To our customers,

Old Company Name in Catalogs and Other Documents

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

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

April 1st, 2010
Renesas Electronics Corporation

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

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

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.

User's Manual

32

The revision list can be viewed directly by clicking the title page.

The revision list summarizes the locations of revisions and additions. Details should always be checked by referring to the relevant text.

SH7040, SH7041, SH7042, SH7043, SH7044, SH7045 Group

Hardware Manual

Renesas 32-Bit RISC Microcomputer

SuperH RISC engine Family/SH7040 Series

(CPU Core SH-2)

Renesas Electronics
www.renesas.com

Rev.6.00 2003.5

Renesas 32-Bit RISC Microcomputer
SuperH RISC engine Family/
SH7040 Series (CPU Core SH-2)

SH7040, SH7041, SH7042, SH7043, SH7044, SH7045 Group

Hardware Manual

REJ09B0044-0600O

Cautions

Keep safety first in your circuit designs!

1. Renesas Technology Corporation 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 nonflammable 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 Corporation 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 Corporation or a third party.
2. Renesas Technology Corporation 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 Corporation without notice due to product improvements or other reasons. It is therefore recommended that customers contact Renesas Technology Corporation or an authorized Renesas Technology Corporation 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 Corporation 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 Corporation by various means, including the Renesas Technology Corporation 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 Corporation assumes no responsibility for any damage, liability or other loss resulting from the information contained herein.
5. Renesas Technology Corporation 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 Corporation or an authorized Renesas Technology Corporation 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 Corporation 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 Corporation for further details on these materials or the products contained therein.

Preface

The SH7040 Series (SH7040, SH7041, SH7042, SH7043, SH7044, SH7045) single-chip RISC (Reduced Instruction Set Computer) microprocessors integrate a Renesas Technology-original RISC CPU core with peripheral functions required for system configuration.

The CPU has a RISC-type instruction set. Most instructions can be executed in one clock cycle, which greatly improves instruction execution speed. In addition, the 32-bit internal-bus architecture enhances data processing power. With this CPU, it has become possible to assemble low cost, high performance/high-functioning systems, even for applications that were previously impossible with microprocessors, such as real-time control, which demands high speeds. In particular, the SH7040 series has a 1-kbyte on-chip cache, which allows an improvement in CPU performance during external memory access.

In addition, the SH7040 series includes on-chip peripheral functions necessary for system configuration, such as large-capacity ROM and RAM, timers, a serial communication interface (SCI), an A/D converter, an interrupt controller, and I/O ports. Memory or peripheral LSIs can be connected efficiently with an external memory access support function. This greatly reduces system cost.

There are versions of on-chip ROM: mask ROM, PROM, and flash memory. The flash memory can be programmed with a programmer that supports SH7040 series programming, and can also be programmed and erased by software.

This hardware manual describes the SH7040 series hardware. Refer to the programming manual for a detailed description of the instruction set.

Related Manual

SH7040 series instructions

SH-1/SH-2/SH-DSP Programming Manual

Please consult your Renesas Technology sales representative for details for development environment system.

List of Items Revised or Added for This Version

1.1.1 SH7040 Series Features

Notes on the SH7040 Series Specifications — 7, 9

Type	Abbreviation	Mask Version	On-chip ROM	External Bus Width	A/D Accuracy (5Vversion)	Package	Operating Temp	Frequency	Voltage	Type Name	Notes on the SH7040 Series Specifications: ROM	Notes on the SH7040 Series Specifications: Electrical Characteristics
ZTAT	SH7042		128 kB	16 bits	±15LSB (High-Speed)	QFP2020-112	–20°C to 75°C	28 MHz	5 V	HD6477042F28	See "128 kB PROM"	See "Electrical Characteristics"
								16 MHz	3.3 V	HD6477042VF16		
	SH7042A	A mask	128 kB	16 bits	±4LSB (Mid-Speed)	QFP2020-112	–20°C to 75°C	28 MHz	5 V	HD6477042AF28	See "128 kB PROM"	See "Electrical Characteristics"
								16 MHz	3.3 V	HD6477042AVF16		
						TQFP1414-120		16 MHz	3.3 V	HD6477042AVX16		
						QFP2020-112Cu*						
								28 MHz	5 V	HD6477042ACF28		
								16 MHz	3.3 V	HD6477042AVCF16		
	SH7043		128 kB	32 bits	±15LSB (High-Speed)	QFP2020-144	–20°C to 75°C	28 MHz	5 V	HD6477043F28	See "128 kB PROM"	See "Electrical Characteristics"
								16 MHz	3.3 V	HD6477043VF16		
	SH7043A	A mask	128 kB	32 bits	±4LSB (Mid-Speed)	QFP2020-144	–20°C to 75°C	28 MHz	5 V	HD6477043AF28	See "128 kB PROM"	See "Electrical Characteristics"
								16 MHz	3.3 V	HD6477043AVF16		
						QFP2020-144Cu*						
								28 MHz	5 V	HD6477043ACF28		
								16 MHz	3.3 V	HD6477043AVCF16		
FLASH	SH7044F	A mask	256 kB	16 bits	±4LSB (Mid-Speed)	QFP2020-112	–20°C to 75°C	28 MHz	5 V	HD64F7044F28	See "256 kB Flash Memory"	See "Electrical Characteristics"
	SH7045F	A mask	256 kB	32 bits	±4LSB (Mid-Speed)	QFP2020-144	–20°C to 75°C	28 MHz	5 V	HD64F7045F28	See "256 kB Flash Memory"	See "Electrical Characteristics"
MASK	SH7040A	A mask	64 kB	16 bits	±4LSB (Mid-Speed)	QFP2020-112	–20°C to 75°C	28 MHz	5 V	HD6437040AF28	See "64 kB Mask ROM"	See "Electrical Characteristics"
								16 MHz	3.3 V	HD6437040AVF16		
						TQFP1414-120		16 MHz	3.3 V	HD6437040AVX16		
						QFP2020-112Cu*						
								28 MHz	5 V	HD6437040ACF28		
								16 MHz	3.3 V	HD6437040AVCF16		
	SH7041A	A mask	64 kB	32 bits	±4LSB (Mid-Speed)	QFP2020-144	–20°C to 75°C	28 MHz	5 V	HD6437041AF28	See "64 kB Mask ROM"	See "Electrical Characteristics"
								16 MHz	3.3 V	HD6437041AVF16		
						QFP2020-144Cu*						
								28 MHz	5 V	HD6437041ACF28		
								16 MHz	3.3 V	HD6437041AVCF16		
	SH7042		128 kB	16 bits	±15LSB (High-Speed)	QFP2020-112	–20°C to 75°C	28 MHz	5 V	HD6437042F28	See "128 kB Mask ROM"	See "Electrical Characteristics"
								16 MHz	3.3 V	HD6437042VF16		
	SH7042A	A mask	128 kB	16 bits	±4LSB (Mid-Speed)	QFP2020-112	–20°C to 75°C	28 MHz	5 V	HD6437042AF28	See "128 kB Mask ROM"	See "Electrical Characteristics"
								16 MHz	3.3 V	HD6437042AVF16		
						TQFP1414-120		16 MHz	3.3 V	HD6437042AVX16		
						QFP2020-112Cu*						
								28 MHz	5 V	HD6437042ACF28		
								16 MHz	3.3 V	HD6437042AVCF16		
	SH7043		128 kB	32 bits	±15LSB (High-Speed)	QFP2020-144	–20°C to 75°C	28 MHz	5 V	HD6437043F28	See "128 kB Mask ROM"	See "Electrical Characteristics"
								16 MHz	3.3 V	HD6437043VF16		
	SH7043A	A mask	128 kB	32 bits	±4LSB (Mid-Speed)	QFP2020-144	–20°C to 75°C	28 MHz	5 V	HD6437043AF28	See "128 kB Mask ROM"	See "Electrical Characteristics"
								16 MHz	3.3 V	HD6437043AVF16		
						QFP2020-144Cu*						
								28 MHz	5 V	HD6437043ACF28		
								16 MHz	3.3 V	HD6437043AVCF16		
	SH7044	A mask	256 kB	16 bits	±4LSB (Mid-Speed)	QFP2020-112	–20°C to 75°C	28 MHz	5 V	HD6437044F28	See "256 kB Mask ROM"	See "Electrical Characteristics"
	SH7045	A mask	256 kB	32bits	±4LSB (Mid-Speed)	QFP2020-144	–20°C to 75°C	28 MHz	5 V	HD6437045F28	See "256 kB Mask ROM"	See "Electrical Characteristics"
ROM less	SH7040A	A mask		16 bits	±4LSB (Mid-Speed)	QFP2020-112	–20°C to 75°C	28 MHz	5 V	HD6417040AF28		See "Electrical Characteristics"
								16 MHz	3.3 V	HD6417040AVF16		
						TQFP1414-120		16 MHz	3.3 V	HD6417040AVX16		
						QFP2020-112Cu*		28 MHz	5 V	HD6417040ACF28		
								16 MHz	3.3 V	HD6417040AVCF16		
	SH7041A	A mask		32 bits	±4LSB (Mid-Speed)	QFP2020-144	–20°C to 75°C	28 MHz	5 V	HD6417041AF28		See "Electrical Characteristics"
								16 MHz	3.3 V	HD6417041AVF16		
						QFP2020-144Cu*						
								28 MHz	5 V	HD6417041ACF28		
								16 MHz	3.3 V	HD6417041AVCF16		

Note: Package with Copper used as the lead material.

1.4 The F-ZTAT Version Onboard Programming

Figure 1.6 Condition Transfer for Flash Memory — 42

Note amended

Notes: For transferring between user mode and user program mode, proceed while CPU is not programming or erasing the flash memory.
* RAM emulation permitted

2.4 Instruction Set by Classification

Table 2.16 Branch Instructions — 70

Table amended

BF/S	label	10001111dddddddd	Delayed branch, if T = 0, disp × 2 + PC → PC; if T = 1, nop	2/1*	—

4.2.3 Notes on Board Design

—

Deleted

4.5 Usage Notes

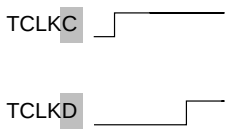
83 to 85

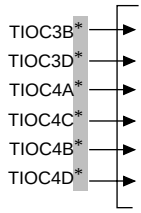
Newly added

11.1.4 Register Configuration

Table 11.2 DMAC Registers — 218

Note *5 deleted

Section	Page	Description
11.2.3 DMA Transfer Count Registers 0–3 (DMATCR0–DMATCR3)	220	Description amended The data for the upper 8 bits of a DMATCR is 0 when read.
11.2.4 DMA Channel Control Registers 0–3 (CHCR0–CHCR3)	221	Description amended • Bits 31–21—Reserved bits: Data are 0 when read. The write value always be 0.
	224	Description amended • Bit 7—Reserved bits: Data is 0 when read. The write value always be 0.
11.2.5 DMAC Operation Register (DMAOR)	226	Description amended • Bits 15–10—Reserved bits: Data are 0 when read. The write value always be 0.
	227	Description amended • Bits 7–3—Reserved bits: Data are 0 when read. The write value always be 0.
11.3.3 Channel Priority Figure 11.3 Round Robin Mode	233	Figure amended Channel 0 is given the lowest priority.
12.4.5 Cascade Connection Mode Figure 12.23 Cascade Connection Operation Example (Phase Counting Mode)	337	Figure amended 
12.4.9 Complementary PWM Mode Figure 12.55 Example of Output Phase Switching by External Input (1)	373	Figure amended **When BDC = 1, N = 0, P = 0, FB = 0, output active level = high**

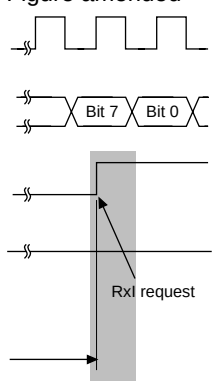
Section	Page	Description
12.9.2 Block Diagram Figure 12.125 POE Block Diagram	444	Note added  Note: * Includes multiplexed pins.
12.11.5 Usage Notes	453	Section added
14.2.8 Bit Rate Register (BRR) Table 14.3 Bit Rates and BRR Settings in Asynchronous Mode (cont)	491	Table amended

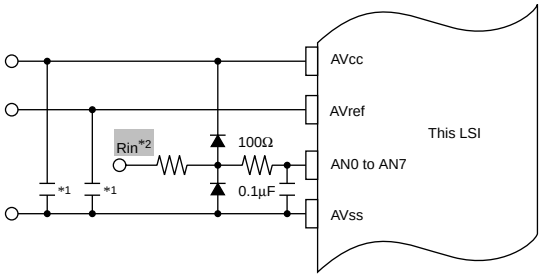
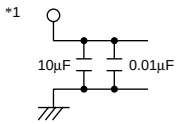
Bit Rate (Bits/s)	27.0336		
	n	N	Error (%)
110	3	119	0.00
150	3	87	0.00
300	2	175	0.00
600	1	87	0.00
1200	1	175	0.00
2400	1	87	0.00
4800	0	175	0.00
9600	0	87	0.00
14400	0	58	–0.56
19200	0	43	0.00
28800	0	28	1.15
31250	0	26	0.12
38400	0	21	0.00

Table 14.4 Bit Rates and BRR Settings in Clocked Synchronous Mode (cont)	495	Table amended

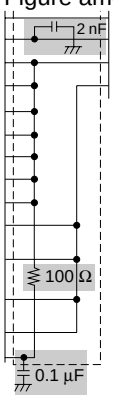
3.5M	—	—	—	—	0	1
5M	0	0*	—	—	—	—
7M			—	—	0	0*

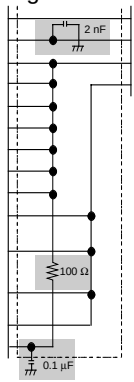
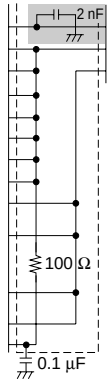
14.3.4 Clock Synchronous Operation Figure 14.22 Example of SCI Receive Operation	529	Figure amended



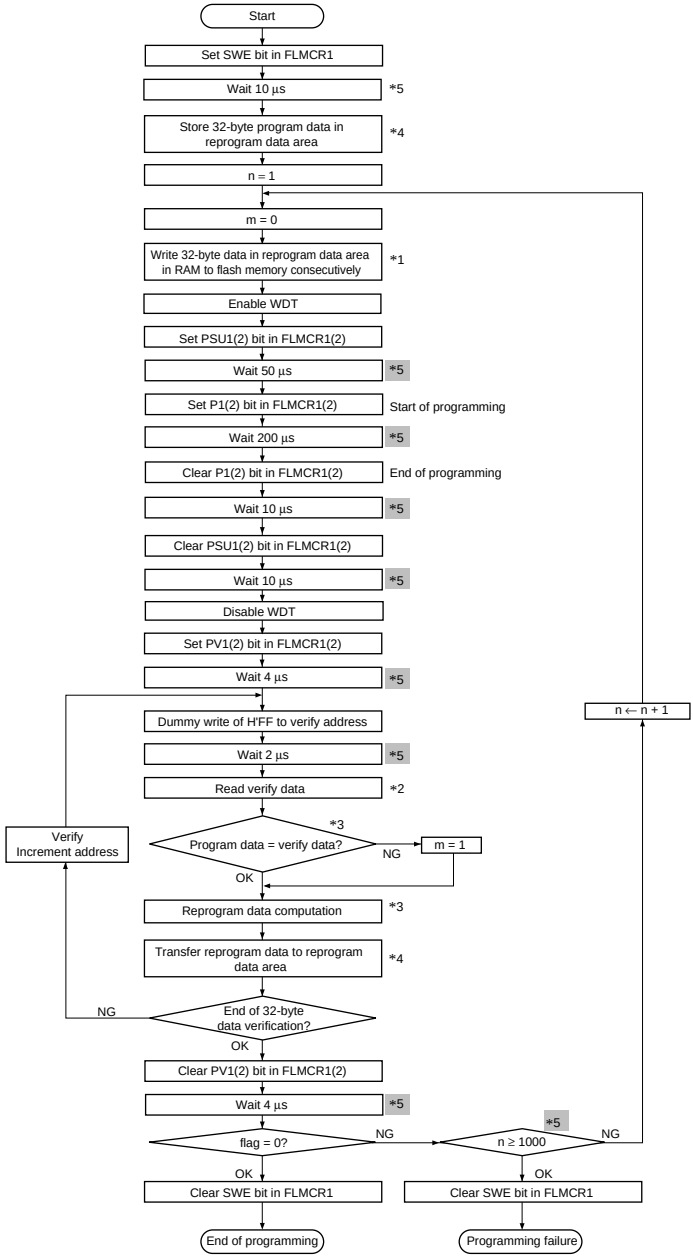
Section	Page	Description
15.4.9 A/D Conversion Time Table 15.8 Operating Frequency and CKS Bit Settings	562	33 MHz deleted
15.6 Notes on Use Figure 15.14 Example of a Protection Circuit for the Analog Input Pins	564	Figure amended  Notes: Numbers are only to be noted as reference value  *2 Rin: Input impedance
16.7.2 Handling of Analog Input Pins Figure 16.8 Example of Analog Input Pin Protection Circuit	585	Note amended Notes: Numbers are only to be noted as reference value
19.2 Port A Table 19.2 Port A, FP-144 Version	649	Table amended

PA16 (I/O)/$\overline{AH}$ (output)	PA16 (I/O)/$\overline{AH}$ (output)	PA16 (I/O)
PA15 (I/O)/CK (output)	PA15 (I/O)/CK (output)	PA15 (I/O)/CK (output)

21.2.2 Socket Adapter Pin Correspondence and Memory Map Figure 21.2 SH7042 Pin and HN27C101 Pin Correspondence (112-Pin Version)	671	Figure amended 

Section	Page	Description
21.2.2 Socket Adapter Pin Correspondence and Memory Map Figure 21.3 SH7042 Pin and HN27C101 Pin Correspondence (120-Pin Version)	672	Figure amended 
Figure 21.4 SH7043 Pin and HN27C101 Pin Correspondence (144-Pin Version)	673	Figure amended 
22.2.2 Mode Transition Diagram Figure 22.2 Flash Memory Mode Transitions	683	Note amended Execute transition between the user mode and user program mode while the CPU is not programming or erasing the flash memory

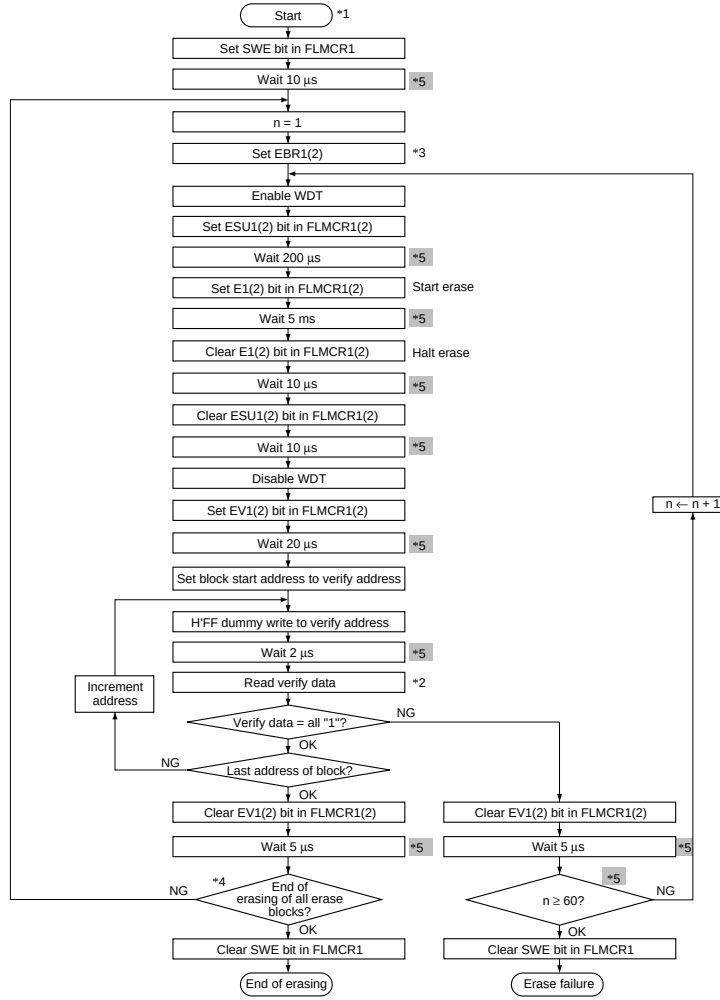
Section	Page	Description
22.7.2 Program-Verify Mode Figure 22.13 Program/Program Verify Flow	706	Figure amended



Note *5 added.

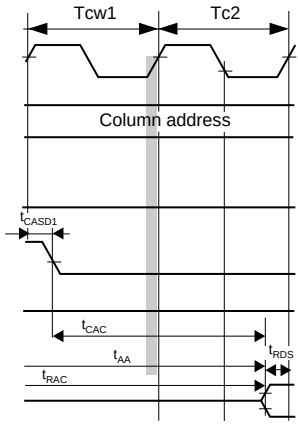
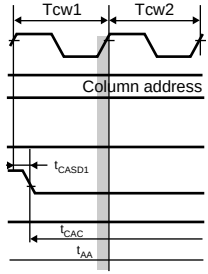
*5 Make sure to set the wait times and repetitions as specified. Programming may not complete correctly if values other than the specified ones are used.

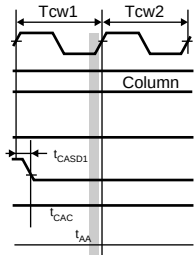
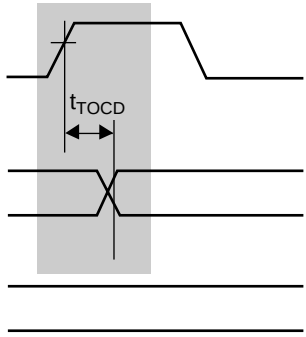
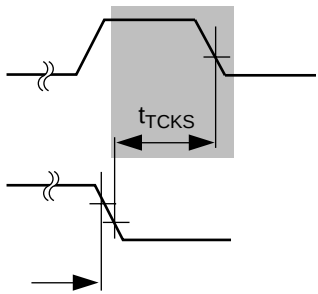
Section	Page	Description
22.7.4 Erase-Verify Mode Figure 22.14 Erase/Erase-Verify Flowchart	713	Figure amended

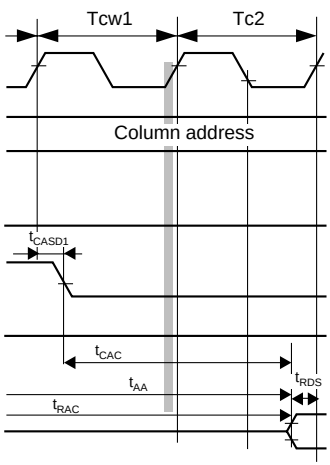


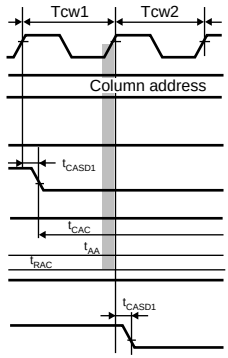
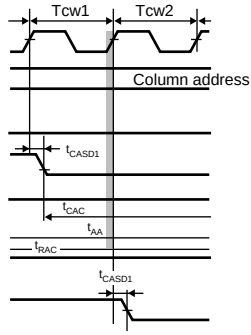
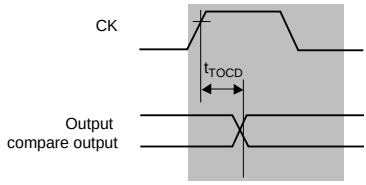
Notes: *1 Preprogramming (setting erase block data to all "0") is not necessary.
*2 Verify data is read in 32-bit (longword) units.
*3 Set only one bit in EBR1(2). More than one bit cannot be set.
*4 Erasing is performed in block units. To erase a number of blocks, each block must be erased in turn.
*5 Make sure to set the wait times and repetitions as specified. Erasing may not complete correctly if values other than the specified ones are used.

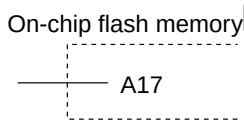
24.4.2 Canceling the Standby Mode	747	Cancellation by a Manual Reset deleted
25. Electrical Characteristics (5V, 33.3 MHz Version)	—	Deleted

Section	Page	Description
25.2 DC Characteristics Table 25.2 DC Characteristics	751	Note amended *2 5 mA in the A mask version, except for F-ZTAT products.
25.3.2 Control Signal Timing Table 25.5 Control Signal Timing	754	Note amended Note: * The $\overline{RES}$, $\overline{MRES}$, NMI, $\overline{BREQ}$, and $\overline{IRQ7}$–$\overline{IRQ0}$ signals are asynchronous inputs, but when thesetup times shown here are provided, the signals are considered to have produced changes at clock rise (for $\overline{RES}$, $\overline{MRES}$, $\overline{BREQ}$) or clock fall (for NMI and $\overline{IRQ7}$–$\overline{IRQ0}$). If the setup times are not provided, recognition is delayed until the next clock rise or fall.
25.3.3 Bus Timing Figure 25.12 DRAM Cycle (Normal Mode, 1 Wait, TPC=0, RCD=0)	763	Figure amended 
Figure 25.13 DRAM Cycle (Normal Mode, 2 Waits, TPC=1, RCD=1)	764	Figure amended 

Section	Page	Description
25.3.3 Bus Timing Figure 25.14 DRAM Cycle (Normal Mode, 3 Waits, TPC=1, RCD=1)	764	Figure amended 
25.3.5 Multifunction Timer Pulse Unit Timing Figure 25.23 MTU I/O Timing	770	Figure amended 
Figure 25.24 MTU Clock Input Timing	770	Figure amended 
25.3.11 Measuring Conditions for AC Characteristics Figure 25.33 Output Load Circuit	778	Title amended Output Load Circuit

Section	Page	Description
25.4 A/D Converter Characteristics Table 25.16 A/D Converter Timing (A mask)	779	Table amended Non-linearity error* Offset error* Full scale error* Quantize error*
26.2 DC Characteristics Table 26.2 DC Characteristics	782	Table amended Schmitt trigger input voltage — PA2, PA5, PA6–PA9, PE0–PE15 — $V_T^+ - V_T^-$ — $V_{CC}\times 0.07$ — — V — $VT^+ \geq V_{CC}\times 0.9V$ (min); $VT^- \leq V_{CC}\times 0.2V$ (max)
	783	Table amended Analog supply current — AI_{CC} — — 4 8 mA f = 16.7MHz AI_{ref} — — 0.5 1*3 mA QFP144 version only *3 2 mA in the A mask version of MASK products.
26.3.2 Control Signal Timing Table 26.5 Control Signal Timing	786	Note amended Notes: *1 SH7042/43 ZTAT (excluding A mask) are 3.2V. *2 The $\overline{RES}$, $\overline{MRES}$, NMI, $\overline{BREQ}$, and $\overline{IRQ7}$–$\overline{IRQ0}$ signals are asynchronous inputs, but when the setup times shown here are provided, the signals are considered to have produced changes at clock rise (for $\overline{RES}$, $\overline{MRES}$, $\overline{BREQ}$) or clock fall (for NMI and $\overline{IRQ7}$–$\overline{IRQ0}$). If the setup times are not provided, recognition is delayed until the next clock rise or fall.
26.3.3 Bus Timing Figure 26.12 DRAM Cycle (Normal Mode, 1 Wait, TPC = 0, RCD = 0)	795	Figure amended 

Section	Page	Description
26.3.3 Bus Timing Figure 26.13 DRAM Cycle (Normal Mode, 2 Waits, TPC = 1, RCD = 1)	796	Figure amended 
Figure 26.14 DRAM Cycle (Normal Mode, 3 Waits, TPC = 1, RCD = 1)	796	Figure amended 
26.3.5 Multifunction Timer Pulse Unit Timing Figure 26.23 MTU I/O Timing	802	Figure amended 
26.3.11 Measurement Conditions for AC Characteristics Figure 26.33 Output Load Circuit	810	Title amended Output Load Circuit

Section	Page	Description
Appendix B Block Diagrams Figure B.19 PB4/IRQ2/POE2/CASH, PB3/IRQ1/POE1/CASL Block Diagram (F-ZTAT Version)	844	Note added  Note: * Only when n = 4.

Appendix C Pin States

Table C.1 Pin Modes During Reset, Power-Down, and Bus Right Release Modes (144 Pin)

865 Table amended

Pin Function		Pin modes					
		Reset		Power-Down		Bus Right Release	Standby in Bus Right Release
Class	Pin Name	Power-On	Manual	Standby	Sleep		
Clock	CK	O	O	H*1	O	O	O
System control	$\overline{RES}$	I	I	I	I	I	I
	$\overline{MRES}$	Z*4	I	Z	I	I	Z
	$\overline{WDTOVF}$	O*3	O*3	O	O	O	O
	$\overline{BREQ}$	Z*4	I	Z	I	I	I
	$\overline{BACK}$	Z*4	O	Z	O	L	L
Interrupt	NMI	I	I	I	I	I	I
	$\overline{IRQ0}$–$\overline{IRQ7}$	Z*4	I	Z	I	I	Z
	$\overline{IRQOUT}$ (PD30)	Z*4	O	H*1	H	O	H*1
	$\overline{IRQOUT}$ (PE15)	Z*4	O	Z	H	O	Z
Address bus	A0–A21	O*2	O	Z	O	Z	Z
Data bus	D0–D31	Z*4	I/O	Z	I/O	Z	Z
Bus control	$\overline{WAIT}$	Z*4	I	Z	I	Z	Z
	RD/$\overline{WR}$, $\overline{RAS}$	Z*4	O	O	O	Z	Z
	$\overline{CASH}$, $\overline{CASL}$, $\overline{CASLH}$, $\overline{CASLL}$	Z*4	O	O	O	Z	Z
	$\overline{RD}$	H	O	Z	O	Z	Z
	$\overline{CS0}$, $\overline{CS1}$	H	O	Z	O	Z	Z
	$\overline{CS2}$, $\overline{CS3}$	Z*4	O	Z	O	Z	Z
	$\overline{WRHH}$, $\overline{WRHL}$, $\overline{WRH}$, $\overline{WRL}$	H	O	Z	O	Z	Z
	$\overline{AH}$	Z*4	O	Z	O	Z	Z
DMAC	DACK0, DACK1 (PD26, PD27)	Z*4	O	O*1	O	O	O*1
	DACK0, DACK1 (PE14, PE15)	Z*4	O	Z	O	O	Z
	DRAK0, DRAK1	Z*4	O	O*1	O	O	O*1
	$\overline{DREQ0}$, $\overline{DREQ1}$	Z*4	I	Z	I	I	Z

Section	Page	Description
Appendix C Pin States	866	Table amended

Table C.1 Pin Modes During Reset, Power-Down, and Bus Right Release Modes (144 Pin) (cont)

Pin Function		Pin modes					
		Reset		Power-Down		Bus Right Release	Standby in Bus Right Release
Class	Pin Name	Power-On	Manual	Standby	Sleep		
MTU	TIOC0A–TIOC0D, TIOC1A–TIOC1D, TIOC2A–TIOC2D, TIOC3A, TIOC3C	Z*4	I/O	K*1	I/O	I/O	K*1
	TIOC3B,TIOC3D, TIOC4A–TIOC4D	Z*4	I/O	Z	I/O	I/O	Z
	TCLKA–TCLKD	Z*4	I	Z	I	I	Z
Port control	$\overline{\text{POE0}}$–$\overline{\text{POE3}}$	Z*4	I	Z	I	I	Z
SCI	SCK0–SCK1	Z*4	I/O	Z	I/O	I/O	Z
	TXD0–TCD1	Z*4	O	O*1	O	O	O*1
	RXD0–RXD1	Z*4	I	Z	I	I	Z
A/D converter	$\overline{\text{ADTRG}}$	Z*4	I	Z	I	I	Z
	AN0–AN7	Z	I	Z	I	I	Z
I/O Port	PA0–PA23	Z*4	I/O	K*1	K	I/O	K*1
	PB0–PB9						
	PC0–PC15						
	PD0–PD31						
	PE0–PE8,PE10						
	PE9,PE11–PE15	Z*4	I/O	Z	K	I/O	Z
	PF0–PF17	Z	I	Z	I	I	Z

Notes: 1. There are instances where bus right release and transition to software standby mode occur simultaneously due to the timing between $\overline{\text{BREQ}}$ and internal operations. In such cases, standby mode results, but the standby state may be different.

The initial pin states depend on the mode. See section 18, Pin Function Controller (PFC), for details.

2. I: Input, O: Output, H: High-level output, L: Low-level output, Z: High impedance, K: Input pin with high impedance, output pin mode maintained.

*1 If the standby control register port high-impedance bits are set to 1, output pins become high impedance.

*2 A21–A18 will become input ports after power-on reset.

*3 Input in the SH7044/SH7045 F-ZTAT version.

*4 General use I/O ports PAn, PBn, PCn, PDn, and PEn, as well as pins multiplexed with them, are unstable during the $\overline{\text{RES}}$ setup time (t_{RESS}) immediately after the $\overline{\text{RES}}$ pin goes to low level.

Section	Page	Description
Appendix C Pin States	867	Table amended

Table C.2 Pin Modes During Reset, Power-Down, and Bus Right Release Modes (112 Pin, 120 Pin)

Pin Function		Pin modes					
		Reset		Power-Down		Bus Right	Standby in Bus
Class	Pin Name	Power-On	Manual	Standby	Sleep	Release	Right Release
Clock	CK	O	O	H*1	O	O	O
System control	$\overline{RES}$	I	I	I	I	I	I
	$\overline{MRES}$	Z*4	I	Z	I	I	Z
	$\overline{WDTOVF}$	O*3	O*3	O	O	O	O
	$\overline{BREQ}$	Z*4	I	Z	I	I	I
	$\overline{BACK}$	Z*4	O	Z	O	L	L
Interrupt	NMI	I	I	I	I	I	I
	$\overline{IRQ0}$–$\overline{IRQ7}$	Z*4	I	Z	I	I	Z
	$\overline{IRQOUT}$	Z*4	O	Z	H	O	Z
Address bus	A0–A21	O*2	O	Z	O	Z	Z
Data bus	D0–D31	Z*4	I/O	Z	I/O	Z	Z
Bus control	$\overline{WAIT}$	Z*4	I	Z	I	Z	Z
	RDWR, $\overline{RAS}$	Z*4	O	O	O	Z	Z
	$\overline{CASH}$, $\overline{CASL}$	Z*4	O	O	O	Z	Z
	$\overline{RD}$	H	O	Z	O	Z	Z
	$\overline{CS0}$, $\overline{CS1}$	H	O	Z	O	Z	Z
	$\overline{CS2}$, $\overline{CS3}$	Z*4	O	Z	O	Z	Z
	$\overline{WRH}$, $\overline{WRL}$	H	O	Z	O	Z	Z
	$\overline{AH}$	Z*4	O	Z	O	Z	Z
DMAC	DACK0–DACK1	Z*4	O	Z	O	O	Z
	DRAK0–DRAK1	Z*4	O	Z	O	O	Z
	$\overline{DREQ0}$–$\overline{DREQ1}$	Z*4	I	Z	I	I	Z
MTU	TIOC0A–TIOC0D, TIOC1A–TIOC1D, TIOC2A–TIOC2D, TIOC3A, TIOC3C	Z*4	I/O	K*1	I/O	I/O	K*1
	TIOC3B,TIOC3D, TIOC4A–TIOC4D	Z*4	I/O	Z	I/O	I/O	Z
	TCLKA–TCLKD	Z*4	I	Z	I	I	Z

Section	Page	Description
Appendix C Pin States	868	Table amended

Table C.2 Pin Modes During Reset, Power-Down, and Bus Right Release Modes (112 Pin, 120 Pin) (cont)

Pin Function		Pin modes					
		Reset		Power-Down		Bus Right	Standby in Bus
Class	Pin Name	Power-On	Manual	Standby	Sleep	Release	Right Release
Port control	$\overline{POE0}$–$\overline{POE3}$	Z*4	I	Z	I	I	Z
SCI	SCK0–SCK1	Z*4	I/O	Z	I/O	I/O	Z
	TXD0–TCD1	Z*4	O	O*1	O	O	O*1
	RXD0–RXD1	Z*4	I	Z	I	I	Z
A/D converter control	$\overline{ADTRG}$	Z*4	I	Z	I	I	Z
	AN0–AN7	Z	I	Z	I	I	Z
I/O Port	PA0–PA15	Z*4	I/O	K*1	K	I/O	K*1
	PB0–PB9						
	PC0–PC15						
	PD0–PD15						
	PE0–PE8–PE10						
	PE9,PE11–PE15	Z*4	I/O	Z	K	I/O	Z
	PF0–PF7	Z	I	Z	I	I	Z

Notes: 1. There are instances where bus right release and transition to software standby mode occur simultaneously due to the timing between $\overline{BREQ}$ and internal operations. In such cases, standby mode results, but the standby state may be different.

The initial pin states depend on the mode. See section 18, Pin Function Controller (PFC), for details.

2. I: Input, O: Output, H: High-level output, L: Low-level output, Z: High impedance, K: Input pin with high impedance, output pin mode maintained.

*1 If the standby control register port high-impedance bits are set to 1, output pins become high impedance.

*2 A21–A18 will become input ports after power-on reset.

*3 Input in the SH7044/SH7045 F-ZTAT version.

*4 General use I/O ports PAn, PBn, PCn, PDn, and PEn, as well as pins multiplexed with them, are unstable during the $\overline{RES}$ setup time (t_{RESS}) immediately after the $\overline{RES}$ pin goes to low level.

Section	Page	Description
Appendix E Product Code Lineup Table E.1 SH7040, SH7041, SH7042, SH7043, SH7044, and SH7045 Product Lineup	876, 877	Table amended

Product Type		Mask Version	Product Code	Mark Code	Package	Order Model No.*2
SH7040A	Mask ROM verion	A MASK	HD6437040AF28	HD6437040A (***)F28	QFP2020-112	HD6437040A***F
			HD6437040AVF16	HD6437040A(***)VF16	QFP2020-112	HD6437040A***F
			HD6437040AVX16	HD6437040A(***)VX16	TQFP1414-120	HD6437040A***X
			HD6437040ACF28	HD6437040A(***)CF28	QFP2020-112Cu*1	HD6437040A***CF
			HD6437040AVCF16	HD6437040A(***)VCF16	QFP2020-112Cu*1	HD6437040A***CF
	ROM less verion	A MASK	HD6417040AF28	HD6417040AF28	QFP2020-112	HD6417040AF28
			HD6417040AVF16	HD6417040AVF16	QFP2020-112	HD6417040AVF16
			HD6417040AVX16	HD6417040AVX16	TQFP1414-120	HD6417040AVX16
			HD6417040ACF28	HD6417040ACF28	QFP2020-112Cu*1	HD6417040ACF28
			HD6417040AVCF16	HD6417040AVCF16	QFP2020-112Cu*1	HD6417040AVCF16
SH7041A	Mask ROM verion	A MASK	HD6437041AF28	HD6437041A(***)F28	QFP2020-144	HD6437041A***F
			HD6437041AVF16	HD6437041A(***)VF16	QFP2020-144	HD6437041A***F
			HD6437041ACF28	HD6437041A(***)CF28	QFP2020-144Cu*1	HD6437041A***CF
			HD6437041AVCF16	HD6437041A(***)VCF16	QFP2020-144Cu*1	HD6437041A***CF
	ROM less verion	A MASK	HD6417041AF28	HD6417041AF28	QFP2020-144	HD6417041AF28
			HD6417041AVF16	HD6417041AVF16	QFP2020-144	HD6417041AVF16
			HD6417041ACF28	HD6417041ACF28	QFP2020-144Cu*1	HD6417041ACF28
			HD6417041AVCF16	HD6417041AVCF16	QFP2020-144Cu*1	HD6417041AVCF16
SH7042	Mask ROM verion	–	HD6437042F28	HD6437042 (***)F28	QFP2020-112	HD6437042***F
			HD6437042VF16	HD6437042 (***)VF16	QFP2020-112	HD6437042***F
	Z-TAT version	–	HD6477042F28	HD6477042F28	QFP2020-112	HD6477042F28
			HD6477042VF16	HD6477042VF16	QFP2020-112	HD6477042VF16
SH7042A	Mask ROM verion	A MASK	HD6437042AF28	HD6437042A(***)F28	QFP2020-112	HD6437042A***F
			HD6437042AVF16	HD6437042A(***)VF16	QFP2020-112	HD6437042A***F
			HD6437042AVX16	HD6437042A(***)VX16	TQFP1414-120	HD6437042A***X
			HD6437042ACF28	HD6437042A(***)CF28	QFP2020-112Cu*1	HD6437042A***CF
			HD6437042AVCF16	HD6437042A(***)VCF16	QFP2020-112Cu*1	HD6437042A***CF

Product Type		Mask Version	Product Code	Mark Code	Package	Order Model No.*2
SH7042A	Z-TAT version	A MASK	HD6477042AF28	HD6477042AF28	QFP2020-112	HD6477042AF28
			HD6477042AVF16	HD6477042AVF16	QFP2020-112	HD6477042AVF16
			HD6477042AVX16	HD6477042AVX16	TOFP1414-120	HD6477042AVX16
			HD6477042ACF28	HD6477042ACF28	QFP2020-112Cu*1	HD6477042ACF28
			HD6477042AVCF16	HD6477042AVCF16	QFP2020-112Cu*1	HD6477042AVCF16
SH7043	Mask ROM version	–	HD6437043F28	HD6437043(***)F28	QFP2020-144	HD6437043***F
			HD6437043VF16	HD6437043(***)VF16	QFP2020-144	HD6437043***F
	Z-TAT version	–	HD6477043F28	HD6477043F28	QFP2020-144	HD6477043F28
			HD6477043VF16	HD6477043VF16	QFP2020-144	HD6477043VF16
SH7043A	Mask ROM version	A MASK	HD6437043AF28	HD6437043A(***)F28	QFP2020-144	HD6437043A***F
			HD6437043AVF16	HD6437043A(***)VF16	QFP2020-144	HD6437043A***F
			HD6437043ACF28	HD6437043A(***)CF28	QFP2020-144Cu*1	HD6437043A***CF
			HD6437043AVCF16	HD6437043A(***)VCF16	QFP2020-144Cu*1	HD6437043A***CF
	Z-TAT version	A MASK	HD6477043AF28	HD6477043AF28	QFP2020-144	HD6477043AF28
			HD6477043AVF16	HD6477043AVF16	QFP2020-144	HD6477043AVF16
			HD6477043ACF28	HD6477043ACF28	QFP2020-144Cu*1	HD6477043ACF28
			HD6477043AVCF16	HD6477043AVCF16	QFP2020-144Cu*1	HD6477043AVCF16
SH7044	Mask ROM version	A MASK	HD6437044F28	HD6437044(***)F28	QFP2020-112	HD6437044***F
	F-ZTAT version		HD64F7044F28	HD64F7044F28	QFP2020-112	HD64F7044F28
SH7045	Mask ROM version	A MASK	HD6437045F28	HD6437045(***)F28	QFP2020-144	HD6437045***F
	F-ZTAT version		HD64F7045F28	HD64F7045F28	QFP2020-144	HD64F7045F28

(***) is the ROM code.

NoteS: 1. Package with Copper used as the lead material.

2. *** in the Order Model No. is the ROM code, consisting of a letter and a two-digit number (ex. E00). The letter indicates the voltage and frequency, as shown below.

- E, F, G, H: 5.0 V, 28 MHz
- P, Q, R: 3.3 V, 16 MHz

Contents

Section 1 SH7040 Series Overview

1.1 SH7040 Series Overview

The SH7040 Series (SH7040/41/42/43/44/45) CMOS single-chip microprocessors integrate a Renesas-original architecture, high-speed CPU with peripheral functions required for system configuration.

The CPU has a RISC-type instruction set. Most instructions can be executed in one clock cycle, which greatly improves instruction execution speed. In addition, the 32-bit internal-bus architecture enhances data processing power. With this CPU, it has become possible to assemble low cost, high performance/high-functioning systems, even for applications that were previously impossible with microprocessors, such as real-time control, which demands high speeds. In particular, the SH7040 series has a 1-kbyte on-chip cache, which allows an improvement in CPU performance during external memory access.

In addition, the SH7040 Series includes on-chip peripheral functions necessary for system configuration, such as large-capacity ROM and RAM, timers, a serial communication interface (SCI), an A/D converter, an interrupt controller, and I/O ports. Memory or peripheral LSIs can be connected efficiently with an external memory access support function. This greatly reduces system cost.

In addition to the masked-ROM versions of the SH7040 series, the SH7042 and SH7043 have a ZTAT™*1 version with user-programmable on-chip PROM and the SH7044 and SH7045 have an F-ZTAT™*2 version with on-chip flash memory. These versions enable users to respond quickly and flexibly to changing application specifications, growing production volumes, and other conditions.

Notes: *1 ZTAT (Zero Turn-Around Time) is a registered trademark of Renesas Technology Corp.
*2 F-ZTAT (Flexible ZTAT) is a trademark of Renesas Technology Corp.

1.1.1 SH7040 Series Features

CPU:

- Original Renesas architecture
- 32-bit internal data bus
- General-register machine
  - Sixteen 32-bit general registers
  - Three 32-bit control registers
  - Four 32-bit system registers
- RISC-type instruction set

— Instruction length: 16-bit fixed length for improved code efficiency
— Load-store architecture (basic operations are executed between registers)
— Delayed branch instructions reduce pipeline disruption during branch
— Instruction set based on C language

- Instruction execution time: one instruction/cycle (35 ns/instruction at 28.7-MHz operation)
- Address space: Architecture supports 4 Gbytes
- On-chip multiplier: multiplication operations (32 bits × 32 bits → 64 bits) and multiplication/accumulation operations (32 bits × 32 bits + 64 bits → 64 bits) executed in two to four cycles
- Five-stage pipeline

Cache Memory:

- 1-kbyte instruction cache
- Caching of instruction codes and PC relative read data
- 4-byte line length (1 longword: 2 instruction lengths)
- 256 entry cache tags
- Direct map method
- On-chip ROM/RAM, and on-chip I/O areas not objects of cache
- Used in common with on-chip RAM; 2 kbytes of on-chip RAM used as address array/data array when cache is enabled

Interrupt Controller (INTC):

- Nine external interrupt pins (NMI, $\overline{\text{IRQ0}}$–$\overline{\text{IRQ7}}$)
- Forty-three internal interrupt sources (forty-four for A mask)
- Sixteen programmable priority levels

User Break Controller (UBC):

- Generates an interrupt when the CPU or DMAC generates a bus cycle with specified conditions
- Simplifies configuration of an on-chip debugger

Bus State Controller (BSC):

- Supports external extended memory access
  — 16-bit (QFP-112, TQFP-120), or 32-bit (QFP-144) external data bus
- Memory address space divided into five areas (four areas of SRAM space, one area of DRAM space) with the following settable features:
  — Bus size (8, 16, or 32 bits)
  — Number of wait cycles

— Outputs chip-select signals for each area
— During DRAM space access:
  - Outputs $\overline{RAS}$ and $\overline{CAS}$ signals for DRAM
  - Can generate a RAS precharge time assurance Tp cycle
- DRAM burst access function
  — Supports high-speed access mode for DRAM
- DRAM refresh function
  — Programmable refresh interval
  — Supports CAS-before-RAS refresh and self-refresh modes
- Wait cycles can be inserted using an external $\overline{WAIT}$ signal
- Address data multiplex I/O devices can be accessed

Direct Memory Access Controller (DMAC) (4 Channels):

- Supports cycle-steal transfers
- Supports dual address transfer mode
- Can be switched between direct and indirect transfer modes (channel 3 only)
  — Direct transfer mode: transfers the data at the transfer source address to the transfer destination address
  — Indirect transfer mode: regards the data at the transfer source address as an address and transfers the data at that address to the transfer destination address

Data Transfer Controller (DTC):

- Data transfer independent of the CPU possible through peripheral I/O interrupt requests
- Transfer mode can be set for each interrupt factor (transfer mode set in memory)
- Multiple data transfers possible for one activating factor
- Abundant transfer modes
  — Normal mode/repeat mode/block transfer mode selectable
- Transfer unit can be set to byte/word/longword
- Interrupts activating the DTC requested of the CPU
  — Interrupts can be generated to the CPU after completion of one data transfer
  — Interrupts can be generated to the CPU after completing all designated data transfers
- Transfer can be activated by software

Multifunction Timer/Pulse Unit (MTU):

- Maximum 16 types of waveform output or maximum 16 types of pulse I/O processing possible based on 16-bit timer, 5 channels
- 16 dual-use output compare/input capture registers

- 16 independent comparators
- 8 types of counter input clock
- Input capture function
- Pulse output mode
  — One shot, toggle, PWM, phase-compensated PWM, reset-synchronized PWM
- Multiple counter synchronization function
- Phase-compensated PWM output mode
  — Non-overlapping waveform output for 6-phase inverter control
  — Automatic setting for dead time
  — PWM duty cycle can be set from 0 to 100%
  — Output off function
- Reset-synchronized PWM mode
  — 3-phase output of any duty cycle positive phase/reverse phase PWM waveforms
- Phase calculation mode
  — 2-phase encoder calculation processing

Compare Match Timer (CMT) (Two Channels):

- 16-bit free-running counter
- One compare register
- Generates an interrupt request upon compare match

Watchdog Timer (WDT) (One Channel):

- Watchdog timer or interval timer
- Count overflow can generate an internal reset, external signal, or interrupt

Serial Communication Interface (SCI) (Two Channels):

(Per Channel):

- Asynchronous or clock-synchronous mode is selectable
- Can transmit and receive simultaneously (full duplex)
- On-chip dedicated baud rate generator
- Multiprocessor communication function

I/O Ports:

- QFP 112 (SH7040, SH7042, SH7044), TQFP-120 (SH7040, SH7042)
  — Input/output: 74
  — Input: 8

— Total: 82
- QFP 144 (SH7041, SH7043, SH7045)
  — Input/output: 98
  — Input: 8
  — Total: 106

A/D Converter:

- 10 bits × 8 channels
- Conversion upon external trigger possible
- Sample and hold function: two on-chip units (two channels can be sampled simultaneously)
- Depending on the product, there is a high speed, mid-accuracy A/D on-chip type and a mid-speed, high accuracy A/D on-chip type. For details, see the product lineup.

Large Capacity On-Chip Memory:

- ROM (128 kbytes PROM, 256 kbytes/128 kbytes/64 kbytes mask ROM, 256 kbytes flash ROM)
  — SH7044, SH7045: 256 kbytes (flash ROM, mask ROM)
  — SH7042, SH7043: 128 kbytes (ZTAT, mask ROM)
  — SH7040, SH7041: 64 kbytes (mask ROM)
- RAM: 4 kbytes (2 kbytes when cache is used)

Operating Modes:

- Operating modes
  — Expanded mode with ROM disabled
  — Expanded mode with ROM enabled
  — Single-chip mode
- Processing states
  — Program execution state
  — Exception processing state
  — Bus-released state
- Power-down modes
  — Sleep mode
  — Software standby mode

Clock Pulse Generator (CPG):

- On-chip clock pulse generator
  — On-chip clock-doubling PLL circuit

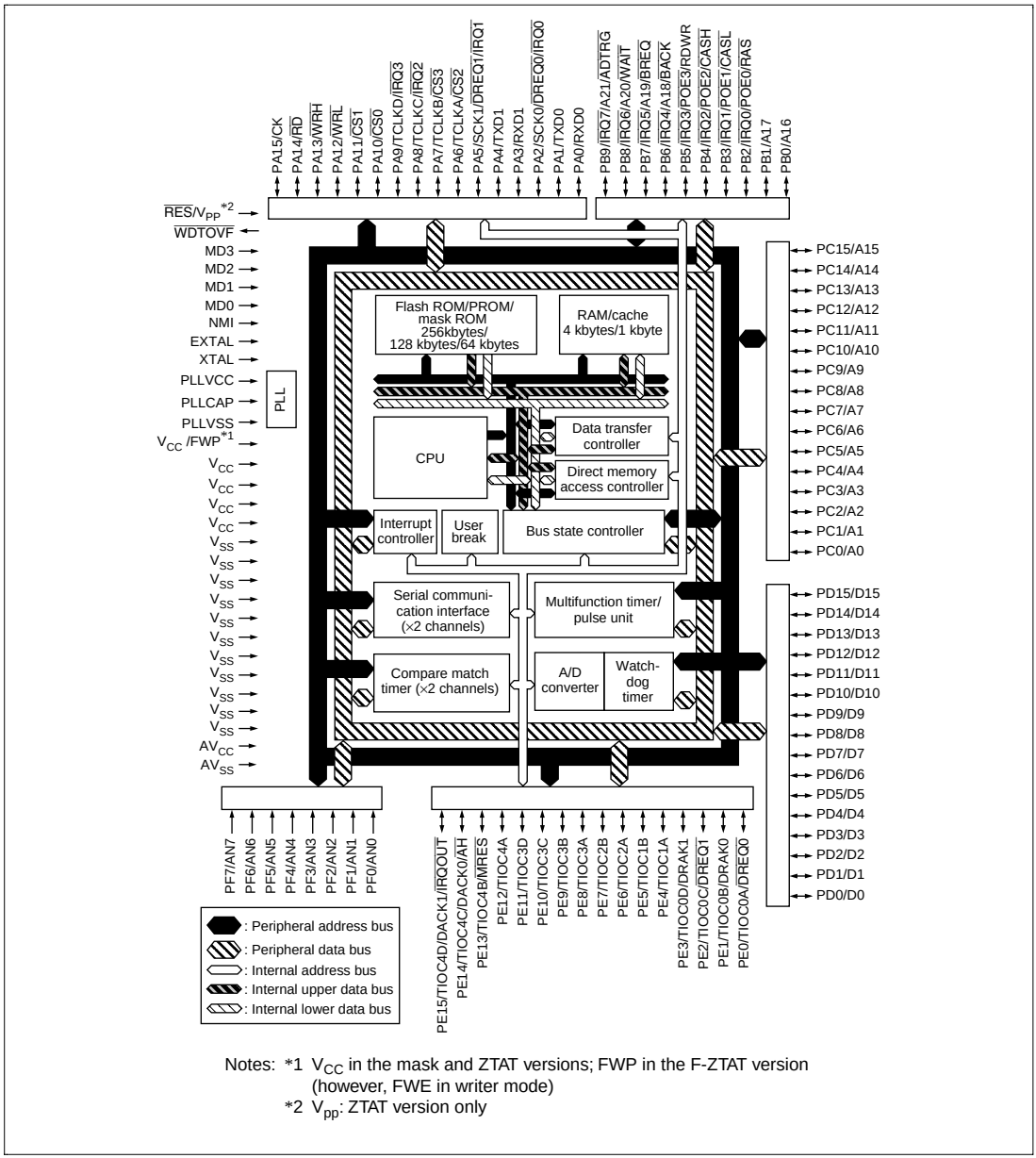
Type	Abbreviation	Mask Version	On-chip ROM	External Bus Width	A/D Accuracy (5V Version)	Package	Operating Temp	Frequency	Voltage	Type Name	Notes on the SH7040 Series Specifications (For details, see each section in this manual)						
											INTC	DTC	DMAC	MTU	A/D Converter	ROM	Electrical Characteristics
ZTAT	SH7042		128 kB	16 bits	±15LSB (High-Speed)	QFP2020-112	–20°C to 75°C	28 MHz 16 MHz	5 V 3.3 V	HD6477042F28 HD6477042VF16					See “High-Speed A/D Converter”	See “128 kB PROM”	See “Electrical Characteristics”
	SH7042A	A mask	128 kB	16 bits	±4LSB (Mid-Speed)	QFP2020-112 TQFP1414-120 QFP2020-112Cu*	–20°C to 75°C	28 MHz 16 MHz 16 MHz 28 MHz 16 MHz	5 V 3.3 V 3.3 V 5 V 3.3 V	HD6477042AF28 HD6477042AVF16 HD6477042AVX16 HD6477042ACF28 HD6477042AVCF16	Change the interrupt vectors related A/D converter	Change the DTER access methods and DTC vectors	Change the setting methods on transfer requests	Change the Usage Notes	See “Mid-Speed A/D Converter”	See “128 kB PROM”	See “Electrical Characteristics”
	SH7043		128 kB	32 bits	±15LSB (High-Speed)	QFP2020-144	–20°C to 75°C	28 MHz 16 MHz	5 V 3.3 V	HD6477043F28 HD6477043VF16					See “High-Speed A/D Converter”	See “128 kB PROM”	See “Electrical Characteristics”
	SH7043A	A mask	128 kB	32 bits	±4LSB (Mid-Speed)	QFP2020-144 QFP2020-144Cu*	–20°C to 75°C	28 MHz 16 MHz 28 MHz 16 MHz	5 V 3.3 V 5 V 3.3 V	HD6477043AF28 HD6477043AVF16 HD6477043ACF28 HD6477043AVCF16	Change the interrupt vectors related A/D converter	Change the DTER access methods and DTC vectors	Change the setting methods on transfer requests	Change the Usage Notes	See “Mid-Speed A/D Converter”	See “128 kB PROM”	See “Electrical Characteristics”
FLASH	SH7044F	A mask	256 kB	16 bits	±4LSB (Mid-Speed)	QFP2020-112	–20°C to 75°C	28 MHz	5 V	HD64F7044F28	Change the interrupt vectors related A/D converter	Change the DTER access methods and DTC vectors	Change the setting methods on transfer requests	Change the Usage Notes	See “Mid-Speed A/D Converter”	See “256 kB Flash Memory”	See “Electrical Characteristics”
	SH7045F	A mask	256 kB	32 bits	±4LSB (Mid-Speed)	QFP2020-144	–20°C to 75°C	28 MHz	5 V	HD64F7045F28	Change the interrupt vectors related A/D converter	Change the DTER access methods and DTC vectors	Change the setting methods on transfer requests	Change the Usage Notes	See “Mid-Speed A/D Converter”	See “256 kB Flash Memory”	See “Electrical Characteristics”
MASK	SH7040A	A mask	64 kB	16 bits	±4LSB (Mid-Speed)	QFP2020-112 TQFP1414-120 QFP2020-112Cu*	–20°C to 75°C	28 MHz 16 MHz 16 MHz 28 MHz 16 MHz	5 V 3.3 V 3.3 V 5 V 3.3 V	HD6437040AF28 HD6437040AVF16 HD6437040AVX16 HD6437040ACF28 HD6437040AVCF16	Change the interrupt vectors related A/D converter	Change the DTER access methods and DTC vectors	Change the setting methods on transfer requests	Change the Usage Notes	See “Mid-Speed A/D Converter”	See “64 kB Mask ROM”	See “Electrical Characteristics”
	SH7041A	A mask	64 kB	32 bits	±4LSB (Mid-Speed)	QFP2020-144 QFP2020-144Cu*	–20°C to 75°C	28 MHz 16 MHz 28 MHz 16 MHz	5 V 3.3 V 5 V 3.3 V	HD6437041AF28 HD6437041AVF16 HD6437041ACF28 HD6437041AVCF16	Change the interrupt vectors related A/D converter	Change the DTER access methods and DTC vectors	Change the setting methods on transfer requests	Change the Usage Notes	See “Mid-Speed A/D Converter”	See “64 kB Mask ROM”	See “Electrical Characteristics”
	SH7042		128 kB	16 bits	±15LSB (High-Speed)	QFP2020-112	–20°C to 75°C	28 MHz 16 MHz	5 V 3.3 V	HD6437042F28 HD6437042VF16					See “High-Speed A/D Converter”	See “128 kB Mask ROM”	See “Electrical Characteristics”
	SH7042A	A mask	128 kB	16 bits	±4LSB (Mid-Speed)	QFP2020-112 TQFP1414-120 QFP2020-112Cu*	–20°C to 75°C	28 MHz 16 MHz 16 MHz 28 MHz 16 MHz	5 V 3.3 V 3.3 V 5 V 3.3 V	HD6437042AF28 HD6437042AVF16 HD6437042AVX16 HD6437042ACF28 HD6437042AVCF16	Change the interrupt vectors related A/D converter	Change the DTER access methods and DTC vectors	Change the setting methods on transfer requests	Change the Usage Notes	See “Mid-Speed A/D Converter”	See “128 kB Mask ROM”	See “Electrical Characteristics”
	SH7043		128 kB	32 bits	±15LSB (High-Speed)	QFP2020-144	–20°C to 75°C	28 MHz 16 MHz	5 V 3.3 V	HD6437043F28 HD6437043VF16					See “High-Speed A/D Converter”	See “128 kB Mask ROM”	See “Electrical Characteristics”
	SH7043A	A mask	128 kB	32 bits	±4LSB (Mid-Speed)	QFP2020-144 QFP2020-144Cu*	–20°C to 75°C	28 MHz 16 MHz 28 MHz 16 MHz	5 V 3.3 V 5 V 3.3 V	HD6437043AF28 HD6437043AVF16 HD6437043ACF28 HD6437043AVCF16	Change the interrupt vectors related A/D converter	Change the DTER access methods and DTC vectors	Change the setting methods on transfer requests	Change the Usage Notes	See “Mid-Speed A/D Converter”	See “128 kB Mask ROM”	See “Electrical Characteristics”

Type	Abbreviation	Mask Version	On-chip ROM	External Bus Width	A/D Accuracy (5V Version)	Package	Operating Temp	Frequency	Voltage	Type Name	Notes on the SH7040 Series Specifications (For details, see each section in this manual)						
											INTC	DTC	DMAC	MTU	A/D Converter	ROM	Electrical Characteristics
MASK	SH7044	A mask	256 kB	16 bits	±4LSB (Mid-Speed)	QFP2020-112	–20°C to 75°C	28 MHz	5 V	HD6437044F28	Change the interrupt vectors related A/D converter	Change the DTER access methods and DTC vectors	Change the setting methods on transfer requests	Change the Usage Notes	See “Mid-Speed A/D Converter”	See “256 kB Mask ROM”	See “Electrical Characteristics”
	SH7045	A mask	256kB	32bits	±4LSB (Mid-Speed)	QFP2020-144	-20°C to 75°C	28 MHz	5 V	HD6437045F28	Change the interrupt vectors related A/D converter	Change the DTER access methods and DTC vectors	Change the setting methods on transfer requests	Change the Usage Notes	See “Mid-Speed A/D Converter”	See “256 kB Mask ROM”	See “Electrical Characteristics”
ROM less	SH7040A	A mask		16 bits	±4LSB (Mid-Speed)	QFP2020-112 TQFP1414-120 QFP2020-112Cu*	-20°C to 75°C	28 MHz 16 MHz 16 MHz 28 MHz 16 MHz	5 V 3.3 V 3.3 V 5 V 3.3 V	HD6417040AF28 HD6417040AVF16 HD6417040AVX16 HD6417040ACF28 HD6417040AVCF16	Change the interrupt vectors related A/D converter	Change the DTER access methods and DTC vectors	Change the setting methods on transfer requests	Change the Usage Notes	See “Mid-Speed A/D Converter”		See “Electrical Characteristics”
	SH7041A	A mask		32 bits	±4LSB (Mid-Speed)	QFP2020-144 QFP2020-144Cu*	-20°C to 75°C	28 MHz 16 MHz 28 MHz 16 MHz	5 V 3.3 V 5 V 3.3 V	HD6417041AF28 HD6417041AVF16 HD6417041ACF28 HD6417041AVCF16	Change the interrupt vectors related A/D converter	Change the DTER access methods and DTC vectors	Change the setting methods on transfer requests	Change the Usage Notes	See “Mid-Speed A/D Converter”		See “Electrical Characteristics”

Note: * Package with Copper used as the lead material.

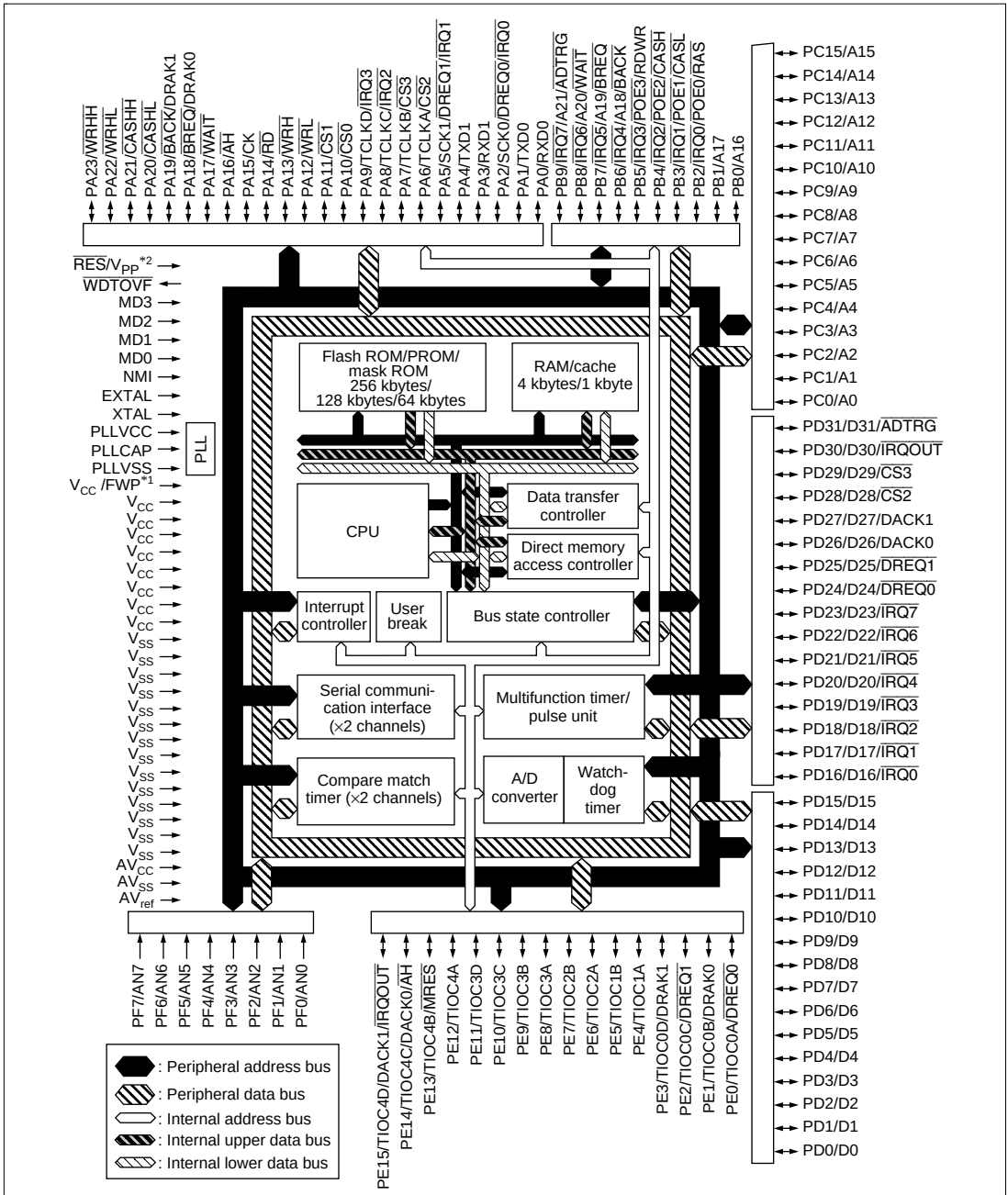
1.2 Block Diagram

Figure 1.1 is a block diagram of the SH7040 Series QFP-112 pin and TQFP-120 pin. Figure 1.2 is a block diagram of the SH7040 Series QFP-144 pin.



Notes: *1 V_{CC} in the mask and ZTAT versions; FWP in the F-ZTAT version (however, FWE in writer mode)
*2 V_{pp}: ZTAT version only

Figure 1.1 Block Diagram of the SH7040, SH7042, SH7044 (QFP-112 Pin), SH7040, SH7042 (TQFP-120 pin)



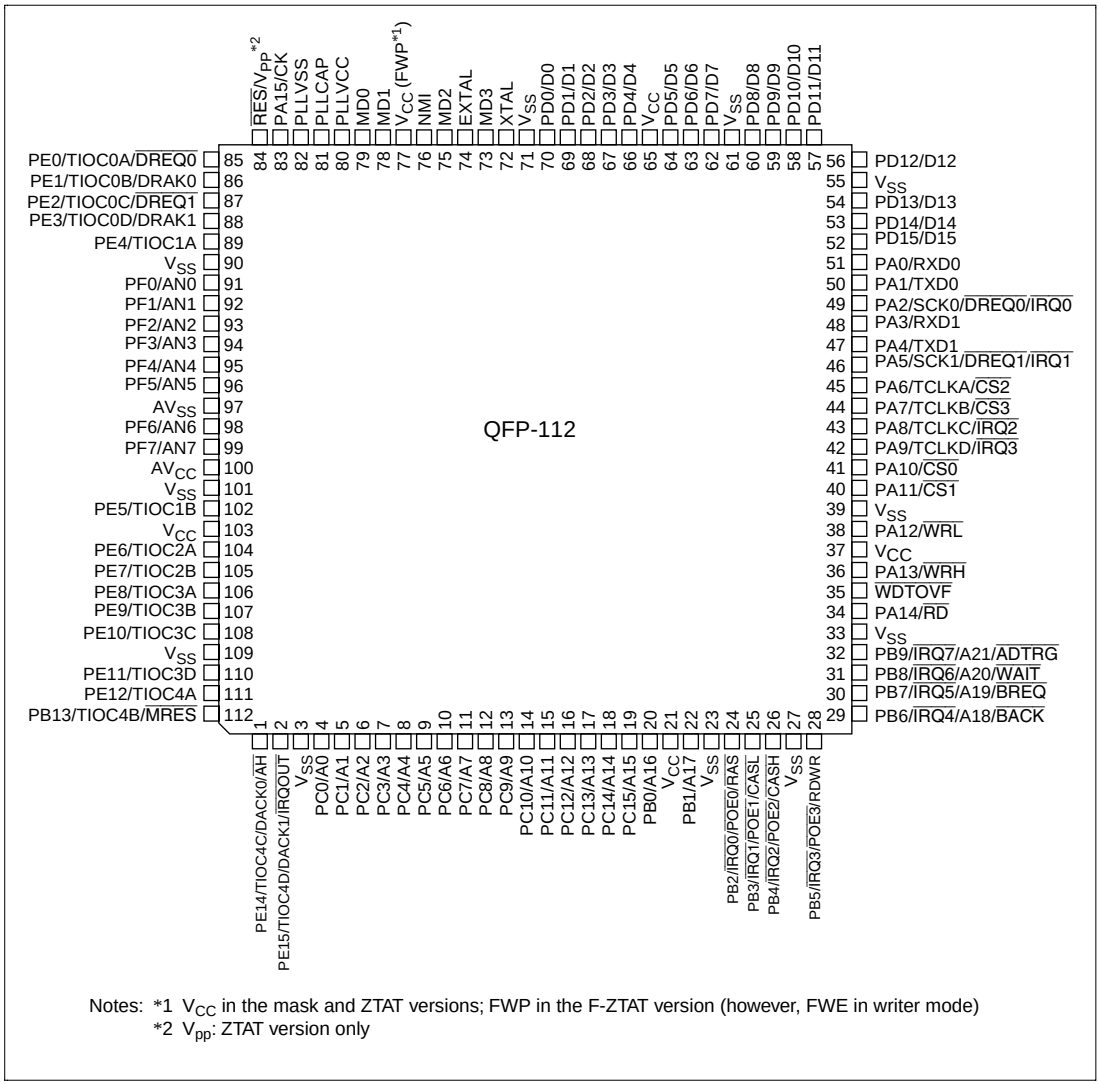
Notes: *1 V_{CC} in the mask and ZTAT versions; FWP in the F-ZTAT version (however, FWE in writer mode)
*2 V_{pp}: ZTAT version only

Figure 1.2 Block Diagram of the SH7041, SH7043, SH7045 (QFP-144 Pin)

1.3 Pin Arrangement and Pin Functions

1.3.1 Pin Arrangment

Figure 1.3 shows the pin arrangement for the QFP-112 (top view).



Notes: *1 V_{CC} in the mask and ZTAT versions; FWP in the F-ZTAT version (however, FWE in writer mode)
*2 V_{pp}: ZTAT version only

Figure 1.3 SH7040, SH7042, SH7044 Pin Arrangement (QFP-112 Top View)

Figure 1.4 shows the pin arrangement for the TFP-120 (top view).

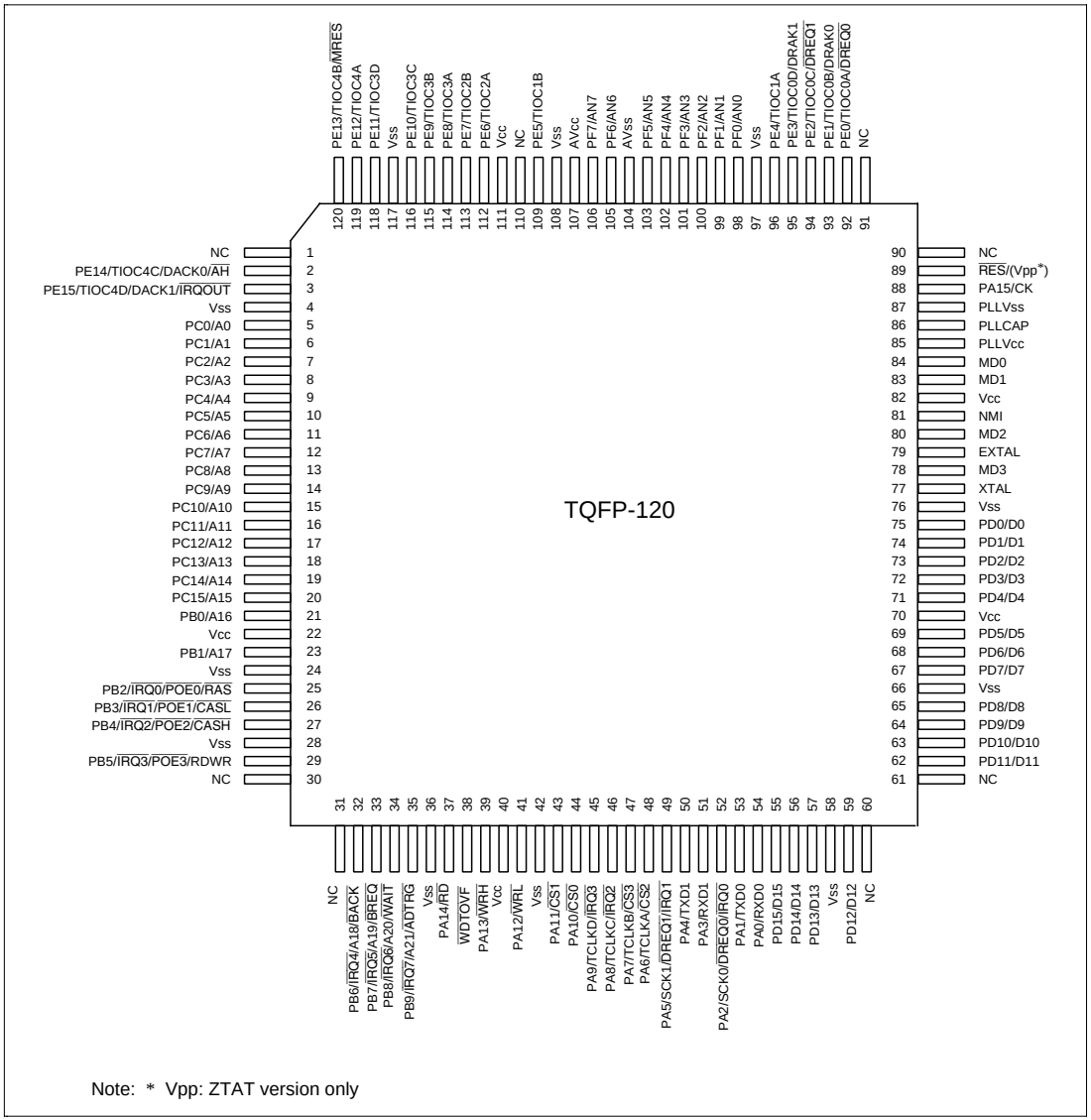
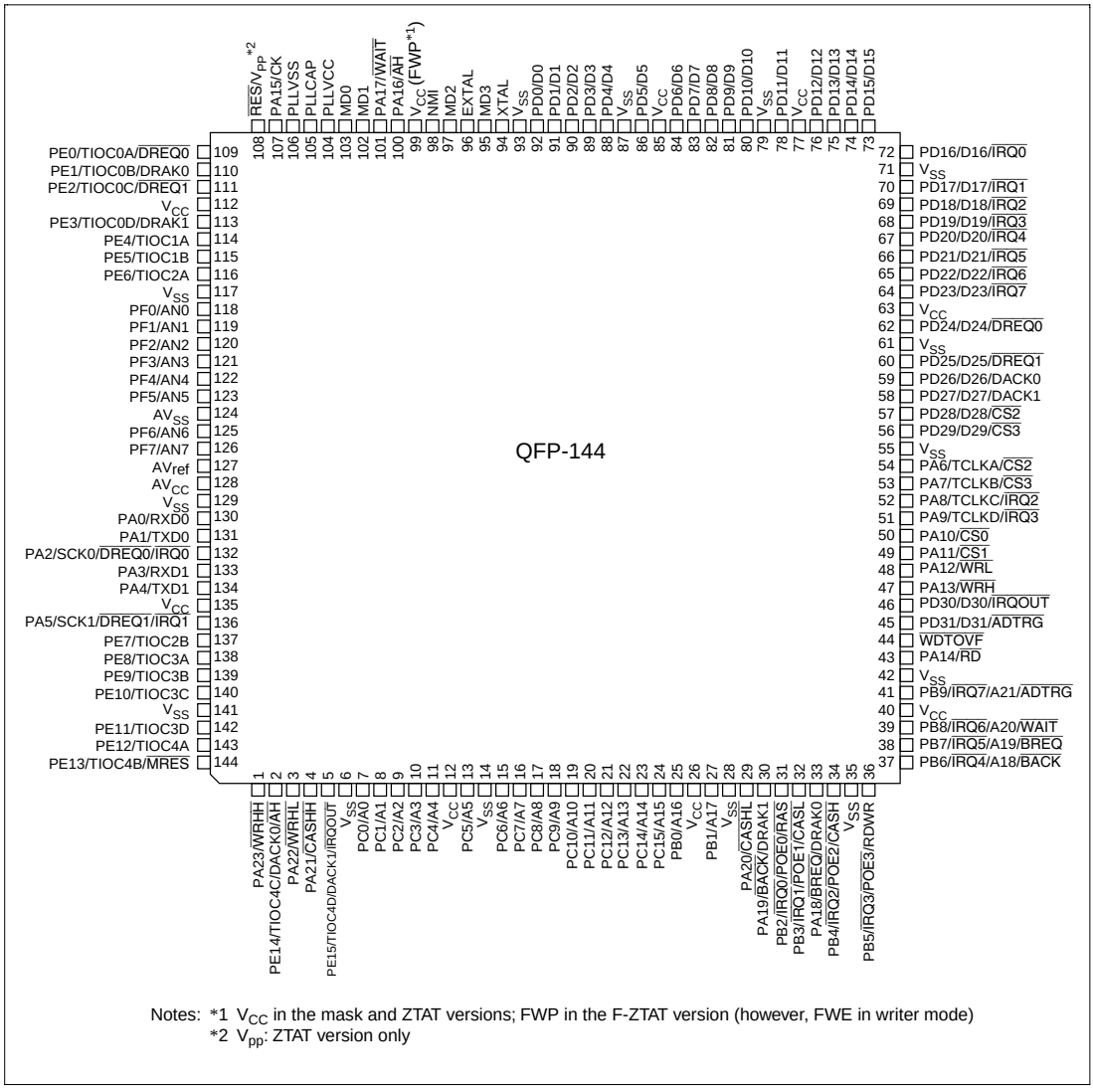


Figure 1.4 SH7040, SH7042 Pin Arrangement (TQFP-120 Top View)

Figure 1.5 shows the pin arrangement for the QFP-144 (top view).



Notes: *1 V_{CC} in the mask and ZTAT versions; FWP in the F-ZTAT version (however, FWE in writer mode)
*2 V_{pp}: ZTAT version only

Figure 1.5 SH7041, SH7043, SH7045 Pin Arrangement (QFP-144 Top View)

1.3.2 Pin Arrangement by Mode

Table 1.2 Pin Arrangement by Mode for SH7040, SH7042 (QFP-112 Pin)

Pin No.	MCU Mode	PROM Mode
1	PE14/TIOC4C/DACK0/$\overline{AH}$	V_{CC}
2	PE15/TIOC4D/DACK1/$\overline{IRQOUT}$	$\overline{CE}$
3	V_{SS}	V_{SS}
4	PC0/A0	A0
5	PC1/A1	A1
6	PC2/A2	A2
7	PC3/A3	A3
8	PC4/A4	A4
9	PC5/A5	A5
10	PC6/A6	A6
11	PC7/A7	A7
12	PC8/A8	A8
13	PC9/A9	NC
14	PC10/A10	A10
15	PC11/A11	A11
16	PC12/A12	A12
17	PC13/A13	A13
18	PC14/A14	A14
19	PC15/A15	A15
20	PB0/A16	A16
21	V_{CC}	V_{CC}
22	PB1/A17	NC
23	V_{SS}	V_{SS}
24	PB2/$\overline{IRQ0}$/$\overline{POE0}$/$\overline{RAS}$	NC
25	PB3/$\overline{IRQ1}$/$\overline{POE1}$/$\overline{CASL}$	$\overline{OE}$
26	PB4/$\overline{IRQ2}$/$\overline{POE2}$/$\overline{CASH}$	$\overline{PGM}$
27	V_{SS}	V_{SS}
28	PB5/$\overline{IRQ3}$/$\overline{POE3}$/RDWR	V_{CC}

Table 1.2 Pin Arrangement by Mode for SH7040, SH7042 (QFP-112 Pin) (cont)

Pin No.	MCU Mode	PROM Mode
29	PB6/$\overline{\text{IRQ4}}$/A18/$\overline{\text{BACK}}$	NC
30	PB7/$\overline{\text{IRQ5}}$/A19/$\overline{\text{BREQ}}$	NC
31	PB8/$\overline{\text{IRQ6}}$/A20/$\overline{\text{WAIT}}$	NC
32	PB9/$\overline{\text{IRQ7}}$/A21/$\overline{\text{ADTRG}}$	NC
33	V_{SS}	V_{SS}
34	PA14/$\overline{\text{RD}}$	NC
35	$\overline{\text{WDTOVF}}$	NC
36	PA13/$\overline{\text{WRH}}$	NC
37	V_{CC}	V_{CC}
38	PA12/$\overline{\text{WRL}}$	NC
39	V_{SS}	V_{SS}
40	PA11/$\overline{\text{CS1}}$	NC
41	PA10/$\overline{\text{CS0}}$	NC
42	PA9/TCLKD/$\overline{\text{IRQ3}}$	NC
43	PA8/TCLKC/$\overline{\text{IRQ2}}$	NC
44	PA7/TCLKB/$\overline{\text{CS3}}$	NC
45	PA6/TCLKA/$\overline{\text{CS2}}$	NC
46	PA5/SCK1/$\overline{\text{DREQ1}}$/$\overline{\text{IRQI}}$	NC
47	PA4/TXD1	NC
48	PA3 /RXD1	NC
49	PA2/SCK0/$\overline{\text{DREQ0}}$/$\overline{\text{IRQ0}}$	NC
50	PA1/TXD0	NC
51	PA0/RXD0	NC
52	PD15/D15	NC
53	PD14/D14	NC
54	PD13/D13	NC
55	V_{SS}	V_{SS}
56	PD12/D12	NC
57	PD11/D11	NC
58	PD10/D10	NC

Table 1.2 Pin Arrangement by Mode for SH7040, SH7042 (QFP-112 Pin) (cont)

Pin No.	MCU Mode	PROM Mode
59	PD9/D9	NC
60	PD8/D8	NC
61	V_{SS}	V_{SS}
62	PD7/D7	D7
63	PD6/D6	D6
64	PD5/D5	D5
65	V_{CC}	V_{CC}
66	PD4/D4	D4
67	PD3/D3	D3
68	PD2/D2	D2
69	PD1/D1	D1
70	PD0/D0	D0
71	V_{SS}	V_{SS}
72	XTAL	NC
73	MD3	V_{CC}
74	EXTAL	V_{SS}
75	MD2	V_{CC}
76	NMI	A9
77	V_{CC}	V_{CC}
78	MD1	V_{CC}
79	MD0	V_{CC}
80	PLLVCC	V_{CC}
81	PLLCAP	V_{SS}
82	PLLVSS	V_{SS}
83	PA15/CK	NC
84	$\overline{RES}$	V_{PP}
85	PE0/TIOC0A/$\overline{DREQ0}$	NC
86	PE1/TIOC0B/DRAK0	NC
87	PE2/TIOC0C/$\overline{DREQ1}$	NC
88	PE3/TIOC0D/DRAK1	NC

Table 1.2 Pin Arrangement by Mode for SH7040, SH7042 (QFP-112 Pin) (cont)

Pin No.	MCU Mode	PROM Mode
89	PE4/TIOC1A	NC
90	V_{SS}	V_{SS}
91	PF0/AN0	V_{SS}
92	PF1/AN1	V_{SS}
93	PF2/AN2	V_{SS}
94	PF3/AN3	V_{SS}
95	PF4/AN4	V_{SS}
96	PF5/AN5	V_{SS}
97	AV_{SS}	V_{SS}
98	PF6/AN6	V_{SS}
99	PF7/AN7	V_{SS}
100	AV_{CC}	V_{CC}
101	V_{SS}	V_{SS}
102	PE5/TIOC1B	NC
103	V_{CC}	V_{CC}
104	PE6/TIOC2A	NC
105	PE7/TIOC2B	NC
106	PE8/TIOC3A	NC
107	PE9/TIOC3B	NC
108	PE10/TIOC3C	NC
109	V_{SS}	V_{SS}
110	PE11/TIOC3D	NC
111	PE12/TIOC4A	NC
112	PE13/TIOC4B/$\overline{\text{MRES}}$	NC

Table 1.3 Pin Arrangement by Mode for SH7040, SH7042 (TQFP-120 Pin)

TQFP120 Pin No.	MCU Mode	PROM Mode
1	NC	NC
2	PE14/TIOC4C/DACK0/$\overline{AH}$	V_{CC}
3	PE15/TIOC4D/DACK1/$\overline{IRQOUT}$	CE
4	V_{SS}	V_{SS}
5	PC0/A0	A0
6	PC1/A1	A1
7	PC2/A2	A2
8	PC3/A3	A3
9	PC4/A4	A4
10	PC5/A5	A5
11	PC6/A6	A6
12	PC7/A7	A7
13	PC8/A8	A8
14	PC9/A9	NC
15	PC10/A10	A10
16	PC11/A11	A11
17	PC12/A12	A12
18	PC13/A13	A13
19	PC14/A14	A14
20	PC15/A15	A15
21	PB0/A16	A16
22	V_{CC}	V_{CC}
23	PB1/A17	NC
24	V_{SS}	V_{SS}
25	PB2/$\overline{IRQ0}$/$\overline{POE0}$/$\overline{RAS}$	NC
26	PB3/$\overline{IRQ1}$/$\overline{POE1}$/$\overline{CASL}$	OE
27	PB4/$\overline{IRQ2}$/$\overline{POE2}$/$\overline{CASH}$	PGM
28	V_{SS}	V_{SS}
29	PB5/$\overline{IRQ3}$/$\overline{POE3}$/RDWR	V_{CC}
30	NC	NC
31	NC	NC

Table 1.3 Pin Arrangement by Mode for SH7040, SH7042 (TQFP-120 Pin) (cont)

TQFP120 Pin No.	MCU Mode	PROM Mode
32	PB6/$\overline{\text{IRQ4}}$/A18/$\overline{\text{BACK}}$	NC
33	PB7/$\overline{\text{IRQ5}}$/A19/$\overline{\text{BREQ}}$	NC
34	PB8/$\overline{\text{IRQ6}}$/A20/$\overline{\text{WAIT}}$	NC
35	PB9/$\overline{\text{IRQ7}}$/A21/$\overline{\text{ADTRG}}$	NC
36	V_{SS}	V_{SS}
37	PA14/$\overline{\text{RD}}$	NC
38	$\overline{\text{WDTOVF}}$	NC
39	PA13/$\overline{\text{WRH}}$	NC
40	V_{CC}	V_{CC}
41	PA12/$\overline{\text{WRL}}$	NC
42	V_{SS}	V_{SS}
43	PA11/$\overline{\text{CS1}}$	NC
44	PA10/$\overline{\text{CS0}}$	NC
45	PA9/TCLKD/$\overline{\text{IRQ3}}$	NC
46	PA8/TCLKC/$\overline{\text{IRQ2}}$	NC
47	PA7/TCLKB/$\overline{\text{CS3}}$	NC
48	PA6/TCLKA/$\overline{\text{CS2}}$	NC
49	PA5/SCK1/$\overline{\text{DREQ1}}$/$\overline{\text{IRQ1}}$	NC
50	PA4/TXD1	NC
51	PA3/RXD2	NC
52	PA2/SCK0/$\overline{\text{DREQ0}}$/$\overline{\text{IRQ0}}$	NC
53	PA1/TXD0	NC
54	PA0/RXD0	NC
55	PD15/D15	NC
56	PD14/D14	NC
57	PD13/D13	NC
58	V_{SS}	V_{SS}
59	PD12/D12	NC
60	NC	NC
61	NC	NC
62	PD11/D11	NC

Table 1.3 Pin Arrangement by Mode for SH7040, SH7042 (TQFP-120 Pin) (cont)

TQFP120 Pin No.	MCU Mode	PROM Mode
63	PD10/D10	NC
64	PD9/D9	NC
65	PD8/D8	NC
66	V_{SS}	V_{SS}
67	PD7/D7	D7
68	PD6/D6	D6
69	PD5/D5	D5
70	V_{CC}	V_{CC}
71	PD4/D4	D4
72	PD3/D3	D3
73	PD2/D2	D2
74	PD1/D1	D1
75	PD0/D0	D0
76	V_{SS}	V_{SS}
77	XTAL	NC
78	MD3	V_{CC}
79	EXTAL	V_{SS}
80	MD2	V_{CC}
81	NMI	A9
82	V_{CC}	V_{CC}
83	MD1	V_{CC}
84	MD0	V_{CC}
85	$PLLV_{CC}$	V_{CC}
86	PLLCAP	V_{SS}
87	$PLLV_{SS}$	V_{SS}
88	PA15/CK	NC
89	$\overline{RES}$	V_{PP}
90	NC	NC
91	NC	NC
92	PE0/TIOC0A/$\overline{DREQ0}$	NC
93	PE1/TIOC0B/$\overline{DRAK0}$	NC

Table 1.3 Pin Arrangement by Mode for SH7040, SH7042 (TQFP-120 Pin) (cont)

TQFP120 Pin No.	MCU Mode	PROM Mode
94	PE2/TIOC0C/$\overline{\text{DREQ1}}$	NC
95	PE3/TIOC0D/$\overline{\text{DRAK1}}$	NC
96	PE4/TIOC1A	NC
97	V_{SS}	V_{SS}
98	PF0/AN0	V_{SS}
99	PF1/AN1	V_{SS}
100	PF2/AN2	V_{SS}
101	PF3/AN3	V_{SS}
102	PF4/AN4	V_{SS}
103	PF5/AN5	V_{SS}
104	AV_{SS}	V_{SS}
105	PF6/AN6	V_{SS}
106	PF7/AN7	V_{SS}
107	AV_{CC}	V_{CC}
108	V_{SS}	V_{SS}
109	PE5/TIOC1B	NC
110	NC	NC
111	V_{CC}	V_{CC}
112	PE6/TIOC2A	NC
113	PE7/TIOC2B	NC
114	PE8/TIOC3A	NC
115	PE9/TIOC3B	NC
116	PE10/TIOC3C	NC
117	V_{SS}	V_{SS}
118	PE11/TIOC3D	NC
119	PE12/TIOC4A	NC
120	PE13/TIOC4B/$\overline{\text{MRES}}$	NC

Table 1.4 Pin Arrangement by Mode for SH7041, SH7043 (QFP-144 Pin)

Pin No.	MCU Mode	PROM Mode
1	PA23/$\overline{\text{WRHH}}$	NC
2	PE14/TIOC4C/DACK0/$\overline{\text{AH}}$	V_{CC}
3	PA22/$\overline{\text{WRHL}}$	NC
4	PA21/$\overline{\text{CASHH}}$	NC
5	PE15/TIOC4D/DACK1/$\overline{\text{IRQOUT}}$	$\overline{\text{CE}}$
6	V_{SS}	V_{SS}
7	PC0/A0	A0
8	PC1/A1	A1
9	PC2/A2	A2
10	PC3/A3	A3
11	PC4/A4	A4
12	V_{CC}	V_{CC}
13	PC5/A5	A5
14	V_{SS}	V_{SS}
15	PC6/A6	A6
16	PC7/A7	A7
17	PC8/A8	A8
18	PC9/A9	NC
19	PC10/A10	A10
20	PC11/A11	A11
21	PC12/A12	A12
22	PC13/A13	A13
23	PC14/A14	A14
24	PC15/A15	A15
25	PB0/A16	A16
26	V_{CC}	V_{CC}
27	PB1/A17	NC
28	V_{SS}	V_{SS}
29	PA20/$\overline{\text{CASHL}}$	NC
30	PA19/$\overline{\text{BACK}}$/DRAK1	NC

Table 1.4 Pin Arrangement by Mode for SH7041, SH7043 (QFP-144 Pin) (cont)

Pin No.	MCU Mode	PROM Mode
31	PB2/$\overline{\text{IRQ0}}$/$\overline{\text{POE0}}$/$\overline{\text{RAS}}$	NC
32	PB3/$\overline{\text{IRQ1}}$/$\overline{\text{POE1}}$/$\overline{\text{CASL}}$	$\overline{\text{OE}}$
33	PA18/$\overline{\text{BREQ}}$/DRAK0	NC
34	PB4/$\overline{\text{IRQ2}}$/$\overline{\text{POE2}}$/$\overline{\text{CASH}}$	$\overline{\text{PGM}}$
35	V_{SS}	V_{SS}
36	PB5/$\overline{\text{IRQ3}}$/$\overline{\text{POE3}}$/RDWR	V_{CC}
37	PB6/$\overline{\text{IRQ4}}$/A18/$\overline{\text{BACK}}$	NC
38	PB7/$\overline{\text{IRQ5}}$/A19/$\overline{\text{BREQ}}$	NC
39	PB8/$\overline{\text{IRQ6}}$/A20/$\overline{\text{WAIT}}$	NC
40	V_{CC}	V_{CC}
41	PB9/$\overline{\text{IRQ7}}$/A21/$\overline{\text{ADTRG}}$	NC
42	V_{SS}	V_{SS}
43	PA14/$\overline{\text{RD}}$	NC
44	$\overline{\text{WDTOVF}}$	NC
45	PD31/D31/$\overline{\text{ADTRG}}$	NC
46	PD30/D30/$\overline{\text{IRQOUT}}$	NC
47	PA13/$\overline{\text{WRH}}$	NC
48	PA12/$\overline{\text{WRL}}$	NC
49	PA11/$\overline{\text{CS1}}$	NC
50	PA10/$\overline{\text{CS0}}$	NC
51	PA9/TCLKD/$\overline{\text{IRQ3}}$	NC
52	PA8/TCLKC/$\overline{\text{IRQ2}}$	NC
53	PA7/TCLKB/$\overline{\text{CS3}}$	NC
54	PA6/TCLKA/$\overline{\text{CS2}}$	NC
55	V_{SS}	V_{SS}
56	PD29/D29/$\overline{\text{CS3}}$	NC
57	PD28/D28/$\overline{\text{CS2}}$	NC
58	PD27/D27/DACK1	NC
59	PD26/D26/DACK0	NC
60	PD25/D25/$\overline{\text{DREQ1}}$	NC

Table 1.4 Pin Arrangement by Mode for SH7041, SH7043 (QFP-144 Pin) (cont)

Pin No.	MCU Mode	PROM Mode
61	V_{SS}	V_{SS}
62	PD24/D24/$\overline{\text{DREQ0}}$	NC
63	V_{CC}	V_{CC}
64	PD23/D23/$\overline{\text{IRQ7}}$	NC
65	PD22/D22/$\overline{\text{IRQ6}}$	NC
66	PD21/D21/$\overline{\text{IRQ5}}$	NC
67	PD20/D20/$\overline{\text{IRQ4}}$	NC
68	PD19/D19/$\overline{\text{IRQ3}}$	NC
69	PD18/D18/$\overline{\text{IRQ2}}$	NC
70	PD17/D17/$\overline{\text{IRQ1}}$	NC
71	V_{SS}	V_{SS}
72	PD16/D16/$\overline{\text{IRQ0}}$	NC
73	PD15/D15	NC
74	PD14/D14	NC
75	PD13/D13	NC
76	PD12/D12	NC
77	V_{CC}	V_{CC}
78	PD11/D11	NC
79	V_{SS}	V_{SS}
80	PD10/D10	NC
81	PD9/D9	NC
82	PD8/D8	NC
83	PD7/D7	D7
84	PD6/D6	D6
85	V_{CC}	V_{CC}
86	PD5 /D5	D5
87	V_{SS}	V_{SS}
88	PD4/D4	D4
89	PD3/D3	D3
90	PD2/D2	D2

Table 1.4 Pin Arrangement by Mode for SH7041, SH7043 (QFP-144 Pin) (cont)

