided in the R5F102 products only.

Downloaded from Arrow.com.

Figure 15-7. Format of Priority Specification Flag Registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H) (30-pin product) (2/2)

Address: FFFD8H After reset: FFH R/W

Symbol	7	6	5	4	3	<2>	<1>	<0>
PR02L	1	1	1	1	1	TMPR007	TMPR006	TMPR005

Address: FFFDCH After reset: FFH R/W

Symbol	7	6	5	4	3	<2>	<1>	<0>
PR12L	1	1	1	1	1	TMPR107	TMPR106	TMPR105

Address: FFFD9H After reset: FFH R/W

Symbol	7	6	<5>	4	3	2	1	0
PR02H	FLPR0	1	MDPR0	1	1	1	1	1

Address: FFFDDH After reset: FFH R/W

Symbol	7	6	<5>	4	3	2	1	0
PR12H	FLPR1	1	MDPR1	1	1	1	1	1

XXPR1X	XXPR0X	Priority Level Selection
0	0	Specifying level 0 (high priority)
0	1	Specifying level 1
1	0	Specifying level 2
1	1	Specifying level 3 (low priority)

Caution Be sure to set bits that are not available to the initial value.

Downloaded from Arrow.com.

15.3.4 External interrupt rising edge enable register (EGP0), external interrupt falling edge enable register (EGN0)

These registers specify the valid edge for INTP0 to INTP5.

The EGP0 and EGN0 registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Figure 15-8. Format of External Interrupt Rising Edge Enable Register (EGP0) and External Interrupt Falling Edge Enable Register (EGN0)

20-, 24-pin products

Address: FFF38H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGP0	0	0	0	0	EGP3	EGP2	EGP1	EGP0

Address: FFF39H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGN0	0	0	0	0	EGN3	EGN2	EGN1	EGN0

30-pin products

Address: FFF38H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGP0	0	0	EGP5	EGP4	EGP3	EGP2	EGP1	EGP0

Address: FFF39H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGN0	0	0	EGN5	EGN4	EGN3	EGN2	EGN1	EGN0

EGPn	EGNn	INTPn pin valid edge selection (n = 0 to 5)
0	0	Edge detection disabled
0	1	Falling edge
1	0	Rising edge
1	1	Both rising and falling edges

Caution When the input port pins used for the external interrupt functions are switched to the output mode, the INTPn interrupt might be generated upon detection of a valid edge. When switching the input port pins to the output mode, set the port mode register (PMxx) to 0 after disabling the edge detection (by setting EGPn and EGNn to 0).

Remark n = 0 to 5

Downloaded from Arrow.com.

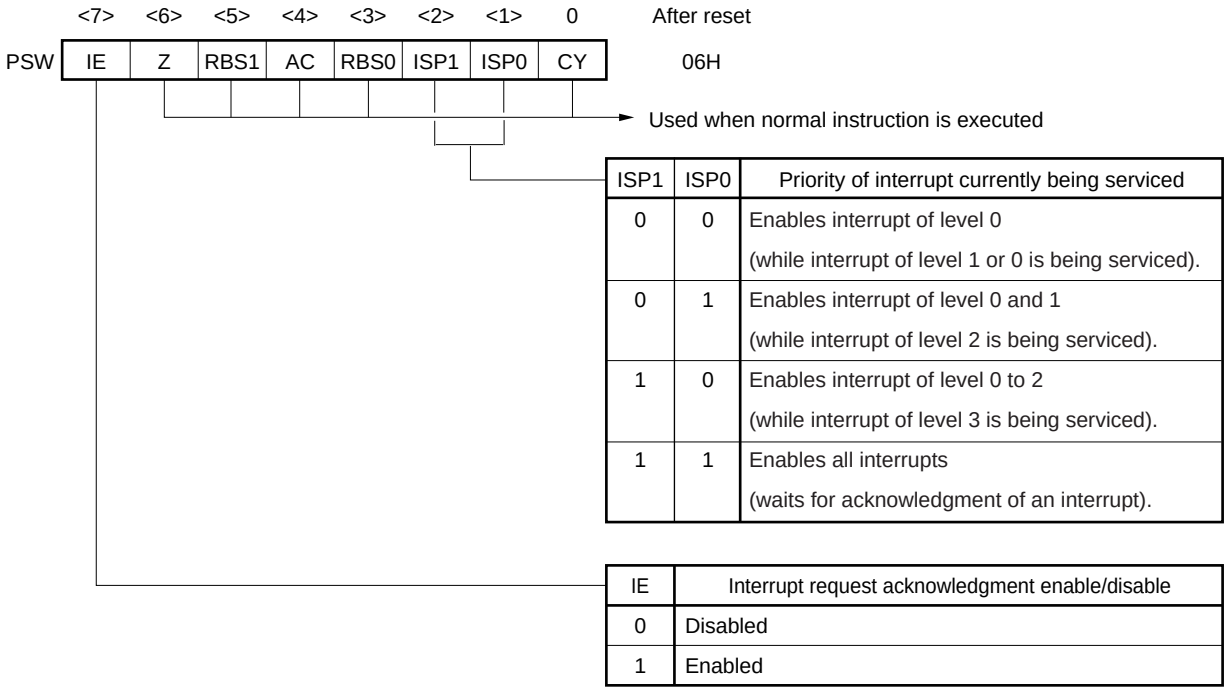
15.3.5 Program status word (PSW)

The program status word is a register used to hold the instruction execution result and the current status for an interrupt request. The IE flag that sets maskable interrupt enable/disable and the ISP0 and ISP1 flags that controls multiple interrupt servicing are mapped to the PSW.

Besides 8-bit read/write, this register can carry out operations using bit manipulation instructions and dedicated instructions (EI and DI). When a vectored interrupt request is acknowledged, if the BRK instruction is executed, the contents of the PSW are automatically saved into a stack and the IE flag is reset to 0. Upon acknowledgment of a maskable interrupt request, if the value of the priority specification flag register of the acknowledged interrupt is not 00, its value minus 1 is transferred to the ISP0 and ISP1 flags. The PSW contents are also saved into the stack with the PUSH PSW instruction. They are restored from the stack with the RETI, RETB, and POP PSW instructions.

Reset signal generation sets PSW to 06H.

Figure 15-9. Configuration of Program Status Word



ISP1	ISP0	Priority of interrupt currently being serviced
0	0	Enables interrupt of level 0 (while interrupt of level 1 or 0 is being serviced).
0	1	Enables interrupt of level 0 and 1 (while interrupt of level 2 is being serviced).
1	0	Enables interrupt of level 0 to 2 (while interrupt of level 3 is being serviced).
1	1	Enables all interrupts (waits for acknowledgment of an interrupt).

IE	Interrupt request acknowledgment enable/disable
0	Disabled
1	Enabled

Downloaded from Arrow.com.

15.4 Interrupt Servicing Operations

15.4.1 Maskable interrupt request acknowledgment

A maskable interrupt request becomes acknowledgeable when the interrupt request flag is set to 1 and the mask (MK) flag corresponding to that interrupt request is cleared to 0. A vectored interrupt request is acknowledged if interrupts are in the interrupt enabled state (when the IE flag is set to 1). However, a low-priority interrupt request is not acknowledged during servicing of a higher priority interrupt request.

The times from generation of a maskable interrupt request until vectored interrupt servicing is performed are listed in Table 15-5 below.

For the interrupt request acknowledgment timing, see **Figures 15-11** and **15-12**.

Table 15-5. Time from Generation of Maskable Interrupt Until Servicing

	Minimum Time	Maximum Time[Note]
Servicing time	9 clocks	16 clocks

Note Maximum time does not apply when an instruction from the internal RAM area is executed.

Remark 1 clock: $1/f_{CLK}$ (f_{CLK}: CPU clock)

If two or more maskable interrupt requests are generated simultaneously, the request with a higher priority level specified in the priority specification flag is acknowledged first. If two or more interrupts requests have the same priority level, the request with the highest default priority is acknowledged first.

An interrupt request that is held pending is acknowledged when it becomes acknowledgeable.

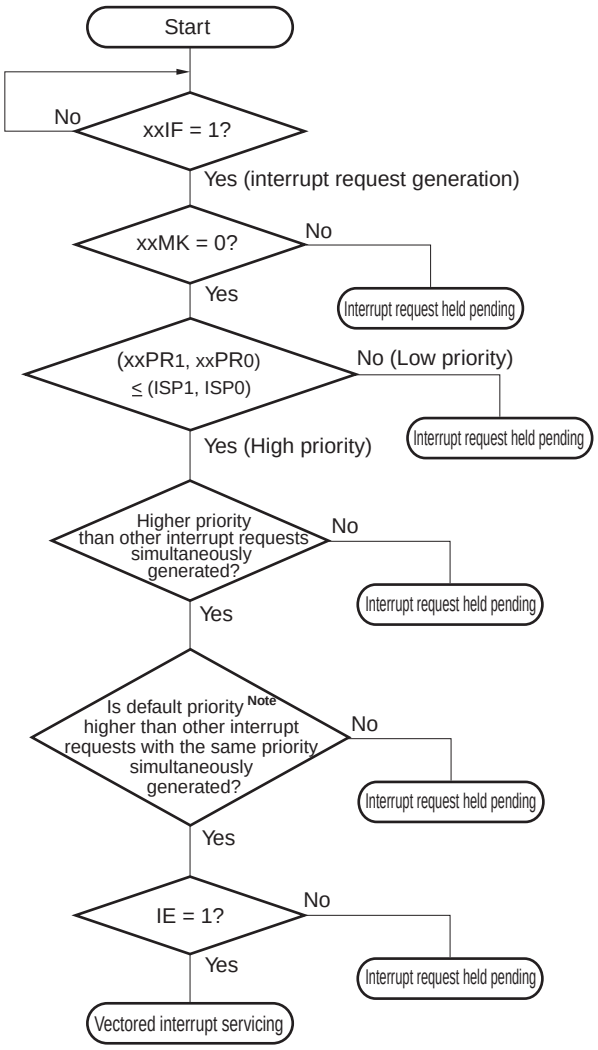
Figure 15-10 shows the interrupt request acknowledgment algorithm.

If a maskable interrupt request is acknowledged, the contents are saved into the stacks in the order of PSW, then PC, the IE flag is reset (0), and the contents of the priority specification flag corresponding to the acknowledged interrupt are transferred to the ISP1 and ISP0 flags. The vector table data determined for each interrupt request is the loaded into the PC and branched.

Restoring from an interrupt is possible by using the RETI instruction.

Downloaded from Arrow.com.

Figure 15-10. Interrupt Request Acknowledgment Processing Algorithm

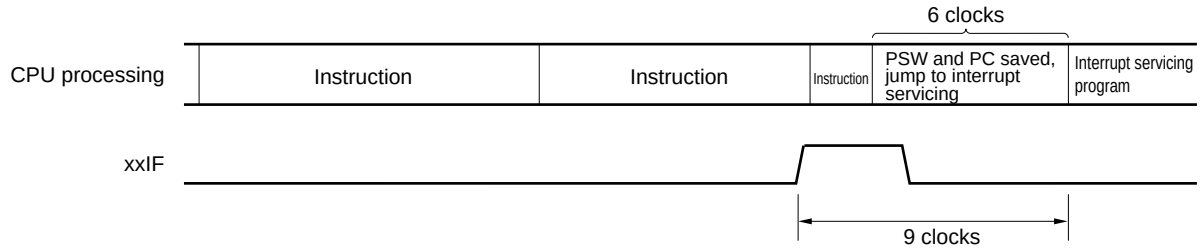


××IF: Interrupt request flag
××MK: Interrupt mask flag
××PR0: Priority specification flag 0
××PR1: Priority specification flag 1
IE: Flag that controls acknowledgment of maskable interrupt request (1 = Enable, 0 = Disable)
ISP0, ISP1: Flag that indicates the priority level of the interrupt currently being serviced (see **Figure 15-9**)

Note For the default priority, refer to **Tables 15-1** and **15-2 Interrupt Source List**.

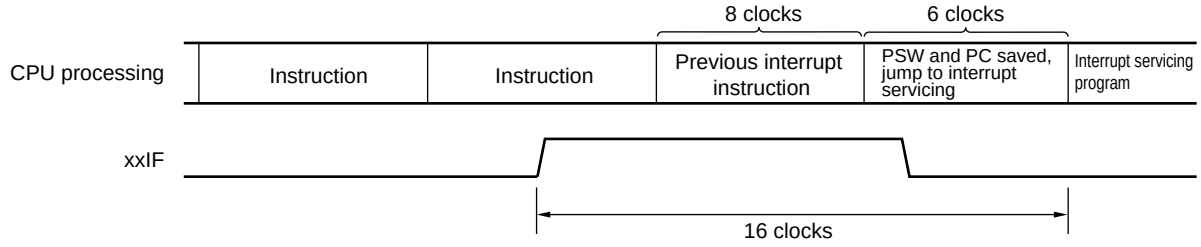
Downloaded from Arrow.com.

<R> **Figure 15-11. Interrupt Request Acknowledgment Timing (Minimum Time)**



Remark 1 clock: 1/f_{CLK} (f_{CLK}: CPU clock)

<R> **Figure 15-12. Interrupt Request Acknowledgment Timing (Maximum Time)**



Remark 1 clock: 1/f_{CLK} (f_{CLK}: CPU clock)

15.4.2 Software interrupt request acknowledgment

A software interrupt request is acknowledged by BRK instruction execution. Software interrupts cannot be disabled.

If a software interrupt request is acknowledged, the contents are saved into the stacks in the order of the program status word (PSW), then program counter (PC), the IE flag is reset (0), and the contents of the vector table (0007EH, 0007FH) are loaded into the PC and branched.

Restoring from a software interrupt is possible by using the RETB instruction.

Caution Can not use the RETI instruction for restoring from the software interrupt.

15.4.3 Multiple interrupt servicing

Multiple interrupt servicing occurs when another interrupt request is acknowledged during execution of an interrupt.

Multiple interrupt servicing does not occur unless the interrupt request acknowledgment enabled state is selected (IE = 1). When an interrupt request is acknowledged, interrupt request acknowledgment becomes disabled (IE = 0). Therefore, to enable multiple interrupt servicing, it is necessary to set (1) the IE flag with the EI instruction during interrupt servicing to enable interrupt acknowledgment.

Moreover, even if interrupts are enabled, multiple interrupt servicing may not be enabled, this being subject to interrupt priority control. Two types of priority control are available: default priority control and programmable priority control. Programmable priority control is used for multiple interrupt servicing.

In the interrupt enabled state, if an interrupt request with a priority equal to or higher than that of the interrupt currently being serviced is generated, it is acknowledged for multiple interrupt servicing. If an interrupt with a priority lower than that of the interrupt currently being serviced is generated during interrupt servicing, it is not acknowledged for multiple interrupt servicing. Interrupt requests that are not enabled because interrupts are in the interrupt disabled state or because they have a lower priority are held pending. When servicing of the current interrupt ends, the pending interrupt request is acknowledged following execution of at least one main processing instruction execution.

Table 15-6 shows relationship between interrupt requests enabled for multiple interrupt servicing and Figure 15-13 shows multiple interrupt servicing examples.

Downloaded from Arrow.com.

Table 15-6. Relationship Between Interrupt Requests Enabled for Multiple Interrupt Servicing During Interrupt Servicing

<R>

Multiple Interrupt Request / Interrupt Being Serviced		Maskable Interrupt Request								Software Interrupt Request
		Priority Level 0 (PR = 00)		Priority Level 1 (PR = 01)		Priority Level 2 (PR = 10)		Priority Level 3 (PR = 11)		
		IE = 1	IE = 0	IE = 1	IE = 0	IE = 1	IE = 0	IE = 1	IE = 0	
Maskable interrupt	ISP1 = 0, ISP0 = 0	○	×	×	×	×	×	×	×	○
	ISP1 = 0, ISP0 = 1	○	×	○	×	×	×	×	×	○
	ISP1 = 1, ISP0 = 0	○	×	○	×	○	×	×	×	○
	ISP1 = 1, ISP0 = 1	○	×	○	×	○	×	○	×	○
Software interrupt		○	×	○	×	○	×	○	×	○

Remarks 1. ○: Multiple interrupt servicing enabled

2. ×: Multiple interrupt servicing disabled

3. ISP0, ISP1, and IE are flags contained in the PSW.

ISP1 = 0, ISP0 = 0: An interrupt of level 1 or level 0 is being serviced.

ISP1 = 0, ISP0 = 1: An interrupt of level 2 is being serviced.

ISP1 = 1, ISP0 = 0: An interrupt of level 3 is being serviced.

ISP1 = 1, ISP0 = 1: Wait for an interrupt acknowledgment (all interrupts are enabled).

IE = 0: Interrupt request acknowledgment is disabled.

IE = 1: Interrupt request acknowledgment is enabled.

4. PR is a flag contained in the PR00L, PR00H, PR01L, PR10L, PR10H, PR11L registers.

PR = 00: Specify level 0 with ××PR1× = 0, ××PR0× = 0 (higher priority level)

PR = 01: Specify level 1 with ××PR1× = 0, ××PR0× = 1

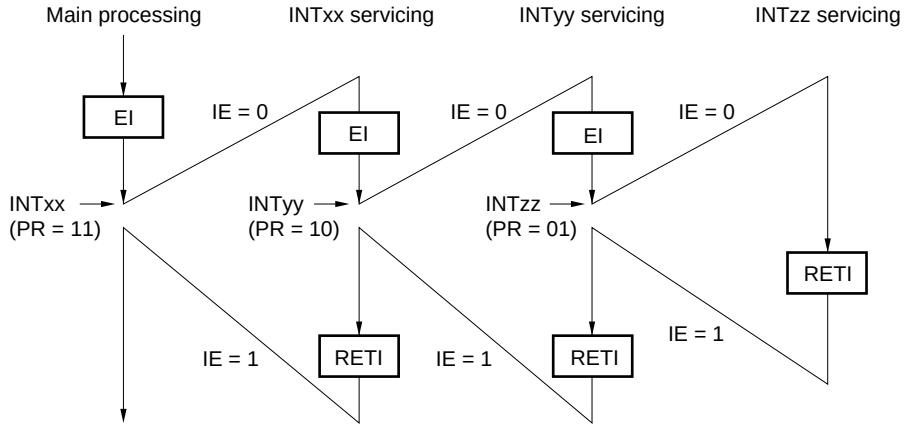
PR = 10: Specify level 2 with ××PR1× = 1, ××PR0× = 0

PR = 11: Specify level 3 with ××PR1× = 1, ××PR0× = 1 (lower priority level)

Downloaded from Arrow.com.

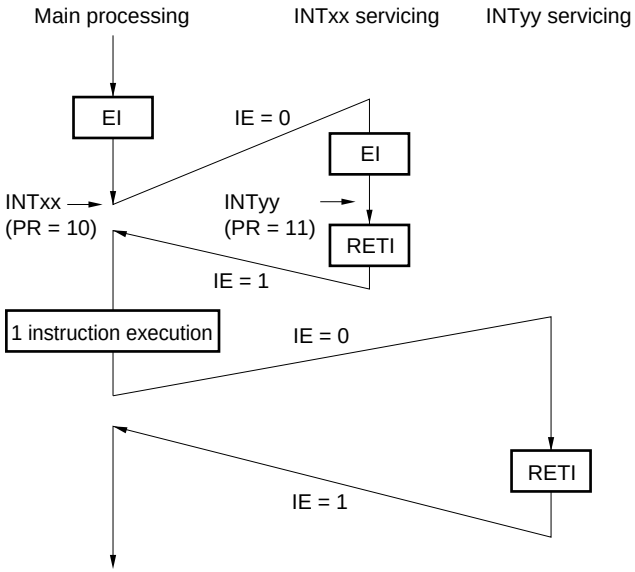
Figure 15-13. Examples of Multiple Interrupt Servicing (1/2)

Example 1. Multiple interrupt servicing occurs twice



During servicing of interrupt INTxx, two interrupt requests, INTyy and INTzz, are acknowledged, and multiple interrupt servicing takes place. Before each interrupt request is acknowledged, the EI instruction must always be issued to enable interrupt request acknowledgment.

Example 2. Multiple interrupt servicing does not occur due to priority control



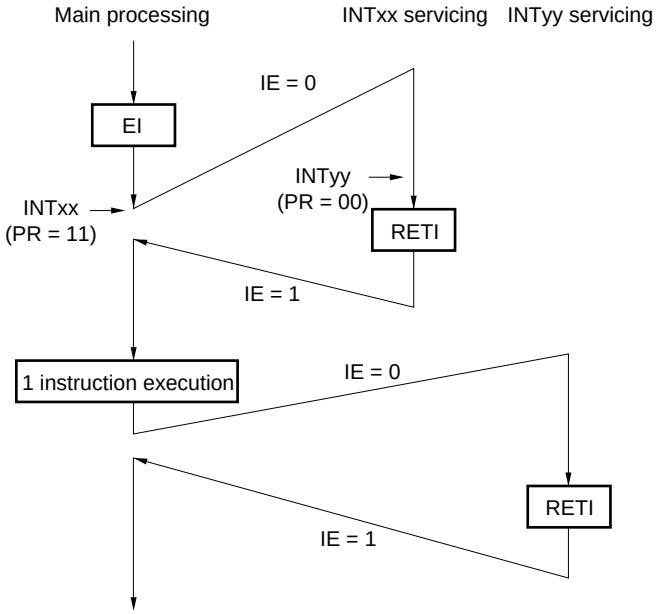
Interrupt request INTyy issued during servicing of interrupt INTxx is not acknowledged because its priority is lower than that of INTxx, and multiple interrupt servicing does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

PR = 00: Specify level 0 with ××PR1× = 0, ××PR0× = 0 (higher priority level)
PR = 01: Specify level 1 with ××PR1× = 0, ××PR0× = 1
PR = 10: Specify level 2 with ××PR1× = 1, ××PR0× = 0
PR = 11: Specify level 3 with ××PR1× = 1, ××PR0× = 1 (lower priority level)
IE = 0: Interrupt request acknowledgment is disabled
IE = 1: Interrupt request acknowledgment is enabled.

Downloaded from Arrow.com.

Figure 15-13. Examples of Multiple Interrupt Servicing (2/2)

Example 3. Multiple interrupt servicing does not occur because interrupts are not enabled



Interrupts are not enabled during servicing of interrupt INTxx (EI instruction is not issued), therefore, interrupt request INTyy is not acknowledged and multiple interrupt servicing does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

PR = 00: Specify level 0 with ××PR1× = 0, ××PR0× = 0 (higher priority level)
PR = 01: Specify level 1 with ××PR1× = 0, ××PR0× = 1
PR = 10: Specify level 2 with ××PR1× = 1, ××PR0× = 0
PR = 11: Specify level 3 with ××PR1× = 1, ××PR0× = 1 (lower priority level)
IE = 0: Interrupt request acknowledgment is disabled
IE = 1: Interrupt request acknowledgment is enabled.

Downloaded from Arrow.com.

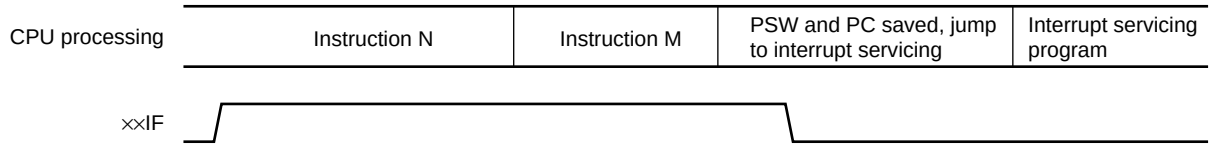
15.4.4 Interrupt request hold

There are instructions where, even if an interrupt request is issued while the instructions are being executed, interrupt request acknowledgment is held pending until the end of execution of the next instruction. These instructions (interrupt request hold instructions) are listed below.

- MOV PSW, #byte
- MOV PSW, A
- MOV1 PSW. bit, CY
- SET1 PSW. bit
- CLR1 PSW. bit
- RETB
- RETI
- POP PSW
- BTCLR PSW. bit, $addr20
- EI
- DI
- SKC
- SKNC
- SKZ
- SKNZ
- SKH
- SKNH
- Write instructions for the IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, and PR12H registers

Figure 15-14 shows the timing at which interrupt requests are held pending.

Figure 15-14. Interrupt Request Hold



Remarks 1. Instruction N: Interrupt request hold instruction

2. Instruction M: Instruction other than interrupt request hold instruction

Downloaded from Arrow.com.

CHAPTER 16 KEY INTERRUPT FUNCTION

<R> The number of key interrupt input channels differs depending on the product.

	20-pin	24-pin	30-pin
Key interrupt input channels	6 ch	4 ch	Not mounted

Remark Most of the subsequent descriptions in this chapter use 24-pin products as an example.

16.1 Functions of Key Interrupt

In 20- and 24-pin products, a key interrupt (INTKR) can be generated by inputting a rising edge/falling edge to the key interrupt input pins (KR0 to KR9).

There are two ways to identify the channel(s) to which a valid edge has been input:

- Identify the channel(s) (KR0 to KR9) by using the port input level.
- Identify the channel(s) (KR0 to KR5) by using the key interrupt flag.

<R> **Table 16-1. Assignment of Key Interrupt Detection Pins**

Key interrupt pin	Key return mode registers (KRM0, KRM1)	Key return flag register (KRF)
KR0	KRM00	KRF0
KR1	KRM01	KRF1
KR2	KRM02	KRF2
KR3	KRM03	KRF3
KR4	KRM04	KRF4
KR5	KRM05	KRF5
KR6	KRM06	–
KR7	KRM07	–
KR8	KRM08	–
KR9	KRM09	–

Remark KR0 to KR5: 20-pin products
KR0 to KR9: 24-pin products

Downloaded from Arrow.com.

16.2 Configuration of Key Interrupt

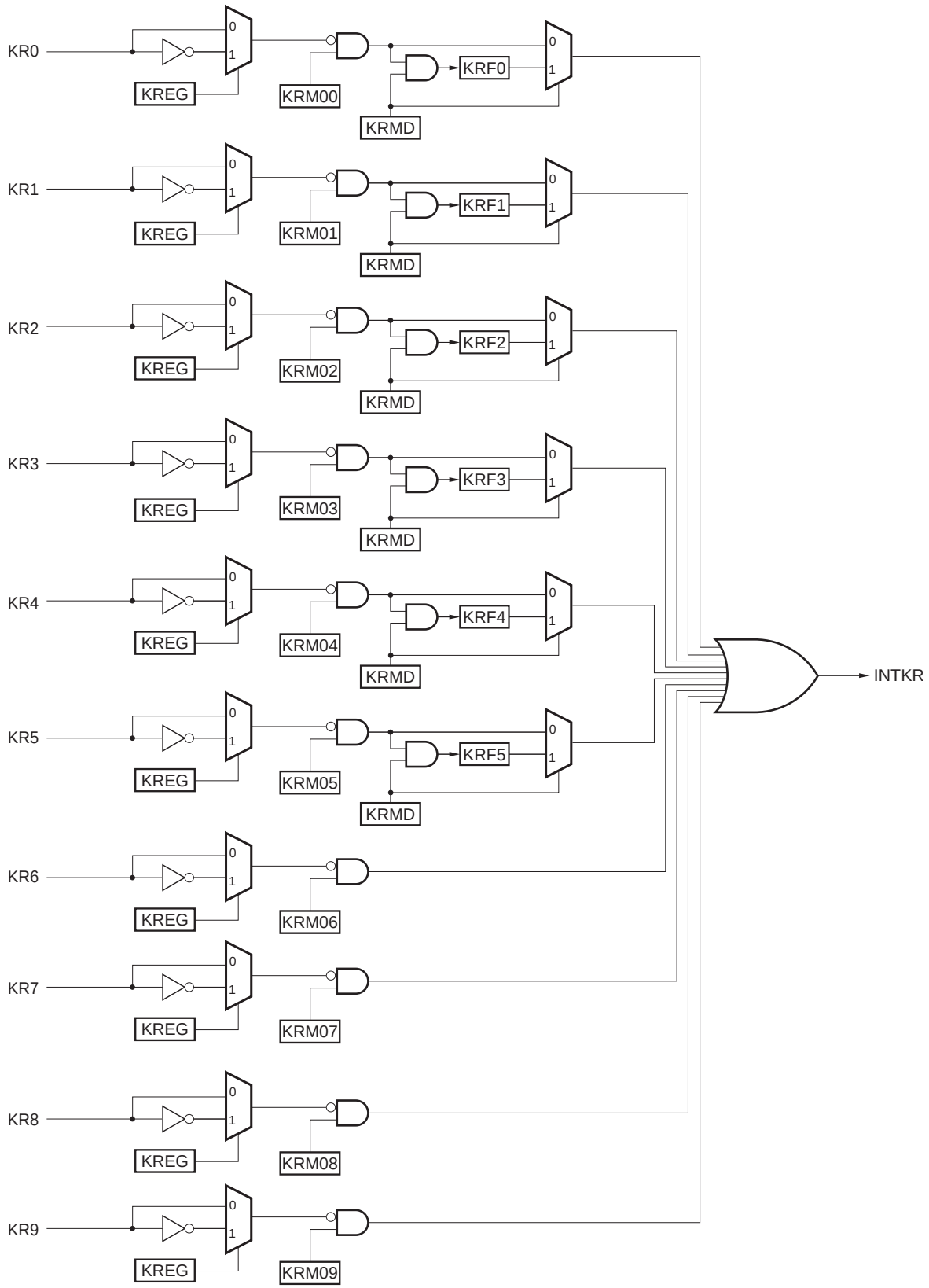
<R> The key interrupt includes the following hardware.

Table 16-2. Configuration of Key Interrupt

Item	Configuration
Control register	Key return control register (KRCTL) Key return mode control registers (KRM0, KRM1) Key return flag register (KRF) Port mode registers 0, 4, 6 (PM0, PM4, PM6)

Downloaded from Arrow.com.

Figure 16-1. Block Diagram of Key Interrupt



Remark KR0 to KR5: 20-pin products
KR0 to KR9: 24-pin products

Downloaded from Arrow.com.

16.3 Register Controlling Key Interrupt

<R> The key interrupt function is controlled by the following registers:

- Key return control register (KRCTL)
- Key return mode control registers (KRM0, KRM1)
- Key return flag register (KRF)
- Port mode registers 0, 4, 6 (PM0, PM4, PM6)

16.3.1 Key return control register (KRCTL)

This register controls the usage of the key interrupt flags (KRF0 to KRF5) and sets the detection edge.

The KRCTL register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 16-2. Format of Key Return Control Register (KRCTL)

Address: FFF34H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
KRCTL	KRMD	0	0	0	0	0	0	KREG

KRMD	Usage of Key Interrupt Flags (KRF0 to KRF5)
0	Does not use key interrupt flags
1	Uses key interrupt flags

KRRG	Selection of Detection Edge (KR0 to KR9)
0	Falling edge
1	Rising edge

<R>

KRMD	KREG	Interrupt Function
0	0	Generates a key interrupt (INTKR) when detecting a falling edge. (Identify the channel by checking the port level).
0	1	Generates a key interrupt (INTKR) when detecting a rising edge. (Identify the channel by checking the port level).
1	0	Generates a key interrupt (INTKR) when detecting a falling edge. (Identify the channel by using the key interrupt flags (KRF0 to KRF5)).
1	1	Generates a key interrupt (INTKR) when detecting a rising edge. (Identify the channel by using the key interrupt flags (KRF0 to KRF5)).

Downloaded from Arrow.com.

16.3.2 Key return mode control registers (KRM0, KRM1)

These registers set the key interrupt mode.

The KRM0 and KRM1 registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Figure 16-3. Format of Key Return Control Registers (KRM0, KRM1)

20-pin products

Address: FFF37H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
KRM0	0	0	KRM05	KRM04	KRM03	KRM02	KRM01	KRM00

24-pin products

Address: FFF37H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
KRM0	KRM07	KRM06	KRM05	KRM04	KRM03	KRM02	KRM01	KRM00

Address: FFF36H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
KRM1	0	0	0	0	0	0	KRM09	KRM08

KRM0n	Key interrupt mode control (n = 0 to 9)
0	Does not detect key interrupt signal
1	Detects key interrupt signal

<R> **Cautions 1. When a falling edge (KRMD = 0) is selected and a key interrupt signal is detected (KRM0n = 1), pull up the relevant input pins to V_{DD} by an external resistor. The internal pull-up resistor can be used by setting the relevant bits to 1 in the key interrupt input pins PU125 and PU00 to PU03 (pull-up resistor registers 12 and 0 (the bit 5 of PU12 and bits 0 to 3 of PU0)).**

<R> **2. An interrupt is generated if the target bits of the KRM0 and KRM1 registers are set while a low level (when KREG = 0) or high level (when KREG = 1) is input to the key interrupt input pin. To ignore this interrupt, set the KRM0 and KRM1 registers after disabling interrupt servicing by using the interrupt mask flag. After waiting for the key interrupt input high-level width or low-level width (see 28.4 AC characteristics or 29.4 AC characteristics), clear the interrupt request flag and enable interrupt servicing.**

3. The bits not used in the key interrupt mode can be used as normal ports.

Downloaded from Arrow.com.

<R> **16.3.3 Key return flag register (KRF)**

This register controls the key interrupt flags (KRF0 to KRF5).

The KRF register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 16-4. Format of Key Return Flag Register (KRF)

Address: FFF35H After reset: 00H R/W [Note]

Symbol	7	6	5	4	3	2	1	0
KRF	0	0	KRF5	KRF4	KRF3	KRF2	KRF1	KRF0

KRFn	Key interrupt flag (n = 0 to 5)
0	No key interrupt signal has been detected.
1	A key interrupt signal has been detected.

Note Writing "1" is invalid. To clear KRFn, write "0" to the target bits and write "1" to other bits, with the 8-bit memory manipulation instruction.

<R> **Caution For KR6 to KR9 pins, identify the channels by sequentially verifying the input levels.**

16.3.4 Port mode registers 0, 4, 6 (PM0, PM4, PM6)

<R> These registers set the input and output of ports 0, 4, and 6 in 1-bit units.

To use a key interrupt input (KR0 to KR9), set 1 to the bit of port mode register (PM0, PM4, PM6) corresponding to each port.

The PM0, PM4, and PM6 registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to FFH.

<R> **Figure 16-5. Format of Port Mode Registers 0, 4, 6 (PM0, PM4, PM6)**

Address: FFF20H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PM0	1	1	1	1	PM03	PM02	PM01	PM00

Address: FFF24H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM4	1	1	1	1	1	PM42	PM41	PM40

Address: FFF26H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM6	1	1	1	1	1	1	PM61	PM60

PMmn	I/O mode selection for PMmn pin (m = 0, 4, 6, n = 0 to 3)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Downloaded from Arrow.com.

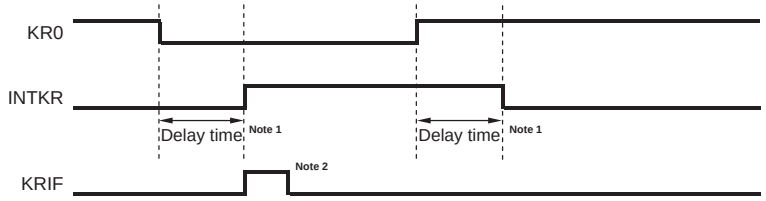
<R> 16.4 Key Interrupt Operation

16.4.1 When not using the key interrupt flag (KRMD = 0)

A key interrupt (INTKR) is generated when the valid edge specified by the setting of the KREG bit is input to a key interrupt pin (KR0 to KR9). The channel to which the valid edge was input can be identified by reading the port register and checking the port level after the key interrupt (INTKR) is generated.

The INTKR signal changes according to the input level of the key interrupt input pin (KR0 to KR5).

Figure 16-6. Operation of INTKR Signal When a Key Interrupt is Input to a Single Channel (When KRMD = 0 and KREG = 0)

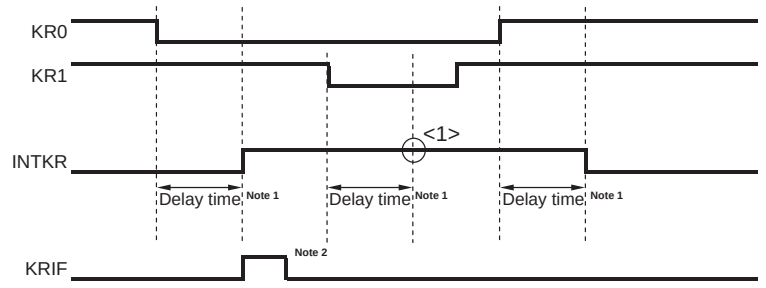


Notes 1. The maximum delay time is the maximum value of the high-level width and low-level width of the key interrupt input (see **28.4 AC Characteristics** or **29.4 AC Characteristics**).

2. Cleared by acknowledgment of vectored interrupt request or software.

The operation when a valid edge is input to multiple key interrupt input pins is shown in Figure 16-7. The INTKR signal is set while a low level is being input to one pin (when KREG is set to 0). Therefore, even if a falling edge is input to another pin in this period, a key interrupt (INTKR) will not be generated again (<1> in the figure).

Figure 16-7. Operation of INTKR Signal When Key Interrupts Are Input to Multiple Channels (When KRMD = 0 and KREG = 0)



Notes 1. The maximum delay time is the maximum value of the high-level width and low-level width of the key interrupt input (see **28.4 AC Characteristics** or **29.4 AC Characteristics**).

2. Cleared by acknowledgment of vectored interrupt request or bit cleared by software.

Downloaded from Arrow.com.

<R> **16.4.2 When using the key interrupt flag (KRMD = 1)**

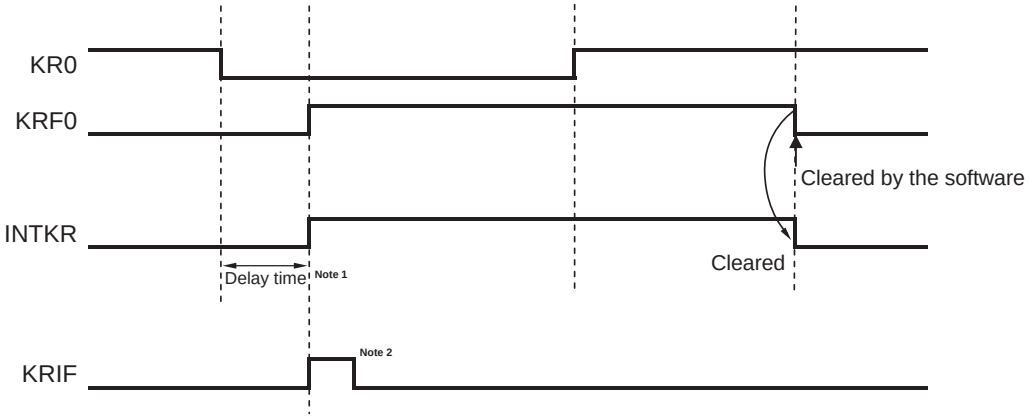
A key interrupt (INTKR) is generated when the valid edge specified by the setting of the KREG bit is input to a key interrupt pin (KR0 to KR5). The channels to which the valid edge was input can be identified by reading the key return flag register (KRF) after the key interrupt (INTKR) is generated.

If the KRMD bit is set to 1, the INTKR signal is cleared by clearing the corresponding bit in the KRF register.

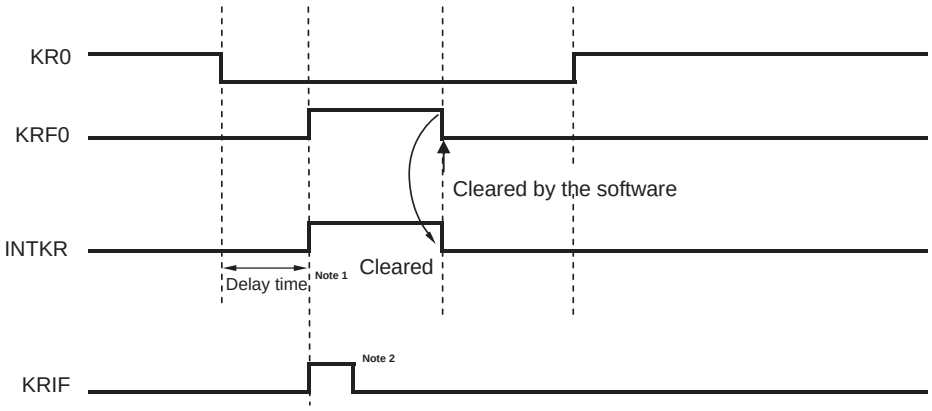
As shown in Figure 16-8, only one interrupt is generated each time a falling edge is input to one channel (when KREG = 0), regardless of whether the KRFn bit is cleared before or after a rising edge is input.

Figure 16-8. Basic Operation of the INTKR Signal When the Key Interrupt Flag Is Used (When KRMD = 1 and KREG = 0)

(a) KRF0 is cleared after a rising edge is input to the KR0 pin



(b) KRF0 is cleared before a rising edge is input to the KR0 pin



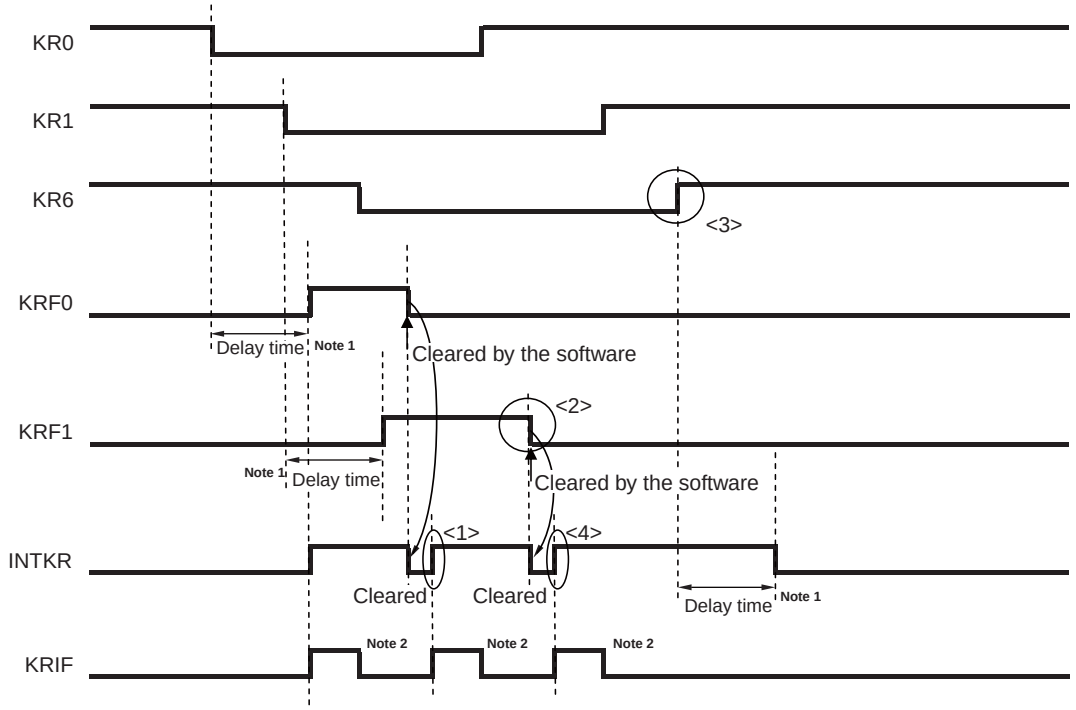
Notes 1. The maximum delay time is the maximum value of the high-level width and low-level width of the key interrupt input (see **28.4 AC Characteristics** or **29.4 AC Characteristics**).

2. Cleared by acknowledgment of vectored interrupt request or bit cleared by software.

Downloaded from Arrow.com.

<R> The operation when a valid edge is input to multiple key interrupt input pins is shown in Figure 16-9. A falling edge is also input to the KR1 and KR6 pins after a falling edge was input to the KR0 pin (when KREG = 0). The KRF1 bit is set when the KRF0 bit is cleared. A key interrupt (INTKR) is therefore generated one clock (f_{CLK}) after the KRF0 bit is cleared (<1> in the figure). Also, after a falling edge has been input to the KR6 pin, a low level continues to be input to this pin (<3> in the figure) until the KRF1 bit is cleared (<2> in the figure). A key interrupt (INTKR) is therefore generated one clock (f_{CLK}) after the KRF1 bit is cleared (<4> in the figure). It is thus possible to generate a key interrupt (INTKR) when a valid edge is input to multiple channels.

Figure 16-9. Operation of INTKR Signal When Key Interrupts Are Input to Multiple Channels (When KRMD = 1 and KREG = 0)



Notes 1. The maximum delay time is the maximum value of the high-level width and low-level width of the key interrupt input (see **28.4 AC Characteristics** or **29.4 AC Characteristics**).

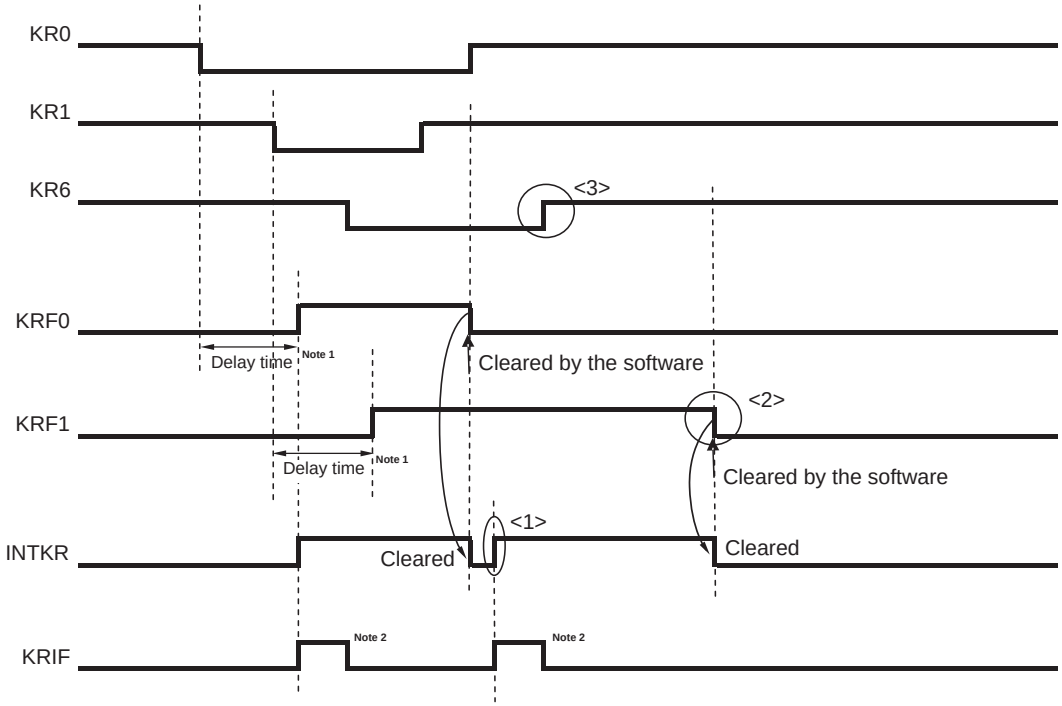
2. Cleared by acknowledgment of vectored interrupt request or bit cleared by software.

Remark f_{CLK}: CPU/peripheral hardware clock frequency

Downloaded from Arrow.com.

<R> The operation when a valid edge is input to the KR6 to KR9 pins without generating a key interrupt (INTKR) is shown in Figure 16-10. A falling edge is also input to the KR1 and KR6 pins after a falling edge was input to the KR0 pin (when KREG = 0). The KR1 pin becomes high level when the KRF0 bit is cleared, but because the KRF1 bit is set, a key interrupt (INTKR) is generated one clock (f_{CLK}) after the KRF0 bit is cleared (<1> in the figure). Also, because the KR6 pin was high level (<3> in the figure) before the KRF1 bit was cleared (<2> in the figure) a key interrupt (INTKR) is not generated for the KR6 pin.

Figure 16-10. Operation When an INTKR Signal Is Not Generated upon Input of a Valid Edge to KR6 to KR9 (When KRMD = 1 and KREG = 0)



Notes 1. The maximum delay time is the maximum value of the high-level width and low-level width of the key interrupt input (see **28.4 AC Characteristics** or **29.4 AC Characteristics**).

2. Cleared by acknowledgment of vectored interrupt request or bit cleared by software.

Remark f_{CLK}: CPU/peripheral hardware clock frequency

Downloaded from Arrow.com.

CHAPTER 17 STANDBY FUNCTION

17.1 Standby Function

The standby function reduces the operating current of the system, and the following three modes are available.

(1) HALT mode

HALT instruction execution sets the HALT mode. In the HALT mode, the CPU operation clock is stopped. If the high-speed system clock oscillator or high-speed on-chip oscillator is operating before the HALT mode is set, oscillation of each clock continues. In this mode, the operating current is not decreased as much as in the STOP mode, but the HALT mode is effective for restarting operation immediately upon interrupt request generation and carrying out intermittent operations frequently.

(2) STOP mode

STOP instruction execution sets the STOP mode. In the STOP mode, the high-speed system clock oscillator and high-speed on-chip oscillator stop, stopping the whole system, thereby considerably reducing the CPU operating current.

Because this mode can be cleared by an interrupt request, it enables intermittent operations to be carried out. However, because a wait time is required to secure the oscillation stabilization time after the STOP mode is released when the X1 clock is selected, select the HALT mode if it is necessary to start processing immediately upon interrupt request generation.

(3) SNOOZE mode

In the case of CSI00 or UART0 data reception and an A/D conversion request by the timer trigger signal (the interrupt request signal (INTIT)), the STOP mode is exited, the CSI00 or UART0 data is received without operating the CPU, and A/D conversion is performed. This can only be specified when the high-speed on-chip oscillator clock is selected for the CPU/peripheral hardware clock (f_{CLK}).

In either of these two modes, all the contents of registers, flags and data memory just before the standby mode is set are held. The I/O port output latches and output buffer statuses are also held.

Cautions 1. When shifting to the STOP mode, be sure to stop the peripheral hardware operation operating with main system clock before executing STOP instruction (other than SNOOZE mode setting unit).

2. When using CSI00, UART0, or the A/D converter in the SNOOZE mode, set up serial standby control register 0 (SSC0) and A/D converter mode register 2 (ADM2) before switching to the STOP mode. For details, see 11.3 Registers Controlling Serial Array Unit and 10.3 Registers Controlling A/D Converter.

3. To reduce the current consumption of the A/D converter when the standby function is used, first clear bit 7 (ADCS) and bit 0 (ADCE) of A/D converter mode register 0 (ADM0) to 0 to stop the A/D conversion, and then execute the STOP instruction.

4. It can be selected by the option byte whether the low-speed on-chip oscillator continues oscillating or stops in the HALT or STOP mode. For details, see CHAPTER 23 OPTION BYTE.

Downloaded from Arrow.com.

17.2 Registers controlling standby function

The registers which control the standby function are described below.

- Oscillation stabilization time counter status register (OSTC)
- Oscillation stabilization time select register (OSTS)

Remark For details of registers described above, see **CHAPTER 5 CLOCK GENERATOR**. For registers which control the SNOOZE mode, **CHAPTER 10 A/D CONVERTER** and **CHAPTER 11 SERIAL ARRAY UNIT**.

17.3 Standby Function Operation

17.3.1 HALT mode

(1) HALT mode

The HALT mode is set by executing the HALT instruction. HALT mode can be set regardless of whether the CPU clock before the setting was the high-speed system clock or the high-speed on-chip oscillator clock.

The operating statuses in the HALT mode are shown below.

<R> **Caution Because the interrupt request signal is used to clear the HALT mode, if the interrupt mask flag is 0 (the interrupt processing is enabled) and the interrupt request flag is 1 (the interrupt request signal is generated), the HALT mode is not entered even if the HALT instruction is executed in such a situation.**

Downloaded from Arrow.com.

Table 17-1. Operating Statuses in HALT Mode

HALT Mode Setting / Item			When HALT Instruction Is Executed While CPU Is Operating on Main System Clock		
			When CPU Is Operating on High-speed On-chip Oscillator Clock (f_{IH})	When CPU Is Operating on X1 Clock (f_X)	When CPU Is Operating on External Main System Clock (f_{EX})
System clock			Clock supply to the CPU is stopped		
	Main system clock	f_{IH}	Operation continues (cannot be stopped)	Operation disabled	
		f_X	Operation disabled	Operation continues (cannot be stopped)	Cannot operate
		f_{EX}		Cannot operate	Operation continues (cannot be stopped)
	f_{IL}		Set by bits 0 (WDSTBYON) and 4 (WDTON) of option byte (000C0H), and WUTMMCK0 bit of operation speed mode control register (OSMC) • WUTMMCK0 = 1: Oscillates • WUTMMCK0 = 0 and WDTON = 0: Stops • WUTMMCK0 = 0, WDTON = 1, and WDSTBYON = 1: Oscillates • WUTMMCK0 = 0, WDTON = 1, and WDSTBYON = 0: Stops		
CPU			Operation stopped		
Code flash memory			Operation stopped		
Data flash memory					
RAM					
Port (latch)			Status before HALT mode was set is retained		
Timer array unit			Operable		
12-bit interval timer					
Watchdog timer			Set by bit 0 (WDSTBYON) of option byte (000C0H) • WDSTBYON = 0: Operation stopped • WDSTBYON = 1: Operation continues (cannot be stopped)		
Clock output/buzzer output			Operable		
A/D converter					
Serial array unit (SAU)					
Serial interface (IICA)					
Multiplier and divider/multiplyaccumulator					
DMA controller					
Power-on-reset function					
Voltage detection function					
External interrupt					
Key interrupt function					
CRC operation function			In the calculation of the RAM area, operable when DMA is executed only		
RAM parity error detection function			Operable when DMA is executed only		
RAM guard function					
SFR guard function					
Illegal-memory access detection function					

Remark Operation stopped: Operation is automatically stopped before switching to the HALT mode.
Operation disabled: Operation is stopped before switching to the HALT mode.
f_{IH}: High-speed on-chip oscillator clock
f_{IL}: Low-speed on-chip oscillator clock
f_X: X1 clock
f_{EX}: External main system clock

Downloaded from Arrow.com.

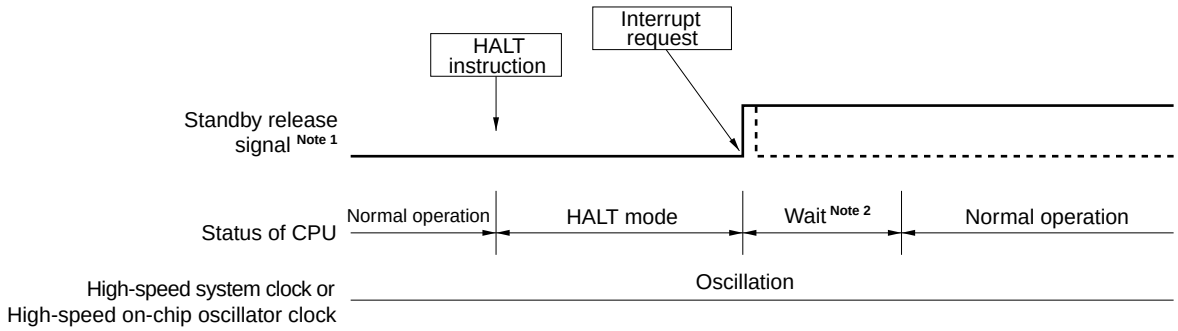
(2) HALT mode release

The HALT mode can be released by the following two sources.

(a) Release by unmasked interrupt request

When an unmasked interrupt request is generated, the HALT mode is released. If interrupt acknowledgment is enabled, vectored interrupt servicing is carried out. If interrupt acknowledgment is disabled, the next address instruction is executed.

Figure 17-1. HALT Mode Release by Interrupt Request Generation



Notes 1. For details of the standby release signal, see **Figure 15-1 Basic Configuration of Interrupt Function**.

2. Wait time for HALT mode release

- When vectored interrupt servicing is carried out: 15 to 16 clocks
- When vectored interrupt servicing is not carried out: 9 to 10 clocks

Remark The broken lines indicate the case when the interrupt request which has released the standby mode is acknowledged.

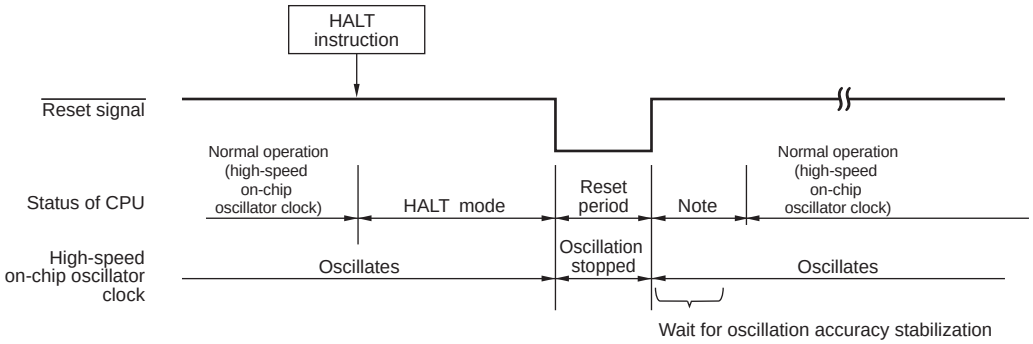
Downloaded from Arrow.com.

(b) Release by reset signal generation

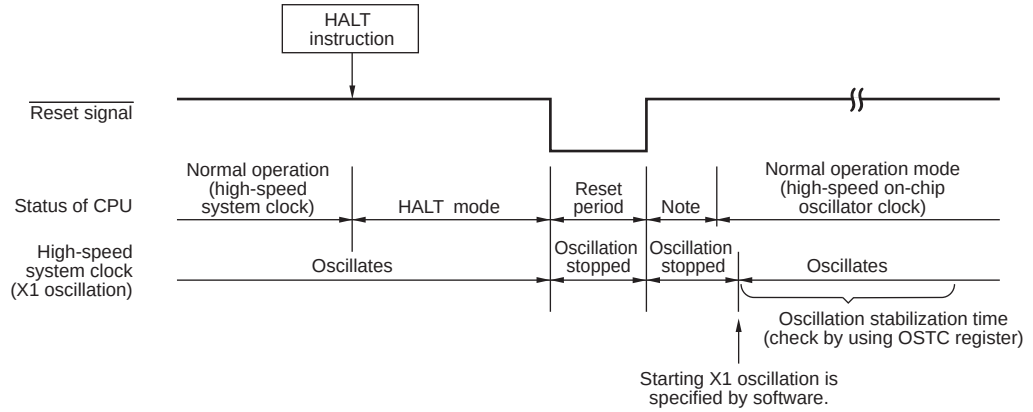
When the reset signal is generated, HALT mode is released, and then, as in the case with a normal reset operation, the program is executed after branching to the reset vector address.

Figure 17-2. HALT Mode Release by Reset

(1) When high-speed on-chip oscillator clock is used as CPU clock



(2) When high-speed system clock is used as CPU clock



Note For reset processing time, see **CHAPTER 18 RESET FUNCTION**. For reset processing time for Power-on-reset circuit (POR) and Voltage detection (LVD) circuit, see **CHAPTER 19 POWER-ON-RESET CIRCUIT.**

Downloaded from Arrow.com.

17.3.2 STOP mode

(1) STOP mode setting and operating statuses

The STOP mode is set by executing the STOP instruction, and the mode can be set regardless of whether the CPU clock before the setting was the high-speed system clock or the high-speed on-chip oscillator clock.

<R> **Caution Because the interrupt request signal is used to clear the STOP mode, if the interrupt mask flag is 0 (the interrupt processing is enabled) and the interrupt request flag is 1 (the interrupt request signal is generated), the STOP mode is immediately cleared if set when the STOP instruction is executed in such a situation. Accordingly, once the STOP instruction is executed, the system returns to its normal operation after the elapse of release time from the STOP mode.**

The operating statuses in the STOP mode are shown below.

Downloaded from Arrow.com.

Table 17-2. Operating Statuses in STOP Mode

STOP Mode Setting / Item	When STOP Instruction Is Executed While CPU Is Operating on Main System Clock		
	When CPU Is Operating on High-speed On-chip Oscillator Clock (f_{IH})	When CPU Is Operating on X1 Clock (f_X)	When CPU Is Operating on External Main System Clock (f_{EX})
System clock	Clock supply to the CPU is stopped		
Main system clock (f_{IH})	Stopped		
Main system clock (f_X)	Stopped		
Main system clock (f_{EX})	Stopped		
f_{IL}	Set by bits 0 (WDSTBYON) and 4 (WDTON) of option byte (000C0H), and WUTMMCK0 bit of operation speed mode control register (OSMC) • WUTMMCK = 1: Oscillates • WUTMMCK = 0 and WDTON = 0: Stops • WUTMMCK = 0, WDTON = 1, and WDSTBYON = 1: Oscillates • WUTMMCK = 0, WDTON = 1, and WDSTBYON = 0: Stops		
CPU	Operation stopped		
Code flash memory			
Data flash memory	Operation stopped		
RAM	Operation stopped		
Port (latch)	Status before STOP mode was set is retained		
Timer array unit	Operation disabled		
12-bit interval timer	Operable		
Watchdog timer	Set by bit 0 (WDSTBYON) of option byte (000C0H) • WDSTBYON = 0: Operation stopped • WDSTBYON = 1: Operation continues (cannot be stopped)		
Clock output/buzzer output	Operation disabled		
A/D converter	Wakeup operation is enabled (switching to the SNOOZE mode)		
Serial array unit (SAU)	Wakeup operation is enabled only for CSI00 and UART0 (switching to the SNOOZE mode) Operation is disabled for anything other than CSI00 and UART0		
Serial interface (IICA)	Wakeup by address match operable		
Multiplier and divider/multiply-accumulator	Operation disabled		
DMA controller			
Power-on-reset function	Operable		
Voltage detection function			
External interrupt			
Key interrupt function			
CRC operation function	Operation stopped		
RAM parity error detection function			
RAM guard function			
SFR guard function			
Illegal-memory access detection function			

Remark Operation stopped: Operation is automatically stopped before switching to the STOP mode.
Operation disabled: Operation is stopped before switching to the STOP mode.
f_{IH}: High-speed on-chip oscillator clock
f_{IL}: Low-speed on-chip oscillator clock
f_X: X1 clock
f_{EX}: External main system clock

Downloaded from Arrow.com.

(2) STOP mode release

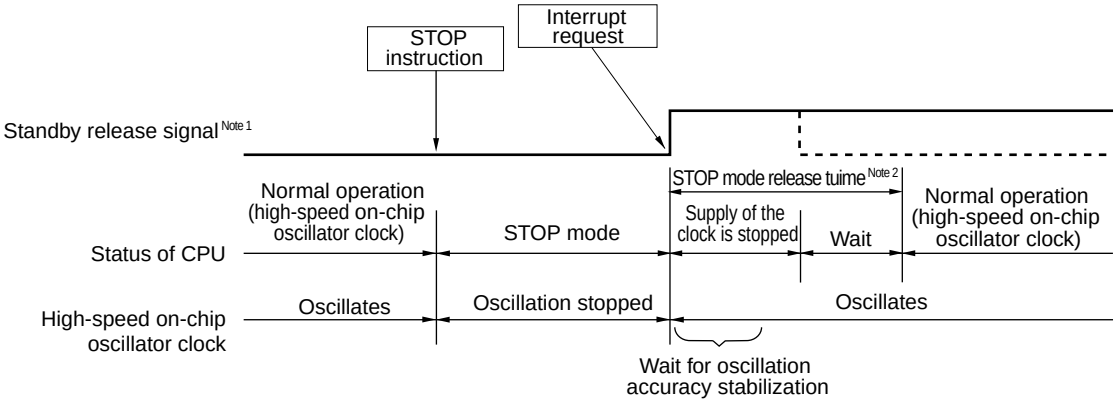
The STOP mode can be released by the following two sources.

(a) Release by unmasked interrupt request

When an unmasked interrupt request is generated, the STOP mode is released. After the oscillation stabilization time has elapsed, if interrupt acknowledgment is enabled, vectored interrupt servicing is carried out. If interrupt acknowledgment is disabled, the next address instruction is executed.

Figure 17-3. STOP Mode Release by Interrupt Request Generation (1/2)

(1) When high-speed on-chip oscillator clock is used as CPU clock



Notes 1. For details of the standby release signal, see **Figure 15-1 Basic Configuration of Interrupt Function**.

2. STOP mode release time

<R> Supply of the clock is stopped: 18 μs to 65 μs

Wait

- When vectored interrupt servicing is carried out: 7 clocks
- When vectored interrupt servicing is not carried out: 1 clock

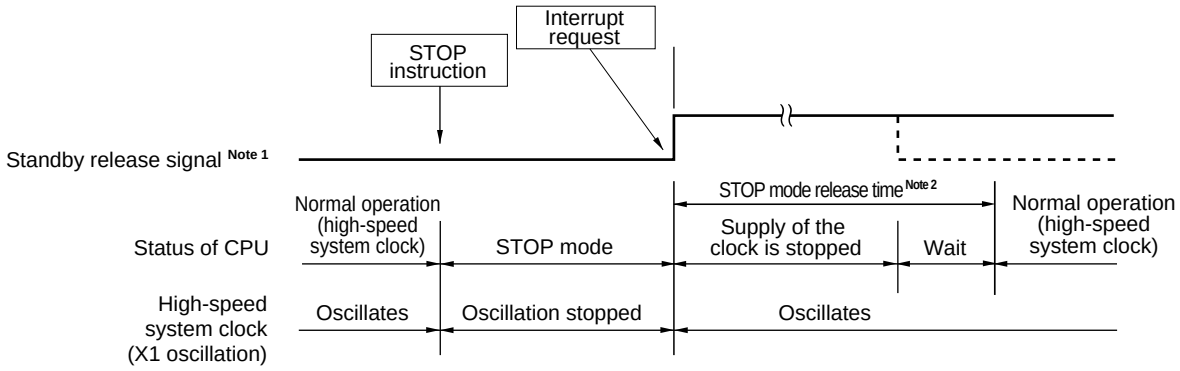
Remarks 1. The clock supply stop time varies depending on the temperature conditions and STOP mode period.

2. The broken lines indicate the case when the interrupt request that has released the standby mode is acknowledged.

Downloaded from Arrow.com.

Figure 17-3. STOP Mode Release by Interrupt Request Generation (2/2)

(2) When high-speed system clock (X1 oscillation) is used as CPU clock



Notes 1. For details of the standby release signal, see **Figure 15-1 Basic Configuration of Interrupt Function**.

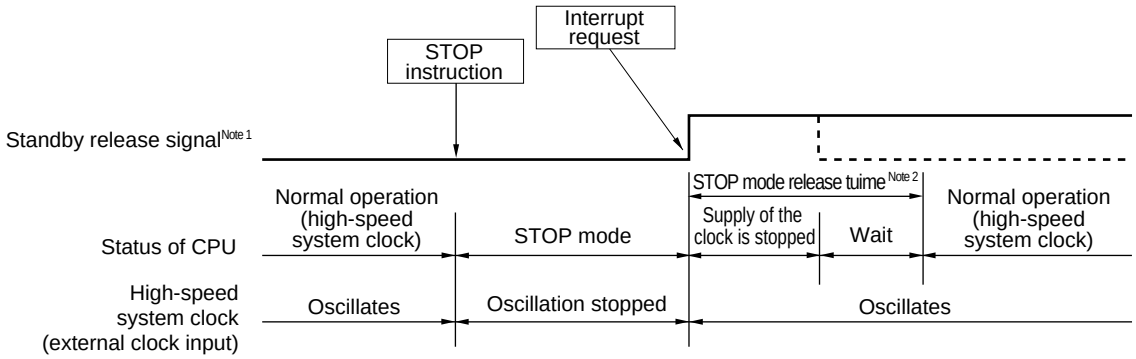
2. STOP mode release time

Supply of the clock is stopped: 18 μs to "whichever is longer 65 μs and the oscillation stabilization time (set by OSTS)"

Wait

- When vectored interrupt servicing is carried out: 10 to 11 clocks
- When vectored interrupt servicing is not carried out: 4 to 5 clocks

(3) When high-speed system clock (external clock input) is used as CPU clock



<R> **Notes 1.** For details of the standby release signal, see **Figure 15-1**.

<R> **2.** STOP mode release time

Supply of the clock is stopped: 18 μs to 65 μs

Wait

- When vectored interrupt servicing is carried out: 7 clocks
- When vectored interrupt servicing is not carried out: 1 clock

<R> **Caution To reduce the oscillation stabilization time after release from the STOP mode while CPU operates based on the high-speed system clock (X1 oscillation), switch the clock to the high-speed on-chip oscillator clock temporarily before executing the STOP instruction.**

<R> **Remarks 1.** The clock supply stop time varies depending on the temperature conditions and STOP mode period.

<R> **2.** The broken lines indicate the case when the interrupt request that has released the standby mode is acknowledged.

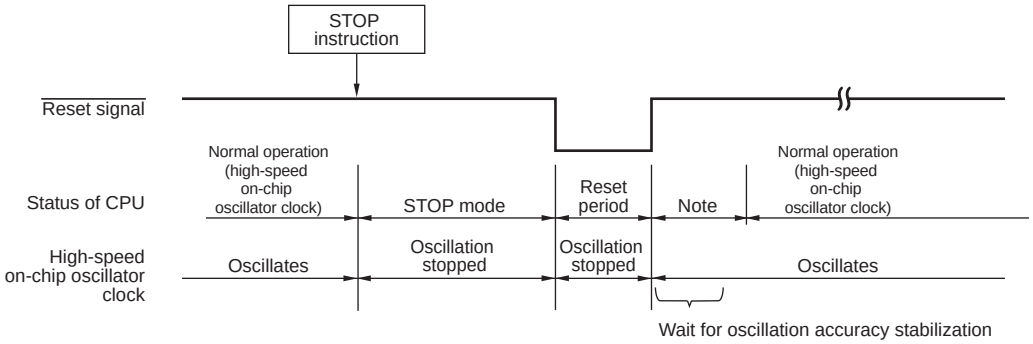
Downloaded from Arrow.com.

(b) Release by reset signal generation

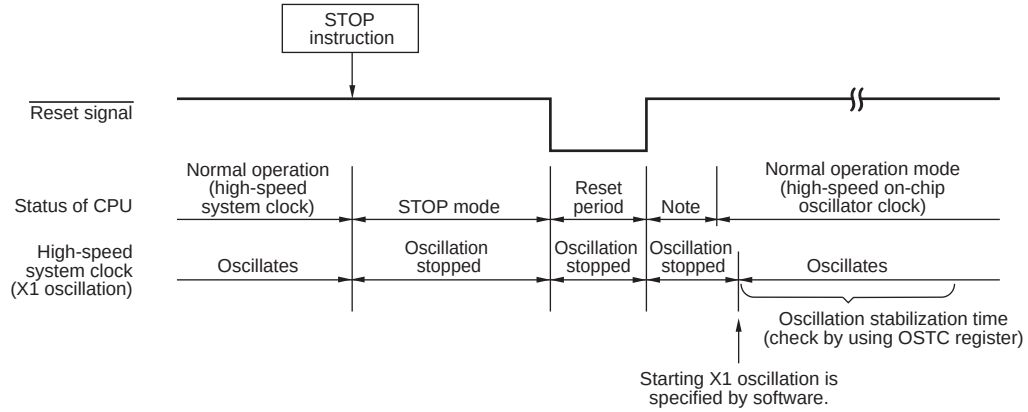
When the reset signal is generated, STOP mode is released, and then, as in the case with a normal reset operation, the program is executed after branching to the reset vector address.

Figure 17-4. STOP Mode Release by Reset

(1) When high-speed on-chip oscillator clock is used as CPU clock



(2) When high-speed system clock is used as CPU clock



Note For reset processing time, see **CHAPTER 18 RESET FUNCTION**. For reset processing time for Power-on-reset circuit (POR) and Voltage detection (LVD) circuit, see **CHAPTER 19 POWER-ON-RESET CIRCUIT.**

Downloaded from Arrow.com.

17.3.3 SNOOZE mode

(1) SNOOZE mode setting and operating statuses

The SNOOZE mode can only be specified for CSI00, UART0, or the A/D converter. Note that this mode can only be specified if the CPU clock is the high-speed on-chip oscillator clock.

When using CSI00 or UART0 in the SNOOZE mode, set the SWC0 bit of the serial standby control register 0 (SSC0) to 1 immediately before switching to the STOP mode. For details, see **11.3 Registers Controlling Serial Array Unit**.

When using the A/D converter in the SNOOZE mode, set the AWC bit of the A/D converter mode register 2 (ADM2) to 1 immediately before switching to the STOP mode. For details, see **10.3 Registers Controlling A/D Converter**.

In SNOOZE mode transition, wait status to be only following time.

Transition time from STOP mode to SNOOZE mode: 18 μs to 65 μs

Remark Transition time from STOP mode to SNOOZE mode varies depending on the temperature conditions and the STOP mode period.

From SNOOZE to normal operation

- When vectored interrupt servicing is carried out:
  - HS (high-speed main) mode : "6.65 μs to 9.44 μs" + 7 clocks
  - LS (low-speed main) mode : "1.10 μs to 5.08 μs" + 7 clocks
- When vectored interrupt servicing is not carried out:
  - HS (high-speed main) mode : "6.65 μs to 9.44 μs" + 1 clock
  - LS (low-speed main) mode : "1.10 μs to 5.08 μs" + 1 clock

The operating statuses in the SNOOZE mode are shown below.

Downloaded from Arrow.com.

Table 17-3. Operating Statuses in SNOOZE Mode

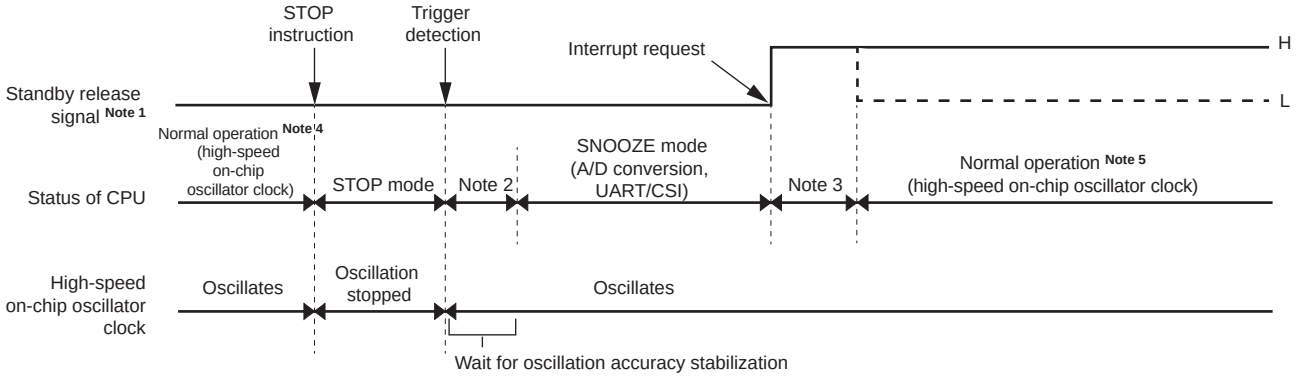
Item \ STOP Mode Setting		When Inputting CSI00/UART0 Data Reception Signal or A/D Converter Timer Trigger Signal While in STOP Mode
		When CPU Is Operating on High-speed On-chip Oscillator Clock (f_{IH})
System clock		Clock supply to the CPU is stopped
Main system clock	f_{IH}	Operation started
	f_X	Stopped
	f_{EX}	
f_{IL}		Set by bits 0 (WDSTBYON) and 4 (WDTON) of option byte (000C0H), and WUTMMCK0 bit of operation speed mode control register (OSMC) • WUTMMCK = 1: Oscillates • WUTMMCK = 0 and WDTON = 0: Stops • WUTMMCK = 0, WDTON = 1, and WDSTBYON = 1: Oscillates • WUTMMCK = 0, WDTON = 1, and WDSTBYON = 0: Stops
CPU		Operation stopped
Code flash memory		
Data flash memory		
RAM		
Port (latch)		Status while in STOP mode continues
Timer array unit		Operation disabled
12-bit interval timer		Operable
Watchdog timer		Set by bit 0 (WDSTBYON) of option byte (000C0H) • WDSTBYON = 0: Operation stopped • WDSTBYON = 1: Operation continues (cannot be stopped)
Clock output/buzzer output		Operation stopped
A/D converter		Operable
Serial array unit (SAU)		Operable only CSI00 and UART0 only. Operation disabled other than CSI00 and UART0.
Serial interface (IICA)		Operation disabled
Multiplier and divider/multiply-accumulator		
DMA controller		
Power-on-reset function		Operable
Voltage detection function		
External interrupt		
Key interrupt function		
CRC operation function		Operation disabled
RAM parity error detection function		
RAM guard function		
SFR guard function		
Illegal-memory access detection function		

Remark Operation stopped: Operation is automatically stopped before switching to the STOP mode.
Operation disabled: Operation is stopped before switching to the STOP mode.
f_{IH}: High-speed on-chip oscillator clock
f_{IL}: Low-speed on-chip oscillator clock
f_X: X1 clock
f_{EX}: External main system clock

Downloaded from Arrow.com.

<R> **(2) Timing diagram when the interrupt request signal is generated in the SNOOZE mode**

Figure 17-5. When the Interrupt Request Signal is Generated in the SNOOZE Mode



Notes 1. For details of the standby release signal, see **Figure 15-1.**

2. Transition time from STOP mode to SNOOZE mode

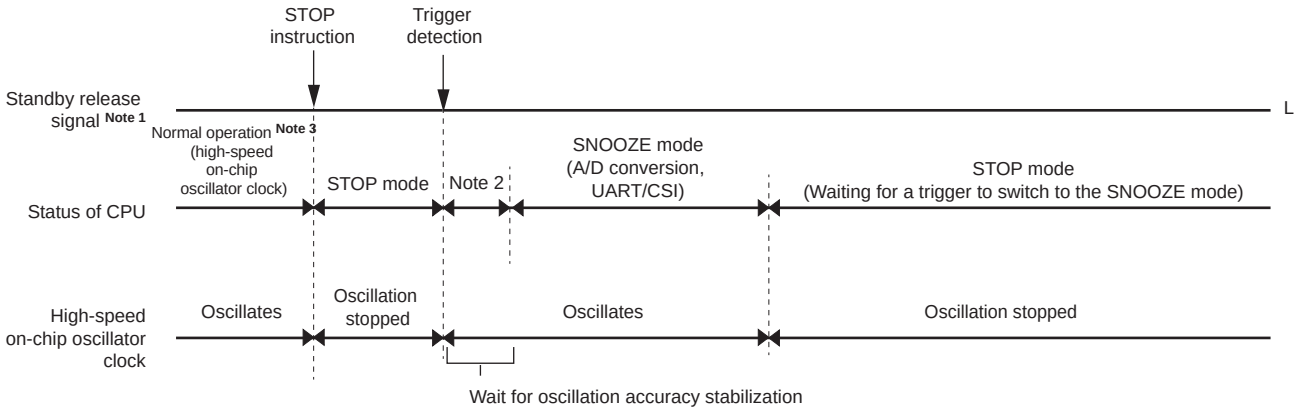
3. Transition time from SNOOZE mode to normal operation

4. Enable the SNOOZE mode (AWC = 1 or SWC = 1) immediately before switching to the STOP mode.

5. Be sure to release the SNOOZE mode (AWC = 0 or SWC = 0) immediately after return to the normal operation.

<R> **(3) Timing diagram when the interrupt request signal is not generated in the SNOOZE mode**

Figure 17-6. When the Interrupt Request Signal is not Generated in the SNOOZE Mode



Notes 1. For details of the standby release signal, see **Figure 15-1.**

2. Transition time from STOP mode to SNOOZE mode

3. Enable the SNOOZE mode (AWC = 1 or SWC = 1) immediately before switching to the STOP mode.

Remark For details of the SNOOZE mode function, see **CHAPTER 10 A/D CONVERTER** and **CHAPTER 11 SERIAL ARRAY UNIT**.

