

To our customers,

---

## Old Company Name in Catalogs and Other Documents

---

On April 1<sup>st</sup>, 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 1<sup>st</sup>, 2010  
Renesas Electronics Corporation

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

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

## 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.

# M16C/80 Group

Hardware Manual

RENESAS 16-BIT SINGLE-CHIP  
MICROCOMPUTER  
M16C FAMILY / M16C/80 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>).

## Keep safety first in your circuit designs!

1. Renesas Technology Corp. puts the maximum effort into making semiconductor products better and more reliable, but there is always the possibility that trouble may occur with them. Trouble with semiconductors may lead to personal injury, fire or property damage. Remember to give due consideration to safety when making your circuit designs, with appropriate measures such as (i) placement of substitutive, auxiliary circuits, (ii) use of non-flammable material or (iii) prevention against any malfunction or mishap.

## Notes regarding these materials

1. These materials are intended as a reference to assist our customers in the selection of the Renesas Technology Corp. product best suited to the customer's application; they do not convey any license under any intellectual property rights, or any other rights, belonging to Renesas Technology Corp. or a third party.
2. Renesas Technology Corp. assumes no responsibility for any damage, or infringement of any third-party's rights, originating in the use of any product data, diagrams, charts, programs, algorithms, or circuit application examples contained in these materials.
3. All information contained in these materials, including product data, diagrams, charts, programs and algorithms represents information on products at the time of publication of these materials, and are subject to change by Renesas Technology Corp. without notice due to product improvements or other reasons. It is therefore recommended that customers contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor for the latest product information before purchasing a product listed herein. The information described here may contain technical inaccuracies or typographical errors. Renesas Technology Corp. assumes no responsibility for any damage, liability, or other loss rising from these inaccuracies or errors. Please also pay attention to information published by Renesas Technology Corp. by various means, including the Renesas Technology Corp. Semiconductor home page (<http://www.renesas.com>).
4. When using any or all of the information contained in these materials, including product data, diagrams, charts, programs, and algorithms, please be sure to evaluate all information as a total system before making a final decision on the applicability of the information and products. Renesas Technology Corp. assumes no responsibility for any damage, liability or other loss resulting from the information contained herein.
5. Renesas Technology Corp. semiconductors are not designed or manufactured for use in a device or system that is used under circumstances in which human life is potentially at stake. Please contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor when considering the use of a product contained herein for any specific purposes, such as apparatus or systems for transportation, vehicular, medical, aerospace, nuclear, or undersea repeater use.
6. The prior written approval of Renesas Technology Corp. is necessary to reprint or reproduce in whole or in part these materials.
7. If these products or technologies are subject to the Japanese export control restrictions, they must be exported under a license from the Japanese government and cannot be imported into a country other than the approved destination. Any diversion or reexport contrary to the export control laws and regulations of Japan and/or the country of destination is prohibited.
8. Please contact Renesas Technology Corp. for further details on these materials or the products contained therein.

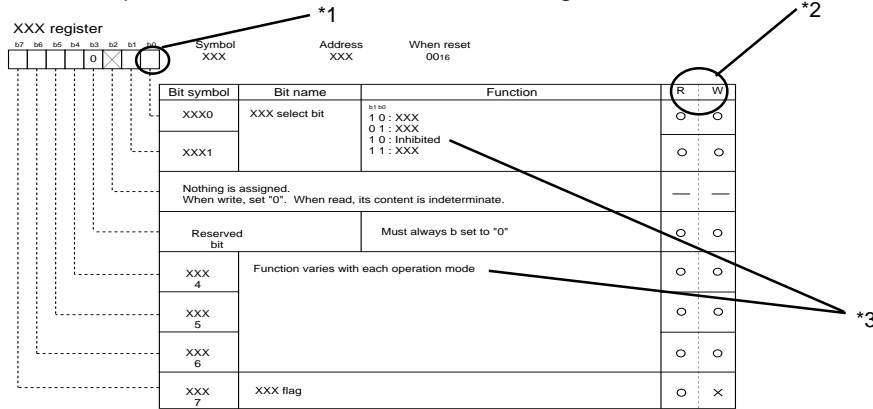
# How to Use This Manual

## 1.Introduction

This hardware manual provides detailed information on the M16C/80 group microcomputers. Users are expected to have basic knowledge of electric circuits, logical circuits and microcomputers.

## 2.Register Diagram

The symbols, and descriptions, used for bit function in each register are shown below.



\*1

Blank: Set to "0" or "1" according to intended use

0: Set to "0"

1: Set to "1"

X: Nothing is assigned

\*2

R: Read

○.....Possible to read

X.....Impossible to read

—.....Nothing is assigned

W: Write

○.....Possible to write

X.....Written value is invalid

When write, value can be "0" or "1"

—.....Nothing is assigned

\*3

Terms to use here are explained as follows.

- Nothing is assigned

Nothing is assigned to the bit concerned. When write, set "0" for new function in future plan.

- Inhibited

Not select. The operation at having selected is not guaranteed.

- Reserved bit

Reserved bit. Set the specified value.

- Function varies with each operation mode

Bit function changes according to the mode of peripheral functions.

- Must be fixed to "0" in A mode

Set the bit concerned to "0" in A mode.

- Invalid in A mode

The bit concerned has no function in A mode. Set the specified value.

- Valid when bit A="0"

When bit A is "1", the bit concerned has no function. When bit A is "0", the bit concerned has function.

### 3. M16C Family Documents

The following documents were prepared for the M16C family. <sup>(1)</sup>

Document	Contents
Short Sheet	Hardware overview
Data Sheet	Hardware overview and electrical characteristics
Hardware Manual	Hardware specifications (pin assignments, memory maps, peripheral specifications, electrical characteristics, timing charts)
Software Manual	Detailed description of assembly instructions and microcomputer performance of each instruction
Application Note	<ul style="list-style-type: none"><li>• Application examples of peripheral functions</li><li>• Sample programs</li><li>• Introduction to the basic functions in the M16C family</li><li>• Programming method with Assembly and C languages</li></ul>
RENESAS TECHNICAL UPDATE	Preliminary report about the specification of a product, a document, etc.

#### NOTES :

1. Before using this material, please visit the our website to verify that this is the most updated document available.

# Table of Contents

<b>Quick Reference by Address</b> .....	<b>B-1</b>
<b>1. Overview</b> .....	<b>1</b>
1.1 Features .....	1
1.2 Applications .....	1
1.3 Pin Configuration .....	2
1.4 Block Diagram .....	5
1.5 Performance Outline .....	6
1.6 Pin Description (1) .....	9
<b>2. Memory</b> .....	<b>12</b>
<b>3. Central Processing Unit (CPU)</b> .....	<b>13</b>
<b>4. Reset</b> .....	<b>18</b>
<b>5. SFR</b> .....	<b>22</b>
<b>6. Processor Mode</b> .....	<b>26</b>
<b>7. Bus</b> .....	<b>30</b>
7.1 Bus Settings .....	30
7.2 Bus Control .....	33
<b>8. Clock Generating Circuit</b> .....	<b>43</b>
8.1 Example of oscillator circuit .....	43
8.2 Clock Control .....	44
8.3 Clock Output .....	47
8.4 Stop Mode .....	48
8.5 Wait Mode .....	50
8.6 Status Transition of BCLK .....	51
8.7 Power Saving .....	53
8.8 Protection .....	55
<b>9. Interrupt Outline</b> .....	<b>56</b>
9.1 Types of Interrupts .....	56
9.2 Software Interrupts .....	57
9.3 Hardware Interrupts .....	58
9.4 High-speed interrupts .....	59
9.5 Interrupts and Interrupt Vector Tables .....	59

9.6 Interrupt control registers .....	62
9.7 Interrupt Enable Flag (I Flag) .....	64
9.8 Interrupt Request Bit .....	64
9.9 Interrupt Priority Level Select Bit and Processor Interrupt Priority Level (IPL) ...	64
9.10 Rewrite the interrupt control register .....	65
9.11 Interrupt Sequence .....	66
9.12 Interrupt Response Time .....	66
9.13 Changes of IPL When Interrupt Request Acknowledged .....	68
9.14 Saving Registers .....	68
9.15 Return from Interrupt Routine .....	69
9.16 Interrupt Priority .....	69
9.17 Interrupt Resolution Circuit .....	69
9.18 $\overline{\text{INT}}$ Interrupts .....	71
9.19 $\overline{\text{NMI}}$ Interrupt .....	72
9.20 Key Input Interrupt .....	72
9.21 Address Match Interrupt .....	73
9.22 Precautions for Interrupts .....	74
<b>10. Watchdog Timer</b> _____	<b>78</b>
<b>11. DMAC</b> _____	<b>80</b>
<b>12. Timer</b> _____	<b>92</b>
<b>13. Timer A</b> _____	<b>94</b>
<b>14. Timer B</b> _____	<b>106</b>
<b>15. Three-phase motor control timers' functions</b> _____	<b>112</b>
<b>16. Serial I/O</b> _____	<b>124</b>
<b>17. Clock synchronous serial I/O mode</b> _____	<b>136</b>
<b>18. Clock asynchronous serial I/O (UART) mode</b> _____	<b>144</b>
<b>19. Clock-asynchronous serial I/O mode (compliant with the SIM interface)</b> ____	<b>151</b>
<b>20. UART<sub>i</sub> Special Mode Register (i = 2 to 4)</b> _____	<b>155</b>
<b>21. A/D Converter</b> _____	<b>166</b>
<b>22. D/A Converter</b> _____	<b>176</b>
<b>23. CRC Calculation Circuit</b> _____	<b>178</b>
<b>24. XY Converter</b> _____	<b>180</b>



<b>25. DRAM Controller</b>	<b>183</b>
<b>26. Programmable I/O Ports</b>	<b>190</b>
<b>27. Usage Precaution</b>	<b>208</b>
<b>28. Electrical characteristics</b>	<b>225</b>
<b>29. Flash Memory Version</b>	<b>271</b>
<b>30. CPU Rewrite Mode</b>	<b>274</b>
<b>31. Parallel I/O Mode</b>	<b>290</b>
<b>32. Standard serial I/O mode</b>	<b>295</b>
32.1 Overview of standard serial I/O mode 1 (clock synchronized) .....	296
32.2 Overview of standard serial I/O mode 2 (clock asynchronous) .....	311
 <b>Package Dimension</b>	 <b>326</b>
 <b>Register Index</b>	 <b>328</b>
 <b>Revision History</b>	 <b>C-1</b>

## Quick Reference by Address

Address	Register	Symbol	page	Address	Register	Symbol	page
0000 <sub>16</sub>				0060 <sub>16</sub>			
0001 <sub>16</sub>				0061 <sub>16</sub>			
0002 <sub>16</sub>				0062 <sub>16</sub>			
0003 <sub>16</sub>				0063 <sub>16</sub>			
0004 <sub>16</sub>	Processor mode register 0	PM0	27	0064 <sub>16</sub>			
0005 <sub>16</sub>	Processor mode register 1	PM1	28	0065 <sub>16</sub>			
0006 <sub>16</sub>	System clock control register 0	CM0	46	0066 <sub>16</sub>			
0007 <sub>16</sub>	System clock control register 1	CM1		0067 <sub>16</sub>			
0008 <sub>16</sub>	Wait control register	WCR	40	0068 <sub>16</sub>	DMA0 interrupt control register	DM0IC	63
0009 <sub>16</sub>	Address match interrupt enable register	AIER	73	0069 <sub>16</sub>	Timer B5 interrupt control register	TB5IC	
000A <sub>16</sub>	Protect register	PRCR	55	006A <sub>16</sub>	DMA2 interrupt control register	DM2IC	
000B <sub>16</sub>	External data bus width control register	DS	31	006B <sub>16</sub>	UART2 receive/ACK interrupt control register	S2RIC	
000C <sub>16</sub>	Main clock division register	MCD	47	006C <sub>16</sub>	Timer A0 interrupt control register	TA0IC	
000D <sub>16</sub>				006D <sub>16</sub>	UART3 receive/ACK interrupt control register	S3RIC	
000E <sub>16</sub>	Watchdog timer start register	WDTS	79	006E <sub>16</sub>	Timer A2 interrupt control register	TA2IC	
000F <sub>16</sub>	Watchdog timer control register	WDC		006F <sub>16</sub>	UART4 receive/ACK interrupt control register	S4RIC	
0010 <sub>16</sub>				0070 <sub>16</sub>	Timer A4 interrupt control register	TA4IC	
0011 <sub>16</sub>	Address match interrupt register 0	RMAD0	73	0071 <sub>16</sub>	Bus collision detection(UART3) interrupt control register	BCN3IC	
0012 <sub>16</sub>							
0013 <sub>16</sub>				0072 <sub>16</sub>	UART0 receive interrupt control register	S0RIC	
0014 <sub>16</sub>				0073 <sub>16</sub>	A/D conversion interrupt control register	ADIC	
0015 <sub>16</sub>	Address match interrupt register 1	RMAD1	73	0074 <sub>16</sub>	UART1 receive interrupt control register	S1RIC	
0016 <sub>16</sub>							
0017 <sub>16</sub>				0075 <sub>16</sub>			
0018 <sub>16</sub>				0076 <sub>16</sub>	Timer B1 interrupt control register	TB1IC	63
0019 <sub>16</sub>	Address match interrupt register 2	RMAD2	73	0077 <sub>16</sub>			
001A <sub>16</sub>							
001B <sub>16</sub>				0078 <sub>16</sub>	Timer B3 interrupt control register	TB3IC	63
001C <sub>16</sub>				0079 <sub>16</sub>			
001D <sub>16</sub>	Address match interrupt register 3	RMAD3	73	007A <sub>16</sub>	INT5 interrupt control register	INT5IC	63
001E <sub>16</sub>							
001F <sub>16</sub>				007B <sub>16</sub>			
0020 <sub>16</sub>		*		007C <sub>16</sub>	INT3 interrupt control register	INT3IC	63
0021 <sub>16</sub>	Emulator interrupt vector table register	EIAD		007D <sub>16</sub>			
0022 <sub>16</sub>				007E <sub>16</sub>	INT1 interrupt control register	INT1IC	63
0023 <sub>16</sub>	Emulator interrupt detect register	EITD		007F <sub>16</sub>			
0024 <sub>16</sub>	Emulator protect register	EPRR		0080 <sub>16</sub>			
0025 <sub>16</sub>				0081 <sub>16</sub>			
0026 <sub>16</sub>				0082 <sub>16</sub>			
0027 <sub>16</sub>				0083 <sub>16</sub>			
0028 <sub>16</sub>				0084 <sub>16</sub>			
0029 <sub>16</sub>				0085 <sub>16</sub>			
002A <sub>16</sub>				0086 <sub>16</sub>			
002B <sub>16</sub>				0087 <sub>16</sub>			
002C <sub>16</sub>				0088 <sub>16</sub>	DMA1 interrupt control register	DM1IC	63
002D <sub>16</sub>				0089 <sub>16</sub>	UART2 transmit/NACK interrupt control register	S2TIC	
002E <sub>16</sub>				008A <sub>16</sub>	DMA3 interrupt control register	DM3IC	
002F <sub>16</sub>				008B <sub>16</sub>	UART3 transmit/NACK interrupt control register	S3TIC	
0030 <sub>16</sub>	ROM areaset register	ROA		008C <sub>16</sub>	Timer A1 interrupt control register	TA1IC	
0031 <sub>16</sub>	Debug monitor area set register	DBA		008D <sub>16</sub>	UART4 transmit/NACK interrupt control register	S4TIC	
0032 <sub>16</sub>	Expansion area set register 0	EXA0		008E <sub>16</sub>	Timer A3 interrupt control register	TA3IC	
0033 <sub>16</sub>	Expansion area set register 1	EXA1		008F <sub>16</sub>	Bus collision detection(UART2) interrupt control register	BCN2IC	
0034 <sub>16</sub>	Expansion area set register 2	EXA2		0090 <sub>16</sub>	UART0 transmit interrupt control register	S0TIC	
0035 <sub>16</sub>	Expansion area set register 3	EXA3		0091 <sub>16</sub>	Bus collision detection(UART4) interrupt control register	BCN4IC	
0036 <sub>16</sub>				0092 <sub>16</sub>	UART1 transmit interrupt control register	S1TIC	
0037 <sub>16</sub>				0093 <sub>16</sub>	Key input interrupt control register	KUPIIC	
0038 <sub>16</sub>				0094 <sub>16</sub>	Timer B0 interrupt control register	TB0IC	
0039 <sub>16</sub>				0095 <sub>16</sub>			
003A <sub>16</sub>				0096 <sub>16</sub>	Timer B2 interrupt control register	TB2IC	63
003B <sub>16</sub>				0097 <sub>16</sub>			
003C <sub>16</sub>				0098 <sub>16</sub>	Timer B4 interrupt control register	TB4IC	63
003D <sub>16</sub>				0099 <sub>16</sub>			
003E <sub>16</sub>				009A <sub>16</sub>	INT4 interrupt control register	INT4IC	63
003F <sub>16</sub>				009B <sub>16</sub>			
0040 <sub>16</sub>	DRAM control register	DRAMCONT	183	009C <sub>16</sub>	INT2 interrupt control register	INT2IC	63
0041 <sub>16</sub>	DRAM refresh interval set register	REFCNT	185	009D <sub>16</sub>			
0042 <sub>16</sub>				009E <sub>16</sub>	INT0 interrupt control register	INT0IC	63
0043 <sub>16</sub>				009F <sub>16</sub>	Exit priority register	RLVL	49
0044 <sub>16</sub>				00A0 <sub>16</sub>			
				00A1 <sub>16</sub>			
				00A2 <sub>16</sub>			
				00A3 <sub>16</sub>			
				00A4 <sub>16</sub>			

## Quick Reference by Address

Address	Register	Symbol	page
02C0 <sub>16</sub> 02C1 <sub>16</sub>	X0 register ,Y0 register	X0R,Y0R	181
02C2 <sub>16</sub> 02C3 <sub>16</sub>	X1 register, Y1 register	X1R,Y1R	
02C4 <sub>16</sub> 02C5 <sub>16</sub>	X2 register ,Y2 register	X2R,Y2R	
02C6 <sub>16</sub> 02C7 <sub>16</sub>	X3 register,Y3 register	X3R,Y3R	
02C8 <sub>16</sub> 02C9 <sub>16</sub>	X4 register , Y4 register	X4R,Y4R	
02CA <sub>16</sub> 02CB <sub>16</sub>	X5 register , Y5 register	X5R,Y5R	
02CC <sub>16</sub> 02CD <sub>16</sub>	X6 register ,Y6 register	X6R,Y6R	
02CE <sub>16</sub> 02CF <sub>16</sub>	X7 register ,Y7 register	X7R,Y7R	
02D0 <sub>16</sub> 02D1 <sub>16</sub>	X8 register , Y8 register	X8R,Y8R	
02D2 <sub>16</sub> 02D3 <sub>16</sub>	X9 register,Y9 register	X9R,Y9R	
02D4 <sub>16</sub> 02D5 <sub>16</sub>	X10 register,Y10 register	X10R,Y10R	
02D6 <sub>16</sub> 02D7 <sub>16</sub>	X11 register, Y11 register	X11R,Y11R	
02D8 <sub>16</sub> 02D9 <sub>16</sub>	X12 register, Y12 register	X12R,Y12R	
02DA <sub>16</sub> 02DB <sub>16</sub>	X13 register,Y13 register	X13R,Y13R	
02DC <sub>16</sub> 02DD <sub>16</sub>	X14 register,Y14 register	X14R,Y14R	
02DE <sub>16</sub> 02DF <sub>16</sub>	X15 register, Y15 register	X15R,Y15R	
02E0 <sub>16</sub>	XY control register	XYC	180
02E1 <sub>16</sub>			
02E2 <sub>16</sub>			
02E3 <sub>16</sub>			
02E4 <sub>16</sub>			
02E5 <sub>16</sub>			
02E6 <sub>16</sub>			
02E7 <sub>16</sub>			
02E8 <sub>16</sub>			
02E9 <sub>16</sub>			
02EA <sub>16</sub>			
02EB <sub>16</sub>			
02EC <sub>16</sub>			
02ED <sub>16</sub>			
02EE <sub>16</sub>			
02EF <sub>16</sub>			
02F0 <sub>16</sub>			
02F1 <sub>16</sub>			
02F2 <sub>16</sub>			
02F3 <sub>16</sub>			
02F4 <sub>16</sub>			
02F5 <sub>16</sub>	UART4 special mode register 3	U4SMR3	136
02F6 <sub>16</sub>	UART4 special mode register 2	U4SMR2	135
02F7 <sub>16</sub>	UART4 special mode register	U4SMR	134
02F8 <sub>16</sub>	UART4 transmit/receive mode register	U4MR	130
02F9 <sub>16</sub>	UART4 bit rate generator	U4BRG	129
02FA <sub>16</sub> 02FB <sub>16</sub>	UART4 transmit buffer register	U4TB	
02FC <sub>16</sub>	UART4 transmit/receive control register 0	U4C0	132
02FD <sub>16</sub>	UART4 transmit/receive control register 1	U4C1	133
02FE <sub>16</sub> 02FF <sub>16</sub>	UART4 receive buffer register	U4RB	129

Address	Register	Symbol	page	
0300 <sub>16</sub>	Timer B3, 4, 5 count start flag	TBSR	107	
0301 <sub>16</sub>				
0302 <sub>16</sub> 0303 <sub>16</sub>	Timer A1-1 register	TA11	114	
0304 <sub>16</sub> 0305 <sub>16</sub>	Timer A2-1 register	TA21		
0306 <sub>16</sub> 0307 <sub>16</sub>	Timer A4-1 register	TA41		
0308 <sub>16</sub>	Three-phase PWM control register 0	INVC0		
0309 <sub>16</sub>	Three-phase PWM control register 1	INVC1		
030A <sub>16</sub> 030B <sub>16</sub>	Three-phase output buffer register 0	IDB0	113	
030C <sub>16</sub>	Three-phase output buffer register 1	IDB1		
030D <sub>16</sub>	Dead time timer	DTT		
030E <sub>16</sub>	Timer B2 interrupt occurrence frequency set counter	ICTB2		
030F <sub>16</sub>				
0310 <sub>16</sub>				
0311 <sub>16</sub>	Timer B3 register	TB3	107	
0312 <sub>16</sub> 0313 <sub>16</sub>	Timer B4 register	TB4		
0314 <sub>16</sub> 0315 <sub>16</sub>	Timer B5 register	TB5		
0316 <sub>16</sub>				
0317 <sub>16</sub>				
0318 <sub>16</sub>				
0319 <sub>16</sub>				
031A <sub>16</sub>				
031B <sub>16</sub>	Timer B3 mode register	TB3MR	106	
031C <sub>16</sub>	Timer B4 mode register	TB4MR		
031D <sub>16</sub>	Timer B5 mode register	TB5MR		
031E <sub>16</sub>				
031F <sub>16</sub>	Interrupt cause select register	IFSR	71	
0320 <sub>16</sub>				
0321 <sub>16</sub>				
0322 <sub>16</sub>				
0323 <sub>16</sub>				
0324 <sub>16</sub>				
0325 <sub>16</sub>	UART3 special mode register 3	U3SMR3	136	
0326 <sub>16</sub>	UART3 special mode register 2	U3SMR2	135	
0327 <sub>16</sub>	UART3 special mode register	U3SMR	134	
0328 <sub>16</sub>	UART3 transmit/receive mode register	U3MR	130	
0329 <sub>16</sub>	UART3 bit rate generator	U3BRG	129	
032A <sub>16</sub> 032B <sub>16</sub>	UART3 transmit buffer register	U3TB	129	
032C <sub>16</sub>	UART3 transmit/receive control register 0	U3C0	132	
032D <sub>16</sub>	UART3 transmit/receive control register 1	U3C1	133	
032E <sub>16</sub> 032F <sub>16</sub>	UART3 receive buffer register	U3RB	129	
0330 <sub>16</sub>				
0331 <sub>16</sub>				
0332 <sub>16</sub>				
0333 <sub>16</sub>				
0334 <sub>16</sub>				
0335 <sub>16</sub>	UART2 special mode register 3	U2SMR3	136	
0336 <sub>16</sub>	UART2 special mode register 2	U2SMR2	135	
0337 <sub>16</sub>	UART2 special mode register	U2SMR	134	
0338 <sub>16</sub>	UART2 transmit/receive mode register	U2MR	130	
0339 <sub>16</sub>	UART2 bit rate generator	U2BRG	129	
033A <sub>16</sub> 033B <sub>16</sub>	UART2 transmit buffer register	U2TB		
033C <sub>16</sub>	UART2 transmit/receive control register 0	U2C0	131	
033D <sub>16</sub>	UART2 transmit/receive control register 1	U2C1	133	
033E <sub>16</sub> 033F <sub>16</sub>	UART2 receive buffer register	U2RB	129	

Blank spaces are reserved. No access is allowed.

## Quick Reference by Address

Address	Register	Symbol	page		
0340 <sub>16</sub>	Count start flag	TABSR	95		
0341 <sub>16</sub>	Clock prescaler reset flag	CPSRF	96		
0342 <sub>16</sub>	One-shot start flag	ONSF			
0343 <sub>16</sub>	Trigger select register	TRGSR			
0344 <sub>16</sub>	Up-down flag	UDF	95		
0345 <sub>16</sub>					
0346 <sub>16</sub>	Timer A0 register	TA0	95		
0347 <sub>16</sub>					
0348 <sub>16</sub>	Timer A1 register	TA1			
0349 <sub>16</sub>					
034A <sub>16</sub>	Timer A2 register	TA2			
034B <sub>16</sub>					
034C <sub>16</sub>	Timer A3 register	TA3			
034D <sub>16</sub>					
034E <sub>16</sub>	Timer A4 register	TA4			
034F <sub>16</sub>					
0350 <sub>16</sub>	Timer B0 register	TB0	107		
0351 <sub>16</sub>					
0352 <sub>16</sub>					
0353 <sub>16</sub>	Timer B1 register	TB1			
0354 <sub>16</sub>					
0355 <sub>16</sub>	Timer B2 register	TB2			
0356 <sub>16</sub>					
0357 <sub>16</sub>	Timer A0 mode register	TA0MR	94		
0358 <sub>16</sub>	Timer A1 mode register	TA1MR			
0359 <sub>16</sub>	Timer A2 mode register	TA2MR			
035A <sub>16</sub>	Timer A3 mode register	TA3MR			
035B <sub>16</sub>	Timer A4 mode register	TA4MR			
035C <sub>16</sub>	Timer B0 mode register	TB0MR	106		
035D <sub>16</sub>	Timer B1 mode register	TB1MR			
035E <sub>16</sub>	Timer B2 mode register	TB2MR			
035F <sub>16</sub>					
0360 <sub>16</sub>	UART0 transmit/receive mode register	U0MR	130		
0361 <sub>16</sub>	UART0 bit rate generator	U0BRG	129		
0362 <sub>16</sub>	UART0 transmit buffer register	U0TB	129		
0363 <sub>16</sub>					
0364 <sub>16</sub>	UART0 transmit/receive control register 0	U0C0	131		
0365 <sub>16</sub>	UART0 transmit/receive control register 1	U0C1	133		
0366 <sub>16</sub>	UART0 receive buffer register	U0RB	129		
0367 <sub>16</sub>					
0368 <sub>16</sub>	UART1 transmit/receive mode register	U1MR	130		
0369 <sub>16</sub>	UART1 bit rate generator	U1BRG	129		
036A <sub>16</sub>	UART1 transmit buffer register	U1TB			
036B <sub>16</sub>					
036C <sub>16</sub>	UART1 transmit/receive control register 0	U1C0	131		
036D <sub>16</sub>	UART1 transmit/receive control register 1	U1C1	133		
036E <sub>16</sub>	UART1 receive buffer register	U1RB	129		
036F <sub>16</sub>					
0370 <sub>16</sub>	UART transmit/receive control register 2	UCON	134		
0371 <sub>16</sub>					
0372 <sub>16</sub>					
0373 <sub>16</sub>					
0374 <sub>16</sub>					
0375 <sub>16</sub>					
0376 <sub>16</sub>	Flash memory control register 1	FMR1	276		
0377 <sub>16</sub>	Flash memory control register 0	FMR0			
0378 <sub>16</sub>	DMA0 request cause select register	DM0SL	82		
0379 <sub>16</sub>	DMA1 request cause select register	DM1SL			
037A <sub>16</sub>	DMA2 request cause select register	DM2SL			
037B <sub>16</sub>	DMA3 request cause select register	DM3SL			
037C <sub>16</sub>	CRC data register	CRCD	178		
037D <sub>16</sub>					
037E <sub>16</sub>	CRC input register	CRCIN			
037F <sub>16</sub>					

Address	Register	Symbol	page
0380 <sub>16</sub>	A/D register 0	AD0	169
0381 <sub>16</sub>			
0382 <sub>16</sub>	A/D register 1	AD1	
0383 <sub>16</sub>			
0384 <sub>16</sub>	A/D register 2	AD2	
0385 <sub>16</sub>			
0386 <sub>16</sub>	A/D register 3	AD3	
0387 <sub>16</sub>			
0388 <sub>16</sub>	A/D register 4	AD4	
0389 <sub>16</sub>			
038A <sub>16</sub>	A/D register 5	AD5	
038B <sub>16</sub>			
038C <sub>16</sub>	A/D register 6	AD6	
038D <sub>16</sub>			
038E <sub>16</sub>	A/D register 7	AD7	
038F <sub>16</sub>			
0390 <sub>16</sub>			
0391 <sub>16</sub>			
0392 <sub>16</sub>			
0393 <sub>16</sub>			
0394 <sub>16</sub>	A/D control register 2	ADCON2	169
0395 <sub>16</sub>			
0396 <sub>16</sub>	A/D control register 0	ADCON0	168
0397 <sub>16</sub>	A/D control register 1	ADCON1	
0398 <sub>16</sub>	D/A register 0	DA0	177
0399 <sub>16</sub>			
039A <sub>16</sub>	D/A register 1	DA1	177
039B <sub>16</sub>			
039C <sub>16</sub>	D/A control register	DACON	177
039D <sub>16</sub>			
039E <sub>16</sub>			
039F <sub>16</sub>			
03A0 <sub>16</sub>			
03A1 <sub>16</sub>			
03A2 <sub>16</sub>			
03A3 <sub>16</sub>			
03A4 <sub>16</sub>			
03A5 <sub>16</sub>			
03A6 <sub>16</sub>			
03A7 <sub>16</sub>			
03A8 <sub>16</sub>			
03A9 <sub>16</sub>			
03AA <sub>16</sub>			
03AB <sub>16</sub>			
03AC <sub>16</sub>			
03AD <sub>16</sub>			
03AE <sub>16</sub>			
03AF <sub>16</sub>	Function select register C	PSC	203
03B0 <sub>16</sub>	Function select register A0	PS0	200
03B1 <sub>16</sub>	Function select register A1	PS1	
03B2 <sub>16</sub>	Function select register B0	PSL0	202
03B3 <sub>16</sub>	Function select register B1	PSL1	
03B4 <sub>16</sub>	Function select register A2	PS2	201
03B5 <sub>16</sub>	Function select register A3	PS3	
03B6 <sub>16</sub>	Function select register B2	PSL2	202
03B7 <sub>16</sub>	Function select register B3	PSL3	203
03B8 <sub>16</sub>			
03B9 <sub>16</sub>			
03BA <sub>16</sub>			
03BB <sub>16</sub>			
03BC <sub>16</sub>			
03BD <sub>16</sub>			
03BE <sub>16</sub>			
03BF <sub>16</sub>			

Blank spaces are reserved. No access is allowed.

## Quick Reference by Address

### <100-pin version>

Address	Register	Symbol	page
03C0 <sub>16</sub>	Port P6	P6	197
03C1 <sub>16</sub>	Port P7	P7	
03C2 <sub>16</sub>	Port P6 direction register	PD6	195
03C3 <sub>16</sub>	Port P7 direction register	PD7	
03C4 <sub>16</sub>	Port P8	P8	197
03C5 <sub>16</sub>	Port P9	P9	
03C6 <sub>16</sub>	Port P8 direction register	PD8	195
03C7 <sub>16</sub>	Port P9 direction register	PD9	
03C8 <sub>16</sub>	Port P10	P10	197
03C9 <sub>16</sub>			
03CA <sub>16</sub>	Port P10 direction register	PD10	195
03CB <sub>16</sub>			
03CC <sub>16</sub>			
03CD <sub>16</sub>			
03CE <sub>16</sub>			
03CF <sub>16</sub>			
03D0 <sub>16</sub>			
03D1 <sub>16</sub>			
03D2 <sub>16</sub>			
03D3 <sub>16</sub>			
03D4 <sub>16</sub>			
03D5 <sub>16</sub>			
03D6 <sub>16</sub>			
03D7 <sub>16</sub>			
03D8 <sub>16</sub>			
03D9 <sub>16</sub>			
03DA <sub>16</sub>	Pull-up control register 2	PUR2	204
03DB <sub>16</sub>	Pull-up control register 3	PUR3	205
03DC <sub>16</sub>			
03DD <sub>16</sub>			
03DE <sub>16</sub>			
03DF <sub>16</sub>			
03E0 <sub>16</sub>	Port P0	P0	197
03E1 <sub>16</sub>	Port P1	P1	
03E2 <sub>16</sub>	Port P0 direction register	PD0	195
03E3 <sub>16</sub>	Port P1 direction register	PD1	
03E4 <sub>16</sub>	Port P2	P2	197
03E5 <sub>16</sub>	Port P3	P3	
03E6 <sub>16</sub>	Port P2 direction register	PD2	195
03E7 <sub>16</sub>	Port P3 direction register	PD3	
03E8 <sub>16</sub>	Port P4	P4	197
03E9 <sub>16</sub>	Port P5	P5	
03EA <sub>16</sub>	Port P4 direction register	PD4	195
03EB <sub>16</sub>	Port P5 direction register	PD5	
03EC <sub>16</sub>			
03ED <sub>16</sub>			
03EE <sub>16</sub>			
03EF <sub>16</sub>			
03F0 <sub>16</sub>	Pull-up control register 0	PUR0	204
03F1 <sub>16</sub>	Pull-up control register 1	PUR1	
03F2 <sub>16</sub>			
03F3 <sub>16</sub>			
03FC <sub>16</sub>			
03FD <sub>16</sub>			
03FE <sub>16</sub>			
03FF <sub>16</sub>	Port control register	PCR	206

### <144-pin version>

Address	Register	Symbol	page
03C0 <sub>16</sub>	Port P6	P6	197
03C1 <sub>16</sub>	Port P7	P7	
03C2 <sub>16</sub>	Port P6 direction register	PD6	195
03C3 <sub>16</sub>	Port P7 direction register	PD7	
03C4 <sub>16</sub>	Port P8	P8	197
03C5 <sub>16</sub>	Port P9	P9	
03C6 <sub>16</sub>	Port P8 direction register	PD8	195
03C7 <sub>16</sub>	Port P9 direction register	PD9	
03C8 <sub>16</sub>	Port P10	P10	197
03C9 <sub>16</sub>	Port P11	P11	198
03CA <sub>16</sub>	Port P10 direction register	PD10	195
03CB <sub>16</sub>	Port P11 direction register	PD11	196
03CC <sub>16</sub>	Port P12	P12	197
03CD <sub>16</sub>	Port P13	P13	
03CE <sub>16</sub>	Port P12 direction register	PD12	195
03CF <sub>16</sub>	Port P13 direction register	PD13	
03D0 <sub>16</sub>	Port P14	P14	198
03D1 <sub>16</sub>	Port P15	P15	197
03D2 <sub>16</sub>	Port P14 direction register	PD14	196
03D3 <sub>16</sub>	Port P15 direction register	PD15	195
03D4 <sub>16</sub>			
03D5 <sub>16</sub>			
03D6 <sub>16</sub>			
03D7 <sub>16</sub>			
03D8 <sub>16</sub>			
03D9 <sub>16</sub>			
03DA <sub>16</sub>	Pull-up control register 2	PUR2	204
03DB <sub>16</sub>	Pull-up control register 3	PUR3	205
03DC <sub>16</sub>	Pull-up control register 4	PUR4	
03DD <sub>16</sub>			
03DE <sub>16</sub>			
03DF <sub>16</sub>			
03E0 <sub>16</sub>	Port P0	P0	197
03E1 <sub>16</sub>	Port P1	P1	
03E2 <sub>16</sub>	Port P0 direction register	PD0	195
03E3 <sub>16</sub>	Port P1 direction register	PD1	
03E4 <sub>16</sub>	Port P2 (P2)	P2	197
03E5 <sub>16</sub>	Port P3 (P3)	P3	
03E6 <sub>16</sub>	Port P2 direction register	PD2	195
03E7 <sub>16</sub>	Port P3 direction register	PD3	
03E8 <sub>16</sub>	Port P4	P4	197
03E9 <sub>16</sub>	Port P5	P5	
03EA <sub>16</sub>	Port P4 direction register	PD4	195
03EB <sub>16</sub>	Port P5 direction register	PD5	
03EC <sub>16</sub>			
03ED <sub>16</sub>			
03EE <sub>16</sub>			
03EF <sub>16</sub>			
03F0 <sub>16</sub>	Pull-up control register 0	PUR0	204
03F1 <sub>16</sub>	Pull-up control register 1	PUR1	
03F2 <sub>16</sub>			
03F3 <sub>16</sub>			
03FC <sub>16</sub>			
03FD <sub>16</sub>			
03FE <sub>16</sub>			
03FF <sub>16</sub>	Port control register	PCR	206

Blank spaces are reserved. No access is allowed.

## 1. Overview

The M16C/80 group of single-chip microcomputers are built using the high-performance silicon gate CMOS process using a M16C/80 Series CPU core and are packaged in a 100-pin and 144-pin plastic molded QFP. The peripheral functions of 100-pin and 144-pin are common. These single-chip microcomputers operate using sophisticated instructions featuring a high level of instruction efficiency. With 16M bytes of address space, they are capable of executing instructions at high speed. They also feature a built-in multiplier and DMAC, making them ideal for controlling office, communications, industrial equipment, and other high-speed processing applications.

### 1.1 Features

- Memory capacity ..... ROM (See ROM expansion figure.)  
RAM 10 to 24 Kbytes
- Shortest instruction execution time ..... 50ns ( $f(X_{IN})=20\text{MHz}$ )
- Supply voltage ..... 4.2 to 5.5V ( $f(X_{IN})=20\text{MHz}$ )  
Mask ROM, external ROM and flash memory versions  
2.7 to 5.5V ( $f(X_{IN})=10\text{MHz}$ )  
Mask ROM, external ROM and flash memory versions
- Low power consumption ..... 45mA (M30800MC-XXXFP)  
( $f(X_{IN}) = 20\text{MHz}$  without software wait,  $V_{CC}=5\text{V}$ )
- Interrupts ..... 29 internal and 8 external interrupt sources, 5 software interrupt sources; 7 levels (including key input interrupt)
- Multifunction 16-bit timer ..... 5 output timers + 6 input timers
- Serial I/O ..... 5 channels for UART or clock synchronous
- DMAC ..... 4 channels (trigger: 31 sources)
- DRAMC ..... Used for EDO, FP, CAS before RAS refresh, self-refresh
- A/D converter ..... 10 bits X 8 channels (Expandable up to 10 channels)
- D/A converter ..... 8 bits X 2 channels
- CRC calculation circuit ..... 1 circuit
- XY converter ..... 1 circuit
- Watchdog timer ..... 1 line
- Programmable I/O ..... 87 lines:100-pin version, 123 lines:144-pin version
- Input port ..... 1 line (P85 shared with  $\overline{\text{NMI}}$  pin)
- Memory expansion ..... Available (16M bytes)
- Chip select output ..... 4 lines
- Clock generating circuit ..... 2 built-in clock generation circuits  
(built-in feedback resistance, and external ceramic or quartz oscillator)

### 1.2 Applications

Audio, cameras, office equipment, communications equipment, portable equipment, etc.

### 1.3 Pin Configuration

Figures 1.1 and 1.2 show the pin configuration (top view) for 100-pin and Figure 1.3 shows the pin configuration (top view) for 144-pin.

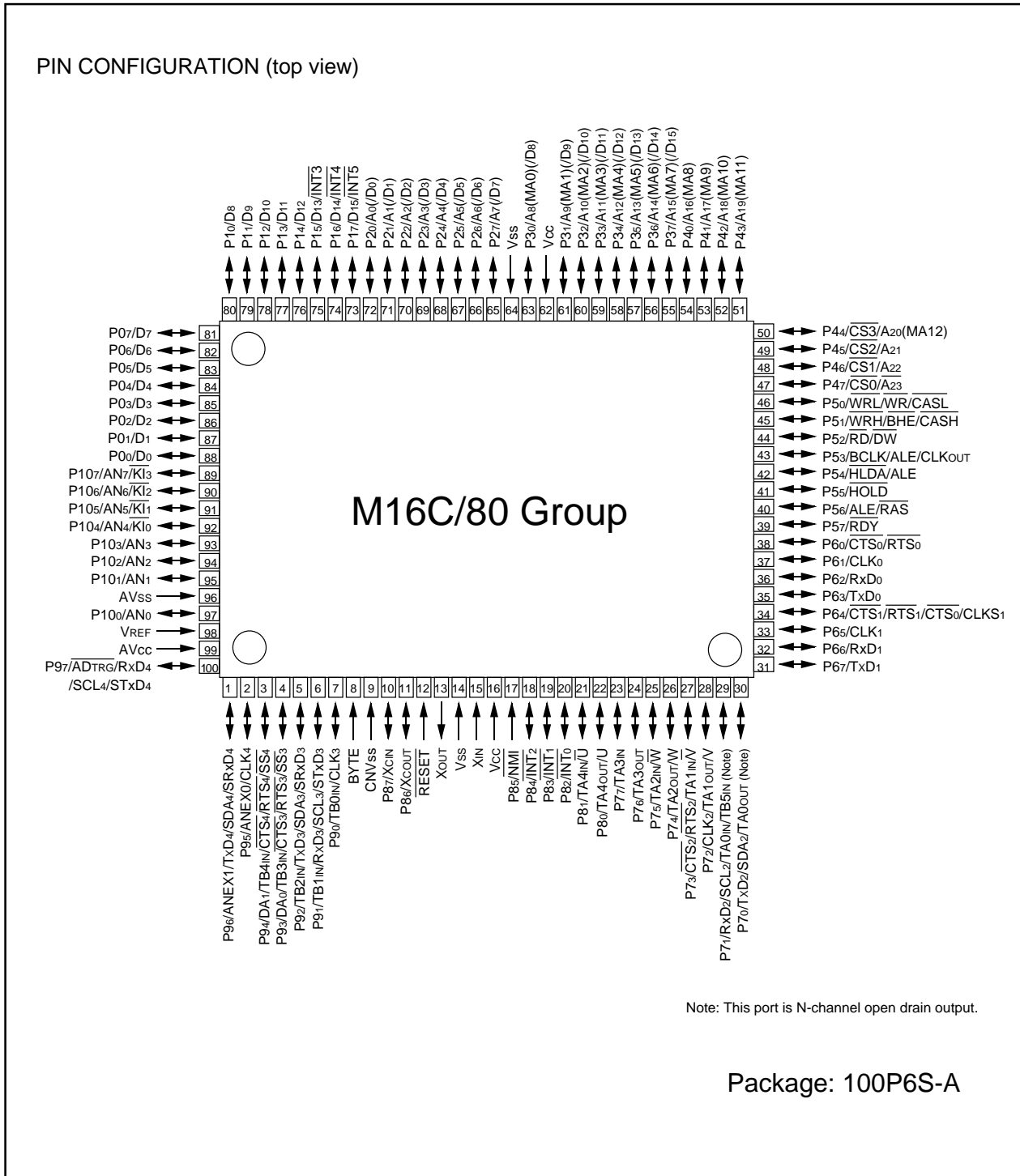


Figure 1.1 Pin configuration for 100-pin version (top view) (1)

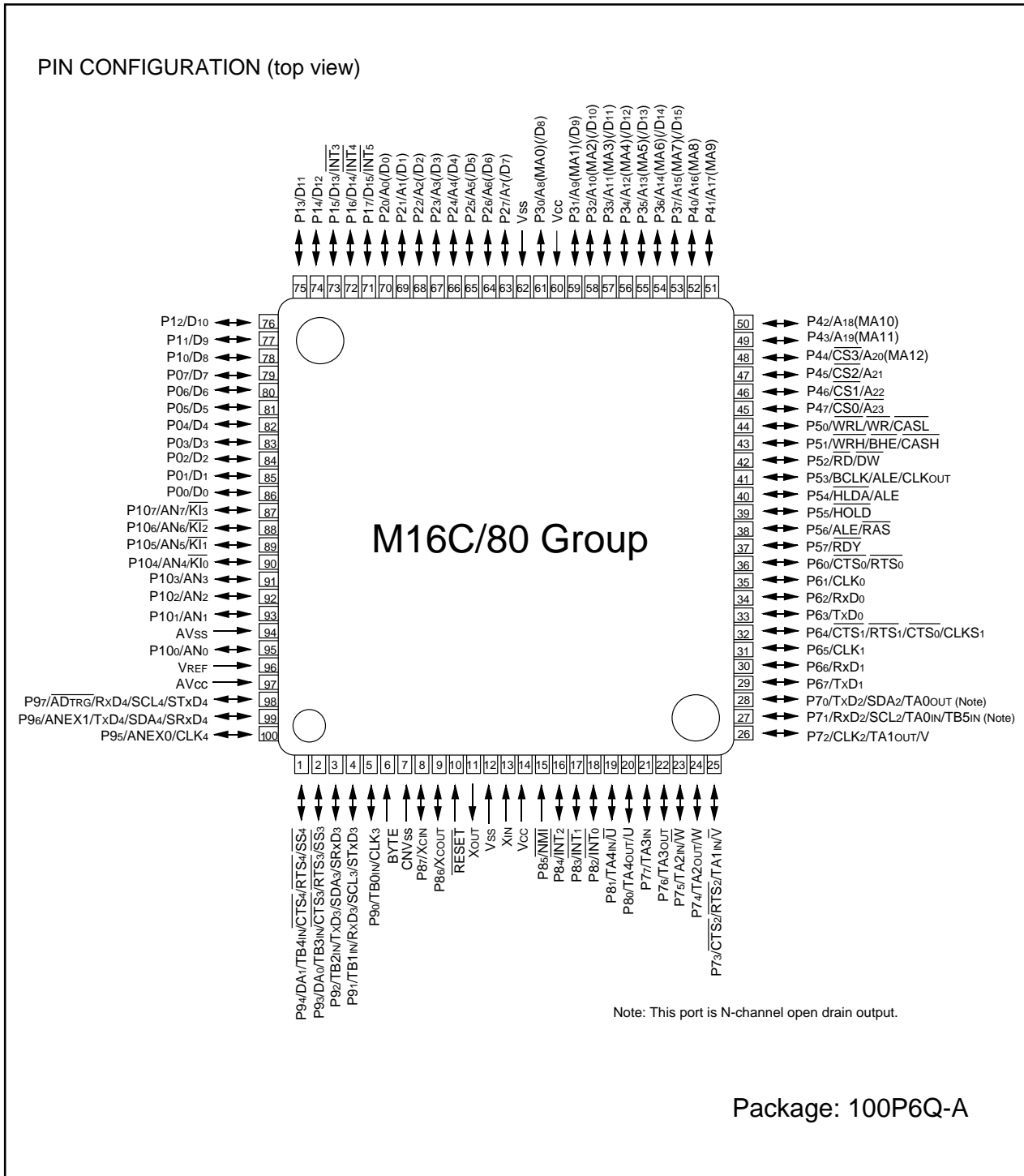


Figure 1.2 Pin configuration for 100-pin version (top view) (2)



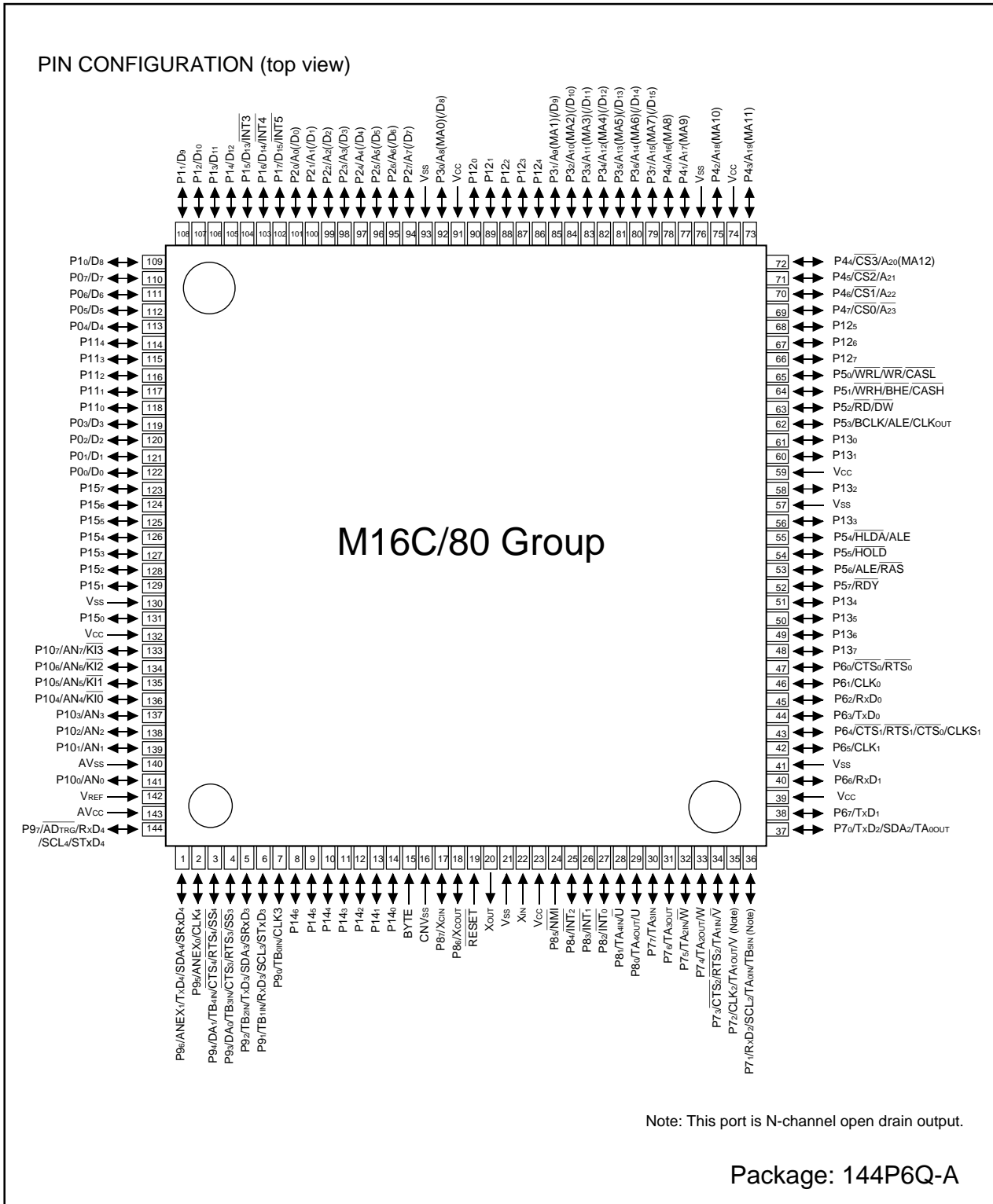


Figure 1.3 Pin configuration for 144-pin version (top view)

### 1.4 Block Diagram

Figure 1.4 is a block diagram of the M16C/80 group.

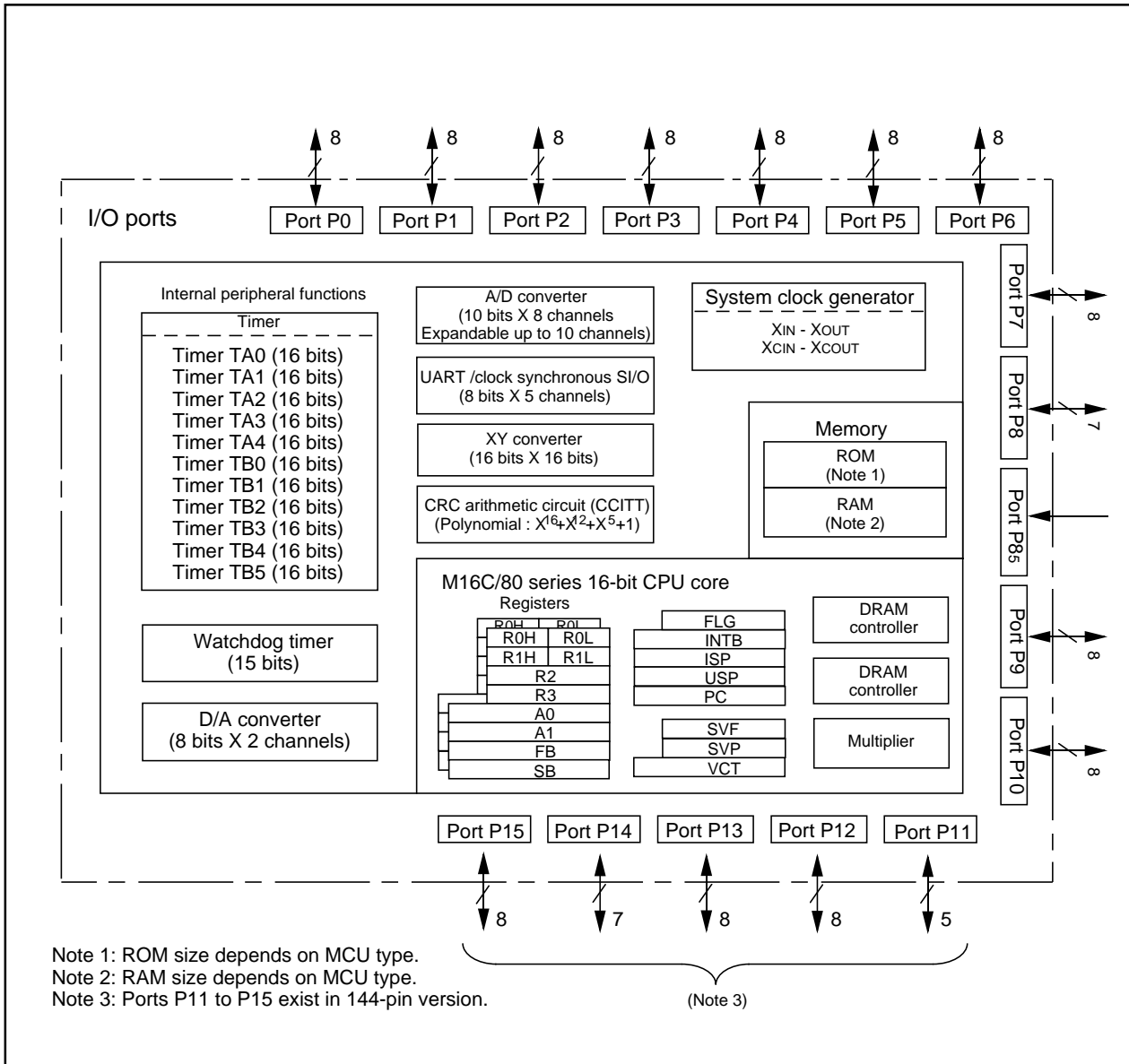


Figure 1.4 Block diagram of the M16C/80 group

## 1.5 Performance Outline

Table 1.1 is a performance outline of M16C/80 group.

**Table 1.1 Performance outline of M16C/80 group**

Item		Performance
Number of basic instructions		106 instructions
Shortest instruction execution time		50ns(f(XIN)=20MHz)
Memory capacity	ROM	See ROM expansion figure.
	RAM	10 to 24 K bytes
I/O port	100-pin	P0 to P10 (except P85) 8-bit x 10, 7-bit x 1
	144-pin	P0 to P15 (except P85) 8-bit x 13, 7-bit x 2, 5-bit x 1
Input port	P85	1 bit x 1
Multifunction timer	TA0, TA1, TA2, TA3, TA4	16 bits x 5
	TB0, TB1, TB2, TB3, TB4, TB5	16 bits x 6
Serial I/O	UART0, UART1, UART2, UART3, UART4	(UART or clock synchronous) x 5
A/D converter		10 bits x (8 + 2) channels
D/A converter		8 bits x 2
DMAC		4 channels
DRAM controller		CAS before RAS refresh, self-refresh, EDO, FP
CRC calculation circuit		CRC-CCITT
XY converter		16 bits X 16 bits
Watchdog timer		15 bits x 1 (with prescaler)
Interrupt		29 internal and 8 external sources, 5 software sources, 7 levels
Clock generating circuit		2 built-in clock generation circuits (built-in feedback resistance, and external ceramic or quartz oscillator)
Supply voltage		4.2 to 5.5V (f(XIN)=20MHz) Mask ROM, external ROM and flash memory versions 2.7 to 5.5V (f(XIN)=10MHz) Mask ROM, external ROM and flash memory versions
Power consumption		45mA (f(XIN) = 20MHz without software wait, Vcc=5V) Mask ROM 128 Kbytes version
I/O characteristics	I/O withstand voltage	5V
	Output current	5mA
Memory expansion		Available (up to 16M bytes)
Operating ambient temperature		-40 to 85°C
Device configuration		CMOS high performance silicon gate
Package		100-pin and 144-pin plastic mold QFP

Renesas plans to release the following products in the M16C/80 group:

- (1) Support for mask ROM version, external ROM version and flash memory version
- (2) ROM capacity
- (3) Package
  - 100P6S-A : Plastic molded QFP (mask ROM version and flash memory version)
  - 100P6Q-A : Plastic molded QFP (mask ROM version and flash memory version)
  - 144P6Q-A : Plastic molded QFP (mask ROM version and flash memory version)

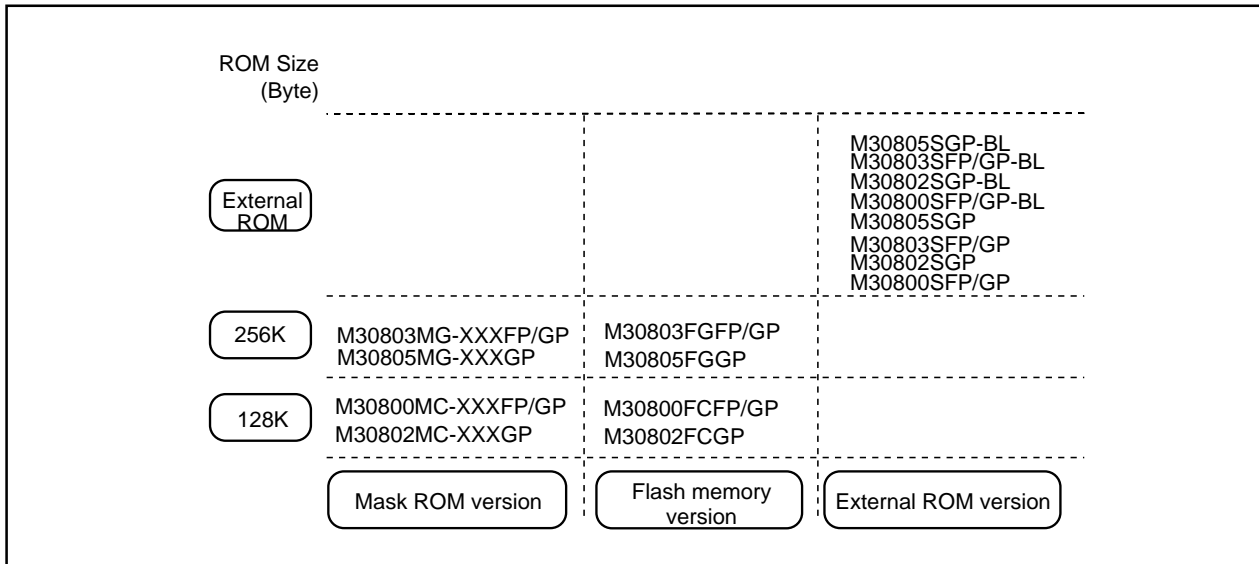


Figure 1.5 ROM expansion

The M16C/80 group products currently supported are listed in Table 1.2.

Table 1.2 M16C/80 group

Type No	ROM capacity	RAM capacity	Package type	Remarks
M30800MC-XXXFP	128K bytes	10K bytes	100P6S-A	Mask ROM version
M30800MC-XXXGP			100P6Q-A	
M30802MC-XXXGP			144P6Q-A	
M30803MG-XXXFP	256K bytes	20K bytes	100P6S-A	
M30803MG-XXXGP			100P6Q-A	
M30805MG-XXXGP			144P6Q-A	
M30800FCFP	128K bytes	10K bytes	100P6S-A	Flash memory version
M30800FCGP			100P6Q-A	
M30802FCGP			144P6Q-A	
M30803FGFP	256K bytes	20K bytes	100P6S-A	
M30803FGGP			100P6Q-A	
M30805FGGP			144P6Q-A	
M30800SFP	—	10K bytes	100P6S-A	External ROM version
M30800SGP			100P6Q-A	
M30802SGP			144P6Q-A	
M30803SFP	—	24K bytes	100P6S-A	
M30803SGP			100P6Q-A	
M30805SGP			144P6Q-A	
M30800SFP-BL	—	10K bytes	100P6S-A	External ROM version with built-in boot loader
M30800SGP-BL			100P6Q-A	
M30802SGP-BL			144P6Q-A	
M30803SFP-BL	—	24K bytes	100P6S-A	
M30803SGP-BL			100P6Q-A	
M30805SGP-BL			144P6Q-A	

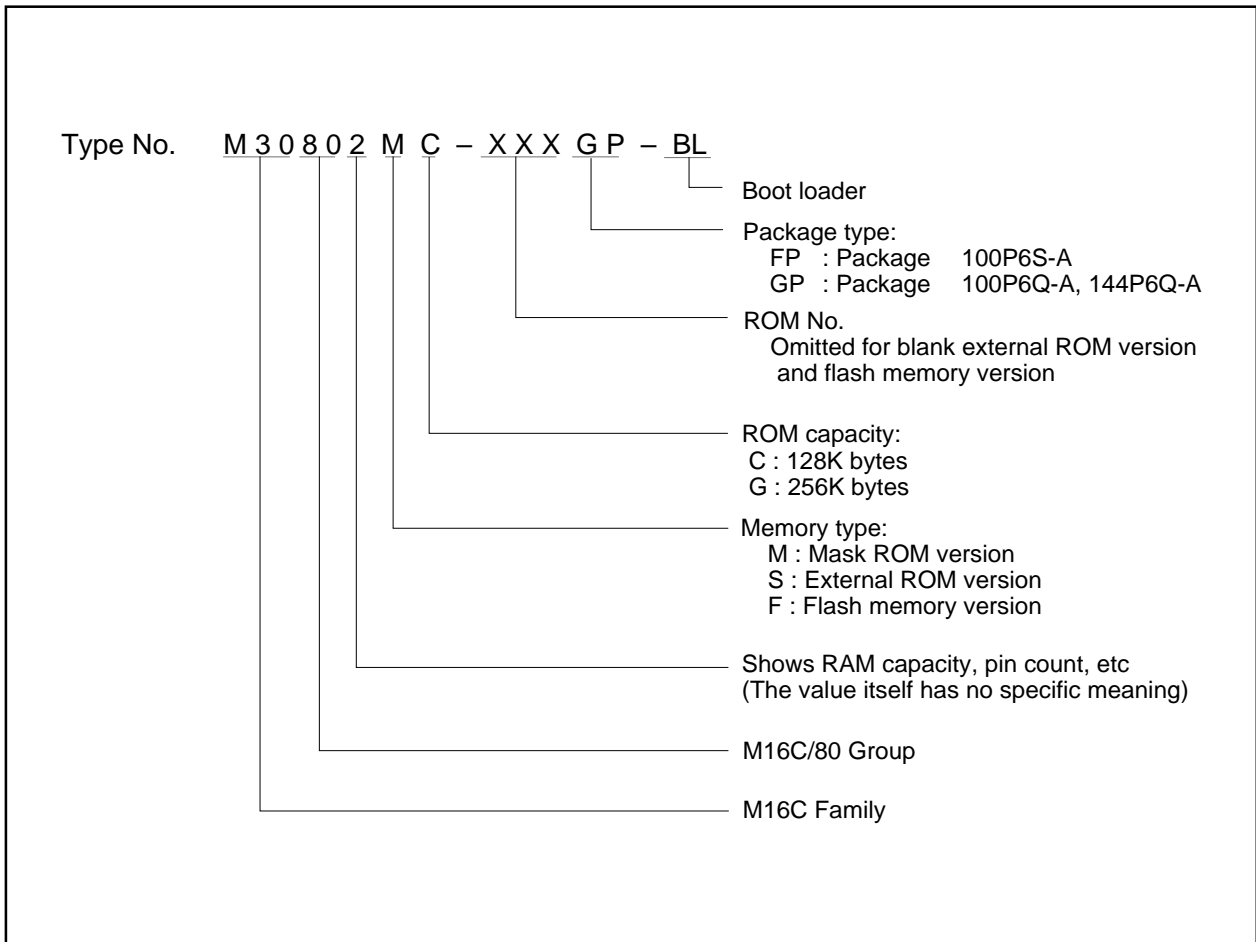


Figure 1.6 Product Numbering System

## 1.6 Pin Description (1)

Pin name	Signal name	I/O type	Function
Vcc, Vss	Power supply input		Supply 4.2 (2.7) to 5.5 V to the Vcc pin. Supply 0 V to the Vss pin.
CNVss	CNVss	I	This pin switches between processor modes. Connect it to the Vss when operating in single-chip or memory expansion mode after reset. Connect it to the Vcc when in microprocessor mode after reset.
$\overline{\text{RESET}}$	Reset input	I	An "L" on this input resets the microcomputer.
XIN XOUT	Clock input Clock output	I O	These pins are provided for the main clock generating circuit. Connect a ceramic resonator or crystal between the XIN and the XOUT pins. To use an externally derived clock, input it to the XIN pin and leave the XOUT pin open.
BYTE	External data bus width select input	I	This pin selects the width of an data bus in the external area 3. A 16-bit width is selected when this input is "L"; an 8-bit width is selected when this input is "H". This input must be fixed to either "H" or "L". When not using the external bus, connect this pin to Vss.
AVcc	Analog power supply input		This pin is a power supply input for the A/D converter. Connect this pin to Vcc.
AVss	Analog power supply input		This pin is a power supply input for the A/D converter. Connect this pin to Vss.
VREF	Reference voltage input	I	This pin is a reference voltage input for the A/D converter.
P00 to P07	I/O port P0	I/O	This is an 8-bit CMOS I/O port. It has an input/output port direction register that allows the user to set each pin for input or output individually. When set for input in single chip mode, the user can specify in units of four bits via software whether or not they are tied to a pull-up resistance. In memory expansion and microprocessor mode, an built-in pull-up resistance cannot be used. However, it is possible to select pull-up resistance presence to the usable port as I/O port by setting.
D0 to D7		I/O	When set as a separate bus, these pins input and output data (D0–D7).
P10 to P17	I/O port P1	I/O	This is an 8-bit I/O port equivalent to P0. P15 to P17 also function as external interrupt pins as selected by software.
D8 to D15		I/O	When set as a separate bus, these pins input and output data (D8–D15).
P20 to P27	I/O port P2	I/O	This is an 8-bit I/O port equivalent to P0.
A0 to A7		O	These pins output 8 low-order address bits (A0–A7).
A0/D0 to A7/D7		I/O	If a multiplexed bus is set, these pins input and output data (D0–D7) and output 8 low-order address bits (A0–A7) separated in time by multiplexing.
P30 to P37	I/O port P3	I/O	This is an 8-bit I/O port equivalent to P0.
A8 to A15		O	These pins output 8 middle-order address bits (A8–A15).
A8/D8 to A15/D15		I/O	If the external bus is set as a 16-bit wide multiplexed bus, these pins input and output data (D8–D15) and output 8 middle-order address bits (A8–A15) separated in time by multiplexing.
MA0 to MA7		O	If accessing to DRAM area, these pins output row address and column address separated in time by multiplexing.

## Pin Description (2)

Pin name	Signal name	I/O type	Function
P40 to P47	I/O port P4	I/O	This is an 8-bit I/O port equivalent to P0.
A16 to A22, A23		O	These pins output 8 high-order address bits (A16–A22, A23). Highest address bit (A23) outputs inversely.
CS0 to CS3		O	These pins output CS0–CS3 signals. CS0–CS3 are chip select signals used to specify an access space.
MA8 to MA12		O	If accessing to DRAM area, these pins output row address and column address separated in time by multiplexing.
P50 to P57	I/O port P5	I/O	This is an 8-bit I/O port equivalent to P0. P53 in this port outputs a divide-by-8 or divide-by-32 clock of XIN or a clock of the same frequency as XCIN as selected by software.
$\overline{WRL}$ / $\overline{WR}$ , $\overline{WRH}$ / BHE, RD, BCLK, HLDA, HOLD,  ALE, RDY		O O O O O I  O I	Output $\overline{WRL}$ , $\overline{WRH}$ ( $\overline{WR}$ and BHE), $\overline{RD}$ , BCLK, $\overline{HLDA}$ , and ALE signals. $\overline{WRL}$ and $\overline{WRH}$ , and BHE and $\overline{WR}$ can be switched using software control. <ul style="list-style-type: none"> <li>■ <math>\overline{WRL}</math>, <math>\overline{WRH}</math>, and <math>\overline{RD}</math> selected With a 16-bit external data bus, data is written to even addresses when the <math>\overline{WRL}</math> signal is "L" and to the odd addresses when the <math>\overline{WRH}</math> signal is "L". Data is read when <math>\overline{RD}</math> is "L".</li> <li>■ <math>\overline{WR}</math>, BHE, and <math>\overline{RD}</math> selected Data is written when <math>\overline{WR}</math> is "L". Data is read when <math>\overline{RD}</math> is "L". Odd addresses are accessed when BHE is "L". Use this mode when using an 8-bit external data bus.</li> </ul> While the input level at the HOLD pin is "L", the microcomputer is placed in the hold state. While in the hold state, HLDA outputs an "L" level. ALE is used to latch the address. While the input level of the RDY pin is "L", the bus of microcomputer is in the wait state.
$\overline{DW}$ , CASL, CASH, RAS		O O O O	When accessing to DRAM area while $\overline{DW}$ signal is "L", write to DRAM. CASL and CASH show timing when latching to line address. When CASL accesses to even address, and CASH to odd, these two pins become "L". RAS signal shows timing when latching to row address.
P60 to P67	I/O port P6	I/O	This is an 8-bit I/O port equivalent to P0. When set for input in single chip mode, microprocessor mode and memory expansion mode the user can specify in units of four bits via software whether or not they are tied to a pull-up resistance. Pins in this port also function as UART0 and UART1 I/O pins as selected by software.
P70 to P77	I/O port P7	I/O	This is an 8-bit I/O port equivalent to P6 (P70 and P71 are N-channel open drain output). Pins in this port also function as timer A0–A3, timer B5 or UART2 I/O pins as selected by software.
P80 to P84, P86, P87, P85	I/O port P8  I/O port P85	I/O I/O I/O I	P80 to P84, P86, and P87 are I/O ports with the same functions as P6. Using software, they can be made to function as the I/O pins for timer A4 and the input pins for external interrupts. P86 and P87 can be set using software to function as the I/O pins for a sub clock generation circuit. In this case, connect a quartz oscillator between P86 (XCOUT pin) and P87 (XCIN pin). P85 is an input-only port that also functions for NMI. The NMI interrupt is generated when the input at this pin changes from "H" to "L". The NMI function cannot be canceled using software. The pull-up cannot be set for this pin.
P90 to P97	I/O port P9	I/O	This is an 8-bit I/O port equivalent to P6. Pins in this port also function as UART3 and UART4 I/O pins, Timer B0–B4 input pins, D/A converter output pins, A/D converter extended input pins, or A/D trigger input pins as selected by software.
P100 to P107	I/O port P10	I/O	This is an 8-bit I/O port equivalent to P6. Pins in this port also function as A/D converter input pins. Furthermore, P104–P107 also function as input pins for the key input interrupt function.

**Pin Description (3)**

Pin name	Signal name	I/O type	Function
P110 to P114 (Note)	I/O port P11	II/O	This is an 5-bit I/O port equivalent to P6.
P120 to P127 (Note)	I/O port P12	II/O	This is an 8-bit I/O port equivalent to P6.
P130 to P137 (Note)	I/O port P13	II/O	This is an 8-bit I/O port equivalent to P6.
P140 to P146 (Note)	I/O port P14	II/O	This is an 7-bit I/O port equivalent to P6.
P150 to P157 (Note)	I/O port P15	II/O	This is an 8-bit I/O port equivalent to P6.

Note : Port P11 to P15 exist in 144-pin version.

**Operation of Functional Blocks**

The M16C/80 group accommodates certain units in a single chip. These units include ROM and RAM to store instructions and data and the central processing unit (CPU) to execute arithmetic/logic operations. Also included are peripheral units such as timers, serial I/O, D/A converter, DMAC, CRC calculation circuit, A/D converter, DRAM controller and I/O ports.

The following explains each unit.



## 2. Memory

Figure 2.1 is a memory map of the M16C/80 group. The address space extends the 16 Mbytes from address 000000<sub>16</sub> to FFFFFFF<sub>16</sub>. From FFFFFFF<sub>16</sub> down is ROM. For example, in the M30800MC-XXXFP, there is 128K bytes of internal ROM from FE0000<sub>16</sub> to FFFFFFF<sub>16</sub>. The vector table for fixed interrupts such as the reset and NMI are mapped to FFFFDC<sub>16</sub> to FFFFFFF<sub>16</sub>. The starting address of the interrupt routine is stored here. The address of the vector table for timer interrupts, etc., can be set as desired using the internal register (INTB). See the section on interrupts for details.

From 000400<sub>16</sub> up is RAM. For example, in the M30800MC-XXXFP, 10 Kbytes of internal RAM is mapped to the space from 000400<sub>16</sub> to 002BFF<sub>16</sub>. In addition to storing data, the RAM also stores the stack used when calling subroutines and when interrupts are generated.

The SFR area is mapped to 000000<sub>16</sub> to 0003FF<sub>16</sub>. This area accommodates the control registers for peripheral devices such as I/O ports, A/D converter, serial I/O, and timers, etc. Figures 5.1 to 5.4 are location of peripheral unit control registers. Any part of the SFR area that is not occupied is reserved and cannot be used for other purposes.

The special page vector table is mapped to FFFE00<sub>16</sub> to FFFFDB<sub>16</sub>. If the starting addresses of subroutines or the destination addresses of jumps are stored here, subroutine call instructions and jump instructions can be used as 2-byte instructions, reducing the number of program steps.

In memory expansion mode and microprocessor mode, a part of the spaces are reserved and cannot be used. For example, in the M30800MC-XXXFP, the following spaces cannot be used.

- The space between 002C00<sub>16</sub> and 008000<sub>16</sub> (Memory expansion and microprocessor modes)
- The space between F00000<sub>16</sub> and FDFFFF<sub>16</sub> (Memory expansion mode)

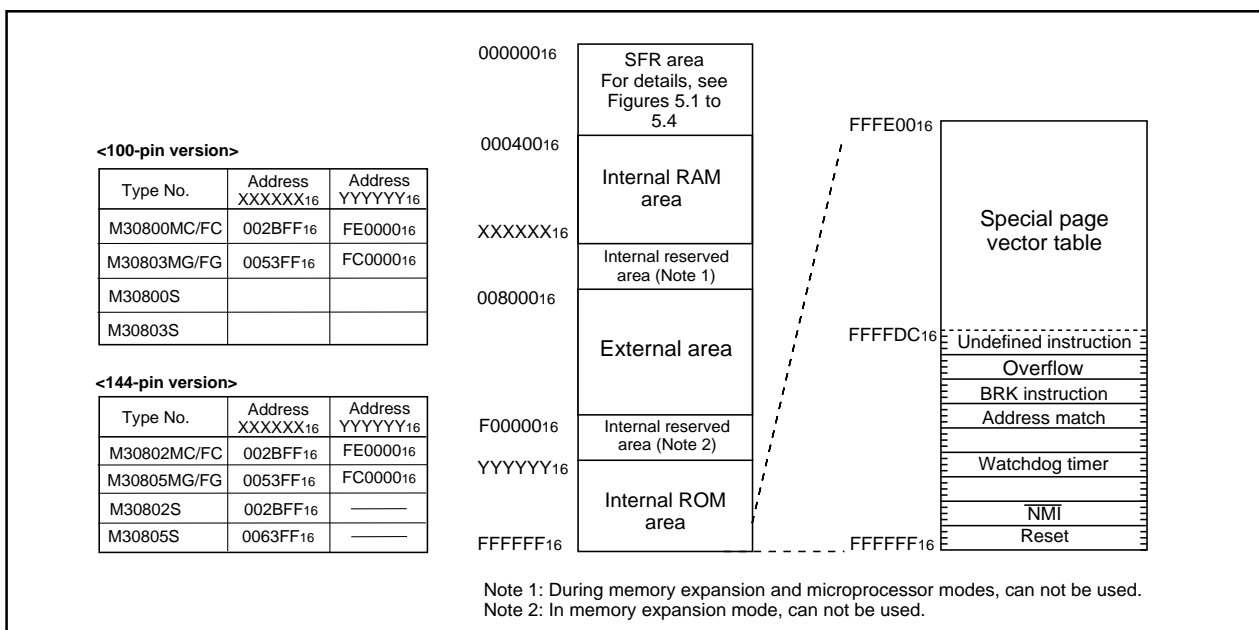


Figure 2.1 Memory map

### 3. Central Processing Unit (CPU)

The CPU has a total of 28 registers shown in Figure 3.1. Eight of these registers (R0, R1, R2, R3, A0, A1, SB and FB) come in two sets; therefore, these have two register banks.

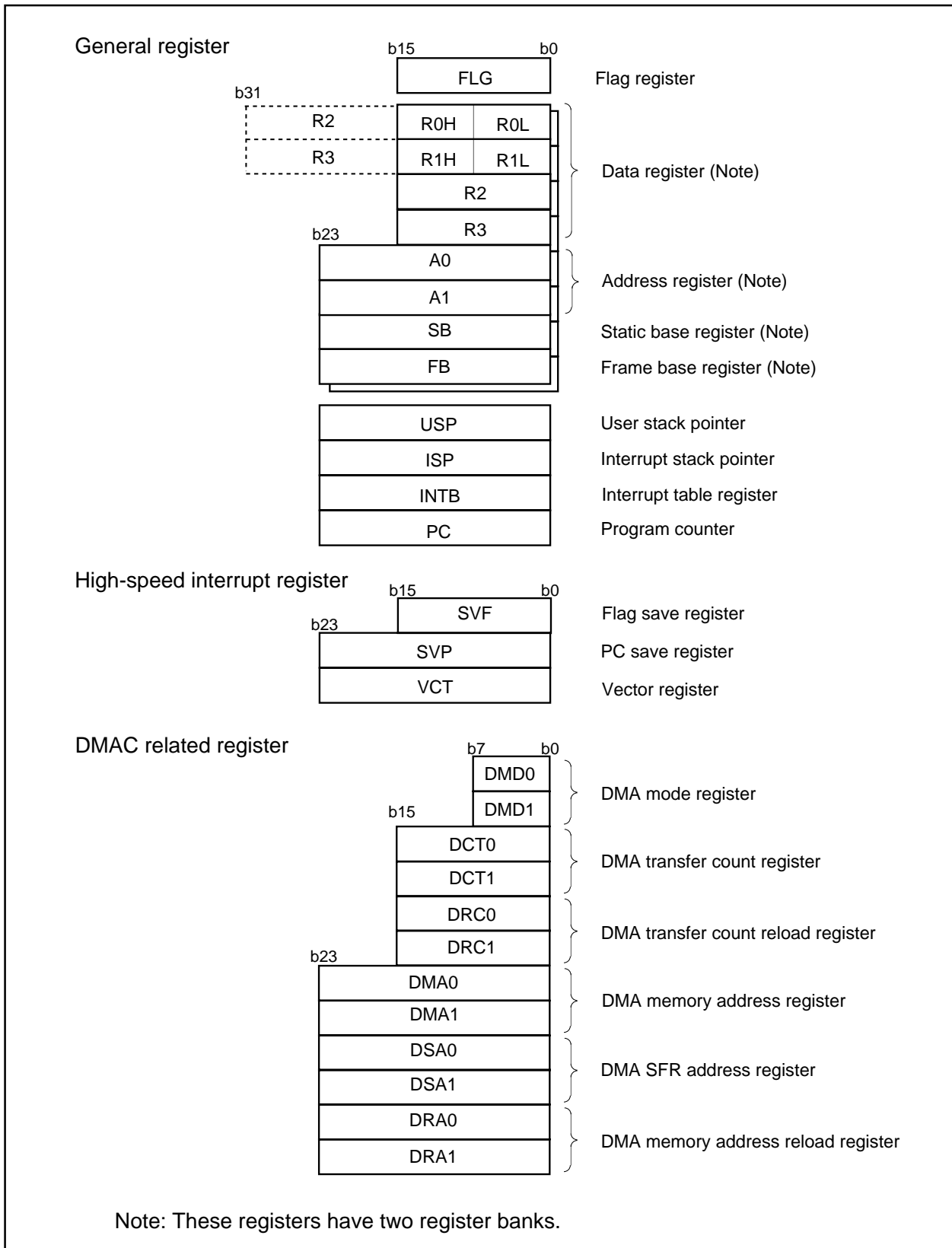


Figure 3.1 Central processing unit register

**(1) Data registers (R0, R0H, R0L, R1, R1H, R1L, R2, R3, R2R0 and R3R1)**

Data registers (R0, R1, R2, and R3) are configured with 16 bits, and are used primarily for transfer and arithmetic/logic operations.

Registers R0 and R1 each can be used as separate 8-bit data registers, high-order bits as (R0H/R1H), and low-order bits as (R0L/R1L). Registers R2 and R0, as well as R3 and R1 can use as 32-bit data registers (R2R0/R3R1).

**(2) Address registers (A0 and A1)**

Address registers (A0 and A1) are configured with 24 bits, and have functions equivalent to those of data registers. These registers can also be used for address register indirect addressing and address register relative addressing.

**(3) Static base register (SB)**

Static base register (SB) is configured with 24 bits, and is used for SB relative addressing.

**(4) Frame base register (FB)**

Frame base register (FB) is configured with 24 bits, and is used for FB relative addressing.

**(5) Program counter (PC)**

Program counter (PC) is configured with 24 bits, indicating the address of an instruction to be executed.

**(6) Interrupt table register (INTB)**

Interrupt table register (INTB) is configured with 24 bits, indicating the start address of an interrupt vector table.

**(7) User stack pointer (USP), interrupt stack pointer (ISP)**

Stack pointer comes in two types: user stack pointer (USP) and interrupt stack pointer (ISP), each configured with 24 bits.

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

This flag is located at the position of bit 7 in the flag register (FLG).

Set USP and ISP to an even number so that execution efficiency is increased.

**(8) Save flag register (SVF)**

This register consists of 16 bits and is used to save the flag register when a high-speed interrupt is generated.

**(9) Save PC register (SVP)**

This register consists of 24 bits and is used to save the program counter when a high-speed interrupt is generated.

**(10) Vector register (VCT)**

This register consists of 24 bits and is used to indicate the jump address when a high-speed interrupt is generated.

**(11) DMA mode registers (DMD0/DMD1)**

These registers consist of 8 bits and are used to set the transfer mode, etc. for DMA.

**(12) DMA transfer count registers (DCT0/DCT1)**

These registers consist of 16 bits and are used to set the number of DMA transfers performed.

**(13) DMA transfer count reload registers (DRC0/DRC1)**

These registers consist of 16 bits and are used to reload the DMA transfer count registers.

**(14) DMA memory address registers (DMA0/DMA1)**

These registers consist of 24 bits and are used to set a memory address at the source or destination of DMA transfer.

**(15) DMA SFR address registers (DSA0/DSA1)**

These registers consist of 24 bits and are used to set a fixed address at the source or destination of DMA transfer.

**(16) DMA memory address reload registers (DRA0/DRA1)**

These registers consist of 24 bits and are used to reload the DMA memory address registers.

### (17) Flag register (FLG)

Flag register (FLG) is configured with 11 bits, each bit is used as a flag. Figure 3.2 shows the flag register (FLG). The following explains the function of each flag:

- **Bit 0: Carry flag (C flag)**

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

- **Bit 1: Debug flag (D flag)**

This flag enables a single-step interrupt.

When this flag is “1”, a single-step interrupt is generated after instruction execution. This flag is cleared to “0” when the interrupt is acknowledged.

- **Bit 2: Zero flag (Z flag)**

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

- **Bit 3: Sign flag (S flag)**

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

- **Bit 4: Register bank select flag (B flag)**

This flag chooses a register bank. Register bank 0 is selected when this flag is “0”; register bank 1 is selected when this flag is “1”.

- **Bit 5: Overflow flag (O flag)**

This flag is set to “1” when an arithmetic operation resulted in overflow; otherwise, cleared to “0”.

- **Bit 6: Interrupt enable flag (I flag)**

This flag enables a maskable interrupt.

An interrupt is disabled when this flag is “0”, and is enabled when this flag is “1”. This flag is cleared to “0” when the interrupt is acknowledged.

- **Bit 7: Stack pointer select flag (U flag)**

Interrupt stack pointer (ISP) is selected when this flag is “0”; user stack pointer (USP) is selected when this flag is “1”.

This flag is cleared to “0” when a hardware interrupt is acknowledged or an INT instruction of software interrupt Nos. 0 to 31 is executed.

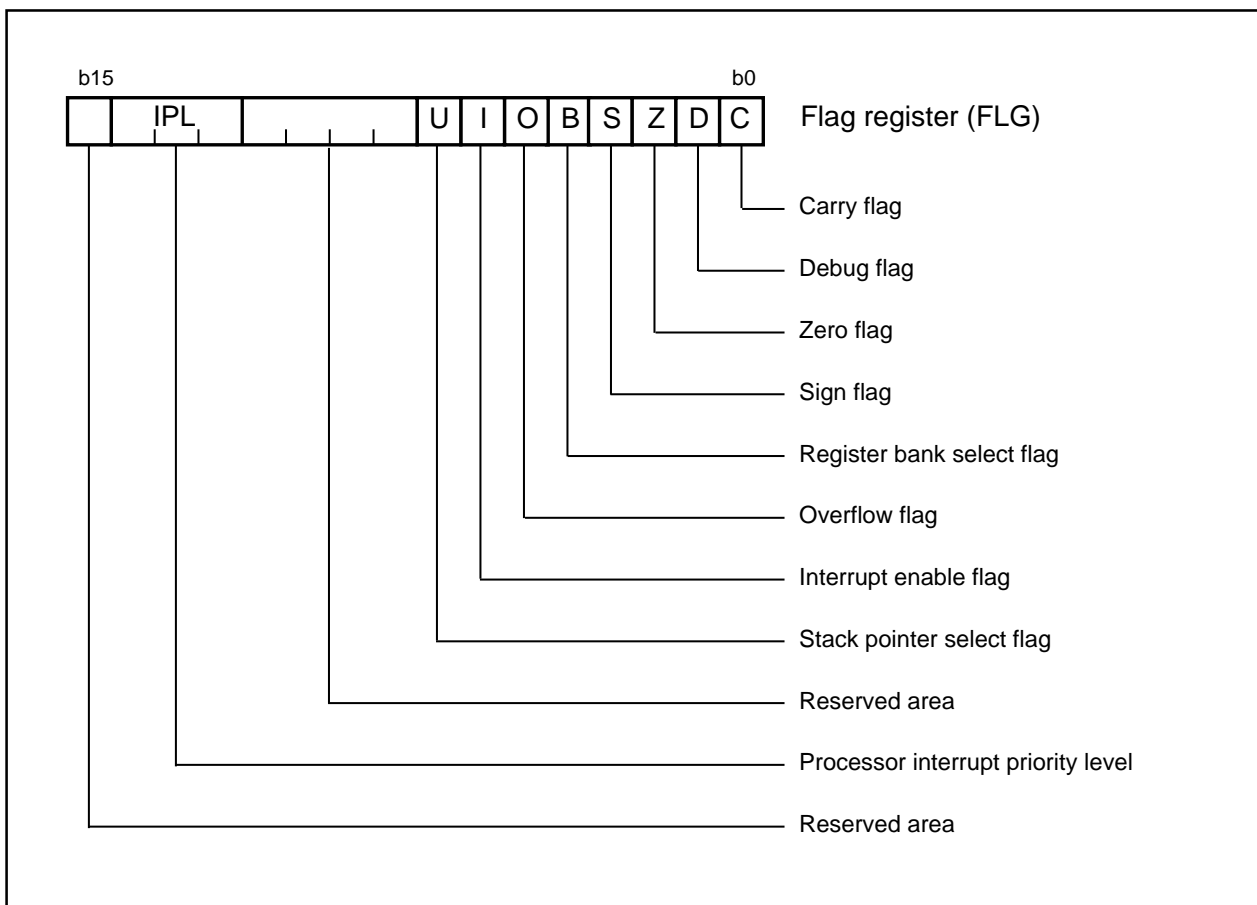
- **Bits 8 to 11: Reserved area**

- **Bits 12 to 14: Processor interrupt priority level (IPL)**

Processor interrupt priority level (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 the processor interrupt priority level (IPL), the interrupt is enabled.

- **Bit 15: Reserved area**



**Figure 3.2 Flag register (FLG)**

### 4. Reset

There are two kinds of resets; hardware and software. In both cases, operation is the same after the reset. (See "Software Reset" for details of software resets.) This section explains on hardware resets.

When the supply voltage is in the range where operation is guaranteed, a reset is effected by holding the reset pin level "L" (0.2V<sub>CC</sub> max.) for at least 20 cycles. When the reset pin level is then returned to the "H" level while main clock is stable, the reset status is cancelled and program execution resumes from the address in the reset vector table.

Figure 4.1 shows the example reset circuit. Figure 4.2 shows the reset sequence.

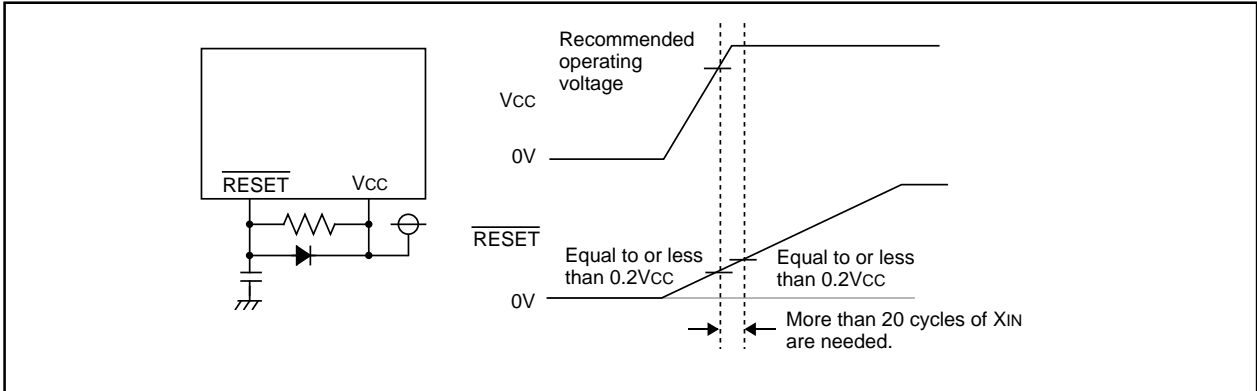


Figure 4.1 Example reset circuit

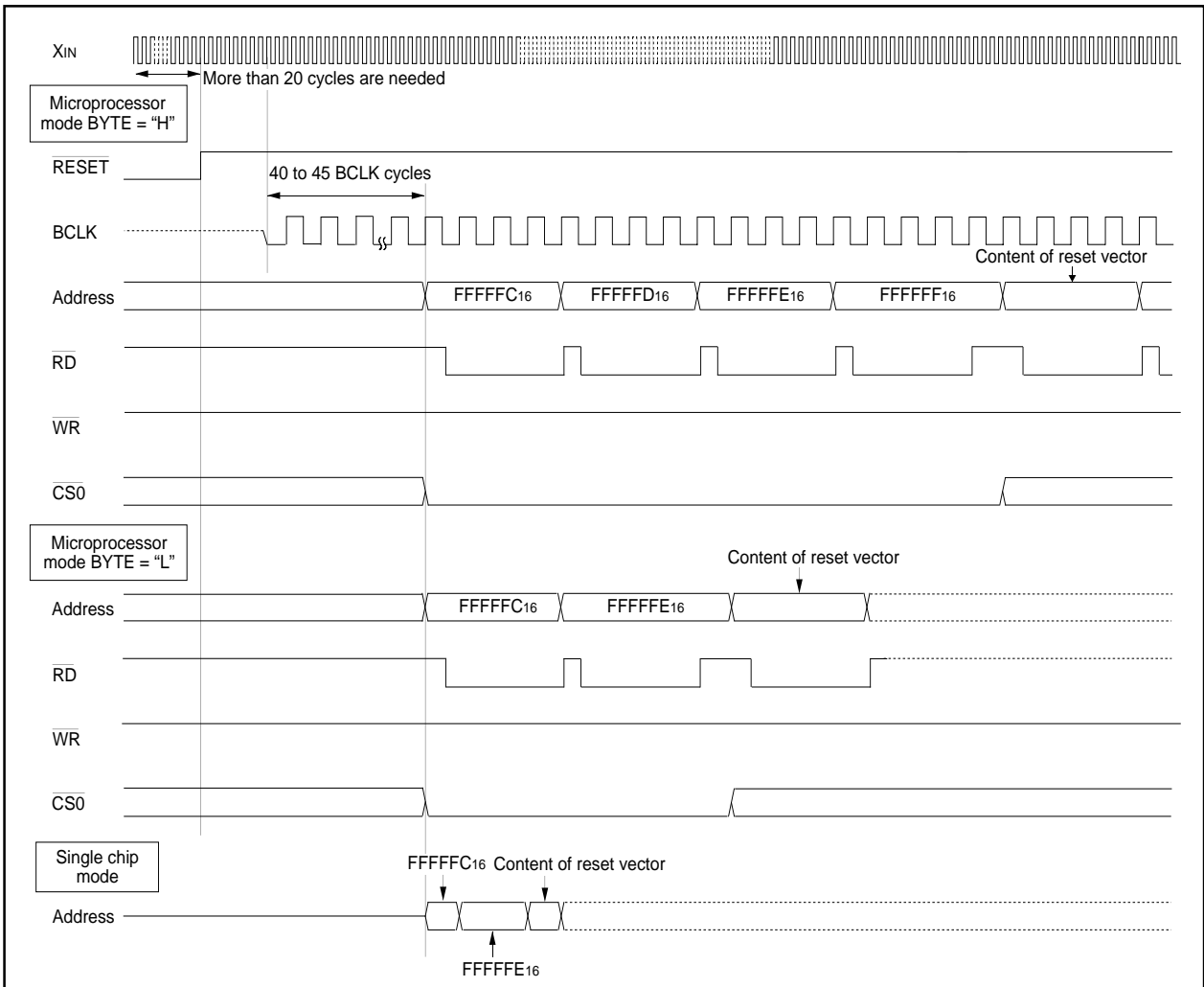


Figure 4.2 Reset sequence

Table 4.1 shows the statuses of the other pins while the  $\overline{\text{RESET}}$  pin level is “L”. Figures 4.3 and 4.4 show the internal status of the microcomputer immediately after the reset is cancelled.

**Table 4.1 Pin status when  $\overline{\text{RESET}}$  pin level is “L”**

Pin name	Status		
	CNVss = Vss	CNVss = Vcc	
		BYTE = Vss	BYTE = Vcc
P0	Input port (floating)	Data input (floating)	Data input (floating)
P1	Input port (floating)	Data input (floating)	Input port (floating)
P2, P3, P4	Input port (floating)	Address output (undefined)	Address output (undefined)
P50	Input port (floating)	$\overline{\text{WR}}$ output (“H” level is output)	$\overline{\text{WR}}$ output (“H” level is output)
P51	Input port (floating)	$\overline{\text{BHE}}$ output (undefined)	$\overline{\text{BHE}}$ output (undefined)
P52	Input port (floating)	$\overline{\text{RD}}$ output (“H” level is output)	$\overline{\text{RD}}$ output (“H” level is output)
P53	Input port (floating)	BCLK output	BCLK output
P54	Input port (floating)	$\overline{\text{HLDA}}$ output (The output value depends on the input to the HOLD pin)	$\overline{\text{HLDA}}$ output (The output value depends on the input to the HOLD pin)
P55	Input port (floating)	$\overline{\text{HOLD}}$ input (floating)	$\overline{\text{HOLD}}$ input (floating)
P56	Input port (floating)	$\overline{\text{RAS}}$ output	$\overline{\text{RAS}}$ output
P57	Input port (floating)	$\overline{\text{RDY}}$ input (floating)	$\overline{\text{RDY}}$ input (floating)
P6, P7, P80 to P84, P86, P87, P9, P10,	Input port (floating)	Input port (floating)	Input port (floating)
P11, P12, P13, P14, P15 (Note)	Input port (floating)	Input port (floating)	Input port (floating)

Note :Port P11 to P15 exist in 144-pin vrsion.



(1) Processor mode register 0 (Note1)	(000416)...	8016	(30) Timer B3 interrupt control register	(007816)...	XXXX?000
(2) Processor mode register 1	(000516)...	0016	(31) INT5 interrupt control register	(007A16)...	XX00?000
(3) System clock control register 0	(000616)...	0816	(32) INT3 interrupt control register	(007C16)...	XX00?000
(4) System clock control register 1	(000716)...	2016	(33) INT1 interrupt control register	(007E16)...	XX00?000
(5) Wait control register	(000816)...	FF16	(34) DMA1 interrupt control register	(008816)...	XXXX?000
(6) Address match interrupt enable register	(000916)...	XXXX0000	(35) UART2 transmit/NACK interrupt control register	(008916)...	XXXX?000
(7) Protect register	(000A16)...	XXXX0000	(36) DMA3 interrupt control register	(008A16)...	XXXX?000
(8) External data bus width control register (Note 2)	(000B16)...	XXXX?000	(37) UART3 transmit/NACK interrupt control register	(008B16)...	XXXX?000
(9) Main clock divided register	(000C16)...	XXXX0100	(38) Timer A1 interrupt control register	(008C16)...	XXXX?000
(10) Watchdog timer control register	(000F16)...	000???	(39) UART4 receive/NACK interrupt control register	(008D16)...	XXXX?000
(11) Address match interrupt register 0	(001016)...	0016	(40) Timer A3 interrupt control register	(008E16)...	XXXX?000
	(001116)...	0016	(41) Bus collision detection(UART2) interrupt control register	(008F16)...	XXXX?000
	(001216)...	0016	(42) UART0 transmit interrupt control register	(009016)...	XXXX?000
(12) Address match interrupt register 1	(001416)...	0016	(43) Bus collision detection(UART4) interrupt control register	(009116)...	XXXX?000
	(001516)...	0016	(44) UART1 transmit interrupt control register	(009216)...	XXXX?000
	(001616)...	0016	(45) Key input interrupt control register	(009316)...	XXXX?000
(13) Address match interrupt register 2	(001816)...	0016	(46) Timer B0 interrupt control register	(009416)...	XXXX?000
	(001916)...	0016	(47) Timer B2 interrupt control register	(009616)...	XXXX?000
	(001A16)...	0016	(48) Timer B4 interrupt control register	(009816)...	XXXX?000
(14) Address match interrupt register 3	(001C16)...	0016	(49) INT4 interrupt control register	(009A16)...	XX00?000
	(001D16)...	0016	(50) INT2 interrupt control register	(009C16)...	XX00?000
	(001E16)...	0016	(51) INTO interrupt control register	(009E16)...	XX00?000
(15) DMAM control register	(004016)...	?XXXX???	(52) Exit priority register	(009F16)...	XXXX0000
(16) DMA0 interrupt control register	(006816)...	XXXX?000	(53) XY control register	(02E016)...	XXXXXXXX00
(17) Timer B5 interrupt control register	(006916)...	XXXX?000	(54) UART4 special mode register 3	(02F516)...	0016
(18) DMA2 interrupt control register	(006A16)...	XXXX?000	(55) UART4 special mode register 2	(02F616)...	0016
(19) UART2 receive/ACK interrupt control register	(006B16)...	XXXX?000	(56) UART4 special mode register	(02F716)...	0016
(20) Timer A0 interrupt control register	(006C16)...	XXXX?000	(57) UART4 transmit/receive mode register	(02F816)...	0016
(21) UART3 receive/ACK interrupt control register	(006D16)...	XXXX?000	(58) UART4 transmit/receive control register 0	(02FC16)...	0816
(22) Timer A2 interrupt control register	(006E16)...	XXXX?000	(59) UART4 transmit/receive control register 1	(02FD16)...	0216
(23) UART4 receive/ACK interrupt control register	(006F16)...	XXXX?000	(60) Timer B3,4,5 count start flag	(030016)...	000XXXXX
(24) Timer A4 interrupt control register	(007016)...	XXXX?000	(61) Three-phase PWM control register 0	(030816)...	0016
(25) Bus collision detection(UART3) interrupt control register	(007116)...	XXXX?000	(62) Three-phase PWM control register 1	(030916)...	0000?000
(26) UART0 receive interrupt control register	(007216)...	XXXX?000	(63) Three-phase output buffer register 0	(030A16)...	3F16
(27) A/D conversion interrupt control register	(007316)...	XXXX?000	(64) Three-phase output buffer register 1	(030B16)...	3F16
(28) UART1 receive interrupt control register	(007416)...	XXXX?000	(65) Timer B3 mode register	(031B16)...	00??0000
(29) Timer B1 interrupt control register	(007616)...	XXXX?000	(66) Timer B4 mode register	(031C16)...	00?X0000
			(67) Timer B5 mode register	(031D16)...	00?X0000

x : Nothing is mapped to this bit  
 ? : Undefined

The content of other registers and RAM is undefined when the microcomputer is reset. The initial values must therefore be set.

Note 1: When the VCC level is applied to the CNVSS pin, it is 0316 at a reset.  
 Note 2: When the BYTE pin is "L", the third bit is "1". When the BYTE pin is "H", the third bit is "0".

Figure 4.3 Device's internal status after a reset is cleared

(68) Interrupt cause select register	(031F <sub>16</sub> )...	00000000	(112) Function select register A0	(03B0 <sub>16</sub> )...	00000000
(69) UART3 special mode register 3	(0325 <sub>16</sub> )...	00 <sub>16</sub>	(113) Function select register A1	(03B1 <sub>16</sub> )...	00000000
(70) UART3 special mode register 2	(0326 <sub>16</sub> )...	00 <sub>16</sub>	(114) Function select register B0	(03B2 <sub>16</sub> )...	XXXXXXXX
(71) UART3 special mode register	(0327 <sub>16</sub> )...	00 <sub>16</sub>	(115) Function select register B1	(03B3 <sub>16</sub> )...	XXXXXXXX00
(72) UART3 transmit/receive mode register	(0328 <sub>16</sub> )...	00 <sub>16</sub>	(116) Function select register A2	(03B4 <sub>16</sub> )...	XXXXXXXX00
(73) UART3 transmit/receive control register 0	(032C <sub>16</sub> )...	08 <sub>16</sub>	(117) Function select register A3	(03B5 <sub>16</sub> )...	00 <sub>16</sub>
(74) UART3 transmit/receive control register 1	(032D <sub>16</sub> )...	02 <sub>16</sub>	(118) Function select register B2	(03B6 <sub>16</sub> )...	XXXXXXXX00
(75) UART2 special mode register 3	(0335 <sub>16</sub> )...	000XXXXX	(119) Function select register B3	(03B7 <sub>16</sub> )...	00000000
(76) UART2 special mode register 2	(0336 <sub>16</sub> )...	00 <sub>16</sub>	(120) Port P6 direction register	(03C2 <sub>16</sub> )...	00 <sub>16</sub>
(77) UART2 special mode register	(0337 <sub>16</sub> )...	00 <sub>16</sub>	(121) Port P7 direction register	(03C3 <sub>16</sub> )...	00 <sub>16</sub>
(78) UART2 transmit/receive mode register	(0338 <sub>16</sub> )...	00 <sub>16</sub>	(122) Port P8 direction register	(03C6 <sub>16</sub> )...	00XXXX00
(79) UART2 transmit/receive control register 0	(033C <sub>16</sub> )...	00X01000	(123) Port P9 direction register	(03C7 <sub>16</sub> )...	00 <sub>16</sub>
(80) UART2 transmit/receive control register 1	(033D <sub>16</sub> )...	02 <sub>16</sub>	(124) Port P10 direction register	(03CA <sub>16</sub> )...	00 <sub>16</sub>
(81) Count start flag	(0340 <sub>16</sub> )...	00 <sub>16</sub>	(125) Port P11 direction register (Note 2)	(03CB <sub>16</sub> )...	XXXX0000
(82) Clock prescaler reset flag	(0341 <sub>16</sub> )...	0XXXXXXX	(126) Port P12 direction register (Note 2)	(03CE <sub>16</sub> )...	00 <sub>16</sub>
(83) One-shot start flag	(0342 <sub>16</sub> )...	00 <sub>16</sub>	(127) Port P13 direction register (Note 2)	(03CF <sub>16</sub> )...	00 <sub>16</sub>
(84) Trigger select flag	(0343 <sub>16</sub> )...	00 <sub>16</sub>	(128) Port P14 direction register (Note 2)	(03D2 <sub>16</sub> )...	XXXX0000
(85) Up-down flag	(0344 <sub>16</sub> )...	00 <sub>16</sub>	(129) Port P15 direction register (Note 2)	(03D3 <sub>16</sub> )...	00 <sub>16</sub>
(86) Timer A0 mode register	(0356 <sub>16</sub> )...	00000000	(130) Pull-up control register 2	(03DA <sub>16</sub> )...	00 <sub>16</sub>
(87) Timer A1 mode register	(0357 <sub>16</sub> )...	00000000	(131) Pull-up control register 3 (Note 2)	(03DB <sub>16</sub> )...	00 <sub>16</sub>
(88) Timer A2 mode register	(0358 <sub>16</sub> )...	00000000	(132) Pull-up control register 4 (Note 2)	(03DC <sub>16</sub> )...	X0 <sub>16</sub>
(89) Timer A3 mode register	(0359 <sub>16</sub> )...	00000000	(133) Port P0 direction register	(03E2 <sub>16</sub> )...	00 <sub>16</sub>
(90) Timer A4 mode register	(035A <sub>16</sub> )...	00000000	(132) Port P1 direction register	(03E3 <sub>16</sub> )...	00 <sub>16</sub>
(91) Timer B0 mode register	(035B <sub>16</sub> )...	00XXXX00	(135) Port P2 direction register	(03E6 <sub>16</sub> )...	00 <sub>16</sub>
(92) Timer B1 mode register	(035C <sub>16</sub> )...	00XXXX00	(136) Port P3 direction register	(03E7 <sub>16</sub> )...	00 <sub>16</sub>
(93) Timer B2 mode register	(035D <sub>16</sub> )...	00XXXX00	(137) Port P4 direction register	(03EA <sub>16</sub> )...	00 <sub>16</sub>
(94) UART0 transmit/receive mode register	(0360 <sub>16</sub> )...	00 <sub>16</sub>	(138) Port P5 direction register	(03EB <sub>16</sub> )...	00 <sub>16</sub>
(95) UART0 transmit/receive control register 0	(0364 <sub>16</sub> )...	08 <sub>16</sub>	(139) Pull-up control register 0	(03F0 <sub>16</sub> )...	00 <sub>16</sub>
(96) UART0 transmit/receive control register 1	(0365 <sub>16</sub> )...	02 <sub>16</sub>	(140) Pull-up control register 1	(03F1 <sub>16</sub> )...	X0 <sub>16</sub>
(97) UART1 transmit/receive mode register	(0368 <sub>16</sub> )...	00 <sub>16</sub>	(141) Port control register	(03FF <sub>16</sub> )...	XXXXXXXX00
(98) UART1 transmit/receive control register 0	(036C <sub>16</sub> )...	08 <sub>16</sub>	(142) Data registers (R0/R1/R2/R3)		0000 <sub>16</sub>
(99) UART1 transmit/receive control register 1	(036D <sub>16</sub> )...	02 <sub>16</sub>	(143) Address registers (A0/A1)		000000 <sub>16</sub>
(100) UART transmit/receive control register 2	(0370 <sub>16</sub> )...	XXXX0000	(144) Static base register (SB)		000000 <sub>16</sub>
(101) Flash memory control register 1 (Note 1)	(0376 <sub>16</sub> )...	XXXXXX	(145) Frame base register (FB)		000000 <sub>16</sub>
(102) Flash memory control register 0 (Note 1)	(0377 <sub>16</sub> )...	XXXX0001	(146) Interrupt table register (INTB)		000000 <sub>16</sub>
(103) DMA0 cause select register	(0378 <sub>16</sub> )...	XXXX0000	(147) User stack pointer (USP)		000000 <sub>16</sub>
(104) DMA1 cause select register	(0379 <sub>16</sub> )...	XXXX0000	(148) Interrupt stack pointer (ISP)		000000 <sub>16</sub>
(105) DMA2 cause select register	(037A <sub>16</sub> )...	XXXX0000	(149) Flag register (FLG)		0000 <sub>16</sub>
(106) DMA3 cause select register	(037B <sub>16</sub> )...	XXXX0000	(150) DMA mode register (DMD0/DMD1)		00 <sub>16</sub>
(107) A/D control register 2	(0394 <sub>16</sub> )...	0000XXXX	(151) DMA transfer count register (DCT0/DCT1)		??
(108) A/D control register 0	(0396 <sub>16</sub> )...	000000??	(152) DMA transfer count reload register (DRC0/DRC1)		??
(109) A/D control register 1	(0397 <sub>16</sub> )...	00 <sub>16</sub>	(153) DMA memory address register (DMA0/DMA1)		??
(110) D/A control register	(039C <sub>16</sub> )...	00 <sub>16</sub>	(154) DMA SFR address register (DSA0/DSA1)		??
(111) Function select register C	(03AF <sub>16</sub> )...	XXXXXX00	(155) DMA memory address reload register (DRA0/DRA1)		??

The content of other registers and RAM is undefined when the microcomputer is reset. The initial values must therefore be set.

Note 1: This register exists in the flash memory version.  
Note 2: This register exists in 144-pin version.

x : Nothing is mapped to this bit  
? : Undefined

Figure 4.4 Device's internal status after a reset is cleared

## 5. SFR

0000 <sub>16</sub>		0060 <sub>16</sub>	
0001 <sub>16</sub>		0061 <sub>16</sub>	
0002 <sub>16</sub>		0062 <sub>16</sub>	
0003 <sub>16</sub>		0063 <sub>16</sub>	
0004 <sub>16</sub>	Processor mode register 0 (PM0)	0064 <sub>16</sub>	
0005 <sub>16</sub>	Processor mode register 1 (PM1)	0065 <sub>16</sub>	
0006 <sub>16</sub>	System clock control register 0 (CM0)	0066 <sub>16</sub>	
0007 <sub>16</sub>	System clock control register 1 (CM1)	0067 <sub>16</sub>	
0008 <sub>16</sub>	Wait control register (WCR)	0068 <sub>16</sub>	DMA0 interrupt control register (DM0IC)
0009 <sub>16</sub>	Address match interrupt enable register (AIER)	0069 <sub>16</sub>	Timer B5 interrupt control register (TB5IC)
000A <sub>16</sub>	Protect register (PRCR)	006A <sub>16</sub>	DMA2 interrupt control register (DM2IC)
000B <sub>16</sub>	External data bus width control register (DS)	006B <sub>16</sub>	UART2 receive/ACK interrupt control register (S2RIC)
000C <sub>16</sub>	Main clock division register (MCD)	006C <sub>16</sub>	Timer A0 interrupt control register (TA0IC)
000D <sub>16</sub>		006D <sub>16</sub>	UART3 receive/ACK interrupt control register (S3RIC)
000E <sub>16</sub>	Watchdog timer start register (WDTS)	006E <sub>16</sub>	Timer A2 interrupt control register (TA2IC)
000F <sub>16</sub>	Watchdog timer control register (WDC)	006F <sub>16</sub>	UART4 receive/ACK interrupt control register (S4RIC)
0010 <sub>16</sub>		0070 <sub>16</sub>	Timer A4 interrupt control register (TA4IC)
0011 <sub>16</sub>	Address match interrupt register 0 (RMAD0)	0071 <sub>16</sub>	Bus collision detection(UART3) interrupt control register (BCN3IC)
0012 <sub>16</sub>		0072 <sub>16</sub>	UART0 receive interrupt control register (S0RIC)
0013 <sub>16</sub>		0073 <sub>16</sub>	A/D conversion interrupt control register (ADIC)
0014 <sub>16</sub>		0074 <sub>16</sub>	UART1 receive interrupt control register (S1RIC)
0015 <sub>16</sub>	Address match interrupt register 1 (RMAD1)	0075 <sub>16</sub>	
0016 <sub>16</sub>		0076 <sub>16</sub>	Timer B1 interrupt control register (TB1IC)
0017 <sub>16</sub>		0077 <sub>16</sub>	
0018 <sub>16</sub>		0078 <sub>16</sub>	Timer B3 interrupt control register (TB3IC)
0019 <sub>16</sub>	Address match interrupt register 2 (RMAD2)	0079 <sub>16</sub>	
001A <sub>16</sub>		007A <sub>16</sub>	INT5 interrupt control register (INT5IC)
001B <sub>16</sub>		007B <sub>16</sub>	
001C <sub>16</sub>		007C <sub>16</sub>	INT3 interrupt control register (INT3IC)
001D <sub>16</sub>	Address match interrupt register 3 (RMAD3)	007D <sub>16</sub>	
001E <sub>16</sub>		007E <sub>16</sub>	INT1 interrupt control register (INT1IC)
001F <sub>16</sub>		007F <sub>16</sub>	
0020 <sub>16</sub>		0080 <sub>16</sub>	
0021 <sub>16</sub>	Emulator interrupt vector table register (EIAD) *	0081 <sub>16</sub>	
0022 <sub>16</sub>		0082 <sub>16</sub>	
0023 <sub>16</sub>	Emulator interrupt detect register (EITD) *	0083 <sub>16</sub>	
0024 <sub>16</sub>	Emulator protect register (EPRR) *	0084 <sub>16</sub>	
0025 <sub>16</sub>		0085 <sub>16</sub>	
0026 <sub>16</sub>		0086 <sub>16</sub>	
0027 <sub>16</sub>		0087 <sub>16</sub>	
0028 <sub>16</sub>		0088 <sub>16</sub>	DMA1 interrupt control register (DM1IC)
0029 <sub>16</sub>		0089 <sub>16</sub>	UART2 transmit/NACK interrupt control register (S2TIC)
002A <sub>16</sub>		008A <sub>16</sub>	DMA3 interrupt control register (DM3IC)
002B <sub>16</sub>		008B <sub>16</sub>	UART3 transmit/NACK interrupt control register (S3TIC)
002C <sub>16</sub>		008C <sub>16</sub>	Timer A1 interrupt control register (TA1IC)
002D <sub>16</sub>		008D <sub>16</sub>	UART4 transmit/NACK interrupt control register (S4TIC)
002E <sub>16</sub>		008E <sub>16</sub>	Timer A3 interrupt control register (TA3IC)
002F <sub>16</sub>		008F <sub>16</sub>	Bus collision detection(UART2) interrupt control register (BCN2IC)
0030 <sub>16</sub>	ROM areaset register (ROA) *	0090 <sub>16</sub>	UART0 transmit interrupt control register (S0TIC)
0031 <sub>16</sub>	Debug monitor area set register (DBA) *	0091 <sub>16</sub>	Bus collision detection(UART4) interrupt control register (BCN4IC)
0032 <sub>16</sub>	Expansion area set register 0 (EXA0) *	0092 <sub>16</sub>	UART1 transmit interrupt control register (S1TIC)
0033 <sub>16</sub>	Expansion area set register 1 (EXA1) *	0093 <sub>16</sub>	Key input interrupt control register (KUPIC)
0034 <sub>16</sub>	Expansion area set register 2 (EXA2) *	0094 <sub>16</sub>	Timer B0 interrupt control register (TB0IC)
0035 <sub>16</sub>	Expansion area set register 3 (EXA3) *	0095 <sub>16</sub>	
0036 <sub>16</sub>		0096 <sub>16</sub>	Timer B2 interrupt control register (TB2IC)
0037 <sub>16</sub>		0097 <sub>16</sub>	
0038 <sub>16</sub>		0098 <sub>16</sub>	Timer B4 interrupt control register (TB4IC)
0039 <sub>16</sub>		0099 <sub>16</sub>	
003A <sub>16</sub>		009A <sub>16</sub>	INT4 interrupt control register (INT4IC)
003B <sub>16</sub>		009B <sub>16</sub>	
003C <sub>16</sub>		009C <sub>16</sub>	INT2 interrupt control register (INT2IC)
003D <sub>16</sub>		009D <sub>16</sub>	
003E <sub>16</sub>		009E <sub>16</sub>	INT0 interrupt control register (INT0IC)
003F <sub>16</sub>		009F <sub>16</sub>	Exit priority register (RLVL)
0040 <sub>16</sub>	DRAM control register (DRAMCONT)	00A0 <sub>16</sub>	
0041 <sub>16</sub>	DRAM refresh interval set register (REFCNT)	00A1 <sub>16</sub>	
0042 <sub>16</sub>		00A2 <sub>16</sub>	
0043 <sub>16</sub>		00A3 <sub>16</sub>	
0044 <sub>16</sub>		00A4 <sub>16</sub>	

\* As this register is used exclusively for debugger purposes, user cannot use this. Do not access to the register.  
(The blank area is reserved and cannot be used by user.)

Figure 5.1 Location of peripheral unit control registers (1)

02C0 <sub>16</sub>	X0 register (X0R) Y0 register (Y0R)	0300 <sub>16</sub>	Timer B3, 4, 5 count start flag (TBSR)
02C1 <sub>16</sub>		0301 <sub>16</sub>	
02C2 <sub>16</sub>	X1 register (X1R) Y1 register (Y1R)	0302 <sub>16</sub>	
02C3 <sub>16</sub>		0303 <sub>16</sub>	Timer A1-1 register (TA11)
02C4 <sub>16</sub>	X2 register (X2R) Y2 register (Y2R)	0304 <sub>16</sub>	
02C5 <sub>16</sub>		0305 <sub>16</sub>	Timer A2-1 register (TA21)
02C6 <sub>16</sub>	X3 register (X3R) Y3 register (Y3R)	0306 <sub>16</sub>	
02C7 <sub>16</sub>		0307 <sub>16</sub>	Timer A4-1 register (TA41)
02C8 <sub>16</sub>	X4 register (X4R) Y4 register (Y4R)	0308 <sub>16</sub>	Three-phase PWM control register 0(INVC0)
02C9 <sub>16</sub>		0309 <sub>16</sub>	Three-phase PWM control register 1(INVC1)
02CA <sub>16</sub>	X5 register (X5R) Y5 register (Y5R)	030A <sub>16</sub>	Three-phase output buffer register 0(IDB0)
02CB <sub>16</sub>		030B <sub>16</sub>	Three-phase output buffer register 1(IDB1)
02CC <sub>16</sub>	X6 register (X6R) Y6 register (Y6R)	030C <sub>16</sub>	Dead time timer(DTT)
02CD <sub>16</sub>		030D <sub>16</sub>	Timer B2 interrupt occurrence frequency set counter(ICTB2)
02CE <sub>16</sub>	X7 register (X7R) Y7 register (Y7R)	030E <sub>16</sub>	
02CF <sub>16</sub>		030F <sub>16</sub>	
02D0 <sub>16</sub>	X8 register (X8R) Y8 register (Y8R)	0310 <sub>16</sub>	
02D1 <sub>16</sub>		0311 <sub>16</sub>	Timer B3 register (TB3)
02D2 <sub>16</sub>	X9 register (X9R) Y9 register (Y9R)	0312 <sub>16</sub>	
02D3 <sub>16</sub>		0313 <sub>16</sub>	Timer B4 register (TB4)
02D4 <sub>16</sub>	X10 register (X10R) Y10 register (Y10R)	0314 <sub>16</sub>	
02D5 <sub>16</sub>		0315 <sub>16</sub>	Timer B5 register (TB5)
02D6 <sub>16</sub>	X11 register (X11R) Y11 register (Y11R)	0316 <sub>16</sub>	
02D7 <sub>16</sub>		0317 <sub>16</sub>	
02D8 <sub>16</sub>	X12 register (X12R) Y12 register (Y12R)	0318 <sub>16</sub>	
02D9 <sub>16</sub>		0319 <sub>16</sub>	
02DA <sub>16</sub>	X13 register (X13R) Y13 register (Y13R)	031A <sub>16</sub>	
02DB <sub>16</sub>		031B <sub>16</sub>	Timer B3 mode register (TB3MR)
02DC <sub>16</sub>	X14 register (X14R) Y14 register (Y14R)	031C <sub>16</sub>	Timer B4 mode register (TB4MR)
02DD <sub>16</sub>		031D <sub>16</sub>	Timer B5 mode register (TB5MR)
02DE <sub>16</sub>	X15 register (X15R) Y15 register (Y15R)	031E <sub>16</sub>	
02DF <sub>16</sub>		031F <sub>16</sub>	Interrupt cause select register (IFSR)
02E0 <sub>16</sub>	XY control register (XYC)	0320 <sub>16</sub>	
02E1 <sub>16</sub>		0321 <sub>16</sub>	
02E2 <sub>16</sub>		0322 <sub>16</sub>	
02E3 <sub>16</sub>		0323 <sub>16</sub>	
02E4 <sub>16</sub>		0324 <sub>16</sub>	
02E5 <sub>16</sub>		0325 <sub>16</sub>	UART3 special mode register 3 (U3SMR3)
02E6 <sub>16</sub>		0326 <sub>16</sub>	UART3 special mode register 2 (U3SMR2)
02E7 <sub>16</sub>		0327 <sub>16</sub>	UART3 special mode register (U3SMR)
02E8 <sub>16</sub>		0328 <sub>16</sub>	UART3 transmit/receive mode register (U3MR)
02E9 <sub>16</sub>		0329 <sub>16</sub>	UART3 bit rate generator (U3BRG)
02EA <sub>16</sub>		032A <sub>16</sub>	
02EB <sub>16</sub>		032B <sub>16</sub>	UART3 transmit buffer register (U3TB)
02EC <sub>16</sub>		032C <sub>16</sub>	UART3 transmit/receive control register 0 (U3C0)
02ED <sub>16</sub>		032D <sub>16</sub>	UART3 transmit/receive control register 1 (U3C1)
02EE <sub>16</sub>		032E <sub>16</sub>	UART3 receive buffer register (U3RB)
02EF <sub>16</sub>		032F <sub>16</sub>	
02F0 <sub>16</sub>		0330 <sub>16</sub>	
02F1 <sub>16</sub>		0331 <sub>16</sub>	
02F2 <sub>16</sub>		0332 <sub>16</sub>	
02F3 <sub>16</sub>		0333 <sub>16</sub>	
02F4 <sub>16</sub>		0334 <sub>16</sub>	
02F5 <sub>16</sub>	UART4 special mode register 3 (U4SMR3)	0335 <sub>16</sub>	UART2 special mode register 3 (U2SMR3)
02F6 <sub>16</sub>	UART4 special mode register 2 (U4SMR2)	0336 <sub>16</sub>	UART2 special mode register 2 (U2SMR2)
02F7 <sub>16</sub>	UART4 special mode register (U4SMR)	0337 <sub>16</sub>	UART2 special mode register (U2SMR)
02F8 <sub>16</sub>	UART4 transmit/receive mode register (U4MR)	0338 <sub>16</sub>	UART2 transmit/receive mode register (U2MR)
02F9 <sub>16</sub>	UART4 bit rate generator (U4BRG)	0339 <sub>16</sub>	UART2 bit rate generator (U2BRG)
02FA <sub>16</sub>		033A <sub>16</sub>	
02FB <sub>16</sub>	UART4 transmit buffer register (U4TB)	033B <sub>16</sub>	UART2 transmit buffer register (U2TB)
02FC <sub>16</sub>	UART4 transmit/receive control register 0 (U4C0)	033C <sub>16</sub>	UART2 transmit/receive control register 0 (U2C0)
02FD <sub>16</sub>	UART4 transmit/receive control register 1 (U4C1)	033D <sub>16</sub>	UART2 transmit/receive control register 1 (U2C1)
02FE <sub>16</sub>		033E <sub>16</sub>	
02FF <sub>16</sub>	UART4 receive buffer register (U4RB)	033F <sub>16</sub>	UART2 receive buffer register (U2RB)

(The blank area is reserved and cannot be used by user.)

Figure 5.2 Location of peripheral unit control registers (2)

0340 <sup>16</sup>	Count start flag (TABSR)	0380 <sup>16</sup>	A/D register 0 (AD0)
0341 <sup>16</sup>	Clock prescaler reset flag (CPSRF)	0381 <sup>16</sup>	
0342 <sup>16</sup>	One-shot start flag (ONSF)	0382 <sup>16</sup>	A/D register 1 (AD1)
0343 <sup>16</sup>	Trigger select register (TRGSR)	0383 <sup>16</sup>	
0344 <sup>16</sup>	Up-down flag (UDF)	0384 <sup>16</sup>	A/D register 2 (AD2)
0345 <sup>16</sup>		0385 <sup>16</sup>	
0346 <sup>16</sup>		0386 <sup>16</sup>	A/D register 3 (AD3)
0347 <sup>16</sup>	Timer A0 register (TA0)	0387 <sup>16</sup>	
0348 <sup>16</sup>		0388 <sup>16</sup>	A/D register 4 (AD4)
0349 <sup>16</sup>	Timer A1 register (TA1)	0389 <sup>16</sup>	
034A <sup>16</sup>		038A <sup>16</sup>	A/D register 5 (AD5)
034B <sup>16</sup>	Timer A2 register (TA2)	038B <sup>16</sup>	
034C <sup>16</sup>		038C <sup>16</sup>	A/D register 6 (AD6)
034D <sup>16</sup>	Timer A3 register (TA3)	038D <sup>16</sup>	
034E <sup>16</sup>		038E <sup>16</sup>	A/D register 7 (AD7)
034F <sup>16</sup>	Timer A4 register (TA4)	038F <sup>16</sup>	
0350 <sup>16</sup>		0390 <sup>16</sup>	
0351 <sup>16</sup>	Timer B0 register (TB0)	0391 <sup>16</sup>	
0352 <sup>16</sup>		0392 <sup>16</sup>	
0353 <sup>16</sup>	Timer B1 register (TB1)	0393 <sup>16</sup>	
0354 <sup>16</sup>		0394 <sup>16</sup>	A/D control register 2 (ADCON2)
0355 <sup>16</sup>	Timer B2 register (TB2)	0395 <sup>16</sup>	
0356 <sup>16</sup>	Timer A0 mode register (TA0MR)	0396 <sup>16</sup>	A/D control register 0 (ADCON0)
0357 <sup>16</sup>	Timer A1 mode register (TA1MR)	0397 <sup>16</sup>	A/D control register 1 (ADCON1)
0358 <sup>16</sup>	Timer A2 mode register (TA2MR)	0398 <sup>16</sup>	D/A register 0 (DA0)
0359 <sup>16</sup>	Timer A3 mode register (TA3MR)	0399 <sup>16</sup>	
035A <sup>16</sup>	Timer A4 mode register (TA4MR)	039A <sup>16</sup>	D/A register 1 (DA1)
035B <sup>16</sup>	Timer B0 mode register (TB0MR)	039B <sup>16</sup>	
035C <sup>16</sup>	Timer B1 mode register (TB1MR)	039C <sup>16</sup>	D/A control register (DACON)
035D <sup>16</sup>	Timer B2 mode register (TB2MR)	039D <sup>16</sup>	
035E <sup>16</sup>		039E <sup>16</sup>	
035F <sup>16</sup>		039F <sup>16</sup>	
0360 <sup>16</sup>	UART0 transmit/receive mode register (U0MR)	03A0 <sup>16</sup>	
0361 <sup>16</sup>	UART0 bit rate generator (U0BRG)	03A1 <sup>16</sup>	
0362 <sup>16</sup>		03A2 <sup>16</sup>	
0363 <sup>16</sup>	UART0 transmit buffer register (U0TB)	03A3 <sup>16</sup>	
0364 <sup>16</sup>	UART0 transmit/receive control register 0 (U0C0)	03A4 <sup>16</sup>	
0365 <sup>16</sup>	UART0 transmit/receive control register 1 (U0C1)	03A5 <sup>16</sup>	
0366 <sup>16</sup>		03A6 <sup>16</sup>	
0367 <sup>16</sup>	UART0 receive buffer register (U0RB)	03A7 <sup>16</sup>	
0368 <sup>16</sup>	UART1 transmit/receive mode register (U1MR)	03A8 <sup>16</sup>	
0369 <sup>16</sup>	UART1 bit rate generator (U1BRG)	03A9 <sup>16</sup>	
036A <sup>16</sup>		03AA <sup>16</sup>	
036B <sup>16</sup>	UART1 transmit buffer register (U1TB)	03AB <sup>16</sup>	
036C <sup>16</sup>	UART1 transmit/receive control register 0 (U1C0)	03AC <sup>16</sup>	
036D <sup>16</sup>	UART1 transmit/receive control register 1 (U1C1)	03AD <sup>16</sup>	
036E <sup>16</sup>		03AE <sup>16</sup>	
036F <sup>16</sup>	UART1 receive buffer register (U1RB)	03AF <sup>16</sup>	Function select register C(PSC)
0370 <sup>16</sup>	UART transmit/receive control register 2 (UCON)	03B0 <sup>16</sup>	Function select register A0 (PS0)
0371 <sup>16</sup>		03B1 <sup>16</sup>	Function select register A1 (PS1)
0372 <sup>16</sup>		03B2 <sup>16</sup>	Function select register B0 (PSL0)
0373 <sup>16</sup>		03B3 <sup>16</sup>	Function select register B1 (PSL1)
0374 <sup>16</sup>		03B4 <sup>16</sup>	Function select register A2 (PS2)
0375 <sup>16</sup>		03B5 <sup>16</sup>	Function select register A3 (PS3)
0376 <sup>16</sup>	Flash memory control register 1 (FMR1) (Note)	03B6 <sup>16</sup>	Function select register B2 (PSL2)
0377 <sup>16</sup>	Flash memory control register 0 (FMR0) (Note)	03B7 <sup>16</sup>	Function select register B3 (PSL3)
0378 <sup>16</sup>	DMA0 request cause select register (DM0SL)	03B8 <sup>16</sup>	
0379 <sup>16</sup>	DMA1 request cause select register (DM1SL)	03B9 <sup>16</sup>	
037A <sup>16</sup>	DMA2 request cause select register (DM2SL)	03BA <sup>16</sup>	
037B <sup>16</sup>	DMA3 request cause select register (DM3SL)	03BB <sup>16</sup>	
037C <sup>16</sup>		03BC <sup>16</sup>	
037D <sup>16</sup>	CRC data register (CRCD)	03BD <sup>16</sup>	
037E <sup>16</sup>	CRC input register (CRCIN)	03BE <sup>16</sup>	
037F <sup>16</sup>		03BF <sup>16</sup>	

Note :This register exists in the flash memory version. (The blank area is reserved and cannot be used by user.)

Figure 5.3 Location of peripheral unit control registers (3)

<100-pin version>		<144-pin version>	
03C0 <sub>16</sub>	Port P6 (P6)	03C0 <sub>16</sub>	Port P6 (P6)
03C1 <sub>16</sub>	Port P7 (P7)	03C1 <sub>16</sub>	Port P7 (P7)
03C2 <sub>16</sub>	Port P6 direction register (PD6)	03C2 <sub>16</sub>	Port P6 direction register (PD6)
03C3 <sub>16</sub>	Port P7 direction register (PD7)	03C3 <sub>16</sub>	Port P7 direction register (PD7)
03C4 <sub>16</sub>	Port P8 (P8)	03C4 <sub>16</sub>	Port P8 (P8)
03C5 <sub>16</sub>	Port P9 (P9)	03C5 <sub>16</sub>	Port P9 (P9)
03C6 <sub>16</sub>	Port P8 direction register (PD8)	03C6 <sub>16</sub>	Port P8 direction register (PD8)
03C7 <sub>16</sub>	Port P9 direction register (PD9)	03C7 <sub>16</sub>	Port P9 direction register (PD9)
03C8 <sub>16</sub>	Port P10 (P10)	03C8 <sub>16</sub>	Port P10 (P10)
03C9 <sub>16</sub>		03C9 <sub>16</sub>	Port P11 (P11)
03CA <sub>16</sub>	Port P10 direction register (PD10)	03CA <sub>16</sub>	Port P10 direction register (PD10)
03CB <sub>16</sub>		03CB <sub>16</sub>	Port P11 direction register (PD11)
03CC <sub>16</sub>		03CC <sub>16</sub>	Port P12 (P12)
03CD <sub>16</sub>		03CD <sub>16</sub>	Port P13 (P13)
03CE <sub>16</sub>		03CE <sub>16</sub>	Port P12 direction register (PD12)
03CF <sub>16</sub>		03CF <sub>16</sub>	Port P13 direction register (PD13)
03D0 <sub>16</sub>		03D0 <sub>16</sub>	Port P14 (P14)
03D1 <sub>16</sub>		03D1 <sub>16</sub>	Port P15 (P15)
03D2 <sub>16</sub>		03D2 <sub>16</sub>	Port P14 direction register (PD14)
03D3 <sub>16</sub>		03D3 <sub>16</sub>	Port P15 direction register (PD15)
03D4 <sub>16</sub>		03D4 <sub>16</sub>	
03D5 <sub>16</sub>		03D5 <sub>16</sub>	
03D6 <sub>16</sub>		03D6 <sub>16</sub>	
03D7 <sub>16</sub>		03D7 <sub>16</sub>	
03D8 <sub>16</sub>		03D8 <sub>16</sub>	
03D9 <sub>16</sub>		03D9 <sub>16</sub>	
03DA <sub>16</sub>	Pull-up control register 2 (PUR2)	03DA <sub>16</sub>	Pull-up control register 2 (PUR2)
03DB <sub>16</sub>	Pull-up control register 3 (PUR3)	03DB <sub>16</sub>	Pull-up control register 3 (PUR3)
03DC <sub>16</sub>		03DC <sub>16</sub>	Pull-up control register 4 (PUR4)
03DD <sub>16</sub>		03DD <sub>16</sub>	
03DE <sub>16</sub>		03DE <sub>16</sub>	
03DF <sub>16</sub>		03DF <sub>16</sub>	
03E0 <sub>16</sub>	Port P0 (P0)	03E0 <sub>16</sub>	Port P0 (P0)
03E1 <sub>16</sub>	Port P1 (P1)	03E1 <sub>16</sub>	Port P1 (P1)
03E2 <sub>16</sub>	Port P0 direction register (PD0)	03E2 <sub>16</sub>	Port P0 direction register (PD0)
03E3 <sub>16</sub>	Port P1 direction register (PD1)	03E3 <sub>16</sub>	Port P1 direction register (PD1)
03E4 <sub>16</sub>	Port P2 (P2)	03E4 <sub>16</sub>	Port P2 (P2)
03E5 <sub>16</sub>	Port P3 (P3)	03E5 <sub>16</sub>	Port P3 (P3)
03E6 <sub>16</sub>	Port P2 direction register (PD2)	03E6 <sub>16</sub>	Port P2 direction register (PD2)
03E7 <sub>16</sub>	Port P3 direction register (PD3)	03E7 <sub>16</sub>	Port P3 direction register (PD3)
03E8 <sub>16</sub>	Port P4 (P4)	03E8 <sub>16</sub>	Port P4 (P4)
03E9 <sub>16</sub>	Port P5 (P5)	03E9 <sub>16</sub>	Port P5 (P5)
03EA <sub>16</sub>	Port P4 direction register (PD4)	03EA <sub>16</sub>	Port P4 direction register (PD4)
03EB <sub>16</sub>	Port P5 direction register (PD5)	03EB <sub>16</sub>	Port P5 direction register (PD5)
03EC <sub>16</sub>		03EC <sub>16</sub>	
03ED <sub>16</sub>		03ED <sub>16</sub>	
03EE <sub>16</sub>		03EE <sub>16</sub>	
03EF <sub>16</sub>		03EF <sub>16</sub>	
03F0 <sub>16</sub>	Pull-up control register 0 (PUR0)	03F0 <sub>16</sub>	Pull-up control register 0 (PUR0)
03F1 <sub>16</sub>	Pull-up control register 1 (PUR1)	03F1 <sub>16</sub>	Pull-up control register 1 (PUR1)
03F2 <sub>16</sub>		03F2 <sub>16</sub>	
03F3 <sub>16</sub>		03F3 <sub>16</sub>	
03FC <sub>16</sub>		03FC <sub>16</sub>	
03FD <sub>16</sub>		03FD <sub>16</sub>	
03FE <sub>16</sub>		03FE <sub>16</sub>	
03FF <sub>16</sub>	Port control register (PCR)	03FF <sub>16</sub>	Port control register (PCR)

(The blank area is reserved and cannot be used by user.)

Note 1:  Addresses 03C9<sub>16</sub>, 03CB<sub>16</sub> to 03D3<sub>16</sub> area is for future plan.  
 Must set "FF<sub>16</sub>" to address 03CB<sub>16</sub>, 03CE<sub>16</sub>, 03CF<sub>16</sub>, 03D2<sub>16</sub>, 03D3<sub>16</sub> at initial setting.

Note 2:  Address 03DC<sub>16</sub> area is for future plan. Must set "00<sub>16</sub>" to address 03DC<sub>16</sub> at initial setting.

Figure 5.4 Location of peripheral unit control registers (4)



## Software Reset

Writing "1" to bit 3 of the processor mode register 0 (address 000416) applies a (software) reset to the microcomputer. A software reset has the same effect as a hardware reset. The contents of internal RAM are preserved.

Carry out a software reset after oscillation of main clock is fully stable.

## 6. Processor Mode

### (1) Types of Processor Mode

One of three processor modes can be selected: single-chip mode, memory expansion mode, and microprocessor mode. The functions of some pins, the memory map, and the access space differ according to the selected processor mode.

- **Single-chip mode**

In single-chip mode, only internal memory space (SFR, internal RAM, and internal ROM) can be accessed. However, after the reset has been released and the operation of shifting from the microprocessor mode has started ("H" applied to the CNVSS pin), the internal ROM area cannot be accessed even if the CPU shifts to the single-chip mode.

Ports P0 to P10 can be used as programmable I/O ports or as I/O ports for the internal peripheral functions.

- **Memory expansion mode**

In memory expansion mode, external memory can be accessed in addition to the internal memory space (SFR, internal RAM, and internal ROM). However, after the reset has been released and the operation of shifting from the microprocessor mode has started ("H" applied to the CNVSS pin), the internal ROM area cannot be accessed even if the CPU shifts to the memory expansion mode.

In this mode, some of the pins function as the address bus, the data bus, and as control signals. The number of pins assigned to these functions depends on the bus and register settings. (See "Bus Settings" for details.)

- **Microprocessor mode**

In microprocessor mode, the SFR, internal RAM, and external memory space can be accessed. The internal ROM area cannot be accessed.

In this mode, some of the pins function as the address bus, the data bus, and as control signals. The number of pins assigned to these functions depends on the bus and register settings. (See "Bus Settings" for details.)

### (2) Setting Processor Modes

The processor mode is set using the CNVSS pin and the processor mode bits (bits 1 and 0 at address 000416). Do not set the processor mode bits to "102".

Regardless of the level of the CNVSS pin, changing the processor mode bits selects the mode. Therefore, never change the processor mode bits when changing the contents of other bits. Do not change the processor mode bits simultaneously with other bits when changing the processor mode bits "012" or "112". Change the processor mode bits after changing the other bits. Also do not attempt to shift to or from the microprocessor mode within the program stored in the internal ROM area.

- **Applying V<sub>SS</sub> to CNV<sub>SS</sub> pin**

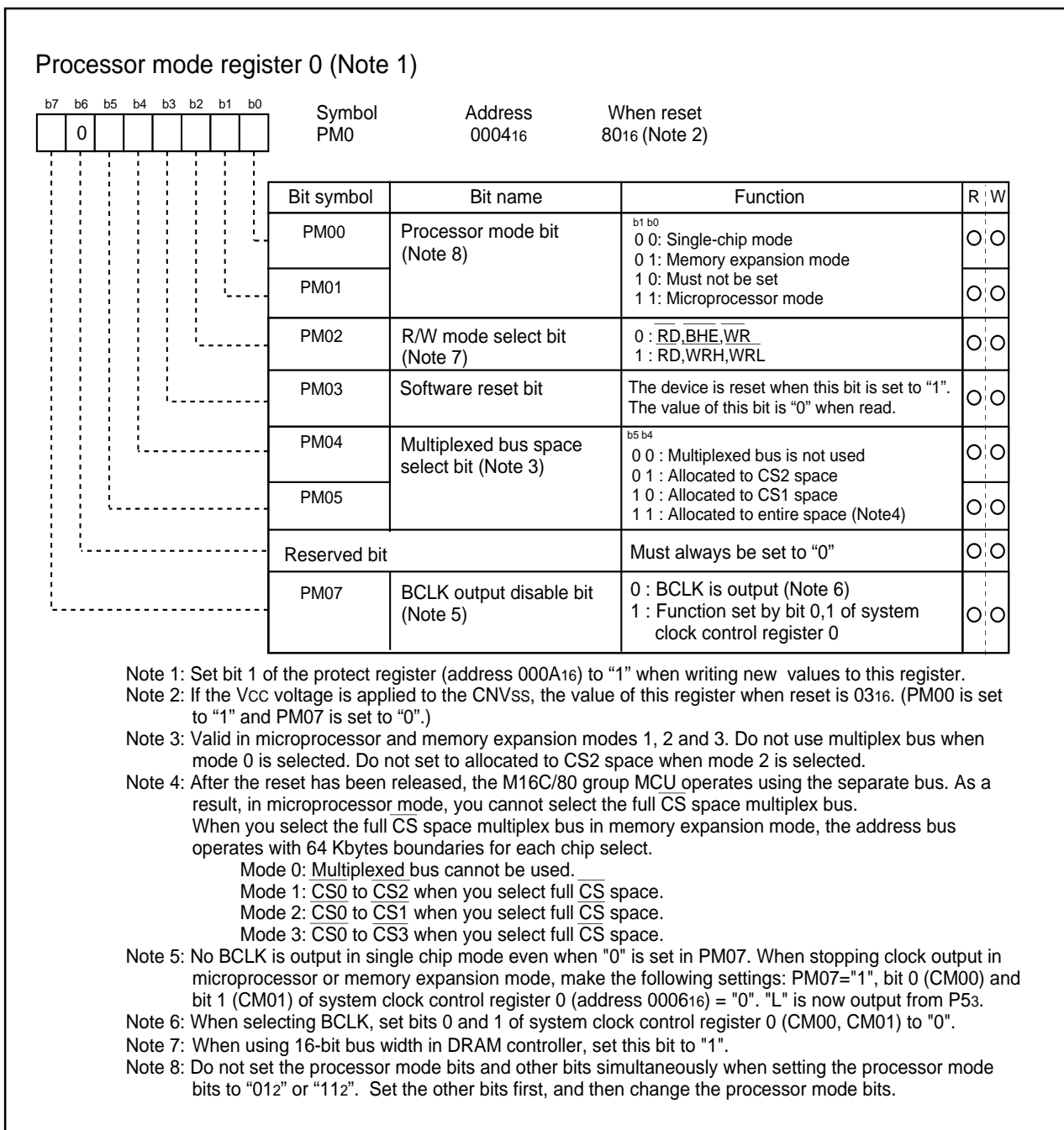
The microcomputer begins operation in single-chip mode after being reset. Memory expansion mode is selected by writing "012" to the processor mode bits.

- **Applying V<sub>CC</sub> to CNV<sub>SS</sub> pin**

The microcomputer starts to operate in microprocessor mode after being reset.

Figures 6.1 and 6.2 show the processor mode register 0 and 1.

Figure 6.3 shows the memory maps applicable for each processor modes.



**Figure 6.1 Processor mode register 0**



Processor mode register 1 (Note 1) :Mask ROM version  
ROMless version (144-pin version)

b7	b6	b5	b4	b3	b2	b1	b0	Symbol	Address	When reset
			0					PM1	0005 <sub>16</sub>	00 <sub>16</sub>

Bit symbol	Bit name	Function	R	W
PM10	External memory area mode bit (Note 3)	b1 b0 0 0 : Mode 0 (P44 to P47 : A20 to $\overline{A23}$ ) 0 1 : Mode 1 (P44 : A20, P45 to P47 : $\overline{CS2}$ to $\overline{CS0}$ )	0	0
PM11				
PM12	Internal memory wait bit	0 : No wait state 1 : Wait state inserted	—	—
Reserved bit		Must always be set to "0"	—	0
PM14	ALE pin select bit (Note 3)	b5 b4 0 0 : No ALE 0 1 : P53/BCLK (Note 4)	0	0
PM15				
		1 0 : P56/RAS 1 1 : P54/HLDA	0	0
Nothing is assigned. When read, the content is indeterminate.			—	—

Note 1: Set bit 1 of the protect register (address 000A<sub>16</sub>) to "1" when writing new values to this register.  
 Note 2: When mode 3 is selected, DRAMC is not used.  
 Note 3: Valid in memory expansion mode or in microprocessor mode.  
 Note 4: When selecting P53/BCLK, set bits 0 and 1 of system clock control register 0 (CM00, CM01) to "0".

Processor mode register 1 (Note 1) :Flash memory version

b7	b6	b5	b4	b3	b2	b1	b0	Symbol	Address	When reset
			0					PM1	0005 <sub>16</sub>	00 <sub>16</sub>

Bit symbol	Bit name	Function	R	W
PM10	External memory area mode bit (Note 3)	b1 b0 0 0 : Mode 0 (P44 to P47 : A20 to $\overline{A23}$ ) 0 1 : Mode 1 (P44 : A20, P45 to P47 : $\overline{CS2}$ to $\overline{CS0}$ )	0	0
PM11				
PM12	Internal memory wait bit	0 : No wait state 1 : Wait state inserted	—	—
Reserved bit		Must always be set to "0"	—	0
PM14	ALE pin select bit (Note 3)	b5 b4 0 0 : No ALE 0 1 : P53/BCLK (Note 4)	0	0
PM15				
		1 0 : P56/RAS 1 1 : P54/HLDA	0	0
Reserved bit		Must always be set to "1" (Note 5)	0	0

Note 1: Set bit 1 of the protect register (address 000A<sub>16</sub>) to "1" when writing new values to this register.  
 Note 2: When mode 3 is selected, DRAMC is not used.  
 Note 3: Valid in memory expansion mode or in microprocessor mode.  
 Note 4: When selecting P53/BCLK, set bits 0 and 1 of system clock control register 0 (CM00, CM01) to "0".  
 Note 5: Rewrite this bit when the main clock is in division by 8 mode.

Figure 6.2 Processor mode register 1

	Single chip mode			Memory expanded mode			Microprocessor mode		
	Mode 0	Mode 1	Mode 2	Mode 3	Mode 0	Mode 1	Mode 2	Mode 3	
00000016	SFR area	SFR area	SFR area	SFR area	SFR area	SFR area	SFR area	SFR area	
00040016	Internal RAM area	Internal RAM area	Internal RAM area	Internal RAM area	Internal RAM area	Internal RAM area	Internal RAM area	Internal RAM area	
00080016	Internal reserved area	Internal reserved area	Internal reserved area	Internal reserved area	Internal reserved area	Internal reserved area	Internal reserved area	Internal reserved area	
20000016	External area 0	$\overline{CS1}$ 2Mbytes (Note1) External area 0	$\overline{CS1}$ 4Mbytes (Note2) External area 0	No use $\overline{CS1}$ , 1Mbytes External area 0 $\overline{CS2}$ , 1Mbytes External area 1	External area 0	$\overline{CS1}$ 2Mbytes (Note1) External area 0 $\overline{CS2}$ 2Mbytes External area 1	$\overline{CS1}$ 4Mbytes (Note2) External area 0	No use $\overline{CS1}$ , 1Mbytes External area 0 $\overline{CS2}$ , 1Mbytes External area 1 No use	
40000016	External area 1	$\overline{CS2}$ 2Mbytes External area 1	External area 1	No use	External area 1	External area 1	External area 1	No use	
	Connect with DRAM 0, 0.5 to 8MB (When not connect with DRAM, use as external area.) (External area 2)	Connect with DRAM 0, 0.5 to 8MB (When open area is under 8MB, cannot use the rest of this area.) (External area 2)	Connect with DRAM 0, 0.5 to 8MB (When open area is under 8MB, cannot use the rest of this area.) (External area 2)	No use (Cannot use as DRAM area or external area.)	Connect with DRAM 0, 0.5 to 8MB (When not connect with DRAM, use as external area.) (External area 2)	Connect with DRAM 0, 0.5 to 8MB (When open area is under 8MB, cannot use the rest of this area.) (External area 2)	Connect with DRAM 0, 0.5 to 8MB (When open area is under 8MB, cannot use the rest of this area.) (External area 2)	No use (Cannot use as DRAM area or external area.)	
C0000016	External area 3	$\overline{CS0}$ 2Mbytes External area 3	$\overline{CS0}$ 3Mbytes External area 3	$\overline{CS3}$ , 1Mbytes External area 2	External area 3	No use	$\overline{CS0}$ 4Mbytes External area 3	$\overline{CS3}$ , 1Mbytes External area 2	
E0000016	Internal reserved area	No use	Internal reserved area	No use	External area 3	External area 3	External area 3	No use	
F0000016	Internal ROM area	Internal reserved area	Internal reserved area	$\overline{CS0}$ , 1Mbytes External area 3	Internal reserved area	Internal reserved area	Internal reserved area	$\overline{CS0}$ , 1Mbytes External area 3	
FFFFFF16	Internal ROM area	Internal ROM area	Internal ROM area	Internal ROM area	Internal ROM area	Internal ROM area	Internal ROM area	$\overline{CS0}$ , 1Mbytes External area 3	

Note 1: 20000016~00800016=2016 Kbytes. 32 K less than 2 MB.  
 Note 2: 40000016~00800016=4064 Kbytes. 32 K less than 4 MB.

Each  $\overline{CS0}$  to  $\overline{CS3}$  can set 0 to 3 WAIT.

Figure 6.3 Memory maps in each processor mode

## 7. Bus

### 7.1 Bus Settings

The BYTE pin, bit 0 to 3 of the external data bus width control register (address 000B<sub>16</sub>), bits 4 and 5 of the processor mode register 0 (address 0004<sub>16</sub>) and bit 0 and 1 of the processor mode register 1 (address 0005<sub>16</sub>) are used to change the bus settings.

Table 7.1 shows the factors used to change the bus settings, Figure 7.1 shows external data bus width control register and Table 7.2 shows external area 0 to 3 and external area mode.

**Table 7.1 Factors for switching bus settings**

Bus setting	Switching factor
Switching external address bus width	External data bus width control register
Switching external data bus width	BYTE pin (external area 3 only)
Switching between separate and multiplex bus	Bits 4 and 5 of processor mode register 0

#### (1) Selecting external address bus width

You can select the width of the address bus output externally from the 16 Mbytes address space, the number of chip select signals, and the address area of the chip select signals. (Note, however, that when you select “Full  $\overline{CS}$  space multiplex bus”, addresses A<sub>0</sub> to A<sub>15</sub> are output.) The combination of bits 0 and 1 of the processor mode register 1 allow you to set the external area mode.

When using DRAM controller, the DRAM area is output by multiplexing of the time splitting of the row and column addresses.

#### (2) Selecting external data bus width

You can select 8-bit or 16-bit for the width of the external data bus for external areas 0, 1, 2, and 3. When the data bus width bit of the external data bus width control register is “0”, the data bus width is 8 bits; when “1”, it is 16 bits. The width can be set for each of the external areas. The default bus width for external area 3 is 16 bits when the BYTE pin is “L” after a reset, or 8 bits when the BYTE pin is “H” after a reset. The bus width selection is valid only for the external bus (the internal bus width is always 16 bits). During operation, fix the level of the BYTE pin to “H” or “L”.

#### (3) Selecting separate/multiplex bus

The bus format can be set to multiplex or separate bus using bits 4 and 5 of the processor mode register 0.

##### • Separate bus

In this bus configuration, input and output is performed on separate data and address buses. The data bus width can be set to 8 bits or 16 bits using the external data bus width control register. For all programmable external areas, P0 is the data bus when the external data bus is set to 8 bits, and P1 is a programmable IO port. When the external data bus width is set to 16 bits for any of the external areas, P0 and P1 (although P1 is undefined for any 8-bit bus areas) are the data bus.

When accessing memory using the separate bus configuration, you can select a software wait using the wait control register.

##### • Multiplex bus

In this bus configuration, data and addresses are input and output on a time-sharing basis. For areas for which 8-bit has been selected using the external data bus width control register, the 8 bits D<sub>0</sub> to D<sub>7</sub> are multiplexed with the 8 bits A<sub>0</sub> to A<sub>7</sub>. For areas for which 16-bit has been selected using the external data bus width control register, the 16 bits D<sub>0</sub> to D<sub>15</sub> are multiplexed with the 16 bits A<sub>0</sub> to A<sub>15</sub>. When accessing memory using the multiplex bus configuration, two waits are inserted regardless of whether you select “No wait” or “1 wait” in the appropriate bit of the wait control register.

The default after a reset is the separate bus configuration, and the full  $\overline{CS}$  space multiplex bus configuration cannot be selected in microprocessor mode. If you select “Full  $\overline{CS}$  space multiplex bus”, the 16 bits from A0 to A15 are output for the address.

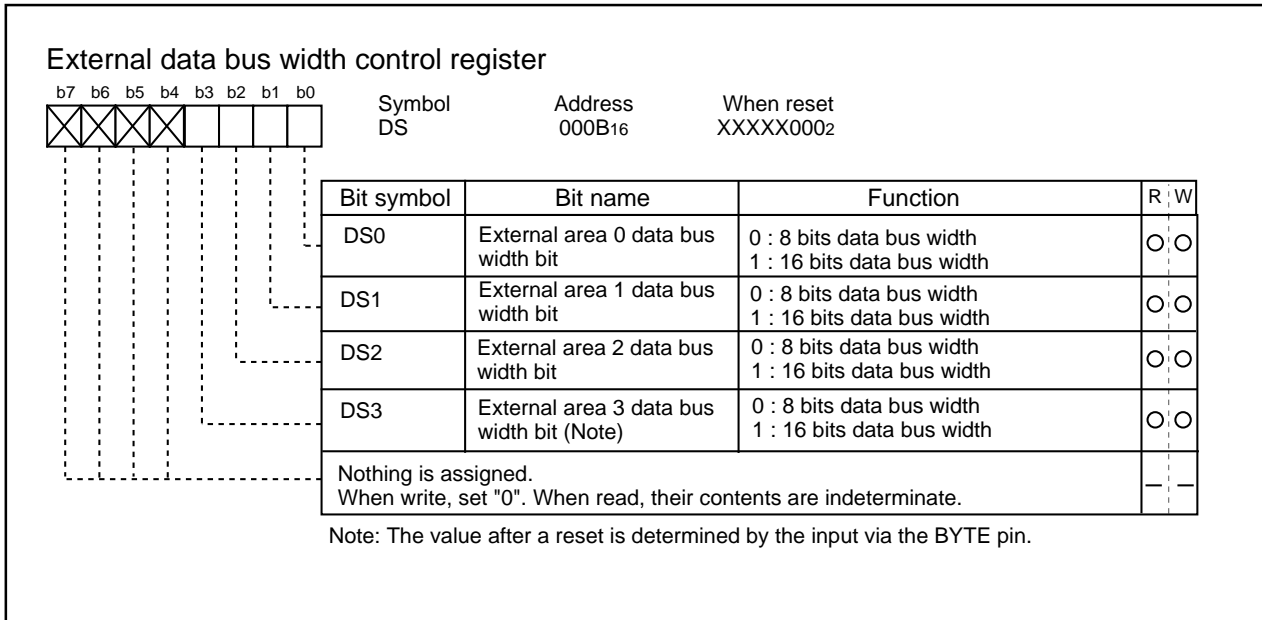


Figure 7.1 External data bus width control register

Table 7.2 External area 0 to 3 and external area mode

		External area mode (Note 2)	Mode 0	Mode 1	Mode 2	Mode 3
External area 0	Memory expansion mode, Microprocessor mode		008000 <sub>16</sub> to 1FFFFFF <sub>16</sub>	<CS1 area> 008000 <sub>16</sub> to 1FFFFFF <sub>16</sub>	<CS1 area> 008000 <sub>16</sub> to 1FFFFFF <sub>16</sub>	<CS1 area> 100000 <sub>16</sub> to 1FFFFFF <sub>16</sub>
External area 1	Memory expansion mode, Microprocessor mode		200000 <sub>16</sub> to 3FFFFFF <sub>16</sub>	<CS2 area> 200000 <sub>16</sub> to 3FFFFFF <sub>16</sub>	No area is selected.	<CS2 area> 200000 <sub>16</sub> to 2FFFFFF <sub>16</sub>
External area 2	Memory expansion mode, Microprocessor mode		400000 <sub>16</sub> to BFFFFFF <sub>16</sub> (Note 1)	<DRAMC area> 400000 <sub>16</sub> to BFFFFFF <sub>16</sub>	<DRAMC area> 400000 <sub>16</sub> to BFFFFFF <sub>16</sub>	<CS3 area> C00000 <sub>16</sub> to CFFFFFF <sub>16</sub>
External area 3	Memory expansion mode		C00000 <sub>16</sub> to EFFFFFF <sub>16</sub>	<CS0 area> C00000 <sub>16</sub> to EFFFFFF <sub>16</sub>	<CS0 area> C00000 <sub>16</sub> to EFFFFFF <sub>16</sub>	<CS0 area> E00000 <sub>16</sub> to EFFFFFF <sub>16</sub>
	Microprocessor mode		C00000 <sub>16</sub> to FFFFFF <sub>16</sub>	<CS0 area> E00000 <sub>16</sub> to FFFFFF <sub>16</sub>	<CS0 area> C00000 <sub>16</sub> to FFFFFF <sub>16</sub>	<CS0 area> F00000 <sub>16</sub> to FFFFFF <sub>16</sub>

Note 1: DRAMC area when using DRAMC.