Pin No.	MCU Mode	PROM Mode
91	PD1/D1	D1
92	PD0/D0	D0
93	V_{SS}	V_{SS}
94	XTAL	NC
95	MD3	V_{CC}
96	EXTAL	V_{SS}
97	MD2	V_{CC}
98	NMI	A9
99	V_{CC}	V_{CC}
100	PA16/$\overline{AH}$	NC
101	PA17/$\overline{WAIT}$	NC
102	MD1	V_{CC}
103	MD0	V_{CC}
104	PLLVCC	V_{CC}
105	PLLCAP	V_{SS}
106	PLLVSS	V_{SS}
107	PA15/CK	NC
108	$\overline{RES}$	V_{PP}
109	PE0/TIOC0A/$\overline{DREQ0}$	NC
110	PE1/TIOC0B/DRAK0	NC
111	PE2/TIOC0C/$\overline{DREQ1}$	NC
112	V_{CC}	V_{CC}
113	PE3/TIOC0D/DRAK1	NC
114	PE4/TIOC1A	NC
115	PE5/TIOC1B	NC
116	PE6/TIOC2A	NC
117	V_{SS}	V_{SS}
118	PF0/AN0	V_{SS}
119	PF1/AN1	V_{SS}
120	PF2/AN2	V_{SS}

Table 1.4 Pin Arrangement by Mode for SH7041, SH7043 (QFP-144 Pin) (cont)

Pin No.	MCU Mode	PROM Mode
121	PF3/AN3	V_{SS}
122	PF4/AN4	V_{SS}
123	PF5/AN5	V_{SS}
124	AV_{SS}	V_{SS}
125	PF6/AN6	V_{SS}
126	PF7/AN7	V_{SS}
127	AVref	V_{CC}
128	AV_{CC}	V_{CC}
129	V_{SS}	V_{SS}
130	PA0/RXD0	NC
131	PA1/TXD0	NC
132	PA2/SCK0/$\overline{\text{DREQ0}}$ /$\overline{\text{IREQ0}}$	NC
133	PA3/RXD1	NC
134	PA4/TXD1	NC
135	V_{CC}	V_{CC}
136	PA5 /SCK1/$\overline{\text{DREQ1}}$/$\overline{\text{IREQ1}}$	NC
137	PE7/TIOC2B	NC
138	PE8/TIOC3A	NC
139	PE9/TIOC3B	NC
140	PE10/TIOC3C	NC
141	V_{SS}	V_{SS}
142	PE11/TIOC3D	NC
143	PE12/TIOC4A	NC
144	PE13/TIOC4B /$\overline{\text{MRES}}$	NC

Table 1.5 Pin Arrangement by Mode for SH7044 (QFP-112 Pin)

PinNo.	MCU	Writer mode
1	PE14/TIOC4C/DACK0/$\overline{AH}$	NC
2	PE15/TIOC4D/DACK1/$\overline{IRQOUT}$	NC
3	V_{SS}	V_{SS}
4	PC0/A0	A0
5	PC1/A1	A1
6	PC2/A2	A2
7	PC3/A3	A3
8	PC4/A4	A4
9	PC5/A5	A5
10	PC6/A6	A6
11	PC7/A7	A7
12	PC8/A8	A8
13	PC9/A9	A9
14	PC10/A10	A10
15	PC11/A11	A11
16	PC12/A12	A12
17	PC13/A13	A13
18	PC14/A14	A14
19	PC15/A15	A15
20	PB0/A16	A16
21	V_{CC}	V_{CC}
22	PB1/A17	NC
23	V_{SS}	V_{SS}
24	PB2/$\overline{IRQ0}$/$\overline{POE0}$/$\overline{RAS}$	NC
25	PB3/$\overline{IRQ1}$/$\overline{POE1}$/$\overline{CASL}$	NC
26	PB4/$\overline{IRQ2}$/$\overline{POE2}$/$\overline{CASH}$	A17
27	V_{SS}	V_{SS}
28	PB5/$\overline{IRQ3}$/$\overline{POE3}$/RDWR	NC
29	PB6/$\overline{IRQ4}$/A18/$\overline{BACK}$	NC
30	PB7/$\overline{IRQ5}$/A19/$\overline{BREQ}$	NC
31	PB8/$\overline{IRQ6}$/A20/$\overline{WAIT}$	NC
32	PB9/$\overline{IRQ7}$/A21/$\overline{ADTRG}$	NC

Table 1.5 Pin Arrangement by Mode for SH7044 (QFP-112 Pin) (cont)

PinNo.	MCU	Writer mode
33	V_{SS}	V_{SS}
34	PA14/$\overline{RD}$	NC
35	$\overline{WDTOVF}$	NC
36	PA13/$\overline{WRH}$	NC
37	V_{CC}	V_{CC}
38	PA12/$\overline{WRL}$	NC
39	V_{SS}	V_{SS}
40	PA11/$\overline{CS1}$	NC
41	PA10/$\overline{CS0}$	NC
42	PA9/TCLKD/$\overline{IRQ3}$	$\overline{CE}$
43	PA8/TCLKC/$\overline{IRQ2}$	$\overline{OE}$
44	PA7/TCLKB/$\overline{CS3}$	$\overline{WE}$
45	PA6/TCLKA/$\overline{CS2}$	NC
46	PA5/SCK1/$\overline{DREQ1}$/$\overline{IRQ1}$	V_{CC}
47	PA4/TXD1	NC
48	PA3/RXD1	NC
49	PA2/SCK0/$\overline{DREQ0}$/$\overline{IRQ0}$	V_{CC}
50	PA1/TXD0	V_{CC}
51	PA0/RXD0	NC
52	PD15/D15	NC
53	PD14/D14	NC
54	PD13/D13	NC
55	V_{SS}	V_{SS}
56	PD12/D12	NC
57	PD11/D11	NC
58	PD10/D10	NC
59	PD9/D9	NC
60	PD8/D8	NC
61	V_{SS}	V_{SS}
62	PD7/D7	D7
63	PD6/D6	D6
64	PD5/D5	D5

Table 1.5 Pin Arrangement by Mode for SH7044 (QFP-112 Pin) (cont)

PinNo.	MCU	Writer mode
65	V_{CC}	V_{CC}
66	PD4/D4	D4
67	PD3/D3	D3
68	PD2/D2	D2
69	PD1/D1	D1
70	PD0/D0	D0
71	V_{SS}	V_{SS}
72	XTAL	XTAL
73	MD3	MD3
74	EXTAL	EXTAL
75	MD2	MD2
76	NMI	V_{CC}
77	V_{CC} (FWP)*	FWE
78	MD1	MD1
79	MD0	MD0
80	$PLLV_{CC}$	$PLLV_{CC}$
81	PLLCAP	PLLCAP
82	$PLLV_{SS}$	$PLLV_{SS}$
83	PA15/CK	NC
84	$\overline{RES}$	$\overline{RES}$
85	PE0/TIOCA/$\overline{DREQ0}$	NC
86	PE1/TIOCB/DRAK0	NC
87	PE2/TIOCC/$\overline{DREQ1}$	NC
88	PE3/TIOCD/DRAK1	NC
89	PE4/TIOC1A	NC
90	V_{SS}	V_{SS}
91	PF0/AN0	V_{SS}
92	PF1/AN1	V_{SS}
93	PF2/AN2	V_{SS}
94	PF3/AN3	V_{SS}
95	PF4/AN4	V_{SS}
96	PF5/AN5	V_{SS}

Note: * V_{CC} in the mask version; FWP in the F-ZTAT version (however, FWE in the writer mode)

Table 1.5 Pin Arrangement by Mode for SH7044 (QFP-112 Pin) (cont)

PinNo.	MCU	Writer mode
97	AV_{SS}	V_{SS}
98	PF6/AN6	V_{SS}
99	PF7/AN7	V_{SS}
100	AV_{CC}	V_{CC}
101	V_{SS}	V_{SS}
102	PE5/TIOC1B	NC
103	V_{CC}	V_{CC}
104	PE6/TIOC2A	NC
105	PE7/TIOC2B	NC
106	PE8/TIOC3A	NC
107	PE9/TIOC3B	NC
108	PE10/TIOC3C	NC
109	V_{SS}	V_{SS}
110	PE11/TIOC3D	NC
111	PE12/TIOC4A	NC
112	PE13/TIOC4B/$\overline{\text{MRES}}$	NC

Table 1.6 Pin Arrangement by Mode for SH7045 (QFP-144 Pin)

PinNo.	MCU	Writer mode
1	PA23/$\overline{\text{WRHH}}$	NC
2	PE14/TIOC4C/DACK0/$\overline{\text{AH}}$	NC
3	PA22/$\overline{\text{WRHL}}$	NC
4	PA21/$\overline{\text{CASHH}}$	NC
5	PE15/TIOC4D/DACK1/$\overline{\text{IRQOUT}}$	NC
6	V_{SS}	V_{SS}
7	PC0/A0	A0
8	PC1/A1	A1
9	PC2/A2	A2
10	PC3/A3	A3
11	PC4/A4	A4
12	V_{CC}	V_{CC}
13	PC5/A5	A5
14	V_{SS}	V_{SS}
15	PC6/A6	A6
16	PC7/A7	A7
17	PC8/A8	A8
18	PC9/A9	A9
19	PC10/A10	A10
20	PC11/A11	A11
21	PC12/A12	A12
22	PC13/A13	A13
23	PC14/A14	A14
24	PC15/A15	A15
25	PB0/A16	A16
26	V_{CC}	V_{CC}
27	PB1/A17	NC
28	V_{SS}	V_{SS}
29	PA20/$\overline{\text{CASHL}}$	NC
30	PA19/$\overline{\text{BACK}}$/DRAK1	NC
31	PB2/$\overline{\text{IRQ0}}$/$\overline{\text{POE0}}$/$\overline{\text{RAS}}$	NC
32	PB3/$\overline{\text{IRQ1}}$/$\overline{\text{POE1}}$/$\overline{\text{CASL}}$	NC
33	PA18/$\overline{\text{BREQ}}$/DRAK0	NC
34	PB4/$\overline{\text{IRQ2}}$/$\overline{\text{POE2}}$/$\overline{\text{CASH}}$	A17
35	V_{SS}	V_{SS}
36	PB5/$\overline{\text{IRQ3}}$/$\overline{\text{POE3}}$/RDWR	NC

Table 1.6 Pin Arrangement by Mode for SH7045 (QFP-144 Pin) (cont)

PinNo.	MCU	Writer mode
37	PB6/$\overline{\text{IRQ4}}$/A18/$\overline{\text{BACK}}$	NC
38	PB7/$\overline{\text{IRQ5}}$/A19/$\overline{\text{BREQ}}$	NC
39	PB8/$\overline{\text{IRQ6}}$/A20/$\overline{\text{WAIT}}$	NC
40	V_{CC}	V_{CC}
41	PB9/$\overline{\text{IRQ7}}$/A21/$\overline{\text{ADTRG}}$	NC
42	V_{SS}	V_{SS}
43	PA14/$\overline{\text{RD}}$	NC
44	$\overline{\text{WDTOVF}}$	NC
45	PD31/D31/$\overline{\text{ADTRG}}$	NC
46	PD30/D30/$\overline{\text{IRQOUT}}$	NC
47	PA13/$\overline{\text{WRH}}$	NC
48	PA12/$\overline{\text{WRL}}$	NC
49	PA11/$\overline{\text{CS1}}$	NC
50	PA10/$\overline{\text{CS0}}$	NC
51	PA9/TCLKD/$\overline{\text{IRQ3}}$	$\overline{\text{CE}}$
52	PA8/TCLKC/$\overline{\text{IRQ2}}$	$\overline{\text{OE}}$
53	PA7/TCLKB/$\overline{\text{CS3}}$	$\overline{\text{WE}}$
54	PA6/TCLKA/$\overline{\text{CS2}}$	NC
55	V_{SS}	V_{SS}
56	PD29/D29/$\overline{\text{CS3}}$	NC
57	PD28/D28/$\overline{\text{CS2}}$	NC
58	PD27/D27/DACK1	NC
59	PD26/D26/DACK0	NC
60	PD25/D25/$\overline{\text{DREQ1}}$	NC
61	V_{SS}	V_{SS}
62	PD24/D24/$\overline{\text{DREQ0}}$	NC
63	V_{CC}	V_{CC}
64	PD23/D23/$\overline{\text{IRQ7}}$	NC
65	PD22/D22/$\overline{\text{IRQ6}}$	NC
66	PD21/D21/$\overline{\text{IRQ5}}$	NC
67	PD20/D20/$\overline{\text{IRQ4}}$	NC
68	PD19/D19/$\overline{\text{IRQ3}}$	NC
69	PD18/D18/$\overline{\text{IRQ2}}$	NC
70	PD17/D17/$\overline{\text{IRQ1}}$	NC
71	V_{SS}	V_{SS}
72	PD16/D16/$\overline{\text{IRQ0}}$	NC

Table 1.6 Pin Arrangement by Mode for SH7045 (QFP-144 Pin) (cont)

PinNo.	MCU	Writer mode
73	PD15/D15	NC
74	PD14/D14	NC
75	PD13/D13	NC
76	PD12/D12	NC
77	V_{CC}	V_{CC}
78	PD11/D11	NC
79	V_{SS}	V_{SS}
80	PD10/D10	NC
81	PD9/D9	NC
82	PD8/D8	NC
83	PD7/D7	D7
84	PD6/D6	D6
85	V_{CC}	V_{CC}
86	PD5/D5	D5
87	V_{SS}	V_{SS}
88	PD4/D4	D4
89	PD3/D3	D3
90	PD2/D2	D2
91	PD1/D1	D1
92	PD0/D0	D0
93	V_{SS}	V_{SS}
94	XTAL	XTAL
95	MD3	MD3
96	EXTAL	EXTAL
97	MD2	MD2
98	NMI	V_{CC}
99	V_{CC} (FWP)*	FWE
100	PA16/$\overline{AH}$	NC
101	PA17/$\overline{WAIT}$	NC
102	MD1	MD1
103	MD0	MD0
104	$PLLV_{CC}$	$PLLV_{CC}$
105	PLLCAP	PLLCAP
106	$PLLV_{SS}$	$PLLV_{SS}$
107	PA15/CK	NC

Note: * V_{CC} in the mask version; FWP in the F-ZTAT version (however, FWE in the writer mode)

Table 1.6 Pin Arrangement by Mode for SH7045 (QFP-144 Pin) (cont)

PinNo.	MCU	Writer mode
108	$\overline{\text{RES}}$	$\overline{\text{RES}}$
109	PE0/TIOC0A/$\overline{\text{DREQ0}}$	NC
110	PE1/TIOC0B/DRAK0	NC
111	PE2/TIOC0C/$\overline{\text{DREQ1}}$	NC
112	V_{CC}	V_{CC}
113	PE3/TIOC0D/DRAK1	NC
114	PE4/TIOC1A	NC
115	PE5/TIOC1B	NC
116	PE6/TIOC2A	NC
117	V_{SS}	V_{SS}
118	PF0/AN0	V_{SS}
119	PF1/AN1	V_{SS}
120	PF2/AN2	V_{SS}
121	PF3/AN3	V_{SS}
122	PF4/AN4	V_{SS}
123	PF5/AN5	V_{SS}
124	AV_{SS}	V_{SS}
125	PF6/AN6	V_{SS}
126	PF7/AN7	V_{SS}
127	AVref	V_{CC}
128	AV_{CC}	V_{CC}
129	V_{SS}	V_{SS}
130	PA0/RXD0	NC
131	PA1/TXD0	V_{CC}
132	PA2/SCK0/$\overline{\text{DREQ0}}$/$\overline{\text{IRQ0}}$	V_{CC}
133	PA3/RXD1	NC
134	PA4/TXD1	NC
135	V_{CC}	V_{CC}
136	PA5/SCK1/$\overline{\text{DREQ1}}$/$\overline{\text{IRQ1}}$	V_{CC}
137	PE7/TIOC2B	NC
138	PE8/TIOC3A	NC
139	PE9/TIOC3B	NC
140	PE10/TIOC3C	NC
141	V_{SS}	V_{SS}
142	PE11/TIOC3D	NC
143	PE12/TIOC4A	NC
144	PE13/TIOC4B/$\overline{\text{MRES}}$	NC

1.3.3 Pin Functions

Table 1.7 lists the pin functions.

Table 1.7 Pin Functions

Classification	Symbol	I/O	Name	Function
Power supply	V_{CC}	I	Supply	Connects to power supply. Connect all V_{CC} pins to the system supply. No operation will occur if there are any open pins.
	V_{SS}	I	Ground	Connects to ground. Connect all V_{SS} pins to the system ground. No operation will occur if there are any open pins.
	V_{PP}	I	Program supply	Connects to the power supply (V_{CC}) during normal operation. When in PROM mode, apply 12.5 V.
Clock	PLLVCC	I	PLL supply	On-chip PLL oscillator supply.
	PLLVSS	I	PLL ground	On-chip PLL oscillator ground.
	PLLCAP	I	PLL capacitance	On-chip PLL oscillator external capacitance connection pin.
	EXTAL	I	External clock	Connect a crystal oscillator. Also, an external clock can be input to the EXTAL pin.
	XTAL	I	Crystal	Connect a crystal oscillator.
	CK	O	System clock	Supplies the system clock to peripheral devices.
System control	$\overline{\text{RES}}$	I	Power-on reset	Power-on reset when low
	$\overline{\text{MRES}}$	I	Manual reset	Manual reset when low
	$\overline{\text{WDTOVF}}$	O	Watchdog timer overflow	Overflow output signal from WDT
	$\overline{\text{BREQ}}$	I	Bus request	Goes low when external device requests bus right release
	$\overline{\text{BACK}}$	O	Bus request acknowledge	Indicates that bus right has been released to external device. The device that output the $\overline{\text{BREQ}}$ signal receives the $\overline{\text{BACK}}$ signal, notifying the device that it has obtained the bus right.

Table 1.7 Pin Functions (cont)

Classification	Symbol	I/O	Name	Function
Operating mode control	MD0–MD3	I	Mode set	Determines the operating mode. Do not change input value during operation.
	FWP	I	Flash memory write protect	Protects flash memory from being written or deleted.
Interrupts	NMI	I	Non-maskable interrupt	Non-maskable interrupt request pin. Enables selection of whether to accept on the rising or falling edge.
	$\overline{\text{IRQ0}}$–$\overline{\text{IRQ7}}$	I	Interrupt requests 0–7	Maskable interrupt request pins. Allows selection of level input and edge input.
	$\overline{\text{IRQOUT}}$	O	Interrupt request output	Indicates that interrupt cause has occurred. Enables notification of interrupt generation also during bus release.
Address bus	A0–A21	O	Address bus	Outputs addresses.
Data bus	D0–D15 (QFP-112) D0–D31 (QFP-144)	I/O	Data bus	16-bit (QFP-112 pin and TQFP-120 pin versions) or 32-bit (QFP-144 pin version) bidirectional data bus.
Bus control	$\overline{\text{CS0}}$–$\overline{\text{CS3}}$	O	Chip selects 0–3	Chip select signals for external memory or devices.
	$\overline{\text{RD}}$	O	Read	Indicates reading from an external device.
	$\overline{\text{WRH}}$	O	Upper write	Indicates writing the upper 8 bits (15–8) of external data.
	$\overline{\text{WRL}}$	O	Lower write	Indicates writing the lower 8 bits (7–0) of external data.
	$\overline{\text{WAIT}}$	I	Wait	Input causes insertion of wait cycles into the bus cycle during external space access.
	$\overline{\text{RAS}}$	O	Row address strobe	Timing signal for DRAM row address strobe.
	$\overline{\text{CASH}}$	O	Upper column address strobe	Timing signal for DRAM column address strobe. Output when the upper 8 bits of data are accessed.

Table 1.7 Pin Functions (cont)

Classification	Symbol	I/O	Name	Function
Bus control (cont)	$\overline{\text{CASL}}$	O	Lower column address strobe	Timing signal for DRAM column address strobe. Output when the lower 8 bits of data are accessed.
	RDWR	O	DRAM read/write	DRAM write strobe signal.
	$\overline{\text{AH}}$	O	Address hold	Address hold timing signal for devices using an address/data multiplex bus.
	$\overline{\text{WRHH}}$ (QFP-144)	O	HH write	Indicates the writing of bits 31 to 24 of external data.
	$\overline{\text{WRHL}}$ (QFP-144)	O	HL write	Indicates the writing of bits 23 to 16 of external data.
	$\overline{\text{CASHH}}$ (QFP-144)	O	HH column address strobe	Timing signal for DRAM column address strobe. Output when bits 31 to 24 of data are accessed.
	$\overline{\text{CASHL}}$ (QFP-144)	O	HL column address strobe	Timing signal for DRAM column address strobe. Output when bits 23 to 16 of data are accessed.
Bus control multifunction timer/pulse unit	TCLKA TCLKB TCLKC TCLKD	I	MTU timer clock input	Input pins for external clocks to the MTU counter.
	TIOC0A TIOC0B TIOC0C TIOC0D	I/O	MTU input capture/ output compare (channel 0)	Channel 0 input capture input/output compare output/PWM output pins.
	TIOC1A TIOC1B	I/O	MTU input capture/output compare (channel 1)	Channel 1 input capture input/output compare output/PWM output pins.
	TIOC2A TIOC2B	I/O	MTU input capture/output compare (channel 2)	Channel 2 input capture input/output compare output/PWM output pins.

Table 1.7 Pin Functions (cont)

Classification	Symbol	I/O	Name	Function
Bus control multifunction timer/pulse unit (cont)	TIOC3A TIOC3B TIOC3C TIOC3D	I/O	MTU input capture/output compare (channel 3)	Channel 3 input capture input/output compare output/PWM output pins.
	TIOC4A TIOC4B TIOC4C TIOC4D	I/O	MTU input capture/output compare (channel 4)	Channel 4 input capture input/output compare output/PWM output pins.
Direct memory access controller (DMAC)	$\overline{\text{DREQ0}}$–$\overline{\text{DREQ1}}$	I	DMA transfer request (channels 0, 1)	Input pin for external requests for DMA transfer.
	DRAK0–DRAK1	O	DREQ request acknowledgment (channels 0, 1)	Output the input sampling acknowledgment of external DMA transfer requests.
	DACK0–DACK1	O	DMA transfer strobe (channels 0, 1)	Output a strobe to the external I/O of external DMA transfer requests.
Serial communication interface (SCI)	TxD0–TxD1	O	Transmit data (channels 0, 1)	SCI0, SCI1 transmit data output pins. (TxD1 is used for data transfer during boot mode of F-ZTAT)
	RxD0–RxD1	I	Receive data (channels 0, 1)	SCI0, SCI1 receive data input pins. (RxD1 is used for data transfer during boot mode of F-ZTAT)
	SCK0–SCK1	I/O	Serial clock (channels 0, 1)	SCI0, SCI1 clock input/output pins.
A/D Converter	AV_{CC}	I	Analog supply	Analog supply; connected to V_{CC}.
	AV_{SS}	I	Analog ground	Analog supply; connected to V_{SS}.
	AVref (QFP-144 only)	I	Analog reference supply	Analog reference supply input pin. (Connected to AV_{CC} internally in QFP-112 and TQFP-120.)
	AN0–AN7	I	Analog input	Analog signal input pins.
	$\overline{\text{ADTRG}}$	I	A/D conversion trigger input	External trigger input for A/D conversion start.

Table 1.7 Pin Functions (cont)

Classification	Symbol	I/O	Name	Function
I/O ports	$\overline{POE0}$–$\overline{POE3}$	I	Port output enable	Input pin for port pin drive control when general use ports are established as output.
	PA0–PA15 (QFP-112) PA0–PA23 (QFP-144)	I/O	General purpose port	General purpose input/output port pins. Each bit can be designated for input/output.
	PB0–PB9	I/O	General purpose port	General purpose input/output port pins. Each bit can be designated for input/output.
	PC0–PC15	I/O	General purpose port	General purpose input/output port pins. Each bit can be designated for input/output.
	PD0–PD15 (QFP-112) PD0–PD31 (QFP-144)	I/O	General purpose port	General purpose input/output port pins. Each bit can be designated for input/output.
	PE0–PE15	I/O	General purpose port	General purpose input/output port pins. Each bit can be designated for input/output.
	PF0–PF7	I	General purpose port	General purpose input port pins.

Usage Notes

1. Unused input pins should be pulled up or pulled down.
2. The $\overline{WDTOVF}$ pin should not be pulled down in the SH7044/SH7045 F-ZTAT version. However, if it is necessary to pull this pin down, a resistance of 100 kΩ or higher should be used.

1.4 The F-ZTAT Version Onboard Programming

There are 2 modes on the F-ZTAT version: a mode that writes and overwrites programs using the special writer and a mode that writes and overwrites programs onboard the application system.

When rebooting after setting each mode pin and FWP pin during the reset condition, the microcomputer will transfer to one of the modes indicated in figure 1.6. In the user mode, data can be read from the flash memory but cannot be written or deleted. Use the boot mode and the user program mode to write to the flash memory or delete data.

In the boot mode, SCI1 (TXD1, RXD1) is used for data transfer. It is possible to automatically adjust the transfer bit rate to the transfer bit rate of the host.

Table 1.8 Pins during the Onboard Programming Mode

Notation	I/O	Function
FWP	Input	Hardware protected flash memory write/delete
MD1	Input	User programming mode/boot mode setting
MD2	Input	Clock mode (PLL) setting
MD3	Input	Clock mode (PLL) setting
TxD1	Output	Serial sent data output
RxD1	Input	Serial receive data input

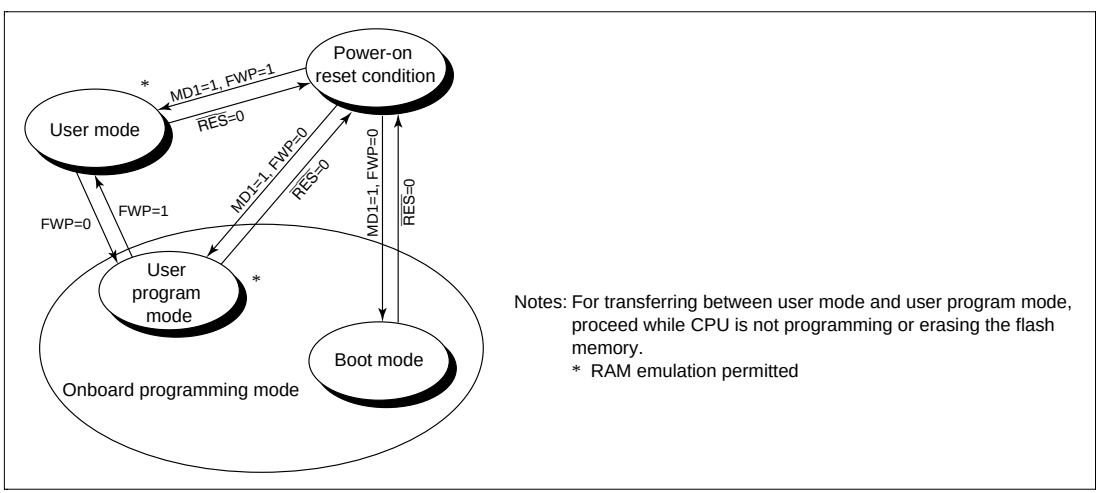


Figure 1.6 Condition Transfer for Flash Memory

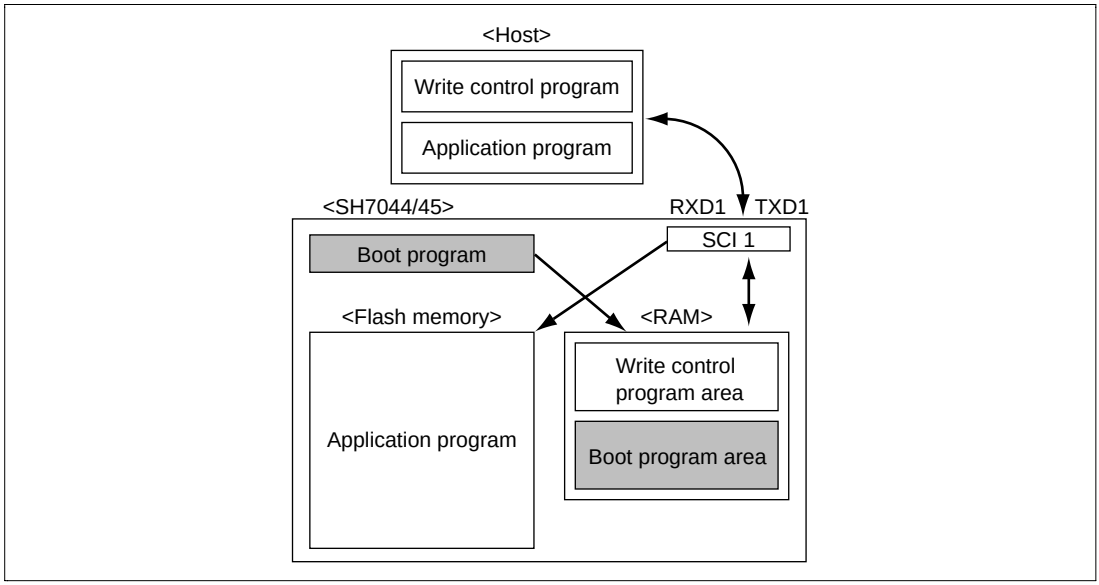


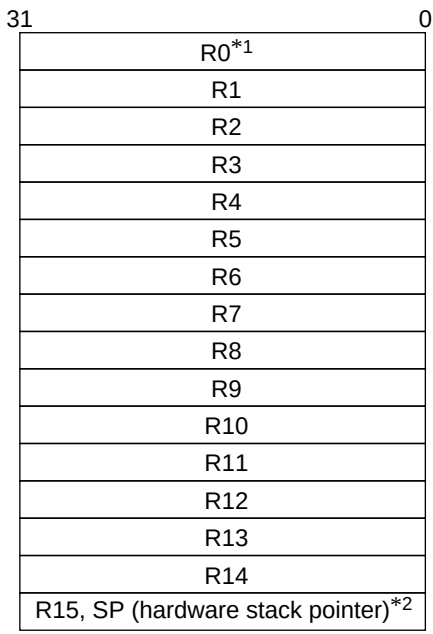
Figure. 1.7 Data Transfer during Boot Mode

2.1 Register Configuration

The register set consists of sixteen 32-bit general registers, three 32-bit control registers and four 32-bit system registers.

2.1.1 General Registers (Rn)

The sixteen 32-bit general registers (Rn) are numbered R0–R15. General registers are used for data processing and address calculation. R0 is also used as an index register. Several instructions have R0 fixed as their only usable register. R15 is used as the hardware stack pointer (SP). Saving and recovering the status register (SR) and program counter (PC) in exception processing is accomplished by referencing the stack using R15. Figure 2.1 shows the general registers.



Notes: *1 R0 functions as an index register in the indirect indexed register addressing mode and indirect indexed GBR addressing mode. In some instructions, R0 functions as a fixed source register or destination register.
*2 R15 functions as a hardware stack pointer (SP) during exception processing.

Figure 2.1 General Registers

2.1.2 Control Registers

The 32-bit control registers consist of the 32-bit status register (SR), global base register (GBR), and vector base register (VBR). The status register indicates processing states. The global base register functions as a base address for the indirect GBR addressing mode to transfer data to the registers of on-chip peripheral modules. The vector base register functions as the base address of the exception processing vector area (including interrupts). Figure 2.2 shows a control register.

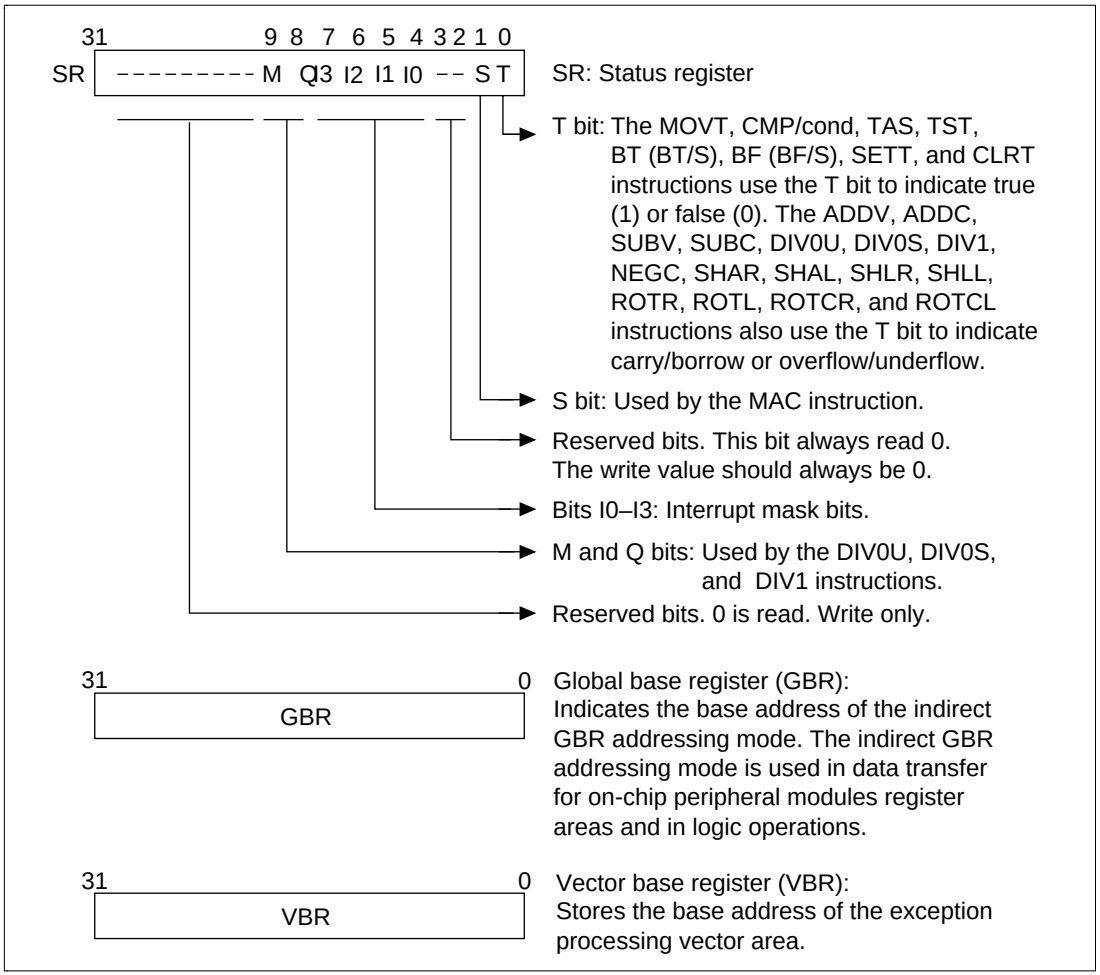


Figure 2.2 Control Registers

2.1.3 System Registers

System registers consist of four 32-bit registers: high and low multiply and accumulate registers (MACH and MACL), the procedure register (PR), and the program counter (PC). The multiply and accumulate registers store the results of multiply and accumulate operations. The procedure register stores the return address from the subroutine procedure. The program counter stores program addresses to control the flow of the processing. Figure 2.3 shows a system register.

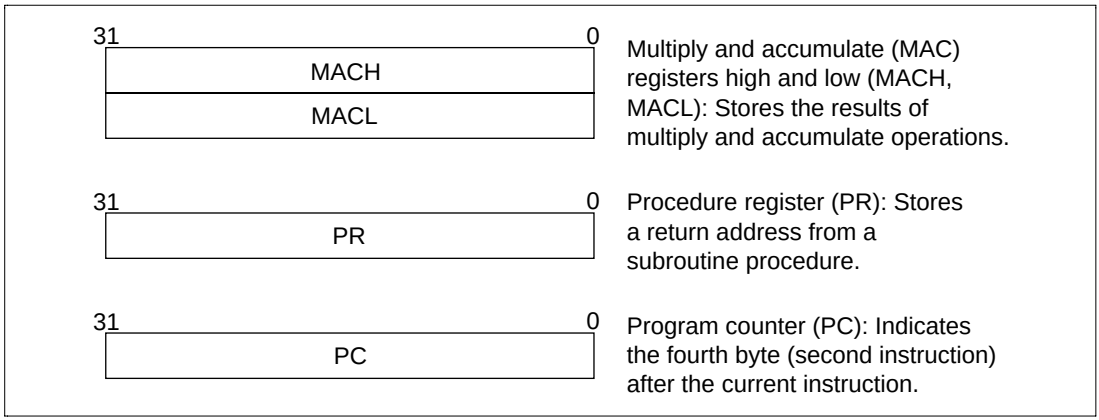


Figure 2.3 System Registers

2.1.4 Initial Values of Registers

Table 2.1 lists the values of the registers after reset.

Table 2.1 Initial Values of Registers

Classification	Register	Initial Value
General registers	R0–R14	Undefined
	R15 (SP)	Value of the stack pointer in the vector address table
Control registers	SR	Bits I3–I0 are 1111 (H'F), reserved bits are 0, and other bits are undefined
	GBR	Undefined
	VBR	H'00000000
System registers	MACH, MACL, PR	Undefined
	PC	Value of the program counter in the vector address table

2.2 Data Formats

2.2.1 Data Format in Registers

Register operands are always longwords (32 bits). When the memory operand is only a byte (8 bits) or a word (16 bits), it is sign-extended into a longword when loaded into a register (figure 2.4).

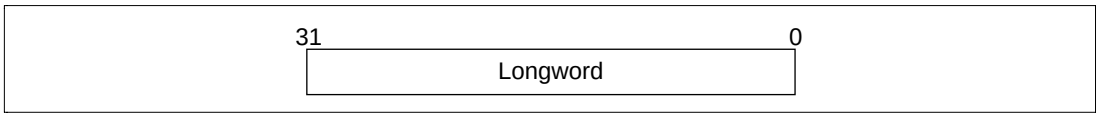


Figure 2.4 Longword Operand

2.2.2 Data Format in Memory

Memory data formats are classified into bytes, words, and longwords. Byte data can be accessed from any address, but an address error will occur if you try to access word data starting from an address other than 2n or longword data starting from an address other than 4n. In such cases, the data accessed cannot be guaranteed. The hardware stack area, referred to by the hardware stack pointer (SP, R15), uses only longword data starting from address 4n because this area holds the program counter and status register (figure 2.5).

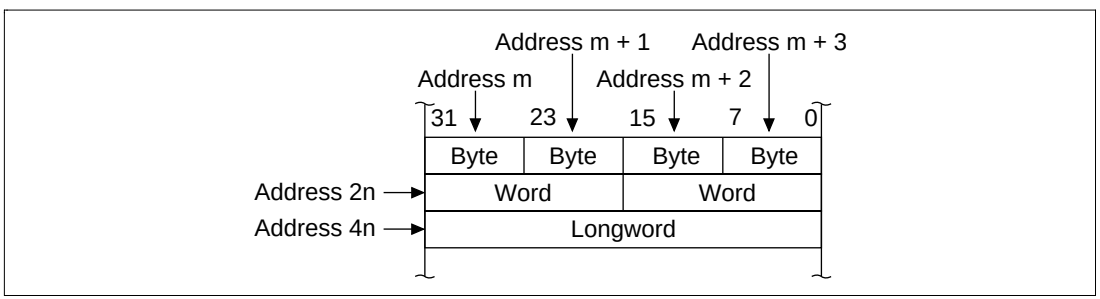


Figure 2.5 Byte, Word, and Longword Alignment

2.2.3 Immediate Data Format

Byte (8-bit) immediate data resides in an instruction code. Immediate data accessed by the MOV, ADD, and CMP/EQ instructions is sign-extended and handled in registers as longword data. Immediate data accessed by the TST, AND, OR, and XOR instructions is zero-extended and handled as longword data. Consequently, AND instructions with immediate data always clear the upper 24-bits of the destination register.

Word or longword immediate data is not located in the instruction code, but instead is stored in a memory table. An immediate data transfer instruction (MOV) accesses the memory table using the PC relative addressing mode with displacement.

2.3 Instruction Features

2.3.1 RISC-Type Instruction Set

All instructions are RISC type. This section details their functions.

16-Bit Fixed Length: All instructions are 16 bits long, increasing program code efficiency.

One Instruction per Cycle: The microprocessor can execute basic instructions in one cycle using the pipeline system. Instructions are executed in 35 ns at 28.7 MHz.

Data Length: Longword is the standard data length for all operations. Memory can be accessed in bytes, words, or longwords. Byte or word data accessed from memory is sign-extended and handled as longword data. Immediate data is sign-extended for arithmetic operations or zero-extended for logic operations. It also is handled as longword data (table 2.2).

Table 2.2 Sign Extension of Word Data

SH7040 Series CPU		Description	Example of Conventional CPU	
MOV.W	@(disp,PC),R1	Data is sign-extended to 32 bits, and R1 becomes H'00001234. It is next operated upon by an ADD instruction.	ADD.W	#H'1234,R0
ADD	R1,R0			
	.....			
.DATA.W	H'1234			

Note: @(disp, PC) accesses the immediate data.

Load-Store Architecture: Basic operations are executed between registers. For operations that involve memory access, data is loaded to the registers and executed (load-store architecture). Instructions such as AND that manipulate bits, however, are executed directly in memory.

Delayed Branch Instructions: Unconditional branch instructions are delayed. Executing the instruction that follows the branch instruction and then branching reduces pipeline disruption during branching (table 2.3). There are two types of conditional branch instructions: delayed branch instructions and ordinary branch instructions.

Table 2.3 Delayed Branch Instructions

SH7040 Series CPU		Description	Example of Conventional CPU	
BRA	TRGET	Executes an ADD before branching to TRGET	ADD.W	R1,R0
ADD	R1,R0		BRA	TRGET

Multiplication/Accumulation Operation: 16-bit × 16-bit → 32-bit multiplication operations are executed in one to two cycles. 16-bit × 16-bit + 64-bit → 64-bit multiplication/accumulation operations are executed in two to three cycles. 32-bit × 32-bit → 64-bit and 32-bit × 32-bit + 64-bit → 64-bit multiplication/accumulation operations are executed in two to four cycles.

T Bit: The T bit in the status register changes according to the result of the comparison, and in turn is the condition (true/false) that determines if the program will branch. The number of instructions that change the T bit is kept to a minimum to improve the processing speed (table 2.4).

Table 2.4 T Bit

SH7040 Series CPU		Description	Example of Conventional CPU	
CMP/GE	R1,R0	T bit is set when R0 ≥ R1. The program branches to TRGET0 when R0 ≥ R1 and to TRGET1 when R0 < R1.	CMP.W	R1,R0
BT	TRGET0		BGE	TRGET0
BF	TRGET1		BLT	TRGET1
ADD	#1,R0	T bit is not changed by ADD. T bit is set when R0 = 0. The program branches if R0 = 0.	SUB.W	#1,R0
CMP/EQ	#0,R0		BEQ	TRGET
BT	TRGET			

Immediate Data: Byte (8-bit) immediate data resides in instruction code. Word or longword immediate data is not input via instruction codes but is stored in a memory table. An immediate data transfer instruction (MOV) accesses the memory table using the PC relative addressing mode with displacement (table 2.5).

Table 2.5 Immediate Data Accessing

Classification	SH7040 Series CPU		Example of Conventional CPU	
8-bit immediate	MOV	#H'12,R0	MOV.B	#H'12,R0
16-bit immediate	MOV.W	@(disp,PC),R0	MOV.W	#H'1234,R0
		.....		
	.DATA.W	H'1234		
32-bit immediate	MOV.L	@(disp,PC),R0	MOV.L	#H'12345678,R0
		.....		
	.DATA.L	H'12345678		

Note: @(disp, PC) accesses the immediate data.

Absolute Address: When data is accessed by absolute address, the value already in the absolute address is placed in the memory table. Loading the immediate data when the instruction is executed transfers that value to the register and the data is accessed in the indirect register addressing mode (table 2.6).

Table 2.6 Absolute Address Accessing

Classification	SH7040 Series CPU		Example of Conventional CPU	
Absolute address	MOV.L	@(disp,PC),R1	MOV.B	@H'12345678,R0
	MOV.B	@R1,R0		
		.....		
	.DATA.L	H'12345678		

Note: @(disp,PC) accesses the immediate data.

16-Bit/32-Bit Displacement: When data is accessed by 16-bit or 32-bit displacement, the pre-existing displacement value is placed in the memory table. Loading the immediate data when the instruction is executed transfers that value to the register and the data is accessed in the indirect indexed register addressing mode (table 2.7).

Table 2.7 Displacement Accessing

Classification	SH7040 Series CPU		Example of Conventional CPU	
16-bit displacement	MOV.W	@(disp,PC),R0	MOV.W	@(H'1234,R1),R2
	MOV.W	@(R0,R1),R2		
		.....		
	.DATA.W	H'1234		

Note: @(disp,PC) accesses the immediate data.

2.3.2 Addressing Modes

Table 2.8 describes addressing modes and effective address calculation.

Table 2.8 Addressing Modes and Effective Addresses

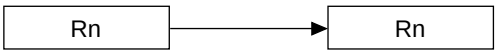
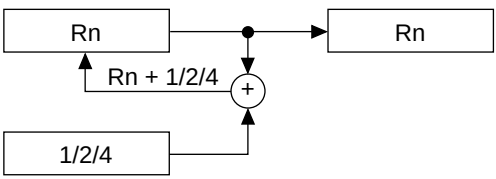
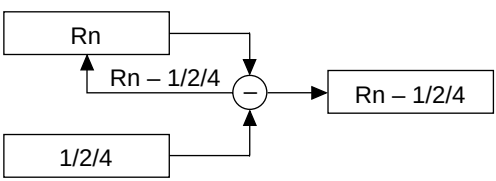
Addressing Mode	Instruction Format	Effective Addresses Calculation	Equation
Direct register addressing	Rn	The effective address is register Rn. (The operand is the contents of register Rn.)	—
Indirect register addressing	@Rn	The effective address is the content of register Rn. 	Rn
Post-increment indirect register addressing	@Rn+	The effective address is the content of register Rn. A constant is added to the content of Rn after the instruction is executed. 1 is added for a byte operation, 2 for a word operation, and 4 for a longword operation. 	Rn (After the instruction executes) Byte: Rn + 1 → Rn Word: Rn + 2 → Rn Longword: Rn + 4 → Rn
Pre-decrement indirect register addressing	@–Rn	The effective address is the value obtained by subtracting a constant from Rn. 1 is subtracted for a byte operation, 2 for a word operation, and 4 for a longword operation. 	Byte: Rn – 1 → Rn Word: Rn – 2 → Rn Longword: Rn – 4 → Rn (Instruction executed with Rn after calculation)

Table 2.8 Addressing Modes and Effective Addresses (cont)

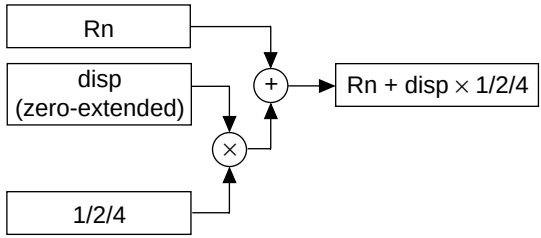
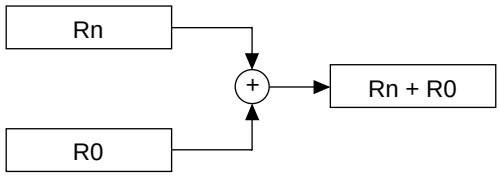
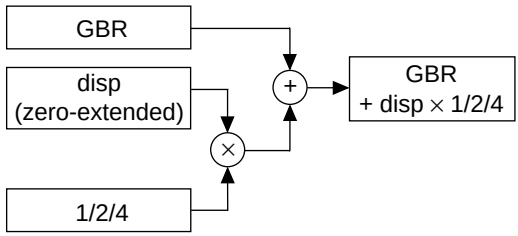
Addressing Mode	Instruction Format	Effective Addresses Calculation	Equation
Indirect register addressing with displacement	@(disp:4, Rn)	The effective address is Rn plus a 4-bit displacement (disp). The value of disp is zero-extended, and remains the same for a byte operation, is doubled for a word operation, and is quadrupled for a longword operation. 	Byte: Rn + disp Word: Rn + disp × 2 Longword: Rn + disp × 4
Indirect indexed register addressing	@(R0, Rn)	The effective address is the Rn value plus R0. 	Rn + R0
Indirect GBR addressing with displacement	@(disp:8, GBR)	The effective address is the GBR value plus an 8-bit displacement (disp). The value of disp is zero-extended, and remains the same for a byte operation, is doubled for a word operation, and is quadrupled for a longword operation. 	Byte: GBR + disp Word: GBR + disp × 2 Longword: GBR + disp × 4

Table 2.8 Addressing Modes and Effective Addresses (cont)

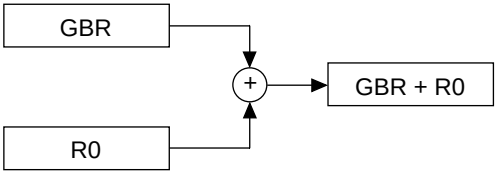
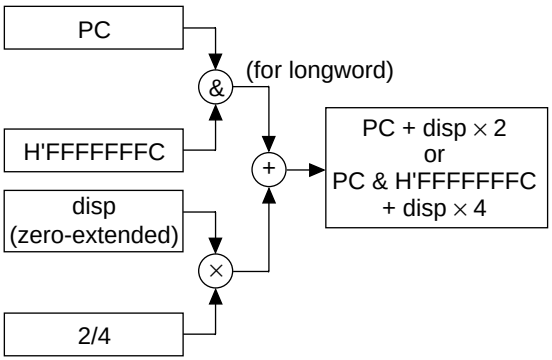
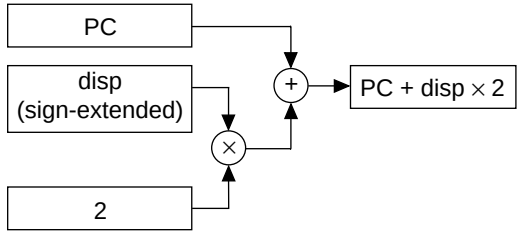
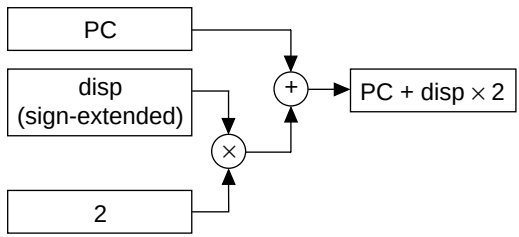
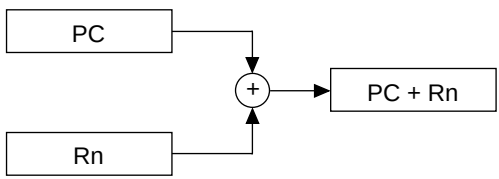
Addressing Mode	Instruction Format	Effective Addresses Calculation	Equation
Indirect indexed GBR addressing	`@(R0, GBR)`	The effective address is the GBR value plus the R0. 	GBR + R0
PC relative addressing with displacement	`@(disp:8, PC)`	The effective address is the PC value plus an 8-bit displacement (disp). The value of disp is zero-extended, and is doubled for a word operation, and quadrupled for a longword operation. For a longword operation, the lowest two bits of the PC value are masked. 	Word: PC + disp × 2 Longword: PC & H'FFFFFFFC + disp × 4

Table 2.8 Addressing Modes and Effective Addresses (cont)

Addressing Mode	Instruction Format	Effective Addresses Calculation	Equation
PC relative addressing	`disp:8`	The effective address is the PC value sign-extended with an 8-bit displacement (disp), doubled, and added to the PC value. 	PC + disp × 2
	`disp:12`	The effective address is the PC value sign-extended with a 12-bit displacement (disp), doubled, and added to the PC value. 	PC + disp × 2
	`Rn`	The effective address is the register PC value plus Rn. 	PC + Rn
Immediate addressing	`#imm:8`	The 8-bit immediate data (imm) for the TST, AND, OR, and XOR instructions are zero-extended.	—
	`#imm:8`	The 8-bit immediate data (imm) for the MOV, ADD, and CMP/EQ instructions are sign-extended.	—
	`#imm:8`	The 8-bit immediate data (imm) for the TRAPA instruction is zero-extended and is quadrupled.	—

2.3.3 Instruction Format

Table 2.9 lists the instruction formats for the source operand and the destination operand. The meaning of the operand depends on the instruction code. The symbols are used as follows:

- xxxx: Instruction code
- mmmm: Source register
- nnnn: Destination register
- iiii: Immediate data
- dddd: Displacement

Table 2.9 Instruction Formats

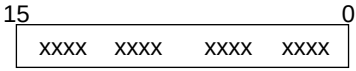
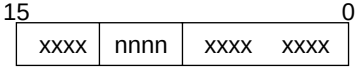
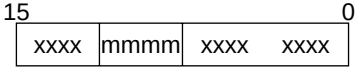
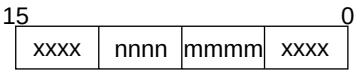
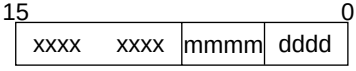
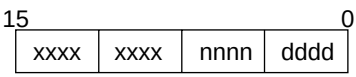
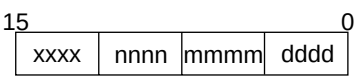
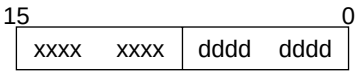
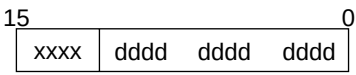
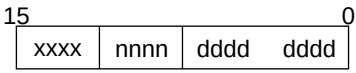
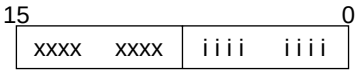
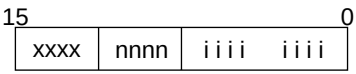
Instruction Formats	Source Operand	Destination Operand	Example
0 format 	—	—	NOP
n format	—	nnnn: Direct register	MOVT Rn
	Control register or system register	nnnn: Direct register	STS MACH,Rn
	Control register or system register	nnnn: Indirect pre-decrement register	STC.L SR,@-Rn
m format	mmmm: Direct register	Control register or system register	LDC Rm,SR
	mmmm: Indirect post-increment register	Control register or system register	LDC.L @Rm+,SR
	mmmm: Direct register	—	JMP @Rm
	mmmm: PC relative using Rm	—	BRAF Rm

Table 2.9 Instruction Formats (cont)

Instruction Formats	Source Operand	Destination Operand	Example
nm format 	mmmm: Direct register	nnnn: Direct register	ADD Rm,Rn
	mmmm: Direct register	nnnn: Indirect register	MOV.L Rm,@Rn
	mmmm: Indirect post-increment register (multiply/ accumulate) nnnn*: Indirect post-increment register (multiply/ accumulate)	MACH, MACL	MAC.W @Rm+,@Rn+
	mmmm: Indirect post-increment register	nnnn: Direct register	MOV.L @Rm+,Rn
	mmmm: Direct register	nnnn: Indirect pre-decrement register	MOV.L Rm,@-Rn
	mmmm: Direct register	nnnn: Indirect indexed register	MOV.L Rm,@(R0,Rn)
md format 	mmmmdddd: indirect register with displacement	R0 (Direct register)	MOV.B @(disp,Rm),R0
nd4 format 	R0 (Direct register)	nnnndddd: Indirect register with displacement	MOV.B R0,@(disp,Rn)
nmd format 	mmmm: Direct register	nnnndddd: Indirect register with displacement	MOV.L Rm,@(disp,Rn)
	mmmmdddd: Indirect register with displacement	nnnn: Direct register	MOV.L @(disp,Rm),Rn

Note: * In multiply/accumulate instructions, nnnn is the source register.

Table 2.9 Instruction Formats (cont)

Instruction Formats	Source Operand	Destination Operand	Example
d format 	dddddddd: Indirect GBR with displacement	R0 (Direct register)	`MOV.L @(disp,GBR),R0`
	R0(Direct register)	dddddddd: Indirect GBR with displacement	`MOV.L R0,@(disp,GBR)`
	dddddddd: PC relative with displacement	R0 (Direct register)	`MOVA @(disp,PC),R0`
	dddddddd: PC relative	—	`BF label`
d12 format 	ddddddddddddd: PC relative	—	`BRA label` `(label = disp + PC)`
nd8 format 	dddddddd: PC relative with displacement	nnnn: Direct register	`MOV.L @(disp,PC),Rn`
i format 	iiiiiiii: Immediate	Indirect indexed GBR	`AND.B #imm,@(R0,GBR)`
	iiiiiiii: Immediate	R0 (Direct register)	`AND #imm,R0`
	iiiiiiii: Immediate	—	`TRAPA #imm`
ni format 	iiiiiiii: Immediate	nnnn: Direct register	`ADD #imm,Rn`

2.4 Instruction Set by Classification

Table 2.10 Classification of Instructions

Classification	Types	Operation Code	Function	No. of Instructions
Data transfer	5	MOV	Data transfer, immediate data transfer, peripheral module data transfer, structure data transfer	39
		MOVA	Effective address transfer	
		MOVT	T bit transfer	
		SWAP	Swap of upper and lower bytes	
		XTRCT	Extraction of the middle of registers connected	
Arithmetic operations	21	ADD	Binary addition	33
		ADDC	Binary addition with carry	
		ADDV	Binary addition with overflow check	
		CMP/cond	Comparison	
		DIV1	Division	
		DIV0S	Initialization of signed division	
		DIV0U	Initialization of unsigned division	
		DMULS	Signed double-length multiplication	
		DMULU	Unsigned double-length multiplication	
		DT	Decrement and test	
		EXTS	Sign extension	
		EXTU	Zero extension	
		MAC	Multiply/accumulate, double-length multiply/accumulate operation	
		MUL	Double-length multiply operation	
		MULS	Signed multiplication	
		MULU	Unsigned multiplication	
		NEG	Negation	
		NEGC	Negation with borrow	
		SUB	Binary subtraction	
		SUBC	Binary subtraction with borrow	
		SUBV	Binary subtraction with underflow	

Table 2.10 Classification of Instructions (cont)

Classification	Types	Operation Code	Function	No. of Instructions
Logic operations	6	AND	Logical AND	14
		NOT	Bit inversion	
		OR	Logical OR	
		TAS	Memory test and bit set	
		TST	Logical AND and T bit set	
		XOR	Exclusive OR	
Shift	10	ROTL	One-bit left rotation	14
		ROTR	One-bit right rotation	
		ROTCL	One-bit left rotation with T bit	
		ROTCR	One-bit right rotation with T bit	
		SHAL	One-bit arithmetic left shift	
		SHAR	One-bit arithmetic right shift	
		SHLL	One-bit logical left shift	
		SHLLn	n-bit logical left shift	
		SHLR	One-bit logical right shift	
		SHLRn	n-bit logical right shift	
Branch	9	BF	Conditional branch, conditional branch with delay (Branch when T = 0)	11
		BT	Conditional branch, conditional branch with delay (Branch when T = 1)	
		BRA	Unconditional branch	
		BRAF	Unconditional branch	
		BSR	Branch to subroutine procedure	
		BSRF	Branch to subroutine procedure	
		JMP	Unconditional branch	
		JSR	Branch to subroutine procedure	
		RTS	Return from subroutine procedure	

Table 2.10 Classification of Instructions (cont)

Classification	Types	Operation Code	Function	No. of Instructions
System control	11	CLRT	T bit clear	31
		CLRMAC	MAC register clear	
		LDC	Load to control register	
		LDS	Load to system register	
		NOP	No operation	
		RTE	Return from exception processing	
		SETT	T bit set	
		SLEEP	Shift into power-down mode	
		STC	Storing control register data	
		STS	Storing system register data	
		TRAPA	Trap exception handling	
Total:	62			142

Table 2.11 shows the format used in tables 2.12 to 2.17, which list instruction codes, operation, and execution states in order by classification.

Table 2.11 Instruction Code Format

Item	Format	Explanation
Instruction	`OP.Sz SRC,DEST`	OP: Operation code Sz: Size (B: byte, W: word, or L: longword) SRC: Source DEST: Destination Rm: Source register Rn: Destination register imm: Immediate data disp: Displacement*1
Instruction code	MSB ↔ LSB	mmmm: Source register nnnn: Destination register 0000: R0 0001: R1 . . . 1111: R15 iiii: Immediate data dddd: Displacement
Operation	→, ←	Direction of transfer
	(xx)	Memory operand
	M/Q/T	Flag bits in the SR
	&	Logical AND of each bit
	|	Logical OR of each bit
	^	Exclusive OR of each bit
	~	Logical NOT of each bit
	<<n	n-bit left shift
	>>n	n-bit right shift
Execution cycles	—	Value when no wait states are inserted*2
T bit	—	Value of T bit after instruction is executed. An em-dash (—) in the column means no change.

Notes: *1 Depending on the operand size, displacement is scaled ×1, ×2, or ×4. For details, see the *SH-1/SH-2/SH-DSP Programming Manual*.

*2 Instruction execution cycles: The execution cycles shown in the table are minimums. The actual number of cycles may be increased when (1) contention occurs between instruction fetches and data access, or (2) when the destination register of the load instruction (memory → register) and the register used by the next instruction are the same.

Table 2.12 Data Transfer Instructions

Instruction	Instruction Code	Operation	Execution Cycles	T Bit
MOV #imm,Rn	1110nnnniiiiiiii	#imm → Sign extension → Rn	1	—
MOV.W @(disp,PC),Rn	1001nnnndddddddd	(disp × 2 + PC) → Sign extension → Rn	1	—
MOV.L @(disp,PC),Rn	1101nnnndddddddd	(disp × 4 + PC) → Rn	1	—
MOV Rm,Rn	0110nnnnmmmm0011	Rm → Rn	1	—
MOV.B Rm,@Rn	0010nnnnmmmm0000	Rm → (Rn)	1	—
MOV.W Rm,@Rn	0010nnnnmmmm0001	Rm → (Rn)	1	—
MOV.L Rm,@Rn	0010nnnnmmmm0010	Rm → (Rn)	1	—
MOV.B @Rm,Rn	0110nnnnmmmm0000	(Rm) → Sign extension → Rn	1	—
MOV.W @Rm,Rn	0110nnnnmmmm0001	(Rm) → Sign extension → Rn	1	—
MOV.L @Rm,Rn	0110nnnnmmmm0010	(Rm) → Rn	1	—
MOV.B Rm,@–Rn	0010nnnnmmmm0100	Rn–1 → Rn, Rm → (Rn)	1	—
MOV.W Rm,@–Rn	0010nnnnmmmm0101	Rn–2 → Rn, Rm → (Rn)	1	—
MOV.L Rm,@–Rn	0010nnnnmmmm0110	Rn–4 → Rn, Rm → (Rn)	1	—
MOV.B @Rm+,Rn	0110nnnnmmmm0100	(Rm) → Sign extension → Rn,Rm + 1 → Rm	1	—
MOV.W @Rm+,Rn	0110nnnnmmmm0101	(Rm) → Sign extension → Rn,Rm + 2 → Rm	1	—
MOV.L @Rm+,Rn	0110nnnnmmmm0110	(Rm) → Rn,Rm + 4 → Rm	1	—
MOV.B R0,@(disp,Rn)	10000000nnnndddd	R0 → (disp + Rn)	1	—
MOV.W R0,@(disp,Rn)	10000001nnnndddd	R0 → (disp × 2 + Rn)	1	—
MOV.L Rm,@(disp,Rn)	0001nnnnmmmmdddd	Rm → (disp × 4 + Rn)	1	—
MOV.B @(disp,Rm),R0	10000100mmmmdddd	(disp + Rm) → Sign extension → R0	1	—
MOV.W @(disp,Rm),R0	10000101mmmmdddd	(disp × 2 + Rm) → Sign extension → R0	1	—
MOV.L @(disp,Rm),Rn	0101nnnnmmmmdddd	(disp × 4 + Rm) → Rn	1	—
MOV.B Rm,@(R0,Rn)	0000nnnnmmmm0100	Rm → (R0 + Rn)	1	—

Table 2.12 Data Transfer Instructions (cont)

Instruction		Instruction Code	Operation	Execution Cycles	T Bit
MOV.W	Rm,@(R0,Rn)	0000nnnnmmmm0101	Rm → (R0 + Rn)	1	—
MOV.L	Rm,@(R0,Rn)	0000nnnnmmmm0110	Rm → (R0 + Rn)	1	—
MOV.B	@(R0,Rm),Rn	0000nnnnmmmm1100	(R0 + Rm) → Sign extension → Rn	1	—
MOV.W	@(R0,Rm),Rn	0000nnnnmmmm1101	(R0 + Rm) → Sign extension → Rn	1	—
MOV.L	@(R0,Rm),Rn	0000nnnnmmmm1110	(R0 + Rm) → Rn	1	—
MOV.B	R0,@(disp,GBR)	11000000dddddddd	R0 → (disp + GBR)	1	—
MOV.W	R0,@(disp,GBR)	11000001dddddddd	R0 → (disp × 2 + GBR)	1	—
MOV.L	R0,@(disp,GBR)	11000010dddddddd	R0 → (disp × 4 + GBR)	1	—
MOV.B	@(disp,GBR),R0	11000100dddddddd	(disp + GBR) → Sign extension → R0	1	—
MOV.W	@(disp,GBR),R0	11000101dddddddd	(disp × 2 + GBR) → Sign extension → R0	1	—
MOV.L	@(disp,GBR),R0	11000110dddddddd	(disp × 4 + GBR) → R0	1	—
MOVA	@(disp,PC),R0	11000111dddddddd	disp × 4 + PC → R0	1	—
MOVT	Rn	0000nnnn00101001	T → Rn	1	—
SWAP.B	Rm,Rn	0110nnnnmmmm1000	Rm → Swap the bottom two bytes → Rn	1	—
SWAP.W	Rm,Rn	0110nnnnmmmm1001	Rm → Swap two consecutive words → Rn	1	—
XTRCT	Rm,Rn	0010nnnnmmmm1101	Rm: Middle 32 bits of Rn → Rn	1	—

Table 2.13 Arithmetic Operation Instructions

Instruction		Instruction Code	Operation	Execution Cycles	T Bit
ADD	Rm,Rn	0011nnnnmmmm1100	Rn + Rm → Rn	1	—
ADD	#imm,Rn	0111nnnniiiiiiii	Rn + imm → Rn	1	—
ADDC	Rm,Rn	0011nnnnmmmm1110	Rn + Rm + T → Rn, Carry → T	1	Carry
ADDV	Rm,Rn	0011nnnnmmmm1111	Rn + Rm → Rn, Overflow → T	1	Overflow
CMP/EQ	#imm,R0	10001000iiiiiiii	If R0 = imm, 1 → T	1	Comparison result
CMP/EQ	Rm,Rn	0011nnnnmmmm0000	If Rn = Rm, 1 → T	1	Comparison result
CMP/HS	Rm,Rn	0011nnnnmmmm0010	If Rn≥Rm with unsigned data, 1 → T	1	Comparison result
CMP/GE	Rm,Rn	0011nnnnmmmm0011	If Rn ≥ Rm with signed data, 1 → T	1	Comparison result
CMP/HI	Rm,Rn	0011nnnnmmmm0110	If Rn > Rm with unsigned data, 1 → T	1	Comparison result
CMP/GT	Rm,Rn	0011nnnnmmmm0111	If Rn > Rm with signed data, 1 → T	1	Comparison result
CMP/PL	Rn	0100nnnn00010101	If Rn > 0, 1 → T	1	Comparison result
CMP/PZ	Rn	0100nnnn00010001	If Rn ≥ 0, 1 → T	1	Comparison result
CMP/STR	Rm,Rn	0010nnnnmmmm1100	If Rn and Rm have an equivalent byte, 1 → T	1	Comparison result
DIV1	Rm,Rn	0011nnnnmmmm0100	Single-step division (Rn/Rm)	1	Calculation result
DIV0S	Rm,Rn	0010nnnnmmmm0111	MSB of Rn → Q, MSB of Rm → M, M ^ Q → T	1	Calculation result
DIV0U		0000000000011001	0 → M/Q/T	1	0

Table 2.13 Arithmetic Operation Instructions (cont)

Instruction		Instruction Code	Operation	Execution Cycles	T Bit
DMULS.L	Rm,Rn	0011nnnnmmmm1101	Signed operation of Rn × Rm → MACH, MACL 32 × 32 → 64 bit	2 to 4*	—
DMULU.L	Rm,Rn	0011nnnnmmmm0101	Unsigned operation of Rn × Rm → MACH, MACL 32 × 32 → 64 bit	2 to 4*	—
DT	Rn	0100nnnn00010000	Rn – 1 → Rn, when Rn is 0, 1 → T. When Rn is nonzero, 0 → T	1	Comparison result
EXTS.B	Rm,Rn	0110nnnnmmmm1110	A byte in Rm is sign-extended → Rn	1	—
EXTS.W	Rm,Rn	0110nnnnmmmm1111	A word in Rm is sign-extended → Rn	1	—
EXTU.B	Rm,Rn	0110nnnnmmmm1100	A byte in Rm is zero-extended → Rn	1	—
EXTU.W	Rm,Rn	0110nnnnmmmm1101	A word in Rm is zero-extended → Rn	1	—
MAC.L	@Rm+,@Rn+	0000nnnnmmmm1111	Signed operation of (Rn) × (Rm) + MAC → MAC 32 × 32 → 64 bit	3/(2 to 4)*	—
MAC.W	@Rm+,@Rn+	0100nnnnmmmm1111	Signed operation of (Rn) × (Rm) + MAC → MAC 16 × 16 + 64 → 64 bit	3/(2)*	—
MUL.L	Rm,Rn	0000nnnnmmmm0111	Rn × Rm → MACL, 32 × 32 → 32 bit	2 to 4*	—
MULS.W	Rm,Rn	0010nnnnmmmm1111	Signed operation of Rn × Rm → MAC 16 × 16 → 32 bit	1 to 3*	—
MULU.W	Rm,Rn	0010nnnnmmmm1110	Unsigned operation of Rn × Rm → MAC 16 × 16 → 32 bit	1 to 3*	—
NEG	Rm,Rn	0110nnnnmmmm1011	0–Rm → Rn	1	—
NEGC	Rm,Rn	0110nnnnmmmm1010	0–Rm–T → Rn, Borrow → T	1	Borrow

Table 2.13 Arithmetic Operation Instructions (cont)

Instruction		Instruction Code	Operation	Execution Cycles	T Bit
SUB	Rm,Rn	0011nnnnmmmm1000	Rn–Rm → Rn	1	—
SUBC	Rm,Rn	0011nnnnmmmm1010	Rn–Rm–T → Rn, Borrow → T	1	Borrow
SUBV	Rm,Rn	0011nnnnmmmm1011	Rn–Rm → Rn, Underflow → T	1	Overflow

Note: * The normal minimum number of execution cycles. (The number in parentheses is the number of cycles when there is contention with following instructions.)

Table 2.14 Logic Operation Instructions

Instruction		Instruction Code	Operation	Execution Cycles	T Bit
AND	Rm,Rn	0010nnnnmmmm1001	Rn & Rm → Rn	1	—
AND	#imm,R0	11001001iiiiiiii	R0 & imm → R0	1	—
AND.B	#imm,@(R0,GBR)	11001101iiiiiiii	(R0 + GBR) & imm → (R0 + GBR)	3	—
NOT	Rm,Rn	0110nnnnmmmm0111	~Rm → Rn	1	—
OR	Rm,Rn	0010nnnnmmmm1011	Rn | Rm → Rn	1	—
OR	#imm,R0	11001011iiiiiiii	R0 | imm → R0	1	—
OR.B	#imm,@(R0,GBR)	11001111iiiiiiii	(R0 + GBR) | imm → (R0 + GBR)	3	—
TAS.B	@Rn	0100nnnn00011011	If (Rn) is 0, 1 → T; 1 → MSB of (Rn)*	4	Test result
TST	Rm,Rn	0010nnnnmmmm1000	Rn & Rm; if the result is 0, 1 → T	1	Test result
TST	#imm,R0	11001000iiiiiiii	R0 & imm; if the result is 0, 1 → T	1	Test result
TST.B	#imm,@(R0,GBR)	11001100iiiiiiii	(R0 + GBR) & imm; if the result is 0, 1 → T	3	Test result
XOR	Rm,Rn	0010nnnnmmmm1010	Rn ^ Rm → Rn	1	—
XOR	#imm,R0	11001010iiiiiiii	R0 ^ imm → R0	1	—
XOR.B	#imm,@(R0,GBR)	11001110iiiiiiii	(R0 + GBR) ^ imm → (R0 + GBR)	3	—

Note: * The on-chip DMAC/DTC bus cycles are not inserted between the read and write cycles of TAS instruction execution. However, bus release due to $\overline{\text{BREQ}}$ is carried out.

Table 2.15 Shift Instructions

Instruction		Instruction Code	Operation	Execution Cycles	T Bit
ROTL	Rn	0100nnnn00000100	T ← Rn ← MSB	1	MSB
ROTR	Rn	0100nnnn00000101	LSB → Rn → T	1	LSB
ROTCL	Rn	0100nnnn00100100	T ← Rn ← T	1	MSB
ROTCR	Rn	0100nnnn00100101	T → Rn → T	1	LSB
SHAL	Rn	0100nnnn00100000	T ← Rn ← 0	1	MSB
SHAR	Rn	0100nnnn00100001	MSB → Rn → T	1	LSB
SHLL	Rn	0100nnnn00000000	T ← Rn ← 0	1	MSB
SHLR	Rn	0100nnnn00000001	0 → Rn → T	1	LSB
SHLL2	Rn	0100nnnn00001000	Rn<<2 → Rn	1	—
SHLR2	Rn	0100nnnn00001001	Rn>>2 → Rn	1	—
SHLL8	Rn	0100nnnn00011000	Rn<<8 → Rn	1	—
SHLR8	Rn	0100nnnn00011001	Rn>>8 → Rn	1	—
SHLL16	Rn	0100nnnn00101000	Rn<<16 → Rn	1	—
SHLR16	Rn	0100nnnn00101001	Rn>>16 → Rn	1	—

Table 2.16 Branch Instructions

Instruction		Instruction Code	Operation	Exec. Cycles	T Bit
BF	label	10001011dddddddd	If T = 0, disp × 2 + PC → PC; if T = 1, nop	3/1*	—
BF/S	label	10001111dddddddd	Delayed branch, if T = 0, disp × 2 + PC → PC; if T = 1, nop	2/1*	—
BT	label	10001001dddddddd	If T = 1, disp × 2 + PC → PC; if T = 0, nop	3/1*	—
BT/S	label	10001101dddddddd	Delayed branch, if T = 1, disp × 2 + PC → PC; if T = 0, nop	2/1*	—
BRA	label	1010dddddddddddd	Delayed branch, disp × 2 + PC → PC	2	—
BRAF	Rm	0000mmmm00100011	Delayed branch, Rm + PC → PC	2	—
BSR	label	1011dddddddddddd	Delayed branch, PC → PR, disp × 2 + PC → PC	2	—
BSRF	Rm	0000mmmm00000011	Delayed branch, PC → PR, Rm + PC → PC	2	—
JMP	@Rm	0100mmmm00101011	Delayed branch, Rm → PC	2	—
JSR	@Rm	0100mmmm00001011	Delayed branch, PC → PR, Rm → PC	2	—
RTS		0000000000001011	Delayed branch, PR → PC	2	—

Note: * One state when it does not branch.

Table 2.17 System Control Instructions

Instruction		Instruction Code	Operation	Exec. Cycles	T Bit
CLRT		0000000000001000	0 → T	1	0
CLRMAC		0000000000101000	0 → MACH, MACL	1	—
LDC	Rm,SR	0100mmmm00001110	Rm → SR	1	LSB
LDC	Rm,GBR	0100mmmm00011110	Rm → GBR	1	—
LDC	Rm,VBR	0100mmmm00101110	Rm → VBR	1	—
LDC.L	@Rm+,SR	0100mmmm00000111	(Rm) → SR, Rm + 4 → Rm	3	LSB
LDC.L	@Rm+,GBR	0100mmmm00010111	(Rm) → GBR, Rm + 4 → Rm	3	—
LDC.L	@Rm+,VBR	0100mmmm00100111	(Rm) → VBR, Rm + 4 → Rm	3	—
LDS	Rm,MACH	0100mmmm00001010	Rm → MACH	1	—
LDS	Rm,MACL	0100mmmm00011010	Rm → MACL	1	—
LDS	Rm,PR	0100mmmm00101010	Rm → PR	1	—
LDS.L	@Rm+,MACH	0100mmmm00000110	(Rm) → MACH, Rm + 4 → Rm	1	—
LDS.L	@Rm+,MACL	0100mmmm00010110	(Rm) → MACL, Rm + 4 → Rm	1	—
LDS.L	@Rm+,PR	0100mmmm00100110	(Rm) → PR, Rm + 4 → Rm	1	—
NOP		0000000000001001	No operation	1	—
RTE		0000000000101011	Delayed branch, stack area → PC/SR	4	—
SETT		0000000000011000	1 → T	1	1
SLEEP		0000000000011011	Sleep	3*	—
STC	SR,Rn	0000nnnn00000010	SR → Rn	1	—
STC	GBR,Rn	0000nnnn00010010	GBR → Rn	1	—
STC	VBR,Rn	0000nnnn00100010	VBR → Rn	1	—
STC.L	SR,@–Rn	0100nnnn00000011	Rn–4 → Rn, SR → (Rn)	2	—
STC.L	GBR,@–Rn	0100nnnn00010011	Rn–4 → Rn, GBR → (Rn)	2	—
STC.L	VBR,@–Rn	0100nnnn00100011	Rn–4 → Rn, BR → (Rn)	2	—
STS	MACH,Rn	0000nnnn00001010	MACH → Rn	1	—
STS	MACL,Rn	0000nnnn00011010	MACL → Rn	1	—
STS	PR,Rn	0000nnnn00101010	PR → Rn	1	—

Table 2.17 System Control Instructions (cont)

Instruction		Instruction Code	Operation	Exec. Cycles	T Bit
STS.L	MACH,@–Rn	0100nnnn00000010	Rn–4 → Rn, MACH → (Rn)	1	—
STS.L	MACL,@–Rn	0100nnnn00010010	Rn–4 → Rn, MACL → (Rn)	1	—
STS.L	PR,@–Rn	0100nnnn00100010	Rn–4 → Rn, PR → (Rn)	1	—
TRAPA	#imm	11000011iiiiiiii	PC/SR → stack area, (imm × 4 + VBR) → PC	8	—

Note: * The number of execution cycles before the chip enters sleep mode: The execution cycles shown in the table are minimums. The actual number of cycles may be increased when (1) contention occurs between instruction fetches and data access, or (2) when the destination register of the load instruction (memory → register) and the register used by the next instruction are the same.

2.5 Processing States

2.5.1 State Transitions

The CPU has five processing states: reset, exception processing, bus release, program execution and power-down. Figure 2.6 shows the transitions between the states.

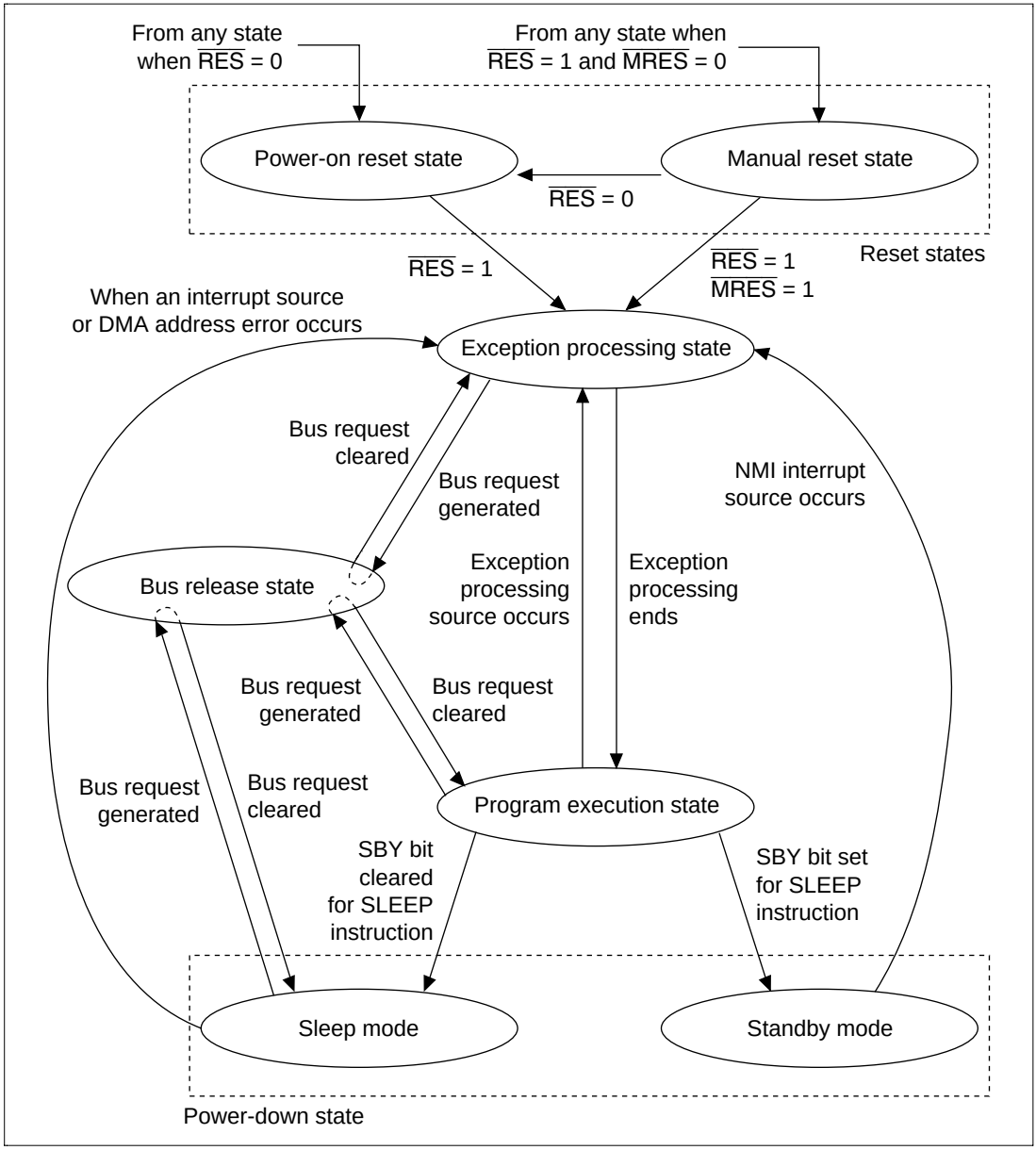


Figure 2.6 Transitions between Processing States

Reset State: The CPU resets in the reset state. When the $\overline{\text{RES}}$ pin level goes low, a power-on reset results. When the $\overline{\text{RES}}$ pin is high and MRES is low, a manual reset will occur.

Exception Processing State: The exception processing state is a transient state that occurs when exception processing sources such as resets or interrupts alter the CPU's processing state flow.

For a reset, the initial values of the program counter (PC) (execution start address) and stack pointer (SP) are fetched from the exception processing vector table and stored; the CPU then branches to the execution start address and execution of the program begins.

For an interrupt, the stack pointer (SP) is accessed and the program counter (PC) and status register (SR) are saved to the stack area. The exception service routine start address is fetched from the exception processing vector table; the CPU then branches to that address and the program starts executing, thereby entering the program execution state.

Program Execution State: In the program execution state, the CPU sequentially executes the program.

Power-Down State: In the power-down state, the CPU operation halts and power consumption declines. The SLEEP instruction places the CPU in the power-down state. This state has two modes: sleep mode and standby mode.

Bus Release State: In the bus release state, the CPU releases access rights to the bus to the device that has requested them.

2.5.2 Power-Down State

Besides the ordinary program execution states, the CPU also has a power-down state in which CPU operation halts, lowering power consumption. There are two power-down state modes: sleep mode and standby mode.

Sleep Mode: When standby bit SBY (in the standby control register SBYCR) is cleared to 0 and a SLEEP instruction executed, the CPU moves from program execution state to sleep mode. In the sleep mode, the CPU halts and the contents of its internal registers and the data in on-chip cache (or on-chip RAM) is maintained. The on-chip peripheral modules other than the CPU do not halt in the sleep mode.

To return from sleep mode, use a reset (power-on or manual), any interrupt, or a DMA address error; the CPU returns to the ordinary program execution state through the exception processing state.

Standby Mode: To enter the standby mode, set the standby bit SBY (in the standby control register SBYCR) to 1 and execute a SLEEP instruction. In standby mode, all CPU, on-chip peripheral module, and oscillator functions are halted. However, when entering standby mode, the DMA master enable bit of the DMAC should be set to 0. If multiplication-related instructions are being executed at the time of entry into standby mode, the values of MACH and MACL will become undefined.

To return from standby mode, use a reset (power-on or manual) or an NMI interrupt. For resets, the CPU returns to ordinary program execution state through the exception processing state when placed in a reset state for the duration of the oscillator stabilization time. For NMI interrupts, the

CPU returns to ordinary program execution state through the exception processing state after the oscillator stabilization time has elapsed. In this mode, power consumption drops markedly, since the oscillator stops (table 2.18).

Table 2.18 Power-Down State

		State						
Mode	**Transition Conditions**	**Clock**	**CPU**	**On-Chip Peripheral Modules**	**CPU Registers**	**On-Chip Cache or On-Chip RAM**	**I/O Port Pins**	**Canceling**
Sleep	Execute SLEEP instruction with SBY bit cleared to 0 in SBYCR	Run	Halt	Run	Held	Held	Held	• Interrupt • DMA address error • Power-on reset • Manual reset
Stand-by	Execute SLEEP instruction with SBY bit set to 1 in SBYCR	Halt	Halt	Halt and initialize*	Held	Held	Held or Hi-Z (select-able)	• NMI interrupt • Power-on reset • Manual reset

Note: * Differs depending on the peripheral module and pin.

Section 3 Operating Modes

3.1 Operating Modes, Types, and Selection

This LSI has five operating modes and three clock modes, determined by the setting of the mode pins (MD3–MD0). Do not change the mode pin settings during LSI operation (while power is on). (In the F-ZTAT version, however, MD1 can be changed in the power-on reset state.)

Table 3.1 indicates the setting method for the operating mode.

Table 3.1 Operating Mode Setting

Mode No.	Pin Setting					Mode Name	On-Chip ROM	CS0 Area	
	FWP	MD3*1	MD2*1	MD1	MD0			112 Pin	144 Pin
0	1	x	x	0	0	MCU mode 0	Not Active	8-bit space	16-bit space
1	1	x	x	0	1	MCU mode 1	Not Active	16-bit space	32-bit space
2	1	x	x	1	0	MCU mode 2	Active	8/16-bit space*2	8/16/32-bit space*2
3	1	x	x	1	1	Single chip mode	Active	—	—
4	1	1	1	1	1	PROM mode*3	Active	—	—
—	0	x	x	0	0	Boot mode*4	Active	8/16-bit space*2	8/16/32-bit space*2
—	0	x	x	0	1			—	—
—	0	x	x	1	0	User programming mode*4	Active	8/16-bit space*2	8/16/32-bit space*2
—	0	x	x	1	1			—	—
—	1	1	1	0	1	Flash programmer mode*4	Active	—	—

Notes: *1 MD2 and MD3 pins select the clock mode in modes 0–3 (table 3.2).
*2 Set by BCR2 of BSC.
*3 Only ZTAT.
*4 Only F-ZTAT.

Table 3.2 indicates the setting method for the clock mode.

Table 3.2 Clock Mode Setting

MD3	MD2	Clock Mode
0	0	PLL ON × 1
0	1	PLL ON × 2
1	0	PLL ON × 4
1	1	Reserved (PROM mode only)

3.2 Explanation of Operating Modes

Table 3.3 describes the operating modes.

Table 3.3 Operating Modes

Mode	Description
(MCU) Mode 0	CS0 area becomes an external memory space with 8-bit bus width for the 112-pin version, and 16-bit for the 144-pin version.
(MCU) Mode 1	CS0 area becomes an external memory space with 16-bit bus width for the 112-pin version, and 32-bit for the 144-pin version
(MCU) Mode 2	The on-chip ROM becomes effective. The bus width for the on-chip ROM space is 32 bit.
Mode 3 (single chip mode)	Any port can be used, but external addresses can not be employed.
Mode 4 (PROM mode)	On-chip ROM can be programmed using a general PROM writer.
Clock mode	The input waveform frequency can be used as is, doubled or quadrupled as an internal clock in modes 0 to 3.

3.3 Pin Configuration

Table 3.4 describes the function of each operating mode related pin.

Table 3.4 Operating Mode Pin Function

Pin Name	Input/Output	Function
XTAL	Input	Connects to a crystal oscillator
EXTAL	Input	Connects to a crystal oscillator, or used for external clock input pin
PLLCAP	Input	Connects to a capacitor for PLL circuit operation
MD0	Input	Designates operating mode through the level applied to this pin
MD1	Input	Designates operating mode through the level applied to this pin
MD2	Input	Designates clock mode through the level applied to this pin
MD3	Input	Designates clock mode through the level applied to this pin

Section 4 Clock Pulse Generator (CPG)

4.1 Overview

The SH7040 Series has an on-chip clock pulse generator (CPG) that generates the system clock (ϕ), as well as the internal clock (ϕ/2 to ϕ/8192). The CPG consists of an oscillator, a PLL, and a prescaler.

4.1.1 Block Diagram

A block diagram of the clock pulse generator is shown in figure 4.1.

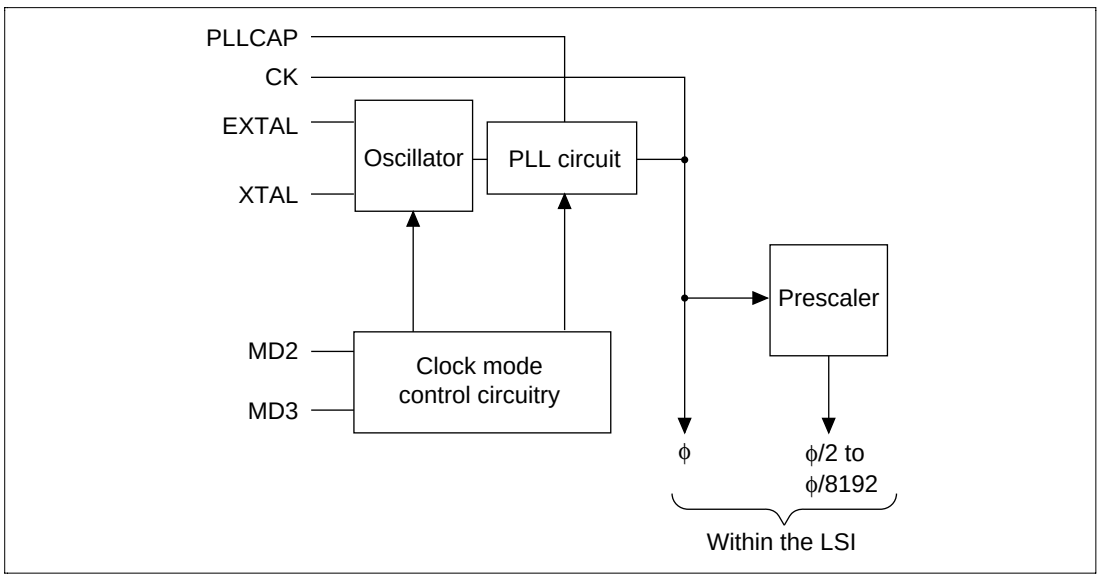


Figure 4.1 Block Diagram of the Clock Pulse Generator

4.2 Oscillator

Clock pulses can be supplied from a connected crystal resonator or an external clock.

4.2.1 Connecting a Crystal Oscillator

Circuit Configuration: A crystal oscillator can be connected as shown in figure 4.2. Use the damping resistance (Rd) listed in table 4.1. Use a 4–10 MHz crystal oscillator (consult your dealer concerning the compatibility of the crystal oscillator and the LSI).

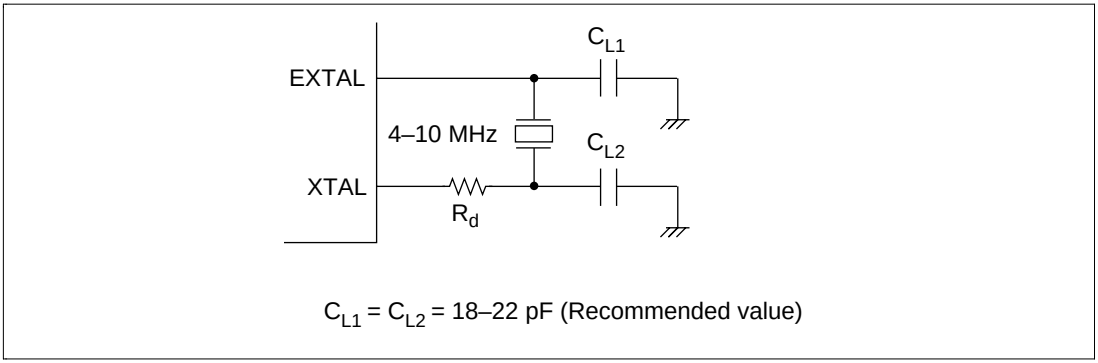


Figure 4.2 Connection of the Crystal Oscillator (Example)

Table 4.1 Damping Resistance Values (Recommended Values)

	Frequency (MHz)		
Parameter	**4**	**8**	**10**
Rd (Ω)	500	200	0

Crystal Oscillator: Figure 4.3 shows an equivalent circuit of the crystal oscillator. Use a crystal oscillator with the characteristics listed in table 4.2.

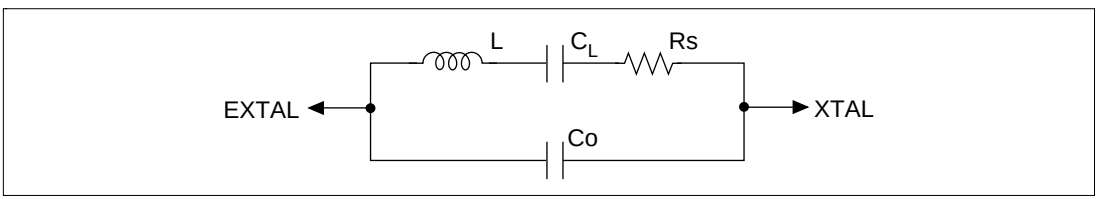


Figure 4.3 Crystal Oscillator Equivalent Circuit

Table 4.2 Crystal Oscillator Parameters

	Frequency (MHz)		
Parameter	**4**	**8**	**10**
Rs max (Ω)	120	80	60
Co max (pF)	7	7	7

4.2.2 External Clock Input Method

Figure 4.4 shows an example of an external clock input connection. In this case, make the external clock high level to stop it when in standby mode. During operation, make the external input clock frequency 4–10 MHz.

When leaving the XTAL pin open, make sure the parasitic capacitance is less than 10 pF.

Even when inputting an external clock, be sure to delay until after the oscillation stabilization time (upon power-on) or after release from standby, in order to ensure the PLL stabilization time.

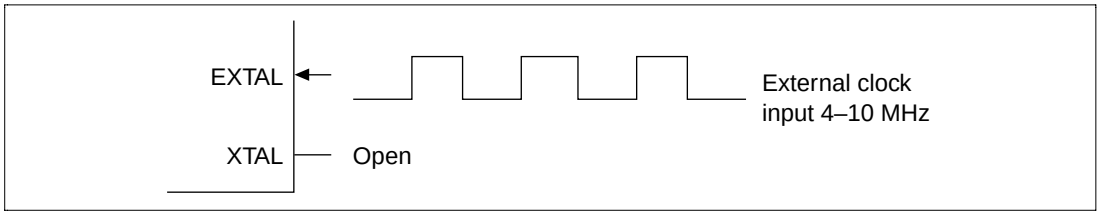


Figure 4.4 Example of External Clock Connection

4.3 Prescaler

The prescaler divides the system clock (ϕ) to generate an internal clock (ϕ/2 to ϕ/8192) for supply to peripheral modules.

4.4 Oscillator Halt Function

This CPG can detect a clock halt and automatically cause the timer pins to become high-impedance when any system abnormality causes the oscillator to halt. That is, when a change of EXTAL has not been detected, the high-current six pins (PE9/TIOC3B, PE11/TIOC3D, PE12/TIOC4A, PE13/TIOC4B/$\overline{\text{MRES}}$, PE14/TIOC4C/DACK0/AH, PE15/TIOC4D/DACK1/$\overline{\text{IRQOUT}}$) are set to high-impedance regardless of PFC setting.

Even in standby mode, these six pins become high-impedance regardless of PFC setting. These pins enter the normal state after standby mode is cancelled.When abnormalities that halt the oscillator occur except in standby mode, other LSI operations become undefined. In this case, LSI operations, including these six pins, become undefined even when the oscillator operation starts again.

4.5 Usage Notes

4.5.1 Oscillator Usage Notes

Since the characteristics of the oscillator are closely related to the user-defined board settings, the user should refer to the connection examples in this section and perform a careful evaluation. The oscillator circuit ratings will differ depending on factors such as the oscillator used and the stray capacitance of the mounted circuitry. Therefore, the oscillator manufacturer should be consulted before a decision is made. Make sure that the voltage applied to the oscillator does not exceed the maximum rating.

4.5.2 Notes on Board Design

When connecting a crystal oscillator, observe the following precautions:

- To prevent induction from interfering with correct oscillation, do not route any signal lines near the oscillator circuitry.
- When designing the board, place the crystal oscillator and its load capacitors as close as possible to the XTAL and EXTAL pins.

Figure 4.5 shows the precautions regarding oscillator block board settings.

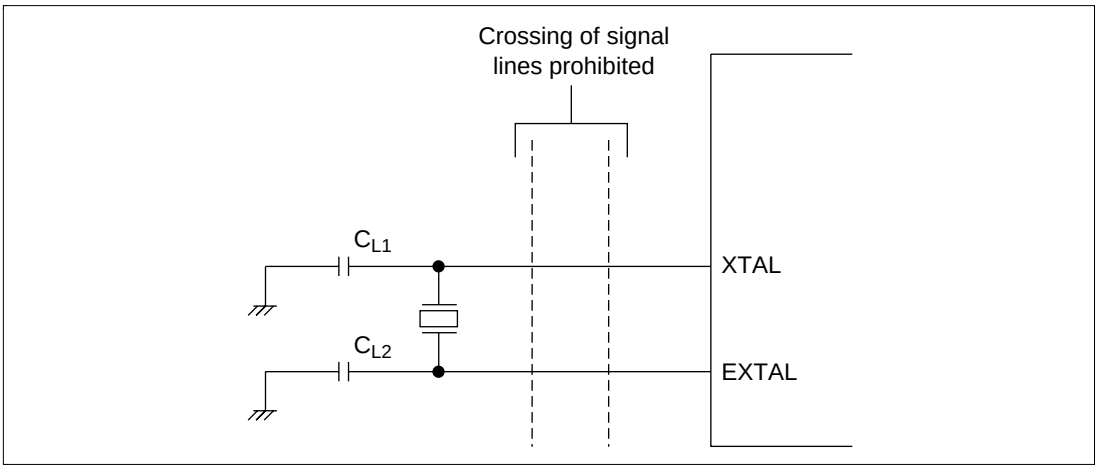


Figure 4.5 Cautions for Oscillator Circuit System Board Design

External circuitry such as that shown in figure 4.6 is recommended around the PLL.