Downloaded from Arrow.com.

CHAPTER 18 RESET FUNCTION

The following seven operations are available to generate a reset signal.

(1) External reset input via $\overline{\text{RESET}}$ pin
(2) Internal reset by watchdog timer program loop detection
(3) Internal reset by comparison of supply voltage and detection voltage of power-on-reset (POR) circuit
(4) Internal reset by comparison of supply voltage of the voltage detector (LVD) and detection voltage
(5) Internal reset by execution of illegal instruction[Note]
(6) Internal reset by RAM parity error
(7) Internal reset by illegal-memory access

External and internal resets start program execution from the address at 0000H and 0001H when the reset signal is generated.

A reset is effected when a low level is input to the $\overline{\text{RESET}}$ pin, the watchdog timer overflows, or by POR and LVD circuit voltage detection, execution of illegal instruction[Note], RAM parity error or illegal-memory access, and each item of hardware is set to the status shown in Table 18-1.

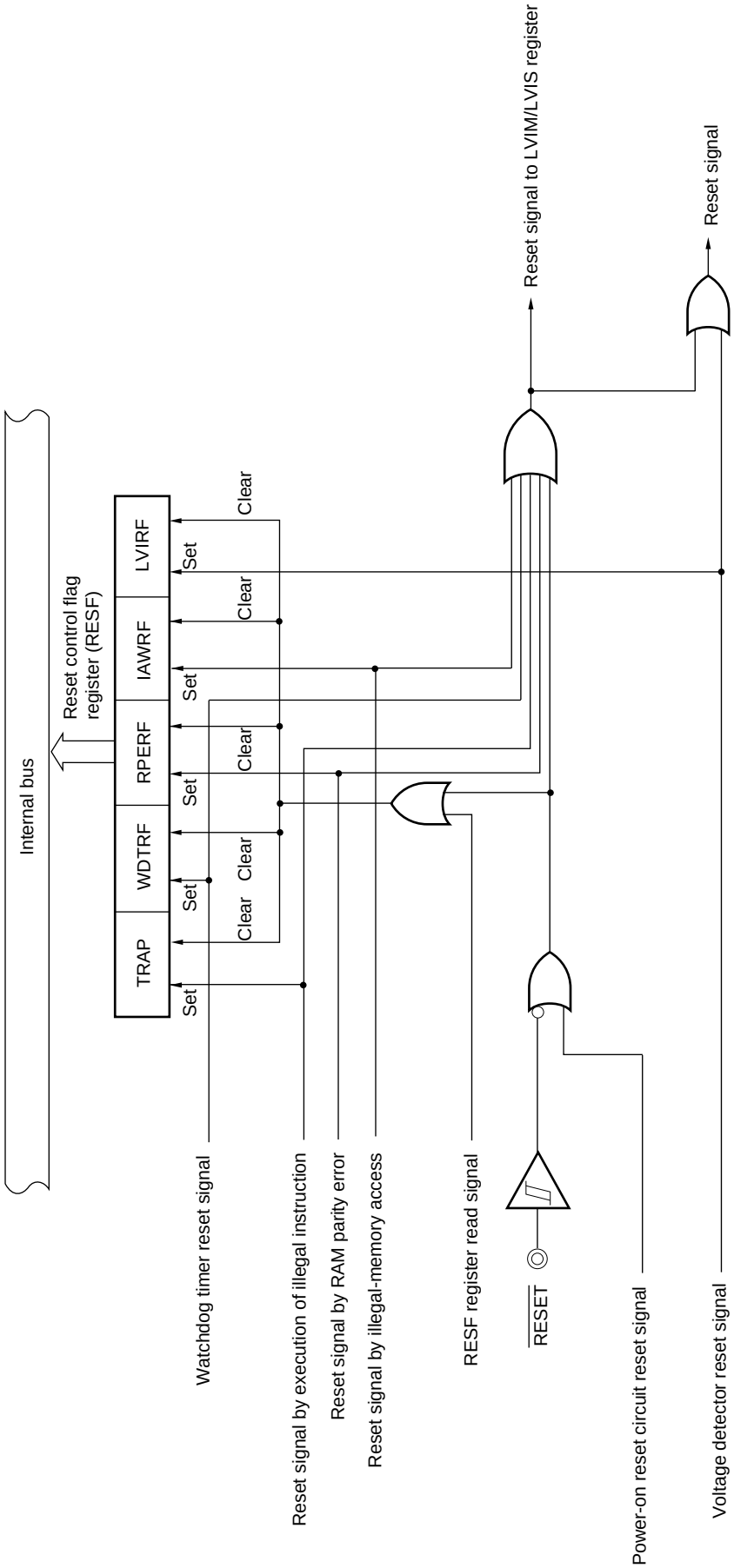
Note The illegal instruction is generated when instruction code FFH is executed.
Reset by the illegal instruction execution not issued by emulation with the in-circuit emulator or on-chip debug emulator.

Cautions 1. After power is turned on, P125 functions as the $\overline{\text{RESET}}$ pin (20- or 24-pin products only). Even if the internal reset signal is released by the power-on reset (POR), the reset state is retained while a low level is input to this pin.
When P125/KR1/SI01 is used, select the port function (PORTSELB = 0) by the option byte (000C1H) and release all reset sources.
2. For an external reset, input a low level for 10 μs or more to the $\overline{\text{RESET}}$ pin. To perform an external reset upon power application, input a low level to the $\overline{\text{RESET}}$ pin, turn power on, continue to input a low level on the pin for 10 us or more within the operating voltage range shown in 28.4 or 29.4 AC Characteristics, and then input a high level to the pin.
3. During reset input, the X1 clock, high-speed on-chip oscillator clock, and low-speed on-chip oscillator clock stop oscillating, and external main system clock input is invalid.
4. The port pins become the following state because each SFR and 2nd SFR are initialized after reset.
- **P40: High-impedance during the external reset period or reset period by the POR. High level during other types of reset or after receiving a reset signal (connected to the internal pull-up resistor).**
- **P125: High level during the reset period or after receiving a reset signal (connected to the internal pull-up resistor).**
- **Ports other than P40 and p125: High-impedance during the reset period or after receiving a reset signal.**

Remark V_{POR}: POR power supply rise detection voltage

Downloaded from Arrow.com.

Figure 18-1. Block Diagram of Reset Function



Caution An LVD circuit internal reset does not reset the LVD circuit.

Remarks 1. LVIM: Voltage detection register

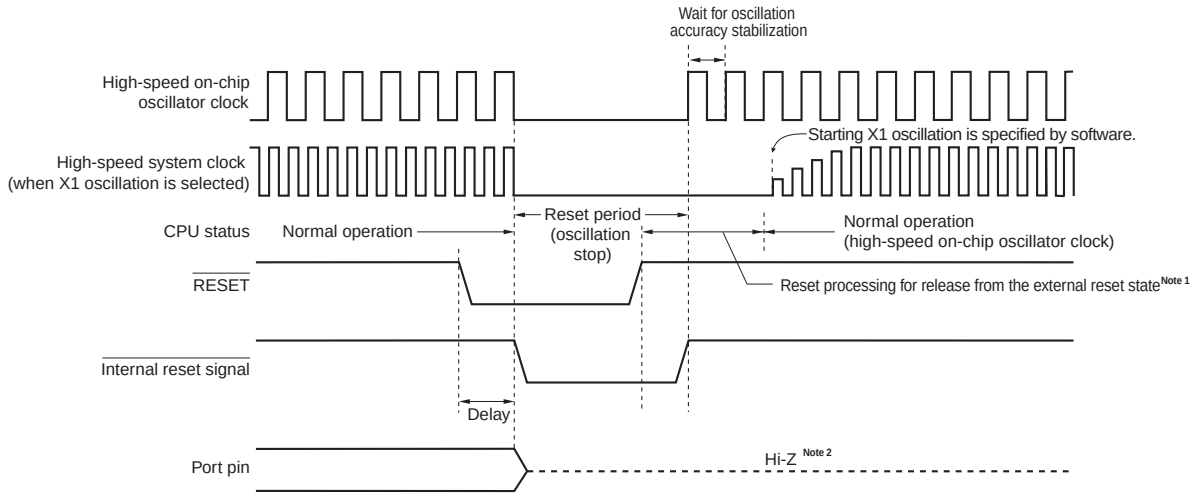
2. LVIS: Voltage detection level register

Downloaded from Arrow.com.

<R> 18.1 Timing of Reset Operation

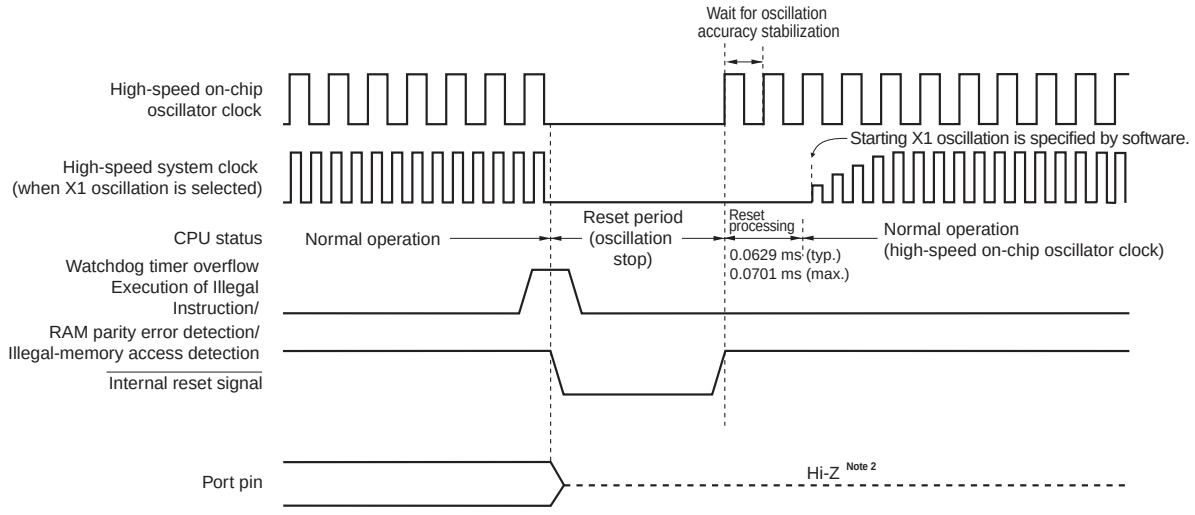
This LSI is reset by input of the low level on the $\overline{\text{RESET}}$ pin and released from the reset state by input of the high level on the $\overline{\text{RESET}}$ pin. After reset processing, execution of the program with the high-speed on-chip oscillator clock as the operating clock starts.

Figure 18-2. Timing of Reset by $\overline{\text{RESET}}$ Input



Release from the reset state is automatic in the case of a reset due to a watchdog timer overflow, execution of an illegal instruction, detection of a RAM parity error, or detection of illegal memory access. After reset processing, execution of the program with the high-speed on-chip oscillator clock as the operating clock starts.

Figure 18-3. Timing of Reset Due to Watchdog Timer Overflow, Execution of Illegal Instruction, RAM Parity Error, or Illegal Memory Access Overflow



(**Notes** are listed on the next page.)

Downloaded from Arrow.com.

Notes 1. Reset times (times for release from the external reset state)
After the first release of the POR: 0.672 ms (typ.), 0.832 ms (max.) when the LVD is in use.
0.399 ms (typ.), 0.519 ms (max.) when the LVD is off.
After the second release of the POR: 0.531 ms (typ.), 0.675 ms (max.) when the LVD is in use.
0.259 ms (typ.), 0.362 ms (max.) when the LVD is off.
After power is supplied, a voltage stabilization waiting time of about 0.99 ms (typ.) and up to 2.30 ms (max.) is required before reset processing starts after release of the external reset.

2. P40 and P125 become the following state.
- P40: High-impedance during the external reset period or reset period by the POR. High level during other types of reset or after receiving a reset signal (connected to the internal pull-up resistor).
- P125: High level during the reset period or after receiving a reset signal (connected to the internal pull-up resistor).

Reset by POR and LVD circuit supply voltage detection is automatically released when $V_{DD} \geq V_{POR}$ or $V_{DD} \geq V_{LVD}$ after the reset. After reset processing, execution of the program with the high-speed on-chip oscillator clock as the operating clock starts. For details, see **CHAPTER 19 POWER-ON-RESET CIRCUIT** or **CHAPTER 20 VOLTAGE DETECTOR**.

Downloaded from Arrow.com.

18.2 States of Operation During Reset Periods

Table 18-1 shows the states of operation during reset periods. Table 18-2 shows the states of the hardware after receiving a reset signal.

Table 18-1. Operation Statuses During Reset Period

Item			During Reset Period
System clock			Clock supply to the CPU is stopped.
	Main system clock	f_{IH}	Operation stopped
		f_X	Operation stopped (the X1 and X2 pins are input port mode)
		f_{EX}	Clock input invalid (the pin is input port mode)
	f_{IL}		Operation stopped
CPU			Operation stopped
Code flash memory			Operation stopped
Data flash memory			Operation stopped
RAM			Operation stopped
Port (latch)			High impedance Note
Timer array unit			Operation stopped
12-bit Interval timer			
Watchdog timer			
Clock output/buzzer output			
A/D converter			
Serial array unit (SAU)			
Serial interface (IICA)			
Multiplier & divider, multiply-accumulator			
DMA controller			
Power-on-reset function			Detection operation possible
Voltage detection function			Operation is possible in the case of an LVD reset and stopped in the case of other types of reset.
External interrupt			Operation stopped
Key interrupt function			
CRC operation function			
RAM parity error detection function			
RAM guard function			
SFR guard function			
Illegal-memory access detection function			

<R> **Note** P40 and P125 become the following state.

- P40: High-impedance during the external reset period or reset period by the POR. High level during other types of reset or after receiving a reset signal (connected to the internal pull-up resistor).
- P125: High level during the reset period or after receiving a reset signal (connected to the internal pull-up resistor).

Remark f_{IH}: High-speed on-chip oscillator clock
f_X: X1 oscillation clock
f_{EX}: External main system clock
f_{IL}: Low-speed on-chip oscillator clock

Downloaded from Arrow.com.

Table 18-2. State of Hardware After Receiving a Reset Signal

Hardware		After Reset Acknowledgment[Note]
Program counter (PC)		The contents of the reset vector table (0000H, 0001H) are set.
Stack pointer (SP)		Undefined
Program status word (PSW)		06H
RAM	Data memory	Undefined
	General-purpose registers	Undefined

Note During reset signal generation or oscillation stabilization time wait, only the PC contents among the hardware statuses become undefined. All other hardware statuses remain unchanged after reset.

<R> **Remark** For the state of the special function register (SFR) after receiving a reset signal, see **3.1.4 Special function register (SFR) area** and **3.1.5 Extended special function register (2nd SFR: 2nd Special Function Register) area**.

Downloaded from Arrow.com.

18.3 Register for Confirming Reset Source

18.3.1 Reset Control Flag Register (RESF)

Many internal reset generation sources exist in the RL78 microcontroller. The reset control flag register (RESF) is used to store which source has generated the reset request.

The RESF register can be read by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input, reset by power-on-reset (POR) circuit, and reading the RESF register clear TRAP, WDTRF, RPERF, IAWRF, and LVIRF flags.

Figure 18-4. Format of Reset Control Flag Register (RESF)

Address: FFFA8H After reset: 00H Note 1 R

Symbol	7	6	5	4	3	2	1	0
RESF	TRAP	0	0	WDTRF	0	RPERF	IAWRF	LVIRF

TRAP	Internal reset request by execution of illegal instruction Note 2
0	Internal reset request is not generated, or the RESF register is cleared.
1	Internal reset request is generated.

WDTRF	Internal reset request by watchdog timer (WDT)
0	Internal reset request is not generated, or the RESF register is cleared.
1	Internal reset request is generated.

RPERF	Internal reset request t by RAM parity
0	Internal reset request is not generated, or the RESF register is cleared.
1	Internal reset request is generated.

IAWRF	Internal reset request t by illegal-memory access
0	Internal reset request is not generated, or the RESF register is cleared.
1	Internal reset request is generated.

LVIRF	Internal reset request by voltage detector (LVD)
0	Internal reset request is not generated, or the RESF register is cleared.
1	Internal reset request is generated.

Notes 1. The value after reset varies depending on the reset source. See table 18-3.

2. The illegal instruction is generated when instruction code FFH is executed.
Reset by the illegal instruction execution not issued by emulation with the in-circuit emulator or on-chip debug emulator.

Cautions 1. Do not read data by a 1-bit memory manipulation instruction.

<R> **2. While RAM parity error resets are enabled (RPERDIS = 0), be sure to initialize RAM areas where data access is to proceed and the RAM area + 10 bytes when instructions are fetched from RAM areas, respectively. Reset signal generation sets RAM parity error resets to enabled (RPERDIS = 0). For details, see 21.3.2 RAM parity error detection function.**

Downloaded from Arrow.com.

The status of the RESF register when a reset request is generated is shown in Table 18-3.

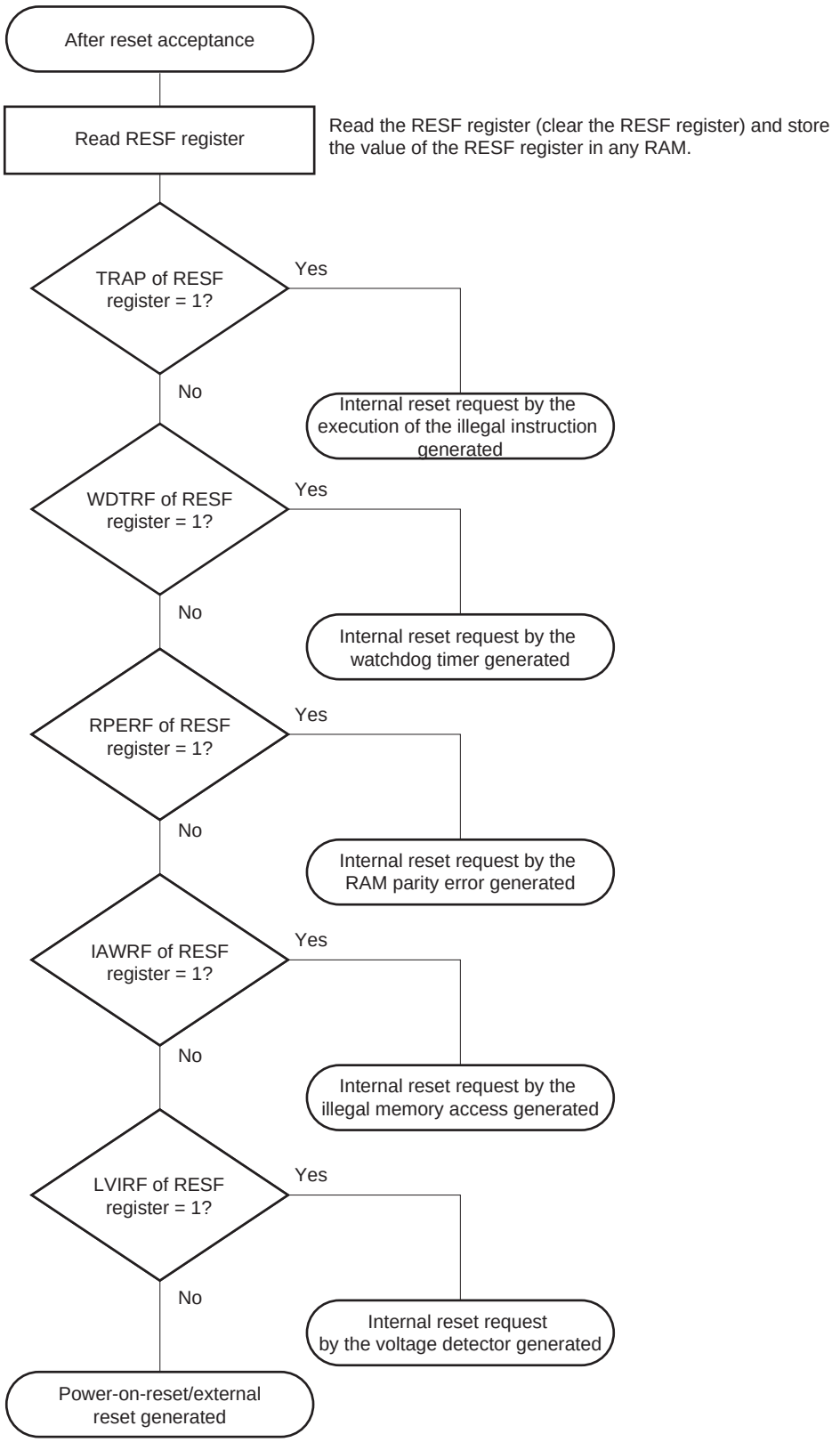
Table 18-3. RESF Register Status When Reset Request Is Generated

Flag \ Reset Source	$\overline{\text{RESET}}$ Input	Reset by POR	Reset by Execution of Illegal Instruction	Reset by WDT	Reset by RAM parity error	Reset by illegal-memory access	Reset by LVD
TRAP bit	Cleared (0)	Cleared (0)	Set (1)	Held	Held	Held	Held
WDTRF bit			Held	Set (1)			
RPERF bit				Held	Set (1)		
IAWRF bit					Held	Set (1)	
LVIRF bit						Held	Set (1)

The RESF register is automatically cleared when it is read by an 8-bit memory manipulation instruction. Figure 18-5 shows the procedure for checking a reset source.

Downloaded from Arrow.com.

Figure 18-5. Procedure for Checking Reset Source



Downloaded from Arrow.com.

CHAPTER 19 POWER-ON-RESET CIRCUIT

19.1 Functions of Power-on-reset Circuit

The power-on-reset circuit (POR) has the following functions.

<R> • Generates internal reset signal at power on.
The reset signal is released when the supply voltage (V_{DD}) exceeds the detection voltage (V_{POR}). Note that the reset state must be retained until the operating voltage becomes in the range defined in **28.4** or **29.4 AC Characteristics**. This is done by utilizing the voltage detection circuit or controlling the externally input reset signal.

<R> • Compares supply voltage (V_{DD}) and detection voltage (V_{PDR}), generates internal reset signal when $V_{DD} < V_{PDR}$. Note that, after power is supplied, this LSI should be placed in the STOP mode, or in the reset state by utilizing the voltage detection circuit or externally input reset signal, before the operation voltage falls below the range defined in **28.4** or **29.4 AC Characteristics**. When restarting the operation, make sure that the operation voltage has returned within the range of operation.

<R> **Caution If an internal reset signal is generated in the power-on-reset circuit, the reset control flag register (RESF) is cleared to 00H.**

Remarks 1. The RL78 microcontroller incorporates multiple hardware functions that generate an internal reset signal. A flag that indicates the reset source is located in the reset control flag register (RESF) for when an internal reset signal is generated by the watchdog timer (WDT), voltage-detector (LVD), illegal instruction execution, RAM parity error, or illegal-memory access. The RESF register is not cleared to 00H and the flag is set to 1 when an internal reset signal is generated by the watchdog timer (WDT), voltage-detector (LVD), illegal instruction execution, RAM parity error, or illegal-memory access.
For details of the RESF register, see **CHAPTER 18 RESET FUNCTION**.

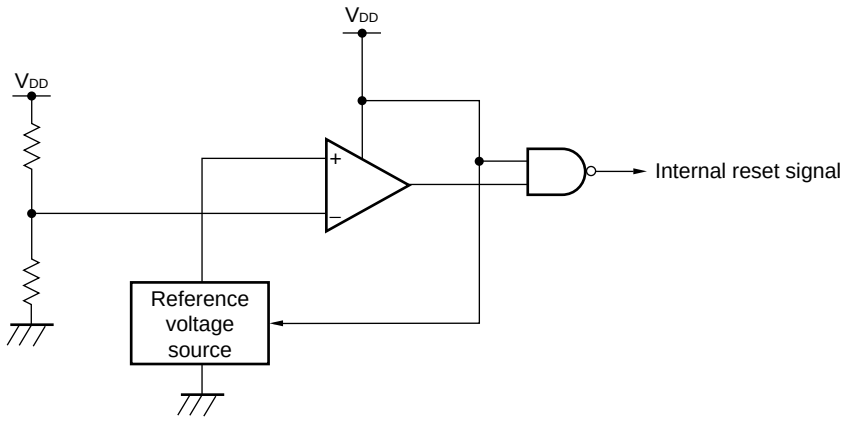
<R> **2.** V_{POR}: POR power supply rise detection voltage
V_{PDR}: POR power supply fall detection voltage
For details, see **28.6.3** or **29.6.3 POR circuit characteristics**.

Downloaded from Arrow.com.

19.2 Configuration of Power-on-reset Circuit

The block diagram of the power-on-reset circuit is shown in Figure 19-1.

Figure 19-1. Block Diagram of Power-on-reset Circuit



19.3 Operation of Power-on-reset Circuit

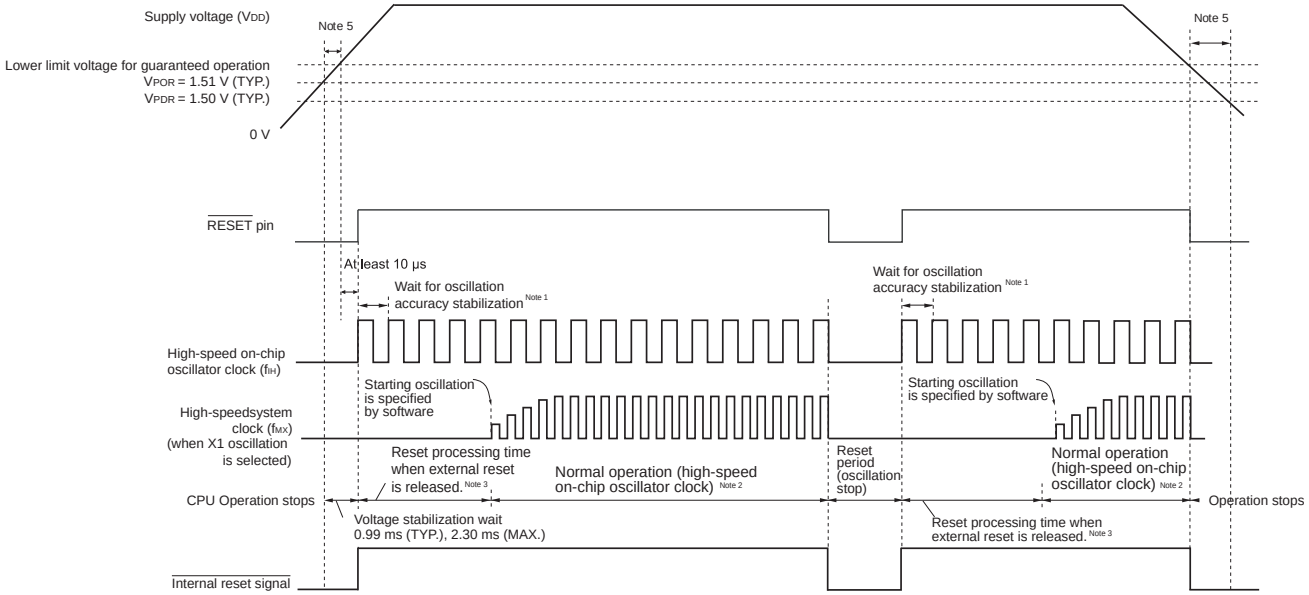
The timing of generation of the internal reset signal by the power-on-reset circuit and voltage detector is shown below.

Downloaded from Arrow.com.

Figure 19-2. Timing of Generation of Internal Reset Signal by Power-on-reset Circuit and Voltage Detector (1/3)

<R>

(1) When the external reset input via $\overline{\text{RESET}}$ pin is used



Notes 1. The internal reset processing time includes the oscillation accuracy stabilization time of the high-speed on-chip oscillator clock.

2. The high-speed on-chip oscillator clock can be switched to the high-speed system clock as the CPU clock. To use the X1 clock, use the oscillation stabilization time counter status register (OSTC) to confirm the lapse of the oscillation stabilization time.

3. The time until normal operation starts includes the following reset processing time when the external reset is released (after the first release of POR) after the $\overline{\text{RESET}}$ signal is driven high (1) as well as the voltage stabilization wait time after VPOR (1.51 V, typ.) is reached.
Reset processing time when the external reset is released is shown below.
After the first release of POR: 0.672 ms (typ.), 0.832 ms (max.) (when the LVD is in use)
0.399 ms (typ.), 0.519 ms (max.) (when the LVD is off)

4. Reset processing time when the external reset is released after the second release of POR is shown below.
After the second release of POR: 0.531 ms (typ.), 0.675 ms (max.) (when the LVD is in use)
0.259 ms (typ.), 0.362 ms (max.) (when the LVD is off)

5. After power is supplied, the reset state must be retained until the operating voltage becomes in the range defined in **28.4** or **29.4 AC Characteristics**. This is done by controlling the externally input reset signal.
After power supply is turned off, this LSI should be placed in the STOP mode, or in the reset state by utilizing the voltage detection circuit or externally input reset signal, before the voltage falls below the operating range. When restarting the operation, make sure that the operation voltage has returned within the range of operation.

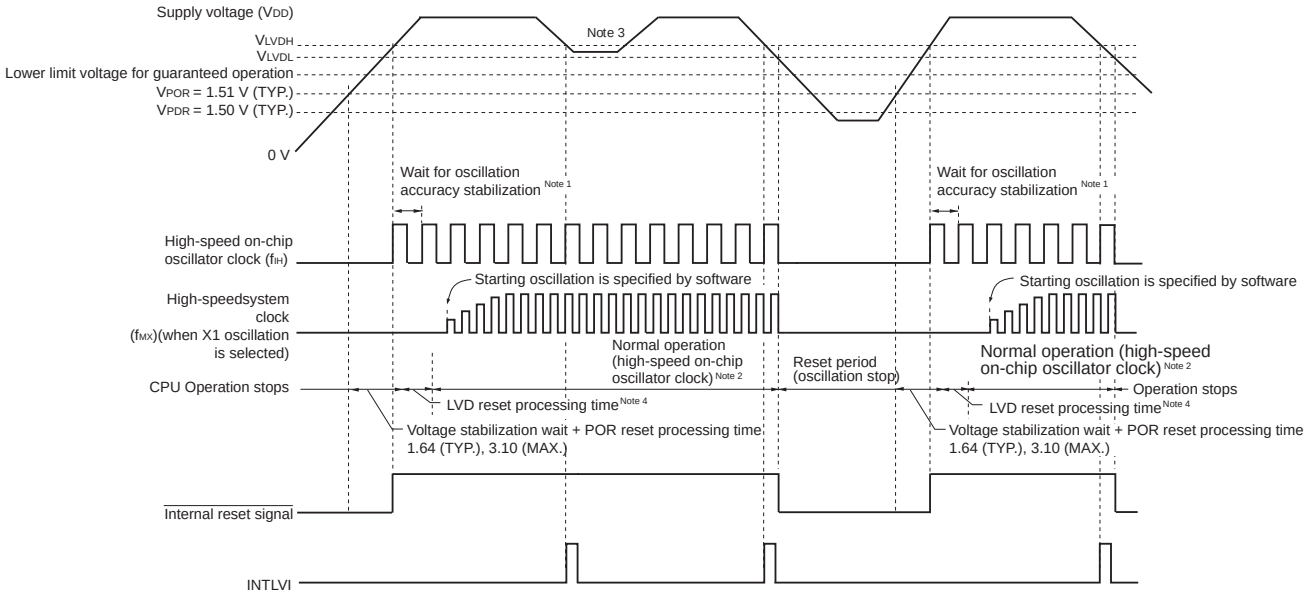
Remark VPOR: POR power supply rise detection voltage
VPDR: POR power supply fall detection voltage

Caution For power-on reset, be sure to use the externally input reset signal on the $\overline{\text{RESET}}$ pin when the LVD is off. For details, see CHAPTER 20 VOLTAGE DETECTOR.

Downloaded from Arrow.com.

Figure 19-2. Timing of Generation of Internal Reset Signal by Power-on-reset Circuit and Voltage Detector (2/3)

<R> **(2) LVD interrupt & reset mode (option byte 000C1H/LVIMDS1, LVIMDS0 = 1, 0)**



Notes 1. The internal reset processing time includes the oscillation accuracy stabilization time of the high-speed on-chip oscillator clock.

2. The high-speed on-chip oscillator clock can be switched to the high-speed system clock as the CPU clock. To use the X1 clock, use the oscillation stabilization time counter status register (OSTC) to confirm the lapse of the oscillation stabilization time.

3. After the interrupt request signal (INTLVI) is generated, the LVIL and LVIMD bits of the voltage detection level register (LVIS) are automatically set to 1. After INTLVI is generated, appropriate settings should be made according to **Figure 20-8 Processing Procedure After an Interrupt Is Generated** and **Figure 20-9 Initial Setting of Interrupt and Reset Mode**, taking into consideration that the supply voltage might return to the high voltage detection level (V_{LVDH}) or higher without falling below the low voltage detection level (V_{LVDL}).

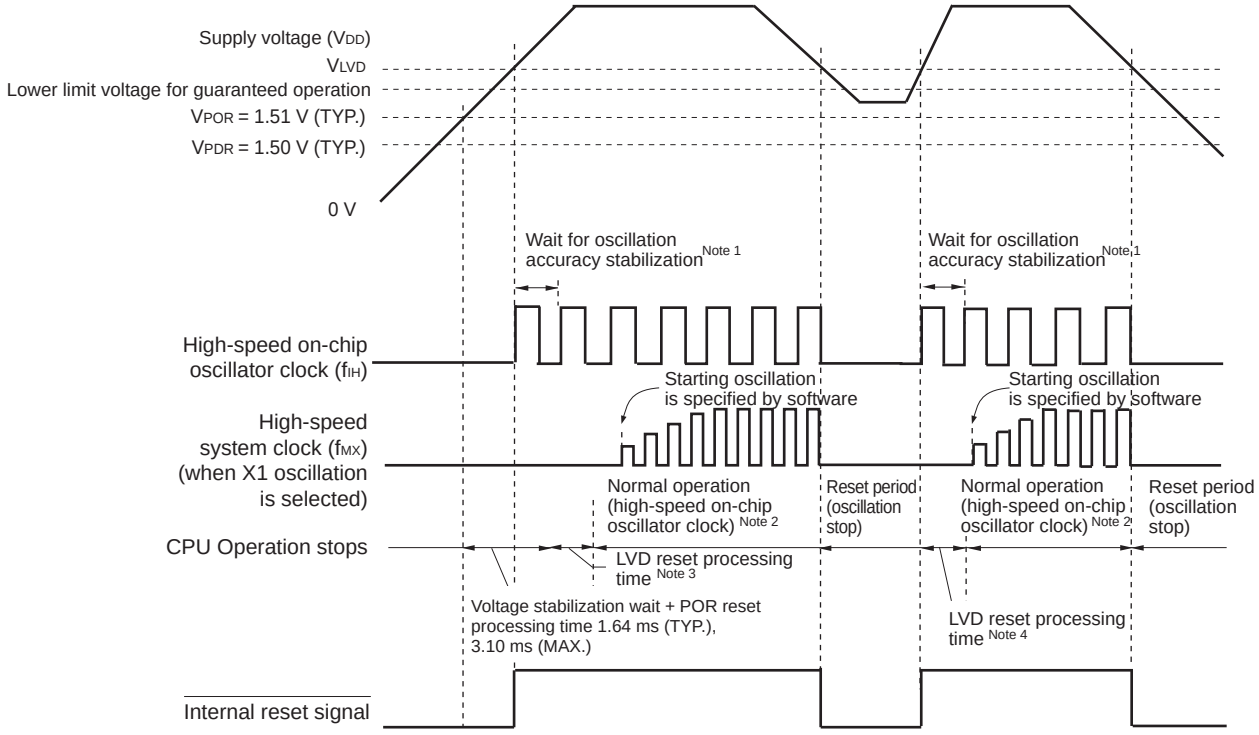
4. The time until normal operation starts includes the following LVD reset processing time after the LVD detection level (V_{LVDH}) is reached as well as the voltage stabilization wait + POR reset processing time after the VPOR (1.51 V, typ.) is reached.
LVD reset processing time: 0 ms to 0.0701 ms (max.)

Remark V_{LVDH}, V_{LVDL}: LVD detection voltage
V_{POR}: POR power supply rise detection voltage
V_{PDR}: POR power supply fall detection voltage

Downloaded from Arrow.com.

Figure 19-2. Timing of Generation of Internal Reset Signal by Power-on-reset Circuit and Voltage Detector (3/3)

<R> **(3) LVD reset mode (option byte 000C1H/LVIMDS1, LVIMDS0 = 1, 1)**



Notes

1. The internal reset processing time includes the oscillation accuracy stabilization time of the high-speed on-chip oscillator clock.
2. The high-speed on-chip oscillator clock can be switched to the high-speed system clock as the CPU clock. To use the X1 clock, use the oscillation stabilization time counter status register (OSTC) to confirm the lapse of the oscillation stabilization time.
3. The time until normal operation starts includes the following LVD reset processing time after the LVD detection level (VLVD) is reached as well as the voltage stabilization wait + POR reset processing time after the VPOR (1.51 V, typ.) is reached.
   LVD reset processing time: 0 ms to 0.0701 ms (max.)
4. When the power supply voltage is below the lower limit for operation and the power supply voltage is then restored after an internal reset is generated only by the voltage detector (LVD), the following LVD reset processing time is required after the LVD detection level (V_{LVD}) is reached.
   LVD reset processing time: 0.0629 ms (typ.), 0.0701 ms (max.)

Remarks

1. V_{LVDH}, V_{LVDL}: LVD detection voltage
   V_{POR}: POR power supply rise detection voltage
   V_{PDR}: POR power supply fall detection voltage
2. When the LVD interrupt mode is selected (option byte 000C1H: LVIMD1 = 0, LVIMD0 = 1), the time until normal operation starts after power is turned on is the same as the time specified in Note 3 of Figure 19-2 (3).

Downloaded from Arrow.com.

CHAPTER 20 VOLTAGE DETECTOR

20.1 Functions of Voltage Detector

The operation mode and detection voltages (V_{LVDH}, V_{LVDL}, V_{LVD}) for the voltage detector is set by using the option byte (000C1H).

The voltage detector (LVD) has the following functions.

- The LVD circuit compares the supply voltage (V_{DD}) with the detection voltage (V_{LVDH}, V_{LVDL}, V_{LVD}), and generates an internal reset or interrupt request signal.
- The detection level for the power supply detection voltage (V_{LVDH}, V_{LVDL}, V_{LVD}) can be selected by using the option byte as one of 12 levels (For details, see **CHAPTER 23 OPTION BYTE**).
- Operable in STOP mode.

After power is supplied, the reset state must be retained until the operating voltage becomes in the range defined in **28.4** or **29.4 AC Characteristics**. This is done by utilizing the voltage detection circuit or controlling the externally input reset signal. After the power supply is turned off, this LSI should be placed in the STOP mode, or placed in the reset state by utilizing the voltage detection circuit or controlling the externally input reset signal before the voltage falls below the operating range. The range of operating voltage varies with the setting of the user option byte (000C2H).

(a) Interrupt & reset mode (option byte LVIMDS1, LVIMDS0 = 1, 0)

The two detection voltages (V_{LVDH}, V_{LVDL}) are selected by the option byte 000C1H. The high-voltage detection level (V_{LVDH}) is used for releasing resets and generating interrupts. The low-voltage detection level (V_{LVDL}) is used for generating resets.

(b) Reset mode (option byte LVIMDS1, LVIMDS0 = 1, 1)

The detection voltage (V_{LVD}) selected by the option byte 000C1H is used for generating/releasing resets.

(c) Interrupt mode (option byte LVIMDS1, LVIMDS0 = 0, 1)

The detection voltage (V_{LVD}) selected by the option byte 000C1H is used for releasing resets/generating interrupts.

The reset and interrupt request signals are generated in each mode as follows.

<R>

Interrupt & reset mode (LVIMDS1, LVIMDS0 = 1, 0)	Reset mode (LVIMDS1, LVIMDS0 = 1, 1)	Interrupt mode (LVIMDS1, LVIMDS0 = 0, 1)
Generates an interrupt request signal by detecting $V_{DD} < V_{LVDH}$ when the operating voltage falls, and an internal reset by detecting $V_{DD} < V_{LVDL}$. Releases an internal reset by detecting $V_{DD} \geq V_{LVDH}$.	Releases an internal reset by detecting $V_{DD} \geq V_{LVD}$. Generates an interrupt request signal by detecting $V_{DD} < V_{LVD}$.	Releases an internal reset by detecting $V_{DD} \geq V_{LVD}$ at power on after the first release of the POR. Generates an interrupt request signal by detecting $V_{DD} < V_{LVD}$ or $V_{DD} \geq V_{LVD}$ at power on after the second release of the POR.

While the voltage detector is operating, whether the supply voltage is more than or less than the detection level can be checked by reading the voltage detection flag (LVIF: bit 0 of the voltage detection register (LVIM)).

Bit 0 (LVIRF) of the reset control flag register (RESF) is set to 1 if reset occurs. For details of the RESF register, see **CHAPTER 18 RESET FUNCTION**.

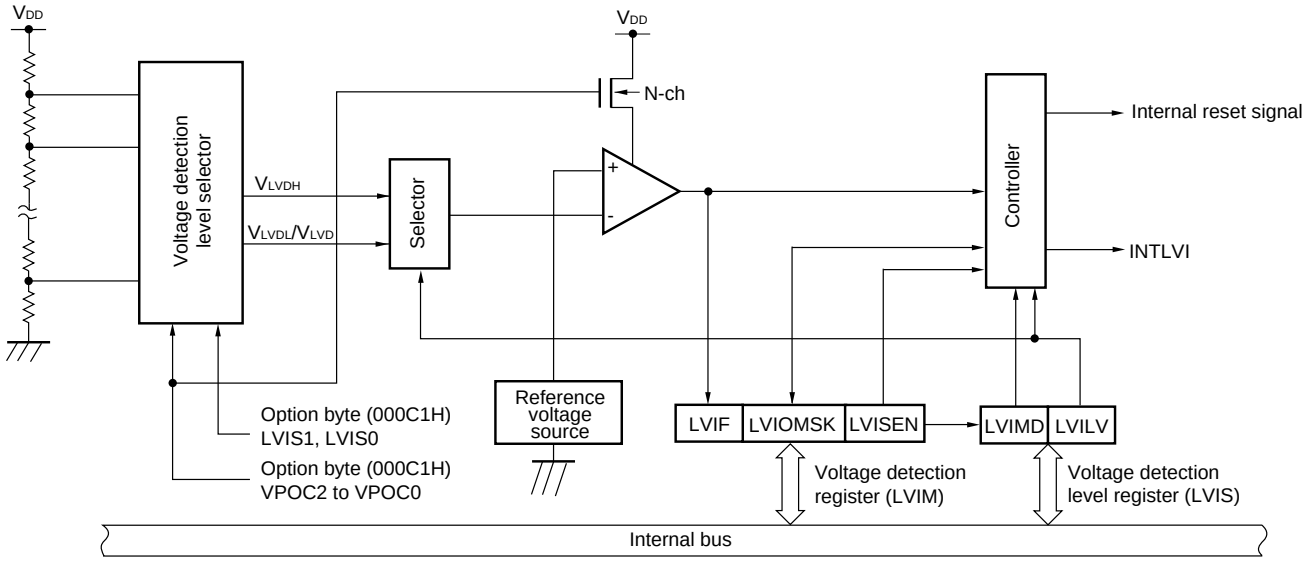
Downloaded from Arrow.com.

20.2 Configuration of Voltage Detector

The block diagram of the voltage detector is shown in Figure 20-1.

<R>

Figure 20-1. Block Diagram of Voltage Detector



20.3 Registers Controlling Voltage Detector

The voltage detector is controlled by the following registers.

- Voltage detection register (LVIM)
- Voltage detection level register (LVIS)

Downloaded from Arrow.com.

20.3.1 Voltage detection register (LVIM)

This register is used to specify whether to enable or disable rewriting the voltage detection level register (LVIS), as well as to check the LVD output mask status.

This register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

<R> **Figure 20-2. Format of Voltage Detection Register (LVIM)**

Address: FFFA9H After reset: Note 1 R/W Note 2

Symbol	<7>	6	5	4	3	2	<1>	<0>
LVIM	LVISEN Note 3	0	0	0	0	0	LVIOMSK	LVIF

LVISEN Note 3	Specification of whether to enable or disable rewriting the voltage detection level register (LVIS)
0	Disabling of rewriting the LVIS register (LVIOMSK = 0 (Mask of LVD output is invalid))
1	Enabling of rewriting the LVIS register (LVIOMSK = 1 (Mask of LVD output is valid))

LVIOMSK	Mask status flag of LVD output
0	Mask of LVD output is invalid
1	Mask of LVD output is valid Note 4

LVIF	Voltage detection flag
0	Supply voltage (V_{DD}) ≥ detection voltage (V_{LVD}), or when LVD is off
1	Supply voltage (V_{DD}) < detection voltage (V_{LVD})

Notes 1. The reset value changes depending on the reset source.
If the LVIS register is reset by LVD, it is not reset but holds the current value. The value of this LVISEN is reset to “0” if a reset other than by LVD is effected.

2. Bits 0 and 1 are read-only.

3. LVISEN and LVIOMSK can only be set in the interrupt & reset mode (option byte LVIMDS1, LVIMDS0 = 1, 0). Do not change the initial value in other modes.

4. LVIOMSK bit is only automatically set to “1” when the interrupt & reset mode is selected (option byte LVIMDS1, LVIMDS0 = 1, 0) and reset or interrupt by LVD is masked.
- Period during LVISEN = 1
- Waiting period from the time when LVD interrupt is generated until LVD detection voltage becomes stable
- Waiting period from the time when the value of LVILV bit changes until LVD detection voltage becomes stable

Downloaded from Arrow.com.

20.3.2 Voltage detection level register (LVIS)

This register selects the voltage detection level.

This register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation input sets this register to 00H/01H/81H Note 1.

Figure 20-3. Format of Voltage Detection Level Select Register (LVIS)

Address: FFFAAH After reset: 00H/01H/81H Note 1 R/W

Symbol	<7>	6	5	4	3	2	1	<0>
LVIS	LVIMD	0	0	0	0	0	0	LVILV

LVIMD Note 2	Operation mode of voltage detection
0	Interrupt mode
1	Reset mode

LVILV Note 2	LVD detection level
0	High-voltage detection level (V_{LVDH})
1	Low-voltage detection level (V_{LVDL} or V_{LVD})

Notes 1. The reset value changes depending on the reset source and the setting of the option byte. This register is not cleared (00H) by LVD reset.
The generation of reset signal other than an LVD reset sets as follows.
- When option byte LVIMDS1, LVIMDS0 = 1, 0: 00H
- When option byte LVIMDS1, LVIMDS0 = 1, 1: 81H
- When option byte LVIMDS1, LVIMDS0 = 0, 1: 01H

<R> **2.** Writing “0” can only be allowed in the interrupt & reset mode (option byte LVIMDS1, LVIMDS0 = 1, 0). Do not set LVIMD and LVILV in other cases. The value is switched automatically when reset or interrupt is generated in the interrupt & reset mode.

<R> **Cautions 1. Rewrite the value of the LVIS register according to Figures 20-8 and 20-9.**

<R> **2. Specify the LVD operation mode and detection voltage (V_{LVDH}, V_{LVDL}, V_{LVD}) of each mode by using the option byte 000C1H. Figure 20-4 shows the format of the user option byte (000C1H). For details about the option byte, see CHAPTER 23 OPTION BYTE.**

Downloaded from Arrow.com.

Figure 20-4. LVD Operation Mode and Detection Voltage Settings for User Option Byte (000C1H) (1/2)

Address: 000C1H

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	PORTSELB	LVIS1	LVIS0	LVIMDS1	LVIMDS0

- LVD setting (interrupt & reset mode)

Detection voltage			Option byte Setting Value						
V_{LVDH}		V_{LVD}	VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting	
Rising edge	Falling edge	Falling edge						LVIMDS1	LVIMDS0
1.98 V	1.94 V	1.84 V	0	0	1	1	0	1	0
2.09 V	2.04 V					0	1		
3.13 V	3.06 V					0	0		
2.61 V	2.55 V	2.45 V		1	0	1	0		
2.71 V	2.65 V					0	1		
3.75 V	3.67 V					0	0		
2.92 V	2.86 V	2.75 V		1	1	1	0		
3.02 V	2.96 V					0	1		
4.06 V	3.98 V					0	0		
—			Setting of values other than above is prohibited.						

- LVD setting (reset mode)

Detection voltage		Option byte Setting Value						
V_{LVD}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting	
Rising edge	Falling edge						LVIMDS1	LVIMDS0
1.88 V	1.84 V	0	0	1	1	1	1	1
1.98 V	1.94 V		0	1	1	0		
2.09 V	2.04 V		0	1	0	1		
2.50 V	2.45 V		1	0	1	1		
2.61 V	2.55 V		1	0	1	0		
2.71 V	2.65 V		1	0	0	1		
2.81 V	2.75 V		1	1	1	1		
2.92 V	2.86 V		1	1	1	0		
3.02 V	2.96 V		1	1	0	1		
3.13 V	3.06 V		0	1	0	0		
3.75 V	3.67 V		1	0	0	0		
4.06 V	3.98 V		1	1	0	0		
—		Setting of values other than above is prohibited.						

Remarks 1. For details on the LVD circuit, see **CHAPTER 20 VOLTAGE DETECTOR**.

2. The detection voltage is a TYP. value. For details, see **28.6.4** or **29.6.4 LVD circuit characteristics**.

(**Cautions** are listed on the next page.)

Downloaded from Arrow.com.

Figure 20-4. LVD Operation Mode and Detection Voltage Settings for User Option Byte (000C1H) (2/2)

Address: 000C1H

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	PORTSELB	LVIS1	LVIS0	LVIMDS1	LVIMDS0

- LVD setting (interrupt mode)

Detection voltage		Option byte Setting Value						
V_{LVD}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting	
Rising edge	Falling edge						LVIMDS1	LVIMDS0
1.88 V	1.84 V	0	0	1	1	1	0	1
1.98 V	1.94 V		0	1	1	0		
2.09 V	2.04 V		0	1	0	1		
2.50 V	2.45 V		1	0	1	1		
2.61 V	2.55 V		1	0	1	0		
2.71 V	2.65 V		1	0	0	1		
2.81 V	2.75 V		1	1	1	1		
2.92 V	2.86 V		1	1	1	0		
3.02 V	2.96 V		1	1	0	1		
3.13 V	3.06 V		0	1	0	0		
3.75 V	3.67 V		1	0	0	0		
4.06 V	3.98 V		1	1	0	0		
—		Setting of values other than above is prohibited.						

- LVD off (use of external reset input via $\overline{\text{RESET}}$ pin)

Detection voltage		Option byte Setting Value						
V_{LVDH}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting	
Rising edge	Falling edge						LVIMDS1	LVIMDS0
–	–	1	×	×	×	×	×	1
—		Setting of values other than above is prohibited.						

Cautions 1. Set bit 4 to 1.

<R> **2. After power is supplied, the reset state must be retained until the operating voltage becomes in the range defined in 28.4 or 29.4 AC Characteristics. This is done by utilizing the voltage detection circuit or controlling the externally input reset signal. After the power supply is turned off, this LSI should be placed in the STOP mode, or placed in the reset state by utilizing the voltage detection circuit or controlling the externally input reset signal, before the voltage falls below the operating range. The range of operating voltage varies with the setting of the user option byte (000C2H).**

Remarks 1. ×: don't care

<R> **2.** For details on the LVD circuit, see **CHAPTER 20 VOLTAGE DETECTOR**.

<R> **3.** The detection voltage is a TYP. value. For details, see **28.6.4** or **29.6.4 LVD circuit characteristics**.

Downloaded from Arrow.com.

20.4 Operation of Voltage Detector

<R> 20.4.1 When used as reset mode

Specify the operation mode (the reset mode (LVIMDS1, LVIMDS0 = 1, 1)) and the detection voltage (V_{LVD}) by using the option byte 000C1H.

The operation is started in the following initial setting state when the reset mode is set.

- Bit 7 (LVISEN) of the voltage detection register (LVIM) is set to 0 (disable rewriting of voltage detection level register (LVIS))
- The initial value of the voltage detection level select register (LVIS) is set to 81H.
  Bit 7 (LVIMD) is 1 (reset mode).
  Bit 0 (LVILV) is 1 (low-voltage detection level: V_{LVD}).

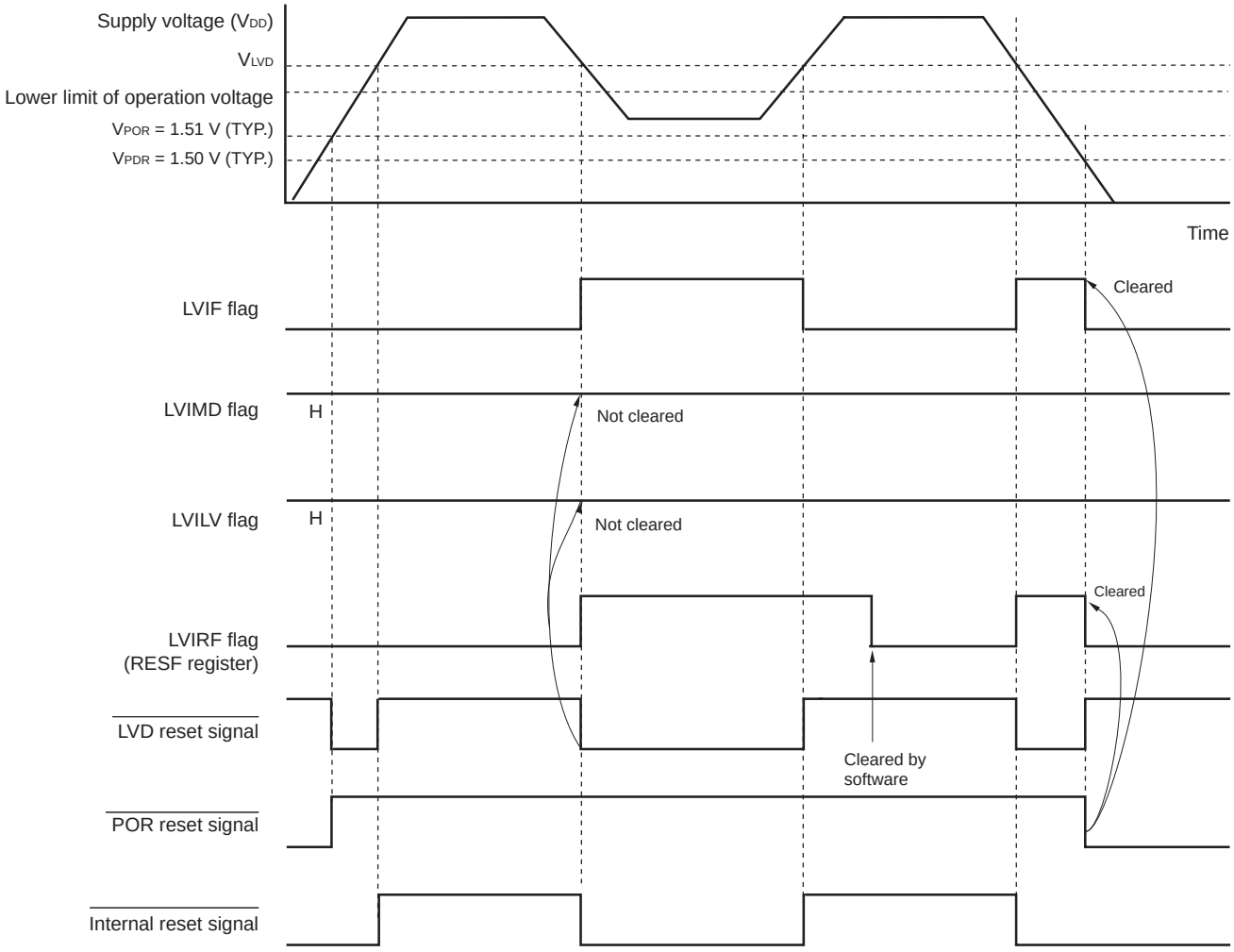
- Operation in LVD reset mode
  In the reset mode (option byte LVIMDS1, LVIMDS0 = 1, 1), the state of an internal reset by LVD is retained until the supply voltage (V_{DD}) exceeds the voltage detection level (V_{LVD}) after power is supplied. The internal reset is released when the supply voltage (V_{DD}) exceeds the voltage detection level (V_{LVD}).
  At the fall of the operating voltage, an internal reset by LVD is generated when the supply voltage (V_{DD}) falls below the voltage detection level (V_{LVD}).

Figure 20-5 shows the timing of the internal reset signal generated in the LVD reset mode.

Downloaded from Arrow.com.

<R>

Figure 20-5. Timing of Voltage Detector Internal Reset Signal Generation (Option Byte LVIMDS1, LVIMDS0 = 1, 1)



Remark VPOR: POR power supply rise detection voltage
VPDR: POR power supply fall detection voltage

Downloaded from Arrow.com.

<R> **20.4.2 When used as interrupt mode**

Specify the operation mode (the interrupt mode (LVIMDS1, LVIMDS0 = 0, 1)) and the detection voltage (V_{LVD}) by using the option byte 000C1H.

The operation is started in the following initial setting state when the interrupt mode is set.

- Bit 7 (LVISEN) of the voltage detection register (LVIM) is set to 0 (disable rewriting of voltage detection level register (LVIS))
- The initial value of the voltage detection level select register (LVIS) is set to 01H.
  Bit 7 (LVIMD) is 0 (interrupt mode).
  Bit 0 (LVILV) is 1 (low-voltage detection level: V_{LVD}).

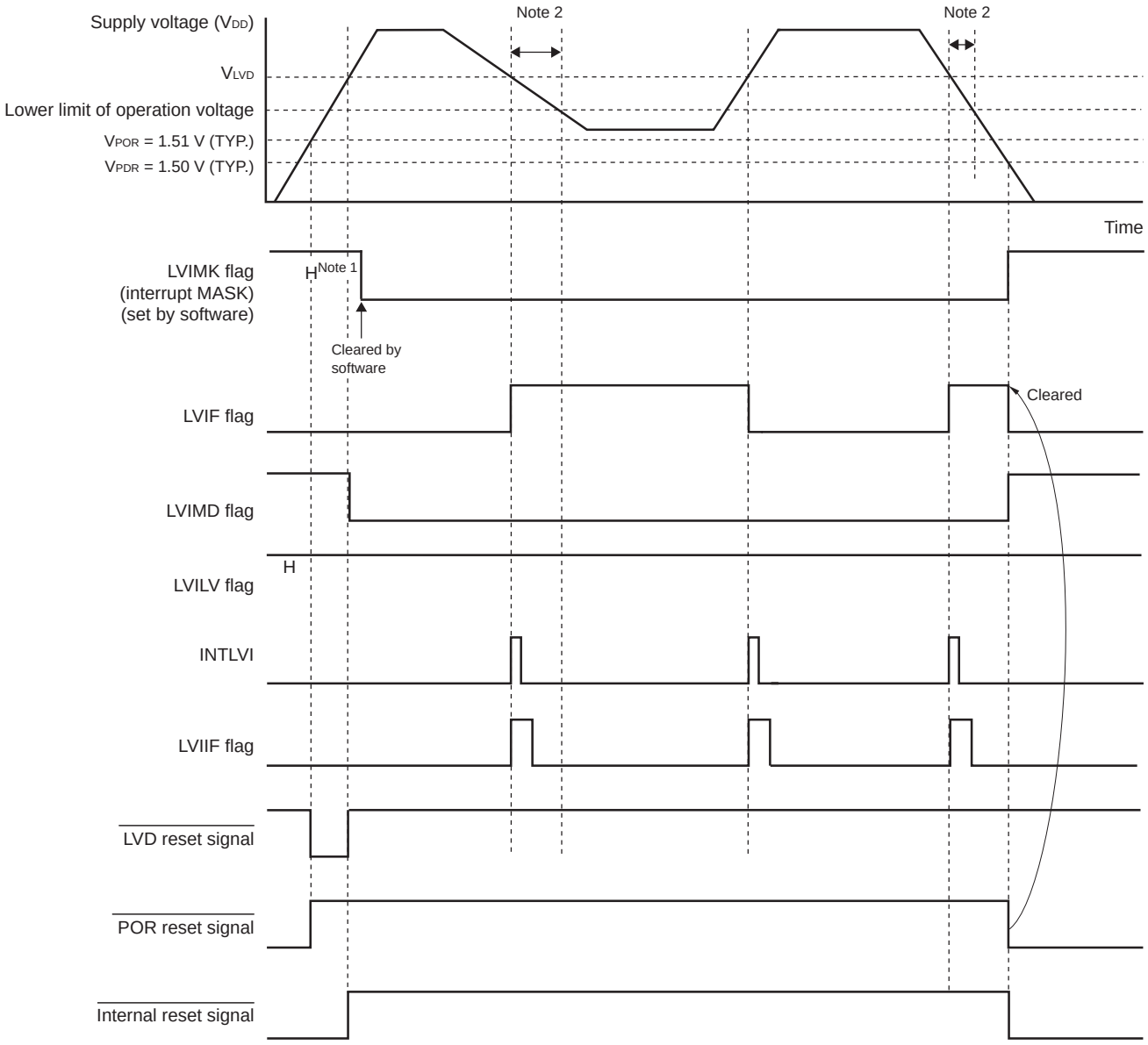
- Operation in LVD interrupt mode
  In the interrupt mode (option byte LVIMDS1, LVIMDS0 = 0, 1), the state of an internal reset by LVD is retained until the supply voltage (V_{DD}) exceeds the voltage detection level (V_{LVD}) after power is supplied (after the first release of the POR). The internal reset is released when the supply voltage (V_{DD}) exceeds the voltage detection level (V_{LVD}).
  An interrupt request signal by LVD (INTLVD) is generated, when the supply voltage (V_{DD}) falls below the voltage detection level (V_{LVD}) or when the supply voltage (V_{DD}) exceeds the voltage detection level (V_{LVD}) after the second release of the POR. When the voltage falls, this LSI should be placed in the STOP mode, or placed in the reset state by controlling the externally input reset signal, before the voltage falls below the operating voltage range defined in **28.4** or **29.4 AC characteristics**. When restarting the operation, make sure that the operation voltage has returned within the range of operation.

Figure 20-6 shows the timing of the interrupt request signal generated in the LVD interrupt mode.

Downloaded from Arrow.com.

<R>

Figure 20-6. Timing of Voltage Detector Interrupt Request Signal Generation (Option Byte LVIMDS1, LVIMDS0 = 0, 1)



Notes 1. The LVIMK flag is set to “1” by reset signal generation.

<R> **2.** When the voltage falls, this LSI should be placed in the STOP mode, or placed in the reset state by controlling the externally input reset signal, before the voltage falls below the operating voltage range defined in **28.4** or **29.4 AC characteristics**. When restarting the operation, make sure that the operation voltage has returned within the range of operation.

Remark VPOR: POR power supply rise detection voltage
VPDR: POR power supply fall detection voltage

Downloaded from Arrow.com.

<R> **20.4.3 When used as interrupt and reset mode**

Specify the operation mode (the interrupt & reset (LVIMDS1, LVIMDS0 = 1, 0)) and the detection voltage (V_{LVDH}, V_{LVDL}) by using the option byte 000C1H.

The operation is started in the following initial setting state when the interrupt & reset mode is set.

- Bit 7 (LVISEN) of the voltage detection register (LVIM) is set to 0 (disable rewriting of voltage detection level register (LVIS))
- The initial value of the voltage detection level select register (LVIS) is set to 00H.
  Bit 7 (LVIMD) is 0 (interrupt mode).
  Bit 0 (LVILV) is 0 (high-voltage detection level: V_{LVDH}).

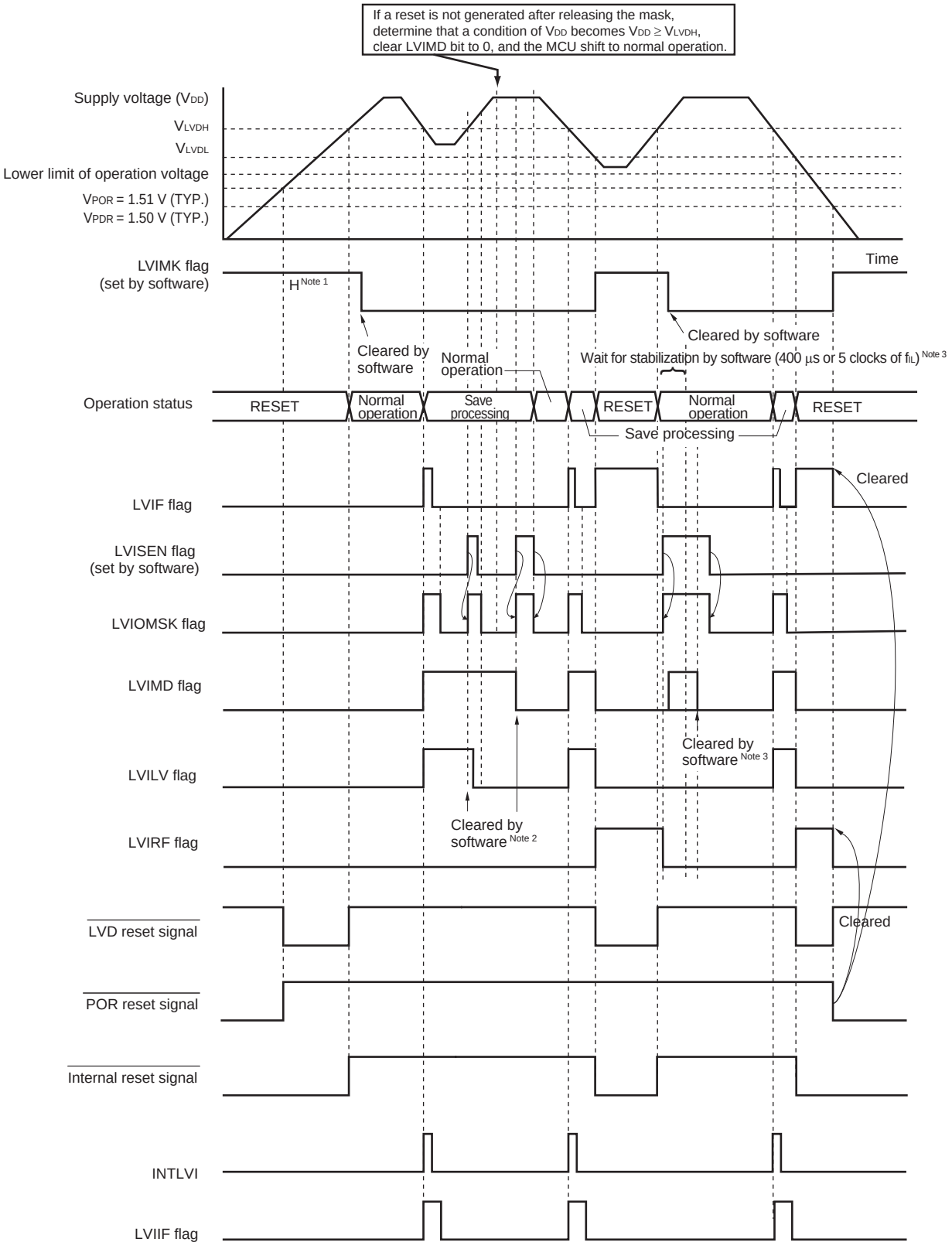
- Operation in LVD interrupt & reset mode
  In the interrupt & reset mode (option byte LVIMDS1, LVIMDS0 = 1, 0), the state of an internal reset by LVD is retained until the supply voltage (V_{DD}) exceeds the high-voltage detection level (V_{LVDH}) after power is supplied. The internal reset is released when the supply voltage (V_{DD}) exceeds the high-voltage detection level (V_{LVDH}). An interrupt request signal by LVD (INTLVD) is generated and arbitrary save processing is performed when the supply voltage (V_{DD}) falls below the high-voltage detection level (V_{LVDH}). After that, an internal reset by LVD is generated when the supply voltage (V_{DD}) falls below the low-voltage detection level (V_{LVDL}). After INTLVD is generated, an interrupt request signal is not generated even if the supply voltage becomes equal to or higher than the high-voltage detection voltage (V_{LVDH}) without falling below the low-voltage detection voltage (V_{LVDL}).
  To use the LVD reset & interrupt mode, perform the processing according to **Figure 20-8 Processing Procedure After an Interrupt Is Generated** and **Figure 20-9 Initial Setting of Interrupt and Reset Mode**.

Figure 20-7 shows the timing of the internal reset signal and interrupt request signal generated in the LVD interrupt & reset mode.

Downloaded from Arrow.com.

<R>

Figure 20-7. Timing of Voltage Detector Reset Signal and Interrupt Signal Generation (Option Byte LVIMDS1, LVIMDS0 = 1, 0) (1/2)



(**Notes** and **Remark** are listed on the next page.)

Downloaded from Arrow.com.

Notes 1. The LVIMK flag is set to “1” by reset signal generation.

2. After an interrupt is generated, perform the processing according to **Figure 20-8 Processing Procedure After an Interrupt Is Generated**.

3. After a reset is released, perform the processing according to **Figure 20-9 Initial Setting of Interrupt and Reset Mode**.

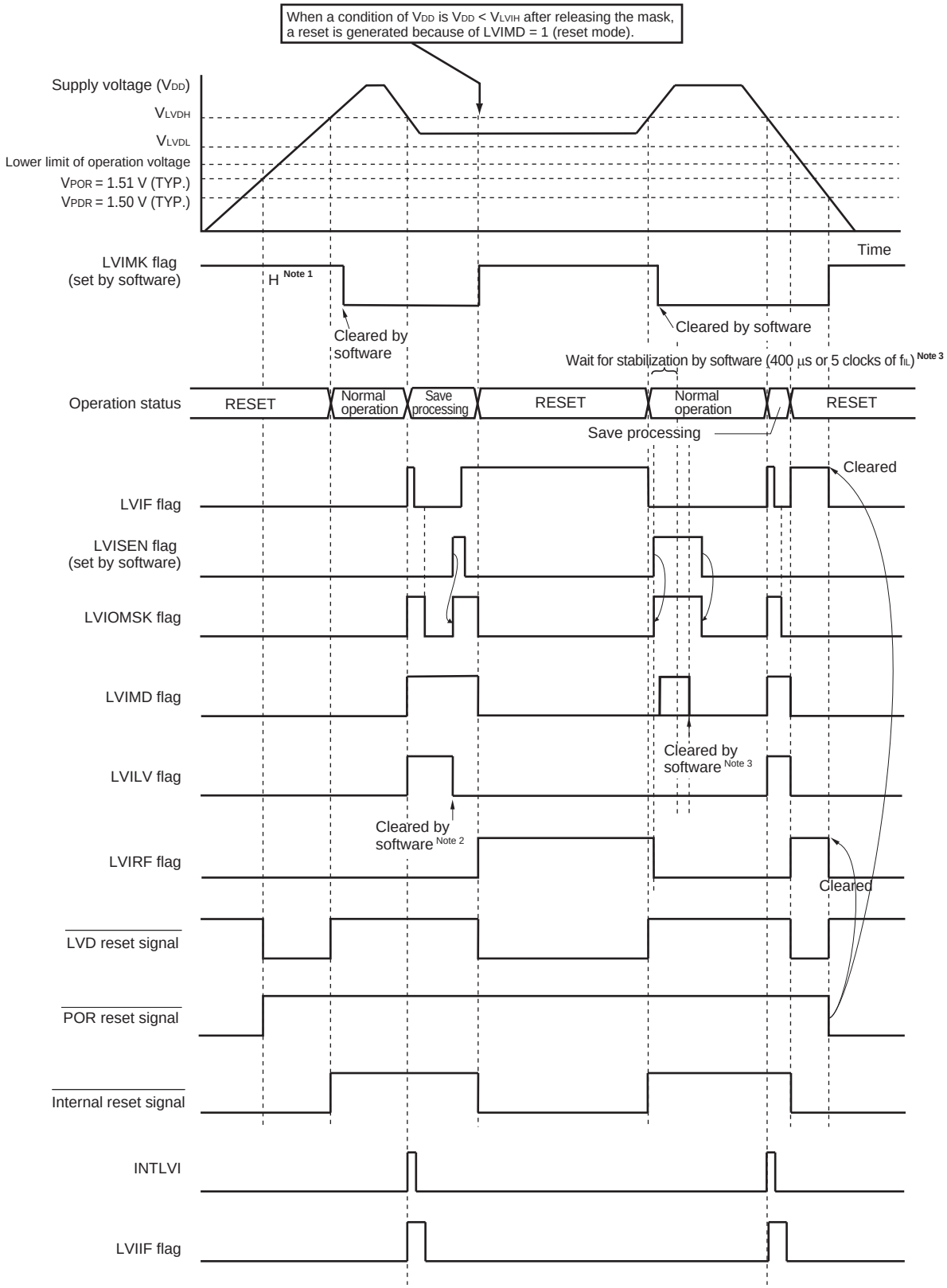
Remark V_{POR}: POR power supply rise detection voltage

V_{PDR}: POR power supply fall detection voltage

Downloaded from Arrow.com.

<R>

Figure 20-7. Timing of Voltage Detector Reset Signal and Interrupt Signal Generation (Option Byte LVIMDS1, LVIMDS0 = 1, 0) (2/2)



(**Notes** and **Remark** are listed on the next page.)

Downloaded from Arrow.com.

Notes 1. The LVIMK flag is set to “1” by reset signal generation.

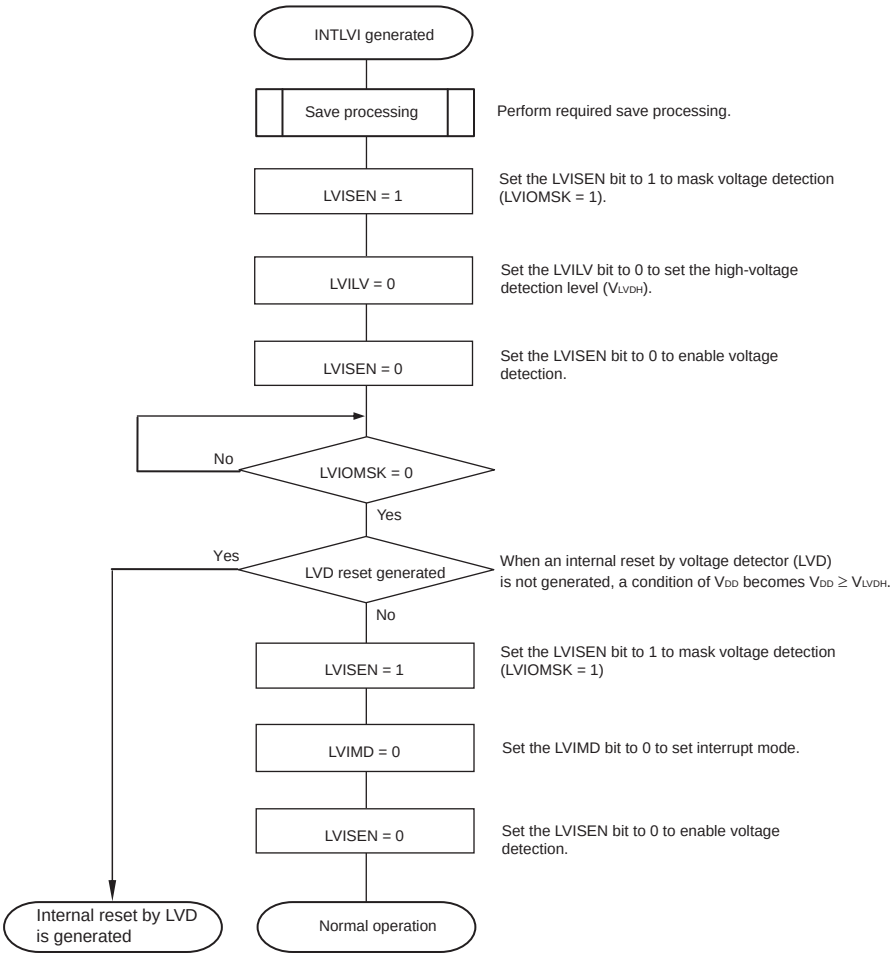
2. After an interrupt is generated, perform the processing according to **Figure 20-8 Processing Procedure After an Interrupt Is Generated** in interrupt and reset mode.

3. After a reset is released, perform the processing according to **Figure 20-9 Initial Setting of Interrupt and Reset Mode** in interrupt and reset mode.

Remark V_{POR}: POR power supply rise detection voltage
V_{PDR}: POR power supply fall detection voltage

<R>

Figure 20-8. Processing Procedure After an Interrupt Is Generated

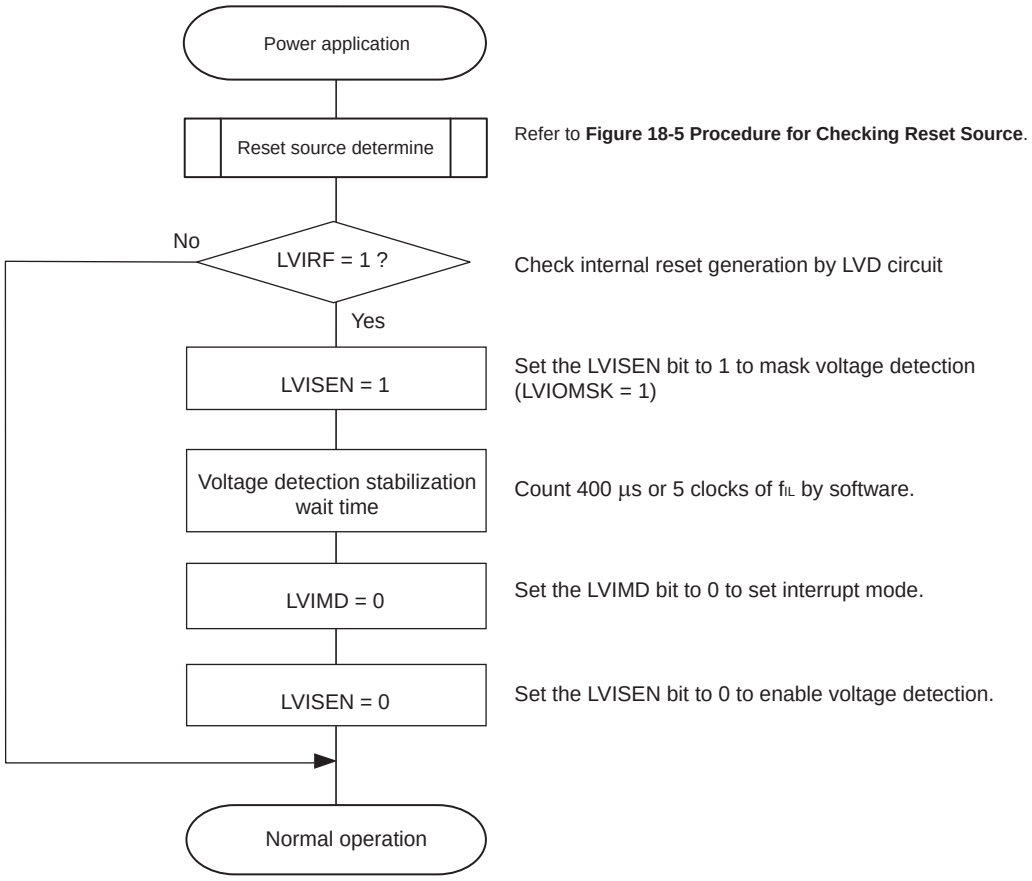


Downloaded from Arrow.com.