Note 2: Set the external area mode (modes 0, 1, 2, and 3) using bits 0 and 1 of the processor mode register 1 (address 0005<sub>16</sub>).

Table 7.3 Each processor mode and port function

Processor mode	Single-chip mode	Memory expansion mode/microprocessor modes				Memory expansion mode	
Multiplexed bus space select bit		"01", "10" CS1 or CS2 : multiplexed bus, and the other : separate bus		"00" Separate bus		"11" (Note 1) All space multiplexed bus	
Data bus width BYTE pin level		All external area is 8 bits	Some external area is 16 bits	All external area is 8 bits	Some external area is 16 bits	All external area is 8 bits	Some external area is 16 bits
P00 to P07	I/O port	Data bus	Data bus	Data bus	Data bus	I/O port	I/O port
P10 to P17	I/O port	I/O port	Data bus	I/O port	Data bus	I/O port	I/O port
P20 to P27	I/O port	Address bus /data bus (Note 2)	Address bus /data bus (Note 2)	Address bus	Address bus	Address bus /data bus	Address bus /data bus
P30 to P37	I/O port	Address bus	Address bus /data bus (Note 2)	Address bus	Address bus	Address bus	Address bus /data bus
P40 to P43	I/O port	Address bus	Address bus	Address bus	Address bus	I/O port	I/O port
P44 to P46	I/O port	$\overline{CS}$ (chip select) or address bus (A20 to A22) (For details, refer to "Bus control") (Note 5)					
P47	I/O port	$\overline{CS}$ (chip select) or address bus ( $\overline{A23}$ ) (For details, refer to "Bus control") (Note 5)					
P50 to P53	I/O port	Outputs $\overline{RD}$ , $\overline{WRL}$ , $\overline{WRH}$ and BCLK, or $\overline{RD}$ , $\overline{BHE}$ , $\overline{WR}$ and BCLK (For details, refer to "Bus control") (Note 3,4)					
P54	I/O port	$\overline{HLDA}$ (Note 3)	$\overline{HLDA}$ (Note 3)	$\overline{HLDA}$ (Note 3)	$\overline{HLDA}$ (Note 3)	$\overline{HLDA}$ (Note 3)	$\overline{HLDA}$ (Note 3)
P55	I/O port	$\overline{HOLD}$	$\overline{HOLD}$	$\overline{HOLD}$	$\overline{HOLD}$	$\overline{HOLD}$	$\overline{HOLD}$
P56	I/O port	RAS (Note 3)	RAS (Note 3)	RAS (Note 3)	RAS (Note 3)	RAS (Note 3)	RAS (Note 3)
P57	I/O port	$\overline{RDY}$	$\overline{RDY}$	$\overline{RDY}$	$\overline{RDY}$	$\overline{RDY}$	$\overline{RDY}$

Note 1: The default after a reset is the separate bus configuration, and "Full  $\overline{CS}$  space multiplex bus" cannot be selected in microprocessor mode. When you select "Full  $\overline{CS}$  space multiplex bus" in extended memory mode, the address bus operates with 64 Kbytes boundaries for each chip select.

Note 2: Address bus in separate bus configuration.

Note 3: The ALE output pin is selected using bits 4 and 5 of the processor mode register 1.

Note 4: When you have selected use of the DRAM controller and you access the DRAM area, these are  $\overline{CASL}$ ,  $\overline{CASH}$ ,  $\overline{DW}$ , and BCLK outputs.

Note 5: The  $\overline{CS}$  signal and address bus selection are set by the external area mode.

## 7.2 Bus Control

The following explains the signals required for accessing external devices and software waits. The signals required for accessing the external devices are valid when the processor mode is set to memory expansion mode and microprocessor mode.

### (1) Address bus/data bus

There are 24 pins, A0 to A22 and  $\overline{A23}$  for the address bus for accessing the 16 Mbytes address space.  $\overline{A23}$  is an inverted output of the MSB of the address.

The data bus consists of pins for data IO. The external data bus control register (address 000B<sub>16</sub>) selects the 8-bit data bus, D0 to D7 for each external area, or the 16-bit data bus, D0 to D15. After a reset, there is by default an 8-bit data bus for the external area 3 when the BYTE pin is "H", or a 16-bit data bus when the BYTE pin is "L".

When shifting from single-chip mode to extended memory mode, the value on the address bus is undefined until an external area is accessed.

When accessing a DRAM area with DRAM control in use, a multiplexed signal consisting of row address and column address is output to A8 to A20.

### (2) Chip select signals

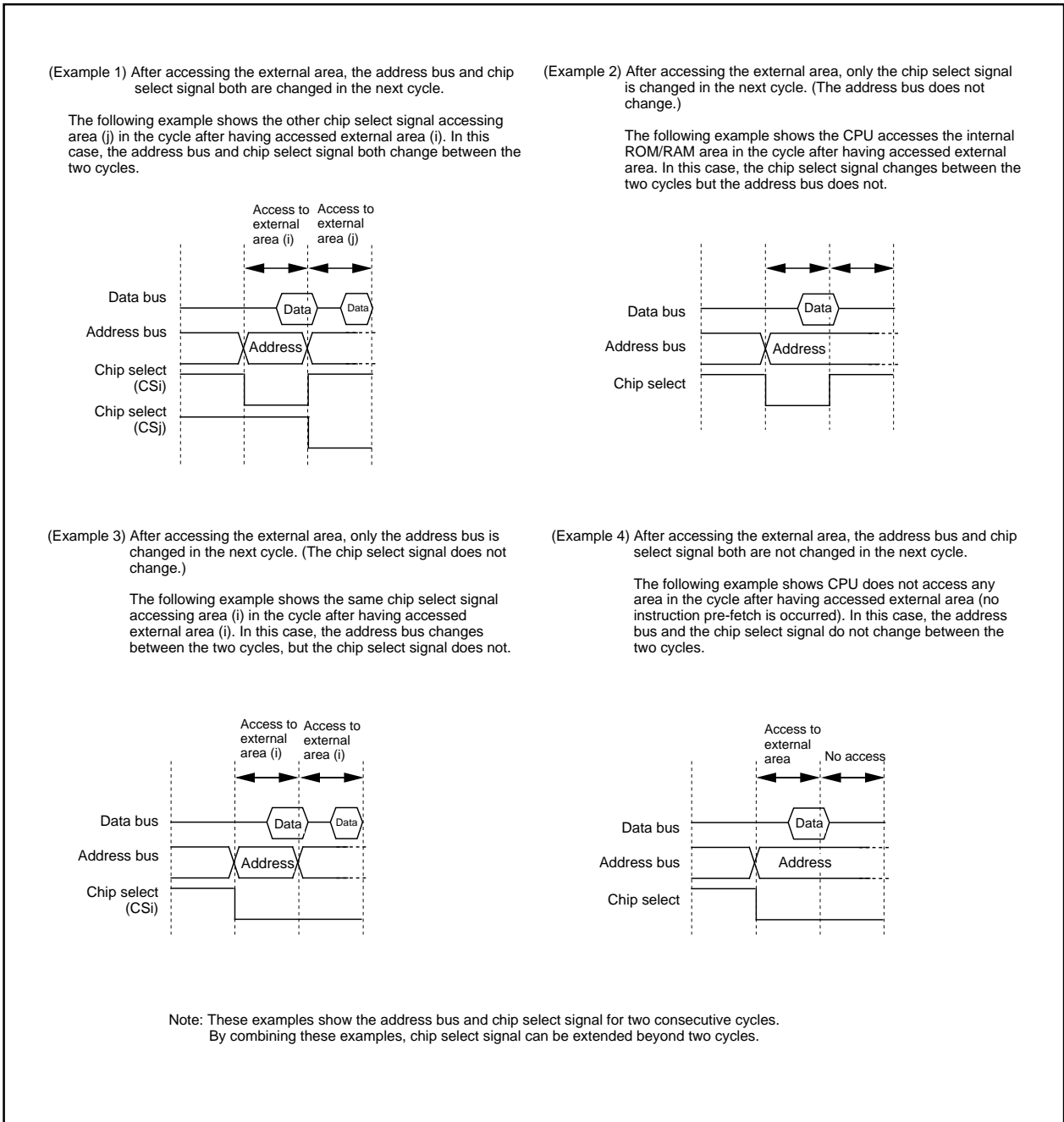
The chip select signals share A0 to A22 and  $\overline{A23}$ . You can use bits 0 and 1 of the processor mode register 1 (address 0005<sub>16</sub>) to set the external area mode, then select the chip select area and number of address outputs.

In microprocessor mode, external area mode 0 is selected after a reset. The external area can be split into a maximum of four using the chip select signals. Table 7.4 shows the external areas specified by the chip select signals.

**Table 7.4 External areas specified by the chip select signals**

Memory space expansion mode	Processor mode	Chip select signal				
		$\overline{CS0}$	$\overline{CS1}$	$\overline{CS2}$	$\overline{CS3}$	
Specified address range	Mode 0	$\overline{(A23)}$	$\overline{(A22)}$	$\overline{(A21)}$	$\overline{(A20)}$	
	Mode 1	Memory expansion mode	C00000 <sub>16</sub> to DFFFFFF <sub>16</sub> (2 Mbytes)	008000 <sub>16</sub> to 1FFFFFF <sub>16</sub> (2016 Kbytes)	200000 <sub>16</sub> to 3FFFFFF <sub>16</sub> (2 Mbytes)	$\overline{(A20)}$
		Microprocessor mode	E00000 <sub>16</sub> to FFFFFFF <sub>16</sub> (2 Mbytes)			
	Mode 2	Memory expansion mode	C00000 <sub>16</sub> to EFFFFFF <sub>16</sub> (3 Mbytes)	008000 <sub>16</sub> to 3FFFFFF <sub>16</sub> (4064 Kbytes)	$\overline{(A21)}$	$\overline{(A20)}$
		Microprocessor mode	C00000 <sub>16</sub> to FFFFFFF <sub>16</sub> (4 Mbytes)			
	Mode 3	Memory expansion mode	E00000 <sub>16</sub> to EFFFFFF <sub>16</sub> (1 Mbytes)	100000 <sub>16</sub> to 1FFFFFF <sub>16</sub> (1 Mbytes)	200000 <sub>16</sub> to 2FFFFFF <sub>16</sub> (1 Mbytes)	C00000 <sub>16</sub> to CFFFFFF <sub>16</sub> (1 Mbytes)
		Microprocessor mode	F00000 <sub>16</sub> to FFFFFFF <sub>16</sub> (1 Mbytes)			

The chip select signal turns “L” (active) in synchronize with the address bus. However, its turning “H” depends on the area accessed in the next cycle. Figure 7.2 shows the output examples of the address bus and chip select signals.



**Figure 7.2 Example of address bus and chip select signal outputs (Separate bus)**

### (3) Read/write signals

With a 16-bit data bus, bit 2 of the processor mode register 0 (address 0004<sub>16</sub>) select the combinations of  $\overline{RD}$ ,  $\overline{BHE}$ , and  $\overline{WR}$  signals or  $\overline{RD}$ ,  $\overline{WRL}$ , and  $\overline{WRH}$  signals. With a 8-bit full space data bus, use the combination of  $\overline{RD}$ ,  $\overline{WR}$ , and  $\overline{BHE}$  signals as read/write signals. (Set "0" to bit 2 of the processor mode register 0 (address 0004<sub>16</sub>.) When using both 8-bit and 16-bit data bus widths and you access an 8-bit data bus area, the  $\overline{RD}$ ,  $\overline{WR}$  and  $\overline{BHE}$  signals combination is selected regardless of the value of bit 2 of the processor mode register 0 (address 0004<sub>16</sub>).

Tables 7.5 and 7.6 show the operation of these signals.

After a reset has been cancelled, the combination of  $\overline{RD}$ ,  $\overline{WR}$ , and  $\overline{BHE}$  signals is automatically selected. When switching to the  $\overline{RD}$ ,  $\overline{WRL}$ , and  $\overline{WRH}$  combination, do not write to external memory until bit 2 of the processor mode register 0 (address 0004<sub>16</sub>) has been set (Note).

Note 1: Before attempting to change the contents of the processor mode register 0, set bit 1 of the protect register (address 000A<sub>16</sub>) to "1".

Note 2: When using 16-bit data bus width for DRAM controller, select  $\overline{RD}$ ,  $\overline{WRL}$ , and  $\overline{WRH}$  signals.

**Table 7.5 Operation of  $\overline{RD}$ ,  $\overline{WRL}$ , and  $\overline{WRH}$  signals**

Data bus width	$\overline{RD}$	$\overline{WRL}$	$\overline{WRH}$	Status of external data bus
16-bit	L	H	H	Read data
	H	L	H	Write 1 byte of data to even address
	H	H	L	Write 1 byte of data to odd address
	H	L	L	Write data to both even and odd addresses
8-bit	H	L (Note)	Not used	Write 1 byte of data
	L	H (Note)	Not used	Read 1 byte of data

Note: It becomes  $\overline{WR}$  signal.

**Table 7.6 Operation of  $\overline{RD}$ ,  $\overline{WR}$ , and  $\overline{BHE}$  signals**

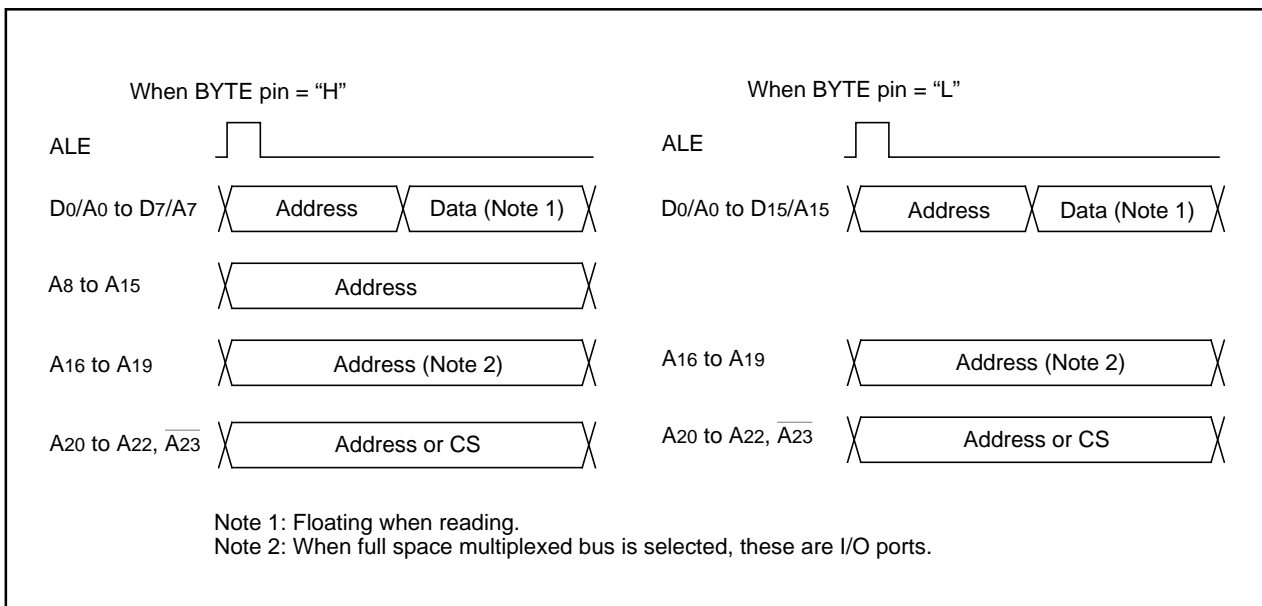
Data bus width	$\overline{RD}$	$\overline{WR}$	$\overline{BHE}$	A0	Status of external data bus
16-bit	H	L	L	H	Write 1 byte of data to odd address
	L	H	L	H	Read 1 byte of data from odd address
	H	L	H	L	Write 1 byte of data to even address
	L	H	H	L	Read 1 byte of data from even address
	H	L	L	L	Write data to both even and odd addresses
	L	H	L	L	Read data from both even and odd addresses
8-bit	H	L	Not used	H / L	Write 1 byte of data
	L	H	Not used	H / L	Read 1 byte of data



### (4) ALE signal

The ALE signal latches the address when accessing the multiplexed bus space. Latch the address when the ALE signal falls. The ALE output pin is selected using bits 4 and 5 of the processor mode register 1 (address 0005<sub>16</sub>).

The ALE signal is occurred regardless of internal area and external area.



**Figure 7.3 ALE signal and address/data bus**

### (5) Ready signal

The ready signal facilitates access of external devices that require a long time for access. As shown in Figure 7.2, inputting "L" to the  $\overline{RDY}$  pin at the falling edge of BCLK causes the microcomputer to enter the ready state. Inputting "H" to the  $\overline{RDY}$  pin at the falling edge of BCLK cancels the ready state. Table 7.7 shows the microcomputer status in the ready state. Figure 7.4 shows the example of the  $\overline{RD}$  signal being extended using the  $\overline{RDY}$  signal.

Ready is valid when accessing the external area during the bus cycle in which the software wait is applied. When no software wait is operating, the  $\overline{RDY}$  signal is ignored, but even in this case, unused pins must be pulled up.

**Table 7.7 Microcomputer status in ready state (Note)**

Item	Status
Oscillation	On
$\overline{RD}/\overline{WR}$ signal, address bus, data bus, $\overline{CS}$ ALE signal, $\overline{HLDA}$ , programmable I/O ports	Maintain status when ready signal received
Internal peripheral circuits	On

Note: The ready signal cannot be received immediately prior to a software wait.

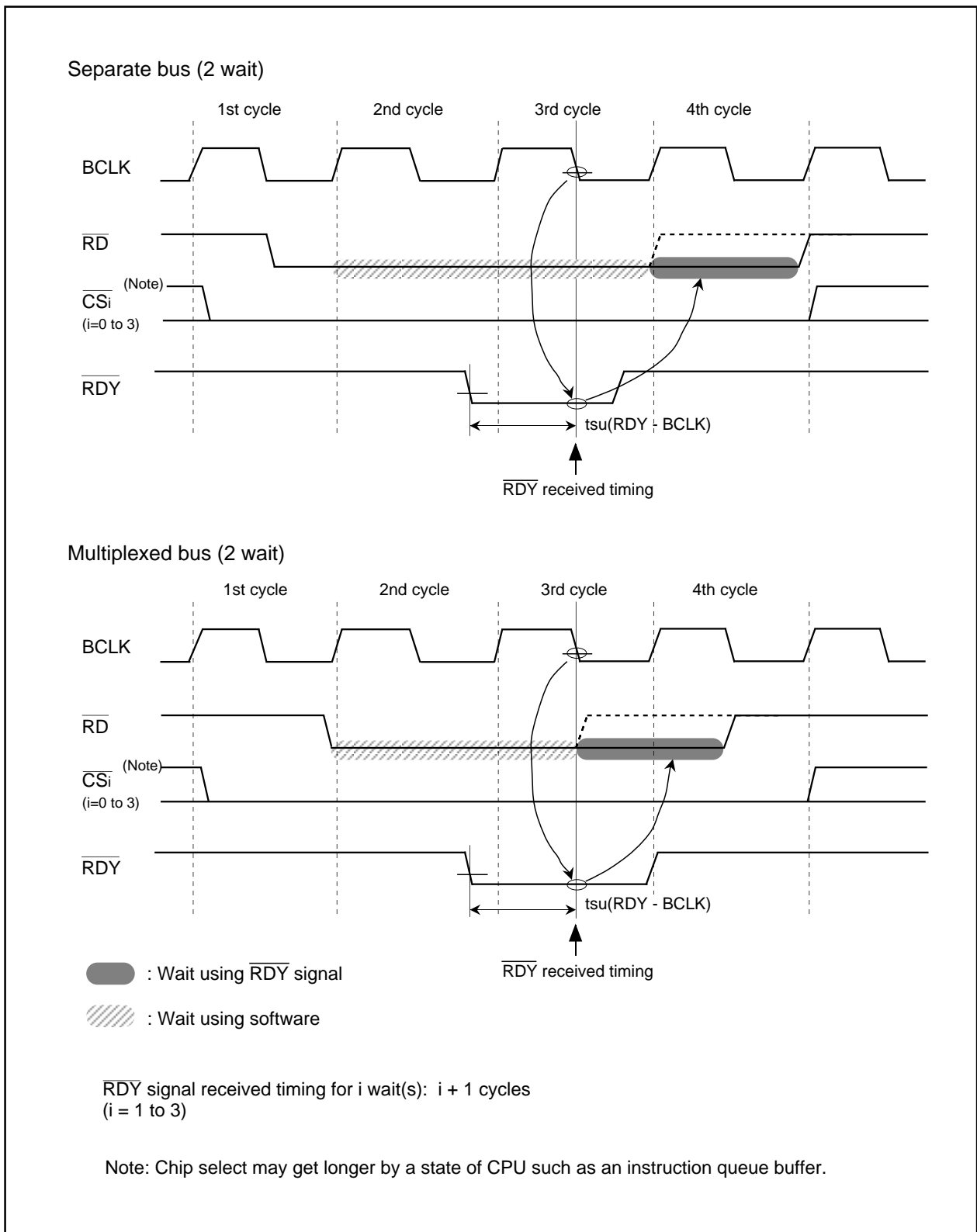


Figure 7.4 Example of RD signal extended by RDY signal

## (6) Hold signal

The hold signal is used to transfer the bus privileges from the CPU to the external circuits. Inputting “L” to the  $\overline{\text{HOLD}}$  pin places the microcomputer in the hold state at the end of the current bus access. This status is maintained and “L” is output from the  $\overline{\text{HLDA}}$  pin as long as “L” is input to the  $\overline{\text{HOLD}}$  pin. Table 7.8 shows the microcomputer status in the hold state. The bus is used in the following descending order of priority:  $\overline{\text{HOLD}}$ , DMAC, CPU.

$\overline{\text{HOLD}} > \text{DMAC} > \text{CPU}$

Figure 7.5 Example of  $\overline{\text{RD}}$  signal extended by  $\overline{\text{RDY}}$  signal

Table 7.8 Microcomputer status in hold state

Item		Status
Oscillation		ON
$\overline{\text{RD}}/\overline{\text{WR}}$ signal, address bus, data bus, $\overline{\text{CS}}$ , BHE		Floating
Programmable I/O ports	P0, P1, P2, P3, P4, P5 P6, P7, P8, P9, P10 P11, P12, P13, P14, P15 (Note)	Maintains status when hold signal is received
$\overline{\text{HLDA}}$		Output “L”
Internal peripheral circuits		ON (but watchdog timer stops)
ALE signal		Undefined

Note: Ports P11 to P15 exist in 144-pin version.

## (7) External bus status when accessing to internal area

Table 7.9 shows external bus status when accessing to internal area

Table 7.9 External bus status when accessing to internal area

Item		SFR accessing status	Internal ROM/RAM accessing status
Address bus		Remain address of external area accessed immediately before	
Data bus	When read	Floating	
	When write	Floating	
$\overline{\text{RD}}$ , $\overline{\text{WR}}$ , $\overline{\text{WRL}}$ , $\overline{\text{WRH}}$		Output "H"	
BHE		Remain external area status accessed immediately before	
$\overline{\text{CS}}$		Output "H"	
ALE		ALE output	

## (8) BCLK output

BCLK output can be selected by bit 7 of the processor mode register 0 (address 0004<sub>16</sub>:PM07) and bit 1 and bit 0 of the system clock select register 0 (address 0006<sub>16</sub>:CM01, CM00). Setting PM07 to “0” and CM01 and CM00 to “00” outputs the BCLK signal from P53. However, in single chip mode, BCLK signal is not output. When setting PM07 to “1”, the function is as set by CM01 and CM00.

### (9) DRAM controller signals (RAS, CASL, CASH, and DW)

Bits 1, 2, and 3 of the DRAM control register (address 000416) select the DRAM space and enable the DRAM controller. The DRAM controller signals are then output when the DRAM area is accessed. Table 7.10 shows the operation of the respective signals.

**Table 7.10 Operation of RAS, CASL, CASH, and DW signals**

Data bus width	RAS	CASL	CASH	DW	Status of external data bus
16-bit	L	L	L	H	Read data from both even and odd addresses
	L	L	L	H	Read 1 byte of data from even address
	L	H	H	H	Read 1 byte of data from odd address
	L	L	L	L	Write data to both even and odd addresses
	L	L	H	L	Write 1 byte of data to even address
	L	H	L	L	Write 1 byte of data to odd address
8-bit	L	L	Not used	H	Read 1 byte of data
	L	L	Not used	L	Write 1 byte of data

### (10) Software wait

A software wait can be inserted by setting the wait control register (address 000816). Figure 7.6 shows wait control register

You can use the external area I wait bits (where I = 0 to 3) of the wait control register to specify from “No wait” to “3 waits” for the external memory area. When you select “No wait”, the read cycle is executed in the BCLK1 cycle. The write cycle is executed in the BCLK2 cycle (which has 1 wait). When accessing external memory using the multiplex bus, access has two waits regardless of whether you specify “No wait” or “1 wait” in the appropriate external area i wait bits in the wait control register.

Software waits in the internal memory (internal RAM and internal ROM) can be set using the internal memory wait bits of the processor mode register 1 (address 000516). Setting the internal memory wait bit = “0” sets “No wait”. Setting the internal memory wait bit = “1” specifies a wait.

The SFR area is not affected by the setting of the internal memory wait bit and is always accessed in the BCLK2 cycle.

Table 7.11 shows the software waits and bus cycles. Figures 7.7 and 7.8 show example bus timings when using software waits.

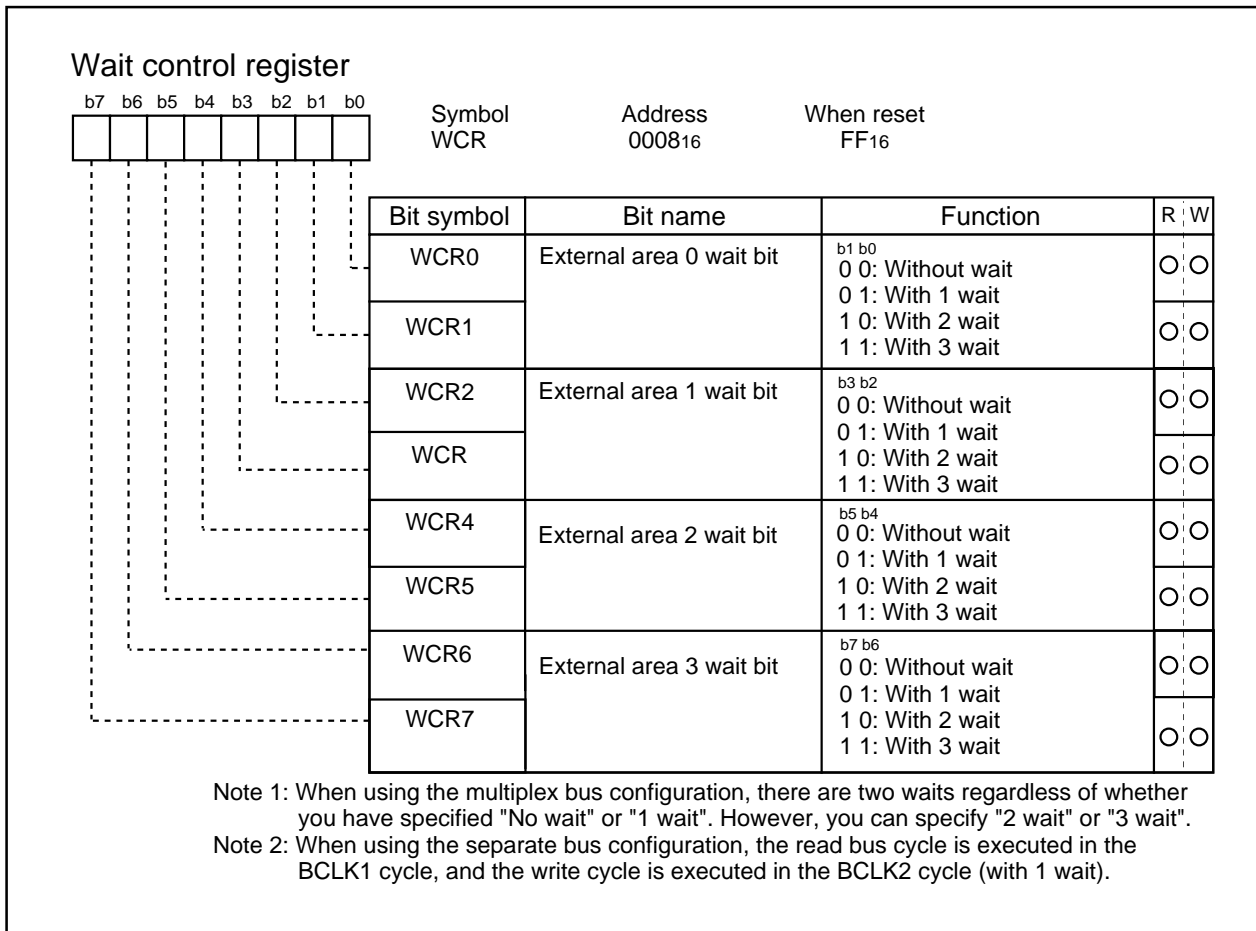


Figure 7.6 Wait control register

Table 7.11 Software waits and bus cycles

Area	Bus status	Internal memory wait bit	External memory area i wait bit	Bus cycle
SFR	_____	_____	_____	2 BCLK cycles
Internal ROM/RAM	_____	0	_____	1 BCLK cycle
	_____	1	_____	2 BCLK cycles
External memory area	Separate bus	_____	002	Read : 1 BCLK cycle Write : 2 BCLK cycles
			012	2 BCLK cycles
			102	3 BCLK cycles
			112	4 BCLK cycles
	Multiplex bus	_____	002	3 BCLK cycle
			012	3 BCLK cycles
			102	3 BCLK cycles
			112	4 BCLK cycles

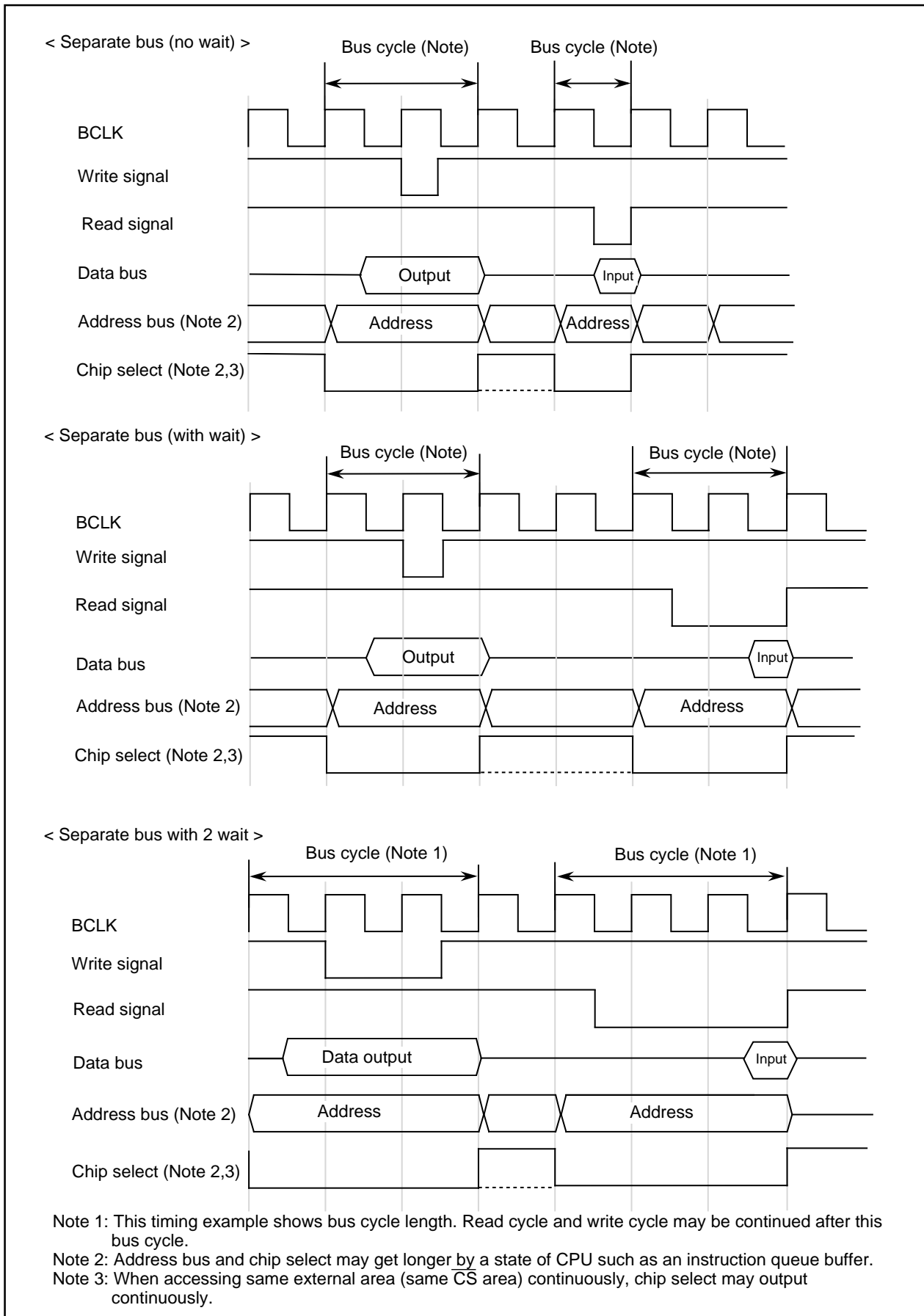


Figure 7.7 Typical bus timings using software wait

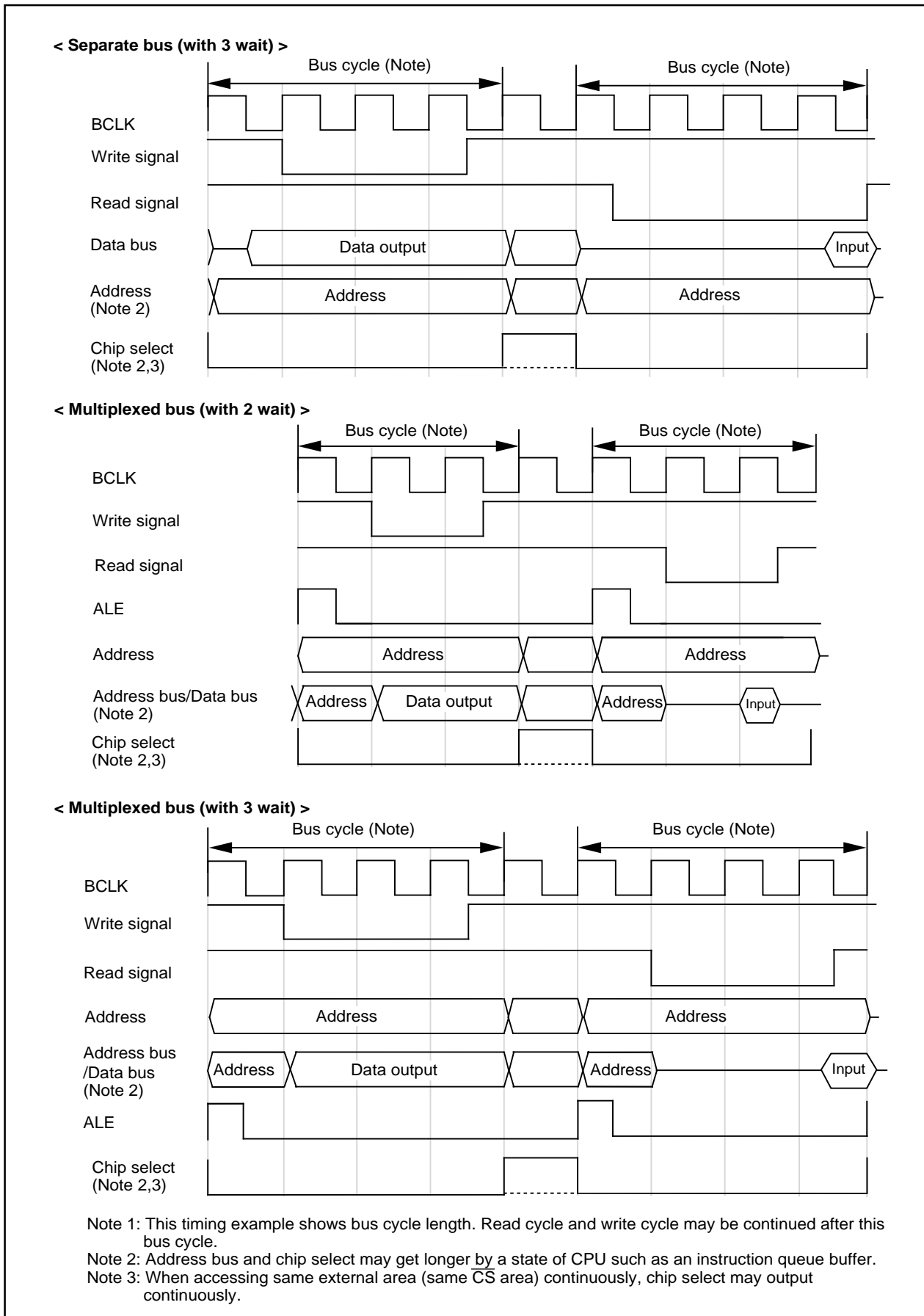


Figure 7.8 Typical bus timings using software wait

### 8. Clock Generating Circuit

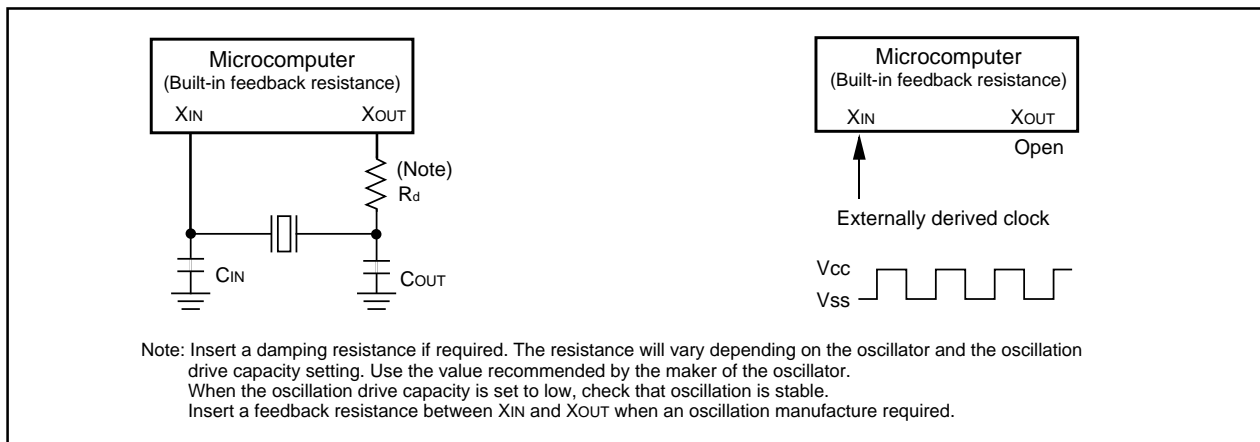
The clock generating circuit contains two oscillator circuits that supply the operating clock sources to the CPU and internal peripheral units.

**Table 8.1 Main clock and sub clock generating circuits**

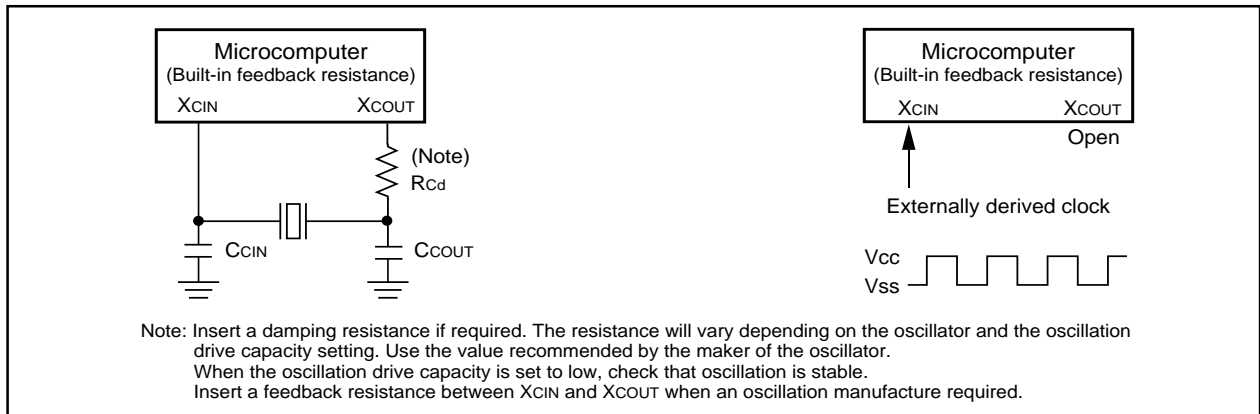
	Main clock generating circuit	Sub clock generating circuit
Use of clock	<ul style="list-style-type: none"> <li>• CPU's operating clock source</li> <li>• Internal peripheral units' operating clock source</li> </ul>	<ul style="list-style-type: none"> <li>• CPU's operating clock source</li> <li>• Timer A/B's count clock source</li> </ul>
Usable oscillator	Ceramic or crystal oscillator	Crystal oscillator
Pins to connect oscillator	XIN, XOUT	XcIN, XcOUT
Oscillation stop/restart function	Available	Available
Oscillator status immediately after reset	Oscillating	Stopped
Other	Externally derived clock can be input	

#### 8.1 Example of oscillator circuit

Figure 8.1 shows some examples of the main clock circuit, one using an oscillator connected to the circuit, and the other one using an externally derived clock for input. Figure 8.2 shows some examples of sub clock circuits, one using an oscillator connected to the circuit, and the other one using an externally derived clock for input. Circuit constants in Figures 8.1 and 8.2 vary with each oscillator used. Use the values recommended by the manufacturer of your oscillator.



**Figure 8.1 Examples of main clock**



**Figure 8.2 Examples of sub clock**



### 8.2 Clock Control

Figure 8.3 shows the block diagram of the clock generating circuit.

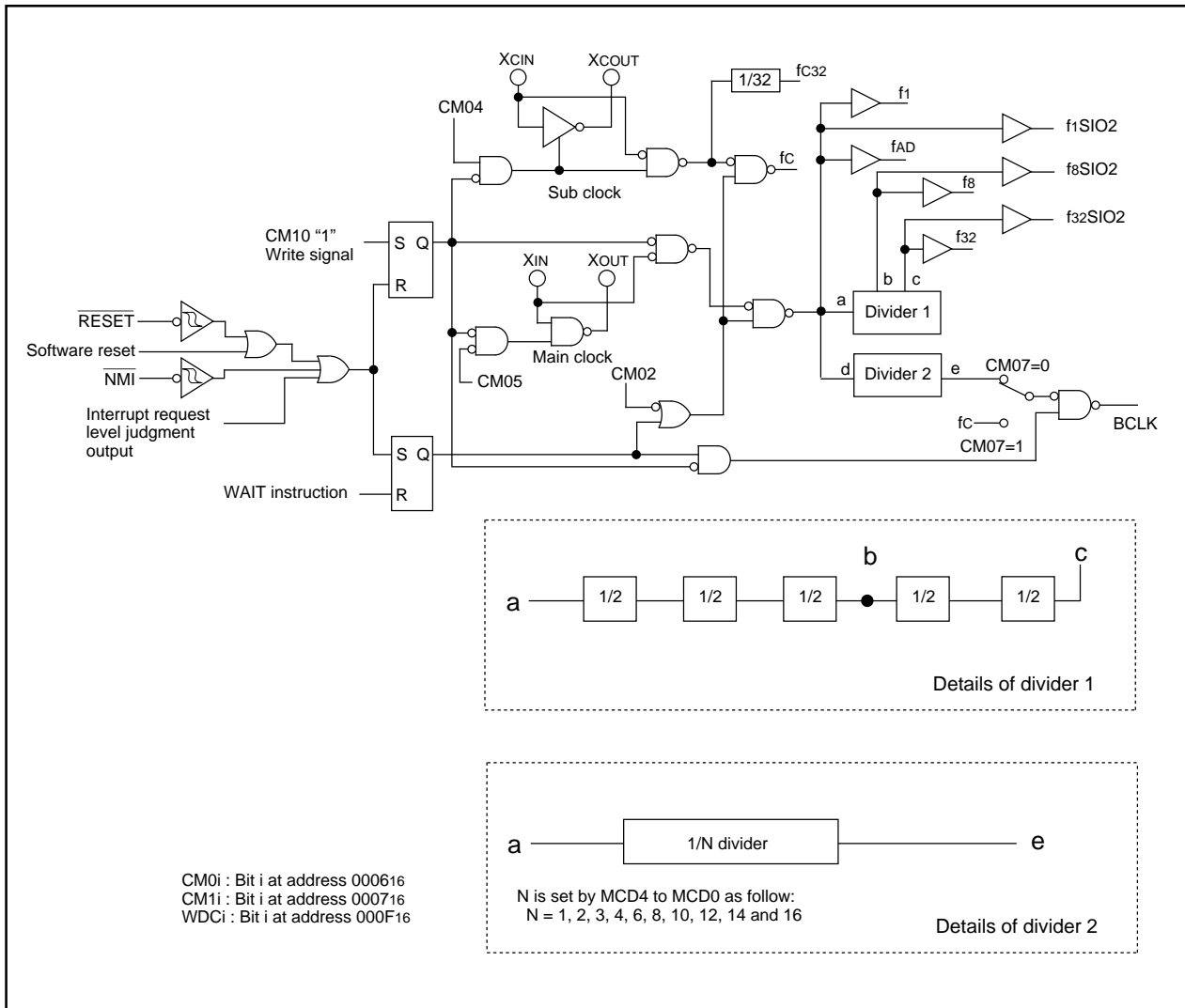


Figure 8.3 Clock generating circuit

The following paragraphs describes the clocks generated by the clock generating circuit.

### (1) Main clock

The main clock is generated by the main clock oscillation circuit. After a reset, the clock is divided by 8 to the BCLK. The clock can be stopped using the main clock stop bit (bit 5 at address 0006<sub>16</sub>). Switching to the sub clock oscillation as CPU operating clock source before stopping the clock reduces the power dissipation.

When the main clock is stopped (bit 5 at address 0006<sub>16</sub> =1) or the mode is shifted to stop mode (bit 0 at address 0007<sub>16</sub> =1), the main clock division register (address 000C<sub>16</sub>) is set to the division by 8 ("08<sub>16</sub>"). After the oscillation of the main clock oscillation circuit has stabilized, the drive capacity of the main clock oscillation circuit can be reduced using the X<sub>IN</sub>-X<sub>OUT</sub> drive capacity select bit (bit 5 at address 0007<sub>16</sub>). Reducing the drive capacity of the main clock oscillation circuit reduces the power dissipation. This bit defaults to "1" when shifting from high-speed or middle-speed mode to stop mode and after a reset.

This bit remains in low-speed and low power dissipation mode.

### (2) Sub clock

The sub clock is generated by the sub clock oscillation circuit. No sub clock is generated after a reset. After oscillation is started using the port Xc select bit (bit 4 at address 0006<sub>16</sub>), the sub clock can be selected as the BCLK by using the system clock select bit (bit 7 at address 0006<sub>16</sub>). However, be sure that the sub clock oscillation has fully stabilized before switching.

After the oscillation of the sub clock oscillation circuit has stabilized, the drive capacity of the sub clock oscillation circuit can be reduced using the X<sub>CIN</sub>-X<sub>COUT</sub> drive capacity select bit (bit 3 at address 0006<sub>16</sub>). Reducing the drive capacity of the sub clock oscillation circuit reduces the power dissipation. This bit changes to "1" when shifting to stop mode and at a reset.

When the sub clock is used, set ports P86 and P87 to no pull-up resistance with the input port.

### (3) BCLK

The BCLK is the clock that drives the CPU, and is either fc or is derived by dividing the main clock by 1, 2, 3, 4, 6, 8, 10, 12, 14 or 16. The BCLK is derived by dividing the main clock by 8 after a reset.

This signal is output from BCLK pin using CM01, CM00 and PM07 in memory expansion mode and microprocessor mode.

When main clock is stopped or shifting to stop mode, the main clock division register (address 000C<sub>16</sub>) is set to the division by 8 ("08<sub>16</sub>").

### (4) Peripheral function clock

- f<sub>1</sub>, f<sub>8</sub>, f<sub>32</sub>, f<sub>1SIO2</sub>, f<sub>8SIO2</sub>, f<sub>32SIO2</sub>

The clock for the peripheral devices is derived from the main clock or by dividing it by 8 or 32. The peripheral function clock is stopped by stopping the main clock or by setting the WAIT peripheral function clock stop bit (bit 2 at 0006<sub>16</sub>) to "1" and then executing a WAIT instruction.

- f<sub>AD</sub>

This clock has the same frequency as the main clock and is used for A/D conversion.

### (5) f<sub>c32</sub>

This clock is derived by dividing the sub clock by 32. It is used for the timer A and timer B counts.

### (6) f<sub>c</sub>

This clock has the same frequency as the sub clock. It is used for BCLK and for the watchdog timer.

Figure 8.4 shows the system clock control registers 0 and 1 and Figure 8.5 shows main clock division register.

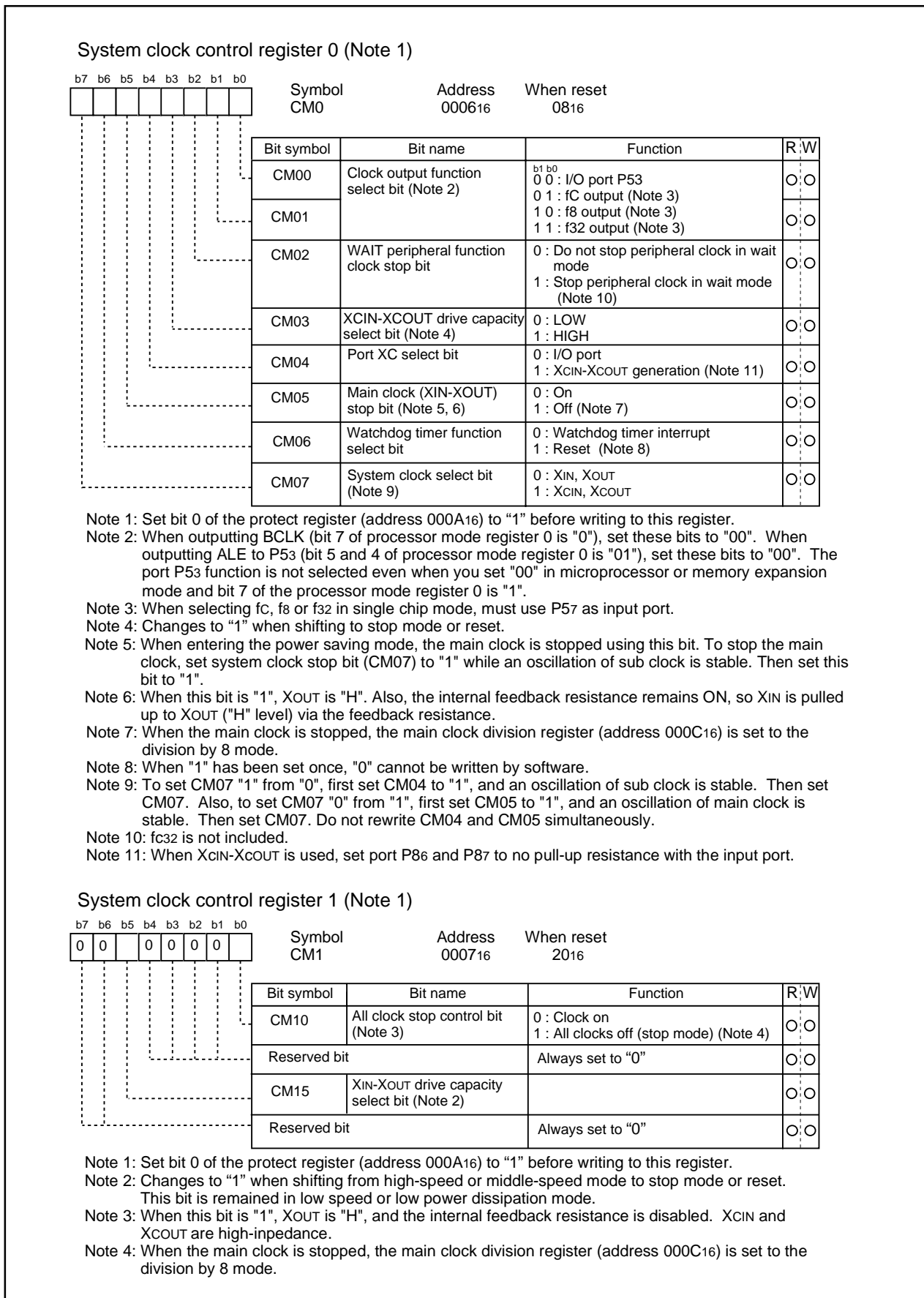
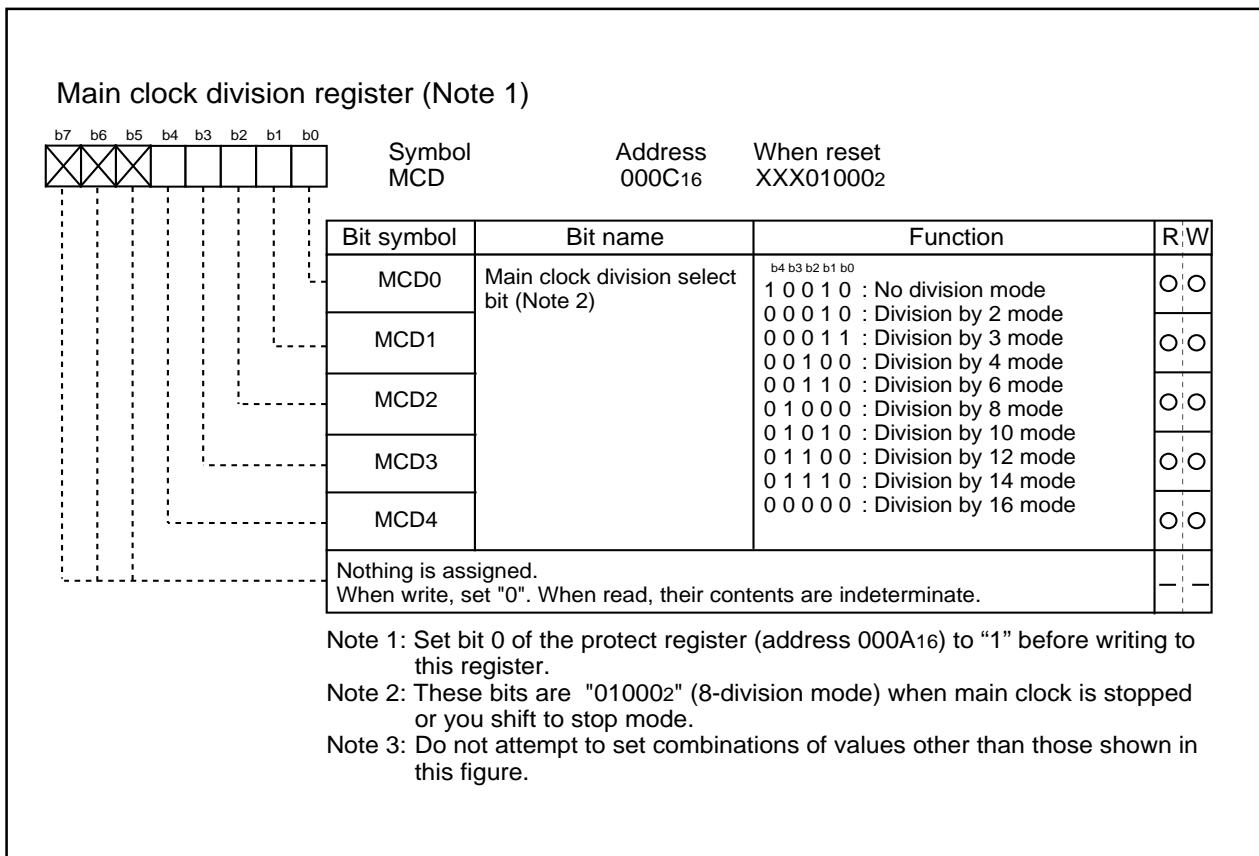


Figure 8.4 System clock control registers 0 and 1



**Figure 8.5 Main clock division register**

### 8.3 Clock Output

In single chip mode, when the BCLK output function select bit (bit 7 at address 000416 :PM07) is "1", you can output f8, f32, or fc from the P53/BCLK/ALE/CLKOUT pins by setting the clock output function select bits (bits 1 and 0 at address 000616 :CM01, CM00).(Note)

Even when you set PM07 to "0" and CM01 and CM00 to "002", no BCLK is output.

In memory expansion mode or microprocessor mode, when the ALE pin select bits (bits 5 and 4 at address 000516 :PM15, PM14) are other than "012(P53/BCLK)" and PM07 is "1", you can output f8, f32, or fc from the P53/BCLK/ALE/CLKOUT pins by setting CM01 and CM00.

In memory expansion mode or microprocessor mode, when PM15 and PM14 are other than "012(P53/BCLK)" and PM07 is "0" and CM01 and CM00 to "002", BCLK is output from the P53/BCLK/ALE/CLKOUT pins.

When stopping clock output in memory expansion mode or microprocessor mode, set PM07 to "1" and CM01 and CM00 to "002" (IO port P53). The P53 function is not selected. When PM15 and PM14 are "012 (P53/BCLK)" and CM01 and CM00 are "002", PM07 is ignored and the P53 pin is set for ALE output.

When the WAIT peripheral function clock stop bit (bit 2 at address 000616) is set to "1", f8 or f32 clock output is stopped when a WAIT command is executed.

Table 8.2 shows clock output setting (single chip mode) and Table 8.3 shows clock output setting (memory expansion/microprocessor mode).

Note :When outputting the f8, f32 or fc from port P53/BCLK/ALE/CLKOUT pin in single chip mode, use port P57/RDY as an input only port.

**Table 8.2 Clock output setting (single chip mode)**

BCLK output function select bit	Clock output function select bit		ALE pin select bit		P53/BCLK/ALE/CLKOUT pin function
	PM07	CM01	CM00	PM15	
0/1	0	0	Ignored	Ignored	P53 I/O port
1	0	1	Ignored	Ignored	fc output (Note)
1	1	0	Ignored	Ignored	f8 output (Note)
1	1	1	Ignored	Ignored	f32 output (Note)

Note :Must use P57 as input port.

**Table 8.3 Clock output setting (memory expansion/microprocessor mode)**

BCLK output function select bit	Clock output function select bit		ALE pin select bit		P53/BCLK/ALE/CLKOUT pin function
	PM07	CM01	CM00	PM15	
0	0	0	0	0	BCLK output
1	0	0			"L" output (not P53)
1	0	1			fc output
1	1	0			f8 output
1	1	1			f32 output
Ignored	0	0	0	1	ALE output

## 8.4 Stop Mode

Writing "1" to the all-clock stop control bit (bit 0 at address 0007<sub>16</sub>) stops all oscillation and the microcomputer enters stop mode. In stop mode, the content of the internal RAM is retained provided that VCC remains above 2V.

Because the oscillation of BCLK, f<sub>1</sub> to f<sub>32</sub>, f<sub>1SIO2</sub> to f<sub>32SIO2</sub>, fc, fc<sub>32</sub>, and fAD stops in stop mode, peripheral functions such as the A/D converter and watchdog timer do not function. However, timer A and timer B operate provided that the event counter mode is set to an external pulse, and UART<sub>i</sub>(i = 0 to 2) functions provided an external clock is selected. Table 8.4 shows the status of the ports in stop mode.

Stop mode is cancelled by a hardware reset or interrupt.

When using an interrupt to exit stop mode, the relevant interrupt must have been enabled and set to a priority level above the level set by the interrupt priority set bits (bits 2, 1, and 0 at address 009F<sub>16</sub>) for exiting a stop/wait state. Set the interrupt priority set bits for the exit from a stop/wait state to the same level as the flag register (FLG) processor interrupt level (IPL). Figure 8.6 shows the exit priority register.

The priority level of the interrupt which is not used to cancel stop mode, must have been changed to 0.

When exiting stop mode using an interrupt, the relevant interrupt routine is executed.

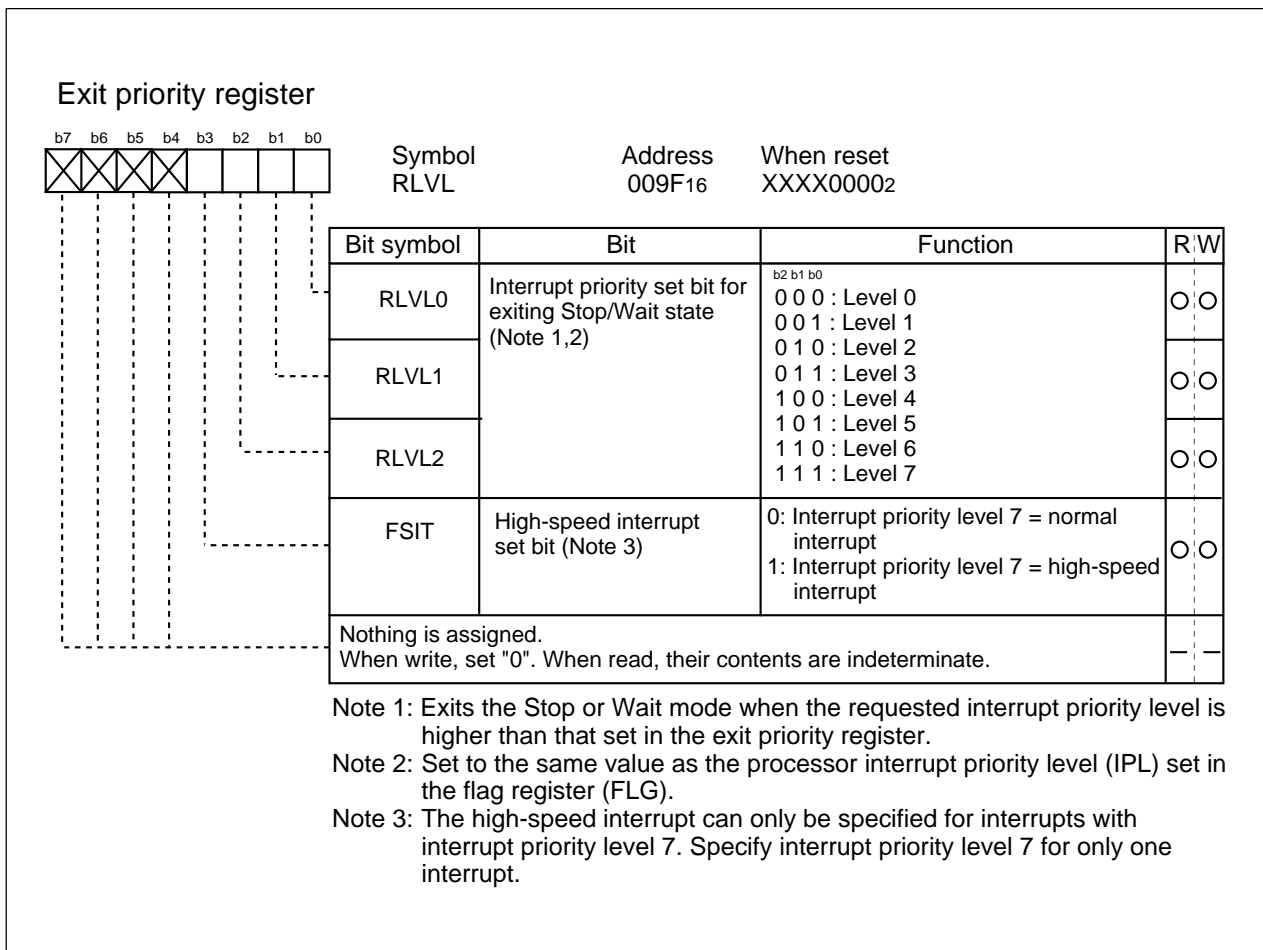
If only a hardware reset or an NMI interrupt is used to cancel stop mode, change the priority level of all interrupt to 0, then shift to stop mode.

When shifting to stop mode and reset, the main clock division register (000C<sub>16</sub>) is set to "08<sub>16</sub>".

**Table 8.4 Port status during stop mode**

Pin		Memory expansion mode Microprocessor mode	Single-chip mode	
Address bus, data bus, CS0 to CS3, BHE		Retains status before stop mode	/	
RD, WR, WRL, WRH, DW, CASL, CASH		"H" (Note)		
RAS		"H" (Note)		
HLDA, BCLK		"H"		
ALE		"H"		
Port		Retains status before stop mode		Retains status before stop mode
CLKOUT	When fc selected	"H"		"H"
	When f8, f32 selected	Retains status before stop mode	Retains status before stop mode	

Note :When self-refresh is done in operating DRAM control, CAS and RAS becomes "L".



**Figure 8.6 Exit priority register**

### 8.5 Wait Mode

When a WAIT instruction is executed, the BCLK stops and the microcomputer enters the wait mode. In this mode, oscillation continues but the BCLK and watchdog timer stop. Writing “1” to the WAIT peripheral function clock stop bit and executing a WAIT instruction stops the clock being supplied to the internal peripheral functions, allowing power dissipation to be reduced. Table 8.5 shows the status of the ports in wait mode.

Wait mode is cancelled by a hardware reset or interrupt. If an interrupt is used to cancel wait mode, the microcomputer restarts using as BCLK the clock that had been selected when the WAIT instruction was executed.

When using an interrupt to exit Wait mode, the relevant interrupt must have been enabled and set to a priority level above the level set by the interrupt priority set bits for exiting a stop/wait state (bits 2, 1, and 0 at address 009F16). Set the interrupt priority set bits for the exit from a stop/wait state to the same level as the flag register (FLG) processor interrupt level (IPL).

The priority level of the interrupt which is not used to cancel wait mode, must have been changed to 0.

When using an interrupt to exit Wait mode, the microcomputer resumes operating the clock that was operating when the WAIT command was executed as BCLK from the interrupt routine.

If only a hardware reset or an NMI interrupt is used to cancel stop mode, change the priority level of all interrupt to 0, then shift to wait mode.

**Table 8.5 Port status during wait mode**

Pin		Memory expansion mode Microprocessor mode	Single-chip mode
Address bus, data bus, $\overline{CS0}$ to $\overline{CS3}$ , BHE		Retains status before wait mode	/
RD, WR, WRL, WRH, DW, CASL, $\overline{CASH}$		“H” (Note)	
RAS		“H” (Note)	
HLDA, BCLK		“H”	
ALE		“L”	
Port		Retains status before wait mode	
CLKOUT	When fc selected	Does not stop	
	When f8, f32 selected	Does not stop when the WAIT peripheral function clock stop bit is “0”. When the WAIT peripheral function clock stop bit is “1”, the status immediately prior to entering wait mode is maintained.	

Note :When self-refresh is done in operating DRAM control,  $\overline{CAS}$  and  $\overline{RAS}$  becomes “L”.

## 8.6 Status Transition of BCLK

Power dissipation can be reduced and low-voltage operation achieved by changing the count source for BCLK. Table 8.6 shows the operating modes corresponding to the settings of system clock control registers 0 and main clock division register.

After a reset, operation defaults to division by 8 mode. When shifting to stop mode, reset or stopping main clock, the main clock division register (address 000C16) is set to "0816".

### (1) Division by 2 mode

The main clock is divided by 2 to obtain the BCLK.

### (2) Division by 3 mode

The main clock is divided by 3 to obtain the BCLK.

### (3) Division by 4 mode

The main clock is divided by 4 to obtain the BCLK.

### (4) Division by 6 mode

The main clock is divided by 6 to obtain the BCLK.

### (5) Division by 8 mode

The main clock is divided by 8 to obtain the BCLK. After reset, this mode is executed. Note that oscillation of the main clock must have stabilized before transferring from this mode to no-division, division by 2, 6, 10, 12, 14 and 16 mode.

Oscillation of the sub clock must have stabilized before transferring to low-speed and low power dissipation mode.

### (6) Division by 10 mode

The main clock is divided by 10 to obtain the BCLK.

### (7) Division by 12 mode

The main clock is divided by 12 to obtain the BCLK.

### (8) Division by 14 mode

The main clock is divided by 14 to obtain the BCLK.

### (9) Division by 16 mode

The main clock is divided by 16 to obtain the BCLK.

### (10) No-division mode

The main clock is divided by 1 to obtain the BCLK.

### (11) Low-speed mode

fc is used as BCLK. Note that oscillation of both the main and sub clocks must have stabilized before transferring from this mode to another or vice versa. At least 2 to 3 seconds are required after the sub clock starts. Therefore, the program must be written to wait until this clock has stabilized immediately after powering up and after stop mode is cancelled.

### (12) Low power dissipation mode

fc is the BCLK and the main clock is stopped.

When the main clock is stopped, the main clock division register (address 000C16) is set to the division by 8 mode.



Note: When count source of BCLK is changed from clock A to clock B (XIN to XCIN or XCIN to XIN), clock B needs to be stable before changing. Please wait to change modes until after oscillation has stabilized.

**Table 8.6 Operating modes dictated by settings of system clock control register 0 and main clock division register**

CM07	CM05	CM04	MCD4	MCD3	MCD2	MCD1	MCD0	Operating mode of BCLK
0	0	Invalid	1	0	0	1	0	No division
0	0	Invalid	0	0	0	1	0	Division by 2 mode
0	0	Invalid	0	0	0	1	1	Division by 3 mode
0	0	Invalid	0	0	1	0	0	Division by 4 mode
0	0	Invalid	0	0	1	1	0	Division by 6 mode
0	0	Invalid	0	1	0	0	0	Division by 8 mode
0	0	Invalid	0	1	0	1	0	Division by 10 mode
0	0	Invalid	0	1	1	0	0	Division by 12 mode
0	0	Invalid	0	1	1	1	0	Division by 14 mode
0	0	Invalid	0	0	0	0	0	Division by 16 mode
1	0	1	Invalid	Invalid	Invalid	Invalid	Invalid	Low-speed mode
1	1	1	Invalid	Invalid	Invalid	Invalid	Invalid	Low power dissipation mode

CM0i: Clock control register 0 (address 0006<sub>16</sub>) bit i

MCDi: Main clock division register (address 000C<sub>16</sub>) bit i

## 8.7 Power Saving

In Power Save modes, the CPU and oscillator stop and the operating clock is slowed to minimize power dissipation by the CPU. The following outlines the Power Save modes.

There are three power save modes.

### (1) Normal operating mode

- **High-speed mode**

In this mode, one main clock cycle forms BCLK. The CPU operates on the selected internal clock. The peripheral functions operate on the clocks specified for each respective function.

- **Medium-speed mode**

In this mode, the main clock is divided into 2, 3, 4, 6, 8, 10, 12, 14, or 16 to form BCLK. The CPU operates on the selected internal clock. The peripheral functions operated on the clocks specified for each respective function.

- **Low-speed mode**

In this mode, fc forms BCLK. The CPU operates on the fc clock. fc is the clock supplied by the subclock. The peripheral functions operate on the clocks specified for each respective function.

- **Low power-dissipation mode**

This mode is selected when the main clock is stopped from low-speed mode. The CPU operates on the fc clock. fc is the clock supplied by the subclock. Only the peripheral functions for which the subclock was selected as the count source continue to run.

### (2) Wait mode

CPU operation is halted in this mode. The oscillator continues to run.

### (3) Stop mode

All oscillators stop in this mode. The CPU and internal peripheral functions all stop. Of all 3 power saving modes, power savings are greatest in this mode.

Figure 8.7 shows the clock transition between each of the three modes, (1), (2), and (3).

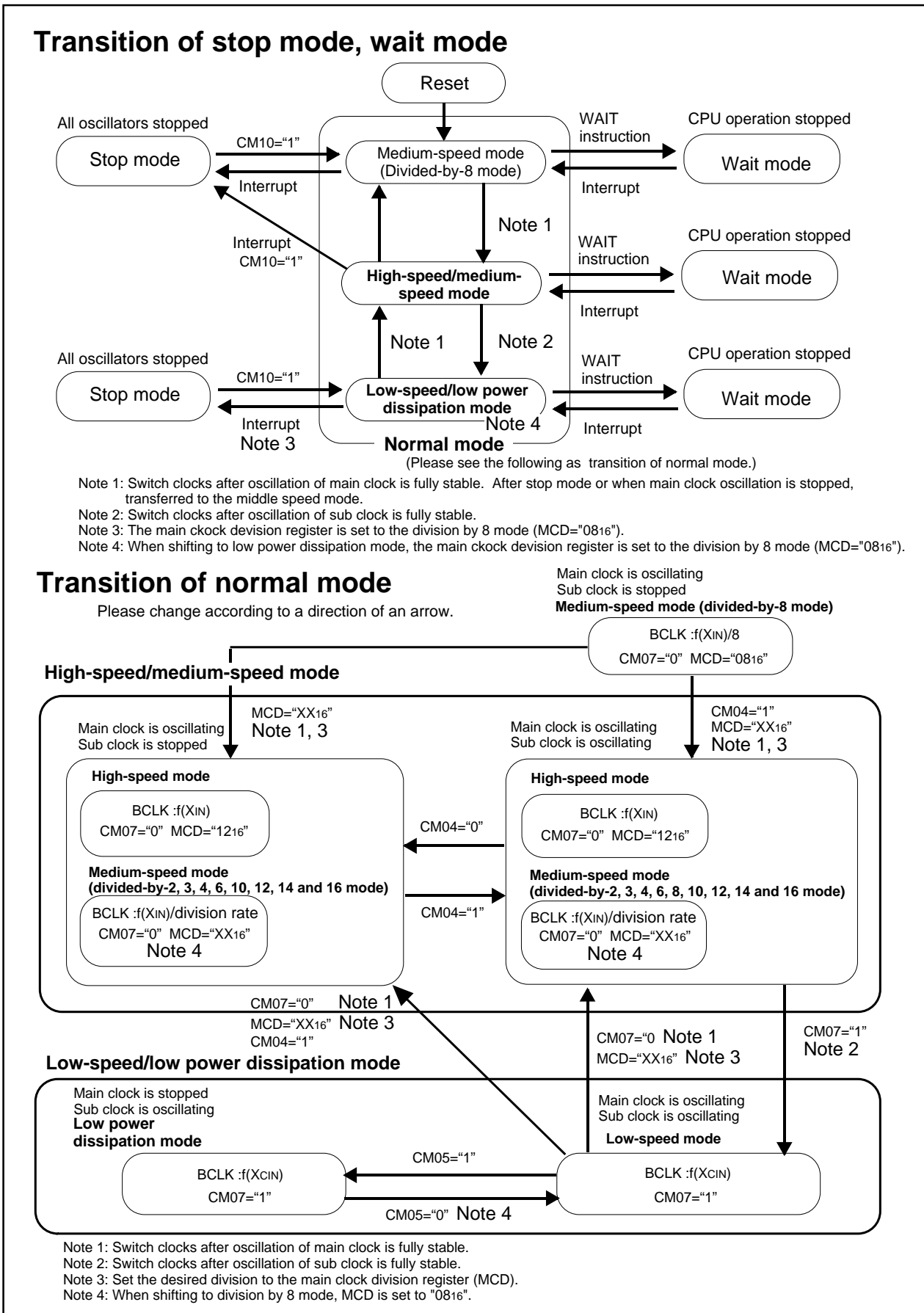


Figure 8.7 Clock transition

### 8.8 Protection

The protection function is provided so that the values in important registers cannot be changed in the event that the program runs out of control. Figure 8.8 shows the protect register. The values in the processor mode register 0 (address 0004<sub>16</sub>), processor mode register 1 (address 0005<sub>16</sub>), system clock control register 0 (address 0006<sub>16</sub>), system clock control register 1 (address 0007<sub>16</sub>), main clock division register (address 000C<sub>16</sub>), port P9 direction register (address 03C7<sub>16</sub>) and function select register A3 (address 03B5<sub>16</sub>) can only be changed when the respective bit in the protect register is set to "1". Therefore, important outputs can be allocated to port P9.

If, after "1" (write-enabled) has been written to the PRC2 (bit 2 at address 000A<sub>16</sub>), a value is written to any address, the bit automatically reverts to "0" (write-inhibited). Change port P9 input/output and function select register A3 immediately after setting "1" to PRC2. Interrupt and DMA transfer should not be inserted between instructions. However, the PRC0 (bit 0 at address 000A<sub>16</sub>) and PRC1 (bit 1 at address 000A<sub>16</sub>) do not automatically return to "0" after a value has been written to an address. The program must therefore be written to return these bits to "0".

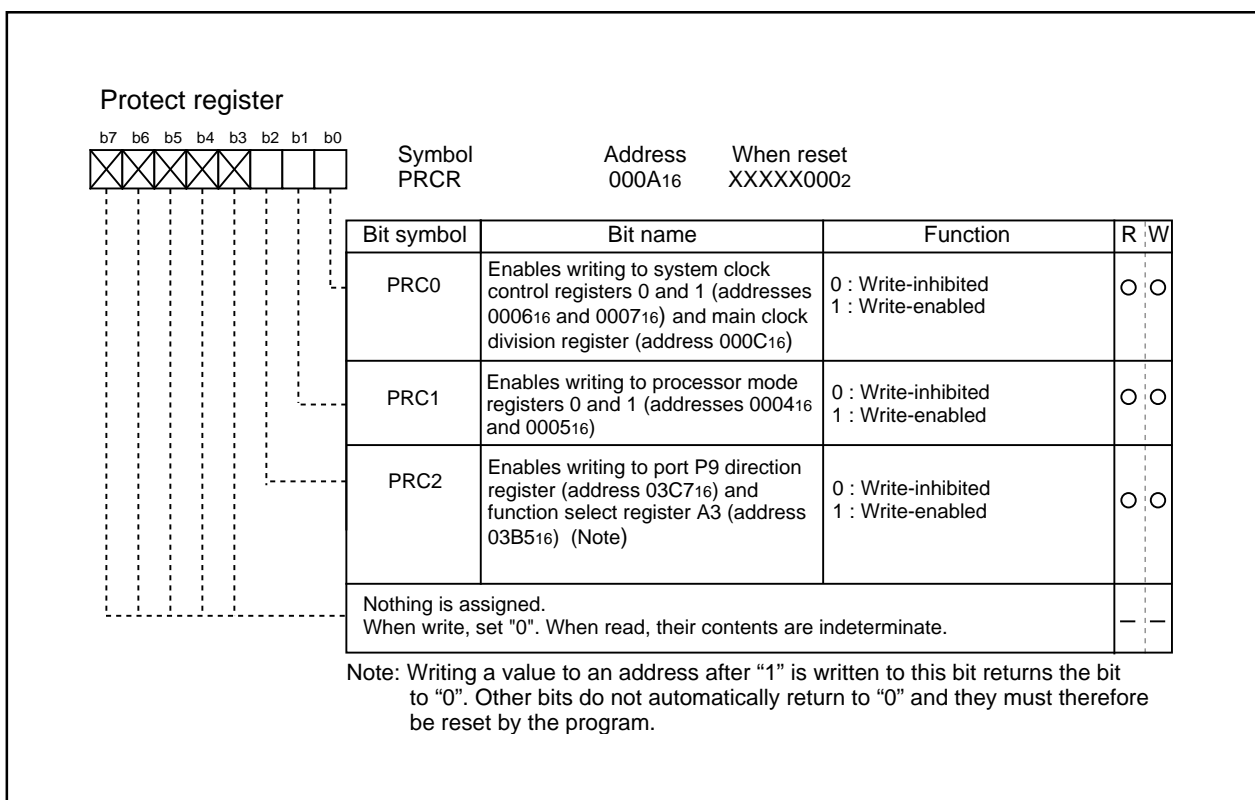
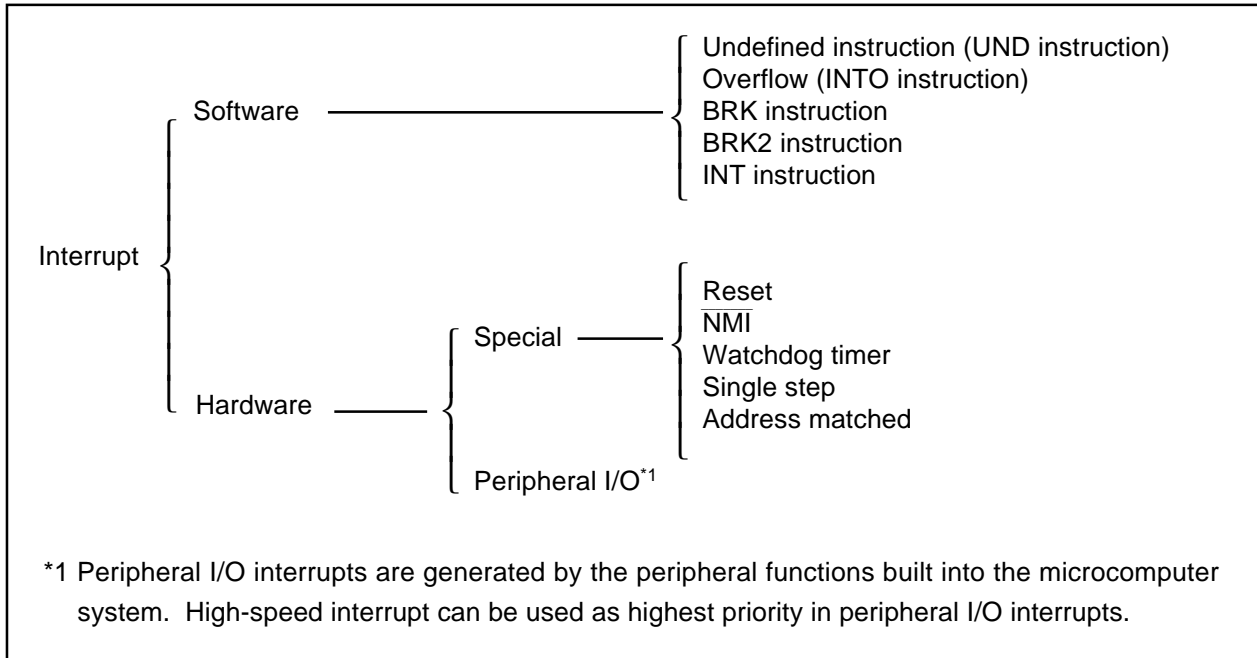


Figure 8.8 Protect register

## 9. Interrupt Outline

### 9.1 Types of Interrupts

Figure 9.1 lists the types of interrupts.



**Figure 9.1 Classification of 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.

## 9.2 Software Interrupts

Software interrupts are generated by some instruction that generates an interrupt request when executed. Software interrupts are nonmaskable interrupts.

### (1) Undefined-instruction interrupt

This interrupt occurs when the UND instruction is executed.

### (2) Overflow interrupt

This interrupt occurs if the INTO instruction is executed when the O flag is 1.

The following lists the instructions that cause the O flag to change:

ABS, ADC, ADCF, ADD, ADDX, CMP, CMPX, DIV, DIVU, DIVX, NEG, RMPA, SBB, SCMPU, SHA, SUB, SUBX

### (3) BRK interrupt

This interrupt occurs when the BRK instruction is executed.

### (4) BRK2 interrupt

This interrupt occurs when the BRK2 instruction is executed. This interrupt is used exclusively for debugger purposes. You normally do not need to use this interrupt.

### (5) INT instruction interrupt

This interrupt occurs when the INT instruction is executed after specifying a software interrupt number from 0 to 63. Note that software interrupt numbers 0 to 43 are assigned to peripheral I/O interrupts. This means that by executing the INT instruction, you can execute the same interrupt routine as used in peripheral I/O interrupts.

The stack pointer used in INT instruction interrupt varies depending on the software interrupt number. For software interrupt numbers 0 to 31, the U flag is saved when an interrupt occurs and the U flag is cleared to 0 to choose the interrupt stack pointer (ISP) before executing the interrupt sequence. The previous U flag before the interrupt occurred is restored when control returns from the interrupt routine. For software interrupt numbers 32 to 63, such stack pointer switchover does not occur.

However, in peripheral I/O interrupts, the U flag is saved when an interrupt occurs and the U flag is cleared to 0 to choose ISP.

Therefore movement of U flag is different by peripheral I/O interrupt or INT instruction in software interrupt number 32 to 43.

### 9.3 Hardware Interrupts

There are Two types in hardware Interrupts; special interrupts and Peripheral I/O interrupts.

#### (1) Special interrupts

Special interrupts are nonmaskable interrupts.

- **Reset**

A reset occurs when the  $\overline{\text{RESET}}$  pin is pulled low.

- **$\overline{\text{NMI}}$  interrupt**

This interrupt occurs when the  $\overline{\text{NMI}}$  pin is pulled low.

- **Watchdog timer interrupt**

This interrupt is caused by the watchdog timer.

- **Address-match interrupt**

This interrupt occurs immediately before the instruction at the address indicated by the address match interrupt register is executed while the address match interrupt enable bit is set to "1".

This interrupt does not occur if any address other than the start address of an instruction is set in the address match register.

- **Single-step interrupt**

This interrupt is used exclusively for debugger purposes, do not use it in other circumstances. A single-step interrupt occurs when the D flag is set (= 1); in this case, an interrupt is generated after one instruction is executed.

#### (2) Peripheral I/O interrupts

A peripheral I/O interrupt is generated by one of built-in peripheral functions. Built-in peripheral functions are dependent on classes of products, so the interrupt factors too are dependent on classes of products. The interrupt vector table is the same as the one for software interrupt numbers 0 through 43 the INT instruction uses. Peripheral I/O interrupts are maskable interrupts.

- **Bus collision detection, start/stop condition detection interrupts (UART2, UART3, UART4), fault error interrupts (UART3, 4)**

This is an interrupt that the serial I/O bus collision detection generates. When I<sup>2</sup>C mode is selected, start, stop condition interrupt is selected. When  $\overline{\text{SS}}$  pin is selected, fault error interrupt is selected.

- **DMA0 through DMA3 interrupts**

These are interrupts that DMA generates.

- **Key-input interrupt**

A key-input interrupt occurs if an "L" is input to the  $\overline{\text{KI}}$  pin.

- **A/D conversion interrupt**

This is an interrupt that the A/D converter generates.

- **UART0, UART1, UART2/NACK, UART3/NACK and UART4/NACK transmission interrupt**

These are interrupts that the serial I/O transmission generates.

- **UART0, UART1, UART2/ACK, UART3/ACK and UART4/ACK reception interrupt**

These are interrupts that the serial I/O reception generates.

- **Timer A0 interrupt through timer A4 interrupt**

These are interrupts that timer A generates

- **Timer B0 interrupt through timer B5 interrupt**

These are interrupts that timer B generates.

- **$\overline{\text{INT0}}$  interrupt through  $\overline{\text{INT5}}$  interrupt**

An  $\overline{\text{INT}}$  interrupt selects a edge sense or a level sense. In edge sense, an  $\overline{\text{INT}}$  interrupt occurs if either a rising edge or a falling edge or a both edge is input to the  $\overline{\text{INT}}$  pin. In level sense, an  $\overline{\text{INT}}$  interrupt occurs if either an "H" level or an "L" level is input to the  $\overline{\text{INT}}$  pin.

## 9.4 High-speed interrupts

High-speed interrupts are interrupts in which the response is executed at 5 cycles and the return is 3 cycles.

When a high-speed interrupt is received, the flag register (FLG) and program counter (PC) are saved to the save flag register (SVF) and save PC register (SVP) and the program is executed from the address shown in the vector register (VCT).

Execute a FREIT instruction to return from the high-speed interrupt routine.

High-speed interrupts can be set by setting "1" in the high-speed interrupt specification bit allocated to bit 3 of the exit priority register. Setting "1" in the high-speed interrupt specification bit makes the interrupt set to level 7 in the interrupt control register into a high-speed interrupt.

You can only set one interrupt as a high-speed interrupt. When using a high-speed interrupt, do not set multiple interrupts as level 7 interrupts.

The interrupt vector for a high-speed interrupt must be set in the vector register (VCT).

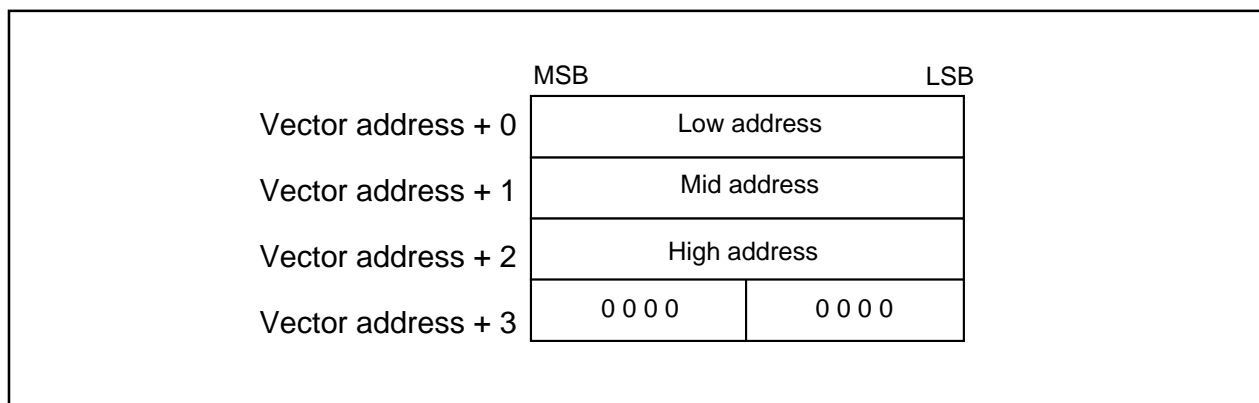
When using a high-speed interrupt, you can use a maximum of two DMAC channels.

The execution speed is improved when register bank 1 is used with high speed interrupt register selected by not saving registers to the stack but to the switching register bank. In this case, switch register bank mode for high-speed interrupt routine.

## 9.5 Interrupts and Interrupt Vector Tables

If an interrupt request is accepted, a program branches to the interrupt routine set in the interrupt vector table. Set the first address of the interrupt routine in each vector table. Figure 9.2 shows the format for specifying the address.

Two types of interrupt vector tables are available — fixed vector table in which addresses are fixed and variable vector table in which addresses can be varied by the setting.



**Figure 9.2 Format for specifying interrupt vector addresses**



- **Fixed vector tables**

The fixed vector table is a table in which addresses are fixed. The vector tables are located in an area extending from FFFFDC<sub>16</sub> to FFFFFFF<sub>16</sub>. One vector table comprises four bytes. Set the first address of interrupt routine in each vector table. Table 9.1 shows the interrupts assigned to the fixed vector tables and addresses of vector tables.

**Table 9.1 Interrupt factors (fixed interrupt vector addresses)**

Interrupt source	Vector table addresses Address (L) to address (H)	Remarks
Undefined instruction	FFFFDC <sub>16</sub> to FFFFDF <sub>16</sub>	Interrupt on UND instruction
Overflow	FFFFE0 <sub>16</sub> to FFFF E3 <sub>16</sub>	Interrupt on INTO instruction
BRK instruction execution	FFFFE4 <sub>16</sub> to FFFF E7 <sub>16</sub>	If content of FFFF E7 <sub>16</sub> is filled with FF <sub>16</sub> , program starts from the address shown by the vector in the variable vector table
Address match	FFFFE8 <sub>16</sub> to FFFF EB <sub>16</sub>	There is an address-matching interrupt enable bit
Watchdog timer	FFFFF0 <sub>16</sub> to FFFF F3 <sub>16</sub>	
NMI	FFFFF8 <sub>16</sub> to FFFF FB <sub>16</sub>	External interrupt by input to NMI pin
Reset	FFFFFC <sub>16</sub> to FFFF FF <sub>16</sub>	

- **Vector table dedicated for emulator**

Table 9.2 shows interrupt vector address which is vector table register dedicated for emulator (address 000020<sub>16</sub> to 000022<sub>16</sub>). These instructions are not effected with interrupt enable flag (I flag) (non maskable interrupt).

This interrupt is used exclusively for debugger purposes. You normally do not need to use this interrupt. Do not access to the interrupt vector table register dedicated for emulator (address 000020<sub>16</sub> to 000022<sub>16</sub>).

**Table 9.2 Interrupt vector table register for emulator**

Interrupt source	Vector table addresses Address (L) to address (H)	Remarks
BRK2 instruction	Interrupt vector table register for emulator 000020 <sub>16</sub> to 000022 <sub>16</sub>	Interrupt for debugger
Single step	Interrupt vector table register for emulator 000020 <sub>16</sub> to 000022 <sub>16</sub>	Interrupt for debugger

- **Variable vector tables**

The addresses in the variable vector table can be modified, according to the user's settings. Indicate the first address using the interrupt table register (INTB). The 256-byte area subsequent to the address the INTB indicates becomes the area for the variable vector tables. One vector table comprises four bytes. Set the first address of the interrupt routine in each vector table. Table 9.3 shows the interrupts assigned to the variable vector tables and addresses of vector tables.

Set an even address to the start address of vector table setting in INTB so that operating efficiency is increased.

**Table 9.3 Interrupt causes (variable interrupt vector addresses)**

Software interrupt number	Vector table address Address (L) to address (H)	Interrupt source	Remarks
Software interrupt number 0	+0 to +3 (Note 1)	BRK instruction	Cannot be masked I flag
Software interrupt number 8	+32 to +35 (Note 1)	DMA0	
Software interrupt number 9	+36 to +39 (Note 1)	DMA1	
Software interrupt number 10	+40 to +43 (Note 1)	DMA2	
Software interrupt number 11	+44 to +47 (Note 1)	DMA3	
Software interrupt number 12	+48 to +51 (Note 1)	Timer A0	
Software interrupt number 13	+52 to +55 (Note 1)	Timer A1	
Software interrupt number 14	+56 to +59 (Note 1)	Timer A2	
Software interrupt number 15	+60 to +63 (Note 1)	Timer A3	
Software interrupt number 16	+64 to +67 (Note 1)	Timer A4	
Software interrupt number 17	+68 to +71 (Note 1)	UART0 transmit	
Software interrupt number 18	+72 to +75 (Note 1)	UART0 receive	
Software interrupt number 19	+76 to +79 (Note 1)	UART1 transmit	
Software interrupt number 20	+80 to +83 (Note 1)	UART1 receive	
Software interrupt number 21	+84 to +87 (Note 1)	Timer B0	
Software interrupt number 22	+88 to +91 (Note 1)	Timer B1	
Software interrupt number 23	+92 to +95 (Note 1)	Timer B2	
Software interrupt number 24	+96 to +99 (Note 1)	Timer B3	
Software interrupt number 25	+100 to +103 (Note 1)	Timer B4	
Software interrupt number 26	+104 to +107 (Note 1)	$\overline{\text{INT5}}$	
Software interrupt number 27	+108 to +111 (Note 1)	$\overline{\text{INT4}}$	
Software interrupt number 28	+112 to +115 (Note 1)	$\overline{\text{INT3}}$	
Software interrupt number 29	+116 to +119 (Note 1)	$\overline{\text{INT2}}$	
Software interrupt number 30	+120 to +123 (Note 1)	$\overline{\text{INT1}}$	
Software interrupt number 31	+124 to +127 (Note 1)	$\overline{\text{INT0}}$	
Software interrupt number 32	+128 to +131 (Note 1)	Timer B5	
Software interrupt number 33	+132 to +135 (Note 1)	UART2 transmit/NACK (Note 2)	
Software interrupt number 34	+136 to +139 (Note 1)	UART2 receive/ACK (Note 2)	
Software interrupt number 35	+140 to +143 (Note 1)	UART3 transmit/NACK (Note 2)	
Software interrupt number 36	+144 to +147 (Note 1)	UART3 receive/ACK (Note 2)	
Software interrupt number 37	+148 to +151 (Note 1)	UART4 transmit/NACK (Note 2)	
Software interrupt number 38	+152 to +155 (Note 1)	UART4 receive/ACK (Note 2)	
Software interrupt number 39	+156 to +159 (Note 1)	Bus collision detection, start/stop condition detection (UART2) (Note 2)	
Software interrupt number 40	+160 to +163 (Note 1)	Bus collision detection, start/stop condition detection, fault error (UART3) (Note 2, 3)	
Software interrupt number 41	+164 to +167 (Note 1)	Bus collision detection, start/stop condition detection, fault error (UART4) (Note 2, 3)	
Software interrupt number 42	+168 to +171 (Note 1)	A/D	
Software interrupt number 43	+172 to +175 (Note 1)	Key input interrupt	
Software interrupt number 44 to Software interrupt number 63	+176 to +179 (Note 1) to +252 to +255 (Note 1)	Software interrupt	Cannot be masked I flag

Note 1: Address relative to address in interrupt table register (INTB).

Note 2: When I<sup>2</sup>C mode is selected, NACK/ACK, start/stop condition detection interrupts are selected.

Note 3: The fault error interrupt is selected when SS pin is selected.

### 9.6 Interrupt control registers

Peripheral I/O interrupts have their own interrupt control registers. Figure 9.3 shows the interrupt control registers.

When using an interrupt to exit Stop mode or Wait mode, the relevant interrupt must have been enabled and set to a priority level above the level set by the interrupt priority set bits for exit a stop/wait state (bits 2, 1, and 0 at address 009F16). Set the interrupt priority set bits for the exit from a stop/wait state to the same level as the flag register (FLG) processor interrupt level (IPL).

Figure 9.4 shows the exit priority register.

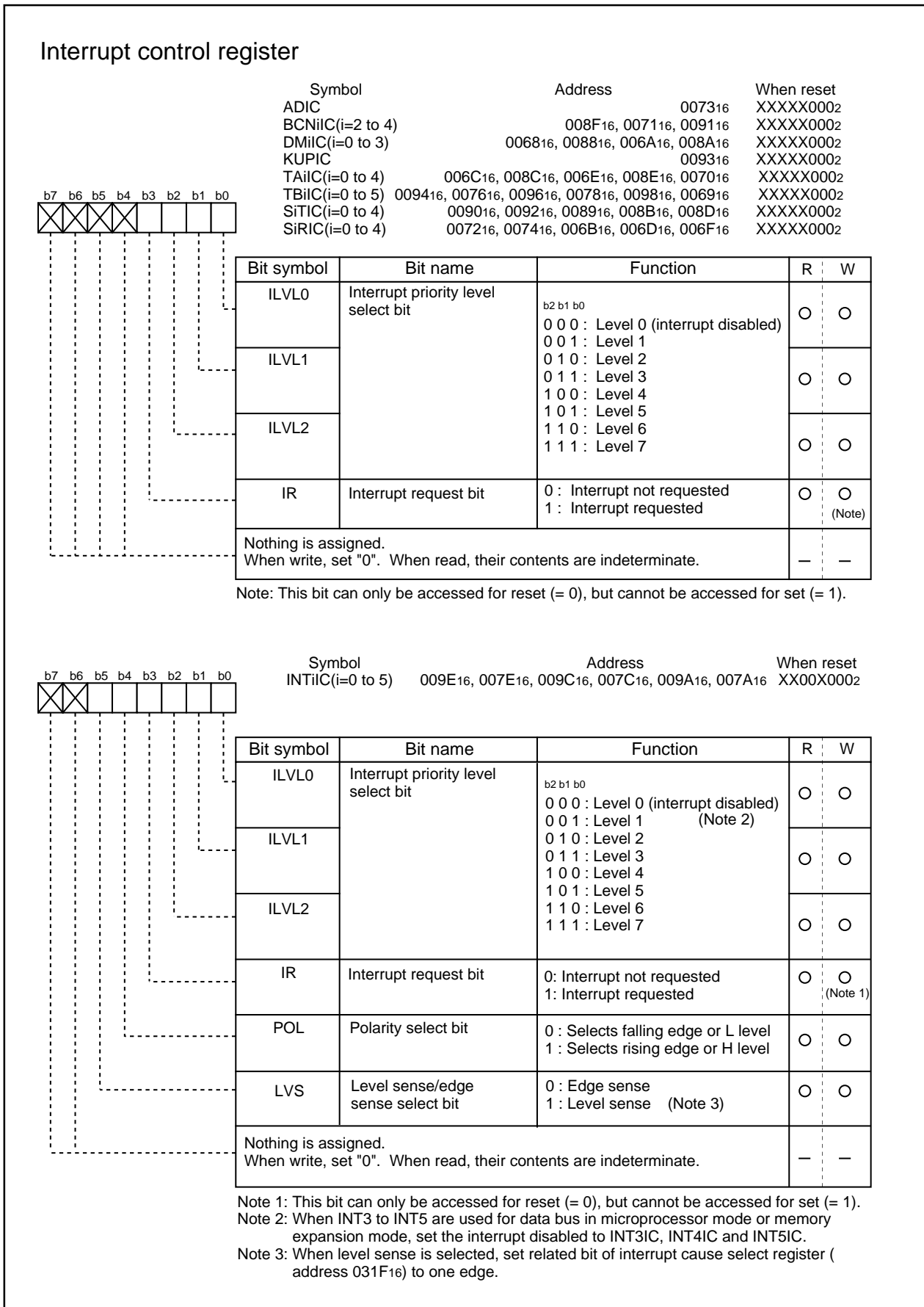


Figure 9.3 Interrupt control register

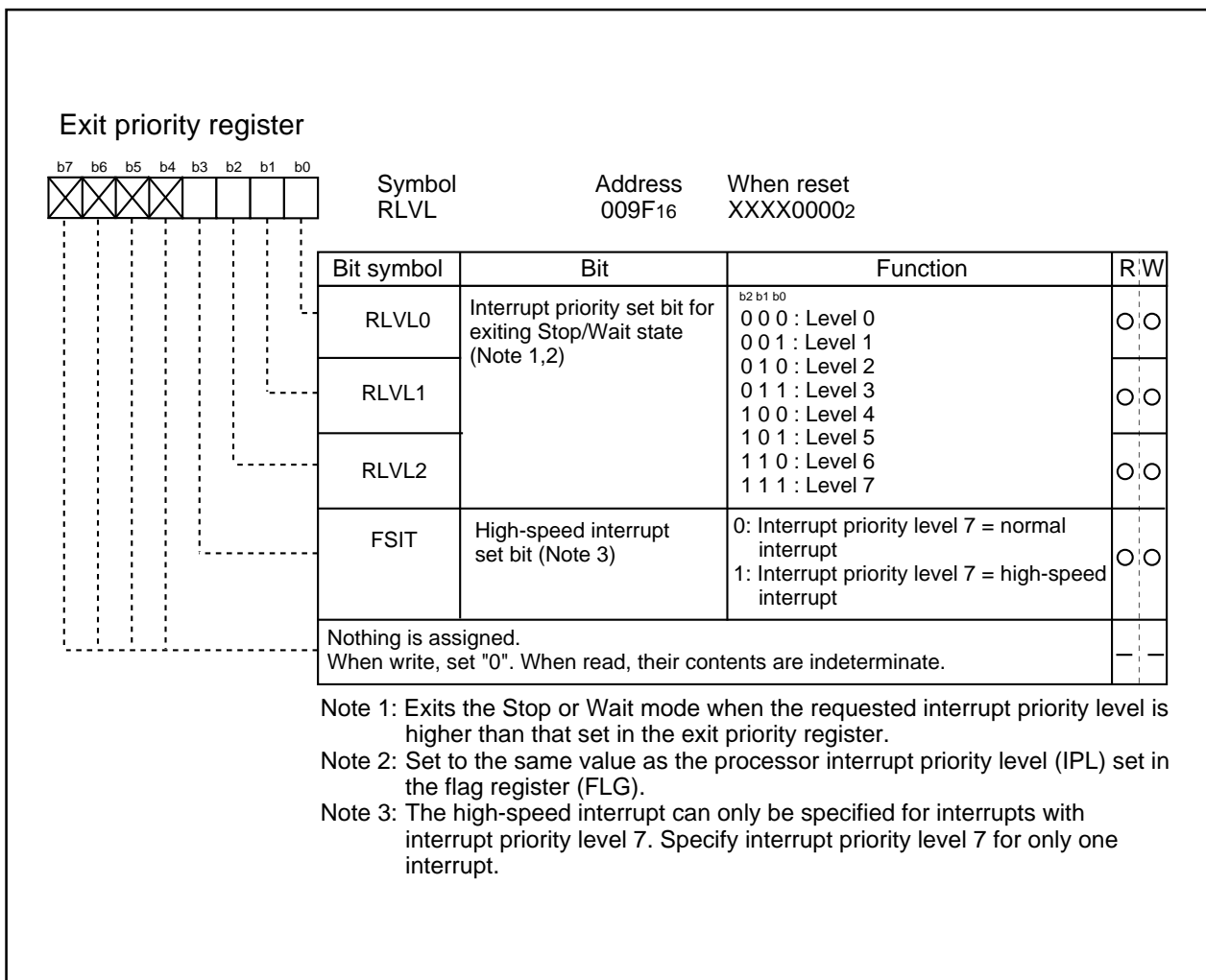


Figure 9.4 Exit priority register

**9.7 Interrupt Enable Flag (I Flag)**

The interrupt enable flag (I flag) is used to disable/enable maskable interrupts. When this flag is set (= 1), all maskable interrupts are enabled; when the flag is cleared to 0, they are disabled. This flag is automatically cleared to 0 after a reset is cleared.

**9.8 Interrupt Request Bit**

This bit is set (= 1) by hardware when an interrupt request is generated. The bit is cleared to 0 by hardware when the interrupt request is acknowledged and jump to the interrupt vector. This bit can be cleared to 0 (but cannot be set to 1) in software.

**9.9 Interrupt Priority Level Select Bit and Processor Interrupt Priority Level (IPL)**

Interrupt priority levels are set by the interrupt priority select bit in an interrupt control register. When an interrupt request is generated, the interrupt priority level of this interrupt is compared with the processor interrupt priority level (IPL). This interrupt is enabled only when its interrupt priority level is greater than the processor interrupt priority level (IPL). This means that you can disable any particular interrupt by setting its interrupt priority level to 0.

Table 9.4 shows how interrupt priority levels are set. Table 9.5 shows interrupt enable levels in relation to the processor interrupt priority level (IPL).

The following lists the conditions under which an interrupt request is acknowledged:

- Interrupt enable flag (I flag) = 1
- Interrupt request bit = 1
- Interrupt priority level > Processor interrupt priority level (IPL)

The interrupt enable flag (I flag), interrupt request bit, interrupt priority level select bit, and the processor interrupt priority level (IPL) all are independent of each other, so they do not affect any other bit.

**Table 9.4 Interrupt Priority Levels**

Interrupt priority level select bit	Interrupt priority level	Priority order
b2 b1 b0 0 0 0	Level 0 (interrupt disabled)	———
0 0 1	Level 1	↓ Low        ↓ High
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	

**Table 9.5 IPL and Interrupt Enable Levels**

Processor interrupt priority level (IPL)	Enabled interrupt priority levels
IPL <sub>2</sub> IPL <sub>1</sub> IPL <sub>0</sub> 0 0 0	Interrupt levels 1 and above are enabled.
0 0 1	Interrupt levels 2 and above are enabled.
0 1 0	Interrupt levels 3 and above are enabled.
0 1 1	Interrupt levels 4 and above are enabled.
1 0 0	Interrupt levels 5 and above are enabled.
1 0 1	Interrupt levels 6 and above are enabled.
1 1 0	Interrupt levels 7 and above are enabled.
1 1 1	All maskable interrupts are disabled.

### 9.10 Rewrite the interrupt control register

When an instruction to rewrite the interrupt control register is executed but the interrupt is disabled, the interrupt request bit is not set sometimes even if the interrupt request for that register has been generated. This will depend on the instruction. If this creates problems, use the below instructions to change the register.

Instructions : AND, OR, BCLR, BSET

### 9.11 Interrupt Sequence

An interrupt sequence — what are performed over a period 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 SCMPU, SIN, SMOVB, SMOVF, SMOVU, SSTR, SOUT or RMPA instruction, the processor temporarily suspends the instruction being executed, and transfers control to the interrupt sequence.

In the interrupt sequence, the processor carries out the following in sequence given:

- (1) CPU gets the interrupt information (the interrupt number and interrupt request level) by reading address 000000<sub>16</sub> (address 000002<sub>16</sub> when high-speed interrupt). After this, the related interrupt request bit is "0".
- (2) Saves the content of the flag register (FLG) as it was immediately before the start of interrupt sequence in the temporary register (Note) within the CPU.
- (3) Sets the interrupt enable flag (I flag), the debug flag (D flag), and the stack pointer select flag (U flag) to "0" (the U flag, however does not change if the INT instruction, in software interrupt numbers 32 through 63, is executed)
- (4) Saves the content of the temporary register (Note 1) within the CPU in the stack area. Saves in the flag save register (SVF) in high-speed interrupt.
- (5) Saves the content of the program counter (PC) in the stack area. Saves in the PC save register (SVP) in high-speed interrupt.
- (6) Sets the interrupt priority level of the accepted instruction in the IPL.

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

Note: This register cannot be utilized by the user.

### 9.12 Interrupt Response Time

'Interrupt response time' is the period between the instant an interrupt occurs and the instant the first instruction within the interrupt routine has been executed. This time comprises the period from the occurrence of an interrupt to the completion of the instruction under execution at that moment (a) and the time required for executing the interrupt sequence (b). Figure 9.5 shows the interrupt response time.

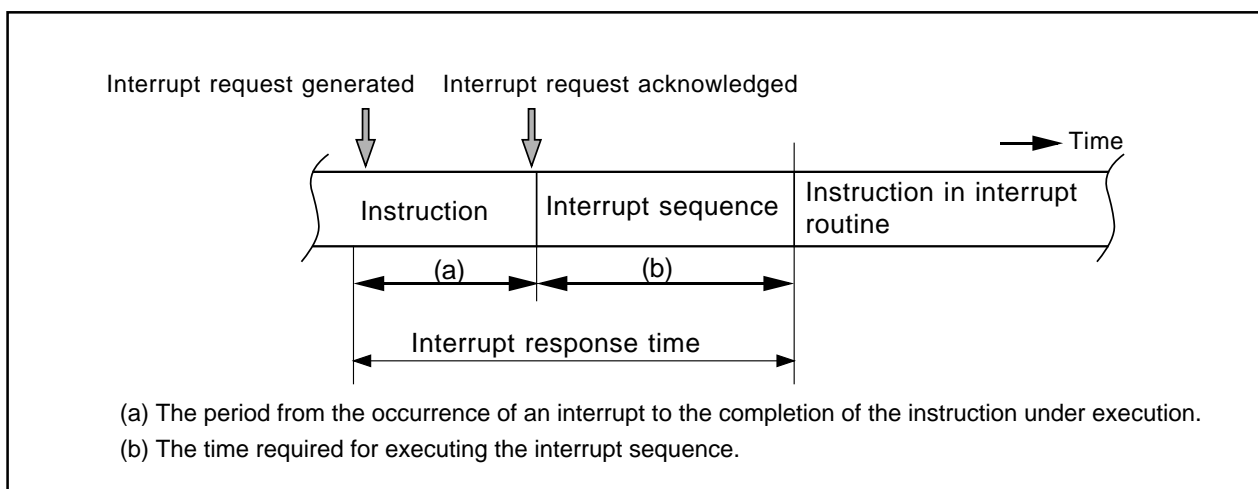


Figure 9.5 Interrupt response time

Time (a) varies with each instruction being executed. The DIVX instruction requires a maximum time that consists of  $24^*$  cycles.

Time (b) is shown in Table 9.6.

\* It is when the divisor is immediate or register. When the divisor is memory, the following value is added.

- Normal addressing :  $2 + X$
- Index addressing :  $3 + X$
- Indirect addressing :  $5 + X + 2Y$
- Indirect index addressing :  $6 + X + 2Y$

X is number of wait of the divisor area. Y is number of wait of the indirect address stored area. When X and Y are in odd address or in 8 bits bus area, double the value of X and Y.

**Table 9.6 Interrupt Sequence Execution Time**

Interrupt	Interrupt vector address	16 bits data bus	8 bits data bus
Peripheral I/O	Even address	14 cycles	16 cycles
	Odd address (Note 1)	16 cycles	16 cycles
INT instruction	Even address	12 cycles	14 cycles
	Odd address (Note 1)	14 cycles	14 cycles
NMI Watchdog timer Undefined instruction Address match	Even address (Note 2)	13 cycles	15 cycles
Overflow	Even address (Note 2)	14 cycles	16 cycles
BRK instruction (Variable vector table)	Even address	17 cycles	19 cycles
	Odd address (Note 1)	19 cycles	19 cycles
Single step BRK2 instruction BRK instruction (Fixed vector table)	Even address (Note 2)	19 cycles	21 cycles
High-speed interrupt (Note 3)	Vector table is internal register	5 cycles	

Note 1: Allocate interrupt vector addresses in even addresses, if possible.

Note 2: The vector table is fixed to even address.

Note 3: The high-speed interrupt is independent of these conditions.



### 9.13 Changes of IPL When Interrupt Request Acknowledged

When an interrupt request is acknowledged, the interrupt priority level of the acknowledged interrupt is set to the processor interrupt priority level (IPL).

If an interrupt request is acknowledged that does not have an interrupt priority level, the value shown in Table 9.7 is set to the IPL.

**Table 9.7 Relationship between Interrupts without Interrupt Priority Levels and IPL**

Interrupt sources without interrupt priority levels	Value that is set to IPL
Watchdog timer, $\overline{\text{NMI}}$	7
Reset	0
Other	Not changed

### 9.14 Saving Registers

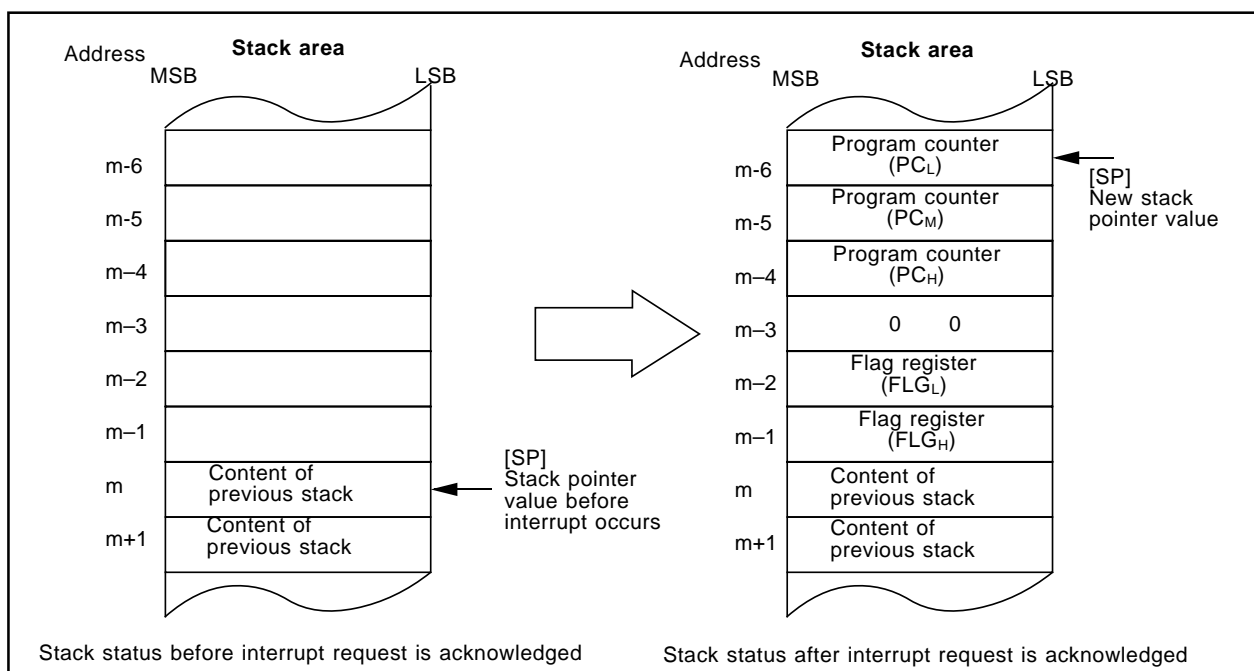
In an interrupt sequence, only the contents of the flag register (FLG) and program counter (PC) are saved to the stack area.

The order in which these contents are saved is as follows: First, the FLG register is saved to the stack area. Next, the 16 high-order bits and 16 low-order bits of the program counter expanded to 32-bit are saved. Figure 9.6 shows the stack status before an interrupt request is acknowledged and the stack status after an interrupt request is acknowledged.

In a high-speed interrupt sequence, the contents of the flag register (FLG) is saved to the flag save register (SVF) and program counter (PC) is saved to PC save register (SVP).

If there are any other registers you want to be saved, save them in software at the beginning of the interrupt routine. The PUSHM instruction allows you to save all registers except the stack pointer (SP) by a single instruction.

The execution speed is improved when register bank 1 is used with high speed interrupt register selected by not saving registers to the stack but to the switching register bank. In this case, switch register bank mode for high-speed interrupt routine.



**Figure 9.6 Stack status before and after an interrupt request is acknowledged**

### 9.15 Return from Interrupt Routine

As you execute the REIT instruction at the end of the interrupt routine, the contents of the flag register (FLG) and program counter (PC) that have been saved to the stack area immediately preceding the interrupt sequence are automatically restored. In high-speed interrupt, as you execute the FREIT instruction at the end of the interrupt routine, the contents of the flag register (FLG) and program counter (PC) that have been saved to the save registers immediately preceding the interrupt sequence are automatically restored.

Then control returns to the routine that was under execution before the interrupt request was acknowledged, and processing is resumed from where control left off.

If there are any registers you saved via software in the interrupt routine, be sure to restore them using an instruction (e.g., POPM instruction) before executing the REIT or FREIT instruction.

When switching the register bank before executing REIT and FREIT instruction, switched to the register bank immediately before the interrupt sequence.

### 9.16 Interrupt Priority

If two or more interrupt requests are sampled active at the same time, whichever interrupt request is acknowledged that has the highest priority.

Maskable interrupts (Peripheral I/O interrupts) can be assigned any desired priority by setting the interrupt priority level select bit accordingly. If some maskable interrupts are assigned the same priority level, the priority between these interrupts is resolved by the priority that is set in hardware.

Certain nonmaskable interrupts such as a reset (reset is given the highest priority) and watchdog timer interrupt have their priority levels set in hardware. Figure 9.7 lists the hardware priority levels of these interrupts.

Software interrupts are not subjected to interrupt priority. They always cause control to branch to an interrupt routine whenever the relevant instruction is executed.

### 9.17 Interrupt Resolution Circuit

Interrupt resolution circuit selects the highest priority interrupt when two or more interrupt requests are sampled active at the same time.

Figure 9.8 shows the interrupt resolution circuit.

**Reset >  $\overline{\text{NMI}}$  > Watchdog > Peripheral I/O > Single step > Address match**

Figure 9.7 Interrupt priority that is set in hardware

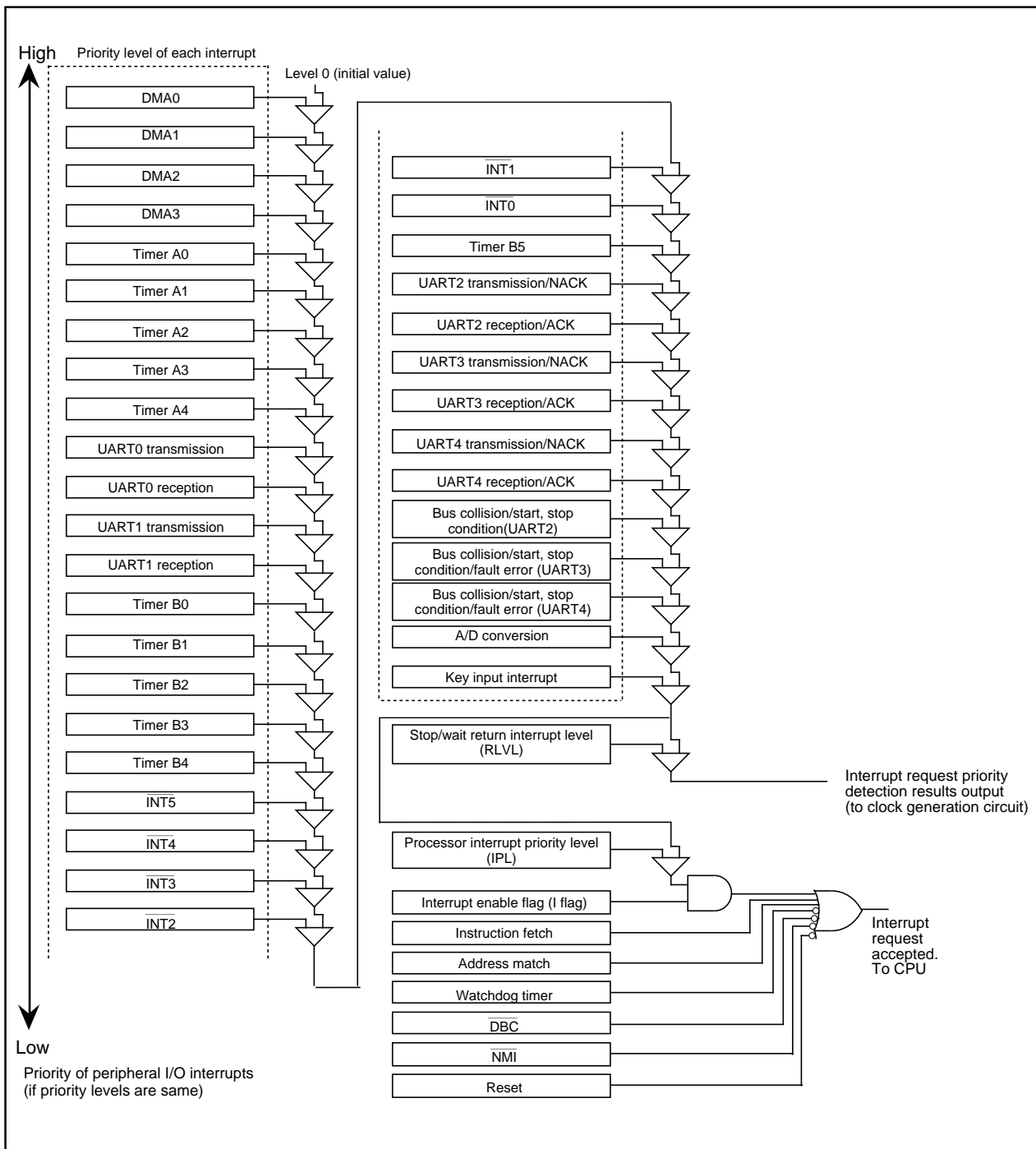


Figure 9.8 Interrupt resolution circuit

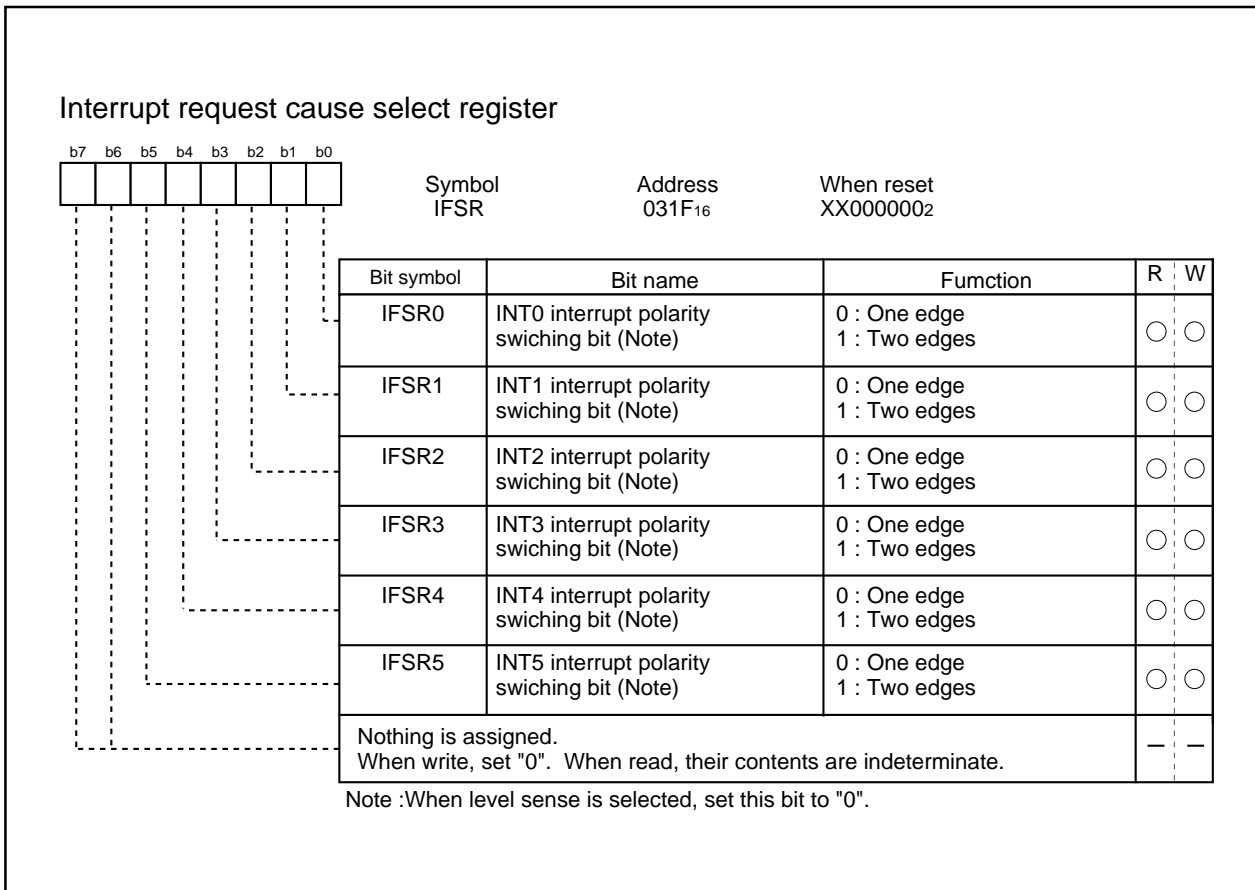
### 9.18 INT Interrupts

INT0 to INT5 are external input interrupts. The level sense/edge sense switching bits of the interrupt control register select the input signal level and edge at which the interrupt can be set to occur on input signal level and input signal edge. The polarity bit selects the polarity.

With the external interrupt input edge sense, the interrupt can be set to occur on both rising and falling edges by setting the INTi interrupt polarity switch bit of the interrupt request select register (address 031F16) to "1". When you select both edges, set the polarity switch bit of the corresponding interrupt control register to the falling edge ("0").

When you select level sense, the INTi interrupt polarity switch bit of the interrupt request select register (address 031F16) to "0".

Figure 9.9 shows the interrupt request select register.



**Figure 9.9** Interrupt request cause select register

### 9.19 $\overline{\text{NMI}}$ Interrupt

An  $\overline{\text{NMI}}$  interrupt is generated when the input to the P85/ $\overline{\text{NMI}}$  pin changes from “H” to “L”. The  $\overline{\text{NMI}}$  interrupt is a non-maskable external interrupt. The pin level can be checked in the port P85 register (bit 5 at address 03C416).

This pin cannot be used as a normal port input.

#### Notes:

When not intending to use the  $\overline{\text{NMI}}$  function, be sure to connect the  $\overline{\text{NMI}}$  pin to VCC (pulled-up). The  $\overline{\text{NMI}}$  interrupt is non-maskable. Because it cannot be disabled, the pin must be pulled up.

### 9.20 Key Input Interrupt

If the direction register of any of P104 to P107 is set for input and a falling edge is input to that port, a key input interrupt is generated. A key input interrupt can also be used as a key-on wakeup function for canceling the wait mode or stop mode. However, if you intend to use the key input interrupt, do not use P104 to P107 as A/D input ports. Figure 9.10 shows the block diagram of the key input interrupt. Note that if an “L” level is input to any pin that has not been disabled for input, inputs to the other pins are not detected as an interrupt.

Setting the key input interrupt disable bit (bit 7 at address 03AF16) to “1” disables key input interrupts from occurring regardless of the setting in the interrupt control register. When “1” is set in the key input interrupt disable register, there is no input via the port pin even when the direction register is set to input.

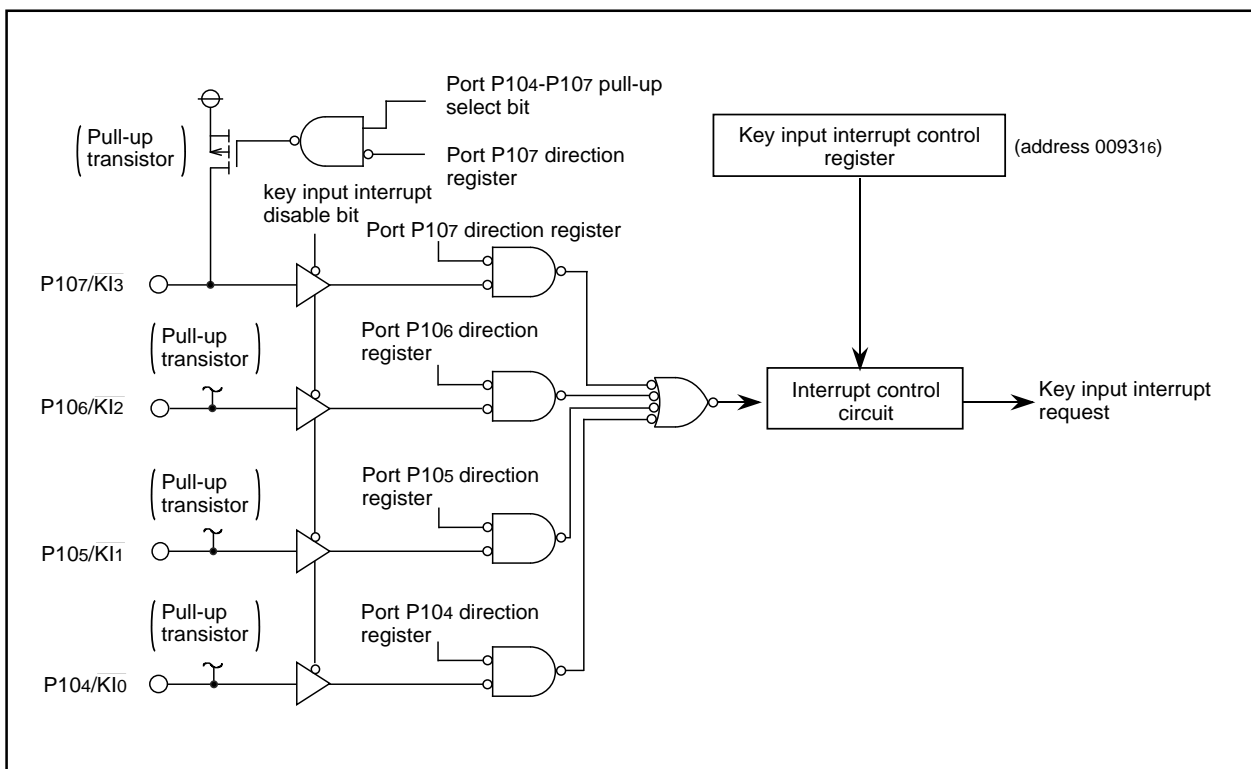


Figure 9.10 Block diagram of key input interrupt

### 9.21 Address Match Interrupt

An address match interrupt is generated when the address match interrupt address register contents match the program counter value. Four address match interrupts can be set, each of which can be enabled and disabled by an address match interrupt enable bit. Address match interrupts are not affected by the interrupt enable flag (I flag) and processor interrupt priority level (IPL).

Figure 9.11 shows the address match interrupt-related registers.

Set the start address of an instruction to the address match interrupt register.

Address match interrupt is not generated when address such as the middle of instruction or table data is set.

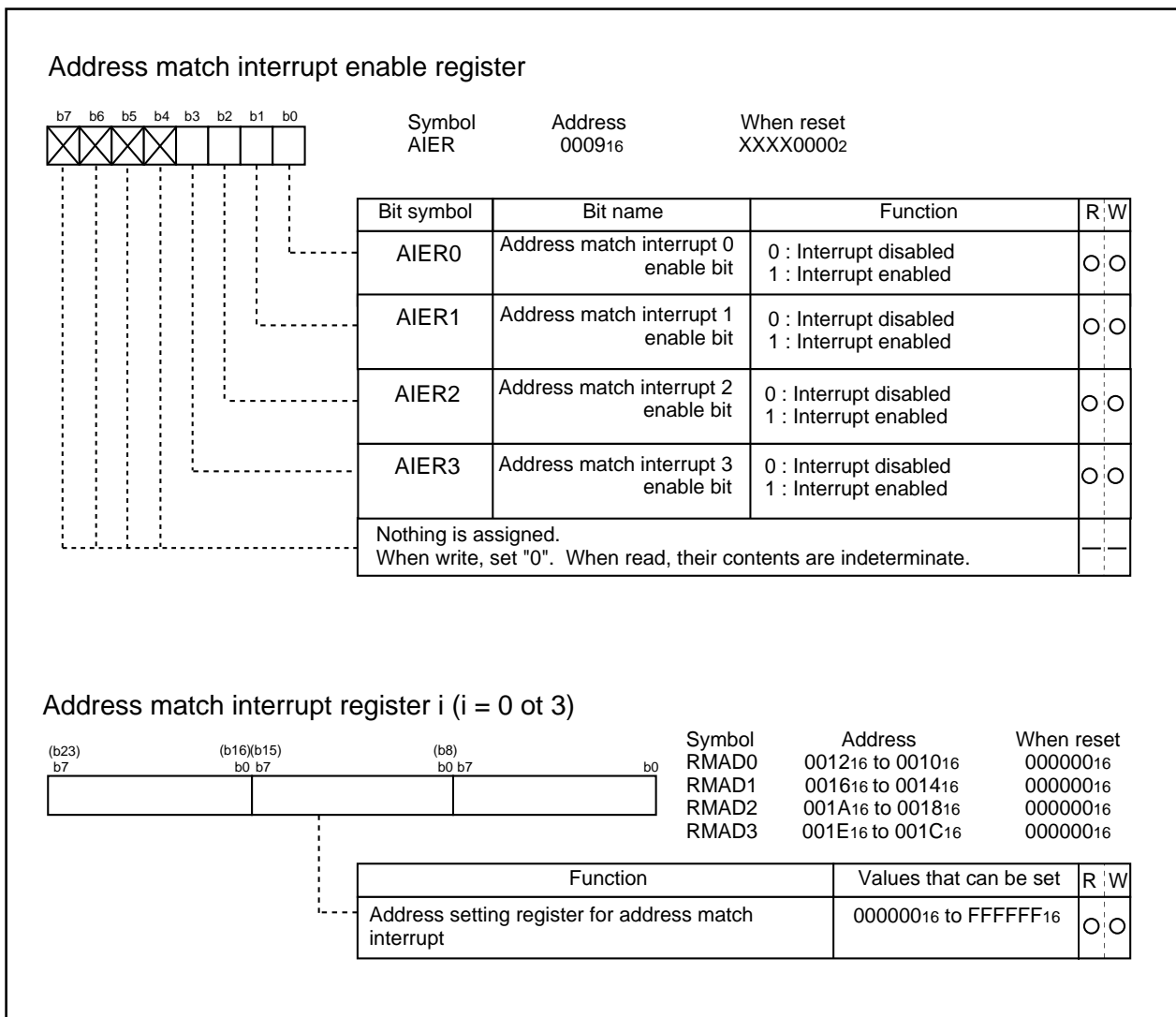


Figure 9.11 Address match interrupt-related registers

## 9.22 Precautions for Interrupts

### (1) Reading addresses 000000<sub>16</sub> and 000002<sub>16</sub>

- When maskable interrupt is occurred, CPU read the interrupt information (the interrupt number and interrupt request level) in the interrupt sequence from address 000000<sub>16</sub>. When high-speed interrupt is occurred, CPU read from address 000002<sub>16</sub>.

The interrupt request bit of the certain interrupt will then be set to "0".

However, reading addresses 000000<sub>16</sub> and 000002<sub>16</sub> by software does not set request bit to "0".

### (2) Setting the stack pointer

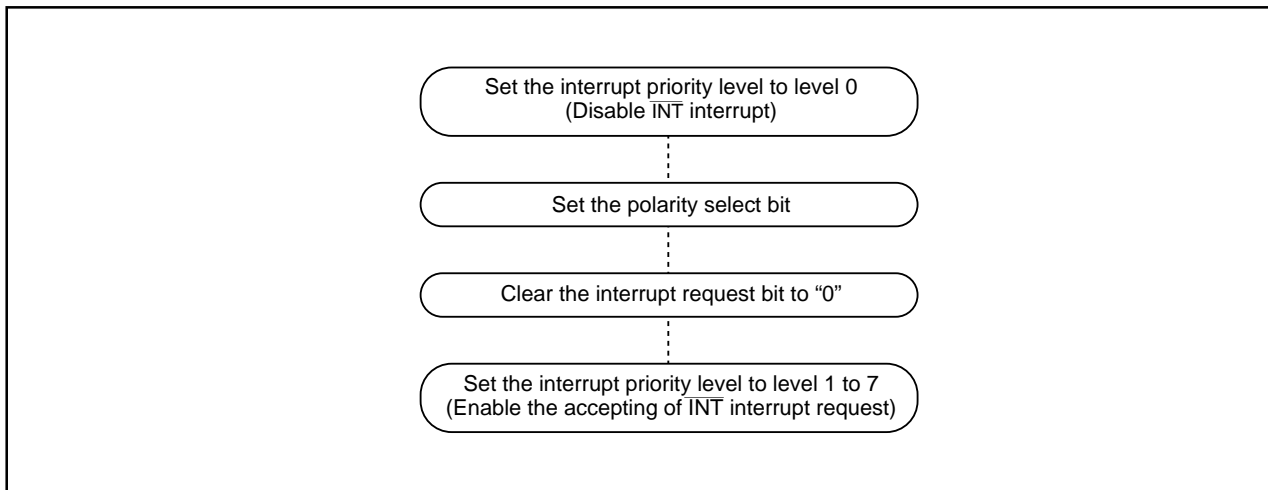
- The value of the stack pointer immediately after reset is initialized to 000000<sub>16</sub>. Accepting an interrupt before setting a value in the stack pointer may become a factor of runaway. Be sure to set a value in the stack pointer before accepting an interrupt. When using the  $\overline{\text{NMI}}$  interrupt, initialize the stack point at the beginning of a program. Any interrupt including the  $\overline{\text{NMI}}$  interrupt is generated immediately after executing the first instruction after reset. Set an even address to the stack pointer so that the operating efficiency of accessign memory is increased.

### (3) The $\overline{\text{NMI}}$ interrupt

- As for the  $\overline{\text{NMI}}$  interrupt pin, an interrupt cannot be disabled. Connect it to the Vcc pin via a resistance (pull-up) if unused. Be sure to work on it.
- The  $\overline{\text{NMI}}$  pin also serves as P85, which is exclusively input. Reading the contents of the P8 register allows reading the pin value. Use the reading of this pin only for establishing the pin level at the time when the  $\overline{\text{NMI}}$  interrupt is input.
- Signals input to the  $\overline{\text{NMI}}$  pin require "L" level and "H" level of 2 clock + 300ns or more, from the operation clock of CPU.

### (4) External interrupt

- Edge sense  
Either an "L" level or an "H" level of at least 250 ns width is necessary for the signal input to pins INT<sub>0</sub> to INT<sub>5</sub> regardless of the CPU operation clock.
- Level sense  
Either an "L" level or an "H" level of 1 cycle of BCLK + at least 200 ns width is necessary for the signal input to pins INT<sub>0</sub> to INT<sub>5</sub> regardless of the CPU operation clock. (When X<sub>IN</sub>=20MHz and no division mode, at least 250 ns width is necessary.)
- When the polarity of the INT<sub>0</sub> to INT<sub>5</sub> pins is changed, the interrupt request bit is sometimes set to "1". After changing the polarity, set the interrupt request bit to "0". Figure 9.12 shows the procedure for changing the  $\overline{\text{INT}}$  interrupt generate factor.



**Figure 9.12 Switching condition of INT interrupt request**

### (5) Rewrite the interrupt control register

- When a instruction to rewrite the interrupt control register is executed but the interrupt is disabled, the interrupt request bit is not set sometimes even if the interrupt request for that register has been generated. This will depend on the instruction. If this creates problems, use the below instructions to change the register.

Instructions : AND, OR, BCLR, BSET

- When attempting to clear the interrupt request bit of an interrupt control register, the interrupt request bit is not cleared sometimes. This will depend on the instruction. If this creates problems, use the below instructions to change the register.

Instructions : MOV

### (6) Address match interrupt

- Do not set the following addresses to the address match interrupt register.
  1. The address of the starting instruction in an interrupt routine.
  2. Any of the next 7 instructions addresses immediately after an instruction to clear an interrupt request bit of an interrupt control register or an instruction to rewrite an interrupt priority level to a smaller value.
  3. Any of the next 3 instructions addresses immediately after an instruction to set the interrupt enable flag (I flag).
  4. Any of the next 3 instructions addresses immediately after an instruction to rewrite a processor interrupt priority level (IPL) to a smaller value.



## Example 1)

```

Interrupt_A:                ; Interrupt A routine
    pushm R0,R1,R2,R3,A0,A1 ; <---- Do not set address match interrupt to the
    .....                  ; start address of an interrupt instruction

```

## Example 2)

```

mov.b #0,TA0IC             ; Change TA0 interrupt priority level to a smaller value
nop                        ; 1st instruction
nop                        ; 2nd instruction
nop                        ; 3rd instruction
nop                        ; 4th instruction
nop                        ; 5th instruction
nop                        ; 6th instruction
nop                        ; 7th instruction

```

} Do not set address match interrupt during this period

## Example 3)

```

fset I                    ; Set I flag ( interrupt enabled)
nop                        ; 1st instruction
nop                        ; 2nd instruction
nop                        ; 3rd instruction

```

} Do not set address match interrupt during this period

## Example 4)

```

ldipl #0                  ; Rewrite IPL to a smaller value
nop                        ; 1st instruction
nop                        ; 2nd instruction
nop                        ; 3rd instruction

```

} Do not set address match interrupt during this period

- To return from an interrupt to the address set in an address match interrupt register using return instruction (reit or freit)

To rewrite the interrupt control register within the interrupt routine, add the below processing to the end of the routine (immediately before the reit or freit instruction). Also, if multiple interrupts are enabled with other interrupts, add the below processing to the end of the interrupt that enables the multiple interrupts.

If the interrupt control register is being rewritten within the non-maskable interrupt routine, add the below processing to the end of all interrupts.

## Additional process

```

                                ; Execute after the register reset instruction (popm instruction)
fclr U                       ; Select ISP (Unnecessary if the ISP has been selected)
pushm R0                      ; Store R0 register
mov.w 6[SP],R0                ; Read FLG on stack (use "stc SVF,R0" when high-speed
                                ; interrupt)
ldc R0,FLG                    ; Set in FLG
popm R0                       ; Restore R0 register
nop                            ; Dummy
reit                          ; Interrupt completed (use freit when high-speed interrupt)

```

## Example 5)

If rewriting the interrupt control register for interrupt B with the interrupt A routine and enabling multiple interrupts with interrupt C, the above processing is required at the end of the interrupt A and interrupt C routines.

## Interrupt A routine

Interrupt\_A:

```

pushm R0,R1,R2,R3,A0,A1 ; Store registers
....
bclr 3,TA0IC ; Rewrite interrupt control register of interrupt B
....
popm R0,R1,R2,R3,A0,A1 ; Restore registers
fclr U ; Select ISP (Unnecessary if the ISP has been selected)
pushm R0 ; Store R0 register
mov.w 6[SP],R0 ; Read FLG on stack
ldc R0,FLG ; Set in FLG
popm R0 ; Restore R0 register
nop ; Dummy
reit ; Interrupt completed

```

## Interrupt C routine

Interrupt\_C:

```

pushm R0,R1,R2,R3,A0,A1 ; Store registers
fset I ; Multiple interrupt enabled
....
....
popm R0,R1,R2,R3,A0,A1 ; Restore registers
fclr U ; Select ISP (Unnecessary if the ISP has been selected)
pushm R0 ; Store R0 register
mov.w 6[SP],R0 ; Read FLG on stack
ldc R0,FLG ; Set in FLG
popm R0 ; Restore R0 register
nop ; Dummy
reit ; Interrupt completed

```

## 10. Watchdog Timer

The watchdog timer has the function of detecting when the program is out of control. Therefore, we recommend using the watchdog timer to improve reliability of a system. The watchdog timer is a 15-bit counter which down-counts the clock derived by dividing the BCLK using the prescaler. Whether a watchdog timer interrupt is generated or reset is selected when an underflow occurs in the watchdog timer. Watchdog timer interrupt is selected when bit 6 of the system control register 0 (address 000816 :CM06) is "0" and reset is selected when CM06 is "1". No value other than "1" can be written in CM06. Once when reset is selected (CM06="1"), watchdog timer interrupt cannot be selected by software.

When XIN is selected for the BCLK, bit 7 of the watchdog timer control register (address 000F16) selects the prescaler division ratio (by 16 or by 128). When XCIN is selected as the BCLK, the prescaler is set for division by 2 regardless of bit 7 of the watchdog timer control register (address 000F16). Therefore, the watchdog timer cycle can be calculated as follows. However, errors can arise in the watchdog timer cycle due to the prescaler.

When XIN is selected in BCLK

$$\text{Watchdog timer cycle} = \frac{\text{Prescaler division ratio (16 or 128)} \times \text{watchdog timer count (32768)}}{\text{BCLK}}$$

When XCIN is selected in BCLK

$$\text{Watchdog timer cycle} = \frac{\text{Prescaler division ratio (2)} \times \text{watchdog timer count (32768)}}{\text{BCLK}}$$

For example, when BCLK is 20MHz and the prescaler division ratio is set to 16, the monitor timer cycle is approximately 26.2 ms.

The watchdog timer is initialized by writing to the watchdog timer start register (address 000E16) and when a watchdog timer interrupt request is generated. The prescaler is initialized only when the microcomputer is reset. After a reset is cancelled, the watchdog timer and prescaler are both stopped. The count is started by writing to the watchdog timer start register (address 000E16). CM06 is initialized only at reset. After reset, watchdog timer interrupt is selected.

The watchdog timer and the prescaler stop in stop mode, wait mode and hold status. After exiting these modes and status, counting starts from the value remained before.

In the stop mode, wait mode and hold state, 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.

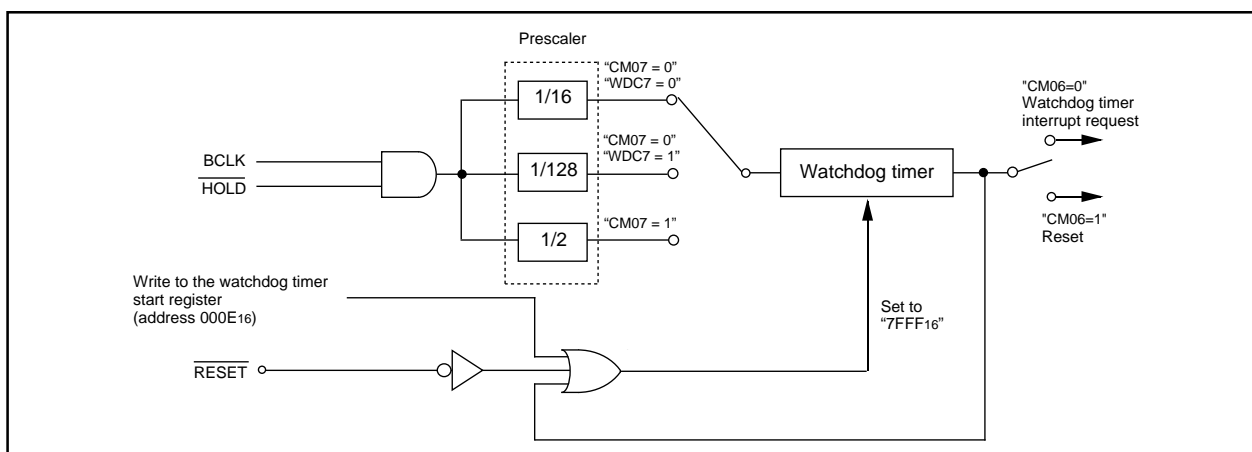


Figure 10.1 Block diagram of watchdog timer

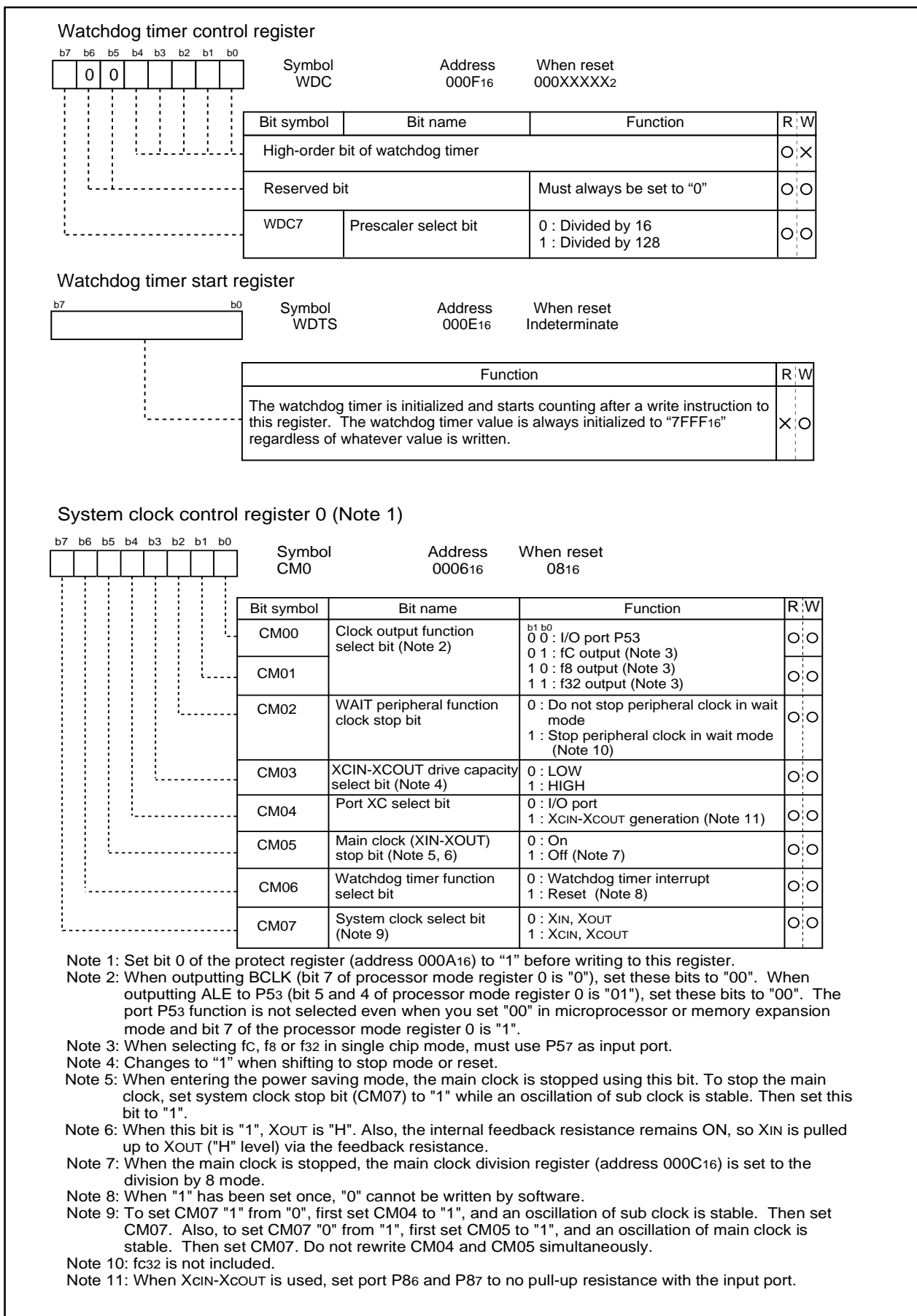


Figure 10.2 Watchdog timer control and start registers

### 11. DMAC

This microcomputer has four DMAC (direct memory access controller) channels that allow data to be sent to memory without using the CPU. DMAC is a function that to transmit 1 data of a source address (8 bits / 16 bits) to a destination address when transmission request occurs. When using three or more DMAC channels, the register bank 1 register and high-speed interrupt register are used as DMAC registers. If you are using three or more DMAC channels, you cannot, therefore, use high-speed interrupts. The CPU and DMAC use the same data bus, but the DMAC has a higher bus access privilege than the CPU, and because of the use of cycle-stealing, operations are performed at high-speed from the occurrence of a transfer request until one word (16 bits) or 1 byte (8 bits) of data have been sent. Figure 11.1 shows the mapping of registers used by the DMAC. Table 11.1 shows DMAC specifications. Figures 11.2 to 11.5 show the structures of the registers used.

As the registers shown in Figure 11.1 is allocated in CPU, use LDC instruction when writing. When writing to DCT2, DCT3, DRC2, DRC3, DMA2 and DMA3, set register bank select flag (B flag) to "1" and use MOV instruction to set R0 to R3, A0 and A1 registers. When writing to DSA2 and DSA3, set register bank select flag (B flag) to "1" and use LDC instruction to set SB and FB registers.

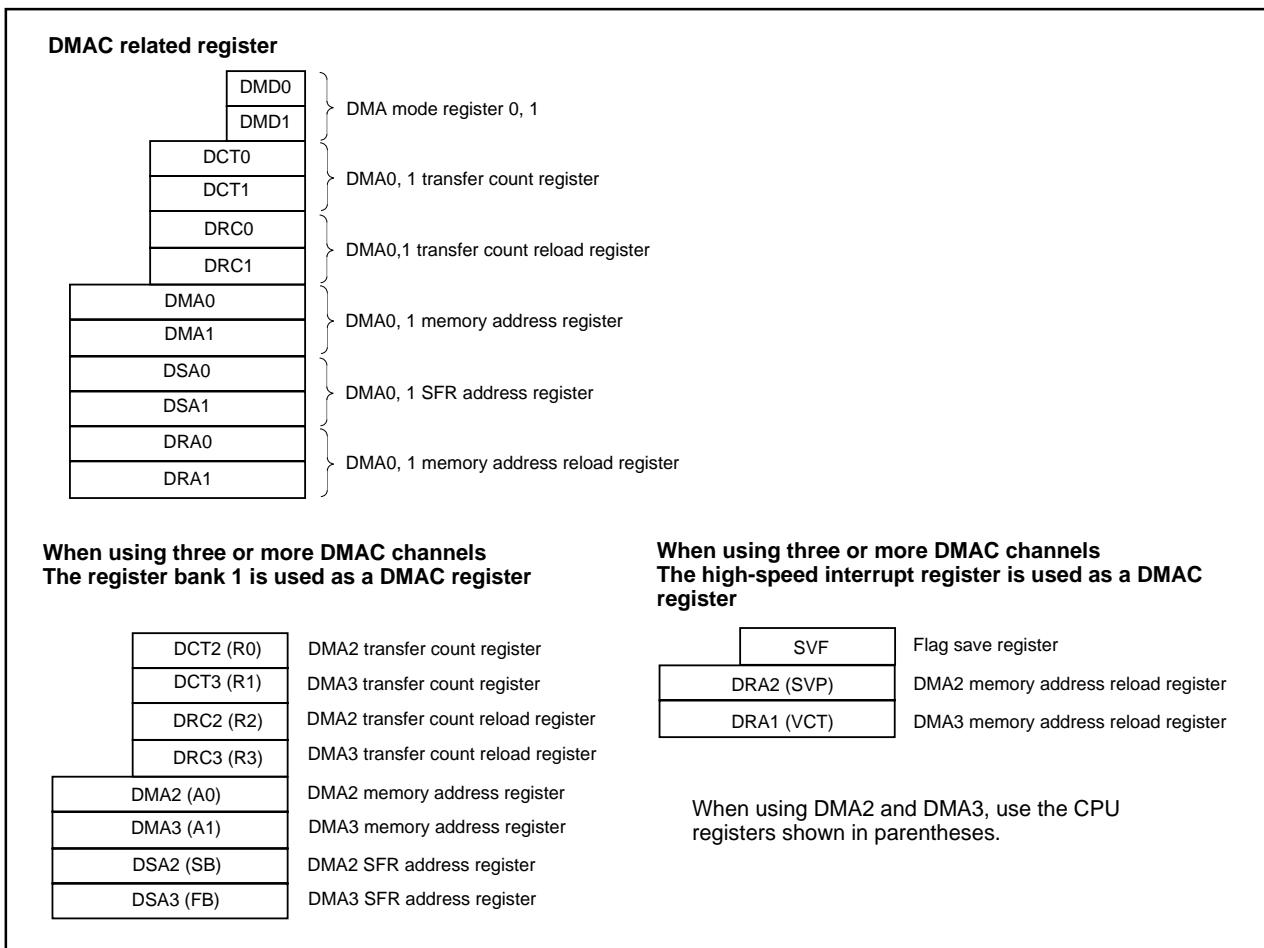


Figure 11.1 Register map using DMAC

In addition to writing to the software DMA request bit to start DMAC transfer, the interrupt request signals output from the functions specified in the DMA request factor select bits are also used. However, in contrast to the interrupt requests, repeated DMA requests can be received, regardless of the interrupt flag. (Note, however, that the number of actual transfers may not match the number of transfer requests if the DMA request cycle is shorter than the DMR transfer cycle. For details, see the description of the DMAC request bit.)