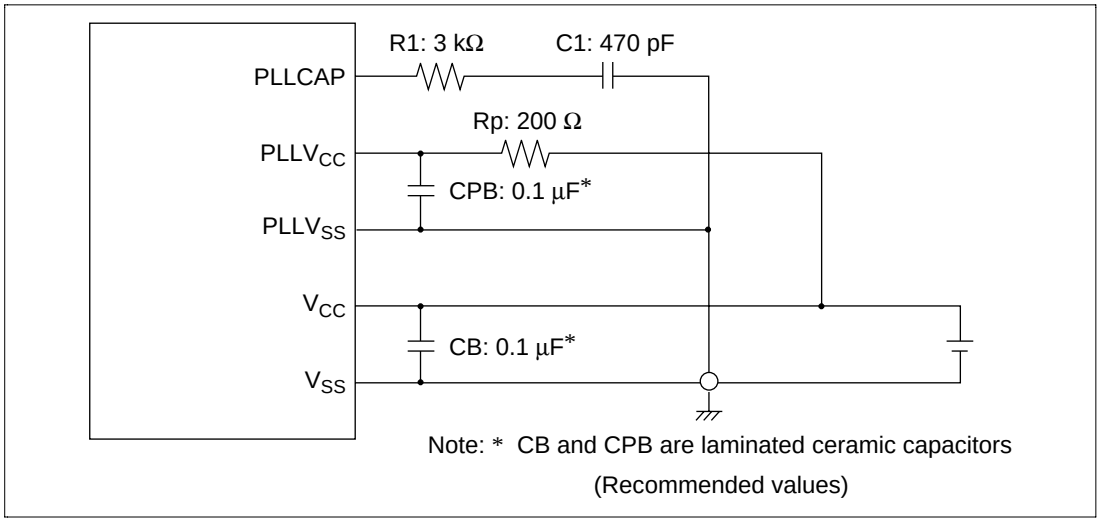


Figure 4.6 Cautions for Use of PLL Oscillator Circuit

Place oscillation stabilization capacitor C1 and resistor R1 near the PLLCAP pin, and ensure that these lines do not cross any other signal lines. Supply the C1 ground from $PLLV_{SS}$.

Also, separate $PLLV_{CC}$ and $PLLV_{SS}$, and the other V_{CC} and V_{SS} pins, from the board power supply source, and be sure to insert bypass capacitors CPB and CB close to the pins.

If V_{CC} and $PLLV_{CC}$ are both 3.3 V ± 0.3 V, it is recommended that Rp be set to 0 Ω.

4.5.3 Spread Spectrum Clock Generator Usage Notes

The following points should be borne in mind when using a spread spectrum clock generator as an external oscillator in order to reduce radiation noise.

- Set the center frequency and the spread amplitude such that the internal clock does not exceed the maximum frequency during spread spectrum operation.
- Using a spread spectrum clock generator may trigger the oscillator halt function described in section 4.4. If the system configuration is such that this function will cause problems, a spread spectrum clock generator should not be used.

Section 5 Exception Processing

5.1 Overview

5.1.1 Types of Exception Processing and Priority

Exception processing is started by four sources: resets, address errors, interrupts and instructions and have the priority shown in table 5.1. When several exception processing sources occur at once, they are processed according to the priority shown.

Table 5.1 Types of Exception Processing and Priority Order

Exception	Source		Priority
Reset	Power-on reset		High
	Manual reset		
Address error	CPU address error		
	DMAC/DTC address error		
Interrupt	NMI		
	User break		
	IRQ		
	On-chip peripheral modules:	• Direct memory access controller (DMAC) • Multifunction timer/pulse unit (MTU) • Serial communications interface (SCI) • A/D converter (A/D)*3 • Data transfer controller (DTC) • Compare match timer (CMT) • Watchdog timer (WDT) • Bus state controller (BSC) • Port output enable control section	
Instructions	Trap instruction (TRAPA instruction)		
	General illegal instructions (undefined code)		
	Illegal slot instructions (undefined code placed directly after a delay branch instruction*1 or instructions that rewrite the PC*2)		Low

Notes: *1 Delayed branch instructions: JMP, JSR, BRA, BSR, RTS, RTE, BF/S, BT/S, BSRF, BRAF.

*2 Instructions that rewrite the PC: JMP, JSR, BRA, BSR, RTS, RTE, BT, BF, TRAPA, BF/S, BT/S, BSRF, BRAF.

*3 A mask products: A/D0, A/D1.

5.1.2 Exception Processing Operations

The exception processing sources are detected and begin processing according to the timing shown in table 5.2.

Table 5.2 Timing of Exception Source Detection and the Start of Exception Processing

Exception	Source	Timing of Source Detection and Start of Processing
Reset	Power-on reset	Starts when the $\overline{\text{RES}}$ pin changes from low to high.
	Manual reset	Starts when the $\overline{\text{RES}}$ pin is high and the $\overline{\text{MRES}}$ pin changes from low to high.
Address error		Detected when instruction is decoded and starts when the previous executing instruction finishes executing.
Interrupts		Detected when instruction is decoded and starts when the previous executing instruction finishes executing.
Instructions	Trap instruction	Starts from the execution of a TRAPA instruction.
	General illegal instructions	Starts from the decoding of undefined code anytime except after a delayed branch instruction (delay slot).
	Illegal slot instructions	Starts from the decoding of undefined code placed in a delayed branch instruction (delay slot) or of instructions that rewrite the PC.

When exception processing starts, the CPU operates as follows:

1. Exception processing triggered by reset:

   The initial values of the program counter (PC) and stack pointer (SP) are fetched from the exception processing vector table (PC and SP are respectively the H'00000000 and H'00000004 addresses for power-on resets and the H'00000008 and H'0000000C addresses for manual resets). See section 5.1.3, Exception Processing Vector Table, for more information. 0 is then written to the vector base register (VBR) and 1111 is written to the interrupt mask bits (I3–I0) of the status register (SR). The program begins running from the PC address fetched from the exception processing vector table.
2. Exception processing triggered by address errors, interrupts and instructions:

   SR and PC are saved to the stack indicated by R15. For interrupt exception processing, the interrupt priority level is written to the SR's interrupt mask bits (I3–I0). For address error and instruction exception processing, the I3–I0 bits are not affected. The start address is then fetched from the exception processing vector table and the program begins running from that address.

5.1.3 Exception Processing Vector Table

Before exception processing begins running, the exception processing vector table must be set in memory. The exception processing vector table stores the start addresses of exception service routines. (The reset exception processing table holds the initial values of PC and SP.)

All exception sources are given different vector numbers and vector table address offsets, from which the vector table addresses are calculated. During exception processing, the start addresses of the exception service routines are fetched from the exception processing vector table, which indicated by this vector table address.

Table 5.3 shows the vector numbers and vector table address offsets. Table 5.4 shows how vector table addresses are calculated.

Table 5.3 Exception Processing Vector Table

Exception Sources		Vector Numbers	Vector Table Address Offset
Power-on reset	PC	0	H'00000000–H'00000003
	SP	1	H'00000004–H'00000007
Manual reset	PC	2	H'00000008–H'0000000B
	SP	3	H'0000000C–H'0000000F
General illegal instruction		4	H'00000010–H'00000013
(Reserved by system)		5	H'00000014–H'00000017
Slot illegal instruction		6	H'00000018–H'0000001B
(Reserved by system)		7	H'0000001C–H'0000001F
(Reserved by system)		8	H'00000020–H'00000023
CPU address error		9	H'00000024–H'00000027
DMAC/DTC address error		10	H'00000028–H'0000002B
Interrupts	NMI	11	H'0000002C–H'0000002F
	User break	12	H'00000030–H'00000033
(Reserved by system)		13	H'00000034–H'00000037
		:	:
		31	H'0000007C–H'0000007F
Trap instruction (user vector)		32	H'00000080–H'00000083
		:	:
		63	H'000000FC–H'000000FF

Table 5.3 Exception Processing Vector Table (cont)

Exception Sources		Vector Numbers	Vector Table Address Offset
Interrupts	IRQ0	64	H'00000100–H'00000103
	IRQ1	65	H'00000104–H'00000107
	IRQ2	66	H'00000108–H'0000010B
	IRQ3	67	H'0000010C–H'0000010F
	IRQ4	68	H'00000110–H'00000113
	IRQ5	69	H'00000114–H'00000117
	IRQ6	70	H'00000118–H'0000011B
	IRQ7	71	H'0000011C–H'0000011F
On-chip peripheral module*		72	H'00000120–H'00000124
		:	:
		255	H'000003FC–H'000003FF

Note: * The vector numbers and vector table address offsets for each on-chip peripheral module interrupt are given in section 6, Interrupt Controller (INTC), and table 6.3, Interrupt Exception Processing Vectors and Priorities.

Table 5.4 Calculating Exception Processing Vector Table Addresses

Exception Source	Vector Table Address Calculation
Resets	Vector table address = (vector table address offset) = (vector number) × 4
Address errors, interrupts, instructions	Vector table address = VBR + (vector table address offset) = VBR + (vector number) × 4

Notes: 1. VBR: Vector base register
2. Vector table address offset: See table 5.3.
3. Vector number: See table 5.3.

5.2 Resets

Resets have the highest priority of any exception source. There are two types of resets: manual resets and power-on resets. As table 5.5 shows, both types of resets initialize the internal status of the CPU. In power-on resets, all registers of the on-chip peripheral modules are initialized; in manual resets, they are not.

Table 5.5 Types of Resets

	Conditions for Transition to Reset Status		Internal Status	
Type	**$\overline{\text{RES}}$**	**$\overline{\text{MRES}}$**	**CPU**	**On-Chip Peripheral Module**
Power-on reset	Low	—	Initialized	Initialized
Manual reset	High	Low	Initialized	Not initialized

5.2.1 Power-On Reset

When the $\overline{\text{RES}}$ pin is driven low, the LSI does a power-on reset. To reliably reset the LSI, the $\overline{\text{RES}}$ pin should be kept at low for at least the duration of the oscillation settling time when applying power or when in standby mode (when the clock circuit is halted) or at least 20 t_{cyc} (when the clock circuit is running). During power-on reset, CPU internal status and all registers of on-chip peripheral modules are initialized. See Appendix C, Pin States, for the status of individual pins during the power-on reset status.

In the power-on reset status, power-on reset exception processing starts when the $\overline{\text{RES}}$ pin is first driven low for a set period of time and then returned to high. The CPU will then operate as follows:

1. The initial value (execution start address) of the program counter (PC) is fetched from the exception processing vector table.
2. The initial value of the stack pointer (SP) is fetched from the exception processing vector table.
3. The vector base register (VBR) is cleared to H'00000000 and the interrupt mask bits (I3–I0) of the status register (SR) are set to H'F (1111).
4. The values fetched from the exception processing vector table are set in the program counter (PC) and SP and the program begins executing.

Be certain to always perform power-on reset processing when turning the system power on.

5.2.2 Manual Reset

When the $\overline{\text{RES}}$ pin is high and the $\overline{\text{MRES}}$ pin is driven low, the LSI does a manual reset. To reliably reset the LSI, the $\overline{\text{MRES}}$ pin should be kept at low for at least the duration of the oscillation settling time when in standby mode (when the clock is halted) or at least 20 t_{cyc} when the clock is operating. During manual reset, the CPU internal status is initialized. Registers of on-chip peripheral modules are not initialized. Since the BSC is not affected, the DRAM refresh control functions remain operational even when the manual reset status continues for a long period of time. When the LSI enters manual reset status in the middle of a bus cycle, manual reset exception processing does not start until the bus cycle has ended. Thus, manual resets do not abort bus cycles. However, the bus cycle ends once $\overline{\text{MRES}}$ is driven low. Hold at low level until manual

reset mode. (Keep at low level for at least the longest bus cycle.) See Appendix C, Pin States, for the status of individual pins during manual reset mode.

In the manual reset status, manual reset exception processing starts when the $\overline{\text{MRES}}$ pin is first kept low for a set period of time and then returned to high. The CPU will then operate the same as described for power-on resets.

5.3 Address Errors

Address errors occur when instructions are fetched or data read or written, as shown in table 5.6.

Table 5.6 Bus Cycles and Address Errors

Bus Cycle Type	Bus Master	Bus Cycle Description	Address Errors
Instruction fetch	CPU	Instruction fetched from even address	None (normal)
		Instruction fetched from odd address	Address error occurs
		Instruction fetched from other than on-chip peripheral module space*	None (normal)
		Instruction fetched from on-chip peripheral module space*	Address error occurs
		Instruction fetched from external memory space when in single chip mode	Address error occurs
Data read/write	CPU or DMAC or DTC	Word data accessed from even address	None (normal)
		Word data accessed from odd address	Address error occurs
		Longword data accessed from a longword boundary	None (normal)
		Longword data accessed from other than a long-word boundary	Address error occurs
		Byte or word data accessed in on-chip peripheral module space*	None (normal)
		Longword data accessed in 16-bit on-chip peripheral module space*	None (normal)
		Longword data accessed in 8-bit on-chip peripheral module space*	Address error occurs
		External memory space accessed when in single chip mode	Address error occurs

Note: * See section 10, Bus State Controller (BSC).

5.3.1 Address Error Exception Processing

When an address error occurs, the bus cycle in which the address error occurred ends. When the executing instruction then finishes, address error exception processing starts up. The CPU operates as follows:

1. The status register (SR) is saved to the stack.
2. The program counter (PC) is saved to the stack. The PC value saved is the start address of the instruction to be executed after the last executed instruction.
3. The exception service routine start address is fetched from the exception processing vector table that corresponds to the address error that occurred and the program starts executing from that address. The jump that occurs is not a delayed branch.

5.4 Interrupts

Table 5.7 shows the sources that start up interrupt exception processing. These are divided into NMI, user breaks, IRQ, and on-chip peripheral modules.

Table 5.7 Interrupt Sources

Type	Request Source	Number of Sources
NMI	NMI pin (external input)	1
User break	User break controller	1
IRQ	$\overline{\text{IRQ0}}$–$\overline{\text{IRQ7}}$ (external input)	8
On-chip peripheral module	Direct memory access controller (DMAC)	4
	Multifunction timer/pulse unit (MTU)	24
	Serial communications interface (SCI)	8
	A/D converter	1*
	Data transfer controller (DTC)	1
	Compare match timer (CMT)	2
	Watchdog timer (WDT)	1
	Bus state controller (BSC)	1
	Port	1

Note: * For A mask products, (A/D0, A/D1) is 2

Each interrupt source is allocated a different vector number and vector table offset. See section 6, Interrupt Controller (INTC), and table 6.3, Interrupt Exception Processing Vectors and Priorities, for more information on vector numbers and vector table address offsets.

5.4.1 Interrupt Priority Level

The interrupt priority order is predetermined. When multiple interrupts occur simultaneously (overlap), the interrupt controller (INTC) determines their relative priorities and starts up processing according to the results.

The priority order of interrupts is expressed as priority levels 0–16, with priority 0 the lowest and priority 16 the highest. The NMI interrupt has priority 16 and cannot be masked, so it is always accepted. The user break interrupt priority level is 15. IRQ interrupts and on-chip peripheral module interrupt priority levels can be set freely using the INTC's interrupt priority level setting registers A through H (IPRA–IPRH) as shown in table 5.8. The priority levels that can be set are 0–15. Level 16 cannot be set. See section 6.3.1, Interrupt Priority Registers A-H (IPRA-IPRH), for more information on IPRA to IPRH.

Table 5.8 Interrupt Priority Order

Type	Priority Level	Comment
NMI	16	Fixed priority level. Cannot be masked.
User break	15	Fixed priority level.
IRQ	0–15	Set with interrupt priority level setting registers A through H (IPRA–IPRH).
On-chip peripheral module	0–15	Set with interrupt priority level setting registers A through H (IPRA–IPRH).

5.4.2 Interrupt Exception Processing

When an interrupt occurs, its priority level is ascertained by the interrupt controller (INTC). NMI is always accepted, but other interrupts are only accepted if they have a priority level higher than the priority level set in the interrupt mask bits (I3–I0) of the status register (SR).

When an interrupt is accepted, exception processing begins. In interrupt exception processing, the CPU saves SR and the program counter (PC) to the stack. The priority level value of the accepted interrupt is written to SR bits I3–I0. For NMI, however, the priority level is 16, but the value set in I3–I0 is H'F (level 15). Next, the start address of the exception service routine is fetched from the exception processing vector table for the accepted interrupt, that address is jumped to and execution begins. See section 6.4, Interrupt Operation, for more information on the interrupt exception processing.

5.5 Exceptions Triggered by Instructions

Exception processing can be triggered by trap instructions, general illegal instructions, and illegal slot instructions, as shown in table 5.9.

Table 5.9 Types of Exceptions Triggered by Instructions

Type	Source Instruction	Comment
Trap instructions	TRAPA	—
Illegal slot instructions	Undefined code placed immediately after a delayed branch instruction (delay slot) and instructions that rewrite the PC	Delayed branch instructions: JMP, JSR, BRA, BSR, RTS, RTE, BF/S, BT/S, BSRF, BRAF Instructions that rewrite the PC: JMP, JSR, BRA, BSR, RTS, RTE, BT, BF, TRAPA, BF/S, BT/S, BSRF, BRAF
General illegal instructions	Undefined code anywhere besides in a delay slot	—

5.5.1 Trap Instructions

When a TRAPA instruction is executed, trap instruction exception processing starts up. The CPU operates as follows:

1. The status register (SR) is saved to the stack.
2. The program counter (PC) is saved to the stack. The PC value saved is the start address of the instruction to be executed after the TRAPA instruction.
3. The exception service routine start address is fetched from the exception processing vector table that corresponds to the vector number specified in the TRAPA instruction. That address is jumped to and the program starts executing. The jump that occurs is not a delayed branch.

5.5.2 Illegal Slot Instructions

An instruction placed immediately after a delayed branch instruction is said to be placed in a delay slot. When the instruction placed in the delay slot is undefined code, illegal slot exception processing starts up when that undefined code is decoded. Illegal slot exception processing also starts up when an instruction that rewrites the program counter (PC) is placed in a delay slot. The processing starts when the instruction is decoded. The CPU handles an illegal slot instruction as follows:

1. The status register (SR) is saved to the stack.
2. The program counter (PC) is saved to the stack. The PC value saved is the jump address of the delayed branch instruction immediately before the undefined code or the instruction that rewrites the PC.
3. The exception service routine start address is fetched from the exception processing vector table that corresponds to the exception that occurred. That address is jumped to and the program starts executing. The jump that occurs is not a delayed branch.

5.5.3 General Illegal Instructions

When undefined code placed anywhere other than immediately after a delayed branch instruction (i.e., in a delay slot) is decoded, general illegal instruction exception processing starts up. The CPU handles general illegal instructions the same as illegal slot instructions. Unlike processing of illegal slot instructions, however, the program counter value stored is the start address of the undefined code.

5.6 When Exception Sources Are Not Accepted

When an address error or interrupt is generated after a delayed branch instruction or interrupt-disabled instruction, it is sometimes not accepted immediately but stored instead, as shown in table 5.10. When this happens, it will be accepted when an instruction that can accept the exception is decoded.

Table 5.10 Generation of Exception Sources Immediately after a Delayed Branch Instruction or Interrupt-Disabled Instruction

	Exception Source	
Point of Occurrence	**Address Error**	**Interrupt**
Immediately after a delayed branch instruction*1	Not accepted	Not accepted
Immediately after an interrupt-disabled instruction*2	Accepted	Not accepted

Notes: *1 Delayed branch instructions: JMP, JSR, BRA, BSR, RTS, RTE, BF/S, BT/S, BSRF, BRAF

*2 Interrupt-disabled instructions: LDC, LDC.L, STC, STC.L, LDS, LDS.L, STS, STS.L

5.6.1 Immediately after a Delayed Branch Instruction

When an instruction placed immediately after a delayed branch instruction (delay slot) is decoded, neither address errors nor interrupts are accepted. The delayed branch instruction and the instruction located immediately after it (delay slot) are always executed consecutively, so no exception processing occurs during this period.

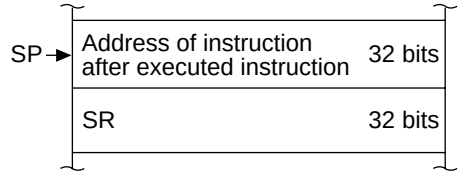
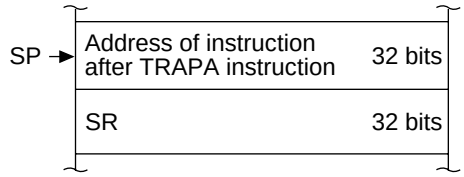
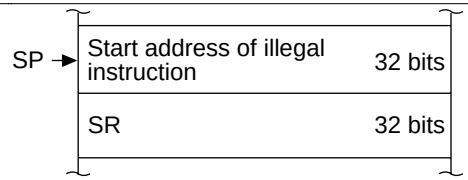
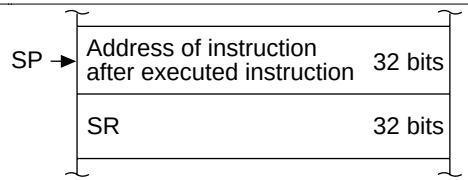
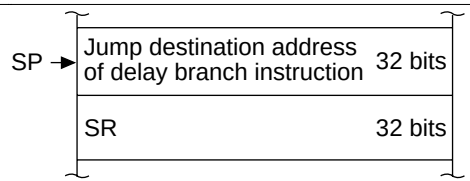
5.6.2 Immediately after an Interrupt-Disabled Instruction

When an instruction immediately following an interrupt-disabled instruction is decoded, interrupts are not accepted. Address errors are accepted.

5.7 Stack Status after Exception Processing Ends

The status of the stack after exception processing ends is as shown in table 5.11.

Table 5.11 Types of Stack Status after Exception Processing Ends

Types	Stack Status
Address error	
Trap instruction	
General illegal instruction	
Interrupt	
Illegal slot instruction	

5.8 Notes on Use

5.8.1 Value of Stack Pointer (SP)

The value of the stack pointer must always be a multiple of four. If it is not, an address error will occur when the stack is accessed during exception processing.

5.8.2 Value of Vector Base Register (VBR)

The value of the vector base register must always be a multiple of four. If it is not, an address error will occur when the stack is accessed during exception processing.

5.8.3 Address Errors Caused by Stacking of Address Error Exception Processing

When the stack pointer is not a multiple of four, an address error will occur during stacking of the exception processing (interrupts, etc.) and address error exception processing will start up as soon as the first exception processing is ended. Address errors will then also occur in the stacking for this address error exception processing. To ensure that address error exception processing does not go into an endless loop, no address errors are accepted at that point. This allows program control to be shifted to the address error exception service routine and enables error processing.

When an address error occurs during exception processing stacking, the stacking bus cycle (write) is executed. During stacking of the status register (SR) and program counter (PC), the SP is –4 for both, so the value of SP will not be a multiple of four after the stacking either. The address value output during stacking is the SP value, so the address where the error occurred is itself output. This means the write data stacked will be undefined.

Section 6 Interrupt Controller (INTC)

6.1 Overview

The interrupt controller (INTC) ascertains the priority of interrupt sources and controls interrupt requests to the CPU. The INTC has registers for setting the priority of each interrupt which can be used by the user to order the priorities in which the interrupt requests are processed.

6.1.1 Features

The INTC has the following features:

- 16 levels of interrupt priority: By setting the eight interrupt-priority level registers, the priorities of IRQ interrupts and on-chip peripheral module interrupts can be set in 16 levels for different request sources.
- NMI noise canceler function: NMI input level bits indicate the NMI pin status. By reading these bits with the interrupt exception service routine, the pin status can be confirmed, enabling it to be used as a noise canceler.
- Notification of interrupt occurrence can be reported externally ($\overline{\text{IRQOUT}}$ pin). For example, it is possible to request bus rights if an external bus master is informed that a peripheral module interrupt has occurred when the LSI has released the bus rights.

6.1.2 Block Diagram

Figure 6.1 is a block diagram of the INTC.

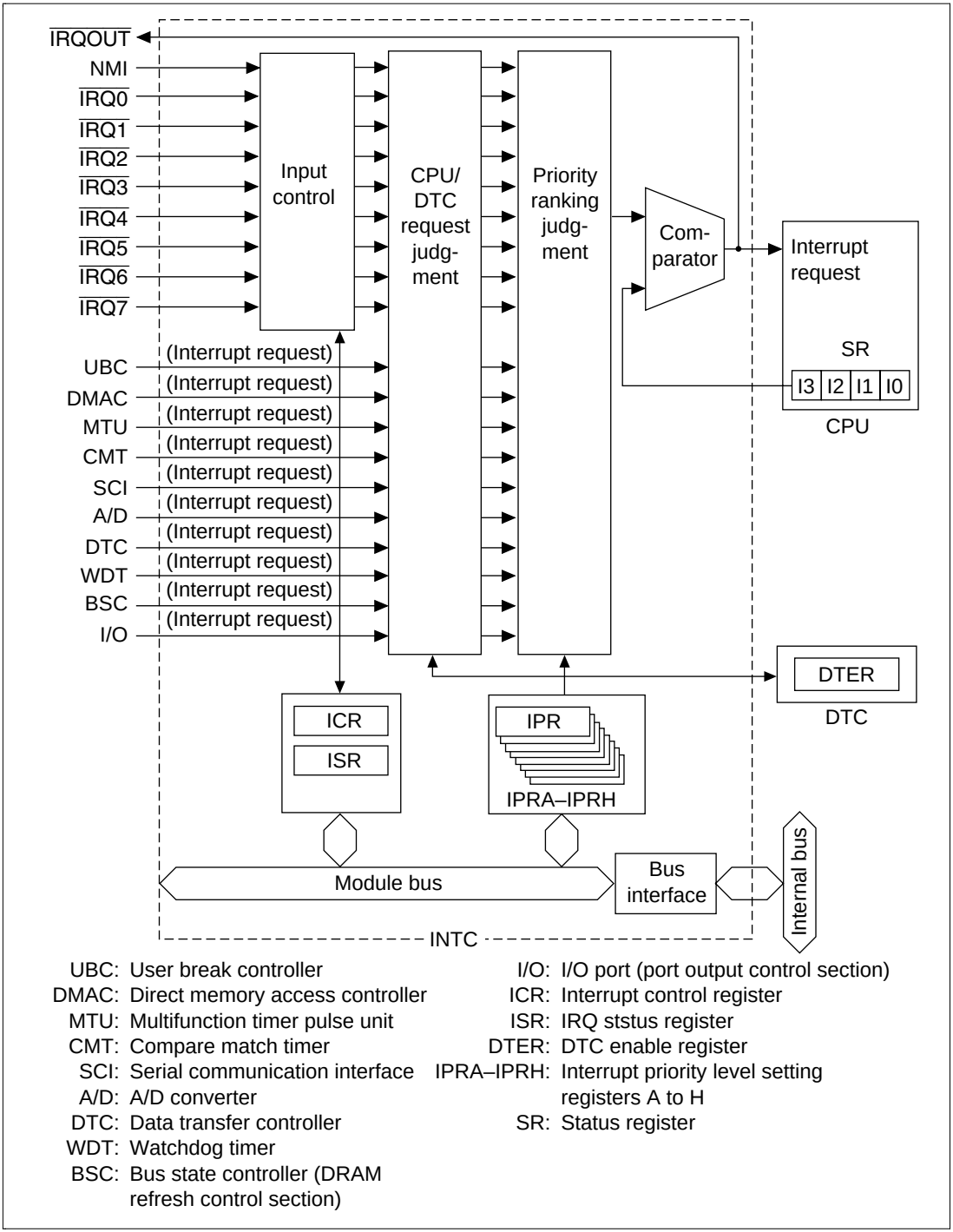


Figure 6.1 INTC Block Diagram

6.1.3 Pin Configuration

Table 6.1 shows the INTC pin configuration.

Table 6.1 Pin Configuration

Name	Abbreviation	I/O	Function
Non-maskable interrupt input pin	NMI	I	Input of non-maskable interrupt request signal
Interrupt request input pins	$\overline{\text{IRQ0}}$–$\overline{\text{IRQ7}}$	I	Input of maskable interrupt request signals
Interrupt request output pin	$\overline{\text{IRQOUT}}$	O	Output of notification signal when an interrupt has occurred

6.1.4 Register Configuration

The INTC has the 10 registers shown in table 6.2. These registers set the priority of the interrupts and control external interrupt input signal detection.

Table 6.2 Register Configuration

Name	Abbr.	R/W	Initial Value	Address	Access Sizes
Interrupt priority register A	IPRA	R/W	H'0000	H'FFFF8348	8, 16, 32
Interrupt priority register B	IPRB	R/W	H'0000	H'FFFF834A	8, 16, 32
Interrupt priority register C	IPRC	R/W	H'0000	H'FFFF834C	8, 16, 32
Interrupt priority register D	IPRD	R/W	H'0000	H'FFFF834E	8, 16, 32
Interrupt priority register E	IPRE	R/W	H'0000	H'FFFF8350	8, 16, 32
Interrupt priority register F	IPRF	R/W	H'0000	H'FFFF8352	8, 16, 32
Interrupt priority register G	IPRG	R/W	H'0000	H'FFFF8354	8, 16, 32
Interrupt priority register H	IPRH	R/W	H'0000	H'FFFF8356	8, 16, 32
Interrupt control register	ICR	R/W	*1	H'FFFF8358	8, 16, 32
IRQ status register	ISR	R(W)*2	H'0000	H'FFFF835A	8, 16, 32

Notes: *1 The value when the NMI pin is high is H'8000; when the NMI pin is low, it is H'0000.
*2 Only 0 can be written, in order to clear flags.

6.2 Interrupt Sources

There are four types of interrupt sources: NMI, user breaks, IRQ, and on-chip peripheral modules. Each interrupt has a priority expressed as a priority level (0 to 16, with 0 the lowest and 16 the highest). Giving an interrupt a priority level of 0 masks it.

6.2.1 NMI Interrupts

The NMI interrupt has priority 16 and is always accepted. Input at the NMI pin is detected by edge. Use the NMI edge select bit (NMIE) in the interrupt control register (ICR) to select either the rising or falling edge. NMI interrupt exception processing sets the interrupt mask level bits (I3–I0) in the status register (SR) to level 15.

6.2.2 User Break Interrupt

A user break interrupt has a priority of level 15, and occurs when the break condition set in the user break controller (UBC) is satisfied. User break interrupt requests are detected by edge and are held until accepted. User break interrupt exception processing sets the interrupt mask level bits (I3–I0) in the status register (SR) to level 15. For more information about the user break interrupt, see section 7, User Break Controller (UBC).

6.2.3 IRQ Interrupts

IRQ interrupts are requested by input from pins $\overline{\text{IRQ0}}$–$\overline{\text{IRQ7}}$. Set the IRQ sense select bits (IRQ0S–IRQ7S) of the interrupt control register (ICR) to select low level detection or falling edge detection for each pin. The priority level can be set from 0 to 15 for each pin using the interrupt priority registers A and B (IPRA–IPRB).

When IRQ interrupts are set to low level detection, an interrupt request signal is sent to the INTC during the period the IRQ pin is low level. Interrupt request signals are not sent to the INTC when the IRQ pin becomes high level. Interrupt request levels can be confirmed by reading the IRQ flags (IRQ0F–IRQ7F) of the IRQ status register (ISR).

When IRQ interrupts are set to falling edge detection, interrupt request signals are sent to the INTC upon detecting a change on the IRQ pin from high to low level. IRQ interrupt request detection results are maintained until the interrupt request is accepted. Confirmation that IRQ interrupt requests have been detected is possible by reading the IRQ flags (IRQ0F–IRQ7F) of the IRQ status register (ISR), and by writing a 0 after reading a 1, IRQ interrupt request detection results can be withdrawn.

In IRQ interrupt exception processing, the interrupt mask bits (I3–I0) of the status register (SR) are set to the priority level value of the accepted IRQ interrupt.

6.2.4 On-Chip Peripheral Module Interrupts

On-chip peripheral module interrupts are interrupts generated by the following on-chip peripheral modules:

- Direct memory access controller (DMAC)
- Multifunction timer/pulse unit (MTU)
- Compare match timer (CMT)
- Serial communications interface (SCI)
- A/D converter (A/D)
- Data transfer controller (DTC)
- Watchdog timer (WDT)
- Bus state controller (BSC)
- I/O port (I/O)

A different interrupt vector is assigned to each interrupt source, so the exception service routine does not have to decide which interrupt has occurred. Priority levels between 0 and 15 can be assigned to individual on-chip peripheral modules in interrupt priority registers C–H (IPRC–IPRH).

On-chip peripheral module interrupt exception processing sets the interrupt mask level bits (I3–I0) in the status register (SR) to the priority level value of the on-chip peripheral module interrupt that was accepted.

6.2.5 Interrupt Exception Vectors and Priority Rankings

Table 6.3 lists interrupt sources and their vector numbers, vector table address offsets and interrupt priorities.

Each interrupt source is allocated a different vector number and vector table address offset. Vector table addresses are calculated from vector numbers and address offsets. In interrupt exception processing, the exception service routine start address is fetched from the vector table indicated by the vector table address. See table 5.4, Calculating Exception Processing Vector Table Addresses.

IRQ interrupts and on-chip peripheral module interrupt priorities can be set freely between 0 and 15 for each pin or module by setting interrupt priority registers A–H (IPRA–IPRH). The ranking of interrupt sources for IPRC–IPRH, however, must be the order listed under Priority Order Within IPR Setting Range in table 6.3 and cannot be changed. A power-on reset assigns priority level 0 to IRQ interrupts and on-chip peripheral module interrupts. If the same priority level is assigned to two or more interrupt sources and interrupts from those sources occur simultaneously, their priority order is the default priority order indicated at the right in table 6.3.

Table 6.3 Interrupt Exception Processing Vectors and Priorities

Interrupt Source		Interrupt Vector		Interrupt Priority (Initial Value)	Corresponding IPR (Bits)	Priority within IPR Setting Range	Default Priority
		Vector No.	Vector Table Address Offset				
NMI		11	H'0000002C–H'0000002F	16	—	—	High
User break		12	H'00000030–H'00000033	15	—	—	
IRQ0		64	H'00000100–H'00000103	0–15 (0)	IPRA (15–12)	—	
IRQ1		65	H'00000104–H'00000107	0–15 (0)	IPRA (11–8)	—	
IRQ2		66	H'00000108–H'0000010B	0–15 (0)	IPRA (7–4)	—	
IRQ3		67	H'0000010C–H'0000010F	0–15 (0)	IPRA (3–0)	—	
IRQ4		68	H'00000110–H'00000113	0–15 (0)	IPRB (15–12)	—	
IRQ5		69	H'00000114–H'00000117	0–15 (0)	IPRB (11–8)	—	
IRQ6		70	H'00000118–H'0000011B	0–15 (0)	IPRB (7–4)	—	
IRQ7		71	H'0000011C–H'0000011F	0–15 (0)	IPRB (3–0)	—	
DMAC0	DEI0	72	H'00000120–H'00000123	0–15 (0)	IPRC (15–12)	—	
DMAC1	DEI1	76	H'00000130–H'00000133	0–15 (0)	IPRC (11–8)	—	
DMAC2	DEI2	80	H'00000140–H'00000143	0–15 (0)	IPRC (7–4)	—	
DMAC3	DEI3	84	H'00000150–H'00000153	0–15 (0)	IPRC (3–0)	—	Low

Table 6.3 Interrupt Exception Processing Vectors and Priorities (cont)

Interrupt Source		Interrupt Vector		Interrupt Priority (Initial Value)	Corresponding IPR (Bits)	Priority within IPR Setting Range	Default Priority
		Vector No.	Vector Table Address Offset				
MTU0	TGI0A	88	H'00000160–H'00000163	0–15 (0)	IPRD (15–12)	High	High
	TGI0B	89	H'00000164–H'00000167	0–15 (0)			
	TGI0C	90	H'00000168–H'0000016B	0–15 (0)			
	TGI0D	91	H'0000016C–H'0000016F	0–15 (0)		Low	
	TCI0V	92	H'00000170–H'00000173	0–15 (0)	IPRD (11–8)	—	
MTU1	TGI1A	96	H'00000180–H'00000183	0–15 (0)	IPRD (7–4)	High	
	TGI1B	97	H'00000184–H'00000187	0–15 (0)		Low	
	TCI1V	100	H'00000190–H'00000193	0–15 (0)	IPRD (3–0)	High	
	TCI1U	101	H'00000194–H'00000197	0–15 (0)		Low	
MTU2	TGI2A	104	H'000001A0–H'000001A3	0–15 (0)	IPRE (15–12)	High	
	TGI2B	105	H'000001A4–H'000001A7	0–15 (0)		Low	
	TCI2V	108	H'000001B0–H'000001B3	0–15 (0)	IPRE (11–8)	High	
	TCI2U	109	H'000001B4–H'000001B7	0–15 (0)		Low	Low

Table 6.3 Interrupt Exception Processing Vectors and Priorities (cont)

Interrupt Source		Interrupt Vector		Interrupt Priority (Initial Value)	Corresponding IPR (Bits)	Priority within IPR Setting Range	Default Priority
		Vector No.	Vector Table Address Offset				
MTU3	TGI3A	112	H'000001C0–H'000001C3	0–15 (0)	IPRE (7–4)	High	High
	TGI3B	113	H'000001C4–H'000001C7	0–15 (0)			
	TGI3C	114	H'000001C8–H'000001CB	0–15 (0)			
	TGI3D	115	H'000001CC–H'000001CF	0–15 (0)		Low	
	TCI3V	116	H'000001D0–H'000001D3	0–15 (0)	IPRE (3–0)	—	
MTU4	TGI4A	120	H'000001E0–H'000001E3	0–15 (0)	IPRF (15–12)	High	
	TGI4B	121	H'000001E4–H'000001E7	0–15 (0)			
	TGI4C	122	H'000001E8–H'000001EB	0–15 (0)			
	TGI4D	123	H'000001EC–H'000001EF	0–15 (0)		Low	
	TCI4V	124	H'000001F0–H'000001F3	0–15 (0)	IPRF (11–8)	High	
	Reserved	125	H'000001F4–H'000001F7	0–15 (0)		Low	
SCI0	ERI0	128	H'00000200–H'00000203	0–15 (0)	IPRF (7–4)	High	
	RXI0	129	H'00000204–H'00000207	0–15 (0)			
	TXI0	130	H'00000208–H'0000020B	0–15 (0)			
	TEI0	131	H'0000020C–H'0000020F	0–15 (0)		Low	Low

Table 6.3 Interrupt Exception Processing Vectors and Priorities (cont)

Interrupt Source		Interrupt Vector		Interrupt Priority (Initial Value)	Corresponding IPR (Bits)	Priority within IPR Setting Range	Default Priority
		Vector No.	Vector Table Address Offset				
SCI1	ERI1	132	H'00000210–H'00000213	0–15 (0)	IPRF (3–0)	High	High
	RXI1	133	H'00000214–H'00000217	0–15 (0)			
	TXI1	134	H'00000218–H'0000021B	0–15 (0)			
	TEI1	135	H'0000021C–H'0000021F	0–15 (0)		Low	
A/D*	ADI	136	H'00000220–H'00000223	0–15 (0)	IPRG (15–12)	—	
DTC	SWDTCE	140	H'00000230–H'00000233	0–15 (0)	IPRG (11–8)	—	
CMT0	CMI0	144	H'00000240–H'00000243	0–15 (0)	IPRG (7–4)	—	
CMT1	CMI1	148	H'00000250–H'00000253	0–15 (0)	IPRG (3–0)	—	
WDT	ITI	152	H'00000260–H'00000263	0–15 (0)	IPRH (15–12)	High	
BSC	CMI	153	H'00000264–H'00000267	0–15 (0)		Low	
I/O	OEI	156	H'00000270–H'00000273	0–15 (0)	IPRH (11–8)	—	Low

Note: * For A mask products, A/D is as follows

A/D	ADI0	136	H'00000220–H'00000223	0–15 (0)	IPRG (15–12)	High	
	ADI1	137	H'00000224–H'00000227	0–15 (0)		Low	

6.3 Description of Registers

6.3.1 Interrupt Priority Registers A–H (IPRA–IPRH)

Interrupt priority registers A–H (IPRA–IPRH) are 16-bit readable/writable registers that set priority levels from 0 to 15 for IRQ interrupts and on-chip peripheral module interrupts. Correspondence between interrupt request sources and each of the IPRA–IPRH bits is shown in table 6.4.

Bit:	15	14	13	12	11	10	9	8
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 6.4 Interrupt Request Sources and IPRA–IPRH

	Bits			
Register	**15–12**	**11–8**	**7–4**	**3–0**
Interrupt priority register A	IRQ0	IRQ1	IRQ2	IRQ3
Interrupt priority register B	IRQ4	IRQ5	IRQ6	IRQ7
Interrupt priority register C	DMAC0	DMAC1	DMAC2	DMAC3
Interrupt priority register D	MTU0	MTU0	MTU1	MTU1
Interrupt priority register E	MTU2	MTTU2	MTU3	MTU3
Interrupt priority register F	MTU4	MTU4	SCI0	SCI1
Interrupt priority register G	A/D(A/D0, A/D1)*	DTC	CMT0	CMT1
Interrupt priority register H	WDT, BSC	I/O	Reserved	Reserved

Note: * Excluding A mask products are A/D, A mask products are A/D0 and A/D1.

As indicated in table 6.4, four $\overline{\text{IRQ}}$ pins or groups of 4 on-chip peripheral modules are allocated to each register. Each of the corresponding interrupt priority ranks are established by setting a value from H'0 (0000) to H'F (1111) in each of the four-bit groups 15–12, 11–8, 7–4, and 3–0. Interrupt priority rank becomes level 0 (lowest) by setting H'0, and level 15 (highest) by setting H'F. If multiple on-chip peripheral modules are assigned to WDT and BSC, those multiple modules are set to the same priority rank.

IPRA–IPRH are initialized to H'0000 by a power-on reset or a manual reset. They are not initialized in standby mode.

6.3.2 Interrupt Control Register (ICR)

The ICR is a 16-bit register that sets the input signal detection mode of the external interrupt input pin NMI and $\overline{\text{IRQ0}}$ –$\overline{\text{IRQ7}}$ and indicates the input signal level to the NMI pin. A power-on reset initializes ICR but the standby mode does not.

Bit:	15	14	13	12	11	10	9	8
	NMIL	—	—	—	—	—	—	NMIE
Initial value:	*	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W

Bit:	7	6	5	4	3	2	1	0
	IRQ0S	IRQ1S	IRQ2S	IRQ3S	IRQ4S	IRQ5S	IRQ6S	IRQ7S
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: * When NMI input is high: 1; when NMI input is low: 0

- Bit 15—NMI Input Level (NMIL): Sets the level of the signal input at the NMI pin. This bit can be read to determine the NMI pin level. This bit cannot be modified.

Bit 15: NMIL	Description
0	NMI input level is low
1	NMI input level is high

- Bits 14–9—Reserved: These bits always read as 0. The write value should always be 0.

- Bit 8—NMI Edge Select (NMIE)

Bit 8: NMIE	Description
0	Interrupt request is detected on falling edge of NMI input (initial value)
1	Interrupt request is detected on rising edge of NMI input

- Bits 7–0—IRQ0–IRQ7 Sense Select (IRQ0S–IRQ7S): These bits set the IRQ0–IRQ7 interrupt request detection mode.

Bits 7-0: IRQ0S–IRQ7S	Description
0	Interrupt request is detected on low level of IRQ input (initial value)
1	Interrupt request is detected on falling edge of IRQ input

6.3.3 IRQ Status Register (ISR)

The ISR is a 16-bit register that indicates the interrupt request status of the external interrupt input pins $\overline{\text{IRQ0}}$–$\overline{\text{IRQ7}}$. When IRQ interrupts are set to edge detection, held interrupt requests can be withdrawn by writing a 0 to IRQnF after reading an IRQnF = 1.

A power-on reset initializes ISR but the standby mode does not.

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	IRQ0F	IRQ1F	IRQ2F	IRQ3F	IRQ4F	IRQ5F	IRQ6F	IRQ7F
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- Bits 15–8—Reserved: These bits always read as 0. The write value should always be 0.

- Bits 7–0—IRQ0–IRQ7 Flags (IRQ0F–IRQ7F): These bits display the IRQ0–IRQ7 interrupt request status.

Bits 7-0: IRQ0F–IRQ7F	Detection Setting	Description
0	Level detection	No IRQn interrupt request exists. Clear conditions: When $\overline{\text{IRQn}}$ input is high level
	Edge detection	No IRQn interrupt request was detected. (initial value) Clear conditions: 1. When a 0 is written after reading IRQnF = 1 status 2. When IRQn interrupt exception processing has been executed 3. When a DTC transfer due to IRQn interrupt has been executed
1	Level detection	An IRQn interrupt request exists. Set conditions: When $\overline{\text{IRQn}}$ input is low level
	Edge detection	An IRQn interrupt request was detected. Set conditions: When a falling edge occurs at an $\overline{\text{IRQn}}$ input

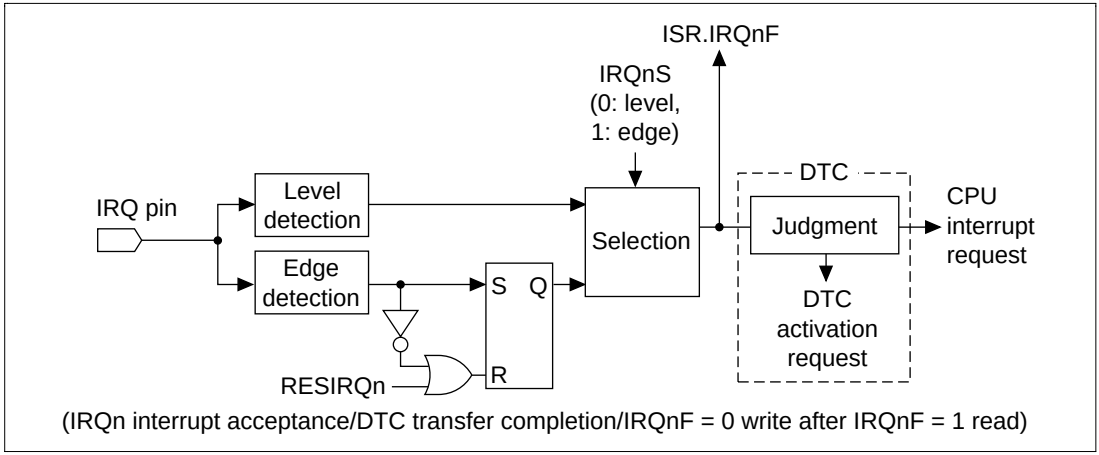


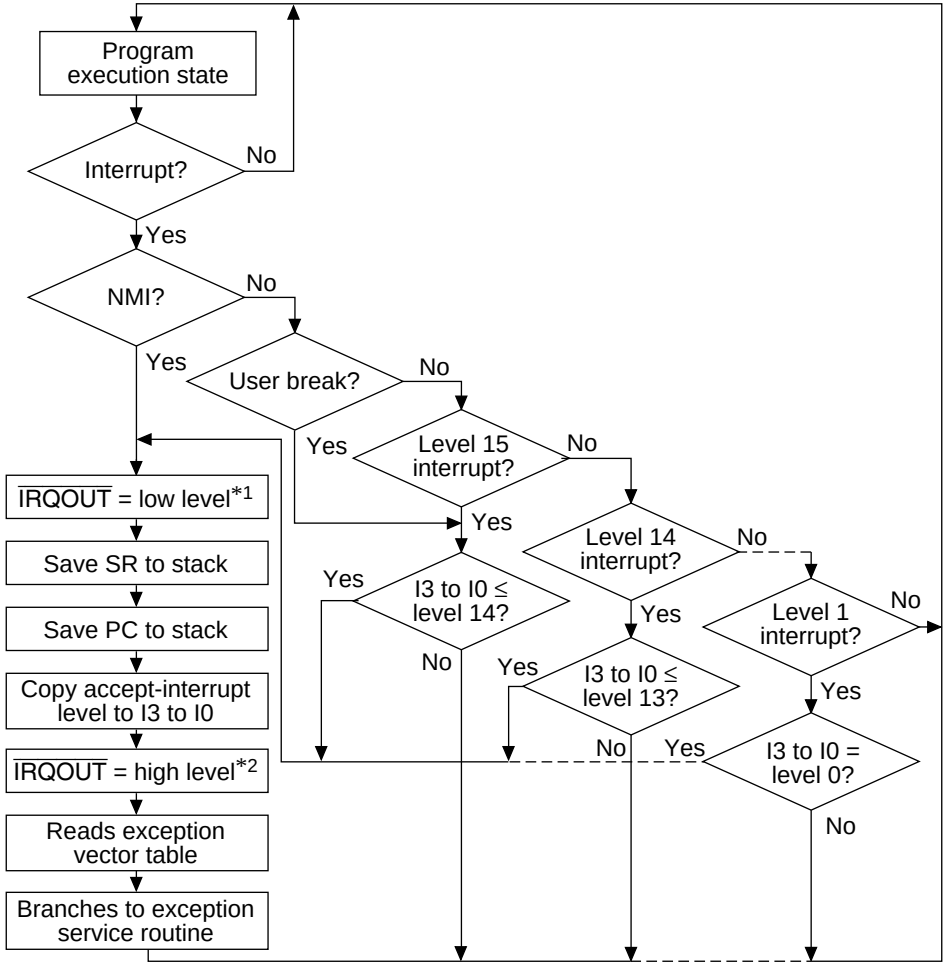
Figure 6.2 External Interrupt Process

6.4 Interrupt Operation

6.4.1 Interrupt Sequence

The sequence of interrupt operations is explained below. Figure 6.3 is a flowchart of the operations.

1. The interrupt request sources send interrupt request signals to the interrupt controller.
2. The interrupt controller selects the highest priority interrupt in the interrupt requests sent, following the priority levels set in interrupt priority level setting registers A–H (IPRA–IPRH). Lower-priority interrupts are ignored. They are held pending until interrupt requests designated as edge-detect type are accepted. For IRQ interrupts, however, withdrawal is possible by accessing the IRQ status register (ISR). See section 6.2.3, IRQ Interrupts, for details. Interrupts held pending due to edge detection are cleared by a power-on reset or a manual reset. If two of these interrupts have the same priority level or if multiple interrupts occur within a single module, the interrupt with the highest default priority or the highest priority within its IPR setting range (as indicated in table 6.3) is selected.
3. The interrupt controller compares the priority level of the selected interrupt request with the interrupt mask bits (I3–I0) in the CPU's status register (SR). If the request priority level is equal to or less than the level set in I3–I0, the request is ignored. If the request priority level is higher than the level in bits I3–I0, the interrupt controller accepts the interrupt and sends an interrupt request signal to the CPU.
4. When the interrupt controller accepts an interrupt, a low level is output from the $\overline{\text{IRQOUT}}$ pin.
5. The CPU detects the interrupt request sent from the interrupt controller when it decodes the next instruction to be executed. Instead of executing the decoded instruction, the CPU starts interrupt exception processing (figure 6.4).
6. SR and PC are saved onto the stack.
7. The priority level of the accepted interrupt is copied to the interrupt mask level bits (I3–I0) in the status register (SR).
8. When the accepted interrupt is sensed by level or is from an on-chip peripheral module, a high level is output from the $\overline{\text{IRQOUT}}$ pin. When the accepted interrupt is sensed by edge, a high level is output from the $\overline{\text{IRQOUT}}$ pin at the point when the CPU starts interrupt exception processing instead of instruction execution as noted in (5) above. However, if the interrupt controller accepts an interrupt with a higher priority than one it is in the midst of accepting, the $\overline{\text{IRQOUT}}$ pin will remain low level.
9. The CPU reads the start address of the exception service routine from the exception vector table for the accepted interrupt, jumps to that address, and starts executing the program there. This jump is not a delay branch.



I3 to I0: Interrupt mask bits of status register

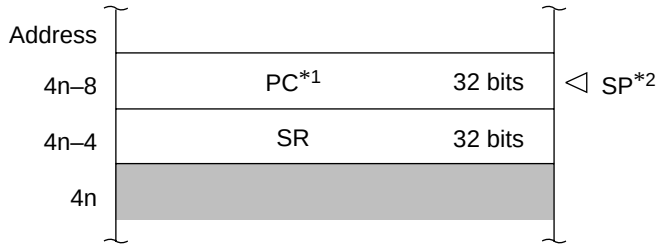
Notes: *1 $\overline{\text{IRQOUT}}$ is the same signal as the interrupt request signal to the CPU (see figure 6.1). Thus, it is output when there is a higher priority interrupt request than the one in the I3 to I0 bits of the SR.

*2 When the accepted interrupt is sensed by edge, the $\overline{\text{IRQOUT}}$ pin becomes high level at the point when the CPU starts interrupt exception processing instead of instruction execution (before SR is saved to the stack).
If the interrupt controller has accepted another interrupt with a higher priority and has output an interrupt request to the CPU, the $\overline{\text{IRQOUT}}$ pin will remain low level.

Figure 6.3 Interrupt Sequence Flowchart

6.4.2 Stack after Interrupt Exception Processing

Figure 6.4 shows the stack after interrupt exception processing.



Notes: *1 PC: Start address of the next instruction (return destination instruction) after the executing instruction
*2 Always be certain that SP is a multiple of 4

Figure 6.4 Stack after Interrupt Exception Processing

6.5 Interrupt Response Time

Table 6.5 indicates the interrupt response time, which is the time from the occurrence of an interrupt request until the interrupt exception processing starts and fetching of the first instruction of the interrupt service routine begins. Figure 6.5 shows the pipeline when an IRQ interrupt is accepted.

Table 6.5 Interrupt Response Time

Item		Number of States: NMI, Peripheral Module	Number of States: IRQ	Notes
DMAC/DTC active judgment		0 or 1	1	1 state required for interrupt signals for which DMAC/DTC activation is possible
Compare identified interrupt priority with SR mask level		2	3	
Wait for completion of sequence currently being executed by CPU		X (≥ 0)		The longest sequence is for interrupt or address-error exception processing (X = 4 + m1 + m2 + m3 + m4). If an interrupt-masking instruction follows, however, the time may be even longer.
Time from start of interrupt exception processing until fetch of first instruction of exception service routine starts		5 + m1 + m2 + m3		Performs the PC and SR saves and vector address fetch.
Interrupt response time	Total:	7 + m1 + m2 + m3	9 + m1 + m2 + m3	
	Minimum:	10	12	0.35–0.42 µs at 28.7 MHz
	Maximum:	12 + 2 (m1 + m2 + m3) + m4	13 + 2 (m1 + m2 + m3) + m4	0.67–0.70 µs at 28.7 MHz*

Note: * When m1 = m2 = m3 = m4 = 1

m1–m4 are the number of states needed for the following memory accesses.

m1: SR save (longword write)

m2: PC save (longword write)

m3: Vector address read (longword read)

m4: Fetch first instruction of interrupt service routine

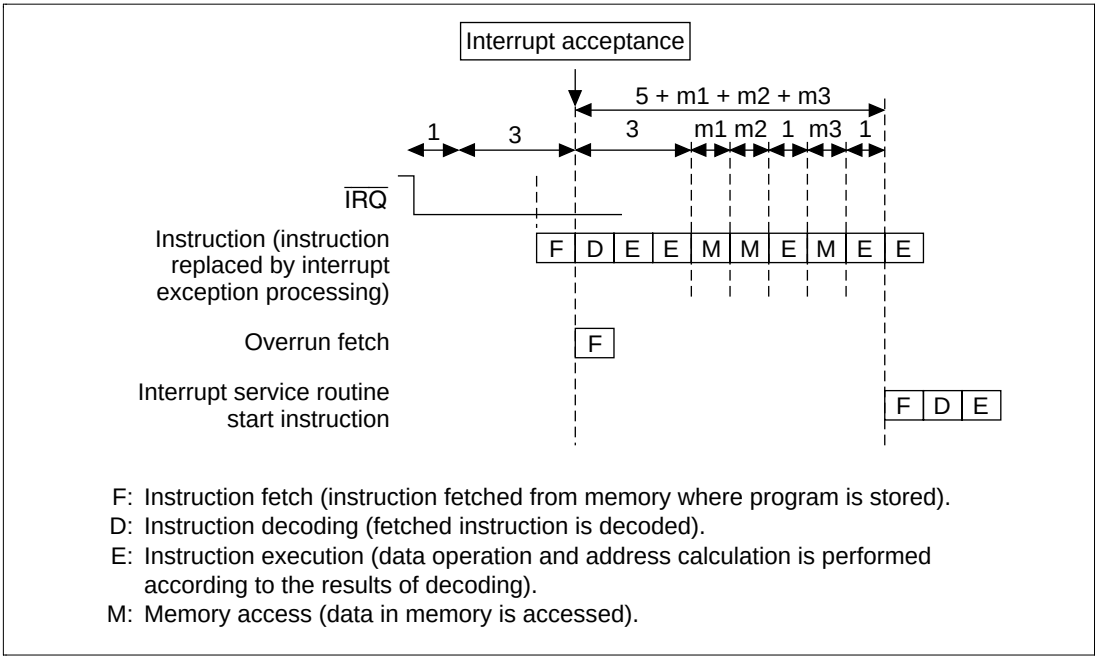


Figure 6.5 Pipeline when an IRQ Interrupt is Accepted

6.6 Data Transfer with Interrupt Request Signals

The following data transfers can be done using interrupt request signals:

- Activate DMAC only, without generating CPU interrupt
- Activate DTC only, CPU interrupts according to DTC settings

Among interrupt sources, those designated as DMAC activating sources are masked and not input to the INTC. The masking condition is listed below:

Mask condition = DME • (DE0 • source selection 0 + DE1 × source selection 1 + DE2 • source selection 2 + DE3 • source selection 3)

The INTC masks CPU interrupts when the corresponding DTE bit is a 1. The DTE clear condition and interrupt source flag clear condition are listed below.

DTE clear condition = DTC transfer end • DTECLR

Interrupt source flag clear condition = DTC transfer end • $\overline{\text{DTECLR}}$ + DMAC transfer end

Where: DTECLR = DISEL + counter 0.

Figure 6.6 shows a control block diagram.

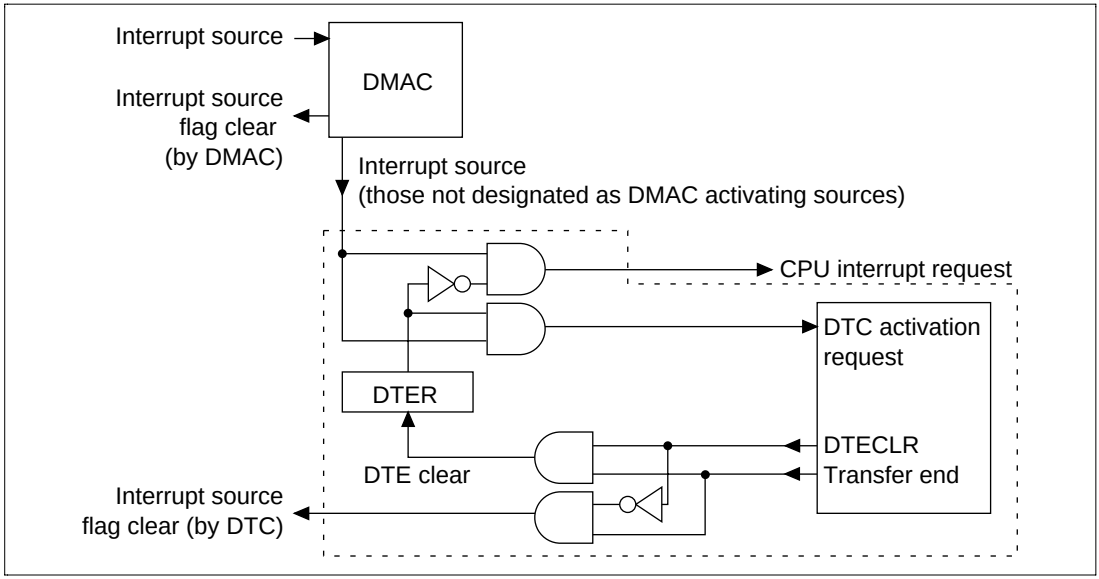


Figure 6.6 Interrupt Control Block Diagram

6.6.1 Handling DTC Activating and CPU Interrupt Sources, but Not DMAC Activating Sources

1. Either do not select the DMAC as a source, or clear the DME bit to 0.
2. For DTC, set the corresponding DTE bits and DISEL bits to 1.
3. Activating sources are applied to the DTC when interrupts occur.
4. When the DTC performs a data transfer, it clears the DTE bit to 0 and sends an interrupt request to the CPU. The activating source does not clear.
5. The CPU clears interrupt sources with its interrupt processing routine. It then confirms the transfer counter value. When the transfer counter value ≠ 0, it sets the DTE bit to 1 and allows the next data transfer. If the transfer counter value = 0, it performs the necessary end processing in the interrupt processing routine.

6.6.2 Handling DMAC Activating Sources but Not CPU Interrupt or DTC Activating Sources

1. Select the DMAC as a source and set the DME bit to 1. CPU interrupt sources and DTC activating sources are masked regardless of the interrupt priority level register settings or DTC register settings.
2. Activating sources are applied to the DMAC when interrupts occur.
3. The DMAC clears activating sources at the time of data transfer.

6.6.3 Handling DTC Activating Sources but Not CPU Interrupt or DMAC Activating Sources

1. Either do not select the DMAC as a source, or clear the DME bit to 0.
2. For DTC, set the corresponding DTE bits to 1 and clear the DISEL bits to 0.
3. Activating sources are applied to the DTC when interrupts occur.
4. When the DTC performs a data transfer, it clears the activating source. An interrupt request is not sent to the CPU, because the DTE bit is maintained as a 1.
5. However, when the transfer counter value = 0 the DTE bit is cleared to 0 and an interrupt request is sent to the CPU.
6. The CPU performs the necessary end processing in the interrupt processing routine.

6.6.4 Treating CPU Interrupt Sources but Not DTC or DMAC Activating Sources

1. Either do not select the DMAC as a source, or clear the DME bit to 0.
2. For DTC, clear the corresponding DTE bits to 0.
3. When interrupts occur, interrupt requests are sent to the CPU.
4. The CPU clears the interrupt source and performs the necessary processing in the interrupt processing routine.

Section 7 User Break Controller (UBC)

7.1 Overview

The user break controller (UBC) provides functions that simplify program debugging. Break conditions are set in the UBC and a user break interrupt is generated according to the conditions of the bus cycle generated by the CPU, DMAC, or DTC. This function makes it easy to design an effective self-monitoring debugger, enabling the chip to easily debug programs without using a large in-circuit emulator.

7.1.1 Features

The features of the user break controller are:

- Break compare conditions can be set:
  - Address
  - CPU cycle or DMA/DTC cycle
  - Instruction fetch or data access
  - Read or write
  - Operand size: byte/word/longword
- User break interrupt generated upon satisfying break conditions. A user-designed user break interrupt exception processing routine can be run.
- Select either to break in the CPU instruction fetch cycle before the instruction is executed or after.

7.1.2 Block Diagram

Figure 7.1 shows a block diagram of the UBC.

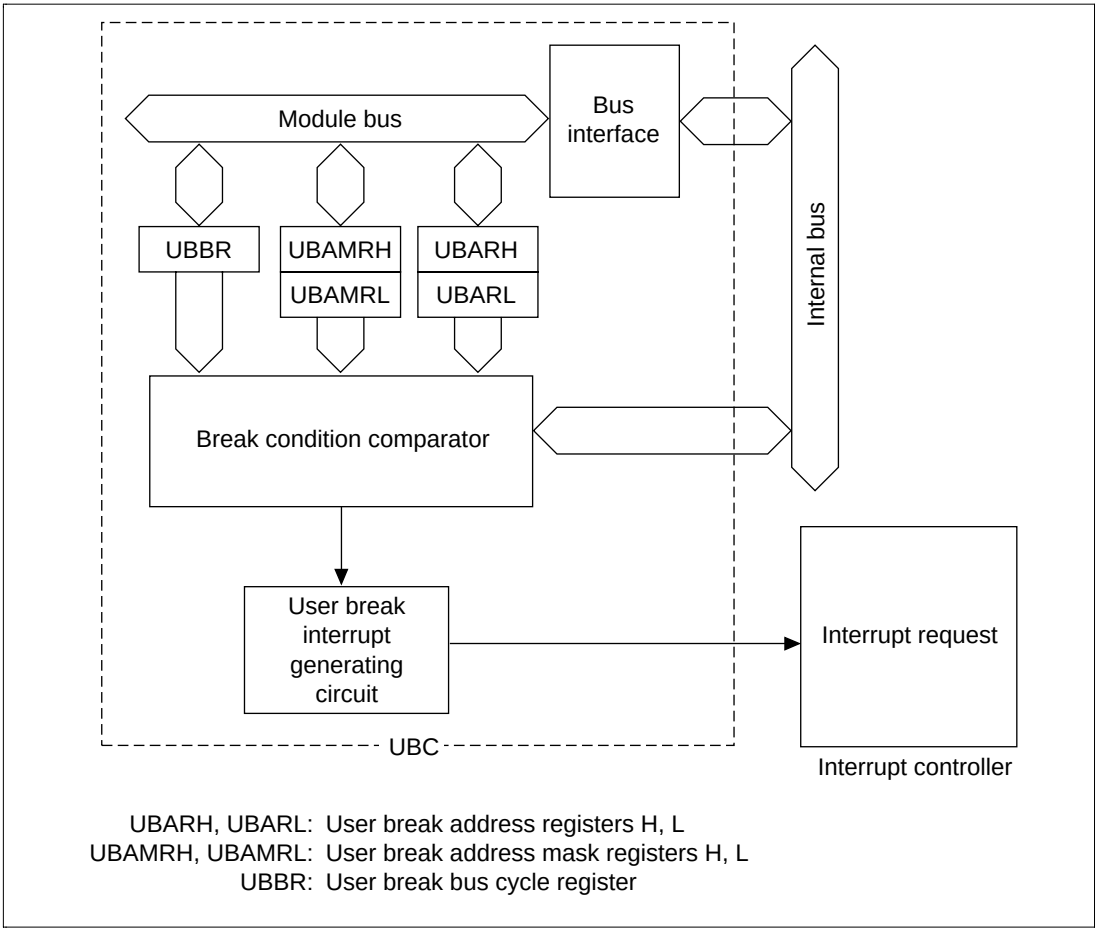


Figure 7.1 User Break Controller Block Diagram

7.1.3 Register Configuration

The UBC has the five registers shown in table 7.1. Break conditions are established using these registers.

Table 7.1 Register Configuration

Name	Abbr.	R/W	Initial Value	Address	Access Size
User break address register H	UBARH	R/W	H'0000	H'FFFF8600	8, 16, 32
User break address register L	UBARL	R/W	H'0000	H'FFFF8602	8, 16, 32
User break address mask register H	UBAMRH	R/W	H'0000	H'FFFF8604	8, 16, 32
User break address mask register L	UBAMRL	R/W	H'0000	H'FFFF8606	8, 16, 32
User break bus cycle register	UBBR	R/W	H'0000	H'FFFF8608	8, 16, 32

7.2 Register Descriptions

7.2.1 User Break Address Register (UBAR)

The user break address register (UBAR) consists of user break address register H (UBARH) and user break address register L (UBARL). Both are 16-bit readable/writable registers. UBARH stores the upper bits (bits 31–16) of the address of the break condition, while UBARL stores the lower bits (bits 15–0). Resets and hardware standbys initialize both UBARH and UBARL to H'0000. They are not initialized in manual reset or software standby mode.

UBARH:

Bit:	15	14	13	12	11	10	9	8
UBARH	UBA31	UBA30	UBA29	UBA28	UBA27	UBA26	UBA25	UBA24
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
UBARH	UBA23	UBA22	UBA21	UBA20	UBA19	UBA18	UBA17	UBA16
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

UBARL:

Bit:	15	14	13	12	11	10	9	8
UBARL	UBA15	UBA14	UBA13	UBA12	UBA11	UBA10	UBA9	UBA8
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
UBARL	UBA7	UBA6	UBA5	UBA4	UBA3	UBA2	UBA1	UBA0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- UBARH Bits 15–0—User Break Address 31–16 (UBA31–UBA16): These bits store the upper bit values (bits 31–16) of the address of the break condition.
- UBARL Bits 15–0—User Break Address 15–0 (UBA15–UBA0): These bits store the lower bit values (bits 15–0) of the address of the break condition.

7.2.2 User Break Address Mask Register (UBAMR)

The user break address mask register (UBAMR) consists of user break address mask register H (UBAMRH) and user break address mask register L (UBAMRL). Both are 16-bit readable/writable registers. UBAMRH designates whether to mask any of the break address bits established in the UBARH, and UBAMRL designates whether to mask any of the break address bits established in the UBARL. Resets and hardware standbys initialize both UBAMRH and UBAMRL to H'0000. They are not initialized in manual reset or software standby mode.

UBAMRH:

Bit:	15	14	13	12	11	10	9	8
UBAMRH	UBM31	UBM30	UBM29	UBM28	UBM27	UBM26	UBM25	UBM24
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
UBAMRH	UBM23	UBM22	UBM21	UBM20	UBM19	UBM18	UBM17	UBM16
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

UBAMRL:

Bit:	15	14	13	12	11	10	9	8
UBAMRL	UBM15	UBM14	UBM13	UBM12	UBM11	UBM10	UBM9	UBM8
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
UBAMRL	UBM7	UBM6	UBM5	UBM4	UBM3	UBM2	UBM1	UBM0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- UBAMRH Bits 15–0—User Break Address Mask 31–16 (UBM31–UBM16): These bits designate whether to mask any of the break address 31–16 bits (UBA31–UBA16) established in the UBARH.
- UBAMRL Bits 15–0—User Break Address Mask 15–0 (UBM15–UBM0): These bits designate whether to mask any of the break address 15–0 bits (UBA15–UBA0) established in the UBARL.

Bits 15–0: UBMn	Description
0	Break address UBAn is included in the break conditions (initial value)
1	Break address UBAn is not included in the break conditions

Note: n = 31–0

7.2.3 User Break Bus Cycle Register (UBBR)

User break bus cycle register (UBBR) is a 16-bit readable/writable register that selects from among the following four break conditions:

1. CPU cycle/ DMAC/DTC cycle
2. Instruction fetch/data access
3. Read/write
4. Operand size (byte, word, longword)

Resets and hardware standbys initialize the UBBR to H'0000. It is not initialized in software standby mode.

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	CP1	CP0	ID1	ID0	RW1	RW0	SZ1	SZ0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 15–8—Reserved: These bits always read as 0. The write value should always be 0.

- Bits 7 and 6—CPU Cycle/Peripheral Cycle Select (CP1, CP0): These bits designate break conditions for CPU cycles or peripheral cycles (DMA/DTC cycles).

Bit 7: CP1	Bit 6: CP0	Description
0	0	No user break interrupt occurs (initial value)
	1	Break on CPU cycles
1	0	Break on peripheral cycles
	1	Break on both CPU and peripheral cycles

- Bits 5 and 4—Instruction Fetch/Data Access Select (ID1, ID0): These bits select whether to break on instruction fetch and/or data access cycles.

Bit 5: ID1	Bit 4: ID0	Description
0	0	No user break interrupt occurs (initial value)
	1	Break on instruction fetch cycles
1	0	Break on data access cycles
	1	Break on both instruction fetch and data access cycles

- Bits 3 and 2—Read/Write Select (RW1, RW0): These bits select whether to break on read and/or write cycles.

Bit 3: RW1	Bit 2: RW0	Description
0	0	No user break interrupt occurs (initial value)
	1	Break on read cycles
1	0	Break on write cycles
	1	Break on both read and write cycles

- Bits 1 and 0—Operand Size Select (SZ1, SZ0): These bits select operand size as a break condition.

Bit 1: SZ1	Bit 0: SZ0	Description
0	0	Operand size is not a break condition (initial value)
	1	Break on byte access
1	0	Break on word access
	1	Break on longword access

Note: When breaking on an instruction fetch, set the SZ0 bit to 0. All instructions are considered to be word-size accesses (even when there are instructions in on-chip memory and 2 instruction fetches are done simultaneously in 1 bus cycle).

Operand size is word for instructions or determined by the operand size specified for the CPU/DMAC data access. It is not determined by the bus width of the space being accessed.

7.3 Operation

7.3.1 Flow of the User Break Operation

The flow from setting of break conditions to user break interrupt exception processing is described below:

1. The user break addresses are set in the user break address register (UBAR), the desired masked bits in the addresses are set in the user break address mask register (UBAMR) and the breaking bus cycle type is set in the user break bus cycle register (UBBR). If even one of the three groups of the UBBR's CPU cycle/peripheral cycle select bits (CP1, CP0), instruction fetch/data access select bits (ID1, ID0), and read/write select bits (RW1, RW0) is set to 00 (no user break interrupt is generated), no user break interrupt will be generated even if all other conditions are in agreement. When using user break interrupts, always be certain to establish bit conditions for all of these three groups.
2. The UBC uses the method shown in figure 7.2 to judge whether set conditions have been fulfilled. When the set conditions are satisfied, the UBC sends a user break interrupt request signal to the interrupt controller (INTC).
3. The interrupt controller checks the accepted user break interrupt request signal's priority level. The user break interrupt has priority level 15, so it is accepted only if the interrupt mask level in bits I3–I0 in the status register (SR) is 14 or lower. When the I3–I0 bit level is 15, the user break interrupt cannot be accepted but it is held pending until user break interrupt exception processing can be carried out. Consequently, user break interrupts within NMI exception service routines cannot be accepted, since the I3–I0 bit level is 15. However, if the I3–I0 bit level is changed to 14 or lower at the start of the NMI exception service routine, user break interrupts become acceptable thereafter. Section 6, Interrupt Controller (INTC), describes the handling of priority levels in greater detail.
4. The INTC sends the user break interrupt request signal to the CPU, which begins user break interrupt exception processing upon receipt. See Section 6.4, Interrupt Operation, for details on interrupt exception processing.

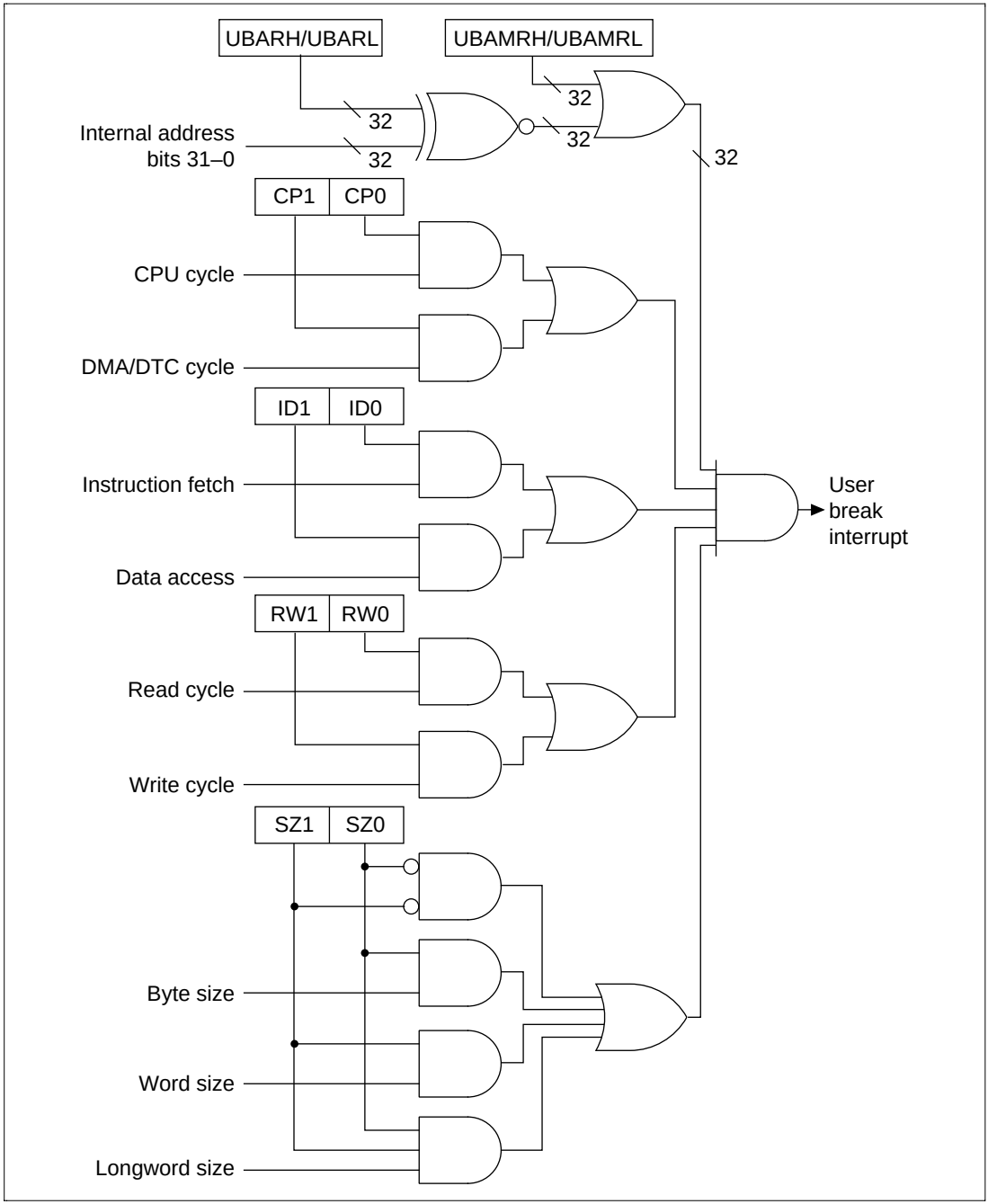


Figure 7.2 Break Condition Judgment Method

7.3.2 Break on On-Chip Memory Instruction Fetch Cycle

On-chip memory (on-chip ROM and/or RAM) is always accessed as 32 bits in 1 bus cycle. Therefore, 2 instructions can be retrieved in 1 bus cycle when fetching instructions from on-chip memory. At such times, only 1 bus cycle is generated, but by setting the start addresses of both instructions in the user break address register (UBAR) it is possible to cause independent breaks. In other words, when wanting to effect a break using the latter of two addresses retrieved in 1 bus cycle, set the start address of that instruction in UBAR. The break will occur after execution of the former instruction.

7.3.3 Program Counter (PC) Values Saved

Break on Instruction Fetch (Before Execution): The program counter (PC) value saved to the stack in user break interrupt exception processing is the address that matches the break condition. The user break interrupt is generated before the fetched instruction is executed. If a break condition is set in an instruction fetch cycle placed immediately after a delayed branch instruction (delay slot), or on an instruction that follows an interrupt-disabled instruction, however, the user break interrupt is not accepted immediately, but the break condition establishing instruction is executed. The user break interrupt is accepted after execution of the instruction that has accepted the interrupt. In this case, the PC value saved is the start address of the instruction that will be executed after the instruction that has accepted the interrupt.

Break on Data Access (CPU/Peripheral): The program counter (PC) value is the top address of the next instruction after the last instruction executed before the user break exception processing started. When data access (CPU/peripheral) is set as a break condition, the place where the break will occur cannot be specified exactly. The break will occur at the instruction fetched close to where the data access that is to receive the break occurs.

7.4 Use Examples

7.4.1 Break on CPU Instruction Fetch Cycle

1. Register settings: UBARH = H'0000
   UBARL = H'0404
   UBBR = H'0054

   Conditions set: Address: H'00000404
   Bus cycle: CPU, instruction fetch, read
   (operand size not included in conditions)

A user break interrupt will occur before the instruction at address H'00000404. If it is possible for the instruction at H'00000402 to accept an interrupt, the user break exception processing will be executed after execution of that instruction. The instruction at H'00000404 is not executed. The PC value saved is H'00000404.

2. Register settings: UBARH = H'0015
   UBARL = H'389C
   UBBR = H'0058
   Conditions set: Address: H'0015389C
   Bus cycle: CPU, instruction fetch, write
   (operand size not included in conditions)

A user break interrupt does not occur because the instruction fetch cycle is not a write cycle.

3. Register settings: UBARH = H'0003
   UBARL = H'0147
   UBBR = H'0054
   Conditions set: Address: H'00030147
   Bus cycle: CPU, instruction fetch, read
   (operand size not included in conditions)

A user break interrupt does not occur because the instruction fetch was performed for an even address. However, if the first instruction fetch address after the branch is an odd address set by these conditions, user break interrupt exception processing will be done after address error exception processing.

7.4.2 Break on CPU Data Access Cycle

1. Register settings: UBARH = H'0012
   UBARL = H'3456
   UBBR = H'006A
   Conditions set: Address: H'00123456
   Bus cycle: CPU, data access, write, word

A user break interrupt occurs when word data is written into address H'00123456.

2. Register settings: UBARH = H'00A8
   UBARL = H'0391
   UBBR = H'0066
   Conditions set: Address: H'00A80391
   Bus cycle: CPU, data access, read, word

A user break interrupt does not occur because the word access was performed on an even address.

7.4.3 Break on DMA/DTC Cycle

1. Register settings: UBARH = H'0076
   UBARL = H'BCDC
   UBBR = H'00A7
   Conditions set: Address: H'0076BCDC
   Bus cycle: DMA/DTC, data access, read, longword

A user break interrupt occurs when longword data is read from address H'0076BCDC.

2. Register settings: UBARH = H'0023
   UBARL = H'45C8
   UBBR = H'0094
   Conditions set: Address: H'002345C8
   Bus cycle: DMA/DTC, instruction fetch, read
   (operand size not included in conditions)

A user break interrupt does not occur because no instruction fetch is performed in the DMA/DTC cycle.

7.5 Cautions on Use

7.5.1 On-Chip Memory Instruction Fetch

Two instructions are simultaneously fetched from on-chip memory. If a break condition is set on the second of these two instructions but the contents of the UBC break condition registers are changed so as to alter the break condition immediately after the first of the two instructions is fetched, a user break interrupt will still occur when the second instruction is fetched.

7.5.2 Instruction Fetch at Branches

When a conditional branch instruction or TRAPA instruction causes a branch, instructions are fetched and executed as follows:

1. Conditional branch instruction, branch taken: BT, BF

   Instruction fetch cycles: Conditional branch instruction fetch → Next-instruction overrun fetch → Next-instruction overrun fetch → Branch destination instruction fetch

   Instruction execution: Conditional branch instruction execution → Branch destination instruction execution

2. TRAPA instruction, branch taken: TRAPA

   Instruction fetch cycles: TRAPA instruction fetch → Next-instruction overrun fetch → Next-instruction overrun fetch → Branch destination instruction fetch

Instruction execution: TRAPA instruction execution → Branch destination instruction execution

3. Conditional delay branch instruction, branch taken: BT/S, BF/S

   Instruction fetch cycles: Conditional delay branch instruction fetch → Next-instruction fetch (delay slot) → Next-instruction overrun fetch → Branch destination instruction fetch

   Instruction execution: Conditional delay branch instruction execution → Delay slot instruction execution → Branch destination instruction execution

When a conditional branch instruction or TRAPA instruction causes a branch, the branch destination will be fetched after the next instruction or the one after that does an overrun fetch. However, because the instruction that is the object of the break first breaks after a definite instruction fetch and execution, the kind of overrun fetch instructions noted above do not become objects of a break. If data access breaks are also included with instruction fetch breaks as break conditions, a break occurs because the instruction overrun fetch is also regarded as becoming a data break.

7.5.3 Contention between User Break and Exception Handling

If a user break is set for the fetch of a particular instruction, and exception handling with higher priority than a user break is in contention and is accepted in the decode stage for that instruction (or the next instruction), user break exception handling may not be performed after completion of the higher-priority exception handling routine (on return by RTE).

Thus, if a user break condition has been set for the fetch of the branch destination instruction following a branch (BRA, BRAF, BT, BF, BT/S, BF/S, BSR, BSRF, JMP, JSR, RTS, RTE, exception handling), and exception handling for this branch destination instruction with a higher priority than a user break interrupt is accepted, user break exception handling will not be performed after completion of that exception handling routine.

Therefore, a user break condition must not be set for the fetch of the branch destination instruction following a branch.

7.5.4 Break at Non-Delay Branch Instruction Jump Destination

When a branch instruction with no delay slot (including exception handling) jumps to the jump destination instruction on execution of the branch, a user break will not be generated even if a user break condition has been set for the first jump destination instruction fetch.

Section 8 Data Transfer Controller (DTC)

8.1 Overview

The SH7040 Series has an on-chip data transfer controller (DTC), which is activated either by interrupts or software and can perform data transfers.

8.1.1 Features

- Arbitrary channel number transfer setting possible
  - Transfer information can be established for each interrupt source
  - Transfer information stored in memory
  - Multiple data transfers possible (chain transfers) for one activating source
- Address space: 32-bit addresses can be designated for both transfer source and destination
- Transfer devices
  - Memory: On-chip ROM, on-chip RAM, external ROM, external RAM
  - On-chip peripheral modules (excluding DMAC/DTC)
  - Memory-mapped external devices
- Abundant transfer modes
  - Can select between normal mode/repeat mode/block transfer mode
  - Can select between increment/decrement/fixed for source/destination address
- Transfer units can be set as byte/word/longword
- Interrupts activating the DTC can be requested of the CPU
  - Interrupt requests can be generated to the CPU after completion of a data transfer
  - Interrupt requests generated to the CPU after completion of all designated data transfers
- Transfers can be activated by software

8.1.2 Block Diagram

Figure 8.1 shows the DTC block diagram. DTC transfer information is located in memory.

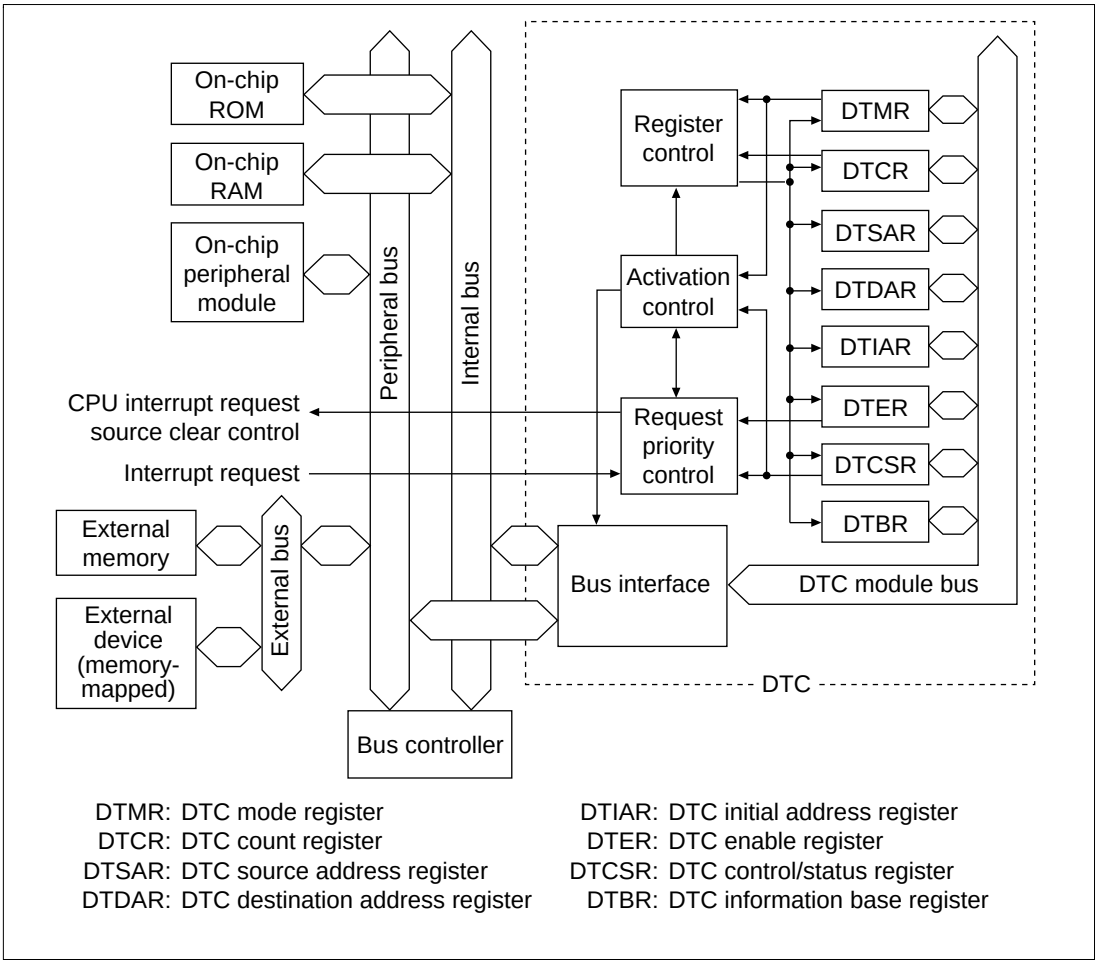


Figure 8.1 DTC Block Diagram

8.1.3 Register Configuration

The DTC has five registers in memory used for storing transfer data: DTMR, DTCR, DTSAR, DTDAR, and DTIAR. It is controlled by the three registers DTER (DTEA–DTEE), DTCSR, and DTBR. The register configurations are listed in table 8.1.

Table 8.1 Register Configuration*1

Name	Abbr.	R/W	Initial Value	Address	Access Size
DTC mode register	DTMR	—*2	Undefined	—*2	—*2
DTC source address register	DTSAR	—*2	Undefined	—*2	—*2
DTC destination address register	DTDAR	—*2	Undefined	—*2	—*2
DTC initial address register	DTIAR	—*2	Undefined	—*2	—*2
DTC transfer count register A	DTCRA	—*2	Undefined	—*2	—*2
DTC transfer count register B	DTCRB	—*2	Undefined	—*2	—*2
DTC enable register A	DTEA	R/W	H'00	H'FFFF8700	8, 16, 32
DTC enable register B	DTEB	R/W	H'00	H'FFFF8701	8, 16, 32
DTC enable register C	DTEC	R/W	H'00	H'FFFF8702	8, 16, 32
DTC enable register D	DTED	R/W	H'00	H'FFFF8703	8, 16, 32
DTC enable register E	DTEE	R/W	H'00	H'FFFF8704	8, 16, 32
DTC control/status register	DTCSR	R/(W)*3	H'0000	H'FFFF8706	8, 16, 32
DTC information base register	DTBR	R/W	Undefined	H'FFFF8708	16, 32

Notes: *1 DTC registers cannot be accessed by DMAC/DTC.
*2 DTC internal registers cannot be directly accessed.
*3 Only a 0 write after a 1 read is possible for the NMIF, AE bits of the DTCSR.

8.2 Register Description

8.2.1 DTC Mode Register (DTMR)

The DTC mode register (DTMR) is a 16-bit register that controls the DTC operation mode. The contents of this register is located in memory.

Bit:	15	14	13	12	11	10	9	8
	SM1	SM0	DM1	DM0	MD1	MD0	SZ1	SZ0
Initial value:	*	*	*	*	*	*	*	*
R/W:	—	—	—	—	—	—	—	—

Bit:	7	6	5	4	3	2	1	0
Bit name:	DTS	CHNE	DISEL	NMIM	—	—	—	—
Initial value:	*	*	*	*	*	*	*	*
R/W:	—	—	—	—	—	—	—	—

Note: * Initial value undefined.

- Bits 15–14—Source Address Mode 1, 0 (SM1, SM0): These bits designate whether to hold, increment, or decrement the DTSAR after a data transfer.

Bit 15 (SM1)	Bit 14 (SM0)	Description
0	—	DTSAR remains fixed
1	0	DTSAR is incremented after transfer (+1 for byte unit transfer, +2 for word, +4 for longword)
1	1	DTSAR is decremented after transfer (–1 for byte unit transfer, –2 for word, –4 for longword)

- Bits 13–12—Destination Address Mode 1, 0 (DM1, DM0): These bits designate whether to hold, increment or decrement the DTDAR after a data transfer.

Bit 13 (DM1)	Bit 12 (DM0)	Description
0	—	DTDAR remains fixed
1	0	DTDAR is incremented after transfer (+1 for byte unit transfer, +2 for word, +4 for longword)
1	1	DTDAR is decremented after transfer (–1 for byte unit transfer, –2 for word, –4 for longword)

- Bits 11–10—DTC Mode 1, 0 (MD1, MD0): These bits designate the DTC transfer mode.

Bit 11 (MD1)	Bit 10 (MD0)	Description
0	0	Normal mode
0	1	Repeat mode
1	0	Block transfer mode
1	1	Reserved (setting prohibited)

- Bits 9–8—DTC Data Transfer Size 1, 0 (SZ1, SZ0): These bits designate the data size for data transfers.

Bit 9 (SZ1)	Bit 8 (SZ0)	Description
0	0	Byte (8 bits)
0	1	Word (16 bits)
1	0	Longword (32 bits)
1	1	Reserved (setting prohibited)

- Bit 7—DTC Transfer Mode Select (DTS): When in repeat mode or block transfer mode, this bit designates whether the source side or destination side will be the repeat area or block area.

Bit 7 (DTS)	Description
0	Destination side is the repeat area or block area
1	Source side is the repeat area or block area

- Bit 6—DTC Chain Enable (CHNE): This bit designates whether to perform continuous DTC data transfers with the same activating source. Continued transfer information is read after the 16th byte from the start address of the previous transfer information.

Bit 6 (CHNE)	Description
0	DTC data transfer end (activation wait state ensues)
1	DTC data transfer continue (read continue register information, execute transfer)

- Bit 5—DTC Interrupt Select (DISEL): This bit designates whether to prohibit or allow interrupt requests to the CPU after one-time DTC transfers.

Bit 5 (DISEL)	Description
0	Prohibit interrupts to the CPU after DTC data transfer completion if the transfer counter is not 0 (DTC clears the interrupt source flag of the activating source to 0)
1	Allow interrupts to the CPU after DTC data transfer completion (DTC clears the DTER bit for the interrupt of the activating source to 0)

- Bit 4—DTC NMI Mode (NMIM): This bit designates whether to terminate transfers when an NMI is input during DTC transfers.

Bit 4 (NMIM)	Description
0	Terminate DTC transfer upon an NMI
1	Continue DTC transfer until end of transfer being executed

- Bits 3–0—Reserved: They have no effect on DTC operation.

8.2.2 DTC Source Address Register (DTSAR)

The DTC source address register (DTSAR) is a 32-bit register that specifies the DTC transfer source address. An even address indicates that the transfer size is word; a multiple-of-four address means it is longword. The contents of this register is located in memory.

Bit:	31	30	29	28	27	…	4	3	2	1	0
						…					
Initial value:	*	*	*	*	*	…	*	*	*	*	*
R/W:	—	—	—	—	—	…	—	—	—	—	—

Note: * Initial value is undefined.

8.2.3 DTC Destination Address Register (DTDAR)

The DTC destination address register (DTDAR) is a 32-bit register that specifies the DTC transfer destination address. An even address indicates that the transfer size is word; a multiple-of-four address means it is longword. The contents of this register are located in memory.

Bit:	31	30	29	28	27	…	4	3	2	1	0
						…					
Initial value:	*	*	*	*	*	…	*	*	*	*	*
R/W:	—	—	—	—	—	…	—	—	—	—	—

Note: * Initial value is undefined.

8.2.4 DTC Initial Address Register (DTIAR)

The DTC initial address register (DTIAR) specifies the initial transfer source/transfer destination address in repeat mode. In repeat mode, when the DTS bit is set to 1, specify the initial transfer source address in the repeat area, and when the DTS bit is cleared to 0, specify the initial transfer destination address in the repeat area.

The contents of this register are located in memory.

Bit:	31	30	29	28	27	…	4	3	2	1	0
						…					
Initial value:	*	*	*	*	*	…	*	*	*	*	*
R/W:	—	—	—	—	—	…	—	—	—	—	—

Note: * Initial value is undefined.

8.2.5 DTC Transfer Count Register A (DTCRA)

DTCRA is a 16-bit register that specifies the number of DTC transfers. The contents of this register are located in memory.

In normal mode it functions as a 16-bit transfer counter. The number of transfers is 1 when the set value is H'0001, 65535 when it is H'FFFF, and 65536 when it is H'0000.

In repeat mode, DTCRAH maintains the transfer count and DTCRAL functions as an 8-bit transfer counter. The number of transfers is 1 when the set value is DTCRAH = DTCRAL = H'01, 255 when they are H'FF, and 256 when it is H'00.

In block transfer mode it functions as a 16-bit transfer counter. The number of transfers is 1 when the set value is H'0001, 65535 when it is H'FFFF, and 65536 when it is H'0000.

Bit:	15	14						8
			DTCRAH					
Initial value:	*	*	*	*	*	*	*	*
R/W:	—	—	—	—	—	—	—	—

Bit:	7	6						0
			DTCRAL					
Initial value:	*	*	*	*	*	*	*	*
R/W:	—	—	—	—	—	—	—	—

Note: * Initial value is undefined.

8.2.6 DTC Transfer Count Register B (DTCRB)

The DTCRB is a 16-bit register that designates the block length in block transfer mode. The contents of this register is located in memory. The block length is 1 when the set value is H'0001, 65535 when it is H'FFFF, and 65536 when it is H'0000.

Bit:	15	14	13	12	11	10	9	8
Initial value:	*	*	*	*	*	*	*	*
R/W:	—	—	—	—	—	—	—	—

Bit:	7	6	5	4	3	2	1	0
Initial value:	*	*	*	*	*	*	*	*
R/W:	—	—	—	—	—	—	—	—

Note: * Initial value is undefined.

