To our customers,

Old Company Name in Catalogs and Other Documents

On April 1st, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

Renesas Electronics website: http://www.renesas.com

April 1st, 2010
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (http://www.renesas.com)

Send any inquiries to http://www.renesas.com/inquiry.

Downloaded from Arrow.com.

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. 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.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. 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.
5. When exporting the 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. You should not use Renesas Electronics products or the 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. 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.
6. 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.
7. Renesas Electronics products are classified according to the following three quality grades: "Standard", "High Quality", and "Specific". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below. 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 categorized as "Specific" without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. 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 an application categorized as "Specific" or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is "Standard" unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

   "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.

   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.

   "Specific": Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. 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.
9. 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 system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. 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.
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.

Downloaded from Arrow.com.

Hardware Manual

16

M16C/26A Group (M16C/26A, M16C/26B, M16C/26T)

Hardware Manual

RENESAS 16-BIT SINGLE-CHIP MICROCOMPUTER
M16C FAMILY / M16C/Tiny SERIES

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

Renesas Electronics
www.renesas.com

Rev. 2.00 2007.02

Downloaded from Arrow.com.

Notes regarding these materials

1. This document is provided for reference purposes only so that Renesas customers may select the appropriate Renesas products for their use. Renesas neither makes warranties or representations with respect to the accuracy or completeness of the information contained in this document nor grants any license to any intellectual property rights or any other rights of Renesas or any third party with respect to the information in this document.
2. Renesas shall have no liability for damages or infringement of any intellectual property or other rights arising out of the use of any information in this document, including, but not limited to, product data, diagrams, charts, programs, algorithms, and application circuit examples.
3. You should not use the products or the technology described in this document for the purpose of military applications such as the development of weapons of mass destruction or for the purpose of any other military use. When exporting the products or technology described herein, you should follow the applicable export control laws and regulations, and procedures required by such laws and regulations.
4. All information included in this document such as product data, diagrams, charts, programs, algorithms, and application circuit examples, is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas products listed in this document, please confirm the latest product information with a Renesas sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas such as that disclosed through our website. (http://www.renesas.com)
5. Renesas has used reasonable care in compiling the information included in this document, but Renesas assumes no liability whatsoever for any damages incurred as a result of errors or omissions in the information included in this document.
6. When using or otherwise relying on the information in this document, you should evaluate the information in light of the total system before deciding about the applicability of such information to the intended application. Renesas makes no representations, warranties or guaranties regarding the suitability of its products for any particular application and specifically disclaims any liability arising out of the application and use of the information in this document or Renesas products.
7. With the exception of products specified by Renesas as suitable for automobile applications, Renesas products are not designed, manufactured or tested for applications or otherwise in systems the failure or malfunction of which may cause a direct threat to human life or create a risk of human injury or which require especially high quality and reliability such as safety systems, or equipment or systems for transportation and traffic, healthcare, combustion control, aerospace and aeronautics, nuclear power, or undersea communication transmission. If you are considering the use of our products for such purposes, please contact a Renesas sales office beforehand. Renesas shall have no liability for damages arising out of the uses set forth above.
8. Notwithstanding the preceding paragraph, you should not use Renesas products for the purposes listed below:
   (1) artificial life support devices or systems
   (2) surgical implantations
   (3) healthcare intervention (e.g., excision, administration of medication, etc.)
   (4) any other purposes that pose a direct threat to human life

   Renesas shall have no liability for damages arising out of the uses set forth in the above and purchasers who elect to use Renesas products in any of the foregoing applications shall indemnify and hold harmless Renesas Technology Corp., its affiliated companies and their officers, directors, and employees against any and all damages arising out of such applications.
9. You should use the products described herein within the range specified by Renesas, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas shall have no liability for malfunctions or damages arising out of the use of Renesas products beyond such specified ranges.
10. Although Renesas endeavors to improve the quality and reliability of its products, IC products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Please be sure to implement safety measures to guard against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas 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 applicable measures. Among others, since the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
11. In case Renesas products listed in this document are detached from the products to which the Renesas products are attached or affixed, the risk of accident such as swallowing by infants and small children is very high. You should implement safety measures so that Renesas products may not be easily detached from your products. Renesas shall have no liability for damages arising out of such detachment.
12. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written approval from Renesas.
13. Please contact a Renesas sales office if you have any questions regarding the information contained in this document, Renesas semiconductor products, or if you have any other inquiries.

Downloaded from Arrow.com.

General Precautions in the Handling of MPU/MCU Products

The following usage notes are applicable to all MPU/MCU products from Renesas. For detailed usage notes on the products covered by this manual, refer to the relevant sections of the manual. If the descriptions under General Precautions in the Handling of MPU/MCU Products and in the body of the manual differ from each other, the description in the body of the manual takes precedence.

1. Handling of Unused Pins
   Handle unused pins in accord with the directions given under Handling of Unused Pins in the manual.
   — The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible. Unused pins should be handled as described under Handling of Unused Pins in the manual.
2. Processing at Power-on
   The state of the product is undefined at the moment when power is supplied.
   — The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.
   In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed.
   In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.
3. Prohibition of Access to Reserved Addresses
   Access to reserved addresses is prohibited.
   — The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.
4. Clock Signals
   After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has stabilized.
   — When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.
5. Differences between Products
   Before changing from one product to another, i.e. to one with a different part number, confirm that the change will not lead to problems.
   — The characteristics of MPU/MCU in the same group but having different part numbers may differ because of the differences in internal memory capacity and layout pattern. When changing to products of different part numbers, implement a system-evaluation test for each of the products.

Downloaded from Arrow.com.

How to Use This Manual

1. Purpose and Target Readers

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

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

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

The following documents apply to the M16C/26A Group (M16C/26A, M16C/26B, and M16C/26T). Make sure to refer to the latest versions of these documents. The newest versions of the documents listed may be obtained from the Renesas Technology Web site.

Document Type	Description	Document Title	Document No.
Hardware manual	Hardware specifications (pin assignments, memory maps, peripheral function specifications, electrical characteristics, timing charts) and operation description Note: Refer to the application notes for details on using peripheral functions.	M16C/26A Group (M16C/26A, M16C/26B, M16C/26T) Hardware Manual	This hardware manual
Software manual	Description of CPU instruction set	M16C/60, M16C/20, M16C/Tiny Series Software Manual	REJ09B0137
Application note	Information on using peripheral functions and application examples Sample programs Information on writing programs in assembly language and C	Available from Renesas Technology Web site.	
Renesas technical update	Product specifications, updates on documents, etc.		

Downloaded from Arrow.com.

2. Notation of Numbers and Symbols

The notation conventions for register names, bit names, numbers, and symbols used in this manual are described below.

(1) Register Names, Bit Names, and Pin Names

Registers, bits, and pins are referred to in the text by symbols. The symbol is accompanied by the word "register," "bit," or "pin" to distinguish the three categories.

Examples the PM03 bit in the PM0 register
P3_5 pin, VCC pin

(2) Notation of Numbers

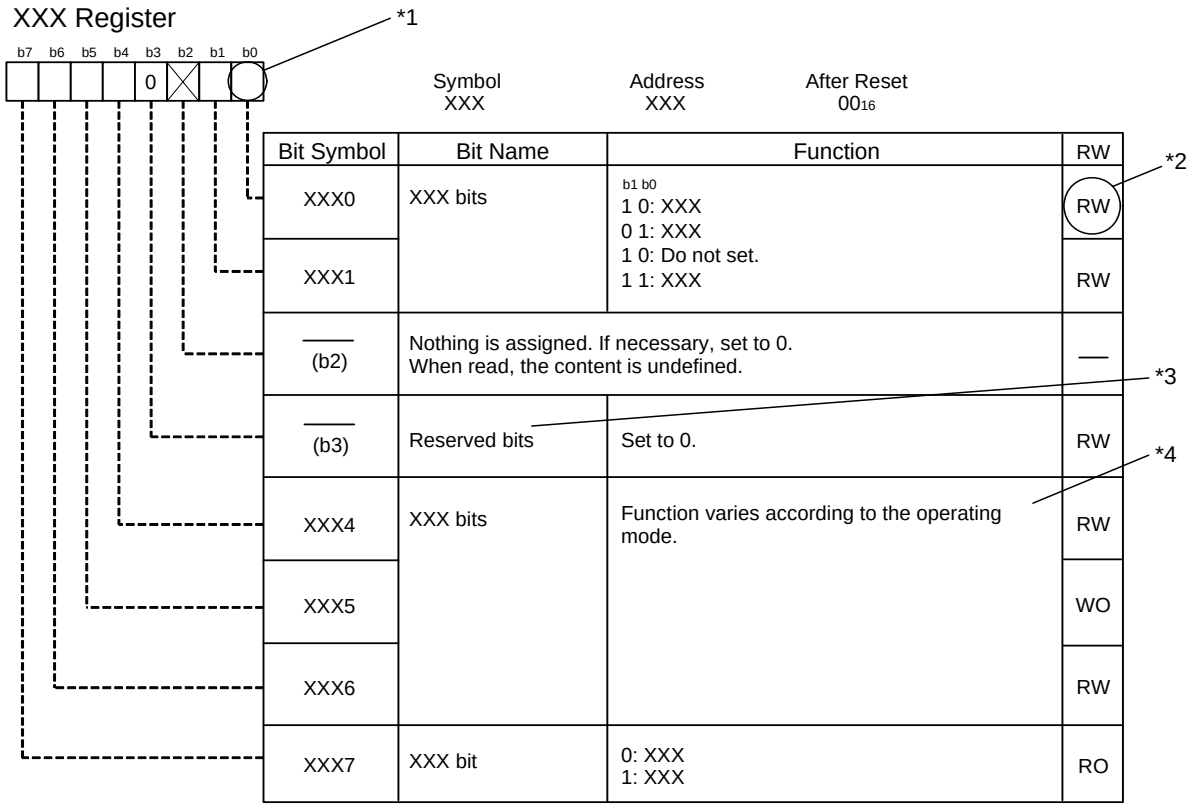
The indication "$_2$" is appended to numeric values given in binary format. However, nothing is appended to the values of single bits. The indication "$_{16}$" is appended to numeric values given in hexadecimal format. Nothing is appended to numeric values given in decimal format.

Examples Binary: 11_2
Hexadecimal: $EFA0_{16}$
Decimal: 1234

Downloaded from Arrow.com.

3. Register Notation

The symbols and terms used in register diagrams are described below.



*1

Blank: Set to 0 or 1 according to the application.
0: Set to 0.
1: Set to 1.
X: Nothing is assigned.

*2

RW: Read and write.
RO: Read only.
WO: Write only.
–: Nothing is assigned.

*3

• Reserved bit
Reserved bit. Set to specified value.

*4

• Nothing is assigned
Nothing is assigned to the bit. As the bit may be used for future functions, if necessary, set to 0.
• Do not set to a value
Operation is not guaranteed when a value is set.
• Function varies according to the operating mode.
The function of the bit varies with the peripheral function mode. Refer to the register diagram for information on the individual modes.

Downloaded from Arrow.com.

4. List of Abbreviations and Acronyms

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

All trademarks and registered trademarks are the property of their respective owners.

IEBus is a registered trademark of NEC Electronics Corporation.

Downloaded from Arrow.com.

Table of Contents

Quick Reference by Address _____ B-1

1. Overview _____ 1

1.1 Applications 1
1.2 Performance Outline 2
1.3 Block Diagram 4
1.4 Product List 6
1.5 Pin Assignments 11
1.6 Pin Description 15

2. Central Processing Unit (CPU) _____ 17

2.1 Data Registers (R0, R1, R2 and R3) 17
2.2 Address Registers (A0 and A1) 17
2.3 Frame Base Register (FB) 18
2.4 Interrupt Table Register (INTB) 18
2.5 Program Counter (PC) 18
2.6 User Stack Pointer (USP) and Interrupt Stack Pointer (ISP) 18
2.7 Static Base Register (SB) 18
2.8 Flag Register (FLG) 18
2.8.1 Carry Flag (C Flag) 18
2.8.2 Debug Flag (D Flag) 18
2.8.3 Zero Flag (Z Flag) 18
2.8.4 Sign Flag (S Flag) 18
2.8.5 Register Bank Select Flag (B Flag) 18
2.8.6 Overflow Flag (O Flag) 18
2.8.7 Interrupt Enable Flag (I Flag) 18
2.8.8 Stack Pointer Select Flag (U Flag) 18
2.8.9 Processor Interrupt Priority Level (IPL) 18
2.8.10 Reserved Area 18

3. Memory _____ 19

4. Special Function Registers (SFRs) _____ 20

Downloaded from Arrow.com.

5. Reset ____ 26
5.1 Hardware Reset ... 26
5.1.1 Hardware Reset 1 ... 26
5.1.2 Hardware Reset 2 ... 26
5.2 Software Reset ... 27
5.3 Watchdog Timer Reset ... 27
5.4 Oscillation Stop Detection Reset ... 27
5.5 Voltage Detection Circuit ... 29
5.5.1 Voltage Down Detection Interrupt ... 32
5.5.2 Limitations on Exiting Stop Mode ... 34
5.5.3 Limitations on Exiting Wait Mode ... 34
6. Processor Mode ____ 35
7. Clock Generation Circuit ____ 38
7.1 Main Clock ... 45
7.2 Sub Clock ... 46
7.3 On-chip Oscillator Clock ... 47
7.4 PLL Clock ... 47
7.5 CPU Clock and Peripheral Function Clock ... 49
7.5.1 CPU Clock ... 49
7.5.2 Peripheral Function Clock (f_1, f_2, f_8, f_{32}, f_{1SIO}, f_{2SIO}, f_{8SIO}, f_{32SIO}, f_{AD}, f_{C32}) ... 49
7.5.3 ClockOutput Function ... 49
7.6 Power Control ... 50
7.6.1 Normal Operation Mode ... 50
7.6.2 Wait Mode ... 51
7.6.3 Stop Mode ... 53
7.7 System Clock Protective Function ... 57
7.8 Oscillation Stop and Re-oscillation Detect Function ... 57
7.8.1 Operation When the CM27 bit is set to "0" (Oscillation Stop Detection Reset) ... 58
7.8.2 Operation When the CM27 bit is set to "1" (Oscillation Stop and Re-oscillation Detect Interrupt) ... 58
7.8.3 How to Use Oscillation Stop and Re-oscillation Detect Function ... 59
8. Protection ____ 60
9. Interrupt ____ 61
9.1 Type of Interrupts ... 61
9.1.1 Software Interrupts ... 62
9.1.2 Hardware Interrupts ... 63

Downloaded from Arrow.com.

9.2 Interrupts and Interrupt Vector 64
9.2.1 Fixed Vector Tables 64
9.2.2 Relocatable Vector Tables 65
9.3 Interrupt Control 66
9.3.1 I Flag 69
9.3.2 IR Bit 69
9.3.3 ILVL2 to ILVL0 Bits and IPL 69
9.4 Interrupt Sequence 70
9.4.1 Interrupt Response Time 71
9.4.2 Variation of IPL when Interrupt Request is Accepted 71
9.4.3 Saving Registers 72
9.4.4 Returning from an Interrupt Routine 74
9.5 Interrupt Priority 74
9.5.1 Interrupt Priority Resolution Circuit 74
9.6 $\overline{\text{INT}}$ Interrupt 76
9.7 $\overline{\text{NMI}}$ Interrupt 77
9.8 Key Input Interrupt 77
9.9 Address Match Interrupt 78
10. Watchdog Timer 80
10.1 Count Source Protective Mode 81
11. DMAC 82
11.1 Transfer Cycles 87
11.2. DMA Transfer Cycles 89
11.3 DMA Enable 90
11.4 DMA Request 90
11.5 Channel Priority and DMA Transfer Timing 91
12. Timer 92
12.1 Timer A 94
12.1.1. Timer Mode 97
12.1.2. Event Counter Mode 98
12.1.3. One-shot Timer Mode 103
12.1.4. Pulse Width Modulation (PWM) Mode 105
12.2 Timer B 108
12.2.1 Timer Mode 111
12.2.2 Event Counter Mode 112
12.2.3 Pulse Period and Pulse Width Measurement Mode 113
12.2.4 A/D Trigger Mode 115

Downloaded from Arrow.com.

12.3 Three-phase Motor Control Timer Function 117
12.3.1 Position-data-retain Function 128
12.3.2 Three-phase/Port Output Switch Function 130
13. Serial I/O 132
13.1. UARTi (i=0 to 2) 132
13.1.1. Clock Synchronous serial I/O Mode 142
13.1.2. Clock Asynchronous Serial I/O (UART) Mode 150
13.1.3 Special Mode 1 (I^2C bus mode)(UART2) 158
13.1.4 Special Mode 2 (UART2) 168
13.1.5 Special Mode 3 (IE Bus mode)(UART2) 173
13.1.6 Special Mode 4 (SIM Mode) (UART2) 175
14. A/D Converter 180
14.1 Operation Modes 186
14.1.1 One-Shot Mode 186
14.1.2 Repeat mode 188
14.1.3 Single Sweep Mode 190
14.1.4 Repeat Sweep Mode 0 192
14.1.5 Repeat Sweep Mode 1 194
14.1.6 Simultaneous Sample Sweep Mode 196
14.1.7 Delayed Trigger Mode 0 199
14.1.8 Delayed Trigger Mode 1 205
14.2 Resolution Select Function 211
14.3 Sample and Hold 211
14.4 Power Consumption Reducing Function 211
14.5 Output Impedance of Sensor under A/D Conversion 212
15. CRC Calculation Circuit 213
15.1. CRC Snoop 213
16. Programmable I/O Ports 216
16.1 Port Pi Direction Register (PDi Register, i = 1, 6 to 10) 216
16.2 Port Pi Register (Pi Register, i = 1, 6 to 10) 216
16.3 Pull-up Control Register 0 to Pull-up Control Register 2 (PUR0 to PUR2 Registers) 216
16.4 Port Control Register 217
16.5 Pin Assignment Control register (PACR) 217
16.6 Digital Debounce function 217
17. Flash Memory Version 230
17.1 Flash Memory Performance 230
17.1.1 Boot Mode 231

Downloaded from Arrow.com.

17.2 Memory Map 232
17.3 Functions To Prevent Flash Memory from Rewriting 235
17.3.1 ROM Code Protect Function 235
17.3.2 ID Code Check Function 235
17.4 CPU Rewrite Mode 237
17.4.1 EW0 Mode 238
17.4.2 EW1 Mode 238
17.5 Register Description 239
17.5.1 Flash memory control register 0 (FMR0) 239
17.5.2 Flash memory control register 1 (FMR1) 240
17.5.3 Flash memory control register 4 (FMR4) 240
17.6 Precautions in CPU Rewrite Mode 245
17.6.1 Operation Speed 245
17.6.2 Prohibited Instructions 245
17.6.3 Interrupts 245
17.6.4 How to Access 245
17.6.5 Writing in the User ROM Space 245
17.6.6 DMA Transfer 246
17.6.7 Writing Command and Data 246
17.6.8 Wait Mode 246
17.6.9 Stop Mode 246
17.6.10 Low Power Consumption Mode and On-chip Oscillator-Low Power Consumption Mode ... 246
17.7 Software Commands 247
17.7.1 Read Array Command (FF_{16}) 247
17.7.2 Read Status Register Command (70_{16}) 247
17.7.3 Clear Status Register Command (50_{16}) 248
17.7.4 Program Command (40_{16}) 248
17.7.5 Block Erase 249
17.8 Status Register 251
17.8.1 Sequence Status (SR7 and FMR00 Bits) 251
17.8.2 Erase Status (SR5 and FMR07 Bits) 251
17.8.3 Program Status (SR4 and FMR06 Bits) 251
17.8.4 Full Status Check 252
17.9 Standard Serial I/O Mode 254
17.9.1 ID Code Check Function 254
17.9.2 Example of Circuit Application in Standard Serial I/O Mode 258
17.10 Parallel I/O Mode 260
17.10.1 ROM Code Protect Function 260

Downloaded from Arrow.com.

18. Electrical Characteristics ____ 261
18.1. M16C/26A, M16C/26B (Normal version) ... 261
18.2. M16C/26T (T version) ... 280
19. Usage Notes ____ 299
19.1 SFR ... 299
19.1.1 Precaution for 48-pin package ... 299
19.1.2 Precaution for 42-pin package ... 299
19.1.3 Register Setting ... 299
19.2 PLL Frequency Synthesizer ... 300
19.3 Power Control ... 301
19.4 Protect ... 303
19.5 Interrupts ... 304
19.5.1 Reading address 0000_{16} ... 304
19.5.2 Setting the SP ... 304
19.5.3 The $\overline{\text{NMI}}$ Interrupt ... 304
19.5.4 Changing the Interrupt Generation Factor ... 304
19.5.5 $\overline{\text{INT}}$ Interrupt ... 305
19.5.6 Rewrite the Interrupt Control Register ... 306
19.5.7 Watchdog Timer Interrupt ... 306
19.6 DMAC ... 307
19.6.1 Write to DMAE Bit in DMiCON Register ... 307
19.7 Timer ... 308
19.7.1 Timer A ... 308
19.7.2 Timer B ... 311
19.7.3 Three-phase Motor Control Timer Function ... 312
19.8 Serial I/O ... 313
19.8.1 Clock-Synchronous Serial I/O ... 313
19.8.2 Serial I/O (UART Mode) ... 314
19.9 A/D Converter ... 315
19.10 Programmable I/O Ports ... 317
19.11 Electric Characteristic Differences Between Mask ROM and Flash Memory Version Microcomputers ... 318
19.12 Mask ROM Version ... 319
19.12.1 Internal ROM area ... 319
19.12.2 Reserve bit ... 319

Downloaded from Arrow.com.

19.13 Flash Memory Version 320
19.13.1 Functions to Inhibit Rewriting Flash Memory 320
19.13.2 Stop mode 320
19.13.3 Wait mode 320
19.13.4 Low power dissipation mode, on-chip oscillator low power dissipation mode ... 320
19.13.5 Writing command and data 320
19.13.6 Program Command 320
19.13.7 Operation speed 320
19.13.8 Instructions prohibited in EW0 Mode 320
19.13.9 Interrupts 321
19.13.10 How to access 321
19.13.11 Writing in the user ROM area 321
19.13.12 DMA transfer 321
19.13.13 Regarding Programming/Erasure Times and Execution Time 321
19.13.14 Definition of Programming/Erasure Times 322
19.13.15 Flash Memory Version Electrical Characteristics 10,000 E/W cycle product .. 322
19.13.16 Boot Mode 322
19.14 Noise 323
19.15 Instruction for a Device Use 324
Appendix 1. Package Dimensions 325
Appendix 2. Functional Difference 326
Appendix 2.1 Differences between M16C/26A, M16C/26B, and M16C/26T 326
Appendix 2.2 Differences between M16C/26A Group and M16C/26 Group 327
Register Index 328

Downloaded from Arrow.com.

Quick Reference by Address

Address	Register	Symbol	Page
0000₁₆			
0001₁₆			
0002₁₆			
0003₁₆			
0004₁₆	Processor mode register 0	PM0	35
0005₁₆	Processor mode register 1	PM1	35
0006₁₆	System clock control register 0	CM0	40
0007₁₆	System clock control register 1	CM1	41
0008₁₆			
0009₁₆	Address match interrupt enable register	AIER	79
000A₁₆	Protect register	PRCR	60
000B₁₆			
000C₁₆	Oscillation stop detection register	CM2	42
000D₁₆			
000E₁₆	Watchdog timer start register	WDTS	81
000F₁₆	Watchdog timer control register	WDC	81
0010₁₆ 0011₁₆ 0012₁₆	Address match interrupt register 0	RMAD0	79
0013₁₆			
0014₁₆ 0015₁₆ 0016₁₆	Address match interrupt register 1	RMAD1	79
0017₁₆			
0018₁₆			
0019₁₆	Voltage detection register 1	VCR1	30
001A₁₆	Voltage detection register 2	VCR2	30
001B₁₆			
001C₁₆	PLL control register 0	PLC0	44
001D₁₆			
001E₁₆	Processor mode register 2	PM2	36, 43
001F₁₆	Voltage down detection interrupt register	D4INT	30
0020₁₆ 0021₁₆ 0022₁₆	DMA0 source pointer	SAR0	81
0023₁₆			
0024₁₆ 0025₁₆ 0026₁₆	DMA0 destination pointer	DAR0	86
0027₁₆			
0028₁₆ 0029₁₆	DMA0 transfer counter	TCR0	86
002A₁₆			
002B₁₆			
002C₁₆	DMA0 control register	DM0CON	85
002D₁₆			
002E₁₆			
002F₁₆			
0030₁₆ 0031₁₆ 0032₁₆	DMA1 source pointer	SAR1	86
0033₁₆			
0034₁₆ 0035₁₆ 0036₁₆	DMA1 destination pointer	DAR1	86
0037₁₆			
0038₁₆ 0039₁₆	DMA1 transfer counter	TCR1	86
003A₁₆			
003B₁₆			
003C₁₆	DMA1 control register	DM1CON	85
003D₁₆			
003E₁₆			
003F₁₆			

Address	Register	Symbol	Page
0040₁₆			
0041₁₆			
0042₁₆			
0043₁₆			
0044₁₆	INT3 interrupt control register	INT3IC	67
0045₁₆			
0046₁₆			
0047₁₆			
0048₁₆	INT5 interrupt control register	INT5IC	67
0049₁₆	INT4 interrupt control register	INT4IC	67
004A₁₆	UART2 Bus collision detection interrupt control register	BCNIC	67
004B₁₆	DMA0 interrupt control register	DM0IC	67
004C₁₆	DMA1 interrupt control register	DM1IC	67
004D₁₆	Key input interrupt control register	KUPIC	67
004E₁₆	A/D conversion interrupt control register	ADIC	67
004F₁₆	UART2 transmit interrupt control register	S2TIC	67
0050₁₆	UART2 receive interrupt control register	S2RIC	67
0051₁₆	UART0 transmit interrupt control register	S0TIC	67
0052₁₆	UART0 receive interrupt control register	S0RIC	67
0053₁₆	UART1 transmit interrupt control register	S1TIC	67
0054₁₆	UART1 receive interrupt control register	S1RIC	67
0055₁₆	Timer A0 interrupt control register	TA0IC	67
0056₁₆	Timer A1 interrupt control register	TA1IC	67
0057₁₆	Timer A2 interrupt control register	TA2IC	67
0058₁₆	Timer A3 interrupt control register	TA3IC	67
0059₁₆	Timer A4 interrupt control register	TA4IC	67
005A₁₆	Timer B0 interrupt control register	TB0IC	67
005B₁₆	Timer B1 interrupt control register	TB1IC	67
005C₁₆	Timer B2 interrupt control register	TB2IC	67
005D₁₆	INT0 interrupt control register	INT0IC	67
005E₁₆	INT1 interrupt control register	INT1IC	67
005F₁₆	INT2 interrupt control register	INT2IC	67
0060₁₆			
0061₁₆			
0062₁₆			
0063₁₆			
0064₁₆			
0065₁₆			
0066₁₆			
0067₁₆			
0068₁₆			
0069₁₆			
006A₁₆			
006B₁₆			
006C₁₆			
006D₁₆			
006E₁₆			
006F₁₆			
0070₁₆			
0071₁₆			
0072₁₆			
0073₁₆			
0074₁₆			
0075₁₆			
0076₁₆			
0077₁₆			
0078₁₆			
0079₁₆			
007A₁₆			
007B₁₆			
007C₁₆			
007D₁₆			
007E₁₆			
007F₁₆			

NOTE:
1. The blank areas are reserved and cannot be accessed by users.

Downloaded from Arrow.com.

Quick Reference by Address

Address	Register	Symbol	Page
0080₁₆			
0081₁₆			
0082₁₆			
0083₁₆			
0084₁₆			
0085₁₆			
0086₁₆			
01B0₁₆			
01B1₁₆			
01B2₁₆			
01B3₁₆	Flash memory control register 4 (2)	FMR4	242
01B4₁₆			
01B5₁₆	Flash memory control register 1 (2)	FMR1	241
01B6₁₆			
01B7₁₆	Flash memory control register 0 (2)	FMR0	241
01B8₁₆			
01B9₁₆			
01BA₁₆			
01BB₁₆			
01BC₁₆			
01BD₁₆			
01BE₁₆			
01BF₁₆			
0250₁₆			
0251₁₆			
0252₁₆			
0253₁₆			
0254₁₆			
0255₁₆			
0256₁₆			
0257₁₆			
0258₁₆			
0259₁₆			
025A₁₆	Three phase protect control register	TPRC	131
025B₁₆			
025C₁₆	On-chip oscillator control register	ROCR	41
025D₁₆	Pin assignment control register	PACR	139, 226
025E₁₆	Peripheral clock select register	PCLKR	43
025F₁₆			
02E0₁₆			
02E1₁₆			
02E2₁₆			
02E3₁₆			
02E4₁₆			
02E5₁₆			
02E6₁₆			
02E7₁₆			
02E8₁₆			
02E9₁₆			
033D₁₆			
033E₁₆	NMI digital debounce register	NDDR	227
033F₁₆	P17 digital debounce register	P17DDR	227

Address	Register	Symbol	Page
0340₁₆			
0341₁₆			
0342₁₆ 0343₁₆	Timer A1-1 register	TA11	122
0344₁₆ 0345₁₆	Timer A2-1 register	TA21	122
0346₁₆ 0347₁₆	Timer A4-1 register	TA41	122
0348₁₆	Three-phase PWM control register 0	INVC0	119
0349₁₆	Three-phase PWM control register 1	INVC1	120
034A₁₆	Three-phase output buffer register 0	IDB0	121
034B₁₆	Three-phase output buffer register 1	IDB1	121
034C₁₆	Dead time timer	DTT	121
034D₁₆	Timer B2 interrupt occurrence frequency set counter	ICTB2	122
034E₁₆	Position-data-retain function contol register	PDRF	129
034F₁₆			
0350₁₆			
0351₁₆			
0352₁₆			
0353₁₆			
0354₁₆			
0355₁₆			
0356₁₆			
0357₁₆			
0358₁₆	Port function contol register	PFCR	131
0359₁₆			
035A₁₆			
035B₁₆			
035C₁₆			
035D₁₆			
035E₁₆	Interrupt request cause select register 2	IFSR2A	68
035F₁₆	Interrupt request cause select register	IFSR	68, 76
0360₁₆			
0361₁₆			
0362₁₆			
0363₁₆			
0364₁₆			
0365₁₆			
0366₁₆			
0367₁₆			
0368₁₆			
0369₁₆			
036A₁₆			
036B₁₆			
036C₁₆			
036D₁₆			
036E₁₆			
036F₁₆			
0370₁₆			
0371₁₆			
0372₁₆			
0373₁₆			
0374₁₆	UART2 special mode register 4	U2SMR4	141
0375₁₆	UART2 special mode register 3	U2SMR3	141
0376₁₆	UART2 special mode register 2	U2SMR2	140
0377₁₆	UART2 special mode register	U2SMR	140
0378₁₆	UART2 transmit/receive mode register	U2MR	137
0379₁₆	UART2 bit rate generator	U2BRG	136
037A₁₆ 037B₁₆	UART2 transmit buffer register	U2TB	136
037C₁₆	UART2 transmit/receive control register 0	U2C0	138
037D₁₆	UART2 transmit/receive control register 1	U2C1	139
037E₁₆ 037F₁₆	UART2 receive buffer register	U2RB	136

NOTES:
1. The blank areas are reserved and cannot be accessed by users.
2. This register is included in the flash memory version.

Downloaded from Arrow.com.

Quick Reference by Address

Address	Register	Symbol	Page
0380₁₆	Count start flag	TABSR	95, 110, 124
0381₁₆	Clock prescaler reset flag	CPSRF	r 0)
0382₁₆	One-shot start flag	ONSF	96
0383₁₆	Trigger select register	TRGSR	96, 124
0384₁₆	Up-down flag	UDF	95
0385₁₆			
0386₁₆ 0387₁₆	Timer A0 register	TA0	95
0388₁₆ 0389₁₆	Timer A1 register	TA1	95, 122
038A₁₆ 038B₁₆	Timer A2 register	TA2	95, 122
038C₁₆ 038D₁₆	Timer A3 register	TA3	95
038E₁₆ 038F₁₆	Timer A4 register	TA4	95, 122
0390₁₆ 0391₁₆	Timer B0 register	TB0	110
0392₁₆ 0393₁₆	Timer B1 register	TB1	110
0394₁₆ 0395₁₆	Timer B2 register	TB2	110, 124
0396₁₆	Timer A0 mode register	TA0MR	94
0397₁₆	Timer A1 mode register	TA1MR	94, 125
0398₁₆	Timer A2 mode register	TA2MR	94, 125
0399₁₆	Timer A3 mode register	TA3MR	94
039A₁₆	Timer A4 mode register	TA4MR	94, 125
039B₁₆	Timer B0 mode register	TB0MR	109
039C₁₆	Timer B1 mode register	TB1MR	109
039D₁₆	Timer B2 mode register	TB2MR	109, 125
039E₁₆	Timer B2 special mode register	TB2SC	123, 185
039F₁₆			
03A0₁₆	UART0 transmit/receive mode register	U0MR	137
03A1₁₆	UART0 bit rate generator	U0BRG	136
03A2₁₆ 03A3₁₆	UART0 transmit buffer register	U0TB	136
03A4₁₆	UART0 transmit/receive control register 0	U0C0	138
03A5₁₆	UART0 transmit/receive control register 1	U0C1	139
03A6₁₆ 03A7₁₆	UART0 receive buffer register	U0RB	136
03A8₁₆	UART1 transmit/receive mode register	U1MR	137
03A9₁₆	UART1 bit rate generator	U1BRG	136
03AA₁₆ 03AB₁₆	UART1 transmit buffer register	U1TB	136
03AC₁₆	UART1 transmit/receive control register 0	U1C0	138
03AD₁₆	UART1 transmit/receive control register 1	U1C1	139
03AE₁₆ 03AF₁₆	UART1 receive buffer register	U1RB	136
03B0₁₆	UART transmit/receive control register 2	UCON	138
03B1₁₆			
03B2₁₆			
03B3₁₆			
03B4₁₆ 03B5₁₆	CRC snoop address register	CRCSAR	214
03B6₁₆	CRC mode register	CRCMR	214
03B7₁₆			
03B8₁₆	DMA0 request cause select register	DM0SL	84
03B9₁₆			
03BA₁₆	DMA1 request cause select register	DM1SL	85
03BB₁₆			
03BC₁₆ 03BD₁₆	CRC data register	CRCD	214
03BE₁₆	CRC input register	CRCIN	214
03BF₁₆			

Address	Register	Symbol	Page
03C0₁₆ 03C1₁₆	A/D register 0	AD0	184
03C2₁₆ 03C3₁₆	A/D register 1	AD1	184
03C4₁₆ 03C5₁₆	A/D register 2	AD2	184
03C6₁₆ 03C7₁₆	A/D register 3	AD3	184
03C8₁₆ 03C9₁₆	A/D register 4	AD4	184
03CA₁₆ 03CB₁₆	A/D register 5	AD5	184
03CC₁₆ 03CD₁₆	A/D register 6	AD6	184
03CE₁₆ 03CF₁₆	A/D register 7	AD7	184
03D0₁₆			
03D1₁₆			
03D2₁₆	A/D trigger control register	ADTRGCON	183
03D3₁₆	A/D convert status register 0	ADSTAT0	184
03D4₁₆	A/D control register 2	ADCON2	182
03D5₁₆			
03D6₁₆	A/D control register 0	ADCON0	182
03D7₁₆	A/D control register 1	ADCON1	182
03D8₁₆			
03D9₁₆			
03DA₁₆			
03DB₁₆			
03DC₁₆			
03DD₁₆			
03DE₁₆			
03DF₁₆			
03E0₁₆			
03E1₁₆	Port P1 register	P1	224
03E2₁₆			
03E3₁₆	Port P1 direction register	PD1	223
03E4₁₆			
03E5₁₆			
03E6₁₆			
03E7₁₆			
03E8₁₆			
03E9₁₆			
03EA₁₆			
03EB₁₆			
03EC₁₆	Port P6 register	P6	224
03ED₁₆	Port P7 register	P7	224
03EE₁₆	Port P6 direction register	PD6	223
03EF₁₆	Port P7 direction register	PD7	223
03F0₁₆	Port P8 register	P8	224
03F1₁₆	Port P9 register	P9	224
03F2₁₆	Port P8 direction register	PD8	223
03F3₁₆	Port P9 direction register	PD9	223
03F4₁₆	Port P10 register	P10	224
03F5₁₆			
03F6₁₆	Port P10 direction register	PD10	223
03F7₁₆			
03F8₁₆			
03F9₁₆			
03FA₁₆			
03FB₁₆			
03FC₁₆	Pull-up control register 0	PUR0	225
03FD₁₆	Pull-up control register 1	PUR1	225
03FE₁₆	Pull-up control register 2	PUR2	225
03FF₁₆	Port control register	PCR	226

NOTE:
1. The blank areas are reserved and cannot be accessed by users.

Downloaded from Arrow.com.

1. Overview

The M16C/26A Group (M16C/26A, M16C/26B, M16C/26T) is a single-chip control MCU, fabricated using high-performance silicon gate CMOS technology, embedding the M16C/60 Series CPU core. The M16C/26A Group (M16C/26A, M16C/26B, M16C/26T) is housed in 42-pin and 48-pin plastic molded packages. This MCU combines advanced instruction manipulation capabilities to process complex instructions by less bytes and execute instructions at higher speed. The M16C/26A Group (M16C/26A, M16C/26B, M16C/26T) has a multiplier and DMAC adequate for office automation, communication devices and industrial equipment, and other high-speed processing applications. The M16C/26A and M16C/26B have normal version. The M16C/26T has T version and V version.

1.1 Applications

Audio, cameras, office/communications/portable/ equipment, air-conditioning equipment, home appliances, etc.

Downloaded from Arrow.com.

1.2 Performance Outline

Table 1.1 and **1.2** outline performance overview of the M16C/26A Group (M16C/26A, M16C/26B, M16C/26T).

Table 1.1. M16C/26A Group(M16C/26A, M16C/26B, M16C/26T) Performance (48-Pin Package)

Item		Specification
CPU	Basic instructions	91 instructions
	Minimun instruction execution time	41.7 ns (f(BCLK) = 24MHz(3), Vcc = 4.2 to 5.5 V) (M16C/26B) 50 ns (f(BCLK) = 20MHz, Vcc = 3.0 to 5.5 V) (M16C/26A, M16C/26B, M16C/26T(T-ver.)) 100 ns (f(BCLK) = 10MHz, Vcc = 2.7 to 5.5 V) (M16C/26A , M16C/26B) 50 ns (f(BCLK) = 20MHz, Vcc = 4.2 to 5.5 V -40 to 105°C) (M16C/26T(V-ver.)) 62.5 ns (f(BCLK) = 16MHz, Vcc = 4.2 to 5.5 V -40 to 125°C) (M16C/26T(V-ver.))
	Operating mode	Single-chip mode
	Address space	1 Mbyte
	Memory capacity	ROM/RAM: See **1.4 Product Information**
Peripheral Function	I/O ports	39 I/O pins
	Multifunction timers	TimerA:16 bits x 5 channels, TimerB:16 bits x 3 channels Three-phase motor control timer
	Serial I/O	2 channels (UART, clock synchronous serial I/O) 1 channel (UART, clock synchronous, I^2C bus, or IEBus(1))
	A/D converter	10 bit A/D Converter : 1 circuit, 12 channels
	DMAC	2 channels
	CRC calcuration circuit	1 circuit (CRC-CCITT and CRC-16) with MSB/LSB selectable
	Watchdog timer	15 bits x 1 channel (with prescaler)
	Interrupts	20 internal and 8 external sources, 4 software sources, Interrupt priority level: 7
	Clock generation circuit	4 circuits Main clock oscillation circuit(*), Sub-clock oscillation circuit(*) On-chip oscillator, PLL frequency synthesizer (*)Equipped with a built-in feedback resister.
	Oscillation stop detection	Main clock oscillation stop, re-oscillation detection function
	Voltage detection circuit	On-chip (M16C/26A, M16C/26B), not on-chip (M16C/26T)
Electrical Characteristics	Power supply voltage	Vcc = 4.2 to 5.5 V (f(BCLK) = 24 MHz)(3) (M16C/26B) Vcc = 3.0 to 5.5 V (f(BCLK) = 20 MHz) (M16C/26A, M16C/26B) Vcc = 2.7 to 5.5 V (f(BCLK) = 10 MHz) Vcc = 3.0 to 5.5 V (M16C/26T(T-ver.)) Vcc = 4.2 to 5.5 V (M16C/26T(V-ver.))
	Power consumption	20 mA (Vcc = 5 V, f(BCLK) = 24 MHz) (M16C/26B) 16 mA (Vcc = 5 V, f(BCLK) = 20 MHz) 25 µA (f(XCIN) = 32 KHz on RAM) 3 µA (Vcc = 3 V, f(XCIN) = 32 KHz, in wait mode) 0.7 µA (Vcc = 3 V, in stop mode)
Flash Memory Version	Programming /erasure voltage	2.7 to 5.5 V (M16C/26A, M16C/26B) 3.0 to 5.5 V (M16C/26T(T-ver.)) 4.2 to 5.5 V (M16C/26T(V-ver.))
	Programming /erasure endurance	100 times (all area) or 1,000 times (block 0 to 3) / 10,000 times (block A, block B)(2)
Operating Ambient Temperature		-20 to 85°C / -40 to 85°C (2) (M16C/26A , M16C/26B) -40 to 85°C (M16C/26T(T-ver.)) -40 to 105°C / -40 to 125°C (M16C/26T(V-ver.))
Package		48-pin plastic molded QFP

NOTES:
1. IEBus is a trademark of NEC Electronics Corporation.
2. See **Tables 1.7 to 1.10 Product Code** for the program and erase endurance, and operating ambient temperature.
3. The PLL frequency synthesizer is used to run the M16C/26B at f(BCLK) = 24 MHz.

Downloaded from Arrow.com.

Table 1.2. Performance outline of M16C/26A group (M16C/26A, M16C/26B) (42-pin package)

	Item	Performance
CPU	Basic instructions	91 instructions
	Minimun instruction execution time	41.7 ns (f(BCLK) = 24 MHz (3), VCC = 4.2 to 5.5 V (M16C/26B) 50 ns (f(BCLK) = 20 MHz, Vcc = 3.0 to 5.5 V) (M16C/26A, M16C/26B) 100 ns (f(BCLK) = 10 MHz, Vcc = 2.7 to 5.5 V) (M16C/26A, M16C/26B)
	Operation mode	Single-chip mode
	Address space	1M byte
	Memory capacity	ROM/RAM: See **1.4 Product Information**
Peripheral function	Port	33 I/O pins
	Multifunction timer	Timer A: 16 bits x 5 channels, Timer B: 16 bits x 3 channels Three-phase motor control timer
	Serial I/O	1 channel (UART, clock synchronous serial I/O) 1 channel (UART, clock synchronous, I^2C bus, or IEBus(1))
	A/D converter	10 bit A/D converter: 1 circuit, 10 channels
	DMAC	2 channels
	CRC calcuration circuit	1 circuits (CRC-CCITT and CRC-16) with MSB/LSB selectable
	Watchdog timer	15 bits x 1 channel (with prescaler)
	Interrupt	18 internal and 8 external sources, 4 software sources, Interrupt priority level: 7
	Clock generation circuit	4 circuits Main clock(*), Sub-clock(*) On-chip oscillator, PLL frequency synthesizer (*)Equipped with a built-in feedback resister.
	Oscillation stop detection	Main clock oscillation stop, re-oscillation detection function
	Voltage detection circuit	On-chip
Electrical Characteristics	Supply voltage	Vcc = 4.2 to 5.5 V (f(BCLK) = 24 MHz)(3) (M16C/26B) Vcc = 3.0 to 5.5 V (f(BCLK) = 20 MHz) (M16C/26A, M16C/26B) Vcc = 2.7 to 5.5 V (f(BCLK) = 10 MHz)
	Power Consumption	20 mA (Vcc = 5 V, f(BCLK) = 24 MHz) (M16C/26B) 16 mA (Vcc = 5 V, f(BCLK) = 20 MHz) 25 µA (f(XCIN) = 32 KHz on RAM) 3 µA (Vcc = 3 V, f(XCIN) = 32 KHz, in wait mode) 0.7 µA (Vcc = 3 V, in stop mode)
Flash memory	Programming/erasure voltage	2.7 to 5.5 V
	Programming/erasure endurance	100 times (all area) or 1,000 times (block 0 to 3) / 10,000 times (block A, block B)(2)
Operating Ambient Temperature		-20 to 85°C / -40 to 85°C (2)
Package		42-pin plastic molded SSOP

NOTES:
1. IEBus is a trademark of NEC Electronics Corporation.
2. See **Tables 1.7 and 1.8 Product Code** for the program and erase endurance, and operating ambient temperature.
3. The PLL frequency synthesizer is used to run the M16C/26B at f(BCLK) = 24 MHz.

Downloaded from Arrow.com.

1.3 Block Diagram

Figure 1.1 and **1.2** show block diagrams of the M16C/26A Group (M16C/26A, M16C/26B, M16C/26T) 48-pin package and 42-pin package.

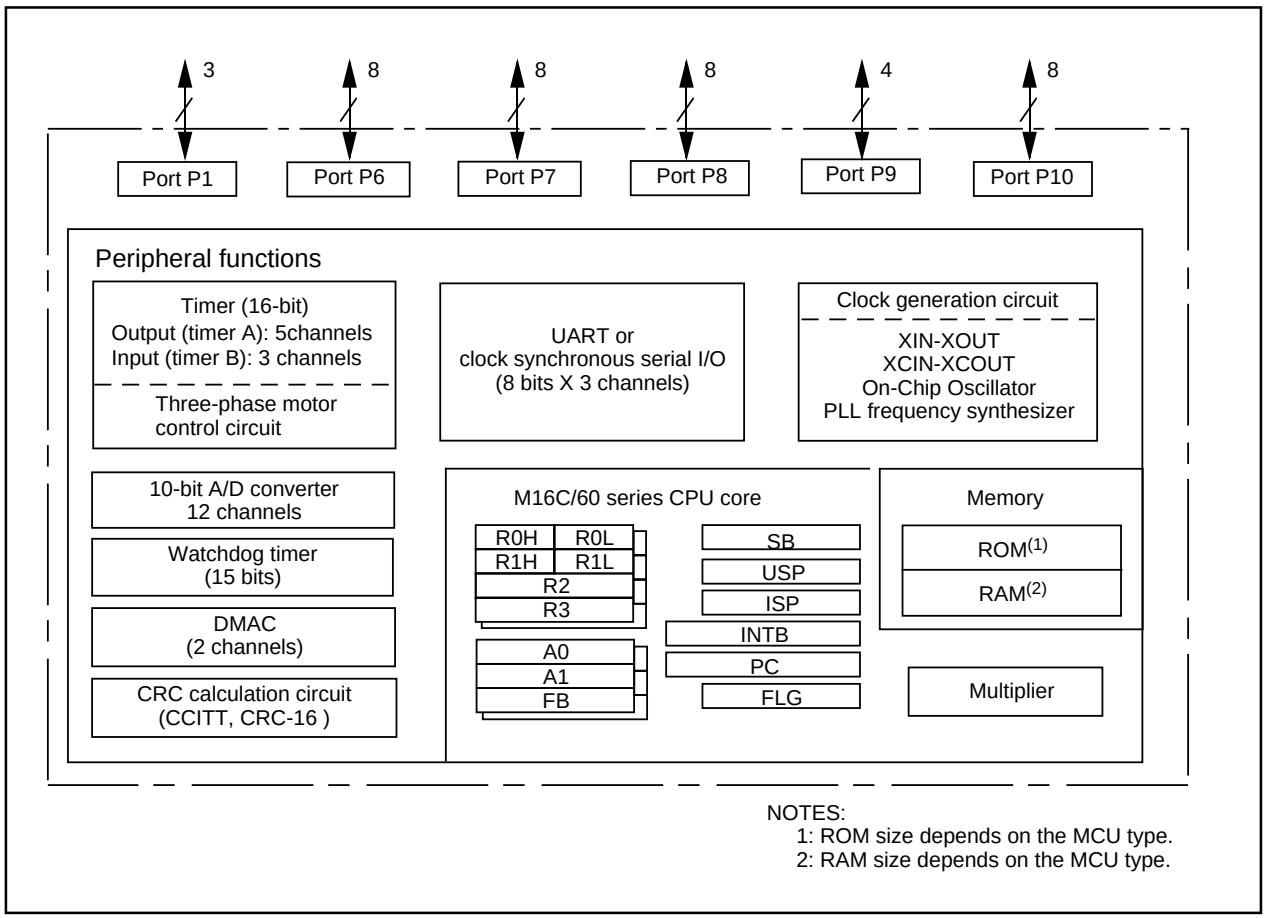


Figure 1.1 Block Diagram(48-pin Package)

Downloaded from Arrow.com.

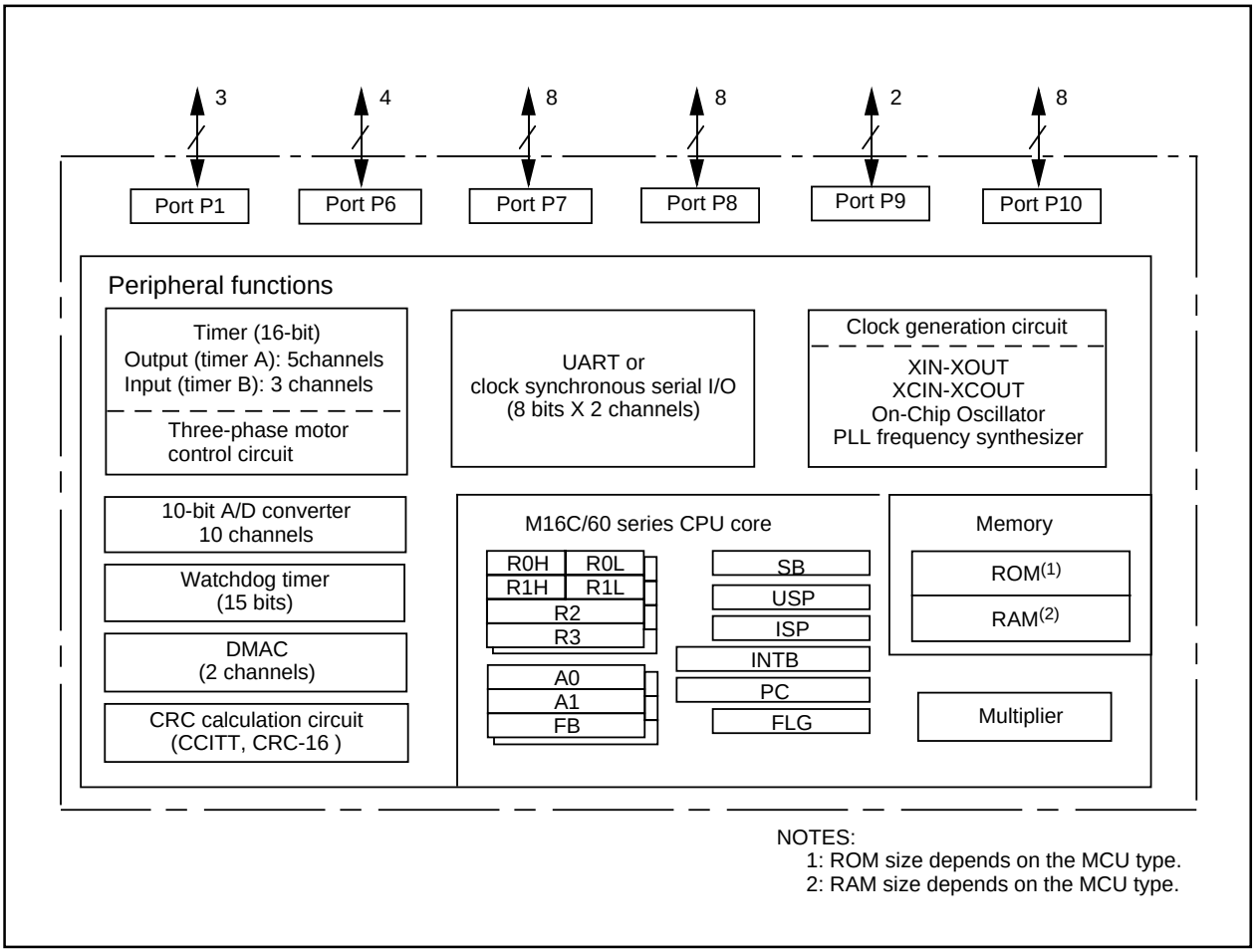


Figure 1.2 Block Diagram(42-pin Package)

Downloaded from Arrow.com.

1.4 Product List

Tables 1.3 to **1.6** lists product information, **Figure 1.3** shows a product numbering system, **Table 1.7** lists the product code, and **Figure 1.4** shows the marking.

Table 1.3 M16C/26A **Current as of Feb., 2007**

Type Number		ROM Capacity	RAM Capacity	Package Type	Remarks	Product Code
M30260F3AGP	(N)	24K + 4K	1K	PLQP0048KB-A (48P6Q-A)	Flash memory	U3, U5, U7, U9
M30260F6AGP	(N)	48K + 4K	2K			
M30260F8AGP	(N)	64K + 4K	2K			
M30263F3AFP	(N)	24K + 4K	1K	PRSP0042GA-B (42P2R)		U5, U9
M30263F6AFP	(N)	48K + 4K	2K			
M30263F8AFP	(N)	64K + 4K	2K			
M30260M3A-XXXGP	(N)	24K	1K	PLQP0048KB-A (48P6Q-A)	Mask ROM	U3, U5
M30260M6A-XXXGP	(N)	48K	2K			
M30260M8A-XXXGP	(N)	64K	2K			
M30263M3A-XXXFP	(N)	24K	1K	PRSP0042GA-B (42P2R)		U5
M30263M6A-XXXFP	(N)	48K	2K			
M30263M8A-XXXFP	(N)	64K	2K			

(N): New

Table 1.4 M16C/26B **Current as of Feb., 2007**

Type Number		ROM Capacity	RAM Capacity	Package Type	Remarks	Product Code
M30260F8BGP	(N)	64K + 4K	2K	PLQP0048KB-A (48P6Q-A)	Flash memory	U7
M30263F8BFP	(N)	64K + 4K	2K	PRSP0042GA-B (42P2R)		U9

(N): New

Table 1.5 M16C/26T T-ver. **Current as of Feb., 2007**

Type Number	ROM Capacity	RAM Capacity	Package Type	Remarks	Product Code
M30260F3TGP	24K + 4K	1K	PLQP0048KB-A (48P6Q-A)	Flash memory	U3, U7
M30260F6TGP	48K + 4K	2K			
M30260F8TGP	64K + 4K	2K			

NOTE:
1. Available in flash memory version only.

Table 1.6 M16C/26T V-ver. **Current as of Feb., 2007**

Type Number	ROM Capacity	RAM Capacity	Package	Remarks	Product Code
M30260F8VGP	64K + 4K	2K	PLQP0048KB-A (48P6Q-A)	Flash memory	U3, U7

NOTE:
1. Available in flash memory version only.

Downloaded from Arrow.com.

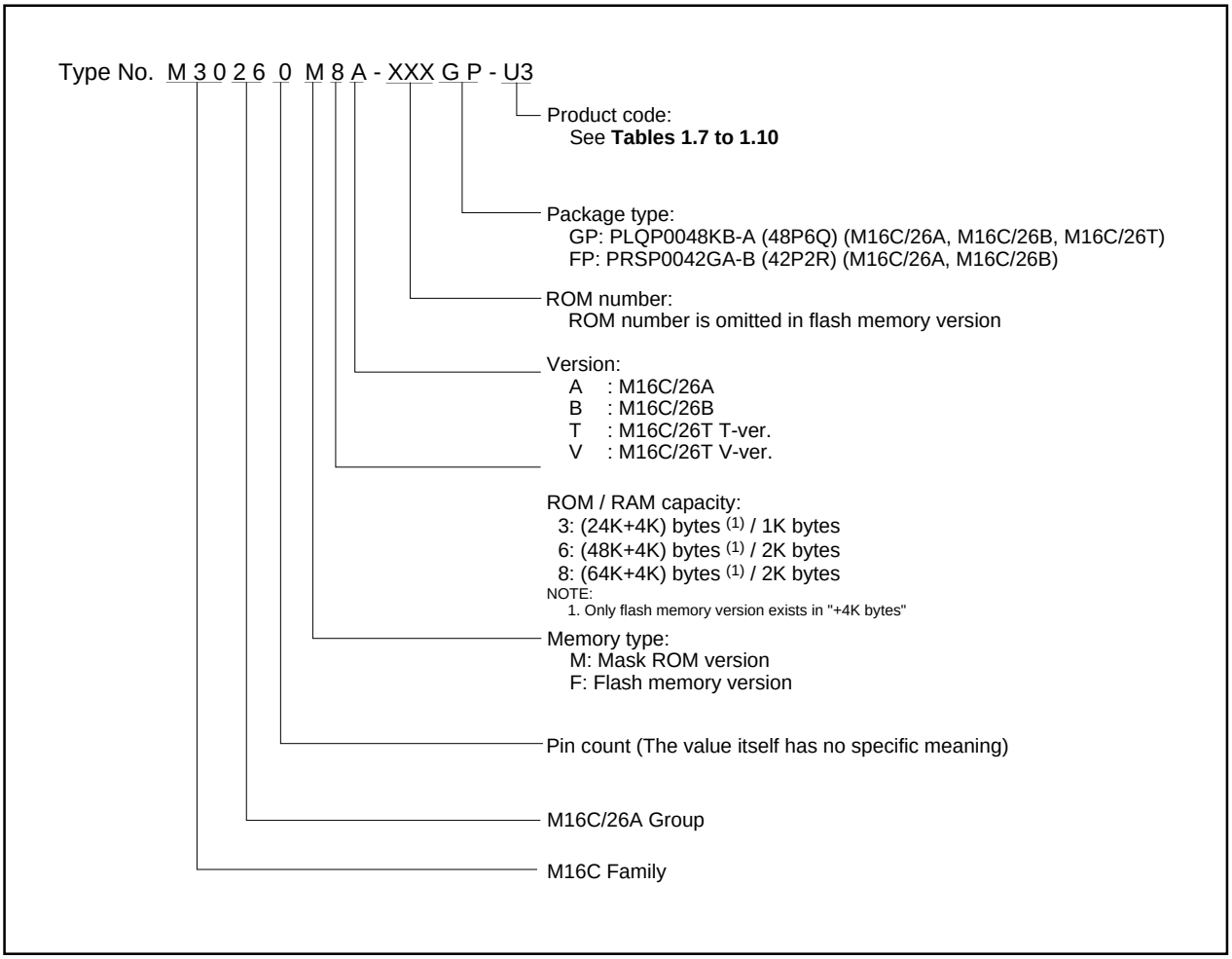


Figure 1.3 Product Numbering System

Downloaded from Arrow.com.

Table 1.7 Product Code (Flash Memory Version) - M16C/26A, M16C/26B

Product Code	Package	Internal ROM (Program Space: Blocks 0 to 3)		Internal ROM (Data Space: Blocks A and B)		Operating Ambient Temperature
		Program and Erase Endurance	Temperature Range	Program and Erase Endurance	Temperature Range	
U3	Lead free	100	0 to 60℃	100	0 to 60℃	-40 to 85℃
U5						-20 to 85℃
U7		1,000		10,000	-40 to 85℃	-40 to 85℃
U9					-20 to 85℃	-20 to 85℃

Table 1.8 Product Code (Mask ROM Version - M16C/26A)

Product Code	Package	Operating Ambient Temperature
U3	Lead free	-40℃ to 85℃
U5		-20℃ to 85℃

NOTE:
1. The lead contained products, D3, D5, D7, and D9 are put together with U3, U5, U7, and U9 respectively. Lead-free products can be mounted by both conventional Sn-Pb paste and Lead-free paste (Sn-Ag-Cu plating).

Table 1.9 Product Code (Flash Memory Version) - M16C/26T T-ver.

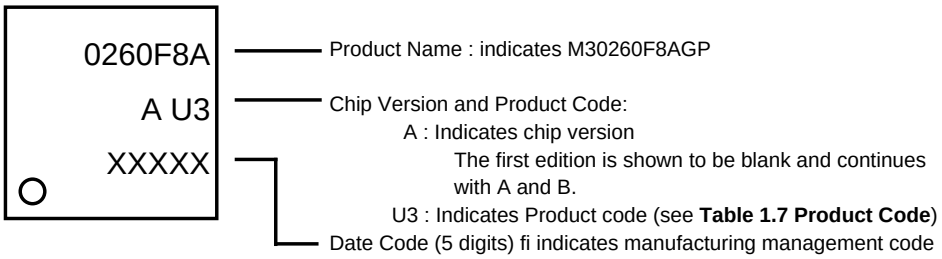
Product Code	Package	Internal ROM (Program Space: Blocks 0 to 3)		Internal ROM (Data Space: Blocks A and B)		Operating Ambient Temerature
		Programming and erasure endurance	Temperature range	Programming and erasure endurance	Temperature range	
U3	Lead free	100	0℃ to 60℃	100	-40℃ to 85℃	-40℃ to 85℃
U7		1,000		10,000		

Table 1.10 Product Code (Flash Memory Version) - M16C/26T V-ver.

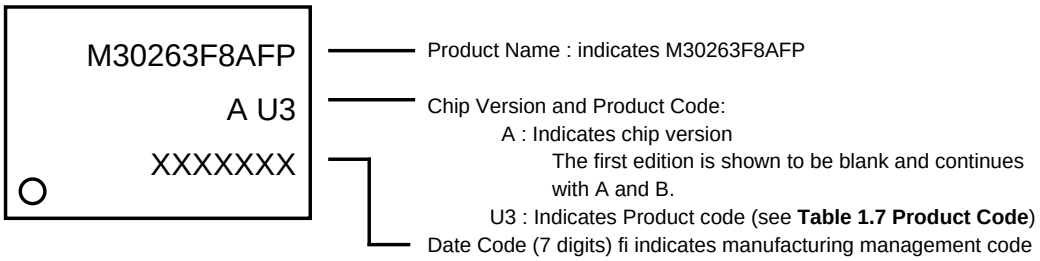
Product Code	Package	Internal ROM (Program Space: Blocks 0 to 3)		Internal ROM (Data Space: Blocks A and B)		Operating Ambient Temerature
		Programming and erasure endurance	Temperature range	Programming and erasure endurance	Temperature range	
U3	Lead free	100	0℃ to 60℃	100	-40℃ to 125℃	-40℃ to 125℃
U7		1,000		10,000		

Downloaded from Arrow.com.

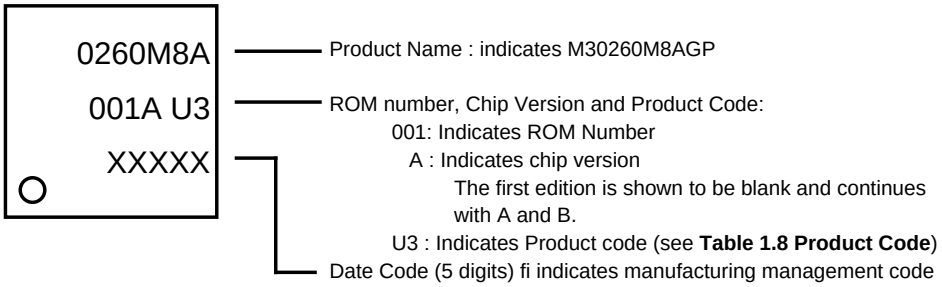
(1) Flash memory version, PLQP0048KB-A (48P6Q), M16C/26A, M16C/26B



(2) Flash memory version, PRSP0042GA-B (42P2R), M16C/26A, M16C/26B



(3) MASK ROM version, PLQP0048KB-A (48P6Q), M16C/26A



(4) MASK ROM version, PRSP0042GA-B (42P2R), M16C/26A

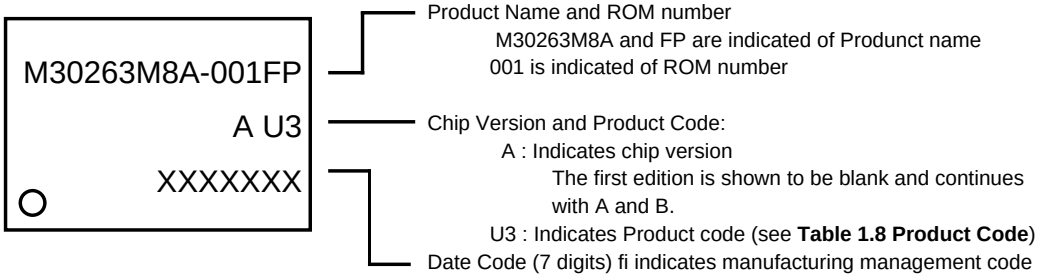
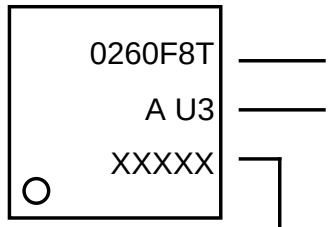


Figure 1.4 Marking Diagram (M16C/26A , M16C/26B)

Downloaded from Arrow.com.

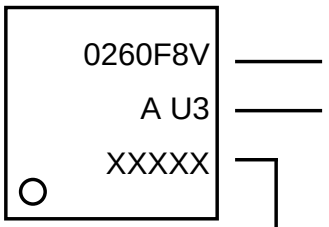
(1) Flash memory version, PLQP0048KB-A (48P6Q), M16C/26T T-ver.



Product Name : indicates M30260F8TGP

Chip Version and Product Code:
A : Indicates chip version
The first edition is shown to be blank and continues with A and B.
U3 : Indicates product code (see **Table 1.9 Product Code**)
Date Code (5 digits) fi indicates manufacturing management code

(2) Flash memory version, PLQP0048KB-A (48P6Q), M16C/26T V-ver.



Product Name : indicates M30260F8VGP

Chip Version and Product Code:
A : Indicates chip version
The first edition is shown to be blank and continues with A and B.
U3 : Indicates product code (see **Table 1.10 Product Code**)
Date Code (5 digits) fi indicates manufacturing management code

Figure 1.5 Marking Diagram (M16C/26T)

Downloaded from Arrow.com.

1.5 Pin Assignments

Figures 1.6 and **1.7** show the Pin Assignments (top view).

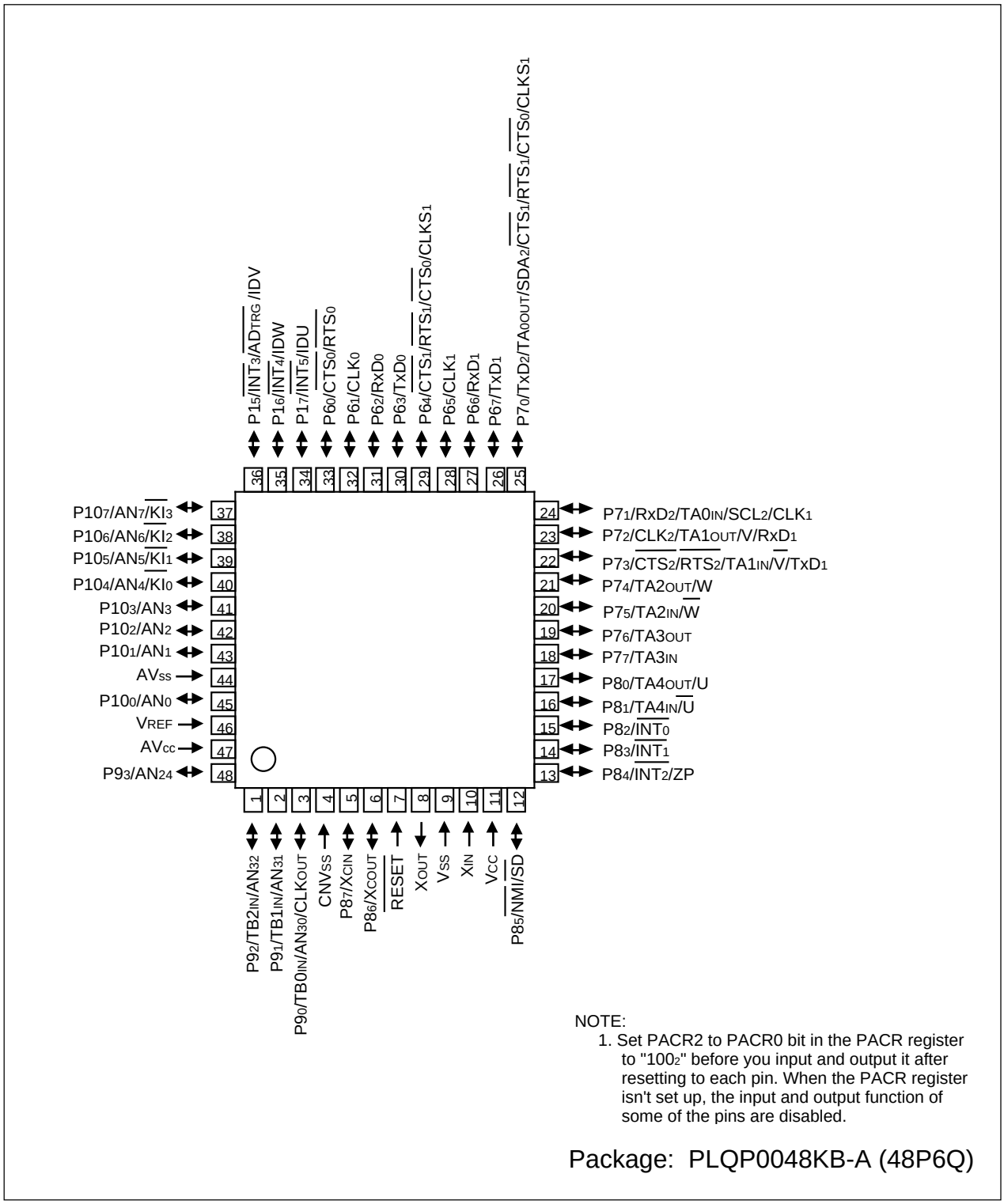


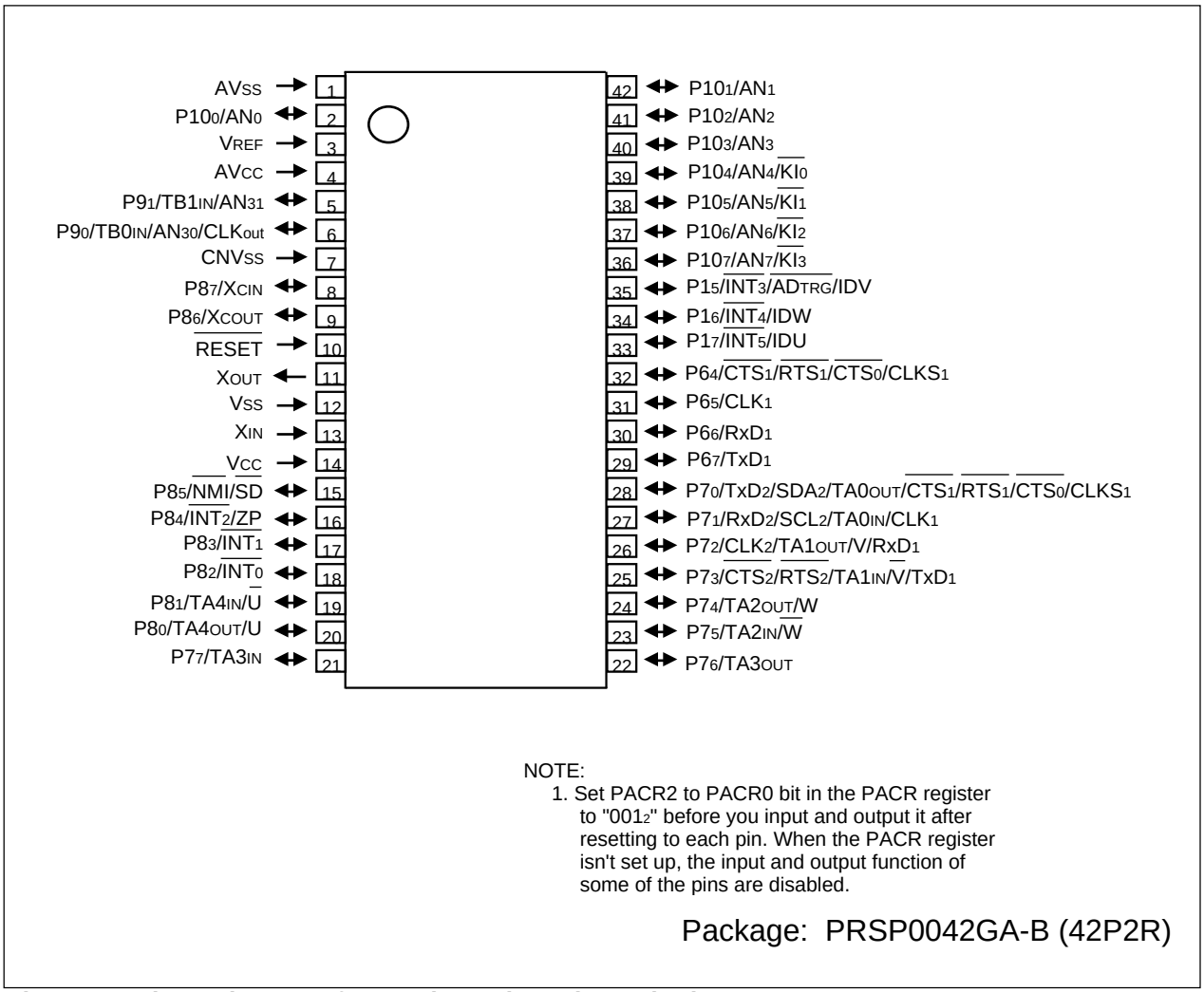
Figure 1.6 Pin Assignment for 48-Pin Package (Top View)

Downloaded from Arrow.com.

Table 1.11 Pin Characteristics for 48-Pin Package

Pin No.	Control Pin	Port	Interrupt Pin	Timer Pin	UART Pin	Analog Pin
1		P9$_2$		TB$_{2IN}$		AN3$_2$
2		P9$_1$		TB$_{1IN}$		AN3$_1$
3		P9$_0$		TB$_{0IN}$	CLK$_{OUT}$	AN3$_0$
4	CNVss					
5	X$_{CIN}$	P8$_7$				
6	X$_{COUT}$	P8$_6$				
7	$\overline{\text{RESET}}$					
8	X$_{OUT}$					
9	Vss					
10	X$_{IN}$					
11	Vcc					
12		P8$_5$	$\overline{\text{NMI}}$	$\overline{\text{SD}}$		
13		P8$_4$	$\overline{\text{INT}}_2$	ZP		
14		P8$_3$	$\overline{\text{INT}}_1$			
15		P8$_2$	$\overline{\text{INT}}_0$			
16		P8$_1$		TA$_{4IN}$ / $\overline{\text{U}}$		
17		P8$_0$		TA$_{4OUT}$ / U		
18		P7$_7$		TA$_{3IN}$		
19		P7$_6$		TA$_{3OUT}$		
20		P7$_5$		TA$_{2IN}$ / $\overline{\text{W}}$		
21		P7$_4$		TA$_{2OUT}$ / W		
22		P7$_3$		TA$_{1IN}$ / $\overline{\text{V}}$	$\overline{\text{CTS}}_2$ / $\overline{\text{RTS}}_2$ / TxD$_1$	
23		P7$_2$		TA$_{1OUT}$ / V	CLK$_2$ / RxD$_1$	
24		P7$_1$		TA$_{0IN}$	RxD$_2$ / SCL$_2$ / CLK$_1$	
25		P7$_0$		TA$_{0OUT}$	TxD$_2$ / SDA$_2$ / $\overline{\text{RTS}}_1$ / $\overline{\text{CTS}}_1$ / $\overline{\text{CTS}}_0$ / CLKS$_1$	
26		P6$_7$			TxD$_1$	
27		P6$_6$			RxD$_1$	
28		P6$_5$			CLK$_1$	
29		P6$_4$			$\overline{\text{RTS}}_1$ / $\overline{\text{CTS}}_1$/ $\overline{\text{CTS}}_0$ / CLKS$_1$	
30		P6$_3$			TxD$_0$	
31		P6$_2$			RxD$_0$	
32		P6$_1$			CLK$_0$	
33		P6$_0$			$\overline{\text{RTS}}_0$ / $\overline{\text{CTS}}_0$	
34		P1$_7$	$\overline{\text{INT}}_5$	IDU		
35		P1$_6$	$\overline{\text{INT}}_4$	IDW		
36		P1$_5$	$\overline{\text{INT}}_3$	IDV		$\overline{\text{ADTRG}}$
37		P10$_7$	$\overline{\text{KI}}_3$			AN$_7$
38		P10$_6$	$\overline{\text{KI}}_2$			AN$_6$
39		P10$_5$	$\overline{\text{KI}}_1$			AN$_5$
40		P10$_4$	$\overline{\text{KI}}_0$			AN$_4$
41		P10$_3$				AN$_3$
42		P10$_2$				AN$_2$
43		P10$_1$				AN$_1$
44	AVss					
45		P10$_0$				AN$_0$
46	V$_{REF}$					
47	AVcc					
48		P9$_3$				AN2$_4$

Downloaded from Arrow.com.



NOTE:
1. Set PACR2 to PACR0 bit in the PACR register to "001$_2$" before you input and output it after resetting to each pin. When the PACR register isn't set up, the input and output function of some of the pins are disabled.

Package: PRSP0042GA-B (42P2R)

Figure 1.7 Pin Assignment for 42-Pin Package (Top View)

Downloaded from Arrow.com.

Table 1.12 Pin Characteristics for 42-Pin Package

Pin No.	Control Pin	Port	Interrupt Pin	Timer Pin	UART Pin	Analog Pin
1	AVss					
2		P10$_0$				AN$_0$
3	VREF					
4	AVCC					
5		P9$_1$		TB$_{1IN}$		AN3$_1$
6		P9$_0$		TB$_{0IN}$	CLK$_{OUT}$	AN3$_0$
7	CNVss					
8	XCIN	P8$_7$				
9	XCOUT	P8$_6$				
10	$\overline{\text{RESET}}$					
11	XOUT					
12	Vss					
13	XIN					
14	VCC					
15		P8$_5$	$\overline{\text{NMI}}$	$\overline{\text{SD}}$		
16		P8$_4$	$\overline{\text{INT}}_2$	ZP		
17		P8$_3$	$\overline{\text{INT}}_1$			
18		P8$_2$	$\overline{\text{INT}}_0$			
19		P8$_1$		TA$_{4IN}$ / $\overline{\text{U}}$		
20		P8$_0$		TA$_{4OUT}$ / U		
21		P7$_7$		TA$_{3IN}$		
22		P7$_6$		TA$_{3OUT}$		
23		P7$_5$		TA$_{2IN}$ / $\overline{\text{W}}$		
24		P7$_4$		TA$_{2OUT}$ / W		
25		P7$_3$		TA$_{1IN}$ / $\overline{\text{V}}$	$\overline{\text{CTS}}_2$ / $\overline{\text{RTS}}_2$ / TxD$_1$	
26		P7$_2$		TA$_{1OUT}$ / V	CLK$_2$ / RxD$_1$	
27		P7$_1$		TA$_{0IN}$	RxD$_2$ / SCL$_2$ / CLK$_1$	
28		P7$_0$		TA$_{0OUT}$	TxD$_2$ / SDA$_2$ / $\overline{\text{RTS}}_1$ / $\overline{\text{CTS}}_1$ / $\overline{\text{CTS}}_0$ / CLKS$_1$	
29		P6$_7$			TxD$_1$	
30		P6$_6$			RxD$_1$	
31		P6$_5$			CLK$_1$	
32		P6$_4$			$\overline{\text{RTS}}_1$ / $\overline{\text{CTS}}_1$/ $\overline{\text{CTS}}_0$ / CLKS$_1$	
33		P1$_7$	$\overline{\text{INT}}_5$	IDU		
34		P1$_6$	$\overline{\text{INT}}_4$	IDW		
35		P1$_5$	$\overline{\text{INT}}_3$	IDV		$\overline{\text{ADTRG}}$
36		P10$_7$	$\overline{\text{KI}}_3$			AN$_7$
37		P10$_6$	$\overline{\text{KI}}_2$			AN$_6$
38		P10$_5$	$\overline{\text{KI}}_1$			AN$_5$
39		P10$_4$	$\overline{\text{KI}}_0$			AN$_4$
40		P10$_3$				AN$_3$
41		P10$_2$				AN$_2$
42		P10$_1$				AN$_1$

Downloaded from Arrow.com.

1.6 Pin Description

Table 1.13 Pin Description (48-Pin and 42-Pin Packages)

Classification	Pin Name	I/O Type	Description
Power Supply	VCC, VSS	I	Apply 0V to the Vss pin. Apply following voltage to the Vcc pin. 2.7 to 5.5 V (M16C/26A, M16C/26B), 3.0 to 5.5 V (M16C/26T T-ver.), 4.2 to 5.5 V (M16C/26T V-ver.)
Analog Power Supply	AVCC AVSS	I	Supplies power to the A/D converter. Connect the AVCC pin to VCC and the AVSS pin to VSS
Reset Input	$\overline{\text{RESET}}$	I	The MCU is in a reset state when "L" is applied to the $\overline{\text{RESET}}$ pin
CNVSS	CNVSS	I	Connect the CNVSS pin to VSS
Main Clock Input	XIN	I	I/O pins for the main clock oscillation circuit. Connect a ceramic resonator or crystal oscillator between XIN and XOUT. To apply external clock, apply it to XIN and leave XOUT open. If XIN is not used (for external oscillator or external clock), connect XIN pin to VCC and leave XOUT open
Main Clock Output	XOUT	O	
Sub Clock Input	XCIN	I	I/O pins for the sub clock oscillation circuit. Connect a crystal oscillator between XCIN and XCOUT
Sub Clock Output	XCOUT	O	
Clock Output	CLKOUT	O	Outputs the clock having the same frequency as f1, f8, f32, or fC
$\overline{\text{INT}}$ Interrupt Input	$\overline{\text{INT0}}$ to $\overline{\text{INT5}}$	I	Input pins for the $\overline{\text{INT}}$ interrupt. $\overline{\text{INT2}}$ can be used for Timer A Z-phase function
$\overline{\text{NMI}}$ Interrupt Input	$\overline{\text{NMI}}$	I	$\overline{\text{NMI}}$ interrupt input pin. $\overline{\text{NMI}}$ cannot be used as I/O port while the three-phase motor control is enabled. Apply a stable "H" to $\overline{\text{NMI}}$ after setting it's direction register to "0" when the three-phase motor control is enabled
Key Input Interrupt	$\overline{\text{KI0}}$ to $\overline{\text{KI3}}$	I	Input pins for the key input interrupt
Timer A	TA0OUT to TA4OUT	I/O	I/O pins for the timer A0 to A4
	TA0IN to TA4IN	I	Input pins for the timer A0 to A4
	ZP	I	Input pin for Z-phase
Timer B	TB0IN to TB1IN	I	Timer B0 to B1 input pins
Three-Phase Motor Control Timer Output	U, $\overline{\text{U}}$, V, $\overline{\text{V}}$, W, $\overline{\text{W}}$	O	Output pins for the three-phase motor control timer
	IDU, IDW, IDV, $\overline{\text{SD}}$	I/O	I/O pins for the three-phase motor control timer
Serial I/O	$\overline{\text{CTS1}}$ to $\overline{\text{CTS2}}$	I	Input pins to control data transmission
	$\overline{\text{RTS1}}$ to $\overline{\text{RTS2}}$	O	Output pins to control data reception
	CLK1 to CLK2	I/O	Inputs and outputs the transfer clock
	RxD1 to RxD2	I	Inputs serial data
	TxD1 to TxD2	O	Outputs serial data
	CLKS1	O	Output pin for transfer clock
Reference Voltage Input	VREF	I	Applies reference voltage to the A/D converter
A/D Converter	AN0 to AN7 AN30 to AN31	I	Analog input pins for the A/D converter
	$\overline{\text{ADTRG}}$	I	Input pin for an external A/D trigger
I/O Ports	P15 to P17	I/O	I/O ports for CMOS. Each port can be programmed for input or output under the control of the direction register. An input port can be set, by program, for a pull-up resistor available or for no pull-up resister available in 3-bit units
	P64 to P67 P70 to P77 P80 to P87 P100 to P107 P90 to P91	I/O	CMOS I/O ports which have a direction register determines an individual pin used as an input port or an output port. A pull-up resistor is selectable for every 4 input ports

I : Input O : Output I/O : Input and output

Downloaded from Arrow.com.

Table 1.13 Pin Description (48-pin packages only) (Continued)

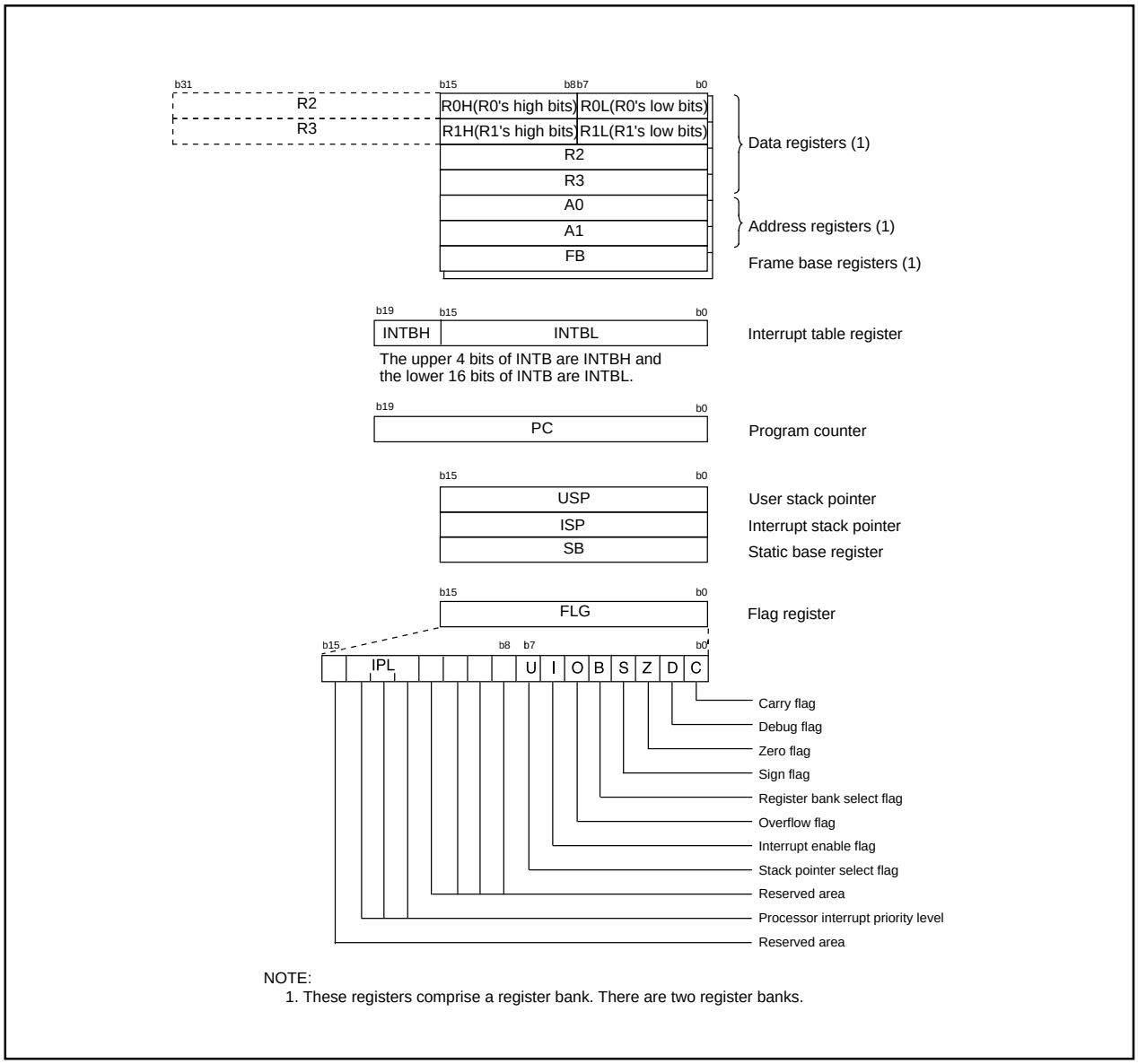
Classification	Pin Name	I/O Type	Description
Serial I/O	$\overline{CTS0}$	I	Inputs pin to control data transmission
	$\overline{RTS0}$	O	Output pin to control data reception
	CLK0	I/O	Inputs and outputs the transfer clock
	RxD0	I	Inputs serial data
	TxD0	O	Outputs serial data
Timer B	TB2IN	I	Timer B2 input pin
A/D Converter	$AN2_4$ $AN3_2$	I	Analog input pins for the A/D converter
I/O Ports	$P6_0$ to $P6_3$ $P9_2$ to $P9_3$	I/O	CMOS I/O ports which have a direction register determines an individual pin used as an input port or an output port. A pull-up resistor is selectable for every 4 input ports

I : Input O : Output I/O : Input and output

Downloaded from Arrow.com.

2. Central Processing Unit (CPU)

Figure 2.1 shows the CPU registers. The register bank is comprised of seven registers (R0, R1, R2, R3, A0, A1 and FB) out of 13 registers. There are two sets of register bank.



NOTE:
1. These registers comprise a register bank. There are two register banks.

Figure 2.1. CPU Register

2.1 Data Registers (R0, R1, R2 and R3)

The R0 register consists of 16 bits, and is used mainly for transfers and arithmetic/logic operations. R1 to R3 are the same as R0.

The R0 register can be separated between high (R0H) and low (R0L) for use as two 8-bit data registers. R1H and R1L are the same as R0H and R0L. Conversely, R2 and R0 can be combined for use as a 32-bit data register (R2R0). R3R1 is the same as R2R0.

2.2 Address Registers (A0 and A1)

The register A0 consists of 16 bits, and is used for address register indirect addressing and address register relative addressing. They also are used for transfers and arithmetic/logic operations. A1 is the same as A0.

In some instructions, registers A1 and A0 can be combined for use as a 32-bit address register (A1A0).

Downloaded from Arrow.com.

2.3 Frame Base Register (FB)

FB is configured with 16 bits, and is used for FB relative addressing.

2.4 Interrupt Table Register (INTB)

INTB is configured with 20 bits, indicating the start address of an interrupt vector table.

2.5 Program Counter (PC)

PC is configured with 20 bits, indicating the address of an instruction to be executed.

2.6 User Stack Pointer (USP) and Interrupt Stack Pointer (ISP)

Stack pointer (SP) comes in two types: USP and ISP, each configured with 16 bits.

Your desired type of stack pointer (USP or ISP) can be selected by the U flag of FLG.

2.7 Static Base Register (SB)

SB is configured with 16 bits, and is used for SB relative addressing.

2.8 Flag Register (FLG)

FLG consists of 11 bits, indicating the CPU status.

2.8.1 Carry Flag (C Flag)

This flag retains a carry, borrow, or shift-out bit that has occurred in the arithmetic/logic unit.

2.8.2 Debug Flag (D Flag)

The D flag is used exclusively for debugging purpose. During normal use, it must be set to 0.

2.8.3 Zero Flag (Z Flag)

This flag is set to 1 when an arithmetic operation resulted in 0; otherwise, it is 0.

2.8.4 Sign Flag (S Flag)

This flag is set to 1 when an arithmetic operation resulted in a negative value; otherwise, it is 0.

2.8.5 Register Bank Select Flag (B Flag)

Register bank 0 is selected when this flag is 0 ; register bank 1 is selected when this flag is 1.

2.8.6 Overflow Flag (O Flag)

This flag is set to 1 when the operation resulted in an overflow; otherwise, it is 0.

2.8.7 Interrupt Enable Flag (I Flag)

This flag enables a maskable interrupt.

Maskable interrupts are disabled when the I flag is 0, and are enabled when the I flag is 1.

The I flag is cleared to 0 when the interrupt request is accepted.

2.8.8 Stack Pointer Select Flag (U Flag)

ISP is selected when the U flag is 0; USP is selected when the U flag is 1.

The U flag is cleared to 0 when a hardware interrupt request is accepted or an INT instruction for software interrupt Nos. 0 to 31 is executed.

2.8.9 Processor Interrupt Priority Level (IPL)

IPL is configured with three bits, for specification of up to eight processor interrupt priority levels from level 0 to level 7.

If a requested interrupt has priority greater than IPL, the interrupt is enabled.

2.8.10 Reserved Area

When write to this bit, write 0. When read, its content is undefined.

Downloaded from Arrow.com.

3. Memory

Figure 3.1 is a memory map of the M16C/26A Group (M16C/26A, M16C/26B, M16C/26T). The M16C/26A Group provides 1-Mbyte address space addresses 00000$_{16}$ to FFFFF$_{16}$.

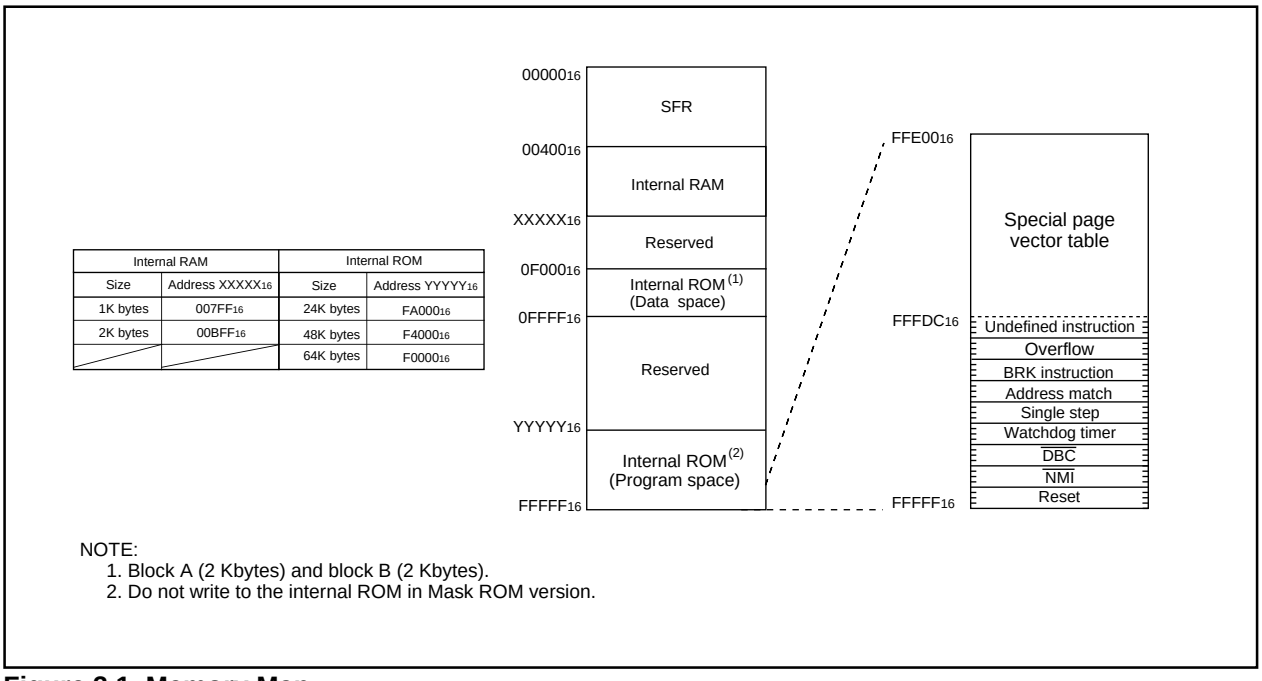
The internal ROM is allocated lower address, beginning with address FFFFF$_{16}$. For example, a 64-Kbyte internal ROM area is allocated in addresses F0000$_{16}$ to FFFFF$_{16}$. The flash memory version has two sets of 2-Kbyte internal ROM area, block A and block B, for data space. These blocks are allocated addresses F000$_{16}$ to FFFF$_{16}$.

The fixed interrupt vectors are allocated addresses FFFDC$_{16}$ to FFFFF$_{16}$ and they store the start address of each interrupt routine.

The internal RAM is allocated higher addresses, beginning with address 00400$_{16}$. For example, a 1-Kbyte internal RAM area is allocated in addresses 00400$_{16}$ to 007FF$_{16}$. The internal RAM is used for temporarily storing data. The area is also used as stacks when subroutines are called or interrupt requests are acknowledged.

The SFR is allocated addresses 00000$_{16}$ to 003FF$_{16}$. The peripheral function control registers are allocated here. All blank spaces within SFR location are reserved and cannot be accessed by users.

The special page vectors are allocated addresses FFE00$_{16}$ to FFFDB$_{16}$. They are used for the JMPS instruction and JSRS instruction. Refer to the Renesas publication **M16C/60 and M16C/20 Series Software Manual** for details.



Internal RAM		Internal ROM	
Size	Address XXXXX$_{16}$	Size	Address YYYYY$_{16}$
1K bytes	007FF$_{16}$	24K bytes	FA000$_{16}$
2K bytes	00BFF$_{16}$	48K bytes	F4000$_{16}$
		64K bytes	F0000$_{16}$

NOTE:
1. Block A (2 Kbytes) and block B (2 Kbytes).
2. Do not write to the internal ROM in Mask ROM version.

Figure 3.1 Memory Map

Downloaded from Arrow.com.

4. Special Function Registers (SFRs)

Table 4.1 SFR Information(1)[1]

Address	Register	Symbol	After reset
0000₁₆			
0001₁₆			
0002₁₆			
0003₁₆			
0004₁₆	Processor mode register 0	PM0	00₁₆
0005₁₆	Processor mode register 1	PM1	00001000₂
0006₁₆	System clock control register 0	CM0	01001000₂[5] 01101000₂(M16C/26T)
0007₁₆	System clock control register 1	CM1	00100000₂
0008₁₆			
0009₁₆	Address match interrupt enable register	AIER	XXXXXX00₂
000A₁₆	Protect register	PRCR	XX000000₂
000B₁₆			
000C₁₆	Oscillation stop detection register[2]	CM2	0X000000₂
000D₁₆			
000E₁₆	Watchdog timer start register	WDTS	XX₁₆
000F₁₆	Watchdog timer control register	WDC	00XXXXX₂
0010₁₆	Address match interrupt register 0	RMAD0	00₁₆
0011₁₆			00₁₆
0012₁₆			X0₁₆
0013₁₆			
0014₁₆	Address match interrupt register 1	RMAD1	00₁₆
0015₁₆			00₁₆
0016₁₆			X0₁₆
0017₁₆			
0018₁₆			
0019₁₆	Voltage detection register 1 [3, 4]	VCR1	00001000₂
001A₁₆	Voltage detection register 2 [3, 4]	VCR2	00₁₆
001B₁₆			
001C₁₆	PLL control register 0	PLC0	0001X010₂
001D₁₆			
001E₁₆	Processor mode register 2	PM2	XXX00000₂
001F₁₆	Low voltage detection interrupt register[4]	D4INT	00₁₆
0020₁₆	DMA0 source pointer	SAR0	XX₁₆
0021₁₆			XX₁₆
0022₁₆			XX₁₆
0023₁₆			
0024₁₆	DMA0 destination pointer	DAR0	XX₁₆
0025₁₆			XX₁₆
0026₁₆			XX₁₆
0027₁₆			
0028₁₆	DMA0 transfer counter	TCR0	XX₁₆
0029₁₆			XX₁₆
002A₁₆			
002B₁₆			
002C₁₆	DMA0 control register	DM0CON	00000X00₂
002D₁₆			
002E₁₆			
002F₁₆			
0030₁₆	DMA1 source pointer	SAR1	XX₁₆
0031₁₆			XX₁₆
0032₁₆			XX₁₆
0033₁₆			
0034₁₆	DMA1 destination pointer	DAR1	XX₁₆
0035₁₆			XX₁₆
0036₁₆			XX₁₆
0037₁₆			
0038₁₆	DMA1 transfer counter	TCR1	XX₁₆
0039₁₆			XX₁₆
003A₁₆			
003B₁₆			
003C₁₆	DMA1 control register	DM1CON	00000X00₂
003D₁₆			
003E₁₆			
003F₁₆			

NOTES:
1. The blank spaces are reserved. No access is allowed.
2. Bits CM27, CM21, and CM20 do not change at oscillation stop detection reset.
3. The VCR1 and VCR2 registers do not change at software reset, watchdog timer reset, and oscillation stop detection reset.
4. Registers VCR1, VCR2, and D4INT cannot be used in M16C/26T.
5. M16C/26A, M16C/26B

X : Undefined

Downloaded from Arrow.com.

Table 4.2 SFR Information(2)(1)

Address	Register	Symbol	After reset
0040₁₆			
0041₁₆			
0042₁₆			
0043₁₆			
0044₁₆	INT3 interrupt control register	INT3IC	XX00X000₂
0045₁₆			
0046₁₆			
0047₁₆			
0048₁₆	INT5 interrupt control register	INT5IC	XX00X000₂
0049₁₆	INT4 interrupt control register	INT4IC	XX00X000₂
004A₁₆	UART2 Bus collision detection interrupt control register	BCNIC	XXXXX000₂
004B₁₆	DMA0 interrupt control register	DM0IC	XXXXX000₂
004C₁₆	DMA1 interrupt control register	DM1IC	XXXXX000₂
004D₁₆	Key input interrupt control register	KUPIC	XXXXX000₂
004E₁₆	A/D conversion interrupt control register	ADIC	XXXXX000₂
004F₁₆	UART2 transmit interrupt control register	S2TIC	XXXXX000₂
0050₁₆	UART2 receive interrupt control register	S2RIC	XXXXX000₂
0051₁₆	UART0 transmit interrupt control register	S0TIC	XXXXX000₂
0052₁₆	UART0 receive interrupt control register	S0RIC	XXXXX000₂
0053₁₆	UART1 transmit interrupt control register	S1TIC	XXXXX000₂
0054₁₆	UART1 receive interrupt control register	S1RIC	XXXXX000₂
0055₁₆	TimerA0 interrupt control register	TA0IC	XXXXX000₂
0056₁₆	TimerA1 interrupt control register	TA1IC	XXXXX000₂
0057₁₆	TimerA2 interrupt control register	TA2IC	XXXXX000₂
0058₁₆	TimerA3 interrupt control register	TA3IC	XXXXX000₂
0059₁₆	TimerA4 interrupt control register	TA4IC	XXXXX000₂
005A₁₆	TimerB0 interrupt control register	TB0IC	XXXXX000₂
005B₁₆	TimerB1 interrupt control register	TB1IC	XXXXX000₂
005C₁₆	TimerB2 interrupt control register	TB2IC	XXXXX000₂
005D₁₆	INT0 interrupt control register	INT0IC	XX00X000₂
005E₁₆	INT1 interrupt control register	INT1IC	XX00X000₂
005F₁₆	INT2 interrupt control register	INT2IC	XX00X000₂
0060₁₆			
0061₁₆			
0062₁₆			
0063₁₆			
0064₁₆			
0065₁₆			
0066₁₆			
0067₁₆			
0068₁₆			
0069₁₆			
006A₁₆			
006B₁₆			
006C₁₆			
006D₁₆			
006E₁₆			
006F₁₆			
0070₁₆			
0071₁₆			
0072₁₆			
0073₁₆			
0074₁₆			
0075₁₆			
0076₁₆			
0077₁₆			
0078₁₆			
0079₁₆			
007A₁₆			
007B₁₆			
007C₁₆			
007D₁₆			
007E₁₆			
007F₁₆			

NOTE:
1. Blank spaces are reserved. No access is allowed.

X: Undefined

Downloaded from Arrow.com.

Table 4.3 SFR Information(3)(1)

Address	Register	Symbol	After reset
008016			
008116			
008216			
008316			
008416			
008516			
008616			
≈			≈
01B016			
01B116			
01B216			
01B316	Flash memory control register 4 (2)	FMR4	010000002
01B416			
01B516	Flash memory control register 1 (2)	FMR1	000XXX0X2
01B616			
01B716	Flash memory control register 0 (2)	FMR0	0116
01B816			
01B916			
01BA16			
01BB16			
01BC16			
01BD16			
01BE16			
01BF16			
≈			≈
025016			
025116			
025216			
025316			
025416			
025516			
025616			
025716			
025816			
025916			
025A16	Three phase protect control register	TPRC	0016
025B16			
025C16	On-chip oscillator control register	ROCR	X00001012
025D16	Pin assignment control register	PACR	0016
025E16	Peripheral clock select register	PCLKR	000000112
025F16			
≈			≈
033016			
033116			
033216			
033316			
033416			
033516			
033616			
033716			
033816			
033916			
033A16			
033B16			
033C16			
033D16			
033E16	NMI digital debounce register	NDDR	FF16
033F16	Port17 digital debounce register	P17DDR	FF16

NOTES:
1. Blank spaces are reserved. No access is allowed.
2. This register is included in the flash memory version.

X: Undefined

Downloaded from Arrow.com.

Table 4.4 SFR Information(4)(1)

Address	Register	Symbol	After reset
034016			
034116			
034216	Timer A1-1 register	TA11	XX16
034316			XX16
034416	Timer A2-1 register	TA21	XX16
034516			XX16
034616	Timer A4-1 register	TA41	XX16
034716			XX16
034816	Three phase PWM control register 0	INVC0	0016
034916	Three phase PWM control register 1	INVC1	0016
034A16	Three phase output buffer register 0	IDB0	001111112
034B16	Three phase output buffer register 1	IDB1	001111112
034C16	Dead time timer	DTT	XX16
034D16	Timer B2 Interrupt occurrence frequency set counter	ICTB2	XX16
034E16	Position-data-retain function control register	PDRF	XXXX00002
034F16			
035016			
035116			
035216			
035316			
035416			
035516			
035616			
035716			
035816	Port function control register	PFCR	001111112
035916			
035A16			
035B16			
035C16			
035D16			
035E16	Interrupt request cause select register 2	IFSR2A	XXXXXXX02(2)
035F16	Interrupt request cause select register	IFSR	0016
036016			
036116			
036216			
036316			
036416			
036516			
036616			
036716			
036816			
036916			
036A16			
036B16			
036C16			
036D16			
036E16			
036F16			
037016			
037116			
037216			
037316			
037416	UART2 special mode register 4	U2SMR4	0016
037516	UART2 special mode register 3	U2SMR3	000X0X0X2
037616	UART2 special mode register 2	U2SMR2	X00000002
037716	UART2 special mode register	U2SMR	X00000002
037816	UART2 transmit/receive mode register	U2MR	0016
037916	UART2 bit rate register	U2BRG	XX16
037A16	UART2 transmit buffer register	U2TB	XXXXXXXX2
037B16			XXXXXXXX2
037C16	UART2 transmit/receive control register 0	U2C0	000010002
037D16	UART2 transmit/receive control register 1	U2C1	000000102
037E16	UART2 receive buffer register	U2RB	XXXXXXXX2
037F16			XXXXXXXX2

NOTE:
1. Blank spaces are reserved. No access is allowed.
2. Write "1" to bit 0 after reset.

X : Undefined

Downloaded from Arrow.com.

Table 4.5 SFR Information(5)(1)

Address	Register	Symbol	After reset
038016	Count start flag	TABSR	0016
038116	Clock prescaler reset flag	CPSRF	0XXXXXXX2
038216	One-shot start flag	ONSF	0016
038316	Trigger select register	TRGSR	0016
038416	Up-dowm flag	UDF	0016
038516			
038616	Timer A0 register	TA0	XX16
038716			XX16
038816	Timer A1 register	TA1	XX16
038916			XX16
038A16	Timer A2 register	TA2	XX16
038B16			XX16
038C16	Timer A3 register	TA3	XX16
038D16			XX16
038E16	Timer A4 register	TA4	XX16
038F16			XX16
039016	Timer B0 register	TB0	XX16
039116			XX16
039216	Timer B1 register	TB1	XX16
039316			XX16
039416	Timer B2 register	TB2	XX16
039516			XX16
039616	Timer A0 mode register	TA0MR	0016
039716	Timer A1 mode register	TA1MR	0016
039816	Timer A2 mode register	TA2MR	0016
039916	Timer A3 mode register	TA3MR	0016
039A16	Timer A4 mode register	TA4MR	0016
039B16	Timer B0 mode register	TB0MR	00XX00002
039C16	Timer B1 mode register	TB1MR	00XX00002
039D16	Timer B2 mode register	TB2MR	00XX00002
039E16	Timer B2 special mode register	TB2SC	X00000002
039F16			
03A016	UART0 transmit/receive mode register	U0MR	0016
03A116	UART0 bit rate register	U0BRG	XX16
03A216	UART0 transmit buffer register	U0TB	XXXXXXXX2
03A316			XXXXXXXX2
03A416	UART0 transmit/receive control register 0	U0C0	000010002
03A516	UART0 transmit/receive control register 1	U0C1	000000102
03A616	UART0 receive buffer register	U0RB	XXXXXXXX2
03A716			XXXXXXXX2
03A816	UART1 transmit/receive mode register	U1MR	0016
03A916	UART1 bit rate register	U1BRG	XX16
03AA16	UART1 transmit buffer register	U1TB	XXXXXXXX2
03AB16			XXXXXXXX2
03AC16	UART1 transmit/receive control register 0	U1C0	000010002
03AD16	UART1 transmit/receive control register 1	U1C1	000000102
03AE16	UART1 receive buffer register	U1RB	XXXXXXXX2
03AF16			XXXXXXXX2
03B016	UART transmit/receive control register 2	UCON	X00000002
03B116			
03B216			
03B316			
03B416	CRC snoop address register	CRCSAR	XX16
03B516			00XXXXXX2
03B616	CRC mode register	CRCMR	0XXXXXX02
03B716			
03B816	DMA0 request cause select register	DM0SL	0016
03B916			
03BA16	DMA1 request cause select register	DM1SL	0016
03BB16			
03BC16	CRC data register	CRCD	XX16
03BD16			XX16
03BE16	CRC input register	CRCIN	XX16
03BF16			

NOTE:
1. Blank spaces are reserved. No access is allowed.

X : Undefined

Downloaded from Arrow.com.

Table 4.6 SFR Information(6)(1)

Address	Register	Symbol	After Reset
03C0₁₆	A/D register 0	AD0	XXXXXXXX₂
03C1₁₆			XXXXXXXX₂
03C2₁₆	A/D register 1	AD1	XXXXXXXX2
03C3₁₆			XXXXXXXX₂
03C4₁₆	A/D register 2	AD2	XXXXXXXX₂
03C5₁₆			XXXXXXXX₂
03C6₁₆	A/D register 3	AD3	XXXXXXXX₂
03C7₁₆			XXXXXXXX₂
03C8₁₆	A/D register 4	AD4	XXXXXXXX₂
03C9₁₆			XXXXXXXX₂
03CA₁₆	A/D register 5	AD5	XXXXXXXX₂
03CB₁₆			XXXXXXXX₂
03CC₁₆	A/D register 6	AD6	XXXXXXXX₂
03CD₁₆			XXXXXXXX₂
03CE₁₆	A/D register 7	AD7	XXXXXXXX₂
03CF₁₆			XXXXXXXX₂
03D0₁₆			
03D1₁₆			
03D2₁₆	A/D trigger control register	ADTRGCON	00₁₆
03D3₁₆	A/D status register 0	ADSTAT0	00000X00₂
03D4₁₆	A/D control register 2	ADCON2	00₁₆
03D5₁₆			
03D6₁₆	A/D control register 0	ADCON0	00000XXX₂
03D7₁₆	A/D control register 1	ADCON1	00₁₆
03D8₁₆			
03D9₁₆			
03DA₁₆			
03DB₁₆			
03DC₁₆			
03DD₁₆			
03DE₁₆			
03DF₁₆			
03E0₁₆			
03E1₁₆	Port P1 register	P1	XX₁₆
03E2₁₆			
03E3₁₆	Port P1 direction register	PD1	00₁₆
03E4₁₆			
03E5₁₆			
03E6₁₆			
03E7₁₆			
03E8₁₆			
03E9₁₆			
03EA₁₆			
03EB₁₆			
03EC₁₆	Port P6 register	P6	XX₁₆
03ED₁₆	Port P7 register	P7	XX₁₆
03EE₁₆	Port P6 direction register	PD6	00₁₆
03EF₁₆	Port P7 direction register	PD7	00₁₆
03F0₁₆	Port P8 register	P8	XX₁₆
03F1₁₆	Port P9 register	P9	XXXXXXXX₂
03F2₁₆	Port P8 direction register	PD8	00₁₆
03F3₁₆	Port P9 direction register	PD9	XXXX0000₂
03F4₁₆	Port P10 register	P10	XX₁₆
03F5₁₆			
03F6₁₆	Port P10 direction register	PD10	00₁₆
03F7₁₆			
03F8₁₆			
03F9₁₆			
03FA₁₆			
03FB₁₆			
03FC₁₆	Pull-up control register 0	PUR0	00₁₆
03FD₁₆	Pull-up control register 1	PUR1	00₁₆
03FE₁₆	Pull-up control register 2	PUR2	00₁₆
03FF₁₆	Port control register	PCR	00₁₆

NOTE:
1. Blank spaces are reserved. No access is allowed.

X: Undefined

Downloaded from Arrow.com.

5. Reset

There are four types of resets: a hardware reset, a software reset, an watchdog timer reset, and an oscillation stop detection reset.

5.1 Hardware Reset

There are two types of hardware resets: a hardware reset 1 and a hardware reset 2.

5.1.1 Hardware Reset 1

A reset is applied using the $\overline{\text{RESET}}$ pin. When an “L” signal is applied to the $\overline{\text{RESET}}$ pin while the power supply voltage is within the recommended operating condition, the pins are initialized (see Table 5.1.1.1 Pin Status When $\overline{\text{RESET}}$ Pin Level is “L”). The internal on-chip oscillator is initialized and used as CPU clock.

When the input level at the $\overline{\text{RESET}}$ pin is released from “L” to “H”, the CPU and SFR are initialized, and the program is executed starting from the address indicated by the reset vector. The internal RAM is not initialized. If the $\overline{\text{RESET}}$ pin is pulled “L” while writing to the internal RAM, the internal RAM becomes indeterminate.

Figure 5.1.1.1 shows the example reset circuit. Figure 5.1.1.2 shows the reset sequence. Table 5.1.1.1 shows the status of the other pins while the $\overline{\text{RESET}}$ pin is “L”. Figure 5.1.1.3 shows the CPU register status after reset. Refer to “SFR Map” for SFR status after reset.

1. When the power supply is stable
(1) Apply an “L” signal to the $\overline{\text{RESET}}$ pin.
(2) Wait td(ROC) or more.
(3) Apply an “H” signal to the $\overline{\text{RESET}}$ pin.

2. Power on
(1) Apply an “L” signal to the $\overline{\text{RESET}}$ pin.
(2) Let the power supply voltage increase until it meets the recommended operating condition.
(3) Wait td(P-R) or more until the internal power supply stabilizes.
(4) Wait td(ROC) or more.
(5) Apply an “H” signal to the $\overline{\text{RESET}}$ pin.

5.1.2 Hardware Reset 2

Note

M16C/26T does not use this function.

This reset is generated by the microcomputer’s internal voltage detection circuit. The voltage detection circuit monitors the voltage supplied to the V_{CC} pin.

If the VC26 bit in the VCR2 register is set to “1” (reset level detection circuit enabled), the microcomputer is reset when the voltage at the V_{CC} input pin drops below Vdet3.

Conversely, when the input voltage at the V_{CC} pin rises to Vdet3r or more, the pins and the CPU and SFR are initialized, and the program is executed starting from the address indicated by the reset vector. It takes about td(S-R) before the program starts running after Vdet3r is detected. The initialized pins and registers and the status thereof are the same as in hardware reset 1.

The microcomputer cannot exit stop mode by voltage down detection reset (hardware reset 2).

Downloaded from Arrow.com.

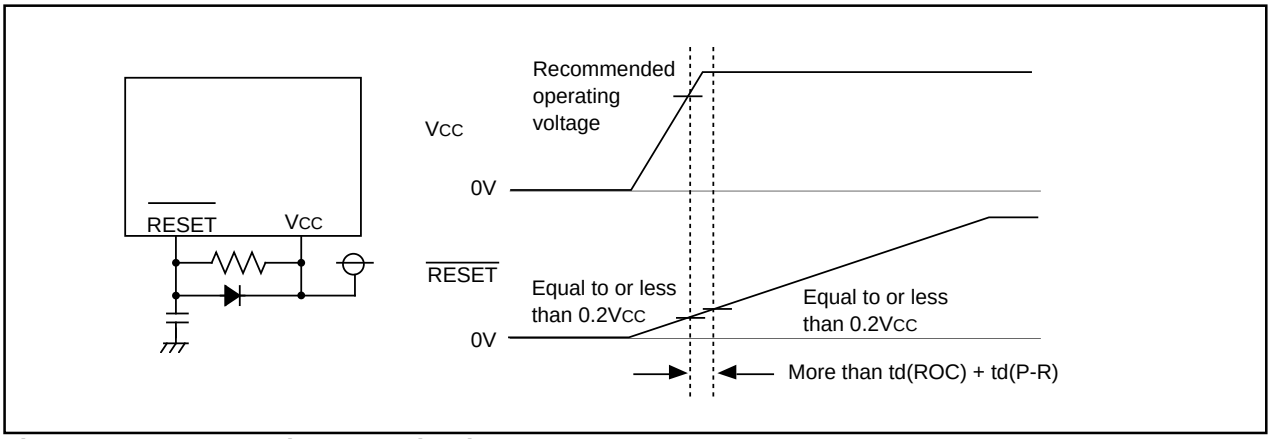


Figure 5.1.1.1. Example Reset Circuit

5.2 Software Reset

When the PM03 bit in the PM0 register is set to “1” (microcomputer reset), the microcomputer has its pins, CPU, and SFR initialized. Then the program is executed starting from the address indicated by the reset vector.

The device will reset using on-chip oscillator as the CPU clock.

At software reset, some SFR’s are not initialized. Refer to “SFR”.

5.3 Watchdog Timer Reset

When the PM12 bit in the PM1 register is “1” (reset when watchdog timer underflows), the microcomputer initializes its pins, CPU and SFR if the watchdog timer underflows.

The device will reset using on-chip oscillator as the system clock. Then the program is executed starting from the address indicated by the reset vector.

At watchdog timer reset, some SFR’s are not initialized. Refer to “SFR”.

5.4 Oscillation Stop Detection Reset

When the CM20 bit in the CM2 register is set to “1”(oscillation stop, re-oscillation detection function enabled) and the CM27 bit is set to “0” (reset at oscillation stop detection), the microcomputer initializes its pins, CPU and SFR, coming to a halt if it detects main clock oscillation circuit stop. Refer to the section “oscillation stop, re-oscillation detection function”.

At oscillation stop detection reset, some SFR’s are not initialized. Refer to the section “SFR”.

Downloaded from Arrow.com.

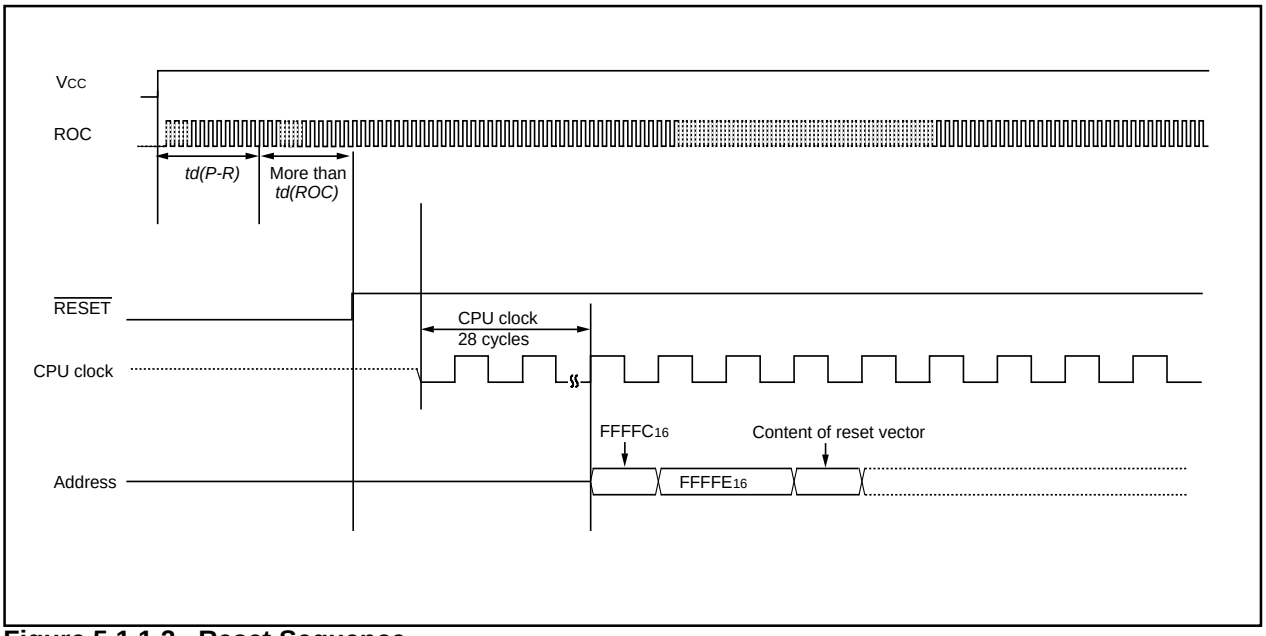


Figure 5.1.1.2. Reset Sequence

Table 5.1.1.1. Pin Status When $\overline{\text{RESET}}$ Pin Level is "L"

Pin name	Status
P1, P6 to P10	Input port (high impedance)

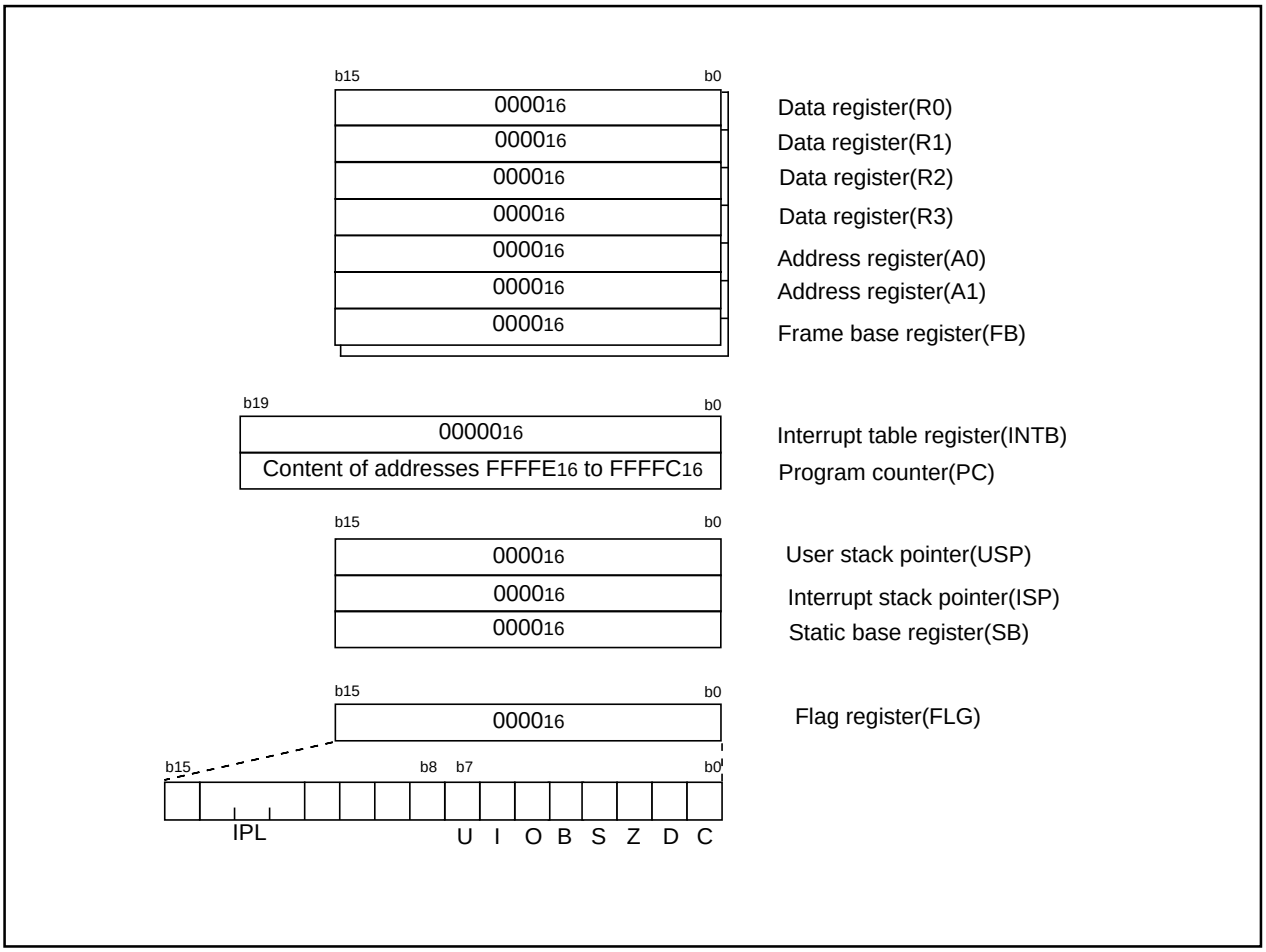


Figure 5.1.1.3. CPU Register Status After Reset

Downloaded from Arrow.com.

5.5 Voltage Detection Circuit

Note

VCC=5 V is assumed. Voltage Detection Circuit is not available in M16C/26T.

The voltage detection circuit has circuits to monitor the input voltage at the VCC pin, each checking the input voltage with respect to Vdet3, and Vdet4, respectively. Use the VC26 to VC27 bits in the VCR2 register to select whether or not to enable these circuits.

Use the reset level detection circuit for hardware reset 2.

The voltage down detection circuit can be set to detect whether the input voltage is equal to or greater than Vdet4 or less than Vdet4 by monitoring the VC13 bit in the VCR1 register. Furthermore, a voltage down detection interrupt can be generated.

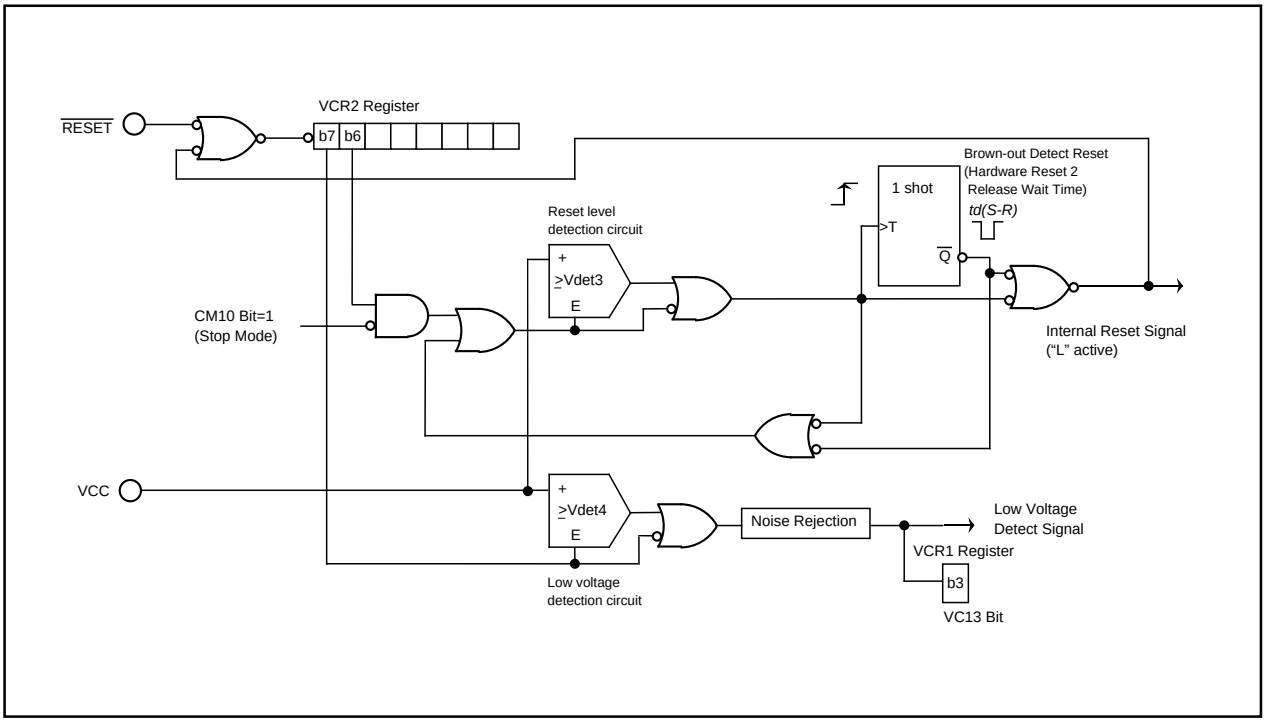


Figure 5.5.1. Voltage Detection Circuit Block

Downloaded from Arrow.com.