**Table 11.1 DMAC specifications**

Item	Specification
No. of channels	4 (cycle steal method)
Transfer memory space	<ul style="list-style-type: none"> <li>From any address in the 16 Mbytes space to a fixed address (16 Mbytes space)</li> <li>From a fixed address (16 Mbytes space) to any address in the 16 Mbytes space</li> </ul>
Maximum No. of bytes transferred	128 Kbytes (with 16-bit transfers) or 64 Kbytes (with 8-bit transfers)
DMA request factors (Note)	Falling edge of $\overline{INT0}$ to $\overline{INT3}$ or both edge Timer A0 to timer A4 interrupt requests Timer B0 to timer B5 interrupt requests UART0 to UART4 transmission and reception interrupt requests A/D conversion interrupt requests Software triggers
Channel priority	DMA0 > DMA1 > DMA2 > DMA3 (DMA0 is the first priority)
Transfer unit	8 bits or 16 bits
Transfer address direction	forward/fixed (forward direction cannot be specified for both source and destination simultaneously)
Transfer mode	<ul style="list-style-type: none"> <li>Single transfer Transfer ends when the transfer count register is "0000<sub>16</sub>".</li> <li>Repeat transfer When the transfer counter is "0000<sub>16</sub>", the value in the transfer counter reload register is reloaded into the transfer counter and the DMA transfer is continued</li> </ul>
DMA interrupt request generation timing	When the transfer counter register changes from "0001 <sub>16</sub> " to "0000 <sub>16</sub> ".
DMA startup	<ul style="list-style-type: none"> <li>Single transfer Transfer starts when DMA transfer count register is more than "0001<sub>16</sub>" and the DMA is requested after "012" is written to the channel i transfer mode select bits</li> <li>Repeat transfer Transfer starts when the DMA is requested after "112" is written to the channel i transfer mode select bits</li> </ul>
DMA shutdown	<ul style="list-style-type: none"> <li>Single transfer When "002" is written to the channel i transfer mode select bits and DMA transfer count register becomes "0000<sub>16</sub>" by DMA transfer or write</li> <li>Repeat transfer When "002" is written to the channel i transfer mode select bits</li> </ul>
Reload timing	When the transfer counter register changes from "0001 <sub>16</sub> " to "0000 <sub>16</sub> " in repeat transfer mode.
Reading / writing the register	Registers are always read/write enabled.
Number of DMA transfer cycles	Between SFR and internal RAM : 3 cycles Between external I/O and external memory : minimum 3 cycles

Note: DMA transfer is not effective to any interrupt.

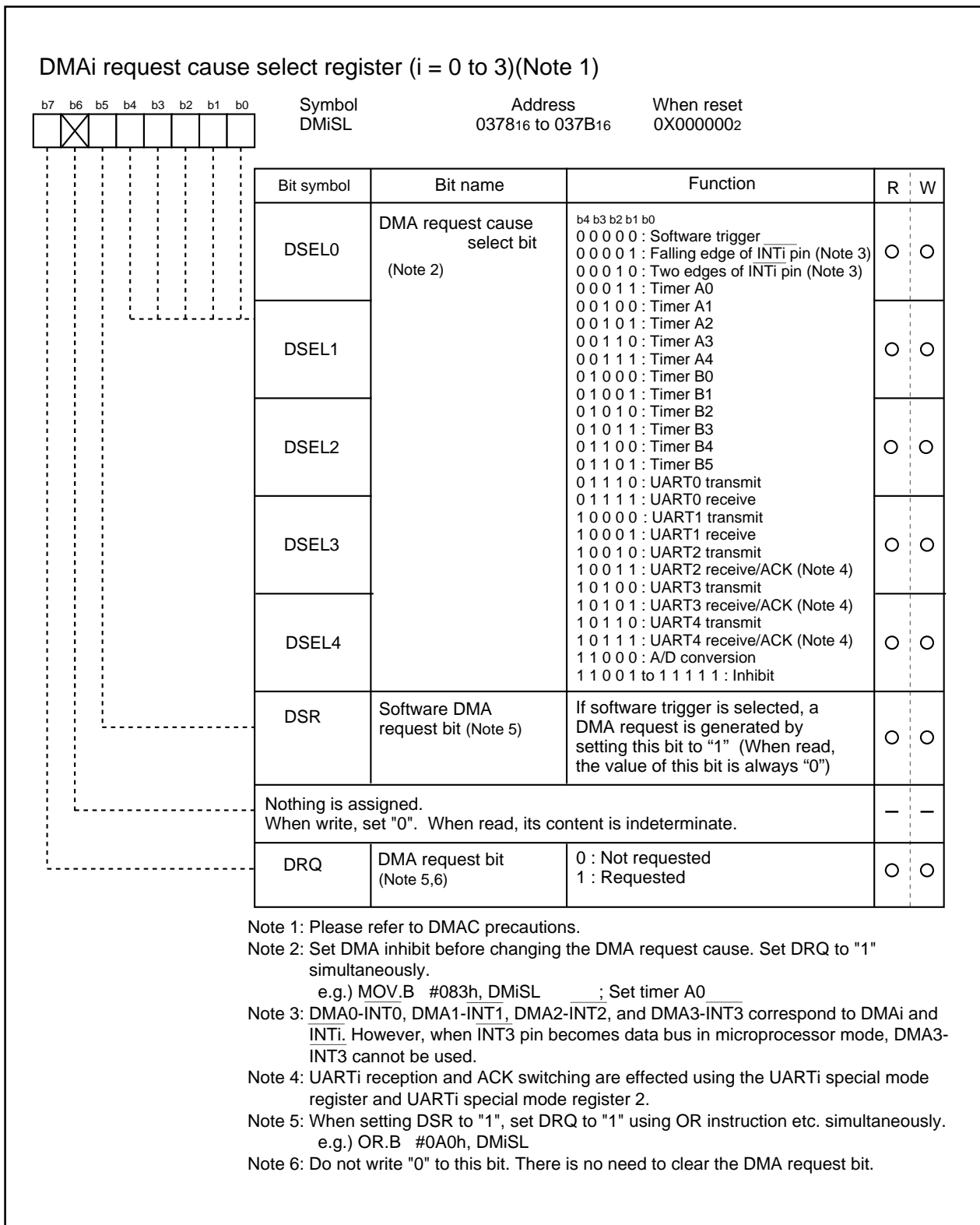


Figure 11.2 DMAC register (1)

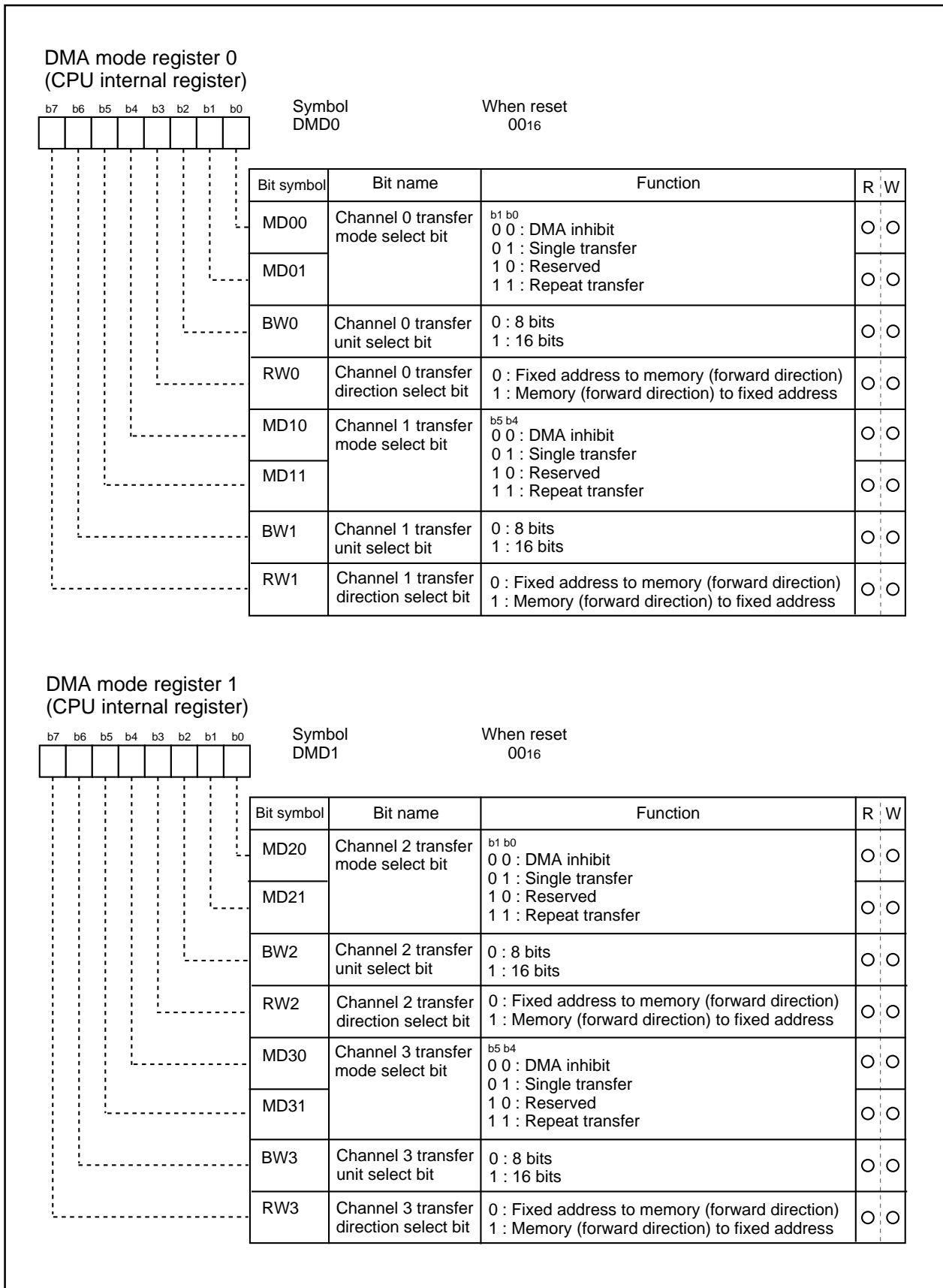


Figure 11.3 DMAC register (2)



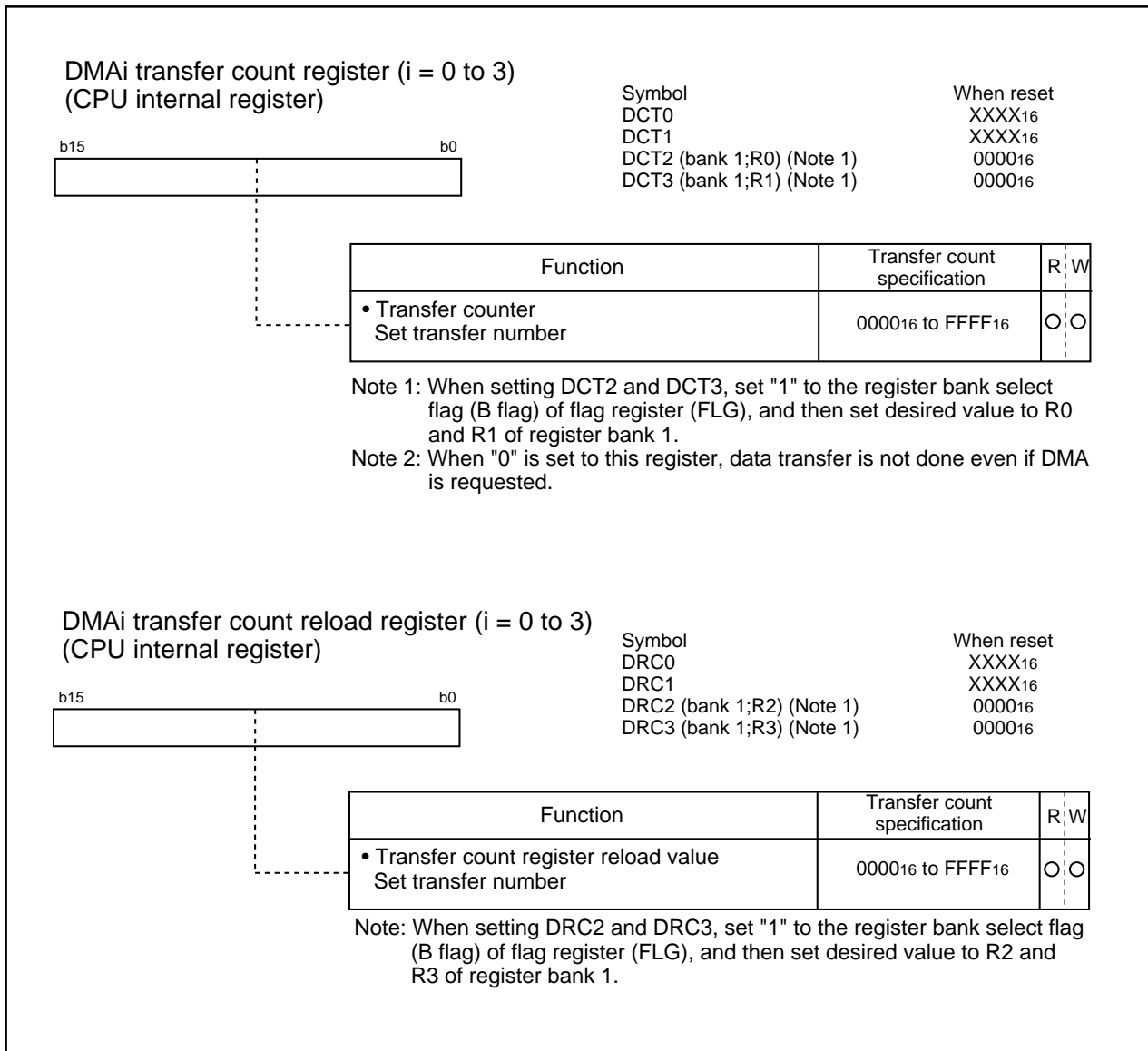


Figure 11.4 DMAC register (3)

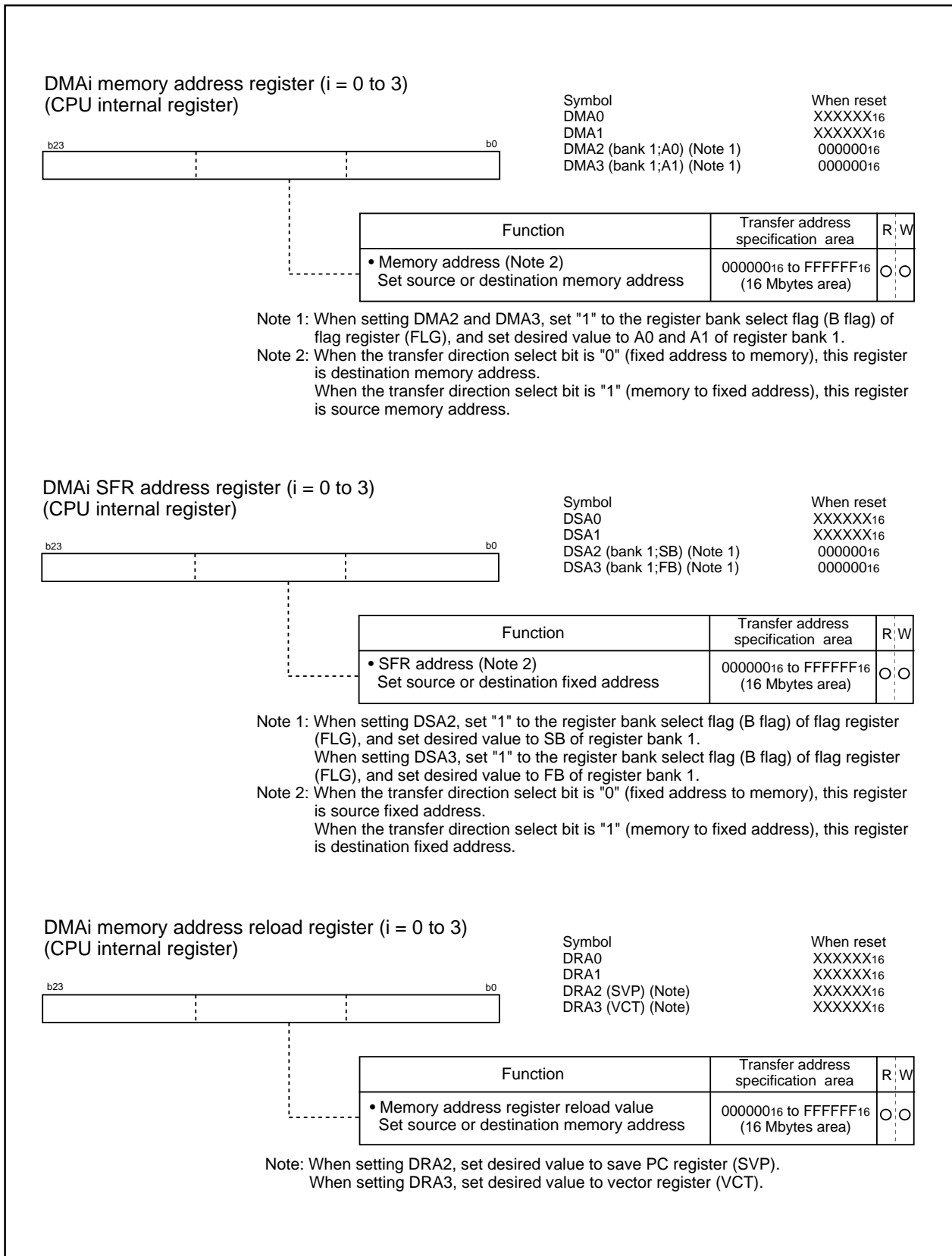


Figure 11.5 DMAC register (4)

## (1) Transfer cycle

The transfer cycle consists of the bus cycle in which data is read from memory or from the SFR area (source read) and the bus cycle in which the data is written to memory or to the SFR area (destination write). The number of read and write bus cycles depends on the source and destination addresses. In memory expansion mode and microprocessor mode, the number of read and write bus cycles also depends on the level of the BYTE pin. Also, the bus cycle itself is longer when software waits are inserted.

### (a) Effect of source and destination addresses

When 16-bit data is transferred on a 16-bit data bus, and the source and destination both start at odd addresses, there are one more source read cycle and destination write cycle than when the source and destination both start at even addresses.

### (b) Effect of external data bus width control register

When in memory expansion mode or microprocessor mode, the transfer cycle changes according to the data bus width at the source and destination.

1. When transferring 16 bits of data and the data bus width at the source and at the destination is 8 bits (data bus width bit = "0"), there are two 8-bit data transfers. Therefore, two bus cycles are required for reading and two cycles for writing.
2. When transferring 16 bits of data and the data bus width at the source is 8 bits (data bus width bit = "0") and the data bus width at the destination is 16 bits (data bus width bit = "1"), the data is read in two 8-bit blocks and written as 16-bit data. Therefore, two bus cycles are required for reading and one cycle for writing.
3. When transferring 16 bits of data and the data bus width at the source is 16 bits (data bus width bit = "1") and the data bus width at the destination is 8 bits (data bus width bit = "0"), 16 bits of data are read and written as two 8-bit blocks. Therefore, one bus cycle is required for reading and two cycles for writing.

### (c) Effect of software wait

When the SFR area or a memory area with a software wait is accessed, the number of cycles is increased for the wait by 1 bus cycle. The length of the cycle is determined by BCLK.

Figure 11.6 shows the example of the transfer cycles for a source read. Figure 11.6 shows the destination is external area, the destination write cycle is shown as two cycle (one bus cycle) and the source read cycles for the different conditions. 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 the transfer cycle, remember to apply the respective conditions to both the destination write cycle and the source read cycle. For example (2) in Figure 11.6, if data is being transferred in 16-bit units on an 8-bit bus, two bus cycles are required for both the source read cycle and the destination write cycle.

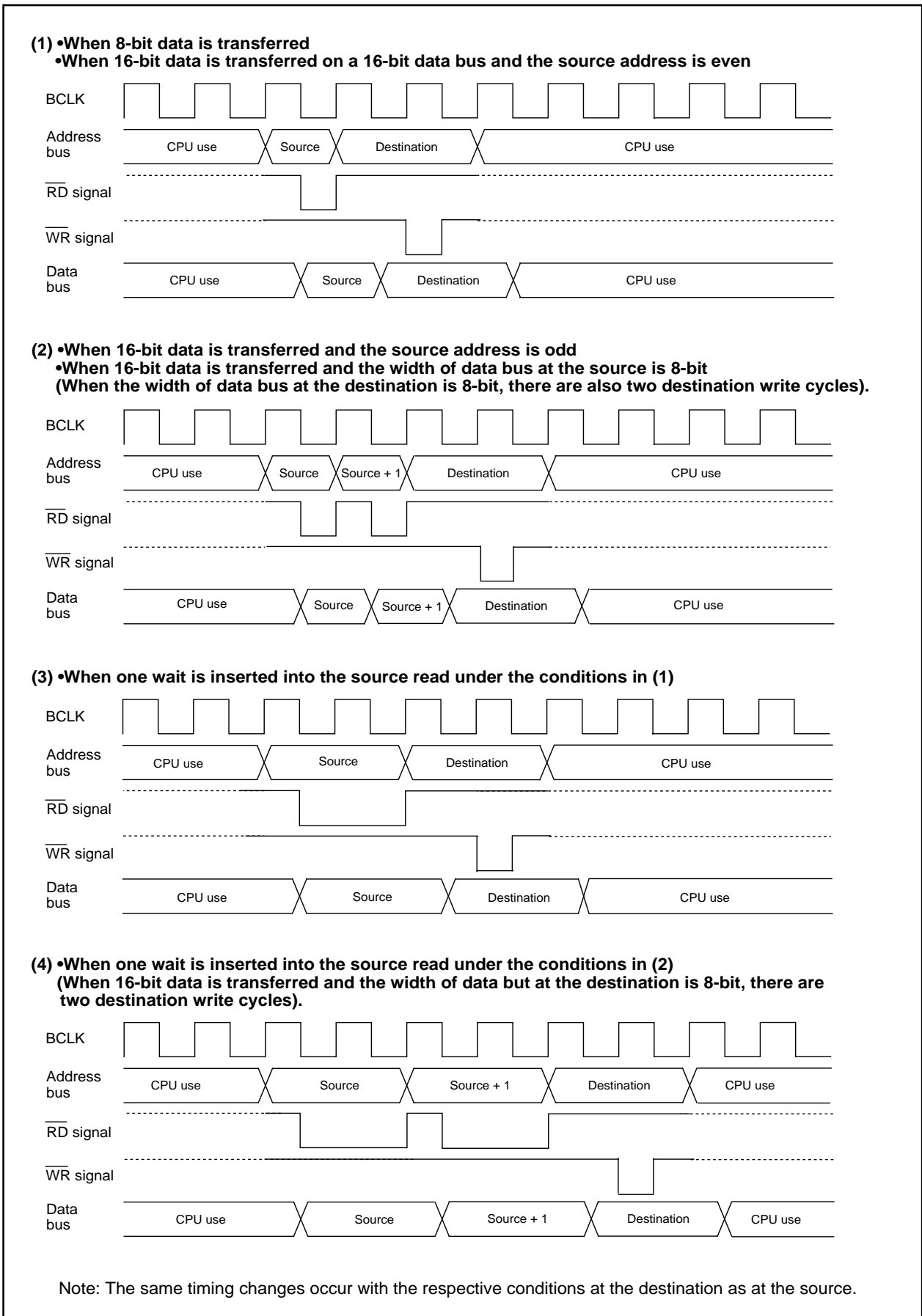


Figure 11.6 Example of the transfer cycles for a source read

## (2) DMAC transfer cycles

Any combination of even or odd transfer read and write addresses is possible. Table 11.2 shows the number of DMAC transfer cycles.

The number of DMAC transfer cycles can be calculated as follows:

$$\text{No. of transfer cycles per transfer unit} = \text{No. of read cycles} \times j + \text{No. of write cycles} \times k$$

**Table 11.2 No. of DMAC transfer cycles**

Transfer unit	Bus width	Access address	Single-chip mode		Memory expansion mode Microprocessor mode	
			No. of read cycles	No. of write cycles	No. of read cycles	No. of write cycles
8-bit transfers (BWi = "0")	16-bit (DSi = "1")	Even	1	1	1	1
		Odd	1	1	1	1
	8-bit (DSi = "0")	Even	—	—	1	1
		Odd	—	—	1	1
16-bit transfers (BWi = "1")	16-bit (DSi = "1")	Even	1	1	1	1
		Odd	2	2	2	2
	8-bit (DSi = "0")	Even	—	—	2	2
		Odd	—	—	2	2

### Coefficient j, k

Internal Memory			External Memory					
Internal ROM/RAM		SFR area	Separate bus				Multiplex bus	
No wait	With wait		No wait	One wait	Two waits	Three waits	Two waits	Three waits
j=1	j=2	j=2	j=1	j=2	j=3	j=4	j=3	j=4
k=1	k=2	k=2	k=2	k=2	k=3	k=4	k=3	k=4

## DMA Request Bit

The DMAC can issue DMA requests using preselected DMA request factors for each channel as triggers.

The DMA transfer request factors include the reception of DMA request signals from the internal peripheral functions, software DMA factors generated by the program, and external factors using input from external interrupt signals.

See the description of the DMA<sub>i</sub> factor selection register for details of how to select DMA request factors. DMA requests are received as DMA requests when the DMA<sub>i</sub> request bit is set to "1" and the channel *i* transfer mode select bits are "01" or "11". Therefore, even if the DMA<sub>i</sub> request bit is "1", no DMA request is received if the channel *i* transfer mode select bit is "00". In this case, DMA<sub>i</sub> request bit is cleared. Because the channel *i* transfer mode select bits default to "00" after a reset, remember to set the channel *i* transfer mode select bit for the channel to be activated after setting the DMAC related registers. This enables receipt of the DMA requests for that channel, and DMA transfers are then performed when the DMA<sub>i</sub> request bit is set.

The following describes when the DMA<sub>i</sub> request bit is set and cleared.

**(1) Internal factors**

The DMA<sub>i</sub> request flag is set to "1" in response to internal factors at the same time as the interrupt request bit of the interrupt control register for each factor is set. This is because, except for software trigger DMA factors, they use the interrupt request signals output by each function.

The DMA<sub>i</sub> request bit is cleared to "0" when the DMA transfer starts or the DMA transfer is in disable state (channel *i* transfer mode select bits are "00" and the DMA<sub>i</sub> transfer count register is "0").

**(2) External factors**

These are DMA request factors that are generated by the input edge from the  $\overline{\text{INT}}_i$  pin (where *i* indicates the DMAC channel). When the  $\overline{\text{INT}}_i$  pin is selected by the DMA<sub>i</sub> request factor select bit as an external factor, the inputs from these pins become the DMA request signals.

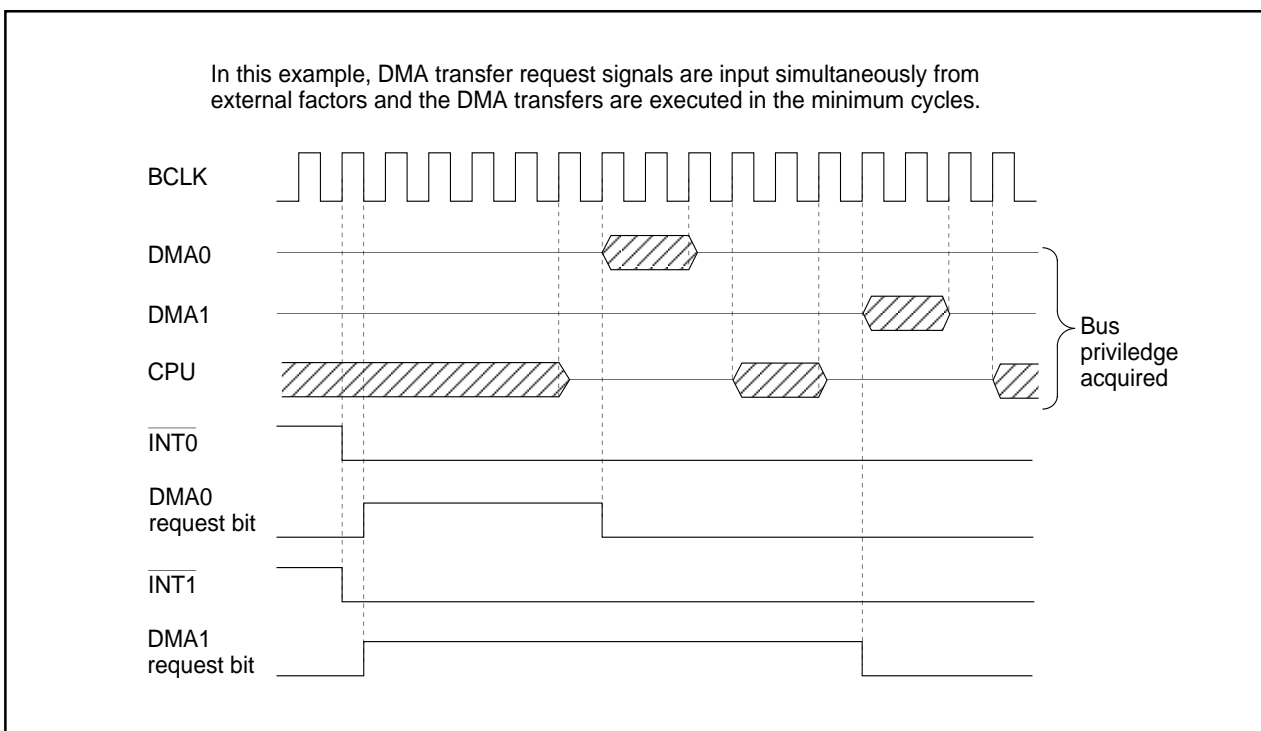
When an external factor is selected, the DMA<sub>i</sub> request bit is set, according to the function specified in the DMA request factor select bit, on either the falling edge of the signal input via the  $\overline{\text{INT}}_i$  pins, or both edges. When an external factor is selected, the DMA<sub>i</sub> request bit is cleared, in the same way as the DMA<sub>i</sub> request bit is cleared for internal factors, when the DMA transfer starts or the DMA transfer is in disable state.

**(3) Relationship between external factor request input and DMA<sub>i</sub> request flag, and DMA transfer timing**

When the request inputs to DMA<sub>i</sub> occur in the same sampling cycle (between the falling edge of BCLK and the next falling edge), the DMA<sub>i</sub> request bits are set simultaneously, but if the DMA<sub>i</sub> enable bits are all set, DMA0 takes priority and the transfer starts. When one transfer unit is complete, the bus privilege is returned to the CPU. When the CPU has completed one bus access, DMA1 transfer starts, and, when one transfer unit is complete, the privilege is again returned to the CPU.

The priority is as follows: DMA0 > DMA1 > DMA2 > DMA3.

Figure 11.7. DMA transfer example by external factors shows what happens when DMA0 and DMA1 requests occur in the same sampling cycle.



**Figure 11.7 DMA transfer example by external factors**

## Precautions for DMAC

(1) Do not clear the DMA request bit of the DMAi request cause select register.

In M16C/80, when a DMA request is generated while the channel is disabled (Note), the DMA transfer is not executed and the DMA request bit is cleared automatically.

Note :The DMA is disabled or the transfer count register is "0".

(2) When DMA transfer is done by a software trigger, set DSR and DRQ of the DMAi request cause select register to "1" simultaneously using the OR instruction.

e.g.) OR.B #0A0h, DMiSL ; DMiSL is DMAi request cause select register

(3) When changing the DMAi request cause select bit of the DMAi request cause select register, set "1" to the DMA request bit, simultaneously. In this case, set the corresponding DMA channel to disabled before changing the DMAi request cause select bit. At least 26 cycles are needed from the instruction to write to the DMAi request cause select register to enable DMA.

Example) When DMA request cause is changed to timer A0 and using DMA0 in single transfer after DMA initial setting

```

push.w    R0                ; Store R0 register
stc       DMD0, R0          ; Read DMA mode register 0
and.b     #11111100b, R0L    ; Clear DMA0 transfer mode select bit to "00"
ldc       R0, DMD0          ; DMA0 disabled
mov.b     #10000011b, DM0SL ; Select timer A0
                                ; (Write "1" to DMA request bit simultaneously)

push.w    R0                ; Store R0 register
mov.w     #6,R0              ;
dummy_loop:
sbjnz.w   #1,R0,dummy_loop  ; Dummy cycle
pop.w     R0                 ; Restore R0 register
or.b      #00000001b, R0L    ; Set DMA0 single transfer
ldc       R0, DMD0          ; DMA0 enabled
pop.w     R0                 ; Restore R0 register

```

} **At least 26 cycles are needed until DMA enabled.**

(4) Recommended procedure for starting DMA transfer

- When writing to the DMAi request cause register including overwriting the same value to the DMAi request cause register;
  1. Disable the corresponding channel i DMA in DMA mode registers 0 and 1.
  2. Set up the peripheral used as the source of the DMA transfer. However, the peripheral should remain disabled at this time. For example, when using UART0 transmit, disable UART0 transmit.
  3. Set the DMAi request cause select register. At this time, write a '1' to the DMA request bit (bit 7)

4. Set the following SFR registers:
  - DMAiSFR address register
  - DMAI memory address reload register
  - DMAi memory address register
  - DMAi transfer count reload register
  - DMAi transfer count register
5. At this point, if the number of elapsed cycles are less than 26, add code (NOP's or other processing) to make up some time.
6. Enable the corresponding channel i DMA in the DMA mode registers 0 and 1.
7. Enable the peripheral used as the source of the DMA transfer. For example, when using UART0 transmit, enable UART0 transmit.

•When not writing to the DMAi request cause register;

1. Disable the corresponding channel i DMA in the DMA mode registers 0 and 1.
2. Set up the peripheral used as the source of the DMA transfer. However, the peripheral should remain disabled at this time. For example, when using UART0 transmit, disable UART0 transmit.
3. Set up the following SFR registers:
  - DMAiSFR address register
  - DMAI memory address reload register
  - DMAi memory address register
  - DMAi transfer count reload register
  - DMAi transfer count register
4. Enable the corresponding channel i DMA in the DMA mode registers 0 and 1.
5. Enable the peripheral used as the source of the DMA transfer. For example, when using UART0 transmit, enable UART0 transmit.

(5) Recommended procedure after completing DMA transfer

- Disable the peripheral used as source of the DMA transfer to prevent generating a DMA request.
- Disable the corresponding channel i DMA in the DMA mode registers 0 and 1.



### 12. Timer

There are eleven 16-bit timers. These timers can be classified by function into timers A (five) and timers B (six). All these timers function independently. Count source for each timer becomes an operation clock for timer operation as counting and reloading, etc. Figures 12.1 and 12.2 show the block diagram of timers.

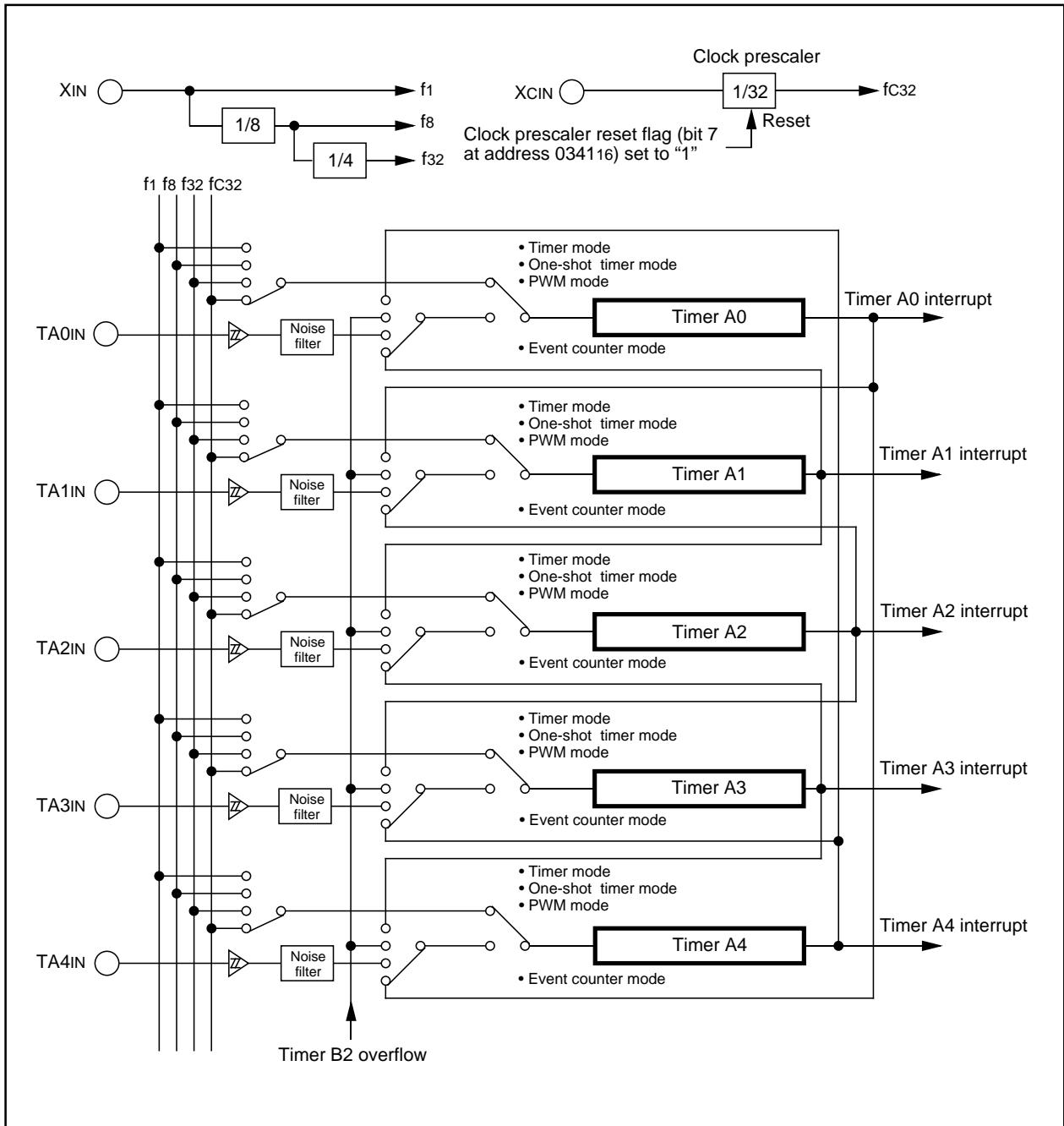


Figure 12.1 Timer A block diagram

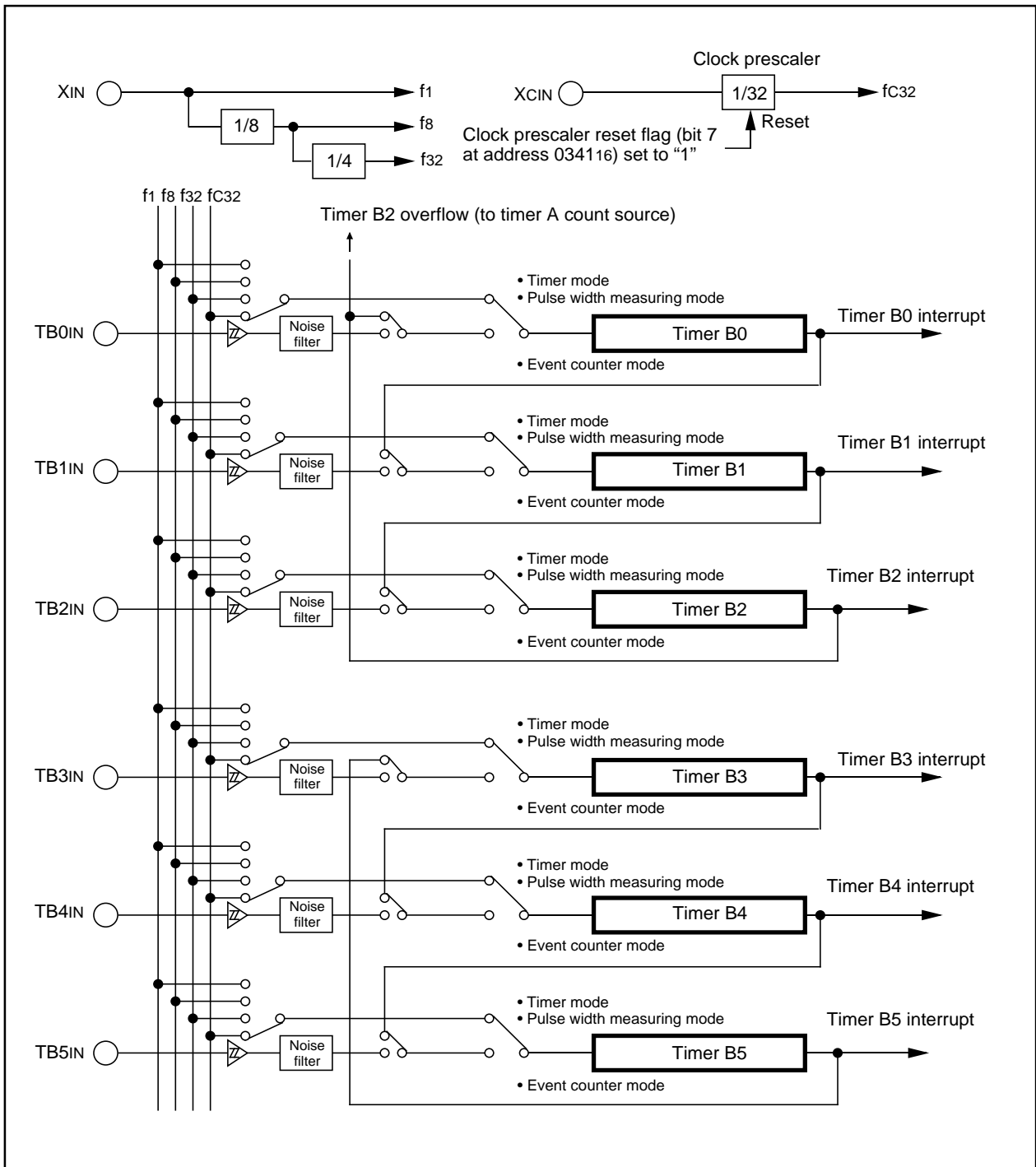


Figure 12.2 Timer B block diagram

### 13. Timer A

Figure 13.1 shows the block diagram of timer A. Figures 13.2 to 13.4 show the timer A-related registers. Except in event counter mode, timers A0 through A4 all have the same function. Use the timer Ai mode register (i = 0 to 4) bits 0 and 1 to choose the desired mode.

Timer A has the four operation modes listed as follows:

- Timer mode: The timer counts an internal count source.
- Event counter mode: The timer counts pulses from an external source or a timer over flow.
- One-shot timer mode: The timer stops counting when the count reaches “0000<sub>16</sub>”.
- Pulse width modulation (PWM) mode: The timer outputs pulses of a given width.

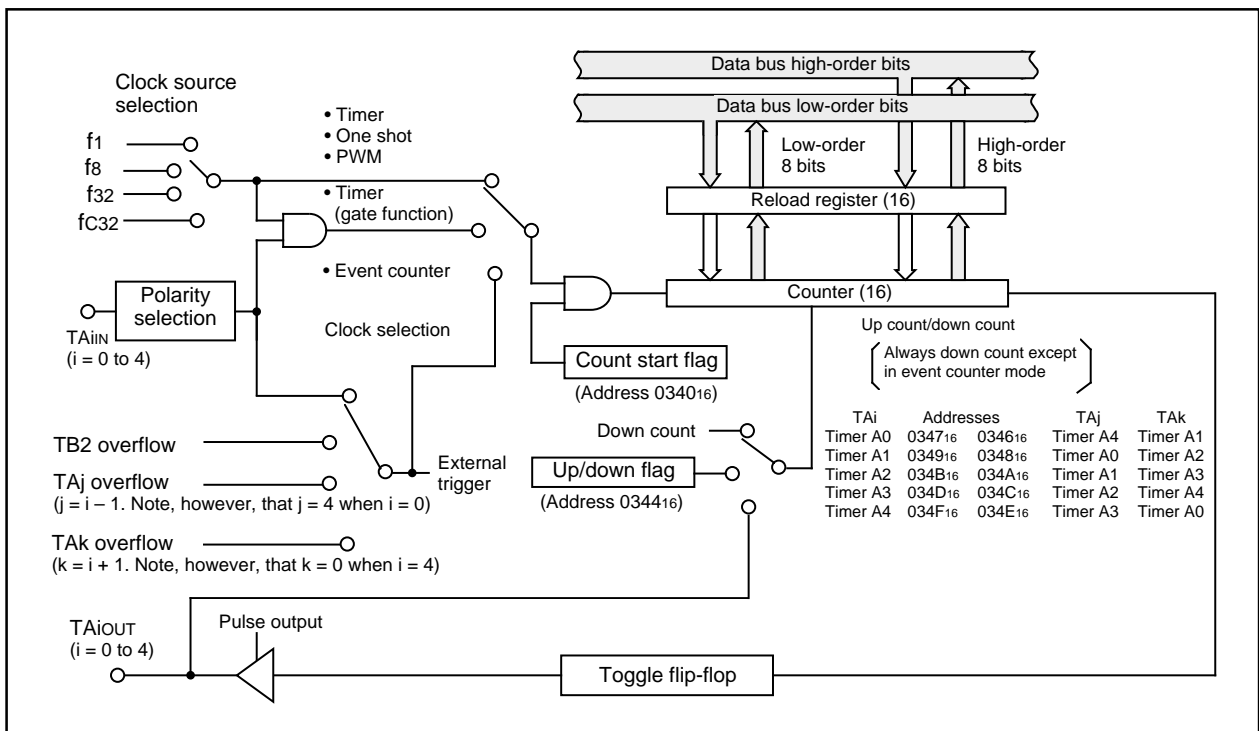


Figure 13.1 Block diagram of timer A

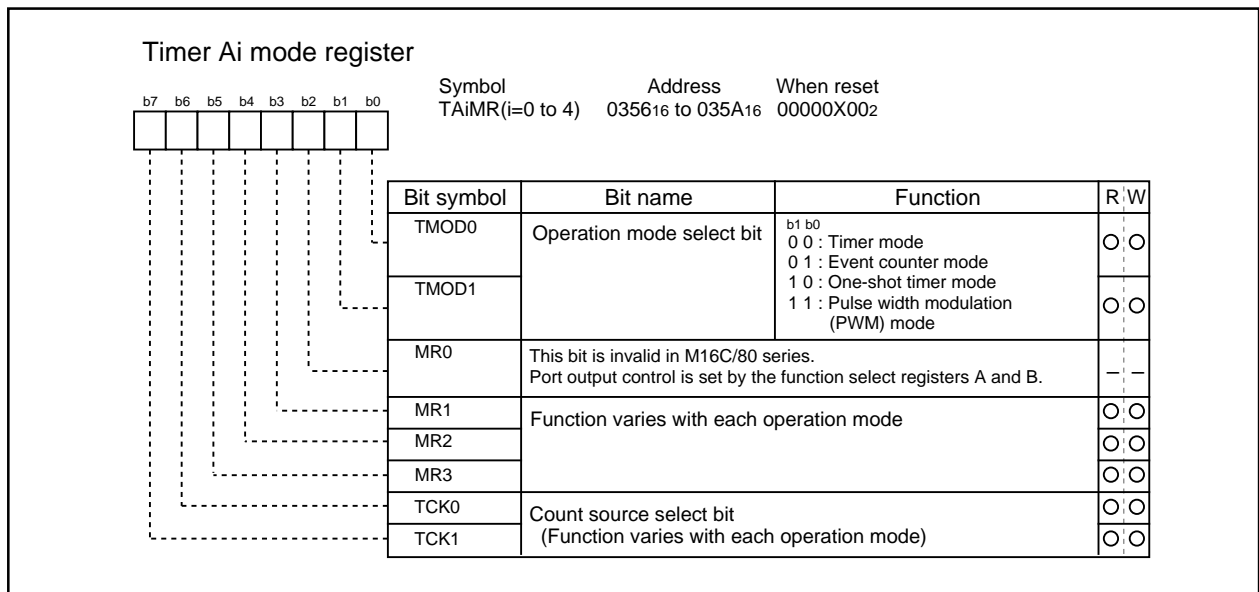


Figure 13.2 Timer A-related registers (1)

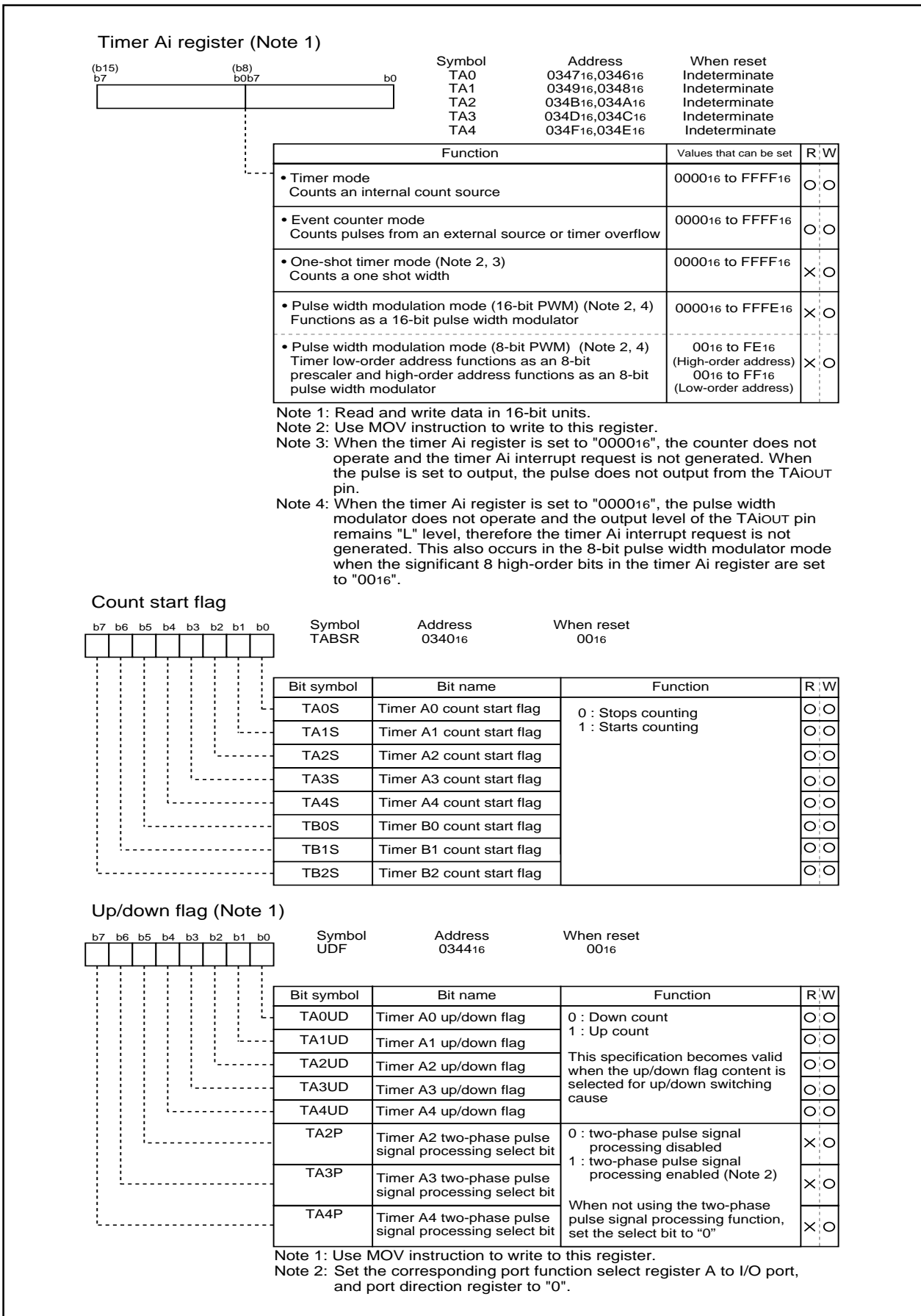


Figure 13.3 Timer A-related registers (2)

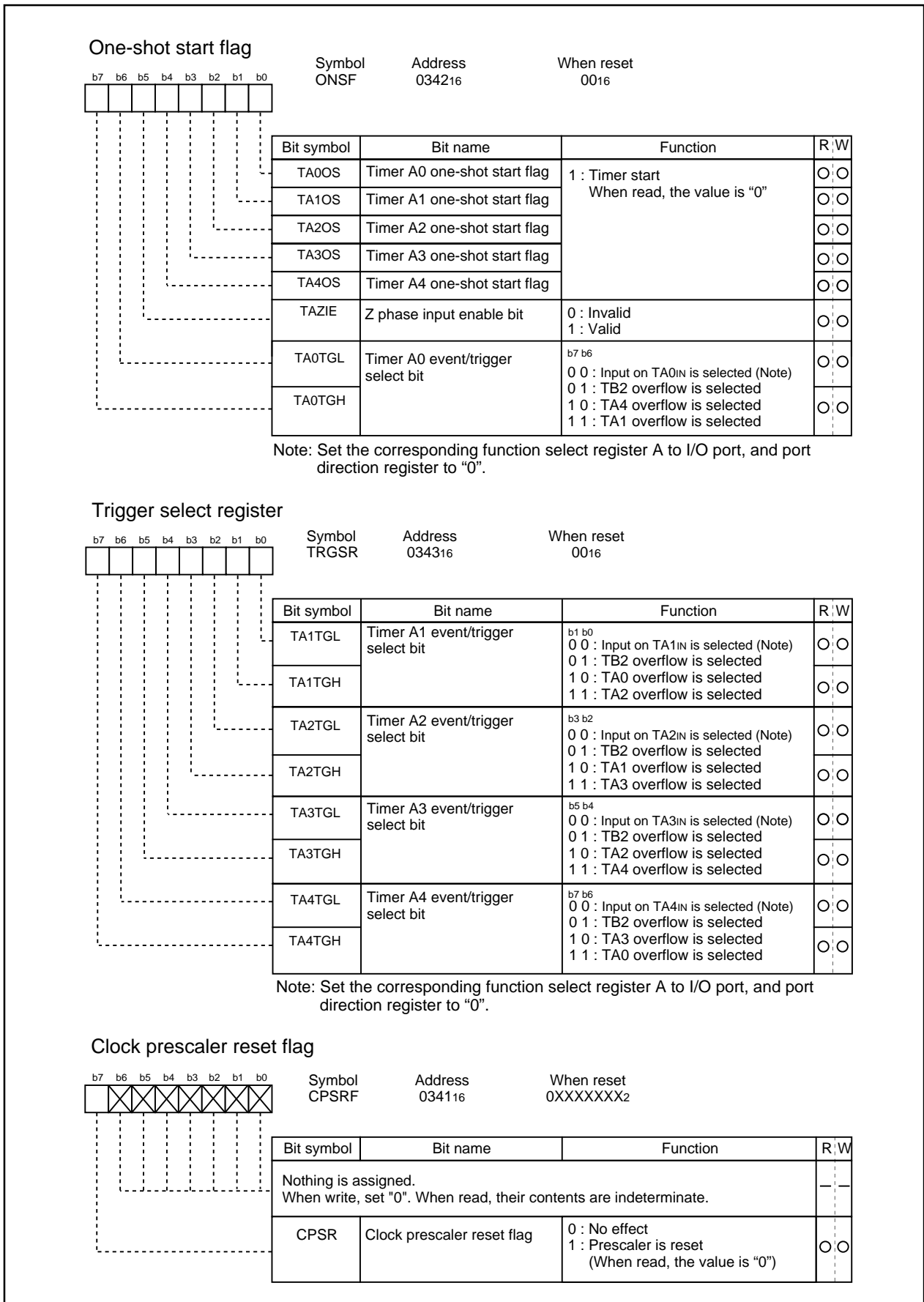


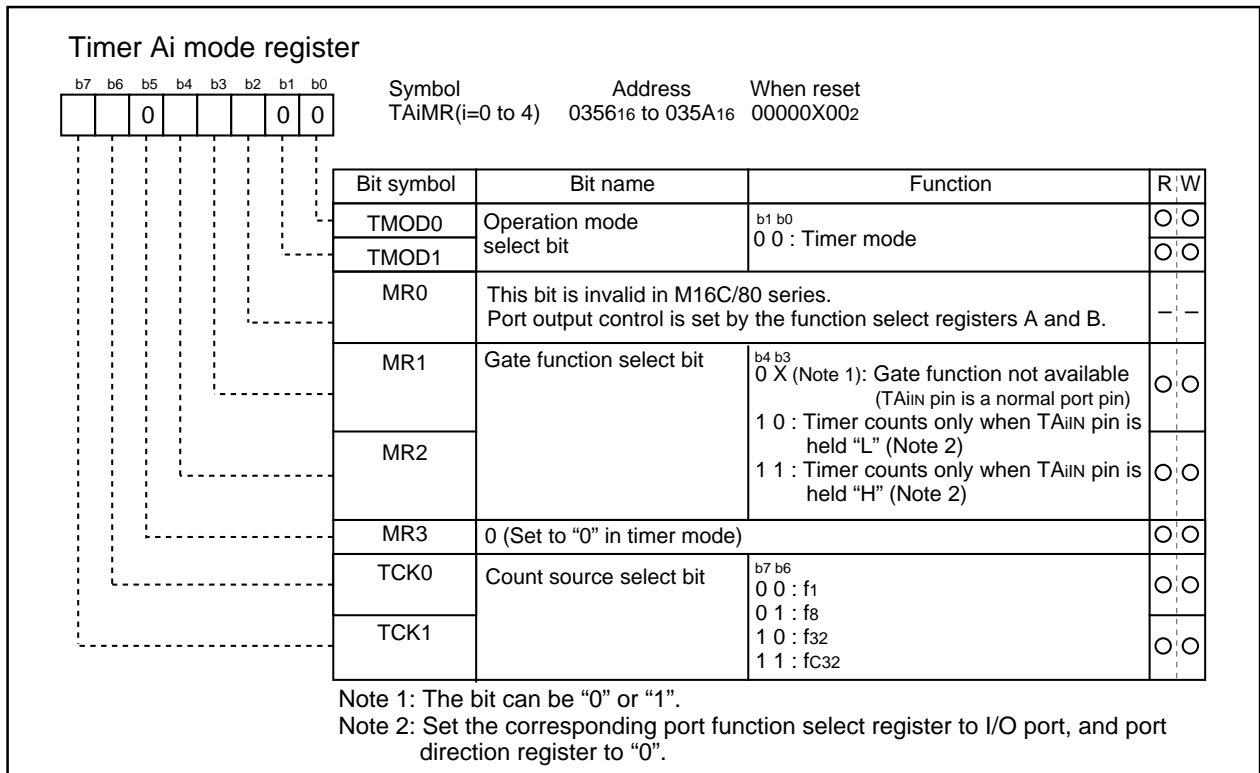
Figure 13.4 Timer A-related registers (3)

**(1) Timer mode**

In this mode, the timer counts an internally generated count source. (See Table 13.1) Figure 13.5 shows the timer Ai mode register in timer mode.

**Table 13.1 Specifications of timer mode**

Item	Specification
Count source	f1, f8, f32, fc32
Count operation	<ul style="list-style-type: none"> <li>• Down count</li> <li>• When the timer underflows, it reloads the reload register contents before continuing counting</li> </ul>
Divide ratio	1/(n+1) n : Set value
Count start condition	Count start flag is set (= 1)
Count stop condition	Count start flag is reset (= 0)
Interrupt request generation timing	When the timer underflows
TAiIN pin function	Programmable I/O port or gate input
TAiOUT pin function	Programmable I/O port or pulse output (Setting by the corresponding function select registers A and B)
Read from timer	Count value can be read out by reading timer Ai register
Write to timer	<ul style="list-style-type: none"> <li>• When not counting Value written to timer Ai register is written to both reload register and counter</li> <li>• When counting Value written to timer Ai register is written to only reload register (Transferred to counter at next reload time)</li> </ul>
Select function	<ul style="list-style-type: none"> <li>• Gate function Counting can be started and stopped by the TAiIN pin's input signal</li> <li>• Pulse output function Each time the timer underflows, the TAiOUT pin's polarity is reversed</li> </ul>



**Figure 13.5 Timer Ai mode register in timer mode**

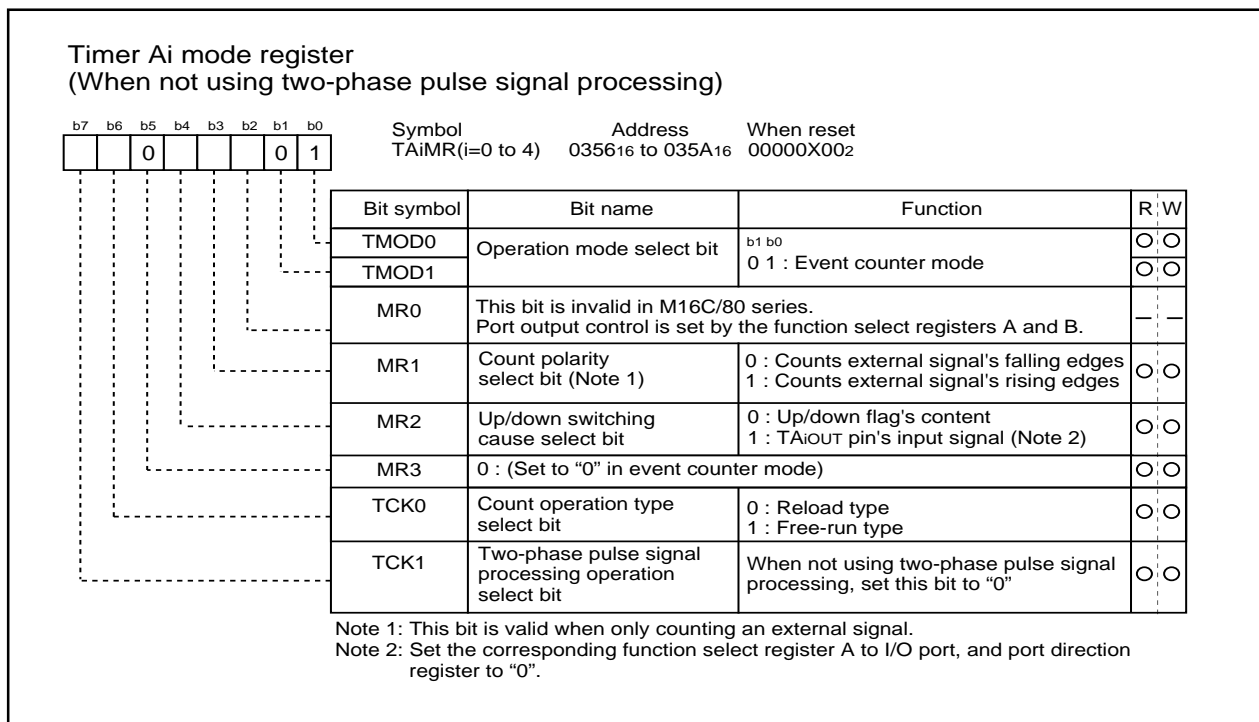
## (2) Event counter mode

In this mode, the timer counts an external signal or an internal timer's overflow. Timers A0 and A1 can count a single-phase external signal. Timers A2, A3, and A4 can count a single-phase and a two-phase external signal. Table 13.2 lists timer specifications when counting a single-phase external signal. Figure 13.6 shows the timer Ai mode register in event counter mode. Table 13.3 lists timer specifications when counting a two-phase external signal. Figure 13.7 shows the timer Ai mode register in event counter mode.

**Table 13.2 Timer specifications in event counter mode (when not processing two-phase pulse signal)**

Item	Specification
Count source	<ul style="list-style-type: none"> <li>External signals input to TAIiN pin (effective edge can be selected by software)</li> <li>TB2 overflows or underflows, TAJ overflows or underflows</li> </ul>
Count operation	<ul style="list-style-type: none"> <li>Up count or down count can be selected by external signal or software</li> <li>When the timer overflows or underflows, it reloads the reload register contents before continuing counting (Note)</li> </ul>
Divide ratio	<ul style="list-style-type: none"> <li>1/ (FFFF<sub>16</sub> - n + 1) for up count</li> <li>1/ (n + 1) for down count      n : Set value</li> </ul>
Count start condition	Count start flag is set (= 1)
Count stop condition	Count start flag is reset (= 0)
Interrupt request generation timing	The timer overflows or underflows
TAiIN pin function	Programmable I/O port or count source input
TAiOUT pin function	Programmable I/O port, pulse output, or up/down count select input (Setting by the corresponding function select registers A and B)
Read from timer	Count value can be read out by reading timer Ai register
Write to timer	<ul style="list-style-type: none"> <li>When not counting Value written to timer Ai register is written to both reload register and counter</li> <li>When counting Value written to timer Ai register is written to only reload register (Transferred to counter at next reload time)</li> </ul>
Select function	<ul style="list-style-type: none"> <li>Free-run count function Even when the timer overflows or underflows, the reload register content is not reloaded to it</li> <li>Pulse output function Each time the timer overflows or underflows, the TAIOUT pin's polarity is reversed</li> </ul>

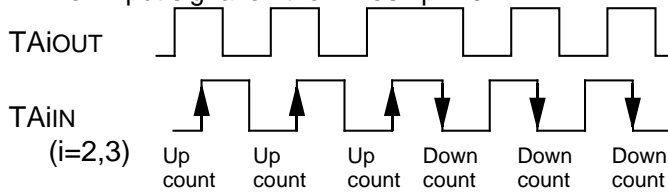
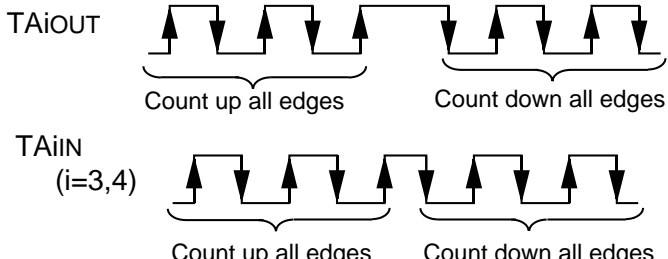
Note: This does not apply when the free-run function is selected.



**Figure 13.6 Timer Ai mode register in event counter mode**

**Table 13.3 Timer specifications in event counter mode**

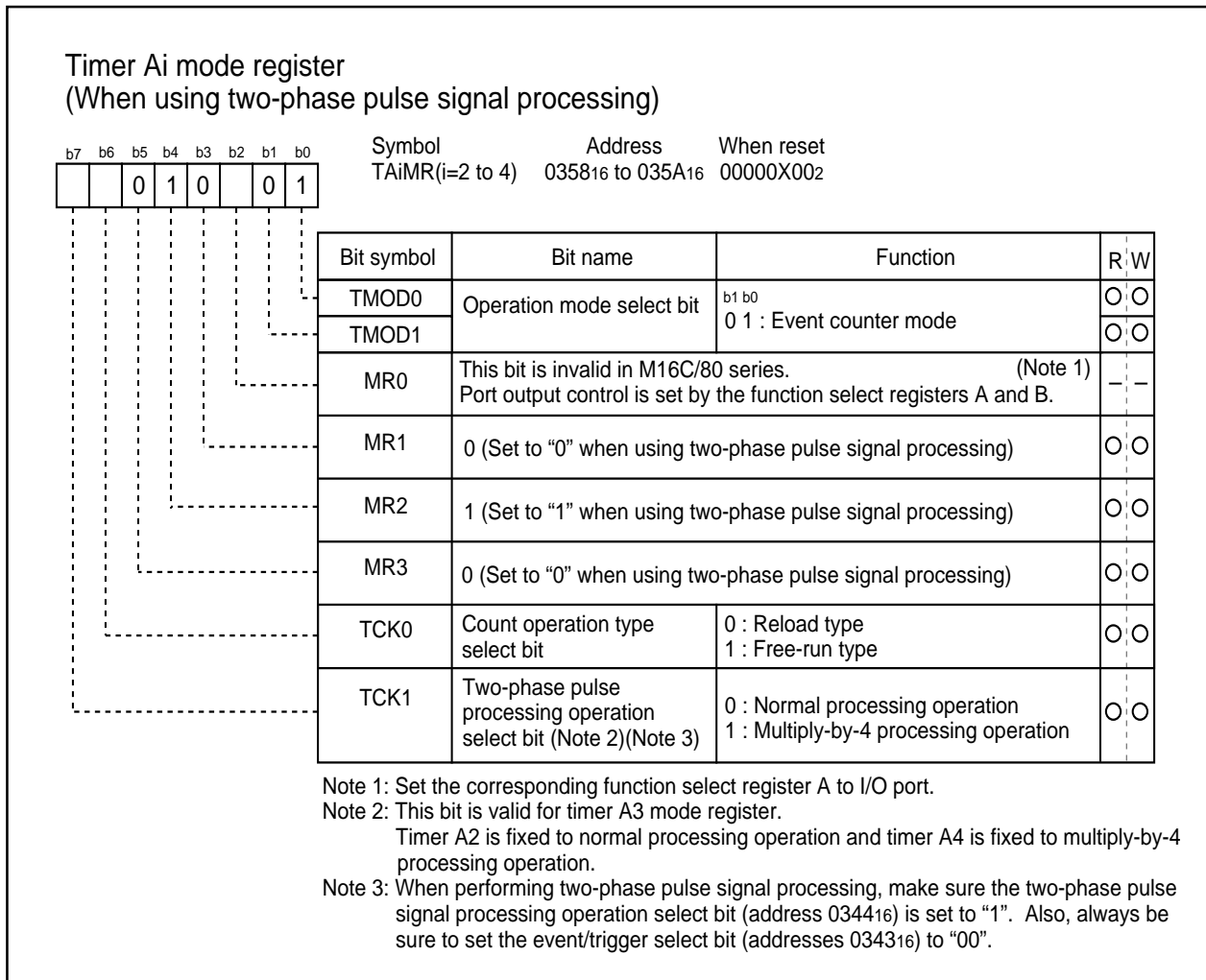
(when processing two-phase pulse signal with timers A2, A3, and A4)

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, the reload register content is reloaded and the timer starts over again (Note)
Divide ratio	• $1 / (FFFF_{16} - n + 1)$ for up count • $1 / (n + 1)$ for down count      n : Set value
Count start condition	Count start flag is set (= 1)
Count stop condition	Count start flag is reset (= 0)
Interrupt request generation timing	Timer overflows or underflows
TAiIN pin function	Two-phase pulse input (Set the corresponding function select registers A to I/O port, and port direction register to "0")
TAiOUT pin function	Two-phase pulse input (Set the corresponding function select registers A to I/O port, and port direction register to "0")
Read from timer	Count value can be read out by reading timer A2, A3, or A4 register
Write to timer	• When not counting Value written to timer Ai register is written to both reload register and counter • When counting Value written to timer Ai register is written to only reload register (Transferred to counter at next reload time)
Select function (Note 2)	<ul style="list-style-type: none"> <li>• Normal processing operation (TimerA2 and timer A3) The timer counts up rising edges or counts down falling edges on the TAIIN pin when input signal on the TAIOUT pin is "H"</li> </ul>  <ul style="list-style-type: none"> <li>• Multiply-by-4 processing operation (TimerA3 and timer A4) If the phase relationship is such that the TAIIN pin goes "H" when the input signal on the TAIOUT pin is "H", the timer counts up rising and falling edges on the TAIOUT and TAIIN pins. If the phase relationship is such that the TAIIN pin goes "L" when the input signal on the TAIOUT pin is "H", the timer counts down rising and falling edges on the TAIOUT and TAIIN pins.</li> </ul> 

Note 1: This does not apply when the free-run function is selected.

Note 2: Timer A3 is selectable. Timer A2 is fixed to normal processing operation and timer A4 is fixed to multiply-by-4 operation.





**Figure 13.7 Timer Ai mode register in event counter mode**

### • Counter Resetting by Two-Phase Pulse Signal Processing

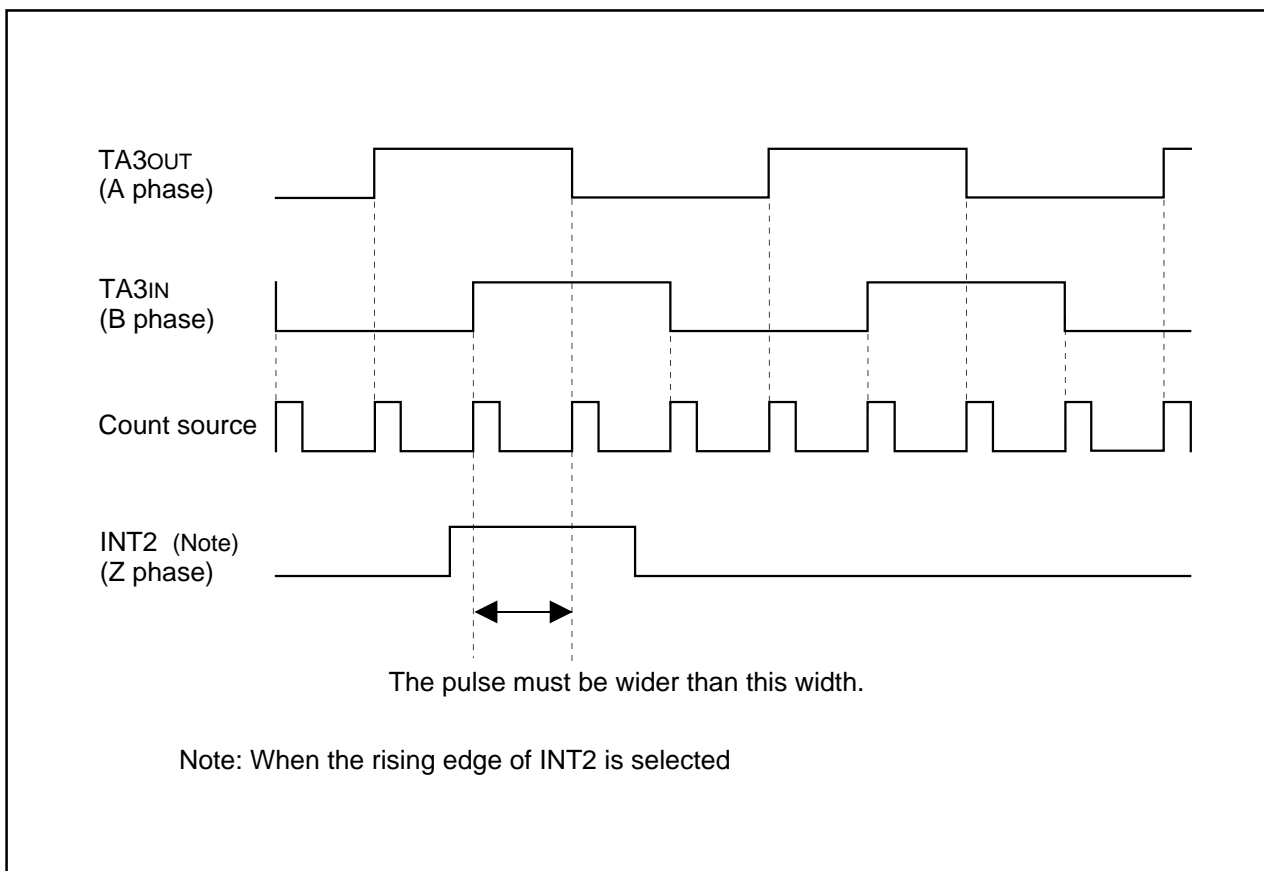
This function resets the timer counter to “0” when the Z-phase (counter reset) is input during two-phase pulse signal processing.

This function can only be used in timer A3 event counter mode, two-phase pulse signal processing, free-run type, and multiply-by-4 processing. The Z phase is input to the INT2 pin.

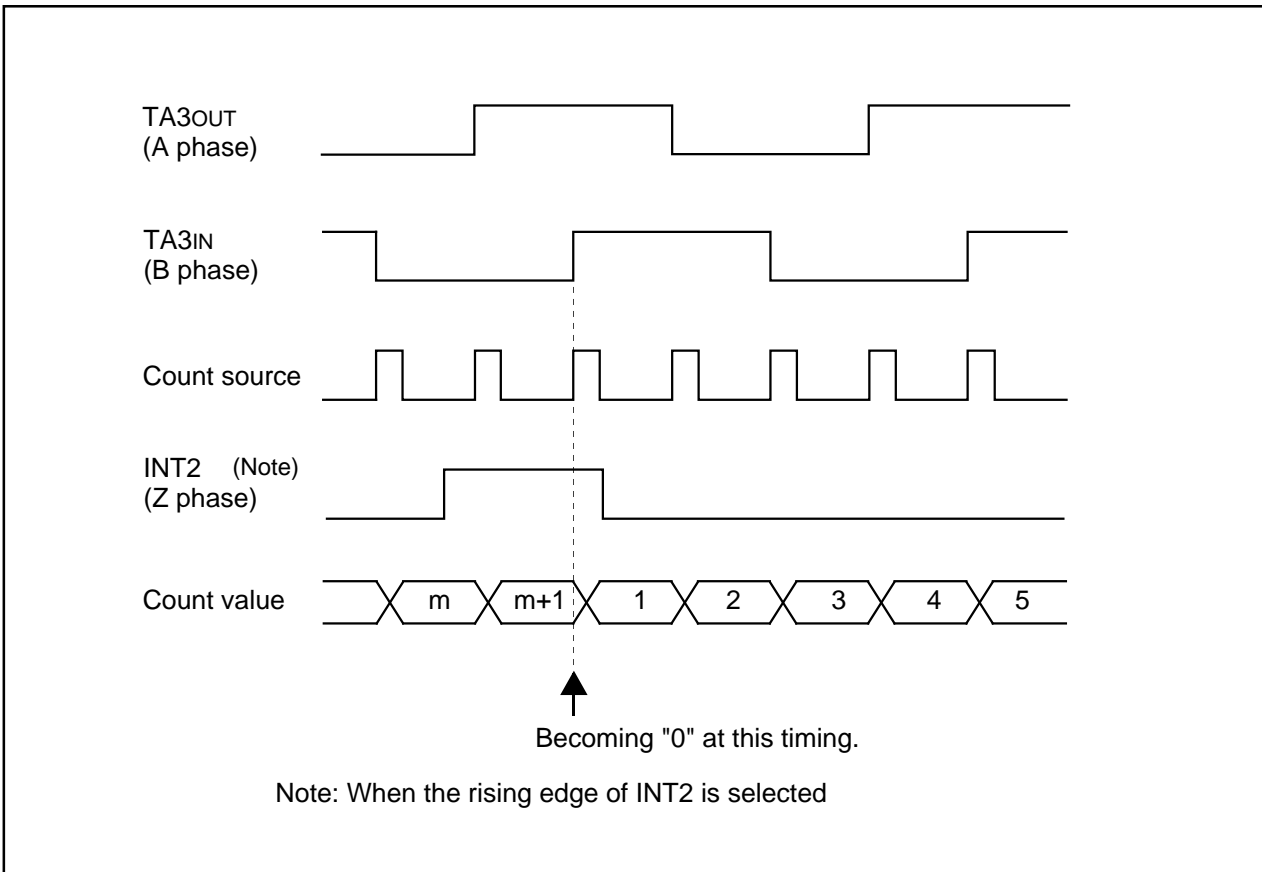
When the Z-phase input enable bit (bit 5 at address 0342<sub>16</sub>) is set to “1”, the counter can be reset by Z-phase input. For the counter to be reset to “0” by Z-phase input, you must first write “0000<sub>16</sub>” to the timer A3 register (address 034D<sub>16</sub> and 034C<sub>16</sub>).

The Z-phase is input when the INT2 input edge is detected. The edge polarity is selected by the INT2 polarity switch bit (bit 4 at address 009C<sub>16</sub>). The Z-phase must have a pulse width greater than 1 cycle of the timer A3 count source. Figure 13.8 shows the relationship between the two-phase pulse (A phase and B phase) and the Z phase.

The counter is reset at the count source following Z-phase input. Figure 13.9 shows the timing at which the counter is reset to “0”.



**Figure 13.8** The relationship between the two-phase pulse (A phase and B phase) and the Z phase



**Figure 13.9 The counter reset timing**

Note that timer A3 interrupt requests occur successively two times when timer A3 underflow and INT2 input reload are happened at the same timing.

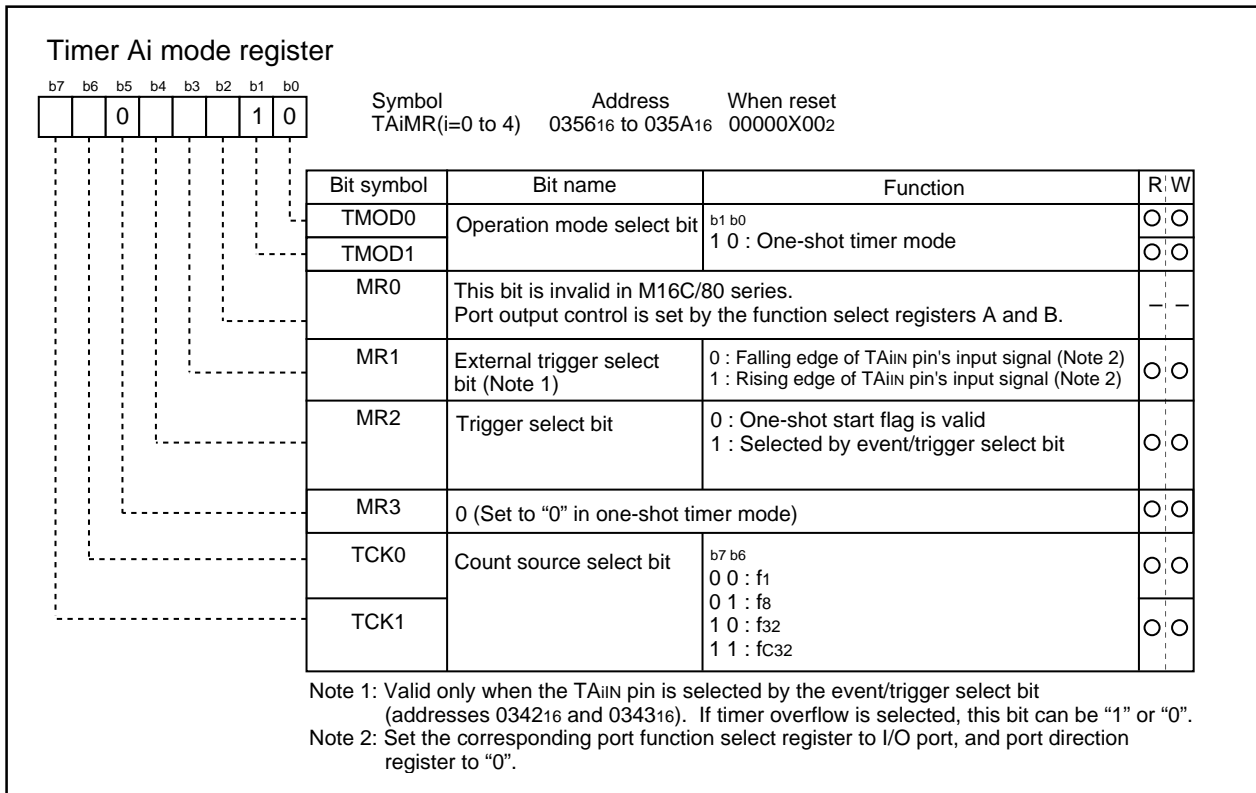
Do not use timer A3 interrupt request when this function is used.

**(3) One-shot timer mode**

In this mode, the timer operates only once. (See Table 13.4) When a trigger occurs, the timer starts up and continues operating for a given period. Figure 13.10 shows the timer Ai mode register in one-shot timer mode.

**Table 13.4 Timer specifications in one-shot timer mode**

Item	Specification
Count source	f1, f8, f32, fc32
Count operation	<ul style="list-style-type: none"> <li>The timer counts down</li> <li>When the count reaches 0000<sub>16</sub>, the timer stops counting after reloading a new count</li> <li>If a trigger occurs when counting, the timer reloads a new count and restarts counting</li> </ul>
Divide ratio	1/n    n : Set value
Count start condition	<ul style="list-style-type: none"> <li>An external trigger is input</li> <li>The timer overflows</li> <li>The one-shot start flag is set (= 1)</li> </ul>
Count stop condition	<ul style="list-style-type: none"> <li>A new count is reloaded after the count has reached 0000<sub>16</sub></li> <li>The count start flag is reset (= 0)</li> </ul>
Interrupt request generation timing	The count reaches 0000 <sub>16</sub>
TAiIN pin function	Programmable I/O port or trigger input
TAiOUT pin function	Programmable I/O port or pulse output (Setting by the corresponding function select registers A and B)
Read from timer	When timer Ai register is read, it indicates an indeterminate value
Write to timer	<ul style="list-style-type: none"> <li>When not counting Value written to timer Ai register is written to both reload register and counter</li> <li>When counting Value written to timer Ai register is written to only reload register (Transferred to counter at next reload time)</li> </ul>



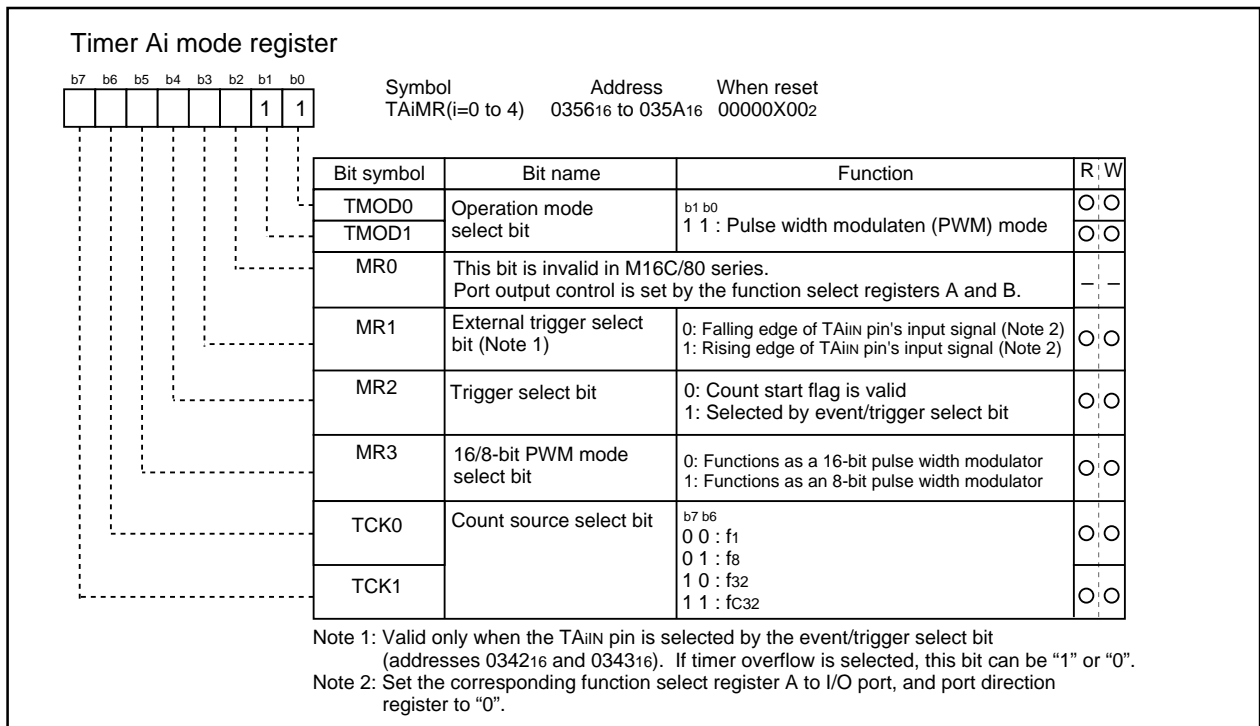
**Figure 13.10 Timer Ai mode register in one-shot timer mode**

**(4) Pulse width modulation (PWM) mode**

In this mode, the timer outputs pulses of a given width in succession. (See Table 13.5) In this mode, the counter functions as either a 16-bit pulse width modulator or an 8-bit pulse width modulator. Figure 13.11 shows the timer Ai mode register in pulse width modulation mode. Figure 13.12 shows the example of how a 16-bit pulse width modulator operates. Figure 13.13 shows the example of how an 8-bit pulse width modulator operates.

**Table 13.5 Timer specifications in pulse width modulation mode**

Item	Specification
Count source	f1, f8, f32, fC32
Count operation	<ul style="list-style-type: none"> <li>The timer counts down (operating as an 8-bit or a 16-bit pulse width modulator)</li> <li>The timer reloads a new count at a rising edge of PWM pulse and continues counting</li> <li>The timer is not affected by a trigger that occurs when counting</li> </ul>
16-bit PWM	<ul style="list-style-type: none"> <li>High level width <math>n / f_i</math> n : Set value</li> <li>Cycle time <math>(2^{16}-1) / f_i</math> fixed</li> </ul>
8-bit PWM	<ul style="list-style-type: none"> <li>High level width <math>n \times (m+1) / f_i</math> n : values set to timer Ai register's high-order address</li> <li>Cycle time <math>(2^8-1) \times (m+1) / f_i</math> m: values set to timer Ai register's low-order address</li> </ul>
Count start condition	<ul style="list-style-type: none"> <li>External trigger is input</li> <li>The timer overflows</li> <li>The count start flag is set (= 1)</li> </ul>
Count stop condition	<ul style="list-style-type: none"> <li>The count start flag is reset (= 0)</li> </ul>
Interrupt request generation timing	PWM pulse goes "L"
TAiIN pin function	Programmable I/O port or trigger input
TAiOUT pin function	Pulse output (TAiOUT output is selected by the corresponding function select registers A and B)
Read from timer	When timer Ai register is read, it indicates an indeterminate value
Write to timer	<ul style="list-style-type: none"> <li>When not counting Value written to timer Ai register is written to both reload register and counter</li> <li>When counting Value written to timer Ai register is written to only reload register (Transferred to counter at next reload time)</li> </ul>



**Figure 13.11 Timer Ai mode register in pulse width modulation mode**

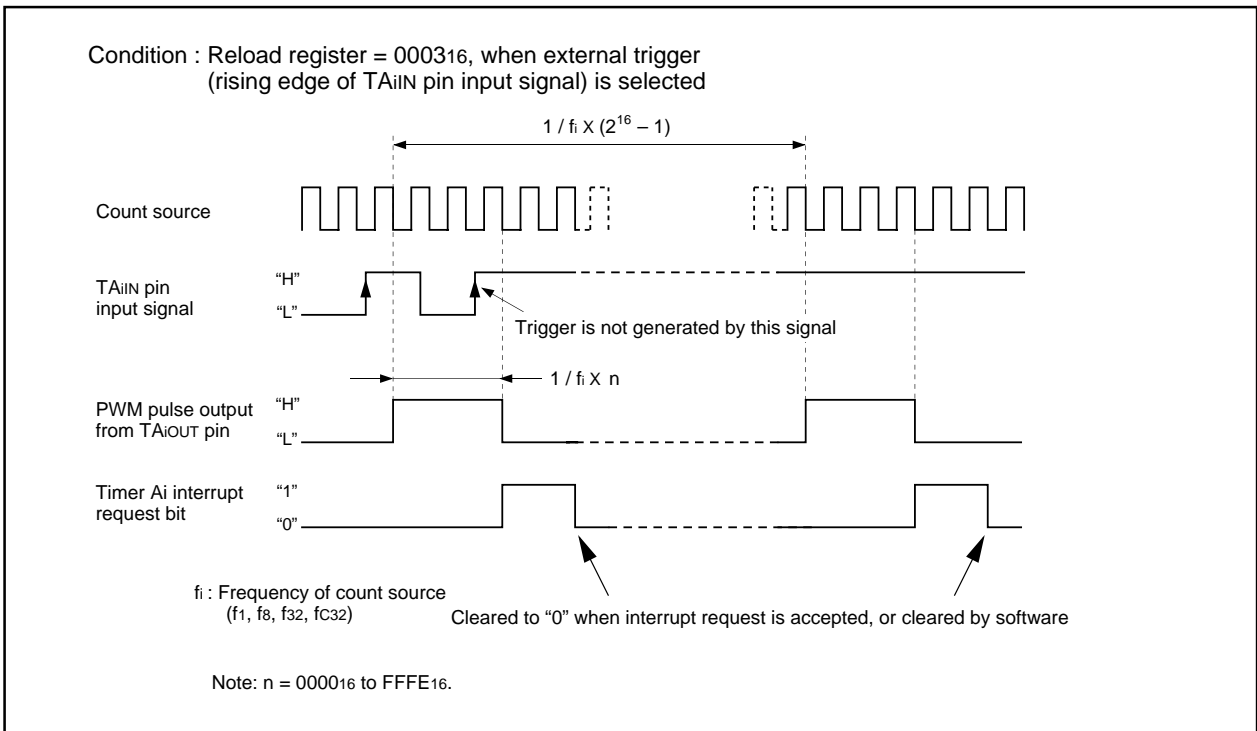


Figure 13.12 Example of how a 16-bit pulse width modulator operates

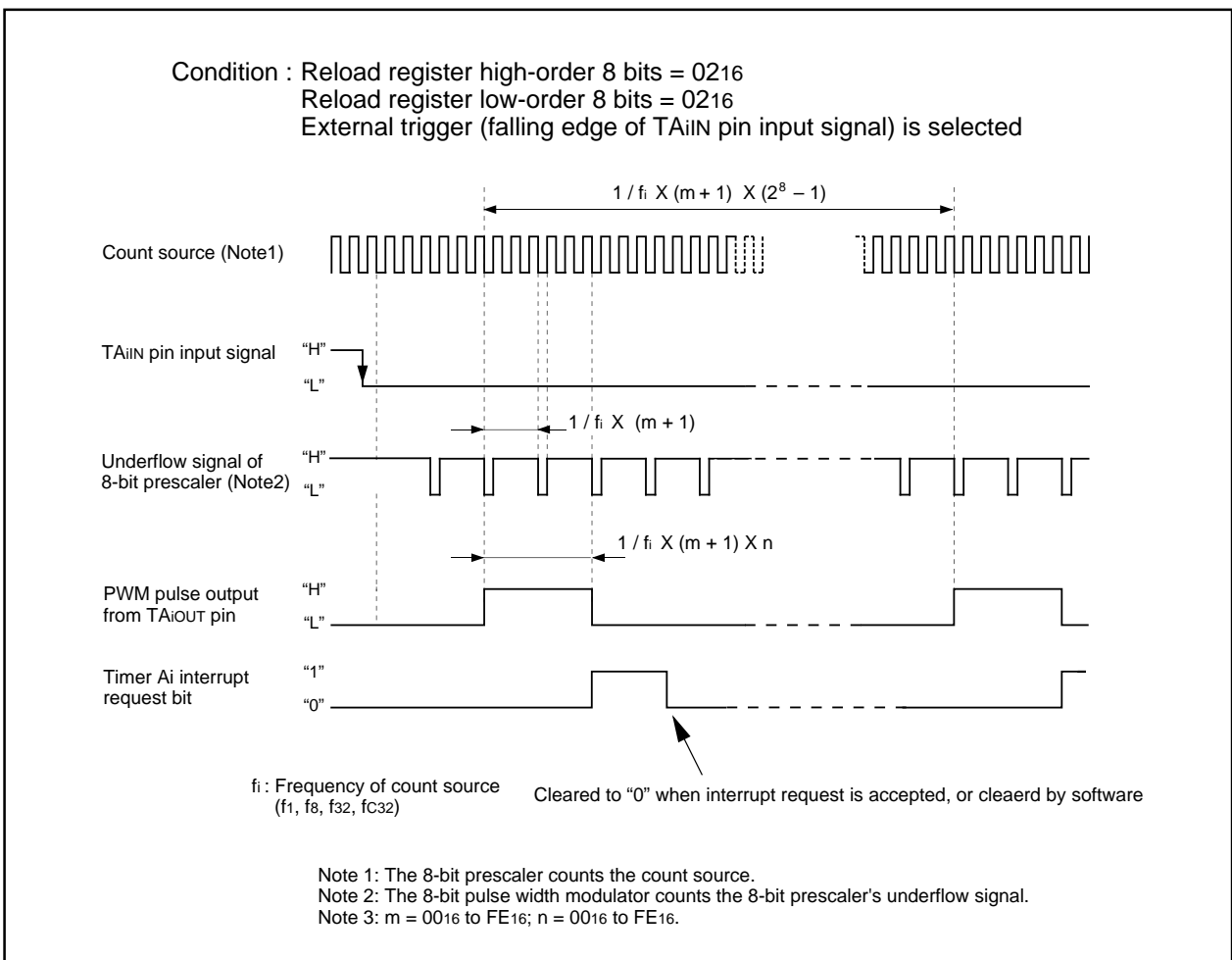


Figure 13.13 Example of how an 8-bit pulse width modulator operates

### 14. Timer B

Figure 14.1 shows the block diagram of timer B. Figures 14.2 and 14.3 show the timer B-related registers. Use the timer Bi mode register (i = 0 to 5) bits 0 and 1 to choose the desired mode.

Timer B has three operation modes listed as follows:

- Timer mode: The timer counts an internal count source.
- Event counter mode: The timer counts pulses from an external source or a timer overflow.
- Pulse period/pulse width measuring mode: The timer measures an external signal's pulse period or pulse width.

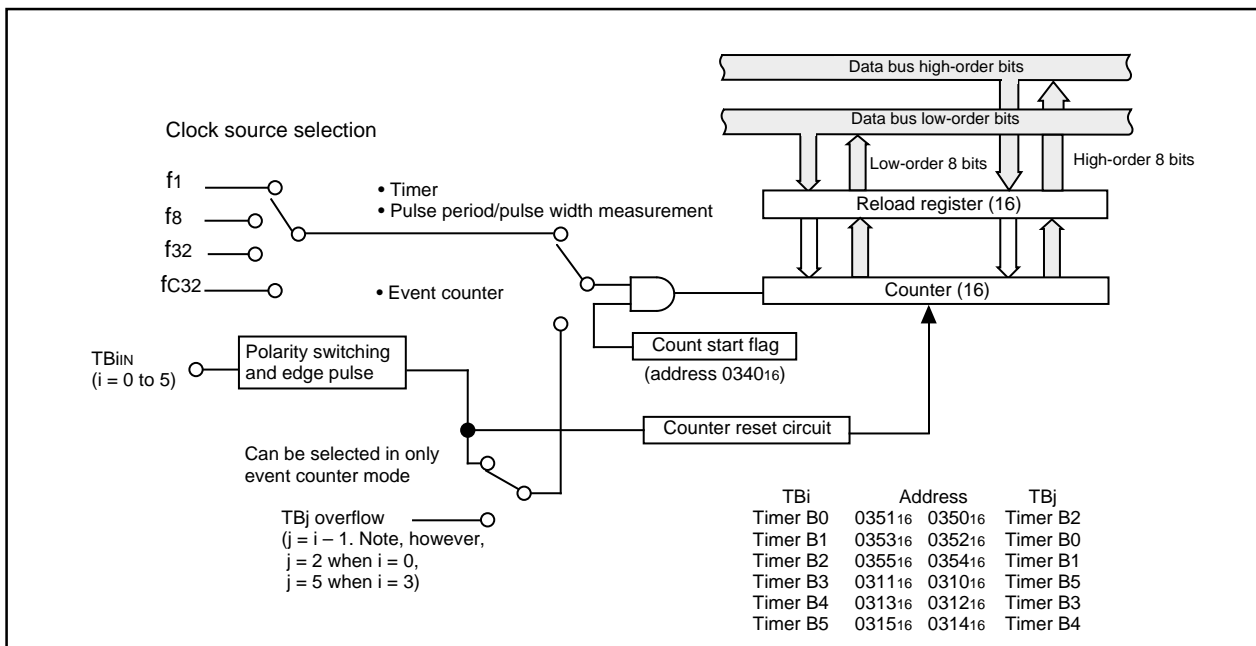


Figure 14.1 Block diagram of timer B

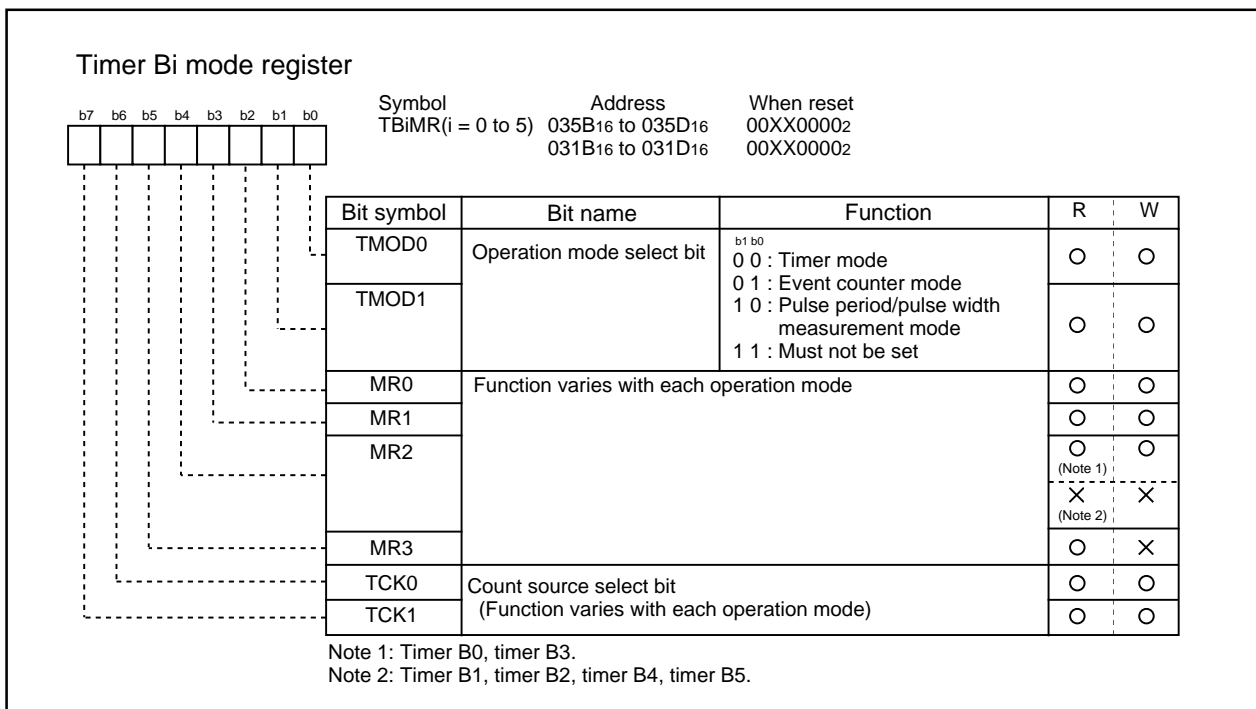
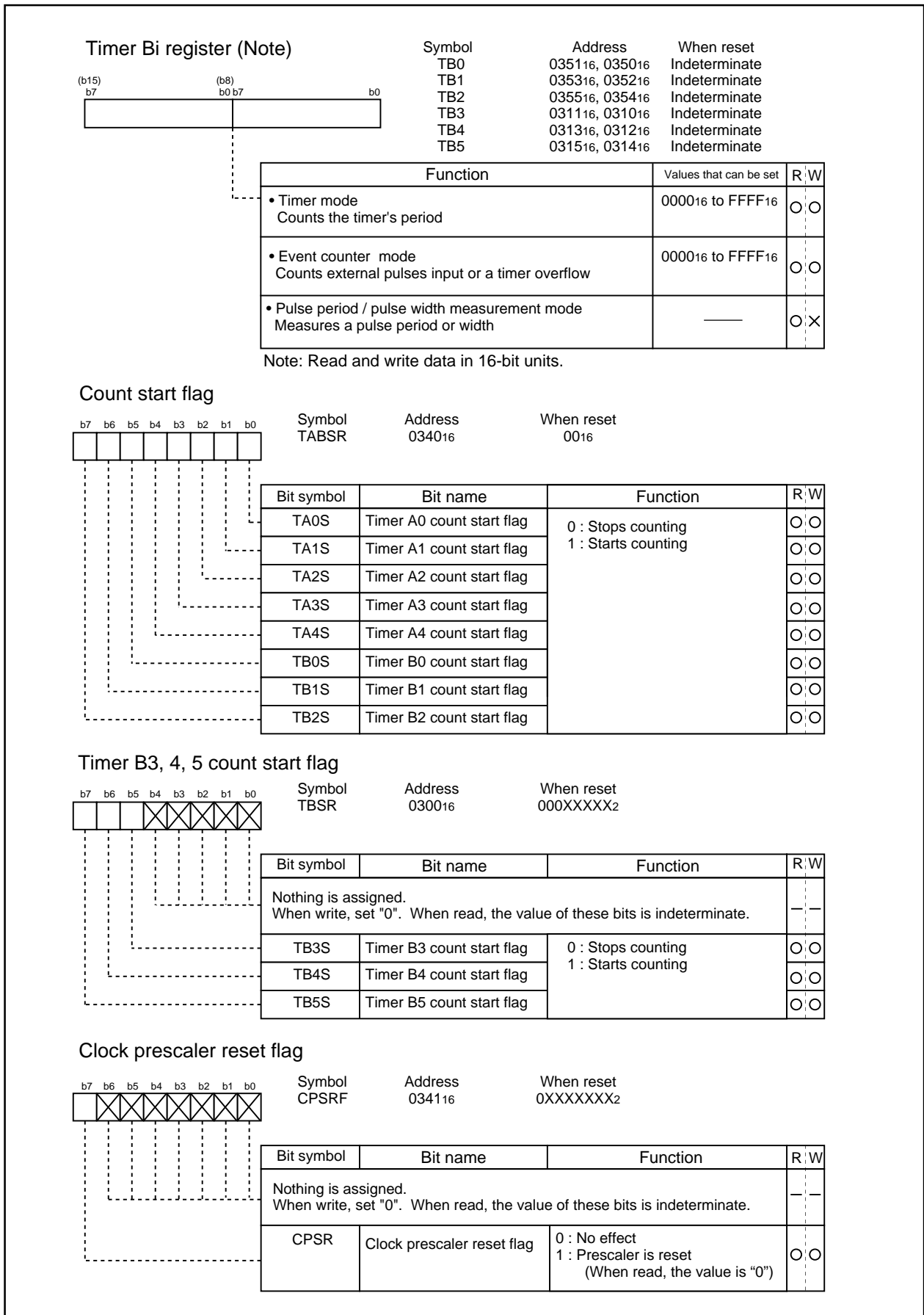


Figure 14.2 Timer B-related registers (1)



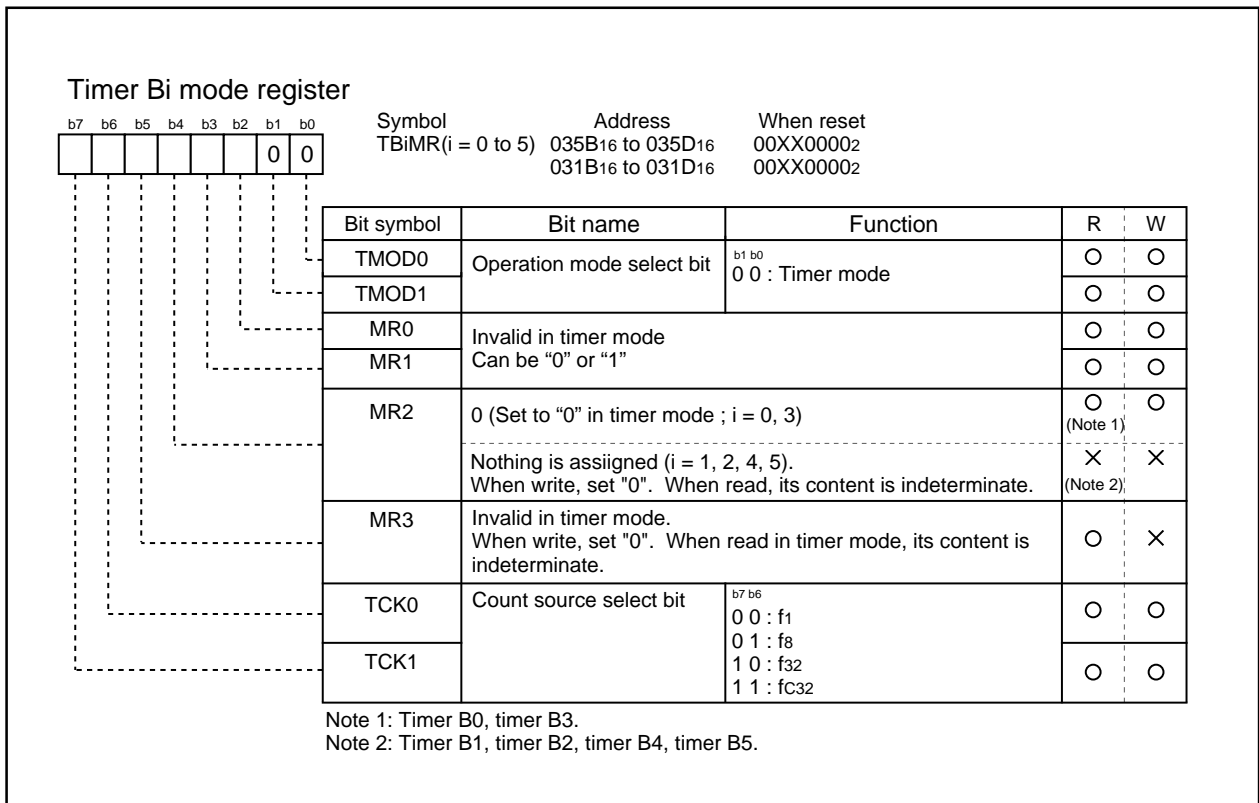


**(1) Timer mode**

In this mode, the timer counts an internally generated count source. (See Table 14.1) Figure 14.4 shows the timer Bi mode register in timer mode.

**Table 14.1 Timer specifications in timer mode**

Item	Specification
Count source	f1, f8, f32, fC32
Count operation	<ul style="list-style-type: none"> <li>Counts down</li> <li>When the timer underflows, it reloads the reload register contents before continuing counting</li> </ul>
Divide ratio	1/(n+1) n : Set value
Count start condition	Count start flag is set (= 1)
Count stop condition	Count start flag is reset (= 0)
Interrupt request generation timing	The timer underflows
TBiIN pin function	Programmable I/O port
Read from timer	Count value is read out by reading timer Bi register
Write to timer	<ul style="list-style-type: none"> <li>When not counting Value written to timer Bi register is written to both reload register and counter</li> <li>When counting Value written to timer Bi register is written to only reload register (Transferred to counter at next reload time)</li> </ul>



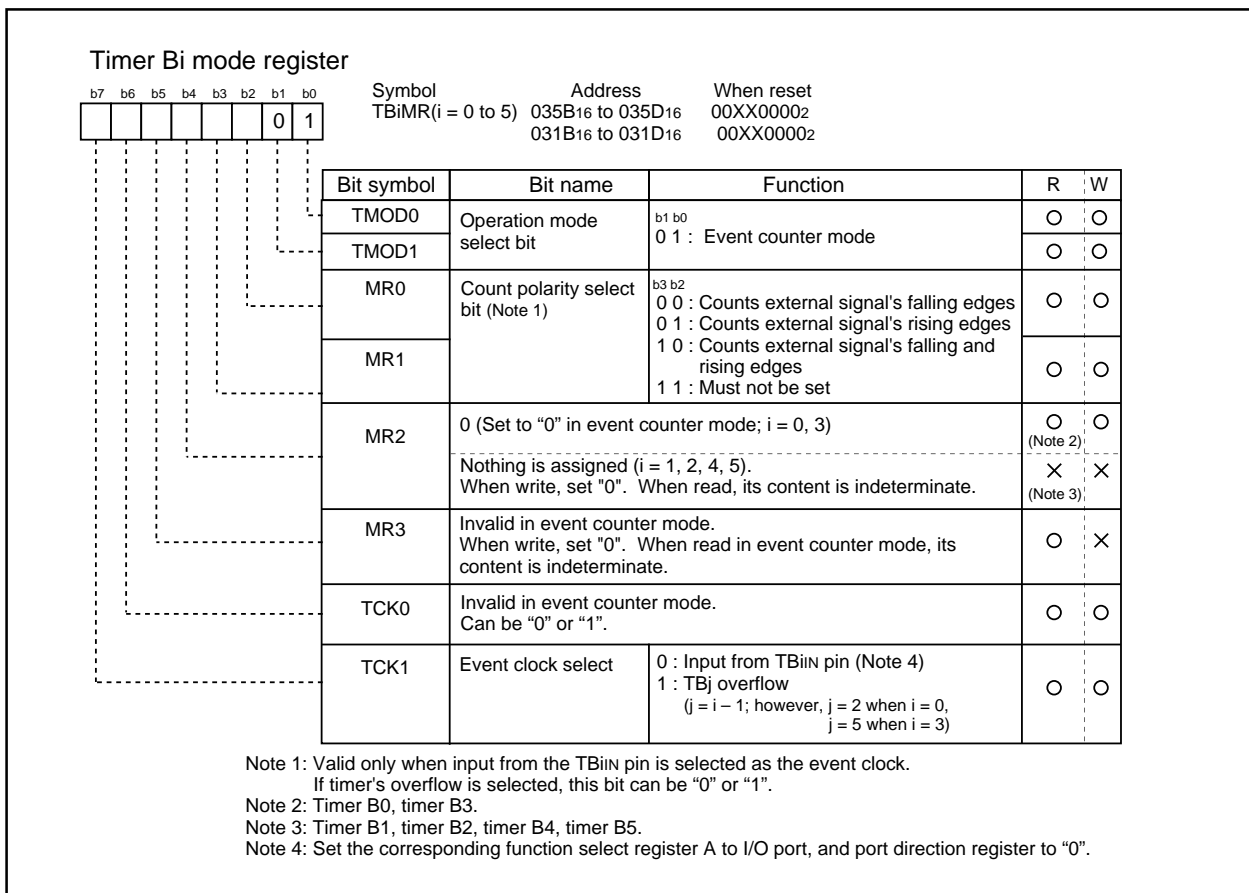
**Figure 14.4 Timer Bi mode register in timer mode**

## (2) Event counter mode

In this mode, the timer counts an external signal or an internal timer's overflow. (See Table 14.2) Figure 14.5 shows the timer Bi mode register in event counter mode.

**Table 14.2 Timer specifications in event counter mode**

Item	Specification
Count source	<ul style="list-style-type: none"> <li>External signals input to TBIiN pin</li> <li>Effective edge of count source can be a rising edge, a falling edge, or falling and rising edges as selected by software</li> <li>TBi overflows or underflows</li> </ul>
Count operation	<ul style="list-style-type: none"> <li>Counts down</li> <li>When the timer underflows, it reloads the reload register contents before continuing counting</li> </ul>
Divide ratio	$1/(n+1)$ $n$ : Set value
Count start condition	Count start flag is set (= 1)
Count stop condition	Count start flag is reset (= 0)
Interrupt request generation timing	The timer underflows
TBIiN pin function	Programable I/O port or Count source input (Set the corresponding function select register A to I/O port.)
Read from timer	Count value can be read out by reading timer Bi register
Write to timer	<ul style="list-style-type: none"> <li>When not counting</li> <li>Value written to timer Bi register is written to both reload register and counter</li> <li>When counting</li> <li>Value written to timer Bi register is written to only reload register (Transferred to counter at next reload time)</li> </ul>



**Figure 14.5 Timer Bi mode register in event counter mode**

### (3) Pulse period/pulse width measurement mode

In this mode, the timer measures the pulse period or pulse width of an external signal. (See Table 14.3) Figure 14.6 shows the timer Bi mode register in pulse period/pulse width measurement mode. Figure 14.7 shows the operation timing when measuring a pulse period. Figure 14.8 shows the operation timing when measuring a pulse width.

**Table 14.3 Timer specifications in pulse period/pulse width measurement mode**

Item	Specification
Count source	f1, f8, f32, fc32
Count operation	<ul style="list-style-type: none"> <li>Up count</li> <li>Counter value "000016" is transferred to reload register at measurement pulse's effective edge and the timer continues counting</li> </ul>
Count start condition	Count start flag is set (= 1)
Count stop condition	Count start flag is reset (= 0)
Interrupt request generation timing	<ul style="list-style-type: none"> <li>When measurement pulse's effective edge is input (Note 1)</li> <li>Timer overflow. When an overflow occurs, the timer Bi overflow flag set to "1" simultaneously. The timer Bi overflow flag cleared to "0" by writing to the timer mode register at the next count timing or later after the timer Bi overflow flag was set to "1". At this time, make sure the timer start flag is set to "1".</li> </ul>
TBiIN pin function	Measurement pulse input (Set the corresponding function select register A to I/O port.)
Read from timer	When timer Bi register is read, it indicates the reload register's content (measurement result) (Note 2)
Write to timer	Cannot be written to

Note 1: An interrupt request is not generated when the first effective edge is input after the timer has started counting.

Note 2: The value read out from the timer Bi register is indeterminate until the second effective edge is input after the timer.

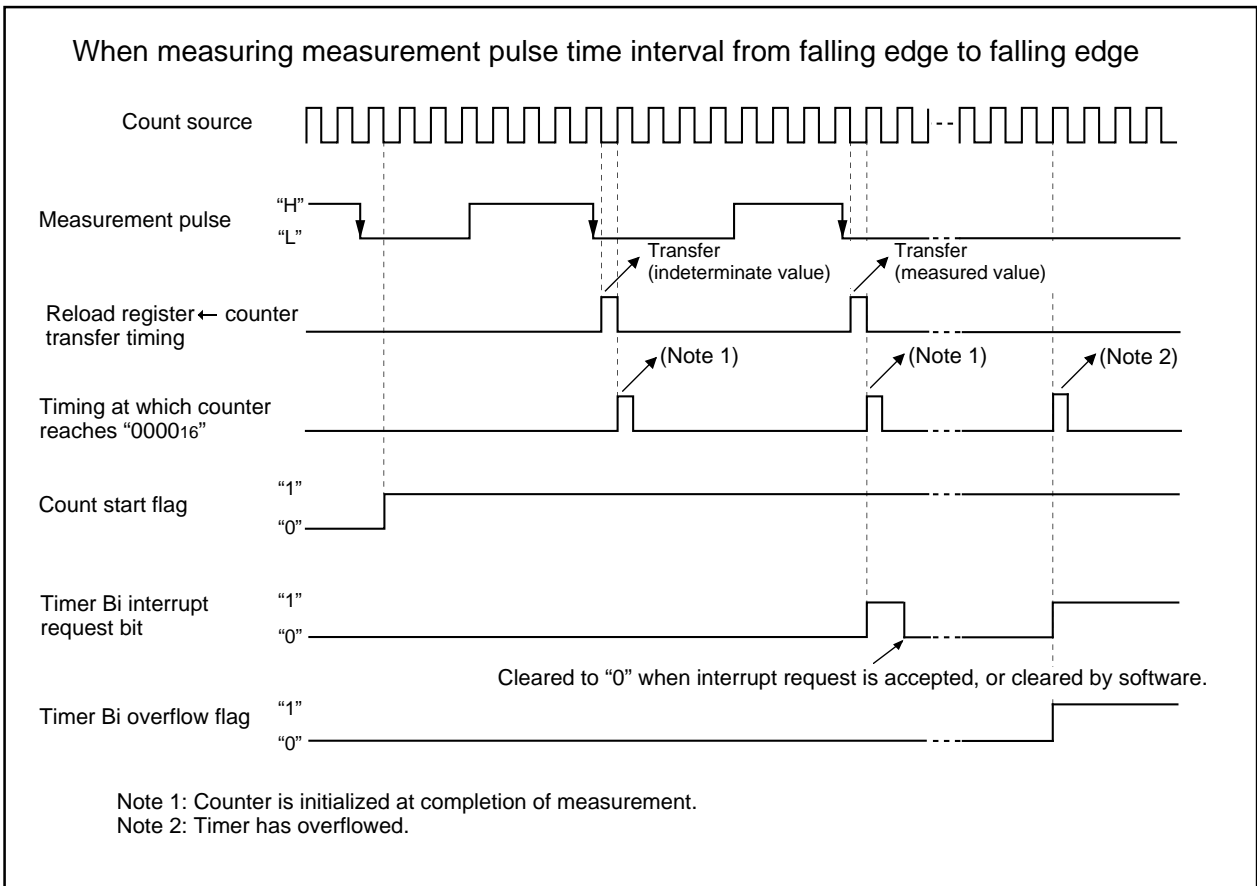
Timer Bi mode register		Symbol	Address	When reset	
		TBiMR(i = 0 to 5)	035B16 to 035D16 031B16 to 031D16	00XX00002 00XX00002	
Bit symbol	Bit name	Function		R	W
TMOD0	Operation mode select bit	b1 b0 1 0 : Pulse period / pulse width measurement mode		○	○
				○	○
MR0	Measurement mode select bit	b3 b2 0 0 : Pulse period measurement (Interval between measurement pulse's falling edge to falling edge) 0 1 : Pulse period measurement (Interval between measurement pulse's rising edge to rising edge)		○	○
				MR1	1 0 : Pulse width measurement (Interval between measurement pulse's falling edge to rising edge, and between rising edge to falling edge) 1 1 : Must not be set
MR2	0 (Set to "0" in pulse period/pulse width measurement mode; i = 0, 3)		○		
	Nothing is assigned (i = 1, 2, 4, 5). When write, set "0". When read, its content is indeterminate.		×	×	
MR3	Timer Bi overflow flag ( Note 1)	0 : Timer did not overflow 1 : Timer has overflowed		○	×
TCK0	Count source select bit	b7 b6 0 0 : f1 0 1 : f8 1 0 : f32 1 1 : fc32		○	○
				○	○

Note 1: This flag is indeterminate after reset. When the timer Bi start flag = 1, the timer Bi overflow flag is cleared to "0" by writing to the timer Bi mode register at the next count timing or later after the timer Bi overflow flag was set to "1". The Timer Bi overflow flag cannot be set to "1" in a program.

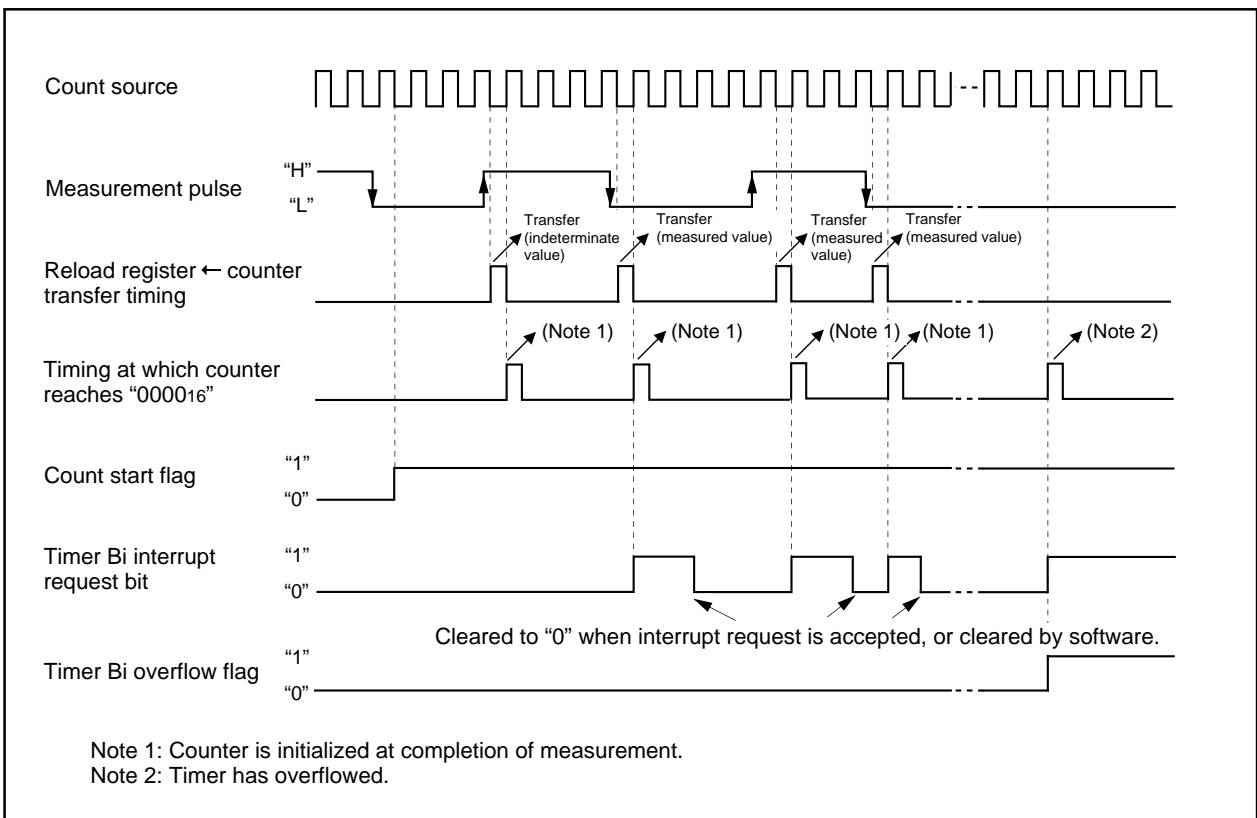
Note 2: Timer B0, timer B3.

Note 3: Timer B1, timer B2, timer B4, timer B5.

**Figure 14.6 Timer Bi mode register in pulse period/pulse width measurement mode**



**Figure 14.7 Operation timing when measuring a pulse period**

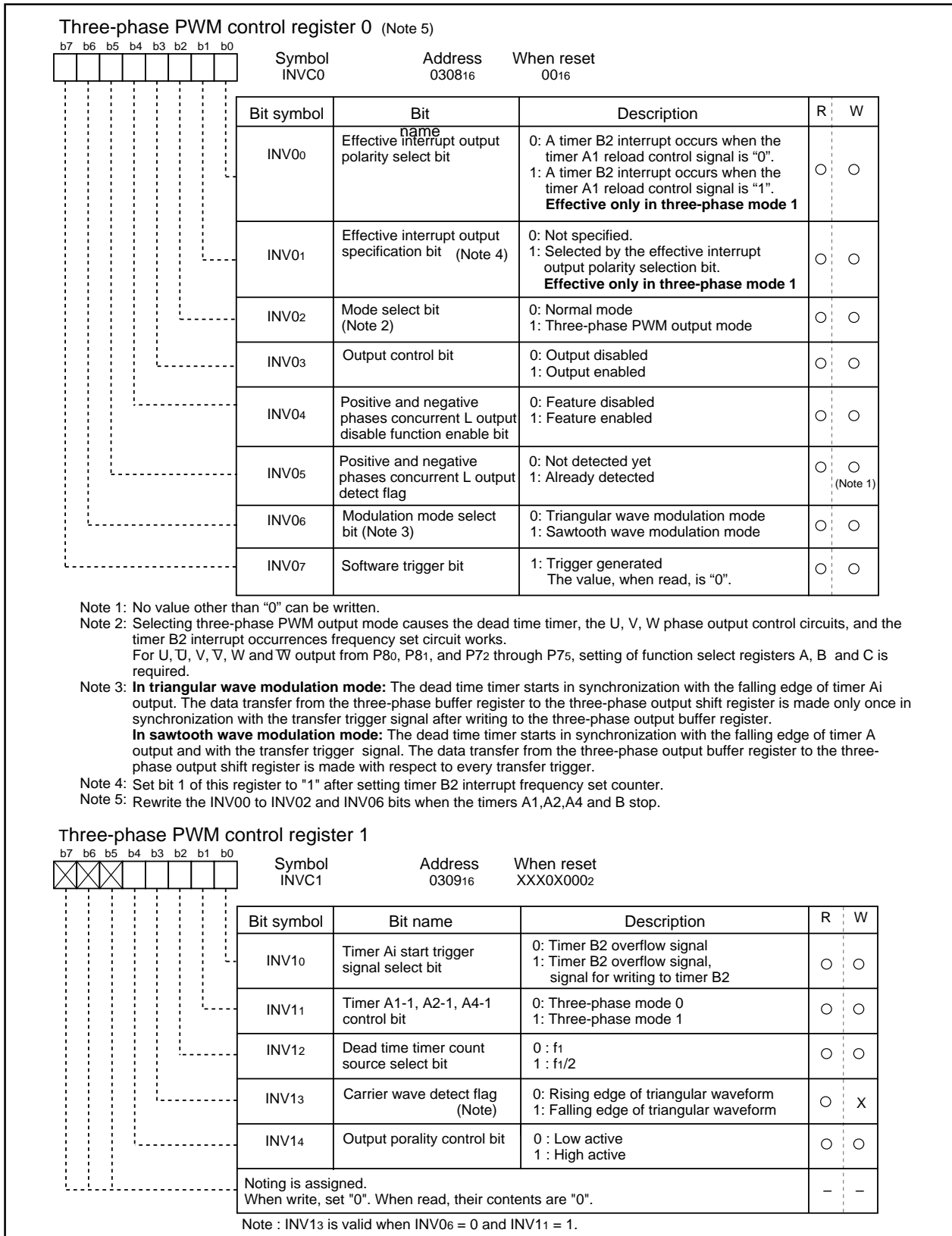


**Figure 14.8 Operation timing when measuring a pulse width**

### 15. Three-phase motor control timers' functions

Use of more than one built-in timer A and timer B provides the means of outputting three-phase motor driving waveforms.

Figures 15.1 through 15.3 show registers related to timers for three-phase motor control.



**Figure 15.1 Registers related to timers for three-phase motor control**

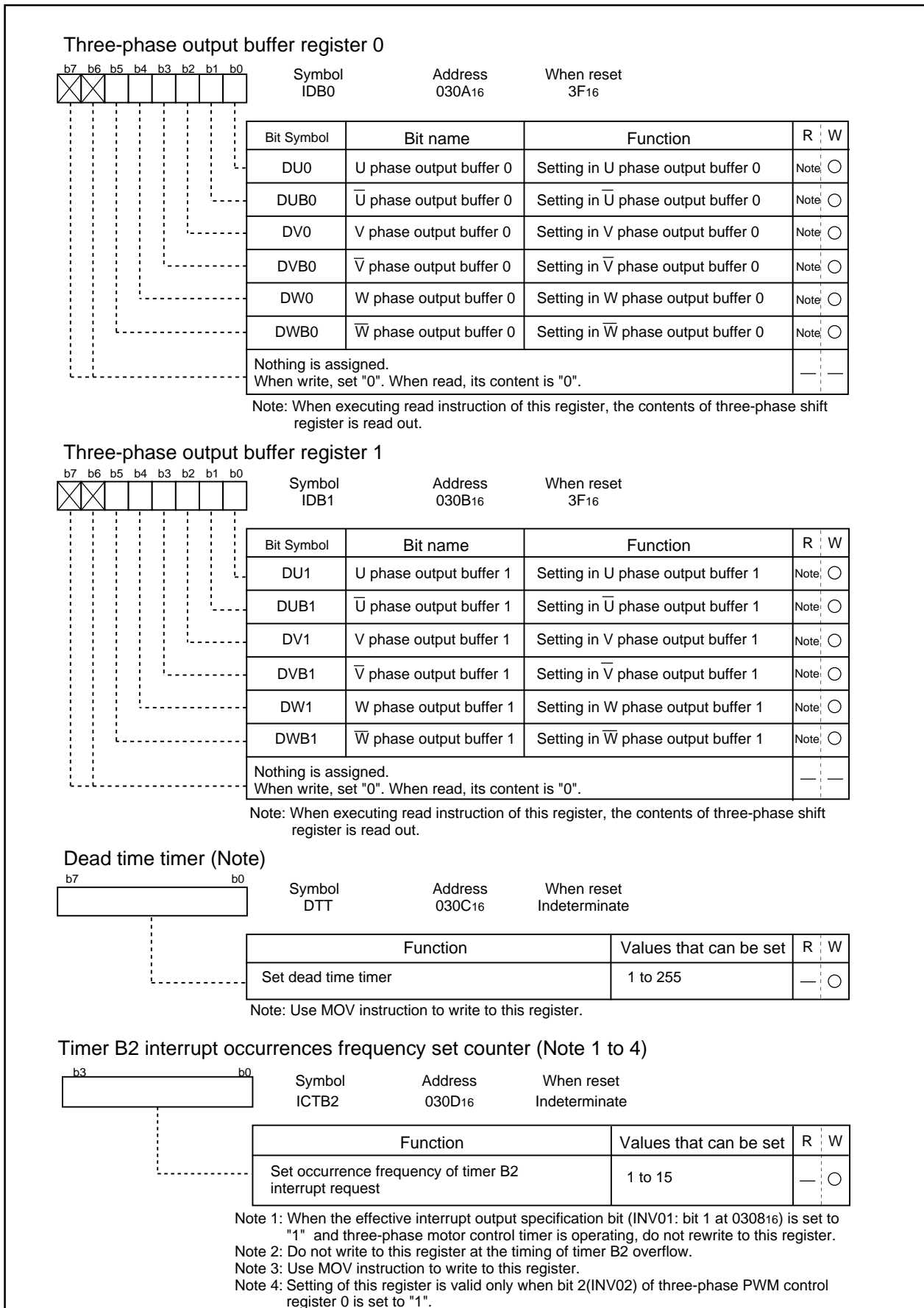


Figure 15.2 Registers related to timers for three-phase motor control

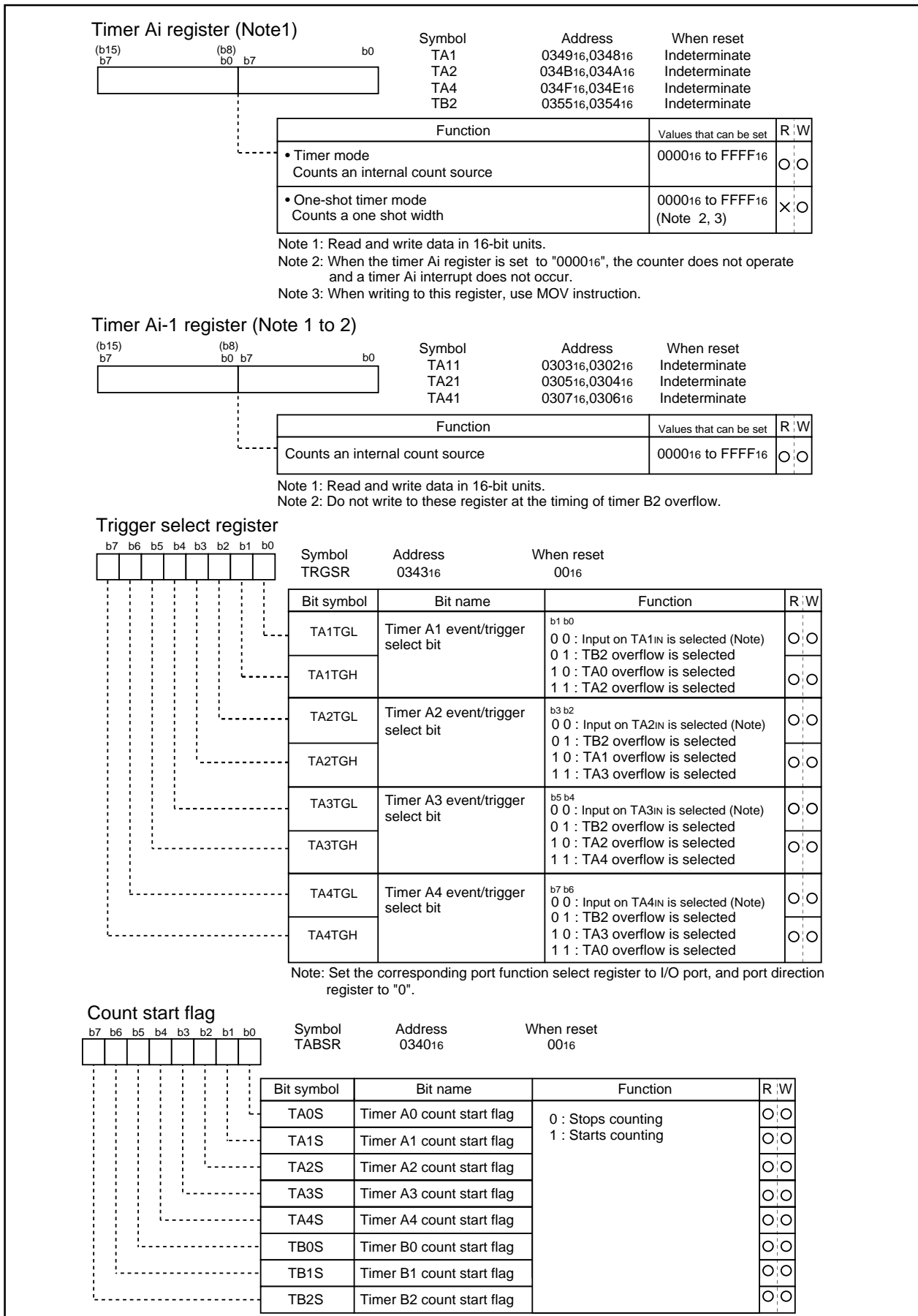


Figure 15.3 Registers related to timers for three-phase motor control

**Three-phase motor driving waveform output mode (three-phase PWM output mode)**

Setting "1" in the mode select bit (bit 2 at 030816) shown in Figure 15.1 causes three-phase PWM output mode that uses four timers A1, A2, A4, and B2 to be selected. As shown in Figure 15.4, set timers A1, A2, and A4 in one-shot timer mode, set the trigger in timer B2, and set timer B2 in timer mode using the respective timer mode registers.

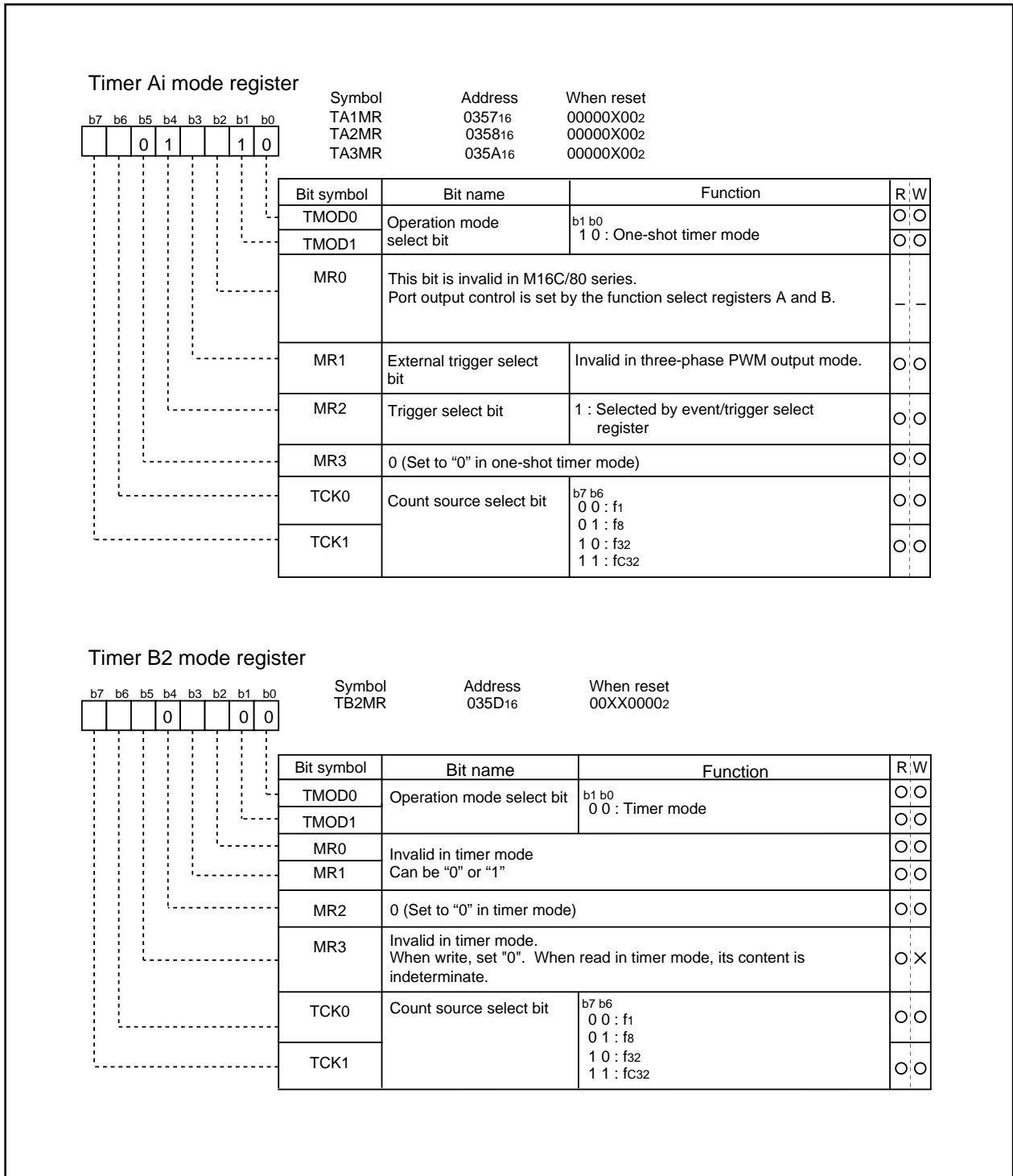


Figure 15.4 Timer mode registers in three-phase PWM output mode



Figure 15.5 shows the block diagram for three-phase waveform mode. In “L” active output polarity in three-phase waveform mode, the positive-phase waveforms (U phase, V phase, and W phase) and negative waveforms ( $\bar{U}$  phase,  $\bar{V}$  phase, and  $\bar{W}$  phase), six waveforms in total, are output from P80, P81, P72, P73, P74, and P75 as active on the “L” level. Of the timers used in this mode, timer A4 controls the U phase and  $\bar{U}$  phase, timer A1 controls the V phase and  $\bar{V}$  phase, and timer A2 controls the W phase and  $\bar{W}$  phase respectively; timer B2 controls the periods of one-shot pulse output from timers A4, A1, and A2. In outputting a waveform, dead time can be set so as to cause the “L” level of the positive waveform output (U phase, V phase, and W phase) not to lap over the “L” level of the negative waveform output ( $\bar{U}$  phase,  $\bar{V}$  phase, and  $\bar{W}$  phase).

To set short circuit time, use three 8-bit timers sharing the reload register for setting dead time. A value from 1 through 255 can be set as the count of the timer for setting dead time. The timer for setting dead time works as a one-shot timer. If a value is written to the dead timer (030C16), the value is written to the reload register shared by the three timers for setting dead time.

Any of the timers for setting dead time takes the value of the reload register into its counter, if a start trigger comes from its corresponding timer, and performs a down count in line with the clock source selected by the dead time timer count source select bit (bit 2 at 030916). The timer can receive another trigger again before the workings due to the previous trigger are completed. In this instance, the timer performs a down count from the reload register's content after its transfer, provoked by the trigger, to the timer for setting dead time.

Since the timer for setting dead time works as a one-shot timer, it starts outputting pulses if a trigger comes; it stops outputting pulses as soon as its content becomes 0016, and waits for the next trigger to come.

The positive waveforms (U phase, V phase, and W phase) and the negative waveforms ( $\bar{U}$  phase,  $\bar{V}$  phase, and  $\bar{W}$  phase) in three-phase waveform mode are output from respective ports by means of setting “1” in the output control bit (bit 3 at 030816). Setting “0” in this bit causes the ports to be the high-impedance state. This bit can be set to “0” not only by use of the applicable instruction, but by entering a falling edge in the  $\overline{\text{NMI}}$  terminal or by resetting. Also, if “1” is set in the positive and negative phases concurrent L output disable function enable bit (bit 4 at 030816) causes one of the pairs of U phase and  $\bar{U}$  phase, V phase and  $\bar{V}$  phase, and W phase and  $\bar{W}$  phase concurrently go to “L”, as a result, the output control bit becomes the high-impedance state.

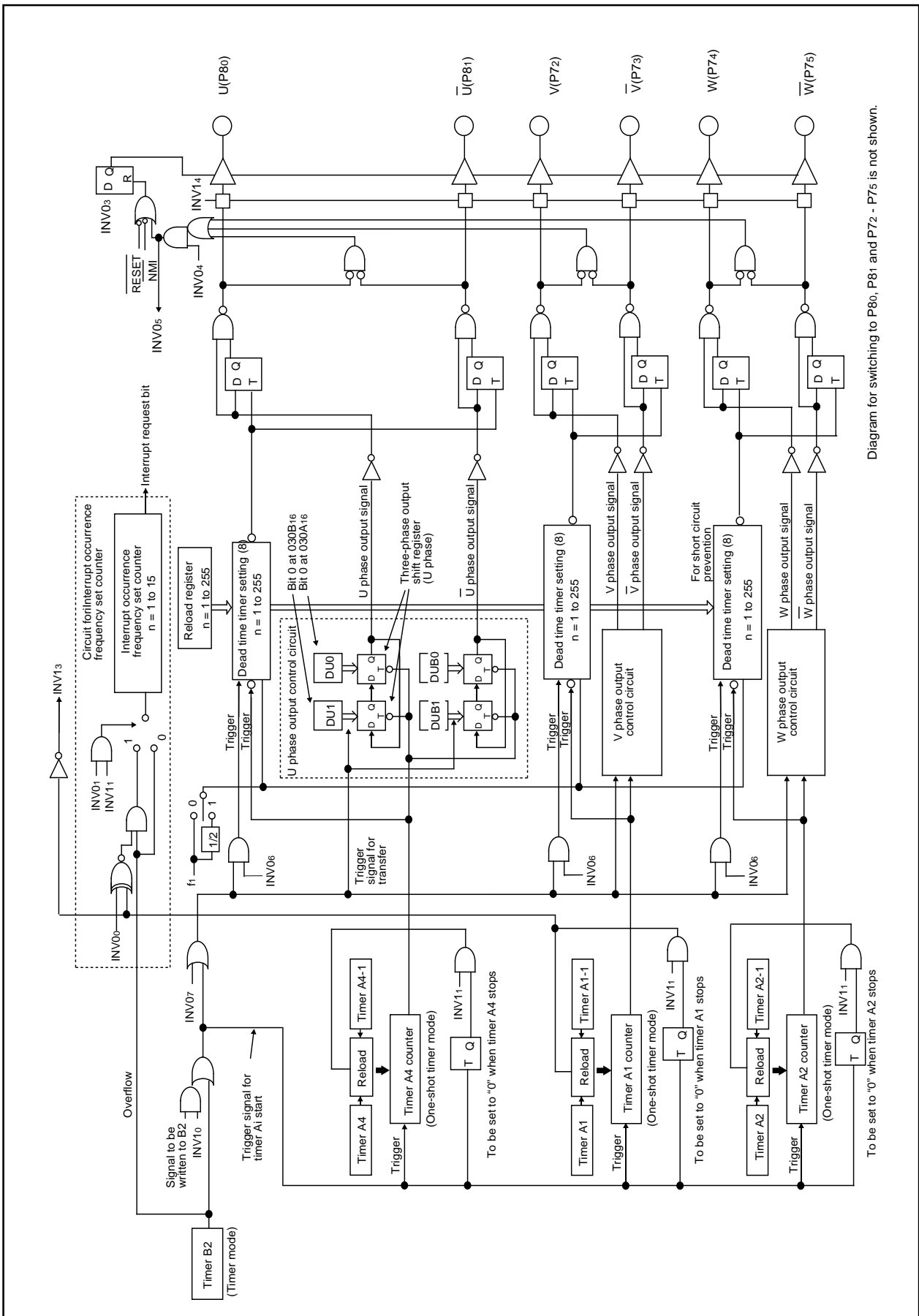


Diagram for switching to P80, P81 and P72 - P75 is not shown.

Figure 15.5 Block diagram for three-phase waveform mode

### Triangular wave modulation

To generate a PWM waveform of triangular wave modulation, set "0" in the modulation mode select bit (bit 6 at 0308<sub>16</sub>). Also, set "1" in the timers A4-1, A1-1, A2-1 control bit (bit 1 at 0309<sub>16</sub>). In this mode, each of timers A4, A1, and A2 has two timer registers, and alternately reloads the timer register's content to the counter every time timer B2 counter's content becomes 0000<sub>16</sub>. If "0" is set to the effective interrupt output specification bit (bit 1 at 0308<sub>16</sub>), the frequency of interrupt requests that occur every time the timer B2 counter's value becomes 0000<sub>16</sub> can be set by use of the timer B2 counter (030D<sub>16</sub>) for setting the frequency of interrupt occurrences. The frequency of occurrences is given by (setting; setting  $\pi$  0).

Setting "1" in the effective interrupt output specification bit (bit 1 at 0308<sub>16</sub>) provides the means to choose which value of the timer A1 reload control signal to use, "0" or "1", to cause timer B2's interrupt request to occur. To make this selection, use the effective interrupt output polarity selection bit (bit 0 at 0308<sub>16</sub>).

An example of U phase waveform is shown in Figure 15.6, and the description of waveform output workings is given below. Set "1" in DU0 (bit 0 at 030A<sub>16</sub>). And set "0" in DUB0 (bit 1 at 030A<sub>16</sub>). In addition, set "0" in DU1 (bit 0 at 030B<sub>16</sub>) and set "1" in DUB1 (bit 1 at 030B<sub>16</sub>). Also, set "0" in the effective interrupt output specification bit (bit 1 at 0308<sub>16</sub>) to set a value in the timer B2 interrupt occurrence frequency set counter. By this setting, a timer B2 interrupt occurs when the timer B2 counter's content becomes 0000<sub>16</sub> as many as (setting) times. Furthermore, set "1" in the effective interrupt output specification bit (bit 1 at 0308<sub>16</sub>), set in the effective interrupt polarity select bit (bit 0 at 0308<sub>16</sub>) and set "1" in the interrupt occurrence frequency set counter (030D<sub>16</sub>). These settings cause a timer B2 interrupt to occur every other interval when the U phase output goes to "H".

When the timer B2 counter's content becomes 0000<sub>16</sub>, timer A4 starts outputting one-shot pulses. In this instance, the content of DU1 (bit 0 at 030B<sub>16</sub>) and that of DU0 (bit 0 at 030A<sub>16</sub>) are set in the three-phase output shift register (U phase), the content of DUB1 (bit 1 at 030B<sub>16</sub>) and that of DUB0 (bit 1 at 030A<sub>16</sub>) are set in the three-phase shift register ( $\bar{U}$  phase). After triangular wave modulation mode is selected, however, no setting is made in the shift register even though the timer B2 counter's content becomes 0000<sub>16</sub>.

The value of DU0 and that of DUB0 are output to the U terminal (P80) and to the  $\bar{U}$  terminal (P81) respectively. When the timer A4 counter counts the value written to timer A4 (034F<sub>16</sub>, 034E<sub>16</sub>) and when timer A4 finishes outputting one-shot pulses, the three-phase shift register's content is shifted one position, and the value of DU1 and that of DUB1 are output to the U phase output signal and to  $\bar{U}$  phase output signal respectively. At this time, one-shot pulses are output from the timer for setting dead time used for setting the time over which the "L" level of the U phase waveform doesn't lap over the "L" level of the  $\bar{U}$  phase waveform, which has the opposite phase of the former. The U phase waveform output that started from the "H" level keeps its level until the timer for setting dead time finishes outputting one-shot pulses even though the three-phase output shift register's content changes from "1" to "0" by the effect of the one-shot pulses. When the timer for setting dead time finishes outputting one-shot pulses, "0" already shifted in the three-phase shift register goes effective, and the U phase waveform changes to the "L" level. When the timer B2 counter's content becomes 0000<sub>16</sub>, the timer A4 counter starts counting the value written to timer A4-1 (0307<sub>16</sub>, 0306<sub>16</sub>), and starts outputting one-shot pulses. When timer A4 finishes outputting one-shot pulses, the three-phase shift register's content is shifted one position, but if the three-phase output shift register's content changes from "0" to "1" as a result of the shift, the output level changes from "L" to "H" without waiting for the timer for setting dead time to finish outputting one-shot pulses. A U phase waveform is generated by these workings repeatedly. With the exception that the three-phase output shift register on the U phase side is used, the workings in generating a U phase waveform, which has the opposite phase of the U phase waveform, are the same as in generating a U

phase waveform. In this way, a waveform can be picked up from the applicable terminal in a manner in which the "L" level of the U phase waveform doesn't lap over that of the U phase waveform, which has the opposite phase of the U phase waveform. The width of the "L" level too can be adjusted by varying the values of timer B2, timer A4, and timer A4-1. In dealing with the V and W phases, and  $\bar{V}$  and  $\bar{W}$  phases, the latter are of opposite phase of the former, have the corresponding timers work similarly to dealing with the U and  $\bar{U}$  phases to generate an intended waveform.

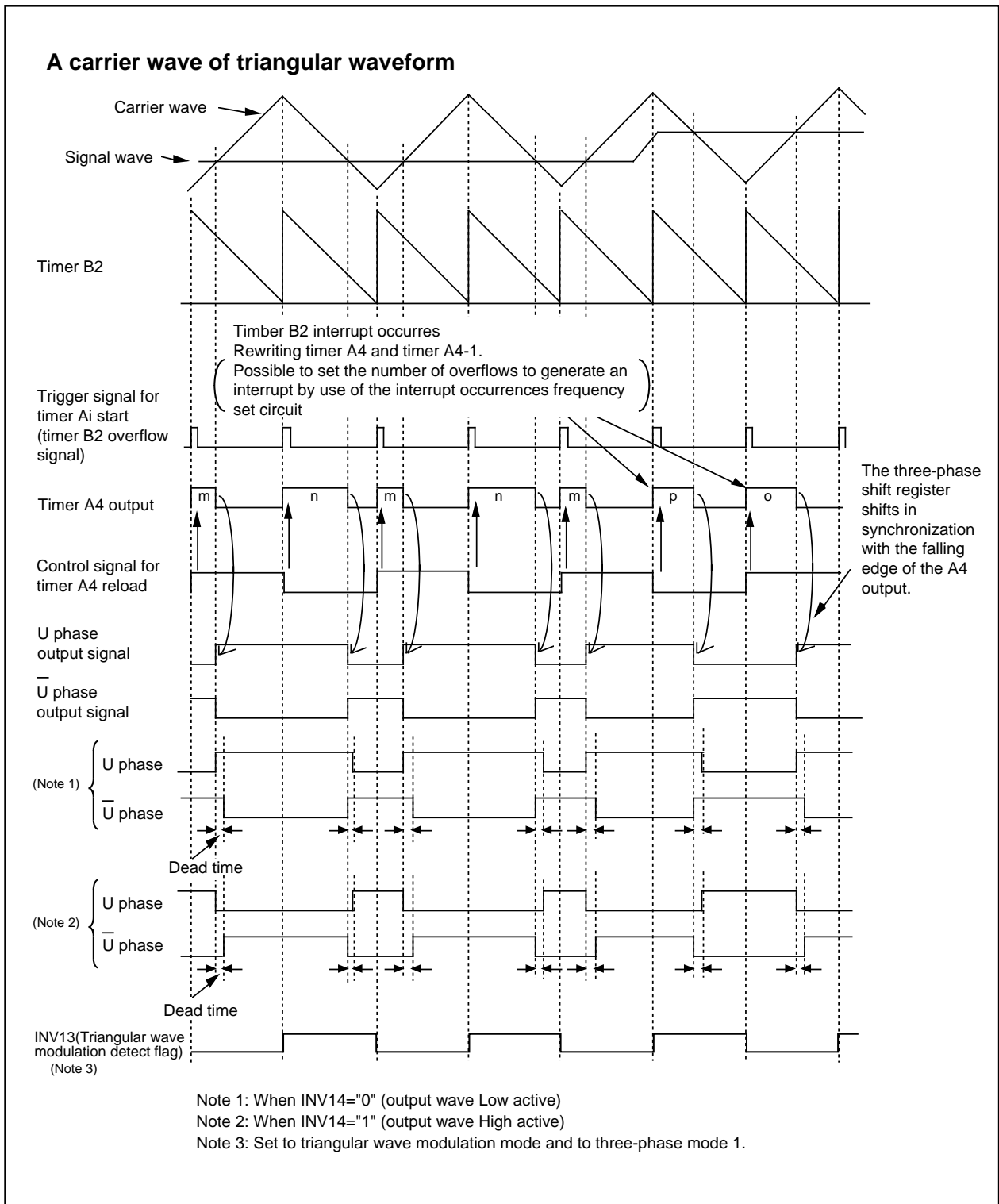


Figure 15.6 Timing chart of operation (1)

Assigning certain values to DU0 (bit 0 at 030A16) and DUB0 (bit 1 at 030A16), and to DU1 (bit 0 at 030B16) and DUB1 (bit 1 at 030B16) allows you to output the waveforms as shown in Figure 15.7, that is, to output the U phase alone, to fix  $\bar{U}$  phase to "H", to fix the U phase to "H," or to output the  $\bar{U}$  phase alone.

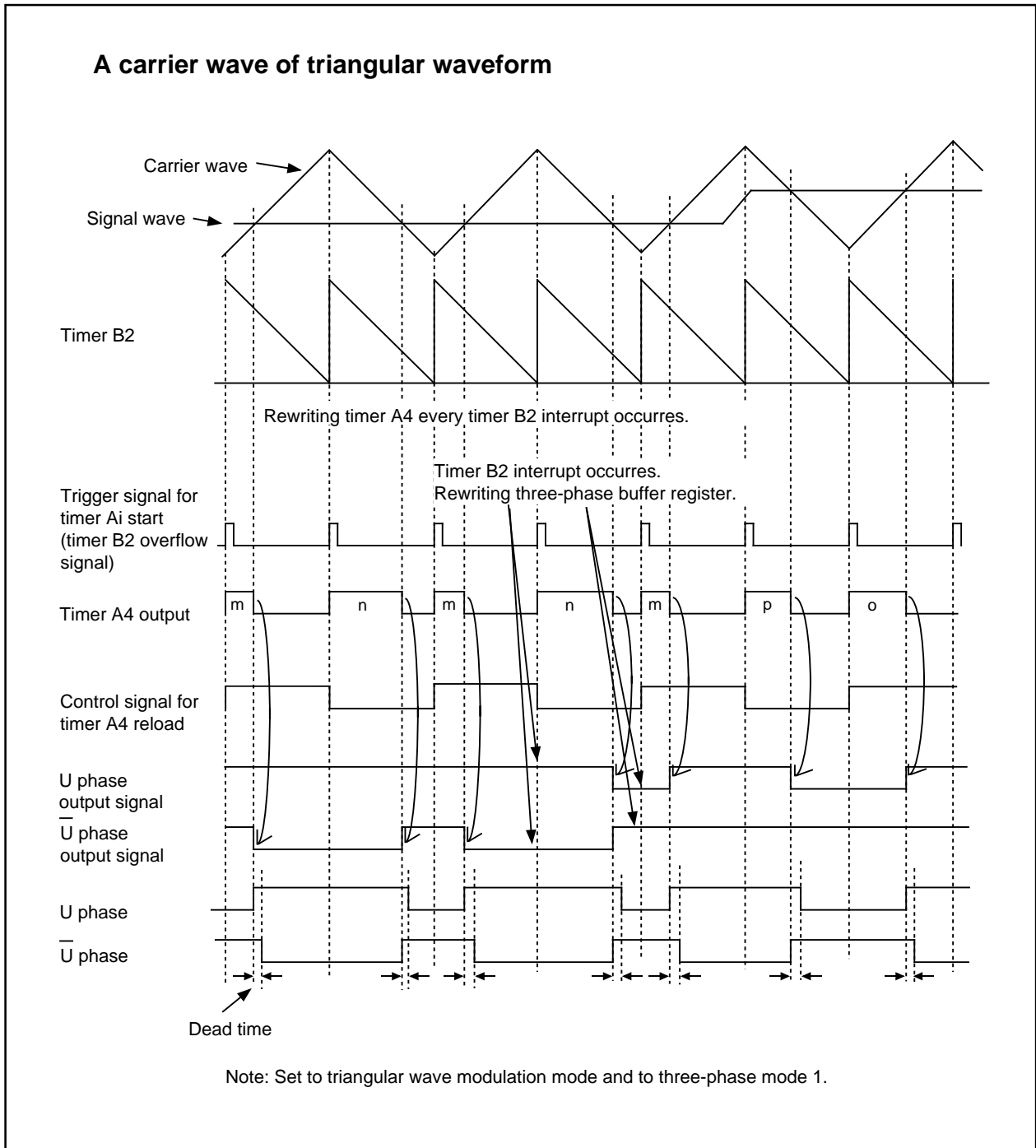


Figure 15.7 Timing chart of operation (2)

### Sawtooth modulation

To generate a PWM waveform of sawtooth wave modulation, set "1" in the modulation mode select bit (bit 6 at 0308<sub>16</sub>). Also, set "0" in the timers A4, A1, and A2-1 control bit (bit 1 at 0309<sub>16</sub>). In this mode, the timer registers of timers A4, A1, and of A2 comprise conventional timers A4, A1, and A2 alone, and reload the corresponding timer register's content to the counter every time the timer B2 counter's content becomes 0000<sub>16</sub>. The effective interrupt output specification bit (bit 1 at 0308<sub>16</sub>) and the effective interrupt output polarity select bit (bit 0 at 0308<sub>16</sub>) go nullified.