8.2.7 DTC Enable Registers (DTER)

The DTER (DTEA–DTEE) are five 8-bit readable/writable registers with bits allocated to each interrupt source that activates the DTC. They set disable/enable for DTC activation for each interrupt source. When a bit is 1, DTC activation by the corresponding interrupt source is enabled. Interrupt sources for each of the DTEA–DTEE registers are indicated in table 8.2.

The DTER are initialized to H'00 by a power-on reset or in standby mode. Manual reset does not initialize DTER.

For the A mask, overwrite this register as follows:

When clearing bit to 0: read the 1 bit to clear and write 0.
When setting bit to 1: read the 0 bit to set and write 1.

Bit:	7	6	5	4	3	2	1	0
	DTE7	DTE6	DTE5	DTE4	DTE3	DTE2	DTE1	DTE0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*

Note: * DTER bits can only be modified by writing 1 after reading 0, or writing 0 after reading 1.

8.2.8 DTC Control/Status Register (DTCSR)

The DTCSR is a 16-bit readable/writable register that sets disable/enable for DTC activation by software, as well as the DTC vector addresses for software activation. It also indicates the DTC transfer status.

The DTCSR is initialized to H'0000 by power-on resets and in standby mode. Manual reset does not initialize DTCSR.

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	NMIF	AE	SWDTE
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W*1	R/W*1	R/W*2

Bit:	7	6	5	4	3	2	1	0
Bit name:	DTVEC7	DTVEC6	DTVEC5	DTVEC4	DTVEC3	DTVEC2	DTVEC1	DTVEC0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W*3	R/W*3	R/W*3	R/W*3	R/W*3	R/W*3	R/W*3	R/W*3 *4

Notes: *1 For the NMIF and AE bits, only a 0 write after a 1 read is possible.
*2 For the SWDTE bit, a 1 write is always possible, but a 0 write is possible only after a 1 is read.
*3 For the DTVEC7–DTVEC0 bits, writes are possible only when SWDTE = 0.
*4 Be sure to write 0 to the DTVEC0 bit.

Bits 15–11—Reserved: These bits always read as 0. The write value should always be 0.

- Bit 10—NMI Flag Bit (NMIF): Indicates that an NMI interrupt has occurred. When the NMIF bit is set, DTC transfers are not allowed even if the DTER bit is set to 1. If, however, a transfer has already started with the NMIM bit of the DTMR set to 1, execution will continue until that transfer ends. To clear the NMIF bit, read the 1 from it, then write a 0.

  The NMIF bit is initialized to 0 by power-on resets and in standby mode.

Bit 10 (NMIF)	Description
0	No NMI interrupts (initial value) (Clear condition) Write a 0 after reading the NMIF bit
1	NMI interrupt has been generated

- Bit 9—Address Error Flag (AE): Indicates that an address error by the DTC has occurred. When the AE bit is set, DTC transfers are not allowed even if the DTER bit is set to 1. To clear the AE bit, read the 1 from it, then write a 0.

  The AE bit is initialized to 0 by power-on resets and in standby mode.

Bit 9 (AE)	Description
0	No address error by the DTC (initial value) (Clear condition) Write a 0 after reading the AE bit
1	An address error by the DTC occurred

- Bit 8—DTC Software Activation Enable Bit (SWDTE): This bit enables/disables DTC activation by software.

  The AE bit is initialized to 0 by resets and standby mode. For details, see section 8.3.2, Activating Sources.

Bit 8 (SWDTE)	Description
0	DTC activation by software disabled (initial value)
1	DTC activation by software enabled

- Bits 7–0—Software Activation Vectors 7–0 (DTVEC7–DTVEC0): These bits set the DTC vector addresses for DTC activation by software. A vector address is calculated as H'0400 + DTVEC[7:0]. Always specify 0 for DTVEC0. 8 bits are available, so you can specify values H'00 (0)–H'FE (254).

8.2.9 DTC Information Base Register (DTBR)

The DTBR is a 16-bit readable/writable register that specifies the upper 16 bits of the memory address containing DTC transfer information. Always access the DTBR in word or longword units. If it is accessed in byte units the register contents will become undefined at the time of a write, and undefined values will be read out upon reads.

The DTBR is not initialized either by resets or in standby mode.

Bit:	15	14	13	12	11	10	9	8
Initial value:	*	*	*	*	*	*	*	*
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
Initial value:	*	*	*	*	*	*	*	*
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: * Initial value is undefined.

8.3 Operation

The DTC stores transfer information in memory. When there are DTC transfer requests, it reads that transfer information and performs data transfers based on it. It rewrites the transfer information to memory after data transfers. Storing transfer information in memory makes it possible to perform data transfers for an arbitrary number of channels. Further, setting the CHNE bit to 1 makes it possible to perform multiple transfers continuously through one DTC transfer request.

There are three DTC transfer modes: normal mode, repeat mode, and block transfer mode. After a DTC transfer, the transfer source address and transfer destination address are incremented, decremented, or kept the same, according to the respective setting.

8.3.1 Overview of Operation

Figure 8.2 shows a flowchart of DTC operation.

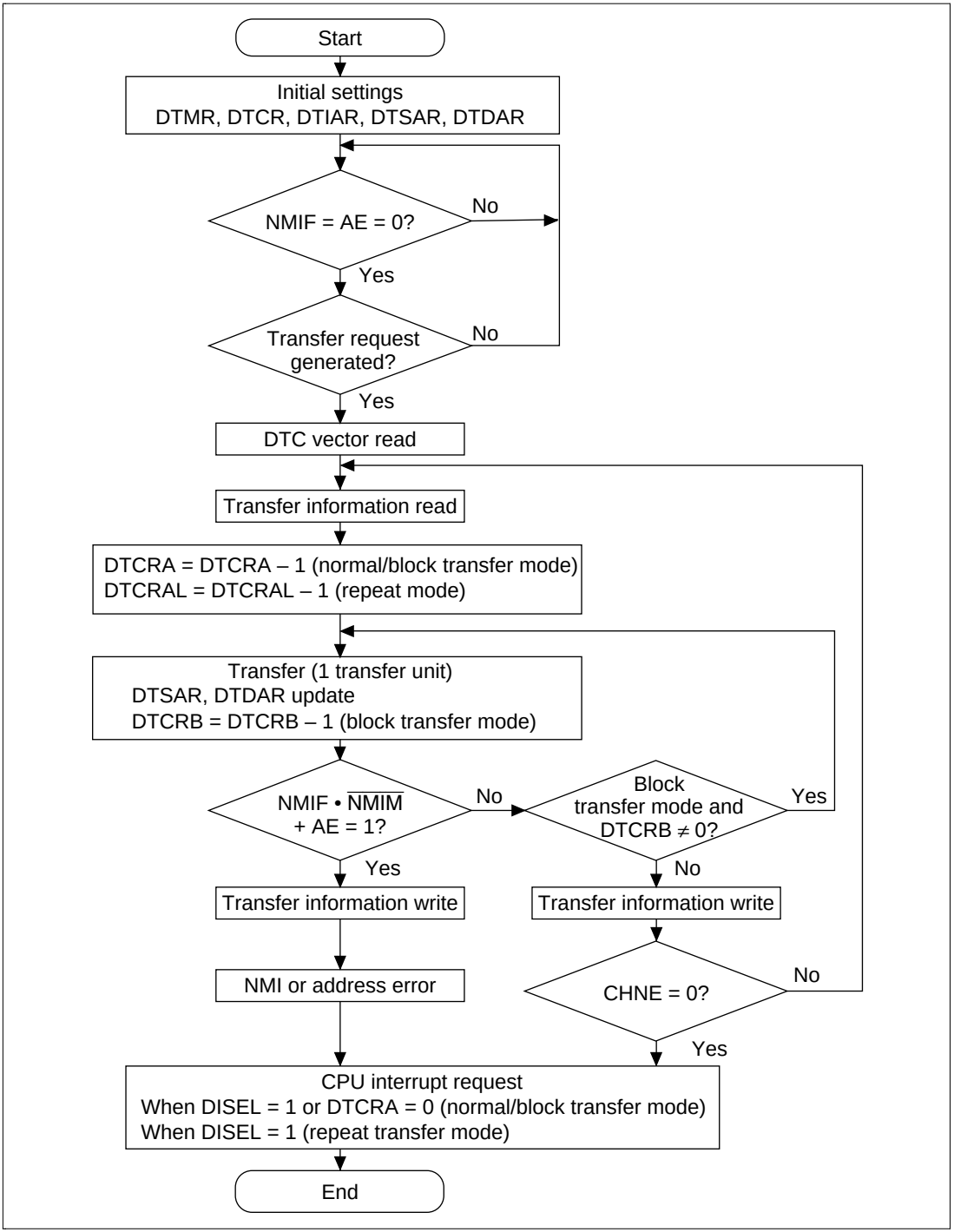


Figure 8.2 DTC Operation Flowchart

8.3.2 Activating Sources

The DTC performs write operations to the DTCSR with either interrupt sources or software as its activating sources. Each interrupt source is designated by specific DTER bits to determine whether it becomes an interrupt request to the CPU or a DTC activating source.

When the DISEL bit is 1, an interrupt, established as the DTC activating source, is requested of the CPU after each data transfer in DRC. When the DISEL bit is a 0, a request is made only after the completion of a designated number of data transfers. When the activating source interrupt is requested of the CPU, the corresponding DTER bit is automatically cleared.

In the case of software activation also, when the DISEL bit is a 1, a software DTC activation interrupt (SWDTCE) is requested of the CPU after each data transfer. When the DISEL bit is a 0, a request is made only after the completion of a designated number of data transfers. When no SWDTCE interrupt is requested of the CPU, the SWDTE bit of the DTCSR is automatically cleared. When a request is made of the CPU, the SWDTE bit is maintained as a 1.

When multiple DTC activating sources occur simultaneously, they are accepted and the DTC is activated in accordance with the default priority rankings shown in table 8.2.

Figure 8.3 shows a block diagram of activating source control.

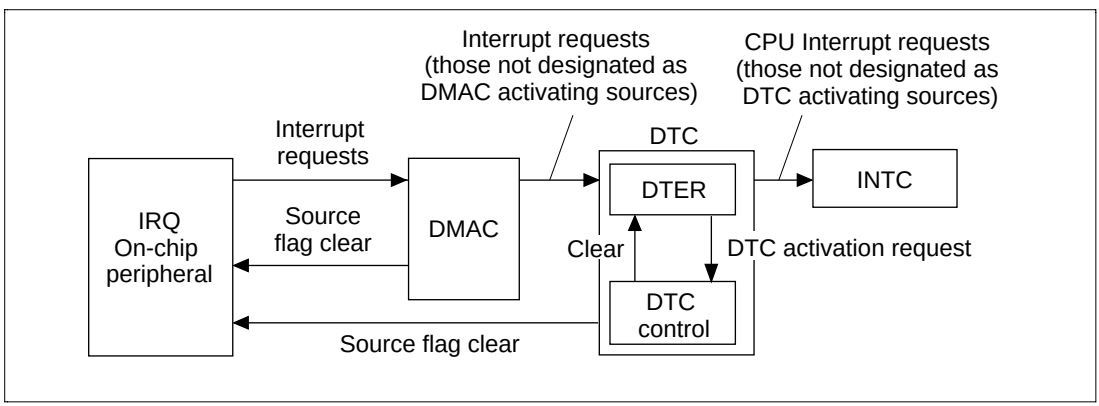


Figure 8.3 Activating Source Control Block Diagram

8.3.3 DTC Vector Table

Figure 8.4 shows the correspondence between DTC vector addresses and register information placement. For each DTC activating source there are 2 bytes in the DTC vector table, which contain the register information start address.

Table 8.2 shows the correspondence between activating sources and vector addresses. When activating with software, the vector address is calculated as H'0400 + DTVEC[7:0].

Through DTC activation, a register information start address is read from the vector table, then register information placed in memory space is read from that register information start address. Always designate register information start addresses in multiples of four.

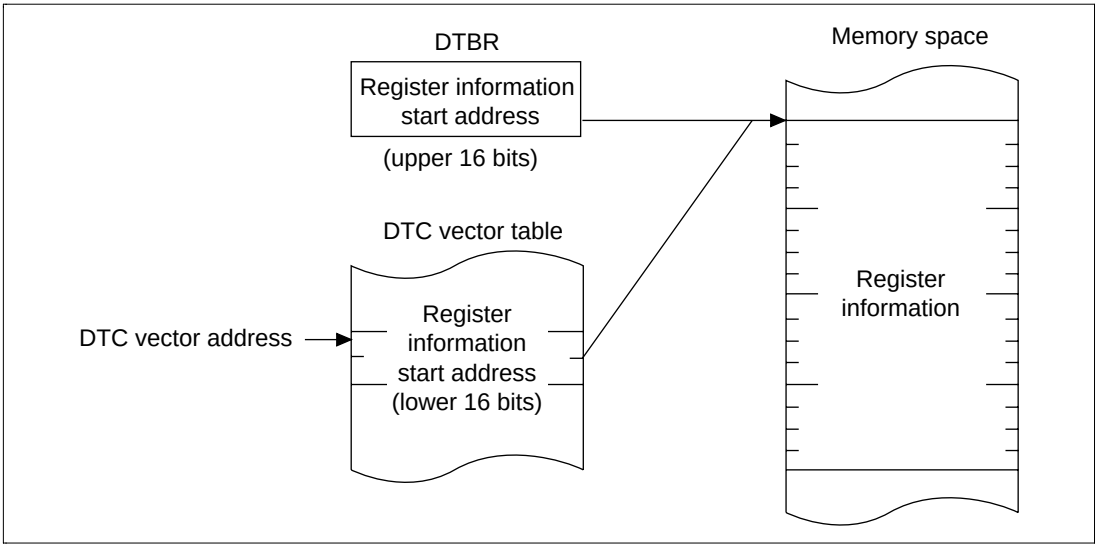


Figure 8.4 Correspondence between DTC Vector Address and Register Information

Table 8.2 Interrupt Sources and DTC Vector Addresses

Source Generator	Activating Source	DTC Vector Address	DTE Bit	Transfer Source	Transfer Destination	Priority
MTU (CH4)	TGI4A	H'00000400–H'00000401	DTEA7	Arbitrary*1	Arbitrary*1	High
	TGI4B	H'00000402–H'00000403	DTEA6	Arbitrary*1	Arbitrary*1	
	TGI4C	H'00000404–H'00000405	DTEA5	Arbitrary*1	Arbitrary*1	
	TGI4D	H'00000406–H'00000407	DTEA4	Arbitrary*1	Arbitrary*1	
	TCI4V	H'00000408–H'00000409	DTEA3	Arbitrary*1	Arbitrary*1	
MTU (CH3)	TGI3A	H'0000040A–H'0000040B	DTEA2	Arbitrary*1	Arbitrary*1	
	TGI3B	H'0000040C–H'0000040D	DTEA1	Arbitrary*1	Arbitrary*1	
	TGI3C	H'0000040E–H'0000040F	DTEA0	Arbitrary*1	Arbitrary*1	
	TGI3D	H'00000410–H'00000411	DTEB7	Arbitrary*1	Arbitrary*1	
MTU (CH2)	TGI2A	H'00000412–H'00000413	DTEB6	Arbitrary*1	Arbitrary*1	
	TGI2B	H'00000414–H'00000415	DTEB5	Arbitrary*1	Arbitrary*1	
MTU (CH1)	TGI1A	H'00000416–H'00000417	DTEB4	Arbitrary*1	Arbitrary*1	
	TGI1B	H'00000418–H'00000419	DTEB3	Arbitrary*1	Arbitrary*1	
MTU (CH0)	TGI0A	H'0000041A–H'0000041B	DTEB2	Arbitrary*1	Arbitrary*1	
	TGI0B	H'0000041C–H'0000041D	DTEB1	Arbitrary*1	Arbitrary*1	
	TGI0C	H'0000041E–H'0000041F	DTEB0	Arbitrary*1	Arbitrary*1	
	TGI0D	H'00000420–H'00000421	DTEC7	Arbitrary*1	Arbitrary*1	
A/D	ADI(ADI0)*2	H'00000422–H'00000423	DTEC6	ADDR	Arbitrary*1	
IRQ0 pin	IRQ0	H'00000424–H'00000425	DTEC5	Arbitrary*1	Arbitrary*1	
IRQ1 pin	IRQ1	H'00000426–H'00000427	DTEC4	Arbitrary*1	Arbitrary*1	
IRQ2 pin	IRQ2	H'00000428–H'00000429	DTEC3	Arbitrary*1	Arbitrary*1	
IRQ3 pin	IRQ3	H'0000042A–H'0000042B	DTEC2	Arbitrary*1	Arbitrary*1	
IRQ4 pin	IRQ4	H'0000042C–H'0000042D	DTEC1	Arbitrary*1	Arbitrary*1	
IRQ5 pin	IRQ5	H'0000042E–H'0000042F	DTEC0	Arbitrary*1	Arbitrary*1	
IRQ6 pin	IRQ6	H'00000430–H'00000431	DTED7	Arbitrary*1	Arbitrary*1	
IRQ7 pin	IRQ7	H'00000432–H'00000433	DTED6	Arbitrary*1	Arbitrary*1	
CMT (CH0)	CMI0	H'00000434–H'00000435	DTED5	Arbitrary*1	Arbitrary*1	
CMT (CH1)	CMI1	H'00000436–H'00000437	DTED4	Arbitrary*1	Arbitrary*1	Low

Table 8.2 Interrupt Sources and DTC Vector Addresses (cont)

Source Generator	Activating Source	DTC Vector Address	DTE Bit	Transfer Source	Transfer Destination	Priority
SCI0	RXI0	H'00000438–H'00000439	DTED3	RDR0	Arbitrary*1	High
	TXI0	H'0000043A–H'0000043B	DTED2	Arbitrary*1	TDR0	
SCI1	RXI1	H'0000043C–H'0000043D	DTED1	RDR1	Arbitrary*1	
	TXI1	H'0000043E–H'0000043F	DTED0	Arbitrary*1	TDR1	
BSC	CMI	H'00000440–H'00000441	DTEE7	Arbitrary*1	Arbitrary*1	
Software	Write to DTCSR	H'00000400 + DTVEC[7:0] to H'00000401 + DTVEC[7:0]	—	Arbitrary*1	Arbitrary*1	Low

Notes: *1 External memory, memory-mapped external devices, on-chip memory, on-chip peripheral modules (excluding DMAC and DTC)

*2 Excluding A mask products are ADI, A mask products are ADI0.

8.3.4 Register Information Placement

Figure 8.5 shows the placement of register information in memory space. The register information start addresses are designated by DTBR for the upper 16 bits, and the DTC vector table for the lower 16 bits.

The placement in order from the register information start address in normal mode is DTMR, DTCRA, 4 bytes empty (no effect on DTC operation), DTSAR, then DTDAR. In repeat mode it is DTMR, DTCRA, DTIAR, DTSAR, and DTDAR. In block transfer mode, it is DTMR, DTCRA, 2 bytes empty (no effect on DTC operation), DTCRB, DTSAR, then DTDAR.

Fundamentally, certain RAM areas are designated for addresses storing register information.

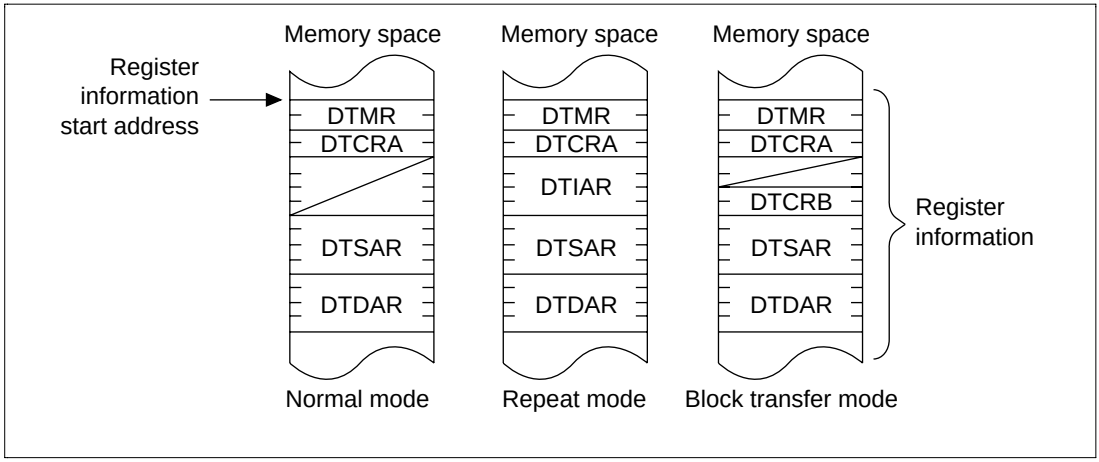


Figure 8.5 DTC Register Information Placement in Memory Space

8.3.5 Normal Mode

Performs the transfer of one byte, one word, or one longword for each activation. The total transfer count is 1 to 65536. An interrupt request is generated to the CPU when the transfer with DTCRA = 1 ends. Transfers of a number of bytes specified by the SCI are possible.

Table 8.3 shows the register functions for normal mode.

Table 8.3 Normal Mode Register Functions

		Values Written Back upon a Transfer Information Write	
Register	**Function**	**When DTCRA is other than 1**	**When DTCRA is 1**
DTMR	Operation mode control	DTMR	DTMR
DTCRA	Transfer count	DTCRA – 1	DTCRA – 1 (= H'0000)
DTSAR	Transfer source address	Increment/decrement/ fixed	Increment/decrement/ fixed
DTDAR	Transfer destination address	Increment/decrement/ fixed	Increment/decrement/ fixed

8.3.6 Repeat Mode

Performs the transfer of one byte, one word, or one longword for each activation. Either the transfer source or transfer destination is designated as the repeat area.

The total transfer count is specified between 1 and 256. When the specified number of transfers ends, the address register of the designated repeat area is returned to its initial state and the transfer is repeated. Other address registers are consecutively incremented, decremented, or remain fixed. While DISEL = 0, no interrupt request is made to the CPU, even if the transfer with DTCRAL = 1 ends.

Pulses for driving the stepping motor can be output. Table 8.4 shows the register functions for repeat mode.

Table 8.4 Repeat Mode Register Functions

		Values Written Back upon a Transfer Information Write	
Register	**Function**	**When DTCRA is other than 1**	**When DTCRA is 1**
DTMR	Operation mode control	DTMR	DTMR
DTCRAH	Transfer count maintenance	DTCRAH	DTCRAH
DTCRAL	Transfer count	DTCRAL – 1	DTCRAH
DTIAR	Initial address	(Not written back)	(Not written back)
DTSAR	Transfer source address	Increment/decrement/ fixed	(DTS = 0) Increment/ decrement/ fixed (DTS = 1) DTIAR
DTDAR	Transfer destination address	Increment/decrement/ fixed	(DTS = 0) DTIAR (DTS = 1) Increment/ decrement/ fixed

8.3.7 Block Transfer Mode

Performs the transfer of one block for each one activation. Either the transfer source or transfer destination is designated as the block area.

The block length is specified between 1 and 65536. When a 1-block transfers ends, the address register of the designated block area is returned to its initial state. Other address registers are consecutively incremented, decremented, or remain fixed. The block transfer count is 1 to 65536. An interrupt request is generated to the CPU when the transfer with DTCRA = 1 ends.

A/D converter group mode transfers and phase compensation PWM data transfers are possible.

Table 8.5 shows the register functions for block transfer mode.

Table 8.5 Block Transfer Mode Register Functions

Register	Function	Values Written Back upon a Transfer Information Write
DTMR	Operation mode control	DTMR
DTCRA	Transfer count	DTCRA – 1
DTCRB	Block length	(Not written back)
DTSAR	Transfer source address	(DTS = 0) Increment/ decrement/ fixed (DTS = 1) DTSAR initial value
DTDAR	Transfer destination address	(DTS = 0) DTDAR initial value (DTS = 1) Increment/ decrement/ fixed

8.3.8 Operation Timing

Figure 8.6 shows a DTC operation timing example.

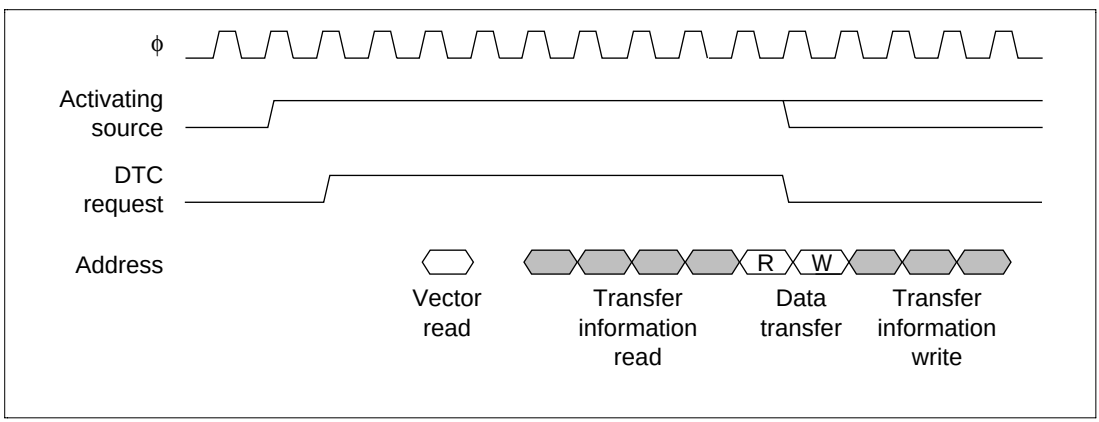


Figure 8.6 DTC Operation Timing Example (Normal Mode)

When register information is located in on-chip RAM, each mode requires 4 cycles for transfer information reads, and 3 cycles for writes.

8.3.9 DTC Execution State Counts

Table 8.6 shows the execution state for one DTC data transfer. Furthermore, table 8.7 shows the state counts needed for execution state.

Table 8.6 Execution State of DTC

Mode	Vector Read I	Register Information Read/Write J	Data Read K	Data Write L	Internal Operation M
Normal	1	7	1	1	1
Repeat	1	7	1	1	1
Block transfer	1	7	N	N	1

Note: N: block size (default set values of DTCRB)

Table 8.7 State Counts Needed for Execution State

Access Objective			On-chip RAM	On-chip ROM	Internal I/O Register		External Device		
Bus width			32	32	32		8	16	32
Access state			1	1	2*1	3*2	2	2	2
Execution state	Vector read	S_I	—	1	—		4	2	2
	Register information read/write	S_J	1	1	—		8	4	2
	Byte data read	S_K	1	1	2	3	2	2	2
	Word data read	S_K	1	1	2	3	4	2	2
	Long word data read	S_K	1	1	4	6	8	4	2
	Byte data write	S_L	1	1	2	3	2	2	2
	Word data write	S_L	1	1	2	3	4	2	2
	Long word write	S_L	1	1	4	6	8	4	2
	Internal operation	S_M	1						

Notes: *1 Two state access module : port, INT, CMT, SCI, etc.
*2 Three state access module : WDT, CACHE, UBC, etc.

The execution state count is calculated using the following formula. $\sum$ indicates the number of transfers by one activating source (count + 1 when CHNE bit is 1).

$$\text{Execution state count} = I \cdot S_I + \sum (J \cdot S_J + K \cdot S_K + L \cdot S_L) + M \cdot S_M$$

8.3.10 DTC Usage Procedure

The procedure for DTC interrupt activation is as follows:

1. Transfer data (DTMR, DTCRA, DTSAR, DTDAR, DTCRB, and DTIAR) is located in memory space.
2. Establish the register information start address with DTBR and the DTC vector table.
3. Set the corresponding DTER bit to 1.
4. The DTC is activated when an interrupt source occurs.
5. When interrupt requests are not made to the CPU, the interrupt source is cleared, but the DTER is not. When interrupts are requested, the interrupt source is not cleared, but the DTER is.
6. Interrupt sources are cleared within the CPU interrupt routine. When doing continuous DTC data transfers, set the DTER to 1.

The procedure for DTC software activation is as follows:

1. Transfer data (DTMR, DTCRA, DTSAR, DTDAR, DTCRB, and DTIAR) is located in memory space.
2. Establish the register information start address with DTBR and the DTC vector table.
3. Confirm that the SWDTE bit of the DTCSR is 0. When the SWDTE bit is 1, the DTC is already being driven by software.
4. Write a 1 to the SWDTE bit and a vector number to the DTVEC (byte data).
5. When SWDTCE interrupt requests are not made to the CPU, the SWDTE bit is cleared. When interrupts are requested, the SWDTE bit is maintained as a 1.
6. The SWDTE bit is cleared to 0 within the CPU interrupt routine. For continuous DTC data transfers, set the SWDTE to 1.

8.3.11 DTC Use Example

The following is a DTC use example of a 128-byte data reception by the SCI:

1. The settings are: DTMR source address fixed (SM1 = 0), destination address incremented (DM1 = 1, DM0 = 0), normal mode (MD1 = MD0 = 0), byte size (SZ1 = SZ0 = 0), one transfer per activating source (CHNE = 0), and a CPU interrupt request after the designated number of data transfers (DISEL = 0). 128 (H'0080) is set in DTCRA, the RDR address of the SCI is set in DTSAR, and the start address of the RAM storing the receive data is set in DTDAR.
2. Establish the register information start address with DTBR and the DTC vector table.
3. Set the corresponding DTER bit to 1.
4. Set the SCI to a specific receive mode and enable RxI interrupts.

5. The RDRF flag of the SSR is set to 1 by each completion of a 1-byte data reception by the SCI, an RxI interrupt is generated, and the DTC is activated. The received data is transferred from RDR to RAM by the DTC, and the RDRF flag is 0 cleared.
6. After completion of 128 data transfers (DTCRA = 0), the DTER is cleared while the RDRF is maintained as 1, and an RxI interrupt request is made to the CPU. The interrupt processing routine clears the RDRF, and performs the other completion processing.

8.4 Cautions on Use

- DMAC and DTC register access by the DTC is prohibited.
- DTC register access by the DMAC is prohibited.
- When setting a bit in DTER, first ensure that all transfers on the DTC channel corresponding to that DTER have ended, or disable the transfer source for each channel so that DTC transfer corresponding to that DTER will not occur. The above restrictions do not apply for A mask due to change in the access method of DTER. However, take caution when changing LSI to A mask, since modification of the program is required.

Section 9 Cache Memory (CAC)

9.1 Overview

The LSI has an on-chip cache memory (CAC) with 1 kbyte of cache data and a 256-entry cache tag. The cache data and cache tag space can be used as on-chip RAM space when the cache is not being used.

9.1.1 Features

The CAC has the following features. The cache tag and cache data configuration is shown in figure 9.1.

- 1-kbyte capacity
- External memory (CS space and DRAM space) instruction code and PC relative data caching
- 256 entry cache tag (tag address 15 bits)
- 4-byte line length
- Direct map replacement algorithm
- Valid flag (1 bit) included for purges

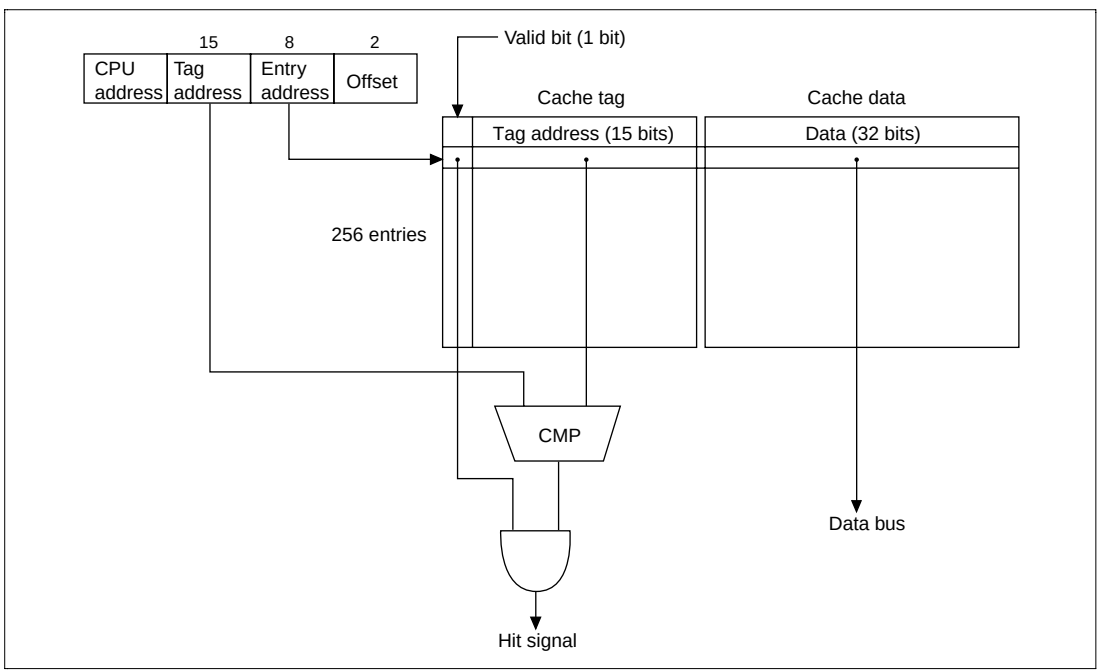


Figure 9.1 Cache Tag and Cache Data Configuration

9.1.2 Block Diagram

Figure 9.2 shows a block diagram of the cache.

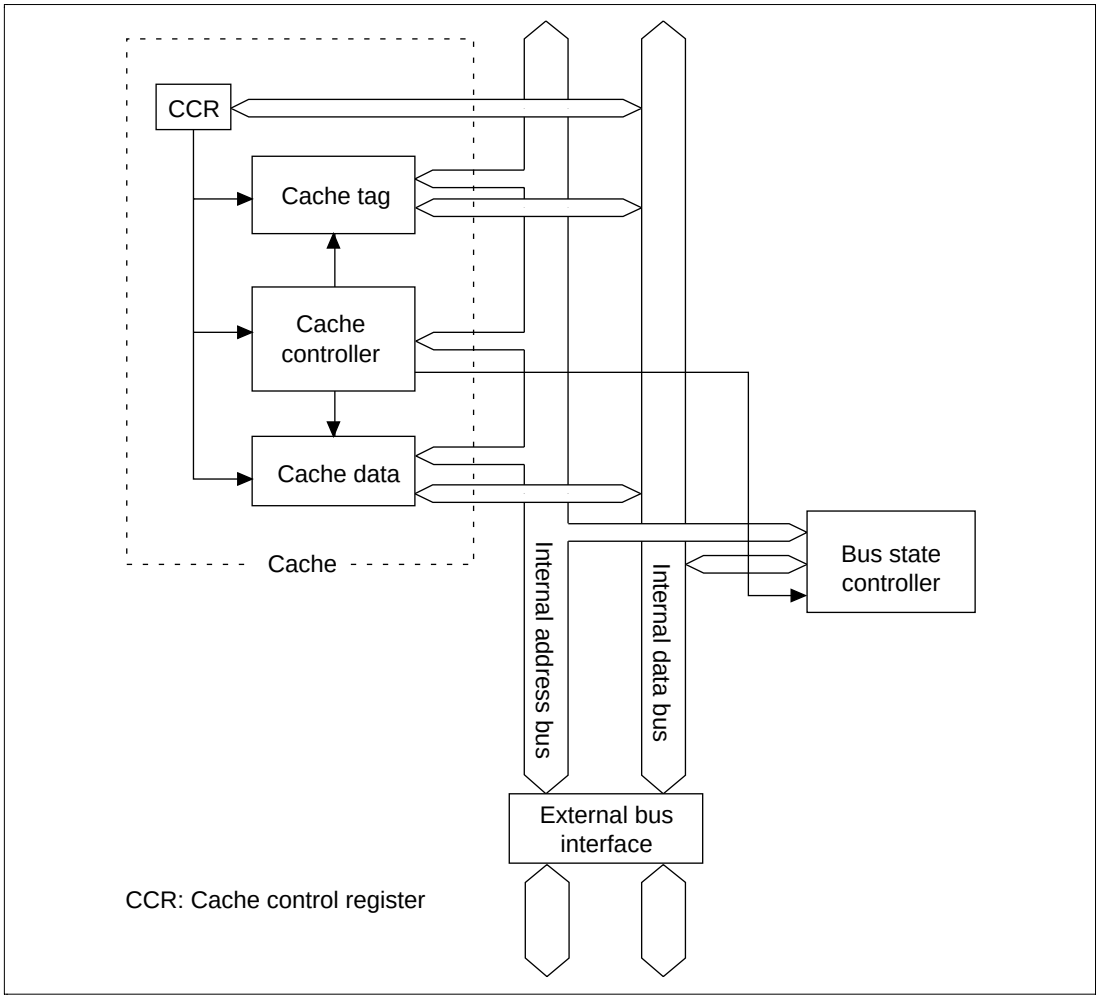


Figure 9.2 Cache Block Diagram

9.1.3 Register Configuration

The cache has one register, which can be used to control the enabling or disabling of each cache space. The register configuration is shown in table 9.1.

Table 9.1 Register Configuration

Name	Abbreviation	R/W	Initial Value	Address	Access Size (Bits)
Cache control register	CCR	R/W	H'0000*	H'FFFF8740	8, 16, 32

Note: * Bits 15–5 are undefined.

9.2 Register Explanation

9.2.1 Cache Control Register (CCR)

The cache control register (CCR) selects the cache enable/disable of each space.

The CCR is a 16-bit readable/writable register. It is initialized to H'0000 by power on resets, but is not initialized by manual resets or standby mode.

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	*	*	*	*	*	*	*	*
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	—	—	—	CE DRAM	CE CS3	CE CS2	CE CS1	CE CS0
Initial value:	*	*	*	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W

Note: * Bits 15–5 are undefined.

- Bits 15–5—Reserved: Reading these bits gives undefined values. The write value should always be 0.
- Bit 4—DRAM Space Cache Enable (CEDRAM): Selects whether to use DRAM space as a cache object (enable) or to exclude it (disable). A 0 disables, and a 1 enables such use.

Bit 4 (CEDRAM)	Description
0	DRAM space cache disabled (initial value)
1	DRAM space cache enabled

- Bit 3—CS3 Space Cache Enable (CECS3): Selects whether to use CS3 space as a cache object (enable) or to exclude it (disable). A 0 disables, and a 1 enables such use.

Bit 3 (CECS3)	Description
0	CS3 space cache disabled (initial value)
1	CS3 space cache enabled

- Bit 2—CS2 Space Cache Enable (CECS2): Selects whether to use CS2 space as a cache object (enable) or to exclude it (disable). A 0 disables, and a 1 enables such use.

Bit 2 (CECS2)	Description
0	CS2 space cache disabled (initial value)
1	CS2 space cache enabled

- Bit 1—CS1 Space Cache Enable (CECS1): Selects whether to use CS1 space as a cache object (enable) or to exclude it (disable). A 0 disables, and a 1 enables such use.

Bit 1 (CECS1)	Description
0	CS1 space cache disabled (initial value)
1	CS1 space cache enabled

- Bit 0—CS0 Space Cache Enable (CECS0): Selects whether to use CS0 space as a cache object (enable) or to exclude it (disable). A 0 disables, and a 1 enables such use.

Bit 0 (CECS0)	Description
0	CS0 space cache disabled (initial value)
1	CS0 space cache enabled

9.3 Address Array and Data Array

There is a special cache space for controlling the cache. This space is divided into an address array and a data array, where addresses (tag address, including valid bit) and data (4-byte line length) for cache control are recorded. The special cache space is shown in table 9.2. It can be used as on-chip RAM space when the cache is not being used.

Table 9.2 Special Cache Space

Space Classification	Address	Size	Bus Width
Address array	H'FFFFF000–H'FFFFF3FF	1 kbyte	32 bit
Data array	H'FFFFF400–H'FFFFF7FF	1 kbyte	32 bit

9.3.1 Cache Address Array Read/Write Space

The cache address array has a compulsory read/write (figure 9.3).

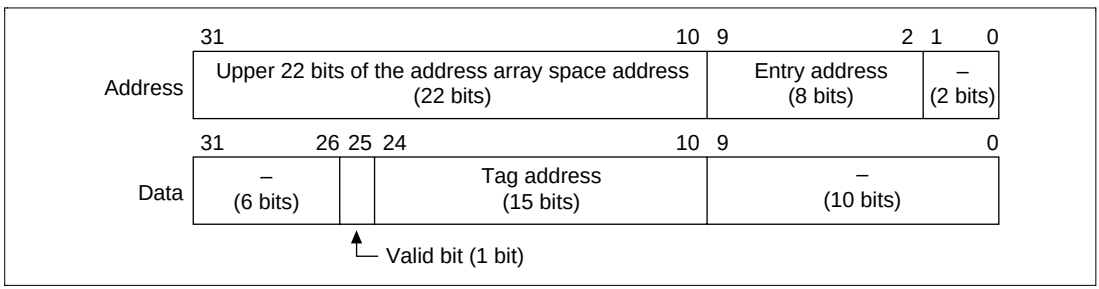


Figure 9.3 Cache Address Array

Address Array Read: Designates entry address and reads out the corresponding tag address value/valid bit value.

Address Array Write: Designates entry address and writes the designated tag address value/valid bit value.

9.3.2 Cache Data Array Read/Write Space

The cache data array has a compulsory read/write (figure 9.4).

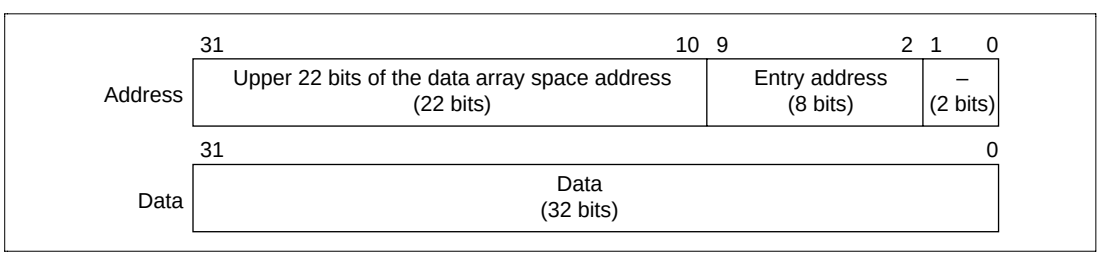


Figure 9.4 Cache Data Array

Data Array Read: Designates entry address and reads out the corresponding line of data.

Data Array Write: Designates entry address and writes designated data to the corresponding line.

9.4 Cautions on Use

9.4.1 Cache Initialization

Always initialize the cache before enabling it. Specifically, use an address array write to write 0 to all valid bits for all entries (256 times), that is,those in the address range H'FFFFF000–H'FFFFF3FF.

Writes to the address array or data array by the CPU, DMAC, or DTC are not possible while the cache is enabled. For reads, undefined values will be read out.

9.4.2 Forced Access to Address Array and Data Array

While the cache is enabled, it is not possible to write to the address array or data array via the CPU, DMAC, or DTC, and a read will return an undefined value. The cache must be disabled before making a forced access to the address array or data array.

9.4.3 Cache Miss Penalty and Cache Fill Timing

When a cache miss occurs, a single idle cycle is generated as a penalty immediately before the cache fill (access from external memory in the event of a cache miss), as shown in figure 9.5. However, in the case of consecutive cache misses, idle cycles are not generated for the second and subsequent cache misses, as shown in figure 9.6.

As the timing for a cache fill from normal space, the CS assert period immediately before the end of the bus cycle (or the last bus cycle when two or four bus cycles are generated, such as in a word access to 8-bit space) is extended by an additional cycle, as shown in figures 9.5 and 9.6.

Similarly, as the timing for a cache fill from DRAM space, the RAS assert period immediately before the end of the bus cycle is extended by an additional cycle. In RAS down mode, the next bus cycle is delayed by one cycle as shown in figure 9.8.

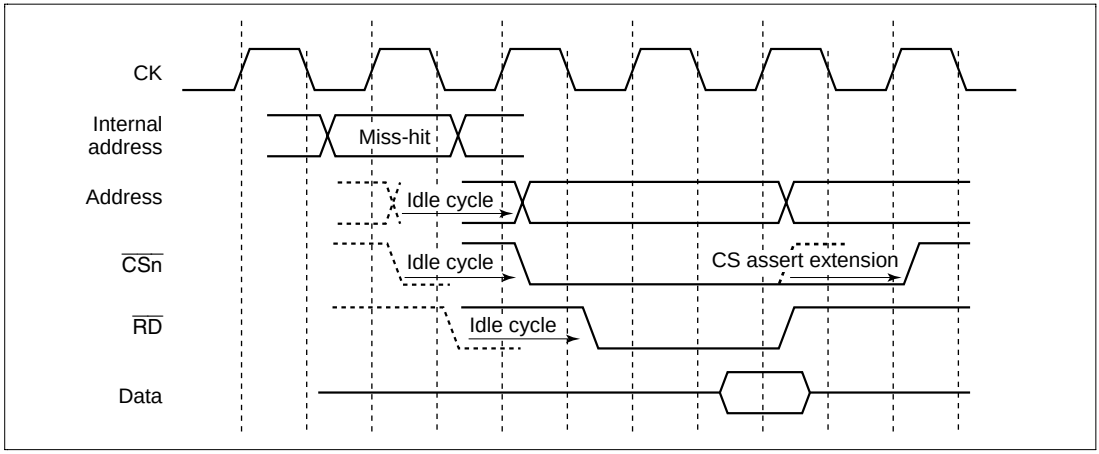


Figure 9.5 Cache Fill Timing in Case of Non-Consecutive Cache Miss from Normal Space (No Wait, No CS Assert Extension)

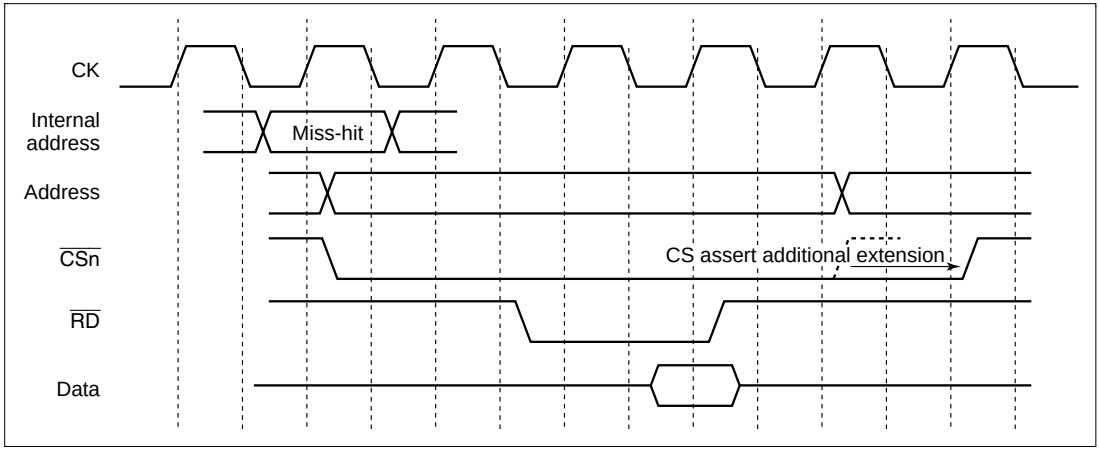


Figure 9.6 Cache Fill Timing in Case of Consecutive Cache Misses from Normal Space (No Wait, CS Assert Extension)

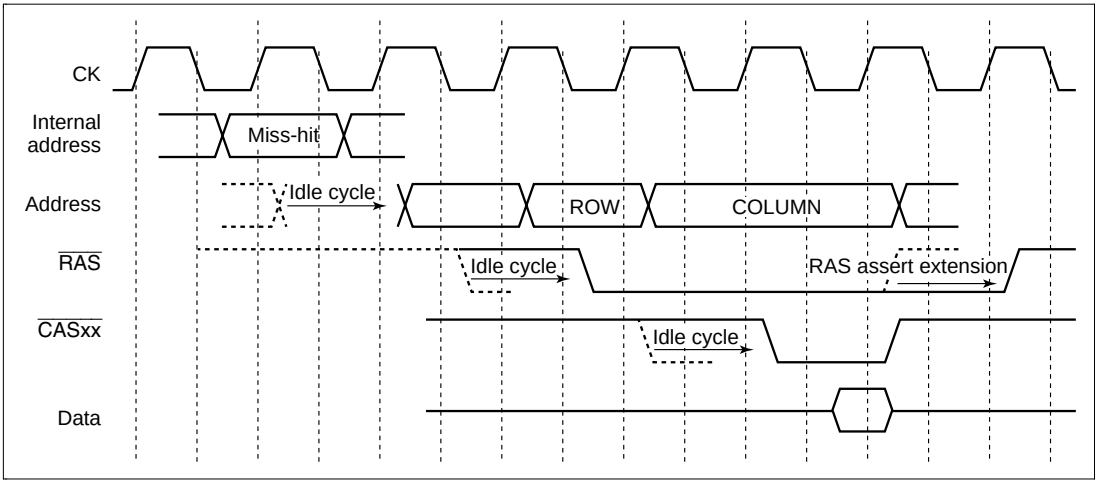


Figure 9.7 Cache Fill Timing in Case of Non-Consecutive Cache Miss from DRAM Space (Normal Mode, TPC = 0, RCD = 0, No Wait)

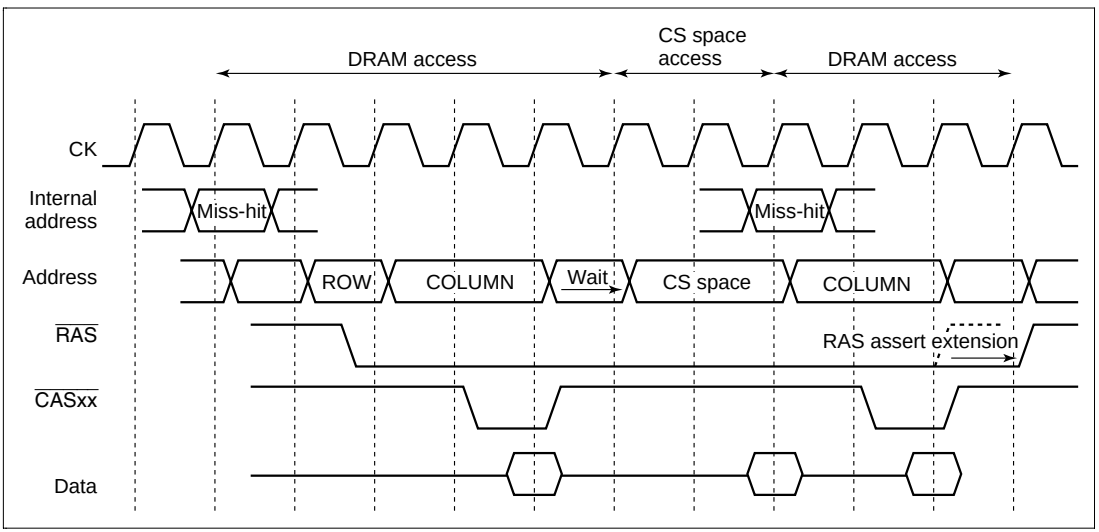


Figure 9.8 Cache Fill Timing in Case of Consecutive Cache Misses from DRAM Space (RAS Down Mode, TPC = 0, RCD = 0, No Wait)

9.4.4 Cache Hit after Cache Miss

The first cache hit after a cache miss is regarded as a cache miss, and a cache fill without idle cycle generation is performed. The next hit operates as a cache hit.

Section 10 Bus State Controller (BSC)

10.1 Overview

The bus state controller (BSC) divides up the address spaces and outputs control for various types of memory. This enables memories like DRAM, SRAM, and ROM to be linked directly to the LSI without external circuitry.

10.1.1 Features

The BSC has the following features:

- Address space is divided into five spaces
  - A maximum linear 2 Mbytes for on-chip ROM effective mode, and a maximum linear 4-Mbyte for on-chip ROM ineffective mode for address space CS0
  - A maximum linear 4 Mbytes for each of the address spaces CS1–CS3
  - A maximum linear 16 Mbytes for DRAM dedicated space
  - Bus width can be selected for each space (8, 16, or 32 bits)
  - Wait states can be inserted by software for each space
  - Wait states can be inserted via the $\overline{\text{WAIT}}$ pin in external memory spce accesses.
  - Outputs control signals for each space according to the type of memory connected
- On-chip ROM and RAM interfaces
  - On-chip RAM access of 32 bits in 1 state
  - On-chip ROM access of 32 bits in 1 state
- Direct interface to DRAM
  - Multiplexes row/column addresses according to DRAM capacity
  - Supports high-speed page mode and RAS down mode
- Access control for each type of memory, peripheral LSI
  - Address/data multiplex function
- Refresh
  - Supports CAS-before-RAS refresh (auto-refresh) and self-refresh
- Refresh counter can be used as an interval timer
  - Interrupt request generated upon compare match (CMI interrupt request signal)

10.1.2 Block Diagram

Figure 10.1 shows the BSC block diagram.

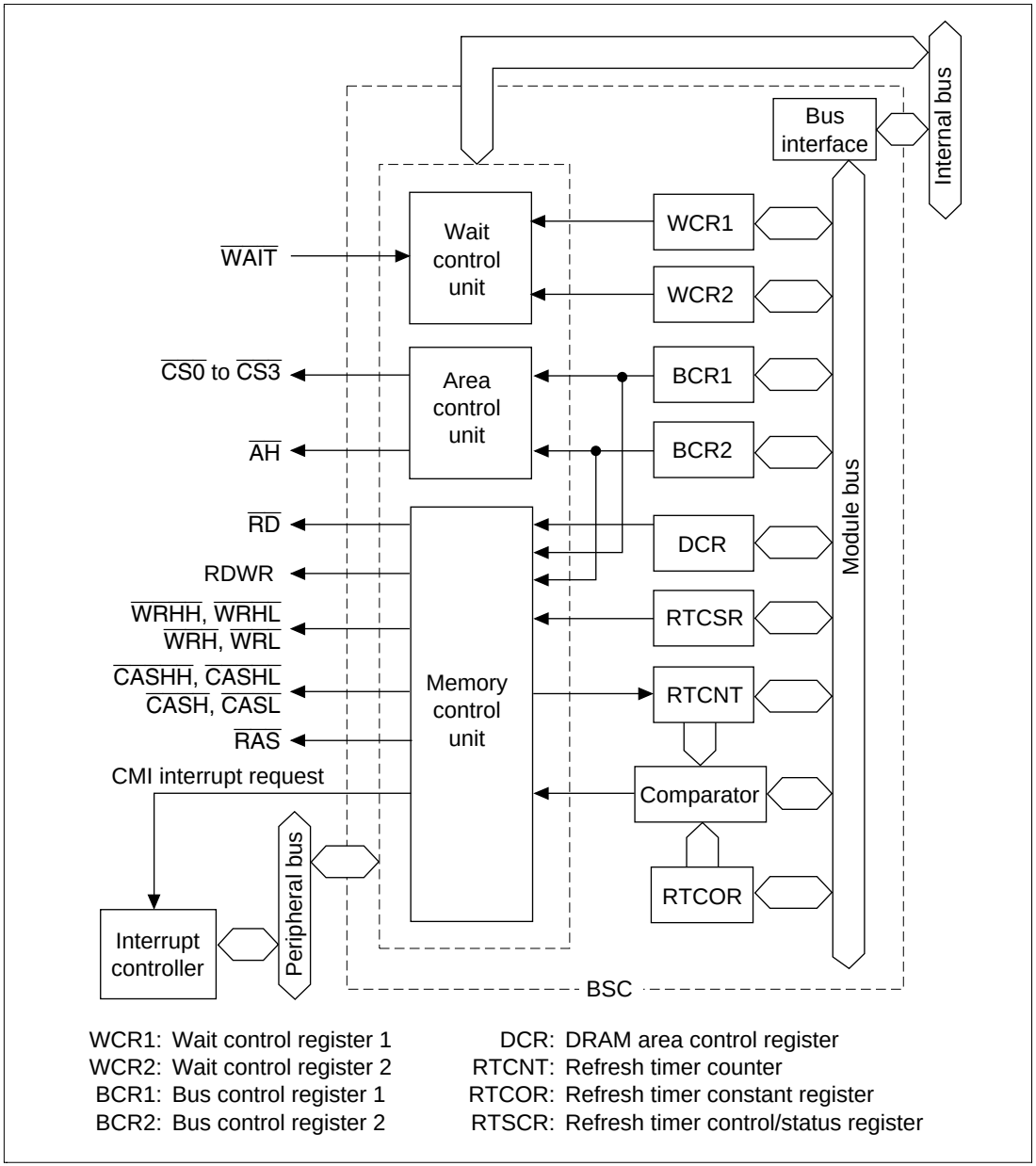


Figure 10.1 BSC Block Diagram

10.1.3 Pin Configuration

Table 10.1 shows the bus state controller pin configuration.

Table 10.1 Pin Configuration

Signal	I/O	Description
A21–A0	O	Address output (A21–A18 will become input ports with power-on reset)
D31–D0	I/O	32-bit data bus. D15-D0 are address output and data I/O during address/data multiplex I/O.
$\overline{CS0}$–$\overline{CS3}$	O	Chip select
$\overline{RD}$	O	Strobe that indicates the read cycle for ordinary space/multiplex I/O. Also output during DRAM access.
$\overline{WRHH}$	O	Strobe that indicates a write cycle to the most significant byte (D31–D24) for ordinary space/multiplex I/O. Also output during DRAM access.
$\overline{WRHL}$	O	Strobe that indicates a write cycle to the 2nd byte (D23–D16) for ordinary space/multiplex I/O. Also output during DRAM access.
$\overline{WRH}$	O	Strobe that indicates a write cycle to the 3rd byte (D15–D8) for ordinary space/multiplex I/O. Also output during DRAM access.
$\overline{WRL}$	O	Strobe that indicates a write cycle to the least significant byte (D7–D0) for ordinary space/multiplex I/O. Also output during DRAM access.
RDWR	O	Strobe indicating a write cycle to DRAM (used for DRAM space)
$\overline{RAS}$	O	RAS signal for DRAM (used for DRAM space)
$\overline{CASHH}$	O	CAS signal when accessing the most significant byte (D31–D24) of DRAM (used for DRAM space)
$\overline{CASHL}$	O	CAS signal when accessing the 2nd byte (D23–D16) of DRAM (used for DRAM space)
$\overline{CASH}$	O	CAS signal when accessing the 3rd byte (D15–D8) of DRAM (used for DRAM space)
$\overline{CASL}$	O	CAS signal when accessing the least significant byte (D7–D0) of DRAM (used for DRAM space)
$\overline{AH}$	O	Signal to hold the address during address/data multiplex
$\overline{WAIT}$	I	Wait state request signal
$\overline{BREQ}$	I	Bus release request input
$\overline{BACK}$	O	Bus use enable output

10.1.4 Register Configuration

The BSC has eight registers. These registers are used to control wait states, bus width, and interfaces with memories like DRAM, ROM, and SRAM, as well as refresh control. The register configurations are listed in table 10.2.

All registers are 16 bits. Do not access DRAM space before completing the memory interface settings. All BSC registers are all initialized by a power-on reset, but are not by a manual reset. Values are maintained in standby mode.

Table 10.2 Register Configuration

Name	Abbr.	R/W	Initial Value	Address	Access Size
Bus control register 1	BCR1	R/W	H'200F	H'FFFF8620	8, 16, 32
Bus control register 2	BCR2	R/W	H'FFFF	H'FFFF8622	8, 16, 32
Wait state control register 1	WCR1	R/W	H'FFFF	H'FFFF8624	8, 16, 32
Wait state control register 2	WCR2	R/W	H'000F	H'FFFF8626	8, 16, 32
DRAM area control register	DCR	R/W	H'0000	H'FFFF862A	8, 16, 32
Refresh timer control/status register	RTCSR	R/W	H'0000	H'FFFF862C	8, 16, 32
Refresh timer counter	RTCNT	R/W	H'0000	H'FFFF862E	8, 16, 32
Refresh time constant register	RTCOR	R/W	H'0000	H'FFFF8630	8, 16, 32

10.1.5 Address Map

Figure 10.2 shows the address format used by the SH7040 Series.

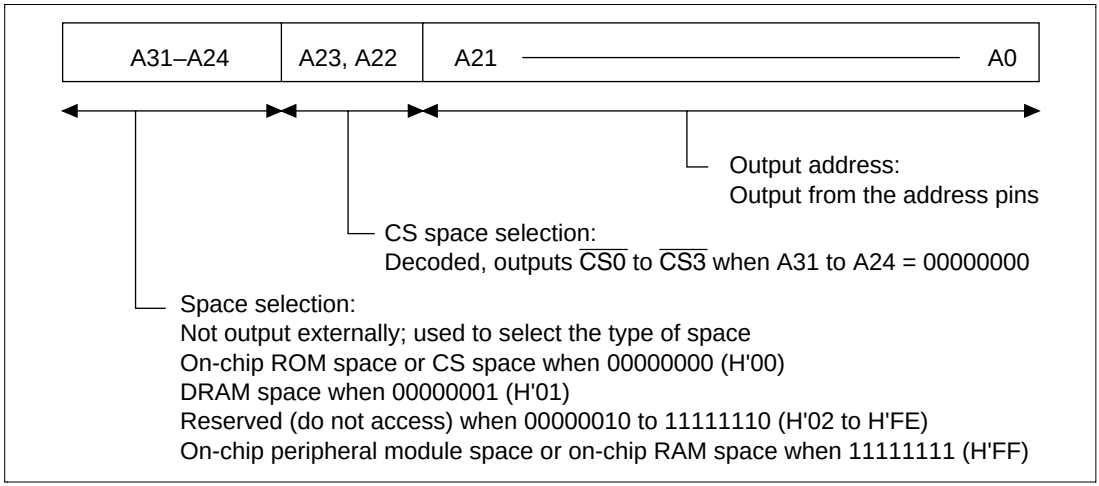


Figure 10.2 Address Format

This LSI uses 32-bit addresses:

- A31–A24 are used to select the type of space and are not output externally.
- Bits A23 and A22 are decoded and output as chip select signals ($\overline{CS0}$–$\overline{CS3}$) for the corresponding areas when bits A31–A24 are 00000000.
- A21–A0 are output externally.

Table 10.3 shows an address map for on-chip ROM effective mode. Table 10.4 shows an address map for on-chip ROM ineffective mode.

Table 10.3 Address Map for On-Chip ROM Effective Mode

Address	Space	Memory	Size	Bus Width
H'00000000–H'0003FFFF*1	On-chip ROM	On-chip ROM memory	256 kbytes	32 bits
H'00040000–H'001FFFFF	Reserved	Reserved		
H'00200000–H'003FFFFF	CS0 space	Ordinary space	2 Mbytes	8/16/32 bits*2
H'00400000–H'007FFFFF	CS1 space	Ordinary space	4 Mbytes	8/16/32 bits*2
H'00800000–H'00BFFFFF	CS2 space	Ordinary space	4 Mbytes	8/16/32 bits*2
H'00C00000–H'00FFFFFF	CS3 space	Ordinary space or multiplex I/O space	4 Mbytes	8/16/32 bits*3
H'01000000–H'01FFFFFF	DRAM space	DRAM	16 Mbytes	8/16/32 bits*2
H'02000000–H'FFFF7FFF	Reserved	Reserved		
H'FFFF8000–H'FFFF87FF	On-chip peripheral module	On-chip peripheral module	2 kbytes	8/16 bits
H'FFFF8800–H'FFFFEFFF	Reserved	Reserved		
H'FFFFF000–H'FFFFFFFF	On-chip RAM	On-chip RAM	4 kbytes	32 bits

Notes: Do not access reserved spaces. Operation cannot be guaranteed if they are accessed.

*1 With the 64-kbyte version of on-chip ROM, the ROM address is H'0000000–H'0000FFFF, and address H'00010000–H'0003FFFF is reserved space.

With the 128-kbyte version of on-chip ROM, the ROM address is H'00000000–H'0001FFFF, and address H'00020000–H'0003FFFF is reserved space.

*2 Selected by on-chip register settings.

*3 Ordinary space: selected by on-chip register settings.

Multiplex I/O space: 8/16 bit selected by the A14 bit.

Table 10.4 Address Map for On-Chip ROM Ineffective Mode

Address	Space	Memory	Size	Bus Width
H'00000000–H'003FFFFF	CS0 space	Ordinary space	4 Mbytes	8/16/32 bist*1
H'00400000–H'007FFFFF	CS1 space	Ordinary space	4 Mbytes	8/16/32 bits*2
H'00800000–H'00BFFFFF	CS2 space	Ordinary space	4 Mbytes	8/16/32 bits*2
H'00C00000–H'00FFFFFF	CS3 space	Ordinary space or multiplex I/O space	4 Mbytes	8/16/32 bits*3
H'01000000–H'01FFFFFF	DRAM space	DRAM	16 Mbytes	8/16/32 bits*2
H'02000000–H'FFFF7FFF	Reserved	Reserved		
H'FFFF8000–H'FFFF87FF	On-chip peripheral module	On-chip peripheral module	2 kbytes	8/16 bits
H'FFFF8800–H'FFFFEFFF	Reserved	Reserved		
H'FFFFF000–H'FFFFFFFF	On-chip RAM	On-chip RAM	4 kbytes	32 bits

Notes: 1. Do not access reserved spaces. Operation cannot be guaranteed if they are accessed.
2. In the single-chip mode, spaces other than on-chip ROM, on-chip RAM and on-chip peripheral modules are unavailable.
*1 Selected by the mode pin:
8/16 bit when 112 pin and 120 pin.
16/32 bit when 144 pin.
*2 Selected by on-chip register settings.
*3 Ordinary space: selected by on-chip register settings.
Multiplex I/O space: 8/16 bit selected by the A14 bit.

10.2 Description of Registers

10.2.1 Bus Control Register 1 (BCR1)

BCR1 is a 16-bit read/write register that enables access to the MTU control register, selects multiplex I/O, and specifies the bus size of the CS spaces. With the 112-pin version (SH7040/SH7042/SH7044), and the 120-pin version (SH7040/SH7042), specify the bus width as word (16 bits) or less.

Write bits 8–0 of BCR1 during the initialization stage after a power-on reset, and do not change the values thereafter. In on-chip ROM effective mode, do not access any of the CS spaces until after completion of register initialization. In on-chip ROM ineffective mode, do not access any CS space other than CS0 until after completion of register initialization.

BCR1 is initialized by power-on resets to H'200F, but is not initialized by manual resets or software standbys.

Bit:	15	14	13	12	11	10	9	8
	—	—	MTU RWE	—	—	—	—	IOE
Initial value:	0	0	1	0	0	0	0	0
R/W:	R	R	R/W	R	R	R	R	R/W

Bit:	7	6	5	4	3	2	1	0
	A3LG	A2LG	A1LG	A0LG	A3SZ	A2SZ	A1SZ	A0SZ
Initial value:	0	0	0	0	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- Bits 15, 14, 12–9—Reserved: These bits always read as 0. The write value should always be 0.
- Bit 13—MTU Read/Write Enable (MTURWE): When this bit is 1, MTU control register access is enabled. See section 12, Multifunction Timer Pulse Unit (MTU), for details.

Bit 13 (MTURWE)	Description
0	MTU control register access is disabled
1	MTU control register access is enabled (initial value)

- Bit 8—Multiplex I/O Enable (IOE): Selects the use of CS3 space as ordinary space or address/data multiplex I/O space. A 0 selects ordinary space and a 1 selects address/data multiplex I/O space. When address/data multiplex I/O space is selected, the address and data are multiplexed and output from the data bus. When CS3 space is an address/data multiplex I/O space, bus size is decided by the A14 bit (A14 = 0: 8 bit, A14 = 1: 16 bit).

Bit 8 (IOE)	Description
0	CS3 space is ordinary space (initial value)
1	CS3 space is address/data multiplex I/O space

- Bit 7—CS3 Space Long Size Specification (A3LG): Specifies the CS3 space bus size. This is effective only when CS3 space is ordinary space. When CS3 space is an address/data multiplex I/O space, bus size is decided by the A14 bit.

Bit 7 (A3LG)	Description
0	According to the A3SZ bit specified value (initial value)
1	Longword (32 bit) size

- Bit 6—CS2 Space Long Size Specification (A2LG): Specifies the CS2 space bus size.

Bit 6 (A2LG)	Description
0	According to the A2SZ bit value (initial value)
1	Longword (32 bit) size

- Bit 5—CS1 Space Long Size Specification (A1LG): Specifies the CS1 space bus size.

Bit 5 (A1LG)	Description
0	According to the A1SZ bit value (initial value)
1	Longword (32 bit) size

- Bit 4—CS0 Space Long Size Specification (A0LG): Specifies the CS0 space bus size.

Bit 4 (A0LG)	Description
0	According to the A0SZ bit value (initial value)
1	Longword (32 bit) size

Note: A0LG is effective only in on-chip ROM effective mode. When in on-chip ROM ineffective mode, the CS0 space bus size is specified by the mode pin.

- Bit 3—CS3 Space Size Specification (A3SZ): Specifies the CS3 space bus size when A3LG = 0. This is effective only when CS3 space is ordinary space. When CS3 space is an address/data multiplex I/O space, bus size is decided by the A14 bit.

Bit 3 (A3SZ)	Description
0	Byte (8 bit) size
1	Word (16 bit) size (initial value)

Note: This bit is ignored when A3LG = 1; CS3 space bus size becomes longword (32 bit) (for ordinary space).

- Bit 2—CS2 Space Size Specification (A2SZ): Specifies the CS2 space bus size when A2LG = 0.

Bit 2 (A2SZ)	Description
0	Byte (8 bit) size
1	Word (16 bit) size (initial value)

Note: This bit is ignored when A2LG = 1; CS2 space bus size becomes longword (32 bit).

- Bit 1—CS1 Space Size Specification (A1SZ): Specifies the CS1 space bus size when A1LG = 0.

Bit 1 (A1SZ)	Description
0	Byte (8 bit) size
1	Word (16 bit) size (initial value)

Note: This bit is ignored when A1LG = 1; CS1 space bus size becomes longword (32 bit).

- Bit 0—CS0 Space Size Specification (A0SZ): Specifies the CS0 space bus size when A0LG = 0.

Bit 0 (A0SZ)	Description
0	Byte (8 bit) size
1	Word (16 bit) size (initial value)

Note: A0SZ is effective only in on-chip ROM effective mode. In on-chip ROM ineffective mode, the CS0 space bus size is specified by the mode pin. However, even in on-chip ROM effective mode, this bit is ignored when A0LG = 1; CS0 space bus size becomes longword (32 bit).

10.2.2 Bus Control Register 2 (BCR2)

BCR2 is a 16-bit read/write register that specifies the number of idle cycles and CS signal assert extension of each CS space.

BCR2 is initialized by power-on resets to H'FFFF, but is not initialized by manual resets or software standbys.

Bit:	15	14	13	12	11	10	9	8
	IW31	IW30	IW21	IW20	IW11	IW10	IW01	IW00
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	CW3	CW2	CW1	CW0	SW3	SW2	SW1	SW0
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- Bits 15–8—Idles between Cycles (IW31, IW30, IW21, IW20, IW11, IW10, IW01, IW00): These bits specify idle cycles inserted between consecutive accesses when the second one is to a different CS area after a read. Idles are used to prevent data conflict between ROM (and other memories, which are slow to turn the read data buffer off), fast memories, and I/O interfaces. Even when access is to the same area, idle cycles must be inserted when a read access is followed immediately by a write access. The idle cycles to be inserted comply with the area specification of the previous access. Refer to section 10.6, Waits between Access Cycles, for details.

  IW31, IW30 specify the idle between cycles for CS3 space; IW21, IW20 specify the idle between cycles for CS2 space; IW11, IW10 specify the idle between cycles for CS1 space and IW01, IW00 specify the idle between cycles for CS0 space.

Bit 15 (IW31)	Bit 14 (IW30)	Description
0	0	No idle cycle after accessing CS3 space
	1	Inserts one idle cycle after accessing CS3 space
1	0	Inserts two idle cycles after accessing CS3 space
	1	Inserts three idle cycles after accessing CS3 space (initial value)

Bit 13 (IW21)	Bit 12 (IW20)	Description
0	0	No idle cycle after accessing CS2 space
	1	Inserts one idle cycle
1	0	Inserts two idle cycles
	1	Inserts three idle cycles (initial value)

Bit 11 (IW11)	Bit 10 (IW10)	Description
0	0	No idle cycle after accessing CS1 space
	1	Inserts one idle cycle
1	0	Inserts two idle cycles
	1	Inserts three idle cycles (initial value)

Bit 9 (IW01)	Bit 8 (IW00)	Description
0	0	No idle cycle after accessing CS0 space
	1	Inserts one idle cycle
1	0	Inserts two idle cycles
	1	Inserts three idle cycles (initial value)

- Bits 7–4—Idle Specification for Continuous Access (CW3, CW2, CW1, CW0): The continuous access idle specification makes insertions to clearly delineate the bus intervals by once negating the $\overline{\text{CSn}}$ signal when doing consecutive accesses of the same CS space. When a write immediately follows a read, the number of idle cycles inserted is the larger of the two values specified by IW and CW. Refer to section 10.6, Waits between Access Cycles, for details.

  CW3 specifies the continuous access idles for CS3 space; CW2 specifies the continuous access idles for CS2 space; CW1 specifies the continuous access idles for CS1 space and CW0 specifies the continuous access idles for CS0 space.

Bit 7 (CW3)	Description
0	No CS3 space continuous access idle cycles
1	One CS3 space continuous access idle cycle (initial value)

Bit 6 (CW2)	Description
0	No CS2 space continuous access idle cycles
1	One CS2 space continuous access idle cycle (initial value)

Bit 5 (CW1)	Description
0	No CS1 space continuous access idle cycles
1	One CS1 space continuous access idle cycle (initial value)

Bit 4 (CW0)	Description
0	No CS0 space continuous access idle cycles
1	One CS0 space continuous access idle cycle (initial value)

- Bits 3–0—$\overline{CS}$ Assert Extension Specification (SW3, SW2, SW1, SW0): The $\overline{CS}$ assert cycle extension specification is for making insertions to prevent extension of the $\overline{RD}$ signal or $\overline{WRx}$ signal assert period beyond the length of the $\overline{CSn}$ signal assert period. Extended cycles insert one cycle before and after each bus cycle, which simplifies interfaces with external devices and also has the effect of extending write data hold time. Refer to section 10.3.3, $\overline{CS}$ Assert Period Extension, for details.

  SW3 specifies the $\overline{CS}$ assert extension for CS3 space access; SW2 specifies the $\overline{CS}$ assert extension for CS2 space access; SW1 specifies the $\overline{CS}$ assert extension for CS1 space access and SW0 specifies the $\overline{CS}$ assert extension for CS0 space access.

Bit 3 (SW3)	Description
0	No CS3 space $\overline{CS}$ assert extension
1	CS3 space $\overline{CS}$ assert extension (initial value)

Bit 2 (SW2)	Description
0	No CS2 space $\overline{CS}$ assert extension
1	CS2 space $\overline{CS}$ assert extension (initial value)

Bit 1 (SW1)	Description
0	No CS1 space $\overline{CS}$ assert extension
1	CS1 space $\overline{CS}$ assert extension (initial value)

Bit 0 (SW0)	Description
0	No CS0 space $\overline{CS}$ assert extension
1	CS0 space $\overline{CS}$ assert extension (initial value)

10.2.3 Wait Control Register 1 (WCR1)

WCR1 is a 16-bit read/write register that specifies the number of wait cycles (0–15) for each CS space.

WCR1 is initialized by power-on resets to H'FFFF, but is not initialized by manual resets or software standbys.

Bit:	15	14	13	12	11	10	9	8
	W33	W32	W31	W30	W23	W22	W21	W20
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	W13	W12	W11	W10	W03	W02	W01	W00
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- Bits 15–12—CS3 Space Wait Specification (W33, W32, W31, W30): Specifies the number of waits for CS3 space access.

Bit 15 (W33)	Bit 14 (W32)	Bit 13 (W31)	Bit 12 (W30)	Description
0	0	0	0	No wait (external wait input disabled)
0	0	0	1	1 wait external wait input enabled
	...			
1	1	1	1	15 wait external wait input enabled (initial value)

- Bits 11–8—CS2 Space Wait Specification (W23, W22, W21, W20): Specifies the number of waits for CS2 space access.

Bit 11 (W23)	Bit 10 (W22)	Bit 9 (W21)	Bit 8 (W20)	Description
0	0	0	0	No wait (external wait input disabled)
0	0	0	1	1 wait external wait input enabled
	...			
1	1	1	1	15 wait external wait input enabled (initial value)

- Bits 7–4—CS1 Space Wait Specification (W13, W12, W11, W10): Specifies the number of waits for CS1 space access.

Bit 7 (W13)	Bit 6 (W12)	Bit 5 (W11)	Bit 4 (W10)	Description
0	0	0	0	No wait (external wait input disabled)
0	0	0	1	1 wait external wait input enabled
	...			
1	1	1	1	15 wait external wait input enabled (initial value)

- Bits 3–0—CS0 Space Wait Specification (W03, W02, W01, W00): Specifies the number of waits for CS0 space access.

Bit 3 (W03)	Bit 2 (W02)	Bit 1 (W01)	Bit 0 (W00)	Description
0	0	0	0	No wait (external wait input disabled)
0	0	0	1	1 wait external wait input enabled
	...			
1	1	1	1	15 wait external wait input enabled (initial value)

10.2.4 Wait Control Register 2 (WCR2)

WCR2 is a 16-bit read/write register that specifies the number of access cycles for DRAM space and CS space for DMA single address mode transfers.

Do not perform any DMA single address transfers before WCR2 is set.

WCR2 is initialized by power-on resets to H'000F, but is not initialized by manual resets or software standbys.

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	—	—	DDW1	DDW0	DSW3	DSW2	DSW1	DSW0
Initial value:	0	0	0	0	1	1	1	1
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W

- Bits 15–6—Reserved: These bits always read as 0. The write value should always be 0.
- Bits 5–4—DRAM Space DMA Single Address Mode Access Wait Specification (DDW1, DDW0): Specifies the number of waits for DRAM space access during DMA single address mode accesses. These bits are independent of the DWW and DWR bits of the DCR.

Bit 5 (DDW1)	Bit 4 (DDW0)	Description
0	0	2-cycle (no wait) external wait disabled (initial value)
	1	3-cycle (1 wait) external wait disabled
1	0	4-cycle (2 wait) external wait enabled
	1	5-cycle (3 wait) external wait enabled

- Bits 3–0—CS Space DMA Single Address Mode Access Wait Specification (DSW3, DSW2, DSW1, DSW0): Specifies the number of waits for CS space access (0–15) during DMA single address mode accesses. These bits are independent of the W bits of the WCR1.

Bit 3 (DSW3)	Bit 2 (DSW2)	Bit 1 (DSW1)	Bit 0 (DSW0)	Description
0	0	0	0	No wait (external wait input disabled)
0	0	0	1	1 wait (external wait input enabled)
	...			
1	1	1	1	15 wait (external wait input enabled) (initial value)

10.2.5 DRAM Area Control Register (DCR)

DCR is a 16-bit read/write register that selects the number of waits, operation mode, number of address multiplex shifts and the like for DRAM control.

Do not perform any DRAM space accesses before DCR initial settings are completed.

DCR is initialized by power-on resets to H'0000, but is not initialized by manual resets or software standbys.

Bit:	15	14	13	12	11	10	9	8
	TPC	RCD	TRAS1	TRAS0	DWW1	DWW0	DWR1	DWR0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	DIW	—	BE	RASD	SZ1	SZ0	AMX1	AMX0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

- Bit 15—RAS Precharge Cycle Count (TPC): Specifies the minimum number of cycles after $\overline{\text{RAS}}$ is negated before next assert.

Bit 15 (TPC)	Description
0	1.5 cycles (initial value)
1	2.5 cycles

- Bit 14—RAS-CAS Delay Cycle Count (RCD): Specifies the number of row address output cycles.

Bit 14 (RCD)	Description
0	1 cycle (initial value)
1	2 cycles

- Bits 13–12—CAS-Before-RAS Refresh RAS Assert Cycle Count (TRAS1–TRAS0): Specify the number of RAS assert cycles for CAS before RAS refreshes.

Bit 13 (TRAS1)	Bit 12 (TRAS0)	Description
0	0	2.5 cycles (initial value)
	1	3.5 cycles
1	0	4.5 cycles
	1	5.5 cycles

- Bits 11–10—DRAM Write Cycle Wait Count (DWW1–DWW0): Specifies the number of DRAM write cycle column address output cycles.

Bit 11 (DWW1)	Bit 10 (DWW0)	Description
0	0	2-cycle (no wait) external wait disabled (initial value)
	1	3-cycle (1 wait) external wait disabled
1	0	4-cycle (2 wait) external wait enabled
	1	5-cycle (3 wait) external wait enabled

- Bits 9–8—DRAM Read Cycle Wait Count (DWR1–DWR0): Specifies the number of DRAM read cycle column address output cycles.

Bit 9 (DWR1)	Bit 8 (DWR0)	Description
0	0	2-cycle (no wait) external wait disabled (initial value)
	1	3-cycle (1 wait) external wait disabled
1	0	4-cycle (2 wait) external wait enabled
	1	5-cycle (3 wait) external wait enabled

- Bit 7—DRAM Idle Cycle Count (DIW): Specifies whether to insert idle cycles, either when accessing a different external space (CS space) or when doing a DRAM write, after DRAM reads.

Bit 7 (DIW)	Description
0	No idle cycles (initial value)
1	1 idle cycle

- Bit 6—Reserved: This bit always reads as 0. The write value should always be 0.
- Bit 5—Burst Enable (BE): Specifies the DRAM operation mode.

Bit 5 (BE)	Description
0	Burst disabled (initial value)
1	DRAM high-speed page mode enabled.

- Bit 4—RAS Down Mode (RASD): Specifies the DRAM operation mode.

Bit 4 (RASD)	Description
0	Access DRAM by RAS up mode (initial value)
1	Access DRAM by RAS down mode

- Bits 3–2—DRAM Bus Width Specification (SZ1, SZ0): Specifies the DRAM space bus width.

Bit 3 (SZ1)	Bit 2 (SZ0)	Description
0	0	Byte (8 bits) (initial value)
	1	Word (16 bits)
1	Don't care	Longword (32 bits)

- Bits 1–0—DRAM Address Multiplex (AMX1–AMX0): Specifies the DRAM address multiplex count.

Bit 1 (AMX1)	Bit 0 (AMX0)	Description
0	0	9 bit (initial value)
	1	10 bit
1	0	11 bit
	1	12 bit

10.2.6 Refresh Timer Control/Status Register (RTCSR)

RTCSR is a 16-bit read/write register that selects the refresh mode and the clock input to the refresh timer counter (RTCNT), and controls compare match interrupts (CMI).

RTCSR is initialized by power-on resets and hardware standbys to H'0000, but is not initialized by manual resets or software standbys.

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	—	CMF	CMIE	CKS2	CKS1	CKS0	RFSH	RMD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- Bits 15–7—Reserved: These bits always read as 0. The write value should always be 0.

- Bit 6—Compare Match Flag (CMF): This status flag, which indicates that the values of RTCNT and RTCOR match, is set/cleared under the following conditions:

Bit 6 (CMF)	Description
0	Clear condition: After RTCSR is read when CMF is 1, 0 is written in CMF. In some cases it will clear when DTC is activated by a compare match interrupt; refer to section 8, Data Transfer Controller (DTC), for details. (initial value)
1	Set condition: RTCNT = RTCOR. When both RTCNT and RTCOR are in an initialized state (when values have not been rewritten since initialization, and RTCNT has not had its value changed due to a count-up), RTCNT and RTCOR match, as both are H'0000, but in this case CMF is not set.

- Bit 5—Compare Match Interrupt Enable (CMIE): Enables or disables an interrupt request caused by the CMF bit of the RTCSR when CMF is set to 1.

Bit 5 (CMIE)	Description
0	Disables an interrupt request caused by CMF (initial value)
1	Enables an interrupt request caused by CMF

- Bits 4–2—Clock Select (CKS2–CKS0): Select the clock to input to RTCNT from among the seven types of internal clock obtained from dividing the system clock (ϕ).

Bit 4 (CKS2)	Bit 3 (CKS1)	Bit 2 (CKS0)	Description
0	0	0	Stops count-up (initial value)
		1	ϕ/2
	1	0	ϕ/8
		1	ϕ/32
1	0	0	ϕ/128
		1	ϕ/512
	1	0	ϕ/2048
		1	ϕ/4096

- Bit 1—Refresh Control (RFSH): Selects whether to use refresh control for DRAM.

Bit 1 (RFSH)	Description
0	Do not refresh DRAM (initial value)
1	Refresh DRAM

- Bit 0—Refresh Mode (RMD): When the RFSH bit is 1, this bit selects normal refresh or self-refresh. When the RFSH bit is 1, self-refresh mode is entered immediately after the RMD bit is set to 1. When RMD is cleared to 0, a CAS-before-RAS refresh is performed at the interval set in the refresh time constant register (RTCNT).

  When set for self-refresh, the SH7040 Series enters self-refresh mode immediately unless it is in the middle of a DRAM access. If it is, it enters self-refresh mode when the access ends. Refresh requests from the interval timer are ignored in self-refresh mode.

Bit 0 (RMD)	Description
0	CAS-before-RAS refresh (initial value)
1	Self-refresh

10.2.7 Refresh Timer Counter (RTCNT)

RTCNT is a 16-bit read/write register that is used as an 8-bit up counter for refreshes or generating interrupt requests.

RTCNT counts up with the clock selected by the CKS2–CKS0 bits of the RTCSR. RTCNT values can always be read/written by the CPU. When RTCNT matches RTCOR, RTCNT is cleared to H'0000 and the CMF flag of the RTCSR is set to 1. If the RFSH bit of RTCSR is 1 and the RMD bit is 0 at this time, a CAS-before-RAS refresh is performed. Additionally, if the CMIE bit of RTCSR is a 1, a compare match interrupt (CMI) is generated.

Bits 15–8 are reserved and play no part in counter operation. They are always read as 0.

RTCNT is initialized by power-on resets H'0000, but is not initialized by manual resets or software standbys.

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

10.2.8 Refresh Time Constant Register (RTCOR)

RTCOR is a 16-bit read/write register that establishes the compare match period with RTCNT. The values of RTCOR and RTCNT are constantly compared. When the values correspond, the compare match flag of RTCSR is set and RTCNT is cleared to 0.

When the refresh bit (RFSH) of the RTCSR is set to 1 and the RMD bit is 0, a refresh request signal is produced by this match. The refresh request signal is held until a refresh operation is performed. If the refresh request is not processed before the next match, the previous request becomes ineffective.

When the CMIE bit of the RTSCR is set to 1, an interrupt request is sent to the interrupt controller by this match signal. The interrupt request is output continuously until the CMF bit of the RTSCR is cleared.

Bits 15–8 are reserved and cannot be used in setting the period. They always read 0.

RTCOR is initialized by power-on resets to H'0000, but is not initialized by manual resets or software standbys.

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

10.3 Accessing Ordinary Space

A strobe signal is output by ordinary space accesses to provide primarily for SRAM or ROM direct connections.

10.3.1 Basic Timing

Figure 10.3 shows the basic timing of ordinary space accesses. Ordinary access bus cycles are performed in 2 states.

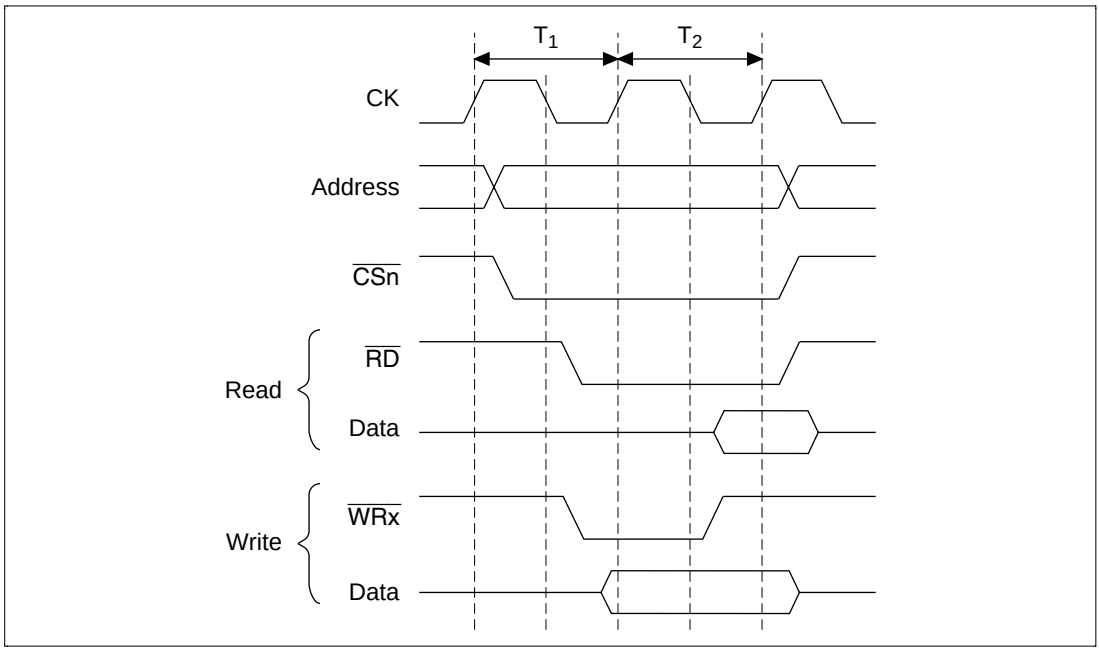


Figure 10.3 Basic Timing of Ordinary Space Access

During a read, irrespective of operand size, all bits in the data bus width for the access space (address) are fetched by the LSI on $\overline{RD}$, using the required byte locations.

During a write, the following signals are associated with transfer of these actual byte locations: $\overline{WRHH}$ (bits 31–24), $\overline{WRHL}$ (bits 23–16), $\overline{WRH}$ (bits 15–8), and $\overline{WRL}$ (bits 7–0).

10.3.2 Wait State Control

The number of wait states inserted into ordinary space access states can be controlled using the WCR settings (figure 10.4).

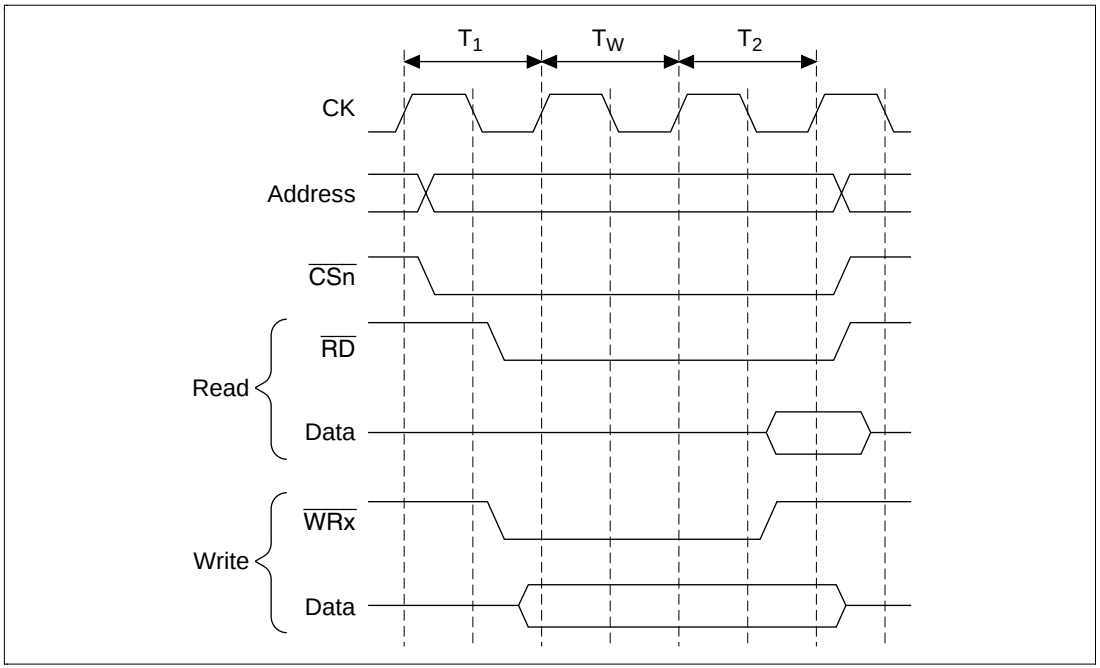


Figure 10.4 Wait Timing of Ordinary Space Access (Software Wait Only)

When the wait is specified by software using WCR, the wait input $\overline{\text{WAIT}}$ signal from outside is sampled. Figure 10.5 shows the $\overline{\text{WAIT}}$ signal sampling. The $\overline{\text{WAIT}}$ signal is sampled at the clock rise one cycle before the clock rise when T_w state shifts to T_2 state.

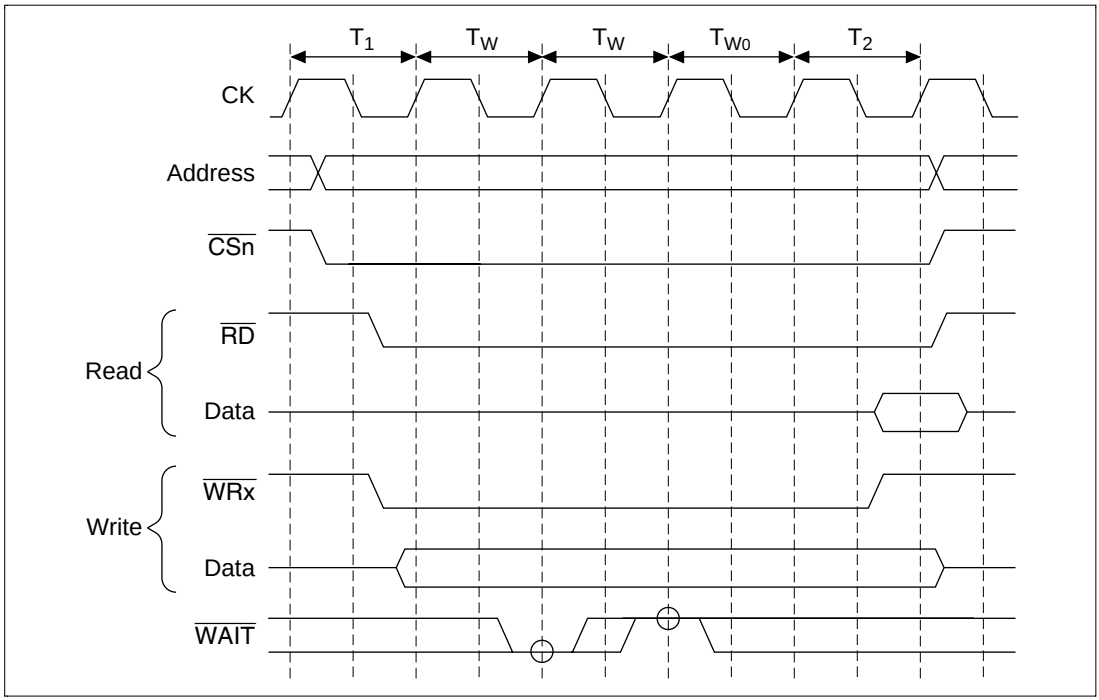


Figure 10.5 Wait State Timing of Ordinary Space Access (Wait States from Software Wait 2 State + $\overline{\text{WAIT}}$ Signal)

10.3.3 $\overline{CS}$ Assert Period Extension

Idle cycles can be inserted to prevent extension of the $\overline{RD}$ signal or $\overline{WRx}$ signal assert period beyond the length of the $\overline{CSn}$ signal assert period by setting the SW3–SW0 bits of BCR2. This allows for flexible interfaces with external circuitry. The timing is shown in figure 10.6. T_h and T_f cycles are added respectively before and after the ordinary cycle. Only $\overline{CSn}$ is asserted in these cycles; $\overline{RD}$ and $\overline{WRx}$ signals are not. Further, data is extended up to the T_f cycle, which is effective for gate arrays and the like, which have slower write operations.

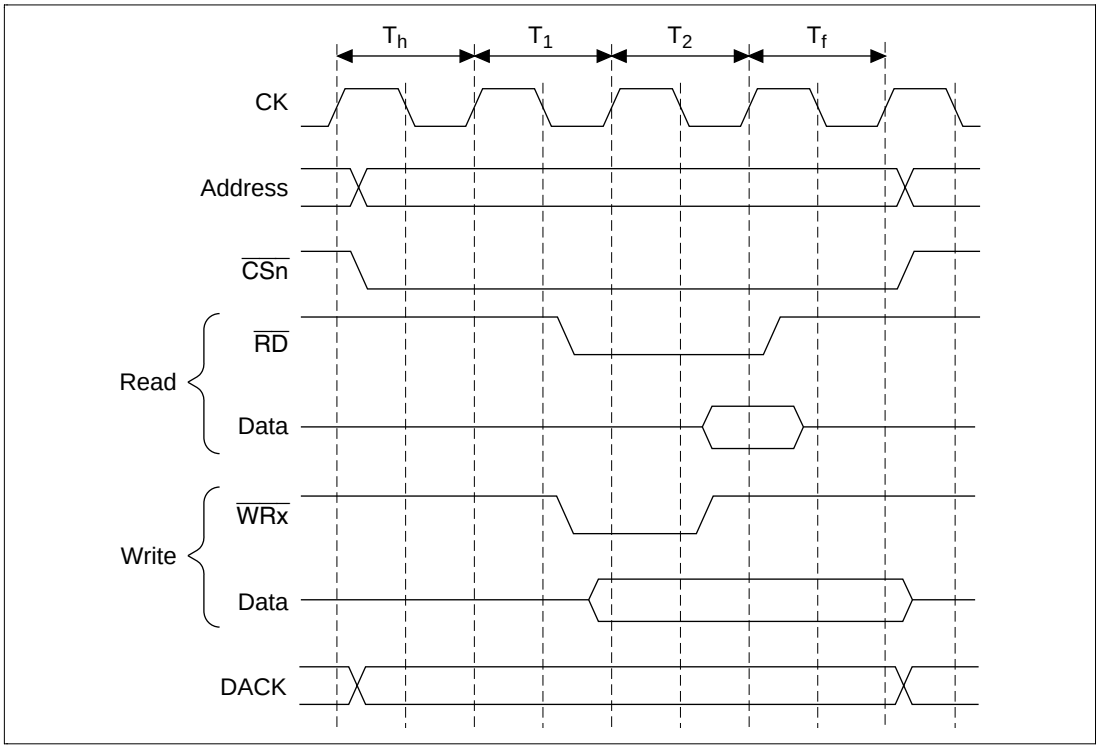


Figure 10.6 $\overline{CS}$ Assert Period Extension Function

10.4 DRAM Access

10.4.1 DRAM Direct Connection

When address space A31–A24 = H'01 has been accessed, the corresponding space becomes a 16-Mbyte DRAM space, and the DRAM interface function can be used to directly connect the SH7040 Series to DRAM.