Voltage detection register 1

b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0		0	0	0

Symbol: VCR1 Address: 0019$_{16}$ After reset (2): 00001000$_2$

Bit symbol	Bit name	F unction	RW
$\overline{\text{(b2-b0)}}$	Reserved bit	Must set to “0”	RW
VC13	Voltage down monitor flag (1)	0:Vcc < Vdet4 1:Vcc ≥ Vdet4	RO
$\overline{\text{(b7-b4)}}$	Reserved bit	Must set to “0”	RW

NOTES:
1. The VC13 bit is useful when the VC27 bit in the VCR2 register is set to "1" (voltage down detection circuit enable). The VC13 bit is always "1" (VCC ≥ Vdet4) when the VC27 bit in the VCR2 register is set to "0" (voltage down detection circuit disable).
2. This register does not change at software reset, watchdog timer reset and oscillation stop detection reset.

Voltage detection register 2 (1)

b7	b6	b5	b4	b3	b2	b1	b0
		0	0	0	0	0	0

Symbol: VCR2 Address: 001A$_{16}$ After reset (5): 00$_{16}$

Bit symbol	Bit name	Function	RW
$\overline{\text{(b5-b0)}}$	Reserved bit	Must set to “0”	RW
VC26	Reset level monitor bit (2, 3, 6)	0: Disable reset level detection circuit 1: Enable reset level detection circuit	RW
VC27	Voltage down monitor bit (4, 6)	0: Disable voltage down detection circuit 1: Enable voltage down detection circuit	RW

NOTES:
1. Write to this register after setting the PRC3 bit in the PRCR register to “1” (write enable).
2. When not in stop mode, to use hardware reset 2, set the VC26 bit to “1” (reset level detection circuit enable).
3. VC26 bit is disabled in stop mode. (The microcomputer is not reset even if the voltage input to Vcc pin becomes lower than Vdet3.)
4. When the VC13 bit in the VCR1 register and D42 bit in the D4INT register are used or the D40 bit is set to “1” (voltage down detection interrupt enable), set the VC27 bit to “1” (voltage down detection circuit enable).
5. This register does not change at software reset, watchdog timer reset and oscillation stop detection reset.
6. The detection circuit does not start operation until td(E-A) elapses after the VC26 bit, or VC27 bit are set to “1”.

Voltage down detection interrupt register (1)

b7	b6	b5	b4	b3	b2	b1	b0
X	X						

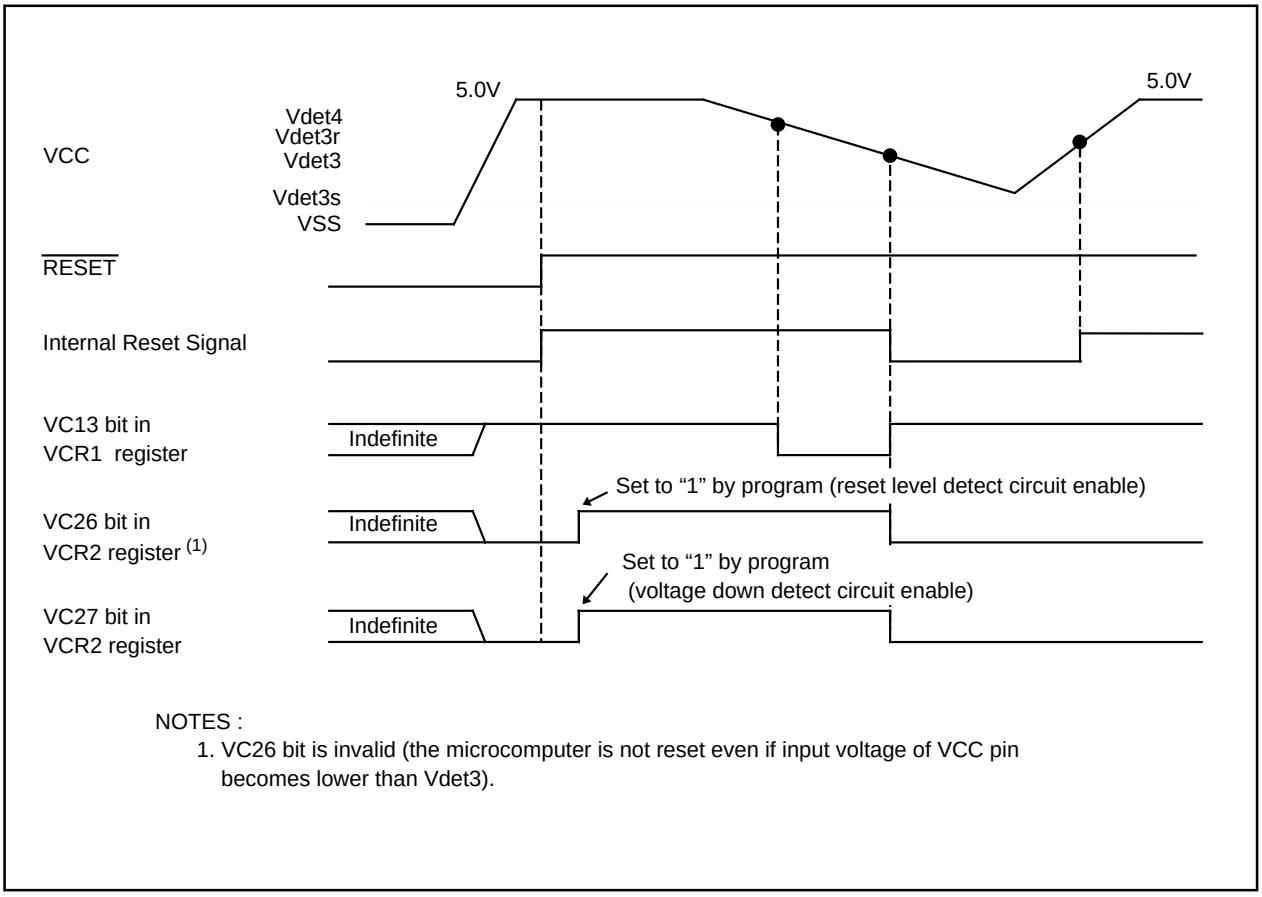
Symbol: D4INT Address: 001F$_{16}$ After reset: 00$_{16}$

Bit symbol	Bit name	Function	RW
D40	Voltage down detection interrupt enable bit (5)	0 : Disable 1 : Enable	RW
D41	STOP mode deactivation control bit (4)	0: Disable (do not use the voltage down detection interrupt to get out of stop mode) 1: Enable (use the voltage down detection interrupt to get out of stop mode)	RW
D42	Voltage change detection flag (2)	0: Not detected 1: Vdet4 passing detection	RW (3)
D43	WDT overflow detect flag	0: Not detected 1: Detected	RW (3)
DF0	Sampling clock select bit	b5b4 00 : CPU clock divided by 8 01 : CPU clock divided by 16 10 : CPU clock divided by 32 11 : CPU clock divided by 64	RW
DF1			RW
— (b7-b6)	Nothing is assigned. When write, set to “0”. When read, its content is “0”.		—

NOTES:
1. Write to this register after setting the PRC3 bit in the PRCR register to “1” (write enable).
2. Useful when the VC27 bit in the VCR2 register is set to “1” (voltage down detection circuit enabled). If the VC27 bit is set to “0” (voltage down detection circuit disable), the D42 bit is set to “0” (Not detect).
3. This bit is set to “0” by writing a “0” in a program. (Writing a “1” has no effect.)
4. If the voltage down detection interrupt needs to be used to get out of stop mode again after once used for that purpose, reset the D41 bit by writing a “0” and then a “1”.
5. The D40 bit is effective when the VC27 bit in the VCR2 register is set to “1”. To set the D40 bit to “1”, follow the procedure described below.
   (1) Set the VC27 bit to “1”.
   (2) Wait for td(E-A) until the detection circuit is actuated.
   (3) Wait for the sampling time (refer to “Table 5.5.1.2 Sampling Clock Periods”).
   (4) Set the D40 bit to “1”.

Figure 5.5.2. VCR1 Register, VCR2 Register, and D4INT Register

Downloaded from Arrow.com.



NOTES :
1. VC26 bit is invalid (the microcomputer is not reset even if input voltage of VCC pin becomes lower than Vdet3).

Figure 5.5.3. Typical Operation of Hardware Reset 2

Downloaded from Arrow.com.

5.5.1 Voltage Down Detection Interrupt

If the D40 bit in the D4INT register is set to “1” (voltage down detection interrupt enabled), the voltage down detection interrupt request is generated when the voltage applied to the VCC pin crosses the Vdet4 voltage level. The voltage down detection interrupt shares the same interrupt vector with the watchdog timer interrupt and oscillation stop, re-oscillation detection interrupt.

Set the D41 bit in the D4INT register to “1” (enabled) to use the voltage down detection interrupt to exit stop mode.

The D42 bit in the D4INT register is set to “1” as soon as the voltage applied to the VCC pin reaches Vdet4 due to the voltage rise and voltage drop. When the D42 bit changes “0” to “1”, the voltage down detection interrupt request is generated. Set the D42 bit to “0” by program. However, when the D41 bit is set to “1” and the microcomputer is in stop mode, the voltage down detection interrupt request is generated regardless of the D42 bit state if the voltage applied to the VCC pin is detected to be above Vdet4. The microcomputer then exits stop mode.

Table 5.5.1.1 shows how the voltage down detection interrupt request is generated.

The DF1 to DF0 bits in the D4INT register determine the sampling period that detects the voltage applied to the VCC pin reaches Vdet4. Table 5.5.1.2 shows the sampling periods.

Table 5.5.1.1 Voltage Down Detection Interrupt Request Generation Conditions

Operation Mode	VC27 Bit	D40 Bit	D41 Bit	D42 Bit	CM02 Bit	VC13 Bit
Normal Operation Mode(1)	1	1	—	0 to 1	—	0 to 1(3)
						1 to 0(3)
Wait Mode(2)			—	0 to 1	0	0 to 1(3)
						1 to 0(3)
				—	1	0 to 1
Stop Mode(2)			1	—	0	0 to 1

– : “0”or “1”

NOTES:
1. The status except the wait mode and stop mode is handled as the normal mode.(Refer to **7. Clock generating circuit**)
2. Refer to **5.5.2 Limitations on stop mode**, **5.5.3 Limitations on wait mode**.
3. An interrupt request for voltage reduction is generated a sampling time after the value of the VC13 bit has changed. See the **Figure 5.5.1.2 Voltage Down Detection Interrupt Generation Circuit Operation Example** for details.

Table 5.5.1.2 Sampling Periods

CPU Clock (MHz)	Sampling Period (µs)			
	DF1 to DF0=00 (CPU clock divided by 8)	DF1 to DF0=01 (CPU clock divided by 16)	DF1 to DF0=10 (CPU clock divided by 32)	DF1 to DF0=11 (CPU clock divided by 64)
16	3.0	6.0	12.0	24.0

Downloaded from Arrow.com.

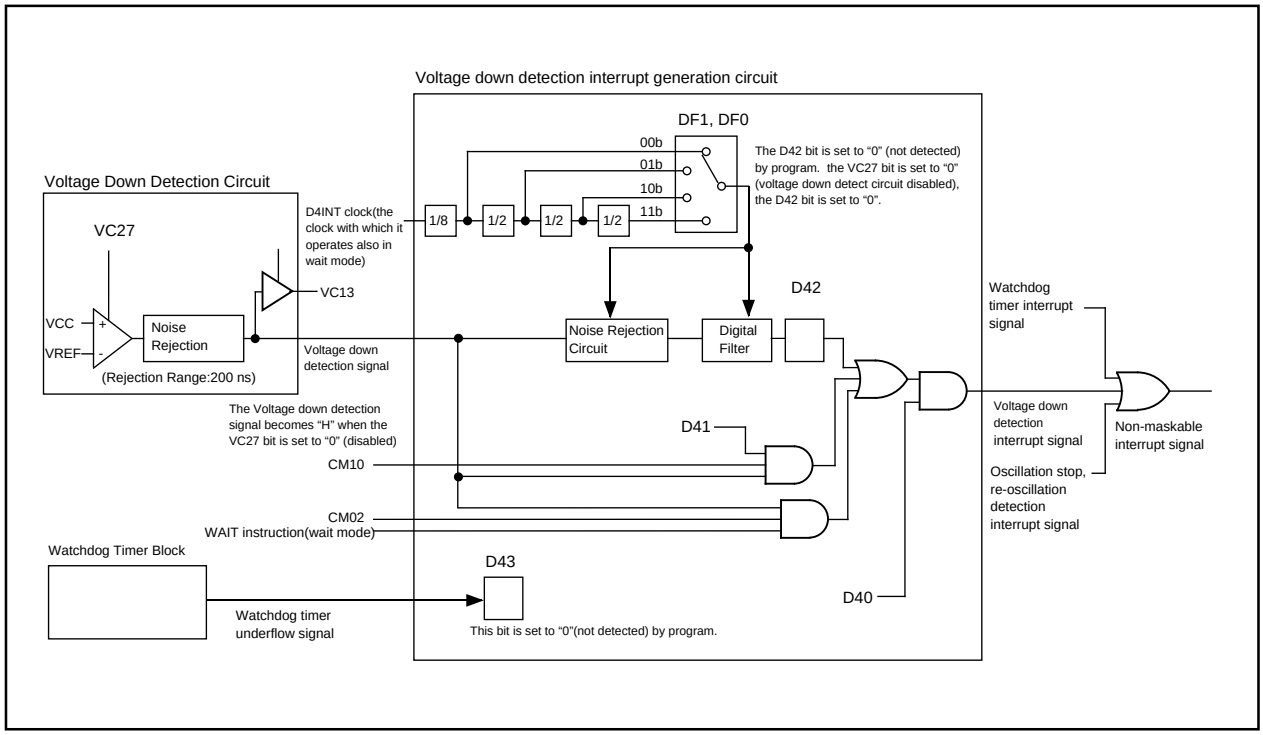
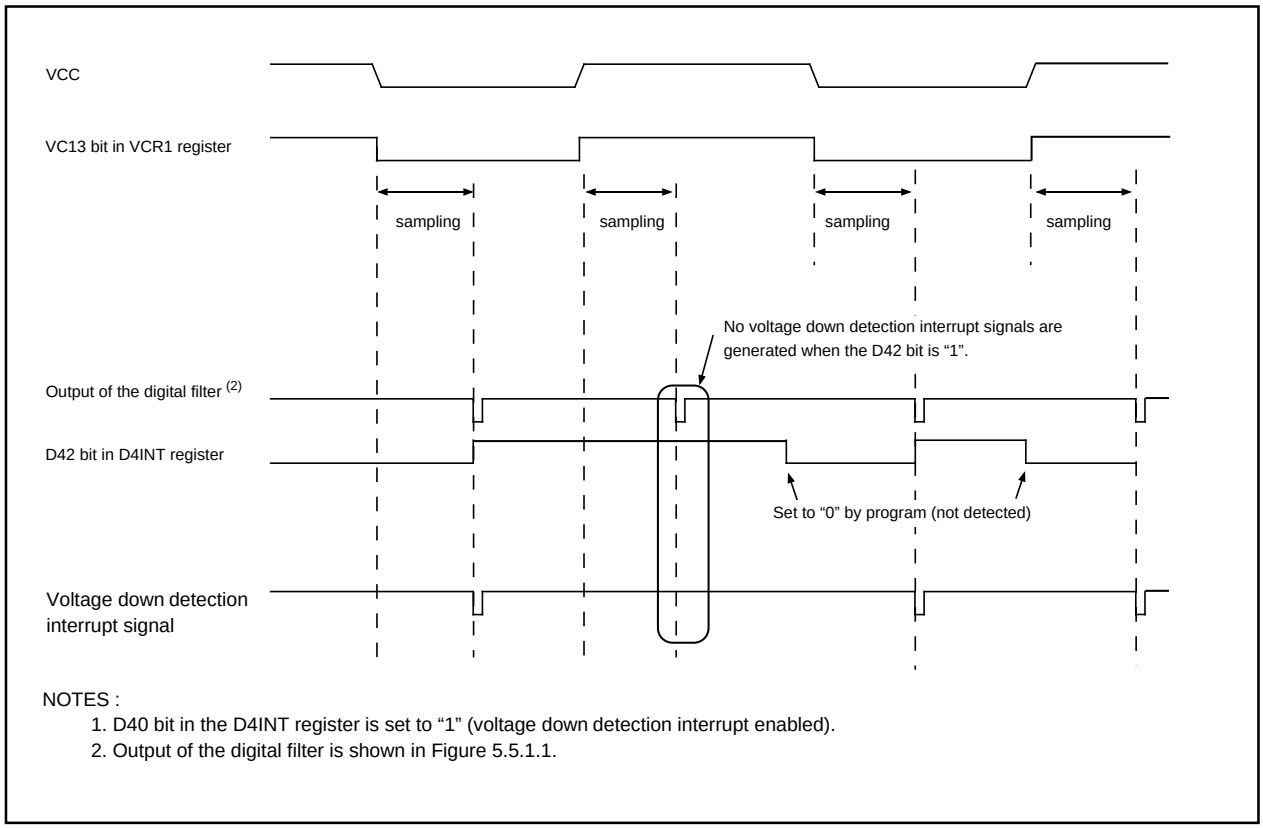


Figure 5.5.1.1 Power Supply Down Detection Interrupt Generation Block



NOTES :
1. D40 bit in the D4INT register is set to "1" (voltage down detection interrupt enabled).
2. Output of the digital filter is shown in Figure 5.5.1.1.

Figure 5.5.1.2 Power Supply Down Detection Interrupt Generation Circuit Operation Example

Downloaded from Arrow.com.

5.5.2 Limitations on Exiting Stop Mode

The voltage down detection interrupt is immediately generated and the microcomputer exits stop mode if the CM10 bit in the CM1 register is set to “1” under the conditions below.

- the VC27 bit in the VCR2 register is set to “1” (voltage down detection circuit enabled),
- the D40 bit in the D4INT register is set to “1” (voltage down detection interrupt enabled),
- the D41 bit in the D4INT register is set to “1” (voltage down detection interrupt is used to exit stop mode), and
- the voltage applied to the VCC pin is higher than Vdet4 (the VC13 bit in the VCR1 register is “1”)

If the microcomputer is set to enter stop mode when the voltage applied to the VCC pin drops below Vdet4 and to exit stop mode when the voltage applied rises to Vdet4 or above, set the CM10 bit to “1” when VC13 bit is “0” (VCC < Vdet4).

5.5.3 Limitations on Exiting Wait Mode

The voltage down detection interrupt is immediately generated and the microcomputer exits wait mode If WAIT instruction is executed under the conditions below.

- the CM02 bit in the CM0 register is set to “1” (stop peripheral function clock),
- the VC27 bit in the VCR2 register is set to “1” (voltage down detection circuit enabled),
- the D40 bit in the D4INT register is set to “1” (voltage down detection interrupt enabled),
- the D41 bit in the D4INT register is set to “1” (voltage down detection interrupt is used to exit wait mode), and
- the voltage applied to the VCC pin is higher than Vdet4 (the VC13 bit in the VCR1 register is “1”)

If the microcomputer is set to enter wait mode when the voltage applied to the VCC pin drops below Vdet4 and to exit wait mode when the voltage applied rises to Vdet4 or above, perform WAIT instruction when VC13 bit is “0” (VCC < Vdet4).

Downloaded from Arrow.com.

6. Processor Mode

The microcomputer supports single-chip mode only. **Figures 6.1** and **6.2** show the associated registers.

Processor Mode Register 0 (1)

b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0		0	0	0

Symbol	Address	After Reset
PM0	0004$_{16}$	00$_{16}$

Bit Symbol	Bit Name	Function	RW
$\overline{\text{(b2-b0)}}$	Reserved bit	Set to "0"	RW
PM03	Software reset bit	The microcomputer is reset when this bit is set to "1". When read, its content is "0".	RW
$\overline{\text{(b7-b4)}}$	Reserved bit	Set to "0"	RW

NOTES:
1. Rewrite the PM0 register after the PRC1 bit in the PRCR register is set to "1" (write enable).

Processor Mode Register 1 (1)

b7	b6	b5	b4	b3	b2	b1	b0
	0	0	0	1		0	

Symbol	Address	After Reset
PM1	0005$_{16}$	00001000$_2$

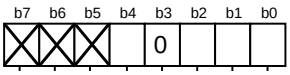
Bit Symbol	Bit Name	Function	RW
PM10	Flash data block access bit (2)	0: Disabled 1: Enabled (3)	RW
$\overline{\text{(b1)}}$	Reserved bit	Set to "0"	RW
PM12	Watchdog timer function select bit	0 : Watchdog timer interrupt 1 : Watchdog timer reset (4)	RW
$\overline{\text{(b3)}}$	Reserved bit	Set to "1"	RW
$\overline{\text{(b6-b4)}}$	Reserved bit	Set to "0"	RW
PM17	Wait bit (5)	0 : No wait state 1 : Wait state (1 wait)	RW

NOTES:
1. Rewrite the PM1 register after the PRC1 bit in the PRCR register is set to "1" (write enable).
2. To access the two 2K-byte data spaces in data block A and data block B, set the PM10 bit to "1". The PM10 bit is not available in mask version.
3. When the FMR01 bit in the FMR0 register is set to "1" (enables CPU rewrite mode), the PM10 bit is automatically set to "1".
4. Set the PM12 bit to "1" by program. (Writing "0" by program has no effect)
5. When the PM17 bit is set to "1" (wait state), one wait is inserted when accessing the internal RAM or the internal ROM.

Figure 6.1 PM0 Register, PM1 Register

Downloaded from Arrow.com.

Processeor Mode Register 2 (1)



Symbol: PM2 | Address: 001E16 | After Reset: XXX000002

Bit Symbol	Bit Name	Function	RW
PM20	Specifying wait when accessing SFR during PLL operation(2)	0: 2 wait 1: 1 wait	RW
PM21	System clock protective bit(3,4)	0: Clock is protected by PRCR register 1: Clock modification disabled	RW
PM22	WDT count source protective bit (3,5)	0: CPU clock is used for the watchdog timer count source 1: On-chip oscillator clock is used for the watchdog timer count source	RW
(b3)	Reserved bit	Set to “0”	RW
PM24	P85/$\overline{\text{NMI}}$ configuration bit(6,7)	0: P85 function ($\overline{\text{NMI}}$ disable) 1: $\overline{\text{NMI}}$ function	RW
(b7-b5)	Nothing is assigned. When write, set to“0”. When read,its content is indeterminate		—

NOTES:
1. Write to this register after setting the PRC1 bit in the PRCR register to “1” (write enable).
2. The PM20 bit become effective when PLC07 bit in the PLC0 register is set to "1" (PLL on). Change the PM20 bit when the PLC07 bit is set to "0" (PLL off). Set the PM20 bit to "0" (2 waits) when PLL clock > 16 MHz.
3. Once this bit is set to “1”, it cannot be set to “0” by program.
4. Writing to the following bits has no effect when the PM21 bit is set to “1”:
   CM02 bit in the CM0 register
   CM05 bit in the CM0 register (main clock is not halted)
   CM07 bit in the CM0 register (CPU clock source does not change)
   CM10 bit in the CM1 register (stop mode is not entered)
   CM11 bit in the CM1 register (CPU clock source does not change)
   CM20 bit in the CM2 register (oscillation stop, re-oscillation detection function settings do not change)
   All bits in the PLC0 register (PLL frequency synthesizer setting do not change)
   When the PM21 bit is set to "1", do not execute the WAIT instruction.
5. Setting the PM22 bit to “1” results in the following conditions:
   - The on-chip oscillator continues oscillating even if the CM21 bit in the CM2 register is set to "0" (main clock or PLL clock) (system clock of count source selected by the CM21 bit is valid)
   - The on-chip oscillator starts oscillating, and the on-chip oscillator clock becomes the watchdog timer count source.
   - The CM10 bit in the CM1 register is disabled against write. (Writing a “1” has no effect, nor is stop mode entered)
   - The watchdog timer does not stop in wait mode.
6. For $\overline{\text{NMI}}$ function, the PM24 bit must be set to “1”($\overline{\text{NMI}}$ function). Once this bit is set to “1”, it cannot be cleared to “0” by program.
7. $\overline{\text{SD}}$ input is valid regardless of the PM24 setting.

Figure 6.2 PM2 Register

Downloaded from Arrow.com.

The internal bus consists of CPU bus, memory bus, and peripheral bus. Bus Interface Unit (BIU) is used to interfere with CPU, ROM/RAM, and perpheral functions by controling CPU bus, memory bus, and peripheral bus. **Figure 6.3** shows the block diagram of the internal bus.

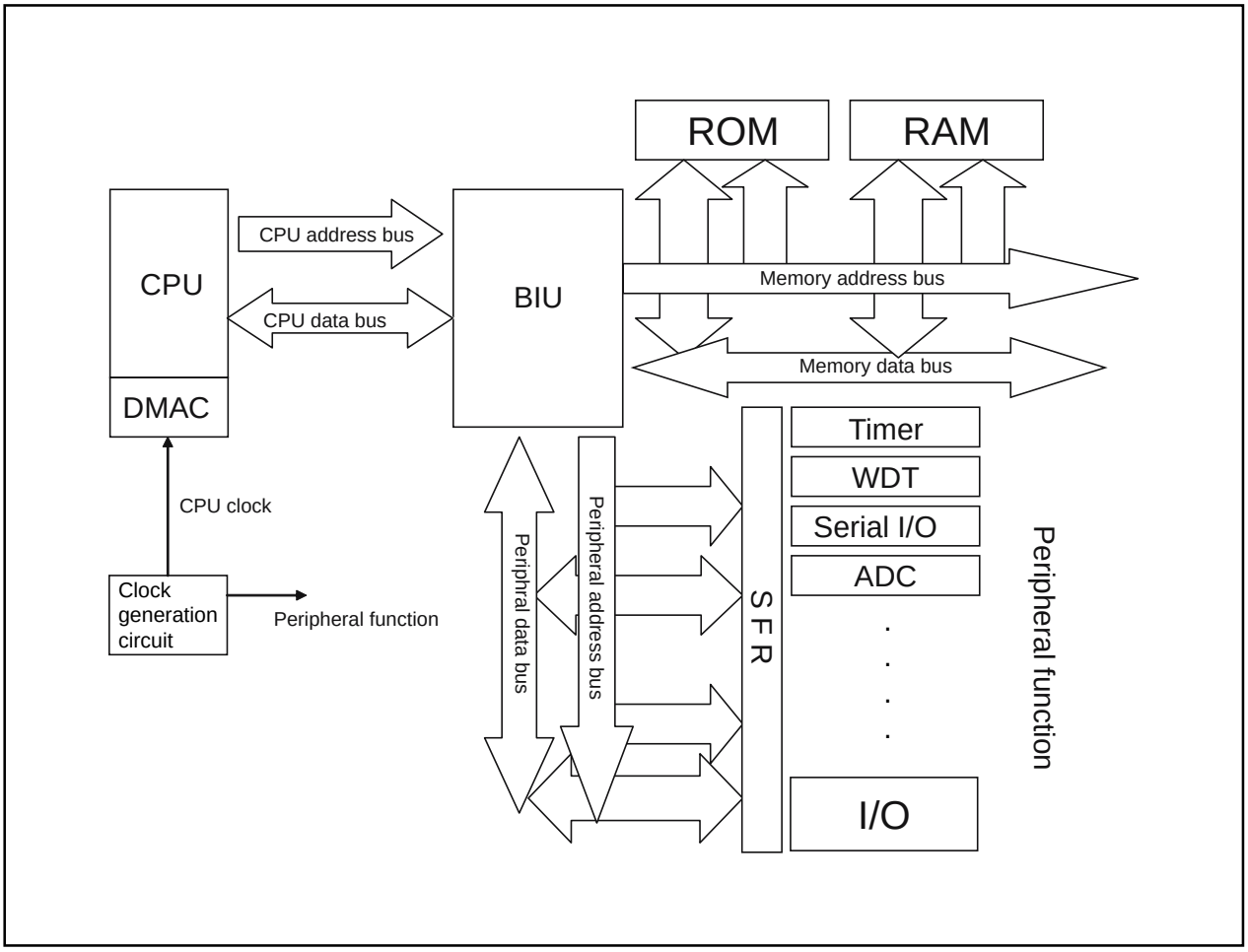


Figure 6.3 Bus Block Diagram

The number of bus cycle varies by the internal bus. **Table 6.1** lists the accessible area and bus cycle.

Table 6.1 Accessible Area and Bus Cycle

	Accessible Area	Bus Cycle
SFR	PM20 bit = 0 (2 waits)	3 CPU clock cycles
	PM20 bit = 1 (1 wait)	2 CPU clock cycles
ROM/RAM	PM17 bit = 0 (no wait)	1 CPU clock cycle
	PM17 bit = 1 (1 wait)	2 CPU clock cycles

Downloaded from Arrow.com.

7. Clock Generation Circuit

The clock generation circuit contains four oscillator circuits as follows:

(1) Main clock oscillation circuit
(2) Sub clock oscillation circuit
(3) On-chip oscillator (available at reset, oscillation stop detect function)
(4) PLL frequency synthesizer

Table 7.1 lists the clock generation circuit specifications. Figure 7.1 shows the clock generation circuit. Figures 7.2 to 7.6 show the clock-related registers.

Table 7.1. Clock Generation Circuit Specifications

Item	Main clock oscillation circuit	Sub clock oscillation circuit	On-chip oscillator	PLL frequency synthesizer
Use of clock	• CPU clock source • Peripheral function clock source	• CPU clock source • Timer A, B's clock source	• CPU clock source • Peripheral function clock source • CPU and peripheral function clock sources when the main clock stops oscillating	• CPU clock source • Peripheral function clock source
Clock frequency	0 to 20 MHz	32.768 kHz	• Selectable source frequency: $f_{1(ROC)}$, $f_{2(ROC)}$, $f_{3(ROC)}$ • Selectable divider: by 2, by 4, by 8	10 to 20 MHz (M16C/26A, M16C/26T) 10 to 24 MHz (M16C/26B)
Usable oscillator	• Ceramic oscillator • Crystal oscillator	• Crystal oscillator	———	———
Pins to connect oscillator	XIN, XOUT	XCIN, XCOUT	———	———
Oscillation stop, restart function	Available	Available	Available	Available
Oscillator status after reset	Oscillating (M16C/26A, M16C/26B) Stopped (M16C/26T)	Stopped	Oscillating (CPU clock source)	Stopped
Other	Externally derived clock can be input		———	———

Downloaded from Arrow.com.

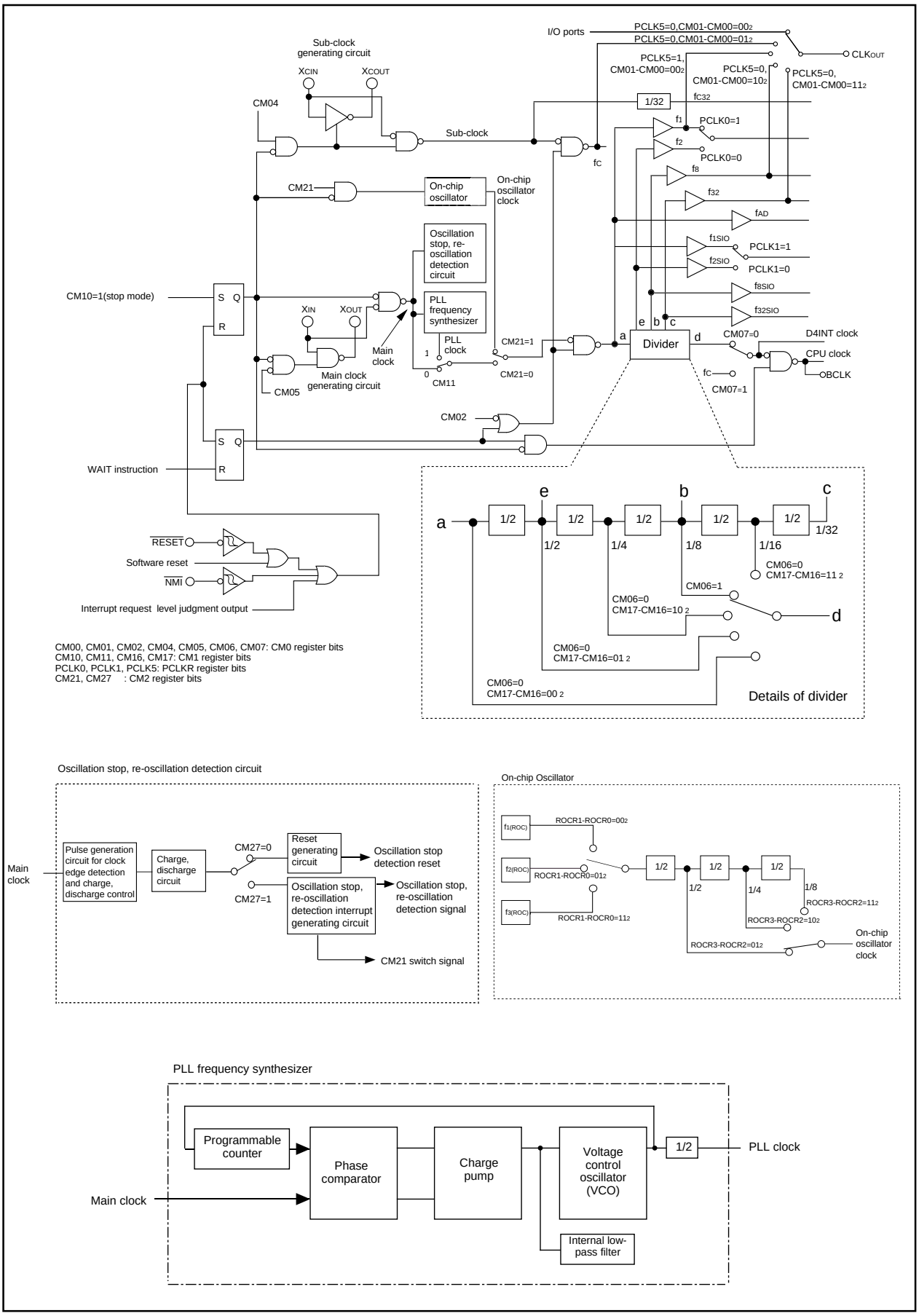


Figure 7.1. Clock Generation Circuit

Downloaded from Arrow.com.

System clock control register 0 (1)

b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	After reset
CM0	000616	010010002 (M16C/26A, M16C/26B) 011010002 (M16C/26T)

Bit symbol	Bit name	Function	RW
CM00	Clock output function select bit	Refer to **Table 7.5.3.1 Function of the CLKout pin**	RW
CM01			RW
CM02	WAIT peripheral function clock stop bit (10)	0 : Do not stop peripheral function clock in wait mode 1 : Stop peripheral function clock in wait mode (8)	RW
CM03	XCIN-XCOUT drive capacity select bit (2)	0 : LOW 1 : HIGH	RW
CM04	Port XC select bit (2)	0 : I/O port P86, P87 1 : XCIN-XCOUT generation function (9)	RW
CM05	Main clock stop bit (3, 10, 12, 13)	0 : On 1 : Off (4, 5)	RW
CM06	Main clock division select bit 0 (7, 13, 14)	0 : CM16 and CM17 valid 1 : Division by 8 mode	RW
CM07	System clock select bit (6, 10, 11, 12)	0 : Main clock, PLL clock, or on-chip oscillator clock 1 : Sub-clock	RW

NOTES:
1. Write to this register after setting the PRC0 bit in the PRCR register to "1" (write enable).
2. The CM03 bit is set to "1" (high) when the CM04 bit is set to "0" (I/O port) or the microcomputer goes to a stop mode.
3. This bit is provided to stop the main clock when the low power dissipation mode or on-chip oscillator low power dissipation mode is selected. This bit cannot be used for detection as to whether the main clock stopped or not. To stop the main clock, the following setting is required:
   (1) Set the CM07 bit to "1" (Sub-clock select) or the CM21 bit in the CM2 register to "1" (on-chip oscillator select) with the sub-clock stably oscillating.
   (2) Set the CM20 bit in CM2 register to "0" (Oscillation stop, re-oscillation detection function disabled).
   (3) Set the CM05 bit to "1" (Stop).
4. During external clock input, only the clock oscillation buffer is turned off and clock input is accepted.
5. When CM05 bit is set to "1", the XOUT pin goes "H". Furthermore, because the internal feedback resistor remains connected, the XIN pin is pulled "H" to the same level as XOUT via the feedback resistor.
6. After setting the CM04 bit to "1" (XCIN-XCOUT oscillator function), wait until the sub-clock oscillates stably before switching the CM07 bit from "0" to "1" (sub-clock).
7. When entering stop mode from high or middle speed mode, on-chip oscillator mode or on-chip oscillator low power mode, the CM06 bit is set to "1" (divide-by-8 mode).
8. The fC32 clock does not stop. During low speed or low power dissipation mode, do not set this bit to "1" (peripheral clock turned off when in wait mode).
9. To use a sub-clock, set this bit to "1". Also make sure ports P86 and P87 are directed for input, with no pull-ups.
10. When the PM21 bit of PM2 register is set to "1" (clock modification disable), writing to the CM02, CM05, and CM07 bits has no effect.
11. If the PM21 bit needs to be set to "1", set the CM07 bit to "0"(main clock) before setting it.
12. To use the main clock as the clock source for the CPU clock, follow the procedure below.
   (1) Set the CM05 bit to "0" (oscillate).
   (2) Wait until td(M-L) elapses or the main clock oscillation stabilizes, whichever is longer.
   (3) Set the CM11, CM21 and CM07 bits all to "0".
13. When the CM21 bit is set to "0" (on-chip oscillaor turned off) and the CM05 bit is set to "1" (main clock turned off), the CM06 bit is fixed to "1" (divide-by-8 mode) and the CM15 bit is fixed to "1" (drive capability High).
14. To return from on-chip oscillator mode to high-speed or middle-speed mode set the CM06 and CM15 bits both to "1".

Figure 7.2. CM0 Register

Downloaded from Arrow.com.

System clock control register 1 (1)

b7	b6	b5	b4	b3	b2	b1	b0
			0	0	0		

Symbol: CM1 — Address: 0007₁₆ — After reset: 00100000₂

Bit symbol	Bit name	Function	RW
CM10	All clock stop control bit (4, 6)	0 : Clock on 1 : All clocks off (stop mode)	RW
CM11	System clock select bit 1 (6, 7)	0 : Main clock 1 : PLL clock (5)	RW
(b4-b2)	Reserved bit	Must set to "0"	RW
CM15	XIN-XOUT drive capacity select bit (2)	0 : LOW 1 : HIGH	RW
CM16	Main clock division select bits (3)	b7 b6 0 0 : No division mode 0 1 : Division by 2 mode 1 0 : Division by 4 mode 1 1 : Division by 16 mode	RW
CM17			RW

NOTES:
1. Write to this register after setting the PRC0 bit in the PRCR register to "1" (write enable).
2. When entering stop mode from high or middle speed mode, or when the CM05 bit is set to "1" (main clock turned off) in low speed mode, the CM15 bit is set to "1" (drive capability high).
3. Effective when the CM06 bit is "0" (CM16 and CM17 bits enable).
4. If the CM10 bit is "1" (stop mode), XOUT goes "H" and the internal feedback resistor is disconnected. The XCIN and XCOUT pins are placed in the high-impedance state. When the CM11 bit is set to "1" (PLL clock), or the CM20 bit in the CM2 register is set to "1" (oscillation stop, re-oscillation detection function enabled), do not set the CM10 bit to "1".
5. After setting the PLC07 bit in the PLC0 register to "1" (PLL operation), wait until Tsu (PLL) elapses before setting the CM11 bit to "1" (PLL clock).
6. When the PM21 bit in the PM2 register is set to "1" (clock modification disable), writing to the CM10, CM011 bits has no effect. When the PM22 bit in the PM2 register is set to "1" (watchdog timer count source is on-chip oscillator clock), writing to the CM10 bit has no effect.
7. Effective when CM07 bit is "0" and CM21 bit is "0" .

Figure 7.3. CM1 Register

On-chip Oscillator Control Register (1)

b7	b6	b5	b4	b3	b2	b1	b0
X	0	0	0				

Symbol: ROCR — Address: 025C₁₆ — After Reset: X0000101₂

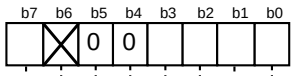
Bit Symbol	Bit Name	Function	RW
ROCR0	Frequency select bits	b1 b0 0 0: f1 (ROC) 0 1: f2 (ROC) 1 0: Do not set to this value 1 1: f3 (ROC)	RW
ROCR1			RW
ROCR2	Divider select bits	b3 b2 0 0: Do not set to this value 0 1: divide by 2 1 0: divide by 4 1 1: divide by 8	RW
ROCR3			RW
(b6-b4)	Reserved bit	Set to 0	RW
(b7)	Nothing is assigned. When write, set to 0. When read, its content is undefined		—

NOTE:
1. Write to this register after setting the PRC0 bit in the PRCR register to 1 (write enable).

Figure 7.4. ROCR Register

Downloaded from Arrow.com.

Oscillation stop detection register (1)



Symbol: CM2 | Address: 000C16 | After reset: 0X000010 2(11)

Bit symbol	Bit name	Function	RW
CM20	Oscillation stop, re-oscillation detection bit (7, 9, 10, 11)	0: Oscillation stop, re-oscillation detection function disabled 1: Oscillation stop, re-oscillation detection function enabled	RW
CM21	System clock select bit 2 (2, 3, 6, 8, 11, 12)	0: Main clock or PLL clock 1: On-chip oscillator clock (On-chip oscillator oscillating)	RW
CM22	Oscillation stop, re-oscillation detection flag (4)	0: Main clock stop or re-oscillation not detected 1: Main clock stop or re-oscillation detected	RW
CM23	XIN monitor flag (5)	0: Main clock oscillating 1: Main clock not oscillating	RO
— (b5-b4)	Reserved bit	Must set to “0”	RW
— (b6)	Nothing is assigned. When write, set to “0”. When read, its content is indeterminate.		—
CM27	Operation select bit (when an oscillation stop, re-oscillation is detected) (11)	0: Oscillation stop detection reset 1: Oscillation stop, re-oscillation detection interrupt	RW

NOTES:
1. Write to this register after setting the PRC0 bit in the PRCR register to “1” (write enable).
2. When the CM20 bit is set to “1” (oscillation stop, re-oscillation detection function enabled), the CM27 bit is set to “1”(oscillation stop, re-oscillation detection interrupt), and the CPU clock source is the main clock, the CM21 bit isautomatically set to “1” (on-chip oscillator clock) if the main clock stop is detected.
3. If the CM20 bit is set to “1” and the CM23 bit is set to “1” (main clock not oscillating), do not set the CM21 bit to “0”.
4. This flag is set to “1” when the main clock is detected to have stopped or when the main clock is detected to haverestarted oscillating. When this flag changes state from “0” to “1”, an oscillation stop, reoscillation detection interrupt isgenerated. Use this flag in an interrupt routine to discriminate the causes of interrupts between theoscillation stop,reoscillation detection interrupts and the watchdog timer interrupt. The flag is cleared to “0” by writing a “0” in aprogram. (Writing a “1” has no effect. Nor is it cleared to “0” by an oscillation stop or an oscillation restart detectiointerrupt request acknowledged.) If when the CM22 bit is set to "1" an oscillation stop or an oscillation restart is detected, no oscillation stop, reoscillation detection interrupts are generated.
5. Read the CM23 bit in an oscillation stop, re-oscillation detection interrupt handling routine to determine the main clockstatus.
6. Effective when the CM07 bit in the CM0 register is set to “0”.
7. When the PM21 bit in the PM2 register is “1” (clock modification disabled), writing to the CM20 bit has no effect.
8. When the CM20 bit is set to “1” (oscillation stop, re-oscillation detection function enabled), the CM27 bit is set to “1” (oscillation stop, re-oscillation detection interrupt), and the CM11 bit is set to “1” (the CPU clock source is PLL clock), the CM21 bit remains unchanged even when main clock stop is detected. If the CM22 bit is set to “0” under these conditions, oscillation stop, re-oscillation detection interrupt occur at main clock stop detection; it is, therefore, necessary to set the CM21 bit to “1” (on-chip oscillator clock) inside the interrupt routine.
9. Set the CM20 bit to “0” (disable) before entering stop mode. After exiting stop mode, set the CM20 bit back to "1” (enable).
10. Set the CM20 bit to “0” (disable) before setting the CM05 bit in the CM0 register.
11. The CM20, CM21 and CM27 bits do not change at oscillation stop detection reset.
12. When the CM21 bit is set to "0" (on-chip oscillator turned off) and the CM05 bit is set to "1" (main clock turned off), the CM06 bit is fixed to “1” (divide-by-8 mode) and the CM15 bit is fixed to “1” (drive capability High).

Figure 7.5. CM2 Register

Downloaded from Arrow.com.

Peripheral Clock Select Register (1)

b7	b6	b5	b4	b3	b2	b1	b0
0	0		0	0	0		

Symbol	Address	After Reset
PCLKR	025E16	00000011₂

Bit Symbol	Bit Name	Function	RW
PCLK0	Timers A, B clock select bit (Clock source for the timers A, B, the timer S, the dead timer, SI/O3, SI/O4 and multi-master I²C bus)	0: f2 1: f1	RW
PCLK1	SI/O clock select bit (Clock source for UART0 to UART2)	0: f2SIO 1: f1SIO	RW
(b4-b2)	Reserved bit	Set to 0	RW
PCLK5	Clock output function expansion select bit	Refer to **Table 7.5.3.1**	RW
(b7-b6)	Reserved bit	Set to 0	RW

NOTE:
1. Write to this register after setting the PRC0 bit in PRCR register to 1 (write enable).

Processeor Mode Register 2 (1)

b7	b6	b5	b4	b3	b2	b1	b0
X	X	X		0			

Symbol	Address	After Reset
PM2	001E16	XXX00000₂

Bit Symbol	Bit Name	Function	RW
PM20	Specifying wait when accessing SFR(2)	0: 2 waits 1: 1 wait	RW
PM21	System clock protective bit(3,4)	0: Clock is protected by PRCR register 1: Clock modification disabled	RW
PM22	WDT count source protective bit(3,5)	0: CPU clock is used for the watchdog timer count source 1: On-chip oscillator clock is used for the watchdog timer count source	RW
(b3)	Reserved bit	Set to 0	RW
PM24	P85/NMI configuration bit(6,7)	0: P85 function (NMI disabled) 1: NMI function	RW
(b7-b5)	Nothing is assigned. When write, set to 0. When read, thecontent is undefined		—

NOTES:
1. Write to this register after setting the PRC1 bit in the PRCR register to 1 (write enable).
2. The PM20 bit becomes effective when PLC07 bit in the PLC0 register is set to 1 (PLL on). Change the PM20 bit when the PLC07 bit is set to 0 (PLL off). Set the PM20 bit to 0 (2 waits) when PLL clock > 16MHz.
3. Once this bit is set to 1, it cannot be cleared to 0 by program.
4. Writting to the following bits has no effect when the PM21 bit is set to 1:
   CM02 bit in the CM0 register
   CM05 bit in the CM0 register (main clock is not halted)
   CM07 bit in the CM0 register (CPU clock source does not change)
   CM10 bit in the CM1 register (stop mode is not entered)
   CM11 bit in the CM1 register (CPU clock source does not change)
   CM20 bit in the CM2 register (oscillation stop, re-oscillation detection function settings do not change)
   All bits in the PLC0 register (PLL frequency synthesizer setting do not change)
   Do not execute WAIT instruction when the PM21 bit is set to 1.
5. Setting the PM22 bit to 1 results in the following conditions:
   - The on-chip oscillator continues oscillating even if the CM21 bit in the CM2 register is set to "0" (main clock or PLL clock) (system clock of count source selected by the CM21 bit is valid)
   - The on-chip oscillator starts oscillating, and the on-chip oscillator clock becomes the watchdog timer count source.
   - The CM10 bit in the CM1 register cannnot be written. (Writing 1 has no effect, stop mode is not entered.)
   - The watchdog timer does not stop in wait mode.
6. For NMI function, the PM24 bit must be set to 1(NMI function). Once this bit is set to 1, it cannot be set to 0 by program.
7. SD input is valid regardless of the PM24 setting.

Figure 7.6. PCLKR Register and PM2 Register

Downloaded from Arrow.com.

PLL control register 0 (1, 2)

b7	b6	b5	b4	b3	b2	b1	b0
	0	0	1	X			

Symbol	Address	After reset
PLC0	$001C_{16}$	$0001\ X010_2$

Bit symbol	Bit name	Function	RW
PLC00	PLL multiplying factor select bit (3)	b2 b1b0 0 0 0: Do not set 0 0 1: Multiply by 2 0 1 0: Multiply by 4 0 1 1: } Do not set 1 0 0: 1 0 1: 1 1 0: 1 1 1:	RW
PLC01			RW
PLC02			RW
(b3)	Nothing is assigned. When write, set to "0". When read, its content is indeterminate.		—
(b4)	Reserved bit	Must set to "1"	RW
(b6-b5)	Reserved bit	Must set to "0"	RW
PLC07	Operation enable bit (4)	0: PLL Off 1: PLL On	RW

NOTES:
1. Write to this register after setting the PRC0 bit in the PRCR register to "1" (write enable).
2. When the PM21 bit in the PM2 register is "1" (clock modification disable), writing to this register has no effect.
3. These three bits can only be modified when the PLC07 bit is set to "0" (PLL turned off). The value once written to this bit cannot be modified.
4. Before setting this bit to "1" , set the CM07 bit to "0" (main clock), set the CM17 and CM16 bits to "00_2" (main clock undivided mode), and set the CM06 bit to "0" (CM16 and CM17 bits enable).

Figure 7.7. PLC0 Register

Downloaded from Arrow.com.

The following describes the clocks generated by the clock generation circuit.

7.1 Main Clock

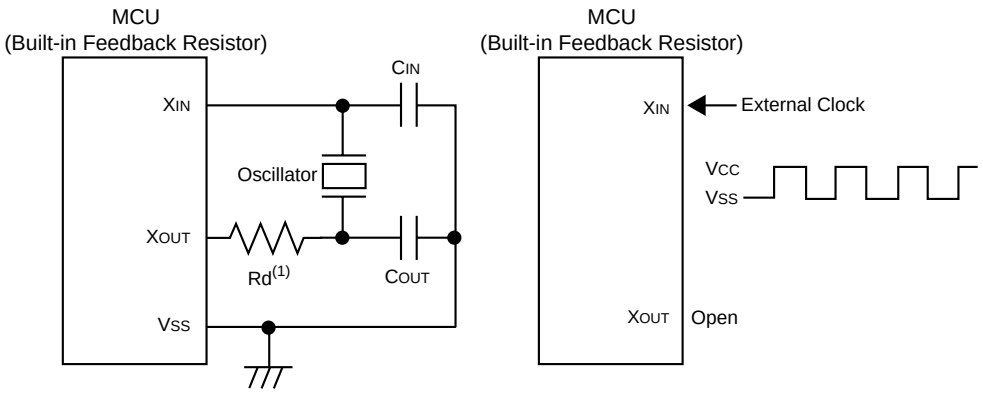
The main clock is generated by the main clock oscillation circuit. This clock is used as the clock source for the CPU and peripheral function clocks. The main clock oscillator circuit is configured by connecting a resonator between the XIN and XOUT pins. The main clock oscillator circuit contains a feedback resistor, which is disconnected from the oscillator circuit during stop mode in order to reduce the amount of power consumed in the chip. The main clock oscillator circuit may also be configured by feeding an externally generated clock to the XIN pin. Figure 7.1.1 shows the examples of main clock connection circuit.

The main clock after reset oscillates in the M16C/26A and M16C/26B, but stop in the M16C/26T.

The power consumption in the chip can be reduced by setting the CM05 bit in the CM0 register to “1” (main clock oscillator circuit turned off) after switching the clock source for the CPU clock to a sub clock or on-chip oscillator clock. In this case, XOUT goes “H”. Furthermore, because the internal feedback resistor remains on, XIN is pulled “H” to XOUT via the feedback resistor.

During stop mode, all clocks including the main clock are turned off. Refer to **7.6 power control**.

If the main clock is not used, it is recommended to connect the XIN pin to VCC to reduce power consumption during reset.



NOTE:
1. Insert a damping resistor if required. Resistance value varies depending on the oscillator setting. Use resistance value recommended by the oscillator manufacturer. If the oscillator manufacturer recommends that a feedback resistor be added to the chip externally, insert a feedback resistor between XIN and XOUT.
2. The external clock should not be stopped when it is connected to the XIN pin and the main clock is selected as the CPU clock.

Figure 7.1.1. Examples of Main Clock Connection Circuit

Downloaded from Arrow.com.

7.2 Sub Clock

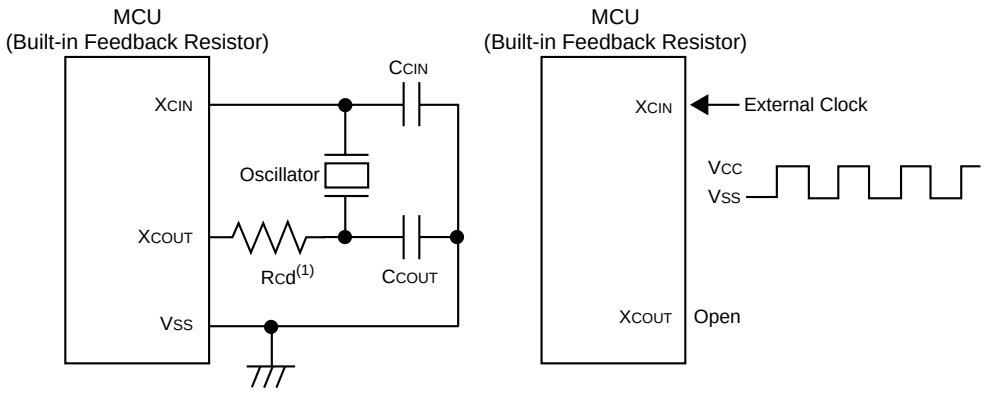
The sub clock is generated by the sub clock oscillation circuit. This clock is used as the clock source for the CPU clock, as well as the timer A and timer B count sources.

The sub clock oscillator circuit is configured by connecting a crystal resonator between the XCIN and XCOUT pins. The sub clock oscillator circuit contains a feedback resistor, which is disconnected from the oscillator circuit during stop mode in order to reduce the amount of power consumed in the chip. The sub clock oscillator circuit may also be configured by feeding an externally generated clock to the XCIN pin. Figure 7.2.1 shows the examples of sub clock connection circuit.

After reset, the sub clock is turned off. At this time, the feedback resistor is disconnected from the oscillator circuit.

To use the sub clock for the CPU clock, set the CM07 bit in the CM0 register to “1 ” (sub clock) after the sub clock becomes oscillating stably.

During stop mode, all clocks including the sub clock are turned off. Refer to **7.6 Power Control**.



NOTE:
1. Place a damping resistor if required. Resistance values vary depending on the oscillator setting. Use values recommended by each oscillator manufacturer. Place a feedback resistor between XCIN and XCOUT if the oscillator manufacturer recommends placing the resistor externally.

Figure 7.2.1. Examples of Sub Clock Connection Circuit

Downloaded from Arrow.com.

7.3 On-chip Oscillator Clock

This clock is supplied by a on-chip oscillator. This clock is used as the clock source for the CPU and peripheral function clocks. In addition, if the PM22 bit in the PM2 register is “1” (on-chip oscillator clock for the watchdog timer count source), this clock is used as the count source for the watchdog timer (Refer to **10.1 Count source protective mode**).

The on-chip oscillator clock after reset oscillates. The on-chip oscillator clock $f_{2(ROC)}$ divided by 16 is used for the CPU clock. It can also be turned off by setting the CM21 bit in the CM2 register to “0” (main clock or PLL clock). If the main clock stops oscillating when the CM20 bit in the CM2 register is “1” (oscillation stop, re-oscillation detection function enabled) and the CM27 bit is “1” (oscillation stop, re-oscillation detection interrupt), the on-chip oscillator automatically starts operating, supplying the necessary clock for the micro-computer.

7.4 PLL Clock

The PLL clock is generated from the main clock by a PLL frequency synthesizer. This clock is used as the clock source for the CPU and peripheral function clocks. After reset, the PLL clock is turned off. The PLL frequency synthesizer is activated by setting the PLC07 bit to “1” (PLL operation). When the PLL clock is used as the clock source for the CPU clock, wait t_{su}(PLL) for the PLL clock to be stable, and then set the CM11 bit in the CM1 register to “1”.

Before entering wait mode or stop mode, be sure to set the CM11 bit to “0” (CPU clock source is the main clock). Furthermore, before entering stop mode, be sure to set the PLC07 bit in the PLC0 register to “0” (PLL stops). Figure 7.4.1 shows the procedure for using the PLL clock as the clock source for the CPU.

The PLL clock frequency is determined by the equation below.

PLL clock frequency=f(X_{IN}) X (multiplying factor set by the PLC02 to PLC00 bits in the PLC0 register)

(However, 10 MHz ≤ PLL clock frequency ≤ 20 MHz in M16C/26A and M16C/26T, 10 MHz ≤ PLL clock frequency ≤ 24 MHz in M16C/26B)

The PLC02 to PLC00 bits can be set only once after reset. Table 7.4.1 shows the example for setting PLL clock frequencies.

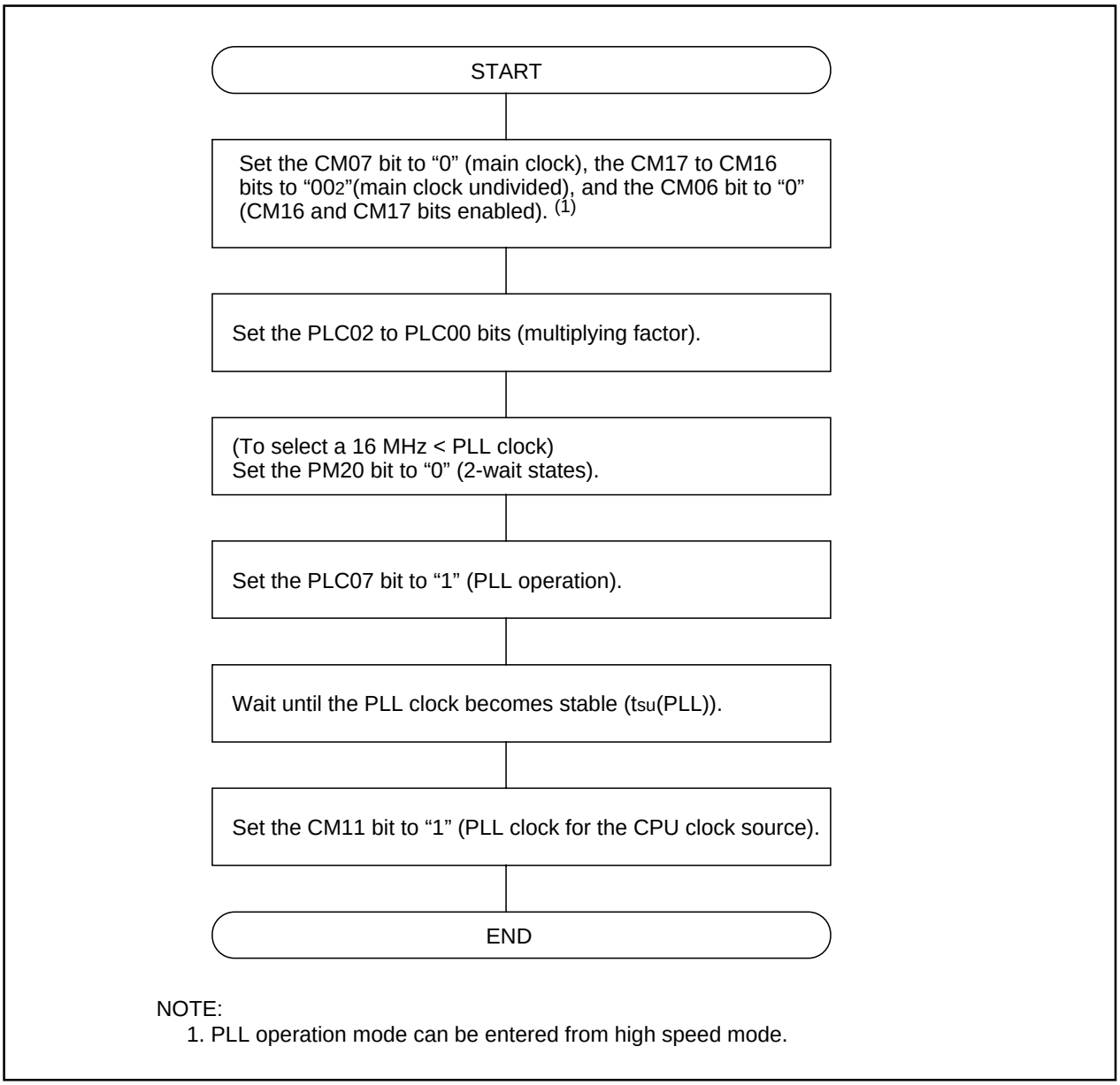
Table 7.4.1. Example for Setting PLL Clock Frequencies

X_{IN} (MHz)	PLC02	PLC01	PLC00	Multiplying factor	PLL clock (MHz)(1)
10	0	0	1	2	20
5	0	1	0	4	

NOTE:

1. 10 MHz ≤ PLL clock frequency ≤ 20 MHz in M16C/26A and M16C/26T, 10 MHz ≤ PLL clock frequency ≤ 24 MHz in M16C/26B)

Downloaded from Arrow.com.



NOTE:
1. PLL operation mode can be entered from high speed mode.

Figure 7.4.1. Procedure to Use PLL Clock as CPU Clock Source

Downloaded from Arrow.com.

7.5 CPU Clock and Peripheral Function Clock

The CPU clock is used to operate the CPU and peripheral function clocks are used to operate the peripheral functions.

7.5.1 CPU Clock

This is the operating clock for the CPU and watchdog timer.

The clock source for the CPU clock can be chosen to be the main clock, sub clock, on-chip oscillator clock or the PLL clock.

If the main clock or on-chip oscillator clock is selected as the clock source for the CPU clock, the selected clock source can be divided by 1 (undivided), 2, 4, 8 or 16 to produce the CPU clock. Use the CM06 bit in CM0 register and the CM17 to CM16 bits in CM1 register to select the divide-by-n value.

When the PLL clock is selected as the clock source for the CPU clock, the CM06 bit should be set to "0" and the CM17 and CM16 bits to "00_2" (undivided).

After reset, the on-chip oscillator clock divided by 16 provides the CPU clock.

Note that when entering stop mode from high or middle speed mode, on-chip oscillator mode or on-chip oscillator low power dissipation mode, or when the CM05 bit in the CM0 register is set to "1" (main clock turned off) in low-speed mode, the CM06 bit in the CM0 register is set to "1" (divide-by-8 mode).

7.5.2 Peripheral Function Clock (f_1, f_2, f_8, f_{32}, f_{1SIO}, f_{2SIO}, f_{8SIO}, f_{32SIO}, f_{AD}, f_{C32})

These are operating clocks for the peripheral functions.

Of these, fi (i = 1, 2, 8, 32) and f_{iSIO} are derived from the main clock, PLL clock or on-chip oscillator clock divided by i. The clock fi is used for Timer A and Timer B while f_{iSIO} is used for UART0 to UART2. Additionally, the f1 and f2 clocks are also used for dead time timer.

The f_{AD} clock is produced from the main clock, PLL clock or on-chip oscillator clock, and is used for the A/D converter.

When the WAIT instruction is executed after setting the CM02 bit in the CM0 register to "1" (peripheral function clock turned off during wait mode), or when the microcomputer is in low power dissipation mode, the fi, f_{iSIO} and f_{AD} clocks are turned off.

The f_{C32} clock is produced from the sub clock, and is used for timers A and B. This clock can only be used when the sub clock is on.

7.5.3 ClockOutput Function

The f_1, f_8, f_{32} or f_C clock can be output from the CLK_{OUT} pin. Use the PCLK5 bit in the PCLKR register and CM01 to CM00 bits in the CM0 register to select. Table 7.5.3.1 shows the function of the CLK_{OUT} pin.

Table 7.5.3.1 The function of the CLK_{OUT} pin

PCLK5	CM01	CM00	The function of the CLK_{OUT} pin
0	0	0	I/O port $P9_0$
0	0	1	f_C
0	1	0	f_8
0	1	1	f_{32}
1	0	0	f_1
1	0	1	Do not set
1	1	0	Do not set
1	1	1	Do not set

Downloaded from Arrow.com.

7.6 Power Control

There are three power control modes. For convenience' sake, all modes other than wait and stop modes are referred to as normal operation mode here.

7.6.1 Normal Operation Mode

Normal operation mode is further classified into seven modes.

In normal operation mode, because the CPU clock and the peripheral function clocks both are on, the CPU and the peripheral functions are operating. Power control is exercised by controlling the CPU clock frequency. The higher the CPU clock frequency, the greater the processing capability. The lower the CPU clock frequency, the smaller the power consumption in the chip. If the unnecessary oscillator circuits are turned off, the power consumption is further reduced.

Before the clock sources for the CPU clock can be switched over, the new clock source to which switched must be oscillating stably. If the new clock source is the main clock, sub clock or PLL clock, allow a sufficient wait time in a program until it becomes oscillating stably.

Note that operation modes cannot be changed directly from low power dissipation mode to on-chip oscillator mode or on-chip oscillator low power dissipation mode. Nor can operation modes be changed directly from on-chip oscillator mode or on-chip oscillator low power dissipation mode to low power dissipation mode.

When the CPU clock source is changed from the on-chip oscillator to the main clock, change the operation mode to the medium speed mode (divided by 8 mode) after the clock was divided by 8 (the CM06 bit in the CM0 register was set to "1") in the on-chip oscillator mode.

7.6.1.1 High-speed Mode

The main clock divided by 1 provides the CPU clock. If the sub clock is on, f_{C32} can be used as the count source for timers A and B.

7.6.1.2 PLL Operation Mode

The main clock multiplied by 2 or 4 provides the PLL clock, and this PLL clock serves as the CPU clock. If the sub clock is on, f_{C32} can be used as the count source for timers A and B. PLL operation mode can be entered from high speed mode. If PLL operation mode is to be changed to wait or stop mode, first go to high speed mode before changing.

7.6.1.3 Medium-speed Mode

The main clock divided by 2, 4, 8 or 16 provides the CPU clock. If the sub clock is on, f_{C32} can be used as the count source for timers A and B.

7.6.1.4 Low-speed Mode

The sub clock provides the CPU clock. The main clock is used as the clock source for the peripheral function clock when the CM21 bit is set to "0" (on-chip oscillator turned off), and the on-chip oscillator clock is used when the CM21 bit is set to "1" (on-chip oscillator oscillating).

The f_{C32} clock can be used as the count source for timers A and B.

7.6.1.5 Low Power Dissipation Mode

In this mode, the main clock is turned off after being placed in low speed mode. The sub clock provides the CPU clock. The f_{C32} clock can be used as the count source for timers A and B. Peripheral function clock can use only f_{C32}.

Simultaneously when this mode is selected, the CM06 bit in the CM0 register becomes "1" (divided by 8 mode). In the low power dissipation mode, do not change the CM06 bit. Consequently, the medium speed (divided by 8) mode is to be selected when the main clock is operated next.

Downloaded from Arrow.com.

7.6.1.6 On-chip Oscillator Mode

The selected on-chip oscillator clock divided by 1 (undivided), 2, 4, 8 or 16 provides the CPU clock. The on-chip oscillator clock is also the clock source for the peripheral function clocks. If the sub clock is on, f_{C32} can be used as the count source for timers A and B. The on-chip oscillator frequency can be selected ROCR3 to ROCR0 bits in ROCR register. When the operation mode is returned to the high and medium speed modes, set the CM06 bit to "1" (divided by 8 mode).

7.6.1.7 On-chip Oscillator Low Power Dissipation Mode

The main clock is turned off after being placed in on-chip oscillator mode. The CPU clock can be selected as in the on-chip oscillator mode. The on-chip oscillator clock is the clock source for the peripheral function clocks. If the sub clock is on, f_{C32} can be used as the count source for timers A and B.

Table 7.6.1.1. Setting Clock Related Bit and Modes

Modes		CM2 register	CM1 register		CM0 register			
		CM21	CM11	CM17, CM16	CM07	CM06	CM05	CM04
PLL operation mode		0	1	00_2	0	0	0	—
High-speed mode		0	0	00_2	0	0	0	—
Medium-speed mode	divided by 2	0	0	01_2	0	0	0	—
	divided by 4	0	0	10_2	0	0	0	—
	divided by 8	0	0	—	0	1	0	—
	divided by 16	0	0	11_2	0	0	0	—
Low-speed mode		—	—	—	1	—	0	1
Low power dissipation mode		—	—	—	1	1(1)	1(1)	1
On-chip oscillator mode (3)	divided by 1	1	—	00_2	0	0	0	—
	divided by 2	1	—	01_2	0	0	0	—
	divided by 4	1	—	10_2	0	0	0	—
	divided by 8	1	—	—	0	1	0	—
	divided by 16	1	—	11_2	0	0	0	—
On-chip oscillator low power dissipation mode		1	—	(2)	0	(2)	1	—

NOTES:
1. When the CM05 bit is set to "1" (main clock turned off) in low-speed mode, the mode goes to low power dissipation mode and CM06 bit is set to "1" (divided by 8 mode) simultaneously.
2. The divide-by-n value can be selected the same way as in on-chip oscillator mode.
3. On-chip oscillator frequency can be any of those described in the section **7.6.1.6 On-chip Oscillator Mode**.

7.6.2 Wait Mode

In wait mode, the CPU clock is turned off, so are the CPU (because operated by the CPU clock) and the watchdog timer. However, if the PM22 bit in the PM2 register is "1" (on-chip oscillator clock for the watchdog timer count source), the watchdog timer remains active. Because the main clock, sub clock, on-chip oscillator clock and PLL clock all are on, the peripheral functions using these clocks keep operating.

7.6.2.1 Peripheral Function Clock Stop Function

If the CM02 bit is "1" (peripheral function clocks turned off during wait mode), the f_1, f_2, f_8, f_{32}, f_{1SIO}, f_{8SIO}, f_{32SIO} and f_{AD} clocks are turned off when in wait mode, with the power consumption reduced that much. However, f_{C32} remains on.

7.6.2.2 Entering Wait Mode

The microcomputer is placed into wait mode by executing the WAIT instruction.

When the CM11 bit is set to "1" (CPU clock source is the PLL clock), be sure to clear the CM11 bit to "0" (CPU clock source is the main clock) before going to wait mode. The power consumption of the chip can be reduced by clearing the PLC07 bit to "0" (PLL stops).

Downloaded from Arrow.com.

7.6.2.3 Pin Status During Wait Mode

Table 7.6.2.3.1 lists pin status during wait mode.

Table 7.6.2.3.1 Pin Status in Wait Mode

Pin		Status
I/O ports		Retains status before wait mode
CLKOUT	When fC selected	Does not stop
	When f1, f8, f32 selected	Does not stop when the CM02 bit is set to “0”.
		Retains status before wait mode when the CM02 bit is set to “1”.

7.6.2.4 Exiting Wait Mode

The microcomputer is moved out of wait mode by a hardware reset, $\overline{\text{NMI}}$ interrupt or peripheral function interrupt.

If the microcomputer is to be moved out of exit wait mode by a hardware reset or $\overline{\text{NMI}}$ interrupt, set the peripheral function interrupt priority ILVL2 to ILVL0 bits to “000_2” (interrupts disabled) before executing the WAIT instruction.

The peripheral function interrupts are affected by the CM02 bit. If the CM02 bit is set to “0” (peripheral function clocks not turned off during wait mode), all peripheral function interrupts can be used to exit wait mode. If the CM02 bit is set to “1” (peripheral function clocks turned off during wait mode), the peripheral functions using the peripheral function clocks stop operating, so that only the peripheral functions clocked by external signals can be used to exit wait mode.

Table 7.6.2.4.1 lists the interrupts to exit wait mode.

Table 7.6.2.4.1. Interrupts to Exit Wait Mode

Interrupt	CM02=0	CM02=1
$\overline{\text{NMI}}$ interrupt	Can be used	Can be used
Serial I/O interrupt	Can be used when operating with internal or external clock	Can be used when operating with external clock
key input interrupt	Can be used	Can be used
A/D conversion interrupt	Can be used in one-shot mode or single sweep mode	— (Do not use)
Timer A interrupt Timer B interrupt	Can be used in all modes	Can be used in event counter mode or when the count source is fC32
$\overline{\text{INT}}$ interrupt	Can be used	Can be used

If the microcomputer is to be moved out of wait mode by a peripheral function interrupt, set up the following before executing the WAIT instruction.

1. In the ILVL2 to ILVL0 bits in the interrupt control register, set the interrupt priority level of the peripheral function interrupt to be used to exit wait mode.
   Also, for all of the peripheral function interrupts not used to exit wait mode, set the ILVL2 to ILVL0 bits to “000_2” (interrupt disable).
2. Set the I flag to “1”.
3. Enable the peripheral function whose interrupt is to be used to exit wait mode.
   In this case, when an interrupt request is generated and the CPU clock is thereby turned on, an interrupt routine is executed.

The CPU clock turned on when exiting wait mode by a peripheral function interrupt is the same CPU clock that was on when the WAIT instruction was executed.

Downloaded from Arrow.com.

7.6.3 Stop Mode

In stop mode, all oscillator circuits are turned off, so are the CPU clock and the peripheral function clocks. Therefore, the CPU and the peripheral functions clocked by these clocks stop operating. The least amount of power is consumed in this mode. If the voltage applied to Vcc pin is VRAM or more, the internal RAM is retained. When applying 2.7 or less voltage to Vcc pin, make sure Vcc≥VRAM.

However, the peripheral functions clocked by external signals keep operating. The following interrupts can be used to exit stop mode.

- $\overline{\text{NMI}}$ interrupt
- Key interrupt
- $\overline{\text{INT}}$ interrupt
- Timer A, Timer B interrupt (when counting external pulses in event counter mode)
- Serial I/O interrupt (when external clock is selected)
- Voltage down detection interrupt

(refer to **5.5.1 Voltage Down Detection Interrupt** for an operating condition)

7.6.3.1 Entering Stop Mode

The microcomputer is placed into stop mode by setting the CM10 bit in the CM1 register to “1” (all clocks turned off). At the same time, the CM06 bit in the CM0 register is set to “1” (divide-by-8 mode) and the CM15 bit in the CM10 register is set to “1” (main clock oscillator circuit drive capability high). Before entering stop mode, set the CM20 bit to “0” (oscillation stop, re-oscillation detection function disable).

Also, if the CM11 bit is “1” (PLL clock for the CPU clock source), set the CM11 bit to “0” (main clock for the CPU clock source) and the PLC07 bit to “0” (PLL turned off) before entering stop mode.

7.6.3.2 Pin Status during Stop Mode

The I/O pins retain their status held just prior to entering stop mode.

7.6.3.3 Exiting Stop Mode

The microcomputer is moved out of stop mode by a hardware reset, $\overline{\text{NMI}}$ interrupt or peripheral function interrupt.

If the microcomputer is to be moved out of stop mode by a hardware reset or $\overline{\text{NMI}}$ interrupt, set the peripheral function interrupt priority ILVL2 to ILVL0 bits to “000_2” (interrupts disable) before setting the CM10 bit to “1”.

If the microcomputer is to be moved out of stop mode by a peripheral function interrupt, set up the following before setting the CM10 bit to “1”.

1. In the ILVL2 to ILVL0 bits in the interrupt control register, set the interrupt priority level of the peripheral function interrupt to be used to exit stop mode.
   Also, for all of the peripheral function interrupts not used to exit stop mode, set the ILVL2 to ILVL0 bits to “000_2”.
2. Set the I flag to “1”.
3. Enable the peripheral function whose interrupt is to be used to exit stop mode.
   In this case, when an interrupt request is generated and the CPU clock is thereby turned on, an interrupt service routine is executed.

Which CPU clock will be used after exiting stop mode by a peripheral function or $\overline{\text{NMI}}$ interrupt is determined by the CPU clock that was on when the microcomputer was placed into stop mode as follows:

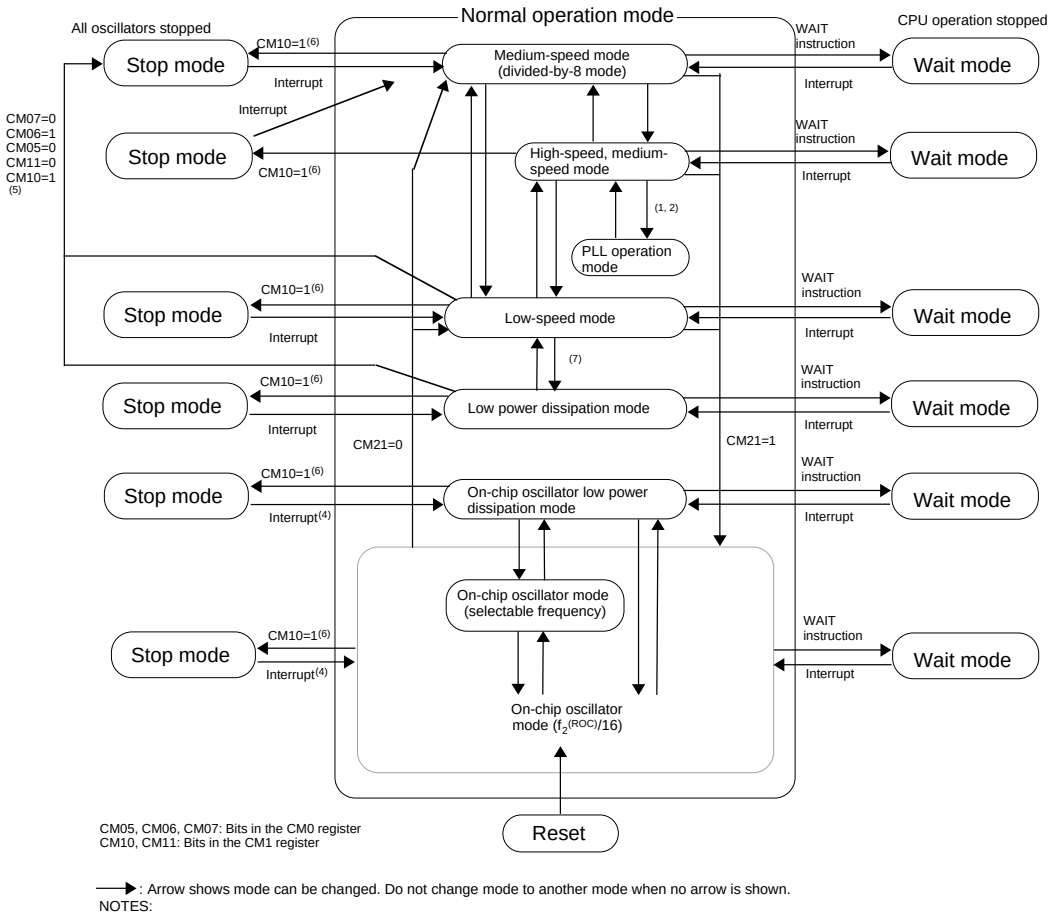
If the CPU clock before entering stop mode was derived from the sub clock : sub clock

If the CPU clock before entering stop mode was derived from the main clock : main clock divide-by-8

If the CPU clock before entering stop mode was derived from the on-chip oscillator clock: on-chip oscillator clock divide-by-8

Downloaded from Arrow.com.

Figure 7.6.1 shows the state transition from normal operation mode to stop mode and wait mode. Figure 7.6.1.1 shows the state transition in normal operation mode.
Table 7.6.1 shows a state transition matrix describing allowed transition and setting. The vertical line shows current state and horizontal line shows state after transition.

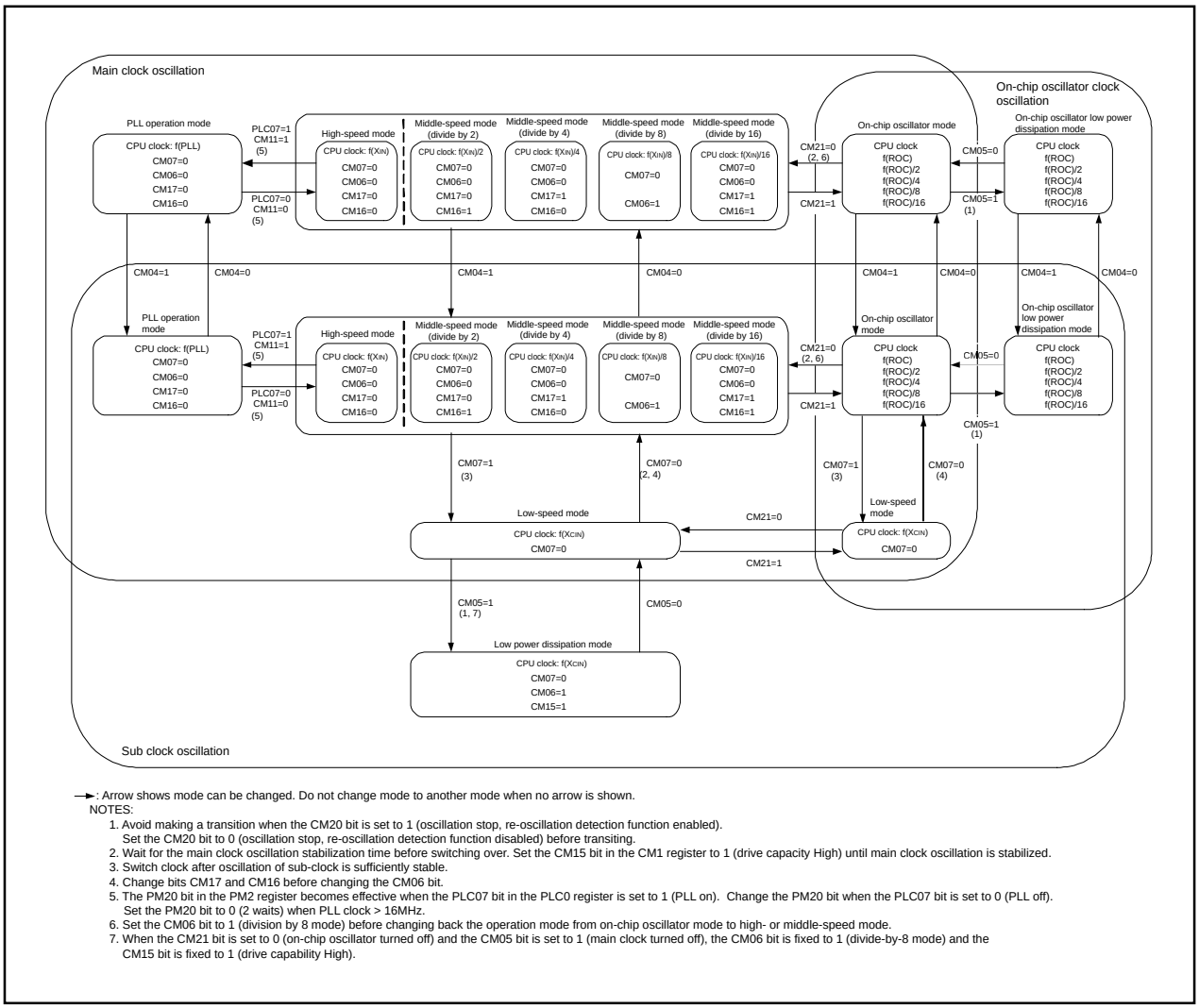


→: Arrow shows mode can be changed. Do not change mode to another mode when no arrow is shown.

NOTES:
1. Do not go directly from PLL operation mode to wait or stop mode.
2. PLL operation mode can be entered from high speed mode. Similarly, PLL operation mode can be changed back to high speed mode.
3. When the PM21 bit is set to 0 (system clock protective function unused).
4. The on-chip oscillator clock divided by 8 provides the CPU clock.
5. Write to the CM0 register and CM1 register simultaneously by accessing in word units while CM21 bit is set to 0 (on-chip oscillator turned off). When the clock generated externally is input to the XCIN pin, transit to stop mode with this process.
6. Before entering stop mode, be sure to clear the CM20 bit in the CM2 register to 0 (oscillation stop and oscillation restart detection function disabled).
7. The CM06 bit is set to 1 (divide-by-8).

Figure 7.6.1. State Transition to Stop Mode and Wait Mode

Downloaded from Arrow.com.



→: Arrow shows mode can be changed. Do not change mode to another mode when no arrow is shown.

NOTES:

1. Avoid making a transition when the CM20 bit is set to 1 (oscillation stop, re-oscillation detection function enabled). Set the CM20 bit to 0 (oscillation stop, re-oscillation detection function disabled) before transiting.
2. Wait for the main clock oscillation stabilization time before switching over. Set the CM15 bit in the CM1 register to 1 (drive capacity High) until main clock oscillation is stabilized.
3. Switch clock after oscillation of sub-clock is sufficiently stable.
4. Change bits CM17 and CM16 before changing the CM06 bit.
5. The PM20 bit in the PM2 register becomes effective when the PLC07 bit in the PLC0 register is set to 1 (PLL on). Change the PM20 bit when the PLC07 bit is set to 0 (PLL off). Set the PM20 bit to 0 (2 waits) when PLL clock > 16MHz.
6. Set the CM06 bit to 1 (division by 8 mode) before changing back the operation mode from on-chip oscillator mode to high- or middle-speed mode.
7. When the CM21 bit is set to 0 (on-chip oscillator turned off) and the CM05 bit is set to 1 (main clock turned off), the CM06 bit is fixed to 1 (divide-by-8 mode) and the CM15 bit is fixed to 1 (drive capability High).

Figure 7.6.1.1. State Transition in Normal Mode

Downloaded from Arrow.com.

Table 7.6.1. Allowed Transition and Setting

		State after transition							
		High-speed mode, middle-speed mode	Low-speed mode[2]	Low power dissipation mode	PLL operation mode[2]	On-chip oscillator mode	On-chip oscillator low power dissipation mode	Stop mode	Wait mode
Current state	High-speed mode, middle-speed mode	8	(9)[7]	--	(13)[3]	(15)	--	(16)[1]	(17)
	Low-speed mode[2]	(8)		(11)[1, 6]	--	(8)	--	(16)[1]	(17)
	Low power dissipation mode	--	(10)		--	--	--	(16)[1]	(17)
	PLL operation mode[2]	(12)[3]	--	--		--	--	--	--
	On-chip oscillator mode	(14)[4]	(9)[7]	--	--	8	(11)[1]	(16)[1]	(17)
	On-chip oscillator low power dissipation mode	--	--	--	--	(10)	8	(16)[1]	(17)
	Stop mode	(18)[5]	(18)	(18)	--	(18)[5]	(18)[5]		--
	Wait mode	(18)	(18)	(18)	--	(18)	(18)	--	

NOTES:

--: Cannot transit

1. Avoid making a transition when the CM20 bit is set to 1 (oscillation stop, re-oscillation detection function enabled). Set the CM20 bit to 0 (oscillation stop, re-oscillation detection function disabled) before transiting.
2. On-chip oscillator clock oscillates and stops in low-speed mode. In this mode, the on-chip oscillator can be used as peripheral function clock. Sub clock oscillates and stops in PLL operation mode. In this mode, sub clock can be used as a clock for the timers A and B.
3. PLL operation mode can only be entered from and changed to high-speed mode.
4. Set the CM06 bit to 1 (division by 8 mode) before transiting from on-chip oscillator mode to high- or middle-speed mode.
5. When exiting stop mode, the CM06 bit is set to 1 (division by 8 mode).
6. If the CM05 bit is set to 1 (main clock stop), then the CM06 bit is set to 1 (division by 8 mode).
7. A transition can be made only when sub clock is oscillating.
8. State transitions within the same mode (divide-by-n values changed or subclock oscillation turned on or off) are shown in the table below.

		Sub clock oscillating					Sub clock turned off				
		No division	Divided by 2	Divided by 4	Divided by 8	Divided by 16	No division	Divided by 2	Divided by 4	Divided by 8	Divided by 16
Sub clock oscillating	No division		(4)	(5)	(7)	(6)	(1)	--	--	--	--
	Divided by 2	(3)		(5)	(7)	(6)	--	(1)	--	--	--
	Divided by 4	(3)	(4)		(7)	(6)	--	--	(1)	--	--
	Divided by 8	(3)	(4)	(5)		(6)	--	--	--	(1)	--
	Divided by 16	(3)	(4)	(5)	(7)		--	--	--	--	(1)
Sub clock turned off	No division	(2)	--	--	--	--		(4)	(5)	(7)	(6)
	Divided by 2	--	(2)	--	--	--	(3)		(5)	(7)	(6)
	Divided by 4	--	--	(2)	--	--	(3)	(4)		(7)	(6)
	Divided by 8	--	--	--	(2)	--	(3)	(4)	(5)		(6)
	Divided by 16	--	--	--	--	(2)	(3)	(4)	(5)	(7)	

9. ( ) : setting method. Refer to following table.

--: Cannot transit

	Setting	Operation
(1)	CM04 = 0	Sub clock turned off
(2)	CM04 = 1	Sub clock oscillating
(3)	CM06 = 0, CM17 = 0 , CM16 = 0	CPU clock no division mode
(4)	CM06 = 0, CM17 = 0 , CM16 = 1	CPU clock division by 2 mode
(5)	CM06 = 0, CM17 = 1 , CM16 = 0	CPU clock division by 4 mode
(6)	CM06 = 0, CM17 = 1 , CM16 = 1	CPU clock division by 16 mode
(7)	CM06 = 1	CPU clock division by 8 mode
(8)	CM07 = 0	Main clock, PLL clock, or on-chip oscillator clock selected
(9)	CM07 = 1	Sub clock selected
(10)	CM05 = 0	Main clock oscillating
(11)	CM05 = 1	Main clock turned off
(12)	PLC07 = 0, CM11 = 0	Main clock selected
(13)	PLC07 = 1, CM11 = 1	PLL clock selected
(14)	CM21 = 0	Main clock or PLL clock selected
(15)	CM21 = 1	On-chip oscillator clock selected
(16)	CM10 = 1	Transition to stop mode
(17)	wait instruction	Transition to wait mode
(18)	Hardware interrupt	Exit stop mode or wait mode

CM04, CM05, CM06, CM07 : Bits in the CM0 register
CM10, CM11, CM16, CM17 : Bits in the CM1 register
CM20, CM21 : Bits in the CM2 register
PLC07 : Bit in the PLC0 register

Downloaded from Arrow.com.

7.7 System Clock Protective Function

When the main clock is selected for the CPU clock source, this function protects the clock from modifications in order to prevent the CPU clock from becoming halted by run-away.

If the PM21 bit in the PM2 register is set to “1” (clock modification disabled), the following bits are protected against writes:

- CM02, CM05, and CM07 bits in CM0 register
- CM10, CM11 bits in CM1 register
- CM20 bit in CM2 register
- All bits in PLC0 register

Before the system clock protective function can be used, the following register settings must be made while the CM05 bit in the CM0 register is “0” (main clock oscillating) and CM07 bit is “0” (main clock selected for the CPU clock source):

(1) Set the PRC1 bit in the PRCR register to “1” (enable writes to PM2 register).

(2) Set the PM21 bit in the PM2 register to “1” (disable clock modification).

(3) Set the PRC1 bit in the PRCR register to “0” (disable writes to PM2 register).

Do not execute the WAIT instruction when the PM21 bit is set to “1”.

7.8 Oscillation Stop and Re-oscillation Detect Function

The oscillation stop and re-oscillation detect function allows the detection of main clock oscillation stop and reoscillation. At oscillation stop or re-oscillation detection, reset or oscillation stop, re-oscillation detection interrupt are generated. Depending on the CM27 bit in the CM2 register. The oscillation stop detection function can be enabled and disabled by the CM20 bit in the CM2 register. Table 7.8.1 lists a specification overview of the oscillation stop and re-oscillation detect function.

Table 7.8.1. Specification Overview of Oscillation Stop and Re-oscillation Detect Function

Item	Specification
Oscillation stop detectable clock and frequency bandwidth	$f(X_{IN}) \geq 2$ MHz
Enabling condition for oscillation stop, re-oscillation detection function	Set the CM20 bit to “1”(enable)
Operation at oscillation stop, re-oscillation detection	•Reset occurs (when the CM27 bit is set to "0") •Oscillation stop, re-oscillation detection interrupt occurs(when the CM27 bit is set to "1")

Downloaded from Arrow.com.

7.8.1 Operation When the CM27 bit is set to "0" (Oscillation Stop Detection Reset)

When main clock stop is detected when the CM20 bit is “1” (oscillation stop, re-oscillation detection function enabled), the microcomputer is initialized, coming to a halt (oscillation stop reset; refer to **4. SFR**, **5. Reset**).

This status is reset with hardware reset 1 or hardware reset 2. Also, even when re-oscillation is detected, the microcomputer can be initialized and stopped; it is, however, necessary to avoid such usage. (During main clock stop, do not set the CM20 bit to “1” and the CM27 bit to “0”.)

7.8.2 Operation When the CM27 bit is set to "1" (Oscillation Stop and Re-oscillation Detect Interrupt)

When the main clock corresponds to the CPU clock source and the CM20 bit is “1” (oscillation stop and re-oscillation detect function enabled), the system is placed in the following state if the main clock comes to a halt:

- Oscillation stop and re-oscillation detect interrupt request occurs.
- The on-chip oscillator starts oscillation, and the on-chip oscillator clock becomes the CPU clock and clock source for peripheral functions in place of the main clock.
- CM21 bit is set to "1" (on-chip oscillator clock for CPU clock source)
- CM22 bit is set to "1" (main clock stop detected)
- CM23 bit is set to "1" (main clock stopped)

When the PLL clock corresponds to the CPU clock source and the CM20 bit is “1”, the system is placed in the following state if the main clock comes to a halt: Since the CM21 bit remains unchanged, set it to “1” (on-chip oscillator clock) inside the interrupt routine.

- Oscillation stop and re-oscillation detect interrupt request occurs.
- CM22 bit is set to "1" (main clock stop detected)
- CM23 bit is set to "1" (main clock stopped)
- CM21 bit remains unchanged

When the CM20 bit is “1”, the system is placed in the following state if the main clock re-oscillates from the stop condition:

- Oscillation stop and re-oscillation detect interrupt request occurs.
- CM22 bit is set to "1" (main clock re-oscillation detected)
- CM23 bit is set to "0" (main clock oscillation)
- CM21 bit remains unchanged

Downloaded from Arrow.com.

7.8.3 How to Use Oscillation Stop and Re-oscillation Detect Function

- The oscillation stop and re-oscillation detect interrupt shares the vector with the watchdog timer interrupt. If the oscillation stop, re-oscillation detection and watchdog timer interrupts both are used, read the CM22 bit in an interrupt routine to determine which interrupt source is requesting the interrupt.
- Where the main clock re-oscillated after oscillation stop, return the main clock to the CPU clock and peripheral function clock source in the program. Figure 7.8.3.1 shows the procedure for switching the clock source from the on-chip oscillator to the main clock.
- Simultaneously with oscillation stop, re-oscillation detection interrupt occurrence, the CM22 bit becomes “1”. When the CM22 bit is set at “1”, oscillation stop, re-oscillation detection interrupt are disabled. By setting the CM22 bit to “0” in the program, oscillation stop, re-oscillation detection interrupt are enabled.
- If the main clock stops during low speed mode where the CM20 bit is “1”, an oscillation stop, re-oscillation detection interrupt request is generated. At the same time, the on-chip oscillator starts oscillating. In this case, although the CPU clock is derived from the sub clock as it was before the interrupt occurred, the peripheral function clocks now are derived from the on-chip oscillator clock.
- To enter wait mode while using the oscillation stop, re-oscillation detection function, set the CM02 bit to “0” (peripheral function clocks not turned off during wait mode).
- Since the oscillation stop, re-oscillation detection function is provided in preparation for main clock stop due to external factors, set the CM20 bit to “0” (Oscillation stop, re-oscillation detection function disabled) where the main clock is stopped or oscillated in the program, that is where the stop mode is selected or the CM05 bit is altered.
- This function cannot be used if the main clock frequency is 2 MHz or less. In that case, set the CM20 bit to “0”.

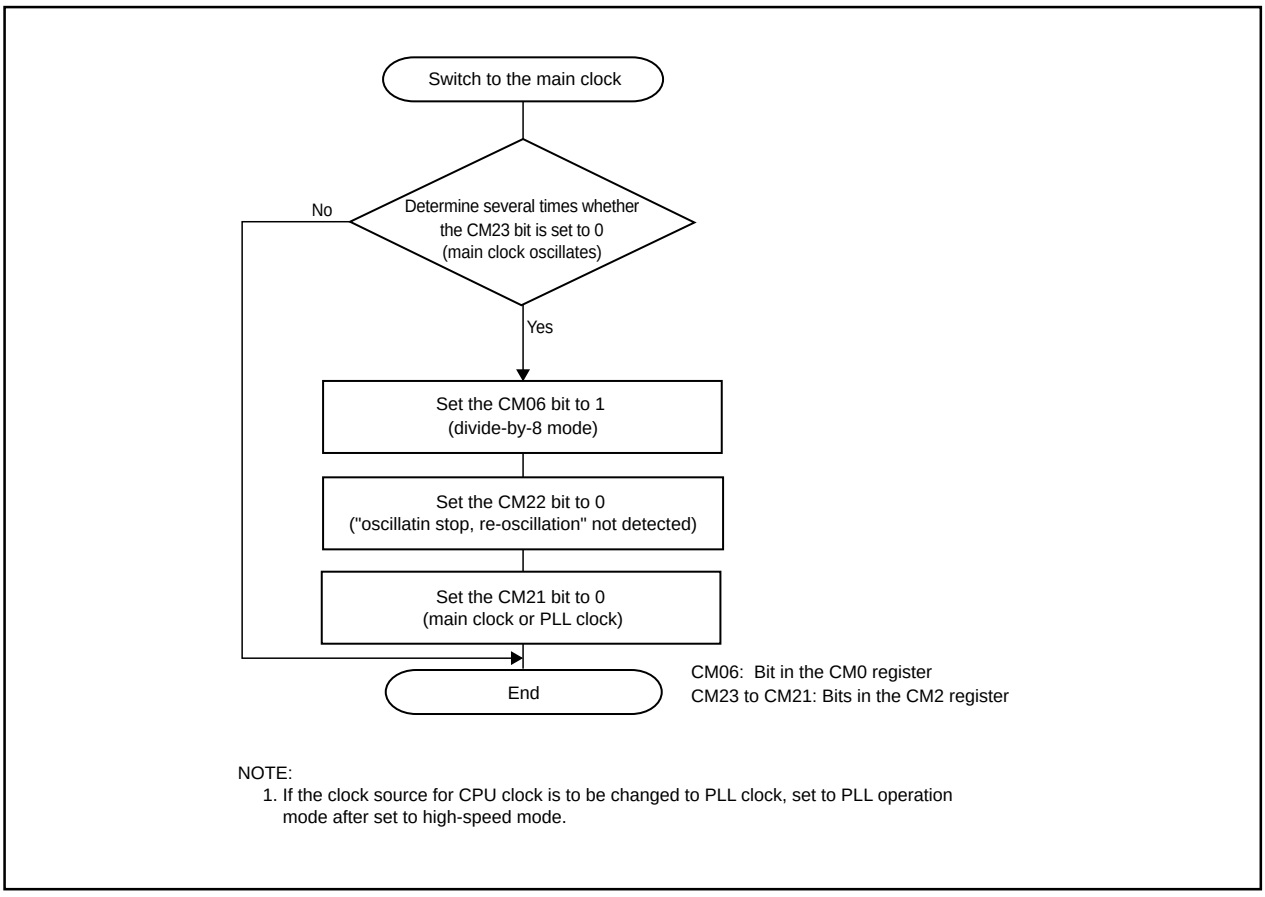


Figure 7.8.3.1. Procedure to Switch Clock Source From On-chip Oscillator to Main Clock

Downloaded from Arrow.com.

8. Protection

Note

The PRC3 bit in the PRCR register is not available in M16C/26T.

In the event that a program runs out of control, this function protects the important registers so that they will not be rewritten easily. Figure 8.1 shows the PRCR register. The following lists the registers protected by the PRCR register.

- Registers protected by PRC0 bit: CM0, CM1, CM2, PLC0, ROCR and PCLKR registers
- Registers protected by PRC1 bit: PM0, PM1, PM2, TB2SC, INVC0 and INVC1 registers
- Registers protected by PRC2 bit: PD9, PACR and NDDR registers
- Registers protected by PRC3 bit: VCR2 and D4INT registers

Set the PRC2 bit to “1” (write enabled) and then write to SFR area, and the PRC2 bit will be cleared to “0” (write protected). The registers protected by the PRC2 bit should be changed in the next instruction after setting the PRC2 bit to “1”. Make sure no interrupts or DMA transfers will occur between the instruction in which the PRC2 bit is set to “1” and the next instruction. The PRC0, PRC1 and PRC3 bits are not automatically cleared to “0” by writing to any address. They can only be cleared in a program.

Protect register

b7	b6	b5	b4	b3	b2	b1	b0
×	×	0	0				

Symbol: PRCR — Address: 000A16 — After reset: XX000000₂

Bit symbol	Bit name	Function	RW
PRC0	Protect bit 0	Enable write to CM0, CM1, CM2, ROCR, PLC0 and PCLKR registers 0 : Write protected 1 : Write enabled	RW
PRC1	Protect bit 1	Enable write to PM0, PM1, PM2, TB2SC, INVC0 and INVC1 registers 0 : Write protected 1 : Write enabled	RW
PRC2	Protect bit 2	Enable write to PD9, PACR and NDDR registers 0 : Write protected 1 : Write enabled	RW
PRC3	Protect bit 3	Enable write to VCR2 and D4INT registers 0 : Write protected 1 : Write enabled	RW
—— (b5-b4)	Reserved bit	Must set to "0"	RW
—— (b7-b6)	Nothing is assigned. When write, set to "0". When read, its content is indeterminate.		—

NOTE:
1. The PRC2 bit is set to "0" if data is written to the SFR area after the PRC2 bit is set to "1". The PRC0, PRC1 and PRC3 bits are not automatically set to "0". Set them to "0" by program.

Figure 8.1. PRCR Register

Downloaded from Arrow.com.

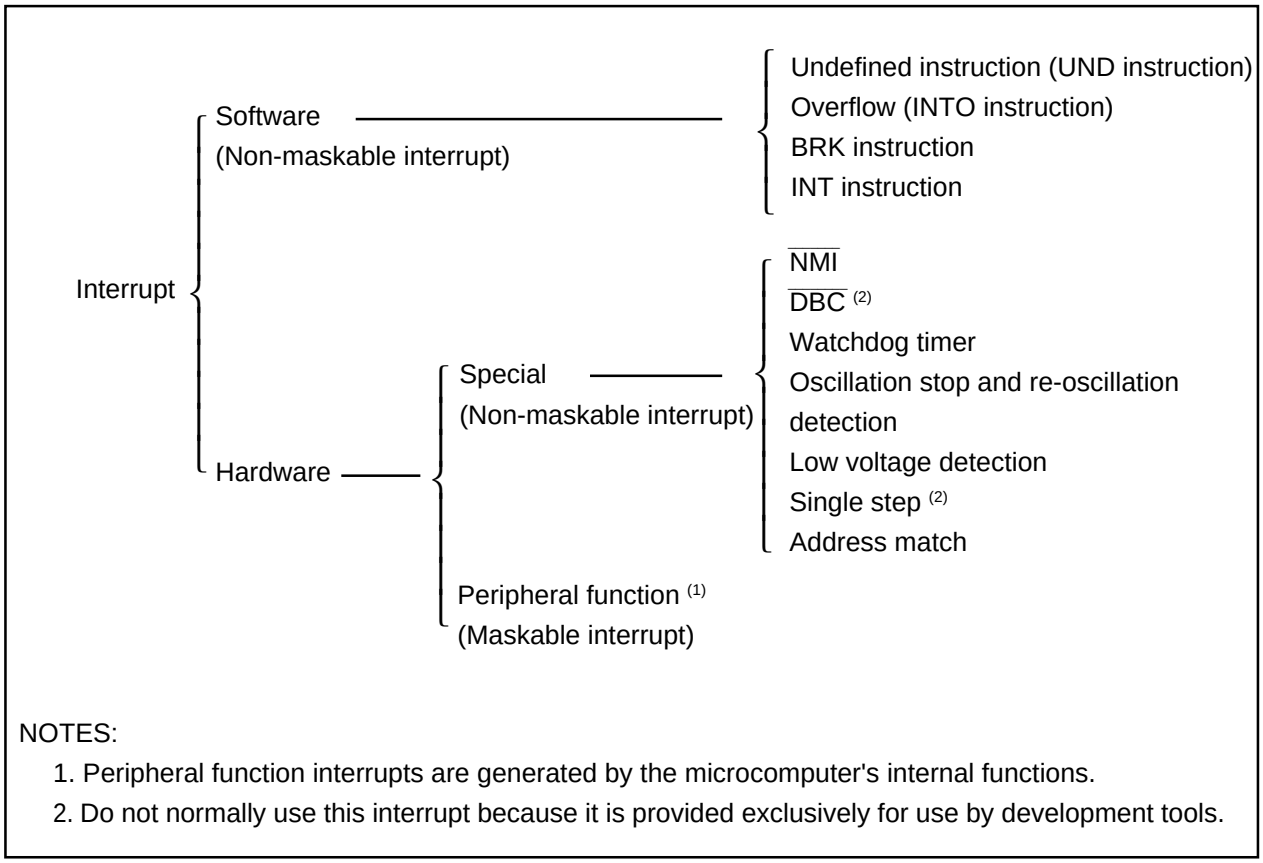
9. Interrupt

Note

The 42-pin package does not use UART0 transmission interrupt and UART0 reception interrupt of peripheral function.
M16C/26T does not use voltage down detection interrupt.

9.1 Type of Interrupts

Figure 9.1.1 shows types of interrupts.



NOTES:
1. Peripheral function interrupts are generated by the microcomputer's internal functions.
2. Do not normally use this interrupt because it is provided exclusively for use by development tools.

Figure 9.1.1. Interrupts

- Maskable Interrupt: An interrupt which can be enabled (disabled) by the interrupt enable flag (I flag) or whose interrupt priority **can be changed** by priority level.
- Non-maskable Interrupt: An interrupt which cannot be enabled (disabled) by the interrupt enable flag (I flag) or whose interrupt priority **cannot be changed** by priority level.

Downloaded from Arrow.com.

9.1.1 Software Interrupts

A software interrupt occurs when executing certain instructions. Software interrupts are non-maskable interrupts.

9.1.1.1 Undefined Instruction Interrupt

An undefined instruction interrupt occurs when executing the UND instruction.

9.1.1.2 Overflow Interrupt

An overflow interrupt occurs when executing the INTO instruction with the O flag set to “1” (the operation resulted in an overflow). The following are instructions whose O flag changes by arithmetic: ABS, ADC, ADCF, ADD, CMP, DIV, DIVU, DIVX, NEG, RMPA, SBB, SHA, SUB

9.1.1.3 BRK Interrupt

A BRK interrupt occurs when executing the BRK instruction.

9.1.1.4 INT Instruction Interrupt

An INT instruction interrupt occurs when executing the INT instruction. Software interrupt Nos. 0 to 63 can be specified for the INT instruction. Because software interrupt Nos. 4, 8 to 31 are assigned to peripheral function interrupts, the same interrupt routine as for peripheral function interrupts can be executed by executing the INT instruction.

In software interrupt Nos. 0 to 31, the U flag is saved to the stack during instruction execution and is cleared to “0” (ISP selected) before executing an interrupt sequence. The U flag is restored from the stack when returning from the interrupt routine. In software interrupt Nos. 32 to 63, the U flag does not change state during instruction execution, and the SP then selected is used.

Downloaded from Arrow.com.

9.1.2 Hardware Interrupts

Hardware interrupts are classified into two types — special interrupts and peripheral function interrupts.

9.1.2.1 Special Interrupts

Special interrupts are non-maskable interrupts.

9.1.2.1.1 $\overline{\text{NMI}}$ Interrupt

An $\overline{\text{NMI}}$ interrupt is generated when input on the $\overline{\text{NMI}}$ pin changes state from high to low. For details about the $\overline{\text{NMI}}$ interrupt, refer to the section **9.7 $\overline{\text{NMI}}$ Interrupt**.

9.1.2.1.2 $\overline{\text{DBC}}$ Interrupt

This interrupt is exclusively for debugger, do not use in any other circumstances.

9.1.2.1.3 Watchdog Timer Interrupt

Generated by the watchdog timer. Once a watchdog timer interrupt is generated, be sure to initialize the watchdog timer. For details about the watchdog timer, refer to the section **10. Watchdog Timer**.

9.1.2.1.4 Oscillation Stop and Re-oscillation Detection Interrupt

Generated by the oscillation stop and re-oscillation detection function. For details about the oscillation stop and re-oscillation detection function, refer to the section **7. Clock Generating Circuit**.

9.1.2.1.5 Voltage Down Detection Interrupt

Generated by the voltage detection circuit. For details about the voltage detection circuit, refer to the section **5.5 Voltage Detection Circuit**.

9.1.2.1.6 Single-step Interrupt

Do not normally use this interrupt because it is provided exclusively for use by development tools.

9.1.2.1.7 Address Match Interrupt

An address match interrupt is generated immediately before executing the instruction at the address indicated by the RMAD0 or RMAD1 register, if the corresponding enable bit (the AIER0 or AIER1 bit in the AIER register) is set to “1”. For details about the address match interrupt, refer to the section **9.9 Address Match Interrupt**.

9.1.2.2 Peripheral Function Interrupts

Peripheral function interrupts are maskable interrupts and generated by the microcomputer's internal functions. The interrupt sources for peripheral function interrupts are listed in **Table 9.2.2.1 Relocatable Vector Tables**. For details about the peripheral functions, refer to the description of each peripheral function in this manual.

Downloaded from Arrow.com.

9.2 Interrupts and Interrupt Vector

One interrupt vector consists of 4 bytes. Set the start address of each interrupt routine in the respective interrupt vectors. When an interrupt request is accepted, the CPU branches to the address set in the corresponding interrupt vector. Figure 9.2.1 shows the interrupt vector.

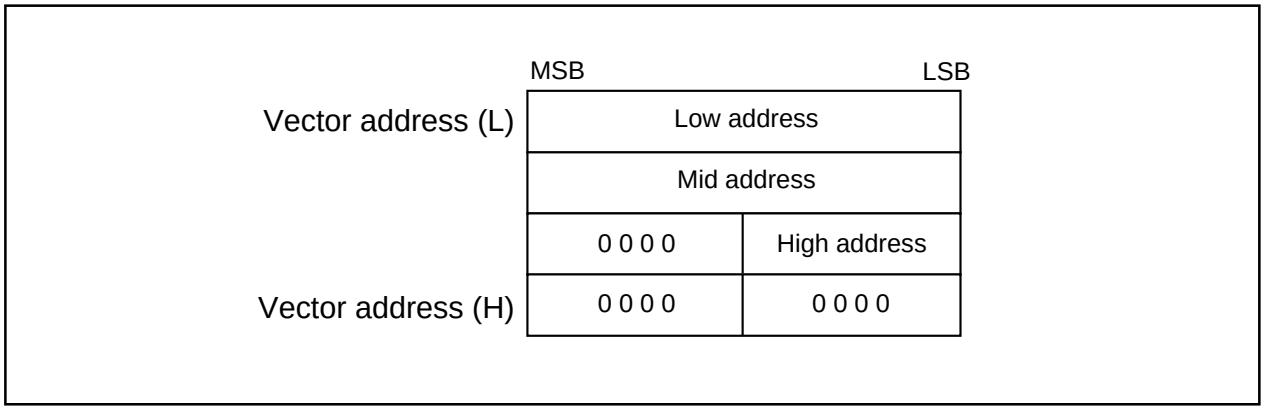


Figure 9.2.1. Interrupt Vector

9.2.1 Fixed Vector Tables

The fixed vector tables are allocated to the addresses from FFFDC16 to FFFFF16. Table 9.2.1.1 lists the fixed vector tables. In the flash memory version of microcomputer, the vector addresses (H) of fixed vectors are used by the ID code check function. For details, refer to the section **17.3 Flash Memory Rewrite Disabling Function**.

Table 9.2.1.1. Fixed Vector Tables

Interrupt source	Vector table addresses Address (L) to address (H)	Remarks	Reference
Undefined instruction	FFFDC16 to FFFDF16	Interrupt on UND instruction	M16C/60, M16C/20 serise software maual
Overflow	FFFE016 to FFFE316	Interrupt on INTO instruction	
BRK instruction	FFFE416 to FFFE716	If the contents of address FFFE716 is FF16, program execution starts from the address shown by the vector in the relocatable vector table.	
Address match	FFFE816 to FFFEB16		Address match interrupt
Single step (1)	FFFEC16 to FFFEF16		
Watchdog timer Oscillation stop and re-oscillation detection Voltage down detection	FFFF016 to FFFF316		Watchdog timer Clock generating circuit Voltage detection circuit
$\overline{\text{DBC}}$ (1)	FFFF416 to FFFF716		
$\overline{\text{NMI}}$	FFFF816 to FFFFB16		$\overline{\text{NMI}}$ interrupt
Reset (2)	FFFFC16 to FFFFF16		Reset

NOTES:

1. Do not normally use this interrupt because it is provided exclusively for use by development tools.
2. The b3 to b0 in address 0FFFFF16 are reserve bits. Set these bits to “11112”.

Downloaded from Arrow.com.

9.2.2 Relocatable Vector Tables

The 256 bytes beginning with the start address set in the INTB register comprise a reloacatable vector table area. Table 9.2.2.1 lists the relocatable vector tables. Setting an even address in the INTB register results in the interrupt sequence being executed faster than in the case of odd addresses.

Table 9.2.2.1. Relocatable Vector Tables

Interrupt source	Vector address (1) Address (L) to address (H)	Software interrupt number	Reference
BRK instruction (4)	+0 to +3 (0000 16 to 0003 16)	0	M16C/60, M16C/20 series software manual
——— (Reserved)		1 to 3	
$\overline{\text{INT3}}$	+16 to +19 (0010 16 to 0013 16)	4	$\overline{\text{INT}}$ interrupt
——— (Reserved)		5 to 7	
$\overline{\text{INT5}}$ (2)	+32 to +35 (0020 16 to 0023 16)	8	$\overline{\text{INT}}$ interrupt
$\overline{\text{INT4}}$ (2)	+36 to +39 (0024 16 to 0027 16)	9	
UART 2 bus collision detection (5)	+40 to +43 (0028 16 to 002B 16)	10	Serial I/O
DMA0	+44 to +47 (002C 16 to 002F 16)	11	DMAC
DMA1	+48 to +51 (0030 16 to 0033 16)	12	
Key input interrupt	+52 to +55 (0034 16 to 0037 16)	13	Key input interrupt
A/D	+56 to +59 (0038 16 to 003B 16)	14	A/D convertor
UART2 transmit, NACK2 (3)	+60 to +63 (003C 16 to 003F 16)	15	Serial I/O
UART2 receive, ACK2 (3)	+64 to +67 (0040 16 to 0043 16)	16	
UART0 transmit	+68 to +71 (0044 16 to 0047 16)	17	
UART0 receive	+72 to +75 (0048 16 to 004B 16)	18	
UART1 transmit	+76 to +79 (004C 16 to 004F 16)	19	
UART1 receive	+80 to +83 (0050 16 to 0053 16)	20	
Timer A0	+84 to +87 (0054 16 to 0057 16)	21	Timer
Timer A1	+88 to +91 (0058 16 to 005B 16)	22	
Timer A2	+92 to +95 (005C 16 to 005F 16)	23	
Timer A3	+96 to +99 (0060 16 to 0063 16)	24	
Timer A4	+100 to +103 (0064 16 to 0067 16)	25	
Timer B0	+104 to +107 (0068 16 to 006B 16)	26	
Timer B1	+108 to +111 (006C 16 to 006F 16)	27	
Timer B2	+112 to +115 (0070 16 to 0073 16)	28	
$\overline{\text{INT0}}$	+116 to +119 (0074 16 to 0077 16)	29	$\overline{\text{INT}}$ interrupt
$\overline{\text{INT1}}$	+120 to +123 (0078 16 to 007B 16)	30	
$\overline{\text{INT2}}$	+124 to +127 (007C 16 to 007F 16)	31	
Software interrupt (4)	+128 to +131 (0080 16 to 0083 16) to +252 to +255 (00FC 16 to 00FF 16)	32 to 63	M16C/60, M16C/20 series software manual

NOTES:
1. Address relative to address in INTB.
2. Set the IFSR6 and IFSR7 bits in the IFSR register.
3. During I^2C bus mode, NACK and ACK interrupts comprise the interrupt source.
4. These interrupts cannot be disabled using the I flag.
5. Bus collision detection:
   During IEBus mode, this bus collision detection constitutes the cause of an interrupt.
   During I^2C bus mode, however, a start condition or a stop condition detection constitutes the cause of an interrupt.

Downloaded from Arrow.com.

9.3 Interrupt Control

The following describes how to enable/disable the maskable interrupts, and how to set the priority in which order they are accepted. What is explained here does not apply to nonmaskable interrupts.

Use the I flag in the FLG register, IPL, and the ILVL2 to ILVL0 bits in the each interrupt control register to enable/disable the maskable interrupts. Whether an interrupt is requested is indicated by the IR bit in each interrupt control register.

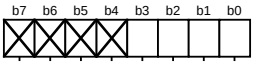
Figure 9.3.1 shows the interrupt control registers.

Figure 9.3.2 shows the IFSR, IFSR2A registers.

Downloaded from Arrow.com.

Interrupt control register (2)

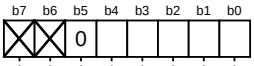
Symbol	Address	After reset
BCNIC	004A16	XXXXX0002
DM0IC, DM1IC	004B16, 004C16	XXXXX0002
KUPIC	004D16	XXXXX0002
ADIC	004E16	XXXXX0002
S0TIC to S2TIC	005116, 005316, 004F16	XXXXX0002
S0RIC to S2RIC	005216, 005416, 005016	XXXXX0002
TA0IC to TA4IC	005516 to 005916	XXXXX0002
TB0IC to TB2IC	005A16 to 005C16	XXXXX0002



Bit symbol	Bit name	Function	RW
ILVL0	Interrupt priority level select bit	b2 b1 b0 0 0 0 : Level 0 (interrupt disabled) 0 0 1 : Level 1 0 1 0 : Level 2 0 1 1 : Level 3 1 0 0 : Level 4 1 0 1 : Level 5 1 1 0 : Level 6 1 1 1 : Level 7	RW
ILVL1			RW
ILVL2			RW
IR	Interrupt request bit	0 : Interrupt not requested 1 : Interrupt requested	RW (1)
—— (b7-b4)	No functions are assigned. When writing to these bits, write “0”. The values in these bits when read are indeterminate.		——

NOTES:
1. This bit can only be reset by writing “0” (Do not write “1”).
2. To rewrite the interrupt control registers, do so at a point that does not generate the interrupt request for that register. For details, see **19.5 Interrupts**.

Symbol	Address	After reset
INT3IC	004416	XX00X0002
INT5IC	004816	XX00X0002
INT4IC	004916	XX00X0002
INT0IC to INT2IC	005D16 to 005F16	XX00X0002



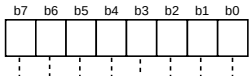
Bit symbol	Bit name	Function	RW
ILVL0	Interrupt priority level select bit	b2 b1 b0 0 0 0 : Level 0 (interrupt disabled) 0 0 1 : Level 1 0 1 0 : Level 2 0 1 1 : Level 3 1 0 0 : Level 4 1 0 1 : Level 5 1 1 0 : Level 6 1 1 1 : Level 7	RW
ILVL1			RW
ILVL2			RW
IR	Interrupt request bit	0: Interrupt not requested 1: Interrupt requested	RW (1)
POL	Polarity select bit	0 : Selects falling edge (3) 1 : Selects rising edge	RW
—— (b5)	Reserved bit	Must always be set to “0”	RW
—— (b7-b6)	No functions are assigned. When writing to these bits, write “0”. The values in these bits when read are indeterminate.		RW

NOTES:
1. This bit can only be reset by writing “0” (Do not write “1”).
2. To rewrite the interrupt control register, do so at a point that does not generate the interrupt request for that register. For details, see **19.5 Interrupts**.
3. If the IFSRi bit (i = 0 to 5) in the IFSR register is “1” (both edges), set the POL bit in the INTiIC register to “0” (falling edge).

Figure 9.3.1. Interrupt Control Registers

Downloaded from Arrow.com.

Interrupt request cause select register

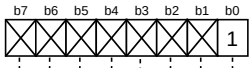


Symbol	Address	After reset
IFSR	$035F_{16}$	00_{16}

Bit symbol	Bit name	Function	RW
IFSR0	INT0 interrupt polarity switching bit	0 : One edge 1 : Both edges (1)	RW
IFSR1	INT1 interrupt polarity switching bit	0 : One edge 1 : Both edges (1)	RW
IFSR2	INT2 interrupt polarity switching bit	0 : One edge 1 : Both edges (1)	RW
IFSR3	INT3 interrupt polarity switching bit	0 : One edge 1 : Both edges (1)	RW
IFSR4	INT4 interrupt polarity switching bit	0 : One edge 1 : Both edges (1)	RW
IFSR5	INT5 interrupt polarity switching bit	0 : One edge 1 : Both edges (1)	RW
IFSR6	Interrupt request cause select bit	0 : Reserved 1 : $\overline{INT4}$	RW
IFSR7	Interrupt request cause select bit	0 : Reserved 1 : $\overline{INT5}$	RW

NOTE:
1. When setting this bit to “1” (= both edges), make sure the POL bit in the INT0IC to INT5IC register is set to “0” (= falling edge).

Interrupt request cause select register 2



Symbol	Address	After reset
IFSR2A	$035E_{16}$	$XXXXXXX0_2$

Bit symbol	Bit name	Function	RW
IFSR20	Reserved bit (1)	Must be set to “1”.	RW
— (b7-b1)	Nothing is assigned. When write, set to “0”. When read, their contents are indeterminate.		—

NOTE:
1. Set this bit to "1" before you enable interrupt after resetting.

Figure 9.3.2. IFSR Register and IFSR2A Register

Downloaded from Arrow.com.

9.3.1 I Flag

The I flag enables or disables the maskable interrupt. Setting the I flag to “1” (= enabled) enables the maskable interrupt. Setting the I flag to “0” (= disabled) disables all maskable interrupts.

9.3.2 IR Bit

The IR bit is set to “1” (= interrupt requested) when an interrupt request is generated. Then, when the interrupt request is accepted and the CPU branches to the corresponding interrupt vector, the IR bit is cleared to “0” (= interrupt not requested).
The IR bit can be cleared to “0” in a program. Note that do not write “1” to this bit.

9.3.3 ILVL2 to ILVL0 Bits and IPL

Interrupt priority levels can be set using the ILVL2 to ILVL0 bits.
Table 9.3.3.1 shows the settings of interrupt priority levels and Table 9.3.3.2 shows the interrupt priority levels enabled by the IPL.

The following are conditions under which an interrupt is accepted:
· I flag is set to “1”
· IR bit is set to “1”
· interrupt priority level > IPL

The I flag, IR bit, ILVL2 to ILVL0 bits and IPL are independent of each other. In no case do they affect one another.

Table 9.3.3.1. Settings of Interrupt Priority Levels

ILVL2 to ILVL0 bits	Interrupt priority level	Priority order
000_2	Level 0 (interrupt disabled)	———
001_2	Level 1	Low ↓
010_2	Level 2	
011_2	Level 3	
100_2	Level 4	
101_2	Level 5	
110_2	Level 6	
111_2	Level 7	High

Table 9.3.3.2. Interrupt Priority Levels Enabled by IPL

IPL	Enabled interrupt priority levels
000_2	Interrupt levels 1 and above are enabled
001_2	Interrupt levels 2 and above are enabled
010_2	Interrupt levels 3 and above are enabled
011_2	Interrupt levels 4 and above are enabled
100_2	Interrupt levels 5 and above are enabled
101_2	Interrupt levels 6 and above are enabled
110_2	Interrupt levels 7 and above are enabled
111_2	All maskable interrupts are disabled

Downloaded from Arrow.com.

9.4 Interrupt Sequence

An interrupt sequence (the devicebehavior from the instant an interrupt is accepted to the instant the interrupt routine is executed) is described here.

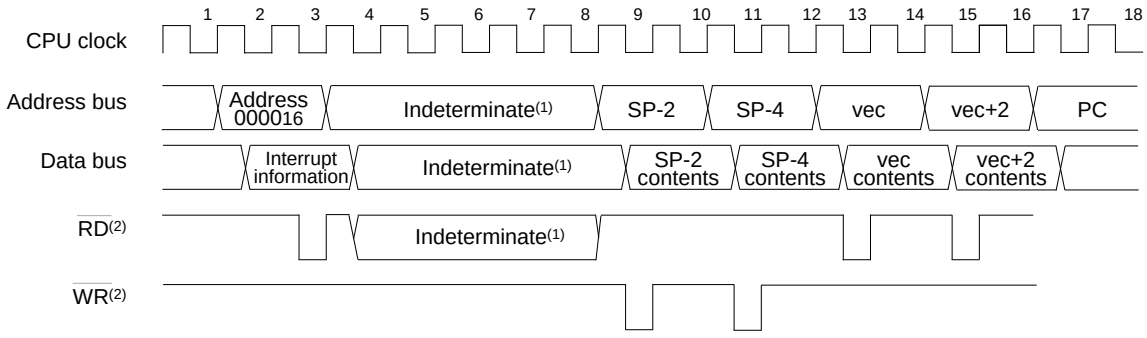
If an interrupt occurs during execution of an instruction, the processor determines its priority when the execution of the instruction is completed, and transfers control to the interrupt sequence from the next cycle. If an interrupt occurs during execution of either the SMOVB, SMOVF, SSTR or RMPA instruction, the processor temporarily suspends the instruction being executed, and transfers control to the interrupt sequence.

The CPU behavior during the interrupt sequence is described below. Figure 9.4.1 shows time required for executing the interrupt sequence.

(1) The CPU gets interrupt information (interrupt number and interrupt request priority level) by reading the address 00000_{16}. Then it clears the IR bit for the corresponding interrupt to “0” (interrupt not requested).
(2) The FLG register immediately before entering the interrupt sequence is saved to the CPU’s internal temporary register(1).
(3) The I, D and U flags in the FLG register become as follows:
The I flag is cleared to “0” (interrupts disabled).
The D flag is cleared to “0” (single-step interrupt disabled).
The U flag is cleared to “0” (ISP selected).
However, the U flag does not change state if an INT instruction for software interrupt Nos. 32 to 63 is executed.
(4) The CPU’s internal temporary register (1) is saved to the stack.
(5) The PC is saved to the stack.
(6) The interrupt priority level of the accepted interrupt is set in the IPL.
(7) The start address of the relevant interrupt routine set in the interrupt vector is stored in the PC.

After the interrupt sequence is completed, the processor resumes executing instructions from the start address of the interrupt routine.

NOTE:
1. This register cannot be used by user.



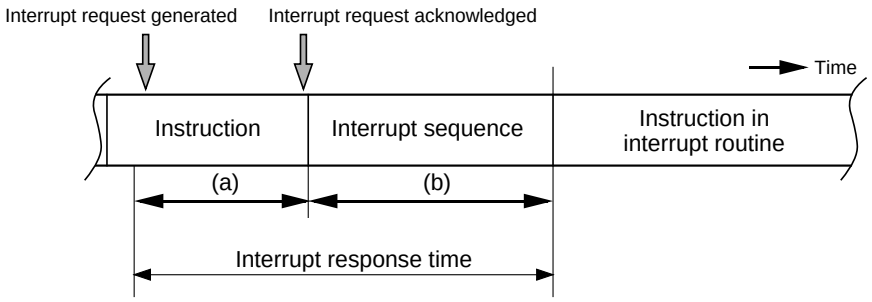
NOTES:
1. The indeterminate state depends on the instruction queue buffer. A read cycle occurs when the instruction queue buffer is ready to accept instructions.
2. $\overline{RD}$ is the internal signal which is set to “L” when the internal memory is read out and $\overline{WR}$ is the internal signal which is set to “L” when the internal memory is written.

Figure 9.4.1. Time Required for Executing Interrupt Sequence

Downloaded from Arrow.com.

9.4.1 Interrupt Response Time

Figure 9.4.1.1 shows the interrupt response time. The interrupt response or interrupt acknowledge time denotes the time from when an interrupt request is generated till when the first instruction in the interrupt routine is executed. Specifically, it consists of the time from when an interrupt request is generated till when the instruction then executing is completed ((a) in Figure 9.4.1.1) and the time during which the interrupt sequence is executed ((b) in Figure 9.4.1.1).



(a) The time from when an interrupt request is generated till when the instruction then executing is completed. The length of this time varies with the instruction being executed. The DIVX instruction requires the longest time, which is equal to 30 cycles (without wait state, the divisor being a register).

(b) The time during which the interrupt sequence is executed. For details, see the table below. Note, however, that the values in this table must be increased 2 cycles for the $\overline{\text{DBC}}$ interrupt and 1 cycle for the address match and single-step interrupts.

Interrupt vector address	SP value	Without wait
Even	Even	18 cycles
Even	Odd	19 cycles
Odd	Even	19 cycles
Odd	Odd	20 cycles

Figure 9.4.1.1. Interrupt response time

9.4.2 Variation of IPL when Interrupt Request is Accepted

When a maskable interrupt request is accepted, the interrupt priority level of the accepted interrupt is set in the IPL.