An example of U phase waveform is shown in Figure 15.8, and the description of waveform output workings is given below. Set "1" in DU0 (bit 0 at 030A<sub>16</sub>), and set "0" in DUB0 (bit 1 at 030A<sub>16</sub>). In addition, set "0" in DU1 (bit 0 at 030B<sub>16</sub>) and set "1" in DUB1 (bit 1 at 030B<sub>16</sub>).

When the timer B2 counter's content becomes 0000<sub>16</sub>, timer B2 generates an interrupt, and timer A4 starts outputting one-shot pulses at the same time. In this instance, the contents of the three-phase buffer registers DU1 and DU0 are set in the three-phase output shift register (U phase), and the contents of DUB1 and DUB0 are set in the three-phase output register (U phase). After this, the three-phase buffer register's content is set in the three-phase shift register every time the timer B2 counter's content becomes 0000<sub>16</sub>.

The value of DU0 and that of DUB0 are output to the U terminal (P80) and to the  $\bar{U}$  terminal (P81) respectively. When the timer A4 counter counts the value written to timer A4 (034F<sub>16</sub>, 034E<sub>16</sub>) and when timer A4 finishes outputting one-shot pulses, the three-phase output shift register's content is shifted one position, and the value of DU1 and that of DUB1 are output to the U phase output signal and to the  $\bar{U}$  output signal respectively. At this time, one-shot pulses are output from the timer for setting dead time used for setting the time over which the "L" level of the U phase waveform doesn't lap over the "L" level of the  $\bar{U}$  phase waveform, which has the opposite phase of the former. The U phase waveform output that started from the "H" level keeps its level until the timer for setting dead time finishes outputting one-shot pulses even though the three-phase output shift register's content changes from "1" to "0" by the effect of the one-shot pulses. When the timer for setting dead time finishes outputting one-shot pulses, 0 already shifted in the three-phase shift register goes effective, and the U phase waveform changes to the "L" level. When the timer B2 counter's content becomes 0000<sub>16</sub>, the contents of the three-phase buffer registers DU1 and DU0 are set in the three-phase shift register (U phase), and the contents of DUB1 and DUB0 are set in the three-phase shift register ( $\bar{U}$  phase) again.

A U phase waveform is generated by these workings repeatedly. With the exception that the three-phase output shift register on the  $\bar{U}$  phase side is used, the workings in generating a  $\bar{U}$  phase waveform, which has the opposite phase of the U phase waveform, are the same as in generating a U phase waveform. In this way, a waveform can be picked up from the applicable terminal in a manner in which the "L" level of the U phase waveform doesn't lap over that of the U phase waveform, which has the opposite phase of the U phase waveform. The width of the "L" level too can be adjusted by varying the values of timer B2 and timer A4. In dealing with the V and W phases, and  $\bar{V}$  and  $\bar{W}$  phases, the latter are of opposite phase of the former, have the corresponding timers work similarly to dealing with the U and  $\bar{U}$  phases to generate an intended waveform.

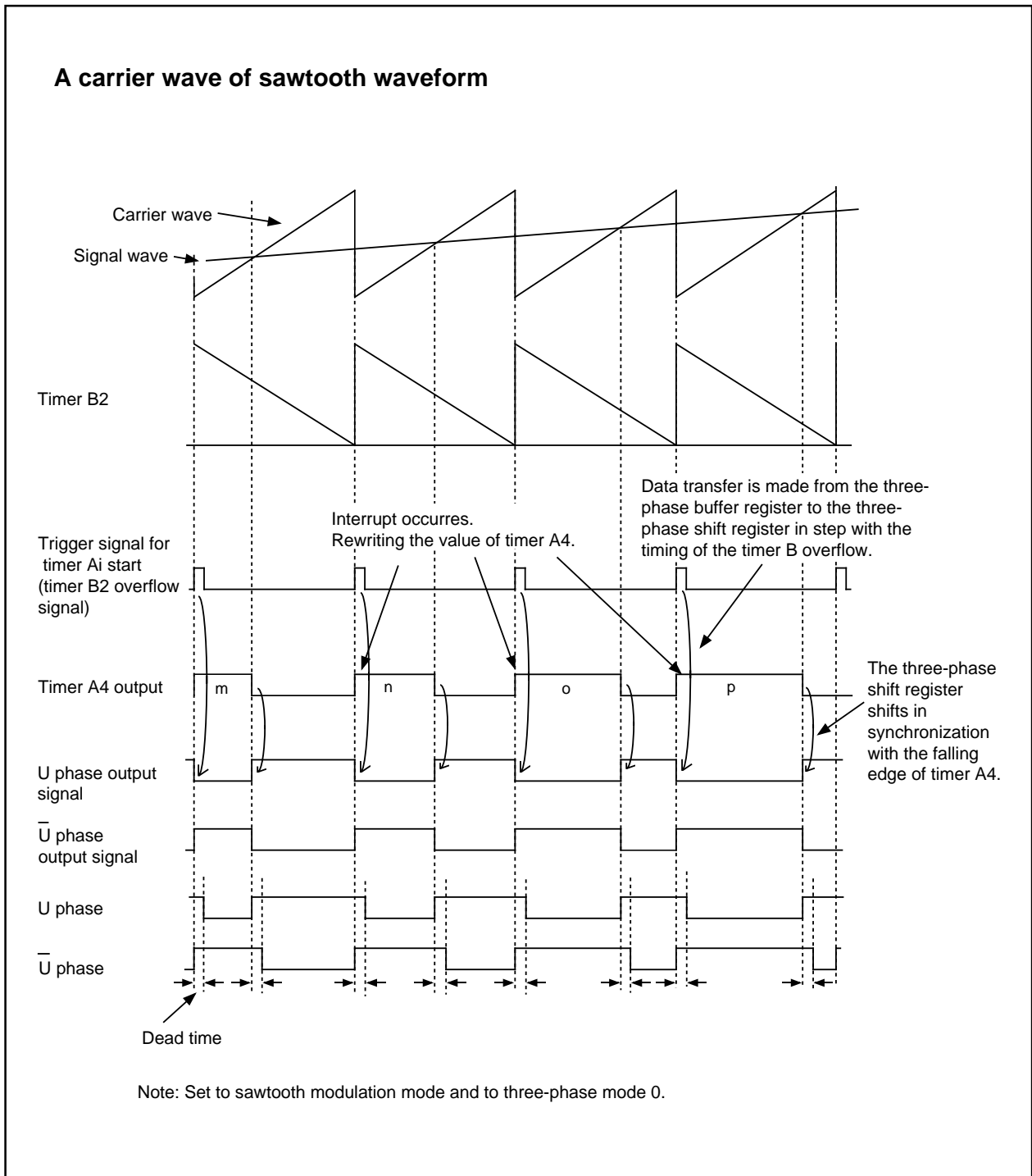


Figure 15.8 Timing chart of operation (3)

Setting "1" both in DUB0 and in DUB1 provides a means to output the U phase alone and to fix the  $\bar{U}$  phase output to "H" as shown in Figure 15.9.

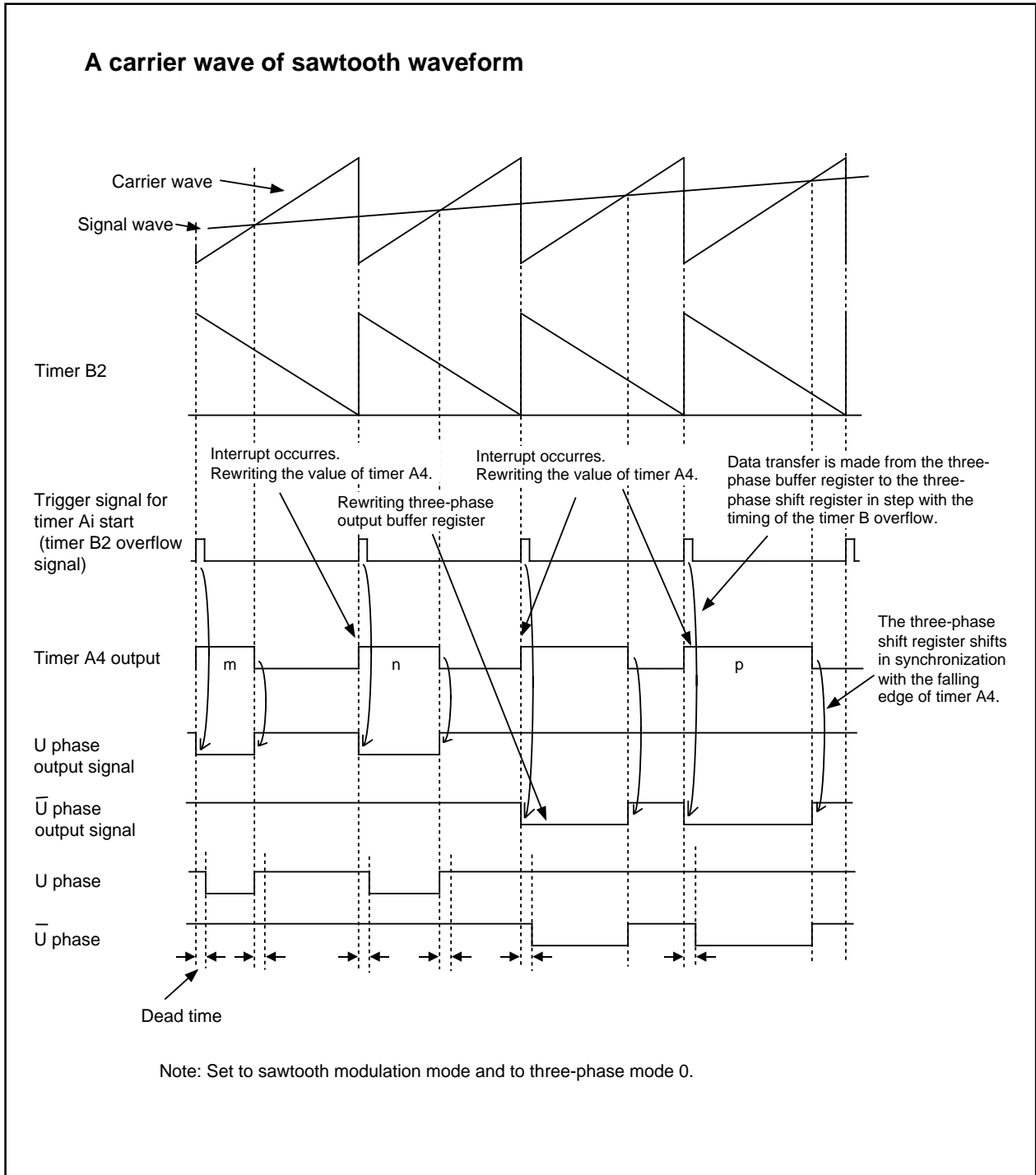


Figure 15.9 Timing chart of operation (4)



## 16. Serial I/O

Serial I/O is configured as five channels: UART0 to UART4.

### UART0 to 4

UART0 to UART4 each have an exclusive timer to generate a transfer clock, so they operate independently of each other.

Figures 16.1 and 16.2 show the block diagram of UART<sub>i</sub> (i=0 to 4). Figures 16.3 and 16.4 show the block diagram of the transmit/receive unit.

UART<sub>i</sub> has two operation modes: a clock synchronous serial I/O mode and a clock asynchronous serial I/O mode (UART mode). The contents of the serial I/O mode select bits (bits 0 to 2 at addresses 0360<sub>16</sub>, 0368<sub>16</sub>, 0338<sub>16</sub>, 0328<sub>16</sub> and 02F8<sub>16</sub>) determine whether UART<sub>i</sub> is used as a clock synchronous serial I/O or as a UART.

Although a few functions are different, UART0 to UART4 have almost the same functions.

UART2 to UART4, in particular, are compliant with the SIM interface with some extra settings added in clock-asynchronous serial I/O mode (Note). It also has the bus collision detection function that generates an interrupt request if the TxD pin and the RxD pin are different in level.

Table 16.1 shows the comparison of functions of UART0 to UART4, and Figures 16.5 through 16.11 show the registers related to UART<sub>i</sub>.

Note: SIM : Subscriber Identity Module

**Table 16.1 Comparison of functions of UART0 to UART4**

Function	UART0	UART1	UART2	UART3	UART4
CLK polarity selection	Possible <sup>(Note 1)</sup>	Possible <sup>(Note 1)</sup>	Possible <sup>(Note 1)</sup>	Possible <sup>(Note 1)</sup>	Possible <sup>(Note 1)</sup>
LSB first / MSB first selection	Possible <sup>(Note 1)</sup>	Possible <sup>(Note 1)</sup>	Possible <sup>(Note 2)</sup>	Possible <sup>(Note 2)</sup>	Possible <sup>(Note 2)</sup>
Continuous receive mode selection	Possible <sup>(Note 1)</sup>	Possible <sup>(Note 1)</sup>	Possible <sup>(Note 1)</sup>	Possible <sup>(Note 1)</sup>	Possible <sup>(Note 1)</sup>
Transfer clock output from multiple pins selection	Impossible	Possible <sup>(Note 1)</sup>	Impossible	Impossible	Impossible
Separate $\overline{\text{CTS}}/\overline{\text{RTS}}$ pins	Possible	Impossible	Impossible	Impossible	Impossible
Serial data logic switch	Impossible	Impossible	Possible <sup>(Note 4)</sup>	Possible <sup>(Note 4)</sup>	Possible <sup>(Note 4)</sup>
Sleep mode selection	Possible <sup>(Note 3)</sup>	Possible <sup>(Note 3)</sup>	Impossible	Impossible	Impossible
TxD, RxD I/O polarity switch	Impossible	Impossible	Possible	Possible	Possible
TxD, RxD port output format	CMOS output	CMOS output	N-channel open drain output	CMOS output	CMOS output
Parity error signal output	Impossible	Impossible	Possible <sup>(Note 4)</sup>	Possible <sup>(Note 4)</sup>	Possible <sup>(Note 4)</sup>
Bus collision detection	Impossible	Impossible	Possible	Possible	Possible

Note 1: Only when clock synchronous serial I/O mode.

Note 2: Only when clock synchronous serial I/O mode and 8-bit UART mode.

Note 3: Only when UART mode.

Note 4: Using for SIM interface.

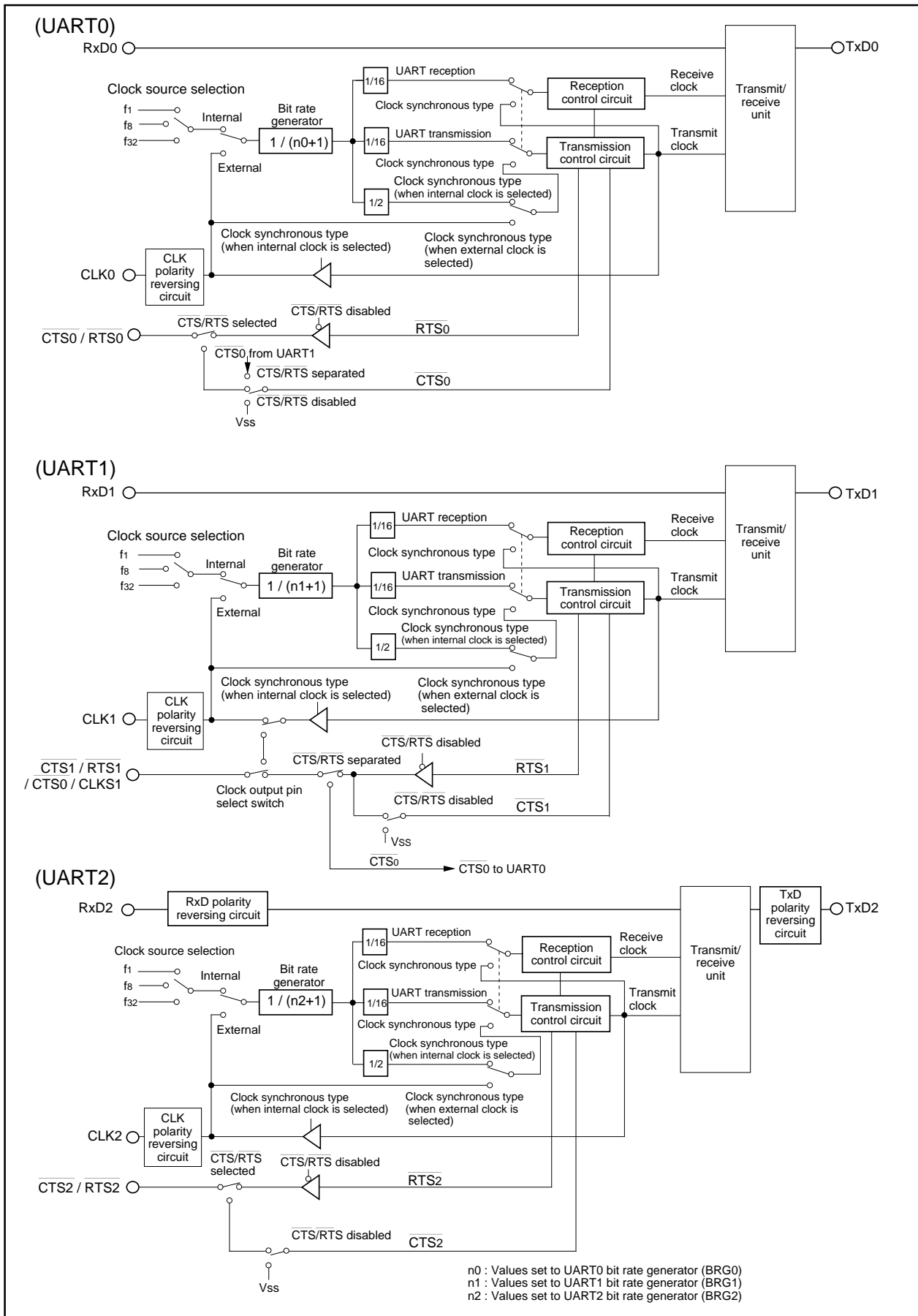


Figure 16.1 Block diagram of UARTi (i = 0 to 2)

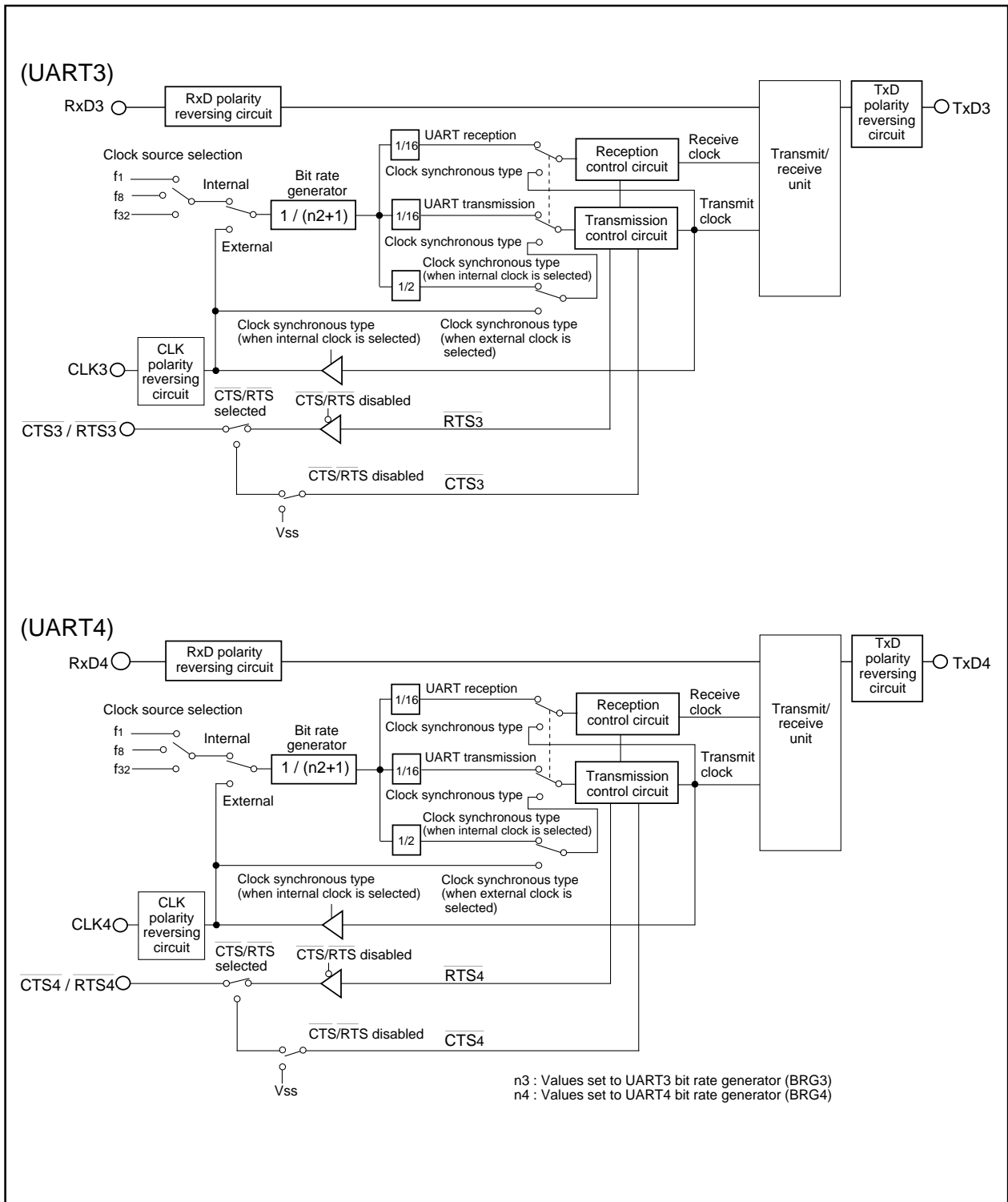


Figure 16.2 Block diagram of UARTi (i = 3, 4)

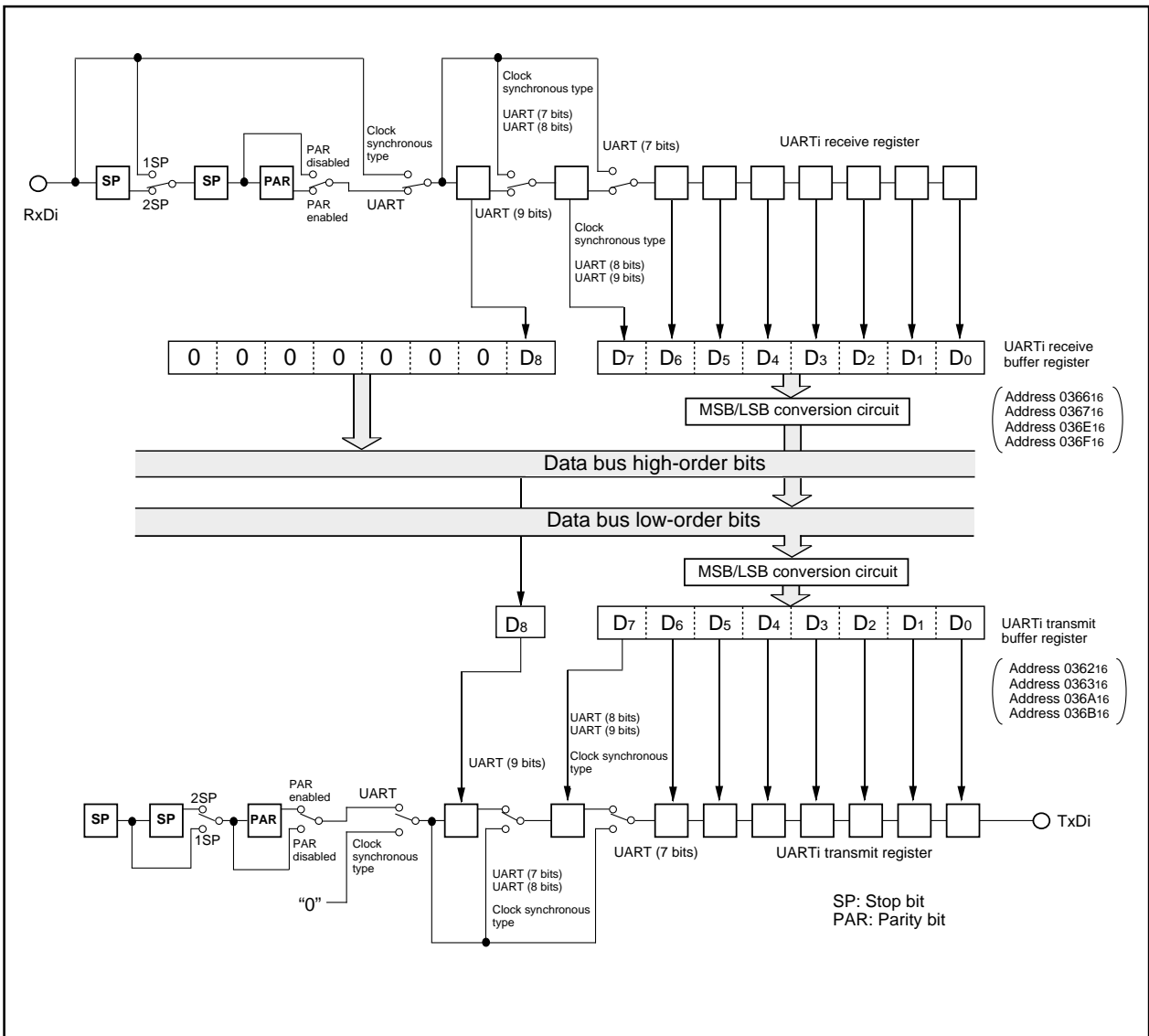


Figure 16.3 Block diagram of UARTi (i = 0, 1) transmit/receive unit

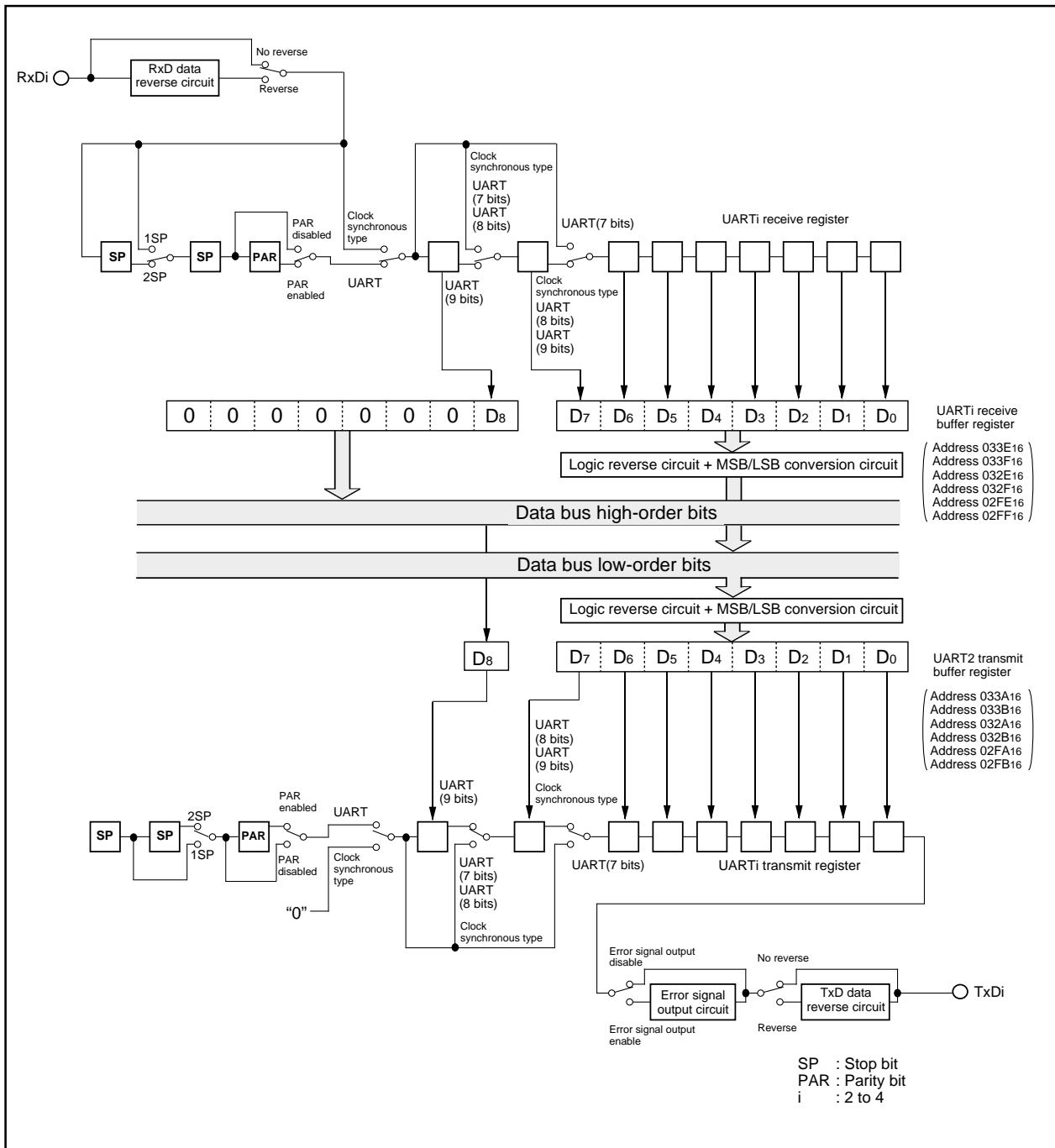


Figure 16.4 Block diagram of UART<sub>i</sub> (i = 2 to 4) transmit/receive unit

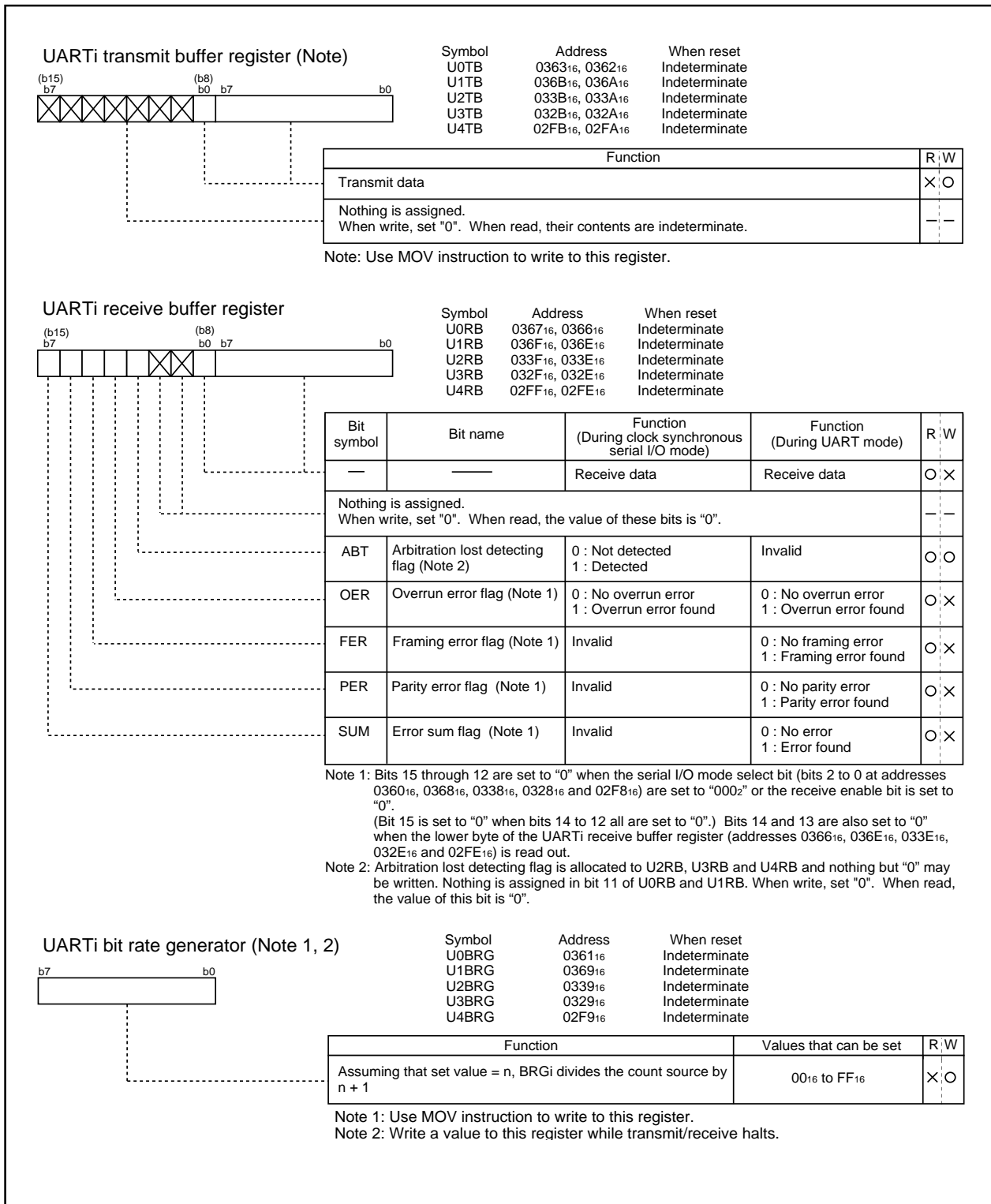


Figure 16.5 Serial I/O-related registers (1)

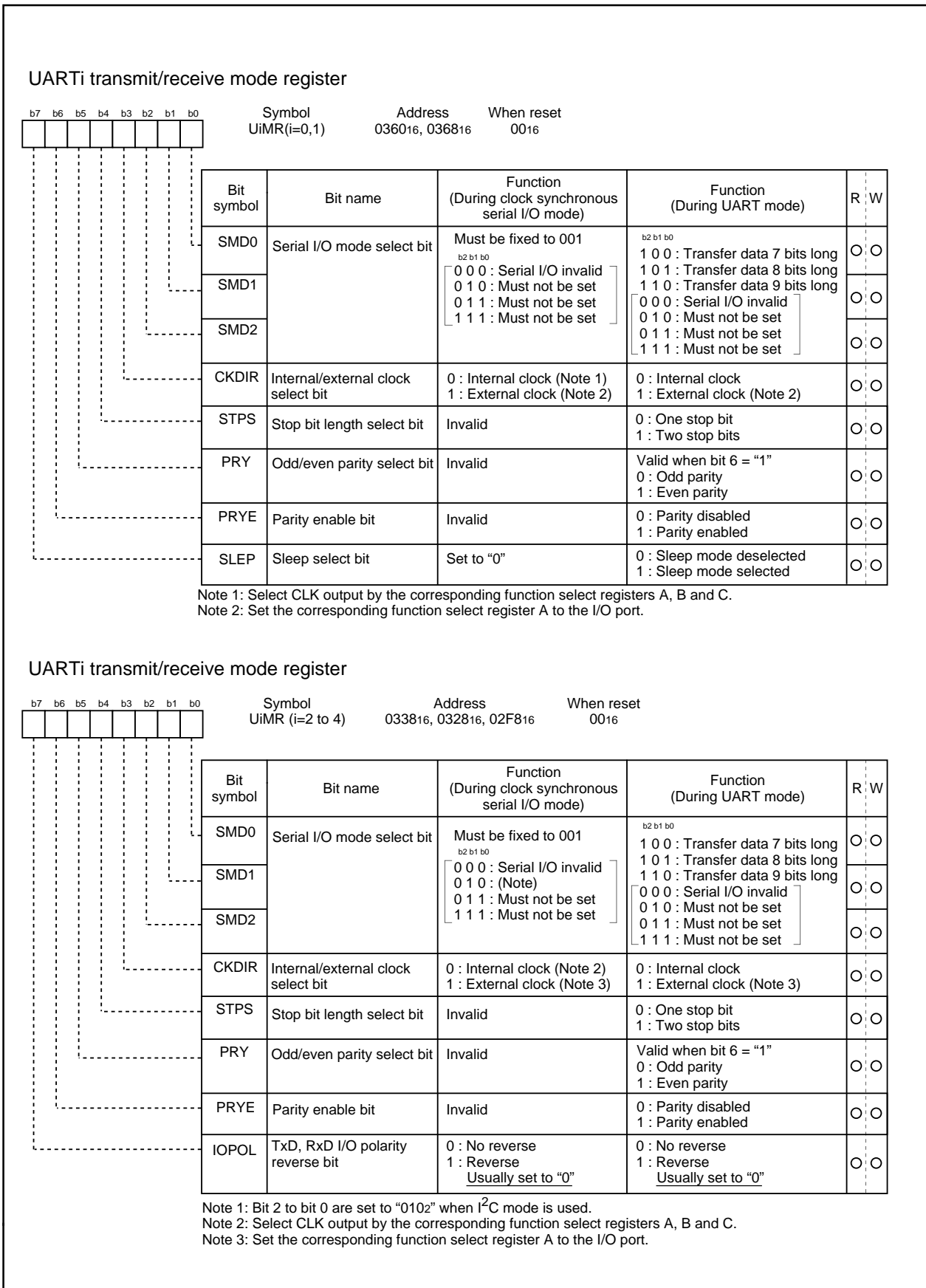


Figure 16.6 Serial I/O-related registers (2)

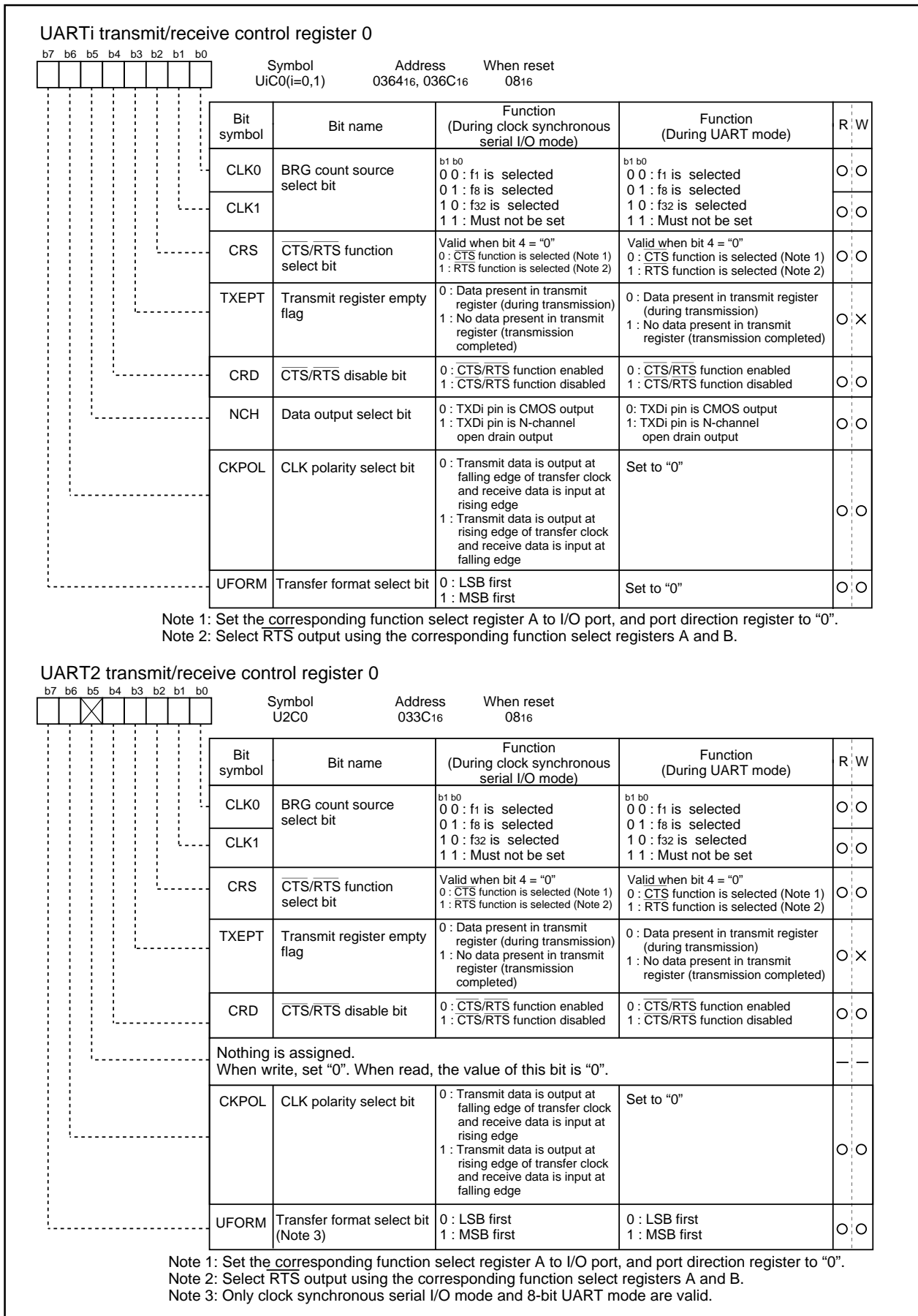


Figure 16.7 Serial I/O-related registers (3)



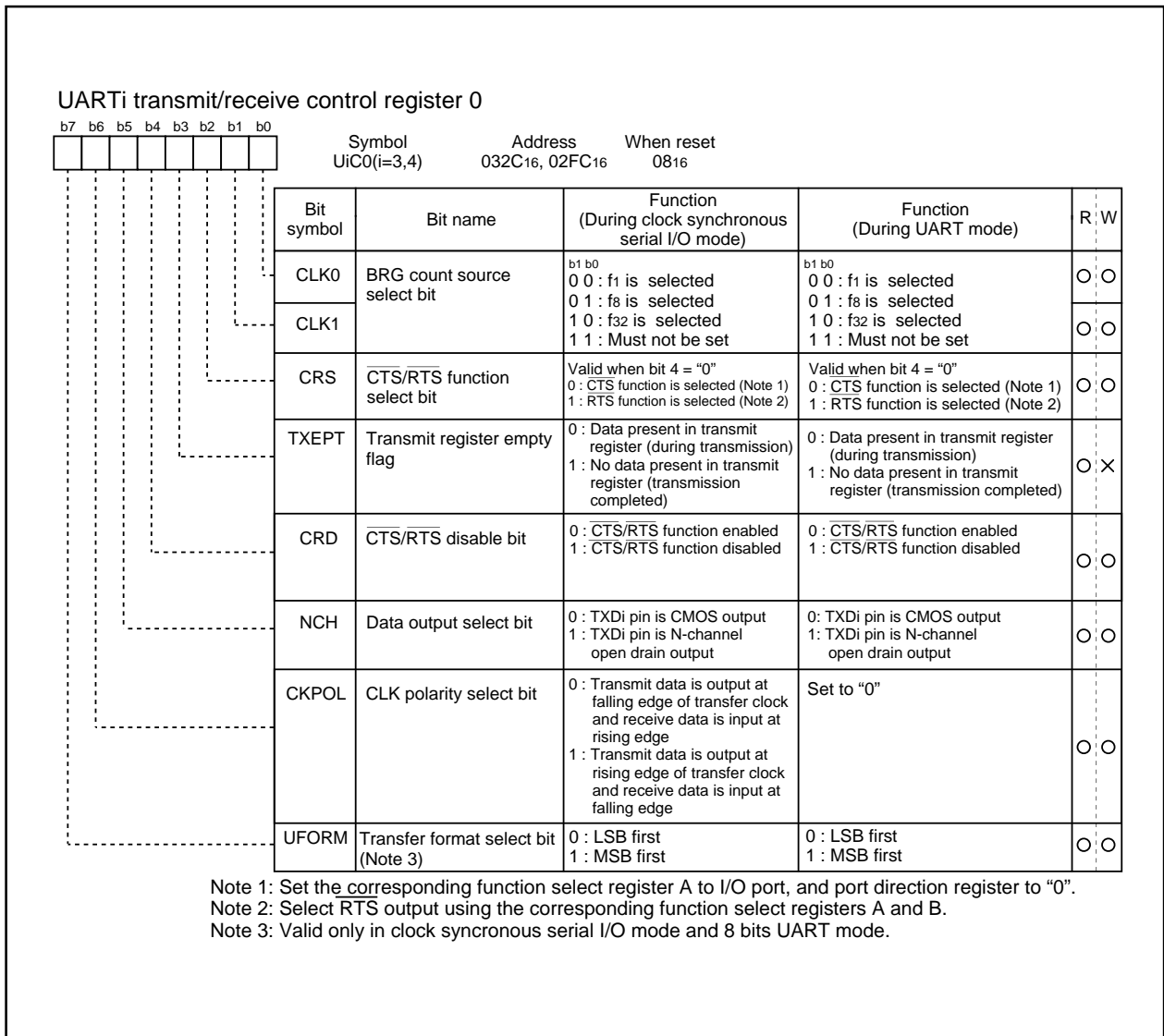


Figure 16.8 Serial I/O-related registers (4)

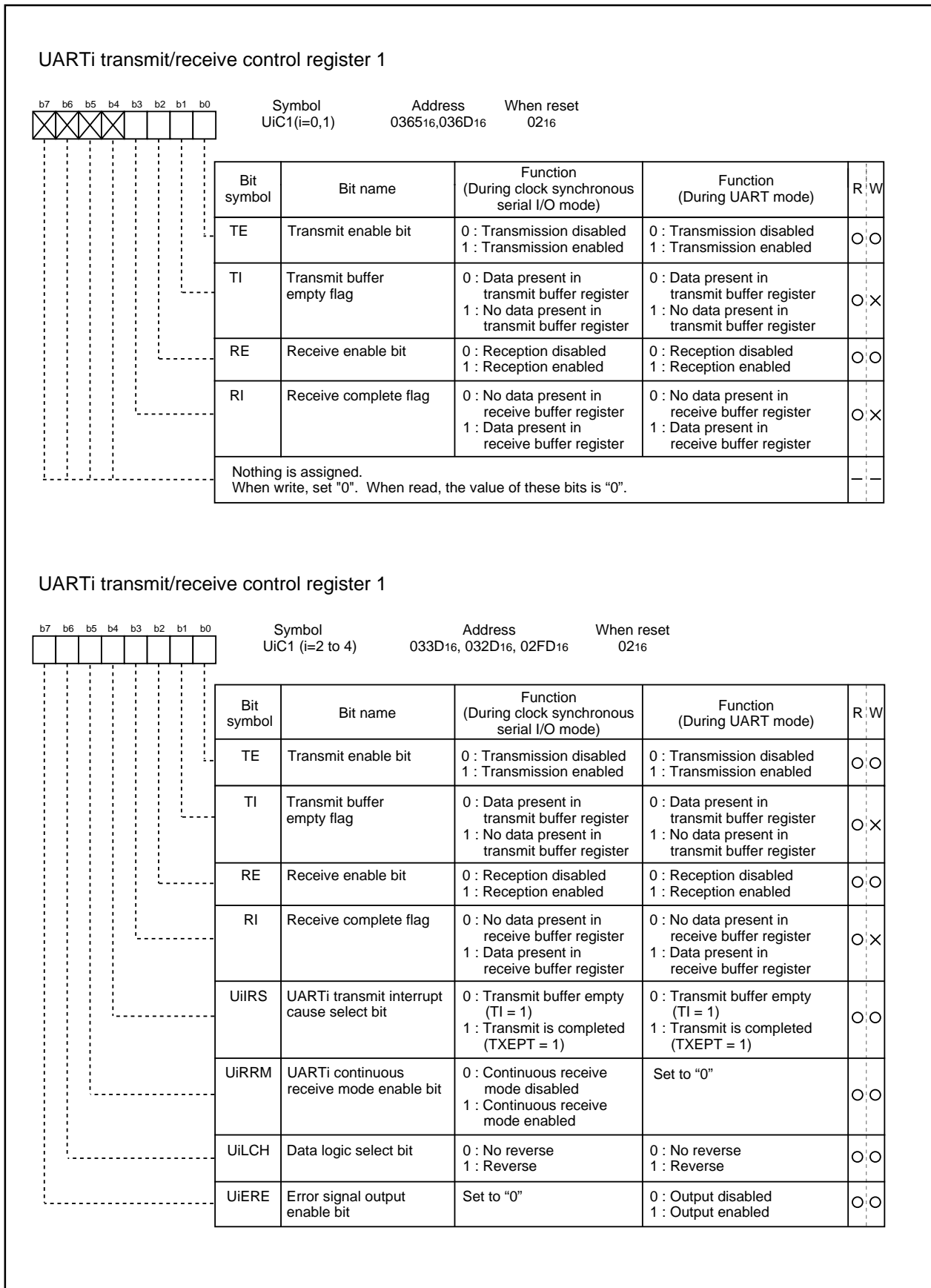


Figure 16.9 Serial I/O-related registers (5)

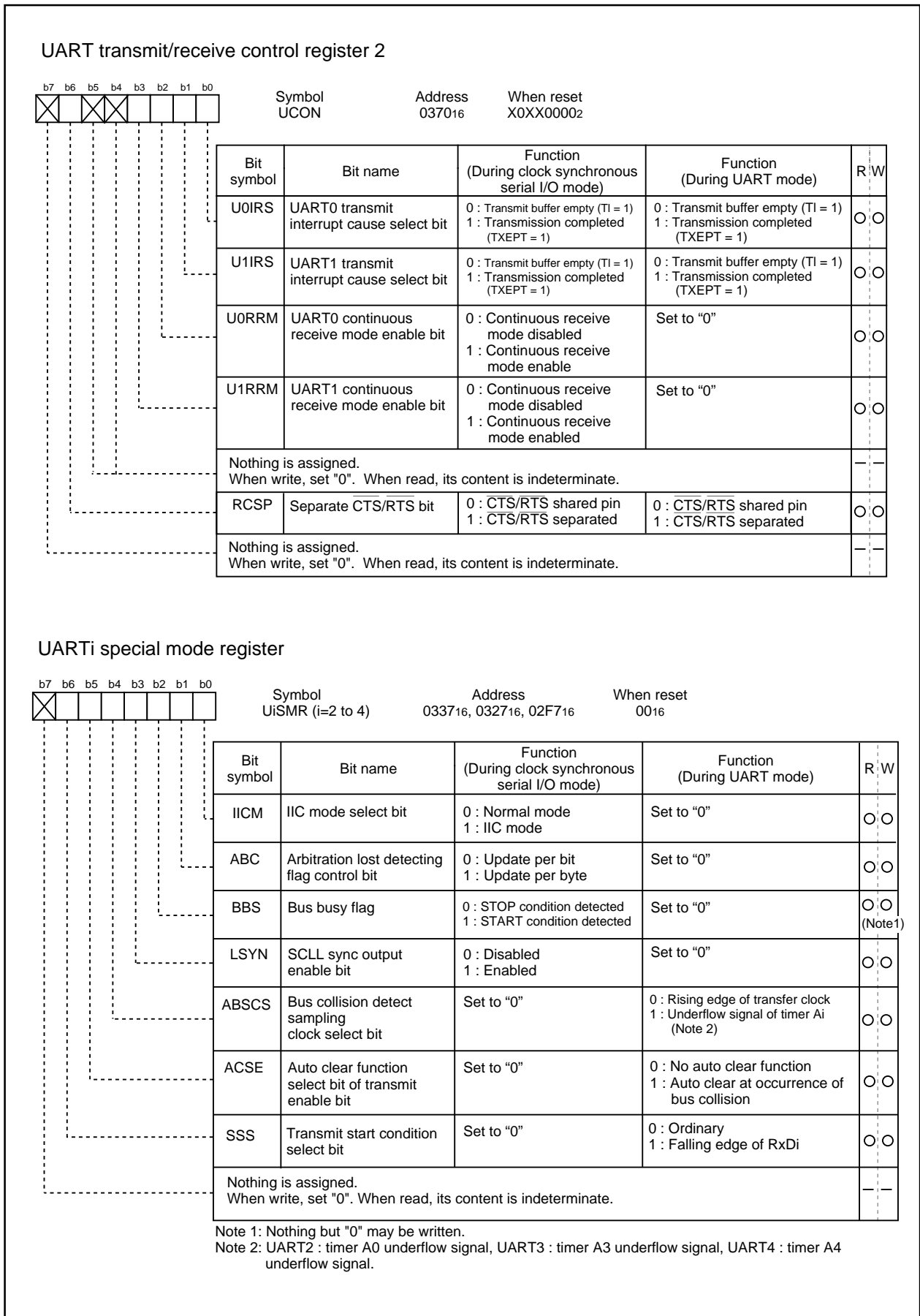


Figure 16.10 Serial I/O-related registers (6)

UARTi special mode register 2

Bit symbol	Bit name	Function	R	W
IICM2	IIC mode select bit 2	0 : NACK/ACK interrupt DMA source - ACK Transfer to receive buffer at the rising edge of last bit of receive clock Receive interrupt is occurred at the rising edge of last bit of receive clock 1 : UART transfer/receive interrupt DMA source - UART receive Transfer to receive buffer at the falling edge of last bit of receive clock Receive interrupt is occurred at the falling edge of last bit of receive clock	○	○
CSC	Clock synchronous bit	0 : Disabled 1 : Enabled	○	○
SWC	SCL wait output bit	0 : Disabled 1 : Enabled	○	○
ALS	SDA output stop flag	0 : Disabled 1 : Enabled	○	○
STC	UARTi initialize bit	0 : Disabled 1 : Enabled	○	○
SWC2	SCL wait output bit 2	0 : UARTi clock 1 : 0 output	○	○
SDHI	SDA output inhibit bit	0 : Enabled 1 : Disabled (high impedance)	○	○
SHTC	Start/stop condition control bit	Must set to "1" in selecting IIC mode.	○	○

b7 b6 b5 b4 b3 b2 b1 b0  
 Symbol: UiSMR2 (i=2 to 4)  
 Address: 033616, 032616, 02F616  
 When reset: 0016

Figure 16.11 Serial I/O-related registers (7)

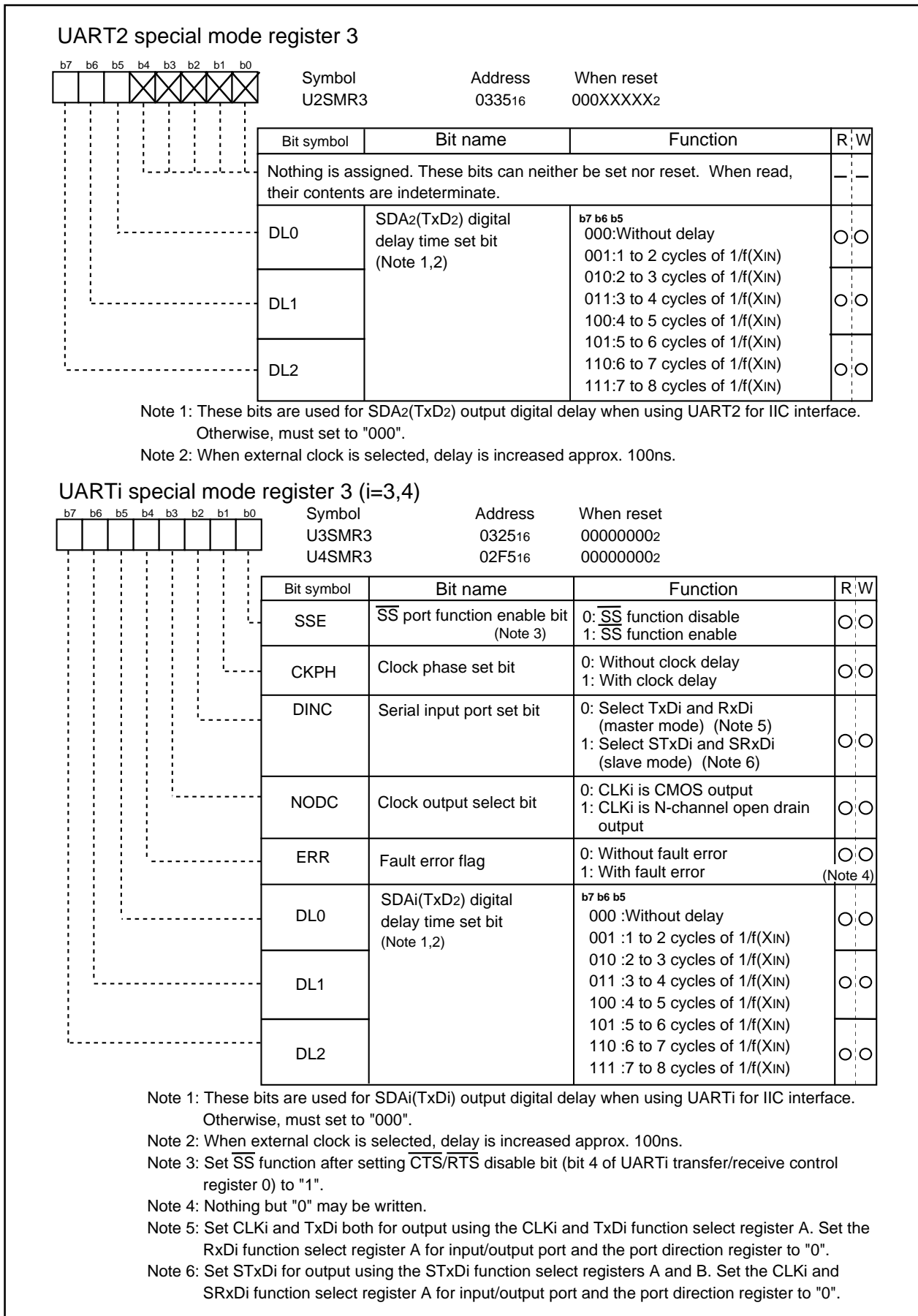


Figure 16.12 Serial I/O-related registers (8)

## 17. Clock synchronous serial I/O mode

The clock synchronous serial I/O mode uses a transfer clock to transmit and receive data. Tables 17.1 and 17.2 list the specifications of the clock synchronous serial I/O mode. Figure 17.1 shows the UARTi transmit/receive mode register.

**Table 17.1 Specifications of clock synchronous serial I/O mode (1)**

Item	Specification
Transfer data format	<ul style="list-style-type: none"> <li>Transfer data length: 8 bits</li> </ul>
Transfer clock	<ul style="list-style-type: none"> <li>When internal clock is selected (bit 3 at addresses 0360<sub>16</sub>, 0368<sub>16</sub>, 0338<sub>16</sub>, 0328<sub>16</sub>, 02F8<sub>16</sub> = "0") : <math>f_i / 2(n+1)</math> (Note) <math>f_i = f_1, f_8, f_{32}</math> <ul style="list-style-type: none"> <li>CLK is selected by the corresponding port function select register, peripheral function select register and peripheral subfunction select register.</li> </ul> </li> <li>When external clock is selected (bit 3 at addresses 0360<sub>16</sub>, 0368<sub>16</sub>, 0338<sub>16</sub>, 0328<sub>16</sub>, 02F8<sub>16</sub> = "1") : Input from CLKi pin <ul style="list-style-type: none"> <li>Set the corresponding function select register A to I/O port</li> </ul> </li> </ul>
Transmission/reception control	<ul style="list-style-type: none"> <li>CTS function/RTS function/<math>\overline{\text{CTS}}</math>, <math>\overline{\text{RTS}}</math> function chosen to be invalid</li> </ul>
Transmission start condition	<ul style="list-style-type: none"> <li>To start transmission, the following requirements must be met: <ul style="list-style-type: none"> <li>Transmit enable bit (bit 0 at addresses 0365<sub>16</sub>, 036D<sub>16</sub>, 033D<sub>16</sub>, 032D<sub>16</sub>, 02FD<sub>16</sub>) = "1"</li> <li>Transmit buffer empty flag (bit 1 at addresses 0365<sub>16</sub>, 036D<sub>16</sub>, 033D<sub>16</sub>, 032D<sub>16</sub>, 02FD<sub>16</sub>) = "0"</li> <li>When <math>\overline{\text{CTS}}</math> function selected, <math>\overline{\text{CTS}}</math> input level = "L"</li> <li>TxD output selected by the corresponding function select register A, B and C.</li> </ul> </li> <li>Furthermore, if external clock is selected, the following requirements must also be met: <ul style="list-style-type: none"> <li>CLKi polarity select bit (bit 6 at addresses 0364<sub>16</sub>, 036C<sub>16</sub>, 033C<sub>16</sub>, 032C<sub>16</sub>, 02FC<sub>16</sub>) = "0": CLKi input level = "H"</li> <li>CLKi polarity select bit (bit 6 at addresses 0364<sub>16</sub>, 036C<sub>16</sub>, 033C<sub>16</sub>, 032C<sub>16</sub>, 02FC<sub>16</sub>) = "1": CLKi input level = "L"</li> </ul> </li> </ul>
Reception start condition	<ul style="list-style-type: none"> <li>To start reception, the following requirements must be met: <ul style="list-style-type: none"> <li>Receive enable bit (bit 2 at addresses 0365<sub>16</sub>, 036D<sub>16</sub>, 033D<sub>16</sub>, 032D<sub>16</sub>, 02FD<sub>16</sub>) = "1"</li> <li>Transmit enable bit (bit 0 at addresses 0365<sub>16</sub>, 036D<sub>16</sub>, 033D<sub>16</sub>, 032D<sub>16</sub>, 02FD<sub>16</sub>) = "1"</li> <li>Transmit buffer empty flag (bit 1 at addresses 0365<sub>16</sub>, 036D<sub>16</sub>, 033D<sub>16</sub>, 032D<sub>16</sub>, 02FD<sub>16</sub>) = "0"</li> </ul> </li> <li>Furthermore, if external clock is selected, the following requirements must also be met: <ul style="list-style-type: none"> <li>CLKi polarity select bit (bit 6 at addresses 0364<sub>16</sub>, 036C<sub>16</sub>, 033C<sub>16</sub>, 032C<sub>16</sub>, 02FC<sub>16</sub>) = "0": CLKi input level = "H"</li> <li>CLKi polarity select bit (bit 6 at addresses 0364<sub>16</sub>, 036C<sub>16</sub>, 033C<sub>16</sub>, 032C<sub>16</sub>, 02FC<sub>16</sub>) = "1": CLKi input level = "L"</li> </ul> </li> </ul>
Interrupt request generation timing	<ul style="list-style-type: none"> <li>When transmitting <ul style="list-style-type: none"> <li>Transmit interrupt cause select bit (bits 0, 1 at address 0370<sub>16</sub>, bit 4 at address 033D<sub>16</sub>, 032D<sub>16</sub>, 02FD<sub>16</sub>) = "0": Interrupts requested when data transfer from UARTi transfer buffer register to UARTi transmit register is completed</li> <li>Transmit interrupt cause select bit (bits 0, 1 at address 0370<sub>16</sub>, bit 4 at address 033D<sub>16</sub>, 032D<sub>16</sub>, 02FD<sub>16</sub>) = "1": Interrupts requested when data transmission from UARTi transfer register is completed</li> </ul> </li> <li>When receiving <ul style="list-style-type: none"> <li>Interrupts requested when data transfer from UARTi receive register to UARTi receive buffer register is completed</li> </ul> </li> </ul>

Note : "n" denotes the value 00<sub>16</sub> to FF<sub>16</sub> that is set to the UART bit rate generator.

**Table 17.2 Specifications of clock synchronous serial I/O mode (2)**

Item	Specification
Error detection	<ul style="list-style-type: none"> <li>• Overrun error (Note 1) This error occurs when the next data is ready before contents of UARTi receive buffer register are read out</li> </ul>
Select function	<ul style="list-style-type: none"> <li>• CLK polarity selection Whether transmit data is output/input at the rising edge or falling edge of the transfer clock can be selected</li> <li>• LSB first/MSB first selection Whether transmission/reception begins with bit 0 or bit 7 can be selected</li> <li>• Continuous receive mode selection Reception is enabled simultaneously by a read from the receive buffer register</li> <li>• Transfer clock output from multiple pins selection (UART1) (Note 2) UART1 transfer clock can be chosen by software to be output from one of the two pins set</li> <li>• Separate CTS/RTS pins (UART0) (Note 2) UART0 CTS and RTS pins each can be assigned to separate pins</li> <li>• Switching serial data logic (UART2 to UART4) Whether to reverse data in writing to the transmission buffer register or reading the reception buffer register can be selected.</li> <li>• TxD, RxD I/O polarity reverse (UART2 to UART4) This function is reversing TxD port output and RxD port input. All I/O data level is reversed.</li> </ul>

Note 1: If an overrun error occurs, the UARTi receive buffer will have the next data written in. Note also that the UARTi receive interrupt request bit will not change.

Note 2: The transfer clock output from multiple pins and the separate CTS/RTS pins functions cannot be selected simultaneously.

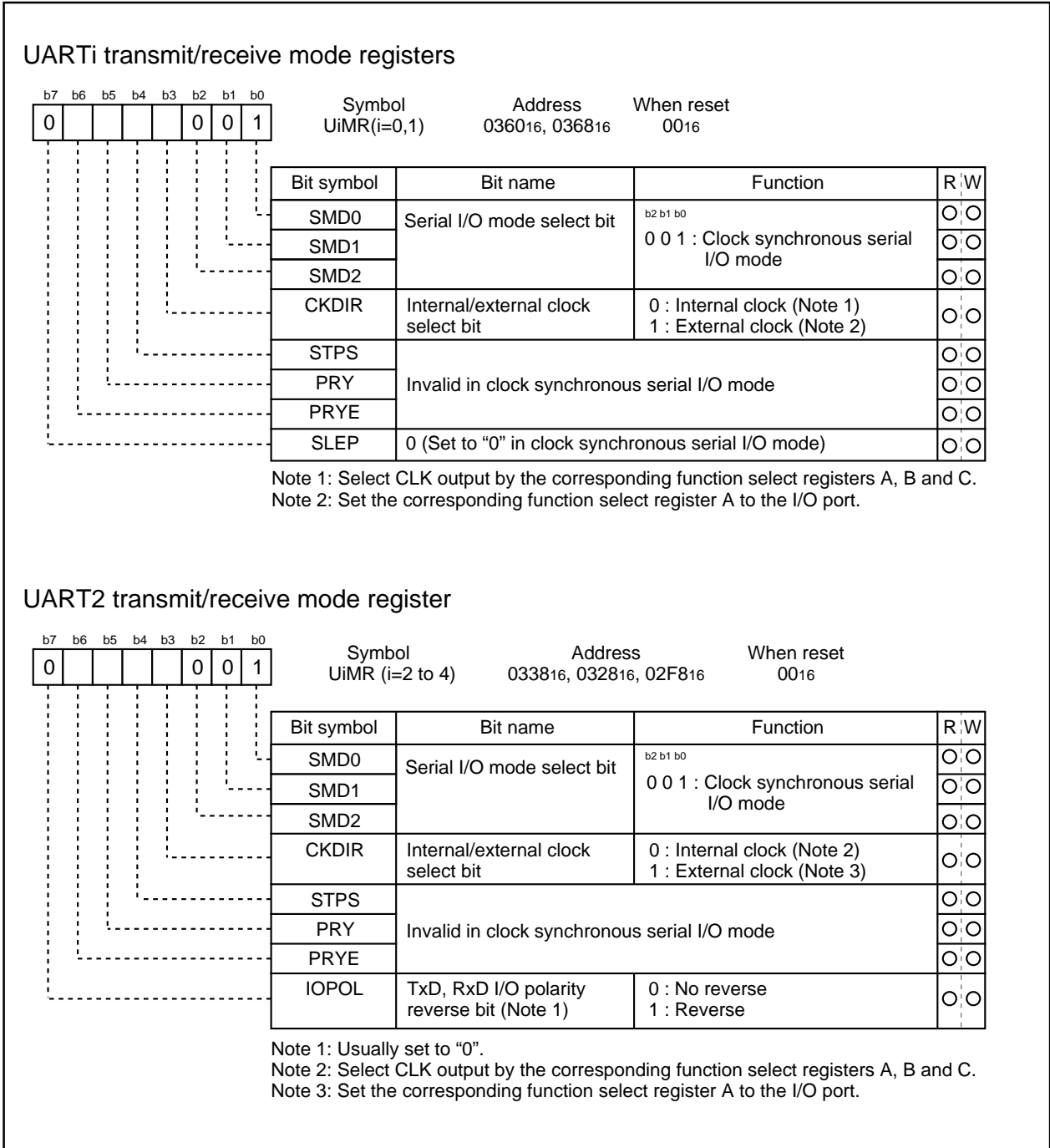


Figure 17.1 UARTi transmit/receive mode register in clock synchronous serial I/O mode



Table 17.3 lists the functions of the input/output pins during clock synchronous serial I/O mode. This table shows the pin functions when the transfer clock output from multiple pins and the separate  $\overline{\text{CTS}}$ / $\overline{\text{RTS}}$  pins functions are not selected. Note that for a period from when the UARTi operation mode is selected to when transfer starts, the TxDi pin outputs a "H". (If the N-channel open drain is selected, this pin is in floating state.)

**Table 17.3 Input/output pin functions in clock synchronous serial I/O mode**

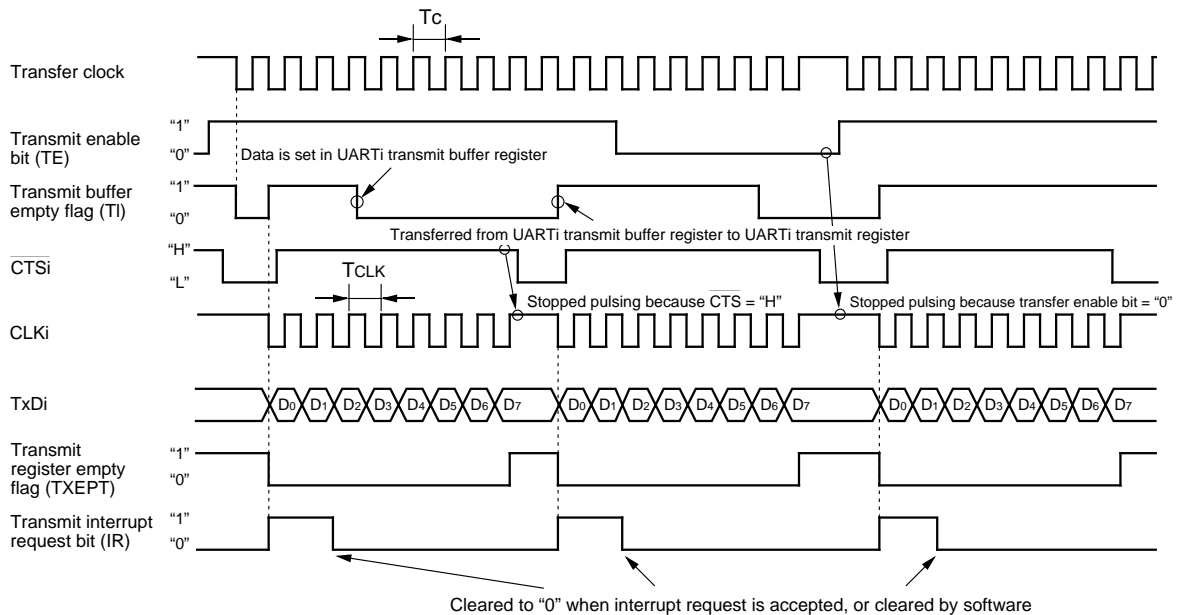
Pin name	Function	Method of selection
TxDi (P63, P67, P70, P92, P96)	Serial data output (Note 1)	(Outputs dummy data when performing reception only)
RxDi (P62, P66, P71, P91, P97)	Serial data input (Note 2)	Port P62, P66, P71, P91 and P97 direction register (bits 2 and 6 at address 03C216, bit 1 at address 03C316, bit 1 and 7 at address 03C716) = "0" (Can be used as an input port when performing transmission only)
CLKi (P61, P65, P72, P90, P95)	Transfer clock output (Note 1)	Internal/external clock select bit (bit 3 at addresses 036016, 036816, 033816, 032816, 02F816) = "0"
	Transfer clock input (Note 2)	Internal/external clock select bit (bit 3 at addresses 036016, 036816, 033816, 032816, 02F816) = "1" Port P61, P65, P72, P90 and P95 direction register (bits 1 and 5 at address 03C216, bit 2 at address 03C316, bit 0 and 5 at address 03C716) = "0"
$\overline{\text{CTS}}$ / $\overline{\text{RTS}}$ i (P60, P64, P73, P93, P94)	$\overline{\text{CTS}}$ input (Note 2)	$\overline{\text{CTS}}$ / $\overline{\text{RTS}}$ disable bit (bit 4 at addresses 036416, 036C16, 033C16, 032C16, 02FC16) = "0" $\overline{\text{CTS}}$ / $\overline{\text{RTS}}$ function select bit (bit 2 at addresses 036416, 036C16, 033C16, 032C16, 02FC16) = "0" Port P60, P64, P73, P93 and P94 direction register (bits 0 and 4 at address 03C216, bit 3 at address 03C316, bits 3 and 4 at address 03C716) = "0"
	$\overline{\text{RTS}}$ output (Note 1)	$\overline{\text{CTS}}$ / $\overline{\text{RTS}}$ disable bit (bit 4 at addresses 036416, 036C16, 033C16, 032C16, 02FC16) = "0" $\overline{\text{CTS}}$ / $\overline{\text{RTS}}$ function select bit (bit 2 at addresses 036416, 036C16, 033C16, 032C16, 02FC16) = "1"
	Programmable I/O port (Note 2)	$\overline{\text{CTS}}$ / $\overline{\text{RTS}}$ disable bit (bit 4 at addresses 036416, 036C16, 033C16, 032C16, 02FC16) = "1"

(when transfer clock output from multiple pins and separate  $\overline{\text{CTS}}$ / $\overline{\text{RTS}}$  pins functions are not selected)

Note 1: Select TxD output, CLK output and  $\overline{\text{RTS}}$  output by the corresponding function select registers A, B and C.

Note 2: Select I/O port by the corresponding function select register A.

• Example of transmit timing (when internal clock is selected)



• Example of receive timing (when external clock is selected)

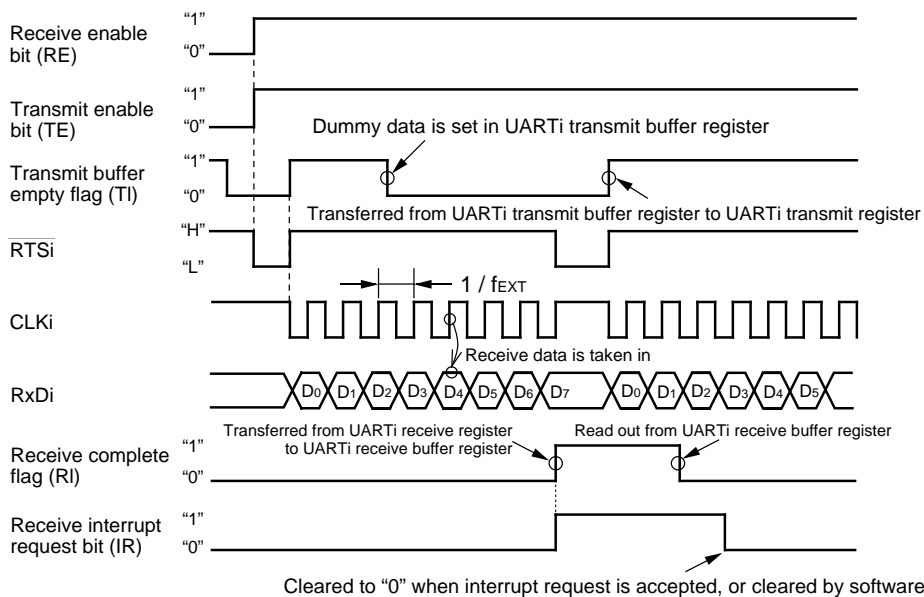
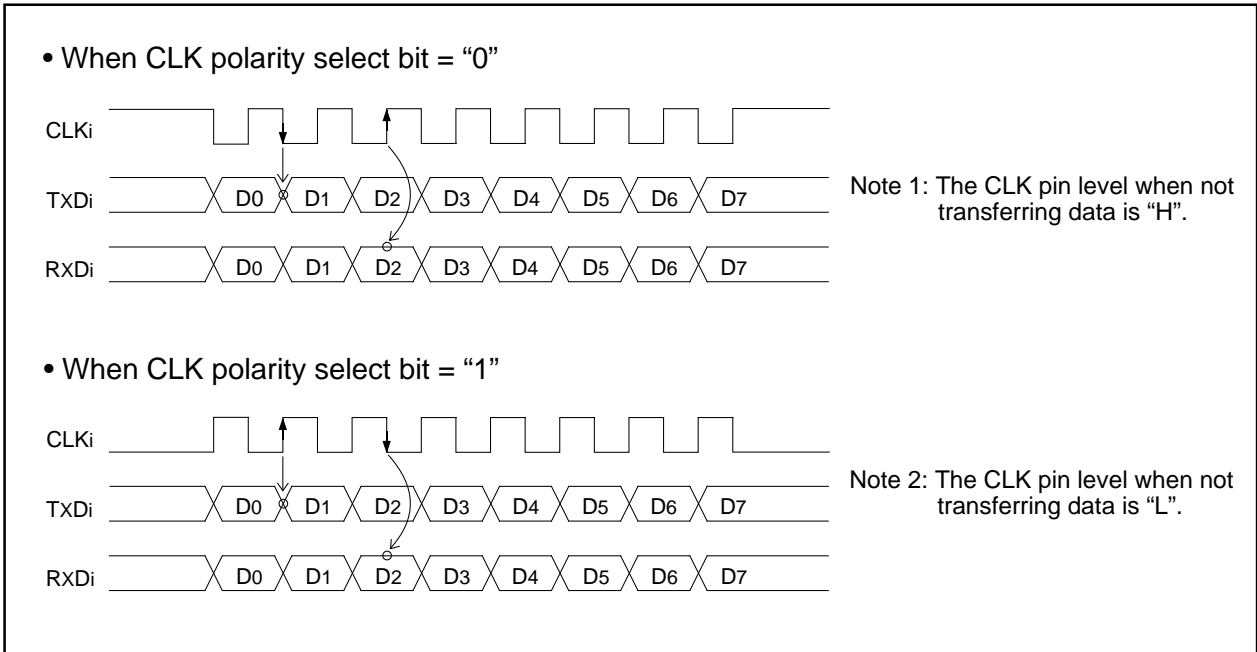


Figure 17.2 Typical transmit/receive timings in clock synchronous serial I/O mode

**(a) Polarity select function**

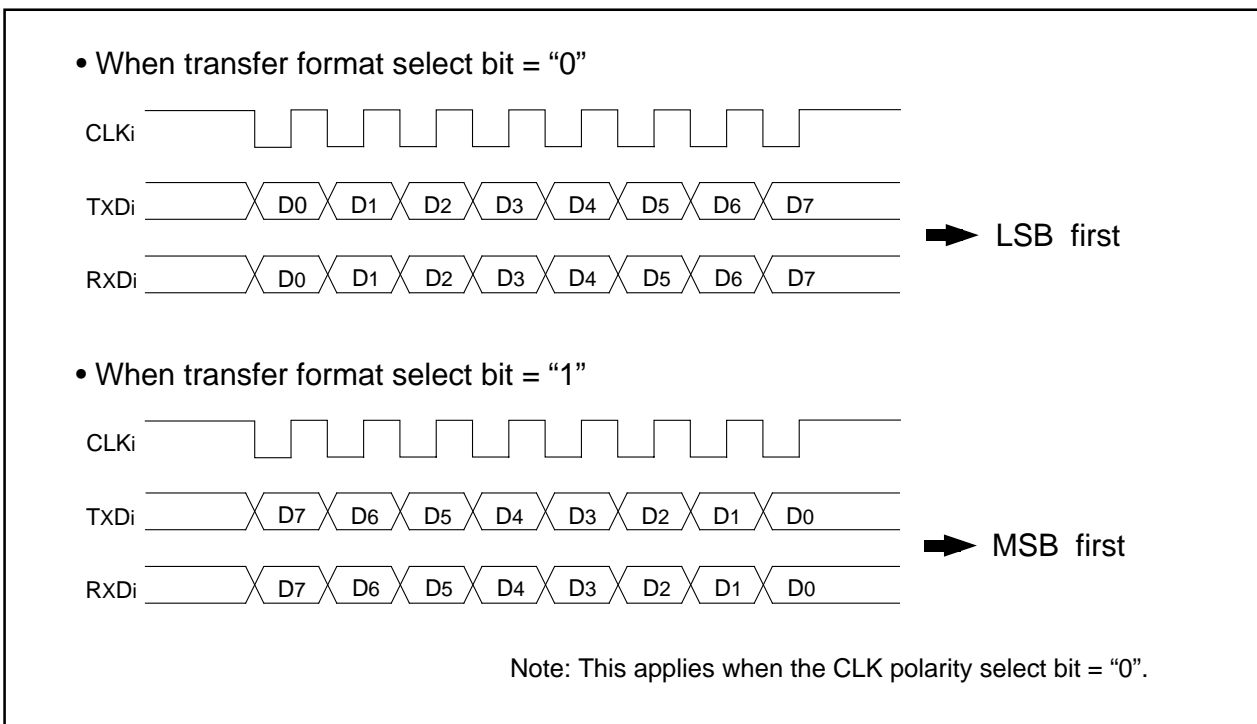
As shown in Figure 17.3, the CLK polarity select bit (bit 6 at addresses 0364<sub>16</sub>, 036C<sub>16</sub>, 033C<sub>16</sub>, 032C<sub>16</sub>, 02FC<sub>16</sub>) allows selection of the polarity of the transfer clock.



**Figure 17.3 Polarity of transfer clock**

**(b) LSB first/MSB first select function**

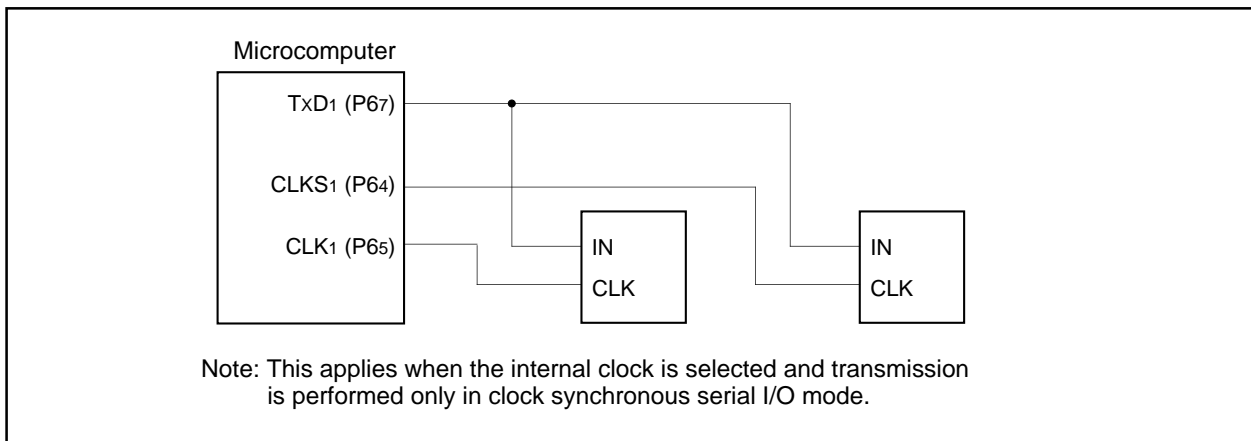
As shown in Figure 17.4, when the transfer format select bit (bit 7 at addresses 0364<sub>16</sub>, 036C<sub>16</sub>, 033C<sub>16</sub>, 032C<sub>16</sub>, 02FC<sub>16</sub>) = "0", the transfer format is "LSB first"; when the bit = "1", the transfer format is "MSB first".



**Figure 17.4 Transfer format**

**(c) Transfer clock output from multiple pins function (UART1)**

This function allows the setting two transfer clock output pins and choosing one of the two to output a clock by using the port function select register (bits of related to-P64 and P65). (See Figure 17.5) The multiple pins function is valid only when the internal clock is selected for UART1. Note that when this function is selected, UART1 CTS/RTS function cannot be used.



**Figure 17.5** The transfer clock output from the multiple pins function usage

**(d) Continuous receive mode**

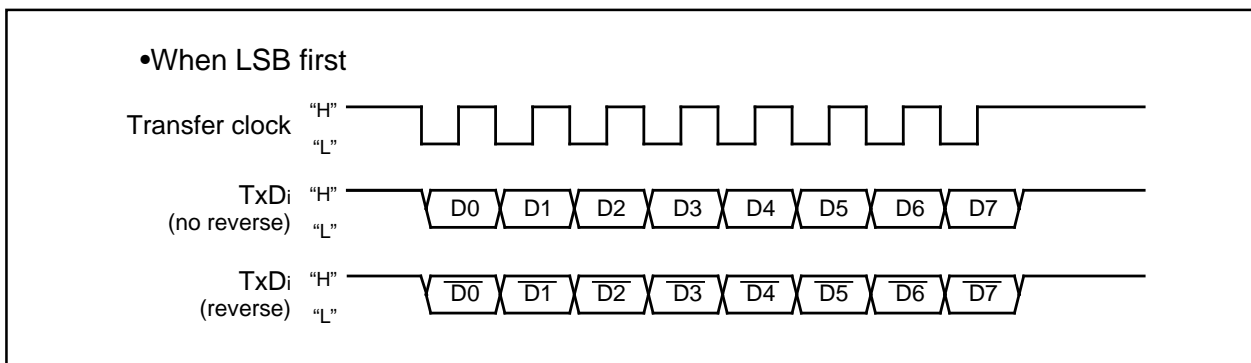
If the continuous receive mode enable bit (bits 2 and 3 at address 0370<sub>16</sub>, bit 5 at address 033D<sub>16</sub>, 032D<sub>16</sub>, 02FD<sub>16</sub>) is set to “1”, the unit is placed in continuous receive mode. In this mode, when the receive buffer register is read out, the unit simultaneously goes to a receive enable state without having to set dummy data to the transmit buffer register back again.

**(e) Separate CTS/RTS pins function (UART0)**

This function works the same way as in the clock asynchronous serial I/O (UART) mode. The method of setting and the input/output pin functions are both the same, so refer to select function in the next section, “(2) Clock asynchronous serial I/O (UART) mode.” Note that this function is invalid if the transfer clock output from the multiple pins function is selected.

**(f) Serial data logic switch function (UART2 to UART4)**

When the data logic select bit (bit6 at address 033D<sub>16</sub>, 032D<sub>16</sub>, 02FD<sub>16</sub>) = “1”, and writing to transmit buffer register or reading from receive buffer register, data is reversed. Figure 17.6 shows the example of serial data logic switch timing.



**Figure 17.6** Serial data logic switch timing

## 18. 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 18.1 and 18.2 list the specifications of the UART mode. Figure 18.1 shows the UARTi transmit/receive mode register.

**Table 18.1 Specifications of UART Mode (1)**

Item	Specification
Transfer data format	<ul style="list-style-type: none"> <li>• Character bit (transfer data): 7 bits, 8 bits, or 9 bits as selected</li> <li>• Start bit: 1 bit</li> <li>• Parity bit: Odd, even, or nothing as selected</li> <li>• Stop bit: 1 bit or 2 bits as selected</li> </ul>
Transfer clock	<ul style="list-style-type: none"> <li>• When internal clock is selected (bit 3 at addresses 0360<sub>16</sub>, 0368<sub>16</sub>, 0338<sub>16</sub>, 0328<sub>16</sub>, 02F8<sub>16</sub> = "0") : <math>f_i/16(n+1)</math> (Note 1) <math>f_i = f_1, f_8, f_{32}</math></li> <li>• When external clock is selected (bit 3 at addresses 0360<sub>16</sub>, 0368<sub>16</sub>, 0338<sub>16</sub>, 0328<sub>16</sub>, 02F8<sub>16</sub> = "1") : <math>f_{EXT}/16(n+1)</math>(Note 1) (Note 2)</li> </ul>
Transmission/reception control	<ul style="list-style-type: none"> <li>• <math>\overline{CTS}</math> function/<math>\overline{RTS}</math> function/<math>\overline{CTS}</math>, <math>\overline{RTS}</math> function chosen to be invalid</li> </ul>
Transmission start condition	<ul style="list-style-type: none"> <li>• To start transmission, the following requirements must be met: <ul style="list-style-type: none"> <li>- Transmit enable bit (bit 0 at addresses 0365<sub>16</sub>, 036D<sub>16</sub>, 033D<sub>16</sub>, 032D<sub>16</sub>, 02FD<sub>16</sub>) = "1"</li> <li>- Transmit buffer empty flag (bit 1 at addresses 0365<sub>16</sub>, 036D<sub>16</sub>, 033D<sub>16</sub>, 032D<sub>16</sub>, 02FD<sub>16</sub>) = "0"</li> <li>- When <math>\overline{CTS}</math> function selected, <math>\overline{CTS}</math> input level = "L"</li> <li>- TxD output is selected by the corresponding function select register A, B and C.</li> </ul> </li> </ul>
Reception start condition	<ul style="list-style-type: none"> <li>• To start reception, the following requirements must be met: <ul style="list-style-type: none"> <li>- Receive enable bit (bit 2 at addresses 0365<sub>16</sub>, 036D<sub>16</sub>, 033D<sub>16</sub>, 032D<sub>16</sub>, 02FD<sub>16</sub>) = "1"</li> <li>- Start bit detection</li> </ul> </li> </ul>
Interrupt request generation timing	<ul style="list-style-type: none"> <li>• When transmitting <ul style="list-style-type: none"> <li>- Transmit interrupt cause select bits (bits 0,1 at address 0370<sub>16</sub>, bit 4 at address 033D<sub>16</sub>, 032D<sub>16</sub>, 02FD<sub>16</sub>) = "0": Interrupts requested when data transfer from UARTi transfer buffer register to UARTi transmit register is completed</li> <li>- Transmit interrupt cause select bits (bits 0, 1 at address 0370<sub>16</sub>, bit 4 at address 033D<sub>16</sub>, 032D<sub>16</sub>, 02FD<sub>16</sub>) = "1": Interrupts requested when data transmission from UARTi transfer register is completed</li> </ul> </li> <li>• When receiving <ul style="list-style-type: none"> <li>- Interrupts requested when data transfer from UARTi receive register to UARTi receive buffer register is completed</li> </ul> </li> </ul>

Note 1: 'n' denotes the value 00<sub>16</sub> to FF<sub>16</sub> that is set to the UARTi bit rate generator.

Note 2: f<sub>EXT</sub> is input from the CLKi pin.

**Table 18.2 Specifications of UART Mode (2)**

Item	Specification
Error detection	<ul style="list-style-type: none"> <li>• Overrun error (Note) This error occurs when the next data is ready before contents of UARTi receive buffer register are read out</li> <li>• Framing error This error occurs when the number of stop bits set is not detected</li> <li>• 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</li> <li>• Error sum flag This flag is set (= 1) when any of the overrun, framing, and parity errors is encountered</li> </ul>
Select function	<ul style="list-style-type: none"> <li>• Separate CTS/RTS pins (UART0) UART0 <math>\overline{\text{CTS}}</math> and <math>\overline{\text{RTS}}</math> pins each can be assigned to separate pins</li> <li>• Sleep mode selection (UART0, UART1) This mode is used to transfer data to and from one of multiple slave micro-computers</li> <li>• Serial data logic switch (UART2 to UART4) This function is reversing logic value of transferring data. Start bit, parity bit and stop bit are not reversed.</li> <li>• TxD, RxD I/O polarity switch (UART2 to UART4) This function is reversing TxD port output and RxD port input. All I/O data level is reversed.</li> </ul>

Note: If an overrun error occurs, the UARTi receive buffer will have the next data written in. Note also that the UARTi receive interrupt request bit will not change.

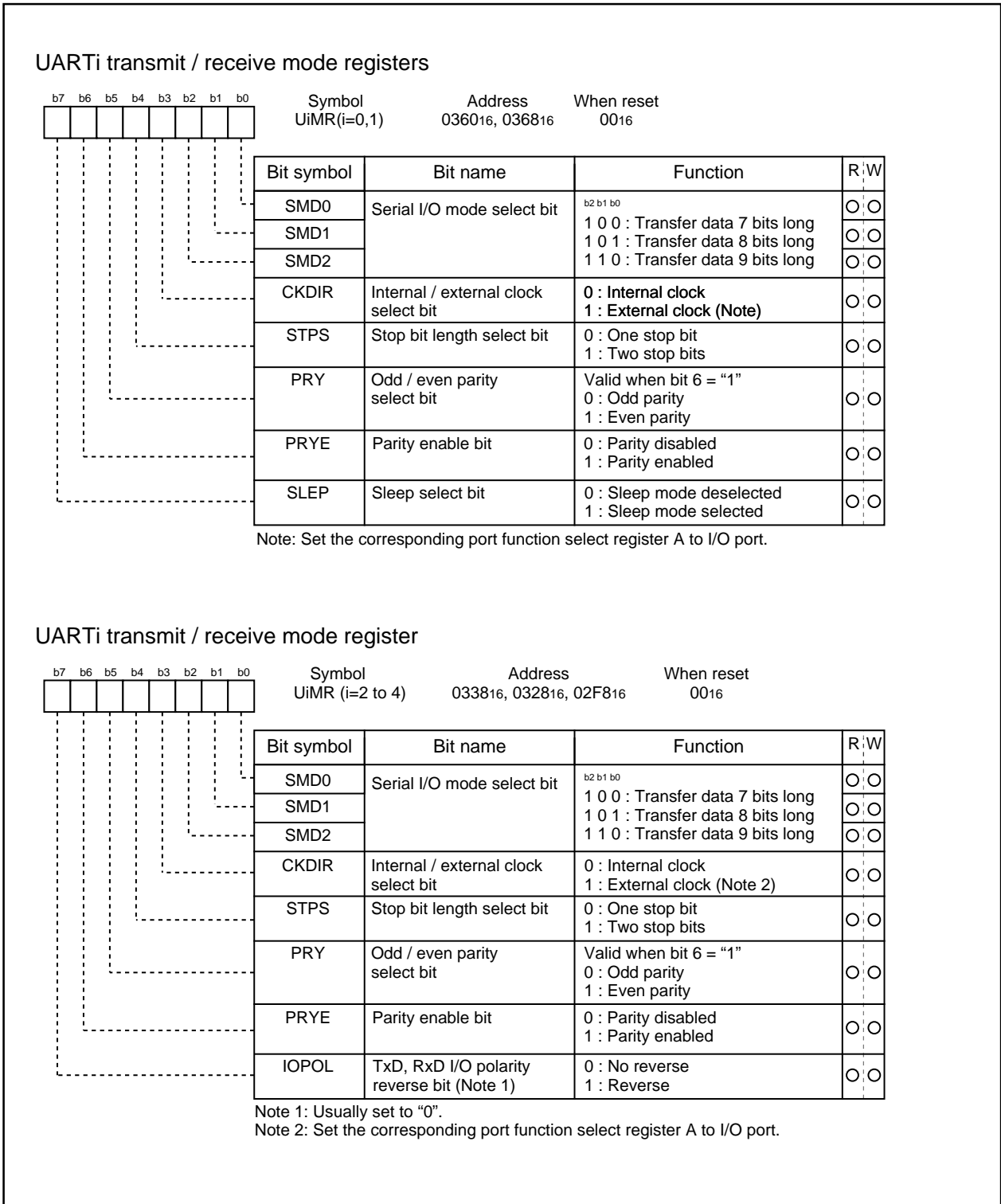


Figure 18.1 UARTi transmit/receive mode register in UART mode

Table 18.3 lists the functions of the input/output pins during UART mode. This table shows the pin functions when the separate  $\overline{\text{CTS}}/\overline{\text{RTS}}$  pins function is not selected. Note that for a period from when the UARTi operation mode is selected to when transfer starts, the TxDi pin outputs a "H". (If the N-channel open drain is selected, this pin is in floating state.)

**Table 18.3 Input/output pin functions in UART mode**

Pin name	Function	Method of selection
TxDi (P63, P67, P70, P92, P96)	Serial data output (Note 1)	
RxDi (P62, P66, P71, P91, P97)	Serial data input (Note 2)	Port P62, P66, P71, P91 and P97 direction register (bits 2 and 6 at address 03C216, bit 1 at address 03C316, bit 1 and 7 at address 03C716) = "0" (Can be used as an input port when performing transmission only)
CLKi (P61, P65, P72, P90, P95)	Programmable I/O port (Note 2)	Internal/external clock select bit (bit 3 at addresses 036016, 036816, 033816, 032816, 02F816) = "0"
	Transfer clock input (Note 2)	Internal/external clock select bit (bit 3 at addresses 036016, 036816, 033816, 032816, 02F816) = "1" Port P61, P65, P72, P90 and P95 direction register (bits 1 and 5 at address 03C216, bit 2 at address 03C316, bits 0 and 5 at address 03C716) = "0"
$\overline{\text{CTS}}/\overline{\text{RTS}}_i$ (P60, P64, P73, P93, P94)	$\overline{\text{CTS}}$ input (Note 2)	$\overline{\text{CTS}}/\overline{\text{RTS}}$ disable bit (bit 4 at addresses 036416, 036C16, 033C16, 032C16, 02FC16) = "0" $\overline{\text{CTS}}/\overline{\text{RTS}}$ function select bit (bit 2 at addresses 036416, 036C16, 033C16, 032C16, 02FC16) = "0" Port P60, P64, P73, P93 and P94 direction register (bits 0 and 4 at address 03C216, bit 3 at address 03C316, bits 3 and 4 at address 03C716) = "0"
	$\overline{\text{RTS}}$ output (Note 1)	$\overline{\text{CTS}}/\overline{\text{RTS}}$ disable bit (bit 4 at addresses 036416, 036C16, 033C16, 032C16, 02FC16) = "0" $\overline{\text{CTS}}/\overline{\text{RTS}}$ function select bit (bit 2 at addresses 036416, 036C16, 033C16, 032C16, 02FC16) = "1"
	Programmable I/O port (Note 2)	$\overline{\text{CTS}}/\overline{\text{RTS}}$ disable bit (bit 4 at addresses 036416, 036C16, 033C16, 032C16, 02FC16) = "1"

(When separate  $\overline{\text{CTS}}/\overline{\text{RTS}}$  pins function is not selected)

Note 1: Select TxD output, CLK output and  $\overline{\text{RTS}}$  output by the corresponding function select registers A, B and C.

Note 2: Select I/O port by the corresponding function select register A.



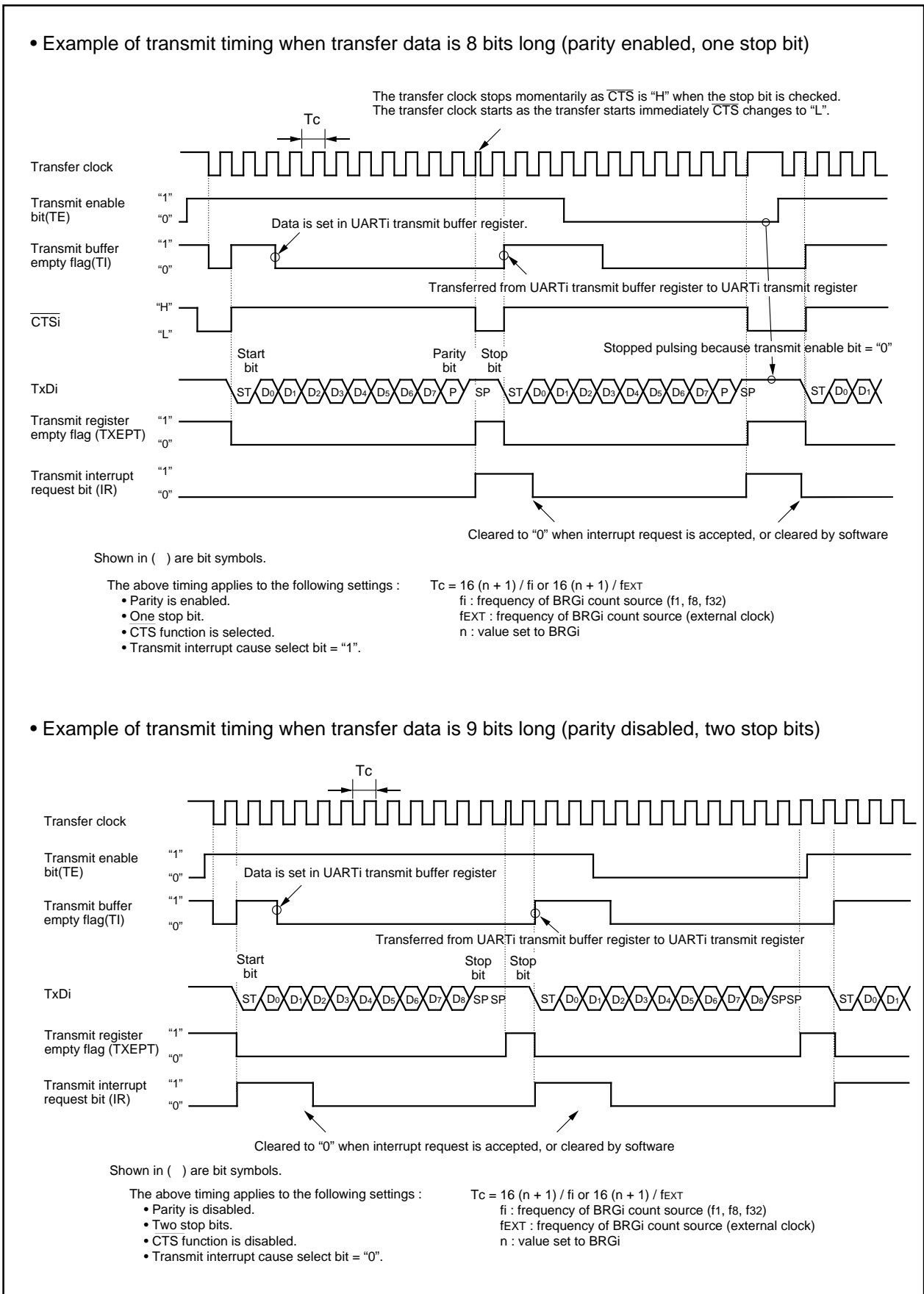


Figure 18.2 Typical transmit timings in UART mode

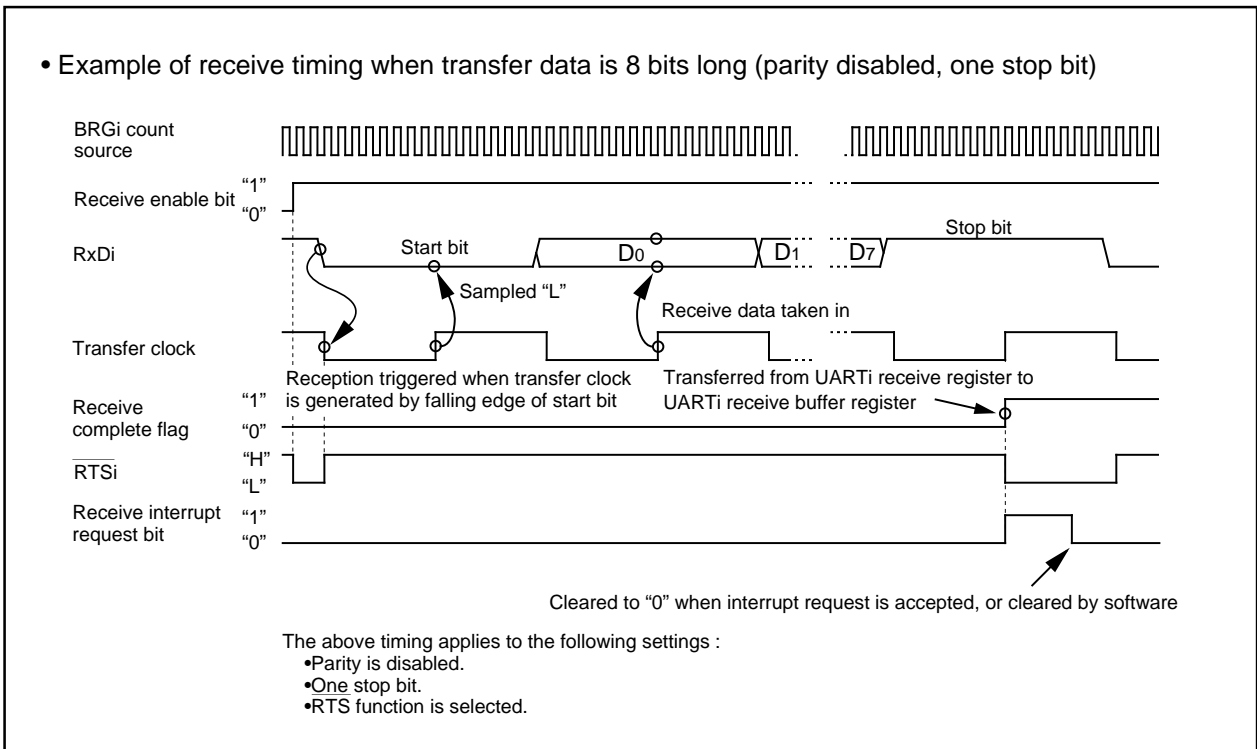


Figure 18.3 Typical receive timing in UART mode

**(a) Separate CTS/RTS pins function (UART0)**

With the separate CTS/RTS bit (bit 6 at address 037016) is set to "1", the unit outputs/inputs the CTS and RTS signals on different pins. (See Figure 18.4) This function is valid only for UART0. Note that if this function is selected, the CTS/RTS function for UART1 cannot be used.

Set both CTS/RTS function select bit (bit 2 at address 036C16) of UART1 and CTS/RTS disable bit (bit 4 at address 036C16) of UART1 to "0" and set P64 to input port by the function select register.

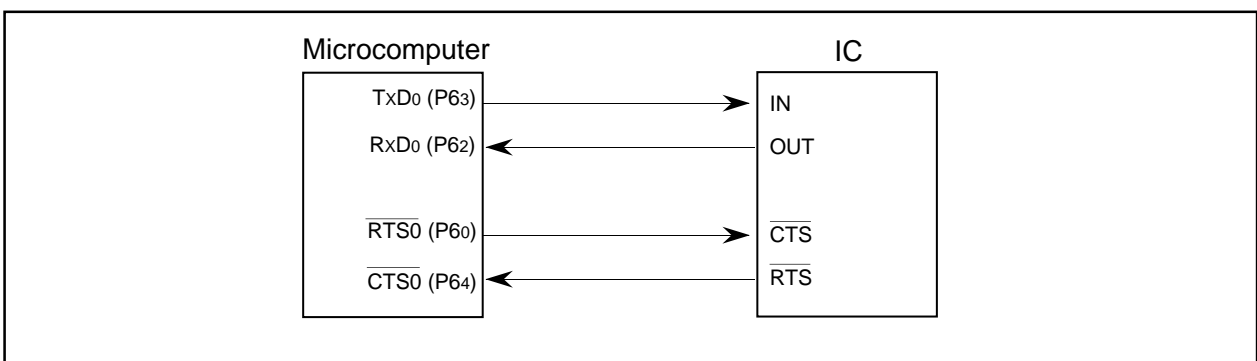


Figure 18.4 The separate CTS/RTS pins function usage

**(b) Sleep mode (UART0, UART1)**

This mode is used to transfer data between specific microcomputers among multiple microcomputers connected using UARTi. The sleep mode is selected when the sleep select bit (bit 7 at addresses 036016, 036816) is set to "1" during reception. In this mode, the unit performs receive operation when the MSB of the received data = "1" and does not perform receive operation when the MSB = "0".

**(c) Function for switching serial data logic (UART2 to UART4)**

When the data logic select bit (bit 6 of address 033D<sub>16</sub>, 032D<sub>16</sub>, 02FD<sub>16</sub>) is assigned 1, data is inverted in writing to the transmission buffer register or reading the reception buffer register. Figure 18.5 shows the example of timing for switching serial data logic.

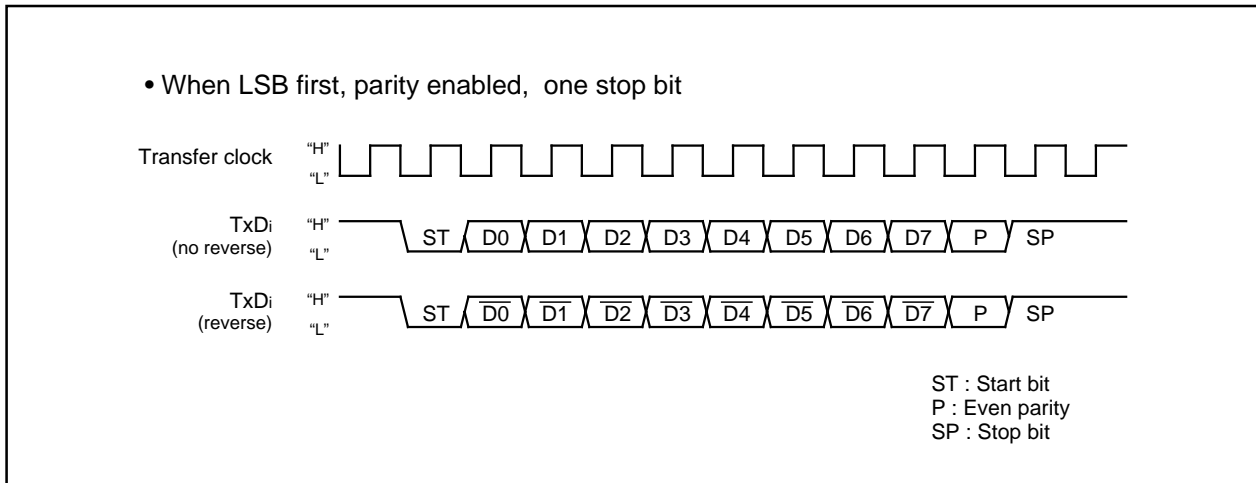


Figure 18.5 Timing for switching serial data logic

**(d) TxD, RxD I/O polarity reverse function (UART2 to UART4)**

This function is to reverse TxD pin output and RxD pin input. The level of any data to be input or output (including the start bit, stop bit(s), and parity bit) is reversed. Set this function to "0" (not to reverse) for usual use.

**(e) Bus collision detection function (UART2 to UART4)**

This function is to sample the output level of the TxD pin and the input level of the RxD pin at the rising edge of the transfer clock; if their values are different, then an interrupt request occurs. Figure 18.6 shows the example of detection timing of a buss collision (in UART mode).

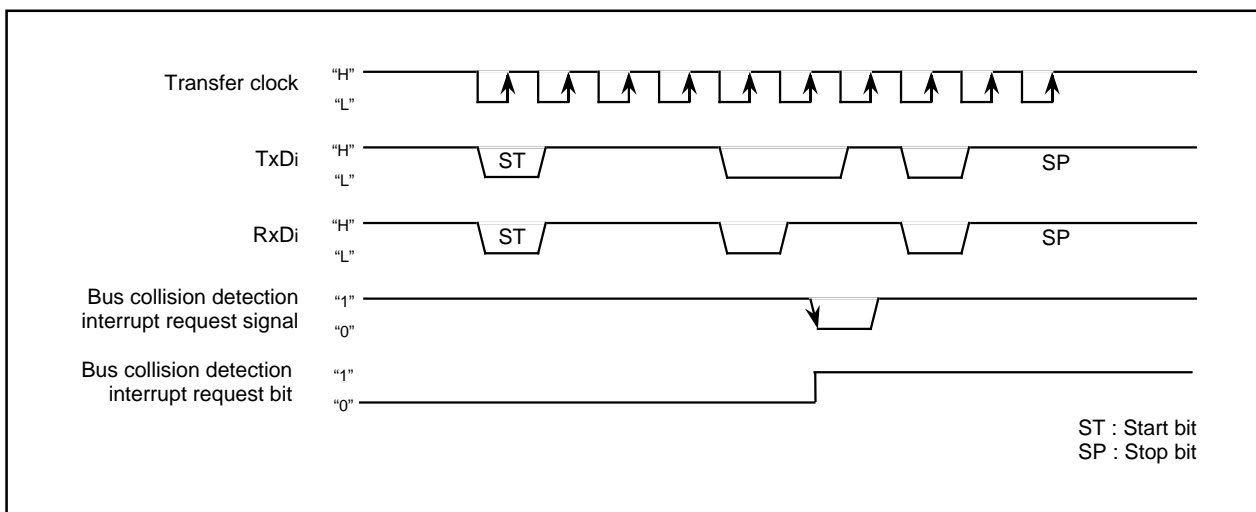


Figure 18.6 Detection timing of a bus collision (in UART mode)

## 19. Clock-asynchronous serial I/O mode (compliant with the SIM interface)

The SIM interface is used for connecting the microcomputer with a memory card I/C or the like; adding some extra settings in UART2 to UART4 clock-asynchronous serial I/O mode allows the user to effect this function.

Table 19.1 shows the specifications of clock-asynchronous serial I/O mode (compliant with the SIM interface).

**Table 19.1 Specifications of clock-asynchronous serial I/O mode (compliant with the SIM interface)**

Item	Specification
Transfer data format	<ul style="list-style-type: none"> <li>Transfer data 8-bit UART mode (bit 2 to 0 of addresses 0338<sub>16</sub>, 0328<sub>16</sub>, 02F8<sub>16</sub> = "1012")</li> <li>One stop bit (bit 4 of addresses 0338<sub>16</sub>, 0328<sub>16</sub>, 02F8<sub>16</sub> = "0")</li> <li>With the direct format chosen           <ul style="list-style-type: none"> <li>Set parity to "even" (bit 5 and 6 of addresses 0338<sub>16</sub>, 0328<sub>16</sub>, 02F8<sub>16</sub> = "1" and "1" respectively)</li> <li>Set data logic to "direct" (bit 6 of address 033D<sub>16</sub>, 032D<sub>16</sub>, 02FD<sub>16</sub> = "0").</li> <li>Set transfer format to LSB (bit 7 of address 033C<sub>16</sub>, 032C<sub>16</sub>, 02FC<sub>16</sub> = "0").</li> </ul> </li> <li>With the inverse format chosen           <ul style="list-style-type: none"> <li>Set parity to "odd" (bit 5 and 6 of addresses 0338<sub>16</sub>, 0328<sub>16</sub>, 02F8<sub>16</sub> = "0" and "1" respectively)</li> <li>Set data logic to "inverse" (bit 6 of address 033D<sub>16</sub>, 032D<sub>16</sub>, 02FD<sub>16</sub> = "1")</li> <li>Set transfer format to MSB (bit 7 of address 033C<sub>16</sub>, 032C<sub>16</sub>, 02FC<sub>16</sub> = "1")</li> </ul> </li> </ul>
Transfer clock	<ul style="list-style-type: none"> <li>With the internal clock chosen (bit 3 of addresses 0338<sub>16</sub>, 0328<sub>16</sub>, 02F8<sub>16</sub> = "0")           <ul style="list-style-type: none"> <li>: <math>f_i / 16 (n + 1)</math> (Note 1) : <math>f_i=f_1, f_8, f_{32}</math></li> </ul> </li> <li>With an external clock chosen (bit 3 of addresses 0338<sub>16</sub>, 0328<sub>16</sub>, 02F8<sub>16</sub> = "1")           <ul style="list-style-type: none"> <li>: <math>f_{EXT} / 16 (n+1)</math> (Note 1) (Note 2)</li> </ul> </li> </ul>
Transmission / reception control	<ul style="list-style-type: none"> <li>Disable the <math>\overline{CTS}</math> and <math>\overline{RTS}</math> function (bit 4 of address 033C<sub>16</sub>, 032C<sub>16</sub>, 02FC<sub>16</sub> = "1")</li> </ul>
Other settings	<ul style="list-style-type: none"> <li>The sleep mode select function is not available for UART2 and UART3</li> <li>Set transmission interrupt factor to "transmission completed" (bit 4 of address 033D<sub>16</sub>, 032D<sub>16</sub>, 02FD<sub>16</sub> = "1")</li> <li>Set N-channel open drain output to TxD and RxD pins in UART3 and 4 (bit 5 of address 032C<sub>16</sub>, 02FC<sub>16</sub> = "1")</li> </ul>
Transmission start condition	<ul style="list-style-type: none"> <li>To start transmission, the following requirements must be met:           <ul style="list-style-type: none"> <li>- Transmit enable bit (bit 0 of address 033D<sub>16</sub>, 032D<sub>16</sub>, 02FD<sub>16</sub>) = "1"</li> <li>- Transmit buffer empty flag (bit 1 of address 033D<sub>16</sub>, 032D<sub>16</sub>, 02FD<sub>16</sub>) = "0"</li> </ul> </li> </ul>
Reception start condition	<ul style="list-style-type: none"> <li>To start reception, the following requirements must be met:           <ul style="list-style-type: none"> <li>- Reception enable bit (bit 2 of address 033D<sub>16</sub>, 032D<sub>16</sub>, 02FD<sub>16</sub>) = "1"</li> <li>- Detection of a start bit</li> </ul> </li> </ul>
Interrupt request generation timing	<ul style="list-style-type: none"> <li>When transmitting           <ul style="list-style-type: none"> <li>When data transmission from the UART2 to UART4 transfer register is completed (bit 4 of address 033D<sub>16</sub>, 032D<sub>16</sub>, 02FD<sub>16</sub> = "1")</li> </ul> </li> <li>When receiving           <ul style="list-style-type: none"> <li>When data transfer from the UART2 to UART4 receive register to the UART2 to UART4 receive buffer register is completed</li> </ul> </li> </ul>
Error detection	<ul style="list-style-type: none"> <li>Overrun error (see the specifications of clock-asynchronous serial I/O) (Note 3)</li> <li>Framing error (see the specifications of clock-asynchronous serial I/O)</li> <li>Parity error (see the specifications of clock-asynchronous serial I/O)           <ul style="list-style-type: none"> <li>- On the reception side, an "L" level is output from the TxDi pin by use of the parity error signal output function (bit 7 of address 033D<sub>16</sub>, 032D<sub>16</sub>, 02FD<sub>16</sub> = "1") when a parity error is detected</li> <li>- On the transmission side, a parity error is detected by the level of input to the RxDi pin when a transmission interrupt occurs</li> </ul> </li> <li>The error sum flag (see the specifications of clock-asynchronous serial I/O)</li> </ul>

Note 1: 'n' denotes the value 00<sub>16</sub> to FF<sub>16</sub> that is set to the UARTi bit rate generator.

Note 2: f<sub>EXT</sub> is input from the CLKi pin.

Note 3: If an overrun error occurs, the UARTi receive buffer will have the next data written in. Note also that the UARTi receive interrupt request bit will not change.

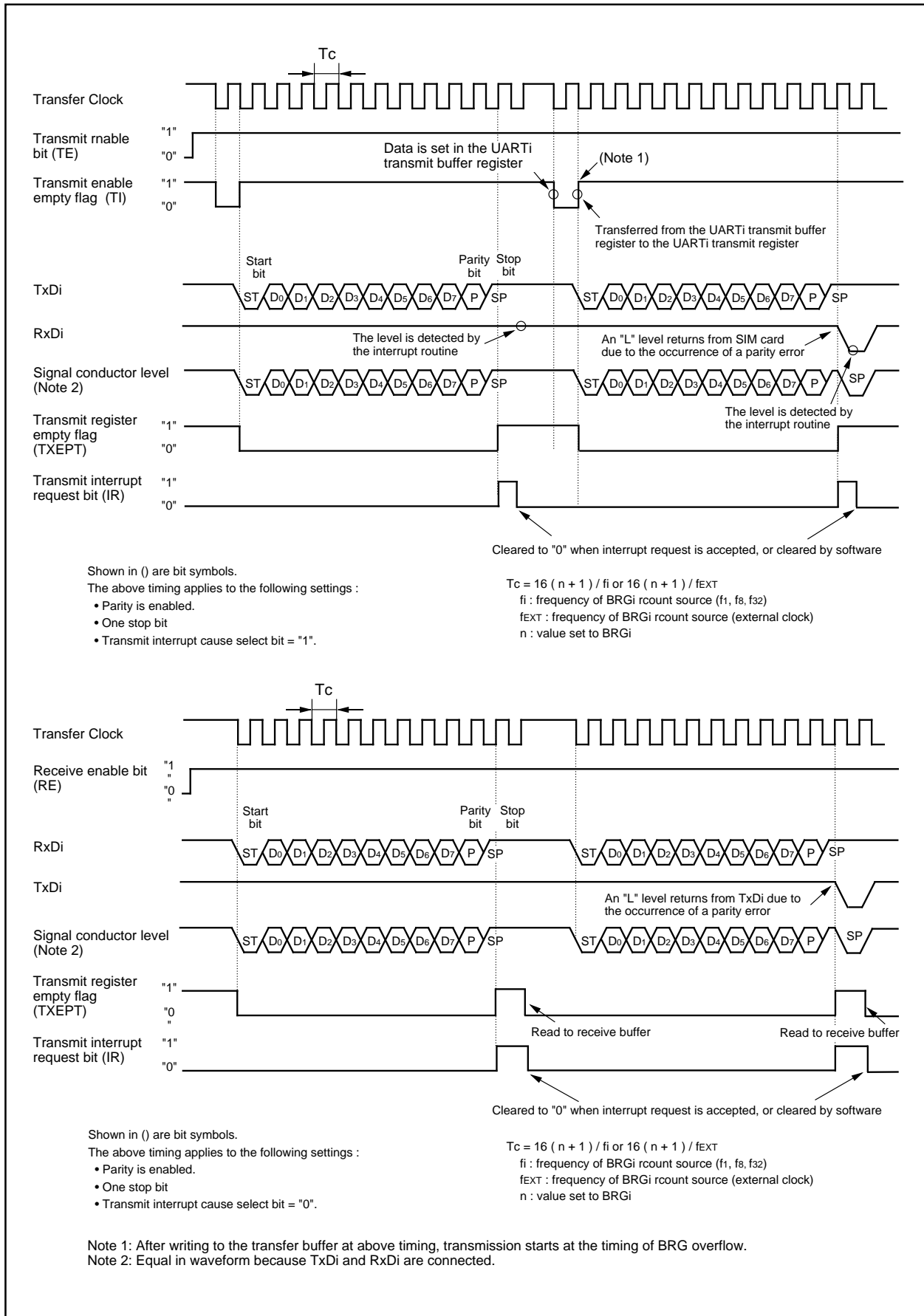
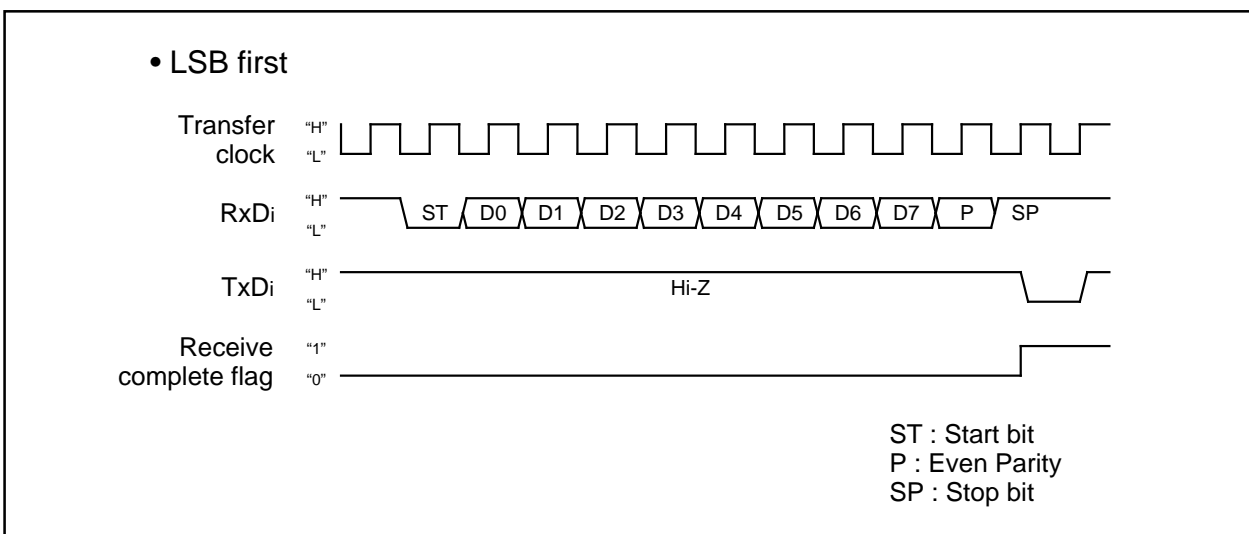


Figure 19.1 Typical transmit/receive timing in UART mode (compliant with the SIM interface)

**(a) Function for outputting a parity error signal**

During reception, with the error signal output enable bit (bit 7 of address 033D16, 032D16, 02FD16) assigned "1", you can output an "L" level from the TxDi pin when a parity error is detected. If the UARTi receive buffer register is read while outputting a parity error signal, the parity error flag is cleared to "0" and at the same time the TxDi output is returned high. And during transmission, comparing with the case in which the error signal output enable bit (bit 7 of address 033D16, 032D16, 02FD16) is assigned "0", the transmission completion interrupt occurs in the half cycle later of the transfer clock. Therefore parity error signals can be detected by a transmission completion interrupt program. Figure 19.2 shows the output timing of the parity error signal.

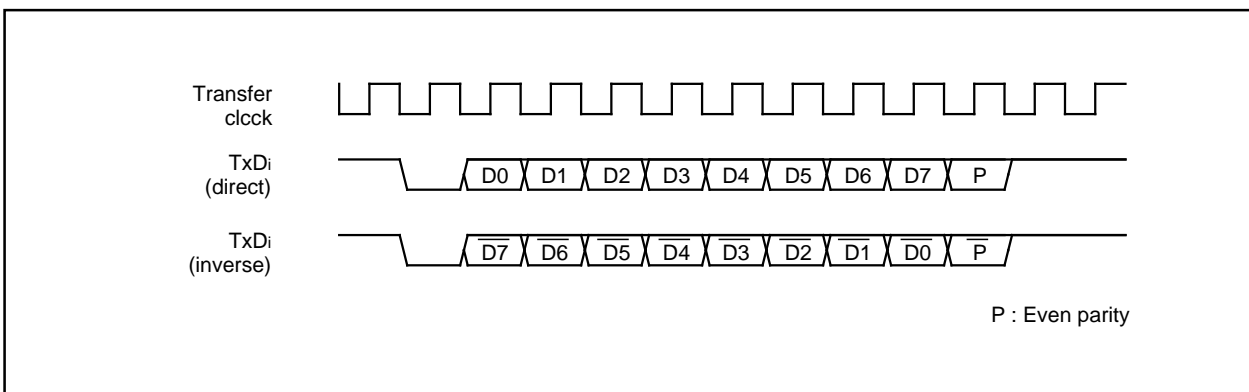


**Figure 19.2 Output timing of the parity error signal**

**(b) Direct format/inverse format**

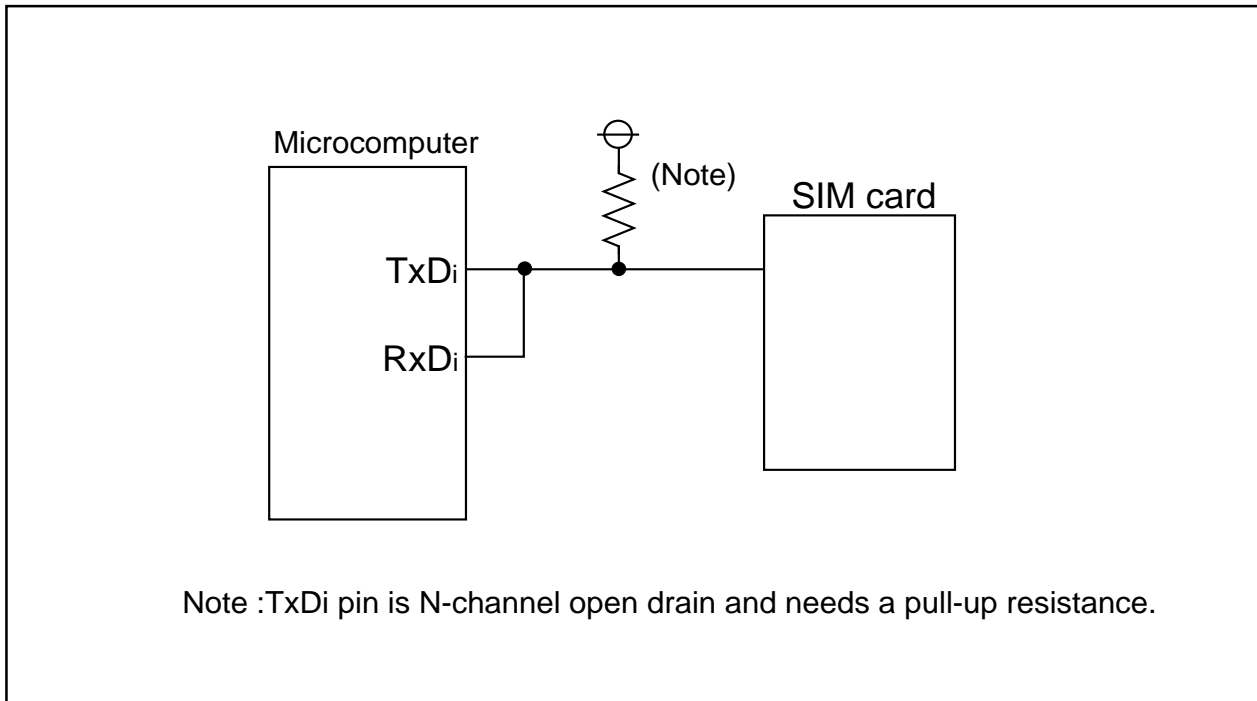
Connecting the SIM card allows you to switch between direct format and inverse format. If you choose the direct format, D0 data is output from TxDi. If you choose the inverse format, D7 data is inverted and output from TxDi.

Figure 19.3 shows the SIM interface format.



**Figure 19.3 SIM interface format**

Figure 19.4 shows the example of connecting the SIM interface. Connect TxDi and RxDi and apply pull-up.



**Figure 19.4 Connecting the SIM interface**

## 20. UARTi Special Mode Register (i = 2 to 4)

UART2 to UART4 operate the IIC bus interface (simple IIC bus) using the UARTi special mode register (addresses 0336<sub>16</sub>, 0326<sub>16</sub> and 02F6<sub>16</sub> [i = 2 to 4]) and UARTi special mode register 2 (addresses 0336<sub>16</sub>, 0326<sub>16</sub> and 02F6<sub>16</sub> [i = 2 to 4]). UART3 and UART4 add special functions using UARTi special mode register 3 (addresses 0325<sub>16</sub> and 02F5<sub>16</sub> [i = 3 or 4]).

### (1) IIC Bus Interface Mode

The I<sup>2</sup>C bus interface mode is provided with UART2 to UART4.

Table 20.1 shows the construction of the UARTi special mode register and UARTi special mode register 2.

When the I<sup>2</sup>C mode select bit (bit 0 in addresses 0337<sub>16</sub>, 0327<sub>16</sub> and 02F7<sub>16</sub>) is set to "1", the I<sup>2</sup>C bus (simple I<sup>2</sup>C bus) interface circuit is enabled.

To use the I<sup>2</sup>C bus, set the SCLi and the SDAi of both master and slave to output with the function select register. In UART3 and 4, set the data output select bit (bit 5 in address 032C<sub>16</sub> and 02FC<sub>16</sub>) to N-channel open drain output.

Table 20.1 shows the relationship of the IIC mode select bit to control. To use the chip in the clock synchronized serial I/O mode or clock asynchronous serial I/O mode, always set this bit to "0".

**Table 20.1 Features in I<sup>2</sup>C mode**

	Function	Normal mode	I <sup>2</sup> C mode (Note 1)
1	Factor of interrupt number 39 to 41 (Note 2)	Bus collision detection	Start condition detection or stop condition detection
2	Factor of interrupt number 33, 35, 37 (Note 2)	UARTi transmission	No acknowledgment detection (NACK)
3	Factor of interrupt number 34, 36, 38 (Note 2)	UARTi reception	Acknowledgment detection (ACK)
4	UARTi transmission output delay	Not delayed	Delayed
5	P7 <sub>0</sub> , P9 <sub>2</sub> , P9 <sub>6</sub> at the time when UARTi is in use	TxDi (output)	SDAi (input/output) (Note 3)
6	P7 <sub>1</sub> , P9 <sub>1</sub> , P9 <sub>7</sub> at the time when UARTi is in use	RxDi (input)	SCLi (input/output)
7	P7 <sub>2</sub> , P9 <sub>0</sub> , P9 <sub>5</sub> at the time when UARTi is in use	CLKi	P7 <sub>2</sub> , P9 <sub>0</sub> , P9 <sub>5</sub>
8	DMA1 factor at the time when 1 1 0 1 is assigned to the DMA request factor selection bits	UARTi reception	Acknowledgment detection (ACK)
9	Noise filter width	15ns	50ns
10	Reading P7 <sub>1</sub> , P9 <sub>1</sub> , P9 <sub>7</sub>	Reading the terminal when 0 is assigned to the direction register	Reading the terminal regardless of the value of the direction register
11	Initial value of UARTi output	H level (when 0 is assigned to the CLK polarity select bit)	The value set in latch P7 <sub>0</sub> , P9 <sub>2</sub> , P9 <sub>6</sub> when the port is selected (Note 3)

Note 1: Make the settings given below when I<sup>2</sup>C mode is in use.

Set 0 1 0 in bits 2, 1, 0 of the UARTi transmission/reception mode register.

Disable the RTS/CTS function. Choose the MSB First function.

Note 2: Follow the steps given below to switch from a factor to another.

1. Disable the interrupt of the corresponding number.
2. Switch from a factor to another.
3. Reset the interrupt request flag of the corresponding number.
4. Set an interrupt level of the corresponding number.

Note 3: Set an initial value of SDA transmission output when IIC mode (IIC mode select bit = "1") is valid and serial I/O is invalid.



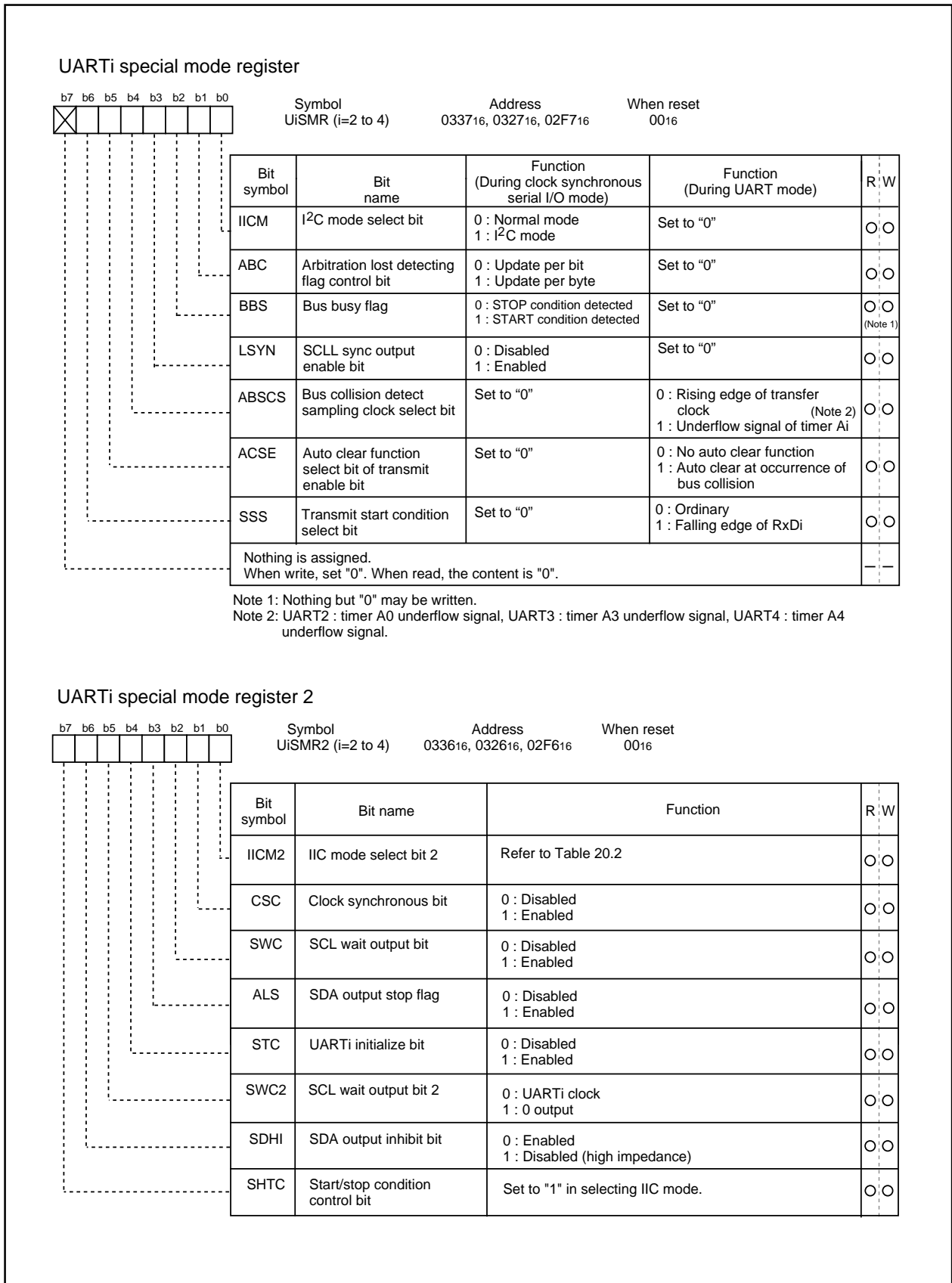


Figure 20.1 UART2 special mode register

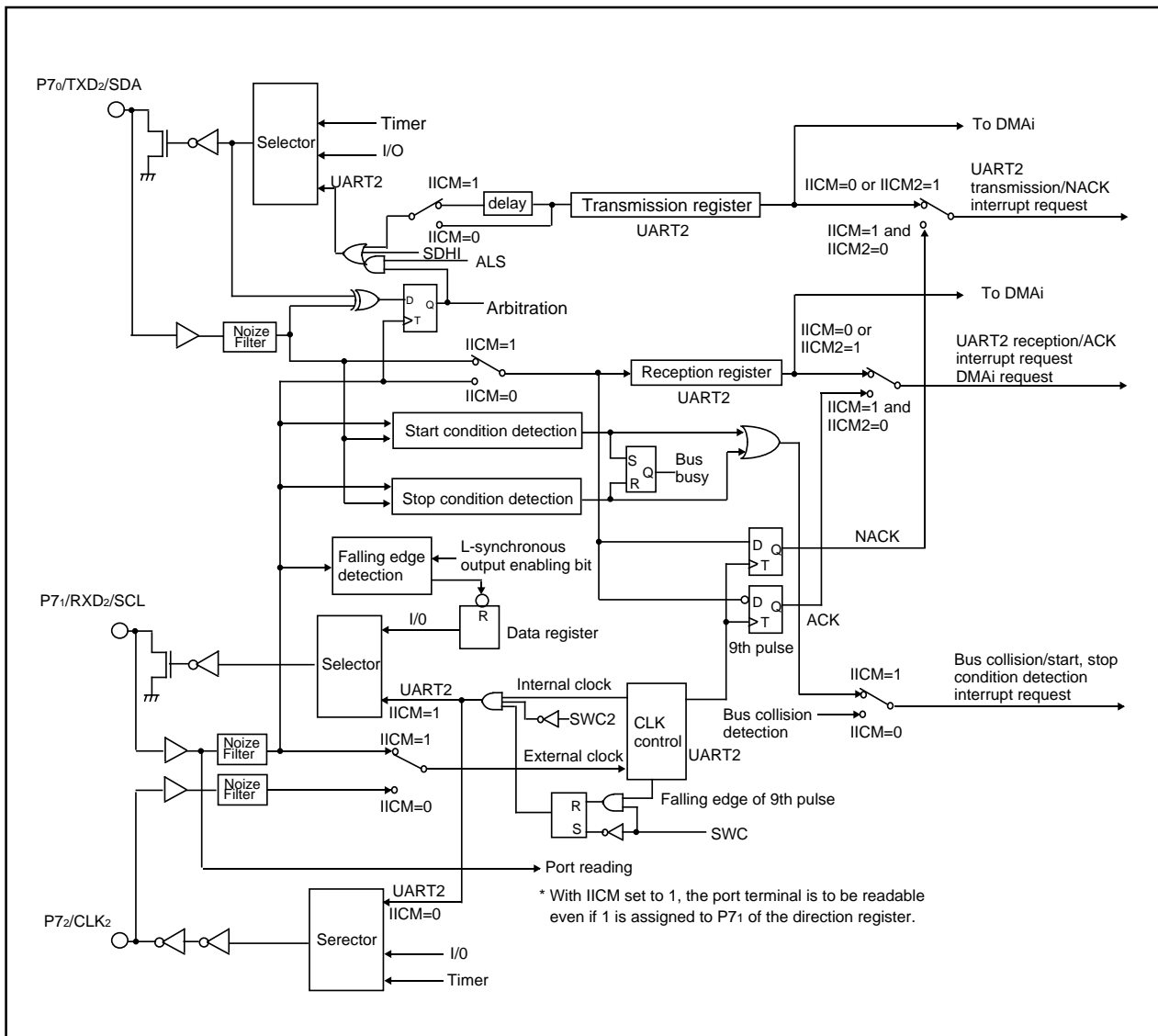


Figure 20.2 Functional block diagram for I<sup>2</sup>C mode

Figure 20.2 is a block diagram of the IIC bus interface.

To explain the control bit of the IIC bus interface, UART2 is used as an example.

**UART2 Special Mode Register (Address 0337<sub>16</sub>)**

Bit 0 is the IIC mode select bit. When set to “1”, ports P7<sub>0</sub>, P7<sub>1</sub> and P7<sub>2</sub> operate respectively as the SDA<sub>2</sub> data transmission-reception pin, SCL<sub>2</sub> clock I/O pin and port P7<sub>2</sub>. A delay circuit is added to SDA<sub>2</sub> transmission output, therefore after SCL<sub>2</sub> is sufficiently L level, SDA<sub>2</sub> output changes. Port P7<sub>1</sub> (SCL<sub>2</sub>) is designed to read pin level regardless of the content of the port direction register. SDA<sub>2</sub> transmission output is initially set to port P7<sub>0</sub> in this mode. Furthermore, interrupt factors for the bus collision detection interrupt, UART2 transmission interrupt and UART2 reception interrupt change respectively to the start/stop condition detection interrupts, acknowledge non-detection interrupt and acknowledge detection interrupt.

The start condition detection interrupt is generated when the fall at the SDA2 pin (P70) is detected while the SCL2 pin (P71) is in the H state. The stop condition detection interrupt is generated when the rise at the SDA2 pin (P70) is detected while the SCL2 pin (P71) is in the H state.

The acknowledge non-detection interrupt is generated when the H level at the SDA2 pin is detected at the 9th rise of the transmission clock.

The acknowledge detection interrupt is generated when the L level at the SDA2 pin is detected at the 9th rise of the transmission clock. Also, DMA transfer can be started when the acknowledge is detected if UART2 transmission is selected as the DMAi request factor.

Bit 2 is the bus busy flag. It is set to "1" when the start condition is detected, and reset to "0" when the stop condition is detected.

Bit 1 is the arbitration lost detection flag control bit. Arbitration detects a conflict between data transmitted at SCL2 rise and data at the SDA2 pin. This detection flag is allocated to bit 11 in UART2 transmission buffer register (address 033E16). It is set to "1" when a conflict is detected. With the arbitration lost detection flag control bit, it can be selected to update the flag in units of bits or bytes. When this bit is set to "1", update is set to units of byte. If a conflict is then detected, the arbitration lost detection flag control bit will be set to "1" at the 9th rise of the clock. When updating in units of byte, always clear ("0" interrupt) the arbitration lost detection flag control bit after the 1st byte has been acknowledged but before the next byte starts transmitting.

Bit 3 is the SCL2 L synchronization output enable bit. When this bit is set to "1", the P71 data register is set to "0" in sync with the L level at the SCL2 pin.

Bit 4 is the bus collision detection sampling clock select bit. The bus collision detection interrupt is generated when RxDi and TxDi level do not conflict with one another. When this bit is "0", a conflict is detected in sync with the rise of the transfer clock. When this bit is "1", detection is made when timer Ai (timer A0 with UART2, timer A3 with UART3 and timer A4 with UART4) underflows. Operation is shown in Figure 20.3.

Bit 5 is the transmission enable bit automatic clear select bit. By setting this bit to "1", the transmission bit is automatically reset to "0" when the bus collision detection interrupt factor bit is "1" (when a conflict is detected).

Bit 6 is the transmission start condition select bit. By setting this bit to "1", TxDi transmission starts in sync with the falling at the RxDi pin.

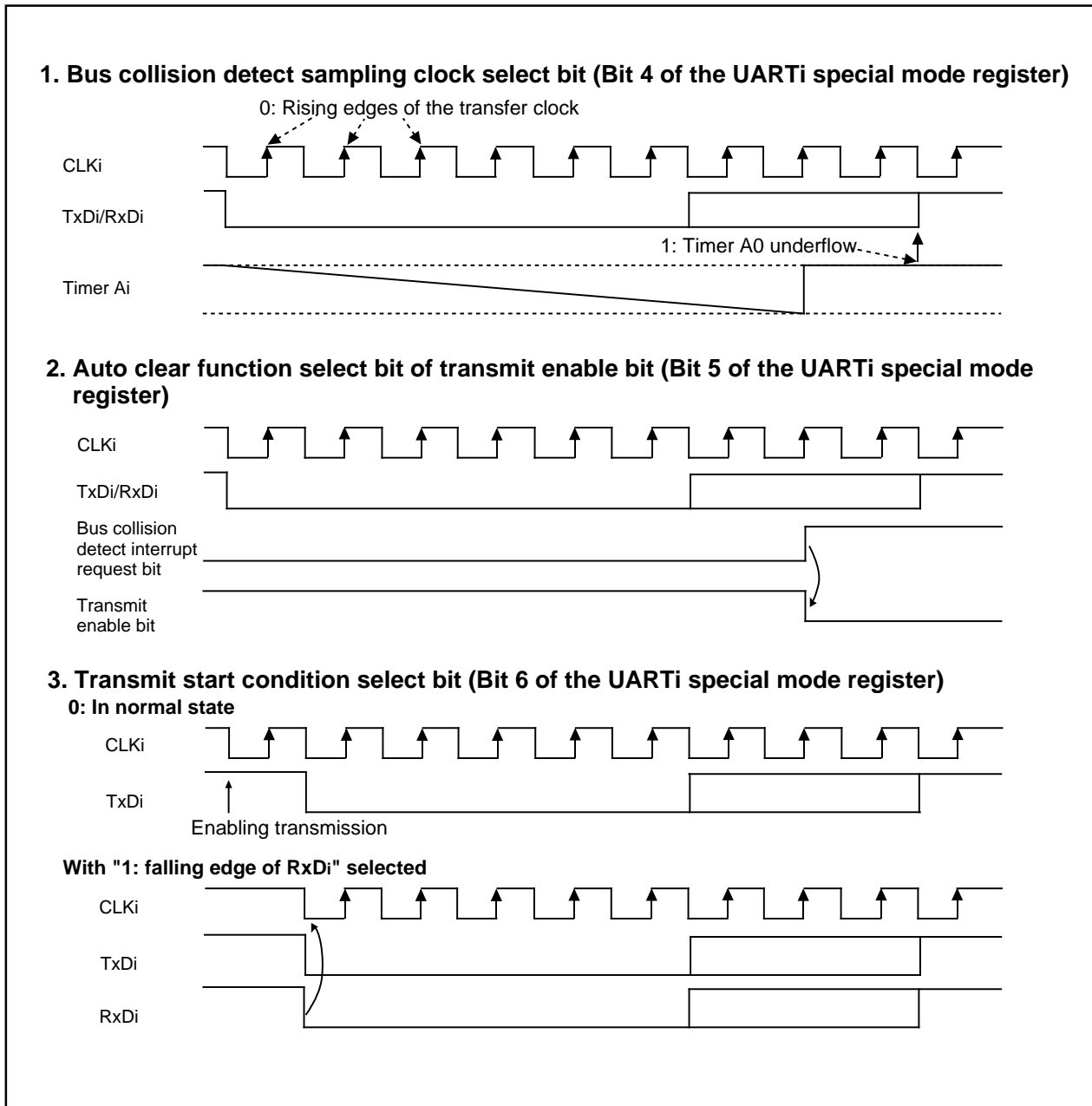


Figure 20.3 Some other functions added

**UARTi Special Mode Register 2(i=2 to 4) (Address 0336<sub>16</sub>,0326<sub>16</sub>,02F6<sub>16</sub>)**

Bit 0 is the IIC mode select bit 2. Table 20.2 gives control changes by bit when the IIC mode select bit is "1". Start and stop condition detection timing characteristics are shown in Figure 20.4. Always set bit 7 (start/stop condition control bit) to "1".

Bit 1 is the clock synchronization bit. When this bit is set to "1", if the rise edge is detected at pin SCLi while the internal SCL is H level, the internal SCL is changed to L level, the UARTi bit rate generator value is reloaded and the L sector count starts. Also, while the SCLi pin is L level, if the internal SCL changes from L level to H, the count stops. If the SCLi pin is H level, counting restarts. Because of this function, the UARTi transmission-reception clock takes the AND condition for the internal SCL and SCLi pin signals. This function operates from the clock half period before the 1st rise of the UARTi clock to the 9th rise. To use this function, select the internal clock as the transfer clock.

Bit 2 is the SCL wait output bit. When this bit is set to "1", output from the SCLi pin is fixed to L level at the clock's 9th rise. When set to "0", the L output lock is released.

Bit 3 is the SDA output stop bit. When this bit is set to "1", an arbitration lost is generated. If the arbitration lost detection flag is "1", the SDAi pin simultaneously becomes high impedance.

Bit 4 is the UARTi initialize bit. While this bit is set to "1", the following operations are performed when the start condition is detected.

1. The transmission shift register is initialized and the content of the transmission register is transmitted to the transmission shift register. As such, transmission starts with the 1st bit of the next input clock. However, the UARTi output value remains the same as when the start condition was detected, without changing from when the clock is input to when the 1st bit of data is output.
2. The reception shift register is initialized and reception starts with the 1st bit of the next input clock.
3. The SCL wait output bit is set to "1". As such, the SCLi pin becomes L level at the rise of the 9th bit of the clock.

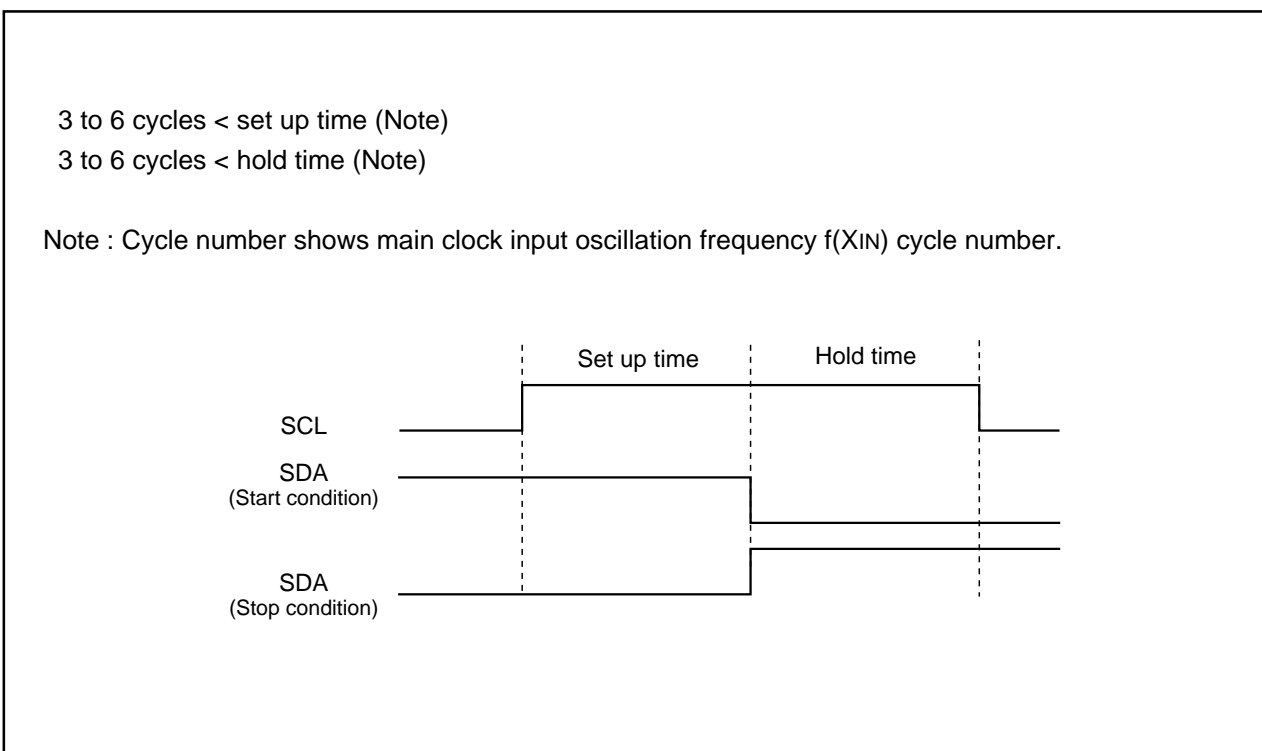
When UART transmission-reception has been started using this function, the content of the transmission buffer available flag does not change. Also, to use this function, select an external clock as the transfer clock.

Bit 5 is SCL wait output bit 2. When this bit is set to "1" and serial I/O has been selected, an L level can be forcefully output from the SCLi pin even during UART operation. When this bit is set to "0", the L output from the SCLi pin is canceled and the UARTi clock is input and output.

Bit 6 is the SDA output disable bit. When this bit is set to "1", the SDAi pin is forcefully made high impedance. To overwrite this bit, do so at the rise of the UARTi transfer clock. The arbitration lost detection flag may be set.

**Table 20.2 Functions changed by I<sup>2</sup>C mode select bit 2**

Function	IICM2 = 0	IICM2 = 1
Interrupt no. 33, 35, 37 factor	Acknowledge not detect (NACK)	UART2 transfer (rising edge of )
Interrupt no. 34, 36, 38 factor	Acknowledge detect (ACK)	Acknowledge detect (ACK)
DMA factor	Acknowledge detect (ACK)	Acknowledge detect (ACK)
Data transfer timing from UARTi (i = 2 to 4) receive shift register to receive buffer	Rising edge of the last bit of receive clock	Rising edge of the last bit of receive clock
UARTi(i = 2 to 4) receive / ACK interrupt request generation timing	Rising edge of the last bit of receive clock	Rising edge of the last bit of receive clock



**Figure 20.4 Start/stop condition detect timing characteristics**

**UARTi Special Mode Register 3(i=2 to 4 )(Address 0335<sub>16</sub>,0325<sub>16</sub>,02F5<sub>16</sub>)**

Bits 5 to 7 are the SDAi digital delay setting bits. By setting these bits, it is possible to turn the SDAi delay OFF or set the f(XIN) delay to 2 to 8 cycles.

## (2) Serial Interface Special Function

UART 3 and UART4 can control communications on the serial bus using the  $\overline{SS}_i$  input pins (Figure 20.5). The master outputting the transfer clock transfers data to the slave inputting the transfer clock. In this case, in order to prevent a data collision on the bus, the master floats the output pin of other slaves/masters using the  $\overline{SS}_i$  input pins. Figure 20.6 shows the structure of UARTi special mode register 3 (addresses 0325<sub>16</sub> and 02F5<sub>16</sub> [i = 3 or 4]) which controls this mode.  $\overline{SS}_i$  input pins function between the master and slave are as follows.