Row address and column address are always multiplexed for DRAM space. The amount of row address multiplexing can be selected as from 9 to 12 bits by setting the AMX1 and AMX0 bits of the DCR.

Table 10.5 AMX Bits and Address Multiplex Output

			Row Address		Column Address	
AMX1	**AMX0**	**Shift Amount**	**Output Pins**	**Output Address**	**Output Address**	**Output Pins**
0	0	9 bit	A21–A15	A21–A15	A21–A0	A21–A0
			A14–A0	A23–A9		
0	1	10 bit	A21–A14	A21–A14	A21–A0	A21–A0
			A13–A0	A23–A10		
1	0	11 bit	A21–A13	A21–A13	A21–A0	A21–A0
			A12–A0	A23–A11		
1	1	12 bit	A21–A12	A21–A12	A21–A0	A21–A0
			A11–A0	A23–A12		

In addition to ordinary read and write accesses, burst mode access using high speed page mode is supported.

10.4.2 Basic Timing

The SH7040 Series supports 2 CAS format DRAM access. The DRAM access basic timing is a minimum of 3 cycles for normal mode. Figure 10.7 shows the basic DRAM access timing. DRAM space access is controlled by $\overline{RAS}$, $\overline{CASx}$, and RDWR signals. The following signals are associated with transfer of these actual byte locations: $\overline{CASHH}$ (bits 31–24), $\overline{CASHL}$ (bits 23–16), $\overline{CASH}$ (bits 15–8), and $\overline{CASL}$ (bits 7–0). However, the signals for ordinary space, $\overline{WRx}$ and $\overline{RD}$, are also output during the DMAC single transfer column address cycle period. T_p is the precharge cycle, T_r is the RAS assert cycle, T_c is the CAS assert cycle and T_{c2} is the read data fetch cycle.

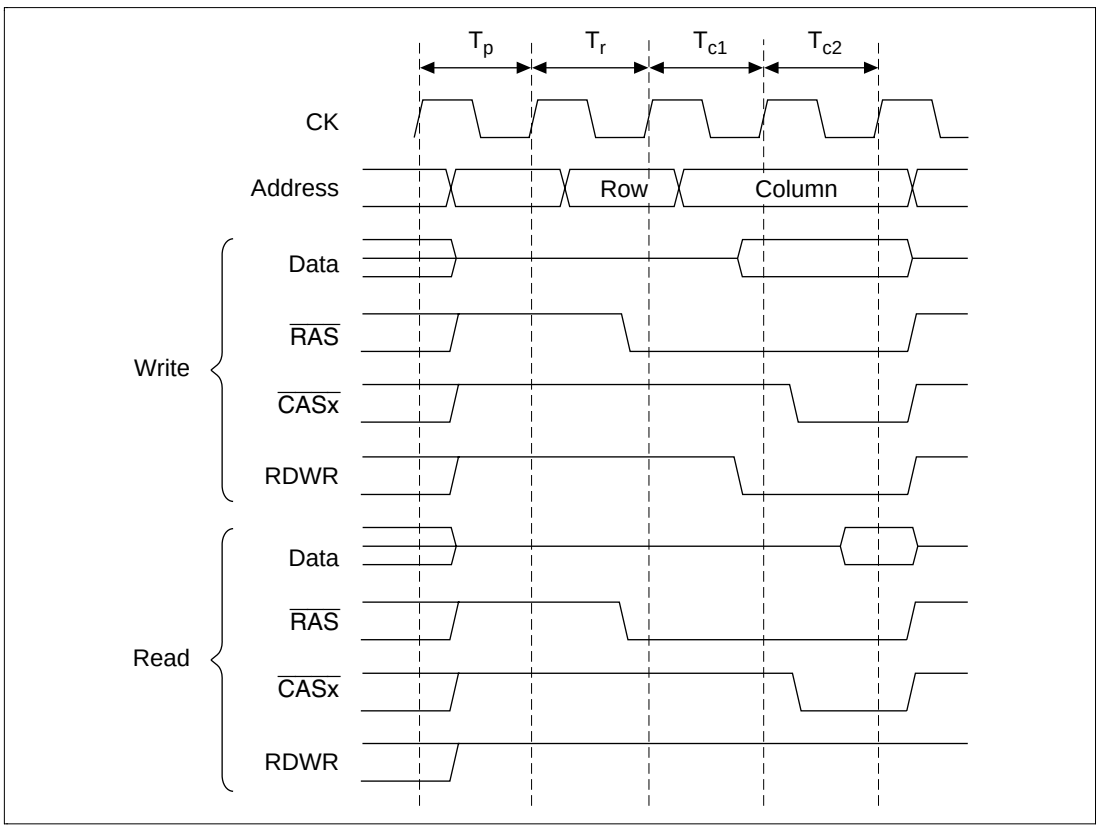


Figure 10.7 DRAM Bus Cycle (Normal Mode, TPC = 0, RCD = 0, No Waits)

10.4.3 Wait State Control

Wait state insertion during DRAM space access is controlled by setting the TPC, RCD, DWW1, DWW0, DWR1, and DWR0 bits of the DCR. TPC and RCD are common to both reads and writes. The timing with waits inserted is shown in figures 10.8 through 10.11. External waits can be inserted at the time of software waits 2, 3. The sampling location is the same as that of ordinary space: at one cycle before the T_{c2} cycle clock rise. Wait cycles are extended by external waits.

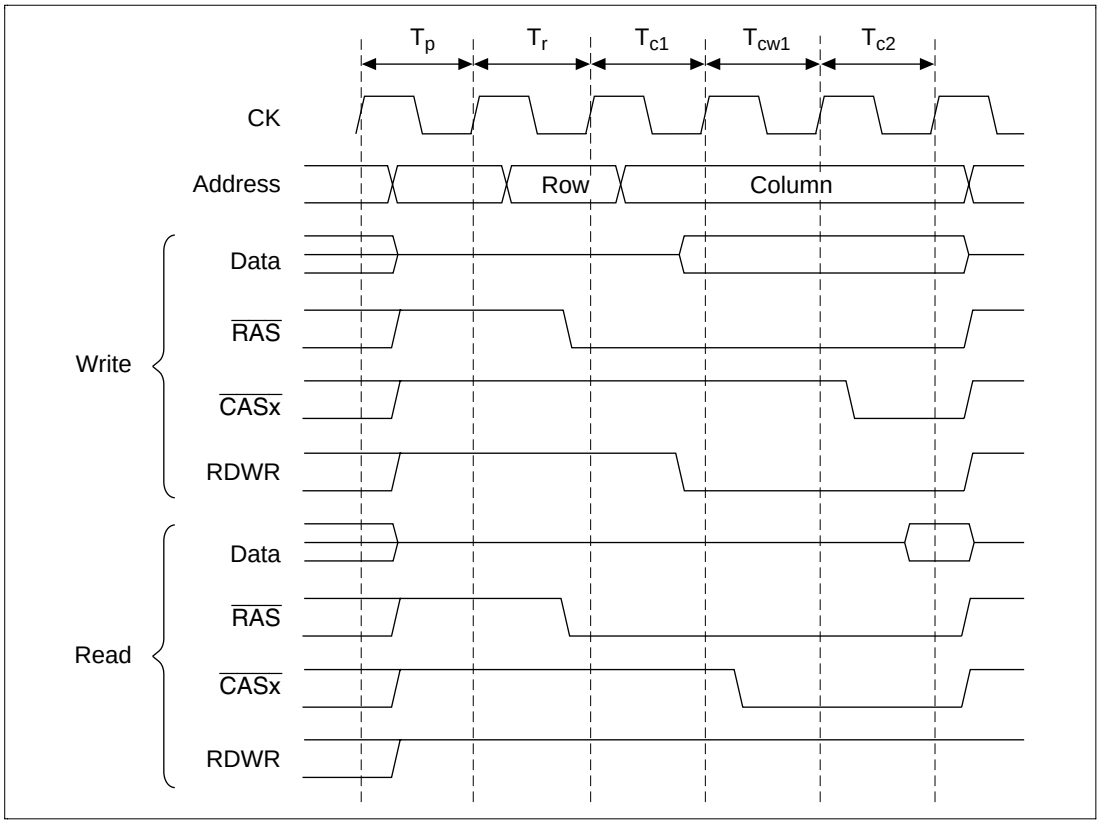


Figure 10.8 DRAM Bus Cycle (Normal Mode, TPC = 0, RCD = 0, One Wait)

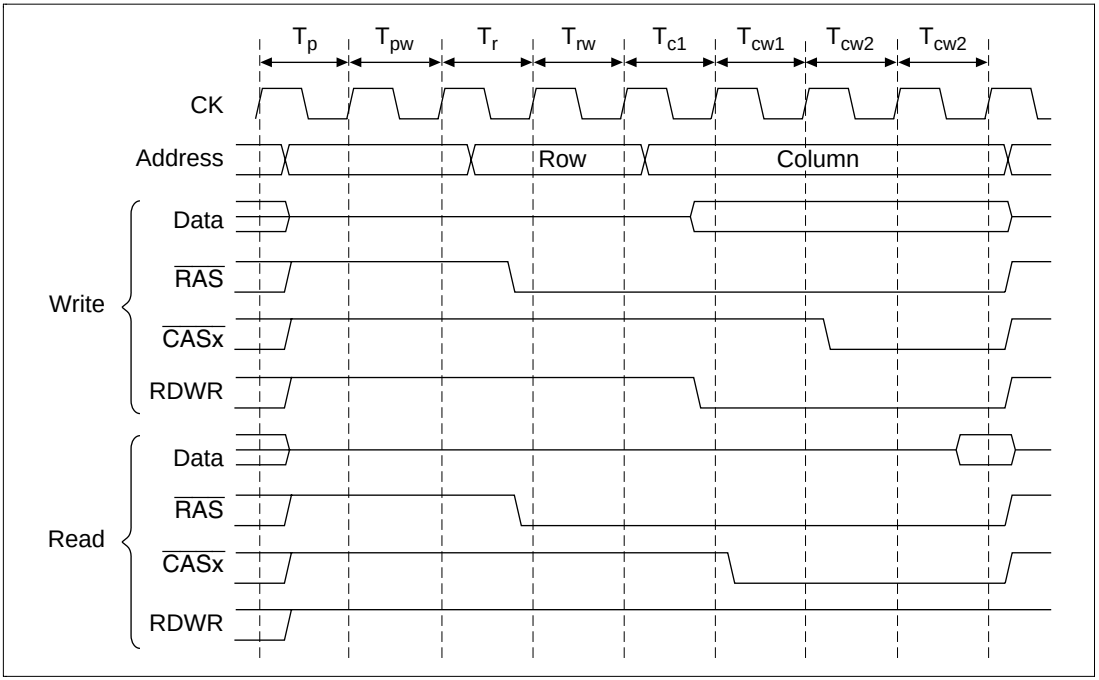


Figure 10.9 DRAM Bus Cycle (Normal Mode, TPC = 1, RCD = 1, Two Waits)

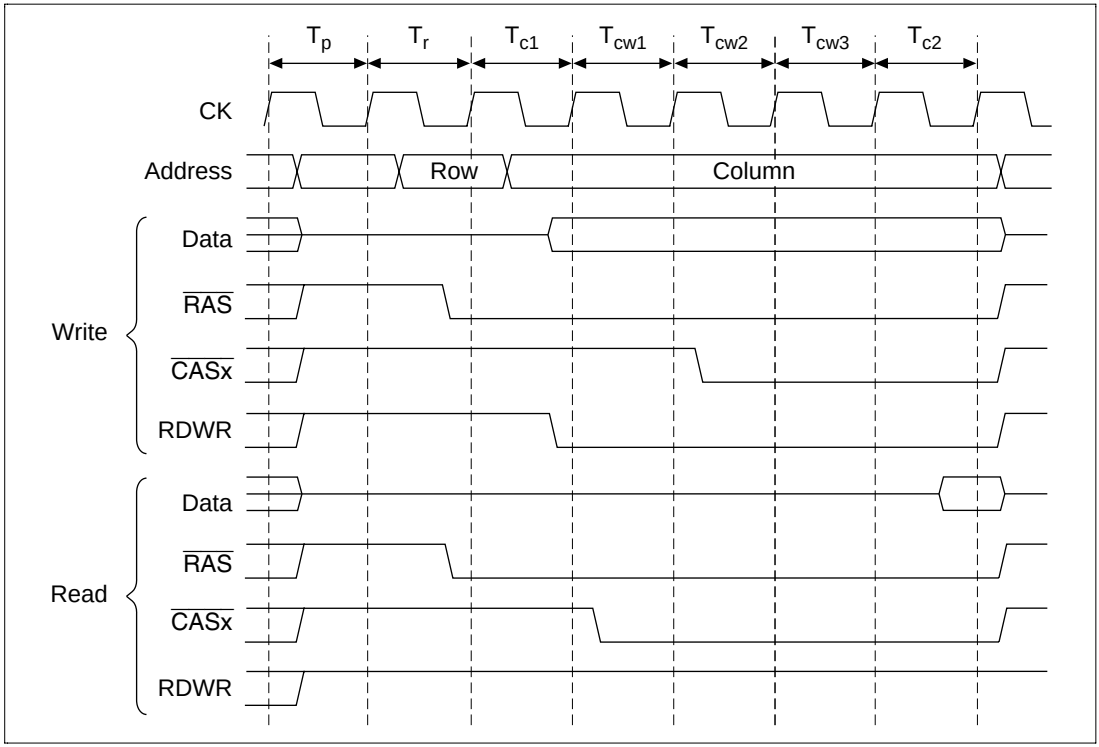


Figure 10.10 DRAM Bus Cycle (Normal Mode, TPC = 0, RCD = 0, Three Waits)

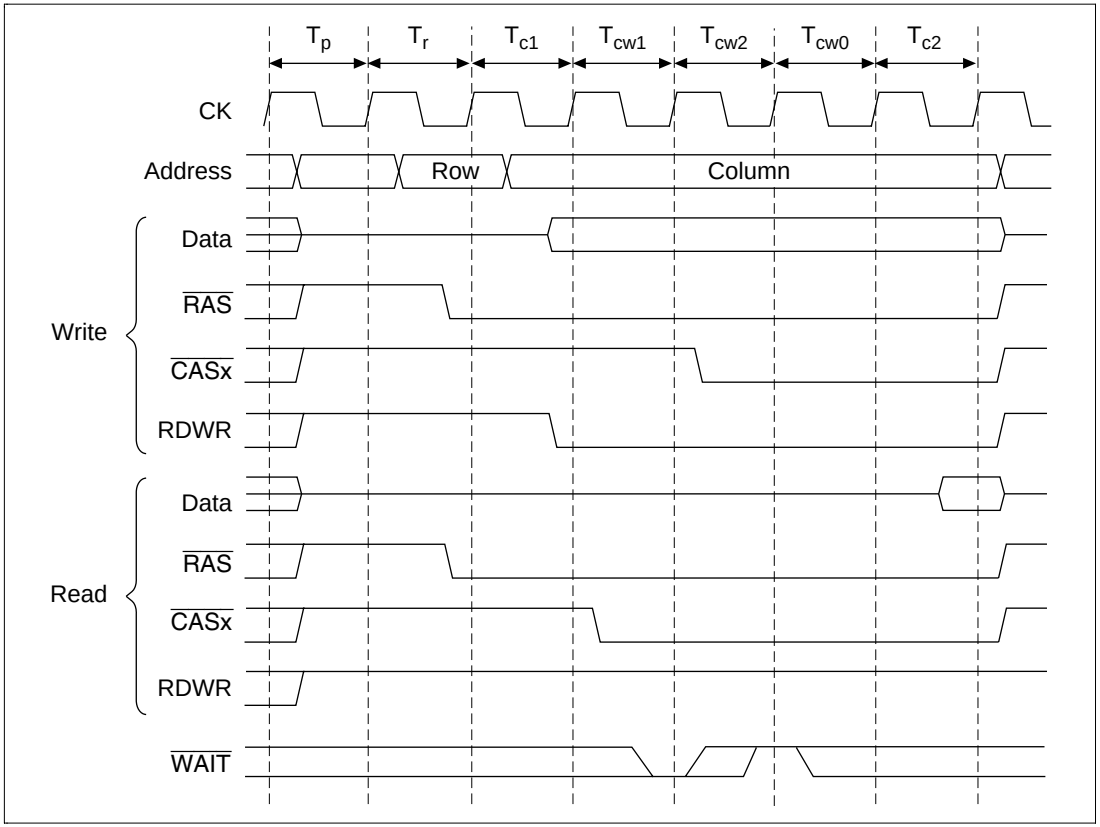


Figure 10.11 DRAM Bus Cycle (Normal Mode, TPC = 0, RCD= 0, Two Waits + Wait Due to $\overline{\text{WAIT}}$ Signal)

10.4.4 Burst Operation

High-Speed Page Mode: When the burst enable bit (BE) of the DCR is set, burst accesses can be performed using high speed page mode. The timing is shown in figure 10.12. Wait cycles can be inserted during burst accesses by using the DCR.

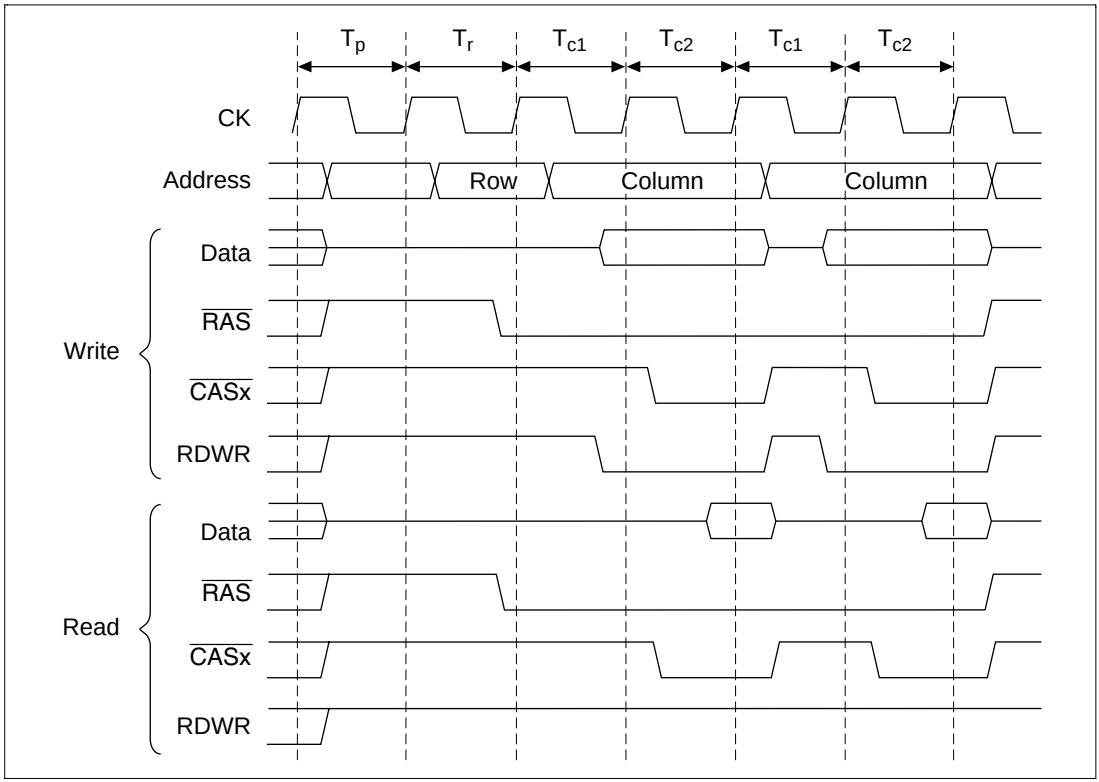


Figure 10.12 DRAM Bus Cycle (High-Speed Page Mode)

RAS Down Mode: There are some instances where even if burst operation is selected, continuous accesses to DRAM will not occur, but another space will be accessed instead part way through the access. In such cases, if the $\overline{RAS}$ signal is maintained at low level during the time the other space is accessed, it is possible to continue burst operation at the time the next DRAM same row address is accessed. This is called RAS down mode.

To use RAS down mode, set both the BE and RASD bits of the DCR to 1.

Figures 10.13 and 10.14 show operation in RAS up and down modes.

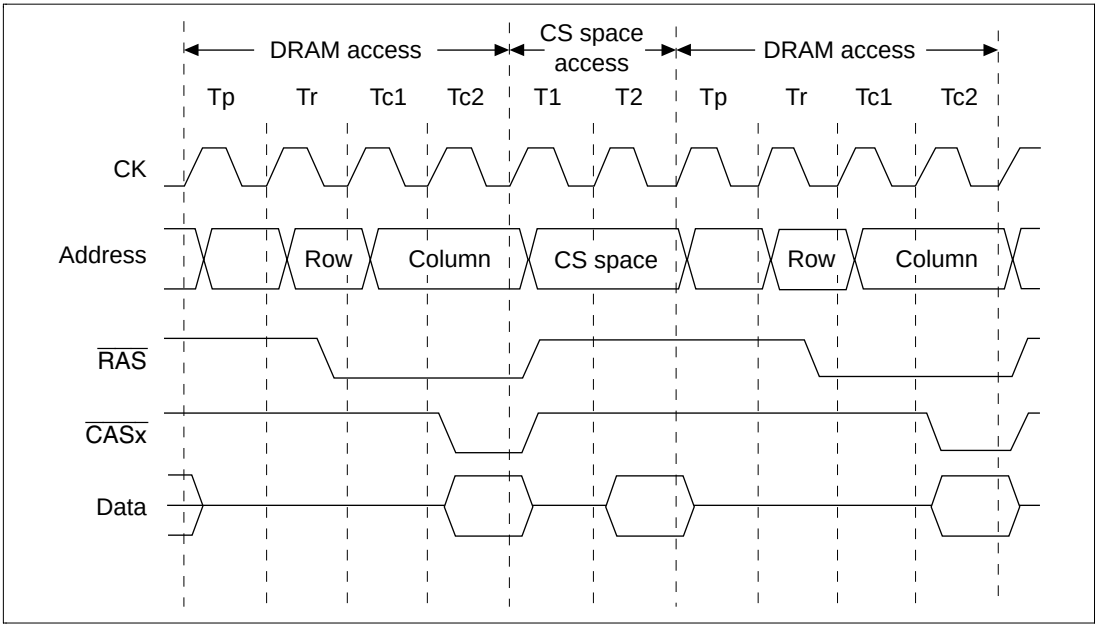


Figure 10.13 DRAM Access Normal Operation (RAS Up Mode)

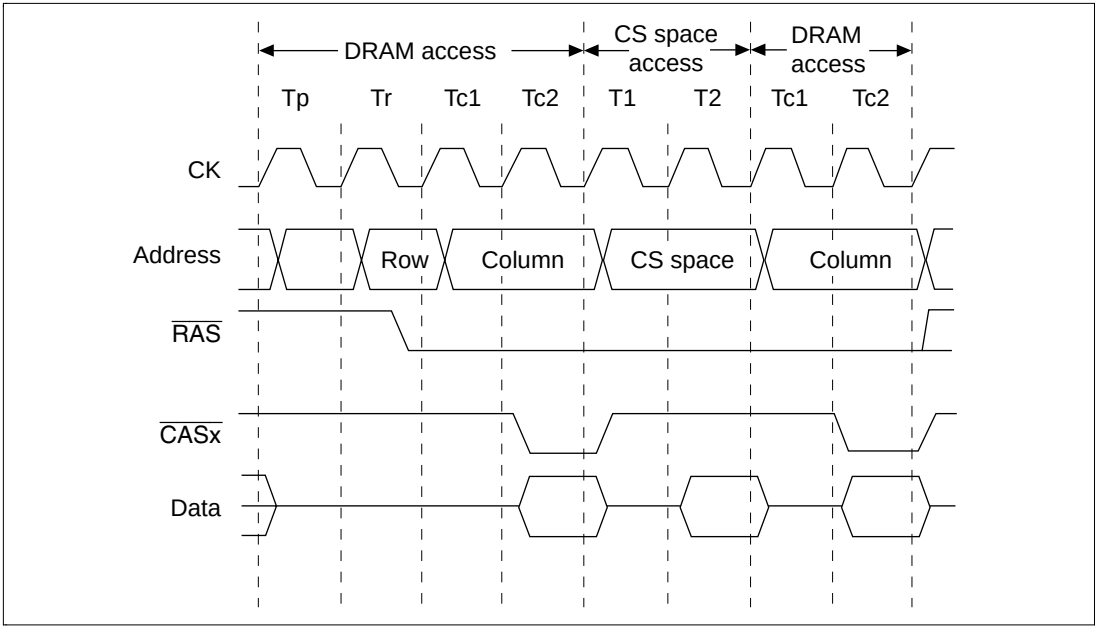


Figure 10.14 RAS Down Mode

10.4.5 Refresh Timing

The bus state controller is equipped with a function to control refreshes of DRAM. CAS-before-RAS (CBR) refresh or self-refresh can be selected by setting the RTCSR's RMD bit.

CAS-before-RAS Refresh: For CBR refreshes, set the RCR's RMD bit to 0 and the RFSH bit to 1. Also write the values in RTCNT and RTCOR necessary to fulfill the refresh interval prescribed for the DRAM being used. When a clock is selected with the CKS2–CKS0 bits of the RSTCR, RTCNT starts counting up from the value at that time. The RTCNT value is constantly compared to the RTCOR value and a CBR refresh is performed when the two match. RTCNT is cleared at that time and the count starts again. Figure 10.15 shows the timing for the CBR refresh operation.

The number of RAS assert cycles in the refresh cycle is set by the TRAS1, TRAS0 bits of the DCR.

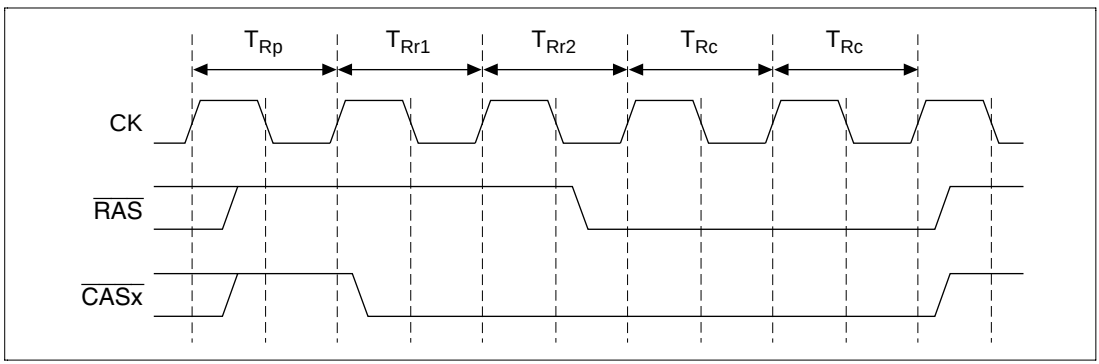


Figure 10.15 CAS-Before-RAS Refresh Timing (TRAS1, TRAS0 = 0, 0)

Self-Refresh: When both the RMD and RFSH bits of the RTCSR are set to 1, the $\overline{CAS}$ signal and $\overline{RAS}$ signal are output and the DRAM enters self-refresh mode, as shown in figure 10.16. Do not access DRAM during self-refreshes, in order to preserve DRAM data. When performing DRAM accesses, first cancel the self-refresh, then access only after doing individual refreshes for all row addresses within the time prescribed for the particular DRAM.

For external bus right requests during self-refreshes, to preserve DRAM data at the time of releasing the bus rights, only $\overline{CASx}$, $\overline{RAS}$, and RDWR are output and the bus rights are released to the external device with the self-refresh maintained. Consequently, do not perform DRAM accesses from external devices at such a time.

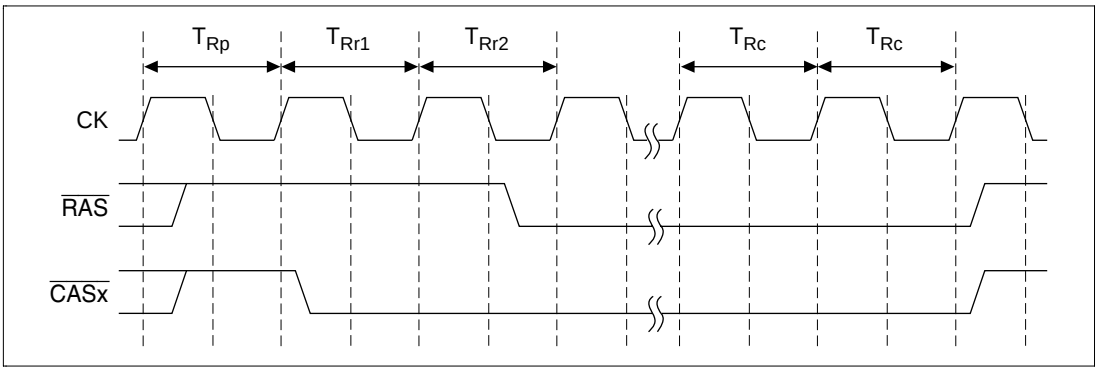


Figure 10.16 Self-Refresh Timing

10.5 Address/Data Multiplex I/O Space Access

When the BCR1 register IOE bit is set to 1, the D15–D0 pins can be used for multiplexed address/data I/O for the CS3 space. Consequently, peripheral LSIs requiring address/data multiplexing can be directly connected to this LSI.

Address/data multiplex I/O space bus width is selected by the A14 bit, and is 8 bit when A14 = 0 and 16 bit when A14 = 1.

10.5.1 Basic Timing

When the IOE bit of the BCR1 is set to 1, CS3 space becomes address/data multiplex I/O space. When this space is accessed, addresses and data are multiplexed. When the A14 address bit is 0, the bus size becomes 8 bit and addresses and data are input and output through the D7–D0 pins. When the A14 address bit is 1, the bus size becomes 16 bit and address output and data I/O occur through the D15–D0 pins. Access for the address/data multiplex I/O space is controlled by the $\overline{AH}$, $\overline{RD}$, and $\overline{WRx}$ signals.

Address/data multiplex I/O space accesses are done after a 3-cycle (fixed) address output, as an ordinary space type access (figure 10.17).

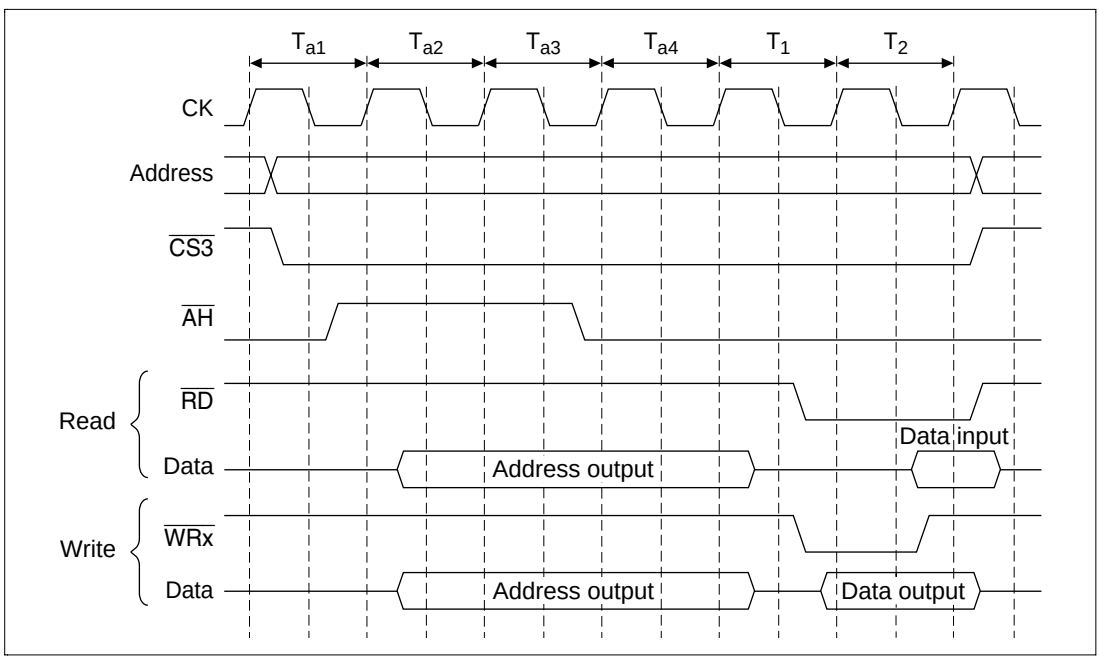


Figure 10.17 Address/Data Multiplex I/O Space Access Timing (No Waits)

10.5.2 Wait State Control

Setting the WCR controls waits during address/data multiplex I/O space accesses. Software wait and external wait insertion timing is the same as during ordinary space accesses. The timing for one software wait + one external wait inserted is shown in figure 10.18.

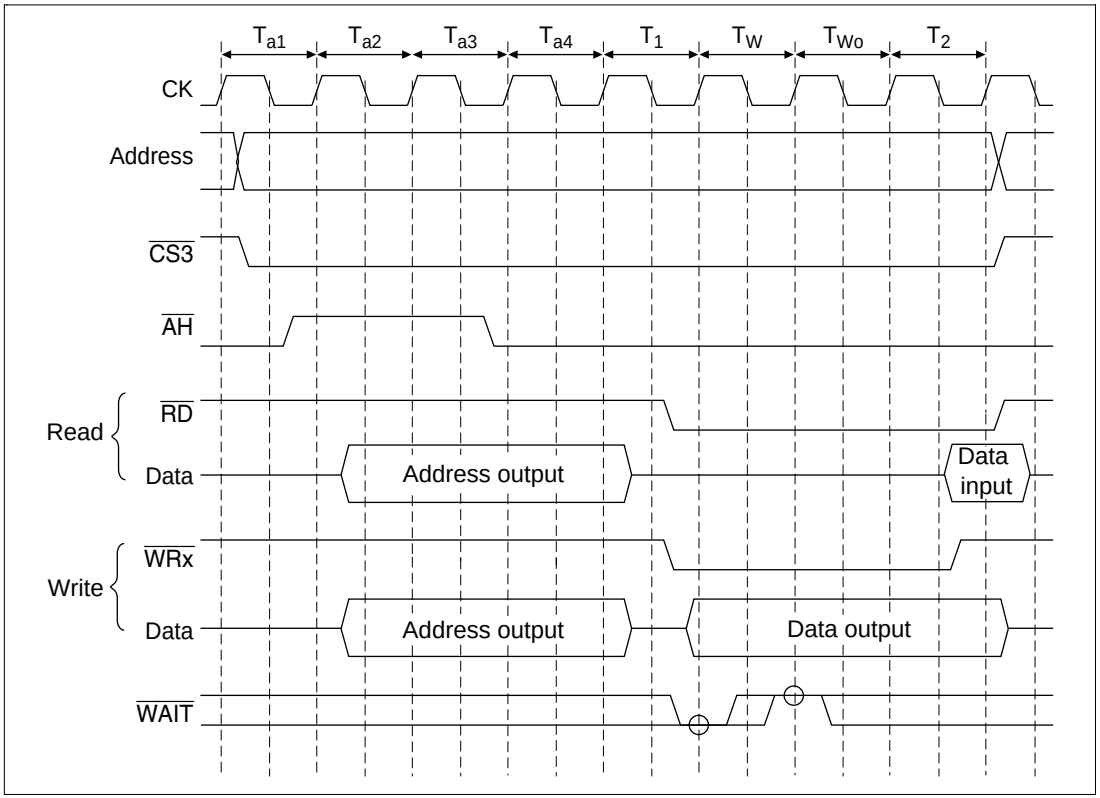


Figure 10.18 Address/Data Multiplex I/O Space Access Wait State Timing (One Software Wait + One External Wait)

10.5.3 CS Assertion Extension

The timing diagram when setting CS assertion extension during address/data multiplex I/O space access is shown in figure 10.19.

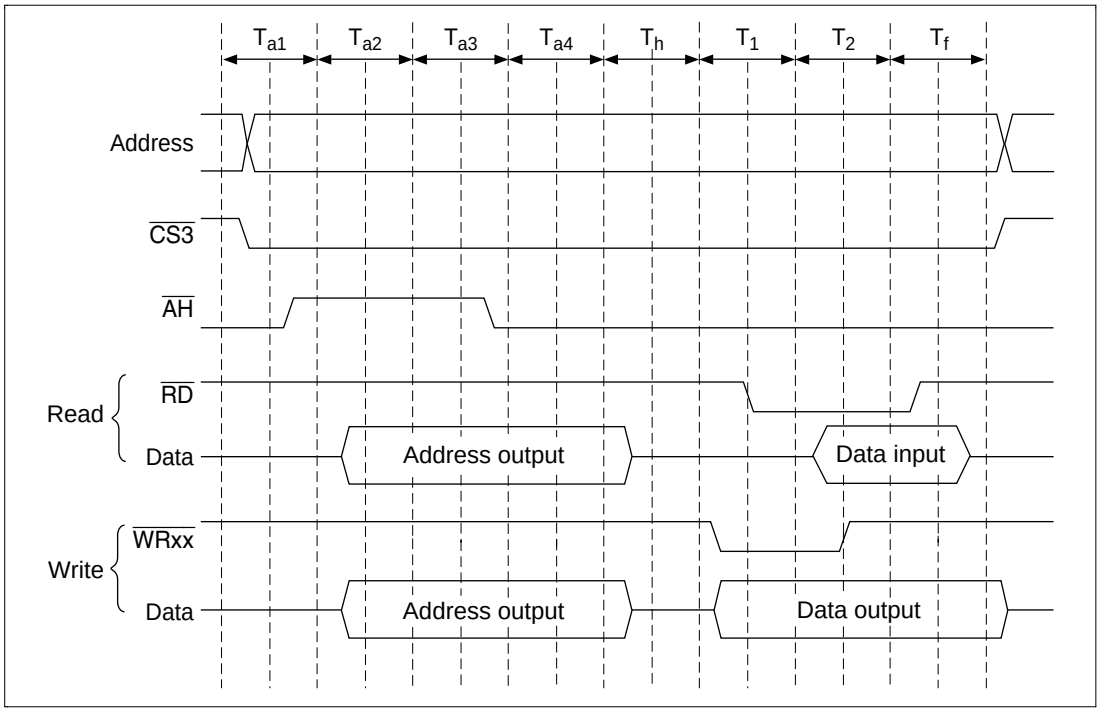


Figure 10.19 Wait Timing in Address/Data Multiplex I/O Space when CS Assertion Extension is Set

10.6 Waits between Access Cycles

When a read from a slow device is completed, data buffers may not go off in time to prevent data conflicts with the next access. If there is a data conflict during memory access, the problem can be solved by inserting a wait in the access cycle.

To enable detection of bus cycle starts, waits can be inserted between access cycles during continuous accesses of the same CS space by negating the $\overline{\text{CSn}}$ signal once.

10.6.1 Prevention of Data Bus Conflicts

For the two cases of write cycles after read cycles, and read cycles for a different area after read cycles, waits are inserted so that the number of idle cycles specified by the IW31–IW00 bits of the

BCR2 and the DIW of the DCR occur. When idle cycles already exist between access cycles, only the number of empty cycles remaining beyond the specified number of idle cycles are inserted.

Figure 10.20 shows an example of idles between cycles. In this example, 1 idle between CSn space cycles has been specified, so when a CSm space write immediately follows a CSn space read cycle, 1 idle cycle is inserted.

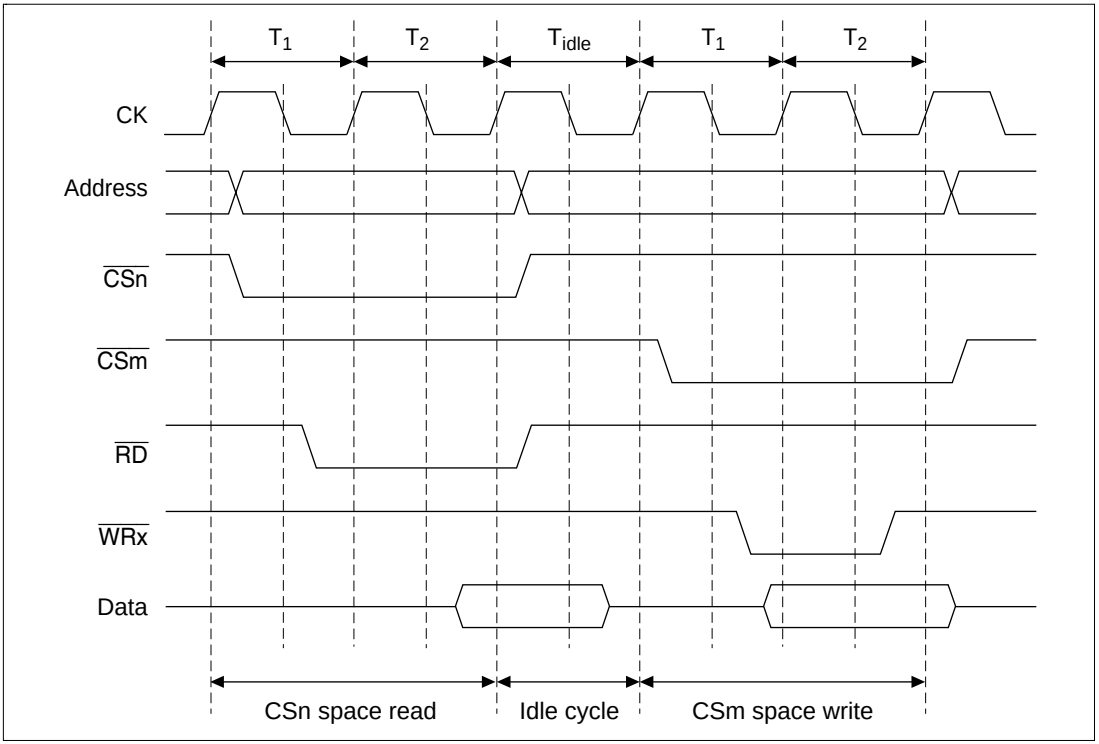


Figure 10.20 Idle Cycle Insertion Example

IW31 and IW30 specify the number of idle cycles required after a CS3 space read either to read other external spaces, or for this LSI, to do write accesses. In the same manner, IW21 and IW20 specify the number of idle cycles after a CS2 space read, IW11 and IW10, the number after a CS1 space read, and IW01 and IW00, the number after a CS0 space read.

DIW specifies the number of idle cycles required, after a DRAM space read either to read other external spaces (CS space), or for this LSI, to do write accesses.

0 to 3 cycles can be specified for CS space, and 0 to 1 cycle for DRAM space.

10.6.2 Simplification of Bus Cycle Start Detection

For consecutive accesses of the same CS space, waits are inserted so that the number of idle cycles designated by the CW3–CW0 bits of the BCR2 occur. However, for write cycles after reads, the number of idle cycles inserted will be the larger of the two values defined by the IW and CW bits. When idle cycles already exist between access cycles, waits are not inserted. Figure 10.21 shows an example. A continuous access idle is specified for CSn space, and CSn space is consecutively write accessed.

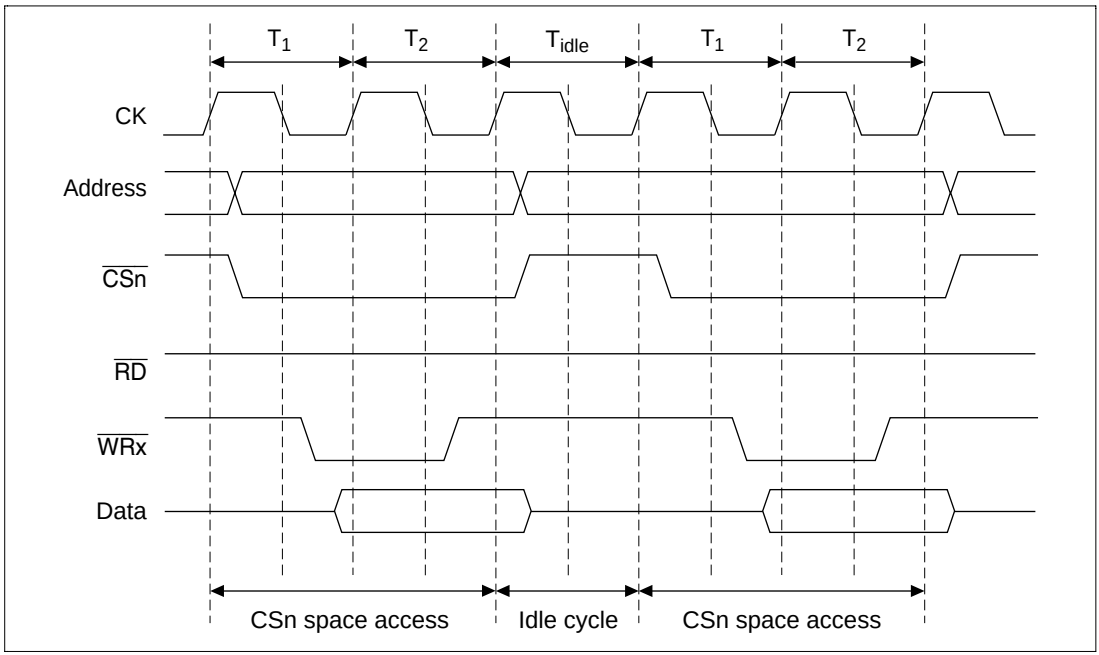


Figure 10.21 Same Space Consecutive Access Idle Cycle Insertion Example

10.7 Bus Arbitration

The SH7040 series has a bus arbitration function that, when a bus release request is received from an external device, releases the bus to that device. It also has two internal bus masters, the CPU and the DMAC, DTC. The priority ranking for determining bus right transfer between these bus masters is:

Bus right request from external device > refresh > DTC > DMAC > CPU

However, during a read or write in DMAC dual address mode, a burst transfer, or indirect address transfer mode operation, the DMAC continues operating even if a DTC request is received.

Through port register settings, $\overline{\text{IRQOUT}}$ is asserted to indicate that a CAS-before-RAS refresh request for DRAM has been generated during release of bus rights to an external device. Use this

to cause the external device to negate the $\overline{\text{BREQ}}$ and return the bus rights to the SH7040 Series. Please note that if the external device does not return the bus rights within the time prescribed for the DRAM refresh interval, this LSI will not be able to perform the refresh operation and the DRAM contents cannot be guaranteed.

Figure 10.22 shows the bus right release procedure.

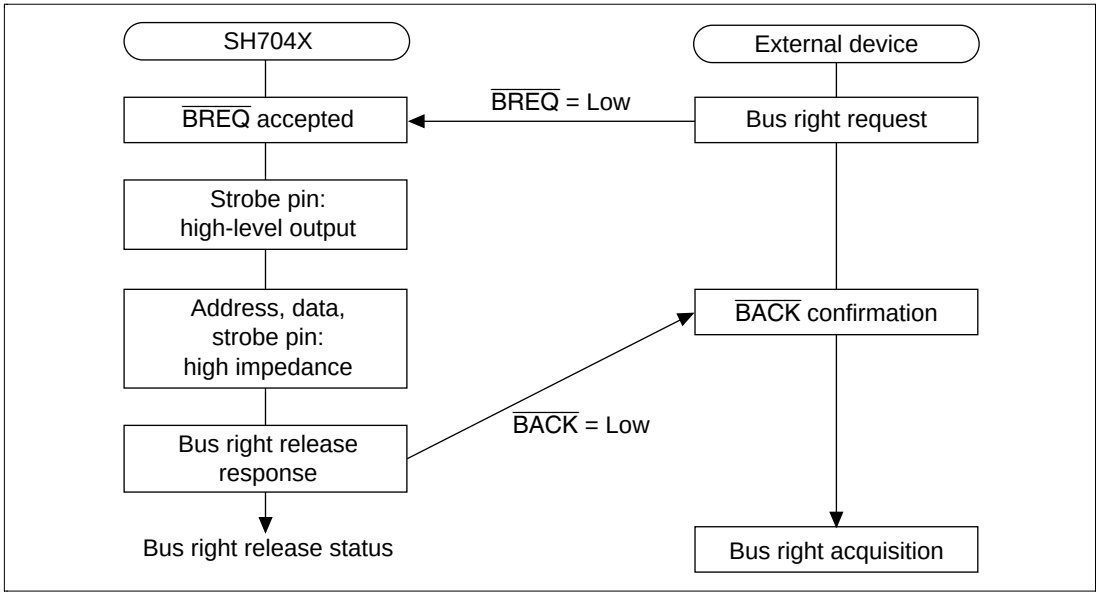


Figure 10.22 Bus Right Release Procedure

10.8 Memory Connection Examples

Figures 10.23–10.31 show examples of the memory connections.

As A21–A18 become input ports in power-on reset, they should be handled (e.g. pulled down) as necessary.

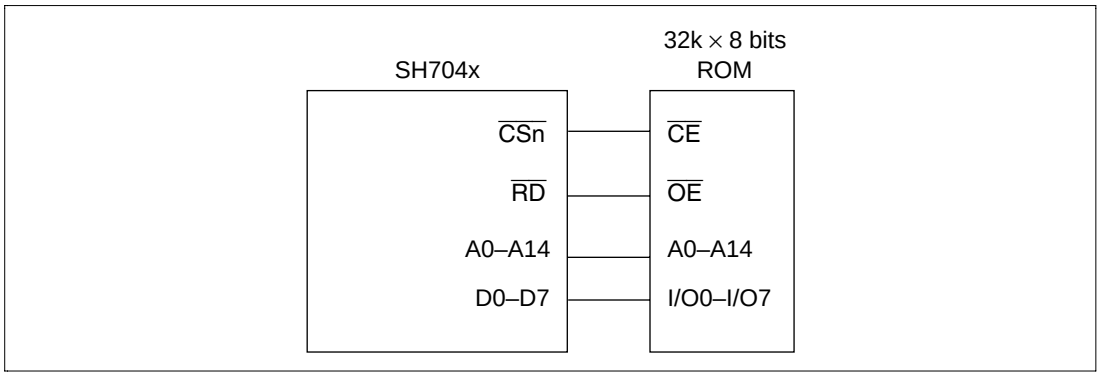


Figure 10.23 8-Bit Data Bus Width ROM Connection

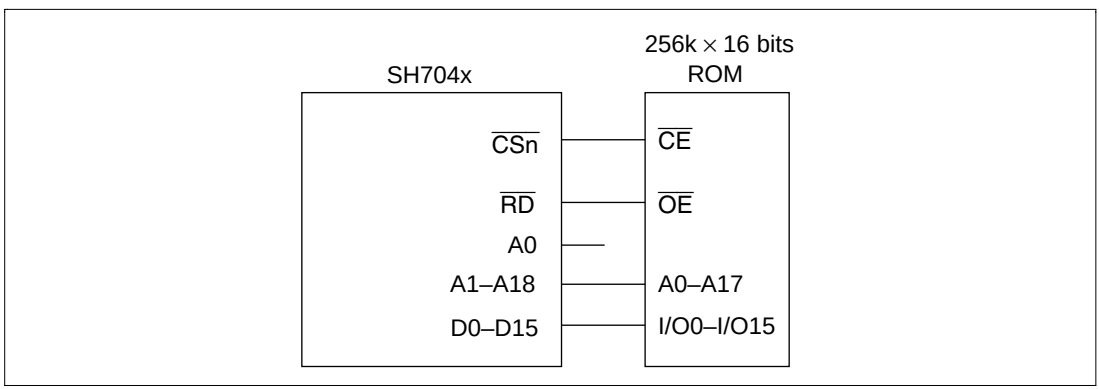


Figure 10.24 16-Bit Data Bus Width ROM Connection

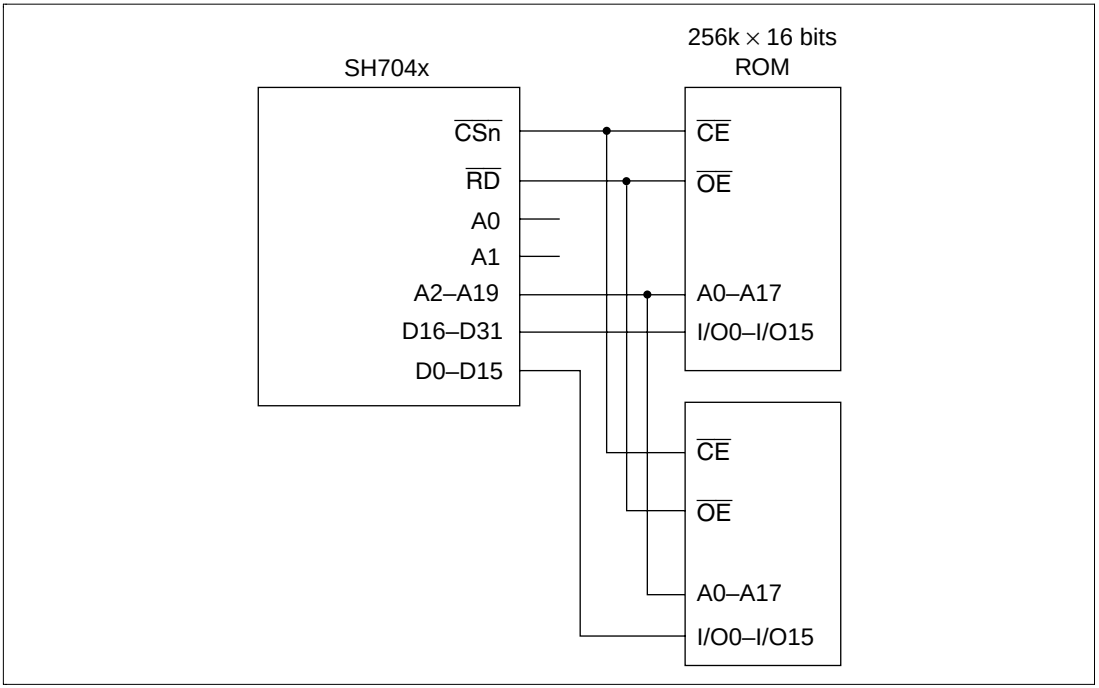


Figure 10.25 32-Bit Data Bus Width ROM Connection

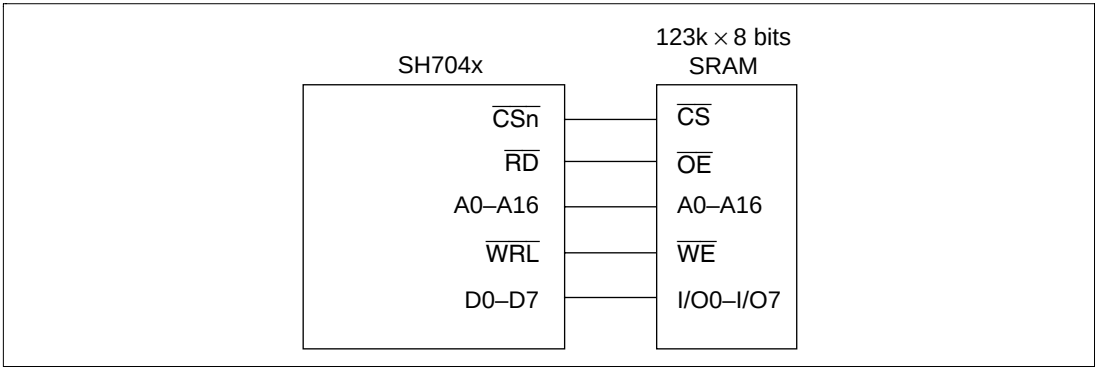


Figure 10.26 8-Bit Data Bus Width SRAM Connection

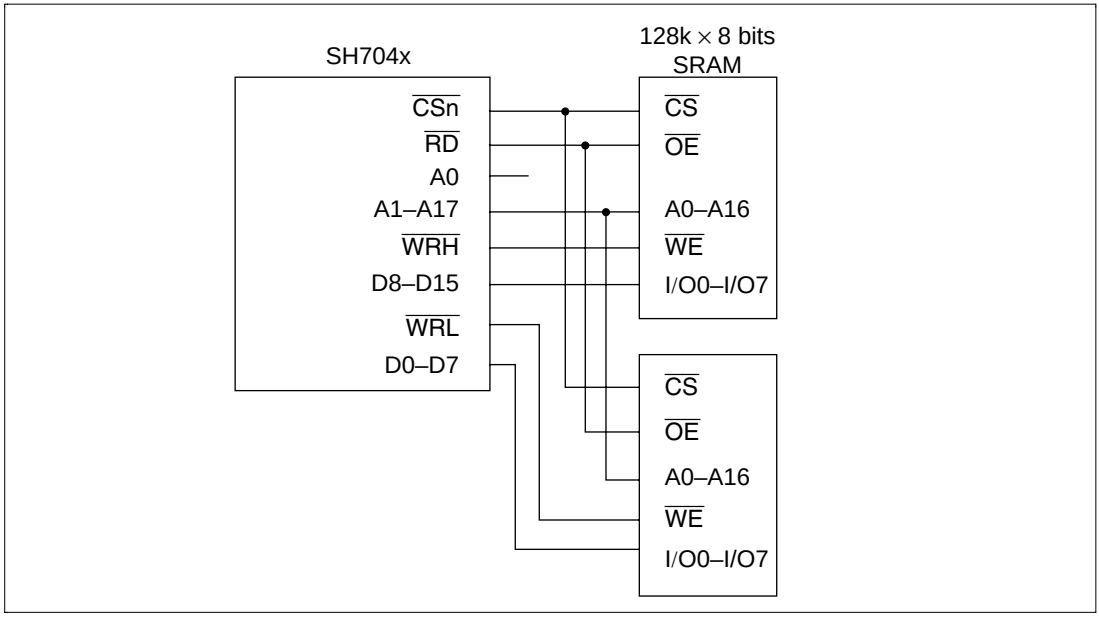


Figure 10.27 16-Bit Data Bus Width SRAM Connection

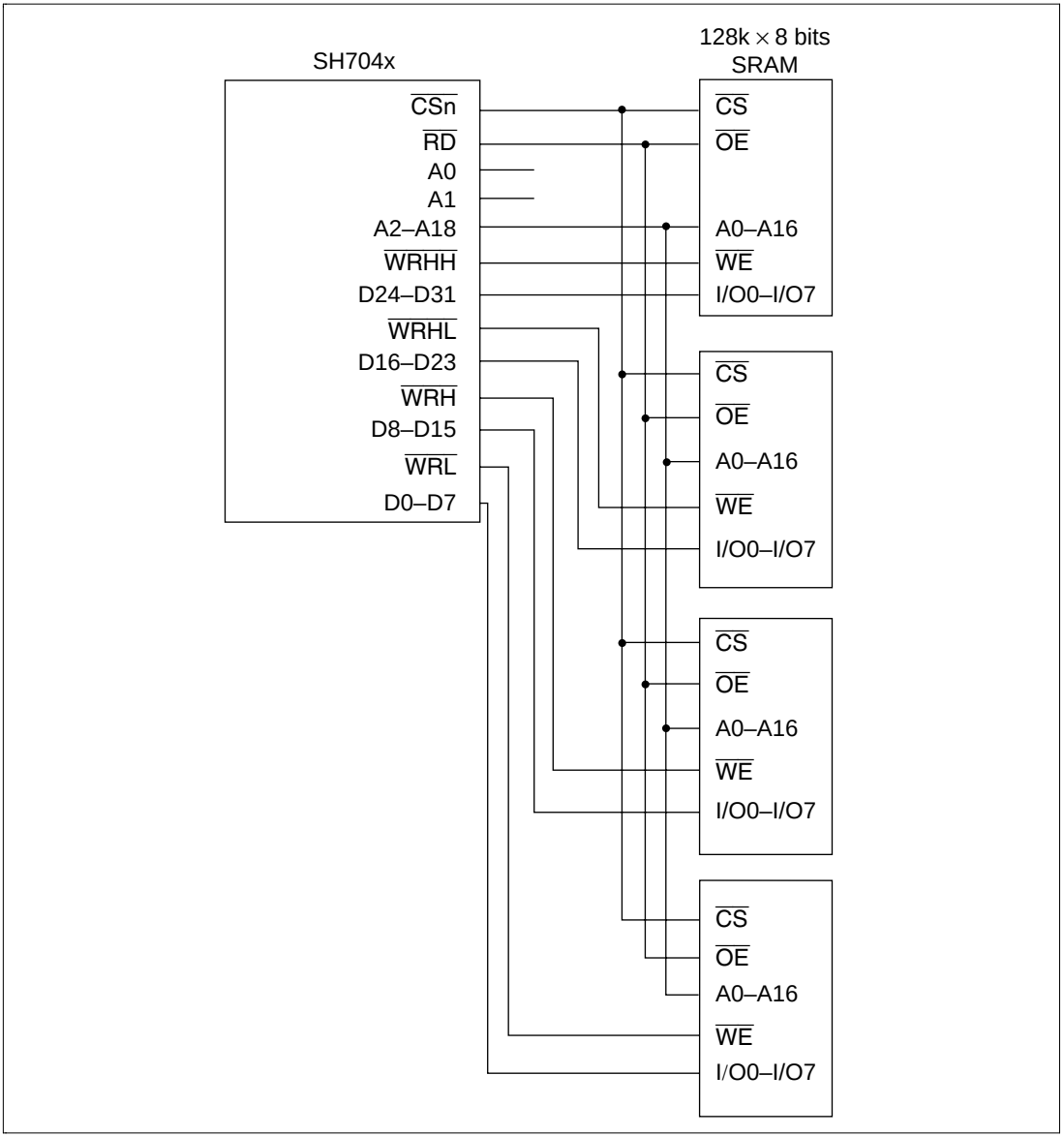


Figure 10.28 32-Bit Data Bus Width SRAM Connection

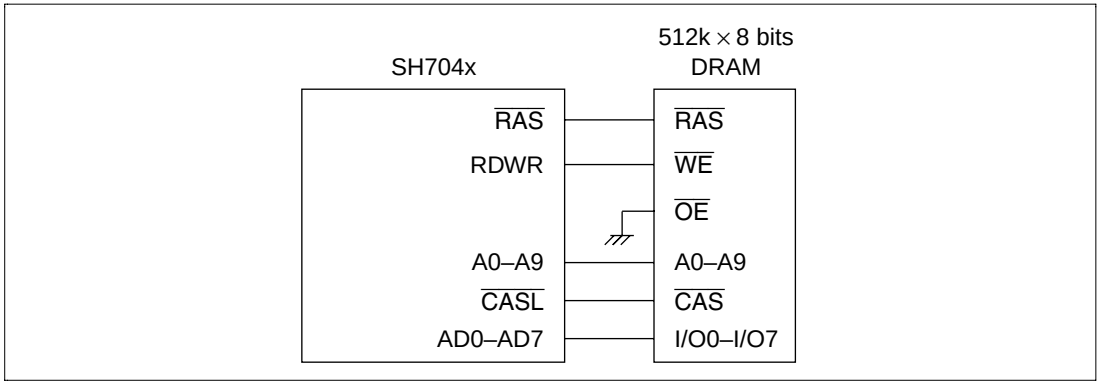


Figure 10.29 8-Bit Data Bus Width DRAM Connection

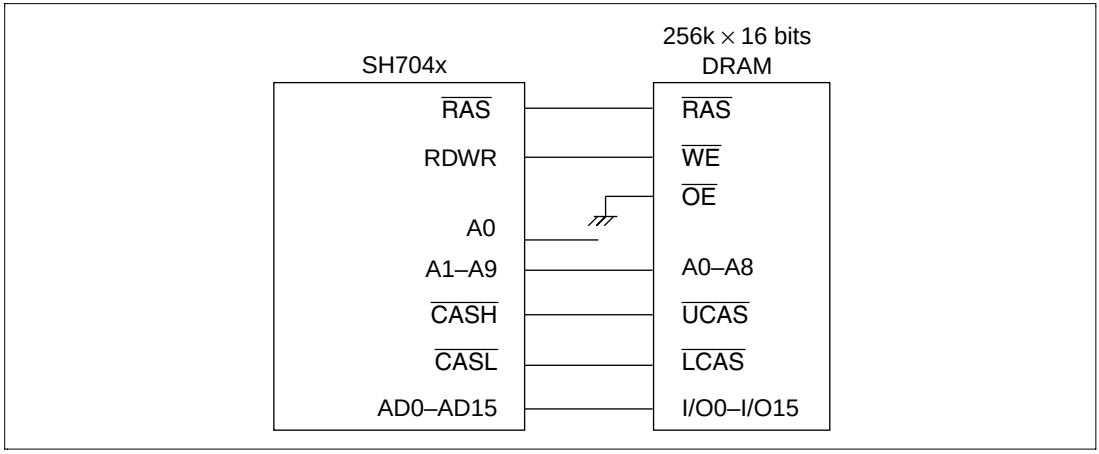


Figure 10.30 16-Bit Data Bus Width DRAM Connection

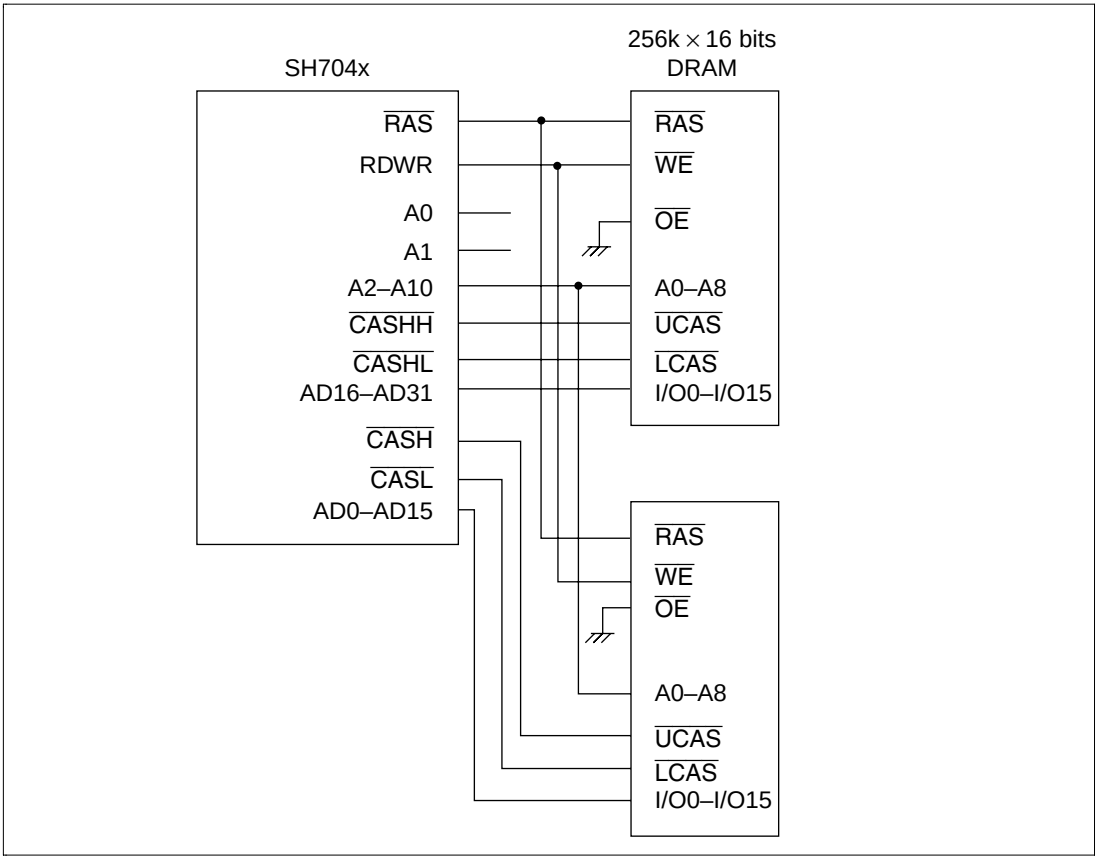


Figure 10.31 32-Bit Data Bus Width DRAM Connection

10.9 On-Chip Peripheral I/O Register Access

On-chip peripheral I/O registers are accessed from the bus state controller, as shown in table 10.6.

Table 10.6 On-Chip Peripheral I/O Register Access

On-chip Peripheral Module	SCI	MTU, POE	INTC	PFC, PORT	CMT	A/D*	UBC	WDT	DMAC	DTC	CACHE
Connected bus width	8bit	16bit	16bit	16bit	16bit	16bit	16bit	16bit	16bit	16bit	16bit
Access cycle	2cyc	2cyc	2cyc	2cyc	2cyc	2cyc	3cyc	3cyc	3cyc	3cyc	3cyc

Note: * A/D of A mask products are accessed in 8-bit width, 3 cyc.

Cycles in which Bus is not Released

(a) One bus cycle:

The bus is never released during a single bus cycle. For example, in the case of a longword read (or write) in 8-bit normal space, the four memory accesses to the 8-bit normal space constitute a single bus cycle, and the bus is never released during this period. Assuming that one memory access requires two states, the bus is not released during an 8-state period.

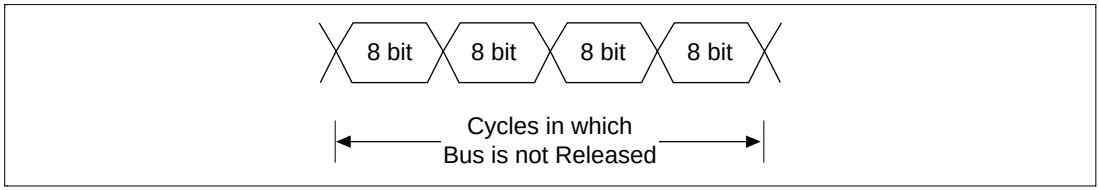


Figure 10.32 One Bus Cycle

10.10 CPU Operation when Program is in External Memory

In the SH7040 Series, two words (equivalent to two instructions) are normally fetched in a single instruction fetch. This is also true when the program is located in external memory, irrespective of whether the external memory bus width is 8 or 16 bits.

If the program counter value immediately after the program branches is an odd-word (2n + 1) address, or if the program counter value immediately before the program branches is an even-word (2n) address, the CPU will always fetch 32 bits (equivalent to two instructions) that include the respective word instruction.

Section 11 Direct Memory Access Controller (DMAC)

11.1 Overview

The SH7040 Series includes an on-chip four-channel direct memory access controller (DMAC). The DMAC can be used in place of the CPU to perform high-speed data transfers among external devices equipped with DACK (transfer request acknowledge signal), external memories, memory-mapped external devices, and on-chip peripheral modules (except for the DMAC, DTC, BSC, and UBC). Using the DMAC reduces the burden on the CPU and increases operating efficiency of the LSI as a whole.

11.1.1 Features

The DMAC has the following features:

- Four channels
- Four Gbytes of address space in the architecture
- Byte, word, or longword selectable data transfer unit
- 16 Mbytes (16,777,216 transfers, maximum)
- Single or dual address mode. Dual address mode can be direct or indirect address transfer.
  - — Single address mode: Either the transfer source or transfer destination (peripheral device) is accessed by a DACK signal while the other is accessed by address. One transfer unit of data is transferred in each bus cycle.
  - — Dual address mode: Both the transfer source and transfer destination are accessed by address. Dual address mode can be direct or indirect address transfer.
    - Direct access: Values set in a DMAC internal register indicate the accessed address for both the transfer source and transfer destination. Two bus cycles are required for one data transfer.
    - Indirect access: The value stored at the location pointed to by the address set in the DMAC internal transfer source register is used as the address. Operation is otherwise the same as direct access. This function can only be set for channel 3. Four bus cycles are required for one data transfer.
- Channel function: Transfer modes that can be set are different for each channel. (Dual address mode indirect access can only be set for channel 1. Only direct access is possible for the other channels.)
  - — Channel 0: Single or dual address mode. External requests are accepted.
  - — Channel 1: Single or dual address mode. External requests are accepted.
  - — Channel 2: Dual address mode only. Source address reload function operates every fourth transfer.

— Channel 3: Dual address mode only. Direct address transfer mode and indirect address transfer mode selectable.

- Reload function: Enables automatic reloading of the value set in the first source address register every fourth DMA transfer. This function can be executed on channel 2 only.
- Transfer requests: There are three DMAC transfer activation requests, as indicated below.
  - — External request: From two DREQ pins. DREQ can be detected either by falling edge or by low level. External requests can only be received on channels 0 or 1.
  - — Requests from on-chip peripheral modules: Transfer requests from on-chip modules such as SCI or A/D. These can be received by all channels.
  - — Auto-request: The transfer request is generated automatically within the DMAC.
- Selectable bus modes: Cycle-steal mode or burst mode
- Two types of DMAC channel priority ranking:
  - — Fixed priority mode: Always fixed
  - — Round robin mode: Sets the lowest priority level for the channel that received the execution request last
- CPU can be interrupted when the specified number of data transfers are complete.

11.1.2 Block Diagram

Figure 11.1 is a block diagram of the DMAC.

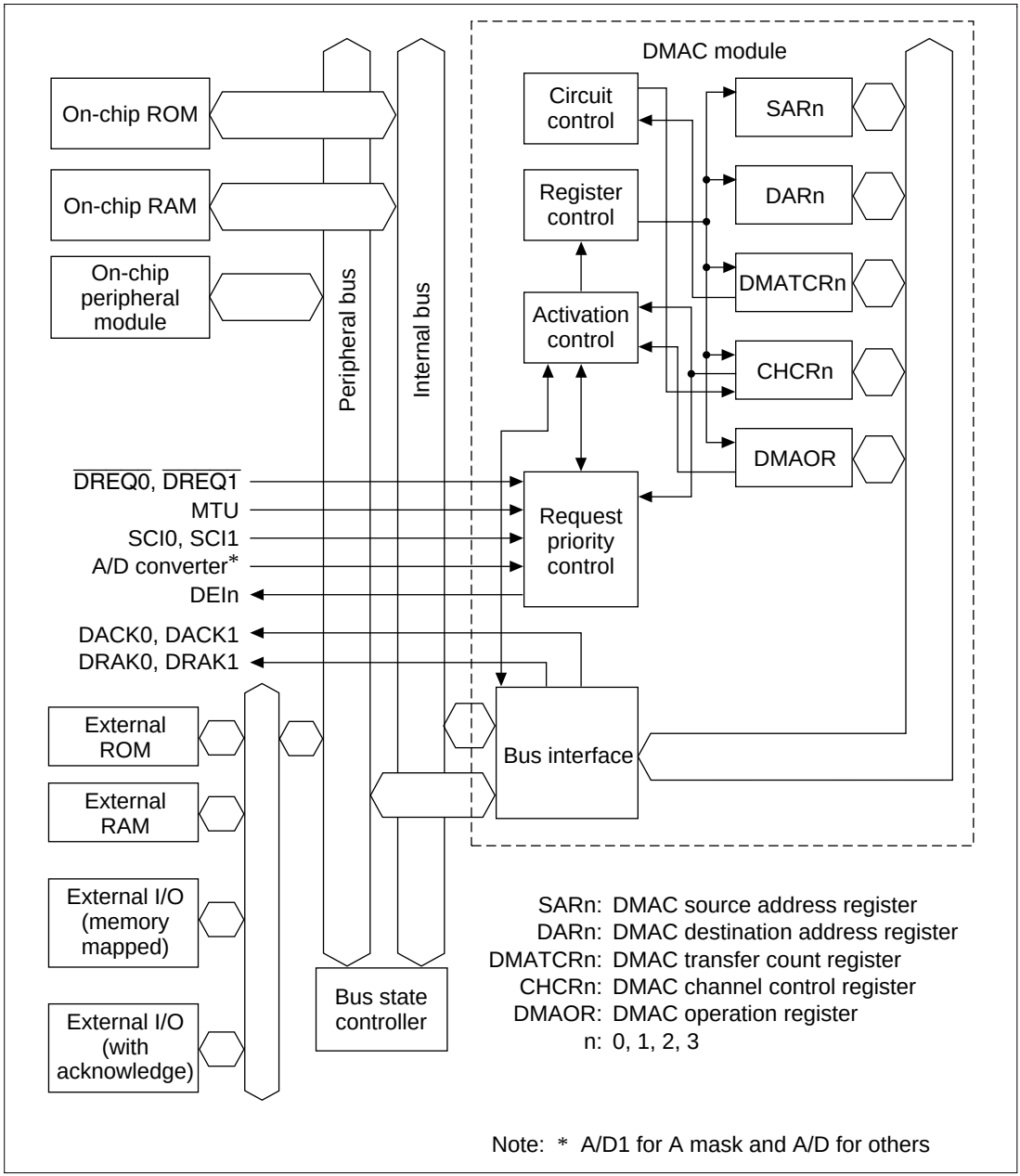


Figure 11.1 DMAC Block Diagram

11.1.3 Pin Configuration

Table 11.1 shows the DMAC pins.

Table 11.1 DMAC Pin Configuration

Channel	Name	Symbol	I/O	Function
0	DMA transfer request	$\overline{\text{DREQ0}}$	I	DMA transfer request input from external device to channel 0
	DMA transfer request acknowledge	DACK0	O	DMA transfer strobe output from channel 0 to external device
	$\overline{\text{DREQ0}}$ acceptance confirmation	DRAK0	O	Sampling receive acknowledge output for DMA transfer request input from external source
1	DMA transfer request	$\overline{\text{DREQ1}}$	I	DMA transfer request input from external device to channel 1
	DMA transfer request acknowledge	DACK1	O	DMA transfer strobe output from channel 1 to external device
	$\overline{\text{DREQ1}}$ acceptance confirmation	DRAK1	O	Sampling receive acknowledge output for DMA transfer request input from external source

11.1.4 Register Configuration

Table 11.2 summarizes the DMAC registers. DMAC has a total of 17 registers. Each channel has four control registers. One other control register is shared by all channels

Table 11.2 DMAC Registers

Channel	Name	Abbreviation	R/W	Initial Value	Address	Register Size	Access Size
0	DMA source address register 0	SAR0	R/W	Undefined	H'FFFF86C0	32 bit	16, 32*2
	DMA destination address register 0	DAR0	R/W	Undefined	H'FFFF86C4	32 bit	16, 32*2
	DMA transfer count register 0	DMATCR0	R/W	Undefined	H'FFFF86C8	32 bit	16, 32*3
	DMA channel control register 0	CHCR0	R/W*1	H'00000000	H'FFFF86CC	32 bit	16, 32*2
1	DMA source address register 1	SAR1	R/W	Undefined	H'FFFF86D0	32 bit	16, 32*2
	DMA destination address register 1	DAR1	R/W	Undefined	H'FFFF86D4	32 bit	16, 32*2
	DMA transfer count register 1	DMATCR1	R/W	Undefined	H'FFFF86D8	32 bit	16, 32*3
	DMA channel control register 1	CHCR1	R/W*1	H'00000000	H'FFFF86DC	32 bit	16, 32*2
2	DMA source address register 2	SAR2	R/W	Undefined	H'FFFF86E0	32 bit	16, 32*2
	DMA destination address register 2	DAR2	R/W	Undefined	H'FFFF86E4	32 bit	16, 32*2

Table 11.2 DMAC Registers (cont)

Channel	Name	Abbreviation	R/W	Initial Value	Address	Register Size	Access Size
2 (cont)	DMA transfer count register 2	DMATCR2	R/W	Undefined	H'FFFF86E8	32 bit	16, 32*3
	DMA channel control register 2	CHCR2	R/W*1	H'00000000	H'FFFF86EC	32 bit	16, 32*2
3	DMA source address register 3	SAR3	R/W	Undefined	H'FFFF86F0	32 bit	16, 32*2
	DMA destination address register 3	DAR3	R/W	Undefined	H'FFFF86F4	32 bit	16, 32*2
	DMA transfer count register 3	DMATCR3	R/W	Undefined	H'FFFF86F8	32 bit	16, 32*3
	DMA channel control register 3	CHCR3	R/W*1	H'00000000	H'FFFF86FC	32 bit	16, 32*2
Shared	DMA operation register	DMAOR	R/W*1	H'0000	H'FFFF86B0	16 bit	16, 32*4

Notes: Do not attempt to access an empty address. If an access is attemped, the system operation is not guarenteed.

*1 Write 0 after reading 1 in bit 1 of CHCR0–CHCR3 and in bits 1 and 2 of the DMAOR to clear flags. No other writes are allowed.

*2 For 16-bit access of SAR0–SAR3, DAR0–DAR3, and CHCR0–CHCR3, the 16-bit value on the side not accessed is held.

*3 DMATCR has a 24-bit configuration: bits 0–23. Writing to the upper 8 bits (bits 24–31) is invalid, and these bits always read 0.

*4 Do not make 32-bit access for DMAOR.

11.2 Register Descriptions

11.2.1 DMA Source Address Registers 0–3 (SAR0–SAR3)

DMA source address registers 0–3 (SAR0–SAR3) are 32-bit read/write registers that specify the source address of a DMA transfer. These registers have a count function, and during a DMA transfer, they indicate the next source address. In single-address mode, SAR values are ignored when a device with DACK has been specified as the transfer source.

Specify a 16-bit or 32-bit boundary address when doing 16-bit or 32-bit data transfers. Operation cannot be guaranteed on any other addresses.

The initial value after power-on resets or in software standby mode is undefined. These registers are not initialized with manual reset.

Bit:	31	30	29	28	27	26	25	24
Initial value:	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	23	22	21	…	…	2	1	0
				…	…			
Initial value:	—	—	—	…	…	—	—	—
R/W:	R/W	R/W	R/W	…	…	R/W	R/W	R/W

11.2.2 DMA Destination Address Registers 0–3 (DAR0–DAR3)

DMA destination address registers 0–3 (DAR0–DAR3) are 32-bit read/write registers that specify the destination address of a DMA transfer. These registers have a count function, and during a DMA transfer, they indicate the next destination address. In single-address mode, DAR values are ignored when a device with DACK has been specified as the transfer destination.

Specify a 16-bit or 32-bit boundary address when doing 16-bit or 32-bit data transfers. Operation cannot be guaranteed on any other address. The initial value after power-on resets or in software standby mode, is undefined. These registers are not initialized with manual reset.

Bit:	31	30	29	28	27	26	25	24
Initial value:	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	23	22	21	…	…	2	1	0
				…	…			
Initial value:	—	—	—	…	…	—	—	—
R/W:	R/W	R/W	R/W	…	…	R/W	R/W	R/W

11.2.3 DMA Transfer Count Registers 0–3 (DMATCR0–DMATCR3)

DMA transfer count registers 0–3 (DMATCR0–DMATCR3) are 24-bit read/write registers that specify the transfer count for the channel (byte count, word count, or longword count). Specifying a H'000001 gives a transfer count of 1, while H'000000 gives the maximum setting, 16,777,216 transfers. The data for the upper 8 bits of a DMATCR is 0 when read. Always write 0. The initial value after power-on resets or in software standby mode is undefined. These registers are not initialized with manual reset.

Always write 0 to the upper 8 bits of a DMATCR.

Bit:	31	30	29	28	27	26	25	24
	—	—	—	—	—	—	—	—
Initial value:	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R

Bit:	23	22	21	20	19	18	17	16
Initial value:	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8
Initial value:	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
Initial value:	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

11.2.4 DMA Channel Control Registers 0–3 (CHCR0–CHCR3)

DMA channel control registers 0–3 (CHCR0–CHCR3) is a 32-bit read/write register where the operation and transmission of each channel is designated. They are initialized by a power-on reset and in software standby mode. There is no initializing with manual reset.

Bit:	31	30	29	28	27	26	25	24
	—	—	—	—	—	—	—	—
Initial value:	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R

Bit:	23	22	21	20	19	18	17	16
	—	—	—	DI*2	RO*2	RL*2	AM*2	AL*2
Initial value:	—	—	—	0	0	0	0	0
	R	R	R	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)

Bit:	15	14	13	12	11	10	9	8
	DM1	DM0	SM1	SM0	RS3	RS2	RS1	RS0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	—	DS*2	TM	TS1	TS0	IE	TE	DE
Initial value:	—	0	0	0	0	0	0	0
R/W:	R	(R/W)	R/W	R/W	R/W	R/W	R/(W) *1	R/W

Notes: *1 TE bit: Allows only 0 write after reading 1.
*2 The DI, RO, RL, AM, AL, or DS bit may be absent, depending on the channel.

- Bits 31–21—Reserved bits: Data are 0 when read. The write value always be 0.
- Bit 20—Direct/Indirect (DI): Specifies either direct address mode operation or indirect address mode operation for channel 3 source address. This bit is valid only in CHCR3. It always reads 0 for CHCR0–CHCR2, and cannot be modified.

Bit 20: DI	Description
0	Direct access mode operation for channel 3 (initial value)
1	Indirect access mode operation for channel 3

- Bit 19—Source Address Reload (RO): Selects whether to reload the source address initial value during channel 2 transfer. This bit is valid only for channel 2. It always reads 0 for CHCR0, CHCR1, and CHCR3, and cannot be modified.

Bit 19: RO	Description
0	Does not reload source address (initial value)
1	Reloads source address

- Bit 18—Request Check Level (RL): Selects whether to output DRAK notifying external device of $\overline{\text{DREQ}}$ received, with active high or active low. This bit is valid only for CHCR0 and CHCR1. It always reads 0 for CHCR2 and CHCR3, and cannot be modified.

Bit 18: RL	Description
0	Output DRAK with active high (initial value)
1	Output DRAK with active low

- Bit 17—Acknowledge Mode (AM): In dual address mode, selects whether to output DACK in the data write cycle or data read cycle. In single address mode, DACK is always output irrespective of the setting of this bit. This bit is valid only for CHCR0 and CHCR1. It always reads as 0 for CHCR2 and CHCR3, and cannot be modified.

Bit 17: AM	Description
0	Outputs DACK during read cycle (initial value)
1	Outputs DACK during write cycle

- Bit 16—Acknowledge Level (AL): Specifies whether to set DACK (acknowledge) signal output to active high or active low. This bit is valid only with CHCR0 and CHCR1. It always reads as 0 for CHCR2 and CHCR3, and cannot be modified.

Bit 16: AL	Description
0	Active high output (initial value)
1	Active low output

- Bits 15 and 14—Destination Address Mode 1, 0 (DM1 and DM0): These bits specify increment/decrement of the DMA transfer destination address. These bit specifications are ignored when transferring data from an external device to address space in single address mode.

Bit 15: DM1	Bit 14: DM0	Description
0	0	Destination address fixed (initial value)
0	1	Destination address incremented (+1 during 8-bit transfer, +2 during 16-bit transfer, +4 during 32-bit transfer)
1	0	Destination address decremented (–1 during 8-bit transfer, –2 during 16-bit transfer, –4 during 32-bit transfer)
1	1	Setting prohibited

- Bits 13 and 12—Source Address Mode 1, 0 (SM1 and SM0): These bits specify increment/decrement of the DMA transfer source address. These bit specifications are ignored when transferring data from an external device to address space in single address mode.

Bit 13: SM1	Bit 12: SM0	Description
0	0	Source address fixed (initial value)
0	1	Source address incremented (+1 during 8-bit transfer, +2 during 16-bit transfer, +4 during 32-bit transfer)
1	0	Source address decremented (–1 during 8-bit transfer, –2 during 16-bit transfer, –4 during 32-bit transfer)
1	1	Setting prohibited

When the transfer source is specified at an indirect address, specify in source address register 3 (SAR3) the actual storage address of the data you want to transfer as the data storage address (indirect address).

During indirect address mode, SAR3 obeys the SM1/SM0 setting for increment/decrement. In this case, SAR3's increment/decrement is fixed at +4/–4 or 0, irrespective of the transfer data size specified by TS1 and TS0.

- Bits 11–8—Resource Select 3–0 (RS3–RS0): These bits specify the transfer request source.

Bit 11: RS3	Bit 10: RS2	Bit 9: RS1	Bit 8: RS0	Description
0	0	0	0	External request, dual address mode (initial value)
0	0	0	1	Prohibited
0	0	1	0	External request, single address mode. External address space → external device.
0	0	1	1	External request, single address mode. External device → external address space.
0	1	0	0	Auto-request
0	1	0	1	Prohibited
0	1	1	0	MTU TGI0A
0	1	1	1	MTU TGI1A
1	0	0	0	MTU TGI2A
1	0	0	1	MTU TGI3A
1	0	1	0	MTU TGI4A
1	0	1	1	A/D ADI*
1	1	0	0	SCI0 TXI0
1	1	0	1	SCI0 RXI0
1	1	1	0	SCI1 TXI1
1	1	1	1	SCI1 RXI1

Notes: External request designations are valid only for channels 0 and 1. No transfer request sources can be set for channels 2 or 3.

* ADI1 for A mask.

- Bit 7—Reserved bits: Data is 0 when read. The write value always be 0.
- Bit 6—$\overline{\text{DREQ}}$ Select (DS): Sets the sampling method for the $\overline{\text{DREQ}}$ pin in external request mode to either low-level detection or falling-edge detection. This bit is valid only with CHCR0 and CHCR1. For CHCR2 and CHCR3, this bit always reads as 0 and cannot be modified.

  Even with channels 0 and 1, when specifying an on-chip peripheral module or auto-request as the transfer request source, this bit setting is ignored. The sampling method is fixed at falling-edge detection in cases other than auto-request.

Bit 6: DS	Description
0	Low-level detection (initial value)
1	Falling-edge detection

- Bit 5—Transfer Mode (TM): Specifies the bus mode for data transfer.

Bit 5: TM	Description
0	Cycle steal mode (initial value)
1	Burst mode

- Bits 4 and 3—Transfer Size 1, 0 (TS1, TS0): Specifies size of data for transfer.

Bit 4: TS1	Bit 3: TS0	Description
0	0	Specifies byte size (8 bits) (initial value)
0	1	Specifies word size (16 bits)
1	0	Specifies longword size (32 bits)
1	1	Prohibited

- Bit 2—Interrupt Enable (IE): When this bit is set to 1, interrupt requests are generated after the number of data transfers specified in the DMATCR (when TE = 1).

Bit 2: IE	Description
0	Interrupt request not generated after DMATCR-specified transfer count (initial value)
1	Interrupt request enabled on completion of DMATCR specified number of transfers

- Bit 1—Transfer End Flag (TE): This bit is set to 1 after the number of data transfers specified by the DMATCR. At this time, if the IE bit is set to 1, an interrupt request is generated.

  If data transfer ends before TE is set to 1 (for example, due to an NMI or address error, or clearing of the DE bit or DME bit of the DMAOR) the TE is not set to 1. With this bit set to 1, data transfer is disabled even if the DE bit is set to 1.

Bit 1: TE	Description
0	DMATCR-specified transfer count not ended (initial value) Clear condition: 0 write after TE = 1 read, Power-on reset, standby mode
1	DMATCR specified number of transfers completed

- Bit 0—DMAC Enable (DE): DE enables operation in the corresponding channel.

Bit 0: DE	Description
0	Operation of the corresponding channel disabled (initial value)
1	Operation of the corresponding channel enabled

Transfer mode is entered if this bit is set to 1 when auto-request is specified (RS3–RS0 settings). With an external request or on-chip module request, when a transfer request occurs after this bit is set to 1, transfer is enabled. If this bit is cleared during a data transfer, transfer is suspended.

If the DE bit has been set, but TE = 1, then if the DME bit of the DMAOR is 0, and the NMI or AE bit of the DMAOR is 1, transfer enable mode is not entered.

11.2.5 DMAC Operation Register (DMAOR)

The DMAOR is a 16-bit read/write register that specifies the transfer mode of the DMAC

Register values are initialized to 0 during power-on reset or in software standby mode. Manual reset does not initialize DMAOR.

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	PR1	PR0
Initial value:	—	—	—	—	—	—	0	0
R/W:	R	R	R	R	R	R	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	AE	NMIF	DME
Initial value:	—	—	—	—	—	—	0	0
R/W:	R	R	R	R	R	R/(W)*	R/(W)*	R

Note: * 0 write only is valid after 1 is read at the AE and NMIF bits.

- Bits 15–10—Reserved bits: Data are 0 when read. The write value always be 0.

- Bits 9–8—Priority Mode 1 and 0 (PR1 and PR0): These bits determine the priority level of channels for execution when transfer requests are made for several channels simultaneously.

Bit 9: PR1	Bit 8: PR0	Description
0	0	CH0 > CH1 > CH2 > CH3 (initial value)
0	1	CH0 > CH2 > CH3 > CH1
1	0	CH2 > CH0 > CH1 > CH3
1	1	Round robin mode

- Bits 7–3—Reserved bits: Data are 0 when read. The write value always be 0.
- Bit 2—Address Error Flag (AE): Indicates that an address error has occurred during DMA transfer. If this bit is set during a data transfer, transfers on all channels are suspended. The CPU cannot write a 1 to the AE bit. Clearing is effected by 0 write after 1 read.

Bit 2: AE	Description
0	No address error, DMA transfer enabled (initial value) Clearing condition: Write AE = 0 after reading AE = 1
1	Address error, DMA transfer disabled Setting condition: Address error due to DMAC

- Bit 1—NMI Flag (NMIF): Indicates input of an NMI. This bit is set irrespective of whether the DMAC is operating or suspended. If this bit is set during a data transfer, transfers on all channels are suspended. The CPU is unable to write a 1 to the NMIF. Clearing is effected by a 0 write after 1 read.

Bit 1: NMIF	Description
0	No NMI interrupt, DMA transfer enabled (initial value) Clearing condition: Write NMIF = 0 after reading NMIF = 1
1	NMI has occurred, DMC transfer prohibited Set condition: NMI interrupt occurrence

- Bit 0—DMAC Master Enable (DME): This bit enables activation of the entire DMAC. When the DME bit and DE bit of the CHCR for the corresponding channel are set to 1, that channel is transfer-enabled. If this bit is cleared during a data transfer, transfers on all channels are suspended.

  Even when the DME bit is set, when the TE bit of the CHCR is 1, or its DE bit is 0, transfer is disabled in the case of an NMI of the DMAOR or when AE = 1.

Bit 0: DME	Description
0	Disable operation on all channels (initial value)
1	Enable operation on all channels

11.3 Operation

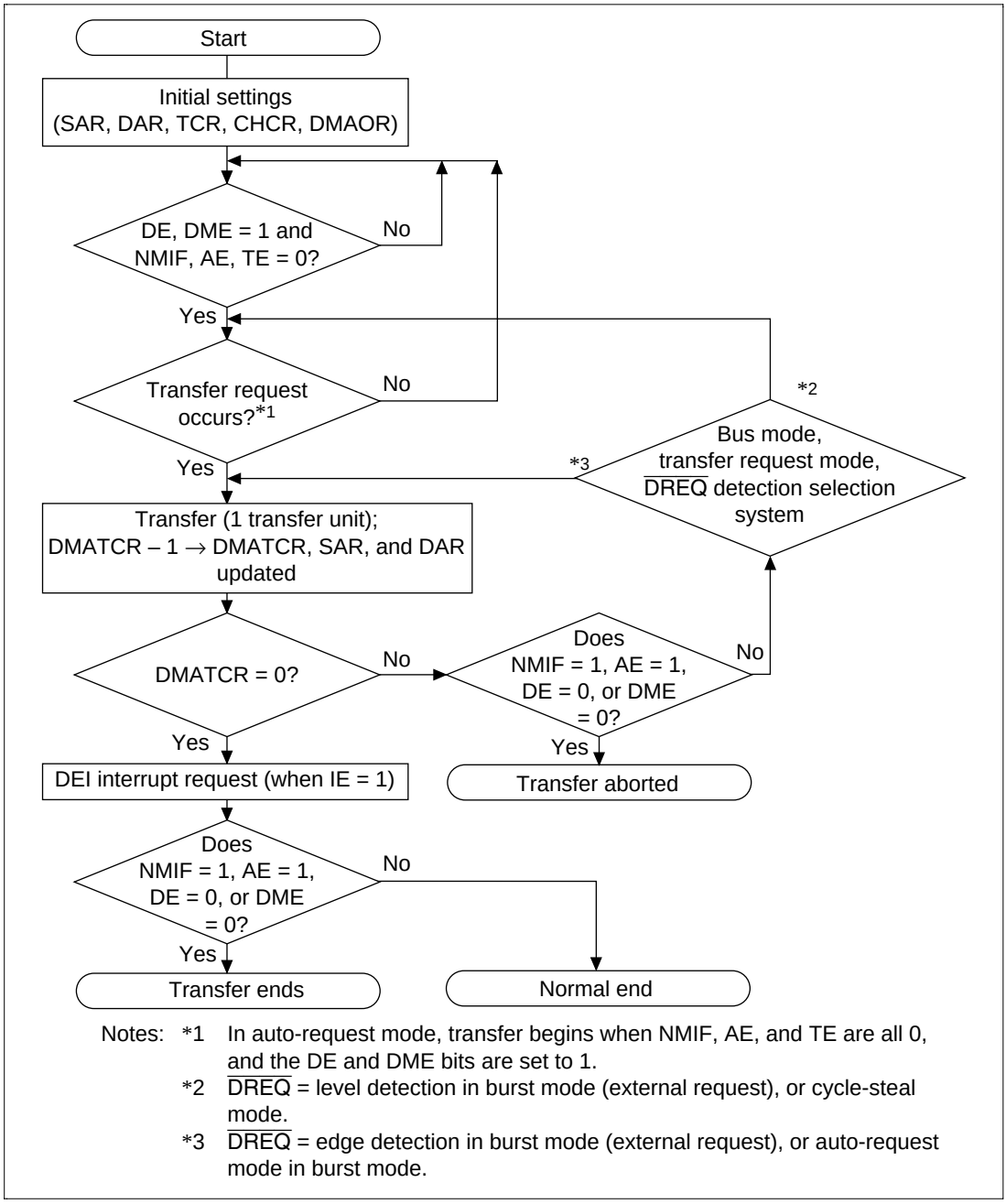
When there is a DMA transfer request, the DMAC starts the transfer according to the predetermined channel priority order; when the transfer end conditions are satisfied, it ends the transfer. Transfers can be requested in three modes: auto-request, external request, and on-chip peripheral module request. Transfer can be in either the single address mode or the dual address mode, and dual address mode can be either direct or indirect address transfer mode. The bus mode can be either burst or cycle steal.