When a software interrupt or special interrupt request is accepted, one of the interrupt priority levels listed in Table 9.4.2.1 is set in the IPL. Shown in Table 9.4.2.1 are the IPL values of software and special interrupts when they are accepted.

Table 9.4.2.1. IPL Level That is Set to IPL When A Software or Special Interrupt Is Accepted

Interrupt sources	Level that is set to IPL
Watchdog timer, $\overline{\text{NMI}}$, Oscillation stop and re-oscillation detection, voltage down detection	7
Software, address match, $\overline{\text{DBC}}$, single-step	Not changed

Downloaded from Arrow.com.

9.4.3 Saving Registers

In the interrupt sequence, the FLG register and PC are saved to the stack.

At this time, the 4 high-order bits of the PC and the 4 high-order (IPL) and 8 low-order bits in the FLG register, 16 bits in total, are saved to the stack first. Next, the 16 low-order bits of the PC are saved. Figure 9.4.3.1 shows the stack status before and after an interrupt request is accepted.

The other necessary registers must be saved in a program at the beginning of the interrupt routine. Use the PUSHM instruction, and all registers except SP can be saved with a single instruction.

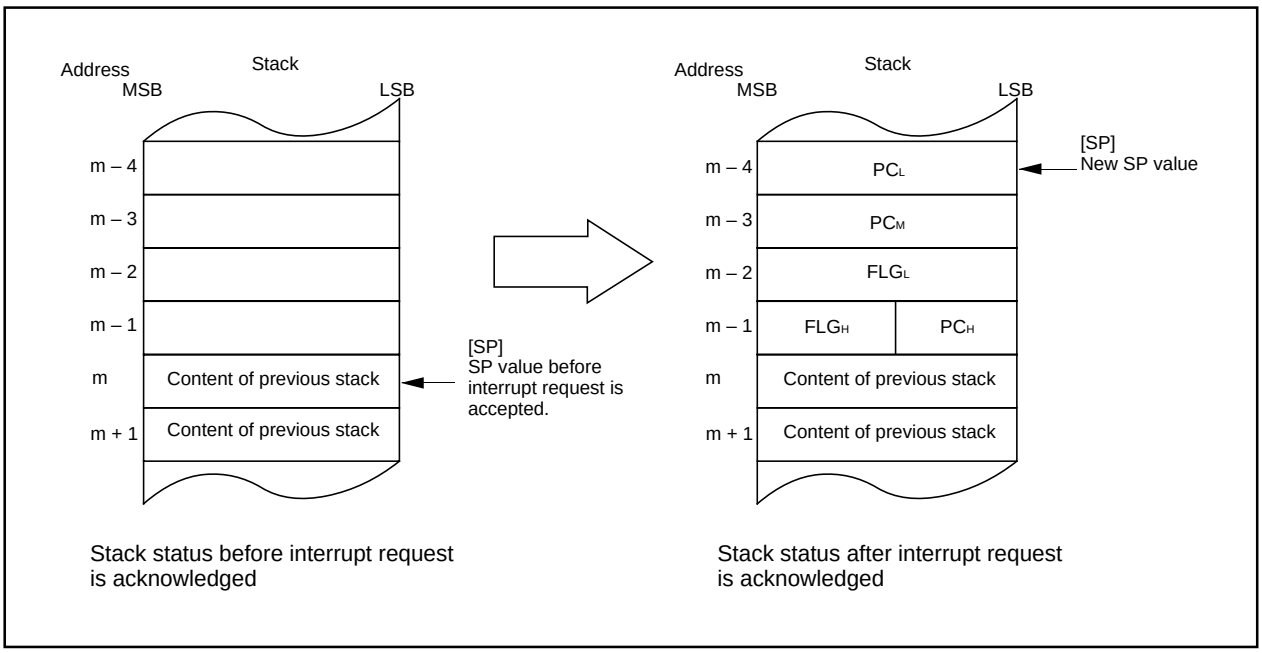


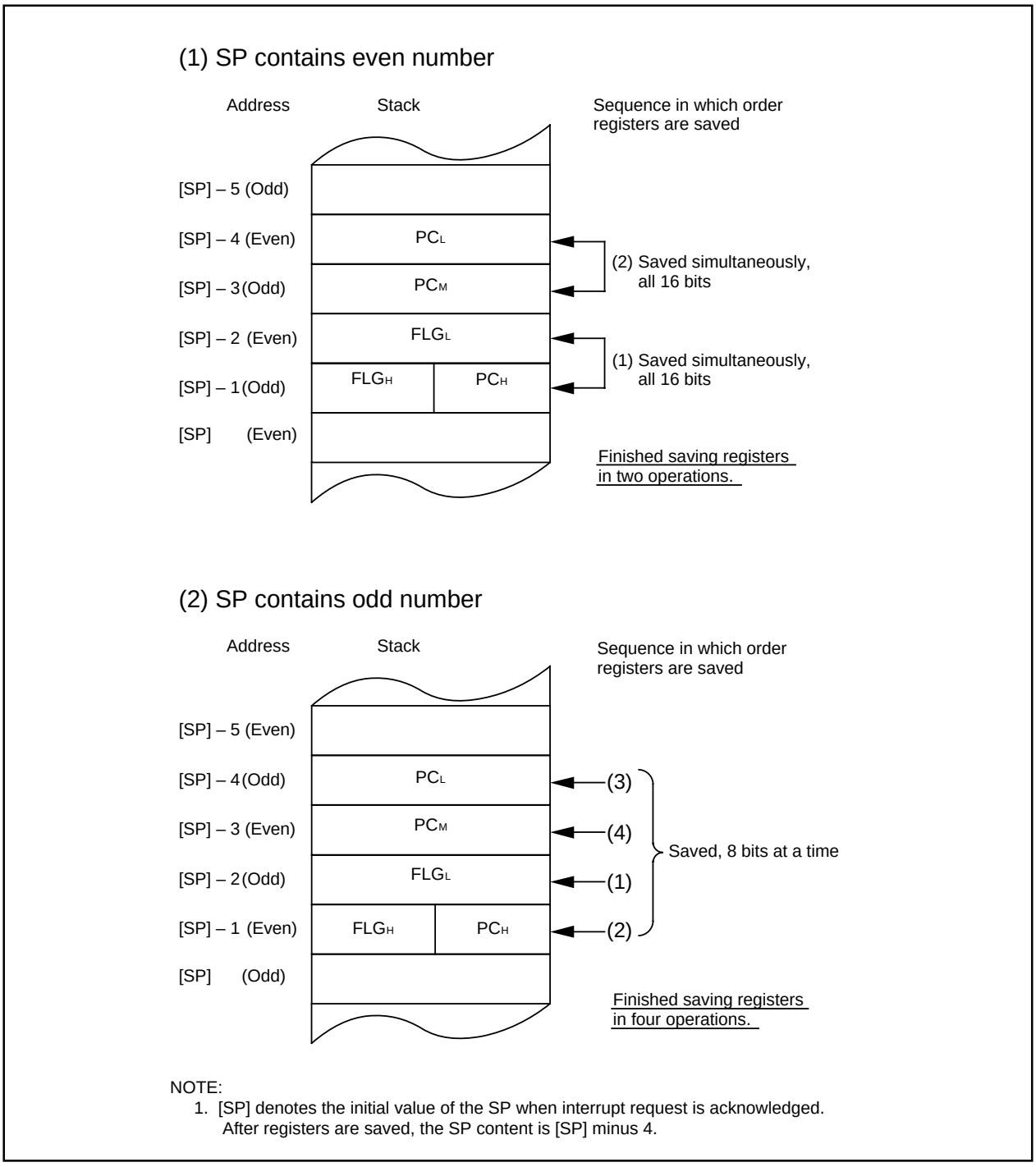
Figure 9.4.3.1. Stack Status Before and After Acceptance of Interrupt Request

Downloaded from Arrow.com.

The operation of saving registers carried out in the interrupt sequence is dependent on whether the SP(1), at the time of acceptance of an interrupt request, is even or odd. If the stack pointer (1) is even, the FLG register and the PC are saved, 16 bits at a time. If odd, they are saved in two steps, 8 bits at a time. **Figure 9.4.3.2** shows the operation of the saving registers.

NOTE:

1. When any INT instruction in software numbers 32 to 63 has been executed, this is the SP indicated by the U flag. Otherwise, it is the ISP.



NOTE:
1. [SP] denotes the initial value of the SP when interrupt request is acknowledged. After registers are saved, the SP content is [SP] minus 4.

Figure 9.4.3.2. Operation of Saving Register

Downloaded from Arrow.com.

9.4.4 Returning from an Interrupt Routine

The FLG register and PC in the state in which they were immediately before entering the interrupt sequence are restored from the stack by executing the REIT instruction at the end of the interrupt routine. Thereafter the CPU returns to the program which was being executed before accepting the interrupt request.

Return the other registers saved by a program within the interrupt routine using the POPM or similar instruction before executing the REIT instruction.

9.5 Interrupt Priority

If two or more interrupt requests are generated while executing one instruction, the interrupt request that has the highest priority is accepted.

For maskable interrupts (peripheral functions), any desired priority level can be selected using the ILVL2 to ILVL0 bits. However, if two or more maskable interrupts have the same priority level, their interrupt priority is resolved by hardware, with the highest priority interrupt accepted.

The watchdog timer and other special interrupts have their priority levels set in hardware. Figure 9.5.1 shows the priorities of hardware interrupts.

Software interrupts are not affected by the interrupt priority. If an instruction is executed, control branches invariably to the interrupt routine.

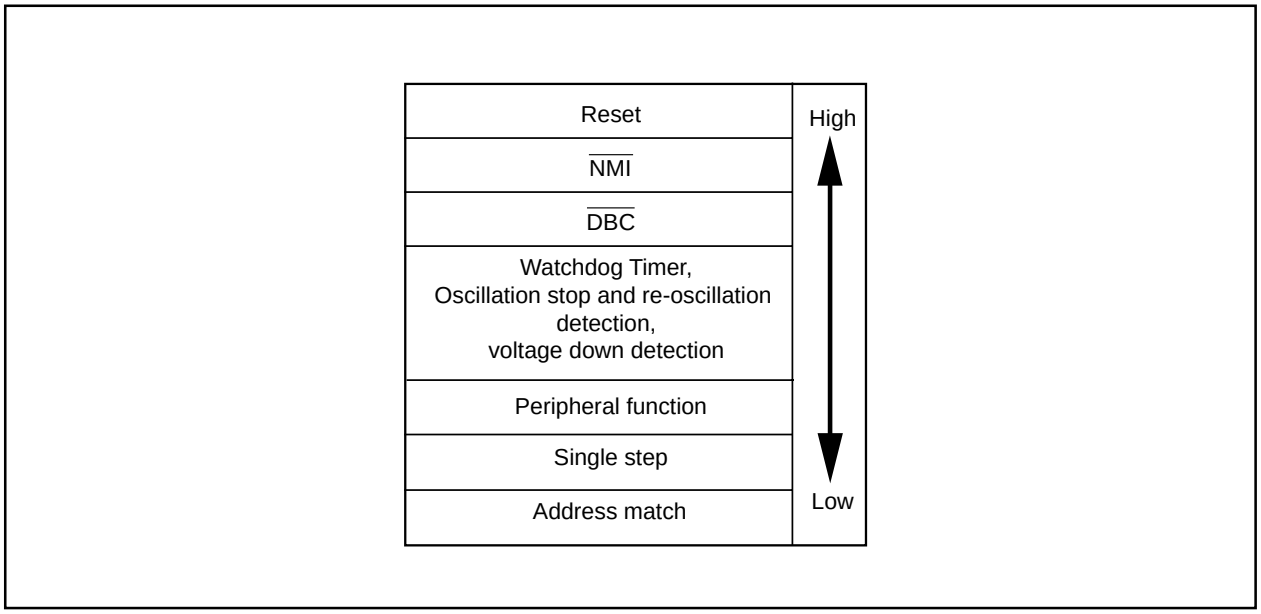


Figure 9.5.1. Hardware Interrupt Priority

9.5.1 Interrupt Priority Resolution Circuit

The interrupt priority resolution circuit is used to select the interrupt with the highest priority among those requested.

Figure 9.5.1.1 shows the circuit that judges the interrupt priority level.

Downloaded from Arrow.com.

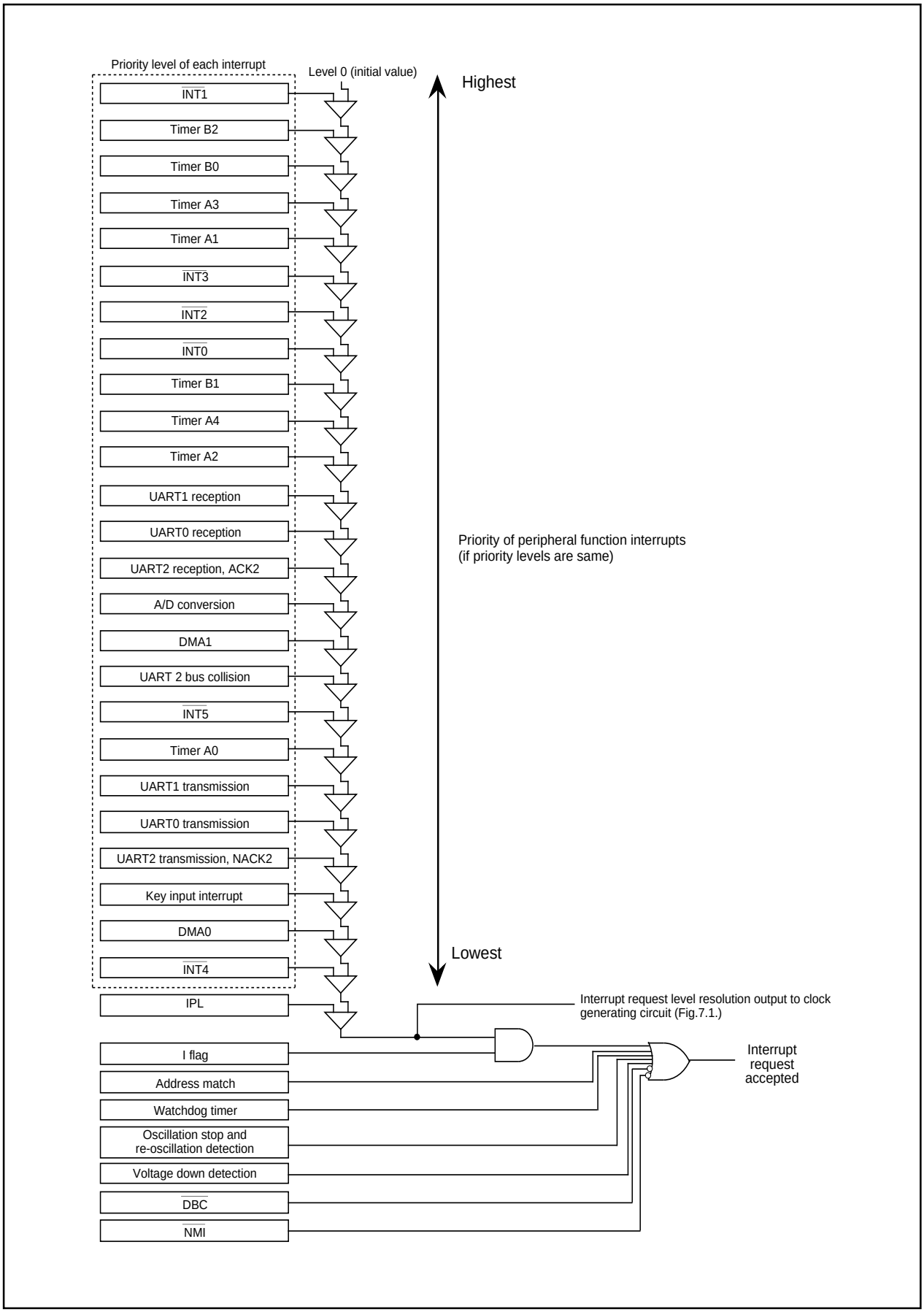


Figure 9.5.1.1. Interrupts Priority Select Circuit

Downloaded from Arrow.com.

9.6 $\overline{\text{INT}}$ Interrupt

$\overline{\text{INT}}$i interrupt (i=0 to 5) is triggered by the edges of external inputs. The edge polarity is selected using the IFSRi bit in the IFSR register.

To use the $\overline{\text{INT4}}$ interrupt, set the IFSR6 bit in the IFSR register to "1" (=$\overline{\text{INT4}}$). To use the $\overline{\text{INT5}}$ interrupt, set the IFSR7 bit in the IFSR register to "1" (=$\overline{\text{INT5}}$).

After modifiying the IFSR6 or IFSR7 bit, clear the corresponding IR bit to "0" (=interrupt not requested) before enabling the interrupt.

The $\overline{\text{INT5}}$ input has an effective digital debounce function for a noize rejection. Refer to **16.6 Digital Debounce function** for this detail. When using $\overline{\text{INT5}}$ interrupt to exit stop mode, set the P17DDR register to "FF16" before entering stop mode.

Figure 9.6.1 shows the IFSR register.

Interrupt request cause select register

b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	After reset
IFSR	035F16	0016

Bit symbol	Bit name	Function	RW
IFSR0	INT0 interrupt polarity switching bit	0 : One edge 1 : Both edges (1)	RW
IFSR1	INT1 interrupt polarity switching bit	0 : One edge 1 : Both edges (1)	RW
IFSR2	INT2 interrupt polarity switching bit	0 : One edge 1 : Both edges (1)	RW
IFSR3	INT3 interrupt polarity switching bit	0 : One edge 1 : Both edges (1)	RW
IFSR4	INT4 interrupt polarity switching bit	0 : One edge 1 : Both edges (1)	RW
IFSR5	INT5 interrupt polarity switching bit	0 : One edge 1 : Both edges (1)	RW
IFSR6	Interrupt request cause select bit	0 : Reserved 1 : $\overline{\text{INT4}}$	RW
IFSR7	Interrupt request cause select bit	0 : Reserved 1 : $\overline{\text{INT5}}$	RW

NOTE:
1. When setting this bit to “1” (= both edges), make sure the POL bit in the INT0IC to INT5IC register is set to “0” (= falling edge).

Figure 9.6.1. IFSR Register

Downloaded from Arrow.com.

9.7 $\overline{\text{NMI}}$ Interrupt

An $\overline{\text{NMI}}$ interrupt request is generated when input on the $\overline{\text{NMI}}$ pin changes state from high to low, after the $\overline{\text{NMI}}$ interrupt was enabled by writing a “1” to PM24 bit in the PM2 register. The $\overline{\text{NMI}}$ interrupt is a non-maskable interrupt, once it is enabled.

The input level of this $\overline{\text{NMI}}$ interrupt input pin can be read by accessing the P8_5 bit in the P8 register.

$\overline{\text{NMI}}$ is disabled by default after reset (the pin is a GPIO pin, $P8_5$) and can be enabled using PM24 bit in the PM2 register. Once enabled, it can only be disabled by a reset signal.

The $\overline{\text{NMI}}$ input has an effective digital debounce function for a noise rejection. Refer to **16.6 Digital Debounce Function** for this detail. When using $\overline{\text{NMI}}$ interrupt to exit stop mode, set the NDDR register to "FF_{16}" before entering stop mode.

9.8 Key Input Interrupt

Of $P10_4$ to $P10_7$, a key input interrupt is generated when input on any of the $P10_4$ to $P10_7$ pins which has had the PD10_4 to PD10_7 bits in the PD10 register set to “0” (= input) goes low. Key input interrupts can be used as a key-on wakeup function, the function which gets the microcomputer out of wait or stop mode. However, if you intend to use the key input interrupt, do not use $P10_4$ to $P10_7$ as analog input ports. Figure 9.8.1 shows the block diagram of the key input interrupt. Note, however, that while input on any pin which has had the PD10_4 to PD10_7 bits set to “0” (= input mode) is pulled low, inputs on all other pins of the port are not detected as interrupts.

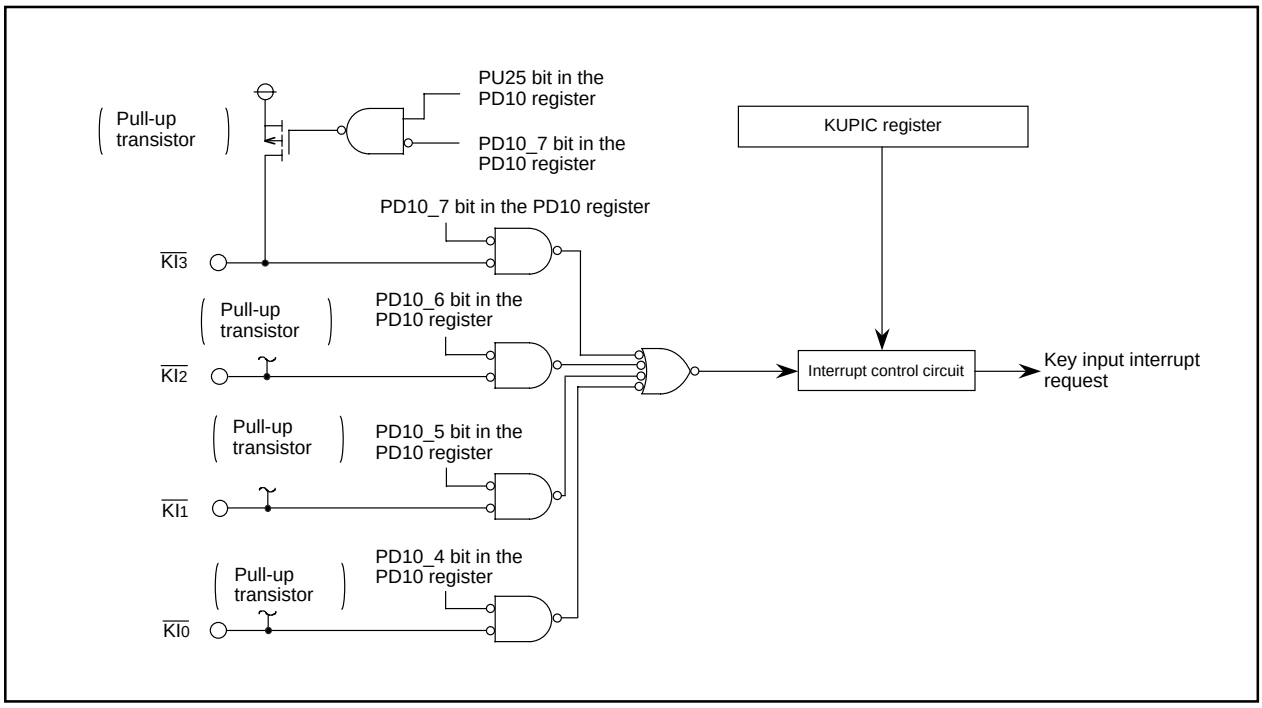


Figure 9.8.1. Key Input Interrupt

Downloaded from Arrow.com.

9.9 Address Match Interrupt

An address match interrupt request is generated immediately before executing the instruction at the address indicated by the RMADi register (i=0 to 1). Set the start address of any instruction in the RMADi register. Use the AIER register's AIER0 and AIER1 bits to enable or disable the interrupt. Note that the address match interrupt is unaffected by the I flag and IPL. For address match interrupts, the value of the PC that is saved to the stack area varies depending on the instruction being executed (refer to "Saving Registers").

(The value of the PC that is saved to the stack area is not the correct return address.) Therefore, follow one of the methods described below to return from the address match interrupt.

• Rewrite the content of the stack and then use the REIT instruction to return.

• Restore the stack to its previous state before the interrupt request was accepted by using the POP or similar other instruction and then use a jump instruction to return.

Table 9.9.1 shows the value of the PC that is saved to the stack area when an address match interrupt request is accepted.

Figure 9.9.1 shows the AIER, RMAD0 and RMAD1 registers.

Table 9.9.1. Value of the PC that is saved to the stack area when an address match interrupt request is accepted.

Instruction at the address indicated by the RMADi register	Value of the PC that is saved to the stack area
• 2-byte op-code instruction • 1-byte op-code instructions which are followed: ADD.B:S #IMM8,dest SUB.B:S #IMM8,dest AND.B:S #IMM8,dest OR.B:S #IMM8,dest MOV.B:S #IMM8,dest STZ.B #IMM8,dest STNZ.B #IMM8,dest STZX.B #IMM81,#IMM82,dest CMP.B:S #IMM8,dest PUSHM src POPM dest JMPS #IMM8 JSRS #IMM8 MOV.B:S #IMM,dest (However, dest=A0 or A1)	The address indicated by the RMADi register +2
Instructions other than the above	The address indicated by the RMADi register +1

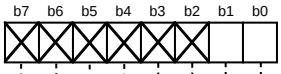
Value of the PC that is saved to the stack area : Refer to "Saving Registers".
Op-code is an abbreviation of Operation Code. It is a portion of instruction code.
Refer to Chapter 4 Instruction Code/Number of Cycles in M16C/60, M16C/20 Series Software Manual. Op-code is shown as a bold-framed figure directly below the Syntax.

Table 9.9.2. Relationship Between Address Match Interrupt Sources and Associated Registers

Address match interrupt sources	Address match interrupt enable bit	Address match interrupt register
Address match interrupt 0	AIER0	RMAD0
Address match interrupt 1	AIER1	RMAD1

Downloaded from Arrow.com.

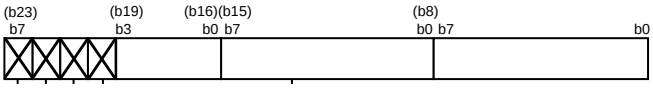
Address match interrupt enable register



Symbol	Address	After reset
AIER	0009_{16}	$XXXXXX00_2$

Bit symbol	Bit name	Function	RW
AIER0	Address match interrupt 0 enable bit	0 : Interrupt disabled 1 : Interrupt enabled	RW
AIER1	Address match interrupt 1 enable bit	0 : Interrupt disabled 1 : Interrupt enabled	RW
—— (b7-b2)	Nothing is assigned. When write, set to “0”. When read, their contents are indeterminate.		—

Address match interrupt register i (i = 0 to 1)



Symbol	Address	After reset
RMAD0	0012_{16} to 0010_{16}	$X0000_{16}$
RMAD1	0016_{16} to 0014_{16}	$X0000_{16}$

Function	Setting range	RW
Address setting register for address match interrupt	00000_{16} to $FFFFF_{16}$	RW
Nothing is assigned. When write, set to “0”. When read, their contents are indeterminate.		—

Figure 9.9.1. AIER Register, RMAD0 and RMAD1 Registers

Downloaded from Arrow.com.

10. Watchdog Timer

The watchdog timer is the function that detects when a program is out of control. Use the watchdog timer is recommended to improve reliability of the system. The watchdog timer contains a 15-bit counter which is decremented by the CPU clock that the prescaler divides. The PM12 bit in the PM1 register determines whether to generate a watchdog timer interrupt request or reset the watchdog timer when the watchdog timer underflows. The PM12 bit can only be set to "1" (reset). Once the PM12 bit is set to "1", it cannot be changed to "0" (watchdog timer interrupt) by program. Refer to "**5.3 Watchdog Timer Reset**" for watchdog timer reset.

When the main clock, on-chip oscillator clock, or PLL clock runs as CPU clock, the WDC7 bit in the WDC register determines whether the prescaler divides the clock by 16 or 128. When the sub clock runs as CPU clock, the prescaler divides the clock by 2 regardless of the WDC7 bit setting. Watchdog timer cycle is calculated as follows. Marginal errors, due to the prescaler, may occur in watchdog timer cycle.

With main clock source chosen for CPU clock, on-chip oscillator clock, PLL clock

$$\text{Watchdog timer period} = \frac{\text{Prescaler dividing (16 or 128) X Watchdog timer count (32768)}}{\text{CPU clock}}$$

With sub-clock chosen for CPU clock

$$\text{Watchdog timer period} = \frac{\text{Prescaler dividing (2) X Watchdog timer count (32768)}}{\text{CPU clock}}$$

For example, when CPU clock = 16 MHz and the divide-by-N value for the prescaler= 16, the watchdog timer period is approx. 32.8 ms.

The watchdog timer is initialized by writing to the WDTS register. The prescaler is initialized after reset. Note that the watchdog timer and the prescaler both are inactive after reset, so that the watchdog timer is activated to start counting by writing to the WDTS register.

Write the WDTS register with shorter cycle than the watchdog timer cycle. Set the WDTS register also in the beginning of the watchdog timer interrupt routine.

In stop mode, wait mode and when erase/program opration is excuting in EW1 mode without erase suspend requeired, the watchdog timer and prescaler are stopped. Counting is resumed from the held value when the modes or state are released.

Figure 10.1 shows the block diagram of the watchdog timer. Figure 10.2 shows the watchdog timer-related registers.

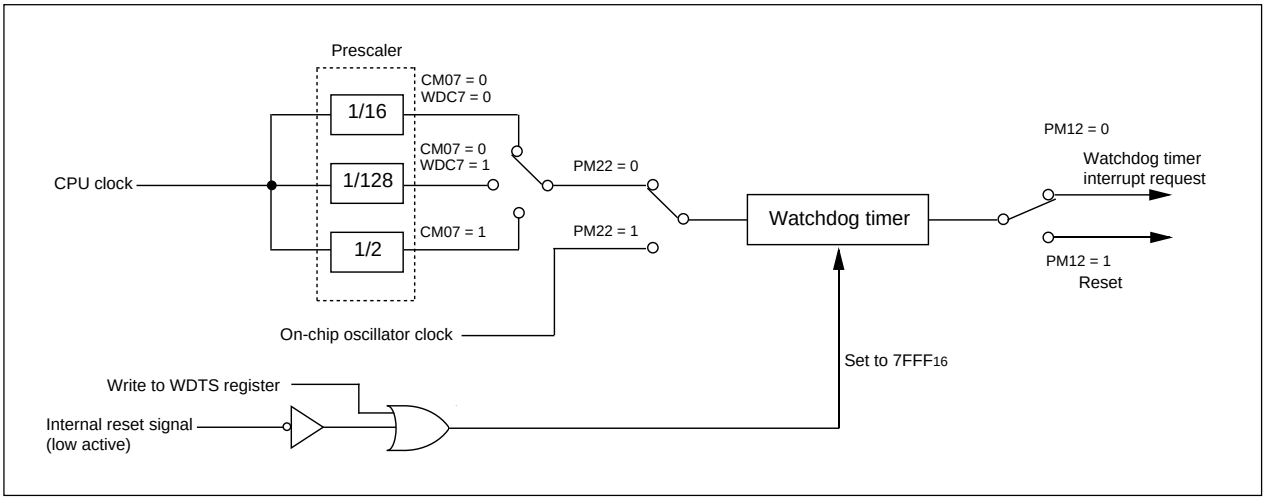


Figure 10.1. Watchdog Timer Block Diagram

Downloaded from Arrow.com.

Watchdog Timer Control Register

b7	b6	b5	b4	b3	b2	b1	b0
	0	0					

Symbol	Address	After Reset
WDC	$000F_{16}$	$00XXXXXX_2$

Bit Symbol	Bit Name	Function	RW
(b4-b0)	High-order bit of watchdog timer		RO
(b6-b5)	Reserved bit	Set to “0”	RW
WDC7	Prescaler select bit	0: Divided by 16 1: Divided by 128	RW

Watchdog Timer Start Register

b7	b0

Symbol	Address	After Reset
WDTS	$000E_{16}$	Indeterminate

Function	RW
The watchdog timer is initialized and starts counting after a write instruction to this register. The watchdog timer value is always initialized to “7FFF $_{16}$” regardless of whatever value is written.	WO

Figure 10.2 WDC Register and WDTS Register

10.1 Count Source Protective Mode

In this mode, a on-chip oscillator clock is used for the watchdog timer count source. The watchdog timer can be kept being clocked even when CPU clock stops as a result of run-away.

Before this mode can be used, the following register settings are required:

(1) Set the PRC1 bit in the PRCR register to “1” (enable writes to PM1 and PM2 registers).

(2) Set the PM12 bit in the PM1 register to “1” (reset when the watchdog timer underflows).

(3) Set the PM22 bit in the PM2 register to “1” (on-chip oscillator clock used for the watchdog timer count source).

(4) Set the PRC1 bit in the PRCR register to “0” (disable writes to PM1 and PM2 registers).

(5) Write to the WDTS register (watchdog timer starts counting).

Setting the PM22 bit to “1” results in the following conditions

- The on-chip oscillator continues oscillating even if the CM21 bit in the CM2 register is set to "0" (main clock or PLL clock) (system clock of count source selected by the CM21 bit is valid)
- The on-chip oscillator starts oscillating, and the in-chip oscillator clock becomes the watchdog timer count source.

$$\text{Watchdog timer period} = \frac{\text{Watchdog timer count (32768)}}{\text{on-chip oscillator clock}}$$

- The CM10 bit in the CM1 register is disabled against write. (Writing a “1” has no effect, nor is stop mode entered.)
- The watchdog timer does not stop when in wait mode.

Downloaded from Arrow.com.

11. DMAC

Note

Do not use UART0 transfer and UART0 reception interrupt request as a DMA request in the 42-pin package.

The DMAC (Direct Memory Access Controller) allows data to be transferred without the CPU intervention. Two DMAC channels are included. Each time a DMA request occurs, the DMAC transfers one (8 or 16-bit) data from the source address to the destination address. The DMAC uses the same data bus as used by the CPU. Because the DMAC has higher priority of bus control than the CPU and because it makes use of a cycle steal method, it can transfer one word (16 bits) or one byte (8 bits) of data within a very short time after a DMA request is generated. Figure 11.1 shows the block diagram of the DMAC. Table 11.1 shows the DMAC specifications. Figures 11.2 to 11.4 show the DMAC-related registers.

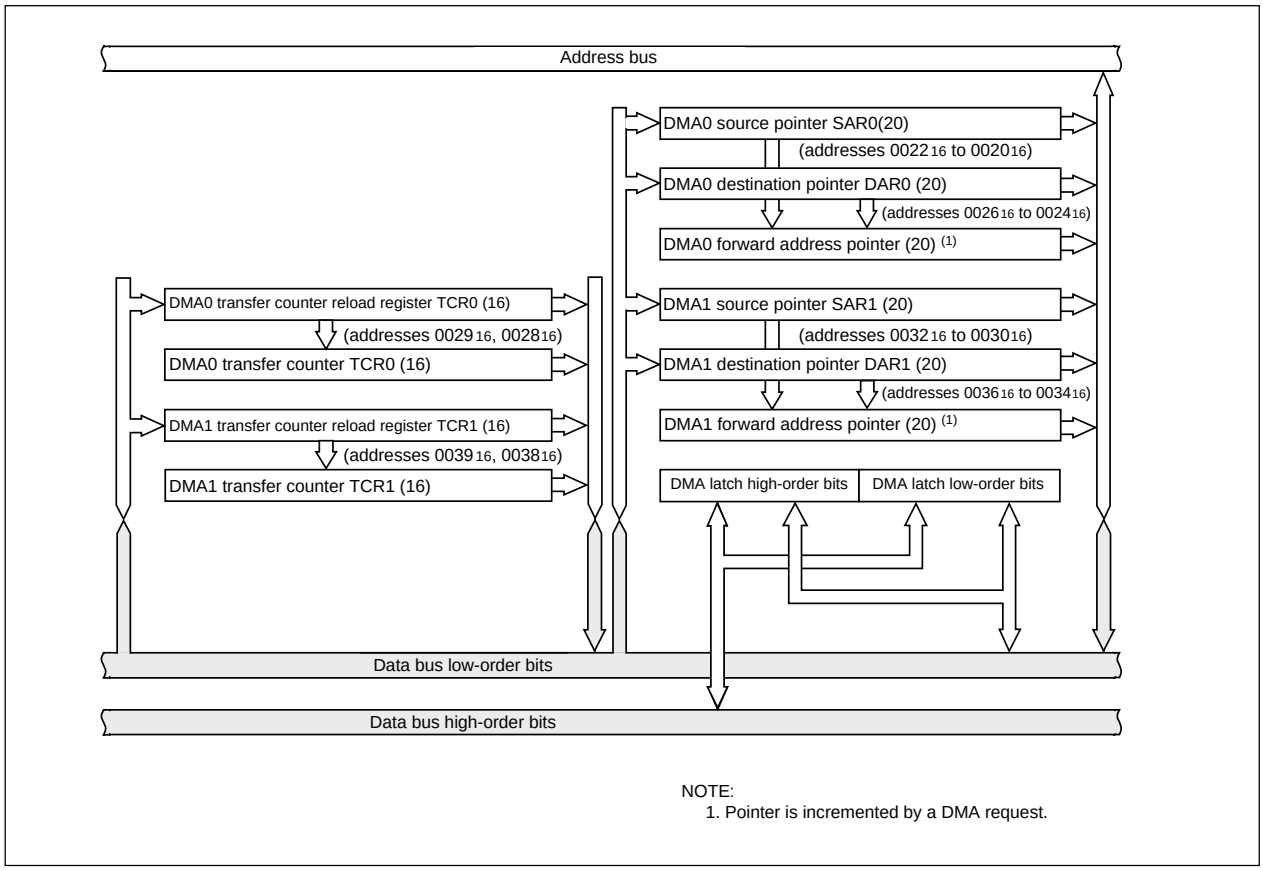


Figure 11.1 DMAC Block Diagram

A DMA request is generated by a write to the DSR bit in the DMiSL register (i = 0,1), as well as by an interrupt request which is generated by any function specified by the DMS and DSEL3 to DSEL0 bits in the DMiSL register. However, unlike in the case of interrupt requests, DMA requests are not affected by the I flag and the interrupt control register, so that even when interrupt requests are disabled and no interrupt request can be accepted, DMA requests are always accepted. Furthermore, because the DMAC does not affect interrupts, the IR bit in the interrupt control register does not change state due to a DMA transfer.

A data transfer is initiated each time a DMA request is generated when the DMAE bit in the DMiCON register is set to “1” (DMA enabled). However, if the cycle in which a DMA request is generated is faster than the DMA transfer cycle, the number of transfer requests generated and the number of times data is transferred may not match. For details, refer to **11.4 DMA Requests**.

Downloaded from Arrow.com.

Table 11.1 DMAC Specifications

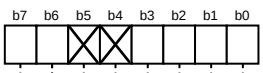
Item		Specification
No. of channels		2 (cycle steal method)
Transfer memory space		• From any address in the 1M bytes space to a fixed address • From a fixed address to any address in the 1M bytes space • From a fixed address to a fixed address
Maximum No. of bytes transferred		128K bytes (with 16-bit transfers) or 64K bytes (with 8-bit transfers)
DMA request factors (1, 2)		Falling edge of $\overline{INT0}$ or $\overline{INT1}$ Both edge of $\overline{INT0}$ or $\overline{INT1}$ Timer A0 to timer A4 interrupt requests Timer B0 to timer B2 interrupt requests UART0 transfer, UART0 reception interrupt requests UART1 transfer, UART1 reception interrupt requests UART2 transfer, UART2 reception interrupt requests A/D conversion interrupt requests Software triggers
Channel priority		DMA0 > DMA1 (DMA0 takes precedence)
Transfer unit		8 bits or 16 bits
Transfer address direction		forward or fixed (The source and destination addresses cannot both be in the forward direction.)
Transfer mode	Single transfer	Transfer is completed when the DMAi transfer counter (i = 0,1) underflows after reaching the terminal count.
	Repeat transfer	When the DMAi transfer counter underflows, it is reloaded with the value of the DMAi transfer counter reload register and a DMA transfer is con tinued with it.
DMA interrupt request generation timing		When the DMAi transfer counter underflowed
DMA startup		Data transfer is initiated each time a DMA request is generated when the DMAE bit in the DMAiCON register is set to “1” (enabled).
DMA shutdown	Single transfer	• When the DMAE bit is set to “0” (disabled) • After the DMAi transfer counter underflows
	Repeat transfer	When the DMAE bit is set to “0” (disabled)
		When a data transfer is started after setting the DMAE bit to “1” (en abled), the forward address pointer is reloaded with the value of the SARi or the DARi pointer whichever is specified to be in the forward direction and the DMAi transfer counter is reloaded with the value of the DMAi transfer counter reload register.

N OTES:

1. DMA transfer is not effective to any interrupt. DMA transfer is affected neither by the I flag nor by the interrupt control register.
2. The selectable causes of DMA requests differ with each channel.
3. Make sure that no DMAC-related registers (addresses 0020_{16} to $003F_{16}$) are accessed by the DMAC.

Downloaded from Arrow.com.

DMA0 request cause select register



Symbol: DM0SL | Address: 03B8$_{16}$ | After reset: 00$_{16}$

Bit symbol	Bit name	Function	RW
DSEL0	DMA request cause select bit	Refer to note	RW
DSEL1			RW
DSEL2			RW
DSEL3			RW
— (b5-b4)	Nothing is assigned. When write, set to “0”. When read, its content is “0”.		—
DMS	DMA request cause expansion select bit	0: Basic cause of request 1: Extended cause of request	RW
DSR	Software DMA request bit	A DMA request is generated by setting this bit to “1” when the DMS bit is “0” (basic cause) and the DSEL3 to DSEL0 bits are “0001 $_2$” (software trigger). The value of this bit when read is “0” .	RW

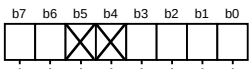
NOTE:
1. The causes of DMA0 requests can be selected by a combination of DMS bit and DSEL3 to DSEL0 bits in the manner described below.

DSEL3 to DSEL0	DMS=0(basic cause of request)	DMS=1(extended cause of request)
0 0 0 0$_2$	Falling edge of INT0 pin	–
0 0 0 1$_2$	Software trigger	–
0 0 1 0$_2$	Timer A0	–
0 0 1 1$_2$	Timer A1	–
0 1 0 0$_2$	Timer A2	–
0 1 0 1$_2$	Timer A3	–
0 1 1 0$_2$	Timer A4	Two edges of $\overline{\text{INT0}}$ pin
0 1 1 1$_2$	Timer B0	–
1 0 0 0$_2$	Timer B1	–
1 0 0 1$_2$	Timer B2	–
1 0 1 0$_2$	UART0 transmit	–
1 0 1 1$_2$	UART0 receive	–
1 1 0 0$_2$	UART2 transmit	–
1 1 0 1$_2$	UART2 receive	–
1 1 1 0$_2$	A/D conversion	–
1 1 1 1$_2$	UART1 transmit	–

Figure 11.2 DM0SL Register

Downloaded from Arrow.com.

DMA1 request cause select register



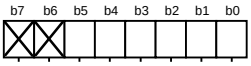
Symbol	Address	After reset
DM1SL	03BA16	0016

Bit symbol	Bit name	Function	RW
DSEL0	DMA request cause select bit	Refer to note	RW
DSEL1			RW
DSEL2			RW
DSEL3			RW
—— (b5-b4)	Nothing is assigned. When write, set to “0”. When read, its content is “0”.		——
DMS	DMA request cause expansion select bit	0: Basic cause of request 1: Extended cause of request	RW
DSR	Software DMA request bit	A DMA request is generated by setting this bit to “1” when the DMS bit is “0” (basic cause) and the DSEL3 to DSEL0 bits are “0001 2” (software trigger). The value of this bit when read is “0” .	RW

NOTE:
1. The causes of DMA1 requests can be selected by a combination of DMS bit and DSEL3 to DSEL0 bits in the manner described below.

DSEL3 to DSEL0	DMS=0(basic cause of request)	DMS=1(extended cause of request)
0 0 0 0 2	Falling edge of $\overline{\text{INT1}}$ pin	–
0 0 0 1 2	Software trigger	–
0 0 1 0 2	Timer A0	–
0 0 1 1 2	Timer A1	–
0 1 0 0 2	Timer A2	–
0 1 0 1 2	Timer A3	–
0 1 1 0 2	Timer A4	–
0 1 1 1 2	Timer B0	Two edges of $\overline{\text{INT1}}$
1 0 0 0 2	Timer B1	–
1 0 0 1 2	Timer B2	–
1 0 1 0 2	UART0 transmit	–
1 0 1 1 2	UART0 receive	–
1 1 0 0 2	UART2 transmit	–
1 1 0 1 2	UART2 receive/ACK2	–
1 1 1 0 2	A/D conversion	–
1 1 1 1 2	UART1 receive	–

DMAi control register(i=0,1)



Symbol	Address	After reset
DM0CON	002C16	00000X002
DM1CON	003C16	00000X002

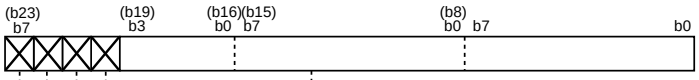
Bit symbol	Bit name	Function	RW
DMBIT	Transfer unit bit select bit	0 : 16 bits 1 : 8 bits	RW
DMASL	Repeat transfer mode select bit	0 : Single transfer 1 : Repeat transfer	RW
DMAS	DMA request bit	0 : DMA not requested 1 : DMA requested	RW (1)
DMAE	DMA enable bit	0 : Disabled 1 : Enabled	RW
DSD	Source address direction select bit (2)	0 : Fixed 1 : Forward	RW
DAD	Destination address direction select bit (2)	0 : Fixed 1 : Forward	RW
—— (b7-b6)	Nothing is assigned. When write, set to “0”. When read, its content is “0”.		——

NOTES:
1.The DMAS bit can be set to “0” by writing “0” in a program (This bit remains unchanged even if “1” is written).
2. At least one of the DAD and DSD bits must be “0” (address direction fixed).

Figure 11.3 DM1SL Register, DM0CON Register, and DM1CON Register

Downloaded from Arrow.com.

DMAi source pointer (i = 0, 1) (1)

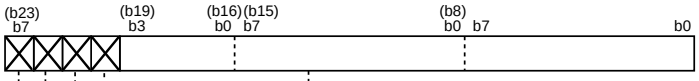


Symbol	Address	After reset
SAR0	0022$_{16}$ to 0020$_{16}$	Indeterminate
SAR1	0032$_{16}$ to 0030$_{16}$	Indeterminate

Function	Setting range	RW
Set the source address of transfer	00000$_{16}$ to FFFFF$_{16}$	RW
Nothing is assigned. When write, set “0”. When read, these contents are “0”.		—

NOTE:
1. If the DSD bit in the DMiCON register is “0” (fixed), this register can only be written to when the DMAE bit in the DMiCON register is “0” (DMA disabled). If the DSD bit is set to “1” (forward direction), this register can be written to at any time. If the DSD bit is set to “1” and the DMAE bit is “1” (DMA enabled), the DMAi forward address pointer can be read from this register. Otherwise, the value written to it can be read.

DMAi destination pointer (i = 0, 1)(1)

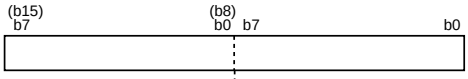


Symbol	Address	After reset
DAR0	0026$_{16}$ to 0024$_{16}$	Indeterminate
DAR1	0036$_{16}$ to 0034$_{16}$	Indeterminate

Function	Setting range	RW
Set the destination address of transfer	00000$_{16}$ to FFFFF$_{16}$	RW
Nothing is assigned. When write, set “0”. When read, these contents are “0”.		—

NOTE:
1. If the DAD bit in the DMiCON register is “0” (fixed), this register can only be written to when the DMAE bit in the DMiCON register is “0”(DMA disabled).If the DAD bit is set to “1” (forward direction), this register can be written to at any time. If the DAD bit is set to “1” and the DMAE bit is “1” (DMA enabled), the DMAi forward address pointer can be read from this register. Otherwise, the value written to it can be read.

DMAi transfer counter (i = 0, 1)



Symbol	Address	After reset
TCR0	0029$_{16}$, 0028$_{16}$	Indeterminate
TCR1	0039$_{16}$, 0038$_{16}$	Indeterminate

Function	Setting range	RW
Set the transfer count minus 1. The written value is stored in the DMAi transfer counter reload register, and when the DMAE bit in the DMiCON register is set to “1” (DMA enabled) or the DMAi transfer counter underflows when the DMASL bit in the DMiCON register is “1” (repeat transfer), the value of the DMAi transfer counter reload register is transferred to the DMAi transfer counter. When read, the DMAi transfer counter is read.	0000$_{16}$ to FFFF$_{16}$	RW

Figure 11.4 SAR0 and SAR1, DAR0 and DAR1, TCR0 and TCR1 Registers

Downloaded from Arrow.com.

11.1 Transfer Cycles

The transfer cycle consists of a memory or SFR read (source read) bus cycle and a write (destination write) bus cycle. The number of read and write bus cycles is affected by the source and destination addresses of transfer. Furthermore, the bus cycle itself is extended by a software wait.

11.1.1 Effect of Source and Destination Addresses

If the transfer unit is 16 bits and the source address of transfer begins with an odd address, the source read cycle consists of one more bus cycle than when the source address of transfer begins with an even address.

Similarly, if the transfer unit is 16 bits and the destination address of transfer begins with an odd address, the destination write cycle consists of one more bus cycle than when the destination address of transfer begins with an even address.

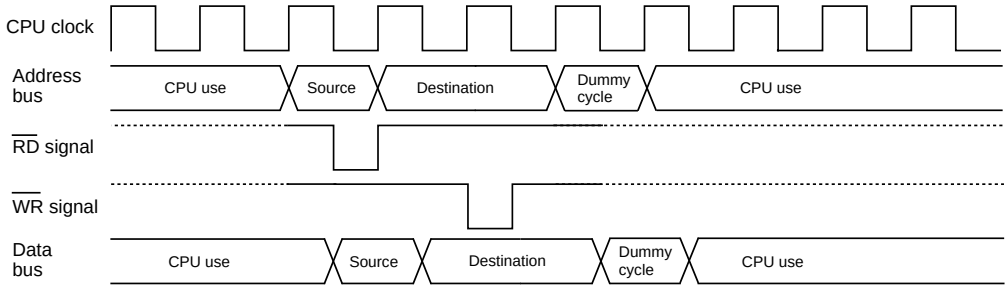
11.1.2 Effect of Software Wait

For memory or SFR accesses in which one or more software wait states are inserted, the number of bus cycles required for that access increases by an amount equal to software wait states.

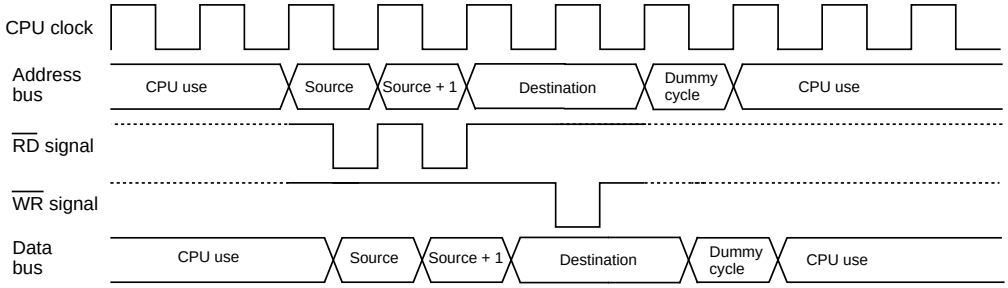
Figure 11.1.1 shows the example of the cycles for a source read. For convenience, the destination write cycle is shown as one cycle and the source read cycles for the different conditions are shown. In reality, the destination write cycle is subject to the same conditions as the source read cycle, with the transfer cycle changing accordingly. When calculating transfer cycles, take into consideration each condition for the source read and the destination write cycle, respectively. For example, when data is transferred in 16 bit units and when both the source address and destination address are an odd address ((2) in Figure 11.1.1), two source read bus cycles and two destination write bus cycles are required.

Downloaded from Arrow.com.

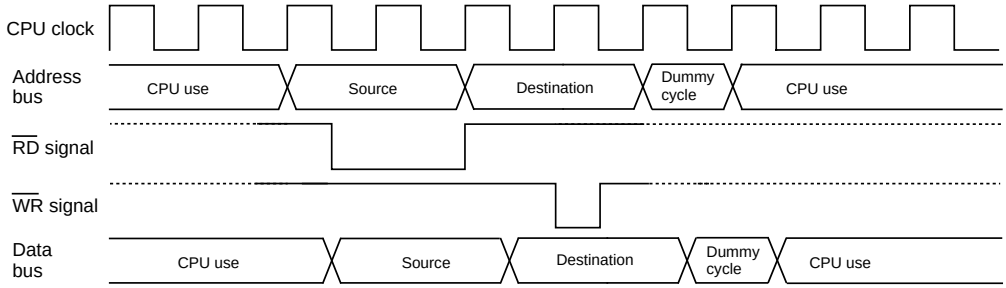
(1) When the transfer unit is 8 or 16 bits and the source of transfer is an even address



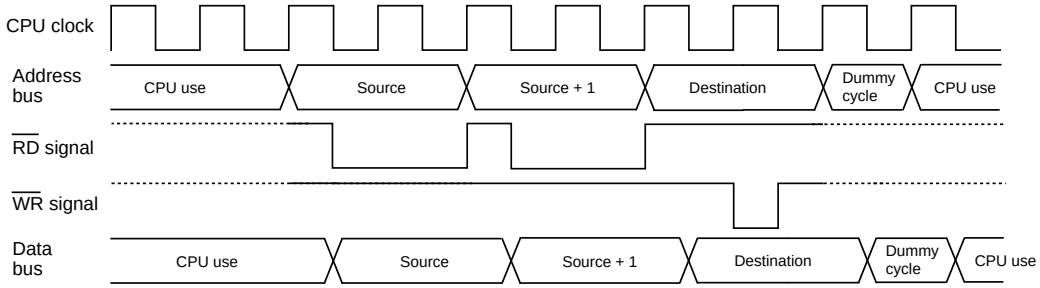
(2) When the transfer unit is 16 bits and the source address of transfer is an odd address



(3) When the source read cycle under condition (1) has one wait state inserted



(4) When the source read cycle under condition (2) has one wait state inserted



NOTE:
1. The same timing changes occur with the respective conditions at the destination as at the source.

Figure 11.1.1 Transfer Cycles for Source Read

Downloaded from Arrow.com.

11.2. DMA Transfer Cycles

Any combination of even or odd transfer read and write adresses is possible. Table 11.2.1 shows the number of DMA transfer cycles. Table 11.2.2 shows the Coefficient j, k.
The number of DMAC transfer cycles can be calculated as follows:

No. of transfer cycles per transfer unit = No. of read cycles x j + No. of write cycles x k

Table 11.2.1 DMA Transfer Cycles

Transfer unit	Access address	No. of read cycles	No. of write cycles
8-bit transfers (DMBIT= “1”)	Even	1	1
	Odd	1	1
16-bit transfers (DMBIT= “0”)	Even	1	1
	Odd	2	2

Table 11.2.2 Coefficient j, k

	Internal area			
	Internal ROM, RAM		SFR	
	No wait	With wait	1 wait (1)	2 wait (1)
j	1	2	2	3
k	1	2	2	3

NOTE:
1. Depends on the set value of PM20 bit in PM2 register.

Downloaded from Arrow.com.

11.3 DMA Enable

When a data transfer starts after setting the DMAE bit in DMiCON register (i = 0, 1) to “1” (enabled), the DMAC operates as follows:

(1) Reload the forward address pointer with the SARi register value when the DSD bit in the DMiCON register is “1” (forward) or the DARi register value when the DAD bit in the DMiCON register is “1” (forward).

(2) Reload the DMAi transfer counter with the DMAi transfer counter reload register value.

If the DMAE bit is set to “1” again while it remains set, the DMAC performs the above operation. However, if a DMA request may occur simultaneously when the DMAE bit is being written, follow the steps below.

Step 1: Write “1” to the DMAE bit and DMAS bit in DMiCON register simultaneously.

Step 2: Make sure that the DMAi is in an initial state as described above (1) and (2) in a program.

If the DMAi is not in an initial state, the above steps should be repeated.

11.4 DMA Request

The DMAC can generate a DMA request as triggered by the cause of request that is selected with the DMS and DSEL3 to DSEL0 bits in the DMiSL register (i = 0, 1) on either channel. Table 11.4.1 shows the timing at which the DMAS bit changes state.

Whenever a DMA request is generated, the DMAS bit is set to “1” (DMA requested) regardless of whether or not the DMAE bit is set. If the DMAE bit was set to “1” (enabled) when this occurred, the DMAS bit is set to “0” (DMA not requested) immediately before a data transfer starts. This bit cannot be set to “1” in a program (it can only be set to “0”).

The DMAS bit may be set to “1” when the DMS or the DSEL3 to DSEL0 bits change state. Therefore, always be sure to set the DMAS bit to “0” after changing the DMS or the DSEL3 to DSEL0 bits.

Because if the DMAE bit is “1”, a data transfer starts immediately after a DMA request is generated, the DMAS bit in almost all cases is “0” when read in a program. Read the DMAE bit to determine whether the DMAC is enabled.

Table 11.4.1 Timing at Which the DMAS Bit Changes State

DMA factor	DMAS bit in the DMiCON register	
	Timing at which the bit is set to “1”	Timing at which the bit is set to “0”
Software trigger	When the DSR bit in the DMiSL register is set to “1”	• Immediately before a data transfer starts • When set by writing “0” in a program
Peripheral function	When the interrupt control register for the peripheral function that is selected by the DSEL3 to DSEL0 and DMS bits in the DMiSL register has its IR bit set to “1”	

Downloaded from Arrow.com.

11.5 Channel Priority and DMA Transfer Timing

If both DMA0 and DMA1 are enabled and DMA transfer request signals from DMA0 and DMA1 are detected active in the same sampling period (one period from a falling edge to the next falling edge of CPU clock), the DMAS bit on each channel is set to “1” (DMA requested) at the same time. In this case, the DMA requests are arbitrated according to the channel priority, DMA0 > DMA1. The following describes DMAC operation when DMA0 and DMA1 requests are detected active in the same sampling period. Figure 11.5.1 shows an example of DMA transfer effected by external factors.

DMA0 request having priority is received first to start a transfer when a DMA0 request and DMA1 request are generated simultanelously. After one DMA0 transfer is completed, a bus arbitration is returned to the CPU. When the CPU has completed one bus access, a DMA1 transfer starts. After one DMA1 transfer is completed, the bus arbitration is again returned to the CPU.

In addition, DMA requsts cannot be counted up since each channel has one DMAS bit. Therefore, when DMA requests, as DMA1 in Figure 11.5.1, occurs more than one time, the DAMS bit is set to "0" as soon as getting the bus arbitration. The bus arbitration is returned to the CPU when one transfer is completed.

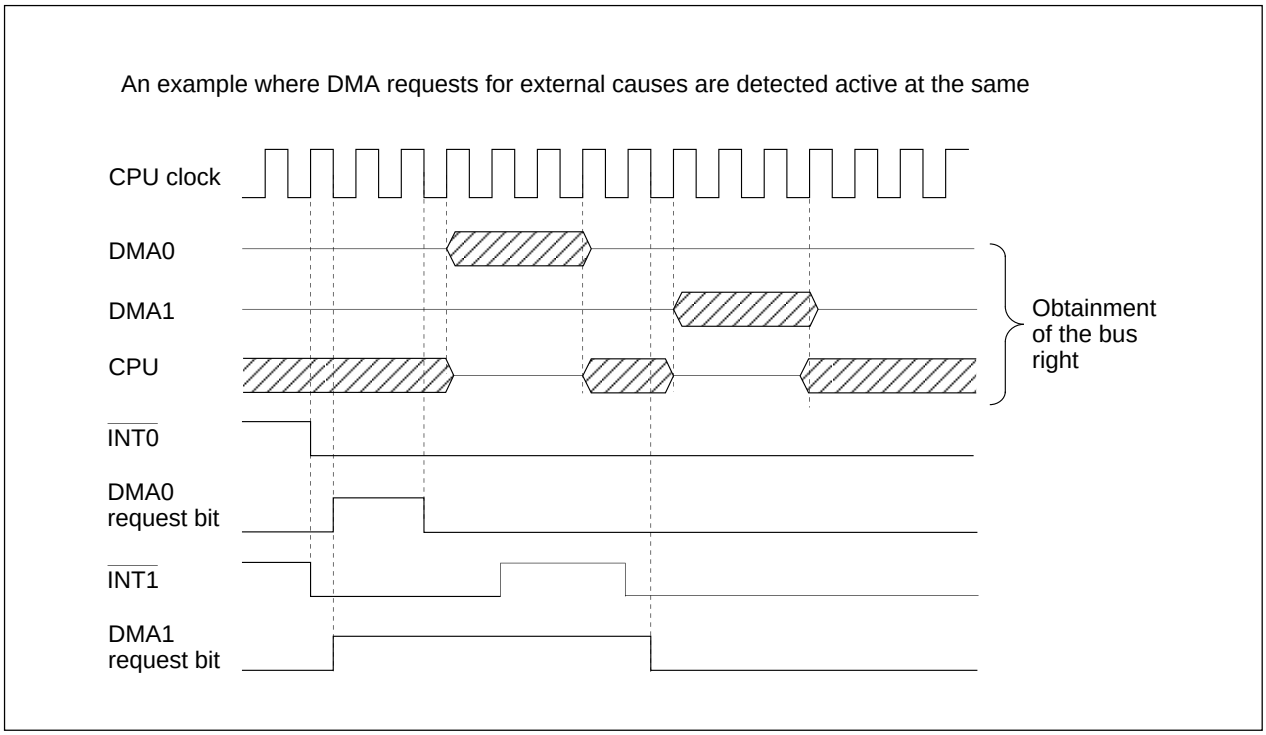


Figure 11.5.1 DMA Transfer by External Factors

Downloaded from Arrow.com.

12. Timer

Note

The TB2IN pin is not available in the 42-pin package. Do not use functions associated with the TB2IN pin.

Eight 16-bit timers, each capable of operating independently of the others, can be classified by function as either timer A (five) and timer B (three). The count source for each timer acts as a clock, to control such timer operations as counting, reloading, etc. Figures 12.1 and 12.2 show block diagrams of timer A and timer B configuration, respectively.

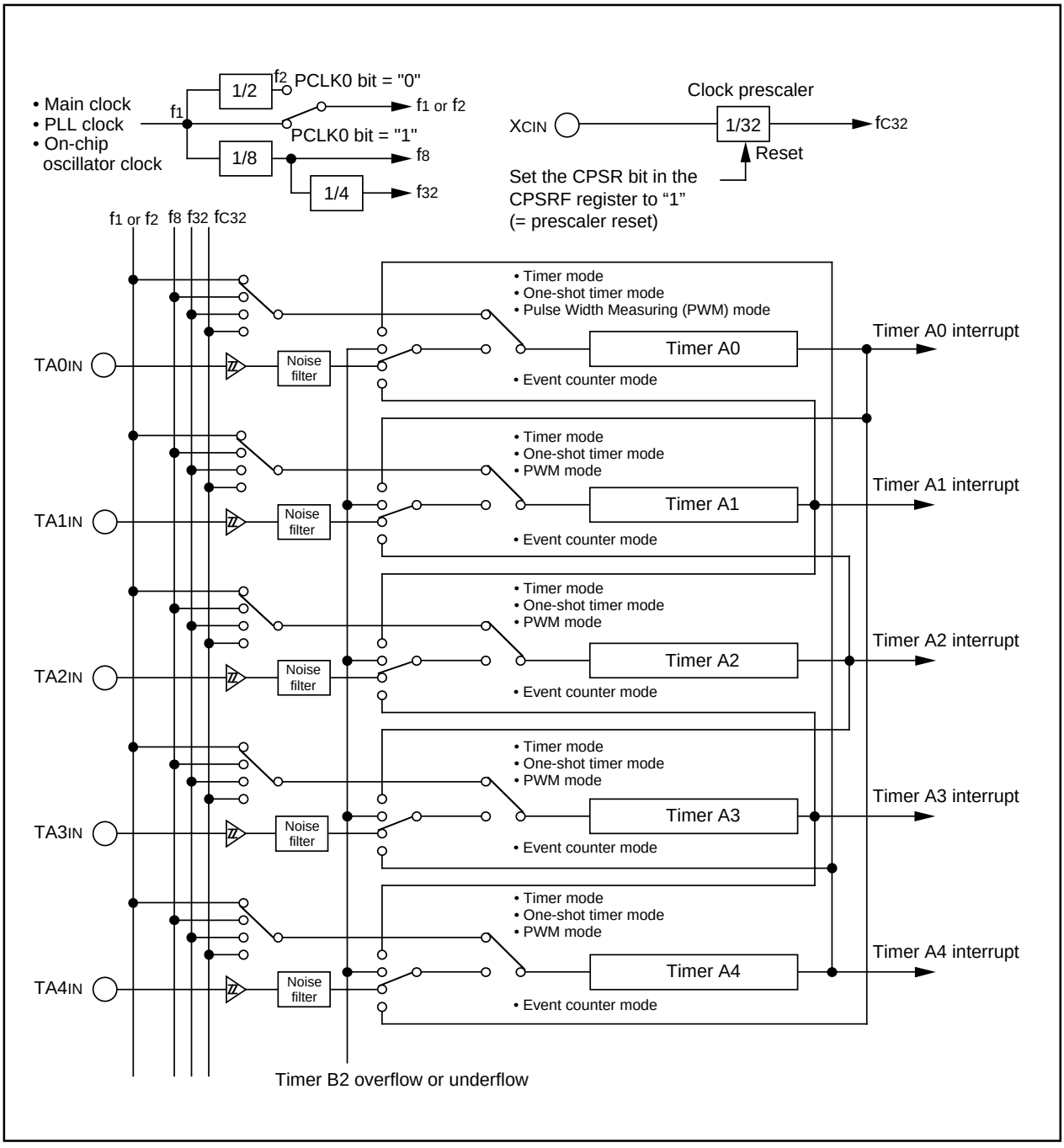


Figure 12.1. Timer A Configuration

Downloaded from Arrow.com.

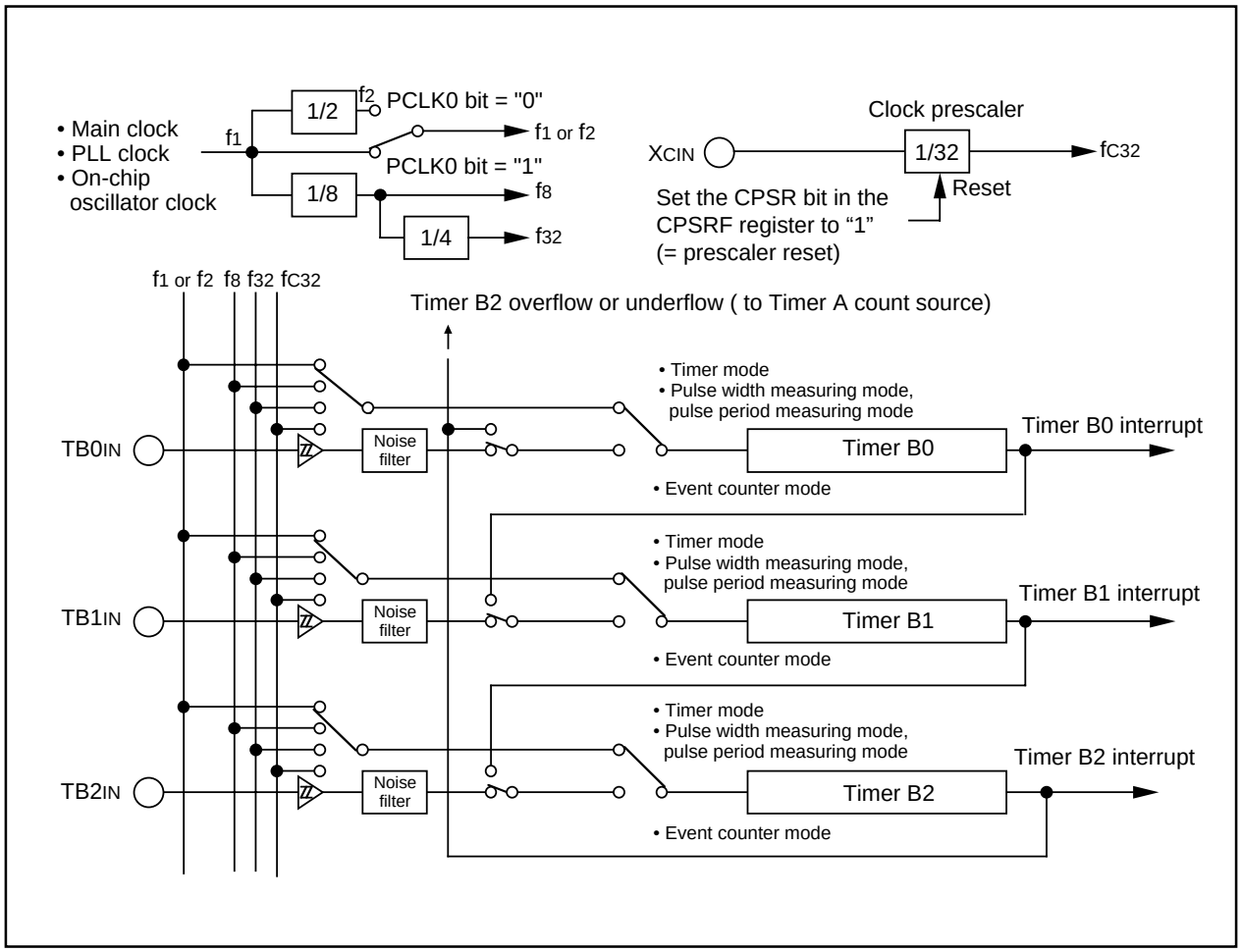


Figure 12.2. Timer B Configuration

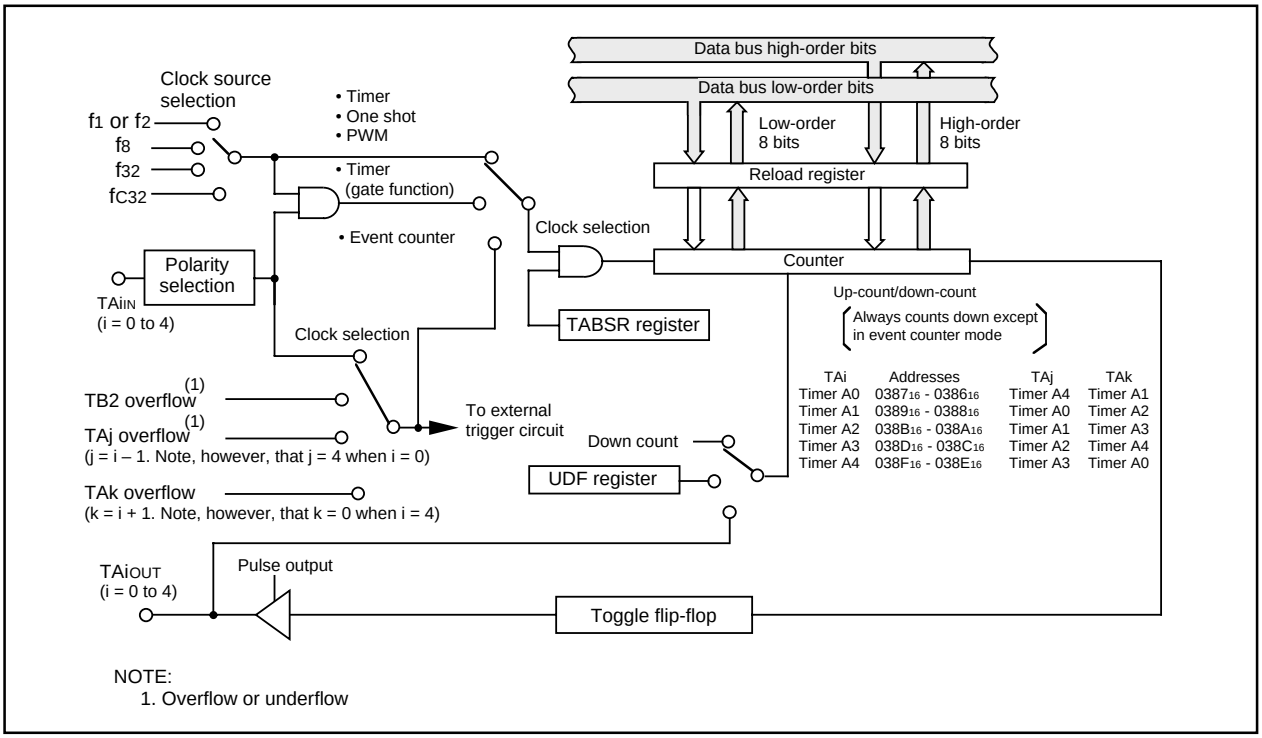
Downloaded from Arrow.com.

12.1 Timer A

Figure 12.1.1 shows a block diagram of the timer A. Figures 12.1.2 to 12.1.4 show registers related to the timer A.

The timer A supports the following four modes. Except in event counter mode, timers A0 to A4 all have the same function. Use the TMOD1 to TMOD0 bits in the TAiMR register (i = 0 to 4) to select the desired mode.

- Timer mode: The timer counts an internal count source.
- Event counter mode: The timer counts pulses from an external device or overflows and underflows of other timers.
- One-shot timer mode: The timer outputs a pulse only once before it reaches the minimum count "0000_{16}."
- Pulse width modulation (PWM) mode: The timer outputs pulses in a given width successively.



TAi	Addresses	TAj	TAk
Timer A0	0387_{16} - 0386_{16}	Timer A4	Timer A1
Timer A1	0389_{16} - 0388_{16}	Timer A0	Timer A2
Timer A2	$038B_{16}$ - $038A_{16}$	Timer A1	Timer A3
Timer A3	$038D_{16}$ - $038C_{16}$	Timer A2	Timer A4
Timer A4	$038F_{16}$ - $038E_{16}$	Timer A3	Timer A0

Figure 12.1.1. Timer A Block Diagram

Timer Ai mode register (i=0 to 4)

Symbol	Address	After reset
TA0MR to TA4MR	0396_{16} to $039A_{16}$	00_{16}

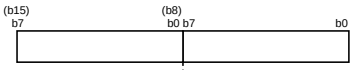
b7	b6	b5	b4	b3	b2	b1	b0

Bit symbol	Bit name	Function	RW
TMOD0	Operation mode select bit	b1 b0 0 0 : Timer mode 0 1 : Event counter mode 1 0 : One-shot timer mode 1 1 : Pulse width modulation (PWM) mode	RW
TMOD1			RW
MR0	———	Function varies with each operation mode	RW
MR1			RW
MR2			RW
MR3			RW
TCK0	Count source select bit	Function varies with each operation mode	RW
TCK1			RW

Figure 12.1.2. TA0MR to TA4MR Registers

Downloaded from Arrow.com.

Timer Ai register (i= 0 to 4) (1)



Symbol	Address	After reset
TA0	0387_{16}, 0386_{16}	Indeterminate
TA1	0389_{16}, 0388_{16}	Indeterminate
TA2	$038B_{16}$, $038A_{16}$	Indeterminate
TA3	$038D_{16}$, $038C_{16}$	Indeterminate
TA4	$038F_{16}$, $038E_{16}$	Indeterminate

Mode	Function	Setting range	RW
Timer mode	Divide the count source by n + 1 where n = set value	0000_{16} to $FFFF_{16}$	RW
Event counter mode	Divide the count source by $FFFF_{16}$ – n + 1 where n = set value when counting up or by n + 1 when counting down (5)	0000_{16} to $FFFF_{16}$	RW
One-shot timer mode	Divide the count source by n where n = set value and cause the timer to stop	0000_{16} to $FFFF_{16}$ (2, 4)	WO
Pulse width modulation mode (16-bit PWM)	Modify the pulse width as follows: PWM period: $(2^{16} – 1)$ / fj High level PWM pulse width: n / fj where n = set value, fj = count source frequency	0000_{16} to $FFFE_{16}$ (3, 4)	WO
Pulse width modulation mode (8-bit PWM)	Modify the pulse width as follows: PWM period: $(2^8 – 1)$ x (m + 1)/ fj High level PWM pulse width: (m + 1)n / fj where n = high-order address set value, m = low-order address set value, fj = count source frequency	00_{16} to FE_{16} (High-order address) 00_{16} to FF_{16} (Low-order address) (3, 4)	WO

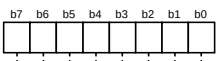
NOTES:
1. The register must be accessed in 16 bit units.
2. : If the TAi register is set to '0000_{16},' the counter does not work and timer Ai interrupt requests are not generated either. Furthermore, if "pulse output" is selected, no pulses are output from the TAiOUT pin.
3. If the TAi register is set to '0000_{16},' the pulse width modulator does not work, the output level on the TAiOUT pin remains low, and timer Ai interrupt requests are not generated either. The same applies when the 8 high-order bits of the timer TAi register are set to '00_{16}' while operating as an 8-bit pulse width modulator.
4. Use the MOV instruction to write to the TAi register.
5. The timer counts pulses from an external device or overflows or underflows in other timers.

Count start flag

Symbol	Address	After reset
TABSR	0380_{16}	00_{16}

Bit symbol	Bit name	Function	RW
TA0S	Timer A0 count start flag	0 : Stops counting 1 : Starts counting	RW
TA1S	Timer A1 count start flag		RW
TA2S	Timer A2 count start flag		RW
TA3S	Timer A3 count start flag		RW
TA4S	Timer A4 count start flag		RW
TB0S	Timer B0 count start flag		RW
TB1S	Timer B1 count start flag		RW
TB2S	Timer B2 count start flag		RW

Up/down flag (1)



Symbol	Address	After reset
UDF	0384_{16}	00_{16}

Bit symbol	Bit name	Function	RW
TA0UD	Timer A0 up/down flag	0 : Down count 1 : Up count Enabled by setting the TAiMR register's MR2 bit to "0" (= switching source in UDF register) during event counter mode.	RW
TA1UD	Timer A1 up/down flag		RW
TA2UD	Timer A2 up/down flag		RW
TA3UD	Timer A3 up/down flag		RW
TA4UD	Timer A4 up/down flag		RW
TA2P	Timer A2 two-phase pulse signal processing select bit	0 : two-phase pulse signal processing disabled 1 : two-phase pulse signal processing enabled (2, 3)	WO
TA3P	Timer A3 two-phase pulse signal processing select bit		WO
TA4P	Timer A4 two-phase pulse signal processing select bit		WO

NOTES:
1. Use MOV instruction to write to this register.
2. Make sure the port direction bits for the TA2IN to TA4IN and TA2OUT to TA4OUT pins are set to "0" (input mode).

Figure 12.1.3. TA0 to TA4 Registers, TABSR Register, and UDF Register

Downloaded from Arrow.com.

One-shot start flag

b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	After reset
ONSF	0382₁₆	00₁₆

Bit symbol	Bit name	Function	RW
TA0OS	Timer A0 one-shot start flag	The timer starts counting by setting this bit to “1” while the TMOD1 to TMOD0 bits in the TAiMR register (i = 0 to 4) is set to ‘10₂’ (= one-shot timer mode) and the MR2 bit in the TAiMR register is set to “0” (=TAiOS bit enabled). When read, its content is “0”.	RW
TA1OS	Timer A1 one-shot start flag		RW
TA2OS	Timer A2 one-shot start flag		RW
TA3OS	Timer A3 one-shot start flag		RW
TA4OS	Timer A4 one-shot start flag		RW
TAZIE	Z-phase input enable bit	0 : Z-phase input disabled 1 : Z-phase input enabled	RW
TA0TGL	Timer A0 event/trigger select bit	b7 b6 0 0 : Input on TA0IN is selected (1) 0 1 : TB2 overflow is selected (2) 1 0 : TA4 overflow is selected (2) 1 1 : TA1 overflow is selected (2)	RW
TA0TGH			RW

NOTES:
1. Make sure the PD7_1 bit in the PD7 register is set to “0” (= input mode).
2. Overflow or underflow.

Trigger select register

b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	After reset
TRGSR	0383₁₆	00₁₆

Bit symbol	Bit name	Function	RW
TA1TGL	Timer A1 event/trigger select bit	b1 b0 0 0 : Input on TA1IN is selected (1) 0 1 : TB2 overflow is selected (2) 1 0 : TA0 overflow is selected (2) 1 1 : TA2 overflow is selected (2)	RW
TA1TGH			RW
TA2TGL	Timer A2 event/trigger select bit	b3 b2 0 0 : Input on TA2IN is selected (1) 0 1 : TB2 overflow is selected (2) 1 0 : TA1 overflow is selected (2) 1 1 : TA3 overflow is selected (2)	RW
TA2TGH			RW
TA3TGL	Timer A3 event/trigger select bit	b5 b4 0 0 : Input on TA3IN is selected (1) 0 1 : TB2 overflow is selected (2) 1 0 : TA2 overflow is selected (2) 1 1 : TA4 overflow is selected (2)	RW
TA3TGH			RW
TA4TGL	Timer A4 event/trigger select bit	b7 b6 0 0 : Input on TA4IN is selected (1) 0 1 : TB2 overflow is selected (2) 1 0 : TA3 overflow is selected (2) 1 1 : TA0 overflow is selected (2)	RW
TA4TGH			RW

NOTES:
1. Make sure the port direction bits for the TA1IN to TA4IN pins are set to “0” (= input mode).
2. Overflow or underflow.

Clock prescaler reset flag

b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	After reset
CPSRF	0381₁₆	0XXXXXXX₂

Bit symbol	Bit name	Function	RW
— (b6-b0)	Nothing is assigned. When write, set to “0”. When read, their contents are indeterminate.		—
CPSR	Clock prescaler reset flag	Setting this bit to “1” initializes the prescaler for the timekeeping clock. (When read, its content is “0”.)	RW

Figure 12.1.4. ONSF Register, TRGSR Register, and CPSRF Register

Downloaded from Arrow.com.

12.1.1. Timer Mode

In timer mode, the timer counts a count source generated internally (see Table 12.1.1.1). Figure 1.2.1.1.1 shows TAiMR register in timer mode.

Table 12.1.1.1. Specifications in Timer Mode

Item	Specification
Count source	f_1, f_2, f_8, f_{32}, f_{C32}
Count operation	• Down-count • When the timer underflows, it reloads the reload register contents and continues counting
Divide ratio	1/(n+1) n: set value of TAi register (i= 0 to 4) 0000_{16} to $FFFF_{16}$
Count start condition	Set TAiS bit in the TABSR register to “1” (= start counting)
Count stop condition	Set TAiS bit to “0” (= stop counting)
Interrupt request generation timing	Timer underflow
TAiIN pin function	I/O port or gate input
TAiOUT pin function	I/O port or pulse output
Read from timer	Count value can be read by reading TAi register
Write to timer	• When not counting and until the 1st count source is input after counting start Value written to TAi register is written to both reload register and counter • When counting (after 1st count source input) Value written to TAi register is written to only reload register (Transferred to counter when reloaded next)
Select function	• Gate function Counting can be started and stopped by an input signal to TAiIN pin • Pulse output function Whenever the timer underflows, the output polarity of TAiOUT pin is inverted. When not counting, the pin outputs a low.

Timer Ai mode register (i=0 to 4)

b7	b6	b5	b4	b3	b2	b1	b0
		0				0	0

Symbol	Address	After reset
TA0MR to TA4MR	0396_{16} to $039A_{16}$	00_{16}

Bit symbol	Bit name	Function	RW
TMOD0	Operation mode select bit	b1 b0 0 0 : Timer mode	RW
TMOD1			RW
MR0	Pulse output function select bit	0 : Pulse is not output (TAiOUT pin is a normal port pin) 1 : Pulse is output (TAiOUT pin is a pulse output pin)	RW
MR1	Gate function select bit	b4 b3 0 0 : 0 1 : } Gate function not available (TAiIN pin functions as I/O port) 1 0 : Counts while input on the TAiIN pin is low (1) 1 1 : Counts while input on the TAiIN pin is high (1)	RW
MR2			RW
MR3	Must be set to “0” in timer mode		RW
TCK0	Count source select bit	b7 b6 0 0 : f_1 or f_2 0 1 : f_8 1 0 : f_{32} 1 1 : f_{C32}	RW
TCK1			RW

NOTE:
1.The port direction bit for the TAiIN pin must be set to “0” (= input mode).

Figure 12.1.1.1. Timer Ai Mode Register in Timer Mode

Downloaded from Arrow.com.

12.1.2. Event Counter Mode

In event counter mode, the timer counts pulses from an external device or overflows and underflows of other timers. Timers A2, A3 and A4 can count two-phase external signals. Table 12.1.2.1 lists specifications in event counter mode (when not processing two-phase pulse signal). Table 12.1.2.2 lists specifications in event counter mode (when processing two-phase pulse signal with the timers A2, A3 and A4). Figure 12.1.2.1 shows TAiMR register in event counter mode (when not processing two-phase pulse signal). Figure 12.1.2.2 shows TA2MR to TA4MR registers in event counter mode (when processing two-phase pulse signal with the timers A2, A3 and A4).

Table 12.1.2.1. Specifications in Event Counter Mode (when not processing two-phase pulse signal)

Item	Specification
Count source	• External signals input to TAiIN pin (i=0 to 4) (effective edge can be selected in program) • Timer B2 overflows or underflows, timer Aj (j=i-1, except j=4 if i=0) overflows or underflows, timer Ak (k=i+1, except k=0 if i=4) overflows or underflows
Count operation	• Up-count or down-count can be selected by external signal or program • When the timer overflows or underflows, it reloads the reload register contents and continues counting. When operating in free-running mode, the timer continues counting without reloading.
Divided ratio	1/ (FFFF16 - n + 1) for up-count 1/ (n + 1) for down-count n : set value of TAi register 000016 to FFFF16
Count start condition	Set TAiS bit in the TABSR register to “1” (= start counting)
Count stop condition	Set TAiS bit to “0” (= stop counting)
Interrupt request generation timing	Timer overflow or underflow
TAiIN pin function	I/O port or count source input
TAiOUT pin function	I/O port, pulse output, or up/down-count select input
Read from timer	Count value can be read by reading TAi register
Write to timer	• When not counting and until the 1st count source is input after counting start Value written to TAi register is written to both reload register and counter • When counting (after 1st count source input) Value written to TAi register is written to only reload register (Transferred to counter when reloaded next)
Select function	• Free-run count function Even when the timer overflows or underflows, the reload register content is not reloaded to it • Pulse output function Whenever the timer underflows or underflows, the output polarity of TAiOUT pin is inverted . When not counting, the pin outputs a low.

Downloaded from Arrow.com.

Timer Ai mode register (i=0 to 4)
(When not using two-phase pulse signal processing)

b7	b6	b5	b4	b3	b2	b1	b0
		0				0	1

Symbol	Address	After reset
TA0MR to TA4MR	039616 to 039A16	0016

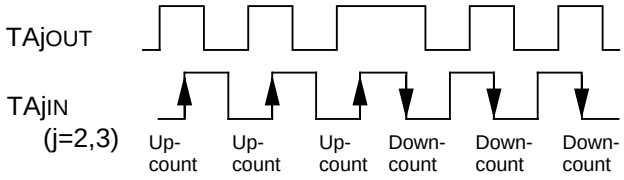
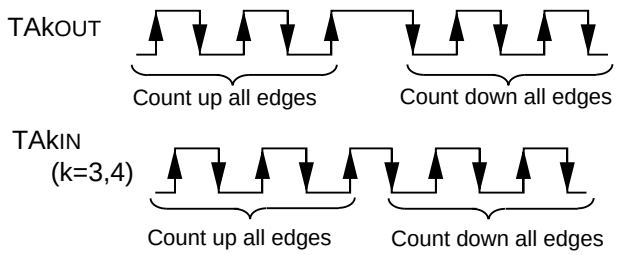
Bit symbol	Bit name	Function	RW
TMOD0	Operation mode select bit	b1 b0 0 1 : Event counter mode (1)	RW
TMOD1			RW
MR0	Pulse output function select bit	0 : Pulse is not output (TAiOUT pin functions as I/O port) 1 : Pulse is output (TAiOUT pin functions as pulse output pin)	RW
MR1	Count polarity select bit (2)	0 : Counts external signal's falling edge 1 : Counts external signal's rising edge	RW
MR2	Up/down switching cause select bit	0 : UDF register 1 : Input signal to TAiOUT pin (3)	RW
MR3	Must be set to "0" in event counter mode		RW
TCK0	Count operation type select bit	0 : Reload type 1 : Free-run type	RW
TCK1	Can be "0" or "1" when not using two-phase pulse signal processing		RW

NOTES:
1. During event counter mode, the count source can be selected using the ONSF and TRGSR registers.
2. Effective when the TAiTGH and TAiTGL bits in the ONSF or TRGSR register are '002' (TAiIN pin input).
3. Count down when input on TAiOUT pin is low or count up when input on that pin is high. The port direction bit for TAiOUT pin must be set to "0" (= input mode).

Figure 12.1.2.1. TAiMR Register in Event Counter Mode (when not using two-phase pulse signal processing)

Downloaded from Arrow.com.

Table 12.1.2.2. Specifications in Event Counter Mode (when processing two-phase pulse signal with timers A2, A3 and A4)

Item	Specification
Count source	• Two-phase pulse signals input to TAiIN or TAiOUT pins (i = 2 to 4)
Count operation	• Up-count or down-count can be selected by two-phase pulse signal • When the timer overflows or underflows, it reloads the reload register contents and continues counting. When operating in free-running mode, the timer continues counting without reloading.
Divide ratio	1/ (FFFF16 - n + 1) for up-count 1/ (n + 1) for down-count n : set value of TAi register 000016 to FFFF16
Count start condition	Set TAiS bit in the TABSR register to “1” (= start counting)
Count stop condition	Set TAiS bit to “0” (= stop counting)
Interrupt request generation timing	Timer overflow or underflow
TAiIN pin function	Two-phase pulse input
TAiOUT pin function	Two-phase pulse input
Read from timer	Count value can be read by reading timer A2, A3 or A4 register
Write to timer	• When not counting and until the 1st count source is input after counting start Value written to TAi register is written to both reload register and counter • When counting (after 1st count source input) Value written to TAi register is written to reload register (Transferred to counter when reloaded next)
Select function (1)	• Normal processing operation (timer A2 and timer A3) The timer counts up rising edges or counts down falling edges on TAjIN (j=2, 3) pin when input signals on TAjOUT pin is “H”.  • Multiply-by-4 processing operation (timer A3 and timer A4) If the phase relationship is such that TAkIN(k=3, 4) pin goes “H” when the input signal on TAkOUT pin is “H”, the timer counts up rising and falling edges on TAkOUT and TAkIN pins. If the phase relationship is such that TAkIN pin goes “L” when the input signal on TAkOUT pin is “H”, the timer counts down rising and falling edges on TAkOUT and TAkIN pins.  • Counter initialization by Z-phase input (timer A3) The timer count value is initialized to 0 by Z-phase input.

NOTE:
1. Only timer A3 is selectable. Timer A2 is fixed to normal processing operation, and timer A4 is fixed to multiply-by-4 processing operation.

Downloaded from Arrow.com.

Timer Ai mode register (i=2 to 4)
(When using two-phase pulse signal processing)

b6	b5	b4	b3	b2	b1	b0
		0	1	0	0	0	1

Symbol	Address	After reset
TA2MR to TA4MR	0398_{16} to $039A_{16}$	00_{16}

Bit symbol	Bit name	Function	RW
TMOD0	Operation mode select bit	b1 b0 0 1 : Event counter mode	RW
TMOD1			RW
MR0	To use two-phase pulse signal processing, set this bit to “0”.		RW
MR1	To use two-phase pulse signal processing, set this bit to “0”.		RW
MR2	To use two-phase pulse signal processing, set this bit to “1”.		RW
MR3	To use two-phase pulse signal processing, set this bit to “0”.		RW
TCK0	Count operation type select bit	0 : Reload type 1 : Free-run type	RW
TCK1	Two-phase pulse signal processing operation select bit (1, 2)	0 : Normal processing operation 1 : Multiply-by-4 processing operation	RW

NOTES:
1. TCK1 bit is valid for timer A3 mode register. No matter how this bit is set, Timers A2 and A4 always operate in normal processing mode and x4 processing mode, respectively.
2. If two-phase pulse signal processing is desired, following register settings are required:
   - Set the TAiP bit in the UDF register to “1” (two-phase pulse signal processing function enabled).
   - Set the TAiTGH and TAiTGL bits in the TRGSR register to ‘00_2’ (TAiIN pin input).
   - Set the port direction bits for TAiIN and TAiOUT to “0” (input mode).

Figure 12.1.2.2. TA2MR to TA4MR Registers in Event Counter Mode (when using two-phase pulse signal processing with timer A2, A3 or A4)

Downloaded from Arrow.com.

12.1.2.1 Counter Initialization by Two-Phase Pulse Signal Processing

This function initializes the timer count value to “0” by Z-phase (counter initialization) input during two-phase pulse signal processing.

This function can only be used in timer A3 event counter mode during two-phase pulse signal processing, free-running type, x4 processing, with Z-phase entered from the $\overline{INT2}$ pin.

Counter initialization by Z-phase input is enabled by writing “0000_{16}” to the TA3 register and setting the TAZIE bit in ONSF register to “1” (= Z-phase input enabled).

Counter initialization is accomplished by detecting Z-phase input edge. The active edge can be chosen to be the rising or falling edge by using the POL bit in the INT2IC register. The Z-phase pulse width applied to the $\overline{INT2}$ pin must be equal to or greater than one clock cycle of the timer A3 count source.

The counter is initialized at the next count timing after recognizing Z-phase input. Figure 12.1.2.1.1 shows the relationship between the two-phase pulse (A phase and B phase) and the Z phase.

If timer A3 overflow or underflow coincides with the counter initialization by Z-phase input, a timer A3 interrupt request is generated twice in succession. Do not use the timer A3 interrupt when using this function.

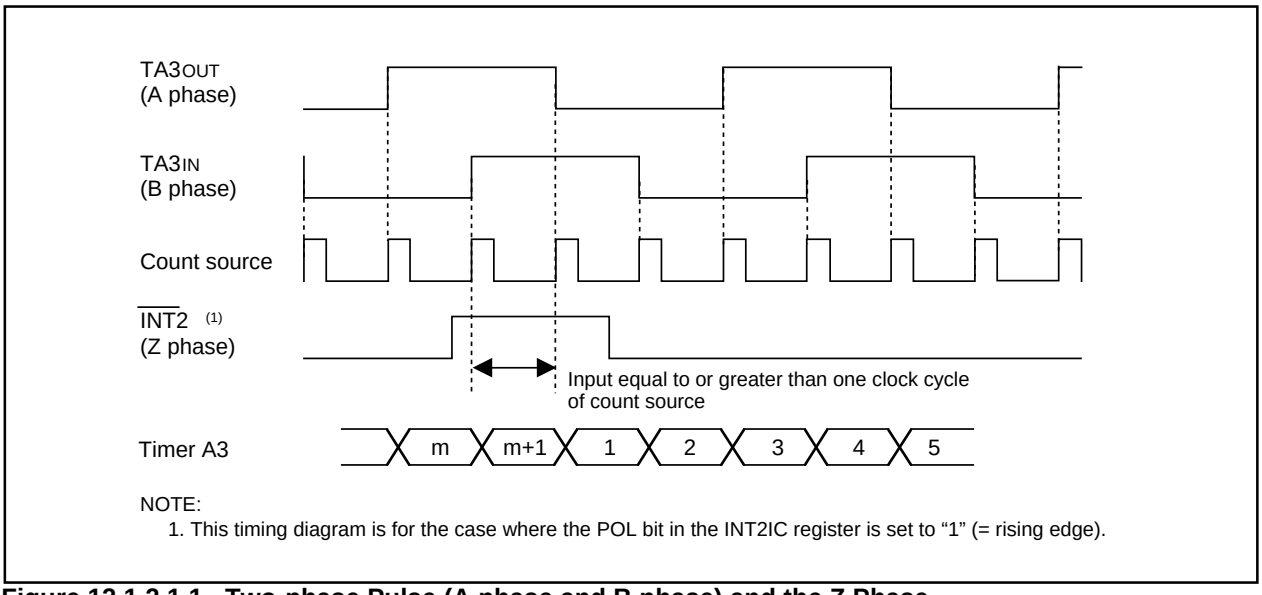


Figure 12.1.2.1.1. Two-phase Pulse (A phase and B phase) and the Z Phase

Downloaded from Arrow.com.

12.1.3. One-shot Timer Mode

In one-shot timer mode, the timer is activated only once by one trigger. (See Table 12.1.3.1.) When the trigger occurs, the timer starts up and continues operating for a given period. Figure 12.1.3.1 shows the TAiMR register in one-shot timer mode.

Table 12.1.3.1. Specifications in One-shot Timer Mode

Item	Specification
Count source	f_1, f_2, f_8, f_{32}, f_{C32}
Count operation	• Down-count • When the counter reaches 0000_{16}, it stops counting after reloading a new value • If a trigger occurs when counting, the timer reloads a new count and restarts counting
Divide ratio	1/n n : set value of TAi register 0000_{16} to $FFFF_{16}$ However, the counter does not work if the divide-by-n value is set to 0000_{16}.
Count start condition	TAiS bit in the TABSR register is set to “1” (start counting) and one of the following triggers occurs. • External trigger input from the TAi_{IN} pin • Timer B2 overflow or underflow, timer Aj (j=i-1, except j=4 if i=0) overflow or underflow, timer Ak (k=i+1, except k=0 if i=4) overflow or underflow • The TAiOS bit in the ONSF register is set to “1” (= timer starts)
Count stop condition	• When the counter is reloaded after reaching “0000_{16}” • TAiS bit is set to “0” (= stop counting)
Interrupt request generation timing	When the counter reaches “0000_{16}”
TAi_{IN} pin function	I/O port or trigger input
TAi_{OUT} pin function	I/O port or pulse output
Read from timer	An indeterminate value is read by reading TAi register
Write to timer	• When not counting and until the 1st count source is input after counting start Value written to TAi register is written to both reload register and counter • When counting (after 1st count source input) Value written to TAi register is written to only reload register (Transferred to counter when reloaded next)
Select function	• Pulse output function The timer outputs a low when not counting and a high when counting.

Downloaded from Arrow.com.

Timer Ai mode register (i=0 to 4)

b7	b6	b5	b4	b3	b2	b1	b0
		0				1	0

Symbol	Address	After reset
TA0MR to TA4MR	396_{16} to $039A_{16}$	00_{16}

Bit symbol	Bit name	Function	RW
TMOD0	Operation mode select bit	b1 b0 1 0 : One-shot timer mode	RW
TMOD1			RW
MR0	Pulse output function select bit	0 : Pulse is not output (TAiOUT pin functions as I/O port) 1 : Pulse is output (TAiOUT pin functions as a pulse output pin)	RW
MR1	External trigger select bit (1)	0 : Falling edge of input signal to TAiIN pin (2) 1 : Rising edge of input signal to TAiIN pin (2)	RW
MR2	Trigger select bit	0 : TAiOS bit is enabled 1 : Selected by TAiTGH to TAiTGL bits	RW
MR3	Must be set to “0” in one-shot timer mode		RW
TCK0	Count source select bit	b7 b6 0 0 : f_1 or f_2 0 1 : f_8 1 0 : f_{32} 1 1 : fC_{32}	RW
TCK1			RW

NOTES:
1. Effective when the TAiTGH and TAiTGL bits in the ONSF or TRGSR register are ‘00_2’ (TAiIN pin input).
2. The port direction bit for the TAiIN pin must be set to “0” (= input mode).

Figure 12.1.3.1. TAiMR Register in One-shot Timer Mode

Downloaded from Arrow.com.

12.1.4. Pulse Width Modulation (PWM) Mode

In PWM mode, the timer outputs pulses of a given width in succession (see Table 12.1.4.1). The counter functions as either 16-bit pulse width modulator or 8-bit pulse width modulator. Figure 12.1.4.1 shows TAiMR register in pulse width modulation mode. Figures 12.1.4.2 and 12.1.4.3 show examples of how a 16-bit pulse width modulator operates and how an 8-bit pulse width modulator operates.

Table 12.1.4.1. Specifications in Pulse Width Modulation Mode

Item	Specification
Count source	f_1, f_2, f_8, f_{32}, f_{C32}
Count operation	• Down-count (operating as an 8-bit or a 16-bit pulse width modulator) • The timer reloads a new value at a rising edge of PWM pulse and continues counting • The timer is not affected by a trigger that occurs during counting
16-bit PWM	• High level width n / fj n : set value of TAi register (i=o to 4) • Cycle time $(2^{16}-1)$ / fj fixed fj: count source frequency (f_1, f_2, f_8, f_{32}, f_{C32})
8-bit PWM	• High level width n x (m+1) / fj n : set value of TAi register high-order address • Cycle time (2^8-1) x (m+1) / fj m : set value of TAi register low-order address
Count start condition	• TAiS bit in theTABSR register is set to “1” (= start counting) • The TAiS bit is set to "1" and external trigger input from the TAiIN pin • The TAiS bit is set to "1" and one of the following external triggers occurs Timer B2 overflow or underflow, timer Aj (j=i-1, except j=4 if i=0) overflow or underflow, timer Ak (k=i+1, except k=0 if i=4) overflow or underflow
Count stop condition	TAiS bit is set to “0” (= stop counting)
Interrupt request generation timing	PWM pulse goes “L”
TAiIN pin function	I/O port or trigger input
TAiOUT pin function	Pulse output
Read from timer	An indeterminate value is read by reading TAi register
Write to timer	• When not counting and until the 1st count source is input after counting start Value written to TAi register is written to both reload register and counter • When counting (after 1st count source input) Value written to TAi register is written to only reload register (Transferred to counter when reloaded next)

Downloaded from Arrow.com.

Timer Ai mode register (i= 0 to 4)

b7	b6	b5	b4	b3	b2	b1	b0
					1	1	1

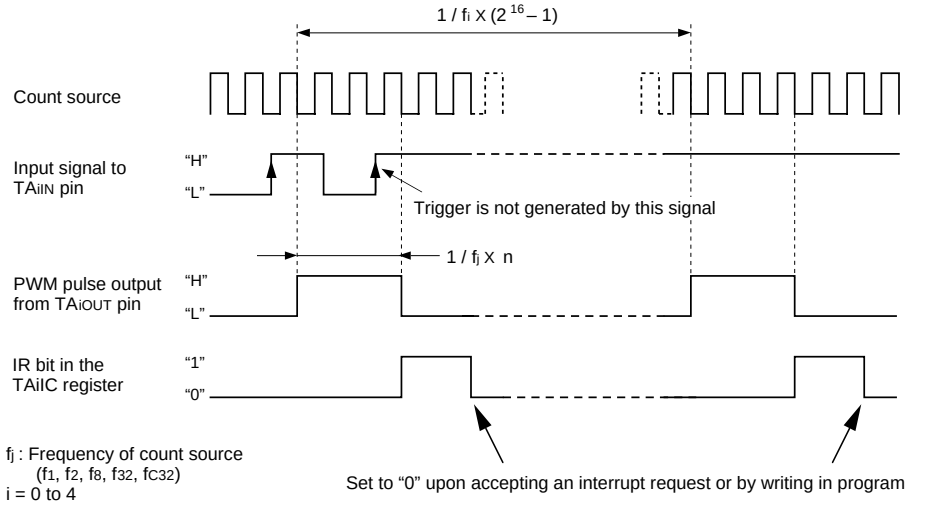
Symbol	Address	After reset
TA0MR to TA4MR	0396_{16} to $039A_{16}$	00_{16}

Bit symbol	Bit name	Function	RW
TMOD0	Operation mode select bit	b1 b0 1 1 : PWM mode	RW
TMOD1			RW
MR0	Pulse output funcion select bit	0: Pulse is not output (TAiOUT pin functions as I/O port) 1: Pulse is output (TAiOUT pin functions as a pulse output pin)	RW
MR1	External trigger select bit (1)	0: Falling edge of input signal to TAiIN pin(2) 1: Rising edge of input signal to TAiIN pin(2)	RW
MR2	Trigger select bit	0 : Write “1” to TAiS bit in the TASF register 1 : Selected by TAiTGH to TAiTGL bits	RW
MR3	16/8-bit PWM mode select bit	0: Functions as a 16-bit pulse width modulator 1: Functions as an 8-bit pulse width modulator	RW
TCK0	Count source select bit	b7 b6 0 0 : f_1 or f_2 0 1 : f_8 1 0 : f_{32} 1 1 : f_{C32}	RW
TCK1			RW

NOTES:
1. Effective when the TAiTGH and TAiTGL bits in the ONSF or TRGSR register are ‘00_2’ (TAiIN pin input).
2. The port direction bit for the TAiIN pin must be set to “0” (= input mode).

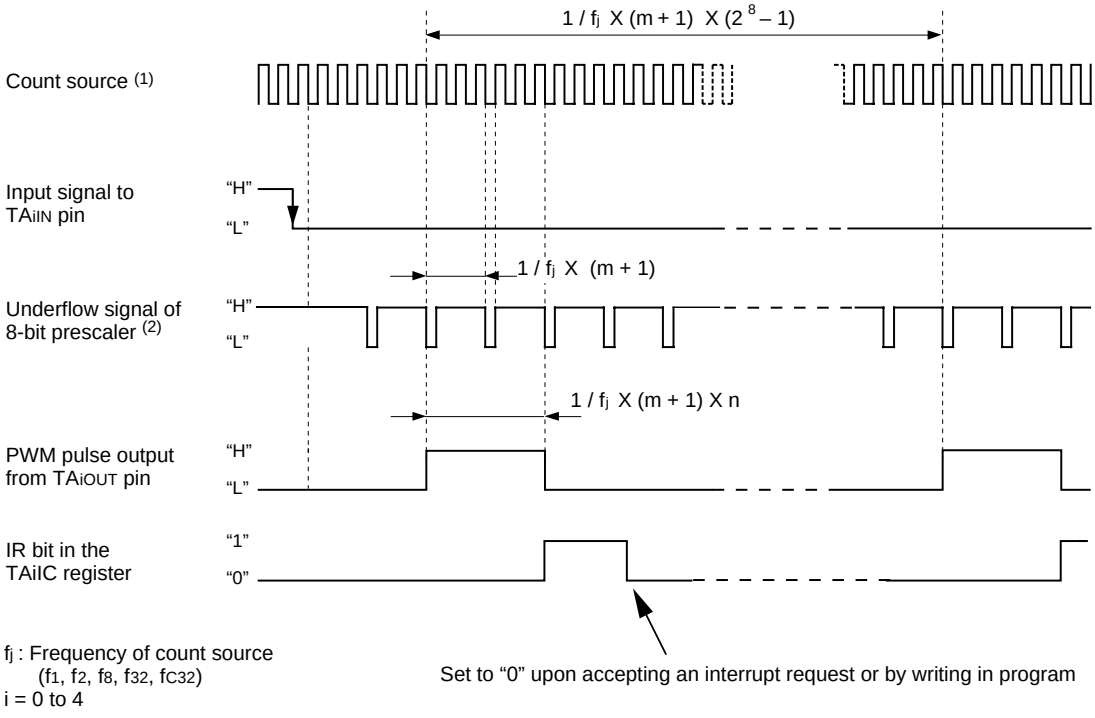
Figure 12.1.4.1. TAiMR Register in Pulse Width Modulation Mode

Downloaded from Arrow.com.



NOTES:
1. n = 0000$_{16}$ to FFFE$_{16}$.
2. This timing diagram is for the case where the TAi register is set to "0003$_{16}$", the TAiTGH and TAiTGL bits in the ONSF or TRGSR register is set to "00$_2$" (TAiIN pin input), the MR1 bit in the TAiMR register is set to "1" (rising edge), and the MR2 bit in the TAiMR register is set to "1" (trigger selected by TAiTGH and TAiTGL bits).

Figure 12.1.4.2. Example of 16-bit Pulse Width Modulator Operation



NOTES:
1. The 8-bit prescaler counts the count source.
2. The 8-bit pulse width modulator counts the 8-bit prescaler's underflow signal.
3. m = 00$_{16}$ to FF$_{16}$; n = 00$_{16}$ to FE$_{16}$.
4. This timing diagram is for the case where the TAi register is set to "0202$_{16}$", the TAiTGH and TAiTGL bits in the ONSF or TRGSR register is set to "00$_2$" (TAiIN pin input), the MR1 bit in the TAiMR register is set to "0"(falling edge), and the MR2 bit in the TAiMR register is set to "1" (trigger selected by TAiTGH and TAiTGL bits).

Figure 12.1.4.3. Example of 8-bit Pulse Width Modulator Operation

Downloaded from Arrow.com.

12.2 Timer B

Note

The TB2IN pin for Timer B2 is not available in 42-pin package.

[Precautions when using Timer B2]

- Event Counter Mode The external input signals cannot be counted. Set the TCK1 bit in the TB2MR register to “1” when using the Event Count Mode.
- Pulse Period/Pulse Width Measurement Mode
  This mode connot be used.

Figure 12.2.1 shows a block diagram of the timer B. Figures 12.2.2 and 12.2.3 show registers related to the timer B.

Timer B supports the following four modes. Use the TMOD1 and TMOD0 bits in the TBiMR register (i = 0 to 2) to select the desired mode.

- Timer mode: The timer counts an internal count source.
- Event counter mode: The timer counts pulses from an external device or overflows or underflows of other timers.
- Pulse period/pulse width measuring mode: The timer measures an external signal's pulse period or pulse width.
- A/D trigger mode: The timer starts counting by one trigger until the count value becomes 0000_{16}. This mode is used together with simultaneous sample sweep mode or delayed trigger mode 0 of A/D converter to start A/D conversion.

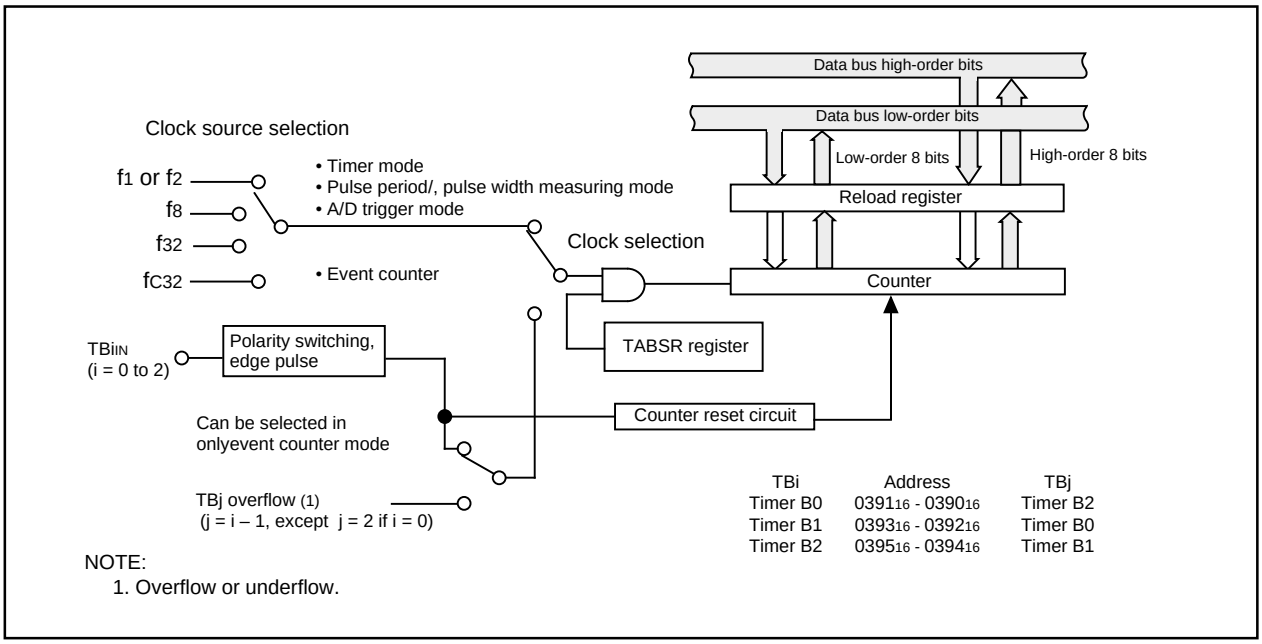
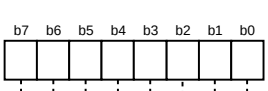


Figure 12.2.1. Timer B Block Diagram

Downloaded from Arrow.com.

Timer Bi mode register (i=0 to 2)



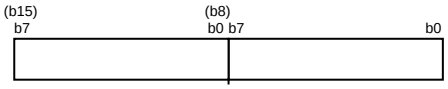
Symbol	Address	After reset
TB0MR to TB2MR	$039B_{16}$ to $039D_{16}$	$00XX0000_2$

Bit symbol	Bit name	Function	RW
TMOD0	Operation mode select bit	b1 b0 0 0 : Timer mode or A/D trigger mode 0 1 : Event counter mode 1 0 : Pulse period measurement mode, pulse width measurement mode 1 1 : Must not be set	RW
TMOD1			RW
MR0	———	Function varies with each operation mode	RW
MR1			RW
MR2			RW (1) — (2)
MR3			RO
TCK0	Count source select bit	Function varies with each operation mode	RW
TCK1			RW

Figure 12.2.2. TB0MR to TB2MR Registers

Downloaded from Arrow.com.

Timer Bi register (i=0 to 2)(1)

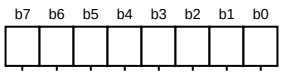


Symbol	Address	After reset
TB0	0391_{16}, 0390_{16}	Indeterminate
TB1	0393_{16}, 0392_{16}	Indeterminate
TB2	0395_{16}, 0394_{16}	Indeterminate

Mode	Function	Setting range	RW
Timer mode	Divide the count source by n + 1 where n = set value	0000_{16} to $FFFF_{16}$	RW
Event counter mode	Divide the count source by n + 1 where n = set value (2)	0000_{16} to $FFFF_{16}$	RW
Pulse period modulation mode, Pulse width modulation mode	Measures a pulse period or width	——	RO
A/D trigger mode (3)	Divide the count source by n + 1 where n = set value and cause the timer stop	0000_{16} to $FFFF_{16}$	RW

NOTES:
1. The register must be accessed in 16 bit units.
2. The timer counts pulses from an external device or overflows or underflows of other timers.
3. When this mode is used combining delayed trigger mode 0, set the larger value than the value of the timer B0 register to the timer B1 register.

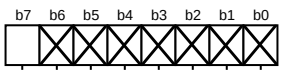
Count start flag



Symbol	Address	After reset
TABSR	0380_{16}	00_{16}

Bit symbol	Bit name	Function	RW
TA0S	Timer A0 count start flag	0 : Stops counting 1 : Starts counting	RW
TA1S	Timer A1 count start flag		RW
TA2S	Timer A2 count start flag		RW
TA3S	Timer A3 count start flag		RW
TA4S	Timer A4 count start flag		RW
TB0S	Timer B0 count start flag		RW
TB1S	Timer B1 count start flag		RW
TB2S	Timer B2 count start flag		RW

Clock prescaler reset flag



Symbol	Address	After reset
CPSRF	0381_{16}	$0XXXXXXX_2$

Bit symbol	Bit name	Function	RW
—— (b6-b0)	Nothing is assigned. When write, set to “0”. When read, their contents are indeterminate.		—
CPSR	Clock prescaler reset flag	Setting this bit to “1” initializes the prescaler for the timekeeping clock. (When read, the value of this bit is “0”.)	RW

Figure 12.2.3. TB0 to TB2 Registers, TABSR Register, CPSRF Register

Downloaded from Arrow.com.

12.2.1 Timer Mode

In timer mode, the timer counts a count source generated internally (see Table 12.2.1.1). Figure 12.2.1.1 shows TBiMR register in timer mode.

Table 12.2.1.1 Specifications in Timer Mode

Item	Specification
Count source	f_1, f_2, f_8, f_{32}, f_{C32}
Count operation	• Down-count • When the timer underflows, it reloads the reload register contents and continues counting
Divide ratio	1/(n+1) n: set value of TBi register (i= 0 to 2) 0000_{16} to $FFFF_{16}$
Count start condition	Set TBiS bit(1) to “1” (= start counting)
Count stop condition	Set TBiS bit to “0” (= stop counting)
Interrupt request generation timing	Timer underflow
TBi_{IN} pin function	I/O port
Read from timer	Count value can be read by reading TBi register
Write to timer	• When not counting and until the 1st count source is input after counting start Value written to TBi register is written to both reload register and counter • When counting (after 1st count source input) Value written to TBi register is written to only reload register (Transferred to counter when reloaded next)

NOTE:

1. The TB0S to TB2S bits are assigned to the bit 5 to bit 7 in the TABSR register.

Timer Bi mode register (i= 0 to 2)

b7	b6	b5	b4	b3	b2	b1	b0
						0	0

Symbol	Address	After reset
TB0MR to TB2MR	$039B_{16}$ to $039D_{16}$	$00XX0000_2$

Bit symbol	Bit name	Function	RW
TMOD0	Operation mode select bit	b1 b0 0 0 : Timer mode or A/D trigger mode	RW
TMOD1			RW
MR0	Has no effect in timer mode Can be set to “0” or “1”		RW
MR1			RW
MR2	TB0MR register Must be set to “0” in timer mode		RW
	TB1MR, TB2MR registers Nothing is assigned. When write, set to “0”. When read, its content is indeterminate		—
MR3	When write in timer mode, set to “0”. When read in timer mode, its content is indeterminate.		RO
TCK0	Count source select bit	b7 b6 0 0 : f_1 or f_2 0 1 : f_8 1 0 : f_{32} 1 1 : f_{C32}	RW
TCK1			RW

Figure 12.2.1.1 TBiMR Register in Timer Mode

Downloaded from Arrow.com.

12.2.2 Event Counter Mode

In event counter mode, the timer counts pulses from an external device or overflows and underflows of other timers (see Table 12.2.2.1) . Figure 12.2.2.1 shows TBiMR register in event counter mode.

Table 12.2.2.1 Specifications in Event Counter Mode

Item	Specification
Count source	• External signals input to TBiIN pin (i=0 to 2) (effective edge can be selected in program) • Timer Bj overflow or underflow (j=i-1, except j=2 if i=0)
Count operation	• Down-count • When the timer underflows, it reloads the reload register contents and continues counting
Divide ratio	1/(n+1) n: set value of TBi register 0000$_{16}$ to FFFF$_{16}$
Count start condition	Set TBiS bit(1) to “1” (= start counting)
Count stop condition	Set TBiS bit to “0” (= stop counting)
Interrupt request generation timing	Timer underflow
TBiIN pin function	Count source input
Read from timer	Count value can be read by reading TBi register
Write to timer	• When not counting and until the 1st count source is input after counting start Value written to TBi register is written to both reload register and counter • When counting (after 1st count source input) Value written to TBi register is written to only reload register (Transferred to counter when reloaded next)

NOTE:
1. The TB0S to TB2S bits are assigned to the bit 5 to bit 7 in the TABSR register.

Timer Bi mode register (i=0 to 2)

b7	b6	b5	b4	b3	b2	b1	b0
						0	1

Symbol: TB0MR to TB2MR — Address: 039B$_{16}$ to 039D$_{16}$ — After reset: 00XX0000$_2$

Bit symbol	Bit name	Function	RW
TMOD0	Operation mode select bit	b1 b0 0 1 : Event counter mode	RW
TMOD1			RW
MR0	Count polarity select bit (1)	b3 b2 0 0 : Counts external signal's falling edges 0 1 : Counts external signal's rising edges 1 0 : Counts external signal's falling and rising edges 1 1 : Must not be set	RW
MR1			RW
MR2	TB0MR register Must be set to “0” in timer mode		RW
	TB1MR, TB2MR registers Nothing is assigned. When write, set to “0”. When read, its content is indeterminate.		—
MR3	When write in event counter mode, set to “0”. When read in event counter mode, its content is indeterminate.		RO
TCK0	Has no effect in event counter mode. Can be set to “0” or “1”.		RW
TCK1	Event clock select	0 : Input from TBiIN pin (2) 1 : TBj overflow or underflow (j = i – 1, except j = 2 if i = 0)	RW

NOTES:
1. Effective when the TCK1 bit is set to “0” (input from TBiIN pin). If the TCK1 bit is set to “1” (TBj overflow or underflow), these bits can be set to “0” or “1”.
2. The port direction bit for the TBiIN pin must be set to “0” (= input mode).

Figure 12.2.2.1 TBiMR Register in Event Counter Mode

Downloaded from Arrow.com.

12.2.3 Pulse Period and Pulse Width Measurement Mode

In pulse period and pulse width measurement mode, the timer measures pulse period or pulse width of an external signal (see Table 12.2.3.1). Figure 12.2.3.1 shows TBiMR register in pulse period and pulse width measurement mode. Figure 12.2.3.2 shows the operation timing when measuring a pulse period. Figure 12.2.3.3 shows the operation timing when measuring a pulse width.

Table 12.2.3.1 Specifications in Pulse Period and Pulse Width Measurement Mode

Item	Specification
Count source	f_1, f_2, f_8, f_{32}, f_{C32}
Count operation	• Up-count • Counter value is transferred to reload register at an effective edge of measurement pulse. The counter value is set to "0000_{16}" to continue counting.
Count start condition	Set TBiS (i=0 to 2) bit(3) to "1" (= start counting)
Count stop condition	Set TBiS bit to "0" (= stop counting)
Interrupt request generation timing	• When an effective edge of measurement pulse is input(1) • Timer overflow. When an overflow occurs, MR3 bit in the TBiMR register is set to "1" (overflowed) simultaneously. MR3 bit is cleared to "0" (no overflow) by writing to TBiMR register at the next count timing or later after MR3 bit was set to "1". At this time, make sure TBiS bit is set to "1" (start counting).
TBi_{IN} pin function	Measurement pulse input
Read from timer	Contents of the reload register (measurement result) can be read by reading TBi register(2)
Write to timer	Value written to TBi register is written to neither reload register nor counter

NOTES:
1. Interrupt request is not generated when the first effective edge is input after the timer started counting.
2. Value read from TBi register is indeterminate until the second valid edge is input after the timer starts counting.
3. The TB0S to TB2S bits are assigned to the bit 5 to bit 7 in the TABSR register.

Timer Bi mode register (i=0 to 2)

b7	b6	b5	b4	b3	b2	b1	b0
						1	0

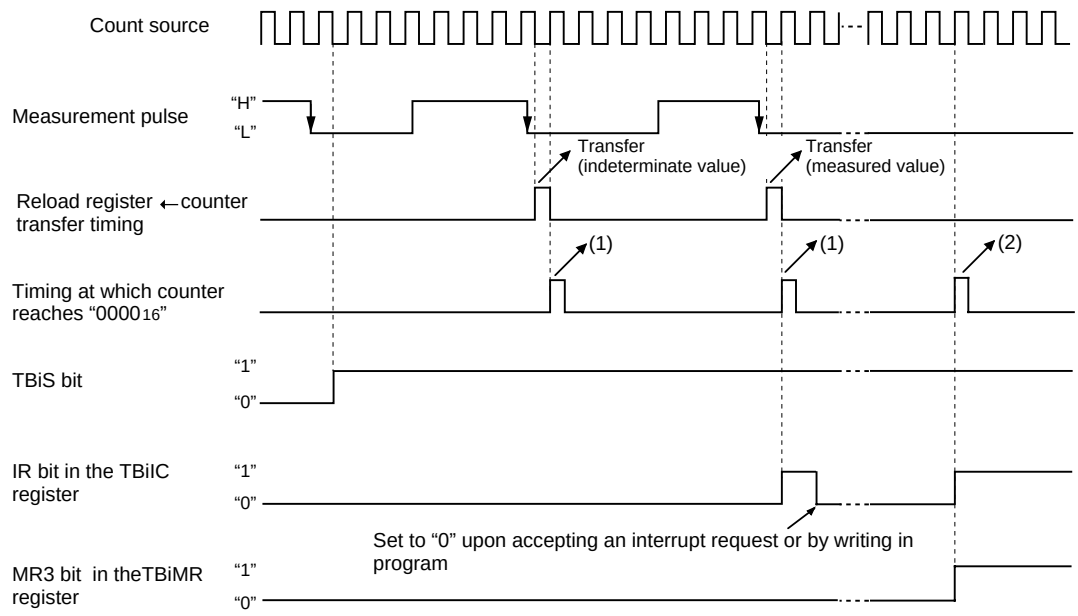
Symbol: TB0MR to TB2MR — Address: $039B_{16}$ to $039D_{16}$ — After reset: $00XX0000_2$

Bit symbol	Bit name	Function	RW
TMOD0	Operation mode select bit	b1 b0 1 0 : Pulse period / pulse width measurement mode	RW
TMOD1			RW
MR0	Measurement mode select bit	b3 b2 0 0 : Pulse period measurement (Measurement between a falling edge and the next falling edge of measured pulse) 0 1 : Pulse period measurement (Measurement between a rising edge and the next rising edge of measured pulse) 1 0 : Pulse width measurement (Measurement between a falling edge and the next rising edge of measured pulse and between a rising edge and the next falling edge) 1 1 : Must not be set.	RW
MR1			RW
MR2	TB0MR register Must be set to "0" in pulse period and pulse width measurement mode		RW
	TB1MR, TB2MR registers Nothing is assigned. When write, set to "0". When read, its content turns out to be indeterminate.		—
MR3	Timer Bi overflow flag (1)	0 : Timer did not overflow 1 : Timer has overflowed	RO
TCK0	Count source select bit	b7 b6 0 0 : f_1 or f_2 0 1 : f_8 1 0 : f_{32} 1 1 : f_{C32}	RW
TCK1			RW

NOTE:
1. This flag is indeterminate after reset. When the TBiS bit is set to "1" (start counting), the MR3 bit is cleared to "0" (no overflow) by writing to the TBiMR register at the next count timing or later after the MR3 bit was set to "1" (overflowed). The MR3 bit cannot be set to "1" in a program. The TB0S to TB2S bits are assigned to the bit 5 to bit 7 in the TABSR register.

Figure 12.2.3.1 TBiMR Register in Pulse Period and Pulse Width Measurement Mode

Downloaded from Arrow.com.

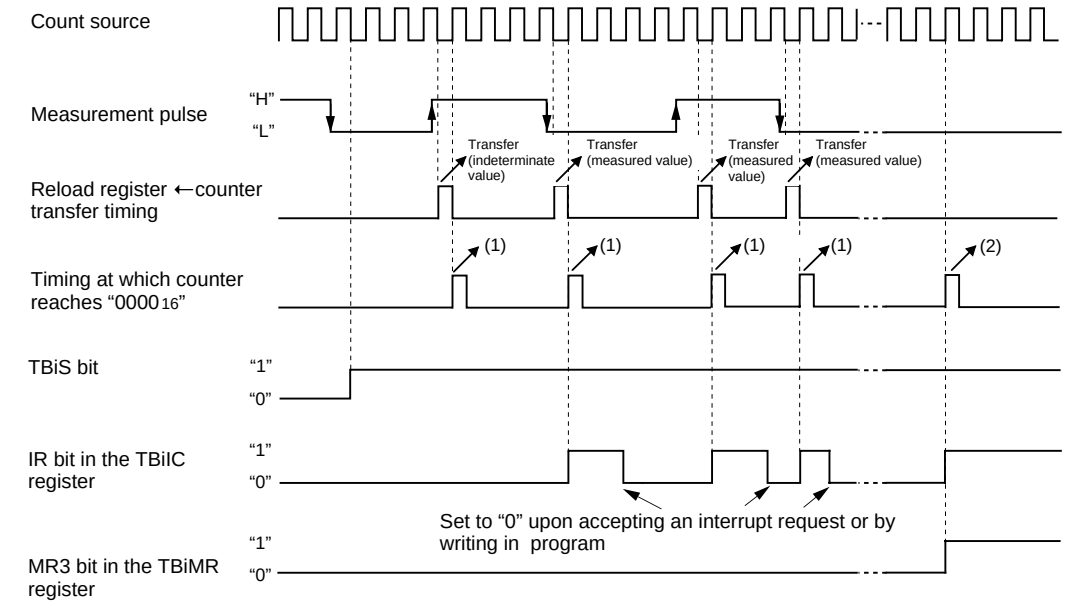


The TB0S to TB2S bits are assigned to the bit 5 to bit 7 in the TABSR register.

i = 0 to 2

NOTES:
1. Counter is initialized at completion of measurement.
2. Timer has overflowed.
3. This timing diagram is for the case where the MR1 to MR0 bits in the TBiMR register are "00$_2$" (measure the interval from falling edge to falling edge of the measurement pulse).

Figure 12.2.3.2 Operation timing when measuring a pulse period



The TB0S to TB2S bits are assigned to the bit 5 to bit 7 in the TABSR register.

i = 0 to 2

NOTES:
1. Counter is initialized at completion of measurement.
2. Timer has overflowed.
3. This timing diagram is for the case where the MR1 to MR0 bits in the TBiMR register are "10$_2$" (measure the interval from a falling edge to the next rising edge and the interval from a rising edge to the next falling edge of the measurement pulse).

Figure 12.2.3.3 Operation timing when measuring a pulse width

Downloaded from Arrow.com.

12.2.4 A/D Trigger Mode

A/D trigger mode is used as conversion start trigger for A/D converter in simultaneous sample sweep mode of A/D conversion or delayed trigger mode 0. This mode is used as conversion start trigger of A/D converter. A/D trigger mode is used in Timer B0 and Timer B1. In this mode, the timer starts counting by one trigger until the count value becomes 0000_{16}. Figure 12.2.4.1 shows the TBiMR register in A/D trigger mode and figure 12.2.4.2 shows the TB2SC register.