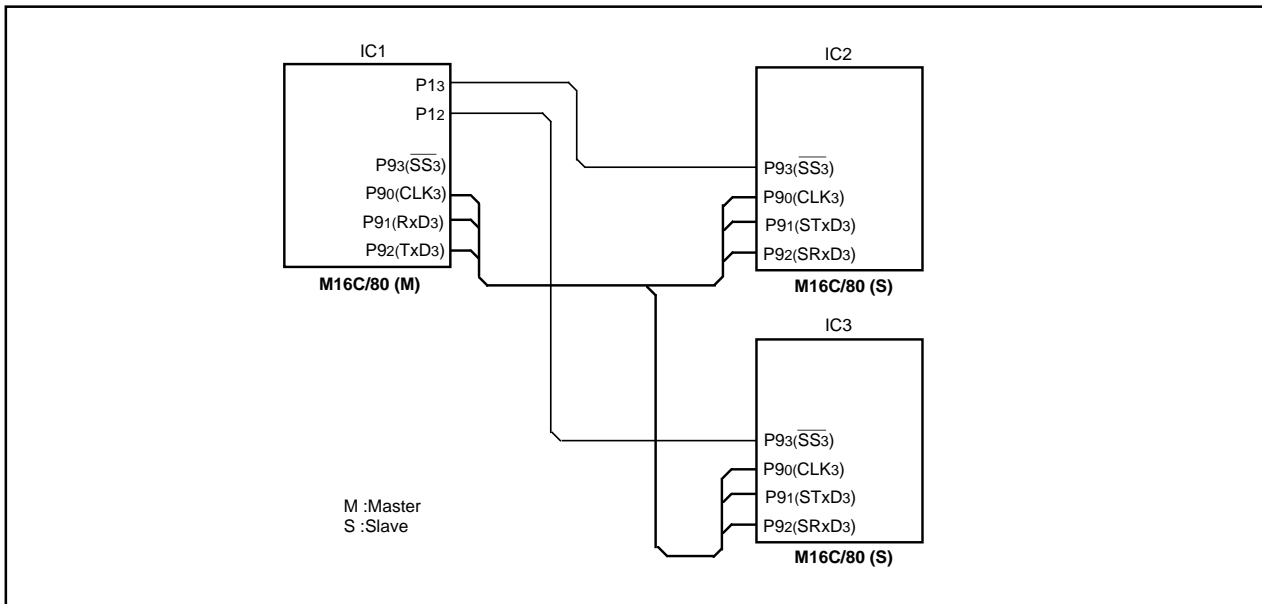


Figure 20.5 Serial bus communication control example using the  $\overline{SS}_i$  input pins

### < Slave Mode (STxDi and SRxDi are selected, DINC = 1) >

When an H level signal is input to an  $\overline{SS}_i$  input pin, the STxDi and SRxDi pins both become high impedance, hence clock input is ignored. When an "L" level signal is input to an  $\overline{SS}_i$  input pin, clock input becomes effective and serial communications are enabled. (i = 3 or 4)

### < Master Mode (TxDi and RxDi are selected, DINC = 0) >

The  $\overline{SS}_i$  input pins are used with a multiple master system. When an  $\overline{SS}_i$  input pin is H level, transmission has priority and serial communications are enabled. When an L signal is input to an  $\overline{SS}_i$  input pin, another master exists, and the TxDi, RxDi and CLKi pins all become high impedance. Moreover, the trouble error interrupt request bit becomes "1". Communications do not stop even when a trouble error is generated during communications. To stop communications, set bits 0, 1 and 2 of the UARTi transmission-reception mode register (address 0328<sub>16</sub> and 02F8<sub>16</sub> [i = 3 or 4]) to "0".

The trouble error interrupt is used by both the bus collision interrupt and start/stop condition detection interrupts, but the trouble error interrupt itself can be selected by setting bit 0 of UARTi special mode register 3 (address 0325<sub>16</sub> and 02F5<sub>16</sub> [i = 3 or 4]) to "1".

When the trouble error flag is set to "0", output is restored to the clock output and data output pins. In the master mode, if an  $\overline{SS}_i$  input pin is H level, "0" can be written for the trouble error flag. When an  $\overline{SS}_i$  input pin is L level, "0" cannot be written for the trouble error flag. In the slave mode, the "0" can be written for the trouble error flag regardless of the input to the  $\overline{SS}_i$  input pins.

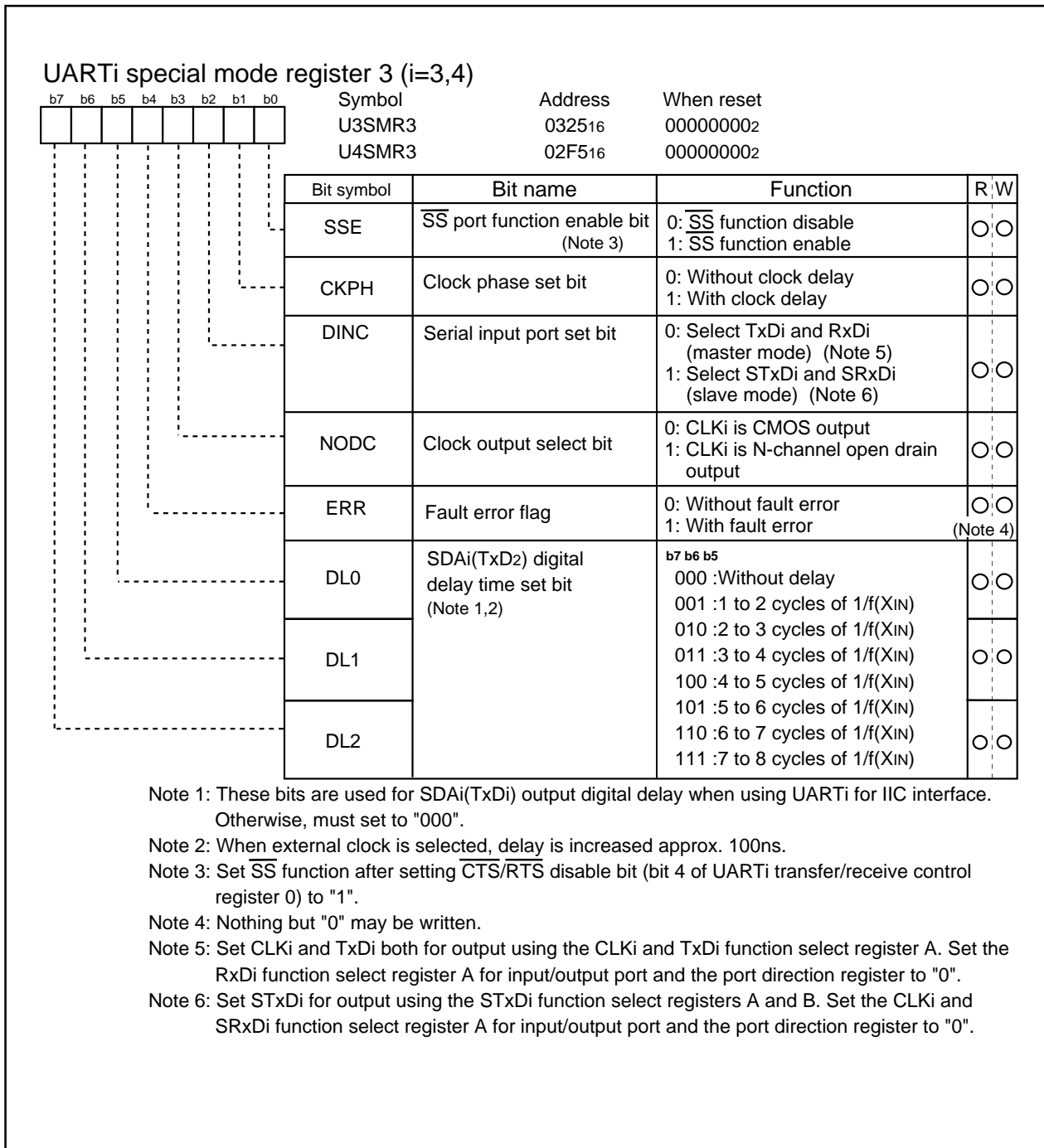


Figure 20.6 UARTi special mode register 3 (i=3,4)



### ■ Clock Phase Setting

With bit 1 of UARTi special mode register 3 (addresses 0325<sub>16</sub> and 02F5<sub>16</sub> [i = 3 or 4]) and bit 6 of UARTi transmission-reception control register 0 (addresses 032C<sub>16</sub> and 02FC<sub>16</sub> [i = 3 or 4]), four combinations of transfer clock phase and polarity can be selected.

Bit 6 of UARTi transmission-reception control register 0 (addresses 032C<sub>16</sub> and 02FC<sub>16</sub> [i = 3 or 4]) sets transfer clock polarity, whereas bit 1 of UARTi special mode register 3 (addresses 0325<sub>16</sub> and 02F5<sub>16</sub> [i = 3 or 4]) sets transfer clock phase.

Transfer clock phase and polarity must be the same between the master and slave involved in the transfer.

#### < Master (Internal Clock) (DINC = 0) >

Figure 20.7 shows the transmission and reception timing.

#### < Slave (External Clock) (DINC = 1) >

- With "0" for bit 1 (CKPH) of UARTi special mode register 3 (addresses 0325<sub>16</sub> and 02F5<sub>16</sub> [i = 3 or 4]), when an SSi input pin is H level, output data is high impedance. When an SSi input pin is L level, the serial transmission start condition is satisfied, though output is indeterminate. After that, serial transmission is synchronized with the clock. Figure 20.8 shows the timing.
- With "1" for bit 1 (CKPH) of UARTi special mode register 3 (addresses 0325<sub>16</sub> and 02F5<sub>16</sub> [i = 3 or 4]), when an SSi input pin is H level, output data is high impedance. When an SSi input pin is L level, the first data is output. After that, serial transmission is synchronized with the clock. Figure 20.9 shows the timing.

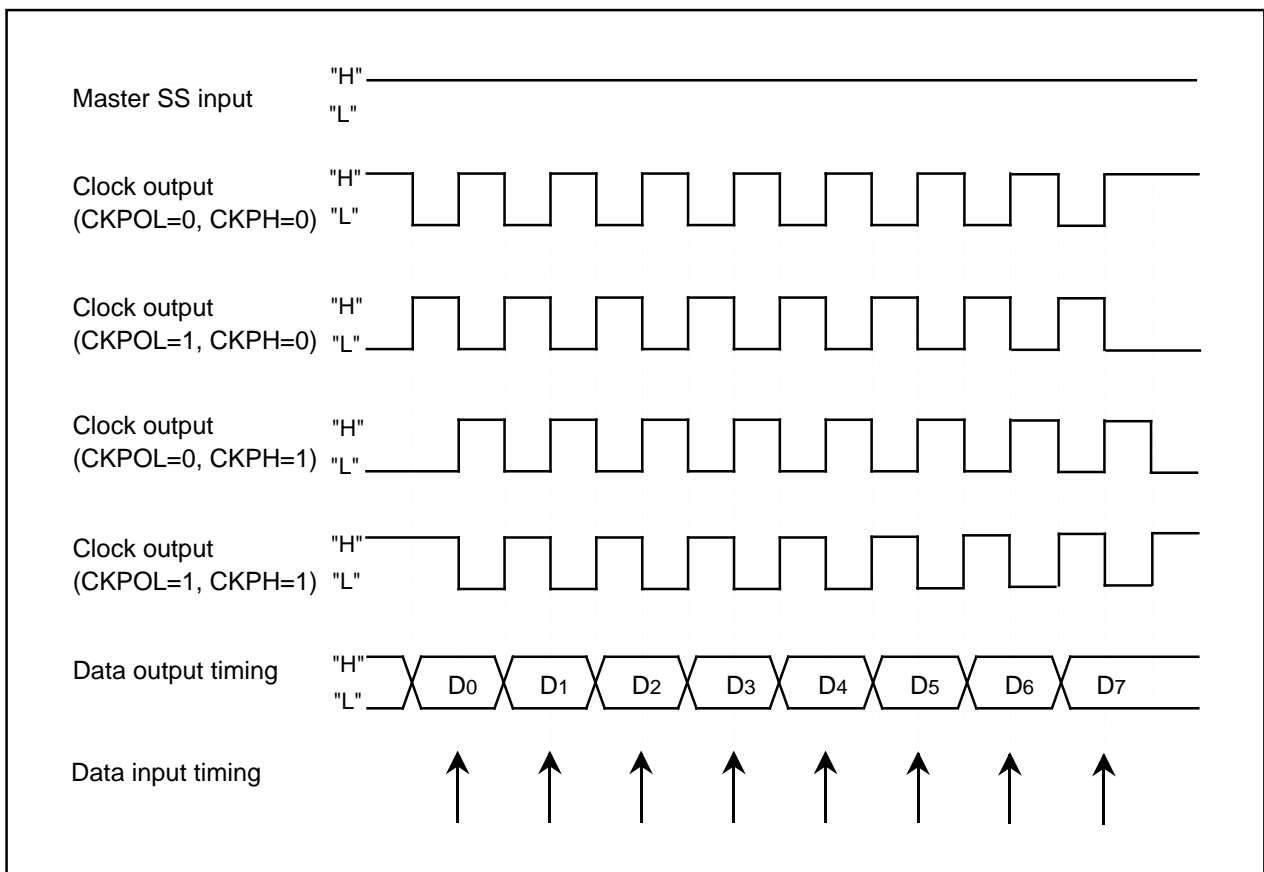


Figure 20.7 The transmission and reception timing in master mode (internal clock)

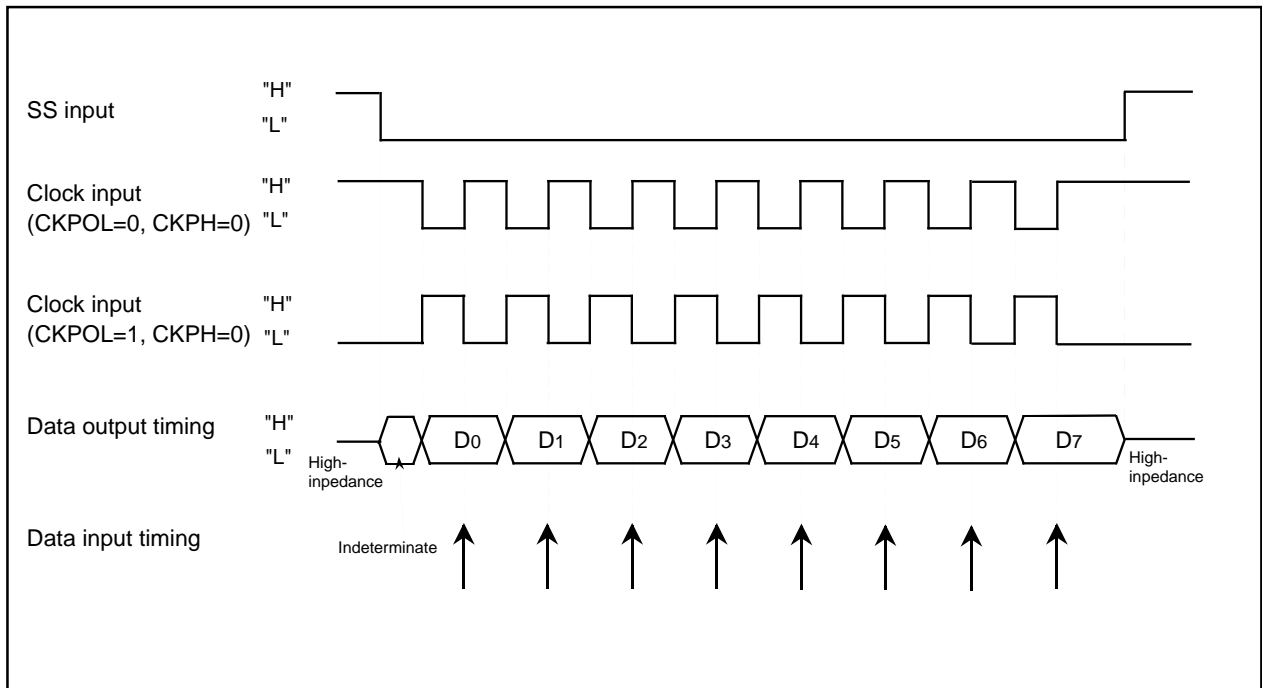


Figure 20.8 The transmission and reception timing (CKPH=0) in slave mode (external clock)

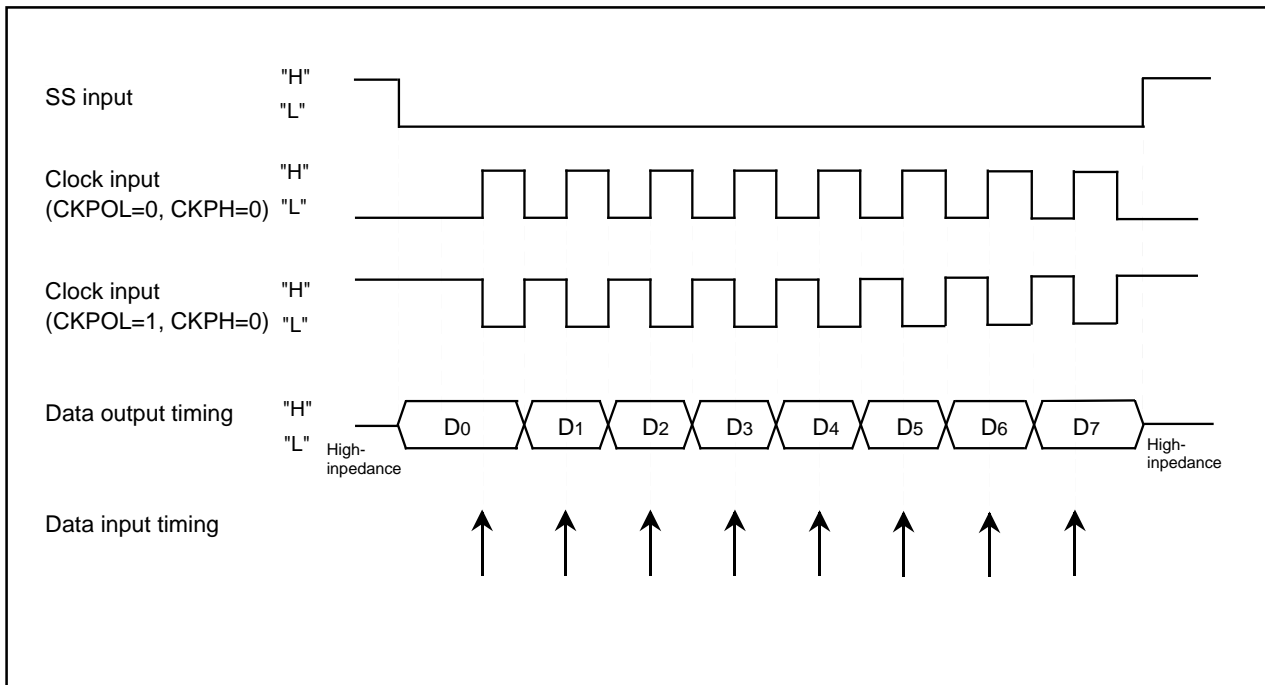


Figure 20.9 The transmission and reception timing (CKPH=1) in slave mode (external clock)

## 21. A/D Converter

The A/D converter consists of one 10-bit successive approximation A/D converter circuit with a capacitive coupling amplifier. Pins P100 to P107, P95, and P96 also function as the analog signal input pins. The direction registers of these pins for A/D conversion must therefore be set to input. The Vref connect bit (bit 5 at address 0397<sub>16</sub>) can be used to isolate the resistance ladder of the A/D converter from the reference voltage input pin (VREF) when the A/D converter is not used. Doing so stops any current flowing into the resistance ladder from VREF, reducing the power dissipation. When using the A/D converter, start A/D conversion only after setting bit 5 of 0397<sub>16</sub> to connect VREF.

The result of A/D conversion is stored in the A/D registers of the selected pins. When set to 10-bit precision, the low 8 bits are stored in the even addresses and the high 2 bits in the odd addresses. When set to 8-bit precision, the low 8 bits are stored in the even addresses.

Table 21.1 shows the performance of the A/D converter. Figure 21.1 shows the block diagram of the A/D converter, and Figures 21.2 and 21.3 show the A/D converter-related registers.

**Table 21.1 Performance of A/D converter**

Item	Performance
Method of A/D conversion	Successive approximation (capacitive coupling amplifier)
Analog input voltage (Note 1)	0V to AVCC (VCC)
Operating clock f <sub>AD</sub> (Note 2)	VCC = 5V      f <sub>AD</sub> , f <sub>AD</sub> /2, f <sub>AD</sub> /4      f <sub>AD</sub> =f(XIN)
	VCC = 3V      f <sub>AD</sub> /2, f <sub>AD</sub> /4      f <sub>AD</sub> =f(XIN)
Resolution	8-bit or 10-bit (selectable)
Absolute precision	VCC = 5V
	<ul style="list-style-type: none"> <li>• 8-bit resolution ±2LSB</li> <li>• 10-bit resolution ±3LSB</li> </ul> However, when using AN <sub>0</sub> to AN <sub>7</sub> in the mode which external operation amp is connected : ±7LSB VCC = 3V <ul style="list-style-type: none"> <li>• Without sample and hold function (8-bit resolution) ±2LSB</li> </ul>
Operating modes	One-shot mode, repeat mode, single sweep mode, repeat sweep mode 0, and repeat sweep mode 1
Analog input pins	8 pins (AN <sub>0</sub> to AN <sub>7</sub> ) + 2 pins (ANEX <sub>0</sub> and ANEX <sub>1</sub> )
A/D conversion start condition	<ul style="list-style-type: none"> <li>• Software trigger A/D conversion starts when the A/D conversion start flag changes to "1"</li> <li>• External trigger (can be retriggered) A/D conversion starts when the A/D conversion start flag is "1" and the <math>\overline{\text{ADTRG/P97}}</math> input changes from "H" to "L"</li> </ul>
Conversion speed per pin	<ul style="list-style-type: none"> <li>• Without sample and hold function 8-bit resolution: 49 f<sub>AD</sub> cycles, 10-bit resolution: 59 f<sub>AD</sub> cycles</li> <li>• With sample and hold function 8-bit resolution: 28 f<sub>AD</sub> cycles, 10-bit resolution: 33 f<sub>AD</sub> cycles</li> </ul>

Note 1: Does not depend on use of sample and hold function.

Note 2: When f(XIN) is over 10 MHz, the f<sub>AD</sub> frequency must be under 10 MHz by dividing.

Without sample and hold function, set the f<sub>AD</sub> frequency to 250kHz min.

With the sample and hold function, set the f<sub>AD</sub> frequency to 1MHz min.

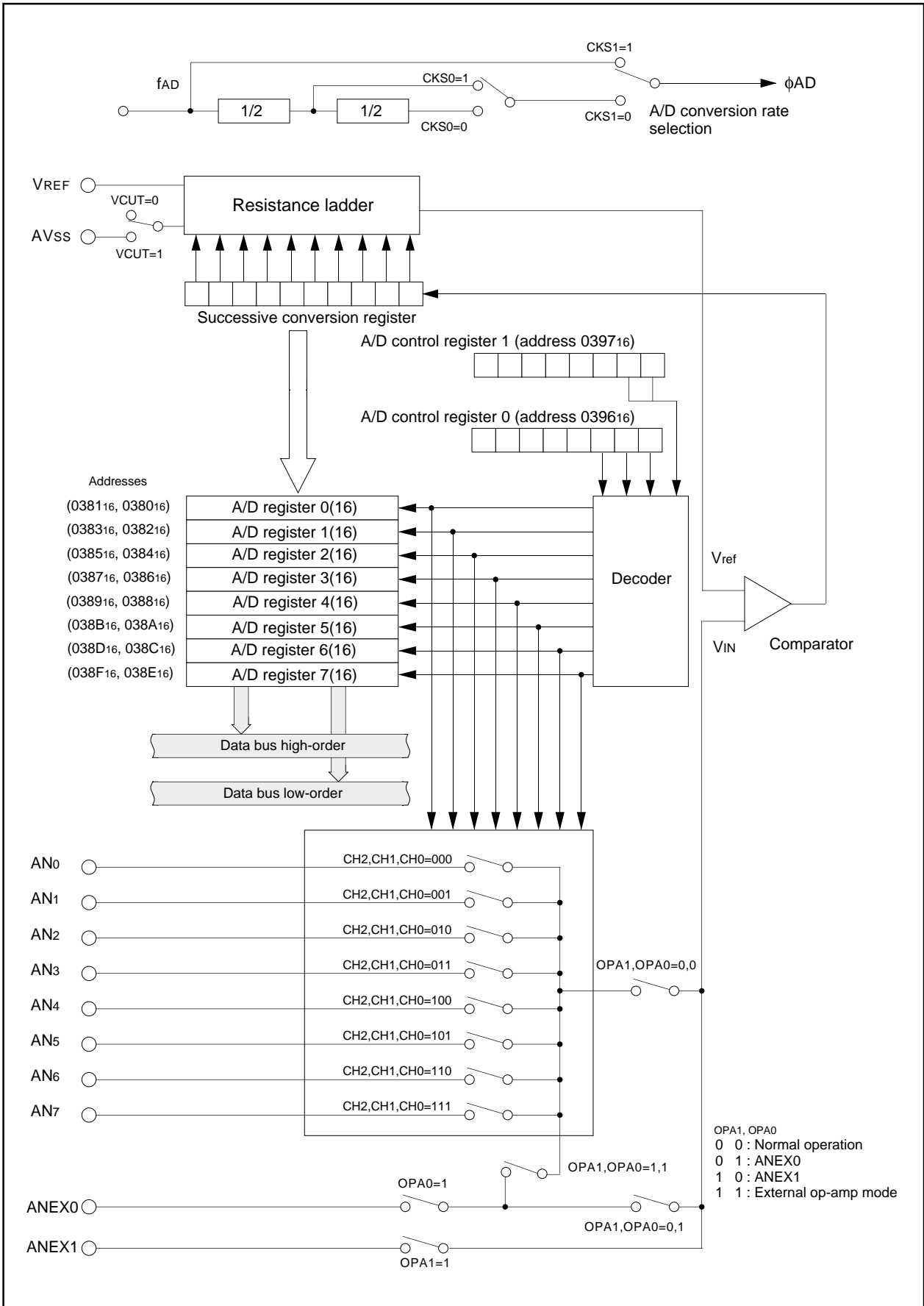


Figure 21.1 Block diagram of A/D converter

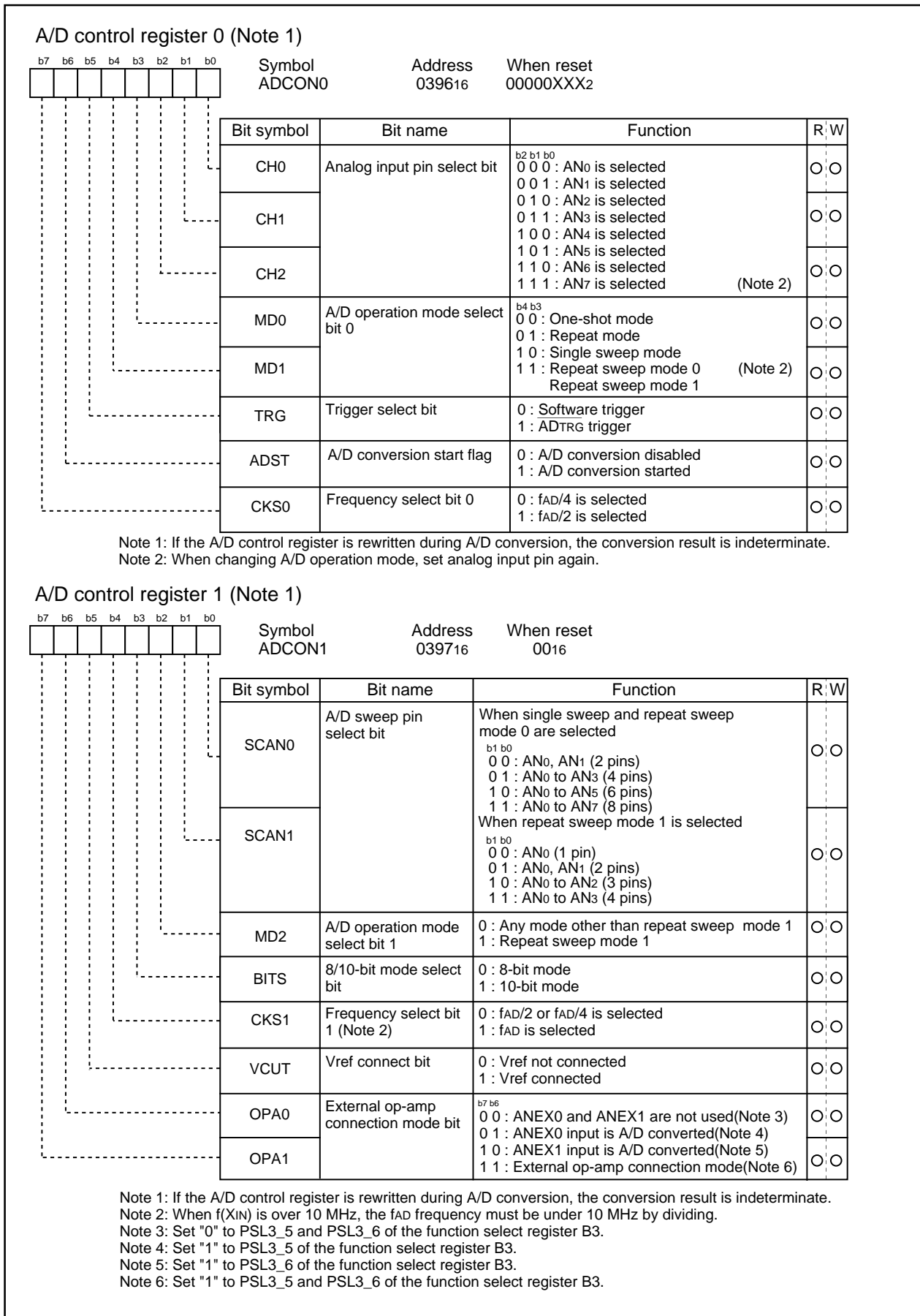


Figure 21.2 A/D converter-related registers (1)

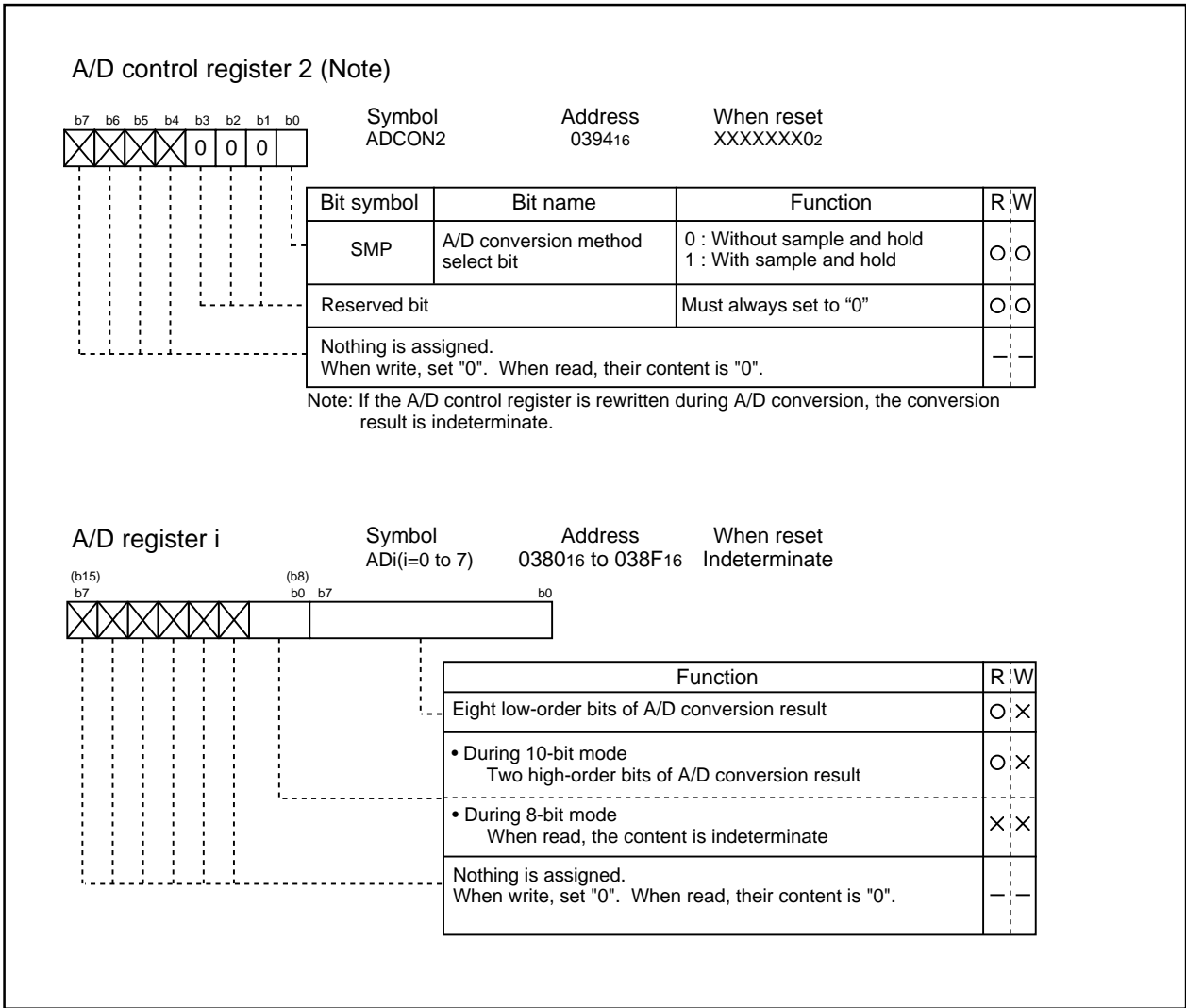


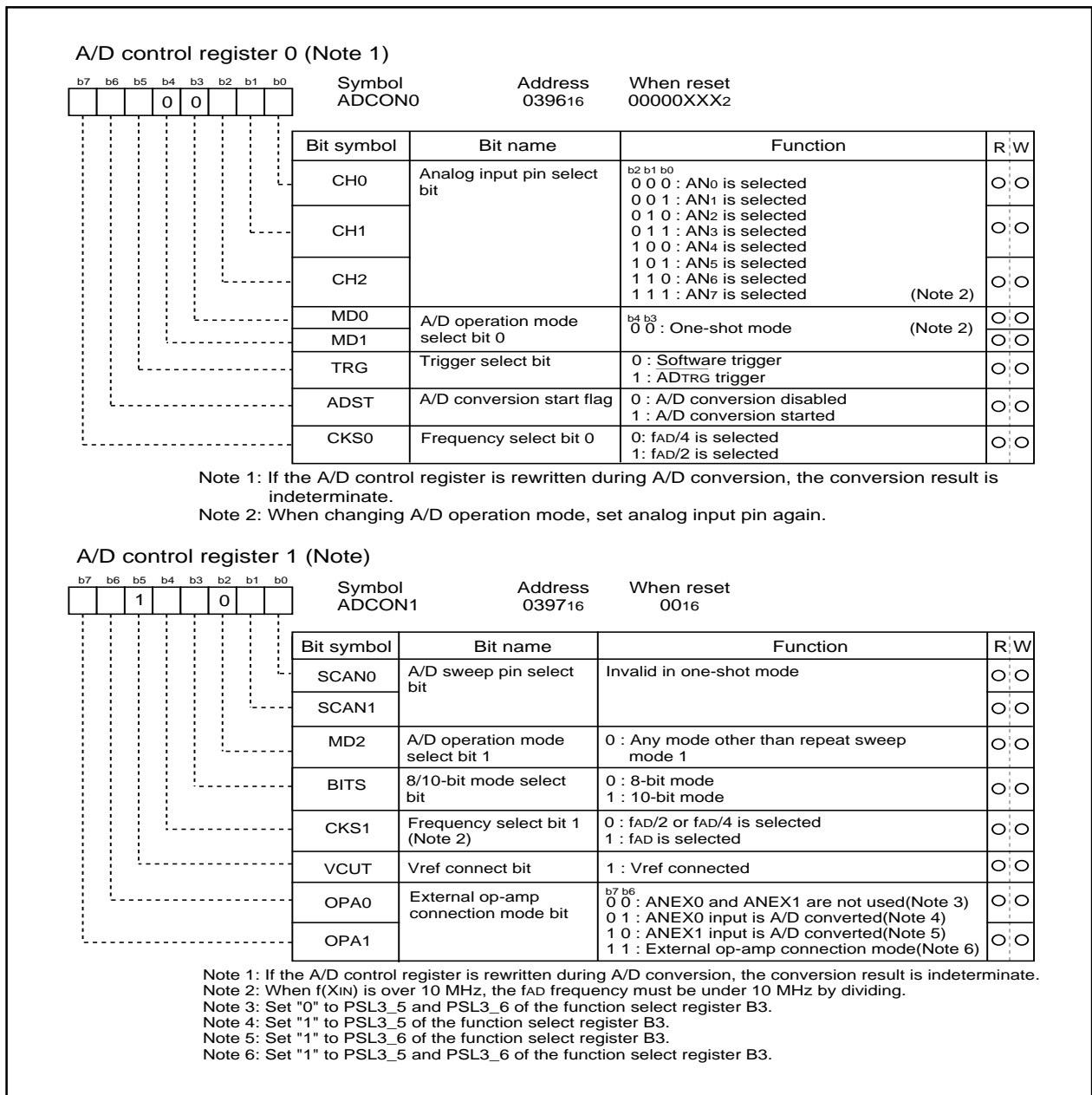
Figure 21.3 A/D converter-related registers (2)

**(1) One-shot mode**

In one-shot mode, the pin selected using the analog input pin select bit is used for one-shot A/D conversion. Table 21.2 shows the specifications of one-shot mode. Figure 21.4 shows the A/D control register in one-shot mode.

**Table 21.2 One-shot mode specifications**

Item	Specification
Function	The pin selected by the analog input pin select bit is used for one A/D conversion
Start condition	Writing "1" to A/D conversion start flag
Stop condition	<ul style="list-style-type: none"> <li>End of A/D conversion (A/D conversion start flag changes to "0", except when external trigger is selected)</li> <li>Writing "0" to A/D conversion start flag</li> </ul>
Interrupt request generation timing	End of A/D conversion
Input pin	One of AN0 to AN7, as selected
Reading of result of A/D converter	Read A/D register corresponding to selected pin



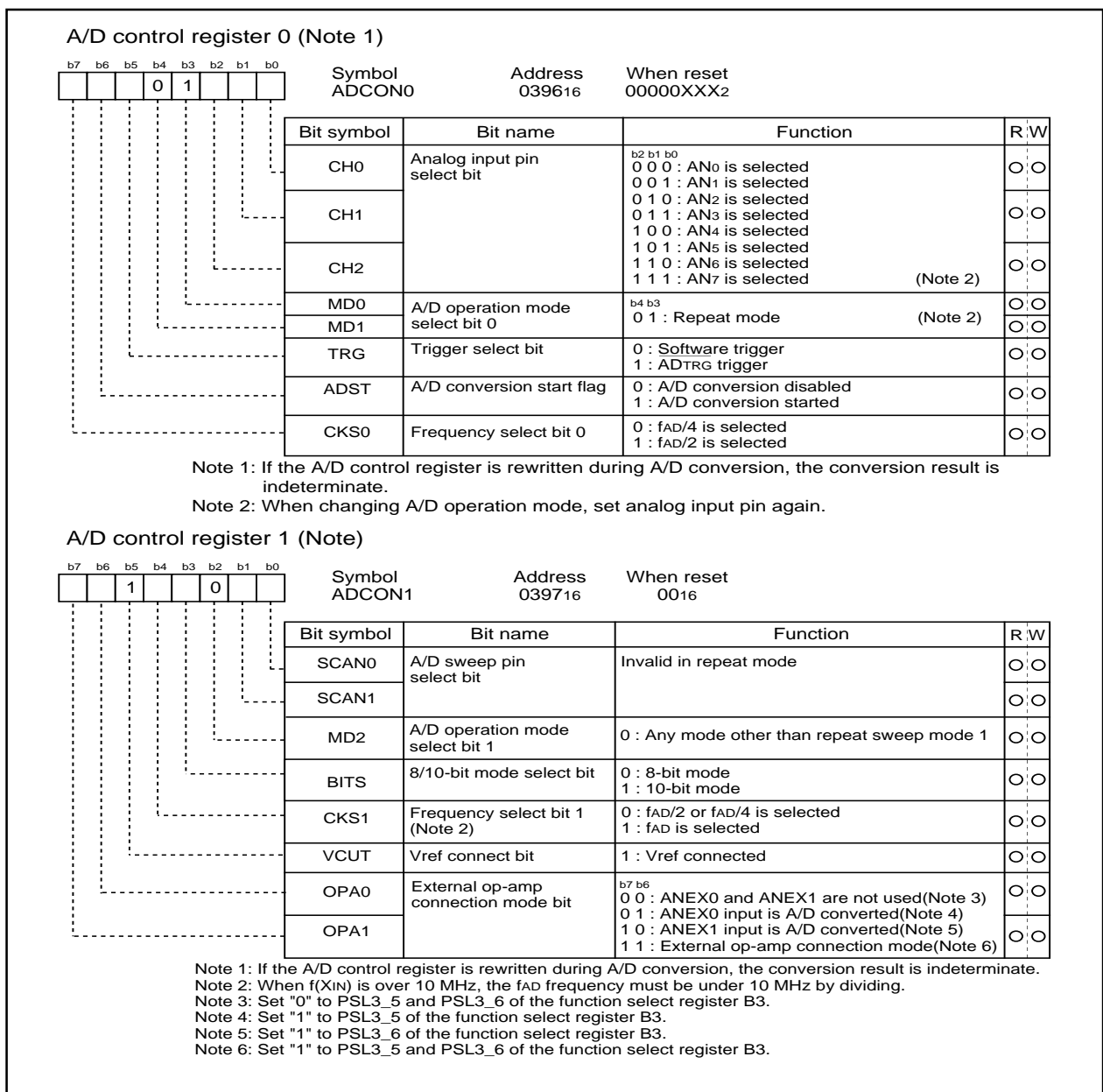
**Figure 21.4 A/D conversion register in one-shot mode**

**(2) Repeat mode**

In repeat mode, the pin selected using the analog input pin select bit is used for repeated A/D conversion. Table 21.3 shows the specifications of repeat mode. Figure 21.5 shows the A/D control register in repeat mode.

**Table 21.3 Repeat mode specifications**

Item	Specification
Function	The pin selected by the analog input pin select bit is used for repeated A/D conversion
Start condition	Writing "1" to A/D conversion start flag
Stop condition	Writing "0" to A/D conversion start flag
Interrupt request generation timing	None generated
Input pin	One of AN0 to AN7, as selected
Reading of result of A/D converter	Read A/D register corresponding to selected pin (at any time)



**Figure 21.5 A/D conversion register in repeat mode**

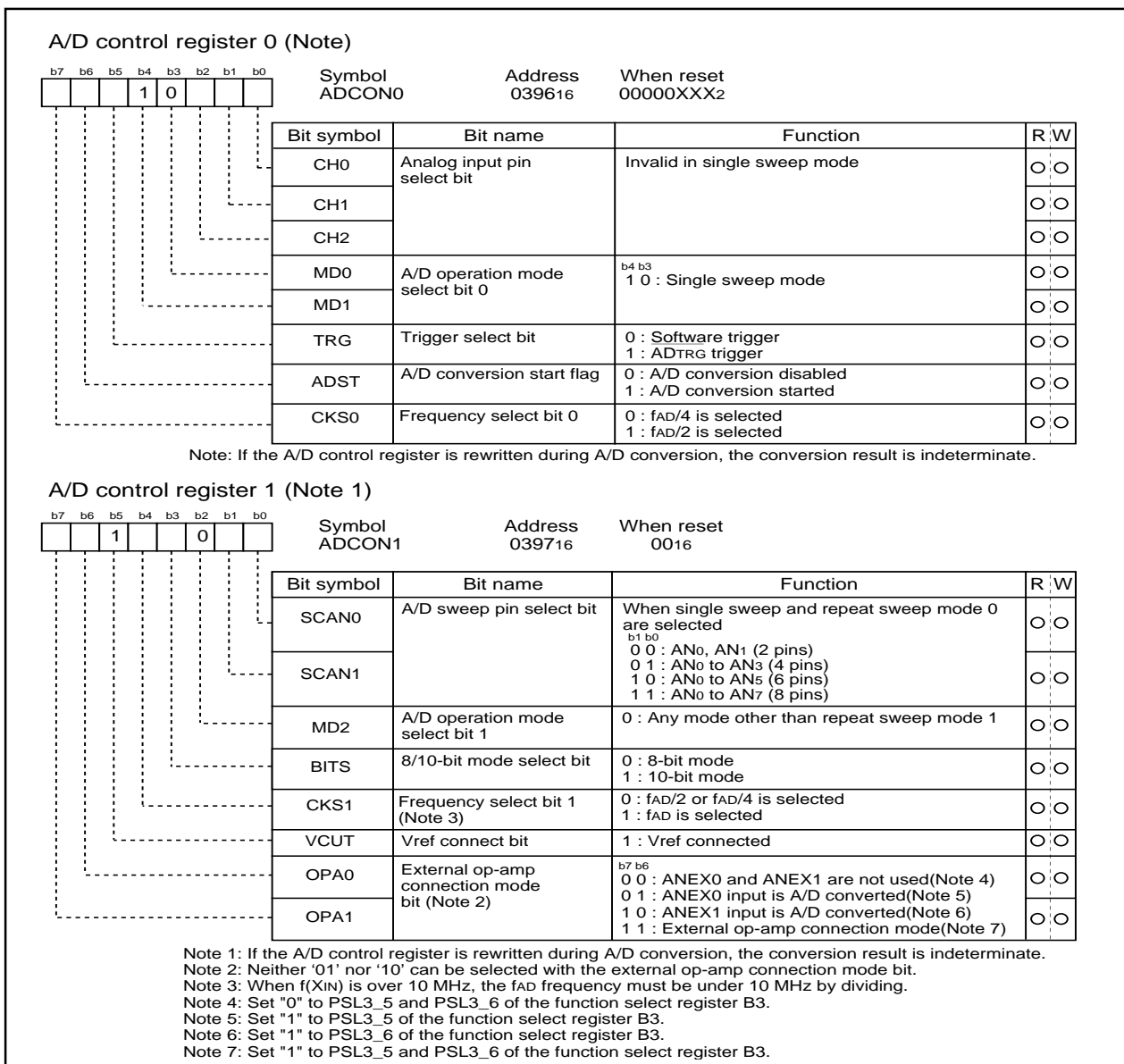


### (3) Single sweep mode

In single sweep mode, the pins selected using the A/D sweep pin select bit are used for one-by-one A/D conversion. Table 21.4 shows the specifications of single sweep mode. Figure 21.6 shows the A/D control register in single sweep mode.

**Table 21.4 Single sweep mode specifications**

Item	Specification
Function	The pins selected by the A/D sweep pin select bit are used for one-by-one A/D conversion
Start condition	Writing "1" to A/D converter start flag
Stop condition	<ul style="list-style-type: none"> <li>End of A/D conversion (A/D conversion start flag changes to "0", except when external trigger is selected)</li> <li>Writing "0" to A/D conversion start flag</li> </ul>
Interrupt request generation timing	End of A/D conversion
Input pin	AN0 and AN1 (2 pins), AN0 to AN3 (4 pins), AN0 to AN5 (6 pins), or AN0 to AN7 (8 pins)
Reading of result of A/D converter	Read A/D register corresponding to selected pin



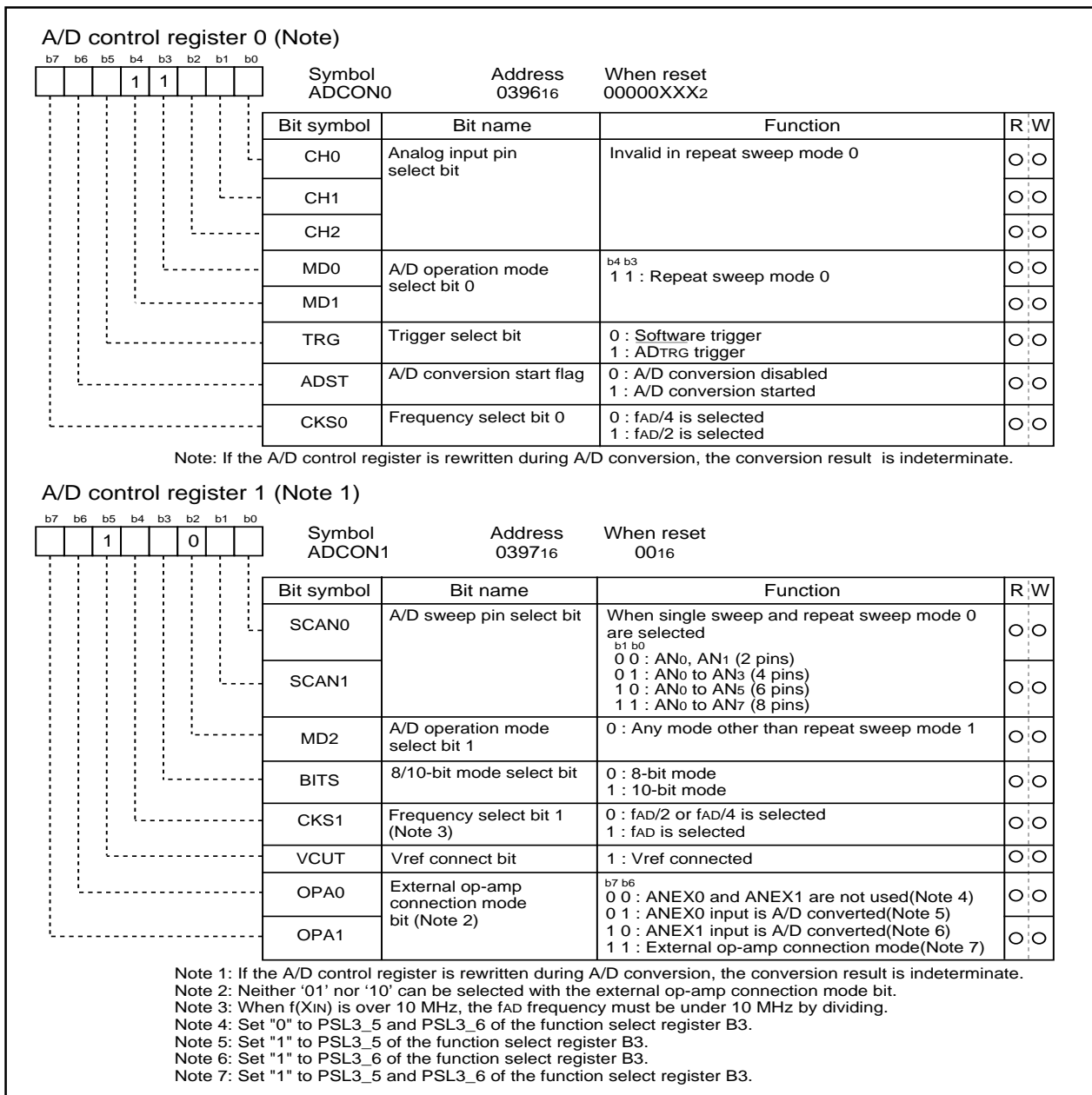
**Figure 21.6 A/D conversion register in single sweep mode**

**(4) Repeat sweep mode 0**

In repeat sweep mode 0, the pins selected using the A/D sweep pin select bit are used for repeat sweep A/D conversion. Table 21.5 shows the specifications of repeat sweep mode 0. Figure 21.7 shows the A/D control register in repeat sweep mode 0.

**Table 21.5 Repeat sweep mode 0 specifications**

Item	Specification
Function	The pins selected by the A/D sweep pin select bit are used for repeat sweep A/D conversion
Start condition	Writing "1" to A/D conversion start flag
Stop condition	Writing "0" to A/D conversion start flag
Interrupt request generation timing	None generated
Input pin	AN0 and AN1 (2 pins), AN0 to AN3 (4 pins), AN0 to AN5 (6 pins), or AN0 to AN7 (8 pins)
Reading of result of A/D converter	Read A/D register corresponding to selected pin (at any time)



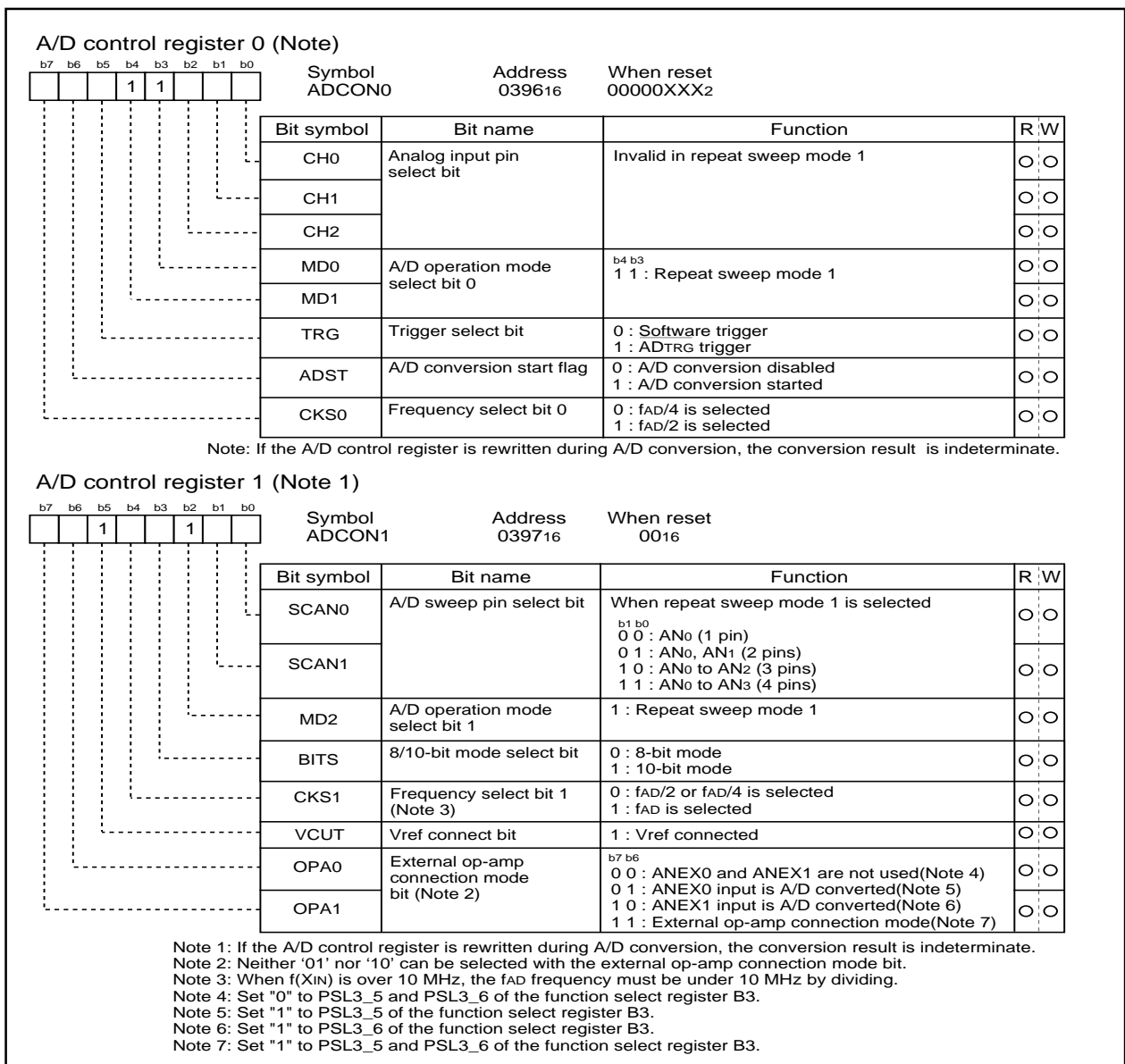
**Figure 21.7 A/D conversion register in repeat sweep mode 0**

**(5) Repeat sweep mode 1**

In repeat sweep mode 1, all pins are used for A/D conversion with emphasis on the pin or pins selected using the A/D sweep pin select bit. Table 21.6 shows the specifications of repeat sweep mode 1. Figure 21.8 shows the A/D control register in repeat sweep mode 1.

**Table 21.6 Repeat sweep mode 1 specifications**

Item	Specification
Function	All pins perform repeat sweep A/D conversion, with emphasis on the pin or pins selected by the A/D sweep pin select bit Example : AN <sub>0</sub> selected AN <sub>0</sub> → AN <sub>1</sub> → AN <sub>0</sub> → AN <sub>2</sub> → AN <sub>0</sub> → AN <sub>3</sub> , etc
Start condition	Writing "1" to A/D conversion start flag
Stop condition	Writing "0" to A/D conversion start flag
Interrupt request generation timing	None generated
Input pin	AN <sub>0</sub> to AN <sub>7</sub>
With emphasis on the pin	AN <sub>0</sub> (1 pin), AN <sub>0</sub> and AN <sub>1</sub> (2 pins), AN <sub>0</sub> to AN <sub>2</sub> (3 pins), AN <sub>0</sub> to AN <sub>3</sub> (4 pins)
Reading of result of A/D converter	Read A/D register corresponding to selected pin (at any time)



**Figure 21.8 A/D conversion register in repeat sweep mode 1**

### (a) Sample and hold

Sample and hold is selected by setting bit 0 of the A/D control register 2 (address 0394<sub>16</sub>) to "1". When sample and hold is selected, the rate of conversion of each pin increases. As a result, a 28 fAD cycle is achieved with 8-bit resolution and 33 fAD with 10-bit resolution. Sample and hold can be selected in all modes. However, in all modes, be sure to specify before starting A/D conversion whether sample and hold is to be used.

### (b) Extended analog input pins

In one-shot mode and repeat mode, the input via the extended analog input pins ANEX0 and ANEX1 can also be converted from analog to digital.

When bit 6 of the A/D control register 1 (address 0397<sub>16</sub>) is "1" and bit 7 is "0", input via ANEX0 is converted from analog to digital. The result of conversion is stored in A/D register 0.

When bit 6 of the A/D control register 1 (address 0397<sub>16</sub>) is "0" and bit 7 is "1", input via ANEX1 is converted from analog to digital. The result of conversion is stored in A/D register 1.

Set the related input peripheral function of the function select register B3 to disabled.

### (c) External operation amp connection mode

In this mode, multiple external analog inputs via the extended analog input pins, ANEX0 and ANEX1, can be amplified together by just one operation amp and used as the input for A/D conversion.

When bit 6 of the A/D control register 1 (address 0397<sub>16</sub>) is "1" and bit 7 is "1", input via AN0 to AN7 is output from ANEX0. The input from ANEX1 is converted from analog to digital and the result stored in the corresponding A/D register. The speed of A/D conversion depends on the response of the external operation amp. Do not connect the ANEX0 and ANEX1 pins directly. Figure 21.9 is an example of how to connect the pins in external operation amp mode.

Set the related input peripheral function of the function select register B3 to disabled.

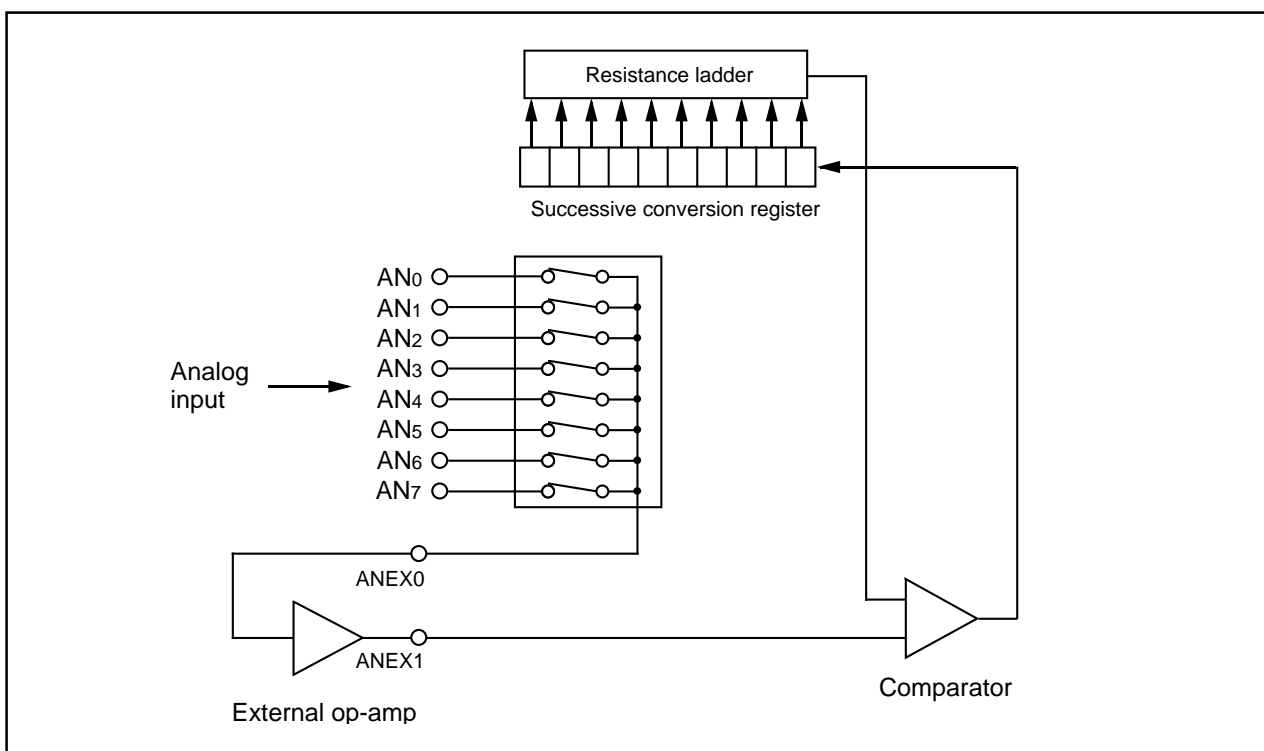


Figure 21.9 Example of external op-amp connection mode

### 22. D/A Converter

This is an 8-bit, R-2R type D/A converter. The microcomputer contains two independent D/A converters of this type. D/A conversion is performed when a value is written to the corresponding D/A register. Bits 0 and 1 (D/A output enable bits) of the D/A control register decide if the result of conversion is to be output. Set the function select register A3 to I/O port, the related input peripheral function of the function select register B3 to disabled and the direction register to input mode. When the D/A output is enabled, the pull-up function of the corresponding port is automatically disabled.

Output analog voltage (V) is determined by a set value (n : decimal) in the D/A register.

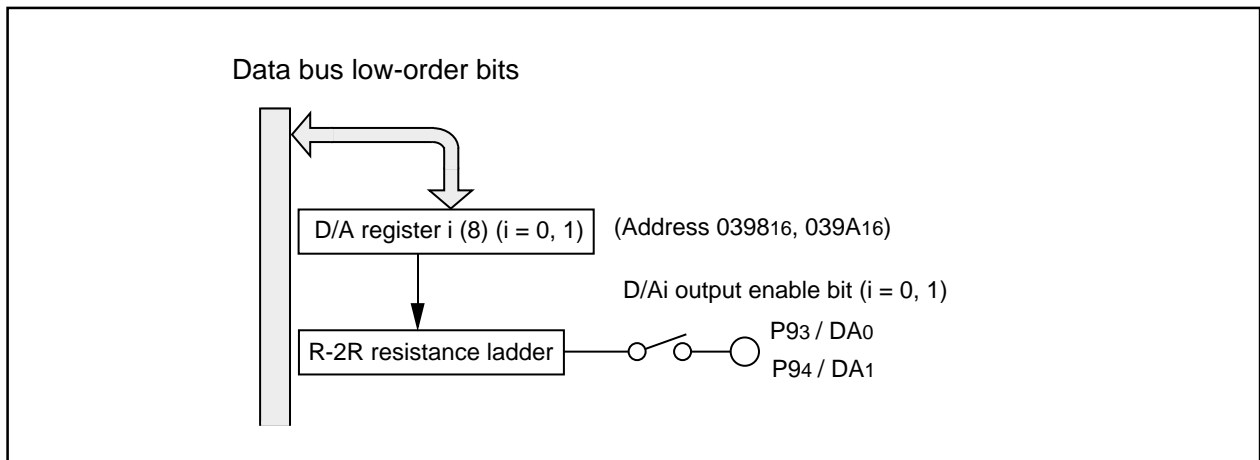
$$V = V_{REF} \times n / 256 \quad (n = 0 \text{ to } 255)$$

V<sub>REF</sub> : reference voltage

Table 22.1 lists the performance of the D/A converter. Figure 22.1 shows the block diagram of the D/A converter. Figure 22.2 shows the D/A control register.

**Table 22.1 Performance of D/A converter**

Item	Performance
Conversion method	R-2R method
Resolution	8 bits
Analog output pin	2 channels



**Figure 22.1 Block diagram of D/A converter**

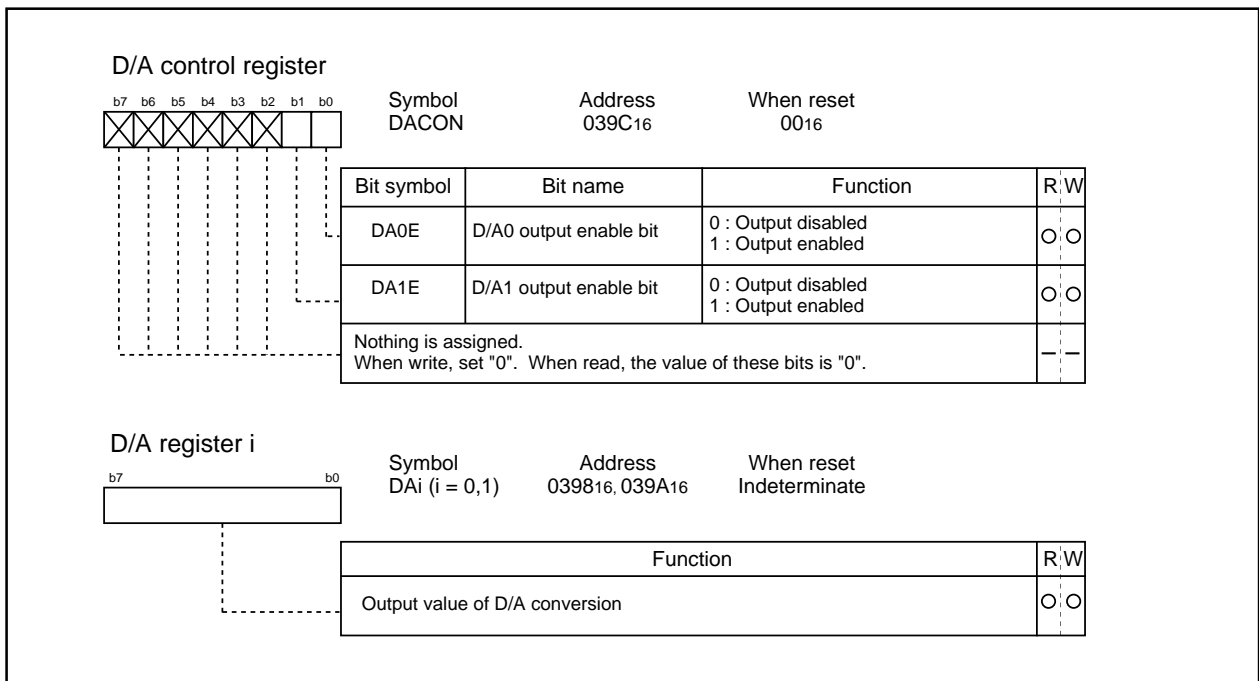


Figure 22.2 D/A control register

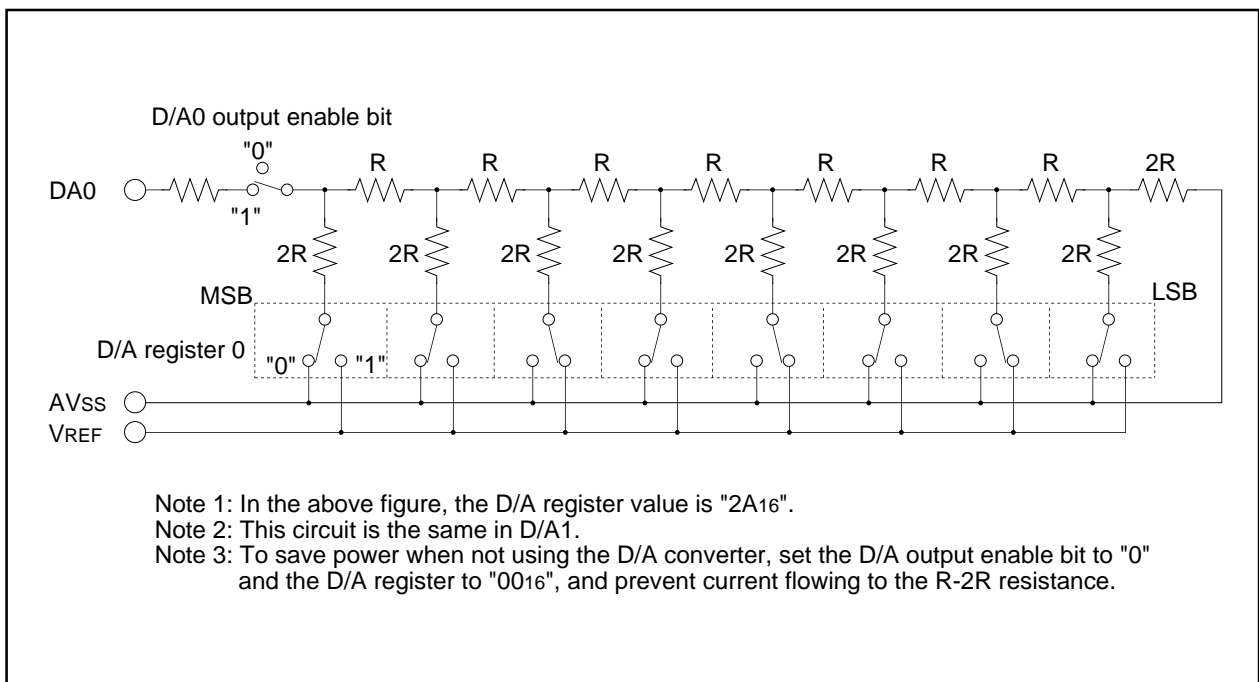


Figure 22.3 D/A converter equivalent circuit

### 23. CRC Calculation Circuit

The Cyclic Redundancy Check (CRC) calculation circuit detects an error in data blocks. The microcomputer uses a generator polynomial of CRC\_CCITT ( $X^{16} + X^{12} + X^5 + 1$ ) to generate CRC code.

The CRC code is a 16-bit code generated for a block of a given data length in multiples of 8 bits. The CRC code is set in a CRC data register each time one byte of data is transferred to a CRC input register after writing an initial value into the CRC data register. Generation of CRC code for one byte of data is completed in two machine cycles.

Figure 23.1 shows the block diagram of the CRC circuit. Figure 23.2 shows the CRC-related registers.

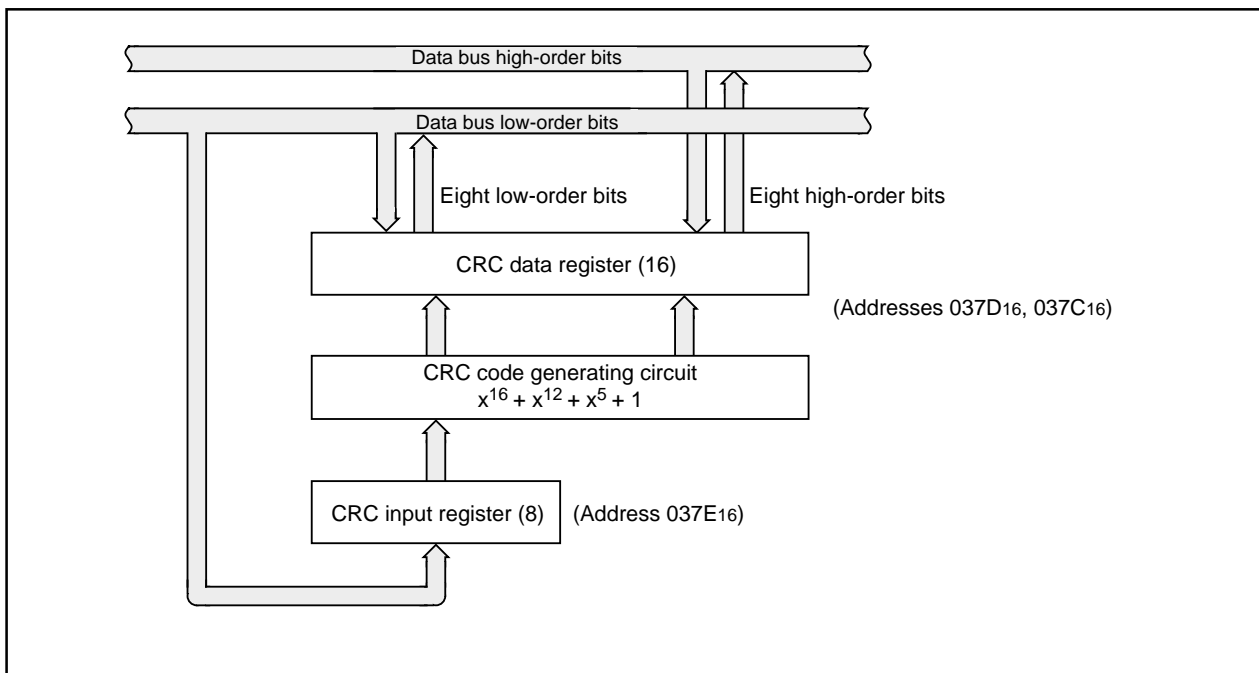


Figure 23.1 Block diagram of CRC circuit

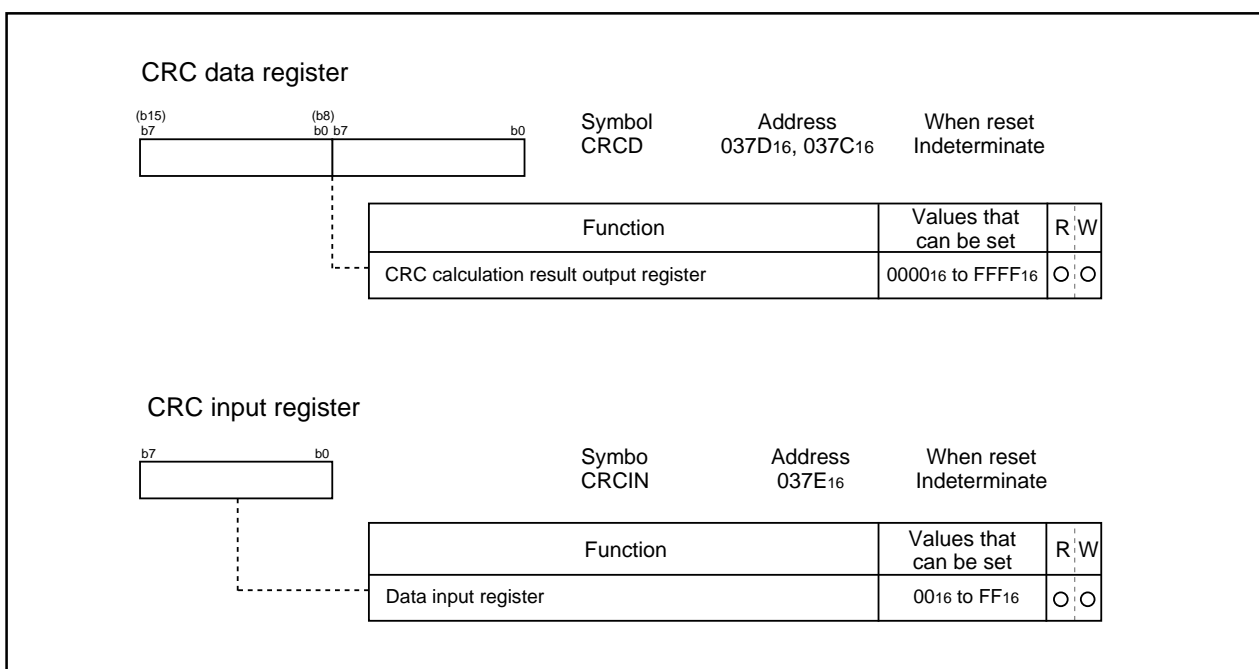


Figure 23.2 CRC-related registers

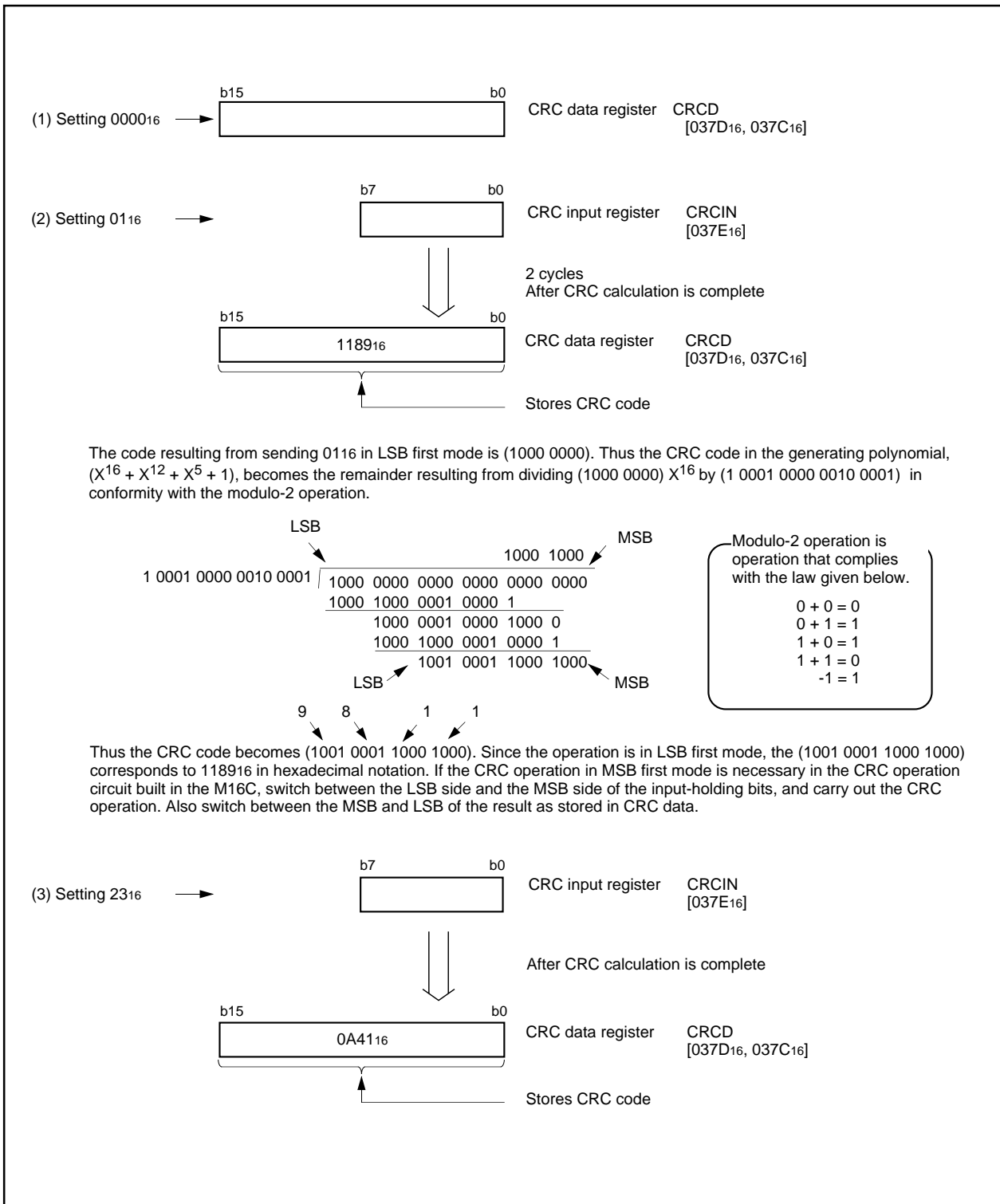


Figure 23.3 CRC example



### 24. XY Converter

XY conversion rotates the 16 x 16 matrix data by 90 degrees. It can also be used to invert the top and bottom of the 16-bit data. Figure 24.1 shows the XY control register.

The Xi and the Yi registers are 16-bit registers. There are 16 of each (where i= 0 to 15).

The Xi and Yi registers are mapped to the same address. The Xi register is a write-only register, while the Yi register is a read-only register. Be sure to access the Xi and Yi registers in 16-bit units from an even address. Operation cannot be guaranteed if you attempt to access these registers in 8-bit units.

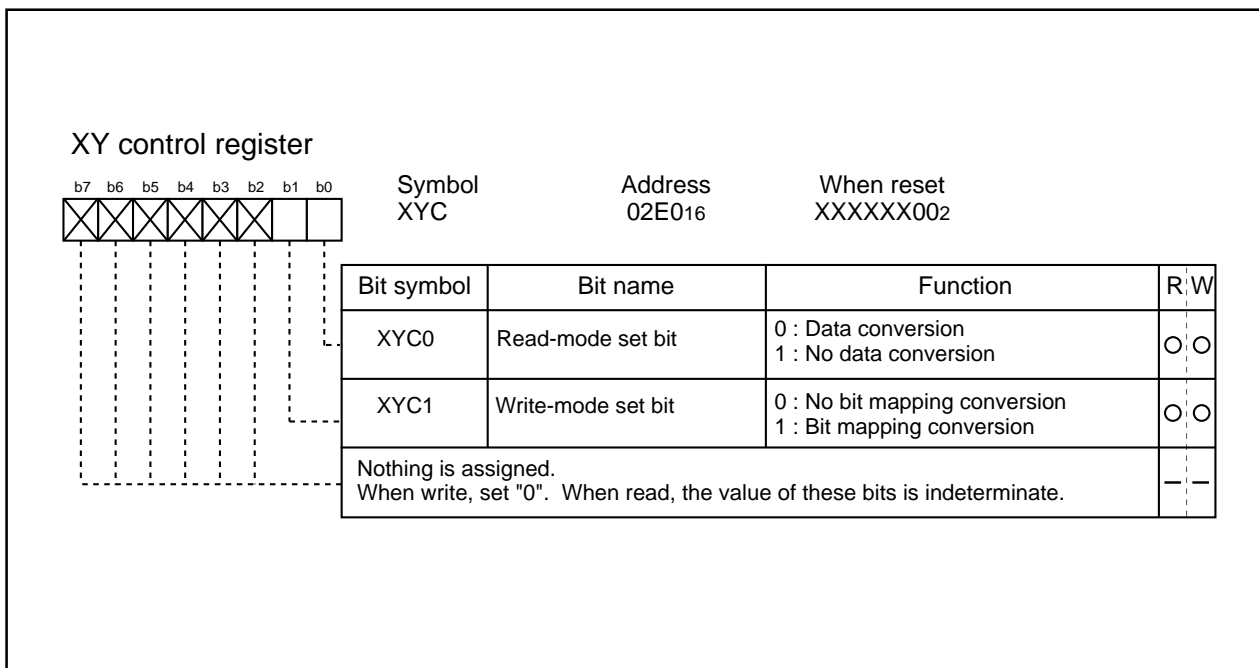


Figure 24.1 XY control register

The reading of the Yi register is controlled by the read-mode set bit (bit 0 at address 02E016).

When the read-mode set bit (bit 0 at address 02E016) is "0", specific bits in the Xi register can be read at the same time as the Yi register is read.

For example, when you read the Y0 register, bit 0 is read as bit 0 of the X0 register, bit 1 is read as bit 0 of the X1 register, ..., bit 14 is read as bit 0 of the X14 register, bit 15 as bit 0 of the X15 register. Similarly, when you read the Y15 register, bit 0 is bit 15 of the X0 register, bit 1 is bit 15 of the X1 register, ..., bit 14 is bit 15 of the X14 register, bit 15 is bit 15 of the X15 register.

Figure 24.2 shows the conversion table when the read mode set bit = "0". Figure 24.3 shows the XY conversion example.

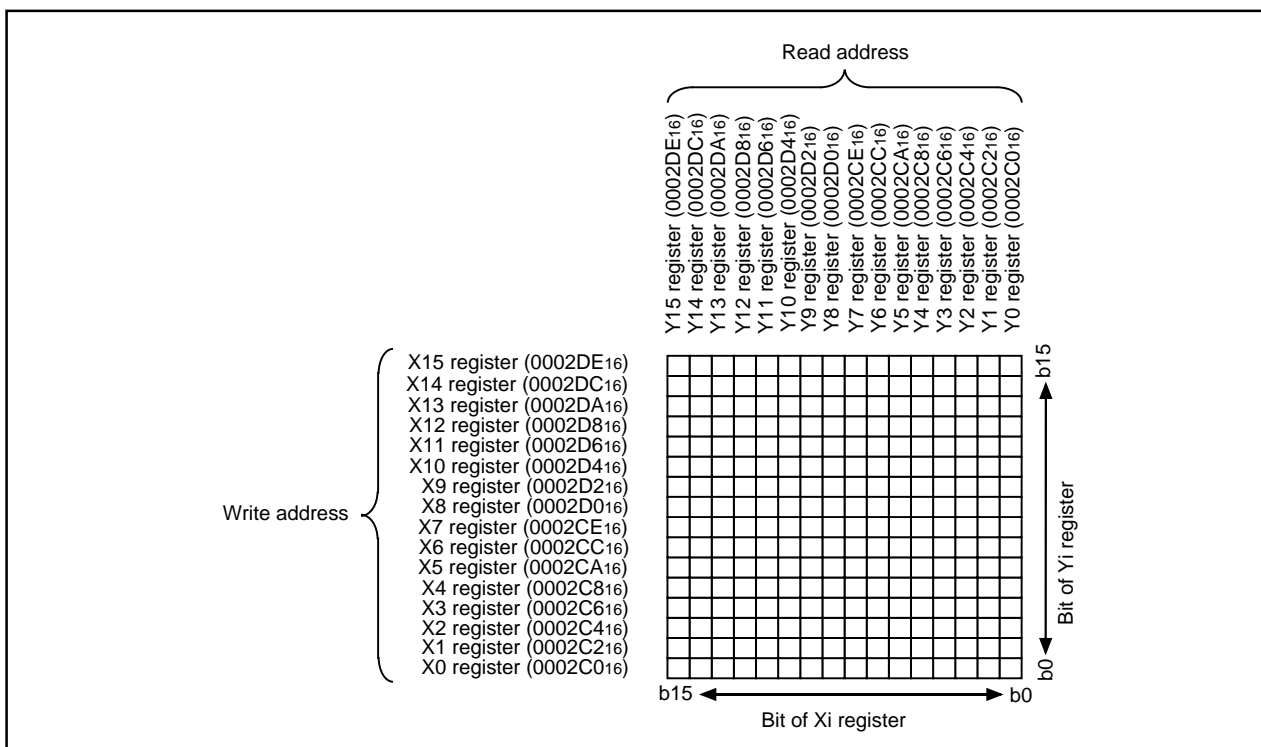


Figure 24.2 Conversion table when the read mode set bit = "0"

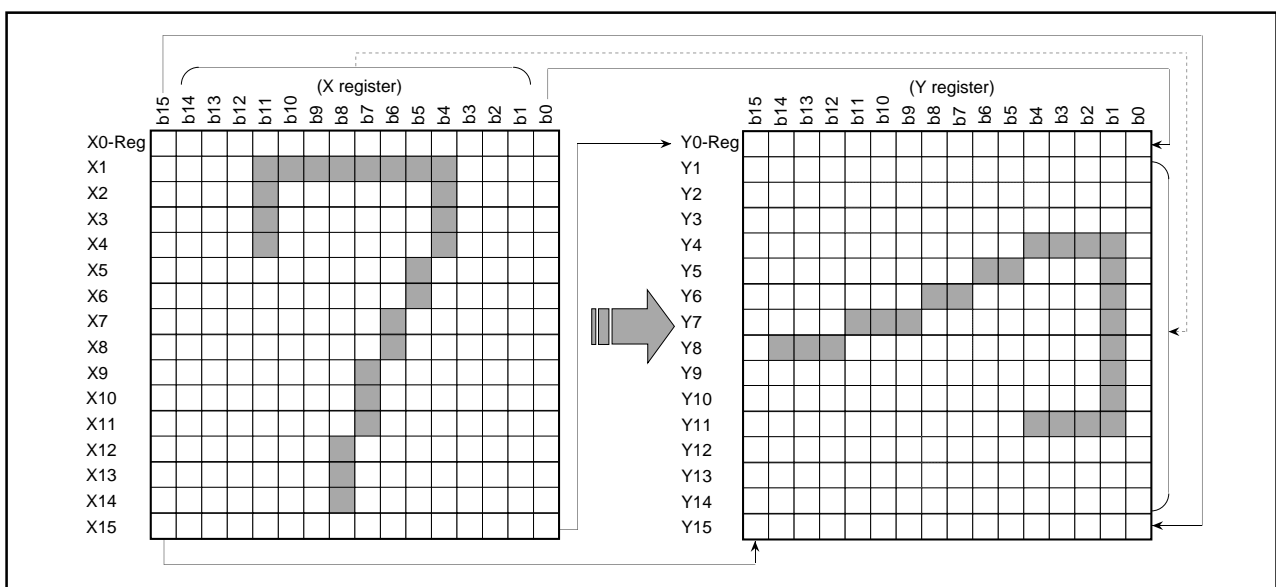


Figure 24.3 XY conversion example

When the read-mode set bit (bit 0 at address 02E016) is “1”, you can read the value written to the Xi register by reading the Yi register. Figure 24.4 shows the conversion table when the read mode set bit = “1”.

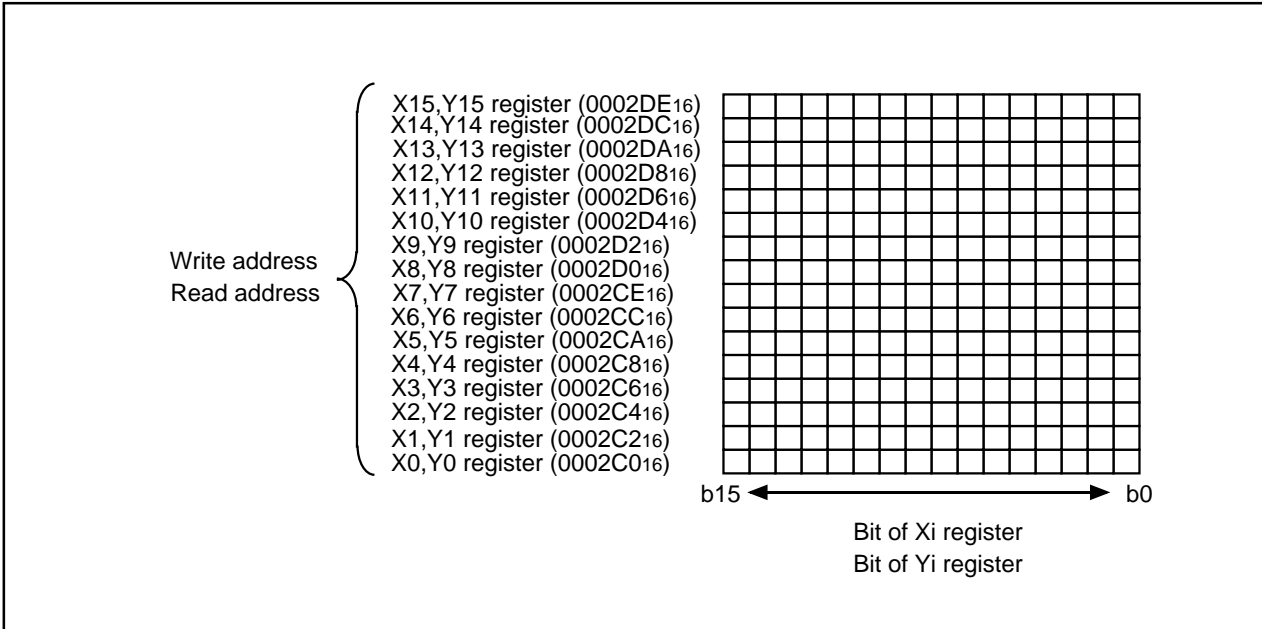


Figure 24.4 Conversion table when the read mode set bit = “1”

The value written to the Xi register is controlled by the write mode set bit (bit 1 at address 02E016). When the write mode set bit (bit 1 at address 02E016) is “0” and data is written to the Xi register, the bit stream is written directly. When the write mode set bit (bit 1 at address 02E016) is “1” and data is written to the Xi register, the bit sequence is reversed so that the high becomes low and vice versa. Figure 24.5 shows the conversion table when the write mode set bit = “1”.

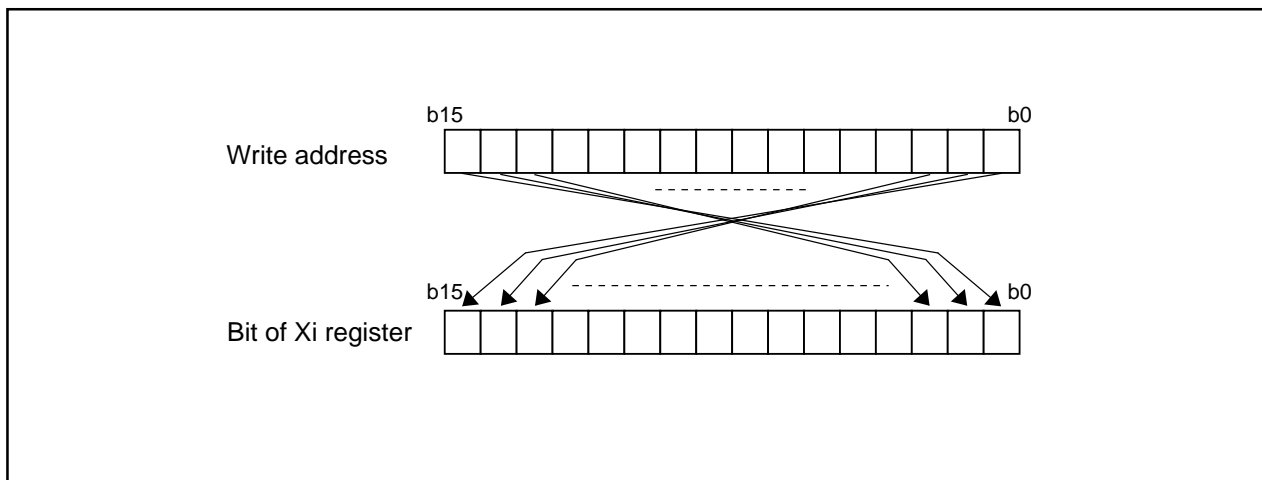


Figure 24.5 Conversion table when the write mode set bit = “1”

## 25. DRAM Controller

There is a built in DRAM controller to which it is possible to connect between 512 Kbytes and 8 Mbytes of DRAM. Table 25.1 shows the functions of the DRAM controller.

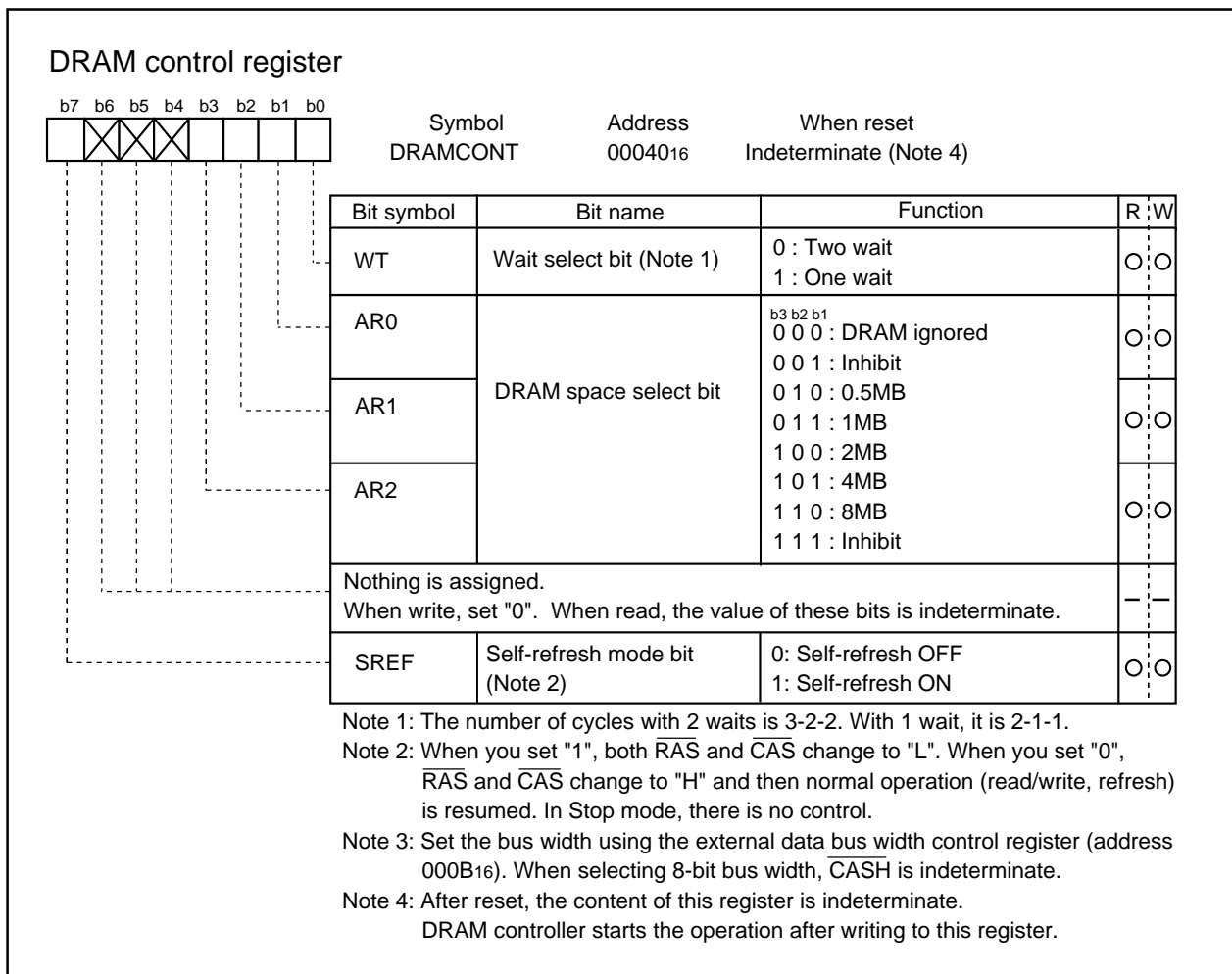
**Table 25.1 DRAM Controller Functions**

DRAM space	512KB, 1MB, 2MB, 4MB, 8MB
Bus control	2CAS/1W
Refresh	CAS before $\overline{\text{RAS}}$ refresh Self refresh-compatible
Function modes	EDO-compatible, fast page mode-compatible
Waits	1 wait or 2 waits, programmable

To use the DRAM controller, use the DRAM space select bit of the DRAM control register (address 0040<sub>16</sub>) to specify the DRAM size. Figure 25.1 shows the DRAM control register.

The DRAM controller cannot be used in external memory mode 3 (bits 1 and 2 at address 0005<sub>16</sub> are "112"). Always use the DRAM controller in external memory modes 0, 1, or 2.

When the data bus width is 16-bit in DRAM area, set "1" to R/W mode select bit (bit 2 at address 0004<sub>16</sub>). Set wait time between after DRAM power ON and before memory processing, and processing necessary for dummy cycle to refresh DRAM by software.



**Figure 25.1 DRAM control register**

• **DRAM Controller Multiplex Address Output**

The DRAM controller outputs the row addresses and column addresses as a multiplexed signal to the address bus A8 to A20. Figure 25.2 shows the output format for multiplexed addresses.

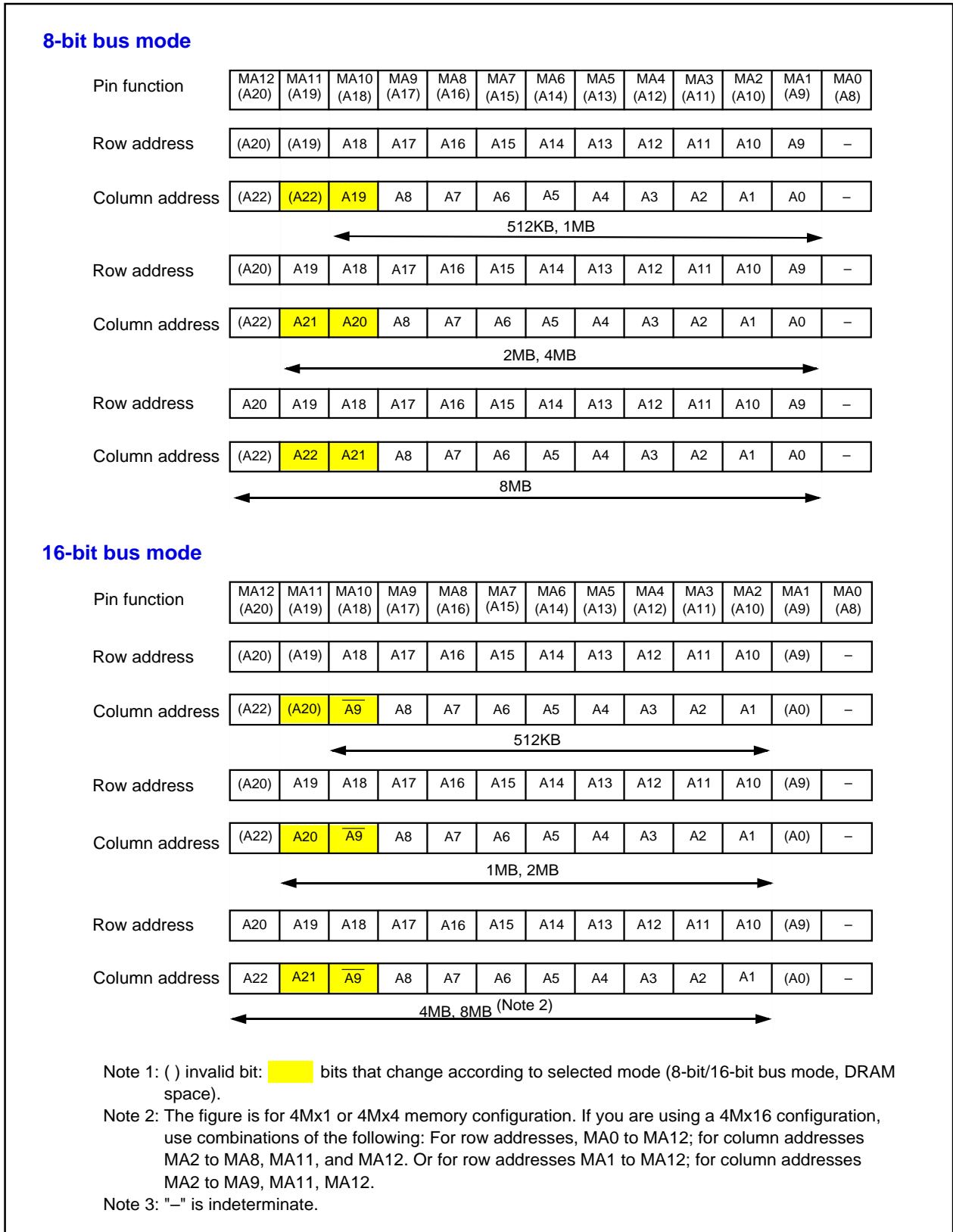


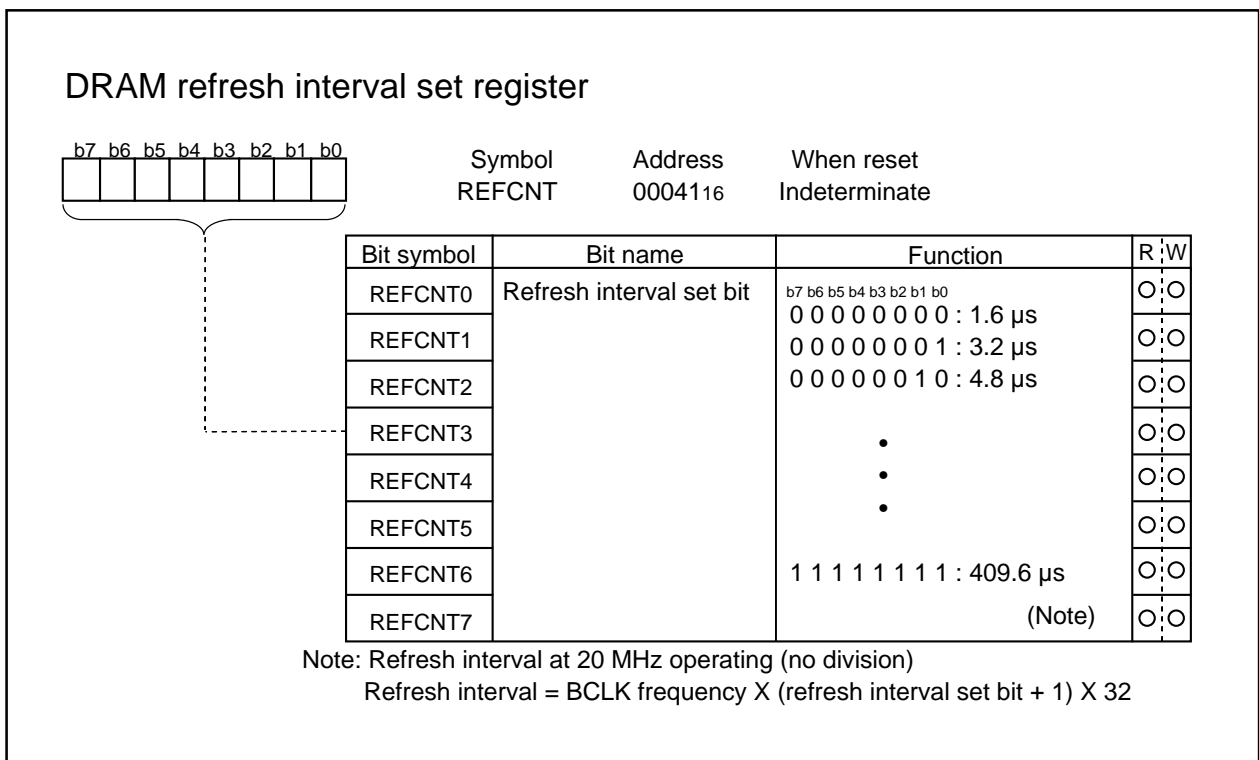
Figure 25.2 Output format for multiplexed addresses

• Refresh

The refresh method is  $\overline{\text{CAS}}$  before  $\overline{\text{RAS}}$ . The refresh interval is set by the DRAM refresh interval set register (address 0041<sub>16</sub>). The refresh signal is not output in HOLD state. Figure 25.3 shows the DRAM refresh interval set register.

Use the following formula to determine the value to set in the refresh interval set register.

$$\text{Refresh interval set register value (0 to 255)} = \text{refresh interval time} / (\text{BCLK frequency} \times 32) - 1$$



**Figure 25.3 DRAM refresh interval set register**

The DRAM self-refresh operates in STOP mode, etc.

When shifting to self-refresh, select DRAM ignored by the DRAM space select bit. In the next instruction, simultaneously set the DRAM space select bit and self-refresh ON by self-refresh mode bit. Also, insert two NOPs after the instruction that sets the self-refresh mode bit to "1".

Do not access external memory while operating in self-refresh. (All external memory space access is inhibited. )

When disabling self-refresh, simultaneously select DRAM ignored by the DRAM space select bit and self-refresh OFF by self-refresh mode bit. In the next instruction, set the DRAM space select bit.

Do not access the DRAM space immediately after setting the DRAM space select bit.

Example) One wait is selected by the wait select bit and 4MB is selected by the DRAM space select bit

Shifting to self-refresh

```

...
mov.b #00000001b,DRAMCONT ;DRAM ignored, one wait is selected
mov.b #10001011b,DRAMCONT ;Set self-refresh, select 4MB and one wait
nop ;Two nops are needed
nop ;
...

```

Disable self-refresh

```

...
mov.b #00000001b,DRAMCONT ;Disable self-refresh, DRAM ignored, one wait is
                          ;selected
mov.b #00001011b,DRAMCONT ;Select 4MB and one wait
nop ;Inhibit instruction to access DRAM area
nop
...

```

Figures 25.4 to 25.6 show the bus timing during DRAM access.

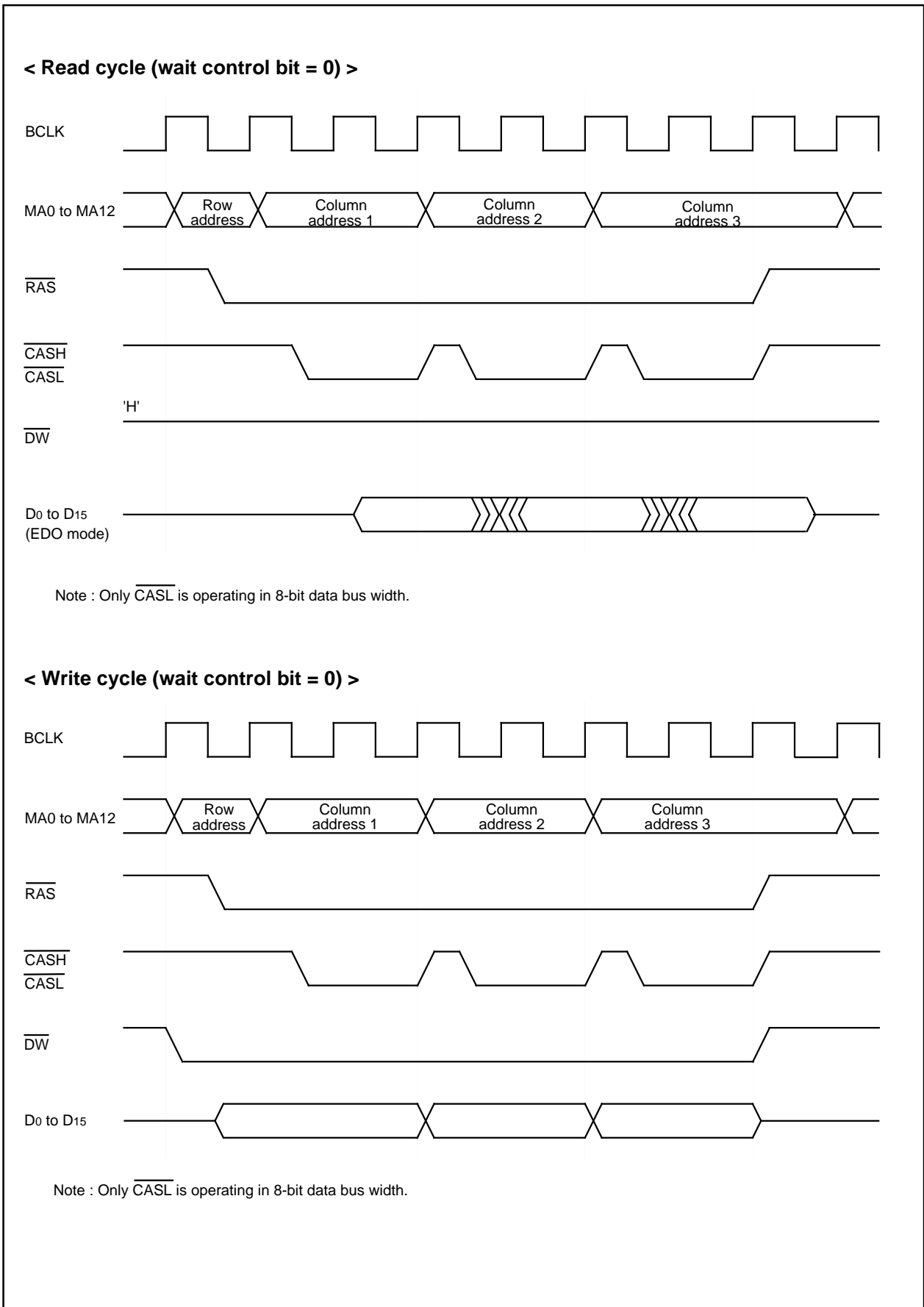


Figure 25.4 The bus timing during DRAM access (1)



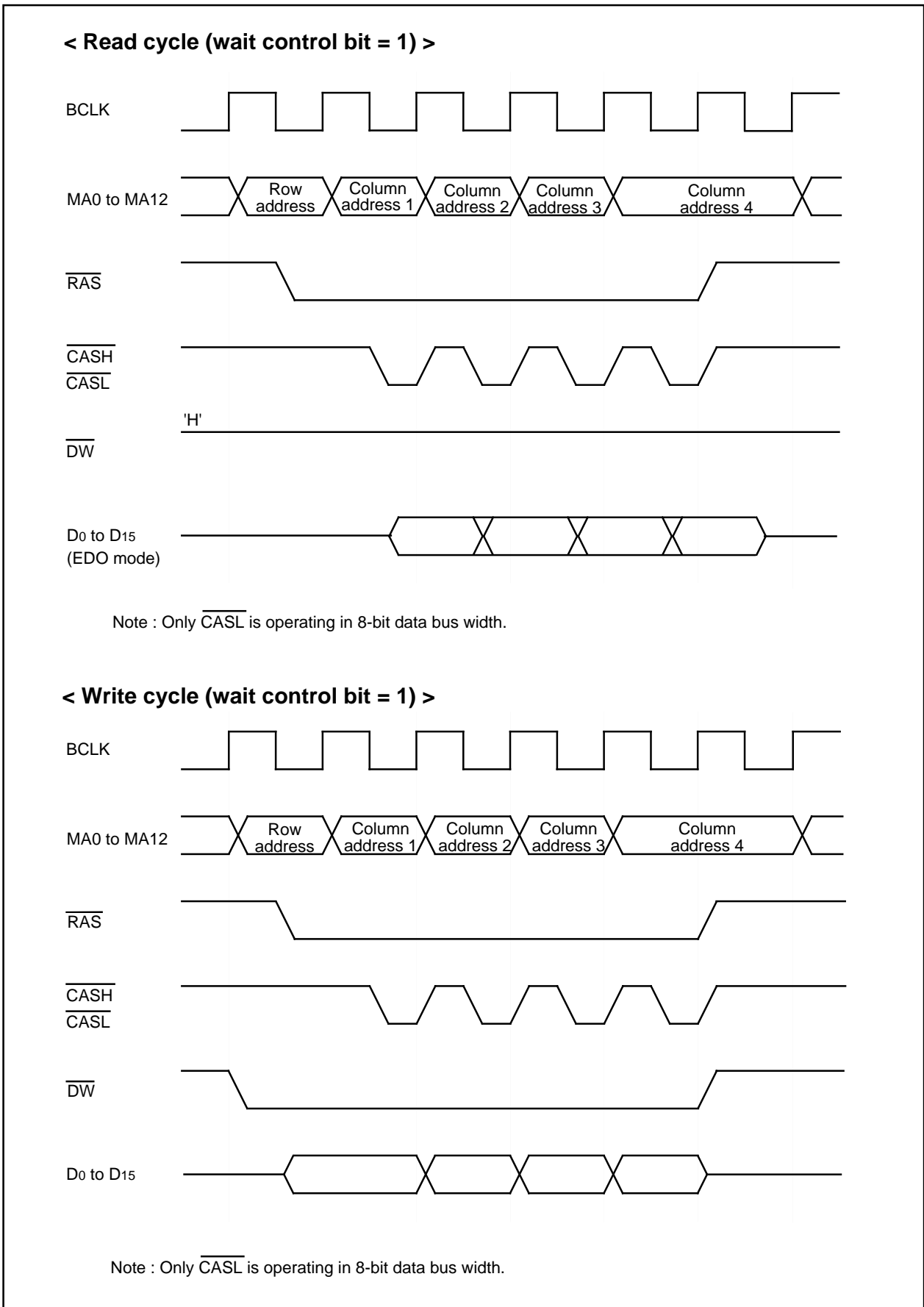


Figure 25.5 The bus timing during DRAM access (2)

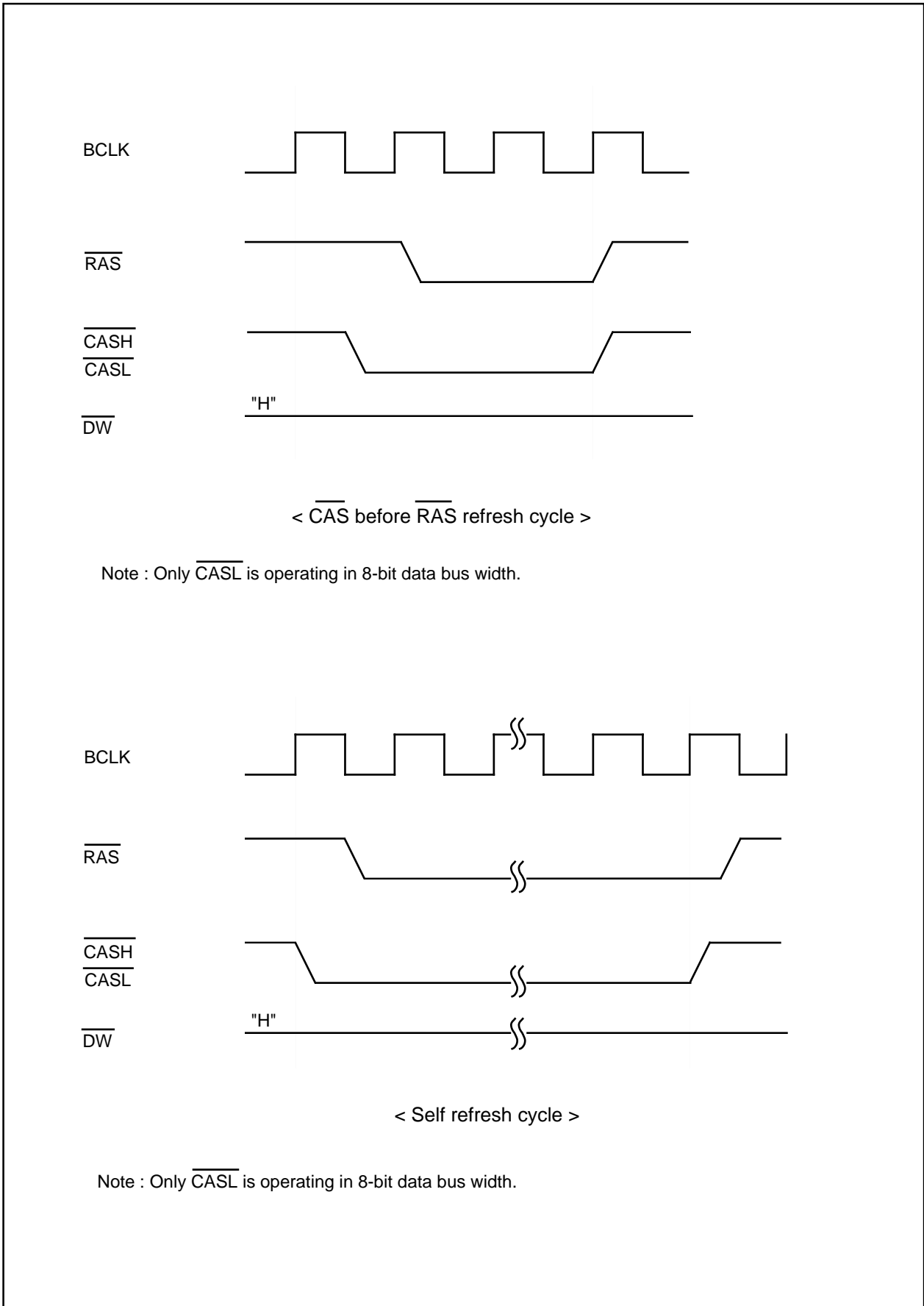


Figure 25.6 The bus timing during DRAM access (3)

## 26. Programmable I/O Ports

There are 87 programmable I/O ports for 100-pin version: P0 to P10 (excluding P85). There are 123 programmable I/O ports for 144-pin version: P0 to P15 (excluding P85). Each port can be set independently for input or output using the direction register. A pull-up resistance for each block of 4 ports can be set. P85 is an input-only port and has no built-in pull-up resistance.

Figures 26.1 to 26.3 show the programmable I/O ports.

Each pin functions as a programmable I/O port and as the I/O for the built-in peripheral devices.

To use the pins as the inputs for the built-in peripheral devices, set the direction register of each pin to input mode. When the pins are used as the outputs for the built-in peripheral devices (other than the D/A converter), set the corresponding function select registers A, B and C. When pins are to be used as the outputs for the D/A converter, set the function select register of each pin to I/O port, and set the direction registers to input mode.

Table 26.1 lists each port and peripheral function.

See the descriptions of the respective functions for how to set up the built-in peripheral devices.

### (1) Direction registers

Figures 26.4 and 26.5 show the direction registers.

These registers are used to choose the direction of the programmable I/O ports. Each bit in these registers corresponds one for one to each I/O pin.

In memory expansion and microprocessor mode, the contents of corresponding direction register of pins A0 to A22,  $\overline{A23}$ , D0 to D15, MA0 to MA12,  $\overline{CS0}$  to  $\overline{CS3}$ ,  $\overline{WRL/WR/CASL}$ ,  $\overline{WRH/BHE/CASH}$ ,  $\overline{RD/DW}$ ,  $\overline{BCLK/ALE/CLKOUT}$ ,  $\overline{HLDA/ALE}$ ,  $\overline{HOLD}$ ,  $\overline{ALE/RAS}$ , and  $\overline{RDY}$  are not changed.

Note: There is no direction register bit for P85.

### (2) Port registers

Figures 26.6 and 26.7 show the port registers.

These registers are used to write and read data for input and output to and from an external device. A port register consists of a port latch to hold output data and a circuit to read the status of a pin. Each bit in port registers corresponds one for one to each I/O pin.

In memory expansion and microprocessor mode, the contents of corresponding port register of pins A0 to A22,  $\overline{A23}$ , D0 to D15, MA0 to MA12,  $\overline{CS0}$  to  $\overline{CS3}$ ,  $\overline{WRL/WR/CASL}$ ,  $\overline{WRH/BHE/CASH}$ ,  $\overline{RD/DW}$ ,  $\overline{BCLK/ALE/CLKOUT}$ ,  $\overline{HLDA/ALE}$ ,  $\overline{HOLD}$ ,  $\overline{ALE/RAS}$ , and  $\overline{RDY}$  are not changed.

### (3) Function select register A

Figures 26.8 and 26.9 show the function select registers A.

The register is used to select port output and peripheral function output when the port functions for both port output and peripheral function output.

Each bit of this register corresponds to each pin that functions for both port output and peripheral function output.

#### (4) Function select register B

Figures 26.10 and 26.11 show the function select registers B.

This register selects the 1st peripheral function output and second peripheral function output when multiple peripheral function outputs are assigned to a pin. For pins with a third peripheral function, this register selects whether to enable the function select register C, or output the second peripheral function.

Each bit of this register corresponds to each pin that has multiple peripheral function outputs assigned to it. This register is enabled when the bits of the corresponding function select register A are set for peripheral functions.

The bit 3 to bit 6 of function select register B3 is ignored bit for input peripheral function. When using DA0/DA1 and ANEX0/ANEX1, set related bit to "1". When not using DA0/DA1 or ANEX0/ANEX1, set related bit to "0".

#### (5) Function select register C

Figure 26.12 shows the function select register C.

This register is used to select the first peripheral function output and the third peripheral function output when three peripheral function outputs are assigned to a pin.

This register is effective when the bits of the function select register A of the counterpart pin have selected a peripheral function and when the function select register B has made effective the function select register C.

The bit 7 (PSC\_7) is assigned the key-in interrupt inhibit bit. Setting 1 in the key-in interrupt inhibit bit causes no key-in interrupts regardless of the settings in the interrupt control register even if L is entered in pins KI0 to KI3. With 1 set in the key-in interrupt inhibit bit, input from a port pin cannot be effected even if the port direction register is set to input mode.

#### (6) Pull-up control registers

Figures 26.13 and 26.14 show the pull-up control registers.

The pull-up control register can be set to apply a pull-up resistance to each block of 4 ports. When ports are set to have a pull-up resistance, the pull-up resistance is connected only when the direction register is set for input.

Since P0 to P5 operate as the bus in memory expansion mode and microprocessor mode, do not set the pull-up control register. However, it is possible to select pull-up resistance presence to the usable port as I/O port by setting.

#### (7) Port control register

Figure 26.15 shows the port control register.

This register is used to choose whether to make port P1 a CMOS port or an Nch open drain. In the Nch open drain, the port P1 has no function that a complete open drain but keeps the CMOS port's Pch always turned off. Thus the absolute maximum rating of the input voltage falls within the range from  $-0.3\text{ V}$  to  $V_{cc} + 0.3\text{ V}$ .

The port control register functions similarly to the above also in the case in which port P1 can be used as a port when the bus width in the full external areas comprises 8 bits in either microprocessor mode or in memory expansion mode.

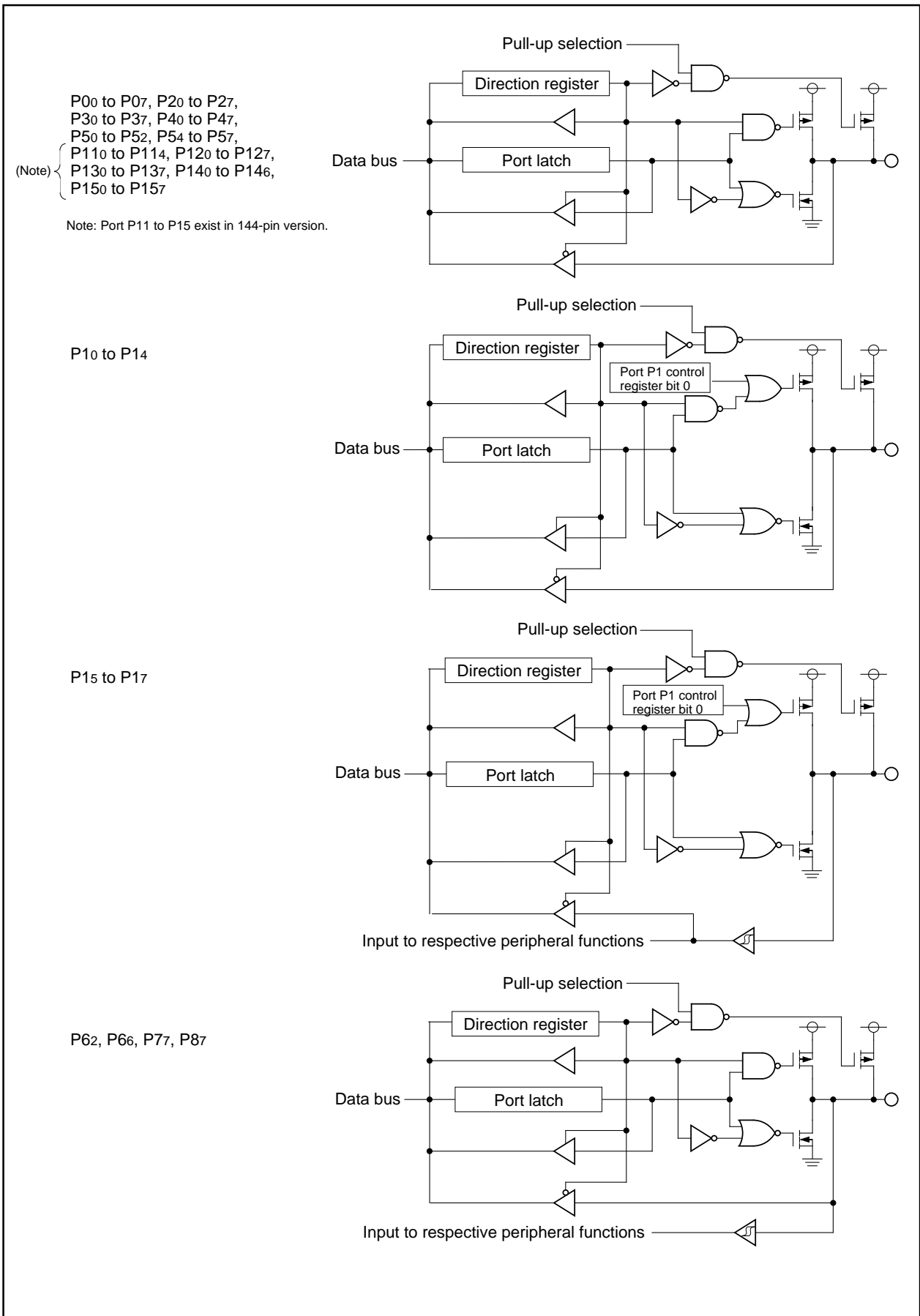


Figure 26.1 Programmable I/O ports (1)

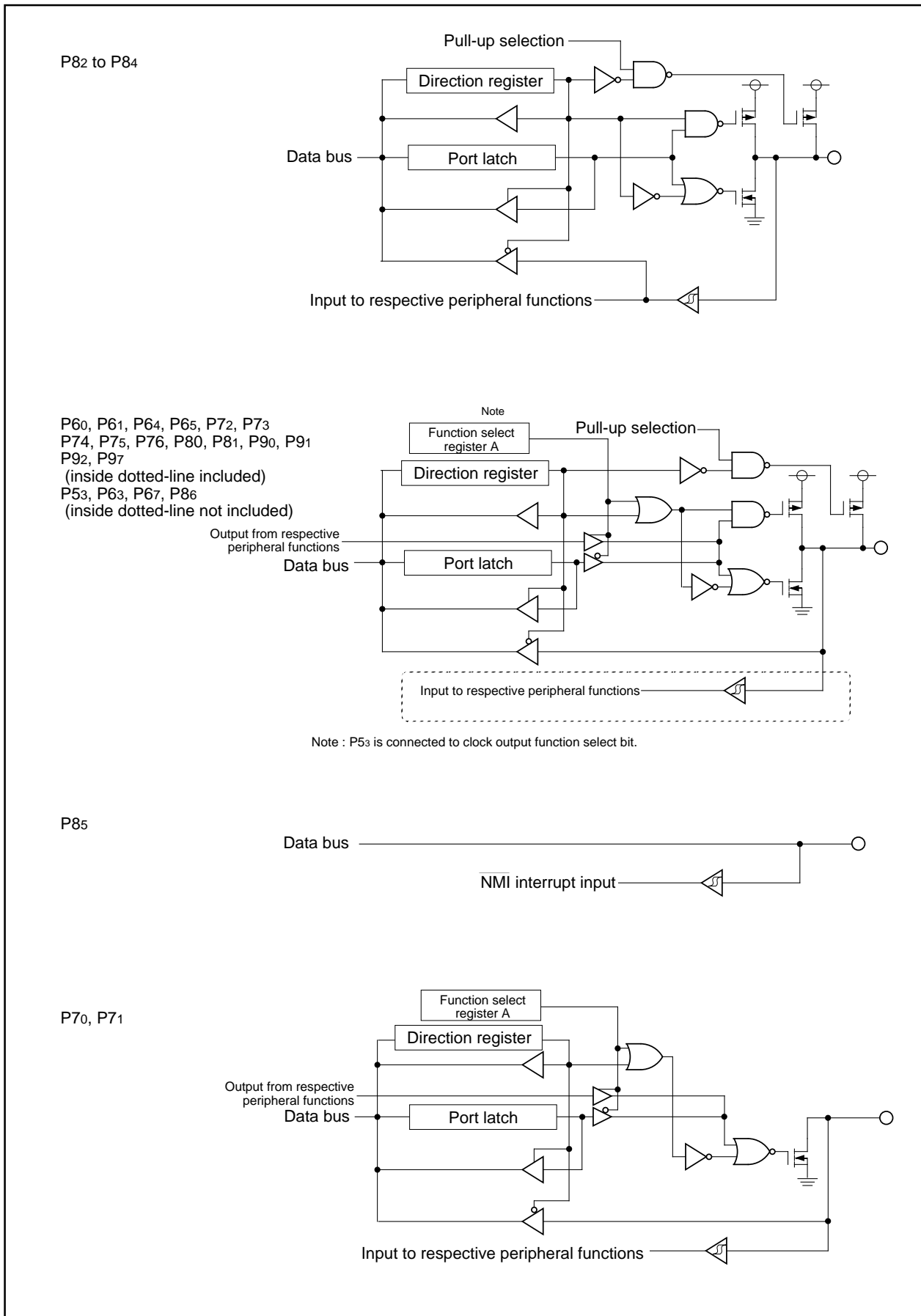


Figure 26.2 Programmable I/O ports (2)

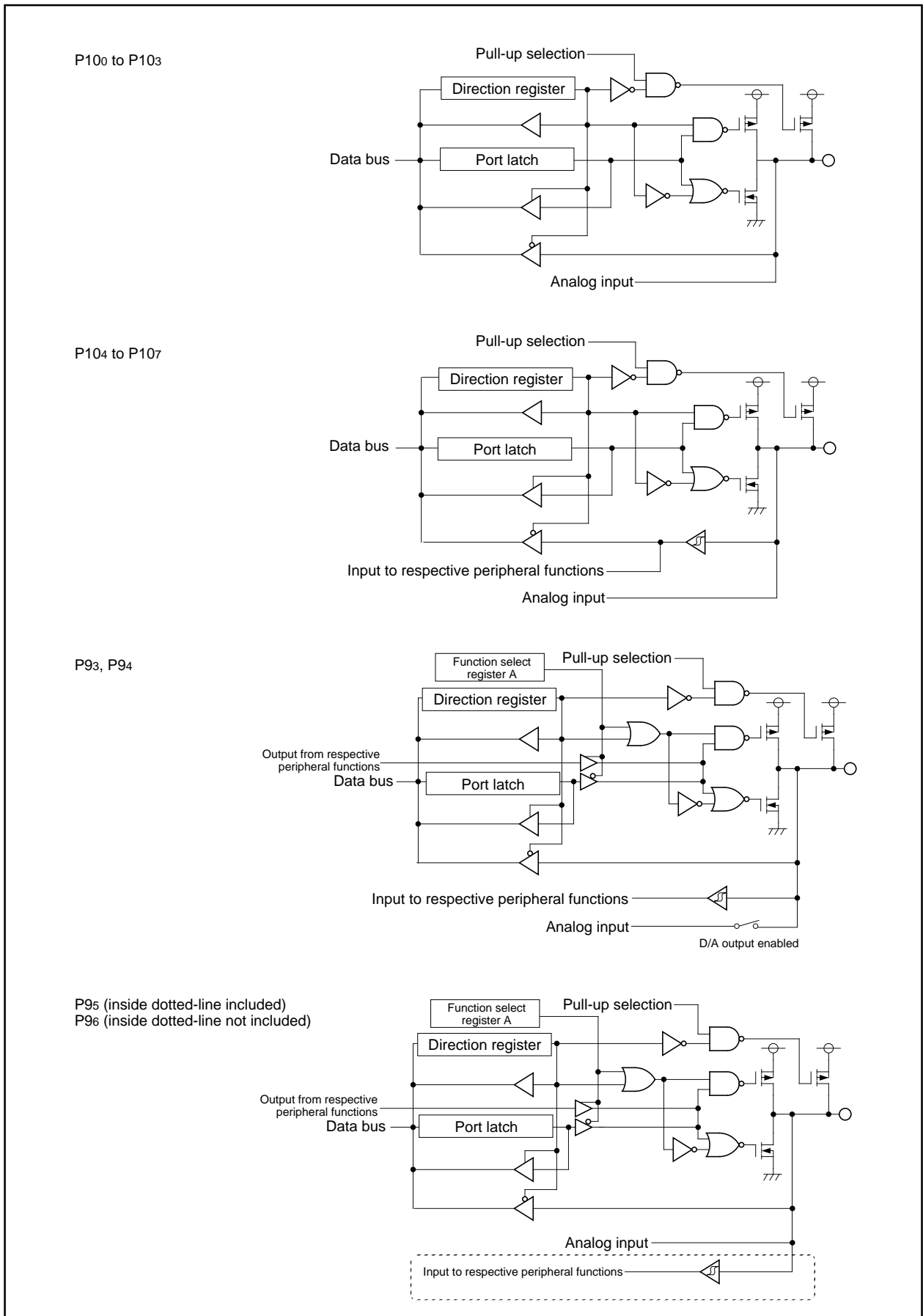


Figure 26.3 Programmable I/O ports (3)

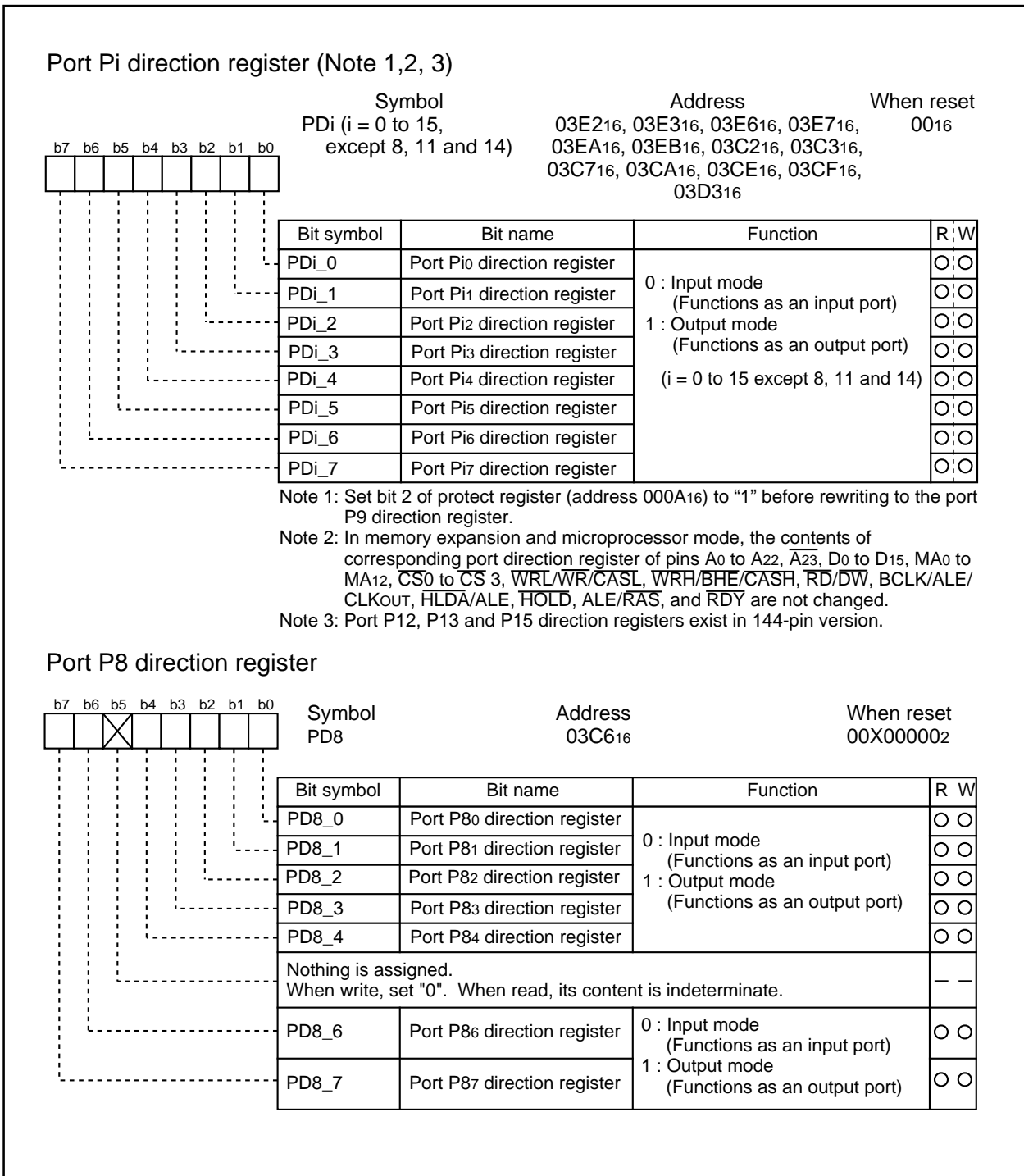


Figure 26.4 Direction register (1)



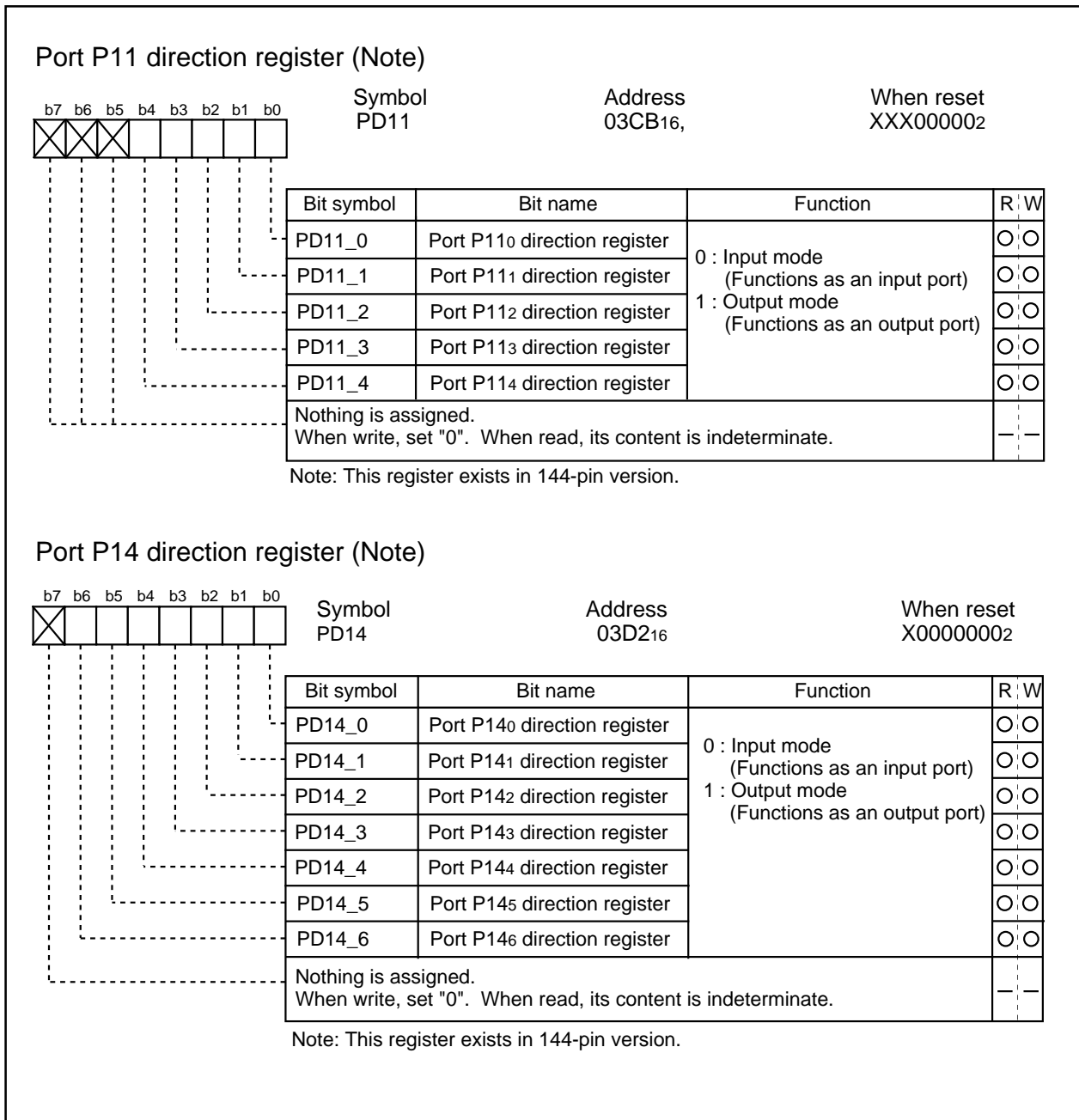


Figure 26.5 Direction register (2)

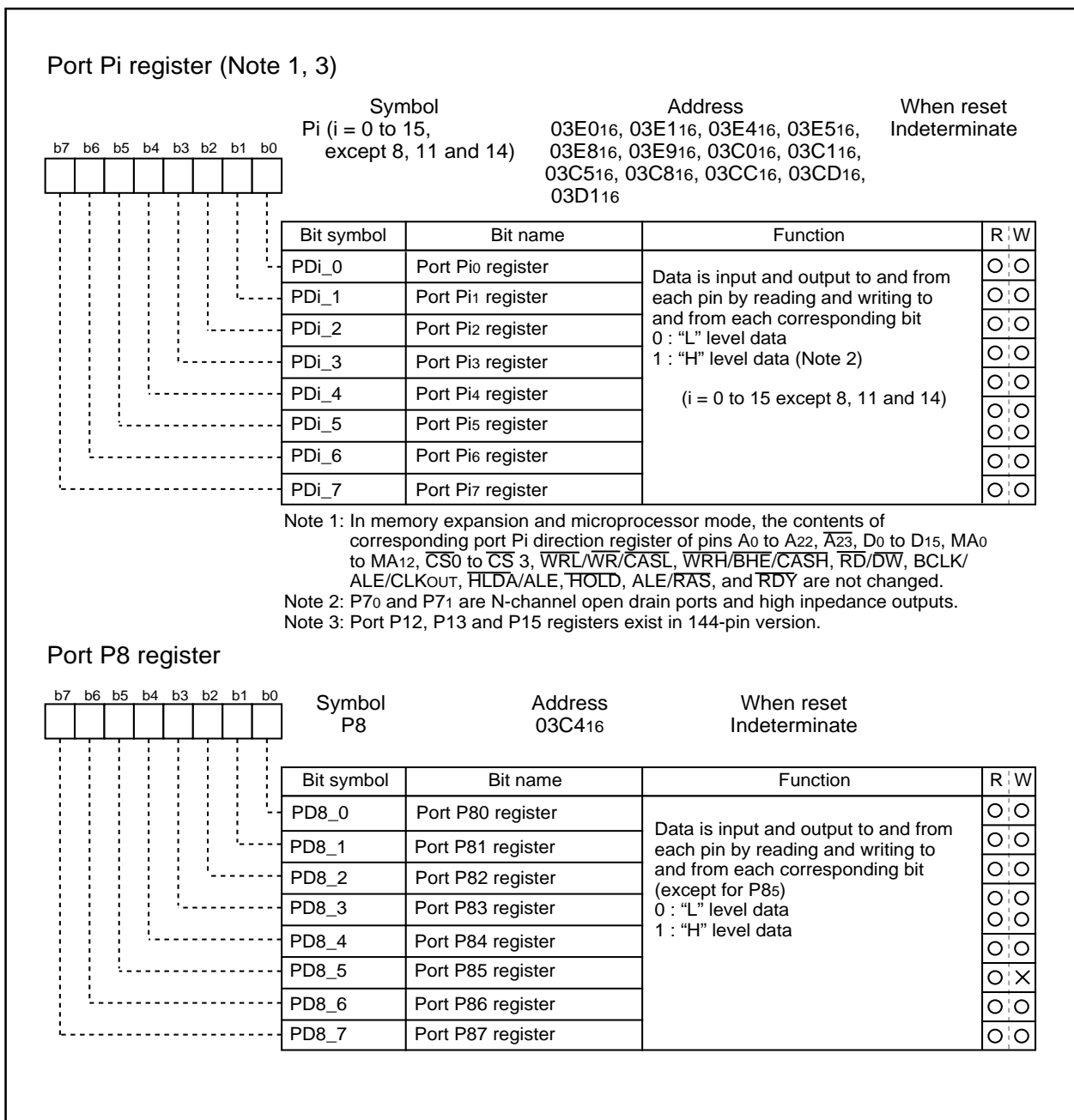


Figure 26.6 Port register (1)

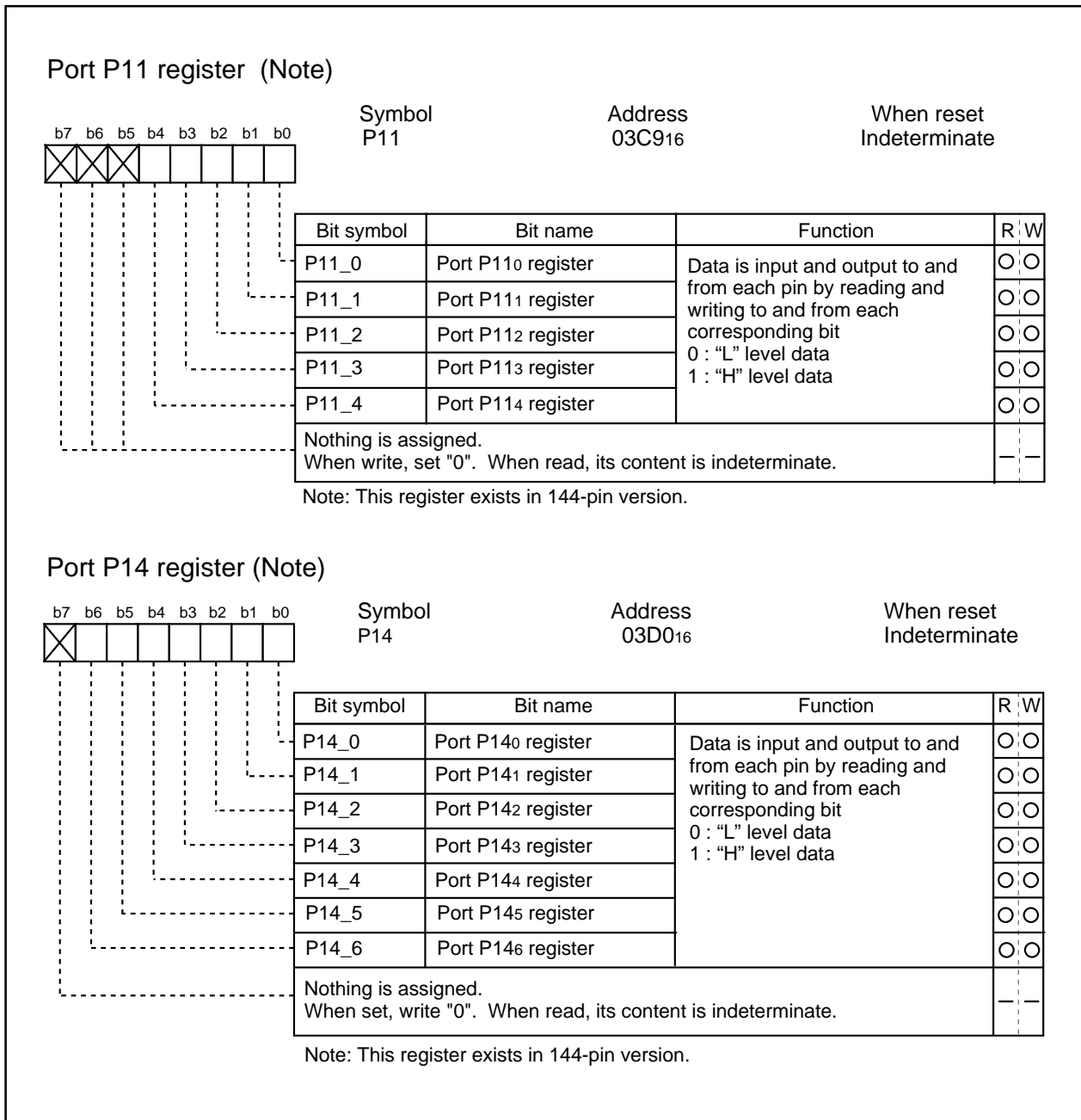


Figure 26.7 Port register (2)

**Table 26.1 Each port and peripheral output function (Note 1)**

Port	Peripheral output function 1	Peripheral output function 2	Peripheral output function 3
P60	$\overline{\text{RTS}}_0$ output	—	—
P61	CLK <sub>0</sub> output	—	—
P62	—	—	—
P63	TxD <sub>0</sub> output	—	—
P64	$\overline{\text{RTS}}_1$ output	CLKS <sub>1</sub> output	—
P65	CLK <sub>1</sub> output	—	—
P66	—	—	—
P67	TxD <sub>1</sub> output	—	—
P7 <sub>0</sub> (Note 2)	TxD <sub>2</sub> (SDA <sub>2</sub> ) output	TA0OUT output	—
P7 <sub>1</sub> (Note 2)	SCL <sub>2</sub> output	—	—
P7 <sub>2</sub>	CLK <sub>2</sub> output	TA1OUT output	V phase output
P7 <sub>3</sub>	$\overline{\text{RTS}}_2$ output	$\overline{\text{V}}$ phase output	—
P7 <sub>4</sub>	TA2OUT output	W phase output	—
P7 <sub>5</sub>	$\overline{\text{W}}$ phase output	—	—
P7 <sub>6</sub>	TA3OUT output	—	—
P7 <sub>7</sub>	—	—	—
P8 <sub>0</sub>	TA4OUT output	U phase output	—
P8 <sub>1</sub>	$\overline{\text{U}}$ phase output	—	—
P8 <sub>2</sub>	—	—	—
P8 <sub>3</sub>	—	—	—
P8 <sub>4</sub>	—	—	—
P8 <sub>5</sub>	—	—	—
P8 <sub>6</sub>	—	—	—
P8 <sub>7</sub>	—	—	—
P9 <sub>0</sub>	CLK <sub>3</sub> output	—	—
P9 <sub>1</sub>	SCL <sub>3</sub> output	STxD <sub>3</sub> output	—
P9 <sub>2</sub>	TxD <sub>3</sub> (SDA <sub>3</sub> ) output	—	—
P9 <sub>3</sub>	$\overline{\text{RTS}}_3$ output	—	—
P9 <sub>4</sub>	$\overline{\text{RTS}}_4$ output	—	—
P9 <sub>5</sub>	CLK <sub>4</sub> output	—	—
P9 <sub>6</sub>	TxD <sub>4</sub> (SDA <sub>4</sub> ) output	—	—
P9 <sub>7</sub>	SCL <sub>3</sub> output	STxD <sub>4</sub> output	—

Note 1: When using peripheral input function, set the corresponding function select register A to "0" (I/O port).

Note 2: N-channel open drain output.