When setting an interrupt and reset mode (LVIMDS1, LVIMDS0 = 1, 0), voltage detection stabilization wait time for 400 μs or 5 clocks of f_{IL} is necessary after LVD reset is released (LVIRF = 1). After waiting until voltage detection stabilizes, (0) clear the LVIMD bit for initialization. While voltage detection stabilization wait time is being counted and when the LVIMD bit is rewritten, set LVISEN to 1 to mask a reset or interrupt generation by LVD.

Figure 20-9. shows the procedure for initial setting of interrupt and reset mode.

Figure 20-9 Initial Setting of Interrupt and Reset Mode



Remark f_{IL}: Low-speed on-chip oscillator clock frequency

Downloaded from Arrow.com.

20.5 Cautions for Voltage Detector

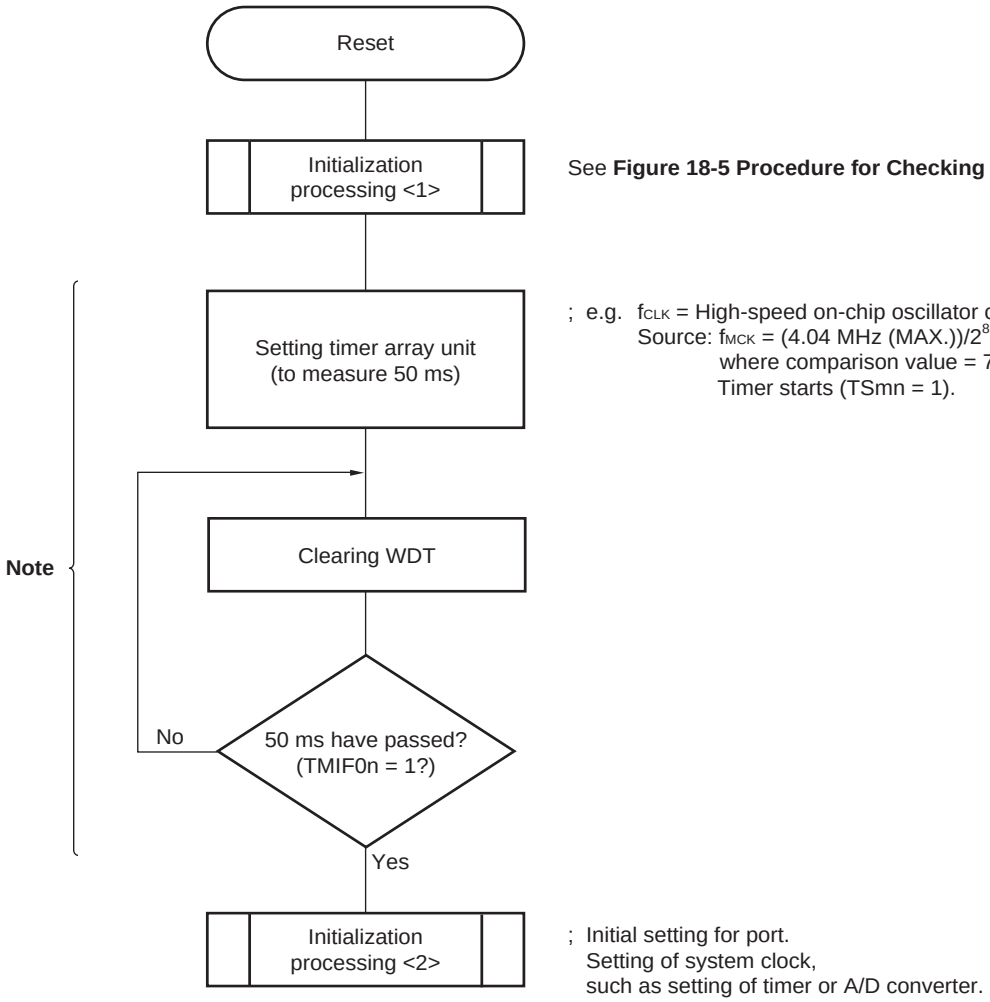
<R> **(1) Voltage fluctuation when power is supplied**

In a system where the supply voltage (V_{DD}) fluctuates for a certain period in the vicinity of the LVD detection voltage, the system may be repeatedly reset and released from the reset status. In this case, the time from release of reset to the start of the operation of the microcontroller can be arbitrarily set by taking the following action.

<Action>

After releasing the reset signal, wait for the supply voltage fluctuation period of each system by means of a software counter that uses a timer, and then initialize the ports.

<R> **Figure 20-10. Example of Software Processing If Supply Voltage Fluctuation is 50 ms or Less in Vicinity of LVD Detection Voltage**



Note If reset is generated again during this period, initialization processing <2> is not started.

Remark n = 0 to 3 (20- and 24-pin products)
n = 0 to 7 (30-pin products)

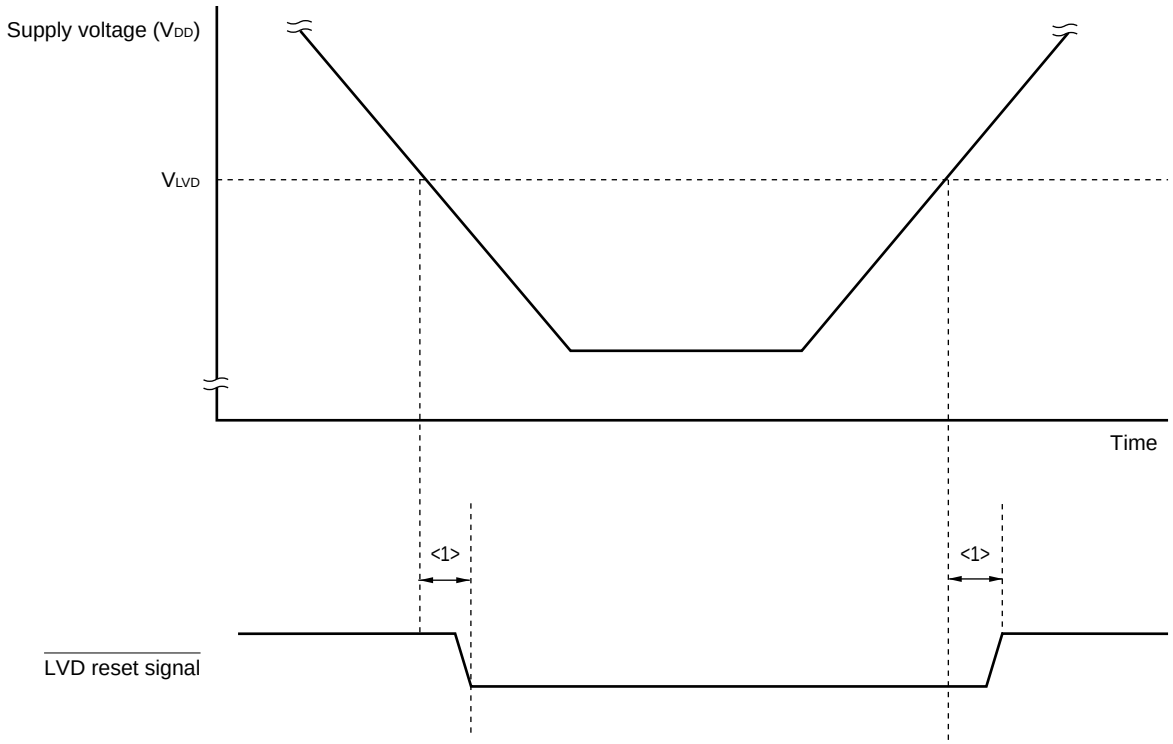
Downloaded from Arrow.com.

(2) Delay from the time LVD reset source is generated until the time LVD reset has been generated or released

There is some delay from the time supply voltage (V_{DD}) < LVD detection voltage (V_{LVD}) until the time LVD reset has been generated.

In the same way, there is also some delay from the time LVD detection voltage (V_{LVD}) ≤ supply voltage (V_{DD}) until the time LVD reset has been released (see **Figure 20-11**).

Figure 20-11. Delay from the time LVD reset source is generated until the time LVD reset has been generated or released



<1>: Detection delay (300 μs (MAX.))

<R> **(3) Power on when LVD is off**

Use the external rest input via the $\overline{\text{RESET}}$ pin when the LVD is off.

For an external reset, input a low level for 10 µs or more to the $\overline{\text{RESET}}$ pin. To perform an external reset upon power application, input a low level to the $\overline{\text{RESET}}$ pin, turn power on, continue to input a low level to the pin for 10 µs or more within the operating voltage range shown in **28.4** or **29.4 AC Characteristics**, and then input a high level to the pin.

<R> **(4) Operating voltage fall when LVD is off or LVD interrupt mode is selected**

When the operating voltage falls with the LVD is off or with the LVD interrupt mode is selected, this LSI should be placed in the STOP mode, or placed in the reset state by controlling the externally input reset signal, before the voltage falls below the operating voltage range defined in **28.4** or **29.4 AC characteristics**. When restarting the operation, make sure that the operation voltage has returned within the range of operation.

Downloaded from Arrow.com.

CHAPTER 21 SAFETY FUNCTIONS

21.1 Overview of Safety Functions

The following safety functions are provided in the RL78/G12 to comply with the IEC60730 and IEC61508 safety standards.

These functions enable the microcontroller to self-diagnose abnormalities and stop operating if an abnormality is detected.

(1) Flash memory CRC operation function

This detects data errors in the flash memory by performing CRC operations.

This can be used for checking various data in addition to the code flash memory area while the CPU is running.

This function is available in the R5F102 products.

(2) RAM parity error detection function

This detects parity errors when reading RAM is read.

(3) RAM guard function

This prevents RAM data from being rewritten when the CPU freezes.

This function is available in the R5F102 products.

(4) SFR guard function

This prevents SFRs from being rewritten when the CPU freezes.

This function is available in the R5F102 products.

(5) Invalid memory access detection function

This detects illegal accesses to invalid memory areas (such as areas where no memory is allocated and areas to which access is restricted).

<R> **(6) Frequency detection function**

This function allows a self-check of the CPU/peripheral hardware clock frequencies using the timer array unit.

<R> **(7) A/D test function**

This is used to perform a self-check of the A/D converter by performing A/D conversion of the A/D converter's positive and negative reference voltages, analog input channel (ANI), temperature sensor output voltage, and internal reference voltage.

Remark For usage examples of the safety functions complying with the IEC60730 safety standards, refer to the RL78 MCU series IEC60730/60335 application notes (R01AN0749).

Downloaded from Arrow.com.

21.2 Registers Used by Safety Functions

The safety functions use the following registers for each function.

Register	Each Function of Safety Function
• CRC input register (CRCIN) Note • CRC data register (CRCD) Note	CRC operation function (general-purpose CRC)
• RAM parity error control register (RPECTL)	RAM parity error detection function
• Invalid memory access detection control register (IAWCTL)	RAM guard function
	SFR guard function
	Invalid memory access detection function
• Timer input select register 0 (TIS0)	Frequency detection function
• A/D test register (ADTES)	A/D test function

Note Only in the R5F102 products.

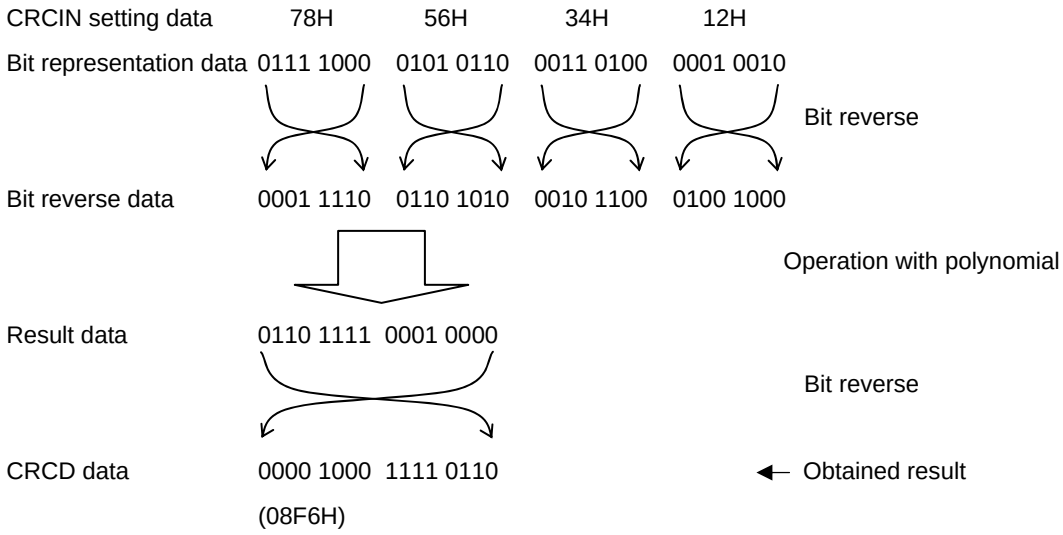
21.3 Operation of Safety Functions

21.3.1 CRC operation function (general-purpose CRC)

In order to guarantee safety during operation, the IEC61508 standard mandates the checking of data even while the CPU is operating.

In the RL78/G12, a general CRC operation can be executed as a peripheral function while the CPU is operating. The general CRC can be used for checking various data in addition to the code flash memory area. The data to be checked can be specified by using software (a user-created program). CRC calculation function in the HALT mode can be used only during the DMA transmission

The CRC generator polynomial used is "$X^{16} + X^{12} + X^5 + 1$" of CRC-16-CCITT. The data to be input is inverted in bit order and then calculated to allow for LSB-first communication. For example, if the data 12345678H is sent from the LSB, values are written to the CRCIN register in the order of 78H, 56H, 34H, and 12H, enabling a value of 08F6H to be obtained from the CRCD register. This is the result obtained by executing a CRC operation on the bit rows shown below, which consist of the data 12345678H inverted in bit order.



Caution Because the debugger rewrites the software break setting line to a break instruction during program execution, the CRC operation result differs if a software break is set in the CRC operation target area.

Downloaded from Arrow.com.

21.3.1.1 CRC input register (CRCIN)

CRCIN register is an 8-bit register that is used to set the CRC operation data.

The possible setting range is 00H to FFH.

The CRCIN register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 21-1. Format of CRC Input Register (CRCIN)

Address: FFFACH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CRCIN								

Bits 7 to 0	Function
00H to FFH	Data input.

21.3.1.2 CRC data register (CRCD)

This register is used to store the CRC operation result.

The setting range is 0000H to FFFFH.

After 1 clock of CPU/peripheral hardware clock (f_{CLK}) has elapsed from the time CRCIN register is written, the CRC operation result is stored to the CRCD register.

The CRCD register can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 21-2. Format of CRC Data Register (CRCD)

Address: F02FAH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CRCD																

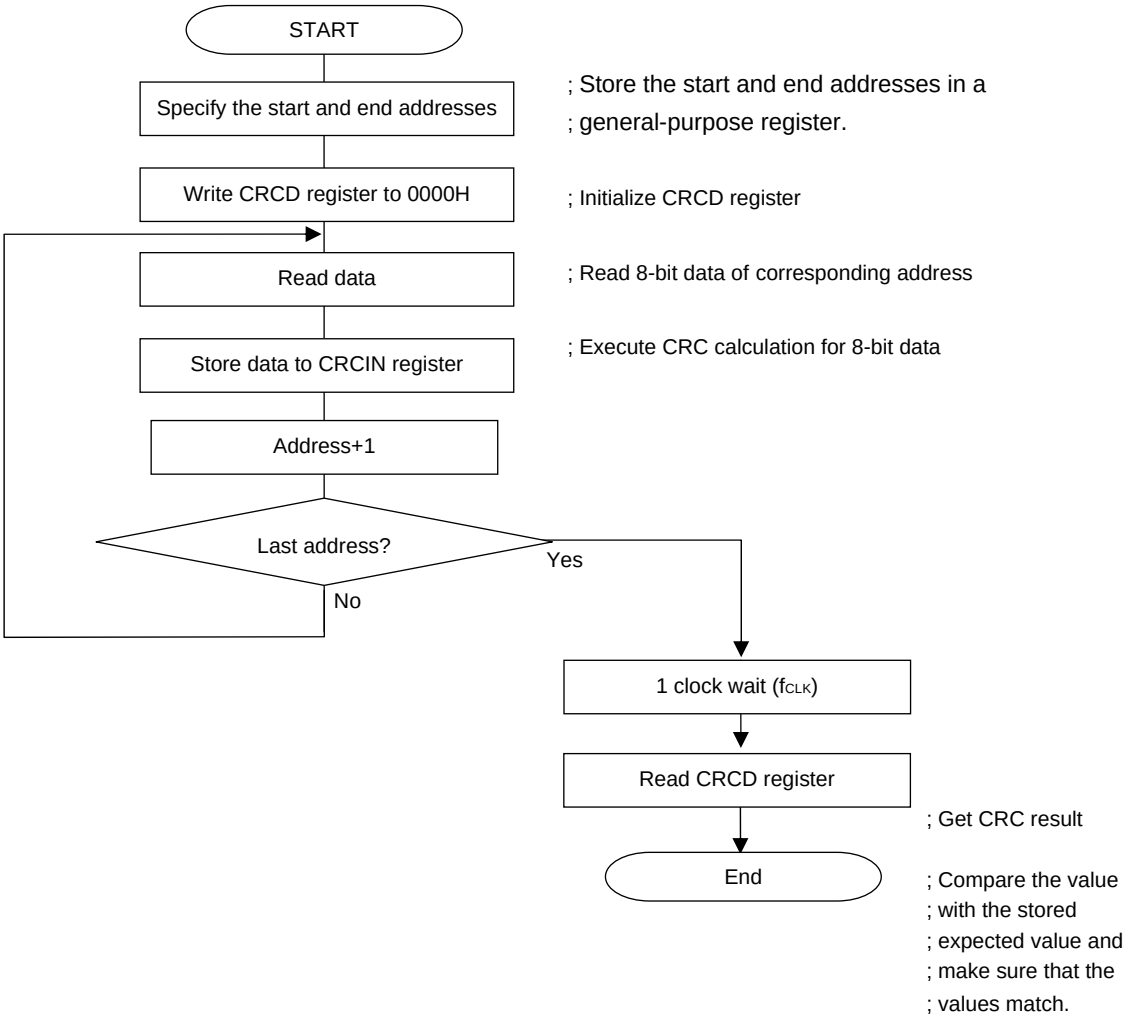
Cautions 1. Read the value written to CRCD register before writing to CRCIN register.

2. If conflict between writing and storing operation result to CRCD register occurs, the writing is ignored.

Downloaded from Arrow.com.

<Operation flow>

Figure 21-3. CRC Operation Function (General-Purpose CRC)



Downloaded from Arrow.com.

21.3.2 RAM parity error detection function

The IEC60730 standard mandates the checking of RAM data. A single-bit parity bit is therefore added to all 8-bit data in the RL78/G12's RAM. By using this RAM parity error detection function, the parity bit is appended when data is written, and the parity is checked when the data is read. This function can also be used to trigger a reset when a parity error occurs.

21.3.2.1 RAM parity error control register (RPECTL)

This register is used to control parity error generation check bit and reset generation due to parity errors.

The RPECTL register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 21-4. Format of RAM Parity Error Control Register (RPECTL)

Address: F00F5H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	<0>
RPECTL	RPERDIS	0	0	0	0	0	0	RPEF

RPERDIS	Parity error reset mask flag
0	Enable parity error resets.
1	Disable parity error resets.

RPEF	Parity error status flag
0	No parity error has occurred.
1	A parity error has occurred.

<R> **Caution The parity bit is appended when data is written, and the parity is checked when the data is read.**
Therefore, while RAM parity error resets are enabled (RPERDIS = 0), be sure to initialize RAM areas where data access is to proceed before reading data.
The RL78's CPU executes look-ahead due to the pipeline operation, the CPU might read an uninitialized RAM area that is allocated beyond the RAM used, which causes a RAM parity error.
Therefore, while RAM parity error resets are enabled (RPERDIS = 0), be sure to initialize the RAM area + 10 bytes when instructions are fetched from RAM areas.

Remarks 1. The parity error reset is enabled by default (RPERDIS = 0).

<R> **2.** Even if the parity error reset is disabled (RPERDIS = 1), the RPEF flag will be set (1) if a parity error occurs. If parity error resets are enabled (RPERDIS = 0) with RPEF set to 1, a parity error reset is generated when the RPERDIS bit is cleared to 0.

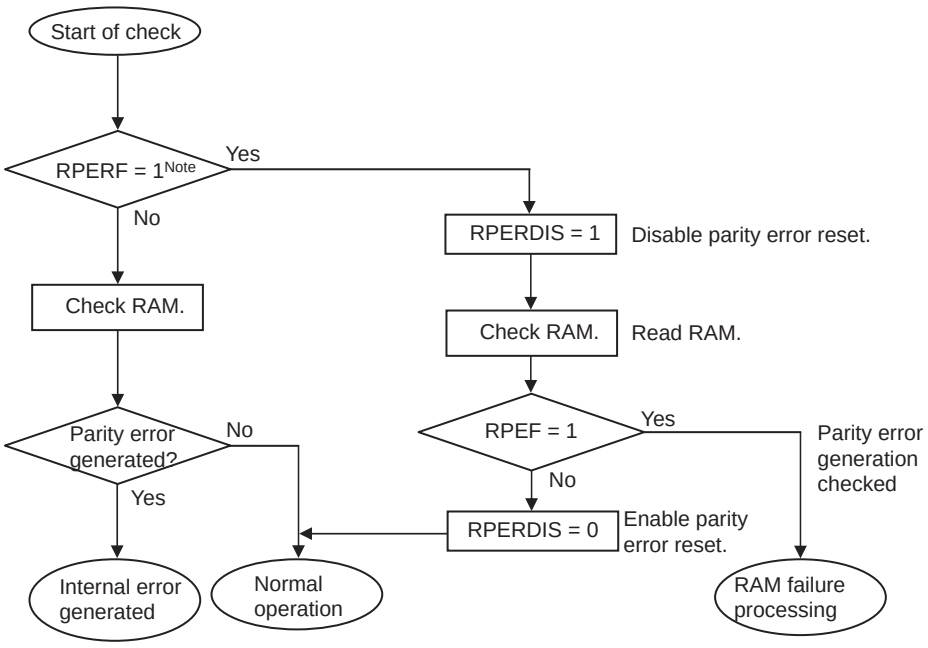
3. The RPEF flag in the RPECTL register is set (1) when the RAM parity error occurs and cleared (0) by writing 0 to it or by any reset source. When RPEF = 1, the value is retained even if RAM for which no parity error has occurred is read.

<R> **4.** The general registers are not included for RAM parity error detection.

Downloaded from Arrow.com.

<R>

Figure 21-5. Flowchart of RAM Parity Check



Note To check internal reset status using a RAM parity error, see **CHAPTER 18 RESET FUNCTION**.

Downloaded from Arrow.com.

21.3.3 RAM guard function

In order to guarantee safety during operation, the IEC61508 standard mandates that important data stored in the RAM be protected, even if the CPU freezes.

This RAM guard function is used to protect data in the specified memory space.

If the RAM guard function is specified, writing to the specified RAM space is disabled, but reading from the space can be carried out as usual.

21.3.3.1 Invalid memory access detection control register (IAWCTL)

This register is used to control the detection of invalid memory access and RAM/SFR guard function.

GRAM1 and GRAM0 bits are used in RAM guard function.

The IAWCTL register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 21-6. Format of Invalid Memory Access Detection Control Register (IAWCTL)

Address: F0078H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IAWCTL	IAWEN	0	GRAM1	GRAM0	0	GPORT	GINT	GCSC

GRAM1	GRAM0	RAM guard space Note
0	0	Disabled. RAM can be written to.
0	1	The 128 bytes starting at the beginning RAM address
1	0	The 256 bytes starting at the beginning RAM address
1	1	The 512 bytes starting at the beginning RAM address (setting prohibited for R5F10266, R5F10366)

Note The RAM start address differs depending on the size of the RAM provided with the product (refer to **Figure 21-8**).
Furthermore, the general-purpose register area (FFEE0H to FFEFFH) is not guarded.

Downloaded from Arrow.com.

21.3.4 SFR guard function

In order to guarantee safety during operation, the IEC61508 standard mandates that important SFRs be protected from being overwritten, even if the CPU freezes.

This SFR guard function is used to protect data in the control registers used by the port function, interrupt function, clock control function, voltage detection function, and RAM parity error detection function.

If the SFR guard function is specified, writing to the specified SFRs is disabled, but reading from the SFRs can be carried out as usual.

21.3.4.1 Invalid memory access detection control register (IAWCTL)

This register is used to control the detection of invalid memory access and RAM/SFR guard function.

GPORT, GINT and GCSC bits are used in SFR guard function.

The IAWCTL register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 21-7. Format of Invalid Memory Access Detection Control Register (IAWCTL)

Address: F0078H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IAWCTL	IAWEN	0	GRAM1	GRAM0	0	GPORT	GINT	GCSC

GPORT	Control registers of port function guard
0	Disabled. Control registers of port function can be read or written to.
1	Enabled. Writing to control registers of port function is disabled. Reading is enabled. [Guarded SFR] PMxx, PUxx, PIMxx, POMxx, PMCxx, ADPC, PIOR Note

GINT	Registers of interrupt function guard
0	Disabled. Registers of interrupt function can be read or written to.
1	Enabled. Writing to registers of interrupt function is disabled. Reading is enabled. [Guarded SFR] IFxx, MKxx, PRxx, EGPx, EGNx

GCSC	Control registers of clock control function, voltage detector and RAM parity error detection function guard
0	Disabled. Control registers of clock control function, voltage detector and RAM parity error detection function can be read or written to.
1	Enabled. Writing to control registers of clock control function, voltage detector and RAM parity error detection function is disabled. Reading is enabled. [Guarded SFR] CMC, CSC, OSTS, CKC, PERx, OSMC, LVIM, LVIS, RPECTL

Note Pxx (Port register) is not guarded.

Downloaded from Arrow.com.

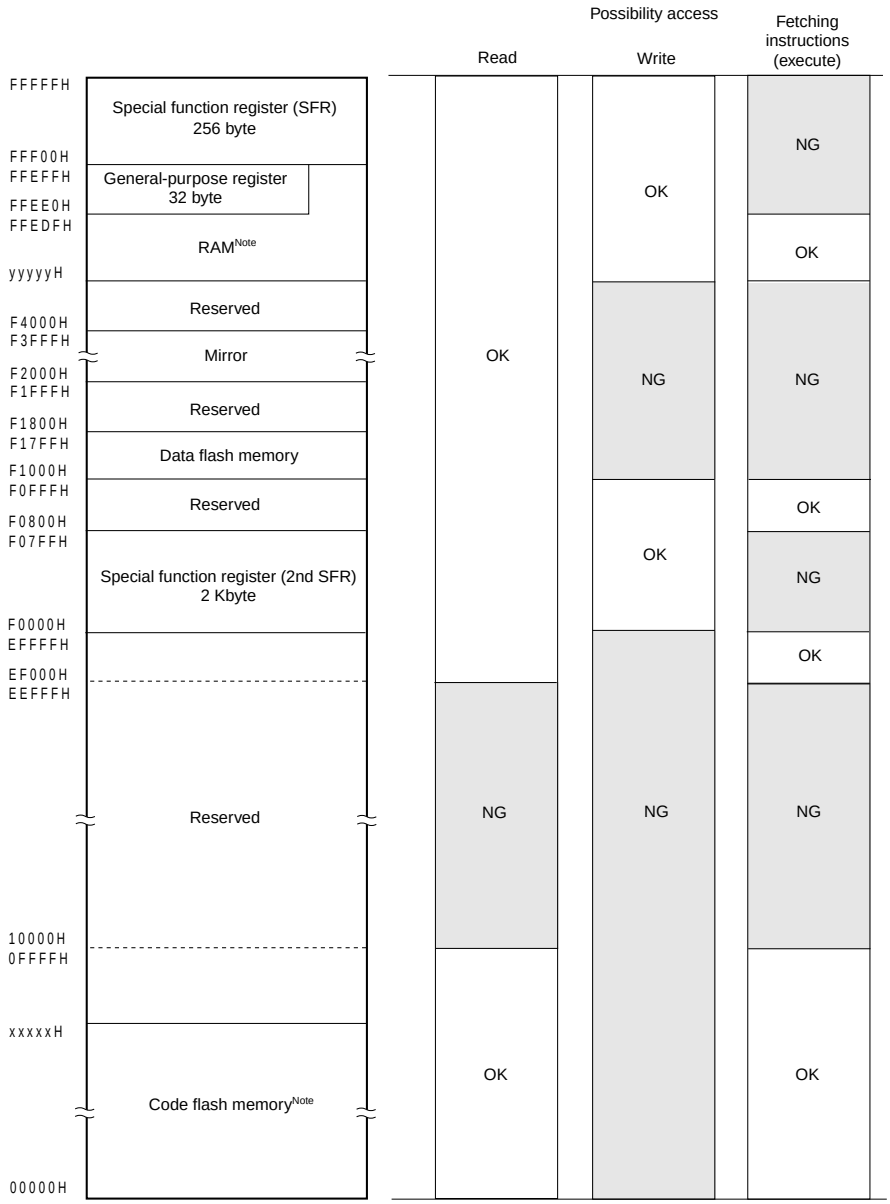
21.3.5 Invalid memory access detection function

The IEC60730 standard mandates checking that the CPU and interrupts are operating correctly.

The illegal memory access detection function triggers a reset if a memory space specified as access-prohibited is accessed.

The illegal memory access detection function applies to the areas indicated by NG in Figure 21-8.

Figure 21-8. Invalid access detection area



Note Code flash memory and RAM address of each product are as follows.

Products (x = 2, 3)	Code flash memory (00000H to xxxxxH)	RAM (yyyyyH to FFEFFH)
R5F10x66	2048 × 8 bit (00000H to 007FFH)	256 × 8 bit (FFE00H to FFEFFH)
R5F10x67, R5F10x77, R5F10xA7	4096 × 8 bit (00000H to 00FFFH)	512 × 8 bit (FFD00H to FFEFFH)
R5F10x68, R5F10x78, R5F10xA8	8192 × 8 bit (00000H to 01FFFH)	768 × 8 bit (FFC00H to FFEFFH)
R5F10x69, R5F10x79, R5F10xA9	12288 × 8 bit (00000H to 02FFFH)	1024 × 8 bit (FFB00H to FFEFFH)
R5F10x6A, R5F10x7A	16384 × 8 bit (00000H to 03FFFH)	1536 × 8 bit (FF900H to FFEFFH)
R5F10xAA		2048 × 8 bit (FF700H to FFEFFH)

Downloaded from Arrow.com.

21.3.5.1 Invalid memory access detection control register (IAWCTL)

This register is used to control the detection of invalid memory access and RAM/SFR guard function.

IAWEN bit is used in invalid memory access detection function.

The IAWCTL register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 21-9. Format of Invalid Memory Access Detection Control Register (IAWCTL)

Address: F0078H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IAWCTL	IAWEN	0	GRAM1	GRAM0	0	GPORT	GINT	GCSC

IAWEN Note	Control of invalid memory access detection
0	Disable the detection of invalid memory access.
1	Enable the detection of invalid memory access.

Note Only writing 1 to the IAWEN bit is enabled, not writing 0 to it after setting it to 1.

Remark By specifying WDTON = 1 (watchdog timer operation enable) for the option byte (000C0H), the invalid memory access detection function is enabled even IAWEN = 0.

21.3.6 Frequency detection function

The IEC60730 standard mandates checking that the oscillation frequency is correct.

By using the CPU/peripheral hardware clock frequency (f_{CLK}) and measuring the pulse width of the input signal to channel 1 (20- and 24-pin products) or channel 5 (30-pin products) of the timer array unit 0 (TAU0), whether the proportional relationship between the two clock frequencies is correct can be determined. Note that, however, if one or both clock operations are completely stopped, the proportional relationship between the clocks cannot be determined.

Caution The following description is applicable in the case of the channel configuration in the 20- and 24-pin products.

<Clocks to be compared>

<1> CPU/peripheral hardware clock frequency (f_{CLK}):

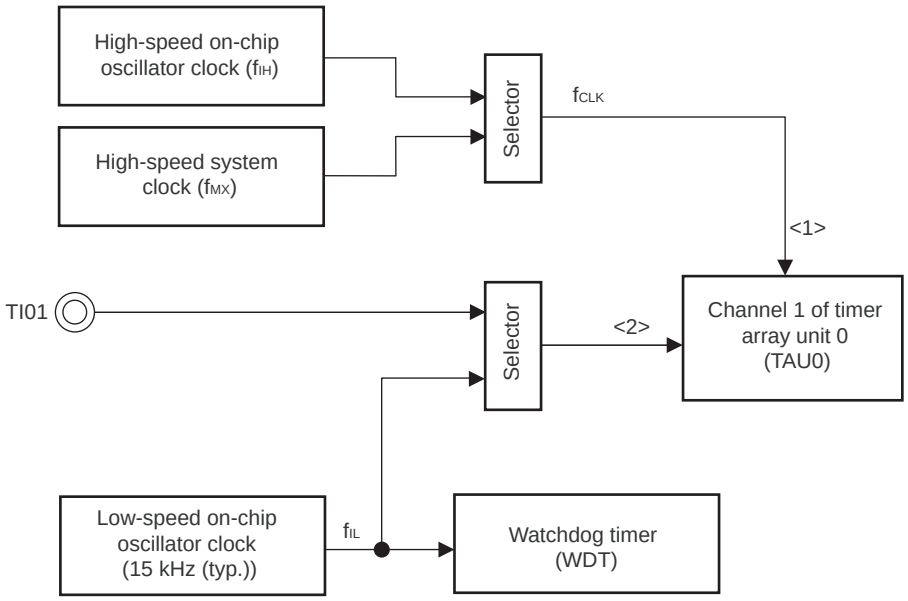
- High-speed on-chip oscillator clock (f_{IH})
- High-speed system clock (f_{MX})

<2> Input to channel 1 of the timer array unit:

- Timer input to channel 1 (TI01)
- Low-speed on-chip oscillator clock (f_{IL}: 15 kHz (typ.))

Downloaded from Arrow.com.

Figure 21-10. Configuration of Frequency Detection Function



If input pulse interval measurement results in an abnormal value, it can be concluded that the clock frequency is abnormal.

For how to execute input pulse interval measurement, see **6.8.4 Operation as input pulse interval measurement**.

21.3.6.1 Timer input select register 0 (TIS0)

This register is used to select the timer input of channel 1 of the timer array unit 0 (TAU0) in 20- and 24-pin products.

The TIS0 register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 21-11. Format of Timer Input Select Register 0 (TIS0)

Address: F0074H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TIS0	0	0	0	0	0	0	TIS01	TIS00

TIS01	TIS00	Selection of timer input used with channel 1
×	0	Input signal of timer input pin (TI01)
0	1	Low-speed on-chip oscillator clock (fIL)
1	1	Setting prohibited

Remark ×: don't care

Downloaded from Arrow.com.

<R> **21.3.7 A/D test function**

The IEC60730 standard mandates testing the A/D converter. The A/D test function checks whether or not the A/D converter is operating normally by executing A/D conversions of the A/D converter's positive and negative reference voltages, analog input channel (ANI), temperature sensor output voltage, and the internal reference voltage. For details of the check method, see the safety function (A/D test) application note (R01AN0955).

The analog multiplexer can be checked using the following procedure.

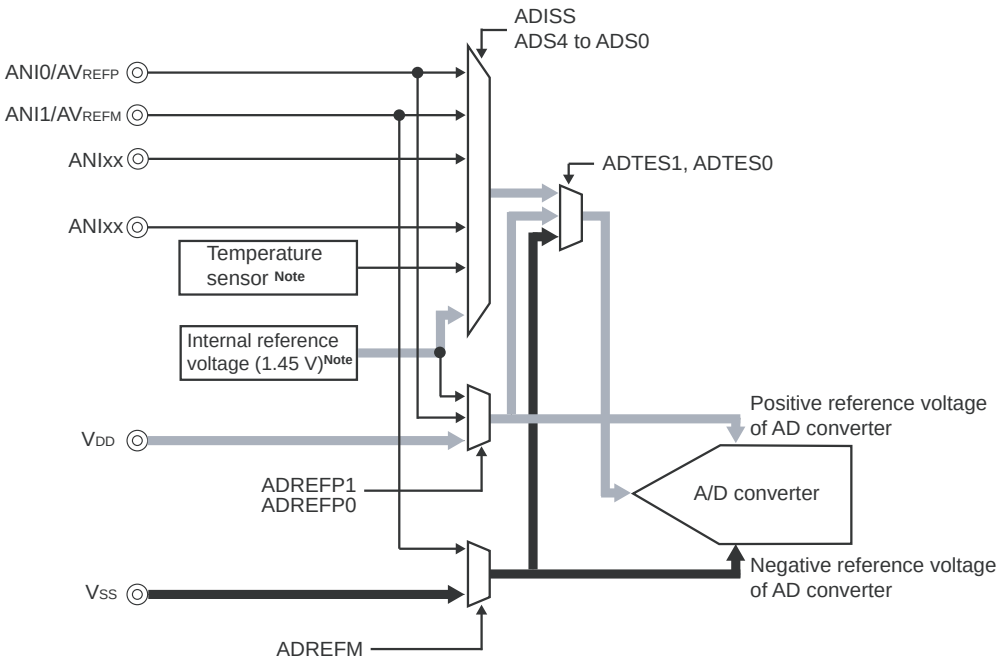
<1> Select the ANIx pin for A/D conversion using the ADTES register (ADTES1 = 0, ADTES0 = 0).
<2> Perform A/D conversion for the ANIx pin (conversion result 1-1).
<3> Select the A/D converter's negative reference voltage for A/D conversion using the ADTES register (ADTES1 = 1, ADTES0 = 0)
<4> Perform A/D conversion of the negative reference voltage of the A/D converter (conversion result 2-1).
<5> Select the ANIx pin for A/D conversion using the ADTES register (ADTES1 = 0, ADTES0 = 0).
<6> Perform A/D conversion for the ANIx pin (conversion result 1-2).
<7> Select the A/D converter's positive reference voltage for A/D conversion using the ADTES register (ADTES1 = 1, ADTES0 = 1)
<8> Perform A/D conversion of the positive reference voltage of the A/D converter (conversion result 2-2).
<9> Select the ANIx pin for A/D conversion using the ADTES register (ADTES1 = 0, ADTES0 = 0).
<10> Perform A/D conversion for the ANIx pin (conversion result 1-3).
<11> Check that the conversion results 1-1, 1-2, and 1-3 are equal.
<12> Check that the A/D conversion result 2-1 is all zero and conversion result 2-2 is all one.

Using the procedure above can confirm that the analog multiplexer is selected and all wiring is connected.

Remarks 1. If the analog input voltage is variable during A/D conversion in steps <1> to <10> above, use another method to check the analog multiplexer.

2. The conversion results might contain an error. Consider an appropriate level of error when comparing the conversion results.

Figure 21-12. Configuration of A/D Test Function



Note This setting can be used only in HS (high-speed main) mode.

Downloaded from Arrow.com.

21.3.7.1 A/D test register (ADTES)

This register is used to select the A/D converter's positive reference voltage, A/D converter's negative reference voltage, analog input channel (ANIxx), temperature sensor output voltage, or internal reference voltage (1.45 V) as the target of A/D conversion.

When using the A/D test function, specify the following settings:

- Select negative reference voltage as the target of A/D conversion for zero-scale measurement.
- Select positive reference voltage as the target of A/D conversion for full-scale measurement.

The ADTES register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 21-13. Format of A/D Test Register (ADTES)

Address: F0013H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADTES	0	0	0	0	0	0	ADTES1	ADTES0

ADTES1	ADTES0	A/D conversion target
0	0	ANIxx / temperature sensor output Note / internal reference voltage (1.45 V) Note (This is specified using the analog input channel specification register (ADS).)
1	0	Negative reference voltage (selected with the ADREFM bit in ADM2)
1	1	Positive reference voltage (selected with the ADREFP1 or ADREFP0 bit in ADM2)
Other than the above		Setting prohibited

Note Temperature sensor output voltage and internal reference voltage (1.45 V) can be used only in HS (high-speed main) mode.

Downloaded from Arrow.com.

<R> 21.3.7.2 Analog input channel specification register (ADS)

This register specifies the input channel of the analog voltage to be A/D converted.

Set A/D test register (ADTES) to 00H when measuring the ANIxx/temperature sensor output voltage/internal reference voltage (1.45 V).

The ADS register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 21-14. Format of Analog Input Channel Specification Register (ADS) (1/2)

Address: FFF31H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADS	ADISS	0	0	ADS4	ADS3	ADS2	ADS1	ADS0

○ Select mode (ADMD = 0) (20- and 24-pin products)

ADISS	ADS4	ADS3	ADS2	ADS1	ADS0	Analog input channel	Input source
0	0	0	0	0	0	ANI0	P20/ANI0/AV_{REFP} pin
0	0	0	0	0	1	ANI1	P21/ANI1/AV_{REFM} pin
0	0	0	0	1	0	ANI2	P22/ANI2 pin
0	0	0	0	1	1	ANI3	P23/ANI3 pin
0	1	0	0	0	0	ANI16	P10/ANI16 pin
0	1	0	0	0	1	ANI17	P11/ANI17 pin
0	1	0	0	1	0	ANI18	P12/ANI18 pin
0	1	0	0	1	1	ANI19	P13/ANI19 pin
0	1	0	1	0	0	ANI20	P14/ANI20 pin
0	1	0	1	0	1	ANI21	P42/ANI21 pin
0	1	0	1	1	0	ANI22	P41/ANI22 pin
1	0	0	0	0	0	–	Temperature sensor output Note
1	0	0	0	0	1	–	Internal reference voltage output (1.45 V) Note
Other than the above						Setting prohibited	

Note This setting can be used only in HS (high-speed main) mode.

○ Select mode (ADMD = 0) (30-pin products)

ADISS	ADS4	ADS3	ADS2	ADS1	ADS0	Analog input channel	Input source
0	0	0	0	0	0	ANI0	P20/ANI0/AVREFP pin
0	0	0	0	0	1	ANI1	P21/ANI1/AVREFM pin
0	0	0	0	1	0	ANI2	P22/ANI2 pin
0	0	0	0	1	1	ANI3	P23/ANI3 pin
0	1	0	0	0	0	ANI16	P01/ANI16 pin
0	1	0	0	0	1	ANI17	P00/ANI17 pin
0	1	0	0	1	0	ANI18	P147/ANI18 pin
0	1	0	0	1	1	ANI19	P120/ANI19 pin
1	0	0	0	0	0	–	Temperature sensor output Note
1	0	0	0	0	1	–	Internal reference voltage output (1.45 V) Note
Other than the above						Setting prohibited	

Note This setting can be used only in HS (high-speed main) mode.

Downloaded from Arrow.com.

Cautions 1. Be sure to clear bits 5 and 6 to 0.

<R> **2. Select input mode for the ports which are set to analog input with the ADPC and PMC registers, using the port mode registers 0, 1, 2, 4, 12, and 14 (PM0, PM1, PM2, PM4, PM12, and PM14).**

<R> **3. Do not use the ADS register to set the pins which should be set as digital I/O with the A/D port configuration register (ADPC).**

<R> **4. Do not use the ADS register to set the pins which should be set as digital I/O with the port mode control registers 0, 1, 4, 12, and 14 (PMC0, PMC1, PMC4, PMC12, and PMC14).**

5. Only rewrite the value of the ADISS bit while conversion operation is stopped (ADCS = 0, ADCE = 0).

6. If using AV_{REFP} as the positive reference voltage source of the A/D converter, do not select ANI0 as an A/D conversion channel.

7. If using AV_{REFM} as the negative reference voltage source of the A/D converter, do not select ANI1 as an A/D conversion channel.

8. When ADISS is 1, the internal reference voltage (1.45 V) cannot be used for the positive reference voltage. In addition, the first conversion result obtained after setting ADISS to 1 is not available. For detailed setting flow, see 10.7.4 Setup when temperature sensor output voltage/internal reference voltage is selected.

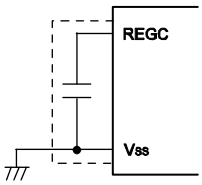
<R> **9. If a transition is made to STOP mode, do not set ADISS to 1. When ADISS is 1, the A/D converter reference voltage current (I_{ADREF}) shown in 28.3.2 Supply current characteristics is added.**

Downloaded from Arrow.com.

CHAPTER 22 REGULATOR

22.1 Overview of Regulators

The 30-pin product of the RL78/G12 incorporates the circuit for constant voltage operation in the device. To stabilize the regulator output, connect the REGC pin to Vss via a capacitor (0.47 to 1 μF) for regulator stabilization. Use a capacitor with good characteristics because it is used for stabilization of internal voltage.



Caution Keep the wiring length as short as possible for the broken-line part in the above figure.

The regulator output voltage, see table 22-1.

Table 22-1. Regulator Output Voltage Conditions

Mode	Output voltage	Condition
LS (low-speed main) mode	1.8 V	–
HS (high-speed main) mode	1.8 V	In the STOP mode
	2.1 V	Other than STOP mode (include during OCD mode) Note

Note When shifting to the STOP mode during on-chip debugging, the regulator output voltage is kept at 2.1 V (not decline to 1.8 V).

Downloaded from Arrow.com.

CHAPTER 23 OPTION BYTE

23.1 Functions of Option Bytes

Addresses 000C0H to 000C3H of the flash memory of the RL78/G12 form an option byte area.

Option bytes consist of user option byte (000C0H to 000C2H) and on-chip debug option byte (000C3H).

Upon power application or resetting and starting, an option byte is automatically referenced and a specified function is set. When using the product, be sure to set the following functions by using the option bytes.

For the bits to which no function is allocated, do not change their initial values.

<R> **Caution The option bytes should always be set regardless of whether each function is used.**

23.1.1 User option byte (000C0H to 000C2H)

(1) 000C0H

- ○ Operation of watchdog timer
  - Enabling or disabling of counter operation
  - Enabling or disabling of counter operation in the HALT or STOP mode
- ○ Setting of overflow time of watchdog timer
- ○ Setting of window open period of watchdog timer
- ○ Setting of interval interrupt of watchdog timer
  - Whether or not to use the interval interrupt is selectable.

(2) 000C1H

- ○ Setting of LVD operation mode
  - Interrupt & reset mode
  - Reset mode
  - Interrupt mode
  - LVD off (by controlling the externally input reset signal on the $\overline{\text{RESET}}$ pin)
- ○ Setting of LVD detection level (V_{LVDH}, V_{LVDL}, V_{LVD})
- ○ Controlling of P125/$\overline{\text{RESET}}$ pin
  - P125/KR1/SI01 or $\overline{\text{RESET}}$

<R> **Caution After power is supplied, the reset state must be retained until the operating voltage becomes in the range defined in 28.4 or 29.4 AC Characteristics. This is done by utilizing the voltage detection circuit or controlling the externally input reset signal. After the power supply is turned off, this LSI should be placed in the STOP mode, or placed in the reset state by utilizing the voltage detection circuit or controlling the externally input reset signal, before the voltage falls below the operating range. The range of operating voltage varies with the setting of the user option byte (000C2H).**

(3) 000C2H

- ○ Setting of flash operation mode
  - LS (low speed main) mode
  - HS (high speed main) mode
- ○ Setting of the frequency of the high-speed on-chip oscillator

<R>
  - Select from 24 MHz/16 MHz/12 MHz/8 MHz/6 MHz/4 MHz /3 MHz/2 MHz/1 MHz (TYP.).

Downloaded from Arrow.com.

23.1.2 On-chip debug option byte (000C3H)

- Control of on-chip debug operation
  - On-chip debug operation is disabled or enabled.
- Handling of data of flash memory in case of failure in on-chip debug security ID authentication
  - Data of flash memory is erased or not erased in case of failure in on-chip debug security ID authentication.

Downloaded from Arrow.com.

23.2 Format of User Option Byte

The format of user option byte is shown below.

Figure 23-1. Format of User Option Byte (000C0H)

Address: 000C0H

7	6	5	4	3	2	1	0
WDTINIT	WINDOW1	WINDOW0	WDTON	WDCS2	WDCS1	WDCS0	WDSTBYON

WDTINIT	Use of interval interrupt of watchdog timer
0	Interval interrupt is not used.
1	Interval interrupt is generated when 75% + 1/2 f_{IL} of the overflow time is reached.

WINDOW1	WINDOW0	Watchdog timer window open period[Note]
0	0	Setting prohibited
0	1	50%
1	0	75%
1	1	100%

WDTON	Operation control of watchdog timer counter
0	Counter operation disabled (counting stopped after reset)
1	Counter operation enabled (counting started after reset)

WDCS2	WDCS1	WDCS0	Watchdog timer overflow time (f_{IL} = 17.25 kHz (MAX.))
0	0	0	$2^6/f_{IL}$ (3.71 ms)
0	0	1	$2^7/f_{IL}$ (7.42 ms)
0	1	0	$2^8/f_{IL}$ (14.84 ms)
0	1	1	$2^9/f_{IL}$ (29.68 ms)
1	0	0	$2^{11}/f_{IL}$ (118.72 ms)
1	0	1	$2^{13}/f_{IL}$ (474.89 ms)
1	1	0	$2^{14}/f_{IL}$ (949.79 ms)
1	1	1	$2^{16}/f_{IL}$ (3799.18 ms)

WDSTBYON	Operation control of watchdog timer counter (HALT/STOP mode)
0	Counter operation stopped in HALT/STOP mode[Note]
1	Counter operation enabled in HALT/STOP mode

Note The window open period is 100% when WDSTBYON = 0, regardless the value of the WINDOW1 and WINDOW0 bits.

Remark f_{IL}: Low-speed on-chip oscillator clock frequency

Downloaded from Arrow.com.

Figure 23-2. Format of User Option Byte (000C1H) (1/2)

Address: 000C1H

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	PORTSELB Note	LVIS1	LVIS0	LVIMDS1	LVIMDS0

<R> • LVD setting (interrupt mode & reset mode)

Detection voltage			Option byte setting value						
V_{LVDH}		V_{LVDL}	VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting	
Rising edge	Falling edge	Falling edge						LVIMDS1	LVIMDS0
1.98 V	1.94 V	1.84 V	0	0	1	1	0	1	0
2.09 V	2.04 V					0	1		
3.13 V	3.06 V					0	0		
2.61 V	2.55 V	2.45 V		1	0	1	0		
2.71 V	2.65 V					0	1		
3.75 V	3.67 V					0	0		
2.92 V	2.86 V	2.75 V		1	1	1	0		
3.02 V	2.96 V					0	1		
4.06 V	3.98 V					0	0		
—			Setting of values other than above is prohibited.						

<R> • LVD setting (reset mode)

Detection voltage		Option byte setting value						
V_{LVD}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting	
Rising edge	Falling edge						LVIMDS1	LVIMDS0
1.88 V	1.84 V	0	0	1	1	1	1	1
1.98 V	1.94 V		0	1	1	0		
2.09 V	2.04 V		0	1	0	1		
2.50 V	2.45 V		1	0	1	1		
2.61 V	2.55 V		1	0	1	0		
2.71 V	2.65 V		1	0	0	1		
2.81 V	2.75 V		1	1	1	1		
2.92 V	2.86 V		1	1	1	0		
3.02 V	2.96 V		1	1	0	1		
3.13 V	3.06 V		0	1	0	0		
3.75 V	3.67 V		1	0	0	0		
4.06 V	3.98 V		1	1	0	0		
—		Setting of values other than above is prohibited.						

Note 20- and 24-pin products only

Downloaded from Arrow.com.

Figure 23-2. Format of User Option Byte (000C1H) (2/2)

Address: 000C1H

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	PORTSELB Note	LVIS1	LVIS0	LVIMDS1	LVIMDS0

<R> • LVD setting (interrupt mode)

Detection voltage		Option byte setting value						
V_{LVD}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting	
Rising edge	Falling edge						LVIMDS1	LVIMDS0
1.88 V	1.84 V	0	0	1	1	1	0	1
1.98 V	1.94 V		0	1	1	0		
2.09 V	2.04 V		0	1	0	1		
2.50 V	2.45 V		1	0	1	1		
2.61 V	2.55 V		1	0	1	0		
2.71 V	2.65 V		1	0	0	1		
2.81 V	2.75 V		1	1	1	1		
2.92 V	2.86 V		1	1	1	0		
3.02 V	2.96 V		1	1	0	1		
3.13 V	3.06 V		0	1	0	0		
3.75 V	3.67 V		1	0	0	0		
4.06 V	3.98 V		1	1	0	0		
—		Setting of values other than above is prohibited.						

• LVD off (by controlling the externally input reset signal on the $\overline{\text{RESET}}$ pin)

<R>

Detection voltage		Option byte setting value						
V_{LVD}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting	
Rising edge	Falling edge						LVIMDS1	LVIMDS0
–	–	1	×	×	×	×	×	1
—		Setting of values other than above is prohibited.						

• Setting of the P125/KR1/SI01/$\overline{\text{RESET}}$ Pin (20-, 24-pin products)

PORTSELB	P125/$\overline{\text{RESET}}$ pin control
0	Port function (P125/KR1/SI01)
1	$\overline{\text{RESET}}$ input (The internal pull-up resistor is always enabled.)

Note 20- and 24-pin products only

Cautions 1. In the 30-pin products, be sure to set bit 4 (PORTSELB) to 1.

2. After power is supplied, the reset state must be retained until the operating voltage becomes in the range defined in 28.4 or 29.4 AC Characteristics. This is done by utilizing the voltage detection circuit or controlling the externally input reset signal. After the power supply is turned off, this LSI should be placed in the STOP mode, or placed in the reset state by utilizing the voltage detection circuit or controlling the externally input reset signal, before the voltage falls below the operating range. The range of operating voltage varies with the setting of the user option byte (000C2H).

Remarks 1. ×: don't care

<R> **2.** For details on the LVD circuit, see **CHAPTER 20 VOLTAGE DETECTOR**.

<R> **3.** The detection voltage is a typical value. For details, see **28.6.4** or **29.6.4 LVD circuit characteristics**.

Downloaded from Arrow.com.

Figure 23-3. Format of Option Byte (000C2H)

Address: 000C2H

<R>

7	6	5	4	3	2	1	0
CMODE1	CMODE0	1	0	FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0

CMODE1	CMODE0	Setting of flash operation mode		
			Operating Frequency Range	Operating Voltage Range
1	0	LS (low speed main) mode	1 to 8 MHz	1.8 to 5.5 V
1	1	HS (high speed main) mode	1 to 16 MHz	2.4 to 5.5 V
			1 to 24 MHz	2.7 to 5.5 V
Other than above		Setting prohibited		

FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0	Frequency of the high-speed on-chip oscillator
0	0	0	0	24 MHz
1	0	0	1	16 MHz
0	0	0	1	12 MHz
1	0	1	0	8 MHz
0	0	1	0	6 MHz
1	0	1	1	4 MHz
0	0	1	1	3 MHz
1	1	0	0	2 MHz
1	1	0	1	1 MHz
Other than above				Setting prohibited

<R>

Cautions 1. Be sure to set bit 5 to “1" and bit 4 to “0”

2. The ranges of operation frequency and operation voltage vary depending on the flash operation mode. For details, see 28.4 or 29.4 AC Characteristics.

Downloaded from Arrow.com.

23.3 Format of On-chip Debug Option Byte

The format of on-chip debug option byte is shown below.

Figure 23-4. Format of On-chip Debug Option Byte (000C3H)

Address: 000C3H

7	6	5	4	3	2	1	0
OCDENSET	0	0	0	0	1	0	OCDERSD

OCDENSET	OCDERSD	Control of on-chip debug operation
0	0	Disables on-chip debug operation.
0	1	Setting prohibited
1	0	Enables on-chip debugging. Erases data of flash memory in case of failures in authenticating on-chip debug security ID.
1	1	Enables on-chip debugging. Does not erases data of flash memory in case of failures in authenticating on-chip debug security ID.

Caution Bits 7 and 0 (OCDENSET and OCDERSD) can only be specified a value.
Be sure to set 000010B to bits 6 to 1.

Remark The value on bits 3 to 1 will be written over when the on-chip debug function is in use and thus it will become unstable after the setting.
However, be sure to set the default values (0, 1, and 0) to bits 3 to 1 at setting.

Downloaded from Arrow.com.

23.4 Setting of Option Byte

<R> The user option byte and on-chip debug option byte can be set using the link option, in addition to describing to the source. When doing so, the contents set by using the link option take precedence, even if descriptions exist in the source, as mentioned below.

A software description example of the option byte setting is shown below.

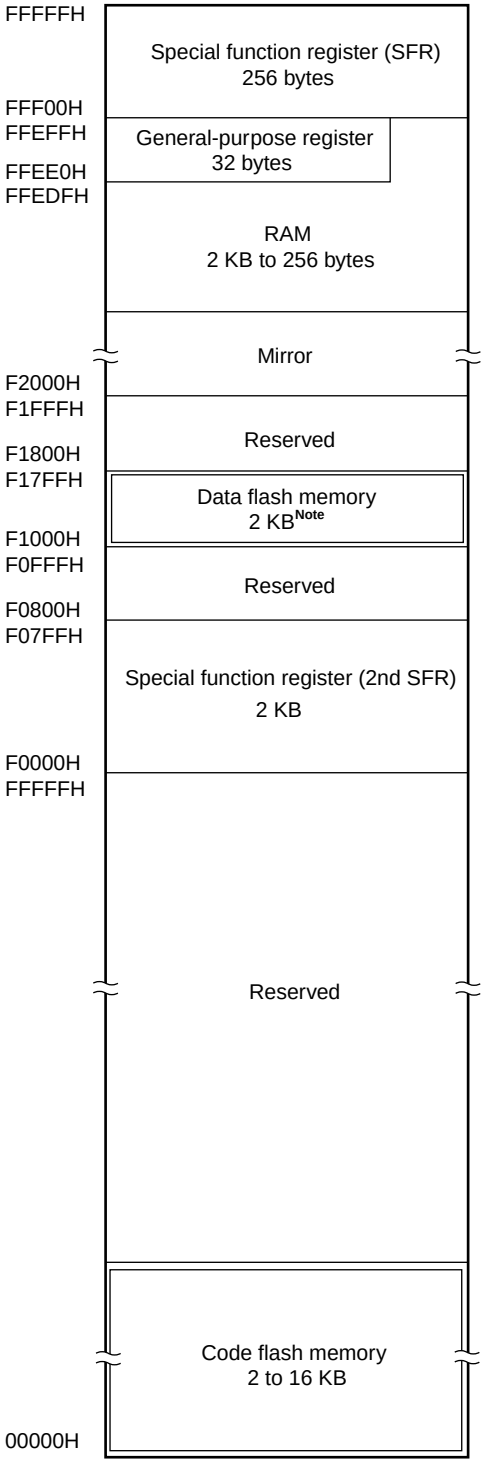
```
OPT   CSEG      OPT_BYTE
      DB        36H     ; Does not use interval interrupt of watchdog timer,
                        ; Enables watchdog timer operation,
                        ; Window open period of watchdog timer is 50%,
                        ; Overflow time of watchdog timer is 2^9/fIL,
                        ; Stops watchdog timer operation during HALT/STOP mode
      DB        2AH     ; Select 1.84 V for VLVDL
                        ; Select 1.94 V for VLVDH
                        ; Select the interrupt & reset mode as the LVD operation mode
                        ; Do not use reset input
<R>   DB        ADH     ; Select the LS (low-speed main) mode as the flash operation mode
                          and 1 MHz as the frequency of the high-speed on-chip oscillator clock
      DB        85H     ; Enables on-chip debug operation, does not erase flash memory
                          data when security ID authorization fails
```

Caution To specify the option byte by using assembly language, use OPT_BYTE as the relocation attribute name of the CSEG pseudo instruction.

Downloaded from Arrow.com.

CHAPTER 24 FLASH MEMORY

The RL78 microcontroller incorporates the flash memory to which a program can be written, erased, and overwritten. The flash memory includes the "code flash memory", in which programs can be executed, and the "data flash memory", an area for storing data.



Note This area is reserved in the R5F103 products.

Downloaded from Arrow.com.

The following methods for programming the flash memory are available.

The code flash memory can be rewritten to through serial programming using a flash memory programmer or an external device (UART communication), or through self-programming.

- Serial programming using flash memory programmer (see **24.4**)
  Data can be written to the flash memory on-board or off-board by using a dedicated flash memory programmer.

- Serial programming using external device (UART communication) (see **24.2**)
  Data can be written to the flash memory on-board through UART communication with an external device (microcontroller or ASIC).

- Self-programming (see **24.6**)
  The user application can execute self-programming of the code flash memory by using the flash self-programming library. **Note**

Note The self-programming function is not available in R5F10266 and R5F10366.

The data flash memory can be rewritten to by using the data flash library during user program execution (background operation). For access and writing to the data flash memory, see **24.8 Data Flash**.

Downloaded from Arrow.com.

24.1 Serial Programming Using Flash Memory Programmer

The following dedicated flash memory programmer can be used to write data to the internal flash memory of the RL78 microcontroller.

- PG-FP5, FL-PR5
- E1 on-chip debugging emulator

Data can be written to the flash memory on-board or off-board, by using a dedicated flash memory programmer.

(1) On-board programming

The contents of the flash memory can be rewritten after the RL78 microcontroller has been mounted on the target system. The connectors that connect the dedicated flash memory programmer must be mounted on the target system.

(2) Off-board programming

Data can be written to the flash memory with a dedicated program adapter (FA series) before the RL78 microcontroller is mounted on the target system.

Remark FL-PR5 and FA series are products of Naito Densei Machida Mfg. Co., Ltd.

Table 24-1. Wiring Between RL78/G12 and Dedicated Flash Memory Programmer

Pin Configuration of Dedicated Flash Memory Programmer				Pin Name	Pin No.		
					20-pin	24-pin	30-pin
Signal Name		I/O	Pin Function		SSOP	WQFN (4 × 4)	SSOP
PG-FP5, FL-PR5	E1 on-chip debugging emulator						
–	TOOL0	I/O	Transmit/receive signal	TOOL0/ P40	4	24	5
SI/RxD	–	I/O	Transmit/receive signal				
–	$\overline{\text{RESET}}$	Output	Reset signal	$\overline{\text{RESET}}$	5	1	6
/RESET	–	Output					
V_{DD}		I/O	V_{DD} voltage generation/ power monitoring	V_{DD}	10	6	12
GND		–	Ground	V_{SS}	9	5	11
				REGC[Note]	–	–	10
EMV_{DD}		–	Driving power for TOOL pin	V_{DD}	10	6	12

Note Connect REGC pin to ground via a capacitor (0.47 to 1 μF).

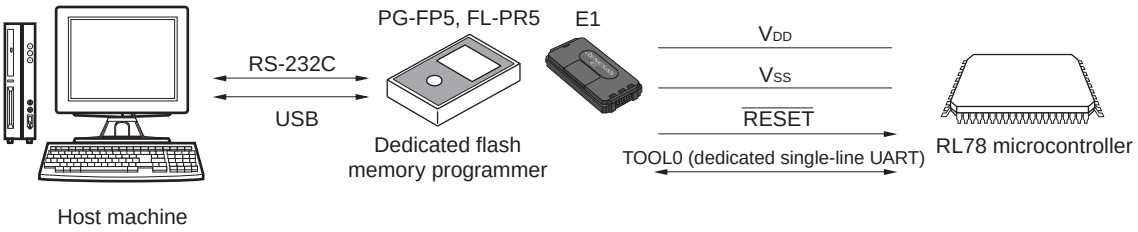
Remark Pins that are not indicated in the above table can be left open when using the flash memory programmer for flash programming.

Downloaded from Arrow.com.

24.1.1 Programming environment

The environment required for writing a program to the flash memory of the RL78 microcontroller is illustrated below.

Figure 24-1. Environment for Writing Program to Flash Memory



A host machine that controls the dedicated flash memory programmer is necessary.

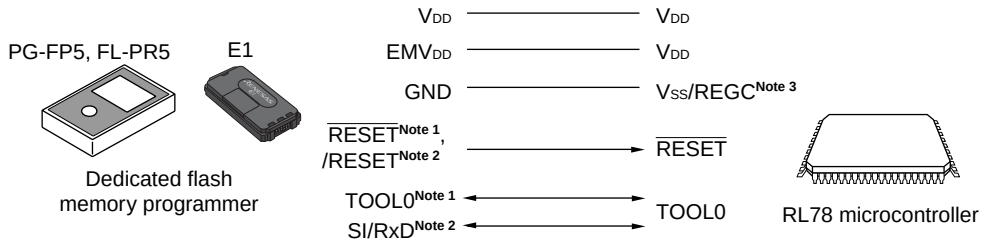
To interface between the dedicated flash memory programmer and the RL78 microcontroller, the TOOL0 pin is used for manipulation such as writing and erasing via a dedicated single-line UART.

24.1.2 Communication mode

Communication between the dedicated flash memory programmer and the RL78 microcontroller is established by serial communication using the TOOL0 pin via a dedicated single-line UART of the RL78 microcontroller.

Transfer rate: 1 M, 500k, 250 k, 115.2 kbps

Figure 24-2. Communication with Dedicated Flash Memory Programmer



Notes 1. When using E1 on-chip debugging emulator.

2. When using PG-FP5 or FL-PR5.

3. Connect the REGC pin to ground via a capacitor (0.47 to 1 μF) (30-pin products only).

Downloaded from Arrow.com.

The dedicated flash memory programmer generates the following signals for the RL78 microcontroller. See the manual of PG-FP5, FL-PR5, or E1 on-chip debugging emulator for details.

Table 24-2. Pin Connection

Dedicated Flash Memory Programmer				RL78 Microcontroller
Signal Name		I/O	Pin Function	Pin Name
PG-FP5, FL-PR5	E1 on-chip debugging emulator			
V_{DD}		I/O	V_{DD} voltage generation/power monitoring	V_{DD}
GND		–	Ground	V_{SS}, REGC Note
EMV_{DD}		–	Driving power for TOOL0 pin	V_{DD}
/RESET	–	Output	Reset signal	$\overline{RESET}$
–	$\overline{RESET}$	Output		
–	TOOL0	I/O	Transmit/receive signal	TOOL0
SI/RxD	–	I/O	Transmit/receive signal	

Note Connect REGC pin to ground via a capacitor (0.47 to 1 μF) (30-pin products only).

Caution Pins to be connected differ with the product. For details, see Table 24-1.

24.2 Serial Programming Using External Device (that Incorporates UART)

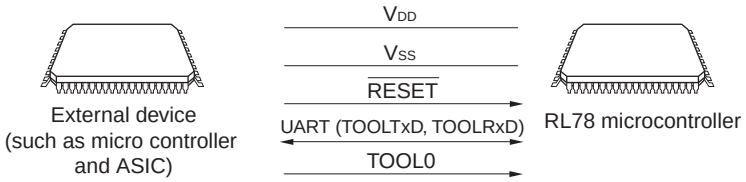
On-board data writing to the internal flash memory is possible by using the RL78 microcontroller and an external device (a microcontroller or ASIC) connected to a UART.

On the development of flash memory programmer by user, refer to the **RL78 Microcontrollers (RL78 Protocol A) Programmer Edition Application Note (R01AN0815)**.

24.2.1 Programming environment

The environment required for writing a program to the flash memory of the RL78 microcontroller is illustrated below.

Figure 24-3. Environment for Writing Program to Flash Memory



Processing to write data to or delete data from the RL78 microcontroller by using an external device is performed on-board. Off-board writing is not possible.

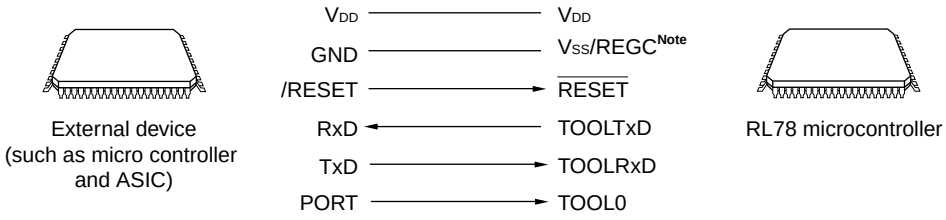
Downloaded from Arrow.com.

24.2.2 Communication mode

Communication between the external device and the RL78 microcontroller is established by serial communication using the TOOLTxD and TOOLRxD pins via the dedicated UART of the RL78 microcontroller.

Transfer rate: 1 M, 500 k, 250 k, 115.2 kbps

Figure 24-4. Communication with External Device



Note Connect the REGC pin to ground via a capacitor (0.47 to 1 μF) (30-pin products only).

The external device generates the following signals for the RL78 microcontroller.

Table 24-3. Pin Connection

External Device			RL78 Microcontroller
Signal Name	I/O	Pin Function	Pin Name
V_{DD}	I/O	V_{DD} voltage generation/power monitoring	V_{DD}
GND	–	Ground	V_{SS}, REGC Note
RESETOUT	Output	Reset signal output	$\overline{RESET}$
RxD	Input	Receive signal	TOOLTxD
TxD	Output	Transmit signal	TOOLRxD
PORT	Output	Mode signal	TOOL0

Note Connect REGC pin to ground via a capacitor (0.47 to 1 μF) (30-pin products only).

Downloaded from Arrow.com.

24.3 Connection of Pins on Board

To write the flash memory on-board by using the flash memory programmer, connectors that connect the dedicated flash memory programmer must be provided on the target system. First provide a function that selects the normal operation mode or flash memory programming mode on the board.

When the flash memory programming mode is set, all the pins not used for programming the flash memory are in the same status as immediately after reset. Therefore, if the external device does not recognize the state immediately after reset, the pins must be handled as described below.

Remark For the flash memory programming mode, see **24.4.2 Flash memory programming mode**.

24.3.1 P40/TOOL0 pin

In the flash memory programming mode, pull up externally with a 1 kΩ resister, and connect it to the dedicated flash memory programmer.

When using it as a port pin, use it as described below.

<R> When used as an input pin: Do not input a low level for t_{HD} period after the external reset release. However, when this pin is used via pull-down resistors, use the 500 kΩ or more resistors.

When used as an output pin: When this pin is used via pull-down resistors, use the 500 kΩ or more resistors.

<R> **Remarks 1.** t_{HD}: How long to keep the TOOL0 pin at the low level from when the external reset ends for setting of the flash memory programming mode (see **28.10** or **29.10 Timing of Entry to Flash Memory Programming Modes**)

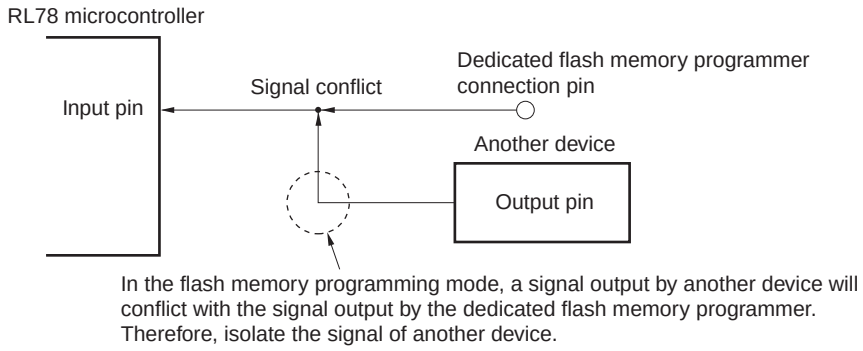
2. The SAU and IICA pins are not used for communication between the RL78 microcontroller and dedicated flash memory programmer, because single-line UART (TOOL0 pin) is used.

24.3.2 $\overline{\text{RESET}}$ pin

Signal conflict will occur if the reset signal of the dedicated flash memory programmer and external device are connected to the $\overline{\text{RESET}}$ pin that is connected to the reset signal generator on the board. To prevent this conflict, isolate the connection with the reset signal generator.

The flash memory will not be correctly programmed if the reset signal is input from the user system while the flash memory programming mode is set. Do not input any signal other than the reset signal of the dedicated flash memory programmer and external device.

Figure 24-5. Signal Conflict ($\overline{\text{RESET}}$ Pin)



Downloaded from Arrow.com.

24.3.3 Port pins

When the flash memory programming mode is set, all the pins not used for flash memory programming enter the same status as that immediately after reset. If external devices connected to the ports do not recognize the port status immediately after reset, the port pin must be connected to either to V_{DD} or V_{SS} via a resistor.

24.3.4 REGC pins

Connect the REGC pin to GND via a capacitor having excellent characteristics (0.47 to 1 μF) in the same manner as during normal operation (30-pin products only). Also, use a capacitor with good characteristics, since it is used to stabilize internal voltage.

24.3.5 X1 and X2 pins

Connect X1 and X2 in the same status as in the normal operation mode.

Remark In the flash memory programming mode, the high-speed on-chip oscillation clock (f_{IH}) is used.

24.3.6 Power supply

To use the supply voltage output of the flash memory programmer, connect the V_{DD} pin to V_{DD} of the flash memory programmer, and the V_{SS} pin to GND of the flash memory programmer.

To use the on-board supply voltage, connect in compliance with the normal operation mode.

However, when writing to the flash memory by using the flash memory programmer and using the on-board supply voltage, be sure to connect the V_{DD} and V_{SS} pins to V_{DD} and GND of the flash memory programmer to use the power monitor function with the flash memory programmer.

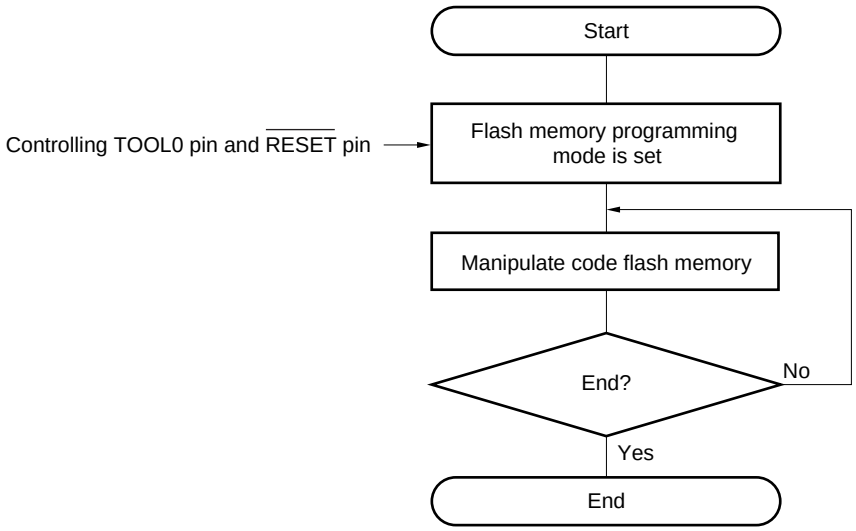
Downloaded from Arrow.com.

24.4 Serial Programming Method

24.4.1 Serial programming procedure

The following figure illustrates a flow for rewriting the code flash memory through serial programming.

Figure 24-6. Code Flash Memory Manipulation Procedure



24.4.2 Flash memory programming mode

To rewrite the contents of the code flash memory through serial programming, specify the flash memory programming mode. To enter the mode, set as follows.

<Programming using the dedicated flash memory programmer>

Connect the RL78 microcontroller to a dedicated flash memory programmer. Communication from the dedicated flash memory programmer is performed to automatically switch to the flash memory programming mode.

<Serial programming using an external device (UART communication)>

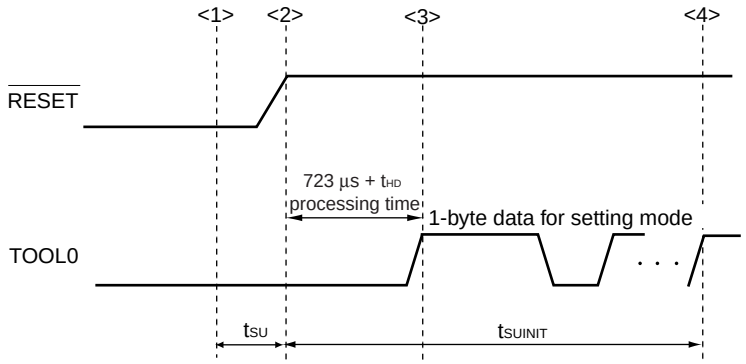
Set the TOOL0 pin to the low level, and then cancel the reset (see **Table 24-4**). After that, enter flash memory programming mode according to the procedures <1> to <4> shown in **Figure 24-7**. For details, refer to the **RL78 Microcontrollers (RL78 Protocol A) Programmer Edition Application Note (R01AN0815)**.

Table 24-4. Relationship Between TOOL0 Pin and Operation Mode After Reset Release

TOOL0	Operation Mode
V_{DD}	Normal operation mode
0 V	Flash memory programming mode

Downloaded from Arrow.com.

Figure 24-7. Setting of Flash Memory Programming Mode



<1> The low level is input to the TOOL0 pin.
<2> The external reset is released (POR and LVD reset must be released before the external reset is released.).
<3> The TOOL0 pin is set to the high level.
<4> Setting of the flash memory programming mode by UART reception and complete the baud rate setting.

Remark t_{SUINIT}: Communication for the initial setting must be completed within 100 ms after the external reset is released during this period.
t_{SU}: Time to release the external reset after the TOOL0 pin is set to the low level
t_{HD}: Time to hold the TOOL0 pin at the low level after the external reset is released (excluding the processing time of the firmware to control the flash memory)

For details, see **28.10** or **29.10 Timing of Entry to Flash Memory Programming Modes**.

There are two flash memory programming modes: wide voltage mode and full speed mode. The supply voltage value applied to the microcontroller during write operations and the setting information of the user option byte for setting of the flash memory programming mode determine which mode is selected.