Table 12.2.4.1 A/D Trigger Mode Specifications

Item	Specification
Count Source	f_1, f_2, f_8, f_{32}, and f_{C32}
Count Operation	• Down count • When the timer underflows, reload register contents are reloaded before stopping counting • When a trigger is generated during the count operation, the count is not affected
Divide Ratio	1/(n+1) n: Setting value of TBi register (i=0,1) 0000_{16}-$FFFF_{16}$
Count Start Condition	When the TBiS (i=0,1) bit in the TABSR register is "1"(count started), TBiEN (i=0,1) bit in TB2SC register is "1", and the following trigger is generated. (Selection based on TB2SEL bit in the TB2SC register) • Timer B2 overflow or underflow • Underflow of Timer B2 interrupt generation frequency counter setting
Count Stop Condition	• After the count value is 0000_{16} and reload register contents are reloaded • Set the TBiS bit to "0"(count stopped)
Interrupt Request Generation Timing	Timer underflows (1)
TBiIN Pin Function	I/O port
Read From Timer	Count value can be read by reading TBi register
Write To Timer (2)	• When writing in the TBi register during count stopped. Value is written to both reload register and counter • When writing in the TBi register during count. Value is written to only reload register (Transfered to counter when reloaded next)

NOTES:

1. A/D conversion is started by the timer underflow. For details refer to **Section 14. A/D Converter**.
2. When using in delayed trigger mode 0, set the larger value than the value of the timer B0 register to the timer B1 register.

Downloaded from Arrow.com.

Timer Bi mode register (i= 0 to 1)

b7	b6	b5	b4	b3	b2	b1	b0
						0	0

Symbol: TB0MR to TB1MR
Address: 039B16 to 039C16
After reset: 00XX00002

Bit symbol	Bit name	Function	RW
TMOD0	Operation Mode Select Bit	b1 b0 0 0 : Timer mode or A/D trigger mode	RW
TMOD1			RW
MR0	Invalid in A/D trigger mode Either "0" or "1" is enabled		RW
MR1			RW
MR2	TB0MR register Set to “0” in A/D trigger mode		RW
	TB1MR register Nothing is assigned. When write, set to “0”. When read, its content is indeterminate		—
MR3	When write in A/D trigger mode, set to “0”. When read in A/D trigger mode, its content is indeterminate.		RO
TCK0	Count Source Select Bit (1)	b7 b6 0 0 : f1 or f2 0 1 : f8 1 0 : f32 1 1 : fC32	RW
TCK1			RW

NOTE:
1. When this bit is used in delayed trigger mode 0, set the same count source to the timer B0 and timer B1.

Figure 12.2.4.1 TBiMR Register in A/D Trigger Mode

Timer B2 special mode register (1)

b7	b6	b5	b4	b3	b2	b1	b0
X	0	0					

Symbol: TB2SC
Address: 039E16
After reset: X00000002

Bit symbol	Bit name	Function	RW
PWCOM	Timer B2 Reload Timing Switch Bit (2)	0 : Timer B2 underflow 1 : Timer A output at odd-numbered	RW
IVPCR1	Three-Phase Output Port $\overline{SD}$ Control Bit 1 (3, 4, 7)	0 : Three-phase output forcible cutoff by $\overline{SD}$ pin input (high impedance) disabled 1 : Three-phase output forcible cutoff by $\overline{SD}$ pin input (high impedance) enabled	RW
TB0EN	Timer B0 Operation Mode Select Bit	0 : Other than A/D trigger mode 1 : A/D trigger mode (5)	RW
TB1EN	Timer B1 Operation Mode Select Bit	0 : Other than A/D trigger mode 1 : A/D trigger mode (5)	RW
TB2SEL	Trigger Select Bit (6)	0 : TB2 interrupt 1 : Underflow of TB2 interrupt generation frequency setting counter [ICTB2]	RW
$\overline{\text{(b6-b5)}}$	Reserved bits	Must set to "0"	RW
— (b7)	Nothing is assigned. When write, set to “0”. When read, its content is “0”.		—

NOTES:
1. Write to this register after setting the PRC1 bit in the PRCR register to "1" (write enabled).
2. If the INV11 bit is "0" (three-phase mode 0) or the INV06 bit is "1" (triangular wave modulation mode), set this bit to "0" (timer B2 underflow).
3. When setting the IVPCR1 bit to "1" (three-phase output forcible cutoff by $\overline{SD}$ pin input enabled), Set the PD8_5 bit to "0" (= input mode).
4. Related pins are U(P80), $\overline{U}$(P81), V(P72), $\overline{V}$(P73), W(P74), $\overline{W}$(P75). When a high-level ("H") signal is applied to the $\overline{SD}$ pin and set the IVPCR1 bit to 0 after forcible cutoff, pins U, $\overline{U}$, V, $\overline{V}$, W, and $\overline{W}$ are exit from the high-impedance state. If a low-level (“L”) signal is applied to the $\overline{SD}$ pin, three-phase motor control timer output will be disabled (INV03=0). At this time, when the IVPCR1 bit is 0, pins U, $\overline{U}$, V, $\overline{V}$, W, and $\overline{W}$ become programmable I/O ports. When the IVPCR1 bit is set to 1, pins U, $\overline{U}$, V, $\overline{V}$, W, and $\overline{W}$ are placed in a high-impedance state regardless of which function of those pins is used.
5. When this bit is used in delayed trigger mode 0, set the TB0EN and TB1EN bits to "1"(A/D trigger mode).
6. When setting the TB2SEL bit to "1" (underflow of TB2 interrupt generation frequency setting counter[ICTB2]), Set the INV02 bit to "1" (three-phase motor control timer function).
7. Refer to **16.6 Digital Debounce function** for $\overline{SD}$ input.

Figure 12.2.4.2 TB2SC Register

Downloaded from Arrow.com.

12.3 Three-phase Motor Control Timer Function

Timers A1, A2, A4 and B2 can be used to output three-phase motor drive waveforms. Table 12.3.1 lists the specifications of the three-phase motor control timer function. Figure 12.3.1 shows the block diagram for three-phase motor control timer function. Also, the related registers are shown on Figure 12.3.2 to Figure 12.3.8.

Table 12.3.1. Three-phase Motor Control Timer Function Specifications

Item	Specification
Three-phase waveform output pin	Six pins (U, $\overline{U}$, V, $\overline{V}$, W, $\overline{W}$)
Forced cutoff input (1)	Input "L" to $\overline{SD}$ pin
Used Timers	Timer A4, A1, A2 (used in the one-shot timer mode) Timer A4: U- and $\overline{U}$-phase waveform control Timer A1: V- and $\overline{V}$-phase waveform control Timer A2: W- and $\overline{W}$-phase waveform control Timer B2 (used in the timer mode) Carrier wave cycle control Dead timer timer (3 eight-bit timer and shared reload register) Dead time control
Output waveform	Triangular wave modulation, Sawtooth wave modification Enable to output "H" or "L" for one cycle Enable to set positive-phase level and negative-phase level respectively
Carrier wave cycle	Triangular wave modulation: count source x (m+1) x 2 Sawtooth wave modulation: count source x (m+1) m: Setting value of TB2 register, 0 to 65535 Count source: f_1, f_2, f_8, f_{32}, f_{C32}
Three-phase PWM output width	Triangular wave modulation: count source x n x 2 Sawtooth wave modulation: count source x n n: Setting value of TA4, TA1 and TA2 register (of TA4, TA41, TA1, TA11, TA2 and TA21 registers when setting the INV11 bit to "1"), 1 to 65535 Count source: f_1, f_2, f_8, f_{32}, f_{C32}
Dead time active disable function	Count source x p, or no dead time p: Setting value of DTT register, 1 to 255 Count source: f_1, f_2, f_1 divided by 2, f_2 divided by 2
Active level	Eable to select "H" or "L"
Positive and negative-phase concurrent	Positive and negative-phases concurrent active disable function Positive and negative-phases concurrent active detect function
Interrupt frequency	For Timer B2 interrupt, select a carrier wave cycle-to-cycle basis through 15 times carrier wave cycle-to-cycle basis

NOTES:

1. When the INV02 bit in the INVC0 register is set to "1" (three-phase motor control timer function), the $\overline{SD}$ function of the P8$_5$/$\overline{SD}$ pin is enabled. At this time, the P8$_5$ pin cannot be used as a programmable I/O port. When the $\overline{SD}$ function is not used, apply "H" to the P8$_5$/$\overline{SD}$ pin.
2. When the IVPCR1 bit in the TB2SC register is set to "1" (enable three-phase output forced cutoff by $\overline{SD}$ pin input), and "L" is applied to the $\overline{SD}$ pin, the related pins enter high-impedance state regardless of the functions which are used. When the IVPCR1 bit is set to "0" (disabled three-phase output forced cutoff by $\overline{SD}$ pin input) and "L" is applied to the $\overline{SD}$ pin, the related pins can be selected as a programmable I/O port and the setting of the port and port direction registers are enable.

Related pins
P7$_2$/CLK$_2$/TA1$_{OUT}$/V/RxD$_1$
P7$_3$/$\overline{CTS_2}$/$\overline{RTS_2}$/TA1$_{IN}$/$\overline{V}$/TxD$_1$
P7$_4$/TA2$_{OUT}$/W
P7$_5$/TA2$_{IN}$/$\overline{W}$
P8$_0$/TA4$_{OUT}$/U
P8$_1$/TA4$_{IN}$/$\overline{U}$

Downloaded from Arrow.com.

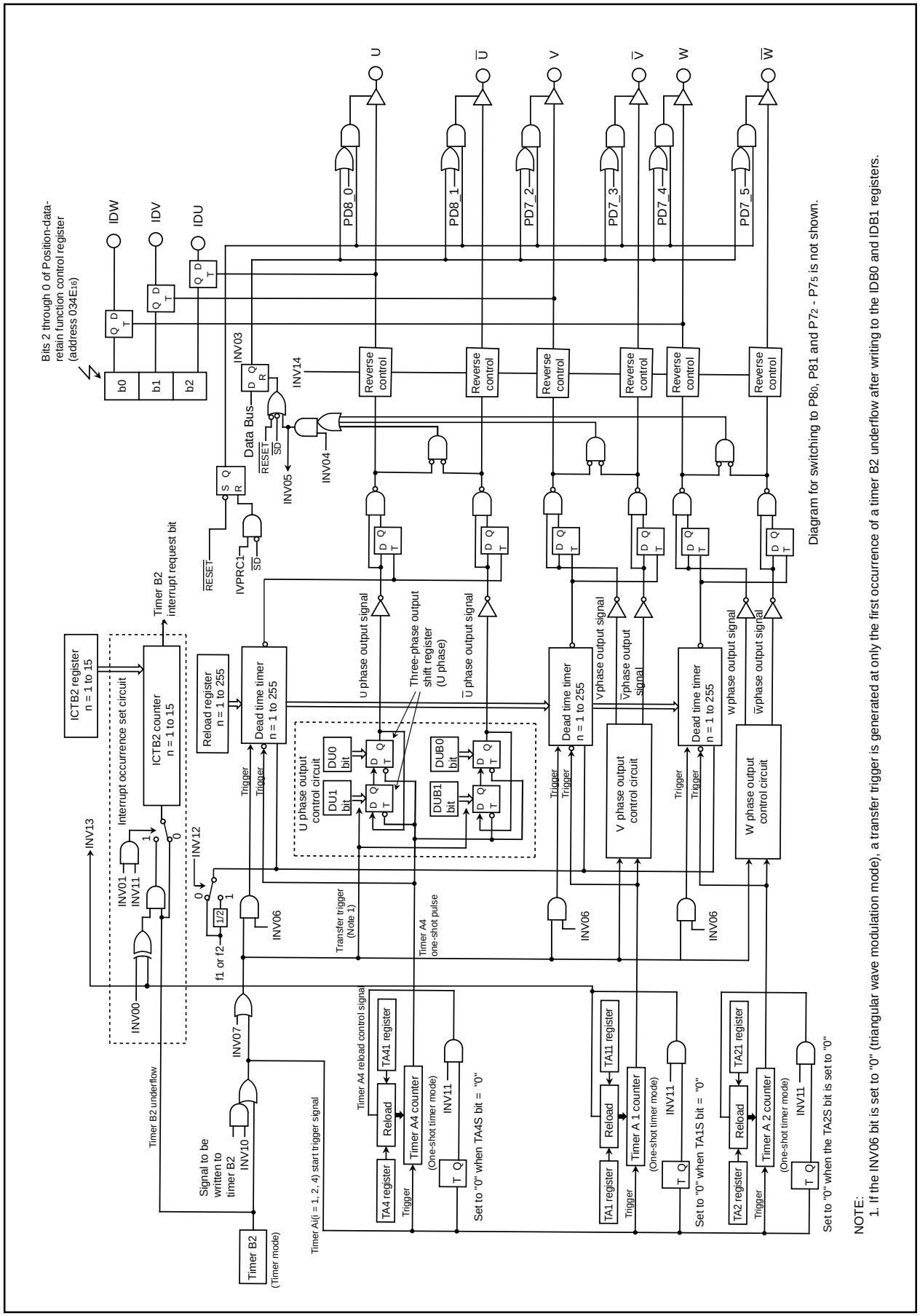


Figure 12.3.1. Three-phase Motor Control Timer Functions Block Diagram

Downloaded from Arrow.com.

Three-phase PWM control register 0 (1)

b7	b6	b5	b4	b3	b2	b1	b0

Symbol: INVC0 — Address: 034816 — After reset: 0016

Bit symbol	Bit name	Description	RW
INV00	Effective interrupt output polarity select bit (3)	0: The ICTB2 counter is incremented by one on the rising edge of the timer A1 reload control signal 1: The ICTB2 counter is incremented by one on the falling edge of the timer A1 reload control signal	RW
INV01	Effective interrupt output specification bit (2, 3)	0: ICTB2 counter incremented by 1 at a timer B2 underflow 1: Selected by INV00 bit	RW
INV02	Mode select bit (4)	0: Three-phase motor control timer function unused 1: Three-phase motor control timer function (5)	RW
INV03	Output control bit (6)	0: Three-phase motor control timer output disabled (5) 1: Three-phase motor control timer output enabled (10)	RW
INV04	Positive and negative phases concurrent output disable bit	0: Simultaneous active output enabled 1: Simultaneous active output disabled	RW
INV05	Positive and negative phases concurrent output detect flag	0: Not detected yet 1: Already detected (7)	RW
INV06	Modulation mode select bit (8)	0: Triangular wave modulation mode (9) 1: Sawtooth wave modulation mode	RW
INV07	Software trigger select bit	Setting this bit to “1” generates a transfer trigger. If the INV06 bit is “1”, a trigger for the dead time timer is also generated. The value of this bit when read is “0”.	RW

NOTES:
1. Write to this register after setting the PRC1 bit in the PRCR register to “1” (write enable). Note also that INV00 to INV02, INV04 and INV06 bits can only be rewritten when timers A1, A2, A4 and B2 are idle.
2. If this bit needs to be set to “1”, set any value in the ICTB2 register before writing to it.
3. Effective when the INV11 bit is set to “1” (three-phase mode 1). If INV11 is set to “0” (three-phase mode 0), the ICTB2 counter is incremented by “1” each time the timer B2 underflows, regardless of whether the INV00 and INV01 bits are set. When setting the INV01 bit to “1”, set the timer A1 count start flag before the first timer B2 underflow.When the INV00 bit is set to “1”, the first interrupt is generated when the timer B2 underflows *n-1* times, if *n* is the value set in the ICTB2 counter. Subsequent interrupts are generated every *n* times the timer B2 underflow.
4. Setting the INV02 bit to “1” activates the dead time timer, U/V/W-phase output control circuits and ICTB2 counter.
5. When the INV02 bit is set to “1”(theee-phase control timer functions) and the INV03 is set to "0"(three-phase motor control timer output disabled), U, U, V, V, W and W pins, including pins shared with other output functions, enter a high-impedance state.
6. The INV03 bit is set to “0” in the following cases:
   - When reset
   - When positive and negative go active (INV05="1") simultaneously while INV04 bit is set to “1”
   - When set to “0” in a program
   - When input on the SD pin changes state from “H” to “L” (The INV03 bit cannot be set to “1” when SD input is “L”.) When both the INV04 and the INV05 bits are set to “1”, the INV03 bit is set to “0”.
7. Can only be set by writing “0” in a program, and cannot be set to “1”.
8. The effects of the INV06 bit are described in the table below.

Item	INV06=0	INV06=1
Mode	Triangular wave modulation mode	Sawtooth wave modulation mode
Timing at which transferred from IDB0 to IDB1 registers to three-phase output shift register	Transferred only once synchronously with the transfer trigger after writing to the IDB0 to IDB1 registers	Transferred every transfer trigger
Timing at which dead time timer trigger is generated when INV16 bit is “0”	Synchronous with the falling edge of timer A1, A2, or A4 one-shot pulse	Synchronous with the transfer trigger and the falling edge of timer A1, A2, or A4 one-shot pulse
INV13 bit	Effective when INV11 is “1” and INV06 is “0”	Has no effect

Transfer trigger: Timer B2 underflow, write to the INV07 bit or write to the TB2 register when INV10 is “1”

9. If the INV06 bit is “1”, set the INV11 bit to “0” (three-phase mode 0) and set the PWCON bit to “0” (timer B2 reloaded by a timer B2 underflow).
10. Individual pins can be disabled using PFCR register.

Figure 12.3.2. INVC0 Register

Downloaded from Arrow.com.

Three-phase PWM control register 1 (1)

b7	b6	b5	b4	b3	b2	b1	b0
0							

Symbol: INVC1 | Address: 034916 | After reset: 0016

Bit symbol	Bit name	Description	RW
INV10	Timer A1, A2, A4 start trigger signal select bit	0: Timer B2 underflow 1: Timer B2 underflow and write to the TB2 register (2)	RW
INV11	Timer A1-1, A2-1, A4-1 control bit (3)	0: Three-phase mode 0 1: Three-phase mode 1 (4)	RW
INV12	Dead time timer count source select bit	0 : f1 or f2 1 : f1 divided by 2 or f2 divided by 2	RW
INV13	Carrier wave detect flag (5)	0: Timer A1 reload control signal is “0” 1: Timer A1 reload control signal is “1”	RO
INV14	Output polarity control bit	0 : Output waveform “L” active 1 : Output waveform “H” active	RW
INV15	Dead time invalid bit	0: Dead time timer enabled 1: Dead time timer disabled	RW
INV16	Dead time timer trigger select bit	0: Falling edge of timer A4, A1 or A2 one-shot pulse 1: Rising edge of three-phase output shift register (U, V or W phase) output (6)	RW
——— (b7)	Reserved bit	This bit should be set to “0”	RW

NOTES:

1. Write to this register after setting the PRC1 bit in the PRCR register to “1” (write enable). Note also that this register can only be rewritten when timers A1, A2, A4 and B2 are idle.
2. A start trigger is generated by writing to the TB2 register only while timer B2 stops.
3. The effects of the INV11 bit are described in the table below.

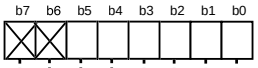
Item	INV11=0	INV11=1
Mode	Three-phase mode 0	Three-phase mode 1
TA11, TA21, TA41 registers	Not used	Used
INV00 bit, INV01 bit	Has no effect. ICTB2 counted every time timer B2 underflows regardless of whether the INV00 to INV01 bits are set.	Effect
INV13 bit	Has no effect	Effective when INV11 bit is set to “1” and INV06 bit is set to “0”

4. If the INV06 bit is set to “1” (sawtooth wave modulation mode), set this bit to “0” (three-phase mode 0). Also, if the INV11 bit is “0”, set the PWCON bit to “0” (timer B2 reloaded by a timer B2 underflow).
5. The INV13 bit is effective only when the INV06 bit is set to “0” (triangular wave modulation mode) and the INV11 bit is set to “1” (three-phase mode 1).
6. If all of the following conditions hold true, set the INV16 bit to “1” (dead time timer triggered by the rising edge of three-phase output shift register output)• The INV15 bit is set to “0” (dead time timer enabled)• When the INV03 bit is set to “1” (three-phase motor control timer output enabled), the Dij bit and DiBj bit (i:U, V, or W, j: 0 to 1) have always different values (the positive-phase and negative-phase always output different levels during the period other than dead time).
   Conversely, if either one of the above conditions holds false, set the INV16 bit to “0” (dead time timer triggered by the falling edge of one-shot pulse).

Figure 12.3.3. INVC1 Register

Downloaded from Arrow.com.

Three-phase output buffer register(i=0,1) (1)

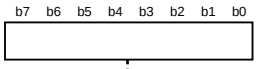


Symbol	Address	When reset
IDB0	034A16	00111111₂
IDB1	034B16	00111111₂

Bit symbol	Bit name	Function	RW
DUi	U phase output buffer i	Write the output level 0: Active level 1: Inactive level When read, these bits show the three-phase output shift register value.	RW
DUBi	$\overline{U}$ phase output buffer i		RW
DVi	V phase output buffer i		RW
DVBi	$\overline{V}$ phase output buffer i		RW
DWi	W phase output buffer i		RW
DWBi	$\overline{W}$ phase output buffer i		RW
$\overline{(b7\text{-}b6)}$	Nothing is assigned. When write, set to "0". When read, these contents are "0".		RO

NOTE:
1. The IDB0 and IDB1 register values are transferred to the three-phase shift register by a transfer trigger. The value written to the IDB0 register aftera transfer trigger represents the output signal of each phase, and the next value written to the IDB1 register at the falling edge of the timer A1, A2 or A4 one-shot pulse represents the output signal of each phase.

Dead time timer (1, 2)

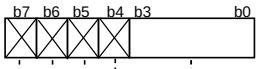


Symbol	Address	When reset
DTT	034C16	Indeterminate

Function	Setting range	RW
Assuming the set value = n, upon a start trigger the timer starts counting the count souce selected by the INV12 bit and stops after counting it n times. The positive or negative phase whichever is going from an inactive to an active level changes at the same time the dead time timer stops.	1 to 255	WO

NOTES:
1. Use MOV instruction to write to this register.
2. Effective when the INV15 bit is set to “0” (dead time timer enable). If the INV15 bit is set to “1”, the dead time timer is disabled and has no effect.

Timer B2 Interrupt Occurrences Frequency Set Counter



Symbol	Address	After Reset
ICTB2	034D16	Indeterminate

Function	Setting Range	RW
If the INV01 bit is "0" (ICTB2 counter counted every time timer B2 underflows), assuming the set value = n, a timer B2 interrupt is generated at every níth occurrence of a timer B2 underflow. If the INV01 bit is "1" (ICTB2 counter count timing selected by the INV00 bit), assuming the set value = n, a timer B2 interrupt is generated at every níth occurrence of a timer B2 underflow that meets the condition selected by the INV00 bit. (1)	1 to 15	WO
Nothing is assigned. When write, set to "0". When read, its content is indeterminate.		—

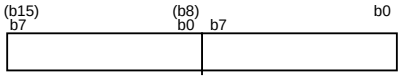
NOTE:
1. Use MOV instruction to write to this register.
If the INV01 bit is set to "1", make sure the TB2S bit also is set to "0" (timer B2 count stopped) when writing to this register. If the INV01 bit is set to "0", although this register can be written even when the TB2S bit is set to "1" (timer B2 count start), do not write synchronously with a timer B2 underflow.

Figure 12.3.4. IDB0 Register, IDB1Register, DTT Register, and ICTB2 Register

Downloaded from Arrow.com.

Timer Ai, Ai-1 register (i=1, 2, 4) (1, 2, 3, 4, 5)

Symbol	Address	After reset
TA1	0389₁₆-0388₁₆	Indeterminate
TA2	038B₁₆-038A₁₆	Indeterminate
TA4	038F₁₆-038E₁₆	Indeterminate
TA11 (6,7)	0343₁₆-0342₁₆	Indeterminate
TA21 (6,7)	0345₁₆-0344₁₆	Indeterminate
TA41 (6,7)	0347₁₆-0346₁₆	Indeterminate



Function	Setting range	RW
Assuming the set value = n, upon a start trigger the timer starts counting the count source and stops after counting it n times. The positive and negative phases change at the same time timer A, A2 or A4 stops.	0000₁₆ to FFFF₁₆	WO

NOTES:
1. The register must be accessed in 16 bit units.
2. When the timer Ai register is set to "0000₁₆", the counter does not operate and a timer Ai interrupt does not occur.
3. Use MOV instruction to write to these registers.
4. If the INV15 bit is "0" (dead time timer enable), the positive or negative phase whichever is going from an inactive to an active level changes at the same time the dead time timer stops.
5. If the INV11 bit is "0" (three-phase mode 0), the TAi register value is transferred to the reload register by a timer Ai (i = 1, 2 or 4) start trigger. If the INV11 bit is "1" (three-phase mode 1), the TAi1 register value is transferred to the reload register by a timer Ai start trigger first and then the TAi register value is transferred to the reload register by the next timer Ai start trigger. Thereafter, the TAi1 register and TAi register values are transferred to the reload register alternately.
6. Do not write to TAi1 registers synchronously with a timer B2 underflow In three-phase mode 1.
7. Write to the TAi1 register as follows:
   (1) Write a value to the TAi1 register
   (2) Wait for one cycle of timer Ai count source.
   (3) Write the same value to the TAi1 register again.

Figure 12.3.5. TA1, TA2, TA4, TA11, TA21 and TA41 Registers

Downloaded from Arrow.com.

Timer B2 Special Mode Register (1)

b7	b6	b5	b4	b3	b2	b1	b0
X	0	0					

Symbol	Address	After Reset
TB2SC	039E16	X0000000₂

Bit Symbol	Bit Name	Function	RW
PWCON	Timer B2 reload timing switch bit (2)	0: Timer B2 underflow 1: Timer A output at odd-numbered	RW
IVPCR1	Three-phase output port $\overline{SD}$ control bit 1 (3, 4, 7)	0: Three-phase output forcible cutoff by $\overline{SD}$ pin input (high impedance) disabled 1: Three-phase output forcible cutoff by $\overline{SD}$ pin input (high impedance) enabled	RW
TB0EN	Timer B0 operation mode select bit	0: Other than A/D trigger mode 1: A/D trigger mode (5)	RW
TB1EN	Timer B1 operation mode select bit	0: Other than A/D trigger mode 1: A/D trigger mode (5)	RW
TB2SEL	Trigger select bit (6)	0: TB2 interrupt 1: Underflow of TB2 interrupt generation frequency setting counter [ICTB2]	RW
$\overline{\text{(b6-b5)}}$	Reserved bits	Set to 0	RW
— (b7)	Nothing is assigned. If necessary, set to 0. When read, the content is 0.		—

NOTES:
1. Write to this register after setting the PRC1 bit in the PRCR register to 1 (write enabled).
2. If the INV11 bit is 0 (three-phase mode 0) or the INV06 bit is 1 (triangular wave modulation mode), set this bit to 0 (timer B2 underflow).
3. When setting the IVPCR1 bit to 1 (three-phase output forcible cutoff by $\overline{SD}$ pin input enabled), Set the PD8$_5$ bit to 0 (= input mode).
4. Related pins are U(P8$_0$), $\overline{U}$(P8$_1$), V(P7$_2$), $\overline{V}$(P7$_3$), W(P7$_4$), $\overline{W}$(P7$_5$). When a high-level ("H") signal is applied to the $\overline{SD}$ pin and set the IVPCR1 bit to 0 after forcible cutoff, pins U, $\overline{U}$, V, $\overline{V}$, W, and $\overline{W}$ are exit from the high-impedance state. If a low-level ("L") signal is applied to the $\overline{SD}$ pin, three-phase motor control timer output will be disabled (INV03=0). At this time, when the IVPCR1 bit is 0, pins U, $\overline{U}$, V, $\overline{V}$, W, and $\overline{W}$ become programmable I/O ports. When the IVPCR1 bit is set to 1, pins U, $\overline{U}$, V, $\overline{V}$, W, and $\overline{W}$ are placed in a high-impedance state regardless of which function of those pins is used.
5. When this bit is used in delayed trigger mode 0, set bits TB0EN and TB1EN to 1 (A/D trigger mode).
6. When setting the TB2SEL bit to 1 (underflow of TB2 interrupt generation frequency setting counter[ICTB2]), set the INV02 bit to 1 (three-phase motor control timer function).

The effect of $\overline{SD}$ pin input is below.
1.Case of INV03 = 1(Three-phase motor control timer output enabled)

IVPCR1 bit	$\overline{SD}$ pin inputs(3)	Status of U/V/W pins	Remarks
1 (Three-phase output forcible cutoff enable)	H	Three-phase PWM output	
	L(1)	High impedance(4)	Three-phase output forcible cutoff
0 (Three-phase output forcible cutoff disable)	H	Three-phase PWM output	
	L(1)	Input/output port(2)	

NOTES:
1. When "L" is applied to the $\overline{SD}$ pin, INV03 bit is changed to 0 at the same time.
2. The value of the port register and the port direction register becomes effective.
3. When $\overline{SD}$ function is not used, set to 0 (Input) in PD8$_5$ and pullup to "H" in $\overline{SD}$ pin from outside.
4. To leave the high-impedance state and restart the three-phase PWM signal output after the three-phase PWM signal output forced cutoff, set the IVPCR1 bit to 0 after the $\overline{SD}$ pin input level becomes high ("H").

2.Case of INV03 = 0(Three-phase motor control timer output disabled)

IVPCR1 bit	$\overline{SD}$ pin inputs	Status of U/V/W pins	Remarks
1 (Three-phase output forcible cutoff enable)	H	Peripheral input/output or input/output port	
	L	High impedance	Three-phase output forcible cutoff(1)
0 (Three-phase output forcible cutoff disable)	H	Peripheral input/output or input/output port	
	L	Peripheral input/output or input/output port	

NOTE:
1. The three-phase output forcible cutoff function becomes effective if the INPCR1 bit is set to 1 (three-phase output forcible cutoff function enable) even when the INV03 bit is 0 (three-phase motor control timer output disalbe)

Figure 12.3.6. TB2SC Registers

Downloaded from Arrow.com.

Timer B2 register (1)

Symbol	Address	After reset
TB2	039516 to 039416	Indeterminate

(b15) b7 (b8) b0 b7 b0

Function	Setting range	RW
Divide the count source by n + 1 where n = set value. Timer A1, A2 and A4 are started at every occurrence of underflow.	000016 to FFFF16	RW

NOTE:
1. The register must be accessed in 16 bit units.

Trigger select register

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	After reset
TRGSR	038316	0016

Bit symbol	Bit name	Function	RW
TA1TGL	Timer A1 event/trigger select bit	To use the V-phase output control circuit, set these bits to "01 2"(TB2 underflow).	RW
TA1TGH			RW
TA2TGL	Timer A2 event/trigger select bit	To use the W-phase output control circuit, set these bits to "01 2"(TB2 underflow).	RW
TA2TGH			RW
TA3TGL	Timer A3 event/trigger select bit	b5 b4 0 0 : Input on TA3IN is selected (1) 0 1 : TB2 overflow is selected (2) 1 0 : TA2 overflow is selected (2) 1 1 : TA4 overflow is selected (2)	RW
TA3TGH			RW
TA4TGL	Timer A4 event/trigger select bit	To use the U-phase output control circuit, set these bits to "01 2"(TB2 underflow).	RW
TA4TGH			RW

NOTES:
1. Set the corresponding port direction bit to "0" (input mode).
2. Overflow or underflow

Count start flag

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	After reset
TABSR	038016	0016

Bit symbol	Bit name	Function	RW
TA0S	Timer A0 count start flag	0 : Stops counting 1 : Starts counting	RW
TA1S	Timer A1 count start flag		RW
TA2S	Timer A2 count start flag		RW
TA3S	Timer A3 count start flag		RW
TA4S	Timer A4 count start flag		RW
TB0S	Timer B0 count start flag		RW
TB1S	Timer B1 count start flag		RW
TB2S	Timer B2 count start flag		RW

Figure 12.3.7. TB2 Register, TRGSR Register, and TABSR Register

Downloaded from Arrow.com.

Timer Ai mode register

b7	b6	b5	b4	b3	b2	b1	b0
		0	1		0	1	0

Symbol	Address	After reset
TA1MR	039716	0016
TA2MR	039816	0016
TA4MR	039A16	0016

Bit symbol	Bit name	Function	RW
TMOD0	Operation mode select bit	Must set to "102" (one-shot timer mode) for the three-phase motor control timer function	RW
TMOD1			RW
MR0	Pulse output function select bit	Must set to "0" for the three-phase motor control timer function	—
MR1	External trigger select bit	Has no effect for the three-phase motor control timer function	RW
MR2	Trigger select bit	Must set to "1" (selected by event/trigger select register) for the three-phase motor control timer function	RW
MR3	Must set to "0" for the three-phase motor control timer function		RW
TCK0	Count source select bit	b7 b6 0 0 : f1 or f2 0 1 : f8 1 0 : f32 1 1 : fC32	RW
TCK1			RW

Timer B2 mode register

b7	b6	b5	b4	b3	b2	b1	b0
			0			0	0

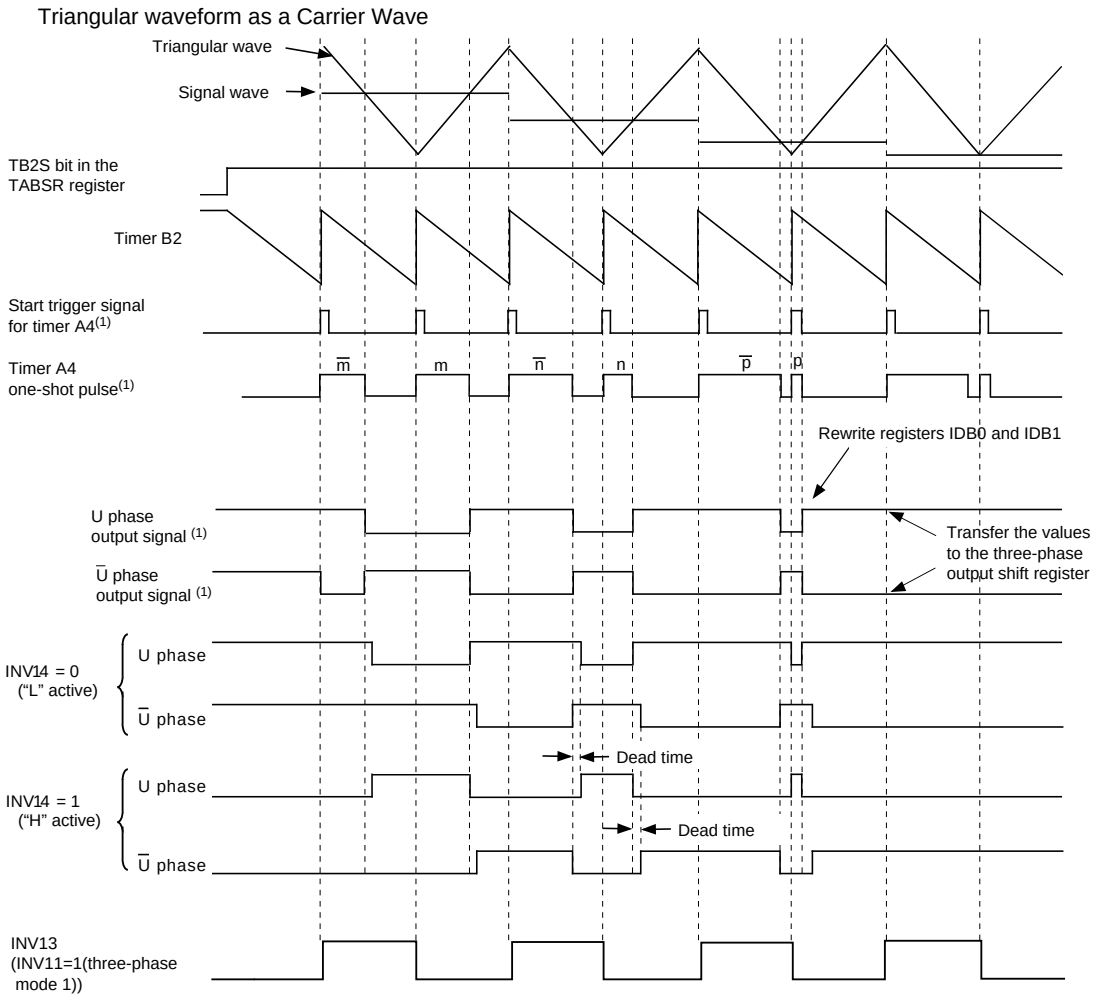
Symbol	Address	After reset
TB2MR	039D16	00XX00002

Bit symbol	Bit name	Function	RW
TMOD0	Operation mode select bit	Set to "002" (timer mode) for the three-phase motor control timer function	RW
TMOD1			RW
MR0	Has no effect for the three-phase motor control timer function. When write, set to "0". When read, its content is indeterminate.		RW
MR1			RW
MR2	Must set to "0" for the three-phase motor control timer function		RW
MR3	When write in three-phase motor control timer function, write "0". When read, its content is indeterminate.		RO
TCK0	Count source select bit	b7 b6 0 0 : f1 or f2 0 1 : f8 1 0 : f32 1 1 : fC32	RW
TCK1			RW

Figure 12.3.8. TA1MR, TA2MR, TA4MR, and TB2MR Registers

Downloaded from Arrow.com.

The three-phase motor control timer function is enabled by setting the INV02 bit in the VC0 register to “1”. When this function is on, timer B2 is used to control the carrier wave, and timers A4, A1 and A2 are used to control three-phase PWM outputs (U, $\overline{U}$, V, $\overline{V}$, W and $\overline{W}$). The dead time is controlled by a dedicated dead-time timer. Figure 12.3.9 shows the example of triangular modulation waveform, and Figure 12.3.10 shows the example of sawtooth modulation waveform.



NOTE:
1. Internal signals. See **Figure 12.3.1**.

The above applies under the following conditions:
INVC0 = 00XX11XX$_2$ (X varies depending on each system) and INVC1 = 010XXXX0$_2$.
Examples of PWM output change are:

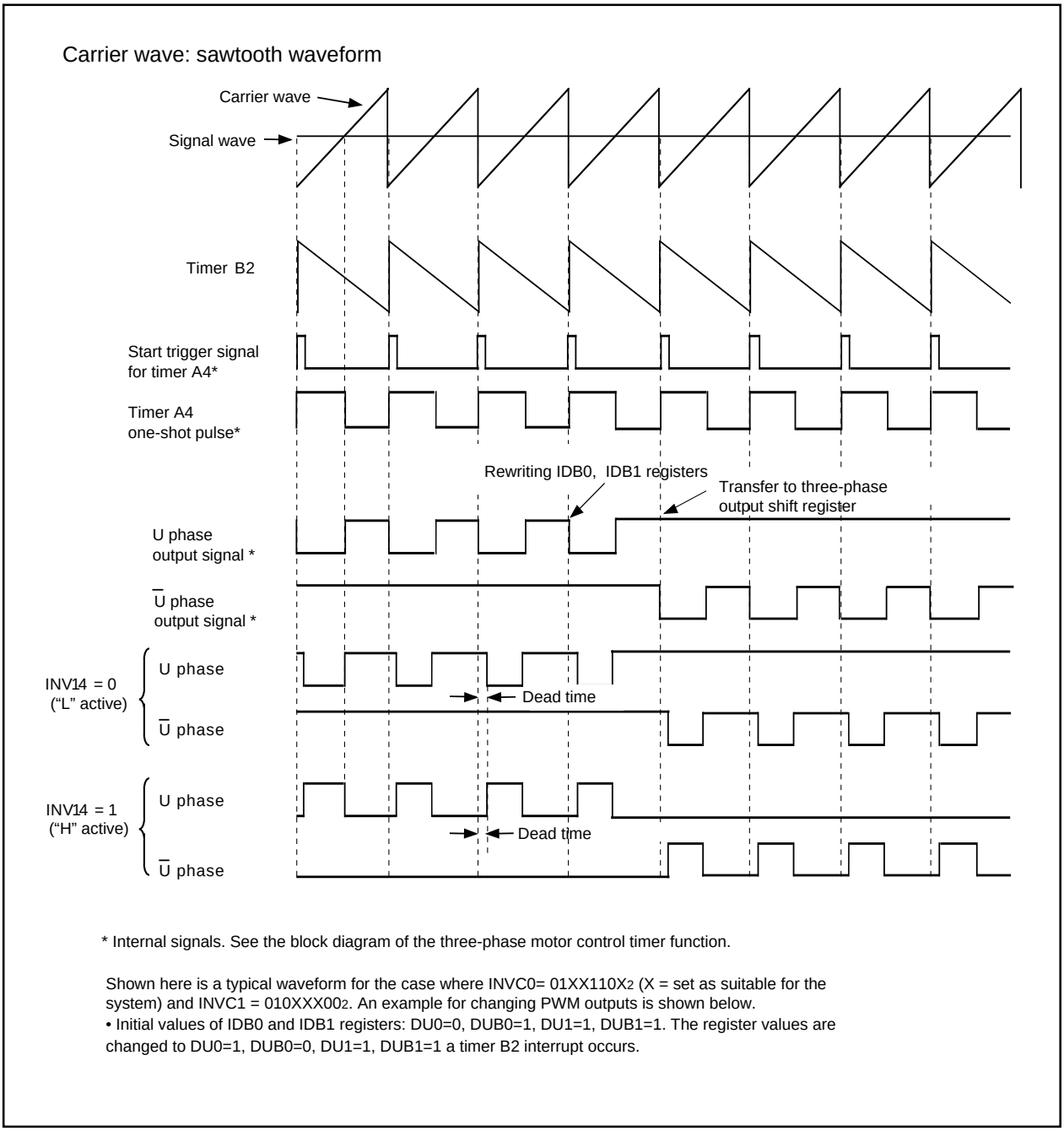
(1)When INV11 = 1 (three-phase mode 1)
- INV01 = 0 and ICTB2 = 2$_{16}$ (the timer B2 interrupt is generated every two times the timer B2 underflows), or INV01 = 1, INV00 = 1, and ICTB2=1$_{16}$ (the timer B2 interrupt is generated at the falling edge of the timer A1 reload control signal.)
- Default value of the timer: TA41 = $\overline{m}$, TA4 = m. Registers TA4 and TA41 are changed whenever the timer B2 interrupt is generated. First time, TA41 = $\overline{n}$, TA4 = n. Second time, TA41 = $\overline{p}$, TA4 = p.
- Default values of registers IDB0 and IDB1: DU0 = 1, DUB0 = 0, DU1 = 0, DUB1 = 1. They are changed to DU0 = 1, DUB0 = 0, DU1= 1 and DUB1 = 0 when the third timer B2 interrupt is generated.

(2)When INV11 = 0 (three-phase mode 0)
- INV01 = 0, ICTB2 = 1$_{16}$ (the timer B2 interrupt is generated whenever timer B2 underflows)
- Default value of the timer: TA4 = $\overline{m}$. The TA4 register is changed whenever the timer B2 interrupt is generated. First time: TA4 = m. Second tim:, TA4 = $\overline{n}$. Third time: TA4 = n. Fourth time: TA4 = $\overline{p}$. Fifth time: TA4 = p.
- Default values of registers IDB0 and IDB1: DU0 = 1, DUB0 = 0, DU1 = 0, DUB1 = 1. They are changed to DU0 = 1, DUB0 = 0, DU1 = 1, and DUB1 = 0 when the sixth timer B2 interrupt is generated.

The value written to registers TA4 and TA41 becomes effective at the rising edge of the timer A1 reload control signal.

Figure 12.3.9. Triangular Wave Modulation Operation

Downloaded from Arrow.com.



* Internal signals. See the block diagram of the three-phase motor control timer function.

Shown here is a typical waveform for the case where INVC0= 01XX110X2 (X = set as suitable for the system) and INVC1 = 010XXX002. An example for changing PWM outputs is shown below.
• Initial values of IDB0 and IDB1 registers: DU0=0, DUB0=1, DU1=1, DUB1=1. The register values are changed to DU0=1, DUB0=0, DU1=1, DUB1=1 a timer B2 interrupt occurs.

Figure 12.3.10. Sawtooth Wave Modulation Operation

Downloaded from Arrow.com.

12.3.1 Position-data-retain Function

This function is used to retain the position data synchronously with the three-phase waveform output.There are three position-data input pins for U, V, and W phases.

A trigger to retain the position data (hereafter, this trigger is referred to as "retain trigger") can be selected by the retain-trigger polarity select bit(bit 3 of the position-data-retain function control register, at address 034E16). This bit selects the retain trigger to be the falling edge of each positive phase, or the rising edge of each positive phase.

12.3.1.1 Operation of the Position-data-retain Function

Figure 12.3.1.1.1 shows a usage example of the position-data-retain function (U phase) when the retain trigger is selected as the falling edge of the positive signal.

(1) At the falling edge of the U-phase waveform ouput, the state at pin IDU is transferred to the U-phase position data retain bit (bit2 at address 034E16).

(2) Until the next falling edge of the Uphase waveform output,the above value is retained.

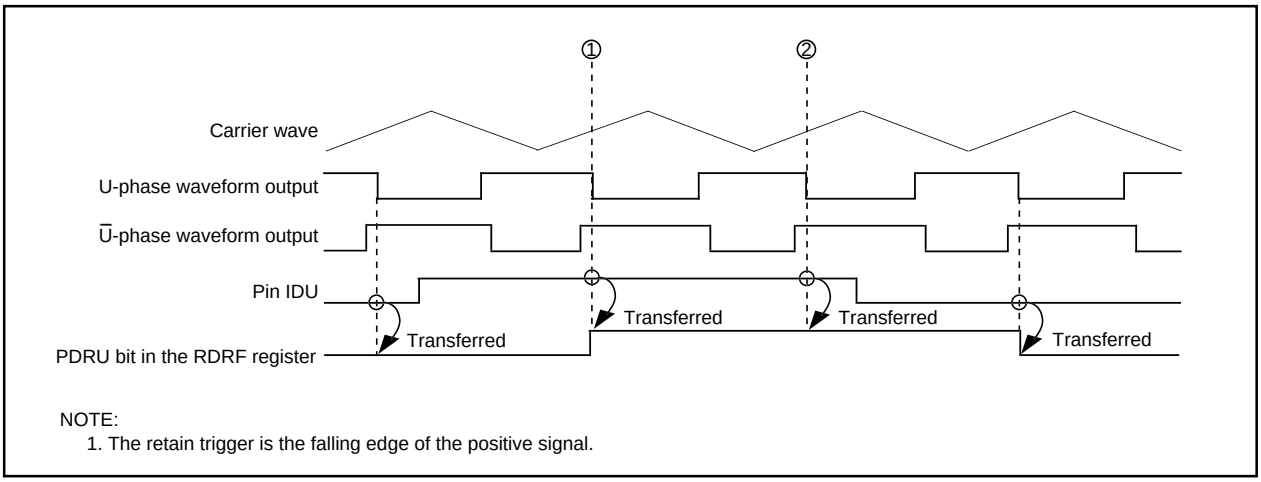


Figure 12.3.1.1.1 Usage Example of Position-data-retain Function (U phase)

Downloaded from Arrow.com.

12.3.1.2 Position-data-retain Function Control Register

Figure 12.3.1.2.1 shows the structure of the position-data-retain function contol register.

Position-data-retain function control register (1)

b7	b6	b5	b4	b3	b2	b1	b0
X	X	X	X				

Symbol	Address	When reset
PDRF	034E16	XXXX 00002

Bit symbol	Bit name	Function	RW
PDRW	W-phase position data retain bit	Input level at pin IDW is read out. 0: "L" level 1: "H" level	RO
PDRV	V-phase position data retain bit	Input level at pin IDV is read out. 0: "L" level 1: "H" level	RO
PDRU	U-phase position data retain bit	Input level at pin IDU is read out. 0: "L" level 1: "H" level	RO
PDRT	Retain-trigger polarity select bit	0: Rising edge of positive phase 1: Falling edge of positive phase	RW
(b7-b4)	Nothing is assigned. When write, set to "0". When read, contents are indeterminate.		—

NOTE:
1. This register is valid only in the three-phase mode.

Figure 12.3.1.2.1. PDRF Register

12.3.1.2.1 W-phase Position Data Retain Bit (PDRW)

This bit is used to retain the input level at pin IDW.

12.3.1.2.2 V-phase Position Data Retain Bit (PDRV)

This bit is used to retain the input level at pin IDV.

12.3.1.2.3 U-phase Position Data Retain Bit (PDRU)

This bit is used to retain the input level at pin IDU.

12.3.1.2.4 Retain-trigger Polarity Select Bit (PDRT)

This bit is used to select the trigger polarity to retain the position data.

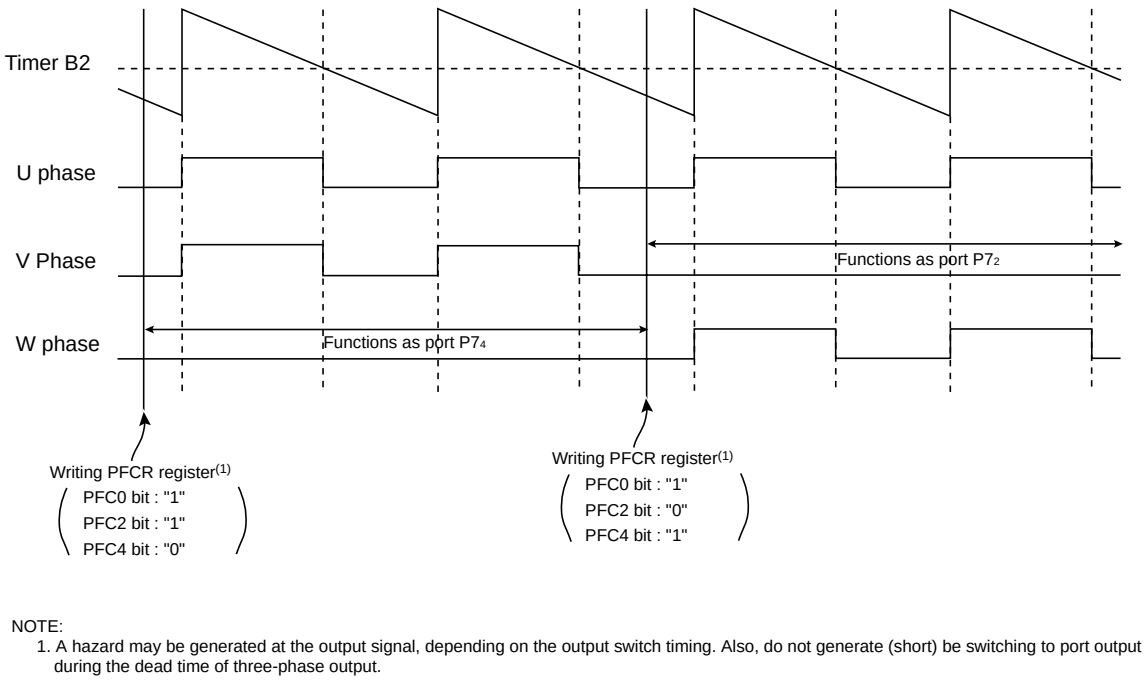
When this bit is set to "0", the rising edge of each positive phase selected.

When this bit is set to "1", the falling edge of each pocitive phase selected.

Downloaded from Arrow.com.

12.3.2 Three-phase/Port Output Switch Function

When the INVC03 bit in the INVC0 register set to “1”(Timer output enabled for three-phase motor control) and setting the PFCi (i=0 to 5) in the PFCR register to “0”(I/O port), the three-phase PWM output pin (U, $\overline{U}$, V, $\overline{V}$, W and $\overline{W}$) functions as I/O port. Each bit in the PFCi bits (i=0 to 5) is applicable for each one of three-phase PWM output pins. Figure 12.3.2.1 shows the example of three-phase/port output switch function. Figure 12.3.2.2 shows the PFCR register and the three-phase protect control register.



NOTE:
1. A hazard may be generated at the output signal, depending on the output switch timing. Also, do not generate (short) be switching to port output during the dead time of three-phase output.

Figure 12.3.2.1. Usage Example of Three-phse/Port output switch function

Downloaded from Arrow.com.

Port function control register (1)

b7	b6	b5	b4	b3	b2	b1	b0
✕	✕						

Symbol	Address	When reset
PFCR	0358_{16}	$0011\ 1111_2$

Bit symbol	Bit name	Function	RW
PFC0	Port $P8_0$ output function select bit	0: Input/Output port $P8_0$ 1: Three-phase PWM output (U phase output)	RW
PFC1	Port $P8_1$ output function select bit	0: Input/Output port $P8_1$ 1: Three-phase PWM output ($\overline{U}$ phase output)	RW
PFC2	Port $P7_2$ output function select bit	0: Input/Output port $P7_2$ 1: Three-phase PWM output (V phase output)	RW
PFC3	Port $P7_3$ output function select bit	0: Input/Output port $P7_3$ 1: Three-phase PWM output ($\overline{V}$ phase output)	RW
PFC4	Port $P7_4$ output function select bit	0: Input/Output port $P7_4$ 1: Three-phase PWM output (W phase output)	RW
PFC5	Port $P7_5$ output function select bit	0: Input/Output port $P7_5$ 1: Three-phase PWM output ($\overline{W}$ phase output)	RW
$\overline{\text{(b7-b6)}}$	Nothing is assigned. When write, set to "0". When read, these contents are "0".		—

NOTE:
1. This register is valid only when the INVC03 bit in the INVC0 register is set to "1"(Three-phase motor control timer output enabled). Write to this register after setting the TPRC0 bit in the TPRC register to "1" (write enable).

Three-phase protect control register

b7	b6	b5	b4	b3	b2	b1	b0
✕	✕	✕	✕	✕	✕	✕	

Symbol	Address	When Reset
TPRC	$025A_{16}$	00_{16}

Bit Symbol	Bit Name	Function	RW
TPRC0	Three-phase protect control bit	Enable write to PFCR register 0: Write protected 1: Write enabled	RW
$\overline{\text{(b7-b1)}}$	Nothing is assigned. When write, set to "0". When read, the contents are "0".		—

Figure 12.3.2.2. PFCR Register, and TPRC Register

Downloaded from Arrow.com.

13. Serial I/O

Note

UART0 is not available in the 42-pin package.

Serial I/O is configured with three channels: UART0 to UART2.

13.1. UARTi (i=0 to 2)

UARTi each have an exclusive timer to generate a transfer clock, so they operate independently of each other.

Figure 13.1.1 shows the block diagram of UARTi. Figures 13.1.2 and 13.1.3 shows the block diagram of the UARTi transmit/receive.

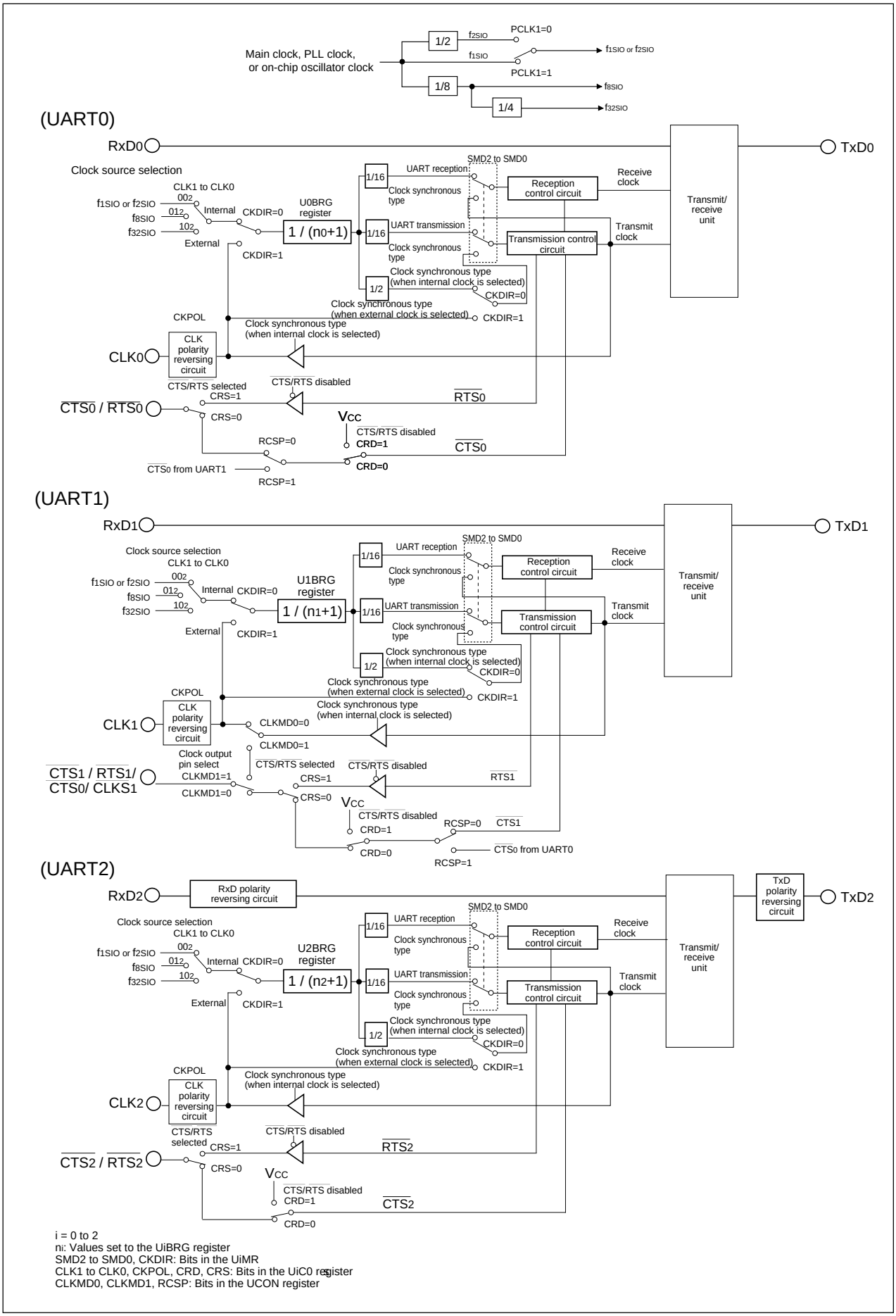
UARTi has the following modes:

- Clock synchronous serial I/O mode
- Clock asynchronous serial I/O mode (UART mode).
- Special mode 1 (I^2C bus mode) : UART2
- Special mode 2 : UART2
- Special mode 3 (Bus collision detection function, IEBus mode) : UART2
- Special mode 4 (SIM mode) : UART2

Figures 13.1.4 to 13.1.9 show the UARTi-related registers.

Refer to tables listing each mode for register setting.

Downloaded from Arrow.com.



i = 0 to 2
ni: Values set to the UiBRG register
SMD2 to SMD0, CKDIR: Bits in the UiMR
CLK1 to CLK0, CKPOL, CRD, CRS: Bits in the UiC0 register
CLKMD0, CLKMD1, RCSP: Bits in the UCON register

Figure 13.1.1. Block diagram of UARTi (i = 0 to 2)

Downloaded from Arrow.com.

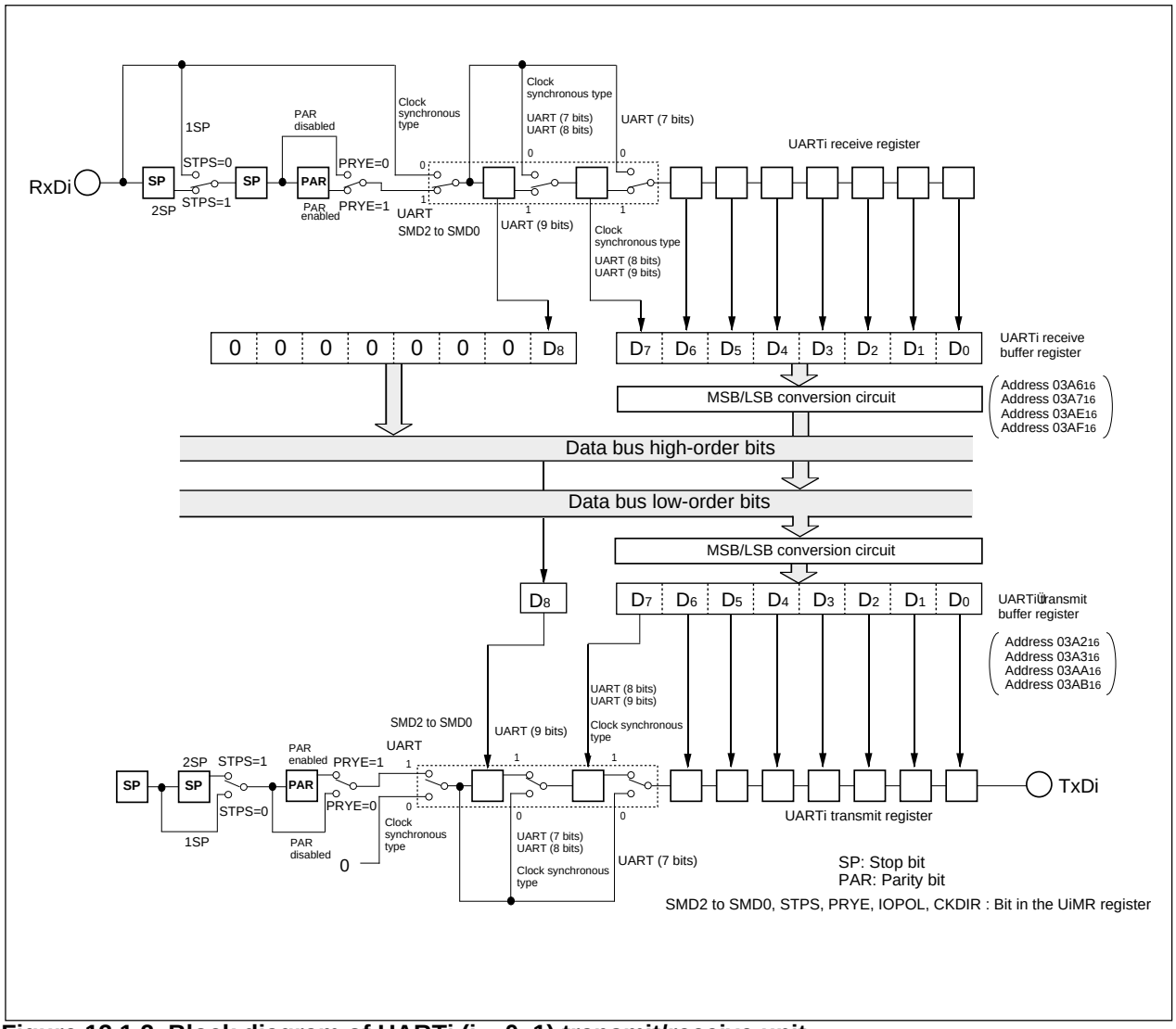


Figure 13.1.2. Block diagram of UARTi (i = 0, 1) transmit/receive unit

Downloaded from Arrow.com.

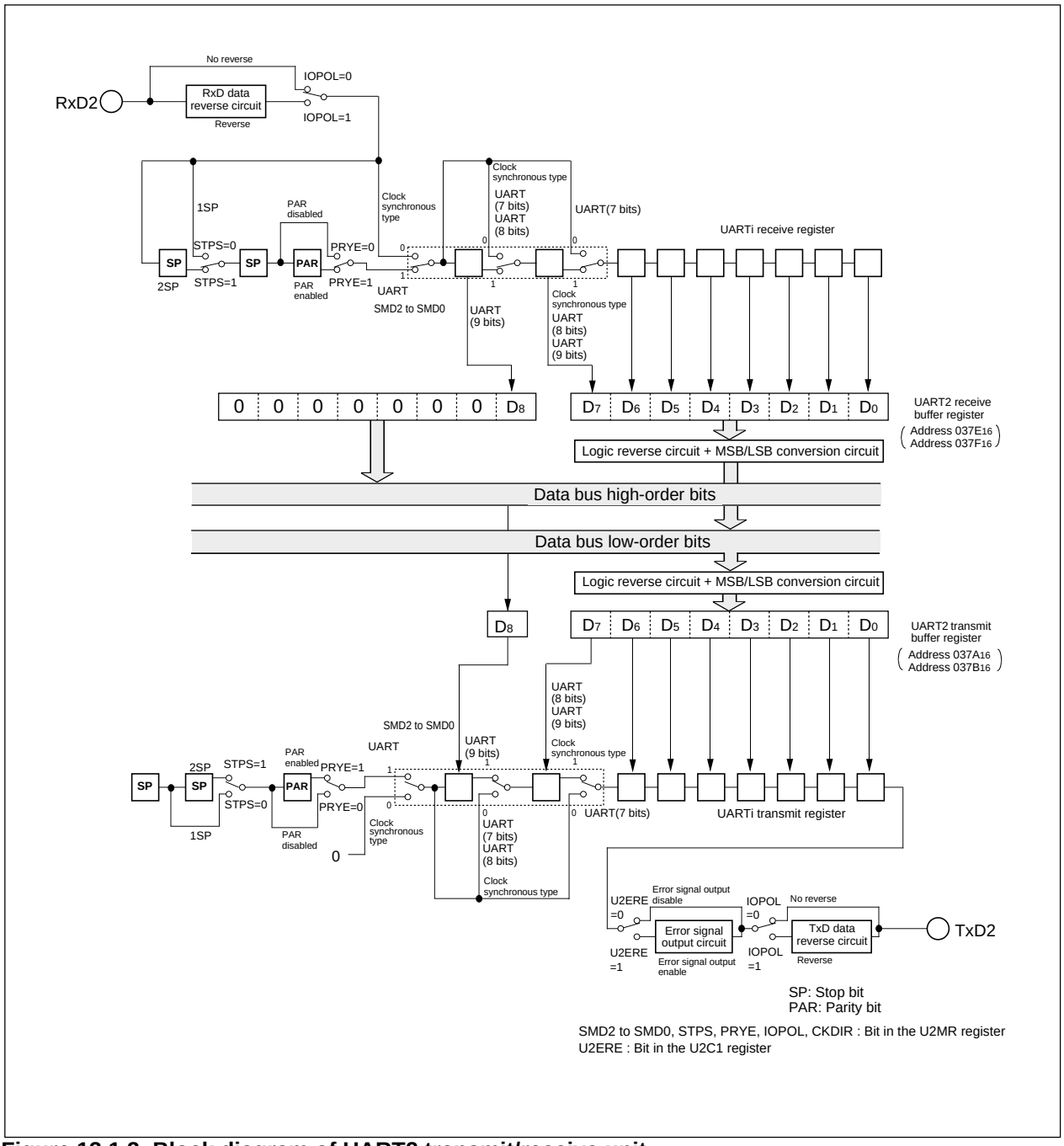
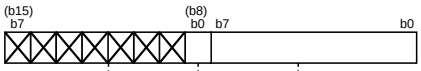


Figure 13.1.3. Block diagram of UART2 transmit/receive unit

Downloaded from Arrow.com.

UARTi Transmit Buffer Register (i=0 to 2)(1)

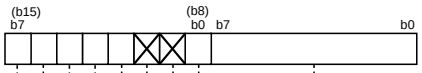


Symbol	Address	After Reset
U0TB	$03A3_{16}$-$03A2_{16}$	Indeterminate
U1TB	$03AB_{16}$-$03AA_{16}$	Indeterminate
U2TB	$037B_{16}$-$037A_{16}$	Indeterminate

Function	RW
Transmit data	WO
Nothing is assigned. When write, set to "0". When read, its content is indeterminate.	—

NOTES:
1. Use MOV instruction to write to this register.

UARTi Receive Buffer Register (i=0 to 2)

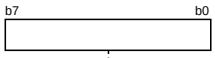


Symbol	Address	After Reset
U0RB	$03A7_{16}$-$03A6_{16}$	Indeterminate
U1RB	$03AF_{16}$-$03AE_{16}$	Indeterminate
U2RB	$037F_{16}$-$037E_{16}$	Indeterminate

Bit Symbol	Bit Name	Function	RW
— (b7-b0)	———	Receive data (D_7 to D_0)	RO
— (b8)	———	Receive data (D_8)	RO
— (b10-b9)	Nothing is assigned. When write, set to "0". When read, its content is indeterminate.		—
ABT	Arbitration lost detecting flag (2)	0 : Not detected 1 : Detected	RW
OER	Overrun error flag (1)	0 : No overrun error 1 : Overrun error found	RO
FER	Framing error flag (1)	0 : No framing error 1 : Framing error found	RO
PER	Parity error flag (1)	0 : No parity error 1 : Parity error found	RO
SUM	Error sum flag (1)	0 : No error 1 : Error found	RO

NOTES:
1. When the SMD2 to SMD0 bits in the UiMR register are set to "000_2" (serial I/O disabled) or the RE bit in the UiC1 register is set to "0" (reception disabled), all of the SUM, PER, FER and OER bits are set to "0" (no error). The SUM bit is set to "0" (no error) when all of the PER, FER and OER bits are set to "0" (no error). Also, the PER and FER bits are set to "0" by reading the lower byte of the UiRB register.
2. The ABT bit is set to "0" by setting to "0" by program. (Writing "1" has no effect.)
   Nothing is assigned at the bit 11 in the U0RB and U1RB registers. When write, set to "0". When read, its content is "0".

UARTi Baud Rate Generation Register (i=0 to 2)(1, 2, 3)



Symbol	Address	After Reset
U0BRG	$03A1_{16}$	Indeterminate
U1BRG	$03A9_{16}$	Indeterminate
U2BRG	0379_{16}	Indeterminate

Function	Setting Range	RW
Assuming that set value = n, UiBRG divides the count source by n + 1	00_{16} to FF_{16}	WO

NOTES:
1. Write to this register while serial I/O is neither transmitting nor receiving.
2. Use MOV instruction to write to this register.
   The transfer clock is shown below when the setting value in the UiBRG register is set as n.
   (1) When the CKDIR bit in the UiMR register to "0" (internal clock)
   - Clock synchronous serial I/O mode : fj/(2(n+1))
   - Clock asynchronous serial I/O (UART) mode : fj/(16(n+1))

   (2) When the CKDIR bit in the UiMR register to "1" (external clock)
   - Clock synchronous serial I/O mode : f_{EXT}
   - Clock asynchronous serial I/O (UART) mode : $f_{EXT}/(16(n+1))$

   fj : f1SIO, f2SIO, f8SIO, f32SIO
   f_{EXT} : Input from CLKi pin
3. Set the UiBRG register after setting the CLK1 and CLK0 bits in the UiC0 registers.

Figure 13.1.4. U0TB to U2TB registers, U0RB to U2RB registers, U0BRG to U2BRG registers

Downloaded from Arrow.com.

UARTi transmit/receive mode register (i = 0, 1)

b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	After reset
U0MR, U1MR	$03A0_{16}$, $03A8_{16}$	00_{16}

Bit symbol	Bit name	Function	RW
SMD0	Serial I/O mode select bit (2)	b2 b1 b0 0 0 0 : Serial I/O disabled 0 0 1 : Clock synchronous serial I/O mode 1 0 0 : UART mode transfer data 7 bits long 1 0 1 : UART mode transfer data 8 bits long 1 1 0 : UART mode transfer data 9 bits long Do not set value other than the above	RW
SMD1			RW
SMD2			RW
CKDIR	Internal/external clock select bit	0 : Internal clock 1 : External clock (1)	RW
STPS	Stop bit length select bit	0 : One stop bit 1 : Two stop bits	RW
PRY	Odd/even parity select bit	Effective when PRYE = 1 0 : Odd parity 1 : Even parity	RW
PRYE	Parity enable bit	0 : Parity disabled 1 : Parity enabled	RW
— (b7)	Reserve bit	Write to "0"	RW

NOTES:
1. Set the corresponding port direction bit for each CLKi pin to “0” (input mode).
2. To receive data, set the corresponding port direction bit for each RxDi pin to “0” (input mode).

UART2 transmit/receive mode register

b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	After reset
U2MR	0378_{16}	00_{16}

Bit symbol	Bit name	Function	RW
SMD0	Serial I/O mode select bit (2)	b2 b1 b0 0 0 0 : Serial I/O disabled 0 0 1 : Clock synchronous serial I/O mode 0 1 0 : I^2C bus mode (3) 1 0 0 : UART mode transfer data 7 bits long 1 0 1 : UART mode transfer data 8 bits long 1 1 0 : UART mode transfer data 9 bits long Must not be set except above	RW
SMD1			RW
SMD2			RW
CKDIR	Internal/external clock select bit	0 : Internal clock 1 : External clock (1)	RW
STPS	Stop bit length select bit	0 : One stop bit 1 : Two stop bits	RW
PRY	Odd/even parity select bit	Effective when PRYE = 1 0 : Odd parity 1 : Even parity	RW
PRYE	Parity enable bit	0 : Parity disabled 1 : Parity enabled	RW
IOPOL	TxD, RxD I/O polarity reverse bit	0 : No reverse 1 : Reverse	RW

NOTES:
1. Set the corresponding port direction bit for each CLK2 pin to “0” (input mode).
2. To receive data, set the corresponding port direction bit for each RxD2 pin to “0” (input mode).
3. Set the corresponding port direction bit for SCL2 and SDA2 pins to “0” (input mode).

Figure 13.1.5. U0MR to U2MR registers

Downloaded from Arrow.com.

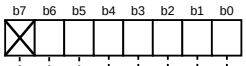
UARTi Transmit/receive Control Rregister 0 (i=0 to 2)

Symbol	Address	After Reset
U0C0 to U2C0	$03A4_{16}$, $03AC_{16}$, $037C_{16}$	00001000_2

Bit Symbol	Bit Name	Function	RW
CLK0	BRG count source select bit (7)	b1 b0 0 0 : $f1_{SIO}$ or $f2_{SIO}$ is selected 0 1 : $f8_{SIO}$ is selected 1 0 : $f32_{SIO}$ is selected 1 1 : Do not set	RW
CLK1			RW
CRS	$\overline{CTS}/\overline{RTS}$ function select bit (3)	Effective when CRD is set to "0" 0 : $\overline{CTS}$ function is selected (1) 1 : $\overline{RTS}$ function is selected	RW
TXEPT	Transmit register empty flag	0 : Data in transmit register (during transmission) 1 : No data in transmit register (transmission completed)	RO
CRD	$\overline{CTS}/\overline{RTS}$ disable bit	0 : $\overline{CTS}/\overline{RTS}$ function enabled 1 : $\overline{CTS}/\overline{RTS}$ function disabled ($P6_0$, $P6_4$ and $P7_3$ can be used as I/O ports)(6)	RW
NCH	Data output select bit(5)	0 : TxDi/SDA2 and SCL2 pins are CMOS output 1 : TxDi/SDA2 and SCL2 pins are N-channel open-drain output(4)	RW
CKPOL	CLK polarity select bit	0 : Transmit data is output at falling edge of transfer clock and receive data is input at rising edge 1 : Transmit data is output at rising edge of transfer clock and receive data is input at falling edge	RW
UFORM	Transfer format select bit (2)	0 : LSB first 1 : MSB first	RW

NOTES:
1. Set the corresponding port direction bit for each $\overline{CTS}i$ pin to "0" (input mode).
2. Effective when the SMD2 to SMD0 bits in the UMR register to "001_2"(clock synchronous serial I/O mode) or "010_2" (UART mode transfer data 8 bits long). Set the UFORM bit to "1" when the SMD2 to SMD0 bits are set to "101_2" (I^2C bus mode) and "0" when they are set to"100_2" (UART mode transfer data 7 bits long) or "110_2" (UART mode transfer data 9 bits long).
3. $\overline{CTS}_1/\overline{RTS}_1$ can be used when the CLKMD1 bit in the UCON register is set to "0" (only CLK_1 output) and the RCSP bit in the UCON register is set to "0" ($\overline{CTS}_0/\overline{RTS}_0$ not separated).
4. SDA2 and SCL2 are effective when i = 2.
5. When the SMD2 to SMD0 bits in UiMR regiser are set to "000_2" (serial I/O disable), do not set NCH bit to "1" (TxDi/SDA2 and SCL2 pins are N-channel open-drain output).
6. When the U1MAP bit in PACR register is "1" ($P7_3$ to $P7_0$), $\overline{CTS}/\overline{RTS}$ pin in UART1 is assigned to $P7_0$.
7. When the CLK1 and CLK0 bit settings are changed, set the UiBRG register.

UART Transmit/receive Control Register 2



Symbol	Address	After Reset
UCON	$03B0_{16}$	$X0000000_2$

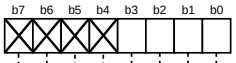
Bit Symbol	Bit Name	Function	RW
U0IRS	UART0 transmit interrupt cause select bit	0: Transmit buffer empty (TI = 1) 1: Transmission completed (TXEPT = 1)	RW
U1IRS	UART1 transmit interrupt cause select bit	0: Transmit buffer empty (TI = 1) 1: Transmission completed (TXEPT = 1)	RW
U0RRM	UART0 continuous receive mode enable bit	0: Continuous receive mode disabled 1: Continuous receive mode enable	RW
U1RRM	UART1 continuous receive mode enable bit	0: Continuous receive mode disabled 1: Continuous receive mode enabled	RW
CLKMD0	UART1 CLK/CLKS select bit 0	Effective when CLKMD1 bit is set to "1" 0: Clock output from CLK1 1: Clock output from CLKS1	RW
CLKMD1	UART1 CLK/CLKS select bit 1 (1)	0: Output from CLK1 only 1: Transfer clock output from multiple pins function selected	RW
RCSP	Separate UART0 $\overline{CTS}/\overline{RTS}$ bit	0: $\overline{CTS}/\overline{RTS}$ shared pin 1: $\overline{CTS}/\overline{RTS}$ separated ($\overline{CTS}_0$ supplied from the $P6_4$ pin)(2)	RW
— (b7)	Nothing is assigned. When write, set to "0". When read, the content is indeterminate		—

NOTES:
1. To use more than one transfer clock output pins, set the CKDIR bit in the U1MR register to "0" (internal clock).
2. When the U1MAP bit in PACR register is set to "1" ($P7_3$ to $P7_0$), $\overline{CTS}_0$ is supplied from the $P7_0$ pin.

Figure 13.1.6. U0C0 to U2C0 registers and UCON register

Downloaded from Arrow.com.

UARTi Transmit/receive Control Register 1 (i=0, 1)



Symbol	Address	After Reset
U0C1, U1C1	$03A5_{16}$,$03AD_{16}$	00000010_2

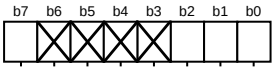
Bit Symbol	Bit Name	Function	RW
TE	Transmit enable bit	0 : Transmission disabled 1 : Transmission enabled	RW
TI	Transmit buffer empty flag	0 : Data in UiTB register 1 : No data in UiTB register	RO
RE	Receive enable bit	0 : Reception disabled 1 : Reception enabled	RW
RI	Receive complete flag	0 : No data in UiRB register 1 : Data in UiRB register	RO
— (b7-b4)	Nothing is assigned. When write, set “0”. When read, the contents are “0”.		—

UART2 Transmit/receive Control Register 1

Symbol	Address	After Reset
U2C1	$037D_{16}$	00000010_2

Bit Symbol	Bit Name	Function	RW
TE	Transmit enable bit	0 : Transmission disabled 1 : Transmission enabled	RW
TI	Transmit buffer empty flag	0 : Data in U2TB register 1 : No data in U2TB register	RO
RE	Receive enable bit	0 : Reception disabled 1 : Reception enabled	RW
RI	Receive complete flag	0 : No data in U2RB register 1 : Data in U2RB register	RO
U2IRS	UART2 transmit interrupt cause select bit	0 : Transmit buffer empty (TI = 1) 1 : Transmit is completed (TXEPT = 1)	RW
U2RRM	UART2 continuous receive mode enable bit	0 : Continuous receive mode disabled 1 : Continuous receive mode enabled	RW
U2LCH	Data logic select bit	0 : No reverse 1 : Reverse	RW
U2ERE	Error signal output enable bit	0 : Output disabled 1 : Output enabled	RW

Pin Assignment Control Register (1)



Symbpl	Address	After Reset
PACR	$025D_{16}$	00_{16}

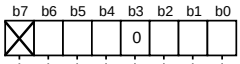
Bit Symbol	Bit Name	Function	RW
PACR0	Pin enabling bit	001 : 42 pin 100 : 48 pin All other values are reserved. Do not use.	RW
PACR1			RW
PACR2			RW
— (b6-b3)	Reserved bits	Nothing is assigned. When write, set to “0”. When read, its content is “0”.	—
U1MAP	UART1 pin remapping bit	UART1 pins assigned to 0 : $P6_7$ to $P6_4$ 1 : $P7_3$ to $P7_0$	RW

NOTES:
1. Set the PACR register by the next instruction after setting the PRC2 bit in the PRCR register to "1"(write enable).

Figure 13.1.7. U0C1 to U2C1 registers, PACR register

Downloaded from Arrow.com.

UART2 Special Mode Register

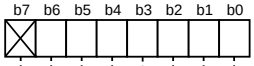


Symbol: U2SMR | Address: 037716 | After Reset: X0000000$_2$

Bit Symbol	Bit Name	Function	RW
IICM	I^2C bus mode select bit	0 : Other than I^2C bus mode 1 : I^2C bus mode	RW
ABC	Arbitration lost detecting flag control bit	0 : Update per bit 1 : Update per byte	RW
BBS	Bus busy flag	0 : STOP condition detected 1 : START condition detected (busy)	RW (1)
— (b3)	Reserved bit	Set to “0”	RW
ABSCS	Bus collision detect sampling clock select bit	0 : Rising edge of transfer clock 1 : Underflow signal of timer A0	RW
ACSE	Auto clear function select bit of transmit enable bit	0 : No auto clear function 1 : Auto clear at occurrence of bus collision	RW
SSS	Transmit start condition select bit	0 : Not synchronized to RxDi 1 : Synchronized to RxDi (2)	RW
— (b7)	Nothing is assigned. When write, set “0”. When read, its content is indeterminate.		—

NOTES:
1: The BBS bit is set to “0” by writing “0" by program. (Writing “1” has no effect).
2: When a transfer begins, the SSS bit is set to “0” (Not synchronized to RxDi).

UART2 Special Mode Register 2



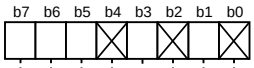
Symbol: U2SMR2 | Address: 037616 | After Reset: X0000000$_2$

Bit Symbol	Bit Name	Function	RW
IICM2	I^2C bus mode select bit 2	Refer to **Table 13.12**	RW
CSC	Clock-synchronous bit	0 : Disabled 1 : Enabled	RW
SWC	SCL_2 wait output bit	0 : Disabled 1 : Enabled	RW
ALS	SDA_2 output stop bit	0 : Disabled 1 : Enabled	RW
STAC	UART initialization bit	0 : Disabled 1 : Enabled	RW
SWC2	SCL_2 wait output bit 2	0: Transfer clock 1: “L” output	RW
SDHI	SDA_2 output disable bit	0: Enabled 1: Disabled (high impedance)	RW
— (b7)	Nothing is assigned. When write, set “0”. When read, its content is indeterminate.		—

Figure 13.1.8. U2SMR register and U2SMR2 register

Downloaded from Arrow.com.

UART2 special mode register 3

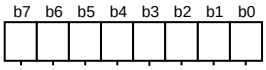


Symbol: U2SMR3 | Address: 037516 | After reset: 000X0X0X2

Bit symbol	Bit name	Function	RW
— (b0)	Nothing is assigned. When write, set "0". When read, its content is indeterminate.		—
CKPH	Clock phase set bit	0 : Without clock delay 1 : With clock delay	RW
— (b2)	Nothing is assigned. When write, set "0". When read, its content is indeterminate.		—
NODC	Clock output select bit	0 : CLKi is CMOS output 1 : CLKi is N-channel open drain output	RW
— (b4)	Nothing is assigned. When write, set "0". When read, its content is indeterminate.		—
DL0	SDA digital delay setup bit (1, 2)	b7 b6 b5 0 0 0 : Without delay 0 0 1 : 1 to 2 cycle(s) of UiBRG count source 0 1 0 : 2 to 3 cycles of UiBRG count source 0 1 1 : 3 to 4 cycles of UiBRG count source 1 0 0 : 4 to 5 cycles of UiBRG count source 1 0 1 : 5 to 6 cycles of UiBRG count source 1 1 0 : 6 to 7 cycles of UiBRG count source 1 1 1 : 7 to 8 cycles of UiBRG count source	RW
DL1			RW
DL2			RW

NOTES:
1. The DL2 to DL0 bits are used to generate a delay in SDA2 output by digital means during I^2C bus mode. In other than I^2C bus mode, set these bits to "000_2" (no delay).
2. The amount of delay varies with the load on SCL2 and SDA2 pins. Also, when using an external clock, the amount of delay increases by about 100 ns.

UART2 Special Mode Register 4



Symbol: U2SMR4 | Address: 037416 | After Reset: 0016

Bit Symbol	Bit Name	Function	RW
STAREQ	Start condition generate bit (1)	0: Clear 1: Start	RW
RSTAREQ	Restart condition generate bit (1)	0: Clear 1: Start	RW
STPREQ	Stop condition	0: Clear 1: Start	RW
STSPSEL	SCL_2, SDA_2 output	0: Start and stop conditions not output 1: Start and stop conditions output	RW
ACKD	ACK data bit	0: ACK 1: NACK	RW
ACKC	ACK data output	0: Serial I/O data output 1: ACK data output	RW
SCLHI	SCL_2 output stop	0: Disabled 1: Enabled	RW
SWC9	SCL_2 wait bit 3	0: SCL_2 "L" hold disabled 1: SCL_2 "L" hold enabled	RW

NOTE:
1. Set to "0" when each condition is generated.

Figure 13.1.9. U2SMR3 register and U2SMR4 register

Downloaded from Arrow.com.

13.1.1. Clock Synchronous serial I/O Mode

The clock synchronous serial I/O mode uses a transfer clock to transmit and receive data. Table 13.1.1.1 lists the specifications of the clock synchronous serial I/O mode. Table 13.1.1.2 lists the registers used in clock synchronous serial I/O mode and the register values set.

Table 13.1.1.1. Clock Synchronous Serial I/O Mode Specifications

Item	Specification
Transfer data format	• Transfer data length: 8 bits
Transfer clock	• The CKDIR bit in the UiMR(i=0 to 2) register is set to “0” (internal clock) : fj/ (2(n+1)) fj = f_{1SIO}, f_{2SIO}, f_{8SIO}, f_{32SIO}. n: Setting value of UiBRG register 00_{16} to FF_{16} • The CKDIR bit is set to “1” (external clock) : Input from CLKi pin
Transmission, reception control	• Selectable from $\overline{CTS}$ function, $\overline{RTS}$ function or $\overline{CTS}/\overline{RTS}$ function disable
Transmission start condition	• Before transmission can start, the following requirements must be met (1) – The TE bit in the UiC1 register is set to "1" (transmission enabled) – The TI bit in the UiC1 register is set to "0" (data present in UiTB register) – If $\overline{CTS}$ function is selected, input on the $\overline{CTS}$i pin is “L”
Reception start condition	• Before reception can start, the following requirements must be met (1) – The RE bit in the UiC1 register is set to "1" (reception enabled) – The TE bit in the UiC1 register is set to "1" (transmission enabled) – The TI bit in the UiC1 register is set to "0" (data present in the UiTB register)
	• For transmission, one of the following conditions can be selected – The UiIRS bit (3) is set to "0" (transmit buffer empty): when transferring data from the UiTB register to the UARTi transmit register (at start of transmission) – The UiIRS bit is set to "1" (transfer completed): when the serial I/O finished sending data from the UARTi transmit register • For reception When transferring data from the UARTi receive register to the UiRB register (at completion of reception)
Error detection	• Overrun error (2) This error occurs if the serial I/O started receiving the next data before reading the UiRB register and received the 7th bit of the next data
Select function	• CLK polarity selection Transfer data input/output can be chosen to occur synchronously with the rising or the falling edge of the transfer clock • LSB first, MSB first selection Whether to start sending/receiving data beginning with bit 0 or beginning with bit 7 can be selected • Continuous receive mode selection Reception is enabled immediately by reading the UiRB register • Switching serial data logic (UART2) This function reverses the logic value of the transmit/receive data • Transfer clock output from multiple pins selection (UART1) The output pin can be selected in a program from two UART1 transfer clock pins that have been set • Separate $\overline{CTS}/\overline{RTS}$ pins (UART0) $\overline{CTS_0}$ and $\overline{RTS_0}$ are input/output from separate pins • UART1 pin remapping selection The UART1 pin can be selected from the $P6_7$ to $P6_4$ or $P7_3$ to $P7_0$.

NOTES:
1. When an external clock is selected, the conditions must be met while if the CKPOL bit in the UiC0 register “0” (transmit data output at the falling edge and the receive data taken in at the rising edge of the transfer clock), the external clock is in the high state; if the CKPOL bit in the UiC0 register “1” (transmit data output at the rising edge and the receive data taken in at the falling edge of the transfer clock), the external clock is in the low state.
2. If an overrun error occurs, bits 8 to 0 in UiRB register are undefined. The IR bit in the SiRIC register remains unchanged.
3. The U0IRS and U1IRS bits respectively are the UCON register bits 0 and 1; the U2IRS bit is the U2C1 register bit 4.

Downloaded from Arrow.com.

Table 13.1.1. 2. Registers to Be Used and Settings in Clock Synchronous Serial I/O Mode

Register	Bit	Function
UiTB(3)	0 to 7	Set transmission data
UiRB(3)	0 to 7	Reception data can be read
	OER	Overrun error flag
UiBRG	0 to 7	Set a transfer rate
UiMR(3)	SMD2 to SMD0	Set to “001_2”
	CKDIR	Select the internal clock or external clock
	IOPOL(i=2)(4)	Set to “0”
UiC0	CLK1 to CLK0	Select the count source for the UiBRG register
	CRS	Select $\overline{CTS}$ or $\overline{RTS}$ to use
	TXEPT	Transmit register empty flag
	CRD	Enable or disable the $\overline{CTS}$ or $\overline{RTS}$ function
	NCH	Select TxDi pin output mode
	CKPOL	Select the transfer clock polarity
	UFORM	Select the LSB first or MSB first
UiC1	TE	Set this bit to “1” to enable transmission/reception
	TI	Transmit buffer empty flag
	RE	Set this bit to “1” to enable reception
	RI	Reception complete flag
	U2IRS (1)	Select the source of UART2 transmit interrupt
	U2RRM (1)	Set this bit to “1” to use UART2 continuous receive mode
	U2LCH(3)	Set this bit to “1” to use UART2 inverted data logic
	U2ERE(3)	Set to “0”
U2SMR	0 to 7	Set to “0”
U2SMR2	0 to 7	Set to “0”
U2SMR3	0 to 2	Set to “0”
	NODC	Select clock output mode
	4 to 7	Set to “0”
U2SMR4	0 to 7	Set to “0”
UCON	U0IRS, U1IRS	Select the source of UART0/UART1 transmit interrupt
	U0RRM, U1RRM	Set this bit to “1” to use continuous receive mode
	CLKMD0	Select the transfer clock output pin when CLKMD1 = 1
	CLKMD1	Set this bit to “1” to output UART1 transfer clock from two pins
	RCSP	Set this bit to “1” to accept as input the UART0 $\overline{CTS_0}$ signal from the $P6_4$ pin or $P7_0$ pin
	7	Set to “0”

NOTES:

1. Set bit 4 and bit 5 in the U0C1 and U1C1 register are set to “0”. The U0IRS, U1IRS, U0RRM and U1RRM bits are in the UCON register.
2. Not all register bits are described above. Set those bits to “0” when writing to the registers in clock synchronous serial I/O mode.
3. Set the bit 6 and bit 7 in the U0C1 and U1C1 register to "0".
4. Set the bit 7 in the U0MR and U1MR register to "0".

i=0 to 2

Downloaded from Arrow.com.

Table 13.1.1.3 lists the functions of the input/output pins during clock synchronous serial I/O mode. Table 13.3 shows pin functions for the case where the multiple transfer clock output pin select function is deselected. Table 13.1.1.4 lists the P6$_4$ pin functions during clock synchronous serial I/O mode.

Note that for a period from when the UARTi operation mode is selected to when transfer starts, the TxDi pin outputs an “H”. (If the N-channel open-drain output is selected, this pin is in a high-impedance state.)

Table 13.1.1.3. Pin Functions(1) (When Not Select Multiple Transfer Clock Output Pin Function)

Pin name	Function	Method of selection
TxDi (i = 0 to 2) (P6$_3$, P6$_7$, P7$_0$)	Serial data output	(Outputs dummy data when performing reception only)
RxDi (P6$_2$, P6$_6$, P7$_1$)	Serial data input	Set the PD6_2 bit and PD6_6 bit in the PD6 register, and PD7_1 bit in the PD7 register to "0"(Can be used as an input port when performing transmission only)
CLKi (P6$_1$, P6$_5$, P7$_2$)	Transfer clock output	Set the CKDIR bit in the UiMR register to "0"
	Transfer clock input	Set the CKDIR bit in the UiMR register to "1" Set the PD6_1 bit and PD6_5 bit in the PD6 register, and the PD7_2 bit in the PD7 register to "0"
$\overline{\text{CTS}}$i/$\overline{\text{RTS}}$i (P6$_0$, P6$_4$, P7$_3$)	$\overline{\text{CTS}}$ input	Set the CRD bit in the UiC0 register to "0" Set the CRS bit in the UiC0 register to "0" Set the PD6_0 bit and PD6_4 bit in the PD6 register' is set to "0", the PD7_3 bit in the PD7 register to "0"
	$\overline{\text{RTS}}$ output	Set the CRD bit in the UiC0 register to "0" Set the CRS bit in the UiC0 register to "1"
	I/O port	Set the CRD bit in the UiC0 register to "1"

NOTE:
1. When the U1MAP bit in PACR register is “1” (P7$_3$ to P7$_0$), UART1 pin is assgined to P7$_3$ to P7$_0$.

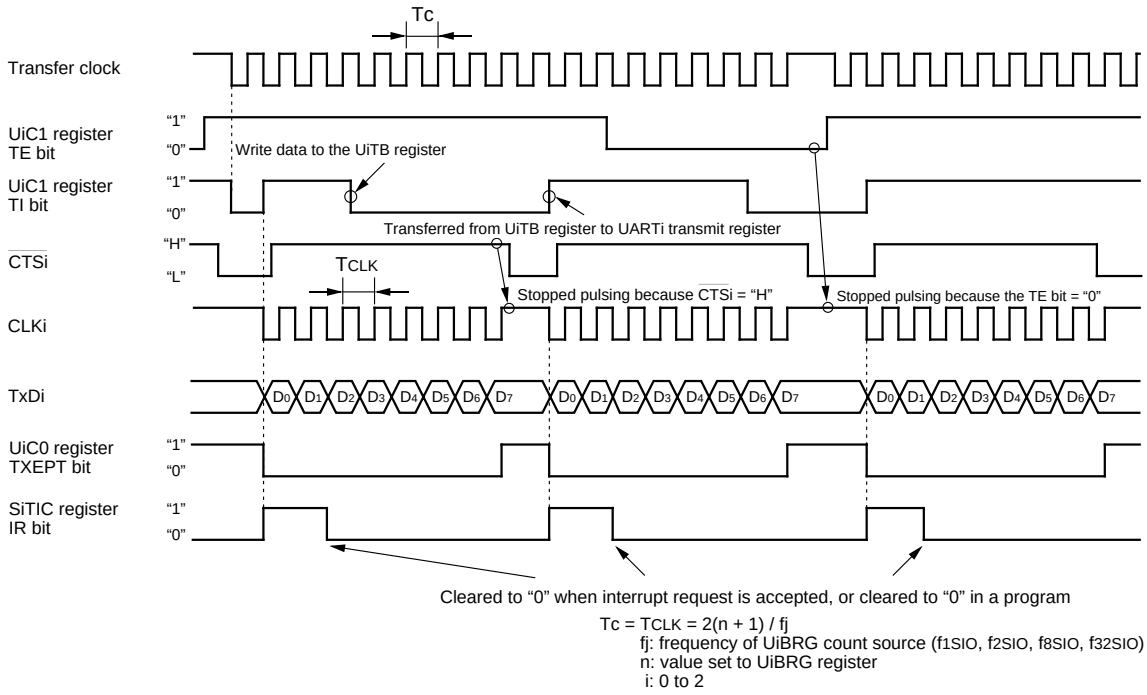
Table 13.1.1.4. P6$_4$ Pin Functions(1)

Pin function	Bit set value					
	U1C0 register		UCON register			PD6 register
	CRD	CRS	RCSP	CLKMD1	CLKMD0	PD6_4
P6$_4$	1	—	0	0	—	Input: 0, Output: 1
$\overline{\text{CTS}}_1$	0	0	0	0	—	0
$\overline{\text{RTS}}_1$	0	1	0	0	—	—
$\overline{\text{CTS}}_0$(2)	0	0	1	0	—	0
CLKS$_1$	—	—	—	1(3)	1	—

NOTES:
1. When the U1MAP bit in PACR register is “1” (P7$_3$ to P7$_0$), this table lists the P7$_0$ functions.
2. In addition to this, set the CRD bit in the U0C0 register to “0” ($\overline{\text{CT0}}_0$/$\overline{\text{RT0}}_0$ enabled) and theCRS bit in the U0C0 register to “1” ($\overline{\text{RTS}}_0$ selected).
3. When the CLKMD1 bit is set to "1" and the CLKMD0 bit is set to "0", the following logiclevels are output:
   • High if the CLKPOL bit in the U1C0 register is set to "0"
   • Low if the CLKPOL bit in the U1C0 register is set to "1"

Downloaded from Arrow.com.

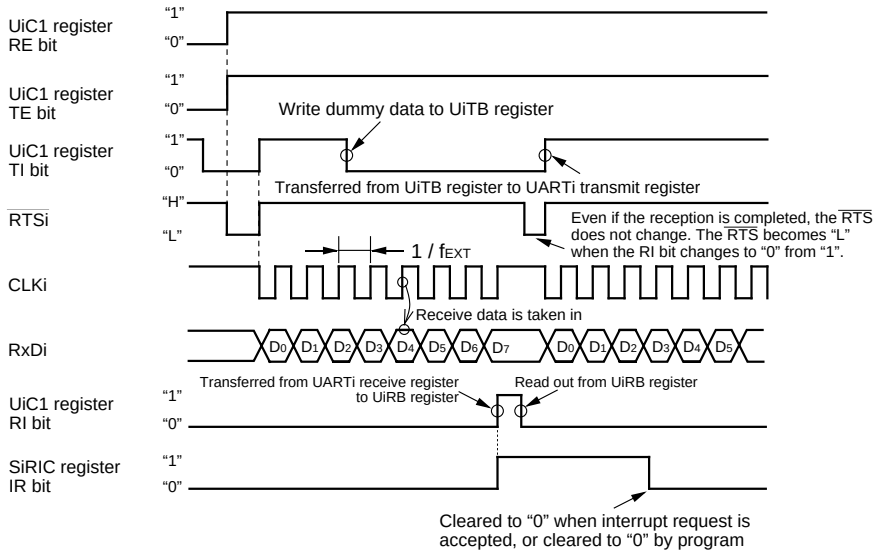
(1) Example of Transmit Timing (Internal clock is selected)



Tc = TCLK = 2(n + 1) / fj
fj: frequency of UiBRG count source (f1SIO, f2SIO, f8SIO, f32SIO)
n: value set to UiBRG register
i: 0 to 2

The above timing diagram applies to the case where the register bits are set as follows:
- The CKDIR bit in the UiMR register is set to "0" (internal clock)
- The CRD bit in the UiC0 register is set to "0" (CTS/RTS enabled); CRS bit is set to "0" (CTS selected)
- The CKPOL bit in the UiC0 register is set to "0" (transmit data output at the falling edge and receive data taken in at the rising edge of the transfer clock)
- The UiIRS bit is set to "0" (an interrupt request occurs when the transmit buffer becomes empty): U0IRS bit is the bit 0 in the UCON register U1IRS bit is the bit 1 in the UCON register, and U2IRS bit is the bit 4 in the U2C1 register.

(2) Example of Receive Timing (External clock is selected)



The above timing diagram applies to the case where the register bits are set as follows:
- The CKDIR bit in the UiMR register is set to "1" (external clock)
- The CRD bit in the UiC0 register is set to "0"(CTS/RTS enabled); The CRS bit is set to "1" (RTS selected)
- UiC0 register CKPOL bit is set to "0"(transmit data output at the falling edge and receive data taken in at the rising edge of the transfer clock)

fEXT: frequency of external clock

Make sure the following conditions are met when input to the CLKi pin before receiving data is high:
- UiC0 register TE bit is set to "1" (transmit enabled)
- UiC0 register RE bit is set to "1" (Receive enabled)
- Write dummy data to the UiTB register

Figure 13.1.1.1. Typical transmit/receive timings in clock synchronous serial I/O mode

Downloaded from Arrow.com.

13.1.1.1 Counter Measure for Communication Error Occurs

If a communication error occurs while transmitting or receiving in clock synchronous serial I/O mode, follow the procedures below.

•Resetting the UiRB register (i=0 to 2)

(1) Set the RE bit in the UiC1 register to “0” (reception disabled)

(2) Set the SMD2 to SMD0 bits in the UiMR register to “000_2” (Serial I/O disabled)

(3) Set the SMD2 to SMD0 bits in the UiMR register to “001_2” (Clock synchronous serial I/O mode)

(4) Set the RE bit in the UiC1 register to “1” (reception enabled)

•Resetting the UiTB register (i=0 to 2)

(1) Set the SMD2 to SMD0 bits in the UiMR register to “000_2” (Serial I/O disabled)

(2) Set the SMD2 to SMD0 bits in the UiMR register to “001_2” (Clock synchronous serial I/O mode)

(3) “1” is written to RE bit in the UiC1 register (reception enabled), regardless to the TE bit in the UiC1 register.

13.1.1.2 CLK Polarity Select Function

Use the CKPOL bit in the UiC0 register (i = 0 to 2) to select the transfer clock polarity. Figure 13.1.1.2.1 shows the polarity of the transfer clock.

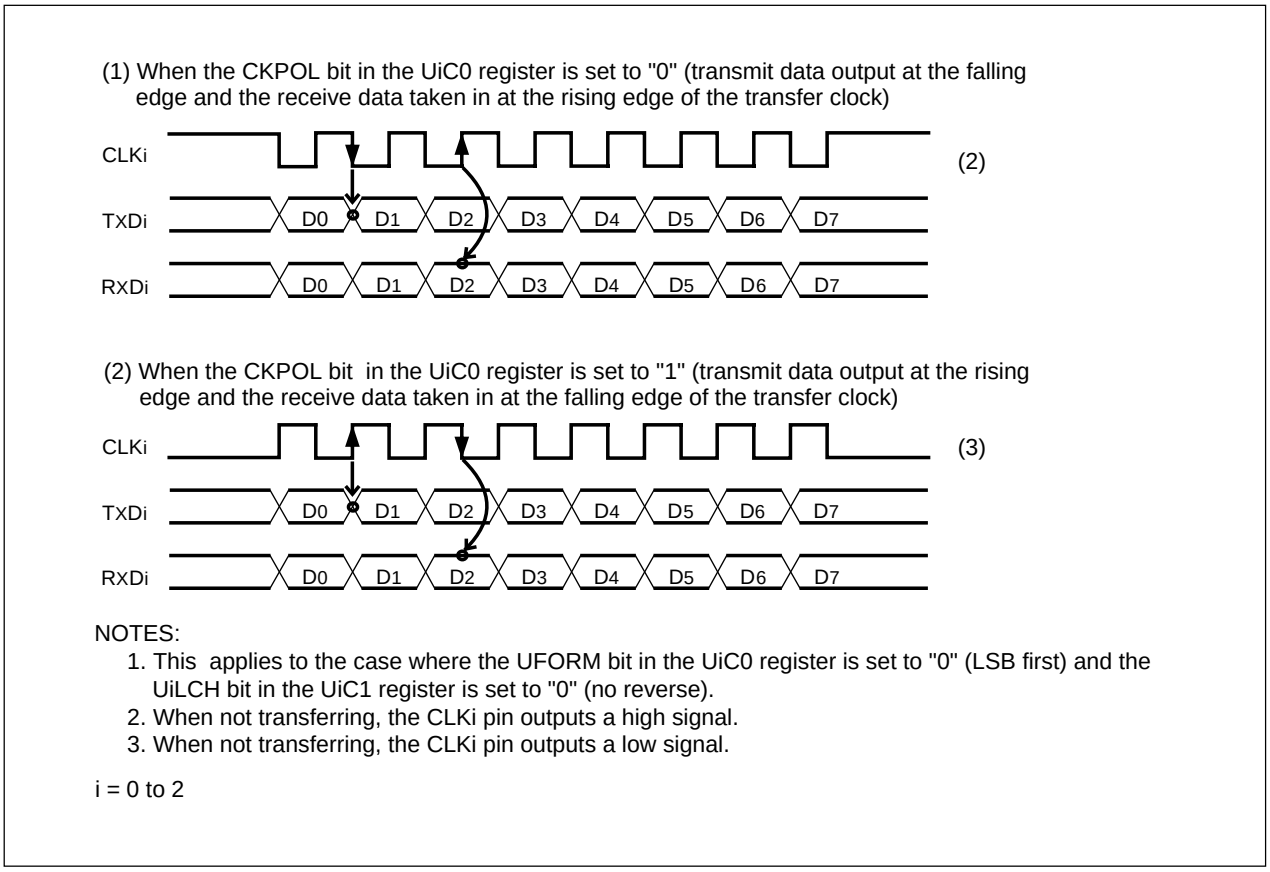
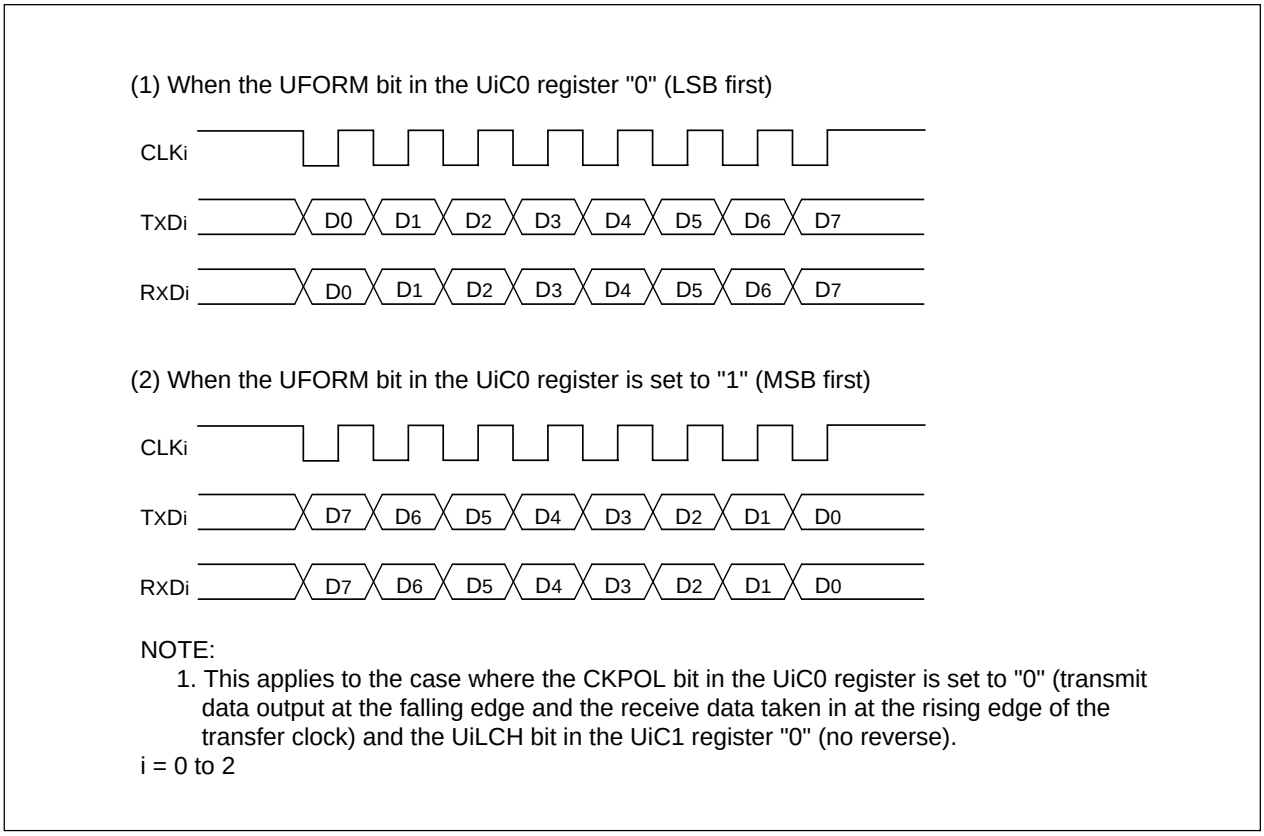


Figure 13.1.1.2.1. Polarity of transfer clock

Downloaded from Arrow.com.

13.1.1.3 LSB First/MSB First Select Function

Use the UFORM bit in the UiC0 register (i = 0 to 2) to select the transfer format. Figure 13.1.1.3.1 shows the transfer format.



NOTE:
1. This applies to the case where the CKPOL bit in the UiC0 register is set to "0" (transmit data output at the falling edge and the receive data taken in at the rising edge of the transfer clock) and the UiLCH bit in the UiC1 register "0" (no reverse).

i = 0 to 2

Figure 13.1.1.3.1 Transfer format

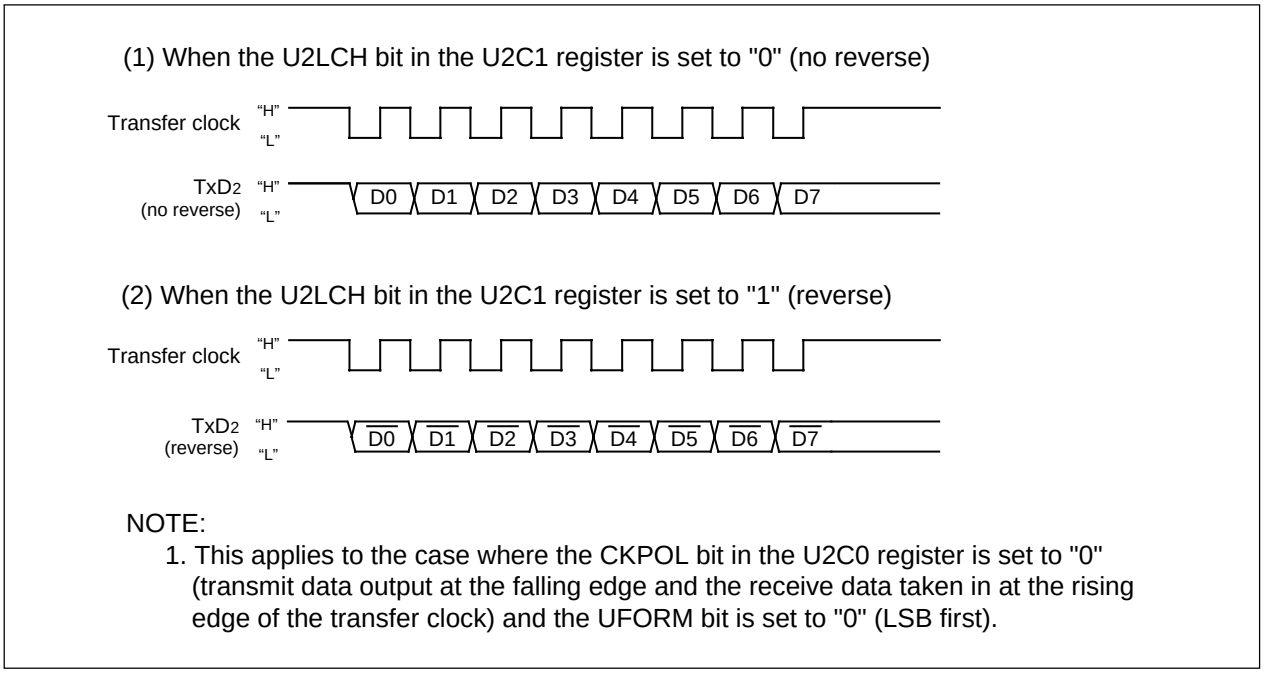
13.1.1.4 Continuous receive mode

When the UiRRM bit (i = 0 to 2) is set to "1" (continuous receive mode), the TI bit in the UiC1 register is set to “0” (data present in the UiTB register) by reading the UiRB register. In this case, i.e., UiRRM bit is set to "1", do not write dummy data to the UiTB register in a program. The U0RRM and U1RRM bits are the bit 2 and bit 3 in the UCON register, respectively, and the U2RRM bit is the bit 5 in the U2C1 register.

Downloaded from Arrow.com.

13.1.1.5 Serial data logic switch function (UART2)

When the U2LCH bit in the U2C1 register is set to "1" (reverse), the data written to the U2TB register has its logic reversed before being transmitted. Similarly, the received data has its logic reversed when read from the U2RB register. Figure 13.1.1.4.1 shows serial data logic.

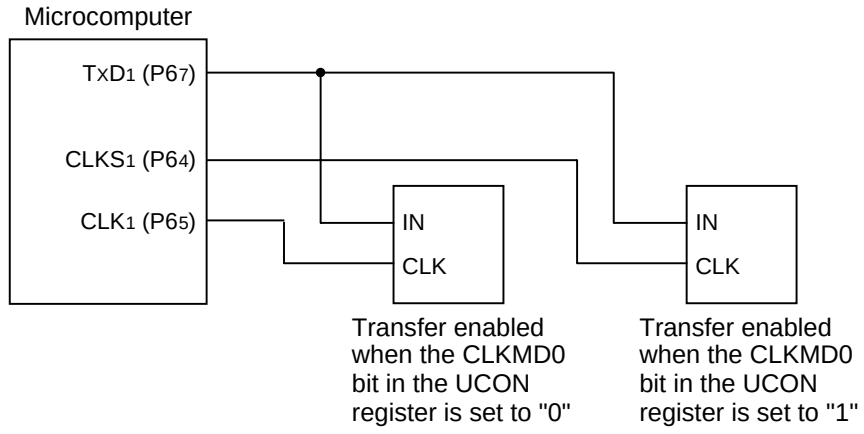


NOTE:
1. This applies to the case where the CKPOL bit in the U2C0 register is set to "0" (transmit data output at the falling edge and the receive data taken in at the rising edge of the transfer clock) and the UFORM bit is set to "0" (LSB first).

Figure 13.1.1.4.1. Serial data logic switch timing

13.1.1.6 Transfer clock output from multiple pins function (UART1)

The CLKMD1 to CLKMD0 bits in the UCON register can choose one from two transfer clock output pins. (See Figure 13.1.1.6.1) This function is valid when the internal clock is selected for UART1.



NOTES:
1. This applies to the case where the CKDIR bit in the U1MRregister is set to "0" (internal clock) and the CLKMD1 bit in the UCON register is set to "1" (transfer clock output from multiple pins).
2. This applies to the case where U1MAP bit in PACR register is set to "0" (P67 to P64).

Figure 13.1.1.6.1 Transfer Clock Output From Multiple Pins

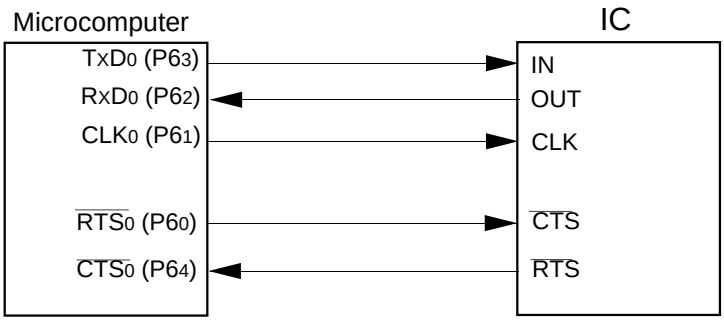
Downloaded from Arrow.com.

13.1.1.7 $\overline{CTS}/\overline{RTS}$ separate function (UART0)

This function separates $\overline{CTS}_0/\overline{RTS}_0$, outputs $\overline{RTS}_0$ from the $P6_0$ pin, and accepts as input the $\overline{CTS}_0$ from the $P6_4$ pin. To use this function, set the register bits as shown below.

- The CRD bit in the U0C0 register is set to "0" (enables UART0 $\overline{CTS}/\overline{RTS}$)
- The CRS bit in the U0C0 register is set to "1" (outputs UART0 $\overline{RTS}$)
- The CRD bit in the U1C0 register is set to "0" (enables UART1 $\overline{CTS}/\overline{RTS}$)
- The CRS bit in the U1C0 register is set to "0" (inputs UART1 $\overline{CTS}$)
- The RCSP bit in the UCON register is set to "1" (inputs $\overline{CTS}_0$ from the $P6_4$ pin)
- The CLKMD1 bit in the UCON register is set to "0" ($CLKS_1$ not used)

Note that when using the $\overline{CTS}/\overline{RTS}$ separate function, UART1 $\overline{CTS}/\overline{RTS}$ separate function cannot be used.



NOTE:

1. This applies to the case where U1MAP bit in PACR register is set to “0” ($P6_7$ to $P6_4$).

Figure 13.1.1.7.1. CTS/RTS separate function usage

Downloaded from Arrow.com.

13.1.2. Clock Asynchronous Serial I/O (UART) Mode

The UART mode allows transmitting and receiving data after setting the desired transfer rate and transfer data format. Tables 13.1.2.1 lists the specifications of the UART mode.

Table 13.1.2.1. UART Mode Specifications

Item	Specification
Transfer data format	• Character bit (transfer data): Selectable from 7, 8 or 9 bits • Start bit: 1 bit • Parity bit: Selectable from odd, even, or none • Stop bit: Selectable from 1 or 2 bits
Transfer clock	• The CKDIR bit in the UiMR(i=0 to 2) register is set to "0" (internal clock) : fj/(16(n+1)) fj = f_{1SIO}, f_{2SIO}, f_{8SIO}, f_{32SIO}. n: Setting value of UiBRG register 00_{16} to FF_{16} • CKDIR bit is set to "1" (external clock) : f_{EXT}/(16(n+1)) f_{EXT}: Input from CLKi pin. n :Setting value of UiBRG register 00_{16} to FF_{16}
Transmission, reception control	• Selectable from $\overline{CTS}$ function, $\overline{RTS}$ function or $\overline{CTS}/\overline{RTS}$ function disable
Transmission start condition	• Before transmission can start, the following requirements must be met – The TE bit in the UiC1 register is set to "1" (transmission enabled) – The TI bit in the UiC1 register "0" (data present in UiTB register) – If $\overline{CTS}$ function is selected, input "L" to the $\overline{CTS}$i pin
Reception start condition	• Before reception can start, the following requirements must be met – The RE bit in the UiC1 register is set to "1" (reception enabled) – Start bit detection
Interrupt request generation timing data from	• For transmission, one of the following conditions can be selected – The UiIRS bit (2) is set to "0" (transmit buffer empty): when transferring data from the UiTB register to the UARTi transmit register (at start of transmission) – The UiIRS bit is set to "1" (transfer completed): when the serial I/O finished sending the UARTi transmit register • For reception When transferring data from the UARTi receive register to the UiRB register (at completion of reception)
Error detection	• Overrun error (1) This error occurs if the serial I/O started receiving the next data before reading the UiRB register and received the bit one before the last stop bit of the next data • Framing error This error occurs when the number of stop bits set is not detected • Parity error This error occurs when if parity is enabled, the number of 1's in parity and character bits does not match the number of 1's set • Error sum flag This flag is set (= 1) when any of the overrun, framing, and parity errors is encountered
Select function	• LSB first, MSB first selection Whether to start sending/receiving data beginning with bit 0 or beginning with bit 7 can be selected • Serial data logic switch (UART2) This function reverses the logic of the transmit/receive data. The start and stop bits are not reversed. • TxD, RxD I/O polarity switch (UART2) This function reverses the polarities of hte TxD pin output and RxD pin input. The logic levels of all I/O data is reversed. • Separate $\overline{CTS}/\overline{RTS}$ pins (UART0) $\overline{CTS_0}$ and $\overline{RTS_0}$ are input/output from separate pins • UART1 pin remapping selection The UART1 pin can be selected from the $P6_7$ to $P6_4$ or $P7_3$ to $P7_0$.

NOTES:

1. If an overrun error occurs, bits 8 to 0 in UiRB register are undefined. The IR bit in the SiRIC register remains unchanged.
2. The U0IRS and U1IRS bits respectively are the bits "0" and "1" in the UCON register; the U2IRS bit is the bit 4 in the U2C1 register.

Downloaded from Arrow.com.

Table 13.1.2.2. Registers to Be Used and Settings in UART Mode

Register	Bit	Function
UiTB	0 to 8	Set transmission data (1)
UiRB	0 to 8	Reception data can be read (1)
	OER,FER,PER,SUM	Error flag
UiBRG	0 to 7	Set a transfer rate
UiMR	SMD2 to SMD0	Set these bits to '100$_2$' when transfer data is 7 bits long Set these bits to '101$_2$' when transfer data is 8 bits long Set these bits to '110$_2$' when transfer data is 9 bits long
	CKDIR	Select the internal clock or external clock
	STPS	Select the stop bit
	PRY, PRYE	Select whether parity is included and whether odd or even
	IOPOL(i=2)(4)	Select the TxD/RxD input/output polarity
UiC0	CLK0, CLK1	Select the count source for the UiBRG register
	CRS	Select $\overline{CTS}$ or $\overline{RTS}$ to use
	TXEPT	Transmit register empty flag
	CRD	Enable or disable the $\overline{CTS}$ or $\overline{RTS}$ function
	NCH	Select TxDi pin output mode
	CKPOL	Set to "0"
	UFORM	LSB first or MSB first can be selected when transfer data is 8 bits long. Set this bit to "0" when transfer data is 7 or 9 bits long.
UiC1	TE	Set this bit to "1" to enable transmission
	TI	Transmit buffer empty flag
	RE	Set this bit to "1" to enable reception
	RI	Reception complete flag
	U2IRS (2)	Select the source of UART2 transmit interrupt
	U2RRM (2)	Set to "0"
	U2LCH (3)	Set this bit to "1" to use UART2 inverted data logic
	U2ERE (3)	Set to "0"
U2SMR	0 to 7	Set to "0"
U2SMR2	0 to 7	Set to "0"
U2SMR3	0 to 7	Set to "0"
U2SMR4	0 to 7	Set to "0"
UCON	U0IRS, U1IRS	Select the source of UART0/UART1 transmit interrupt
	U0RRM, U1RRM	Set to "0"
	CLKMD0	Invalid because CLKMD1 = 0
	CLKMD1	Set to "0"
	RCSP	Set this bit to "1" to accept as input the UART0 $\overline{CTS_0}$ signal from the $P6_4$ pin or $P7_0$ pin
	7	Set to "0"

NOTES:

1. The bits used for transmit/receive data are as follows: Bit 0 to bit 6 when transfer data is 7 bits long; bit 0 to bit 7 when transfer data is 8 bits long; bit 0 to bit 8 when transfer data is 9 bits long.
2. Set the bit 4 to bit 5 in the U0C1 and U1C1 registers to "0". The U0IRS, U1IRS, U0RRM and U1RRM bits are included in the UCON register.
3. Set the bit 6 to bit 7 in the U0C1 and U1C1 registers to "0".
4. Set the bit 7 the U0MR and U1MR registers to "0".

i=0 to 2

Downloaded from Arrow.com.

Table 13.1.2.3 lists the functions of the input/output pins during UART mode. Table 13.1.2.4 lists the P6$_4$ pin functions during UART mode. Note that for a period from when the UARTi operation mode is selected to when transfer starts, the TxDi pin outputs an “H”. (If the N-channel open-drain output is selected, this pin is in a high-impedance state.)

Table 13.1.2.3. I/O Pin Functions in UART mode(1)

Pin name	Function	Method of selection
TxDi (i = 0 to 2) (P6$_3$, P6$_7$, P7$_0$)	Serial data output	(Outputs "H" when performing reception only)
RxDi (P6$_2$, P6$_6$, P7$_1$)	Serial data input	PD6_2 bit, PD6_6 bit in the PD6 register and the PD7_1 bit in the PD7 register (Can be used as an input port when performing transmission only)
CLKi (P6$_1$, P6$_5$, P7$_2$)	Input/output port	Set the CKDIR bit in the UiMR register to "0"
	Transfer clock input	Set the CKDIR bit in the UiMR register to "1" Set the PD6_1 bit and PD6_5 bit in the PD6 register to "0", PD7_2 bit in the PD7 register to "0"
$\overline{\text{CTS}}$i/$\overline{\text{RTS}}$i (P6$_0$, P6$_4$, P7$_3$)	$\overline{\text{CTS}}$ input	Set the CRD bit in the UiC0 register to "0" Set the CRS bit in the UiC0 register to "0" Set the PD6_0 bit and PD6_4 bit in the PD6 register to "0", the PD7_3 bit in the PD7 register "0"
	$\overline{\text{RTS}}$ output	Set the CRD bit in the UiC0 register to "0" Set the CRS bit in the UiC0 register to "1"
	Input/output port	Set the CRD bit in the UiC0 register "1"

NOTE:
1. When the U1MAP bit in PACR register is set to “1” (P7$_3$ to P7$_0$), UART1 pin is assgined to P7$_3$ to P7$_0$.

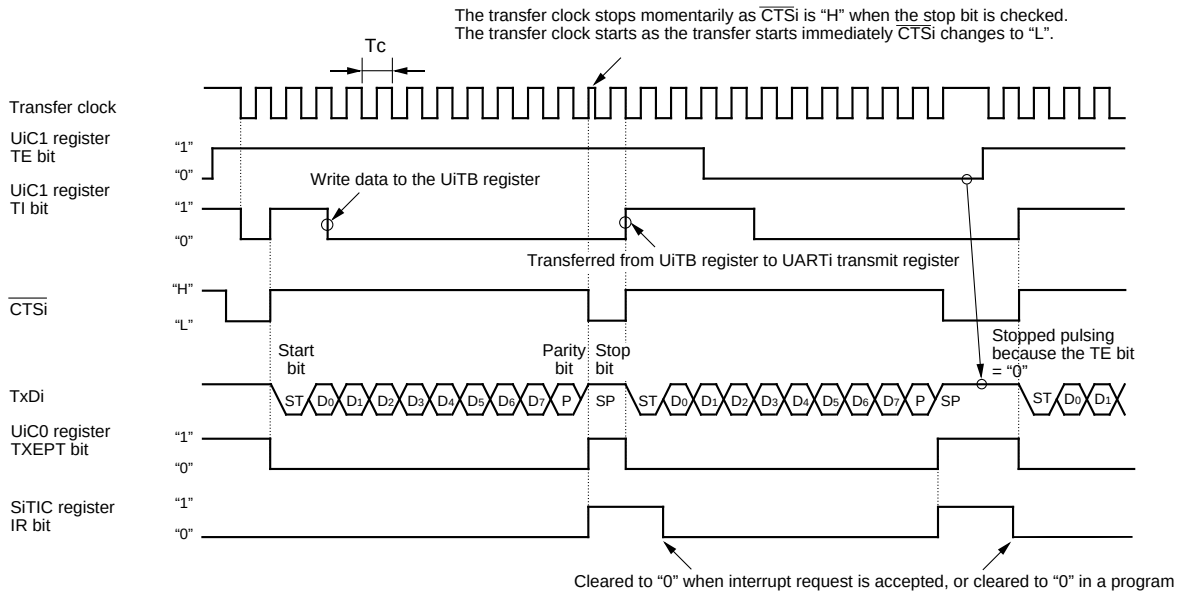
Table 13.1.2.4. P6$_4$ Pin Functions in UART mode(1)

Pin function	Bit set value				
	U1C0 register		UCON register		PD6 register
	CRD	CRS	RCSP	CLKMD1	PD6_4
P6$_4$	1	—	0	0	Input: 0, Output: 1
$\overline{\text{CTS}}_1$	0	0	0	0	0
$\overline{\text{RTS}}_1$	0	1	0	0	—
$\overline{\text{CTS}}_0$ (2)	0	0	1	0	0

NOTES:
1. When the U1MAP bit in PACR register is “1” (P7$_3$ to P7$_0$), this table lists the P7$_0$ functions.
2. In addition to this, set the CRD bit in the U0C0 register to “0” ($\overline{\text{CTS}}_0$/$\overline{\text{RTS}}_0$ enabled) and the CRS bit in the U0C0 register to “1” ($\overline{\text{RTS}}_0$ selected).

Downloaded from Arrow.com.