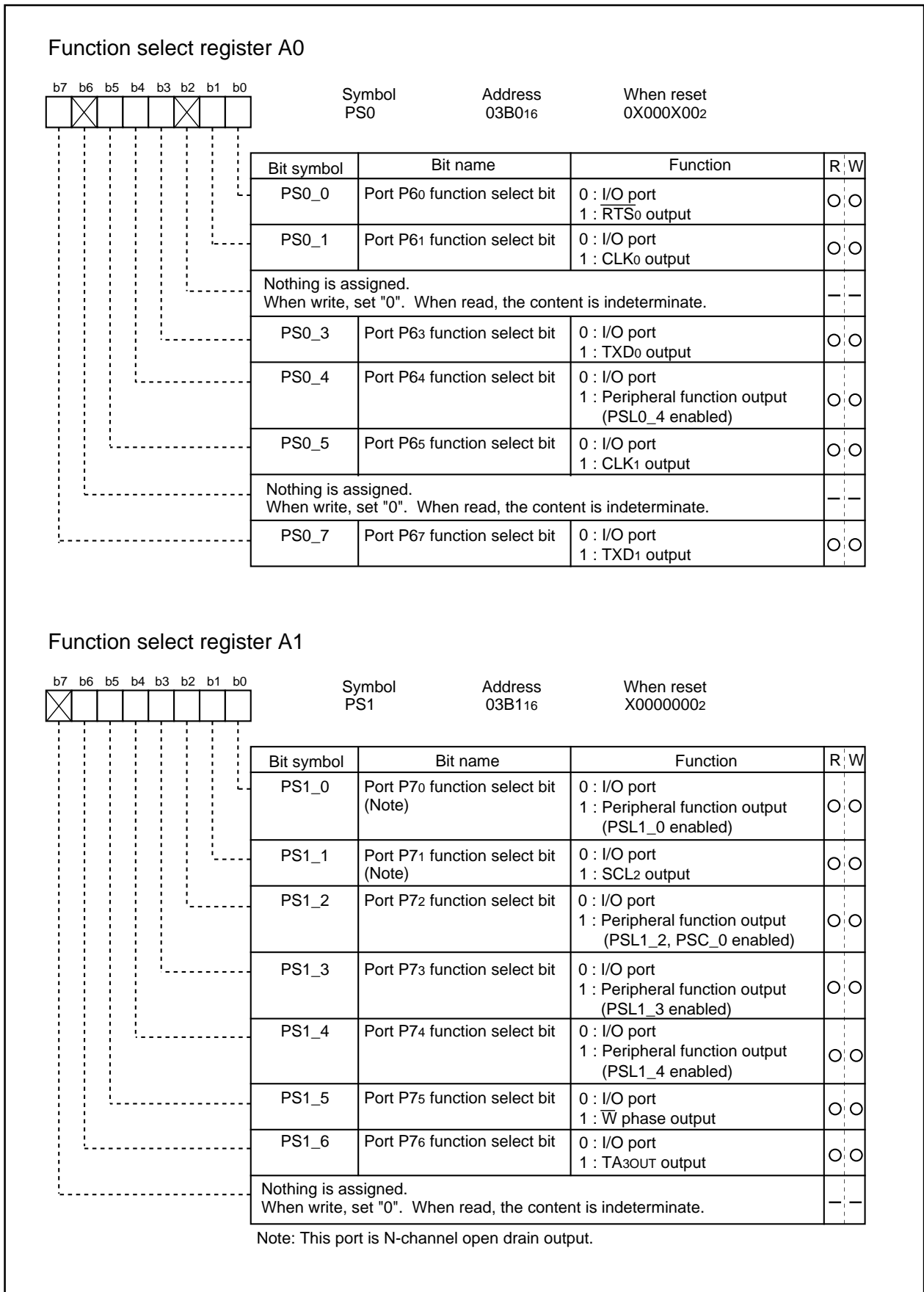


Figure 26.8 Function select register A (1)

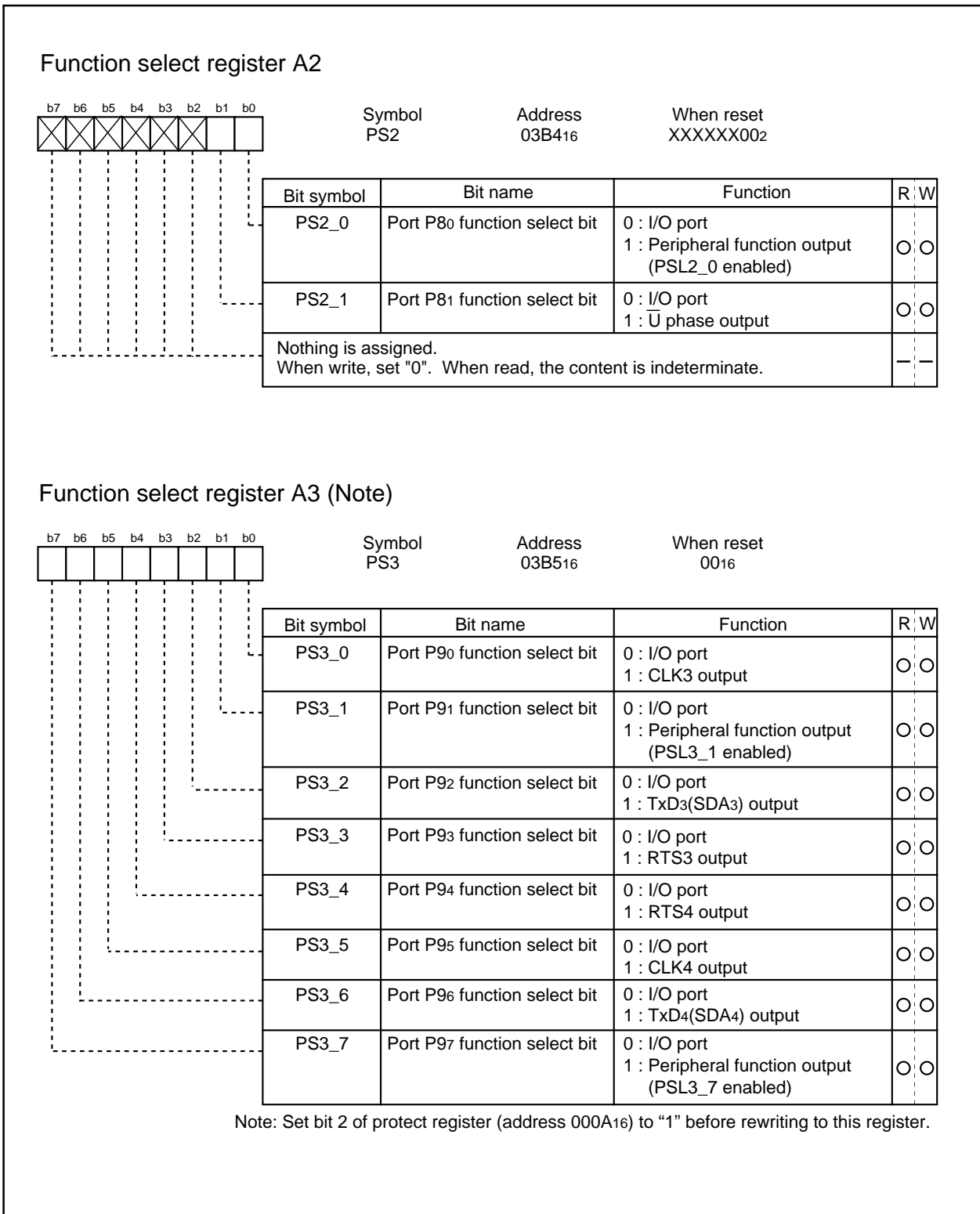


Figure 26.9 Function select register A (2)

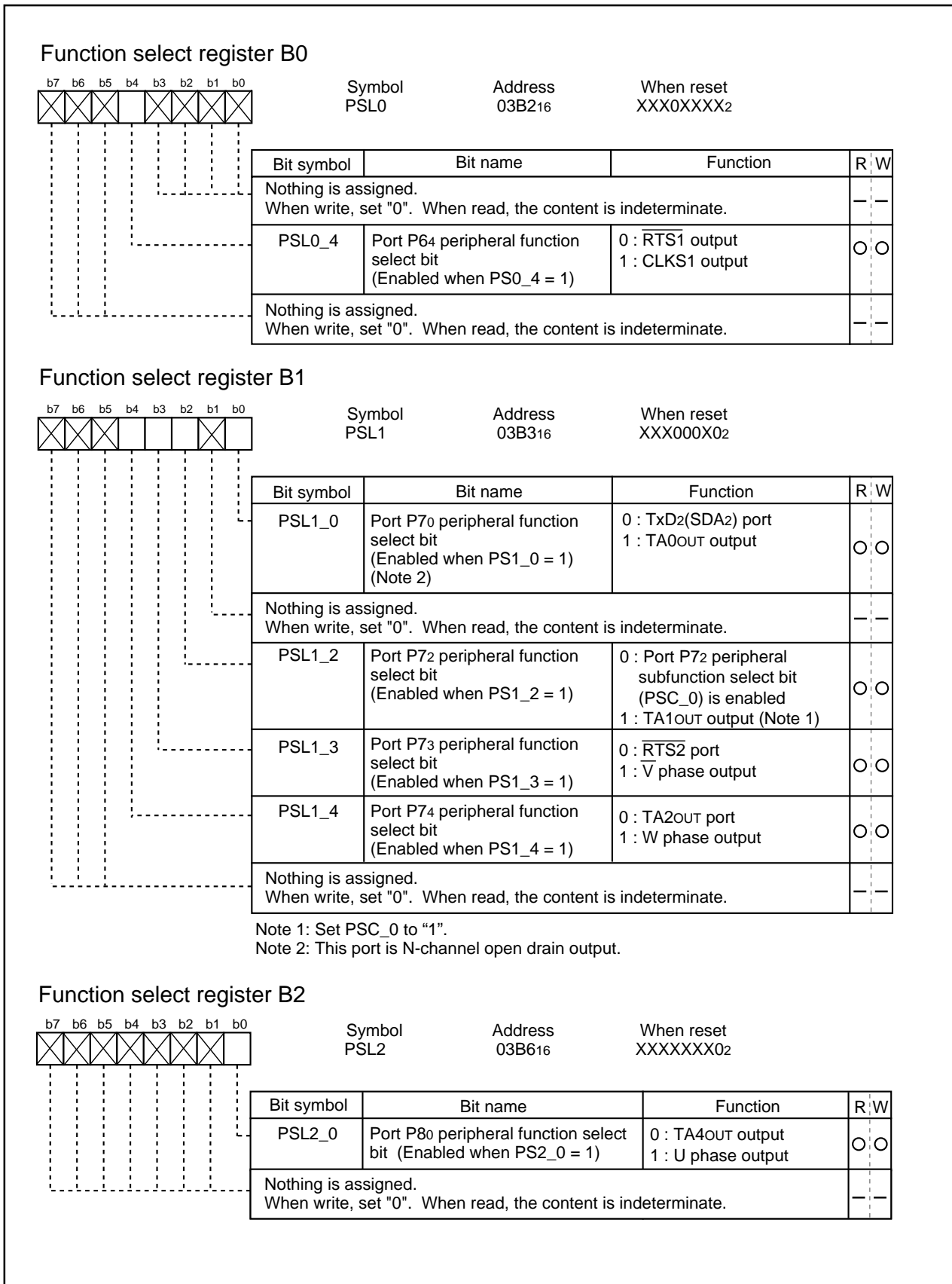


Figure 26.10 Function select register B (1)

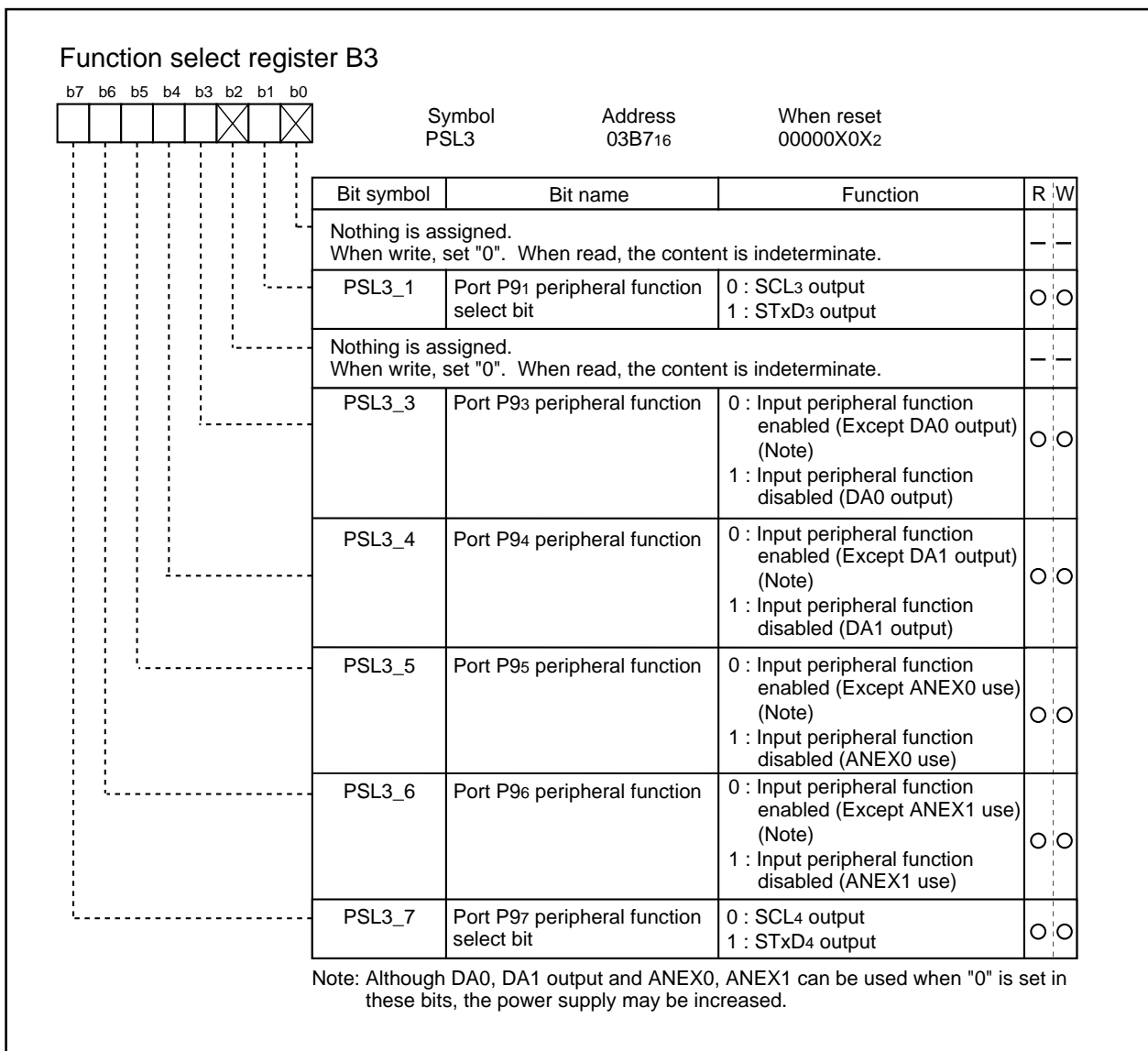


Figure 26.11 Function select register B (2)

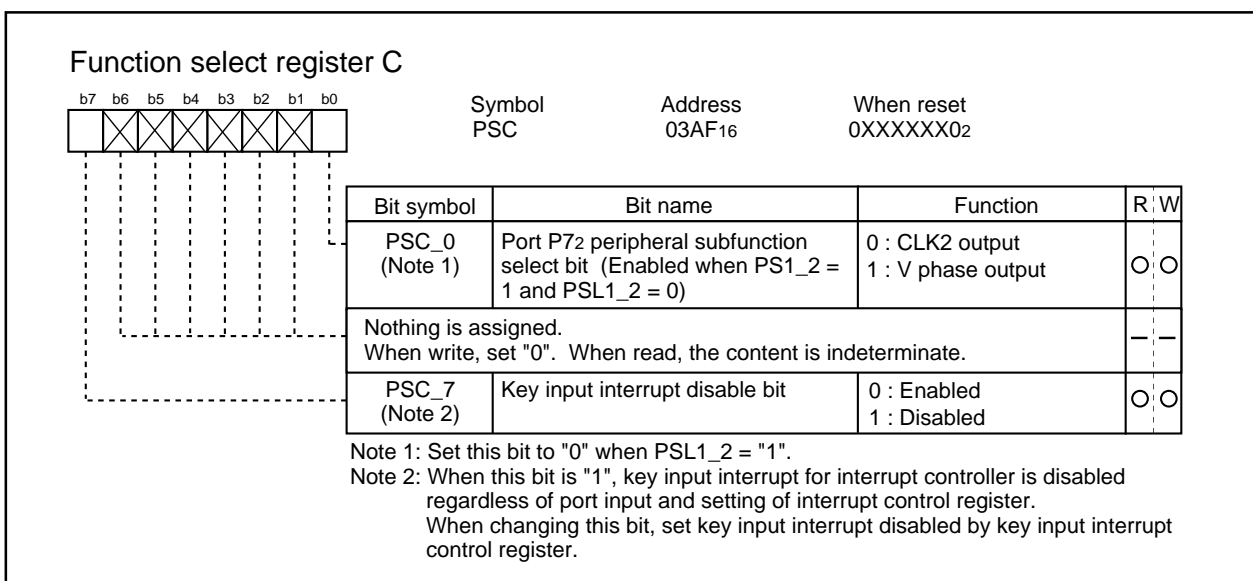


Figure 26.12 Function select register C



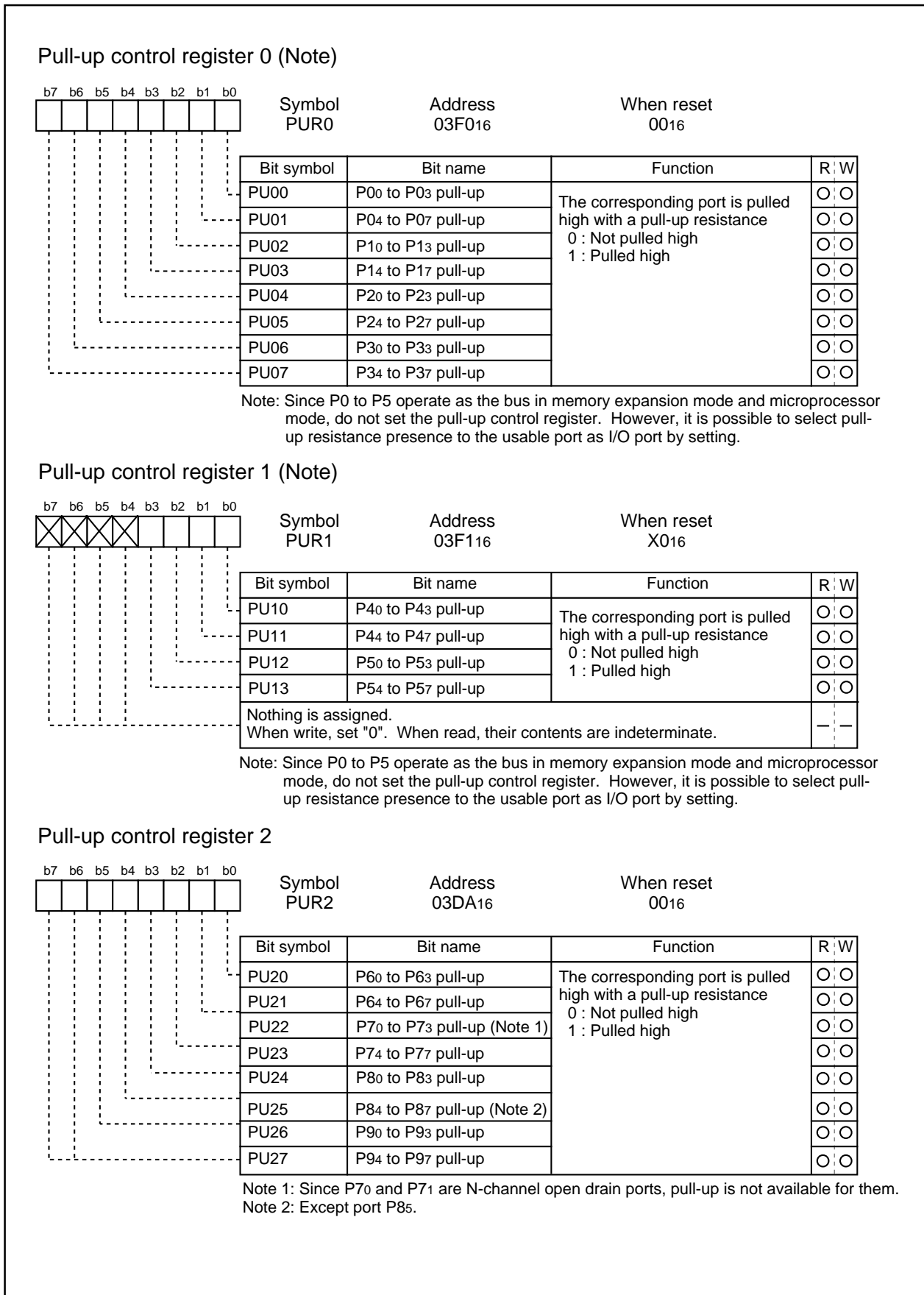


Figure 26.13 Pull-up control register (1)

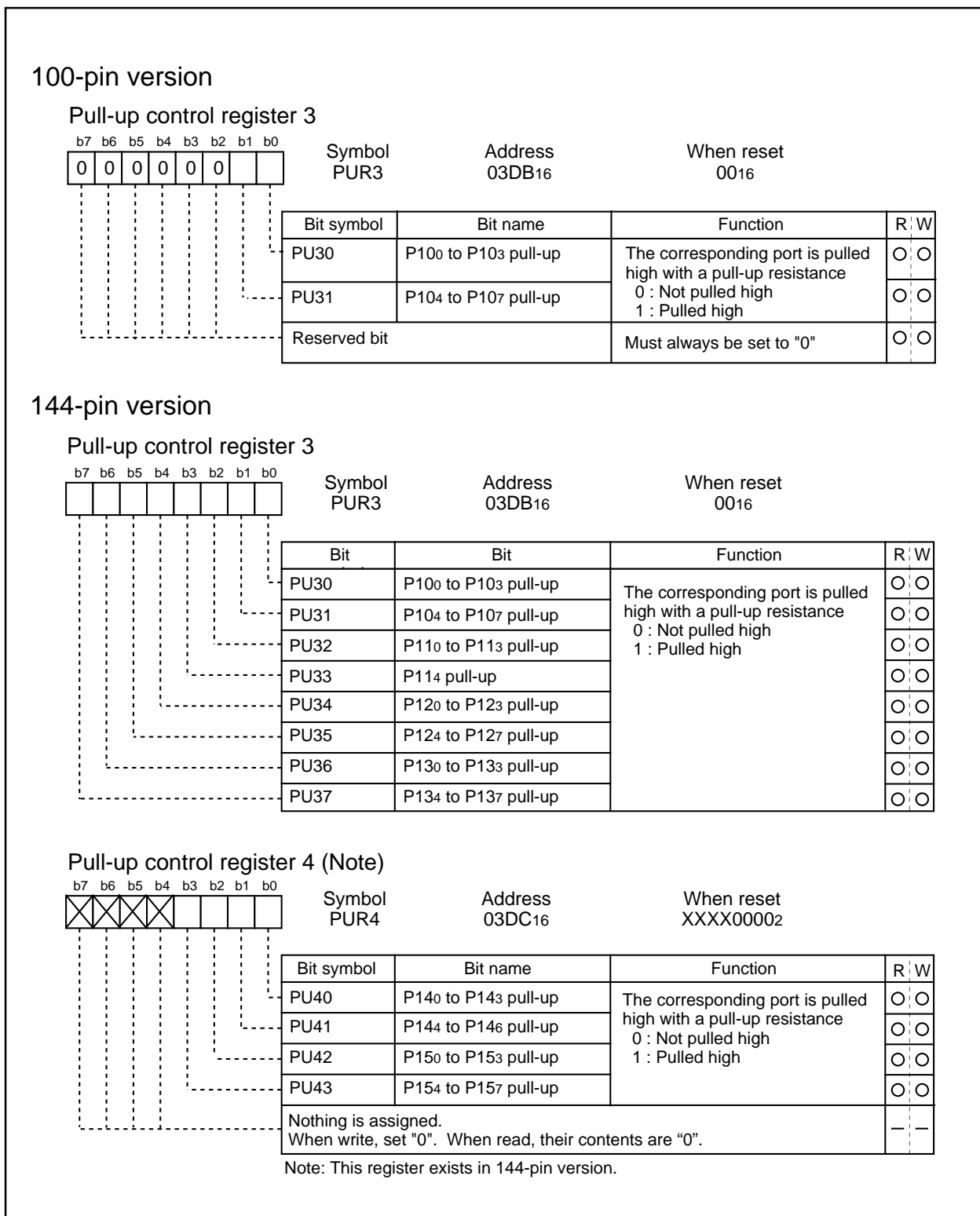
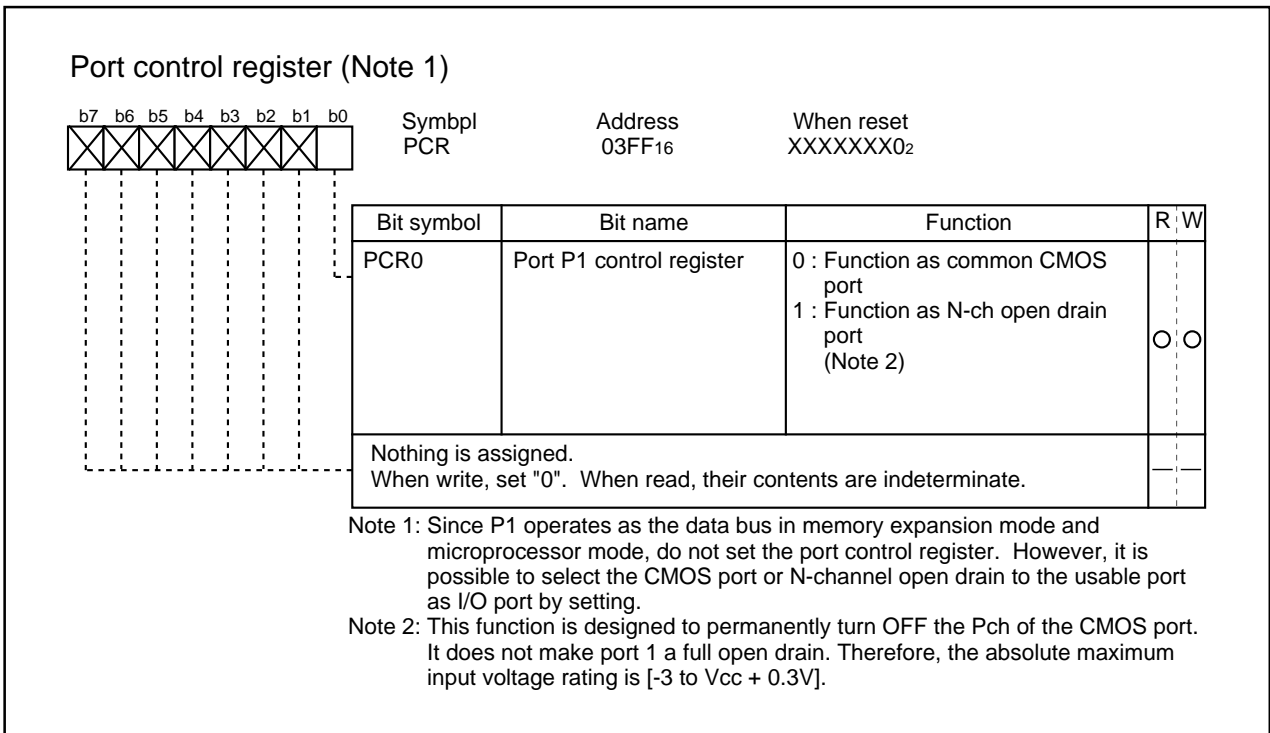


Figure 26.14 Pull-up control register (2)



**Figure 26.15 Port control register**

**Table 26.2 Example connection of unused pins in single-chip mode**

Pin name	Connection
Ports P0 to P15 (excluding P85) Note 1)	(After setting for input mode, connect every pin to Vss via a resistance (pull-down); or after setting for output mode, leave these pins open.
XOUT (Note 2)	Open
NMI	Connect via resistance to Vcc (pull-up)
AVcc	Connect to Vcc
AVss, VREF, BYTE	Connect to Vss

Note 1: Port P11 to P15 exist in 144-pin version.

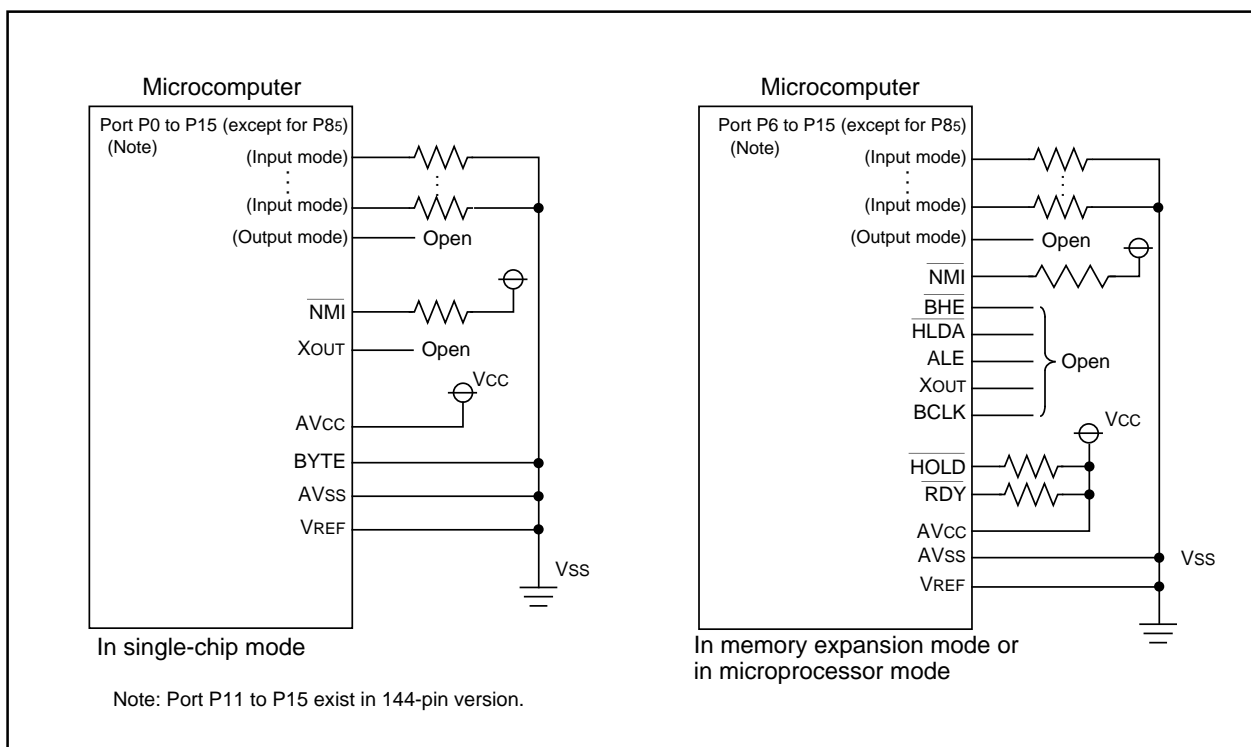
Note 2: With external clock input to XIN pin.

**Table 26.3 Example connection of unused pins in memory expansion mode and microprocessor mode**

Pin name	Connection
Ports P6 to P15(excluding P85) Note 1)	(After setting for input mode, connect every pin to Vss via a resistance (pull-down); or after setting for output mode, leave these pins open.
BHE, ALE, HLDA, XOUT(Note 2), BCLK	Open
HOLD, RDY, NMI	Connect via resistance to Vcc (pull-up)
AVcc	Connect to Vcc
AVss, VREF	Connect to Vss

Note 1: Port P11 to P15 exist in 144-pin version.

Note 2: With external clock input to XIN pin.



**Figure 26.16 Example connection of unused pins**

## 27. Usage Precaution

### SFR (100-pin version)

- (1) Addresses 03C9<sub>16</sub>, 03CB<sub>16</sub> to 03D3<sub>16</sub>, 03DC<sub>16</sub> area is for future plan. Must set "FF<sub>16</sub>" to address 03CB<sub>16</sub>, 03CE<sub>16</sub>, 03CF<sub>16</sub>, 03D2<sub>16</sub>, 03D3<sub>16</sub> and "00<sub>16</sub>" to address 03DC<sub>16</sub> at initial setting.

### Timer

- (1) A timer Ai register and a timer Bi register are unstable after MCU resetting. Please start a count after setting a value as the timer Ai register or timer Bi register to be used, when using a timer.

### Timer A (timer mode)

- (1) Reading the timer Ai register while a count is in progress allows reading, with arbitrary timing, the value of the counter. Reading the timer Ai register with the reload timing gets "FFFF<sub>16</sub>". Reading the timer Ai register after setting a value in the timer Ai register with a count halted but before the counter starts counting gets a proper value.

### Timer A (event counter mode)

- (1) Reading the timer Ai register while a count is in progress allows reading, with arbitrary timing, the value of the counter. Reading the timer Ai register with the reload timing gets "FFFF<sub>16</sub>" by underflow or "0000<sub>16</sub>" by overflow. Reading the timer Ai register after setting a value in the timer Ai register with a count halted but before the counter starts counting gets a proper value.
- (2) When stop counting in free run type, set timer again.
- (3) In the case of using as "Free-Run type", the timer register contents may be unknown when counting begins. If the timer register is set before counting has started, then the starting value will be unknown.
  - In the case where the up/down count will not be changed.

Enable the "Reload" function and write to the timer register before counting begins. Rewrite the value to the timer register immediately after counting has started. If counting up, rewrite "0000<sub>16</sub>" to the timer register. If counting down, rewrite "FFFF<sub>16</sub>" to the timer register. This will cause the same operation as "Free-Run type" mode.
  - In the case where the up/down count has changed.

First set to "Reload type" operation. Once the first counting pulse has occurred, the timer may be changed to "Free-Run type".

### Timer A (one-shot timer mode)

- (1) Setting the count start flag to "0" while a count is in progress causes as follows:
  - The counter stops counting and a content of reload register is reloaded.
  - The TAIOUT pin outputs "L" level.
  - The interrupt request generated and the timer Ai interrupt request bit goes to "1".
- (2) The output from the one-shot timer synchronizes with the count source generated internally. Therefore, when an external trigger has been selected, a delay of one cycle of count source as maximum occurs between the trigger input to the TAIIN pin and the one-shot timer output.
- (3) The timer Ai interrupt request bit goes to "1" if the timer's operation mode is set using any of the following procedures:
  - Selecting one-shot timer mode after reset.
  - Changing operation mode from timer mode to one-shot timer mode.
  - Changing operation mode from event counter mode to one-shot timer mode.

Therefore, to use timer Ai interrupt (interrupt request bit), set timer Ai interrupt request bit to "0" after the above listed changes have been made.

- (4) If a trigger occurs while a count is in progress, after the counter performs one down count following the reoccurrence of a trigger, the reload register contents are reloaded, and the count continues. To generate a trigger while a count is in progress, generate the second trigger after an elapse longer than one cycle of the timer's count source after the previous trigger occurred.
- (5) If an external trigger input is used to start counting, the next external trigger input must be avoided within 300ns before the timer A reaches "0000h".

### Timer A (pulse width modulation mode)

- (1) The timer Ai interrupt request bit becomes "1" if setting operation mode of the timer in compliance with any of the following procedures:
  - Selecting PWM mode after reset.
  - Changing operation mode from timer mode to PWM mode.
  - Changing operation mode from event counter mode to PWM mode.

Therefore, to use timer Ai interrupt (interrupt request bit), set timer Ai interrupt request bit to "0" after the above listed changes have been made.

- (2) Setting the count start flag to "0" while PWM pulses are being output causes the counter to stop counting. If the TAIOUT pin is outputting an "H" level in this instance, the output level goes to "L", and the timer Ai interrupt request bit goes to "1". If the TAIOUT pin is outputting an "L" level in this instance, the level does not change, and the timer Ai interrupt request bit does not becomes "1".

**Timer B (timer mode, event counter mode)**

- (1) The TBi (i=0 to 5) register indicates the countervalue during counting at any given time. However, the counter is "FFFF16" when reloading. The setting value can be read after setting the TBi register while the counter stops and before the counter starts counting.

**Timer B (pulse period/pulse width measurement mode)**

- (1) If the measurement mode select bit setting is changed after counting is started, the timer Bi interrupt request bit is set to "1".
- (2) Indeterminate values are transferred to the reload register during the first valid edge input after counting is started. The timer Bi interrupt request is not generated at this time.
- (3) The counter value is indeterminate when counting is started. Therefore, the timer Bi overflow flag setting may change to "1" and causes the timer Bi interrupt requests to be generated until a valid edge is input after counting is started.
- (4) The timer Bi overflow flag is set to "0" by writing to the timer Bi mode register at or after counting timing of the next count source, after the count start flag is set to "1" and the timer Bi overflow flag is set to "1".

**Stop Mode and Wait Mode**

- (1) To exit stop mode by hardware reset, provide an "L" signal input to the RESET pin until main clock oscillation is stable.
- (2) When entering wait mode, the instruction queue reads ahead to instructions following the WAIT instruction, and the program stops. Write at least 4 NOP instructions after the WAIT instruction.
- (3) When entering stop mode, the instruction lined in the instruction queue is executed before the interrupt for recovery is done. Write the JMP.B instruction, as follows, after the instruction setting the all clock stop control bit to "1".

```

                bset 0,prcr          ; protection removed
                bset 0,cm1          ; all clocks stopped (entering stop mode)
                jmp.b LABEL_001     ; JMP.B instruction executed (Jump to the next instruction soon
LABEL_001:      ; with no instruction between JMP.B and LABEL.)
                nop                 ; nop(1)
                nop                 ; nop(2)
                nop                 ; nop(3)
                nop                 ; nop(4)
                mov.b #0,prcr       ; protection set

```

(4) Use the following procedure to enter stop mode.

- Initial Setting

Set each interrupt priority level after setting the interrupt priority level required to exit stop mode, controlled by the RLVL2 to RLVL0 bits in the RLVL register, to "7".

- Before Entering Stop Mode

[1] Set the interrupt priority level of the interrupt being used to exit stop mode

[2] Set the interrupt priority levels of the interrupts, not being used to exit stop mode, to "0".

[3] Set the IPL in the FLG register. Then set the exit priority level to the same level as the IPL.

(Interrupt priority level of the interrupt used to exit stop mode > exit priority level  
 ≥ interrupt priority level of the interrupts not used to exit stop mode)

[4] Set the I flag to "1"

[5] Set the CM10 bit in the CM1 register to "1" (all clocks stop) after setting the PRC0 bit in the PRCR register to "1" (write enabled)

- After Exiting Stop Mode

Set the exit priority level to "7" as soon as exiting stop mode.

(5) When microcomputer enters stop mode again after exiting from stop mode using the NMI interrupt, use the following procedure to set the CM10 bit to "1".

[1] Exit stop mode using the NMI interrupt

[2] Generate a dummy interrupt

[3] Set the CM10 bit to "1"

Example:

```
INT    #63    ; Dummy interrupt
BSET   CM1    ; All clocks stopped (in stop mode)
```

```
; /*for dummy interrupt*/
```

```
DUMMY:
```

```
REIT
```

(6) Use the following procedure to enter wait mode.

- Initial Setting

Set each interrupt priority level after setting the interrupt priority level required to exit wait mode, controlled by the RLVL2 to RLVL0 bits in the RLVL register, to "7".

- Before Entering Wait Mode

[1] Set the interrupt priority level of the interrupt being used to exit wait mode

[2] Set the interrupt priority levels of the interrupts, not being used to exit wait mode, to "0".

[3] Set the IPL in the FLG register. Then set the exit priority level to the same level as the IPL.

(Interrupt priority level of the interrupt used to exit wait mode > exit priority level  
 ≥ interrupt priority level of the interrupts not used to exit wait mode)

[4] Set the I flag to "1"

[5] Execute the WAIT instruction

- After Exiting Wait Mode

Set the exit priority level to "7" as soon as exiting wait mode.



## A/D Converter

- (1) Write to each bit (except bit 6) of A/D control register 0, to each bit of A/D control register 1, and to bit 0 of A/D control register 2 when A/D conversion is stopped (before a trigger occurs).  
In particular, when the Vref connection bit is changed from "0" to "1", start A/D conversion after an elapse of 1  $\mu$ s or longer.
- (2) When changing A/D operation mode, select analog input pin again.
- (3) Using one-shot mode or single sweep mode  
Read the correspondence A/D register after confirming A/D conversion is finished. (It is known by A/D conversion interrupt request bit.)  
Use the undivided main clock as the internal CPU clock.
- (4) Using repeat mode, repeat sweep mode 0 or repeat sweep mode 1
- (5) When  $f(X_{IN})$  is faster than 10 MHz, make the frequency 10 MHz or less by dividing.
- (6) If A/D conversion is stopped by program while in progress of A/D conversion, the conversion result of A/D converter becomes indeterminate. The contents of A/D registers irrelevant to A/D conversion may become indeterminate. If A/D conversion is stopped by program while in progress of A/D conversion, ignore the values of all A/D registers.
- (7) Output impedance of sensor at A/D conversion (Reference value)

To carry out A/D conversion properly, charging the internal capacitor C shown in Figure 27.1 has to be completed within a specified period of time T. Let output impedance of sensor equivalent circuit be  $R_0$ , 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).

$$V_c \text{ is generally } V_c = V_{IN} \left\{ 1 - e^{-\frac{t}{C(R_0 + R)}} \right\}$$

$$\text{And when } t = T, \quad V_c = V_{IN} - \frac{X}{Y} V_{IN} = V_{IN} \left( 1 - \frac{X}{Y} \right)$$

$$e^{-\frac{T}{C(R_0 + R)}} = \frac{X}{Y}$$

$$-\frac{T}{C(R_0 + R)} = \ln \frac{X}{Y}$$

$$\text{Hence, } R_0 = -\frac{T}{C \cdot \ln \frac{X}{Y}} - R$$

With the model shown in Figure 27.1 as an example, when the difference between  $V_{IN}$  and  $V_C$  becomes 0.1LSB, we find impedance  $R_0$  when voltage between pins  $V_C$  changes from 0 to  $V_{IN} - (0.1/1024) V_{IN}$  in time  $T$ . (0.1/1024) means that A/D precision drop due to insufficient capacitor charge 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  $f(X_{IN}) = 10$  MHz,  $T = 0.3$   $\mu$ s in the A/D conversion mode with sample & hold. Output impedance  $R_0$  for sufficiently charging capacitor  $C$  within time  $T$  is determined as follows.

$T = 0.3$   $\mu$ s,  $R = 7.8$  k $\Omega$ ,  $C = 3$  pF,  $X = 0.1$ , and  $Y = 1024$ . Hence,

$$R_0 = - \frac{0.3 \times 10^{-6}}{3.0 \times 10^{-12} \cdot \ln \frac{0.1}{1024}} - 7.8 \times 10^3 \approx 3.0 \times 10^3$$

Thus, the allowable output impedance of the sensor circuit capable of thoroughly driving the A/D converter turns out to be approximately 3.0 k $\Omega$ . Tables 27.1 and 27.2 show output impedance values based on the LSB values.

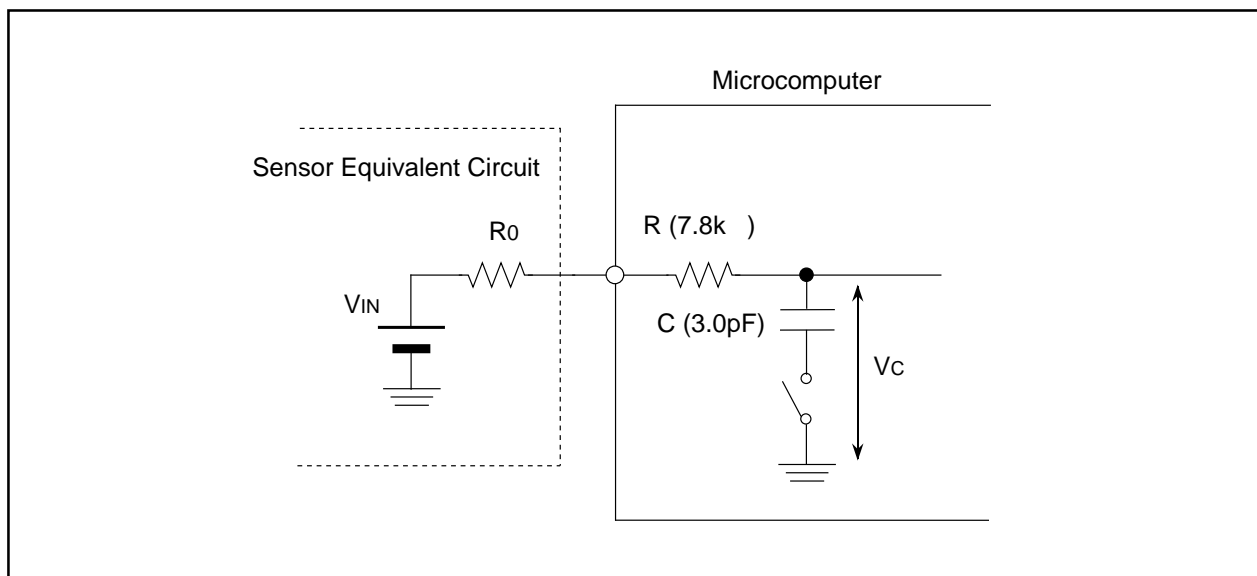


Figure 27.1 Analog Input Pin and External Sensor Equivalent Circuit

**Tables 27.1 Output impedance values based on the LSB values (10-bit mode) Reference value**

f(XIN) (MHz)	Cycle ( $\mu$ s)	Sampling time ( $\mu$ s)	R (Kohm)	C (pF)	Resolution (LSB)	R0max (Kohm)
10	0.1	0.3 (3 X cycle, Sample & hold bit is enabled)	7.8	3.0	0.1	3.0
					0.3	4.5
					0.5	5.3
					0.7	5.9
					0.9	6.4
					1.1	6.8
					1.3	7.2
					1.5	7.5
					1.7	7.8
10	0.1	0.2 (2 X cycle, Sample & hold bit is enabled)	7.8	3.0	0.3	0.4
					0.5	0.9
					0.7	1.3
					0.9	1.7
					1.1	2.0
					1.3	2.2
					1.5	2.4
					1.7	2.6
					1.9	2.8

**Tables 27.2 Output impedance values based on the LSB values (8-bit mode) Reference value**

f(XIN) (MHz)	Cycle ( $\mu$ s)	Sampling time ( $\mu$ s)	R (Kohm)	C (pF)	Resolution (LSB)	R0max (Kohm)
10	0.1	0.3 (3 X cycle, Sample & hold bit is enabled)	7.8	3.0	0.1	4.9
					0.3	7.0
					0.5	8.2
					0.7	9.1
					0.9	9.9
					1.1	10.5
					1.3	11.1
					1.5	11.7
					1.7	12.1
10	0.1	0.2 (2 X cycle, Sample & hold bit is enabled)	7.8	3.0	0.1	0.7
					0.3	2.1
					0.5	2.9
					0.7	3.5
					0.9	4.0
					1.1	4.4
					1.3	4.8
					1.5	5.2
					1.7	5.5
1.9	5.8					

## Interrupts

### (1) Setting the stack pointer

- The value of the stack pointer is initialized to 000000<sub>16</sub> immediately after reset. Accepting an interrupt before setting a value in the stack pointer may cause runaway. Be sure to set a value in the stack pointer before accepting an interrupt.

When using the  $\overline{\text{NMI}}$  interrupt, initialize the stack pointer at the beginning of a program. Regarding the first instruction immediately after reset, generating any interrupts including the  $\overline{\text{NMI}}$  interrupt is prohibited.

Set an even address to the stack pointer so that operating efficiency is increased.

### (2) The $\overline{\text{NMI}}$ interrupt

- As for the  $\overline{\text{NMI}}$  interrupt pin, an interrupt cannot be prohibited. Connect it to the VCC pin via a resistance (pulled-up) if unused.
- The  $\overline{\text{NMI}}$  pin also serves as P85, which is exclusively input. Reading the contents of the P8 register allows reading the pin value. Use the reading of this pin only for establishing the pin level at the time when the  $\overline{\text{NMI}}$  interrupt is input.
- Signals input to  $\overline{\text{NMI}}$  pin require "L" level and "H" level of 2 clock + 300ns or more, from the operation clock of CPU.

### (3) Address match interrupt

- Do not set the following addresses to the address match interrupt register.
  1. The address of the starting instruction in an interrupt routine.
  2. Any of the next 7 instructions addresses immediately after an instruction to clear an interrupt request bit of an interrupt control register or an instruction to rewrite an interrupt priority level to a smaller value.
  3. Any of the next 3 instructions addresses immediately after an instruction to set the interrupt enable flag (I flag).
  4. Any of the next 3 instructions addresses immediately after an instruction to rewrite a processor interrupt priority level (IPL) to a smaller value.

#### Example 1)

```

Interrupt_A:                                ; Interrupt A routine
        pushm  R0,R1,R2,R3,A0,A1           ; <---- Do not set address match interrupt to the
        .....                               ;      start address of an interrupt instruction

```

#### Example 2)

```

mov.b  #0,TA0IC      ;Change TA0 interrupt priority level to a smaller value
nop                                     ; 1st instruction
nop                                     ; 2nd instruction
nop                                     ; 3rd instruction
nop                                     ; 4th instruction
nop                                     ; 5th instruction
nop                                     ; 6th instruction
nop                                     ; 7th instruction

```

} Do not set address match interrupt during this period

#### Example 3)

```

fset  I      ; Set I flag ( interrupt enabled)
nop                                     ; 1st instruction
nop                                     ; 2nd instruction
nop                                     ; 3rd instruction

```

} Do not set address match interrupt during this period

## Example 4)

```

ldipl #0 ; Rewrite IPL to a smaller value
nop      ; 1st instruction
nop      ; 2nd instruction
nop      ; 3rd instruction

```

} Do not set address match interrupt during this period

- To return from an interrupt to the address set in an address match interrupt register using return instruction (reit or freit)

To rewrite the interrupt control register within the interrupt routine, add the below processing to the end of the routine (immediately before the reit or freit instruction). Also, if multiple interrupts are enabled with other interrupts, add the below processing to the end of the interrupt that enables the multiple interrupts.

If the interrupt control register is being rewritten within the non-maskable interrupt routine, add the below processing to the end of all interrupts.

## Additional process

```

; Execute after the register reset instruction (popm instruction)
fclr  U ; Select ISP (Unnecessary if the ISP has been selected)
pushm R0 ; Store R0 register
mov.w 6[SP],R0 ; Read FLG on stack (use "stc SVF,R0" when high-speed
; interrupt)
ldc  R0,FLG ; Set in FLG
popm R0 ; Restore R0 register
nop ; Dummy
reit ; Interrupt completed (use freit when high-speed interrupt)

```

## Example 5)

If rewriting the interrupt control register for interrupt B with the interrupt A routine and enabling multiple interrupts with interrupt C, the above processing is required at the end of the interrupt A and interrupt C routines.

## Interrupt A routine

```

Interrupt_A:
pushm R0,R1,R2,R3,A0,A1 ; Store registers
....
bclr  3,TA0IC ; Rewrite interrupt control register of interrupt B
....
popm  R0,R1,R2,R3,A0,A1 ; Restore registers
fclr  U ; Select ISP (Unnecessary if the ISP has been selected)
pushm R0 ; Store R0 register
mov.w 6[SP],R0 ; Read FLG on stack
ldc  R0,FLG ; Set in FLG
popm  R0 ; Restore R0 register
nop ; Dummy
reit ; Interrupt completed

```

```

Interrupt C routine
Interrupt_C:
    pushm R0,R1,R2,R3,A0,A1    ; Store registers
    fset    I                    ; Multiple interrupt enabled
    .....
    .....
    popm   R0,R1,R2,R3,A0,A1   ; Restore registers
    fclr   U                    ; Select ISP (Unnecessary if the ISP has been selected)
    pushm R0                    ; Store R0 register
    mov.w  6[SP],R0            ; Read FLG on stack
    ldc    R0,FLG              ; Set in FLG
    popm   R0                  ; Restore R0 register
    nop                               ; Dummy
    reit                               ; Interrupt completed

```

#### (4) External interrupt

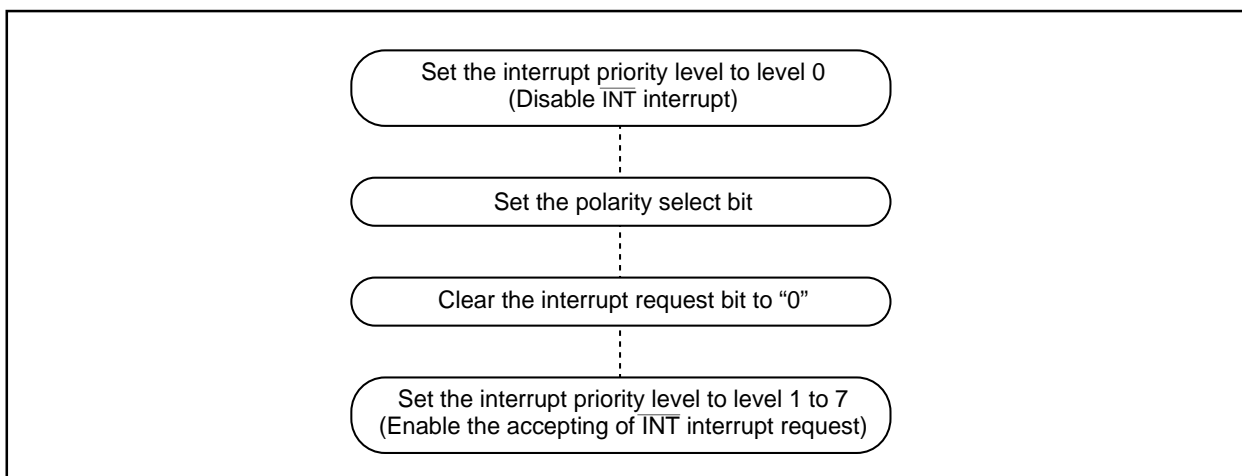
- Edge sense

Either an "L" level or an "H" level of at least 250 ns width is necessary for the signal input to pins INT0 to INT5 regardless of the CPU operation clock.

- Level sense

Either an "L" level or an "H" level of 1 cycle of BCLK + at least 200 ns width is necessary for the signal input to pins INT0 to INT5 regardless of the CPU operation clock. (When XIN=20MHz and no division mode, at least 250 ns width is necessary.)

- When the polarity of the INT0 to INT5 pins is changed, the interrupt request bit is sometimes set to "1". After changing the polarity, set the interrupt request bit to "0". Figure 27.2 shows the procedure for changing the  $\overline{\text{INT}}$  interrupt generate factor.



**Figure 27.2 Switching condition of  $\overline{\text{INT}}$  interrupt request**

## (5) Rewrite the interrupt control register

- When a instruction to rewrite the interrupt control register is executed but the interrupt is disabled, the interrupt request bit is not set sometimes even if the interrupt request for that register has been generated. This will depend on the instruction. If this creates problems, use the below instructions to change the register.

Instructions : AND, OR, BCLR, BSET

- When attempting to clear the interrupt request bit of an interrupt control register, the interrupt request bit is not cleared sometimes. This will depend on the instruction. If this creates problems, use the below instructions to change the register.

Instructions : MOV

**DMAC**

## (1) Do not clear the DMA request bit of the DMAi request cause select register.

In M16C/80, when a DMA request is generated while the channel is disabled (Note), the DMA transfer is not executed and the DMA request bit is cleared automatically.

Note :The DMA is disabled or the transfer count register is "0".

## (2) When DMA transfer is done by a software trigger, set DSR and DRQ of the DMAi request cause select register to "1" simultaneously using the OR instruction.

e.g.) OR.B #0A0h, DMiSL ; DMiSL is DMAi request cause select register

## (3) When changing the DMAi request cause select bit of the DMAi request cause select register, set "1" to the DMA request bit, simultaneously. In this case, set the corresponding DMA channel to disabled before changing the DMAi request cause select bit. At least 26 cycles are needed from the instruction to write to the DMAi request cause select register to enable DMA.

Example) When DMA request cause is changed to timer A0 and using DMA0 in single transfer after DMA initial setting

```

push.w    R0                ; Store R0 register
stc       DMD0, R0          ; Read DMA mode register 0
and.b     #11111100b, R0L    ; Clear DMA0 transfer mode select bit to "00"
ldc       R0, DMD0          ; DMA0 disabled
mov.b     #10000011b, DM0SL ; Select timer A0
                               ; (Write "1" to DMA request bit simultaneously)

push.w    R0                ; Sotre R0 register
mov.w     #6,R0              ;
dummy_loop:
sbjnz.w   #1,R0,dummy_loop  ; Dummy cycle
pop.w     R0                 ; Restore R0 register
or.b      #00000001b, R0L    ; Set DMA0 single transfer
ldc       R0, DMD0          ; DMA0 enabled
pop.w     R0                 ; Restore R0 register

```

} **At least 26 cycles are needed until DMA enabled.**

## (4) Recommended procedure for starting DMA transfer

- When writing to the DMAi request cause register including overwriting the same value to the DMAi request cause register;

1. Disable the corresponding channel i DMA in DMA mode registers 0 and 1.
2. Set up the peripheral used as the source of the DMA transfer. However, the peripheral should remain disabled at this time. For example, when using UART0 transmit, disable UART0 transmit.
3. Set the DMAi request cause select register. At this time, write a '1' to the DMA request bit (bit 7)
4. Set the following SFR registers:
  - DMAiSFR address register
  - DMAI memory address reload register
  - DMAi memory address register
  - DMAi transfer count reload register
  - DMAi transfer count register
5. At this point, if the number of elapsed cycles are less than 26, add code (NOP's or other processing) to make up some time.
6. Enable the corresponding channel i DMA in the DMA mode registers 0 and 1.
7. Enable the peripheral used as the source of the DMA transfer. For example, when using UART0 transmit, enable UART0 transmit.

- When not writing to the DMAi request cause register;

1. Disable the corresponding channel i DMA in the DMA mode registers 0 and 1.
2. Set up the peripheral used as the source of the DMA transfer. However, the peripheral should remain disabled at this time. For example, when using UART0 transmit, disable UART0 transmit.
3. Set up the following SFR registers:
  - DMAiSFR address register
  - DMAI memory address reload register
  - DMAi memory address register
  - DMAi transfer count reload register
  - DMAi transfer count register
4. Enable the corresponding channel i DMA in the DMA mode registers 0 and 1.
5. Enable the peripheral used as the source of the DMA transfer. For example, when using UART0 transmit, enable UART0 transmit.

## (5) Recommended procedure after completing DMA transfer

- Disable the peripheral used as source of the DMA transfer to prevent generating a DMA request.
- Disable the corresponding channel i DMA in the DMA mode registers 0 and 1.



## Noise

- (1) A bypass capacitor should be inserted between Vcc-Vss line for reducing noise and latch-up  
Connect a bypass capacitor (approx. 0.1 $\mu$ F) between the Vcc and Vss pins using short wiring and thicker circuit traces.

## Precautions for using CLKOUT pin

When using the Clock Output function of P53/CLKOUT pin (f8, f32 or fc output) in single chip mode, use port P57 as an input only port (port P57 direction register is "0").

Although port P57 may be set as an output port, it will become high impedance and will not output "H" or "L" levels.

## HOLD signal

When P40 to P47 and P50 to P52 are set to output port (the direction register is "1") in single-chip mode, then the MCU is changed to microprocessor mode or memory expansion mode.

Although the  $\overline{\text{HOLD}}$  pin may be held "L", P40 to P47 ( $\overline{\text{A16}}$  to  $\overline{\text{A23}}$ ,  $\overline{\text{CS0}}$  to  $\overline{\text{CS3}}$ , MA8 to MA12) and P50 to P52 ( $\overline{\text{RD/WR/BHE}}$ ,  $\overline{\text{RD/WRL/WRH}}$ ,  $\overline{\text{CASL/CASH/DW}}$ ) will not become high-impedance ports.

When using the  $\overline{\text{HOLD}}$  input while P40 to P47 and P50 to P52 are set as output ports in single-chip mode, you must first set all pins for P40 to P47 and P50 to P52 as input ports, then shift to microprocessor mode or memory expansion mode.

## Reducing power consumption

- (1) When A/D conversion is not performed, select the Vref not connected with the Vref connect bit of A/D control register 1. When A/D conversion is performed, start the A/D conversion at least 1  $\mu$ s or longer after connecting Vref.
- (2) When using AN4 (P104) to AN7 (P107), select the input disable of the key input interrupt signal with the key input interrupt disable bit of the function select register C .  
When selecting the input disable of the key input interrupt signal, the key input interrupt cannot be used. Also, the port cannot be input even if the direction register of P104 to P107 is set to input (the input result becomes undefined). When the input disable of the key input interrupt signal is selected, use all AN4 to AN7 as A/D inputs.
- (3) When ANEX0 and ANEX1 are used, select the input peripheral function disable with port P95 and P96 input peripheral function select bit of the function select register B3.  
When the input peripheral function disable is selected, the port cannot be input even if the port direction register is set to input (the input result becomes undefined).  
Also, it is not possible to input a peripheral function except ANEX0 and ANEX1.
- (4) When D/A converter is not used, set output disabled with the D/A output enable bit of D/A control register and set the D/A register to "0016".
- (5) When D/A conversion is used, select the input peripheral function disabled with port P93 and P94 input peripheral function select bit of the function select register B3.  
When the input peripheral function disabled is selected, the port cannot be input even if the port direction register is set to input (the input result becomes undefined).  
Also, it is not possible to input a peripheral function.

## DRAM controller

When shifting to self-refresh, select DRAM ignored by the DRAM space select bit. In the next instruction, simultaneously set the DRAM space select bit and self-refresh ON by self-refresh mode bit. Also, insert two NOPs after the instruction that sets the self-refresh mode bit to "1".

Do not access external memory while operating in self-refresh. (All external memory space access is inhibited. )

When disabling self-refresh, simultaneously select DRAM ignored by the DRAM space select bit and self-refresh OFF by self-refresh mode bit. In the next instruction, set the DRAM space select bit.

Do not access the DRAM space immediately after setting the DRAM space select bit.

Example) One wait is selected by the wait select bit and 4MB is selected by the DRAM space select bit

Shifting to self-refresh

```

...
mov.b #00000001b,DRAMCONT ;DRAM ignored, one wait is selected
mov.b #10001011b,DRAMCONT ;Set self-refresh, select 4MB and one wait
nop ;Two nops are needed
nop ;
...

```

Disable self-refresh

```

...
mov.b #00000001b,DRAMCONT ;Disable self-refresh, DRAM ignored, one wait is
                          ;selected
mov.b #00001011b,DRAMCONT ;Select 4MB and one wait
nop ;Inhibit instruction to access DRAM area
nop
...

```

## Setting the registers

The registers shown in Table 27.3 include indeterminate bit when read. Set immediate to these registers. Store the content of the frequently used register to RAM, change the content of RAM, then transfer to the register.

**Table 27.3 The object registers**

Register name	Symbol	Address
UART4 bit rate generator	U4BRG	02F9 <sub>16</sub>
UART4 transfer buffer register	U4TB	02FB <sub>16</sub> , 02FA <sub>16</sub>
Dead time timer	DTT	030C <sub>16</sub>
Timer B2 interrupt occurrence frequency set counter	ICTB2	030D <sub>16</sub>
UART3 bit rate generator	U3BRG	0329 <sub>16</sub>
UART3 transfer buffer register	U3TB	032B <sub>16</sub> , 032A <sub>16</sub>
UART2 bit rate generator	U2BRG	0339 <sub>16</sub>
UART2 transfer buffer register	U2TB	033B <sub>16</sub> , 033A <sub>16</sub>
Up-down flag	UDF	0344 <sub>16</sub>
Timer A0 register (Note)	TA0	0347 <sub>16</sub> , 0346 <sub>16</sub>
Timer A1 register (Note)	TA1	0349 <sub>16</sub> , 0348 <sub>16</sub>
Timer A2 register (Note)	TA2	034B <sub>16</sub> , 034A <sub>16</sub>
Timer A3 register (Note)	TA3	034D <sub>16</sub> , 034C <sub>16</sub>
Timer A4 register (Note)	TA4	034F <sub>16</sub> , 034E <sub>16</sub>
UART0 bit rate generator	U0BRG	0361 <sub>16</sub>
UART0 transfer buffer register	U0TB	0363 <sub>16</sub> , 0362 <sub>16</sub>
UART1 bit rate generator	U1BRG	0369 <sub>16</sub>
UART1 transfer buffer register	U1TB	036B <sub>16</sub> , 036A <sub>16</sub>

Note: In one-shot timer mode and pulse width modulation mode.

## External ROM version (144-pin version)

The external ROM version is operated only in microprocessor mode, so be sure to perform the following:

- Connect CNVss pin to Vcc.

## Notes on CNVss pin reset at "H" level

When the CNVss pin is reset at "H" level, the contents of internal ROM cannot be read out.

### Microprocessor mode or Memory expansion mode

When the MCU enters wait mode while operating in memory expansion mode or microprocessor mode, a pin functioning as part of the address or data bus retains its state on the bus before wait mode is entered. Shift to single-chip mode and output an arbitrary value in order to reduce current consumption. By shifting to single-chip mode, a pin which was functioning as part of the bus becomes a general-purpose port and can output an arbitrary value. Set the port registers and direction registers after shifting to single-chip mode (this implies that any control pins (CS,WR,RD,etc..) being used for access of an external device be changed as well).

If the port registers and direction registers are set while in memory expansion mode or microprocessor mode, the operation will be ignored.

This is similar when entering stop mode.

Setting procedure is following.

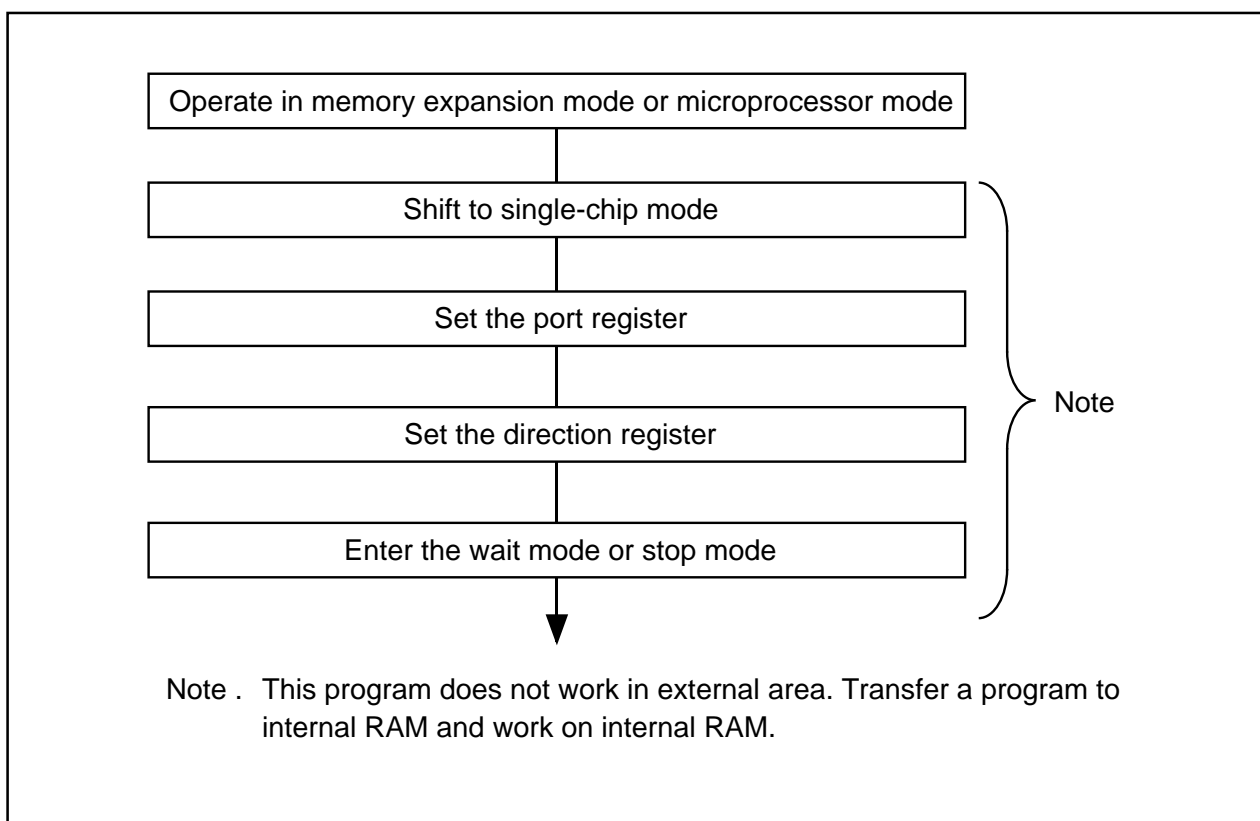


Figure 27.3 Setting procedure of the port register and direction register.

### Microprocessor

If the software reset is executed when the CNVss pin is connected to Vcc to start up in microprocessor mode, write at least three NOP instructions following the writing instruction to the PM0 Register.

example:

```

mov.b #02H,PRCR
bset 3,PM0      ; or "mov.b #8BH,PM0" (instruction to execute software reset)
nop             ; write at least three NOP instructions
nop
nop
nop

```

**Flash memory version**

Bit 7 and bit 6 of the processor mode register 1 (address 000516) must be set to "112" and this setting should be done when the main clock is divided by 8.

**Rewrite program of external ROM version with built-in boot loader**

- Do not use interrupts in rewrite program.
- Do not use absolute address jump instructions (JMP.A, JMPL.A) and absolute address subroutine call instructions (JSR.A, JSRL.A) in rewrite program.

## 28. Electrical characteristics

**Table 28.1 Absolute maximum ratings**

Symbol	Parameter		Condition	Rated value	Unit
V <sub>cc</sub>	Supply voltage		V <sub>cc</sub> =AV <sub>cc</sub>	-0.3 to 6.5	V
AV <sub>cc</sub>	Analog supply voltage		V <sub>cc</sub> =AV <sub>cc</sub>	-0.3 to 6.5	V
V <sub>i</sub>	Input voltage	RESET <sub>̄</sub> , (maskROM : CNV <sub>ss</sub> , BYTE), P00-P07, P10-P17, P20-P27, P30-P37, P40-P47, P50-P57, P60-P67, P72-P77, P80-P87, P90-P97, P100-P107, P110-P114, P120-P127, P130-P137, P140-P146, P150-P157, V <sub>REF</sub> , X <sub>IN</sub> (Note 1)		-0.3 to V <sub>cc</sub> +0.3	V
		P70, P71		-0.3 to 6.5	V
V <sub>o</sub>	Output voltage	P00-P07, P10-P17, P20-P27, P30-P37, P40-P47, P50-P57, P60-P67, P72-P77, P80-P84, P86, P87, P90-P97, P100-P107, P110-P114, P120-P127, P130-P137, P140-P146, P150-P157, X <sub>OUT</sub> (Note 1)		-0.3 to V <sub>cc</sub> +0.3	V
		P70, P71		-0.3 to 6.5	V
P <sub>d</sub>	Power dissipation		T <sub>opr</sub> =25 °C	500	mW
T <sub>opr</sub>	Operating ambient temperature			-20 to 85 / -40 to 85 (Note 2)	°C
T <sub>stg</sub>	Storage temperature			-65 to 150	°C

Note 1: Port P11 to P15 exist in 144-pin version.

Note 2: Specify a product of -40 to 85°C to use it.

**Table 28.2 Recommended operating conditions (referenced to V<sub>CC</sub> = 2.7V to 5.5V at Topr = – 20 to 85°C / – 40 to 85°C(Note3) unless otherwise specified)**

Symbol	Parameter		Standard			Unit
			Min	Typ.	Max.	
V <sub>CC</sub>	Supply voltage		2.7	5.0	5.5	V
AV <sub>CC</sub>	Analog supply voltage			V <sub>CC</sub>		V
V <sub>SS</sub>	Supply voltage			0		V
AV <sub>SS</sub>	Analog supply voltage			0		V
V <sub>IH</sub>	HIGH input voltage	P40-P47, P50-P57, P60-P67, P72-P77, P80-P87, P90-P97, P100-P107, P110-P114, P120-P127, P130-P137, P140-P146, P150-P157 (Note 5), X <sub>IN</sub> , RESET, CNV <sub>SS</sub> , BYTE	0.8V <sub>CC</sub>		V <sub>CC</sub>	V
		P70, P71	0.8V <sub>CC</sub>		6.5	V
		P00-P07, P10-P17, P20-P27, P30-P37 (during single-chip mode)	0.8V <sub>CC</sub>		V <sub>CC</sub>	V
		P00-P07, P10-P17, P20-P27, P30-P37 (data input function during memory expansion and microprocessor modes)	0.5V <sub>CC</sub>		V <sub>CC</sub>	V
V <sub>IL</sub>	LOW input voltage	P40-P47, P50-P57, P60-P67, P70-P77, P80-P87, P90-P97, P100-P107, P110-P114, P120-P127, P130-P137, P140-P146, P150-P157 (Note 5), X <sub>IN</sub> , RESET, CNV <sub>SS</sub> , BYTE	0		0.2V <sub>CC</sub>	V
		P00-P07, P10-P17, P20-P27, P30-P37 (during single-chip mode)	0		0.2V <sub>CC</sub>	V
		P00-P07, P10-P17, P20-P27, P30-P37 (data input function during memory expansion and microprocessor modes)	0		0.16V <sub>CC</sub>	V
I <sub>OH (peak)</sub>	HIGH peak output current	P00-P07, P10-P17, P20-P27, P30-P37, P40-P47, P50-P57, P60-P67, P72-P77, P80-P84, P86, P87, P90-P97, P100-P107, P110-P114, P120-P127, P130-P137, P140-P146, P150-P157 (Note 5)			-10.0	mA
I <sub>OH (avg)</sub>	HIGH average output current	P00-P07, P10-P17, P20-P27, P30-P37, P40-P47, P50-P57, P60-P67, P72-P77, P80-P84, P86, P87, P90-P97, P100-P107, P110-P114, P120-P127, P130-P137, P140-P146, P150-P157 (Note 5)			-5.0	mA
I <sub>OL (peak)</sub>	LOW peak output current	P00-P07, P10-P17, P20-P27, P30-P37, P40-P47, P50-P57, P60-P67, P70-P77, P80-P84, P86, P87, P90-P97, P100-P107, P110-P114, P120-P127, P130-P137, P140-P146, P150-P157 (Note 5)			10.0	mA
I <sub>OL (avg)</sub>	LOW average output current	P00-P07, P10-P17, P20-P27, P30-P37, P40-P47, P50-P57, P60-P67, P70-P77, P80-P84, P86, P87, P90-P97, P100-P107, P110-P114, P120-P127, P130-P137, P140-P146, P150-P157 (Note 5)			5.0	mA
f (X <sub>IN</sub> )	Main clock input oscillation frequency	No wait	V <sub>CC</sub> =4.2V to 5.5V	0	20	MHz
			V <sub>CC</sub> =2.7V to 4.2V	0	10	MHz
f (X <sub>CIN</sub> )	Subclock oscillation frequency			32.768	50	kHz

Note 1: The mean output current is the mean value within 100ms.

Note 2: The total I<sub>OL</sub> (peak) for ports P0, P1, P2, P86, P87, P9, P10, P11, P14 and P15 must be 80mA max. The total I<sub>OH</sub> (peak) for ports P0, P1, P2, P86, P87, P9, P10, P11, P14 and P15 must be -80mA max. The total I<sub>OL</sub> (peak) for ports P3, P4, P5, P6, P7, P80 to P84, P12 and P13 must be 80mA max. The total I<sub>OH</sub> (peak) for ports P3, P4, P5, P6, P72 to P77, P80 to P84, P12 and P13 must be -80mA max.

Note 3: Specify a product of -40 to 85°C to use it.

Note 4: The specification of V<sub>IH</sub> and V<sub>IL</sub> of P87 is not when using as X<sub>CIN</sub> but when using programmable input port.

Note 5: Port P11 to P15 exist in 144-pin version.

**Table 28.3 A/D conversion characteristics (referenced to  $V_{CC} = AV_{CC} = V_{REF} = 5V$ ,  $V_{SS} = AV_{SS} = 0V$  at  $T_{opr} = 25^{\circ}C$ ,  $f(X_{IN})=20MHz$  unless otherwise specified)**

Symbol	Parameter		Measurement Condition	Standard			Unit	
				Min.	Typ.	Max.		
-	Resolution		$V_{REF}=V_{CC}$			10	Bits	
-	Absolute accuracy (10 bits)		$V_{REF}=V_{CC}=5V$	AN <sub>0</sub> to AN <sub>7</sub> input			$\pm 3$	LSB
				ANEX <sub>0</sub> , ANEX <sub>1</sub> input				
			External op-amp connection mode				$\pm 7$	LSB
	Absolute accuracy (8 bits)		$V_{REF}=V_{CC}=5V$			$\pm 2$	LSB	
	Absolute accuracy (8 bits)	Sample & hold function not available	$V_{REF}=V_{CC}=3V$ , $\emptyset AD=f_{AD}/2$			$\pm 2$	LSB	
RLADDER	Ladder resistance		$V_{REF}=V_{CC}$	10		40	k $\Omega$	
t <sub>CONV</sub>	Conversion time (10 bits)	Sample & hold function available	$V_{REF}=V_{CC}=5V$ , $\emptyset AD=10MHz$	3.3			$\mu s$	
t <sub>CONV</sub>	Conversion time (8 bits)	Sample & hold function available	$V_{REF}=V_{CC}=5V$ , $\emptyset AD=10MHz$	2.8			$\mu s$	
t <sub>CONV</sub>	Conversion time (8 bits)	Sample & hold function not available	$V_{REF}=V_{CC}=3V$ , $\emptyset AD=f_{AD}/2=5MHz$	9.8			$\mu s$	
t <sub>SAMP</sub>	Sampling time			0.3			$\mu s$	
V <sub>REF</sub>	Reference voltage		$V_{REF}=V_{CC}=4.2$ to $5.5V$	2.0			V	
			$V_{REF}=V_{CC}=2.7$ to $5.5V$	2.7			V	
V <sub>IA</sub>	Analog input voltage			0		V <sub>REF</sub>	V	

Note 1: DO  $f(X_{IN})$  in range of main clock input oscillation frequency prescribed with recommended operating conditions of table 28.2. Divide the  $f_{AD}$  if  $f(X_{IN})$  exceeds 10 MHz, and make AD operation clock frequency ( $\emptyset AD$ ) equal to or lower than 10 MHz. And divide the  $f_{AD}$  if  $V_{CC}$  is less than 4.2V, and make AD operation clock frequency ( $\emptyset AD$ ) equal to or lower than  $f_{AD}/2$ .

Note 2 : A case without sample & hold function turn AD operation clock frequency ( $\emptyset AD$ ) into 250 kHz or more in addition to a limit of Note 1.

Note 3 : Connect AV<sub>CC</sub> pin to V<sub>CC</sub> pin and apply the same electric potential.

**Table 28.4 D/A conversion characteristics (referenced to  $V_{CC} = V_{REF} = 5V$ ,  $V_{SS} = AV_{SS} = 0V$  at  $T_{opr} = 25^{\circ}C$ ,  $f(X_{IN})=20MHz$  unless otherwise specified)**

Symbol	Parameter		Measuring condition	Standard			Unit
				Min.	Typ.	Max.	
-	Resolution					8	Bits
-	Absolute accuracy					1.0	%
t <sub>su</sub>	Setup time					3	$\mu s$
R <sub>o</sub>	Output resistance			4	10	20	k $\Omega$
I <sub>VREF</sub>	Reference power supply input current		$V_{REF} = V_{CC} = 5V$ (Note 1)			1.5	mA
			$V_{REF} = V_{CC} = 3V$ (Note 1)			1.0	mA

Note 1: This applies when using one D/A converter, with the D/A register for the unused D/A converter set to "0016".

The A/D converter's ladder resistance is not included.

Also, when D/A register contents are not "0016" the current I<sub>VREF</sub> always flows even though V<sub>ref</sub> may have been set to be unconnected by the A/D control register.



VCC = 5V

**Table 28.5 Electrical characteristics (referenced to VCC=5V, VSS=0V at Topr=25°C, f(XIN)=20MHZ unless otherwise specified)**

Symbol	Parameter		Measuring condition	Standard			Unit			
				Min	Typ.	Max.				
V <sub>OH</sub>	HIGH output voltage	P00-P07, P10-P17, P20-P27, P30-P37, P40-P47, P50-P57, P60-P67, P72-P77, P80-P84, P86, P87, P90-P97, P100-P107, P110-P114, P120-P127, P130-P137, P140-P146, P150-P157 (Note 1)	I <sub>OH</sub> = - 5mA, V <sub>CC</sub> =5.0V	3.0			V			
V <sub>OH</sub>	HIGH output voltage	P00-P07, P10-P17, P20-P27, P30-P37, P40-P47, P50-P57, P60-P67, P72-P77, P80-P84, P86, P87, P90-P97, P100-P107, P110-P114, P120-P127, P130-P137, P140-P146, P150-P157 (Note 1)	I <sub>OH</sub> = - 200µA, V <sub>CC</sub> =5.0V	4.7			V			
V <sub>OH</sub>	HIGH output voltage	X <sub>OUT</sub>	HIGHPOWER	I <sub>OH</sub> = - 1mA, V <sub>CC</sub> =5.0V	3.0		V			
			LOWPOWER	I <sub>OH</sub> = - 0.5mA, V <sub>CC</sub> =5.0V	3.0		V			
	HIGH output voltage	X <sub>COU</sub> T	HIGHPOWER	With no load applied, V <sub>CC</sub> =5.0V		3.0	V			
			LOWPOWER	With no load applied, V <sub>CC</sub> =5.0V		1.6	V			
V <sub>OL</sub>	LOW output voltage	P00-P07, P10-P17, P20-P27, P30-P37, P40-P47, P50-P57, P60-P67, P70-P77, P80-P84, P86, P87, P90-P97, P100-P107, P110-P114, P120-P127, P130-P137, P140-P146, P150-P157 (Note 1)	I <sub>OL</sub> =5mA, V <sub>CC</sub> =5.0V			2.0	V			
V <sub>OL</sub>	LOW output voltage	P00-P07, P10-P17, P20-P27, P30-P37, P40-P47, P50-P57, P60-P67, P70-P77, P80-P84, P86, P87, P90-P97, P100-P107, P110-P114, P120-P127, P130-P137, P140-P146, P150-P157 (Note 1)	I <sub>OL</sub> =200µA, V <sub>CC</sub> =5.0V			0.45	V			
V <sub>OL</sub>	LOW output voltage	X <sub>OUT</sub>	HIGHPOWER	I <sub>OL</sub> =1mA, V <sub>CC</sub> =5.0V		2.0	V			
			LOWPOWER	I <sub>OL</sub> =0.5mA, V <sub>CC</sub> =5.0V		2.0	V			
	LOW output voltage	X <sub>COU</sub> T	HIGHPOWER	With no load applied, V <sub>CC</sub> =5.0V		0	V			
			LOWPOWER	With no load applied, V <sub>CC</sub> =5.0V		0	V			
V <sub>T+</sub> -V <sub>T-</sub>	Hysteresis	HOLD, RDY, TA0IN-TA4IN, TB0IN-TB5IN, INT0-INT5, ADTRG, CTS0-CTS4, CLK0-CLK4, TA0OUT-TA4OUT, NMI, KI0-KI3, RxD0-RxD4, SCL2-SCL4, SDA2-SDA4	V <sub>CC</sub> =5.0V	0.2		1.0	V			
V <sub>T+</sub> -V <sub>T-</sub>	Hysteresis	RESET	V <sub>CC</sub> =5.0V	0.2		1.8	V			
I <sub>IH</sub>	HIGH input current	P00-P07, P10-P17, P20-P27, P30-P37, P40-P47, P50-P57, P60-P67, P70-P77, P80-P87, P90-P97, P100-P107, P110-P114, P120-P127, P130-P137, P140-P146, P150-P157, X <sub>IN</sub> , RESET, CNV <sub>SS</sub> , BYTE (Note 1)	V <sub>I</sub> =5V, V <sub>CC</sub> =5.0V			5.0	µA			
I <sub>IL</sub>	LOW input current	P00-P07, P10-P17, P20-P27, P30-P37, P40-P47, P50-P57, P60-P67, P70-P77, P80-P87, P90-P97, P100-P107, P110-P114, P120-P127, P130-P137, P140-P146, P150-P157, X <sub>IN</sub> , RESET, CNV <sub>SS</sub> , BYTE (Note 1)	V <sub>I</sub> =0V, V <sub>CC</sub> =5.0V			- 5.0	µA			
R <sub>PULLUP</sub>	Pull-up resistance	P00-P07, P10-P17, P20-P27, P30-P37, P40-P47, P50-P57, P60-P67, P72-P77, P80-P84, P86, P87, P90-P97, P100-P107, P110-P114, P120-P127, P130-P137, P140-P146, P150-P157 (Note 1)	V <sub>I</sub> =0V, V <sub>CC</sub> =5.0V	30.0	50.0	167.0	kΩ			
R <sub>FXIN</sub>	Feedback resistance	X <sub>IN</sub>			1.0		MΩ			
R <sub>FXCIN</sub>	Feedback resistance	X <sub>CIN</sub>			6.0		MΩ			
V <sub>RAM</sub>	RAM retention voltage		When clock is stopped	2.0			V			
I <sub>CC</sub>	Power supply current	Measuring condition: In single-chip mode, the output pins are open and other pins are VSS	f(XIN)=20MHz Square wave, no division	Mask ROM 128 KB version	45.0	72.0	mA			
				ROMless RAM 10 KB version (Note 2)						
				Mask ROM 256 KB version	50.0	80.0				
							ROMless RAM 24 KB version (Note 2)			
							Flash memory version	50.0	80.0	
						f(XCIN)=32kHz Square wave	Mask ROM 128 KB version	90.0		µA
							ROMless RAM 10 KB version (Note 2)			
							Mask ROM 256 KB version	100.0		
							ROMless RAM 24 KB version (Note 2)			
							Flash memory version	7.0		mA
		f(XCIN)=32kHz When a WAIT instruction is executed		4.0		µA				
			Topr=25°C when clock is stopped	Mask ROM 128 KB version		1.0	µA			
				ROMless RAM 10KB version (Note 2)						
		Mask ROM 256 KB version			2.0					
			ROMless RAM 24KB version (Note 2)							
			Flash memory version		1.0					
			Topr=85°C when clock is stopped		20.0					

Note 1: Port P11 to P15 exist in 144-pin version.

Note 2: ROMless version exists in 144-pin version.

$$V_{CC} = 5V$$

Timing requirements (referenced to  $V_{CC} = 5V$ ,  $V_{SS} = 0V$  at  $T_{op} = 25^{\circ}C$  unless otherwise specified)

**Table 28.6 External clock 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	22		ns
$t_{w(L)}$	External clock input LOW pulse width	22		ns
$t_r$	External clock rise time		5	ns
$t_f$	External clock fall time		5	ns

**Table 28.7 Memory expansion and microprocessor modes**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{ac1(RD-DB)}$	Data input access time (RD standard, no wait)		(Note)	ns
$t_{ac1(AD-DB)}$	Data input access time (AD standard, CS standard, no wait)		(Note)	ns
$t_{ac2(RD-DB)}$	Data input access time (RD standard, with wait)		(Note)	ns
$t_{ac2(AD-DB)}$	Data input access time (AD standard, CS standard, with wait)		(Note)	ns
$t_{ac3(RD-DB)}$	Data input access time (RD standard, when accessing multiplex bus area)		(Note)	ns
$t_{ac3(AD-DB)}$	Data input access time (AD standard, CS standard, when accessing multiplex bus area)		(Note)	ns
$t_{ac4(RAS-DB)}$	Data input access time (RAS standard, DRAM access)		(Note)	ns
$t_{ac4(CAS-DB)}$	Data input access time (CAS standard, DRAM access)		(Note)	ns
$t_{ac4(CAD-DB)}$	Data input access time (CAD standard, DRAM access)		(Note)	ns
$t_{su(DB-BCLK)}$	Data input setup time	26		ns
$t_{su(RDY-BCLK)}$	$\overline{RDY}$ input setup time	26		ns
$t_{su(HOLD-BCLK)}$	HOLD input setup time	30		ns
$t_h(RD-DB)$	Data input hold time	0		ns
$t_h(CAS-DB)$	Data input hold time	0		ns
$t_h(BCLK-RDY)$	$\overline{RDY}$ input hold time	0		ns
$t_h(BCLK-HOLD)$	HOLD input hold time	0		ns
$t_d(BCLK-HLDA)$	HLDA output delay time		25	ns

Note: Calculated according to the BCLK frequency as follows:

Note that inserting wait or using lower operation frequency  $f(BCLK)$  is needed when calculated value is negative.

$$t_{ac1(RD-DB)} = \frac{10^9}{f(BCLK) \times 2} - 35 \quad [ns]$$

$$t_{ac1(AD-DB)} = \frac{10^9}{f(BCLK)} - 35 \quad [ns]$$

$$t_{ac2(RD-DB)} = \frac{10^9 \times m}{f(BCLK) \times 2} - 35 \quad [ns] \quad (m=3, 5 \text{ and } 7 \text{ when } 1 \text{ wait, } 2 \text{ wait and } 3 \text{ wait, respectively})$$

$$t_{ac2(AD-DB)} = \frac{10^9 \times n}{f(BCLK)} - 35 \quad [ns] \quad (n=2, 3 \text{ and } 4 \text{ when } 1 \text{ wait, } 2 \text{ wait and } 3 \text{ wait, respectively})$$

$$t_{ac3(RD-DB)} = \frac{10^9 \times m}{f(BCLK) \times 2} - 35 \quad [ns] \quad (m=3 \text{ and } 5 \text{ when } 2 \text{ wait and } 3 \text{ wait, respectively})$$

$$t_{ac3(AD-DB)} = \frac{10^9 \times n}{f(BCLK) \times 2} - 35 \quad [ns] \quad (n=5 \text{ and } 7 \text{ when } 2 \text{ wait and } 3 \text{ wait, respectively})$$

$$t_{ac4(RAS-DB)} = \frac{10^9 \times m}{f(BCLK) \times 2} - 35 \quad [ns] \quad (m=3 \text{ and } 5 \text{ when } 1 \text{ wait and } 2 \text{ wait, respectively})$$

$$t_{ac4(CAS-DB)} = \frac{10^9 \times n}{f(BCLK) \times 2} - 35 \quad [ns] \quad (n=1 \text{ and } 3 \text{ when } 1 \text{ wait and } 2 \text{ wait, respectively})$$

$$t_{ac4(CAD-DB)} = \frac{10^9 \times l}{f(BCLK)} - 35 \quad [ns] \quad (l=1 \text{ and } 2 \text{ when } 1 \text{ wait and } 2 \text{ wait, respectively})$$

$$V_{CC} = 5V$$

Timing requirements (referenced to  $V_{CC} = 5V$ ,  $V_{SS} = 0V$  at  $T_{opr} = 25^{\circ}C$  unless otherwise specified)

**Table 28.8 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 28.9 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 28.10 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 28.11 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 28.12 Timer A input (up/down 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

$$V_{CC} = 5V$$

Timing requirements (referenced to  $V_{CC} = 5V$ ,  $V_{SS} = 0V$  at  $T_{opr} = 25^{\circ}C$  unless otherwise specified)

**Table 28.13 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 28.14 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 28.15 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 28.16 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 28.17 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	30		ns
$t_{h(C-D)}$	RxDi input hold time	90		ns

**Table 28.18 External interrupt  $\overline{INTi}$  inputs**

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

$$V_{CC} = 5V$$

Switching characteristics (referenced to  $V_{CC} = 5V$ ,  $V_{SS} = 0V$  at  $T_{opr} = 25^{\circ}C$  unless otherwise specified)

Table 28.19 Memory expansion mode and microprocessor mode (no wait)

Symbol	Parameter	Measuring condition	Standard		Unit
			Min.	Max.	
$t_d(\text{BCLK-AD})$	Address output delay time	Figure 28.1		18	ns
$t_h(\text{BCLK-AD})$	Address output hold time (BCLK standard)		-3		ns
$t_h(\text{RD-AD})$	Address output hold time (RD standard)		0		ns
$t_h(\text{WR-AD})$	Address output hold time (WR standard)		(Note 1)		ns
$t_d(\text{BCLK-CS})$	Chip select output delay time			18	ns
$t_h(\text{BCLK-CS})$	Chip select output hold time (BCLK standard)		-3		ns
$t_h(\text{RD-CS})$	Chip select output hold time (RD standard)		0		ns
$t_h(\text{WR-CS})$	Chip select output hold time (WR standard)		(Note 1)		ns
$t_d(\text{BCLK-ALE})$	ALE signal output delay time			18	ns
$t_h(\text{BCLK-ALE})$	ALE signal output hold time		-2		ns
$t_d(\text{BCLK-RD})$	RD signal output delay time			10	ns
$t_h(\text{BCLK-RD})$	RD signal output hold time		-5		ns
$t_d(\text{BCLK-WR})$	WR signal output delay time			18	ns
$t_h(\text{BCLK-WR})$	WR signal output hold time		-3		ns
$t_d(\text{DB-WR})$	Data output delay time (WR standard)		(Note 1)		ns
$t_h(\text{WR-DB})$	Data output hold time (WR standard)		(Note 1)		ns
$t_w(\text{WR})$	WR signal width		(Note 1)		ns

Note 1: Calculated according to the BCLK frequency as follows:

$$t_{d(\text{DB-WR})} = \frac{10^9}{f_{(\text{BCLK})}} - 20 \quad [\text{ns}]$$

$$t_{h(\text{WR-DB})} = \frac{10^9}{f_{(\text{BCLK})} \times 2} - 10 \quad [\text{ns}]$$

$$t_{h(\text{WR-AD})} = \frac{10^9}{f_{(\text{BCLK})} \times 2} - 10 \quad [\text{ns}]$$

$$t_{h(\text{WR-CS})} = \frac{10^9}{f_{(\text{BCLK})} \times 2} - 10 \quad [\text{ns}]$$

$$t_w(\text{WR}) = \frac{10^9}{f_{(\text{BCLK})} \times 2} - 15 \quad [\text{ns}]$$

$$V_{CC} = 5V$$

**Switching characteristics (referenced to  $V_{CC} = 5V$ ,  $V_{SS} = 0V$  at  $T_{opr} = 25^{\circ}C$  unless otherwise specified)**

**Table 28.20 Memory expansion mode and microprocessor mode  
(with wait, accessing external memory)**

Symbol	Parameter	Measuring condition	Standard		Unit
			Min.	Max.	
$t_d(\text{BCLK-AD})$	Address output delay time	Figure 28.1		18	ns
$t_h(\text{BCLK-AD})$	Address output hold time (BCLK standard)		- 3		ns
$t_h(\text{RD-AD})$	Address output hold time (RD standard)		0		ns
$t_h(\text{WR-AD})$	Address output hold time (WR standard)		(Note 1)		ns
$t_d(\text{BCLK-CS})$	Chip select output delay time			18	ns
$t_h(\text{BCLK-CS})$	Chip select output hold time (BCLK standard)		- 3		ns
$t_h(\text{RD-CS})$	Chip select output hold time (RD standard)		0		ns
$t_h(\text{WR-CS})$	Chip select output hold time (WR standard)		(Note 1)		ns
$t_d(\text{BCLK-ALE})$	ALE signal output delay time			18	ns
$t_h(\text{BCLK-ALE})$	ALE signal output hold time		- 2		ns
$t_d(\text{BCLK-RD})$	RD signal output delay time			10	ns
$t_h(\text{BCLK-RD})$	RD signal output hold time		- 5		ns
$t_d(\text{BCLK-WR})$	WR signal output delay time			18	ns
$t_h(\text{BCLK-WR})$	WR signal output hold time		- 3		ns
$t_d(\text{DB-WR})$	Data output delay time (WR standard)		(Note 1)		ns
$t_h(\text{WR-DB})$	Data output hold time (WR standard)		(Note 1)		ns
$t_w(\text{WR})$	WR signal width		(Note 1)		ns

Note 1: Calculated according to the BCLK frequency as follows:

$$t_d(\text{DB-WR}) = \frac{10^9 \times n}{f(\text{BCLK})} - 20 \quad [\text{ns}] \quad (n=1, 2 \text{ and } 3 \text{ when } 1 \text{ wait, } 2 \text{ wait and } 3 \text{ wait, respectively})$$

$$t_h(\text{WR-DB}) = \frac{10^9}{f(\text{BCLK}) \times 2} - 10 \quad [\text{ns}]$$

$$t_h(\text{WR-AD}) = \frac{10^9}{f(\text{BCLK}) \times 2} - 10 \quad [\text{ns}]$$

$$t_h(\text{WR-CS}) = \frac{10^9}{f(\text{BCLK}) \times 2} - 10 \quad [\text{ns}]$$

$$t_w(\text{WR}) = \frac{10^9 \times n}{f(\text{BCLK}) \times 2} - 15 \quad [\text{ns}] \quad (n=1, 3 \text{ and } 5 \text{ when } 1 \text{ wait, } 2 \text{ wait and } 3 \text{ wait, respectively})$$

$$V_{CC} = 5V$$

**Switching characteristics (referenced to  $V_{CC} = 5V$ ,  $V_{SS} = 0V$  at  $T_{opr} = 25^{\circ}C$  unless otherwise specified)**

**Table 28.21 Memory expansion mode and microprocessor mode  
(with wait, accessing external memory, multiplex bus area selected)**

Symbol	Parameter	Measuring condition	Standard		Unit
			Min.	Max.	
td(BCLK-AD)	Address output delay time	Figure 28.1		18	ns
th(BCLK-AD)	Address output hold time (BCLK standard)		-3		ns
th(RD-AD)	Address output hold time (RD standard)		(Note 1)		ns
th(WR-AD)	Address output hold time (WR standard)		(Note 1)		ns
td(BCLK-CS)	Chip select output delay time			18	ns
th(BCLK-CS)	Chip select output hold time (BCLK standard)		-3		ns
th(RD-CS)	Chip select output hold time (RD standard)		(Note 1)		ns
th(WR-CS)	Chip select output hold time (WR standard)		(Note 1)		ns
td(BCLK-RD)	RD signal output delay time			18	ns
th(BCLK-RD)	RD signal output hold time		-5		ns
td(BCLK-WR)	WR signal output delay time			18	ns
th(BCLK-WR)	WR signal output hold time		-3		ns
td(DB-WR)	Data output delay time (WR standard)		(Note 1)		ns
th(WR-DB)	Data output hold time (WR standard)		(Note 1)		ns
td(BCLK-ALE)	ALE signal output delay time (BCLK standard)			18	ns
th(BCLK-ALE)	ALE signal output hold time (BCLK standard)		-2		ns
td(AD-ALE)	ALE signal output delay time (address standard)		(Note 1)		ns
th(ALE-AD)	ALE signal output hold time (address standard)		(Note 1)		ns
tdz(RD-AD)	Address output floating start time			8	ns
th(BCLK-DB)	Data output hold time (BCLK standard)		-5		ns

Note 1: Calculated according to the BCLK frequency as follows:

$$th(RD - AD) = \frac{10^9}{f(BCLK) \times 2} - 10 \quad [ns]$$

$$th(WR - AD) = \frac{10^9}{f(BCLK) \times 2} - 10 \quad [ns]$$

$$th(RD - CS) = \frac{10^9}{f(BCLK) \times 2} - 10 \quad [ns]$$

$$th(WR - CS) = \frac{10^9}{f(BCLK) \times 2} - 10 \quad [ns]$$

$$td(DB - WR) = \frac{10^9 \times m}{f(BCLK) \times 2} - 25 \quad [ns] \quad (m=3 \text{ and } 5 \text{ when } 2 \text{ wait and } 3 \text{ wait, respectively})$$

$$th(WR - DB) = \frac{10^9}{f(BCLK) \times 2} - 10 \quad [ns]$$

$$td(AD - ALE) = \frac{10^9}{f(BCLK) \times 2} - 23 \quad [ns]$$

$$th(ALE - AD) = \frac{10^9}{f(BCLK) \times 2} - 10 \quad [ns]$$

$$V_{CC} = 5V$$

Switching characteristics (referenced to  $V_{CC} = 5V$ ,  $V_{SS} = 0V$  at  $T_{opr} = 25^{\circ}C$  unless otherwise specified)

**Table 28.22 Memory expansion mode and microprocessor mode  
(with wait, accessing external memory, DRAM area selected)**

Symbol	Parameter	Measuring condition	Standard		Unit
			Min.	Max.	
$t_{d(BCLK-RAD)}$	Row address output delay time	Figure 28.1		18	ns
$t_{h(BCLK-RAD)}$	Row address output hold time (BCLK standard)		-3		ns
$t_{d(BCLK-CAD)}$	String address output delay time			18	ns
$t_{h(BCLK-CAD)}$	String address output hold time (BCLK standard)		-3		ns
$t_{h(RAS-RAD)}$	Row address output hold time after RAS output		(Note 1)		ns
$t_{d(BCLK-RAS)}$	RAS output delay time (BCLK standard)			18	ns
$t_{h(BCLK-RAS)}$	RAS output hold time (BCLK standard)		-3		ns
$t_{RP}$	RAS "H" hold time		(Note 1)		ns
$t_{d(BCLK-CAS)}$	CAS output delay time (BCLK standard)			18	ns
$t_{h(BCLK-CAS)}$	CAS output hold time (BCLK standard)		-3		ns
$t_{d(BCLK-DW)}$	Data output delay time (BCLK standard)			18	ns
$t_{h(BCLK-DW)}$	Data output hold time (BCLK standard)		-5		ns
$t_{su(DB-CAS)}$	CAS after DB output setup time		(Note 1)		ns
$t_{h(BCLK-DB)}$	DB signal output hold time (BCLK standard)		-7		ns
$t_{su(CAS-RAS)}$	CAS before RAS setup time (refresh)		(Note 1)		ns

Note 1: Calculated according to the BCLK frequency as follows:

$$t_{h(RAS - RAD)} = \frac{10^9}{f_{(BCLK)} \times 2} - 13 \quad [ns]$$

$$t_{RP} = \frac{10^9 \times 3}{f_{(BCLK)} \times 2} - 20 \quad [ns]$$

$$t_{su(DB - CAS)} = \frac{10^9}{f_{(BCLK)}} - 20 \quad [ns]$$

$$t_{su(CAS - RAS)} = \frac{10^9}{f_{(BCLK)} \times 2} - 13 \quad [ns]$$



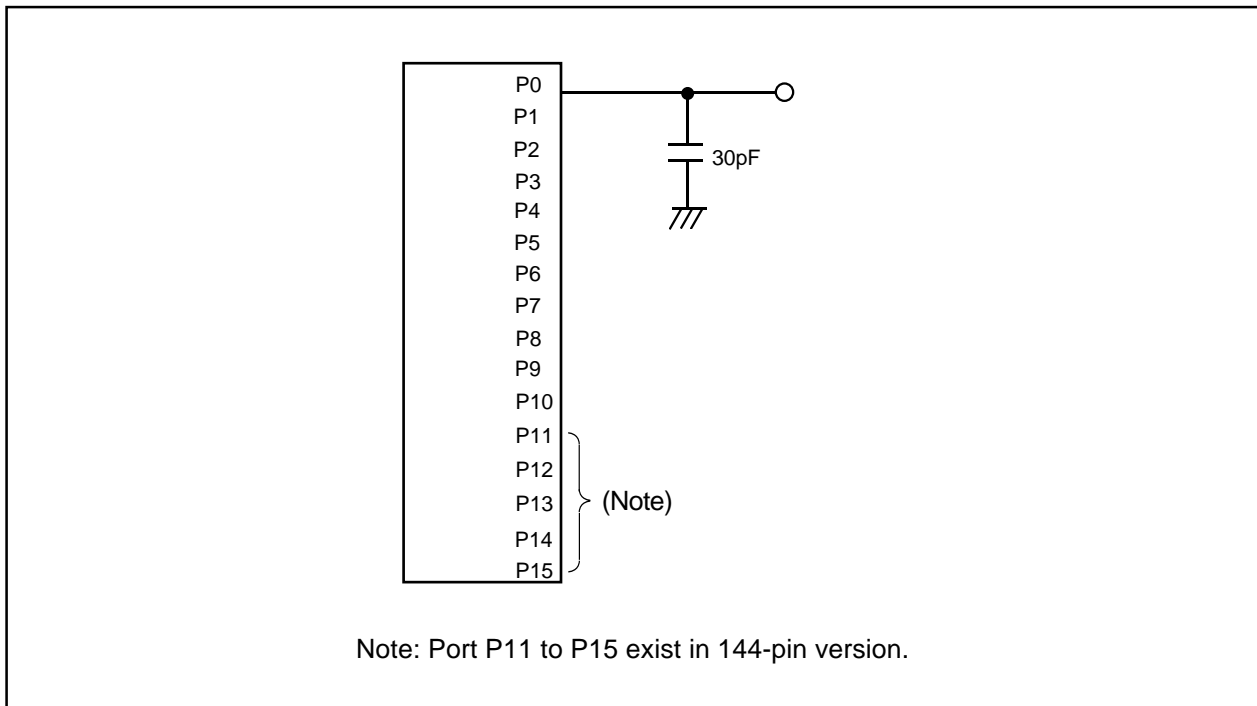


Figure 28.1 Port P0 to P15 measurement circuit

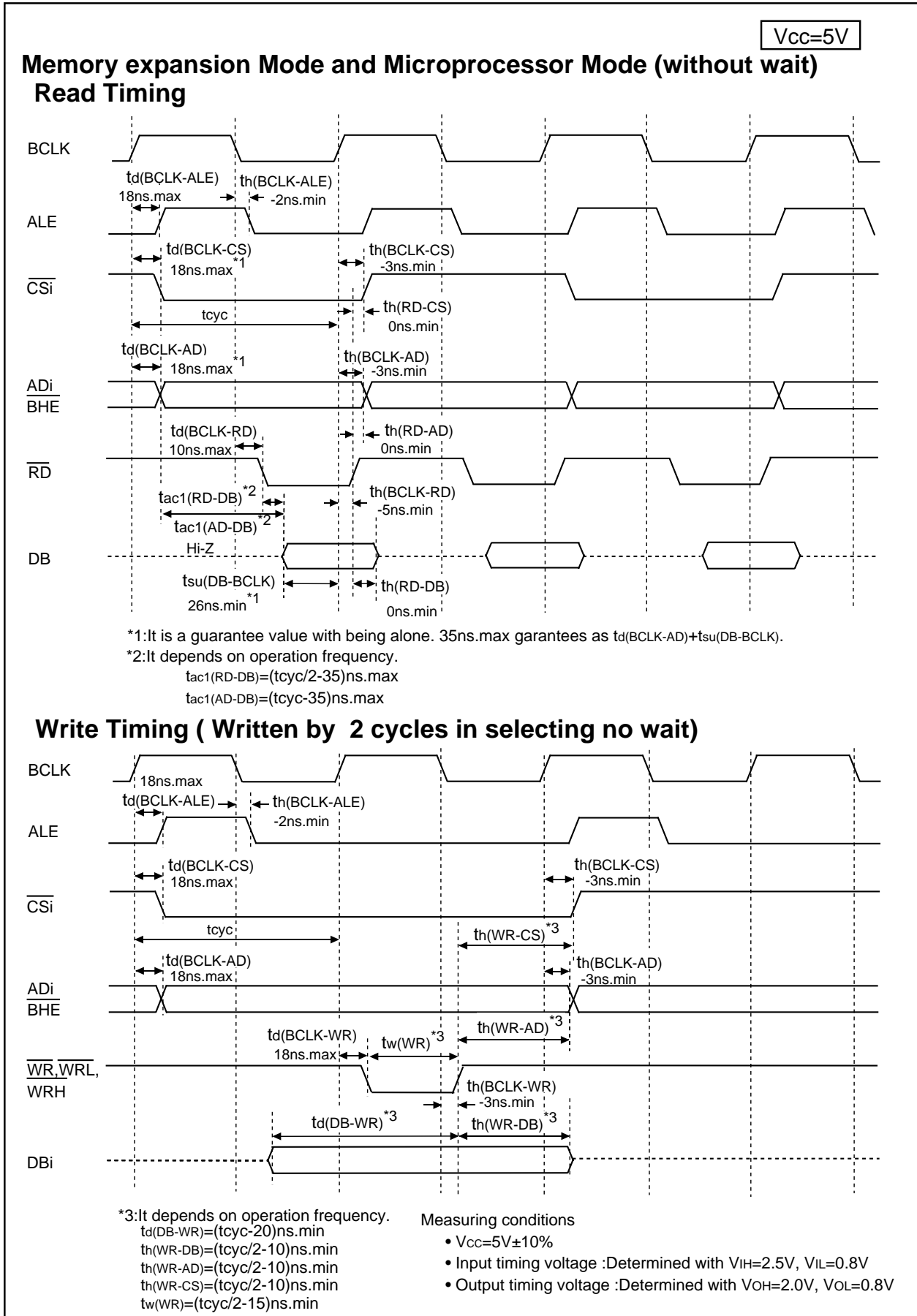


Figure 28.2 V<sub>CC</sub>=5V timing diagram (1)

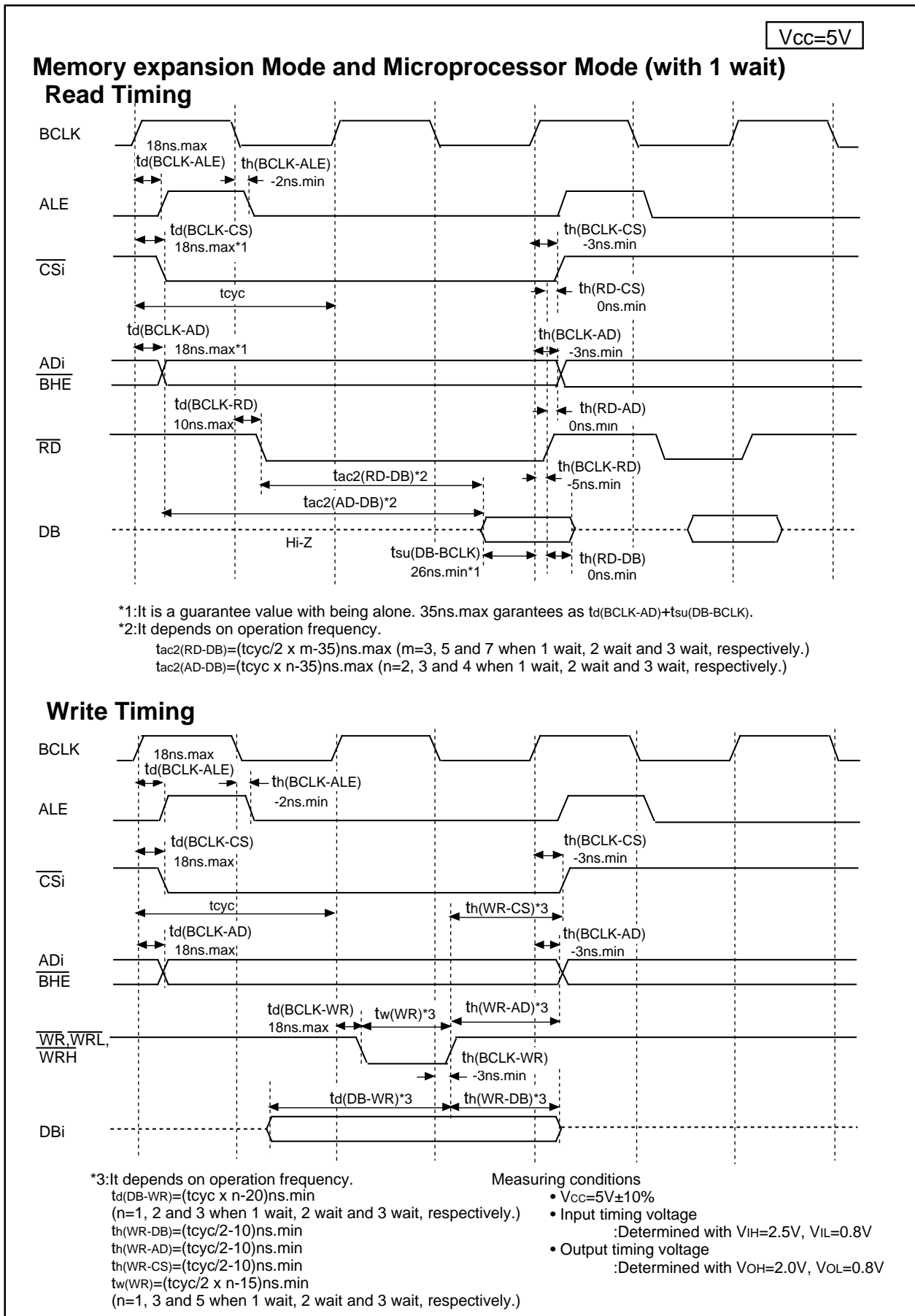


Figure 28.3 V<sub>CC</sub>=5V timing diagram (2)

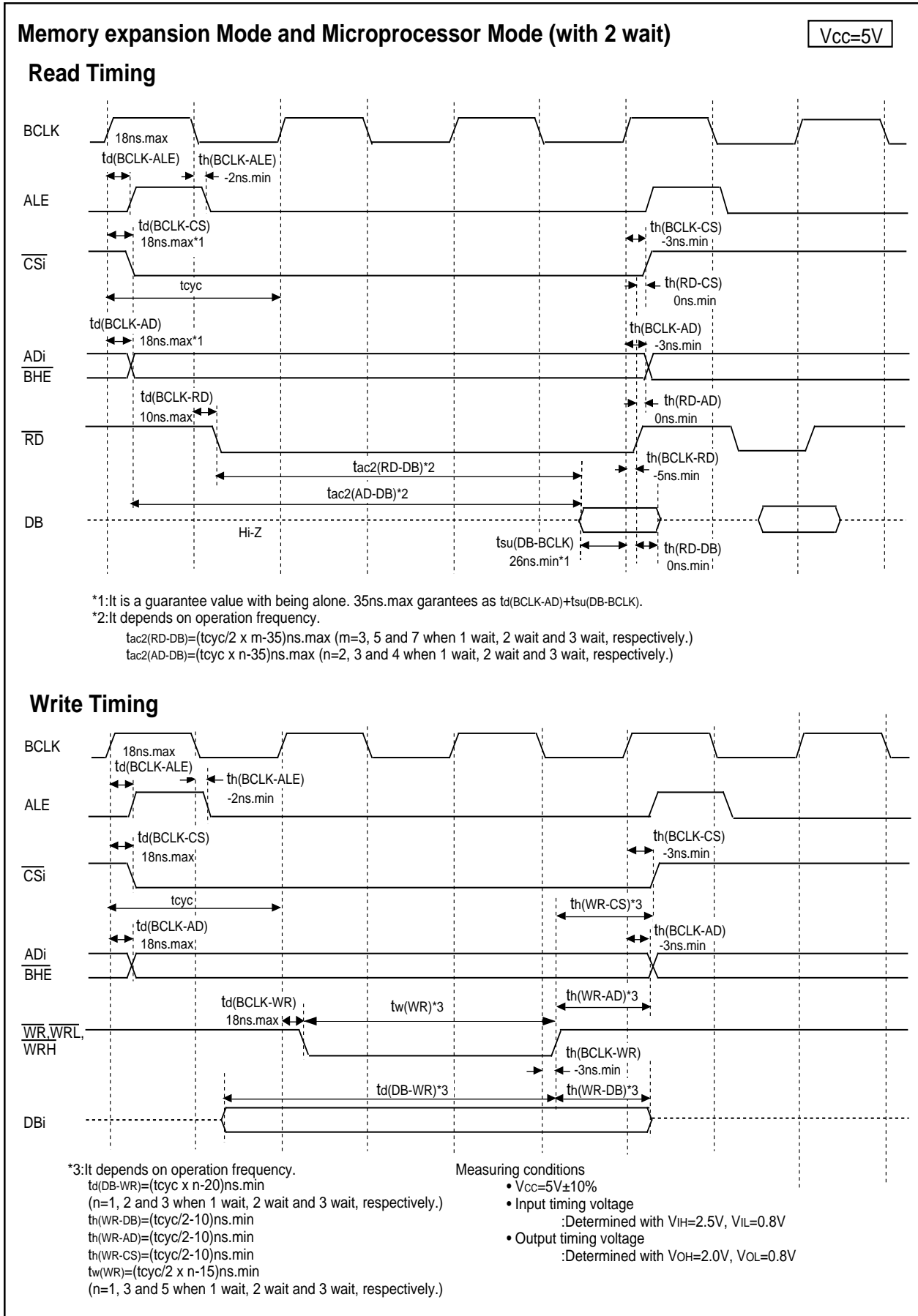


Figure 28.4 V<sub>CC</sub>=5V timing diagram (3)

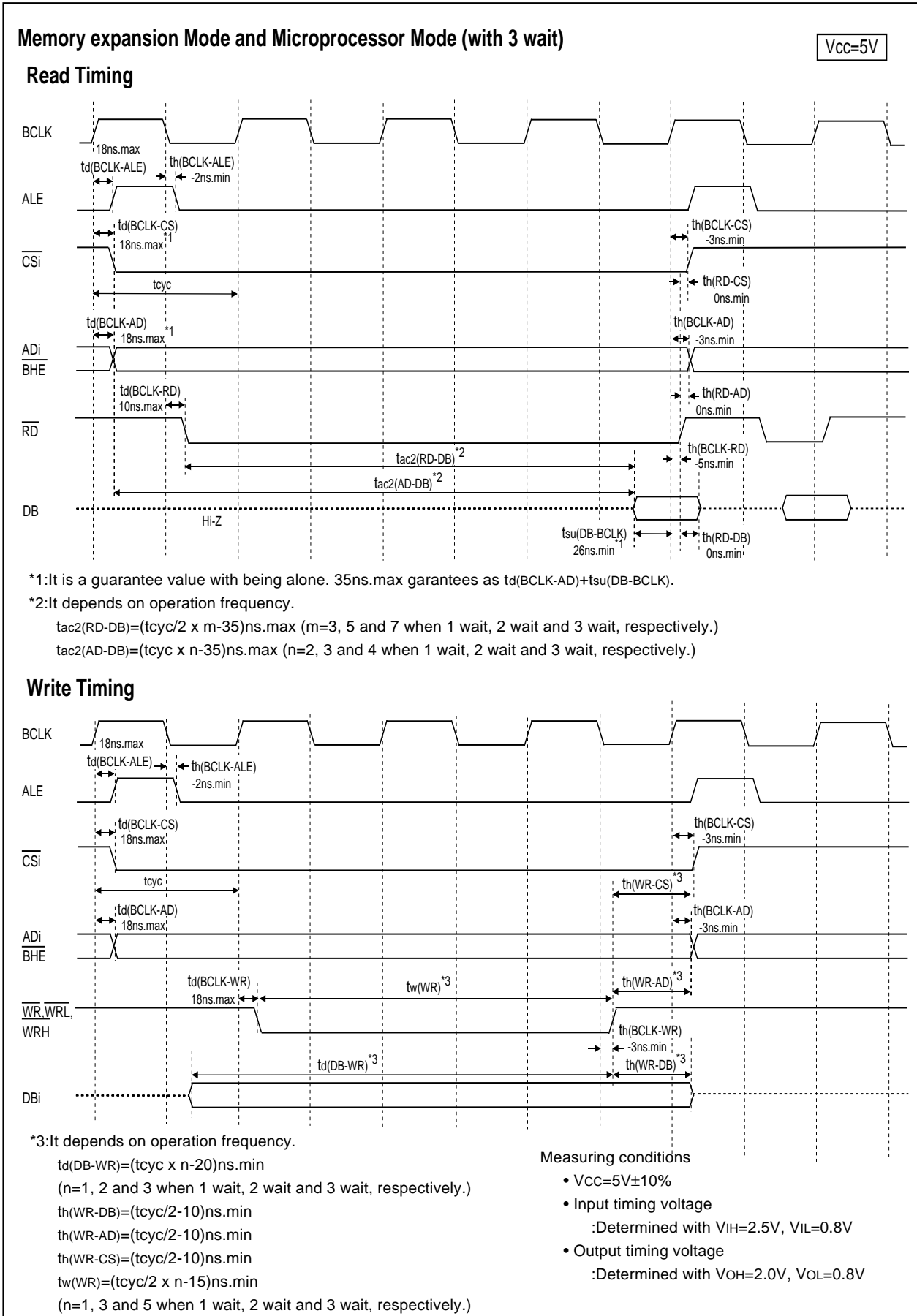


Figure 28.5 V<sub>CC</sub>=5V timing diagram (4)

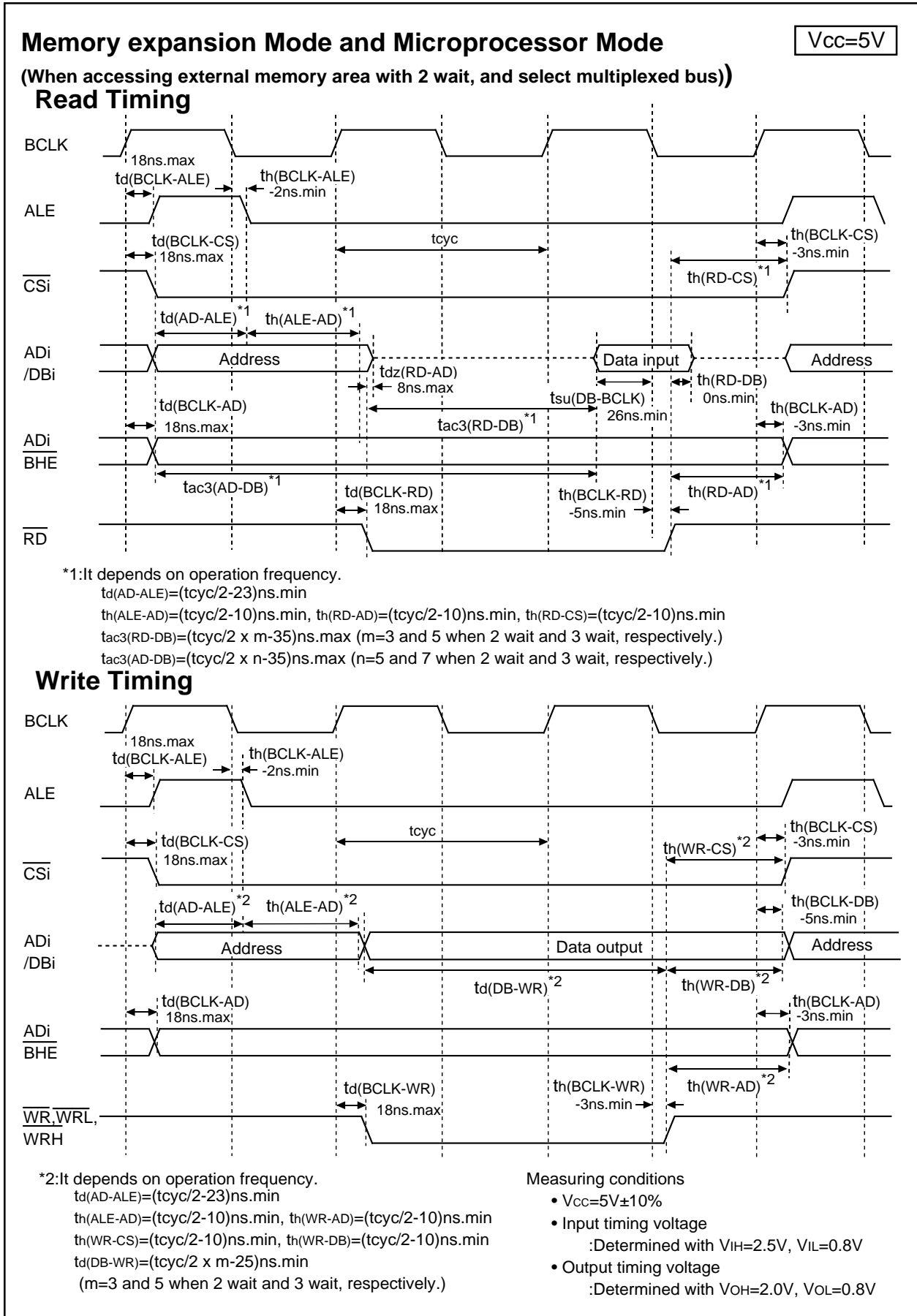


Figure 28.6 V<sub>CC</sub>=5V timing diagram (5)

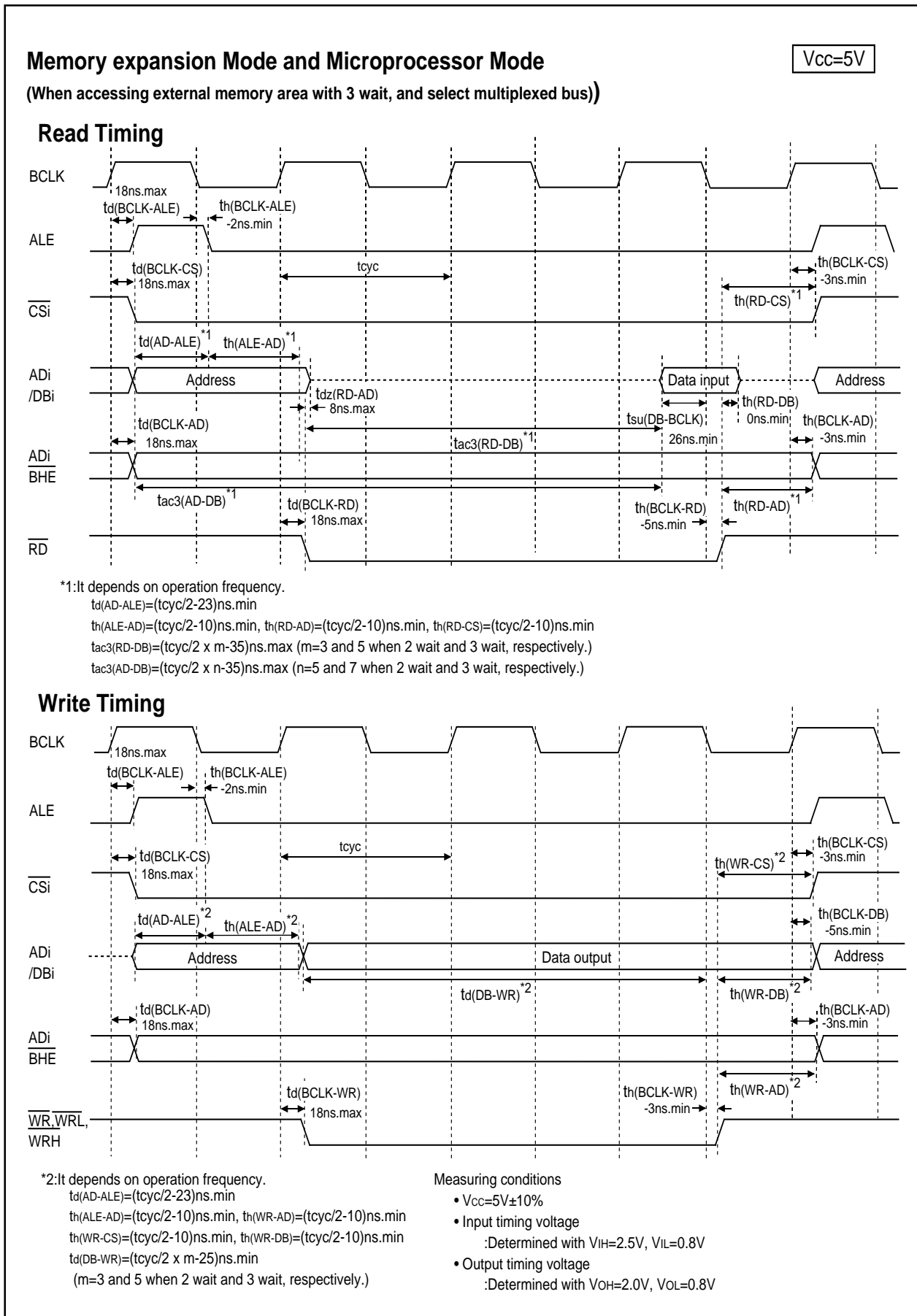
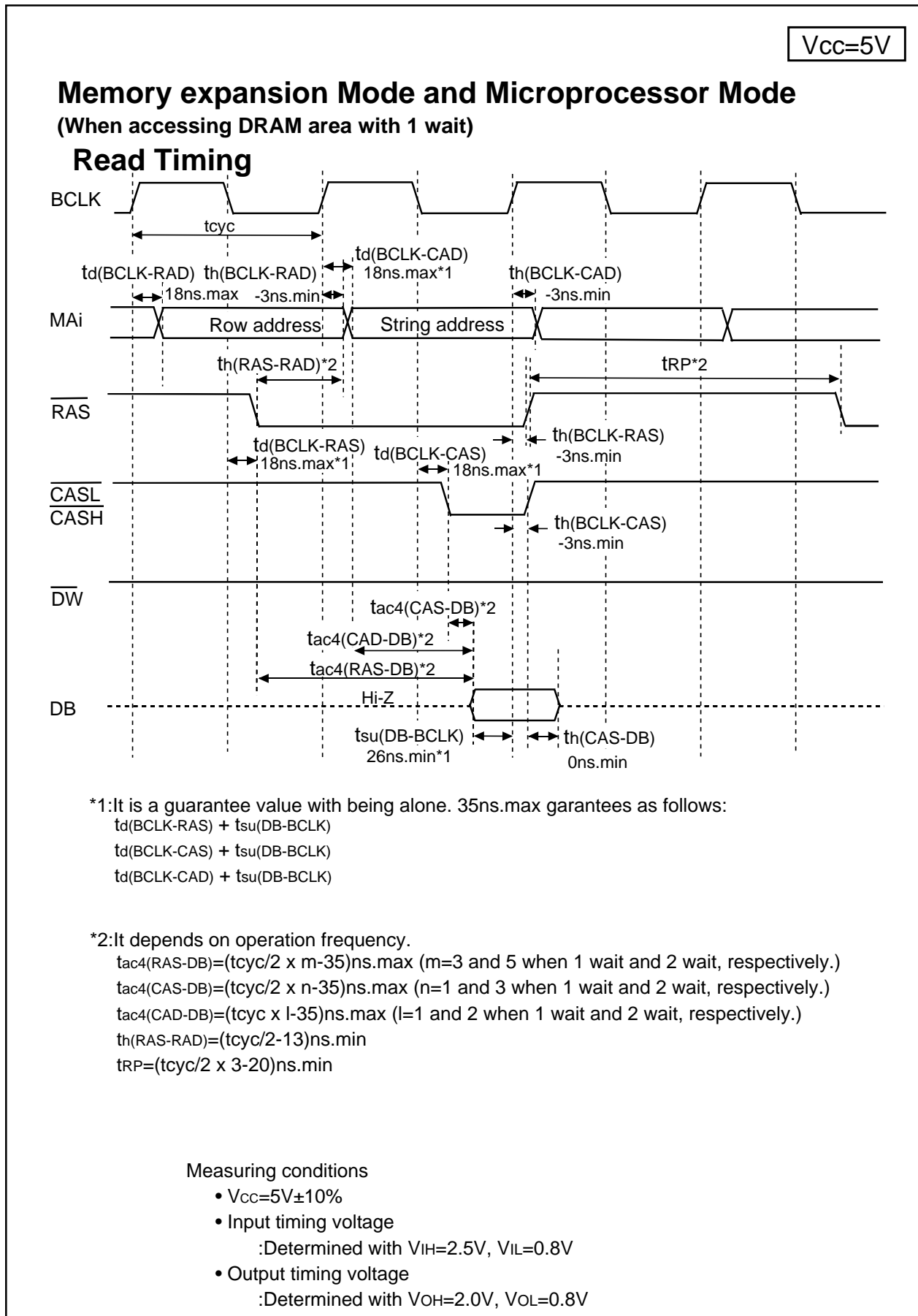


Figure 28.7 V<sub>CC</sub>=5V timing diagram (6)

Figure 28.8 V<sub>CC</sub>=5V timing diagram (7)



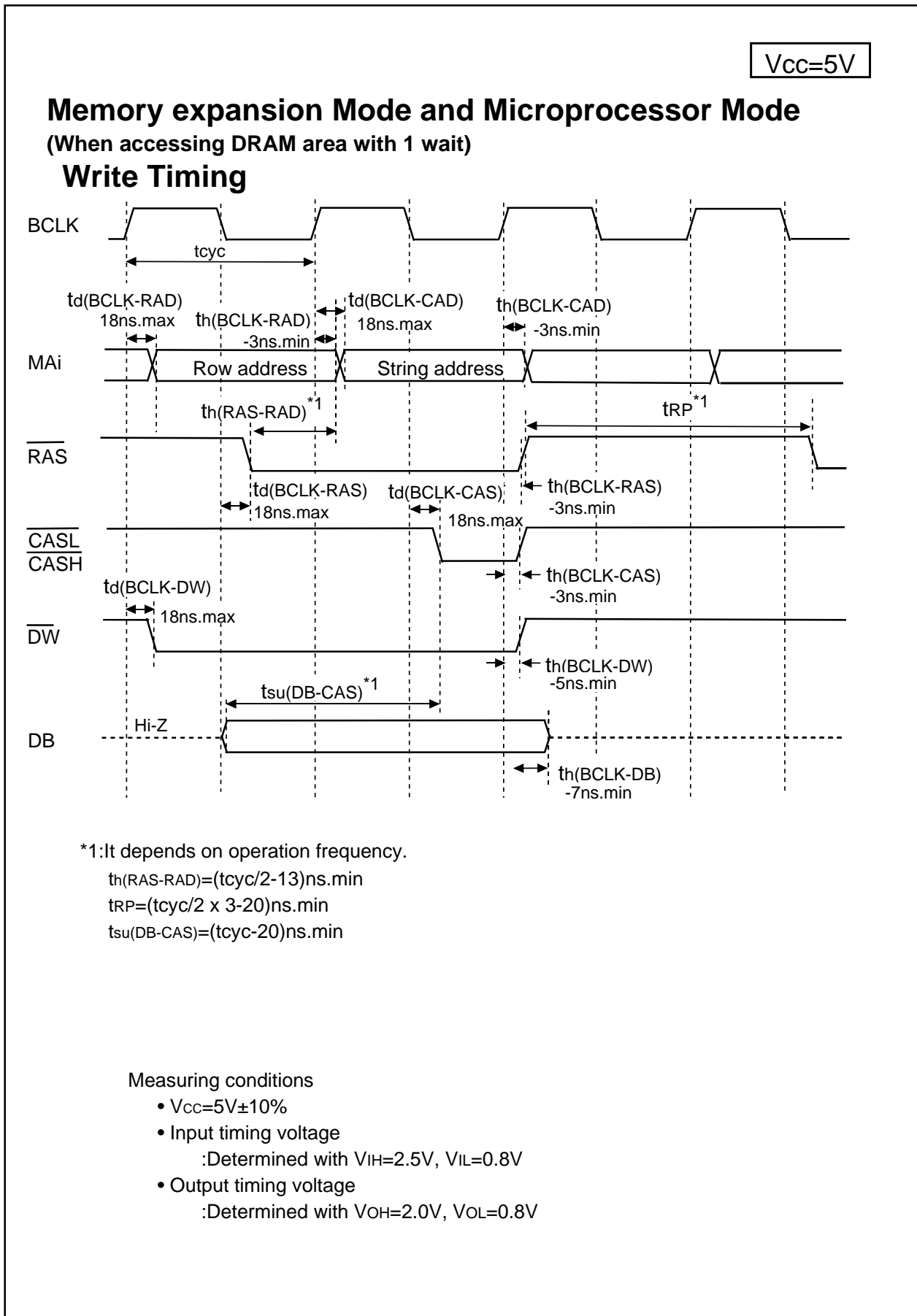


Figure 28.9 V<sub>CC</sub>=5V timing diagram (8)

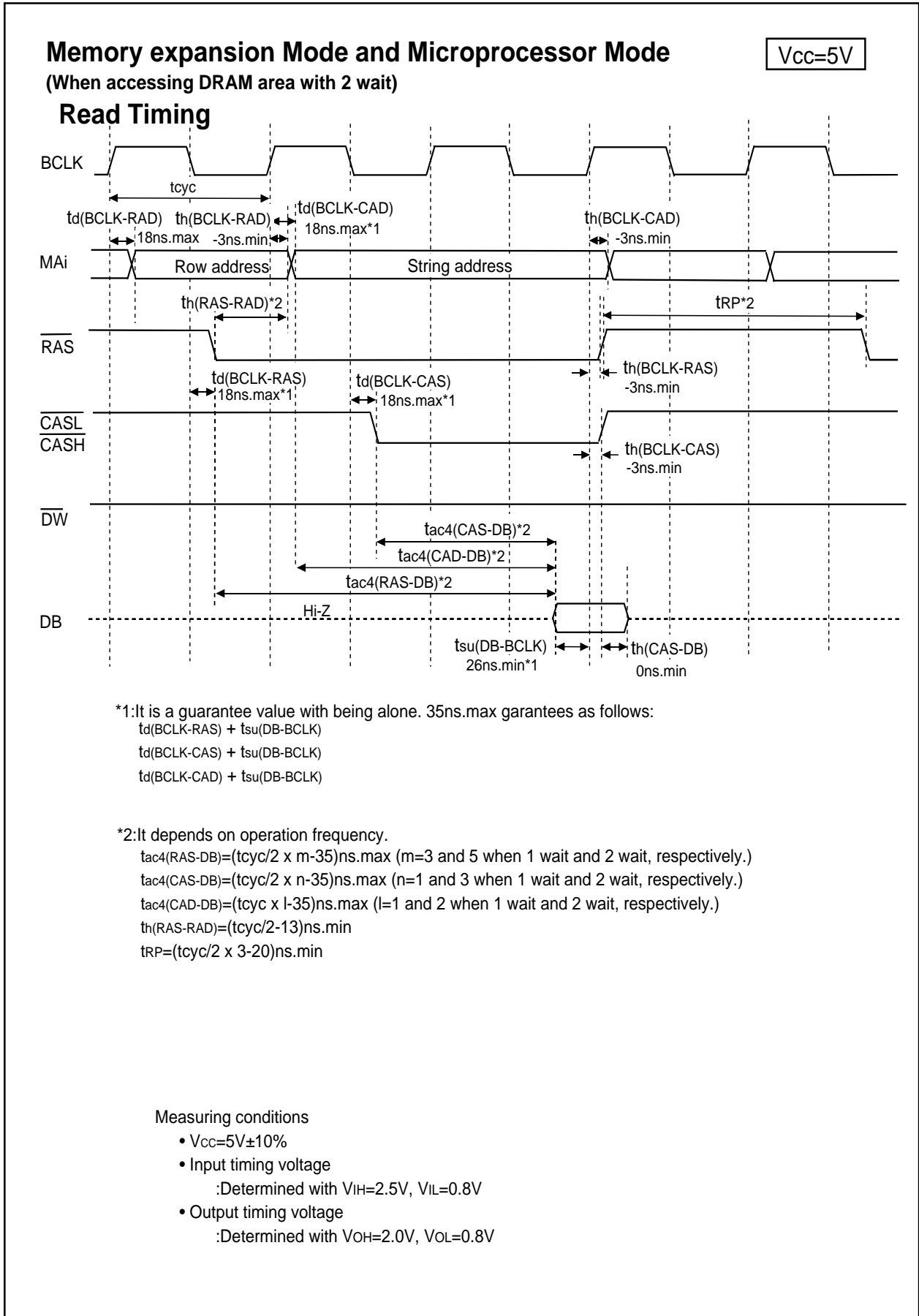


Figure 28.10 V<sub>CC</sub>=5V timing diagram (9)

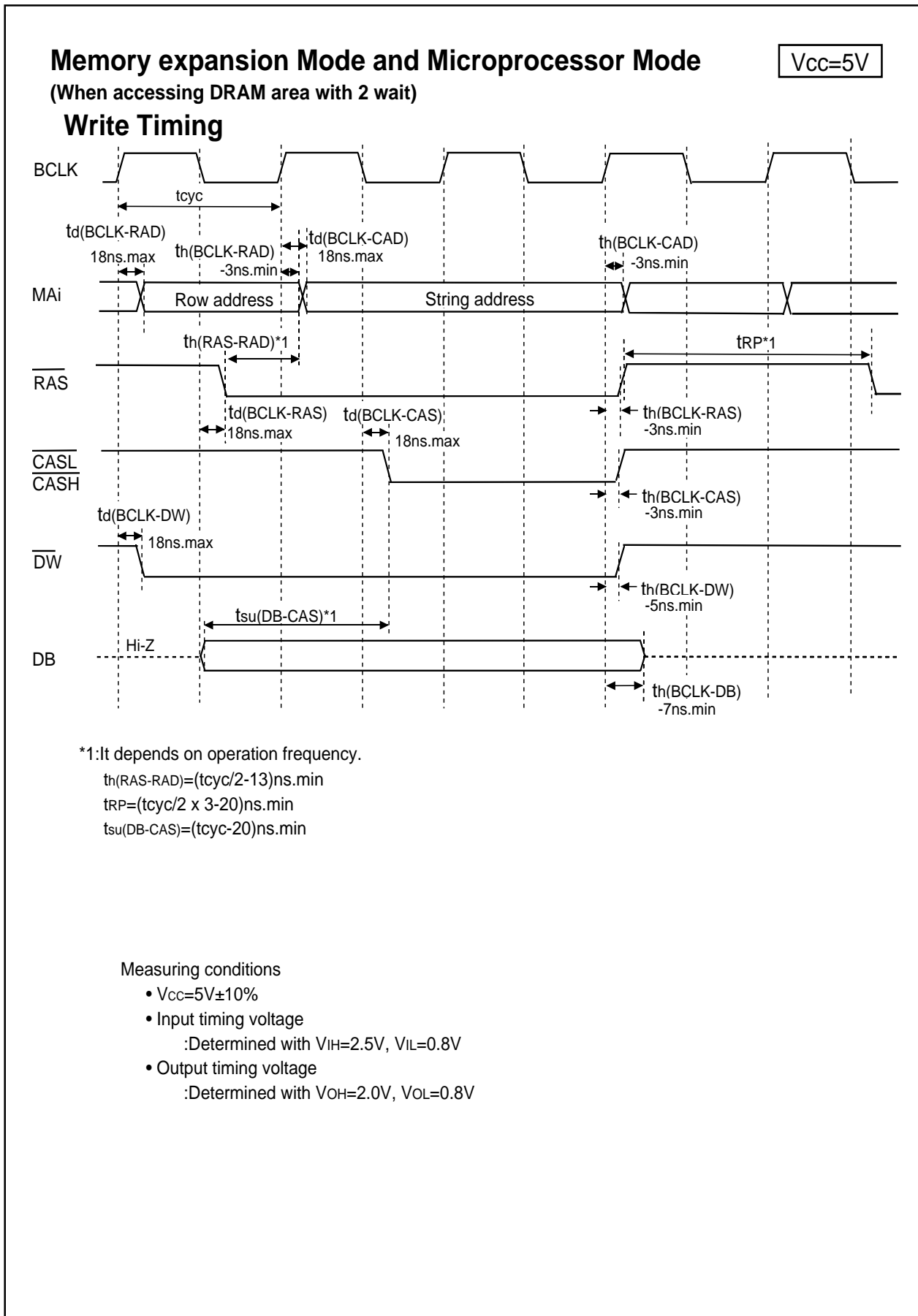


Figure 28.11 V<sub>CC</sub>=5V timing diagram (10)

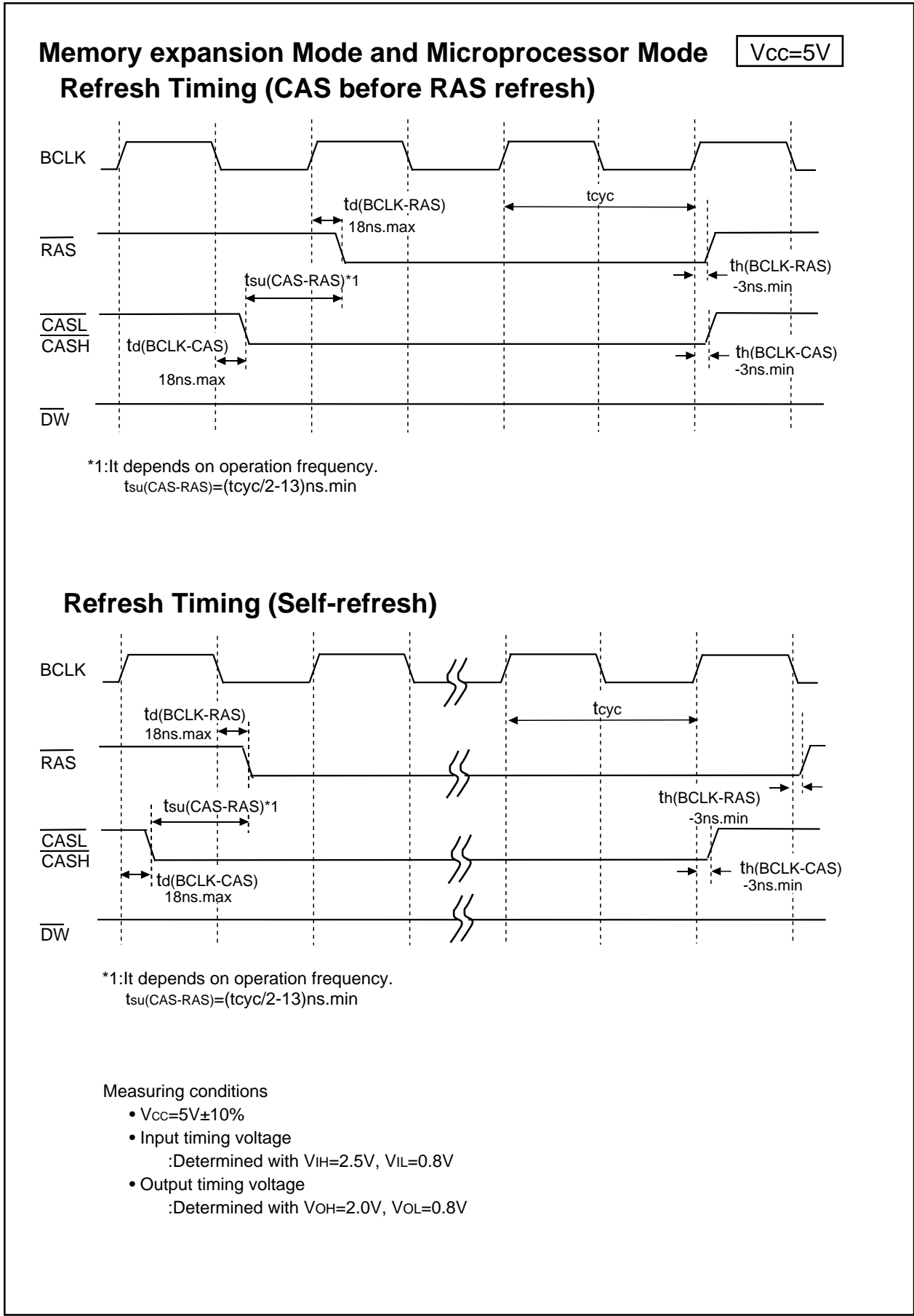


Figure 28.12 V<sub>CC</sub>=5V timing diagram (11)

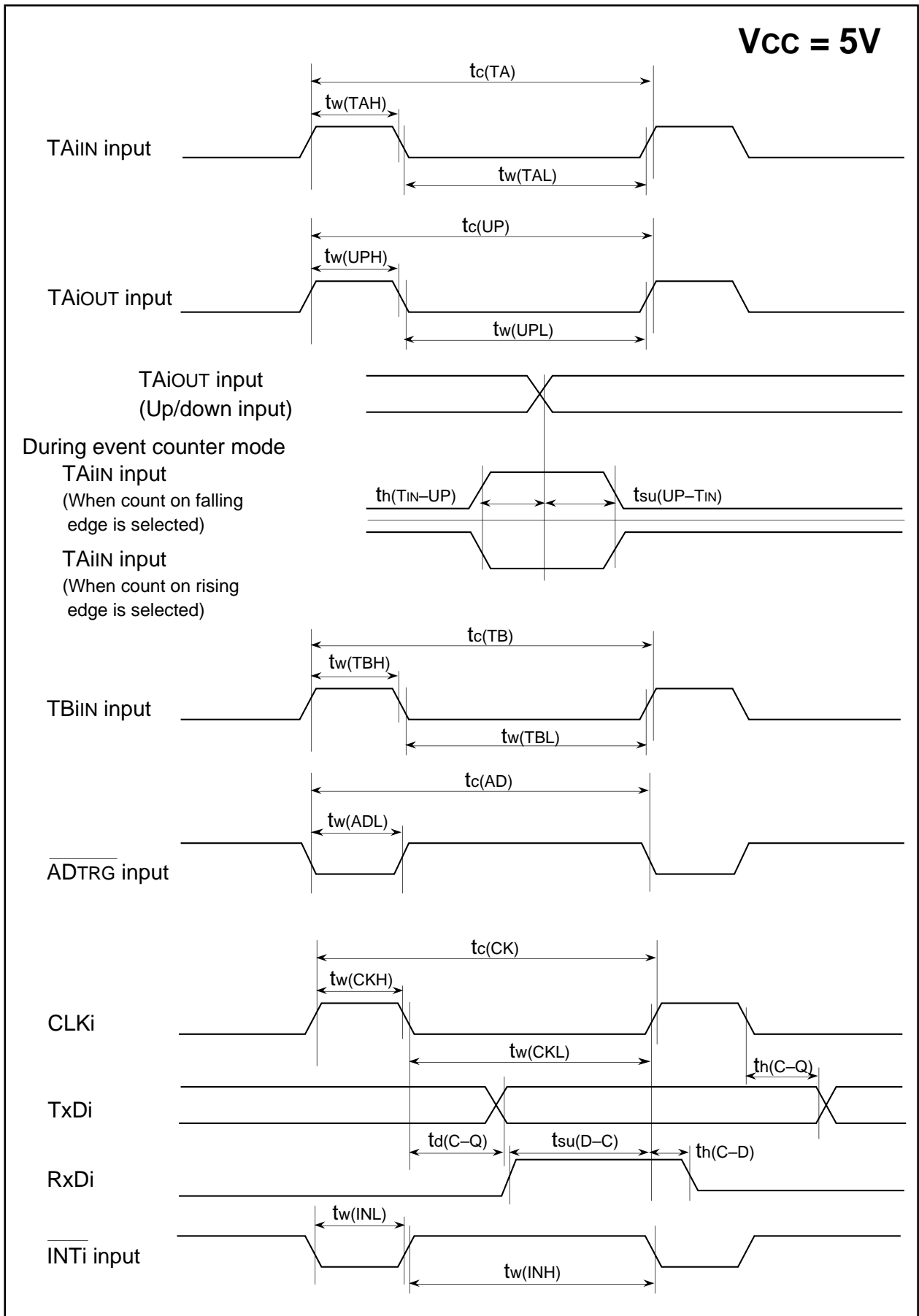


Figure 28.13 Vcc=5V timing diagram (12)

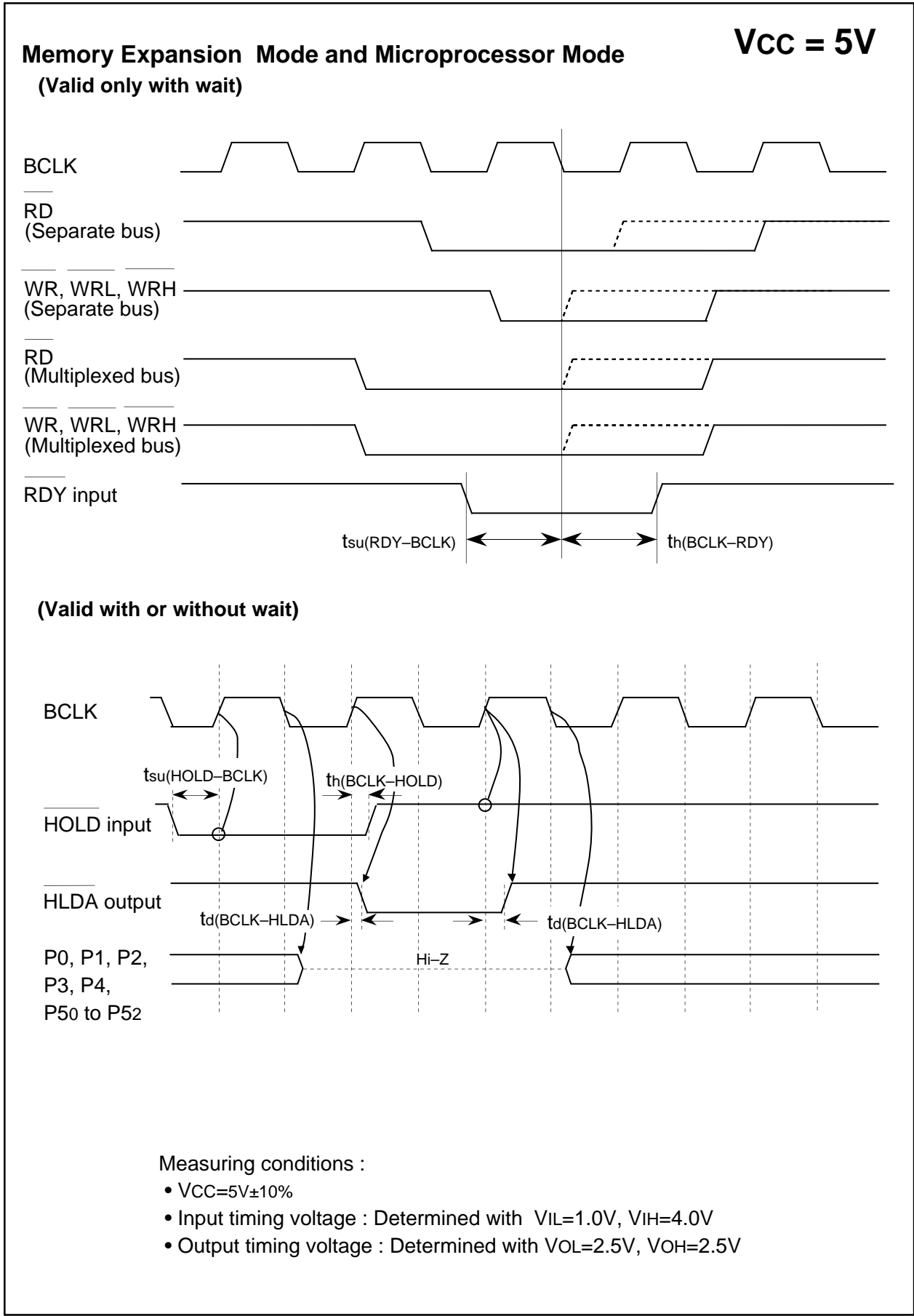


Figure 28.14 Vcc=5V timing diagram (13)

Electrical characteristics (Vcc = 3V)

VCC = 3V

Table 28.23 Electrical characteristics (referenced to VCC = 3V, VSS = 0V at Topr = 25°C, f(XIN) = 10MHz unless otherwise specified)

Symbol	Parameter		Measuring condition	Standard			Unit
				Min	Typ.	Max.	
VOH	HIGH output voltage	P00-P07,P10-P17,P20-P27, P30-P37,P40-P47,P50-P57, P60-P67,P72-P77,P80-P84, P86,P87,P90-P97,P100-P107, P110-P114, P120-P127,P130-P137, P140-P146, P150-P157 (Note 1)	IOH= - 1mA , Vcc = 3.0V	2.5			V
VOH	HIGH output voltage	XOUT	HIGHPOWER	IOH= - 0.1 mA , Vcc = 3.0V	2.5		V
			LOWPOWER	IOH= - 50 μA , Vcc = 3.0V	2.5		V
	HIGH output voltage	XCOUT	HIGHPOWER	With no load applied , Vcc = 3.0V		3.0	V
			LOWPOWER	With no load applied , Vcc = 3.0V		1.6	V
VOL	LOW output voltage	P00-P07,P10-P17,P20-P27, P30-P37,P40-P47,P50-P57, P60-P67,P70-P77,P80-P84, P86,P87,P90-P97,P100-P107, P110-P114, P120-P127,P130-P137, P140-P146, P150-P157 (Note 1)	IOl=1mA , Vcc = 3.0V			0.5	V
VOL	LOW output voltage	XOUT	HIGHPOWER	IOl=0.1mA , Vcc = 3.0V		0.5	V
			LOWPOWER	IOl=50μA , Vcc = 3.0V		0.5	V
	LOW output voltage	XCOUT	HIGHPOWER	With no load applied , Vcc = 3.0V		0	V
			LOWPOWER	With no load applied , Vcc = 3.0V		0	V
VT+-VT-	Hysteresis	HOLD, RDY, TA0IN-TA4IN, TB0IN-TB2IN, INT0-INT5, ADTRG, CTS0-CTS4,CLK0-CLK4,TA2OUT-TA4OUT, NMI, KI0-KI3, RxD0-RxD4, SCL2-SCL4, SDA2-SDA4	Vcc = 3.0V	0.2		1.0	V
VT+-VT-	Hysteresis	RESET	Vcc = 3.0V	0.2		1.8	V
IiH	HIGH input current	P00-P07,P10-P17,P20-P27, P30-P37,P40-P47,P50-P57, P60-P67,P70-P77,P80-P87, P90-P97,P100-P107, P110-P114, P120-P127,P130-P137, P140-P146, P150-P157 (Note 1) XIN, RESET, CNVss, BYTE	Vi=3V , Vcc = 3.0V			4.0	μA
IiL	LOW input current	P00-P07,P10-P17,P20-P27, P30-P37,P40-P47,P50-P57, P60-P67,P70-P77,P80-P87, P90-P97,P100-P107, P110-P114, P120-P127,P130-P137, P140-P146, P150-P157 (Note 1) XIN, RESET, CNVss, BYTE	Vi=0V , Vcc = 3.0V			- 4.0	μA
Rpullup	Pull-up resistance	P00-P07,P10-P17,P20-P27, P30-P37,P40-P47,P50-P57, P60-P67,P72-P77,P80-P84, P86,P87,P90-P97,P100-P107, P110-P114, P120-P127,P130-P137, P140-P146, P150-P157 (Note 1)	Vi=0V , Vcc = 3.0V	66.0	120.0	500.0	kΩ
RiXIN	Feedback resistance	XIN			3.0		MΩ
RiXCIN	Feedback resistance	XCIN			10.0		MΩ
Vram	RAM retention voltage		When clock is stopped	2.0			V
Icc	Power supply current	In single-chip mode, the output pins are open and other pins are VSS	f(XIN)=10MHz Square wave, no division	Mask ROM 128 KB version ROMless RAM 10 KB version (Note 2)	12.0	20.0	mA
				Mask ROM 256 KB version ROMless RAM 24 KB version (Note 2)	14.0	23.0	
				Flash memory version	14.0	23.0	
			f(XCIN)=32kHz Square wave	Mask ROM 128 KB version ROMless RAM 10 KB version (Note 2)	45.0		μA
				Mask ROM 256 KB version ROMless RAM 24 KB version (Note 2)	60.0		
				Flash memory version	3.5		
				Flash memory version	3.5		
			f(XCIN)=32kHz When a WAIT instruction is executed. Oscillation drive capacity is High.		3.0		μA
					1.5		
			f(XCIN)=32kHz When a WAIT instruction is executed. Oscillation drive capacity is Low.	Mask ROM 128 KB version ROMless RAM 10 KB version (Note 2)		1.0	μA
Mask ROM 256 KB version ROMless RAM 24 KB version (Note 2)		1.0					
Flash memory version		1.0					
Topr=25°C, when clock is stopped						20.0	
Topr=85°C, when clock is stopped							

Note 1: Ports P11 to P15 exist in 144-pin version.