When a dedicated flash memory programmer is used for serial programming, setting the voltage on GUI selects the mode automatically.

Table 24-5. Programming Modes and Voltages at Which Data Can Be Written, Erased, or Verified

<R>

Power Supply Voltage (V_{DD})	User Option Byte Setting for Switching to Flash Memory Programming Mode		Flash Programming Mode
	Flash Operation Mode	Operating Frequency	
2.7 V ≤ V_{DD} ≤ 5.5 V	Blank state		Full speed mode
	HS (high speed main) mode	1 MHz to 24 MHz	Full speed mode
	LS (low speed main) mode	1 MHz to 8 MHz	Wide voltage mode
2.4 V ≤ V_{DD} < 2.7 V	Blank state		Full speed mode
	HS (high speed main) mode	1 MHz to 16 MHz	Full speed mode
	LS (low speed main) mode	1 MHz to 8 MHz	Wide voltage mode
1.8 V ≤ V_{DD} < 2.4 V	Blank state		Wide voltage mode
	LS (low speed main) mode	1 MHz to 8 MHz	Wide voltage mode

Remarks 1. Using both the wide voltage mode and full speed mode imposes no restrictions on writing, erasing, or verification.
2. For details about communication commands, see **24.4.4 Communication commands**.

Downloaded from Arrow.com.

24.4.3 Selecting communication mode

Communication modes of the RL78 microcontroller are as follows.

Table 24-6. Communication Modes

Communication Mode	Standard Setting[Note 1]				Pins Used
	Port	Speed [Note 2]	Frequency	Multiply Rate	
1-line UART (when flash memory programmer is used, or when external device is used)	UART	115200 bps, 250000 bps, 500000 bps, 1 Mbps	–	–	TOOL0
Dedicated UART (when external device is used)	UART	115200 bps, 250000 bps, 500000 bps, 1 Mbps	–	–	TOOLTxD, TOOLRxD

Notes 1. Selection items for standard settings on GUI of the flash memory programmer.

2. Because factors other than the baud rate error, such as the signal waveform slew, also affect UART communication, thoroughly evaluate the slew as well as the baud rate error.

24.4.4 Communication commands

The RL78 microcontroller executes serial programming through the commands listed in **Table 24-7.**

The signals sent from the dedicated flash memory programmer or external device to the RL78 microcontroller are called commands, and programming functions corresponding to the commands are executed. For details, refer to the **RL78 Microcontrollers (RL78 Protocol A) Programmer Edition Application Note (R01AN0815)**.

Table 24-7. Flash Memory Control Commands

Classification	Command Name	Function
Verify	Verify	Compares the contents of a specified area of the flash memory with data transmitted from the programmer.
Erase	Block Erase	Erases a specified area in the flash memory.
Blank check	Block Blank Check	Checks if a specified block in the flash memory has been correctly erased.
Write	Programming	Writes data to a specified area in the flash memory. [Note]
Getting information	Silicon Signature	Gets the RL78 microcontroller information (such as the part number and flash memory configuration, firmware version).
	Checksum	Gets the checksum data for a specified area.
Security	Security Set	Sets security information.
	Security Get	Gets security information.
	Security Release	Releases the write prohibition setting.
Others	Reset	Used to detect synchronization status of communication.
	Baud Rate Set	Sets baud rate when UART communication mode is selected.

Note Confirm that no data has been written to the write area. Because data cannot be erased after block erase is prohibited, do not write data if the data has not been erased.

Downloaded from Arrow.com.

Product information (such as product name and firmware version) can be obtained by executing the "Silicon Signature" command.

Table 24-8 is a list of signature data and **Table 24-9** shows an example of signature data.

Table 24-8. Signature Data List

Field name	Description	Number of transmit data
Device code	The serial number assigned to the device	3 bytes
Device name	Device name (ASCII code)	10 bytes
Code flash memory area last address	Last address of code flash memory area (Sent from lower address. Example. 00000H to 03FFFH (16 KB) → FFH, 3FH, 00H)	3 bytes
Data flash memory area last address	Last address of data flash memory area (Sent from lower address. Example. F1000H to F17FFH (2 KB) → FFH, 17H, 0FH)	3 bytes
Firmware version	Version information of firmware for programming (Sent from upper address. Example. From Ver. 1.23 → 01H, 02H, 03H)	3 bytes

Table 24-9. Example of Signature Data

Field name	Description	Number of transmit data	Data (hexadecimal)
Device code	RL78 protocol A	3 bytes	10 00 06
Device name	R5F102AA	10 bytes	52 = “R” 35 = “5” 46 = “F” 31 = “1” 30 = “0” 32 = “2” 41 = “A” 41 = “A” 20 = “ ” 20 = “ ”
Code flash memory area last address	Code flash memory area 00000H to 03FFFH (16 KB)	3 bytes	FF FF 00
Data flash memory area last address	Data flash memory area F1000H to F17FFH (2 KB)	3 bytes	FF 17 0F
Firmware version	Ver.1.23 → 01H, 02H, 03H)	3 bytes	01 02 03

Downloaded from Arrow.com.

24.5 Processing Time for Each Command When PG-FP5 Is in Use (Reference Value)

The following shows the processing time for each command (reference value) when PG-FP5 is used as a dedicated flash memory programmer.

<R> **Table 24-10. Processing Time for Each Command When PG-FP5 Is in Use (Reference Value)**

PG-FP5 Command	Code Flash					Unit
	2 Kbytes	4 Kbytes	8 Kbytes	12 Kbytes	16 Kbytes	
Erasing	0.5	0.5	1	1	1.5	S
Writing	1	1	1	1.5	1.5	S
Verification	1	1	1	1	1.5	S
Writing after erasing	1	1	1	1.5	2	S

Remark The command processing times (reference values) shown in the table are typical values under the following conditions.
Port: TOOL0 (single-line UART)
Speed: 1,000,000 bps
Mode: Full speed mode (flash operation mode: HS (high speed main) mode)

Downloaded from Arrow.com.

24.6 Self-Programming

The RL78 microcontroller supports a self-programming function that can be used to rewrite the code flash memory via a user program. Because this function allows a user application to rewrite the code flash memory by using the flash self-programming library, it can be used to upgrade the program in the field.

Cautions 1. To prohibit an interrupt during self-programming, in the same way as in the normal operation mode, execute the flash self-programming library in the state where the IE flag is cleared (0) by the DI instruction. To enable an interrupt, clear (0) the interrupt mask flag to accept in the state where the IE flag is set (1) by the EI instruction, and then execute the flash self-programming library.

2. The high-speed on-chip oscillator should be kept operating during self-programming. If it is kept stopping, the high-speed on-chip oscillator clock should be operated (HIOSTOP = 0). The flash self-programming library should be executed after 30 μs have elapsed.

3. The self-programming function cannot be used in the R5F10266 and R5F10366.

4. RL78/G12 does not support the boot swap function.

Remarks 1. For details of the self-programming function, refer to **RL78 Microcontroller Flash Self Programming Library Type01 User's Manual (R01AN0350)**.

2. For details of the time required to execute self programming, see the notes on use that accompany the flash self programming library tool.

The self-programming function has two flash memory programming modes; wide voltage mode and full speed mode.

Specify the mode that corresponds to the flash operation mode specified in bits CMODE1 and CMODE0 in option byte 000C2H.

Specify the full speed mode when the HS (high speed main) mode is specified. Specify the wide voltage mode when the LS (low speed main) mode is specified.

If the argument fsl_flash_voltage_u08 is 00H when the FSL_Init function of the flash self-programming library provided by Renesas Electronics is executed, full speed mode is specified. If the argument is other than 00H, the wide voltage mode is specified.

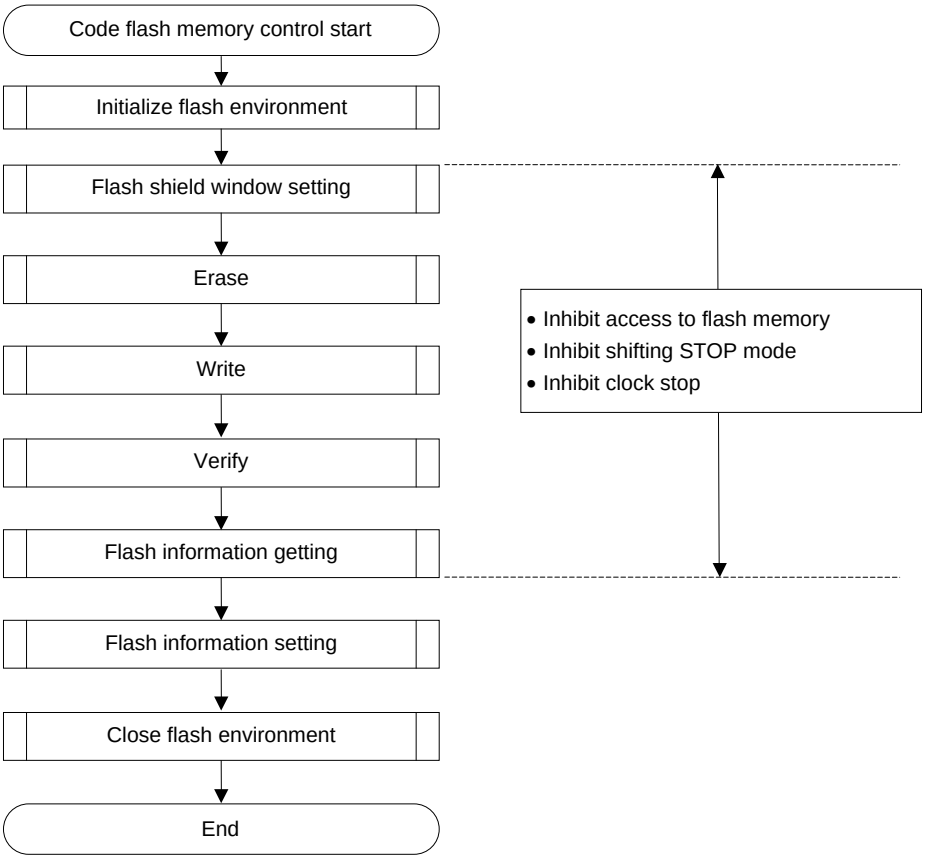
Remark Using both the wide voltage mode and full speed mode imposes no restrictions on writing, erasing, or verification.

Downloaded from Arrow.com.

24.6.1 Self-programming procedure

The following figure illustrates a flow for rewriting the code flash memory by using a flash self-programming library.

Figure 24-8. Flow of Self Programming (Rewriting Flash Memory)



Downloaded from Arrow.com.

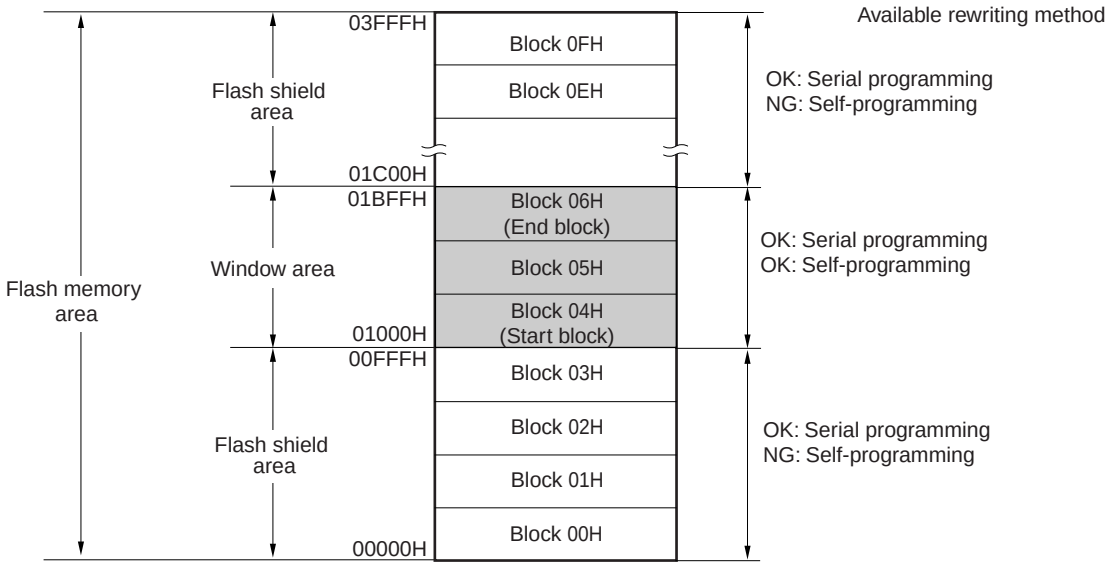
24.6.2 Flash shield window function

The flash shield window function is provided as one of the security functions for self-programming. It disables writing and erasing areas outside the range specified as window only during self-programming.

The window range can be set by specifying the start and end blocks. The window range can be set or changed by only serial programming.

Writing to and erasing areas outside the window range are disabled during self-programming. During serial programming, however, areas outside the range specified as a window can be written and erased.

Figure 24-9. An example of setting a flash shield window
(Target device: R5F1026A, start block: 04H, and end block: 06H)



Cautions 1. If the rewrite-prohibited area of the boot cluster 0 overlaps with the flash shield window range, prohibition to rewrite the boot cluster 0 takes priority.

2. The flash shield window can only be used for the code flash memory (and is not supported for the data flash memory).

Table 24-11. Relationship between Flash Shield Window Function Setting/Change Methods and Commands

Programming condition	Setting/Changing window range	Execution command	
		Block erasure	Writing
During serial programming	Specify the start block and end block of the window on the GUI of the dedicated flash memory programmer.	Block erasure can be done also outside the window range.	Writing can be done also outside the window range.

Note To prohibit writing and erasing during serial programming, refer to **24.7 Security Settings.**

Downloaded from Arrow.com.

<R> 24.7 Security Settings

The RL78 microcontroller supports a security function that prohibits rewriting the user program written to the internal flash memory, so that the program cannot be changed by an unauthorized person.

The operations shown below can be performed by using the Security Set command.

- Disabling block erase
  Execution of the block erase command for a specific block in the flash memory is prohibited during serial programming. However, blocks can be erased by means of self-programming.

- Disabling write
  Execution of the write command for entire blocks in the code flash memory is prohibited during serial programming. However, data can be written by means of self-programming.
  After the setting of prohibition of writing is specified, releasing the setting by the Security Release command is enabled by a reset.

- Disabling rewriting boot cluster 0
  Execution of the block erase command and write command on boot cluster 0 (00000H to 00FFFH) in the code flash memory is prohibited by this setting.

The block erase, write, and rewriting boot cluster0 commands are enabled by the default setting when the flash memory is shipped. Security can be set by serial programming and self-programming. The security settings can be used in combination.

Table 24-12 shows the relationship between the erase and write commands when the RL78 microcontroller security function is enabled.

Caution The security function of the dedicated flash programmer does not support self-programming.

Remark To prohibit writing and erasing during self-programming, use the flash shield window function (see **24.6.2** for detail).

Downloaded from Arrow.com.

Table 24-12. Relationship between Enabled Security Function and Commands

(1) During serial programming

Enabled Security Function	Executed Command	
	Block Erase	Write
Prohibition of block erasure	Blocks cannot be erased.	Can be performed[Note].
Prohibition of writing	Blocks can be erased.	Cannot be performed.
Prohibition of rewriting boot cluster 0	Boot cluster 0 cannot be erased.	Boot cluster 0 cannot be written.

Note Confirm that no data has been written to the write area. Because data cannot be erased when block erase is prohibited, do not write data if the data has not been erased.

(2) During self-programming

Enabled Security Function	Executed Command	
	Block Erase	Write
Prohibition of block erasure	Blocks can be erased.	Can be performed.
Prohibition of writing		
Prohibition of rewriting boot cluster 0	Boot cluster 0 cannot be erased.	Boot cluster 0 cannot be written.

Remark To prohibit writing and erasing during self-programming, use the flash shield window function (see **24.6.2** for detail).

Table 24-13. Security Setting in Each Programming Mode

Serial programming

Security	Security Setting	How to Disable Security Setting
Prohibition of block erasure	Set via GUI of dedicated flash memory programmer, etc.	Cannot be disabled after setting.
Prohibition of writing		Set via GUI of dedicated flash memory programmer, etc.
Prohibition of rewriting boot cluster 0		Cannot be disabled after setting.

Caution Releasing the setting of prohibition of writing is enabled only when the security is not set as the block erase prohibition and the boot cluster 0 rewrite prohibition with code flash memory area and data flash memory area being blanks.

Downloaded from Arrow.com.

<R> 24.8 Data Flash

24.8.1 Data flash overview

An overview of the data flash memory is provided below.

- The user program can rewrite the data flash memory by using the data flash library. For details, refer to **RL78 Family Data Flash Library User's Manual**.
- The data flash memory can also be rewritten to through serial programming using the dedicated flash memory programmer or an external device.
- The data flash can be erased in 1-block (1-Kbyte) units.
- The data flash can be accessed only in 8-bit units.
- The data flash can be directly read by CPU instructions.
- Instructions can be executed from the code flash memory while rewriting the data flash memory (that is, background operation (BGO) is supported).
- Because the data flash memory is an area exclusively used for data, it cannot be used to execute instructions.
- Accessing the data flash memory is not possible while rewriting the code flash memory (during self-programming).
- Manipulating the DFLCTL register is not possible while rewriting the data flash memory.
- Transition to the STOP mode is not possible while rewriting the data flash memory.

Cautions 1. The data flash memory is stopped after a reset is canceled. The data flash control register (DFLCTL) must be set up in order to use the data flash memory.

2. Interrupts are disabled during data flash rewrite for only the R5F10266. Execute the data flash library with the IE flag cleared (0) by the DI instruction.

3. The high-speed on-chip oscillator should be kept operating during data flash rewrite. If it is kept stopping, the high-speed on-chip oscillator clock should be operated (HIOSTOP = 0). The data flash library should be executed after 30 μs have elapsed.

Remark For rewriting the code flash memory via a user program, see **24.6 Self-Programming**.

24.8.2 Register controlling data flash memory

24.8.2.1 Data flash control register (DFLCTL)

This register is used to enable or disable accessing to the data flash.

The DFLCTL register is set by a 1-bit or 8-bit memory manipulation instruction.

Reset input sets this register to 00H.

Figure 24-10. Format of Data Flash Control Register (DFLCTL)

Address: F0090H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	<0>
DFLCTL	0	0	0	0	0	0	0	DFLEN

DFLEN	Data flash access control
0	Disables data flash access
1	Enables data flash access

Caution Manipulating the DFLCTL register is not possible while rewriting the data flash memory.

Downloaded from Arrow.com.

24.8.3 Procedure for accessing data flash memory

The data flash memory is stopped after a reset ends. To access the data flash, make initial settings according to the following procedure.

<1> Set bit 0 (DFLEN) of the data flash control register (DFLCTL) to 1.
<2> Wait for the setup to finish for software timer, etc.
The time setup takes differs for each flash operation mode for the main clock.
<Setup time for each flash operation mode>
- HS (High speed main): 5 μs
- LS (Low speed main): 720 ns

<3> After the wait, the data flash memory can be accessed.

Cautions 1. Accessing the data flash memory is not possible during the setup time.
2. Transition to the STOP mode is not possible during the setup time. To enter the STOP mode during the setup time, clear DFLEN to 0 and then execute the STOP instruction.
3. The high-speed on-chip oscillator should be kept operating during data flash rewrite. If it is kept stopping, the high-speed on-chip oscillator clock should be operated (HIOSTOP = 0). The data flash library should be executed after 30 μs have elapsed.

After initial setting, the data flash can be read through CPU instructions and can be read or rewritten to by using the data flash library.

Follow one of the procedures below when the DMA controller operates during access to the data flash memory.

(A) Hold DMA transfer pending or forcibly terminate it
Before reading the data flash memory, hold the DMA transfer pending in all the channels which are in use. The data flash memory should be read 3 clocks (f_{CLK}) or more after the DWAITn bit is set to 1. After reading the data flash memory, set the DWAITn bit to 0 and then cancel the pending status.
Or, before reading the data flash memory, forcibly terminate the DMA transfer in accordance with the process described in **15.5.5 Forced termination by software**. Resume the DMA transfer after reading the data flash memory.

(B) Access the data flash memory by using the library
Access the data flash memory by using the latest data flash library.

(C) Insert the NOP instruction
Insert the NOP instruction immediately before the data flash read instruction.
<Example>

```
MOVW  HL, !addr16  ; Read RAM
NOP                ; Insert the NOP instruction before reading the data flash memory
MOV  A,[DE]        ; Read the data flash memory
```

If high-level language like C language is used, the compiler may generate two instructions per code. At that time, the NOP instruction is not inserted immediately before the data flash read instruction. Read the data flash memory by following procedure (A) or (B).

Remark f_{CLK}: CPU/peripheral hardware clock frequency

Downloaded from Arrow.com.

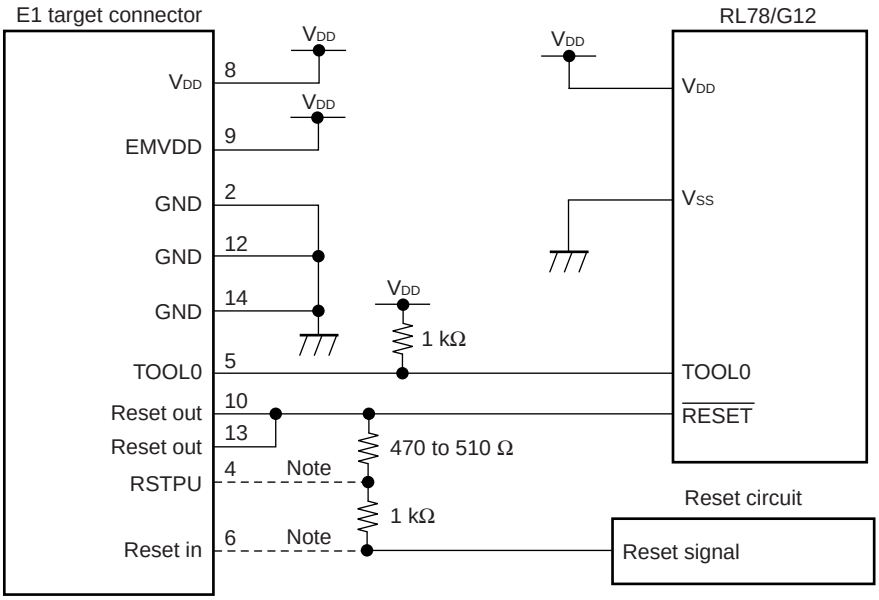
CHAPTER 25 ON-CHIP DEBUG FUNCTION

<R> 25.1 Connecting E1 On-chip Debugging Emulator

The RL78 microcontroller uses the V_{DD}, $\overline{RESET}$, TOOL0, and V_{SS} pins to communicate with the host machine via an E1 on-chip debugging emulator. Serial communication is performed by using a single-line UART that uses the TOOL0 pin.

Caution The RL78 microcontroller has an on-chip debug function, which is provided for development and evaluation. Do not use the on-chip debug function in products designated for mass production, because the guaranteed number of rewritable times of the flash memory may be exceeded when this function is used, and product reliability therefore cannot be guaranteed. Renesas Electronics is not liable for problems occurring when the on-chip debug function is used.

<R> **Figure 25-1. Connection Example of E1 On-chip Debugging Emulator and RL78/G12 (20, 24-pin products)**

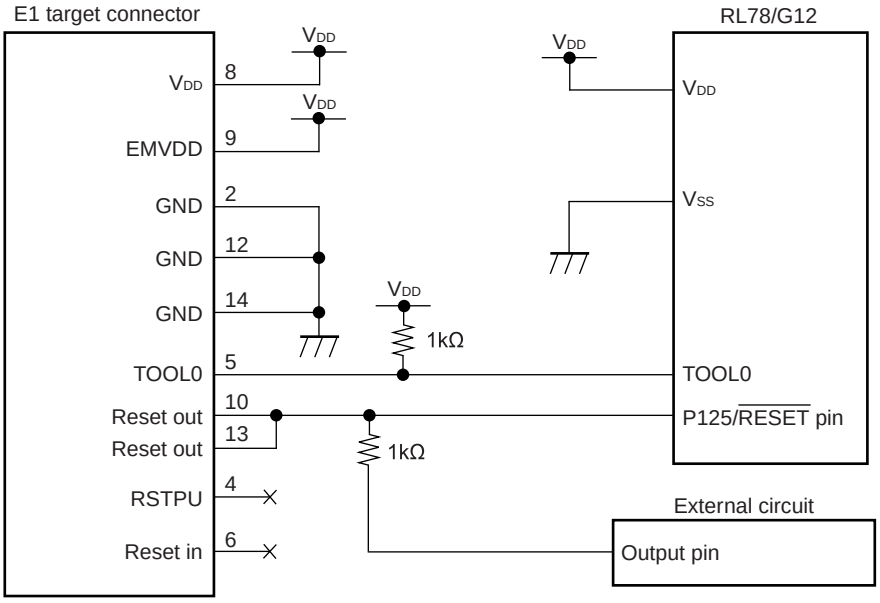


Note Connecting the dotted line is not necessary during flash programming.

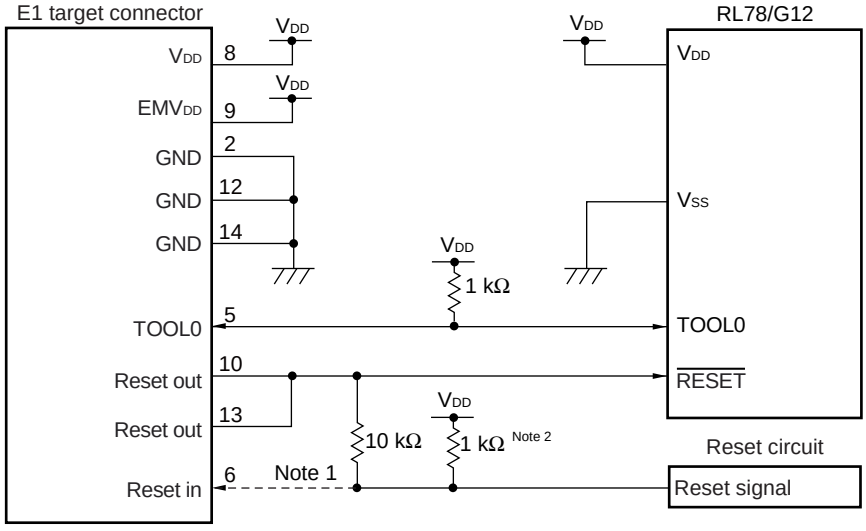
Downloaded from Arrow.com.

For the target system which uses the $\overline{\text{RESET}}$ pin as a port function (P125) in 20, 24-pin products, its connection to an external circuit should be isolated.

<R> **Figure 25-2. Connection Example of E1 On-chip Debugging Emulator and RL78/G12 (When using the $\overline{\text{RESET}}$ pin as a port function (P125))**



<R> **Figure 25-3. Connection Example of E1 On-chip Debugging Emulator and RL78/G12 (30-pin products)**



Notes 1. Connecting the dotted line is not necessary during serial programming.

2. If the reset circuit on the target system does not have a buffer and generates a reset signal only with resistors and capacitors, this pull-up resistor is not necessary.

Caution This circuit diagram is assumed that the reset signal outputs from an N-ch O.D. buffer (output resistor: 100 Ω or less)

Downloaded from Arrow.com.

25.2 On-Chip Debug Security ID

The RL78 microcontroller has an on-chip debug operation control bit in the flash memory at 000C3H (see **CHAPTER 23 OPTION BYTE**) and an on-chip debug security ID setting area at 000C4H to 000CDH, to prevent third parties from reading memory content.

Table 25-1. On-Chip Debug Security ID

Address	On-Chip Debug Security ID
000C4H to 000CDH	Any ID code of 10 bytes

25.3 Securing of User Resources

To perform communication between the RL78 microcontroller and E1 on-chip debugging emulator, as well as each debug function, the securing of memory space must be done beforehand.

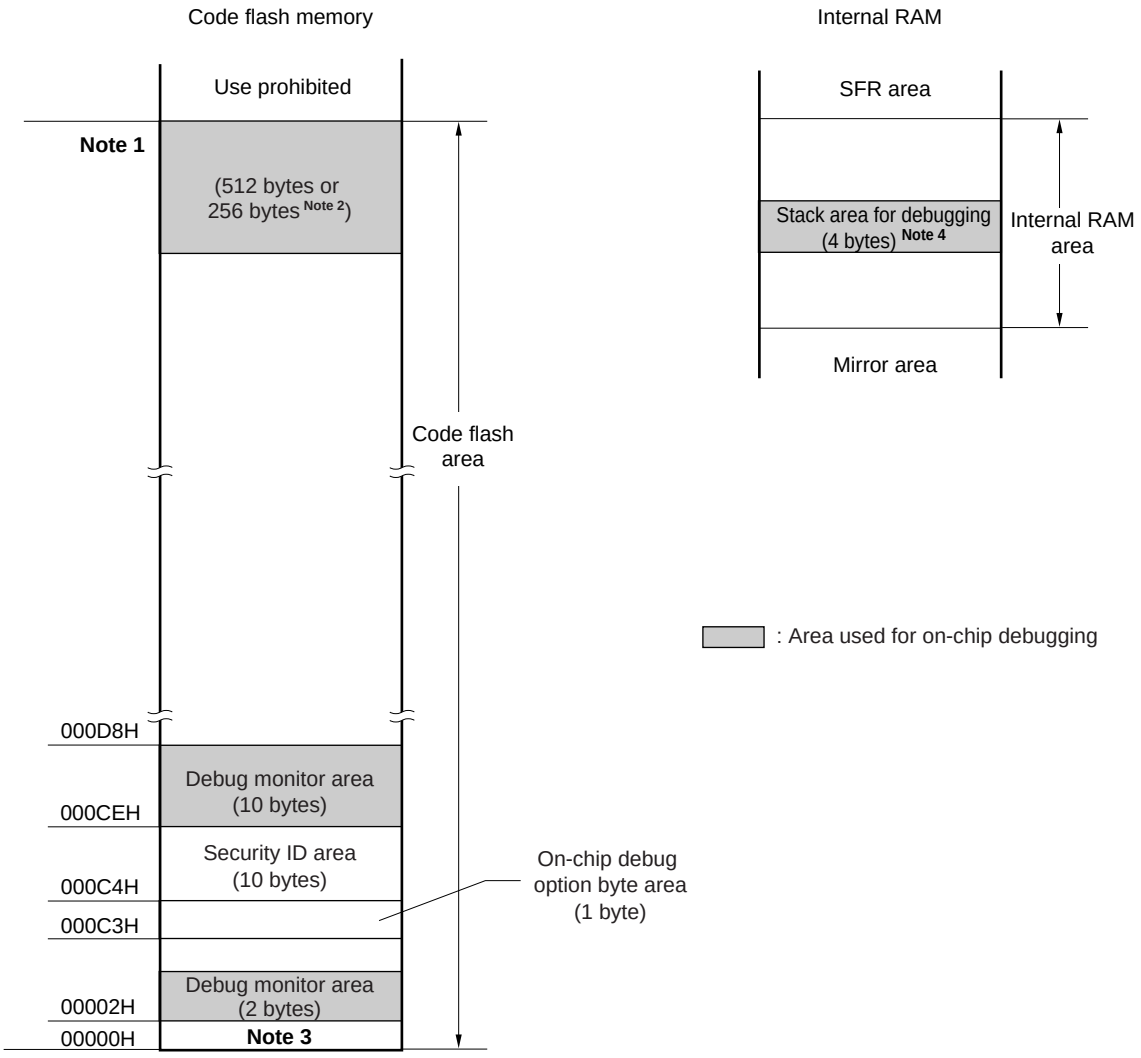
If Renesas Electronics assembler or compiler is used, the items can be set by using link options.

(1) Securement of memory space

The shaded portions in Figure 25-4 are the areas reserved for placing the debug monitor program, so user programs or data cannot be allocated in these spaces. When using the on-chip debug function, these spaces must be secured so as not to be used by the user program. Moreover, this area must not be rewritten by the user program.

Downloaded from Arrow.com.

Figure 25-4. Memory Spaces Where Debug Monitor Programs Are Allocated



Notes 1. Address differs depending on products as follows.

Products (code flash memory capacity)	Address of **Note 1**
R5F10266, R5F10366	007FFH
R5F10x67, R5F10x77, R5F10xA7	00FFFH
R5F10x68, R5F10x78, R5F10xA8	01FFFH
R5F10x69, R5F10x79, R5F10xA9	02FFFH
R5F10x6A, R5F10x7A, R5F10xAA	03FFFH

(x = 2, 3)

2. When real-time RAM monitor (RRM) function and dynamic memory modification (DMM) function are not used, it is 256 bytes.

3. In debugging, reset vector is rewritten to address allocated to a monitor program.

4. Since this area is allocated immediately before the stack area, the address of this area varies depending on the stack increase and decrease. That is, 4 extra bytes are consumed for the stack area used. When using self-programming, 12 extra bytes are consumed for the stack area used.

Downloaded from Arrow.com.

CHAPTER 26 BCD CORRECTION CIRCUIT

26.1 BCD Correction Circuit Function

The result of addition/subtraction of the BCD (binary-coded decimal) code and BCD code can be obtained as BCD code with this circuit.

The decimal correction operation result is obtained by performing addition/subtraction having the A register as the operand and then adding/ subtracting the BCD correction result register (BCDADJ).

26.2 Registers Used by BCD Correction Circuit

The BCD correction circuit uses the following registers.

- BCD correction result register (BCDADJ)

<R> 26.2.1 BCD correction result register (BCDADJ)

The BCDADJ register stores correction values for obtaining the add/subtract result as BCD code through add/subtract instructions using the A register as the operand.

The value read from the BCDADJ register varies depending on the value of the A register when it is read and those of the CY and AC flags.

The BCDADJ register is read by an 8-bit memory manipulation instruction.

Reset input sets this register to undefined.

Figure 26-1. Format of BCD Correction Result Register (BCDADJ)

Address: F00FEH After reset: undefined R

Symbol	7	6	5	4	3	2	1	0
BCDADJ								

Downloaded from Arrow.com.

26.3 BCD Correction Circuit Operation

The basic operation of the BCD correction circuit is as follows.

(1) Addition: Calculating the result of adding a BCD code value and another BCD code value by using a BCD code value

<1> The BCD code value to which addition is performed is stored in the A register.

<2> By adding the value of the A register and the second operand (value of one more BCD code to be added) as are in binary, the binary operation result is stored in the A register and the correction value is stored in the BCD correction result register (BCDADJ).

<3> Decimal correction is performed by adding in binary the value of the A register (addition result in binary) and the BCDADJ register (correction value), and the correction result is stored in the A register and CY flag.

Caution The value read from the BCDADJ register varies depending on the value of the A register when it is read and those of the CY and AC flags. Therefore, execute the instruction <3> after the instruction <2> instead of executing any other instructions. To perform BCD correction in the interrupt enabled state, saving and restoring the A register is required within the interrupt function. PSW (CY flag and AC flag) is restored by the RETI instruction.

An example is shown below.

Examples 1: 99 + 89 = 188

Instruction	A Register	CY Flag	AC Flag	BCDADJ Register
`MOV A, #99H ; <1>`	99H	–	–	–
`ADD A, #89H ; <2>`	22H	1	1	66H
`ADD A, !BCDADJ ; <3>`	88H	1	0	–

Examples 2: 85 + 15 = 100

Instruction	A Register	CY Flag	AC Flag	BCDADJ Register
`MOV A, #85H ; <1>`	85H	–	–	–
`ADD A, #15H ; <2>`	9AH	0	0	66H
`ADD A, !BCDADJ ; <3>`	00H	1	1	–

Examples 3: 80 + 80 = 160

Instruction	A Register	CY Flag	AC Flag	BCDADJ Register
`MOV A, #80H ; <1>`	80H	–	–	–
`ADD A, #80H ; <2>`	00H	1	0	60H
`ADD A, !BCDADJ ; <3>`	60H	1	0	–

Downloaded from Arrow.com.

(2) Subtraction: Calculating the result of subtracting a BCD code value from another BCD code value by using a BCD code value

<1> The BCD code value from which subtraction is performed is stored in the A register.

<2> By subtracting the value of the second operand (value of BCD code to be subtracted) from the A register as is in binary, the calculation result in binary is stored in the A register, and the correction value is stored in the BCD correction result register (BCDADJ).

<3> Decimal correction is performed by subtracting the value of the BCDADJ register (correction value) from the A register (subtraction result in binary) in binary, and the correction result is stored in the A register and CY flag.

Caution The value read from the BCDADJ register varies depending on the value of the A register when it is read and those of the CY and AC flags. Therefore, execute the instruction <3> after the instruction <2> instead of executing any other instructions. To perform BCD correction in the interrupt enabled state, saving and restoring the A register is required within the interrupt function. PSW (CY flag and AC flag) is restored by the RETI instruction.

An example is shown below.

Example: 91 – 52 = 39

Instruction	A Register	CY Flag	AC Flag	BCDADJ Register
`MOV A, #91H ; <1>`	91H	–	–	–
`SUB A, #52H ; <2>`	3FH	0	1	06H
`SUB A, !BCDADJ ; <3>`	39H	0	0	–

Downloaded from Arrow.com.

CHAPTER 27 INSTRUCTION SET

This chapter lists the instructions in the RL78 microcontroller instruction set. For details of each operation and operation code, refer to the separate document **RL78 Microcontrollers User's Manual: software (R01US0015E)**.

Downloaded from Arrow.com.

27.1 Conventions Used in Operation List

27.1.1 Operand identifiers and specification methods

Operands are described in the “Operand” column of each instruction in accordance with the description method of the instruction operand identifier (refer to the assembler specifications for details). When there are two or more description methods, select one of them. Alphabetic letters in capitals and the symbols, #, !, !!, $, $!, [ ], and ES: are keywords and are described as they are. Each symbol has the following meaning.

- #: Immediate data specification
- !: 16-bit absolute address specification
- !!: 20-bit absolute address specification
- $: 8-bit relative address specification
- $!: 16-bit relative address specification
- [ ]: Indirect address specification
- ES:: Extension address specification

In the case of immediate data, describe an appropriate numeric value or a label. When using a label, be sure to describe the #, !, !!, $, $!, [ ], and ES: symbols.

For operand register identifiers, r and rp, either function names (X, A, C, etc.) or absolute names (names in parentheses in Table 27-1 below, R0, R1, R2, etc.) can be used for description.

Table 27-1. Operand Identifiers and Specification Methods

Identifier	Description Method
r rp sfr sfrp	X (R0), A (R1), C (R2), B (R3), E (R4), D (R5), L (R6), H (R7) AX (RP0), BC (RP1), DE (RP2), HL (RP3) Special-function register symbol (SFR symbol) FFF00H to FFFFFH Special-function register symbols (16-bit manipulatable SFR symbol. Even addresses only[Note]) FFF00H to FFFFFH
saddr saddrp	FFE20H to FFF1FH Immediate data or labels FFE20H to FF1FH Immediate data or labels (even addresses only[Note])
addr20 addr16 addr5	00000H to FFFFFH Immediate data or labels 0000H to FFFFH Immediate data or labels (Automatically adds F to the top. only even addresses for 16-bit data transfer instructions[Note]) 0080H to 00BFH Immediate data or labels (specification to bits 5 to 1, even addresses only)
word byte bit	16-bit immediate data or label 8-bit immediate data or label 3-bit immediate data or label
RBn	RB0 to RB3

Note Bit 0 = 0 when an odd address is specified.

Remark The special function registers can be described to operand sfr as symbols. See **Table 3-6 SFR List** for the symbols of the special function registers. The extended special function registers can be described to operand !addr16 as symbols. See **Table 3-7 Extended SFR (2nd SFR) List** for the symbols of the extended special function registers.

Downloaded from Arrow.com.

27.1.2 Description of operation column

The operation when the instruction is executed is shown in the “Operation” column using the following symbols.

Table 27-2. Symbols in “Operation” Column

Symbol	Function
A	A register; 8-bit accumulator
X	X register
B	B register
C	C register
D	D register
E	E register
H	H register
L	L register
ES	ES register
CS	CS register
AX	AX register pair; 16-bit accumulator
BC	BC register pair
DE	DE register pair
HL	HL register pair
PC	Program counter
SP	Stack pointer
PSW	Program status word
CY	Carry flag
AC	Auxiliary carry flag
Z	Zero flag
RBS	Register bank select flag
IE	Interrupt request enable flag
()	Memory contents indicated by address or register contents in parentheses
X_H, X_L X_S, X_H, X_L	16-bit registers: X_H = higher 8 bits, X_L = lower 8 bits 20-bit registers: X_S = (bits 19 to 16), X_H = (bits 15 to 8), X_L = (bits 7 to 0)
∧	Logical product (AND)
∨	Logical sum (OR)
∨	Exclusive logical sum (exclusive OR)
¯	Inverted data
addr5	16-bit immediate data (even addresses only in 0080H to 00BFH)
addr16	16-bit immediate data
addr20	20-bit immediate data
jdisp8	Signed 8-bit data (displacement value)
jdisp16	Signed 16-bit data (displacement value)

Downloaded from Arrow.com.

27.1.3 Description of flag operation column

The change of the flag value when the instruction is executed is shown in the “Flag” column using the following symbols.

Table 27-3. Symbols in “Flag” Column

Symbol	Change of Flag Value
(Blank)	Unchanged
0	Cleared to 0
1	Set to 1
×	Set/cleared according to the result
R	Previously saved value is restored

27.1.4 PREFIX instruction

Instructions with “ES:” have a PREFIX operation code as a prefix to extend the accessible data area to the 1 MB space (00000H to FFFFFH), by adding the ES register value to the 64 KB space from F0000H to FFFFFH. When a PREFIX operation code is attached as a prefix to the target instruction, only one instruction immediately after the PREFIX operation code is executed as the addresses with the ES register value added.

A interrupt and DMA transfer are not acknowledged between a PREFIX instruction code and the instruction immediately after.

Table 27-4. Use Example of PREFIX Operation Code

Instruction	Opcode				
	1	2	3	4	5
MOV !addr16, #byte	CFH	!addr16		#byte	–
MOV ES:!addr16, #byte	11H	CFH	!addr16		#byte
MOV A, [HL]	8BH	–	–	–	–
MOV A, ES:[HL]	11H	8BH	–	–	–

Caution Set the ES register value with MOV ES, A, etc., before executing the PREFIX instruction.

Downloaded from Arrow.com.

27.2 Operation List

Table 27-5. Operation List (1/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit data transfer	MOV	r, #byte	2	1	–	r ← byte			
		PSW, #byte	3	3	–	PSW ← byte	×	×	×
		CS, #byte	3	1	–	CS ← byte			
		ES, #byte	2	1	–	ES ← byte			
		!addr16, #byte	4	1	–	(addr16) ← byte			
		ES:!addr16, #byte	5	2	–	(ES, addr16) ← byte			
		saddr, #byte	3	1	–	(saddr) ← byte			
		sfr, #byte	3	1	–	sfr ← byte			
		[DE+byte], #byte	3	1	–	(DE+byte) ← byte			
		ES:[DE+byte],#byte	4	2	–	((ES, DE)+byte) ← byte			
		[HL+byte], #byte	3	1	–	(HL+byte) ← byte			
		ES:[HL+byte],#byte	4	2	–	((ES, HL)+byte) ← byte			
		[SP+byte], #byte	3	1	–	(SP+byte) ← byte			
		word[B], #byte	4	1	–	(B+word) ← byte			
		ES:word[B], #byte	5	2	–	((ES, B)+word) ← byte			
		word[C], #byte	4	1	–	(C+word) ← byte			
		ES:word[C], #byte	5	2	–	((ES, C)+word) ← byte			
		word[BC], #byte	4	1	–	(BC+word) ← byte			
		ES:word[BC], #byte	5	2	–	((ES, BC)+word) ← byte			
		A, r Note 3	1	1	–	A ← r			
		r, A Note 3	1	1	–	r ← A			
		A, PSW	2	1	–	A ← PSW			
		PSW, A	2	3	–	PSW ← A	×	×	×
		A, CS	2	1	–	A ← CS			
		CS, A	2	1	–	CS ← A			
		A, ES	2	1	–	A ← ES			
		ES, A	2	1	–	ES ← A			
		A, !addr16	3	1	4	A ← (addr16)			
		A, ES:!addr16	4	2	5	A ← (ES, addr16)			
		!addr16, A	3	1	–	(addr16) ← A			
		ES:!addr16, A	4	2	–	(ES, addr16) ← A			
		A, saddr	2	1	–	A ← (saddr)			
		saddr, A	2	1	–	(saddr) ← A			

Notes 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

<R> **2.** Number of CPU clocks (f_{CLK}) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

3. Except r = A

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Downloaded from Arrow.com.

Table 27-5. Operation List (2/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag
				Note 1	Note 2		Z AC CY
8-bit data transfer	MOV	A, sfr	2	1	–	A ← sfr	
		sfr, A	2	1	–	sfr ← A	
		A, [DE]	1	1	4	A ← (DE)	
		[DE], A	1	1	–	(DE) ← A	
		A, ES:[DE]	2	2	5	A ← (ES, DE)	
		ES:[DE], A	2	2	–	(ES, DE) ← A	
		A, [HL]	1	1	4	A ← (HL)	
		[HL], A	1	1	–	(HL) ← A	
		A, ES:[HL]	2	2	5	A ← (ES, HL)	
		ES:[HL], A	2	2	–	(ES, HL) ← A	
		A, [DE+byte]	2	1	4	A ← (DE + byte)	
		[DE+byte], A	2	1	–	(DE + byte) ← A	
		A, ES:[DE+byte]	3	2	5	A ← ((ES, DE) + byte)	
		ES:[DE+byte], A	3	2	–	((ES, DE) + byte) ← A	
		A, [HL+byte]	2	1	4	A ← (HL + byte)	
		[HL+byte], A	2	1	–	(HL + byte) ← A	
		A, ES:[HL+byte]	3	2	5	A ← ((ES, HL) + byte)	
		ES:[HL+byte], A	3	2	–	((ES, HL) + byte) ← A	
		A, [SP+byte]	2	1	–	A ← (SP + byte)	
		[SP+byte], A	2	1	–	(SP + byte) ← A	
		A, word[B]	3	1	4	A ← (B + word)	
		word[B], A	3	1	–	(B + word) ← A	
		A, ES:word[B]	4	2	5	A ← ((ES, B) + word)	
		ES:word[B], A	4	2	–	((ES, B) + word) ← A	
		A, word[C]	3	1	4	A ← (C + word)	
		word[C], A	3	1	–	(C + word) ← A	
		A, ES:word[C]	4	2	5	A ← ((ES, C) + word)	
		ES:word[C], A	4	2	–	((ES, C) + word) ← A	
		A, word[BC]	3	1	4	A ← (BC + word)	
		word[BC], A	3	1	–	(BC + word) ← A	
		A, ES:word[BC]	4	2	5	A ← ((ES, BC) + word)	
		ES:word[BC], A	4	2	–	((ES, BC) + word) ← A	

Notes 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

<R> **2.** Number of CPU clocks (f_{CLK}) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Downloaded from Arrow.com.

Table 27-5. Operation List (3/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag
				Note 1	Note 2		Z AC CY
8-bit data transfer	MOV	A, [HL+B]	2	1	4	A ← (HL + B)	
		[HL+B], A	2	1	–	(HL + B) ← A	
		A, ES:[HL+B]	3	2	5	A ← ((ES, HL) + B)	
		ES:[HL+B], A	3	2	–	((ES, HL) + B) ← A	
		A, [HL+C]	2	1	4	A ← (HL + C)	
		[HL+C], A	2	1	–	(HL + C) ← A	
		A, ES:[HL+C]	3	2	5	A ← ((ES, HL) + C)	
		ES:[HL+C], A	3	2	–	((ES, HL) + C) ← A	
		X, !addr16	3	1	4	X ← (addr16)	
		X, ES:!addr16	4	2	5	X ← (ES, addr16)	
		X, saddr	2	1	–	X ← (saddr)	
		B, !addr16	3	1	4	B ← (addr16)	
		B, ES:!addr16	4	2	5	B ← (ES, addr16)	
		B, saddr	2	1	–	B ← (saddr)	
		C, !addr16	3	1	4	C ← (addr16)	
		C, ES:!addr16	4	2	5	C ← (ES, addr16)	
		C, saddr	2	1	–	C ← (saddr)	
		ES, saddr	3	1	–	ES ← (saddr)	
	XCH	A, r Note 3	1 (r = X) 2 (other than r = X)	1	–	A ←→ r	
		A, !addr16	4	2	–	A ←→ (addr16)	
		A, ES:!addr16	5	3	–	A ←→ (ES, addr16)	
		A, saddr	3	2	–	A ←→ (saddr)	
		A, sfr	3	2	–	A ←→ sfr	
		A, [DE]	2	2	–	A ←→ (DE)	
		A, ES:[DE]	3	3	–	A ←→ (ES, DE)	
		A, [HL]	2	2	–	A ←→ (HL)	
		A, ES:[HL]	3	3	–	A ←→ (ES, HL)	
		A, [DE+byte]	3	2	–	A ←→ (DE + byte)	
		A, ES:[DE+byte]	4	3	–	A ←→ ((ES, DE) + byte)	
		A, [HL+byte]	3	2	–	A ←→ (HL + byte)	
		A, ES:[HL+byte]	4	3	–	A ←→ ((ES, HL) + byte)	

Notes 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

<R> **2.** Number of CPU clocks (f_{CLK}) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

3. Except r = A

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Downloaded from Arrow.com.

Table 27-5. Operation List (4/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag
				Note 1	Note 2		Z AC CY
8-bit data transfer	XCH	A, [HL+B]	2	2	–	A ←→ (HL+B)	
		A, ES:[HL+B]	3	3	–	A ←→ ((ES, HL)+B)	
		A, [HL+C]	2	2	–	A ←→ (HL+C)	
		A, ES:[HL+C]	3	3	–	A ←→ ((ES, HL)+C)	
	ONEB	A	1	1	–	A ← 01H	
		X	1	1	–	X ← 01H	
		B	1	1	–	B ← 01H	
		C	1	1	–	C ← 01H	
		!addr16	3	1	–	(addr16) ← 01H	
		ES:!addr16	4	2	–	(ES, addr16) ← 01H	
		saddr	2	1	–	(saddr) ← 01H	
	CLRB	A	1	1	–	A ← 00H	
		X	1	1	–	X ← 00H	
		B	1	1	–	B ← 00H	
		C	1	1	–	C ← 00H	
		!addr16	3	1	–	(addr16) ← 00H	
		ES:!addr16	4	2	–	(ES,addr16) ← 00H	
		saddr	2	1	–	(saddr) ← 00H	
	MOVS	[HL+byte], X	3	1	–	(HL+byte) ← X	× ×
		ES:[HL+byte], X	4	2	–	(ES, HL+byte) ← X	× ×
16-bit data transfer	MOVW	rp, #word	3	1	–	rp ← word	
		saddrp, #word	4	1	–	(saddrp) ← word	
		sfrp, #word	4	1	–	sfrp ← word	
		AX, rp Note 3	1	1	–	AX ← rp	
		rp, AX Note 3	1	1	–	rp ← AX	
		AX, !addr16	3	1	4	AX ← (addr16)	
		!addr16, AX	3	1	–	(addr16) ← AX	
		AX, ES:!addr16	4	2	5	AX ← (ES, addr16)	
		ES:!addr16, AX	4	2	–	(ES, addr16) ← AX	
		AX, saddrp	2	1	–	AX ← (saddrp)	
		saddrp, AX	2	1	–	(saddrp) ← AX	
		AX, sfrp	2	1	–	AX ← sfrp	
		sfrp, AX	2	1	–	sfrp ← AX	

Notes 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

<R> **2.** Number of CPU clocks (f_{CLK}) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

3. Except rp = AX

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Downloaded from Arrow.com.

Table 27-5. Operation List (5/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag
				Note 1	Note 2		Z AC CY
16-bit data transfer	MOVW	AX, [DE]	1	1	4	AX ← (DE)	
		[DE], AX	1	1	–	(DE) ← AX	
		AX, ES:[DE]	2	2	5	AX ← (ES, DE)	
		ES:[DE], AX	2	2	–	(ES, DE) ← AX	
		AX, [HL]	1	1	4	AX ← (HL)	
		[HL], AX	1	1	–	(HL) ← AX	
		AX, ES:[HL]	2	2	5	AX ← (ES, HL)	
		ES:[HL], AX	2	2	–	(ES, HL) ← AX	
		AX, [DE+byte]	2	1	4	AX ← (DE+byte)	
		[DE+byte], AX	2	1	–	(DE+byte) ← AX	
		AX, ES:[DE+byte]	3	2	5	AX ← ((ES, DE) + byte)	
		ES:[DE+byte], AX	3	2	–	((ES, DE) + byte) ← AX	
		AX, [HL+byte]	2	1	4	AX ← (HL + byte)	
		[HL+byte], AX	2	1	–	(HL + byte) ← AX	
		AX, ES:[HL+byte]	3	2	5	AX ← ((ES, HL) + byte)	
		ES:[HL+byte], AX	3	2	–	((ES, HL) + byte) ← AX	
		AX, [SP+byte]	2	1	–	AX ← (SP + byte)	
		[SP+byte], AX	2	1	–	(SP + byte) ← AX	
		AX, word[B]	3	1	4	AX ← (B + word)	
		word[B], AX	3	1	–	(B+ word) ← AX	
		AX, ES:word[B]	4	2	5	AX ← ((ES, B) + word)	
		ES:word[B], AX	4	2	–	((ES, B) + word) ← AX	
		AX, word[C]	3	1	4	AX ← (C + word)	
		word[C], AX	3	1	–	(C + word) ← AX	
		AX, ES:word[C]	4	2	5	AX ← ((ES, C) + word)	
		ES:word[C], AX	4	2	–	((ES, C) + word) ← AX	
		AX, word[BC]	3	1	4	AX ← (BC + word)	
		word[BC], AX	3	1	–	(BC + word) ← AX	
		AX, ES:word[BC]	4	2	5	AX ← ((ES, BC) + word)	
		ES:word[BC], AX	4	2	–	((ES, BC) + word) ← AX	

Notes 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

<R> **2.** Number of CPU clocks (f_{CLK}) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Downloaded from Arrow.com.

Table 27-5. Operation List (6/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
16-bit data transfer	MOVW	BC, !addr16	3	1	4	BC ← (addr16)			
		BC, ES:!addr16	4	2	5	BC ← (ES, addr16)			
		DE, !addr16	3	1	4	DE ← (addr16)			
		DE, ES:!addr16	4	2	5	DE ← (ES, addr16)			
		HL, !addr16	3	1	4	HL ← (addr16)			
		HL, ES:!addr16	4	2	5	HL ← (ES, addr16)			
		BC, saddrp	2	1	–	BC ← (saddrp)			
		DE, saddrp	2	1	–	DE ← (saddrp)			
		HL, saddrp	2	1	–	HL ← (saddrp)			
	XCHW	AX, rp Note 3	1	1	–	AX ←→ rp			
	ONEW	AX	1	1	–	AX ← 0001H			
		BC	1	1	–	BC ← 0001H			
	CLRW	AX	1	1	–	AX ← 0000H			
		BC	1	1	–	BC ← 0000H			
8-bit operation	ADD	A, #byte	2	1	–	A, CY ← A + byte	×	×	×
		saddr, #byte	3	2	–	(saddr), CY ← (saddr)+byte	×	×	×
		A, r Note 4	2	1	–	A, CY ← A + r	×	×	×
		r, A	2	1	–	r, CY ← r + A	×	×	×
		A, !addr16	3	1	4	A, CY ← A + (addr16)	×	×	×
		A, ES:!addr16	4	2	5	A, CY ← A + (ES, addr16)	×	×	×
		A, saddr	2	1	–	A, CY ← A + (saddr)	×	×	×
		A, [HL]	1	1	4	A, CY ← A+ (HL)	×	×	×
		A, ES:[HL]	2	2	5	A,CY ← A + (ES, HL)	×	×	×
		A, [HL+byte]	2	1	4	A, CY ← A + (HL+byte)	×	×	×
		A, ES:[HL+byte]	3	2	5	A,CY ← A + ((ES, HL)+byte)	×	×	×
		A, [HL+B]	2	1	4	A, CY ← A + (HL+B)	×	×	×
		A, ES:[HL+B]	3	2	5	A,CY ← A+((ES, HL)+B)	×	×	×
		A, [HL+C]	2	1	4	A, CY ← A + (HL+C)	×	×	×
		A, ES:[HL+C]	3	2	5	A,CY ← A + ((ES, HL) + C)	×	×	×

Notes 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

<R> **2.** Number of CPU clocks (f_{CLK}) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

3. Except rp = AX

4. Except r = A

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Downloaded from Arrow.com.

Table 27-5. Operation List (7/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit operation	ADDC	A, #byte	2	1	–	A, CY ← A+byte+CY	×	×	×
		saddr, #byte	3	2	–	(saddr), CY ← (saddr) +byte+CY	×	×	×
		A, rv Note 3	2	1	–	A, CY ← A + r + CY	×	×	×
		r, A	2	1	–	r, CY ← r + A + CY	×	×	×
		A, !addr16	3	1	4	A, CY ← A + (addr16)+CY	×	×	×
		A, ES:!addr16	4	2	5	A, CY ← A + (ES, addr16)+CY	×	×	×
		A, saddr	2	1	–	A, CY ← A + (saddr)+CY	×	×	×
		A, [HL]	1	1	4	A, CY ← A+ (HL) + CY	×	×	×
		A, ES:[HL]	2	2	5	A,CY ← A+ (ES, HL) + CY	×	×	×
		A, [HL+byte]	2	1	4	A, CY ← A+ (HL+byte) + CY	×	×	×
		A, ES:[HL+byte]	3	2	5	A,CY ← A+ ((ES, HL)+byte) + CY	×	×	×
		A, [HL+B]	2	1	4	A, CY ← A+ (HL+B) +CY	×	×	×
		A, ES:[HL+B]	3	2	5	A,CY ← A+((ES, HL)+B)+CY	×	×	×
		A, [HL+C]	2	1	4	A, CY ← A+ (HL+C)+CY	×	×	×
		A, ES:[HL+C]	3	2	5	A,CY ← A+ ((ES, HL)+C)+CY	×	×	×
	SUB	A, #byte	2	1	–	A, CY ← A – byte	×	×	×
		saddr, #byte	3	2	–	(saddr), CY ← (saddr) – byte	×	×	×
		A, r Note 3	2	1	–	A, CY ← A – r	×	×	×
		r, A	2	1	–	r, CY ← r – A	×	×	×
		A, !addr16	3	1	4	A, CY ← A – (addr16)	×	×	×
		A, ES:!addr16	4	2	5	A, CY ← A – (ES, addr16)	×	×	×
		A, saddr	2	1	–	A, CY ← A – (saddr)	×	×	×
		A, [HL]	1	1	4	A, CY ← A – (HL)	×	×	×
		A, ES:[HL]	2	2	5	A,CY ← A – (ES, HL)	×	×	×
		A, [HL+byte]	2	1	4	A, CY ← A – (HL+byte)	×	×	×
		A, ES:[HL+byte]	3	2	5	A,CY ← A – ((ES, HL)+byte)	×	×	×
		A, [HL+B]	2	1	4	A, CY ← A – (HL+B)	×	×	×
		A, ES:[HL+B]	3	2	5	A,CY ← A – ((ES, HL)+B)	×	×	×
		A, [HL+C]	2	1	4	A, CY ← A – (HL+C)	×	×	×
		A, ES:[HL+C]	3	2	5	A,CY ← A – ((ES, HL)+C)	×	×	×

Notes 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

<R> **2.** Number of CPU clocks (f_{CLK}) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

3. Except r = A

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Downloaded from Arrow.com.

Table 27-5. Operation List (8/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit operation	SUBC	A, #byte	2	1	–	A, CY ← A – byte – CY	×	×	×
		saddr, #byte	3	2	–	(saddr), CY ← (saddr) – byte – CY	×	×	×
		A, r Note 3	2	1	–	A, CY ← A – r – CY	×	×	×
		r, A	2	1	–	r, CY ← r – A – CY	×	×	×
		A, !addr16	3	1	4	A, CY ← A – (addr16) – CY	×	×	×
		A, ES:!addr16	4	2	5	A, CY ← A – (ES, addr16) – CY	×	×	×
		A, saddr	2	1	–	A, CY ← A – (saddr) – CY	×	×	×
		A, [HL]	1	1	4	A, CY ← A – (HL) – CY	×	×	×
		A, ES:[HL]	2	2	5	A,CY ← A – (ES, HL) – CY	×	×	×
		A, [HL+byte]	2	1	4	A, CY ← A – (HL+byte) – CY	×	×	×
		A, ES:[HL+byte]	3	2	5	A,CY ← A – ((ES, HL)+byte) – CY	×	×	×
		A, [HL+B]	2	1	4	A, CY ← A – (HL+B) – CY	×	×	×
		A, ES:[HL+B]	3	2	5	A,CY ← A – ((ES, HL)+B) – CY	×	×	×
		A, [HL+C]	2	1	4	A, CY ← A – (HL+C) – CY	×	×	×
		A, ES:[HL+C]	3	2	5	A, CY ← A – ((ES:HL)+C) – CY	×	×	×
	AND	A, #byte	2	1	–	A ← A ∧ byte	×		
		saddr, #byte	3	2	–	(saddr) ← (saddr) ∧ byte	×		
		A, r Note 3	2	1	–	A ← A ∧ r	×		
		r, A	2	1	–	R ← r ∧ A	×		
		A, !addr16	3	1	4	A ← A ∧ (addr16)	×		
		A, ES:!addr16	4	2	5	A ← A ∧ (ES:addr16)	×		
		A, saddr	2	1	–	A ← A ∧ (saddr)	×		
		A, [HL]	1	1	4	A ← A ∧ (HL)	×		
		A, ES:[HL]	2	2	5	A ← A ∧ (ES:HL)	×		
		A, [HL+byte]	2	1	4	A ← A ∧ (HL+byte)	×		
		A, ES:[HL+byte]	3	2	5	A ← A ∧ ((ES:HL)+byte)	×		
		A, [HL+B]	2	1	4	A ← A ∧ (HL+B)	×		
		A, ES:[HL+B]	3	2	5	A ← A ∧ ((ES:HL)+B)	×		
		A, [HL+C]	2	1	4	A ← A ∧ (HL+C)	×		
		A, ES:[HL+C]	3	2	5	A ← A ∧ ((ES:HL)+C)	×		

Notes 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

<R> **2.** Number of CPU clocks (f_{CLK}) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

3. Except r = A

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Downloaded from Arrow.com.

Table 27-5. Operation List (9/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag
				Note 1	Note 2		Z AC CY
8-bit operation	OR	A, #byte	2	1	–	A ← A∨byte	×
		saddr, #byte	3	2	–	(saddr) ← (saddr)∨byte	×
		A, r Note 3	2	1	–	A ← A∨r	×
		r, A	2	1	–	r ← r∨A	×
		A, !addr16	3	1	4	A ← A∨(addr16)	×
		A, ES:!addr16	4	2	5	A ← A∨(ES:addr16)	×
		A, saddr	2	1	–	A ← A∨(saddr)	×
		A, [HL]	1	1	4	A ← A∨(H)	×
		A, ES:[HL]	2	2	5	A ← A∨(ES:HL)	×
		A, [HL+byte]	2	1	4	A ← A∨(HL+byte)	×
		A, ES:[HL+byte]	3	2	5	A ← A∨((ES:HL)+byte)	×
		A, [HL+B]	2	1	4	A ← A∨(HL+B)	×
		A, ES:[HL+B]	3	2	5	A ← A∨((ES:HL)+B)	×
		A, [HL+C]	2	1	4	A ← A∨(HL+C)	×
		A, ES:[HL+C]	3	2	5	A ← A∨((ES:HL)+C)	×
	XOR	A, #byte	2	1	–	A ← A$\overline{\vee}$byte	×
		saddr, #byte	3	2	–	(saddr) ← (saddr)$\overline{\vee}$byte	×
		A, r Note 3	2	1	–	A ← A$\overline{\vee}$r	×
		r, A	2	1	–	r ← r$\overline{\vee}$A	×
		A, !addr16	3	1	4	A ← A$\overline{\vee}$(addr16)	×
		A, ES:!addr16	4	2	5	A ← A$\overline{\vee}$(ES:addr16)	×
		A, saddr	2	1	–	A ← A$\overline{\vee}$(saddr)	×
		A, [HL]	1	1	4	A ← A$\overline{\vee}$(HL)	×
		A, ES:[HL]	2	2	5	A ← A$\overline{\vee}$(ES:HL)	×
		A, [HL+byte]	2	1	4	A ← A$\overline{\vee}$(HL+byte)	×
		A, ES:[HL+byte]	3	2	5	A ← A$\overline{\vee}$((ES:HL)+byte)	×
		A, [HL+B]	2	1	4	A ← A$\overline{\vee}$(HL+B)	×
		A, ES:[HL+B]	3	2	5	A ← A$\overline{\vee}$((ES:HL)+B)	×
		A, [HL+C]	2	1	4	A ← A$\overline{\vee}$(HL+C)	×
		A, ES:[HL+C]	3	2	5	A ← A$\overline{\vee}$((ES:HL)+C)	×

Notes 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

<R> **2.** Number of CPU clocks (f_{CLK}) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

3. Except r = A

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Downloaded from Arrow.com.

Table 27-5. Operation List (10/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit operation	CMP	A, #byte	2	1	–	A – byte	×	×	×
		!addr16, #byte	4	1	4	(addr16) – byte	×	×	×
		ES:!addr16, #byte	5	2	5	(ES:addr16) – byte	×	×	×
		saddr, #byte	3	1	–	(saddr) – byte	×	×	×
		A, r Note3	2	1	–	A – r	×	×	×
		r, A	2	1	–	r – A	×	×	×
		A, !addr16	3	1	4	A – (addr16)	×	×	×
		A, ES:!addr16	4	2	5	A – (ES:addr16)	×	×	×
		A, saddr	2	1	–	A – (saddr)	×	×	×
		A, [HL]	1	1	4	A – (HL)	×	×	×
		A, ES:[HL]	2	2	5	A – (ES:HL)	×	×	×
		A, [HL+byte]	2	1	4	A – (HL+byte)	×	×	×
		A, ES:[HL+byte]	3	2	5	A – ((ES:HL)+byte)	×	×	×
		A, [HL+B]	2	1	4	A – (HL+B)	×	×	×
		A, ES:[HL+B]	3	2	5	A – ((ES:HL)+B)	×	×	×
		A, [HL+C]	2	1	4	A – (HL+C)	×	×	×
		A, ES:[HL+C]	3	2	5	A – ((ES:HL)+C)	×	×	×
	CMP0	A	1	1	–	A – 00H	×	0	0
		X	1	1	–	X – 00H	×	0	0
		B	1	1	–	B – 00H	×	0	0
		C	1	1	–	C – 00H	×	0	0
		!addr16	3	1	4	(addr16) – 00H	×	0	0
		ES:!addr16	4	2	5	(ES:addr16) – 00H	×	0	0
		saddr	2	1	–	(saddr) – 00H	×	0	0
	CMPS	X, [HL+byte]	3	1	4	X – (HL+byte)	×	×	×
		X, ES:[HL+byte]	4	2	5	X – ((ES:HL)+byte)	×	×	×

Notes 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

<R> **2.** Number of CPU clocks (f_{CLK}) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

3. Except r = A

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Downloaded from Arrow.com.

Table 27-5. Operation List (11/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
16-bit operation	ADDW	AX, #word	3	1	–	AX, CY ← AX+word	×	×	×
		AX, AX	1	1	–	AX, CY ← AX+AX	×	×	×
		AX, BC	1	1	–	AX, CY ← AX+BC	×	×	×
		AX, DE	1	1	–	AX, CY ← AX+DE	×	×	×
		AX, HL	1	1	–	AX, CY ← AX+HL	×	×	×
		AX, !addr16	3	1	4	AX, CY ← AX+(addr16)	×	×	×
		AX, ES:!addr16	4	2	5	AX, CY ← AX+(ES:addr16)	×	×	×
		AX, saddrp	2	1	–	AX, CY ← AX+(saddrp)	×	×	×
		AX, [HL+byte]	3	1	4	AX, CY ← AX+(HL+byte)	×	×	×
		AX, ES: [HL+byte]	4	2	5	AX, CY ← AX+((ES:HL)+byte)	×	×	×
	SUBW	AX, #word	3	1	–	AX, CY ← AX – word	×	×	×
		AX, BC	1	1	–	AX, CY ← AX – BC	×	×	×
		AX, DE	1	1	–	AX, CY ← AX – DE	×	×	×
		AX, HL	1	1	–	AX, CY ← AX – HL	×	×	×
		AX, !addr16	3	1	4	AX, CY ← AX – (addr16)	×	×	×
		AX, ES:!addr16	4	2	5	AX, CY ← AX – (ES:addr16)	×	×	×
		AX, saddrp	2	1	–	AX, CY ← AX – (saddrp)	×	×	×
		AX, [HL+byte]	3	1	4	AX, CY ← AX – (HL+byte)	×	×	×
		AX, ES: [HL+byte]	4	2	5	AX, CY ← AX – ((ES:HL)+byte)	×	×	×
	CMPW	AX, #word	3	1	–	AX – word	×	×	×
		AX, BC	1	1	–	AX – BC	×	×	×
		AX, DE	1	1	–	AX – DE	×	×	×
		AX, HL	1	1	–	AX – HL	×	×	×
		AX, !addr16	3	1	4	AX – (addr16)	×	×	×
		AX, ES:!addr16	4	2	5	AX – (ES:addr16)	×	×	×
		AX, saddrp	2	1	–	AX – (saddrp)	×	×	×
		AX, [HL+byte]	3	1	4	AX – (HL+byte)	×	×	×
		AX, ES: [HL+byte]	4	2	5	AX – ((ES:HL)+byte)	×	×	×
Multiply	MULU	X	1	1	–	AX ← A×X			

Notes 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

<R> **2.** Number of CPU clocks (f_{CLK}) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Downloaded from Arrow.com.

Table 27-5. Operation List (12/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
Increment/ decrement	INC	r	1	1	–	r ← r+1	×	×	
		!addr16	3	2	–	(addr16) ← (addr16)+1	×	×	
		ES:!addr16	4	3	–	(ES, addr16) ← (ES, addr16)+1	×	×	
		saddr	2	2	–	(saddr) ← (saddr)+1	×	×	
		[HL+byte]	3	2	–	(HL+byte) ← (HL+byte)+1	×	×	
		ES: [HL+byte]	4	3	–	((ES:HL)+byte) ← ((ES:HL)+byte)+1	×	×	
	DEC	r	1	1	–	r ← r – 1	×	×	
		!addr16	3	2	–	(addr16) ← (addr16) – 1	×	×	
		ES:!addr16	4	3	–	(ES, addr16) ← (ES, addr16) – 1	×	×	
		saddr	2	2	–	(saddr) ← (saddr) – 1	×	×	
		[HL+byte]	3	2	–	(HL+byte) ← (HL+byte) – 1	×	×	
		ES: [HL+byte]	4	3	–	((ES:HL)+byte) ← ((ES:HL)+byte) – 1	×	×	
	INCW	rp	1	1	–	rp ← rp+1			
		!addr16	3	2	–	(addr16) ← (addr16)+1			
		ES:!addr16	4	3	–	(ES, addr16) ← (ES, addr16)+1			
		saddrp	2	2	–	(saddrp) ← (saddrp)+1			
		[HL+byte]	3	2	–	(HL+byte) ← (HL+byte)+1			
		ES: [HL+byte]	4	3	–	((ES:HL)+byte) ← ((ES:HL)+byte)+1			
	DECW	rp	1	1	–	rp ← rp – 1			
		!addr16	3	2	–	(addr16) ← (addr16) – 1			
		ES:!addr16	4	3	–	(ES, addr16) ← (ES, addr16) – 1			
		saddrp	2	2	–	(saddrp) ← (saddrp) – 1			
		[HL+byte]	3	2	–	(HL+byte) ← (HL+byte) – 1			
		ES: [HL+byte]	4	3	–	((ES:HL)+byte) ← ((ES:HL)+byte) – 1			
Shift	SHR	A, cnt	2	1	–	(CY ← A_0, A_{m-1} ← A_m, A_7 ← 0) ×cnt			×
	SHRW	AX, cnt	2	1	–	(CY ← AX_0, AX_{m-1} ← AX_m, AX_{15} ← 0) ×cnt			×
	SHL	A, cnt	2	1	–	(CY ← A_7, A_m ← A_{m-1}, A_0 ← 0) ×cnt			×
		B, cnt	2	1	–	(CY ← B_7, B_m ← B_{m-1}, B_0 ← 0) ×cnt			×
		C, cnt	2	1	–	(CY ← C_7, C_m ← C_{m-1}, C_0 ← 0) ×cnt			×
	SHLW	AX, cnt	2	1	–	(CY ← AX_{15}, AX_m ← AX_{m-1}, AX_0 ← 0) ×cnt			×
		BC, cnt	2	1	–	(CY ← BC_{15}, BC_m ← BC_{m-1}, BC_0 ← 0) ×cnt			×
	SAR	A, cnt	2	1	–	(CY ← A_0, A_{m-1} ← A_m, A_7 ← A_7) ×cnt			×
	SARW	AX, cnt	2	1	–	(CY ← AX_0, AX_{m-1} ← AX_m, AX_{15}← AX_{15}) ×cnt			×

Notes 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

<R> **2.** Number of CPU clocks (f_{CLK}) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

Remarks 1. Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

2. cnt indicates the bit shift count.

Downloaded from Arrow.com.

Table 27-5. Operation List (13/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
Rotate	ROR	A, 1	2	1	–	(CY, $A_7 \leftarrow A_0$, $A_{m-1} \leftarrow A_m$)×1			×
	ROL	A, 1	2	1	–	(CY, $A_0 \leftarrow A_7$, $A_{m+1} \leftarrow A_m$)×1			×
	RORC	A, 1	2	1	–	(CY $\leftarrow A_0$, $A_7 \leftarrow$ CY, $A_{m-1} \leftarrow A_m$)×1			×
	ROLC	A, 1	2	1	–	(CY $\leftarrow A_7$, $A_0 \leftarrow$ CY, $A_{m+1} \leftarrow A_m$)×1			×
	ROLWC	AX,1	2	1	–	(CY $\leftarrow AX_{15}$, $AX_0 \leftarrow$ CY, $AX_{m+1} \leftarrow AX_m$) ×1			×
		BC,1	2	1	–	(CY $\leftarrow BC_{15}$, $BC_0 \leftarrow$ CY, $BC_{m+1} \leftarrow BC_m$) ×1			×
Bit manipulate	MOV1	CY, A.bit	2	1	–	CY ← A.bit			×
		A.bit, CY	2	1	–	A.bit ← CY			
		CY, PSW.bit	3	1	–	CY ← PSW.bit			×
		PSW.bit, CY	3	4	–	PSW.bit ← CY	×	×	
		CY, saddr.bit	3	1	–	CY ← (saddr).bit			×
		saddr.bit, CY	3	2	–	(saddr).bit ← CY			
		CY, sfr.bit	3	1	–	CY ← sfr.bit			×
		sfr.bit, CY	3	2	–	sfr.bit ← CY			
		CY,[HL].bit	2	1	4	CY ← (HL).bit			×
		[HL].bit, CY	2	2	–	(HL).bit ← CY			
		CY, ES:[HL].bit	3	2	5	CY ← (ES, HL).bit			×
		ES:[HL].bit, CY	3	3	–	(ES, HL).bit ← CY			
	AND1	CY, A.bit	2	1	–	CY ← CY ∧ A.bit			×
		CY, PSW.bit	3	1	–	CY ← CY ∧ PSW.bit			×
		CY, saddr.bit	3	1	–	CY ← CY ∧ (saddr).bit			×
		CY, sfr.bit	3	1	–	CY ← CY ∧ sfr.bit			×
		CY,[HL].bit	2	1	4	CY ← CY ∧ (HL).bit			×
		CY, ES:[HL].bit	3	2	5	CY ← CY ∧ (ES, HL).bit			×
	OR1	CY, A.bit	2	1	–	CY ← CY ∨ A.bit			×
		CY, PSW.bit	3	1	–	CYX ← CY ∨ ∨PSW.bit			×
		CY, saddr.bit	3	1	–	CY ← CY ∨ (saddr).bit			×
		CY, sfr.bit	3	1	–	CY ← CY ∨ sfr.bit			×
		CY, [HL].bit	2	1	4	CY ← CY ∨ (HL).bit			×
		CY, ES:[HL].bit	3	2	5	CY ← CY ∨ (ES, HL).bit			×

Notes 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

<R> **2.** Number of CPU clocks (f_{CLK}) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Downloaded from Arrow.com.

Table 27-5. Operation List (14/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
Bit manipulate	XOR1	CY, A.bit	2	1	–	CY ← CY ⊻ A.bit			×
		CY, PSW.bit	3	1	–	CY ← CY ⊻ PSW.bit			×
		CY, saddr.bit	3	1	–	CY ← CY ⊻ (saddr).bit			×
		CY, sfr.bit	3	1	–	CY ← CY ⊻ sfr.bit			×
		CY, [HL].bit	2	1	4	CY ← CY ⊻ (HL).bit			×
		CY, ES:[HL].bit	3	2	5	CY ← CY ⊻ (ES, HL).bit			×
	SET1	A.bit	2	1	–	A.bit ← 1			
		PSW.bit	3	4	–	PSW.bit ← 1	×	×	×
		!addr16.bit	4	2	–	(addr16).bit ← 1			
		ES:!addr16.bit	5	3	–	(ES, addr16).bit ← 1			
		saddr.bit	3	2	–	(saddr).bit ← 1			
		sfr.bit	3	2	–	sfr.bit ← 1			
		[HL].bit	2	2	–	(HL).bit ← 1			
		ES:[HL].bit	3	3	–	(ES, HL).bit ← 1			
	CLR1	A.bit	2	1	–	A.bit ← 0			
		PSW.bit	3	4	–	PSW.bit ← 0	×	×	×
		!addr16.bit	4	2	–	(addr16).bit ← 0			
		ES:!addr16.bit	5	3	–	(ES, addr16).bit ← 0			
		saddr.bit	3	2	–	(saddr.bit) ← 0			
		sfr.bit	3	2	–	sfr.bit ← 0			
		[HL].bit	2	2	–	(HL).bit ← 0			
		ES:[HL].bit	3	3	–	(ES, HL).bit ← 0			
	SET1	CY	2	1	–	CY ← 1			1
	CLR1	CY	2	1	–	CY ← 0			0
	NOT1	CY	2	1	–	CY ← $\overline{CY}$			×

Notes 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

<R> **2.** Number of CPU clocks (f_{CLK}) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Downloaded from Arrow.com.