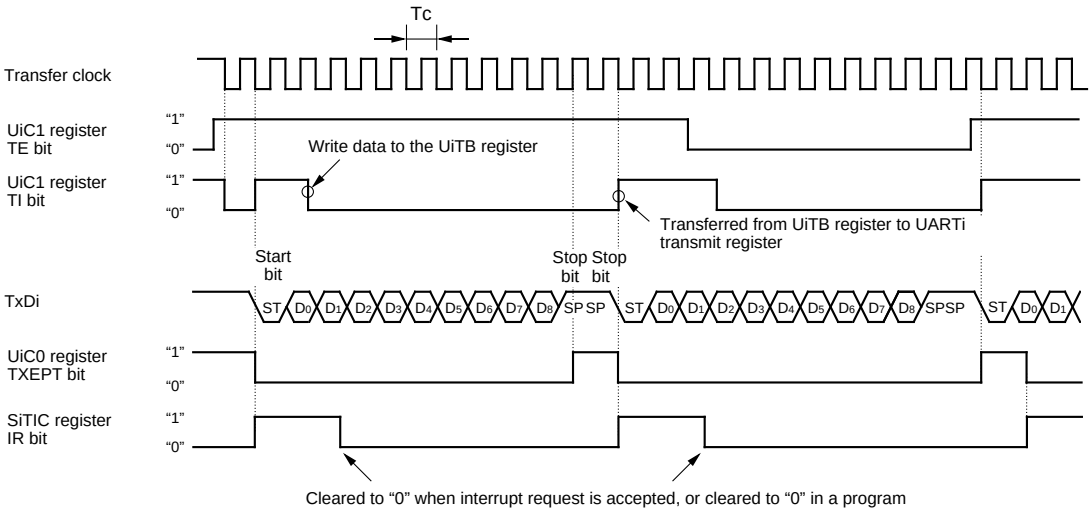
• Example of transmit timing when transfer data is 8 bits long (parity enabled, one stop bit)



The above timing diagram applies to the case where the register bits are set as follows:
- Set the PRYE bit in the UiMR register to "1" (parity enabled)
- Set the STPS bit in the UiMR register to "0" (1 stop bit)
- Set the CRD bit in the UiC0 register to "0" (CTS/RTS enabled), the CRS bit to "0" (CTS selected)
- Set the UiIRS bit to "1" (an interrupt request occurs when transmit completed): U0IRS bit is the UCON register bit 0, U1IRS bit is the UCON register bit 1, and U2IRS bit is the U2C1 register bit 4

$Tc = 16 (n + 1) / fj$ or $16 (n + 1) / f_{EXT}$
fj : frequency of UiBRG count source (f_{1SIO}, f_{2SIO}, f_{8SIO}, f_{32SIO})
f_{EXT} : frequency of UiBRG count source (external clock)
n : value set to UiBRG
i: 0 to 2

• Example of transmit timing when transfer data is 9 bits long (parity disabled, two stop bits)



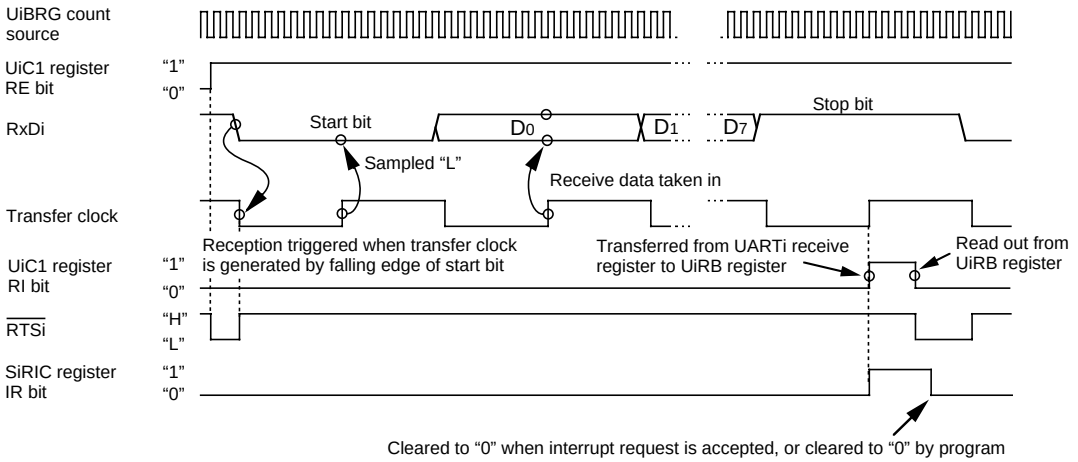
The above timing diagram applies to the case where the register bits are set as follows:
- Set the PRYE bit in the UiMR register to "0" (parity disabled)
- Set the STPS bit in the UiMR register to "1" (2 stop bits)
- Set the CRD bit in the UiC0 register to "1"(CTS/RTS disabled)
- Set the UiIRS bit to "0" (an interrupt request occurs when transmit buffer becomes empty): U0IRS bit is the UCON register bit 0, U1IRS bit is the UCON register bit 1, and U2IRS bit is the U2C1 register bit 4

$Tc = 16 (n + 1) / fj$ or $16 (n + 1) / f_{EXT}$
fj : frequency of UiBRG count source (f_{1SIO}, f_{2SIO}, f_{8SIO}, f_{32SIO})
f_{EXT} : frequency of UiBRG count source (external clock)
n : value set to UiBRG
i: 0 to 2

Figure 13.1.2.1. Typical transmit timing in UART mode (UART0, UART1)

Downloaded from Arrow.com.

• Example of receive timing when transfer data is 8 bits long (parity disabled, one stop bit)



The above timing diagram applies to the case where the register bits are set as follows:
- Set the PRYE bit in the UiMR register to "0"(parity disabled)
- Set the STPS bit in the UiMR register to "0" (1 stop bit)
- Set the CRD bit in the UiC0 register to "0" (CTSi/RTSi enabled), the CRS bit to "1" (RTSi selected)

i = 0 to 2

Figure 13.1.2.2. Receive Operation

13.1.2.1. Bit Rates

In UART mode, the frequency set by the UiBRG register (i=0 to 2) divided by 16 become the bit rates. Table 13.1.2.1.1 lists example of bit rate and settings.

Table 13.1.2.1.1 Example of Bit Rates and Settings

Bit Rate (bps)	Count Source of BRG	Peripheral Function Clock : 16MHz		Peripheral Function Clock : 20MHz	
		Set Value of BRG : n	Actual Time (bps)	Set Value of BRG : n	Actual Time (bps)
1200	f8	103(67h)	1202	129(81h)	1202
2400	f8	51(33h)	2404	64(40h)	2404
4800	f8	25(19h)	4808	32(20h)	4735
9600	f1	103(67h)	9615	129(81h)	9615
14400	f1	68(44h)	14493	86(56h)	14368
19200	f1	51(33h)	19231	64(40h)	19231
28800	f1	34(22h)	28571	42(2Ah)	29070
31250	f1	31(1Fh)	31250	39(27h)	31250
38400	f1	25(19h)	38462	32(20h)	37879
51200	f1	19(13h)	50000	24(18h)	50000

Downloaded from Arrow.com.

13.1.2.2. Counter Measure for Communication Error

If a communication error occurs while transmitting or receiving in UART mode, follow the procedure below.

• Resetting the UiRB register (i=0 to 2)

(1) Set the RE bit in the UiC1 register to “0” (reception disabled)

(2) Set the RE bit in the UiC1 register to “1” (reception enabled)

• Resetting the UiTB register (i=0 to 2)

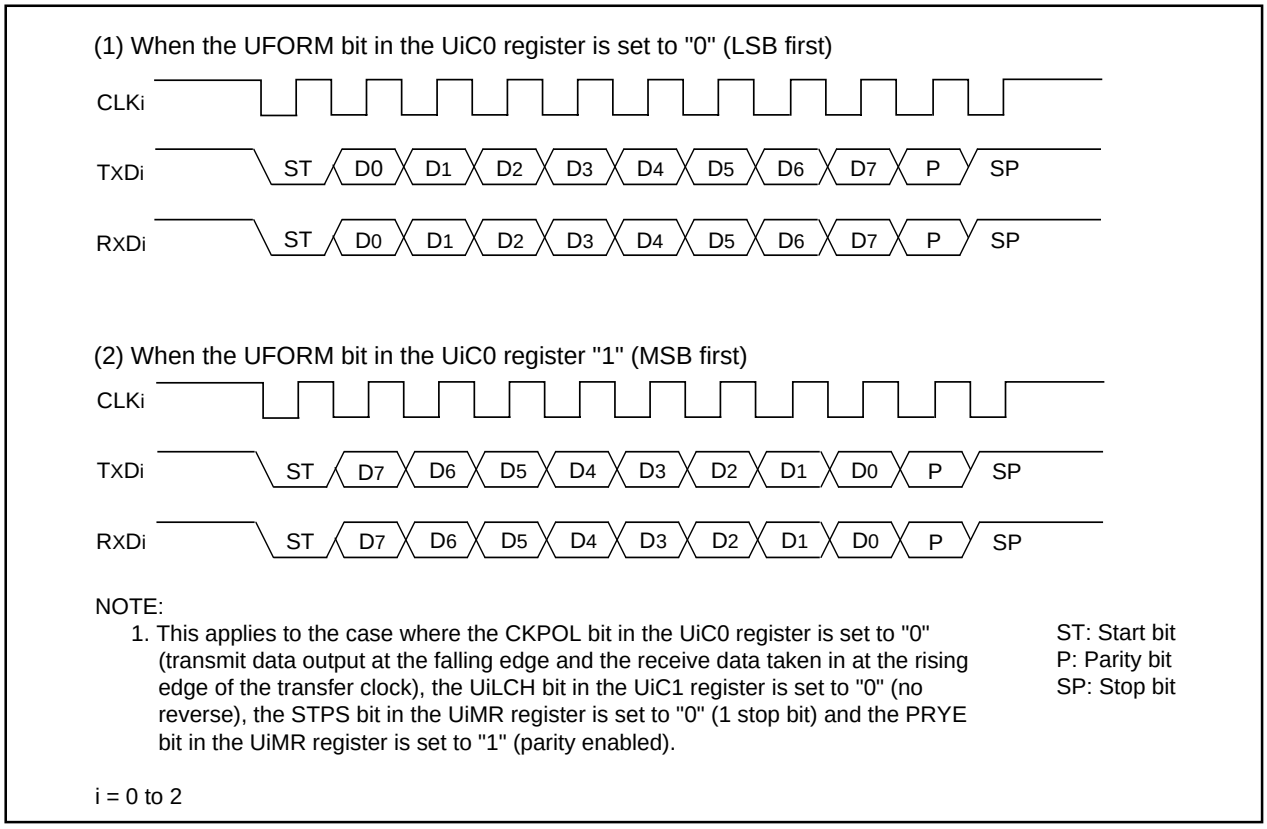
(1) Set the SMD2 to SMD0 bits in UiMR register “000_2” (Serial I/O disabled)

(2) Set the SMD2 to SMD0 bits in UiMR register “001_2”, “101_2”, “110_2”

(3) “1” is written to RE bit in the UiC1 register (reception enabled), regardless of the TE bit in the UiC1 register

13.1.2.3. LSB First/MSB First Select Function

As shown in Figure 14.1.2.3.1, use the UFORM bit in the UiC0 register to select the transfer format. This function is valid when transfer data is 8 bits long.



NOTE:

1. This applies to the case where the CKPOL bit in the UiC0 register is set to "0" (transmit data output at the falling edge and the receive data taken in at the rising edge of the transfer clock), the UiLCH bit in the UiC1 register is set to "0" (no reverse), the STPS bit in the UiMR register is set to "0" (1 stop bit) and the PRYE bit in the UiMR register is set to "1" (parity enabled).

ST: Start bit
P: Parity bit
SP: Stop bit

i = 0 to 2

Figure 13.1.2.3.1. Transfer Format

Downloaded from Arrow.com.

13.1.2.4. Serial Data Logic Switching Function (UART2)

The data written to the U2TB register has its logic reversed before being transmitted. Similarly, the received data has its logic reversed when read from the U2RB register. Figure 13.1.2.4.1 shows serial data logic.

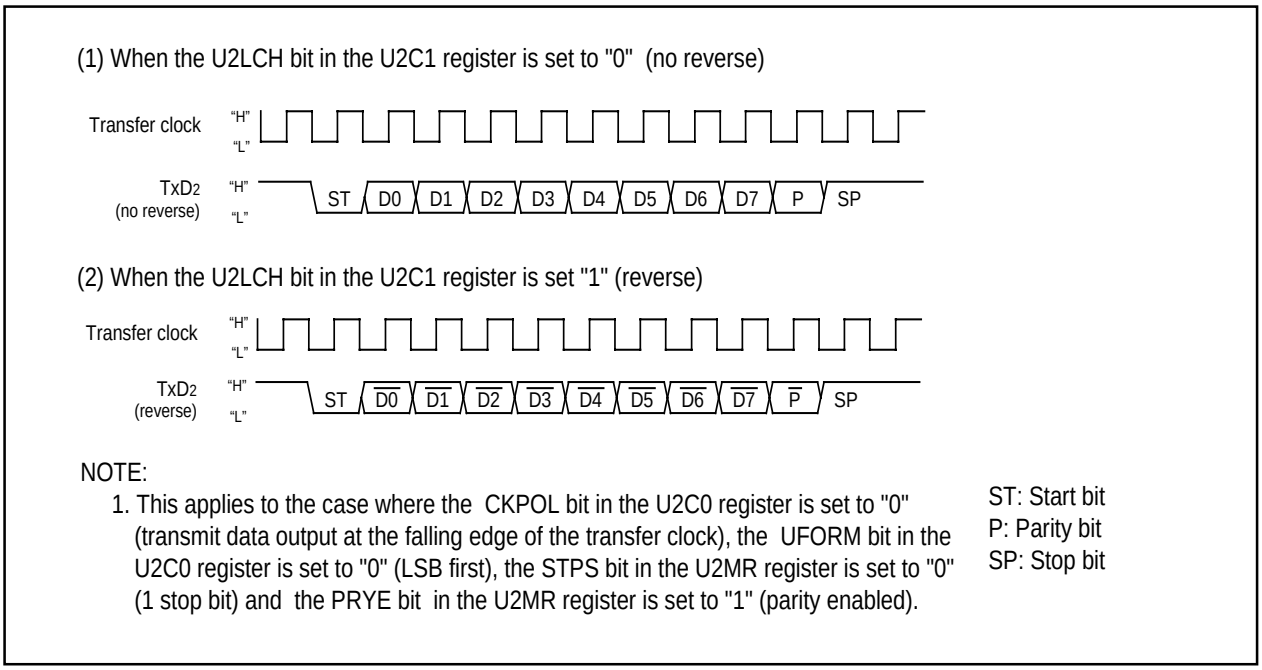


Figure 13.1.2.4.1. Serial Data Logic Switching

13.1.2.5. TxD and RxD I/O Polarity Inverse Function (UART2)

This function inverses the polarities of the TxD2 pin output and RxD2 pin input. The logic levels of all input/output data (including the start, stop and parity bits) are inversed. Figure 13.1.2.5.1 shows the TxD pin output and RxD pin input polarity inverse.

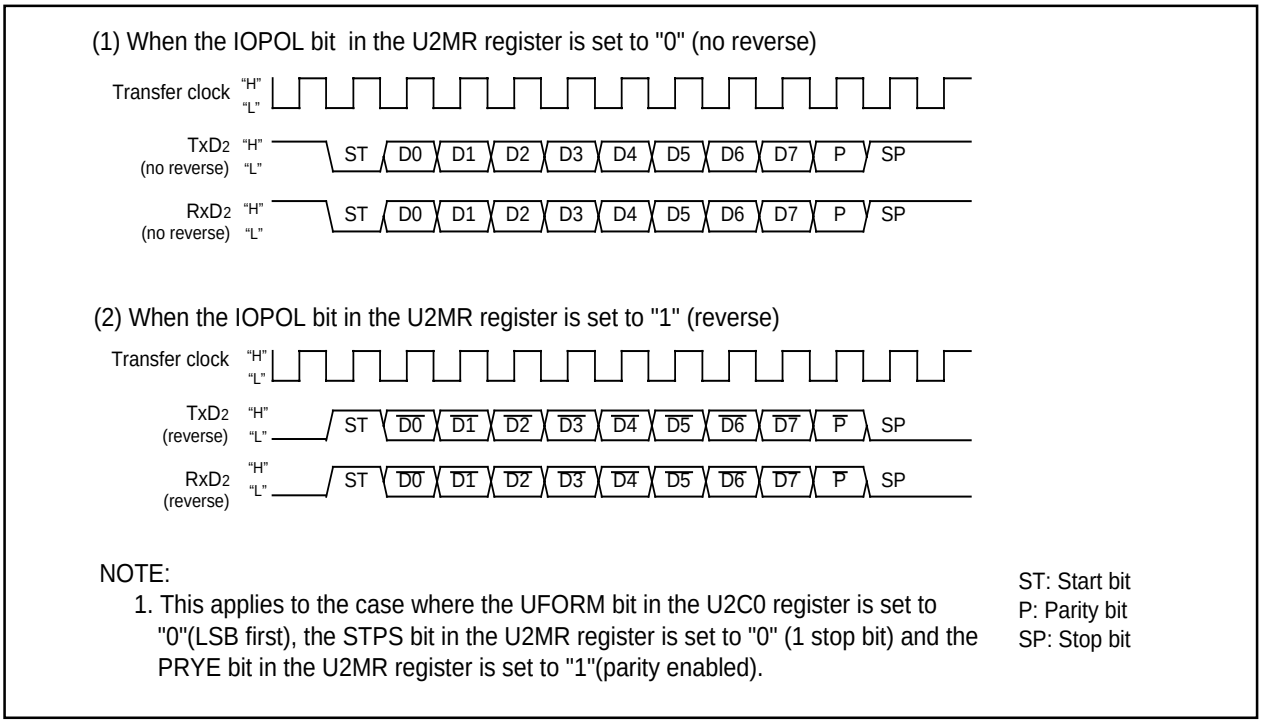


Figure 13.1.2.5.1. TxD and RxD I/O Polarity Inverse

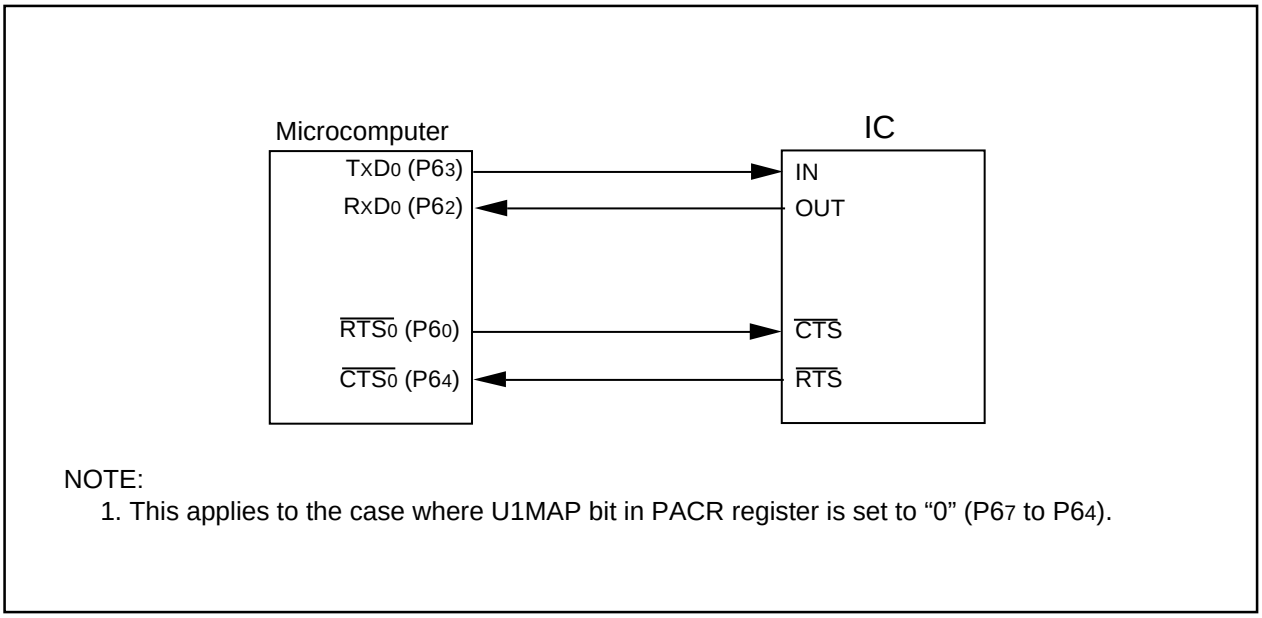
Downloaded from Arrow.com.

13.1.2.6. $\overline{CTS}/\overline{RTS}$ Separate Function (UART0)

This function separates $\overline{CTS}_0/\overline{RTS}_0$, outputs $\overline{RTS}_0$ from the $P6_0$ pin, and accepts as input the $\overline{CTS}_0$ from the $P6_4$ pin. To use this function, set the register bits as shown below.

- Set the CRD bit in the U0C0 register to "0" (enables UART0 $\overline{CTS}/\overline{RTS}$)
- Set the CRS bit in the U0C0 register to "1"(outputs UART0 $\overline{RTS}$)
- Set the CRD bit in the U1C0 register to "0" (enables UART1 $\overline{CTS}/\overline{RTS}$)
- Set the CRS bit in the U1C0 register to "0" (inputs UART1 $\overline{CTS}$)
- Set the RCSP bit in the UCON register to "1" (inputs $\overline{CTS}_0$ from the $P6_4$ pin)
- Set the CLKMD1 bit in the UCON register to "0" ($CLKS_1$ not used)

Note that when using the $\overline{CTS}/\overline{RTS}$ separate function, UART1 $\overline{CTS}/\overline{RTS}$ separate function cannot be used.



NOTE:
1. This applies to the case where U1MAP bit in PACR register is set to “0” ($P6_7$ to $P6_4$).

Figure 13.1.2.6.1. $\overline{CTS}/\overline{RTS}$ Separate Function

Downloaded from Arrow.com.

13.1.3 Special Mode 1 (I^2C bus mode)(UART2)

I^2C bus mode is provided for use as a simplified I^2C bus interface compatible mode. Table 13.1.3.1 lists the specifications of the I^2C bus mode. Table 13.1.3.2 and 13.1.3.3 list the registers used in the I^2C bus mode and the register values set. Table 13.1.3.4 lists the I^2C bus mode fuctions. Figure 13.1.3.1 shows the block diagram for I^2C bus mode. Figure 13.1.3.2 shows SCL_2 timing.

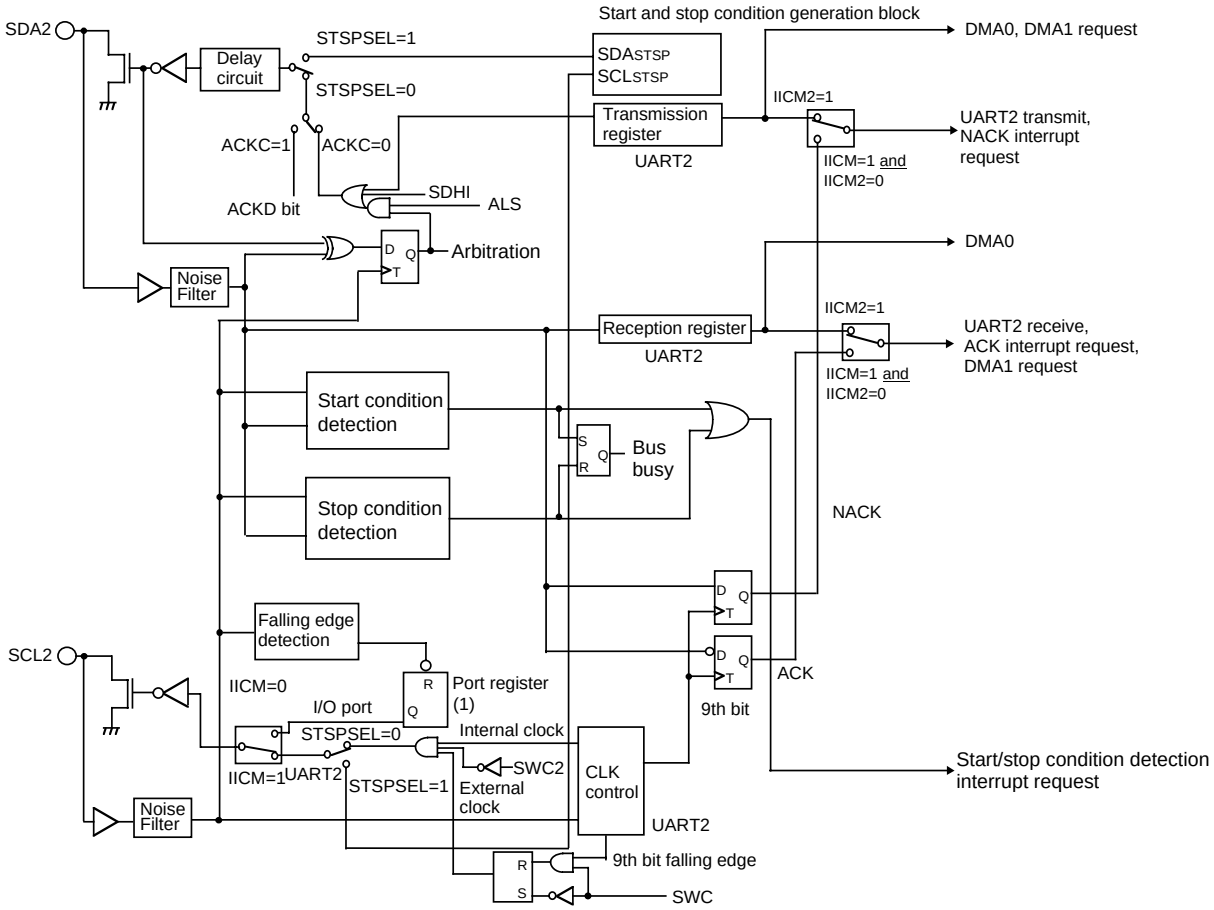
As shown in Table 13.1.3.2, the microcomputer is placed in I^2C bus mode by setting the SMD2 to SMD0 bits to '010_2' and the IICM bit to "1". Because SDA_2 transmit output has a delay circuit attached, SDA output does not change state until SCL_2 goes low and remains stably low.

Table 13.1.3.1. I^2C bus Mode Specifications

Item	Specification
Transfer data format	• Transfer data length: 8 bits
Transfer clock	• During master The CKDIR bit in the U2MR register is set to "0" (internal clock) : fj/ (2(n+1)) fj = f_{1SIO}, f_{2SIO}, f_{8SIO}, f_{32SIO}. n: Setting value in the U2BRG register 00_{16} to FF_{16} • During slave The CKDIR bit is set to "1" (external clock) : Input from SCL_2 pin
Transmission start condition	• Before transmission can start, the following requirements must be met (1) – The TE bit in the U2C1 register is set to "1" (transmission enabled) – The TI bit in the U2C1 register is set to "0" (data present in U2TB register)
Reception start condition	• Before reception can start, the following requirements must be met (1) – The RE bit in the U2C1 register is set to "1" (reception enabled) – The TE bit in the U2C1 register is set to "1" (transmission enabled) – The TI bit in the U2C1 register is set to "0" (data present in the UiTB register)
Interrupt request generation timing	When start or stop condition is detected, acknowledge undetected, and acknowledge detected
Error detection	• Overrun error (2) This error occurs if the serial I/O started receiving the next data before reading the U2RB register and received the 8th bit of the next data
Select function	• Arbitration lost Timing at which the ABT bit in the U2RB register is updated can be selected • SDA2 digital delay No digital delay or a delay of 2 to 8 U2BRG count source clock cycles selectable • Clock phase setting With or without clock delay selectable

NOTES:
1. When an external clock is selected, the conditions must be met while the external clock is in the high state.
2. If an overrun error occurs, bits 8 to 0 in UiRB register are undefined. The IR bit in the SiRIC register remains unchanged.

Downloaded from Arrow.com.



This diagram applies to the case where the SMD2 to SMD0 bits in the the U2MR register is set to "010 2" and the IICM bit in the U2SMR register is set to "1".

IICM : Bit in the U2SMR
IICM2, SWC, ALS, SWC2, SDHI : Bits in the U2SMR2
STSPSEL, ACKD, ACKC : Bits in the U2SMR4

NOTE:
1. If the IICM bit is set to "1", the pin can be read even when the PD7_1 bit is set to "1" (output mode).

Figure 13.1.3.1. I²C bus Mode Block Diagram

Downloaded from Arrow.com.

Table 13.1.3.2. Registers to Be Used and Settings in I²C bus Mode (1) (Continued)

Register	Bit	Function	
		Master	Slave
U2TB (1)	0 to 7	Set transmission data	Set transmission data
U2RB (1)	0 to 7	Reception data can be read	Reception data can be read
	8	ACK or NACK is set in this bit	ACK or NACK is set in this bit
	ABT	Arbitration lost detection flag	Invalid
	OER	Overrun error flag	Overrun error flag
U2BRG	0 to 7	Set a transfer rate	Invalid
U2MR (1)	SMD2 to SMD0	Set to '010$_2$'	Set to '010$_2$'
	CKDIR	Set to "0"	Set to "1"
	IOPOL	Set to "0"	Set to "0"
U2C0	CLK1, CLK0	Select the count source for the U2BRG register	Invalid
	CRS	Invalid because CRD = 1	Invalid because CRD = 1
	TXEPT	Transmit buffer empty flag	Transmit buffer empty flag
	CRD	Set to "1"	Set to "1"
	NCH	Set to "1"	Set to "1"
	CKPOL	Set to "0"	Set to "0"
	UFORM	Set to "1"	Set to "1"
U2C1	TE	Set this bit to "1" to enable transmission	Set this bit to "1" to enable transmission
	TI	Transmit buffer empty flag	Transmit buffer empty flag
	RE	Set this bit to "1" to enable reception	Set this bit to "1" to enable reception
	RI	Reception complete flag	Reception complete flag
	U2IRS	Invalid	Invalid
	U2RRM, U2LCH, U2ERE	Set to "0"	Set to "0"
U2SMR	IICM	Set to "1"	Set to "1"
	ABC	Select the timing at which arbitration-lost is detected	Invalid
	BBS	Bus busy flag	Bus busy flag
	3 to 7	Set to "0"	Set to "0"
U2SMR2	IICM2	Refer to **Table 13.1.3.4 I²C bus Mode Functions**	Refer to **Table 13.1.3.4 I²C bus Mode Functions**
	CSC	Set this bit to "1" to enable clock synchronization	Set to "0"
	SWC	Set this bit to "1" to have SCL$_2$ output fixed to "L" at the falling edge of the 9th bit of clock	Set this bit to "1" to have SCL$_2$ output fixed to "L" at the falling edge of the 9th bit of clock
	ALS	Set this bit to "1" to have SDA$_2$ output stopped when arbitration-lost is detected	Set to "0"
	STAC	Set to "0"	Set this bit to "1" to initialize UART2 at start condition detection
	SWC2	Set this bit to "1" to have SCL$_2$ output forcibly pulled low	Set this bit to "1" to have SCL$_2$ output forcibly pulled low
	SDHI	Set this bit to "1" to disable SDA$_2$ output	Set this bit to "1" to disable SDA$_2$ output
	7	Set to "0"	Set to "0"
U2SMR3	0, 2, 4 and NODC	Set to "0"	Set to "0"
	CKPH	Refer to **Table 13.1.3.4 I²C bus Mode Functions**	Refer to **Table 13.1.3.4 I²C bus Mode Functions**
	DL2 to DL0	Set the amount of SDA$_2$ digital delay	Set the amount of SDA$_2$ digital delay

NOTE:

1. Not all register bits are described above. Set those bits to "0" when writing to the registers in I²C bus mode.

Downloaded from Arrow.com.

Table 13.1.3.3. Registers to Be Used and Settings in I²C bus Mode (2) (Continued)

Register	Bit	Function	
		Master	Slave
U2SMR4	STAREQ	Set this bit to “1” to generate start condition	Set to “0”
	RSTAREQ	Set this bit to “1” to generate restart condition	Set to “0”
	STPREQ	Set this bit to “1” to generate stop condition	Set to “0”
	STSPSEL	Set this bit to “1” to output each condition	Set to “0”
	ACKD	Select ACK or NACK	Select ACK or NACK
	ACKC	Set this bit to “1” to output ACK data	Set this bit to “1” to output ACK data
	SCLHI	Set this bit to “1” to have SCL2 output stopped when stop condition is detected	Set to “0”
	SWC9	Set to “0”	Set this bit to “1” to set the SCL2 to “L” hold at the falling edge of the 9th bit of clock

NOTE:

1. Not all bits in the register are described above. Set those bits to “0” when writing to the registers in I²C bus mode.

Downloaded from Arrow.com.

Table 13.1.3.4. I²C bus Mode Functions

Function	Clock synchronous serial I/O mode (SMD2 to SMD0 = 001₂, IICM = 0)	I²C bus mode (SMD2 to SMD0 = 010₂, IICM = 1)			
		IICM2 = 0 (NACK/ACK interrupt)		IICM2 = 1 (UART transmit/ receive interrupt)	
		CKPH = 0 (No clock delay)	CKPH = 1 (Clock delay)	CKPH = 0 (No clock delay)	CKPH = 1 (Clock delay)
Factor of interrupt number 10 (1) (Refer to **Fig.13.1.3.2**.)	———	Start condition detection or stop condition detection (Refer to **Figure 13.1.3.2.1. STSPSEL Bit Function**)			
Factor of interrupt number 15 (1) (Refer to **Fig.13.1.3.2**.)	UART2 transmission Transmission started or completed (selected by U2IRS)	No acknowledgment detection (NACK) Rising edge of SCL₂ 9th bit		UART2 transmission Rising edge of SCL₂ 9th bit	UART2 transmission Falling edge of SCL₂ next to the 9th bit
Factor of interrupt number 16 (1) 1(Refer to **Fig.13.1.3.2**.)	UART2 reception When 8th bit received CKPOL = 0 (rising edge) CKPOL = 1 (falling edge)	Acknowledgment detection (ACK) Rising edge of SCL₂ 9th bit		UART2 transmission Falling edge of SCL₂ 9th bit	
Timing for transferring data from the UART reception shift register to the U2RB register	CKPOL = 0 (rising edge) CKPOL = 1 (falling edge)	Rising edge of SCL₂ 9th bit		Falling edge of SCL₂ 9th bit	Falling and rising edges of SCL₂ 9th bit
UART2 transmission output delay	Not delayed	Delayed			
Functions of P7₀ pin	TxD2 output	SDA₂ input/output			
Functions of P7₁ pin	RxD2 input	SCL₂ input/output			
Functions of P7₂ pin	CLK2 input or output selected	——— (Cannot be used in I²C mode)			
Noise filter width	15ns	200ns			
Read RxD2 and SCL₂ pin levels	Possible when the corresponding port direction bit = 0	Always possible no matter how the corresponding port direction bit is set			
Initial value of TxD2 and SDA₂ outputs	CKPOL = 0 (H) CKPOL = 1 (L)	The value set in the port register before setting I²C bus mode (2)			
Initial and end values of SCL₂	———	H	L	H	L
DMA1 factor (Refer to **Fig. 14.1.3.2.**)	UART2 reception	Acknowledgment detection (ACK)		UART2 reception Falling edge of SCL₂ 9th bit	
Store received data	1st to 8th bits are stored in U2RB register bit 0 to bit 7	1st to 8th bits are stored in U2RB register bit 7 to bit 0		1st to 7th bits are stored in U2RB register bit 6 to bit 0, with 8th bit stored in U2RB register bit 8	1st to 8th bits are stored in U2RB register bit 7 to bit 0 (3)
Read received data	U2RB register status is read directly as is				Read U2RB register Bit 6 to bit 0 as bit 7 to bit 1, and bit 8 as bit 0 (4)

NOTES:
1. If the source or cause of any interrupt is changed, the IR bit in the interrupt control register for the changed interrupt may inadvertently be set to 1 (interrupt requested). (Refer to “Notes on interrupts” in Usage Notes) If one of the bits shown below is changed, the interrupt source, the interrupt timing, etc. change. Therefore, always be sure to clear the IR bit to 0 (interrupt not requested) after changing those bits. SMD2 to SMD0 bits in the U2MR register, IICM bit in the U2SMR register, IICM2 bit in the U2SMR2 register, CKPH bit in the U2SMR3 register
2. Set the initial value of SDA₂ output while the SMD2 to SMD0 bits in the U2MR register is set to ‘000₂’ (serial I/O disabled).
3. Second data transfer to U2RB register (Rising edge of SCL₂ 9th bit)
4. First data transfer to U2RB register (Falling edge of SCL₂ 9th bit)

Downloaded from Arrow.com.

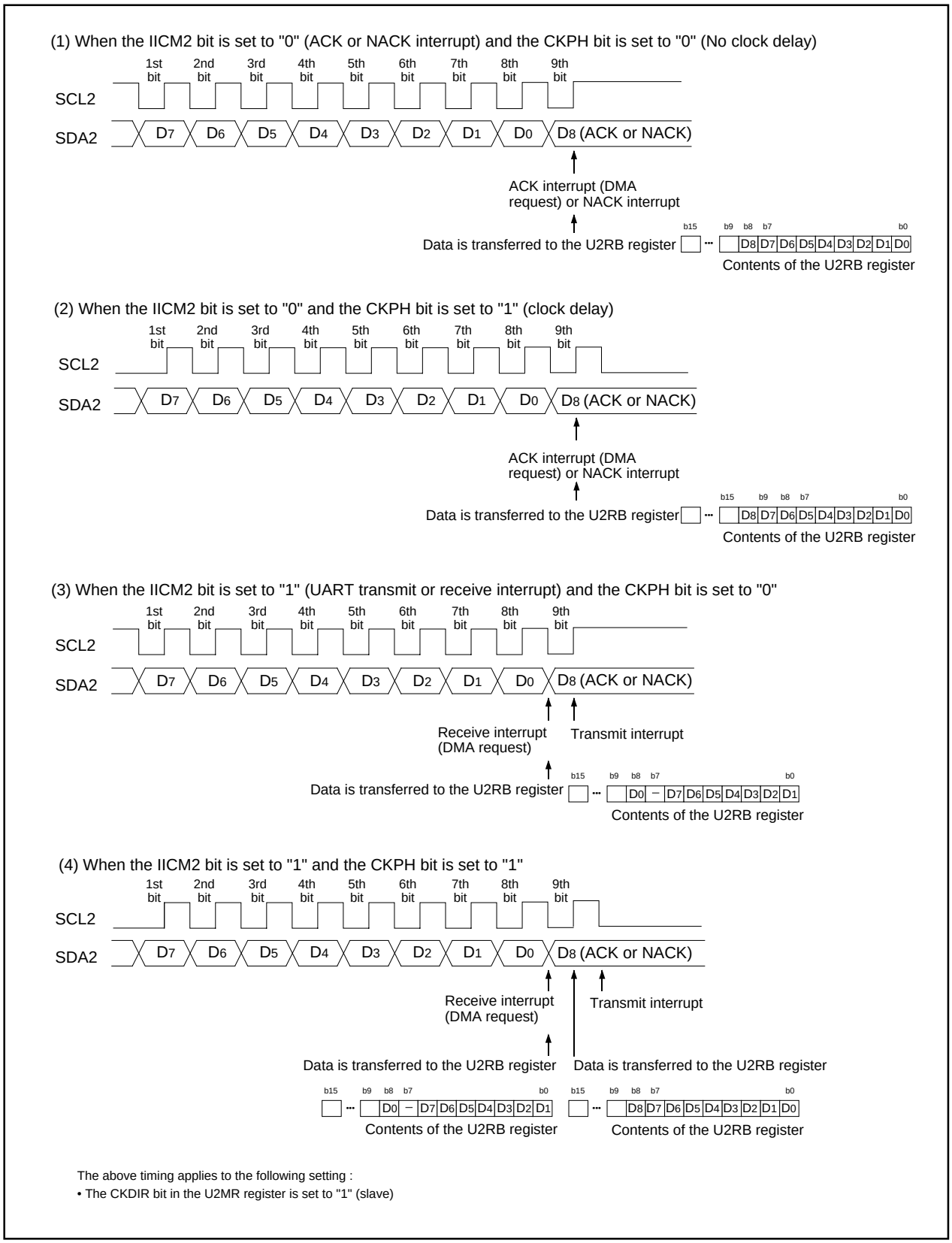


Figure 13.1.3.2. Transfer to U2RB Register and Interrupt Timing

Downloaded from Arrow.com.

13.1.3.1 Detection of Start and Stop Condition

Whether a start or a stop condition has been detected is determined.

A start condition-detected interrupt request is generated when the SDA2 pin changes state from high to low while the SCL2 pin is in the high state. A stop condition-detected interrupt request is generated when the SDA2 pin changes state from low to high while the SCL2 pin is in the high state.

Because the start and stop condition-detected interrupts share the interrupt control register and vector, check the BBS bit in the U2SMR register to determine which interrupt source is requesting the interrupt.

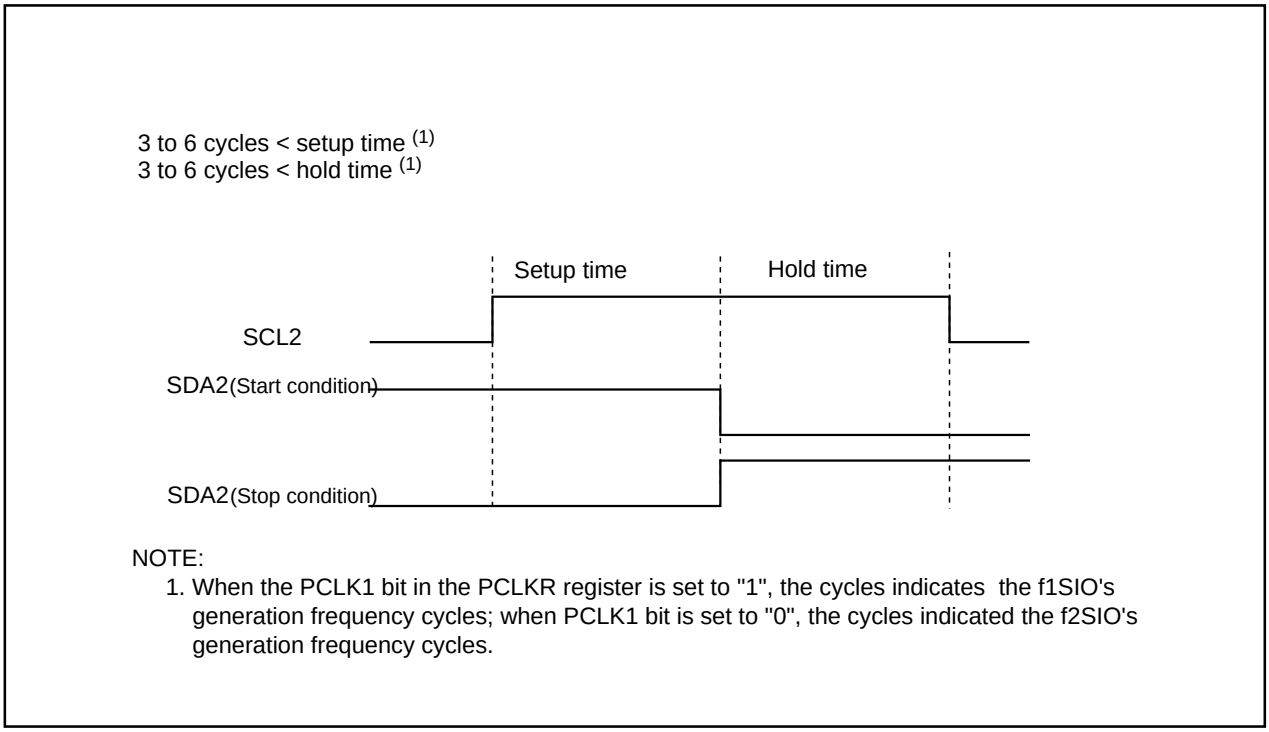


Figure 13.1.3.1.1. Detection of Start and Stop Condition

13.1.3.2 Output of Start and Stop Condition

A start condition is generated by setting the STAREQ bit in the U2SMR4 register to “1” (start).

A restart condition is generated by setting the RSTAREQ bit in the U2SMR4 register to “1” (start).

A stop condition is generated by setting the STPREQ bit in the U2SMR4 register to “1” (start).

The output procedure is described below.

(1) Set the STAREQ bit, RSTAREQ bit or STPREQ bit to “1” (start).

(2) Set the STSPSEL bit in the U2SMR4 register to “1” (output).

Make sure that no interrupts or DMA transfers will occur between (1) and (2).

The function of the STSPSEL bit is shown in Table 13.1.3.2.1 and Figure 13.1.3.2.1.

Downloaded from Arrow.com.

Table 13.1.3.2.1. STSPSEL Bit Functions

Function	STSPSEL = 0	STSPSEL = 1
Output of SCL2 and SDA2 pins	Output transfer clock and data/ Program with a port determines how the start condition or stop condition is output	The STAREQ, RSTAREQ and STPREQ bit determine how the start condition or stop condition is output
Start/stop condition interrupt request generation timing	Start/stop condition are detected	Start/stop condition generation are completed

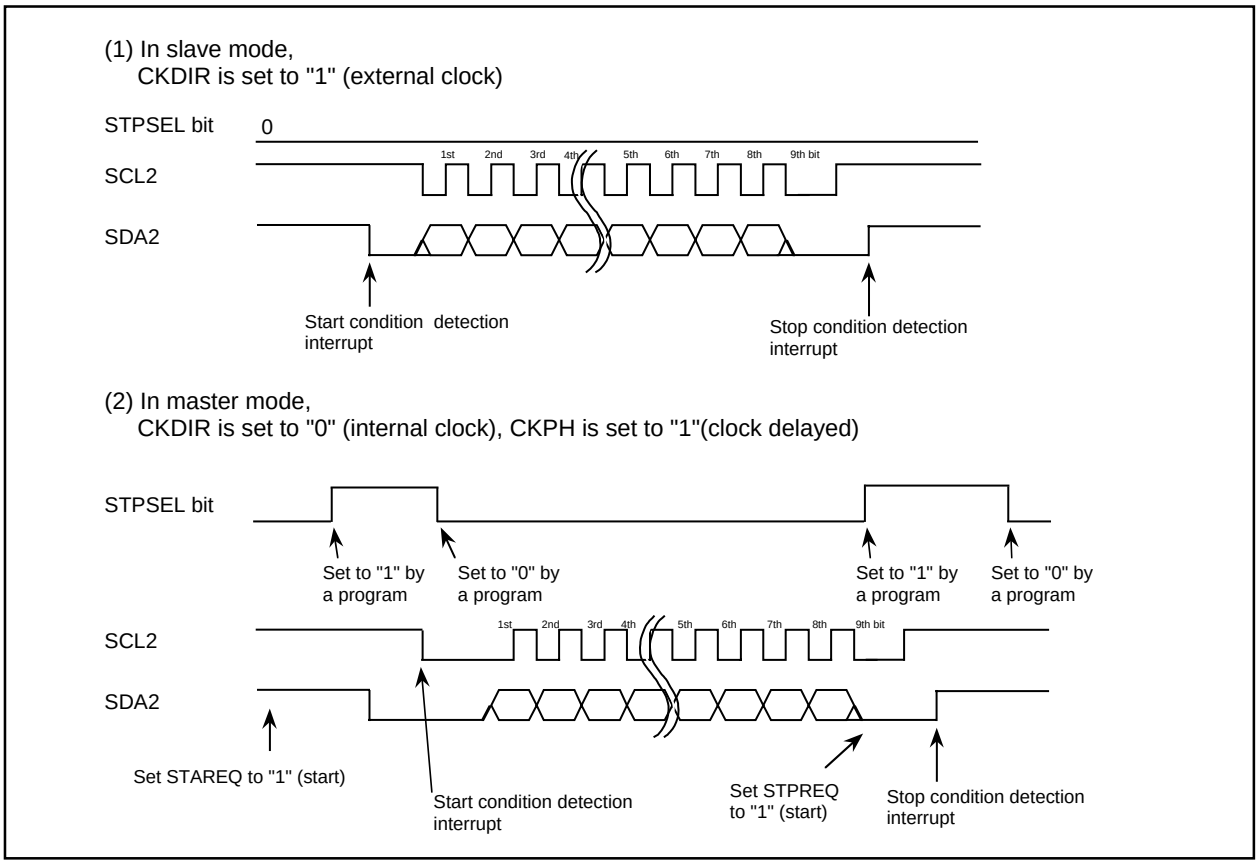


Figure 13.1.3.2.1. STSPSEL Bit Functions

13.1.3.3 Arbitration

Unmatching of the transmit data and SDA2 pin input data is checked synchronously with the rising edge of SCL2. Use the ABC bit in the U2SMR register to select the timing at which the ABT bit in the U2RB register is updated. If the ABC bit is set to "0" (updated bitwise), the ABT bit is set to “1” at the same time unmatching is detected during check, and is cleared to “0” when not detected. In cases when the ABC bit is set to “1”, if unmatching is detected even once during check, the ABT bit is set to “1” (unmatching detected) at the falling edge of the clock pulse of 9th bit. If the ABT bit needs to be updated bytewise, clear the ABT bit to “0” (undetected) after detecting acknowledge in the first byte, before transferring the next byte.

Setting the ALS bit in the U2SMR2 register to “1” (SDA output stop enabled) causes arbitration-lost to occur, in which case the SDA2 pin is placed in the high-impedance state at the same time the ABT bit is set to “1” (unmatching detected).

Downloaded from Arrow.com.

13.1.3.4 Transfer Clock

Data is transmitted/received using a transfer clock like the one shown in Figure 13.1.3.2.1.

The CSC bit in the U2SMR2 register is used to synchronize the internally generated clock (internal SCL2) and an external clock supplied to the SCL_2 pin. In cases when the CSC bit is set to “1” (clock synchronization enabled), if a falling edge on the SCL_2 pin is detected while the internal SCL_2 is high, the internal SCL_2 goes low, at which time the U2BRG register value is reloaded with and starts counting in the low-level interval. If the internal SCL_2 changes state from low to high while the SCL_2 pin is low, counting stops, and when the SCL_2 pin goes high, counting restarts.

In this way, the UART2 transfer clock is comprised of the logical product of the internal SCL_2 and SCL_2 pin signal. The transfer clock works from a half period before the falling edge of the internal SCL_2 1st bit to the rising edge of the 9^{th} bit. To use this function, select an internal clock for the transfer clock.

The SWC bit in the U2SMR2 register allows to select whether the SCL_2 pin should be fixed to or freed from low-level output at the falling edge of the 9th clock pulse.

If the SCLHI bit in the U2SMR4 register is set to “1” (enabled), SCL_2 output is turned off (placed in the high-impedance state) when a stop condition is detected.

Setting the SWC2 bit in the U2SMR2 register is set to "1" (0 output) makes it possible to forcibly output a low-level signal from the SCL_2 pin even while sending or receiving data. Clearing the SWC2 bit to “0” (transfer clock) allows the transfer clock to be output from or supplied to the SCL_2 pin, instead of outputting a low-level signal.

If the SWC9 bit in the U2SMR4 register is set to “1” (SCL hold low enabled) when the CKPH bit in the U2SMR3 register is set to "1", the SCL_2 pin is fixed to low-level output at the falling edge of the clock pulse next to the ninth. Setting the SWC9 bit is set to "0" (SCL hold low disabled) frees the SCL_2 pin from low-level output.

13.1.3.5 SDA Output

The data written to the bit 7 to bit 0 (D_7 to D_0) in the U2TB register is sequentially output beginning with D_7. The ninth bit (D_8) is ACK or NACK.

The initial value of SDA_2 transmit output can only be set when IICM is set to "1" (I^2C Bus mode) and the SMD2 to SMD0 bits in the the U2MR register are set to ‘000_2’ (serial I/O disabled).

The DL2 to DL0 bits in the U2SMR3 register allow to add no delays or a delay of 2 to 8 U2BRG count source clock cycles to SDA_2 output.

Setting the SDHI bit in the U2SMR2 register is set to "1" (SDA output disabled) forcibly places the SDA_2 pin in the high-impedance state. Do not write to the SDHI bit synchronously with the rising edge of the UART2 transfer clock. This is because the ABT bit may inadvertently be set to “1” (detected).

13.1.3.6 SDA Input

When the IICM2 bit is set to "0", the 1st to 8th bits (D_7 to D_0) of received data are stored in the bit 7 to bit 0 in the U2RB register. The 9th bit (D_8) is ACK or NACK.

When the IICM2 bit is set to "1", the 1st to 7th bits (D_7 to D_1) of received data are stored in the bit 6 to bit 0 in the U2RB register and the 8th bit (D_0) is stored in the bit 8 in the U2RB register. Even when the IICM2 bit is set to "1", providing the CKPH bit to "1", the same data as when the IICM2 bit is set to "0" can be read out by reading the U2RB register after the rising edge of the corresponding clock pulse of 9th bit.

Downloaded from Arrow.com.

13.1.3.7 ACK and NACK

If the STSPSEL bit in the U2SMR4 register is set to “0” (start and stop conditions not generated) and the ACKC bit in the U2SMR4 register is set to “1” (ACK data output), the value of the ACKD bit in the U2SMR4 register is output from the SDA2 pin.

If the IICM2 bit is set to "0", a NACK interrupt request is generated if the SDA2 pin remains high at the rising edge of the 9th bit of transmit clock pulse. An ACK interrupt request is generated if the SDA2 pin is low at the rising edge of the 9th bit of transmit clock pulse.

If ACK2 is selected for the cause of DMA1 request, a DMA transfer can be activated by detection of an acknowledge.

13.1.3.8 Initialization of Transmission/Reception

If a start condition is detected while the STAC bit is set to "1" (UART2 initialization enabled), the serial I/O operates as described below.

- The transmit shift register is initialized, and the content of the U2TB register is transferred to the transmit shift register. In this way, the serial I/O starts sending data synchronously with the next clock pulse applied. However, the UART2 output value does not change state and remains the same as when a start condition was detected until the first bit of data is output synchronously with the input clock.
- The receive shift register is initialized, and the serial I/O starts receiving data synchronously with the next clock pulse applied.
- The SWC bit is set to “1” (SCL wait output enabled). Consequently, the SCL2 pin is pulled low at the falling edge of the ninth clock pulse.

Note that when UART2 transmission/reception is started using this function, the TI does not change state. Note also that when using this function, the selected transfer clock should be an external clock.

Downloaded from Arrow.com.

13.1.4 Special Mode 2 (UART2)

Multiple slaves can be serially communicated from one master. Transfer clock polarity and phase are selectable. Table 13.1.4.1 lists the specifications of Special Mode 2. Table 13.1.4.2 lists the registers used in Special Mode 2 and the register values set. Figure 13.1.4.1 shows communication control example for Special Mode 2.

Table 13.1.4.1. Special Mode 2 Specifications

Item	Specification
Transfer data format	• Transfer data length: 8 bits
Transfer clock	• Master mode The CKDIR bit in the U2MR register is set to "0" (internal clock) : fj/ (2(n+1)) fj = f_{1SIO}, f_{2SIO}, f_{8SIO}, f_{32SIO}. n: Setting value of U2BRG register 00_{16} to FF_{16} • Slave mode The CKDIR bit is set to "1" (external clock selected) : Input from CLK2 pin
Transmit/receive control	Controlled by input/output ports
Transmission start condition	• Before transmission can start, the following requirements must be met (1) – The TE bit in the U2C1 register is set to "1" (transmission enabled) – The TI bit in the U2C1 register is set to "0" (data present in U2TB register)
Reception start condition	• Before reception can start, the following requirements must be met (1) – The RE bit in the U2C1 register is set to "1" (reception enabled) – The TE bit in the U2C1 register is set to "1" (transmission enabled) – The TI bit in the U2C1 register is set to "0" (data present in the U2TB register)
Interrupt request generation timing	• While transmitting, one of the following conditions can be selected – The U2IRS bit in the U2C1 register is set to "0" (transmit buffer empty): when transferring data from the U2TB register to the UART2 transmit register (at start of transmission) – The U2IRS bit is set to "1" (transfer completed): when the serial I/O finished sending data from the UART2 transmit register • While receiving When transferring data from the UART2 receive register to the U2RB register (at completion of reception)
Error detection	• Overrun error (2) This error occurs if the serial I/O started receiving the next data before reading the U2RB register and received the 7th bit of the next data
Select function	• Clock phase setting Selectable from four combinations of transfer clock polarities and phases

NOTES:

1. When an external clock is selected, the conditions must be met while if the CKPOL bit in the U2C0 register "0" (transmit data output at the falling edge and the receive data taken in at the rising edge of the transfer clock), the external clock is in the high state; if the CKPOL bit in the U2C0 register "1" (transmit data output at the rising edge and the receive data taken in at the falling edge of the transfer clock), the external clock is in the low state.
2. If an overrun error occurs, bits 8 to 0 in UiRB register are undefined. The IR bit in the SiRIC register remains unchanged.

Downloaded from Arrow.com.

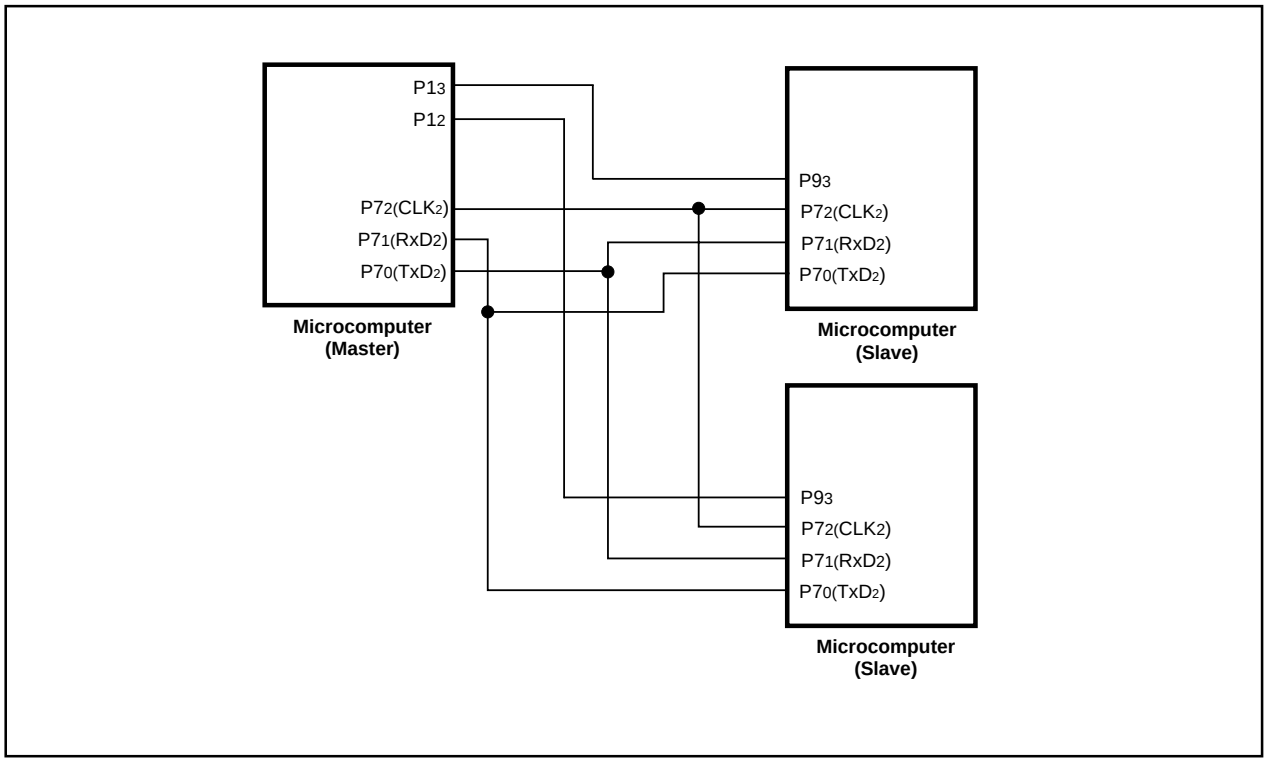


Figure 13.1.4.1. Serial Bus Communication Control Example (UART2)

Downloaded from Arrow.com.

Table 13.1.4.2. Registers to Be Used and Settings in Special Mode 2

Register	Bit	Function
U2TB(1)	0 to 7	Set transmission data
U2RB(1)	0 to 7	Reception data can be read
	OER	Overrun error flag
U2BRG	0 to 7	Set a transfer rate
U2MR(1)	SMD2 to SMD0	Set to '001z'
	CKDIR	Set this bit to "0" for master mode or "1" for slave mode
	IOPOL	Set to "0"
U2C0	CLK1, CLK0	Select the count source for the U2BRG register
	CRS	Invalid because CRD = 1
	TXEPT	Transmit register empty flag
	CRD	Set to "1"
	NCH	Select TxD2 pin output format
	CKPOL	Clock phases can be set in combination with the CKPH bit in the U2SMR3 register
	UFORM	Set to "0"
U2C1	TE	Set this bit to "1" to enable transmission
	TI	Transmit buffer empty flag
	RE	Set this bit to "1" to enable reception
	RI	Reception complete flag
	U2IRS	Select UART2 transmit interrupt cause
	U2RRM, U2LCH, U2ERE	Set to "0"
U2SMR	0 to 7	Set to "0"
U2SMR2	0 to 7	Set to "0"
U2SMR3	CKPH	Clock phases can be set in combination with the CKPOL bit in the U2C0 register
	NODC	Set to "0"
	0, 2, 4 to 7	Set to "0"
U2SMR4	0 to 7	Set to "0"

NOTE:

1. Not all bits in the register are described above. Set those bits to "0" when writing to the registers in Special Mode 2.

Downloaded from Arrow.com.

13.1.4.1 Clock Phase Setting Function

One of four combinations of transfer clock phases and polarities can be selected using the CKPH bit in the U2SMR3 register and the CKPOL bit in the U2C0 register.

Make sure the transfer clock polarity and phase are the same for the master and slave to communicate.

13.1.4.1.1 Master (Internal Clock)

Figure 13.1.4.1.1.1 shows the transmission and reception timing in master (internal clock).

13.1.4.1.2 Slave (External Clock)

Figure 13.1.4.1.2.1 shows the transmission and reception timing (CKPH=0) in slave (external clock) while Figure 13.1.4.1.2.2 shows the transmission and reception timing (CKPH=1) in slave (external clock).

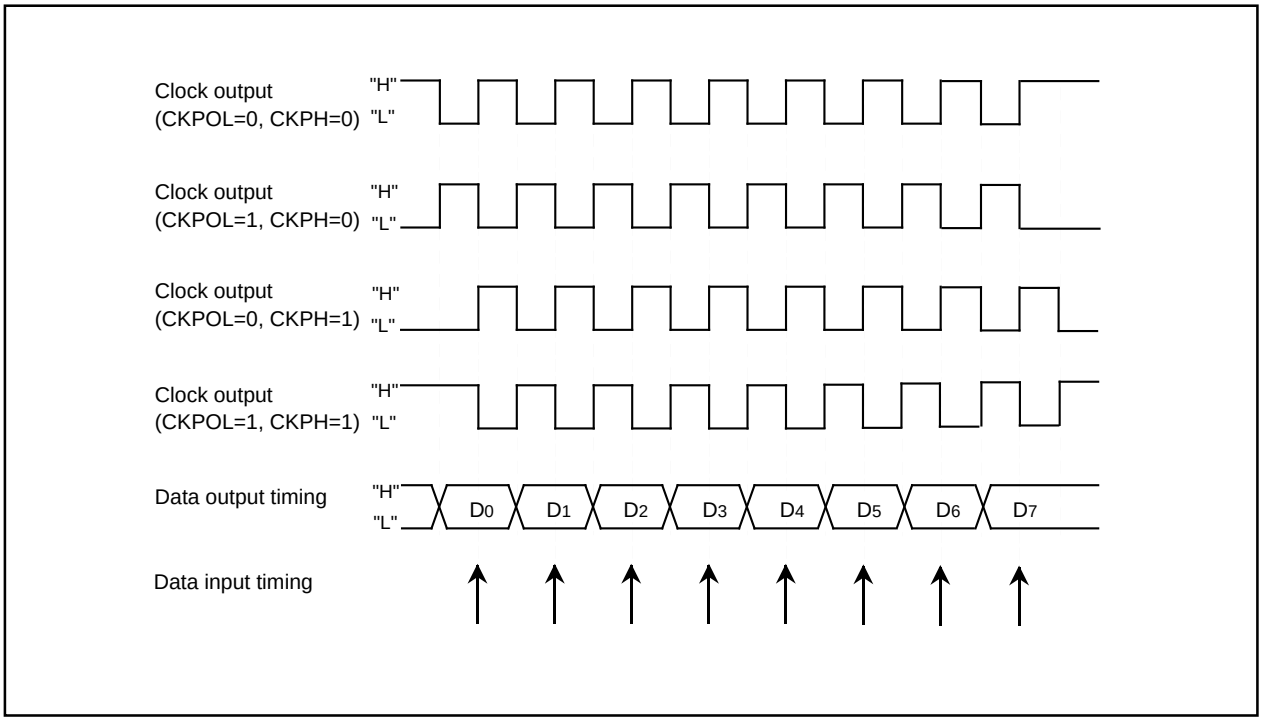


Figure 13.1.4.1.1.1. Transmission and Reception Timing in Master Mode (Internal Clock)

Downloaded from Arrow.com.

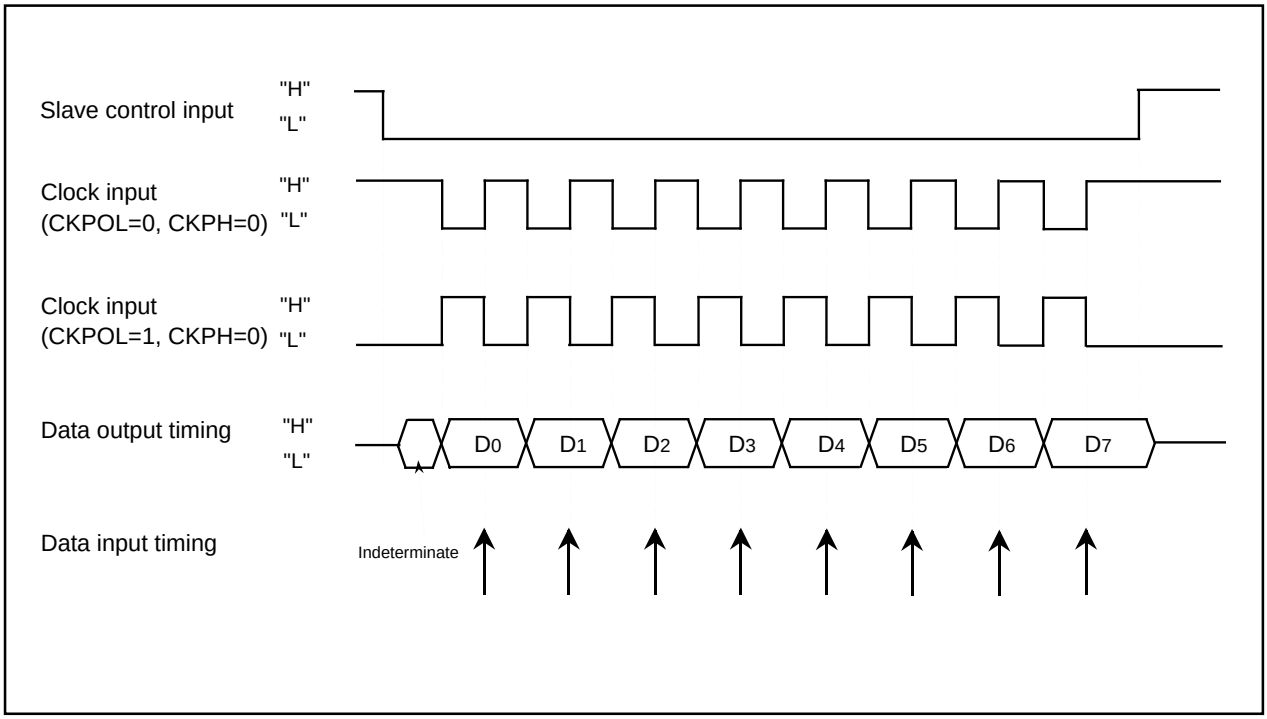


Figure 13.1.4.1.2.1. Transmission and Reception Timing (CKPH=0) in Slave Mode (External Clock)

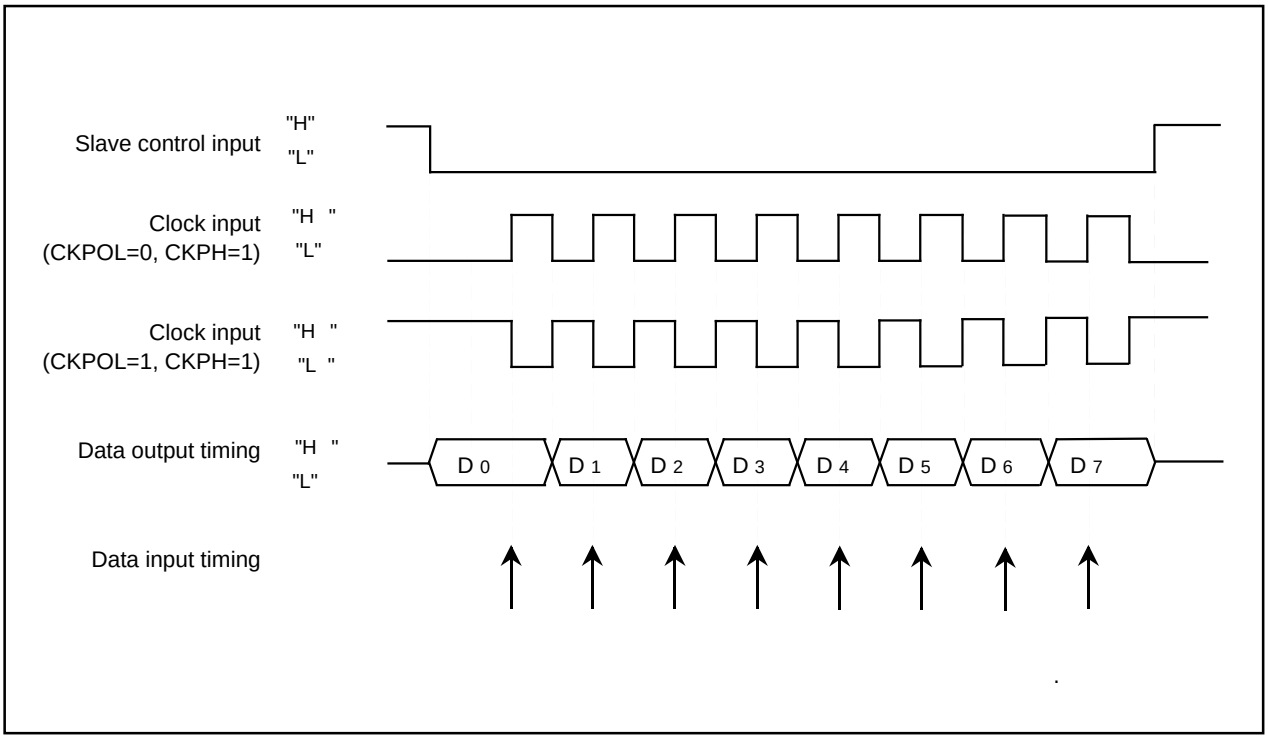


Figure 13.1.4.1.2.2. Transmission and Reception Timing (CKPH=1) in Slave Mode (External Clock)

Downloaded from Arrow.com.

13.1.5 Special Mode 3 (IE Bus mode)(UART2)

In this mode, one bit of IE Bus is approximated with one byte of UART mode waveform.

Table 13.1.5.1 lists the registers used in IE Bus mode and the register values set. Figure 13.1.5.1 shows the functions of bus collision detect function related bits.

If the TxD2 pin output level and RxD2 pin input level do not match, a UART2 bus collision detect interrupt request is generated.

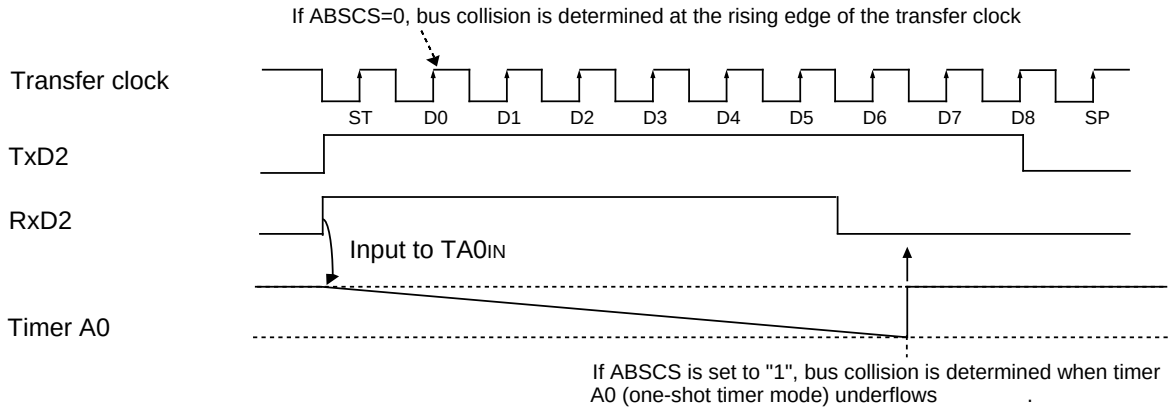
Table 13.1.5.1. Registers to Be Used and Settings in IE Bus Mode

Register	Bit	Function
U2TB	0 to 8	Set transmission data
U2RB(1)	0 to 8	Reception data can be read
	OER,FER,PER,SUM	Error flag
U2BRG	0 to 7	Set a transfer rate
U2MR	SMD2 to SMD0	Set to '110$_2$'
	CKDIR	Select the internal clock or external clock
	STPS	Set to "0"
	PRY	Invalid because PRYE=0
	PRYE	Set to "0"
	IOPOL	Select the TxD/RxD input/output polarity
U2C0	CLK1, CLK0	Select the count source for the U2BRG register
	CRS	Invalid because CRD=1
	TXEPT	Transmit register empty flag
	CRD	Set to "1"
	NCH	Select TxD2 pin output mode
	CKPOL	Set to "0"
	UFORM	Set to "0"
U2C1	TE	Set this bit to "1" to enable transmission
	TI	Transmit buffer empty flag
	RE	Set this bit to "1" to enable reception
	RI	Reception complete flag
	U2IRS	Select the source of UART2 transmit interrupt
	U2RRM, U2LCH, U2ERE	Set to "0"
U2SMR	0 to 3, 7	Set to "0"
	ABSCS	Select the sampling timing at which to detect a bus collision
	ACSE	Set this bit to "1" to use the auto clear function of transmit enable bit
	SSS	Select the transmit start condition
U2SMR2	0 to 7	Set to "0"
U2SMR3	0 to 7	Set to "0"
U2SMR4	0 to 7	Set to "0"

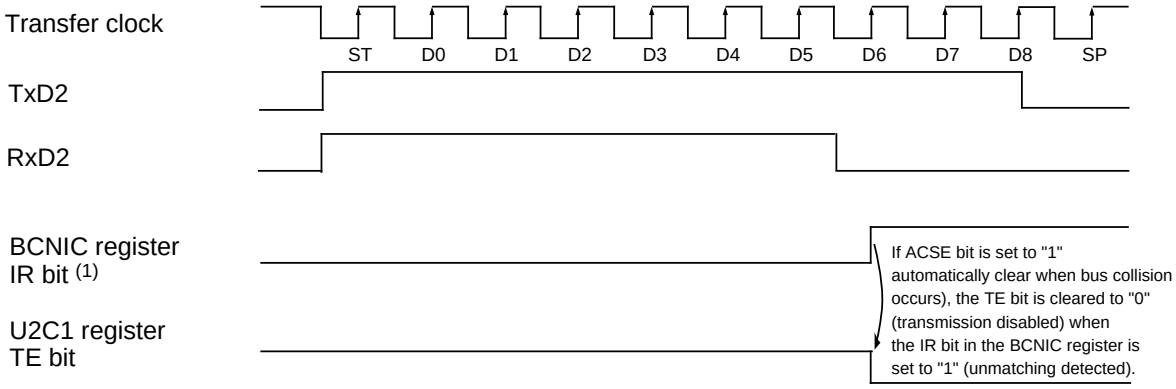
NOTE:
1. Not all bits in the registers are described above. Set those bits to "0" when writing to the registers in IEBus mode.

Downloaded from Arrow.com.

(1) The ABSCS bit in the U2SMR register (bus collision detect sampling clock select)

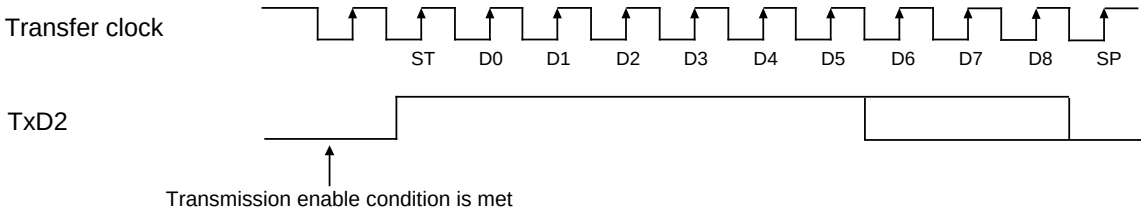


(2) The ACSE bit in the U2SMR register (auto clear of transmit enable bit)

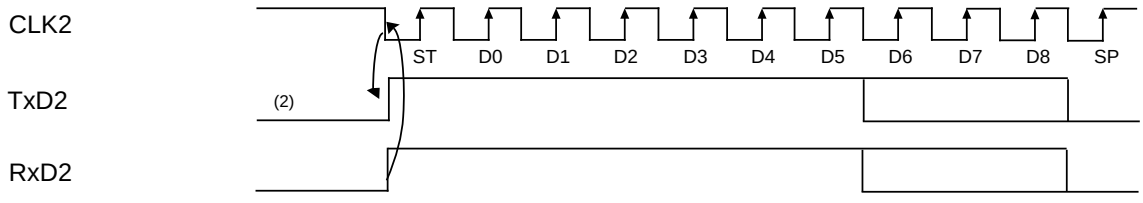


(3) The SSS bit in the U2SMR register (Transmit start condition select)

If SSS bit is set to "0", the serial I/O starts sending data one transfer clock cycle after the transmission enable condition is met.



If SSS bit = 1, the serial I/O starts sending data at the rising edge (1) of RxD2



NOTES:
1. The falling edge of RxD2 when the IOPOL is set to "0"; the rising edge of RxD2 when the IOPOL is set to "1".
2. The transmit condition must be met before the falling edge (Note 1) of RxD.

This diagram applies to the case where the IOPOL is set to "1" (reversed)

Figure 13.1.5.1. Bus Collision Detect Function-Related Bits

Downloaded from Arrow.com.

13.1.6 Special Mode 4 (SIM Mode) (UART2)

Based on UART mode, this is an SIM interface compatible mode. Direct and inverse formats can be implemented, and this mode allows output of a low from the TxD2 pin when a parity error is detected. Tables 13.1.6.1 lists the specifications of SIM mode. Table 13.1.6.2 lists the registers used in the SIM mode and the register values set.

Table 13.1.6.1. SIM Mode Specifications

Item	Specification
Transfer data format	• Direct format • Inverse format
Transfer clock	• The CKDIR bit in the U2MR register is set to "0" (internal clock) : fi/(16(n+1)) fi = f1SIO, f2SIO, f8SIO, f32SIO. n: Setting value in U2BRG register 00$_{16}$ to FF$_{16}$ • The CKDIR bit is set to "1" (external clock) : fEXT/(16(n+1)) fEXT: Input from CLK2 pin. n: Setting value in U2BRG register 00$_{16}$ to FF$_{16}$
Transmission start condition	• Before transmission can start, the following requirements must be met – The TE bit in the U2C1 register is set to "1" (transmission enabled) – The TI bit in the U2C1 register is set to "0" (data present in U2TB register)
Reception start condition	• Before reception can start, the following requirements must be met – The RE bit in the U2C1 register is set to "1" (reception enabled) – Start bit detection
Interrupt request generation timing (2)	• For transmission When the serial I/O finished sending data from the U2TB transfer register (the U2IRS bit is set to "1") • For reception When transferring data from the UART2 receive register to the U2RB register (at completion of reception)
Error detection	• Overrun error (1) This error occurs if the serial I/O started receiving the next data before reading the U2RB register and received the bit one before the last stop bit of the next data • Framing error This error occurs when the number of stop bits set is not detected • Parity error During reception, if a parity error is detected, parity error signal is output from the TxD2 pin. During transmission, a parity error is detected by the level of input to the RxD2 pin when a transmission interrupt occurs • Error sum flag This flag is set to "1" when any of the overrun, framing, and parity errors is encountered

NOTES:

1. If an overrun error occurs, bits 8 to 0 in UiRB register are undefined. The IR bit in the SiRIC register remains unchanged.
2. A transmit interrupt request is generated by setting the U2IRS bit in the U2C1 register to "1" (transmission complete) and the U2ERE bit to "1" (error signal output) after reset. Therefore, when using SIM mode, be sure to clear the IR bit to "0" (no interrupt request) after setting these bits.

Downloaded from Arrow.com.

Table 13.1.6.2. Registers to Be Used and Settings in SIM Mode

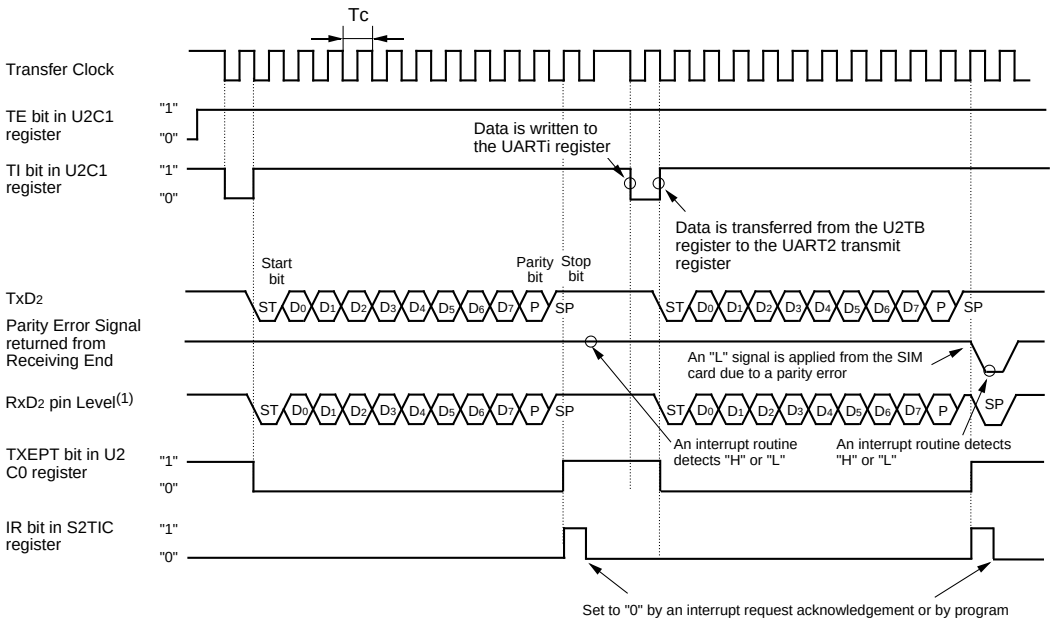
Register	Bit	Function
U2TB(1)	0 to 7	Set transmission data
U2RB(1)	0 to 7	Reception data can be read
	OER,FER,PER,SUM	Error flag
U2BRG	0 to 7	Set a transfer rate
U2MR	SMD2 to SMD0	Set to '101$_2$'
	CKDIR	Select the internal clock or external clock
	STPS	Set to "0"
	PRY	Set this bit to "1" for direct format or "0" for inverse format
	PRYE	Set to "1"
	IOPOL	Set to "0"
U2C0	CLK1, CLK0	Select the count source for the U2BRG register
	CRS	Invalid because CRD=1
	TXEPT	Transmit register empty flag
	CRD	Set to "1"
	NCH	Set to "0"
	CKPOL	Set to "0"
	UFORM	Set this bit to "0" for direct format or "1" for inverse format
U2C1	TE	Set this bit to "1" to enable transmission
	TI	Transmit buffer empty flag
	RE	Set this bit to "1" to enable reception
	RI	Reception complete flag
	U2IRS	Set to "1"
	U2RRM	Set to "0"
	U2LCH	Set this bit to "0" for direct format or "1" for inverse format
	U2ERE	Set to "1"
U2SMR(1)	0 to 3	Set to "0"
U2SMR2	0 to 7	Set to "0"
U2SMR3	0 to 7	Set to "0"
U2SMR4	0 to 7	Set to "0"

NOTE:

1. Not all bits in registers are described above. Set those bits to "0" when writing to the registers in SIM mode.

Downloaded from Arrow.com.

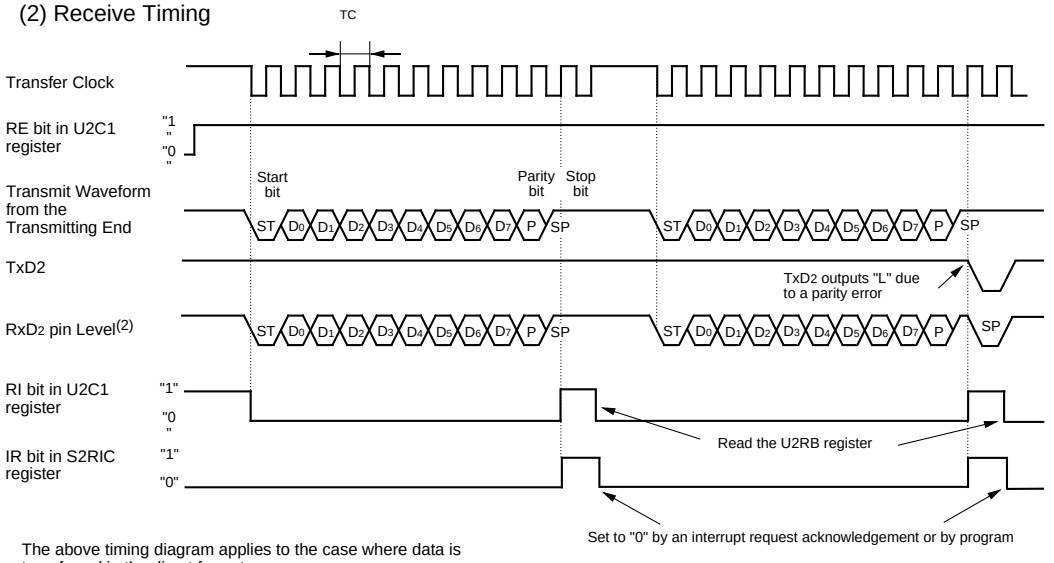
(1) Transmit Timing



The above timing diagram applies to the case where data is transferred in the direct format.
- U2MR register STPS bit = 0 (1 stop bit)
- U2MR register PRY bit = 1 (even)
- U2C0 register UFORM bit = 0 (LSB first)
- U2C1 register U2LCH bit = 0 (no reverse)
- U2C1 register U2IRSCH bit = 1 (transmit is completed)

$Tc = 16 (n + 1) / fi$ or $16 (n + 1) / f_{EXT}$
fi : frequency of U2BRG count source (f1SIO, f2SIO, f8SIO, f32SIO)
f_{EXT} : frequency of U2BRG count source (external clock)
n : value set to U2BRG

(2) Receive Timing



The above timing diagram applies to the case where data is transferred in the direct format.
- U2MR register STPS bit = 0 (1 stop bit)
- U2MR register PRY bit = 1 (even)
- U2C0 register UFORM bit = 0 (LSB first)
- U2C1 register U2LCH bit = 0 (no reverse)
- U2C1 register U2IRSCH bit = 1 (transmit is completed)

$Tc = 16 (n + 1) / fi$ or $16 (n + 1) / f_{EXT}$
fi : frequency of U2BRG count source (f1SIO, f2SIO, f8SIO, f32SIO)
f_{EXT} : frequency of U2BRG count source (external clock)
n : value set to U2BRG

NOTES:
1. Because TxD2 and RxD2 are connected, this is composite waveform consisting of the TxD2 output and the parity error signal sent back from receiver.
2. Because TxD2 and RxD2 are connected, this is composite waveform consisting of the transmitter's transmit waveform and the parity error signal received.

Figure 13.1.6.1. Transmit and Receive Timing in SIM Mode

Downloaded from Arrow.com.

Figure 13.1.6.2 shows the example of connecting the SIM interface. Connect TxD2 and RxD2 and apply pull-up.

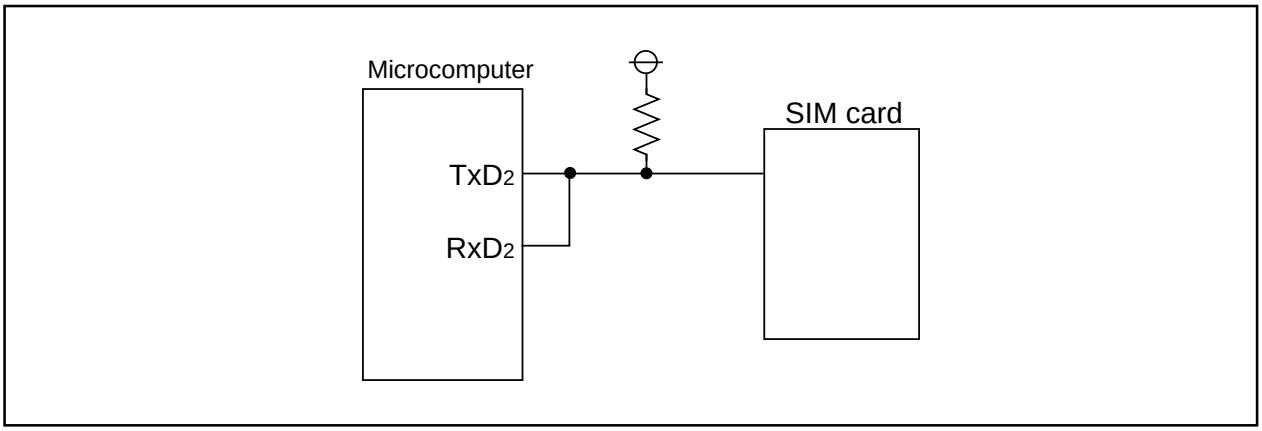


Figure 13.1.6.2. SIM Interface Connection

13.1.6.1 Parity Error Signal Output

The parity error signal is enabled by setting the U2ERE bit in the U2C1 register' to "1".

• When receiving

The parity error signal is output when a parity error is detected while receiving data. This is achieved by pulling the TxD2 output low with the timing shown in Figure 13.1.6.1.1. If the R2RB register is read while outputting a parity error signal, the PER bit is cleared to "0" and at the same time the TxD2 output is returned high.

• When transmitting

A transmission-finished interrupt request is generated at the falling edge of the transfer clock pulse that immediately follows the stop bit. Therefore, whether a parity signal has been returned can be determined by reading the port that shares the RxD2 pin in a transmission-finished interrupt service routine.

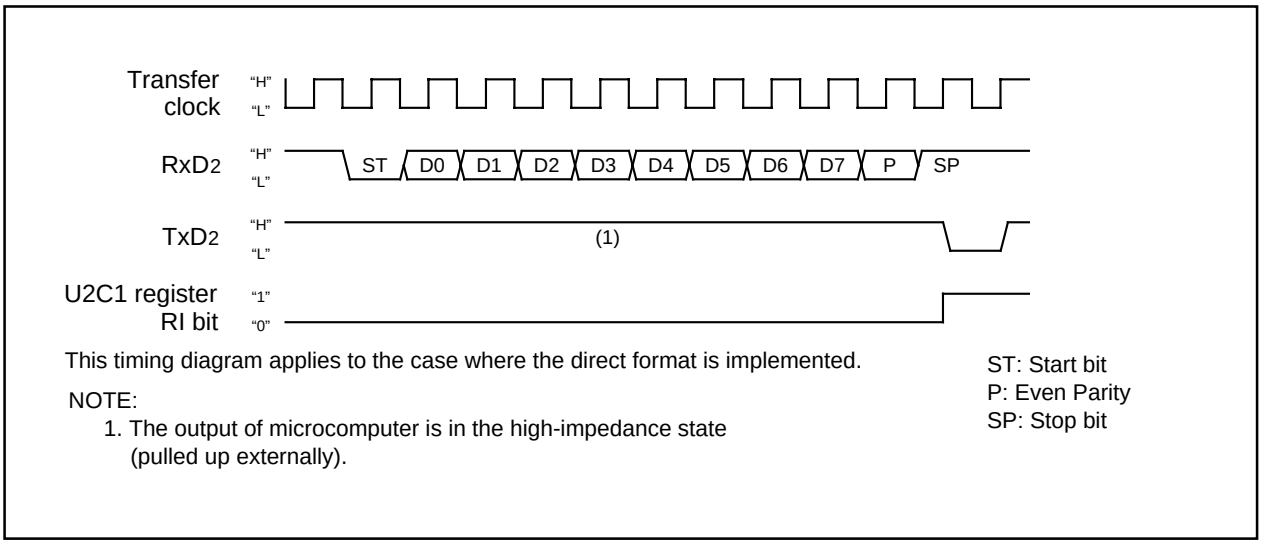


Figure 13.1.6.1.1. Parity Error Signal Output Timing

Downloaded from Arrow.com.

13.1.6.2 Format

• Direct Format

Set the PRY bit in the U2MR register to “1”, the UFORM bit in the U2C0 register to “0” and the U2LCH bit in the U2C1 register to “0”.

• Inverse Format

Set the PRY bit to “0”, UFORM bit to “1” and U2LCH bit to “1”.

Figure 13.1.6.2.1 shows the SIM interface format.

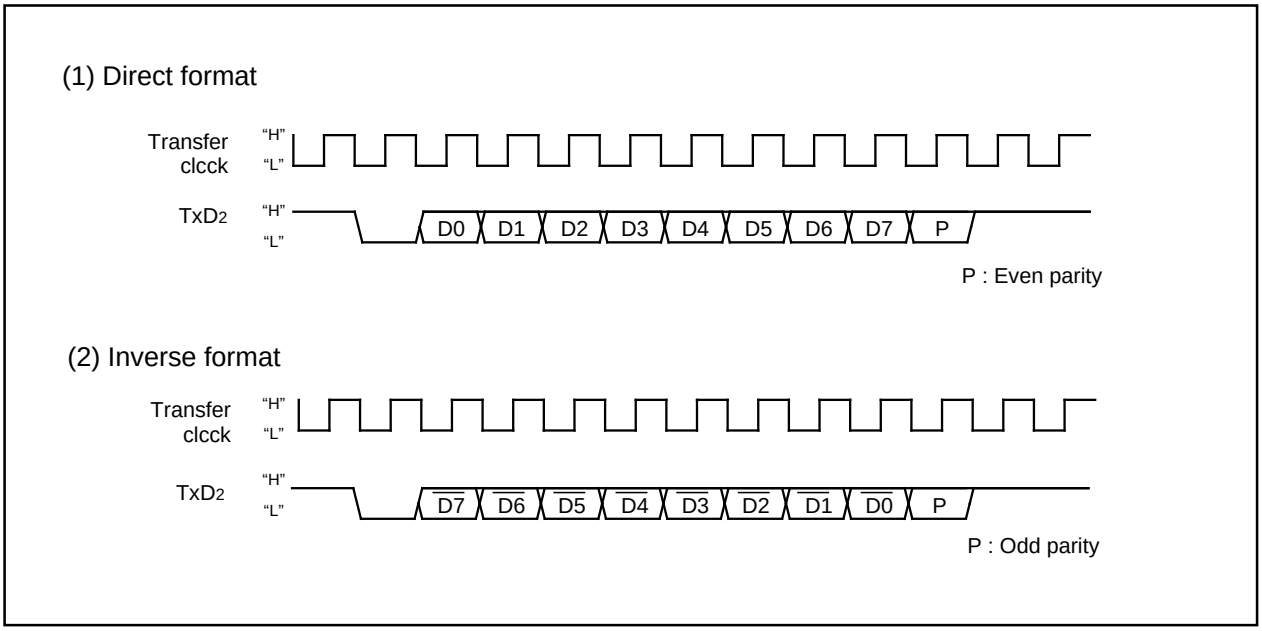


Figure 13.1.6.2.1. SIM Interface Format

Downloaded from Arrow.com.

14. A/D Converter

Note

P9$_2$ and P9$_3$ (AN$_{32}$, AN$_{24}$) are not available in the 42-pin package.
Do not use P9$_2$ and P9$_3$ (AN$_{32}$, AN$_{24}$) as analog input pins in the 42-pin package.

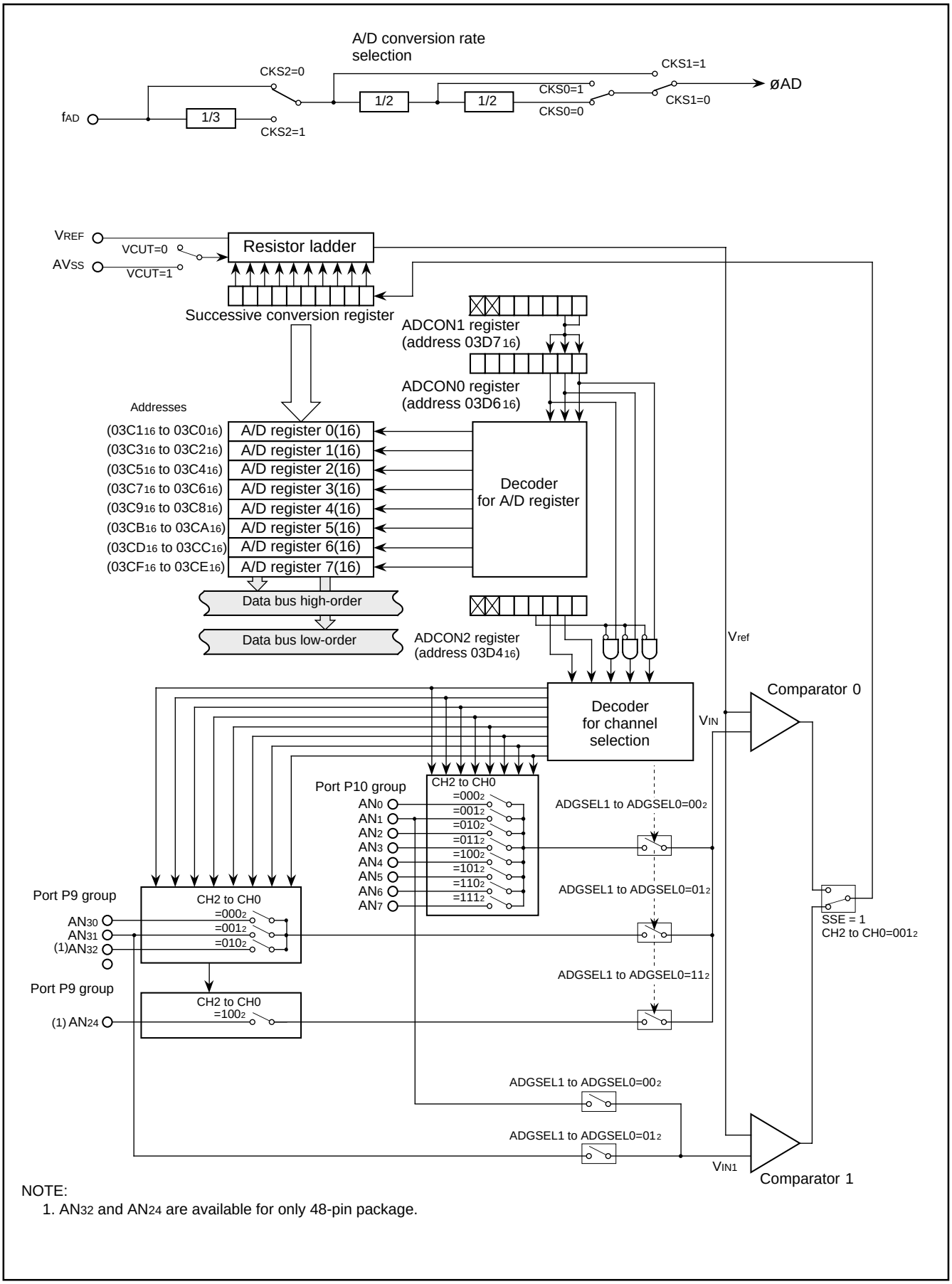
The microcomputer contains one A/D converter circuit based on 10-bit successive approximation method configured with a capacitive-coupling amplifier. The analog inputs share the pins with P10$_0$ to P10$_7$ (AN$_0$ to AN$_7$), P9$_0$ to P9$_3$ (AN3$_0$ to AN3$_2$, AN2$_4$). Similarly, $\overline{\text{ADTRG}}$ input shares the pin with P1$_5$. Therefore, when using these inputs, make sure the corresponding port direction bits are set to “0” (input mode).
When not using the A/D converter, set the VCUT bit to “0” (V_{REF} unconnected), so that no current will flow from the V_{REF} pin into the resistor ladder, helping to reduce the power consumption of the chip.
The A/D conversion result is stored in the i bits in the A/D register for AN$_i$, AN3$_i$, and AN2$_i$ pins (i = 0 to 7).
Table 14.1 shows the A/D converter performance. Figure 14.1 shows the A/D converter block diagram and Figures 14.2 to 14.4 show the A/D converter associated with registers.

Table 14.1 A/D Converter Performance

Item	Performance
A/D Conversion Method	Successive approximation (capacitive coupling amplifier)
Analog Input Voltage (1)	0V to AV_{CC} (V_{CC})
Operating Clock fAD (2)	f_{AD}/divided-by-2 or f_{AD}/divided-by-3 or f_{AD}/divided-by-4 or f_{AD}/divided-by-6 or f_{AD}/divided-by-12 or f_{AD}
Resolution	8-bit or 10-bit (selectable)
Integral Nonlinearity Error	When AV_{CC} = V_{REF} = 5V • With 8-bit resolution: ±2LSB • With 10-bit resolution: ±3LSB When AV_{CC} = V_{REF} = 3.3V • With 8-bit resolution: ±2LSB • With 10-bit resolution: ±5LSB
Operating Modes	One-shot mode, repeat mode, single sweep mode, repeat sweep mode 0, repeat sweep mode 1, simultaneous sample sweep mode and delayed trigger mode 0,1
Analog Input Pins (3)	8 pins (AN$_0$ to AN$_7$) + 3 pins (AN$_{30}$ to AN$_{32}$) + 1 pins (AN$_{24}$) (48-pin package) 8 pins (AN$_0$ to AN$_7$) + 2 pins (AN$_{30}$, AN$_{31}$) (42-pin package)
Conversion Speed Per Pin	• Without sample and hold function 8-bit resolution: 49 f_{AD} cycles, 10-bit resolution: 59 f_{AD} cycles • With sample and hold function 8-bit resolution: 28 f_{AD} cycles, 10-bit resolution: 33 f_{AD} cycles

NOTES:
1. Not dependent on use of sample and hold function.
2. Set the ϕAD frequency to 10 MHz or less. For M16C/26B, set it to 12 MHz or less.
   Without sample-and-hold function, set the fAD frequency to 250kHz or more.
   With the sample and hold function, set the fAD frequency to 1MHz or more.

Downloaded from Arrow.com.



NOTE:
1. AN32 and AN24 are available for only 48-pin package.

Figure 14.1 A/D Converter Block Diagram

Downloaded from Arrow.com.

A/D control register 0 (1)

b7	b6	b5	b4	b3	b2	b1	b0	Symbol	Address	After reset
								ADCON0	03D6$_{16}$	00000XXX$_2$

Bit symbol	Bit name	Function	RW
CH0	Analog Input Pin Select Bit	Function varies with each operation mode	RW
CH1			RW
CH2			RW
MD0	A/D Operation Mode Select Bit 0	b4 b3 0 0 : One-shot mode or Delayed trigger mode 0,1 0 1 : Repeat mode 1 0 : Single sweep mode or Simultaneous sample sweep mode 1 1 : Repeat sweep mode 0 or Repeat sweep mode 1	RW
MD1			RW
TRG	Trigger Select Bit	0 : Software trigger 1 : Hardware trigger	RW
ADST	A/D Conversion Start Flag	0 : A/D conversion disabled 1 : A/D conversion started	RW
CKS0	Frequency Select Bit 0	See **Table 14.2 A/D Conversion Frequency Select**	RW

NOTE:
1. If the ADCON0 register is rewritten during A/D conversion, the conversion result will be indeterminate.

A/D control register 1 (1)

b7	b6	b5	b4	b3	b2	b1	b0	Symbol	Address	After reset
X	X							ADCON1	03D7$_{16}$	00$_{16}$

Bit symbol	Bit name	Function	RW
SCAN0	A/D Sweep Pin Select Bit	Function varies with each operation mode	RW
SCAN1			RW
MD2	A/D Operation Mode Select Bit 1	0 : Other than repeat sweep mode 1 1 : Repeat sweep mode 1	RW
BITS	8/10-Bit Mode Select Bit	0 : 8-bit mode 1 : 10-bit mode	RW
CKS1	Frequency Select Bit 1	See **Table 14.2 A/D Conversion Frequency Select**	RW
VCUT	VREF Connect Bit (2)	0 : VREF not connected 1 : VREF connected	RW
— (b7-b6)	Nothing is assigned. When write, set to “0”. When read, its content is “0”.		—

NOTES:
1. If the ADCON1 register is rewritten during A/D conversion, the conversion result will be indeterminate.
2. If the VCUT bit is reset from “0” (VREF unconnected) to “1” (VREF connected), wait for 1 µs or more before starting A/D conversion.

A/D control register 2 (1)

b7	b6	b5	b4	b3	b2	b1	b0	Symbol	Address	After reset
X	X			0				ADCON2	03D4$_{16}$	00$_{16}$

Bit symbol	Bit name	Function	RW
SMP	A/D Conversion Method Select Bit	0 : Without sample and hold 1 : With sample and hold	RW
ADGSEL0	A/D Input Group Select Bit	b2 b1 0 0 : Select port P10 group 0 1 : Select port P9 group (AN3i) 1 0 : Do not set 1 1 : Select port P9 group (AN24)	RW
ADGSEL1			RW
— (b3)	Reserved Bit	Set to “0”	RW
CKS2	Frequency Select Bit 2	See **Table 14.2 A/D Conversion Frequency Select**	RW
TRG1	Trigger Select Bit	Function varies with each operation mode	RW
— (b7-b6)	Nothing is assigned. When write, set to “0”. When read, its content is “0”.		—

NOTE:
1. If the ADCON2 register is rewritten during A/D conversion, the conversion result will be indeterminate.

Figure 14.2 ADCON0 to ADCON2 Registers

Downloaded from Arrow.com.

A/D trigger control register (1,2)

b7	b6	b5	b4	b3	b2	b1	b0
X	X	X	X				

Symbol	Address	After reset
ADTRGCON	03D2$_{16}$	00$_{16}$

Bit symbol	Bit name	Function	RW
SSE	A/D Operation Mode Select Bit 2	0 : Other than simultaneous sample sweep mode or delayed trigger mode 0,1 1 : Simultaneous sample sweep mode or delayed trigger mode 0,1	RW
DTE	A/D Operation Mode Select Bit 3	0 : Other than delayed trigger mode 0,1 1 : Delayed trigger mode 0,1	RW
HPTRG0	AN0 Trigger Select Bit	Function varies with each operation mode	RW
HPTRG1	AN1 Trigger Select Bit	Function varies with each operation mode	RW
— (b7-b4)	Nothing is assigned. When write, set to “0”. When read, its content is “0”.		—

NOTES:
1. If the ADTRGCON register is rewritten during A/D conversion, the conversion result will be indeterminate.
2. Set “00$_{16}$” in this register in one-shot mode, repeat mode, single sweep mode, repeat sweep mode 0 and repeat sweep mode 1.

Figure 14.3 ADTRGCON Register

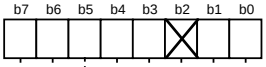
Table 14.2 A/D Conversion Frequency Select

CKS2	CKS1	CKS0	ØAD
0	0	0	Divided-by-4 of f_{AD}
0	0	1	Divided-by-2 of f_{AD}
0	1	0	f_{AD}
0	1	1	
1	0	0	Divided-by-12 of f_{AD}
1	0	1	Divided-by-6 of f_{AD}
1	1	0	Divided-by-3 of f_{AD}
1	1	1	

NOTE:
1. Set the ϕAD frequency to 10 MHz or less (12 MHz or less in M16C/26B). The ϕAD is selected with combinations of the CKS0 bit in the ADCON0 register, CKS1 bit in the ADCON1 register, and the CKS2 bit in the ADCON2 register.

Downloaded from Arrow.com.

A/D conversion status register 0 (1)



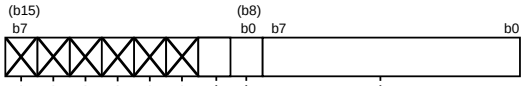
Symbol	Address	After reset
ADSTAT0	$03D3_{16}$	00_{16}

Bit symbol	Bit name	Function	RW
ADERR0	AN1 Trigger Status Flag	0 : AN1 trigger did not occur during AN0 conversion 1 : AN1 trigger occured during AN0 conversion	RW
ADERR1	Conversion Termination Flag	0 : Conversion not terminated 1 : Conversion terminated by Timer B0 underflow	RW
(b2)	Nothing is assigned. When write, set to “0”. When read, its content is “0”.		—
ADTCSF	Delayed Trigger Sweep Status Flag	0 : Sweep not in progress 1 : Sweep in progress	RO
ADSTT0	AN0 Conversion Status Flag	0 : AN0 conversion not in progress 1 : AN0 conversion in progress	RO
ADSTT1	AN1 Conversion Status Flag	0 : AN1 conversion not in progress 1 : AN1 conversion in progress	RO
ADSTRT0	AN0 Conversion Completion Status Flag	0 : AN0 conversion not completed 1 : AN0 conversion completed	RW
ADSTRT1	AN1 Conversion Completion Status Flag	0 : AN1 conversion not completed 1 : AN1 conversion completed	RW

NOTE:
1. ADSTAT0 register is valid only when the DTE bit in the ADTRGCON register is set to “1”.

A/D Register i (i=0 to 7)

Symbol	Address	After reset
AD0	$03C1_{16}$ to $03C0_{16}$	Indeterminate
AD1	$03C3_{16}$ to $03C2_{16}$	Indeterminate
AD2	$03C5_{16}$ to $03C4_{16}$	Indeterminate
AD3	$03C7_{16}$ to $03C6_{16}$	Indeterminate
AD4	$03C9_{16}$ to $03C8_{16}$	Indeterminate
AD5	$03CB_{16}$ to $03CA_{16}$	Indeterminate
AD6	$03CD_{16}$ to $03CC_{16}$	Indeterminate
AD7	$03CF_{16}$ to $03CE_{16}$	Indeterminate



Function		RW
When the BITS bit in the ADCON1 register is “1” (10-bit mode)	When the BITS bit in the ADCON1 register is “0” (8-bit mode)	RW
Eight low-order bits of A/D conversion result	A/D conversion result	RO
Two high-order bits of A/D conversion result	When read, its content is indeterminate	RO
Nothing is assigned. When write, set to “0”. When read, its content is “0”.		—

Figure 14.4 ADSTAT0 Register and AD0 to AD7 Registers

Downloaded from Arrow.com.

Timer B2 special mode register (1)

b7	b6	b5	b4	b3	b2	b1	b0
X	0	0					

Symbol	Address	After reset
TB2SC	039E16	X0000000₂

Bit symbol	Bit name	Function	RW
PWCOM	Timer B2 Reload Timing Switch Bit (2)	0 : Timer B2 underflow 1 : Timer A output at odd-numbered	RW
IVPCR1	Three-Phase Output Port $\overline{SD}$ Control Bit 1 (3, 4, 7)	0 : Three-phase output forcible cutoff by $\overline{SD}$ pin input (high impedance) disabled 1 : Three-phase output forcible cutoff by $\overline{SD}$ pin input (high impedance) enabled	RW
TB0EN	Timer B0 Operation Mode Select Bit	0 : Other than A/D trigger mode 1 : A/D trigger mode (5)	RW
TB1EN	Timer B1 Operation Mode Select Bit	0 : Other than A/D trigger mode 1 : A/D trigger mode (5)	RW
TB2SEL	Trigger Select Bit (6)	0 : TB2 interrupt 1 : Underflow of TB2 interrupt generation frequency setting counter [ICTB2]	RW
$\overline{\text{(b6-b5)}}$	Reserved bits	Must set to "0"	RW
— (b7)	Nothing is assigned. When write, set to “0”. When read, its content is “0”.		—

NOTES:
1. Write to this register after setting the PRC1 bit in the PRCR register to "1" (write enabled).
2. If the INV11 bit is "0" (three-phase mode 0) or the INV06 bit is "1" (triangular wave modulation mode), set this bit to "0" (timer B2 underflow).
3. When setting the IVPCR1 bit to "1" (three-phase output forcible cutoff by $\overline{SD}$ pin input enabled), Set the PD8_5 bit to "0" (= input mode).
4. Related pins are U(P8$_0$), $\overline{U}$(P8$_1$), V(P7$_2$), $\overline{V}$(P7$_3$), W(P7$_4$), $\overline{W}$(P7$_5$). When a high-level ("H") signal is applied to the $\overline{SD}$ pin and set the IVPCR1 bit to 0 after forcible cutoff, pins U, $\overline{U}$, V, $\overline{V}$, W, and $\overline{W}$ are exit from the high-impedance state. If a low-level (“L”) signal is applied to the $\overline{SD}$ pin, three-phase motor control timer output will be disabled (INV03=0). At this time, when the IVPCR1 bit is 0, pins U, $\overline{U}$, V, $\overline{V}$, W, and $\overline{W}$ become programmable I/O ports. When the IVPCR1 bit is set to 1, pins U, $\overline{U}$, V, $\overline{V}$, W, and $\overline{W}$ are placed in a high-impedance state regardless of which function of those pins is used.
5. When this bit is used in delayed trigger mode 0, set the TB0EN and TB1EN bits to "1"(A/D trigger mode).
6. When setting the TB2SEL bit to "1" (underflow of TB2 interrupt generation frequency setting counter[ICTB2]), Set the INV02 bit to "1" (three-phase motor control timer function).
7. Refer to **16.6 Digital Debounce function** for $\overline{SD}$ input.

Figure 14.5 TB2SC Register

Downloaded from Arrow.com.

14.1 Operation Modes

14.1.1 One-Shot Mode

In one-shot mode, analog voltage applied to a selected pin is once converted to a digital code. Table 14.1.1.1 shows the one-shot mode specifications. Figure 14.1.1.1 shows the operation example in one-shot mode. Figure 14.1.1.2 shows the ADCON0 to ADCON2 registers in one-shot mode.

Table 14.1.1.1 One-shot Mode Specifications

Item	Specification
Function	The CH2 to CH0 bits in the ADCON0 register and the ADGSEL1 to ADGSEL0 bits in the ADCON2 register select pins. Analog voltage applied to a selected pin is once converted to a digital code
A/D Conversion Start Condition	• When the TRG bit in the ADCON0 register is “0” (software trigger) Set the ADST bit in the ADCON0 register to “1” (A/D conversion started) • When the TRG bit in the ADCON0 register is “1” (hardware trigger) The $\overline{\text{ADTRG}}$ pin input changes state from “H” to “L” after setting the ADST bit to “1” (A/D conversion started)
A/D Conversion Stop Condition	• A/D conversion completed (If a software trigger is selected, the ADST bit is set to “0” (A/D conversion halted)). • Set the ADST bit to “0”
Interrupt Request Generation Timing	A/D conversion completed
Analog Input Pin	Select one pin from AN_0 to AN_7, AN_{30} to AN_{32}, AN_{24}
Readout of A/D Conversion Result	Readout one of the AD0 to AD7 registers that corresponds to the selected pin

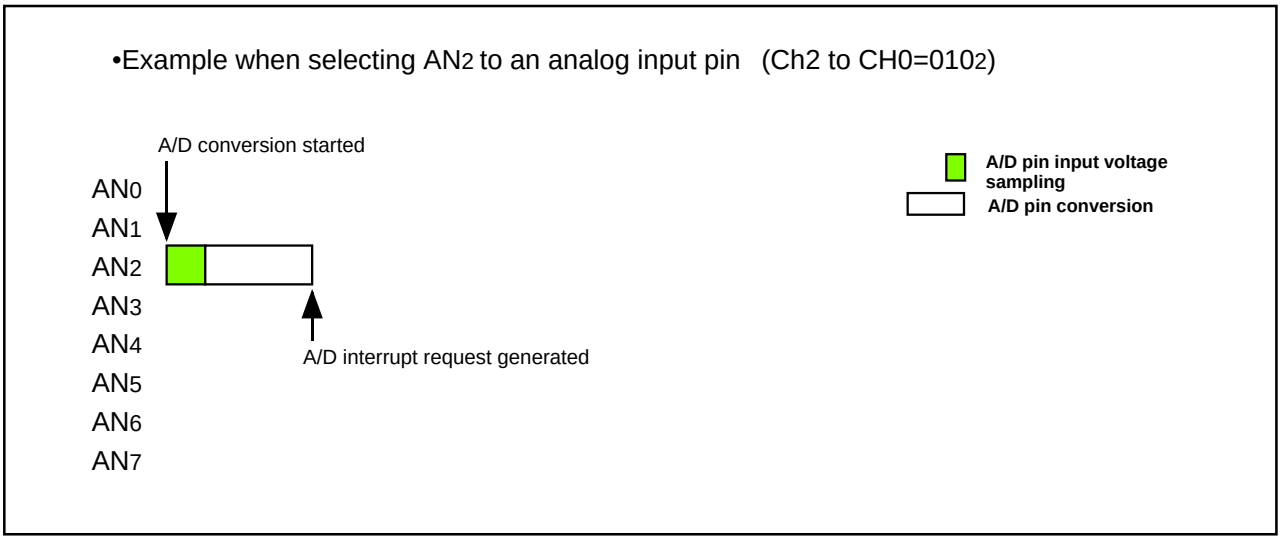
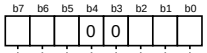


Figure 14.1.1.1 Operation Example in One-Shot Mode

Downloaded from Arrow.com.

A/D control register 0 (1)



Symbol ADCON0 — Address 03D6₁₆ — After reset 00000XXX₂

Bit symbol	Bit name	Function	RW
CH0	Analog Input Pin Select Bit (2, 3)	b2 b1 b0 0 0 0 : Select AN0 0 0 1 : Select AN1 0 1 0 : Select AN2 0 1 1 : Select AN3 1 0 0 : Select AN4 1 0 1 : Select AN5 1 1 0 : Select AN6 1 1 1 : Select AN7	RW
CH1			RW
CH2			RW
MD0	A/D Operation Mode Select Bit 0 (3)	b4 b3 0 0 : One-shot mode or delayed trigger mode 0,1	RW
MD1			RW
TRG	Trigger Select Bit	0 : Software trigger 1 : Hardware trigger ($\overline{\text{ADTRG}}$ trigger)	RW
ADST	A/D Conversion Start Flag	0 : A/D conversion disabled 1 : A/D conversion started	RW
CKS0	Frequency Select Bit 0	See **Table 14.2 A/D Conversion Frequency Select**	RW

NOTES:
1. If the ADCON0 register is rewritten during A/D conversion, the conversion result will be indeterminate.
2. AN30 to AN32 and AN24 can be used in the same way as AN0 to AN7 . Use the ADGSEL1 to ADGSEL0 bits in the ADCON2 register to select the desired pin.
3. After rewriting the MD1 to MD0 bits, set the CH2 to CH0 bits over again using an another instruction.

A/D control register 1 (1)

Symbol ADCON1 — Address 03D7₁₆ — After reset 00₁₆

Bit symbol	Bit name	Function	RW
SCAN0	A/D Sweep Pin Select Bit	Invalid in one-shot mode	RW
SCAN1			RW
MD2	A/D Operation Mode Select Bit 1	0 : Any mode other than repeat sweep mode 1	RW
BITS	8/10-Bit Mode Select Bit	0 : 8-bit mode 1 : 10-bit mode	RW
CKS1	Frequency Select Bit 1	Refer to **Table 14.2 A/D Conversion Frequency Select**	RW
VCUT	VREF Connect Bit (2)	1 : VREF connected	RW
— (b7-b6)	Nothing is assigned. When write, set to "0". When read, its content is "0".		—

NOTES:
1. If the ADCON1 register is rewritten during A/D conversion, the conversion result will be indeterminate.
2. If the VCUT bit is reset from "0" (VREF unconnected) to "1" (VREF connected), wait for 1 µs or more before starting A/D conversion.

A/D control register 2 (1)

Symbol ADCON2 — Address 03D4₁₆ — After reset 00₁₆

Bit symbol	Bit name	Function	RW
SMP	A/D Conversion Method Select Bit	0 : Without sample and hold 1 : With sample and hold	RW
ADGSEL0	A/D Input Group Select Bit	b2 b1 0 0 : Select port P10 group (ANi) 0 1 : Select port P9 group (AN3i) 1 0 : Do not set 1 1 : Select port P9 group (AN24)	RW
ADGSEL1			RW
— (b3)	Reserved Bit	Set to "0"	RW
CKS2	Frequency Select Bit 2	See **Table 14.2 A/D Conversion Frequency Select**	RW
TRG1	Trigger Select Bit 1	Set to "0" in one-shot mode	RW
— (b7-b6)	Nothing is assigned. When write, set to "0". When read, its content is "0".		—

NOTE:
1. If the ADCON2 register is rewritten during A/D conversion, the conversion result will be indeterminate.

Figure 14.1.1.2 ADCON0 to ADCON2 Registers in One-Shot Mode

Downloaded from Arrow.com.

14.1.2 Repeat mode

In repeat mode, analog voltage applied to a selected pin is repeatedly converted to a digital code. Table 14.1.2.1 shows the repeat mode specifications. Figure 14.1.2.1 shows the operation example in repeat mode. Figure 14.1.2.2 shows the ADCON0 to ADCON2 registers in repeat mode.

Table 14.1.2.1 Repeat Mode Specifications

Item	Specification
Function	The CH2 to CH0 bits in the ADCON0 register and the ADGSEL1 to ADGSEL0 bits in the ADCON2 register select pins. Analog voltage applied to a selected pin is repeatedly converted to a digital code
A/D Conversion Start Condition	• When the TRG bit in the ADCON0 register is “0” (software trigger) Set the ADST bit in the ADCON0 register to “1” (A/D conversion started) • When the TRG bit in the ADCON0 register is “1” (hardware trigger) The $\overline{\text{ADTRG}}$ pin input changes state from “H” to “L” after setting the ADST bit to “1” (A/D conversion started)
A/D Conversion Stop Condition	Set the ADST bit to “0” (A/D conversion halted)
Interrupt Request Generation Timing	None generated
Analog Input Pin	Select one pin from AN_0 to AN_7, AN_{30} to AN_{32} and AN_{24}
Readout of A/D Conversion Result	Readout one of the AD0 to AD7 registers that corresponds to the selected pin

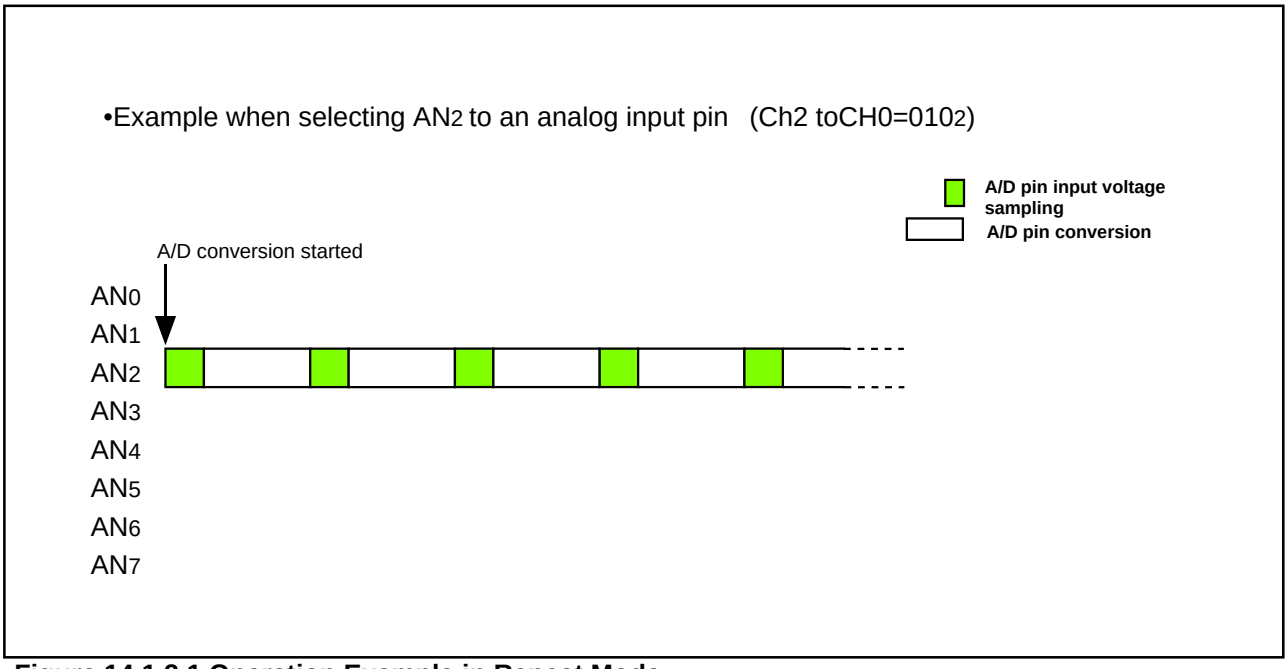
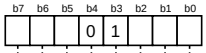


Figure 14.1.2.1 Operation Example in Repeat Mode

Downloaded from Arrow.com.

A/D control register 0 (1)



Symbol: ADCON0 | Address: 03D6$_{16}$ | After reset: 00000XXX$_2$

Bit symbol	Bit name	Function	RW
CH0	Analog Input Pin Select Bit (2, 3)	b2 b1 b0 0 0 0 : Select AN$_0$ 0 0 1 : Select AN$_1$ 0 1 0 : Select AN$_2$ 0 1 1 : Select AN$_3$ 1 0 0 : Select AN$_4$ 1 0 1 : Select AN$_5$ 1 1 0 : Select AN$_6$ 1 1 1 : Select AN$_7$	RW
CH1			RW
CH2			RW
MD0	A/D Operation Mode Select Bit 0 (3)	b4 b3 0 1 : Repeat mode	RW
MD1			RW
TRG	Trigger Select Bit	0 : Software trigger 1 : Hardware trigger ($\overline{\text{AD}}_{\text{TRG}}$ trigger)	RW
ADST	A/D Conversion Start Flag	0 : A/D conversion disabled 1 : A/D conversion started	RW
CKS0	Frequency Select Bit 0	Refer to **Table 14.2 A/D Conversion Frequency Select**	RW

NOTES:
1. If the ADCON0 register is rewritten during A/D conversion, the conversion result will be indeterminate.
2. AN$_{30}$ to AN$_{32}$ and AN$_{24}$ can be used in the same way as AN$_0$ to AN$_7$. Use the ADGSEL1 to ADGSEL0 bits inthe ADCON2 register to select the desired pin.
3. After rewriting the MD1 to MD0 bits, set the CH2 to CH0 bits over again using an another instruction.

A/D control register 1 (1)

Symbol: ADCON1 | Address: 03D7$_{16}$ | After reset: 00$_{16}$

Bit symbol	Bit name	Function	RW
SCAN0	A/D Sweep Pin Select Bit	Invalid in repeat mode	RW
SCAN1			RW
MD2	A/D Operation Mode Select Bit 1	0 : Any mode other than repeat sweep mode 1	RW
BITS	8/10-Bit Mode Select Bit	0 : 8-bit mode 1 : 10-bit mode	RW
CKS1	Frequency Select Bit 1	Refer to **Table 14.2 A/D Conversion Frequency Select**	RW
VCUT	V$_{\text{REF}}$ connect bit (2)	1 : V$_{\text{REF}}$ connected	RW
— (b7-b6)	Nothing is assigned. When write, set to “0”. When read, its content is “0”.		—

NOTES:
1. If the ADCON1 register is rewritten during A/D conversion, the conversion result will be indeterminate.
2. If the VCUT bit is reset from “0” (V$_{\text{REF}}$ unconnected) to “1” (V$_{\text{REF}}$ connected), wait for 1 µs or more before starting A/D conversion.

A/D control register 2 (1)

Symbol: ADCON2 | Address: 03D4$_{16}$ | After reset: 00$_{16}$

Bit symbol	Bit name	Function	RW
SMP	A/D Conversion Method Select Bit	0 : Without sample and hold 1 : With sample and hold	RW
ADGSEL0	A/D Input Group Select Bit	b2 b1 0 0 : Select port P10 group (AN$_i$) 0 1 : Select port P9 group (AN$_{3i}$) 1 0 : Do not set 1 1 : Select port P9 group (AN$_{24}$)	RW
ADGSEL1			RW
— (b3)	Reserved Bit	Set to “0”	RW
CKS2	Frequency Select Bit 2	See **Table 14.2 A/D Conversion Frequency Select**	RW
TRG1	Trigger Select Bit 1	Set to "0" in repeat mode	RW
— (b7-b6)	Nothing is assigned. When write, set to “0”. When read, its content is “0”.		—

NOTE:
1. If the ADCON2 register is rewritten during A/D conversion, the conversion result will be indeterminate.

Figure 14.1.2.2 ADCON0 to ADCON2 Registers in Repeat Mode

Downloaded from Arrow.com.

14.1.3 Single Sweep Mode

In single sweep mode, analog voltages applied to the selected pins are converted one-by-one to a digital code. Table 14.1.3.1 shows the single sweep mode specifications. Figure 14.1.3.1 shows the operation example in single sweep mode. Figure 14.1.3.2 shows the ADCON0 to ADCON2 registers in single sweep mode.