Note 2: ROMless version exists in 144-pin version.

$$V_{CC} = 3V$$

Timing requirements (referenced to  $V_{CC} = 3V$ ,  $V_{SS} = 0V$  at  $T_{opr} = 25^{\circ}C$  unless otherwise specified)

**Table 28.24 External clock input**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t <sub>c</sub>	External clock input cycle time	100		ns
t <sub>w(H)</sub>	External clock input HIGH pulse width	40		ns
t <sub>w(L)</sub>	External clock input LOW pulse width	40		ns
t <sub>r</sub>	External clock rise time		18	ns
t <sub>f</sub>	External clock fall time		18	ns

**Table 28.25 Memory expansion and microprocessor modes**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t <sub>ac1</sub> (RD-DB)	Data input access time (RD standard, no wait)		(Note)	ns
t <sub>ac1</sub> (AD-DB)	Data input access time (AD standard, CS standard, no wait)		(Note)	ns
t <sub>ac2</sub> (RD-DB)	Data input access time (RD standard, with wait)		(Note)	ns
t <sub>ac2</sub> (AD-DB)	Data input access time (AD standard, CS standard, with wait)		(Note)	ns
t <sub>ac3</sub> (RD-DB)	Data input access time (RD standard, when accessing multiplex bus area)		(Note)	ns
t <sub>ac3</sub> (AD-DB)	Data input access time (AD standard, CS standard, when accessing multiplex bus area)		(Note)	ns
t <sub>ac4</sub> (RAS-DB)	Data input access time (RAS standard, DRAM access)		(Note)	ns
t <sub>ac4</sub> (CAS-DB)	Data input access time (CAS standard, DRAM access)		(Note)	ns
t <sub>ac4</sub> (CAD-DB)	Data input access time (CAD standard, DRAM access)		(Note)	ns
t <sub>su</sub> (DB-BCLK)	Data input setup time	40		ns
t <sub>su</sub> (RDY-BCLK)	RDY input setup time	60		ns
t <sub>su</sub> (HOLD-BCLK)	HOLD input setup time	80		ns
t <sub>h</sub> (RD-DB)	Data input hold time	0		ns
t <sub>h</sub> (CAS-DB)	Data input hold time	0		ns
t <sub>h</sub> (BCLK-RDY)	RDY input hold time	0		ns
t <sub>h</sub> (BCLK-HOLD)	HOLD input hold time	0		ns
t <sub>d</sub> (BCLK-HLDA)	HLDA output delay time		100	ns

Note: Calculated according to the BCLK frequency as follows:

Note that inserting wait or using lower operation frequency f(BCLK) is needed when calculated value is negative.

$$t_{ac1}(RD-DB) = \frac{10^9}{f(BCLK) \times 2} - 42 \quad [ns]$$

$$t_{ac1}(AD-DB) = \frac{10^9}{f(BCLK)} - 55 \quad [ns]$$

$$t_{ac2}(RD-DB) = \frac{10^9 \times m}{f(BCLK) \times 2} - 42 \quad [ns] \quad (m=3, 5 \text{ and } 7 \text{ when } 1 \text{ wait, } 2 \text{ wait and } 3 \text{ wait, respectively})$$

$$t_{ac2}(AD-DB) = \frac{10^9 \times n}{f(BCLK)} - 55 \quad [ns] \quad (n=2, 3 \text{ and } 4 \text{ when } 1 \text{ wait, } 2 \text{ wait and } 3 \text{ wait, respectively})$$

$$t_{ac3}(RD-DB) = \frac{10^9 \times m}{f(BCLK) \times 2} - 55 \quad [ns] \quad (m=3 \text{ and } 5 \text{ when } 2 \text{ wait and } 3 \text{ wait, respectively})$$

$$t_{ac3}(AD-DB) = \frac{10^9 \times n}{f(BCLK) \times 2} - 55 \quad [ns] \quad (n=5 \text{ and } 7 \text{ when } 2 \text{ wait and } 3 \text{ wait, respectively})$$

$$t_{ac4}(RAS-DB) = \frac{10^9 \times m}{f(BCLK) \times 2} - 55 \quad [ns] \quad (m=3 \text{ and } 5 \text{ when } 1 \text{ wait and } 2 \text{ wait, respectively})$$

$$t_{ac4}(CAS-DB) = \frac{10^9 \times n}{f(BCLK) \times 2} - 55 \quad [ns] \quad (n=1 \text{ and } 3 \text{ when } 1 \text{ wait and } 2 \text{ wait, respectively})$$

$$t_{ac4}(CAD-DB) = \frac{10^9 \times l}{f(BCLK)} - 55 \quad [ns] \quad (l=1 \text{ and } 2 \text{ when } 1 \text{ wait and } 2 \text{ wait, respectively})$$



$$V_{CC} = 3V$$

Timing requirements (referenced to  $V_{CC} = 3V$ ,  $V_{SS} = 0V$  at  $T_{op} = 25^{\circ}C$  unless otherwise specified)

**Table 28.26 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 28.27 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 28.28 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 28.29 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 28.30 Timer A input (up/down 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

$$V_{CC} = 3V$$

Timing requirements (referenced to  $V_{CC} = 3V$ ,  $V_{SS} = 0V$  at  $T_{op} = 25^{\circ}C$  unless otherwise specified)

**Table 28.31 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)	160		ns
$t_{w(TBL)}$	TBiIn input LOW pulse width (counted on both edges)	160		ns

**Table 28.32 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 28.33 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 28.34 A/D trigger input**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(AD)}$	ADTRG input cycle time (trigger able minimum)	1500		ns
$t_{w(ADL)}$	ADTRG input LOW pulse width	200		ns

**Table 28.35 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	50		ns
$t_{h(C-D)}$	RxDi input hold time	90		ns

**Table 28.36 External interrupt  $\overline{INT}_i$  inputs**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{w(INH)}$	$\overline{INT}_i$ input HIGH pulse width	380		ns
$t_{w(INL)}$	$\overline{INT}_i$ input LOW pulse width	380		ns

$$V_{CC} = 3V$$

Switching characteristics (referenced to  $V_{CC} = 3V$ ,  $V_{SS} = 0V$  at  $T_{opr} = 25^{\circ}C$  unless otherwise specified)

Table 28.37 Memory expansion and microprocessor modes (with no wait)

Symbol	Parameter	Measuring condition	Standard		Unit
			Min.	Max.	
$t_d(\text{BCLK-AD})$	Address output delay time	Figure 28.1		25	ns
$t_h(\text{BCLK-AD})$	Address output hold time (BCLK standard)		0		ns
$t_h(\text{RD-AD})$	Address output hold time (RD standard)		0		ns
$t_h(\text{WR-AD})$	Address output hold time (WR standard)		(Note 1)		ns
$t_d(\text{BCLK-CS})$	Chip select output delay time			25	ns
$t_h(\text{BCLK-CS})$	Chip select output hold time (BCLK standard)		0		ns
$t_h(\text{RD-CS})$	Chip select output hold time (RD standard)		0		ns
$t_h(\text{WR-CS})$	Chip select output hold time (WR standard)		(Note 1)		ns
$t_d(\text{BCLK-ALE})$	ALE signal output delay time			25	ns
$t_h(\text{BCLK-ALE})$	ALE signal output hold time		- 2		ns
$t_d(\text{BCLK-RD})$	RD signal output delay time			10	ns
$t_h(\text{BCLK-RD})$	RD signal output hold time		- 3		ns
$t_d(\text{BCLK-WR})$	WR signal output delay time			25	ns
$t_h(\text{BCLK-WR})$	WR signal output hold time		0		ns
$t_d(\text{DB-WR})$	Data output delay time (WR standard)		(Note 1)		ns
$t_h(\text{WR-DB})$	Data output hold time (WR standard)		(Note 1)		ns
$t_w(\text{WR})$	WR signal width		(Note 1)		ns

Note 1: Calculated according to the BCLK frequency as follows:

$$t_d(\text{DB-WR}) = \frac{10^9}{f(\text{BCLK})} - 40 \quad [\text{ns}]$$

$$t_h(\text{WR-DB}) = \frac{10^9}{f(\text{BCLK}) \times 2} - 20 \quad [\text{ns}]$$

$$t_h(\text{WR-AD}) = \frac{10^9}{f(\text{BCLK}) \times 2} - 20 \quad [\text{ns}]$$

$$t_h(\text{WR-CS}) = \frac{10^9}{f(\text{BCLK}) \times 2} - 20 \quad [\text{ns}]$$

$$t_w(\text{WR}) = \frac{10^9}{f(\text{BCLK}) \times 2} - 20 \quad [\text{ns}]$$

$$V_{CC} = 3V$$

Switching characteristics (referenced to  $V_{CC} = 3V$ ,  $V_{SS} = 0V$  at  $T_{opr} = 25^{\circ}C$  unless otherwise specified)

**Table 28.38 Memory expansion and microprocessor modes  
(with wait, accessing external memory)**

Symbol	Parameter	Measuring condition	Standard		Unit
			Min.	Max.	
$t_{d(BCLK-AD)}$	Address output delay time	Figure 28.1		25	ns
$t_{h(BCLK-AD)}$	Address output hold time (BCLK standard)		0		ns
$t_{h(RD-AD)}$	Address output hold time (RD standard)		0		ns
$t_{h(WR-AD)}$	Address output hold time (WR standard)		(Note 1)		ns
$t_{d(BCLK-CS)}$	Chip select output delay time			25	ns
$t_{h(BCLK-CS)}$	Chip select output hold time (BCLK standard)		0		ns
$t_{h(RD-CS)}$	Chip select output hold time (RD standard)		0		ns
$t_{h(WR-CS)}$	Chip select output hold time (WR standard)		(Note 1)		ns
$t_{d(BCLK-ALE)}$	ALE signal output delay time			25	ns
$t_{h(BCLK-ALE)}$	ALE signal output hold time		- 2		ns
$t_{d(BCLK-RD)}$	RD signal output delay time			10	ns
$t_{h(BCLK-RD)}$	RD signal output hold time		- 3		ns
$t_{d(BCLK-WR)}$	WR signal output delay time			25	ns
$t_{h(BCLK-WR)}$	WR signal output hold time		0		ns
$t_{d(DB-WR)}$	Data output delay time (WR standard)		(Note 1)		ns
$t_{h(WR-DB)}$	Data output hold time (WR standard)		(Note 1)		ns
$t_w(WR)$	WR signal width		(Note 1)		ns

Note 1: Calculated according to the BCLK frequency as follows:

$$t_{d(DB-WR)} = \frac{10^9 \times n}{f_{(BCLK)}} - 40 \quad [\text{ns}] \quad (n=1, 2 \text{ and } 3 \text{ when } 1 \text{ wait, } 2 \text{ wait and } 3 \text{ wait, respectively})$$

$$t_{h(WR-DB)} = \frac{10^9}{f_{(BCLK)} \times 2} - 20 \quad [\text{ns}]$$

$$t_{h(WR-AD)} = \frac{10^9}{f_{(BCLK)} \times 2} - 20 \quad [\text{ns}]$$

$$t_{h(WR-CS)} = \frac{10^9}{f_{(BCLK)} \times 2} - 20 \quad [\text{ns}]$$

$$t_w(WR) = \frac{10^9 \times n}{f_{(BCLK)} \times 2} - 20 \quad [\text{ns}] \quad (n=1, 3 \text{ and } 5 \text{ when } 1 \text{ wait, } 2 \text{ wait and } 3 \text{ wait, respectively})$$

$$V_{CC} = 3V$$

**Switching characteristics (referenced to  $V_{CC} = 3V$ ,  $V_{SS} = 0V$  at  $T_{opr} = 25^{\circ}C$  unless otherwise specified)**

**Table 28.39 Memory expansion and microprocessor modes  
(with wait, accessing external memory, multiplex bus area selected)**

Symbol	Parameter	Measuring condition	Standard		Unit
			Min.	Max.	
td(BCLK-AD)	Address output delay time	Figure 28.1		25	ns
th(BCLK-AD)	Address output hold time (BCLK standard)		0		ns
th(RD-AD)	Address output hold time (RD standard)		(Note 1)		ns
th(WR-AD)	Address output hold time (WR standard)		(Note 1)		ns
td(BCLK-CS)	Chip select output delay time			25	ns
th(BCLK-CS)	Chip select output hold time (BCLK standard)		0		ns
th(RD-CS)	Chip select output hold time (RD standard)		(Note 1)		ns
th(WR-CS)	Chip select output hold time (WR standard)		(Note 1)		ns
td(BCLK-RD)	RD signal output delay time			25	ns
th(BCLK-RD)	RD signal output hold time		- 3		ns
td(BCLK-WR)	WR signal output delay time			25	ns
th(BCLK-WR)	WR signal output hold time		0		ns
td(DB-WR)	Data output delay time (WR standard)		(Note 1)		ns
th(WR-DB)	Data output hold time (WR standard)		(Note 1)		ns
td(BCLK-ALE)	ALE signal output delay time (BCLK standard)			25	ns
th(BCLK-ALE)	ALE signal output hold time (BCLK standard)		- 2		ns
td(AD-ALE)	ALE signal output delay time (address standard)		(Note 1)		ns
th(ALE-AD)	ALE signal output hold time (address standard)		(Note 1)		ns
tdz(RD-AD)	Address output flowing start time			8	ns
th(BCLK-DB)	DB signal output hold time (BCLK standard)		0		ns

Note 1: Calculated according to the BCLK frequency as follows:

$$th(RD - AD) = \frac{10^9}{f(BCLK) \times 2} - 20 \quad [ns]$$

$$th(WR - AD) = \frac{10^9}{f(BCLK) \times 2} - 20 \quad [ns]$$

$$th(RD - CS) = \frac{10^9}{f(BCLK) \times 2} - 20 \quad [ns]$$

$$th(WR - CS) = \frac{10^9}{f(BCLK) \times 2} - 20 \quad [ns]$$

$$td(DB - WR) = \frac{10^9 \times m}{f(BCLK) \times 2} - 40 \quad [ns] \quad (m=3 \text{ and } 5 \text{ when } 2 \text{ wait and } 3 \text{ wait, respectively})$$

$$th(WR - DB) = \frac{10^9}{f(BCLK) \times 2} - 20 \quad [ns]$$

$$td(AD - ALE) = \frac{10^9}{f(BCLK) \times 2} - 27 \quad [ns]$$

$$th(ALE - AD) = \frac{10^9}{f(BCLK) \times 2} - 20 \quad [ns]$$

$$V_{CC} = 3V$$

Switching characteristics (referenced to  $V_{CC} = 3V$ ,  $V_{SS} = 0V$  at  $T_{opr} = 25^{\circ}C$  unless otherwise specified)

**Table 28.40 Memory expansion and microprocessor modes  
(with wait, accessing external memory, DRAM area selected)**

Symbol	Parameter	Measuring condition	Standard		Unit
			Min.	Max.	
$t_{d(BCLK-RAD)}$	Row address output delay time	Figure 28.1		25	ns
$t_{h(BCLK-RAD)}$	Row address output hold time (BCLK standard)		0		ns
$t_{d(BCLK-CAD)}$	String address output delay time			25	ns
$t_{h(BCLK-CAD)}$	String address output hold time (BCLK standard)		0		ns
$t_{h(RAS-RAD)}$	Row address output hold time after RAS output		(Note 1)		ns
$t_{d(BCLK-RAS)}$	RAS output delay time (BCLK standard)			25	ns
$t_{h(BCLK-RAS)}$	RAS output hold time (BCLK standard)		0		ns
$t_{RP}$	RAS "H" hold time		(Note 1)		ns
$t_{d(BCLK-CAS)}$	CAS output delay time (BCLK standard)			25	ns
$t_{h(BCLK-CAS)}$	CAS output hold time (BCLK standard)		0		ns
$t_{d(BCLK-DW)}$	Data output delay time (BCLK standard)			25	ns
$t_{h(BCLK-DW)}$	Data output hold time (BCLK standard)		- 3		ns
$t_{su(DB-CAS)}$	CAS after DB output setup time		(Note 1)		ns
$t_{h(BCLK-DB)}$	DB signal output hold time (BCLK standard)		- 7		ns
$t_{su(CAS-RAS)}$	CAS before RAS setup time (refresh)		(Note 1)		ns

Note 1: Calculated according to the BCLK frequency as follows:

$$t_{h(RAS - RAD)} = \frac{10^9}{f_{(BCLK)} \times 2} - 25 \quad [ns]$$

$$t_{RP} = \frac{10^9 \times 3}{f_{(BCLK)} \times 2} - 40 \quad [ns]$$

$$t_{su(DB - CAS)} = \frac{10^9}{f_{(BCLK)}} - 40 \quad [ns]$$

$$t_{su(CAS - RAS)} = \frac{10^9}{f_{(BCLK)} \times 2} - 25 \quad [ns]$$

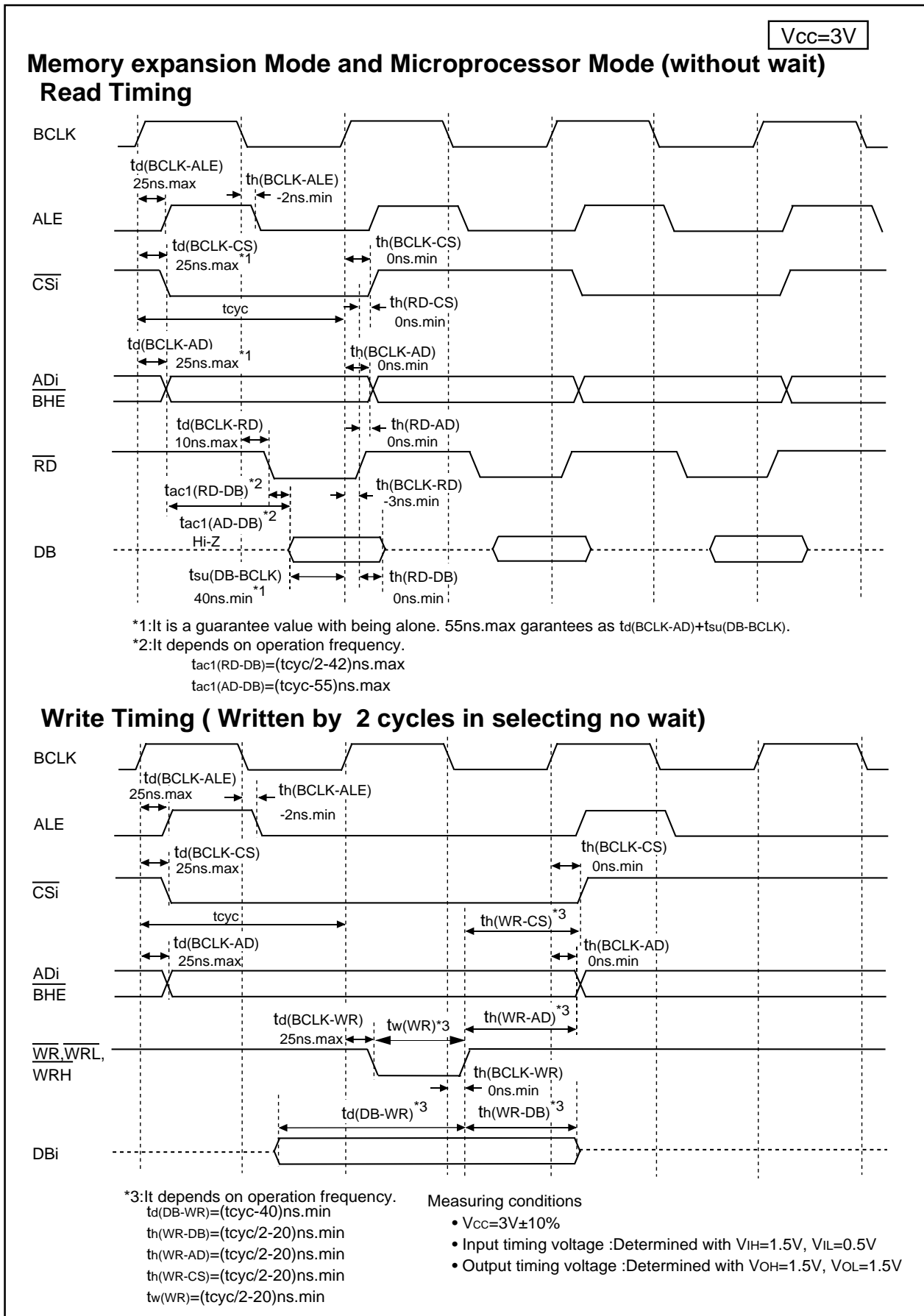


Figure 28.15 V<sub>CC</sub>=3V timing diagram (1)

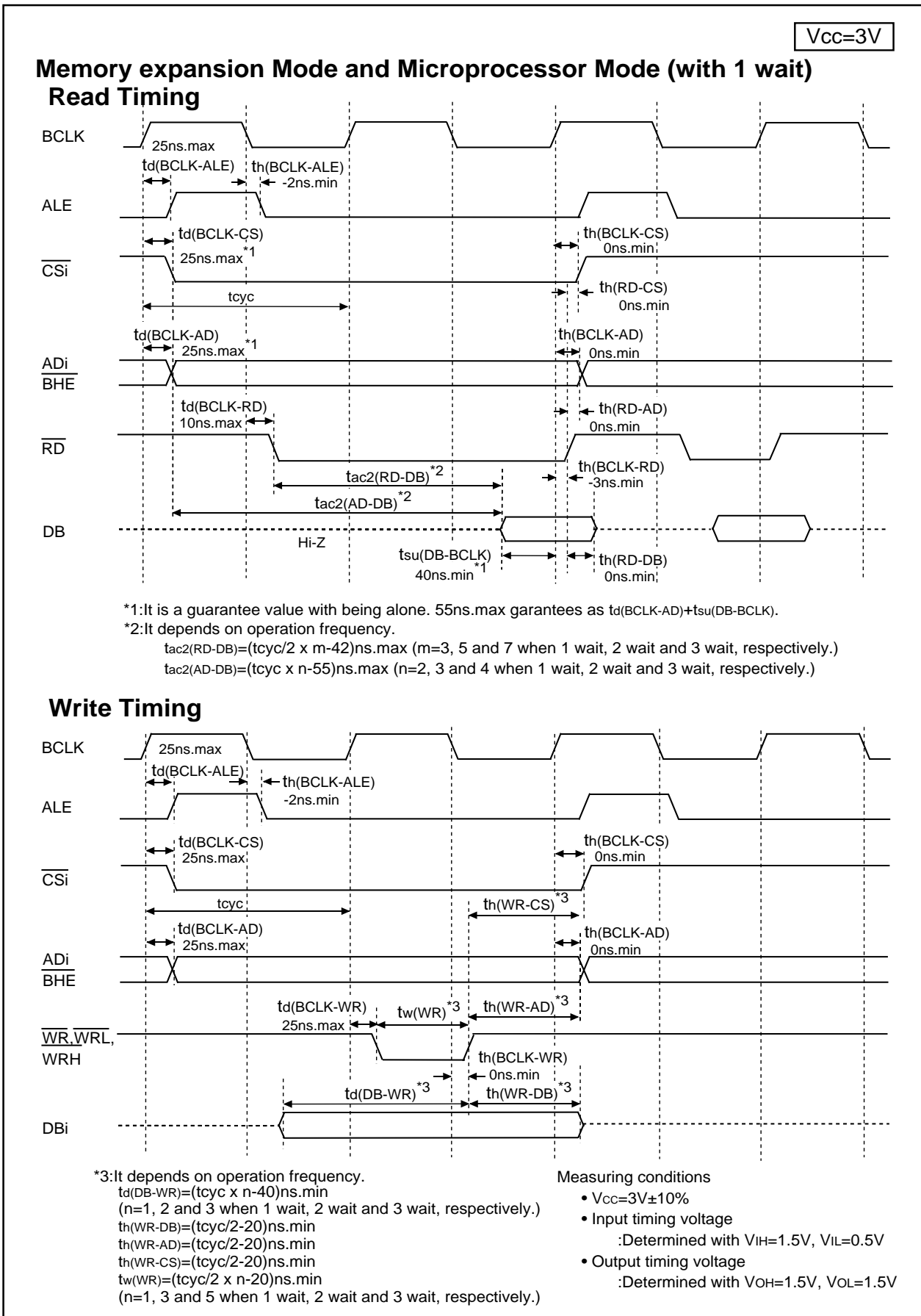


Figure 28.16 V<sub>CC</sub>=3V timing diagram (2)



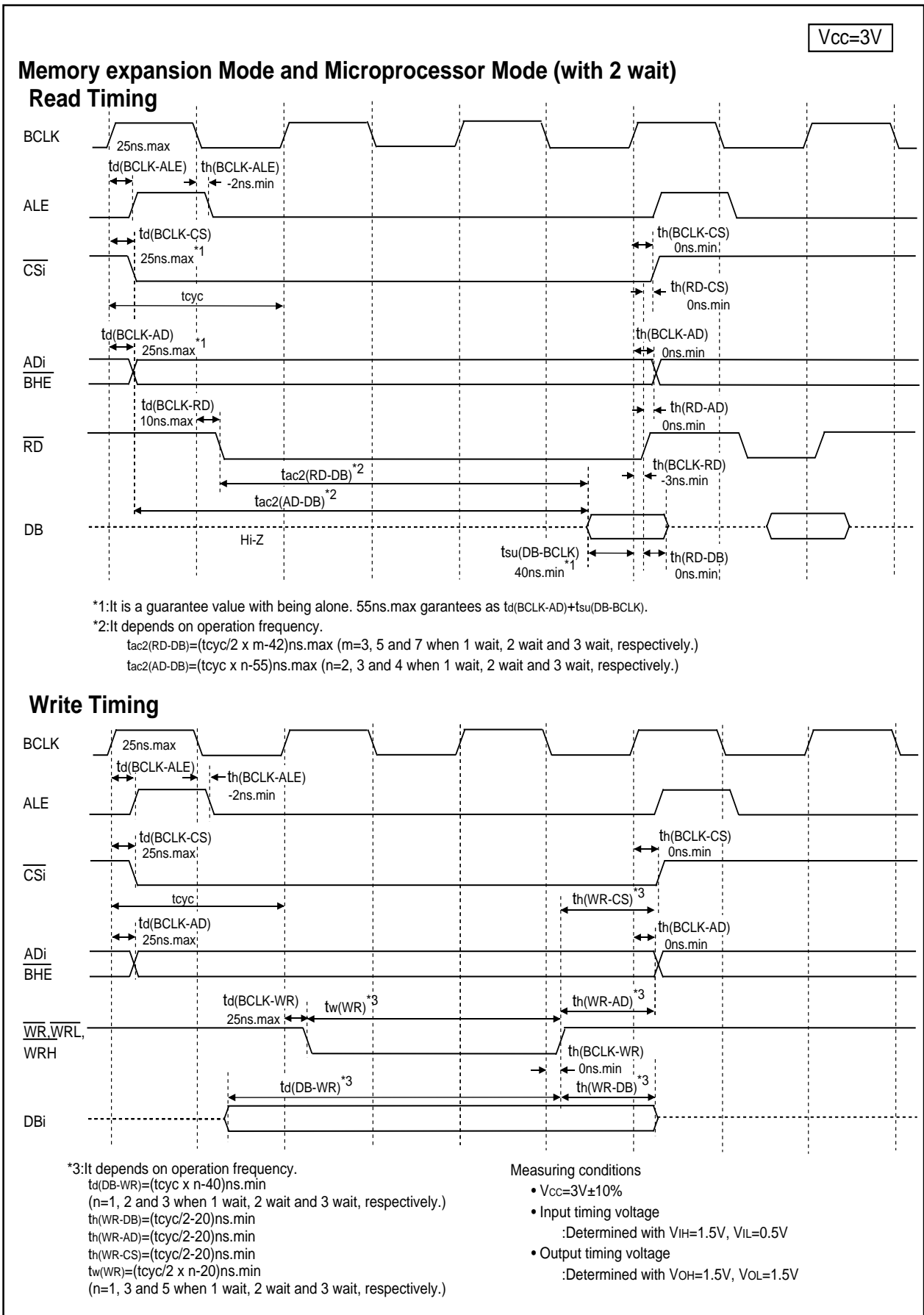


Figure 28.17 V<sub>CC</sub>=3V timing diagram (3)

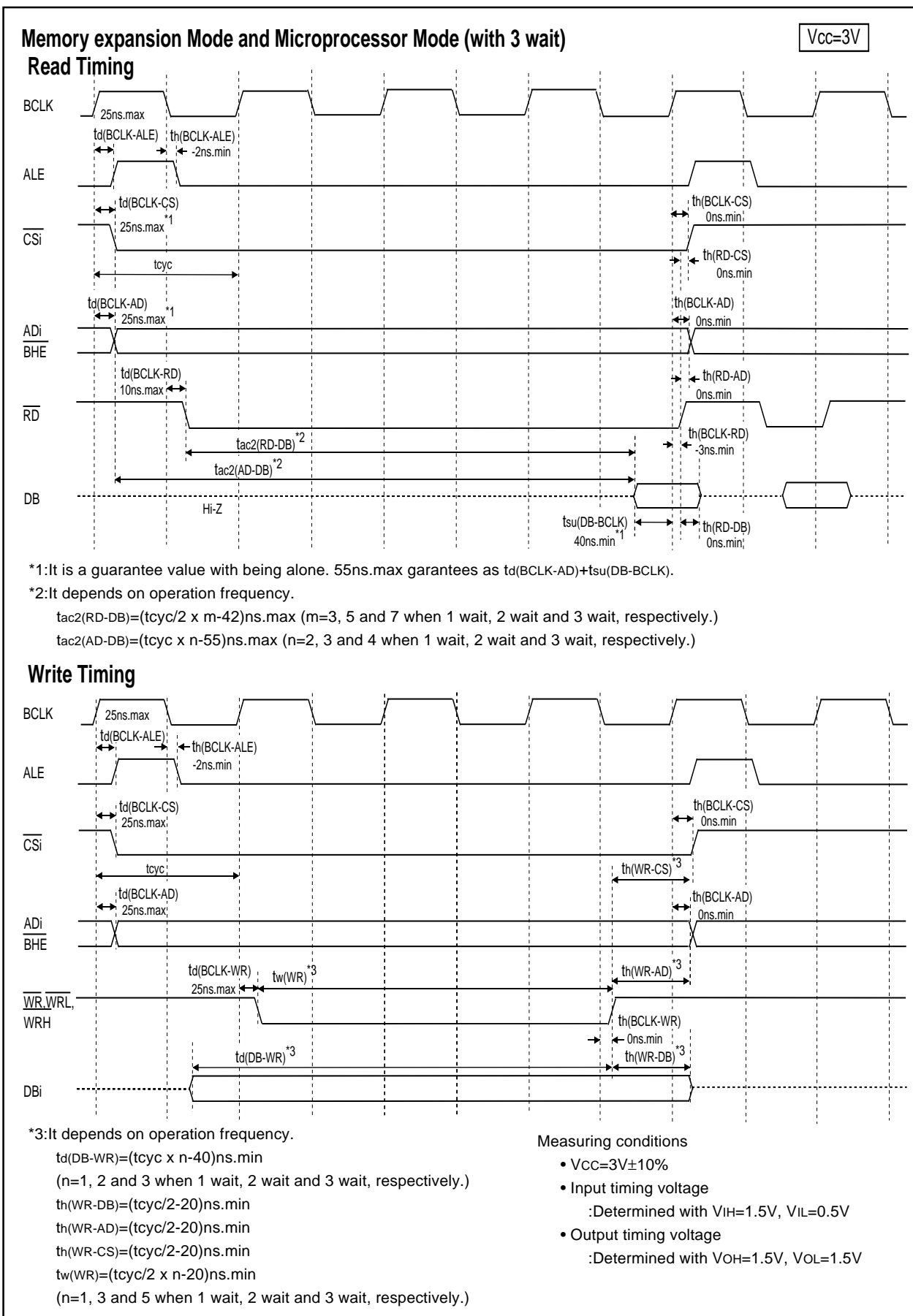


Figure 28.18 V<sub>CC</sub>=3V timing diagram (4)

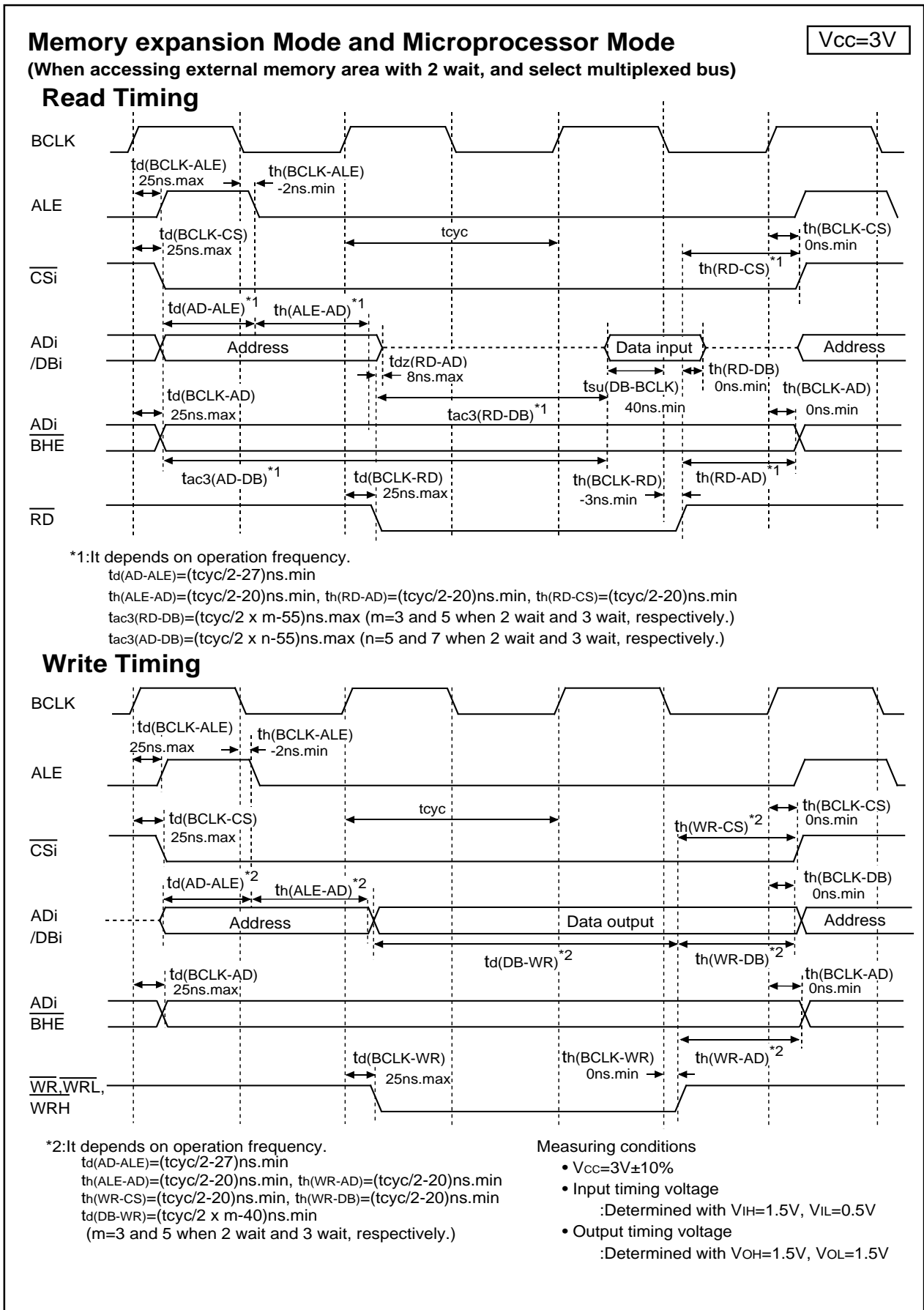


Figure 28.19 V<sub>CC</sub>=3V timing diagram (5)

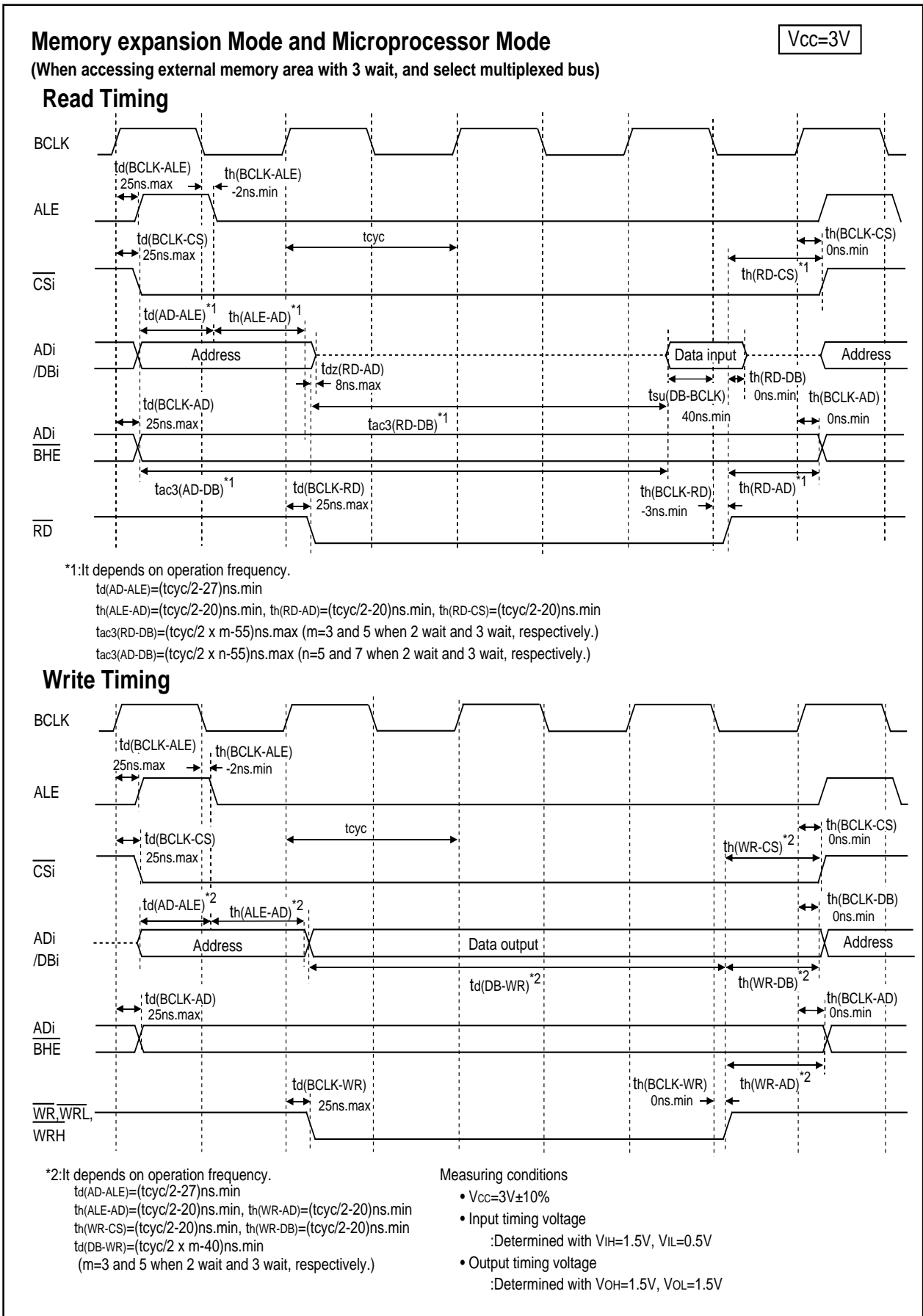


Figure 28.20 V<sub>CC</sub>=3V timing diagram (6)

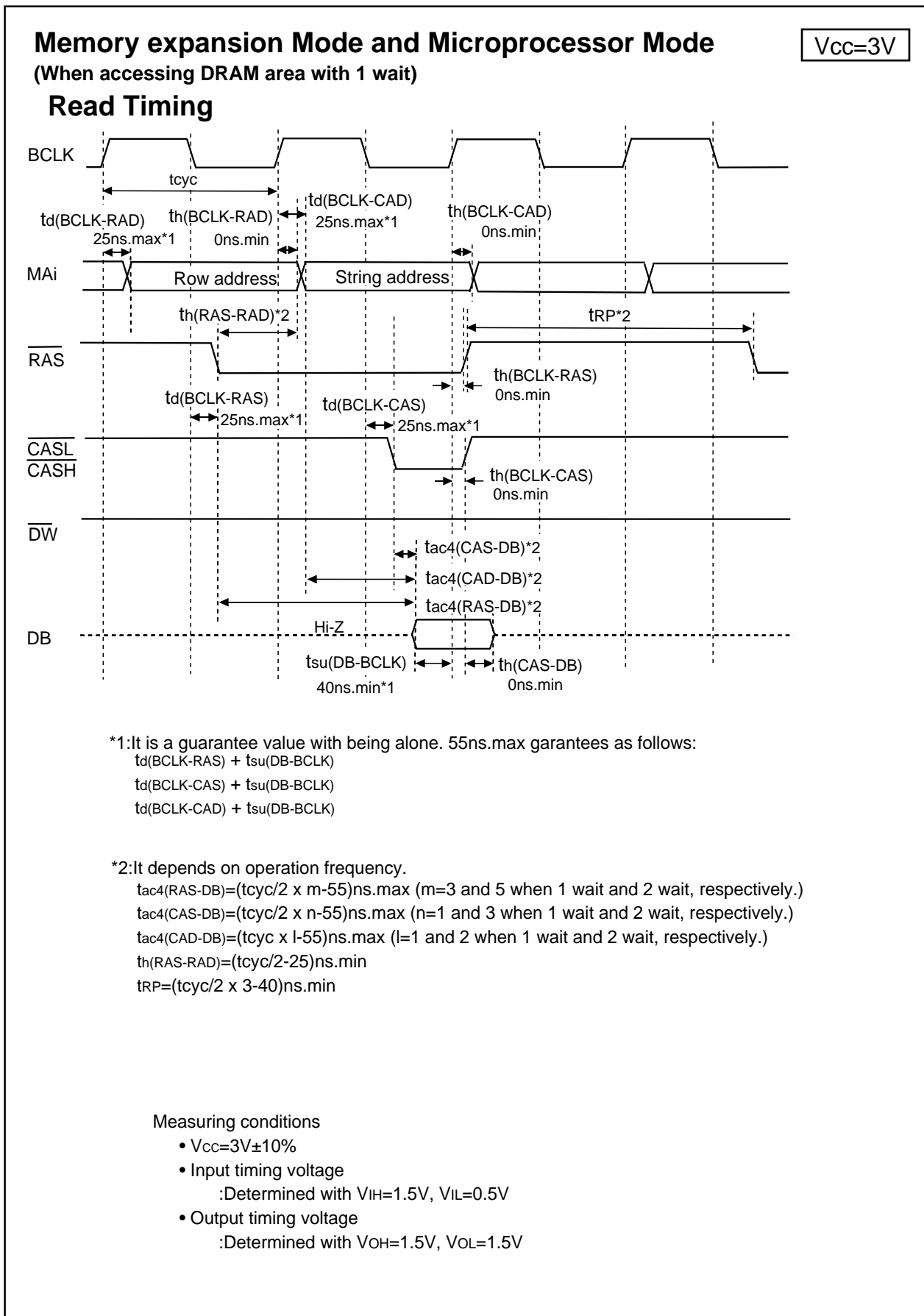
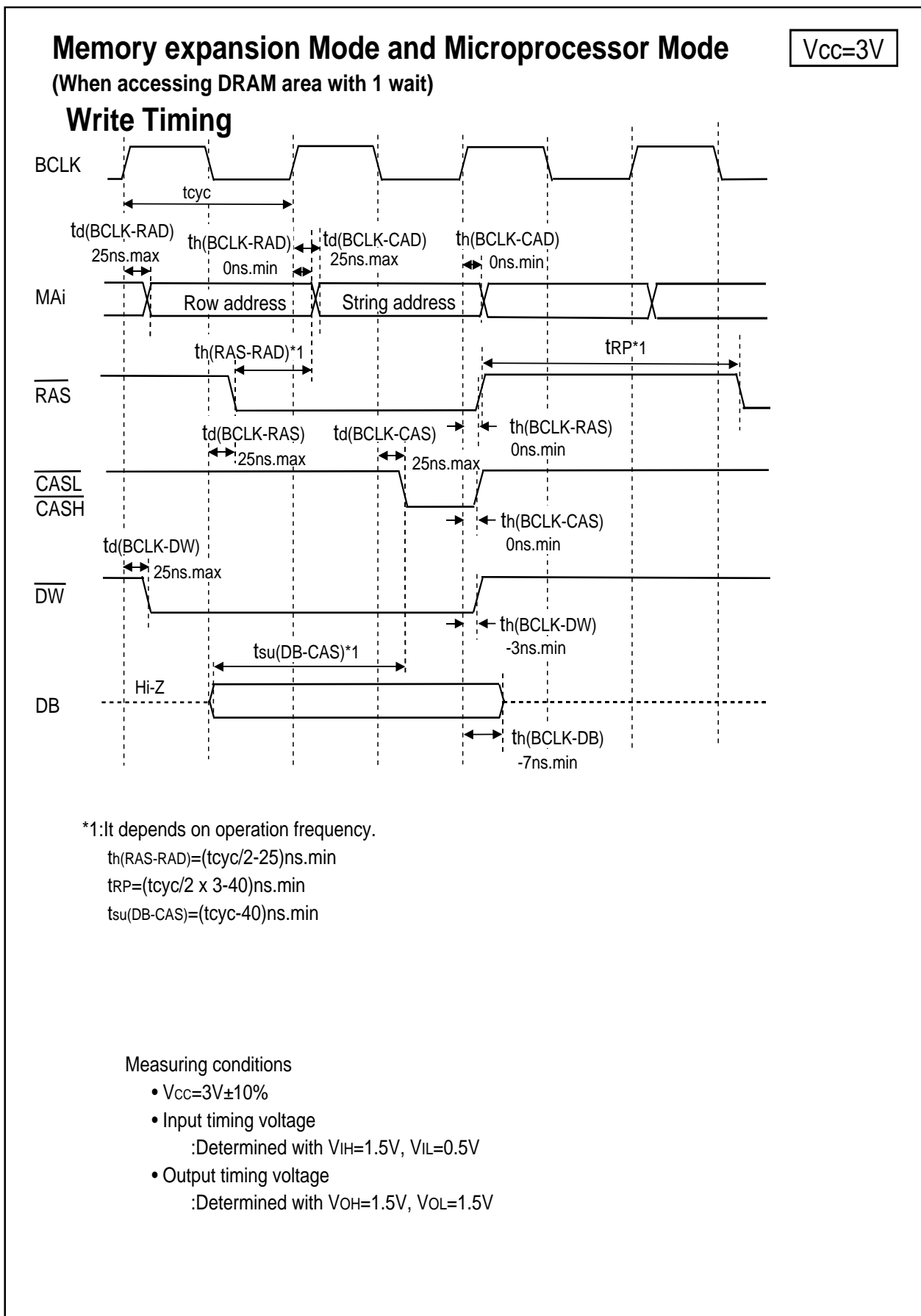
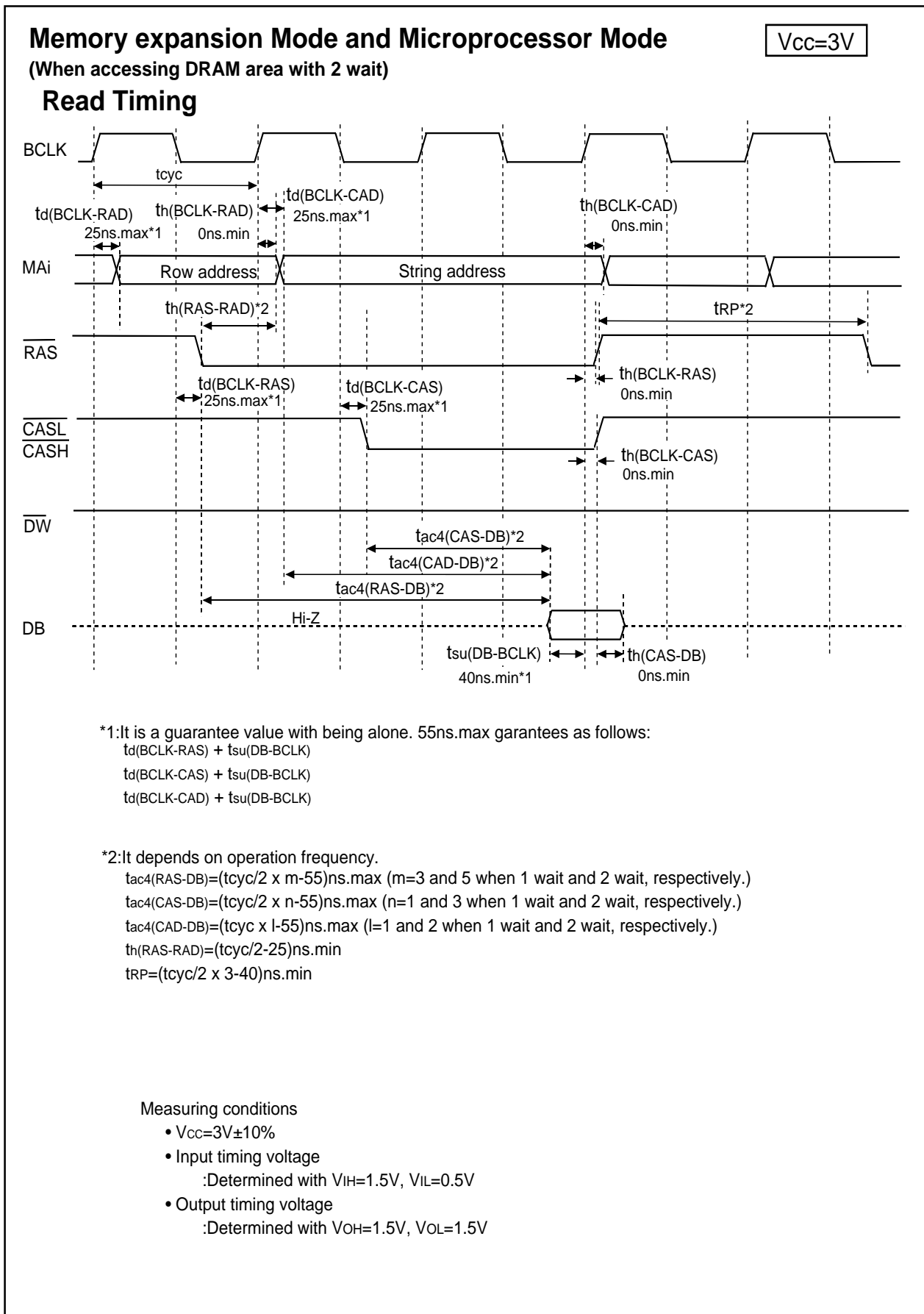


Figure 28.21 V<sub>CC</sub>=3V timing diagram (7)

Figure 28.22 V<sub>CC</sub>=3V timing diagram (8)

Figure 28.23  $V_{CC}=3V$  timing diagram (9)

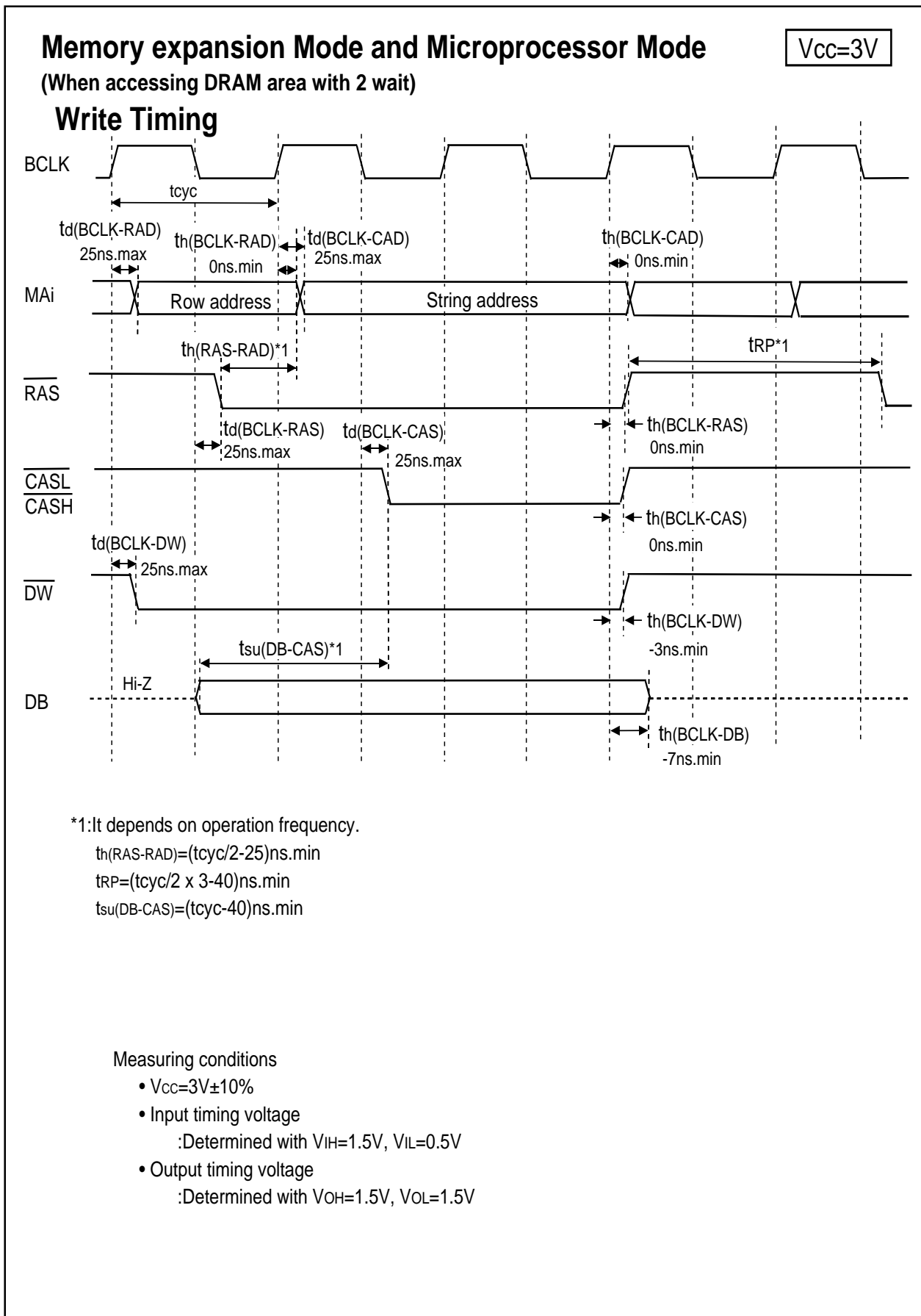


Figure 28.24 V<sub>CC</sub>=3V timing diagram (10)



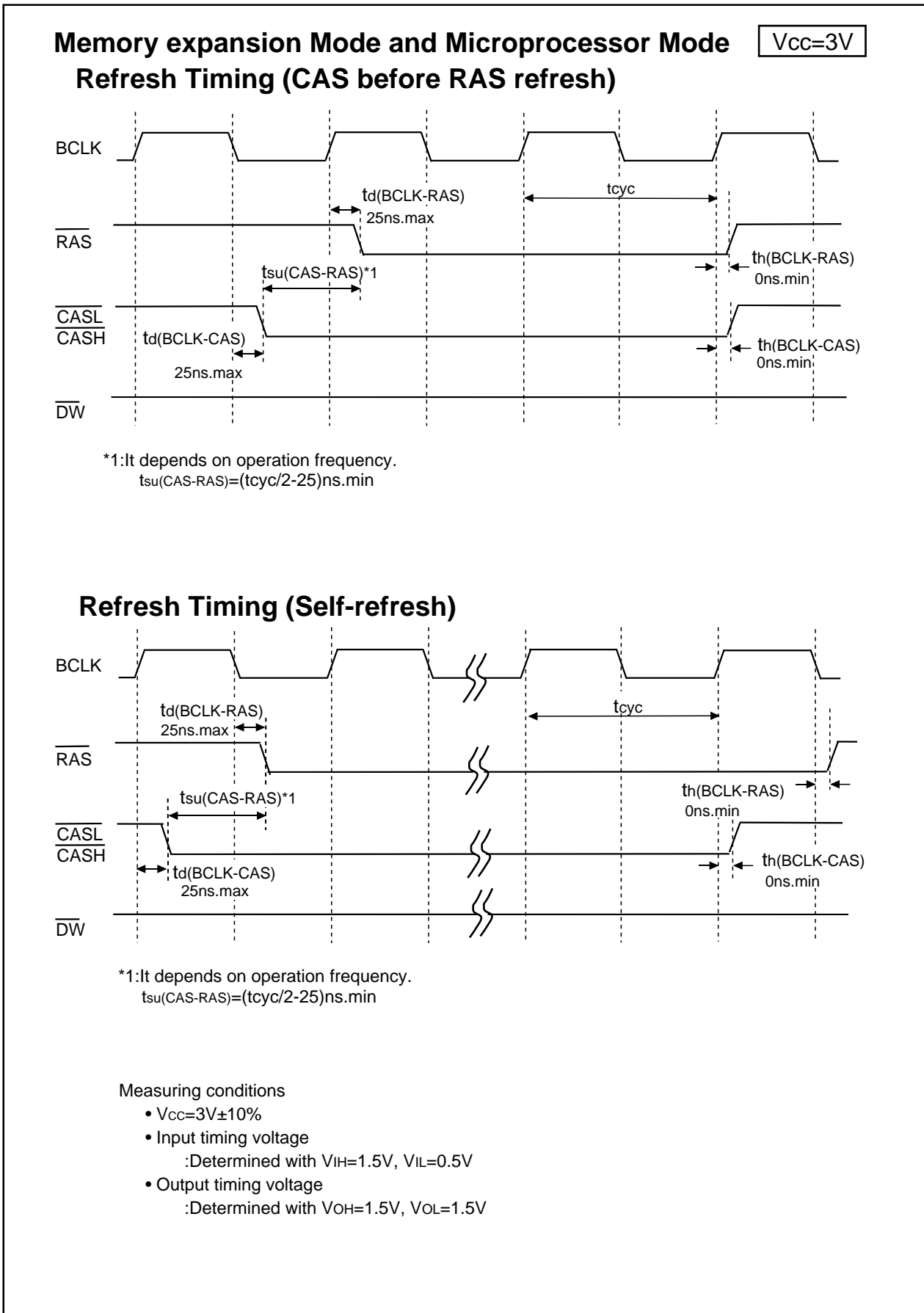


Figure 28.25 V<sub>CC</sub>=3V timing diagram (11)

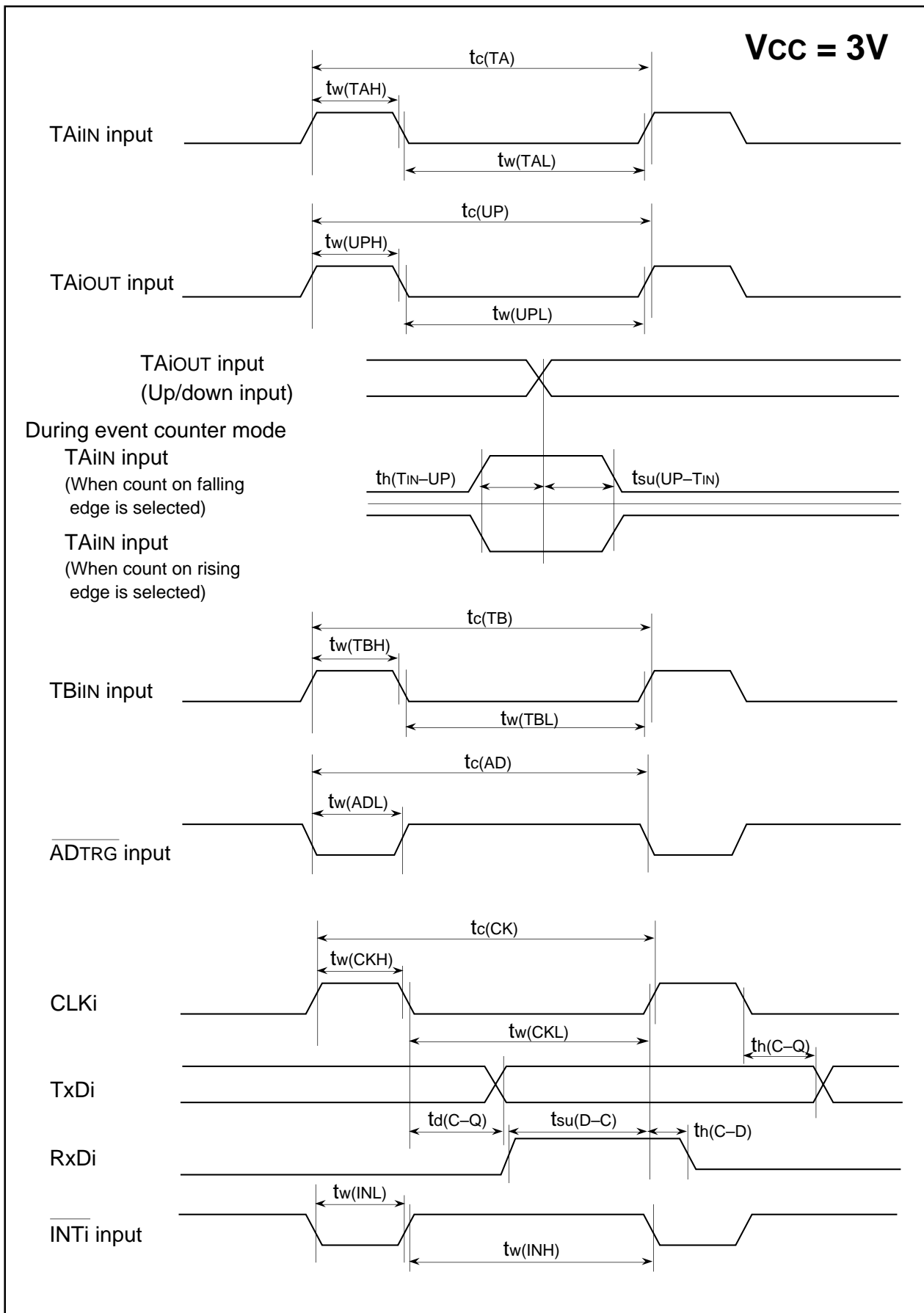


Figure 28.26 VCC=3V timing diagram (12)

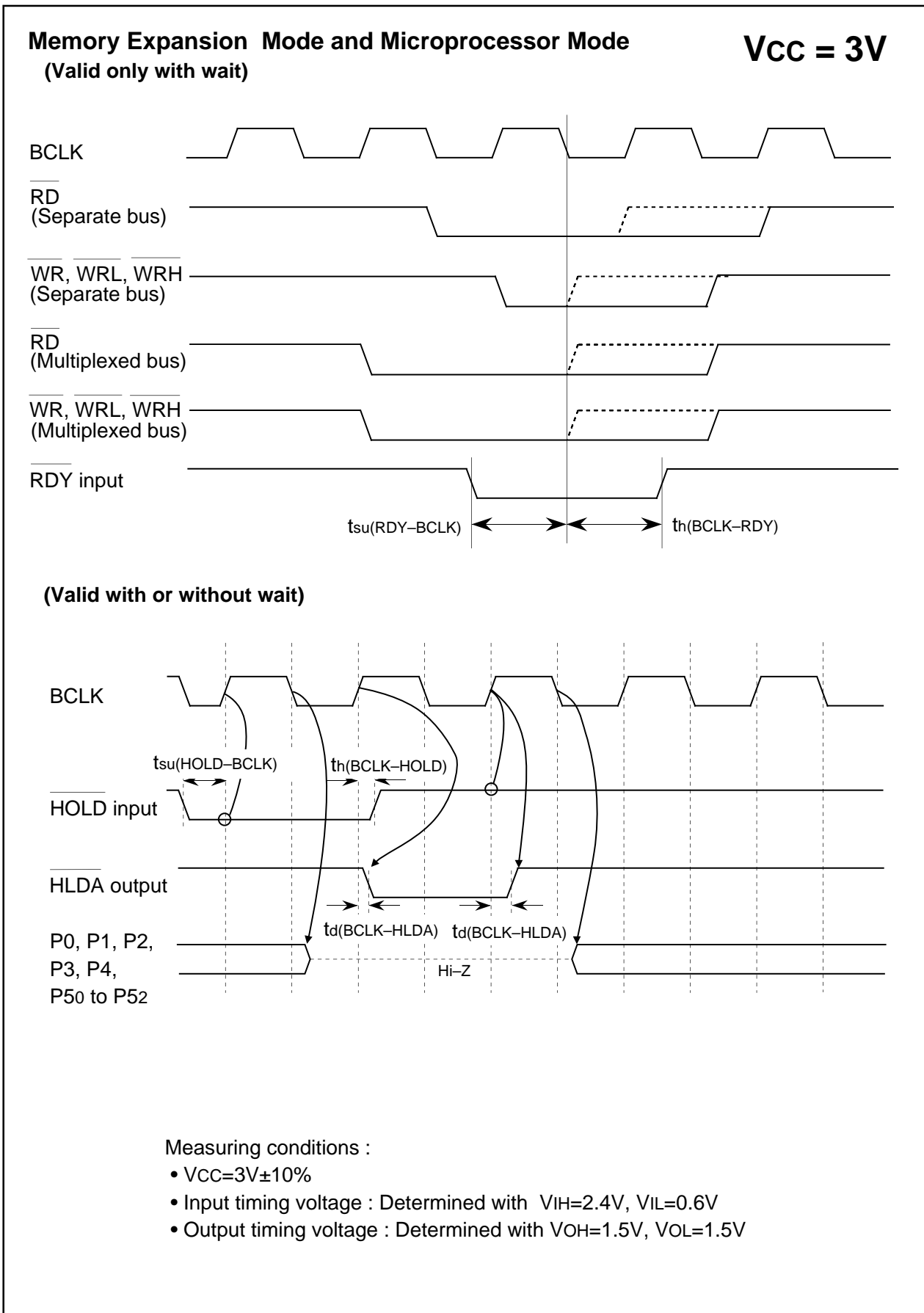


Figure 28.27 VCC=3V timing diagram (13)

## 29. Flash Memory Version

### Outline Performance

Table 29.1 shows the outline performance of the M16C/80 (flash memory version).

**Table 29.1 Outline Performance of the M16C/80 (flash memory version)**

Item		Performance
Power supply voltage		5V version: f(XIN)=20MHz, without wait, 4.2V to 5.5V f(XIN)=10MHz, without wait, 2.7V to 5.5V
Program/erase voltage		5V version: 4.2V to 5.5 V f(BCLK)=12.5MHz, with one wait f(BCLK)=6.25MHz, without wait
Flash memory operation mode		Three modes (parallel I/O, standard serial I/O, CPU rewrite)
Erase block division	User ROM area	See Figure 29.3
	Boot ROM area	One division (8 Kbytes) (Note 1)
Program method		In units of pages (in units of 256 bytes)
Erase method		Collective erase/block erase
Program/erase control method		Program/erase control by software command
Protect method		Protected for each block by lock bit
Number of commands		8 commands
Program/erase count		100 times
Data holding		10 years
ROM code protect		Parallel I/O and standard serial modes are supported.

Note: The boot ROM area contains a standard serial I/O mode control program which is stored in it when shipped from the factory. This area can be erased and programmed in only parallel I/O mode.

The following shows Renesas plans to develop a line of M16C/80 products (flash memory version).

- (1) ROM capacity
- (2) Package    100P6S-A ... Plastic molded QFP  
                   100P6Q-A ... Plastic molded QFP  
                   144P6Q-A ... Plastic molded QFP

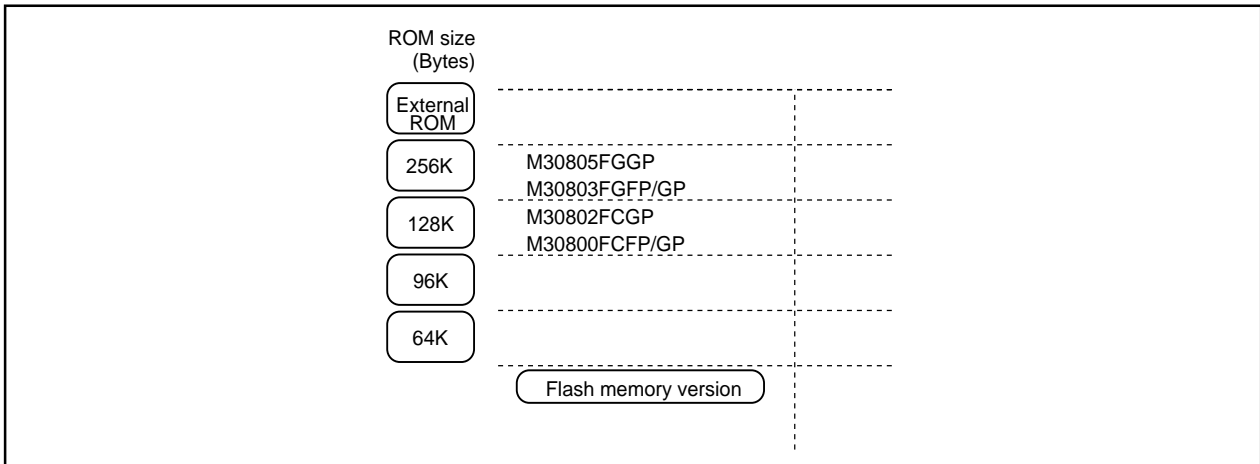


Figure 29.1 ROM Expansion

The following lists the M16C/80 products to be supported in the future.

Table 29.2 Product List

Type No	ROM capacity	RAM capacity	Package type	Remarks
M30800FCFP	128 Kbytes	10 Kbytes	100P6S-A	
M30800FCGP			100P6Q-A	
M30803FGFP	256 Kbytes	20 Kbytes	100P6S-A	
M30803FGGP			100P6Q-A	
M30802FCGP	128 Kbytes	10 Kbytes	144P6Q-A	
M30805FGGP	256 Kbytes	20 Kbytes		

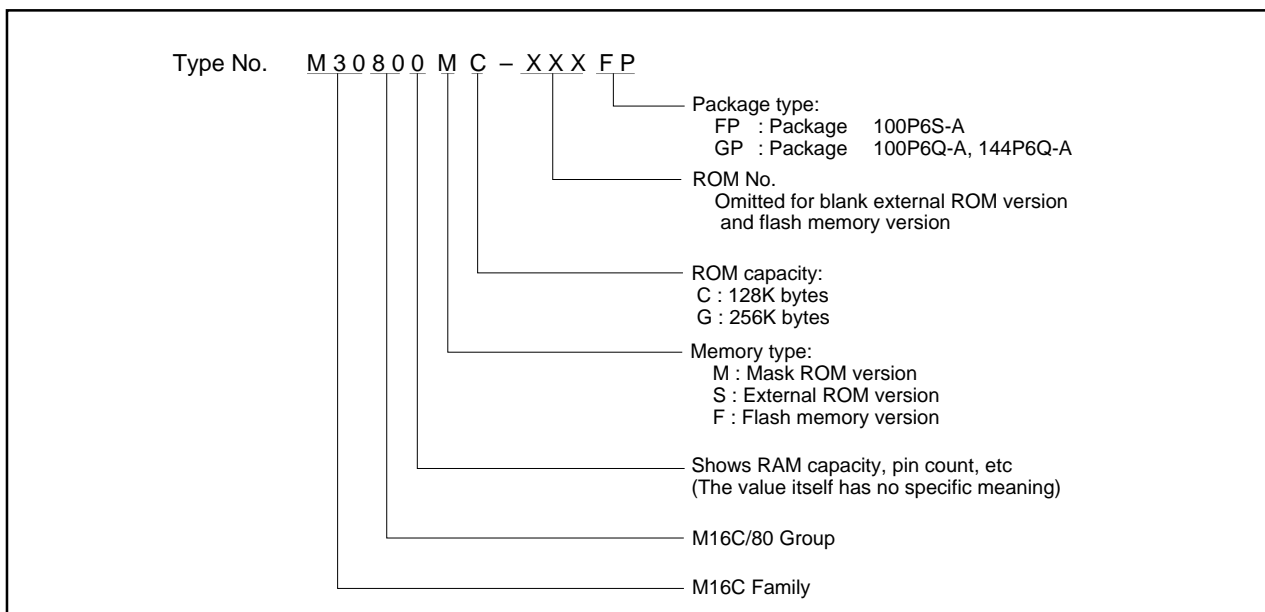


Figure 29.2 Type No., memory size, and package

### Flash Memory

The M16C/80 (flash memory version) contains the flash memory that can be rewritten with a single voltage of 5 V. For this flash memory, three flash memory modes are available in which to read, program, and erase: parallel I/O and standard serial I/O modes in which the flash memory can be manipulated using a programmer and a CPU rewrite mode in which the flash memory can be manipulated by the Central Processing Unit (CPU). Each mode is detailed in the pages to follow.

The flash memory is divided into several blocks as shown in Figure 29.3, so that memory can be erased one block at a time. Each block has a lock bit to enable or disable execution of an erase or program operation, allowing for data in each block to be protected.

In addition to the ordinary user ROM area to store a microcomputer operation control program, the flash memory has a boot ROM area that is used to store a program to control rewriting in CPU rewrite and standard serial I/O modes. This boot ROM area has had a standard serial I/O mode control program stored in it when shipped from the factory. However, the user can write a rewrite control program in this area that suits the user's application system. This boot ROM area can be rewritten in only parallel I/O mode.

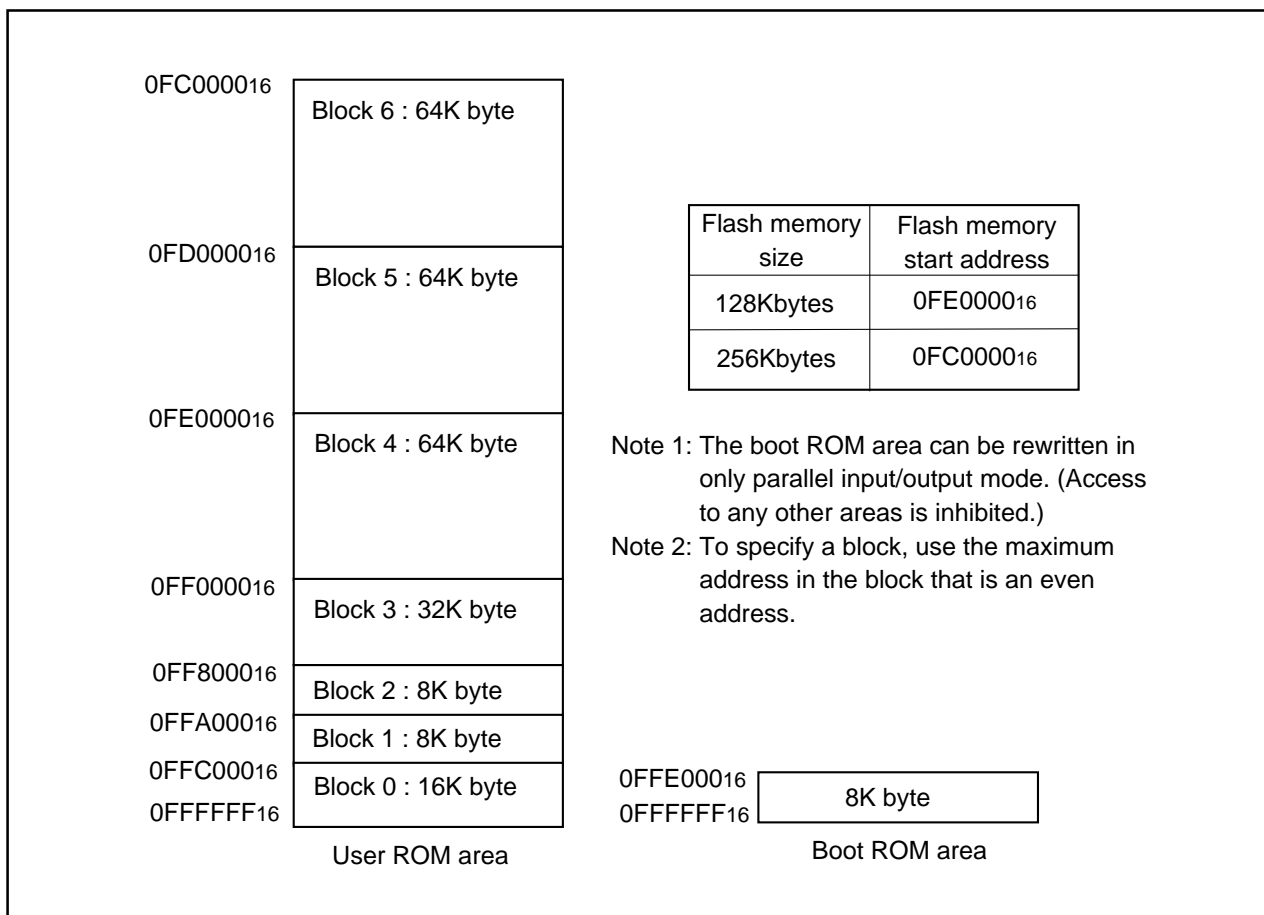


Figure 29.3 Block diagram of flash memory version

### 30. CPU Rewrite Mode

In CPU rewrite mode, the on-chip flash memory can be operated on (read, program, or erase) under control of the Central Processing Unit (CPU).

In CPU rewrite mode, only the user ROM area shown in Figure 29.3 can be rewritten; the boot ROM area cannot be rewritten. Make sure the program and block erase commands are issued for only the user ROM area and each block area.

The control program for CPU rewrite mode can be stored in either user ROM or boot ROM area. In the CPU rewrite mode, because the flash memory cannot be read from the CPU, the rewrite control program must be transferred to any area other than the internal flash memory before it can be executed.

#### Microcomputer Mode and Boot Mode

The control program for CPU rewrite mode must be written into the user ROM or boot ROM area in parallel I/O mode beforehand. (If the control program is written into the boot ROM area, the standard serial I/O mode becomes unusable.)

See Figure 29.3 for details about the boot ROM area.

Normal microcomputer mode is entered when the microcomputer is reset with pulling CNVss pin low. In this case, the CPU starts operating using the control program in the user ROM area.

When the microcomputer is reset by pulling the P55 pin low, the CNVss pin high, and the P50 pin high, the CPU starts operating using the control program in the boot ROM area. This mode is called the "boot" mode. The control program in the boot ROM area can also be used to rewrite the user ROM area.

#### Block Address

Block addresses refer to the maximum even address of each block. These addresses are used in the block erase command, lock bit program command, and read lock status command.

### Outline Performance (CPU Rewrite Mode)

In the CPU rewrite mode, the CPU erases, programs and reads the internal flash memory as instructed by software commands. Operations must be executed from a memory other than the internal flash memory, such as the internal RAM.

When the CPU rewrite mode select bit (bit 1 at address 037716) is set to "1", transition to CPU rewrite mode occurs and software commands can be accepted.

In the CPU rewrite mode, write to and read from software commands and data into even-numbered address ("0" for byte address A0) in 16-bit units. Always write 8-bit software commands into even-numbered address. Commands are ignored with odd-numbered addresses.

Use software commands to control program and erase operations. Whether a program or erase operation has terminated normally or in error can be verified by reading the status register.

Read data from an even address in the user ROM area when reading the status register.

Figure 30.1 shows the flash memory control register 0 and the flash memory control register 1.

Bit 0 of the flash memory control register 0 is the RY/BY status flag used exclusively to read the operating status of the flash memory. During programming and erase operations, it is "0". Otherwise, it is "1".

Bit 1 of the flash memory control register 0 is the CPU rewrite mode select bit. The CPU rewrite mode is entered by setting this bit to "1", so that software commands become acceptable. In CPU rewrite mode, the CPU becomes unable to access the internal flash memory directly. Therefore, write bit 1 in an area other than the internal flash memory. To set this bit to "1", it is necessary to write "0" and then write "1" in succession when NMI pin is "H" level. The bit can be set to "0" by only writing a "0".

Bit 2 of the flash memory control register 0 is a lock bit disable bit. By setting this bit to "1", it is possible to disable erase and write protect (block lock) effectuated by the lock bit data. The lock bit disable select bit only disables the lock bit function; it does not change the lock data bit value. However, if an erase operation is performed when this bit = "1", the lock bit data that is "0" (locked) is set to "1" (unlocked) after erasure. To set this bit to "1", it is necessary to write "0" and then write "1" in succession. This bit can be manipulated only when the CPU rewrite mode select bit = "1".

Bit 3 of the flash memory control register 0 is the flash memory reset bit used to reset the control circuit of the internal flash memory. This bit is used when exiting CPU rewrite mode and when flash memory access has failed. When the CPU rewrite mode select bit is "1", writing "1" for this bit resets the control circuit. To release the reset, it is necessary to set this bit to "0".

Bit 5 of the flash memory control register 0 is a user ROM area select bit which is effective in only boot mode. If this bit is set to "1" in boot mode, the area to be accessed is switched from the boot ROM area to the user ROM area. When the CPU rewrite mode needs to be used in boot mode, set this bit to "1". Note that if the microcomputer is booted from the user ROM area, it is always the user ROM area that can be accessed and this bit has no effect. When in boot mode, the function of this bit is effective regardless of whether the CPU rewrite mode is on or off. Use the control program except in the internal flash memory to rewrite this bit.

Bit 3 of the flash memory control register 1 turns power supply to the internal flash memory on/off. When this bit is set to "1", power is not supplied to the internal flash memory, thus power consumption can be reduced. However, in this state, the internal flash memory cannot be accessed. To set this bit to "1", it is necessary to write "0" and then write "1" in succession. Use this bit mainly in the low speed mode (when XCIN is the block count source of BCLK).

When the CPU is shifted to the stop or wait modes, power to the internal flash memory is automatically shut off. It is reconnected automatically when CPU operation is restored. Therefore, it is not particularly necessary to set flash memory control register 1.



Figure 30.2 shows a flowchart for setting/releasing the CPU rewrite mode. Figure 30.3 shows a flowchart for shifting to the low speed mode. Always perform operation as indicated in these flowcharts.

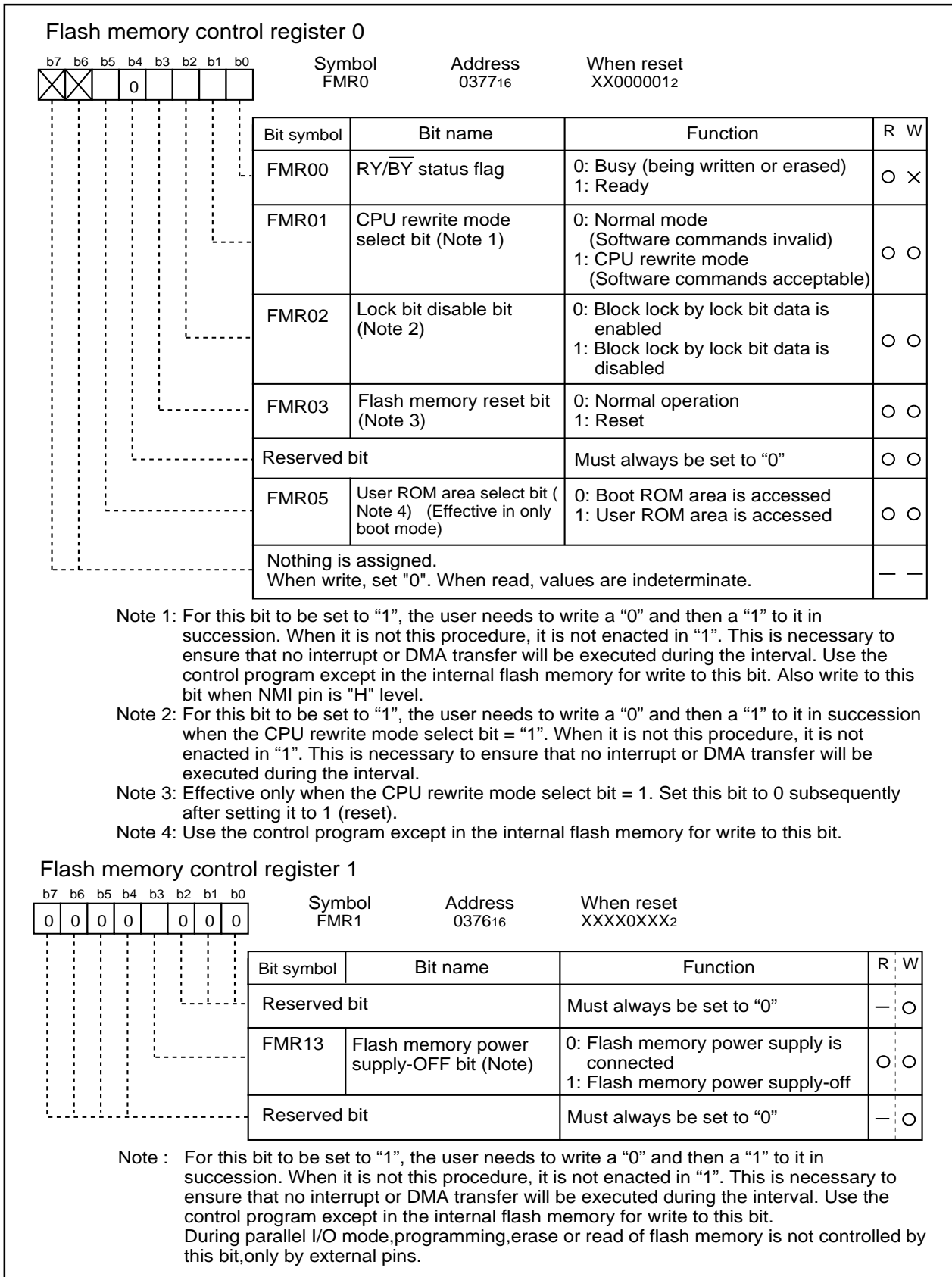


Figure 30.1 Flash memory control registers

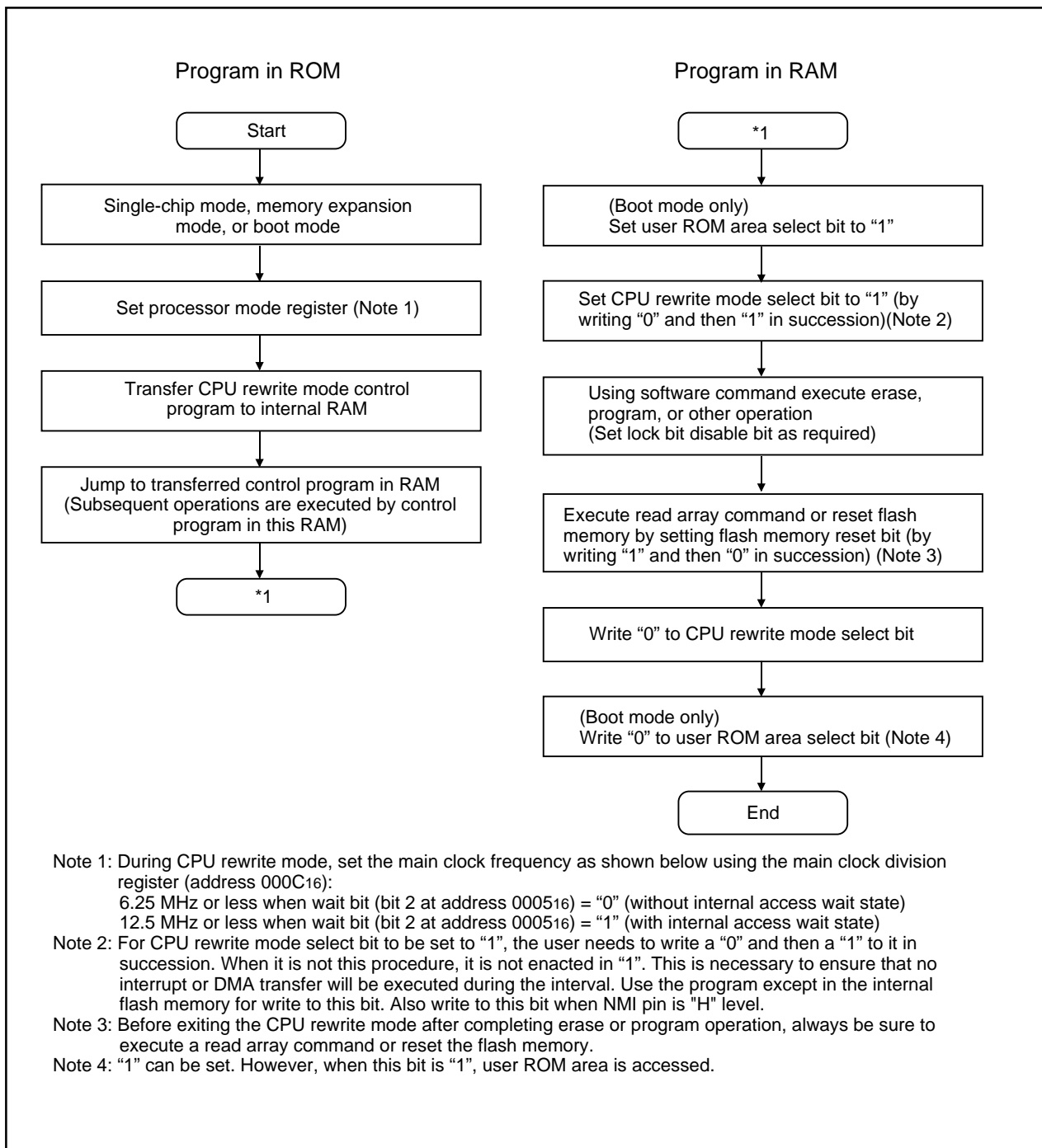


Figure 30.2 CPU rewrite mode set/reset flowchart

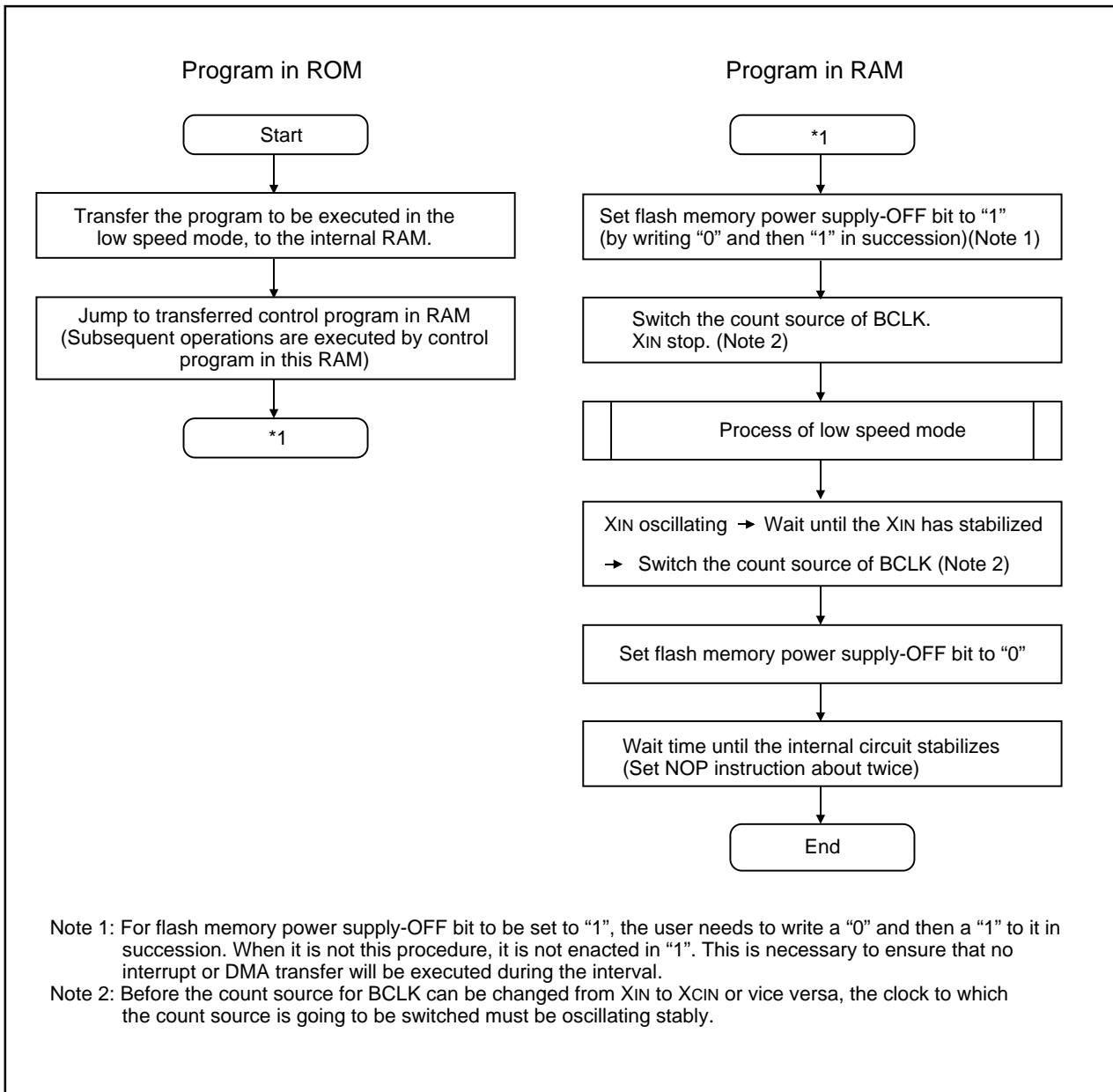


Figure 30.3 Shifting to the low speed mode flowchart

## Precautions on CPU Rewrite Mode

Described below are the precautions to be observed when rewriting the flash memory in CPU rewrite mode.

### (1) Operation speed

During CPU rewrite mode, set the BCLK as shown below using the main clock division register (address 000C16):

6.25 MHz or less when wait bit (bit 2 at address 000516) = 0 (without internal access wait state)

12.5 MHz or less when wait bit (bit 2 at address 000516) = 1 (with internal access wait state)

### (2) Instructions inhibited against use

The instructions listed below cannot be used during CPU rewrite mode because they refer to the internal data of the flash memory:

UND instruction, INTO instruction, JMPS instruction, JSRS instruction, and BRK instruction

### (3) Interrupts inhibited against use

The address match interrupt cannot be used during CPU rewrite mode because they refer to the internal data of the flash memory. If interrupts have their vector in the variable vector table, they can be used by transferring the vector into the RAM area. The  $\overline{\text{NMI}}$  and watchdog timer interrupts each can be used to change the CPU rewrite mode select bit forcibly to normal mode (FMR01="0") upon occurrence of the interrupt. Since the rewrite operation is halted when the  $\overline{\text{NMI}}$  and watchdog timer interrupts occur, set the CPU rewrite mode select bit to "1" and the erase/program operation needs to be performed over again.

### (4) Reset

Reset input is always accepted.

### (5) Access disable

Write CPU rewrite mode select bit, flash memory power supply-OFF bit and user ROM area select bit in an area other than the internal flash memory.

### (6) How to access

For CPU rewrite mode select bit, lock bit disable bit, and flash memory power supply-OFF bit to be set to "1", the user needs to write a "0" and then a "1" to it in succession. When it is not this procedure, it is not enacted in "1". This is necessary to ensure that no interrupt or DMA transfer will be executed during the interval.

Write to the CPU rewrite mode select bit when NMI pin is "H" level.

### (7) Writing in the user ROM area

If power is lost while rewriting blocks that contain the flash rewrite program with the CPU rewrite mode, those blocks may not be correctly rewritten and it is possible that the flash memory can no longer be rewritten after that. Therefore, it is recommended to use the standard serial I/O mode or parallel I/O mode to rewrite these blocks.

### (8) Using the lock bit

To use the CPU rewrite mode, use a boot program that can set and cancel the lock command.

## Software Commands

Table 30.1 lists the software commands available with the M16C/62A (flash memory version).

After setting the CPU rewrite mode select bit to 1, write a software command to specify an erase or program operation. Note that when entering a software command, the upper byte (D8 to D15) is ignored.

The content of each software command is explained below.

**Table 30.1 List of software commands (CPU rewrite mode)**

Command	First bus cycle			Second bus cycle			Third bus cycle		
	Mode	Address	Data (D0 to D7)	Mode	Address	Data (D0 to D7)	Mode	Address	Data (D0 to D7)
Read array	Write	X (Note 6)	FF <sub>16</sub>						
Read status register	Write	X	70 <sub>16</sub>	Read	X (Note 6)	SRD (Note 2)			
Clear status register	Write	X	50 <sub>16</sub>						
Page program (Note 3)	Write	X	41 <sub>16</sub>	Write	WA0 (Note 3)	WD0 (Note 3)	Write	WA1	WD1
Block erase	Write	X	20 <sub>16</sub>	Write	BA (Note 4)	D0 <sub>16</sub>			
Erase all unlock block	Write	X	A7 <sub>16</sub>	Write	X	D0 <sub>16</sub>			
Lock bit program	Write	X	77 <sub>16</sub>	Write	BA	D0 <sub>16</sub>			
Read lock bit status	Write	X	71 <sub>16</sub>	Read	BA	D <sub>6</sub> (Note 5)			

Note 1: When a software command is input, the high-order byte of data (D8 to D15) is ignored.

Note 2: SRD = Status Register Data

Note 3: WA = Write Address, WD = Write Data

WA and WD must be set sequentially from 00<sub>16</sub> to FE<sub>16</sub> (byte address; however, an even address). The page size is 256 bytes.

Note 4: BA = Block Address (Enter the maximum address of each block that is an even address.)

Note 5: D<sub>6</sub> corresponds to the block lock status. Block not locked when D<sub>6</sub> = 1, block locked when D<sub>6</sub> = 0.

Note 6: X denotes a given address in the user ROM area (that is an even address).

### Read Array Command (FF<sub>16</sub>)

The read array mode is entered by writing the command code "FF<sub>16</sub>" in the first bus cycle. When an even address to be read is input in one of the bus cycles that follow, the content of the specified address is read out at the data bus (D0–D15), 16 bits at a time.

The read array mode is retained intact until another command is written.

### Read Status Register Command (70<sub>16</sub>)

When the command code "70<sub>16</sub>" is written in the first bus cycle, the content of the status register is read out at the data bus (D0–D7) by a read in the second bus cycle. (Set an address to even address in the user ROM area).

The status register is explained in the next section.

### Clear Status Register Command (50<sub>16</sub>)

This command is used to clear the bits SR3 to 5 of the status register after they have been set. These bits indicate that operation has ended in an error. To use this command, write the command code "50<sub>16</sub>" in the first bus cycle.

### Page Program Command (41<sub>16</sub>)

Page program allows for high-speed programming in units of 256 bytes. Page program operation starts when the command code "41<sub>16</sub>" is written in the first bus cycle. In the second bus cycle through the 129th bus cycle, the write data is sequentially written 16 bits at a time. At this time, the addresses A0-A7 need to be incremented by 2 from "00<sub>16</sub>" to "FE<sub>16</sub>." When the system finishes loading the data, it starts an auto write operation (data program and verify operation).

Whether the auto write operation is completed can be confirmed by reading the status register or the flash memory control register 0. At the same time the auto write operation starts, the read status register mode is automatically entered, so the content of the status register can be read out. The status register bit 7 (SR7) is set to 0 at the same time the auto write operation starts and is returned to 1 upon completion of the auto write operation. In this case, the read status register mode remains active until the Read Array command (FF<sub>16</sub>) or Read Lock Bit Status command (71<sub>16</sub>) is written or the flash memory is reset using its reset bit.

The RY/ $\overline{\text{BY}}$  status flag of the flash memory control register 0 is 0 during auto write operation and 1 when the auto write operation is completed as is the status register bit 7.

After the auto write operation is completed, the status register can be read out to know the result of the auto write operation. For details, refer to the section where the status register is detailed.

Figure 30.4 shows an example of a page program flowchart.

Each block of the flash memory can be write protected by using a lock bit. For details, refer to the section where the data protect function is detailed.

Additional writes to the already programmed pages are prohibited.

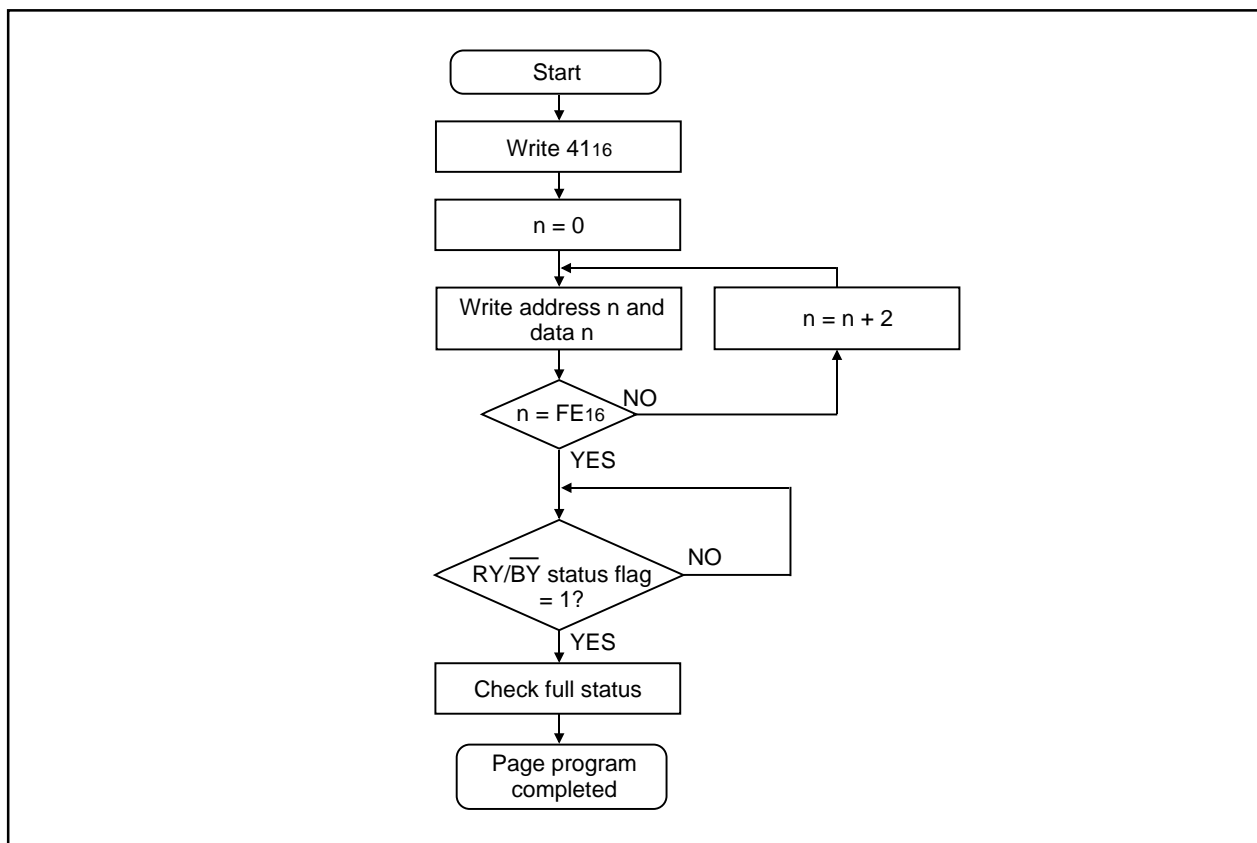


Figure 30.4 Page program flowchart

### Block Erase Command (20<sub>16</sub>/D0<sub>16</sub>)

By writing the command code “20<sub>16</sub>” in the first bus cycle and the confirmation command code “D0<sub>16</sub>” in the second bus cycle that follows to the block address of a flash memory block, the system initiates an auto erase (erase and erase verify) operation.

Whether the auto erase operation is completed can be confirmed by reading the status register or the flash memory control register 0. At the same time the auto erase operation starts, the read status register mode is automatically entered, so the content of the status register can be read out. The status register bit 7 (SR7) is set to 0 at the same time the auto erase operation starts and is returned to 1 upon completion of the auto erase operation. In this case, the read status register mode remains active until the Read Array command (FF<sub>16</sub>) or Read Lock Bit Status command (71<sub>16</sub>) is written or the flash memory is reset using its reset bit.

The RY/ $\overline{\text{BY}}$  status flag of the flash memory control register 0 is 0 during auto erase operation and 1 when the auto erase operation is completed as is the status register bit 7.

After the auto erase operation is completed, the status register can be read out to know the result of the auto erase operation. For details, refer to the section where the status register is detailed.

Figure 30.5 shows an example of a block erase flowchart.

Each block of the flash memory can be protected against erasure by using a lock bit. For details, refer to the section where the data protect function is detailed.

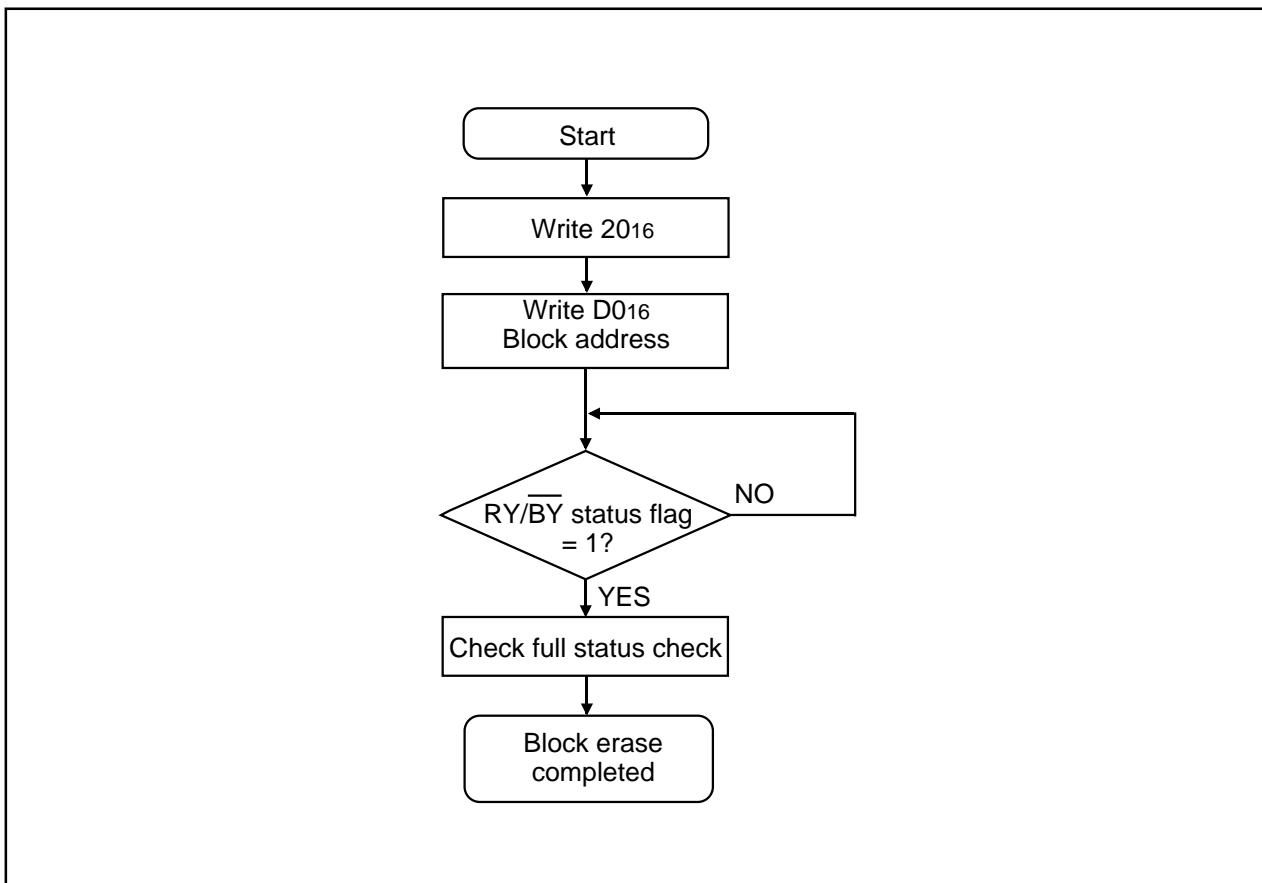


Figure 30.5 Block erase flowchart

**Erase All Unlock Blocks Command (A7<sub>16</sub>/D0<sub>16</sub>)**

By writing the command code "A7<sub>16</sub>" in the first bus cycle and the confirmation command code "D0<sub>16</sub>" in the second bus cycle that follows, the system starts erasing blocks successively.

Whether the erase all unlock blocks command is terminated can be confirmed by reading the status register or the flash memory control register 0, in the same way as for block erase. Also, the status register can be read out to know the result of the auto erase operation.

When the lock bit disable bit of the flash memory control register 0 = 1, all blocks are erased no matter how the lock bit is set. On the other hand, when the lock bit disable bit = 0, the function of the lock bit is effective and only nonlocked blocks (where lock bit data = 1) are erased.

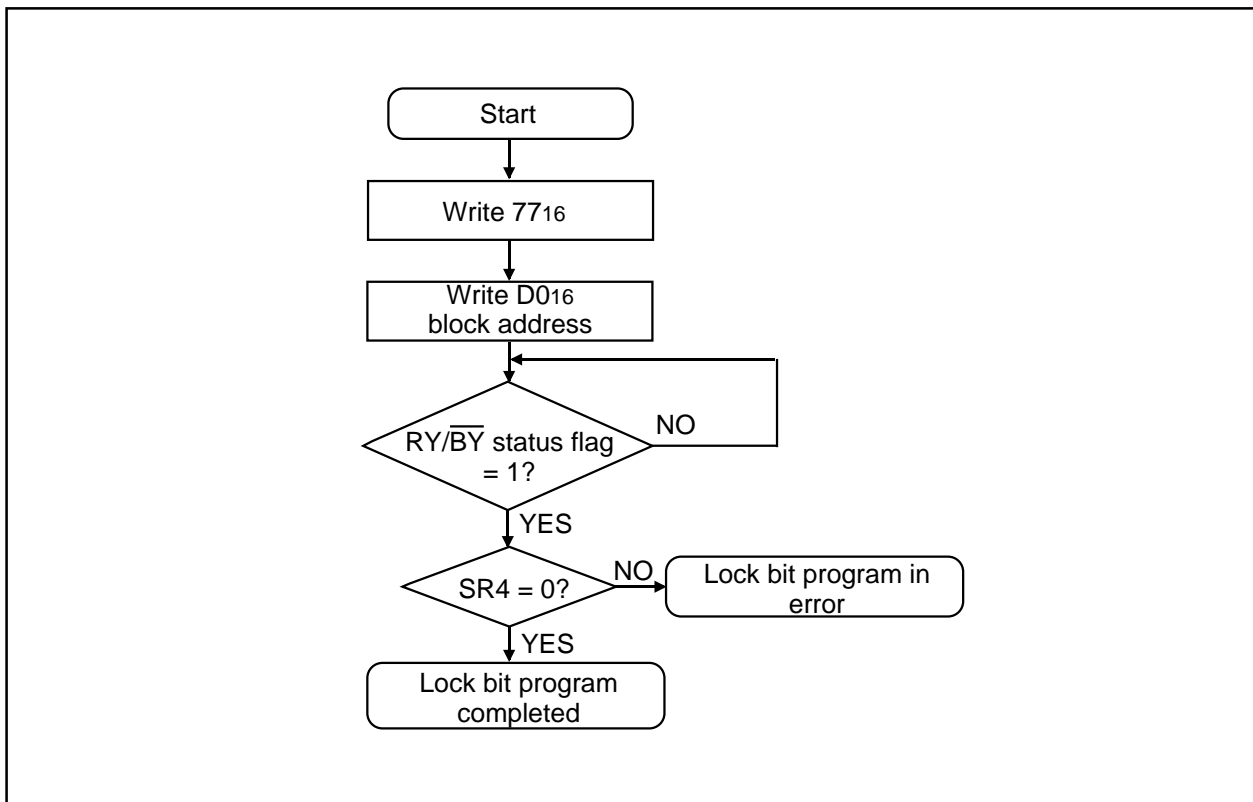
**Lock Bit Program Command (77<sub>16</sub>/D0<sub>16</sub>)**

By writing the command code "77<sub>16</sub>" in the first bus cycle and the confirmation command code "D0<sub>16</sub>" in the second bus cycle that follows to the block address of a flash memory block, the system sets the lock bit for the specified block to 0 (locked).

Figure 30.6 shows an example of a lock bit program flowchart. The status of the lock bit (lock bit data) can be read out by a read lock bit status command.

Whether the lock bit program command is terminated can be confirmed by reading the status register or the flash memory control register 0, in the same way as for page program.

For details about the function of the lock bit and how to reset the lock bit, refer to the section where the data protect function is detailed.



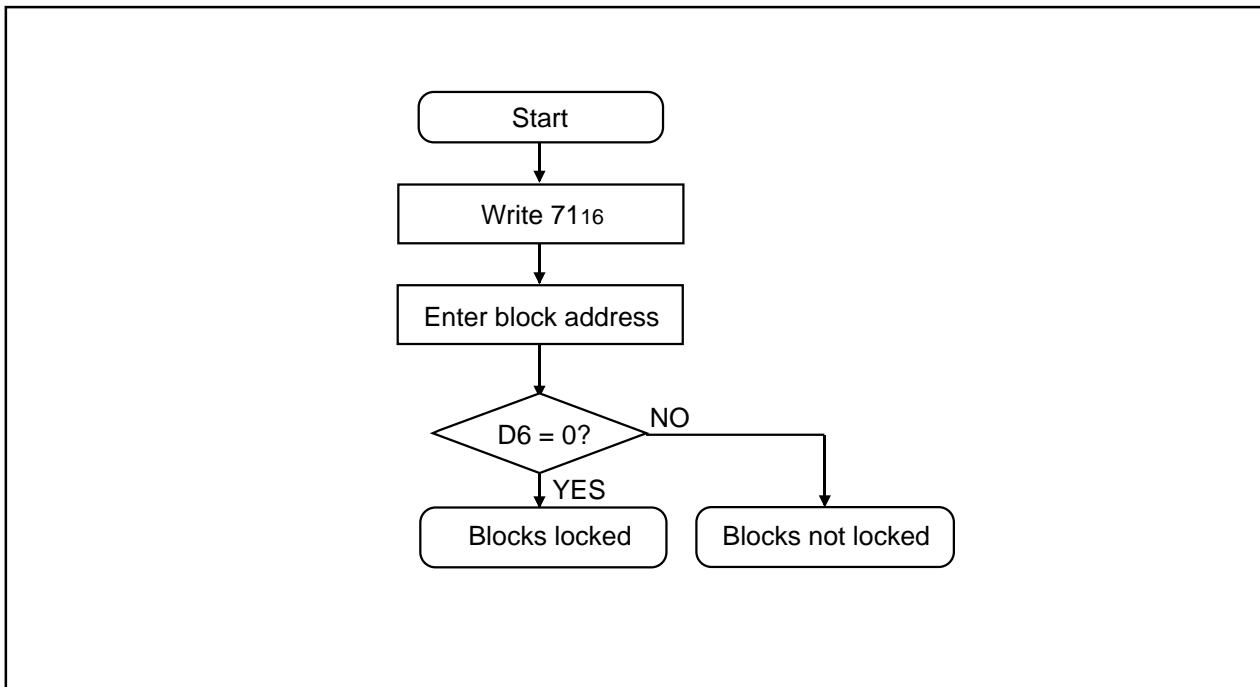
**Figure 30.6 Lock bit program flowchart**



**Read Lock Bit Status Command (7116)**

By writing the command code "7116" in the first bus cycle and then the block address of a flash memory block in the second bus cycle that follows, the system reads out the status of the lock bit of the specified block on to the data (D6).

Figure 30.7 shows an example of a read lock bit program flowchart.



**Figure 30.7 Read lock bit status flowchart**

### Data Protect Function (Block Lock)

Each block in Figure 29.3 has a nonvolatile lock bit to specify that the block be protected (locked) against erase/write. The lock bit program command is used to set the lock bit to 0 (locked). The lock bit of each block can be read out using the read lock bit status command.

Whether block lock is enabled or disabled is determined by the status of the lock bit and how the flash memory control register 0's lock bit disable bit is set.

- (1) When the lock bit disable bit = 0, a specified block can be locked or unlocked by the lock bit status (lock bit data). Blocks whose lock bit data = 0 are locked, so they are disabled against erase/write. On the other hand, the blocks whose lock bit data = 1 are not locked, so they are enabled for erase/write.
- (2) When the lock bit disable bit = 1, all blocks are nonlocked regardless of the lock bit data, so they are enabled for erase/write. In this case, the lock bit data that is 0 (locked) is set to 1 (nonlocked) after erasure, so that the lock bit-actuated lock is removed.

### Status Register

The status register indicates the operating status of the flash memory and whether an erase or program operation has terminated normally or in an error. The content of this register can be read out by only writing the read status register command (70<sub>16</sub>). Table 30.2 details the status register.

The status register is cleared by writing the Clear Status Register command (50<sub>16</sub>).

After a reset, the status register is set to "80<sub>16</sub>."

Each bit in this register is explained below.

#### Write state machine (WSM) status (SR7)

After power-on, the write state machine (WSM) status is set to 1.

The write state machine (WSM) status indicates the operating status of the device, as for output on the RY/ $\overline{\text{BY}}$  pin. This status bit is set to 0 during auto write or auto erase operation and is set to 1 upon completion of these operations.

#### Erase status (SR5)

The erase status informs the operating status of auto erase operation to the CPU. When an erase error occurs, it is set to 1.

The erase status is reset to 0 when cleared.

**Program status (SR4)**

The program status informs the operating status of auto write operation to the CPU. When a write error occurs, it is set to 1.

The program status is reset to 0 when cleared.

When an erase command is in error (which occurs if the command entered after the block erase command (20<sub>16</sub>) is not the confirmation command (D0<sub>16</sub>), both the program status and erase status (SR5) are set to 1.

When the program status or erase status = 1, the following commands entered by command write are not accepted.

Also, in one of the following cases, both SR4 and SR5 are set to 1 (command sequence error):

- (1) When the valid command is not entered correctly
- (2) When the data entered in the second bus cycle of lock bit program (77<sub>16</sub>/D0<sub>16</sub>), block erase (20<sub>16</sub>/D0<sub>16</sub>), or erase all unlock blocks (A7<sub>16</sub>/D0<sub>16</sub>) is not the D0<sub>16</sub> or FF<sub>16</sub>. However, if FF<sub>16</sub> is entered, read array is assumed and the command that has been set up in the first bus cycle is canceled.

**Block status after program (SR3)**

If excessive data is written (phenomenon whereby the memory cell becomes depressed which results in data not being read correctly), "1" is set for the program status after-program at the end of the page write operation. In other words, when writing ends successfully, "80<sub>16</sub>" is output; when writing fails, "90<sub>16</sub>" is output; and when excessive data is written, "88<sub>16</sub>" is output.

**Table 30.2 Definition of each bit in status register**

Each bit of SRD	Status name	Definition	
		"1"	"0"
SR7 (bit7)	Write state machine (WSM) status	Ready	Busy
SR6 (bit6)	Reserved	-	-
SR5 (bit5)	Erase status	Terminated in error	Terminated normally
SR4 (bit4)	Program status	Terminated in error	Terminated normally
SR3 (bit3)	Block status after program	Terminated in error	Terminated normally
SR2 (bit2)	Reserved	-	-
SR1 (bit1)	Reserved	-	-
SR0 (bit0)	Reserved	-	-

### Full Status Check

By performing full status check, it is possible to know the execution results of erase and program operations. Figure 30.8 shows a full status check flowchart and the action to be taken when each error occurs.

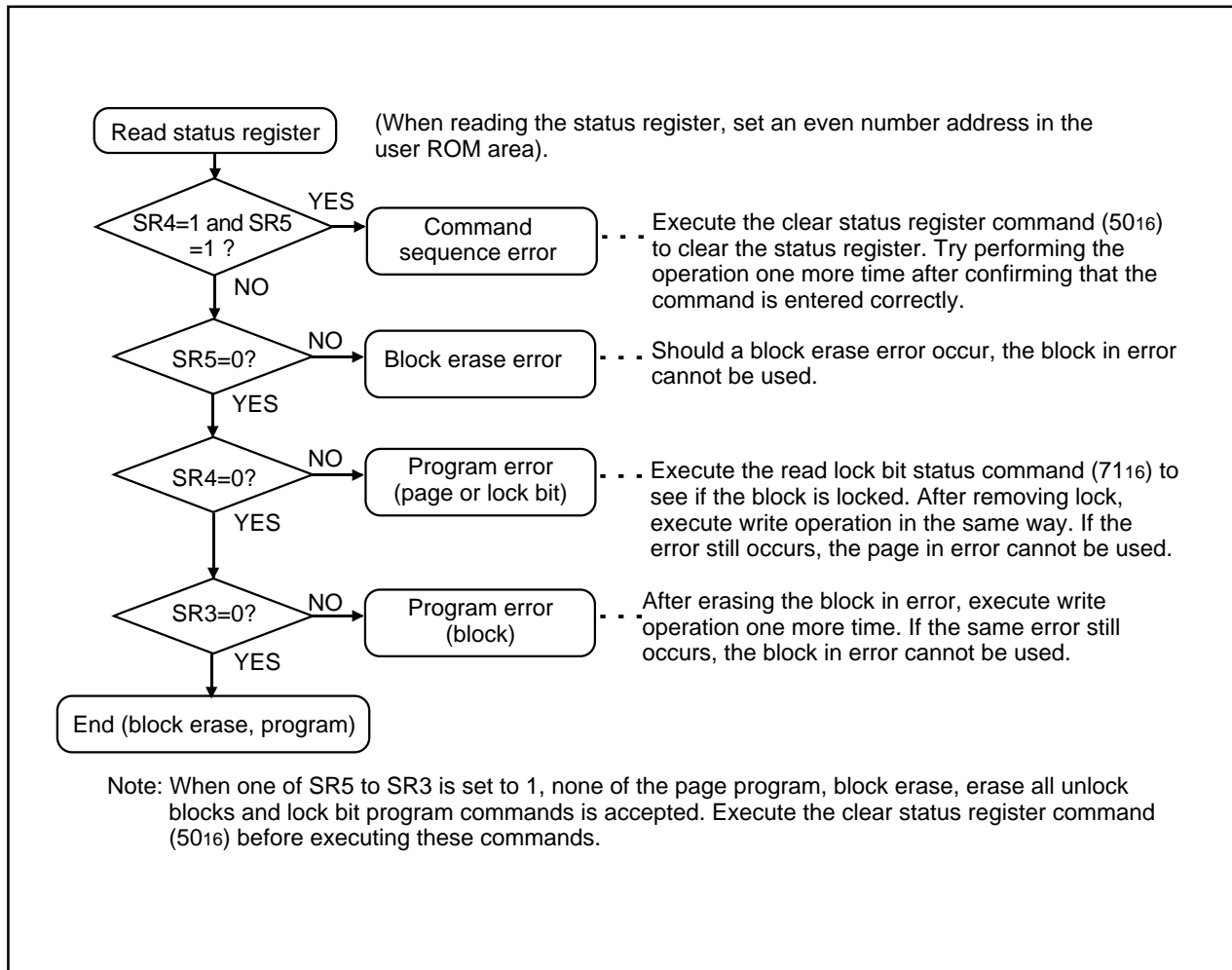


Figure 30.8 Full status check flowchart and remedial procedure for errors

## Functions To Prevent the Flash Memory from Rewriting

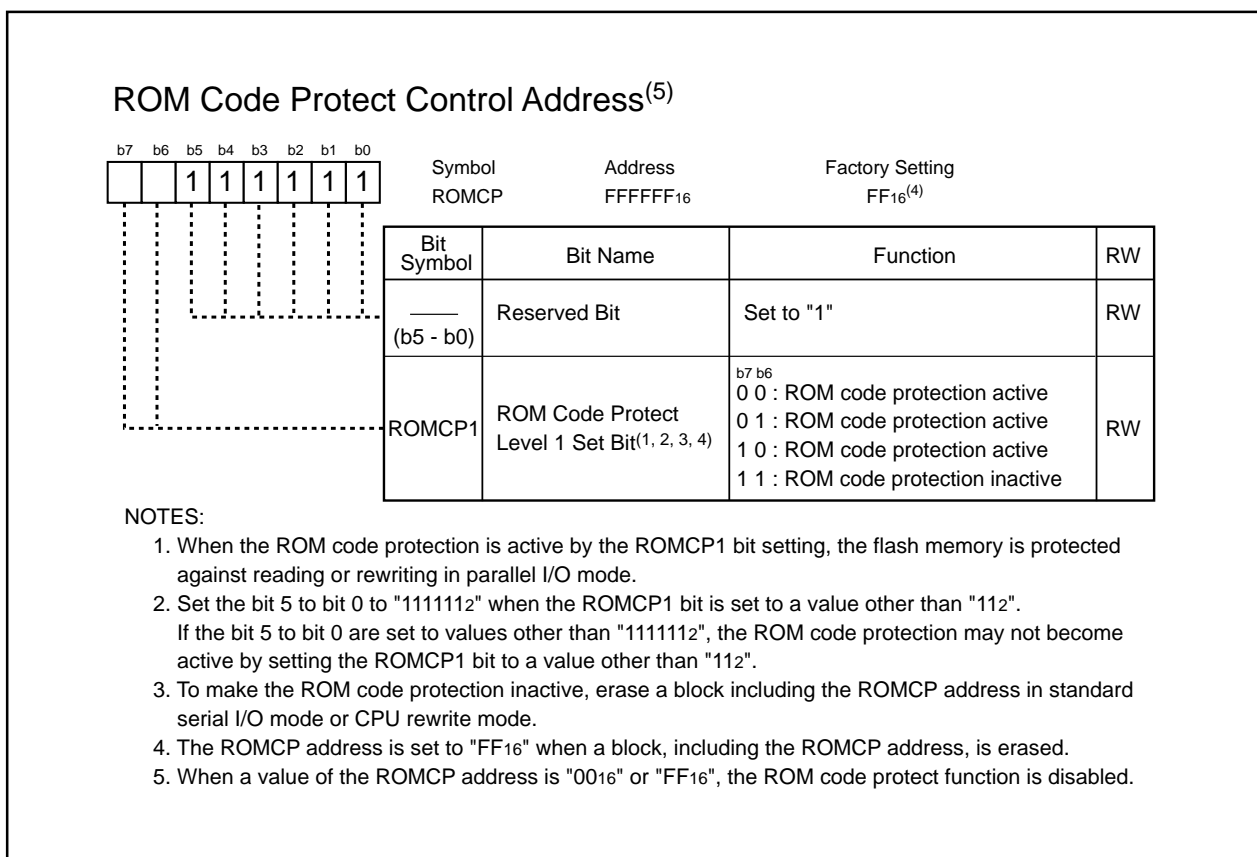
To prevent the contents of the flash memory version from being read out or rewritten easily, the device incorporates a ROM code protect function for use in parallel I/O mode and an ID code verify function for use in standard serial I/O mode.

### ROM code protect function

The ROM code protect function reading out or modifying the contents of the flash memory version by using the ROM code protect control address (0FFFFFFF<sub>16</sub>) during parallel I/O mode. Figure 30.9 shows the ROM code protect control address (0FFFFFFF<sub>16</sub>). (This address exists in the user ROM area.)

If one of the pair of ROM code protect bits is set to 0, ROM code protect is turned on, so that the contents of the flash memory version are protected against readout and modification.

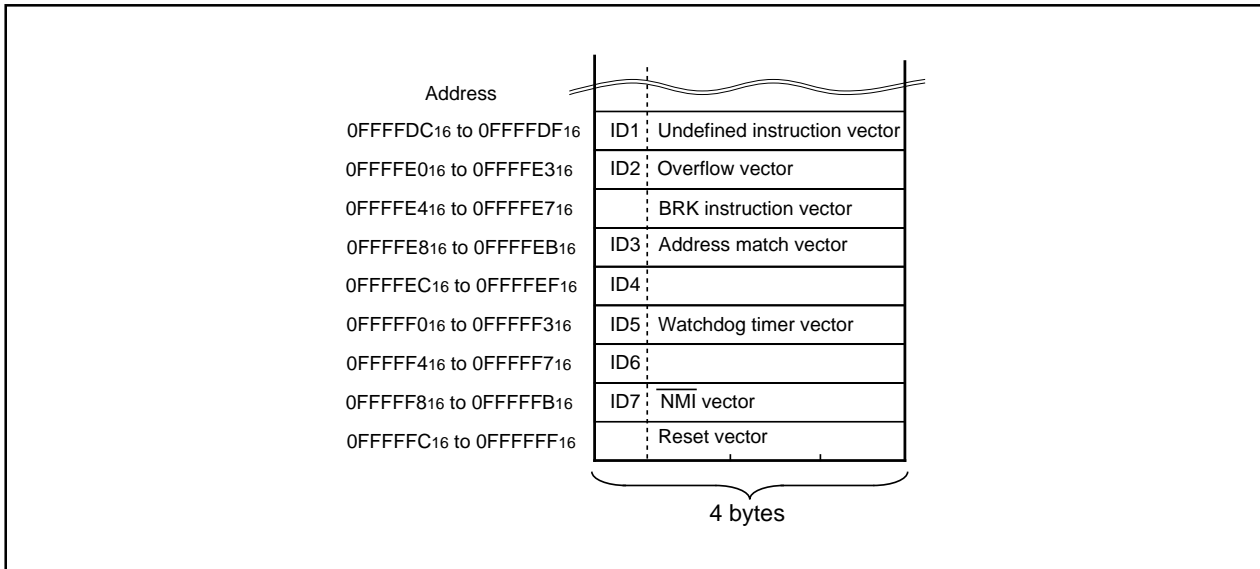
If both of the two ROM code protect reset bits are set to "00," ROM code protect is turned off, so that the contents of the flash memory version can be read out or modified. Once ROM code protect is turned on, the contents of the ROM code protect reset bits cannot be modified in parallel I/O mode. Use the serial I/O or some other mode to rewrite the contents of the ROM code protect reset bits.



**Figure 30.9 ROM code protect control address**

## ID Code Verify Function

Use this function in standard serial I/O mode. When the contents of the flash memory are not blank, the ID code sent from the peripheral unit is compared with the ID code written in the flash memory to see if they match. If the ID codes do not match, the commands sent from the peripheral unit are not accepted. The ID code consists of 8-bit data, the areas of which, beginning with the first byte, are 0FFFFDF<sub>16</sub>, 0FFFFE3<sub>16</sub>, 0FFFFE8<sub>16</sub>, 0FFFFEF<sub>16</sub>, 0FFFFF3<sub>16</sub>, 0FFFFF7<sub>16</sub>, and 0FFFFFB<sub>16</sub>. Write a program which has had the ID code preset at these addresses to the flash memory.



**Figure 30.10** ID code store addresses

### 31. Parallel I/O Mode

Use an exclusive programmer supporting M16C/80 (flash memory version).

Refer to the instruction manual of each programmer maker for the details of use.

#### User ROM and Boot ROM Areas

In parallel I/O mode, the user ROM and boot ROM areas shown in Figure 29.3 can be rewritten. Both areas of flash memory can be operated on in the same way.

Program and block erase operations can be performed in the user ROM area. The user ROM area and its blocks are shown in Figure 29.3.

The boot ROM area is 8 Kbytes in size. In parallel I/O mode, it is located at addresses 0FFE000<sub>16</sub> through 0FFFFFF<sub>16</sub>. Make sure program and block erase operations are always performed within this address range. (Access to any location outside this address range is prohibited.)

In the boot ROM area, an erase block operation is applied to only one 8 Kbyte block. The boot ROM area has had a standard serial I/O mode control program stored in it when shipped from the factory.

Therefore, using the device in standard serial input/output mode, you do not need to write to the boot ROM area.

**Pin functions (Flash memory standard serial I/O mode)**

Pin	Name	I/O	Description
Vcc, Vss	Power input		Apply 4.2V to 5.5V to Vcc pin and 0 V to Vss pin.
CNVss	CNVss	I	Connect to Vcc pin.
RESET	Reset input	I	Reset input pin. While reset is "L" level, a 20 cycle or longer clock must be input to XIN pin.
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	
BYTE	BYTE	I	Connect this pin to Vcc or Vss.
AVcc, AVss	Analog power supply input	I	Connect AVSS to Vss and AVcc to Vcc, respectively.
VREF	Reference voltage input	I	Enter the reference voltage for A/D converter from this pin.
P00 to P07	Input port P0	I	Input "H" or "L" level signal or open.
P10 to P17	Input port P1	I	Input "H" or "L" level signal or open.
P20 to P27	Input port P2	I	Input "H" or "L" level signal or open.
P30 to P37	Input port P3	I	Input "H" or "L" level signal or open.
P40 to P47	Input port P4	I	Input "H" or "L" level signal or open.
P51 to P54, P56, P57	Input port P5	I	Input "H" or "L" level signal or open.
P50	CE input	I	Input "H" level signal.
P55	EPM input	I	Input "L" level signal.
P60 to P63	Input port P6	I	Input "H" or "L" level signal or open.
P64	BUSY output	O	Standard serial mode 1: BUSY signal output pin Standard serial mode 2: Monitors the program operation check
P65	SCLK input	I	Standard serial mode 1: Serial clock input pin Standard serial mode 2: Input "L" level signal.
P66	RxD input	I	Serial data input pin
P67	TxD output	O	Serial data output pin
P70 to P77	Input port P7	I	Input "H" or "L" level signal or open.
P80 to P84, P86, P87	Input port P8	I	Input "H" or "L" level signal or open.
P85	NMI input	I	Connect this pin to Vcc.
P90 to P97	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.
P110 to P114	Input port P11	I	Input "H" or "L" level signal or open. (Note)
P120 to P127	Input port P12	I	Input "H" or "L" level signal or open. (Note)
P130 to P137	Input port P13	I	Input "H" or "L" level signal or open. (Note)
P140 to P146	Input port P14	I	Input "H" or "L" level signal or open. (Note)
P150 to P157	Input port P15	I	Input "H" or "L" level signal or open. (Note)

Note: Port P11 to P15 exist in 144-pin version.





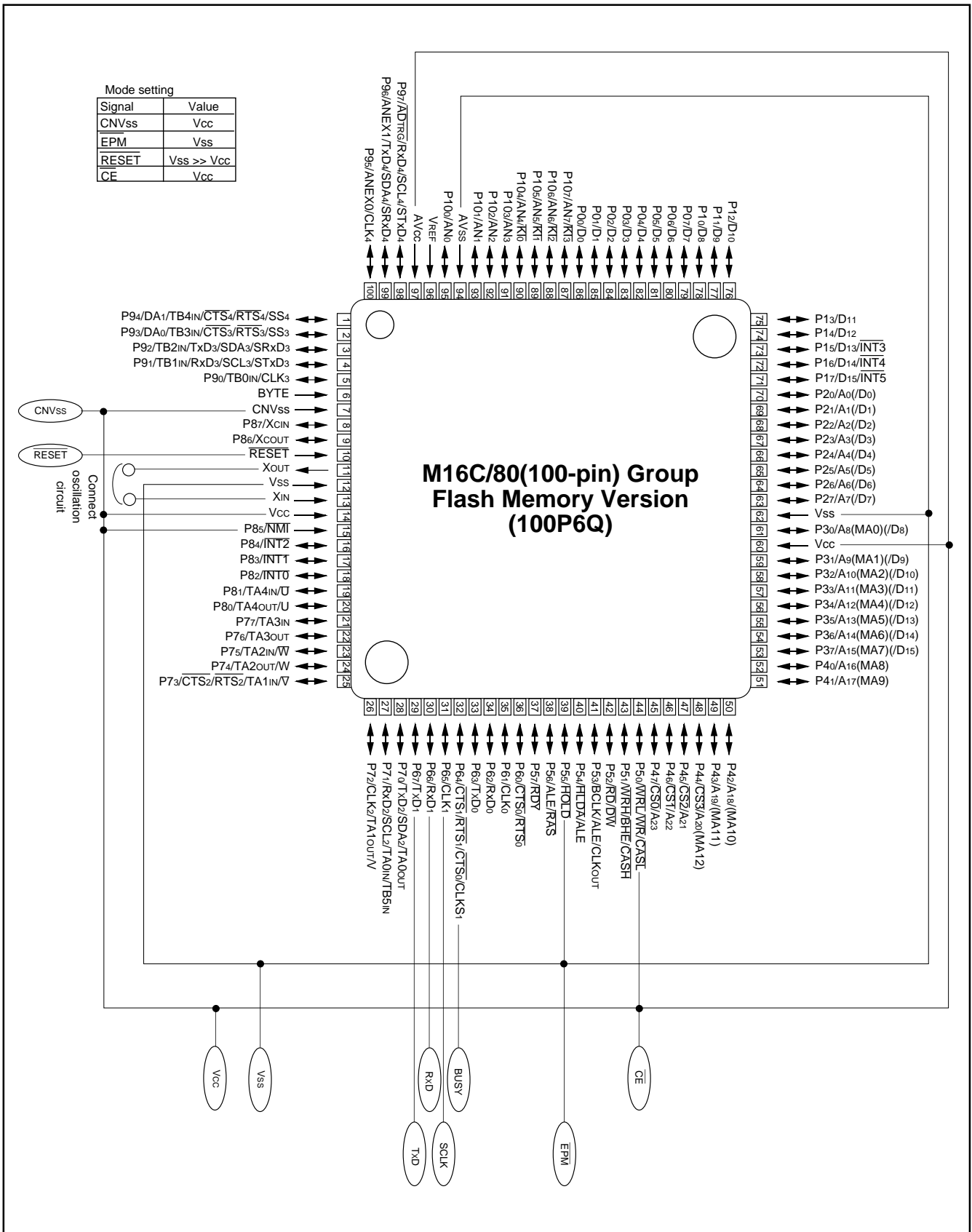


Figure 31.2 Pin connections for standard serial I/O mode (2)

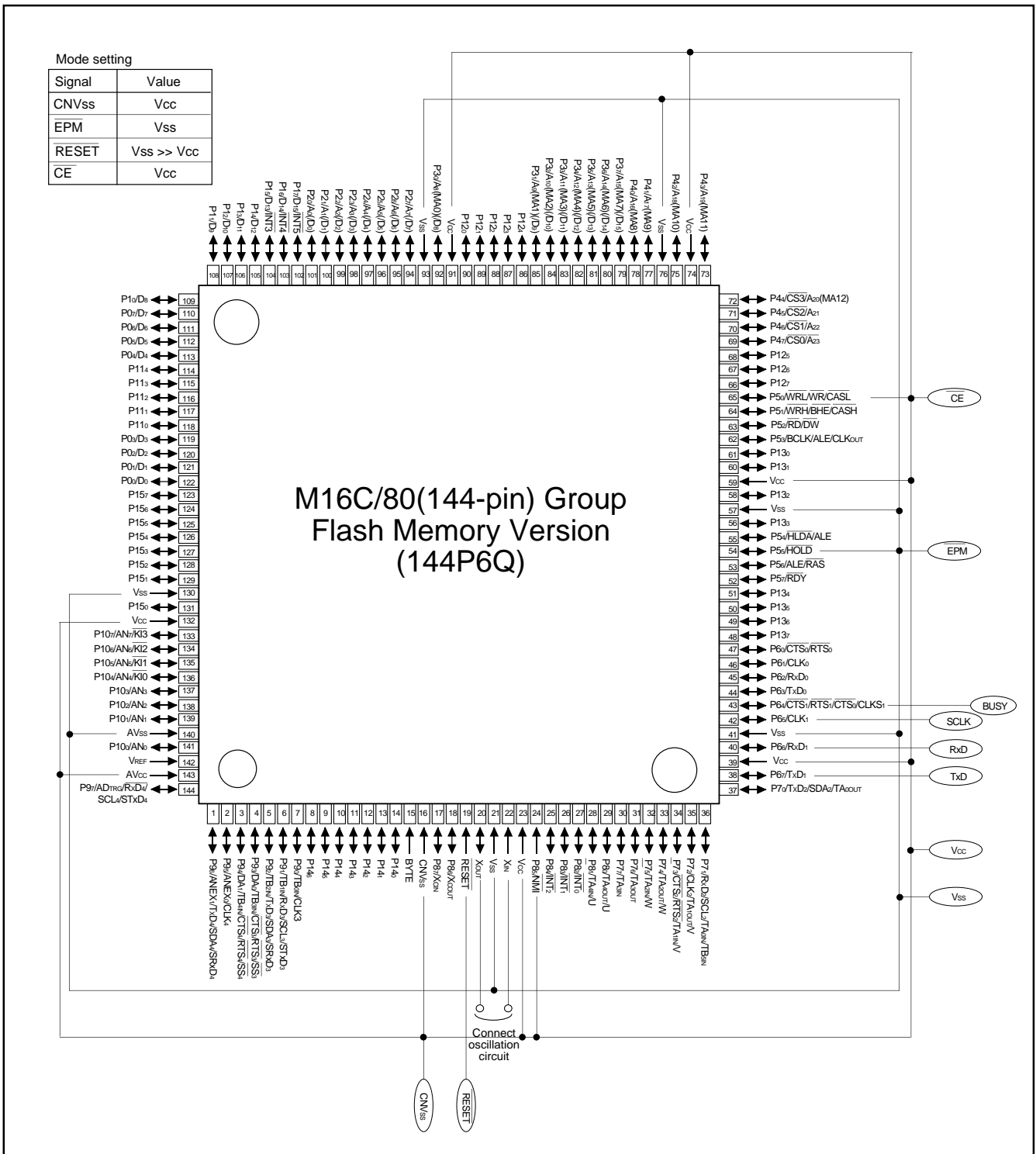


Figure 31.3 Pin connections for standard serial I/O mode (3)

### 32. Standard serial I/O mode

The standard serial I/O mode inputs and outputs the software commands, addresses and data needed to operate (read, program, erase, etc.) the internal flash memory. This I/O is serial. There are actually two standard serial I/O modes: mode 1, which is clock synchronized, and mode 2, which is asynchronous. Both modes require a purpose-specific peripheral unit.

The standard serial I/O mode is different from the parallel I/O mode in that the CPU controls flash memory rewrite (uses the CPU's rewrite mode), rewrite data input and so forth. It is started when the reset is released, which is done when the P50 ( $\overline{CE}$ ) pin is "H" level, the P55 ( $\overline{EPM}$ ) pin "L" level and the CNVss pin "H" level. (In the ordinary command mode, set CNVss pin to "L" level.)

This control program is written in the boot ROM area when the product is shipped from the factory. Accordingly, make note of the fact that the standard serial I/O mode cannot be used if the boot ROM area is rewritten in the parallel I/O mode. Figures 31.1 and 31.3 show the pin connections for the standard serial I/O mode. Serial data I/O uses UART1 and transfers the data serially in 8-bit units. Standard serial I/O switches between mode 1 (clock synchronized) and mode 2 (clock asynchronous) according to the level of CLK1 pin when the reset is released.

To use standard serial I/O mode 1 (clock synchronized), set the CLK1 pin to "H" level and release the reset. The operation uses the four UART1 pins CLK1, RxD1, TxD1 and RTS1 (BUSY). The CLK1 pin is the transfer clock input pin through which an external transfer clock is input. The TxD1 pin is for CMOS output. The RTS1 (BUSY) pin outputs an "L" level when ready for reception and an "H" level when reception starts.

To use standard serial I/O mode 2 (clock asynchronous), set the CLK1 pin to "L" level and release the reset. The operation uses the two UART1 pins RxD1 and TxD1.

In the standard serial I/O mode, only the user ROM area indicated in Figure 32.17 can be rewritten. The boot ROM cannot.

In the standard serial I/O mode, a 7-byte ID code is used. When there is data in the flash memory, commands sent from the peripheral unit (programmer) are not accepted unless the ID code matches.

### 32.1 Overview of standard serial I/O mode 1 (clock synchronized)

In standard serial I/O mode 1, software commands, addresses and data are input and output between the MCU and peripheral units (serial programmer, etc.) using 4-wire clock-synchronized serial I/O (UART1). Standard serial I/O mode 1 is engaged by releasing the reset with the P65 (CLK1) pin "H" level.

In reception, software commands, addresses and program data are synchronized with the rise of the transfer clock that is input to the CLK1 pin, and are then input to the MCU via the RxD1 pin. In transmission, the read data and status are synchronized with the fall of the transfer clock, and output from the TxD1 pin.

The TxD1 pin is for CMOS output. Transfer is in 8-bit units with LSB first.

When busy, such as during transmission, reception, erasing or program execution, the RTS1 (BUSY) pin is "H" level. Accordingly, always start the next transfer after the RST1 (BUSY) pin is "L" level.

Also, data and status registers in memory can be read after inputting software commands. Status, such as the operating state of the flash memory or whether a program or erase operation ended successfully or not, can be checked by reading the status register. Here following are explained software commands, status registers, etc.

## Software Commands

Table 31.1 lists software commands. In the standard serial I/O mode 1, erase operations, programs and reading are controlled by transferring software commands via the RxD1 pin. Software commands are explained here below.

**Table 32.1 Software commands (Standard serial I/O mode 1)**

	Control command	1st byte transfer	2nd byte	3rd byte	4th byte	5th byte	6th byte		When ID is not verified
1	Page read	FF <sub>16</sub>	Address (middle)	Address (high)	Data output	Data output	Data output	Data output to 259th byte	Not acceptable
2	Page program	41 <sub>16</sub>	Address (middle)	Address (high)	Data input	Data input	Data input	Data input to 259th byte	Not acceptable
3	Block erase	20 <sub>16</sub>	Address (middle)	Address (high)	D0 <sub>16</sub>				Not acceptable
4	Erase all unlocked blocks	A7 <sub>16</sub>	D0 <sub>16</sub>						Not acceptable
5	Read status register	70 <sub>16</sub>	SRD output	SRD1 output					Acceptable
6	Clear status register	50 <sub>16</sub>							Not acceptable
7	Read lock bit status	71 <sub>16</sub>	Address (middle)	Address (high)	Lock bit data output				Not acceptable
8	Lock bit program	77 <sub>16</sub>	Address (middle)	Address (high)	D0 <sub>16</sub>				Not acceptable
9	Lock bit enable	7A <sub>16</sub>							Not acceptable
10	Lock bit disable	75 <sub>16</sub>							Not acceptable
11	Code processing function	F5 <sub>16</sub>	Address (low)	Address (middle)	Address (high)	ID size	ID1	To ID7	Acceptable
12	Download function	FA <sub>16</sub>	Size (low)	Size (high)	Check-sum	Data input	To required number of times		Not acceptable
13	Version data output function	FB <sub>16</sub>	Version data output	Version data output	Version data output	Version data output	Version data output	Version data output to 9th byte	Acceptable
14	Boot ROM area output function	FC <sub>16</sub>	Address (middle)	Address (high)	Data output	Data output	Data output	Data output to 259th byte	Not acceptable
15	Read check data	FD <sub>16</sub>	Check data (low)	Check data (high)					Not acceptable

Note 1: Shading indicates transfer from flash memory microcomputer to peripheral unit. All other data is transferred from the peripheral unit to the flash memory microcomputer.

Note 2: SRD refers to status register data. SRD1 refers to status register data1 .

Note 3: All commands can be accepted when the flash memory is totally blank.

**Page Read Command**

This command reads the specified page (256 bytes) in the flash memory sequentially one byte at a time. Execute the page read command as explained here following.

- (1) Transfer the “FF16” command code with the 1st byte.
- (2) Transfer addresses A8 to A15 and A16 to A23 with the 2nd and 3rd bytes respectively.
- (3) From the 4th byte onward, data (D0–D7) for the page (256 bytes) specified with addresses A8 to A23 will be output sequentially from the smallest address first in sync with the rise of the clock.

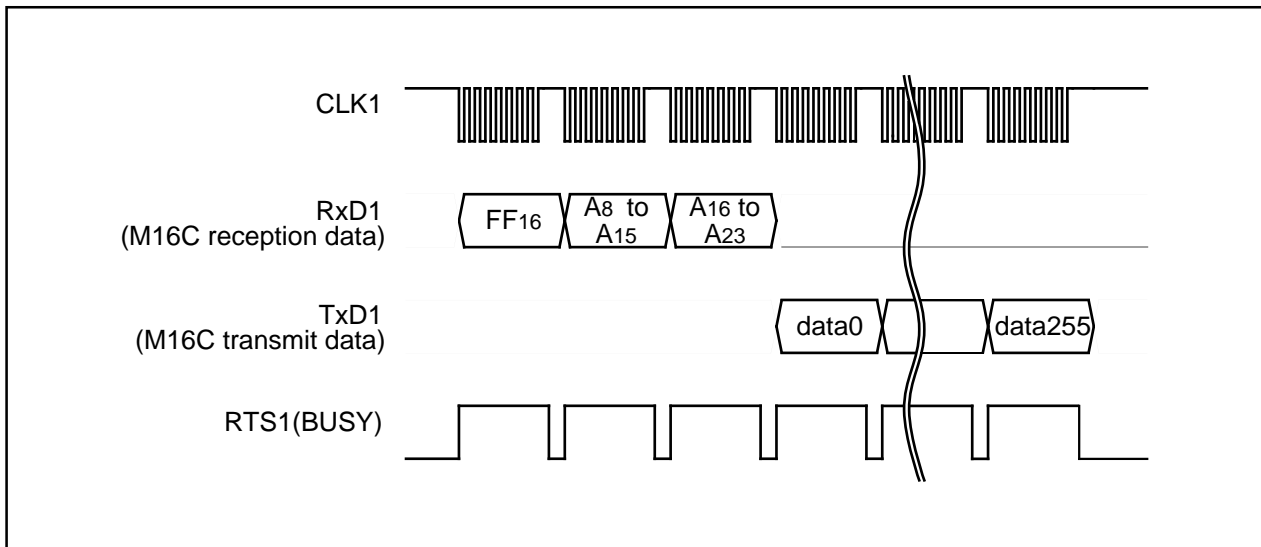


Figure 32.1 Timing for page read

**Read Status Register Command**

This command reads status information. When the “7016” command code is sent with the 1st byte, the contents of the status register (SRD) specified with the 2nd byte and the contents of status register 1 (SRD1) specified with the 3rd byte are read.

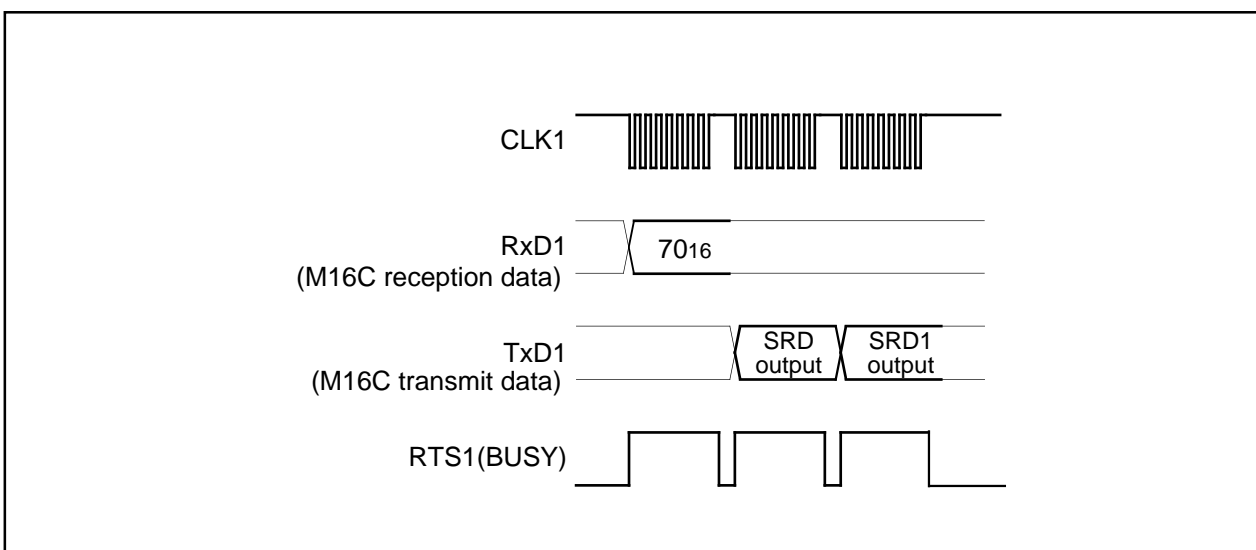
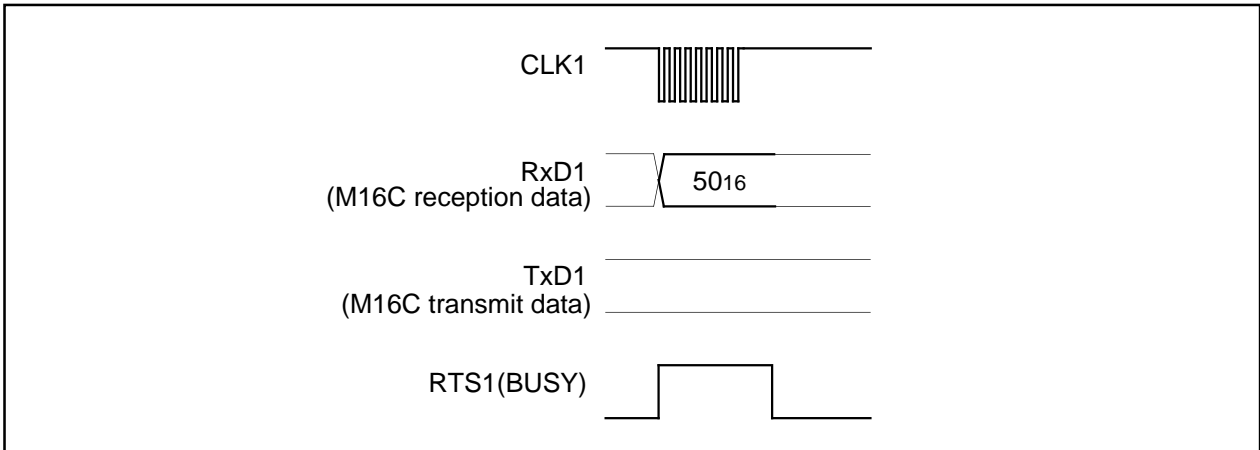


Figure 32.2 Timing for reading the status register

**Clear Status Register Command**

This command clears the bits (SR3–SR5) which are set when the status register operation ends in error. When the “5016” command code is sent with the 1st byte, the aforementioned bits are cleared. When the clear status register operation ends, the RTS1 (BUSY) signal changes from the “H” to the “L” level.



**Figure 32.3 Timing for clearing the status register**

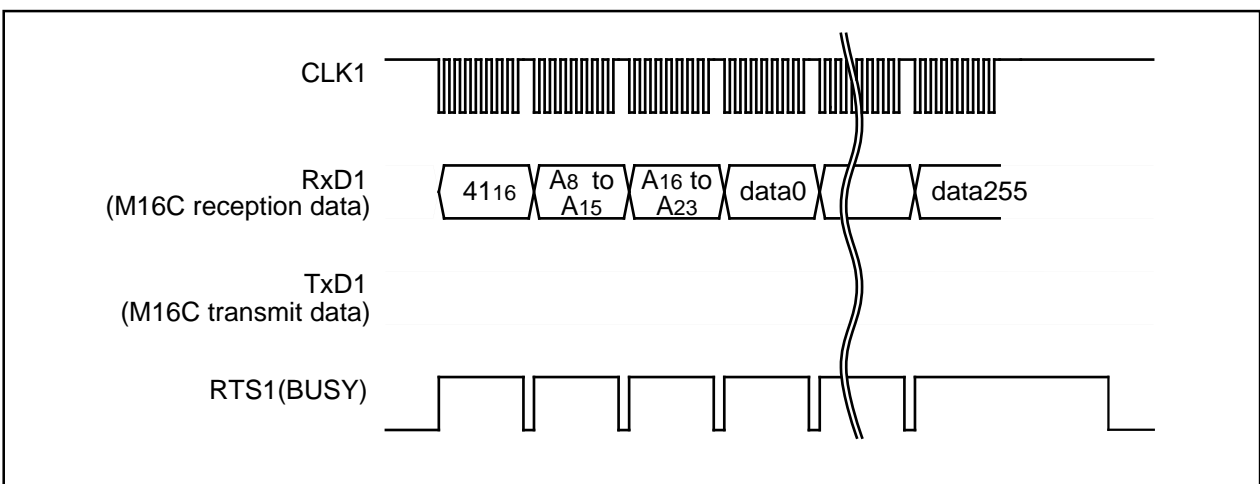
**Page Program Command**

This command writes the specified page (256 bytes) in the flash memory sequentially one byte at a time. Execute the page program command as explained here following.

- (1) Transfer the “4116” command code with the 1st byte.
- (2) Transfer addresses A8 to A15 and A16 to A23 with the 2nd and 3rd bytes respectively.
- (3) From the 4th byte onward, as write data (D0–D7) for the page (256 bytes) specified with addresses A8 to A23 is input sequentially from the smallest address first, that page is automatically written.

When reception setup for the next 256 bytes ends, the RTS1 (BUSY) signal changes from the “H” to the “L” level. The result of the page program can be known by reading the status register. For more information, see the section on the status register.

Each block can be write-protected with the lock bit. For more information, see the section on the data protection function. Additional writing is not allowed with already programmed pages.



**Figure 32.4 Timing for the page program**



### Block Erase Command

This command erases the data in the specified block. Execute the block erase command as explained here following.

- (1) Transfer the "2016" command code with the 1st byte.
- (2) Transfer addresses A8 to A15 and A16 to A23 with the 2nd and 3rd bytes respectively.
- (3) Transfer the verify command code "D016" with the 4th byte. With the verify command code, the erase operation will start for the specified block in the flash memory. Write the highest address of the specified block for addresses A16 to A23.

When block erasing ends, the RTS1 (BUSY) signal changes from the "H" to the "L" level. After block erase ends, the result of the block erase operation can be known by reading the status register. For more information, see the section on the status register.

Each block can be erase-protected with the lock bit. For more information, see the section on the data protection function.

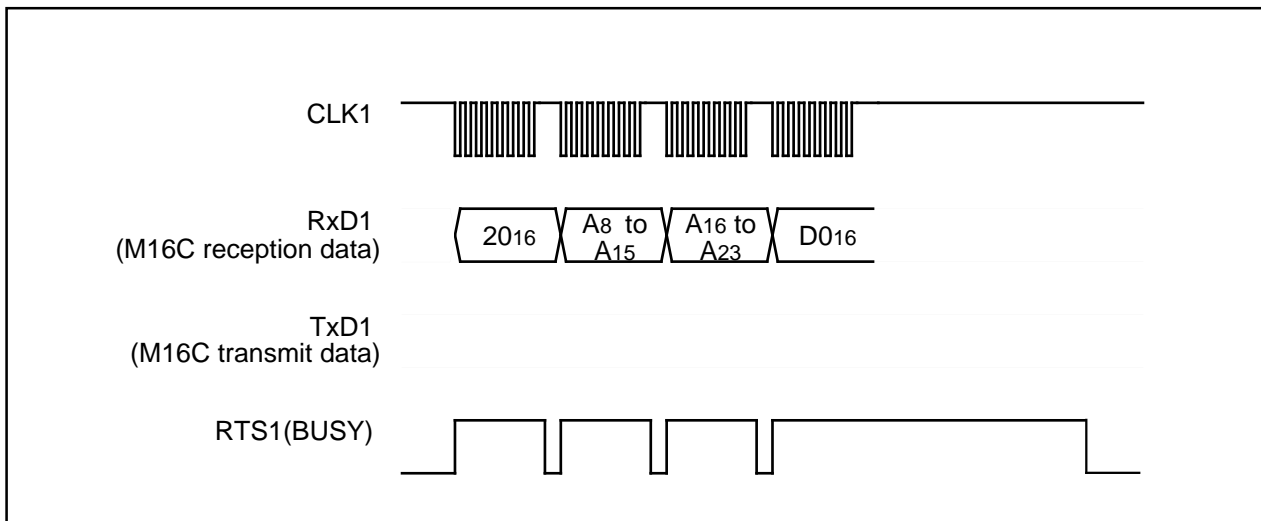


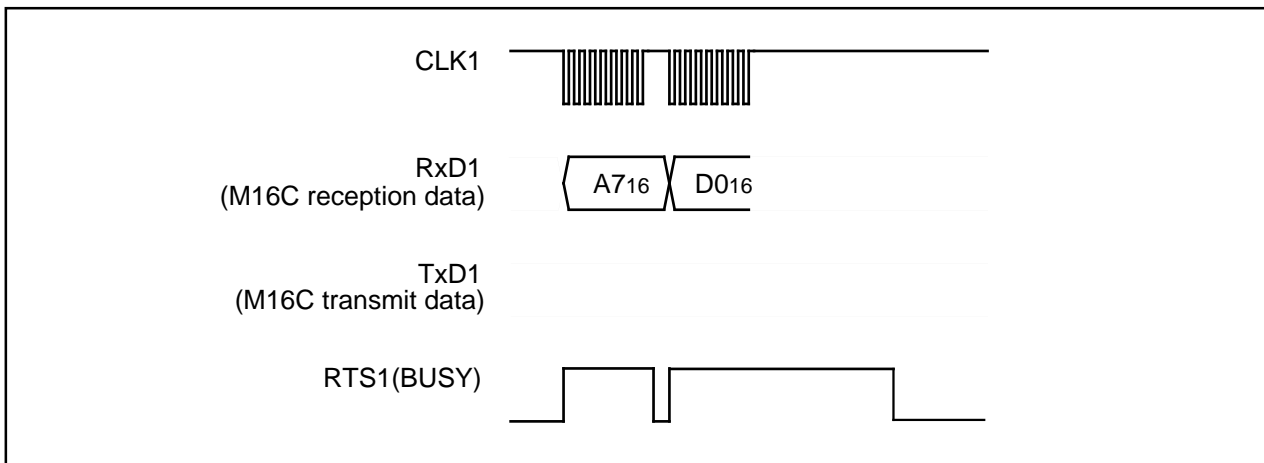
Figure 32.5 Timing for block erasing

**Erase All Unlocked Blocks Command**

This command erases the content of all blocks. Execute the erase all unlocked blocks command as explained here following.

- (1) Transfer the "A716" command code with the 1st byte.
- (2) Transfer the verify command code "D016" with the 2nd byte. With the verify command code, the erase operation will start and continue for all blocks in the flash memory.