Table 27-5. Operation List (15/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag
				Note 1	Note 2		Z AC CY
Call/ return	CALL	rp	2	3	–	(SP – 2) ← (PC+2)S, (SP – 3) ← (PC+2)H, (SP – 4) ← (PC+2)L, PC ← CS, rp, SP ← SP – 4	
		$!addr20	3	3	–	(SP – 2) ← (PC+3)S, (SP – 3) ← (PC+3)H, (SP – 4) ← (PC+3)L, PC ← PC+3+jdisp16, SP ← SP – 4	
		!addr16	3	3	–	(SP – 2) ← (PC+3)S, (SP – 3) ← (PC+3)H, (SP – 4) ← (PC+3)L, PC ← 0000, addr16, SP ← SP – 4	
		!!addr20	4	3	–	(SP – 2) ← (PC+4)S, (SP – 3) ← (PC+4)H, (SP – 4) ← (PC+4)L, PC ← addr20, SP ← SP – 4	
	CALLT	[addr5]	2	5	–	(SP – 2) ← (PC+2)S , (SP – 3) ← (PC+2)H, (SP – 4) ← (PC+2)L , PCS ← 0000, PCH ← (0000, addr5+1), PCL ← (0000, addr5), SP ← SP – 4	
	BRK	-	2	5	–	(SP – 1) ← PSW, (SP – 2) ← (PC+2)S, (SP – 3) ← (PC+2)H, (SP – 4) ← (PC+2)L, PCS ← 0000, PCH ← (0007FH), PCL ← (0007EH), SP ← SP – 4, IE ← 0	
	RET	-	1	6	–	PCL ← (SP), PCH ← (SP+1), PCS ← (SP+2), SP ← SP+4	
	RETI	-	2	6	–	PCL ← (SP), PCH ← (SP+1), PCS ← (SP+2), PSW← (SP+3), SP ← SP+4	R R R
	RETB	-	2	6	–	PCL ← (SP), PCH ← (SP+1), PCS ← (SP+2), PSW ← (SP+3), SP ← SP+4	R R R

Notes 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

<R> **2.** Number of CPU clocks (f_{CLK}) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Downloaded from Arrow.com.

Table 27-5. Operation List (16/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag
				Note 1	Note 2		Z AC CY
Stack manipulate	PUSH	PSW	2	1	–	(SP – 1) ← PSW, (SP – 2) ← 00H, SP ← SP–2	
		rp	1	1	–	(SP – 1) ← rp_H, (SP – 2) ← rp_L, SP ← SP – 2	
	POP	PSW	2	3	–	PSW ← (SP+1), SP ← SP + 2	R R R
		rp	1	1	–	rp_L ←(SP), rp_H ← (SP+1), SP ← SP + 2	
	MOVW	SP, #word	4	1	–	SP ← word	
		SP, AX	2	1	–	SP ← AX	
		AX, SP	2	1	–	AX ← SP	
		HL, SP	3	1	–	HL ← SP	
		BC, SP	3	1	–	BC ← SP	
		DE, SP	3	1	–	DE ← SP	
	ADDW	SP, #byte	2	1	–	SP ← SP + byte	
	SUBW	SP, #byte	2	1	–	SP ← SP – byte	
Unconditional branch	BR	AX	2	3	–	PC ← CS, AX	
		$addr20	2	3	–	PC ← PC + 2 + jdisp8	
		$!addr20	3	3	–	PC ← PC + 3 + jdisp16	
		!addr16	3	3	–	PC ← 0000, addr16	
		!!addr20	4	3	–	PC ← addr20	
Conditional branch	BC	$addr20	2	2/4 Note3	–	PC ← PC + 2 + jdisp8 if CY = 1	
	BNC	$addr20	2	2/4 Note3	–	PC ← PC + 2 + jdisp8 if CY = 0	
	BZ	$addr20	2	2/4 Note3	–	PC ← PC + 2 + jdisp8 if Z = 1	
	BNZ	$addr20	2	2/4 Note3	–	PC ← PC + 2 + jdisp8 if Z = 0	
	BH	$addr20	3	2/4 Note3	–	PC ← PC + 3 + jdisp8 if (Z∨CY)=0	
	BNH	$addr20	3	2/4 Note3	–	PC ← PC + 3 + jdisp8 if (Z∨CY)=1	
	BT	saddr.bit, $addr20	4	3/5 Note3	–	PC ← PC + 4 + jdisp8 if (saddr).bit = 1	
		sfr.bit, $addr20	4	3/5 Note3	–	PC ← PC + 4 + jdisp8 if sfr.bit = 1	
		A.bit, $addr20	3	3/5 Note3	–	PC ← PC + 3 + jdisp8 if A.bit = 1	
		PSW.bit, $addr20	4	3/5 Note3	–	PC ← PC + 4 + jdisp8 if PSW.bit = 1	
		[HL].bit, $addr20	3	3/5 Note3	6/7	PC ← PC + 3 + jdisp8 if (HL).bit = 1	
		ES:[HL].bit, $addr20	4	4/6 Note3	7/8	PC ← PC + 4 + jdisp8 if (ES, HL).bit = 1	

Notes 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

<R> **2.** Number of CPU clocks (f_{CLK}) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

3. This indicates the number of clocks “when condition is not met/when condition is met”.

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Downloaded from Arrow.com.

Table 27-5. Operation List (17/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag
				Note 1	Note 2		Z AC CY
Conditional branch	BF	saddr.bit, $addr20	4	3/5 Note3	–	PC ← PC + 4 + jdisp8 if (saddr).bit = 0	
		sfr.bit, $addr20	4	3/5 Note3	–	PC ← PC + 4 + jdisp8 if sfr.bit = 0	
		A.bit, $addr20	3	3/5 Note3	–	PC ← PC + 3 + jdisp8 if A.bit = 0	
		PSW.bit, $addr20	4	3/5 Note3	–	PC ← PC + 4 + jdisp8 if PSW.bit = 0	
		[HL].bit, $addr20	3	3/5 Note3	6/7	PC ← PC + 3 + jdisp8 if (HL).bit = 0	
		ES:[HL].bit, $addr20	4	4/6 Note3	7/8	PC ← PC + 4 + jdisp8 if (ES, HL).bit = 0	
	BTCLR	saddr.bit, $addr20	4	3/5 Note3	–	PC ← PC + 4 + jdisp8 if (saddr).bit = 1 then reset (saddr).bit	
		sfr.bit, $addr20	4	3/5 Note3	–	PC ← PC + 4 + jdisp8 if sfr.bit = 1 then reset sfr.bit	
		A.bit, $addr20	3	3/5 Note3	–	PC ← PC + 3 + jdisp8 if A.bit = 1 then reset A.bit	
		PSW.bit, $addr20	4	3/5 Note3	–	PC ← PC + 4 + jdisp8 if PSW.bit = 1 then reset PSW.bit	× × ×
		[HL].bit, $addr20	3	3/5 Note3	–	PC ← PC + 3 + jdisp8 if (HL).bit = 1 then reset (HL).bit	
		ES:[HL].bit, $addr20	4	4/6 Note3	–	PC ← PC + 4 + jdisp8 if (ES, HL).bit = 1 then reset (ES, HL).bit	
Conditional skip	SKC	–	2	1	–	Next instruction skip if CY = 1	
	SKNC	–	2	1	–	Next instruction skip if CY = 0	
	SKZ	–	2	1	–	Next instruction skip if Z = 1	
	SKNZ	–	2	1	–	Next instruction skip if Z = 0	
	SKH	–	2	1	–	Next instruction skip if (Z∨CY)=0	
	SKNH	–	2	1	–	Next instruction skip if (Z∨CY)=1	
CPU control	SEL Note4	RBn	2	1	–	RBS[1:0] ← n	
	NOP	–	1	1	–	No Operation	
	EI	–	3	4	–	IE ← 1 (Enable Interrupt)	
	DI	–	3	4	–	IE ← 0 (Disable Interrupt)	
	HALT	–	2	3	–	Set HALT Mode	
	STOP	–	2	3	–	Set STOP Mode	

Notes 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

<R> **2.** Number of CPU clocks (f_{CLK}) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

3. This indicates the number of clocks “when condition is not met/when condition is met”.

4. n indicates the number of register banks (n = 0 to 3).

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Downloaded from Arrow.com.

CHAPTER 28 ELECTRICAL SPECIFICATIONS (A, D: T_A = –40 to +85°C)

This chapter describes the electrical specifications for the products "A: Consumer applications (T_A = -40 to +85°C)" and "D: Industrial applications (T_A = -40 to +85°C)".

Cautions 1. The RL78 microcontrollers have an on-chip debug function, which is provided for development and evaluation. Do not use the on-chip debug function in products designated for mass production, because the guaranteed number of rewritable times of the flash memory may be exceeded when this function is used, and product reliability therefore cannot be guaranteed. Renesas Electronics is not liable for problems occurring when the on-chip debug function is used.

2. The pins mounted depend on the product. Refer to 2.1 Port Functions to 2.2.1 Functions for each product.

Downloaded from Arrow.com.

28.1 Absolute Maximum Ratings

Absolute Maximum Ratings (T_A = 25°C)

<R>

Parameter	Symbols	Conditions		Ratings	Unit
Supply Voltage	V_{DD}			–0.5 to + 6.5	V
REGC terminal input voltage[Note1]	V_{IREGC}	REGC		–0.3 to +2.8 and –0.3 to V_{DD} + 0.3 [Note 2]	V
Input Voltage	V_{I1}	Other than P60, P61		–0.3 to V_{DD} + 0.3[Note 3]	V
	V_{I2}	P60, P61 (N-ch open drain)		–0.3 to 6.5	V
Output Voltage	V_O			–0.3 to V_{DD} + 0.3[Note 3]	V
Analog input voltage	V_{AI}	20-, 24-pin products: ANI0 to ANI3, ANI16 to ANI22 30-pin products: ANI0 to ANI3, ANI16 to ANI19		–0.3 to V_{DD} + 0.3 and –0.3 to AVREF(+)+0.3 [Notes 3, 4]	V
Output current, high	I_{OH1}	Per pin	Other than P20 to P23	–40	mA
		Total of all pins	All the terminals other than P20 to P23	–170	mA
			20-, 24-pin products: P40 to P42 30-pin products: P00, P01, P40, P120	–70	mA
			20-, 24-pin products: P00 to P03[Note 5], P10 to P14 30-pin products: P10 to P17, P30, P31, P50, P51, P147	–100	mA
	I_{OH2}	Per pin	P20 to P23	–0.5	mA
		Total of all pins		–2	mA
Output current, low	I_{OL1}	Per pin	Other than P20 to P23	40	mA
		Total of all pins	All the terminals other than P20 to P23	170	mA
			20-, 24-pin products: P40 to P42 30-pin products: P00, P01, P40, P120	70	mA
			20-, 24-pin products: P00 to P03[Note 5], P10 to P14, P60, P61 30-pin products: P10 to P17, P30, P31, P50, P51, P60, P61, P147	100	mA
	I_{OL2}	Per pin	P20 to P23	1	mA
		Total of all pins		5	mA
Operating ambient temperature	T_A			–40 to +85	°C
Storage temperature	T_{stg}			–65 to +150	°C

Notes 1. 30-pin product only.

2. Connect the REGC pin to V_{SS} via a capacitor (0.47 to 1 µF). This value determines the absolute maximum rating of the REGC pin. Do not use it with voltage applied.

3. Must be 6.5 V or lower.

4. Do not exceed $AV_{REF}(+)$ + 0.3 V in case of A/D conversion target pin.

5. 24-pin products only.

Caution Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.

Remarks 1. Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

2. $AV_{REF}(+)$: + side reference voltage of the A/D converter.

3. V_{SS} : Reference voltage

Downloaded from Arrow.com.

28.2 Oscillator Characteristics

28.2.1 X1 oscillator characteristics

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

<R>

Parameter	Resonator	Conditions	MIN.	TYP.	MAX.	Unit
X1 clock oscillation frequency (f_X)Note	Ceramic resonator / crystal oscillator	2.7 V ≤ V_{DD} ≤ 5.5 V	1.0		20.0	MHz
		1.8 V ≤ V_{DD} < 2.7 V	1.0		8.0	

<R> **Note** Indicates only permissible oscillator frequency ranges. Refer to AC Characteristics for instruction execution time. Request evaluation by the manufacturer of the oscillator circuit mounted on a board to check the oscillator characteristics.

<R> **Caution Since the CPU is started by the high-speed on-chip oscillator clock after a reset release, check the X1 clock oscillation stabilization time using the oscillation stabilization time counter status register (OSTC) by the user. Determine the oscillation stabilization time of the OSTC register and the oscillation stabilization time select register (OSTS) after sufficiently evaluating the oscillation stabilization time with the resonator to be used.**

Remark When using the X1 oscillator, refer to **5.4 System Clock Oscillator**.

28.2.2 On-chip oscillator characteristics

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

<R>

Oscillators	Parameters	Conditions		MIN.	TYP.	MAX.	Unit
High-speed on-chip oscillator clock frequency Notes 1, 2	f_{IH}			1		24	MHz
High-speed on-chip oscillator clock frequency accuracy		R5F102 products	T_A = –20 to +85°C	-1.0		+1.0	%
			T_A = –40 to –20°C	-1.5		+1.5	%
		R5F103 products		-5.0		+5.0	%
Low-speed on-chip oscillator clock frequency	f_{IL}				15		kHz
Low-speed on-chip oscillator clock frequency accuracy				-15		+15	%

Notes 1. High-speed on-chip oscillator frequency is selected by bits 0 to 3 of option byte (000C2H) and bits 0 to 2 of HOCODIV register.

2. This only indicates the oscillator characteristics. Refer to AC Characteristics for instruction execution time.

Downloaded from Arrow.com.

28.3 DC Characteristics

28.3.1 Pin characteristics

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V) (1/4)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output current, high[Note 1]	I_{OH1}	20-, 24-pin products: Per pin for P00 to P03[Note 4], P10 to P14, P40 to P42 30-pin products: Per pin for P00, P01, P10 to P17, P30, P31, P40, P50, P51, P120, P147				–10.0 [Note 2]	mA
		20-, 24-pin products: Total of P40 to P42 30-pin products: Total of P00, P01, P40, P120 (When duty ≤ 70% [Note 3])	4.0 V ≤ V_{DD} ≤ 5.5 V			–30.0	mA
			2.7 V ≤ V_{DD} < 4.0 V			–6.0	mA
			1.8 V ≤ V_{DD} < 2.7 V			–4.5	mA
		20-, 24-pin products: Total of P00 to P03[Note 4], P10 to P14 30-pin products: Total of P10 to P17, P30, P31, P50, P51, P147 (When duty ≤ 70% [Note 3])	4.0 V ≤ V_{DD} ≤ 5.5 V			–80.0	mA
			2.7 V ≤ V_{DD} < 4.0 V			–18.0	mA
			1.8 V ≤ V_{DD} < 2.7 V			–10.0	mA
		Total of all pins (When duty ≤ 70%[Note 3])				–100	mA
	I_{OH2}	Per pin for P20 to P23				–0.1	mA
		Total of all pins				–0.4	mA

Notes 1. value of current at which the device operation is guaranteed even if the current flows from the V_{DD} pin to an output pin.

2. However, do not exceed the total current value.

<R> **3.** The output current value under conditions where the duty factor ≤ 70%.
If duty factor > 70%: The output current value can be calculated with the following expression (where n represents the duty factor as a percentage).

- Total output current of pins = (I_{OH} × 0.7)/(n × 0.01)

<Example> Where n = 80% and I_{OH} = –10.0 mA

Total output current of pins = (–10.0 × 0.7)/(80 × 0.01) ≅ –8.7 mA

However, the current that is allowed to flow into one pin does not vary depending on the duty factor. A current higher than the absolute maximum rating must not flow into one pin.

4. 24-pin products only.

Caution P10 to P12 and P41 for 20-pin products, P01, P10 to P12, and P41 for 24-pin products, and P00, P10 to P15, P17, and P50 for 30-pin products do not output high level in N-ch open-drain mode.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

Downloaded from Arrow.com.

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V) (2/4)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output current, low Note 1	I_{OL1}	20-, 24-pin products: Per pin for P00 to P03 Note 4, P10 to P14, P40 to P42 30-pin products: Per pin for P00, P01, P10 to P17, P30, P31, P40, P50, P51, P120, P147				20.0 Note 2	mA
		Per pin for P60, P61				15.0 Note 2	mA
		20-, 24-pin products: Total of P40 to P42 30-pin products: Total of P00, P01, P40, P120 (When duty ≤ 70% Note 3)	4.0 V ≤ V_{DD} ≤ 5.5 V			60.0	mA
			2.7 V ≤ V_{DD} < 4.0 V			9.0	mA
			1.8 V ≤ V_{DD} < 2.7 V			1.8	mA
		20-, 24-pin products: Total of P00 to P03 Note 4, P10 to P14, P60, P61 30-pin products: Total of P10 to P17, P30, P31, P50, P51, P60, P61, P147 (When duty ≤ 70% Note 3)	4.0 V ≤ V_{DD} ≤ 5.5 V			80.0	mA
			2.7 V ≤ V_{DD} < 4.0 V			27.0	mA
			1.8 V ≤ V_{DD} < 2.7 V			5.4	mA
		Total of all pins (When duty ≤ 70% Note 3)				140	mA
	I_{OL2}	Per pin for P20 to P23				0.4	mA
		Total of all pins				1.6	mA

Notes 1. Value of current at which the device operation is guaranteed even if the current flows from an output pin to the V_{SS} pin.

2. However, do not exceed the total current value.

<R> **3.** The output current value under conditions where the duty factor ≤ 70%.

If duty factor > 70%: The output current value can be calculated with the following expression (where n represents the duty factor as a percentage).

- Total output current of pins = (I_{OL} × 0.7)/(n × 0.01)

<Example> Where n = 80% and I_{OL} = 10.0 mA

Total output current of pins = (10.0 × 0.7)/(80 × 0.01) ≅ 8.7 mA

However, the current that is allowed to flow into one pin does not vary depending on the duty factor. A current higher than the absolute maximum rating must not flow into one pin.

4. 24-pin products only.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

Downloaded from Arrow.com.

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V) (3/4)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input voltage, high	V_{IH1}	Normal input buffer 20-, 24-pin products: P00 to P03[Note 2], P10 to P14, P40 to P42 30-pin products: P00, P01, P10 to P17, P30, P31, P40, P50, P51, P120, P147		0.8V_{DD}		V_{DD}	V
	V_{IH2}	TTL input buffer 20-, 24-pin products: P10, P11 30-pin products: P01, P10, P11, P13 to P17	4.0 V ≤ V_{DD} ≤ 5.5 V	2.2		V_{DD}	V
			3.3 V ≤ V_{DD} < 4.0 V	2.0		V_{DD}	V
			1.8 V ≤ V_{DD} < 3.3 V	1.5		V_{DD}	V
	V_{IH3}	P20 to P23		0.7V_{DD}		V_{DD}	V
	V_{IH4}	P60, P61		0.7V_{DD}		6.0	V
	V_{IH5}	P121, P122, P125[Note 1], P137, EXCLK, $\overline{\text{RESET}}$		0.8V_{DD}		V_{DD}	V
Input voltage, low	V_{IL1}	Normal input buffer 20-, 24-pin products: P00 to P03[Note 2], P10 to P14, P40 to P42 30-pin products: P00, P01, P10 to P17, P30, P31, P40, P50, P51, P120, P147		0		0.2V_{DD}	V
	V_{IL2}	TTL input buffer 20-, 24-pin products: P10, P11 30-pin products: P01, P10, P11, P13 to P17	4.0 V ≤ V_{DD} ≤ 5.5 V	0		0.8	V
			3.3 V ≤ V_{DD} < 4.0 V	0		0.5	V
			1.8 V ≤ V_{DD} < 3.3 V	0		0.32	V
	V_{IL3}	P20 to P23		0		0.3V_{DD}	V
	V_{IL4}	P60, P61		0		0.3V_{DD}	V
	V_{IL5}	P121, P122, P125[Note 1], P137, EXCLK, $\overline{\text{RESET}}$		0		0.2V_{DD}	V
Output voltage, high	V_{OH1}	20-, 24-pin products: P00 to P03[Note 2], P10 to P14, P40 to P42 30-pin products: P00, P01, P10 to P17, P30, P31, P40, P50, P51, P120, P147	4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OH1} = –10.0 mA	V_{DD}–1.5			V
			4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OH1} = –3.0 mA	V_{DD}–0.7			V
			2.7 V ≤ V_{DD} ≤ 5.5 V, I_{OH1} = –2.0 mA	V_{DD}–0.6			V
			1.8 V ≤ V_{DD} ≤ 5.5 V, I_{OH1} = –1.5 mA	V_{DD}–0.5			V
	V_{OH2}	P20 to P23	I_{OH2} = –100 μA	V_{DD}–0.5			V

Notes 1. 20, 24-pin products only.

2. 24-pin products only.

Caution The maximum value of V_{IH} of pins P10 to P12 and P41 for 20-pin products, P01, P10 to P12, and P41 for 24-pin products, and P00, P10 to P15, P17, and P50 for 30-pin products is V_{DD} even in N-ch open-drain mode.
High level is not output in the N-ch open-drain mode.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

Downloaded from Arrow.com.

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V) (4/4)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Output voltage, low	V_{OL1}	20-, 24-pin products: P00 to P03[Note], P10 to P14, P40 to P42 30-pin products: P00, P01, P10 to P17, P30, P31, P40, P50, P51, P120, P147		4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 20.0 mA			1.3	V
				4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 8.5 mA			0.7	V
				2.7 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 3.0 mA			0.6	V
				2.7 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 1.5 mA			0.4	V
				1.8 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 0.6 mA			0.4	V
	V_{OL2}	P20 to P23		I_{OL2} = 400 μA			0.4	V
	V_{OL3}	P60, P61		4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 15.0 mA			2.0	V
				4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 5.0 mA			0.4	V
				2.7 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 3.0 mA			0.4	V
				1.8 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 2.0 mA			0.4	V
Input leakage current, high	I_{LIH1}	Other than P121, P122	V_I = V_{DD}				1	μA
	I_{LIH2}	P121, P122 (X1, X2/EXCLK)	V_I = V_{DD}	Input port or external clock input			1	μA
				When resonator connected			10	μA
Input leakage current, low	I_{LIL1}	Other than P121, P122	V_I = V_{SS}				–1	μA
	I_{LIL2}	P121, P122 (X1, X2/EXCLK)	V_I = V_{SS}	Input port or external clock input			–1	μA
				When resonator connected			–10	μA
On-chip pull-up resistance	R_U	20-, 24-pin products: P00 to P03[Note], P10 to P14, P40 to P42, P125, $\overline{\text{RESET}}$ 30-pin products: P00, P01, P10 to P17, P30, P31, P40, P50, P51, P120, P147		V_I = V_{SS}, input port	10	20	100	kΩ

Note 24-pin products only.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

Downloaded from Arrow.com.

28.3.2 Supply current characteristics

(1) 20-, 24-pin products

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V) **(1/2)**

Parameter	Symbol	Conditions					MIN.	TYP.	MAX.	Unit
Supply current[Note 1]	I_{DD1}	Operating mode	HS(High-speed main) mode[Note 4]	f_{IH} = 24 MHz[Note 3]	Basic operation	V_{DD} = 5.0 V		1.5		mA
						V_{DD} = 3.0 V		1.5		
					Normal operation	V_{DD} = 5.0 V		3.3	5.0	mA
						V_{DD} = 3.0 V		3.3	5.0	
				f_{IH} = 16 MHz[Note 3]		V_{DD} = 5.0 V		2.5	3.7	mA
						V_{DD} = 3.0 V		2.5	3.7	
			LS(Low-speed main) mode[Note 4]	f_{IH} = 8 MHz[Note 3]		V_{DD} = 3.0 V		1.2	1.8	mA
						V_{DD} = 2.0 V		1.2	1.8	
			HS(High-speed main) mode[Note4]	f_{MX} = 20 MHz[Note 2], V_{DD} = 5.0 V		Square wave input		2.8	4.4	mA
						Resonator connection		3.0	4.6	
				f_{MX} = 20 MHz[Note 2], V_{DD} = 3.0 V		Square wave input		2.8	4.4	mA
						Resonator connection		3.0	4.6	
				f_{MX} = 10 MHz[Note 2], V_{DD} = 5.0 V		Square wave input		1.8	2.6	mA
						Resonator connection		1.8	2.6	
				f_{MX} = 10 MHz[Note 2], V_{DD} = 3.0 V		Square wave input		1.8	2.6	mA
						Resonator connection		1.8	2.6	
			LS(Low-speed main) mode[Note 4]	f_{MX} = 8 MHz[Note 2], V_{DD} = 3.0 V		Square wave input		1.1	1.7	mA
						Resonator connection		1.1	1.7	
				f_{MX} = 8 MHz[Note 2], V_{DD} = 2.0 V		Square wave input		1.1	1.7	mA
						Resonator connection		1.1	1.7	

<R> **Notes 1.** Total current flowing into V_{DD}, including the input leakage current flowing when the level of the input pin is fixed to V_{DD} or V_{SS}. The values below the MAX. column include the peripheral operation current. However, not including the current flowing into the A/D converter, LVD circuit, I/O port, and on-chip pull-up/pull-down resistors and the current flowing during data flash rewrite.

<R> **2.** When high-speed on-chip oscillator clock is stopped.

3. When high-speed system clock is stopped

4. Relationship between operation voltage width, operation frequency of CPU and operation mode is as follows.

HS(High speed main) mode: V_{DD} = 2.7 V to 5.5 V @1 MHz to 24 MHz
V_{DD} = 2.4 V to 5.5 V @1 MHz to 16 MHz

LS(Low speed main) mode: V_{DD} = 1.8 V to 5.5 V @1 MHz to 8 MHz

Remarks 1. f_{MX}: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)

2. f_{IH}: high-speed on-chip oscillator clock frequency

3. Temperature condition of the TYP. value is T_A = 25°C.

Downloaded from Arrow.com.

(1) 20-, 24-pin products

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V) **(2/2)**

Parameter	Symbol	Conditions				MIN.	TYP.	MAX.	Unit
Supply current Note 1	I_{DD2} Note 2	HALT mode	HS (High-speed main) mode Note 6	f_{IH} = 24 MHz Note 4	V_{DD} = 5.0 V		440	1210	μA
					V_{DD} = 3.0 V		440	1210	
				f_{IH} = 16 MHz Note 4	V_{DD} = 5.0 V		400	950	μA
					V_{DD} = 3.0 V		400	950	
			LS (Low-speed main) mode Note 6	f_{IH} = 8 MHz Note 4	V_{DD} = 3.0 V		270	542	μA
					V_{DD} = 2.0 V		270	542	
			HS (High-speed main) mode Note 6	f_{MX} = 20 MHz Note 3, V_{DD} = 5.0 V	Square wave input		280	1000	μA
					Resonator connection		450	1170	
				f_{MX} = 20 MHz Note 3, V_{DD} = 3.0 V	Square wave input		280	1000	μA
					Resonator connection		450	1170	
				f_{MX} = 10 MHz Note 3, V_{DD} = 5.0 V	Square wave input		190	590	μA
					Resonator connection		260	660	
				f_{MX} = 10 MHz Note 3, V_{DD} = 3.0 V	Square wave input		190	590	μA
					Resonator connection		260	660	
			LS (Low-speed main) mode Note 6	f_{MX} = 8 MHz Note 3, V_{DD} = 3.0 V	Square wave input		110	360	μA
					Resonator connection		150	416	
				f_{MX} = 8 MHz Note 3, V_{DD} = 2.0 V	Square wave input		110	360	μA
					Resonator connection		150	416	
	I_{DD3} Note 5	STOP mode	T_A = –40°C				0.19	0.50	μA
			T_A = +25°C				0.24	0.50	
			T_A = +50°C				0.32	0.80	
			T_A = +70°C				0.48	1.20	
			T_A = +85°C				0.74	2.20	

<R> **Notes 1.** Total current flowing into V_{DD}, including the input leakage current flowing when the level of the input pin is fixed to V_{DD} or V_{SS}. The values below the MAX. column include the peripheral operation current. However, not including the current flowing into the A/D converter, LVD circuit, I/O port, and on-chip pull-up/pull-down resistors and the current flowing during data flash rewrite.

2. During HALT instruction execution by flash memory.

<R> **3.** When high-speed on-chip oscillator clock is stopped.

4. When high-speed system clock is stopped.

5. Not including the current flowing into the 12-bit interval timer and watchdog timer.

6. Relationship between operation voltage width, operation frequency of CPU and operation mode is as follows.

HS(High speed main) mode: V_{DD} = 2.7 V to 5.5 V @1 MHz to 24 MHz
V_{DD} = 2.4 V to 5.5 V @1 MHz to 16 MHz

LS(Low speed main) mode: V_{DD} = 1.8 V to 5.5 V @1 MHz to 8 MHz

Remarks 1. f_{MX}: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)

2. f_{IH}: high-speed on-chip oscillator clock frequency

3. Except temperature condition of the TYP. value is T_A = 25°C, other than STOP mode

Downloaded from Arrow.com.

(2) 30-pin products

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V) **(1/2)**

Parameter	Symbol	Conditions					MIN.	TYP.	MAX.	Unit
Supply current Note 1	I_{DD1}	Operating mode	HS (High-speed main) mode Note 4	f_{IH} = 24 MHz Note 3	Basic operation	V_{DD} = 5.0 V		1.5		mA
						V_{DD} = 3.0 V		1.5		
					Normal operation	V_{DD} = 5.0 V		3.7	5.5	mA
						V_{DD} = 3.0 V		3.7	5.5	
				f_{IH} = 16 MHz Note 3		V_{DD} = 5.0 V		2.7	4.0	mA
						V_{DD} = 3.0 V		2.7	4.0	
			LS (Low-speed main) mode Note 4	f_{IH} = 8 MHz Note 3		V_{DD} = 3.0 V		1.2	1.8	mA
						V_{DD} = 2.0 V		1.2	1.8	
			HS (High-speed main) mode Note 4	f_{MX} = 20 MHz Note 2, V_{DD} = 5.0 V		Square wave input		3.0	4.6	mA
						Resonator connection		3.2	4.8	
				f_{MX} = 20 MHz Note 2, V_{DD} = 3.0 V		Square wave input		3.0	4.6	mA
						Resonator connection		3.2	4.8	
				f_{MX} = 10 MHz Note 2, V_{DD} = 5.0 V		Square wave input		1.9	2.7	mA
						Resonator connection		1.9	2.7	
				f_{MX} = 10 MHz Note 2, V_{DD} = 3.0 V		Square wave input		1.9	2.7	mA
						Resonator connection		1.9	2.7	
			LS (Low-speed main) mode Note 4	f_{MX} = 8 MHz Note 2, V_{DD} = 3.0 V		Square wave input		1.1	1.7	mA
						Resonator connection		1.1	1.7	
				f_{MX} = 8 MHz Note 2, V_{DD} = 2.0 V		Square wave input		1.1	1.7	mA
						Resonator connection		1.1	1.7	

<R> **Notes 1.** Total current flowing into V_{DD}, including the input leakage current flowing when the level of the input pin is fixed to V_{DD} or V_{SS}. The values below the MAX. column include the peripheral operation current. However, not including the current flowing into the A/D converter, LVD circuit, I/O port, and on-chip pull-up/pull-down resistors and the current flowing during data flash rewrite.

<R> **2.** When high-speed on-chip oscillator clock is stopped.

3. When high-speed system clock is stopped

4. Relationship between operation voltage width, operation frequency of CPU and operation mode is as follows.

HS(High speed main) mode: V_{DD} = 2.7 V to 5.5 V @1 MHz to 24 MHz
V_{DD} = 2.4 V to 5.5 V @1 MHz to 16 MHz

LS(Low speed main) mode: V_{DD} = 1.8 V to 5.5 V @1 MHz to 8 MHz

Remarks 1. f_{MX}: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)

2. f_{IH}: high-speed on-chip oscillator clock frequency

3. Temperature condition of the TYP. value is T_A = 25°C.

Downloaded from Arrow.com.

(2) 30-pin products

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V) **(2/2)**

Parameter	Symbol	Conditions				MIN.	TYP.	MAX.	Unit
Supply current Note 1	I_{DD2} Note 2	HALT mode	HS (High-speed main) mode Note 6	f_{IH} = 24 MHz Note 4	V_{DD} = 5.0 V		440	1280	μA
					V_{DD} = 3.0 V		440	1280	
				f_{IH} = 16 MHz Note 4	V_{DD} = 5.0 V		400	1000	μA
					V_{DD} = 3.0 V		400	1000	
			LS (Low-speed main) mode Note 6	f_{IH} = 8 MHz Note 4	V_{DD} = 3.0 V		260	530	μA
					V_{DD} = 2.0 V		260	530	
			HS (High-speed main) mode Note 6	f_{MX} = 20 MHz Note 3, V_{DD} = 5.0 V	Square wave input		280	1000	μA
					Resonator connection		450	1170	
				f_{MX} = 20 MHz Note 3, V_{DD} = 3.0 V	Square wave input		280	1000	μA
					Resonator connection		450	1170	
				f_{MX} = 10 MHz Note 3, V_{DD} = 5.0 V	Square wave input		190	600	μA
					Resonator connection		260	670	
				f_{MX} = 10 MHz Note 3, V_{DD} = 3.0 V	Square wave input		190	600	μA
					Resonator connection		260	670	
			LS (Low-speed main) mode Note 6	f_{MX} = 8 MHz Note 3, V_{DD} = 3.0 V	Square wave input		95	330	μA
					Resonator connection		145	380	
				f_{MX} = 8 MHz Note 3 V_{DD} = 2.0 V	Square wave input		95	330	μA
					Resonator connection		145	380	
	I_{DD3} Note 5	STOP mode	T_A = –40°C				0.18	0.50	μA
			T_A = +25°C				0.23	0.50	
			T_A = +50°C				0.30	1.10	
			T_A = +70°C				0.46	1.90	
			T_A = +85°C				0.75	3.30	

<R> **Notes 1.** Total current flowing into V_{DD}, including the input leakage current flowing when the level of the input pin is fixed to V_{DD} or V_{SS}. The values below the MAX. column include the peripheral operation current. However, not including the current flowing into the A/D converter, LVD circuit, I/O port, and on-chip pull-up/pull-down resistors and the current flowing during data flash rewrite.

2. During HALT instruction execution by flash memory.

<R> **3.** When high-speed on-chip oscillator clock is stopped.

4. When high-speed system clock is stopped.

5. Not including the current flowing into the 12-bit interval timer and watchdog timer.

6. Relationship between operation voltage width, operation frequency of CPU and operation mode is as follows.

HS (High speed main) mode: V_{DD} = 2.7 V to 5.5 V @1 MHz to 24 MHz
V_{DD} = 2.4 V to 5.5 V @1 MHz to 16 MHz

LS (Low speed main) mode: V_{DD} = 1.8 V to 5.5 V @1 MHz to 8 MHz

Remarks 1. f_{MX}: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)

2. f_{IH}: high-speed on-chip oscillator clock frequency

3. Except STOP mode, temperature condition of the TYP. value is T_A = 25°C.

Downloaded from Arrow.com.

<R> **(3) Peripheral functions (Common to all products)**

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Low-speed onchip oscillator operating current	I_{FIL} Note 1				0.20		μA
12-bit interval timer operating current	I_{TMKA} Notes 1, 2, 3				0.02		μA
Watchdog timer operating current	I_{WDT} Notes 1, 2, 4	f_{IL} = 15 kHz			0.22		μA
A/D converter operating current	I_{ADC} Notes 1, 5	When conversion at maximum speed	Normal mode, AV_{REFP} = V_{DD} = 5.0 V		1.30	1.70	mA
			Low voltage mode, AV_{REFP} = V_{DD} = 3.0 V		0.50	0.70	mA
A/D converter reference voltage operating current	I_{ADREF} Note 1				75.0		μA
Temperature sensor operating current	I_{TMPS} Note 1				75.0		μA
LVD operating current	I_{LVD} Notes 1, 6				0.08		μA
Self-programming operating current	I_{FSP} Notes 1, 8				2.00	12.20	mA
BGO operating current	I_{BGO} Notes 1, 7				2.00	12.20	mA
SNOOZE operating current	I_{SNOZ} Note 1	ADC operation	The mode is performed Note 9		0.50	0.60	mA
			The A/D conversion operations are performed, Low voltage mode, AV_{REFP} = V_{DD} = 3.0 V		1.20	1.44	mA
		CSI/UART operation			0.70	0.84	mA

Notes 1. Current flowing to the V_{DD}.

2. When high speed on-chip oscillator and high-speed system clock are stopped.

3. Current flowing only to the 12-bit interval timer (excluding the operating current of the low-speed on-chip oscillator). The current value of the RL78 microcontrollers is the sum of I_{DD1}, I_{DD2} or I_{DD3}, and I_{FIL} and I_{TMKA} when the 12-bit interval timer operates.

4. Current flowing only to the watchdog timer (including the operating current of the low-speed on-chip oscillator). The current value of the RL78 microcontrollers is the sum of I_{DD1}, I_{DD2} or I_{DD3} and I_{WDT} when the watchdog timer operates.

5. Current flowing only to the A/D converter. The current value of the RL78 microcontrollers is the sum of I_{DD1} or I_{DD2} and I_{ADC} when the A/D converter operates in an operation mode or the HALT mode.

6. Current flowing only to the LVD circuit. The current value of the RL78 microcontrollers is the sum of I_{DD1}, I_{DD2} or I_{DD3} and I_{LVD} when the LVD circuit operates.

7. Current flowing only during data flash rewrite.

8. Current flowing only during self programming.

9. For shift time to the SNOOZE mode, see **17.3.3 SNOOZE mode**.

Remarks 1. f_{IL}: Low-speed on-chip oscillator clock frequency

2. Temperature condition of the TYP. value is T_A = 25°C

Downloaded from Arrow.com.

28.4 AC Characteristics

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

<R>

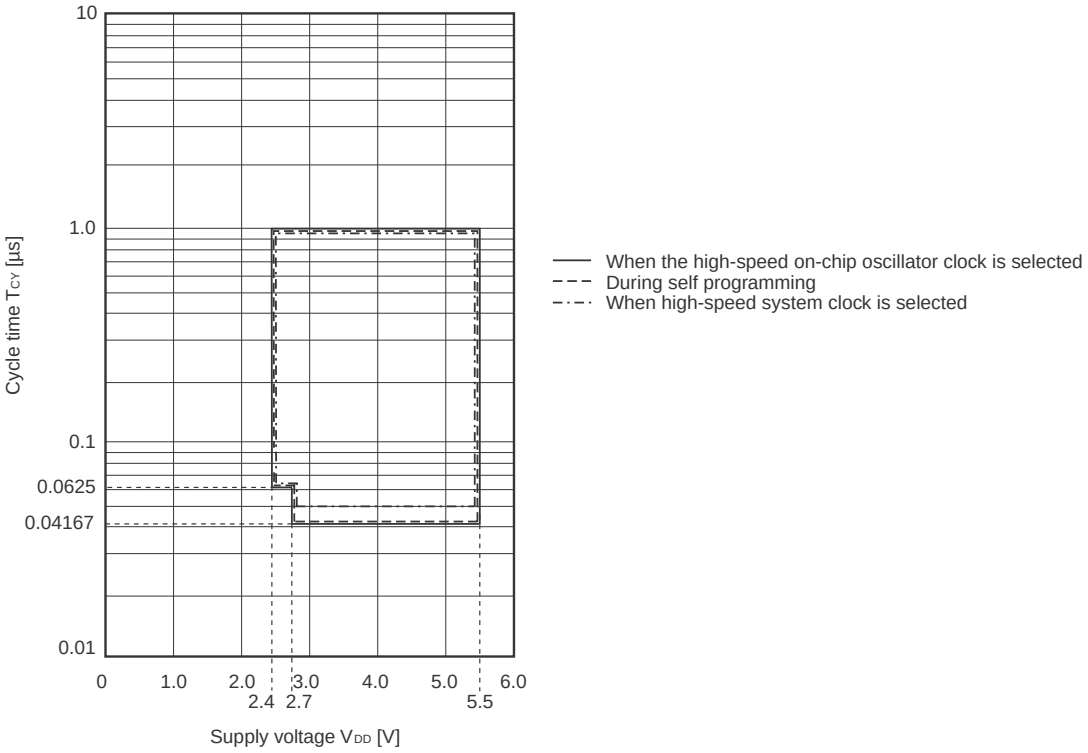
Items	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Instruction cycle (minimum instruction execution time)	T_{CY}	Main system clock (f_{MAIN}) operation	HS (High-speed main) mode	2.7 V ≤ V_{DD} ≤ 5.5 V	0.04167		1	μs
				2.4 V ≤ V_{DD} < 2.7 V	0.0625		1	μs
			LS (Low-speed main) mode	1.8 V ≤ V_{DD} ≤ 5.5 V	0.125		1	μs
		During self programming	HS (High-speed main) mode	2.7 V ≤ V_{DD} ≤ 5.5 V	0.04167		1	μs
				2.4 V ≤ V_{DD} < 2.7 V	0.0625		1	μs
			LS (Low-speed main) mode	1.8 V ≤ V_{DD} ≤ 5.5 V	0.125		1	μs
External main system clock frequency	f_{EX}	2.7 V ≤ V_{DD} ≤ 5.5 V			1.0		20.0	MHz
		2.4 V ≤ V_{DD} < 2.7 V			1.0		16.0	MHz
		1.8 V ≤ V_{DD} < 2.4 V			1.0		8.0	MHz
External main system clock input high-level width, low-level width	t_{EXH}, t_{EXL}	2.7 V ≤ V_{DD} ≤ 5.5 V			24			ns
		2.4 V ≤ V_{DD} < 2.7 V			30			ns
		1.8 V ≤ V_{DD} < 2.4 V			60			ns
TI00 to TI07 input high-level width, low-level width	t_{TIH}, t_{TIL}				1/f_{MCK} + 10			ns
TO00 to TO07 output frequency	f_{TO}	4.0 V ≤ V_{DD} ≤ 5.5 V					12	MHz
		2.7 V ≤ V_{DD} < 4.0 V					8	MHz
		1.8 V ≤ V_{DD} < 2.7 V					4	MHz
PCLBUZ0, or PCLBUZ1 output frequency	f_{PCL}	4.0 V ≤ V_{DD} ≤ 5.5 V					16	MHz
		2.7 V ≤ V_{DD} < 4.0 V					8	MHz
		1.8 V ≤ V_{DD} < 2.7 V					4	MHz
INTP0 to INTP5 input high-level width, low-level width	t_{INTH}, t_{INTL}				1			μs
KR0 to KR9 input available width	t_{KR}				250			ns
$\overline{\text{RESET}}$ low-level width	t_{RSL}				10			μs

Remark f_{MCK}: Timer array unit operation clock frequency
(Operation clock to be set by the timer clock select register 0 (TPS0) and the CKS0n bit of timer mode register 0n (TMR0n). n: Channel number (n = 0 to 7))

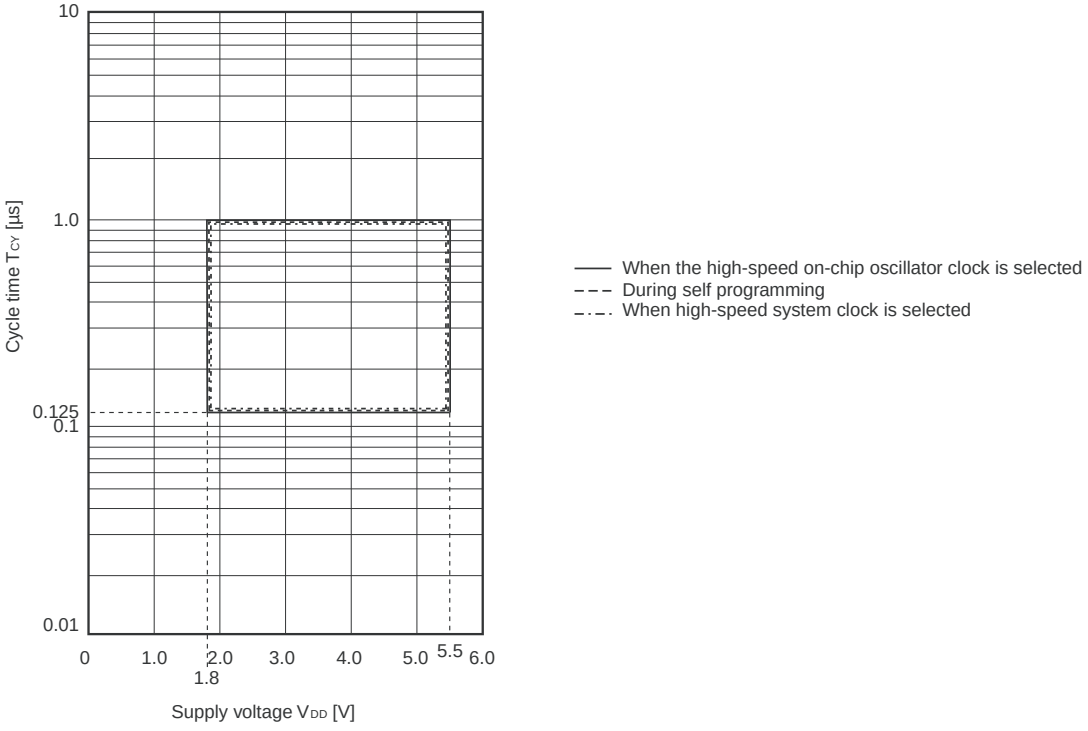
Downloaded from Arrow.com.

<R> **Minimum Instruction Execution Time during Main System Clock Operation**

T_{CY} vs V_{DD} (HS (high-speed main) mode)

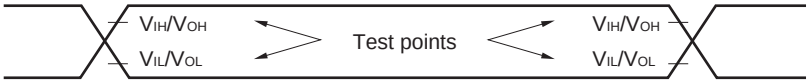


T_{CY} vs V_{DD} (LS (low-speed main) mode)

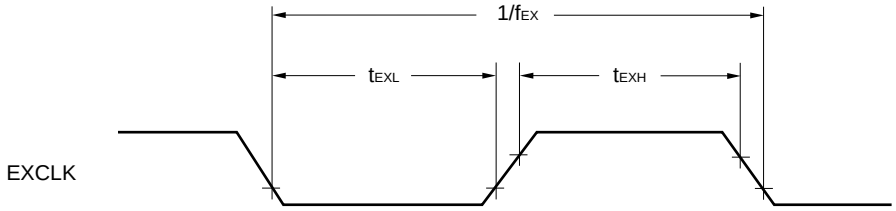


Downloaded from Arrow.com.

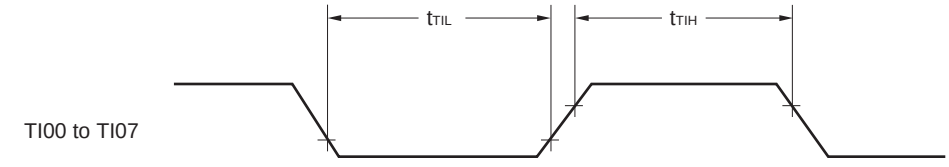
<R> **AC Timing Test Point**

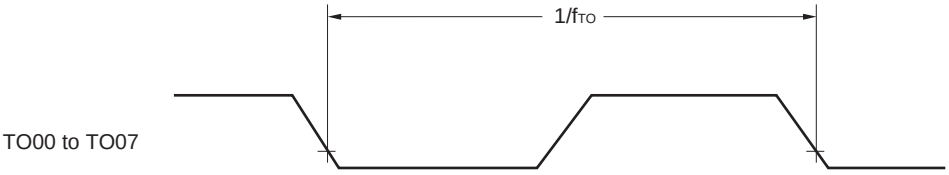


<R> **External Main System Clock Timing**

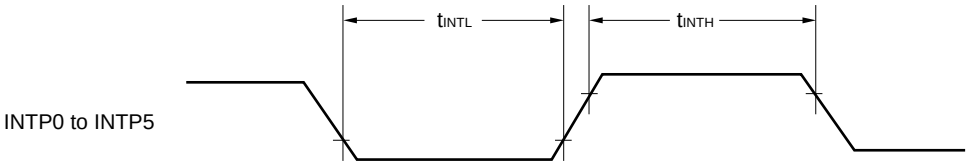


TI/TO Timing

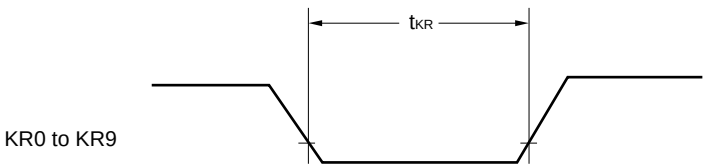




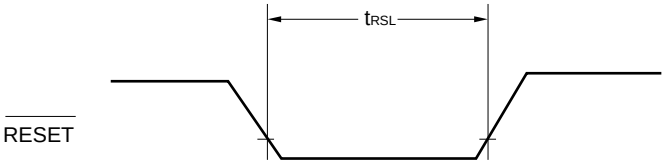
Interrupt Request Input Timing



Key Interrupt Input Timing



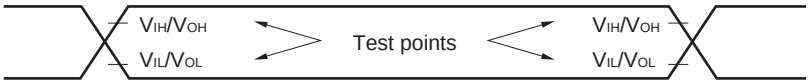
$\overline{\text{RESET}}$ Input Timing



Downloaded from Arrow.com.

28.5 Peripheral Functions Characteristics

<R> **AC Timing Test Point**



28.5.1 Serial array unit

<R> **(1) During communication at same potential (UART mode)**

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	HS (high-speed main) Mode		LS (low-speed main) Mode		Unit
			MIN.	MAX.	MIN.	MAX.	
Transfer rate Note 1				f_{MCK}/6		f_{MCK}/6	bps
		Theoretical value of the maximum transfer rate $f_{CLK} = f_{MCK}$ Note2		4.0		1.3	Mbps

Notes 1. Transfer rate in the SNOOZE mode is 4800 bps only.

<R> **2.** The maximum operating frequencies of the CPU/peripheral hardware clock (f_{CLK}) are:

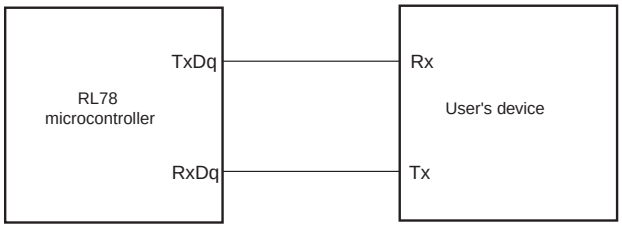
HS (high-speed main) mode: 24 MHz (2.7 V ≤ V_{DD} ≤ 5.5 V)

16 MHz (2.4 V ≤ V_{DD} ≤ 5.5 V)

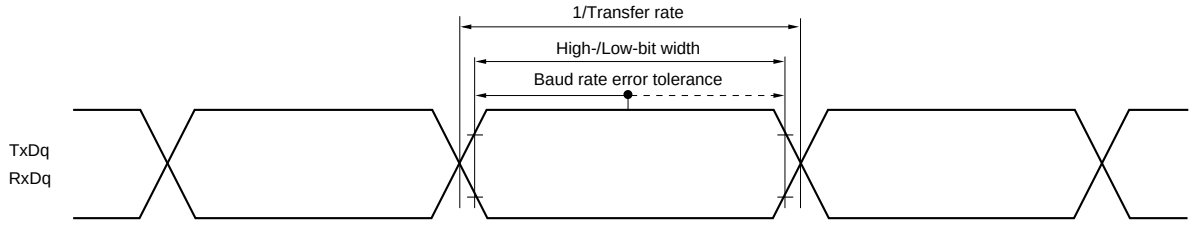
LS (low-speed main) mode: 8 MHz (1.8 V ≤ V_{DD} ≤ 5.5 V)

Caution Select the normal input buffer for the RxDq pin and the normal output mode for the TxDq pin by using port input mode register g (PIMg) and port output mode register g (POMg).

UART mode connection diagram (during communication at same potential)



UART mode bit width (during communication at same potential) (reference)



Remarks 1. q: UART number (q = 0 to 2), g: PIM, POM number (g = 0, 1)

2. f_{MCK}: Serial array unit operation clock frequency

(Operation clock to be set by the serial clock select register m (SPSm) and the CKSmn bit of serial mode register mn (SMRmn).

m: Unit number, n: Channel number (mn = 00 to 03, 10, 11))

Downloaded from Arrow.com.

<R> **(2) During communication at same potential (CSI mode) (master mode, SCK00... internal clock output, corresponding CSI00 only)**

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	HS (high-speed main) Mode		LS (low-speed main) Mode		Unit
			MIN.	MAX.	MIN.	MAX.	
SCK00 cycle time	t_{KCY1}	$t_{KCY1} \geq 2/f_{CLK}$	83.3		250		ns
SCK00 high-/low-level width	t_{KH1}, t_{KL1}	4.0 V ≤ V_{DD} ≤ 5.5 V	$t_{KCY1}/2-7$		$t_{KCY1}/2-50$		ns
		2.7 V ≤ V_{DD} ≤ 5.5 V	$t_{KCY1}/2-10$		$t_{KCY1}/2-50$		ns
SI00 setup time (to SCK00↑) Note 1	t_{SIK1}	4.0 V ≤ V_{DD} ≤ 5.5 V	23		110		ns
		2.7 V ≤ V_{DD} ≤ 5.5 V	33		110		ns
SI00 hold time (from SCK00↑) Note2	t_{KSI1}		10		10		ns
Delay time from SCK00↓ to SO00 output Note 3	t_{KSO1}	C = 20 pF Note 4		10		10	ns

Notes 1. When DAP00 = 0 and CKP00 = 0, or DAP00 = 1 and CKP00 = 1. The SI00 setup time becomes "to SCK00↓" when DAP00 = 0 and CKP00 = 1, or DAP00 = 1 and CKP00 = 0.

2. When DAP00 = 0 and CKP00 = 0, or DAP00 = 1 and CKP00 = 1. The SI00 hold time becomes "from SCK00↓" when DAP00 = 0 and CKP00 = 1, or DAP00 = 1 and CKP00 = 0.

3. When DAP00 = 0 and CKP00 = 0, or DAP00 = 1 and CKP00 = 1. The delay time to SO00 output becomes "from SCK00↑" when DAP00 = 0 and CKP00 = 1, or DAP00 = 1 and CKP00 = 0.

4. C is the load capacitance of the SCK00 and SO00 output lines.

Caution Select the normal input buffer for the SI00 pin and the normal output mode for the SO00 and SCK00 pins by using port input mode register 1 (PIM1) and port output mode register 1 (POM1).

Remarks 1. This specification is valid only when CSI00's peripheral I/O redirect function is not used.

2. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the serial clock select register 0 (SPS0) and the CKS00 bit of serial mode register 00 (SMR00).)

Downloaded from Arrow.com.

<R> **(3) During communication at same potential (CSI mode) (master mode, SCKp... internal clock output)**

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		HS (high-speed main) Mode		LS (low-speed main) Mode		Unit
				MIN.	MAX.	MIN.	MAX.	
SCKp cycle time	t_{KCY1}	$t_{KCY1} \geq 4/f_{CLK}$	2.7 V ≤ V_{DD} ≤ 5.5 V	167		500		ns
			2.4 V ≤ V_{DD} ≤ 5.5 V	250		500		ns
			1.8 V ≤ V_{DD} ≤ 5.5 V	–		500		ns
SCKp high-/low-level width	t_{KH1}, t_{KL1}	4.0 V ≤ V_{DD} ≤ 5.5 V		t_{KCY1}/2–12		t_{KCY1}/2–50		ns
		2.7 V ≤ V_{DD} ≤ 5.5 V		t_{KCY1}/2–18		t_{KCY1}/2–50		ns
		2.4 V ≤ V_{DD} ≤ 5.5 V		t_{KCY1}/2–38		t_{KCY1}/2–50		ns
		1.8 V ≤ V_{DD} ≤ 5.5 V		–		t_{KCY1}/2–50		ns
SIp setup time (to SCKp↑) Note 1	t_{SIK1}	4.0 V ≤ V_{DD} ≤ 5.5 V		44		110		ns
		2.7 V ≤ V_{DD} ≤ 5.5 V		44		110		ns
		2.4 V ≤ V_{DD} ≤ 5.5 V		75		110		ns
		1.8 V ≤ V_{DD} ≤ 5.5 V		–		110		ns
SIp hold time (from SCKp↑) Note 2	t_{KSI1}			19		19		ns
Delay time from SCKp↓ to SOp output Note 3	t_{KSO1}	C = 30 pF Note4			25		25	ns

Notes 1. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

2. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp hold time becomes "from SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

3. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to SOp output becomes "from SCKp↑" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

4. C is the load capacitance of the SCKp and SOp output lines.

Caution Select the normal input buffer for the SIp pin and the normal output mode for the SOp and SCKp pins by using port input mode register 1 (PIM1) and port output mode registers 0, 1, 4 (POM0, POM1, POM4).

Remarks 1. p: CSI number (p = 00, 01, 11, 20), m: Unit number (m = 0, 1), n: Channel number (n = 0, 1, 3: "1, 3" is only for the R5F102 products)

2. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the serial clock select register m (SPSm) and the CKSmn bit of serial mode register mn (SMRmn). m: Unit number (m = 0, 1), n: Channel number (n = 0, 1, 3: "1, 3" is only for the R5F102 products.))

Downloaded from Arrow.com.

<R> **(4) During communication at same potential (CSI mode) (slave mode, SCKp... external clock input)**

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		HS (high-speed main) Mode		LS (low-speed main) Mode		Unit
				MIN.	MAX.	MIN.	MAX.	
SCKp cycle time Note4	t_{KCY2}	4.0 V ≤ V_{DD} ≤ 5.5 V	20 MHz < f_{MCK}	8/f_{MCK}		–		ns
			f_{MCK} ≤ 20 MHz	6/f_{MCK}		6/f_{MCK}		ns
		2.7 V ≤ V_{DD} ≤ 5.5 V	16 MHz < f_{MCK}	8/f_{MCK}		–		ns
			f_{MCK} ≤ 16 MHz	6/f_{MCK}		6/f_{MCK}		ns
		2.4 V ≤ V_{DD} ≤ 5.5 V		6/f_{MCK} and 500		6/f_{MCK} and 500		ns
		1.8 V ≤ V_{DD} ≤ 5.5 V		–		6/f_{MCK} and 750		ns
SCKp high-/low-level width	t_{KH2}, t_{KL2}	4.0 V ≤ V_{DD} ≤ 5.5 V		t_{KCY2}/2–7		t_{KCY2}/2–7		ns
		2.7 V ≤ V_{DD} ≤ 5.5 V		t_{KCY2}/2–8		t_{KCY2}/2–8		ns
		2.4 V ≤ V_{DD} ≤ 5.5 V		t_{KCY2}/2–18		t_{KCY2}/2–18		ns
		1.8 V ≤ V_{DD} ≤ 5.5 V		–		t_{KCY2}/2–18		ns
SIp setup time (to SCKp↑) Note 1	t_{SIK2}	2.7 V ≤ V_{DD} ≤ 5.5 V		1/f_{MCK} + 20		1/f_{MCK} + 30		ns
		2.4 V ≤ V_{DD} ≤ 5.5 V		1/f_{MCK} + 30		1/f_{MCK} + 30		ns
		1.8 V ≤ V_{DD} ≤ 5.5 V		–		1/f_{MCK} + 30		ns
SIp hold time (from SCKp↑) Note 2	t_{KSI2}			1/f_{MCK} + 31		1/f_{MCK} + 31		ns
Delay time from SCKp↓ to SOp output Note 3	t_{KSO2}	C = 30 pF Note4	2.7 V ≤ V_{DD} ≤ 5.5 V		2/f_{MCK} + 44		2/f_{MCK} + 110	ns
			2.4 V ≤ V_{DD} ≤ 5.5 V		2/f_{MCK} + 75		2/f_{MCK} + 110	ns
			1.8 V ≤ V_{DD} ≤ 5.5 V		–		2/f_{MCK} + 110	ns

Notes 1. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

2. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp hold time becomes "from SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

3. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to SOp output becomes "from SCKp↑" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

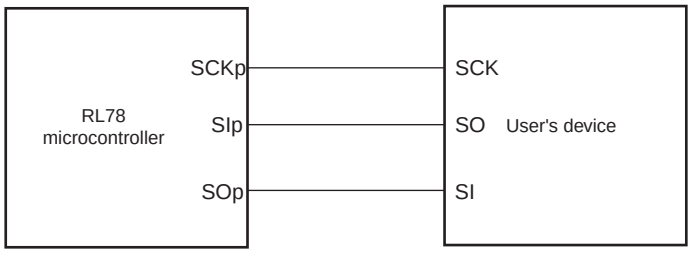
4. C is the load capacitance of the SOp output lines.

5. Transfer rate in the SNOOZE mode: MAX. 1 Mbps

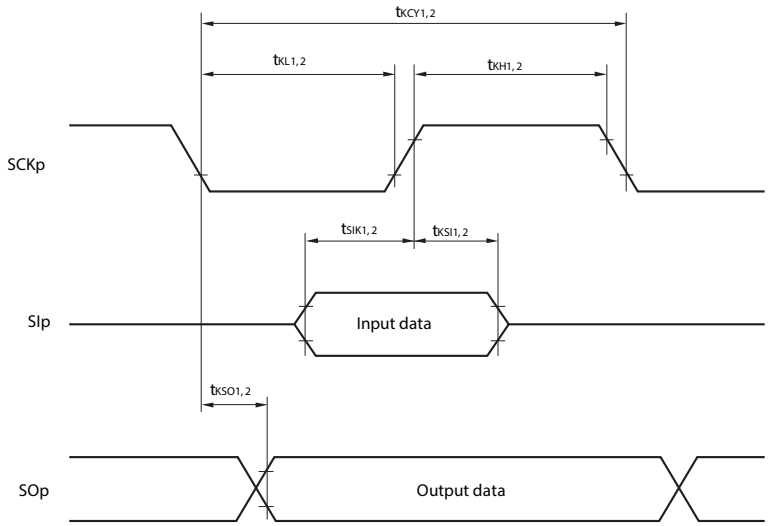
Caution Select the normal input buffer for the SIp and SCKp pins and the normal output mode for the SOp pin by using port input mode register 1 (PIM1) and port output mode registers 0, 1, 4 (POM0, POM1, POM4).

Downloaded from Arrow.com.

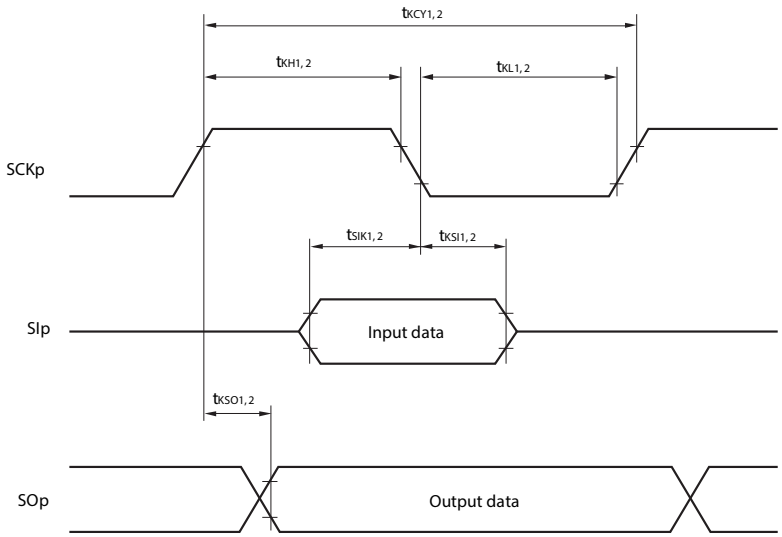
CSI mode connection diagram (during communication at same potential)



CSI mode serial transfer timing (during communication at same potential)
(When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.)



CSI mode serial transfer timing (during communication at same potential)
(When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.)



(**Remarks** are listed on the next page.)

Downloaded from Arrow.com.

Remarks 1. p: CSI number (p = 00, 01, 11, 20), m: Unit number (m = 0, 1), n: Channel number (n = 0, 1, 3: “1, 3” is only for the R5F102 products.)

2. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the serial clock select register m (SPSm) and the CKSmn bit of serial mode register mn (SMRmn). m: Unit number (m = 0, 1), n: Channel number (n = 0, 1, 3: “1, 3” is only for the R5F102 products.))

<R> **(5) During communication at same potential (simplified I^2C mode)**

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	HS (high-speed main) Mode LS (low-speed main) Mode		Unit
			MIN.	MAX.	
SCLr clock frequency	f_{SCL}	1.8 V ≤ V_{DD} ≤ 5.5 V, C_b = 100 pF, R_b = 3 kΩ		400 Note 1	kHz
		1.8 V ≤ V_{DD} < 2.7 V, C_b = 100 pF, R_b = 5 kΩ		300 Note 1	kHz
Hold time when SCLr = “L”	t_{LOW}	1.8 V ≤ V_{DD} ≤ 5.5 V, C_b = 100 pF, R_b = 3 kΩ	1150		ns
		1.8 V ≤ V_{DD} < 2.7 V, C_b = 100 pF, R_b = 5 kΩ	1550		ns
Hold time when SCLr = “H”	t_{HIGH}	1.8 V ≤ V_{DD} ≤ 5.5 V, C_b = 100 pF, R_b = 3 kΩ	1150		ns
		1.8 V ≤ V_{DD} < 2.7 V, C_b = 100 pF, R_b = 5 kΩ	1550		ns
Data setup time (reception)	$t_{SU:DAT}$	1.8 V ≤ V_{DD} ≤ 5.5 V, C_b = 100 pF, R_b = 3 kΩ	$1/f_{MCK}$ + 145 Note 2		ns
		1.8 V ≤ V_{DD} < 2.7 V, C_b = 100 pF, R_b = 5 kΩ	$1/f_{MCK}$ + 230 Note 2		ns
Data hold time (transmission)	$t_{HD:DAT}$	1.8 V ≤ V_{DD} ≤ 5.5 V, C_b = 100 pF, R_b = 3 kΩ	0	355	ns
		1.8 V ≤ V_{DD} < 2.7 V, C_b = 100 pF, R_b = 5 kΩ	0	405	ns

Notes 1. The value must also be equal to or less than f_{MCK}/4.

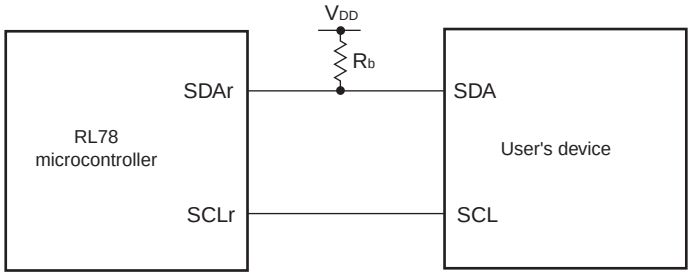
<R> **2.** Set $t_{SU:DAT}$ so that it will not exceed the hold time when SCLr = "L" or SCLr = "H".

Caution Select the N-ch open drain output (V_{DD} tolerance) mode for SDAr by using port output mode register h (POMh).

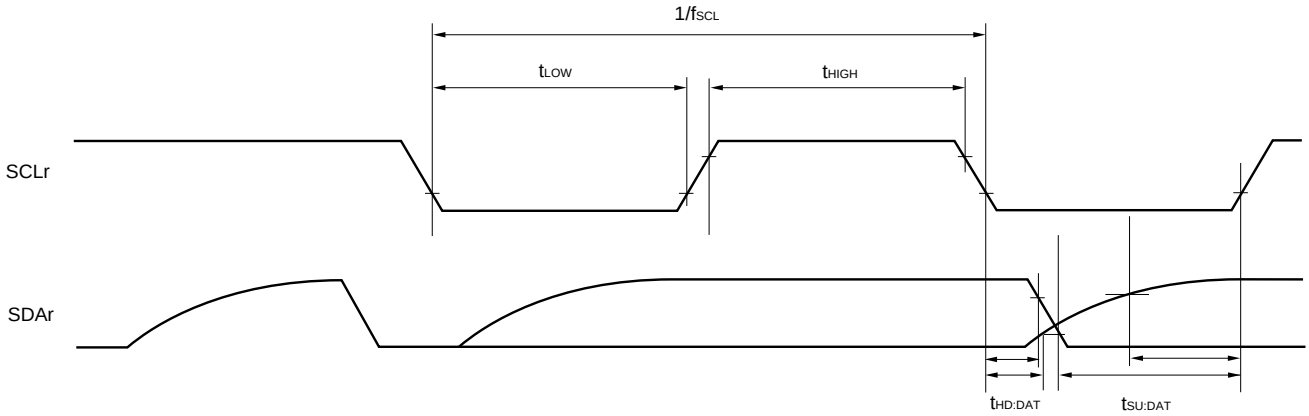
(**Remarks** are listed on the next page.)

Downloaded from Arrow.com.

Simplified I^2C mode connection diagram (during communication at same potential)



Simplified I^2C mode serial transfer timing (during communication at same potential)



Remarks 1. R_b [Ω]:Communication line (SDAr) pull-up resistance
C_b [F]: Communication line (SCLr, SDAr) load capacitance

2. r: IIC number (r = 00, 01, 11, 20), h: = POM number (h = 0, 1, 4, 5)

3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the serial clock select register m (SPSm) and the CKSmn bit of serial mode register mn (SMRmn). m: Unit number (m = 0, 1), n: Channel number (0, 1, 3))

4. Simplified I^2C mode is supported only by the R5F102 products.

Downloaded from Arrow.com.

<R> **(6) Communication at different potential (1.8 V, 2.5 V, 3 V) (UART mode)**

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions			HS (high-speed main) Mode		LS (low-speed main) Mode		Unit
					MIN.	MAX.	MIN.	MAX.	
Transfer rate Note4		Reception	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V			f_{MCK}/6 Note1		f_{MCK}/6 Note1	bps
				Theoretical value of the maximum transfer rate f_{MCK} = f_{CLK} Note3		4.0		1.3	Mbps
			2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V			f_{MCK}/6 Note1		f_{MCK}/6 Note1	bps
				Theoretical value of the maximum transfer rate f_{MCK} = f_{CLK} Note3		4.0		1.3	Mbps
			1.8 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V			f_{MCK}/6 Notes1, 2		f_{MCK}/6 Notes1, 2	bps
				Theoretical value of the maximum transfer rate f_{MCK} = f_{CLK} Note3		4.0		1.3	Mbps
		Transmission	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V			Note4		Note4	bps
				Theoretical value of the maximum transfer rate C_b = 50 pF, R_b = 1.4 kΩ, V_b = 2.7 V		2.8 Note5		2.8 Note5	Mbps
			2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V,			Note6		Note6	bps
				Theoretical value of the maximum transfer rate C_b = 50 pF, R_b = 2.7 kΩ, V_b = 2.3 V		1.2 Note7		1.2 Note7	Mbps
			1.8 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V			Notes 2, 8		Notes 2, 8	bps
				Theoretical value of the maximum transfer rate C_b = 50 pF, R_b = 5.5 kΩ, V_b = 1.6 V		0.43 Note9		0.43 Note9	Mbps

<R> **Notes 1.** Transfer rate in the SNOOZE mode is 4800 bps only.

<R> **2.** Use it with $V_{DD} \geq V_b$.

<R> **3.** The maximum operating frequencies of the CPU/peripheral hardware clock (f_{CLK}) are:

HS (high-speed main) mode: 24 MHz (2.7 V ≤ V_{DD} ≤ 5.5 V)

16 MHz (2.4 V ≤ V_{DD} ≤ 5.5 V)

LS (low-speed main) mode: 8 MHz (1.8 V ≤ V_{DD} ≤ 5.5 V)

<R> **4.** The smaller maximum transfer rate derived by using f_{MCK}/6 or the following expression is the valid maximum transfer rate.

Expression for calculating the transfer rate when 4.0 V ≤ V_{DD} ≤ 5.5 V and 2.7 V ≤ V_b ≤ 4.0 V

$$\text{Maximum transfer rate} = \frac{1}{\{-C_b \times R_b \times \ln(1 - \frac{2.2}{V_b})\} \times 3} \text{ [bps]}$$

$$\text{Baud rate error (theoretical value)} = \frac{\frac{1}{\text{Transfer rate} \times 2} - \{-C_b \times R_b \times \ln(1 - \frac{2.2}{V_b})\}}{(\frac{1}{\text{Transfer rate}}) \times \text{Number of transferred bits}} \times 100 \text{ [\%]}$$

* This value is the theoretical value of the relative difference between the transmission and reception sides.

Downloaded from Arrow.com.

<R> **5.** This value as an example is calculated when the conditions described in the “Conditions” column are met. Refer to **Note 4** above to calculate the maximum transfer rate under conditions of the customer.

<R> **6.** The smaller maximum transfer rate derived by using f_{MCK}/6 or the following expression is the valid maximum transfer rate.
Expression for calculating the transfer rate when 2.7 V ≤ V_{DD} < 4.0 V and 2.3 V ≤ V_b ≤ 2.7 V

$$\text{Maximum transfer rate} = \frac{1}{\{-C_b \times R_b \times \ln(1 - \frac{2.0}{V_b})\} \times 3} \text{ [bps]}$$

$$\text{Baud rate error (theoretical value)} = \frac{\frac{1}{\text{Transfer rate} \times 2} - \{-C_b \times R_b \times \ln(1 - \frac{2.0}{V_b})\}}{(\frac{1}{\text{Transfer rate}}) \times \text{Number of transferred bits}} \times 100 \text{ [\%]}$$

* This value is the theoretical value of the relative difference between the transmission and reception sides.

<R> **7.** This value as an example is calculated when the conditions described in the “Conditions” column are met. Refer to **Note 6** above to calculate the maximum transfer rate under conditions of the customer.

<R> **8.** The smaller maximum transfer rate derived by using f_{MCK}/6 or the following expression is the valid maximum transfer rate.
Expression for calculating the transfer rate when 1.8 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V

$$\text{Maximum transfer rate} = \frac{1}{\{-C_b \times R_b \times \ln(1 - \frac{1.5}{V_b})\} \times 3} \text{ [bps]}$$

$$\text{Baud rate error (theoretical value)} = \frac{\frac{1}{\text{Transfer rate} \times 2} - \{-C_b \times R_b \times \ln(1 - \frac{1.5}{V_b})\}}{(\frac{1}{\text{Transfer rate}}) \times \text{Number of transferred bits}} \times 100 \text{ [\%]}$$

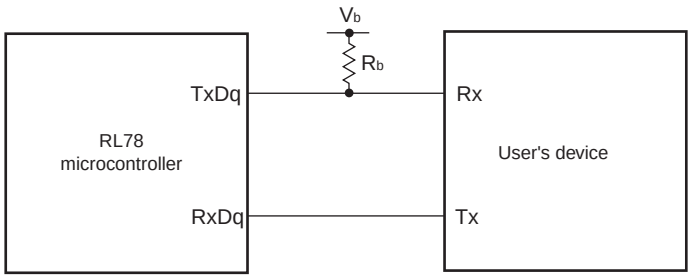
* This value is the theoretical value of the relative difference between the transmission and reception sides.

<R> **9.** This value as an example is calculated when the conditions described in the “Conditions” column are met. Refer to **Note 8** above to calculate the maximum transfer rate under conditions of the customer.

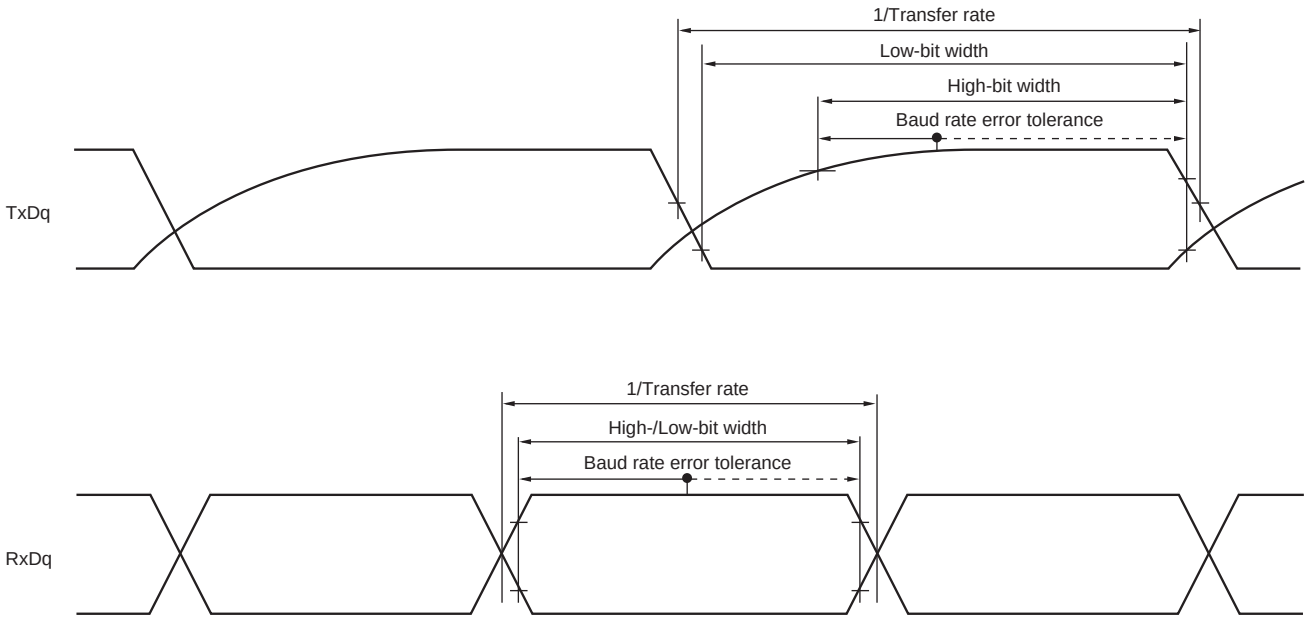
Caution Select the TTL input buffer for the RxDq pin and the N-ch open drain output (V_{DD} tolerance) mode for the TxDq pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.

Downloaded from Arrow.com.

UART mode connection diagram (during communication at different potential)



UART mode bit width (during communication at different potential) (reference)



<R> **Remarks 1.** R_b[Ω]: Communication line (TxDq) pull-up resistance, C_b[F]: Communication line (TxDq) load capacitance, V_b[V]: Communication line voltage

<R> **2.** q: UART number (q = 0 to 2), g: PIM and POM number (g = 0, 1)

<R> **3.** f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the serial clock select register m (SPSm) and the CKSmn bit of serial mode register mn (SMRmn).
m: Unit number, n: Channel number (mn = 00 to 03, 10, 11))

4. UART0 of the 20- and 24-pin products supports communication at different potential only when the peripheral I/O redirection function is not used.

Downloaded from Arrow.com.