11.3.1 DMA Transfer Flow

After the DMA source address registers (SAR), DMA destination address registers (DAR), DMA transfer count register (DMATCR), DMA channel control registers (CHCR), and DMA operation register (DMAOR) are set to the desired transfer conditions, the DMAC transfers data according to the following procedure:

1. The DMAC checks to see if transfer is enabled (DE = 1, DME = 1, TE = 0, NMIF = 0, AE = 0).
2. When a transfer request comes and transfer has been enabled, the DMAC transfers 1 transfer unit of data (determined by TS0 and TS1 setting). For an auto-request, the transfer begins automatically when the DE bit and DME bit are set to 1. The DMATCR value will be decremented by 1 upon each transfer. The actual transfer flows vary by address mode and bus mode.
3. When the specified number of transfers have been completed (when DMATCR reaches 0), the transfer ends normally. If the IE bit of the CHCR is set to 1 at this time, a DEI interrupt is sent to the CPU.
4. When an address error occurs in the DMAC or an NMI interrupt is generated, the transfer is aborted. Transfers are also aborted when the DE bit of the CHCR or the DME bit of the DMAOR are changed to 0.

Figure 11.2 is a flowchart of this procedure.



Notes: *1 In auto-request mode, transfer begins when NMIF, AE, and TE are all 0, and the DE and DME bits are set to 1.

*2 $\overline{\text{DREQ}}$ = level detection in burst mode (external request), or cycle-steal mode.

*3 $\overline{\text{DREQ}}$ = edge detection in burst mode (external request), or auto-request mode in burst mode.

Figure 11.2 DMAC Transfer Flowchart

11.3.2 DMA Transfer Requests

DMA transfer requests are usually generated in either the data transfer source or destination, but they can also be generated by devices and on-chip peripheral modules that are neither the source nor the destination. Transfers can be requested in three modes: auto-request, external request, and on-chip peripheral module request. The request mode is selected in the RS3–RS0 bits of the DMA channel control registers 0–3 (CHCR0–CHCR3).

Auto-Request Mode: When there is no transfer request signal from an external source, as in a memory-to-memory transfer or a transfer between memory and an on-chip peripheral module unable to request a transfer, the auto-request mode allows the DMAC to automatically generate a transfer request signal internally. When the DE bits of CHCR0–CHCR3 and the DME bit of the DMAOR are set to 1, the transfer begins (so long as the TE bits of CHCR0–CHCR3 and the NMIF and AE bits of DMAOR are all 0).

External Request Mode: In this mode a transfer is performed at the request signal ($\overline{\text{DREQ}}$) of an external device. Choose one of the modes shown in table 11.3 according to the application system. When this mode is selected, if the DMA transfer is enabled (DE = 1, DME = 1, TE = 0, NMIF = 0, AE = 0), a transfer is performed upon a request at the $\overline{\text{DREQ}}$ input. Choose to detect $\overline{\text{DREQ}}$ by either the falling edge or low level of the signal input with the DS bit of CHCR0–CHCR3 (DS = 0 is level detection, DS = 1 is edge detection). The source of the transfer request does not have to be the data transfer source or destination.

Table 11.3 Selecting External Request Modes with the RS Bits

RS3	RS2	RS1	RS0	Address Mode	Source	Destination
0	0	0	0	Dual address mode	Any*	Any*
0	0	1	0	Single address mode	External memory or memory-mapped external device	External device with DACK
0	0	1	1	Single address mode	External device with DACK	External memory or memory-mapped external device

Note: * External memory, memory-mapped external device, on-chip memory, on-chip peripheral module (excluding DMAC, DTC, BSC, UBC).

On-Chip Peripheral Module Request Mode: In this mode a transfer is performed at the transfer request signal (interrupt request signal) of an on-chip peripheral module. As indicated in table 11.4, there are ten transfer request signals: five from the multifunction timer pulse unit (MTU), which are compare match or input capture interrupts; the receive data full interrupts (RxI) and transmit data empty interrupts (TxI) of the two serial communication interfaces (SCI); and the A/D conversion end interrupt (ADI1 for A mask, ADI for others) of the A/D converter. When DMA

transfers are enabled (DE = 1, DME = 1, TE = 0, NMIF = 0, AE = 0), a transfer is performed upon the input of a transfer request signal.

The transfer request source need not be the data transfer source or transfer destination. However, when the transfer request is set by RxI (transfer request because SCI's receive data is full), the transfer source must be the SCI's receive data register (RDR). When the transfer request is set by TxI (transfer request because SCI's transmit data is empty), the transfer destination must be the SCI's transmit data register (TDR). Also, if the transfer request is set to the A/D converter, the data transfer destination must be the A/D converter register.

Table 11.4 Selecting On-Chip Peripheral Module Request Modes with the RS Bits

RS3	RS2	RS1	RS0	DMAC Transfer Request Source	DMA Transfer Request Signal	Source	Destination	Bus Mode
0	1	1	0	MTU*2	TGI0A	Any*1	Any*1	Burst/cycle steal
0	1	1	1	MTU*2	TGI1A	Any*1	Any*1	Burst/cycle steal
1	0	0	0	MTU*2	TGI2A	Any*1	Any*1	Burst/cycle steal
1	0	0	1	MTU*2	TGI3A	Any*1	Any*1	Burst/cycle steal
1	0	1	0	MTU*2	TGI4A	Any*1	Any*1	Burst/cycle steal
1	0	1	1	A/D	ADI*5	ADDR*4	Any*1	Burst/cycle steal
1	1	0	0	SCI0*3 transmit block	TxI0	Any*1	TDR0	Burst/cycle steal
1	1	0	1	SCI0*3 transmit block	RxI0	RDR0	Any*1	Burst/cycle steal
1	1	1	0	SCI1*3 transmit block	TxI1	Any*1	TDR1	Burst/cycle steal
1	1	1	1	SCI1*3 transmit block	RxI1	RDR1	Any*1	Burst/cycle steal

Notes: *1 External memory, memory-mapped external device, on-chip memory, on-chip peripheral module (excluding DMAC, DTC, BSC, UBC).
*2 MTU: Multifunction timer pulse unit.
*3 SCI0, SCI1: Serial communications interface.
*4 ADDR0, ADDR1: A/D converter's A/D register.
*5 ADI1 for A mask.

In order to output a transfer request from an on-chip peripheral module, set the relevant interrupt enable bit for each module, and output an interrupt signal.

When an on-chip peripheral module's interrupt request signal is used as a DMA transfer request signal, interrupts for the CPU are not generated.

When a DMA transfer is conducted corresponding with one of the transfer request signals in table 11.4, it is automatically discontinued. In cycle steal mode this occurs in the first transfer, and in burst mode with the last transfer.

11.3.3 Channel Priority

When the DMAC receives simultaneous transfer requests on two or more channels, it selects a channel according to a predetermined priority order, either in a fixed mode or in round robin mode. These modes are selected by priority bits PR1 and PR0 in the DMA operation register (DMAOR).

Fixed Mode: In these modes, the priority levels among the channels remain fixed.

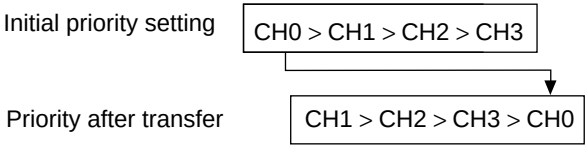
The following priority orders are available for fixed mode:

- CH0 > CH1 > CH2 > CH3
- CH0 > CH2 > CH3 > CH1
- CH2 > CH0 > CH1 > CH3

These are selected by settings of the PR1 and PR0 bits of the DMA operation register (DMAOR).

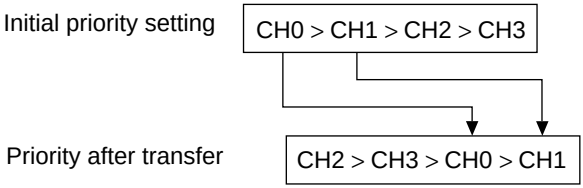
Round Robin Mode: In round robin mode, each time the transfer of one transfer unit (byte, word or long word) ends on a given channel, that channel receives the lowest priority level (figure 11.3). The priority level in round robin mode immediately after a reset is CH0 > CH1 > CH2 > CH3.

Transfer on channel 0



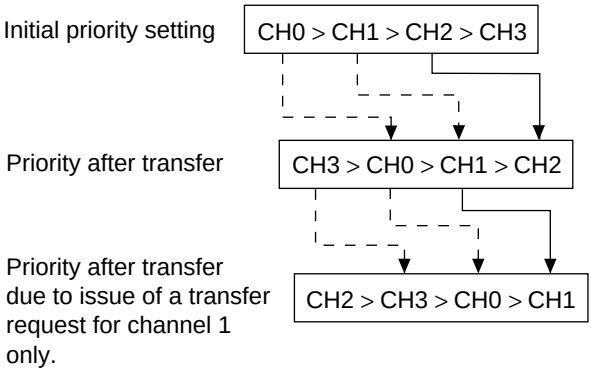
Channel 0 is given the lowest priority.

Transfer on channel 1



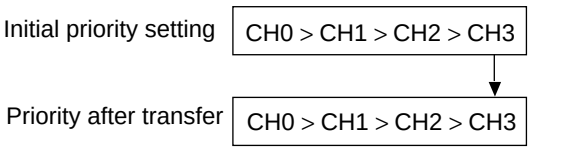
When channel 1 is given the lowest priority, the priority of channel 0, which was above channel 1, is also shifted simultaneously.

Transfer on channel 2



When channel 2 receives the lowest priority, the priorities of channel 0 and 1, which were above channel 2, are also shifted simultaneously. Immediately thereafter, if there is a transfer request for channel 1 only, channel 1 is given the lowest priority, and the priorities of channels 3 and 0 are simultaneously shifted down.

Transfer on channel 3



No change in priority.

Figure 11.3 Round Robin Mode

Figure 11.4 shows the changes in priority levels when transfer requests are issued simultaneously for channels 0 and 3, and channel 1 receives a transfer request during a transfer on channel 0. The DMAC operates in the following manner under these circumstances:

1. Transfer requests are issued simultaneously for channels 0 and 3.
2. Since channel 0 has a higher priority level than channel 3, the channel 0 transfer is conducted first (channel 3 is on transfer standby).
3. A transfer request is issued for channel 1 during a transfer on channel 0 (channels 1 and 3 are on transfer standby).
4. At the end of the channel 0 transfer, channel 0 shifts to the lowest priority level.
5. At this point, channel 1 has a higher priority level than channel 3, so the channel 1 transfer comes first (channel 3 is on transfer standby).
6. When the channel 1 transfer ends, channel 1 shifts to the lowest priority level.
7. Channel 3 transfer begins.
8. When the channel 3 transfer ends, channel 3 and channel 2 priority levels are lowered, giving channel 3 the lowest priority.

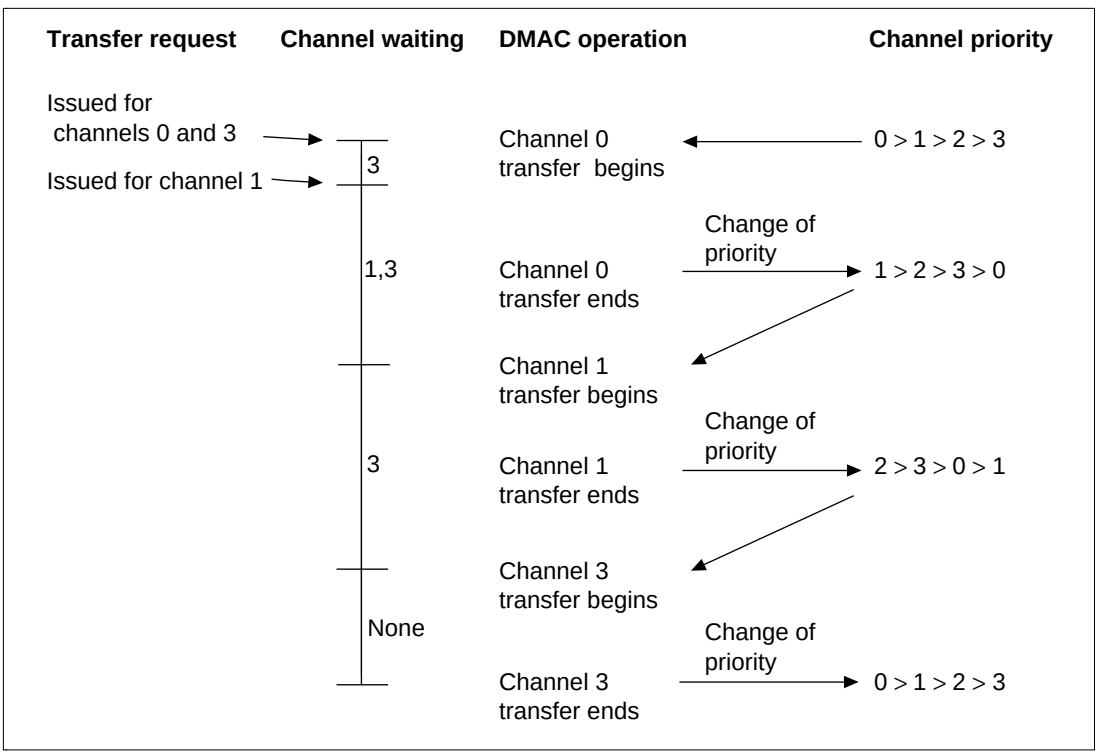


Figure 11.4 Example of Changes in Priority in Round Robin Mode

11.3.4 DMA Transfer Types

The DMAC supports the transfers shown in table 11.5. It can operate in the single address mode, in which either the transfer source or destination is accessed using an acknowledge signal, or dual access mode, in which both the transfer source and destination addresses are output. The dual access mode consists of a direct address mode, in which the output address value is the object of a direct data transfer, and an indirect address mode, in which the output address value is not the object of the data transfer, but the value stored at the output address becomes the transfer object address. The actual transfer operation timing varies with the bus mode. The DMAC has two bus modes: cycle-steal mode and burst mode.

Table 11.5 Supported DMA Transfers

		Destination			
Source	**External Device with DACK**	**External Memory**	**Memory-Mapped External Device**	**On-Chip Memory**	**On-Chip Peripheral Module**
External device with DACK	Not available	Single	Single	Not available	Not available
External memory	Single	Dual	Dual	Dual	Dual
Memory-mapped external device	Single	Dual	Dual	Dual	Dual
On-chip memory	Not available	Dual	Dual	Dual	Dual
On-chip peripheral module	Not available	Dual	Dual	Dual	Dual

Notes: 1. Single: Single address mode
2. Dual: Dual address mode; includes both direct address mode and indirect address mode.

11.3.5 Address Modes

Single Address Mode: In the single address mode, both the transfer source and destination are external; one (selectable) is accessed by a DACK signal while the other is accessed by an address. In this mode, the DMAC performs the DMA transfer in 1 bus cycle by simultaneously outputting a transfer request acknowledge DACK signal to one external device to access it while outputting an address to the other end of the transfer. Figure 11.5 shows an example of a transfer between an external memory and an external device with DACK in which the external device outputs data to the data bus while that data is written in external memory in the same bus cycle.

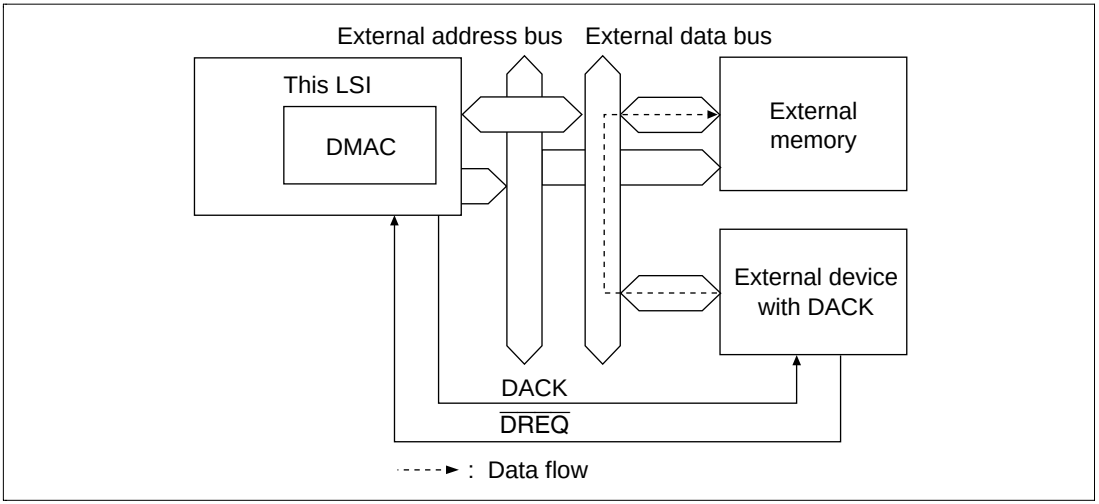


Figure 11.5 Data Flow in Single Address Mode

Two types of transfers are possible in the single address mode: (a) transfers between external devices with DACK and memory-mapped external devices, and (b) transfers between external devices with DACK and external memory. The only transfer requests for either of these is the external request ($\overline{\text{DREQ}}$). Figure 11.6 shows the DMA transfer timing for the single address mode.

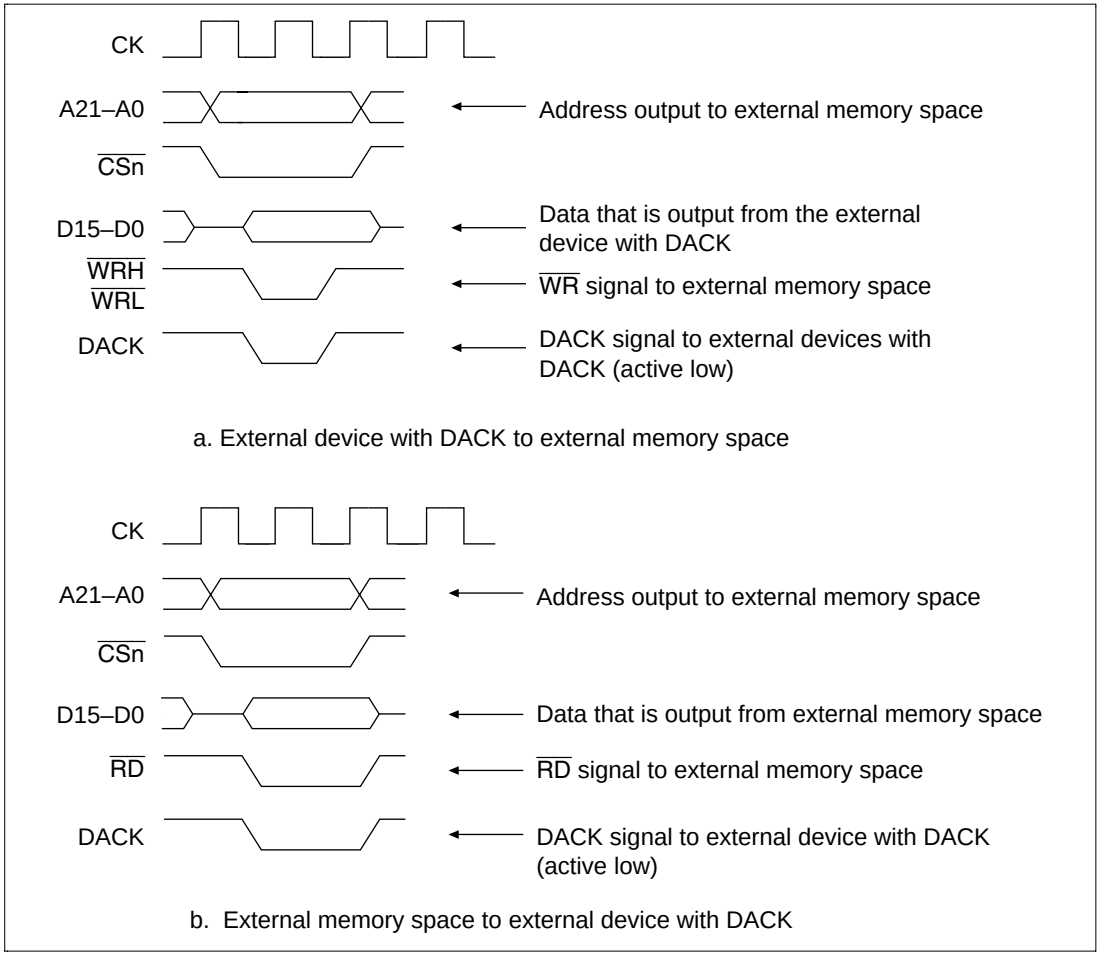


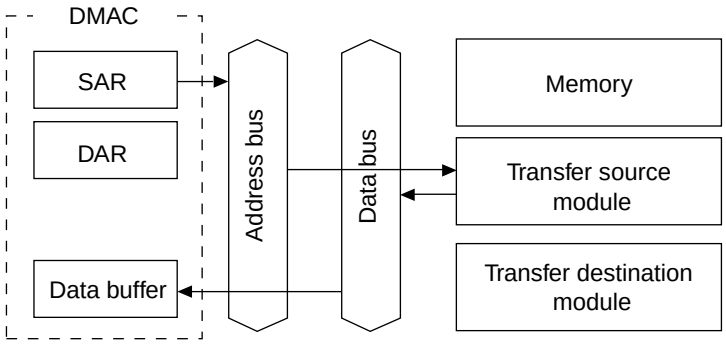
Figure 11.6 Example of DMA Transfer Timing in the Single Address Mode

11.3.6 Dual Address Mode

Dual address mode is used for access of both the transfer source and destination by address. Transfer source and destination can be accessed either internally or externally. Dual address mode is subdivided into two other modes: direct address transfer mode and indirect address transfer mode.

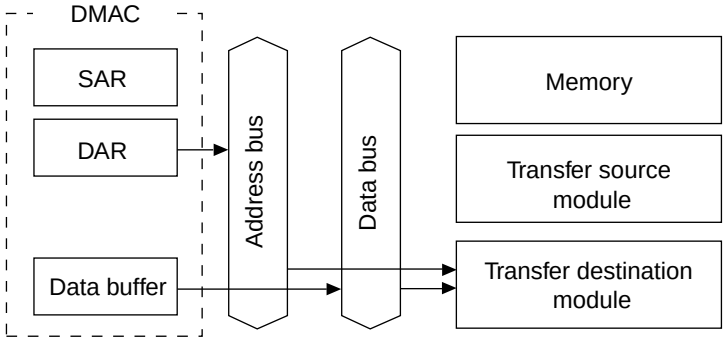
Direct Address Transfer Mode: Data is read from the transfer source during the data read cycle, and written to the transfer destination during the write cycle, so transfer is conducted in two bus cycles. At this time, the transfer data is temporarily stored in the DMAC. With the kind of external memory transfer shown in figure 11.7, data is read from one of the memories by the DMAC during a read cycle, then written to the other external memory during the subsequent write cycle. Figure 11.8 shows the timing for this operation.

1st bus cycle



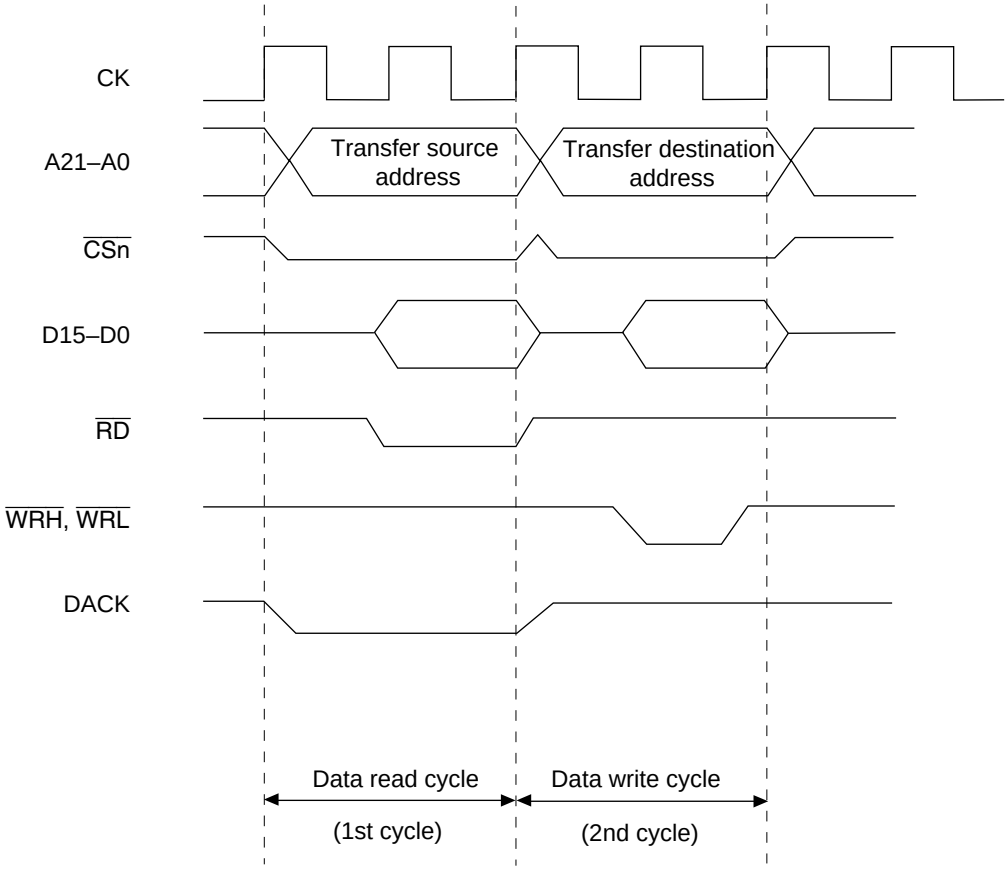
The SAR value is taken as the address, and data is read from the transfer source module and stored temporarily in the DMAC.

2nd bus cycle



The DAR value is taken as the address, and data stored in the DMAC's data buffer is written to the transfer destination module.

Figure 11.7 Direct Address Operation during Dual Address Mode



Note: Transfer between external memories with DACK are output during read cycle.

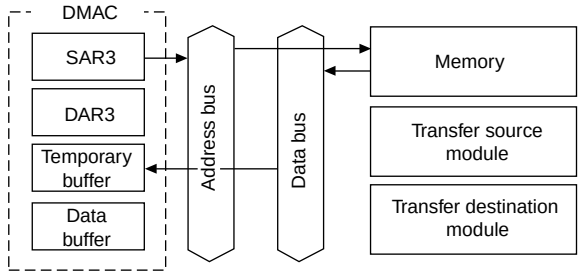
Figure 11.8 Example of Direct Address Transfer Timing in Dual Address Mode

Indirect Address Transfer Mode: In this mode the memory address storing the data you actually want to transfer is specified in DMAC internal transfer source address register (SAR3). Therefore, in indirect address transfer mode, the DMAC internal transfer source address register value is read first. This value is stored once in the DMAC. Next, the read value is output as the address, and the value stored at that address is again stored in the DMAC. Finally, the subsequent read value is written to the address specified by the transfer destination address register, ending one cycle of DMA transfer.

In indirect address mode (figure 11.9), transfer destination, transfer source, and indirect address storage destination are all 16-bit external memory locations, and transfer in this example is conducted in 16-bit or 8-bit units. Timing for this transfer example is shown in figure 11.10.

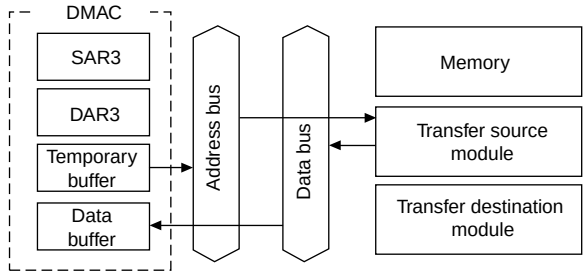
In indirect address mode, one NOP cycle (figure 11.10) is required until the data read as the indirect address is output to the address bus. When transfer data is 32-bit, the third and fourth bus cycles each need to be doubled, giving a required total of six bus cycles and one NOP cycle for the whole operation.

1st, 2nd bus cycles



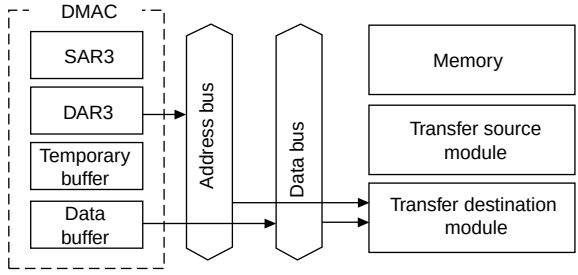
The SAR3 value is taken as the address, memory data is read, and the value is stored in the temporary buffer. Since the value read at this time is used as the address, it must be 32 bits. When external connection data bus is 16 bits, two bus cycles are required.

3rd bus cycle



The value in the temporary buffer is taken as the address, and data is read from the transfer source module to the data buffer.

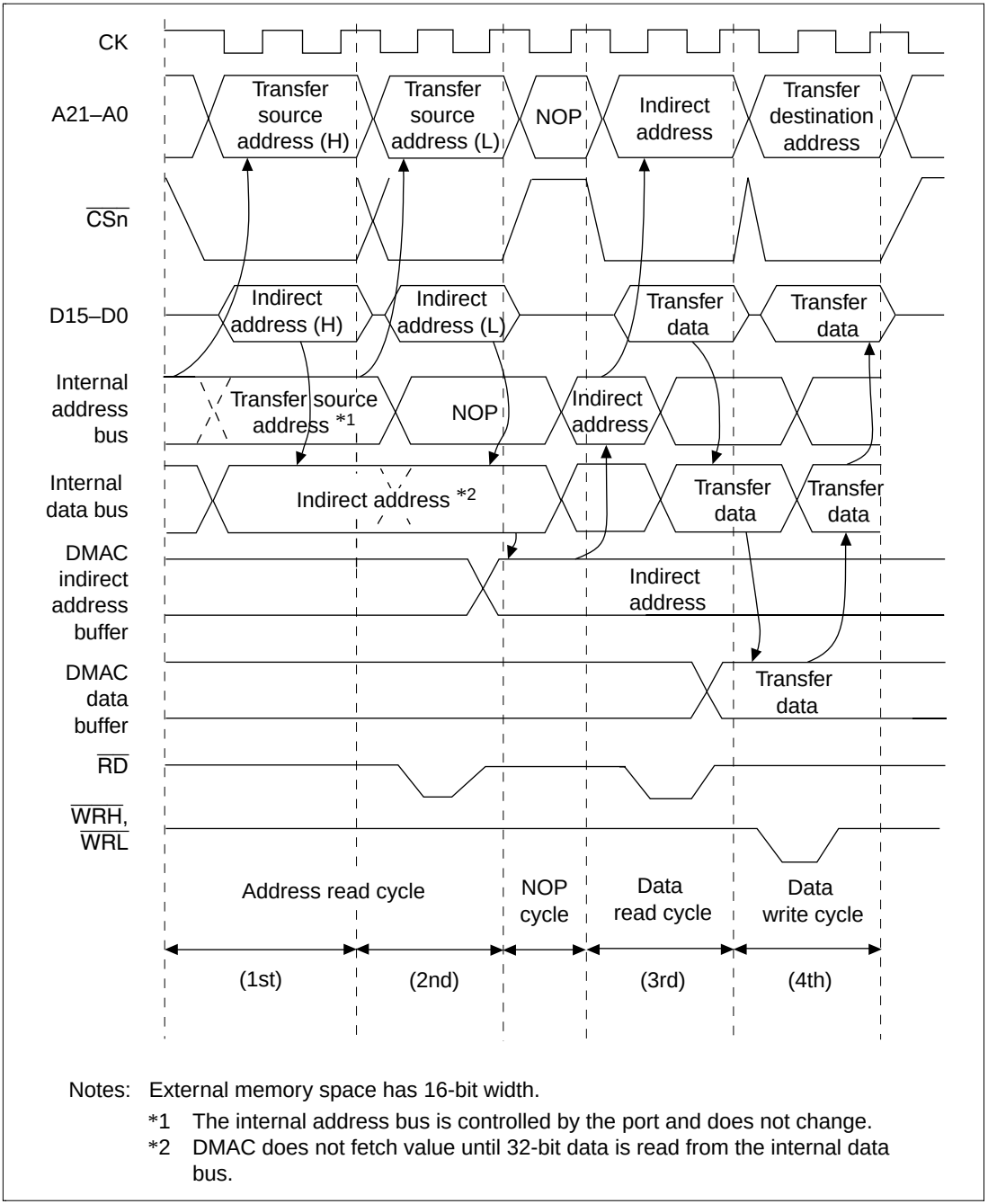
4th bus cycle



The DAR3 value is taken as the address, and the value in the data buffer is written to the transfer destination module.

Note: Memory, transfer source, and transfer destination modules are shown here. In practice, connection can be made anywhere there is address space.

Figure 11.9 Dual Address Mode and Indirect Address Operation (When External Memory Space is 16 bits)



Notes: External memory space has 16-bit width.

*1 The internal address bus is controlled by the port and does not change.

*2 DMAC does not fetch value until 32-bit data is read from the internal data bus.

Figure 11.10 Dual Address Mode and Indirect Address Transfer Timing Example 1 (External Memory Space to External Memory Space)

Figure 11.11 shows an example of timing in indirect address mode when transfer source and indirect address storage locations are in internal memory, the transfer destination is an on-chip peripheral module with 2-cycle access space, and transfer data is 8-bit.

Since the indirect address storage destination and the transfer source are in internal memory, these can be accessed in one cycle. The transfer destination is 2-cycle access space, so two data write cycles are required. One NOP cycle is required until the data read as the indirect address is output to the address bus.

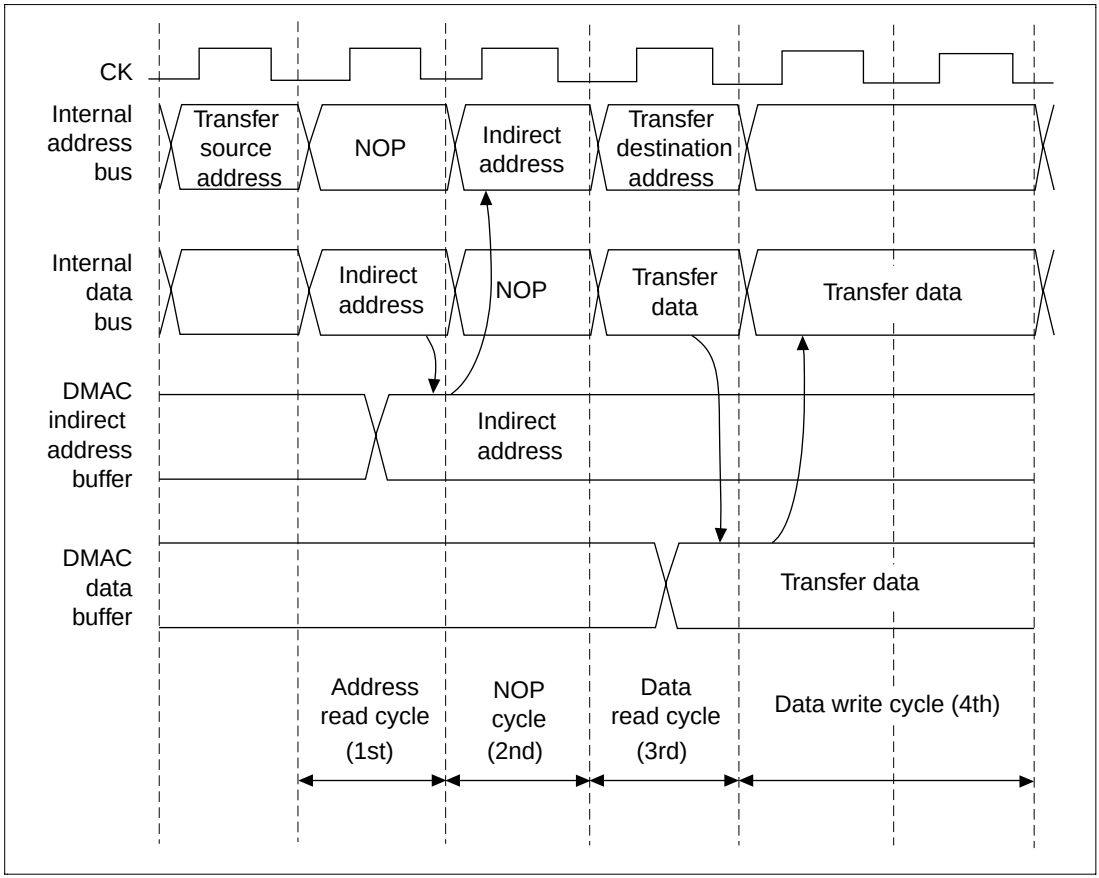


Figure 11.11 Dual Address Mode and Indirect Address Transfer Timing Example 2 (On-chip Memory Space to On-chip Memory Space)

11.3.7 Bus Modes

Select the appropriate bus mode in the TM bits of CHCR0–CHCR3. There are two bus modes: cycle steal and burst.

Cycle-Steal Mode: In the cycle steal mode, the bus right is given to another bus master after each one-transfer-unit (byte, word, or longword) DMAC transfer. When the next transfer request occurs, the bus rights are obtained from the other bus master and a transfer is performed for one transfer unit. When that transfer ends, the bus right is passed to the other bus master. This is repeated until the transfer end conditions are satisfied.

The cycle steal mode can be used with all categories of transfer destination, transfer source and transfer request. Figure 11.12 shows an example of DMA transfer timing in the cycle steal mode. Transfer conditions are dual address mode and $\overline{\text{DREQ}}$ level detection.

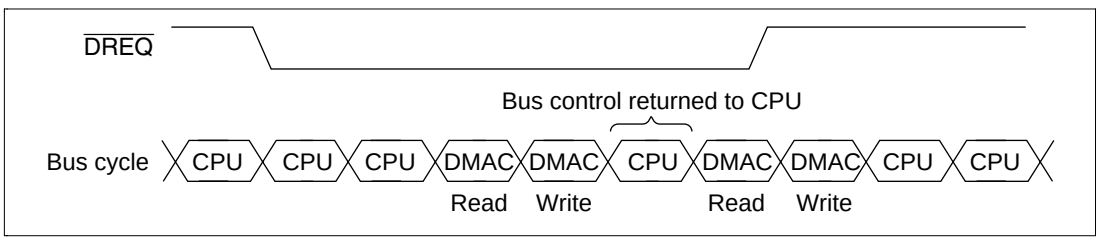


Figure 11.12 DMA Transfer Example in the Cycle-Steal Mode

Burst Mode: Once the bus right is obtained, the transfer is performed continuously until the transfer end condition is satisfied. In the external request mode with low level detection of the $\overline{\text{DREQ}}$ pin, however, when the $\overline{\text{DREQ}}$ pin is driven high, the bus passes to the other bus master after the bus cycle of the DMAC that currently has an acknowledged request ends, even if the transfer end conditions have not been satisfied.

Figure 11.13 shows an example of DMA transfer timing in the burst mode. Transfer conditions are single address mode and $\overline{\text{DREQ}}$ level detection.

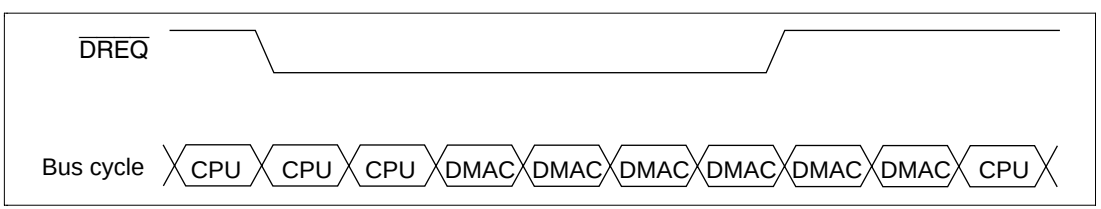


Figure 11.13 DMA Transfer Example in the Burst Mode

11.3.8 Relationship between Request Modes and Bus Modes by DMA Transfer Category

Table 11.6 shows the relationship between request modes and bus modes by DMA transfer category.

Table 11.6 Relationship of Request Modes and Bus Modes by DMA Transfer Category

Address Mode	Transfer Category	Request Mode	Bus*6 Mode	Transfer Size (Bits)	Usable Channels
Single	External device with DACK and external memory	External	B/C	8/16/32	0, 1
	External device with DACK and memory-mapped external device	External	B/C	8/16/32	0, 1
Dual	External memory and external memory	Any*1	B/C	8/16/32	0–3*5
	External memory and memory-mapped external device	Any*1	B/C	8/16/32	0–3*5
	Memory-mapped external device and memory-mapped external device	Any*1	B/C	8/16/32	0–3*5
	External memory and on-chip memory	Any*1	B/C	8/16/32	0–3*5
	External memory and on-chip peripheral module	Any*2	B/C*3	8/16/32*4	0–3*5
	Memory-mapped external device and on-chip memory	Any*1	B/C	8/16/32	0–3*5
	Memory-mapped external device and on-chip peripheral module	Any*2	B/C*3	8/16/32*4	0–3*5
	On-chip memory and on-chip memory	Any*1	B/C	8/16/32	0–3*5
	On-chip memory and on-chip peripheral module	Any*2	B/C*3	8/16/32*4	0–3*5
	On-chip peripheral module and on-chip peripheral module	Any*2	B/C*3	8/16/32*4	0–3*5

Notes: *1 External request, auto-request or on-chip peripheral module request enabled. However, in the case of on-chip peripheral module request, it is not possible to specify the SCI or A/D converter for the transfer request source.

*2 External request, auto-request or on-chip peripheral module request possible. However, if transfer request source is also the SCI or A/D converter (A/D1 for A mask), the transfer source or transfer destination must be the SCI or A/D converter (A/D1 for A mask). For A mask, setting A/D0 as the transfer request source is not permitted.

*3 When the transfer request source is the SCI, only cycle steal mode is possible.

*4 Access size permitted by register of on-chip peripheral module that is the transfer source or transfer destination.

*5 When the transfer request is an external request, channels 0 and 1 only can be used.

*6 B: Burst, C: Cycle steal

11.3.9 Bus Mode and Channel Priority Order

When a given channel is transferring in burst mode, and a transfer request is issued to channel 0, which has a higher priority ranking, transfer on channel 0 begins immediately. If the priority level setting is fixed mode (CH0 > CH1), channel 1 transfer is continued after transfer on channel 0 are completely ended, whether the channel 0 setting is cycle steal mode or burst mode.

When the priority level setting is for round robin mode, transfer on channel 1 begins after transfer of one transfer unit on channel 0, whether channel 0 is set to cycle steal mode or burst mode. Thereafter, bus right alternates in the order: channel 1 > channel 0 > channel 1 > channel 0. Whether the priority level setting is for fixed mode or round robin mode, since channel 1 is set to burst mode, the bus right is not given to the CPU. An example of round robin mode is shown in figure 11.14.

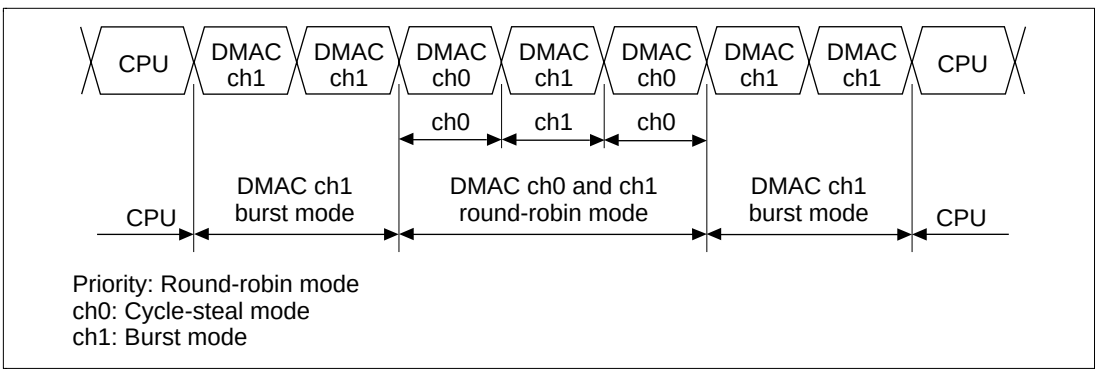


Figure 11.14 Bus Handling when Multiple Channels Are Operating

11.3.10 Number of Bus Cycle States and $\overline{\text{DREQ}}$ Pin Sample Timing

Number of States in Bus Cycle: The number of states in the bus cycle when the DMAC is the bus master is controlled by the bus state controller (BSC) just as it is when the CPU is the bus master. For details, see section 10, Bus State Controller (BSC).

$\overline{\text{DREQ}}$ Pin Sampling Timing and DRAK Signal: In external request mode, the $\overline{\text{DREQ}}$ pin is sampled by either falling edge or low-level detection. When a $\overline{\text{DREQ}}$ input is detected, a DMAC bus cycle is issued and DMA transfer effected, at the earliest, after three states. However, in burst mode when single address operation is specified, a dummy cycle is inserted for the first bus cycle. In this case, the actual data transfer starts from the second bus cycle. Data is transferred continuously from the second bus cycle. The dummy cycle is not counted in the number of transfer cycles, so there is no need to recognize the dummy cycle when setting the TCR.

$\overline{\text{DREQ}}$ sampling from the second time begins from the start of the transfer one bus cycle prior to the DMAC transfer generated by the previous sampling.

DRAK is output once for the first $\overline{\text{DREQ}}$ sampling, irrespective of transfer mode or $\overline{\text{DREQ}}$ detection method. In burst mode, using edge detection, $\overline{\text{DREQ}}$ is sampled for the first cycle only, so DRAK is also output for the first cycle only. Therefore, the $\overline{\text{DREQ}}$ signal negate timing can be ascertained, and this facilitates handshake operations of transfer requests with the DMAC.

Cycle Steal Mode Operations: In cycle steal mode, $\overline{\text{DREQ}}$ sampling timing is the same irrespective of dual or single address mode, or whether edge or low-level $\overline{\text{DREQ}}$ detection is used.

For example, DMAC transfer begins (figure 11.15), at the earliest, three cycles from the first sampling timing. The second sampling begins at the start of the transfer one bus cycle prior to the start of the DMAC transfer initiated by the first sampling (i.e., from the start of the CPU(3) transfer). At this point, if DREQ detection has not occurred, sampling is executed every cycle thereafter.

As in figure 11.16, whatever cycle the CPU transfer cycle is, the next sampling begins from the start of the transfer one bus cycle before the DMAC transfer begins.

Figure 11.15 shows an example of output during DACK read and figure 11.16 an example of output during DACK write.

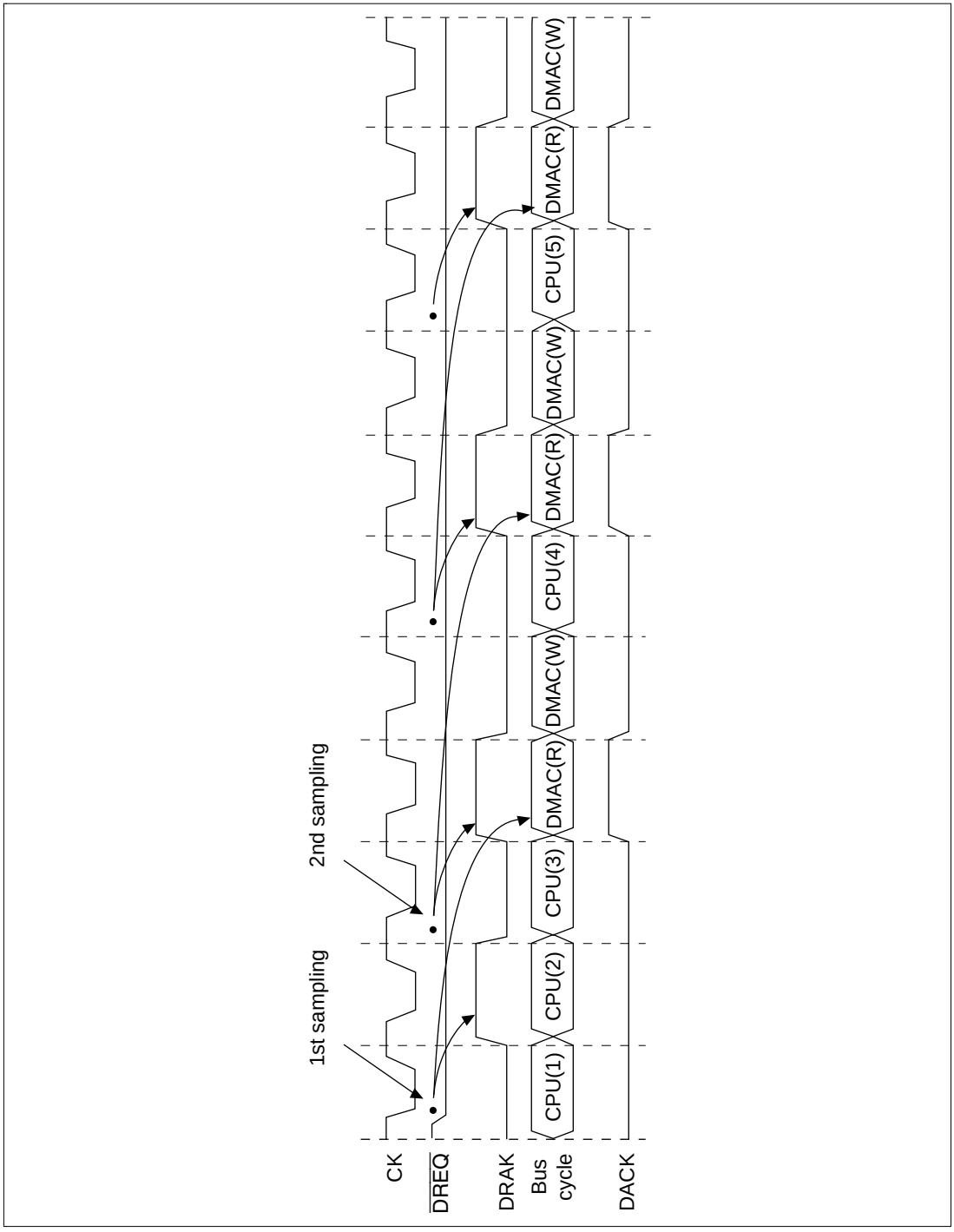


Figure 11.15 Cycle Steal, Dual Address, and Level Detection (Fastest Operation)

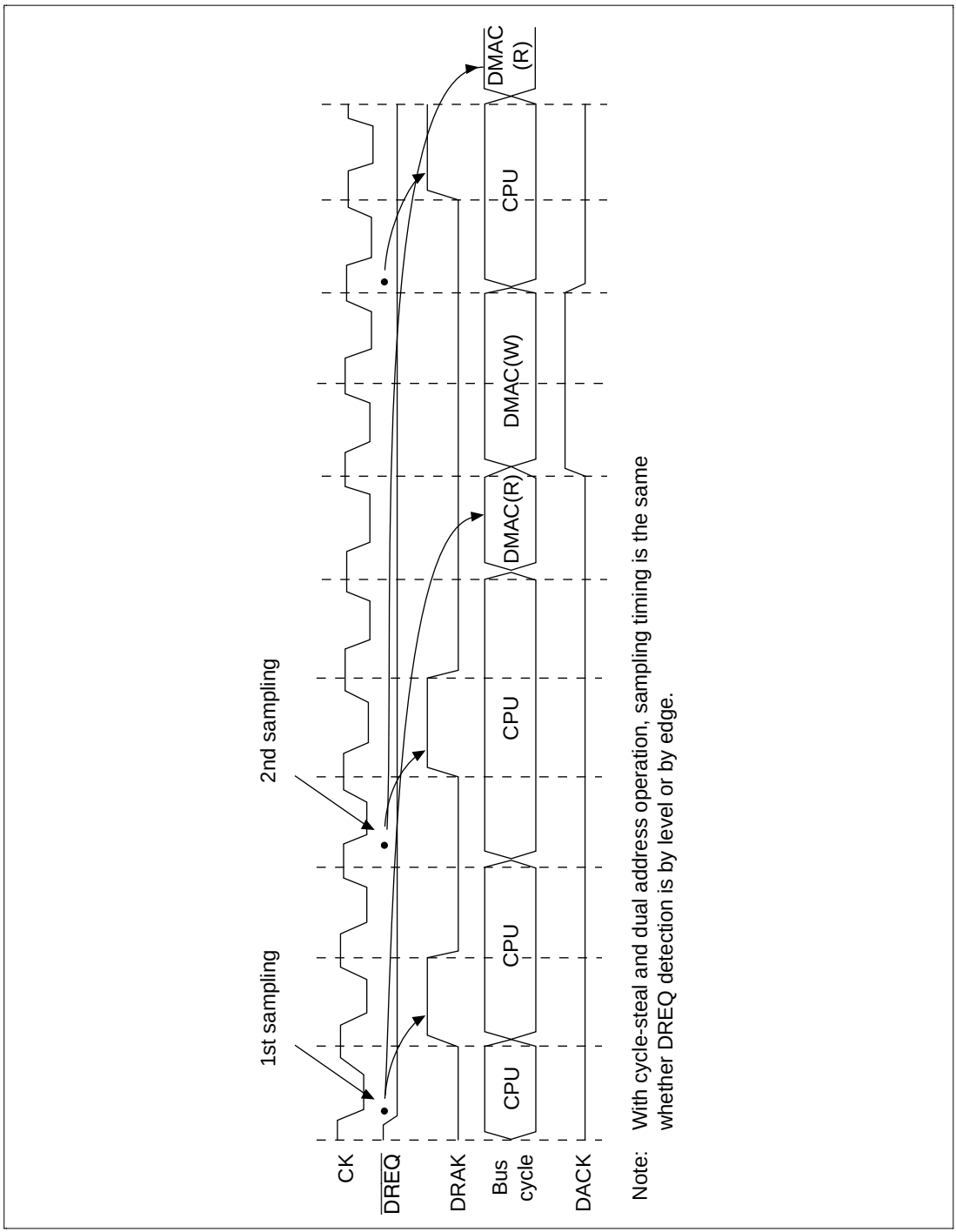


Figure 11.16 Cycle Steal, Dual Address, and Level Detection (Normal Operation)

Figures 11.17 and 11.18 show cycle steal mode and single address mode. In this case, transfer begins at earliest three cycles after the first $\overline{\text{DREQ}}$ sampling. The second sampling begins from the start of the transfer one bus cycle before the start of the first DMAC transfer. In single address mode, the DACK signal is output during the DMAC transfer period.

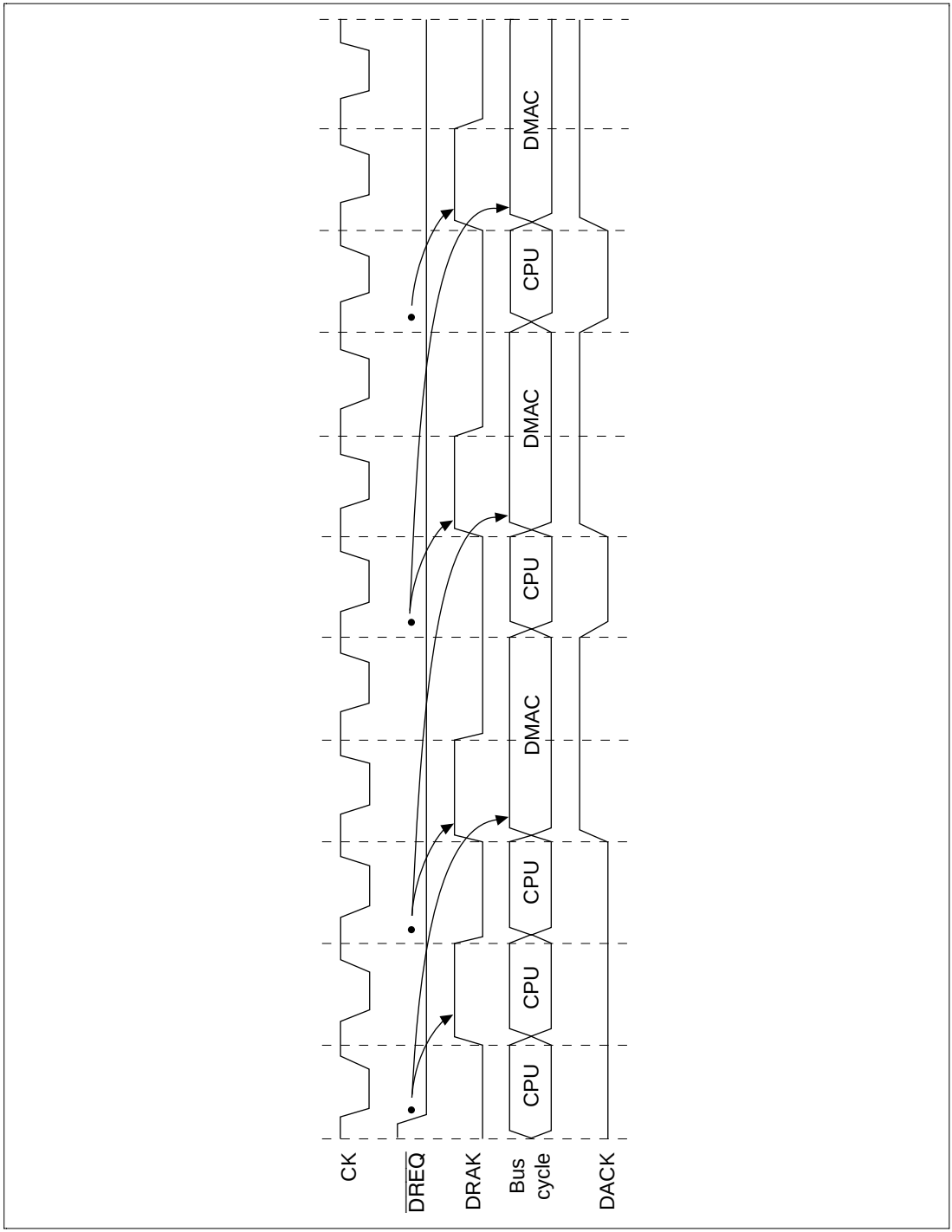


Figure 11.17 Cycle Steal, Single Address, and Level Detection (Fastest Operation)

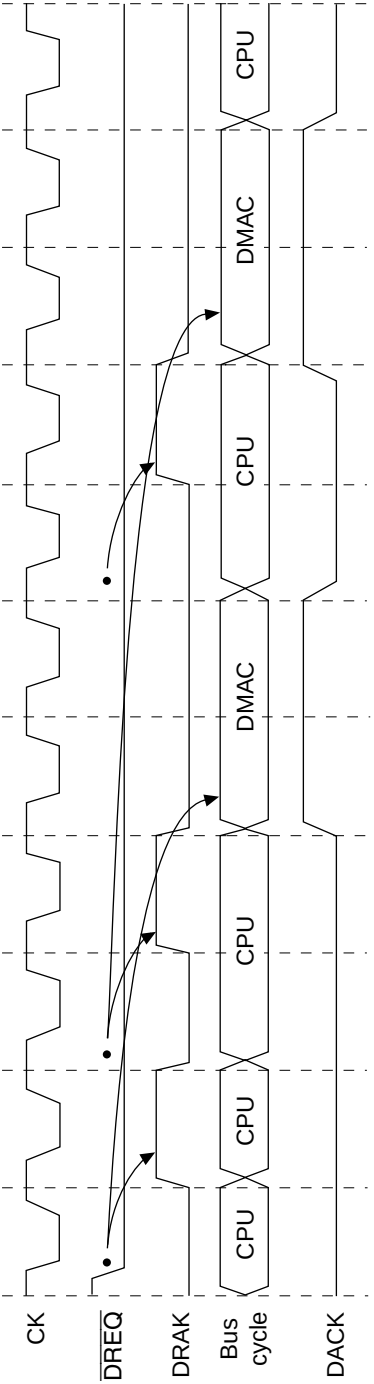


Figure 11.18 Cycle Steal, Single Address, and Level Detection (Normal Operation)

Burst Mode, Dual Address, and Level Detection: $\overline{\text{DREQ}}$ sampling timing in burst mode with dual address and level detection is virtually the same as that of cycle steal mode.

For example, DMAC transfer begins (figure 11.19), at the earliest, three cycles after the timing of the first sampling. The second sampling also begins from the start of the transfer one bus cycle before the start of the first DMAC transfer. In burst mode, as long as transfer requests are issued, DMAC transfer continues. Therefore, the “transfer one bus cycle before the start of the DMAC transfer” may be a DMAC transfer.

In burst mode, the DACK output period is the same as that of cycle steal mode. Figure 11.20 shows the normal operation of this burst mode.

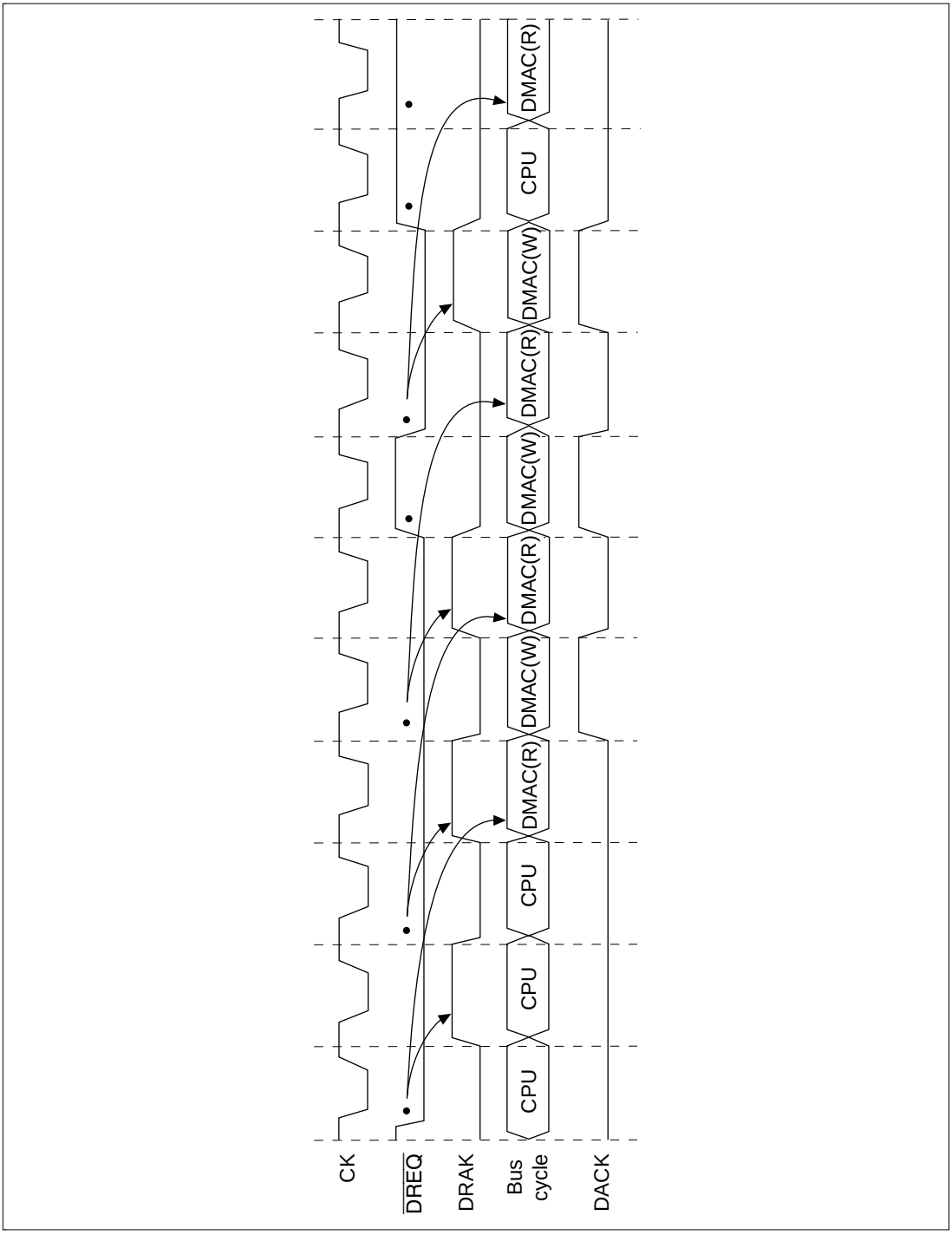


Figure 11.19 Burst Mode, Dual Address, and Level Detection (Fastest Operation)

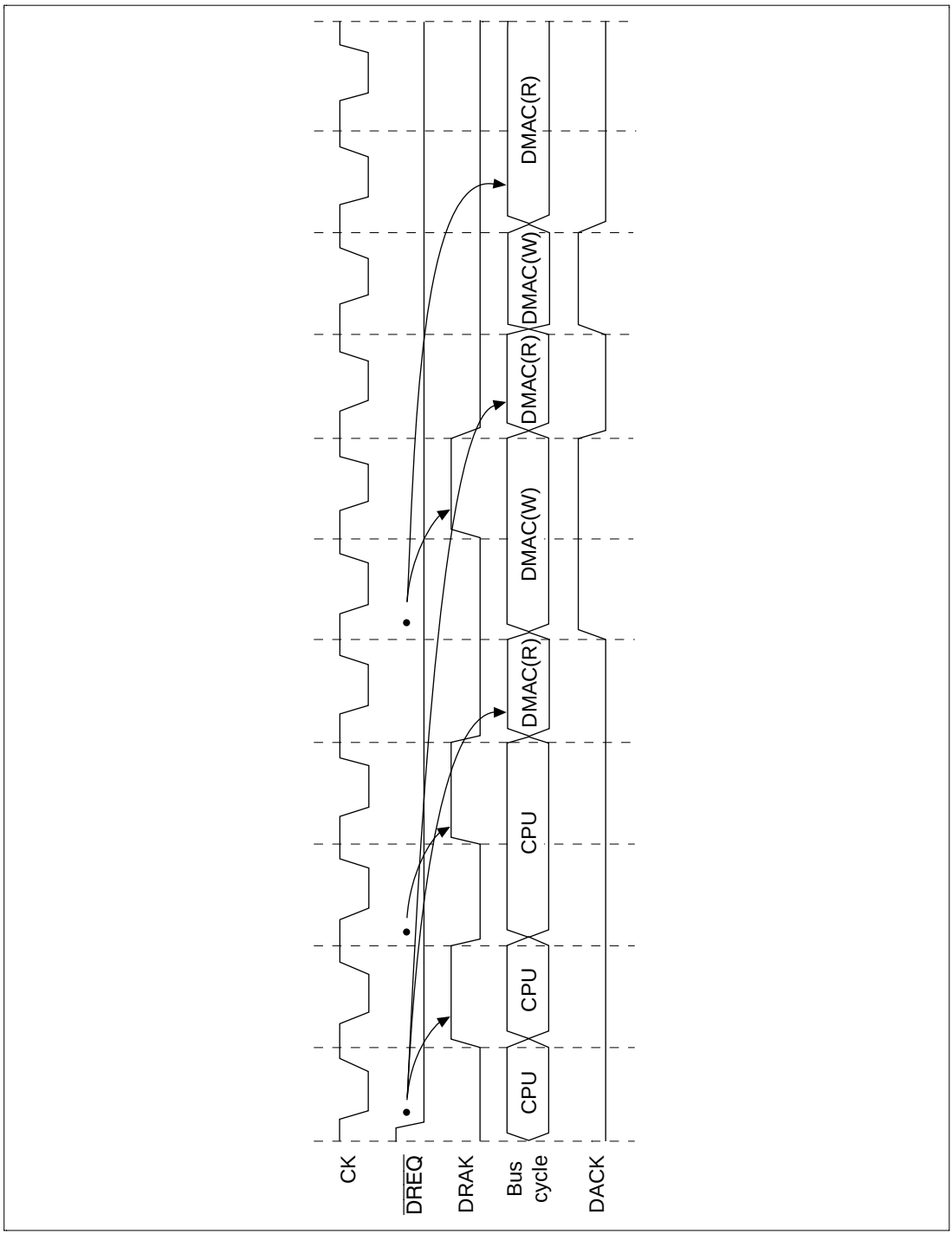


Figure 11.20 Burst Mode, Dual Address, and Level Detection (Normal Operation)

Burst Mode, Single Address, and Level Detection: $\overline{\text{DREQ}}$ sampling timing in burst mode with single address and level detection is shown in figures 11.21 and 11.22.

In burst mode with single address and level detection, a dummy cycle is inserted as one bus cycle, at the earliest, three cycles after timing of the first sampling. Data during this period is undefined, and the DACK signal is not output. Nor is the number of DMAC transfers counted. The actual DMAC transfer begins after one dummy bus cycle output.

The dummy cycle is not counted either at the start of the second sampling (transfer one bus cycle before the start of the first DMAC transfer). Therefore, the second sampling is not conducted from the bus cycle starting the dummy cycle, but from the start of the CPU(3) bus cycle.

Thereafter, as long the $\overline{\text{DREQ}}$ is continuously sampled, no dummy cycle is inserted. $\overline{\text{DREQ}}$ sampling timing during this period begins from the start of the transfer one bus cycle before the start of DMAC transfer, in the same way as with cycle steal mode.

As with the fourth sampling in figure 11.21, once DMAC transfer is interrupted, a dummy cycle is again inserted at the start as soon as DMAC transfer is resumed.

The DACK output period in burst mode is the same as in cycle steal mode.

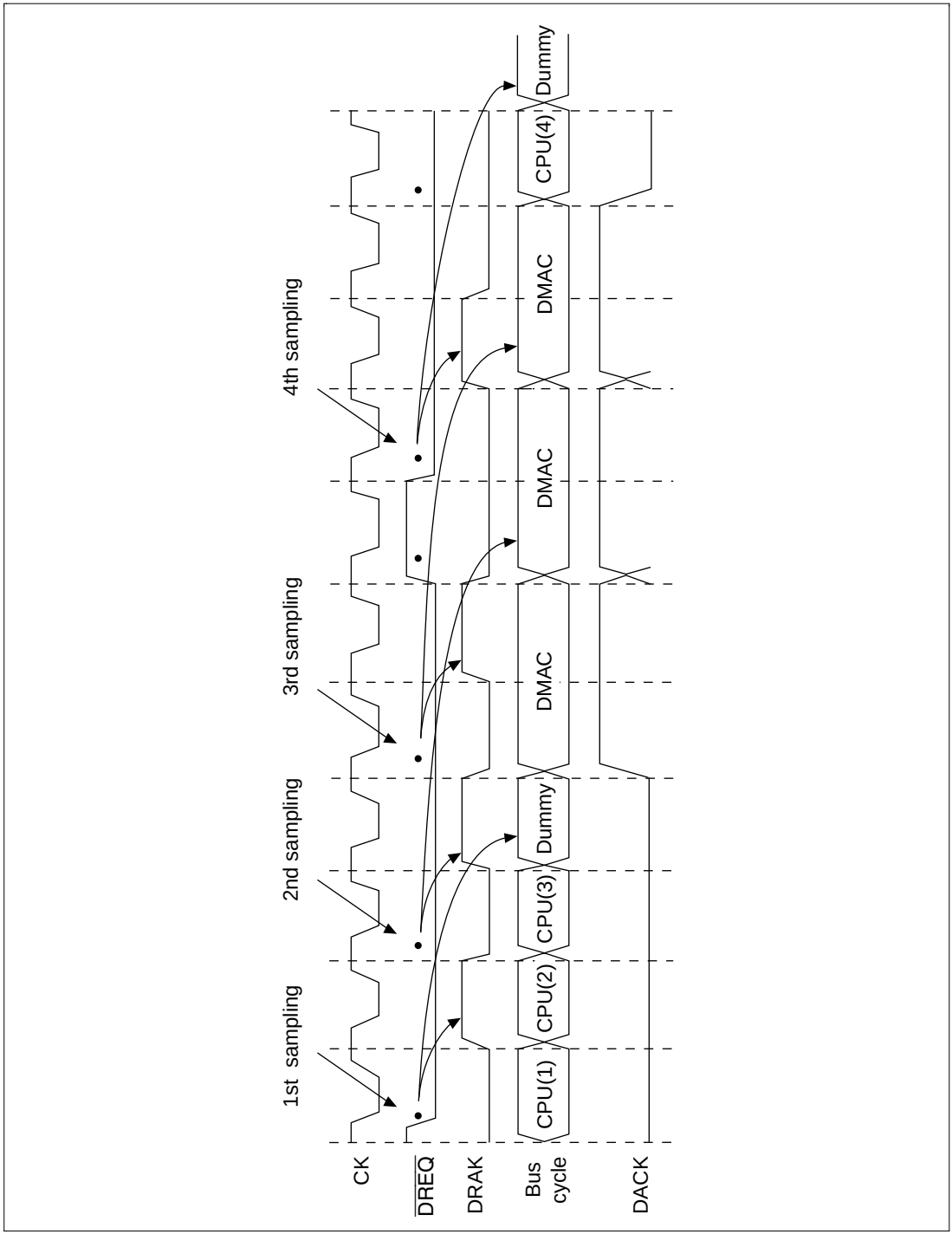


Figure 11.21 Burst Mode, Single Address, and Level Detection (Fastest Operation)

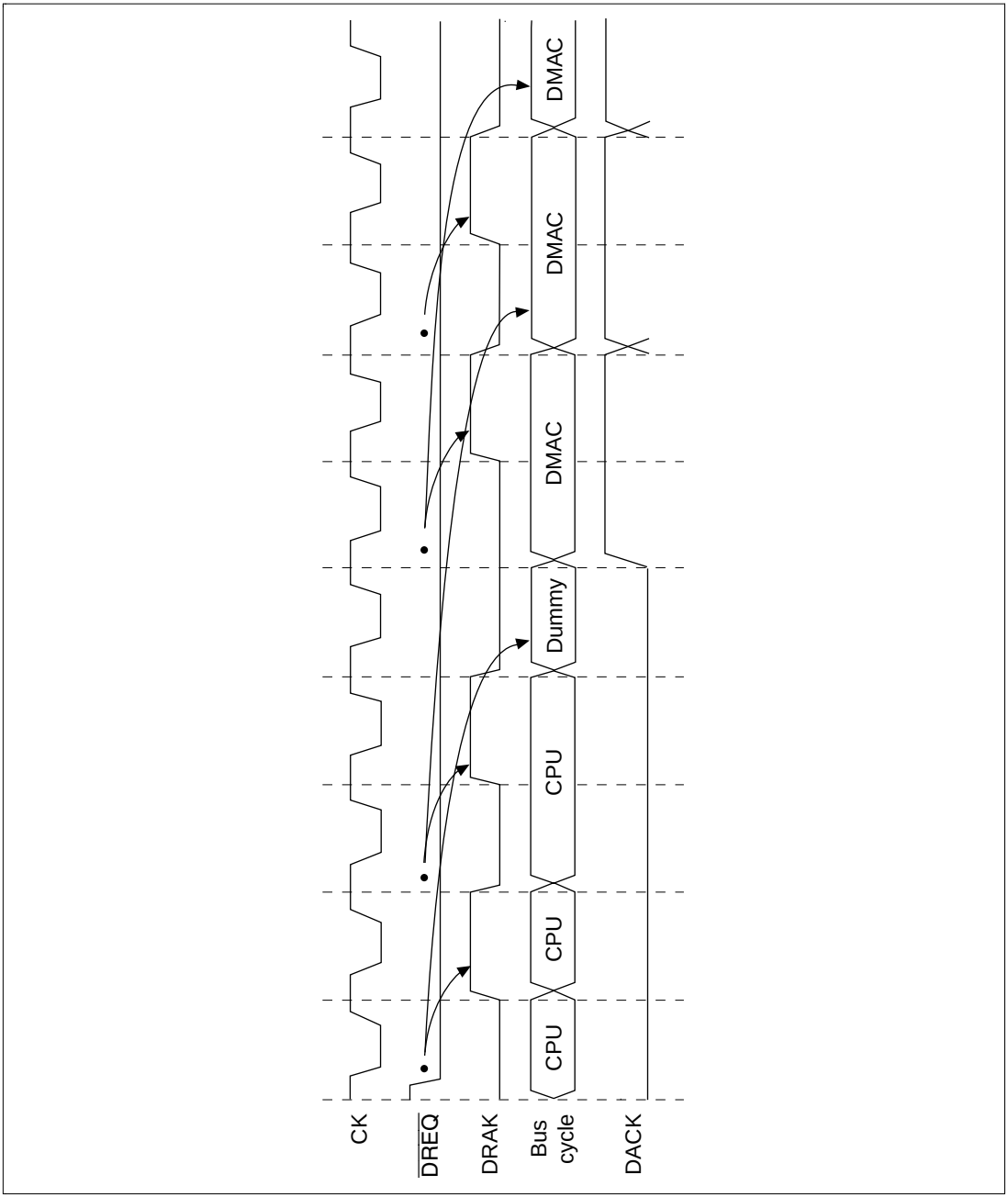


Figure 11.22 Burst Mode, Single Address, and Level Detection (Normal Operation)

Burst Mode, Dual Address, and Edge Detection: In burst mode with dual address and edge detection, $\overline{\text{DREQ}}$ sampling is conducted only on the first cycle.

In figure 11.23, DMAC transfer begins, at the earliest, three cycles after the timing of the first sampling. Thereafter, DMAC transfer continues until the end of the data transfer count set in the TCR. $\overline{\text{DREQ}}$ sampling is not conducted during this period. Therefore, DRAK is output on the first cycle only.

When DMAC transfer is resumed after being halted by a NMI or address error, be sure to reinput an edge request. The remaining transfer restarts after the first DRAK output.

The DACK output period in burst mode is the same as in cycle steal mode.

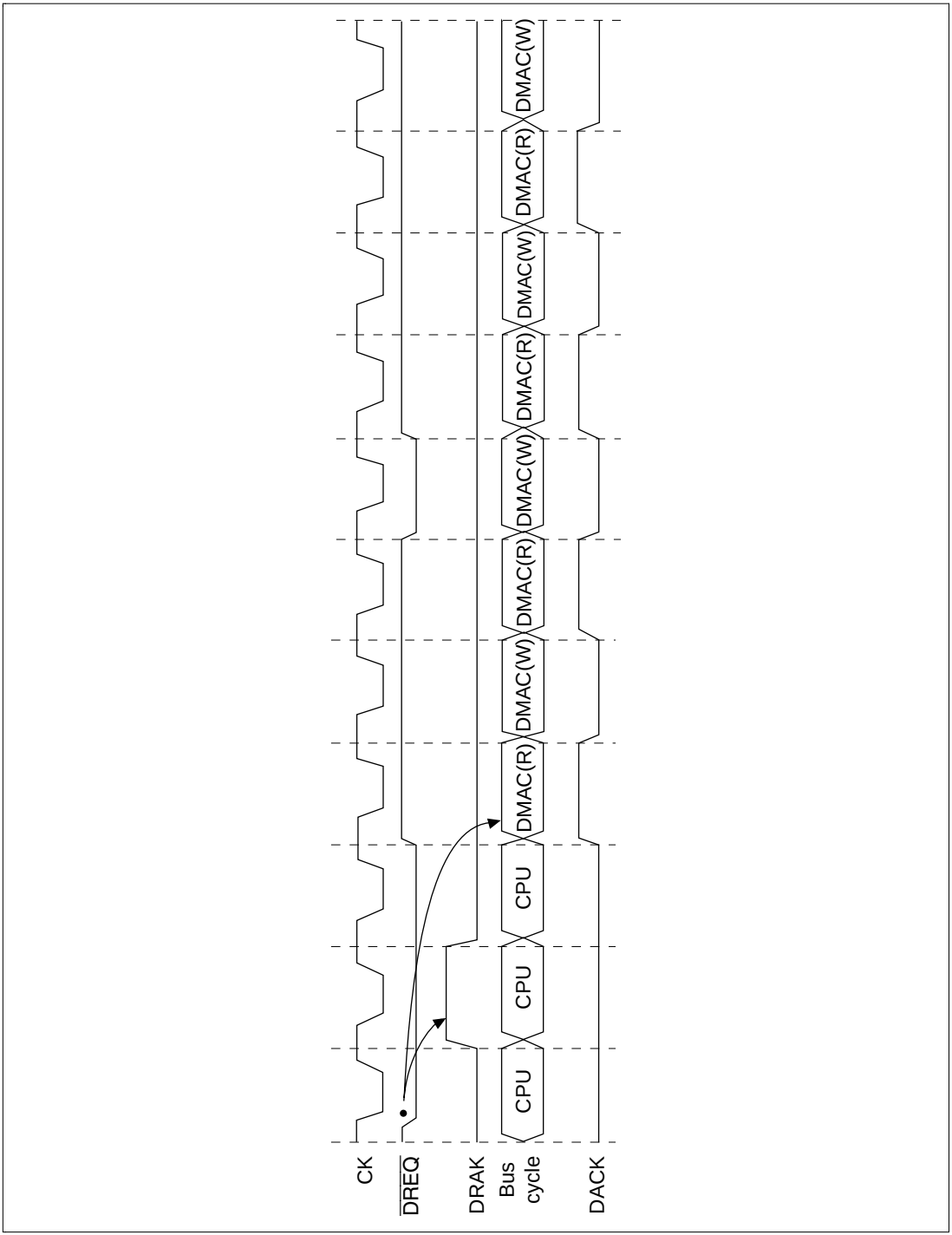


Figure 11.23 Burst Mode, Dual Address, and Edge Detection

Burst Mode, Single Address, and Edge Detection: In burst mode with single address and edge detection, $\overline{\text{DREQ}}$ sampling is conducted only on the first cycle. In figure 11.24, a dummy cycle is inserted, at the earliest, three cycles after the timing for the first sampling. During this period, data is undefined, and DACK is not output. Nor is the number of DMAC transfers counted. Thereafter, DMAC transfer continues until the data transfer count set in the DMATCR has ended. $\overline{\text{DREQ}}$ sampling is not conducted during this period. Therefore, DRAK is output on the first cycle only.

When DMAC transfer is resumed after being halted by a NMI or address error, be sure to reinput an edge request. DRAK is output once, and the remaining transfer restarts after output of one dummy cycle.

The DACK output period in burst mode is the same as in cycle steal mode.

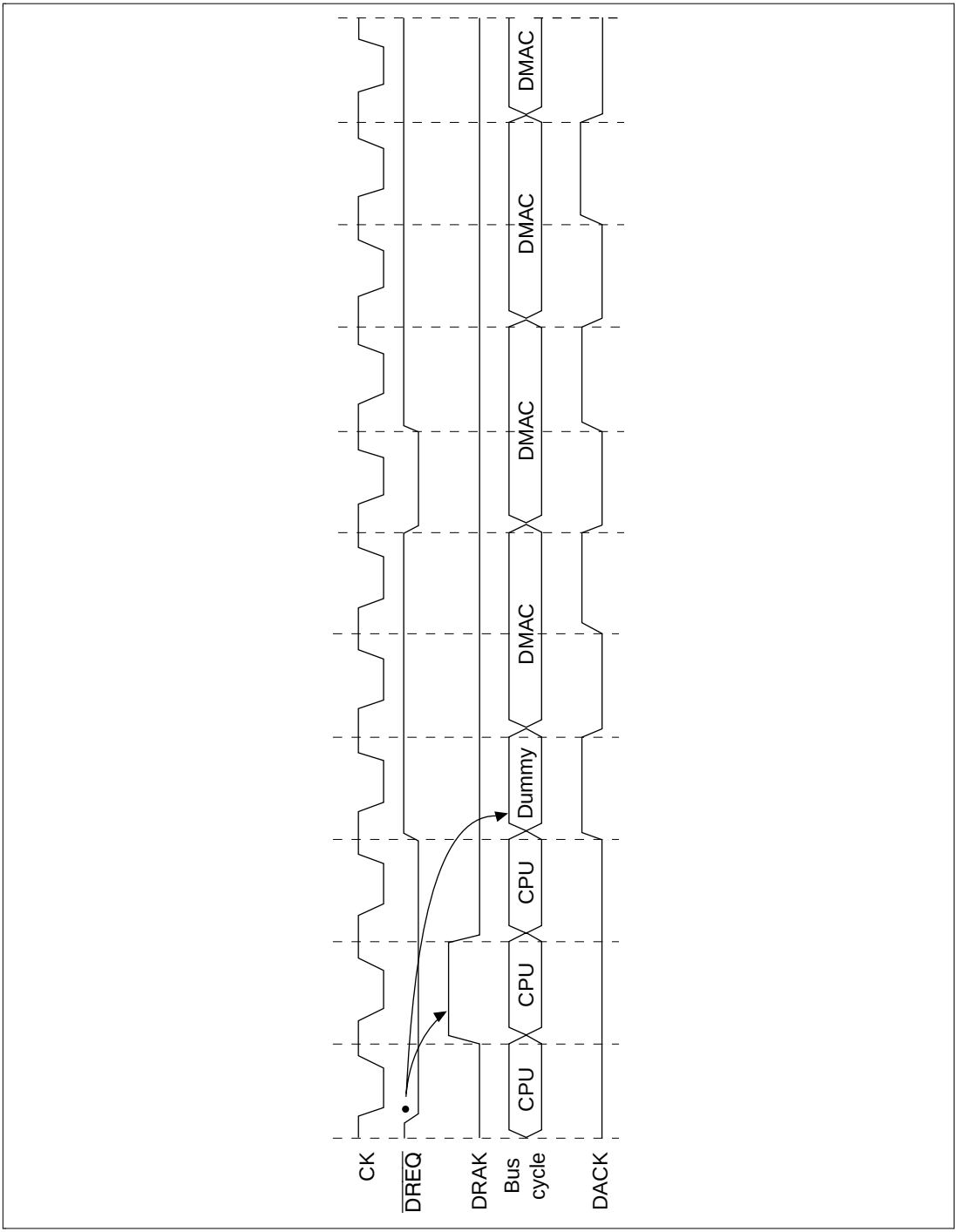


Figure 11.24 Burst Mode, Single Address and Edge Detection

11.3.11 Source Address Reload Function

Channel 2 has a source address reload function. This returns to the first value set in the source address register (SAR2) every four transfers by setting the RO bit of CHCR2 to 1. Figure 11.25 illustrates this operation. Figure 11.26 is a timing chart for reload ON mode, with burst mode, autorequest, 16-bit transfer data size, SAR2 increment, and DAR2 fixed mode.

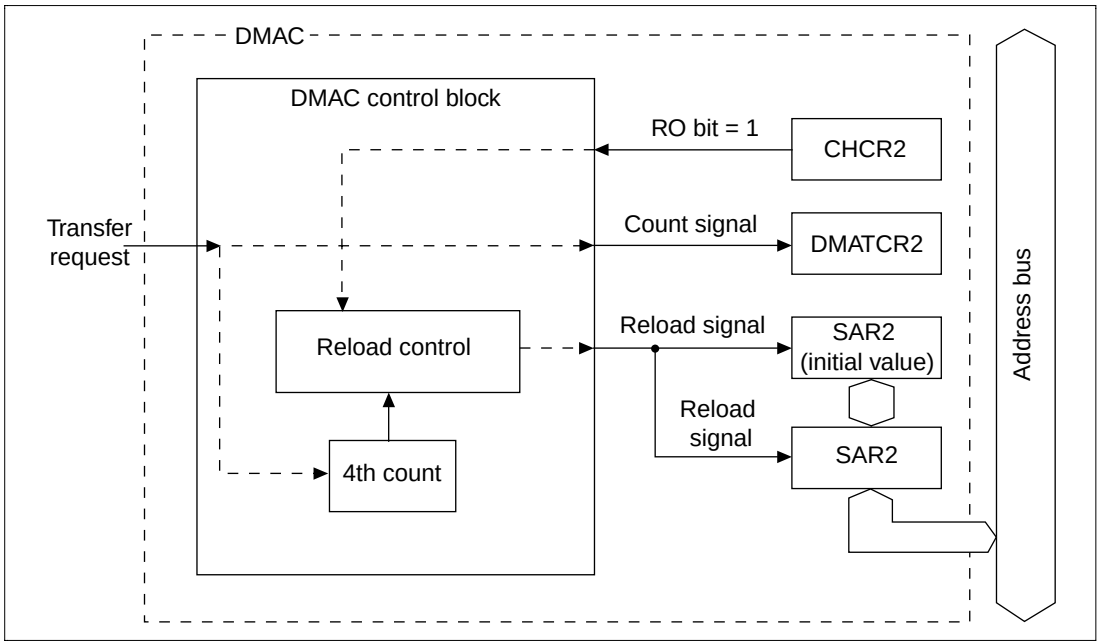


Figure 11.25 Source Address Reload Function

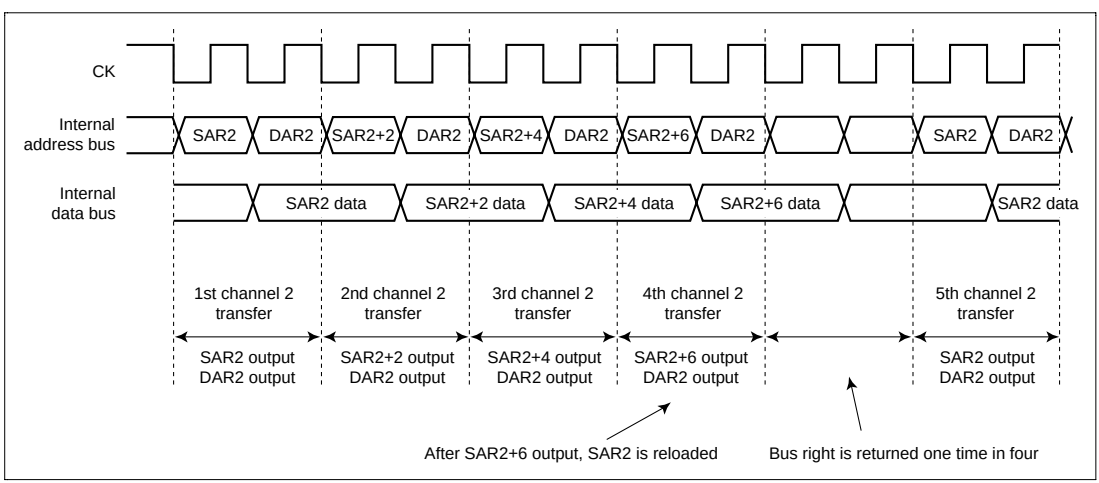


Figure 11.26 Source Address Reload Function Timing Chart

The reload function can be executed whether the transfer data size is 8, 16, or 32 bits.

DMATCR2, which specifies the number of transfers, is decremented by 1 at the end of every single-transfer-unit transfer, regardless of whether the reload function is on or off. Therefore, when using the reload function in the on state, a multiple of 4 must be specified in DMATCR2. Operation will not be guaranteed if any other value is set. Also, the counter which counts the occurrence of four transfers for address reloading is reset by clearing of the DME bit in DMAOR or the DE bit in CHCR2, setting of the transfer end flag (the TE bit in CHCR2), NMI input, and setting of the AE flag (address error generation in DMAC transfer), as well as by a reset and in software standby mode, but SAR2, DAR2, DMATCR2, and other registers are not reset. Consequently, when one of these sources occurs, there is a mixture of initialized counters and uninitialized registers in the DMAC, and incorrect operation may result if a restart is executed in this state. Therefore, when one of the above sources, other than TE setting, occurs during use of the address reload function, SAR, DAR2, and DMATCR2 settings must be carried out before re-execution.

11.3.12 DMA Transfer Ending Conditions

The DMA transfer ending conditions vary for individual channels ending and for all channels ending together.

Individual Channel Ending Conditions: There are two ending conditions. A transfer ends when the value of the channel’s DMA transfer count register (DMATCR) is 0, or when the DE bit of the channel’s CHCR is cleared to 0.

- When DMATCR is 0: When the DMATCR value becomes 0 and the corresponding channel's DMA transfer ends, the transfer end flag bit (TE) is set in the CHCR. If the IE (interrupt enable) bit has been set, a DMAC interrupt (DEI) is requested of the CPU.
- When DE of CHCR is 0: Software can halt a DMA transfer by clearing the DE bit in the channel’s CHCR. The TE bit is not set when this happens.

Conditions for Ending All Channels Simultaneously: Transfers on all channels end when the NMIF (NMI flag) bit or AE (address error flag) bit is set to 1 in the DMAOR, or when the DME bit in the DMAOR is cleared to 0.

- When the NMIF or AE bit is set to 1 in DMAOR: When an NMI interrupt or DMAC address error occurs, the NMIF or AE bit is set to 1 in the DMAOR and all channels stop their transfers. The DMAC obtains the bus rights, and if these flags are set to 1 during execution of a transfer, DMAC halts operation when the transfer processing currently being executed ends, and transfers the bus right to the other bus master. Consequently, even if the NMIF or AE bits are set to 1 during a transfer, the DMA source address register (SAR), designation address register (DAR), and transfer count register (TCR) are all updated. The TE bit is not set. To resume the transfers after NMI interrupt or address error processing, clear the appropriate flag bit to 0. To avoid restarting a transfer on a particular channel, clear its DE bit to 0.

When the processing of a one unit transfer is complete. In a dual address mode direct address transfer, even if an address error occurs or the NMI flag is set during read processing, the transfer will not be halted until after completion of the following write processing. In such a case, SAR, DAR, and TCR values are updated. In the same manner, the transfer is not halted in dual address mode indirect address transfers until after the final write processing has ended.

- When DME is cleared to 0 in DMAOR: Clearing the DME bit to 0 in the DMAOR aborts the transfers on all channels. The TE bit is not set.

11.3.13 DMAC Access from CPU

The space addressed by the DMAC is 3-cycle space. Therefore, when the CPU becomes the bus master and accesses the DMAC, a minimum of three basic clock (CLK) cycles are required for one bus cycle. Also, since the DMAC is located in word space, while a word-size access to the DMAC is completed in one bus cycle, a longword-size access is automatically divided into two word accesses, requiring two bus cycles (six basic clock cycles). These two bus cycles are executed consecutively; a different bus cycle is never inserted between the two word accesses. This applies to both write accesses and read accesses.

11.4 Examples of Use

11.4.1 Example of DMA Transfer between On-Chip SCI and External Memory

In this example, on-chip serial communication interface channel 0 (SCI0) received data is transferred to external memory using the DMAC channel 3.

Table 11.7 indicates the transfer conditions and the setting values of each of the registers.

Table 11.7 Transfer Conditions and Register Set Values for Transfer between On-Chip SCI and External Memory

Transfer Conditions	Register	Value
Transfer source: RDR0 of on-chip SCI0	SAR3	H'FFFF81A5
Transfer destination: external memory	DAR3	H'00400000
Transfer count: 64 times	DMATCR3	H'00000040
Transfer source address: fixed	CHCR3	H'00004D05
Transfer destination address: incremented		
Transfer request source: SCI0 (RDR0)		
Bus mode: cycle steal		
Transfer unit: byte		
Interrupt request generation at end of transfer		
Channel priority ranking: 0 > 1 > 2 > 3	DMAOR	H'0001

11.4.2 Example of DMA Transfer between External RAM and External Device with DACK

In this example, an external request, serial address mode transfer with external memory as the transfer source and an external device with DACK as the transfer destination is executed using DMAC channel 1.

Table 11.8 indicates the transfer conditions and the setting values of each of the registers.

Table 11.8 Transfer Conditions and Register Set Values for Transfer between External RAM and External Device with DACK

Transfer Conditions	Register	Value
Transfer source: external RAM	SAR1	H'00400000
Transfer destination: external device with DACK	DAR1	(access by DACK)
Transfer count: 32 times	DMATCR1	H'00000020
Transfer source address: decremented	CHCR1	H'00002269
Transfer destination address: (setting ineffective)		
Transfer request source: external pin ($\overline{\text{DREQ1}}$) edge detection		
Bus mode: burst		
Transfer unit: word		
No interrupt request generation at end of transfer		
Channel priority ranking: 2 > 0 > 1 > 3	DMAOR	H'0201

11.4.3 Example of DMA Transfer between A/D Converter and On-Chip Memory (Address Reload On) (Excluding A Mask)

In this example, the on-chip A/D converter channel 0 is the transfer source and on-chip memory is the transfer destination, and the address reload function is on.

Table 11.9 indicates the transfer conditions and the setting values of each of the registers.

Table 11.9 Transfer Conditions and Register Set Values for Transfer between A/D Converter and On-Chip Memory

Transfer Conditions	Register	Value
Transfer source: on-chip A/D converter ch0	SAR2	H'FFFF83F0
Transfer destination: on-chip memory	DAR2	H'FFFFF000
Transfer count: 128 times (reload count 32 times)	DMATCR2	H'00000080
Transfer source address: incremented	CHCR2	H'00085B25
Transfer destination address: incremented		
Transfer request source: A/D converter		
Bus mode: burst		
Transfer unit: byte		
Interrupt request generation at end of transfer		
Channel priority ranking: 0 > 2 > 3 > 1	DMAOR	H'0101

When address reload is on, the SAR value returns to its initially established value every four transfers. In the above example, when a transfer request is input from the A/D converter, the byte size data is first read in from the H'FFFF83F0 register of AD0 and that data is written to the on-chip memory address H'FFFFF001. Because a byte size transfer was performed, the SAR and DAR values at this point are H'FFFF83F1 and H'FFFFF001, respectively. Also, because this is a burst transfer, the bus rights remain secured, so continuous data transfer is possible.

When four transfers are completed, if the address reload is off, execution continues with the fifth and sixth transfers and the SAR value continues to increment from H'FFFF83F3 to H'FFFF83F4 to H'FFFF83F5 and so on. However, when the address reload is on, the DMAC transfer is halted upon completion of the fourth one and the bus right request signal to the CPU is cleared. At this time, the value stored in SAR is not H'FFFF83F3–H'FFFF83F4, but H'FFFF83F3–H'FFFF83F0, a return to the initially established address. The DAR value always continues to be decremented regardless of whether the address reload is on or off.

The DMAC internal status, due to the above operation after completion of the fourth transfer, is indicated in table 11.10 for both address reload on and off.