Table 14.1.3.1 Single Sweep Mode Specifications

Item	Specification
Function	The SCAN1 to SCAN0 bits in the ADCON1 register and the ADGSEL1 to ADGSEL0 bits in the ADCON2 register select pins. Analog voltage applied to the selected pins is converted one-by-one to a digital code
A/D Conversion Start Condition	• When the TRG bit in the ADCON0 register is “0” (software trigger) Set the ADST bit in the ADCON0 register to “1” (A/D conversion started) • When the TRG bit in the ADCON0 register is “1” (hardware trigger) The $\overline{\text{ADTRG}}$ pin input changes state from “H” to “L” after setting the ADST bit to “1” (A/D conversion started)
A/D Conversion Stop Condition	• A/D conversion completed(When selecting a software trigger, the ADST bit is set to “0” (A/D conversion halted)). • Set the ADST bit to “0”
Interrupt Request Generation Timing	A/D conversion completed
Analog Input Pin	Select from AN_0 to AN_1 (2 pins), AN_0 to AN_3 (4 pins), AN_0 to AN_5 (6 pins), AN_0 to AN_7 (8 pins) (1)
Readout of A/D Conversion Result	Readout one of the AD0 to AD7 registers that corresponds to the selected pin

NOTE:
1. AN_{30} to AN_{32} can be used in the same way as AN_0 to AN_7. However, all input pins need to belong to the same group.

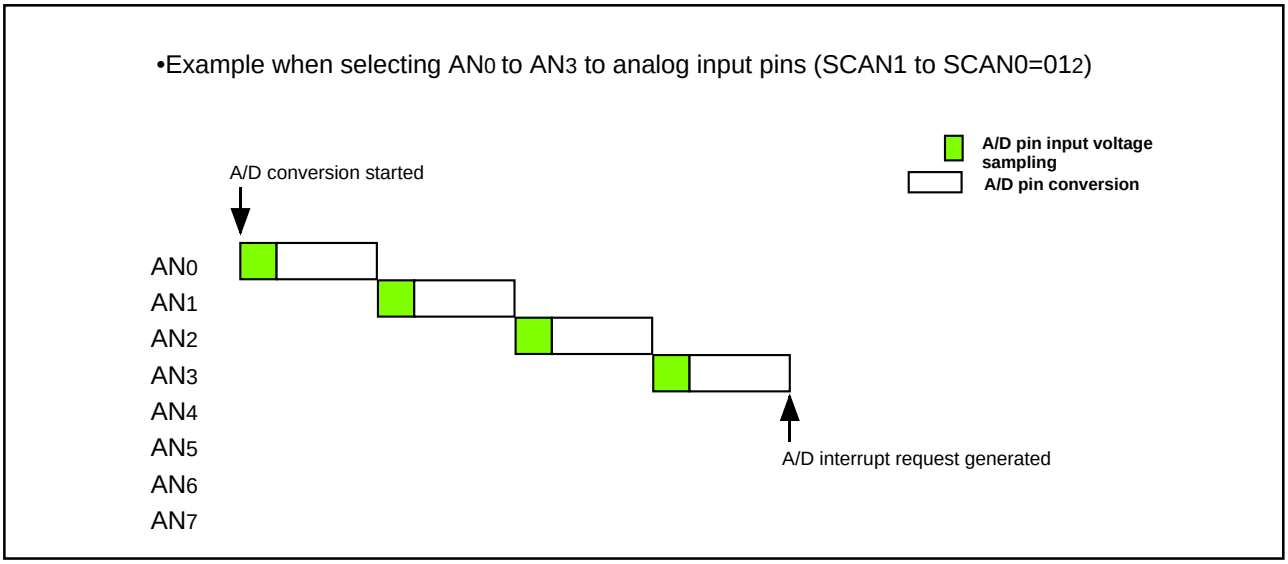


Figure 14.1.3.1 Operation Example in Single Sweep Mode

Downloaded from Arrow.com.

A/D control register 0 (1)

b7	b6	b5	b4	b3	b2	b1	b0
			1	0			

Symbol: ADCON0 — Address: 03D6$_{16}$ — After reset: 00000XXX$_2$

Bit symbol	Bit name	Function	RW
CH0	Analog Input Pin Select Bit	Invalid in single sweep mode	RW
CH1			RW
CH2			RW
MD0	A/D Operation Mode Select Bit 0	b4 b3 1 0 : Single sweep mode or simultaneous sample sweep mode	RW
MD1			RW
TRG	Trigger Select Bit	0 : Software trigger 1 : Hardware trigger ($\overline{\text{AD TRG}}$ trigger)	RW
ADST	A/D Conversion Start Flag	0 : A/D conversion disabled 1 : A/D conversion started	RW
CKS0	Frequency Select Bit 0	Refer to **Table 14.2 A/D Conversion Frequency Select**	RW

NOTE:
1. If the ADCON0 register is rewritten during A/D conversion, the conversion result will be indeterminate.

A/D control register 1 (1)

b7	b6	b5	b4	b3	b2	b1	b0
X	X	1			0		

Symbol: ADCON1 — Address: 03D7$_{16}$ — After reset: 00$_{16}$

Bit symbol	Bit name	Function	RW
SCAN0	A/D Sweep Pin Select Bit (2)	When selecting single sweep mode b1 b0 0 0 : AN$_0$ to AN$_1$ (2 pins) 0 1 : AN$_0$ to AN$_3$ (4 pins) 1 0 : AN$_0$ to AN$_5$ (6 pins) 1 1 : AN$_0$ to AN$_7$ (8 pins)	RW
SCAN1			RW
MD2	A/D Operation Mode Select Bit 1	0 : Any mode other than repeat sweep mode 1	RW
BITS	8/10-Bit Mode Select Bit	0 : 8-bit mode 1 : 10-bit mode	RW
CKS1	Frequency Select Bit 1	Refer to **Table 14.2 A/D Conversion Frequency Select**	RW
VCUT	V$_{REF}$ Connect Bit (3)	1 : V$_{REF}$ connected	RW
— (b7-b6)	Nothing is assigned. When write, set to “0”. When read, its content is “0”.		—

NOTES:
1. If the ADCON1 register is rewritten during A/D conversion, the conversion result will be indeterminate.
2. AN$_{30}$ to AN$_{32}$ can be used in the same way as AN$_0$ to AN$_7$. Use the ADGSEL1 to ADGSEL0 bits in the ADCON2 register to select the desired pin.
3. If the VCUT bit is reset from “0” (V$_{REF}$ unconnected) to “1” (V$_{REF}$connected), wait for 1 µs or more before starting A/D conversion.

A/D control register 2 (1)

b7	b6	b5	b4	b3	b2	b1	b0
X	X	0		0			

Symbol: ADCON2 — Address: 03D4$_{16}$ — After reset: 00$_{16}$

Bit symbol	Bit name	Function	RW
SMP	A/D Conversion Method Select Bit	0 : Without sample and hold 1 : With sample and hold	RW
ADGSEL0	A/D Input Group Select Bit	b2 b1 0 0 : Select port P10 group (AN$_i$) 0 1 : Select port P9 group (AN$_{3i}$) 1 0 : Do not set 1 1 : Do not set	RW
ADGSEL1			RW
— (b3)	Reserved Bit	Set to “0”	RW
CKS2	Frequency Select Bit 2	Refer to **Table 14.2 A/D Conversion Frequency Select**	RW
TRG1	Trigger Select Bit 1	Set to "0" in single sweep mode	RW
— (b7-b6)	Nothing is assigned. When write, set to “0”. When read, its content is “0”.		—

NOTE:
1. If the ADCON2 register is rewritten during A/D conversion, the conversion result will be indeterminate.

Figure 14.1.3.2 ADCON0 to ADCON2 Registers in Single Sweep Mode

Downloaded from Arrow.com.

14.1.4 Repeat Sweep Mode 0

In repeat sweep mode 0, analog voltages applied to the selected pins are repeatedly converted to a digital code. Table 14.1.4.1 shows the repeat sweep mode 0 specifications. Figure 14.1.4.1 shows the operation example in repeat sweep mode 0. Figure 14.1.4.2 shows the ADCON0 to ADCON2 registers in repeat sweep mode 0.

Table 14.1.4.1 Repeat Sweep Mode 0 Specifications

Item	Specification
Function	The SCAN1 to SCAN0 bits in the ADCON1 register and the ADGSEL1 to ADGSEL0 bits in the ADCON2 register select pins. Analog voltage applied to the selected pins is repeatedly converted to a digital code
A/D Conversion Start Condition	• When the TRG bit in the ADCON0 register is “0” (software trigger) Set the ADST bit in the ADCON0 register to “1” (A/D conversion started) • When the TRG bit in the ADCON0 register is “1” (Hardware trigger) The $\overline{\text{ADTRG}}$ pin input changes state from “H” to “L” after setting the ADST bit to “1” (A/D conversion started)
A/D Conversion Stop Condition	Set the ADST bit to “0” (A/D conversion halted)
Interrupt Request Generation Timing	None generated
Analog Input Pin	Select from AN_0 to AN_1 (2 pins), AN_0 to AN_3 (4 pins), AN_0 to AN_5 (6 pins), AN_0 to AN_7 (8 pins) (1)
Readout of A/D Conversion Result	Readout one of the AD0 to AD7 registers that corresponds to the selected pin

NOTE:

1. AN_{30} to AN_{32} can be used in the same way as AN_0 to AN_7. However, all input pins need to belong to the same group.

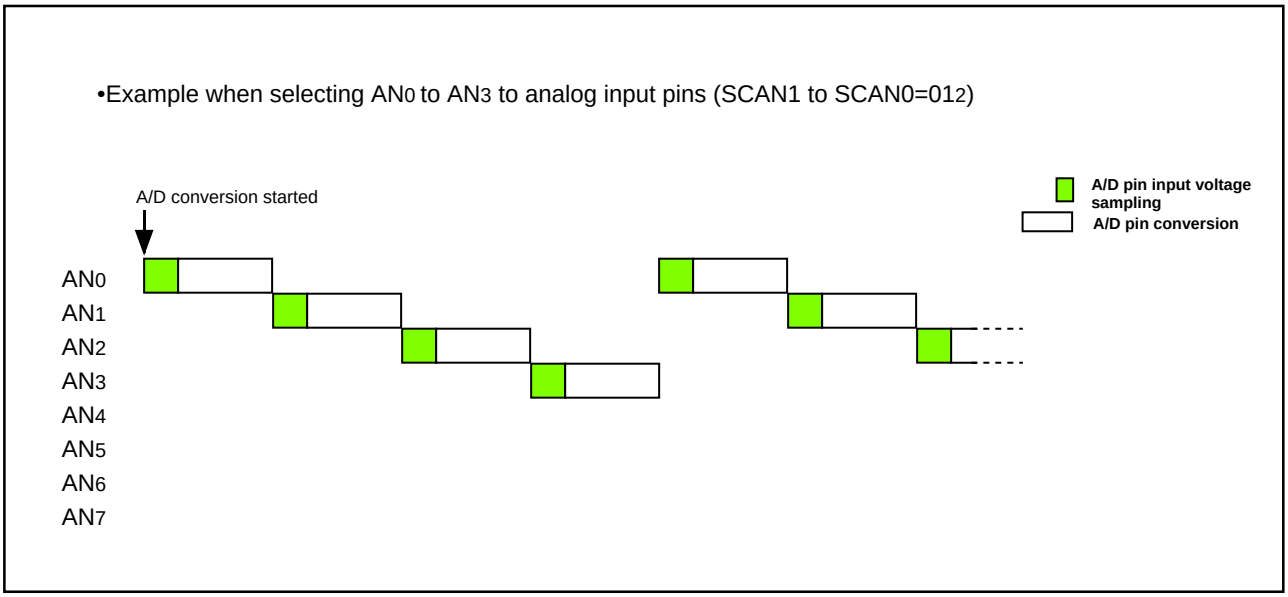


Figure 14.1.4.1 Operation Example in Repeat Sweep Mode 0

Downloaded from Arrow.com.

A/D control register 0 (1)

b7	b6	b5	b4	b3	b2	b1	b0
			1	1			

Symbol: ADCON0 — Address: 03D6$_{16}$ — After reset: 00000XXX$_2$

Bit symbol	Bit name	Function	RW
CH0	Analog Input Pin Select Bit	Invalid in repeat sweep mode 0	RW
CH1			RW
CH2			RW
MD0	A/D Operation Mode Select Bit 0	b4 b3 1 1 : Repeat sweep mode 0 or Repeat sweep mode 1	RW
MD1			RW
TRG	Trigger Select Bit	0 : Software trigger 1 : Hardware trigger ($\overline{\text{AD}}_{\text{TRG}}$ trigger)	RW
ADST	A/D Conversion Start Flag	0 : A/D conversion disabled 1 : A/D conversion started	RW
CKS0	Frequency Select Bit 0	Refer to **Table 14.2 A/D Conversion Frequency Select**	RW

NOTE:
1. If the ADCON0 register is rewritten during A/D conversion, the conversion result will be indeterminate.

A/D control register 1 (1)

b7	b6	b5	b4	b3	b2	b1	b0
X	X	1			0		

Symbol: ADCON1 — Address: 03D7$_{16}$ — After reset: 00$_{16}$

Bit symbol	Bit name	Function	RW
SCAN0	A/D Sweep Pin Select Bit (2)	When selecting repeat sweep mode 0 b1 b0 0 0 : AN$_0$ to AN$_1$ (2 pins) 0 1 : AN$_0$ to AN$_3$ (4 pins) 1 0 : AN$_0$ to AN$_5$ (6 pins) 1 1 : AN$_0$ to AN$_7$ (8 pins)	RW
SCAN1			RW
MD2	A/D Operation Mode Select Bit 1	0 : Any mode other than repeat sweep mode 1	RW
BITS	8/10-Bit Mode Select Bit	0 : 8-bit mode 1 : 10-bit mode	RW
CKS1	Frequency Select Bit 1	Refer to **Table 14.2 A/D Conversion Frequency Select**	RW
VCUT	V_{REF} Connect Bit (3)	1 : V_{REF} connected	RW
— (b7-b6)	Nothing is assigned. When write, set to “0”. When read, its content is “0”.		—

NOTES:
1. If the ADCON1 register is rewritten during A/D conversion, the conversion result will be indeterminate.
2. AN$_{30}$ to AN$_{32}$ can be used in the same way as AN$_0$ to AN$_7$. Use the ADGSEL1 to ADGSET0 bits in the ADCON2 register to select the desired pin.
3. If the VCUT bit is reset from “0” (V_{REF} unconnected) to “1” (V_{REF} connected), wait for 1 µs or more before starting A/D conversion.

A/D control register 2 (1)

b7	b6	b5	b4	b3	b2	b1	b0
X	X	0		0			

Symbol: ADCON2 — Address: 03D4$_{16}$ — After reset: 00$_{16}$

Bit symbol	Bit name	Function	RW
SMP	A/D Conversion Method Select Bit	0 : Without sample and hold 1 : With sample and hold	RW
ADGSEL0	A/D Input Group Select Bit	b2 b1 0 0 : Select port P10 group (AN$_i$) 0 1 : Select port P9 group (AN$_{3i}$) 1 0 : Do not set 1 1 : Do not set	RW
ADGSEL1			RW
— (b3)	Reserved Bit	Set to “0”	RW
CKS2	Frequency Select Bit 2	Refer to **Table 14.2 A/D Conversion Frequency Select**	RW
TRG1	Trigger Select Bit 1	Set to "0" in repeat sweep mode 0	RW
— (b7-b6)	Nothing is assigned. When write, set to “0”. When read, its content is “0”.		—

NOTE:
1. If the ADCON2 register is rewritten during A/D conversion, the conversion result will be indeterminate.

Figure 14.1.4.2 ADCON0 to ADCON2 Registers in Repeat Sweep Mode 0

Downloaded from Arrow.com.

14.1.5 Repeat Sweep Mode 1

In repeat sweep mode 1, analog voltages applied to the all selected pins are converted to a digital code, with mainly used in the selected pins. Table 14.1.5.1 shows the repeat sweep mode 1 specifications. Figure 14.1.5.1 shows the operation example in repeat sweep mode 1. Figure 14.1.5.2 shows the ADCON0 to ADCON2 registers in repeat sweep mode 1.

Table 14.1.5.1 Repeat Sweep Mode 1 Specifications

Item	Specification
Function	The SCAN1 to SCAN0 bits in the ADCON1 register and the ADGSEL1 to ADGSEL0 bits in the ADCON2 register mainly select pins. Analog voltage applied to the all selected pins is repeatedly converted to a digital code Example : When selecting AN_0 Analog voltage is converted to a digital code in the following order $AN_0 \rightarrow AN_1 \rightarrow AN_0 \rightarrow AN_2 \rightarrow AN_0 \rightarrow AN_3$, and so on.
A/D Conversion Start Condition	• When the TRG bit in the ADCON0 register is “0” (software trigger) Set the ADST bit in the ADCON0 register to “1” (A/D conversion started) • When the TRG bit in the ADCON0 register is “1” (hardware trigger) The $\overline{ADTRG}$ pin input changes state from “H” to “L” after setting the ADST bit to “1” (A/D conversion started)
A/D Conversion Stop Condition	Set the ADST bit to “0” (A/D conversion halted)
Interrupt Request Generation Timing	None generated
Analog Input Pins Mainly Used in A/D Conversions	Select from AN_0 (1 pins), AN_0 to AN_1 (2 pins), AN_0 to AN_2 (3 pins), AN_0 to AN_3 (4 pins) (1)
Readout of A/D Conversion Result	Readout one of the AD0 to AD7 registers that corresponds to the selected pin

NOTE:

1. AN_{30} to AN_{32} can be used in the same way as AN_0 to AN_7. However, all input pins need to belong to the same group.

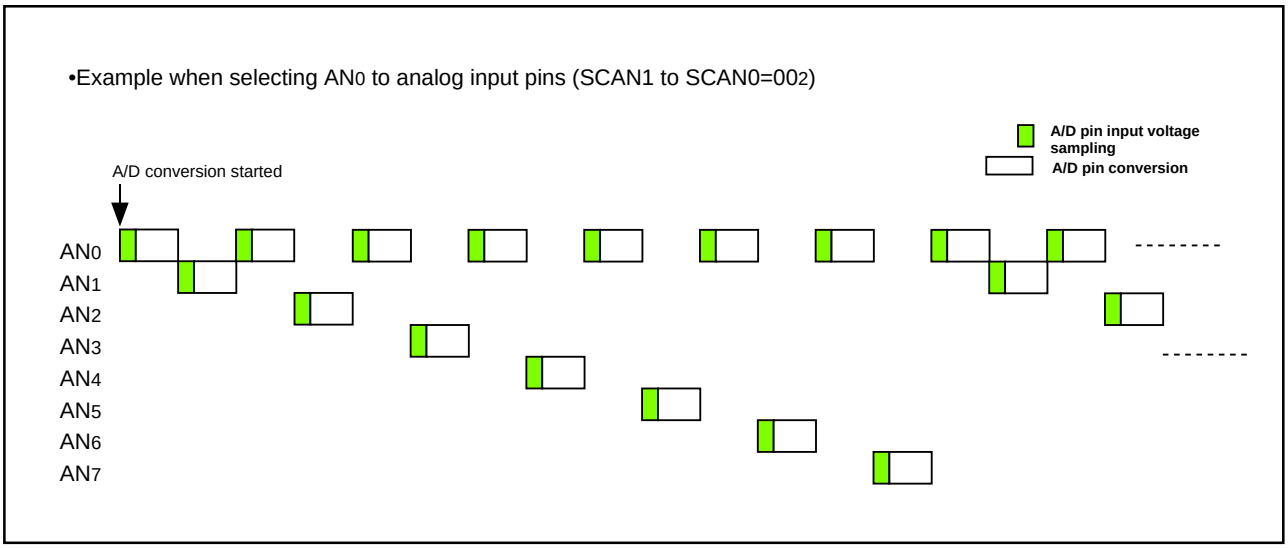
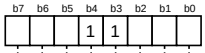


Figure 14.1.5.1 Operation Example in Repeat Sweep Mode 1

Downloaded from Arrow.com.

A/D control register 0 (1)



Symbol ADCON0 | Address 03D6₁₆ | After reset 00000XXX₂

Bit symbol	Bit name	Function	RW
CH0	Analog Input Pin Select Bit	Invalid in repeat sweep mode 1	RW
CH1			RW
CH2			RW
MD0	A/D Operation Mode Select Bit 0	b4 b3 1 1 : Repeat sweep mode 0 or Repeat sweep mode 1	RW
MD1			RW
TRG	Trigger Select Bit	0 : Software trigger 1 : Hardware trigger (AD TRG trigger)	RW
ADST	A/D Conversion Start Flag	0 : A/D conversion disabled 1 : A/D conversion started	RW
CKS0	Frequency Select Bit 0	Refer to **Table 14.2 A/D Conversion Frequency Select**	RW

NOTE:
1. If the ADCON0 register is rewritten during A/D conversion, the conversion result will be indeterminate.

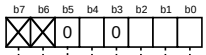
A/D control register 1 (1)

Symbol ADCON1 | Address 03D7₁₆ | After reset 00₁₆

Bit symbol	Bit name	Function	RW
SCAN0	A/D Sweep Pin Select Bit (2)	When selecting repeat sweep mode 1 b1 b0 0 0 : AN0 (1 pin) 0 1 : AN0 to AN1 (2 pins) 1 0 : AN0 to AN2 (3 pins) 1 1 : AN0 to AN3 (4 pins)	RW
SCAN1			RW
MD2	A/D Operation Mode Select Bit 1	1 : Repeat sweep mode 1	RW
BITS	8/10-Bit Mode Select Bit	0 : 8-bit mode 1 : 10-bit mode	RW
CKS1	Frequency Select Bit 1	Refer to **Table 14.2 A/D Conversion Frequency Select**	RW
VCUT	VREF Connect Bit (3)	1 : VREF connected	RW
— (b7-b6)	Nothing is assigned. When write, set to “0”. When read, its content is “0”.		—

NOTES:
1. If the ADCON1 register is rewritten during A/D conversion, the conversion result will be indeterminate.
2. AN30 to AN32 can be used in the same way as AN0 to AN7. Use the ADGSEL1 to ADGSEL0 bits in the ADCON2 register to select the desired pin.
3. If the VCUT bit is reset from “0” (VREF unconnected) to “1” (VREF connected), wait for 1 µs or more before starting A/D conversion.

A/D control register 2 (1)



Symbol ADCON2 | Address 03D4h | After reset 00h

Bit symbol	Bit name	Function	RW
SMP	A/D Conversion Method Select Bit	0 : Without sample and hold 1 : With sample and hold	RW
ADGSEL0	A/D Input Group Select Bit	b2 b1 0 0 : Select port P10 group (AN i) 0 1 : Select port P9 group (AN 3i) 1 0 : Do not set 1 1 : Do not set	RW
ADGSEL1			RW
— (b3)	Reserved Bit	Set to “0”	RW
CKS2	Frequency Select Bit 2	Refer to **Table 14.2 A/D Conversion Frequency Select**	RW
TRG1	Trigger Select Bit 1	Set to "0" in repeat sweep mode 1	RW
— (b7-b6)	Nothing is assigned. When write, set to “0”. When read, its content is “0”.		—

NOTE:
1. If the ADCON2 register is rewritten during A/D conversion, the conversion result will be indeterminate.

Figure 14.1.5.2 ADCON0 to ADCON2 Registers in Repeat Sweep Mode 1

Downloaded from Arrow.com.

14.1.6 Simultaneous Sample Sweep Mode

In simultaneous sample sweep mode, analog voltages applied to the selected pins are converted one-by-one to a digital code. At this time, the input voltage of AN0 and AN1 are sampled simultaneously using two circuits of sample and hold circuit. Table 14.1.6.1 shows the simultaneous sample sweep mode specifications. Figure 14.1.6.1 shows the operation example in simultaneous sample sweep mode. Figure 14.1.6.2 shows ADCON0 to ADCON2 registers and Figure 14.1.6.3 shows ADTRGCON registers in simultaneous sample sweep mode. Table 14.1.6.2 shows the trigger select bit setting in simultaneous sample sweep mode. In simultaneous sample sweep mode, Timer B0 underflow can be selected as a trigger by combining software trigger, $\overline{\text{ADTRG}}$ trigger, Timer B2 underflow, Timer B2 interrupt generation frequency setting counter underflow or A/D trigger mode of Timer B.

Table 14.1.6.1 Simultaneous Sample Sweep Mode Specifications

Item	Specification
Function	The SCAN1 to SCAN0 bits in the ADCON1 register and ADGSEL1 to ADGSEL0 bits in the ADCON2 register select pins. Analog voltage applied to the selected pins is converted one-by-one to a digital code. At this time, the input voltage of AN0 and AN1 are sampled simultaneously.
A/D Conversion Start Condition	When the TRG bit in the ADCON0 register is "0" (software trigger) Set the ADST bit in the ADCON0 register to “1” (A/D conversion started) When the TRG bit in the ADCON0 register is "1" (hardware trigger) The trigger is selected by TRG1 and HPTRG0 bits (See **Table 14.1.6.2**) The $\overline{\text{ADTRG}}$ pin input changes state from “H” to “L” after setting the ADST bit to “1” (A/D conversion started) Timer B0, B2 or Timer B2 interrupt generation frequency setting counter underflow after setting the ADST bit to “1” (A/D conversion started)
A/D Conversion Stop Condition	A/D conversion completed (If selecting software trigger, the ADST bit is automatically set to "0". Set the ADST bit to "0" (A/D conversion halted)
Interrupt Generation Timing	A/D conversion completed
Analog Input Pin	Select from AN0 to AN1 (2 pins), AN0 to AN3 (4 pins), AN0 to AN5 (6 pins), or AN0 to AN7 (8 pins) (1)
Readout of A/D conversion result	Readout one of the AN0 to AN7 registers that corresponds to the selected pin

NOTE:

1. AN30 to AN32 can be used in the same way as AN0 to AN7. However, all input pins need to belong to the same group.

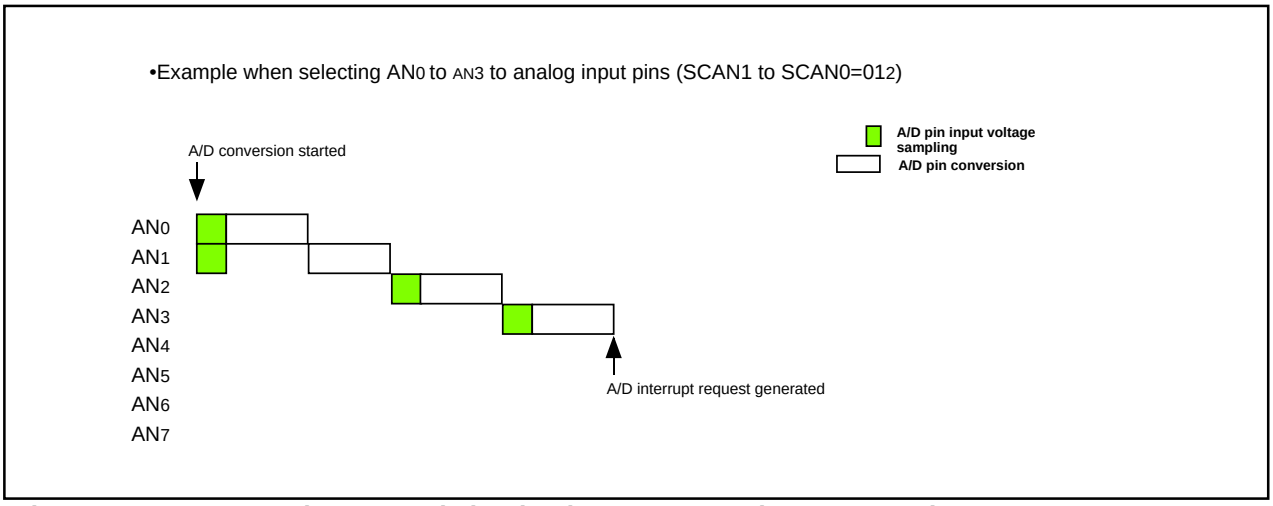


Figure 14.1.6.1 Operation Example in Simultaneous Sample Sweep Mode

Downloaded from Arrow.com.

A/D control register 0 (1)

b7	b6	b5	b4	b3	b2	b1	b0
			1	0			

Symbol: ADCON0 — Address: 03D6$_{16}$ — After reset: 00000XXX$_2$

Bit symbol	Bit name	Function	RW
CH0	Analog Input Pin Select Bit	Invalid in simultaneous sample sweep mode	RW
CH1			RW
CH2			RW
MD0	A/D Operation Mode Select Bit 0	b4 b3 1 0 : Single sweep mode or simultaneous sample sweep mode	RW
MD1			RW
TRG	Trigger Select Bit	Refer to **Table 14.1.6.2 Trigger Select Bit Setting in Simultaneous Sample Sweep Mode**	RW
ADST	A/D Conversion Start Fag	0 : A/D conversion disabled 1 : A/D conversion started	RW
CKS0	Frequency Select Bit 0	Refer to **Table 14.2 A/D Conversion Frequency Select**	RW

NOTE:
1. If the ADCON0 register is rewritten during A/D conversion, the conversion result will be indeterminate.

A/D control register 1 (1)

b7	b6	b5	b4	b3	b2	b1	b0
×	×	1			0		

Symbol: ADCON1 — Address: 03D7$_{16}$ — After reset: 00$_{16}$

Bit symbol	Bit name	Function	RW
SCAN0	A/D Sweep Pin Select Bit (2)	When selecting simultaneous sample sweep mode b1 b0 0 0 : AN$_0$ to AN$_1$ (2 pins) 0 1 : AN$_0$ to AN$_3$ (4 pins) 1 0 : AN$_0$ to AN$_5$ (6 pins) 1 1 : AN$_0$ to AN$_7$ (8 pins)	RW
SCAN1			RW
MD2	A/D Operation Mode Select Bit 1	0 : Any mode other than repeat sweep mode 1	RW
BITS	8/10-Bit Mode Select Bit	0 : 8-bit mode 1 : 10-bit mode	RW
CKS1	Frequency Select Bit 1	Refer to **Table 14.2 A/D Conversion Frequency Select**	RW
VCUT	V$_{REF}$ Connect Bit (3)	1 : V$_{REF}$ connected	RW
— (b7-b6)	Nothing is assigned. When write, set to “0”. When read, its content is “0”.		—

NOTES:
1. If the ADCON1 register is rewritten during A/D conversion, the conversion result will be indeterminate.
2. AN$_{30}$ to AN$_{32}$ can be used in the same way as AN$_0$ to AN$_7$. Use the ADGSEL1 to ADGSET0 bits in the ADCON2 register to select the desired pin.
3. If the VCUT bit is reset from “0” (V$_{REF}$ unconnected) to “1” (V$_{REF}$ connected), wait for 1 µs or more before starting A/D conversion.

A/D control register 2 (1)

b7	b6	b5	b4	b3	b2	b1	b0
×	×			0			1

Symbol: ADCON2 — Address: 03D4$_{16}$ — After reset: 00$_{16}$

Bit symbol	Bit name	Function	RW
SMP	A/D Conversion Method Select Bit	Set to “1” in simultaneous sample sweep mode	RW
ADGSEL0	A/D Input Group Select Bit	b2 b1 0 0 : Select port P10 group (AN$_i$) 0 1 : Select port P9 group (AN$_{3i}$) 1 0 : Do not set 1 1 : Do not set	RW
ADGSEL1			RW
— (b3)	Reserved Bit	Set to “0”	RW
CKS2	Frequency Select Bit 2	Refer to **Table 14.2 A/D Conversion Frequency Select**	RW
TRG1	Trigger select bit 1	Refer to **Table 14.1.6.2 Trigger Select Bit Setting in Simultaneous Sample Sweep Mode**	RW
— (b7-b6)	Nothing is assigned. When write, set to “0”. When read, its content is “0”.		—

NOTE:
1. If the ADCON2 register is rewritten during A/D conversion, the conversion result will be indeterminate.

Figure 14.1.6.2 ADCON0 to ADCON2 Registers for Simultaneous Sample Sweep Mode

Downloaded from Arrow.com.

A/D trigger control register (1)

b7	b6	b5	b4	b3	b2	b1	b0
X	X	X	X	0		0	1

Symbol	Address	After reset
ADTRGCON	03D2$_{16}$	00$_{16}$

Bit symbol	Bit name	Function	RW
SSE	A/D Operation Mode Select Bit 2	1 : Simultaneous sample sweep mode or delayed trigger mode 0, 1	RW
DTE	A/D Operation Mode Select Bit 3	0 : Any mode other than delayed trigger mode 0,1	RW
HPTRG0	AN0 Trigger Select Bit	Refer to **Table 14.1.6.2 Trigger Select Bit Setting in Simultaneous Sample Sweep Mode**	RW
HPTRG1	AN1 Trigger Select Bit	Set to "0" in simultaneous sample sweep mode	RW
— (b7-b4)	Nothing is assigned. When write, set to “0”. When read, its content is “0”.		—

NOTE:
1. If ADTRGCON register is rewritten during A/D conversion, the conversion result will be indeterminate.

Figure 14.1.6.3 ADTRGCON Register in Simultaneous Sample Sweep Mode

Table 14.1.6.2 Trigger Select Bit Setting in Simultaneous Sample Sweep Mode

TRG	TRG1	HPTRG0	TRIGGER
0	-	-	Software trigger
1	-	1	Timer B0 underflow (1)
1	0	0	$\overline{\text{AD}}_{\text{TRG}}$
1	1	0	Timer B2 or Timer B2 interrupt generation frequency setting counter underflow (2)

NOTE:
1. A count can be started for Timer B2, Timer B2 interrupt generation frequency setting counter underflow or the $\overline{\text{INT5}}$ pin falling edge as count start conditions of Timer B0.
2. Select Timer B2 or Timer B2 interrupt generation frequency setting counter using the TB2SEL bit in the TB2SC register.

Downloaded from Arrow.com.

14.1.7 Delayed Trigger Mode 0

In delayed trigger mode 0, analog voltages applied to the selected pins are converted one-by-one to a digital code. The delayed trigger mode 0 used in combination with A/D trigger mode of Timer B. The Timer B0 underflow starts a single sweep conversion. After completing the AN_0 pin conversion, the AN_1 pin is not sampled and converted until the Timer B1 underflow is generated. When the Timer B1 underflow is generated, the single sweep conversion is restarted with the AN_1 pin. Table 14.1.7.1 shows the delayed trigger mode 0 specifications. Figure 14.1.7.1 shows the operation example in delayed trigger mode 0. Figure 14.1.7.2 and Figure 14.1.7.3 show each flag operation in the ADSTAT0 register that corresponds to the operation example. Figure 14.1.7.4 shows the ADCON0 to ADCON2 registers in delayed trigger mode 0. Figure 14.1.7.5 shows the ADTRGCON register in delayed trigger mode 0 and Table 14.1.7.2 shows the trigger select bit setting in delayed trigger mode 0.

Table 14.1.7.1 Delayed Trigger Mode 0 Specifications

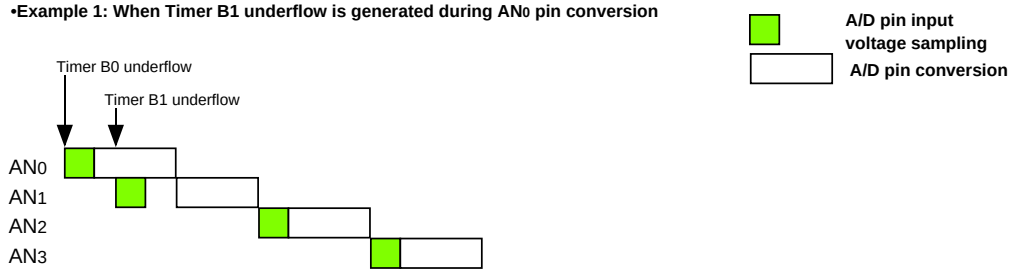
Item	Specification
Function	The SCAN1 to SCAN0 bits in the ADCON1 register and ADGSEL1 to ADGSEL0 bits in the ADCON2 register select pins. Analog voltage applied to the input voltage of the selected pins are converted one-by-one to the digital code. At this time, Timer B0 under flow generation starts AN_0 pin conversion. Timer B1 underflow generation starts con version after the AN_1 pin. (1)
A/D Conversion Start	AN_0 pin conversion start condition •When Timer B0 underflow is generated if Timer B0 underflow is generated again before Timer B1 underflow is generated , the conversion is not affected •When Timer B0 underflow is generated during A/D conversion of pins after the AN_1 pin, conversion is halted and the sweep is restarted from AN_0 pin AN_1 pin conversion start condition •When Timer B1 underflow is generated during A/D conversion of the AN_0 pin, the input voltage of the AN_1 pin is sampled. The AN_1 conversion and the rest of the sweep start when AN_0 conversion is completed.
A/D Conversion Stop Condition	•When single sweep conversion from the AN_0 pin is completed •Set the ADST bit to "0" (A/D conversion halted)(2)
Interrupt Request Generation Timing	A/D conversion completed
Analog Input Pin	Select from AN_0 to AN_1 (2 pins), AN_0 to AN_3 (4 pins), AN_0 to AN_5 (6 pins) and AN_0 to AN_7 (8 pins)(3)
Readout of A/D Conversion Result	Readout one of the AN0 to AN7 registers that corresponds to the selected pins

NOTES:
1. Set the larger value than the value of the timer B0 register to the timer B1 register.
2. Do not write “1” (A/D conversion started) to the ADST bit in delayed trigger mode 0. When write “1”, unexpected interrupts may be generated.
3. AN_{30} to AN_{32} can be used in the same way as AN_0 to AN_7. However, all input pins need to belong to the same group.

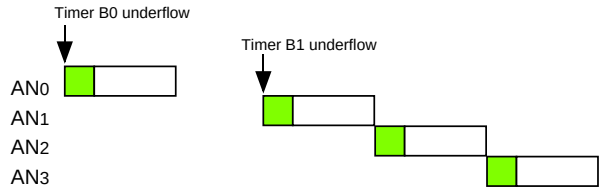
Downloaded from Arrow.com.

•Example when selecting AN0 to AN3 to analog input pins (SCAN1 to SCAN0=012)

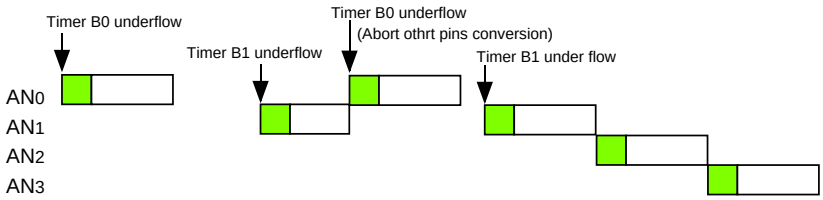
•Example 1: When Timer B1 underflow is generated during AN0 pin conversion



•Example 2: When Timer B1 underflow is generated after AN0 pin conversion



•Example 3: When Timer B0 underflow is generated during A/D conversion of any pins except AN0 pin



•Example 4: When Timer B0 underflow is generated again before Timer B1 underflow is generated after Timer B0 underflow generation

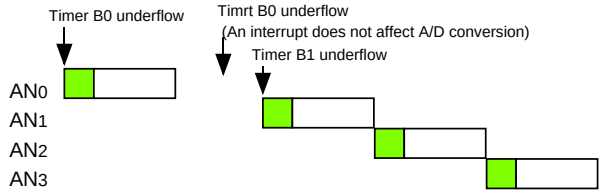
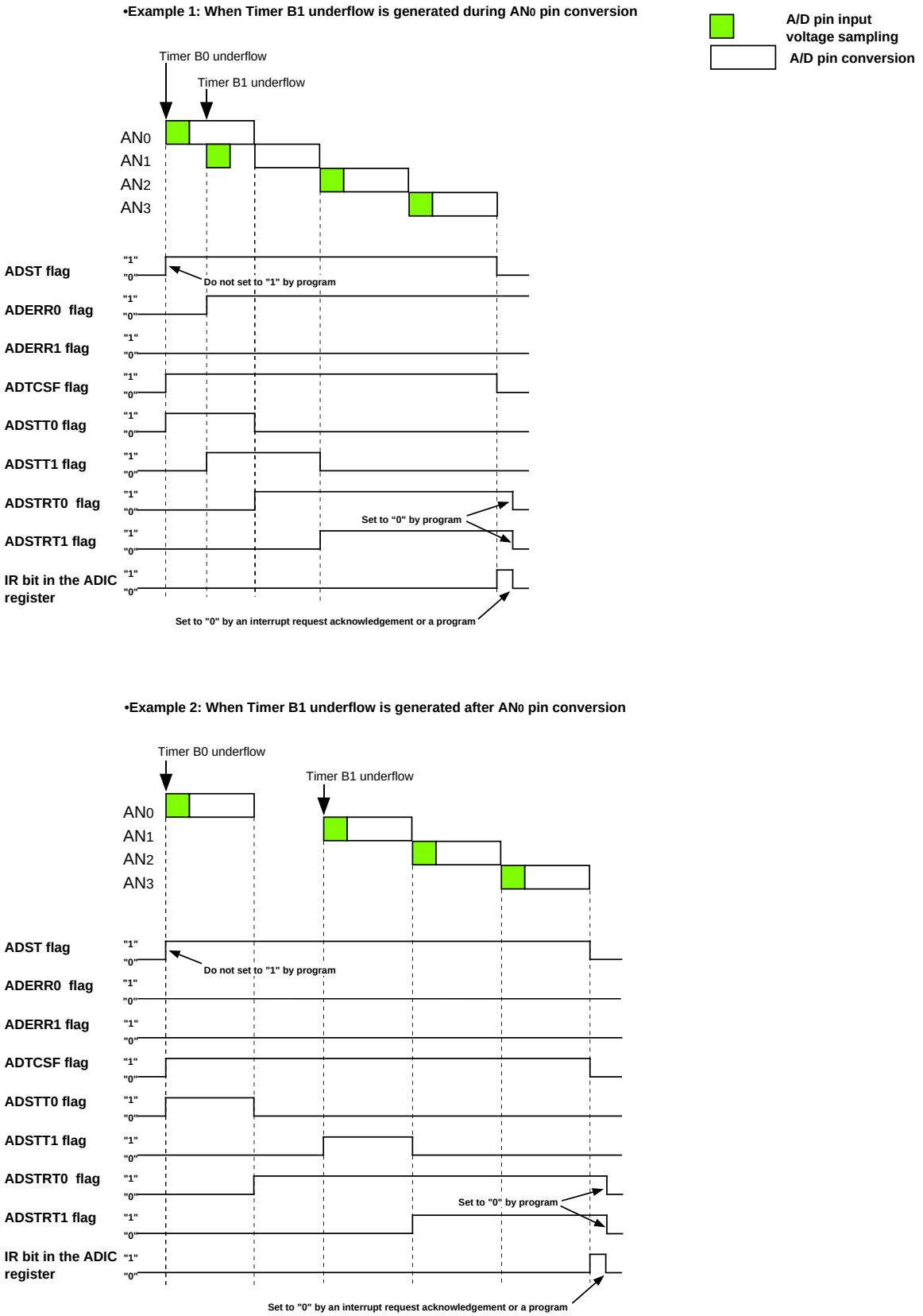


Figure 14.1.7.1 Operation Example in Delayed Trigger Mode 0

Downloaded from Arrow.com.

•Example when selecting AN0 to AN3 to analog input pins (SCAN1 to SCAN0=012)



ADST flag: Bit 6 in the ADCON0 register
ADERR0, ADERR1, ADTCSF, ADSTT0, ADSTT1, ADSTRT0 and ADSTRT1 flag: bits 0, 1, 3, 4, 5, 6 and 7 in the ADSTAT0 register

Figure 14.1.7.2 Each Flag Operation in ADSTAT0 Register Associated with the Operation Example in Delayed Trigger Mode 0 (1)

Downloaded from Arrow.com.

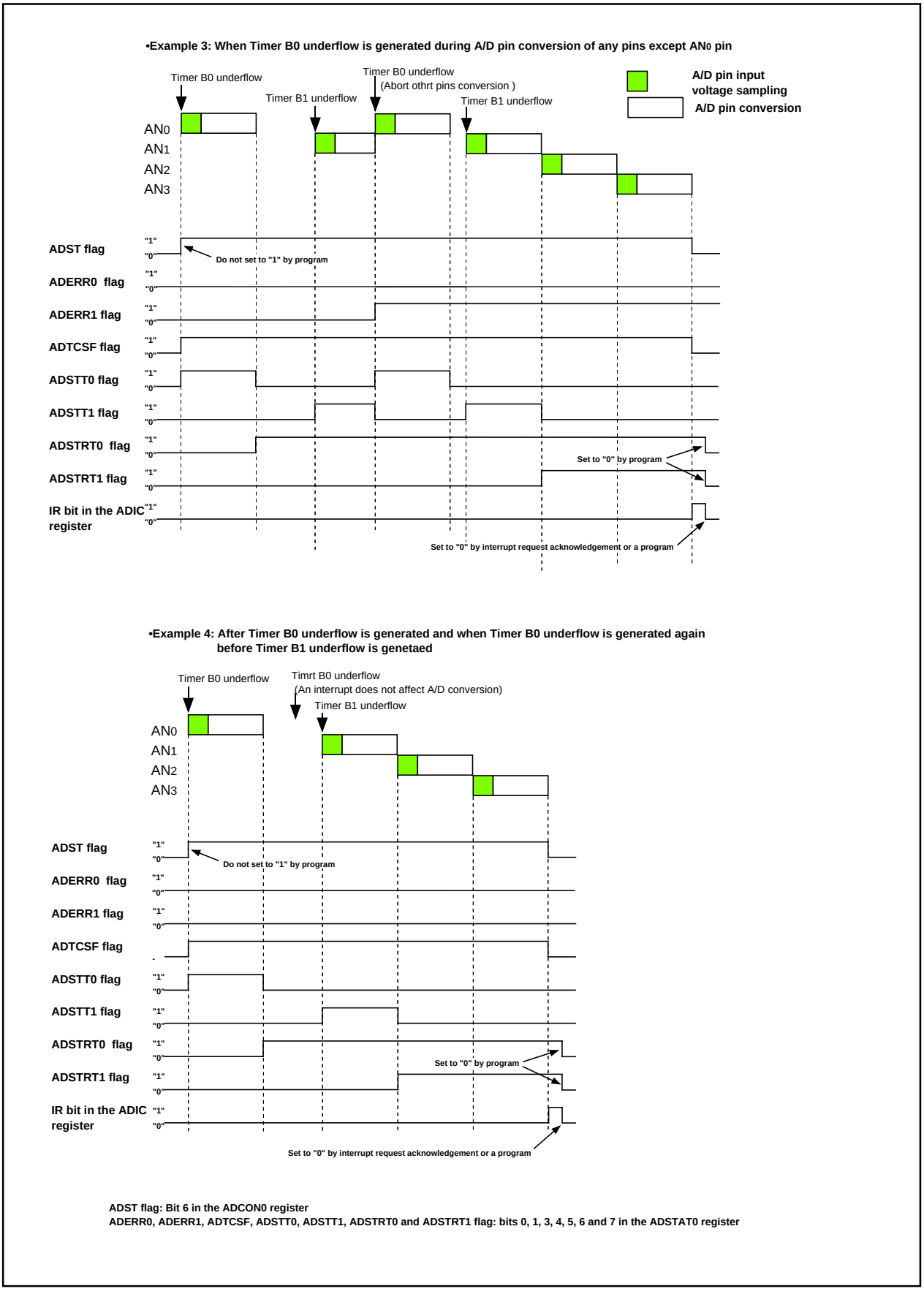


Figure 14.1.7.3 Each Flag Operation in ADSTAT0 Register Associated with the Operation Example in Delayed Trigger Mode 0 (2)

Downloaded from Arrow.com.

A/D control register 0 (1)

b7	b6	b5	b4	b3	b2	b1	b0
		0	0	0	1	1	1

Symbol: ADCON0 — Address: 03D6$_{16}$ — After reset: 00000XXX$_2$

Bit symbol	Bit name	Function	RW
CH0	Analog Input Pin Select Bit	b2 b1 b0 1 1 1 : Set to "111b" in delayed trigger mode 0	RW
CH1			RW
CH2			RW
MD0	A/D Operation Mode Select Bit 0	b4 b3 0 0 : One-shot mode or delayed trigger mode 0,1	RW
MD1			RW
TRG	Trigger Select Bit	Refer to **Table 14.1.7.2 Trigger Select Bit Setting in Delayed Trigger Mode 0**	RW
ADST	A/D Conversion Start Flag (2)	0 : A/D conversion disabled 1 : A/D conversion started	RW
CKS0	Frequency Select Bit 0	Refer to **Table 14.2 A/D Conversion Frequency Select**	RW

NOTES:
1. If the ADCON0 register is rewritten during A/D conversion, the conversion result will be indeterminate.
2. Do not write “1” in delayed trigger mode 0. When write, set to "0".

A/D control register 1 (1)

b7	b6	b5	b4	b3	b2	b1	b0
		1			0		

Symbol: ADCON1 — Address: 03D7$_{16}$ — After reset: 00$_{16}$

Bit symbol	Bit name	Function	RW
SCAN0	A/D Sweep Pin Select Bit (2)	When selecting delayed trigger sweep mode 0 b1 b0 0 0: AN$_0$ to AN$_1$ (2 pins) 0 1: AN$_0$ to AN$_3$ (4 pins) 1 0: AN$_0$ to AN$_5$ (6 pins) 1 1: AN$_0$ to AN$_7$ (8 pins)	RW
SCAN1			RW
MD2	A/D Operation Mode Select Bit 1	0 : Any mode other than repeat sweep mode 1	RW
BITS	8/10-Bit Mode Select Bit	0 : 8-bit mode 1 : 10-bit mode	RW
CKS1	Frequency Select Bit 1	Refer to **Table 14.2 A/D Conversion Frequency Select**	RW
VCUT	V$_{REF}$ Connect Bit (3)	1 : V$_{REF}$ connected	RW
— (b7-b6)	Nothing is assigned. When write, set to “0”. When read, its content is “0”.		—

NOTES:
1. If the ADCON1 register is rewritten during A/D conversion, the conversion result will be indeterminate.
2. AN$_{30}$ to AN$_{32}$ can be used in the same way as AN$_0$ to AN$_7$. Use the ADGSEL1 to ADGSEL0 bits in the ADCON2 register to select the desired pin.
3. If the VCUT bit is reset from “0” (V$_{REF}$ unconnected) to “1” (V$_{REF}$ connected), wait for 1 µs or more before starting A/D conversion.

A/D control register 2 (1)

b7	b6	b5	b4	b3	b2	b1	b0
		0		0			1

Symbol: ADCON2 — Address: 03D4$_{16}$ — After reset: 00$_{16}$

Bit symbol	Bit name	Function	RW
SMP	A/D Conversion Method Select Bit (2)	1 : With sample and hold	RW
ADGSEL0	A/D Input Group Select Bit	b2 b1 0 0 : Select port P10 group (AN $_i$) 0 1 : Select port P9 group (AN $_{3i}$) 1 0 : Do not set 1 1 : Do not set	RW
ADGSEL1			RW
— (b3)	Reserved Bit	Set to “0”	RW
CKS2	Frequency Select Bit 2	Refer to **Table 14.2 A/D Conversion Frequency Select**	RW
TRG1	Trigger Select Bit 1	Refer to **Table 14.1.7.2 Trigger Select Bit Setting in Delayed Trigger Mode 0**	RW
— (b7-b6)	Nothing is assigned. When write, set to “0”. When read, its content is “0”.		—

NOTES:
1. If the ADCON2 register is rewritten during A/D conversion, the conversion result will be indeterminate.
2. Set to “1” in delayed trigger mode 0.

Figure 14.1.7.4 ADCON0 to ADCON2 Registers in Delayed Trigger Mode 0

Downloaded from Arrow.com.

A/D trigger control register (1)

b7	b6	b5	b4	b3	b2	b1	b0
X	X	X	X	1	1	1	1

Symbol	Address	After reset
ADTRGCON	03D2$_{16}$	00$_{16}$

Bit symbol	Bit name	Function	RW
SSE	A/D Operation Mode Select Bit 2	Simultaneous sample sweep mode or delayed trigger mode 0,1	RW
DTE	A/D Operation Mode Select Bit 3	Delayed trigger mode 0, 1	RW
HPTRG0	AN0 Trigger Select Bit	Refer to **Table 14.1.7.2 Trigger Select Bit Setting in Delayed Trigger Mode 0**	RW
HPTRG1	AN1 Trigger Select Bit	Refer to **Table 14.1.7.2 Trigger Select Bit Setting in Delayed Trigger Mode 0**	RW
— (b7-b4)	Nothing is assigned. When write, set to “0”. When read, its content is “0”.		—

NOTE:
1. If ADTRGCON reigster is rewritten during A/D conversion, the conversion result will be indeterminate.

Figure 14.1.7.5 ADTRGCON Register in Delayed Trigger Mode 0

Table 14.1.7.2 Trigger Select Bit Setting in Delayed Trigger Mode 0

TRG	TRG1	HPTRG0	HPTRG1	Trigger
0	0	1	1	Timer B0, B1 underflow

Downloaded from Arrow.com.

14.1.8 Delayed Trigger Mode 1

In delayed trigger mode 1, analog voltages applied to the selected pins are converted one-by-one to a digital code. When the input of the $\overline{\text{ADTRG}}$ pin (falling edge) changes state from “H” to “L”, a single sweep conversion is started. After completing the AN0 pin conversion, the AN1 pin is not sampled and converted until the second $\overline{\text{ADTRG}}$ pin falling edge is generated. When the second $\overline{\text{ADTRG}}$ falling edge is generated, The single sweep conversion of the pins after the AN1 pin is restarted. Table 14.1.8.1 shows the delayed trigger mode 1 specifications. Figure 14.1.8.1 shows the operation example of delayed trigger mode 1. Figure 14.1.8.2 to Figure 14.1.8.3 show each flag operation in the ADSTAT0 register that corresponds to the operation example. Figure 14.1.8.4 shows the ADCON0 to ADCON2 registers in delayed trigger mode 1. Figure 14.1.8.5 shows the ADTRGCON register in delayed trigger mode 1 and Table 15.1.8.2 shows the trigger select bit setting in delayed trigger mode 1.

Table 14.1.8.1 Delayed Trigger Mode 1 Specifications

Item	Specification
Function	The SCAN1 to SCAN0 bits in the ADCON1 register and ADGSEL1 to ADGSEL0 bits in the ADCON2 register select pins. Analog voltages applied to the selected pins are converted one-by-one to a digital code. At this time, the $\overline{\text{ADTRG}}$ pin falling edge starts AN0 pin conversion and the second $\overline{\text{ADTRG}}$ pin falling edge starts conversion of the pins after AN1 pin
A/D Conversion Start Condition	AN0 pin conversion start condition The $\overline{\text{ADTRG}}$ pin input changes state from “H” to “L” (falling edge)(1) AN1 pin conversion start condition (2) The $\overline{\text{ADTRG}}$ pin input changes state from “H” to “L” (falling edge) •When the second $\overline{\text{ADTRG}}$ pin falling edge is generated during or after A/D conversion of the AN0 pin, input voltage of AN1 pin is sampled at the time of $\overline{\text{ADTRG}}$ falling edge. The conversion of AN1 and the rest of the sweep starts when AN0 conversion is completed. •When the $\overline{\text{ADTRG}}$ pin falling edge is generated again during single sweep conver sion of pins after the AN1 pin, the conversion is not affected
A/D Conversion Stop Condition	•A/D conversion completed •Set the ADST bit to "0" (A/D conversion halted)(3)
Interrupt Request Generation Timing	Single sweep conversion completed
Analog Input Pin	Select from AN0 to AN1 (2 pins), AN0 to AN3 (4 pins), AN0 to AN5 (6 pins) and AN0 to AN7 (8 pins)(4)
Readout of A/D Conversion Result	Readout one of the AN0 to AN7 registers that corresponds to the selected pins

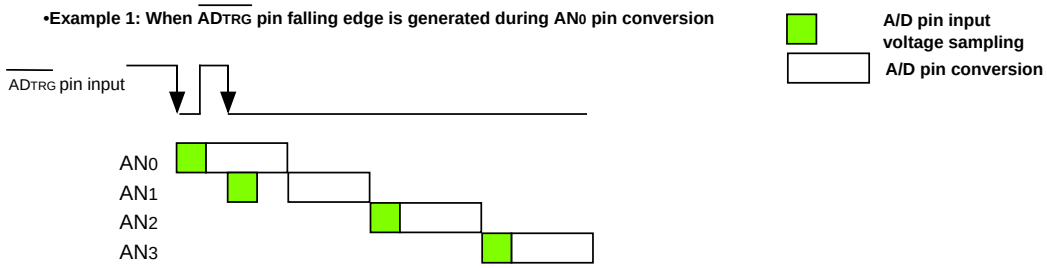
NOTES:

1. Do not generate the next $\overline{\text{ADTRG}}$ pin falling edge after the AN1 pin conversion is started until all selected pins complete A/D conversion. When an $\overline{\text{ADTRG}}$ pin falling edge is generated again during A/D conversion, its trigger is ignored. The falling edge of $\overline{\text{ADTRG}}$ pin, which was input after all selected pins complete A/D conversion, is considered to be the next AN0 pin conversion start condition.
2. The $\overline{\text{ADTRG}}$ pin falling edge is detected synchronized with the operation clock ϕ_{AD}. Therefore, when the $\overline{\text{ADTRG}}$ pin falling edge is generated in shorter periods than ϕ_{AD}, the second $\overline{\text{ADTRG}}$ pin falling edge may not be detected. Do not generate the $\overline{\text{ADTRG}}$ pin falling edge in shorter periods than ϕ_{AD}.
3. Do not write “1” (A/D conversion started) to the ADST bit in delayed trigger mode 1. When write “1”, unexpected interrupts may be generated.
4. AN30 to AN32 can be used in the same way as AN0 to AN7. However, all input pins need to belong to the same group.

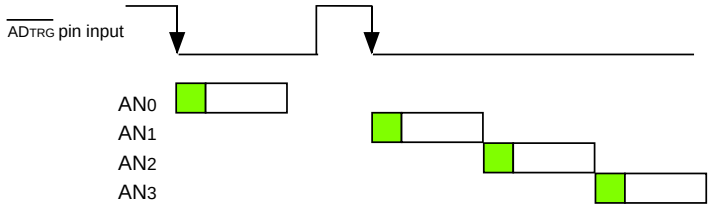
Downloaded from Arrow.com.

•Example when selecting AN0 to AN3 to analog input pins (SCAN1 to SCAN0=012)

•Example 1: When $\overline{\text{ADTRG}}$ pin falling edge is generated during AN0 pin conversion



•Example 2: When $\overline{\text{ADTRG}}$ pin falling edge is generated again after AN0 pin conversion



•Example 3: When $\overline{\text{ADTRG}}$ pin falling edge is generated more than two times after AN0 pin conversion

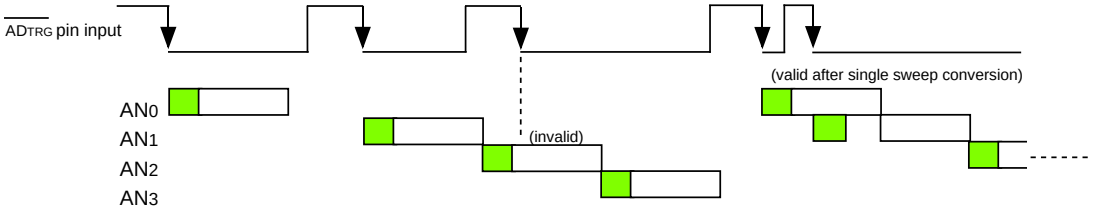
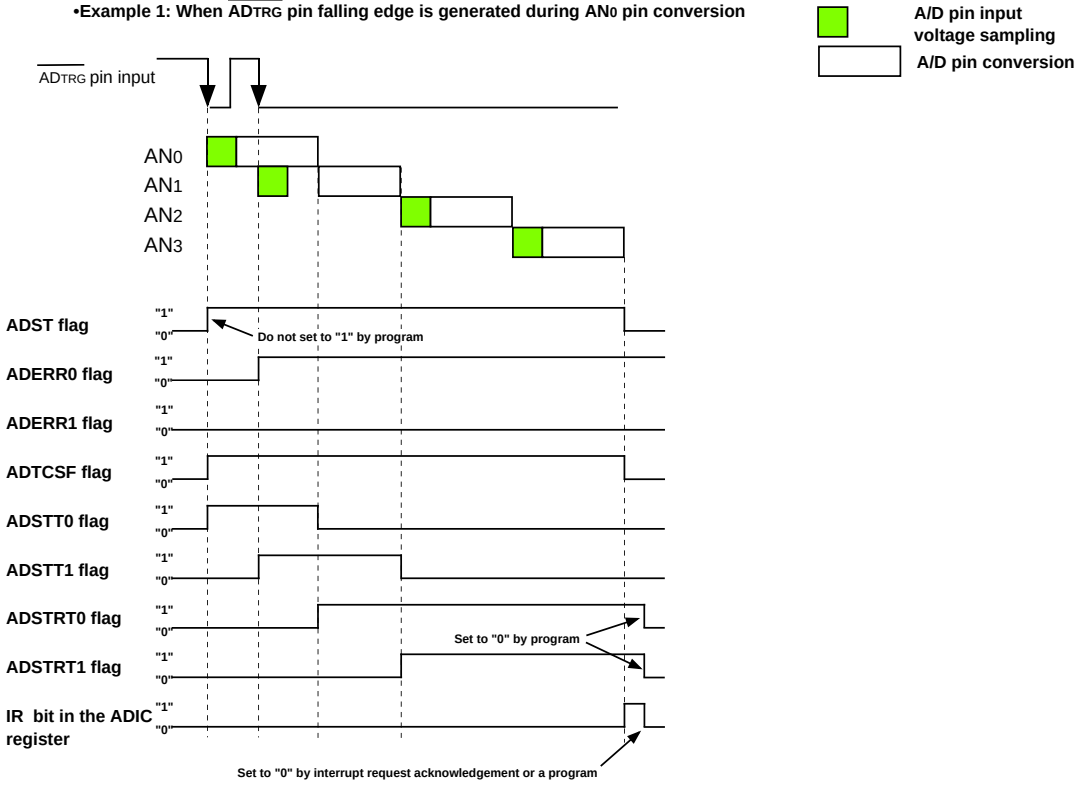


Figure 14.1.8.1 Operation Example in Delayed Trigger Mode1

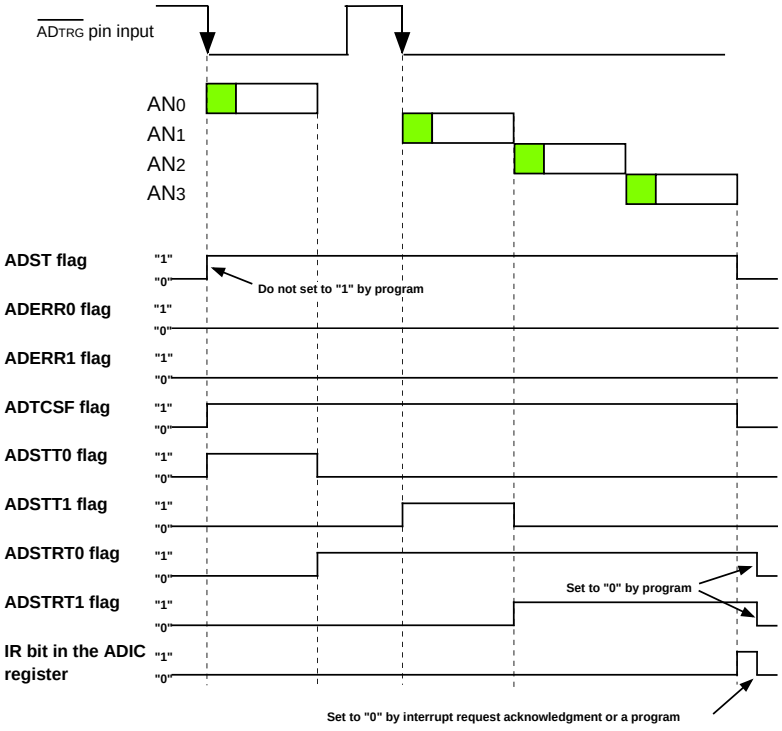
Downloaded from Arrow.com.

•Example when selecting AN0 to AN3 to analog input pins (SCAN1 to SCAN0=012)

•Example 1: When $\overline{\text{ADTRG}}$ pin falling edge is generated during AN0 pin conversion



•Example 2: When $\overline{\text{ADTRG}}$ pin falling edge is generated again after AN0 pin conversion



ADST flag: Bit 6 in the ADCON0 register
ADERR0, ADERR1, ADTCSF, ADSTT0, ADSTT1, ADSTRT0 and ADSTRT1 flag: bits 0, 1, 3, 4, 5, 6 and 7 in the ADSTAT0 register

Figure 14.1.8.2 Each Flag Operation in ADSTAT0 Register Associated with the Operation Example in Delayed Trigger Mode 1 (1)

Downloaded from Arrow.com.

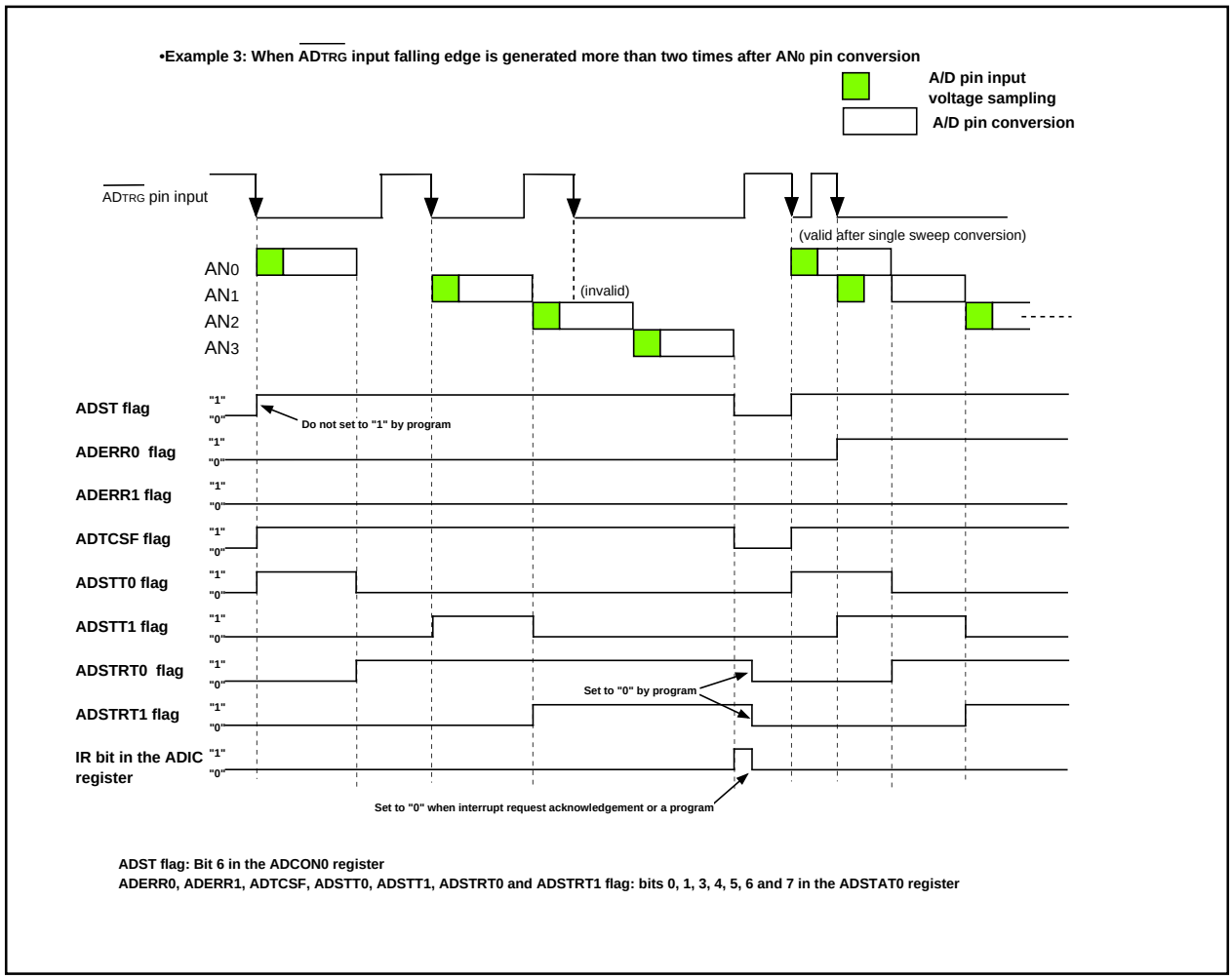


Figure 14.1.8.2 Each Flag Operation in ADSTAT0 Register Associated with the Operation Example in Delayed Trigger Mode 1 (2)

Downloaded from Arrow.com.

A/D control register 0 (1)

b7	b6	b5	b4	b3	b2	b1	b0
		0	0	0	1	1	1

Symbol: ADCON0 Address: 03D6$_{16}$ After reset: 00000XXX$_2$

Bit symbol	Bit name	Function	RW
CH0	Analog Input Pin Select Bit	b2 b1 b0 1 1 1 : Set to "111b" in delayed trigger mode 1	RW
CH1			RW
CH2			RW
MD0	A/D Operation Mode Select Bit 0	b4 b3 0 0 : One-shot mode or delayed trigger mode 0,1	RW
MD1			RW
TRG	Trigger Select Bit	Refer to **Table 14.1.8.2 Trigger Select Bit Setting in Delayed Trigger Mode 1**	RW
ADST	A/D Conversion Start Flag (2)	0 : A/D conversion disabled 1 : A/D conversion started	RW
CKS0	Frequency Select Bit 0	Refer to **Table 14.2 A/D Conversion Frequency Select**	RW

NOTES:
1. If the ADCON0 register is rewritten during A/D conversion, the conversion result will be indeterminate.
2. Do not write “1” in delayed trigger mode 1. When write, set to “0”.

A/D control register 1 (1)

b7	b6	b5	b4	b3	b2	b1	b0
X	X	1			0		

Symbol: ADCON1 Address: 03D7$_{16}$ After reset: 00$_{16}$

Bit symbol	Bit name	Function	RW
SCAN0	A/D Sweep Pin Select Bit (2)	When selecting delayed trigger mode 1 b1 b0 0 0: AN$_0$ to AN$_1$ (2 pins) 0 1: AN$_0$ to AN$_3$ (4 pins) 1 0: AN$_0$ to AN$_5$ (6 pins) 1 1: AN$_0$ to AN$_7$ (8 pins)	RW
SCAN1			RW
MD2	A/D Operation Mode Select Bit 1	0 : Any mode other than repeat sweep mode 1	RW
BITS	8/10-Bit Mode Select Bit	0 : 8-bit mode 1 : 10-bit mode	RW
CKS1	Frequency Select Bit 1	Refer to **Table 14.2 A/D Conversion Frequency Select**	RW
VCUT	V$_{REF}$ Connect Bit (3)	1 : V$_{REF}$ connected	RW
— (b7-b6)	Nothing is assigned. When write, set to “0”. When read, its content is “0”.		—

NOTES:
1. If the ADCON1 register is rewritten during A/D conversion, the conversion result will be indeterminate.
2. AN$_{30}$ to AN$_{32}$ can be used in the same way as AN$_0$ to AN$_7$. Use the ADGSEL1 to ADGSET0 bits in the ADCON2 register to select the desired pin.
3. If the VCUT bit is reset from “0” (V$_{REF}$ unconnected) to “1” (V$_{REF}$ connected), wait for 1 µs or more before startingA/D conversion.

A/D control register 2 (1)

b7	b6	b5	b4	b3	b2	b1	b0
X	X	1		0			1

Symbol: ADCON2 Address: 03D4$_{16}$ After reset: 00$_{16}$

Bit symbol	Bit name	Function	RW
SMP	A/D Conversion Method Select Bit (2)	1 : With sample and hold	RW
ADGSEL0	A/D Input Group Select Bit	b2 b1 0 0 : Select port P10 group (AN$_i$) 0 1 : Select port P9 group (AN$_{3i}$) 1 0 : Do not set 1 1 : Do not set	RW
ADGSEL1			RW
— (b3)	Reserved Bit	Set to “0”	RW
CKS2	Frequency Select Bit 2	Refer to **Table 14.2 A/D Conversion Frequency Select**	RW
TRG1	Trigger Select Bit 1	Refer to **Table 14.1.8.2 Trigger Select Bit Setting in Delayed Trigger Mode 1**	RW
— (b7-b6)	Nothing is assigned. When write, set to “0”. When read, its content is “0”.		—

NOTES:
1. If the ADCON2 register is rewritten during A/D conversion, the conversion result will be indeterminate.
2. Set to “1” in delayed trigger mode 1.

Figure 14.1.8.4 ADCON0 to ADCON2 Registers in Delayed Trigger Mode 1

Downloaded from Arrow.com.

A/D trigger control register (1)

b7	b6	b5	b4	b3	b2	b1	b0
X	X	X	X	0	0	1	1

Symbol	Address	After reset
ADTRGCON	03D2₁₆	00₁₆

Bit symbol	Bit name	Function	RW
SSE	A/D Operation Mode Select Bit 2	Simultaneous sample sweep mode or delayed trigger mode 0,1	RW
DTE	A/D Operation Mode Select Bit 3	Delayed trigger mode 0, 1	RW
HPTRG0	AN0 Trigger Select Bit	Refer to **Table 14.1.8.2 Trigger Select Bit Setting in Delayed Trigger Mode 1**	RW
HPTRG1	AN1 Trigger Select Bit	Refer to **Table 14.1.8.2 Trigger Select Bit Setting in Delayed Trigger Mode 1**	RW
—— (b7-b4)	Nothing is assigned. When write, set to “0”. When read, its content is “0”.		—

NOTE:
1. If ADTRGCON is rewritten during A/D conversion, the conversion result will be indeterminate.

Figure 14.1.8.5 ADTRGCON Register in Delayed Trigger Mode 1

Table 14.1.8.2 Trigger Select Bit Setting in Delayed Trigger Mode 1

TRG	TRG1	HPTRG0	HPTRG1	Trigger
0	1	0	0	$\overline{\text{ADTRG}}$

Downloaded from Arrow.com.

14.2 Resolution Select Function

The BITS bit in the ADCON1 register determines the resolution. When the BITS bit is set to “1” (10-bit precision), the A/D conversion result is stored into bits 0 to 9 in the A/D register i (i=0 to 7). When the BITS bit is set to “0” (8-bit precision), the A/D conversion result is stored into bits 0 to 7 in the ADi register.

14.3 Sample and Hold

When the SMP bit in the ADCON 2 register is set to “1” (with the sample and hold function), A/D conversion rate per pin increases to 28 ϕ_{AD} cycles for 8-bit resolution or 33 ϕ_{AD} cycles for 10-bit resolution. The sample and hold function is available in one-shot mode, repeat mode, single sweep mode, repeat sweep mode 0 and repeat sweep mode 1. In these modes, start A/D conversion after selecting whether the sample and hold circuit is to be used or not. In simultaneous sample sweep mode, delayed trigger mode 0 or delayed trigger mode 1, set to use the Sample and Hold function before starting A/D conversion.

14.4 Power Consumption Reducing Function

When the A/D converter is not used, the VCUT bit in the ADCON1 register isolates the resistor ladder of the A/D converter from the reference voltage input pin (V_{REF}). Power consumption is reduced by shutting off any current flow into the resistor ladder from the V_{REF} pin.

When using the A/D converter, set the VCUT bit to “1” (V_{REF} connected) before setting the ADST bit in the ADCON0 register to “1” (A/D conversion started). Do not set the ADST bit and VCUT bit to “1” simultaneously, nor set the VCUT bit to “0” (V_{REF} unconnected) during A/D conversion.

Downloaded from Arrow.com.

14.5 Output Impedance of Sensor under A/D Conversion

To carry out A/D conversion properly, charging the internal capacitor C shown in Figure 14.5.1 has to be completed within a specified period of time. T (sampling time) as the specified time. Let output impedance of sensor equivalent circuit be R0, microcomputer's internal resistance be R, precision (error) of the A/D converter be X, and the A/D converter's resolution be Y (Y is 1024 in the 10-bit mode, and 256 in the 8-bit mode).

$$\text{VC is generally } VC = VIN\{1-e^{-\frac{1}{c(R0+R)}t}\}$$

$$\text{And when } t = T,\quad VC = VIN - \frac{X}{Y}VIN = VIN\left(1-\frac{X}{Y}\right)$$

$$e^{-\frac{1}{c(R0+R)}T} = \frac{X}{Y}$$

$$-\frac{1}{C(R0+R)}T = \ln\frac{X}{Y}$$

$$\text{Hence,}\quad R0 = -\frac{T}{C\bullet\ln\frac{X}{Y}} - R$$

Figure 14.5.1 shows analog input pin and externalsensor equivalent circuit. When the difference between VIN and VC becomes 0.1LSB, we find impedance R0 when voltage between pins. VC changes from 0 to VIN-(0.1/1024) VIN in timer T. (0.1/1024) means that A/D precision drop due to insufficient capacitor chage is held to 0.1LSB at time of A/D conversion in the 10-bit mode. Actual error however is the value of absolute precision added to 0.1LSB. When f(XIN) = 10MHz, T=0.3µs in the A/D conversion mode with sample & hold. Output inpedance R0 for sufficiently charging capacitor C within time T is determined as follows.

T = 0.3µs, R = 7.8kΩ, C = 1.5pF, X = 0.1, and Y = 1024. Hence,

$$R0 = -\frac{0.3X10^{-6}}{1.5X10^{-12}\bullet\ln\frac{0.1}{1024}} - 7.8 \times 10^{3} \cong 13.9 \times 10^{3}$$

Thus, the allowable output impedance of the sensor circuit capable of thoroughly driving the A/D converter turns out of be approximately 13.9kΩ.

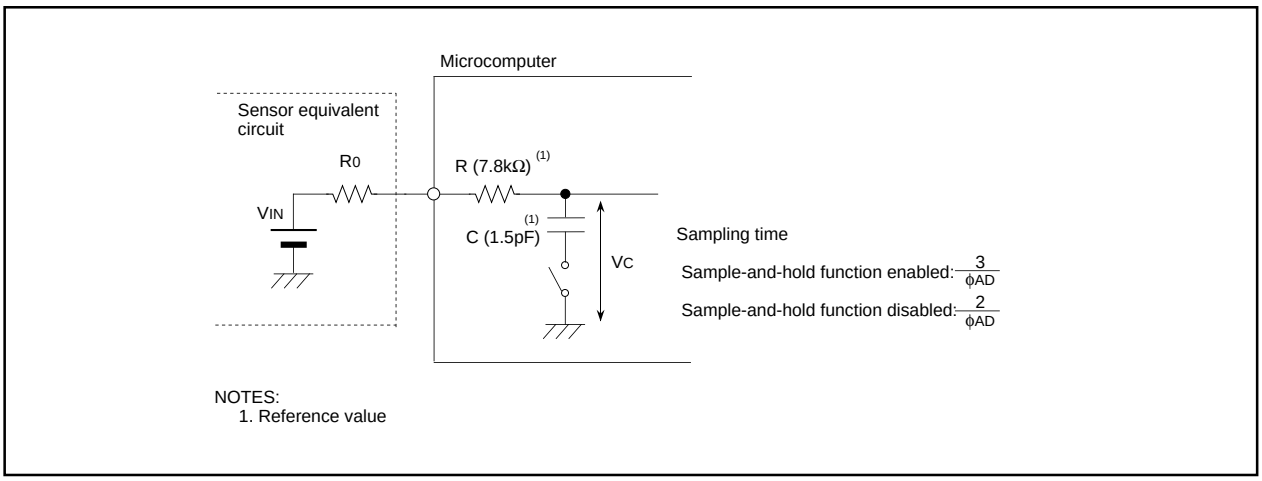


Figure 14.5.1 Analog Input Pin and External Sensor Equivalent Circuit

Downloaded from Arrow.com.

15. CRC Calculation Circuit

The Cyclic Redundancy Check (CRC) operation detects an error in data blocks. The microcomputer uses a generator polynomial of CRC_CCITT ($X^{16} + X^{12} + X^5 + 1$) or CRC-16 ($X^{16} + X^{15} + X^2 + 1$) to generate CRC code.

The CRC code is a 16-bit code generated for a block of a given data length in multiples of bytes. The code is updated in the CRC data register everytime one byte of data is transferred to a CRC input register. The data register needs to be initialized before use. Generation of CRC code for one byte of data is completed in two machine cycles.

Figure 15.1 shows the block diagram of the CRC circuit. Figure 15.2 shows the CRC-related registers. Figure 15.3 shows the calculation example using the CRC_CCITT operation.

15.1. CRC Snoop

The CRC circuit includes the ability to snoop reads and writes to certain SFR addresses. This can be used to accumulate the CRC value on a stream of data without using extra bandwidth to explicitly write data into the CRCIN register. All SFR addresses after 0020_{16} are subject to the CRC snoop. The CRC snoop is useful to snoop the writes to a UART TX buffer, or the reads from a UART RX buffer.

To snoop an SFR address, the target address is written to the CRC snoop Address Register (bits 9 to 0 in the CRCSAR register). The two most significant bits in this register enable snooping on reads or writes to the target address. If the target SFR is written to by the CPU or DMA, and the CRC snoop write bit is set (the CRCSW bit is set to "1"), the CRC will latch the data into the CRCIN register. The new CRC code will be set in the CRCD register.

Similarly, if the target SFR is read by the CRC or DMA, and the CRC snoop read bit is set (the CRCSR bit is set to "1"), the CRC will latch the data from the target into the CRCIN register and calculate the CRC.

The CRC circuit can only calculate CRC codes on data byte at a time. Therefore, if a target SFR is accessed in a word (16 bit) bus cycle, only the byte of data going to or from the target snooped into CRCIN, the other byte of the word access is ignored.

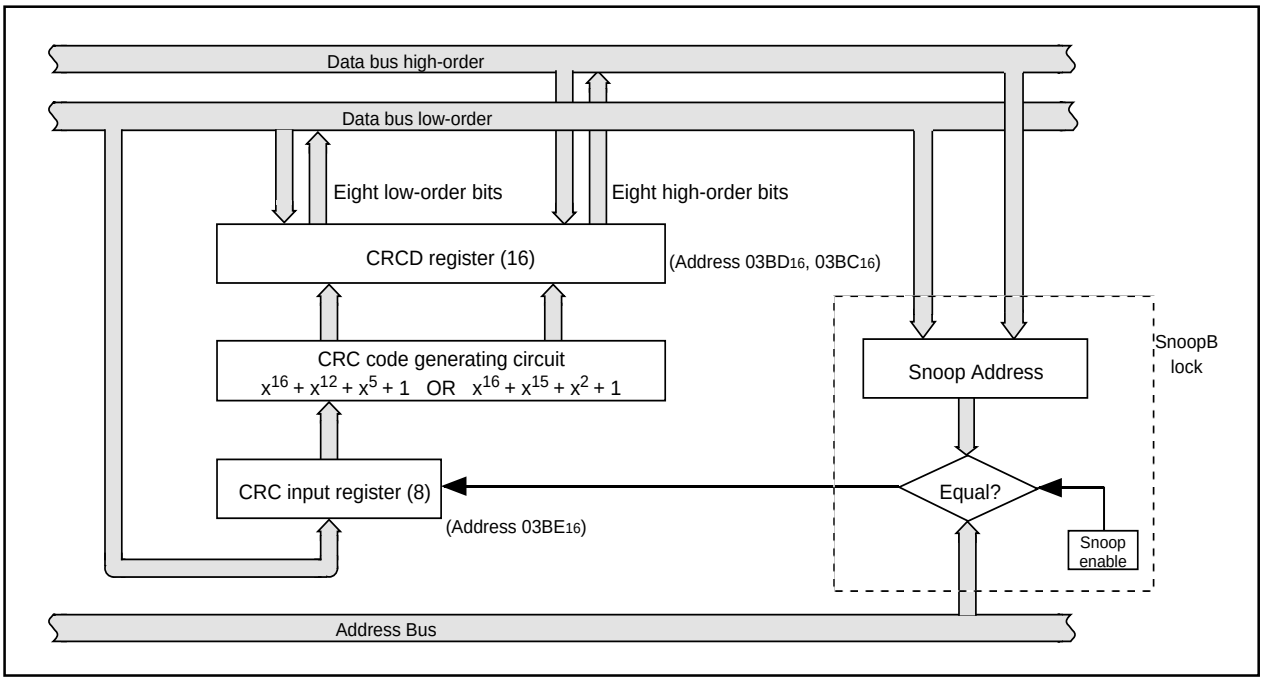
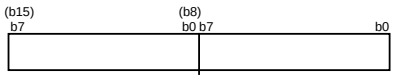


Figure 15.1 CRC circuit block diagram

Downloaded from Arrow.com.

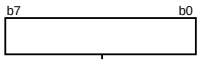
CRC Data Register



Symbol	Address	After Reset
CRCD	03BD16 to 03BC16	Undefined

Function	Setting Range	RW
CRC calculation result output	000016 to FFFF16	RW

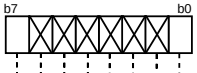
CRC Input Register



Symbol	Address	After Reset
CRCIN	03BE16	Undefined

Function	Setting Range	RW
Data input	0016 to FF16	RW

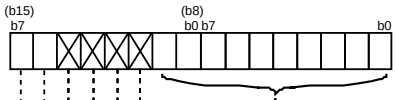
CRC Mode Register



Symbol	Address	After Reset
CRCMR	03B616	0XXXXXX02

Bit Symbol	Bit Name	Function	RW
CRCPS	CRC mode polynomial selection bit	0: $X^{16}+X^{12}+X^5+1$ (CRC-CCITT) 1: $X^{16}+X^{15}+X^2+1$ (CRC-16)	RW
$\overline{\text{(b6-b1)}}$	Nothing is assigned. If necessary, set to 0. When read, the content is undefined		—
CRCMS	CRC mode selection bit	0: LSB first 1: MSB first	RW

SFR Snoop Address Register



Symbol	Address	After Reset
CRCSAR	03B516 to 03B416	00XXXXXX XXXXXXXX2

Bit Symbol	Bit Name	Function	RW
CRCSAR9-0	CRC mode polynomial selection bit	SFR address to snoop	RW
$\overline{\text{(b13-b10)}}$	Nothing is assigned. If necessary, set to 0. When read, the content is undefined		—
CRCSR	CRC snoop on read enable bit	0: Disabled 1: Enabled(1)	RW
CRCSW	CRC snoop on write enable bit	0: Disabled 1: Enabled(1)	RW

NOTE:
1. Set bits CRCSR and CRCSW to 0 if the PLC07 bit in the PLC0 register is set to 1 (PLL on) and the PM20 bit in the PM2 register is set to 0 (SFR access 2 wait).

Figure 15.2. CRCD, CRCIN, CRCMR, CRCSAR Register

Downloaded from Arrow.com.

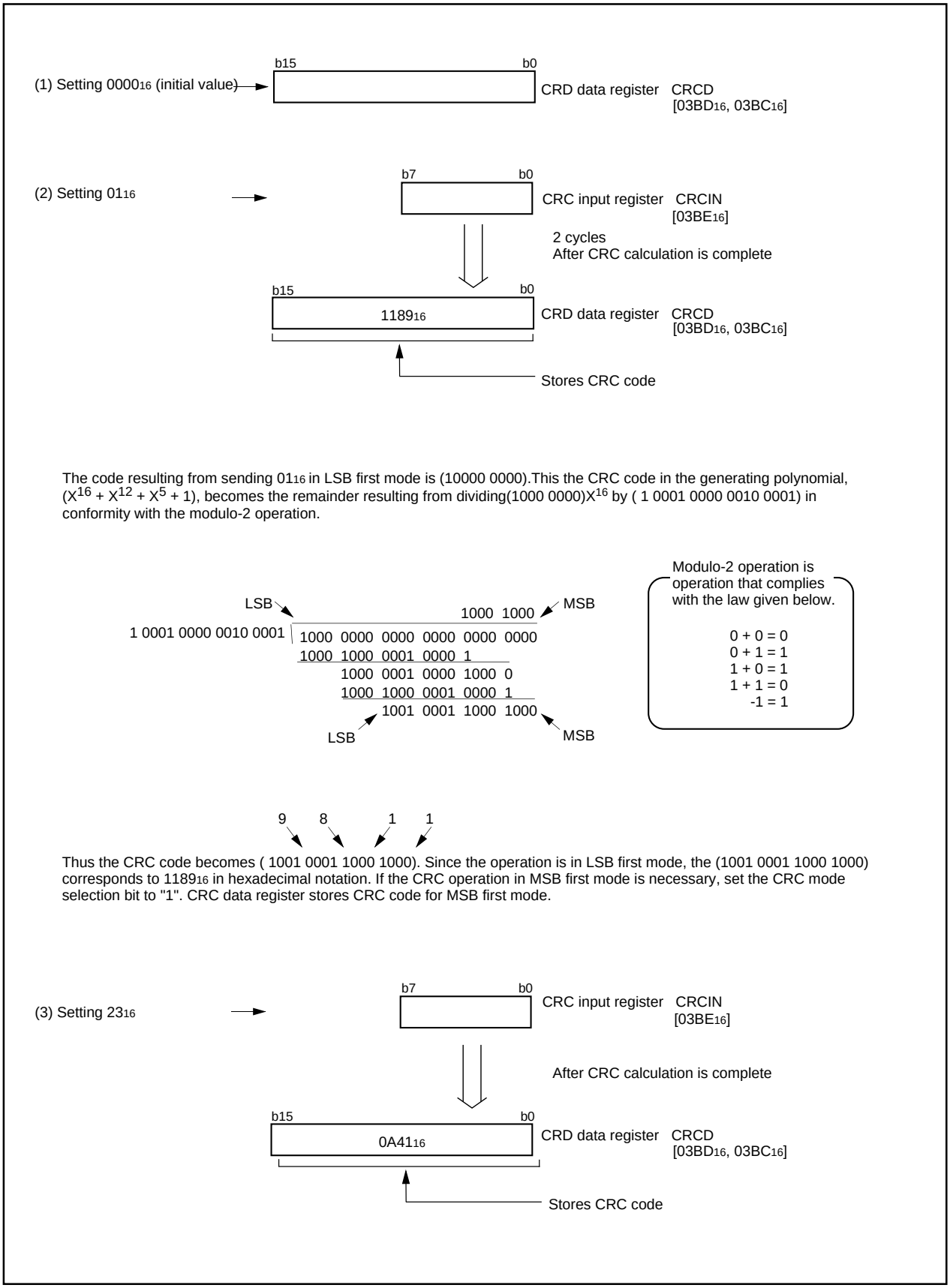


Figure 15.3. CRC Calculation

Downloaded from Arrow.com.

16. Programmable I/O Ports

Note

P6$_0$ to P6$_3$, P9$_2$ and P9$_3$ are not available in the 42-pin package.

The programmable input/output ports (hereafter referred to simply as “I/O ports”) consist of 39 lines P1$_5$ to P1$_7$, P6, P7, P8, P9$_0$ to P9$_3$, P10 for the 48-pin package, or 33 lines P1$_5$ to P1$_7$, P6$_4$ to P6$_7$, P7, P8, P9$_0$ to P9$_1$, P10 for the 42-pin package. Each port can be set for input or output every line by using a direction register, and can also be chosen to be or not be pulled high in sets of 4 lines.

Figures 16.1 to 16.4 show the I/O ports. Figure 16.5 shows the I/O pins.

Each pin functions as an I/O port, a peripheral function input/output.

For details on how to set peripheral functions, refer to each functional description in this manual. If any pin is used as a peripheral function input, set the direction bit for that pin to “0” (input mode). Any pin used as an output pin for peripheral functions is directed for output no matter how the corresponding direction bit is set.

16.1 Port Pi Direction Register (PDi Register, i = 1, 6 to 10)

Figure 16.1.1 shows the direction registers.

This register selects whether the I/O port is to be used for input or output. The bits in this register correspond one for one to each port.

16.2 Port Pi Register (Pi Register, i = 1, 6 to 10)

Figure 16.2.1 shows the Pi registers.

Data input/output to and from external devices are accomplished by reading and writing to the Pi register. The Pi register consists of a port latch to hold the output data and a circuit to read the pin status. For ports set for input mode, the input level of the pin can be read by reading the corresponding Pi register, and data can be written to the port latch by writing to the Pi register.

For ports set for output mode, the port latch can be read by reading the corresponding Pi register, and data can be written to the port latch by writing to the Pi register. The data written to the port latch is output from the pin. The bits in the Pi register correspond one for one to each port.

16.3 Pull-up Control Register 0 to Pull-up Control Register 2 (PUR0 to PUR2 Registers)

Figure 16.3.1 shows the PUR0 to PUR2 registers.

The PUR0 to PUR2 registers select whether the ports, divided into groups of four ports, are pulled up or not. The ports, selected by setting the bits in registers PUR2 to PUR0 to “1” (pull-up), are pulled up when the direction registers are set to “0” (input mode). The ports are pulled up regardless of their function.

Downloaded from Arrow.com.

16.4 Port Control Register

Figure 16.4.1 shows the port control register.

When the P1 register is read after setting the PCR0 bit in the PCR register to “1”, the corresponding port latch can be read no matter how the PD1 register is set.

16.5 Pin Assignment Control register (PACR)

Figure 16.5.1 shows the PACR. After reset set the PACR2 to PACR0 bit before you input and output it to each pin. When the PACR register isn’t set up, the input and output function of some of the pins doesn’t work.

PACR2 to PACR0 bits: control the pins enabled for use.

At reset, these bits are “000”.

In 48-pin package, set these bits to “100_2”.

In 42-pin package, set these bits to “001_2”.

U1MAP: controls the assignment of UART1 pins.

If the U1MAP bit is set to “0” ($P6_7$ to $P6_4$) the UART1 functions are mapped to $P6_4$/CTS_1/RTS_1, $P6_5$/CLK_1, $P6_6$/RxD_1, and $P6_7$/TxD_1.

If the U1MAP bit is set to “1” ($P7_3$ to $P7_0$) the UART1 functions are mapped to $P7_0$/CTS_1/RTS_1, $P7_1$/CLK_1, $P7_2$/RxD_1, and $P7_3$/TxD_1.

PACR is write protected by PRC2 bit in the PRCR register. PRC2 bit must be set immediately before the write to PACR.

16.6 Digital Debounce function

Two digital debounce function circuits are provided. Level is determined when level is held, after applying either a falling edge or rising edge to the pin, longer than the programmed filter width time. This enables noise reduction.

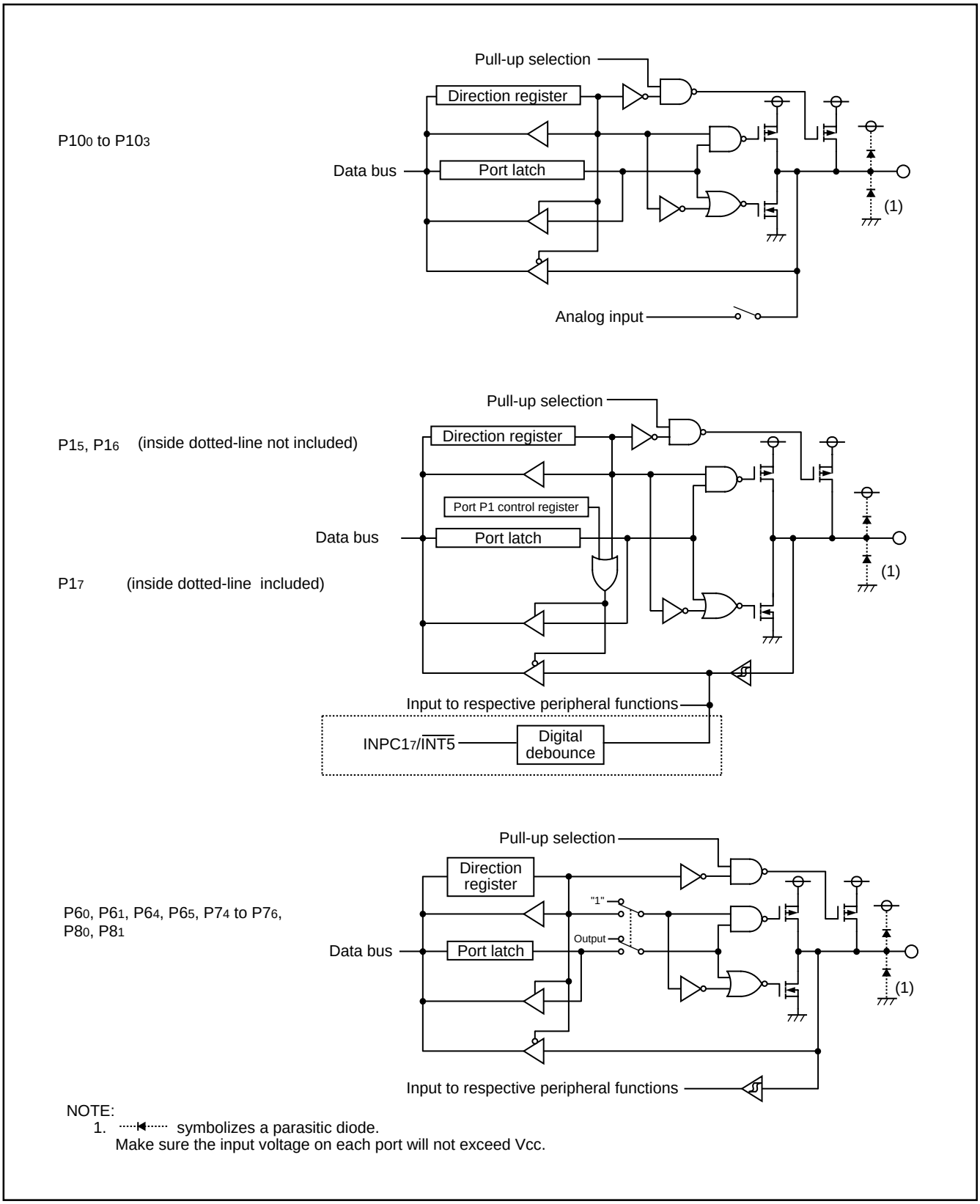
This function is assigned to $\overline{INT5}$/INPC17 and $\overline{NMI}$/$\overline{SD}$. Digital filter width is set in the NDDR register and the P17DDR register respectively. Additionally, a digital debounce function is disabled to the port $P1_7$ input and port $P8_5$ input. Figure 16.6.1 shows the NDDR register and the P17DDR register.

Filter width : $(n+1) \times 1/f8$ n: count value set in the NDDR register and P17DDr register

The NDDR register and the P17DDR register decrement count value with f8 as the count source. The NDDR register and the P17DDR register indicate count time. Count value is reloaded if a falling edge or a rising edge is applied to the pin.

The NDDR register and the P17DDR register can be set 00_{16} to FF_{16} when using the digital debounce function. Setting to FF_{16} disables the digital filter. See Figure 16.6.2 for details.

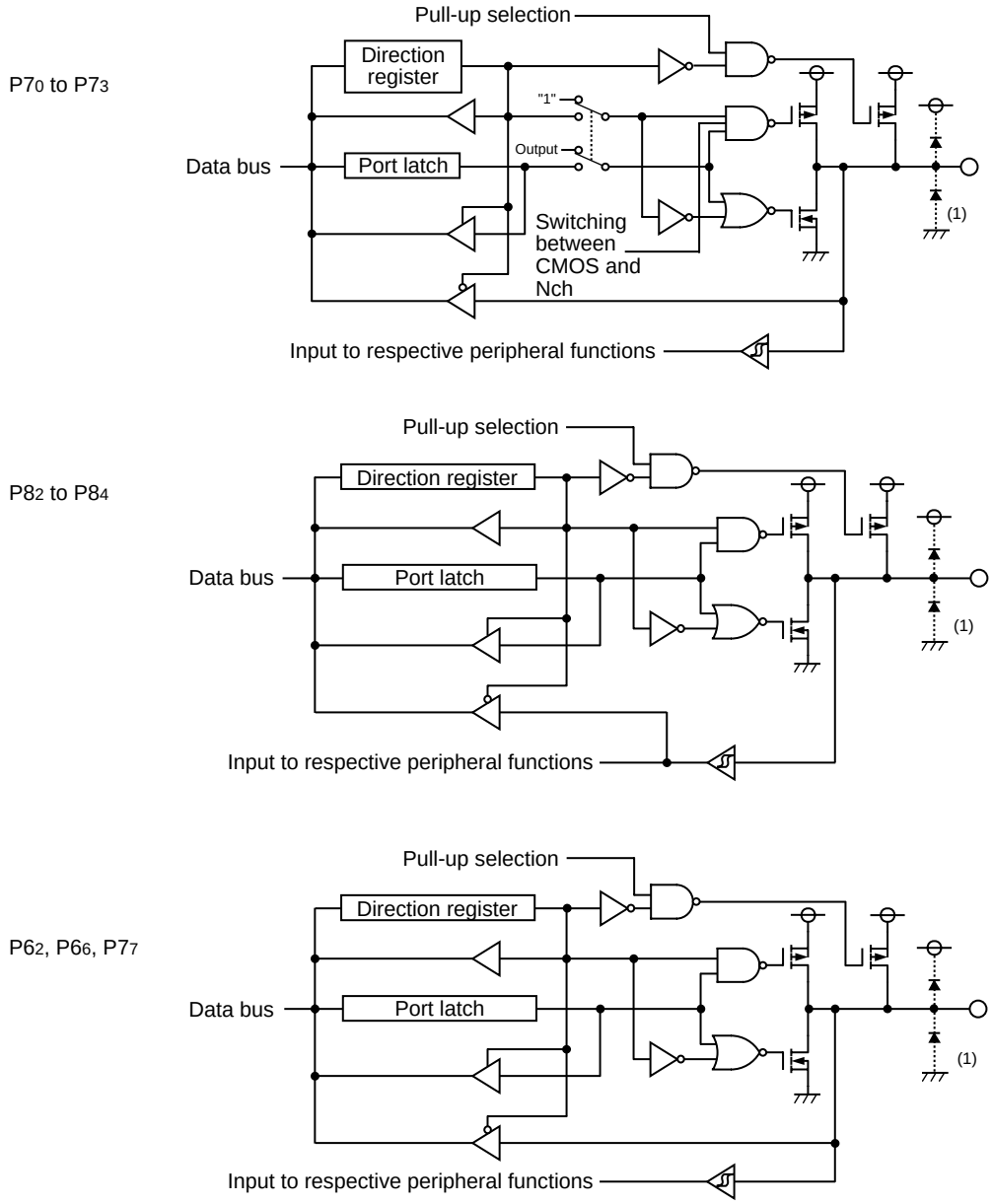
Downloaded from Arrow.com.



NOTE:
1. ⋯◀⋯ symbolizes a parasitic diode.
Make sure the input voltage on each port will not exceed Vcc.

Figure 16.1. I/O Ports (1)

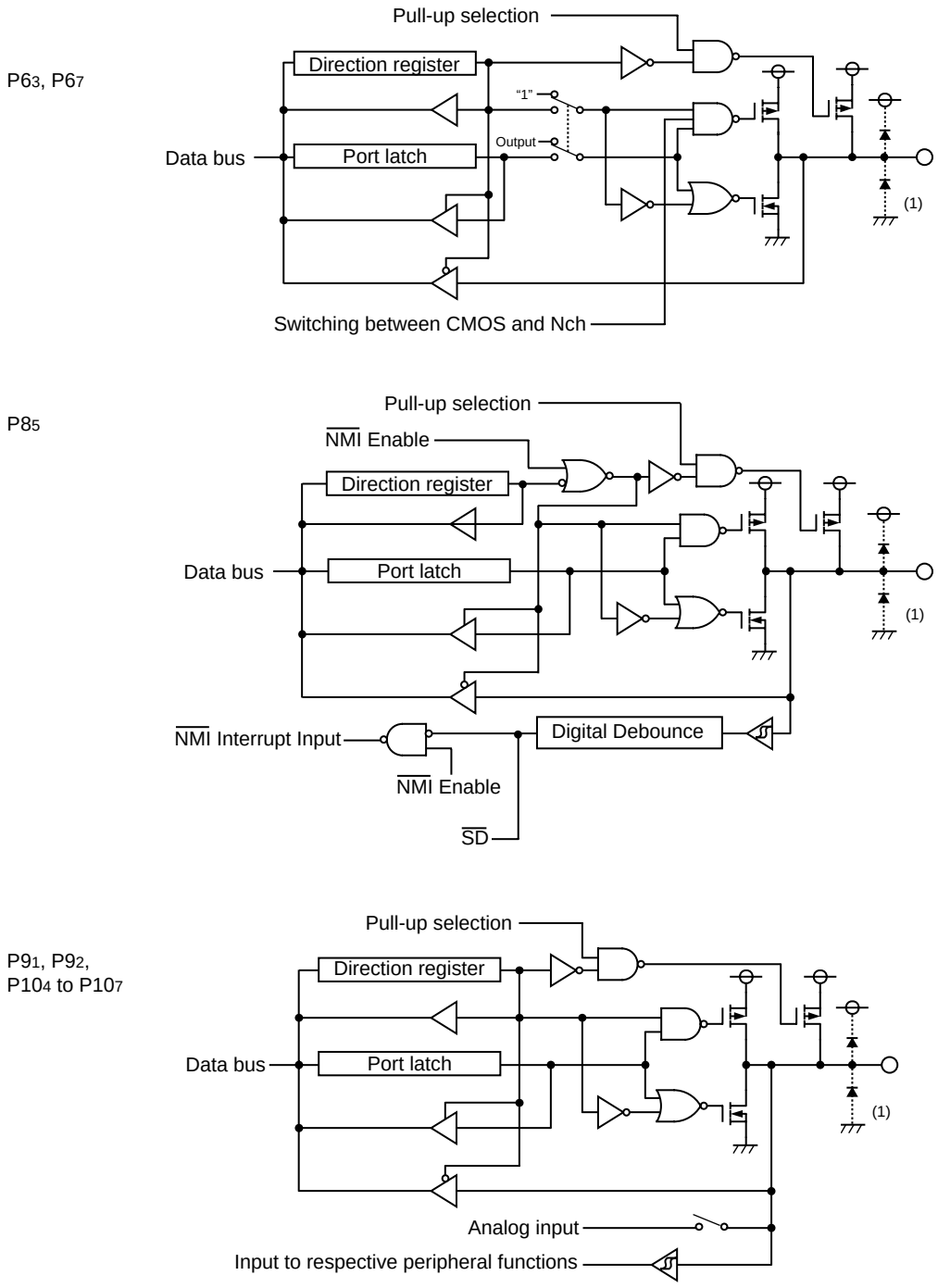
Downloaded from Arrow.com.



NOTE:
1. ·····⏮····· symbolizes a parasitic diode.
Make sure the input voltage on each port will not exceed Vcc.

Figure 16.2. I/O Ports (2)

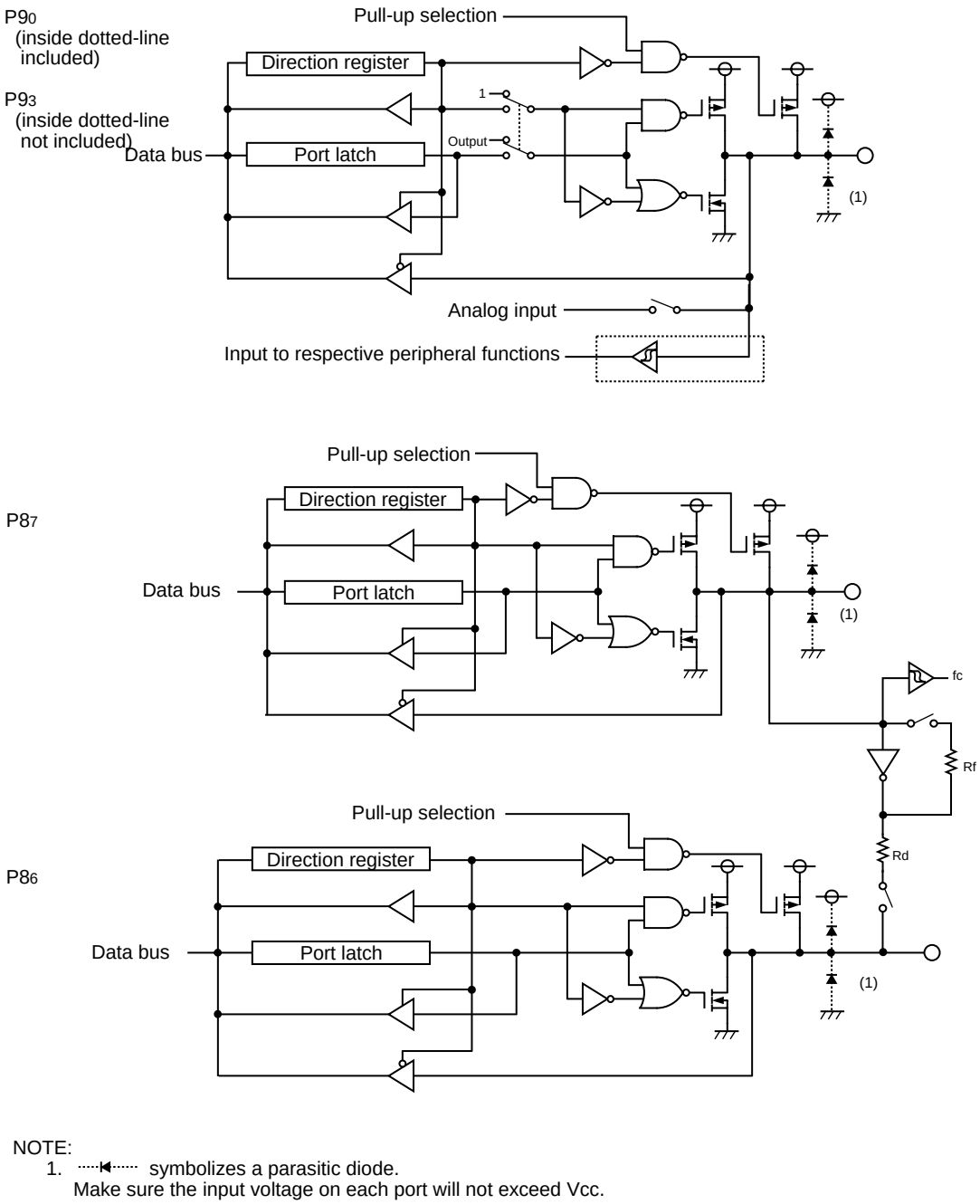
Downloaded from Arrow.com.



NOTE:
1. ⋯◀⋯ symbolizes a parasitic diode.
Make sure the input voltage on each port will not exceed Vcc.

Figure 16.3. I/O Ports (3)

Downloaded from Arrow.com.



NOTE:
1. ⋯◀⋯ symbolizes a parasitic diode.
Make sure the input voltage on each port will not exceed Vcc.

Figure 16.4. I/O Ports (4)

Downloaded from Arrow.com.

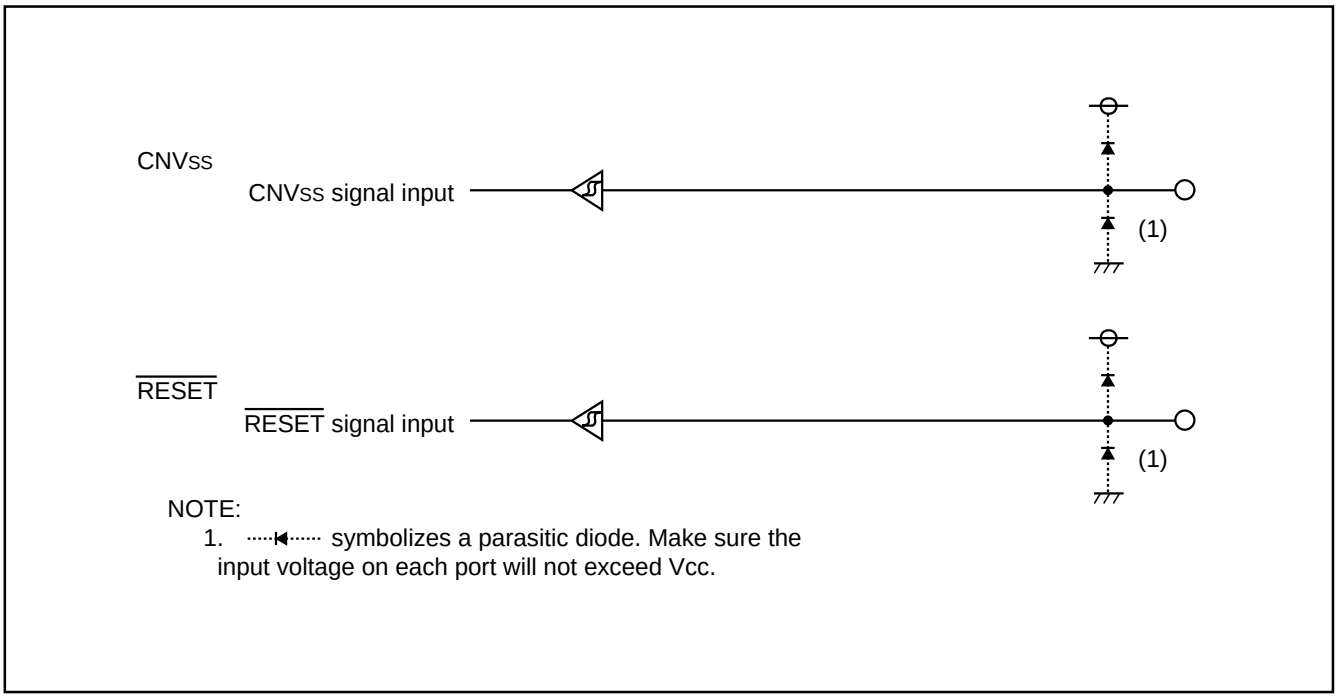
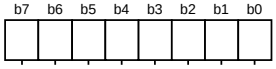


Figure 16.5. I/O Pins

Downloaded from Arrow.com.

Port Pi direction register (i=6 to 8, and 10) (1)

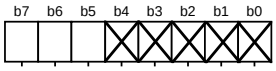


Symbol	Address	After reset
PD6 to PD8	03EE16, 03EF16, 03F216	0016
PD10	03F616	0016

Bit symbol	Bit name	Function	RW
PDi_0	Port Pi0 direction bit	0 : Input mode (Functions as an input port) 1 : Output mode (Functions as an output port) (i = 6 to 8, and 10)	RW
PDi_1	Port Pi1 direction bit		RW
PDi_2	Port Pi2 direction bit		RW
PDi_3	Port Pi3 direction bit		RW
PDi_4	Port Pi4 direction bit		RW
PDi_5	Port Pi5 direction bit		RW
PDi_6	Port Pi6 direction bit		RW
PDi_7	Port Pi7 direction bit		RW

NOTE:
1. Ports must be enabled using the PACR register.
   In 48-pin package, set PACR2, PACR1, PACR0 to "1002"
   In 42-pin package, set PACR2, PACR1, PACR0 to "0012"

Port P1 direction register (1)

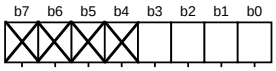


Symbol	Address	After reset
PD1	03E316	0016

Bit symbol	Bit name	Function	RW
(b4-b0)	Nothing is assigned. In an attempt to write to this bit, write “0”. The value, if read, turns out to be indeterminate.		—
PD1_5	Port P15 direction bit	0 : Input mode (Functions as an input port) 1 : Output mode (Functions as an output port)	RW
PD1_6	Port P16 direction bit		RW
PD1_7	Port P17 direction bit		RW

NOTE:
1. Ports must be enabled using the PACR register.
   In 48-pin package, set PACR2, PACR1, PACR0 to "1002"
   In 42-pin package, set PACR2, PACR1, PACR0 to "0012"

Port P9 direction register (1,2)



Symbol	Address	After reset
PD9	03F316	XXXX00002

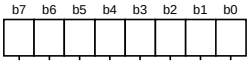
Bit symbol	Bit name	Function	RW
PD9_0	Port P90 direction bit	0 : Input mode (Functions as an input port) 1 : Output mode (Functions as an output port)	RW
PD9_1	Port P91 direction bit		RW
PD9_2	Port P92 direction bit		RW
PD9_3	Port P93 direction bit		RW
(B7-b4)	Nothing is assigned. In an attempt to write to this bit, write “0”. The value, if read, turns out to be indeterminate.		—

NOTE:
1. Make sure the PD9 register is written to by the next instruction after setting the PRC2 bit in the PRCR register to "1" (write enabled).
2. Ports must be enabled using the PACR register.

Figure 16.1.1. PD1, PD6, PD7, PD8, PD9, and PD10 Registers

Downloaded from Arrow.com.

Port Pi register (i=6 to 8 and 10) (1)

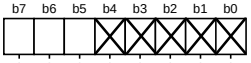


Symbol	Address	After reset
P6 to P8	03EC16, 03ED16, 03F016	Indeterminate
P10	03F416	Indeterminate

Bit symbol	Bit name	Function	RW
Pi_0	Port Pi0 bit	The pin level on any I/O port which is set for input mode can be read by reading the corresponding bit in this register. The pin level on any I/O port which is set for output mode can be controlled by writing to the corresponding bit in this register 0 : "L" level 1 : "H" level (1) (i = 6 to 8 and 10)	RW
Pi_1	Port Pi1 bit		RW
Pi_2	Port Pi2 bit		RW
Pi_3	Port Pi3 bit		RW
Pi_4	Port Pi4 bit		RW
Pi_5	Port Pi5 bit		RW
Pi_6	Port Pi6 bit		RW
Pi_7	Port Pi7 bit		RW

NOTE:
1. Ports must be enabled using the PACR register.
   In 48-pin package, set PACR2, PACR1, PACR0 to "100$_2$"
   In 42-pin package, set PACR2, PACR1, PACR0 to "001$_2$"

Port P1 register (1)

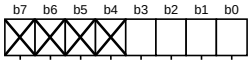


Symbol	Address	After reset
P1	03E116	Indeterminate

Bit symbol	Bit name	Function	RW
$\overline{\text{(b4-b0)}}$	Nothing is assigned. In an attempt to write to this bit, write "0". The value, if read, turns out to be indeterminate.		—
P1_5	Port P15 bit	The pin level on any I/O port which is set for input mode can be read by reading the corresponding bit in this register. The pin level on any I/O port which is set for output mode can be controlled by writing to the corresponding bit in this register 0 : "L" level 1 : "H" level	RW
P1_6	Port P16 bit		RW
P1_7	Port P17 bit		RW

NOTE:
1. Ports must be enabled using the PACR register.
   In 48-pin package, set PACR2, PACR1, PACR0 to "100$_2$"
   In 42-pin package, set PACR2, PACR1, PACR0 to "001$_2$"

Port P9 register (1)



Symbol	Address	After reset
P9	03F116	Indeterminate

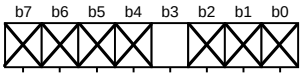
Bit symbol	Bit name	Function	RW
P9_0	Port P90 bit	The pin level on any I/O port which is set for input mode can be read by reading the corresponding bit in this register. The pin level on any I/O port which is set for output mode can be controlled by writing to the corresponding bit in this register 0 : "L" level 1 : "H" level	RW
P9_1	Port P91 bit		RW
P9_2	Port P92 bit		RW
P9_3	Port P93 bit		RW
$\overline{\text{(b7-b4)}}$	Nothing is assigned. In an attempt to write to this bit, write "0". The value, if read, turns out to be indeterminate.		—

NOTE:
1. Ports must be enabled using the PACR register.
   In 48-pin package, set PACR2, PACR1, PACR0 to "100$_2$"
   In 42-pin package, set PACR2, PACR1, PACR0 to "001$_2$"

Figure 16.2.1. P1, P6, P7, P8, P9, and P10 Registers

Downloaded from Arrow.com.

Pull-up control register 0

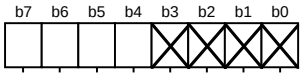


Symbol	Address	After reset
PUR0	03FC16	0016

Bit symbol	Bit name	Function	RW
—— (b2-b0)	Nothing is assigned. In an attempt to write to these bits, write "0". The value, if read, turns out to be "0".		—
PU03	P15 to P17 pull-up	0 : Not pulled high 1 : Pulled high (1)	RW
—— (b7-b4)	Nothing is assigned. In an attempt to write to these bits, write "0". The value, if read, turns out to be "0".		—

NOTE:
1. The pin for which this bit is "1" (pulled high) and the direction bit is "0" (input mode) is pulled high.

Pull-up control register 1

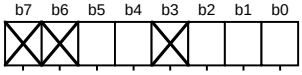


Symbol	Address	After reset
PUR1	03FD16	000000002

Bit symbol	Bit name	Function	RW
—— (b3-b0)	Nothing is assigned. In an attempt to write to these bits, write "0". The value, if read, turns out to be "0".		—
PU14	P60 to P63 pull-up	0 : Not pulled high 1 : Pulled high (1)	RW
PU15	P64 to P67 pull-up		RW
PU16	P70 to P73 pull-up		RW
PU17	P74 to P77 pull-up		RW

NOTE:
1. The pin for which this bit is "1" (pulled high) and the direction bit is "0" (input mode) is pulled high.

Pull-up control register 2



Symbol	Address	After reset
PUR2	03FE16	0016

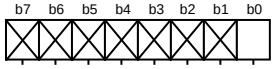
Bit symbol	Bit name	Function	RW
PU20	P80 to P83 pull-up	0 : Not pulled high 1 : Pulled high (1)	RW
PU21	P84 to P87 pull-up		RW
PU22	P90 to P93 pull-up		RW
—— (b3)	Nothing is assigned. In an attempt to write to these bits, write "0". The value, if read, turns out to be "0".		—
PU24	P100 to P103 pull-up	0 : Not pulled high 1 : Pulled high (1)	RW
PU25	P104 to P107 pull-up		RW
—— (b7-b6)	Nothing is assigned. In an attempt to write to these bits, write "0". The value, if read, turns out to be "0".		—

NOTE:
1. The pin for which this bit is "1" (pulled high) and the direction bit is "0" (input mode) is pulled high.

Figure 16.3.1. PUR0 to PUR2 Registers

Downloaded from Arrow.com.

Port control register

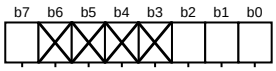


Symbpl	Address	After reset
PCR	$03FF_{16}$	00_{16}

Bit symbol	Bit name	Function	RW
PCR0	Port P1 control bit	Operation performed when the P1 register is read 0: When the port is set for input, the input levels of P10 to P17 pins are read. When set for output, the port latch is read. 1: The port latch is read regardless of whether the port is set for input or output.	RW
— (b7-b1)	Nothing is assigned. In an attempt to write to these bits, write "0". The value, if read, turns out to be "0".		—

Figure 16.4.1. PCR Register

Pin Assignment Control Register (1)



Symbpl	Address	After Reset
PACR	$025D_{16}$	00_{16}

Bit Symbol	Bit Name	Function	RW
PACR0	Pin enabling bit	001 : 42 pin 100 : 48 pin All other values are reserved. Do not use.	RW
PACR1			RW
PACR2			RW
— (b6-b3)	Reserved bits	Nothing is assigned. When write, set to "0". When read, its content is "0".	—
U1MAP	UART1 pin remapping bit	UART1 pins assigned to 0 : $P6_7$ to $P6_4$ 1 : $P7_3$ to $P7_0$	RW

NOTES:
1. Set the PACR register by the next instruction after setting the PRC2 bit in the PRCR register to "1"(write enable).

Figure 16.5.1. PACR Register

Downloaded from Arrow.com.

$\overline{NMI}$ Digital Debounce Register (1,2)

b7	b0	Symbol	Address	After Reset
		NDDR	033E16	FF16

Function	Setting Range	RW
If the set value =n, - n = 0 to FE16; a signal with pulse width, greater than (n+1)/f8, is input into $\overline{NMI}$ / $\overline{SD}$ - n = FF16; the digital debounce filter is disabled and all signals are input	0016 to FF16	RW

NOTES:
1. Set the PACR register by the next instruction after setting the PRC2 bit in the PRCR register to "1"(write enable).
2. When using the $\overline{NMI}$ interrupt to exit from stop mode, set the NDDR registert to "FF16" before entering stop mode.

P17 Digital Debounce Register(1)

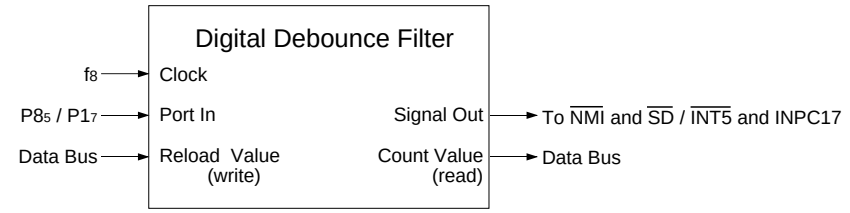
b7	b0	Symbol	Address	After Reset
		P17DDR	033F16	FF16

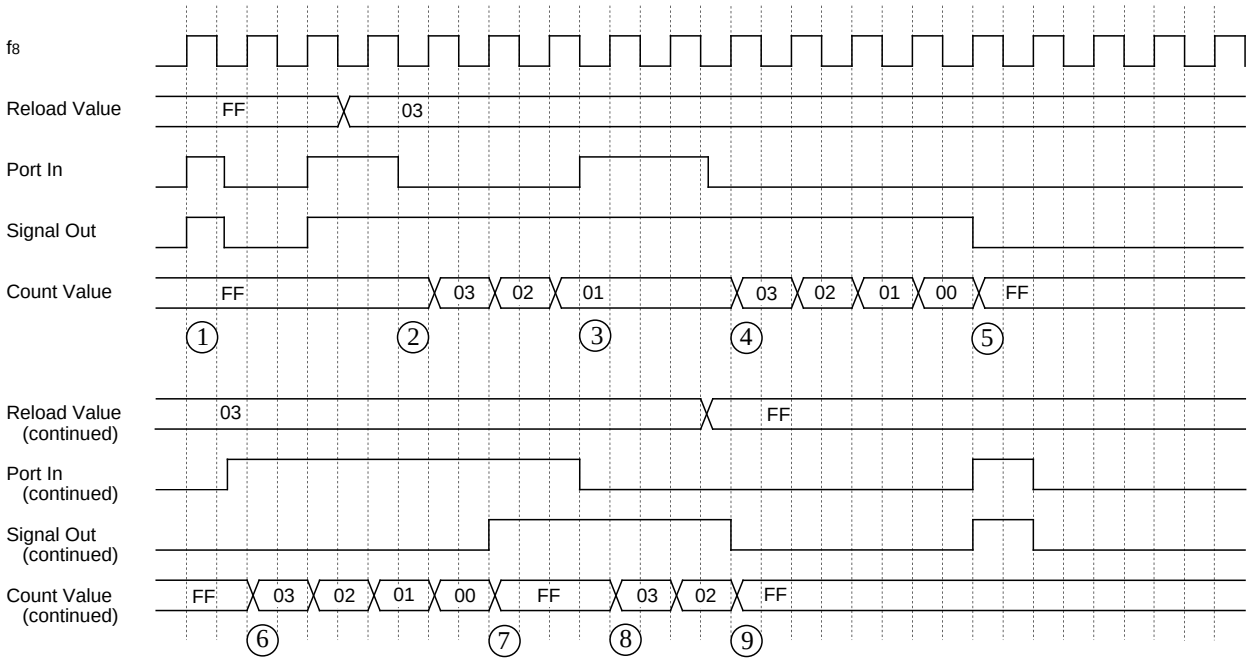
Function	Setting Range	RW
If the set value =n, - n = 0 to FE16; a signal with pulse width, greater than (n+1)/f8, is input into INPC17/ $\overline{INT5}$ - n = FF16; the digital debounce filter is disabled and all signals are input	0016 to FF16	RW

NOTE:
1. When using the $\overline{INT5}$ interrupt to exit from stop mode, set the P17DDR registert to "FF16" before entering stop mode.

Figure 16.6.1. NDDR and P17DDR Registers

Downloaded from Arrow.com.





1. (Condition after reset). Reload = FF, Port In = signal Out continuosly.
2. Reload = 03. At edge of Port In != Signal Out, Counter gets Reload Value and stats counting down.
3. Port In = Signal Out, counting stops.
4. At edge of Port In != Signal Out, Counter gets Reload Value and starts counting.
5. Counter underflows, stops, and Port In is driven to Signal Out.
6. At edge of Port In != Signal Out, counter gets Reload Value and starts counting.
7. Counter underflows, stops, and Port In is driven to Signal Out.
8. At edge of Port In != Signal Out, counter gets Reload Value and starts counting.
9. FF is written to Reload Value. Counter is stopped and loaded with FF.
   Port In = Signal Out continuously.

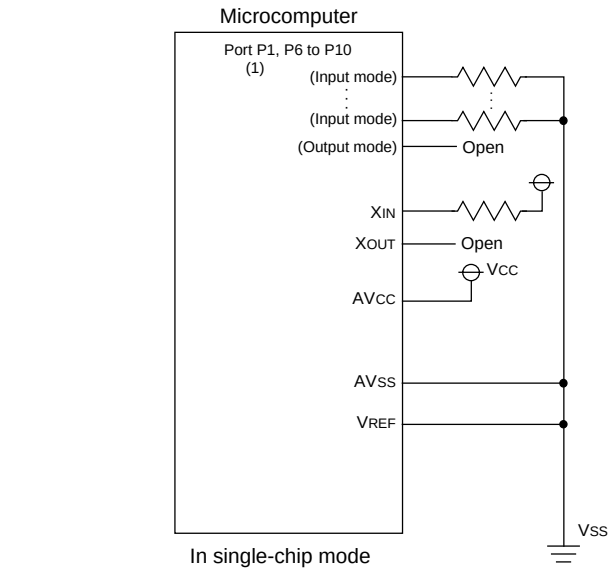
Figure 16.6.2. Functioning of Digital Debounce Filter

Downloaded from Arrow.com.