<R> **(7) Communication at different potential (2.5 V, 3 V) (CSI mode) (master mode, SCK00... internal clock output, corresponding CSI00 only)**

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		HS (high-speed main) Mode		LS (low-speed main) Mode		Unit
				MIN.	MAX.	MIN.	MAX.	
SCK00 cycle time	t_{KCY1}	$t_{KCY1} \geq 2/f_{CLK}$	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ	200		1150		ns
			2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ	300		1150		ns
SCK00 high-level width	t_{KH1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ		t_{KCY1}/2 – 50		t_{KCY1}/2– 50		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ		t_{KCY1}/2 – 120		t_{KCY1}/2 – 120		ns
SCK00 low-level width	t_{KL1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ		t_{KCY1}/2 – 7		t_{KCY1}/2 – 50		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ		t_{KCY1}/2 – 10		t_{KCY1}/2 – 50		ns
SI00 setup time (to SCK00↑) Note 1	t_{SIK1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ		58		479		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ		121		479		ns
SI00 hold time (from SCK00↑) Note 1	t_{KSI1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ		10		10		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ		10		10		ns
Delay time from SCK00↓ to SO00 output Note 1	t_{KSO1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ			60		60	ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ			130		130	ns
SI00 setup time (to SCK00↓) Note 2	t_{SIK1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ		23		110		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ		33		110		ns
SI00 hold time (from SCK00↓) Note 2	t_{KSI1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ		10		10		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ		10		10		ns
Delay time from SCK00↑ to SO00 output Note 2	t_{KSO1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ			10		10	ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ			10		10	ns

(**Notes**, **Caution**, and **Remarks** are listed on the next page.)

Downloaded from Arrow.com.

Notes 1. When DAP00 = 0 and CKP00 = 0, or DAP00 = 1 and CKP00 = 1

2. When DAP00 = 0 and CKP00 = 1, or DAP00 = 1 and CKP00 = 0.

<R> **Caution Select the TTL input buffer for the SI00 pin and the N-ch open drain output (V_{DD} tolerance) mode for the SO00 pin and SCK00 pin by using port input mode register 1 (PIM1) and port output mode register 1 (POM1). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.**

Remarks 1. R_b [Ω]:Communication line (SCK00, SO00) pull-up resistance, C_b [F]: Communication line (SCK00, SO00) load capacitance, V_b [V]: Communication line voltage

2. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the serial clock select register 0 (SPS0) and the CKS00 bit of serial mode register 00 (SMR00).)

Downloaded from Arrow.com.

<R> **(8) Communication at different potential (1.8 V, 2.5 V, 3 V) (CSI mode) (master mode, SCKp... internal clock output) (1/3)**

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		HS (high-speed main) Mode		LS (low-speed main) Mode		Unit
				MIN.	MAX.	MIN.	MAX.	
SCKp cycle time	t_{KCY1}	$t_{KCY1} \geq 4/f_{CLK}$	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	300		1150		ns
			2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	500		1150		ns
			1.8 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note, C_b = 30 pF, R_b = 5.5 kΩ	1150		1150		ns
SCKp high-level width	t_{KH1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ		t_{KCY1}/2 –75		t_{KCY1}/2–75		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		t_{KCY1}/2 –170		t_{KCY1}/2–170		ns
		1.8 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note, C_b = 30 pF, R_b = 5.5 kΩ		t_{KCY1}/2 –458		t_{KCY1}/2–458		ns
SCKp low-level width	t_{KL1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ		t_{KCY1}/2 –12		t_{KCY1}/2–50		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		t_{KCY1}/2 –18		t_{KCY1}/2–50		ns
		1.8 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note, C_b = 30 pF, R_b = 5.5 kΩ		t_{KCY1}/2 –50		t_{KCY1}/2–50		ns

Note Use it with $V_{DD} \geq V_b$.

Cautions 1. Select the TTL input buffer for the SIp pin and the N-ch open drain output (V_{DD} tolerance) mode for the SOp pin and SCKp pin by using port input mode register 1 (PIM1) and port output mode register 1 (POM1). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.

2. CSI01 and CSI11 cannot communicate at different potential.

Remarks 1. R_b [Ω]: Communication line (SCKp, SOp) pull-up resistance, C_b [F]: Communication line (SCKp, SOp) load capacitance, V_b [V]: Communication line voltage

2. p: CSI number (p = 00, 20)

Downloaded from Arrow.com.

<R> **(8) Communication at different potential (1.8 V, 2.5 V, 3 V) (CSI mode) (master mode, SCKp... internal clock output) (2/3)**

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	HS (high-speed main) Mode		LS (low-speed main) Mode		Unit
			MIN.	MAX.	MIN.	MAX.	
SIp setup time (to SCKp↑) Note 1	t_{SIK1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	81		479		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	177		479		ns
		1.8 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 2, C_b = 30 pF, R_b = 5.5 kΩ	479		479		ns
SIp hold time (from SCKp↑) Note 1	t_{KSI1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	19		19		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	19		19		ns
		1.8 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 2, C_b = 30 pF, R_b = 5.5 kΩ	19		19		ns
Delay time from SCKp↓ to SOp output Note 1	t_{KSO1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ		100		100	ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		195		195	ns
		1.8 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 2, C_b = 30 pF, R_b = 5.5 kΩ		483		483	ns

<R> **Notes 1.** When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.

<R> **2.** Use it with $V_{DD} \geq V_b$.

(**Cautions** and **Remarks** are listed on the next page.)

Downloaded from Arrow.com.

<R> **(8) Communication at different potential (1.8 V, 2.5 V, 3 V) (CSI mode) (master mode, SCKp... internal clock output) (3/3)**

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	HS (high-speed main) Mode		LS (low-speed main) Mode		Unit
			MIN.	MAX.	MIN.	MAX.	
SIp setup time (to SCKp↓) Note 1	t_{SIK1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	44		110		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	44		110		ns
		1.8 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 2, C_b = 30 pF, R_b = 5.5 kΩ	110		110		ns
SIp hold time (from SCKp↓) Note 1	t_{KSI1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	19		19		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	19		19		ns
		1.8 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 2, C_b = 30 pF, R_b = 5.5 kΩ	19		19		ns
Delay time from SCKp↑ to SOp output Note 1	t_{KSO1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ		25		25	ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		25		25	ns
		1.8 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 2, C_b = 30 pF, R_b = 5.5 kΩ		25		25	ns

<R> **Notes 1.** When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

2. Use it with $V_{DD} \geq V_b$.

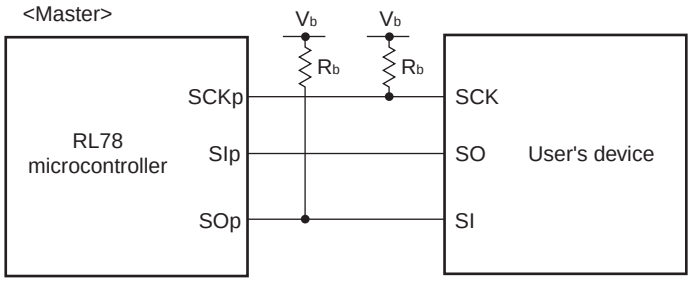
<R> **Cautions 1. Select the TTL input buffer for the SIp pin and the N-ch open drain output (V_{DD} tolerance) mode for the SOp pin and SCKp pin by using port input mode register 1 (PIM1) and port output mode register 1 (POM1). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.**

2. CSI01 and CSI11 cannot communicate at different potential.

Remarks 1. R_b [Ω]: Communication line (SCKp, SOp) pull-up resistance, C_b [F]: Communication line (SCKp, SOp) load capacitance, V_b [V]: Communication line voltage

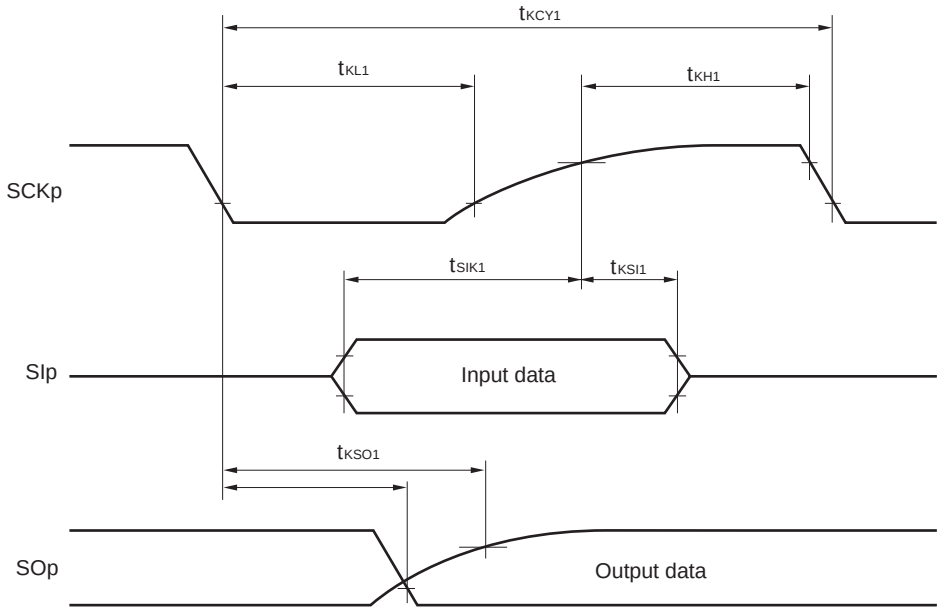
2. p: CSI number (p = 00, 20), m: Unit number (m = 0, 1), n: Channel number (n = 0)

CSI mode connection diagram (during communication at different potential)

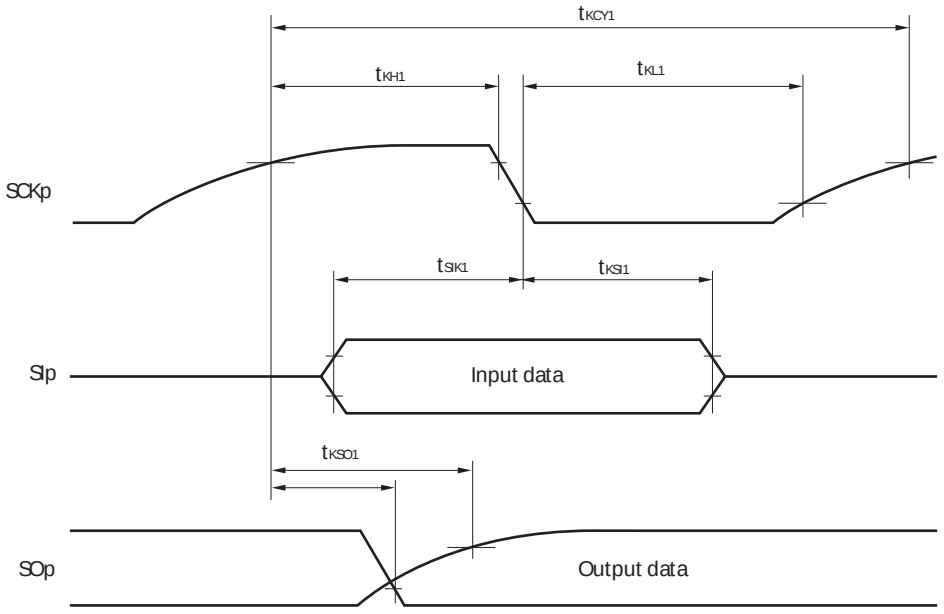


Downloaded from Arrow.com.

CSI mode serial transfer timing (master mode) (during communication at different potential)
(When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1)



CSI mode serial transfer timing (master mode) (during communication at different potential)
(When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.)



Downloaded from Arrow.com.

<R> **(9) Communication at different potential (1.8 V, 2.5 V, 3 V) (CSI mode) (slave mode, SCKp... external clock input)**
(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		HS (high-speed main) Mode		LS (low-speed main) Mode		Unit
				MIN.	MAX.	MIN.	MAX.	
SCKp cycle time Note 1	t_{KCY2}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V	20 MHz < f_{MCK} ≤ 24 MHz	12/f_{MCK}		–		ns
			8 MHz < f_{MCK} ≤ 20 MHz	10/f_{MCK}		–		ns
			4 MHz < f_{MCK} ≤ 8 MHz	8/f_{MCK}		16/f_{MCK}		ns
			f_{MCK} ≤ 4 MHz	6/f_{MCK}		10/f_{MCK}		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V	20 MHz < f_{MCK} ≤ 24 MHz	16/f_{MCK}		–		ns
			16 MHz < f_{MCK} ≤ 20 MHz	14/f_{MCK}		–		ns
			8 MHz < f_{MCK} ≤ 16 MHz	12/f_{MCK}		–		ns
			4 MHz < f_{MCK} ≤ 8 MHz	8/f_{MCK}		16/f_{MCK}		ns
			f_{MCK} ≤ 4 MHz	6/f_{MCK}		10/f_{MCK}		ns
		1.8 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 2	20 MHz < f_{MCK} ≤ 24 MHz	36/f_{MCK}		–		ns
			16 MHz < f_{MCK} ≤ 20 MHz	32/f_{MCK}		–		ns
			8 MHz < f_{MCK} ≤ 16 MHz	26/f_{MCK}		–		ns
			4 MHz < f_{MCK} ≤ 8 MHz	16/f_{MCK}		16/f_{MCK}		ns
			f_{MCK} ≤ 4 MHz	10/f_{MCK}		10/f_{MCK}		ns
SCKp high-/low-level width	t_{KH2}, t_{KL2}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V		t_{KCY2}/2 – 12		t_{KCY2}/2 – 50		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V		t_{KCY2}/2 – 18		t_{KCY2}/2 – 50		ns
		1.8 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 2		t_{KCY2}/2 – 50		t_{KCY2}/2 – 50		ns
SIp setup time (to SCKp↑) Note 3	t_{SIK2}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_{DD} ≤ 4.0 V		1/f_{MCK} + 20		1/f_{MCK} + 30		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V		1/f_{MCK} + 20		1/f_{MCK} + 30		ns
		1.8 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_{DD} ≤ 2.0 V Note 2		1/f_{MCK} + 30		1/f_{MCK} + 30		ns
SIp hold time (from SCKp↑) Note 4	t_{KSI2}			1/f_{MCK} + 31		1/f_{MCK} + 31		ns
Delay time from SCKp↓ to SOp output Note 5	t_{KSO2}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ			2/f_{MCK} + 120		2/f_{MCK} + 573	ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ			2/f_{MCK} + 214		2/f_{MCK} + 573	ns
		1.8 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 2, C_b = 30 pF, R_b = 5.5 kΩ			2/f_{MCK} + 573		2/f_{MCK} + 573	ns

Notes 1. Transfer rate in the SNOOZE mode: MAX. 1 Mbps

2. Use it with $V_{DD} \geq V_b$.

3. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

4. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp hold time becomes "from SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

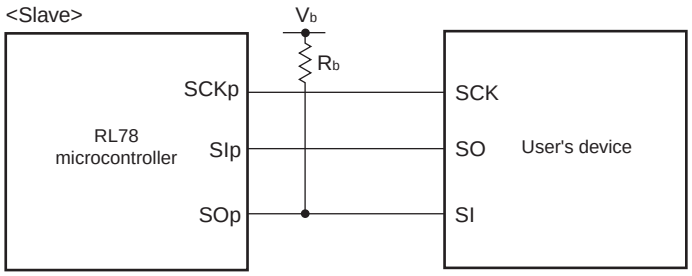
5. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to SOp output becomes "from SCKp↑" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Cautions 1. Select the TTL input buffer for the SIp and SCKp pins and the N-ch open drain output (V_{DD} tolerance) mode for the SOp pin by using port input mode register 1 (PIM1) and port output mode register 1 (POM1). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.

2. CSI01 and CSI11 cannot communicate at different potential.

Downloaded from Arrow.com.

CSI mode connection diagram (during communication at different potential)

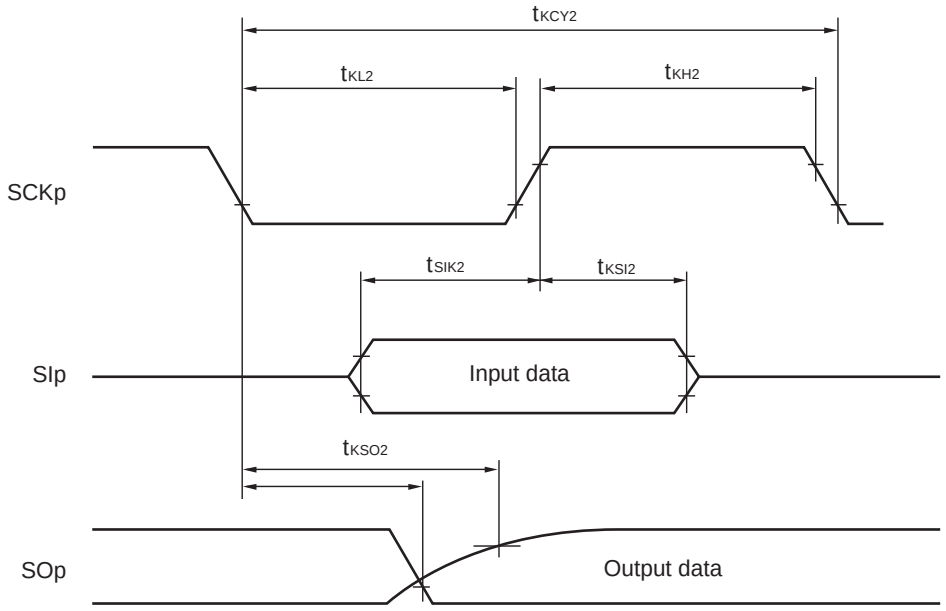


Remarks 1. R_b [Ω]: Communication line (SOp) pull-up resistance, C_b [F]: Communication line (SOp) load capacitance, V_b [V]: Communication line voltage

2. p: CSI number (p = 00, 20), m: Unit number (m = 0, 1), n: Channel number (n = 0)

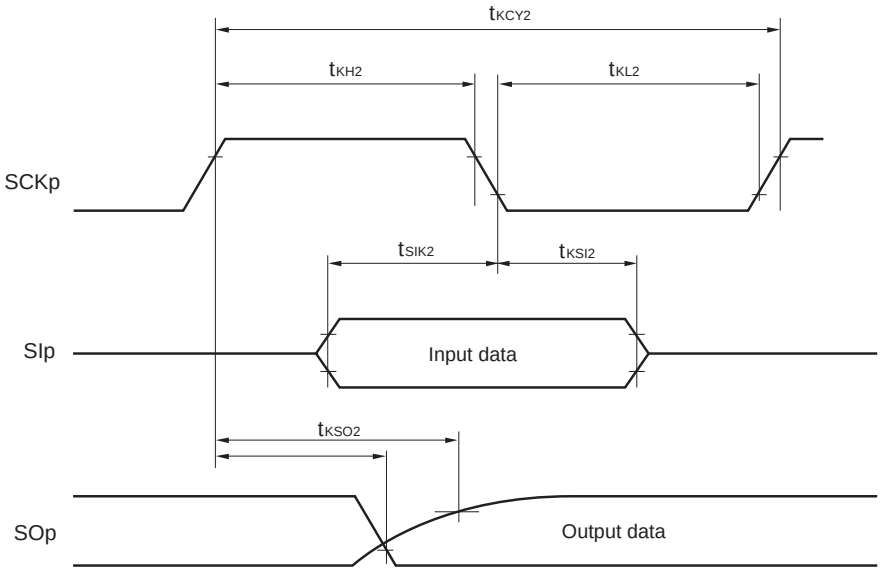
3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the serial clock select register m (SPSm) and the CKSmn bit of serial mode register mn (SMRmn). m: Unit number, n: Channel number (mn = 00, 10))

CSI mode serial transfer timing (slave mode) (during communication at different potential)
(When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.)



Downloaded from Arrow.com.

CSI mode serial transfer timing (slave mode) (during communication at different potential)
(When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.)



Remark p: CSI number (p = 00, 20), m: Unit number (m = 0, 1), n: Channel number (n = 0)

Downloaded from Arrow.com.

<R> **(10) Communication at different potential (1.8 V, 2.5 V, 3 V) (simplified I^2C mode)**

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	HS (high-speed main) Mode		LS (low-speed main) Mode		Unit
			MIN.	MAX.	MIN.	MAX.	
SCLr clock frequency	f_{SCL}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 2.8 kΩ		400 Note1		300 Note1	kHz
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ		400 Note1		300 Note1	kHz
		1.8 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V, Note2 C_b = 100 pF, R_b = 5.5 kΩ		300 Note1		300 Note1	kHz
Hold time when SCLr = "L"	t_{LOW}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 2.8 kΩ	1150		1550		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ	1150		1550		ns
		1.8 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V, Note2 C_b = 100 pF, R_b = 5.5 kΩ	1550		1550		ns
Hold time when SCLr = "H"	t_{HIGH}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 2.8 kΩ	675		610		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ	600		610		ns
		1.8 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V, Note2 C_b = 100 pF, R_b = 5.5 kΩ	610		610		ns
Data setup time (reception)	$t_{SU:DAT}$	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 2.8 kΩ	1/f_{MCK} + 190 Note3		1/f_{MCK} + 190 Note3		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ	1/f_{MCK} + 190 Note3		1/f_{MCK} + 190 Note3		ns
		1.8 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V, Note2 C_b = 100 pF, R_b = 5.5 kΩ	1/f_{MCK} + 190 Note3		1/f_{MCK} + 190 Note3		ns
Data hold time (transmission)	$t_{HD:DAT}$	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 2.8 kΩ	0	355	0	355	ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ	0	355	0	355	ns
		1.8 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V, Note2 C_b = 100 pF, R_b = 5.5 kΩ	0	405	0	405	ns

<R> **Notes 1.** The value must also be equal to or less than f_{MCK}/4.

2. Use it with $V_{DD} \geq V_b$.

3. Set $t_{SU:DAT}$ so that it will not exceed the hold time when SCLr = "L" or SCLr = "H".

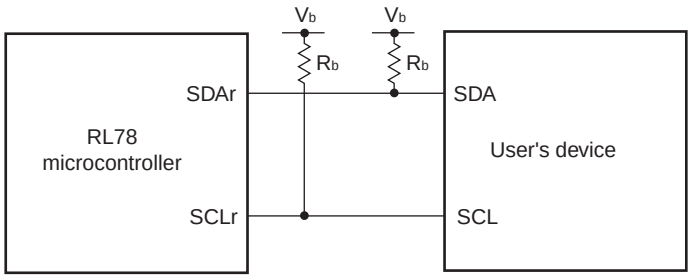
<R> **Cautions 1. Select the TTL input buffer and the N-ch open drain output (V_{DD} tolerance) mode for the SDAr pin and the N-ch open drain output (V_{DD} tolerance) mode for the SCLr pin by using port input mode register 1 (PIM1) and port output mode register 1 (POM1). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.**

2. IIC01 and IIC11 cannot communicate at different potential.

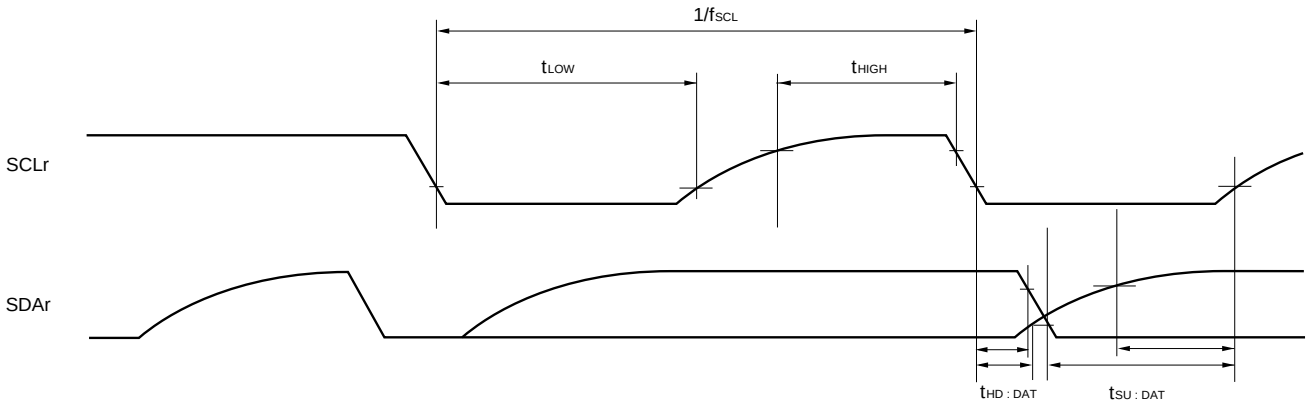
(**Remarks** are listed on the next page.)

Downloaded from Arrow.com.

Simplified I^2C mode connection diagram (during communication at different potential)



Simplified I^2C mode serial transfer timing (during communication at different potential)



Remarks 1. R_b [Ω]: Communication line (SDAr, SCLr) pull-up resistance, C_b [F]: Communication line (SDAr, SCLr) load capacitance, V_b [V]: Communication line voltage

2. r: IIC Number (r = 00, 20)

3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the serial clock select register m (SPSm) and the CKSmn bit of serial mode register mn (SMRmn).
m: Unit number (m = 0,1), n: Channel number (n = 0))

4. Simplified I^2C mode is supported only by the R5F102 products.

Downloaded from Arrow.com.

28.5.2 Serial interface IICA

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	HS (high-speed main) mode / LS (low-speed main) mode				Unit
			Standard Mode		Fast Mode		
			MIN.	MAX.	MIN.	MAX.	
SCLA0 clock frequency	f_{SCL}	Fast mode: f_{CLK} ≥ 3.5 MHz			0	400	kHz
		Normal mode: f_{CLK} ≥ 1 MHz	0	100			kHz
Setup time of restart condition	$t_{SU:STA}$		4.7		0.6		μs
Hold time[Note 1]	$t_{HD:STA}$		4.0		0.6		μs
Hold time when SCLA0 = "L"	t_{LOW}		4.7		1.3		μs
Hold time when SCLA0 = "H"	t_{HIGH}		4.0		0.6		μs
Data setup time (reception)	$t_{SU:DAT}$		250		100		ns
Data hold time (transmission)[Note 2]	$t_{HD:DAT}$		0	3.45	0	0.9	μs
Setup time of stop condition	$t_{SU:STO}$		4.0		0.6		μs
Bus-free time	t_{BUF}		4.7		1.3		μs

Notes 1. The first clock pulse is generated after this period when the start/restart condition is detected.

2. The maximum value (MAX.) of $t_{HD:DAT}$ is during normal transfer and a wait state is inserted in the $\overline{ACK}$ (acknowledge) timing.

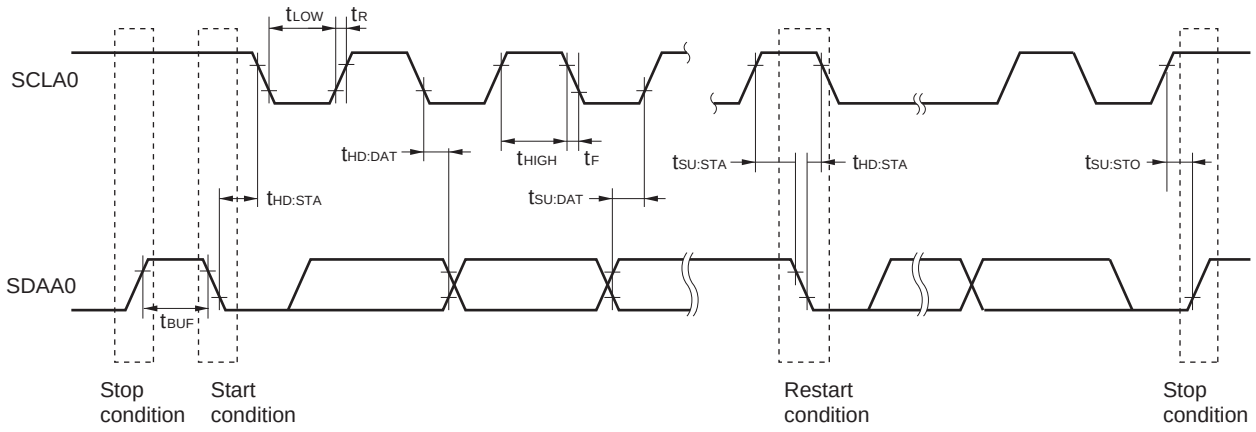
Caution Only in the 30-pin products, the values in the above table are applied even when bit 2 (PIOR2) in the peripheral I/O redirection register (PIOR) is 1. At this time, the pin characteristics (I_{OH1}, I_{OL1}, V_{OH1}, V_{OL1}) must satisfy the values in the redirect destination.

Remark The maximum value of C_b (communication line capacitance) and the value of R_b (communication line pull-up resistor) at that time in each mode are as follows.

Normal mode: C_b = 400 pF, Rb = 2.7 kΩ

<R> Fast mode: C_b = 320 pF, Rb = 1.1 kΩ

IICA serial transfer timing



Downloaded from Arrow.com.

28.6 Analog Characteristics

<R> 28.6.1 A/D converter characteristics

Classification of A/D converter characteristics

Input channel	Reference Voltage		
	Reference voltage (+) = AV_{REFP} Reference voltage (–) = AV_{REFM}	Reference voltage (+) = V_{DD} Reference voltage (–) = V_{SS}	Reference voltage (+) = V_{BGR} Reference voltage (–) = AV_{REFM}
ANI0 to ANI3	Refer to **28.6.1 (1)**.	Refer to **28.6.1 (3)**.	Refer to **28.6.1 (4)**.
ANI16 to ANI22	Refer to **28.6.1 (2)**.		
Internal reference voltage Temperature sensor output voltage	Refer to **28.6.1 (1)**.		–

<R> **(1) When reference voltage (+) = AV_{REFP}/ANI0 (ADREFP1 = 0, ADREFP0 = 1), reference voltage (–) = AV_{REFM}/ANI1 (ADREFM = 1), target pin: ANI2, ANI3, internal reference voltage, and temperature sensor output voltage**

(T_A = –40 to +85°C, 1.8 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V, Reference voltage (+) = AV_{REFP}, Reference voltage (–) = AV_{REFM} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	RES			8		10	bit
Overall error[Note 1]	AINL	10-bit resolution AV_{REFP} = V_{DD} [Note 3]			1.2	±3.5	LSB
					1.2	±7.0 [Note 4]	LSB
Conversion time	t_{CONV}	10-bit resolution Target pin: ANI2, ANI3	3.6 V ≤ V_{DD} ≤ 5.5 V	2.125		39	μs
			2.7 V ≤ V_{DD} ≤ 5.5 V	3.1875		39	μs
			1.8 V ≤ V_{DD} ≤ 5.5 V	17		39	μs
				57		95	μs
		10-bit resolution Target pin: Internal reference voltage, and temperature sensor output voltage (HS (high-speed main) mode)	3.6 V ≤ V_{DD} ≤ 5.5 V	2.375		39	μs
			2.7 V ≤ V_{DD} ≤ 5.5 V	3.5625		39	μs
			2.4 V ≤ V_{DD} ≤ 5.5 V	17		39	μs
Zero-scale error[Notes 1, 2]	EZS	10-bit resolution AV_{REFP} = V_{DD} [Note 3]				±0.25	%FSR
						±0.50 [Note 4]	%FSR
Full-scale error[Notes 1, 2]	EFS	10-bit resolution AV_{REFP} = V_{DD} [Note 3]				±0.25	%FSR
						±0.50 [Note 4]	%FSR
Integral linearity error[Note 1]	ILE	10-bit resolution AV_{REFP} = V_{DD} [Note 3]				±2.5	LSB
						±5.0 [Note 4]	LSB
Differential linearity error [Note 1]	DLE	10-bit resolution AV_{REFP} = V_{DD} [Note 3]				±1.5	LSB
						±2.0 [Note 4]	LSB
Analog input voltage	V_{AIN}	ANI2, ANI3		0		AV_{REFP}	V
		Internal reference voltage (2.4 V ≤ V_{DD} ≤ 5.5 V, HS (high-speed main) mode)		V_{BGR} [Note 5]			V
		Temperature sensor output voltage (2.4 V ≤ V_{DD} ≤ 5.5 V, HS (high-speed main) mode)		V_{TMPS25} [Note 5]			V

(**Notes** are listed on the next page.)

Downloaded from Arrow.com.

Notes 1. Excludes quantization error (±1/2 LSB).

2. This value is indicated as a ratio (%FSR) to the full-scale value.

<R> **3.** When $AV_{REFP} < V_{DD}$, the MAX. values are as follows.
Overall error: Add ±1.0 LSB to the MAX. value when $AV_{REFP} = V_{DD}$.
Zero-scale error/Full-scale error: Add ±0.05%FSR to the MAX. value when $AV_{REFP} = V_{DD}$.
Integral linearity error/ Differential linearity error: Add ±0.5 LSB to the MAX. value when $AV_{REFP} = V_{DD}$.

<R> **4.** Values when the conversion time is set to 57 μs (min.) and 95 μs (max.).

<R> **5.** Refer to **28.6.2 Temperature sensor/internal reference voltage characteristics**.

<R> **(2) When reference voltage (+) = AV_{REFP}/ANI0 (ADREFP1 = 0, ADREFP0 = 1), reference voltage (–) = AV_{REFM}/ANI1 (ADREFM = 1), target pin: ANI16 to ANI22**

(T_A = –40 to +85°C, 1.8 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V, Reference voltage (+) = AV_{REFP}, Reference voltage (–) = AV_{REFM} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	R_{ES}			8		10	bit
Overall error Note 1	AINL	10-bit resolution $AV_{REFP} = V_{DD}$ Note 3			1.2	±5.0	LSB
					1.2	±8.5 Note 4	LSB
Conversion time	t_{CONV}	10-bit resolution Target ANI pin: ANI16 to ANI22	3.6 V ≤ V_{DD} ≤ 5.5 V	2.125		39	μs
			2.7 V ≤ V_{DD} ≤ 5.5 V	3.1875		39	μs
			1.8 V ≤ V_{DD} ≤ 5.5 V	17		39	μs
				57		95	μs
Zero-scale error Notes 1, 2	EZS	10-bit resolution $AV_{REFP} = V_{DD}$ Note 3				±0.35	%FSR
						±0.60 Note 4	%FSR
Full-scale error Notes 1, 2	EFS	10-bit resolution $AV_{REFP} = V_{DD}$ Note 3				±0.35	%FSR
						±0.60 Note 4	%FSR
Integral linearity error Note 1	ILE	10-bit resolution $AV_{REFP} = V_{DD}$ Note 3				±3.5	LSB
						±6.0 Note 4	LSB
Differential linearity error Note 1	DLE	10-bit resolution $AV_{REFP} = V_{DD}$ Note 3				±2.0	LSB
						±2.5 Note 4	LSB
Analog input voltage	V_{AIN}	ANI16 to ANI22		0		AV_{REFP} and V_{DD}	V

Notes 1. Excludes quantization error (±1/2 LSB).

<R> **2.** This value is indicated as a ratio (%FSR) to the full-scale value.

<R> **3.** When $AV_{REFP} \leq V_{DD}$, the MAX. values are as follows.
Overall error: Add ±4.0 LSB to the MAX. value when $AV_{REFP} = V_{DD}$.
Zero-scale error/Full-scale error: Add ±0.20%FSR to the MAX. value when $AV_{REFP} = V_{DD}$.
Integral linearity error/ Differential linearity error: Add ±2.0 LSB to the MAX. value when $AV_{REFP} = V_{DD}$.

<R> **4.** When the conversion time is set to 57 μs (min.) and 95 μs (max.).

Downloaded from Arrow.com.

<R> **(3) When reference voltage (+) = V_{DD} (ADREFP1 = 0, ADREFP0 = 0), reference voltage (–) = V_{SS} (ADREFM = 0), target pin: ANI0 to ANI3, ANI16 to ANI22, internal reference voltage, and temperature sensor output voltage**

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V, Reference voltage (+) = V_{DD}, Reference voltage (–) = V_{SS})

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	R_{ES}			8		10	bit
Overall error Note 1	AINL	10-bit resolution			1.2	±7.0	LSB
					1.2	±10.5 Note 3	LSB
Conversion time	t_{CONV}	10-bit resolution Target pin: ANI0 to ANI3, ANI16 to ANI22	3.6 V ≤ V_{DD} ≤ 5.5 V	2.125		39	μs
			2.7 V ≤ V_{DD} ≤ 5.5 V	3.1875		39	μs
			1.8 V ≤ V_{DD} ≤ 5.5 V	17		39	μs
				57		95	μs
Conversion time	t_{CONV}	10-bit resolution Target pin: internal reference voltage, and temperature sensor output voltage (HS (high-speed main) mode)	3.6 V ≤ V_{DD} ≤ 5.5 V	2.375		39	μs
			2.7 V ≤ V_{DD} ≤ 5.5 V	3.5625		39	μs
			2.4 V ≤ V_{DD} ≤ 5.5 V	17		39	μs
Zero-scale error Notes 1, 2	EZS	10-bit resolution				±0.60	%FSR
						±0.85 Note 3	%FSR
Full-scale error Notes 1, 2	EFS	10-bit resolution				±0.60	%FSR
						±0.85 Note 3	%FSR
Integral linearity error Note 1	ILE	10-bit resolution				±4.0	LSB
						±6.5 Note 3	LSB
Differential linearity error Note 1	DLE	10-bit resolution				±2.0	LSB
						±2.5 Note 3	LSB
Analog input voltage	V_{AIN}	ANI0 to ANI3, ANI16 to ANI22		0		V_{DD}	V
		Internal reference voltage (2.4 V ≤ V_{DD} ≤ 5.5 V, HS (high-speed main) mode)		V_{BGR} Note 4			V
		Temperature sensor output voltage (2.4 V ≤ V_{DD} ≤ 5.5 V, HS (high-speed main) mode)		V_{TMPS25} Note 4			V

Notes 1. Excludes quantization error (±1/2 LSB).

2. This value is indicated as a ratio (%FSR) to the full-scale value.

<R> **3.** When the conversion time is set to 57 μs (min.) and 95 μs (max.).

<R> **4.** Refer to **28.6.2 Temperature sensor/internal reference voltage characteristics**.

Downloaded from Arrow.com.

<R> **(4) When reference voltage (+) = Internal reference voltage (ADREFP1 = 1, ADREFP0 = 0), reference voltage (–) = AV_{REFM} (ADREFM = 1), target pin: ANI0, ANI2, ANI3, and ANI16 to ANI22**

(T_A = –40 to +85°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V, Reference voltage (+) = V_{BGR} Note 3, Reference voltage (–) = AV_{REFM} Note 4 = 0 V, HS (high-speed main) mode)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution	RES			8		bit
Conversion time	tCONV	8-bit resolution	17		39	μs
Zero-scale error Notes 1, 2	EZS	8-bit resolution			±0.60	%FSR
Integral linearity error Note 1	ILE	8-bit resolution			±2.0	LSB
Differential linearity error Note 1	DLE	8-bit resolution			±1.0	LSB
Analog input voltage	VAIN		0		V_{BGR} Note 3	V

Notes 1. Excludes quantization error (±1/2 LSB).

2. This value is indicated as a ratio (%FSR) to the full-scale value.

<R> **3.** Refer to **28.6.2 Temperature sensor/internal reference voltage characteristics**.

<R> **4.** When reference voltage (–) = V_{SS}, the MAX. values are as follows.
Zero-scale error: Add ±0.35%FSR to the MAX. value when reference voltage (–) = AV_{REFM}.
Integral linearity error: Add ±0.5 LSB to the MAX. value when reference voltage (–) = AV_{REFM}.
Differential linearity error: Add ±0.2 LSB to the MAX. value when reference voltage (–) = AV_{REFM}.

Downloaded from Arrow.com.

28.6.2 Temperature sensor/internal reference voltage characteristics

(T_A = –40 to +85°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V, HS (high-speed main) mode

<R>

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Temperature sensor output voltage	V_{TMPS25}	Setting ADS register = 80H, T_A = +25°C		1.05		V
Internal reference voltage	V_{BGR}	Setting ADS register = 81H	1.38	1.45	1.50	V
Temperature coefficient	F_{VTMPS}	Temperature sensor output voltage that depends on the temperature		–3.6		mV/°C
Operation stabilization wait time	t_{AMP}		5			μs

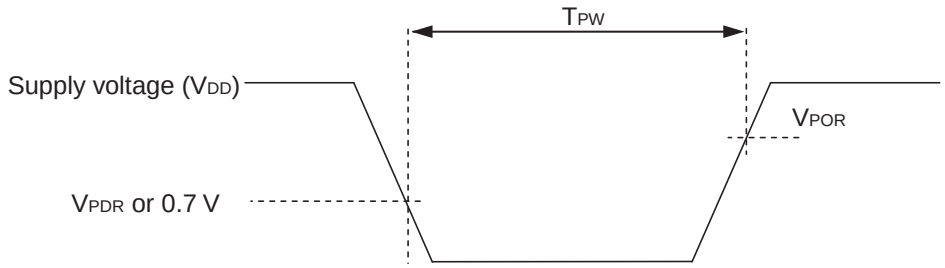
28.6.3 POR circuit characteristics

(T_A = –40 to +85°C, V_{SS} = 0 V)

<R>

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	V_{POR}	Power supply rise time	1.47	1.51	1.55	V
	V_{PDR}	Power supply fall time	1.46	1.50	1.54	V
Minimum pulse width Note	T_{PW}		300			μs

<R> **Note** Minimum time required for a POR reset when V_{DD} exceeds below V_{PDR}. This is also the minimum time required for a POR reset from when V_{DD} exceeds below 0.7 V to when V_{DD} exceeds V_{POR} while STOP mode is entered or the main system clock is stopped through setting bit 0 (HIOSTOP) and bit 7 (MSTOP) in the clock operation status control register (CSC).



Downloaded from Arrow.com.

28.6.4 LVD circuit characteristics

LVD Detection Voltage of Reset Mode and Interrupt Mode

(T_A = –40 to +85°C, $V_{PDR} \leq V_{DD} \leq 5.5$ V, V_{SS} = 0 V)

<R>

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection supply voltage	V_{LVD0}	Power supply rise time	3.98	4.06	4.14	V
		Power supply fall time	3.90	3.98	4.06	V
	V_{LVD1}	Power supply rise time	3.68	3.75	3.82	V
		Power supply fall time	3.60	3.67	3.74	V
	V_{LVD2}	Power supply rise time	3.07	3.13	3.19	V
		Power supply fall time	3.00	3.06	3.12	V
	V_{LVD3}	Power supply rise time	2.96	3.02	3.08	V
		Power supply fall time	2.90	2.96	3.02	V
	V_{LVD4}	Power supply rise time	2.86	2.92	2.97	V
		Power supply fall time	2.80	2.86	2.91	V
	V_{LVD5}	Power supply rise time	2.76	2.81	2.87	V
		Power supply fall time	2.70	2.75	2.81	V
	V_{LVD6}	Power supply rise time	2.66	2.71	2.76	V
		Power supply fall time	2.60	2.65	2.70	V
	V_{LVD7}	Power supply rise time	2.56	2.61	2.66	V
		Power supply fall time	2.50	2.55	2.60	V
	V_{LVD8}	Power supply rise time	2.45	2.50	2.55	V
		Power supply fall time	2.40	2.45	2.50	V
	V_{LVD9}	Power supply rise time	2.05	2.09	2.13	V
		Power supply fall time	2.00	2.04	2.08	V
	V_{LVD10}	Power supply rise time	1.94	1.98	2.02	V
		Power supply fall time	1.90	1.94	1.98	V
	V_{LVD11}	Power supply rise time	1.84	1.88	1.91	V
		Power supply fall time	1.80	1.84	1.87	V
Minimum pulse width	t_{LW}		300			μs
Detection delay time					300	μs

Downloaded from Arrow.com.

<R> **LVD detection voltage of interrupt & reset mode**

(T_A = –40 to +85°C, $V_{PDR} \leq V_{DD} \leq$ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Interrupt and reset mode	V_{LVDB0}	V_{POC2}, V_{POC1}, V_{POC0} = 0, 0, 1, falling reset voltage		1.80	1.84	1.87	V
	V_{LVDB1}	LVIS1, LVIS0 = 1, 0	Rising reset release voltage	1.94	1.98	2.02	V
			Falling interrupt voltage	1.90	1.94	1.98	V
	V_{LVDB2}	LVIS1, LVIS0 = 0, 1	Rising reset release voltage	2.05	2.09	2.13	V
			Falling interrupt voltage	2.00	2.04	2.08	V
	V_{LVDB3}	LVIS1, LVIS0 = 0, 0	Rising reset release voltage	3.07	3.13	3.19	V
			Falling interrupt voltage	3.00	3.06	3.12	V
	V_{LVDC0}	V_{POC2}, V_{POC1}, V_{POC0} = 0, 1, 0, falling reset voltage		2.40	2.45	2.50	V
	V_{LVDC1}	LVIS1, LVIS0 = 1, 0	Rising reset release voltage	2.56	2.61	2.66	V
			Falling interrupt voltage	2.50	2.55	2.60	V
	V_{LVDC2}	LVIS1, LVIS0 = 0, 1	Rising reset release voltage	2.66	2.71	2.76	V
			Falling interrupt voltage	2.60	2.65	2.70	V
	V_{LVDC3}	LVIS1, LVIS0 = 0, 0	Rising reset release voltage	3.68	3.75	3.82	V
			Falling interrupt voltage	3.60	3.67	3.74	V
	V_{LVDD0}	V_{POC2}, V_{POC1}, V_{POC1} = 0, 1, 1, falling reset voltage		2.70	2.75	2.81	V
	V_{LVDD1}	LVIS1, LVIS0 = 1, 0	Rising reset release voltage	2.86	2.92	2.97	V
			Falling interrupt voltage	2.80	2.86	2.91	V
	V_{LVDD2}	LVIS1, LVIS0 = 0, 1	Rising reset release voltage	2.96	3.02	3.08	V
			Falling interrupt voltage	2.90	2.96	3.02	V
	V_{LVDD3}	LVIS1, LVIS0 = 0, 0	Rising reset release voltage	3.98	4.06	4.14	V
			Falling interrupt voltage	3.90	3.98	4.06	V

<R> **28.6.5 Power supply voltage rising slope characteristics**

(T_A = –40 to +85°C, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Power supply voltage rising slope	S_{VDD}				54	V/ms

Caution Make sure to keep the internal reset state by the LVD circuit or an external reset until V_{DD} reaches the operating voltage range shown in 28.4 AC Characteristics.

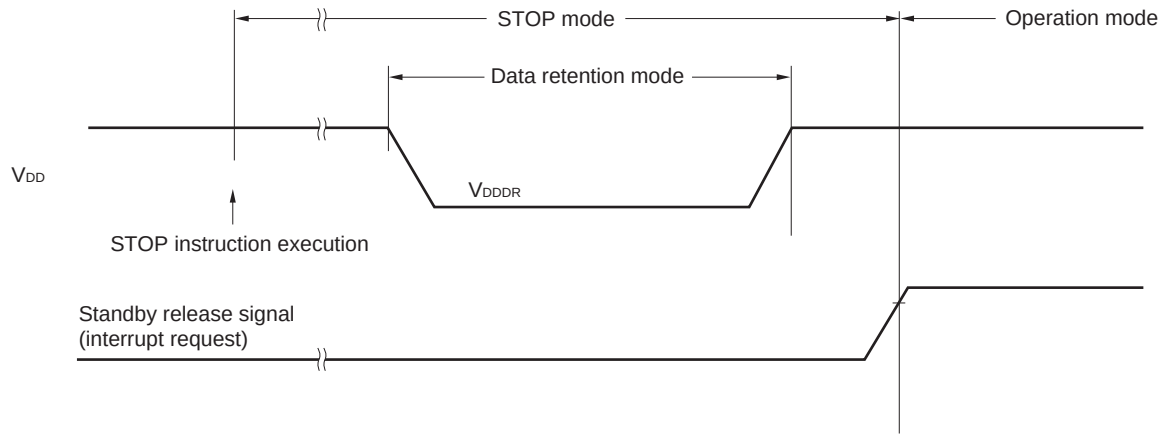
Downloaded from Arrow.com.

28.7 Data Memory STOP Mode Low Supply Voltage Data Retention Characteristics

(T_A = –40 to +85°C, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Data retention supply voltage	V_{DDDR}		1.46 Note		5.5	V

Note The value depends on the POR detection voltage. When the voltage drops, the data is retained before a POR reset is affected, but data is not retained when a POR reset is affected.



28.8 Flash Memory Programming Characteristics

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
System clock frequency	f_{CLK}			1		24	MHz
Code flash memory rewritable times Notes 1, 2, 3	C_{erwr}	Retained for 20 years	T_A = 85°C	1,000			Times
Data flash memory rewritable times Notes 1, 2, 3		Retained for 1 year	T_A = 25°C		1,000,000		
		Retained for 5 years	T_A = 85°C	100,000			
		Retained for 20 years	T_A = 85°C	10,000			

Notes 1. 1 erase + 1 write after the erase is regarded as 1 rewrite. The retaining years are until next rewrite after the rewrite.

2. When using flash memory programmer and Renesas Electronics self programming library

3. These are the characteristics of the flash memory and the results obtained from reliability testing by Renesas Electronics Corporation.

Downloaded from Arrow.com.

28.9 Dedicated Flash Memory Programmer Communication (UART)

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate		During serial programming	115,200		1,000,000	bps

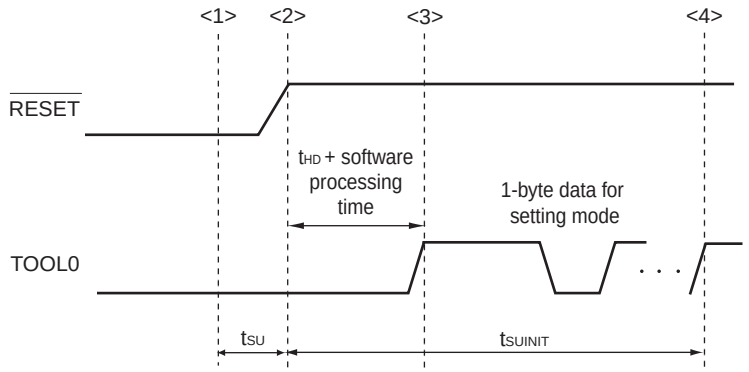
28.10 Timing of Entry to Flash Memory Programming Modes

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

<R>

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Time to complete the communication for the initial setting after the external reset is released	t_{SUINIT}	POR and LVD reset are released before external reset release			100	ms
Time to release the external reset after the TOOL0 pin is set to the low level	t_{SU}	POR and LVD reset are released before external reset release	10			μs
Time to hold the TOOL0 pin at the low level after the external reset is released (excluding the processing time of the firmware to control the flash memory)	t_{HD}	POR and LVD reset are released before external reset release	1			ms

<R>



<1> The low level is input to the TOOL0 pin.

<2> The external reset is released (POR and LVD reset must be released before the external reset is released.).

<3> The TOOL0 pin is set to the high level.

<R> <4> Setting of the flash memory programming mode by UART reception and complete the baud rate setting.

Remark t_{SUINIT}: Communication for the initial setting must be completed within 100 ms after the external reset is released during this period.

t_{SU}: Time to release the external reset after the TOOL0 pin is set to the low level

<R> t_{HD}: Time to hold the TOOL0 pin at the low level after the external reset is released (excluding the processing time of the firmware to control the flash memory)

Downloaded from Arrow.com.

<R>

CHAPTER 29 ELECTRICAL SPECIFICATIONS (G: T_A = –40 to +105°C)

This chapter describes the electrical specifications for the products "G: Industrial applications (T_A = -40 to +105°C)".

Cautions 1. The RL78 microcontrollers have an on-chip debug function, which is provided for development and evaluation. Do not use the on-chip debug function in products designated for mass production, because the guaranteed number of rewritable times of the flash memory may be exceeded when this function is used, and product reliability therefore cannot be guaranteed. Renesas Electronics is not liable for problems occurring when the on-chip debug function is used.

2. The pins mounted depend on the product. Refer to 2.1 Port Functions to 2.2.1 Functions for each product.

3. Please contact Renesas Electronics sales office for derating of operation under T_A = +85°C to +105°C. Derating is the systematic reduction of load for the sake of improved reliability.

There are following differences between the products "G: Industrial applications (T_A = -40 to +105°C)" and the products “A: Consumer applications, and D: Industrial applications”.

Parameter	Application	
	A: Consumer applications, D: Industrial applications	G: Industrial applications
Operating ambient temperature	T_A = -40 to +85°C	T_A = -40 to +105°C
Operating mode Operating voltage range	HS (high-speed main) mode: 2.7 V ≤ V_{DD} ≤ 5.5 V@1 MHz to 24 MHz 2.4 V ≤ V_{DD} ≤ 5.5 V@1 MHz to 16 MHz LS (low-speed main) mode: 1.8 V ≤ V_{DD} ≤ 5.5 V@1 MHz to 8 MHz	HS (high-speed main) mode only: 2.7 V ≤ V_{DD} ≤ 5.5 V@1 MHz to 24 MHz 2.4 V ≤ V_{DD} ≤ 5.5 V@1 MHz to 16 MHz
High-speed on-chip oscillator clock accuracy	R5F102 products, 1.8 V ≤ V_{DD} ≤ 5.5 V: ±1.0%@ T_A = -20 to +85°C ±1.5%@ T_A = -40 to -20°C R5F103 products, 1.8 V ≤ V_{DD} ≤ 5.5 V: ±5.0%@ T_A = -40 to +85°C	R5F102 products, 2.4 V ≤ V_{DD} ≤ 5.5 V: ±2.0%@ T_A = +85 to +105°C ±1.0%@ T_A = -20 to +85°C ±1.5%@ T_A = -40 to -20°C
Serial array unit	UART CSI: f_{CLK}/2 (supporting 12 Mbps), f_{CLK}/4 Simplified I^2C communication	UART CSI: f_{CLK}/4 Simplified I^2C communication
Voltage detector	Rise detection voltage: 1.88 V to 4.06 V (12 levels) Fall detection voltage: 1.84 V to 3.98 V (12 levels)	Rise detection voltage: 2.61 V to 4.06 V (8 levels) Fall detection voltage: 2.55 V to 3.98 V (8 levels)

Remark The electrical characteristics of the products G: Industrial applications (T_A = -40 to +105°C) are different from those of the products “A: Consumer applications, and D: Industrial applications”. For details, refer to **29.1** to **29.10**.

Downloaded from Arrow.com.

29.1 Absolute Maximum Ratings

Absolute Maximum Ratings (T_A = 25°C)

Parameter	Symbols	Conditions		Ratings	Unit
Supply Voltage	V_{DD}			–0.5 to + 6.5	V
REGC terminal input voltage Note1	V_{IREGC}	REGC		–0.3 to +2.8 and –0.3 to V_{DD} + 0.3 Note 2	V
Input Voltage	V_{I1}	Other than P60, P61		–0.3 to V_{DD} + 0.3 Note 3	V
	V_{I2}	P60, P61 (N-ch open drain)		–0.3 to 6.5	V
Output Voltage	V_O			–0.3 to V_{DD} + 0.3 Note 3	V
Analog input voltage	V_{AI}	20, 24-pin products: ANI0 to ANI3, ANI16 to ANI22 30-pin products: ANI0 to ANI3, ANI16 to ANI19		–0.3 to V_{DD} + 0.3 and –0.3 to AVREF(+)+0.3 Notes 3, 4	V
Output current, high	I_{OH1}	Per pin	Other than P20 to P23	–40	mA
		Total of all pins	All the terminals other than P20 to P23	–170	mA
			20-, 24-pin products: P40 to P42 30-pin products: P00, P01, P40, P120	–70	mA
			20-, 24-pin products: P00 to P03 Note 5, P10 to P14 30-pin products: P10 to P17, P30, P31, P50, P51, P147	–100	mA
	I_{OH2}	Per pin	P20 to P23	–0.5	mA
		Total of all pins		–2	mA
Output current, low	I_{OL1}	Per pin	Other than P20 to P23	40	mA
		Total of all pins	All the terminals other than P20 to P23	170	mA
			20-, 24-pin products: P40 to P42 30-pin products: P00, P01, P40, P120	70	mA
			20-, 24-pin products: P00 to P03 Note 5, P10 to P14, P60, P61 30-pin products: P10 to P17, P30, P31, P50, P51, P60, P61, P147	100	mA
	I_{OL2}	Per pin	P20 to P23	1	mA
		Total of all pins		5	mA
Operating ambient temperature	T_A			–40 to +105	°C
Storage temperature	T_{stg}			–65 to +150	°C

Notes 1. 30-pin product only.

2. Connect the REGC pin to V_{SS} via a capacitor (0.47 to 1 μF). This value determines the absolute maximum rating of the REGC pin. Do not use it with voltage applied.

3. Must be 6.5 V or lower.

4. Do not exceed $AV_{REF}(+)$ + 0.3 V in case of A/D conversion target pin.

5. 24-pin products only.

Caution Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.

Remarks 1. Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

2. $AV_{REF}(+)$: + side reference voltage of the A/D converter.

3. V_{SS} : Reference voltage

Downloaded from Arrow.com.

29.2 Oscillator Characteristics

29.2.1 X1 oscillator characteristics

(T_A = –40 to +105°C, 2.4 V ≤ V_{DD} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Resonator	Conditions	MIN.	TYP.	MAX.	Unit
X1 clock oscillation frequency (f_X)[Note]	Ceramic resonator / crystal oscillator	2.7 V ≤ V_{DD} ≤ 5.5 V	1.0		20.0	MHz
		2.4 V ≤ V_{DD} < 2.7 V	1.0		8.0	

Note Indicates only permissible oscillator frequency ranges. Refer to AC Characteristics for instruction execution time. Request evaluation by the manufacturer of the oscillator circuit mounted on a board to check the oscillator characteristics.

Caution Since the CPU is started by the high-speed on-chip oscillator clock after a reset release, check the X1 clock oscillation stabilization time using the oscillation stabilization time counter status register (OSTC) by the user. Determine the oscillation stabilization time of the OSTC register and the oscillation stabilization time select register (OSTS) after sufficiently evaluating the oscillation stabilization time with the resonator to be used.

Remark When using the X1 oscillator, refer to **5.4 System Clock Oscillator**.

29.2.2 On-chip oscillator characteristics

(T_A = –40 to +105°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Oscillators	Parameters	Conditions		MIN.	TYP.	MAX.	Unit
High-speed on-chip oscillator clock frequency [Notes 1, 2]	f_{IH}			1		24	MHz
High-speed on-chip oscillator clock frequency accuracy		R5F102 products	T_A = –20 to +85°C	-1.0		+1.0	%
			T_A = –40 to –20°C	-1.5		+1.5	%
			T_A = +85 to +105°C	-2.0		+2.0	%
Low-speed on-chip oscillator clock frequency	f_{IL}				15		kHz
Low-speed on-chip oscillator clock frequency accuracy				-15		+15	%

Notes 1. High-speed on-chip oscillator frequency is selected by bits 0 to 3 of option byte (000C2H) and bits 0 to 2 of HOCODIV register.

2. This only indicates the oscillator characteristics. Refer to AC Characteristics for instruction execution time.

Downloaded from Arrow.com.

29.3 DC Characteristics

29.3.1 Pin characteristics

(T_A = –40 to +105°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V) **(1/4)**

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output current, high[Note 1]	I_{OH1}	20-, 24-pin products: Per pin for P00 to P03[Note 4], P10 to P14, P40 to P42 30-pin products: Per pin for P00, P01, P10 to P17, P30, P31, P40, P50, P51, P120, P147				–3.0 [Note 2]	mA
		20-, 24-pin products: Total of P40 to P42 30-pin products: Total of P00, P01, P40, P120 (When duty ≤ 70% [Note 3])	4.0 V ≤ V_{DD} ≤ 5.5 V			–9.0	mA
			2.7 V ≤ V_{DD} < 4.0 V			–6.0	mA
			2.4 V ≤ V_{DD} < 2.7 V			–4.5	mA
		20-, 24-pin products: Total of P00 to P03[Note 4], P10 to P14 30-pin products: Total of P10 to P17, P30, P31, P50, P51, P147 (When duty ≤ 70% [Note 3])	4.0 V ≤ V_{DD} ≤ 5.5 V			–27.0	mA
			2.7 V ≤ V_{DD} < 4.0 V			–18.0	mA
			2.4 V ≤ V_{DD} < 2.7 V			–10.0	mA
		Total of all pins (When duty ≤ 70%[Note 3])				–36.0	mA
	I_{OH2}	Per pin for P20 to P23				–0.1	mA
		Total of all pins				–0.4	mA

Notes 1. value of current at which the device operation is guaranteed even if the current flows from the V_{DD} pin to an output pin.

2. However, do not exceed the total current value.

3. The output current value under conditions where the duty factor ≤ 70%.
If duty factor > 70%: The output current value can be calculated with the following expression (where n represents the duty factor as a percentage).

- Total output current of pins = (I_{OH} × 0.7)/(n × 0.01)

<Example> Where n = 80% and I_{OH} = –10.0 mA

Total output current of pins = (–10.0 × 0.7)/(80 × 0.01) ≅ –8.7 mA

However, the current that is allowed to flow into one pin does not vary depending on the duty factor. A current higher than the absolute maximum rating must not flow into one pin.

4. 24-pin products only.

Caution P10 to P12 and P41 for 20-pin products, P01, P10 to P12, and P41 for 24-pin products, and P00, P10 to P15, P17, and P50 for 30-pin products do not output high level in N-ch open-drain mode.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

Downloaded from Arrow.com.

(T_A = –40 to +105°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V) (2/4)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output current, low Note 1	I_{OL1}	20-, 24-pin products: Per pin for P00 to P03 Note 4, P10 to P14, P40 to P42 30-pin products: Per pin for P00, P01, P10 to P17, P30, P31, P40, P50, P51, P120, P147				8.5 Note 2	mA
		Per pin for P60, P61				15.0 Note 2	mA
		20-, 24-pin products: Total of P40 to P42 30-pin products: Total of P00, P01, P40, P120 (When duty ≤ 70% Note 3)	4.0 V ≤ V_{DD} ≤ 5.5 V			25.5	mA
			2.7 V ≤ V_{DD} < 4.0 V			9.0	mA
			2.4 V ≤ V_{DD} < 2.7 V			1.8	mA
		20-, 24-pin products: Total of P00 to P03 Note 4, P10 to P14, P60, P61 30-pin products: Total of P10 to P17, P30, P31, P50, P51, P60, P61, P147 (When duty ≤ 70% Note 3)	4.0 V ≤ V_{DD} ≤ 5.5 V			40.0	mA
			2.7 V ≤ V_{DD} < 4.0 V			27.0	mA
			2.4 V ≤ V_{DD} < 2.7 V			5.4	mA
		Total of all pins (When duty ≤ 70% Note 3)				65.5	mA
	I_{OL2}	Per pin for P20 to P23				0.4	mA
		Total of all pins				1.6	mA

Notes 1. Value of current at which the device operation is guaranteed even if the current flows from an output pin to the V_{SS} pin.

2. However, do not exceed the total current value.

3. The output current value under conditions where the duty factor ≤ 70%.

If duty factor > 70%: The output current value can be calculated with the following expression (where n represents the duty factor as a percentage).

- Total output current of pins = (I_{OL} × 0.7)/(n × 0.01)

<Example> Where n = 80% and I_{OL} = 10.0 mA

Total output current of pins = (10.0 × 0.7)/(80 × 0.01) ≅ 8.7 mA

However, the current that is allowed to flow into one pin does not vary depending on the duty factor. A current higher than the absolute maximum rating must not flow into one pin.

4. 24-pin products only.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

Downloaded from Arrow.com.

(T_A = –40 to +105°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V) (3/4)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input voltage, high	V_{IH1}	Normal input buffer 20-, 24-pin products: P00 to P03[Note 2], P10 to P14, P40 to P42 30-pin products: P00, P01, P10 to P17, P30, P31, P40, P50, P51, P120, P147		0.8V_{DD}		V_{DD}	V
	V_{IH2}	TTL input buffer 20-, 24-pin products: P10, P11 30-pin products: P01, P10, P11, P13 to P17	4.0 V ≤ V_{DD} ≤ 5.5 V	2.2		V_{DD}	V
			3.3 V ≤ V_{DD} < 4.0 V	2.0		V_{DD}	V
			2.4 V ≤ V_{DD} < 3.3 V	1.5		V_{DD}	V
	V_{IH3}	Normal input buffer P20 to P23		0.7V_{DD}		V_{DD}	V
	V_{IH4}	P60, P61		0.7V_{DD}		6.0	V
	V_{IH5}	P121, P122, P125[Note 1], P137, EXCLK, $\overline{\text{RESET}}$		0.8V_{DD}		V_{DD}	V
Input voltage, low	V_{IL1}	Normal input buffer 20-, 24-pin products: P00 to P03[Note 2], P10 to P14, P40 to P42 30-pin products: P00, P01, P10 to P17, P30, P31, P40, P50, P51, P120, P147		0		0.2V_{DD}	V
	V_{IL2}	TTL input buffer 20-, 24-pin products: P10, P11 30-pin products: P01, P10, P11, P13 to P17	4.0 V ≤ V_{DD} ≤ 5.5 V	0		0.8	V
			3.3 V ≤ V_{DD} < 4.0 V	0		0.5	V
			2.4 V ≤ V_{DD} < 3.3 V	0		0.32	V
	V_{IL3}	P20 to P23		0		0.3V_{DD}	V
	V_{IL4}	P60, P61		0		0.3V_{DD}	V
	V_{IL5}	P121, P122, P125[Note 1], P137, EXCLK, $\overline{\text{RESET}}$		0		0.2V_{DD}	V
Output voltage, high	V_{OH1}	20-, 24-pin products: P00 to P03[Note 2], P10 to P14, P40 to P42 30-pin products: P00, P01, P10 to P17, P30, P31, P40, P50, P51, P120, P147	4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OH1} = –3.0 mA	V_{DD}–0.7			V
			2.7 V ≤ V_{DD} ≤ 5.5 V, I_{OH1} = –2.0 mA	V_{DD}–0.6			V
			2.4 V ≤ V_{DD} ≤ 5.5 V, I_{OH1} = –1.5 mA	V_{DD}–0.5			V
	V_{OH2}	P20 to P23	I_{OH2} = –100 μA	V_{DD}–0.5			V

Notes 1. 20, 24-pin products only.

2. 24-pin products only.

Caution The maximum value of V_{IH} of pins P10 to P12 and P41 for 20-pin products, P01, P10 to P12, and P41 for 24-pin products, and P00, P10 to P15, P17, and P50 for 30-pin products is V_{DD} even in N-ch open-drain mode.
High level is not output in the N-ch open-drain mode.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

Downloaded from Arrow.com.

(T_A = –40 to +105°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V) **(4/4)**

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output voltage, low	V_{OL1}	20-, 24-pin products: P00 to P03[Note], P10 to P14, P40 to P42 30-pin products: P00, P01, P10 to P17, P30, P31, P40, P50, P51, P120, P147	4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 8.5 mA			0.7	V
			2.7 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 3.0 mA			0.6	V
			2.7 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 1.5 mA			0.4	V
			2.4 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 0.6 mA			0.4	V
	V_{OL2}	P20 to P23	I_{OL2} = 400 μA			0.4	V
	V_{OL3}	P60, P61	4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 15.0 mA			2.0	V
			4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 5.0 mA			0.4	V
			2.7 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 3.0 mA			0.4	V
			2.4 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 2.0 mA			0.4	V
Input leakage current, high	I_{LIH1}	Other than P121, P122	V_I = V_{DD}			1	μA
	I_{LIH2}	P121, P122 (X1, X2/EXCLK)	V_I = V_{DD} Input port or external clock input			1	μA
			V_I = V_{DD} When resonator connected			10	μA
Input leakage current, low	I_{LIL1}	Other than P121, P122	V_I = V_{SS}			–1	μA
	I_{LIL2}	P121, P122 (X1, X2/EXCLK)	V_I = V_{SS} Input port or external clock input			–1	μA
			V_I = V_{SS} When resonator connected			–10	μA
On-chip pull-up resistance	R_U	20-, 24-pin products: P00 to P03[Note], P10 to P14, P40 to P42, P125, $\overline{\text{RESET}}$ 30-pin products: P00, P01, P10 to P17, P30, P31, P40, P50, P51, P120, P147	V_I = V_{SS}, input port	10	20	100	kΩ

Note 24-pin products only.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

Downloaded from Arrow.com.

29.3.2 Supply current characteristics

(1) 20-, 24-pin products

(T_A = –40 to +105°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V) **(1/2)**

Parameter	Symbol	Conditions					MIN.	TYP.	MAX.	Unit
Supply current[Note 1]	I_{DD1}	Operating mode	HS (High-speed main) mode[Note 4]	f_{IH} = 24 MHz[Note 3]	Basic operation	V_{DD} = 5.0 V		1.5		mA
						V_{DD} = 3.0 V		1.5		
					Normal operation	V_{DD} = 5.0 V		3.3	5.3	mA
						V_{DD} = 3.0 V		3.3	5.3	
				f_{IH} = 16 MHz[Note 3]		V_{DD} = 5.0 V		2.5	3.9	mA
						V_{DD} = 3.0 V		2.5	3.9	
				f_{MX} = 20 MHz[Note 2], V_{DD} = 5.0 V		Square wave input		2.8	4.7	mA
						Resonator connection		3.0	4.8	
				f_{MX} = 20 MHz[Note 2], V_{DD} = 3.0 V		Square wave input		2.8	4.7	mA
						Resonator connection		3.0	4.8	
				f_{MX} = 10 MHz[Note 2], V_{DD} = 5.0 V		Square wave input		1.8	2.8	mA
						Resonator connection		1.8	2.8	
				f_{MX} = 10 MHz[Note 2], V_{DD} = 3.0 V		Square wave input		1.8	2.8	mA
						Resonator connection		1.8	2.8	

Notes 1. Total current flowing into V_{DD}, including the input leakage current flowing when the level of the input pin is fixed to V_{DD} or V_{SS}. The values below the MAX. column include the peripheral operation current. However, not including the current flowing into the A/D converter, LVD circuit, I/O port, and on-chip pull-up/pull-down resistors and the current flowing during data flash rewrite.

2. When high-speed on-chip oscillator clock is stopped.

3. When high-speed system clock is stopped

4. Relationship between operation voltage width, operation frequency of CPU and operation mode is as follows.

HS(High speed main) mode: V_{DD} = 2.7 V to 5.5 V @1 MHz to 24 MHz
V_{DD} = 2.4 V to 5.5 V @1 MHz to 16 MHz

Remarks 1. f_{MX}: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)

2. f_{IH}: high-speed on-chip oscillator clock frequency

3. Temperature condition of the TYP. value is T_A = 25°C.

Downloaded from Arrow.com.

(1) 20-, 24-pin products

(T_A = –40 to +105°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V) **(2/2)**

Parameter	Symbol	Conditions				MIN.	TYP.	MAX.	Unit
Supply current[Note 1]	I_{DD2}[Note 2]	HALT mode	HS (High-speed main) mode[Note 6]	f_{IH} = 24 MHz[Note 4]	V_{DD} = 5.0 V		440	2230	μA
					V_{DD} = 3.0 V		440	2230	
				f_{IH} = 16 MHz[Note 4]	V_{DD} = 5.0 V		400	1650	μA
					V_{DD} = 3.0 V		400	1650	
				f_{MX} = 20 MHz[Note 3], V_{DD} = 5.0 V	Square wave input		280	1900	μA
					Resonator connection		450	2000	
				f_{MX} = 20 MHz[Note 3], V_{DD} = 3.0 V	Square wave input		280	1900	μA
					Resonator connection		450	2000	
				f_{MX} = 10 MHz[Note 3], V_{DD} = 5.0 V	Square wave input		190	1010	μA
					Resonator connection		260	1090	
				f_{MX} = 10 MHz[Note 3], V_{DD} = 3.0 V	Square wave input		190	1010	μA
					Resonator connection		260	1090	
	I_{DD3}[Note 5]	STOP mode	T_A = –40°C				0.19	0.50	μA
			T_A = +25°C				0.24	0.50	
			T_A = +50°C				0.32	0.80	
			T_A = +70°C				0.48	1.20	
			T_A = +85°C				0.74	2.20	
			T_A = +105°C				1.50	10.20	

Notes 1. Total current flowing into V_{DD}, including the input leakage current flowing when the level of the input pin is fixed to V_{DD} or V_{SS}. The values below the MAX. column include the peripheral operation current. However, not including the current flowing into the A/D converter, LVD circuit, I/O port, and on-chip pull-up/pull-down resistors and the current flowing during data flash rewrite.

2. During HALT instruction execution by flash memory.

3. When high-speed on-chip oscillator clock is stopped.

4. When high-speed system clock is stopped.

5. Not including the current flowing into the 12-bit interval timer and watchdog timer.

6. Relationship between operation voltage width, operation frequency of CPU and operation mode is as follows.

HS (High speed main) mode: V_{DD} = 2.7 V to 5.5 V @1 MHz to 24 MHz
V_{DD} = 2.4 V to 5.5 V @1 MHz to 16 MHz

Remarks 1. f_{MX}: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)

2. f_{IH}: high-speed on-chip oscillator clock frequency

3. Except temperature condition of the TYP. value is T_A = 25°C, other than STOP mode

Downloaded from Arrow.com.

(2) 30-pin products

(T_A = –40 to +105°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V) **(1/2)**

Parameter	Symbol	Conditions					MIN.	TYP.	MAX.	Unit
Supply current[Note 1]	I_{DD1}	Operating mode	HS (High-speed main) mode[Note 4]	f_{IH} = 24 MHz[Note 3]	Basic operation	V_{DD} = 5.0 V		1.5		mA
						V_{DD} = 3.0 V		1.5		
					Normal operation	V_{DD} = 5.0 V		3.7	5.8	mA
						V_{DD} = 3.0 V		3.7	5.8	
				f_{IH} = 16 MHz[Note 3]		V_{DD} = 5.0 V		2.7	4.2	mA
						V_{DD} = 3.0 V		2.7	4.2	
				f_{MX} = 20 MHz[Note 2], V_{DD} = 5.0 V		Square wave input		3.0	4.9	mA
						Resonator connection		3.2	5.0	
				f_{MX} = 20 MHz[Note 2], V_{DD} = 3.0 V		Square wave input		3.0	4.9	mA
						Resonator connection		3.2	5.0	
				f_{MX} = 10 MHz[Note 2], V_{DD} = 5.0 V		Square wave input		1.9	2.9	mA
						Resonator connection		1.9	2.9	
				f_{MX} = 10 MHz[Note 2], V_{DD} = 3.0 V		Square wave input		1.9	2.9	mA
						Resonator connection		1.9	2.9	

Notes 1. Total current flowing into V_{DD}, including the input leakage current flowing when the level of the input pin is fixed to V_{DD} or V_{SS}. The values below the MAX. column include the peripheral operation current. However, not including the current flowing into the A/D converter, LVD circuit, I/O port, and on-chip pull-up/pull-down resistors and the current flowing during data flash rewrite.

2. When high-speed on-chip oscillator clock is stopped.

3. When high-speed system clock is stopped

4. Relationship between operation voltage width, operation frequency of CPU and operation mode is as follows.

HS(High speed main) mode: V_{DD} = 2.7 V to 5.5 V @1 MHz to 24 MHz
V_{DD} = 2.4 V to 5.5 V @1 MHz to 16 MHz

Remarks 1. f_{MX}: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)

2. f_{IH}: high-speed on-chip oscillator clock frequency

3. Temperature condition of the TYP. value is T_A = 25°C.

Downloaded from Arrow.com.

(2) 30-pin products

(T_A = –40 to +105°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V) **(2/2)**

Parameter	Symbol	Conditions				MIN.	TYP.	MAX.	Unit
Supply current Note 1	I_{DD2} Note 2	HALT mode	HS (High-speed main) mode Note 6	f_{IH} = 24 MHz Note 4	V_{DD} = 5.0 V		440	2300	μA
					V_{DD} = 3.0 V		440	2300	
				f_{IH} = 16 MHz Note 4	V_{DD} = 5.0 V		400	1700	μA
					V_{DD} = 3.0 V		400	1700	
				f_{MX} = 20 MHz Note 3, V_{DD} = 5.0 V	Square wave input		280	1900	μA
					Resonator connection		450	2000	
				f_{MX} = 20 MHz Note 3, V_{DD} = 3.0 V	Square wave input		280	1900	μA
					Resonator connection		450	2000	
				f_{MX} = 10 MHz Note 3, V_{DD} = 5.0 V	Square wave input		190	1020	μA
					Resonator connection		260	1100	
				f_{MX} = 10 MHz Note 3, V_{DD} = 3.0 V	Square wave input		190	1020	μA
					Resonator connection		260	1100	
	I_{DD3} Note 5	STOP mode	T_A = –40°C				0.18	0.50	μA
			T_A = +25°C				0.23	0.50	
			T_A = +50°C				0.30	1.10	
			T_A = +70°C				0.46	1.90	
			T_A = +85°C				0.75	3.30	
			T_A = +105°C				2.94	15.30	

Notes 1. Total current flowing into V_{DD}, including the input leakage current flowing when the level of the input pin is fixed to V_{DD} or V_{SS}. The values below the MAX. column include the peripheral operation current. However, not including the current flowing into the A/D converter, LVD circuit, I/O port, and on-chip pull-up/pull-down resistors and the current flowing during data flash rewrite.

2. During HALT instruction execution by flash memory.

3. When high-speed on-chip oscillator clock is stopped.

4. When high-speed system clock is stopped.

5. Not including the current flowing into the 12-bit interval timer and watchdog timer.

6. Relationship between operation voltage width, operation frequency of CPU and operation mode is as follows.

HS (High speed main) mode: V_{DD} = 2.7 V to 5.5 V @1 MHz to 24 MHz
V_{DD} = 2.4 V to 5.5 V @1 MHz to 16 MHz

Remarks 1. f_{MX}: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)

2. f_{IH}: high-speed on-chip oscillator clock frequency

3. Except STOP mode, temperature condition of the TYP. value is T_A = 25°C.

Downloaded from Arrow.com.

(3) Peripheral functions (Common to all products)

(T_A = –40 to +105°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Low-speed onchip oscillator operating current	I_{FIL} Note 1				0.20		μA
12-bit interval timer operating current	I_{TMKA} Notes 1, 2, 3				0.02		μA
Watchdog timer operating current	I_{WDT} Notes 1, 2, 4	f_{IL} = 15 kHz			0.22		μA
A/D converter operating current	I_{ADC} Notes 1, 5	When conversion at maximum speed	Normal mode, AV_{REFP} = V_{DD} = 5.0 V		1.30	1.70	mA
			Low voltage mode, AV_{REFP} = V_{DD} = 3.0 V		0.50	0.70	mA
A/D converter reference voltage operating current	I_{ADREF} Note 1				75.0		μA
Temperature sensor operating current	I_{TMPS} Note 1				75.0		μA
LVD operating current	I_{LVD} Notes 1, 6				0.08		μA
Self-programming operating current	I_{FSP} Notes 1, 8				2.00	12.20	mA
BGO operating current	I_{BGO} Notes 1, 7				2.00	12.20	mA
SNOOZE operating current	I_{SNOZ} Note 1	ADC operation	The mode is performed Note 9		0.50	1.10	mA
			The A/D conversion operations are performed, Low voltage mode, AV_{REFP} = V_{DD} = 3.0 V		1.20	2.04	mA
		CSI/UART operation			0.70	1.54	mA

Notes 1. Current flowing to the V_{DD}.

2. When high speed on-chip oscillator and high-speed system clock are stopped.

3. Current flowing only to the 12-bit interval timer (excluding the operating current of the low-speed on-chip oscillator). The current value of the RL78 microcontrollers is the sum of I_{DD1}, I_{DD2} or I_{DD3}, and I_{FIL} and I_{TMKA} when the 12-bit interval timer operates.