Table 11.10 DMAC Internal Status

Item	Address Reload On	Address Reload Off
SAR	H'FFFF83F0	H'FFFF83F4
DAR	H'FFFFF004	H'FFFFF004
DMATCR	H'0000007C	H'0000007C
Bus rights	Released	Maintained
DMAC operation	Halted	Processing continues
Interrupts	Not issued	Not issued
Transfer request source flag clear	Executed	Not executed

Notes: 1. Interrupts are executed until the DMATCR value becomes 0, and if the IE bit of the CHCR is set to 1, are issued regardless of whether the address reload is on or off.
2. If transfer request source flag clears are executed until the DMATCR value becomes 0, they are executed regardless of whether the address reload is on or off.
3. Designate burst mode when using the address reload function. There are cases where abnormal operation will result if it is executed in cycle steal mode.
4. Designate a multiple of four for the TCR value when using the address reload function. There are cases where abnormal operation will result if anything else is designated.

To execute transfers after the fifth one when the address reload is on, make the transfer request source issue another transfer request signal.

11.4.4 Example of DMA Transfer between A/D Converter and Internal Memory (Address Reload On) (A Mask)

In this example the on-chip A/D converter (A/D1) is the transfer source and the internal memory is the transfer destination, and the address reload on.

Table 11.11 indicates the transfer conditions and the setting values of each of the registers.

Table 11.11 Transfer Conditions and Register Set Values for Transfer between A/D Converter (A/D1) and Internal Memory

Transfer Conditions	Register	Value
Transfer source: on-chip A/D converter (A/D1)	SAR2	H'FFFF8408
Transfer destination: internal memory	DAR2	H'FFFFF000
Transfer count: 128 times (reload count 32 times)	DMATCR2	H'00000080
Transfer source address: incremented	CHCR2	H'00085B25
Transfer target address: incremented		
Transfer request source: A/D converter (A/D1)		
Bus mode: burst		
Transfer unit: byte		
Interrupt request generated at end of transfer		
Channel priority sequence: 0>2>3>1	DMAOR	H'0101

When address reload is on, the SAR value returns to its initially established value every four transfers. In the above example, when a transfer request is input from the A/D converter (A/D1), the byte size data is first read in from the H'FFFF8408 register and that data is written to the on-chip memory address H'FFFFF001. Because a byte size transfer was performed, the SAR and DAR values at this point are H'FFFF8409 and H'FFFFF001, respectively. Also, because this is a burst transfer, the bus rights remain secured, so continuous data transfer is possible.

When four transfers are completed, if the address reload is off, execution continues with the fifth and sixth transfers and the SAR value continues to increment from H'FFFF840B to H'FFFF840C to H'FFFF840D and so on. However, when the address reload is on, the DMAC transfer is halted upon completion of the fourth transfer and the bus right request signal to the CPU is cleared. At this time, the values stored in SAR are not H'FFFF840B–H'FFFF840C, but H'FFFF840B–H'FFFF8408, a return to the initially established address. The DAR value always continues to be decremented regardless of whether the address reload is on or off.

The DMAC internal status, due to the above operation after completion of the fourth transfer, is indicated in table 11.12 for both address reload on and off.

Table 11.12 DMAC Internal Status

Item	Address Reload On	Address Reload Off
SAR	H'FFFF8408	H'FFFF840C
DAR	H'FFFFF004	H'FFFFF004
DMATCR	H'0000007C	H'0000007C
Bus rights	Released	Maintained
DMAC operation	Halted	Processing continues
Interrupts	Not issued	Not issued
Transfer request source flag clear	Executed	Not executed

Notes: 1. Interrupts are executed until the DMATCR value becomes 0, and if the IE bit of the CHCR is set to 1, are issued regardless of whether the address reload is on or off.
2. If transfer request source flag clears are executed until the DMATCR value becomes 0, they are executed regardless of whether the address reload is on or off.
3. Designate burst mode when using the address reload function. There are cases where abnormal operation will result if it is executed in cycle steal mode.
4. Designate a multiple of four for the TCR value when using the address reload function. There are cases where abnormal operation will result if anything else is designated.

To execute more than four transfers with the address reload on, make the transfer request source issue another transfer request signal.

11.4.5 Example of DMA Transfer between External Memory and SCI1 Send Side (Indirect Address On)

In this example, DMAC channel 3 is used, an indirect address designated external memory is the transfer source and the SCI1 sending side is the transfer destination.

Table 11.13 indicates the transfer conditions and the setting values of each of the registers.

Table 11.13 Transfer Conditions and Register Set Values for Transfer between External Memory and SCI1 Sending Side

Transfer Conditions	Register	Value
Transfer source: external memory	SAR3	H'00400000
Value stored in address H'00400000	—	H'00450000
Value stored in address H'00450000	—	H'55
Transfer destination: on-chip SCI TDR1	DAR3	H'FFFF81B3
Transfer count: 10 times	DMATCR3	H'0000000A
Transfer source address: incremented	CHCR3	H'00011E01
Transfer destination address: fixed		
Transfer request source: SCI1 (TDR1)		
Bus mode: cycle steal		
Transfer unit: byte		
Interrupt request not generated at end of transfer		
Channel priority ranking: 0 > 1 > 2 > 3	DMAOR	H'0001

When indirect address mode is on, the data stored in the address established in SAR is not used as the transfer source data. In the case of indirect addressing, the value stored in the SAR address is read, then that value is used as the address and the data read from that address is used as the transfer source data, then that data is stored in the address designated by the DAR.

In the table 11.13 example, when a transfer request from the TDR1 of SCI1 is generated, a read of the address located at H'00400000, which is the value set in SAR3, is performed first. The data H'00450000 is stored at this H'00400000 address, and the DMAC first reads this H'00450000 value. It then uses this read value of H'00450000 as an address and reads the value of H'55 that is stored in the H'00450000 address. It then writes the value H'55 to the address H'FFFF81B3 designated by DAR3 to complete one indirect address transfer.

With indirect addressing, the first executed data read from the address established in SAR3 always results in a longword size transfer regardless of the TS0, TS1 bit designations for transfer data size. However, the transfer source address fixed and increment or decrement designations are as according to the SM0, SM1 bits. Consequently, despite the fact that the transfer data size designation is byte in this example, the SAR3 value at the end of one transfer is H'00400004. The write operation is exactly the same as an ordinary dual address transfer write operation.

11.5 Cautions on Use

1. Other than the DMA operation register (DMAOR) accessing in word (16-bit) units, access all registers in word (16-bit) or longword (32-bit) units.
2. When rewriting the RS0–RS3 bits of CHCR0–CHCR3, first clear the DE bit to 0 (set the DE bit to 0 before doing rewrites with a CHCR byte address).
3. When an NMI interrupt is input, the NMIF bit of the DMAOR is set even when the DMAC is not operating.
4. Set the DME bit of the DMAOR to 0 and make certain that any DMAC received transfer request processing has been completed before entering standby mode.
5. Do not access the DMAC, DTC, BSC, or UBC on-chip peripheral modules from the DMAC.
6. When activating the DMAC, do the CHCR or DMAOR setting as the final step. There are instances where abnormal operation will result if any other registers are established last.
7. After the DMATCR count becomes 0 and the DMA transfer ends normally, always write a 0 to the DMATCR, even when executing the maximum number of transfers on the same channel. There are instances where abnormal operation will result if this is not done.
8. Designate burst mode as the transfer mode when using the address reload function. There are instances where abnormal operation will result in cycle steal mode.
9. Designate a multiple of four for the DMATCR value when using the address reload function. There are instances where abnormal operation will result if anything else is designated.
10. When detecting external requests by falling edge, maintain the external request pin at high level when performing the DMAC establishment.
11. When operating in single address mode, establish an external address as the address. There are instances where abnormal operation will result if an internal address is established.
12. Do not access DMAC register empty addresses (H'FFFF86B2–H'FFFF86BF). Operation cannot be guaranteed when empty addresses are accessed.

Section 12 Multifunction Timer Pulse Unit (MTU)

12.1 Overview

The SuperH microprocessor has an on-chip 16-bit multifunction timer pulse unit (MTU) with five channels of 16-bit timers.

12.1.1 Features

- Can process a maximum of sixteen different pulse outputs and inputs.
- Has sixteen timer general registers (TGR): four each for channels 0, 3, and 4, and two each for channels 1 and 2 that can be set to function independently as output compare or input capture. The channel 0, 3, and 4 TGRC and TGRD registers can be used as buffer registers.
- Can select eight counter input clock sources for all channels
- All channels can be set for the following operating modes:
  - Compare match waveform output: 0 output/1 output/toggle output selectable.
  - Input capture function: Selectable rising edge, falling edge, or both rising and falling edge detection.
  - Counter clearing function: Counters can be cleared by a compare-match or input capture.
  - Synchronizing mode: Two or more timer counters (TCNT) can be written to simultaneously. Two or more timer counters can be simultaneously cleared by a compare-match or input capture. Counter synchronization functions enable synchronized register input/output.
  - PWM mode: PWM output can be provided with any duty cycle. When combined with the counter synchronizing function, enables up to twelve-phase PWM output. (With channels 0–2 set to PWM mode 2, channels 3–4, and channels 0–4 synchronized with TGR3A of channel 3 as the sync register (channels 0–4 phase output: 4, 2, 2, 2, 2).)
- Channels 0, 3, and 4 can be set for buffer operation
  - Input capture register double buffer configuration possible
  - Output compare register automatic re-write possible
- Channels 1 and 2 can be independently set to the phase counting mode
  - Two-phase encoder pulse up/down count possible
- Cascade connection operation
  - Can be operated as a 32-bit counter by using the channel 2 input clock for channel 1 overflow/underflow
- Channels 3 and 4 can be set in the following modes:
  - Reset-synchronized PWM mode: By combining channels 3 and 4, a sawtooth wave comparator type six-phase PWM waveform can be output.

— Complementary PWM mode: By combining channels 3 and 4, a triangle wave comparator type six-phase PWM output is possible with non-overlapping times.

- High speed access via internal 16-bit bus
- Twenty-three interrupt sources
  — Channels 0, 3, and 4 have four compare-match/input capture interrupts and one overflow interrupt which can be requested independently.
  — Channels 1 and 2 have two compare-match/input capture interrupts, one overflow interrupt, and one underflow interrupt which can be requested independently.
- Automatic transfer of register data

  Block transfer, 1-word data transfers and 1-byte data transfers are possible through DTC or DMAC activation.
- A/D converter conversion start trigger can be generated
  — Channels 0–4 compare-match/input capture signals can be used as A/D converter conversion start triggers.

Table 12.1 summarizes the MTU functions.

Table 12.1 MTU Functions

Item		Channel 0	Channel 1	Channel 2	Channel 3	Channel 4
Counter clocks		Internal: φ/1, φ/4, φ/16, φ/64, φ/256, φ/1024 External: Eight to each channel from TCLKA, TCLKB, TCLKC, and TCLKD				
General registers		TGR0A	TGR1A	TGR2A	TGR3A	TGR4A
		TGR0B	TGR1B	TGR2B	TGR3B	TGR4B
General registers/buffer registers		TGR0C TGR0D	No	No	TGR3C TGR3D	TGR4C TGR4D
Input/output pins		TIOC0A	TIOC1A	TIOC2A	TIOC3A	TIOC4A
		TIOC0B	TIOC1B	TIOC2B	TIOC3B	TIOC4B
		TIOC0C			TIOC3C	TIOC4C
		TIOC0D			TIOC3D	TIOC4D
Counter clear function		TGR compare-match or input capture	TGR compare-match or input capture	TGR compare-match or input capture	TGR compare-match or input capture	TGR compare-match or input capture
Compare match output	0	Yes	Yes	Yes	Yes	Yes
	1	Yes	Yes	Yes	Yes	Yes
	Toggle	Yes	Yes	Yes	Yes	Yes
Input capture function		Yes	Yes	Yes	Yes	Yes
Synchronization		Yes	Yes	Yes	Yes	Yes
Buffer operation		Yes	No	No	Yes	No
PWM mode 1		Yes	Yes	Yes	Yes	Yes
PWM mode 2		Yes	Yes	Yes	No	No
Phase counting mode		No	Yes	Yes	No	No
Reset-synchronized PWM mode		No	No	No	Yes	Yes
Complementary PWM mode		No	No	No	Yes	Yes
DMAC activation		TGR0A compare match or input capture	TGR1A compare match or input capture	TGR2A compare match or input capture	TGR3A compare match or input capture	TGR4A compare match or input capture

Table 12.1 MTU Functions (cont)

Item	Channel 0	Channel 1	Channel 2	Channel 3	Channel 4
Hard DTC activation	TGR compare-match or input capture	TGR compare-match or input capture	TGR compare-match or input capture	TGR compare-match or input capture	TGR compare-match or input capture and TCNT4 overflow/ underflow
A/D conversion start trigger	TGR0A compare match or input capture	TGR1A compare match or input capture	TGR2A compare match or input capture	TGR3A compare match or input capture	TGR4A compare match or input capture
Interrupt sources	Compare match/input capture 0A	Compare match/input capture 1A	Compare match/input capture 2A	Compare match/input capture 3A	Compare match/input capture 4A
	Compare match/input capture 0B	Compare match/input capture 1B	Compare match/input capture 2B	Compare match/input capture 3B	Compare match/input capture 4B
	Compare match/input capture 0C	Overflow	Overflow	Compare match/input capture 3C	Compare match/input capture 4C
	Compare match/input capture 0D	Underflow	Underflow	Compare match/input capture 3D	Compare match/input capture 4D
	Overflow	—	—	Overflow	Overflow/ underflow

12.1.2 Block Diagram

Figure 12.1 is the block diagram of the MTU.

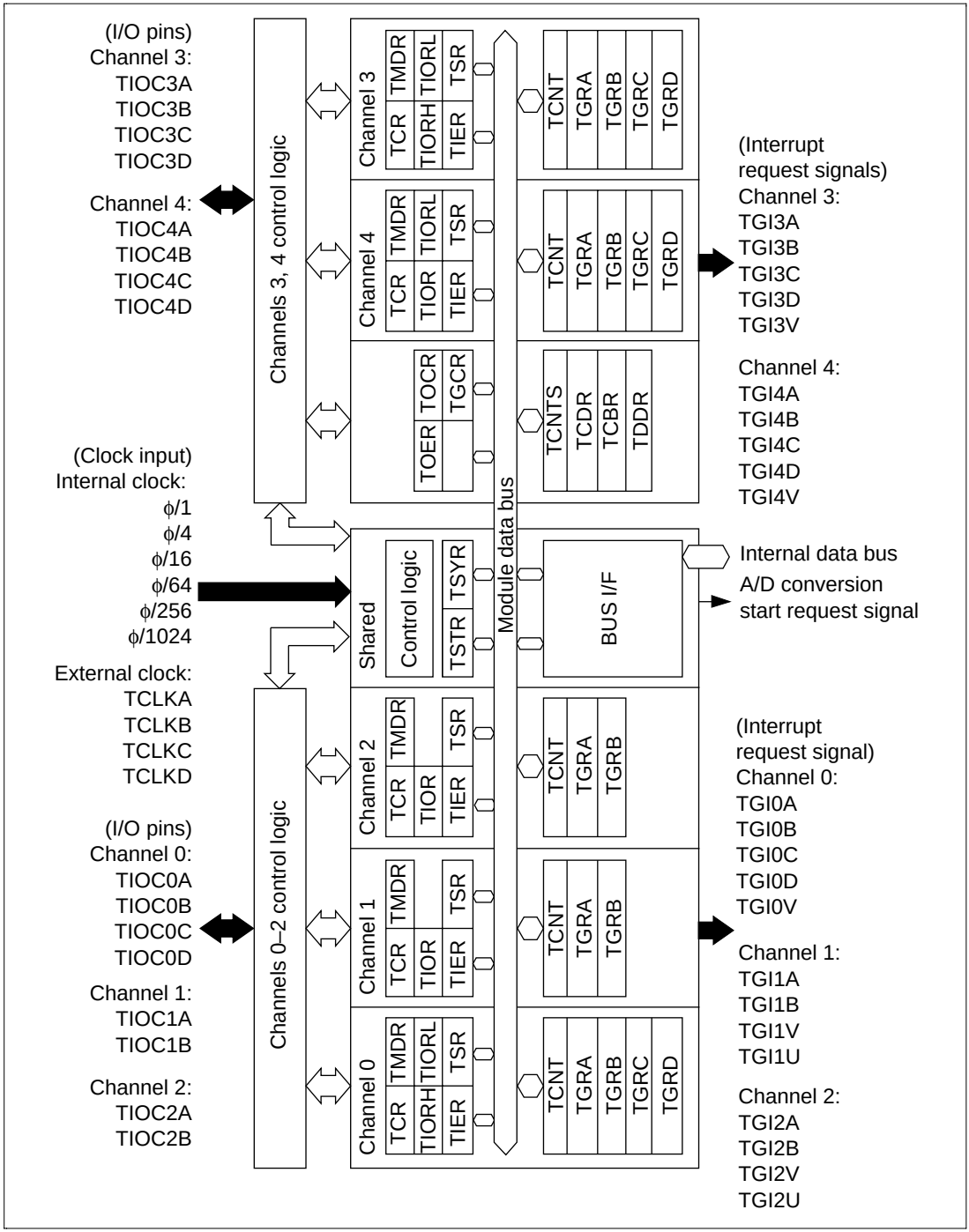


Figure 12.1 MTU Block Diagram

12.1.3 Pin Configuration

Table 12.2 summarizes the MTU pins.

Table 12.2 Pin Configuration

Channel	Name	Pin Name	I/O	Function
Shared	Clock input A	TCLKA	I	Clock A input pin (A-phase input pin in channel 1 phase counting mode)
	Clock input B	TCLKB	I	Clock B input pin (B-phase input pin in channel 1 phase counting mode)
	Clock input C	TCLKC	I	Clock C input pin (A-phase input pin in channel 2 phase counting mode)
	Clock input D	TCLKD	I	Clock D input pin (B-phase input pin in channel 2 phase counting mode)
0	Input capture/output compare-match 0A	TIOC0A	I/O	TGR0A input capture input/output compare output/PWM output pin
	Input capture/output compare-match 0B	TIOC0B	I/O	TGR0B input capture input/output compare output/PWM output pin
	Input capture/output compare-match 0C	TIOC0C	I/O	TGR0C input capture input/output compare output/PWM output pin
	Input capture/output compare-match 0D	TIOC0D	I/O	TGR0D input capture input/output compare output/PWM output pin
1	Input capture/output compare-match 1A	TIOC1A	I/O	TGR1A input capture input/output compare output/PWM output pin
	Input capture/output compare-match 1B	TIOC1B	I/O	TGR1B input capture input/output compare output/PWM output pin
2	Input capture/output compare-match 2A	TIOC2A	I/O	TGR2A input capture input/output compare output/PWM output pin
	Input capture/output compare-match 2B	TIOC2B	I/O	TGR2B input capture input/output compare output/PWM output pin

Table 12.2 Pin Configuration (cont)

Channel	Name	Pin Name	I/O	Function
3	Input capture/output compare-match 3A	TIOC3A	I/O	TGR3A input capture input/output compare output/PWM output pin In complementary PWM/reset synchronous PWM mode, 1/2 PWM period toggle output pin
	Input capture/output compare-match 3B	TIOC3B	I/O	TGR3B input capture input/output compare output pin In complementary PWM/reset synchronous PWM mode, PWM output/U phase output pin
	Input capture/output compare-match 3C	TIOC3C	I/O	TGR3C input capture input/output compare output/PWM output pin In complementary PWM/reset synchronous PWM mode
	Input capture/output compare-match 3D	TIOC3D	I/O	TGR3D input capture input/output compare output pin In complementary PWM/reset synchronous PWM mode, PWM output/U phase output pin
4	Input capture/output compare-match 4A	TIOC4A	I/O	TGR4A input capture input/output compare output/PWM output pin In complementary PWM/reset synchronous PWM mode, PWM output/V phase output pin
	Input capture/output compare-match 4B	TIOC4B	I/O	TGR4B input capture input/output compare output pin In complementary PWM/reset synchronous PWM mode, PWM output/W phase output pin
	Input capture/output compare-match 4C	TIOC4C	I/O	TGR4C input capture input/output compare output/PWM output pin In complementary PWM/reset synchronous PWM mode, PWM output/V phase output pin
	Input capture/output compare-match 4D	TIOC4D	I/O	TGR4D input capture input/output compare output pin In complementary PWM/reset synchronous PWM mode, PWM output/W phase output pin

Note: The TIOC pins output undefined values when they are set to input capture and timer output by the pin function controller (PFC).

12.1.4 Register Configuration

Table 12.3 summarizes the MTU register configuration.

Table 12.3 Register Configuration

Channel	Name	Abbreviation	R/W	Initial Value	Address	Access Size (Bits) *1
Shared	Timer start register	TSTR	R/W	H'00	H'FFFF8240	8, 16, 32
	Timer synchro register	TSYR	R/W	H'00	H'FFFF8241	
0	Timer control register 0	TCR0	R/W	H'00	H'FFFF8260	
	Timer mode register 0	TMDR0	R/W	H'C0	H'FFFF8261	
	Timer I/O control register 0H	TIOR0H	R/W	H'00	H'FFFF8262	
	Timer I/O control register 0L	TIOR0L	R/W	H'00	H'FFFF8263	
	Timer interrupt enable register 0	TIER0	R/W	H'40	H'FFFF8264	
	Timer status register 0	TSR0	R/(W)*2	H'C0	H'FFFF8265	
	Timer counter 0	TCNT0	R/W	H'0000	H'FFFF8266	16, 32
	General register 0A	TGR0A	R/W	H'FFFF	H'FFFF8268	
	General register 0B	TGR0B	R/W	H'FFFF	H'FFFF826A	
	General register 0C	TGR0C	R/W	H'FFFF	H'FFFF826C	
	General register 0D	TGR0D	R/W	H'FFFF	H'FFFF826E	
1	Timer control register 1	TCR1	R/W	H'00	H'FFFF8280	8, 16, 32
	Timer mode register 1	TMDR1	R/W	H'C0	H'FFFF8281	
	Timer I/O control register 1	TIOR1	R/W	H'00	H'FFFF8282	
	Timer interrupt enable register 1	TIER1	R/W	H'40	H'FFFF8284	
	Timer status register 1	TSR1	R/(W)*2	H'C0	H'FFFF8285	
	Timer counter 1	TCNT1	R/W	H'0000	H'FFFF8286	16, 32
	General register 1A	TGR1A	R/W	H'FFFF	H'FFFF8288	
	General register 1B	TGR1B	R/W	H'FFFF	H'FFFF828A	

Table 12.3 Register Configuration (cont)

Channel	Name	Abbreviation	R/W	Initial Value	Address	Access Size (Bits)*1
2	Timer control register 2	TCR2	R/W	H'00	H'FFFF82A0	8, 16, 32
	Timer mode register 2	TMDR2	R/W	H'C0	H'FFFF82A1	
	Timer I/O control register 2	TIOR2	R/W	H'00	H'FFFF82A2	
	Timer interrupt enable register 2	TIER2	R/W	H'40	H'FFFF82A4	
	Timer status register 2	TSR2	R/(W)*2	H'C0	H'FFFF82A5	
	Timer counter 2	TCNT2	R/W	H'0000	H'FFFF82A6	16, 32
	General register 2A	TGR2A	R/W	H'FFFF	H'FFFF82A8	
	General register 2B	TGR2B	R/W	H'FFFF	H'FFFF82AA	
3	Timer control register 3	TCR3	R/W*3	H'00	H'FFFF8200	8, 16, 32
	Timer mode register 3	TMDR3	R/W*3	H'C0	H'FFFF8202	
	Timer I/O control register 3H	TIOR3H	R/W*3	H'00	H'FFFF8204	
	Timer I/O control register 3L	TIOR3L	R/W*3	H'00	H'FFFF8205	
	Timer interrupt enable register 3	TIER3	R/W*3	H'40	H'FFFF8208	
	Timer status register 3	TSR3	R/(W)*2	H'C0	H'FFFF822C	8, 16, 32
	Timer counter 3	TCNT3	R/W*3	H'0000	H'FFFF8210	16, 32
	General register 3A	TGR3A	R/W*3	H'FFFF	H'FFFF8218	
	General register 3B	TGR3B	R/W*3	H'FFFF	H'FFFF821A	
	General register 3C	TGR3C	R/W	H'FFFF	H'FFFF8224	16, 32
	General register 3D	TGR3D	R/W	H'FFFF	H'FFFF8226	
4	Timer control register 4	TCR4	R/W*3	H'00	H'FFFF8201	8, 16, 32
	Timer mode register 4	TMDR4	R/W*3	H'C0	H'FFFF8203	
	Timer I/O control register 4H	TIOR4H	R/W*3	H'00	H'FFFF8206	
	Timer I/O control register 4L	TIOR4L	R/W*3	H'00	H'FFFF8207	
	Timer interrupt enable register 4	TIER4	R/W*3	H'40	H'FFFF8209	
	Timer status register 4	TSR4	R/(W)*2	H'C0	H'FFFF822D	8, 16, 32

Table 12.3 Register Configuration (cont)

Channel	Name	Abbreviation	R/W	Initial Value	Address	Access Size (Bits) *1
4 (cont)	Timer counter 4	TCNT4	R/W*3	H'0000	H'FFFF8212	16, 32
	General register 4A	TGR4A	R/W*3	H'FFFF	H'FFFF821C	
	General register 4B	TGR4B	R/W*3	H'FFFF	H'FFFF821E	
	General register 4C	TGR4C	R/W	H'FFFF	H'FFFF8228	16, 32
	General register 4D	TGR4D	R/W	H'FFFF	H'FFFF822A	
3 and 4	Timer output master enable register	TOER	R/W*3	H'C0	H'FFFF820A	8, 16, 32
	Timer output control register	TOCR	R/W*3	H'00	H'FFFF820B	
	Timer gate control register	TGCR	R/W*3	H'80	H'FFFF820D	
	Timer cycle data register	TCDR	R/W*3	H'FFFF	H'FFFF8214	16, 32
	Timer dead time data register	TDDR	R/W*3	H'FFFF	H'FFFF8216	
	Timer subcounter	TCNTS	R	H'0000	H'FFFF8220	16, 32
	Timer cycle buffer register	TCBR	R/W	H'FFFF	H'FFFF8222	

Notes: *1 16-bit registers (TCNT, TGR) cannot be read or written in 8-bit units.

*2 Write 0 to clear flags.

*3 If the MTURWE bit of bus control register 1 (BCR) in the bus state controller (BSC) is 0 cleared, access becomes impossible (undefined read/write disabled).

12.2 MTU Register Descriptions

12.2.1 Timer Control Register (TCR)

The TCR is an 8-bit read/write register for controlling the TCNT counter for each channel. The MTU has five TCR registers, one for each of the channels 0 to 4. TCR is initialized to H'00 by a power-on reset or the standby mode. Manual reset does not initialize TCR.

Channels 0, 3, 4: TCR0, TCR3, TCR4:

Bit:	7	6	5	4	3	2	1	0
	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Channels 1, 2: TCR1, TCR2:

Bit:	7	6	5	4	3	2	1	0
	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- Bits 7–5—Counter Clear 2, 1, 0 (CCLR2, CCLR1, CCLR0): Select the counter clear source for the TCNT counter.

Channels 0, 3, 4:

Bit 7: CCLR2	Bit 6: CCLR1	Bit 5: CCLR0	Description
0	0	0	TCNT clear disabled (initial value)
		1	TCNT is cleared by TGRA compare-match or input capture
	1	0	TCNT is cleared by TGRB compare-match or input capture
		1	Synchronizing clear: TCNT is cleared in synchronization with clear of other channel counters operating in sync.*1
1	0	0	TCNT clear disabled
		1	TCNT is cleared by TGRC compare-match or input capture*2
	1	0	TCNT is cleared by TGRD compare-match or input capture*2
		1	Synchronizing clear: TCNT is cleared in synchronization with clear of other channel counters operating in sync*1

Notes: *1 Setting the SYNC bit of the TSYR to 1 sets the synchronization.

*2 When TGRC or TGRD are functioning as buffer registers, TCNT is not cleared because the buffer registers have priority and compare-match/input captures do not occur.

Channels 1, 2:

Bit 7: Reserved*1	Bit 6: CCLR1	Bit 5: CCLR0	Description
0	0	0	TCNT clear disabled (initial value)
		1	TCNT is cleared by TGRA compare-match or input capture
	1	0	TCNT is cleared by TGRB compare-match or input capture
		1	Synchronizing clear: TCNT is cleared in synchronization with clear of other channel counters operating in sync*2

Notes: *1 The bit 7 of channels 1 and 2 is reserved. It always reads 0, and cannot be modified.

*2 Setting the SYNC bit of the TSYR to 1 sets the synchronization.

- Bits 4–3—Clock Edge 1, 0 (CKEG1 and CKEG0): CKEG1 and CKEG0 select the input clock edges. When counting is done on both edges of the internal clock the input clock frequency becomes 1/2 (Example: both edges of $\phi/4$ = rising edge of $\phi/2$). When phase count mode is used with channels 1, 2, these settings are ignored, as the phase count mode settings have priority.

Bit 4: CKEG1	Bit 3: CKEG0	Description
0	0	Count on rising edges (initial value)
	1	Count on falling edges
1	X	Count on both rising and falling edges

Notes: 1. X: 0 or 1, don't care.
2. Internal clock edge selection is effective when the input clock is $\phi/4$ or slower. When ø/1 or the overflow/underflow of another channel is selected for the input clock, although values can be written, counter operation complies with the initial value (count on rising edges).

- Bits 2–0—Timer Prescaler 2–0 (TPSC2–TPSC0): TPSC2–TPSC0 select the counter clock source for the TCNT. An independent clock source can be selected for each channel. Table 12.4 shows the possible settings for each channel.

Table 12.4 MTU Clock Sources

	Internal Clock						Other Channel	External Clock			
Channel	ϕ/1	ϕ/4	ϕ/16	ϕ/64	ϕ/256	ϕ/1024	Overflow/ Underflow	TCLKA	TCLKB	TCLKC	TCLKD
0	O	O	O	O	X	X	X	O	O	O	O
1	O	O	O	O	O	X	O	O	O	X	X
2	O	O	O	O	X	O	X	O	O	O	X
3	O	O	O	O	O	O	X	O	O	X	X
4	O	O	O	O	O	O	X	O	O	X	X

Note: Symbols: O: Setting possible X: Setting not possible

Channel 0:

Bit 2: TPSC2	Bit 1: TPSC1	Bit 0: TPSC0	Description
0	0	0	Internal clock: count with ϕ/1 (initial value)
		1	Internal clock: count with ϕ/4
	1	0	Internal clock: count with ϕ/16
		1	Internal clock: count with ϕ/64
1	0	0	External clock: count with the TCLKA pin input
		1	External clock: count with the TCLKB pin input
	1	0	External clock: count with the TCLKC pin input
		1	External clock: count with the TCLKD pin input

Channel 1:

Bit 2: TPSC2	Bit 1: TPSC1	Bit 0: TPSC0	Description
0	0	0	Internal clock: count with ϕ/1 (initial value)
		1	Internal clock: count with ϕ/4
	1	0	Internal clock: count with ϕ/16
		1	Internal clock: count with ϕ/64
1	0	0	External clock: count with the TCLKA pin input
		1	External clock: count with the TCLKB pin input
	1	0	Internal clock: count with ϕ/256
		1	Count with the TCNT2 overflow/underflow

Note: These settings are ineffective when channel 1 is in phase counting mode.

Channel 2:

Bit 2: TPSC2	Bit 1: TPSC1	Bit 0: TPSC0	Description
0	0	0	Internal clock: count with ϕ/1 (initial value)
		1	Internal clock: count with ϕ/4
	1	0	Internal clock: count with ϕ/16
		1	Internal clock: count with ϕ/64
1	0	0	External clock: count with the TCLKA pin input
		1	External clock: count with the TCLKB pin input
	1	0	External clock: count with the TCLKC pin input
		1	Internal clock: count with ϕ/1024

Note: These settings are ineffective when channel 2 is in phase counting mode.

Channel 3:

Bit 2: TPSC2	Bit 1: TPSC1	Bit 0: TPSC0	Description
0	0	0	Internal clock: count with ϕ/1 (initial value)
		1	Internal clock: count with ϕ/4
	1	0	Internal clock: count with ϕ/16
		1	Internal clock: count with ϕ/64
1	0	0	Internal clock: count with ϕ/256
		1	Internal clock: count with ϕ/1024
	1	0	External clock: count with the TCLKA pin input
		1	External clock: count with the TCLKB pin input

Channel 4:

Bit 2: TPSC2	Bit 1: TPSC1	Bit 0: TPSC0	Description
0	0	0	Internal clock: count with ϕ/1 (initial value)
		1	Internal clock: count with ϕ/4
	1	0	Internal clock: count with ϕ/16
		1	Internal clock: count with ϕ/64
1	0	0	Internal clock: count with ϕ/256
		1	Internal clock: count with ϕ/1024
	1	0	External clock: count with the TCLKA pin input
		1	External clock: count with the TCLKB pin input

12.2.2 Timer Mode Register (TMDR)

The TMDR is an 8-bit read/write register that sets the operating mode for each channel. The MTU has five TMDR registers, one for each channel. TMDR is initialized to H'C0 by a power-on reset or the standby mode. Manual reset does not initialize TMDR.

Channels 0, 3, 4: TMDR0, TMDR3, TMDR4:

Bit:	7	6	5	4	3	2	1	0
	—	—	BFB	BFA	MD3	MD2	MD1	MD0
Initial value:	1	1	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Channels 1, 2: TMDR1, TMDR2:

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	MD3	MD2	MD1	MD0
Initial value:	1	1	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W

- Bits 7, 6—Reserved: These bits are reserved. They always read as 1, and cannot be modified.
- Bit 5—Buffer Operation B (BFB): Designates whether to use the TGRB register for normal operation, or buffer operation in combination with the TGRD register. When using TGRD as a buffer register, no TGRD register input capture/output compares are generated.

  This bit is reserved in channels 1 and 2, which have no TGRD registers. It is always read as 0, and cannot be modified.

Bit 5: BFB	Description
0	TGRB operates normally (initial value)
1	TGRB and TGRD buffer operation

- Bit 4—Buffer Operation A (BFA): Designates whether to use the TGRA register for normal operation, or buffer operation in combination with the TGRC register. When using TGRC as a buffer register, no TGRC register input capture/output compares are generated.

  This bit is reserved in channels 1 and 2, which have no TGRC registers. It is always read as 0, and cannot be modified.

Bit 4: BFA	Description
0	TGRA operates normally (initial value)
1	TGRA and TGRC buffer operation

- Bits 3–0—Modes 3–0 (MD3–MD0): These bits set the timer operation mode.

Bit 3: MD3	Bit 2: MD2	Bit 1: MD1	Bit 0: MD0	Description
0	0	0	0	Normal operation (initial value)
			1	Reserved (do not set)
		1	0	PWM mode 1
			1	PWM mode 2*1
	1	0	0	Phase counting mode 1*2
			1	Phase counting mode 2*2
		1	0	Phase counting mode 3*2
			1	Phase counting mode 4*2
1	0	0	0	Reset synchronous PWM mode*3
			1	Reserved (do not set)
		1	0	Reserved (do not set)
			1	Reserved (do not set)
	1	0	0	Reserved (do not set)
			1	Complementary PWM mode 1 (transmit at peak)*3
		1	0	Complementary PWM mode 2 (transmit at valley)*3
			1	Complementary PWM mode 3 (transmit at peak and valley) *3

Notes: *1 PWM mode 2 can not be set for channels 3, 4.

*2 Phase measurement mode can not be set for channels 0, 3, 4.

*3 Reset synchronous PWM mode, complementary PWM mode can only be set for channel 3. When channel 3 is set to reset synchronous PWM mode or complementary PWM mode, the channel 4 settings become ineffective and automatically conform to the channel 3 settings. However, do not set channel 4 to reset synchronous PWM mode or complementary PWM mode. Reset synchronous PWM mode and complementary PWM mode can not be set for channels 0, 1, 2.

12.2.3 Timer I/O Control Register (TIOR)

The TIOR is a register that controls the TGR. The MTU has eight TIOR registers, two each for channels 0, 3, and 4, and one each for channels 1 and 2. TIOR is initialized to H'00 by a power-on reset or the standby mode. Manual reset does not initialize TIOR.

Channels 0, 3, 4: TIOR0H, TIOR3H, TIOR4H

Channels 1, 2: TIOR1, TIOR2:

Bit:	7	6	5	4	3	2	1	0
	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- Bits 7–4—I/O Control B3–B0 (IOB3–IOB0): These bits set the TGRB register function.
- Bits 3–0—I/O Control A3–B0 (IOA3–IOA0): These bits set the TGRA register function.

Channels 0, 3, 4: TIOR0L, TIOR3L, TIOR4L:

Bit:	7	6	5	4	3	2	1	0
	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: When the TGRC or TGRD registers are set for buffer operation, these settings become ineffective and the operation is as a buffer register.

- Bits 7–4—I/O Control D3–D0 (IOD3–IOD0): These bits set the TGRD register function.
- Bits 3–0—I/O Control C3–C0 (IOC3–IOC0): These bits set the TGRC register function.

Channel 0 (TIOR0H Register):

- Bits 7–4—I/O Control B3–B0 (IOB3–IOB0): These bits set the TGR0B register function.

Bit 7: IOB3	Bit 6: IOB2	Bit 5: IOB1	Bit 4: IOB0	Description		
0	0	0	0	TGR0B	Output disabled (initial value)	
			1	is an	Initial	Output 0 on compare-match
		1	0	output	output	Output 1 on compare-match
			1	compare	is 0	Toggle output on compare-match
	1	0	0	register	Output disabled	
			1		Initial	Output 0 on compare-match
		1	0		output	Output 1 on compare-match
			1		is 1	Toggle output on compare-match
1	0	0	0	TGR0B	Capture	Input capture on rising edge
			1	is an	input source	Input capture on falling edge
		1	0	input	is the	Input capture on both edges
			1	capture	TIOC0B pin	
	1	0	0	register	Capture	Input capture
			1		input source	on TCNT1
		1	0		is channel 1/	count up/count down
			1		count clock	

- Bits 3–0—I/O Control A3–A0 (IOA3–IOA0): These bits set the TGR0A register function.

Bit 3: IOA3	Bit 2: IOA2	Bit 1: IOA1	Bit 0: IOA0	Description		
0	0	0	0	TGR0A	Output disabled (initial value)	
			1	is an	Initial	Output 0 on compare-match
		1	0	output	output	Output 1 on compare-match
			1	compare	is 0	Toggle output on compare-match
	1	0	0	register	Output disabled	
			1		Initial	Output 0 on compare-match
		1	0		output	Output 1 on compare-match
			1		is 1	Toggle output on compare-match
1	0	0	0	TGR0A	Capture	Input capture on rising edge
			1	is an	input source	Input capture on falling edge
		1	0	input	is the	Input capture on both edges
			1	capture	TIOC0A pin	
	1	0	0	register	Capture	Input capture
			1		input source	on TCNT1
		1	0		is channel 1/	count up/count down
			1		count clock	

Channel 0 (TIOR0L Register):

- Bits 7–4—I/O Control D3–D0 (IOD3–IOD0): These bits set the TGR0D register function.

Bit 7: IOD3	Bit 6: IOD2	Bit 5: IOD1	Bit 4: IOD0	Description		
0	0	0	0	TGR0D	Output disabled (initial value)	
			1	is an	Initial	Output 0 on compare-match
		1	0	output	output	Output 1 on compare-match
			1	compare	is 0	Toggle output on compare-match
	1	0	0	register	Output disabled	
			1		Initial	Output 0 on compare-match
		1	0		output	Output 1 on compare-match
			1		is 1	Toggle output on compare-match
1	0	0	0	TGR0D	Capture	Input capture on rising edge
			1	is an	input source	Input capture on falling edge
		1	0	input	is the	Input capture on both edges
			1	capture	TIOC0D pin	
	1	0	0	register	Capture	Input capture
			1		input source	on TCNT1
		1	0		is channel 1/	count up/count down
			1		count clock	

Note: When the BFB bit of TMDR0 is set to 1 and TGR0D is being used as a buffer register, these settings become ineffective and input capture/output compares do not occur.

- Bits 3–0—I/O Control C3–C0 (IOC3–IOC0): These bits set the TGR0C register function.

Bit 3: IOC3	Bit 2: IOC2	Bit 1: IOC1	Bit 0: IOC0	Description		
0	0	0	0	TGR0C	Output disabled (initial value)	
			1	is an	Initial	Output 0 on compare-match
		1	0	output	output	Output 1 on compare-match
			1	compare	is 0	Toggle output on compare-match
	1	0	0	register	Output disabled	
			1		Initial	Output 0 on compare-match
		1	0		output	Output 1 on compare-match
			1		is 1	Toggle output on compare-match
1	0	0	0	TGR0C	Capture	Input capture on rising edge
			1	is an	input source	Input capture on falling edge
		1	0	input	is the	Input capture on both edges
			1	capture	TIOC0C pin	
	1	0	0	register	Capture	Input capture
			1		input source	on TCNT1
		1	0		is channel 1/	count up/count down
			1		count clock	

Note: When the BFA bit of TMDR0 is set to 1 and TGR0C is being used as a buffer register, these settings become ineffective and input capture/output compares do not occur.

Channel 1 (TIOR1 Register):

- Bits 7–4—I/O Control B3–B0 (IOB3–IOB0): These bits set the TGR1B register function.

Bit 7: IOB3	Bit 6: IOB2	Bit 5: IOB1	Bit 4: IOB0	Description		
0	0	0	0	TGR1B	Output disabled (initial value)	
			1	is an	Initial	Output 0 on compare-match
		1	0	output	output	Output 1 on compare-match
			1	compare	is 0	Toggle output on compare-match
	1	0	0	register	Output disabled	
			1		Initial	Output 0 on compare-match
		1	0		output	Output 1 on compare-match
			1		is 1	Toggle output on compare-match
1	0	0	0	TGR1B	Capture	Input capture on rising edge
			1	is an	input source	Input capture on falling edge
		1	0	input	is the	Input capture on both edges
			1	capture	TIOC1B pin	
	1	0	0	register	Capture input	Input capture
			1		source TGR0C	on channel TGR0C
		1	0		compare/match	compare-match/input
			1		input capture	capture generation

- Bits 3–0—I/O Control A3–A0 (IOA3–IOA0): These bits set the TGR1A register function.

Bit 3: IOA3	Bit 2: IOA2	Bit 1: IOA1	Bit 0: IOA0	Description		
0	0	0	0	TGR1A	Output disabled (initial value)	
			1	is an	Initial	Output 0 on compare-match
		1	0	output	output	Output 1 on compare-match
			1	compare	is 0	Toggle output on compare-match
	1	0	0	register	Output disabled	
			1		Initial	Output 0 on compare-match
		1	0		output	Output 1 on compare-match
			1		is 1	Toggle output on compare-match
1	0	0	0	TGR1A	Capture	Input capture on rising edge
			1	is an	input source	Input capture on falling edge
		1	0	input	is the	Input capture on both edges
			1	capture	TIOC1A pin	
	1	0	0	register	Capture input	Input capture
			1		source is TGR0A	on channel 0/TGR0A
		1	0		compare-match/input	compare-match/input capture generation
			1		capture	

Channel 2 (TIOR2 Register):

- Bits 7–4—I/O Control B3–B0 (IOB3–IOB0): These bits set the TGR2B register function.

Bit 7: IOB3	Bit 6: IOB2	Bit 5: IOB1	Bit 4: IOB0	Description		
0	0	0	0	TGR2B	Output disabled (initial value)	
			1	is an	Initial	Output 0 on compare-match
		1	0	output	output	Output 1 on compare-match
			1	compare	is 0	Toggle output on compare-match
	1	0	0	register	Output disabled	
			1		Initial	Output 0 on compare-match
		1	0		output	Output 1 on compare-match
			1		is 1	Toggle output on compare-match
1	0	0	0	TGR2B	Capture	Input capture on rising edge
			1	is an	input source	Input capture on falling edge
		1	0	input	is the	Input capture on both edges
			1	capture	TIOC2B pin	
	1	0	0	register		Input capture on rising edge
			1			Input capture on falling edge
		1	0			Input capture on both edges
			1			

- Bits 3–0—I/O Control A3–A0 (IOA3–IOA0): These bits set the TGR2A register function.

Bit 3: IOA3	Bit 2: IOA2	Bit 1: IOA1	Bit 0: IOA0	Description		
0	0	0	0	TGR2A	Output disabled (initial value)	
			1	is an	Initial	Output 0 on compare-match
		1	0	output	output	Output 1 on compare-match
			1	compare	is 0	Toggle output on compare-match
	1	0	0	register	Output disabled	
			1		Initial	Output 0 on compare-match
		1	0		output	Output 1 on compare-match
			1		is 1	Toggle output on compare-match
1	0	0	0	TGR2A	Capture	Input capture on rising edge
			1	is an	input source	Input capture on falling edge
		1	0	input	is the	Input capture on both edges
			1	capture	TIOC2A pin	
	1	0	0	register		Input capture on rising edge
			1			Input capture on falling edge
		1	0			Input capture on both edges
			1			

Channel 3 (TIOR3H Register):

- Bits 7–4—I/O Control B3–B0 (IOB3–IOB0): These bits set the TGR3B register function.

Bit 7: IOB3	Bit 6: IOB2	Bit 5: IOB1	Bit 4: IOB0	Description		
0	0	0	0	TGR3B	Output disabled (initial value)	
			1	is an	Initial	Output 0 on compare-match
		1	0	output	output	Output 1 on compare-match
			1	compare	is 0	Toggle output on compare-match
	1	0	0	register	Output disabled	
			1		Initial	Output 0 on compare-match
		1	0		output	Output 1 on compare-match
			1		is 1	Toggle output on compare-match
1	0	0	0	TGR3B	Capture	Input capture on rising edge
			1	is an	input source	Input capture on falling edge
		1	0	input	is the	Input capture on both edges
			1	capture	TIOC3B pin	
	1	0	0	register		Input capture on rising edge
			1			Input capture on falling edge
		1	0			Input capture on both edges
			1			

- Bits 3–0—I/O Control A3–A0 (IOA3–IOA0): These bits set the TGR3A register function.

Bit 3: IOA3	Bit 2: IOA2	Bit 1: IOA1	Bit 0: IOA0	Description		
0	0	0	0	TGR3A	Output disabled (initial value)	
			1	is an	Initial	Output 0 on compare-match
		1	0	output	output	Output 1 on compare-match
			1	compare	is 0	Toggle output on compare-match
	1	0	0	register	Output disabled	
			1		Initial	Output 0 on compare-match
		1	0		output	Output 1 on compare-match
			1		is 1	Toggle output on compare-match
1	0	0	0	TGR3A	Capture	Input capture on rising edge
			1	is an	input source	Input capture on falling edge
		1	0	input	is the	Input capture on both edges
			1	capture	TIOC3A pin	
	1	0	0	register		Input capture on rising edge
			1			Input capture on falling edge
		1	0			Input capture on both edges
			1			

Channel 3 (TIOR3L Register):

- Bits 7–4—I/O Control D3–D0 (IOD3–IOD0): These bits set the TGR4D register function.

Bit 7: IOD3	Bit 6: IOD2	Bit 5: IOD1	Bit 4: IOD0	Description		
0	0	0	0	TGR3D	Output disabled (initial value)	
			1	is an	Initial	Output 0 on compare-match
		1	0	output	output	Output 1 on compare-match
			1	compare	is 0	Toggle output on compare-match
	1	0	0	register	Output disabled	
			1		Initial	Output 0 on compare-match
		1	0		output	Output 1 on compare-match
			1		is 1	Toggle output on compare-match
1	0	0	0	TGR3D	Capture	Input capture on rising edge
			1	is an	input source	Input capture on falling edge
		1	0	input	is the	Input capture on both edges
			1	capture	TIOC3D pin	
	1	0	0	register		Input capture on rising edge
			1			Input capture on falling edge
		1	0			Input capture on both edges
			1			

Note: When the BFB bit of TMDR3 is set to 1 and TGR3D is being used as a buffer register, these settings become ineffective and input capture/output compares do not occur.

- Bits 3–0—I/O Control C3–C0 (IOC3–IOC0): These bits set the TGR4C register function.

Bit 3: IOC3	Bit 2: IOC2	Bit 1: IOC1	Bit 0: IOC0	Description		
0	0	0	0	TGR3C	Output disabled (initial value)	
			1	is an	Initial	Output 0 on compare-match
		1	0	output	output	Output 1 on compare-match
			1	compare	is 0	Toggle output on compare-match
	1	0	0	register	Output disabled	
			1		Initial	Output 0 on compare-match
		1	0		output	Output 1 on compare-match
			1		is 1	Toggle output on compare-match
1	0	0	0	TGR3C	Capture	Input capture on rising edge
			1	is an	input source	Input capture on falling edge
		1	0	input	is the	Input capture on both edges
			1	capture	TIOC3C pin	
	1	0	0	register		Input capture on rising edge
			1			Input capture on falling edge
		1	0			Input capture on both edges
			1			

Note: When the BFA bit of TMDR3 is set to 1 and TGR3C is being used as a buffer register, these settings become ineffective and input capture/output compares do not occur.

Channel 4 (TIOR4H Register):

- Bits 7–4—I/O Control B3–B0 (IOB3–IOB0): These bits set the TGR4B register function.

Bit 7: IOB3	Bit 6: IOB2	Bit 5: IOB1	Bit 4: IOB0	Description		
0	0	0	0	TGR4B	Output disabled (initial value)	
			1	is an	Initial	Output 0 on compare-match
		1	0	output	output	Output 1 on compare-match
			1	compare	is 0	Toggle output on compare-match
	1	0	0	register	Output disabled	
			1		Initial	Output 0 on compare-match
		1	0		output	Output 1 on compare-match
			1		is 1	Toggle output on compare-match
1	0	0	0	TGR4B	Capture	Input capture on rising edge
			1	is an	input source	Input capture on falling edge
		1	0	input	is the	Input capture on both edges
			1	capture	TIOC4B pin	
	1	0	0	register		Input capture on rising edge
			1			Input capture on falling edge
		1	0			Input capture on both edges
			1			

- Bits 3–0—I/O Control A3–A0 (IOA3–IOA0): These bits set the TGR4A register function.

Bit 3: IOA3	Bit 2: IOA2	Bit 1: IOA1	Bit 0: IOA0	Description		
0	0	0	0	TGR4A	Output disabled (initial value)	
			1	is an	Initial	Output 0 on compare-match
		1	0	output	output	Output 1 on compare-match
			1	compare	is 0	Toggle output on compare-match
	1	0	0	register	Output disabled	
			1		Initial	Output 0 on compare-match
		1	0		output	Output 1 on compare-match
			1		is 1	Toggle output on compare-match
1	0	0	0	TGR4A	Capture	Input capture on rising edge
			1	is an	input source	Input capture on falling edge
		1	0	input	is the	Input capture on both edges
			1	capture	TIOC4A pin	
	1	0	0	register		Input capture on rising edge
			1			Input capture on falling edge
		1	0			Input capture on both edges
			1			

Channel 4 (TIOR4L Register):

- Bits 7–4—I/O Control D3–D0 (IOD3–IOD0): These bits set the TGR4D register function.

Bit 7: IOD3	Bit 6: IOD2	Bit 5: IOD1	Bit 4: IOD0	Description		
0	0	0	0	TGR4D	Output disabled (initial value)	
			1	is an	Initial	Output 0 on compare-match
		1	0	output	output	Output 1 on compare-match
			1	compare	is 0	Toggle output on compare-match
	1	0	0	register	Output disabled	
			1		Initial	Output 0 on compare-match
		1	0		output	Output 1 on compare-match
			1		is 1	Toggle output on compare-match
1	0	0	0	TGR4D	Capture	Input capture on rising edge
			1	is an	input source	Input capture on falling edge
		1	0	input	is the	Input capture on both edges
			1	capture	TIOC4D pin	
	1	0	0	register		Input capture on rising edge
			1			Input capture on falling edge
		1	0			Input capture on both edges
			1			

Note: When the BFB bit of TMDR4 is set to 1 and TGR4D is being used as a buffer register, these settings become ineffective and input capture/output compares do not occur.

- Bits 3–0—I/O Control C3–C0 (IOC3–IOC0): These bits set the TGR4C register function.

Bit 3: IOC3	Bit 2: IOC2	Bit 1: IOC1	Bit 0: IOC0	Description		
0	0	0	0	TGR4C	Output disabled (initial value)	
			1	is an	Initial	Output 0 on compare-match
		1	0	output	output	Output 1 on compare-match
			1	compare	is 0	Toggle output on compare-match
	1	0	0	register	Output disabled	
			1		Initial	Output 0 on compare-match
		1	0		output	Output 1 on compare-match
			1		is 1	Toggle output on compare-match
1	0	0	0	TGR4C	Capture	Input capture on rising edge
			1	is an	input source	Input capture on falling edge
		1	0	input	is the	Input capture on both edges
			1	capture	TIOC4C pin	
	1	0	0	register		Input capture on rising edge
			1			Input capture on falling edge
		1	0			Input capture on both edges
			1			

Note: When the BFA bit of TMDR4 is set to 1 and TGR4C is being used as a buffer register, these settings become ineffective and input capture/output compares do not occur.

12.2.4 Timer Interrupt Enable Register (TIER)

The TIER is an 8-bit register that controls the enable/disable of interrupt requests for each channel. The MTU has five TIER registers, one each for channel. TIER is initialized to H'40 by a reset or by standby mode.

Channel 0: TIER0:

Bit:	7	6	5	4	3	2	1	0
	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
Initial value:	0	1	0	0	0	0	0	0
R/W:	R/W	R	R	R/W	R/W	R/W	R/W	R/W

Channels 1, 2: TIER1, TIER2:

Bit:	7	6	5	4	3	2	1	0
	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA
Initial value:	0	1	0	0	0	0	0	0
R/W:	R/W	R	R/W	R/W	R	R	R/W	R/W

Channels 3, 4: TIER3, TIER4:

Bit:	7	6	5	4	3	2	1	0
	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
Initial value:	0	1	0	0	0	0	0	0
R/W:	R/W	R	R	R/W	R/W	R/W	R/W	R/W

- Bit 7—A/D Conversion Start Request Enable (TTGE): Enables or disables generation of an A/D conversion start request by a TGRA register input capture/compare-match.

Bit 7: TTGE	Description
0	Disable A/D conversion start requests (initial value)
1	Enable A/D conversion start request generation

- Bit 6—Reserved: This bit is reserved. It always reads as 0, and cannot be modified.
- Bit 5—Underflow Interrupt Enable (TCIEU): Enables or disables interrupt requests when the underflow flag (TCFU) of the channel 1, 2 timer status register (TSR) is set to 1.

  This bit is reserved for channels 0, 3, and 4. It always reads as 0. The write value should always be 1.

Bit 5: TCIEU	Description
0	Disable UDF interrupt requests (TCIU) (initial value)
1	Enable UDF interrupt requests (TCIU)

- Bit 4—Overflow Interrupt Enable (TCIEV): Enables or disables interrupt requests when the overflow flag TCFV of the timer status register (TSR) is set to 1.

Bit 4: TCIEV	Description
0	Disable TCFV interrupt requests (TCIV) (initial value)
1	Enable TCFV interrupt requests (TCIV)

- Bit 3—TGR Interrupt Enable D (TGIED): Enables or disables interrupt TGFD requests when the TGFD bit of the channel 0, 3, 4 TSR register is set to 1.

  This bit is reserved for channels 1 and 2. It always reads as 1. The write value should always be 1.

Bit 3: TGIED	Description
0	Disable interrupt requests (TGID) due to the TGFD bit (initial value)
1	Enable interrupt requests (TGID) due to the TGFD bit

- Bit 2—TGR Interrupt Enable C (TGIEC): Enables or disables TGFC interrupt requests when the TGFC bit of the channel 0, 3, 4 TSR register is set to 1.

  This bit is reserved for channels 1 and 2. It always reads as 1. The write value should always be 1.

Bit 2: TGIEC	Description
0	Disable interrupt requests (TGIC) due to the TGFC bit (initial value)
1	Enable interrupt requests (TGIC) due to the TGFC bit

- Bit 1—TGR Interrupt Enable B (TGIEB): Enables or disables TGFB interrupt requests when the TGFB bit of the TSR register is set to 1.

Bit 1: TGIEB	Description
0	Disable interrupt requests (TGIB) due to the TGFB bit (initial value)
1	Enable interrupt requests (TGIB) due to the TGFB bit

- Bit 0—TGR Interrupt Enable A (TGIEA): Enables or disables TGFA interrupt requests when the TGFA bit of the TSR register is set to 1.

Bit 0: TGIEA	Description
0	Disable interrupt requests (TGIA) due to the TGFA bit (initial value)
1	Enable interrupt requests (TGIA) due to the TGFA bit

12.2.5 Timer Status Register (TSR)

The timer status register (TSR) is an 8-bit register that indicates the status of each channel. The MTU has five TSR registers, one each for channel. TSR is initialized to H'C0 by a power-on reset or by standby mode. This register is not initialized by a manual reset.

Channel 0: TSR0:

Bit:	7	6	5	4	3	2	1	0
	—	—	—	TCFV	TGFD	TGFC	TGFB	TGFA
Initial value:	1	1	0	0	0	0	0	0
R/W:	R	R	R	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: * Only 0 writes to clear the flags are possible.

Channels 1, 2: TSR1, TSR2:

Bit:	7	6	5	4	3	2	1	0
	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA
Initial value:	1	1	0	0	0	0	0	0
R/W:	R	R	R/(W)*	R/(W)*	R	R	R/(W)*	R/(W)*

Note: * Only 0 writes to clear the flags are possible.

Channels 3, 4: TSR3, TSR4:

Bit:	7	6	5	4	3	2	1	0
	TCFD	—	—	TCFV	TGFD	TGFC	TGFB	TGFA
Initial value:	1	1	0	0	0	0	0	0
R/W:	R	R	R	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: * Only 0 writes to clear the flags are possible.

- Bit 7—Count Direction Flag (TCFD): This status flag indicates the count direction of the channel 1, 2, 3, 4 TCNT counters.

  This bit is reserved in channel 0. This bit always reads as 1. The write value should always be 1.

Bit 7: TCFD	Description
0	TCNT counts down
1	TCNT counts up (initial value)

- Bit 6—Reserved: This bit always reads as 1. The write value should always be 1.

- Bit 5—Underflow Flag (TCFU): This status flag indicates the occurrence of a channel 1, 2 TCNT counter underflow.

  This bit is reserved in channels 0, 3, and 4. This bit always reads as 0. The write value should always be 0.

Bit 5: TCFU	Description
0	Clear condition: With TCFU=1, a 0 write to TCFU after reading it (initial value)
1	Set condition: When the TCNT value underflows (H'0000 → H'FFFF)

- Bit 4—Overflow Flag (TCFV): This status flag indicates the occurrence of a TCNT counter overflow.

Bit 4: TCFV	Description
0	Clear condition: With TCFV =1, a 0 write to TCFV after reading it*1 (initial value)
1	Set condition: When the TCNT value overflows (H'FFFF → H'0000)*2

Notes: *1 For channel 4, this flag is cleared by DTC transfer due to TCFV.

*2 For channel 4, this flag is also set when the TCNT value underflows (H'0001 → H'0000) in complementary PWM mode.

- Bit 3—Input Capture/Output Compare Flag D (TGFD): This status flag indicates the occurrence of a channel 0, 3, or 4 TGRD register input capture or compare-match.

  This bit is reserved in channels 1 and 2. It always reads as 0. The write value should always be 0.

Bit 3: TGFD	Description
0	Clear condition: With TGFD = 1, a 0 write to TGFD following a read (Cleared by DTC transfer due to TGFD) (initial value)
1	Set conditions: • When TGRD is functioning as an output compare register (TCNT = TGRD) • When TGRD is functioning as input capture (the TCNT value is sent to TGRD by the input capture signal)

- Bit 2—Input Capture/Output Compare Flag C (TGFC): This status flag indicates the occurrence of a channel 0, 3, or 4 TGRC register input capture or compare-match.

  This bit is reserved for channels 1 and 2. It always reads as 0. The write value should always be 0.

Bit 2: TGFC	Description
0	Clear condition: With TGFC = 1, a 0 write to TGFC following a read (Cleared by DTC transfer due to TGFC) (initial value)
1	Set conditions: • When TGRC is functioning as an output compare register (TCNT = TGRC) • When TGRC is functioning as input capture (the TCNT value is sent to TGRC by the input capture signal)

- Bit 1—Input Capture/Output Compare Flag B (TGFB): This status flag indicates the occurrence of a TGRB register input capture or compare-match.

Bit 1: TGFB	Description
0	Clear condition: With TGFB = 1, a 0 write to TGFB following a read (Cleared by DTC transfer due to TGFB) (initial value)
1	Set conditions: • When TGRB is functioning as an output compare register (TCNT = TGRB) • When TGRB is functioning as input capture (the TCNT value is sent to TGRB by the input capture signal)

- Bit 0—Input Capture/Output Compare Flag A (TGFA): This status flag indicates the occurrence of a TGRA register input capture or compare-match.

Bit 0: TGFA	Description
0	Clear condition: With TGFA = 1, a 0 write to TGFA following a read (Cleared by DMAC transfer due to TGFA) (initial value)
1	Set conditions: • When TGRA is functioning as an output compare register (TCNT = TGRA) • When TGRA is functioning as input capture (the TCNT value is sent to TGRA by the input capture signal)

12.2.6 Timer Counters (TCNT)

The timer counters (TCNT) are 16-bit counters, with one for each channel, for a total of five.

The TCNT are initialized to H'0000 by a power-on reset and when in standby mode. Manual reset does not initialize TCNT. Accessing the TCNT counters in 8-bit units is prohibited. Always access in 16-bit units.

Channel	Abbreviation	Function
0	TCNT0	Increment counter
1	TCNT1	Increment/decrement counter*1
2	TCNT2	Increment/decrement counter*1
3	TCNT3	Increment/decrement counter*2
4	TCNT4	Increment/decrement counter*2

Notes: *1 Can only be used as an increment/decrement counter in phase counting mode, with other channel overflow/underflow counting. It becomes an increment counter in all other cases.

*2 Can only be used as an increment counter in complementary PWM mode. It becomes an increment counter in all other cases.

Bit:	15	14	13	12	11	10	9	8
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

12.2.7 Timer General Register (TGR)

Each timer general register (TGR) is a 16-bit register that can function as either an output compare register or an input capture register. There are a total of sixteen TGR, four each for channels 0, 3, and 4, and two each for channels 1 and 2. The TGRC and TGRD of channels 0, 3, and 4 can be set to operate as buffer registers. The TGR register and buffer register combinations are TGRA with TGRC, and TGRB with TGRD.

The TGRs are initialized to H'FFFF by a power-on reset or in standby mode. Manual reset does not initialize TGR. Accessing of the TGRs in 8-bit units is disabled; they may only be accessed in 16-bit units.

Bit:	15	14	13	12	11	10	9	8
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

12.2.8 Timer Start Register (TSTR)

The timer start register (TSTR) is an 8-bit read/write register that starts and stops the timer counters (TCNT) of channels 0–4. TSTR is initialized to H'00 upon power-on reset or standby mode. Manual reset does not initialize TSTR.

Bit:	7	6	5	4	3	2	1	0
	CST4	CST3	—	—	—	CST2	CST1	CST0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R	R	R/W	R/W	R/W

- Bits 7, 6, 2–0—Counter Start 4–0 (CST4–CST0): Select the start and stop of the timer counters (TCNT). The counter start to channel and bit to channel correspondence are indicated in the tables below.

Counter Start	Channel
CST4	Channel 4 (TCNT4)
CST3	Channel 3 (TCNT3)
CST2	Channel 2 (TCNT2)
CST1	Channel 1 (TCNT1)
CST0	Channel 0 (TCNT0)

Bit n: CSTn	Description
0	TCNTn count is halted (initial value)
1	TCNTn counts

Note: n = 4 to 0. However, CST4 is bit 7, CST3 is bit 6.

If 0 is written to the CST bit during operation with the TIOC pin in output status, the counter stops, but the TIOC pin output compare output level is maintained. If a write is done to the TIOR register while the CST bit is a 0, the pin output level is updated to the established initial output value. In complementary PWM mode or reset sync PWM mode, when a 0 is written to the CST bit of a TIOC pin in output mode during operation, it returns to the initial output.

- Bits 5–3—Reserved: These bits always read as 0. The write value should always be 0.

12.2.9 Timer Synchro Register (TSYR)

The timer synchro register (TSYR) is an 8-bit read/write register that selects independent or synchronous TCNT counter operation for channels 0–4. Channels for which 1 is set in the corresponding bit will be synchronized. TSYR is initialized to H'00 upon power-on reset or standby mode. Manual reset does not initialize TSYR.

Bit:	7	6	5	4	3	2	1	0
	SYNC4	SYNC3	—	—	—	SYNC2	SYNC1	SYNC0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R	R	R/W	R/W	R/W

- Bits 7, 6, 2–0—Timer Synchronization 4–0 (SYNC4–SYNC0): Selects operation independent of, or synchronized to, other channels. Synchronous operation allows synchronous clears due to multiple TCNT synchronous presets and other channel counter clears. A minimum of two channels must have SYNC bits set to 1 for synchronous operation. For synchronization clearing, it is necessary to set the TCNT counter clear sources (the CCLR2–CCLR0 bits of the TCR register), in addition to the SYNC bit. The counter start to channel and bit-to-channel correspondence are indicated in the tables below.

Counter Start	Channel
SYNC4	Channel 4 (TCNT4)
SYNC3	Channel 3 (TCNT3)
SYNC2	Channel 2 (TCNT2)
SYNC1	Channel 1 (TCNT1)
SYNC0	Channel 0 (TCNT0)

Bit n: SYNCn	Description
0	Timer counter (TCNTn) independent operation (initial value) (TCNTn preset/clear unrelated to other channels)
1	Timer counter synchronous operation*1 TCNTn synchronous preset/ synchronous clear*2 possible

Notes: n = 4 to 0. However, SYNC4 is bit 7, SYNC3 is bit 6.

*1 Minimum of two channel SYNC bits must be set to 1 for synchronous operation.

*2 TCNT counter clear sources (CCLR2–CCLR0 bits of the TCR register) must be set in addition to the SYNC bit in order to have clear synchronization.

- Bits 5–3—Reserved: These bits always read as 0. The write value should always be 0.

12.2.10 Timer Output Master Enable Register (TOER)

The timer output master enable register (TOER) enables/disables output settings for output pins TIOC4D, TIOC4C, TIOC3D, TIOC4B, TIOC4A, and TIOC3B. These pins do not output correctly if the TOER bits have not been set. Set TOER of CH3 and CH2 prior to setting TIOR of CH3 and CH4. The TOER is an 8-bit read/write register. The register is initialized to H'C0 by a power-on reset or in standby mode. Manual reset does not initialize TOER.

Bit:	7	6	5	4	3	2	1	0
	—	—	OE4D	OE4C	OE3D	OE4B	OE4A	OE3B
Initial value:	1	1	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W

- Bits 7–6—Reserved: These bits always read as 1. The write value should always be 1.

- Bit 5—Master Enable TIOC4D (OE4D): Enables or disables the TIOC4D pin MTU output.

Bit 5: OE4D	Description
0	Disable TIOC4D pin MTU output (initial value)
1	Enable TIOC4D pin MTU output

- Bit 4—Master Enable TIOC4C (OE4C): Enables or disables the TIOC4C pin MTU output.

Bit 4: OE4C	Description
0	Disable TIOC4C pin MTU output (initial value)
1	Enable TIOC4C pin MTU output

- Bit 3—Master Enable TIOC3D (OE3D): Enables or disables the TIOC3D pin MTU output.

Bit 3: OE3D	Description
0	Disable TIOC3D pin MTU output (initial value)
1	Enable TIOC3D pin MTU output

- Bit 2—Master Enable TIOC4B (OE4B): Enables or disables the TIOC4B pin MTU output.

Bit 2: OE4B	Description
0	Disable TIOC4B pin MTU output (initial value)
1	Enable TIOC4B pin MTU output

- Bit 1—Master Enable TIOC4A (OE4A): Enables or disables the TIOC4A pin MTU output.

Bit 1: OE4A	Description
0	Disable TIOC4A pin MTU output (initial value)
1	Enable TIOC4A pin MTU output

- Bit 0—Master Enable TIOC3B (OE3B): Enables or disables the TIOC3B pin MTU output.

Bit 0: OE3B	Description
0	Disable TIOC3B pin MTU output (initial value)
1	Enable TIOC3B pin MTU output

12.2.11 Timer Output Control Register (TOCR)

The timer output control register (TOCR) enables/disables PWM synchronized toggle output in complementary PWM mode and reset sync PWM mode, and controls output level inversion of PWM output. The TOCR is initialized to H'00 by a power-on reset or in the standby mode. Manual reset does not initialize TOCR. These register settings are ineffective for anything other than complementary PWM mode/reset-synchronized PWM mode.

Bit:	7	6	5	4	3	2	1	0
	—	PSYE	—	—	—	—	OLSN	OLSP
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R	R	R	R/W	R/W

- Bits 7, 5–2—Reserved: These bits always read as 1. The write value should always be 1.
- Bit 6—PWM Synchronous Output Enable (PSYE): Selects the enable/disable of toggle output synchronized with the PWM period.

Bit 6: PSYE	Description
0	Toggle output synchronous with PWM period disabled (initial value)
1	Toggle output synchronous with PWM period enabled

- Bit 1—Output Level Select N (OLSN): Selects the reverse phase output level of the complementary PWM mode or reset-synchronized PWM mode.

			Compare Match Output	
OLSN	**Initial Output**	**Active Level**	**Increment Count**	**Decrement Count**
0	High level*	Low level	High level	Low level (initial value)
1	Low level*	High level	Low level	High level

Note: * The reverse phase waveform initial output value changes to active level after elapse of the dead time after count start.

- Bit 0—Output Level Select P (OLSP): Selects the positive phase output level of the complementary PWM mode or reset-synchronized PWM mode.

			Compare Match Output	
OLSP	**Initial Output**	**Active Level**	**Increment Count**	**Decrement Count**
0	High level	Low level	Low level	High level (initial value)
1	Low level	High level	High level	Low level

Figure 12.2 shows an example of complementary PWM mode output (1 phase) when OLSN = 1, OLSP = 1.

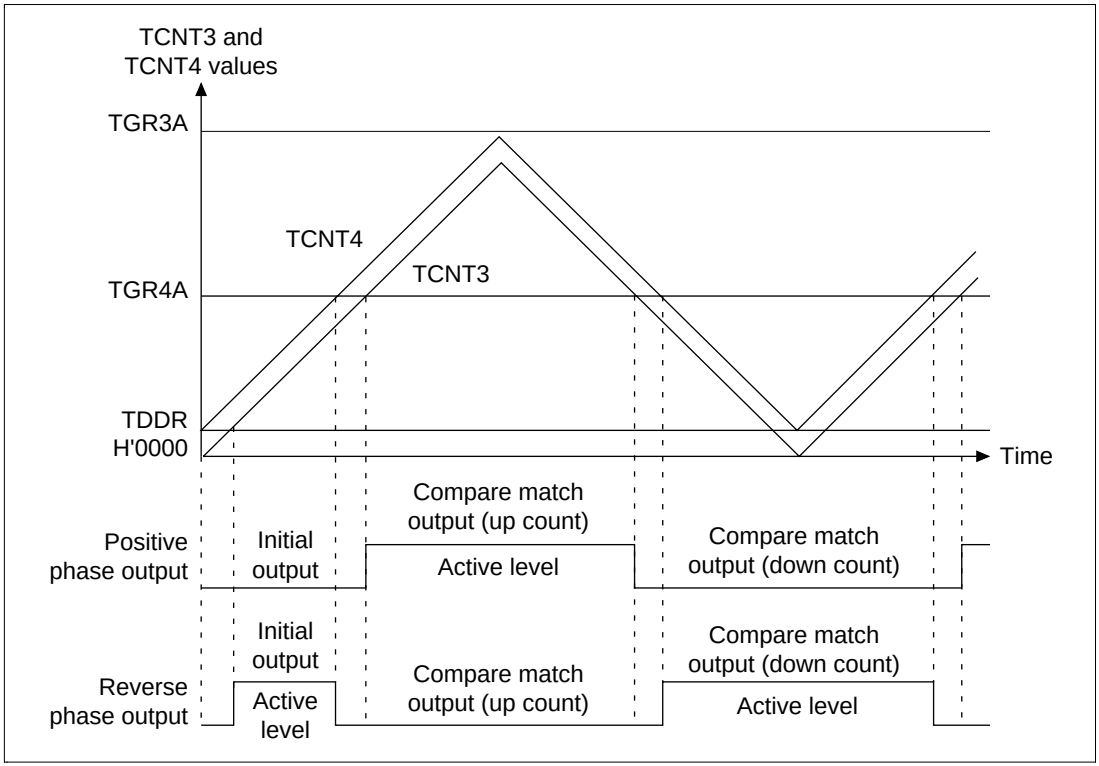


Figure 12.2 Complementary PWM Mode Output Level Example

12.2.12 Timer Gate Control Register (TGCR)

The timer gate control register (TGCR) is an 8-bit read/write register that controls the waveform output necessary for brushless DC motor control in complementary PWM mode/reset-synchronized PWM mode. The TGCR is initialized to H'80 by a power-on reset or in the standby mode. Manual reset does not initialize TGCR. These register settings are ineffective for anything other than complementary PWM mode/reset-synchronized PWM mode.

Bit:	7	6	5	4	3	2	1	0
	—	BDC	N	P	FB	WF	VF	UF
Initial value:	1	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- Bit 7—Reserved: This bit always reads as 1. The write value should always be 1.

- Bit 6—Brushless DC Motor (BDC): Selects gate signal output/chopping output function for brushless DC motor control.

Bit 6: BDC	Description
0	Ordinary output (initial value)
1	Gate signal/chopping output for brushless DC motor

- Bit 5—Reverse Phase Output (N): Selects whether to output gate signals directly to the reverse phase pin (TIOC3D, TIOC4C, and TIOC4D) output, or to output by chopping the gate signal and the complementary PWM/reset-synchronized PWM output.

Bit 5: N	Description
0	Output gate signals directly to reverse phase pin output (initial value)
1	Output chopped gate signal and complementary PWM /reset-synchronized PWM output to reverse phase pin output

- Bit 4—Positive Phase Output (P): Selects whether to output gate signals directly to the positive phase pin (TIOC3B, TIOC4A, and TIOC4B) output, or to output by chopping the gate signal and the complementary PWM/reset-synchronized PWM output.

Bit 4: P	Description
0	Output gate signals directly to positive phase pin output (initial value)
1	Output chopped gate signal and complementary PWM /reset-synchronized PWM output to positive phase pin output

- Bit 3—Feedback Input (FB): Selects whether to use external input or register input for the feedback input to generate gate signals.

Bit 3: FB	Description
0	Feedback input is external input (initial value) (Input sources are channel 0 TGRA, TGRB, TGRC input capture signals)
1	Feedback input is register input (TGCR's UF, VF, WF settings)

- Bits 2–0—Output Phase Switch 2–0 (WF, VF, UF): These bits set the positive phase/negative phase output phase on or off state. The setting of these bits is valid only when the FB bit in this register is set to 1. In this case, the setting of bits 2–0 is a substitute for external input.

			Function						
			TIOC3B	TIOC4A	TIOC4B	TIOC3D	TIOC4C	TIOC4D	
Bit 2: WF	Bit 1: VF	Bit 0: UF	U Phase	V Phase	W Phase	U Phase	V Phase	W Phase	
0	0	0	Off	Off	Off	Off	Off	Off	Initial value
		1	On	Off	Off	Off	Off	On	—
	1	0	Off	On	Off	On	Off	Off	—
		1	Off	On	Off	Off	Off	On	—
1	0	0	Off	Off	On	Off	On	Off	—
		1	On	Off	Off	Off	On	Off	—
	1	0	Off	Off	On	On	Off	Off	—
		1	Off	Off	Off	Off	Off	Off	—

12.2.13 Timer Subcounter (TCNTS)

The timer subcounter (TCNTS) is a 16-bit read-only counter that is used only in complementary PWM mode. The TCNTS counter is initialized to H'00 by a power-on reset or in standby mode. Manual reset does not initialize TCNTS. Accessing the TCNTS counter in 8-bit units is prohibited. Always access in 16-bit units.

Bit:	15	14	13	12	11	10	9	8
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

12.2.14 Timer Dead Time Data Register (TDDR)

The timer dead time data register (TDDR) is a 16-bit register, used only in complementary PWM mode, that specifies the TCNT3 and TCNT4 counter offset values. In complementary PWM mode, when the TCNT3 and TCNT4 counters are cleared and then restarted, the TDDR register value is loaded into the TCNT3 counter and the count operation starts. The TDDR register is initialized to H'FFFF by a power-on reset or in standby mode. Manual reset does not initialize TDDR. Accessing the TDDR in 8-bit units is prohibited. Always access in 16-bit units.

Bit:	15	14	13	12	11	10	9	8
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

12.2.15 Timer Period Data Register (TCDR)

The timer period data register (TCDR) is a 16-bit register used only in complementary PWM mode. Set the PWM carrier sync value as the TCDR register value. This register is constantly compared with the TCNTS counter in complementary PWM mode, and when a match occurs the TCNTS counter switches direction (decrement to increment).

The TCDR register is initialized to H'FFFF by a reset or in standby mode. Manual reset does not initialize TCDR. Accessing the TCDR in 8-bit units is prohibited. Always access in 16-bit units.

Bit:	15	14	13	12	11	10	9	8
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

12.2.16 Timer Period Buffer Register (TCBR)

The timer period buffer register (TCBR) is a 16-bit register used only in complementary PWM mode. It functions as a buffer register for the TCDR register. The TCBR register values are transferred to the TCDR register with the transfer timing established in the TMDR register. The TCBR register is initialized to H'FFFF by a power-on reset or in standby mode. Manual reset does not initialize TCBR. Accessing the TCBR in 8-bit units is prohibited. Always access in 16-bit units.

Bit:	15	14	13	12	11	10	9	8
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

12.3 Bus Master Interface

12.3.1 16-Bit Registers

The timer counters (TCNT) and general registers (TGR) are 16-bit registers. A 16-bit data bus to the bus master enables 16-bit read/writes. 8-bit read/write is not possible. Always access in 16-bit units. Figure 12.3 shows an example of 16-bit register access operation.

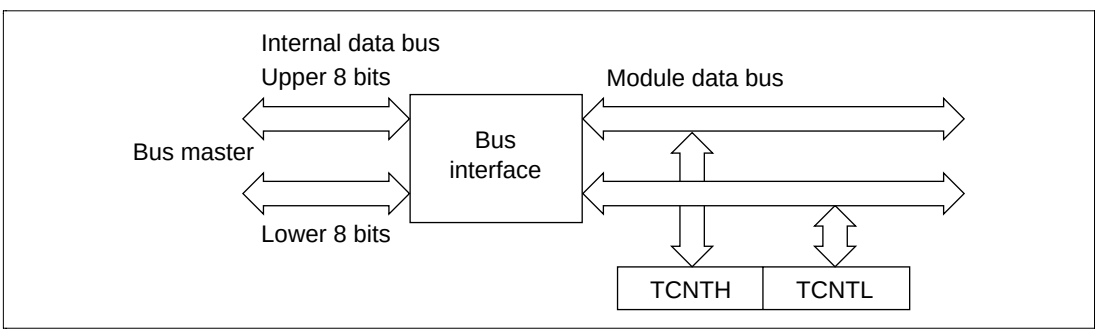


Figure 12.3 16-Bit Register Access Operation (Bus Master ↔ TCNT (16 Bit))

12.3.2 8-Bit Registers

All registers other than the TCNT and general registers (TGR) are 8-bit registers. These are connected to the CPU by a 16-bit data bus, so 16-bit read/writes and as 8-bit read/writes are both possible (figures 12.4 ,12.5, and 12.6).

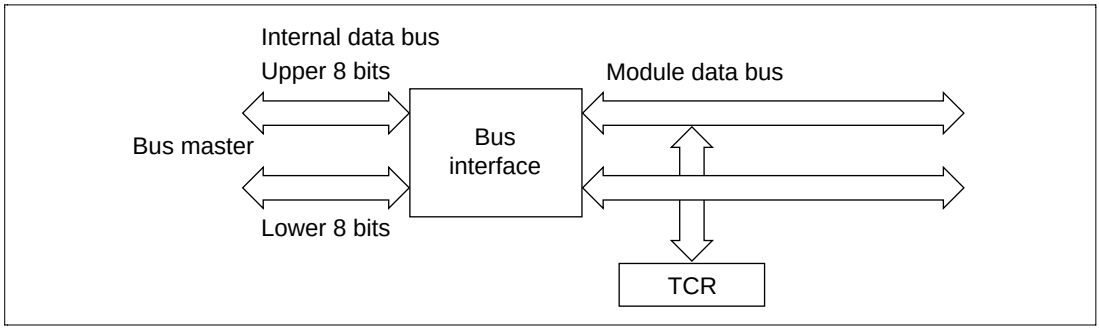


Figure 12.4 8-Bit Register Access Operation (Bus Master ↔ TCR (Upper 8 Bits))

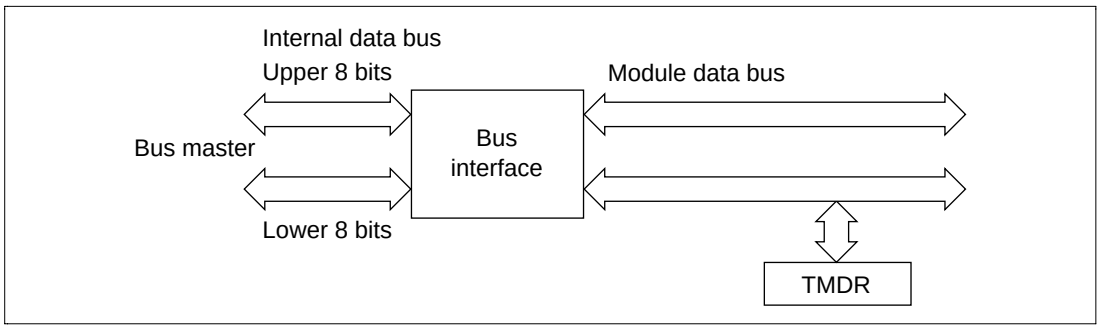


Figure 12.5 8-Bit Register Access Operation (Bus Master ↔ TMDR (Lower 8 Bits))

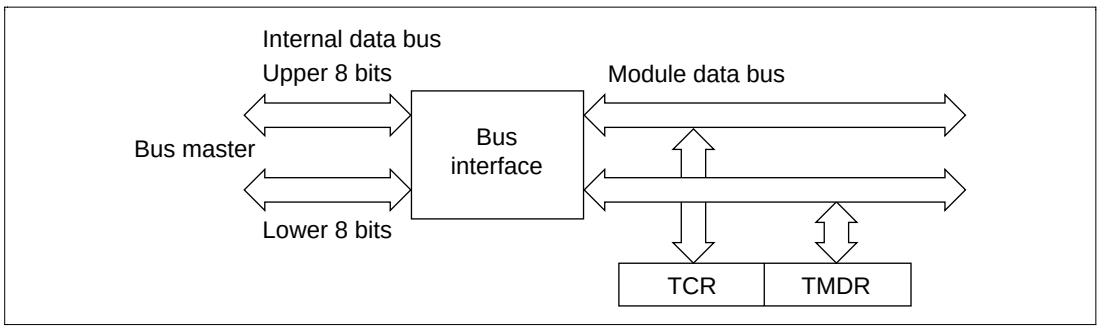


Figure 12.6 8-Bit Register Access Operation (Bus Master ↔ TCR, TMDR (16 Bit))

12.4 Operation

12.4.1 Overview

The operation modes are described below.

Ordinary Operation: Each channel has a timer counter (TCNT) and general register (TGR). The TCNT is an upcounter and can also operate as a free-running counter, periodic counter or external event counter. General registers (TGR) can be used as output compare registers or input capture registers.

Synchronized Operation: The TCNT of a channel set for synchronized operation does a synchronized preset. When any TCNT of a channel operating in the synchronized mode is rewritten, the TCNTs of other channels are simultaneously rewritten as well. The timer synchronization bits of the TSYR registers of multiple channels set for synchronous operation can be set to clear the TCNTs simultaneously.

Buffer Operation: When TGR is an output compare register, the buffer register value of the corresponding channel is transferred to the TGR when a compare-match occurs. When TGR is an input capture register, the TCNT counter value is transferred to the TGR when an input capture occur simultaneously the value previously stored in the TGR is transferred to the buffer register.

Cascade Connection Operation: The channel 1 and channel 2 counters (TCNT1 and TCNT2) can be connected together to operate as a 32-bit counter.

PWM Mode: In PWM mode, a PWM waveform is output. The output level can be set by the TIOR register. Each TGR can be set for PWM waveform output with a duty cycle between 0% and 100%.

Phase Counting Mode: In phase counting mode, the phase differential between two clocks input from the channel 1 and channel 2 external clock input pins is detected and the TCNT counter operates as an up/down counter. In phase counting mode, the corresponding TCLK pins become clock inputs and TCNT functions as an up/down counter. It can be used as a two-phase encoder pulse input.

Reset-Synchronized PWM Mode: Three-phase positive and negative PWM waveforms can be obtained using channels 3 and 4 (the three phases of the PWM waveform share a transition point on one side). When set for reset-synchronized PWM mode, TGR3A, TGR3B, TGR4A, and TGR4B automatically become output compare registers. The TIOC3A, TIOC3B, TIOC4A, TIOC4B, TIOC4C, and TIOC4D pins also become PWM output pins, and TCNT3 and TCNT4 become upcounters. TCNT4, TGR4A, and TGR4B are isolated from TCNT4.

Complementary PWM Mode: Three-phase complementary positive and negative PWM waveforms whose positive and negative phases do not overlap can be obtained using channels 3 and 4. When set for complementary PWM mode, TGR3A, TGR3B, TGR4A, and TGR4B become output compare registers. The TIOC3B, TIOC3D, TIOC4A, TIOC4B, TIOC4C, and TIOC4D pins also automatically become PWM output pins while TCNT3 and TCNT4 become up/down counters.

12.4.2 Basic Functions

Always select MTU external pin set function using the pin function controller (PFC).

Counter Operation: When a start bit (CST0–CST4) in the timer start register (TSTR) is set to 1, the corresponding timer counter (TCNT) starts counting. There are two counting modes: a free-running mode and a periodic mode.

To select the counting operation (figure 12.7):

1. Set bits TPSC2–TPSC0 in the TCR to select the counter clock. At the same time, set bits CKEG1 and CKEG0 in the TCR to select the desired edge of the input clock.
2. To operate as a periodic counter, set the CCLR2–CCLR0 bits in the TCR to select TGR as a clearing source for the TCNT.
3. Set the TGR selected in step 2 as an output compare register using the timer I/O control register (TIOR).
4. Write the desired cycle value in the TGR selected in step 2.
5. Set the CST bit in the TSTR to 1 to start counting.

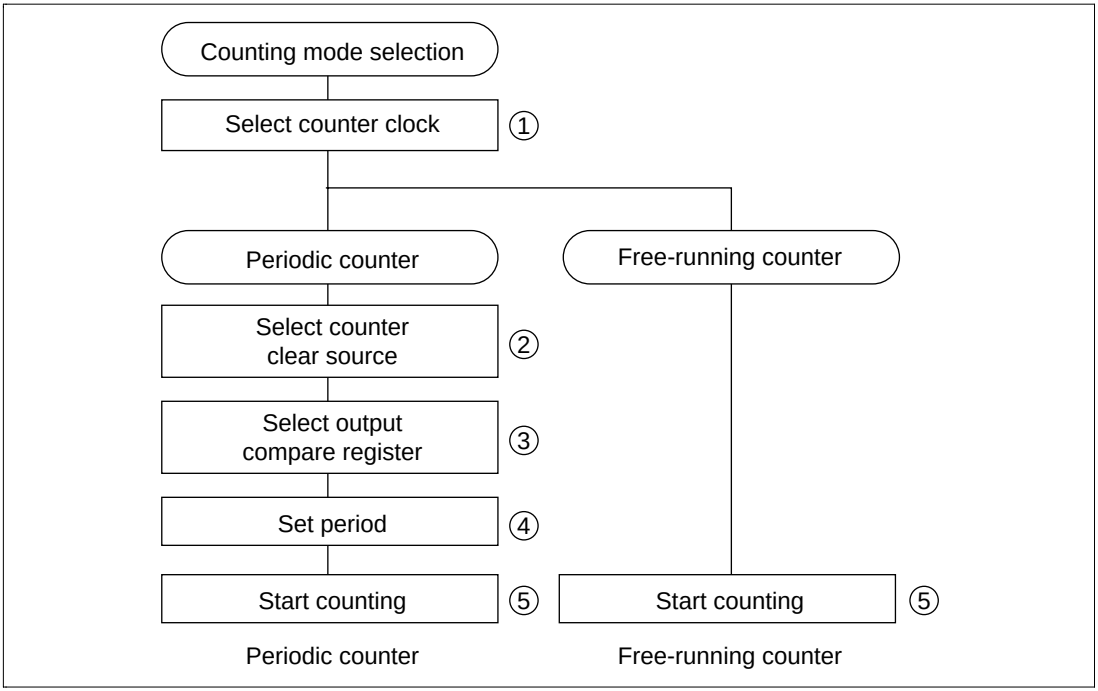


Figure 12.7 Procedure for Selecting the Counting Operation

Free-Running Counter Operation Example: A reset of the MTU timer counters (TCNT) leaves them all in the free-running mode. When a bit in the TSTR is set to 1, the corresponding timer counter operates as a free-running counter and begins to increment. When the count overflows from H'FFFF–H'0000, the TCFV bit in the timer status register (TSR) is set to 1. If the TCIEV bit in the timer's corresponding timer interrupt enable register (TIER) is set to 1, the MTU will make an interrupt request to the interrupt controller. After the TCNT overflows, counting continues from H'0000. Figure 12.8 shows an example of free-running counter operation.

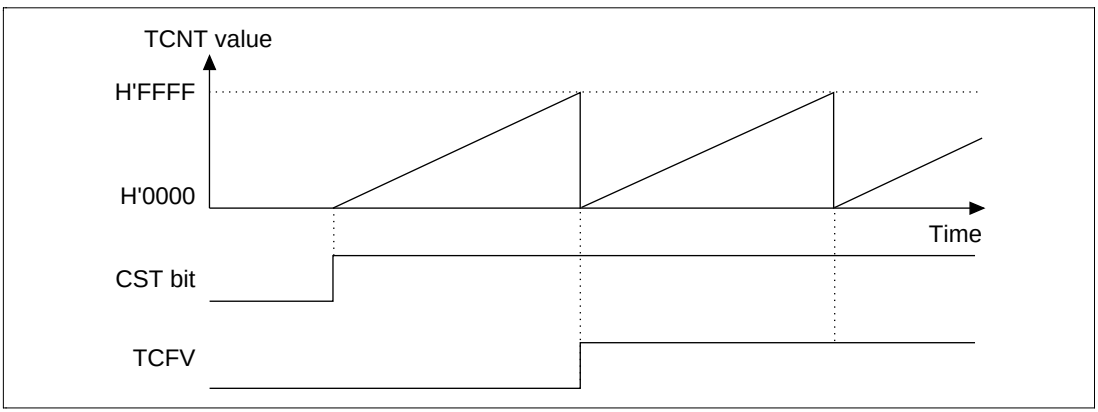


Figure 12.8 Free-Running Counter Operation

Periodic Counter Operation Example: Periodic counter operation is obtained for a given channel's TCNT by selecting compare-match as a TCNT clear source. Set the TGR register for period setting to output compare register and select counter clear upon compare-match using the CCLR2–CCLR0 bits of the timer control register (TCR). After these settings, the TCNT begins incrementing as a periodic counter when the corresponding bit of TSTR is set to 1. When the count matches the TGR register value, the TGF bit in the TSR is set to 1 and the counter is cleared to H'0000. If the TGIE bit of the corresponding TIER is set to 1 at this point, the MTU will make an interrupt request to the interrupt controller. After the compare-match, TCNT continues counting from H'0000. Figure 12.9 shows an example of periodic counting.

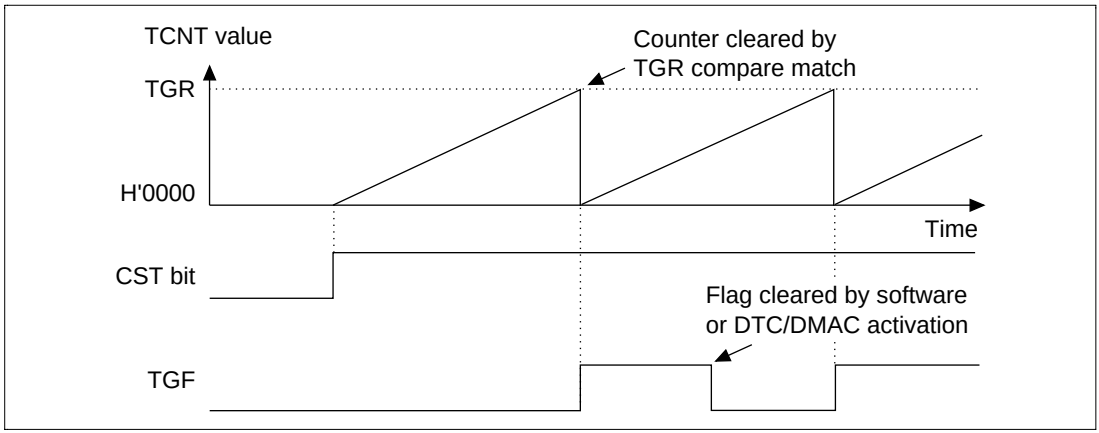


Figure 12.9 Periodic Counter Operation

Compare-Match Waveform Output Function: The MTU can output 0 level, 1 level, or toggle output from the corresponding output pins upon compare-matches.

Procedure for selecting the compare-match waveform output operation (figure 12.10):

1. Set the TIOR to select 0 output or 1 output for the initial value, and 0 output, 1 output, or toggle output for compare-match output. The TIOC pin will output the set initial value until the first compare-match occurs.
2. Set a value in the TGR to select the compare-match timing.
3. Set the CST bit in the TSTR to 1 to start counting.

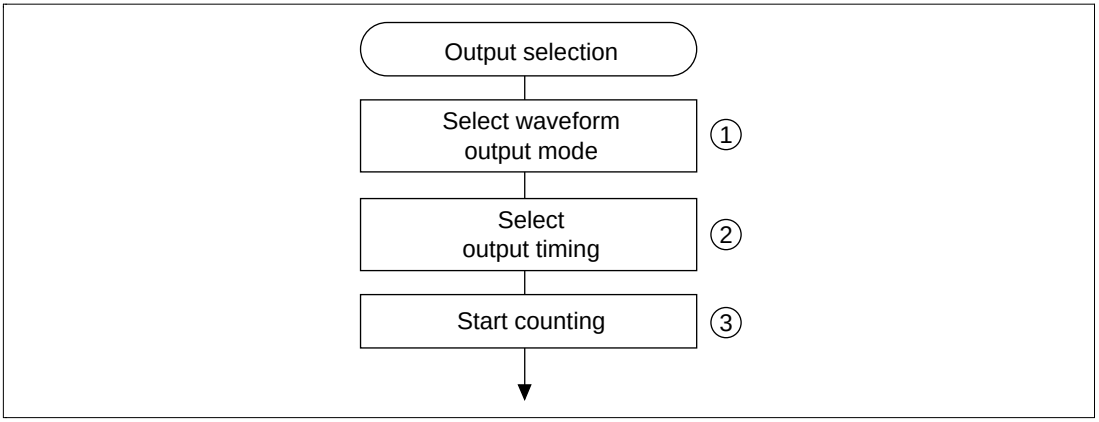


Figure 12.10 Procedure for Selecting Compare Match Waveform Output Operation

Waveform Output Operation (0 Output/1 Output): Figure 12.11 shows 0 output/1 output. In the example, TCNT is a free-running counter, 1 is output upon compare-match A and 0 is output upon compare-match B. When the pin level matches the set level, the pin level does not change.

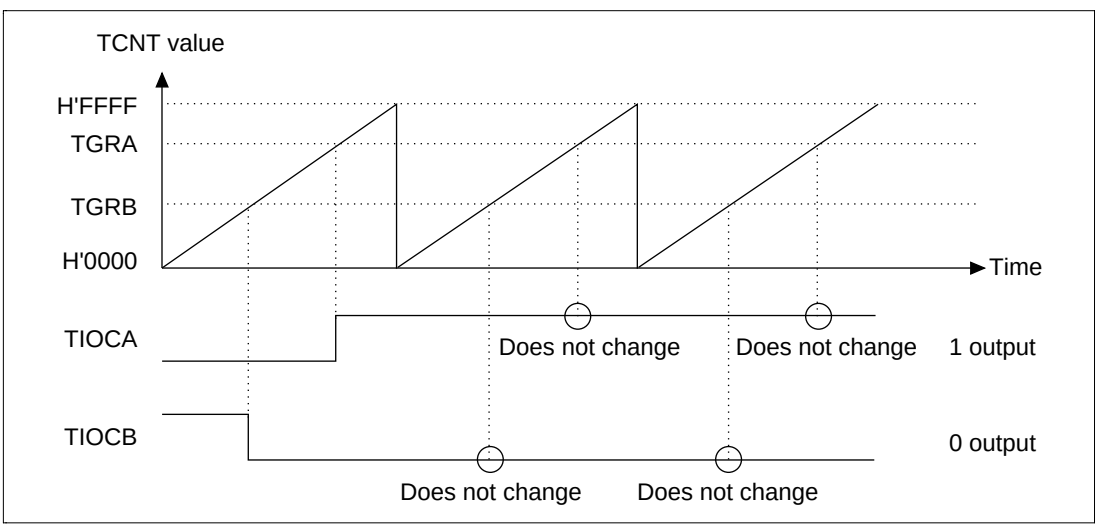


Figure 12.11 Example of 0 Output/1 Output

Waveform Output Operation (Toggle Output): Figure 12.12 shows the toggle output. In the example, the TCNT operates as a periodic counter cleared by compare-match B, with toggle output at both compare-match A and compare-match B.