Table 16.1. Unassigned Pin Handling in Single-chip Mode

Pin name	Connection
Ports P1, P6 to P10	After setting for input mode, connect every pin to V_{SS} via a resistor(pull-down); or after setting for output mode, leave these pins open. (1, 2, 4)
X_{OUT} (3)	Open
X_{IN}	Connect via resistor to V_{CC} (pull-up) (5)
AV_{CC}	Connect to V_{CC}
AV_{SS}, V_{REF}	Connect to V_{SS}

NOTES:
1. When setting the port for output mode and leave it open, be aware that the port remains in input mode until it is switched to output mode in a program after reset. For this reason, the voltage level on the pin becomes indeterminate, causing the power supply current to increase while the port remains in input mode. Futhermore, by considering a possibility that the contents of the direction registers could be changed by noise or noise-induced runaway, it is recommended that the contents of the direction registers be periodically reset in software, for the increased reliability of the program.
2. Make sure the unused pins are processed with the shortest possible wiring from the microcomputer pins (within 2 cm).
3. With external clock or VCC input to XIN pin.
4. When using the 48-pin package, set PACR2, PACR1, PACR0 to "100_2".When using the 42-pin package, set PACR2, PACR1, PACR0 to "001_2".
5. When the main clock oscillation circuit is not used, set the CM05 bit in the CM0 register to “0” (main clock stops) to reduce power consumption.



NOTE:
1. When using the 48-pin package, set PACR2, PACR1, PACR0 to "100_2".
When using the 42pin-package, set PACR2, PACR1, PACR0 to "001_2".

Figure 16.7. Unassigned Pins Handling

Downloaded from Arrow.com.

17. Flash Memory Version

17.1 Flash Memory Performance

The flash memory version is functionally the same as the mask ROM version except that it internally contains flash memory.

In the flash memory version, the flash memory can perform in three rewrite mode : CPU rewrite mode, standard serial I/O mode and parallel I/O mode.

Table 17.1 shows the flash memory version specifications. (Refer to Table 1.1 or Table 1.2 for the items not listed in Table 17.1.)

Table 17.1. Flash Memory Version Specifications

Item		Specification
Flash memory operating mode		3 modes (CPU rewrite, standard serial I/O, parallel I/O)
Erase block		See **Figure 17.2.1** to **17.2.3** Flash Memory Block Diagram
Program method		In units of word
Erase method		Block erase
Program, erase control method		Program and erase controlled by software command
Protect method		All user blocks are write protected by bit FMR16. In addition, the block 0 and block 1 are write protected by bit FMR02
Number of commands		5 commands
Program/Erase Endurance(1)	Block 0 to 3 (program area)	100 times, 1,000 times (See **Tables 1.7, 1.9, and 1.10**)
	Block A and B (data are) (2)	100 times, 10,000 times (See **Tables 1.7, 1.9, and 1.10**)
Data Retention		20 years (Topr = 55°C)
ROM code protection		Parallel I/O and standard serial I/O modes are supported.

NOTES:

1. Program and erase endurance definitionProgram and erase endurance are the erase endurance of each block. If the program and erase endurance are n times (n=100,1,000,10,000), each block can be erased n times. For example, if a 2-Kbyte block A is erased after writing 1 word data 1024 times, each to different addresses, this is counted as one program and erasure.However, data cannot be written to the same address more than once without erasing the block. (Rewrite disabled)
2. To use the limited number of erasure efficiently, write to unused address within the block instead of rewrite. Erase block only after all possible address are used. For example, an 8-word program can be written 128 times before erase is necessary. Maintaining an equal number of erasure between Block A and B will also improve efficiency. We recommend keeping track of the number of times erasure is used.

Downloaded from Arrow.com.

Table 17.2. Flash Memory Rewrite Modes Overview

Flash memory rewrite mode	CPU rewrite mode	Standard serial I/O mode	Parallel I/O mode
Function	The user ROM area is rewritten when the CPU executes software command EW0 mode: Rewrite in area other than flash memory EW1 mode: Rewrite in flash memory	The user ROM area is rewritten using a dedicated serial programmer. Standard serial I/O mode 1: Clock synchronous serial I/O Standard serial I/O mode 2: UART	The user ROM area is rewritten using a dedicated parallel programmer
Area which can be rewritten	User ROM area	User ROM area	User ROM area
Operation mode	Single chip mode	Boot mode	Parallel I/O mode
ROM programmer	None	Serial programmer	Parallel programmer

17.1.1 Boot Mode

The MCU enters boot mode when a hardware reset is performed while a high-level ("H") signal is applied to pins CNV_{SS} and $P8_6$ or while an "H" signal is applied to pins CNV_{SS} and $P1_6$ and a low-level ("L") signal is applied to the $P8_5$. A program in the boot ROM area is executed.

The boot ROM area is reserved. The boot ROM area stores the rewrite control program for a standard serial I/O mode before shipping. Do not rewrite the boot ROM area.

Downloaded from Arrow.com.

17.2 Memory Map

The flash memory contains the user ROM area and the boot ROM area (reserved area). Figures 17.2.1 to 17.2.3 show the flash memory block diagram. The user ROM area has space to store the microcomputer operation program in single-chip mode and a separate 2-Kbyte space as the block A and B.

The user ROM area is divided into several blocks. The user ROM area can be rewritten in CPU rewrite, standard serial input/output, and parallel input/output modes. However, if block 0 and 1 are rewritten in CPU rewrite mode, setting the FMR02 bit in the FMR0 register to “1” (block 0, 1 rewrite enabled) and the FMR16 bit in the FMR1 register to “1”(blocks 0 to 3 rewrite enabled) enable rewriting. Also, if blocks 2 to 3 are rewritten in CPU rewrite mode, setting the FMR16 bit in the FMR1 register to “1” (blocks 0 to 3 rewrite enabled) enables writing. Setting the PM10 bit in the PM1 register to “1”(data area access enabled) for block A and B enables to use.

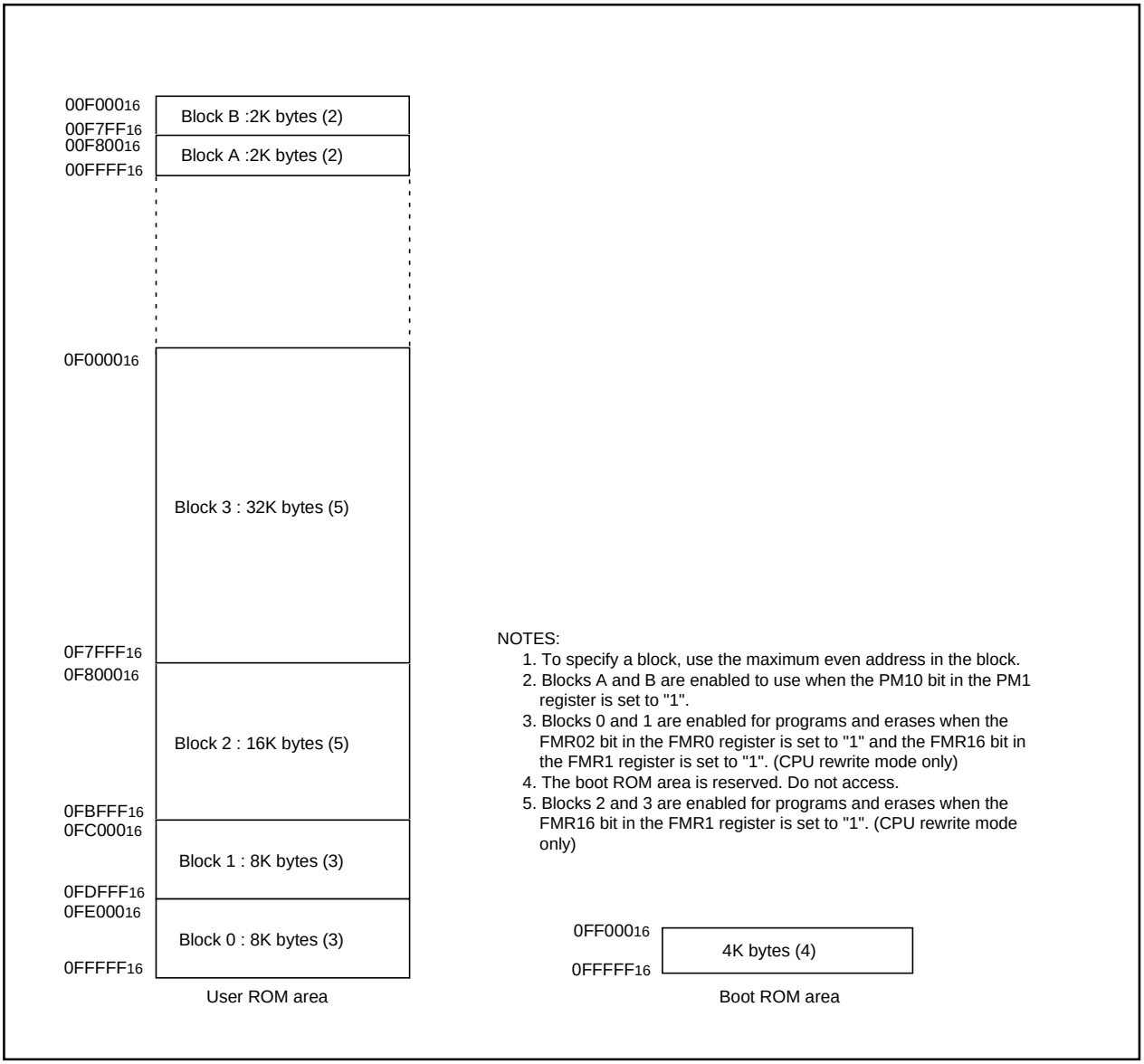


Figure 17.2.1. Flash Memory Block Diagram (ROM capacity 64K byte)

Downloaded from Arrow.com.

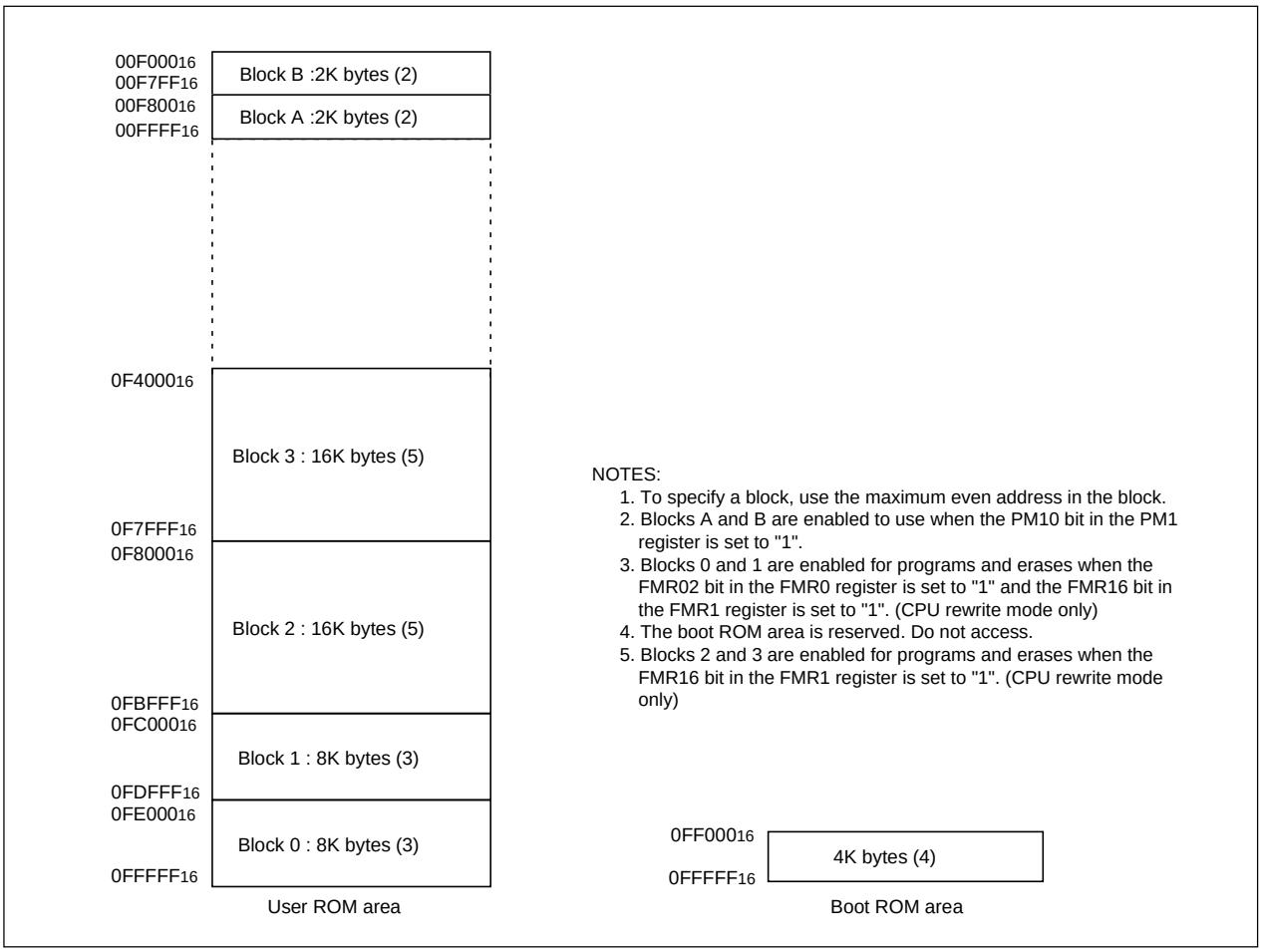


Figure 17.2.2. Flash Memory Block Diagram (ROM capacity 48K byte)

Downloaded from Arrow.com.

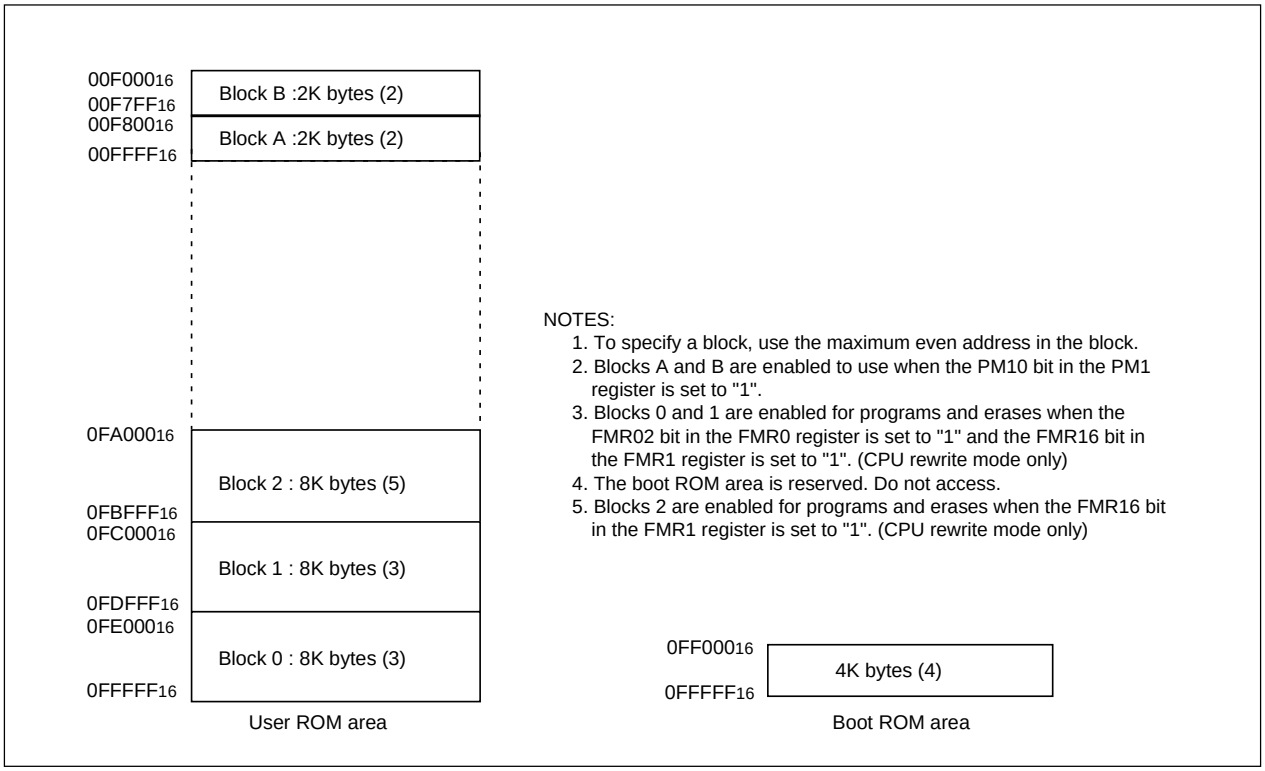


Figure 17.2.3. Flash Memory Block Diagram (ROM capacity 24K byte)

Downloaded from Arrow.com.

17.3 Functions To Prevent Flash Memory from Rewriting

The flash memory has a built-in ROM code protect function for parallel I/O mode and a built-in ID code check function for standard input/output mode to prevent the flash memory from reading or rewriting.

17.3.1 ROM Code Protect Function

The ROM code protect function disables reading or changing the contents of the on-chip flash memory in parallel I/O mode. **Figure 17.3.1.1** shows the ROMCP address. The ROMCP address is located in a user ROM area.

To enable ROM code protect, set the ROMCP1 bit to “00_2”, “01_2”, or “10_2” and set the bit 5 to bit 0 to “111111_2”.

To cancel ROM code protect, erase the block including the the ROMCP address in CPU rewrite mode or standard serial I/O mode.

17.3.2 ID Code Check Function

Use the ID code check function in standard serial input/output mode. Unless the flash memory is blank, the ID codes sent from the programmer and the seven bytes ID codes written in the flash memory are compared to see if they match. If the ID codes do not match, the commands sent from the programmer are not acknowledged. The ID code consists of 8-bit data, starting with the first byte, into addresses, $0FFFDF_{16}$, $0FFFE3_{16}$, $0FFFEB_{16}$, $0FFFEF_{16}$, $0FFFF3_{16}$, $0FFFF7_{16}$, and $0FFFFB_{16}$. The flash memory has a program with the ID code set in these addresses.

Downloaded from Arrow.com.

ROM Code Protect Control Address(5)

b7	b6	b5	b4	b3	b2	b1	b0
		1	1	1	1	1	1

Symbol: ROMCP
Address: 0FFFFF16
Factory Setting: FF16 (4)

Bit Symbol	Bit Name	Function	RW
$\overline{\text{(b5-b0)}}$	Reserved Bit	Set to 1	RW
ROMCP1	ROM Code Protect Level 1 Set Bit (1, 2, 3, 4)	b7 b6 00: } Enables protect 01: } Enables protect 10: } Enables protect 11: Disables protect	RW RW

NOTES:
1. When the ROM code protect is active by the ROMCP1 bit setting, the flash memory is protected against reading or rewriting in parallel I/O mode.
2. Set the bit 5 to bit 0 to 1111112 when the ROMCP1 bit is set to a value other than 112. If the bit 5 to bit 0 are set to values other than 1111112, the ROM code protection may not become active by setting the ROMCP1 bit to a value other than 112.
3. To make the ROM code protection inactive, erase a block including the ROMCP address in standard serial I/O mode or CPU rewrite mode.
4. The ROMCP address is set to FF16 when a block, including the ROMCP address, is erased.
5. When a value of the ROMCP address is 0016 or FF16, the ROM code protect function is disabled.

Figure 17.3.1.1. ROMCP Address

Address		
0FFFDF16 to 0FFFDC16	ID1	Undefined instruction vector
0FFFE316 to 0FFFE016	ID2	Overflow vector
0FFFE716 to 0FFFE416		BRK instruction vector
0FFFEB16 to 0FFFE816	ID3	Address match vector
0FFFEF16 to 0FFFEC16	ID4	Single step vector
0FFFF316 to 0FFFF016	ID5	Watchdog timer vector
0FFFF716 to 0FFFF416	ID6	$\overline{\text{DBC}}$ vector
0FFFFB16 to 0FFFF816	ID7	$\overline{\text{NMI}}$ vector
0FFFFF16 to 0FFFFC16	ROMCP	Reset vector

4 bytes

Figure 17.3.2.1. Address for ID Code Stored

Downloaded from Arrow.com.

17.4 CPU Rewrite Mode

In CPU rewrite mode, the user ROM area can be rewritten when the CPU executes software commands. Therefore, the user ROM area can be rewritten directly while the microcomputer is mounted on-board without using a ROM programmer, etc. Verify the Program and the Block Erase commands are executed only on blocks in the user ROM area.

For interrupts requested during an erasing operation in CPU rewrite mode, the M16C/26A Group flash module offers an erase-suspend function which the erasing operation to be suspended, and access made available to the flash. Erase-write 0 (EW0) mode and erase-write 1 (EW1) mode are provided as CPU rewrite mode. Table 17.4.1 shows the differences between erase-write 0 (EW0) and erase-write 1 (EW1) modes. 1 wait is required for the CPU erase-write control.

Table 17.4.1. EW0 Mode and EW1 Mode

Item	EW0 mode	EW1 mode
Operation mode	Single chip mode	Single chip mode
Area where rewrite control program can be placed	User ROM area	User ROM area
Area where rewrite control program can be executed	The rewrite control program must be transferred to any area other than the flash memory (e.g., RAM) before being executed	The rewrite control program can be executed in the user ROM area
Area which can be rewritten	User ROM area	User ROM area However, this excludes blocks with the rewrite control program
Software command Restrictions	None	• Program, block erase command Cannot be executed in a block having the rewrite control program • Read status register command Can not be used
Mode after programming or erasing	Read Status Register mode	Read Array mode
CPU state during auto-write and auto-erase	Operation	Hold state (I/O ports retain the state before the command is executed (1)
Flash memory status detection (2)	• Read the FMR00, FMR06 and FMR07 bits in the FMR0 register by a program • Execute the read status register command and read the SR7, SR5 and SR4 bits	Read the FMR0 register's FMR00, FMR06, and FMR07 bits in a program
Condition for transferring to erase-suspend (3)	Set the FMR40 and FMR41 bits in the FMR4 register to "1" by program.	The FMR40 bit in the FMR4 register is set to "1" and the interrupt request of

NOTES:
1. Do not generate a DMA transfer.
2. Block 1 and 0 are enabled to rewrite by setting the FMR02 bit in the FMR0 register to "1" and setting the FMR16 bit in the FMR1 register to "1". Block 2 to 3 are enabled to rewrite by setting the FMR16 bit in the FMR1 register to "1".
3. The time, until entering erase suspend and reading flash is enabled, is maximum td (SR-ES) after satisfying the conditions.

Downloaded from Arrow.com.

17.4.1 EW0 Mode

The microcomputer enters CPU rewrite mode by setting the FMR01 bit in the FMR0 register to “1” (CPU rewrite mode enabled) and is ready to acknowledge the software commands. EW0 mode is selected by setting the FMR11 bit in the FMR1 register to “0”.

When setting the FMR01 bit to “1”, set to “1” after first writing “0”. The software commands control programming and erasing. The FMR0 register or the status register indicates whether a programming or erasing operations is completed.

When entering the erase-suspend during the auto-erasing, set the FMR40 bit to “1” (erase-suspend enabled) and the FMR41 bit to “1” (suspend request). And wait for td(SR-ES). After verifying the FMR46 bit is set to “1” (auto-erase stop), access to the user ROM area. When setting the FMR41 bit to “0” (erase restart), auto-erasing is restarted.

17.4.2 EW1 Mode

EW1 mode is selected by setting the FMR11 bit to “1” after the FMR01 bit is set to “1”. (set to “1” after first writing “0”). The FMR0 register indicates whether or not a programming or an erasing operation is completed. Do not execute the software commands of read status register in EW1 mode.

When an erase/program operation is initiated the CPU halts all program execution until the operation is completed or erase-suspend is requested.

When enabling an erase suspend function, set the FMR40 bit to “1” (erase suspend enabled) and execute block erase commands. Also, preliminarily set an interrupt to enter the erase-suspend to an interrupt enabled status. After td(SR-ES) from an interrupt request and entering erase suspend, an interrupt can be acknowledged.

When an interrupt request is generated, the FMR41 bit is automatically set to “1” (suspend request) and an auto-erasing is halted. If an auto-erasing is not completed (the FMR00 bit is “0”) after an interrupt process completed, set the FMR41 bit to “0” (erase restart) and execute block erase commands again.

Downloaded from Arrow.com.

17.5 Register Description

Figure 17.5.1 shows the flash memory control register 0 and flash memory control register 1. Figure 17.5.2 shows the flash memory control register 4.

17.5.1 Flash memory control register 0 (FMR0)

•FMR 00 Bit

This bit indicates the operation status of the flash memory. The bit is “0” during programming, erasing, or erase-suspend mode; otherwise, the bit is “1”.

•FMR01 Bit

The microcomputer enables to acknowledge commands by setting the FMR01 bit to “1” (CPU rewrite mode). To set this bit to “1”, it is necessary to set to “1” after first setting to “0”. Set this bit to “0” by only writing “0”.

•FMR02 Bit

The combined setting of the FMR02 bit and the FMR16 bit enable to program and erase in the user ROM area. See Table 17.5.2.1 for setting details. To set this bit to “1”, it is necessary to set to “1” after first setting to “0”. Set this bit to “0” by only writing “0”. This bit is enabled only when the FMR01 bit is “1” (CPU rewrite mode enable).

•FMSTP Bit

This bit resets the flash memory control circuits and minimizes power consumption in the flash memory. Access to the flash memory is disabled when the FMSTP bit is set to “1”. Set the FMSTP bit by a program in a space other than the flash memory.

Set the FMSTP bit to “1” if one of the following occurs:

- A flash memory access error occurs during erasing or programming in EW0 mode (FMR00 bit does not switch back to “1” (ready)).
- Low-power consumption mode or on-chip oscillator low-power consumption mode is entered.

Figure 17.5.1.3 shows a flow chart illustrating how to start and stop the flash memory before and after entering low power mode. Follow the procedure on this flow chart.

To enter stop or wait mode when CPU rewrite mode is disabled, do not set the FMR0 register. The flash memory is automatically turned off when entering and turned back on when exiting.

•FMR06 Bit

This is a read-only bit indicating an auto-program operation status. This bit is set to “1” when a program error occurs; otherwise, it is set to “0”. For details, refer to **17.8.4 Full Status Check**.

•FMR07 Bit

This is a read-only bit indicating an auto-erase operation status. The bit is set to “1” when an erase error occurs; otherwise, it is set to “0”. For details, refer to **17.8.4 Full Status Check**.

Figure 17.5.1.1 shows a EW0 mode set/reset flowchart, figure 17.5.1.2 shows a EW1 mode set/reset flowchart.

Downloaded from Arrow.com.

17.5.2 Flash memory control register 1 (FMR1)

•FMR11 Bit

EW1 mode is entered by setting the FMR11 bit to “1” (EW1 mode). This bit is enabled only when the FMR01 bit is “1”.

•FMR16 Bit

The combined setting of the FMR02 bit and the FMR16 bit enables to program and erase in the user ROM area. To set this bit to “1”, it is necessary to set to “1” after first setting to “0”. Set this bit to “0” by only writing “0”. This bit is enabled only when the FMR01 bit is “1”.

•FMR17 Bit

If FMR17 bit is “1” (with wait state), regardless of the content of the PM17 bit, 1 wait is inserted at the access to block A and block B. Regardless of the content of the FMR17 bit, access to other block and the internal RAM is determined by PM17 bit setting.

Set this bit to “1” (with wait state) when rewriting more than 100 times (U7 and U9).

Table 17.5.2.1. Protection using FMR16 and FMR02

FMR16	FMR02	Block A, Block B	Block 0, Block 1	other user block
0	0	write enabled	write disabled	write disabled
0	1	write enabled	write disabled	write disabled
1	0	write enabled	write disabled	write enabled
1	1	write enabled	write enabled	write enabled

17.5.3 Flash memory control register 4 (FMR4)

•FMR40 Bit

The erase-suspend function is enabled by setting the FMR40 bit is set to “1” (enabled).

•FMR41 Bit

When setting the FMR41 bit to “1” in a program during auto-erasing in EW0 mode the flash module enters erase suspend mode. In EW1 mode, the FMR41 bit is automatically set to “1” (suspend request) when an interrupt request of an enabled interrupt is generated, the FMR41 bit is automatically set to “1” (suspend request) and when an auto-erasing operation is restarted, set the FMR41 bit to “0” (erase restart).

•FMR46 Bit

The FMR46 bit is set to “0” during auto-erasing execution and set to “1” during erase-suspend mode. Do not access to flash memory while this bit is “0”.

Downloaded from Arrow.com.

Flash memory control register 0

b7	b6	b5	b4	b3	b2	b1	b0
		0	0				

Symbol	Address	After reset
FMR0	$01B7_{16}$	00000001_2

Bit symbol	Bit name	Function	RW
FMR00	RY/$\overline{BY}$ status flag	0: Busy (during writing or erasing) 1: Ready	RO
FMR01	CPU rewrite mode select bit (1)	0: Disables CPU rewrite mode (Disables software command) 1: Enables CPU rewrite mode (Enables software commands)	RW
FMR02	Block 0, 1 rewrite enable bit (2)	Set write protection for user ROM area (see **Table 17.5.2.1**)	RW
FMSTP	Flash memory stop bit (3, 5)	0: Starts flash memory operation 1: Stops flash memory operation (Enters low-power consumption state and flash memory reset)	RW
($\overline{b5}$-$\overline{b4}$)	Reserved bit	Set to “0”	RW
FMR06	Program status flag (4)	0: Terminated normally 1: Terminated in error	RO
FMR07	Erase status flag (4)	0: Terminated normally 1: Terminated in error	RO

NOTES:
1. When setting this bit to “1”, set to “1” immdediately after setting it first to “0”. Do not generate an interrupt or a DMA transfer between setting the bit to “0” and setting it to “1”. Set this bit while the P8$_5$/$\overline{NMI}$/$\overline{SD}$ pin is “H” when selecting the $\overline{NMI}$ function. Set by program in a space other than the flash memory in EW0 mode. Set this bit to read alley mode and “0”
2. Set this bit to “1” immediately after setting it first to “0” while the FMR01 bit is set to “1”. Do not generate an interrupt or a DMA transfer between setting this bit to “0” and setting it to “1”.
3. Set this bit in a pace other than the flash memory by program. When this bit is set to 1, access to flash memory will be denied. To set this bit to 0 after setting it to 1, wait for 10 µsec. or more after setting it to 1. To read data from flash memory after setting this bit to 0, maintain tps wait time before accessing flash memory.
4. This bit is set to “0” by executing the clear status command.
5. This bit is enabled when the FMR01 bit is set to “1” (CPU rewrite mode). This bit can be set to “1” when the FMR01 bit is set to “0”. However, the flash memory does not enter low-power consumption status

Flash memory control register 1

b7	b6	b5	b4	b3	b2	b1	b0
		0	X	X	X		X

Symbol	Address	After reset
FMR1	$01B5_{16}$	$000XXX0X_2$

Bit symbol	Bit name	Function	RW
($\overline{b0}$)	Reserved bit	When read, its content is indeterminate	RO
FMR11	EW1 mode select bit (1)	0: EW0 mode 1: EW1 mode	RW
($\overline{b3}$-$\overline{b2}$)	Reserved bit	When read, its content is indeterminate	RO
($\overline{b4}$)	Nothing is assigned. When write, set to “0”. When read, its contect is indeterminate.		—
($\overline{b5}$)	Reserved bit	Set to “0”	RW
FMR16	Block 0 to 3 rewrite enable bit (2)	Set write protection for user ROM area (see **Table 17.5.2.1**) 0: Disable 1: Enable	RW
FMR17	Block A, B access wait bit (3)	0: PM17 enabled 1: With wait state (1 wait)	RW

NOTES:
1. Set this bit to “1” immediately after setting it first to “0”. Do not generate an interrupt or a DMA transfer between setting the bit to “0” and setting it to “1”. Set this bit while the P8$_5$/$\overline{NMI}$/$\overline{SD}$ pin is “H” when the $\overline{NMI}$ function is selected. If the FMR01 bit is set to “0”, the FMR01 bit and FMR11 bit are both set to “0”
2. Set this bit to “1” immediately after setting it first to “0”. Do not generate an interrupt or a DMA transfer after setting to “0”.
3. When rewriting more than 100 times, set this bit to “1” (with wait state). When the FMR17 bit is “1” (with wait state), regardless of the content of the PM17 bit, 1 wait is inserted at the access to the block A and B. Regardless of the content of the FMR17 bit, access to other block and the internal RAM is determined be PM17 bit setting.

Figure 17.5.1. FMR0 and FMR1 register

Downloaded from Arrow.com.

Flash Memory Control Register 4

b7	b6	b5	b4	b3	b2	b1	b0
0		0	0	0	0		

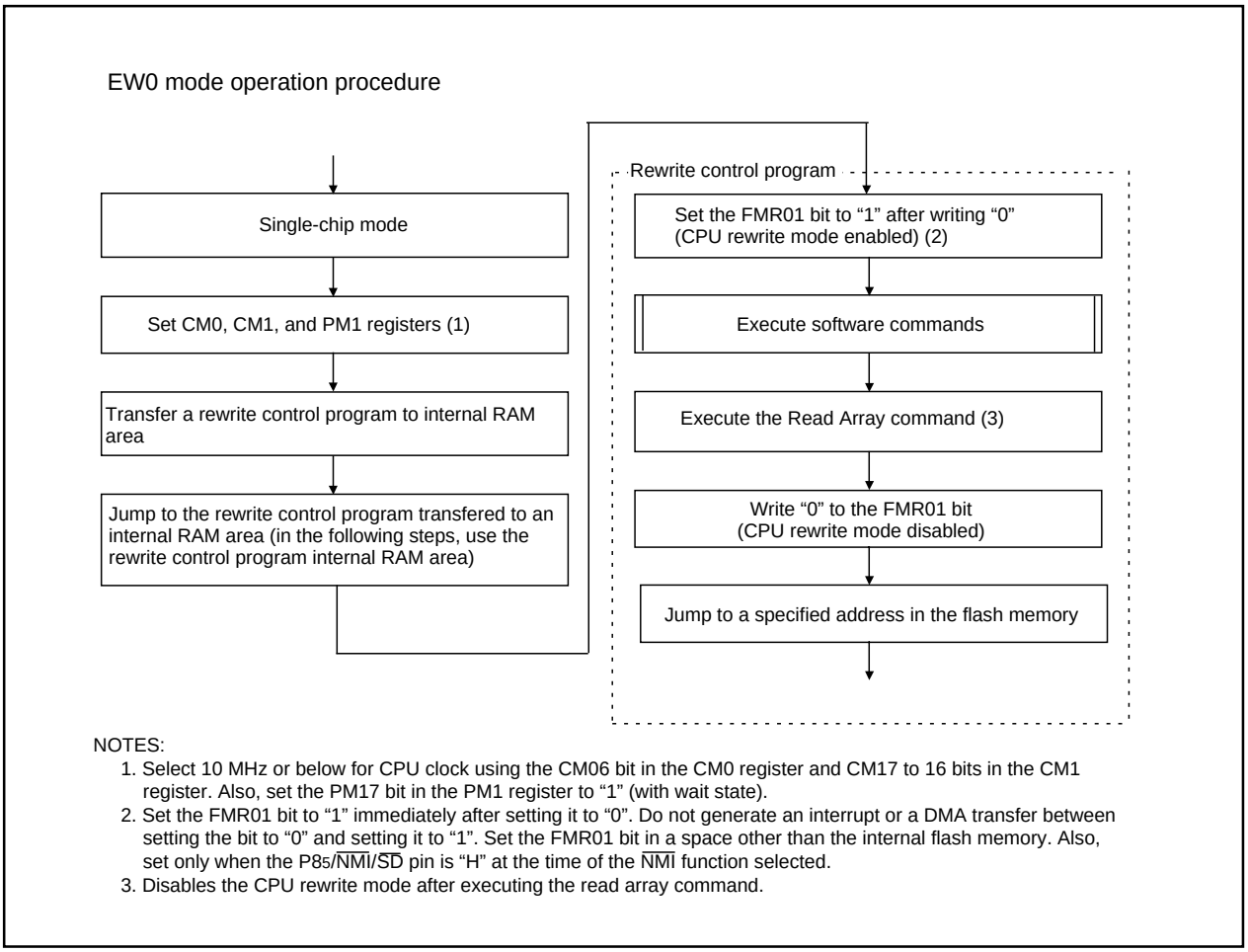
Symbol	Address	After Reset
FMR4	$01B3_{16}$	01000000_2

Bit Symbol	Bit Name	Function	RW
FMR40	Erase suspend function enable bit (1)	0: Disabled 1: Enabled	RW
FMR41	Erase suspend request bit (2)	0: Erase restart 1: Suspend request	RW
$\overline{(b5\text{-}b2)}$	Reserved bit	Set to 0	RO
FMR46	Erase status	0: During auto-erase operation 1: Auto-erase stop (erase suspend mode)	RO
$\overline{(b7)}$	Reserved bit	Set to 0	RW

NOTES:
1. Set the FMR40 bit to 1 immediately after setting it first to 0. Do not generate any interrupt or DMA transfer between setting the bit to 0 and setting it to 1. Set by program in space other than the flash memory in EW mode 0.
2. The FMR41 bit is valid only when the FMR40 bit is set to 1. The FMR41 bit can be written only between executing an erase command and completing erase (this bit is set to 0 other than the above duration). The FMR41 bit can be set to 0 or 1 by program in EW mode 0. In EW mode 1, the FMR41 bit is automatically set to 1 when the FMR40 bit is 1 and a maskable interrupt is generated during erasing. The FMR41 bit cannot be set to 1 by program (it can be set to 0 by program).

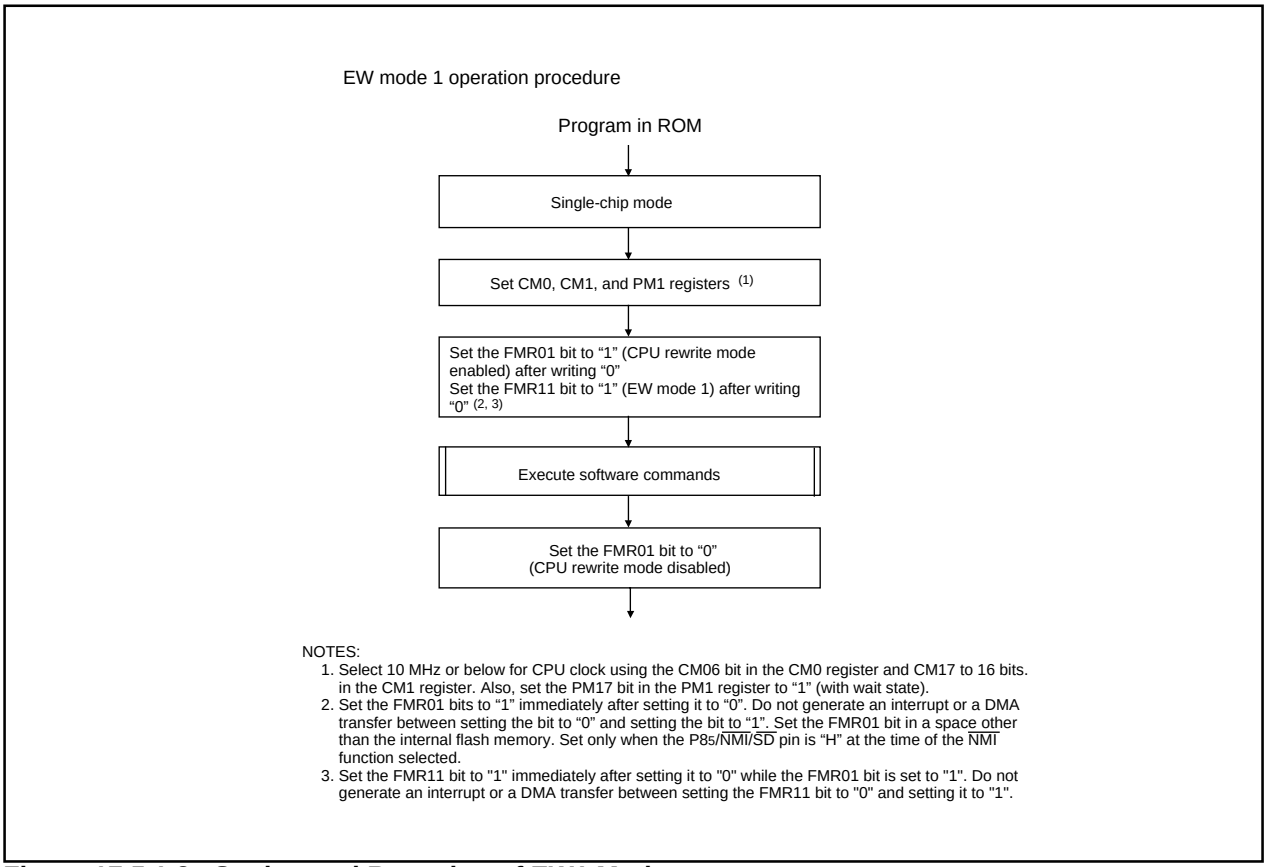
Figure 17.5.2. FMR4 register

Downloaded from Arrow.com.



NOTES:
1. Select 10 MHz or below for CPU clock using the CM06 bit in the CM0 register and CM17 to 16 bits in the CM1 register. Also, set the PM17 bit in the PM1 register to “1” (with wait state).
2. Set the FMR01 bit to “1” immediately after setting it to “0”. Do not generate an interrupt or a DMA transfer between setting the bit to “0” and setting it to “1”. Set the FMR01 bit in a space other than the internal flash memory. Also, set only when the P8$_5$/$\overline{\text{NMI}}$/$\overline{\text{SD}}$ pin is “H” at the time of the $\overline{\text{NMI}}$ function selected.
3. Disables the CPU rewrite mode after executing the read array command.

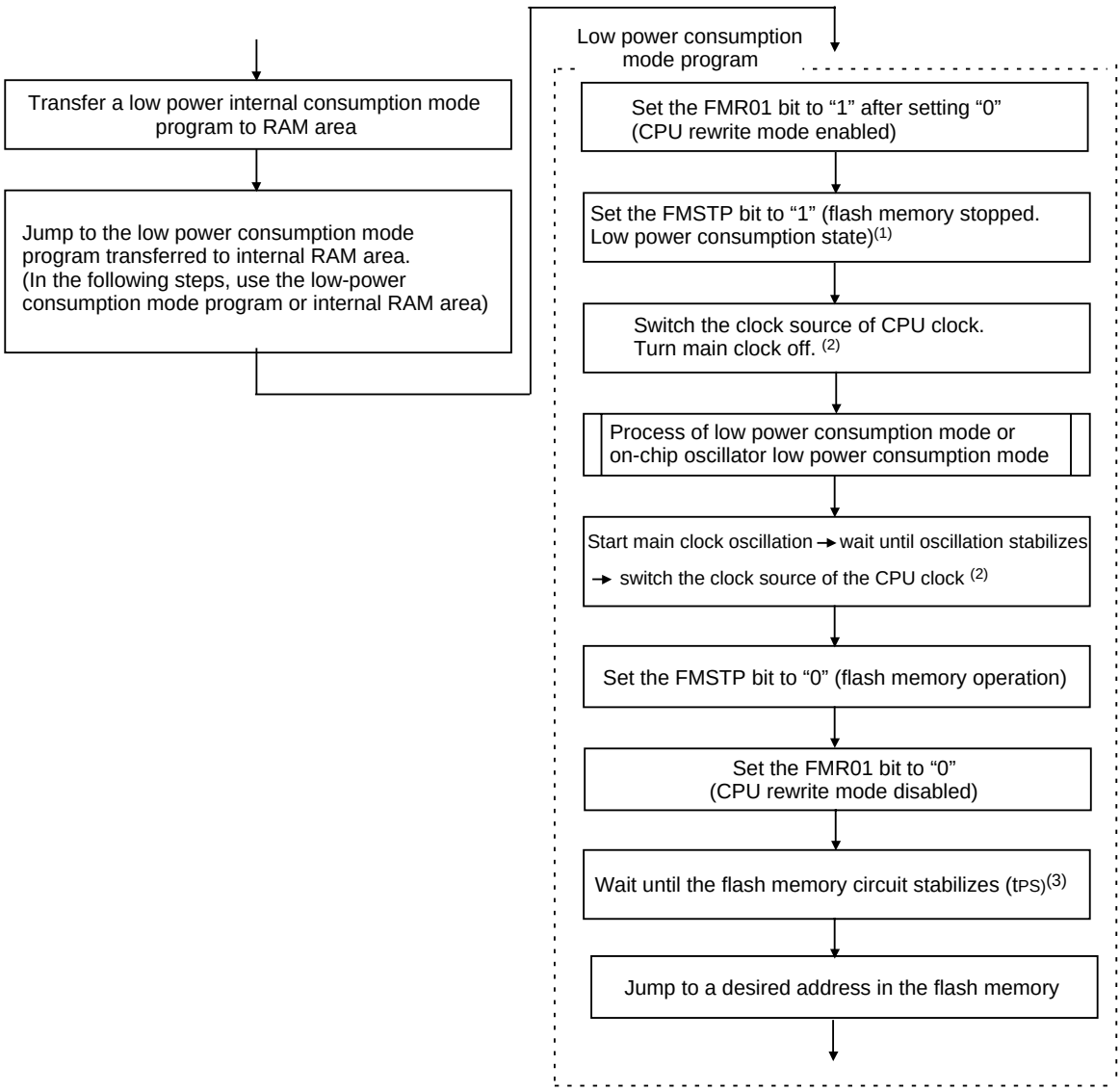
Figure 17.5.1.1. Setting and Resetting of EW0 Mode



NOTES:
1. Select 10 MHz or below for CPU clock using the CM06 bit in the CM0 register and CM17 to 16 bits. in the CM1 register. Also, set the PM17 bit in the PM1 register to “1” (with wait state).
2. Set the FMR01 bits to “1” immediately after setting it to “0”. Do not generate an interrupt or a DMA transfer between setting the bit to “0” and setting the bit to “1”. Set the FMR01 bit in a space other than the internal flash memory. Set only when the P8$_5$/$\overline{\text{NMI}}$/$\overline{\text{SD}}$ pin is “H” at the time of the $\overline{\text{NMI}}$ function selected.
3. Set the FMR11 bit to "1" immediately after setting it to "0" while the FMR01 bit is set to "1". Do not generate an interrupt or a DMA transfer between setting the FMR11 bit to "0" and setting it to "1".

Figure 17.5.1.2. Setting and Resetting of EW1 Mode

Downloaded from Arrow.com.



NOTES:
1. Set the FMRSTP bit to “1” after setting the FMR01 bit to “1”(CPU rewrite mode).
2. Wait until the clock stabilizes to switch the clock source of the CPU clock to the main clock or the sub clock.
3. Add a tPS wait time by a program. Do not access the flash memory during this wait time.

Figure 17.5.1.3. Processing Before and After Low Power Dissipation Mode

Downloaded from Arrow.com.

17.6 Precautions in CPU Rewrite Mode

Described below are the precautions to be observed when rewriting the flash memory in CPU rewrite mode.

17.6.1 Operation Speed

When CPU clock source is the main clock, before entering CPU rewrite mode (EW0 or EW1 mode), select 10 MHz or below for CPU clock using the CM06 bit in the CM0 register and the CM17 to CM16 bits in the CM1 register. Also, when selecting f_3(ROC) of a on-chip oscillator as a CPU clock source, before entering CPU rewrite mode (EW0 or EW1 mode), the ROCR3 to ROCR2 bits in the ROCR register set the CPU clock division rate to “divide-by-4” or “divide-by-8”.
On both cases, set the PM17 bit in the PM1 register to “1” (with wait state).

17.6.2 Prohibited Instructions

The following instructions cannot be used in EW0 mode because the CPU tries to read data in the flash memory: UND instruction, INTO instruction, JMPS instruction, JSRS instruction, and BRK instruction

17.6.3 Interrupts

EW0 Mode

- To use interrupts having vectors in a relocatable vector table, the vectors must be relocated to the RAM area.
- The $\overline{\text{NMI}}$ and watchdog timer interrupts can be used since the FMR0 and FMR1 registers are forcibly reset when either interrupt is generated. However, the jump addresses for each interrupt service routines to the fixed vector table are set and interrupt programs are required. Flash memory rewrite operation is halted when the NMI or watchdog timer interrupt is generated. Set the FMR01 bit to “1” and execute the rewrite and erase program again after exiting the interrupt routine.
- The address match interrupt can not be used since the CPU tries to read data in the flash memory.

EW1 Mode

- Do not acknowledge any interrupts with vectors in the relocatable vector table or the address match interrupt during the auto-program or erase-suspend function.

17.6.4 How to Access

To set the FMR01, FMR02, FMR11 or FMR16 bit to “1”, write “1” after first setting the bit to “0”. Do not generate an interrupt or a DMA transfer between the instruction to set the bit to “0” and the instruction to set it to “1”. When the $\overline{\text{NMI}}$ function is selected, set the bit while an “H” signal is applied to the $P8_5$/$\overline{\text{NMI}}$/$\overline{\text{SD}}$ pin.

17.6.5 Writing in the User ROM Space

17.6.5.1 EW0 Mode

- If the supply voltage drops while rewriting the block where the rewrite control program is stored, the flash memory can not be rewritten, because the rewrite control program is not correctly rewritten. If this error occurs, rewrite the user ROM area in standard serial I/O mode or parallel I/O mode.

17.6.5.2 EW1 Mode

- Do not rewrite the block where the rewrite control program is stored.

Downloaded from Arrow.com.

17.6.6 DMA Transfer

In EW1 mode, do not perform a DMA transfer while the FMR00 bit in the FMR0 register is set to “0”. (the auto-programming or auto-erasing duration).

17.6.7 Writing Command and Data

Write the command code and data to even addresses in the user ROM area.

17.6.8 Wait Mode

When entering wait mode, set the FMR01 bit to “0” (CPU rewrite mode disabled) before executing the WAIT instruction.

17.6.9 Stop Mode

When entering stop mode, set the FMR01 bit to “0” (CPU rewrite mode disabled) and disable the DMA transfer before setting the CM10 bit to “1” (stop mode).

17.6.10 Low Power Consumption Mode and On-chip Oscillator-Low Power Consumption Mode

If the CM05 bit is set to “1” (main clock stopped), do not execute the following commands.

- Program
- Block erase

Downloaded from Arrow.com.

17.7 Software Commands

Read or write 16-bit commands and data from or to even addresses in the user ROM area. When writing a command code, 8 high-order bits (D_{15}–D_8) are ignored.

Table 17.7.1. Software Commands

Command	First bus cycle			Second bus cycle		
	Mode	Address	Data (D_{15} to D_0)	Mode	Address	Data (D_{15} to D_0)
Read array	Write	X	xxFF16			
Read status register	Write	X	xx7016	Read	X	SRD
Clear status register	Write	X	xx5016			
Program	Write	WA	xx4016	Write	WA	WD
Block erase	Write	X	xx2016	Write	BA	xxD016

SRD: Status register data (D_7 to D_0)
WA : Write address (However,even address)
WD : Write data (16 bits)
BA : Highest-order block address (However,even address)
X : Any even address in the user ROM area
xx : 8 high-order bits of command code (ignored)

17.7.1 Read Array Command (FF16)

This command reads the flash memory.

By writing command code 'xxFF16' in the first bus cycle, read array mode is entered. Content of a specified address can be read in 16-bit unit after the next bus cycle. The microcomputer remains in read array mode until an another command is written. Therefore, contents of multiple addresses can be read consecutively.

17.7.2 Read Status Register Command (7016)

This command reads the status register.

By writing command code 'xx7016' in the first bus cycle, the status register can be read in the second bus cycle (Refer to **17.8 Status Register**). Read an even address in the user ROM area. Do not execute this command in EW1 mode.

Downloaded from Arrow.com.

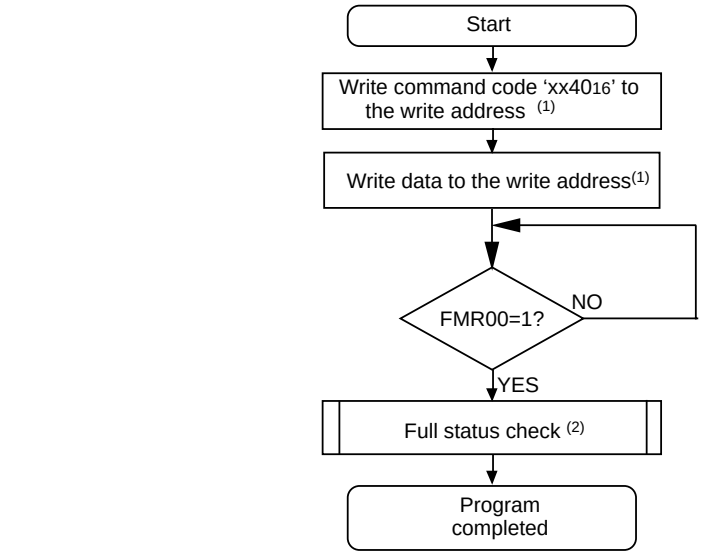
17.7.3 Clear Status Register Command (5016)

This command clears the status register to “0”.

By writing ‘xx5016’ in the first bus cycle, and the FMR06 to FMR07 bits in the FMR0 register and SR4 to SR5 bits in the status register are set to “0”.

17.7.4 Program Command (4016)

The program command writes 2-byte data to the flash memory. By writing ‘xx4016’ in the first bus cycle and data to the write address specified in the second bus cycle, the auto-programming/erasing (data prorgramming and verify) start. Set the address value specified in the first bus cycle to same and even address as the write address specified in the second bus cycle. The FMR00 bit in the FMR0 register indicates whether an auto-programming operation has been completed. The FMR00 bit is set to “0” during the auto-programming and “1” when the auto-programming operation is completed. After the auto-programming operation is completed, the FMR06 bit in the FMR0 register indicates whether or not the auto-programming operation has been completed as expected. (Refer to **17.8.4 Full Status Check**). Also, each block disables writing (Refer to “Table 17.5.2.1”). Do not write additions to the address which is already programmed. When commands other than a program command are executed immediately after a program command, set the same address as the write address specified in the second bus cycle of the program command, to the specified address value in the first bus cycle of the following command. In EW1 mode, do not execute this command on the blocks where the rewrite control program is allocated. In EW0 mode, the microcomputer enters read status register mode as soon as the auto-programming operation starts and the status register can be read. The SR7 bit in the status register is set to “0” as soon as the auto-programming operation starts. This bit is set to “1” when the auto-programming operation is completed. The microcomputer remains in read status register mode until the read array command is written. After completion of the auto-programming operation, the status register indicates whether or not the auto-programming operation has been completed as expected.



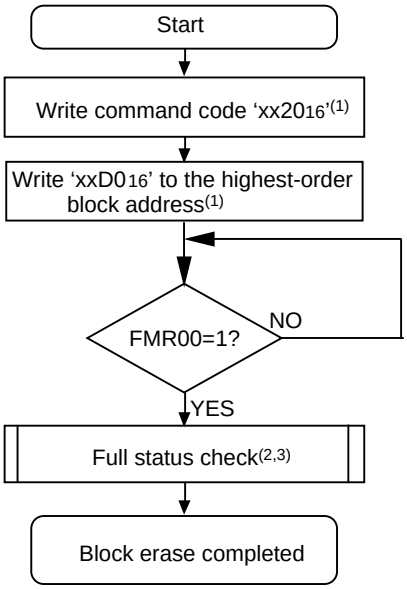
NOTE:
1: Write the command code and data at even address.Note 2: Refer to "**Figure 17.8.4.1**. Full Status Check and Handling Procedure for Each Error"

Figure 17.7.4.1. Flow Chart of Program Command

Downloaded from Arrow.com.

17.7.5 Block Erase

By writing 'xx20$_{16}$' in the first bus cycle and 'xxD0$_{16}$' in the second bus cycle to the highest-order (even addresse of a block) and the auto-programming/erasing (erase and erase verify) start. The FMR00 bit in the FMR0 register indicates whether the auto-programming operation has been completed. The FMR00 bit is set to "0" (busy) during the auto-erasing operation and "1" (ready) when the auto-erasing operation is completed. When using the erase-suspend function in EW0 mode, the FMR46 bit in the FMR4 register indicates whether a flash memory has entered erase-suspend mode. The FMR46 bit is set to "0" during auto-erasing operation and "1" when the auto-erasing operation is completed (entering erase-suspend). After the completion of an auto-erasing operation, the FMR07 bit in the FMR0 register indicates whether or not the auto erasing-operation has been completed as expected. (Refer to **17.8.4 Full Status Check**). Also, each block disables erasing. (Refer to "Table 17.5.2.1"). Figure 17.7.5.1 shows a flow chart of the block erase command programming when not using the erase-suspend function. Figure 17.7.5.2 shows a flow chart of the block erase command programming when using an erase-suspend function. In EW1 mode, do not execute this command on the block where the rewrite control program is allocated. In EW0 mode, the microcomputer enters read status register mode as soon as the auto-erasing operation starts and the status register can be read. The SR7 bit in the status register is set to "0" as soon as the auto-erasing operation starts. This bit is set to "1" when the auto-erasing operation is completed. The microcomputer remains in read status register mode until the read array command is written. Also excute the clear status register command and block erase command at least 3 times until an erase error is not generated when an erase error is generated.

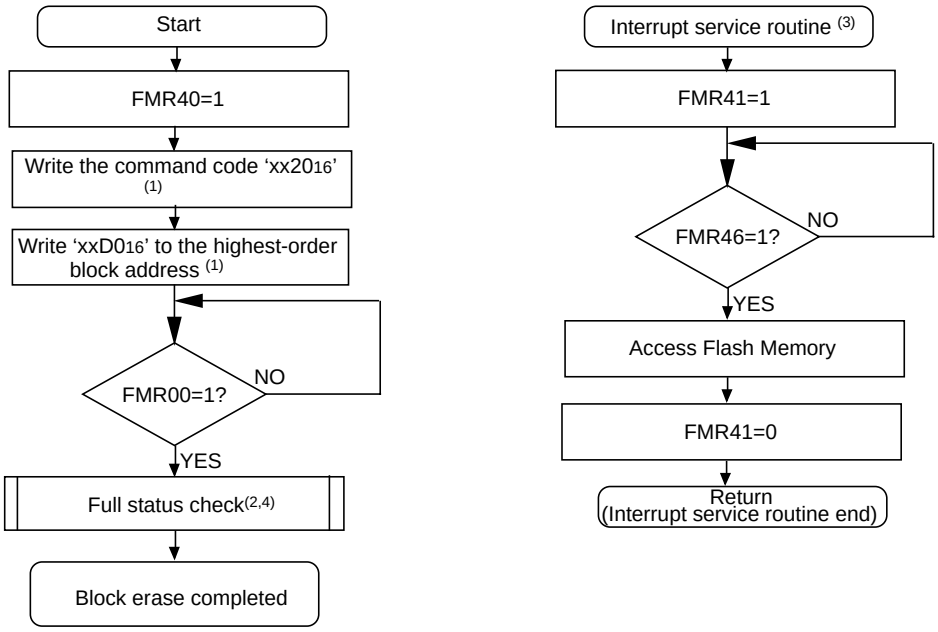


NOTES:
1. Write the command code and data at even address.
2. Refer to "**Figure 17.8.4.1. Full Status Check and Handling Porcedure for Each Error**".
3. Execute the clear status register command and block erase command at least 3 times until an erase error is not generated when an erase error is generated.

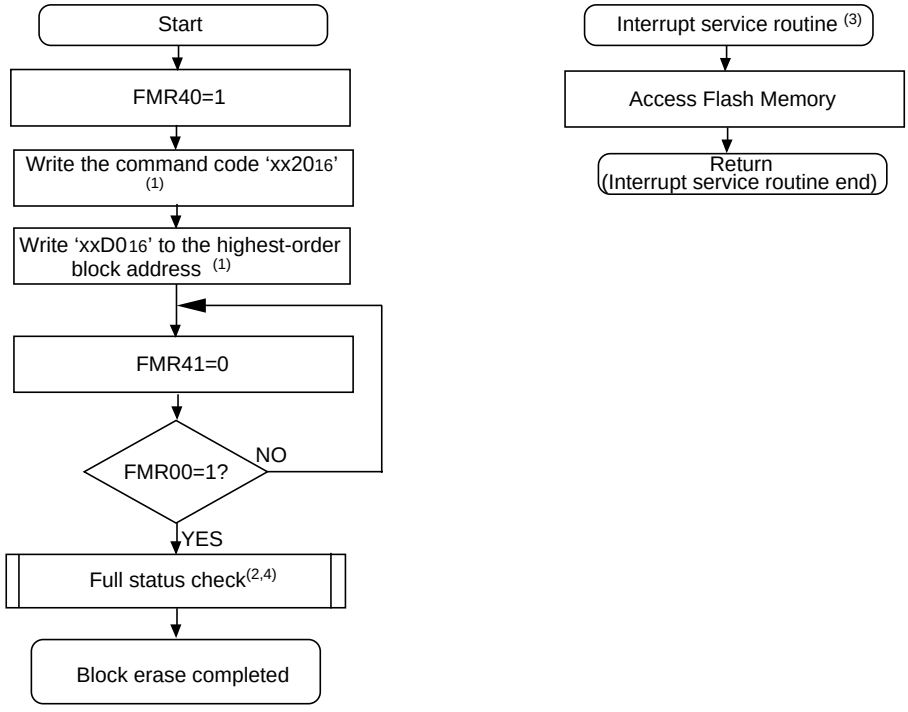
Figure 17.7.5.1. Flow Chart of Block Erase Command (when not using erase suspend function)

Downloaded from Arrow.com.

(EW0 mode)



(EW1 mode)



NOTES:
1. Write the command code and data to even address.
2. Execute the clear status register command and block erase command at least 3 times until an erase error is not generated when an erase error is generated.
3. In EW0 mode, allocate an interrupt vector table of an interrupt, to be used, to a RAM area.
4. Refer to "**Figure 17.8.4.1 Full Status Check and Handling Procedure for Each Error**."

Figure 17.7.5.2. Block Erase Command (at use erase suspend)

Downloaded from Arrow.com.

17.8 Status Register

The status register indicates the operating status of the flash memory and whether an erasing or a programming operates normally and an error ends. The FMR00, FMR06, and FMR07 bits in the FMR0 register indicate the status of the status register.

Table 17.8.1 shows the status register.

In EW0 mode, the status register can be read in the following cases:

(1) When a given even address in the user ROM area is read after writing the read status register command

(2) When a given even address in the user ROM area is read after executing the program or block erase command but before executing the read a rray command.

17.8.1 Sequence Status (SR7 and FMR00 Bits)

The sequence status indicates the operating status of the flash memory. This bit is set to “0” (busy) during an auto-programming and auto-erasing and “1” (ready) as soon as these operations are completed. This bit indicates “0” (busy) in erase-suspend mode.

17.8.2 Erase Status (SR5 and FMR07 Bits)

Refer to **17.8.4 Full Status Check**.

17.8.3 Program Status (SR4 and FMR06 Bits)

Refer to **17.8.4 Full Status Check**.

Table 17.8.1. Status Register

Bits in the SRD register	Bits in the FMR0 register	Status name	Contents		Value after reset
			"0"	"1"	
SR7 (D_7)	FMR00	Sequence status	Busy	Ready	1
SR6 (D_6)	———	Reserved	-	-	———
SR5 (D_5)	FMR07	Erase status	Completed normally	Terminated by error	0
SR4 (D_4)	FMR06	Program status	Completed normally	Terminated by error	0
SR3 (D_3)	———	Reserved	-	-	———
SR2 (D_2)	———	Reserved	-	-	———
SR1 (D_1)	———	Reserved	-	-	———
SR0 (D_0)	———	Reserved	-	-	———

- D_7 to D_0: Indicates the data bus which is read out when executing the read status register command.
- The FMR07 bit (SR5) and FMR06 bit (SR4) are set to “0” by executing the clear status register command.
- When the FMR07 bit (SR5) or FMR06 bit (SR4) is 1, the program, and block erase command are not acknowledged.

Downloaded from Arrow.com.

17.8.4 Full Status Check

When an error occurs, the FMR06 to FMR07 bits in the FMR0 register are set to “1”, indicating occurrence of each specific error. Therefore, execution results can be verified by checking these status bits (full status check). Table 17.8.4.1 shows errors and the status of FMR0 register. Figure 17.8.4.1 shows a flow chart of the full status check and handling procedure for each error.

Table 17.8.4.1. Errors and FMR0 Register Status

FMR0 register (SRD register) status		Error	Error occurrence condition
FMR07 (SR5)	FMR06 (SR4)		
1	1	Command sequence error	• When any commands are not written correctly • A value other than ‘xxD0$_{16}$’ or ‘xxFF$_{16}$’ is written in the second bus cycle of the block erase command (1) • When the block erase command is executed on protected blocks • When the program command is executed on protected blocks
1	0	Erase error	• When the block erase command is executed on unprotected blocks but the blocks are not automatically erased correctly
0	1	Program error	• When the program command is executed on unprotected blocks but the blocks are not automatically programmed correctly.

NOTE:

1. The flash memory enters read array mode by writing command code ‘xxFF$_{16}$’ in the second bus cycle of these commands. The command code written in the first bus cycle becomes invalid.

Downloaded from Arrow.com.

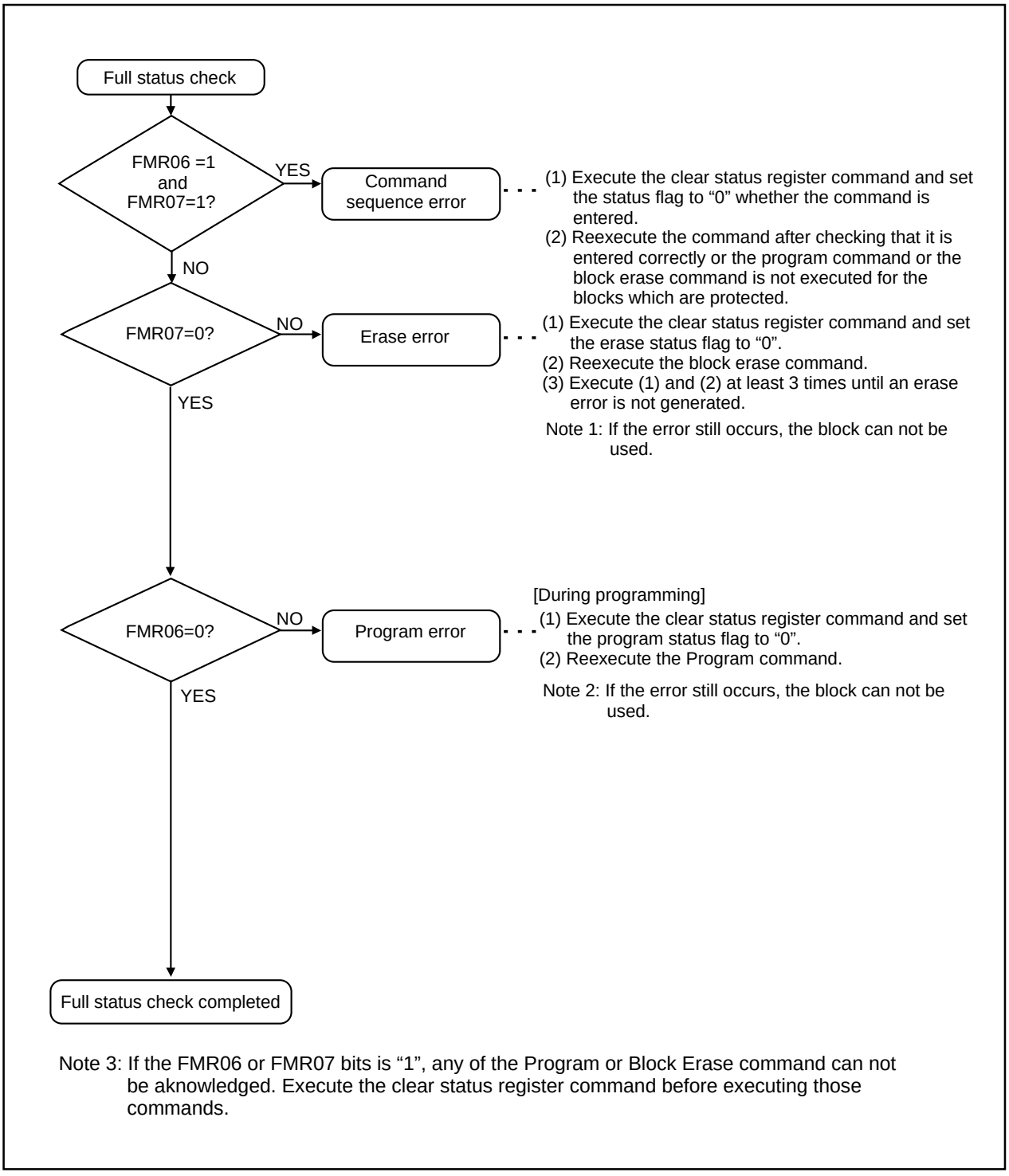


Figure 17.8.4.1. Full Status Check and Handling Procedure for Each Error

Downloaded from Arrow.com.

17.9 Standard Serial I/O Mode

In standard serial input/output mode, the user ROM area can be rewritten while the microcomputer is mounted on-board by using a serial programmer which is applicable for the M16C/26A group. For more information about serial programmers, contact the manufacturer of your serial programmer. For details on how to use the serial programmer, refer to the user's manual included with your serial programmer.
Table 17.9.1 shows pin functions (flash memory standard serial input/output mode). Figures 17.9.1 and 17.9.2 show pin connections for standard serial input/output mode.

17.9.1 ID Code Check Function

This function determines whether the ID codes sent from the serial programmer and those written in the flash memory match. (Refer to **17.3 Functions To Prevent Flash Memory from Rewriting.**)

Downloaded from Arrow.com.

Table 17.9.1. Pin Functions (Flash Memory Standard Serial I/O Mode)

Pin	Name	I/O	Description
VCC,VSS	Power input		Apply the voltage guaranteed for Program and Erase to Vcc pin and 0 V to Vss pin.
CNVSS	CNVSS	I	Connect to Vcc pin.
$\overline{\text{RESET}}$	Reset input	I	Reset input pin. While $\overline{\text{RESET}}$ pin is "L" level, wait for td(ROC).
XIN	Clock input	I	Connect a ceramic resonator or crystal oscillator between XIN and XOUT pins. To input an externally generated clock, input it to XIN pin and open XOUT pin.
XOUT	Clock output	O	
AVCC, AVSS	Analog power supply input		Connect AVss to Vss and AVcc to Vcc, respectively.
VREF	Reference voltage input	I	Enter the reference voltage for AD from this pin.
P15, P17	Input port P1	I	Input "H" or "L" level signal or open.
P16	P16 input	I	Connect this pin to Vcc while $\overline{\text{RESET}}$ is low. (2)
P60 to P63	Input port P6	I	Input "H" or "L" level signal or open.
P64	BUSY output	O	Standard serial I/O mode 1: BUSY signal output pin Standard serial I/O mode 2: Monitor signal output pin for boot program operation check
P65	SCLK input	I	Standard serial I/O mode 1: Serial clock input pin Standard serial I/O mode 2: Input "L".
P66	RxD input	I	Serial data input pin
P67	TxD output	O	Serial data output pin (1)
P70 to P77	Input port P7	I	Input "H" or "L" level signal or open.
P80 to P84, P87	Input port P8	I	Input "H" or "L" level signal or open.
P85	$\overline{\text{RP}}$ input	I	Connect this pin to Vss while $\overline{\text{RESET}}$ is low. (2)
P86	$\overline{\text{CE}}$ input	I	Connect this pin to Vcc while $\overline{\text{RESET}}$ is low. (2)
P90 to P93,	Input port P9	I	Input "H" or "L" level signal or open.
P100 to P107	Input port P10	I	Input "H" or "L" level signal or open.

NOTES:

1. When using standard serial input/output mode 1, to input “H” to the TxD pin is necessary while the $\overline{\text{RESET}}$ pin is “L”. Therefore, connect this pin to VCC via a resistor. Adjust the pull-up resistor value on a system not to affect a data transfer after reset, because this pin changes to a data-output pin
2. Set following either or both
   •Connect the $\overline{\text{CE}}$ pin to VCC.
   •Connect the $\overline{\text{RP}}$ pin to VSS and the P16 pin to VCC.

Downloaded from Arrow.com.

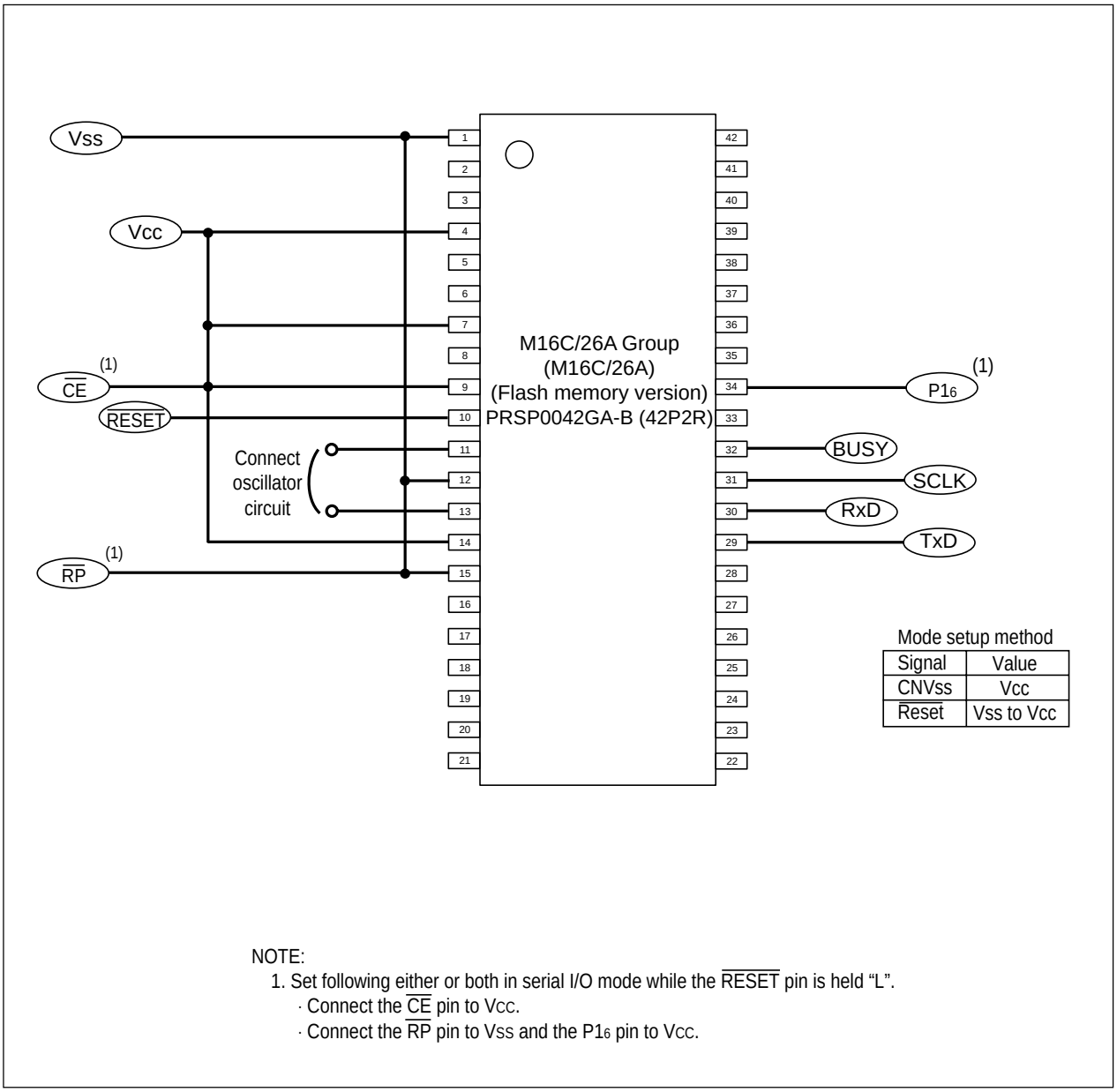
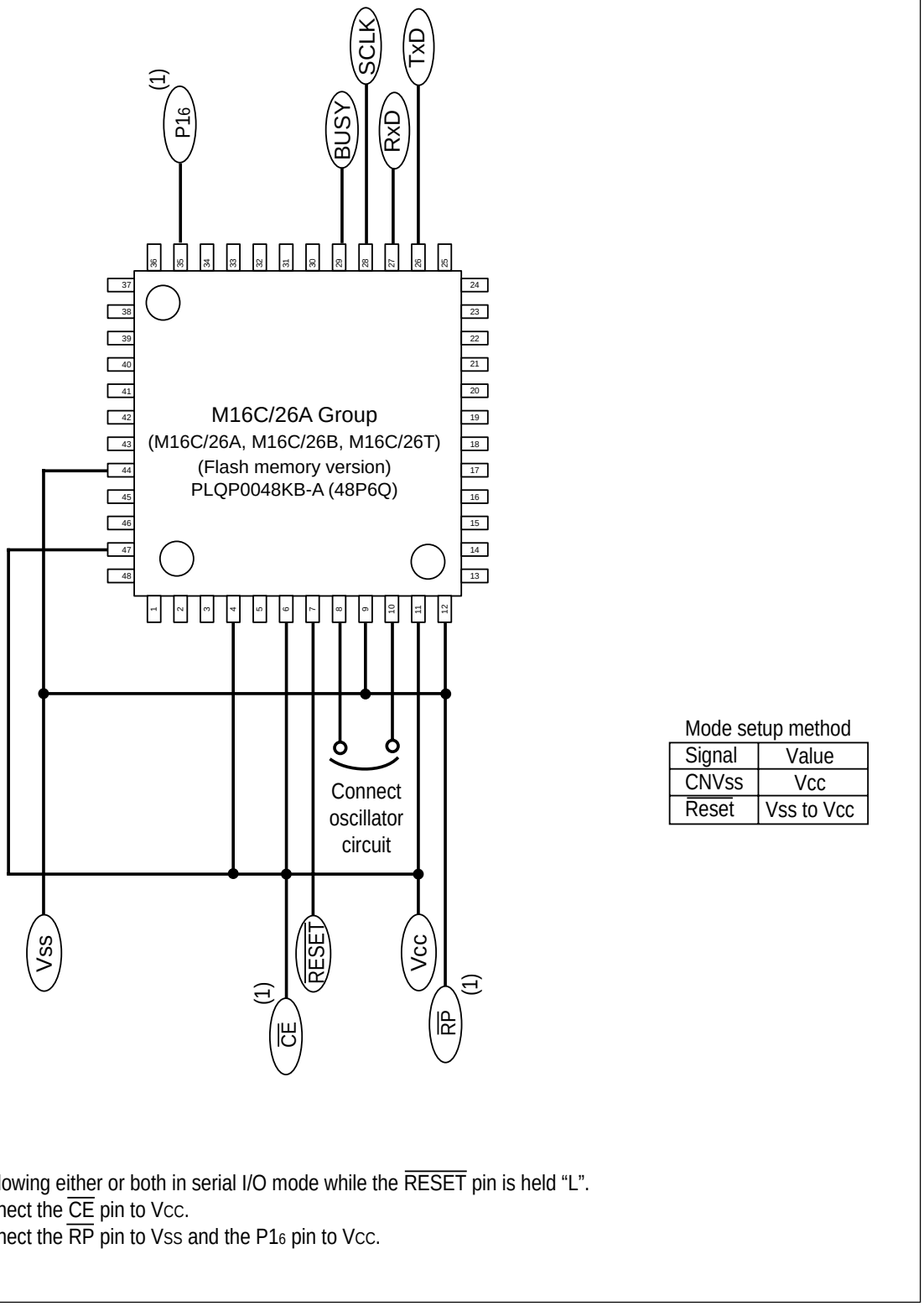


Figure 17.9.1. Pin Connections for Serial I/O Mode (1)

Downloaded from Arrow.com.



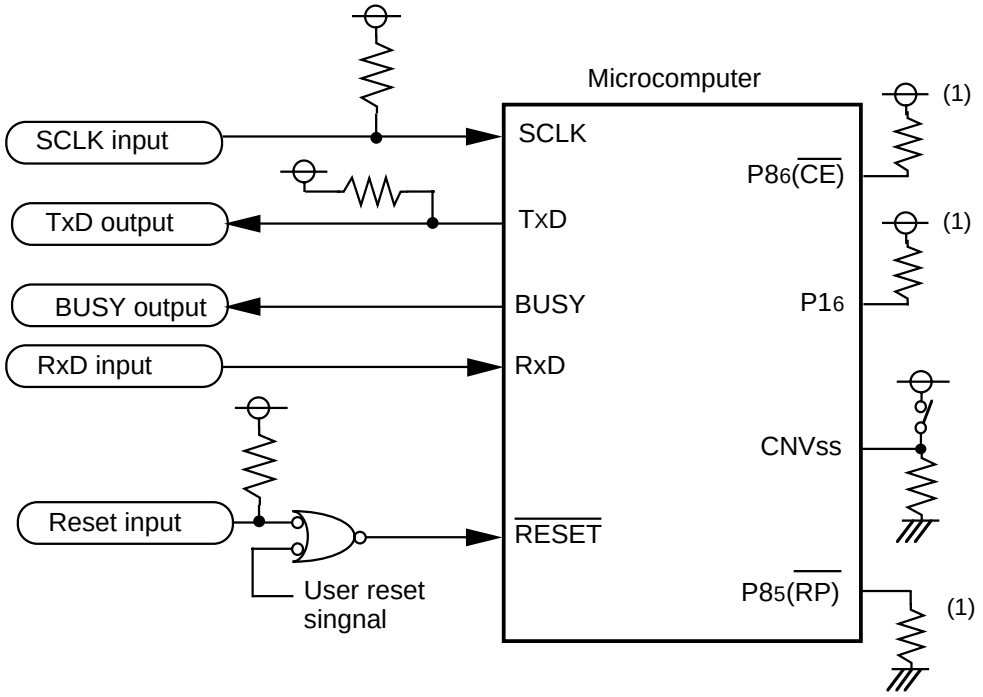
NOTE:
1. Set following either or both in serial I/O mode while the $\overline{\text{RESET}}$ pin is held "L".
   · Connect the $\overline{\text{CE}}$ pin to V_{CC}.
   · Connect the $\overline{\text{RP}}$ pin to V_{SS} and the $P1_6$ pin to V_{CC}.

Figure 17.9.2. Pin Connections for Serial I/O Mode (2)

Downloaded from Arrow.com.

17.9.2 Example of Circuit Application in Standard Serial I/O Mode

Figure 17.9.2.1 shows an example of a circuit application in standard serial I/O mode 1 and Figure 17.9.2.2 shows an example of a circuit application in standard serial I/O mode 2. Refer to the user's manual for a serial writer to handle pins controlled by the serial writer.

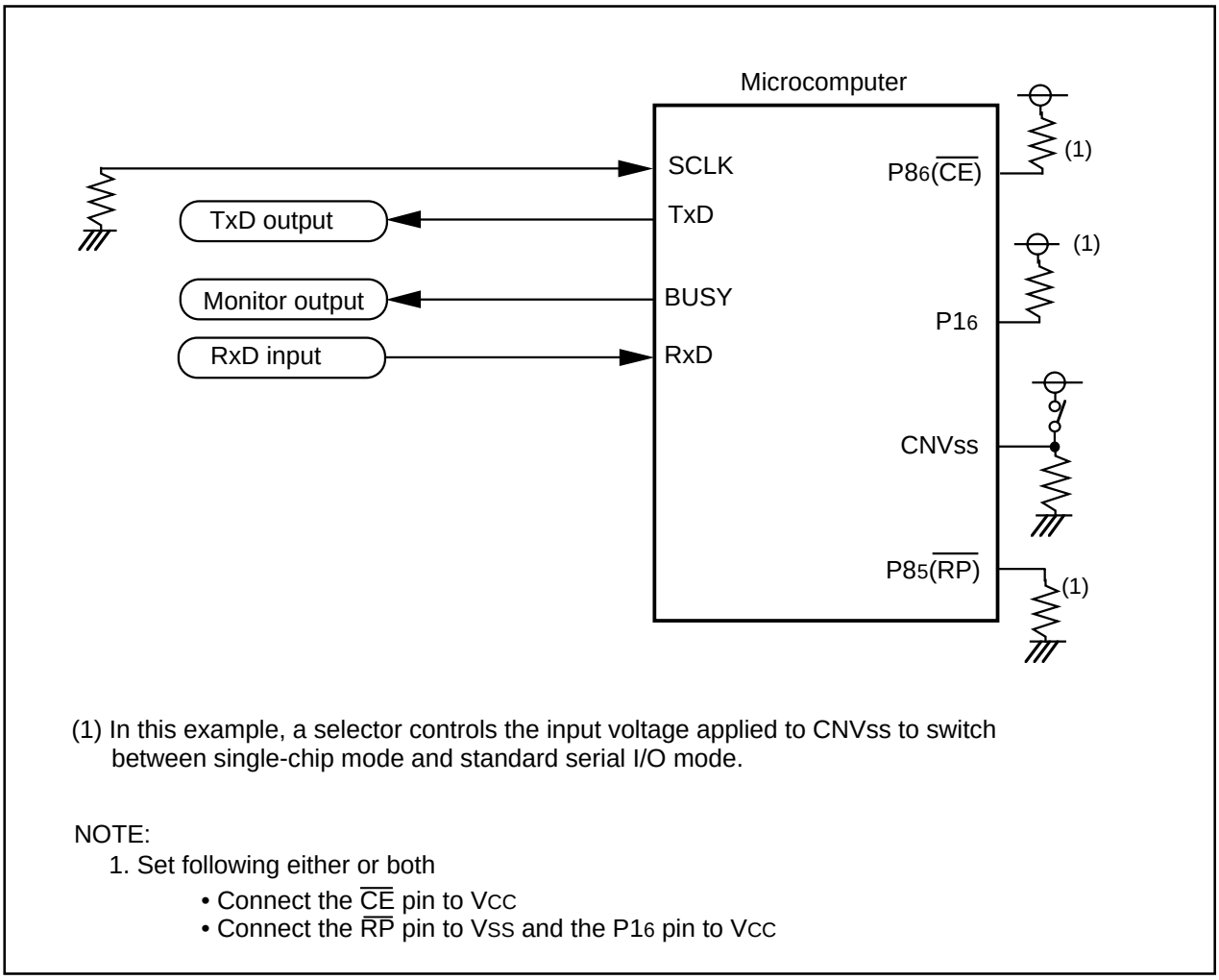


(1) Controlling pins and external circuits vary with the serial programmer. For more information, refer to the user's manual included with the serial programmer.
(2) In this example, a selector controls the input voltage applied to CNVss to switch between single-chip mode and standard serial I/O mode.
(3) In standard serial input/output mode 1, if the user reset signal becomes “L” while the microcomputer is communicating with the serial programmer, break the connection between the user reset signal and the RESET pin using a jumper switch.

NOTE:
1. Set following either or both
   - Connect the CE pin to VCC
   - Connect the RP pin to VSS and the P16 pin to VCC

Figure 17.9.2.1. Circuit Application in Standard Serial I/O Mode 1

Downloaded from Arrow.com.



(1) In this example, a selector controls the input voltage applied to CNVss to switch between single-chip mode and standard serial I/O mode.

NOTE:
1. Set following either or both
   - Connect the $\overline{CE}$ pin to V_{CC}
   - Connect the $\overline{RP}$ pin to V_{SS} and the $P1_6$ pin to V_{CC}

Figure 17.9.2.2. Circuit Application in Standard Serial I/o Mode 2

Downloaded from Arrow.com.

17.10 Parallel I/O Mode

In parallel input/output mode, the user ROM can be rewritten using a parallel programmer which is applicable for the M16C/26A group. For more information about the parallel programmer, contact your parallel programmer manufacturer. For details on how to use the parallel programmer, refer to the user's manual of the parallel programmer.

17.10.1 ROM Code Protect Function

The ROM code protect function prevents the flash memory from being read or rewritten. (Refer to **17.3 Function to Prevent Flash Memory from Rewriting**.)

Downloaded from Arrow.com.

18. Electrical Characteristics

Please contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor for electrical characteristics of V-ver.

18.1. M16C/26A, M16C/26B (Normal version)

Table 18.1. Absolute Maximum Ratings

Symbol	Parameter			Condition	Value	Unit
V_{CC}	Supply Voltage			V_{CC} = AV_{CC}	-0.3 to 6.5	V
AV_{CC}	Analog Supply Voltage			V_{CC} = AV_{CC}	-0.3 to 6.5	V
V_I	Input Voltage	$P1_5$ to $P1_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P10_0$ to $P10_7$, X_{IN}, V_{REF}, $\overline{RESET}$, CNV_{SS}			-0.3 to V_{CC}+0.3	V
V_O	Output Voltage	$P1_5$ to $P1_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P10_0$ to $P10_7$, X_{OUT}			-0.3 to V_{CC}+0.3	V
Pd	Power Dissipation			-40 ≤ Topr ≤ 85° C	300	mW
Topr	Operating Ambient Temperature	during CPU operation			-20 to 85 / -40 to 85(1)	° C
		during flash memory program and erase operation	Program Space (Block 0 to Block 3)		0 to 60	° C
			Data Space (Block A, Block B)		0 to 60 / -20 to 85 / -40 to 85(1)	° C
Tstg	Storage Temperature				-65 to 150	° C

NOTE:

1. Refer to **Tables 1.7** and **1.8**.

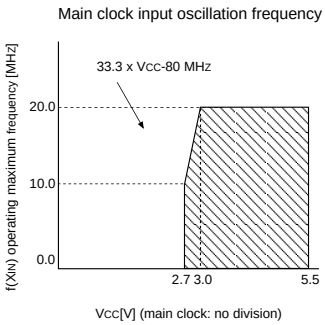
Downloaded from Arrow.com.

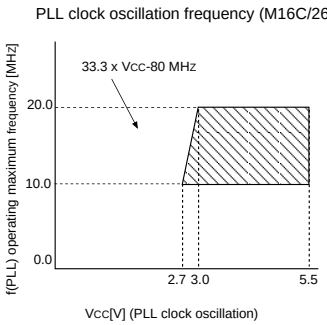
Table 18.2. Recommended Operating Conditions (1)

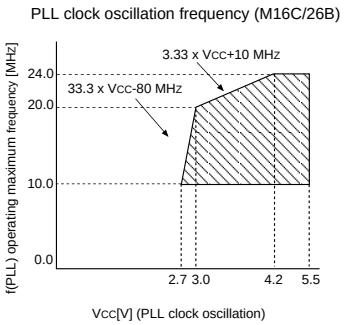
Symbol	Parameter		Standard			Unit
			Min.	Typ.	Max.	
V_{CC}	Supply Voltage		2.7		5.5	V
AV_{CC}	Analog Supply Voltage			V_{CC}		V
V_{SS}	Supply Voltage			0		V
AV_{SS}	Analog Supply Voltage			0		V
V_{IH}	Input High ("H") Voltage	$P1_5$ to $P1_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P10_0$ to $P10_7$	0.7 V_{CC}		V_{CC}	V
		XIN, $\overline{RESET}$, CNVSS	0.8 V_{CC}		V_{CC}	V
V_{IL}	Input Low ("L") Voltage	$P1_5$ to $P1_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P10_0$ to $P10_7$	0		0.3 V_{CC}	V
		XIN, $\overline{RESET}$, CNVSS	0		0.2 V_{CC}	V
$I_{OH(peak)}$	Peak Output High ("H") Current	$P1_5$ to $P1_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P10_0$ to $P10_7$			-10.0	mA
$I_{OH(avg)}$	Average Output High ("H") Current	$P1_5$ to $P1_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P10_0$ to $P10_7$			-5.0	mA
$I_{OL(peak)}$	Peak Output Low ("L") Current	$P1_5$ to $P1_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P10_0$ to $P10_7$			10.0	mA
$I_{OL(avg)}$	Average Output Low ("L") Current	$P1_5$ to $P1_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P10_0$ to $P10_7$			5.0	mA
$f(X_{IN})$	Main Clock Input Oscillation Frequency(4)	V_{CC} = 3.0 to 5.5 V	0		20	MHz
		V_{CC} = 2.7 to 3.0 V	0		33 X V_{CC}-80	MHz
$f(X_{CIN})$	Sub Clock Oscillation Frequency			32.768	50	kHz
f_1(ROC)	On-chip Oscillator Frequency 1		0.5	1	2	MHz
f2(ROC)	On-chip Oscillator Frequency 2		1	2	4	MHz
f3(ROC)	On-chip Oscillator Frequency 3		8	16	26	MHz
f(PLL)	PLL Clock Oscillation Frequency(4)	V_{CC} = 4.2 to 5.5 V (M16C/26B)	10		24	MHz
		V_{CC} = 3.0 to 4.2 V (M16C/26B)	10		3.33 X V_{CC}+10	MHz
		V_{CC} = 3.0 to 5.5 V (M16C/26A)	10		20	MHz
		V_{CC} = 2.7 to 3.0 V	10		33 X V_{CC}-80	MHz
f(BCLK)	CPU Operation Clock Frequency	M16C/26A	0		20	MHz
		M16C/26B	0		24	MHz
t_{SU}(PLL)	Wait Time to Stabilize PLL Frequency Synthesizer	V_{CC}=5.0V			20	ms
		V_{CC}=3.0V			50	ms

NOTES:

1. Referenced to V_{CC} = 2.7 to 5.5 V at Topr = -20 to 85 ° C / -40 to 85 ° C unless otherwise specified.
2. The mean output current is the mean value within 100 ms.
3. The total $I_{OL(peak)}$ for all ports must be 80 mA or less. The total $I_{OH(peak)}$ for all ports must be -80 mA or less.
4. Relationship among main clock oscillation frequency, PLL clock oscillation frequency and supply voltage.







Downloaded from Arrow.com.

Table 18.3. A /D Conversion Characteristics(1)

Symbol	Parameter		Measurement Condition	Standard			Unit
				Min.	Typ.	Max.	
-	Resolution		$V_{REF} = V_{CC}$			10	Bits
INL	Integral Nonlinearity Error	10 bit	$V_{REF} = V_{CC} = 5V$			±3	LSB
			$V_{REF} = V_{CC} = 3.3$ V			±5	LSB
		8 bit	$V_{REF} = V_{CC} = 3.3$ V, 5 V			±2	LSB
-	Absolute Accuracy	10 bit	$V_{REF} = V_{CC} = 5$ V			±3	LSB
			$V_{REF} = V_{CC} = 3.3$ V			±5	LSB
		8 bit	$V_{REF} = V_{CC} = 3.3$ V, 5 V			±2	LSB
DNL	Differential Nonlinearity Error					±1	LSB
-	Offset Error					±3	LSB
-	Gain Error					±3	LSB
R_{LADDER}	Resistor Ladder		$V_{REF} = V_{CC}$	10		40	kΩ
t_{CONV}	10-bit Conversion Time Sample & Hold Function Available		$V_{REF} = V_{CC} = 5$ V, ϕAD = 10 MHz	3.3			µs
t_{CONV}	8-bit Conversion Time Sample & Hold Function Available		$V_{REF} = V_{CC} = 5$ V, ϕAD = 10 MHz	2.8			µs
V_{REF}	Reference Voltage			2.0		V_{CC}	V
V_{IA}	Analog Input Voltage			0		V_{REF}	V

NOTES:
1. Referenced to V_{CC}=AV_{CC}=V_{REF}= 3.3 to 5.5V, V_{SS}=AV_{SS}=0V at Topr = -20 to 85 ° C / -40 to 85° C unless otherwise specified.
2. Keep ϕAD frequency at 10 MHz or less (12 MHz or less in M16C/26B). Additionally, divide the f_{AD} if V_{CC} is less than 4.2V, and make ϕAD frequency equal to or lower than $f_{AD}/2$.
3. When sample & hold function is disabled, keep ϕAD frequency at 250 kHz or more in addition to the limitation in Note 2. When sample & hold function is enabled, keep ϕAD frequency at 1 MHz or more in addition to the limitation in Note 2.
4. When sample & hold function is enabled, sampling time is 3/ ϕAD frequency. When sample & hold function is disabled, sampling time is 2/ ϕAD frequency.

Downloaded from Arrow.com.

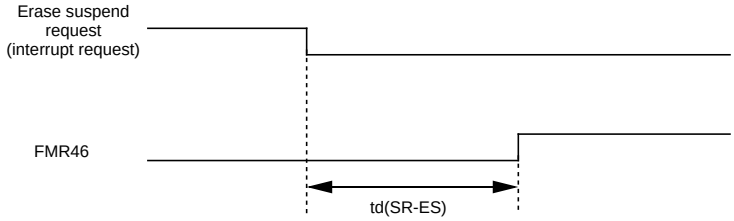
Table 18.4. Flash Memory Version Electrical Characteristic (1): Program Space and Data Space for U3 and U5, Program Space for U7 and U9

Symbol	Parameter		Standard			Unit
			Min.	Typ.(2)	Max.	
-	Program and Erase Endurance(3)		100/1000(4, 11)			cycles
-	Word Program Time (Vcc=5.0V, Topr=25° C)			75	600	µs
-	Block Erase Time (Vcc=5.0V, Topr=25° C)	2-Kbyte Block		0.2	9	s
		8-Kbyte Block		0.4	9	s
		16-Kbyte Block		0.7	9	s
		32-Kbyte Block		1.2	9	s
td(SR-ES)	Duration between Suspend Request and Erase Suspend				8	ms
tPS	Wait Time to Stabilize Flash Memory Circuit				15	µs
-	Data Hold Time (5)		20			years

Table 18.5. Flash Memory Version Electrical Characteristics (6): Data Space for U7 and U9 (7)

Symbol	Parameter	Standard			Unit
		Min.	Typ.(2)	Max.	
-	Program and Erase Endurance(3, 8, 9)	10000(4, 10)			cycles
-	Word Program Time (Vcc=5.0V, Topr=25° C)		100		µs
-	Block Erase Time (Vcc=5.0V, Topr=25° C) (2-Kbyte block)		0.3		s
td(SR-ES)	Duration between Suspend Request and Erase Suspend			8	ms
tPS	Wait Time to Stabilize Flash Memory Circuit			15	µs
-	Data Hold Time (5)	20			years

NOTES:
1. Referenced to Vcc = 2.7 to 5.5 V at Topr = 0 to 60° C (program space), -40 to 85° C (data space), unless otherwise specified.
2. Vcc = 5.0 V; Topr = 25° C
3. Program and erase endurance is defined as number of program-erase cycles per block.
   If program and erase endurance is *n* cycle (*n* = 100, 1000, 10000), each block can be erased and programmed *n* cycles.
   For example, if a 2-Kbyte block A is erased after programming one-word data to each address 1,024 times, this counts as one program and erase endurance. Data cannot be programmed to the same address more than once without erasing the block. (rewrite prohibited).
4. Number of E/W cycles for which operation is guranteed (1 to minimum value are guaranteed).
5. Topr = 55° C
6. Referenced to Vcc = 2.7 to 5.5 V at Topr = -40 to 85° C (U7) / -20 to 85° C (U9) unless otherwise specified.
7. Table 18.5 applies for data space in U7 and U9 when program and erase endurance is more than 1,000 cycles. Otherwise, use Table 18.4.
8. To reduce the number of program and erase endurance when working with systems requiring numerous rewrites, write to unused word addresses within the block instead of rewrite. Erase block only after all possible addresses are used. For example, an 8-word program can be written 128 times maximum before erase becomes necessary. Maintaining an equal number of times erasure between block A and block B will also improve efficiency. It is recommended to track the total number of erasure performed per block and to limit the number of erasure.
9. Execute the clear status register command and block erase command at least 3 times until an erase error is not generated when an erase error is generated.
10. When executing more than 100 times rewrites, set one wait state per block access by setting the FMR17 bit in the FMR1 register 1 to "1" (wait state). When accessing to all other blocks and internal RAM, wait state can be set by the PM17 bit, regardless of the FMR17 bit setting value.
11. The program and erase endurance is 100 cycles for program space and data space in U3 and U5; 1,000 cycles for program space in U7 and U9.
12. Customers desiring E/W failure rate information should contact their Renesas technical support representative.



Downloaded from Arrow.com.

Table 18.6. Voltage Detection Circuit Electrical Characteristics (1, 3)

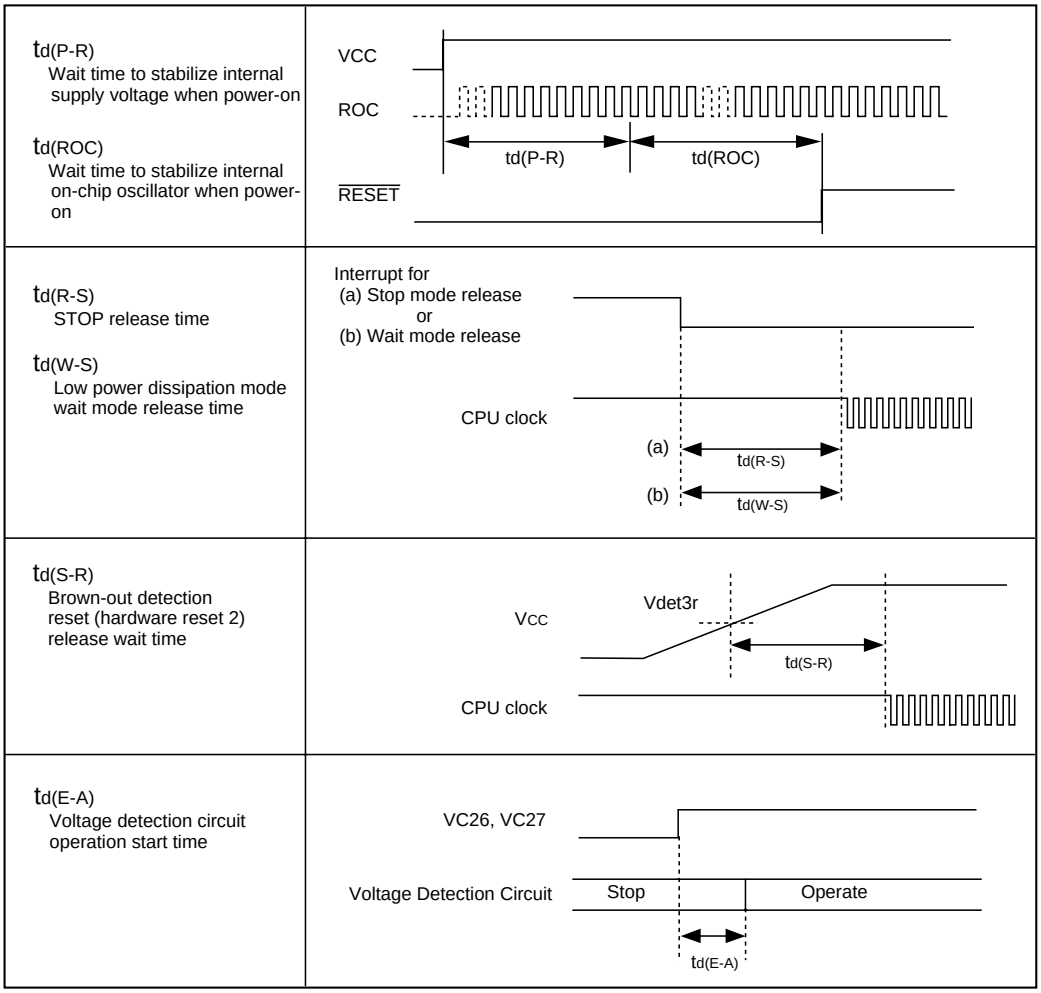
Symbol	Parameter	Measurement Condition	Standard			Unit
			Min.	Typ.	Max.	
Vdet4	Low Voltage Detection Voltage(1)	Vcc=0.8 to 5.5V	3.2	3.8	4.45	V
Vdet3	Reset Level Detection Voltage(1)		2.3	2.8	3.4	V
Vdet3s	Low Voltage Reset Hold Voltage(2)				1.7	V
Vdet3r	Low Voltage Reset Release Voltage		2.35	2.9	3.5	V

NOTES:
1. Vdet4 >Vdet3
2. Vdet3s is the minmum voltage to maintain "hardware reset 2".
3. The voltage detection circuit is designed to use when Vcc is set to 5V.
4. If the supply power voltage is greater than the reset level detection voltage when the reset level detection voltage is less than 2.7V, the operation at f(BCLK) ≤ 10MHz is guranteed. However, A/D conversion, serial I/O, flash memory program and erase are excluded.

Table 18.7. Power Supply Circuit Timing Characteristics

Symbol	Parameter	Measurement Condition	Standard			Unit
			Min.	Typ.	Max.	
td(P-R)	Wait Time to Stabilize Internal Supply Voltage when Power-on	Vcc = 2.7 to 5.5 V			2	ms
td(ROC)	Wait Time to Stabilize Internal On-chip Oscillator when Power-on				40	µs
td(R-S)	STOP Release Time				150	µs
td(W-S)	Low Power Dissipation Mode Wait Mode Release Time				150	µs
td(S-R)	Hardware Reset 2 Release Wait Time	Vcc = Vdet3r to 5.5 V		6(1)	20	ms
td(E-A)	Voltage Detection Circuit Operation Start Time	Vcc = 2.7 to 5.5 V			20	µs

NOTES:
1. When Vcc=5V



Downloaded from Arrow.com.

V_{CC} = 5V

Table 18.8. Electrical Characteristics (1)

Symbol	Parameter				Condition	Standard			Unit
						Min.	Typ.	Max.	
V_{OH}	Output High ("H") Voltage	$P1_5$ to $P1_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P10_0$ to $P10_7$			I_{OH} = -5 mA	V_{CC}-2.0		V_{CC}	V
V_{OH}	Output High ("H") Voltage	$P1_5$ to $P1_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P10_0$ to $P10_7$			I_{OH} =-200 µA	V_{CC}-0.3		V_{CC}	V
V_{OH}	Output High ("H") Voltage		X_{OUT}	High Power	I_{OH} = -1mA	V_{CC}-2.0		V_{CC}	V
				Low Power	I_{OH} = -0.5mA	V_{CC}-2.0		V_{CC}	
	Output High ("H") Voltage		X_{COUT}	High Power	No load applied		2.5		V
				Low Power	No load applied		1.6		
V_{OL}	Output Low ("L") Voltage	$P1_5$ to $P1_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P10_0$ to $P10_7$			I_{OL} = 5 mA			2.0	V
V_{OL}	Output Low ("L") Voltage	$P1_5$ to $P1_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P10_0$ to $P10_7$			I_{OL} = 200 µA			0.45	V
V_{OL}	Output Low ("L") Voltage		X_{OUT}	High Power	I_{OL} = 1 mA			2.0	V
				Low Power	I_{OL} = 0.5 mA			2.0	
	Output Low ("L") Voltage		X_{COUT}	High Power	No load applied		0		V
				Low Power	No load applied		0		
V_{T+}-V_{T-}	Hysteresis	$TA0_{IN}$-$TA4_{IN}$, $TB0_{IN}$-$TB2_{IN}$, INT_0-INT_5, NMI, AD_{TRG}, CTS_0-CTS_2, CLK_0-CLK_2, $TA2_{OUT}$-$TA4_{OUT}$, KI_0-KI_3, RXD_0-RXD_2				0.2		1.0	V
V_{T+}-V_{T-}	Hysteresis	RESET				0.2		2.5	V
V_{T+}-V_{T-}	Hysteresis	X_{IN}				0.2		0.8	V
I_{IH}	Input High ("H") Current	$P1_5$ to $P1_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P10_0$ to $P10_7$, X_{IN}, RESET, CNV_{SS}			V_I=5V			5.0	µA
I_{IL}	Input Low ("L") Current	$P1_5$ to $P1_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P10_0$ to $P10_7$, X_{IN}, RESET, CNV_{SS}			V_I=0V			-5.0	µA
R_{PULLUP}	Pull-up Resistance	$P1_5$ to $P1_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P10_0$ to $P10_7$			V_I=0V	30	50	170	kΩ
Rf_{XIN}	Feedback Resistance		X_{IN}				1.5		MΩ
Rf_{XCIN}	Feedback Resistance		X_{CIN}				15		MΩ
V_{RAM}	RAM Standby Voltage				In stop mode	2.0			V

NOTE:

1. Referenced to V_{CC}=4.2 to 5.5V, V_{SS}=0V at Topr=-20 to 85 ° C / -40 to 85 ° C, f(BCLK)=20MHz unless otherwise specified.

Downloaded from Arrow.com.

V_{CC} = 5V

Table 18.9. Electrical Characteristics (2) [1]

Symbol	Parameter		Measurement Condition		Standard			Unit
					Min.	Typ.	Max.	
I_{CC}	Power Supply Current (V_{CC} = 4.0 to 5.5 V)	Output pins are left open and other pins are connected to V_{SS}	Mask ROM	f(BCLK) = 20 MHz, main clock, no division		12	17	mA
				On-chip oscillation $f_{2(ROC)}$ selected, f(BCLK) = 1 MHz		1		mA
			Flash memory	f(BCLK) = 24 MHz, PLL operates (M16C/26B)		20	23	mA
				f(BCLK) = 20 MHz, main clock, no division		16	19	mA
				On-chip oscillator operates, $f_{2(ROC)}$ selected, f(BCLK) = 1 MHz		1		mA
			Flash memory program	f(BCLK) = 10 MHz, Vcc = 5.0 V		11		mA
			Flash memory erase	f(BCLK) = 10 MHz, Vcc = 5.0 V		12		mA
			Mask ROM	f(BCLK) = 32 kHz, In low-power consumption mode, Program running on ROM[3]		25		µA
				On-chip oscillator operates, $f_{2(ROC)}$ selected, f(BCLK) = 1 MHz, In wait mode		30		µA
			Flash memory	f(BCLK) = 32 kHz, In low-power consumption mode, Program running on RAM[3]		25		µA
				f(BCLK) = 32 kHz, In low-power consumption mode, Program running on flash memory[3]		450		µA
				On-chip oscillator operates, $f_{2(ROC)}$ selected, f(BCLK) = 1 MHz, In wait mode		50		µA
			Mask ROM, Flash memory	f(BCLK) = 32 kHz, In wait mode[2], Oscillation capacity HIGH		10		µA
				f(BCLK) = 32 kHz, In wait mode[2], Oscillation capacity LOW		3		µA
				In stop mode, Topr = 25° C		0.8	3	µA
Idet4	Low voltage detection dissipation current[4]					0.7	4	µA
Idet3	Reset level detection dissipation current[4]					1.2	8	µA

NOTES:

1. Referenced to V_{CC} = 4.2 to 5.5 V, V_{SS} = 0 V at Topr = -20 to 85° C / -40 to 85 ° C, f(BCLK) = 20 MHz unless otherwise specified.
2. With one timer operates, using f_{C32}.
3. This indicates the memory in which the program to be executed exists.
4. Idet is dissipation current when the following bit is set to "1" (detection circuit enabled).
   Idet4: VC27 bit in the VCR2 register
   Idet3: VC26 bit in the VCR2 register

Downloaded from Arrow.com.

V_{CC} = 5V

Timing Requirements

(V_{CC} = 5V, V_{SS} = 0V, at Topr = – 20 to 85°C / – 40 to 85°C unless otherwise specified)

Table 18.10. External Clock Input (X_{IN} input)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tc	External Clock Input Cycle Time	50		ns
$t_{w(H)}$	External Clock Input High ("H") Width	20		ns
$t_{w(L)}$	External Clock Input Low ("L") Width	20		ns
tr	External Clock Rise Time		9	ns
tf	External Clock Fall Time		9	ns

Downloaded from Arrow.com.

V_{CC} = 5V

Timing Requirements

(V_{CC} = 5V, V_{SS} = 0V, at Topr = – 20 to 85°C / – 40 to 85°C unless otherwise specified)

Table 18.11. Timer A Input (Counter Input in Event Counter Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TA_{iIN} input cycle time	100		ns
$t_{w(TAH)}$	TA_{iIN} input HIGH pulse width	40		ns
$t_{w(TAL)}$	TA_{iIN} input LOW pulse width	40		ns

Table 18.12. Timer A Input (Gating Input in Timer Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TA_{iIN} input cycle time	400		ns
$t_{w(TAH)}$	TA_{iIN} input HIGH pulse width	200		ns
$t_{w(TAL)}$	TA_{iIN} input LOW pulse width	200		ns

Table 18.13. Timer A Input (External Trigger Input in One-shot Timer Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TA_{iIN} input cycle time	200		ns
$t_{w(TAH)}$	TA_{iIN} input HIGH pulse width	100		ns
$t_{w(TAL)}$	TA_{iIN} input LOW pulse width	100		ns

Table 18.14. Timer A Input (External Trigger Input in Pulse Width Modulation Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{w(TAH)}$	TA_{iIN} input HIGH pulse width	100		ns
$t_{w(TAL)}$	TA_{iIN} input LOW pulse width	100		ns

Table 18.15. Timer A Input (Counter Increment/decrement Input in Event Counter Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(UP)}$	TA_{iOUT} input cycle time	2000		ns
$t_{w(UPH)}$	TA_{iOUT} input HIGH pulse width	1000		ns
$t_{w(UPL)}$	TA_{iOUT} input LOW pulse width	1000		ns
$t_{su(UP-TIN)}$	TA_{iOUT} input setup time	400		ns
$t_{h(TIN-UP)}$	TA_{iOUT} input hold time	400		ns

Table 18.16. Timer A Input (Two-phase Pulse Input in Event Counter Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TA_{iIN} input cycle time	800		ns
$t_{su(TAIN-TAOUT)}$	TA_{iOUT} input setup time	200		ns
$t_{su(TAOUT-TAIN)}$	TA_{iIN} input setup time	200		ns

Downloaded from Arrow.com.

V_{CC} = 5V

Timing Requirements

(V_{CC} = 5V, V_{SS} = 0V, at Topr = – 20 to 85°C / – 40 to 85°C unless otherwise specified)

Table 18.17. Timer B Input (Counter Input in Event Counter Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIN input cycle time (counted on one edge)	100		ns
$t_{w(TBH)}$	TBiIN input HIGH pulse width (counted on one edge)	40		ns
$t_{w(TBL)}$	TBiIN input LOW pulse width (counted on one edge)	40		ns
$t_{c(TB)}$	TBiIN input cycle time (counted on both edges)	200		ns
$t_{w(TBH)}$	TBiIN input HIGH pulse width (counted on both edges)	80		ns
$t_{w(TBL)}$	TBiIN input LOW pulse width (counted on both edges)	80		ns

Table 18.18. Timer B Input (Pulse Period Measurement Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIN input cycle time	400		ns
$t_{w(TBH)}$	TBiIN input HIGH pulse width	200		ns
$t_{w(TBL)}$	TBiIN input LOW pulse width	200		ns

Table 18.19. Timer B Input (Pulse Width Measurement Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIN input cycle time	400		ns
$t_{w(TBH)}$	TBiIN input HIGH pulse width	200		ns
$t_{w(TBL)}$	TBiIN input LOW pulse width	200		ns

Table 18.20. A/D Trigger Input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(AD)}$	$\overline{ADTRG}$ input cycle time (trigger able minimum)	1000		ns
$t_{w(ADL)}$	$\overline{ADTRG}$ input LOW pulse width	125		ns

Table 18.21. Serial I/O

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(CK)}$	CLKi input cycle time	200		ns
$t_{w(CKH)}$	CLKi input HIGH pulse width	100		ns
$t_{w(CKL)}$	CLKi input LOW pulse width	100		ns
$t_{d(C-Q)}$	TxDi output delay time		80	ns
$t_{h(C-Q)}$	TxDi hold time	0		ns
$t_{su(D-C)}$	RxDi input setup time	70		ns
$t_{h(C-D)}$	RxDi input hold time	90		ns

Table 18.22. External Interrupt $\overline{INTi}$ Input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{w(INH)}$	$\overline{INTi}$ input HIGH pulse width	250		ns
$t_{w(INL)}$	$\overline{INTi}$ input LOW pulse width	250		ns

Downloaded from Arrow.com.

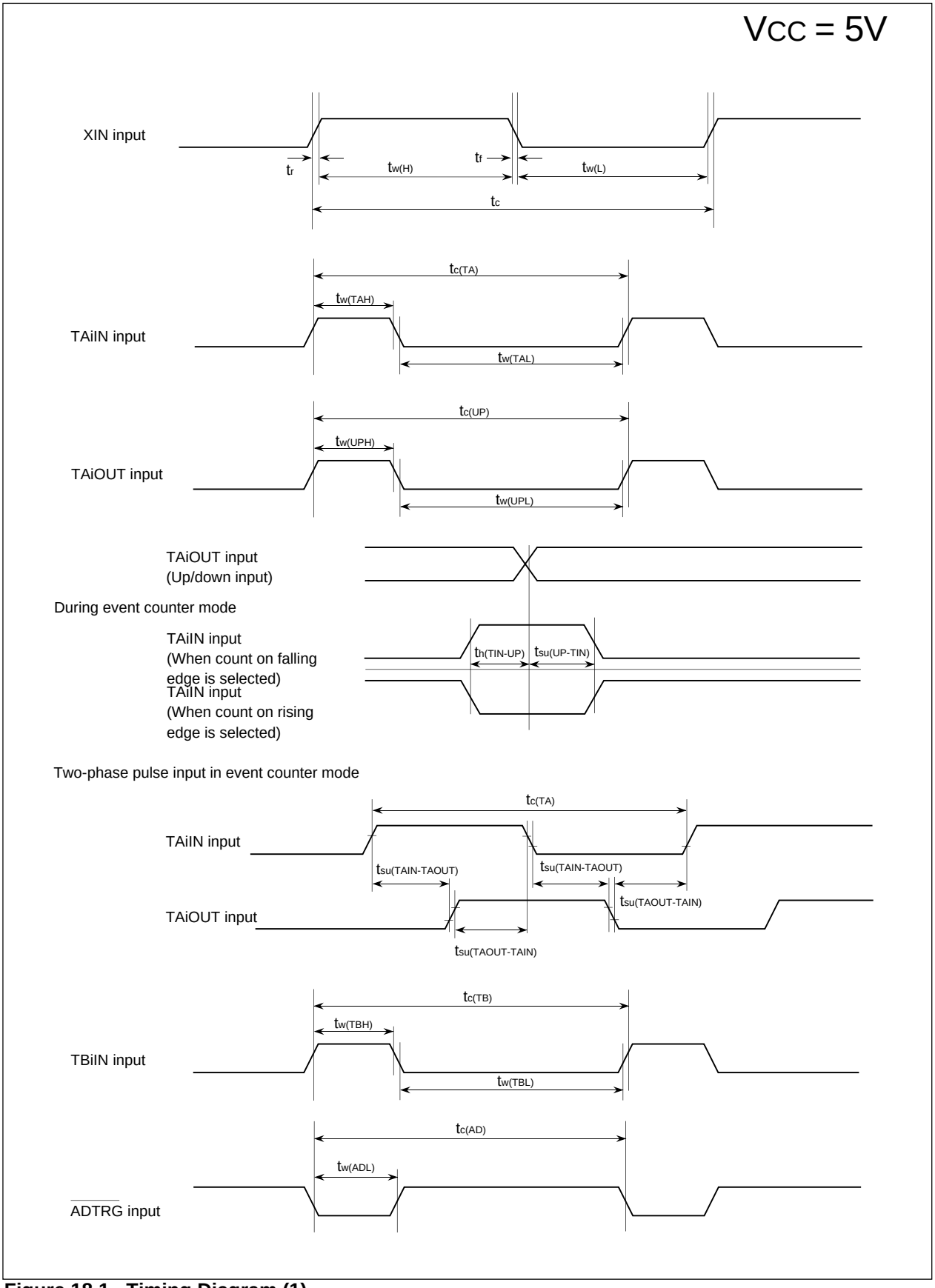


Figure 18.1. Timing Diagram (1)

Downloaded from Arrow.com.

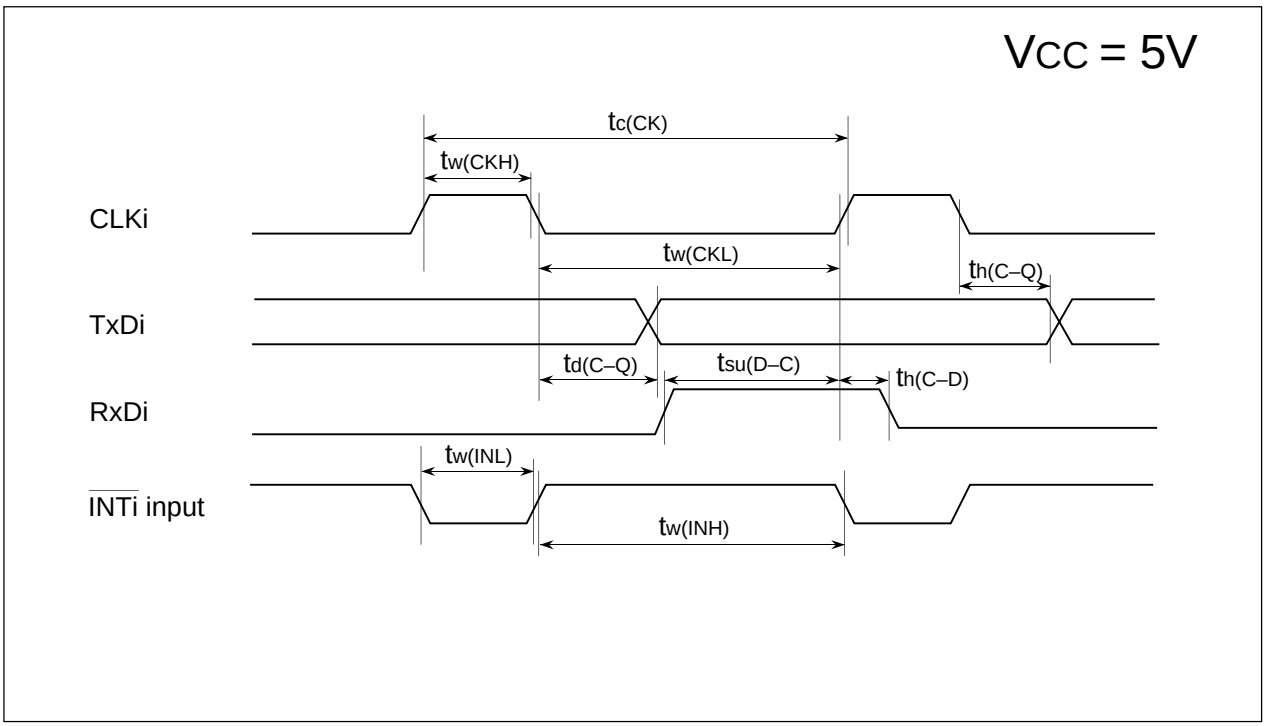


Figure 18.2. Timing Diagram (2)

Downloaded from Arrow.com.

V_{CC} = 3V

Table 18.23. Electrical Characteristics (1)

Symbol	Parameter			Condition	Standard			Unit
					Min.	Typ.	Max.	
V_{OH}	Output High ("H") Voltage	$P1_5$ to $P1_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P10_0$ to $P10_7$		I_{OH}=-1mA	V_{CC}-0.5		V_{CC}	V
V_{OH}	Output High ("H") Voltage	X_{OUT}	High Power	I_{OH}=-0.1mA	V_{CC}-0.5		V_{CC}	V
			Low Power	I_{OH}=-50µA	V_{CC}-0.5		V_{CC}	
	Output High ("H") Voltage	X_{COUT}	High Power	No load applied		2.5		V
			Low Power	No load applied		1.6		
V_{OL}	Output Low ("L") Voltage	$P1_5$ to $P1_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P10_0$ to $P10_7$		I_{OL}=1mA			0.5	V
V_{OL}	Output Low ("L") Voltage	X_{OUT}	High Power	I_{OL}=0.1mA			0.5	V
			Low Power	I_{OL}=50µA			0.5	
	Output Low ("L") Voltage	X_{COUT}	High Power	No load applied		0		V
			Low Power	No load applied		0		
V_{T+}-V_{T-}	Hysteresis	$TA0_{IN}$-$TA4_{IN}$, $TB0_{IN}$-$TB2_{IN}$, $\overline{INT_0}$-$\overline{INT_5}$, $\overline{NMI}$, $\overline{ADTRG}$, $\overline{CTS_0}$-$\overline{CTS_2}$, CLK_0-CLK_2, $TA2_{OUT}$-$TA4_{OUT}$, $\overline{KI_0}$-$\overline{KI_3}$, RXD_0-RXD_2					0.8	V
V_{T+}-V_{T-}	Hysteresis	RESET					1.8	V
V_{T+}-V_{T-}	Hysteresis	X_{IN}					0.8	V
I_{IH}	Input High ("H") Current	$P1_5$ to $P1_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P10_0$ to $P10_7$ X_{IN}, $\overline{RESET}$, CNV_{SS}		V_I=3V			4.0	µA
I_{IL}	Input Low ("L") Current	$P1_5$ to $P1_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P10_0$ to $P10_7$ X_{IN}, $\overline{RESET}$, CNV_{SS}		V_I=0V			-4.0	µA
R_{PULLUP}	Pull-up Resistance	$P1_5$ to $P1_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P10_0$ to $P10_7$		V_I=0V	50	100	500	kΩ
Rf_{XIN}	Feedback Resistance	X_{IN}				3.0		MΩ
Rf_{XCIN}	Feedback Resistance	X_{CIN}				25		MΩ
V_{RAM}	RAM Standby Voltage			In stop mode	2.0			V

NOTE:

1. Referenced to V_{CC} = 2.7 to 3.6 V, V_{SS} = 0 V at Topr = -20 to 85 ° C / -40 to 85 ° C, f(BCLK) = 10 MHz unless otherwise specified.

Downloaded from Arrow.com.

Vcc = 3V

Table 18.24. Electrical Characteristics (2) [1]

Symbol	Parameter	Measurement Condition			Standard			Unit
					Min.	Typ.	Max.	
Icc	Power Supply Current (Vcc = 2.7 to 3.6V)	Output pins are left open and other pins are connected to Vss	Mask ROM	f(BCLK) = 10 MHz, Main clock, no division		7	10	mA
				On-chip oscillator operates, $f_{2(ROC)}$ selected, f(BCLK) = 1 MHz		1		mA
			Flash memory	f(BCLK) = 10 MHz, Main clock, no division		7	12	mA
				On-chip oscillator operates, $f_{2(ROC)}$ selected, f(BCLK) = 1 MHz		1		mA
			Flash memory program	f(BCLK) = 10 MHz, Vcc = 3.0 V		10		mA
			Flash memory erase	f(BCLK) = 10 MHz, Vcc= 3.0 V		11		mA
			Mask ROM	f(BCLK) = 32 kHz, In low-power consumption mode, Program running on ROM[3]		25		µA
				On-chip oscillator operates, $f_{2(ROC)}$ selected, f(BCLK) = 1 MHz, In wait mode		25		µA
			Flash memory	f(BCLK) = 32 kHz, In low-power consumption mode, Program running on RAM[3]		25		µA
				f(BCLK) = 32 kHz, In low-power consumption mode, Program running on flash memory[3]		450		µA
				On-chip oscillator operates, $f_{2(ROC)}$ selected, f(BCLK) = 1 MHz, In wait mode		45		µA
			Mask ROM, Flash memory	f(BCLK) = 32 kHz, In wait mode[2], Oscillation capacity HIGH		10		µA
				f(BCLK) = 32 kHz, In wait mode[2], Oscillation capacity LOW		3		µA
				While clock stops, Topr = 25° C		0.7	3	µA
Idet4	Low voltage detection dissipation current[4]					0.6	4	µA
Idet3	Reset level detection dissipation current[4]					1.0	5	µA

NOTES:

1. Referenced to Vcc = 2.7 to 3.6 V, Vss = 0 V at Topr = -20 to 85 ° C / -40 to 85 ° C, f(BCLK) = 10 MHz unless otherwise specified.
2. With one timer operates, using f_{C32}.
3. This indicates the memory in which the program to be executed exists.
4. Idet is dissipation current when the following bit is set to 1 (detection circuit enabled).
   Idet4: the VC27 bit of the VCR2 register
   Idet3: the VC26 bit in the VCR2 register

Downloaded from Arrow.com.

VCC = 3V

Timing Requirements

(VCC = 3V, VSS = 0V, at Topr = – 20 to 85°C / – 40 to 85°C unless otherwise specified)

Table 18.25. External Clock Input (XIN input)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tc	External Clock Input Cycle Time	100		ns
$tw_{(H)}$	External Clock Input High ("H") Width	40		ns
$tw_{(L)}$	External Clock Input Low ("L") Width	40		ns
tr	External Clock Rise Time		18	ns
tf	External Clock Fall Time		18	ns

Downloaded from Arrow.com.