4. Current flowing only to the watchdog timer (including the operating current of the low-speed on-chip oscillator). The current value of the RL78 microcontrollers is the sum of I_{DD1}, I_{DD2} or I_{DD3} and I_{WDT} when the watchdog timer operates.

5. Current flowing only to the A/D converter. The current value of the RL78 microcontrollers is the sum of I_{DD1} or I_{DD2} and I_{ADC} when the A/D converter operates in an operation mode or the HALT mode.

6. Current flowing only to the LVD circuit. The current value of the RL78 microcontrollers is the sum of I_{DD1}, I_{DD2} or I_{DD3} and I_{LVD} when the LVD circuit operates.

7. Current flowing only during data flash rewrite.

8. Current flowing only during self programming.

9. For shift time to the SNOOZE mode, see **17.3.3 SNOOZE mode**.

Remarks 1. f_{IL}: Low-speed on-chip oscillator clock frequency

2. Temperature condition of the TYP. value is T_A = 25°C

Downloaded from Arrow.com.

29.4 AC Characteristics

(T_A = –40 to +105°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

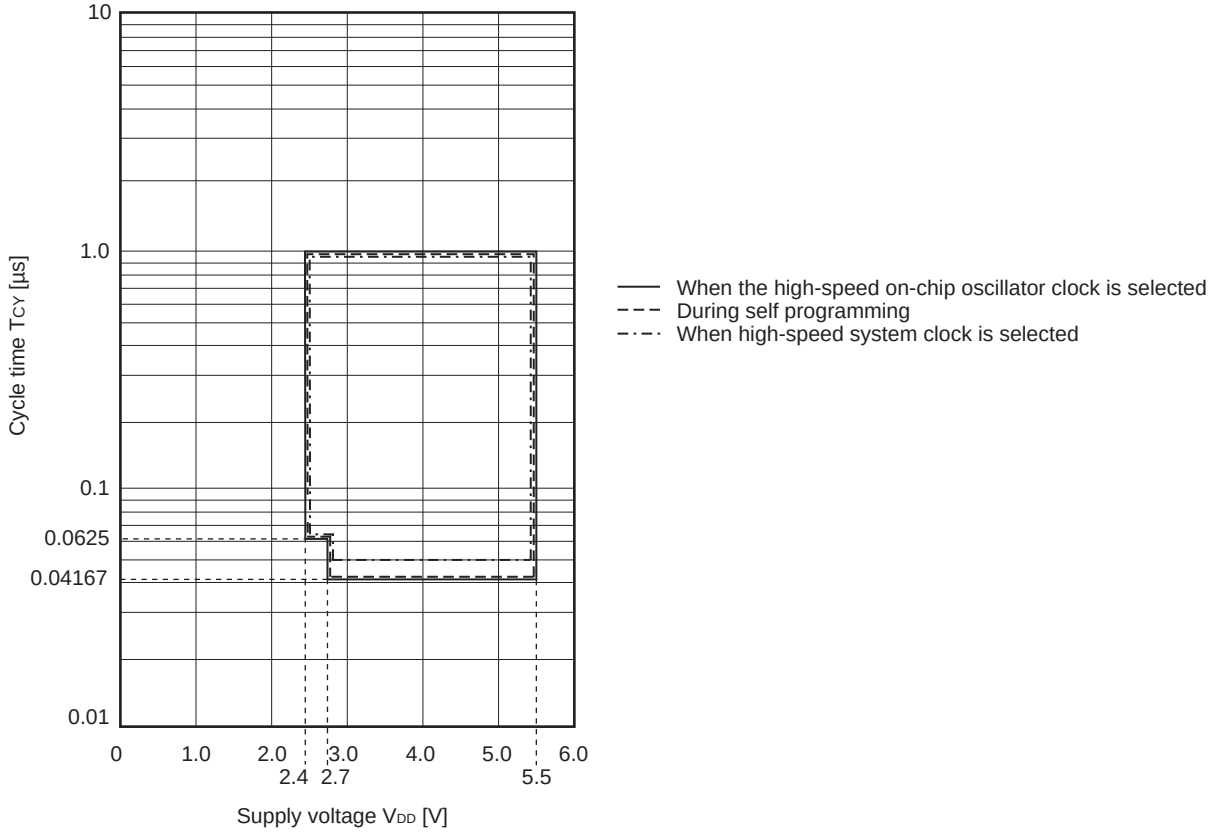
Items	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Instruction cycle (minimum instruction execution time)	T_{CY}	Main system clock (f_{MAIN}) operation	HS (High-speed main) mode	2.7 V ≤ V_{DD} ≤ 5.5 V	0.04167		1	μs
				2.4 V ≤ V_{DD} < 2.7 V	0.0625		1	μs
		During self programming	HS (High-speed main) mode	2.7 V ≤ V_{DD} ≤ 5.5 V	0.04167		1	μs
				2.4 V ≤ V_{DD} < 2.7 V	0.0625		1	μs
External main system clock frequency	f_{EX}	2.7 V ≤ V_{DD} ≤ 5.5 V			1.0		20.0	MHz
		2.4 V ≤ V_{DD} < 2.7 V			1.0		16.0	MHz
External main system clock input high-level width, low-level width	t_{EXH}, t_{EXL}	2.7 V ≤ V_{DD} ≤ 5.5 V			24			ns
		2.4 V ≤ V_{DD} < 2.7 V			30			ns
TI00 to TI07 input high-level width, low-level width	t_{TIH}, t_{TIL}				1/f_{MCK} + 10			ns
TO00 to TO07 output frequency	f_{TO}	4.0 V ≤ V_{DD} ≤ 5.5 V					12	MHz
		2.7 V ≤ V_{DD} < 4.0 V					8	MHz
		2.4 V ≤ V_{DD} < 2.7 V					4	MHz
PCLBUZ0, or PCLBUZ1 output frequency	f_{PCL}	4.0 V ≤ V_{DD} ≤ 5.5 V					16	MHz
		2.7 V ≤ V_{DD} < 4.0 V					8	MHz
		2.4 V ≤ V_{DD} < 2.7 V					4	MHz
INTP0 to INTP5 input high-level width, low-level width	t_{INTH}, t_{INTL}				1			μs
KR0 to KR9 input available width	t_{KR}				250			ns
$\overline{\text{RESET}}$ low-level width	t_{RSL}				10			μs

Remark f_{MCK}: Timer array unit operation clock frequency
(Operation clock to be set by the timer clock select register 0 (TPS0) and the CKS0n bit of timer mode register 0n (TMR0n). n: Channel number (n = 0 to 7))

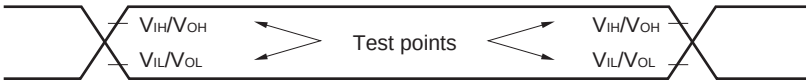
Downloaded from Arrow.com.

Minimum Instruction Execution Time during Main System Clock Operation

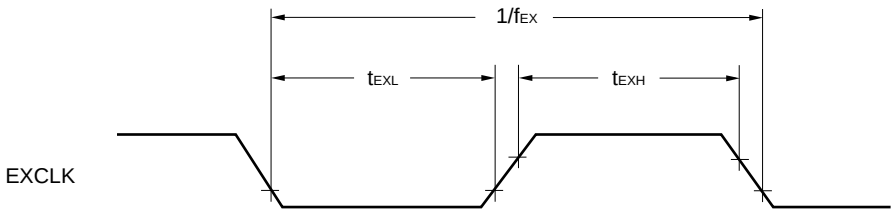
T_{CY} vs V_{DD} (HS (high-speed main) mode)



AC Timing Test Point

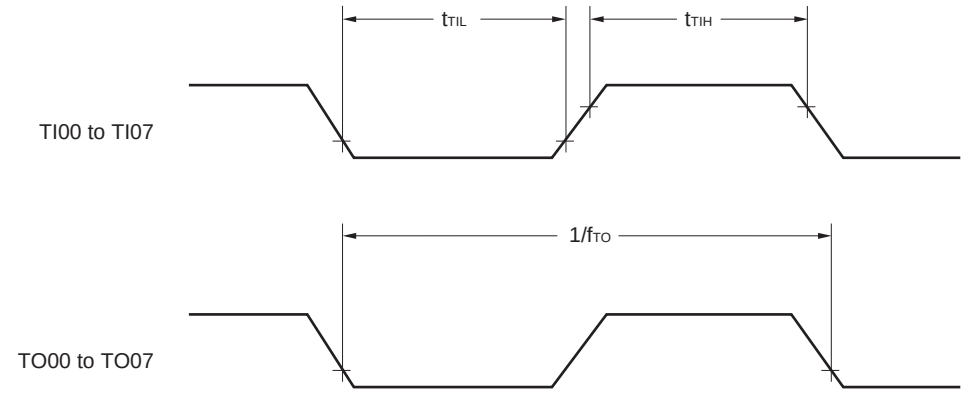


External Main System Clock Timing

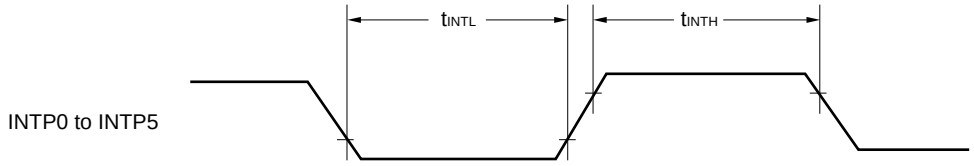


Downloaded from Arrow.com.

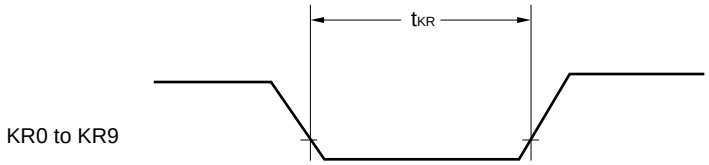
TI/TO Timing



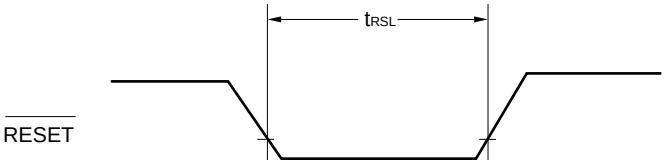
Interrupt Request Input Timing



Key Interrupt Input Timing



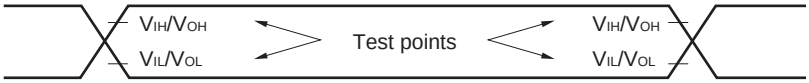
$\overline{\text{RESET}}$ Input Timing



Downloaded from Arrow.com.

29.5 Peripheral Functions Characteristics

AC Timing Test Point



29.5.1 Serial array unit

(1) During communication at same potential (UART mode)

(T_A = –40 to +105°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

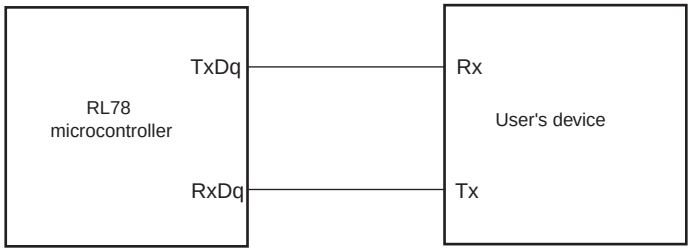
Parameter	Symbol	Conditions	HS (high-speed main) Mode		Unit
			MIN.	MAX.	
Transfer rate Note 1				f_{MCK}/12	bps
		Theoretical value of the maximum transfer rate $f_{CLK} = f_{MCK}$ Note2		2.0	Mbps

Notes 1. Transfer rate in the SNOOZE mode is 4800 bps only.

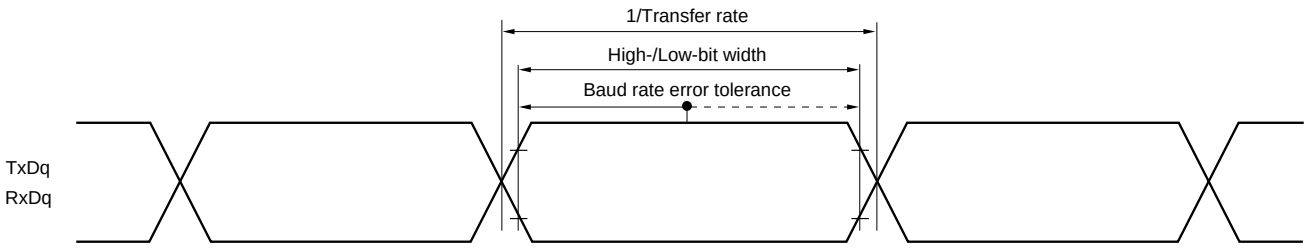
2. The maximum operating frequencies of the CPU/peripheral hardware clock (f_{CLK}) are:
HS (high-speed main) mode: 24 MHz (2.7 V ≤ V_{DD} ≤ 5.5 V)
16 MHz (2.4 V ≤ V_{DD} ≤ 5.5 V)

Caution Select the normal input buffer for the RxDq pin and the normal output mode for the TxDq pin by using port input mode register g (PIMg) and port output mode register g (POMg).

UART mode connection diagram (during communication at same potential)



UART mode bit width (during communication at same potential) (reference)



Remarks 1. q: UART number (q = 0 to 2), g: PIM, POM number (g = 0, 1)

2. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the serial clock select register m (SPSm) and the CKSmn bit of serial mode register mn (SMRmn).
m: Unit number, n: Channel number (mn = 00 to 03, 10, 11))

Downloaded from Arrow.com.

(2) During communication at same potential (CSI mode) (master mode, SCKp... internal clock output)

(T_A = –40 to +105°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		HS (high-speed main) Mode		Unit
				MIN.	MAX.	
SCKp cycle time	t_{KCY1}	$t_{KCY1} \geq 4/f_{CLK}$	2.7 V ≤ V_{DD} ≤ 5.5 V	334		ns
			2.4 V ≤ V_{DD} ≤ 5.5 V	500		ns
SCKp high-/low-level width	t_{KH1}, t_{KL1}	4.0 V ≤ V_{DD} ≤ 5.5 V		t_{KCY1}/2–24		ns
		2.7 V ≤ V_{DD} ≤ 5.5 V		t_{KCY1}/2–36		ns
		2.4 V ≤ V_{DD} ≤ 5.5 V		t_{KCY1}/2–76		ns
SIp setup time (to SCKp↑) Note 1	t_{SIK1}	4.0 V ≤ V_{DD} ≤ 5.5 V		66		ns
		2.7 V ≤ V_{DD} ≤ 5.5 V		66		ns
		2.4 V ≤ V_{DD} ≤ 5.5 V		113		ns
SIp hold time (from SCKp↑) Note 2	t_{KSI1}			38		ns
Delay time from SCKp↓ to SOp output Note 3	t_{KSO1}	C = 30 pF Note4			50	ns

Notes 1. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

2. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp hold time becomes "from SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

3. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to SOp output becomes "from SCKp↑" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

4. C is the load capacitance of the SCKp and SOp output lines.

Caution Select the normal input buffer for the SIp pin and the normal output mode for the SOp and SCKp pins by using port input mode register 1 (PIM1) and port output mode registers 0, 1, 4 (POM0, POM1, POM4).

Remarks 1. p: CSI number (p = 00, 01, 11, 20), m: Unit number (m = 0, 1), n: Channel number (n = 0, 1, 3)

2. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the serial clock select register m (SPSm) and the CKSmn bit of serial mode register mn (SMRmn). m: Unit number (m = 0, 1), n: Channel number (n = 0, 1, 3))

Downloaded from Arrow.com.

(3) During communication at same potential (CSI mode) (slave mode, SCKp... external clock input)

(T_A = –40 to +105°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		HS (high-speed main) Mode		Unit
				MIN.	MAX.	
SCKp cycle time Note4	t_{KCY2}	4.0 V ≤ V_{DD} ≤ 5.5 V	20 MHz < f_{MCK}	16/f_{MCK}		ns
			f_{MCK} ≤ 20 MHz	12/f_{MCK}		ns
		2.7 V ≤ V_{DD} ≤ 5.5 V	16 MHz < f_{MCK}	16/f_{MCK}		ns
			f_{MCK} ≤ 16 MHz	12/f_{MCK}		ns
		2.4 V ≤ V_{DD} ≤ 5.5 V		12/f_{MCK} and 1000		ns
SCKp high-/low-level width	t_{KH2}, t_{KL2}	4.0 V ≤ V_{DD} ≤ 5.5 V		t_{KCY2}/2–14		ns
		2.7 V ≤ V_{DD} ≤ 5.5 V		t_{KCY2}/2–16		ns
		2.4 V ≤ V_{DD} ≤ 5.5 V		t_{KCY2}/2–36		ns
SIp setup time (to SCKp↑) Note 1	t_{SIK2}	2.7 V ≤ V_{DD} ≤ 5.5 V		1/f_{MCK} + 40		ns
		2.4 V ≤ V_{DD} ≤ 5.5 V		1/f_{MCK} + 60		ns
SIp hold time (from SCKp↑) Note 2	t_{KSI2}			1/f_{MCK} + 62		ns
Delay time from SCKp↓ to SOp output Note 3	t_{KSO2}	C = 30 pF Note4	2.7 V ≤ V_{DD} ≤ 5.5 V		2/f_{MCK} + 66	ns
			2.4 V ≤ V_{DD} ≤ 5.5 V		2/f_{MCK} + 113	ns

Notes 1. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

2. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp hold time becomes "from SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

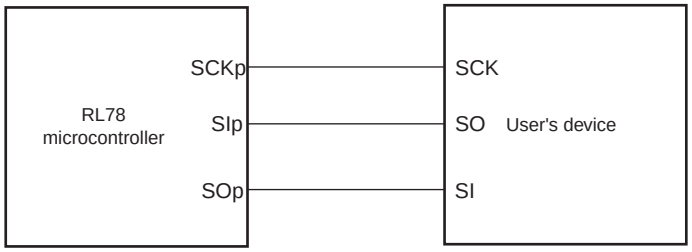
3. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to SOp output becomes "from SCKp↑" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

4. C is the load capacitance of the SOp output lines.

5. Transfer rate in the SNOOZE mode: MAX. 1 Mbps

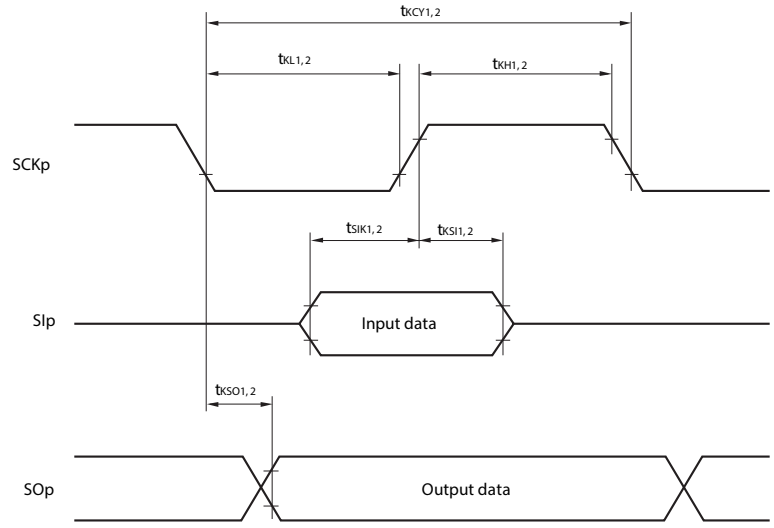
Caution Select the normal input buffer for the SIp and SCKp pins and the normal output mode for the SOp pin by using port input mode register 1 (PIM1) and port output mode registers 0, 1, 4 (POM0, POM1, POM4).

CSI mode connection diagram (during communication at same potential)

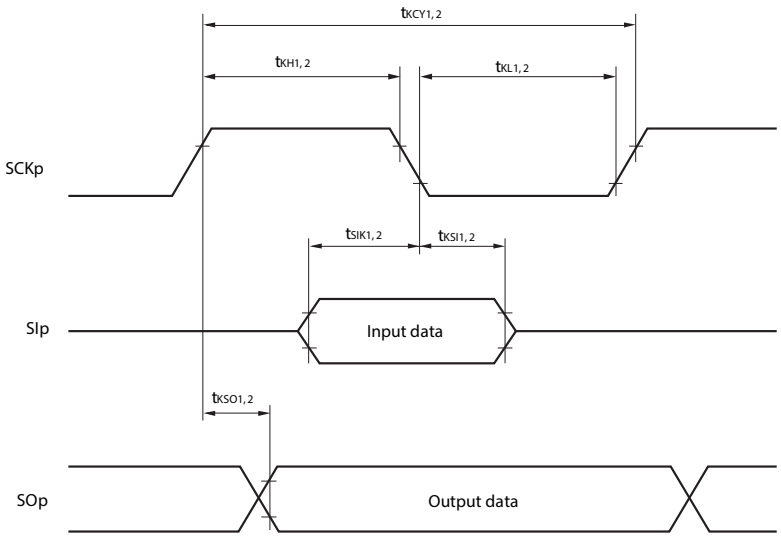


Downloaded from Arrow.com.

CSI mode serial transfer timing (during communication at same potential)
(When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.)



CSI mode serial transfer timing (during communication at same potential)
(When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.)



Remarks 1. p: CSI number (p = 00, 01, 11, 20), m: Unit number (m = 0, 1), n: Channel number (n = 0, 1, 3)

2. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the serial clock select register m (SPSm) and the CKSmn bit of serial mode register mn (SMRmn). m: Unit number (m = 0,1), n: Channel number (n = 0, 1, 3))

Downloaded from Arrow.com.

(4) During communication at same potential (simplified I^2C mode)

(T_A = –40 to +105°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

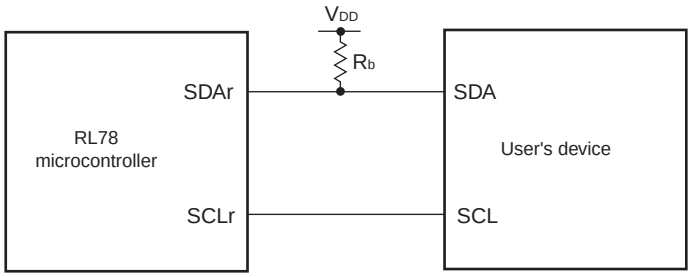
Parameter	Symbol	Conditions	HS (high-speed main) Mode		Unit
			MIN.	MAX.	
SCLr clock frequency	f_{SCL}	C_b = 100 pF, R_b = 3 kΩ		100 Note 1	kHz
Hold time when SCLr = "L"	t_{LOW}	C_b = 100 pF, R_b = 3 kΩ	4600		ns
Hold time when SCLr = "H"	t_{HIGH}	C_b = 100 pF, R_b = 3 kΩ	4600		ns
Data setup time (reception)	$t_{SU:DAT}$	C_b = 100 pF, R_b = 3 kΩ	$1/f_{MCK}$ + 580 Note 2		ns
Data hold time (transmission)	$t_{HD:DAT}$	C_b = 100 pF, R_b = 3 kΩ	0	1420	ns

Notes 1. The value must also be equal to or less than f_{MCK}/4.

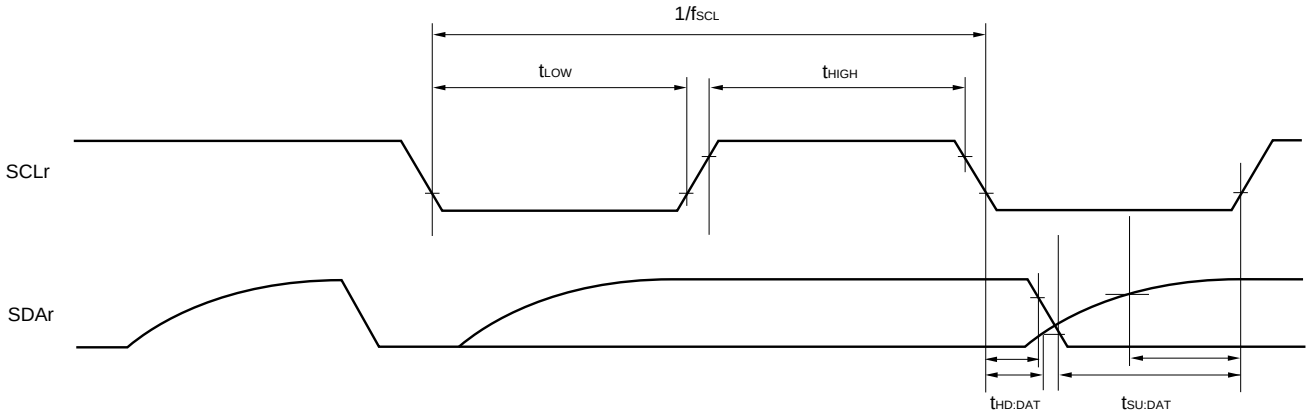
2. Set $t_{SU:DAT}$ so that it will not exceed the hold time when SCLr = "L" or SCLr = "H".

Caution Select the N-ch open drain output (V_{DD} tolerance) mode for SDAr by using port output mode register h (POMh).

Simplified I^2C mode connection diagram (during communication at same potential)



Simplified I^2C mode serial transfer timing (during communication at same potential)



Remarks 1. R_b [Ω]:Communication line (SDAr) pull-up resistance

C_b [F]: Communication line (SCLr, SDAr) load capacitance

2. r: IIC number (r = 00, 01, 11, 20), h: = POM number (h = 0, 1, 4, 5)

3. f_{MCK}: Serial array unit operation clock frequency

(Operation clock to be set by the serial clock select register m (SPSm) and the CKSmn bit of serial mode register mn (SMRmn).

m: Unit number (m = 0, 1), n: Channel number (0, 1, 3))

Downloaded from Arrow.com.

(5) Communication at different potential (1.8 V, 2.5 V, 3 V) (UART mode)

(T_A = –40 to +105°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		HS (high-speed main) Mode		Unit
				MIN.	MAX.	
Transfer rate Note4		Reception	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V		f_{MCK}/12 **Note 1**	bps
			Theoretical value of the maximum transfer rate f_{MCK} = f_{CLK} **Note 2**		2.0	Mbps
			2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V		f_{MCK}/12 **Note 1**	bps
			Theoretical value of the maximum transfer rate f_{MCK} = f_{CLK} **Note 2**		2.0	Mbps
			2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V		f_{MCK}/12 **Note 1**	bps
			Theoretical value of the maximum transfer rate f_{MCK} = f_{CLK} **Note 2**		2.0	Mbps
		Transmission	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V		**Note 3**	bps
			Theoretical value of the maximum transfer rate C_b = 50 pF, R_b = 1.4 kΩ, V_b = 2.7 V		2.0 **Note 4**	Mbps
			2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V,		**Note 5**	bps
			Theoretical value of the maximum transfer rate C_b = 50 pF, R_b = 2.7 kΩ, V_b = 2.3 V		1.2 **Note 6**	Mbps
			2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V		**Notes 2, 7**	bps
			Theoretical value of the maximum transfer rate C_b = 50 pF, R_b = 5.5 kΩ, V_b = 1.6 V		0.43 **Note 8**	Mbps

Notes 1. Transfer rate in the SNOOZE mode is 4800 bps only.

2. The maximum operating frequencies of the CPU/peripheral hardware clock (f_{CLK}) are:
HS (high-speed main) mode: 24 MHz (2.7 V ≤ V_{DD} ≤ 5.5 V)
16 MHz (2.4 V ≤ V_{DD} ≤ 5.5 V)

3. The smaller maximum transfer rate derived by using f_{MCK}/12 or the following expression is the valid maximum transfer rate.
Expression for calculating the transfer rate when 4.0 V ≤ V_{DD} ≤ 5.5 V and 2.7 V ≤ V_b ≤ 4.0 V

$$\text{Maximum transfer rate} = \frac{1}{\{-C_b \times R_b \times \ln(1 - \frac{2.2}{V_b})\} \times 3} \text{ [bps]}$$

Downloaded from Arrow.com.

$$\text{Baud rate error (theoretical value)} = \frac{\frac{1}{\text{Transfer rate} \times 2} - \{-C_b \times R_b \times \ln(1 - \frac{2.2}{V_b})\}}{(\frac{1}{\text{Transfer rate}}) \times \text{Number of transferred bits}} \times 100\ [\%]$$

* This value is the theoretical value of the relative difference between the transmission and reception sides.

4. This value as an example is calculated when the conditions described in the “Conditions” column are met. Refer to **Note 3** above to calculate the maximum transfer rate under conditions of the customer.

5. The smaller maximum transfer rate derived by using f_{MCK}/12 or the following expression is the valid maximum transfer rate.

Expression for calculating the transfer rate when 2.7 V ≤ V_{DD} < 4.0 V and 2.3 V ≤ V_b ≤ 2.7 V

$$\text{Maximum transfer rate} = \frac{1}{\{-C_b \times R_b \times \ln(1 - \frac{2.0}{V_b})\} \times 3}\ [\text{bps}]$$

$$\text{Baud rate error (theoretical value)} = \frac{\frac{1}{\text{Transfer rate} \times 2} - \{-C_b \times R_b \times \ln(1 - \frac{2.0}{V_b})\}}{(\frac{1}{\text{Transfer rate}}) \times \text{Number of transferred bits}} \times 100\ [\%]$$

* This value is the theoretical value of the relative difference between the transmission and reception sides.

6. This value as an example is calculated when the conditions described in the “Conditions” column are met. Refer to **Note 5** above to calculate the maximum transfer rate under conditions of the customer.

7. The smaller maximum transfer rate derived by using f_{MCK}/12 or the following expression is the valid maximum transfer rate.

Expression for calculating the transfer rate when 2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V

$$\text{Maximum transfer rate} = \frac{1}{\{-C_b \times R_b \times \ln(1 - \frac{1.5}{V_b})\} \times 3}\ [\text{bps}]$$

$$\text{Baud rate error (theoretical value)} = \frac{\frac{1}{\text{Transfer rate} \times 2} - \{-C_b \times R_b \times \ln(1 - \frac{1.5}{V_b})\}}{(\frac{1}{\text{Transfer rate}}) \times \text{Number of transferred bits}} \times 100\ [\%]$$

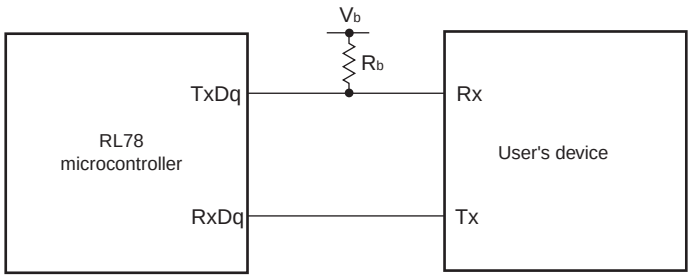
* This value is the theoretical value of the relative difference between the transmission and reception sides.

8. This value as an example is calculated when the conditions described in the “Conditions” column are met. Refer to **Note 7** above to calculate the maximum transfer rate under conditions of the customer.

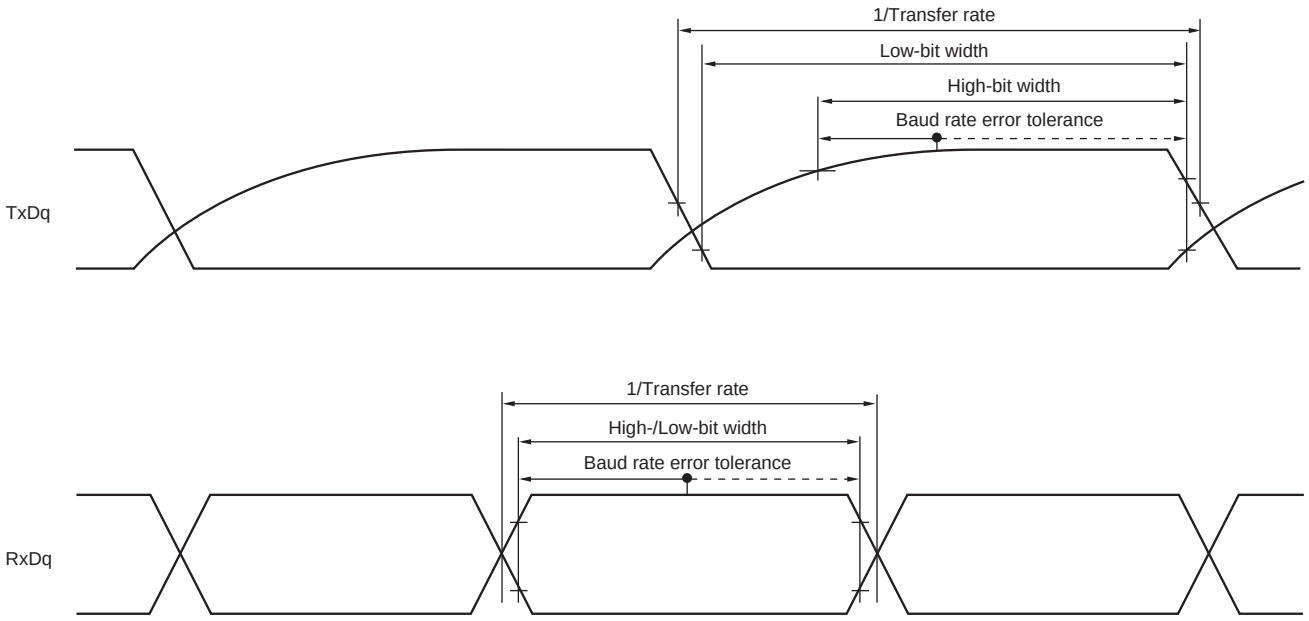
Caution Select the TTL input buffer for the RxDq pin and the N-ch open drain output (V_{DD} tolerance) mode for the TxDq pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.

Downloaded from Arrow.com.

UART mode connection diagram (during communication at different potential)



UART mode bit width (during communication at different potential) (reference)



Remarks 1. R_b[Ω]: Communication line (TxDq) pull-up resistance, C_b[F]: Communication line (TxDq) load capacitance, V_b[V]: Communication line voltage

2. q: UART number (q = 0 to 2), g: PIM and POM number (g = 0, 1)

3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the serial clock select register m (SPSm) and the CKSmn bit of serial mode register mn (SMRmn).
m: Unit number, n: Channel number (mn = 00 to 03, 10, 11))

4. UART0 of the 20- and 24-pin products supports communication at different potential only when the peripheral I/O redirection function is not used.

Downloaded from Arrow.com.

(6) Communication at different potential (1.8 V, 2.5 V, 3 V) (CSI mode) (master mode, SCKp... internal clock output) (1/3)

(T_A = –40 to +105°C, 2.4 V ≤ V_{DD} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		HS (high-speed main) Mode		Unit
				MIN.	MAX.	
SCKp cycle time	t_{KCY1}	$t_{KCY1} \geq 4/f_{CLK}$	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	600		ns
			2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	1000		ns
			2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V, C_b = 30 pF, R_b = 5.5 kΩ	2300		ns
SCKp high-level width	t_{KH1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ		t_{KCY1}/2 –150		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		t_{KCY1}/2 –340		ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V, C_b = 30 pF, R_b = 5.5 kΩ		t_{KCY1}/2 –916		ns
SCKp low-level width	t_{KL1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ		t_{KCY1}/2 –24		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		t_{KCY1}/2 –36		ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V, C_b = 30 pF, R_b = 5.5 kΩ		t_{KCY1}/2 –100		ns

Cautions 1. Select the TTL input buffer for the SIp pin and the N-ch open drain output (V_{DD} tolerance) mode for the SOp pin and SCKp pin by using port input mode register 1 (PIM1) and port output mode register 1 (POM1). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.

2. CSI01 and CSI11 cannot communicate at different potential.

Remarks 1. R_b [Ω]: Communication line (SCKp, SOp) pull-up resistance, C_b [F]: Communication line (SCKp, SOp) load capacitance, V_b [V]: Communication line voltage

2. p: CSI number (p = 00, 20)

Downloaded from Arrow.com.

(6) Communication at different potential (1.8 V, 2.5 V, 3 V) (CSI mode) (master mode, SCKp... internal clock output) (2/3)

(T_A = –40 to +105°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	HS (high-speed main) Mode		Unit
			MIN.	MAX.	
SIp setup time (to SCKp↑) Note	t_{SIK1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	162		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	354		ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V, C_b = 30 pF, R_b = 5.5 kΩ	958		ns
SIp hold time (from SCKp↑) Note	t_{KSI1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	38		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	38		ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V, C_b = 30 pF, R_b = 5.5 kΩ	38		ns
Delay time from SCKp↓ to SOp output Note	t_{KSO1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ		200	ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		390	ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V, C_b = 30 pF, R_b = 5.5 kΩ		966	ns

Note When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.

(**Cautions** and **Remarks** are listed on the next page.)

Downloaded from Arrow.com.

(6) Communication at different potential (1.8 V, 2.5 V, 3 V) (CSI mode) (master mode, SCKp... internal clock output) (3/3)

(T_A = –40 to +105°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	HS (high-speed main) Mode		Unit
			MIN.	MAX.	
SIp setup time (to SCKp↓) Note	t_{SIK1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	88		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	88		ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V, C_b = 30 pF, R_b = 5.5 kΩ	220		ns
SIp hold time (from SCKp↓) Note	t_{KSI1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	38		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	38		ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V, C_b = 30 pF, R_b = 5.5 kΩ	38		ns
Delay time from SCKp↑ to SOp output Note	t_{KSO1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ		50	ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		50	ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V, C_b = 30 pF, R_b = 5.5 kΩ		50	ns

Note When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

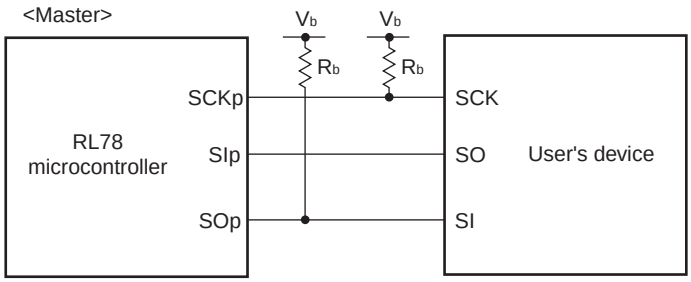
Cautions 1. Select the TTL input buffer for the SIp pin and the N-ch open drain output (V_{DD} tolerance) mode for the SOp pin and SCKp pin by using port input mode register 1 (PIM1) and port output mode register 1 (POM1). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.

2. CSI01 and CSI11 cannot communicate at different potential.

Remarks 1. R_b [Ω]: Communication line (SCKp, SOp) pull-up resistance, C_b [F]: Communication line (SCKp, SOp) load capacitance, V_b [V]: Communication line voltage

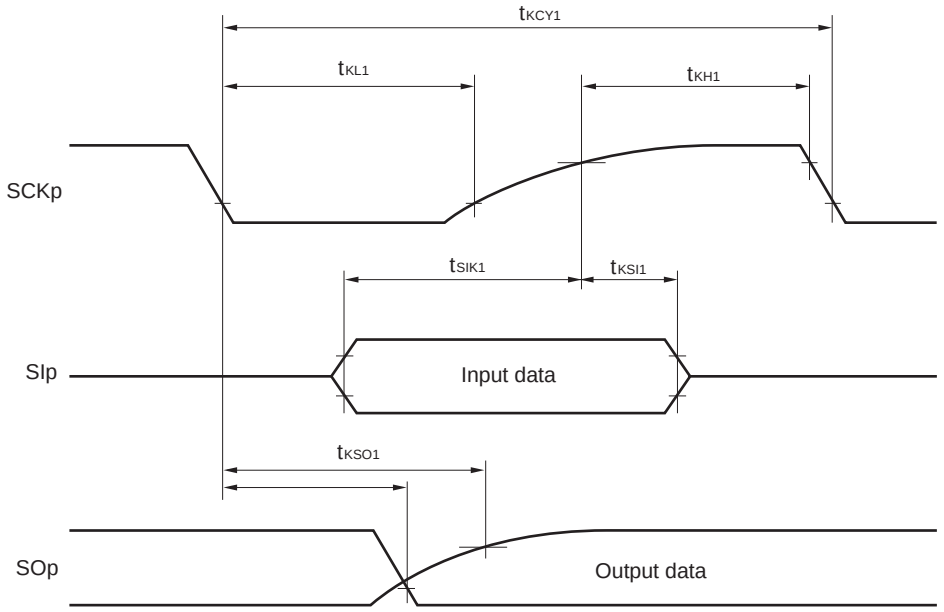
2. p: CSI number (p = 00, 20), m: Unit number (m = 0, 1), n: Channel number (n = 0)

CSI mode connection diagram (during communication at different potential)

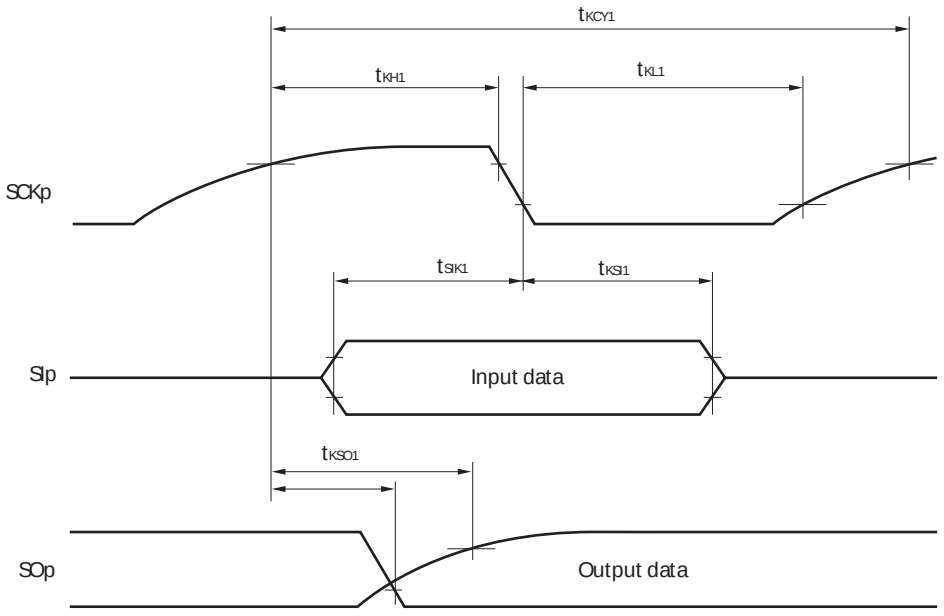


Downloaded from Arrow.com.

CSI mode serial transfer timing (master mode) (during communication at different potential)
(When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1)



CSI mode serial transfer timing (master mode) (during communication at different potential)
(When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.)



Remark p: CSI number (p = 00, 20), m: Unit number (m = 0, 1), n: Channel number (n = 0)

Downloaded from Arrow.com.

(7) Communication at different potential (1.8 V, 2.5 V, 3 V) (CSI mode) (slave mode, SCKp... external clock input)
(T_A = –40 to +105°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		HS (high-speed main) Mode		Unit
				MIN.	MAX.	
SCKp cycle time Note 1	t_{KCY2}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V	20 MHz < f_{MCK} ≤ 24 MHz	24/f_{MCK}		ns
			8 MHz < f_{MCK} ≤ 20 MHz	20/f_{MCK}		ns
			4 MHz < f_{MCK} ≤ 8 MHz	16/f_{MCK}		ns
			f_{MCK} ≤ 4 MHz	12/f_{MCK}		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V	20 MHz < f_{MCK} ≤ 24 MHz	32/f_{MCK}		ns
			16 MHz < f_{MCK} ≤ 20 MHz	28/f_{MCK}		ns
			8 MHz < f_{MCK} ≤ 16 MHz	24/f_{MCK}		ns
			4 MHz < f_{MCK} ≤ 8 MHz	16/f_{MCK}		ns
			f_{MCK} ≤ 4 MHz	12/f_{MCK}		ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V	20 MHz < f_{MCK} ≤ 24 MHz	72/f_{MCK}		ns
			16 MHz < f_{MCK} ≤ 20 MHz	64/f_{MCK}		ns
			8 MHz < f_{MCK} ≤ 16 MHz	52/f_{MCK}		ns
			4 MHz < f_{MCK} ≤ 8 MHz	32/f_{MCK}		ns
			f_{MCK} ≤ 4 MHz	20/f_{MCK}		ns
SCKp high-/low-level width	t_{KH2}, t_{KL2}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V		t_{KCY2}/2 – 24		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V		t_{KCY2}/2 – 36		ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V		t_{KCY2}/2 – 100		ns
SIp setup time (to SCKp↑) Note 2	t_{SIK2}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_{DD} ≤ 4.0 V		1/f_{MCK} + 40		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V		1/f_{MCK} + 40		ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_{DD} ≤ 2.0 V		1/f_{MCK} + 60		ns
SIp hold time (from SCKp↑) Note 3	t_{KSI2}			1/f_{MCK} + 62		ns
Delay time from SCKp↓ to SOp output Note 4	t_{KSO2}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ			2/f_{MCK} + 240	ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ			2/f_{MCK} + 428	ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V, C_b = 30 pF, R_b = 5.5 kΩ			2/f_{MCK} + 1146	ns

Notes 1. Transfer rate in the SNOOZE mode: MAX. 1 Mbps

2. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

3. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp hold time becomes "from SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

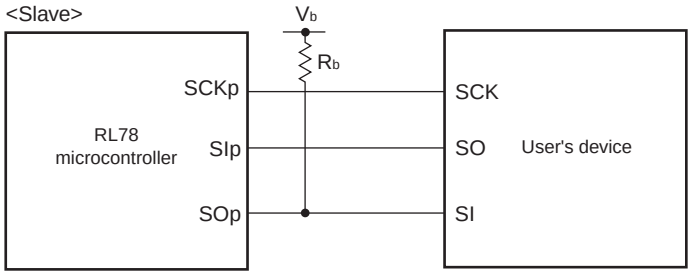
4. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to SOp output becomes "from SCKp↑" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Cautions 1. Select the TTL input buffer for the SIp and SCKp pins and the N-ch open drain output (V_{DD} tolerance) mode for the SOp pin by using port input mode register 1 (PIM1) and port output mode register 1 (POM1). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.

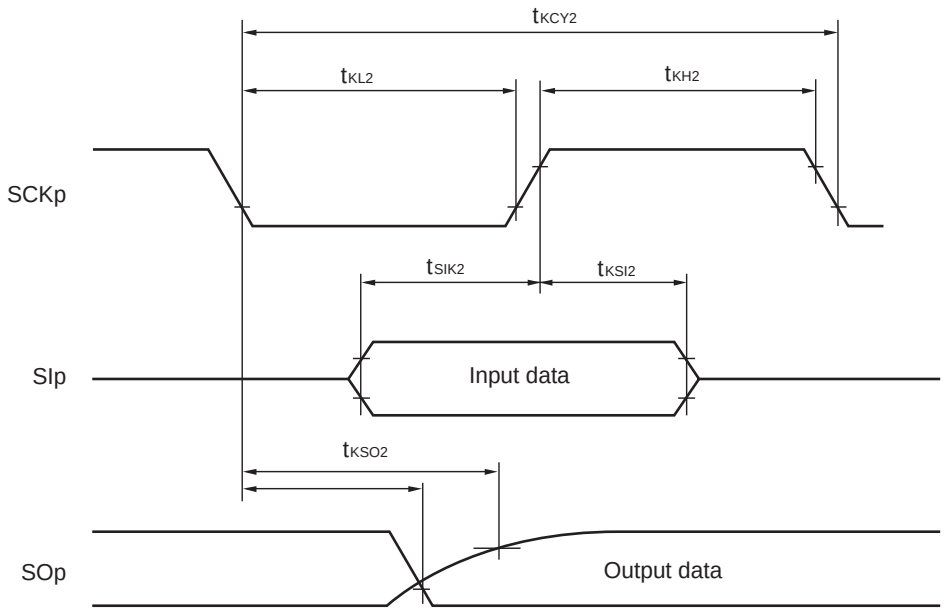
2. CSI01 and CSI11 cannot communicate at different potential.

Downloaded from Arrow.com.

CSI mode connection diagram (during communication at different potential)



CSI mode serial transfer timing (slave mode) (during communication at different potential)
(When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.)



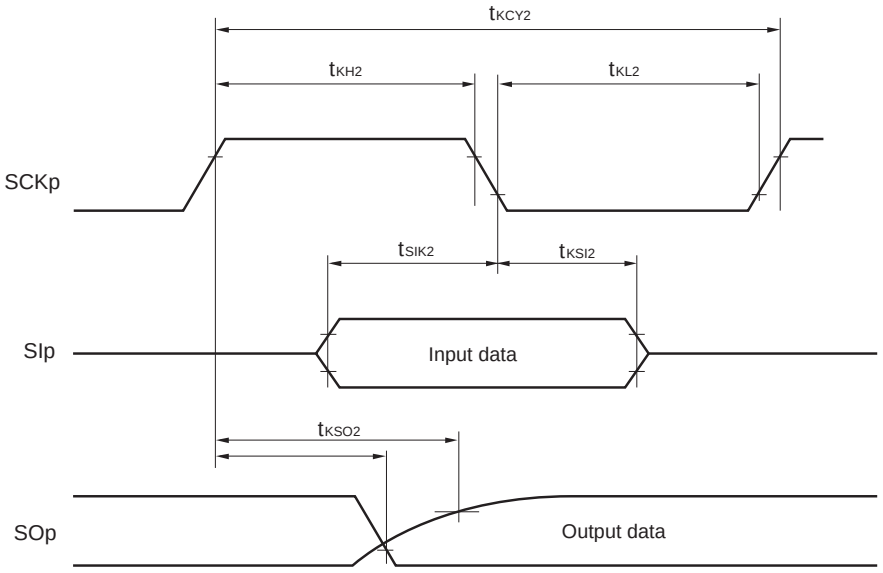
Remarks 1. R_b [Ω]: Communication line (SOp) pull-up resistance, C_b [F]: Communication line (SOp) load capacitance, V_b [V]: Communication line voltage

2. p: CSI number (p = 00, 20), m: Unit number (m = 0, 1), n: Channel number (n = 0)

3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the serial clock select register m (SPSm) and the CKSmn bit of serial mode register mn (SMRmn))

Downloaded from Arrow.com.

CSI mode serial transfer timing (slave mode) (during communication at different potential)
(When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.)



Remark p: CSI number (p = 00, 20), m: Unit number (m = 0, 1), n: Channel number (n = 0)

Downloaded from Arrow.com.

(8) Communication at different potential (1.8 V, 2.5 V, 3 V) (simplified I^2C mode)

(T_A = –40 to +105°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	HS (high-speed main) Mode		Unit
			MIN.	MAX.	
SCLr clock frequency	f_{SCL}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 2.8 kΩ		100 Note1	kHz
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ		100 Note1	kHz
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V, C_b = 100 pF, R_b = 5.5 kΩ		100 Note1	kHz
Hold time when SCLr = "L"	t_{LOW}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 2.8 kΩ	4600		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ	4600		ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V, C_b = 100 pF, R_b = 5.5 kΩ	4650		ns
Hold time when SCLr = "H"	t_{HIGH}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 2.8 kΩ	2700		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ	2400		ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V, C_b = 100 pF, R_b = 5.5 kΩ	1830		ns
Data setup time (reception)	$t_{SU:DAT}$	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 2.8 kΩ	1/f_{MCK} + 760 Note3		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ	1/f_{MCK} + 760 Note3		ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V, C_b = 100 pF, R_b = 5.5 kΩ	1/f_{MCK} + 570 Note3		ns
Data hold time (transmission)	$t_{HD:DAT}$	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 2.8 kΩ	0	1420	ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ	0	1420	ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V, C_b = 100 pF, R_b = 5.5 kΩ	0	1215	ns

Notes 1. The value must also be equal to or less than f_{MCK}/4.

2. Set $t_{SU:DAT}$ so that it will not exceed the hold time when SCLr = "L" or SCLr = "H".

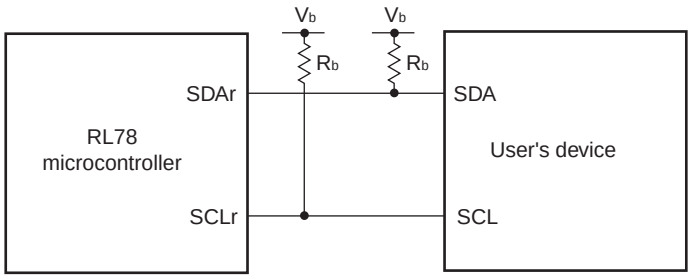
Cautions 1. Select the TTL input buffer and the N-ch open drain output (V_{DD} tolerance) mode for the SDAr pin and the N-ch open drain output (V_{DD} tolerance) mode for the SCLr pin by using port input mode register 1 (PIM1) and port output mode register 1 (POM1). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.

2. IIC01 and IIC11 cannot communicate at different potential.

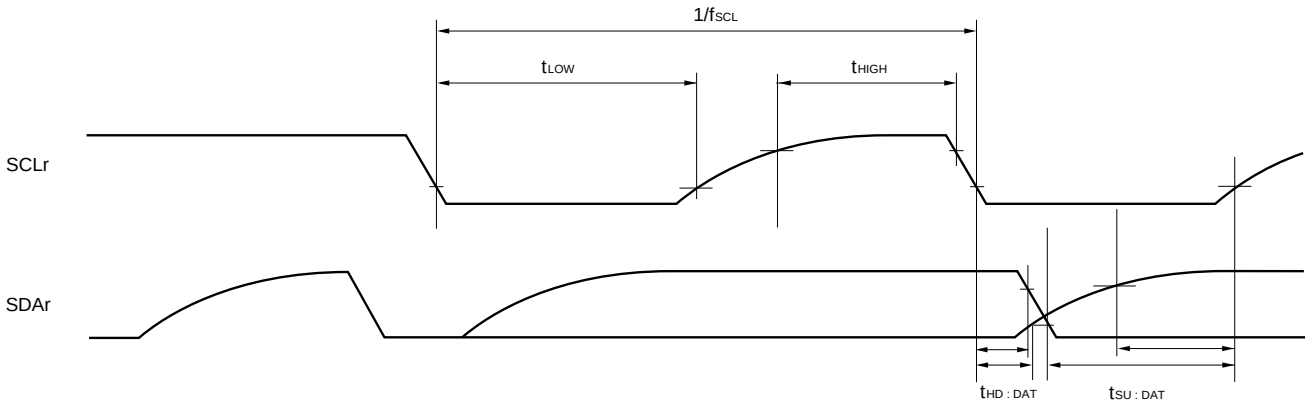
(**Remarks** are listed on the next page.)

Downloaded from Arrow.com.

Simplified I^2C mode connection diagram (during communication at different potential)



Simplified I^2C mode serial transfer timing (during communication at different potential)



Remarks 1. R_b [Ω]: Communication line (SDAr, SCLr) pull-up resistance, C_b [F]: Communication line (SDAr, SCLr) load capacitance, V_b [V]: Communication line voltage

2. r: IIC Number (r = 00, 20)

3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the serial clock select register m (SPSm) and the CKSmn bit of serial mode register mn (SMRmn).
m: Unit number (m = 0,1), n: Channel number (n = 0))

Downloaded from Arrow.com.

29.5.2 Serial interface IICA

(T_A = –40 to +105°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	HS (high-speed main) mode				Unit
			Standard Mode		Fast Mode		
			MIN.	MAX.	MIN.	MAX.	
SCLA0 clock frequency	f_{SCL}	Fast mode: f_{CLK} ≥ 3.5 MHz			0	400	kHz
		Normal mode: f_{CLK} ≥ 1 MHz	0	100			kHz
Setup time of restart condition	$t_{SU:STA}$		4.7		0.6		μs
Hold time[Note 1]	$t_{HD:STA}$		4.0		0.6		μs
Hold time when SCLA0 = “L”	t_{LOW}		4.7		1.3		μs
Hold time when SCLA0 = “H”	t_{HIGH}		4.0		0.6		μs
Data setup time (reception)	$t_{SU:DAT}$		250		100		ns
Data hold time (transmission)[Note 2]	$t_{HD:DAT}$		0	3.45	0	0.9	μs
Setup time of stop condition	$t_{SU:STO}$		4.0		0.6		μs
Bus-free time	t_{BUF}		4.7		1.3		μs

Notes 1. The first clock pulse is generated after this period when the start/restart condition is detected.

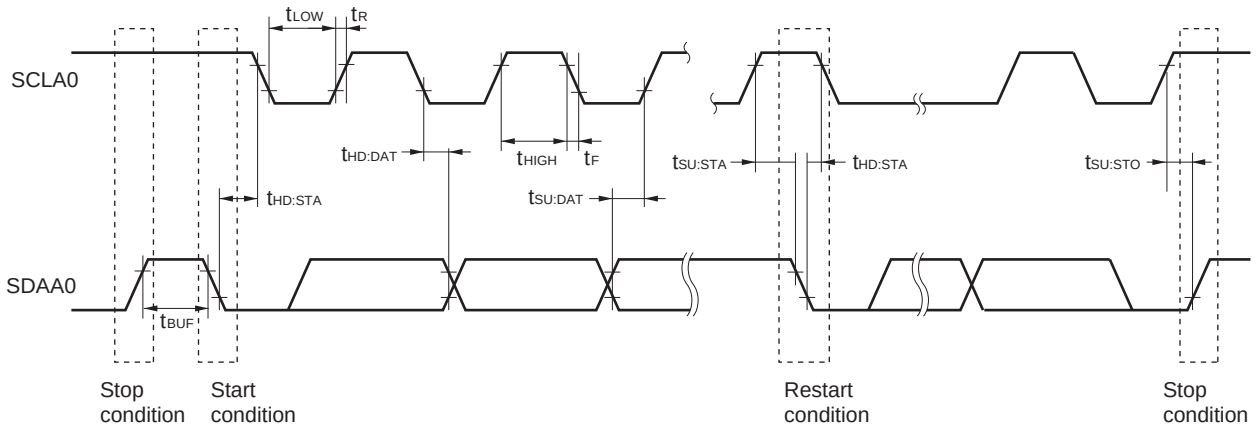
2. The maximum value (MAX.) of $t_{HD:DAT}$ is during normal transfer and a wait state is inserted in the $\overline{ACK}$ (acknowledge) timing.

Caution Only in the 30-pin products, the values in the above table are applied even when bit 2 (PIOR2) in the peripheral I/O redirection register (PIOR) is 1. At this time, the pin characteristics (I_{OH1}, I_{OL1}, V_{OH1}, V_{OL1}) must satisfy the values in the redirect destination.

Remark The maximum value of C_b (communication line capacitance) and the value of R_b (communication line pull-up resistor) at that time in each mode are as follows.

Normal mode: C_b = 400 pF, Rb = 2.7 kΩ
Fast mode: C_b = 320 pF, Rb = 1.1 kΩ

IICA serial transfer timing



Downloaded from Arrow.com.

29.6 Analog Characteristics

29.6.1 A/D converter characteristics

Classification of A/D converter characteristics

Input channel	Reference Voltage		
	Reference voltage (+) = AV_{REFP} Reference voltage (–) = AV_{REFM}	Reference voltage (+) = V_{DD} Reference voltage (–) = V_{SS}	Reference voltage (+) = V_{BGR} Reference voltage (–) = AV_{REFM}
ANI0 to ANI3	Refer to **29.6.1 (1)**.	Refer to **29.6.1 (3)**.	Refer to **29.6.1 (4)**.
ANI16 to ANI22	Refer to **29.6.1 (2)**.		
Internal reference voltage Temperature sensor output voltage	Refer to **29.6.1 (1)**.		–

(1) When reference voltage (+) = AV_{REFP}/ANI0 (ADREFP1 = 0, ADREFP0 = 1), reference voltage (–) = AV_{REFM}/ANI1 (ADREFM = 1), target pin: ANI2, ANI3, internal reference voltage, and temperature sensor output voltage

(T_A = –40 to +105°C, 2.4 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V, Reference voltage (+) = AV_{REFP}, Reference voltage (–) = AV_{REFM} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	RES			8		10	bit
Overall error[Note 1]	AINL	10-bit resolution AV_{REFP} = V_{DD} [Note 3]			1.2	±3.5	LSB
Conversion time	t_{CONV}	10-bit resolution Target pin: ANI2, ANI3	3.6 V ≤ V_{DD} ≤ 5.5 V	2.125		39	μs
			2.7 V ≤ V_{DD} ≤ 5.5 V	3.1875		39	μs
			2.4 V ≤ V_{DD} ≤ 5.5 V	17		39	μs
		10-bit resolution Target pin: Internal reference voltage, and temperature sensor output voltage (HS (high-speed main) mode)	3.6 V ≤ V_{DD} ≤ 5.5 V	2.375		39	μs
			2.7 V ≤ V_{DD} ≤ 5.5 V	3.5625		39	μs
			2.4 V ≤ V_{DD} ≤ 5.5 V	17		39	μs
Zero-scale error[Notes 1, 2]	EZS	10-bit resolution AV_{REFP} = V_{DD} [Note 3]				±0.25	%FSR
Full-scale error[Notes 1, 2]	EFS	10-bit resolution AV_{REFP} = V_{DD} [Note 3]				±0.25	%FSR
Integral linearity error[Note 1]	ILE	10-bit resolution AV_{REFP} = V_{DD} [Note 3]				±2.5	LSB
Differential linearity error [Note 1]	DLE	10-bit resolution AV_{REFP} = V_{DD} [Note 3]				±1.5	LSB
Analog input voltage	V_{AIN}	ANI2, ANI3		0		AV_{REFP}	V
		Internal reference voltage (HS (high-speed main) mode)		V_{BGR} [Note 4]			V
		Temperature sensor output voltage (HS (high-speed main) mode)		V_{TMPS25} [Note 4]			V

(**Notes** are listed on the next page.)

Downloaded from Arrow.com.

Notes 1. Excludes quantization error (±1/2 LSB).

2. This value is indicated as a ratio (%FSR) to the full-scale value.

3. When $AV_{REFP} < V_{DD}$, the MAX. values are as follows.
Overall error: Add ±1.0 LSB to the MAX. value when $AV_{REFP} = V_{DD}$.
Zero-scale error/Full-scale error: Add ±0.05%FSR to the MAX. value when $AV_{REFP} = V_{DD}$.
Integral linearity error/ Differential linearity error: Add ±0.5 LSB to the MAX. value when $AV_{REFP} = V_{DD}$.

4. Refer to **29.6.2 Temperature sensor/internal reference voltage characteristics**.

(2) When reference voltage (+) = AV_{REFP}/ANI0 (ADREFP1 = 0, ADREFP0 = 1), reference voltage (–) = AV_{REFM}/ANI1 (ADREFM = 1), target pin: ANI16 to ANI22

(T_A = –40 to +105°C, 2.4 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V, Reference voltage (+) = AV_{REFP}, Reference voltage (–) = AV_{REFM} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	R_{ES}			8		10	bit
Overall error Note 1	AINL	10-bit resolution $AV_{REFP} = V_{DD}$ Note 3			1.2	±5.0	LSB
Conversion time	t_{CONV}	10-bit resolution Target ANI pin: ANI16 to ANI22	3.6 V ≤ V_{DD} ≤ 5.5 V	2.125		39	μs
			2.7 V ≤ V_{DD} ≤ 5.5 V	3.1875		39	μs
			2.4 V ≤ V_{DD} ≤ 5.5 V	17		39	μs
Zero-scale error Notes 1, 2	EZS	10-bit resolution $AV_{REFP} = V_{DD}$ Note 3				±0.35	%FSR
Full-scale error Notes 1, 2	EFS	10-bit resolution $AV_{REFP} = V_{DD}$ Note 3				±0.35	%FSR
Integral linearity error Note 1	ILE	10-bit resolution $AV_{REFP} = V_{DD}$ Note 3				±3.5	LSB
Differential linearity error Note 1	DLE	10-bit resolution $AV_{REFP} = V_{DD}$ Note 3				±2.0	LSB
Analog input voltage	V_{AIN}	ANI16 to ANI22		0		AV_{REFP} and V_{DD}	V

Notes 1. Excludes quantization error (±1/2 LSB).

2. This value is indicated as a ratio (%FSR) to the full-scale value.

3. When $AV_{REFP} \leq V_{DD}$, the MAX. values are as follows.
Overall error: Add ±4.0 LSB to the MAX. value when $AV_{REFP} = V_{DD}$.
Zero-scale error/Full-scale error: Add ±0.20%FSR to the MAX. value when $AV_{REFP} = V_{DD}$.
Integral linearity error/ Differential linearity error: Add ±2.0 LSB to the MAX. value when $AV_{REFP} = V_{DD}$.

Downloaded from Arrow.com.

(3) When reference voltage (+) = V_{DD} (ADREFP1 = 0, ADREFP0 = 0), reference voltage (–) = V_{SS} (ADREFM = 0), target pin: ANI0 to ANI3, ANI16 to ANI22, internal reference voltage, and temperature sensor output voltage

(T_A = –40 to +105°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V, Reference voltage (+) = V_{DD}, Reference voltage (–) = V_{SS})

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	R_{ES}			8		10	bit
Overall error[Note 1]	AINL	10-bit resolution			1.2	±7.0	LSB
Conversion time	t_{CONV}	10-bit resolution Target pin: ANI0 to ANI3, ANI16 to ANI22	3.6 V ≤ V_{DD} ≤ 5.5 V	2.125		39	μs
			2.7 V ≤ V_{DD} ≤ 5.5 V	3.1875		39	μs
			2.4 V ≤ V_{DD} ≤ 5.5 V	17		39	μs
Conversion time	t_{CONV}	10-bit resolution Target pin: internal reference voltage, and temperature sensor output voltage (HS (high-speed main) mode)	3.6 V ≤ V_{DD} ≤ 5.5 V	2.375		39	μs
			2.7 V ≤ V_{DD} ≤ 5.5 V	3.5625		39	μs
			2.4 V ≤ V_{DD} ≤ 5.5 V	17		39	μs
Zero-scale error[Notes 1, 2]	EZS	10-bit resolution				±0.60	%FSR
Full-scale error[Notes 1, 2]	EFS	10-bit resolution				±0.60	%FSR
Integral linearity error[Note 1]	ILE	10-bit resolution				±4.0	LSB
Differential linearity error[Note 1]	DLE	10-bit resolution				±2.0	LSB
Analog input voltage	V_{AIN}	ANI0 to ANI3, ANI16 to ANI22		0		V_{DD}	V
		Internal reference voltage (HS (high-speed main) mode)		V_{BGR} [Note 3]			V
		Temperature sensor output voltage (HS (high-speed main) mode)		V_{TMPS25} [Note 3]			V

Notes 1. Excludes quantization error (±1/2 LSB).

2. This value is indicated as a ratio (%FSR) to the full-scale value.

3. Refer to **29.6.2 Temperature sensor/internal reference voltage characteristics**.

Downloaded from Arrow.com.

(4) When reference voltage (+) = Internal reference voltage (ADREFP1 = 1, ADREFP0 = 0), reference voltage (–) = AV_{REFM} (ADREFM = 1), target pin: ANI0, ANI2, ANI3, and ANI16 to ANI22

(T_A = –40 to +105°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V, Reference voltage (+) = V_{BGR} Note 3, Reference voltage (–) = AV_{REFM} Note 4 = 0 V, HS (high-speed main) mode)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution	R_{ES}		8			bit
Conversion time	t_{CONV}	8-bit resolution	17		39	μs
Zero-scale error Notes 1, 2	EZS	8-bit resolution			±0.60	%FSR
Integral linearity error Note 1	ILE	8-bit resolution			±2.0	LSB
Differential linearity error Note 1	DLE	8-bit resolution			±1.0	LSB
Analog input voltage	V_{AIN}		0		V_{BGR} Note 3	V

Notes 1. Excludes quantization error (±1/2 LSB).

2. This value is indicated as a ratio (%FSR) to the full-scale value.

3. Refer to **29.6.2 Temperature sensor/internal reference voltage characteristics**.

4. When reference voltage (–) = V_{SS}, the MAX. values are as follows.
Zero-scale error: Add ±0.35%FSR to the MAX. value when reference voltage (–) = AV_{REFM}.
Integral linearity error: Add ±0.5 LSB to the MAX. value when reference voltage (–) = AV_{REFM}.
Differential linearity error: Add ±0.2 LSB to the MAX. value when reference voltage (–) = AV_{REFM}.

Downloaded from Arrow.com.

29.6.2 Temperature sensor/internal reference voltage characteristics

(T_A = –40 to +105°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V, HS (high-speed main) mode

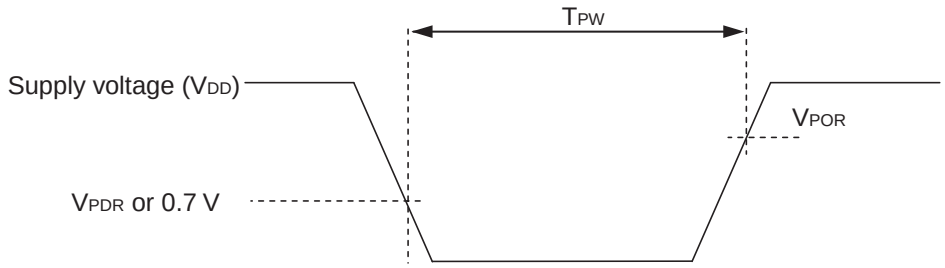
Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Temperature sensor output voltage	V_{TMPS25}	Setting ADS register = 80H, T_A = +25°C		1.05		V
Internal reference voltage	V_{BGR}	Setting ADS register = 81H	1.38	1.45	1.50	V
Temperature coefficient	F_{VTMPS}	Temperature sensor output voltage that depends on the temperature		–3.6		mV/°C
Operation stabilization wait time	t_{AMP}		5			μs

29.6.3 POR circuit characteristics

(T_A = –40 to +105°C, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	V_{POR}	Power supply rise time	1.45	1.51	1.57	V
	V_{PDR}	Power supply fall time	1.44	1.50	1.56	V
Minimum pulse width Note	T_{PW}		300			μs

Note Minimum time required for a POR reset when V_{DD} exceeds below V_{PDR}. This is also the minimum time required for a POR reset from when V_{DD} exceeds below 0.7 V to when V_{DD} exceeds V_{POR} while STOP mode is entered or the main system clock is stopped through setting bit 0 (HIOSTOP) and bit 7 (MSTOP) in the clock operation status control register (CSC).



Downloaded from Arrow.com.

29.6.4 LVD circuit characteristics

LVD Detection Voltage of Reset Mode and Interrupt Mode

(T_A = –40 to +105°C, $V_{PDR} \leq V_{DD} \leq 5.5$ V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection supply voltage	V_{LVD0}	Power supply rise time	3.90	4.06	4.22	V
		Power supply fall time	3.83	3.98	4.13	V
	V_{LVD1}	Power supply rise time	3.60	3.75	3.90	V
		Power supply fall time	3.53	3.67	3.81	V
	V_{LVD2}	Power supply rise time	3.01	3.13	3.25	V
		Power supply fall time	2.94	3.06	3.18	V
	V_{LVD3}	Power supply rise time	2.90	3.02	3.14	V
		Power supply fall time	2.85	2.96	3.07	V
	V_{LVD4}	Power supply rise time	2.81	2.92	3.03	V
		Power supply fall time	2.75	2.86	2.97	V
	V_{LVD5}	Power supply rise time	2.70	2.81	2.92	V
		Power supply fall time	2.64	2.75	2.86	V
	V_{LVD6}	Power supply rise time	2.61	2.71	2.81	V
		Power supply fall time	2.55	2.65	2.75	V
	V_{LVD7}	Power supply rise time	2.51	2.61	2.71	V
		Power supply fall time	2.45	2.55	2.65	V
Minimum pulse width	t_{LW}		300			μs
Detection delay time					300	μs

Downloaded from Arrow.com.

LVD detection voltage of interrupt & reset mode

(T_A = –40 to +105°C, $V_{PDR} \leq V_{DD} \leq$ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Interrupt and reset mode	V_{LVDD0}	V_{POC2}, V_{POC1}, V_{POC1} = 0, 1, 1, falling reset voltage			2.64	2.75	2.86	V
	V_{LVDD1}		LVIS1, LVIS0 = 1, 0	Rising reset release voltage	2.81	2.92	3.03	V
				Falling interrupt voltage	2.75	2.86	2.97	V
	V_{LVDD2}		LVIS1, LVIS0 = 0, 1	Rising reset release voltage	2.90	3.02	3.14	V
				Falling interrupt voltage	2.85	2.96	3.07	V
	V_{LVDD3}		LVIS1, LVIS0 = 0, 0	Rising reset release voltage	3.90	4.06	4.22	V
				Falling interrupt voltage	3.83	3.98	4.13	V

29.6.5 Power supply voltage rising slope characteristics

(T_A = –40 to +105°C, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Power supply voltage rising slope	S_{VDD}				54	V/ms

Caution Make sure to keep the internal reset state by the LVD circuit or an external reset until V_{DD} reaches the operating voltage range shown in 29.4 AC Characteristics.

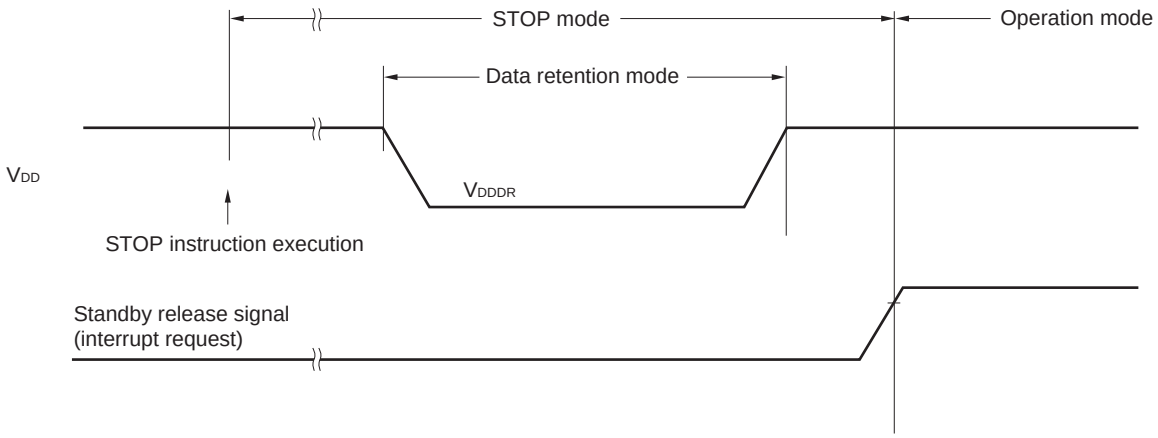
Downloaded from Arrow.com.

29.7 Data Memory STOP Mode Low Supply Voltage Data Retention Characteristics

(T_A = –40 to +105°C, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Data retention supply voltage	V_{DDDR}		1.44 Note		5.5	V

Note The value depends on the POR detection voltage. When the voltage drops, the data is retained before a POR reset is affected, but data is not retained when a POR reset is affected.



29.8 Flash Memory Programming Characteristics

(T_A = –40 to +105°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
System clock frequency	f_{CLK}			1		24	MHz
Code flash memory rewritable times Notes 1, 2, 3	C_{erwr}	Retained for 20 years	T_A = 85°C	1,000			Times
Data flash memory rewritable times Notes 1, 2, 3		Retained for 1 year	T_A = 25°C		1,000,000		
		Retained for 5 years	T_A = 85°C	100,000			
		Retained for 20 years	T_A = 85°C	10,000			

Notes 1. 1 erase + 1 write after the erase is regarded as 1 rewrite. The retaining years are until next rewrite after the rewrite.

2. When using flash memory programmer and Renesas Electronics self programming library

3. These are the characteristics of the flash memory and the results obtained from reliability testing by Renesas Electronics Corporation.

Downloaded from Arrow.com.

29.9 Dedicated Flash Memory Programmer Communication (UART)

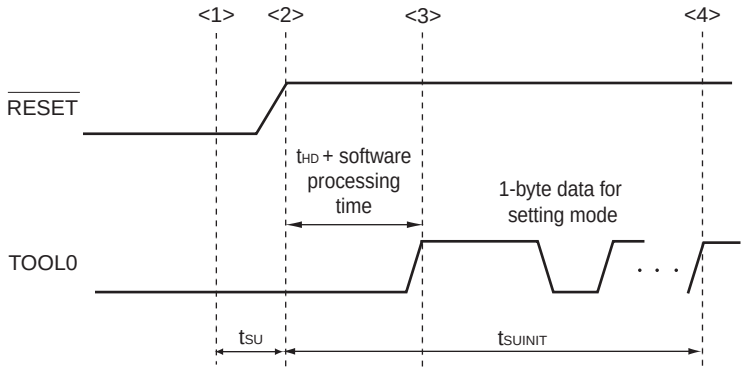
(T_A = –40 to +105°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate		During serial programming	115,200		1,000,000	bps

29.10 Timing of Entry to Flash Memory Programming Modes

(T_A = –40 to +105°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Time to complete the communication for the initial setting after the external reset is released	t_{SUINIT}	POR and LVD reset are released before external release			100	ms
Time to release the external reset after the TOOL0 pin is set to the low level	t_{SU}	POR and LVD reset are released before external release	10			μs
Time to hold the TOOL0 pin at the low level after the external reset is released (excluding the processing time of the firmware to control the flash memory)	t_{HD}	POR and LVD reset are released before external release	1			ms



<1> The low level is input to the TOOL0 pin.
<2> The external reset is released (POR and LVD reset must be released before the external reset is released.).
<3> The TOOL0 pin is set to the high level.
<4> Setting of the flash memory programming mode by UART reception and complete the baud rate setting.

Remark t_{SUINIT}: Communication for the initial setting must be completed within 100 ms after the external reset is released during this period.
t_{SU}: Time to release the external reset after the TOOL0 pin is set to the low level
t_{HD}: Time to hold the TOOL0 pin at the low level after the external reset is released (excluding the processing time of the firmware to control the flash memory)

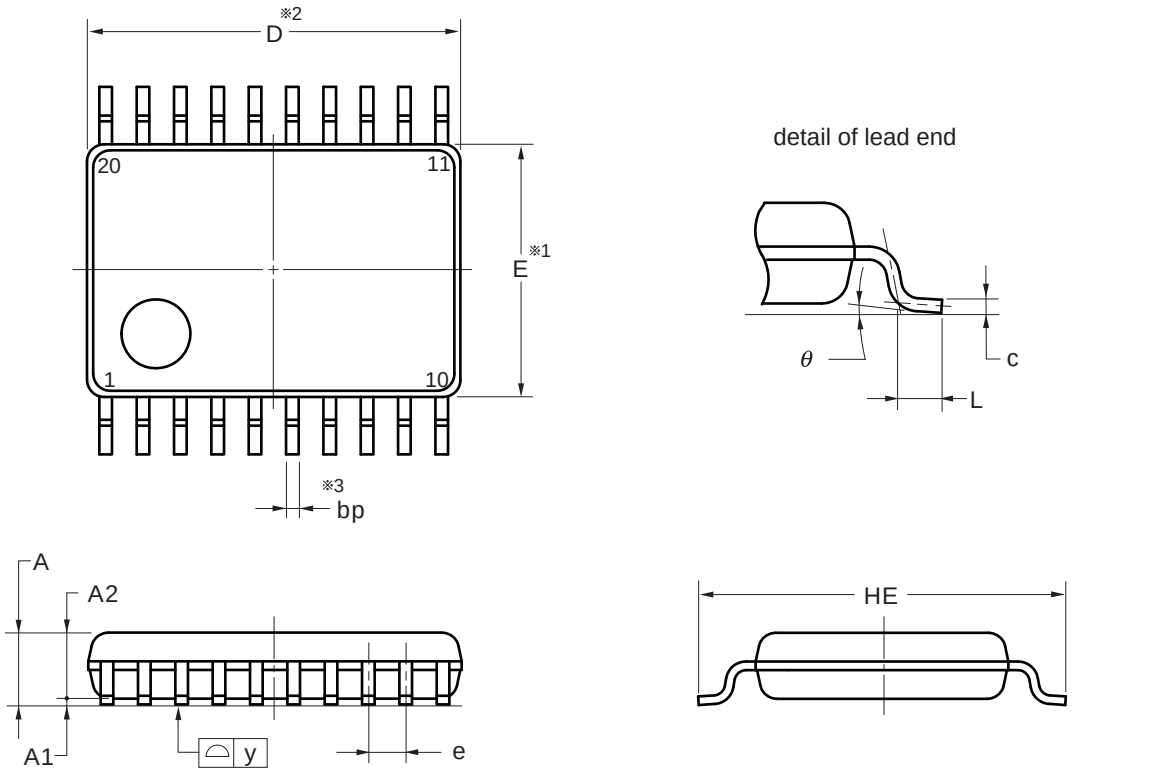
Downloaded from Arrow.com.