When block erasing ends, the RTS1 (BUSY) signal changes from the "H" to the "L" level. The result of the erase operation can be known by reading the status register. Each block can be erase-protected with the lock bit. For more information, see the section on the data protection function.



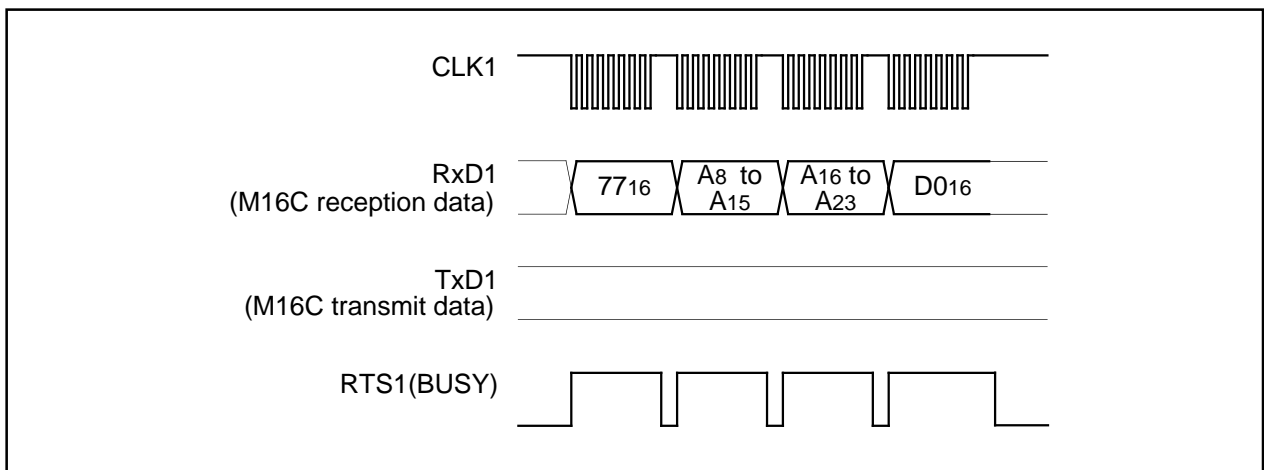
**Figure 32.6 Timing for erasing all unlocked blocks**

**Lock Bit Program Command**

This command writes "0" (lock) for the lock bit of the specified block. Execute the lock bit program command as explained here following.

- (1) Transfer the "7716" command code with the 1st byte.
- (2) Transfer addresses A8 to A15 and A16 to A23 with the 2nd and 3rd bytes respectively.
- (3) Transfer the verify command code "D016" with the 4th byte. With the verify command code, "0" is written for the lock bit of the specified block. Write the highest address of the specified block for addresses A8 to A23.

When writing ends, the RTS1 (BUSY) signal changes from the "H" to the "L" level. Lock bit status can be read with the read lock bit status command. For information on the lock bit function, reset procedure and so on, see the section on the data protection function.

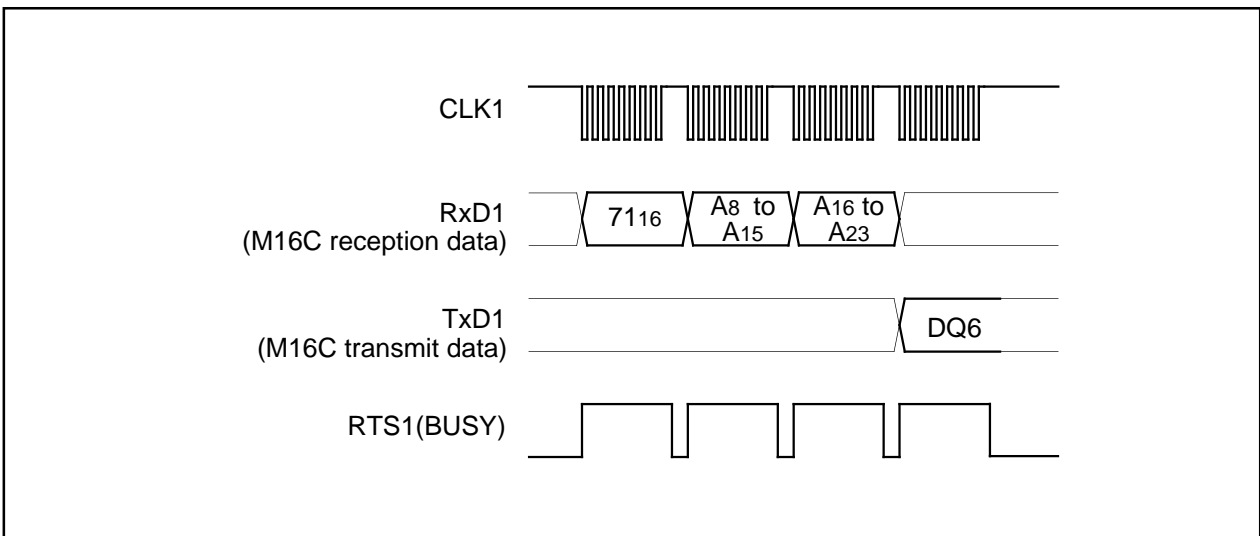


**Figure 32.7 Timing for the lock bit program**

**Read Lock Bit Status Command**

This command reads the lock bit status of the specified block. Execute the read lock bit status command as explained here following.

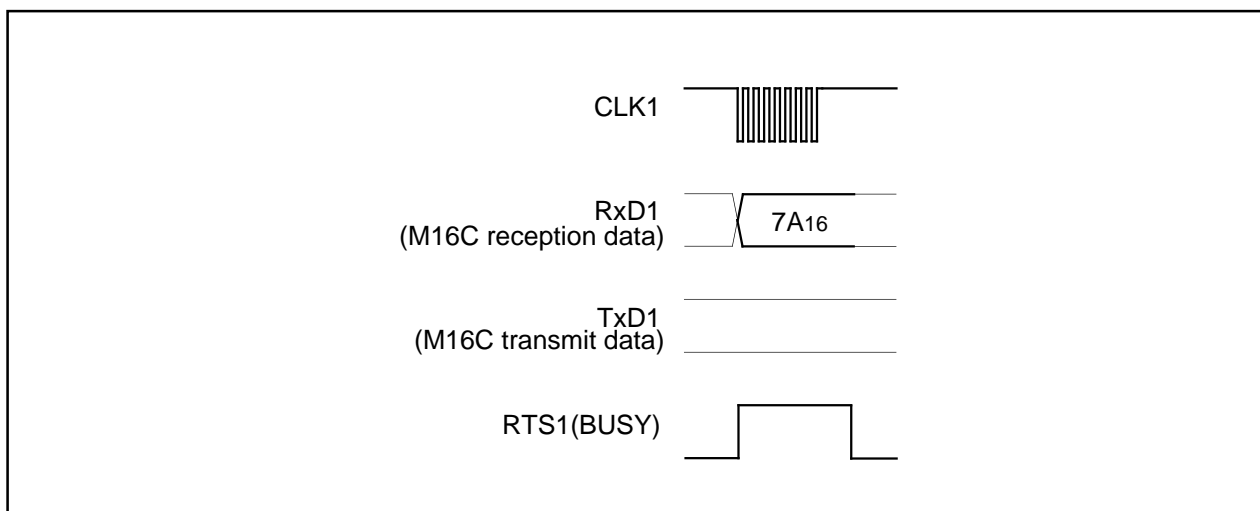
- (1) Transfer the “7116” command code with the 1st byte.
- (2) Transfer addresses A8 to A15 and A16 to A23 with the 2nd and 3rd bytes respectively.
- (3) The lock bit data of the specified block is output with the 4th byte. The 6th bit (D6) of output data is the lock bit data. Write the highest address of the specified block for addresses A8 to A23.



**Figure 32.8 Timing for reading lock bit status**

**Lock Bit Enable Command**

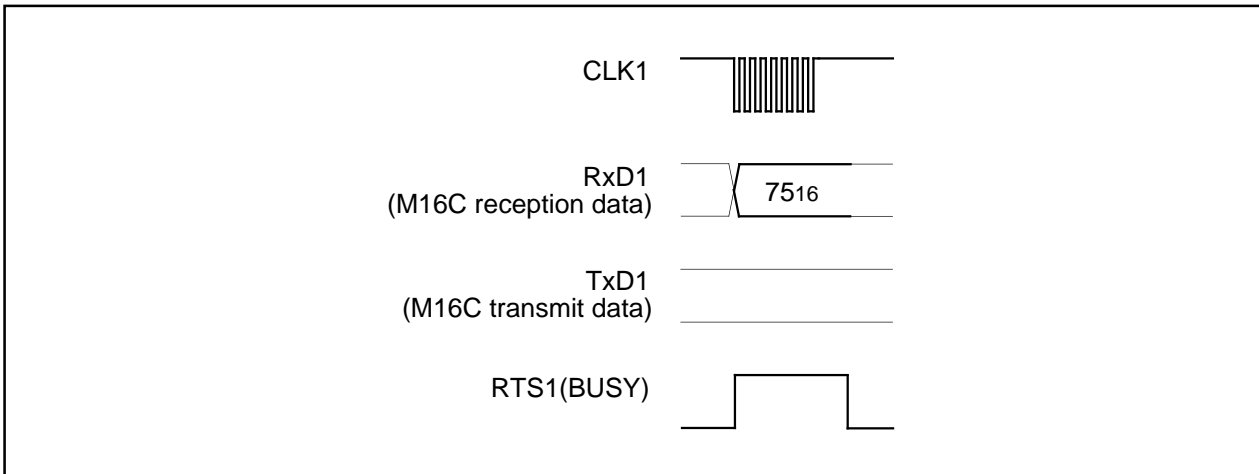
This command enables the lock bit in blocks whose bit was disabled with the lock bit disable command. The command code “7A16” is sent with the 1st byte of the serial transmission. This command only enables the lock bit function; it does not set the lock bit itself.



**Figure 32.9 Timing for enabling the lock bit**

**Lock Bit Disable Command**

This command disables the lock bit. The command code “7516” is sent with the 1st byte of the serial transmission. This command only disables the lock bit function; it does not set the lock bit itself. However, if an erase command is executed after executing the lock bit disable command, “0” (locked) lock bit data is set to “1” (unlocked) after the erase operation ends. In any case, after the reset is cancelled, the lock bit is enabled.



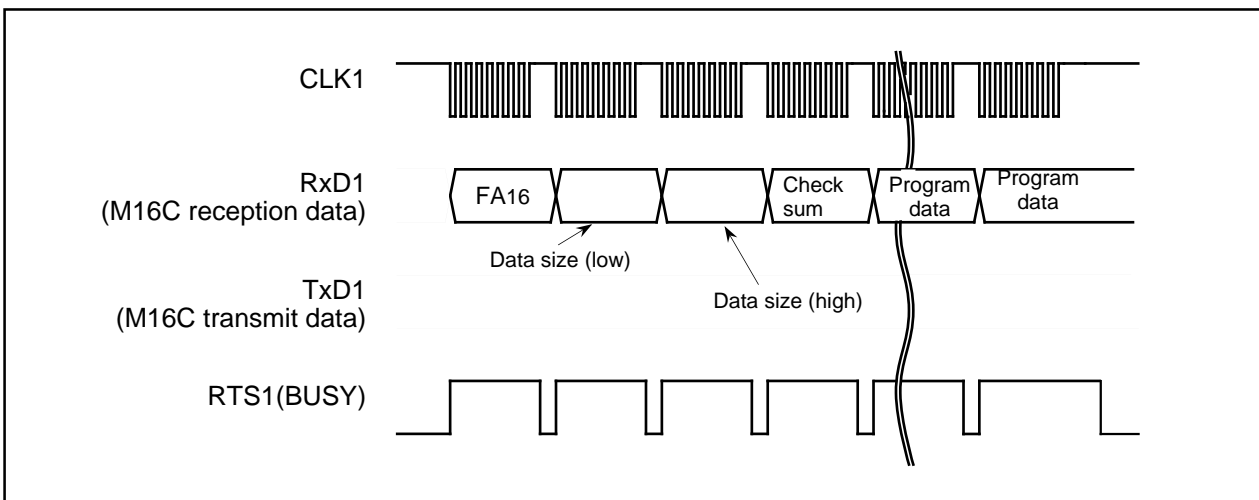
**Figure 32.10 Timing for disabling the lock bit**

**Download Command**

This command downloads a program to the RAM for execution. Execute the download command as explained here following.

- (1) Transfer the “FA16” command code with the 1st byte.
- (2) Transfer the program size with the 2nd and 3rd bytes.
- (3) Transfer the check sum with the 4th byte. The check sum is added to all data sent with the 5th byte onward.
- (4) The program to execute is sent with the 5th byte onward.

When all data has been transmitted, if the check sum matches, the downloaded program is executed. The size of the program will vary according to the internal RAM.

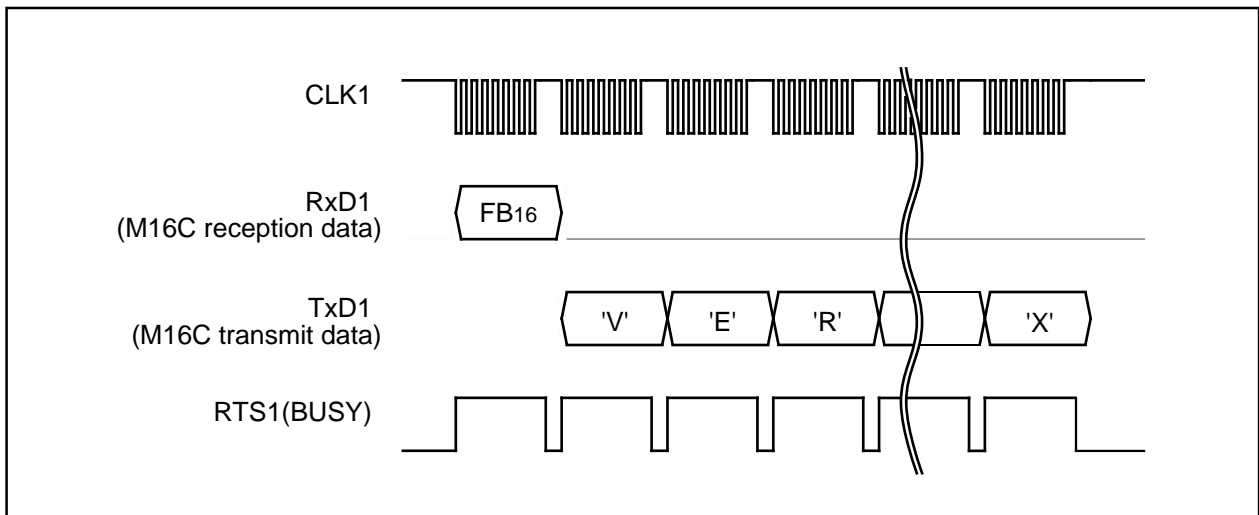


**Figure 32.11 Timing for download**

**Version Information Output Command**

This command outputs the version information of the control program stored in the boot area. Execute the version information output command as explained here following.

- (1) Transfer the "FB16" command code with the 1st byte.
- (2) The version information will be output from the 2nd byte onward. This data is composed of 8 ASCII code characters.

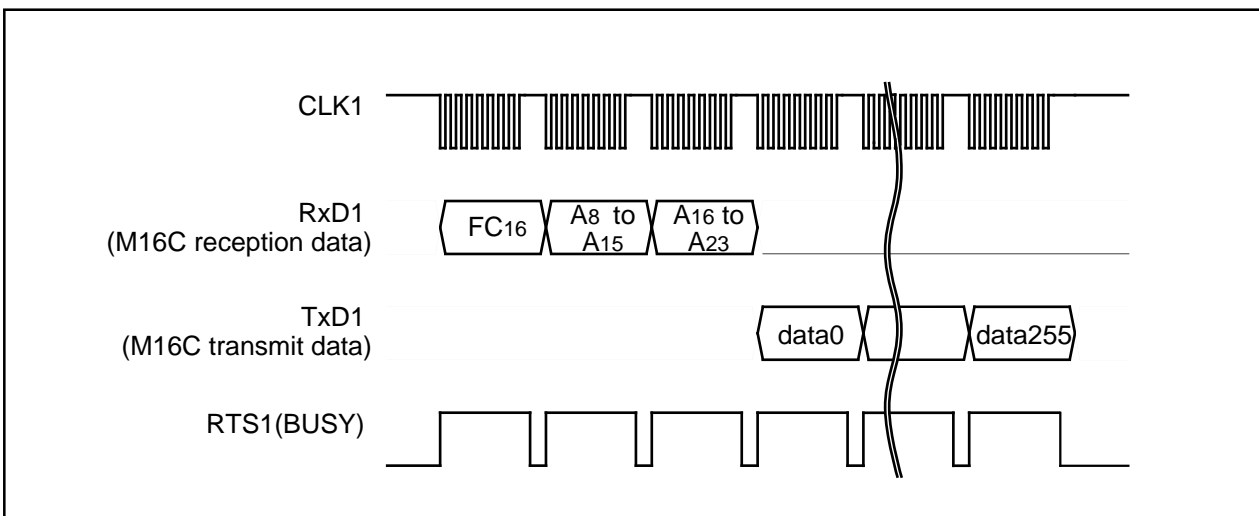


**Figure 32.12 Timing for version information output**

**Boot ROM Area Output Command**

This command outputs the control program stored in the boot ROM area in one page blocks (256 bytes). Execute the boot ROM area output command as explained here following.

- (1) Transfer the "FC16" command code with the 1st byte.
- (2) Transfer addresses A8 to A15 and A16 to A23 with the 2nd and 3rd bytes respectively.
- (3) From the 4th byte onward, data (D0–D7) for the page (256 bytes) specified with addresses A8 to A23 will be output sequentially from the smallest address first, in sync with the rise of the clock.

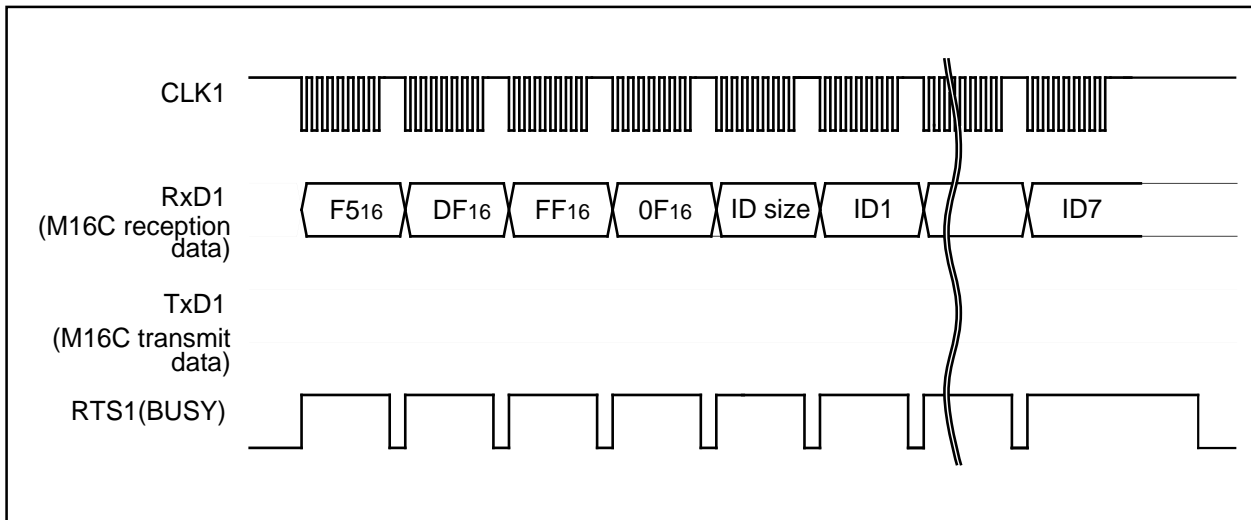


**Figure 32.13 Timing for boot ROM area output**

**ID Check**

This command checks the ID code. Execute the boot ID check command as explained here following.

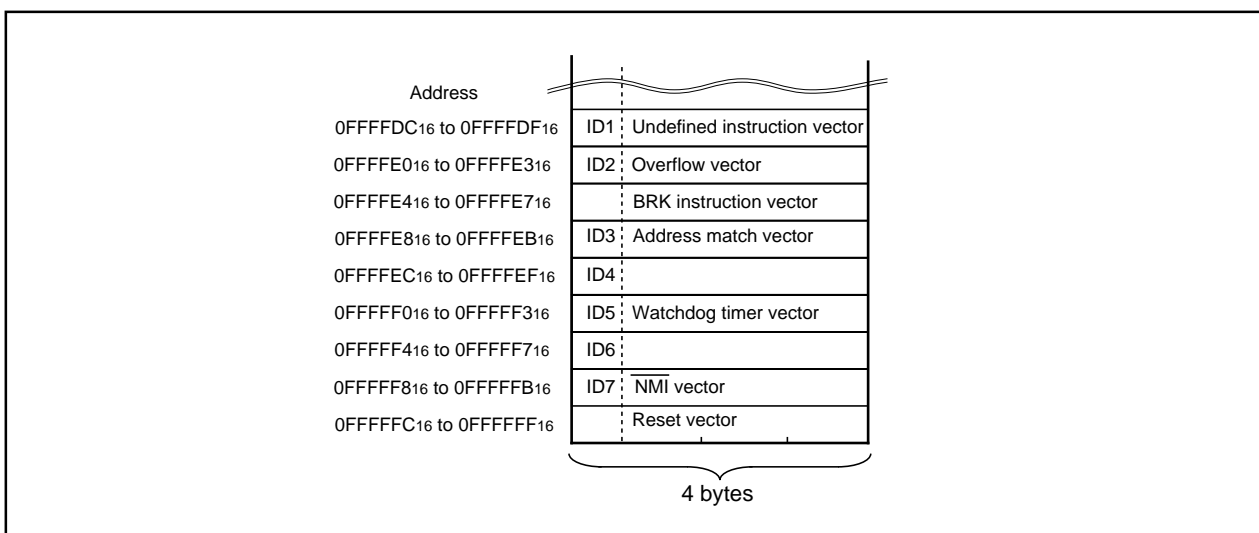
- (1) Transfer the "F5<sub>16</sub>" command code with the 1st byte.
- (2) Transfer addresses A<sub>0</sub> to A<sub>7</sub>, A<sub>8</sub> to A<sub>15</sub> and A<sub>16</sub> to A<sub>23</sub> of the 1st byte of the ID code with the 2nd, 3rd and 4th bytes respectively.
- (3) Transfer the number of data sets of the ID code with the 5th byte.
- (4) The ID code is sent with the 6th byte onward, starting with the 1st byte of the code.



**Figure 32.14 Timing for the ID check**

**ID Code**

When the flash memory is not blank, the ID code sent from the peripheral units and the ID code written in the flash memory are compared to see if they match. If the codes do not match, the command sent from the peripheral units is not accepted. An ID code contains 8 bits of data. Area is, from the 1st byte, addresses 0FFFFDF<sub>16</sub>, 0FFFFE3<sub>16</sub>, 0FFFFEB<sub>16</sub>, 0FFFFEF<sub>16</sub>, 0FFFFF3<sub>16</sub>, 0FFFFF7<sub>16</sub> and 0FFFFFB<sub>16</sub>. Write a program into the flash memory, which already has the ID code set for these addresses.



**Figure 32.15 ID code storage addresses**

### Read Check Data

This command reads the check data that confirms that the write data, which was sent with the page program command, was successfully received.

- (1) Transfer the "FD16" command code with the 1st byte.
- (2) The check data (low) is received with the 2nd byte and the check data (high) with the 3rd.

To use this read check data command, first execute the command and then initialize the check data. Next, execute the page program command the required number of times. After that, when the read check command is executed again, the check data for all of the read data that was sent with the page program command during this time is read. The check data is the result of CRC operation of write data.

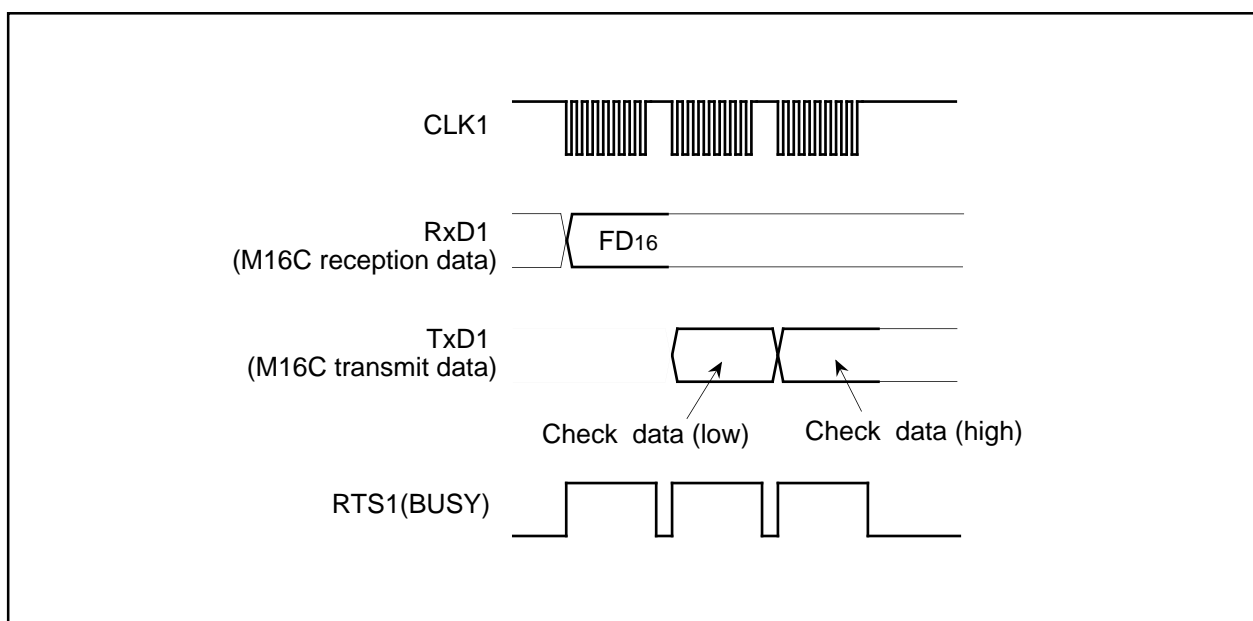


Figure 32.16 Timing for the read check data

### Data Protection (Block Lock)

Each of the blocks in Figure 32.17 have a nonvolatile lock bit that specifies protection (block lock) against erasing/writing. A block is locked (writing "0" for the lock bit) with the lock bit program command. Also, the lock bit of any block can be read with the read lock bit status command.

Block lock disable/enable is determined by the status of the lock bit itself and execution status of the lock bit disable and lock enable bit commands.

- (1) After the reset has been cancelled and the lock bit enable command executed, the specified block can be locked/unlocked using the lock bit (lock bit data). Blocks with a "0" lock bit data are locked and cannot be erased or written in. On the other hand, blocks with a "1" lock bit data are unlocked and can be erased or written in.
- (2) After the lock bit enable command has been executed, all blocks are unlocked regardless of lock bit data status and can be erased or written in. In this case, lock bit data that was "0" before the block was erased is set to "1" (unlocked) after erasing, therefore the block is actually unlocked with the lock bit.

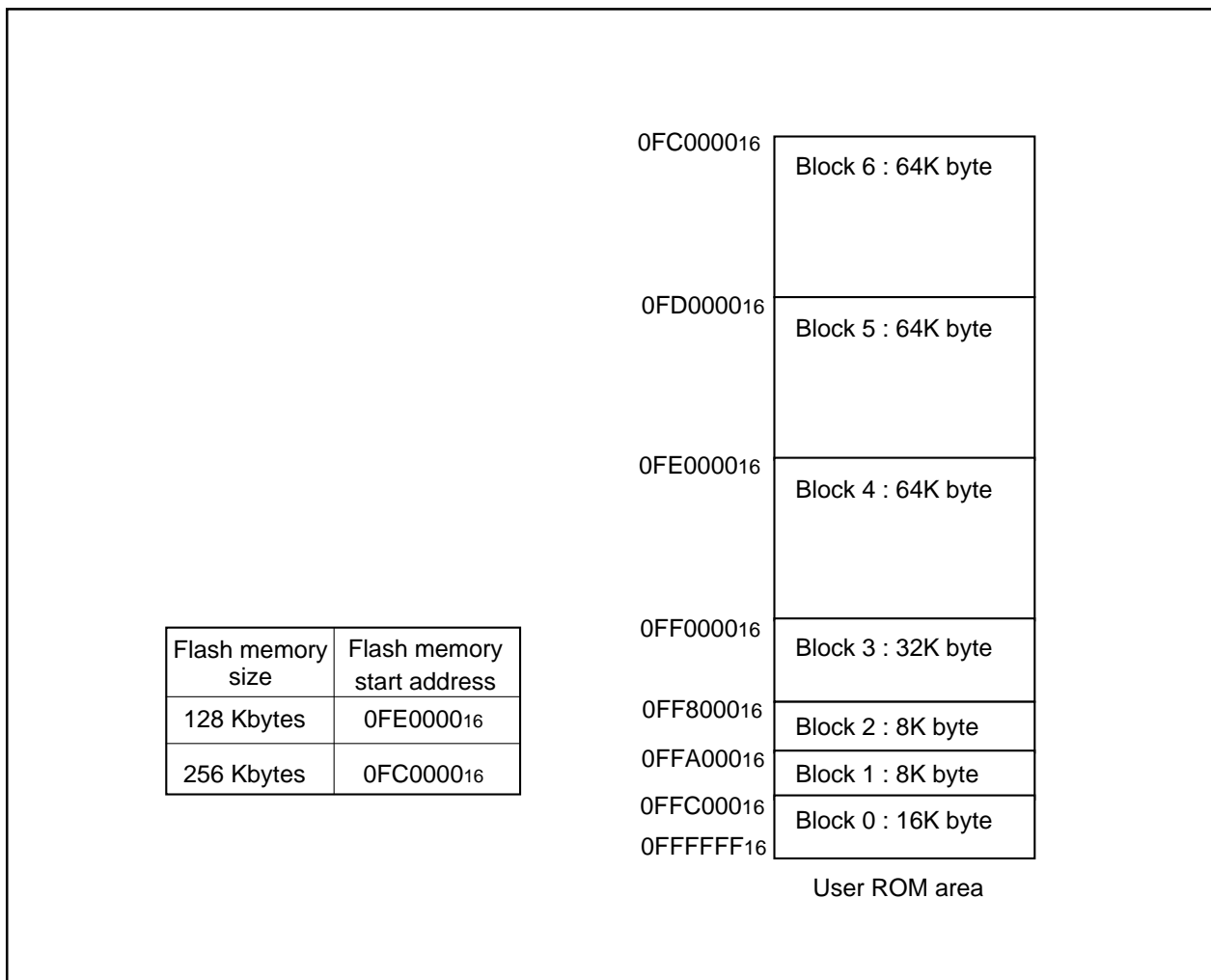


Figure 32.17 Blocks in the user area



## Status Register (SRD)

The status register indicates operating status of the flash memory and status such as whether an erase operation or a program ended successfully or in error. It can be read by writing the read status register command (70<sub>16</sub>). Also, the status register is cleared by writing the clear status register command (50<sub>16</sub>). Table 32.2 gives the definition of each status register bit. After clearing the reset, the status register outputs "80<sub>16</sub>".

**Table 32.2 Status register (SRD)**

SRD0 bits	Status name	Definition	
		"1"	"0"
SR7 (bit7)	Write state machine (WSM) status	Ready	Busy
SR6 (bit6)	Reserved	-	-
SR5 (bit5)	Erase status	Terminated in error	Terminated normally
SR4 (bit4)	Program status	Terminated in error	Terminated normally
SR3 (bit3)	Block status after program	Terminated in error	Terminated normally
SR2 (bit2)	Reserved	-	-
SR1 (bit1)	Reserved	-	-
SR0 (bit0)	Reserved	-	-

### Write State Machine (WSM) Status (SR7)

The write state machine (WSM) status indicates the operating status of the flash memory. When power is turned on, "1" (ready) is set for it. The bit is set to "0" (busy) during an auto write or auto erase operation, but it is set back to "1" when the operation ends.

### Erase Status (SR5)

The erase status reports the operating status of the auto erase operation. If an erase error occurs, it is set to "1". When the erase status is cleared, it is set to "0".

### Program Status (SR4)

The program status reports the operating status of the auto write operation. If a write error occurs, it is set to "1". When the program status is cleared, it is set to "0".

### Program Status After Program (SR3)

If excessive data is written (phenomenon whereby the memory cell becomes depressed which results in data not being read correctly), "1" is set for the program status after-program at the end of the page write operation. In other words, when writing ends successfully, "80<sub>16</sub>" is output; when writing fails, "90<sub>16</sub>" is output; and when excessive data is written, "88<sub>16</sub>" is output.

If "1" is written for any of the SR5, SR4 or SR3 bits, the page program, block erase, erase all unlocked blocks and lock bit program commands are not accepted. Before executing these commands, execute the clear status register command (50<sub>16</sub>) and clear the status register.

### Status Register 1 (SRD1)

Status register 1 indicates the status of serial communications, results from ID checks and results from check sum comparisons. It can be read after the SRD by writing the read status register command (7016). Also, status register 1 is cleared by writing the clear status register command (5016).

Table 31.3 gives the definition of each status register 1 bit. "0016" is output when power is turned ON and the flag status is maintained even after the reset.

**Table 32.3 Status register 1 (SRD1)**

SRD1 bits	Status name	Definition	
		"1"	"0"
SR15 (bit7)	Boot update completed bit	Update completed	Not update
SR14 (bit6)	Reserved	-	-
SR13 (bit5)	Reserved	-	-
SR12 (bit4)	Checksum match bit	Match	Mismatch
SR11 (bit3) SR10 (bit2)	ID check completed bits	00 01 10 11	Not verified Verification mismatch Reserved Verified
SR9 (bit1)	Data receive time out	Time out	Normal operation
SR8 (bit0)	Reserved	-	-

#### Boot Update Completed Bit (SR15)

This flag indicates whether the control program was downloaded to the RAM or not, using the download function.

#### Check Sum Consistency Bit (SR12)

This flag indicates whether the check sum matches or not when a program, is downloaded for execution using the download function.

#### ID Check Completed Bits (SR11 and SR10)

These flags indicate the result of ID checks. Some commands cannot be accepted without an ID check.

#### Data Reception Time Out (SR9)

This flag indicates when a time out error is generated during data reception. If this flag is attached during data reception, the received data is discarded and the microcomputer returns to the command wait state.

### Full Status Check

Results from executed erase and program operations can be known by running a full status check. Figure 32.18 shows a flowchart of the full status check and explains how to remedy errors which occur.

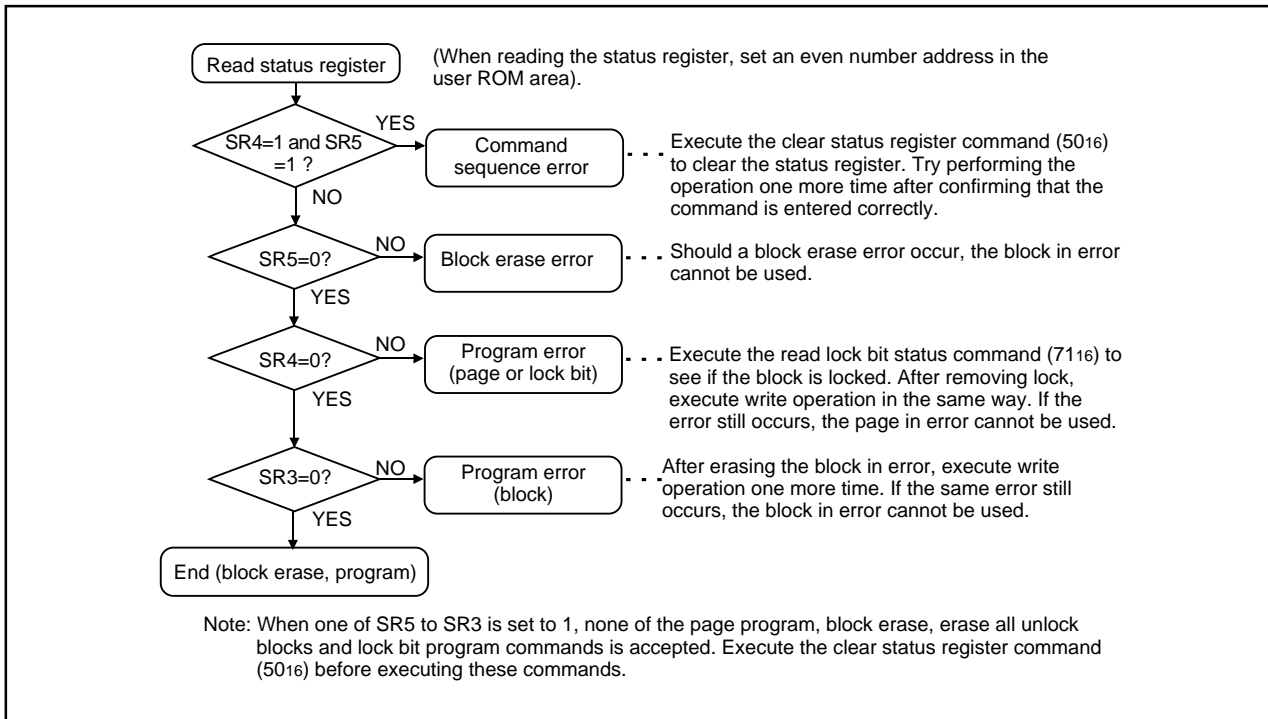


Figure 32.18 Full status check flowchart and remedial procedure for errors

### Example Circuit Application for The Standard Serial I/O Mode 1

The below figure shows a circuit application for the standard serial I/O mode 1. Control pins will vary according to peripheral unit (programmer), therefore see the peripheral unit (programmer) manual for more information.

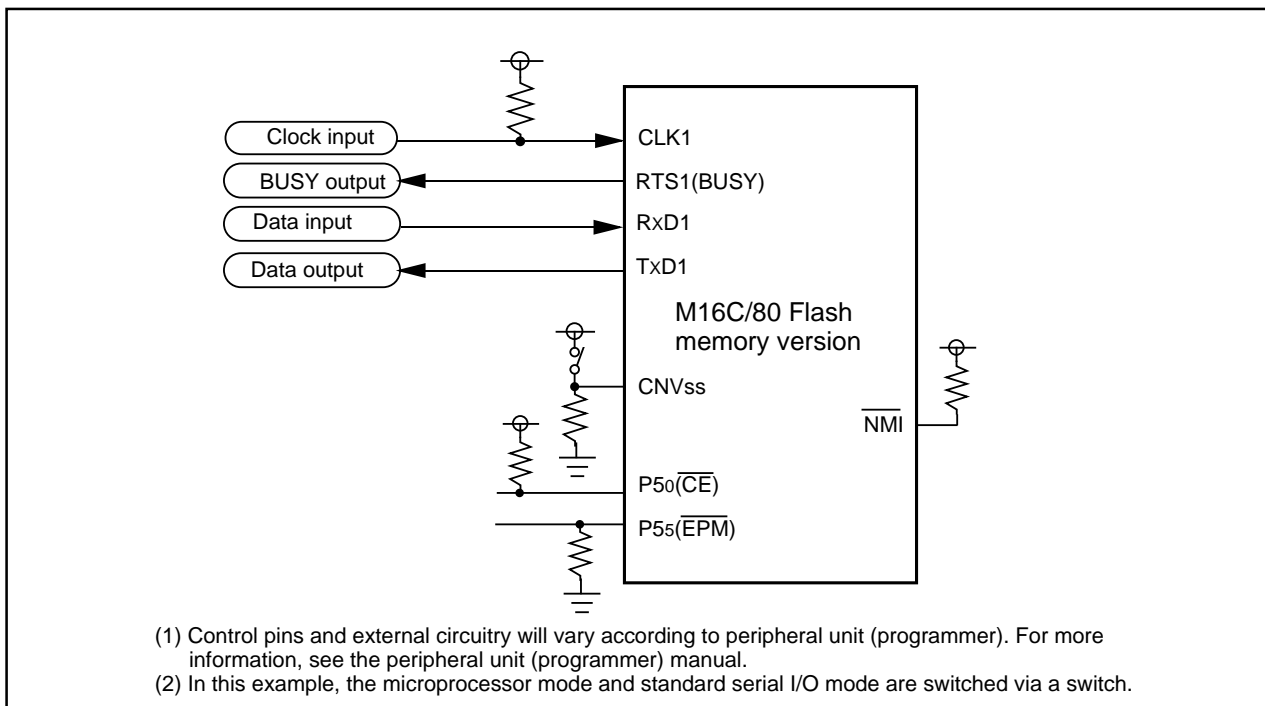


Figure 32.19 Example circuit application for the standard serial I/O mode 1

### 32.2 Overview of standard serial I/O mode 2 (clock asynchronized)

In standard serial I/O mode 2, software commands, addresses and data are input and output between the MCU and peripheral units (serial programmer, etc.) using 2-wire clock-asynchronized serial I/O (UART1). Standard serial I/O mode 2 is engaged by releasing the reset with the P65 (CLK1) pin "L" level.

The TxD1 pin is for CMOS output. Data transfer is in 8-bit units with LSB first, 1 stop bit and parity OFF.

After the reset is released, connections can be established at 9,600 bps when initial communications (Figure 32.20) are made with a peripheral unit. However, this requires a main clock with a minimum 2 MHz input oscillation frequency. Baud rate can also be changed from 9,600 bps to 19,200, 38,400, 57,600 or 115,200 bps by executing software commands. However, communication errors may occur because of the oscillation frequency of the main clock. If errors occur, change the main clock's oscillation frequency and the baud rate.

After executing commands from a peripheral unit that requires time to erase and write data, as with erase and program commands, allow a sufficient time interval or execute the read status command and check how processing ended, before executing the next command.

Data and status registers in memory can be read after transmitting software commands. Status, such as the operating state of the flash memory or whether a program or erase operation ended successfully or not, can be checked by reading the status register. Here following are explained initial communications with peripheral units, how frequency is identified and software commands.

#### Initial communications with peripheral units

After the reset is released, the bit rate generator is adjusted to 9,600 bps to match the oscillation frequency of the main clock, by sending the code as prescribed by the protocol for initial communications with peripheral units (Figure 32.20).

- (1) Transmit "0016" from a peripheral unit 16 times. (The MCU with internal flash memory sets the bit rate generator so that "0016" can be successfully received.)
- (2) The MCU with internal flash memory outputs the "B016" check code and initial communications end successfully \*1. Initial communications must be transmitted at a speed of 9,600 bps and a transfer interval of a minimum 15 ms. Also, the baud rate at the end of initial communications is 9,600 bps.

\*1. If the peripheral unit cannot receive "B016" successfully, change the oscillation frequency of the main clock.

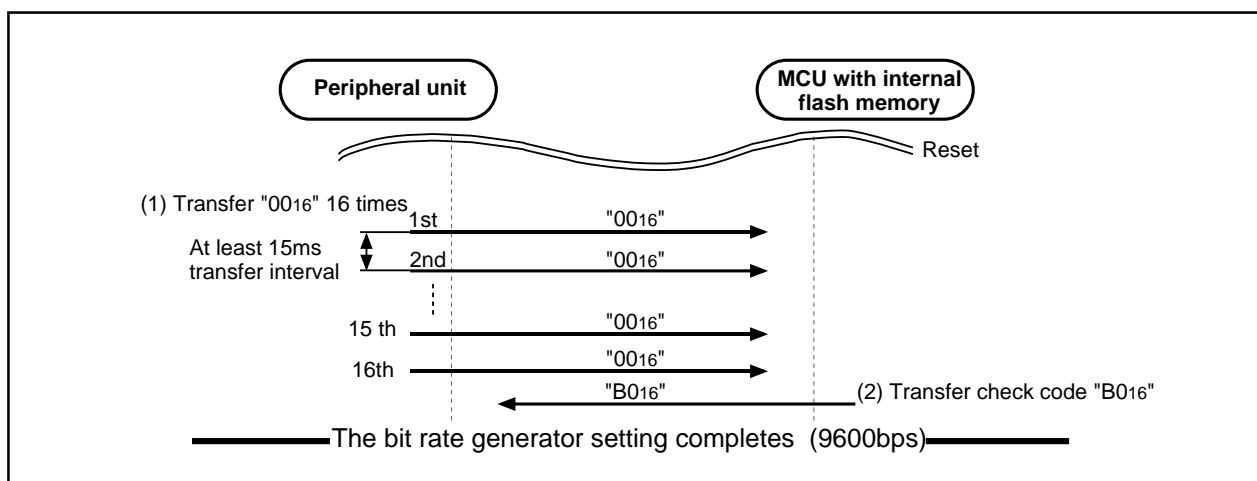


Figure 32.20 Peripheral unit and initial communication

### How frequency is identified

When "0016" data is received 16 times from a peripheral unit at a baud rate of 9,600 bps, the value of the bit rate generator is set to match the operating frequency (2 - 20 MHz). The highest speed is taken from the first 8 transmissions and the lowest from the last 8. These values are then used to calculate the bit rate generator value for a baud rate of 9,600 bps.

Baud rate cannot be attained with some operating frequencies. Table 32.4 gives the operation frequency and the baud rate that can be attained for.

**Table 32.4 Operation frequency and the baud rate**

Operation frequency (MHz)	Baud rate 9,600bps	Baud rate 19,200bps	Baud rate 38,400bps	Baud rate 57,600bps	Baud rate 115,200bps
20MHz	√	√	√	√	√
16MHz	√	√	√	√	—
12MHz	√	√	√	√	—
11MHz	√	√	√	√	—
10MHz	√	√	√	√	—
8MHz	√	√	√	√	—
7.3728MHz	√	√	√	√	—
6MHz	√	√	√	—	—
5MHz	√	√	√	—	—
4.5MHz	√	√	√	√	—
4.194304MHz	√	√	√	—	—
4MHz	√	√	—	—	—
3.58MHz	√	√	√	√	—
3MHz	√	√	√	—	—
2MHz	√	—	—	—	—

√ : Communications possible

— : Communications not possible

## Software Commands

Table 32.5 lists software commands. In the standard serial I/O mode 2, erase operations, programs and reading are controlled by transferring software commands via the RxD1 pin. Standard serial I/O mode 2 adds five transmission speed commands - 9,600, 19,200, 38,400, 57,600 and 115,200 bps - to the software commands of standard serial I/O mode 1. Software commands are explained here below.

**Table 32.5 Software commands (Standard serial I/O mode 2)**

	Control command	1st byte transfer	2nd byte	3rd byte	4th byte	5th byte	6th byte		When ID is not verified
1	Page read	FF <sub>16</sub>	Address (middle)	Address (high)	Data output	Data output	Data output	Data output to 259th byte	Not acceptable
2	Page program	41 <sub>16</sub>	Address (middle)	Address (high)	Data input	Data input	Data input	Data input to 259th byte	Not acceptable
3	Block erase	20 <sub>16</sub>	Address (middle)	Address (high)	D0 <sub>16</sub>				Not acceptable
4	Erase all unlocked blocks	A7 <sub>16</sub>	D0 <sub>16</sub>						Not acceptable
5	Read status register	70 <sub>16</sub>	SRD output	SRD1 output					Acceptable
6	Clear status register	50 <sub>16</sub>							Not acceptable
7	Read lock bit status	71 <sub>16</sub>	Address (middle)	Address (high)	Lock bit data output				Not acceptable
8	Lock bit program	77 <sub>16</sub>	Address (middle)	Address (high)	D0 <sub>16</sub>				Not acceptable
9	Lock bit enable	7A <sub>16</sub>							Not acceptable
10	Lock bit disable	75 <sub>16</sub>							Not acceptable
11	Code processing function	F5 <sub>16</sub>	Address (low)	Address (middle)	Address (high)	ID size	ID1	To ID7	Acceptable
12	Download function	FA <sub>16</sub>	Size (low)	Size (high)	Check-sum	Data input	To required number of times		Not acceptable
13	Version data output function	FB <sub>16</sub>	Version data output	Version data output	Version data output	Version data output	Version data output	Version data output to 9th byte	Acceptable
14	Boot ROM area output function	FC <sub>16</sub>	Address (middle)	Address (high)	Data output	Data output	Data output	Data output to 259th byte	Not acceptable
15	Read check data	FD <sub>16</sub>	Check data (low)	Check data (high)					Not acceptable
16	Baud rate 9600	B0 <sub>16</sub>	B0 <sub>16</sub>						Acceptable
17	Baud rate 19200	B1 <sub>16</sub>	B1 <sub>16</sub>						Acceptable
18	Baud rate 38400	B2 <sub>16</sub>	B2 <sub>16</sub>						Acceptable
19	Baud rate 57600	B3 <sub>16</sub>	B3 <sub>16</sub>						Acceptable
20	Baud rate 115200	B4 <sub>16</sub>	B4 <sub>16</sub>						Acceptable

Note 1: Shading indicates transfer from flash memory microcomputer to peripheral unit. All other data is transferred from the peripheral unit to the flash memory microcomputer.

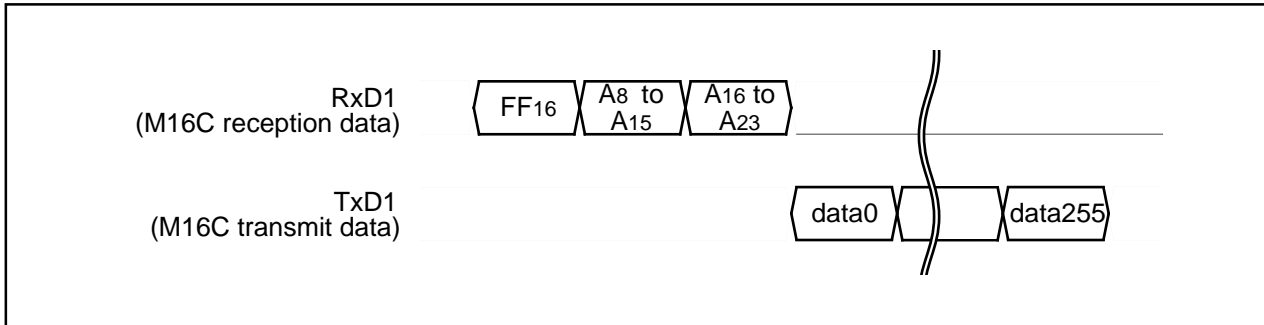
Note 2: SRD refers to status register data. SRD1 refers to status register data 1.

Note 3: All commands can be accepted when the flash memory is totally blank.

**Page Read Command**

This command reads the specified page (256 bytes) in the flash memory sequentially one byte at a time. Execute the page read command as explained here following.

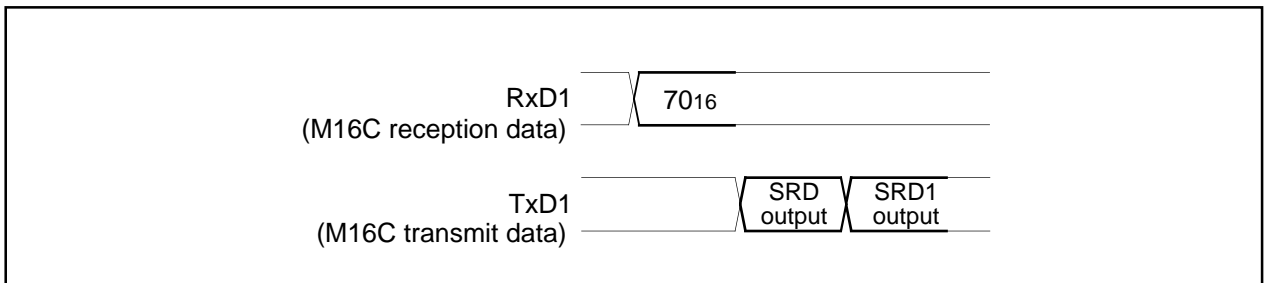
- (1) Transfer the “FF16” command code with the 1st byte.
- (2) Transfer addresses A8 to A15 and A16 to A23 with the 2nd and 3rd bytes respectively.
- (3) From the 4th byte onward, data (D0–D7) for the page (256 bytes) specified with addresses A8 to A23 will be output sequentially from the smallest address first in sync with the rise of the clock.



**Figure 32.21 Timing for page read**

**Read Status Register Command**

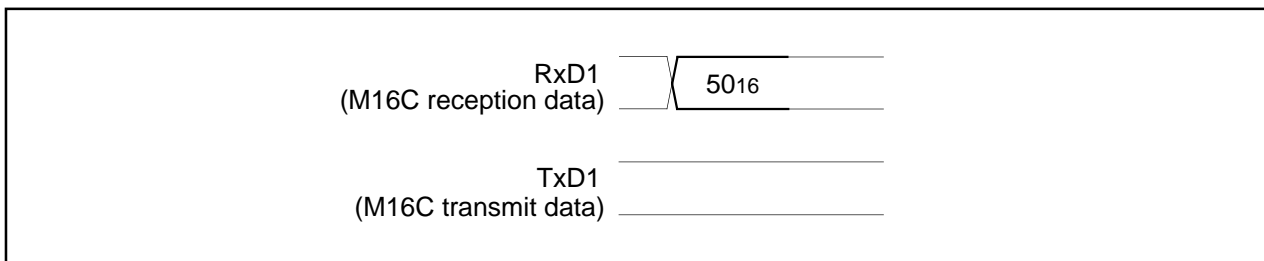
This command reads status information. When the “7016” command code is sent with the 1st byte, the contents of the status register (SRD) specified with the 2nd byte and the contents of status register 1 (SRD1) specified with the 3rd byte are read.



**Figure 32.22 Timing for reading the status register**

**Clear Status Register Command**

This command clears the bits (SR3–SR5) which are set when the status register operation ends in error. When the “5016” command code is sent with the 1st byte, the aforementioned bits are cleared.



**Figure 32.23 Timing for clearing the status register**

### Page Program Command

This command writes the specified page (256 bytes) in the flash memory sequentially one byte at a time. Execute the page program command as explained here following.

- (1) Transfer the “4116” command code with the 1st byte.
- (2) Transfer addresses A8 to A15 and A16 to A23 with the 2nd and 3rd bytes respectively.
- (3) From the 4th byte onward, as write data (D0–D7) for the page (256 bytes) specified with addresses A8 to A23 is input sequentially from the smallest address first, that page is automatically written.

The result of the page program can be known by reading the status register. For more information, see the section on the status register.

Each block can be write-protected with the lock bit. For more information, see the section on the data protection function. Additional writing is not allowed with already programmed pages.

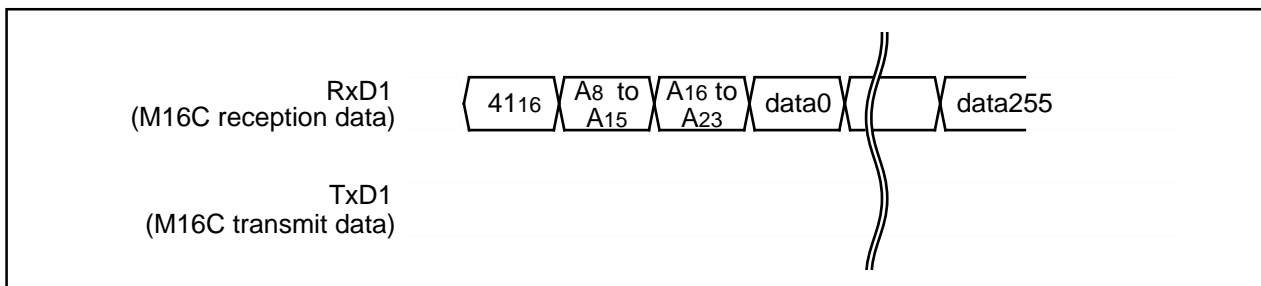


Figure 32.24 Timing for the page program

### Block Erase Command

This command erases the data in the specified block. Execute the block erase command as explained here following.

- (1) Transfer the “2016” command code with the 1st byte.
- (2) Transfer addresses A8 to A15 and A16 to A23 with the 2nd and 3rd bytes respectively.
- (3) Transfer the verify command code “D016” with the 4th byte. With the verify command code, the erase operation will start for the specified block in the flash memory. Write the highest address of the specified block for addresses A16 to A23.

After block erase ends, the result of the block erase operation can be known by reading the status register. For more information, see the section on the status register.

Each block can be erase-protected with the lock bit. For more information, see the section on the data protection function.

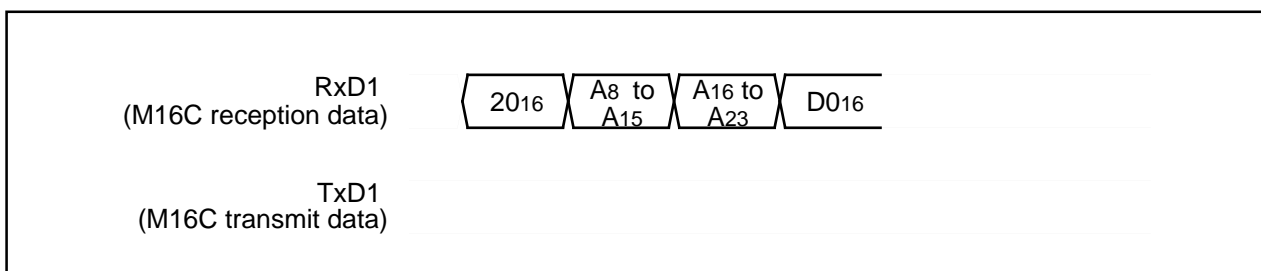


Figure 32.25 Timing for block erasing

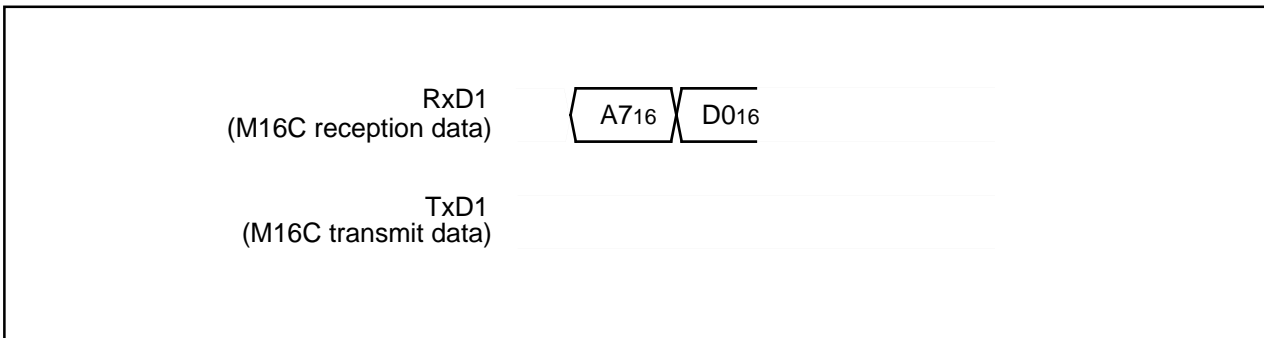


**Erase All Unlocked Blocks Command**

This command erases the content of all blocks. Execute the erase all unlocked blocks command as explained here following.

- (1) Transfer the “A716” command code with the 1st byte.
- (2) Transfer the verify command code “D016” with the 2nd byte. With the verify command code, the erase operation will start and continue for all blocks in the flash memory.

The result of the erase operation can be known by reading the status register. Each block can be erase-protected with the lock bit. For more information, see the section on the data protection function.



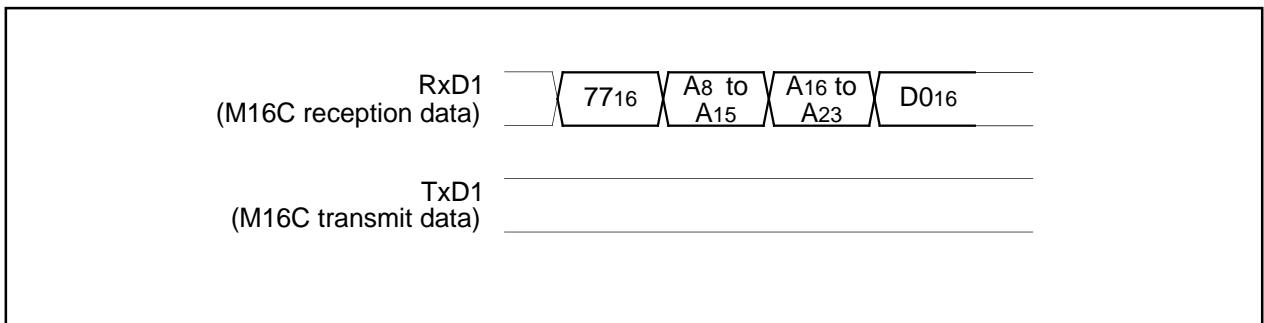
**Figure 32.26 Timing for erasing all unlocked blocks**

**Lock Bit Program Command**

This command writes “0” (lock) for the lock bit of the specified block. Execute the lock bit program command as explained here following.

- (1) Transfer the “7716” command code with the 1st byte.
- (2) Transfer addresses A8 to A15 and A16 to A23 with the 2nd and 3rd bytes respectively.
- (3) Transfer the verify command code “D016” with the 4th byte. With the verify command code, “0” is written for the lock bit of the specified block. Write the highest address of the specified block for addresses A8 to A23.

Lock bit status can be read with the read lock bit status command. For information on the lock bit function, reset procedure and so on, see the section on the data protection function.

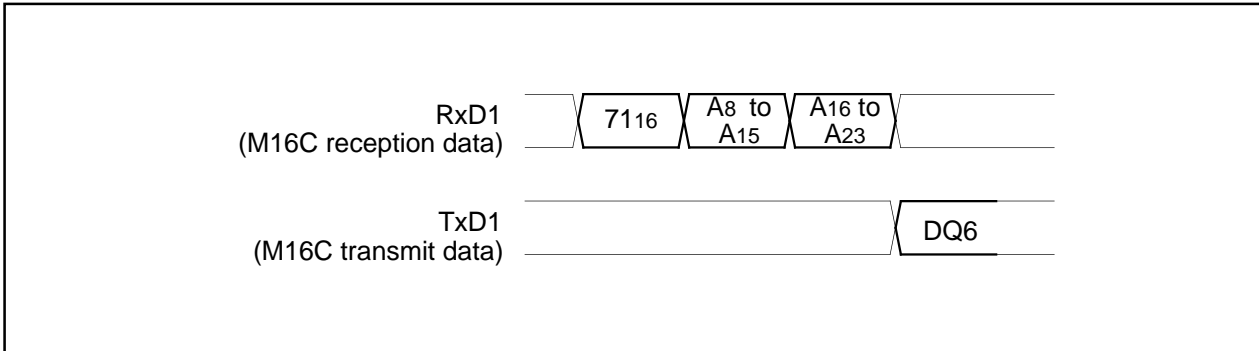


**Figure 32.27 Timing for the lock bit program**

**Read Lock Bit Status Command**

This command reads the lock bit status of the specified block. Execute the read lock bit status command as explained here following.

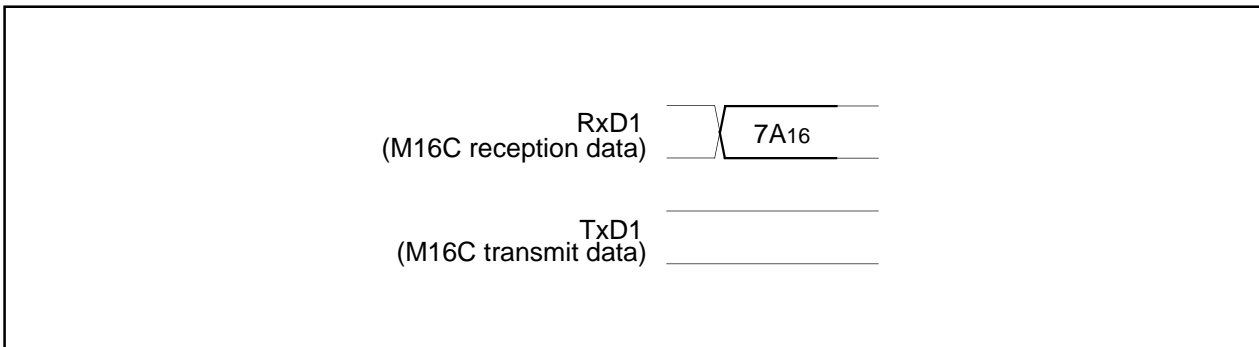
- (1) Transfer the “71<sub>16</sub>” command code with the 1st byte.
- (2) Transfer addresses A<sub>8</sub> to A<sub>15</sub> and A<sub>16</sub> to A<sub>23</sub> with the 2nd and 3rd bytes respectively.
- (3) The lock bit data of the specified block is output with the 4th byte. Write the highest address of the specified block for addresses A<sub>8</sub> to A<sub>23</sub>.



**Figure 32.28 Timing for reading lock bit status**

**Lock Bit Enable Command**

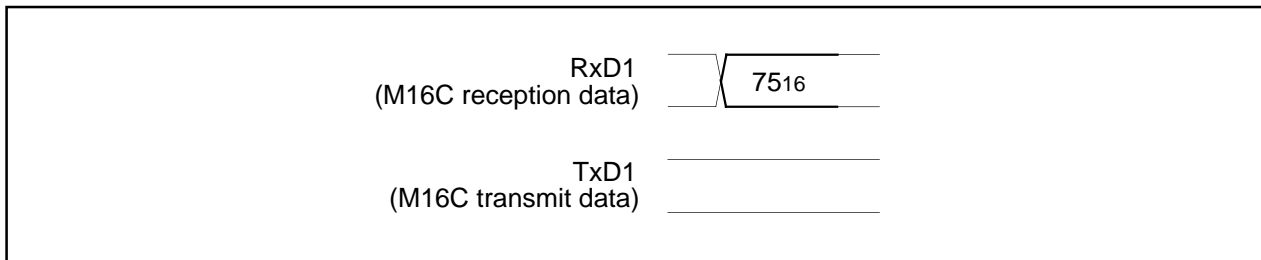
This command enables the lock bit in blocks whose bit was disabled with the lock bit disable command. The command code “7A<sub>16</sub>” is sent with the 1st byte of the serial transmission. This command only enables the lock bit function; it does not set the lock bit itself.



**Figure 32.29 Timing for enabling the lock bit**

**Lock Bit Disable Command**

This command disables the lock bit. The command code “7516” is sent with the 1st byte of the serial transmission. This command only disables the lock bit function; it does not set the lock bit itself. However, if an erase command is executed after executing the lock bit disable command, “0” (locked) lock bit data is set to “1” (unlocked) after the erase operation ends. In any case, after the reset is cancelled, the lock bit is enabled.



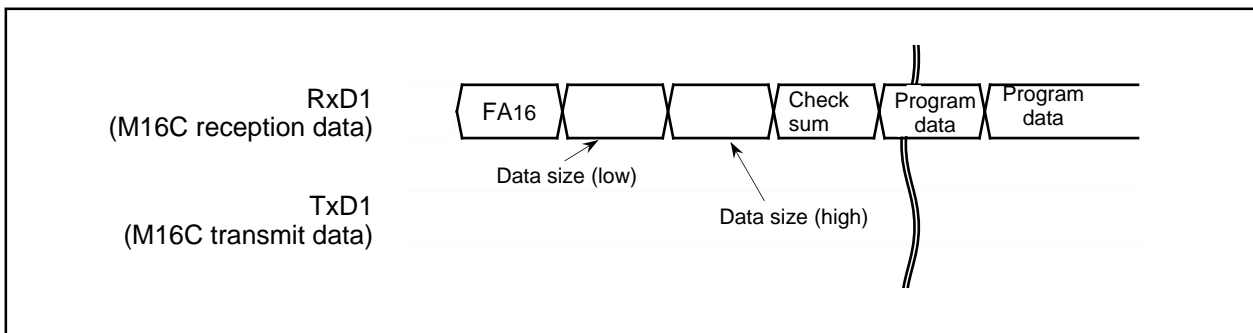
**Figure 32.30 Timing for disabling the lock bit**

**Download Command**

This command downloads a program to the RAM for execution. Execute the download command as explained here following.

- (1) Transfer the “FA16” command code with the 1st byte.
- (2) Transfer the program size with the 2nd and 3rd bytes.
- (3) Transfer the check sum with the 4th byte. The check sum is added to all data sent with the 5th byte onward.
- (4) The program to execute is sent with the 5th byte onward.

When all data has been transmitted, if the check sum matches, the downloaded program is executed. The size of the program will vary according to the internal RAM.

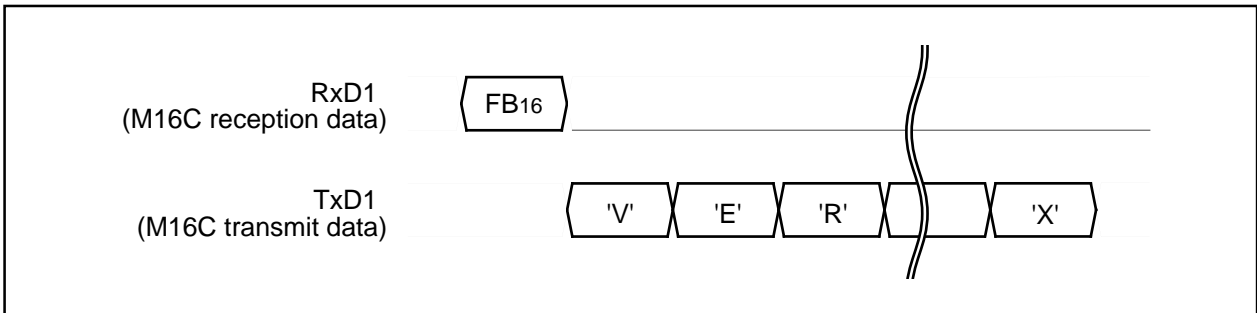


**Figure 32.31 Timing for download**

**Version Information Output Command**

This command outputs the version information of the control program stored in the boot area. Execute the version information output command as explained here following.

- (1) Transfer the “FB16” command code with the 1st byte.
- (2) The version information will be output from the 2nd byte onward. This data is composed of 8 ASCII code characters.

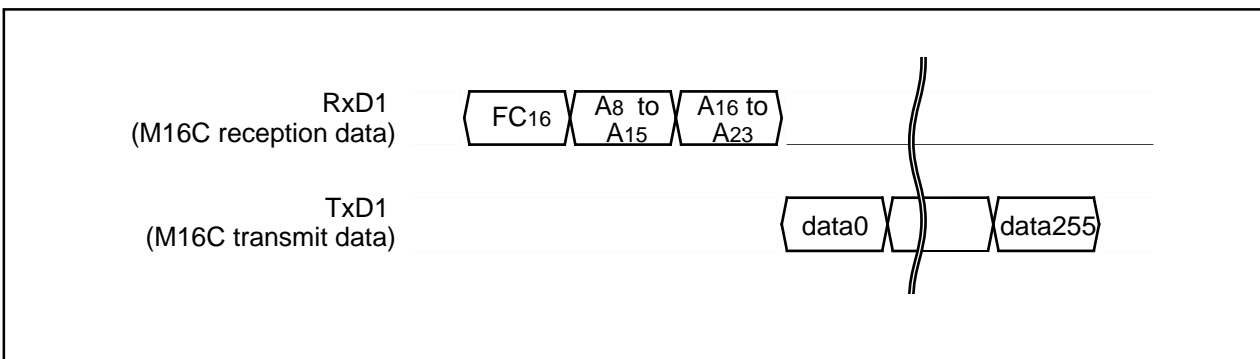


**Figure 32.34 Timing for version information output**

**Boot ROM Area Output Command**

This command outputs the control program stored in the boot ROM area in one page blocks (256 bytes). Execute the boot ROM area output command as explained here following.

- (1) Transfer the “FC16” command code with the 1st byte.
- (2) Transfer addresses A8 to A15 and A16 to A23 with the 2nd and 3rd bytes respectively.
- (3) From the 4th byte onward, data (D0–D7) for the page (256 bytes) specified with addresses A8 to A23 will be output sequentially from the smallest address first, in sync with the rise of the clock.

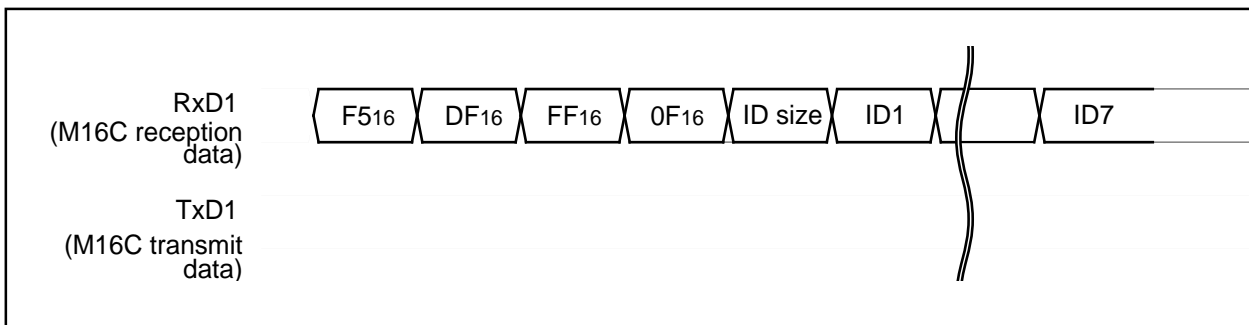


**Figure 32.33 Timing for boot ROM area output**

**ID Check**

This command checks the ID code. Execute the boot ID check command as explained here following.

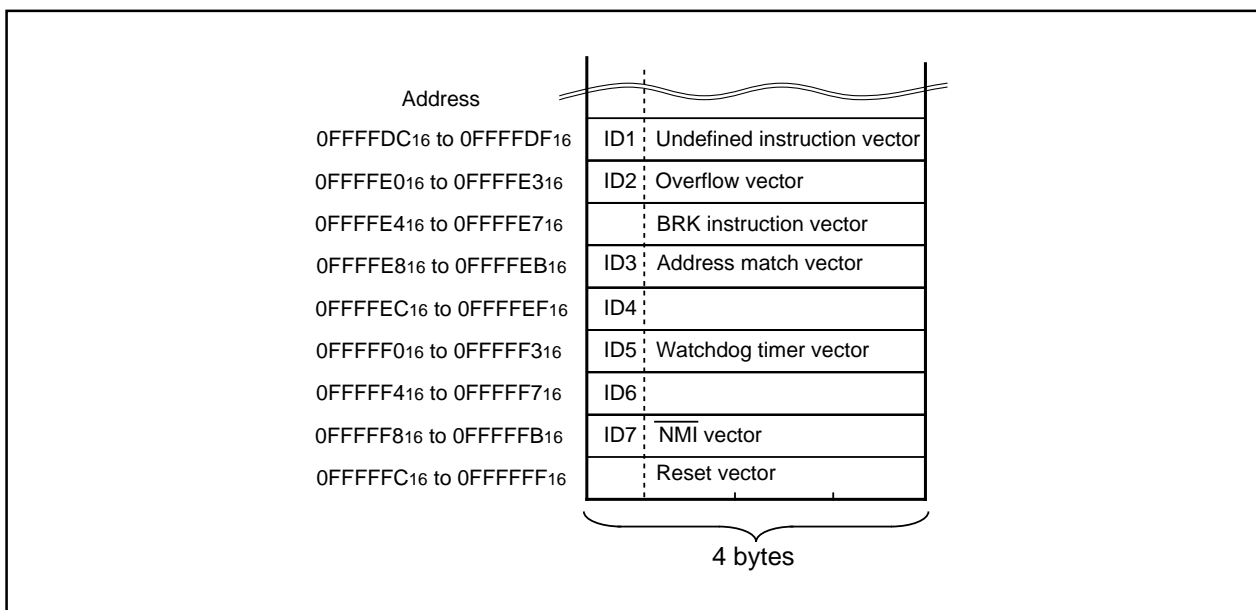
- (1) Transfer the "F5<sub>16</sub>" command code with the 1st byte.
- (2) Transfer addresses A0 to A7, A8 to A15 and A16 to A23 of the 1st byte of the ID code with the 2nd, 3rd and 4th bytes respectively.
- (3) Transfer the number of data sets of the ID code with the 5th byte.
- (4) The ID code is sent with the 6th byte onward, starting with the 1st byte of the code.



**Figure 32.34 Timing for the ID check**

**ID Code**

When the flash memory is not blank, the ID code sent from the peripheral units and the ID code written in the flash memory are compared to see if they match. If the codes do not match, the command sent from the peripheral units is not accepted. An ID code contains 8 bits of data. Area is, from the 1st byte, addresses 0FFFFDF<sub>16</sub>, 0FFFFE3<sub>16</sub>, 0FFFFEB<sub>16</sub>, 0FFFFEF<sub>16</sub>, 0FFFFF3<sub>16</sub>, 0FFFFF7<sub>16</sub> and 0FFFFFB<sub>16</sub>. Write a program into the flash memory, which already has the ID code set for these addresses.



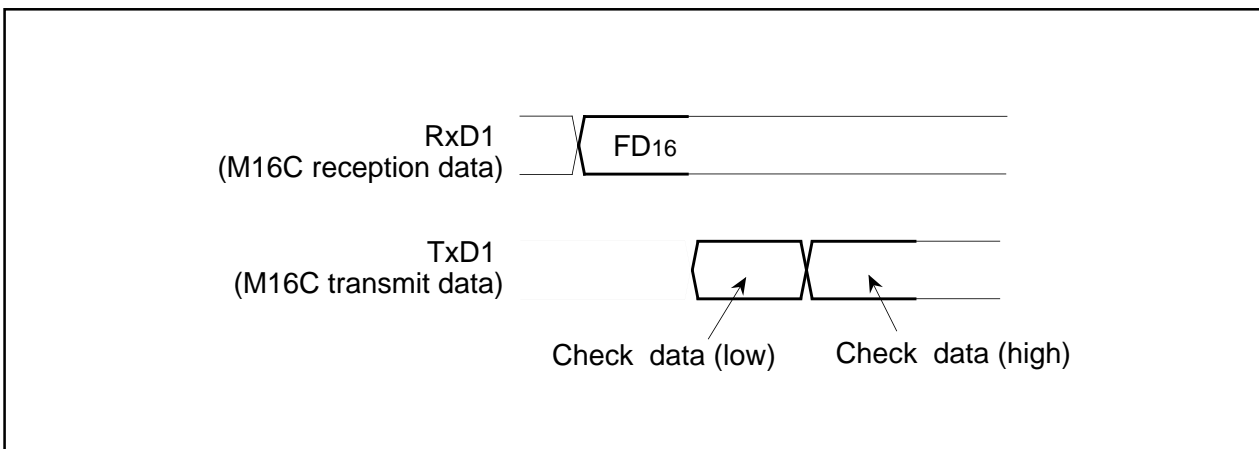
**Figure 32.35 ID code storage addresses**

**Read Check Data**

This command reads the check data that confirms that the write data, which was sent with the page program command, was successfully received.

- (1) Transfer the "FD16" command code with the 1st byte.
- (2) The check data (low) is received with the 2nd byte and the check data (high) with the 3rd.

To use this read check data command, first execute the command and then initialize the check data. Next, execute the page program command the required number of times. After that, when the read check command is executed again, the check data for all of the read data that was sent with the page program command during this time is read. The check data is the result of CRC operation of write data.

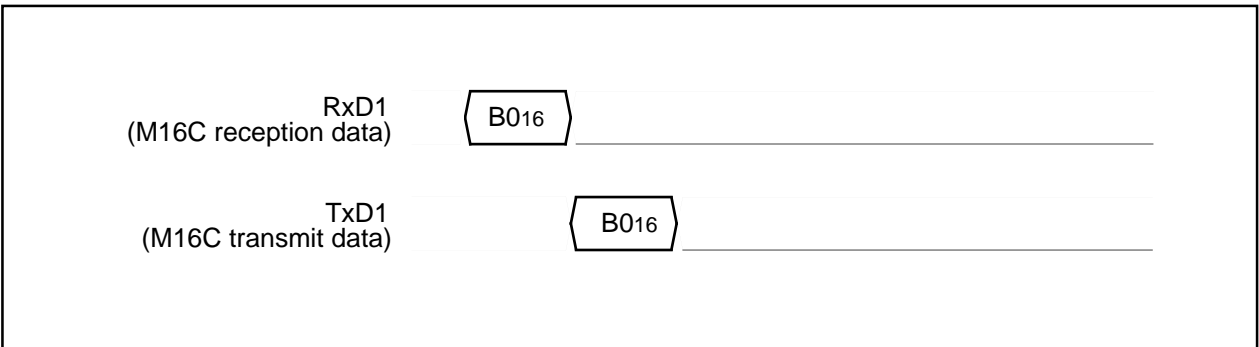


**Figure 32.36 Timing for the read check data**

**Baud Rate 9600**

This command changes baud rate to 9,600 bps. Execute it as follows.

- (1) Transfer the "B016" command code with the 1st byte.
- (2) After the "B016" check code is output with the 2nd byte, change the baud rate to 9,600 bps.



**Figure 32.37 Timing of baud rate 9600**

**Baud Rate 19200**

This command changes baud rate to 19,200 bps. Execute it as follows.

- (1) Transfer the "B116" command code with the 1st byte.
- (2) After the "B116" check code is output with the 2nd byte, change the baud rate to 19,200 bps.

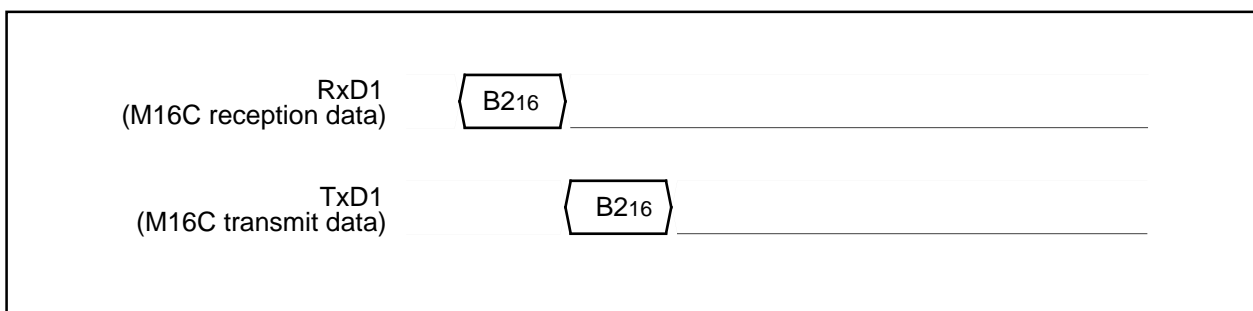


**Figure 32.38 Timing of baud rate 19200**

**Baud Rate 38400**

This command changes baud rate to 38,400 bps. Execute it as follows.

- (1) Transfer the "B216" command code with the 1st byte.
- (2) After the "B216" check code is output with the 2nd byte, change the baud rate to 38,400 bps.

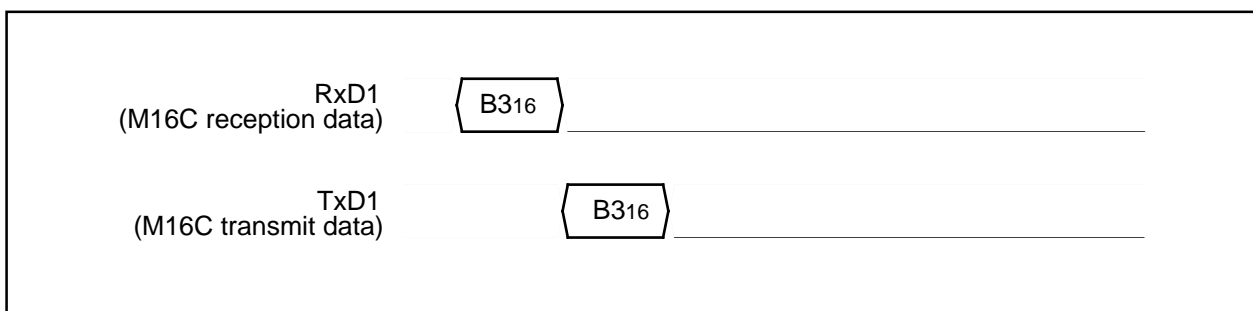


**Figure 32.39 Timing of baud rate 38400**

**Baud Rate 57600**

This command changes baud rate to 57,600 bps. Execute it as follows.

- (1) Transfer the "B316" command code with the 1st byte.
- (2) After the "B316" check code is output with the 2nd byte, change the baud rate to 57,600 bps.

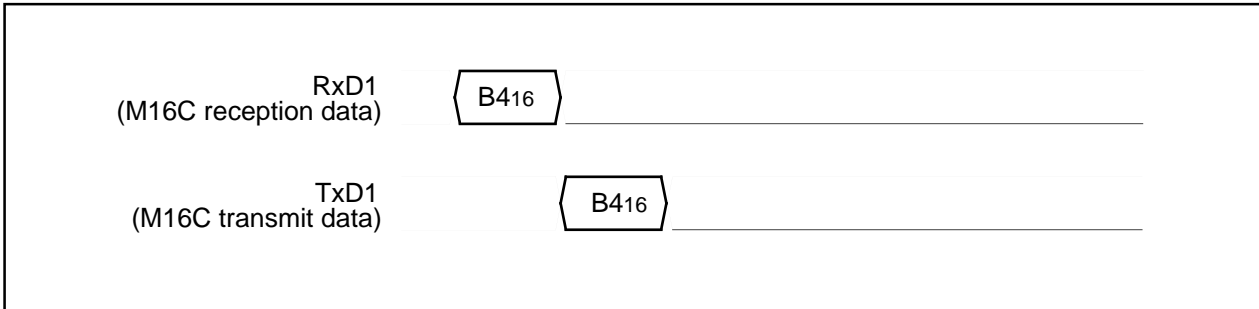


**Figure 32.40 Timing of baud rate 57600**

**Baud Rate 115200**

This command changes baud rate to 115,200 bps. Execute it as follows.

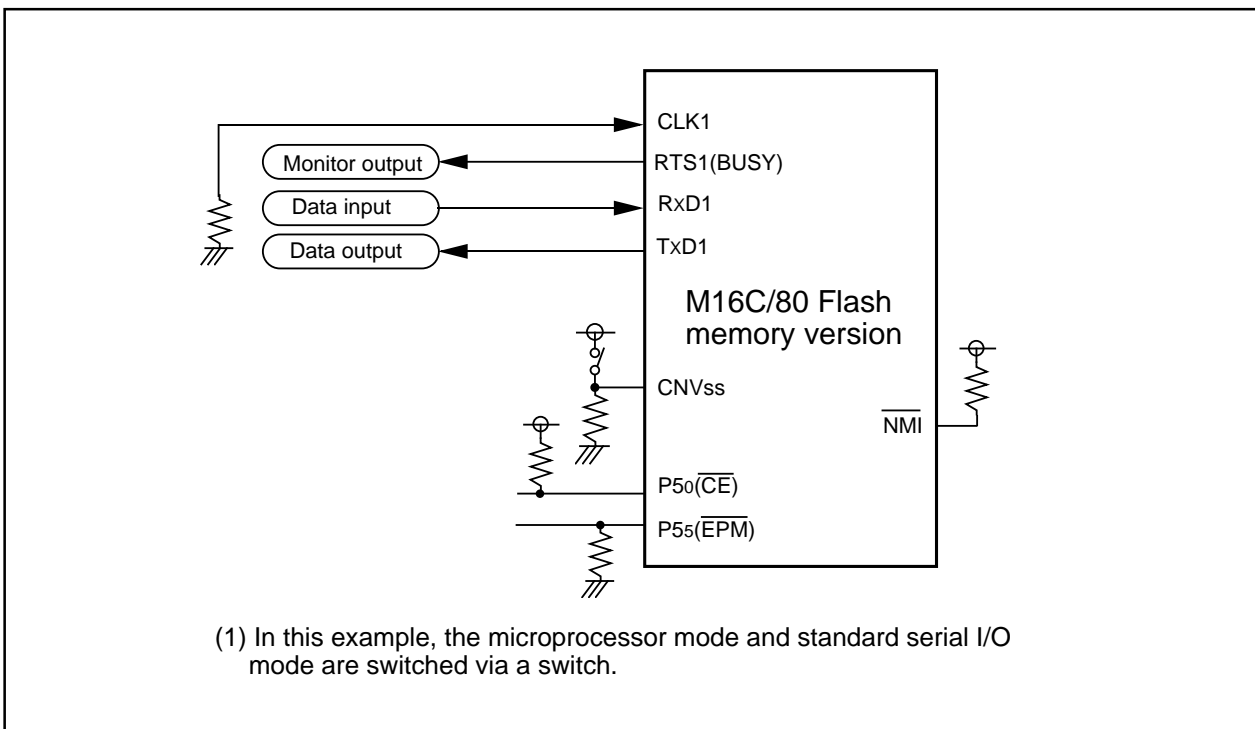
- (1) Transfer the "B416" command code with the 1st byte.
- (2) After the "B416" check code is output with the 2nd byte, change the baud rate to 19,200 bps.



**Figure 32.41 Timing of baud rate 115200**

**Example Circuit Application for The Standard Serial I/O Mode 2**

The below figure shows a circuit application for the standard serial I/O mode 2.



**Figure 32.42 Example circuit application for the standard serial I/O mode 2**



### 33. Appendix External ROM version with built-in boot loader

External ROM version of M16C/80 is available with built-in boot loader (firmware). By using the boot loader, users can download their rewrite program of Flash memory to the internal RAM. When using the following Flash memory\*, reprogramming of the external Flash memory can be done without downloading the rewrite program.

For more detail, please refer to the "Volume Boot Loader" in the application note of M16C/80 external ROM version.

\*: M5M29GB/T160BVP, M5M29GB/T320BVP and the equivalent of these.

The following shows Renesas plans to develop a line of M16C/80 products with built-in boot loader.

- (1) ROM capacity
- (2) Package    100P6S-A ... Plastic molded QFP  
                   100P6Q-A ... Plastic molded QFP  
                   144P6Q-A ... Plastic molded QFP

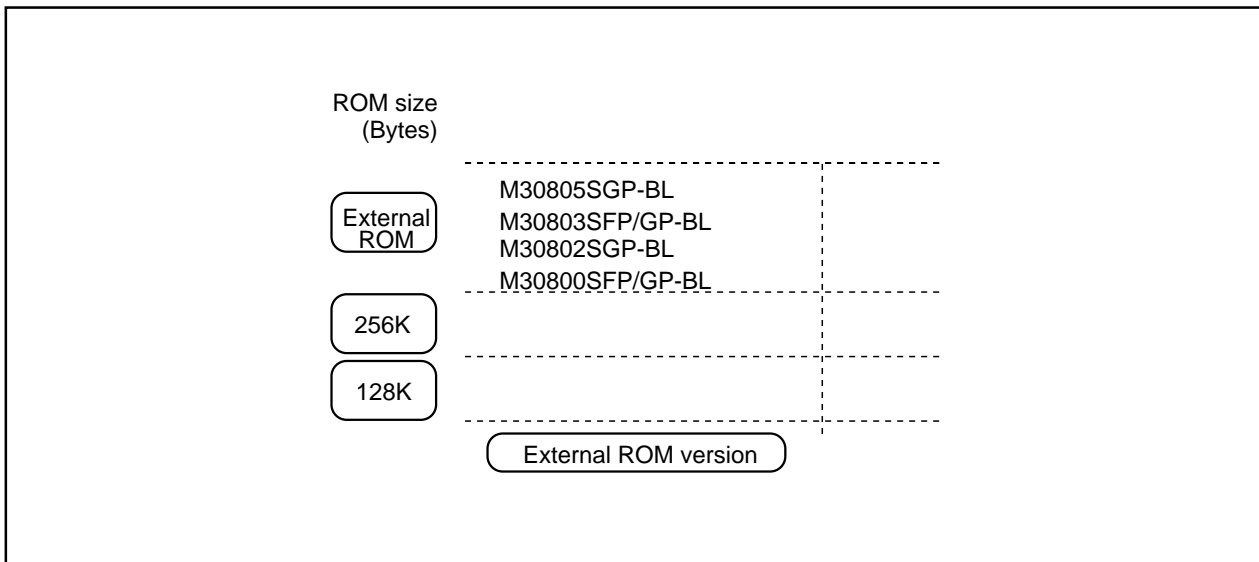


Figure 33.1 ROM Expansion

The following lists the M16C/80 products to be supported in the future.

Table 33.1 Product List

Type No	ROM capacity	RAM capacity	Package type	Remarks
M30800SFP-BL	—	10 Kbytes	100P6S-A	External ROM version with built-in boot loader
M30800SGP-BL			100P6Q-A	
M30802SGP-BL			144P6Q-A	
M30803SFP-BL	—	24 Kbytes	100P6S-A	
M30803SGP-BL			100P6Q-A	
M30805SGP-BL			144P6Q-A	

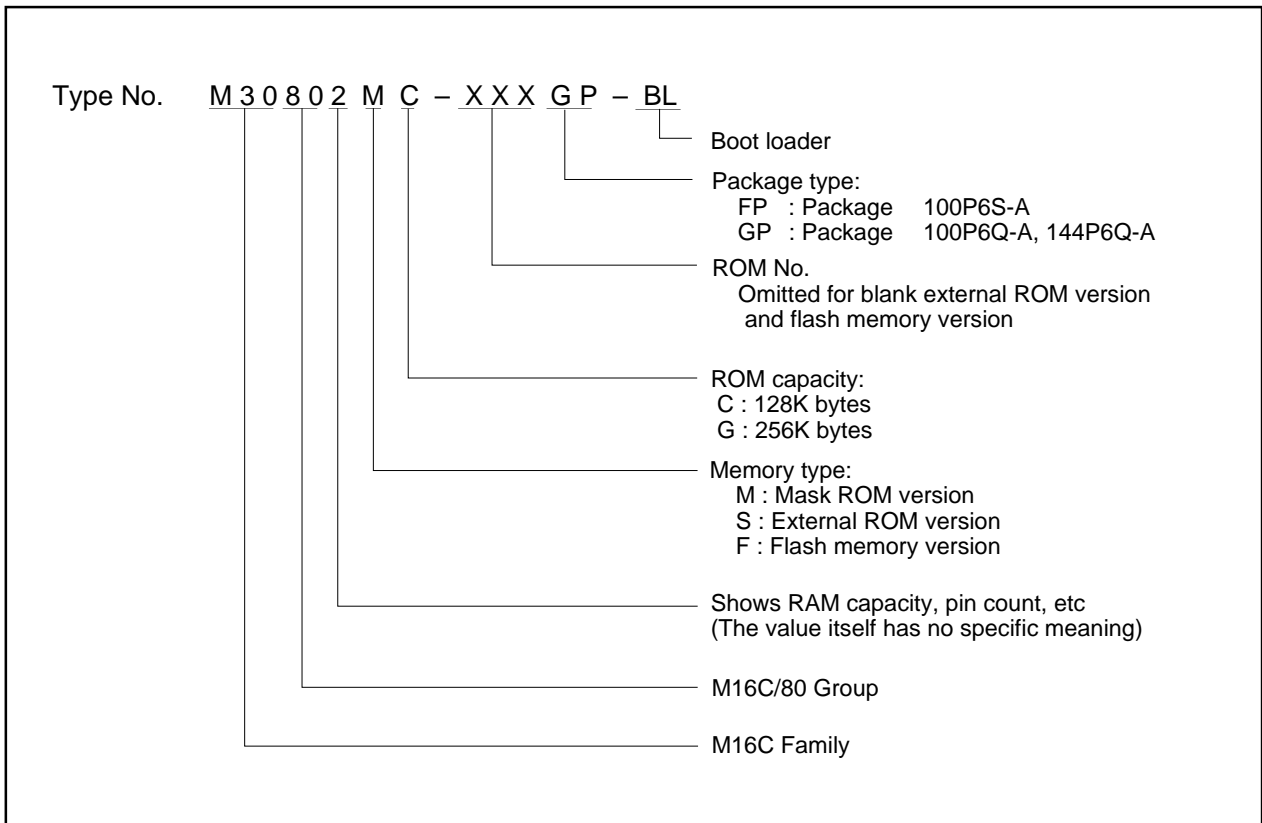


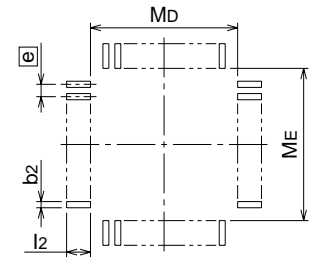
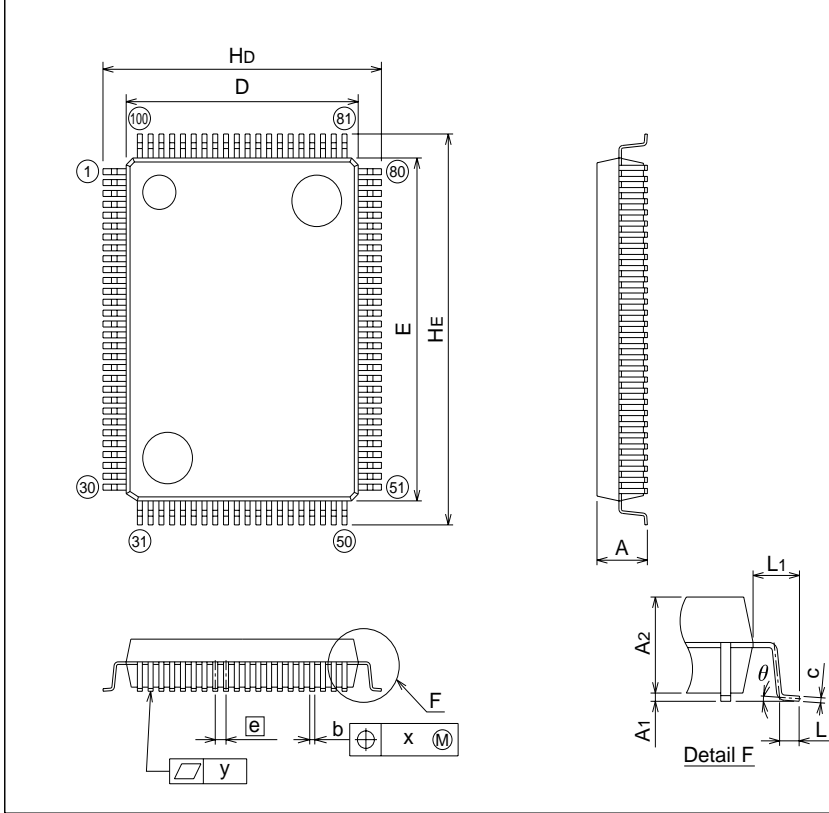
Figure 33.2 Type No., memory size, and package

Package Dimension

**100P6S-A** Recommended

Plastic 100pin 14X20mm body QFP

EIAJ Package Code	JEDEC Code	Weight(g)	Lead Material
QFP100-P-1420-0.65	-	1.58	Alloy 42

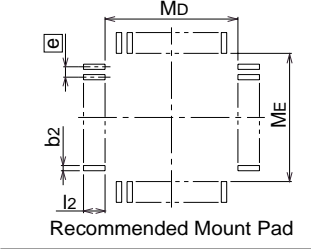
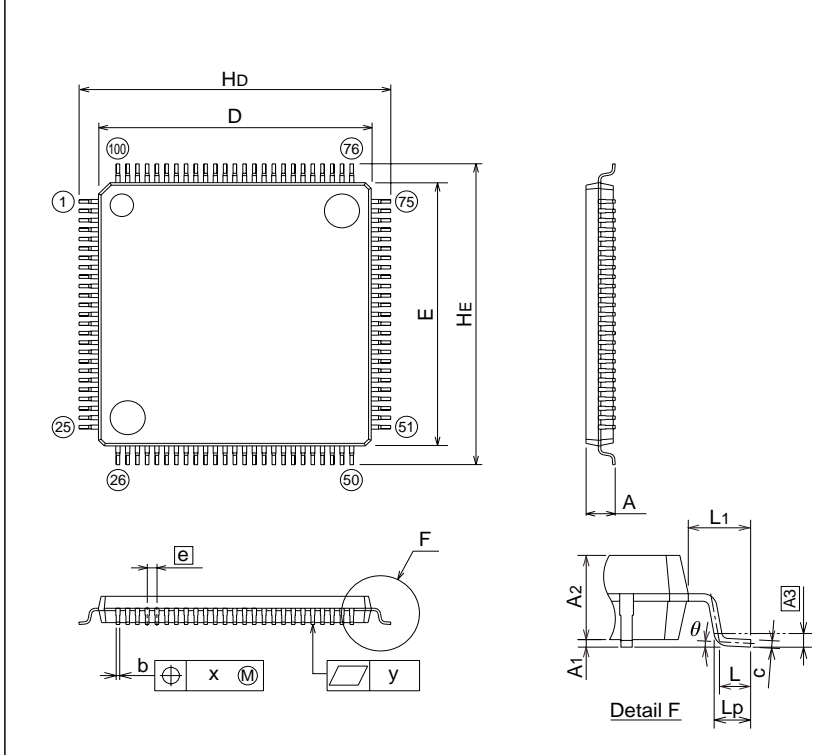


Symbol	Dimension in Millimeters		
	Min	Nom	Max
A	-	-	3.05
A1	0	0.1	0.2
A2	-	2.8	-
b	0.25	0.3	0.4
c	0.13	0.15	0.2
D	13.8	14.0	14.2
E	19.8	20.0	20.2
e	-	0.65	-
Hd	16.5	16.8	17.1
HE	22.5	22.8	23.1
L	0.4	0.6	0.8
L1	-	1.4	-
x	-	-	0.13
y	-	-	0.1
theta	0°	-	10°
b2	-	0.35	-
l2	1.3	-	-
MD	-	14.6	-
ME	-	20.6	-

**100P6Q-A** Recommended

Plastic 100pin 14X14mm body LQFP

EIAJ Package Code	JEDEC Code	Weight(g)	Lead Material
LQFP100-P-1414-0.50	-	0.63	Cu Alloy

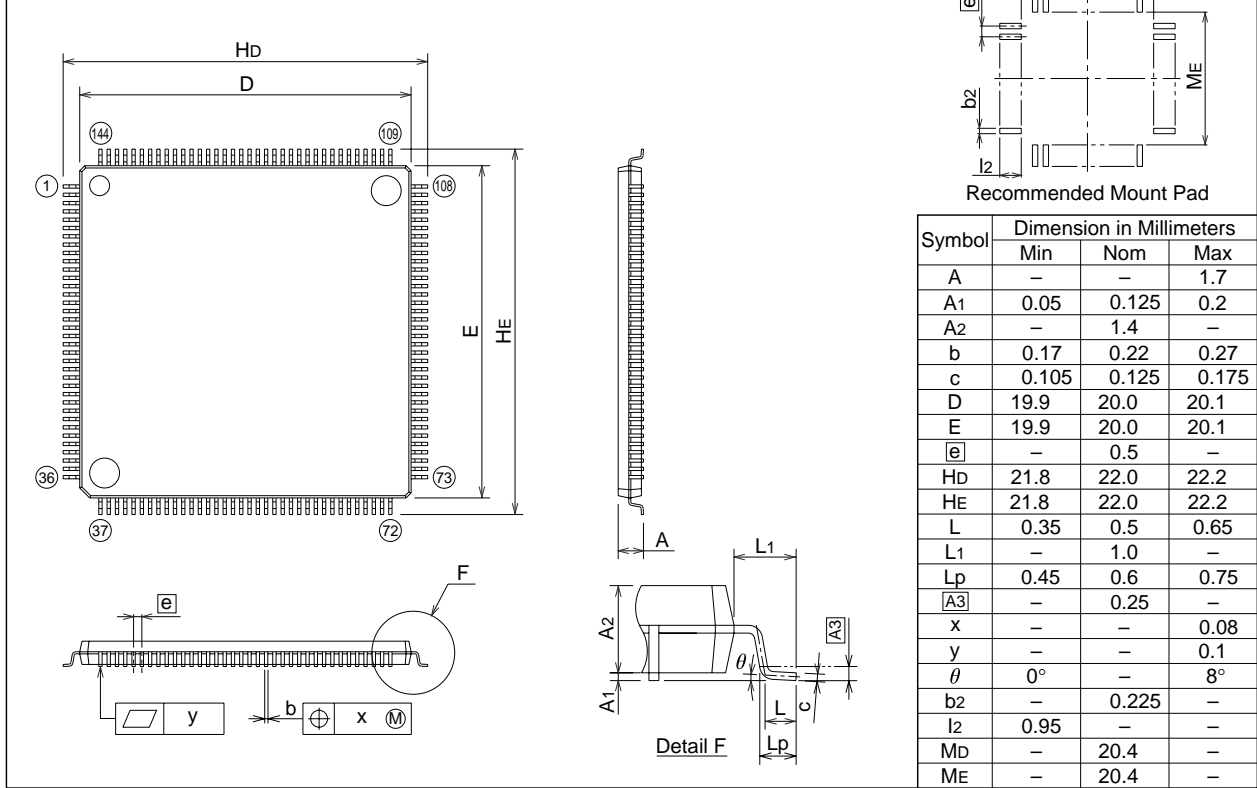


Symbol	Dimension in Millimeters		
	Min	Nom	Max
A	-	-	1.7
A1	0	0.1	0.2
A2	-	1.4	-
b	0.13	0.18	0.28
c	0.105	0.125	0.175
D	13.9	14.0	14.1
E	13.9	14.0	14.1
e	-	0.5	-
Hd	15.8	16.0	16.2
HE	15.8	16.0	16.2
L	0.3	0.5	0.7
L1	-	1.0	-
Lp	0.45	0.6	0.75
A3	-	0.25	-
x	-	-	0.08
y	-	-	0.1
theta	0°	-	10°
b2	-	0.225	-
l2	0.9	-	-
MD	-	14.4	-
ME	-	14.4	-

**144P6Q-A** Recommended

**Plastic 144pin 20×20mm body LQFP**

EIAJ Package Code	JEDEC Code	Weight(g)	Lead Material
LQFP144-P-2020-0.50	—	1.23	Cu Alloy



## Register Index

### A

AD0 to AD7 169  
 ADCON0 168, 170, 171, 172, 173, 174  
 ADCON1 168, 170, 171, 172, 173, 174  
 ADCON2 169  
 ADIC 63  
 AIER 73

### B

BCN2IC to BCN4IC 63

### C

CM0 46, 79  
 CM1 46  
 CPSRF 96, 107  
 CRCD 178  
 CRCIN 178

### D

DA0 to DA1 177  
 DACON 177  
 DCT0 to DCT3 84  
 DM0IC to DM3IC 63  
 DM0SL to DM3SL 82  
 DMA0 to DMA3 85  
 DMD0 to DMD1 83  
 DRA0 to DRA3 85  
 DRAMCONT 183  
 DRC0 to DRC3 84  
 DS 31  
 DSA0 to DSA3 85  
 DTT 113

### F

FMR0 to FMR1 276

### I

ICTB2 113  
 IDB0 to IDB1 113  
 IFSR 71  
 INT0IC to INT5IC 63  
 INVC0 to INVC1 112

### K

KUPIC 63

### M

MCD 47

### O

ONSF 96

### P

P0 to P10 197  
 P11 198  
 P12 to P13 197  
 P14 198  
 P15 197  
 PCR 206  
 PD0 to PD10 195  
 PD11 196  
 PD12 to P13 195  
 PD14 196  
 PD15 195  
 PM0 27  
 PM1 28  
 PRCR 55  
 PS0 to PS1 200  
 PS2 to PS3 201  
 PSC 203  
 PSL0 PSL2 202  
 PSL3 203  
 PUR0 to PUR2 204  
 PUR3 to PUR4 205

### R

REFCNT 185  
 RLVL 49, 64  
 RMAD0 to RMAD3 73  
 ROMCP 288

### S

S0RIC to S4RIC 63  
 S0TIC to S4TIC 63

**T**

TA0,TA3 **95**  
TA0IC to TA4IC **63**  
TA0MR to TA4MR **94, 97, 98, 103, 104**  
TA1,TA2,TA4 **95, 114**  
TA11,TA21,TA41 **114**  
TA1MR,TA2MR,TA4MR **115**  
TA2MR to TA4MR **100**  
TABS **95, 107, 114**  
TB0,TB1,TB3,TB4,TB5 **107**  
TB0IC to TB5IC **63**  
TB0MR to TB5MR **106, 108, 109, 110**  
TB2 **107, 114**  
TB2MR **115**  
TBSR **107**  
TRGSR **96, 114**

**U**

U0BRG to U4BRG **129**  
U0C0 to U2C0 **131**  
U0C1 to U4C1 **133**  
U0MR to U4MR **130, 139, 146**  
U0RB to U4RB **129**  
U0TB to U4TB **129**  
U2SMR to U4SMR **134, 156**  
U2SMR2 to U4SMR2 **135, 156**  
U2SMR3 **136**  
U3C0 to U4C0 **132**  
U3SMR3 to U4SMR3 **136, 163**  
UCON **134**  
UDF **95**

**W**

WCR **40**  
WDC **79**  
WDTS **79**

**X**

X0R to X15R **181**  
XYC **180**

**Y**

Y0R to Y15R **181**

Version	Contents for change	Revision date
REV.B	<ul style="list-style-type: none"> <li>• Page 1 line 5      1 M byte --&gt; 16 M bytes</li> <li>• Page 1 line 15     10 MHz with software one wait --&gt; 10 MHz : under planning</li> <li>• Page 1 line 16     35 mW (f(XIN)=20MHz, without software wait, Vcc=5V; M30800MC-XXXXFP target value ) --&gt; 45 mA (M30800MC-XXXXFP)</li> <li>• Page 1      X-Y converter ---- 1 circuit Addition</li> <li>• Page 4 line 28     35 mA --&gt; 45 mA</li> <li>• Page 6 figure 1.1.4</li> <li>• Page 18 figure 1.5.4 and corresponding pages <ul style="list-style-type: none"> <li>(106) Peripheral subfunction select register --&gt; Function select register C</li> <li>(107) Port function select register 0 --&gt; Function select register A0</li> <li>(108) Port function select register 1 --&gt; Function select register A1</li> <li>(109) Peripheral function select register 0 --&gt; Function select register B0</li> <li>(110) Peripheral function select register 1 --&gt; Function select register B1</li> <li>(111) Port function select register 2 --&gt; Function select register A2</li> <li>(112) Port function select register 3 --&gt; Function select register A3</li> <li>(113) Peripheral function select register 2 --&gt; Function select register B2</li> </ul> </li> <li>• Page 21 figure 1.6.3      Register name change same as figure 1.5.4</li> <li>• Page 24 figure 1.8.1      Processor mode register 0 <ul style="list-style-type: none"> <li>Note 6 --&gt; Note 7, Note 6 Addition</li> <li>Processor mode register 1 Note 3</li> </ul> </li> <li>• Page 31 line 4 <ul style="list-style-type: none"> <li>Addition: The ALE signal is occurred regardless of internal area and external area.</li> </ul> </li> <li>• Page 31 table 1.10.4, Page 33 table 1.10.5      R/W --&gt; RD/WR</li> <li>• Page 42 table 1.11.4      System clock control register 0 Note 2</li> <li>• Page 51 line 7, table 1.15.1 <ul style="list-style-type: none"> <li>port function select register 3 (address 03B516) --&gt; port function select register 3 (address 03B516) and D/A control register (address 039C16)</li> </ul> </li> <li>• Page 60 line 3      the interrupt occurs. --&gt; the interrupt can be set to occur on input signal level and input signal edge.</li> <li>• Page 65 line 10      Set register --&gt; When writing to DCT2, DCT3, DRC2, DRC3, DMA2 and DMA3, set register</li> <li>• Page 67 table 1.20.2      Addition: Note 5</li> <li>• Page 86 line 1      successively when --&gt; successively two times when</li> <li>• Page 93 line 16      Count source input --&gt; Count source input (Set the corresponding function select register A to I/O port.)</li> <li>• Page 114 table 1.25.6, page 122 table 1.26.1 <ul style="list-style-type: none"> <li>UARTi transmit/receive mode register      Addition: Note 2</li> <li>UARTi transmit/receive mode register      Addition: Note 3</li> </ul> </li> <li>• Page 115 table 1.25.7      UARTi transmit/receive mode register 0      Delate: Note 3</li> <li>• Page 120 line 13      Addition: -Set the corresponding function select register A to I/O port</li> <li>• Page 123 table 1.26.3</li> <li>• Page 130 table 1.27.3</li> <li>• Page 139 figure 1.29.2</li> <li>• Page 142 line 2      062316 --&gt; 032616</li> <li>• Page 144 table 1.29.5</li> <li>• Page 156 table 1.31.2      D/A control register (Note)      Addition: Note</li> <li>• Page 164 figure 1.34.2      16-bit bus mode A9 --&gt; A9</li> <li>• Page 165 line 5      f32 --&gt; BCLK(frequency x 32)</li> </ul>	'98. 10.19

Version	Contents for change	Revision date
REV.B	<ul style="list-style-type: none"> <li>• Page 165 figure 1.34.3 operation clock --&gt; BCLK</li> <li>• Page 169               <ul style="list-style-type: none"> <li>they function as output regardless of the contents of the direction registers. When pins are to be used as the outputs for the D/A converter, do not set the direction registers to output mode.</li> <li>--&gt;</li> <li>set the corresponding function select registers A, B and C. When pins are to be used as the outputs for the D/A converter, set the function select register of each pin to I/O port, and set the direction registers to input mode.</li> <li>Table 1.35.1 lists each port and peripheral function.</li> </ul> </li> </ul>	'98.10.19
REV.C	<p>All page M30800MC-XXXFP --&gt; M16C/80 (100-pin version) group</p> <p>Page 2 Figure 1 changed, GP package is added</p> <p>Page 3 Figure 3 Note 1 and Note 2 is added</p> <p>Page 5 Figure 4, Table 2 New type no. is added</p> <p>Page 6 Figure 5 GP is added</p> <p>Page 10 Line 2 18 registers --&gt; 28 registers</p> <p>Page 11 (7) Set USP and ISP to an even number so that execution efficiency is increased. --&gt; added</p> <p>Page 17 Figure 11 (54) UART4 special mode register 3 --&gt; added</p> <p>Page 18 Figure 12 UART3 special mode register 3 --&gt; added UART2 special mode register 3 --&gt; added Function select register B3 --&gt; added</p> <p>Page 20 Figure 14 UART4 special mode register 3 --&gt; added UART3 special mode register 3 --&gt; added UART2 special mode register 3 --&gt; added</p> <p>Page 21 Figure 15 Function select register B3 --&gt; added</p> <p>Page 24 Figure 23 PM1 Note 4 --&gt;added</p> <p>Page 31 Figure 26</p> <p>Page 45 Table 14, Page 46 Table 15 Note --&gt; added</p> <p>Page 50 Figure 32-4 Changed</p> <p>Page 51 Line 6 port function select register 3 --&gt; function select register A3</p> <p>Page 52 Line 17 FFFFE4<sub>16</sub> to FFFFE7<sub>16</sub> are all --&gt; FFFFE7<sub>16</sub> is</p> <p>Page 53 Table 17 BRK instruction If the vector is --&gt; If the contents of FFFFE7<sub>16</sub> is</p> <p>Page 53 Table 18 Instruction fetch and DBC --&gt; delated</p> <p>Page 58 Figure 36 IPL --&gt; RLVL</p> <p>Page 61 Figure 38 004D<sub>16</sub> --&gt; 0093<sub>16</sub></p> <p>Page 67 Figure 44-1Note 3 and 6 --&gt; added</p> <p>Page 68 Figure 45 memory --&gt; memory (forward direction)</p> <p>Page 70 Figure 46-2DMAi memory address reload register Note: vector register (SVP) --&gt; save PC register (SVP)</p> <p>Page 84 Figure 56 Note 4 addresses 0342<sub>16</sub> and 0343<sub>16</sub> --&gt; address 0343<sub>16</sub></p> <p>Page 93 Table 30 Count source: TBj overflow --&gt; added</p> <p>Page 96 Figure 69 Three-phase PWM control register 0 Note 4:both bit 0 and 1 --&gt; bit 1</p>	'98.3.2



Version	Contents for change	Revision date
REV.C	<p>Three-phase PWM control register 1</p> <p>Page 100 Line 1 In three-phase --&gt; In "L" active output polarity in three-phase</p> <p>Page 100 Line 26,31 the state of set by port direction register --&gt; the high-impedance state</p> <p>Page 101 Figure 73 Right: INV14 --&gt; added</p> <p>Page 103 Figure 74</p> <p>Page 108 Table 32 UART4 LSB first/MSB first selection : Note 1 --&gt; Note 2</p> <p>Page 118 Figure 83 UART transmit/receive control register 2</p> <p>Page 119 Figure UART 3,4 special mode register 3 --&gt; added</p> <p>Page 126 Line 3 CLK and CLKS select bit (bits 4 and 5 at address 037016) --&gt; port function select register (bits of related to-P64 and P65)</p> <p>Page 145</p> <p>Page 176 Table 124 P91: STxD3 output --&gt; added P97: STxD4 output --&gt; added</p> <p>Page 178 Figure 125-2 Function select register A3</p> <p>Page 179 Figure 125-3 Function select register B0</p> <p>Page 180 Figure 125-4 Function select register B3</p> <p>Page 187 A/D Converter (5)</p> <p>Page 188 DMAC</p>	'98.3.2
	<p>Page1 Supply voltage 4.0V-5.5V, Mask ROM version is added.</p> <p>Page 5 Table 1.1.1 DMAC 2 channels --&gt; 4 channels</p> <p>Page 8 P0 description is changed</p> <p>Page 9 P6 description is changed P7, 8, 9, 10 equivalent to P0 --&gt; P6</p> <p>Page 10 Figure 1.2.1 M30800FC, M30803FG are added.</p> <p>Page 18 Figure 1.4.3 (15) DRAM control register 0XXX0000 --&gt; ?XXX????</p> <p>Page 19 Figure 1.4.4 Delate Note, (143)-(147) 00 --&gt; ??</p> <p>Page 20 Figure 1.5.1 Add Note</p> <p>Page 25 Figure 1.6.1 Processor mode register 1 When reset 0016 --&gt; C016</p> <p>Page 30 Line 15 ... output to A9 to A20 --&gt; A8 to A20</p> <p>Page 32 Figure 1.7.2</p> <p>Page 35 (8) BCLK output</p> <p>Page 38 Figure 1.7.6</p> <p>Page 39 Figure 1.7.7</p> <p>Page 40 Figure 1.8.1 and 1.8.2 Note</p> <p>Page 42</p> <p>Page 43 Figure 1.8.4</p> <p>Page 44 Figure 1.8.5 Note 2, Line 6 Pin outputs "L" is delated.</p> <p>Page 45</p> <p>Page 47 Line 15 ... as BCLK --&gt; as BCLK from the interrupt routine</p> <p>Table 1.8.3</p> <p>Page 48 Status Transition of BCLK</p> <p>Page 51 Figure 1.8.7</p> <p>Page 52 Line 6, Figure 1.8.6 Delate D/A control register</p> <p>Page 56 Line 14</p>	'98.4.12

Version	Contents for change	Revision date
REV.C	<p>Page 58 Table 1.9.3 Software interrupt number 40,41, Add fault error, Add Note 2</p> <p>Page 59 Interrupt control register Line 4 delate</p> <p>Page 64 Interrupt sequence (1)</p> <p>Page 66 Saving registers Last line added</p> <p>Page 67 Interrupt Priority *1 delated, Last line added</p> <p>Page 72 (2) Setting the stack pointer Last line added</p> <p>Page 74 Watchdog timer Line 2  A watchdog timer interrupt is generated when --&gt; Whether a watchdog timer interrupt is generated or reset is selected when  Last part :Watchdog timer function select bit is initialized only at reset. After reset, watchdog timer interrupt is selected. added</p> <p>Page 75 Figure 75 System clock control register 0 added</p> <p>Page 97 Figure 1.13.9 Count value</p> <p>Page 181 Figure 1.25.4</p> <p>Page 182 Figure 1.25.5</p> <p>Page 131 Figure 1.16.12 Both register Note2 added</p> <p>Page 135 Table 1.17.3 RxDi bit 1 and 6 at address 03C716 --&gt; bit 1 and 7 ...</p> <p>Page 132 Table 1.18.3 RxDi bit 1 and 6 at address 03C716 --&gt; bit 1 and 7 ...</p> <p>Page 147 Figure 1.19.1 Upper figure changed, note added</p> <p>Page 153 Bit 4 overflow --&gt; underflow</p> <p>Page 154 Figure 1.20.3 overflow --&gt; underflow</p> <p>Page 159 Clock phase setting  UARTi transmission-reception control register 0 ..., whereas UARTi special mode register 3 ... --&gt; Bit 6 of UARTi transmission-reception control register 0 ..., whereas bit 1 of UARTi special mode register 3 ...  Line 15  ... output is high impedance. --&gt; ... output is indeterminate.</p> <p>Page 171 Line 3 Set the function select register A to I/O port and the direction register to input mode. added</p> <p>Page 171 Figure 1.22.2 Note delate</p> <p>Page 176 Figure 1.24.3 added</p> <p>Page 178 Figure 1.25.1 When reset --&gt; indeterminate, Note 4 is added.</p> <p>Page 200 Table 1.26.2 and 1.26.3 and Figure 1.26.14  CNVss is added</p> <p>Page 204- Electric characteristics added</p>	
Rev.C1	<p>Page 214 Table 1.28.22 th(BCLK-DW) add</p> <p>Page 220 Figure 1.28.6 th(BCLK-CAS) --&gt; th(BCLK-DW)</p> <p>Page 223 Figure 1.28.9 WR, WRL, WRH(sepalate bus) wave change</p>	99.5.12
Rev.C2	<p>Page 24 Line 3 A software reset has almost the same ... --&gt; A software reset has the same ...</p>	99.5.20
	<p>Page 161 Note 2:  When f(XIN) is over 10 MHz, the fAD frequency must be under 10 MHz by dividing. --&gt; addition</p>	99.6.4
	<p>Page 18 Figure 1.4.3 (60) Timer B3,4,5 count start flag value change</p> <p>Page 19 Figure 1.4.4 Flash memory control register 0 and 1 added</p> <p>Page 22 Figure 1.5.3 Flash memory control register 0 and 1 added</p>	99.7.6

Version	Contents for change	Revision date
	<p>Page 43 Figure 1.8.4 CM0 Note 5 delate</p> <p>Page 81 Figure 1.11.5 DMAi memory address reload register Address DRA2, DRA3 000000<sub>16</sub> --&gt; XXXXXX<sub>16</sub></p> <p>Page 181, 182 Figures 1.25.4-1.25.5 D0-D15 waveform changed</p> <p>Page 185 (6) Pull up control register changed</p> <p>Page 208 Table 1.28.3 VT+VT- TB0IN-TB2IN --&gt; TB0IN-TB5IN, TA2OUT-TA4OUT --&gt; TA0OUT-TA4OUT</p> <p>Page 211 Table 1.28.19</p> <p>Page 212 Table 1.28.20</p> <p>Page 213 Table 1.28.21</p> <p>Page 214 Table 1.28.22</p> <p>Page 216 Figure 1.28.2</p> <p>Page 217 Figure 1.28.3</p> <p>Page 218 Figure 1.28.4</p> <p>Page 219 Figure 1.28.5</p> <p>Page 220 Figure 1.28.6</p> <p>Page 221 Figure 1.28.7</p> <p>Page 223 Figure 1.28.9</p>	
Rev.C3	Flash memory ROM version added	99.9.24
	Page 2,3 Figure 1.1.1, 1.1.2 Japanese font change to English font	99.12.8
Rev.D	<p>Page 1</p> <ul style="list-style-type: none"> <li>• DMAC...4 channels (trigger: 24 sources) --&gt; 31 sources</li> <li>• Supply voltage 4.2 to 5.5V (f(XIN)=20MHz) Flash memory version--&gt; addition</li> <li>• Interrupt...4 software --&gt; 5 software</li> </ul> <p>Page 1,5 Table 1.1.1 Feature • Memory capacity ROM 128 Kbytes --&gt; (See ROM expansion figure.) RAM 10K --&gt; 10/20 Kbytes</p> <p>Page 5 Table 1.1.1 Interrupt...4 software --&gt; 5 software</p> <p>Page 2, 3 Figure 1.1.1, 1.1.2 Note 1 addition</p> <p>Page 6 Figure 1.1.4, Table 1.1.2 M30803MG-XXXFP/GP addition</p> <p>Page 7 Figure 1.1.5 ROM capacity G:256 Kbytes addition</p> <p>Page 8 P00 to P07 However, it is possible to select pull-up resistance presence to the usable port as I/O port by setting. --&gt; addition CNVss Connect it to the Vss pin when operating in single-chip or memory expansion mode. Connect it to the Vcc pin when in microprocessor mode. --&gt; Connect it to the Vss pin when operating in single-chip or memory expansion mode after reset. Connect it to the Vcc pin when in microprocessor mode after reset. BYTE When operating in single-chip mode,connect this pin to VSS. --&gt; When not using the external bus,connect this pin to VSS.</p> <p>Page 9 P50 to P57 In single chip mode, --&gt; delate</p> <p>Page 10 Figure 1.2.1 M30803FG --&gt; M30803MG/FG</p> <p>Page 13 Figure 1.4.3 (2) processor mode register C0<sub>16</sub> --&gt; 00<sub>16</sub></p> <p>Page 20 to 23 Figure 1.5.1 to 1.5.4 Note addition</p> <p>Page 23 Figure 1.5.4 Note 2 addition</p>	14/3/00

Version	Contents for change	Revision date
	<p>Page 25 Figure 1.6.1, 1.6.2 Figure 1.6.1 is divided to Figure 1.6.1 and 1.6.2</p> <p>Page 30 Table 1.7.4</p> <p>Page 34 Figure 1.7.3 Note addition</p> <p>Page 36 Line 3 the chip select control register --&gt; the wait control register</p> <p>Page 38, 39 Figure 1.7.6, 1.7.7 Note change</p> <p>Page 42 Line 7 addition</p> <p>When the main clock is stopped (bit 5 at address 0006<sub>16</sub> =1) or the mode is shifted to stop mode (bit 0 at address 0007<sub>16</sub> =1), the main clock division register (address 000C<sub>16</sub>) is set to the divided-8 mode.</p> <p>Page 42 (3)BCLK When shifting to stop mode, --&gt; When main clock is stopped or shifting to stop mode,</p> <p>Page 43 Figure 1.8.4 CM0 Note 6 change, Note 7, 8 addition, CM1 Note 4 addition</p> <p>Page 44 Figure 1.8.5 Note 2 change</p> <p>Page 48 Line 5 When shifting to stop mode and reset, --&gt; When shifting to stop mode, reset or stopping main clock,</p> <p>(12) Low power dissipation mode addition</p> <p>When the main clock is stopped, the main clock division register (address 000C<sub>16</sub>) is set to the division by 8 mode.</p> <p>Page 51 Figure 1.8.7. Clock transition Note 3, 4 addition</p> <p>Page 52 Line 9 addition</p> <p>Page 54 Software Interrupts (2) Overflow interrupt, "CMPX" addition</p> <p>Page 55 (2) Peripheral I/O interrupts</p> <ul style="list-style-type: none"> <li>• Bus collision detection/start, stop condition (UART2, UART3, UART4) interrupts --&gt; change</li> </ul> <p>Page 57 • Variable vector tables addition</p> <p>Set an even address to the start address of vector table setting in INTB so that operating efficiency is increased.</p> <p>Page 58 Table 1.9.3</p> <p>Bus collision detection/start, stop condition interrupts --&gt; Bus collision detection, start/stop condition detection interrupts</p> <p>Page 58 Table 1.9.3, page 68 Figure 1.9.8</p> <p>Software interrupt number 40, 41 fault error --&gt; addition</p> <p>Page 71 Address match interrupt Line 7 addition</p> <p>Page 72 (3) The NMI interrupt</p> <ul style="list-style-type: none"> <li>• Do not reset the CPU with the input to the <math>\overline{\text{NMI}}</math> pin being in the "L" state. --&gt; • Signal of "L" level width more than 1 clock of CPU operation clock (BCLK) is necessary for <math>\overline{\text{NMI}}</math> pin.</li> </ul> <p>Page 72 (4) External interrupt</p> <p>Page 74 Figure 1.10.1</p> <p>Page 76 Line 2</p> <p>"DMAC is a function that to transmit 1 data of a source address (8 bits /16 bits) to a destination address when transmission request occurs." addition.</p> <p>Page 76 Line 12 addition</p> <p>When writing to DSA2 and DSA3, set register bank select flag (B flag) to "1" and use LDC instruction to set SB and FB registers.</p> <p>Page 76 Figure 1.11.1</p> <p>Page 77 Table 1.11.1 Transfer memory space (16 Mbyte space) --&gt; addition</p>	

Version	Contents for change	Revision date
	<p>Page 78 Figure 1.11.2 Note :6 OR instruction --&gt; OR instruction etc.</p> <p>Page 80 Figure 1.11.4 DRCi • Transfer counter --&gt; • Transfer count register</p> <p>Page 81 Figure 1.11.5 DMAi, DSAi, DRAi Transfer count specification "(16 Mbytes area)" addition DRAi memory address counter --&gt; memory address register</p> <p>Page 85 Line 9 addition (1) Internal factors, (2) External factors change</p> <p>Page 87 Fugure 1.12.1 "Timer B2 overflow" addition</p> <p>Page 88 Fugure 1.12.2 Timer A --&gt; Timer B2 overflow (to timer A count source)</p> <p>Page 93 Table 1.13.2 Cout source • TB2 overflows, TAj overflows --&gt; •TB2 overflows or underflows , TAj overflows or underflows</p> <p>Page 95 Figure 1.13.7 When using two-phase signal processing Note 3 --&gt; addition</p> <p>Page 102 Figure 1.14.3 TBSR When reset 00<sub>16</sub> --&gt; 000XXXXXX<sub>16</sub> b4-b0 When read, the value is "0" --&gt; indeterminate</p> <p>Page 104 Table 1.14.2 Cout source • TBj overflows --&gt; •TBj overflows or underflows</p> <p>Page 124 Figure 1.16.5 UiTB Note 1 delate</p> <p>Page 126-127 Figure 1.16.7 to 1.16.8 CRD change</p> <p>Page 130 Figure 1.16.11 SDHI Enabled &lt;--&gt; Disabled</p> <p>Page 144 (a) Separate CTS/RTS pins function (UART0)</p> <p>Page 146 Table 1.19.1 Addition in "Other things"</p> <p>Page 147 Figure 1.19.1 è„Figure A "L" level returns from TxD due to the occurrence of a parity error. --&gt; A "L" level returns from SIM card...</p> <p>Page 149 Figure 1.19.4 Note addition</p> <p>Page 150 Table 1.20.1 Note 1: LSB first --&gt; MSB first, Note 3 Change</p> <p>Page 156 Figure 1.20.4 4 to 5 cycles --&gt; 3 to 6 cycles</p> <p>Page 163, 165-169 Figure 1.21.2-Figure 1.21.8 ADCON1 Note 2-6 addition</p> <p>Page 170 Line 14,23 addition</p> <p>Page 171 Line 5 addition</p> <p>Page 172 Figure 1.22.3 Note :3 D/A control register --&gt; D/A register</p> <p>Page 176 Figure 1.24.3</p> <p>Page 178 Figure 1.25.1 Note 1 position change</p> <p>Page 178 Line 10 DRAM controler --&gt; addition</p> <p>Page 179 Figure 1.25.2 Note 1 --&gt; change</p> <p>Page 184 (1) Direction registers, (2) Port registers --&gt; change</p> <p>Page 185 (4) Function select register B --&gt; change</p> <p>Page 189 Figure 1.26.4 Port Pi direction register Note 2 addition</p> <p>Page 190 Figure 1.26.5 Port Pi register Note 1 and 2 addition</p> <p>Page 191 Table 1.26.1 Note addition</p> <p>Page 192 Figure 1.26.6 Function select register A1 Note 1 addition</p> <p>Page 194 Figure 1.26.8 Function select register B1 Note 2 addition</p> <p>Page 195 Figure 1.26.9 Function select register B3 Note 1 --&gt; addition, PSL3_3-PSL3_6 change</p> <p>Page 196 Figure 1.26.13 Port control register Note 2 addition</p> <p>Page 197 Figure 1.26.4 Port Pi direction register Note 2 addition</p> <p>Page 200 Precaution on A/D converter (6) --&gt; addition</p> <p>Page 203 Stop Mode and Wait Mode (2) all clock stop bits --&gt; all clock stop control bits</p> <p>Page 203 Noise addition</p> <p>Page 203 Precaution on interrupt (1) line 7 --&gt; addition</p>	

Version	Contents for change	Revision date
	<p>Page 204 Making power consumption electricity small --&gt; addition  Page 207 Table 1.28.3 <math>V_{T+} - V_{T-}</math> SCL2-SCL4, SDA2-SDA4 Addition  Page 208 Table 1.28.5 Note Change  Page 215 Table 1.28.22 <math>t_{RP}</math> expression change  Page 217-220 Figure 1.28.2-1.28.5 <math>t_{w(WR)}</math> addition, <math>t_{h(BCLK-DB)}</math> delate  Page 219, 220, 222, 223, 225  Figure 1.28.4, 1.28.5, 1.28.7, 1.28.8, 1.28.10 addition  Page 225, 226 Figure 1.28.10, 1.28.11 <math>t_{h(BCLK-DB)}</math> -5 ns.min --&gt; -7 ns.min  Page 227 Figure 1.28.12 Refresh timing (self refresh) RAS timing  Page 230 3V of electric characteristics addition  Page 246 Table 1.29.1 Data hold --&gt; addition  Page 247 Figure 1.29.2 Package type 144P6Q --&gt; 144P6Q-A  Page 248 Flash memory line 5 change  Page 250 Function outline Line 24 (Parallel ... function) --&gt; delate  Page 269 Standard serial I/O mode Line 26 external device --&gt; external device ( programmer)  Page 285 Figure1.31.21 programer --&gt; peripheral unit ( programmer)</p>	
Rev.D3	<p>Page 43 Figure1.8.4 Note of the system clock control register 0--&gt;addition  Page 44 Line 4 Note--&gt;addition  Page 45 Table1.8.2 Note--&gt;addition  Page 71 Line 9 "Address match interrupt is not generated with a start instruction of interrupt routine."--&gt;Delete  Page 73 (6) Precaution of Address mach interrupt--&gt;addition  Page 79 Figure1.11.2 Note--&gt;change  Page 87 Precaution for DMAC--&gt;addition  Page 131 Figure1.16.11 Bit 7--&gt;Must set to "1" in selecting IIC mode.  Page 152 Figure1.20.1 Bit 7--&gt;Must set to "1" in selecting IIC mode.  Page 182 Addition  Page 205 (3) Address match interrupt in Interrupt precautions--&gt;addition  Page 206 (2) DMAC--&gt;addition  Page 207 Precautions for using CLKOUT pin--&gt;addition  Page 210 Table1.28.3 <math>I_{CC}</math> when clock stop <math>T_{opr}=25C^{\circ}</math>--&gt;change  Page 212 Table1.28.6 External clock input HIGH and LOW pulse waidth 22--&gt;20  External clock rise and fall time 10--&gt;5  Page 215, 216 Table1.28.19, 20 <math>t_{h(BCLK-DB)}</math>--&gt;delete, <math>t_{w(WR)}</math>--&gt;addition  Page 218 Table1.28.22 <math>t_{h(BCLK-DB)}</math> -5ns --&gt; -7ns  Page 233 Table1.28.23 <math>I_{CC}</math> when clock stop <math>T_{opr}=25C^{\circ}</math>--&gt;change  Page 235 Table1.28.27 <math>t_{h(CAS-DB)}</math>--&gt;addition  Page 238, 239 Table1.28.39, 40 <math>t_{w(WR)}</math> --&gt;addition, <math>t_{h(BCLK-RD)}</math> 0ns--&gt;-3ns  Page 240 Table1.28.41 <math>t_d(AD-ALE)=10^9/(f(BCLK)X2)-20</math> --&gt;<math>10^9/(f(BCLK)X2)-27</math>  Page 241 Table1.28.42 <math>t_{h(BCLK-CAS)}</math> 0ns--&gt;-3ns  Page 242 Figure1.28.15 <math>t_{ac1(RD-DB)}</math> min--&gt;max, <math>t_{ac1(AD-DB)}</math> min--&gt;max  Page 243 Figure1.28.16 <math>t_{ac2(RD-DB)}</math> min--&gt;max, <math>t_{ac2(AD-DB)}</math> min--&gt;max  Page 244, 255 Figure1.28.17 2 wait, Figure1.28.18 3 wait--&gt;addition  Page 246 Figure1.28.19 <math>t_{ac3(AD-DB)}</math>--&gt;addition, <math>t_{su(DB-RD)}</math>--&gt;<math>t_{su(DB-BCLK)}</math>, <math>t_{h(BCLK-RD)}</math> 0ns --&gt;-3ns, <math>t_d(AD-ALE)=(tcyc/2-20)ns</math>--&gt; ... -27)ns  Page 247 Figure1.28.20 Addition  Page 248, 249 Figure1.28.21, 1.28.22 --&gt;addition</p>	19/6/'00





Version	Contents for change	Revision date
	<p>ing) --&gt; "Note 2 Timer A2 is fixed to ... multiply-by-4 processing operation." is added, note 3 change</p> <p>Page 109 Figure 1.14.6 Note 1 It is indeterminate when reset --&gt; addition</p> <p>Page 112 Figure 1.15.2 Dead time timer--&gt;Note 1 addition, Timer B2 interrupt occurrence frequency set counter--&gt;Note 3 addition</p> <p>Page 113 Notes 2, 3 --&gt; addition</p> <p>Page 114 three-phase waveform mode --&gt; three phase PWM output mode</p> <p>Page 128 Figure 1.16.5 UARTi bit rate generator --&gt; Note 2 addition</p> <p>Page 128 Figure 1.16.5 UARTi transmit buffer register, UARTi bit rate generator--&gt;Note 1 addition</p> <p>Page 129 Figure 1.16.6 UARTi transmit/receive mode register--&gt;Note 2 addition in CKDIR of UART mode</p> <p>Page 133 Figure 1.16.10 UART transmit/receive control register 2--&gt;Note delete</p> <p>Page 136 Note 2, Page 143 Note 3 ... the UARTi receive interrupt request bit <u>is not set to "1"</u> --&gt; ... the UARTi receive interrupt request bit <u>will not change</u></p> <p>Page 145 Figure 1.18.1 UARTi transmit/receive mode register (i=0,1) --&gt; Note 1 addition, UARTi transmit/receive mode register (i=2 to 4) --&gt; Note 2 addition</p> <p>Page 157 On the 12th line, ... allocated to bit <u>3</u> in UART2 transmission buffer register <u>1</u> (address 033E16) ... --&gt; ... allocated to bit <u>11</u> in UART2 transmission buffer register (address 033E16) ...</p> <p>Page 161 &lt; Master Mode (TxDi and RxDi are selected, DINC = 0) &gt;  ..., and the <u>STxDi, SRxDi</u> and CLKi pins ...--&gt; ..., and the <u>TxDi, RxDi</u> and CLKi pins ...</p> <p>Page 165 Table 1.21.1 Absolute precision --&gt; change</p> <p>Page 170 Table 1.21.3 Reading of result of A/D converter --&gt; (at any time) addition</p> <p>Page 173 Table 1.21.6 Input pin --&gt; change to AN0 to AN7, With emphasis on the pin --&gt; addition</p> <p>Page 182 On the second line from the bottom, ..., and dummy cycle for refresh ... --&gt; ..., and processing necessary for dummy cycle to refresh DRAM ...</p> <p>Page 183 Figure 1.25.2 is partly revised.</p> <p>Page 189 On the 18th and 27th lines, page 194 Port Pi direction register Note 2, page 196 Port Pi register Note 1 ... for setting of bus control such as address bus and data bus is ... --&gt; of pins A0 to A22, <math>\overline{A}23</math>, D0 to D15, MA0 to MA12, <math>\overline{CS}0</math> to <math>\overline{CS}3</math>, <math>\overline{WRL}/\overline{WR}/\overline{CASL}</math>, <math>\overline{WRH}/\overline{BHE}/\overline{CASH}</math>, <math>\overline{RD}/\overline{DW}</math>, <math>\overline{BCLK}/\overline{ALE}/\overline{CLKOUT}</math>, <math>\overline{HLDA}/\overline{ALE}</math>, <math>\overline{HOLD}</math>, <math>\overline{ALE}/\overline{RAS}</math>, and <math>\overline{RDY}</math> are ...</p> <p>Page 203 Figure 1.26.13 is partly revised.</p> <p>Page 207, 208 Timer A (event counter mode) --&gt; (3) addition, Timer A (one-shot timer mode) --&gt; (2) changes to (3), (2) and (4) addition</p> <p>Page 209 Timer B (pulse period/pulse width measurement mode) --&gt; (3) addition</p> <p>Page 212 to 214 (2) <math>\overline{NMI}</math> interrupt •The <math>\overline{NMI}</math> pin also serves as P85, ... •Signal of "L" level ... --&gt; addition</p> <p>(3) Address match interrupt From "• To return from an interrupt..." to "; Interrupt completed" on page 77 --&gt; addition</p> <p>(4) External interrupt, (5) Rewrite the interrupt control register --&gt; addition</p> <p>Page 215 Explanation of (3) is partly revised.</p> <p>Page 215 <math>\overline{HOLD}</math> signal --&gt; addition</p> <p>Page 216 DRAM controller --&gt; addition</p>	



Version	Contents for change	Revision date
	<p>Page 217 Setting the registers, Notes on the microprocessor mode ... single-chip mode - -&gt;addition</p> <p>Page 217 Explanation of note on Flash memroy version is added.</p> <p>Page 219 Note 2 80mA --&gt; -80mA</p> <p>Page 220 Table 1.28.3 Ta --&gt; Topr, Note2 --&gt; addition</p> <p>Pages 220, 243 Tables 1.28.3, 1.28.23 Icc Power supply current ROMless version --&gt; addition, Ta --&gt; Topr, Note 2 --&gt; addition</p> <p>Page 227 Calculation for td(AD-ALE) is partly revised.</p> <p>Page 234, 235 Figures 1.28.6 and 1.28.7 Timing for td(AD-ALE) is partly revised.</p> <p>Page 243 Table 1.28.23 Topr=25°C, when clock is stopped: 2.0µA --&gt; 1.0µA, Notes 1, 2 --&gt; addition</p> <p>Page 250 Table 1.28.41 th(BCLK-RD) Min. 0 ns --&gt; -3 ns</p> <p>Pages 251, 258 to 262 Table 1.28.42, figures 1.28.21 to 1.28.25 th(BCLK-CAS): -3 ns --&gt; 0 ns</p> <p>Pages 251, 259, 261 Table 1.28.42, figures 1.28.22, 1.28.24 th(BCLK-DW): 0 ns --&gt; -3 ns</p> <p>Page 266 Table 1.29.2 M30805FGGP RAM capacity <u>24</u> Kbytes --&gt; <u>20</u> Kbytes</p> <p>Page 270 Figure 1.30.1 Flash memory control register 0 Note 1 Also write to this bit ... "H" level. --&gt; addition</p> <p>Page 273 (3) Disabling erase or ... serial I/O mode --&gt; delete, (7), (8) --&gt; addition</p> <p>Page 285 Note --&gt; addition</p> <p>Page 288 --&gt; addition</p> <p>Pages 291, 307 Tables 1.31.1, 1.31.5 Note 2 ... status register 1 data --&gt; ... status register data 1</p> <p>Page 319 144P6Q-A version --&gt; addition</p>	
Rev.E1	Page 32 Table 1.7.3 --> change	16/03/'01
Rev.E2	<p>Page 28 Figure 1.6.1 Note 8 --&gt; addition</p> <p>Page 88 Table for "Coefficient j, k" is partly revised.</p>	10/05/'01
Rev.E3	<p>Page 7 Figure 1.1.5 and Table 1.1.2, product names --&gt; added</p> <p>Page 8 Figure 1.1.6, Boot loader (BL) --&gt;addition</p> <p>Page 85 Figure 1.11.5, DMAi SFR address register, Note 2, <u>destination</u> fixed address --&gt; <u>source</u> fixed address, <u>source</u> fixed address --&gt; <u>destination</u> fixed address</p> <p>Page 218 Precaution of boot loader --&gt; addition</p> <p>Page 319 Appendix boot loader --&gt; addition</p>	20/08/'01
Rev.1.0	<p>Page 4 Figure 1.3 XOUT --&gt; revised</p> <p>Page 18 Figure 4.1 and 4.2 --&gt; revised</p> <p>Page 20 Figure 4.3, Timer B3 mode register value 00?x0000 --&gt; 00??0000 Three-phase output buffer register 0, 1 value 00<sub>16</sub> --&gt; 3F<sub>16</sub></p> <p>Page 21 Figure 4.4, Timer B0 mode register value 00?x0000 --&gt; 00??0000 UART transmit/receive control register 2 value x0000000 --&gt; x0xx0000 Flash memory control register1 value ?????0?? --&gt; ?????0???</p> <p>Page 22 (address006A<sub>16</sub>) DM1IC --&gt; DM2IC revised</p> <p>Page 24 (address0370<sub>16</sub>) UCON2 --&gt; UCON revised</p> <p>Page 26 "Carry out a software reset after oscillation of main clock is fully stable"--&gt; added, (1), (2) --&gt; revised</p> <p>Page 27 Figure 6.1 10:Inhibited --&gt; 10:Must not be set</p>	02/08/'05

Version	Contents for change	Revision date
Rev.1.0	<p>Page 46, 79 Note 9 'Do not set CM04 and CM07 simultaneously.' --&gt; deleted  Note 9 'In addition,do not rewrite CM04 and CM05 simultaneously' --&gt; added</p> <p>Page 48 'The priority level of the interrupt which is not used to cancel stop mode,must have been changed to 0.', 'If only a hardware reset or an NMI interrupt is used to cancel stop mode,change the priority level of all interrupt to 0,then shift to stop mode.' --&gt; added</p> <p>Page 50 'The priority level of the interrupt which is not used to cancel wait mode,must have been changed to 0.', 'If only a hardware reset or an NMI interrupt is used to cancel stop mode,change the priority level of all interrupt to 0,then shift to wait mode.' --&gt; added</p> <p>Page 54 Figure 8.7 The arrow of CM07="1" and CM05="1" deleted.</p> <p>Page 70 Interrupt request accepted. To CLK --&gt; Interrupt request priority detection results output(to clock generation circuit)</p> <p>Page 74 'Signal of "L" level width...for NMI pin.' --&gt; 'Signals input to the NMI ...from the operation clock of CPU.'</p> <p>Page 75 (5) Rewrite the interrupt control register  'When attempting to clear the interrupt...Instructions :MOV' --&gt; added</p> <p>Page 78 'Therefore,we recom-mend using the watchdog timer to improve reliability of a system.' --&gt; added</p> <p>Page 90 '2 instructions' --&gt; '26 cycles', Program example revised, (4) --&gt; added</p> <p>Page 91 (5) Recommended procedure for starting DMA transfer--&gt; added</p> <p>Page 92 'Count source for each timer becomes an operation clock for timer operation as counting and reloading,etc.' --&gt; added,  Figure 12.1 One-shot mode --&gt; One-shot timer mode</p> <p>Page 95 Figure 13.1 Up/Down flag Note2 --&gt; added  Figure13.3 Timer Ai register (FFFF --&gt; FFFE revised)</p> <p>Page 99 'TAiIN pin function', 'TAiOUT pin function' specification revised  Two-phase pulse input (Set the corresponding function select registers A to I/O port,and port direction register to "0")</p> <p>Page 103, 104 Selected by event/trigger select register --&gt; Selected by event/trigger select bit</p> <p>Page 106 Figure 14.2, 11: Inhibited --&gt; 11: Must not be set</p> <p>Page 109 TBiIN Pin function  Programable I/O port or --&gt; added, Figure 14.5 11: Inhibited --&gt; 11: Must not be set</p> <p>Page 110 Interrupt request generation timing, Figure 14.6  'When an overflow occurs.(Simultaneously,the timer Bi overflow flag... the timer Bi mode register.)' --&gt; 'Timer overflow.When an overflow occurs... the timer start flag is set to "1".</p> <p>Page 112 Figure 15.1 Note 5 Rewrite the INV00 to INV02 and INV06 bits when the timers A1,A2,A4 and B stop. --&gt;added</p> <p>Page 113 Figure 15.2 Note 1--&gt; added  Note 4 --&gt; added  Three-phase output buffer register 0, 1 0016 --&gt; 3F16</p> <p>Page 114 Figure 15.3, Timer Ai register value Note 2 --&gt; added</p> <p>Page 130 to 132 Inhibited --&gt; Must not be set</p> <p>Page 134 Figure 16.10 UART transmit/receive control register 2  x0000000 --&gt; x0xx0000</p>	02/08/05

Version	Contents for change	Revision date
Rev.1.0	<p>Page 136 Figure 16.12 Special mode register 3 'SDAi digital delay time set bit' --&gt; revised                      001: 1 to 2 cycles of 1/f(XIN)                      010: 1 to 2 cycles of 1/f(XIN)                      011: 1 to 2 cycles of 1/f(XIN)                      100: 1 to 2 cycles of 1/f(XIN)                      101: 1 to 2 cycles of 1/f(XIN)                      110: 1 to 2 cycles of 1/f(XIN)                      111: 1 to 2 cycles of 1/f(XIN)</p> <p>Page 150 Figure18.5 /P --&gt; P revised</p> <p>Page 152 Typical transmit/receive timing in UART mode (compliant with the SIM interface) Diagram revised</p> <p>Page 153 (a) Function for outputting a parity error signal                      With the error signal output enable bit ... TxDi pin when a parity error is detected.                      --&gt; During reception,with the error signal output enable bit ... TxDi pin when a parity error is detected.                      In step with this function, ... of a parity error signal. --&gt; deleted                      Therefore parity error signals ... interrupt program. --&gt; added                      And during transmission, ... of the transfer clock. --&gt; added</p> <p>Page 160 the baud rate generator --&gt; the UARTi bit rate generator                      baud rate generator stops counting. --&gt;the count stops</p> <p>Page 160 UART2 Special Mode Register 2 (Address 033616) --&gt; UARTi Special Mode Register 2(i=2 to 4) (Address 033616,032616,02F616), SCL2 --&gt; SCLi, SDA2 --&gt; SDAi, UART2 --&gt; UARTi revised</p> <p>Page 161 UART2 Special Mode Register 3(Address 033516) --&gt; UARTi Special Mode Register 3(i=2 to 4)(Address 033516,032516,02F516)                      SDA2 --&gt; SDSi revised</p> <p>Page 163 Figure 20.6 Special mode register 3 'SDAi digital delay time set bit' --&gt; revised                      001: 1 to 2 cycles of 1/f(XIN)                      010: 1 to 2 cycles of 1/f(XIN)                      011: 1 to 2 cycles of 1/f(XIN)                      100: 1 to 2 cycles of 1/f(XIN)                      101: 1 to 2 cycles of 1/f(XIN)                      110: 1 to 2 cycles of 1/f(XIN)                      111: 1 to 2 cycles of 1/f(XIN)</p> <p>Page 176 Set the function select register A... --&gt; Set the function select register A3...</p> <p>Page 193 Figure 26.2, P74, P75, P80 move to (inside dotted-line included)</p> <p>Page 208 Timer --&gt; added</p> <p>Page 209 (5) If an external trigger input is used to start counting, the next external trigger input must be avoided within 300ns before the timer A reaches "0000h". --&gt; added</p> <p>Page 210 Timer B (pulse period/pulse width measurement mode)                      (4) --&gt; added</p>	02/08/05

Version	Contents for change	Revision date
Rev.1.0	<p>Page 210 Stop Mode and Wait Mode (4),(5) --&gt; added</p> <p>(4) Follow the procedure below to enter stop mode.</p> <ul style="list-style-type: none"> <li>• Initial Setting Set each interrupt priority level after setting the minimum interrupt priority level required to exit stop mode and wait mode, controlled by the RLVL2 to RLVL0 bits in the RLVL register, to "7".</li> <li>• Before Execution of WAIT Instruction <ul style="list-style-type: none"> <li>[1] Set the interrupt priority level of the interrupt being used to exit stop mode</li> <li>[2] Set the interrupt priority levels of the interrupts not being used to exit stop mode</li> <li>[3] Set the IPL in the FLG register. Then set the minimum interrupt priority level required to exit stop mode and wait mode to the same level as the IPL. (Interrupt priority level of the interrupt used to exit stop mode &gt; minimum interrupt priority level to exit stop mode ≥ interrupt priority level of the interrupts not used to exit stop mode)</li> <li>[4] Set the I flag to "1"</li> <li>[5] Set the CM10 bit in the CM1 register to "1" (all clocks stop) after setting the PRC0 bit in the PRCR register to "1" (write enabled) <ul style="list-style-type: none"> <li>• After Exiting Stop Mode Set the interrupt priority level required to exit stop mode to "7" immediately after exiting stop mode.</li> </ul> </li> </ul> </li> </ul> <p>(5) When microcomputer enters stop mode again after exiting from stop mode using the NMI interrupt, use the following procedure to set the CM10 bit to "1".</p> <ul style="list-style-type: none"> <li>[1] Exit stop mode using the /NMI interrupt</li> <li>[2] Generate a dummy interrupt</li> <li>[3] Set the CM10 bit to "1"</li> </ul> <p>example:</p> <pre> INT    #63    ; Dummy interrupt BSET   CM1    ; All clocks stopped (in stop mode) ; /*for dummy interrupt*/ DUMMY: REIT </pre>	02/08/05

Version	Contents for change	Revision date
Rev.1.0	<p>Page 210 (2) --&gt; (2)Wait and (3)Stop divided</p> <p>(2) When entering wait mode, the instruction queue reads ahead to instructions following the WAIT instruction, and the program stops. The next instruction may be executed before entering wait mode, depending on the combination of instructions and their execution timing. Therefore, write at least 4 NOP instructions following the WAIT instruction.</p> <p>(3) When entering stop mode, the instruction queue reads ahead to instructions to set the CM10 bit to "1", and the program stops. The next instruction may be executed, before entering stop mode or executing the interrupt routine to exit stop mode, depending on the combination of instructions and their execution timing. Therefore, write a jmp.b instruction and at least 4 NOP instructions following the instruction to enter stop mode as shown below.</p> <pre>                 bset 0,prcr          ; Removing protection                 bset 0,cm1          ; Stopping all clocks(Entering stop mode)                 jmp.b LABEL_001     ; Executing jmp.b instruction(Jump to the                                    ; next instruction soon LABEL_001:      ; with adding no instruction between jmp.b                                    ; and LABEL.)                 nop                 ; nop(1)                  nop                 ; nop(2)                 nop                 ; nop(3)                 nop                 ; nop(4)                 mov.b #0,prcr       ; Setting protection                 .                 .                 . </pre> <p>Page 211 (4) (6) --&gt; Description revised  Page 211 A/D Converter (6) --&gt; added  Page 214 (2)The NMI interrupt  'Signal of "L" level width...for NMI pin.' --&gt; 'Signals input to the NMI ...from the operation clock of CPU.'</p> <p>Page 217 (5) Rewrite the interrupt control register  'When attempting to clear the interrupt...Instructions :MOV' --&gt; added</p> <p>Page 217 DMAC  '2 instructions' --&gt; '26 cycles', Program example revised</p> <p>Page 218 (4), (5) --&gt; added</p> <p>Page 219 HOLD signal  When P40 to P47 and P50 to P52 are set to... will not become high-impedance ports. --&gt; added  Memory expansion mode added</p> <p>Page 221 Notes on the microprocessor mode and transition after shifting from the microprocessor mode to the memory expansion mode / sigle-chip mode --&gt; deleted</p>	02/08/05

Version	Contents for change	Revision date
Rev.1.0	<p>Page 221 Notes on CNVSS pin reset at "H" level --&gt; added</p> <p>Page 222 Microprocessor mode or Memory expansion mode --&gt; added</p> <p>Page 222 Microprocessor(Usage Precaution) --&gt; added                      If the software reset is executed when the CNVss pin is connected to Vcc to start up in microprocessor mode, write at least three NOP instructions following the writing instruction to the PM0 Register.                      example:                      mov.b #02H,PRCR                      bset 3,PM0 ; or "mov.b #8BH,PM0" (instruction to execute software reset)                      nop ; write at least three NOP instructions                      nop                      nop                      nop</p> <p>Page 227 Table 28.3 , Table 28.4 --&gt; added</p> <p>Page 249 (Note 4) contained in the title of Table 28.23 --&gt; deleted</p> <p>Page 274 Read data from an even address in the user ROM area when reading the status register. --&gt; added</p> <p>Page 278 main clock frequency --&gt; BCLK</p> <p>Page 279 Table 30.1 Read status register command's second bus cycle 'X' --&gt; 'X'(Note 6)                      Note 6 (that is an even address) --&gt; added</p> <p>Page 286 Figure 30.8 (When reading the status register,set an even number address in the user ROM area). --&gt; added</p> <p>Page 287 Figure 30.9 ROM code protect level 2 set bit --&gt; deleted, Notes 3 to 5 --&gt; added</p> <p>Page 289 In this mode, the M16C/80 (flash memory version) operates in a manner similar to the flash memory M5M29FB/T800 from Mitsubishi. Since there are some differences with regard to the functions not available with the microcomputer and matters related to memory capacity, the M16C/80 cannot be programmed by a programmer for the flash memory. --&gt; deleted</p> <p>Page 7,271,323 Mitsubishi--&gt;Renesas revised</p> <p>Page 289 Mitsubishi --&gt;deleted</p> <p>Page 294 from Mitsubisi --&gt; from the factory revised</p> <p>Page 323 Mitsubisi --&gt; the following revised</p> <p>All Pages Words standardized: A/D converter, D/A converter and XY converter</p> <p>All Pages Figure number and Table number change</p>	02/08/05

---

**RENESAS 16-BIT SINGLE-CHIP MICROCOMPUTER  
HARDWARE MANUAL  
M16C/80 Group**

**Publication Data : Rev.B    Oct. 19, 1998  
Rev.1.00    Aug. 02, 2005**

**Published by : Sales Strategic Planning Div.  
Renesas Technology Corp.**

---

# M16C/80 Group Hardware Manual



Renesas Electronics Corporation

1753, Shimonumabe, Nakahara-ku, Kawasaki-shi, Kanagawa 211-8668 Japan

REJ09B0187-0100