VCC = 3V

Timing Requirements

(VCC = 3V, VSS = 0V, at Topr = – 20 to 85°C / – 40 to 85°C unless otherwise specified)

Table 18.26. Timer A Input (Counter Input in Event Counter Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIN input cycle time	150		ns
$t_{w(TAH)}$	TAiIN input HIGH pulse width	60		ns
$t_{w(TAL)}$	TAiIN input LOW pulse width	60		ns

Table 18.27. Timer A Input (Gating Input in Timer Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIN input cycle time	600		ns
$t_{w(TAH)}$	TAiIN input HIGH pulse width	300		ns
$t_{w(TAL)}$	TAiIN input LOW pulse width	300		ns

Table 18.28. Timer A Input (External Trigger Input in One-shot Timer Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIN input cycle time	300		ns
$t_{w(TAH)}$	TAiIN input HIGH pulse width	150		ns
$t_{w(TAL)}$	TAiIN input LOW pulse width	150		ns

Table 18.29. Timer A Input (External Trigger Input in Pulse Width Modulation Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{w(TAH)}$	TAiIN input HIGH pulse width	150		ns
$t_{w(TAL)}$	TAiIN input LOW pulse width	150		ns

Table 18.30. Timer A Input (Counter Increment/decrement Input in Event Counter Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(UP)}$	TAiOUT input cycle time	3000		ns
$t_{w(UPH)}$	TAiOUT input HIGH pulse width	1500		ns
$t_{w(UPL)}$	TAiOUT input LOW pulse width	1500		ns
$t_{su(UP-TIN)}$	TAiOUT input setup time	600		ns
$t_{h(TIN-UP)}$	TAiOUT input hold time	600		ns

Table 18.31. Timer A Input (Two-phase Pulse Input in Event Counter Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIN input cycle time	2		µs
$t_{su(TAIN-TAOUT)}$	TAiOUT input setup time	500		ns
$t_{su(TAOUT-TAIN)}$	TAiIN input setup time	500		ns

Downloaded from Arrow.com.

V_{CC} = 3V

Timing Requirements

(V_{CC} = 3V, V_{SS} = 0V, at Topr = – 20 to 85°C / – 40 to 85°C unless otherwise specified)

Table 18.32. Timer B Input (Counter Input in Event Counter Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIN input cycle time (counted on one edge)	150		ns
$t_{w(TBH)}$	TBiIN input HIGH pulse width (counted on one edge)	60		ns
$t_{w(TBL)}$	TBiIN input LOW pulse width (counted on one edge)	60		ns
$t_{c(TB)}$	TBiIN input cycle time (counted on both edges)	300		ns
$t_{w(TBH)}$	TBiIN input HIGH pulse width (counted on both edges)	120		ns
$t_{w(TBL)}$	TBiIN input LOW pulse width (counted on both edges)	120		ns

Table 18.33. Timer B Input (Pulse Period Measurement Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIN input cycle time	600		ns
$t_{w(TBH)}$	TBiIN input HIGH pulse width	300		ns
$t_{w(TBL)}$	TBiIN input LOW pulse width	300		ns

Table 18.34. Timer B Input (Pulse Width Measurement Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIN input cycle time	600		ns
$t_{w(TBH)}$	TBiIN input HIGH pulse width	300		ns
$t_{w(TBL)}$	TBiIN input LOW pulse width	300		ns

Table 18.35. A/D Trigger Input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(AD)}$	$\overline{ADTRG}$ input cycle time (trigger able minimum)	1500		ns
$t_{w(ADL)}$	$\overline{ADTRG}$ input LOW pulse width	200		ns

Table 18.36. Serial I/O

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(CK)}$	CLKi input cycle time	300		ns
$t_{w(CKH)}$	CLKi input HIGH pulse width	150		ns
$t_{w(CKL)}$	CLKi input LOW pulse width	150		ns
$t_{d(C-Q)}$	TxDi output delay time		160	ns
$t_{h(C-Q)}$	TxDi hold time	0		ns
$t_{su(D-C)}$	RxDi input setup time	100		ns
$t_{h(C-D)}$	RxDi input hold time	90		ns

Table 18.37. External Interrupt $\overline{INTi}$ Input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{w(INH)}$	$\overline{INTi}$ input HIGH pulse width	380		ns
$t_{w(INL)}$	$\overline{INTi}$ input LOW pulse width	380		ns

Downloaded from Arrow.com.

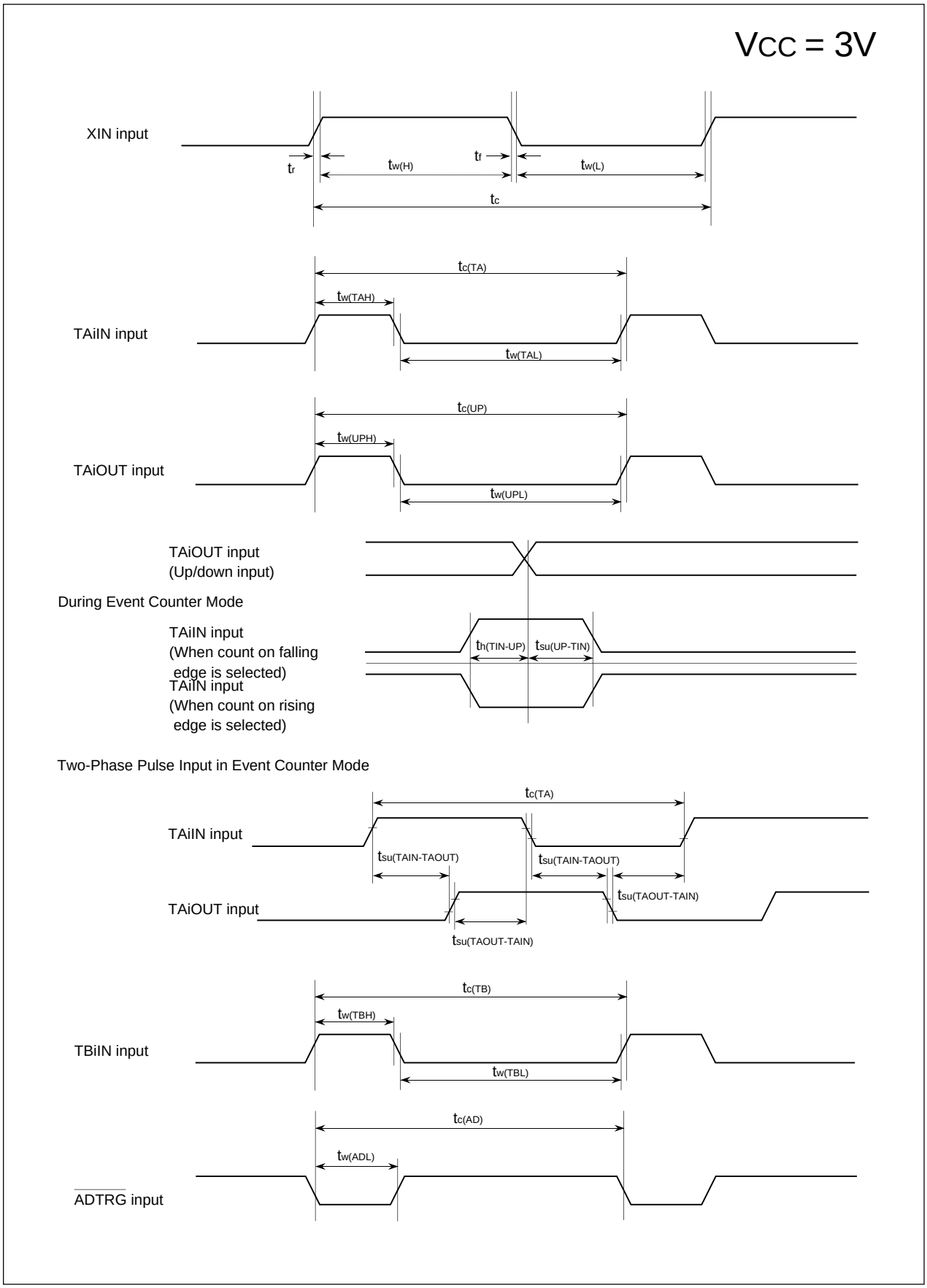


Figure 18.3. Timing Diagram (1)

Downloaded from Arrow.com.

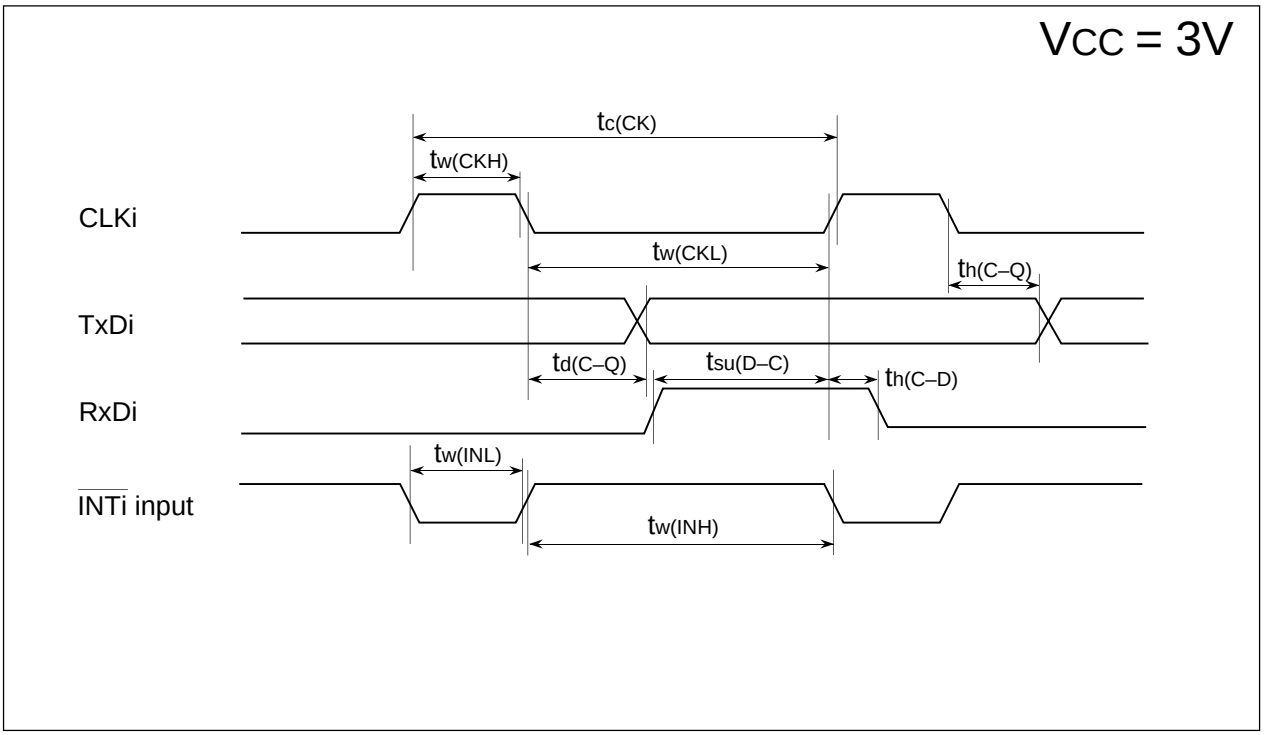


Figure 18.4. Timing Diagram (2)

Downloaded from Arrow.com.

18.2. M16C/26T (T version)

Table 18.38. Absolute Maximum Ratings

Symbol	Parameter			Condition	Value	Unit
V_{CC}	Supply Voltage			V_{CC} = AV_{CC}	-0.3 to 6.5	V
AV_{CC}	Analog Supply Voltage			V_{CC} = AV_{CC}	-0.3 to 6.5	V
V_I	Input Voltage	$P1_5$ to $P1_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P10_0$ to $P10_7$, X_{IN}, V_{REF}, $\overline{RESET}$, CNV_{SS}			-0.3 to V_{CC}+0.3	V
V_O	Output Voltage	$P1_5$ to $P1_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P10_0$ to $P10_7$, X_{OUT}			-0.3 to V_{CC}+0.3	V
Pd	Power Dissipation			-40 ≤ Topr ≤ 85° C	300	mW
Topr	Operating Ambient Temperature	during CPU operation			-40 to 85	° C
		during flash memory program and erase operation	Program Space (Block 0 to Block 3)		0 to 60	° C
			Data Space (Block A, Block B)		-40 to 85	° C
Tstg	Storage Temperature				-65 to 150	° C

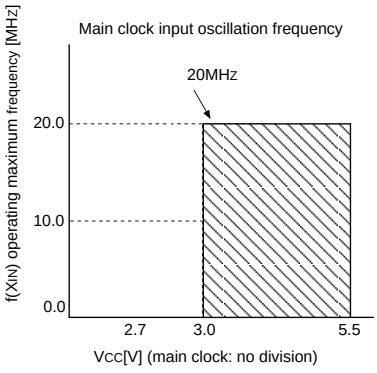
Downloaded from Arrow.com.

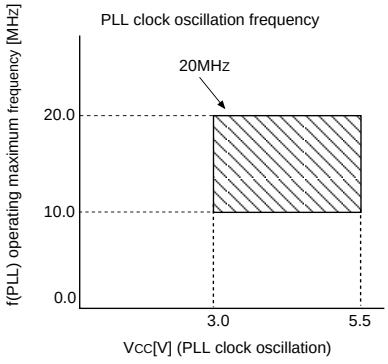
Table 18.39. Recommended Operating Conditions (1)

Symbol	Parameter		Standard			Unit
			Min.	Typ.	Max.	
V_{CC}	Supply Voltage		3.0		5.5	V
AV_{CC}	Analog Supply Voltage			V_{CC}		V
V_{SS}	Supply Voltage			0		V
AV_{SS}	Analog Supply Voltage			0		V
V_{IH}	Input High ("H") Voltage	$P1_5$ to $P1_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P10_0$ to $P10_7$	0.7 V_{CC}		V_{CC}	V
		XIN, $\overline{RESET}$, CNVSS	0.8 V_{CC}		V_{CC}	V
V_{IL}	Input Low ("L") Voltage	$P1_5$ to $P1_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P10_0$ to $P10_7$	0		0.3 V_{CC}	V
		XIN, $\overline{RESET}$, CNVSS	0		0.2V_{CC}	V
$I_{OH(peak)}$	Peak Output High ("H") Current	$P1_5$ to $P1_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P10_0$ to $P10_7$			-10.0	mA
$I_{OH(avg)}$	Average Output High ("H") Current	$P1_5$ to $P1_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P10_0$ to $P10_7$			-5.0	mA
$I_{OL(peak)}$	Peak Output Low ("L") Current	$P1_5$ to $P1_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P10_0$ to $P10_7$			10.0	mA
$I_{OL(avg)}$	Average Output Low ("L") Current	$P1_5$ to $P1_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P10_0$ to $P10_7$			5.0	mA
$f(X_{IN})$	Main Clock Input Oscillation Frequency(4)		0		20	MHz
$f(X_{CIN})$	Sub Clock Oscillation Frequency			32.768	50	kHz
f_1(ROC)	On-chip Oscillator Frequency 1		0.5	1	2	MHz
f2(ROC)	On-chip Oscillator Frequency 2		1	2	4	MHz
f3(ROC)	On-chip Oscillator Frequency 3		8	16	26	MHz
f(PLL)	PLL Clock Oscillation Frequency(4)		10		20	MHz
f(BCLK)	CPU Operation Clock Frequency		0		20	MHz
t_{SU}(PLL)	Wait Time to Stabilize PLL Frequency Synthesizer	V_{CC} = 5.0 V			20	ms
		V_{CC} = 3.0 V			50	ms

NOTES:

1. Referenced to V_{CC} = 3.0 to 5.5 V at Topr = -40 to 85 ° C unless otherwise specified.
2. The mean output current is the mean value within 100 ms.
3. The total $I_{OL(peak)}$ for all ports must be 80 mA or less. The total $I_{OH(peak)}$ for all ports must be -80 mA or less.
4. Relationship among main clock oscillation frequency, PLL clock oscillation frequency and supply voltage.





Downloaded from Arrow.com.

Table 18.40. A/D Conversion Characteristics (1)

Symbol	Parameter		Measurement Condition	Standard			Unit
				Min.	Typ.	Max.	
-	Resolution		$V_{REF} = V_{CC}$			10	Bits
INL	Integral Nonlinearity Error	10 bit	$V_{REF} = V_{CC} = 5$ V			±3	LSB
			$V_{REF} = V_{CC} = 3.3$ V			±5	LSB
		8 bit	$V_{REF} = V_{CC} = 3.3$ V, 5 V			±2	LSB
-	Absolute Accuracy	10 bit	$V_{REF} = V_{CC} = 5$ V			±3	LSB
			$V_{REF} = V_{CC} = 3.3$ V			±5	LSB
		8 bit	$V_{REF} = V_{CC} = 3.3$ V, 5 V			±2	LSB
DNL	Differential Nonlinearity Error					±1	LSB
-	Offset Error					±3	LSB
-	Gain Error					±3	LSB
R_{LADDER}	Resistor Ladder		$V_{REF} = V_{CC}$	10		40	kΩ
t_{CONV}	10-bit Conversion Time Sample & Hold Function Available		$V_{REF} = V_{CC}$=5 V, ϕAD = 10 MHz	3.3			µs
t_{CONV}	8-bit Conversion Time Sample & Hold Function Available		$V_{REF} = V_{CC} = 5$ V, ϕAD = 10 MHz	2.8			µs
V_{REF}	Reference Voltage			2.0		V_{CC}	V
V_{IA}	Analog Input Voltage			0		V_{REF}	V

NOTES:
1. Referenced to $V_{CC} = AV_{CC} = V_{REF} = 3.3$ to 5.5 V, $V_{SS} = AV_{SS} = 0$ V at Topr = -40 to 85° C unless otherwise specified.
2. Keep ϕAD frequency at 10 MHz or less. Additionally, divide the f_{AD} if V_{CC} is less than 4.2 V, and make ϕAD frequency equal to or lower than $f_{AD}/2$.
3. When sample & hold function is disabled, keep ϕAD frequency at 250 kHz or more in addition to the limitation in Note 2. When sample & hold function is enabled, keep ϕAD frequency at 1 MHz or more in addition to the limitation in Note 2.
4. When sample & hold function is enabled, sampling time is 3/ ϕAD frequency. When sample & hold function is disabled, sampling time is 2/ ϕAD frequency.

Downloaded from Arrow.com.

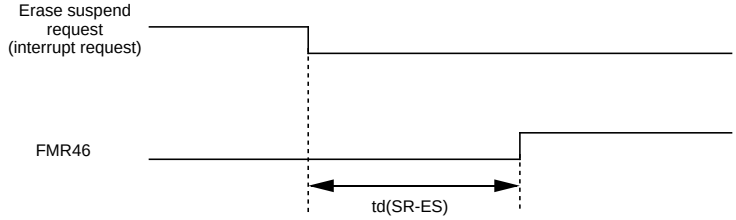
Table 18.41. Flash Memory Version Electrical Characteristics (1): Program Space and Data Space for U3, Program Space for U7

Symbol	Parameter		Standard			Unit
			Min.	Typ.(2)	Max.	
-	Program and Erase Endurance(3)		100/1000(4, 11)			cycles
-	Word Program Time (Vcc = 5.0 V, Topr = 25° C)			75	600	µs
-	Block Erase Time (Vcc = 5.0 V, Topr = 25° C)	2-Kbyte Block		0.2	9	s
		8-Kbyte Block		0.4	9	s
		16-Kbyte Block		0.7	9	s
		32-Kbyte Block		1.2	9	s
td(SR-ES)	Duration between Suspend Request and Erase Suspend				8	ms
tPS	Wait Time to Stabilize Flash Memory Circuit				15	µs
-	Data Hold Time (5)		20			years

Table 18.42. Flash Memory Version Electrical Characteristics (6): Data Space for U7(7)

Symbol	Parameter	Standard			Unit
		Min.	Typ.(2)	Max.	
-	Program and Erase Endurance(3, 8, 9)	10000(4, 10)			cycles
-	Word Program Time (Vcc = 5.0 V, Topr = 25° C)		100		µs
-	Block Erase Time (Vcc = 5.0V, Topr = 25° C) (2-Kbyte block)		0.3		s
td(SR-ES)	Duration between Suspend Request and Erase Suspend			8	ms
tPS	Wait Time to Stabilize Flash Memory Circuit			15	µs
-	Data Hold Time (5)	20			years

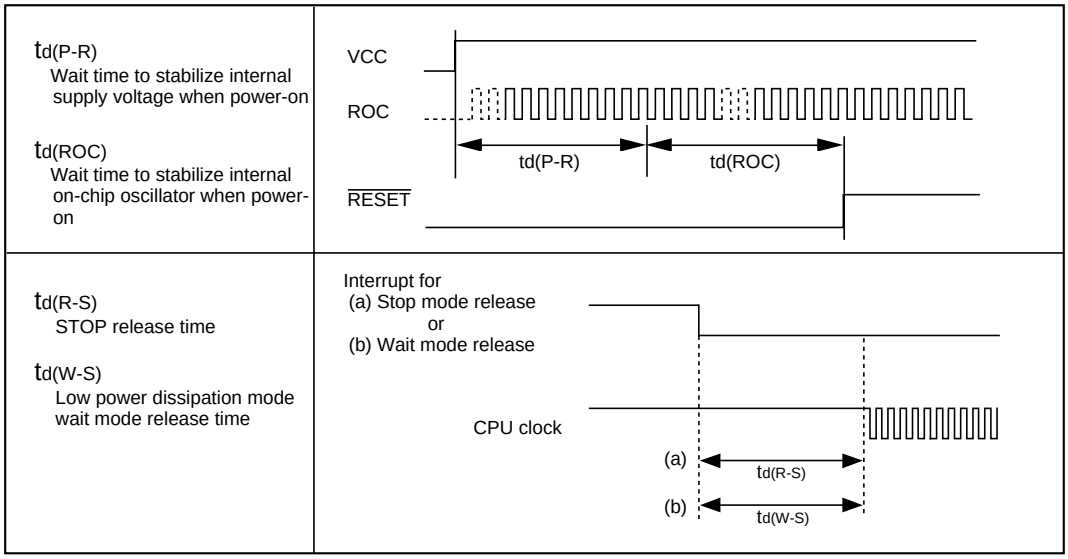
NOTES:
1. Referenced to VCC = 3.0 to 5.5 V at Topr = 0 to 60° C (program space)/ Topr = -40 to 85° C(data space), unless otherwise specified.
2. Vcc = 5.0 V; Topr = 25° C
3. Program and erase endurance is defined as number of program-erase cycles per block.
   If program and erase endurance is *n* cycle (*n* = 100, 1000, 10000), each block can be erased and programmed *n* cycles.
   For example, if a 2-Kbyte block A is erased after programming one-word data to each address 1,024 times, this counts as one program and erase endurance. Data cannot be programmed to the same address more than once without erasing the block. (rewrite prohibited).
4. Number of E/W cycles for which operation is guranteed (1 to minimum value are guranteed).
5. Topr = 55° C
6. Referenced to VCC = 3.0 to 5.5 V at Topr = -40 to 85° C unless otherwise specified.
7. **Table 18.42** applies for data space in U7 when program and erase endurance is more than 1,000 cycles. Otherwise, use **Table 18.41**.
8. To reduce the number of program and erase endurance when working with systems requiring numerous rewrites, write to unused word addresses within the block instead of rewrite. Erase block only after all possible addresses are used. For example, an 8-word program can be written 128 times maximum before erase becomes necessary. Maintaining an equal number of times erasure between block A and block B will also improve efficiency. It is recommended to track the total number of erasure performed per block and to limit the number of erasure.
9. If an erase error is generated during block erase, execute the clear status register command and block erase command at least 3 times until an erase error is not generated.
10. When executing more than 100 times rewrites, set one wait state per block access by setting the FMR17 bit in the FMR1 register to 1 (wait state). When accessing to all other blocks and internal RAM, wait state can be set by the PM17 bit, regardless of the FMR17 bit setting value.
11. The program and erase endurance is 100 cycles for program space and data space in U3; 1,000 cycles for program space in U7.
12. Please contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor for further details on the E/W failure rate.



Downloaded from Arrow.com.

Table 18.43. Power Supply Circuit Timing Characteristics

Symbol	Parameter	Measurement Condition	Standard			Unit
			Min.	Typ.	Max.	
td(P-R)	Wait Time to Stabilize Internal Supply Voltage when Power-on	V_{CC} = 3.0 to 5.5V			2	ms
td(ROC)	Wait Time to Stabilize Internal On-chip Oscillator when Power-on				40	µs
td(R-S)	STOP Release Time(1)				1.5	ms
td(W-S)	Low Power Dissipation Mode Wait Mode Release Time				250	µs



Downloaded from Arrow.com.

V_{CC} = 5V

Table 18.44. Electrical Characteristics (1)

Symbol	Parameter				Condition	Standard			Unit
						Min.	Typ.	Max.	
V_{OH}	Output High ("H") Voltage	$P1_5$ to $P1_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P10_0$ to $P10_7$			I_{OH} = -5 mA	V_{CC}-2.0		V_{CC}	V
V_{OH}	Output High ("H") Voltage	$P1_5$ to $P1_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P10_0$ to $P10_7$			I_{OH} = -200 µA	V_{CC}-0.3		V_{CC}	V
V_{OH}	Output High ("H") Voltage		X_{OUT}	High Power	I_{OH} = -1 mA	V_{CC}-2.0		V_{CC}	V
				Low Power	I_{OH} = -0.5 mA	V_{CC}-2.0		V_{CC}	
	Output High ("H") Voltage		X_{COUT}	High Power	No load applied		2.5		V
				Low Power	No load applied		1.6		
V_{OL}	Output Low ("L") Voltage	$P1_5$ to $P1_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P10_0$ to $P10_7$			I_{OL} = 5 mA			2.0	V
V_{OL}	Output Low ("L") Voltage	$P1_5$ to $P1_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P10_0$ to $P10_7$			I_{OL} = 200 µA			0.45	V
V_{OL}	Output Low ("L") Voltage		X_{OUT}	High Power	I_{OL} = 1 mA			2.0	V
				Low Power	I_{OL} = 0.5 mA			2.0	
	Output Low ("L") Voltage		X_{COUT}	High Power	No load applied		0		V
				Low Power	No load applied		0		
V_{T+}-V_{T-}	Hysteresis	TA0IN-TA4IN, TB0IN-TB2IN, $\overline{INT_0}$-$\overline{INT_5}$, $\overline{NMI}$, $\overline{ADTRG}$, $\overline{CTS_0}$-$\overline{CTS_2}$, CLK0-CLK2, TA2OUT-TA4OUT, $\overline{KI_0}$-$\overline{KI_3}$, R_{XD0}-R_{XD2}				0.2		1.0	V
V_{T+}-V_{T-}	Hysteresis	$\overline{RESET}$				0.2		2.5	V
V_{T+}-V_{T-}	Hysteresis	X_{IN}				0.2		0.8	V
I_{IH}	Input High ("H") Current	$P1_5$ to $P1_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P10_0$ to $P10_7$, X_{IN}, $\overline{RESET}$, CNV_{SS}			V_I = 5 V			5.0	µA
I_{IL}	Input Low ("L") Current	$P1_5$ to $P1_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P10_0$ to $P10_7$, X_{IN}, $\overline{RESET}$, CNV_{SS}			V_I = 0 V			-5.0	µA
R_{PULLUP}	Pull-up Resistance	$P1_5$ to $P1_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P10_0$ to $P10_7$			V_I = 0 V	30	50	170	kΩ
Rf_{XIN}	Feedback Resistance		X_{IN}				1.5		MΩ
Rf_{XCIN}	Feedback Resistance		X_{CIN}				15		MΩ
V_{RAM}	RAM Standby Voltage				In stop mode	2.0			V

NOTE:

1. Referenced to V_{CC} = 4.2 to 5.5 V, V_{SS} = 0 V at Topr = -40 to 85 ° C, f(BCLK) = 20 MHz unless otherwise specified.

Downloaded from Arrow.com.

V_{CC} = 5V

Table 18.45. Electrical Characteristics (2) [1]

Symbol	Parameter	Measurement Condition			Standard			Unit
					Min.	Typ.	Max.	
I_{CC}	Power Supply Current (V_{CC}=4.0 to 5.5V)	Output pins are left open and other pins are connected to V_{SS}	Flash memory	f(BCLK) = 20 MHz, Main clock, no division		16	19	mA
				On-chip oscillator operates, $f_{2(ROC)}$ selected, f(BCLK) = 1 MHz		1		mA
			Flash memory program	f(BCLK) = 10 MHz, Vcc = 5.0 V		11		mA
			Flash memory erase	f(BCLK) = 10 MHz, Vcc = 5.0 V		12		mA
			Flash memory	f(BCLK) = 32 kHz, In low-power consumption mode, Program running on RAM[3]		25		µA
				f(BCLK) = 32 kHz, In low-power consumption mode, Program running on flash memory[3]		450		µA
				On-chip oscillation, $f_{2(ROC)}$ selected, f(BCLK) = 1 MHz, In wait mode		50		µA
				f(BCLK) = 32 kHz, In wait mode[2], Oscillation capacity HIGH		10		µA
				f(BCLK) = 32 kHz, In wait mode[2], Oscillation capacity LOW		3		µA
				While clock stops, Topr = 25° C		0.8	3	µA

NOTES:

1. Referenced to V_{CC} = 4.2 to 5.5 V, V_{SS} = 0 V at Topr = -40 to 85 ° C, f(BCLK) = 20 MHz unless otherwise specified.
2. With one timer operates, using f_{C32}.
3. This indicates the memory in which the program to be executed exists.

Downloaded from Arrow.com.

V_{CC} = 5V

Timing Requirements

(V_{CC} = 5V, V_{SS} = 0V, at Topr = – 40 to 85°C unless otherwise specified)

Table 18.46. External Clock Input (X_{IN} input)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t_c	External clock input cycle time	50		ns
$t_{w(H)}$	External clock input HIGH pulse width	20		ns
$t_{w(L)}$	External clock input LOW pulse width	20		ns
t_r	External clock rise time		9	ns
t_f	External clock fall time		9	ns

Downloaded from Arrow.com.

VCC = 5V

Timing Requirements

(VCC = 5V, VSS = 0V, at Topr = – 40 to 85°C unless otherwise specified)

Table 18.47. Timer A Input (Counter Input in Event Counter Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIN input cycle time	100		ns
$t_{w(TAH)}$	TAiIN input HIGH pulse width	40		ns
$t_{w(TAL)}$	TAiIN input LOW pulse width	40		ns

Table 18.48. Timer A Input (Gating Input in Timer Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIN input cycle time	400		ns
$t_{w(TAH)}$	TAiIN input HIGH pulse width	200		ns
$t_{w(TAL)}$	TAiIN input LOW pulse width	200		ns

Table 18.49. Timer A Input (External Trigger Input in One-shot Timer Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIN input cycle time	200		ns
$t_{w(TAH)}$	TAiIN input HIGH pulse width	100		ns
$t_{w(TAL)}$	TAiIN input LOW pulse width	100		ns

Table 18.50. Timer A Input (External Trigger Input in Pulse Width Modulation Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{w(TAH)}$	TAiIN input HIGH pulse width	100		ns
$t_{w(TAL)}$	TAiIN input LOW pulse width	100		ns

Table 18.51. Timer A Input (Counter Increment/decrement Input in Event Counter Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(UP)}$	TAiOUT input cycle time	2000		ns
$t_{w(UPH)}$	TAiOUT input HIGH pulse width	1000		ns
$t_{w(UPL)}$	TAiOUT input LOW pulse width	1000		ns
$t_{su(UP-TIN)}$	TAiOUT input setup time	400		ns
$t_{h(TIN-UP)}$	TAiOUT input hold time	400		ns

Table 18.52. Timer A Input (Two-phase Pulse Input in Event Counter Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIN input cycle time	800		ns
$t_{su(TAIN-TAOUT)}$	TAiOUT input setup time	200		ns
$t_{su(TAOUT-TAIN)}$	TAiIN input setup time	200		ns

Downloaded from Arrow.com.

V_{CC} = 5V

Timing Requirements

(V_{CC} = 5V, V_{SS} = 0V, at Topr = – 40 to 85°C unless otherwise specified)

Table 18.53. Timer B Input (Counter Input in Event Counter Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIN input cycle time (counted on one edge)	100		ns
$t_{w(TBH)}$	TBiIN input HIGH pulse width (counted on one edge)	40		ns
$t_{w(TBL)}$	TBiIN input LOW pulse width (counted on one edge)	40		ns
$t_{c(TB)}$	TBiIN input cycle time (counted on both edges)	200		ns
$t_{w(TBH)}$	TBiIN input HIGH pulse width (counted on both edges)	80		ns
$t_{w(TBL)}$	TBiIN input LOW pulse width (counted on both edges)	80		ns

Table 18.54. Timer B Input (Pulse Period Measurement Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIN input cycle time	400		ns
$t_{w(TBH)}$	TBiIN input HIGH pulse width	200		ns
$t_{w(TBL)}$	TBiIN input LOW pulse width	200		ns

Table 18.55. Timer B Input (Pulse Width Measurement Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIN input cycle time	400		ns
$t_{w(TBH)}$	TBiIN input HIGH pulse width	200		ns
$t_{w(TBL)}$	TBiIN input LOW pulse width	200		ns

Table 18.56. A/D Trigger Input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(AD)}$	$\overline{ADTRG}$ input cycle time (trigger able minimum)	1000		ns
$t_{w(ADL)}$	$\overline{ADTRG}$ input LOW pulse width	125		ns

Table 18.57. Serial I/O

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(CK)}$	CLKi input cycle time	200		ns
$t_{w(CKH)}$	CLKi input HIGH pulse width	100		ns
$t_{w(CKL)}$	CLKi input LOW pulse width	100		ns
$t_{d(C-Q)}$	TxDi output delay time		80	ns
$t_{h(C-Q)}$	TxDi hold time	0		ns
$t_{su(D-C)}$	RxDi input setup time	70		ns
$t_{h(C-D)}$	RxDi input hold time	90		ns

Table 18.58. External Interrupt $\overline{INTi}$ Input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{w(INH)}$	$\overline{INTi}$ input HIGH pulse width	250		ns
$t_{w(INL)}$	$\overline{INTi}$ input LOW pulse width	250		ns

Downloaded from Arrow.com.

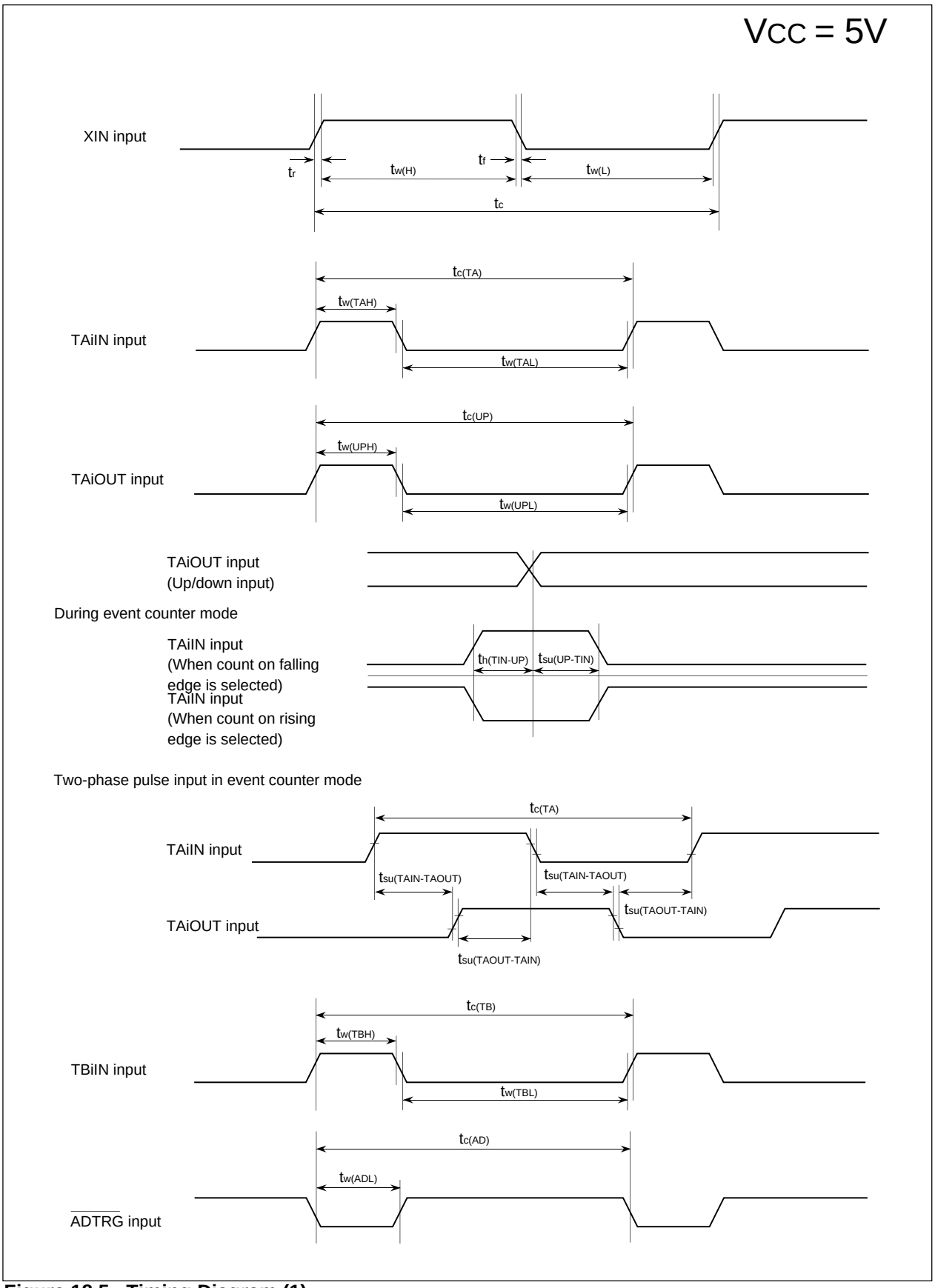


Figure 18.5. Timing Diagram (1)

Downloaded from Arrow.com.

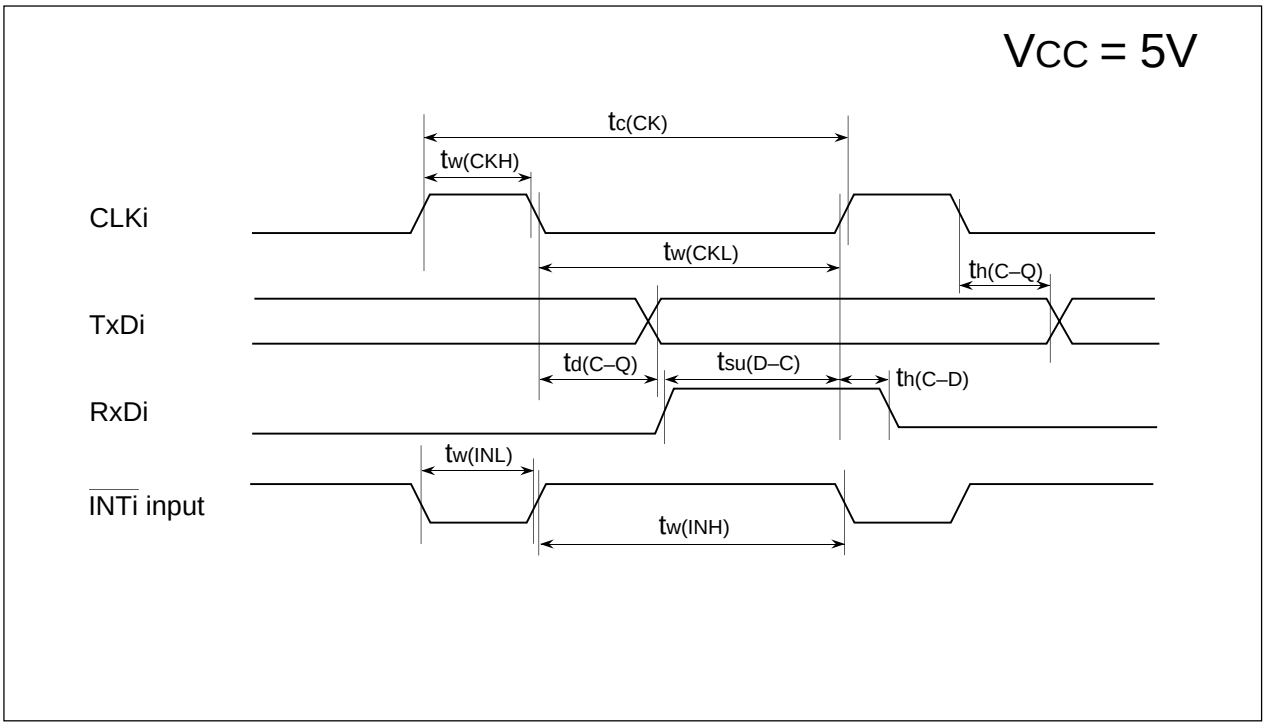


Figure 18.6. Timing Diagram (2)

Downloaded from Arrow.com.

V_{CC} = 3V

Table 18.59. Electrical Characteristics (1)

Symbol	Parameter			Condition	Standard			Unit
					Min.	Typ.	Max.	
V_{OH}	Output High ("H") Voltage	$P1_5$ to $P1_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P10_0$ to $P10_7$		I_{OH} = -1 mA	V_{CC}-0.5		V_{CC}	V
V_{OH}	Output High ("H") Voltage	X_{OUT}	High Power	I_{OH} = -0.1 mA	V_{CC}-0.5		V_{CC}	V
			Low Power	I_{OH} = -50 µA	V_{CC}-0.5		V_{CC}	
	Output High ("H") Voltage	X_{COUT}	High Power	No load applied		2.5		V
			Low Power	No load applied		1.6		
V_{OL}	Output Low ("L") Voltage	$P1_5$ to $P1_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P10_0$ to $P10_7$		I_{OL} = 1 mA			0.5	V
V_{OL}	Output Low ("L") Voltage	X_{OUT}	High Power	I_{OL} = 0.1 mA			0.5	V
			Low Power	I_{OL} = 50 µA			0.5	
	Output Low ("L") Voltage	X_{COUT}	High Power	No load applied		0		V
			Low Power	No load applied		0		
V_{T+}-V_{T-}	Hysteresis	$TA0_{IN}$-$TA4_{IN}$, $TB0_{IN}$-$TB2_{IN}$, $\overline{INT_0}$-$\overline{INT_5}$, $\overline{NMI}$, $\overline{ADTRG}$, $\overline{CTS_0}$-$\overline{CTS_2}$, CLK_0-CLK_2, $TA2_{OUT}$-$TA4_{OUT}$, $\overline{KI_0}$-$\overline{KI_3}$, RXD_0-RXD_2					0.8	V
V_{T+}-V_{T-}	Hysteresis	$\overline{RESET}$					1.8	V
V_{T+}-V_{T-}	Hysteresis	X_{IN}					0.8	V
I_{IH}	Input High ("H") Current	$P1_5$ to $P1_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P10_0$ to $P10_7$, X_{IN}, $\overline{RESET}$, CNV_{SS}		V_I = 3 V			4.0	µA
I_{IL}	Input Low ("L") Current	$P1_5$ to $P1_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P10_0$ to $P10_7$, X_{IN}, $\overline{RESET}$, CNV_{SS}		V_I = 0 V			-4.0	µA
R_{PULLUP}	Pull-up Resistance	$P1_5$ to $P1_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P10_0$ to $P10_7$		V_I = 0 V	50	100	500	kΩ
Rf_{XIN}	Feedback Resistance	X_{IN}				3.0		MΩ
Rf_{XCIN}	Feedback Resistance	X_{CIN}				25		MΩ
V_{RAM}	RAM Standby Voltage			In stop mode	2.0			V

NOTE:

1. Referenced to V_{CC} = 3.0 to 3.6 V, V_{SS} = 0 V at Topr = -40 to 85 ° C, f(BCLK) = 20 MHz unless otherwise specified.

Downloaded from Arrow.com.

V_{CC} = 3V

Table 18.60. Electrical Characteristics (2) [1]

Symbol	Parameter	Measurement Condition			Standard			Unit
					Min.	Typ.	Max.	
I_{CC}	Power Supply Current (V_{CC}=3.0 to 3.6V)	Output pins are left open and other pins are connected to V_{SS}	Flash memory	f(BCLK) = 10 MHz, Main clock, no division		7	12	mA
				On-chip oscillator operates, $f_{2(ROC)}$ selected, f(BCLK) = 1 MHz		1		mA
			Flash memory program	f(BCLK) = 10 MHz, Vcc = 3.0 V		10		mA
			Flash memory erase	f(BCLK) = 10MHz, Vcc = 3.0 V		11		mA
			Flash memory	f(BCLK) = 32 kHz, In low-power consumption mode, Program running on RAM[3]		25		µA
				f(BCLK) = 32 kHz, In low-power consumption mode, Program running on flash memory[3]		450		µA
				On-chip oscillator operates, $f_{2(ROC)}$ selected, f(BCLK) = 1 MHz, In wait mode		45		µA
				f(BCLK) = 32 kHz, In wait mode[2], Oscillation capacity HIGH		10		µA
				f(BCLK) = 32 kHz, In wait mode[2], Oscillation capacity LOW		3		µA
				While clock stops, Topr = 25° C		0.7	3	µA

NOTES:

1. Referenced to V_{CC} = 3.0 to 3.6 V, V_{SS} = 0 V at Topr = -40 to 85 ° C, f(BCLK) = 20 MHz unless otherwise specified.
2. With one timer operates, using f_{C32}.
3. This indicates the memory in which the program to be executed exists.

Downloaded from Arrow.com.

VCC = 3V

Timing Requirements

(VCC = 3V, VSS = 0V, at Topr = – 40 to 85°C unless otherwise specified)

Table 18.61. External Clock Input (XIN input)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t_c	External clock input cycle time	100		ns
$t_{w(H)}$	External clock input HIGH pulse width	40		ns
$t_{w(L)}$	External clock input LOW pulse width	40		ns
t_r	External clock rise time		18	ns
t_f	External clock fall time		18	ns

Downloaded from Arrow.com.

Vcc = 3V

Timing Requirements

(Vcc = 3V, Vss = 0V, at Topr = – 40 to 85°C unless otherwise specified)

Table 18.62. Timer A Input (Counter Input in Event Counter Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIN input cycle time	150		ns
$t_{w(TAH)}$	TAiIN input HIGH pulse width	60		ns
$t_{w(TAL)}$	TAiIN input LOW pulse width	60		ns

Table 18.63. Timer A Input (Gating Input in Timer Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIN input cycle time	600		ns
$t_{w(TAH)}$	TAiIN input HIGH pulse width	300		ns
$t_{w(TAL)}$	TAiIN input LOW pulse width	300		ns

Table 18.64. Timer A Input (External Trigger Input in One-shot Timer Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIN input cycle time	300		ns
$t_{w(TAH)}$	TAiIN input HIGH pulse width	150		ns
$t_{w(TAL)}$	TAiIN input LOW pulse width	150		ns

Table 18.65. Timer A Input (External Trigger Input in Pulse Width Modulation Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{w(TAH)}$	TAiIN input HIGH pulse width	150		ns
$t_{w(TAL)}$	TAiIN input LOW pulse width	150		ns

Table 18.66. Timer A Input (Counter Increment/decrement Input in Event Counter Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(UP)}$	TAiOUT input cycle time	3000		ns
$t_{w(UPH)}$	TAiOUT input HIGH pulse width	1500		ns
$t_{w(UPL)}$	TAiOUT input LOW pulse width	1500		ns
$t_{su(UP-TIN)}$	TAiOUT input setup time	600		ns
$t_{h(TIN-UP)}$	TAiOUT input hold time	600		ns

Table 18.67. Timer A Input (Two-phase Pulse Input in Event Counter Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIN input cycle time	2		µs
$t_{su(TAIN-TAOUT)}$	TAiOUT input setup time	500		ns
$t_{su(TAOUT-TAIN)}$	TAiIN input setup time	500		ns

Downloaded from Arrow.com.

V_{CC} = 3V

Timing Requirements

(V_{CC} = 3V, V_{SS} = 0V, at Topr = – 40 to 85°C unless otherwise specified)

Table 18.68. Timer B Input (Counter Input in Event Counter Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBi_{IN} input cycle time (counted on one edge)	150		ns
$t_{w(TBH)}$	TBi_{IN} input HIGH pulse width (counted on one edge)	60		ns
$t_{w(TBL)}$	TBi_{IN} input LOW pulse width (counted on one edge)	60		ns
$t_{c(TB)}$	TBi_{IN} input cycle time (counted on both edges)	300		ns
$t_{w(TBH)}$	TBi_{IN} input HIGH pulse width (counted on both edges)	120		ns
$t_{w(TBL)}$	TBi_{IN} input LOW pulse width (counted on both edges)	120		ns

Table 18.69. Timer B Input (Pulse Period Measurement Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBi_{IN} input cycle time	600		ns
$t_{w(TBH)}$	TBi_{IN} input HIGH pulse width	300		ns
$t_{w(TBL)}$	TBi_{IN} input LOW pulse width	300		ns

Table 18.70. Timer B Input (Pulse Width Measurement Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBi_{IN} input cycle time	600		ns
$t_{w(TBH)}$	TBi_{IN} input HIGH pulse width	300		ns
$t_{w(TBL)}$	TBi_{IN} input LOW pulse width	300		ns

Table 18.71. A/D Trigger Input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(AD)}$	$\overline{ADTRG}$ input cycle time (trigger able minimum)	1500		ns
$t_{w(ADL)}$	$\overline{ADTRG}$ input LOW pulse width	200		ns

Table 18.72. Serial I/O

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(CK)}$	CLKi input cycle time	300		ns
$t_{w(CKH)}$	CLKi input HIGH pulse width	150		ns
$t_{w(CKL)}$	CLKi input LOW pulse width	150		ns
$t_{d(C-Q)}$	TxDi output delay time		160	ns
$t_{h(C-Q)}$	TxDi hold time	0		ns
$t_{su(D-C)}$	RxDi input setup time	100		ns
$t_{h(C-D)}$	RxDi input hold time	90		ns

Table 18.73. External Interrupt $\overline{INTi}$ Input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{w(INH)}$	$\overline{INTi}$ input HIGH pulse width	380		ns
$t_{w(INL)}$	$\overline{INTi}$ input LOW pulse width	380		ns

Downloaded from Arrow.com.

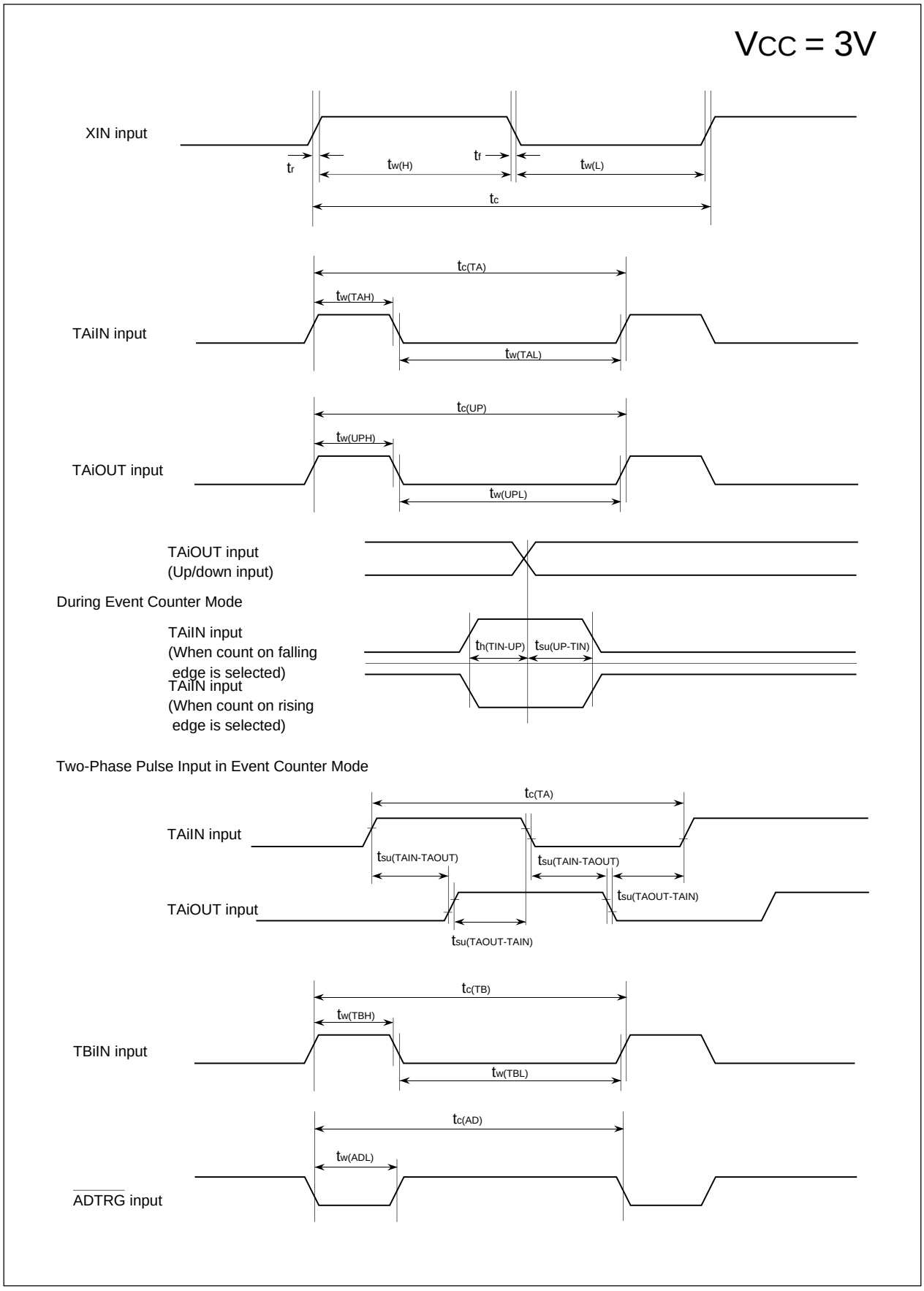


Figure 18.7. Timing Diagram (1)

Downloaded from Arrow.com.

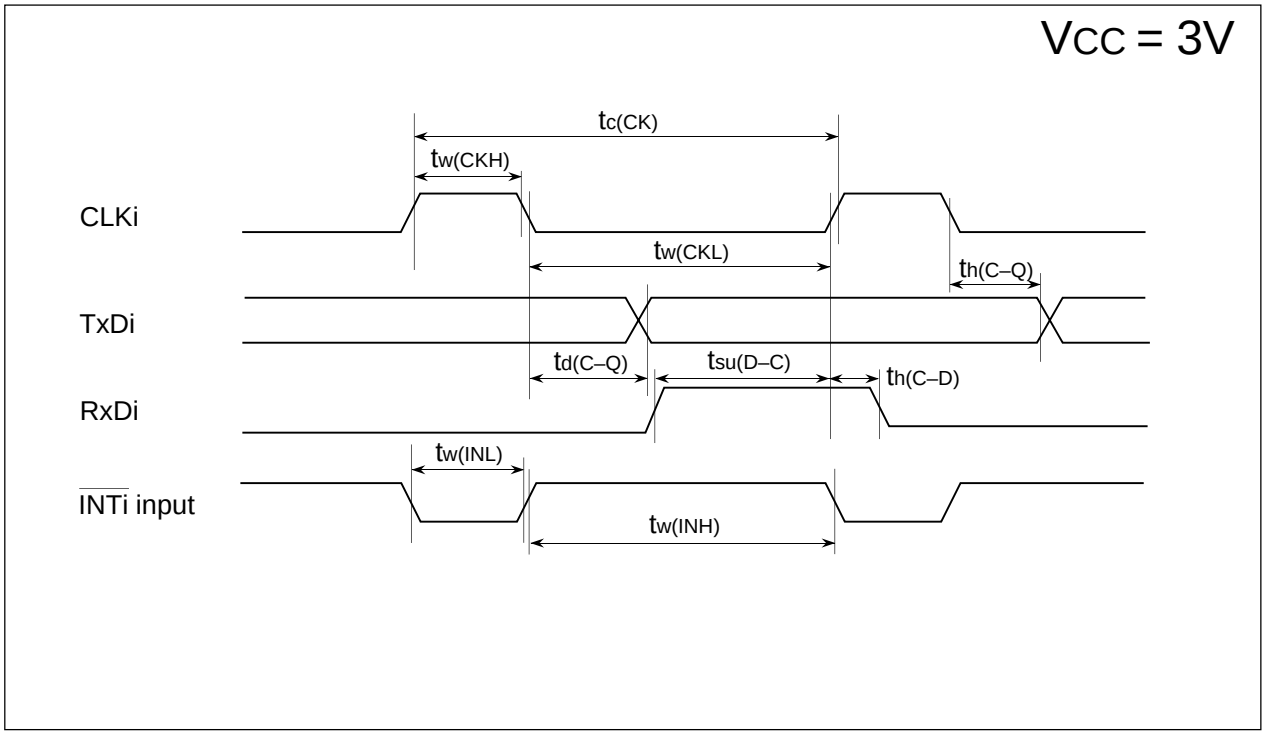


Figure 18.8. Timing Diagram (2)

Downloaded from Arrow.com.

19. Usage Notes

19.1 SFR

19.1.1 Precaution for 48-pin package

Set the IFSR20 bit in the IFSR2A register to "1" after reset and set the PACR2 to PACR0 bits in the PACR register to "100_2".

19.1.2 Precaution for 42-pin package

Set the IFSR20 bit in the IFSR2A register to "1" after reset and set the PACR2 to PACR0 bits in the PACR register to "001_2".

19.1.3 Register Setting

Immediate values should be set in the registers containing write-only bits. When establishing a new value by modifying a previous value, write the previous value into RAM as well as the register. Change the contents of the RAM and then transfer the new value to the register.

Downloaded from Arrow.com.

19.2 PLL Frequency Synthesizer

Stabilize supply voltage so that the standard of the power supply ripple is met.

Symbol	Parameter		Standard			Unit
			Min.	Typ.	Max.	
$f_{(ripple)}$	Power supply ripple allowable frequency(V_{CC})				10	kHz
$V_{p-p(ripple)}$	Power supply ripple allowabled amplitude voltage	(V_{CC}=5V)			0.5	V
		(V_{CC}=3V)			0.3	V
$V_{CC(\lvert\Delta V/\Delta T\rvert)}$	Power supply ripple rising/falling gradient	(V_{CC}=5V)			0.3	V/ms
		(V_{CC}=3V)			0.3	V/ms

$f_{(ripple)}$
Power supply ripple allowable frequency (V_{CC})
$V_{p-p(ripple)}$
Power supply ripple allowable amplitude voltage

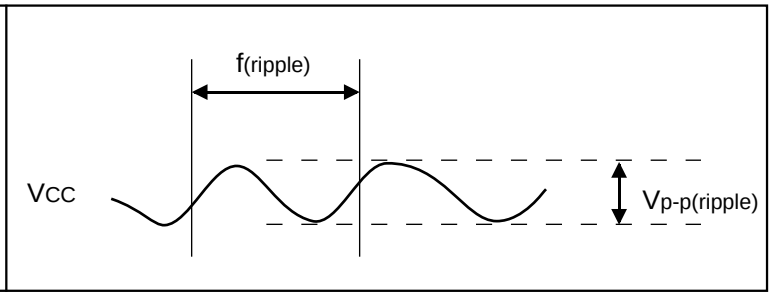


Figure 19.1 Timing of Voltage Fluctuation

Downloaded from Arrow.com.

19.3 Power Control

1. When exiting stop mode by hardware reset, the device will startup using the on-chip oscillator.

2. Set the MR0 bit in the TAiMR register(i=0 to 4) to “0”(pulse is not output) to use the timer A to exit stop mode.

3. When entering wait mode, insert a JMP.B instruction before a WAIT instruction. Do not excute any instructions which can generate a write to RAM between the JMP.B and WAIT instructions. Disable the DMA transfers, if a DMA transfer may occur between the JMP.B and WAIT instructions. After the WAIT instruction, insert at least 4 NOP instructions. When entering wait mode, the instruction queue reads ahead the instructions following WAIT, and depending on timing, some of these may execute before the microcomputer enters wait mode.

Program example when entering wait mode

```
Program Example:    JMP.B   L1    ; Insert JMP.B instruction before WAIT instruction
                L1:
                    FSET    I     ;
                    WAIT          ; Enter wait mode
                    NOP           ; More than 4 NOP instructions
                    NOP
                    NOP
                    NOP
```

4. When entering stop mode, insert a JMP.B instruction immediately after executing an instruction which sets the CM10 bit in the CM1 register to “1”, and then insert at least 4 NOP instructions. When entering stop mode, the instruction queue reads ahead the instructions following the instruction which sets the CM10 bit to “1” (all clock stops), and, some of these may execute before the microcomputer enters stop mode or before the interrupt routine for returning from stop mode.

Program example when entering stop mode

```
Program Example:    FSET    I
                    BSET    CM10    ; Enter stop mode
                    JMP.B   L1      ; Insert JMP.B instruction
                L1:
                    NOP             ; More than 4 NOP instructions
                    NOP
                    NOP
                    NOP
```


Downloaded from Arrow.com.

5. Wait until the main clock oscillation stabilization time, before switching the CPU clock source to the main clock.
Similarly, wait until the sub clock oscillates stably before switching the CPU clock source to the sub clock.

6. Suggestions to reduce power consumption

(a) Ports

The processor retains the state of each I/O port even when it goes to wait mode or to stop mode. A current flows in active I/O ports. A dash current may flow through the input ports in high impedance state, if the input is floating. When entering wait mode or stop mode, set non-used ports to input and stabilize the potential.

(b) A/D converter

When A/D conversion is not performed, set the VCUT bit in the ADCON1 register to “0” (no V_{REF} connection). When A/D conversion is performed, start the A/D conversion at least 1 µs or longer after setting the VCUT bit to “1” (V_{REF} connection).

(c) Stopping peripheral functions

Use the CM02 bit in the CM0 register to stop the unnecessary peripheral functions during wait mode. However, because the peripheral function clock (f_{C32}) generated from the sub-clock does not stop, this measure is not conducive to reducing the power consumption of the chip. If low speed mode or low power dissipation mode is to be changed to wait mode, set the CM02 bit to “0” (do not stop peripheral function clocks in wait mode), before changing wait mode.

(d) Switching the oscillation-driving capacity

Set the driving capacity to “LOW” when oscillation is stable.

Downloaded from Arrow.com.

19.4 Protect

Set the PRC2 bit to “1” (write enabled) and then write to any address, and the PRC2 bit will be cleared to “0” (write protected). The registers protected by the PRC2 bit should be changed in the next instruction after setting the PRC2 bit to “1”. Make sure no interrupts or DMA transfers will occur between the instruction in which the PRC2 bit is set to “1” and the next instruction.

Downloaded from Arrow.com.

19.5 Interrupts

19.5.1 Reading address 00000$_{16}$

Do not read the address 00000$_{16}$ in a program. When a maskable interrupt request is accepted, the CPU reads interrupt information (interrupt number and interrupt request priority level) from the address 00000$_{16}$ during the interrupt sequence. At this time, the IR bit for the accepted interrupt is cleared to “0”. If the address 00000$_{16}$ is read in a program, the IR bit for the interrupt which has the highest priority among the enabled interrupts is cleared to “0”. This causes a problem that the interrupt is canceled, or an unexpected interrupt request is generated.

19.5.2 Setting the SP

Set any value in the SP(USP, ISP) before accepting an interrupt. The SP(USP, ISP) is cleared to ‘0000$_{16}$’ after reset. Therefore, if an interrupt is accepted before setting any value in the SP(USP, ISP), the program may go out of control.

19.5.3 The $\overline{\text{NMI}}$ Interrupt

1. The $\overline{\text{NMI}}$ interrupt is invalid after reset. The $\overline{\text{NMI}}$ interrupt becomes effective by setting to “1” the PM24 bit in the PM2 register. Set the PM24 bit to "1" when a high-level signal ("H") is applied to the $\overline{\text{NMI}}$ pin. If the PM24 bit is set to "1" when a low-level signal ("L") is applied, $\overline{\text{NMI}}$ interrupt is generated. Once $\overline{\text{NMI}}$ interrupt is enabled, it will not be disabled unless a reset is applied.
2. The input level of the $\overline{\text{NMI}}$ pin can be read by accessing the P8_5 bit in the P8 register.
3. When selecting $\overline{\text{NMI}}$ function, stop mode cannot be entered into while input on the $\overline{\text{NMI}}$ pin is low. This is because while input on the $\overline{\text{NMI}}$ pin is low the CM10 bit in the CM1 register is fixed to “0”.
4. When selecting $\overline{\text{NMI}}$ function, do not go to wait mode while input on the $\overline{\text{NMI}}$ pin is low. This is because when input on the $\overline{\text{NMI}}$ pin goes low, the CPU stops but CPU clock remains active; therefore, the current consumption in the chip does not drop. In this case, normal condition is restored by an interrupt generated thereafter.
5. When selecting $\overline{\text{NMI}}$ function, the low and high level durations of the input signal to the $\overline{\text{NMI}}$ pin must each be 2 CPU clock cycles + 300 ns or more.
6. When using the $\overline{\text{NMI}}$ interrupt for exiting stop mode, set the NDDR register to “FF$_{16}$” (disable digital debounce filter) before entering stop mode.

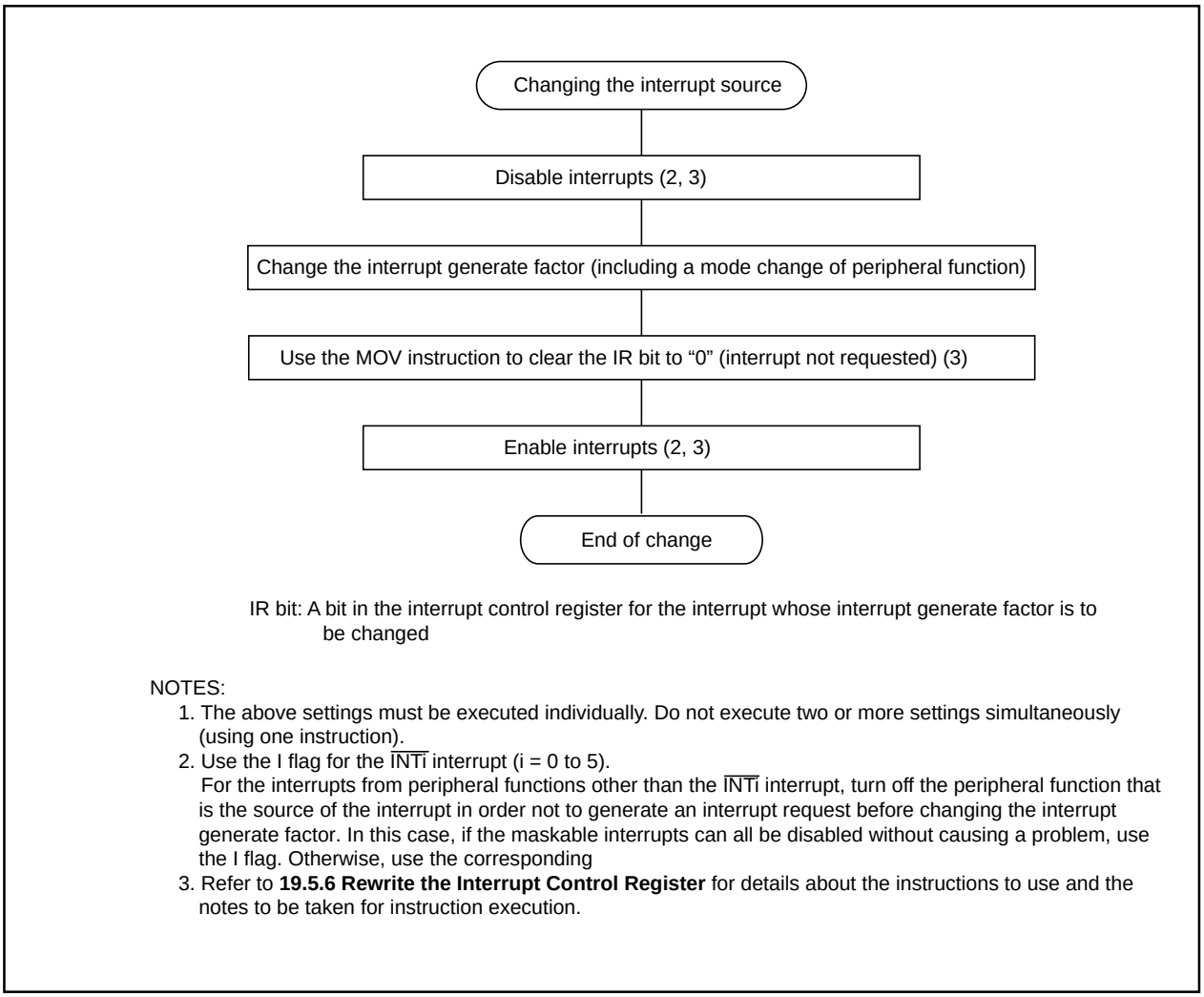
19.5.4 Changing the Interrupt Generation Factor

If the interrupt generate factor is changed, the IR bit in the interrupt control register for the changed interrupt may inadvertently be set to “1” (interrupt requested). If you changed the interrupt generate factor for an interrupt that needs to be used, be sure to clear the IR bit for that interrupt to “0” (interrupt not requested).

“Changing the interrupt generate factor” referred to here means any act of changing the source, polarity or timing of the interrupt assigned to each software interrupt number. Therefore, if a mode change of any peripheral function involves changing the generate factor, polarity or timing of an interrupt, be sure to clear the IR bit for that interrupt to “0” (interrupt not requested) after making such changes. Refer to the description of each peripheral function for details about the interrupts from peripheral functions.

Figure 19.2 shows the procedure for changing the interrupt generate factor.

Downloaded from Arrow.com.



IR bit: A bit in the interrupt control register for the interrupt whose interrupt generate factor is to be changed

NOTES:
1. The above settings must be executed individually. Do not execute two or more settings simultaneously (using one instruction).
2. Use the I flag for the $\overline{INTi}$ interrupt (i = 0 to 5).
   For the interrupts from peripheral functions other than the $\overline{INTi}$ interrupt, turn off the peripheral function that is the source of the interrupt in order not to generate an interrupt request before changing the interrupt generate factor. In this case, if the maskable interrupts can all be disabled without causing a problem, use the I flag. Otherwise, use the corresponding
3. Refer to **19.5.6 Rewrite the Interrupt Control Register** for details about the instructions to use and the notes to be taken for instruction execution.

Figure 19.2. Procedure for Changing the Interrupt Generate Factor

19.5.5 $\overline{INT}$ Interrupt

1. Either an “L” level of at least $t_{w(INH)}$ or an “H” level of at least $t_{w(INL)}$ width is necessary for the signal input to pins $\overline{INT_0}$ through $\overline{INT_5}$ regardless of the CPU operation clock.
2. If the POL bit in the INT0IC to INT5IC registers or the IFSR7 to IFSR0 bits in the IFSR register are changed, the IR bit may inadvertently set to 1 (interrupt requested). Be sure to clear the IR bit to 0 (interrupt not requested) after changing any of those register bits.
3. When using the INT_5 interrupt for exiting stop mode, set the P17DDR register to “FF_{16}” (disable digital debounce filter) before entering stop mode.

Downloaded from Arrow.com.

19.5.6 Rewrite the Interrupt Control Register

(1) The interrupt control register for any interrupt should be modified in places where no requests for that interrupt may occur. Otherwise, disable the interrupt before rewriting the interrupt control register.

(2) To rewrite the interrupt control register for any interrupt after disabling that interrupt, be careful with the instruction to be used.

Changing any bit other than the IR bit

If while executing an instruction, a request for an interrupt controlled by the register being modified occurs, the IR bit in the register may not be set to “1” (interrupt requested), with the result that the interrupt request is ignored. If such a situation presents a problem, use the instructions shown below to modify the register.

Usable instructions: AND, OR, BCLR, BSET

Changing the IR bit

Depending on the instruction used, the IR bit may not always be cleared to “0” (interrupt not requested). Therefore, be sure to use the MOV instruction to clear the IR bit.

(3) When using the I flag to disable an interrupt, refer to the sample program fragments shown below as you set the I flag. (Refer to (2) for details about rewrite the interrupt control registers in the sample program fragments.)

Examples 1 through 3 show how to prevent the I flag from being set to “1” (interrupts enabled) before the interrupt control register is rewritten, due to the internal bus and the instruction queue buffer timing.

Example 1: Using the NOP instruction to keep the program waiting until the interrupt control register is modified

```
INT_SWITCH1:
        FCLR    I               ; Disable interrupts
        AND.B   #00h, 0055h     ;Set the TA0IC register to 00₁₆
        NOP                     ;
        NOP
        FSET    I               ; Enable interrupts
```

The number of NOP instruction is as follows.
PM20 = 1 (1 wait) : 2, PM20 = 0 (2 waits): 3

Example 2:Using the dummy read to keep the FSET instruction waiting

```
INT_SWITCH2:
        FCLR    I               ; Disable interrupts
        AND.B   #00h, 0055h     ; Set the TA0IC register to 00₁₆
        MOV.W   MEM, R0         ; Dummy read
        FSET    I               ; Enable interrupts
```

Example 3:Using the POPC instruction to changing the I flag

```
INT_SWITCH3:
        PUSHC   FLG
        FCLR    I               ; Disable interrupts
        AND.B   #00h, 0055h     ; Set the TA0IC register to 00₁₆
        POPC    FLG             ; Enable interrupts
```

19.5.7 Watchdog Timer Interrupt

Initialize the watchdog timer after the watchdog timer interrupt occurs.

Downloaded from Arrow.com.

19.6 DMAC

19.6.1 Write to DMAE Bit in DMiCON Register

When both of the conditions below are met, follow the steps below.

Conditions

- The DMAE bit is set to “1” again while it remains set (DMAi is in an active state).
- A DMA request may occur simultaneously when the DMAE bit is being written.

Step 1: Write “1” to the DMAE bit and DMAS bit in DMiCON register simultaneously(*1).
Step 2: Make sure that the DMAi is in an initial state(*2) in a program.
If the DMAi is not in an initial state, the above steps should be repeated.

Notes:

1. The DMAS bit remains unchanged even if “1” is written. However, if “0” is written to this bit, it is set to “0” (DMA not requested). In order to prevent the DMAS bit from being modified to “0”, “1” should be written to the DMAS bit when “1” is written to the DMAE bit. In this way the state of the DMAS bit immediately before being written can be maintained.
   Similarly, when writing to the DMAE bit with a read-modify-write instruction, “1” should be written to the DMAS bit in order to maintain a DMA request which is generated during execution.

2. Read the TCRi register to verify whether the DMAi is in an initial state. If the read value is equal to a value which was written to the TCRi register before DMA transfer start, the DMAi is in an initial state. (If a DMA request occurs after writing to the DMAE bit, the value written to the TCRi register is “1”.) If the read value is a value in the middle of transfer, the DMAi is not in an initial state.

Downloaded from Arrow.com.

19.7 Timer

19.7.1 Timer A

19.7.1.1 Timer A (Timer Mode)

1. The timer remains idle after reset. Set the mode, count source, counter value, etc. using the TAiMR (i = 0 to 4) register and the TAi register before setting the TAiS bit in the TABSR register to “1” (count starts).
   Always make sure the TAiMR register is modified while the TAiS bit remains “0” (count stops) regardless whether after reset or not.

2. While counting is in progress, the counter value can be read out at any time by reading the TAi register. However, if the TAi register is read at the same time the counter is reloaded, the read value is always “FFFF$_{16}$”. If the TAi register is read after setting a value in it, but before the counter starts counting, the read value is the one that has been set in the register.

3. If a low-level signal is applied to the $\overline{SD}$ pin when the IVPCR1 bit in the TB2SC register is set to “1” (three-phase output forcible cutoff by input on $\overline{SD}$ pin enabled), the TA1$_{OUT}$, TA2$_{OUT}$ and TA4$_{OUT}$ pins go to a high-impedance state.

19.7.1.2 Timer A (Event Counter Mode)

1. The timer remains idle after reset. Set the mode, count source, counter value, etc. using the TAiMR (i = 0 to 4) register, the TAi register, the UDF register, the TAZIE, TA0TGL and TA0TGH bits in the ONSF register and the TRGSR register before setting the TAiS bit in the TABSR register to “1” (count starts).
   Always make sure the TAiMR register, the UDF register, the TAZIE, TA0TGL and TA0TGH bits and the TRGSR register are modified while the TAiS bit remains “0” (count stops) regardless whether after reset or not.

2. While counting is in progress, the counter value can be read out at any time by reading the TAi register. However, if the TAi register is read at the same time the counter is reloaded, the read value is always “FFFF$_{16}$” when the timer counter underflows and “0000$_{16}$” when the timer counter overflows. If the TAi register is read after setting a value in it, but before the counter starts counting, the read value is the one that has been set in the register.

3. If a low-level signal is applied to the $\overline{SD}$ pin when the IVPCR1 bit in the TB2SC register is set to “1” (three-phase output forcible cutoff by input on $\overline{SD}$ pin enabled), the TA1$_{OUT}$, TA2$_{OUT}$ and TA4$_{OUT}$ pins go to a high-impedance state.

Downloaded from Arrow.com.

19.7.1.3 Timer A (One-shot Timer Mode)

1. The timer remains idle after reset. Set the mode, count source, counter value, etc. using the TAiMR (i = 0 to 4) register, the TAi register, the TA0TGL and TA0TGH bits in the ONSF register and the TRGSR register before setting the TAiS bit in the TABSR register to “1” (count starts).
   Always make sure the TAiMR register, the TA0TGL and TA0TGH bits and the TRGSR register are modified while the TAiS bit remains “0” (count stops) regardless whether after reset or not.

2. When setting TAiS bit to “0” (count stop), the following occur:
   • The counter stops counting and the content of reload register is reloaded.
   • TAi$_{OUT}$ pin outputs “L”.
   • After one cycle of the CPU clock, the IR bit in the TAiIC register is set to “1” (interrupt request).

3. Output in one-shot timer mode synchronizes with a count source internally generated. When the external trigger has been selected, a maximun delay of one cycle of the count source occurs between the trigger input to TAi$_{IN}$ pin and output in one-shot timer mode.

4. The IR bit is set to “1” when timer operation mode is set with any of the following procedures:
   • Select one-shot timer mode after reset.
   • Change the operation mode from timer mode to one-shot timer mode.
   • Change the operation mode from event counter mode to one-shot timer mode.
   To use the timer Ai interrupt (the IR bit), set the IR bit to “0” after the changes listed above have been made.

5. When a trigger occurs while the timer is counting, the counter reloads the reload register value, and continues counting after a second trigger is generated and the counter is decremented once. To generate a trigger while counting, space more than one cycle of the timer count source from the first trigger and generate again.

6. When selecting the external trigger for the count start conditions in timer A one-shot timer mode, do generate an external trigger 300ns before the count value of timer A is set to “0000$_{16}$”. The one-shot timer does not continue counting and may stop.

7. If a low-level signal is applied to the $\overline{SD}$ pin when the IVPCR1 bit in the TB2SC register is set to “1” (three-phase output forcible cutoff by input on $\overline{SD}$ pin enabled), the TA1$_{OUT}$, TA2$_{OUT}$ and TA4$_{OUT}$ pins go to a high-impedance state.

Downloaded from Arrow.com.

19.7.1.4 Timer A (Pulse Width Modulation Mode)

1. The timer remains idle after reset. Set the mode, count source, counter value, etc. using the TAiMR (i = 0 to 4) register, the TAi register, the TA0TGL and TA0TGH bits in the ONSF register and the TRGSR register before setting the TAiS bit in the TABSR register to “1” (count starts).
   Always make sure the TAiMR register, the TA0TGL and TA0TGH bits and the TRGSR register are modified while the TAiS bit remains “0” (count stops) regardless whether after reset or not.

2. The IR bit is set to “1” when setting a timer operation mode with any of the following procedures:
   • Select the PWM mode after reset.
   • Change an operation mode from timer mode to PWM mode.
   • Change an operation mode from event counter mode to PWM mode.
   To use the timer Ai interrupt (interrupt request bit), set the IR bit to “0” by program after the above listed changes have been made.

3. When setting TAiS register to “0” (count stop) during PWM pulse output, the following action occurs:
   • Stop counting.
   • When TAiOUT pin is output “H”, output level is set to “L” and the IR bit is set to “1”.
   • When TAiOUT pin is output “L”, both output level and the IR bit remains unchanged.

4. If a low-level signal is applied to the $\overline{SD}$ pin when the IVPCR1 bit in the TB2SC register is set to “1” (three-phase output forcible cutoff by input on $\overline{SD}$ pin enabled), the TA1OUT, TA2OUT and TA4OUT pins go to a high-impedance state.

Downloaded from Arrow.com.

19.7.2 Timer B

19.7.2.1 Timer B (Timer Mode)

1. The timer remains idle after reset. Set the mode, count source, counter value, etc. using the TBiMR (i = 0 to 2) register and TBi register before setting the TBiS bit in the TABSR register to “1” (count starts).
   Always make sure the TBiMR register is modified while the TBiS bit remains “0” (count stops) regardless whether after reset or not.

2. The counter value can be read out at any time by reading the TBi register. However, if this register is read at the same time the counter is reloaded, the read value is always “FFFF16.” If the TBi register is read after setting a value in it but before the counter starts counting, the read value is the one that has been set in the register.

19.7.2.2 Timer B (Event Counter Mode)

1. The timer remains idle after reset. Set the mode, count source, counter value, etc. using the TBiMR (i = 0 to 2) register and TBi register before setting the TBiS bit in the TABSR register to “1” (count starts).
   Always make sure the TBiMR register is modified while the TBiS bit remains “0” (count stops) regardless whether after reset or not.

2. The counter value can be read out at any time by reading the TBi register. However, if this register is read at the same time the counter is reloaded, the read value is always “FFFF16.” If the TBi register is read after setting a value in it but before the counter starts counting, the read value is the one that has been set in the register.

Downloaded from Arrow.com.

19.7.2.3 Timer B (Pulse Period/pulse Width Measurement Mode)

1. The timer remains idle after reset. Set the mode, count source, etc. using the TBiMR (i = 0 to 2) register before setting the TBiS bit in the TABSR register to “1” (count starts).
   Always make sure the TBiMR register is modified while the TBiS bit remains “0” (count stops) regardless whether after reset or not. To clear the MR3 bit to “0” by writing to the TBiMR register while the TBiS bit is set to “1” (count starts), be sure to set the TM0D0, TM0D1, MR0, MR1, TCK0 and TCK1 bits to the same value as previously written and the MR2 bit to "0".

2. The IR bit in the TBiIC register (i=0 to 2) goes to “1” (interrupt request), when an effective edge of a measurement pulse is input or timer Bi is overflowed. The factor of interrupt request can be determined by use of the MR3 bit in the TBiMR register within the interrupt routine.

3. If the source of interrupt cannot be identified by the MR3 bit such as when the measurement pulse input and a timer overflow occur at the same time, use another timer to count the number of times timer B has overflowed.

4. To set the MR3 bit to “0” (no overflow), set TBiMR register with setting the TBiS bit to “1” and counting the next count source after setting the MR3 bit to “1” (overflow).

5. Use the IR bit in the TBiIC register to detect only overflows. Use the MR3 bit only to determine the interrupt factor within the interrupt routine.

6. When the count is started and the first effective edge is input, an indeterminate value is transferred to the reload register. At this time, timer Bi interrupt request is not generated.

7. The value of the counter is indeterminate at the beginning of a count. MR3 may be set to “1” and timer Bi interrupt request may be generated between the count start and an effective edge input.

8. For pulse width measurement, pulse widths are successively measured. Use program to check whether the measurement result is an “H” level width or an “L” level width.

19.7.3 Three-phase Motor Control Timer Function

When the IVPCR1 bit in the TB2SC register is set to 1 (three-phase output forced cutoff by SD pin input (high-impedance) enabled), the INV03 bit in the INVC0 register is set to 1 (three-phase motor control timer output enabled), and a low-level ("L") signal is applied to the $\overline{SD}$ pin while a three-phase PWM signal is output, the MCU is forced to cutoff and pins U, $\overline{U}$, V, $\overline{V}$, W, and $\overline{W}$ are placed in a high-impedance state and the INV03 bit is set to 0 (three-phase motor control timer output disabled).
To resume the three-phase PWM signal output from pins U, $\overline{U}$, V, $\overline{V}$, W, and $\overline{W}$, set the INV03 bit to 1 and the IVPCR1 bit to 0 (three-phase output forced cutoff disabled) after the $\overline{SD}$ pin level becomes "H". Then set the IVPCR1 bit to 1 (three-phase output forced cutoff enabled) in order to enable the three-phase output forced cutoff function by input to the SD pin again.
The INV03 bit cannot be set to 1 while an "L" signal is input to the $\overline{SD}$ pin. To set the INV03 bit to 1 after forcible cutoff, write 1 to the INV03 bit and read the bit to ensure that it is set to 1 by program. Then set the IVPCR1 bit to 1 after setting it to 0.

Downloaded from Arrow.com.

19.8 Serial I/O

19.8.1 Clock-Synchronous Serial I/O

19.8.1.1 Transmission/reception

1. With an external clock selected, and choosing the $\overline{RTS}$ function, the output level of the $\overline{RTSi}$ pin goes to “L” when the data-receivable status becomes ready, which informs the transmission side that the reception has become ready. The output level of the $\overline{RTSi}$ pin goes to “H” when reception starts. So if the $\overline{RTSi}$ pin is connected to the $\overline{CTSi}$ pin on the transmission side, the circuit can transmit and receive data with consistent timing. With the internal clock, the $\overline{RTS}$ function has no effect.
2. If a low-level signal is applied to the $\overline{SD}$ pin when the IVPCR1 bit in the TB2SC register is set to “1” (three-phase output forcible cutoff by input on $\overline{SD}$ pin enabled), the $P7_3/\overline{RTS2}$/TxD1(when the U1MAP bit in PACR register is “1”) and CLK_2 pins go to a high-impedance state.

19.8.1.2 Transmission

When an external clock is selected, the conditions must be met while if the CKPOL bit in the UiC0 register is set to “0” (transmit data output at the falling edge and the receive data taken in at the rising edge of the transfer clock), the external clock is in the high state; if the CKPOL bit in the UiC0 register is set to “1” (transmit data output at the rising edge and the receive data taken in at the falling edge of the transfer clock), the external clock is in the low state.

- The TE bit in the UiC1 register is set to “1” (transmission enabled)
- The TI bit in the UiC1 register is set to “0” (data present in UiTB register)
- If $\overline{CTS}$ function is selected, input on the $\overline{CTSi}$ pin is “L”

19.8.1.3 Reception

1. In operating the clock-synchronous serial I/O, operating the transmitter generates a clock for the receiver shift register. Fix settings for transmission even when using the device only for reception. Dummy data is output to the outside from the TxDi pin when receiving data.
2. When an internal clock is selected, set the TE bit in the UiC1 register (i = 0 to 2) to 1 (transmission enabled) and write dummy data to the UiTB register, and the clock for the receiver shift register will thereby be generated. When an external clock is selected, set the TE bit to "1" and write dummy data to the UiTB register, and the clock for the receiver shift register will be generated when the external clock is fed to the CLKi input pin.
3. When successively receiving data, if all bits of the next receive data are prepared in the UARTi receive register while the RE bit in the UiC1 register (i = 0 to 2) is set to “1” (data present in the UiRB register), an overrun error occurs and the OER bit in the UiRB register is set to “1” (overrun error occurred). In this case, because the content of the UiRB register is indeterminate, a corrective measure must be taken by programs on the transmit and receive sides so that the valid data before the overrun error occurred will be retransmitted. Note that when an overrun error occurred, the IR bit in the SiRIC register does not change state.
4. To receive data in succession, set dummy data in the lower-order byte of the UiTB register every time reception is made.
5. When an external clock is selected, make sure the external clock is in high state if the CKPOL bit is set to “0”, and in low state if the CKPOL bit is set to “1” before the following conditions are met:

- Set the RE bit in the UiC1 register to “1” (reception enabled)
- Set the TE bit in the UiC1 register to “1” (transmission enabled)
- Set the TI bit in the UiC1 register to “0” (data present in the UiTB register)

Downloaded from Arrow.com.

19.8.2 Serial I/O (UART Mode)

19.8.1.1 Special Mode 1 (I^2C bus Mode)

When generating start, stop and restart conditions, set the STSPSEL bit in the U2SMR4 register to “0” and wait for more than half cycle of the transfer clock before setting each condition generate bit (STAREQ, RSTAREQ and STPREQ) from “0” to “1”.

19.8.1.2 Special Mode 2

If a low-level signal is applied to the P8$_5$/$\overline{\text{NMI}}$/$\overline{\text{SD}}$ pin when the IVPCR1 bit in the TB2SC register is set to "1" (three-phase output forcible cutoff by input on $\overline{\text{SD}}$ pin enabled), the P7$_3$/RTS$_2$/TxD1(when the U1MAP bit in PACR register is “1”) and CLK$_2$ pins go to a high-impedance state.

19.8.1.3 Special Mode 4 (SIM Mode)

A transmit interrupt request is generated by setting the U2IRS bit in the U2C1 register to “1” (transmission complete) and U2ERE bit to “1” (error signal output) after reset. Therefore, when using SIM mode, be sure to clear the IR bit to “0” (no interrupt request) after setting these bits.

Downloaded from Arrow.com.

19.9 A/D Converter

1. Set ADCON0 (except bit 6), ADCON1 and ADCON2 registers when A/D conversion is stopped (before a trigger occurs).

2. When the VCUT bit in the ADCON1 register is changed from “0” (Vref not connected) to “1” (V_{REF} connected), start A/D conversion after waiting 1 µs or longer.

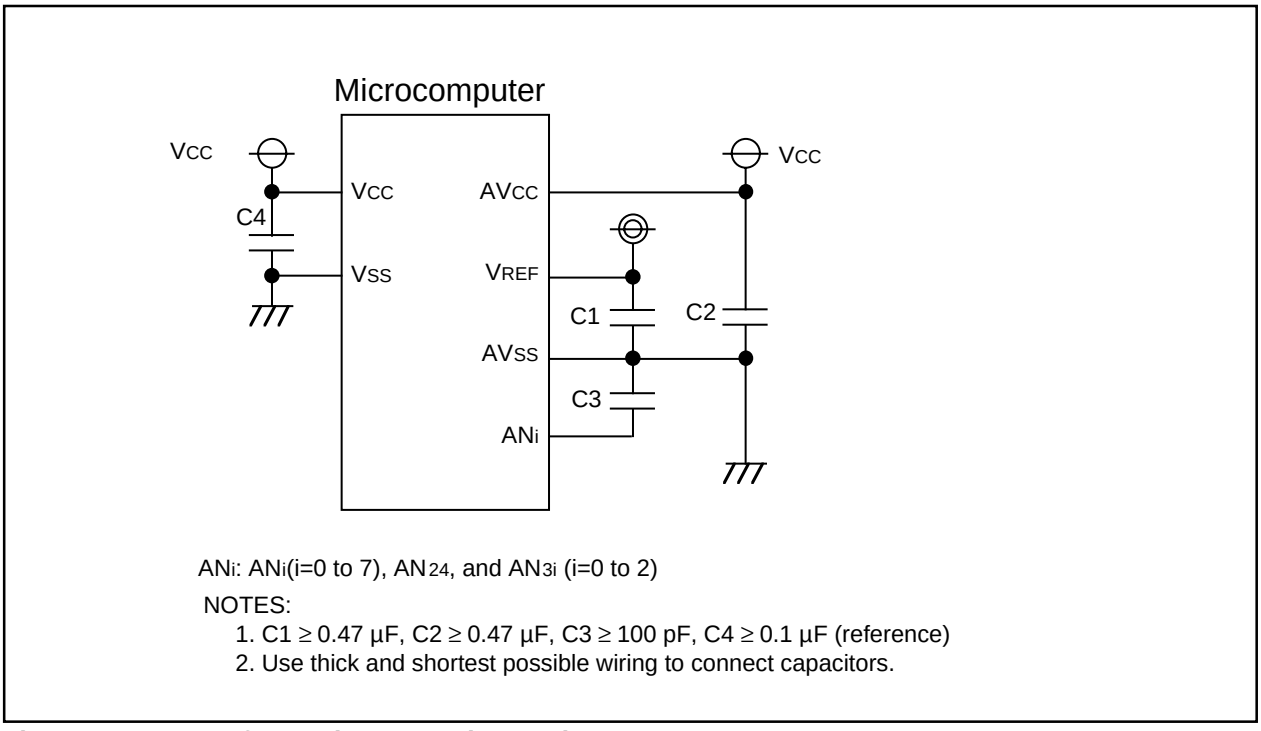
3. To prevent noise-induced device malfunction or latchup, as well as to reduce conversion errors, insert capacitors between the AV_{CC}, V_{REF}, and analog input pins (AN_i(i=0 to 7), AN_{24}, AN_{3i}(i=0 to 2)) each and the AV_{SS} pin. Similarly, insert a capacitor between the V_{CC} pin and the V_{SS} pin. Figure 19.4 is an example connection of each pin.

4. Make sure the port direction bits for those pins that are used as analog inputs are set to “0” (input mode). Also, if the TGR bit in ADCON0 register is set to "1" (external trigger), make sure the port direction bit for the $\overline{ADTRG}$ pin is set to “0” (input mode).

5. When using key input interrupts, do not use any of the four AN_4 to AN_7 pins as analog inputs. (A key input interrupt request is generated when the A/D input voltage goes low.)

6. The ϕ_{AD} frequency must be 10 MHz or less (12 MHz or less in M16C/26B). Without sample-and-hold function, limit the ϕ_{AD} frequency to 250kHz or more. With the sample and hold function, limit the ϕ_{AD} frequency to 1MHz or more.

7. When changing an A/D operation mode, select analog input pin again in the CH2 to CH0 bits in the ADCON0 register and the SCAN1 to SCAN0 bits in the ADCON1 register.



AN_i: AN_i(i=0 to 7), AN_{24}, and AN_{3i} (i=0 to 2)

NOTES:
1. C1 ≥ 0.47 µF, C2 ≥ 0.47 µF, C3 ≥ 100 pF, C4 ≥ 0.1 µF (reference)
2. Use thick and shortest possible wiring to connect capacitors.

Figure 19.3. Use of capacitors to reduce noise

Downloaded from Arrow.com.

8. If the CPU reads the A/D register i (i = 0 to 7) at the same time the conversion result is stored in the A/D register i after completion of A/D conversion, an incorrect value may be stored in the A/D register i. This problem occurs when a divide-by-n clock derived from the main clock or a subclock is selected for CPU clock.
   - When operating in one-shot mode, single-sweep mode, simultaneous sample sweep mode, delayed trigger mode 0 or delayed trigger mode 1
     Check to see that A/D conversion is completed before reading the target A/D register i. (Check the IR bit in the ADIC register to see if A/D conversion is completed.)
   - When operating in repeat mode or repeat sweep mode 0 or 1
     Use the main clock for CPU clock directly without dividing it.

9. If A/D conversion is forcibly terminated while in progress by setting the ADST bit in the ADCON0 register to “0” (A/D conversion halted), the conversion result of the A/D converter is indeterminate. The contents of A/D register i irrelevant to A/D conversion may also become indeterminate. If while A/D conversion is underway the ADST bit is cleared to “0” in a program, ignore the values of all A/D register i.

10. When setting the ADST bit in the ADCON register to "0" to terminate a conversion forcefully by the program in single sweep conversion mode, A/D delayed trigger mode 0 and A/D delayed trigger mode 1 during A/D conversion operation, the A/D interrupt request may be generated. If this causes a problem, set the ADST bit to "0" after the interrupt is disabled.

Downloaded from Arrow.com.

19.10 Programmable I/O Ports

1. If a low-level signal is applied to the $\overline{SD}$ pin when the IVPCR1 bit in the TB2SC register is set to “1” (three-phase output forcible cutoff by input on $\overline{SD}$ pin enabled), the $P7_2$ to $P7_5$, $P8_0$ and $P8_1$ pins go to a high-impedance state.

2. The input threshold voltage of pins differs between programmable input/output ports and peripheral functions.
   Therefore, if any pin is shared by a programmable input/output port and a peripheral function and the input level at this pin is outside the range of recommended operating conditions V_{IH} and V_{IL} (neither “high” nor “low”), the input level may be determined differently depending on which side—the programmable input/output port or the peripheral function—is currently selected.

3. When the INV03 bit in the INVC0 register is "1"(three-phase motor control timer output enabled), an "L" input on the $P8_5$ /$\overline{NMI}$/$\overline{SD}$ pin, has the following effect:

   •When the IVPCR1 bit in the TB2SC register is set to “1” (three-phase output forcible cutoff by input on the $\overline{SD}$ pin enabled), the U/ $\overline{U}$/ V/ $\overline{V}$/ W/ $\overline{W}$ pins go to a high-impedance state.
   •When the IVPCR1 bit is set to “0” (three-phase output forcible cutoff by input on $\overline{SD}$ pin disabled), the U/ $\overline{U}$/ V/ $\overline{V}$/ W/ $\overline{W}$ pins go to a normal port.

   Therefore, the $P8_5$ pin can not be used as programmable I/O port when the INV03 bit is set to "1".
   When the $\overline{SD}$ function isn't used, set $PD8_5$ to “0” (Input) and pull the $P8_5$ /$\overline{NMI}$/$\overline{SD}$ pin to “H” externally.

Downloaded from Arrow.com.

19.11 Electric Characteristic Differences Between Mask ROM and Flash Memory Version Microcomputers

Flash memory version and mask ROM version may have different characteristics, operating margin, noise tolerated dose, noise width dose in electrical characteristics due to internal ROM, different layout pattern, etc. When switching to the mask ROM version, conduct equivalent tests as system evaluation tests conducted in the flash memory version.

Downloaded from Arrow.com.

19.12 Mask ROM Version

19.12.1 Internal ROM area

When using the masked ROM version, write nothing to internal ROM area. Writing to the area may increase power consumption.

19.12.2 Reserve bit

The b3 to b0 in address $0FFFFF_{16}$ are reserved bits. Set these bits to “1111_2”.

Downloaded from Arrow.com.

19.13 Flash Memory Version

19.13.1 Functions to Inhibit Rewriting Flash Memory

ID codes are stored in addresses 0FFFDF16, 0FFFE316, 0FFFEB16, 0FFFEF16, 0FFFF316, 0FFFF716, and 0FFFFB16. If wrong data is written to these addresses, the flash memory cannot be read or written in standard serial I/O mode.

The ROMCP register is mapped in address 0FFFFF16. If wrong data is written to this address, the flash memory cannot be read or written in parallel I/O mode.

In the flash memory version of microcomputer, these addresses are allocated to the vector addresses (H) of fixed vectors.The b3 to b0 in address 0FFFFF16 are reserved bits. Set these bits to “11112”.

19.13.2 Stop mode

When the microcomputer enters stop mode, execute the instruction which sets the CM10 bit to “1”(stop mode) after setting the FMR01 bit to “0”(CPU rewrite mode disabled) and disabling the DMA transfer.

19.13.3 Wait mode

When the microcomputer enters wait mode, excute the WAIT instruction after setting the FMR01 bit to “0”(CPU rewrite mode disabled).

19.13.4 Low power dissipation mode, on-chip oscillator low power dissipation mode

If the CM05 bit is set to “1” (main clock stop), the following commands must not be executed.

- Program
- Block erase

19.13.5 Writing command and data

Write the command code and data at even addresses.

19.13.6 Program Command

Write ‘xx4016’ in the first bus cycle and write data to the write address in the second bus cycle, and an auto program operation (data program and verify) will start. Make sure the address value specified in the first bus cycle is the same even address as the write address specified in the second bus cycle.

19.13.7 Operation speed

When CPU clock source is main clock, before entering CPU rewrite mode (EW0 or EW1 mode), select 10 MHz or less for BCLK using the CM06 bit in the CM0 register and the CM17 to CM16 bits in the CM1 register. Also, when CPU clock is f3(ROC) on-chip oscillator clock, before entering CPU rewrite mode (EW0 or EW1 mode), set the ROCR3 to ROCR2 bits in the ROCR register to “divied by 4” or “divide by 8”. On both cases, set the PM17 bit in the PM1 register to “1” (with wait state).

19.13.8 Instructions prohibited in EW0 Mode

The following instructions cannot be used in EW0 mode because the flash memory’s internal data is referenced: UND instruction, INTO instruction, JMPS instruction, JSRS instruction, and BRK instruction

Downloaded from Arrow.com.

19.13.9 Interrupts

EW0 Mode

- Any interrupt which has a vector in the variable vector table can be used, providing that its vector is transferred into the RAM area.
- The $\overline{\text{NMI}}$ and watchdog timer interrupts can be used because the FMR0 register and FMR1 register are initialized when one of those interrupts occurs. The jump addresses for those interrupt service routines should be set in the fixed vector table.

  Because the rewrite operation is halted when a $\overline{\text{NMI}}$ or watchdog timer interrupt occurs, the rewrite program must be executed again after exiting the interrupt service routine.
- The address match interrupt cannot be used because the flash memory's internal data is referenced.

EW1 Mode

- Make sure that any interrupt which has a vector in the relocatable vector table or address match interrupt will not be accepted during the auto program period or auto erase period with erase-suspend function disabled.
- The $\overline{\text{NMI}}$ interrupt can be used because the FMR0 register and FMR1 register are initialized when this interrupt occurs. The jump address for the interrupt service routine should be set in the fixed vector table.

  Because the rewrite operation is halted when a $\overline{\text{NMI}}$ interrupt occurs, the rewrite program must be executed again after exiting the interrupt service routine.

19.13.10 How to access

To set the FMR01, FMR02, FMR11 or FMR16 bit to "1", set the subject bit to "1" immediately after setting to "0". Do not generate an interrupt or a DMA transfer between the instruction to set the bit to "0" and the instruction to set the bit to "1". When the PM24 bit is set to "1" ($\overline{\text{NMI}}$ funciton), apply a high-level ("H") signal to the $\overline{\text{NMI}}$ pin to set those bits.

19.13.11 Writing in the user ROM area

EW0 Mode

- If the power supply voltage drops while rewriting any block in which the rewrite control program is stored, a problem may occur that the rewrite control program is not correctly rewritten and, consequently, the flash memory becomes unable to be rewritten thereafter. In this case, standard serial I/O or parallel I/O mode should be used.

EW1 Mode

- Avoid rewriting any block in which the rewrite control program is stored.

19.13.12 DMA transfer

In EW1 mode, make sure that no DMA transfers will occur while the FMR00 bit in the FMR0 register is set to "0" (during the auto program or auto erase period).

19.13.13 Regarding Programming/Erasure Times and Execution Time

As the number of programming/erasure times increases, so does the execution time for software commands (Program, and Block Erase).

The software commands are aborted by hardware reset 1, hardware reset 2, $\overline{\text{NMI}}$ interrupt, and watchdog timer interrupt. If a software command is aborted by such reset or interrupt, the affected block must be erased before reexecuting the aborted command.

Downloaded from Arrow.com.

19.13.14 Definition of Programming/Erasure Times

"Number of programs and erasure" refers to the number of erasure per block.
If the number of program and erasure is n (n=100 1,000 10,000) each block can be erased n times.
For example, if a 2K byte block A is erased after writing 1 word data 1024 times, each to a different address, this is counted as one program and erasure. However, data cannot be written to the same adrress more than once without erasing the block. (Rewrite prohibited)

19.13.15 Flash Memory Version Electrical Characteristics 10,000 E/W cycle products (U7, U9)

When Block A or B E/W cycles exceed 100, select one wait state per block access. When FMR17 is set to "1", one wait state is inserted per access to Block A or B - regardless of the value of PM17. Wait state insertion during access to all other blocks, as well as to internal RAM, is controlled by PM17 - regardless of the setting of FMR17.
To use the limited number of erasure efficiently, write to unused address within the block instead of rewite. Erase block only after all possible address are used. For example, an 8-word program can be written 128 times before erase becomes necessary.
Maintaining an equal number of erasure between Block A and B will also improve efficiency.
We recommend keeping track of the number of times erasure is used.

19.13.16 Boot Mode

An indeterminate value is sometimes output in the I/O port until the internal power supply becomes stable when "H" is applied to the CNVSS pin and "L" is applied to the $\overline{\text{RESET}}$ pin.
When setting the CNVSS pin to "H", the following procedure is required:

(1) Apply an "L" signal to the $\overline{\text{RESET}}$ pin and the CNVSS pin.
(2) Bring VCC to more than 2.7V, and wait at least 2msec. (Internal power supply stable waiting time)
(3) Apply an "H" signal to the CNVSS pin.
(4) Apply an "H" signal to the $\overline{\text{RESET}}$ pin.

When the CNVSS pin is “H” and RESET pin is “L”, P67 pin is connected to the pull-up resister.

Downloaded from Arrow.com.

19.14 Noise

Connect a bypass capacitor (approximately 0.1µF) across the VCC and VSS pins using the shortest and thicker possible wiring. Figure 19.4 shows the bypass capacitor connection.

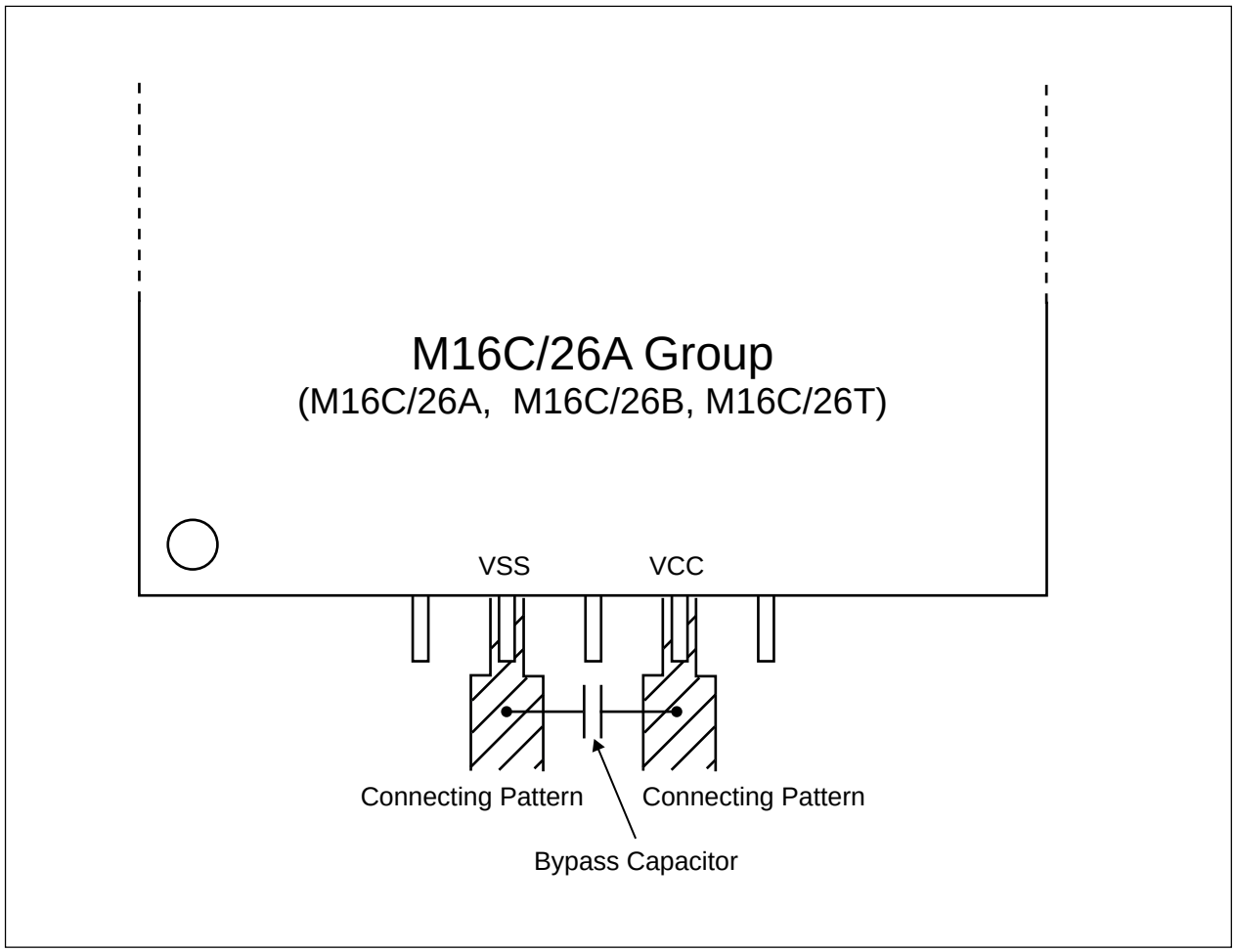


Figure 19.4 Bypass Capacitor Connection

Downloaded from Arrow.com.

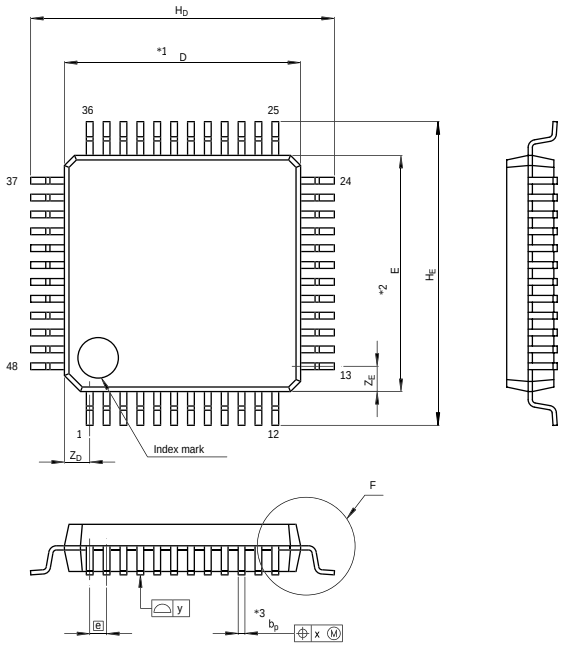
19.15 Instruction for a Device Use

When handling a device, extra attention is necessary to prevent it from crashing during the electrostatic discharge period.

Downloaded from Arrow.com.

Appendix 1. Package Dimensions

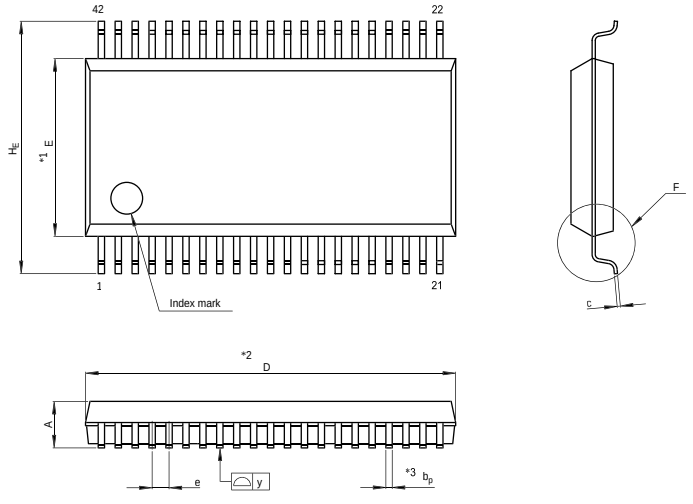
JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LQFP48-7x7-0.50	PLQP0048KB-A	48P6Q-A	0.2g



NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.

Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	6.9	7.0	7.1
E	6.9	7.0	7.1
A_2	—	1.4	—
H_D	8.8	9.0	9.2
H_E	8.8	9.0	9.2
A	—	—	1.7
A_1	0	0.1	0.2
b_p	0.17	0.22	0.27
b_1	—	0.20	—
c	0.09	0.145	0.20
c_1	—	0.125	—
θ	0°	—	8°
e	—	0.5	—
x	—	—	0.08
y	—	—	0.10
Z_D	—	0.75	—
Z_E	—	0.75	—
L	0.35	0.5	0.65
L_1	—	1.0	—

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-SSOP42-8.4x17.5-0.80	PRSP0042GA-B	42P2R-E	0.6g



NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.

Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	17.3	17.5	17.7
E	8.2	8.4	8.6
A_2	—	2.0	—
A	—	—	2.4
A_1	0.05	—	—
b_p	0.25	0.3	0.4
c	0.13	0.15	0.2
θ	0°	—	10°
H_E	11.63	11.93	12.23
e	0.65	0.8	0.95
y	—	—	0.15
L	0.3	0.5	0.7

Downloaded from Arrow.com.

Appendix 2. Functional Difference

Appendix 2.1 Differences between M16C/26A, M16C/26B, and M16C/26T

Item	M16C/26A, M16C/26B	M16C/26T
Main Clock during and after Reset	Oscillating (Default value “0” while and after the CM05 bit is reset.)	Stoped (Default value “1” while and after the CM05 bit is reset.)
Voltage Detection Circuit (Function of 0019_{16}, $001A_{16}$, $001F_{16}$)	Available (VCR1 register, VCR2 register, D4INT register)	Not available (reserved register)
Package	PLQP0048KB-A(48P6Q), PRSP0042GA-B(42P2R)	PLP0048KB-A(48P6Q)

NOTE:

1. Since the emulator between the M16C/26A Group and M16C/29 Group are the same, all functions of M16C/29 are built in the emulator. When evaluating M16C/26A Group, do not access to the SFR which is not built in M16C/26A Group. Refer to Hardware Manual about detail and electrical characteristics.

Downloaded from Arrow.com.

Appendix 2.2 Differences between M16C/26A Group and M16C/26 Group

Item	M16C/26A Group	M16C/26 Group
Clock Generation Circuit	4 circuits (Main clock oscillation circuit, Sub clock oscillation circuit, on-chip oscillator, PLL frequency synthesizer)	3 circuits (Main clock oscillation circuit, Sub clock oscillation circuit, on-chip oscillator)
System Clock Source After Reset (Initial value of the CM21 bit in the CM2 register)	On-chip oscillator (Initial value "1" of CM21 bit)	Main clock (Initial value "0" of CM21 bit)
On-chip Oscillator Clock	Selectable (8MHz/1MHz/500KHz)	Fixed (1MHz)
PACR2 to PACR0 in the PACR register	Necessary to set after reset 48pin:"100$_2$", 42pin:"001$_2$"	No PACR register
IFSR20 bit in the IFSR2A register	Necessary to set to "1" after reset	No IFSR2A register
External Interrupt	8 causes ($\overline{INT2}$ added)	7 causes
13 pin (48-pin version) Function	P8$_4$/$\overline{INT2}$/ZP	IV$_{CC}$
P7$_0$, P7$_1$	N-ch open drain output and CMOS output are selectable by S/W	N-ch open drain output
A/D Input Pin (48-pin version)	12 channels	8 channels
A/D operation Mode	8 modes (single, repeat, single sweep, repeat sweep mode 0, repeat sweep mode 1, simultaneous sampling, delayed trigger mode 0, delayed trigger mode 1) 1 shunt current measurement function is available	5 modes (single, repeat, single sweep, repeat sweep mode 0, repeat sweep mode 1)
Timer B Operation Mode	5 modes (timer, event counter, pulse periods measurement, pulse width measurment, A/D trigger) 1 shunt current measurement function is available	4 modes (timer, event counter, pulse periods measurement, pulse width measurment)
CRC Calculation	Available (compatible to CRC-CCITT and CRC-16 methods)	Not available
Three-phase motor Control	•Waveform output/Switching port output by software is enabled •Position data retention function	•Waveform output/Switching port output by software is disabled •No position data retention function
Digital Debounce Function	This function is in the $\overline{NMI}$/$\overline{SD}$ pin and $\overline{INT5}$ pin	Not available
3 pin (48-pin version) function	P9$_0$/CLK$_{OUT}$/TB0$_{IN}$/AN3$_0$ (CLK$_{OUT}$: f1, f8, f32, and f$_C$ output)	P9$_0$/TB0$_{IN}$
UART1 Compatible pin	Switching to P6$_4$ to P6$_7$ or P7$_0$ to P7$_3$ is enabled	P6$_4$ to P6$_7$
Flash Memory Protect Function	Protection to blocks 0, 1 by FMR02 bit Protection to the blocks 0 to 3 by FMR16 bit	Protection to blocks 0,1 by FMR02 bit
Package	PLQP0048KB-A(48P6Q), PRSP0042GA-B(42P2R)	PLQP0048KB-A(48P6Q)

NOTE:

1. Since the emulator between the M16C/26A Group and M16C/29 Group are the same, all functions of M16C/29 are built in the emulator. When evaluating M16C/26A Group, do not access to the SFR which is not built in M16C/26A Group. Refer to Hardware Manual about detail and electrical characteristics.

Downloaded from Arrow.com.

Register Index

A

AD0 to AD7 **184**
ADCON0 to ADCON2 **182**
ADIC **67**
ADSTAT0 **184**
ADTRGCON 183
AIER **79**

B

BCNIC **67**

C

CM0 **40**
CM1 **41**
CM2 **42**
CPSRF **96**, **110**
CRCD **214**
CRCIN **214**
CRCMR **214**
CRCSAR **214**

D

D4INT **30**
DAR0 **86**
DAR1 **86**
DM0CON **85**
DM0IC **67**
DM0SL **84**
DM1CON **85**
DM1IC **67**
DM1SL **85**
DTT **121**

F

FMR0 **241**
FMR1 **241**
FMR4 **242**

I

ICTB2 **121**
IDB0 **121**
IDB1 **121**
IFSR **68**, **76**
IFSR2A **68**
INT0IC to INT2IC **67**
INT3IC **67**
INT4IC **67**
INT5IC **67**
INVC0 **119**
INVC1 **120**

K

KUPIC **67**

N

NDDR **227**

O

ONSF **96**

P

P0 to P13 **224**
P17DDR **227**
PACR **139**, **226**
PCLKR **43**
PCR **226**
PD0 to PD13 **223**
PDRF **129**
PFCR **131**
PLC0 **44**
PM0 **35**
PM1 **35**
PM2 **36**, **43**
PRCR **60**
PUR0 to PUR2 **225**

R

RMAD0 **79**
RMAD1 **79**
ROCR **41**
ROMCP **236**

S

S0RIC to S2RIC **67**
S0TIC to S2TIC **67**
SAR0 **86**
SAR1 **86**

Downloaded from Arrow.com.

T

TA0 to TA4 **95**
TA0IC to TA4IC **67**
TA0MR to TA4MR **94**
TA1 **122**
TA11 **122**
TA1MR **125**
TA2 **122**
TA21 **122**
TA2MR **125**
TA2MR to TA4MR **101**
TA4 **122**
TA41 **122**
TA4MR **125**
TABSR **95**, **110**, **124**
TAiMR **99**, **106**
TB0 to TB5 **110**
TB0IC TO TB2IC **67**
TB0MR to TB5MR **109**
TB2 **124**
TB2MR **125**
TB2SC **123**, **185**
TCR0 **86**
TCR1 **86**
TPRC **131**
TRGSR **96**, **124**

U

U0BRG to U2BRG **136**
U0C0 to U2C0 **138**
U0C1 to U2C1 **139**
U0MR to U2MR **137**
U0RB to U2RB **136**
U0TB to U2TB **136**
U2SMR **140**
U2SMR2 **140**
U2SMR3 **141**
U2SMR4 **141**
UCON **138**
UDF **95**

V

VCR1 **30**
VCR2 **30**

W

WDC **81**
WDTS **81**

Downloaded from Arrow.com.

REVISION HISTORY	M16C/26A Group (M16C/26A, M16C/26B, M16C/26T) Hardware Manual

Rev.	Date	Description	
		Page	Summary
2.00	Feb.15,07	-	M16C/26B newly added, word standardized: on-chip oscillator, development tool
			Overview
		1	•Description partially deleted
		2 - 3	•**1.2 Performance Outline** modified
		4 - 5	•**Figure 1.1 and 1.2 Block Diagrams** updated
		6	•**1.4 Product List** updated
		7	•**Figure 1.3 Product Numbering System** updated
		8	•**Tables 1.7 to 1.10 Product Codes** updated
		12, 14	•**Tables 1.11 and 1.12 Pin Characteristics** newly added
		15 - 16	•**Tables 1.13 Pin Description** newly added
			SFRs
		20	•**Table 4.1 SFR Information(1)** Note about WDC register is deleted
		22	•**Table 4.3 SFR Information(3)** Value after reset for ROCR register modified
		23	•**Table 4.4 SFR Information(4)** Note 2 added to IFSR2A register
			Reset
		28	•**Figure 5.1.1.2. Reset Sequence** Vcc line and ROC line are modified
		29	•**Figure 5.5.1. Voltage Detection Circuit Block** WDC register's block is deleted
			Processor Mode
		35	•**Figure 6.1 PM1 Register** Note 2 partially added
		36	•**Figure 6.2 PM2 Register** newly added
		37	•**Figure 6.3 Bus Block Diagram** and **Table 6.1 Accessible Area and Bus Cycle** newly added
			C**lock Generation Circuit**
		41	•**Figure 7.4 ROCR Register** modified
		43	•**Figure 7.6 PM2 Registe**r Notes 2, 5, 6 modified
		45 - 46	•**Figure 7.1.1 and 7.2.1 Examples of Main Clock Connection Circuit** updated
		47	•7**.4 PLL Clock** Description modified for M16C/26B
			•**Table 7.4.1 Example for Setting PLL Clock Frequencies** Note 1 modified
		50	•**7.6.1 Normal Operation Mode** Description modifie**d**
		51	•**Table 7.6.1.1 Setting Clock Related Bit and Modes** modified
		54	•**Figure 7.6.1 State Transition to Stop Mode and Wait Mode** modified
		55	•**Figure 7.6.1.1. State Transtion in Normal Mode** modified
		56	•**Table 7.6.1 Allowed Transition and Setting modified**, Notes 1 and 2 modified
		59	•**Figure 7.8.3.1 Procedure to Switch Clock Source From On-chip Oscillator to Main Clock** upadated
			Protection
		60	•Description partially modified
			Interrupt
		76	•**9.6 $\overline{\text{INT}}$ Interrupt** Description partially added

Downloaded from Arrow.com.

REVISION HISTORY	M16C/26A Group (M16C/26A, M16C/26B, M16C/26T) Hardware Manual

Rev.	Date	Description	
		Page	Summary
		77	•**9.7 $\overline{\text{NMI}}$ Interrupt** Description partially added
		78	•**Table 9.9.1 Value of the PC that is saved to the stack area when an address match interrupt request is accepted** modified, note 1 added
			Watchdog Timer
		-	•Section of Cold Start/Warm Start deleted
		80	•Description partially added
			•**Figure 10.1 Watchdog Timer Block Diagram** partially modified
		81	•**Figure 10.2 WDC Register and WDTS Register** notes deleted, WDC5 bit deleted
			•**10.1 Count source protective mode** description partially added
			Timer
		108	•Description about A/D trigger mode modified
			•**Figure 12.2.1 Timber B Block Diagram** A/D trigger mode added
		115	•**12.2.4 A/D Trigger Mode** Description modified
		121	•**Figure 12.3.4 IDB0 Register, IDB1 Register, DTT Register, and ICTB2 Register** modified
		123	•**Figure 12.3.6 TB2SC Register** modified, note 4 added
		131	•**Figure 12.3.2.2 TPRC Register** bit map modified
			Serial I/O
		133	•**Figure 13.1.1 Block Diagram of UARTi (i = 0 to 2)** PLL clock added
		136	•**Figure 13.1.4 U0TB to U2TB Registers, U0RB to U2RB Registers, U0BRG to U2BRG Registers** modified, note 3 for UiBRG added
		138	•**Figure 13.1.6 U0C0 to U2C0 Registers** Note 2 modified, note 7 added
		139	•**Figure 13.1.7 PACR Register** note 1 modified
		142	•**Table 13.1.1.1 Clock Synchronous Serial I/O Mode Specification** note 2 modified
		145	•**Figure 13.1.1.1 Typical Transmit/Receive Timings in Clock Synchronous Serial I/O Mode** partially modified
		150	•**Table 13.1.2.1 UART Mode Specifications** Note 1 modified
		154	•**Figure 13.1.2.2 Receive Operation** Figure modified
		158	•**Table 13.1.3.1 I^2C bus Mode Specifications** note 2 modified
		168	•**Table 13.1.4.1 Special Mode 2 Specifications** note 2 modified
		175	•**Table 13.1.6.1 SIM Mode Specifications** note 1 modified
		177	•**Figure 13.1.6.1 Transmit and Receive Timing in SIM Mode** timing modified
			A/D Converter
		180	•**Table 14.1 A/D Converter Performance** note 2 partially added
		183	•**Table 14.2 A/D Conversion Frequency Select** note 1 partially added
		205	•**Table 14.1.8.1 Delayed Trigger Mode 1 Specifications** note 1 modified
		212	•**Figure 14.5.1 Analog Input Pin and External Sensor Equivalent Circuit** note

Downloaded from Arrow.com.

REVISION HISTORY	M16C/26A Group (M16C/26A, M16C/26B, M16C/26T) Hardware Manual

Rev.	Date	Description	
		Page	Summary
			1 added
			CRC Calculation Circuit
		213	•**15.1 CRC Snoop** Description partially modified
		214	•**Figure 15.2 CRCSAR Register** note 1 added
			Programable I/O Ports
		216	•**16.3 Pull-up Control Register 0 to Pull-up Control Register 2** description modified
		217	•**16.6 Digital Debounce function** equation modified
		218 - 221	•**Figure 16.1 I/O Ports (1) to 16.4 I/O Ports (4)** modified
		227	•**Figure 16.6.1 NDDR and P17DDR Registers** equation modified, note 2 modified
			Flash Memory Version
		231	•**17.1.1 Boot Mode** newly added
		232	•**17.2 Memory Map** partially deleted
		235	•**17.3.1 ROM Code Protect Function** description modified
		236	•**Figure 17.3.1.1 ROMCP Address** modified
		237	•**Table 17.4.1 EW0 Mode and EW1 Mode** note 2 mark deleted
		239	•**17.5.1 Flash Memory Control Register 0** Description about low power consumption mode or on-chip oscillator low-power consumption mode is entered partially modified
		240	•**17.5.2 Flash Memory Control Register 1** Description about FMR17 bit modified
		241	•**Figure 17.5.1 FMR0 Register** note 3 modified, **FMR1 Register:** bit map modified
		242	•**Figure 17.5.2 FMR4 Register** note 2 modified
		243	•**Figure 17.5.1.2 Setting and Resetting of EW1 Mode** note for single-chip mode deleted, note 3 for FMR11 bit added
			Electrical Characteristics
		261	•**Table 18.1 Absolute Maximum Ratings** Rated value modifed, note 1 added
		262	•**Table 18.2 Recommended Operating Conditions** value partially added, figures in note 4 partially added
		263	•**Table 18.3 A/D Conversion Characteristics** note 4 modified
		264	•**Table 18.4 and Table 18.5 Flash Memory Version Electrical Characteristic** note 4 partially added, note 6 and note 7 modified
		265	•Table 18.6 Voltage Detection Circuit Electrical Characteristics conditions modified •Figure for td(P-R) and td(ROC) modified
		267	•**Table 18.9 eElectrical Characteristics (2)** flash memory's value for M16C/26B added, note 5 deleted
		274	•**Table 18.24 Electrical Characteristics (2)** note 5 deleted

Downloaded from Arrow.com.

REVISION HISTORY	M16C/26A Group (M16C/26A, M16C/26B, M16C/26T) Hardware Manual

Rev.	Date	Description	
		Page	Summary
		283	•**Tables 18.41 and 18.42 Flash Memory Version Electrical Characteristics** note 4 , note 10, note 11 modified
		284	•Figure for td(P-R) and td(ROC) modified
		286	•**Table 18.45 Electrical Characteristics** note 4 deleted
		293	•**Table 18.60 Electrical Characteristics (2)** note 4 deleted
			Usage Notes
		299	•**19.1.3 Register Setting** newly added
		306	•**19.5.6 Rewrite the Interrupt Control Register** Example 1 modified
		312	•**19.7.3 Three-phase Motor Control Timer Function** newly added
		315	•**19.9 A/D Converter** Description of section 6 modified
		319	•**19.12.1 Internal ROM Area** description partially added
		321	•**19.13.9 Interrupts** description of EW1 Mode modified, note on watchdog timer interrupts deleted
			•**19.13.10 How to Access** description modified
			Appendix 2. functional Difference
		326	•**Appendix 2.1 Differences between M16C/26A and M16C/26T** description on cold start/warm start detection function deleted

Downloaded from Arrow.com.

RENESAS 16-BIT CMOS SINGLE-CHIP MICROCOMPUTER
HARDWARE MANUAL
M16C/26A Group (M16C/26A, M16C/26B, M16C/26T)

Publication Date: Rev.1.00 Mar.15, 2005
Rev.2.00 Feb.15, 2007

Published by: Sales Strategic Planning Div.
Renesas Technology Corp.

© 2007. Renesas Technology Corp., All rights reserved. Printed in Japan.

Downloaded from Arrow.com.

M16C/26A Group (M16C/26A, M16C/26B, M16C/26T)
Hardware Manual

Renesas Electronics Corporation
1753, Shimonumabe, Nakahara-ku, Kawasaki-shi, Kanagawa 211-8668 Japan

REJ09B0202-0200

Downloaded from Arrow.com.