CHAPTER 30 PACKAGE DRAWINGS

30.1 20-pin products

R5F1026AASP, R5F10269ASP, R5F10268ASP, R5F10267ASP, R5F10266ASP
R5F1036AASP, R5F10369ASP, R5F10368ASP, R5F10367ASP, R5F10366ASP
R5F1026ADSP, R5F10269DSP, R5F10268DSP, R5F10267DSP, R5F10266DSP
R5F1036ADSP, R5F10369DSP, R5F10368DSP, R5F10367DSP, R5F10366DSP
<R> R5F1026AGSP, R5F10269GSP, R5F10268GSP, R5F10267GSP, R5F10266GSP
<R> R5F1036AGSP, R5F10369GSP, R5F10368GSP, R5F10367GSP, R5F10366GSP

JEITA Package Code	RENESAS Code	Previous Code	MASS (TYP.) [g]
P-LSSOP20-4.4x6.5-0.65	PLSP0020JB-A	P20MA-65-NAA-1	0.1



(UNIT:mm)

ITEM	DIMENSIONS
D	6.50±0.10
E	4.40±0.10
HE	6.40±0.20
A	1.45 MAX.
A1	0.10±0.10
A2	1.15
e	0.65±0.12
bp	$0.22^{+0.10}_{-0.05}$
c	$0.15^{+0.05}_{-0.02}$
L	0.50±0.20
y	0.10
θ	0° to 10°

NOTE

1.Dimensions "※1" and "※2" do not include mold flash.

2.Dimension "※3" does not include trim offset.

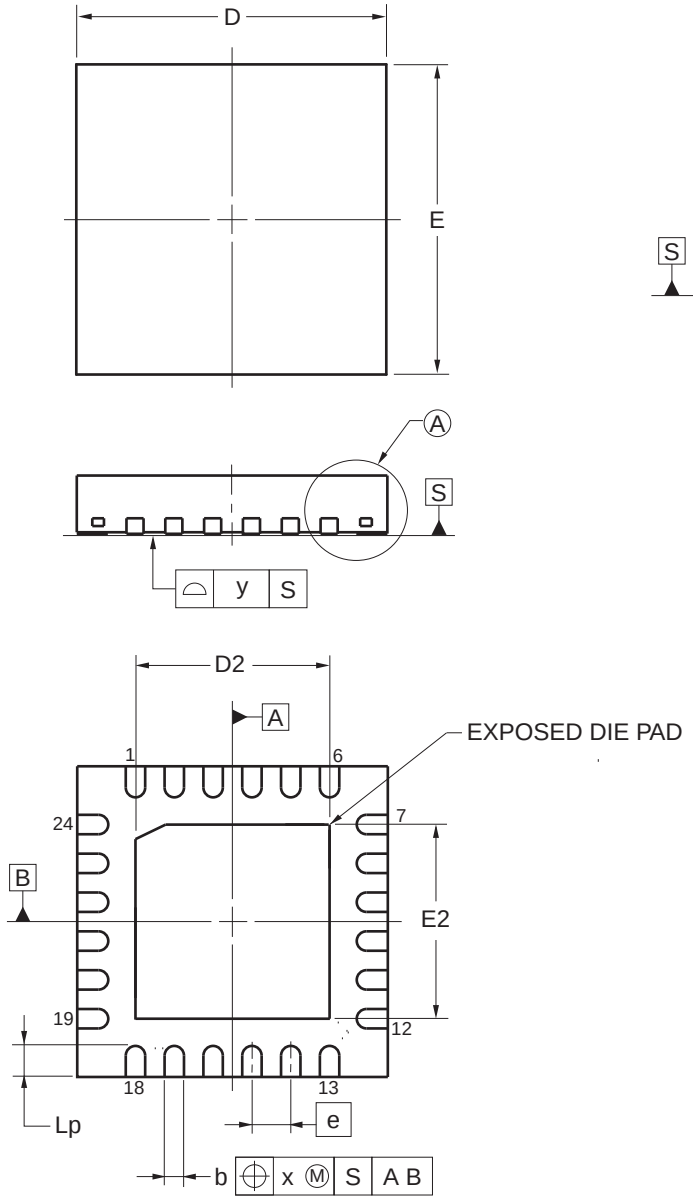
©2012 Renesas Electronics Corporation. All rights reserved.

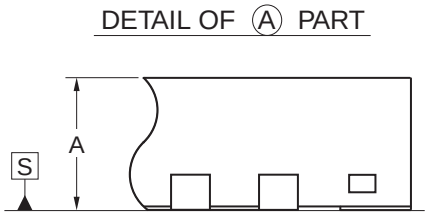
Downloaded from Arrow.com.

30.2 24-pin products

R5F1027AANA, R5F10279ANA, R5F10278ANA, R5F10277ANA
R5F1037AANA, R5F10379ANA, R5F10378ANA, R5F10377ANA
R5F1027ADNA, R5F10279DNA, R5F10278DNA, R5F10277DNA
R5F1037ADNA, R5F10379DNA, R5F10378DNA, R5F10377DNA
<R> R5F1027AGNA, R5F10279GNA, R5F10278GNA, R5F10277GNA
<R> R5F1037AGNA, R5F10379GNA, R5F10378GNA, R5F10377GNA

JEITA Package Code	RENESAS Code	Previous Code	MASS (TYP.) [g]
P-HWQFN24-4x4-0.50	PWQN0024KE-A	P24K8-50-CAB-1	0.04





(UNIT:mm)

ITEM	DIMENSIONS
D	4.00±0.05
E	4.00±0.05
A	0.75±0.05
b	$0.25^{+0.05}_{-0.07}$
e	0.50
Lp	0.40±0.10
x	0.05
y	0.05

ITEM		D2			E2		
		MIN	NOM	MAX	MIN	NOM	MAX
EXPOSED DIE PAD VARIATIONS	A	2.45	2.50	2.55	2.45	2.50	2.55

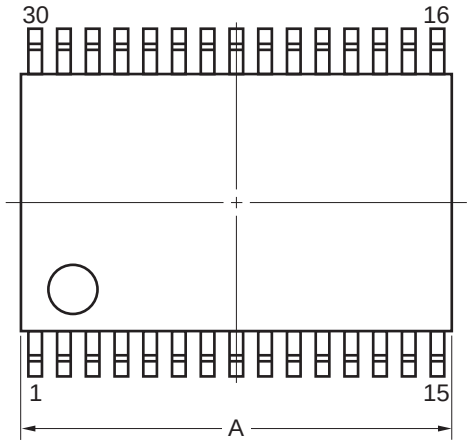
©2012 Renesas Electronics Corporation. All rights reserved.

Downloaded from Arrow.com.

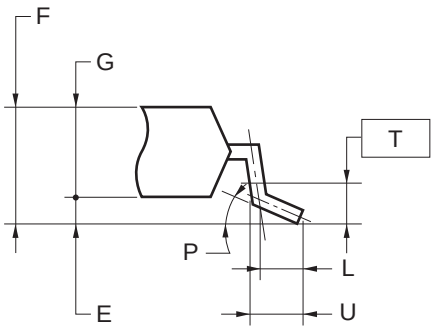
30.3 30-pin products

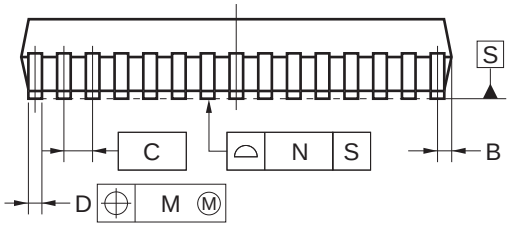
R5F102AAASP, R5F102A9ASP, R5F102A8ASP, R5F102A7ASP
R5F103AAASP, R5F103A9ASP, R5F103A8ASP, R5F103A7ASP
R5F102AADSP, R5F102A9DSP, R5F102A8DSP, R5F102A7DSP
R5F103AADSP, R5F103A9DSP, R5F103A8DSP, R5F103A7DSP
<R> R5F102AAGSP, R5F102A9GSP, R5F102A8GSP, R5F102A7GSP
<R> R5F103AAGSP, R5F103A9GSP, R5F103A8GSP, R5F103A7GSP

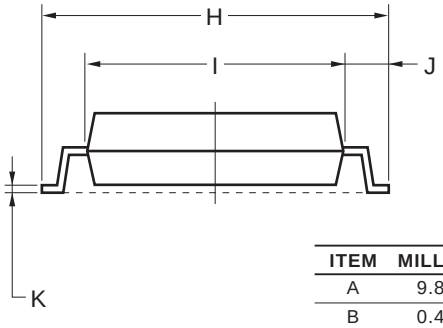
JEITA Package Code	RENESAS Code	Previous Code	MASS (TYP.) [g]
P-LSSOP30-0300-0.65	PLSP0030JB-B	S30MC-65-5A4-3	0.18



detail of lead end







NOTE
Each lead centerline is located within 0.13 mm of its true position (T.P.) at maximum material condition.

ITEM	MILLIMETERS
A	9.85±0.15
B	0.45 MAX.
C	0.65 (T.P.)
D	$0.24^{+0.08}_{-0.07}$
E	0.1±0.05
F	1.3±0.1
G	1.2
H	8.1±0.2
I	6.1±0.2
J	1.0±0.2
K	0.17±0.03
L	0.5
M	0.13
N	0.10
P	$3^{\circ}{}^{+5^{\circ}}_{-3^{\circ}}$
T	0.25
U	0.6±0.15

©2012 Renesas Electronics Corporation. All rights reserved.

Downloaded from Arrow.com.

APPENDIX A REVISION HISTORY

A.1 Major Revisions in This Edition

(1/9)

Page	Description	Classification
CHAPTER 1 OUTLINE		
p.3, 4	Modification of 1.2 Features	(b)
p.6	Modification of 1.3 List of Part Numbers	(b)
p.7	Modification of Table 1-1 List of Ordering Part Numbers, Note, and Caution	(b)
p.8 to 10	Modification of package name in 1.4.1 to 1.4.3	(d)
p.15	Modification of tables in 1.7 Outline of Functions	(c)
CHAPTER 2 PIN FUNCTIONS		
p.17	Modification of description in 2.1 Port Functions	(c)
p.17 to 21	Modification of tables, Notes, and Remark in 2. 1. 1 to 2. 1. 3	(c)
p.25 to 36	Modification of Figures 2-1 to 2-13 in 2.4 Block Diagrams of Pins	(c)
CHAPTER 3 CPU ARCHITECTURE		
p.37, 39 to 43	Modification of Notes of Figures 3-1 to 3-6 in 3.1 Memory Space	(c)
p.51	Modification of Caution 2 in 3.1.3 Internal data memory space	(c)
p.53	Integration of figures in 3.1.6 Data memory addressing into Figure 3-8	(c)
p.55	Error correction and modification of Caution 3 in (3) Stack pointer (SP)	(a) (c)
p.57	Error correction in 3.2.3 ES and CS registers	(a)
p.60, 61	Modification of description in Table 3-6 and Note 1	(c)
p.64	Modification of description in Table 3-7 and Notes 2 and 3	(c)
CHAPTER 4 PORT FUNCTIONS		
p.83	Modification of description of Table 4-1 in 4.2 Port Configuration	(c)
p.90	Modification of Caution in Figure 4-1	(c)
p.92	Modification of Caution in Figure 4-2	(c)
p.93	Modification of description in 4.3.3 Pull-up resistor option registers (PUxx)	(c)
p.93	Modification of Caution in Figure 4-3	(c)
p.94	Modification of Caution in 4.3.4 Port input mode register (PIMx)	(c)
p.95	Modification of Caution in 4.3.5 Port output mode registers (POMx)	(c)
p.95	Modification of Caution in Figure 4-5	(c)
p.96	Modification of Cautions 1 and 3 in 4.3.6 Port mode control registers (PMCxx)	(c)
p.98	Modification of Note 3 in Figure 4-8	(c)
p.102 to 104	Modification of description in 4.5 Register Settings When Using Alternate Function	(c)
p.104 to 107	Modification of description in Table 4-7	(c)
p.108 to 111	Modification of description in Table 4-8	(c)
CHAPTER 5 CLOCK GENERATOR		
p.114	Modification of description in 5.1 Functions of Clock Generator	(c)
p.116	Modification of Figure 5-1	(c)
p.118	Modification of description in 5.3.1 Clock operation mode control register (CMC)	(c)
p.118	Modification of Notes 3 and 4 in Figure 5-2	(c)
p.119	Modification of Caution in Figure 5-3	(c)
p.120	Modification of description in Table 5-2	(c)
p.122	Modification of description in Figure 5-5	(c)

Remark “Classification” in the above table classifies revisions as follows.
(a): Error correction, (b): Addition/change of specifications, (c): Addition/change of description or note,
(d): Addition/change of package, part number, or management division, (e): Addition/change of related documents

Downloaded from Arrow.com.

(2/9)

Page	Description	Classification
p.123	Modification of description in 5.3.5 Oscillation stabilization time select register (OSTS)	(c)
p.123	Modification of Figure 5-6	(c)
p.125	Modification of description of Figure 5-7 (1/2) and Caution	(c)
p.126	Modification of description of Figure 5-7 (2/2) and Caution	(c)
p.126	Modification of description in 5.3.7 Operation speed mode control register (OSMC)	(c)
p.127	Modification of description in 5.3.8 High-speed on-chip oscillator frequency selection register (HOCODIV)	(c)
p.127	Modification of Figure 5-9	(c)
p.128	Modification of Figure 5-10, Note, and Remarks 1 and 2	(c)
p.132	Modification of description in 5.4.2 High-speed on-chip oscillator and 5.4.3 Low-speed on-chip oscillator	(c)
p.133	Modification of Figure 5-13	(b)
p.134	Modification of description in 5.6.1 Example of setting high-speed on-chip oscillator	(c)
p.135	Modification of description in 5.6.2 Example of setting X1 oscillation clock	(c)
p.136	Modification of Figure 5-14	(b)
p.138	Modification of description in Table 5-3 (2/3)	(c)
p.142	Modification of description in 5.7 Resonator and Oscillator Constants and addition of Note 3	(b)
CHAPTER 6 TIMER ARRAY UNIT		
p.143	Addition of description to the beginning of the chapter	(c)
p.145	Modification of 6.1.1 Independent channel operation function (4) Divider and addition of Note	(c)
p.146	Modification of figure of (1) One-shot pulse output in 6.1.2 Simultaneous channel operation function	(b)
p.148	Addition of Note to Table 6-1	(c)
p.148	Modification of Table 6-2	(c)
p.149	Modification of Figure 6-1	(b)
p.150	Modification of Figure 6-2	(b)
p.151 to 153	Modification of Figure 6-3 (a), (b), (c), (d), (e), and (f)	(b)
p.154	Addition of 6.2.1 Timer/counter register 0n (TCR0n)	(c)
p.156	Addition of 6.2.2 Timer data register 0n (TDR0n)	(c)
p.157	Modification of 6.3 Registers Controlling Timer Array Unit and addition of Caution	(c)
p.158	Modification of Caution 1 in Figure 6-7 and addition of Note	(c)
p.159	Modification of description in 6.3.2 Timer clock select register 0 (TPS0)	(c)
p.160	Modification of Figure 6-8	(c)
p.165, 166	Modification of Figure 6-9 (4/4) and addition of Note 1	(c)
p.171	Modification of Figure 6-15	(c)
p.172	Modification of description in 6.3.10 Timer output register 0 (TO0)	(c)
p.175	Modification of description in 6.3.13 Noise filter enable register 1 (NFEN1)	(c)
p.176	Modification of description in 6.3.14 Registers controlling port functions of pins to be used for timer I/O	(c)
p.185	Modification of Figure 6-24	(b)
p.186	Modification of Figure 6-25	(b)
p.187	Modification of Figure 6-26	(b)
p.196	Addition of 6.7 Timer Input (TI0n) Control	(c)
p.200	Modification of Figure 6-40	(b)
p.202	Modification of Figure 6-41	(c)
p.205	Modification of Figure 6-44	(b)
p.207	Modification of Figure 6-45	(c)
p.211	Modification of Figure 6-49	(c)
p.212	Modification of Note in Figure 6-50	(c)

Remark “Classification” in the above table classifies revisions as follows.
(a): Error correction, (b): Addition/change of specifications, (c): Addition/change of description or note,
(d): Addition/change of package, part number, or management division, (e): Addition/change of related documents

Downloaded from Arrow.com.

(3/9)

Page	Description	Classification
p.216	Modification of Figure 6-53	(c)
p.220	Modification of Figure 6-57	(c)
p.223	Modification of Figure 6-60	(b)
p.225	Modification of Figure 6-61	(c)
p.229	Modification of Figure 6-64	(b)
p.231, 232	Modification of Figure 6-66	(c)
p.236	Modification of Figure 6-69	(b)
p.237	Modification of Figure 6-70	(b)
p.240	Modification of description in 6.9.3 Operation as multiple PWM output function	(c)
p.243	Modification of Figure 6-74	(b)
p.247	Modification of description in 6.10.1 Cautions When Using Timer output	(c)
CHAPTER 7 12-BIT INTERVAL TIMER		
p.248	Modification of description in Figure 7-1	(c)
p.249	Modification of Caution 1 in Figure 7-2	(c)
p.252	Modification of description in 7.4.2 Start of count operation and re-enter to HALT/STOP mode after returned from HALT/STOP mode	(c)
CHAPTER 8 CLOCK OUTPUT/BUZZER OUTPUT CONTROLLER		
p.253	Addition of description to the beginning of the chapter	(c)
p.253	Addition of description of 8.1 Functions of Clock Output/Buzzer Output Controller	(c)
p.254	Modification of description in 8.3 Registers Controlling Clock Output/Buzzer Output Controller	(c)
p.255	Addition of Note 1 to and modification of Note 2 in Figure 8-2	(c)
p.256	Modification of title of 8.3.2 to Registers controlling port functions of pins to be used for clock or buzzer output	(a)
p.257	Modification of description in 8.4.1 Operation as output pin	(c)
p.257	Modification of title of Figure 8-3 to Timing of Outputting Clock from PCLBUZn Pin	(a)
p.257	Addition of 8.5 Cautions of Clock Output/Buzzer Output Controller	(c)
CHAPTER 9 WATCHDOG TIMER		
p.258	Modification of description in 9.1 Functions of Watchdog Timer	(c)
p.259	Modification of Table 9-1	(c)
p.259	Modification of Figure 9-1	(c)
p.261	Modification of Caution 2 in 9.4.1 Controlling operation of watchdog timer	(c)
CHAPTER 10 A/D CONVERTER		
p.265	Modification of description in 10.1 Function of A/D Converter	(c)
p.266	Addition of description of Figure 10-1	(c)
p.269	Modification of description in 10.3 Registers Controlling A/D Converter	(c)
p.270	Modification of Caution 1 in Figure 10-2	(c)
p.271	Modification of description of ADCS bit in Figure 10-3	(c)
p.275	Modification of Table 10-3, Note, and Caution 1	(c)
p.276	Modification of Table 10-3, Note 3, and Caution 1	(c)
p.277	Modification of Table 10-3, Note 1, and Caution 1	(c)
p.278	Modification of Table 10-3, Note 1, and Caution 1	(c)
p.280	Modification of Caution 2 in Figure 10-6	(c)

Remark “Classification” in the above table classifies revisions as follows.
(a): Error correction, (b): Addition/change of specifications, (c): Addition/change of description or note,
(d): Addition/change of package, part number, or management division, (e): Addition/change of related documents

Downloaded from Arrow.com.

(4/9)

Page	Description	Classification
p.281, 282	Modification of description in Figure 10-7	(c)
p.287	Modification of Cautions 2 and 3 in Figure 10-13	(c)
p.290	Modification of description in Figure 10-15	(c)
p.312	Modification of Figure 10-37	(c)
CHAPTER 11 SERIAL ARRAY UNIT		
p.320	Modification of description in 11.1.1 3-wire serial I/O (CSI00, CSI01, CSI11, CSI20)	(c)
p.323	Modification of Table 11-1	(c)
p.324	Modification of Figure 11-1	(b)
p.325	Modification of Figure 11-2	(b)
p.326	Modification of Figure 11-3	(b)
p.327	Addition of 11.2.1 Shift register	(c)
p.327	Addition of 11.2.2 Lower 8/9 bits of the serial data register mn (SDRmn)	(c)
p.330	Modification of Note and Caution 1 in Figure 11-6	(c)
p.331	Modification of description in Figure 11-7 and Note	(c)
p.332, 333	Modification of Caution in Figure 11-8 (1/2) and (2/2)	(c)
p.334, 335	Modification of description and Caution in Figure 11-9 (1/2) and (2/2)	(c)
p.336	Modification of description in 11.3.5 Serial data register mn (SDRmn)	(c)
p.336	Modification of Figure 11-10	(c)
p.338	Modification of Remarks in Figure 11-11	(c)
p.339	Modification of Caution in Figure 11-12 (1/2)	(c)
p.340	Modification of Cautions 1 and 2 in Figure 11-12 (2/2)	(c)
p.341	Modification of Caution 1 in Figure 11-13	(c)
p.342	Modification of Caution in Figure 11-14	(c)
p.344	Modification of Caution in Figure 11-16	(c)
p.345	Modification of description in 11.3.12 Serial output register m (SOm)	(c)
p.345	Modification of Caution in Figure 11-17	(c)
p.346	Modification of Caution in Figure 11-18	(c)
p.347	Addition of Figure 11-19	(b)
p.348	Modification of Caution in 11.3.14 Serial standby control register 0 (SSC0)	(c)
p.348	Modification of Figure 11-20	(c)
p.348	Addition of Figure 11-21	(b)
p.349	Modification of description in 11.3.15 Noise filter enable register 0 (NFEN0)	(c)
p.349	Modification of Caution in Figure 11-22	(c)
p.350	Modification of description in 11.3.16 Registers controlling port functions of serial input/output pins	(c)
p.353	Modification of description of 11.5 Operation of 3-Wire Serial I/O (CSI00, CSI01, CSI11, CSI20) Communication and Note	(c)
p.354	Modification of description in 11.5.1 Master transmission and addition of Remark	(c)
p.358 to 362	Modification of Figures 11-28 to 11-32	(b), (c)
p.363	Modification of description in 11.5.2 Master reception and addition of Remark	(c)
p.365 to 367, 369, 370	Modification of Figures 11-35 to 11-37, 11-39, and 11-40	(b), (c)
p.371	Modification of description in 11.5.3 Master transmission/reception and addition of Remark	(c)
p.374, 376, 378, 379	Modification of Figures 11-43, 11-45, 11-47, and 11-48	(b), (c)
p.380	Modification of description in 11.5.4 Slave transmission, Notes 1 and 2, and Remark 2	(c)
p.381, 383 to 388	Modification of Figures 11-49 and 11-51 to 11-56	(b), (c)
p.389	Modification of description in 11.5.5 Slave reception and Notes 1 and 2	(c)
p.390, 391, 393, 394	Modification of Figures 11-57 to 11-59, 11-61, and 11-62	(b), (c)
p.395	Modification of description in 11.5.6 Slave transmission/reception and Notes 1 and 2	(c)

Remark “Classification” in the above table classifies revisions as follows.
(a): Error correction, (b): Addition/change of specifications, (c): Addition/change of description or note,
(d): Addition/change of package, part number, or management division, (e): Addition/change of related documents

Downloaded from Arrow.com.

(5/9)

Page	Description	Classification
p.398, 400 to 403	Modification of Figures 11-65 and 11-67 to 11-70	(b), (c)
p.404	Modification of description in 11.5.7 SNOOZE mode function	(c)
p.404 to 407	Modification of Figures 11-71 to 11-74	(b), (c)
p.411	Modification of description in 11.6 Operation of UART (UART0 to UART2) Communication and addition of Note 1	(c)
p.412	Modification of description in 11.6.1 UART transmission and addition of Notes 1 and 2	(c)
p.415 to 421	Modification of Figures 11-77 to 11-83	(b), (c)
p.422	Modification of description in 11.6.2 UART reception and addition of Notes 1 and 2	(c)
p.425, 427, 428	Modification of Figure 11-86, 11-88, and 11-89	(b), (c)
p.429	Modification of description in 11.6.3 SNOOZE mode function and addition of Cautions 2 to 4	(c)
p.430	Modification of Table 11-3	(c)
p.431	Modification of description in 11. 6. 3 (1) and Figure 11-90	(b), (c)
p.432	Modification of description in 11. 6. 3 (2) and Figure 11-91	(b), (c)
p.433	Modification of Figure 11-92	(b)
p.434	Modification of description in 11. 6. 3 (3) and Figure 11-93	(b), (c)
p.435	Modification of Figure 11-94	(b)
p.442	Modification of description in 11.7 Operation of Simplified I^2C (IIC00, IIC01, IIC11, IIC20) Communication	(c)
p.444	Modification of description in 11.7.1 Address field transmission and Notes 1 and 2	(c)
p.448	Modification of description in 11.7.2 Data transmission and Notes 1 and 2	(c)
p.451	Modification of description in 11.7.3 Data reception and Notes 1 and 2	(c)
p.454	Modification of Figure 11-107	(c)
p.455	Modification of Figure 11-109	(c)
p.458	Modification of Figure 11-110	(c)
CHAPTER 12 SERIAL INTERFACE IICA		
p.460	Modification of Figure 12-1	(c)
p.465	Modification of Figure 12-5	(c)
p.468, 469	Modification of Figure 12-6 (3/4) and (4/4)	(c)
p.475	Modification of Figure 12-9 (2/2)	(c)
p.476	Modification of description in 12.3.6 IICA low-level width setting register 0 (IICWL0)	(c)
p.476	Modification of description in 12.3.7 IICA high-level width setting register 0 (IICWH0)	(c)
p.479	Modification of description in 12.4.2 Setting transfer clock by using IICWL0 and IICWH0 registers	(c)
p.491, 492	Modification of Figures 12-22 and 12-23	(c)
p.493	Modification of Figure 12-24	(c)
p.494	Modification of description in 12.5.14 Communication reservation (1)	(c)
p.496	Modification of Note 1 in Figure 12-27	(c)
p.497	Modification of description in 12.5.14 Communication reservation (2)	(c)
p.498	Modification of description in 12.5.15 Cautions	(c)
p.500	Modification of Figure 12-28	(b)
p.503	Modification of Figure 12-29	(b)
p.505	Modification of Figure 12-30	(b)

Remark “Classification” in the above table classifies revisions as follows.
(a): Error correction, (b): Addition/change of specifications, (c): Addition/change of description or note,
(d): Addition/change of package, part number, or management division, (e): Addition/change of related documents

Downloaded from Arrow.com.

(6/9)

Page	Description	Classification
CHAPTER 13 MULTIPLIER AND DIVIDER/MULTIPLY-ACCUMULATOR		
p.544	Modification of Figure 13-1 and addition of Remark	(b), (c)
p.545 to 547	Addition of 13.2.1 Multiplication/division data register A (MDAH, MDAL) to 13.2.3 Multiplication/division data register C (MDCL, MDCH)	(c)
p.545 to 547	Modification of Table 13-2 to Table 13-4	(b)
p.549, 550	Modification of 13.3.1 Multiplication/division control register (MDUC) and Figure 13-5	(b)
p.551 to 553, 555, 557	Modification of description in 13.4.1 Multiplication (unsigned) operation to 13.4.5 Division operation	(b)
p.556	Modification of Figure 13-9	(a)
CHAPTER 14 DMA CONTROLLER		
p.559	Modification of description in 14.1 Functions of DMA Controller	(c)
p.570	Modification of Figure 14-9	(c)
p.575	Modification of description in 14.6 (4) DMA pending forwarding	(c)
p.576	Modification of description in 14.6 (6) Operation if instructions for accessing the data flash area	(c)
CHAPTER 15 INTERRUPT FUNCTIONS		
p.600	Modification of Figure 15-11	(c)
p.600	Modification of Figure 15-12	(c)
p.601	Modification of description in Table 15-6	(c)
CHAPTER 16 KEY INTERRUPT FUNCTION		
p.605	Addition of description to the beginning of the chapter	(c)
p.605	Modification of Table 16-1	(c)
p.606	Modification of Table 16-2	(c)
p.607	Modification of Figure 16-1	(b)
p.608	Modification of Figure 16-2	(b)
p.609	Modification of Cautions 1 and 2 in Figure 16-3	(c)
p.610	Modification of description in 16.3.3 Key return flag register (KRF) and Caution in Figure 16-4	(c)
p.610	Modification of description in 16.3.4 Port mode registers 0, 4, 6 (PM0, PM4, PM6) and Figure 16-5	(c)
p.611	Addition of description of 16.4 Key Interrupt Operation	(c)
CHAPTER 17 STANDBY FUNCTION		
p.616	Modification of Caution in 17.3.1 HALT mode	(c)
p.620	Modification of Caution in 17.3.2 STOP mode	(c)
p.622	Modification of Note 2 in Figure 17-3 (1)	(c)
p.623	Modification of Notes 1 and 2, Caution, and Remarks 1 and 2 in Figure 17-3 (3)	(c)
p.627	Modification of description in 17.3.3 SNOOZE mode (2) and (3)	(c)
CHAPTER 18 RESET FUNCTION		
p.630	Modification of description in 18.1 Timing of Reset Operation	(c)
p.632	Addition of Note to Table 18-1	(c)
p.633	Modification of Remark in Table 18-2	(c)
p.634	Modification of Caution 2 in Figure 18-4	(c)
CHAPTER 19 POWER-ON-RESET CIRCUIT		
p.637	Modification of description in 19.1 Functions of Power-on-reset Circuit and addition of Remark 2	(c)

Remark "Classification" in the above table classifies revisions as follows.
(a): Error correction, (b): Addition/change of specifications, (c): Addition/change of description or note,
(d): Addition/change of package, part number, or management division, (e): Addition/change of related documents

Downloaded from Arrow.com.

(7/9)

Page	Description	Classification
p.639	Modification of Figure 19-2. Timing of Generation of Internal Reset Signal by Power-on-reset Circuit (1) When the external reset input via $\overline{\text{RESET}}$ pin is used	(b)
p.640	Addition of description of Figure 19-2. Timing of Generation of Internal Reset Signal by Power-on-reset Circuit (3) LVD reset mode (2) LVD interrupt & reset mode (option byte 000C1H/LVIMDS1, LVIMDS0 = 1, 0)	(b)
p.641	Modification of Figure 19-2. Timing of Generation of Internal Reset Signal by Power-on-reset Circuit (3) LVD reset mode (option byte 000C1H/LVIMDS1, LVIMDS0 = 1, 1)	(b)
CHAPTER 20 VOLTAGE DETECTOR		
p.642	Modification of description in 20.1 Functions of Voltage Detector	(c)
p.643	Modification of Figure 20-1	(b)
p.644	Modification of Figure 20-2	(c)
p.645	Modification of Cautions 1 and 2 and Note 2 in Figure 20-3	(c)
p.647	Modification of Caution 2 in Figure 20-4 and addition of Remarks 2 and 3	(c)
p.648	Modification of description in 20.4.1 When used as reset mode	(c)
p.649	Modification of Figure 20-5	(c)
p.650	Modification of description in 20.4.2 When used as interrupt mode	(c)
p.651	Modification of Figure 20-6 and Note 2	(c)
p.652	Modification of description in 20.4.3 When used as interrupt and reset mode	(c)
p.653	Modification of description in Figure 20-7 (1/2)	(c)
p.655	Modification of description in Figure 20-7 (2/2)	(c)
p.656	Modification of description in Figure 20-8	(c)
p.658	Modification of item (1) in 20.5 Cautions for Voltage Detector	(c)
p.658	Addition of Figure 20-10	(c)
p.659	Modification of description of items (3) and (4) in 20.5 Cautions for Voltage Detector	(c)
CHAPTER 21 SAFETY FUNCTIONS		
p.660	Modification of description of items (6) and (7) and Remark in 21.1 Overview of Safety Functions	(c)
p.664	Modification of Caution and Remarks 2 and 4 in Figure 21-4	(c)
p.665	Modification of Figure 21-5	(c)
p.671	Modification of description in 21.3.7 A/D test function	(c)
p.673	Modification of description in 21.3.7.2 Analog input channel specification register (ADS)	(c)
p.674	Modification of Cautions 2 to 4 and 9 in 21.3.7.2 Analog input channel specification register (ADS)	(c)
CHAPTER 23 OPTION BYTE		
p.676	Addition of Caution to 23.1 Functions of Option Bytes	(c)
p.676	Modification of description in 23.1.1 User option byte (000C0H to 000C2H) and addition of Caution	(c)
p.679, 680	Modification of Figure 23-2 and addition of Remarks 2 and 3	(c)
p.681	Modification of Figure 23-3 and addition of Caution 1	(c)
p.683	Modification of description in 23.4 Setting of Option Byte and Caution	(c)

Remark “Classification” in the above table classifies revisions as follows.
(a): Error correction, (b): Addition/change of specifications, (c): Addition/change of description or note,
(d): Addition/change of package, part number, or management division, (e): Addition/change of related documents

Downloaded from Arrow.com.

(8/9)

Page	Description	Classification
CHAPTER 24 FLASH MEMORY		
p.690	Modification of description in 24.3.1 P40/TOOL0 pin and Remark 1	(c)
p.693	Addition of Table 24-5. Programming Modes and Voltages at Which Data Can Be Written, Erased, or Verified	(c)
p.696	Modification of description in Table 24-10	(c)
p.700	Modification of description in 24.7 Security Settings	(c)
p.702	Modification of description in 24.8 Data Flash	(c)
CHAPTER 25 ON-CHIP DEBUG FUNCTION		
p.704	Modification of title of 25.1 to Connecting E1 On-chip Debugging Emulator	(a)
p.704, 705	Modification of Figure 25-1 to Figure 25-3	(a)
CHAPTER 26 BCD CORRECTION CIRCUIT		
p.708	Addition of 26.2.1 BCD correction result register (BCDADJ)	(c)
CHAPTER 27 INSTRUCTION SET		
p.715	Modification of Note 2 in Table 27-5 (1/17)	(c)
p.716	Modification of Note 2 in Table 27-5 (2/17)	(c)
p.717	Modification of Note 2 in Table 27-5 (3/17)	(c)
p.718	Modification of Note 2 in Table 27-5 (4/17)	(c)
p.719	Modification of Note 2 in Table 27-5 (5/17)	(c)
p.720	Modification of Note 2 in Table 27-5 (6/17)	(c)
p.721	Modification of Note 2 in Table 27-7 (7/17)	(c)
p.722	Modification of Note 2 in Table 27-5 (8/17)	(c)
p.723	Modification of Note 2 in Table 27-5 (9/17)	(c)
p.724	Modification of Note 2 in Table 27-5 (10/17)	(c)
p.725	Modification of Note 2 in Table 27-5 (11/17)	(c)
p.726	Modification of Note 2 in Table 27-5 (12/17)	(c)
p.727	Modification of Note 2 in Table 27-5 (13/17)	(c)
p.728	Modification of Note 2 in Table 27-5 (14/17)	(c)
p.729	Modification of Note 2 in Table 27-5 (15/17)	(c)
p.730	Modification of Note 2 in Table 27-5 (16/17)	(c)
p.731	Modification of Note 2 in Table 27-5 (17/17)	(c)
CHAPTER 28 ELECTRICAL SPECIFICATIONS (A, D: T_A = −40 to +85°C)		
p.733	Modification of description of table in 28.1 Absolute Maximum Ratings (T_A = 25°C)	(c)
p.734	Modification of table, Note, and Caution in 28.2.1 X1 oscillator characteristics	(c)
p.734	Modification of table in 28.2.2 On-chip oscillator characteristics	(c)
p.735	Modification of Note 3 in 28.3.1 Pin characteristics (1/4)	(c)
p.736	Modification of Note 3 in 28.3.1 Pin characteristics (2/4)	(c)
p.739	Modification of Notes 1 and 2 in (1) 20-, 24-pin products (1/2)	(c)
p.740	Modification of Notes 1 and 3 in (1) 20-, 24-pin products (2/2)	(c)
p.741	Modification of Notes 1 and 2 in (2) 30-pin products (1/2)	(c)
p.742	Modification of Notes 1 and 3 in (2) 30-pin products (2/2)	(c)
p.743	Modification of (3) Peripheral functions (Common to all products)	(b)
p.744	Modification of table in 28.4 AC Characteristics	(b)
p.745	Addition of Minimum Instruction Execution Time during Main System Clock Operation	(b)

Remark “Classification” in the above table classifies revisions as follows.
(a): Error correction, (b): Addition/change of specifications, (c): Addition/change of description or note,
(d): Addition/change of package, part number, or management division, (e): Addition/change of related documents

Downloaded from Arrow.com.

(9/9)

Page	Description	Classification
p.746	Modification of figures of AC Timing Test Point and External Main System Clock Timing	(c)
p.747	Modification of figure of AC Timing Test Point	(c)
p.747	Modification of description and Note 2 in (1) During communication at same potential (UART mode)	(b)
p.748	Modification of description in (2) During communication at same potential (CSI mode)	(b)
p.749	Modification of description in (3) During communication at same potential (CSI mode)	(b)
p.750	Modification of description in (4) During communication at same potential (CSI mode)	(b)
p.752	Modification of table and Note 2 in (5) During communication at same potential (simplified I^2C mode)	(b)
p.754, 755	Modification of table and Notes 1 to 9 in (6) Communication at different potential (1.8 V, 2.5 V, 3 V) (UART mode)	(b)
p.756	Modification of Remarks 1 to 3 in (6) Communication at different potential (1.8 V, 2.5 V, 3 V) (UART mode)	(b)
p.757	Modification of table in (7) Communication at different potential (2.5 V, 3 V) (CSI mode)	(b)
p.758	Modification of Caution in (7) Communication at different potential (2.5 V, 3 V) (CSI mode)	(b)
p.759	Modification of table in (8) Communication at different potential (1.8 V, 2.5 V, 3 V) (CSI mode) (1/3)	(b)
p.760	Modification of table and Notes 1 and 2 in (8) Communication at different potential (1.8 V, 2.5 V, 3 V) (CSI mode) (2/3)	(b)
p.761	Modification of table, Note 1, and Caution 1 in (8) Communication at different potential (1.8 V, 2.5 V, 3 V) (CSI mode) (3/3)	(b)
p.763	Modification of table in (9) Communication at different potential (1.8 V, 2.5 V, 3 V) (CSI mode)	(b)
p.766	Modification of table, Note 1, and Caution 1 in (10) Communication at different potential (1.8 V, 2.5 V, 3 V) (simplified I^2C mode)	(b)
p.768	Modification of Remark in 28.5.2 Serial interface IICA	(b)
p.769	Addition of table to 28.6.1 A/D converter characteristics	(c)
p.769	Modification of description in 28.6.1 (1)	(c)
p.770	Modification of Notes 3 to 5 in 28.6.1 (1)	(c)
p.770	Modification of description and Notes 2 to 4 in 28. 6. 1 (2)	(c)
p.771	Modification of description and Notes 3 and 4 in 28. 6. 1 (3)	(c)
p.772	Modification of description and Notes 3 and 4 in 28. 6. 1 (4)	(c)
p.773	Modification of table in 28.6.2 Temperature sensor/internal reference voltage characteristics	(c)
p.773	Modification of table and Note in 28.6.3 POR circuit characteristics	(c)
p.774	Modification of table in 28.6.4 LVD circuit characteristics	(c)
p.775	Modification of table of LVD detection voltage of interrupt & reset mode	(c)
p.775	Modification of number and title to 28.6.5 Power supply voltage rising slope characteristics	(c)
p.777	Modification of table, figure, and Remark in 28.10 Timing of Entry to Flash Memory Programming Modes	(c)
CHAPTER 29 ELECTRICAL SPECIFICATIONS (G: T_A = −40 to +105°C)		
p.778 to 819	Addition of products of industrial applications (G: T_A = -40 to +105°C)	(b)
CHAPTER 30 PACKAGE DRAWINGS		
p.820 to 822	Addition of products of industrial applications (G: T_A = -40 to +105°C)	(b)

Remark “Classification” in the above table classifies revisions as follows.
(a): Error correction, (b): Addition/change of specifications, (c): Addition/change of description or note,
(d): Addition/change of package, part number, or management division, (e): Addition/change of related documents

Downloaded from Arrow.com.

A.2 Revision History of Preceding Editions

Here is the revision history of the preceding editions. Chapter indicates the chapter of each edition.

(1/10)

Edition	Description	Chapter
1.10	Deletion of overbars for SCK(serial clock signal)	Throughout
	Titles of register description headlined	
	Addition of 1.1 Differences between R5F102 and R5F103	CHAPTER 1 OUTLINE
	Modification of description and Notes 1 to 3 and addition of Caution in 1.7 Outline of Functions	
	Modification of Note 1 and Caution 1 and addition of Cautions 2 and 3 in Figure 3-1. Memory Map	CHAPTER 3 CPU ARCHITECTURE
	Modification of Caution in Figure 3-2 to 3-6. Memory Maps	
	Modification of description and Cautions 1 and 2 and addition of Caution 3 in 3.1.3 Internal data memory space	
	Modification of Note 1 and Caution 1 and addition of Cautions 2 and 3 in Figure 3-8. Correspondence Between Data Memory and Addressing	
	Modification of Caution in Figure 3-9 to Figure 3-13. Correspondence Between Data Memory and Addressing	
	Modification of Caution 3 and addition of Caution 4 in 3.2.1 (3) Stack pointer (SP)	
	Modification of Caution 2 and addition of Caution 3 in 3.2.2 General-purpose registers	
	Modification of Figure 3-27 and Figure 3-28	
	Modification of Figure 3-31 to Figure 3-41	
	Addition of Figure 3-42 to Figure 3-47	
	Modification of Cautions 1 to 3 in 5.3.8 High-speed on-chip oscillator frequency selection register (HOCODIV)	CHAPTER 5 CLOCK GENERATOR
	Addition of 5.7 Resonator and Oscillator Constants	
	Modification of description in (2) Timer data register 0n (TDR0n)	CHAPTER 6 TIMER ARRAY UNIT
	Modification of Caution 1 and addition of Caution 2 in 10.3.2 A/D converter mode register 0 (ADM0)	CHAPTER 10 A/D CONVERTER
	Modification of Caution 1 and LV1 in Tables 10-3 A/D Conversion Time Selection (2/4) and (4/4)	
	Modification of description and deletion of Note in Table 18-1. Operation Statuses During Reset Period	CHAPTER 18 RESET FUNCTION
	Modification of Caution 2 in 18.1.1 Reset Control Flag Register (RESF)	
	Modification of Note 3 in Figure 19-2. Timing of Generation of Internal Reset Signal by Power-on-reset Circuit and Voltage Detector (1/3)	CHAPTER 19 POWERON-RESET CIRCUIT
	Modification of Note 4 in Figure 19-2. Timing of Generation of Internal Reset Signal by Power-on-reset Circuit and Voltage Detector (2/3)	
	Addition of Figure 19-2. Timing of Generation of Internal Reset Signal by Power-on-reset Circuit and Voltage Detector (3/3)	
	Modification of 21.1 (2) RAM parity error detection function	CHAPTER 21 SAFETY FUNCTIONS
	Modification of Caution in 21.3.2 RAM parity error detection function	
	Modification of description and addition of Cautions 1 and 2 in 24.4.1 Data flash overview	CHAPTER 24 FLASH MEMORY
	Modification of description in 24.6 Security Settings	
	Deletion of (2) Self programming in Table 24-12. Security Setting in Each Programming Mode	
	Modification of Cautions in 24.7 Flash Memory Programming by Self-Programming	
	Modification of description in 24.7.1 Flash shield window function	
	Modification of description in Table 24-14. Setting and changing of the flash shield window function and relations with commands	
	Modification of description in 28.8 Flash Memory Programming Characteristics	CHAPTER 28 ELECTRICAL SPECIFICATIONS

Downloaded from Arrow.com.

(2/10)

Edition	Description	Chapter
1.00	Addition of products of industrial application	Throughout
	Renamed interval timer (unit) to 12-bit interval timer	
	Addition of pin name of the peripheral I/O redirection function	
	Modification of reset processing time	
	Deletion of LIN communication function	
	Renamed V_{LVI}, V_{LVIH}, V_{LVIL} to V_{LVD}, V_{LVDH}, V_{LVDL} (LVD detection voltage)	
	Renamed RAMTOP to RPE, renamed ITIF, ITMK, ITKAPR0, ITKAPR1 to TMKAIF, TMKAMK, TMKAPR0, TMKAPR1 (interrupt source, flag)	
	Addition of description to 1.1 Features	CHAPTER 1 OUTLINE
	Modification of description in 1.2 Ordering Information	
	Addition of Figure 1-1. Part Number, Memory Size, and Package of RL78/G12	
	Addition and Modification of description in 1.6 Outline of Functions	
	Modification of description in 2.1 Port Function	CHAPTER 2 PIN FUNCTIONS
	Modification of description in 2.2 Functions other than port pins (Deletion of description of port function)	
	Modification of description in 2.3 Pin I/O Circuits and Recommended Connection of Unused Pins	
	Addition of remark to Table 3-1. Correspondence Between Address Values and Block Numbers in Flash Memory	CHAPTER 3 CPU ARCHITECTURE
	Addition of product in Table 3-2. Internal ROM Capacity	
	Addition of INTFL to Table 3-3. Vector Table (20-, 24-pin products)	
	Modification of description in 3.1.2 Mirror area	
	Addition of description to Caution in Table 3-5. Internal RAM Capacity	
	Modification of Figure 3-23. Outline of Table Indirect Addressing	
	Addition of setting of registers when using port xx to Table 4-2 to 4-4, 4-6 to 4-12, 4-14 to 4-20	CHAPTER 4 PORT FUNCTIONS
	Modification of block diagrams for Pxxx	
	Addition of description to (3) Port 2	
	Addition of description to (4) Port 4	
	Addition of Note to (6) Port 12	
	Addition of description to (3) Port 2	
	Addition of description to (2) Port register (Pxx)	
	Addition of description to (3) Pull-up resistor option registers (PUxx)	
	Modification of description in Figure 4-37. Format of Port Output Mode Register	
	Addition of Caution to Figure 4-38. Format of Port Mode Control Register	
	Addition of description to (8) Peripheral I/O redirection register (PIOR)	
	Addition of description to 4.4.1 Writing to I/O port, and 4.4.3 Operations on I/O port	
	Addition of description to 4.4.4 Connecting to external device with different potential (1.8 V, 2.5 V, 3 V)	
	Addition of description to 4.5 Settings of Port Mode Register, and Output Latch When Using Alternate Function	
	Addition of 4.6.2 Notes on specifying the pin settings	
	Addition of description to 5.1 (1) <2> High-speed on-chip oscillator	CHAPTER 5 CLOCK GENERATOR
	Addition of Caution to Figure 5-2. Format of Clock Operation Mode Control Register (CMC)	
	Modification of Figure 5-1. Block Diagram of Clock Generator	
	Modification of description in Table 5-2. Condition Before Stopping Clock Oscillation and Flag Setting	
	Deletion of Note to Figure 5-7. Format of Peripheral Enable Register 0 (PER0) (1/2)	
	Addition of description to (7) Operation speed mode control register (OSMC)	
	Modification of Caution 3 to Figure 5-9 Format of High-Speed On-Chip Oscillator Frequency Selection Register (HOCODIV)	
	Modification of description in Figure 5-13. Clock Generator Operation When Power Supply Voltage Is Turned On (When voltage detector (LVD) is used)	
	Addition of description to 5.6.2 Example of setting X1 oscillation clock	

Downloaded from Arrow.com.

(3/10)

Edition	Description	Chapter
1.00	Addition of description to Figure 5-14. CPU Clock Status Transition Diagram	CHAPTER 5 CLOCK GENERATOR
	Addition of description to (2) CPU operating with high-speed system clock (C) after reset release (A)	
	Modification and deletion of description in Table 5-4. Changing CPU Clock	
	Modification of description in Table 5-5. Maximum Number of Clocks Required for $f_{IH} \leftrightarrow f_{MX}$, and Table 5-6. Conditions Before the Clock Oscillation Is Stopped and Flag Settings	
	Addition of Figure to (7) Delay counter	CHAPTER 6 TIMER ARRAY UNIT
	Addition of description to Figure 6-2. Entire Configuration of Timer Array Unit (30-pin products)	
	Addition of Figure 6-3. Internal Block Diagram of Channel of Timer Array Unit	
	Addition of description to (1) Timer/counter register 0n (TCR0n)	
	Addition of description to (2) Timer data register 0n (TDR0n)	
	Modification of description and addition of caution to Figure 6-8. Format of Timer Clock Select register 0 (TPS0)	
	Modification of description in Table 6-4. Interval Times Available for Operation Clock CKS02 or CKS03, and addition of Caution in (3) Timer mode register 0n (TMR0n)	
	Modification of Figure 6-9. Format of Timer Mode Register 0n (TMR0n)	
	Addition of description to (5) Timer channel enable status register 0 (TE0)	
	Modification of Figure 6-12. Format of Timer Channel Start register 0 (TS0)	
	Addition of description to Figure 6-13. Format of Timer Channel Stop register 0 (TT0)	
	Addition of description to (8) Timer input select register 0 (TIS0)	
	Modification of description to Figure 6-15. Format of Timer Output Enable register 0 (TOE0)	
	Addition of description to (14) Port mode registers 0, 1, 3, or 4 (PM0, PM1, PM3, or PM4)	
	Addition of description to 6.4.1 Basic Rules of Simultaneous Channel Operation Function	
	Addition of description to 6.5.1 Count clock (f_{TCLK})	
	Modification of description to Table 6-6. Operations from Count Operation Enabled State to Timer count Register 0n (TCR0n) Count Start	
	Addition of title to 6.5.3 Operation of counter	
	Modification of Figure 6-27. Operation Timing (In Capture & One-count Mode: High-level Width Measurement)	
	Addition of description to 6.6.2 TO0n Pin Output Setting	
	Modification of description to Figure 6-39, 43, 51, 55, 59, 64, 69, 74 Example of Set Contents of Registers	
	Modification of Figure 6-41, 45, 49, 53, 61	
	Addition of 6.9 Cautions When Using Timer Array Unit	
	Addition of description to (2) Operation speed mode control register (OSMC)	CHAPTER 7 INTERVAL TIMER
	Addition of Caution to Figure 7-4. Format of Interval Timer Control Register (ITMC)	
	Modification of Figure 7-5. 12-Bit Interval Timer Operation Timing	
	Modification of Figure 8-1. Block Diagram of Clock Output/Buzzer Output Controller	CHAPTER 8 CLOCK OUTPUT/BUZZER OUTPUT CONTROLLER
	Modification of Figure 8-2. Format of Clock Output Select Register n (CKSn)	
	Addition of description to (2) Port mode register 1, 3 (PM1, PM3)	
	Addition of Caution to 8.4.1 Operation as output pin	
	Modification of description to 9.1 Functions of Watchdog Timer, 9.4.4 Setting watchdog timer interval interrupt	CHAPTER 9 WATCHDOG TIMER
	Modification of Figure 10-1. Block Diagram of A/D Converter	CHAPTER 10 A/D CONVERTER
	Modification of description to (3) A/D voltage comparator	
	Addition of Caution to Figure 10-3. Format of A/D Converter Mode Register 0 (ADM0)	
	Modification of description to Figure 10-4. Timing Chart When A/D Voltage Comparator Is Used	
	Addition of description to Table 10-3. A/D Conversion Time Selection	
	Modification of Caution to Figure 10-6. Format of A/D Converter Mode Register 1 (ADM1)	

Downloaded from Arrow.com.

(4/10)

Edition	Description	Chapter
1.00	Addition of description to Figure 10-7. Format of A/D Converter Mode Register 2 (ADM2)	CHAPTER 10 A/D CONVERTER
	Addition of description to Figure 10-11. Format of Analog Input Channel Specification Register (ADS)	
	Addition of description and Caution to (10) A/D test register (ADTES), (11) A/D port configuration register (ADPC), (12) Port mode control registers 0, 1, 4, 12, and 14 (PMC0, PMC1, PMC4, PMC12, and PMC14), and (13) Port mode registers 0, 1, 2, 4, 12, and 14 (PM0, PM1, PM2, PM4, PM12 and PM14)	
	Addition of Note to 10.4 A/D Converter Conversion Operations	
	Modification of Figure 10-32. Setting up Software Trigger Mode to Figure 10-36. Setting up Test Trigger Mode	
	Addition of description to 10.8 SNOOZE mode function	
	Addition and modification of description to 10.10 Cautions for A/D Converter	
	Change the value to Table 10-6. Resistance and Capacitance Values of Equivalent Circuit (Reference Values)	
	Addition of description to CHAPTER 11 SERIAL ARRAY UNIT	CHAPTER 11 SERIAL ARRAY UNIT
	Addition of description to 11.1.2 UART (UART0 to UART2)	
	Modification of Figure 11-1 to 11-3. Block Diagram of Serial Array Unit 0 (20- or 24-pin products)	
	Modification of Caution to Figure 11-6. Format of Peripheral Enable Register 0 (PER0)	
	Modification of frequency to Figure 11-7. Format of Serial Clock Select Register m (SPSm)	
	Addition of description for Note to Figure 11-9. Format of Serial Communication Operation Setting Register mn (SCRmn)	
	Addition of description to Figure 11-10. Format of Serial Data Register mn (SDRmn)	
	Addition of description to Figure 11-12. Format of Serial Status Register mn (SSRmn) (2/2)	
	Addition and modification of Note and Caution to Figure 11-13. Format of Serial Channel Start Register m (SSm)	
	Addition and modification of Note to Figure 11-14. Format of Serial Channel Stop Register m (STm), Figure 11-15. Format of Serial Channel Enable Status Register m (SEm)	
	Addition of description to Figure 11-16. Format of Serial Output Enable Register m (SOEm) and Figure 11-17. Format of Serial Output Register m (SOm)	
	Addition of description to (13) Serial output level register m (SOLm)	
	Modification of description to Figure 11-19. Format of Serial Standby Control Register 0 (SSC0)	
	Addition of description to (18) Port mode registers 0, 1, 3 to 6 (PM0, PM1, PM3 to PM6)	
	Addition of description to Figure 11-25. Each Register Setting When Stopping Operation by Channels	
	Modification of description to 11.5.1 Master transmission, 11.5.2 Master reception, 11.5.3 Master transmission/reception	
	Modification of description to Figure 11-26, 34, 42, 50, 58, 64, 77, 85, 99, 103, 106 (Example of Contents of Registers)	
	Modification of description to Figure 11-28, 29, 31, 33, 36, 37, 39, 41, 44, 45, 47, 49, 52, 53, 55, 57, 60, 61, 63, 66, 69, 71, 73, 75, 79, 80, 82, 84, 86, 87, 88, 90, 93, 95, 100, 102, 105, 108 (flow chart)	
	Addition of description of notes to 11.5.4 Slave transmission, 11.5.5 Slave reception, 11.5.6 Slave transmission/reception	
	Addition of Caution to 11.5.7 SNOOZE mode function (only CSI00), 11.6.3 SNOOZE mode function (only UART0 reception)	
	Addition of Caution to 11.6 Operation of UART (UART0 to UART2) Communication	
	Modification of description to 11.7.1 Address field transmission, 11.7.2 Data transmission, 11.7.3 Data reception	
	Addition of Caution to 11.7.5 Calculating transfer rate	
	Modification of description for example of setting IIC transfer rate	

Downloaded from Arrow.com.

(5/10)

Edition	Description	Chapter
1.00	Modification of description to Figure 12-6. Format of IICA Control Register 00 (IICCTL00)	CHAPTER 12 SERIAL INTERFACE IICA
	Addition of description to Figure 12-7. Format of IICA Status Register 0 (IICS0)	
	Modification of Figure 12-28, 29, 30	
	Modification of Figure 13-1. Block Diagram of Multiplier and Divider/Multiply-Accumulator	CHAPTER 13 MULTIPLIER AND DIVIDER/MULTIPLY ACCUMULATOR
	Modification of value to Figure 13-6. Timing Diagram of Multiplication (Unsigned) Operation (2 × 3 = 6)	
	Addition of description to 13.4.5 Division operation	
	Addition of description	CHAPTER 15 INTERRUPT FUNCTION
	Addition of description to Table 15-1 and 15-2. Interrupt Source List	
	Addition of INTFL to Table 15-3 and 15-4. Flags Corresponding to Interrupt Request Sources	
	Modification of description to Table 15-5. Time from Generation of Maskable Interrupt Until Servicing	
	Modification of Figure 15-12. Interrupt Request Acknowledgment Timing (Maximum Time)	
	Modification of Table 15-6. Relationship Between Interrupt Requests Enabled for Multiple Interrupt Servicing During Interrupt Servicing	
	Addition and modification of description to Though out	CHAPTER 16 KEY INTERRUPT FUNCTION
	Modification of Caution to (3) SNOOZE mode	CHAPTER 17 STANDBY FUNCTION
	Modification of description to Table 17-1. Operating Statuses in HALT Mode	
	Addition and modification of release to standby function, wait time for SNOOZE status	
	Addition of description to Table 18-1. Operation Statuses During Reset Period	CHAPTER 18 RESET FUNCTION
	Addition of description to Table 18-2. Hardware Statuses After Reset Acknowledgment	
	Addition of Note to Table 18-2. Hardware Statuses After Reset Acknowledgment	
	Addition and modification of 19.1 Functions of Power-on-reset Circuit, 19.3 Operation of Power-on-reset Circuit	CHAPTER 19 POWERON-RESET CIRCUIT
	Modification of Figure 20-1. Block Diagram of Voltage Detector	CHAPTER 20 VOLTAGE DETECTOR
	Modification of description to Figure 20-2. Format of Voltage Detection Register (LVIM)	
	Addition of description to Figure 20-3. Format of Voltage Detection Level Select Register (LVIS)	
	Addition of Caution to Table 20-1. LVD Operation Mode and Detection Voltage Settings for User Option Byte (000C1H)	
	Modification of Figure 20-4, 20-5, 20-6	
	Addition of description to Figure 20-7, 20-8	
	Addition of description to 21.3.1 CRC operation function (general-purpose CRC)	CHAPTER 21 SAFETY FUNCTIONS
	Addition of description to Figure 21-3. CRC Operation Function (General-Purpose CRC)	
	Addition of description to Figure 21-5. Format of Invalid Memory Access Detection Control Register (IAWCTL)	
	Modification of Figure 21-7. Invalid access detection area	
	Addition of description to 21.3.7 A/D test function	
	Addition of Figure (moved from description of CHAPTER 2 PIN FUNCTIONS of the previous edition)	CHAPTER 22 REGULATOR
	Addition of description to (2) 000C1H	CHAPTER 23 OPTION BYTE
	Addition of description to Figure 23-1. Format of User Option Byte (000C0H)	
	Modification and addition of Caution to Figure 23-2. Format of User Option Byte (000C1H)	
	Deletion of description to 24.1.1 Programming environment	CHAPTER 24 FLASH MEMORY
	Addition of description to 24.2 Writing to Flash Memory by Using External Device (that Incorporates UART)	
	Addition of description to Figure 24-8. Setting of Flash Memory Programming Mode	
	Addition of description to Table 24-5, Table 24-13. Programming Modes and Voltages at Which Data Can Be Written, Erased, or Verified	

Downloaded from Arrow.com.

(6/10)

Edition	Description	Chapter
1.00	Modification of description to Table 24-10. Example of Signature Data	CHAPTER 24 FLASH MEMORY
	Addition of description to 24.6 Security Settings	
	Addition of Figure 25-3. Connection Example of E1 On-chip Debugging Emulator and RL78/G12 (30-pin products)	CHAPTER 25 ON-CHIP DEBUG FUNCTION
	Modification of description to Figure 25-4. Memory Spaces Where Debug Monitor Programs Are Allocated	
	Modification of flag status	CHAPTER 27 INSTRUCTION SET
	Deletion of target, and public release	CHAPTER 28 ELECTRICAL SPECIFICATIONS
0.03	Deletion of temperature trimming registers 0 to 3 (TEMPCAL0 to TEMPCAL3)	Throughout
	Addition of description to 1.1 Features	CHAPTER 1 OUTLINE
	Addition of description to 1.6 Outline of Functions	
	Change of description for 2.1.4 Pins for each product (pins other than port pins)	CHAPTER 2 PIN FUNCTIONS
	Addition of 2.2 Description of Pin Functions	
	Addition of note in Figures 3-3 to 3-5 Memory Map, Correspondence Between Data Memory and Addressing	CHAPTER 3 CPU ARCHITECTURE
	Addition of description to 3.1.2 Mirror area	
	Addition of caution 2 to 3.2.2 General-purpose registers	
	Change of description in Table 3-7. Extended SFR (2nd SFR) List (2/5)	
	Addition of description to 3.4.3 Direct addressing	
	Change of Table 4-5. PMxx, Pxx, PUxx, PIMx, POMx, PMCxx Registers and the Bits (30-pin Products) and notation in Figure 4-32. Format of Pull-up Resistor Option Register	CHAPTER 4 PORT FUNCTIONS
	Addition of 4.4 Port Function Operations	
	Change of setting value in Table 4-6. Settings of Port Mode Register and Output Latch When Using Alternate Function and Table 4-9. Settings of Port Mode Register and Output Latch When Using Alternate Function (30-pin products)	
	Addition of value to 5.1 (1) <2> High-speed on-chip oscillator (HOCO)	CHAPTER 5 CLOCK GENERATOR
	Change of Figure 5-1. Block Diagram of Clock Generator	
	Addition of description to 5.3 (2) System clock control register (CKC)	
	Change of value in Figure 5-4. Format of Clock Operation Status Control Register (CSC)	
	Change of description in 5.3 (4) Oscillation stabilization time counter status register (OSTC)	
	Addition of value to Figure 5-6. Format of Oscillation Stabilization Time Select Register (OSTS)	
	Addition of description to 5.3 (6) Peripheral enable register 0 (PER0)	
	Addition of description to 5.3 (9) High-speed on-chip oscillator trimming register (HIOTRM)	
	Addition of note 3 to Figure 5-13. Clock Generator Operation When Power Supply Voltage Is Turned On	
	Addition of 5.6.1 Example of setting high-speed on-chip oscillator	
	Addition of description to 5.6.2 Example of setting X1 oscillation clock	
	Addition of description to 5.6.3 CPU clock status transition diagram	
	Addition of X1 clock to Table 5-4. Changing CPU Clock	
	Addition of description to 5.6.5 Time required for switchover of CPU clock and main system clock Change of description in Table 5-6. Conditions Before the Clock Oscillation Is Stopped and Flag Settings	
	Addition of 6.1.3 8-bit timer operation function (channels 1 and 3 only)	CHAPTER 6 TIMER ARRAY UNIT
	Addition of description of caution 1 to Figure 6-7. Format of Peripheral Enable Register 0 (PER0)	
	Addition of remark to Figure 6-8. Format of Timer Clock Select register 0 (TPS0)	
	Addition of address and change of description of note 2 in Figure 6-9. Format of Timer Mode Register 0n (TMR0n)	
	Addition of address to Figure 6-10. Format of Timer Status Register 0n (TSR0n)	
	Addition of 30-pin products to Figure 6-11. Format of Timer Channel Enable Status register 0 (TE0) to Figure 6-13. Format of Timer Channel Stop register 0 (TT0), and Figure 6-15. Format of Timer Output Enable register 0 (TOE0)	

Downloaded from Arrow.com.

(7/10)

Edition	Description	Chapter
0.03	Change of description in Figure 6-17. Format of Timer Output Level register 0 (TOL0)	CHAPTER 6 TIMER ARRAY UNIT
	Addition of description to 6.4.1 Basic Rules of Simultaneous Channel Operation Function	
	Addition of 6.5 Operation Timing of Counter	
	Change of description in 6.6.1 TO0n pin output circuit configuration and 6.6.2 TO0n Pin Output Setting	
	Addition of 6.6.3 Cautions on Channel Output Operation	
	Addition of 6.6.4 Collective manipulation of TO0n bit	
	Addition of 6.6.5 Timer Interrupt and TO0n Pin Output at Operation Start	
	Addition of 6.7 Independent Channel Operation Function of Timer Array Unit	
	Addition of 6.8 Simultaneous Channel Operation Function of Timer Array Unit	
	Change of description and addition of caution to Figure 7-2. Format of Peripheral Enable Register 0 (PER0)	CHAPTER 7 INTERVAL TIMER
	Addition of caution to Figure 7-4. Format of Interval Timer Control Register (ITMC)	
	Change of Figure 8-1. Block Diagram of Clock Output/Buzzer Output Controller	CHAPTER 8 CLOCK OUTPUT/BUZZER OUTPUT CONTROLLER
	Addition of frequency to Figure 8-2. Format of Clock Output Select Register n (CKSn)	
	Change of values in 9.4.3 Setting window open period of watchdog timer	CHAPTER 9 WATCHDOG TIMER
	Change of internal reference voltage	CHAPTER 10 A/D CONVERTER
	Change of Figure 10-1. Block Diagram of A/D Converter	
	Addition of caution to Table 10-1. Settings of ADCS and ADCE Bits Change of description in Table 10-2. Setting and Clearing Conditions for ADCS Bit	
	Addition of frequency to Table 10-3. A/D Conversion Time Selection	
	Change of Figure 10-6. Format of A/D Converter Mode Register 1 (ADM1)	
	Change of bit name in Figure 10-7. Format of A/D Converter Mode Register 2 (ADM2)	
	Change of caution 5 in Figure 10-11. Format of Analog Input Channel Specification Register (ADS)	
	Change of description of note 2 in 10.8 SNOOZE mode function	
	Addition of 10.10 Cautions for A/D Converter	
	Addition of description to CHAPTER 11 SERIAL ARRAY UNIT	CHAPTER 11 SERIAL ARRAY UNIT
	Addition of description to 11.1.1 3-wire serial I/O (CSI00, CSI01, CSI11, CSI20)	
	Addition of description to 11.1.2 UART (UART0 to UART2)	
	Addition of description to 11.1.3 Simplified I^2C (IIC00, IIC01, IIC11, IIC20)	
	Addition of SSC1 to Figure 11-3. Block Diagram of Serial Array Unit 1 (30-pin products)	
	Addition of value to Figure 11-7. Format of Serial Clock Select Register m (SPSm)	
	Change of address in Figure 11-10. Format of Serial Data Register mn (SDRmn)	
	Addition of description to 11.3 (14) Serial standby control register m (SSCm)	
	Addition of description to 11.3 (16) Noise filter enable register 0 (NFEN0)	
	Addition of description to 11.5 Operation of 3-Wire Serial I/O (CSI00, CSI01, CSI11, CSI20) Communication	
	Change of value in (c) Serial communication operation setting register mn (SCRmn) of Figure 11-86	
	Change of value of remark 1 in 11.6.4 (1) Baud rate calculation expression	
	Change of description and addition of caution to Figure 12-9. Format of IICA Control Register 01 (IICCTL01)	CHAPTER 12 SERIAL INTERFACE IICA
	Addition of description to 12.3 (6) IICA low-level width setting register 0 (IICWL0)	
	Change of description in Figure 12-32. Example of Master to Slave Communication and Figure 12-33. Example of Slave to Master Communication	
	Change of value in Figure 13-6. Timing Diagram of Multiplication (Unsigned) Operation (2 × 3 = 6)	CHAPTER 13 MULTIPLIER AND DIVIDER/MULTIPLY ACCUMULATOR
	Addition of description to 13.4.3 Multiply-accumulation (unsigned) operation	

Downloaded from Arrow.com.

(8/10)

Edition	Description	Chapter
0.03	Addition of description to 13.4.4 Multiply-accumulation (signed) operation	CHAPTER 13 MULTIPLIER AND DIVIDER/MULTIPLY ACCUMULATOR
	Change of Figure 13-9. Timing Diagram of Multiply-Accumulation (signed) Operation	
	Change of description in 14.2 (2) DMA RAM address register n (DRAn)	CHAPTER 14 DMA CONTROLLER
	Addition of description to Figure 14-4. Format of DMA Mode Control Register n (DMCn)	
	Addition of 14.5 Example of Setting of DMA Controller	
	Addition of 14.6 Cautions on Using DMA Controller	
	Addition of 30-pin products to Figure 15-8. Format of External Interrupt Rising Edge Enable Register (EGP0) and External Interrupt Falling Edge Enable Register (EGN0)	CHAPTER 15 INTERRUPT FUNCTION
	Addition of value to Figure 17-2. Format of Oscillation Stabilization Time Select Register (OSTS)	CHAPTER 17 STANDBY FUNCTION
	Addition of description to Table 17-1. Operating Statuses in HALT Mode to Table 17-3. Operating Statuses in SNOOZE Mode	
	Addition of note in Figure 17-3. HALT Mode Release by Interrupt Request Generation to Figure 17-6. STOP Mode Release by Reset	
	Change of value and hardware name in Table 18-2. Hardware Statuses After Reset Acknowledgment	CHAPTER 18 RESET FUNCTION
	Addition of name and value of note 2 to Table 18-2. Hardware Statuses After Reset Acknowledgment (3/3)	
	Addition of note to Figure 19-2. Timing of Generation of Internal Reset Signal by Power-on-reset Circuit and Voltage Detector	CHAPTER 19 POWERON-RESET CIRCUIT
	Change of value in Figure 19-3. Example of Software Processing After Reset Release	
	Deletion of description of (c) Interrupt mode (option byte LVIMDS1, LVIMDS0 = 0, 1)	CHAPTER 20 VOLTAGE DETECTOR
	Addition of note to Figure 20-2. Format of Voltage Detection Register (LVIM)	
	Change of note 1 to Figure 20-3. Format of Voltage Detection Level Select Register (LVIS)	
	Change of Table 20-1. LVD Operation Mode and Detection Voltage Settings for User Option Byte (000C1H)	
	Change of description in 20.4.1 When used as reset mode to 20.4.3 When used as interrupt and reset mode	
	Change of Figure 20-4. Timing of Voltage Detector Internal Reset Signal Generation (Option Byte LVIMDS1, LVIMDS0 = 1, 1) to Figure 20-6. Timing of Voltage Detector Reset Signal and Interrupt Signal Generation (Option Byte LVIMDS1, LVIMDS0 = 1, 0)	
	Change of Figure 20-8. Delay from the time LVD reset source is generated until the time LVD reset has been generated or released	
	Change of all	CHAPTER 21 SAFETY FUNCTIONS
	Change of Table 22-1. Regulator Output Voltage Conditions	CHAPTER 22 REGULATOR
	Deletion of description of 23.1.1 (2) 000C1H	CHAPTER 23 OPTION BYTE
	Change of description in Figure 23-2. Format of User Option Byte (000C1H) and Figure 23-3. Format of Option Byte (000C2H)	
	Change of setting value in 23.4 Setting of Option Byte	
	Addition of description to 24.1.2 Communication mode	CHAPTER 24 FLASH MEMORY
	Change of description in Table 24-2. Pin Connection	
	Change of description in 24.2.2 Communication mode	
	Addition of description to 24.4.1 Data flash overview	
	Change of description in 24.5.2 Flash memory programming mode	
	Addition of 24.5.5 Description of signature data	
	Change of description in (2) Self programming of Table 24-12. Security Setting in Each Programming Mode	
	Addition of description to 24.7.1 Flash shield window function Change of description in Figure 24-14. Setting and changing of the flash shield window function and relations with commands	
	Change of Figure 25-1. Connection Example of E1 On-chip Debugging Emulator and RL78/G12	CHAPTER 25 ON-CHIP DEBUG FUNCTION

Downloaded from Arrow.com.

(9/10)

Edition	Description	Chapter
0.03	Addition of Figure 25-2. Connection Example of E1 On-chip Debugging Emulator and RL78/G12 (When using to the alternative function of $\overline{\text{RESET}}$ pin)	CHAPTER 25 ON-CHIP DEBUG FUNCTION
	Change of description in Figure 25-3. Memory Spaces Where Debug Monitor Programs Are Allocated	
	Addition of description to Table 27-1. Operand Identifiers and Specification Methods	CHAPTER 27 INSTRUCTION SET
	Addition of value to 28.3.2 On-chip oscillator characteristics	CHAPTER 28 ELECTRICAL SPECIFICATIONS (target)
	Change of 28.4.1 Pin characteristics	
	Addition of value to 28.4.2 Supply current characteristics	
	Change of value in 28.5.1 Basic operation	
	Change of value in 28.6.1 Serial array unit	
	Addition of value to 28.7.1 A/D converter characteristics	
	Addition of value to 28.7.2 Temperature sensor characteristics and 28.7.3 POR circuit characteristics	
	Deletion of value of LVD detection voltage of interrupt & reset mode	
	Change of value in 28.8 Data Memory STOP Mode Low Supply Voltage Data Retention Characteristics	
	Addition of all	CHAPTER 29 PACKAGE DRAWINGS
0.02	Addition of the 30-pin product	Through out
	Modification of the names "internal high-speed oscillator circuit" and "internal low speed oscillator circuit" to "high-speed on-chip oscillator (HOCO)" and "low-speed on-chip oscillator (LOCO)"	
	Addition of cautions in Figures 3-1 through 3-6 Memory maps and Figure 3-8 through 3-13 Correspondences between data memory and addressing	CHAPTER 3 CPU ARCHITECTURE
	Modification of reset values of P13 in Table 3-6 SFR list	
	Addition of high-speed on-chip oscillator frequency selecting register (HOCODIV) to Table 3-7 Extended SFR (2ndSFR) list	
	Modification of description method of operand	
	Addition of Cautions 1 through 3 in 4.2.1 20-, 24-pin product (2) Port 1	CHAPTER 4 PORT FUNCTIONS
	Addition of Caution 2 in 4.2.1 20-, 24-pin product (4) Port 4	
	Addition of Caution in 4.2.1 20-, 24-pin product (5) Port 6	
	Addition of the high-speed on-chip oscillator frequency selection register (HOCODIV) in control registers	CHAPTER 5 CLOCK GENERATOR
	Deletion of LV (low-voltage main) mode in 5.6.1 Example of setting high-speed on-chip oscillator	
	Modification of voltages in Figure 5-14 CPU clock status transition diagram	
	Addition of description of alternate ports in 6.2 Timer array unit configuration	CHAPTER 6 TIMER ARRAY UNIT
	Modification in Figure 11-10 Format of Serial Data Register mn (SDRmn)	CHAPTER 11 SERIAL ARRAY UNIT
	Deletion of values in Caution 2 in Figure 11-10 Format of Serial Data Register mn (SDRmn) and Caution in 11.6.4 Calculating baud rate	
	Addition of description in cautions in 12.4.2 Setting transfer clock by using the IICWL0 and IICWH0 registers	CHAPTER 12 SERIAL INTERFACE IICA
	Addition of setting values in Figure 15-6 Format of Priority Specification Flag Registers (PR00L, PR00H, PR01L, PR10L, PR10H, PR11L) (20-, 24-pin product)	CHAPTER 15 INTERRUPT FUNCTIONS
	Addition of Cautions 2 and 3 in Figure 18-5 Format of Reset Control Flag Register (RESF)	CHAPTER 18 RESET FUNCTION

Downloaded from Arrow.com.

(10/10)

Edition	Description	Chapter
0.02	Modification of voltage in Figure 19-2 Timing of Generation of Internal Reset Signal by Power-on-reset Circuit and Voltage Detector	CHAPTER 19 POWERON-RESET CIRCUIT
	Deletion of description in 20.1 Function of Voltage Detector	CHAPTER 20 VOLTAGE DETECTOR
	Modification in Figure 20-5 Timing of Interrupt Signal Generation	
	Addition of Caution 2 in Figure 21-4 Format of RAM Parity Error Control Register (RPECTL)	CHAPTER 21 SAFETY FUNCTIONS
	Deletion of the description of boot swap	CHAPTER 23 OPTION BYTE
	Modification of description in Table 24-1 Wiring Between RL78/G12 and Dedicated Flash Memory Programmer and Table 24-2 Pin Connection	CHAPTER 24 FLASH MEMORY
	Modification of description in Notes of 24.1.2 Communication Mode	
	Addition of description in 24.3.1 TOOL0 pin	
	Modification of description in 24.4.1 Data flash overview	
	Modification in Figure 24-8 Setting of Flash Memory Programming Mode	
	Modification in Table 24-7 Flash Memory Control Commands	
	Addition of Caution 2 in 24.7 Flash Memory Programming by Self-Programming	
	Addition of 24.7.1 Flash shield window function	
	Modification of values of Absolute Maximum Ratings	CHAPTER 28 ELECTRICAL SPECIFICATIONS (TARGET)
	Modification of values of On-chip Oscillator Characteristics	
	Modification of values in 28.4 DC Characteristics	
	Modification of values in 28.6.1 Serial array unit	
	Addition of Caution in 28.6.1 Serial array unit	
	Addition of description in Caution of 28.6.1 Serial array unit	
	Addition and deletion of values in 28.7 Analog Characteristics, and modification of the value in the same section	
	Modification in 28.9 Flash Memory Programming Characteristics	
	Modification in 28.10 Timing Specs for Switching Modes and of t_{HD} unit	

Downloaded from Arrow.com.

RL78/G12 User's Manual: Hardware

Publication Date: Rev.2.00 Aug 23, 2013

Published by: Renesas Electronics Corporation

Downloaded from Arrow.com.

SALES OFFICES Renesas Electronics Corporation http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-651-700, Fax: +44-1628-651-804

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-65030, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898

Renesas Electronics Hong Kong Limited
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2886-9318, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics Korea Co., Ltd.
11F., Samik Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

© 2013 Renesas Electronics Corporation. All rights reserved.

Colophon 1.3

Downloaded from Arrow.com.

RL78/G12

Renesas Electronics Corporation

R01UH0200EJ0200

Downloaded from Arrow.com.