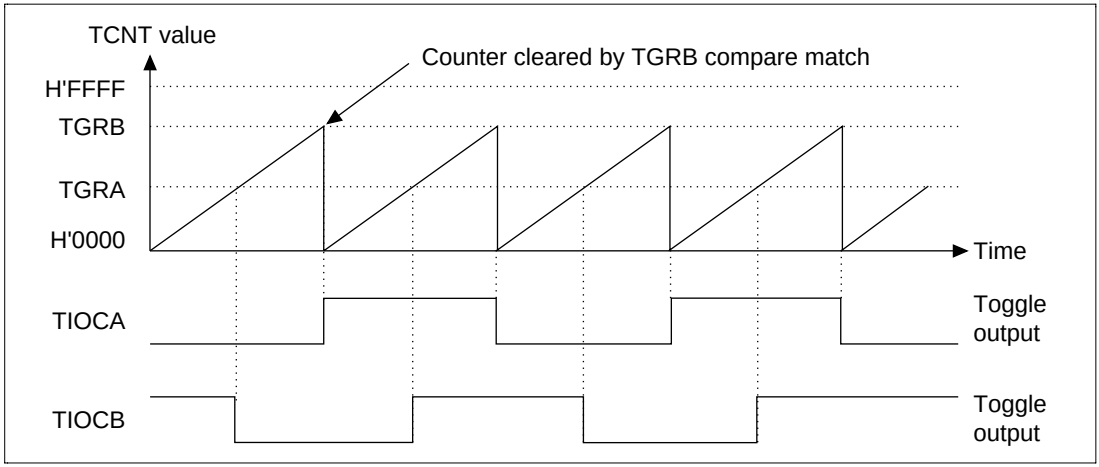


Figure 12.12 Example of Toggle Output

Input Capture Function: In the input capture mode, the TCNT value is transferred into the TGR register when the input edge is detected at the input capture/output compare pin (TIOC).

Detection can take place on the rising edge, falling edge, or both edges. Channels 0 and 1 can use other channel counter input clocks or compare-match signals as input capture sources.

The procedure for selecting the input capture operation (figure 12.13) is:

1. Set the TIOR to select the input capture function of the TGR, then select the input capture source, and rising edge, falling edge, or both edges as the input edge.
2. Set the CST bit in the TSTR to 1 to start the TCNT counting.

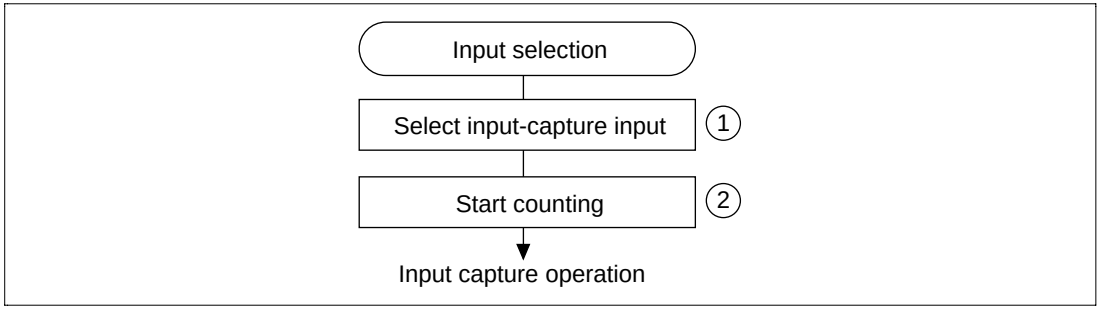


Figure 12.13 Procedure for Selecting Input Capture Operation

Input Capture Operation: Figure 12.14 shows input capture. The falling edge of TIOCB and both edges of TIOCA are selected as input capture input edges. In the example, TCNT is set to clear at the input capture of the TGRB register.

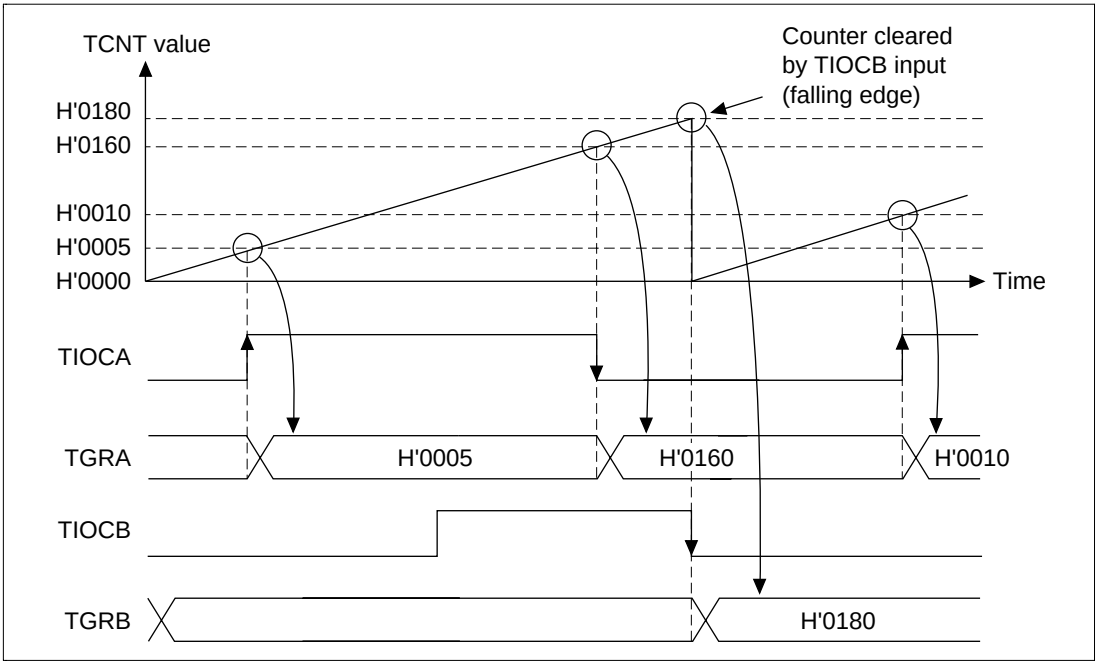


Figure 12.14 Input Capture Operation

12.4.3 Synchronous Operation

In the synchronizing mode, two or more timer counters can be rewritten simultaneously (synchronized preset). Multiple timer counters can also be cleared simultaneously using TCR settings (synchronized clear).

The synchronizing mode can increase the number of TGR registers for a single time base. All five channels can be set for synchronous operation.

Procedure for Selecting the Synchronizing Mode (Figure 12.15):

1. Set 1 in the SYNC bit of the timer synchro register (TSYR) to use the corresponding channel in the synchronizing mode.
2. When a value is written in the TCNT in any of the synchronized channels, the same value is simultaneously written in the TCNT in the other channels.
3. Set the counter to clear with output compare/input capture using bits CCLR2–CCLR0 in the TCR.
4. Set the counter clear source to synchronized clear using the CCLR2–CCLR0 bits of the TCR.
5. Set the CST bits for the corresponding channels in the TSTR to 1 to start counting in the TCNT.

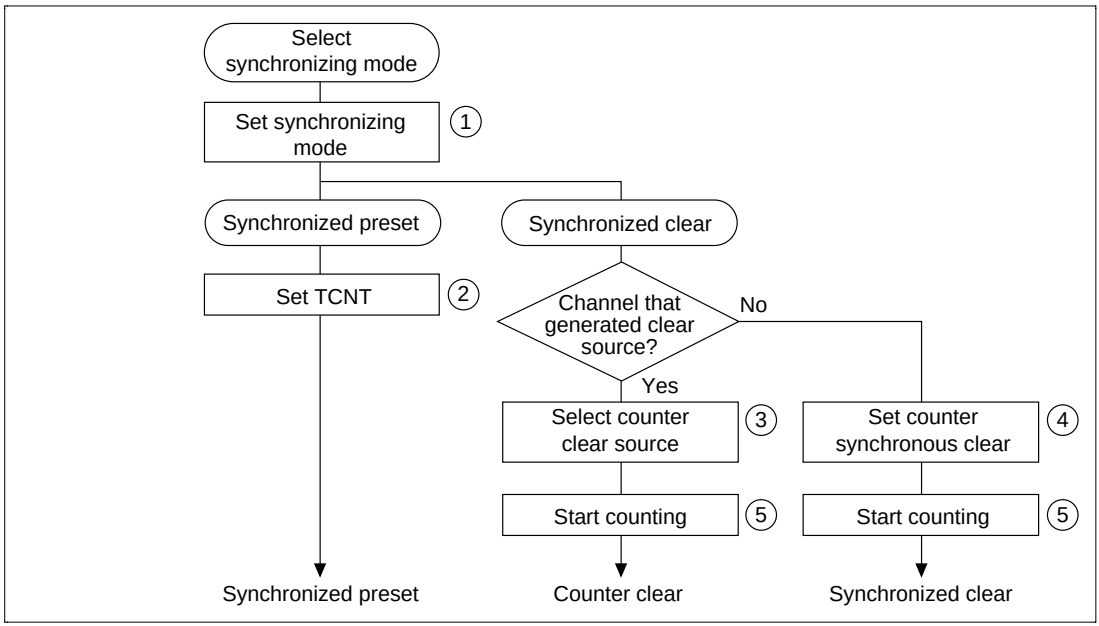


Figure 12.15 Procedure for Selecting Synchronizing Operation

Synchronized Operation: Figure 12.16 shows an example of synchronized operation. Channels 0, 1, and 2 are set to synchronized operation and PWM mode 1. Channel 0 is set for a counter clear upon compare-match with TGR0B. Channels 1 and 2 are set for synchronous counter clears by synchronous presets and TGR0B register compare-matches. Accordingly, a three-phase PWM waveform with the data set in the TGR0B register as its PWM period is output from the TIOC0A, TIOC1A, and TIOC2A pins.

See section 12.4.6, PWM Mode, for details on the PWM mode.

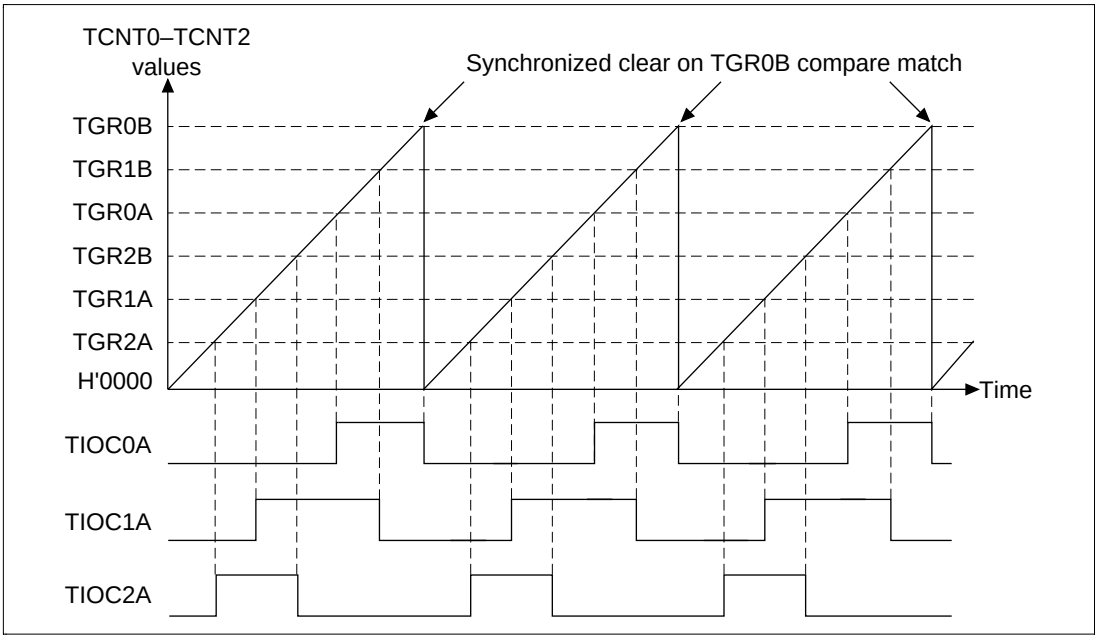


Figure 12.16 Synchronized Operation Example

12.4.4 Buffer Operation

Buffer operation is a function of channels 0, 3, and 4. TGRC and TGRD can be used as buffer registers. Table 12.5 shows the register combinations for buffer operation.

Table 12.5 Register Combinations

Channel	General Register	Buffer Register
0	TGR0A	TGR0C
	TGR0B	TGR0D
3	TGR3A	TGR3C
	TGR3B	TGR3D
4	TGR4A	TGR4C
	TGR4B	TGR4D

The buffer operation differs, depending on whether the TGR has been set as an input capture register or an output compare register.

When TGR Is an Output Compare Register: When a compare-match occurs, the corresponding channel buffer register value is transferred to the general register. Figure 12.17 shows an example.

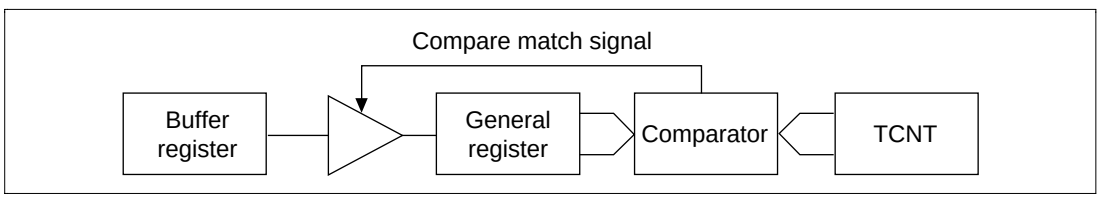


Figure 12.17 Compare Match Buffer Operation

When TGR Is an Input Capture Register: When an input capture occurs, the timer counter (TCNT) value is transferred to the general register (TGR), and the value that had been held up to that time in the TGR is transferred to the buffer register (figure 12.18).

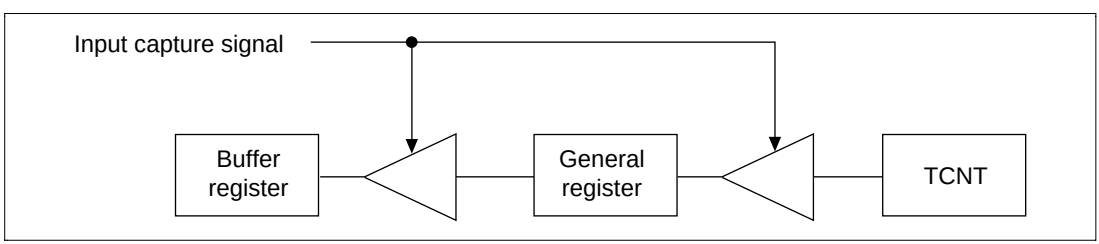


Figure 12.18 Input Capture Buffer Operation

Procedure for Setting Buffer Mode (Figure 12.19):

1. Use the timer I/O control register (TIOR) to set the TGR as either an input capture or output compare register.
2. Use the timer mode register (TMDR) BFA, and BFB bits to set the TGR for buffer mode.
3. Set the CST bit in the TSTR to 1 to start the count operation.

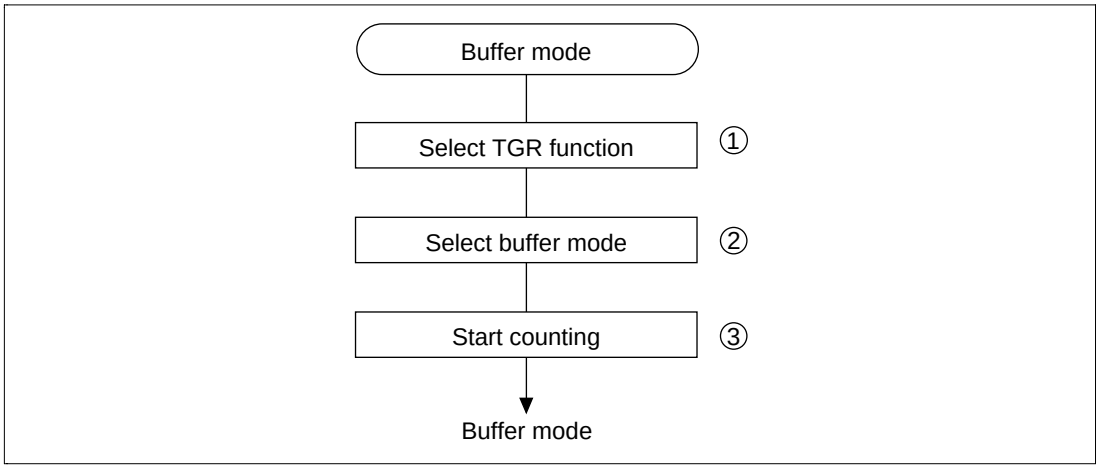


Figure 12.19 Buffer Operation Setting Procedure

Buffer Operation Examples—when TGR Is an Output Compare Register: Figure 12.20 shows an example of channel 0 set to PWM mode 1, and the TGRA and TGRC registers set for buffer operation.

The TCNT counter is cleared by a compare-match B, and the output is a 1 upon compare-match A and 0 output upon compare-match B. Because buffer mode is selected, a compare-match A changes the output, and the buffer register TGRC value is simultaneously transferred to the general register TGRA. This operation is repeated with each occurrence of a compare-match A.

See section 12.4.6, PWM Mode, for details on the PWM mode.

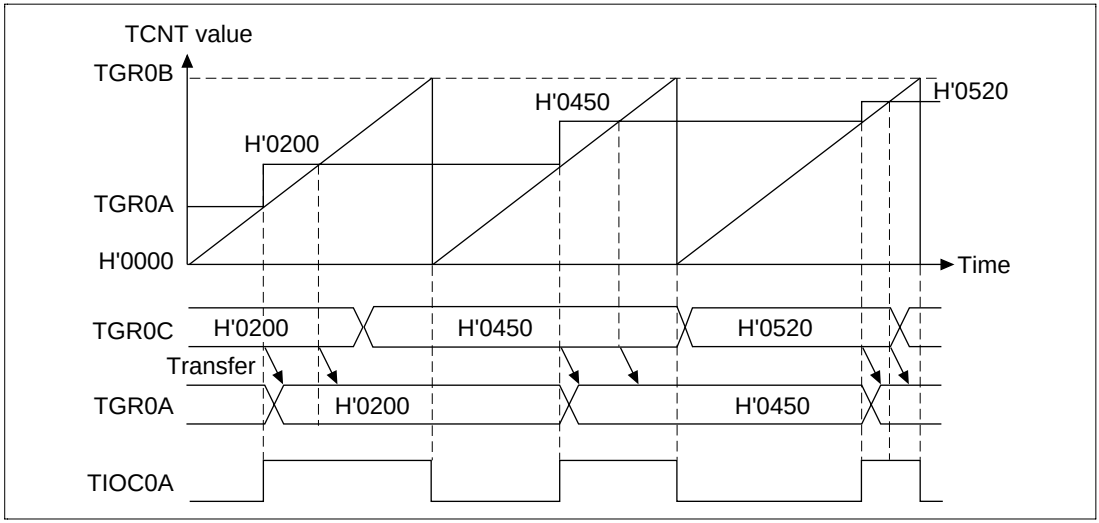


Figure 12.20 Buffer Operation Example (Output Compare Register)

Buffer Operation Examples—when TGR Is an Input Capture Register: Figure 12.21 shows an example of TGRA set as an input capture register with the TGRA and TGRB registers set for buffer operation.

The TCNT counter is cleared by a TGRA register input capture, and the TIOCA pin input capture input edge is selected as both rising and falling edge. Because buffer mode is selected, an input capture A causes the TCNT counter value to be stored in the TGRA register, and the value that was stored in the TGRA up until that time is simultaneously transferred to the TGRC register.

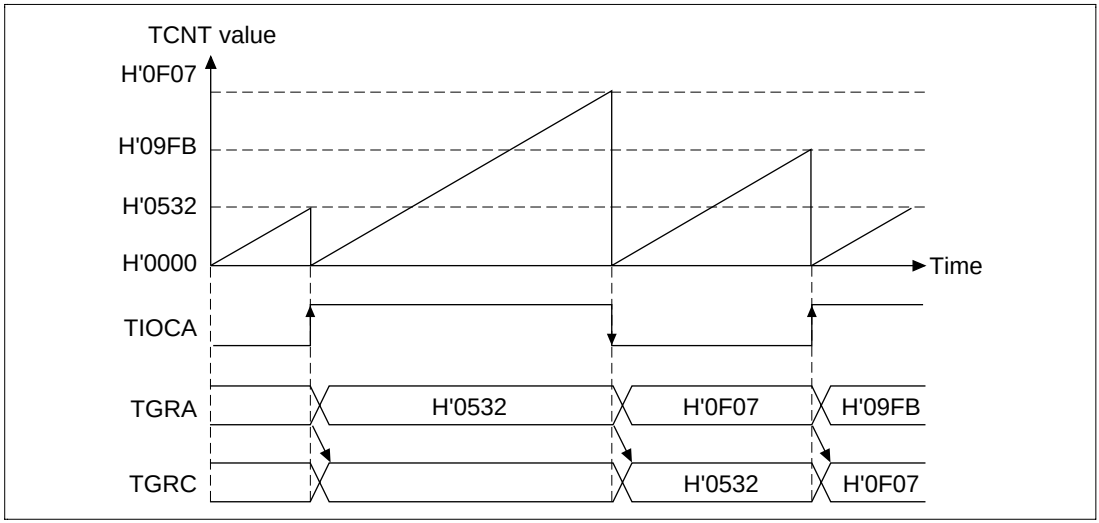


Figure 12.21 Buffer Operation Example (Input Capture Register)

12.4.5 Cascade Connection Mode

Cascade connection mode is a function that connects the 16-bit counters of two channels together to act as a 32-bit counter.

This function operates by using the TPSC2–TPSC0 bits of the TCR register to set the channel 1 counter clock to count by TCNT2 counter overflow/underflow.

Note: When channel 1 is set to phase counting mode, the counter clock settings become ineffective.

Table 12.6 shows the cascade connection combinations.

Table 12.6 Cascade Connection Combinations

Combination	Upper 16 Bits	Lower 16 Bits
Channel 1, channel 2	TCNT1	TCNT2

Procedure for Setting Cascade Connection Mode (Figure 12.22):

1. Set the TPSC2–TPSC 0 bits of the channel 1 timer control register (TCR) to B'111 to select "count by TCNT2 overflow/underflow."
2. Set the CST bits corresponding to the upper and lower 16 bits in the TSTR to 1 to start the count operation.

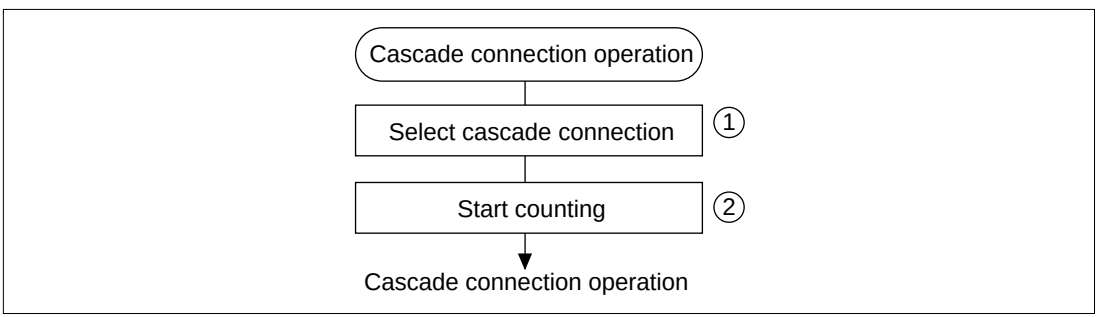


Figure 12.22 Procedure for Selecting Cascade Connection Mode

Cascade Connection Operation Examples—Phase Counting Mode: Figure 12.23 shows an example of operation when the TCNT1 counter is set to count on TCNT2 overflow/underflow and channel 2 is set to phase counting mode.

The TCNT1 counter increments with a TCNT2 counter overflow and decrements with a TCNT2 underflow.

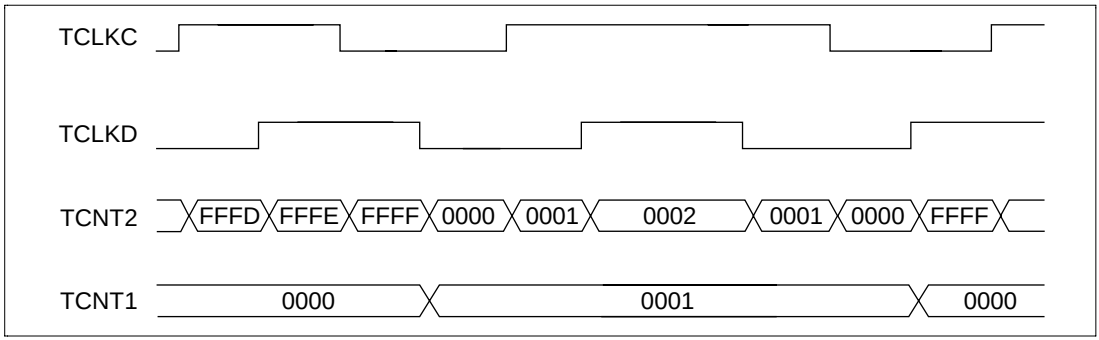


Figure 12.23 Cascade Connection Operation Example (Phase Counting Mode)

12.4.6 PWM Mode

PWM mode outputs the various PWM waveforms from output pins. Output levels of 0 output, 1 output, or toggle output can be selected as the output level for the compare-match of each TGR.

A period can be set for a register by using the TGR compare-match as a counter clear source. All five channels can be independently set to PWM mode. Synchronous operation is also possible.

There are two PWM modes:

- PWM mode 1

  Generates PWM output using the TGRA and TGRB registers, and TGRC and TGRD registers as pairs. The initial output values are those established in the TGRA and TGRC registers. When the values set in TGR registers being used as a pair are equal, output values will not change even if a compare-match occurs.

  A maximum of 8-phase PWM output is possible for PWM mode 1.
- PWM mode 2

  Generates PWM output using one TGR register as a period register and another as a duty cycle register. The output value of each pin upon a counter clear is the initial value established by the TIOR register. When the values set in the period register and duty register are equal, output values will not change even if a compare-match occurs. PWM mode 2 can be set only for channels 0, 1, and 2.

Table 12.7 lists the combinations of PWM output pins and registers.

Table 12.7 Combinations of PWM Output Pins and Registers

		Output Pin	
Channel	**Register**	**PWM Mode 1**	**PWM Mode 2**
0 (AB pair)	TGR0A TGR0B	TIOC0A	TIOC 0A TIOC 0B
0 (CD pair)	TGR0C TGR0D	TIOC0C	TIOC 0C TIOC 0D
1	TGR1A TGR1B	TIOC1A	TIOC 1A TIOC 1B
2	TGR2A TGR2B	TIOC2A	TIOC 2A TIOC 2B
3 (AB pair)	TGR3A TGR3B	TIOC3A	Setting not possible
3 (CD pair)	TGR3C TGR3D	TIOC3C	
4 (AB pair)	TGR4A TGR4B	TIOC4A	
4 (CD pair)	TGR4C TGR4D	TIOC4C	

Note: PWM output of the period setting TGR is not possible in PWM mode 2.

Procedure for Selecting the PWM Mode (Figure 12.24):

1. Set bits TPSC2–TPSC0 in the TCR to select the counter clock source. At the same time, set bits CKEG1 and CKEG0 in the TCR to select the desired edge of the input clock.
2. Set bits CCLR2–CCLR0 in the TCR to select the TGR to be used as a counter clear source.
3. Set the period in the TGR selected in step 2, and the duty cycle in another TGR.
4. Using the timer I/O control register (TIOR), set the TGR selected in step 3 to act as an output compare register, and select the initial value and output value.
5. Set the MD3–MD 0 bits in TMDR to select the PWM mode.
6. Set the CST bit in the TSTR to 1 to let the TCNT start counting.

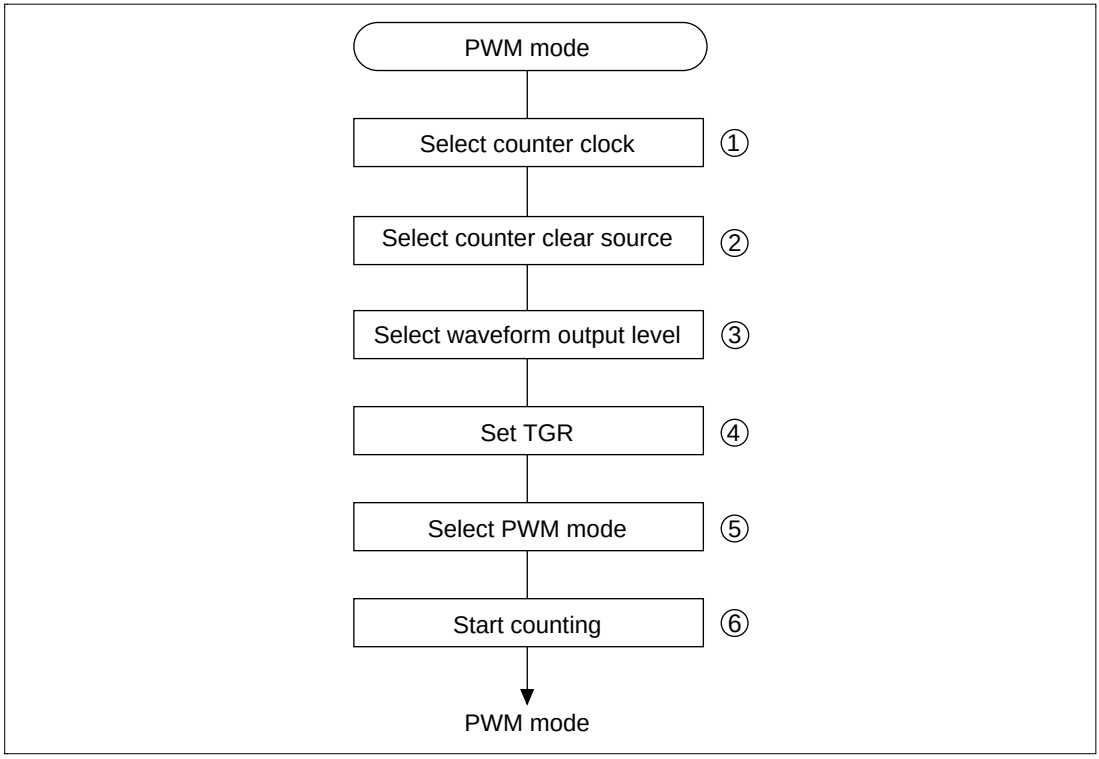


Figure 12.24 Procedure for Selecting the PWM Mode

PWM Mode Operation Examples—PWM Mode 1 (Figure 12.25): A TGRA register compare-match is used as a TCNT counter clear source, the TGRA register initial output value and output compare output value are both 0, and the TGRB register output compare output value is a 1. In this example, the value established in the TGRA register becomes the period and the value established in the TGRB register becomes the duty cycle.

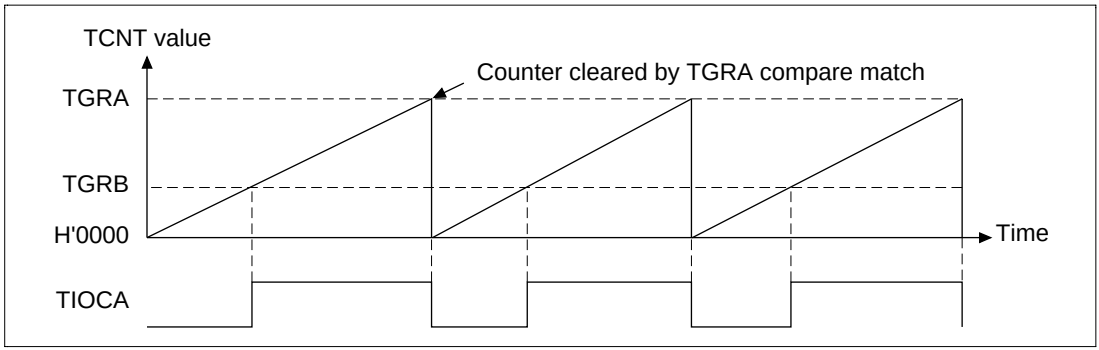


Figure 12.25 PWM Mode Operation Example (Mode 1)

PWM Mode Operation Examples—PWM Mode 2 (Figure 12.26): Channels 0 and 1 are set for synchronous operation, TGR1B register compare-match is used as a TCNT counter clear source, the other TGR register initial output value is 0 and output compare output value is 1, and a 5-phase PWM waveform is output. In this example, the value established in the TGR1B register becomes the period and the value established in the other TGR register becomes the duty cycle.

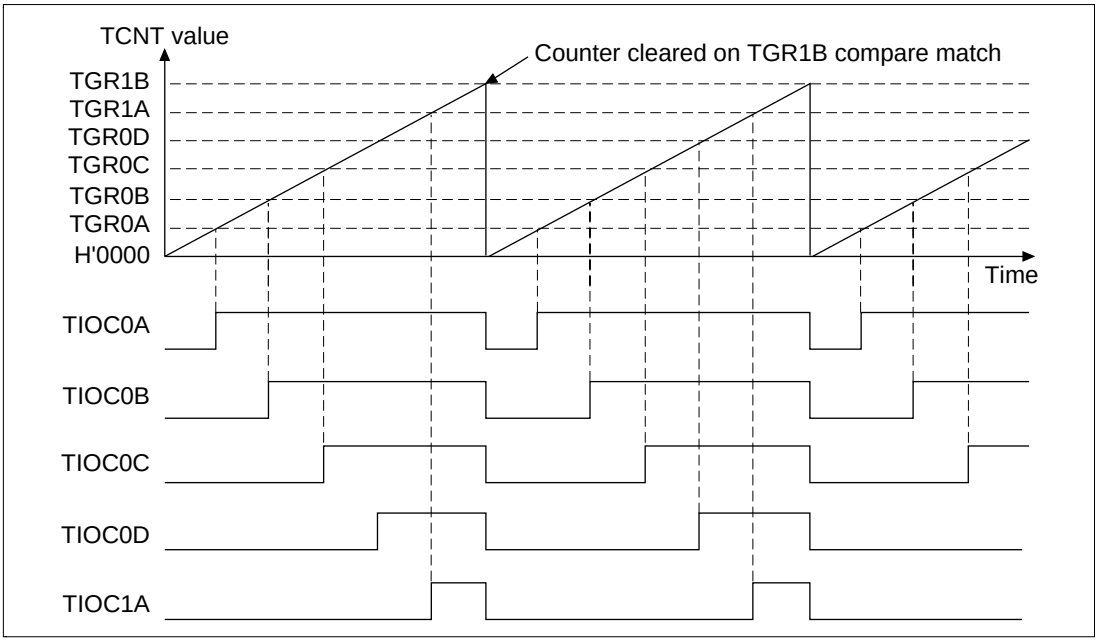


Figure 12.26 PWM Mode Operation Example (Mode 2)

0% Duty Cycle: Figure 12.27 shows an example of a 0% duty cycle PWM waveform output in PWM mode.

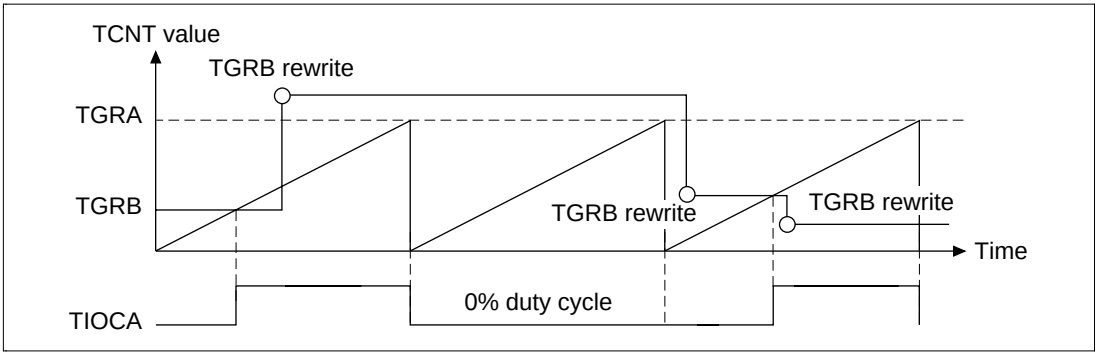


Figure 12.27 PWM Mode Operation Example (0% Duty Cycle)

100% Duty Cycle: Figure 12.28 shows an example of a 100% duty cycle PWM waveform output in PWM mode.

In PWM mode, when setting cycle = duty cycle the output waveform does not change, nor is there a change of waveform for the first pulse immediately after clearing the counter.

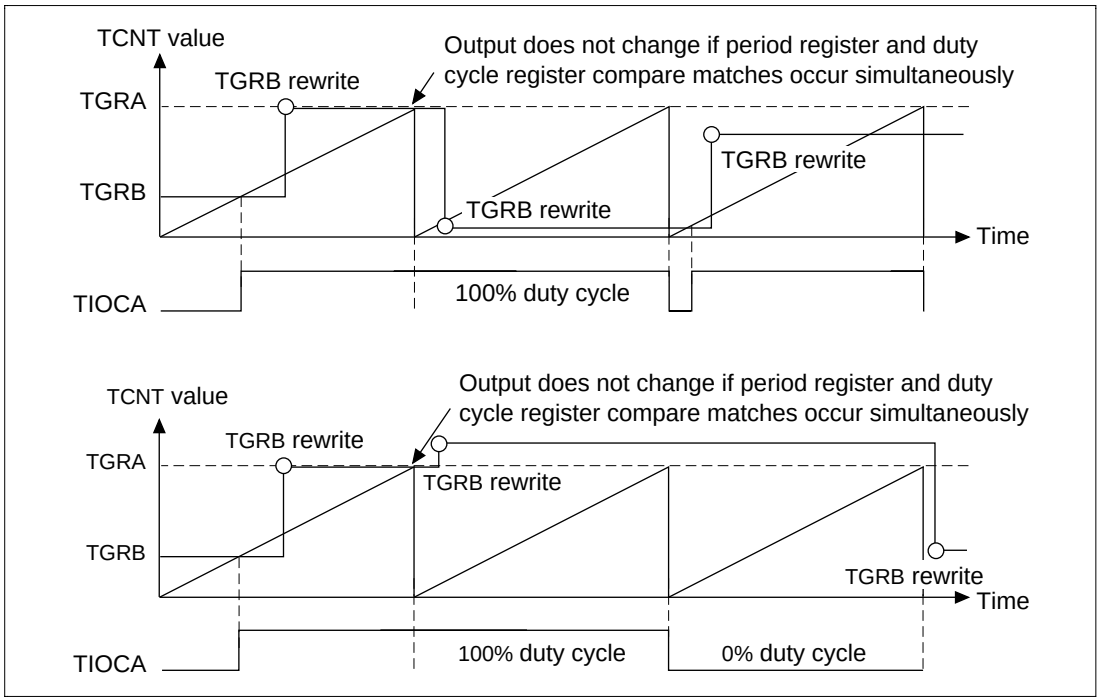


Figure 12.28 PWM Mode Operation Example (100% Duty Cycle)

12.4.7 Phase Counting Mode

The phase counting mode detects the phase differential of two external clock inputs and counts the TCNT counter up or down. This mode can be set for channels 1 and 2.

When set in the phase counting mode, an external clock is selected for the counter input clock, regardless of the settings of the TPSC2–TPSC0 bits of TCR or the CKEG1 and CKEG0 bits. TCNT also becomes an up/down counter. Since the TCR CCLR1/CCLR0 bits, TIOR, TIER, and TGR functions are all enabled, input capture and compare-match functions and interrupt sources can be used.

When the TCNT counter is incrementing, an overflow sets the TSR register TCFV (overflow flag). When it is decrementing, an underflow sets the TCFU (underflow flag).

The TSR register TCFD bit is a count direction flag. Read the TCFD flag to confirm whether the TCNT is incrementing or decrementing.

Table 12.8 shows the correspondence between channels and external clock pins.

Table 12.8 Phase Counting Mode Clock Input Pins

Channel	A Phase Input Pin	B Phase Input Pin
1	TCLKA	TCLKB
2	TCLKC	TCLKD

Procedure for Selecting the Phase Counting Mode (Figure 12.29):

1. Set the MD3–MD0 bits of the timer mode register (TMDR) to select the phase counting mode.
2. Set the CST bit of the timer start register (TSTR) to 1 to start the count.

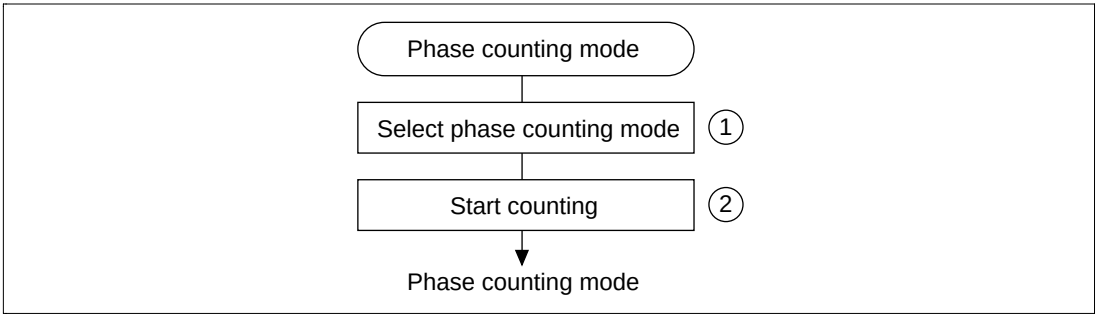


Figure 12.29 Procedure for Selecting the Phase Counting Mode

Phase Counting Operation Examples: The phase counting mode uses the phase difference between two external clocks to increment/decrement the TCNT counter. There are 4 modes, depending on the count conditions.

Phase Counting Mode 1: Figure 12.30 shows an example of phase counting mode 1 operation. Table 12.9 lists the up counting and down counting conditions for the TCNT.

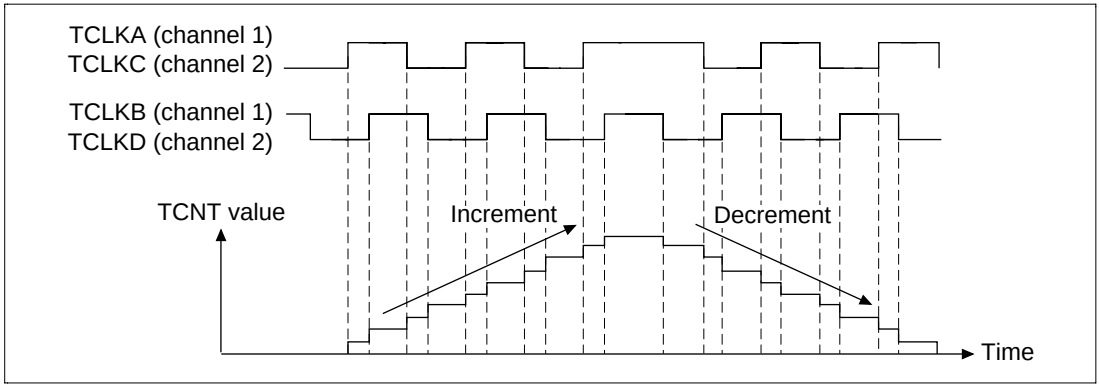


Figure 12.30 Phase Counting Mode 1 Operation

Table 12.9 Phase Count Mode 1 Up/Down Counting Conditions

TCLKA (Channel 1) TCLKC (Channel 2)	TCLKB (Channel 1) TCLKD (Channel 2)	Operation
1 (high level)	Rising edge	Increment
0 (low level)	Falling edge	
Rising edge	0 (low level)	
Falling edge	1 (high level)	
1 (high level)	Falling edge	Decrement
0 (low level)	Rising edge	
Rising edge	1 (high level)	
Falling edge	0 (low level)	

Phase Count Mode 2: Figure 12.31 shows an example of phase counting mode 2 operation. Table 12.10 lists the up counting and down counting conditions for the TCNT.

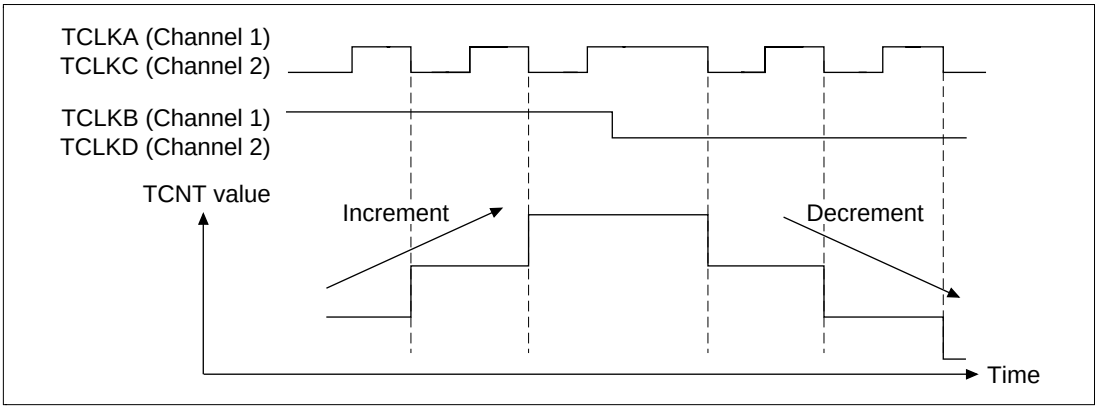


Figure 12.31 Phase Counting Mode 2 Operation

Table 12.10 Phase Count Mode 2 Up/Down Counting Conditions

TCLKA (Channel 1) TCLKC (Channel 2)	TCLKB (Channel 1) TCLKD (Channel 2)	Operation
1 (high level)	Rising edge	Does not count (don't care)
0 (low level)	Falling edge	Does not count (don't care)
Rising edge	0 (low level)	Does not count (don't care)
Falling edge	1 (high level)	Increment
1 (high level)	Falling edge	Does not count (don't care)
0 (low level)	Rising edge	Does not count (don't care)
Rising edge	1 (high level)	Does not count (don't care)
Falling edge	0 (low level)	Decrement

Phase Count Mode 3: Figure 12.32 shows an example of phase counting mode 3 operation. Table 12.11 lists the up counting and down counting conditions for the TCNT.

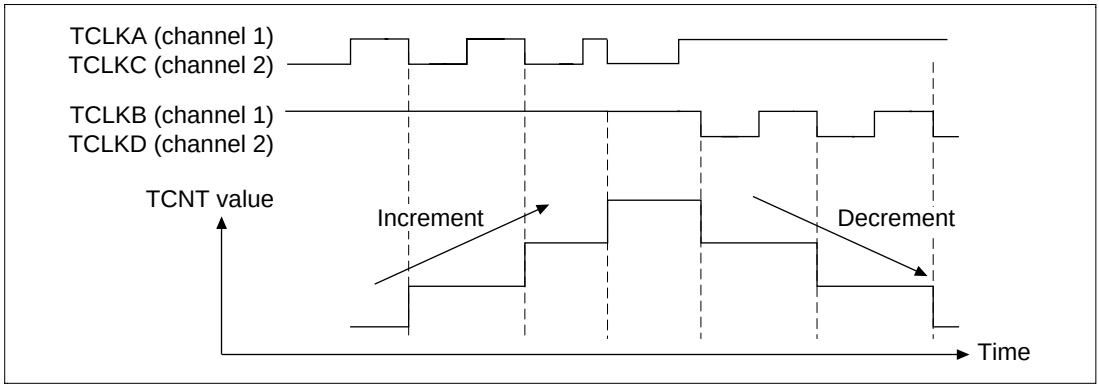


Figure 12.32 Phase Counting Mode 3 Operation

Table 12.11 Phase Count Mode 3 Up/Down Counting Conditions

TCLKA (Channel 1) TCLKC (Channel 2)	TCLKB (Channel 1) TCLKD (Channel 2)	Operation
1 (high level)	Rising edge	Does not count (don't care)
0 (low level)	Falling edge	Does not count (don't care)
Rising edge	0 (low level)	Does not count (don't care)
Falling edge	1 (high level)	Increment
1 (high level)	Falling edge	Decrement
0 (low level)	Rising edge	Does not count (don't care)
Rising edge	1 (high level)	Does not count (don't care)
Falling edge	0 (low level)	Does not count (don't care)

Phase Count Mode 4: Figure 12.33 shows an example of phase counting mode 4 operation. Table 12.12 lists the up counting and down counting conditions for the TCNT.

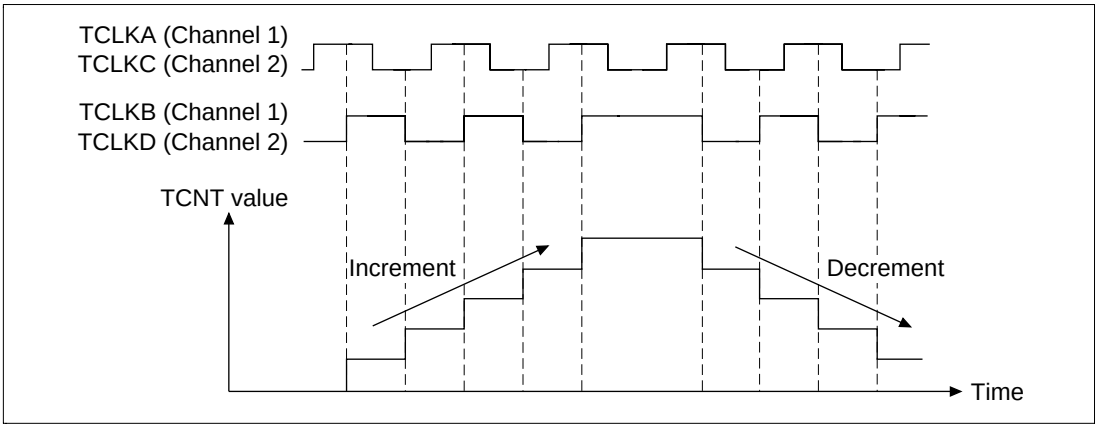


Figure 12.33 Phase Counting Mode 4 Operation

Table 12.12 Phase Count Mode 4 Up/Down Counting Conditions

TCLKA (Channel 1) TCLKC (Channel 2)	TCLKB (Channel 1) TCLKD (Channel 2)	Operation
1 (high level)	Rising edge	Increment
0 (low level)	Falling edge	
Rising edge	0 (low level)	Does not count (don't care)
Falling edge	1 (high level)	
1 (high level)	Falling edge	Decrement
0 (low level)	Rising edge	
Rising edge	1 (high level)	Does not count (don't care)
Falling edge	0 (low level)	

Phase Counting Mode Application Example: Figure 12.34 shows an example where channel 1 is set to phase counting mode and is teamed with channel 0 to input a two-phase encoder pulse for a servo motor to accurately detect position and speed.

Channel 1 is set to phase counting mode 1, and the encoder pulse A phase and B phase are input to the TCLKA and TCLKB pins.

Channel 0 is set so that the TCNT counter is cleared on a TGR0C register compare-match, and the TGR0A and TGR0C registers are used with the compare-match function to establish the speed control and position control periods. The TGR0B register is used with the input capture function, and the TGR0B and TGR0D registers are employed for buffer operation. The channel 1 counter

input clock is used as the TGR0B register input capture source, and a pulse width of four times the 2-phase encoder pulse is detected.

The channel 1 TGR1A and TGR1B registers are set for the input capture function, the channel 0 TGR0A and TGR0C register compare-match is used as an input capture source, and all of the control period increment and decrement values are stored.

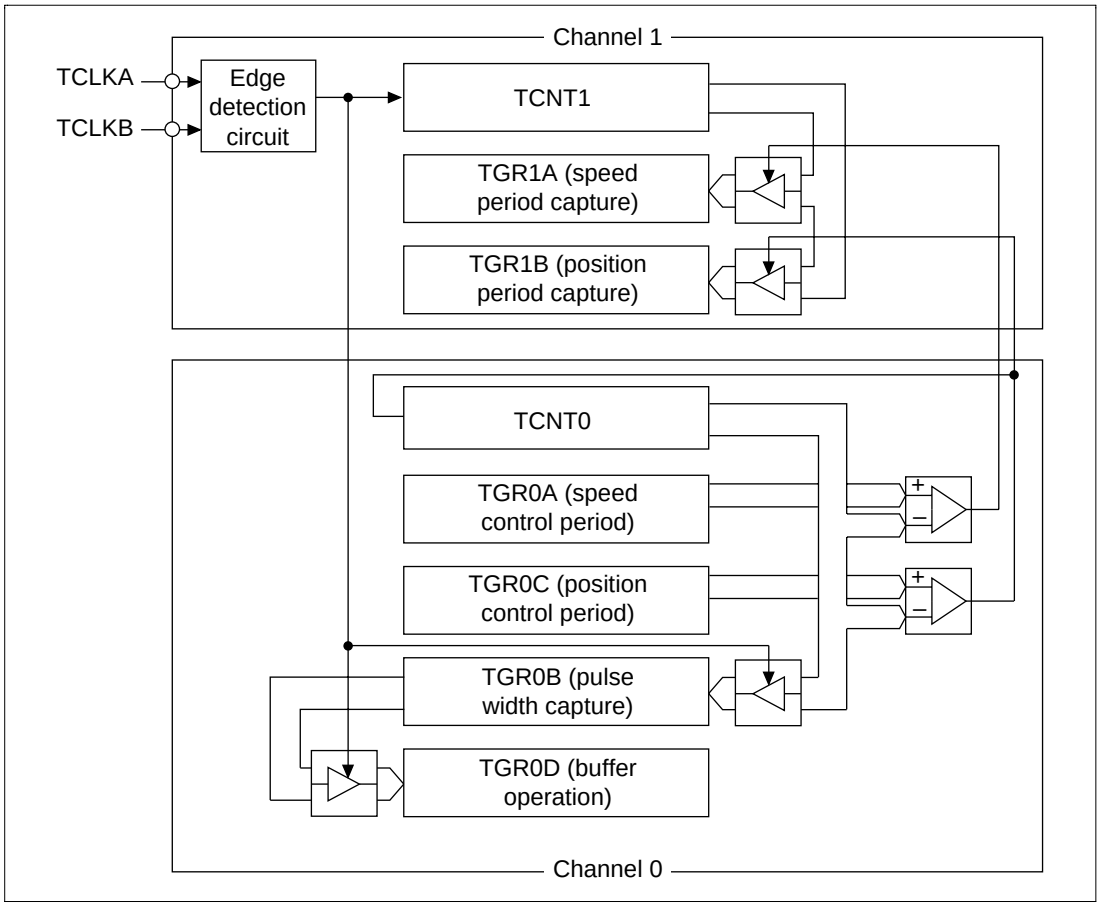


Figure 12.34 Phase Count Mode Application Example

12.4.8 Reset-Synchronized PWM Mode

In the reset-synchronized PWM mode, three-phase output of positive and negative PWM waveforms that share a common wave transition point can be obtained using channels 3 and 4.

When set for reset-synchronized PWM mode, the TIOC3B, TIOC3D, TIOC4A, TIOC4C, TIOC4B, and TIOC4D pins become PWM output pins and TCNT3 becomes an upcounter.

Table 12.13 shows the PWM output pins used. Table 12.14 shows the settings of the registers.

Table 12.13 Output Pins for Reset-Synchronized PWM Mode

Channel	Output Pin	Description
3	TIOC3B	PWM output 1
	TIOC3D	PWM output 1' (negative-phase waveform of PWM output 1)
4	TIOC4A	PWM output 2
	TIOC4C	PWM output 2' (negative-phase waveform of PWM output 2)
	TIOC4B	PWM output 3
	TIOC4D	PWM output 3' (negative-phase waveform of PWM output 3)

Table 12.14 Register Settings for Reset-Synchronized PWM Mode

Register	Description of Contents
TCNT3	Initial setting of H'0000
TCNT4	Initial setting of H'0000
TGR3A	Set count cycle for TCNT3
TGR3B	Sets the turning point for PWM waveform output by the TIOC3B and TIOC3D pins
TGR4A	Sets the turning point for PWM waveform output by the TIOC4A and TIOC4C pins
TGR4B	Sets the turning point for PWM waveform output by the TIOC4B and TIOC4D pins

Procedure for Selecting the Reset-Synchronized PWM Mode (Figure 12.35):

1. Clear the CST3 and CST4 bits in the TSTR to 0 to halt TCNT3 and TCNT4. The reset-synchronized PWM mode must be set up while TCNT3 and TCNT4 are halted.
2. Set bits TPSC2–TPSC0 and CKEG1 and CKEG0 in the TCR to select the counter clock and clock edge for channel 3.
3. Set bits CCLR2–CCLR0 in the TCR3 to select TGRA compare-match as a counter clear source.

4. When performing brushless DC motor control, set bit BDC in the timer gate control register (TGCR) and set the feedback signal input source and output chopping or gate signal direct output.
5. Reset TCNT3 and TCNT4 to H'0000.
6. TGR3A is the period register. Set the waveform period value in TGR3A. Set the transition times of the PWM output waveforms in TGR3B, TGR4A, and TGR4B. Set times within the compare-match range of TCNT3. X ≤ TGR3A (X: set value). With X = TGRA (cycle = duty cycle), the output waveform goes into toggle operation at the point where TCNT3 = TGR3A = X.
7. Select enabling/disabling of toggle output synchronized with the PMW cycle using bit PSYE in the timer output control register (TOCR), and set the PWM output level with bits OLSP and OLSN.
8. Set bits MD3–MD0 in TMDR3 to B'1000 to select the reset-synchronized PWM mode. TIOC3B, TIOC3D, TIOC4A, TIOC4B, TIOC4C, and TIOC4D become PWM output pins.
9. Set the CST3 bit in the TSTR to 1 to start the count operation.
10. Set the STR3 bit in the TSTR to 1 to let the TCNT3 start counting.

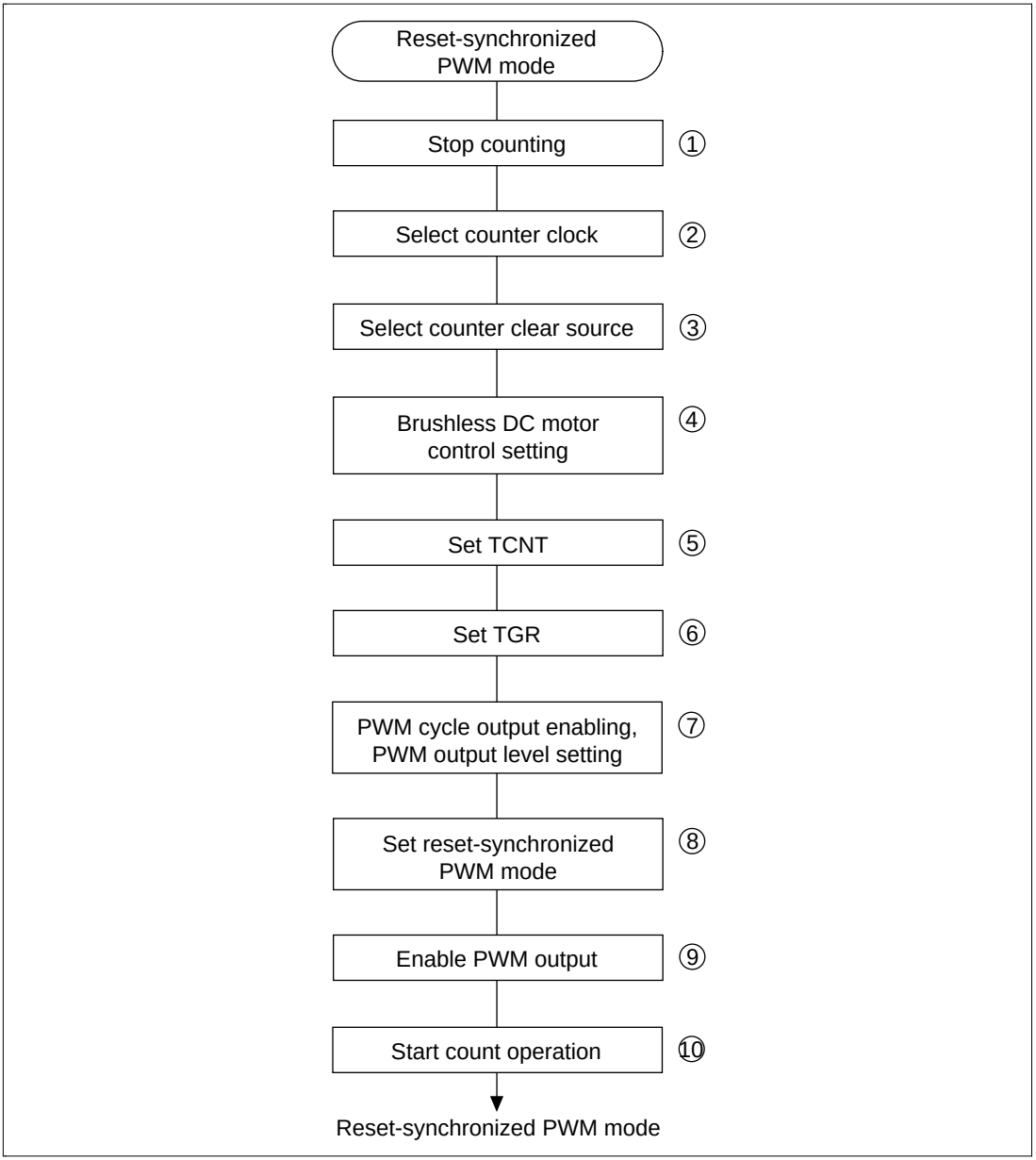


Figure 12.35 Procedure for Selecting the Reset-Synchronized PWM Mode

Reset-Synchronized PWM Mode Operation: Figure 12.36 shows an example of operation in the reset-synchronized PWM mode. TCNT3 and TCNT4 operate as upcounters. The counter is cleared when a TCNT3 and TGR3A compare-match occurs, and then begins incrementing from H'0000. The PWM output pin output toggles with each occurrence of a TGR3B, TGR4A, TGR4B compare-match, and upon counter clears.

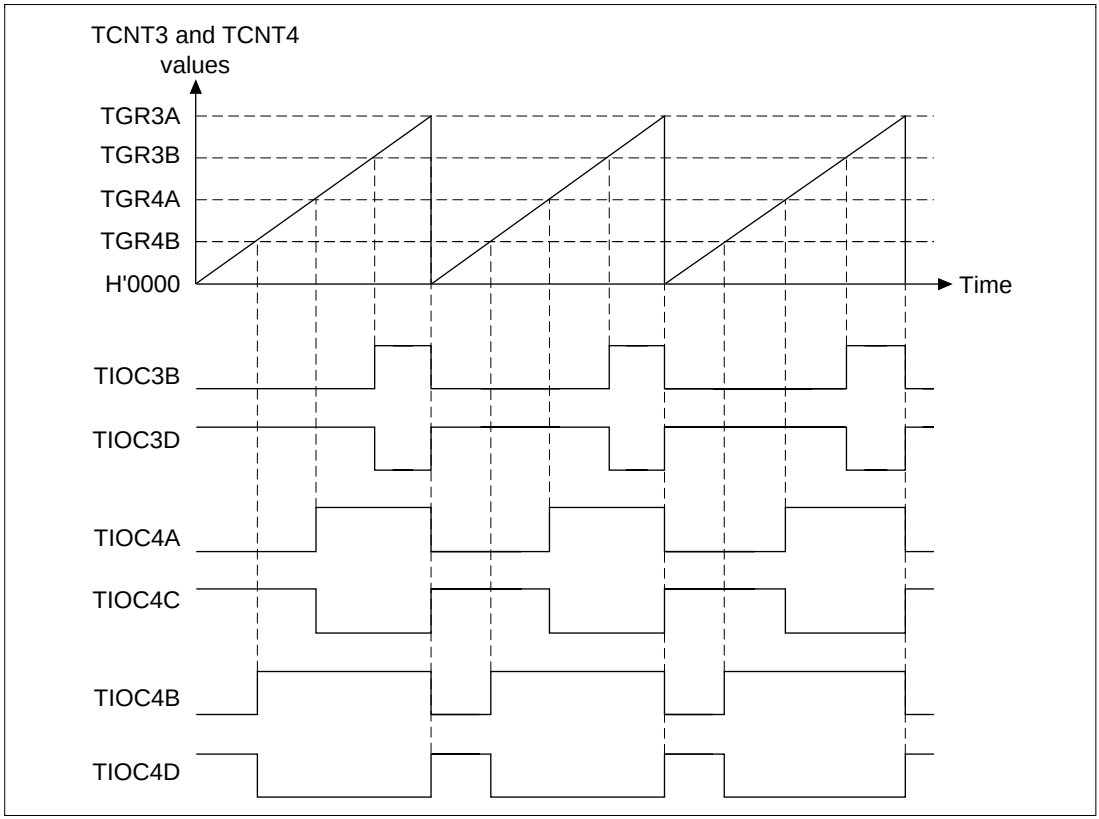


Figure 12.36 Reset-Synchronized PWM Mode Operation Example (When the TOCR's OLSN = 1 and OLSP = 1)

12.4.9 Complementary PWM Mode

In the complementary PWM mode, three-phase output of non-overlapping positive and negative PWM waveforms can be obtained using channels 3 and 4.

In complementary PWM mode, TIOC3B, TIOC3D, TIOC4A, TIOC4B, TIOC4C, and TIOC4D pins become PWM output pins, the TIOC3A pin can be set for toggle output synchronized with the PWM period. TCNT3 and TCNT4 function as increment/decrement counters.

Table 12.15 shows the PWM output pins used. Table 12.16 shows the settings of the registers.

A function to directly cut off the PWM output by using an external signal is supported as a port function.

Table 12.15 Output Pins for Complementary PWM Mode

Channel	Output Pin	Description
3	TIOC3A	Toggle output synchronized with PWM period (or I/O port)
	TIOC3B	PWM output 1
	TIOC3C	I/O port (Avoid setting this pin as a timer I/O pin in the complementary PWM mode.)
	TIOC3D	$\overline{\text{PWM output 1}}$ (non-overlapping negative-phase waveform of PWM output 1)
4	TIOC4A	PWM output 2
	TIOC4B	PWM output 3
	TIOC4C	$\overline{\text{PWM output 2}}$ (non-overlapping negative-phase waveform of PWM output 2)
	TIOC4D	$\overline{\text{PWM output 3}}$ (non-overlapping negative-phase waveform of PWM output 3)

Table 12.16 Register Settings for Complementary PWM Mode

Channel	Counter/Register	Description	Read/Write from CPU
3	TCNT3	Start of up-count from value set in dead time register	Maskable by BSC/BCR1 setting*
	TGR3A	Set TCNT3 upper limit value (1/2 carrier cycle + dead time)	Maskable by BSC/BCR1 setting*
	TGR3B	PWM output 1 compare register	Maskable by BSC/BCR1 setting*
	TGR3C	TGR3A buffer register	Always readable/writable
	TGR3D	PWM output 1/TGR3B buffer register	Always readable/writable
4	TCNT4	Up-count start, initialized to H'0000	Maskable by BSC/BCR1 setting*
	TGR4A	PWM output 2 compare register	Maskable by BSC/BCR1 setting*
	TGR4B	PWM output 3 compare register	Maskable by BSC/BCR1 setting*
	TGR4C	PWM output 2/TGR4A buffer register	Always readable/writable
	TGR4D	PWM output 3/TGR4B buffer register	Always readable/writable
Timer dead time data register (TDDR)		Set TCNT4 and TCNT3 offset value (dead time value)	Maskable by BSC/BCR1 setting*
Timer cycle data register (TCDR)		Set TCNT4 upper limit value (1/2 carrier cycle)	Maskable by BSC/BCR1 setting*
Timer cycle buffer register (TCBR)		TCDR buffer register	Always readable/writable
Subcounter (TCNTS)		Subcounter for dead time generation	Read-only
Temporary register 1 (TEMP1)		PWM output 1/TGR3B temporary register	Not readable/writable
Temporary register 2 (TEMP2)		PWM output 2/TGR4A temporary register	Not readable/writable
Temporary register 3 (TEMP3)		PWM output 3/TGR4B temporary register	Not readable/writable

Note: * Access can be enabled or disabled according to the setting of bit 13 (MTURWE) in BSC/BCR1 (bus controller/bus control register 1).

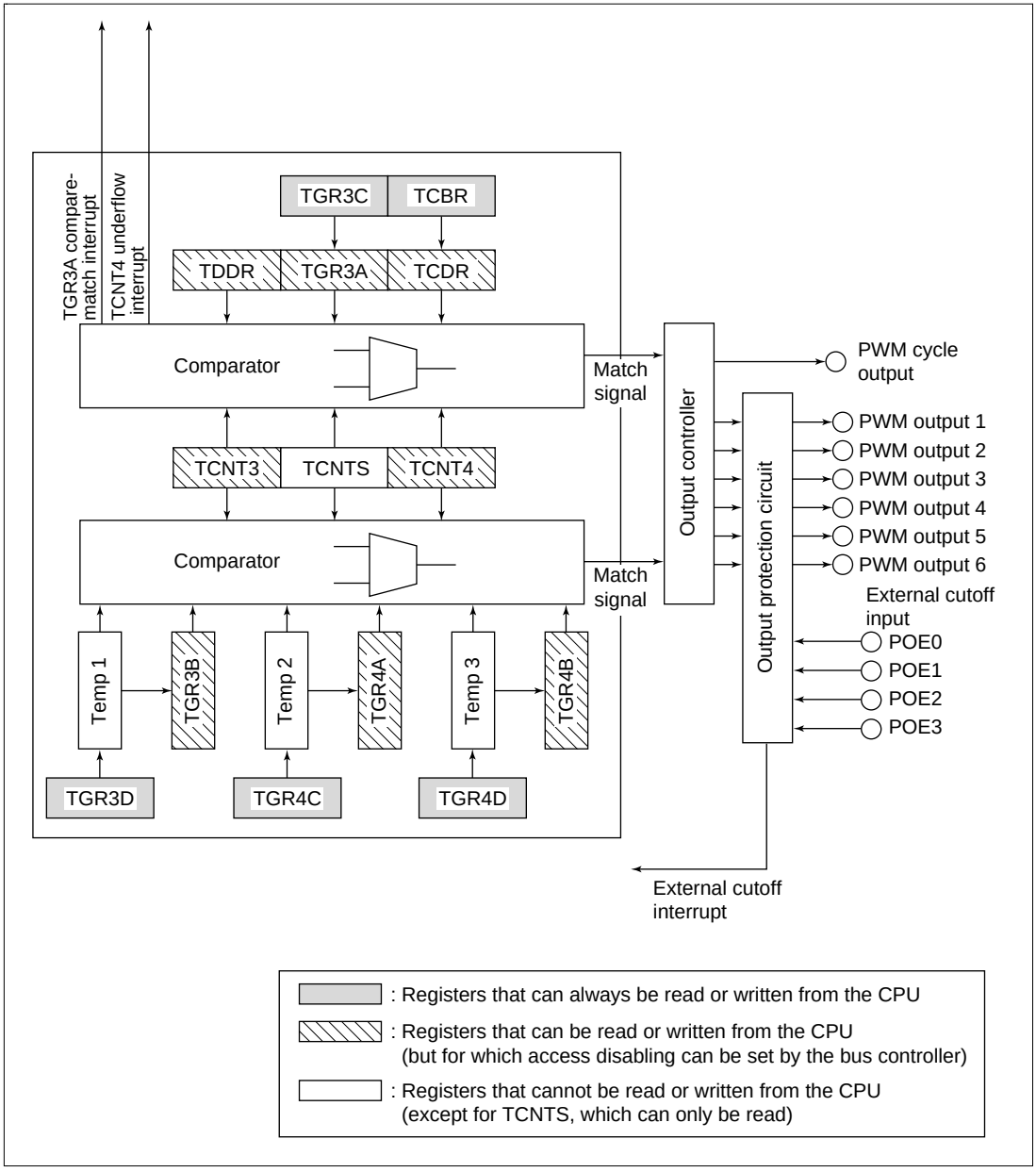


Figure 12.37 Block Diagram of Channels 3 and 4 in Complementary PWM Mode

Example of Complementary PWM Mode Setting Procedure: An example of the complementary PWM mode setting procedure is shown in figure 12.38.

1. Clear bits CST3 and CST4 in the timer start register (TSTR) to 0, and halt timer counter (TCNT) operation. Perform complementary PWM mode setting when TCNT3 and TCNT4 are stopped.
2. Set the same counter clock and clock edge for channels 3 and 4 with bits TPSC2–TPSC0 and bits CKEG1 and CKEG0 in the timer control register (TCR). Use bits CCLR2–CCLR0 to set synchronous clearing only when restarting by a synchronous clear from another channel during complementary PWM mode operation.
3. When performing brushless DC motor control, set bit BDC in the timer gate control register (TGCR) and set the feedback signal input source and output chopping or gate signal direct output.
4. Set the dead time in TCNT3. Set TCNT4 to H'0000.
5. Set only when restarting by a synchronous clear from another channel during complementary PWM mode operation. In this case, synchronize the channel generating the synchronous clear with channels 3 and 4 using the timer synchro register (TSYR).
6. Set the output PWM duty in the duty registers (TGR3B, TGR4A, TGR4B) and buffer registers (TGR3D, TGR4C, TGR4D). Set the same initial value in each corresponding TGR.
7. Set the dead time in the dead time register (TDDR), 1/2 the carrier cycle in the carrier cycle data register (TCDR) and carrier cycle buffer register (TCBR), and 1/2 the carrier cycle plus the dead time in TGR3A and TGR3C.
8. Select enabling/disabling of toggle output synchronized with the PWM cycle using bit PSYE in the timer output control register (TOCR), and set the PWM output level with bits OLSP and OLSN.
9. Select complementary PWM mode in timer mode register 3 (TMDR3). Pins TIOC3A, TIOC3B, TIOC3D, TIOC4A, TIOC4B, TIOC4C, and TIOC4D function as output pins. Do not set in TMDR4.
10. Set enabling/disabling of PWM waveform output pin output in the timer output master enable register (TOER).
11. Set bits CST3 and CST4 in TSTR to 1 simultaneously to start the count operation.

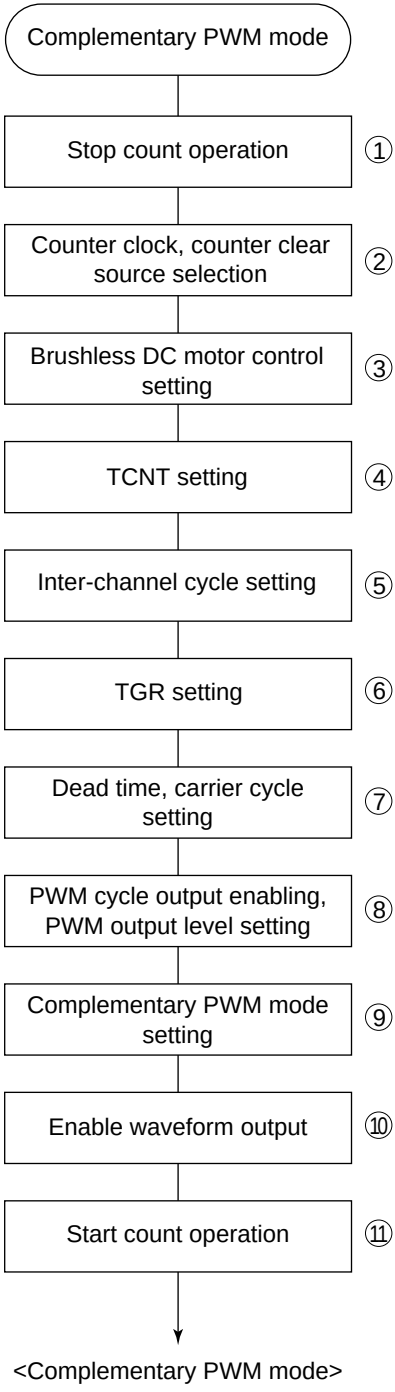


Figure 12.38 Example of Complementary PWM Mode Setting Procedure

Outline of Complementary PWM Mode Operation: In complementary PWM mode, 6-phase PWM output is possible. Figure 12.39 illustrates counter operation in complementary PWM mode, and figure 12.40 shows an example of complementary PWM mode operation.

- Counter operation

  In complementary PWM mode, three counters—TCNT3, TCNT4, and TCNTS—perform up/down-count operations.

  TCNT3 is automatically initialized to the value set in TDDR when complementary PWM mode is selected and the CST bit in TSTR is 0.

  When the CST bit is set to 1, TCNT3 counts up to the value set in TGR3A, then switches to down-counting when it matches TGR3A,. When the TCNT3 value matches TDDR, the counter switches to up-counting, and the operation is repeated in this way.

  TCNT4 is initialized to H'0000.

  When the CST bit is set to 1, TCNT4 counts up in synchronization with TCNT3, and switches to down-counting when it matches TCDR . On reaching H'0000, TCNT4 switches to up-counting, and the operation is repeated in this way.

  TCNTS is a read-only counter. It need not be initialized.

  When TCNT3 matches TCDR during TCNT3 and TCNT4 up/down-counting, down-counting is started, and when TCNTS matches TCDR, the operation switches to up-counting. When TCNTS matches TGR3A, it is cleared to H'0000.

  When TCNT4 matches TDDR during TCNT3 and TCNT4 down-counting, up-counting is started, and when TCNTS matches TDDR, the operation switches to down-counting. When TCNTS reaches H'0000, it is set with the value in TGR3A.

  TCNTS is compared with the compare register and temporary register in which the PWM duty is set during the count operation only.

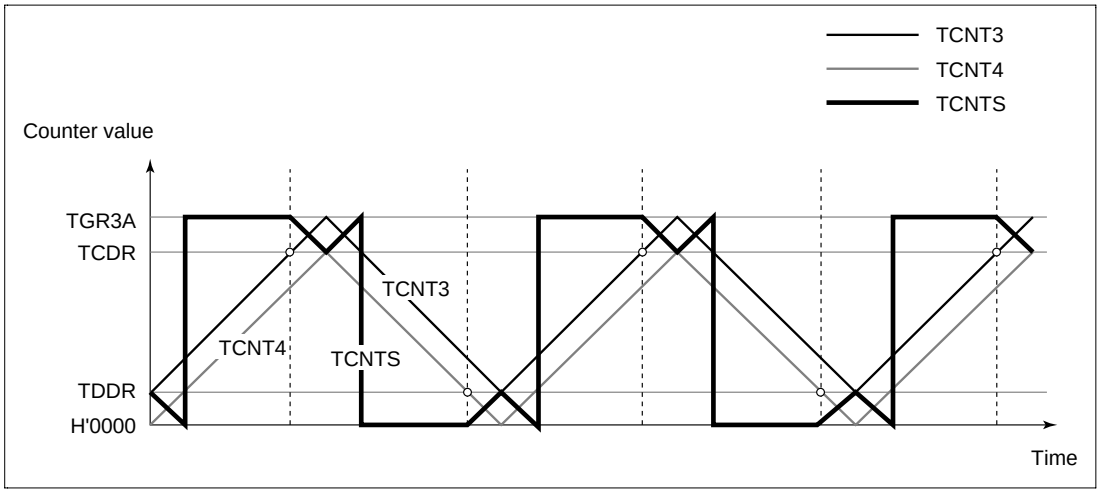


Figure 12.39 Complementary PWM Mode Counter Operation

- Register operation

  In complementary PWM mode, nine registers are used, comprising compare registers, buffer registers, and temporary registers. Figure 12.40 shows an example of complementary PWM mode operation.

  The registers which are constantly compared with the counters to perform PWM output are TGR3B, TGR4A, and TGR4B. When these registers match the counter, the value set in bits OLSN and OLSP in the timer output control register (TOCR) is output.

  The buffer registers for these compare registers are TGR3D, TGR4C, and TGR4D.

  Between a buffer register and compare register there is a temporary register. The temporary registers cannot be accessed by the CPU.

  Data in a compare register is changed by writing the new data to the corresponding buffer register. The buffer registers can be read or written at any time.

  The data written to a buffer register is constantly transferred to the temporary register in the Ta interval. Data is not transferred to the temporary register in the Tb interval. Data written to a buffer register in this interval is transferred to the temporary register at the end of the Tb interval.

  The value transferred to a temporary register is transferred to the compare register when TCNTS for which the Tb interval ends matches TGR3A when counting up, or H'0000 when counting down. The timing for transfer from the temporary register to the compare register can be selected with bits MD3–MD0 in the timer mode register (TMDR). Figure 12.40 shows an example in which the mode is selected in which the change is made in the trough.

  In the tb interval (tb2 in figure 12.40) in which data transfer to the temporary register is not performed, the temporary register has the same function as the compare register, and is compared with the counter. In this interval, therefore, there are two compare registers for one-phase output, with the compare register containing the pre-change data, and the temporary register containing the new data. In this interval, the three counters—TCNT3, TCNT4, and TCNTS—and two registers—compare register and temporary register—are compared, and PWM output controlled accordingly.

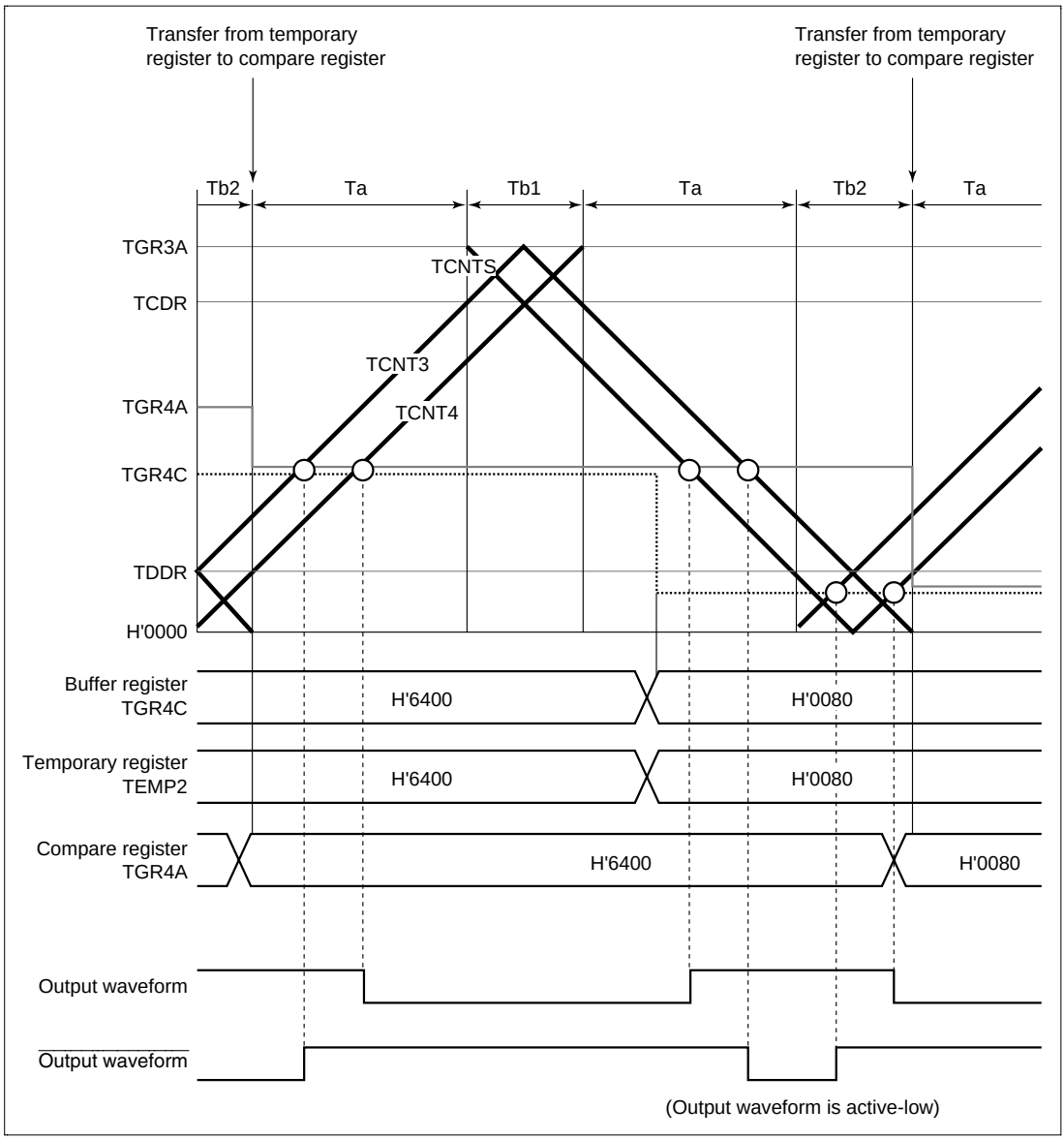


Figure 12.40 Example of Complementary PWM Mode Operation

- Initialization

  In complementary PWM mode, there are six registers that must be initialized.

  Before setting complementary PWM mode with bits MD3–MD0 in the timer mode register (TMDR), the following initial register values must be set.

  TGR3C operates as the buffer register for TGR3A, and should be set with 1/2 the PWM carrier cycle + dead time Td. The timer cycle buffer register (TCBR) operates as the buffer register for the timer cycle data register (TCDR), and should be set with 1/2 the PWM carrier cycle. Set dead time Td in the timer dead time data register (TDDR).

  Set the respective initial PWM duty values in buffer registers TGR3D, TGR4C, and TGR4D.

  The values set in the five buffer registers excluding TDDR are transferred simultaneously to the corresponding compare registers when complementary PWM mode is set.

  Set TCNT4 to H'0000 before setting complementary PWM mode.

Table 12.17 Registers and Counters Requiring Initialization

Register/Counter	Set Value
TGR3C	1/2 PWM carrier cycle + dead time Td
TDDR	Dead time Td
TCBR	1/2 PWM carrier cycle
TGR3D, TGR4C, TGR4D	Initial PWM duty value for each phase
TCNT4	H'0000

Note: The TGR3C set value must be the sum of 1/2 the PWM carrier cycle set in TCBR and dead time Td set in TDDR.

- PWM output level setting

  In complementary PWM mode, the PWM pulse output level is set with bits OLSN and OLSP in the timer output control register (TOCR).

  The output level can be set for each of the three positive phases and three negative phases of 6-phase output.

  Complementary PWM mode should be cleared before setting or changing output levels.

- Dead time setting

  In complementary PWM mode, PWM pulses are output with a non-overlapping relationship between the positive and negative phases. This non-overlap time is called the dead time.

  The non-overlap time is set in the timer dead time data register (TDDR). The value set in TDDR is used as the TCNT3 counter start value, and creates non-overlap between TCNT3 and TCNT4. Complementary PWM mode should be cleared before changing the contents of TDDR.

- PWM cycle setting

  In complementary PWM mode, the PWM pulse cycle is set in two registers—TGR3A, in which the TCNT3 upper limit value is set, and TCDR, in which the TCNT4 upper limit value is set. The settings should be made so as to achieve the following relationship between these two registers:

  TGR3A set value = TCDR set value + TDDR set value

  The TGR3A and TCDR settings are made by setting the values in buffer registers TGR3C and TCBR. The values set in TGR3C and TCBR are transferred simultaneously to TGR3A and TCDR in accordance with the transfer timing selected with bits MD3–MD0 in the timer mode register (TMDR).

  The updated PWM cycle is reflected from the next cycle when the data update is performed at the crest, and from the current cycle when performed in the trough. Figure 12.41 illustrates the operation when the PWM cycle is updated at the crest.

  See the following section, Register data updating, for the method of updating the data in each buffer register.

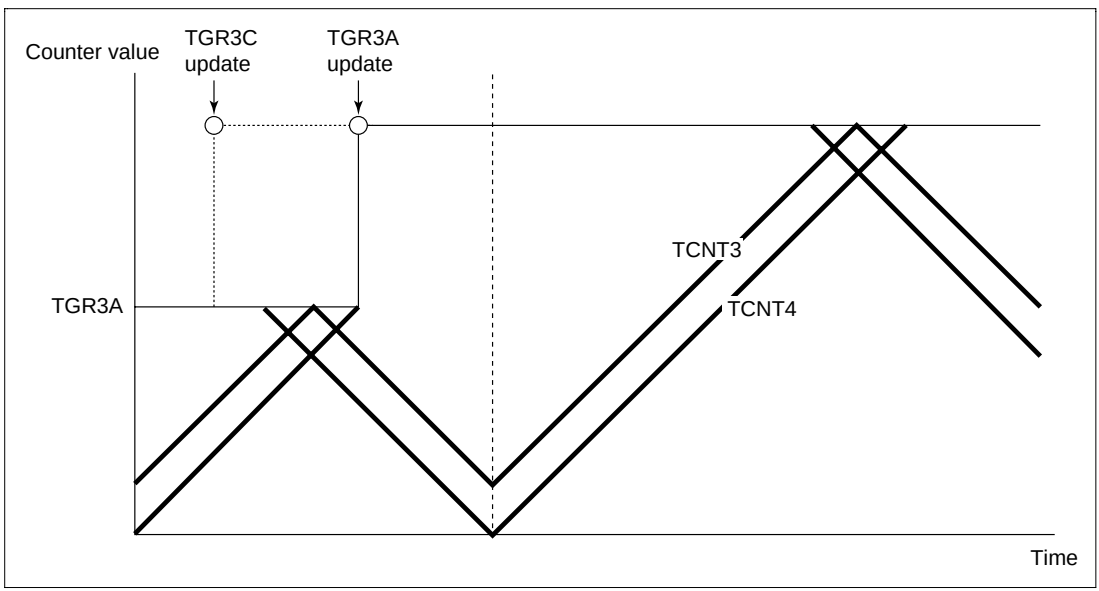


Figure 12.41 Example of PWM Cycle Updating

- Register data updating

  In complementary PWM mode, the buffer register is used to update the data in a compare register. The update data can be written to the buffer register at any time. There are five PWM duty and carrier cycle registers that have buffer registers and can be updated during operation.

  There is a temporary register between each of these registers and its buffer register. When subcounter TCNTS is not counting, if buffer register data is updated, the temporary register value is also rewritten. Transfer is not performed from buffer registers to temporary registers when TCNTS is counting; in this case, the value written to a buffer register is transferred after TCNTS halts.

  The temporary register value is transferred to the compare register at the data update timing set with bits MD3–MD0 in the timer mode register (TMDR). Figure 12.42 shows an example of data updating in complementary PWM mode. This example shows the mode in which data updating is performed at both the counter crest and trough.

  When rewriting buffer register data, a write to TGR4D must be performed at the end of the update. Data transfer from the buffer registers to the temporary registers is performed simultaneously for all five registers after the write to TGR4D.

  A write to TGR4D must be performed after writing data to the registers to be updated, even when not updating all five registers, or when updating the TGR4D data. In this case, the data written to TGR4D should be the same as the data prior to the write operation.

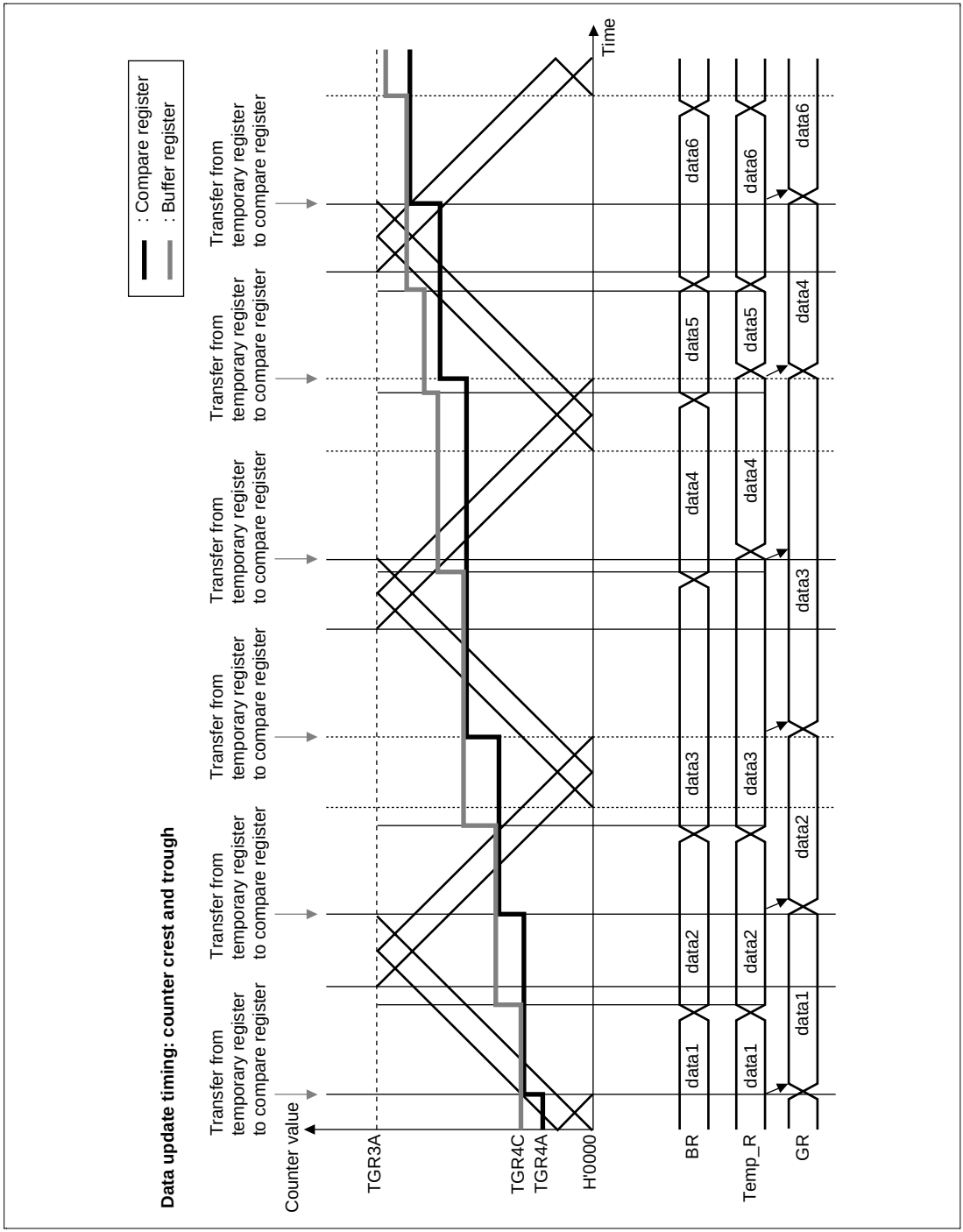


Figure 12.42 Example of Data Update in Complementary PWM Mode

- Initial output in complementary PWM mode

  In complementary PWM mode, the initial output is determined by the setting of bits OLSN and OLSP in the timer output control register (TOCR).

  This initial output is the PWM pulse non-active level, and is output from when complementary PWM mode is set with the timer mode register (TMDR) until TCNT4 exceeds the value set in the dead time register (TDDR). Figure 12.43 shows an example of the initial output in complementary PWM mode.

  An example of the waveform when the initial PWM duty value is smaller than the TDDR value is shown in figure 12.44.

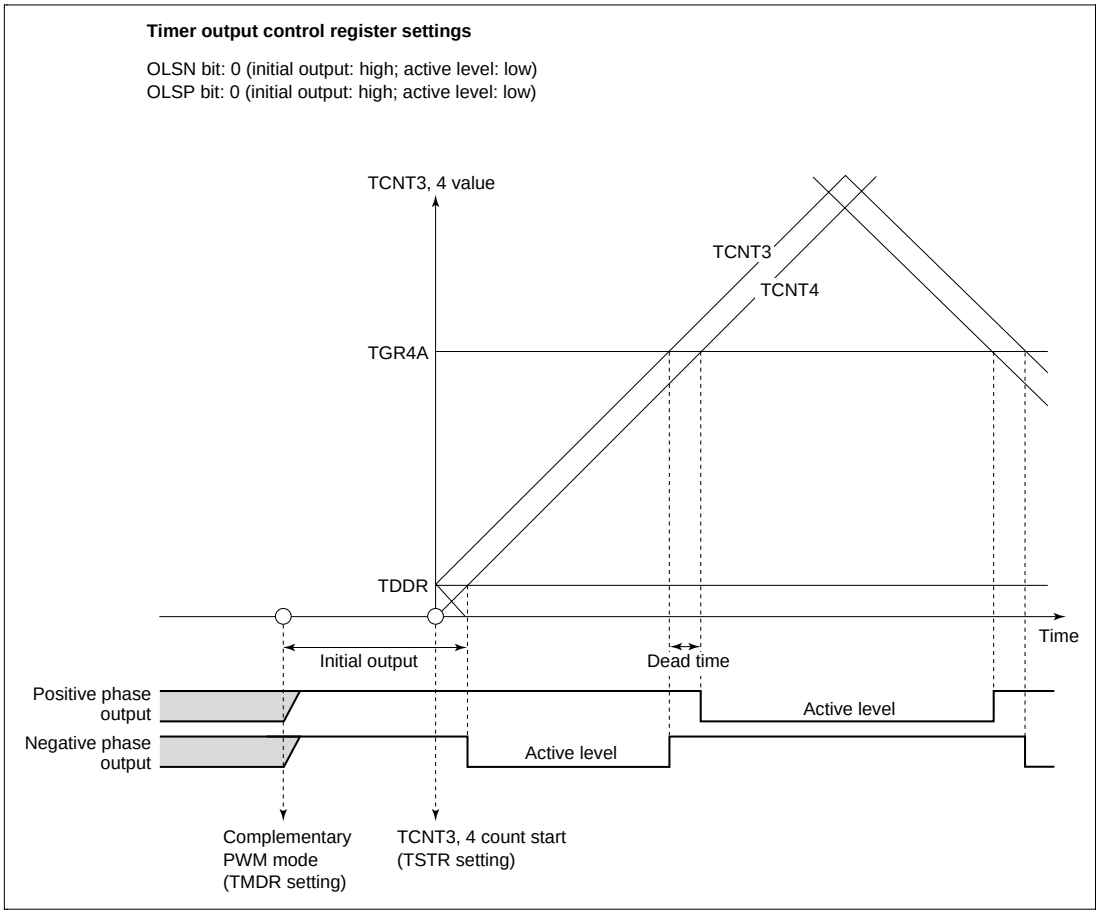


Figure 12.43 Example of Initial Output in Complementary PWM Mode (1)

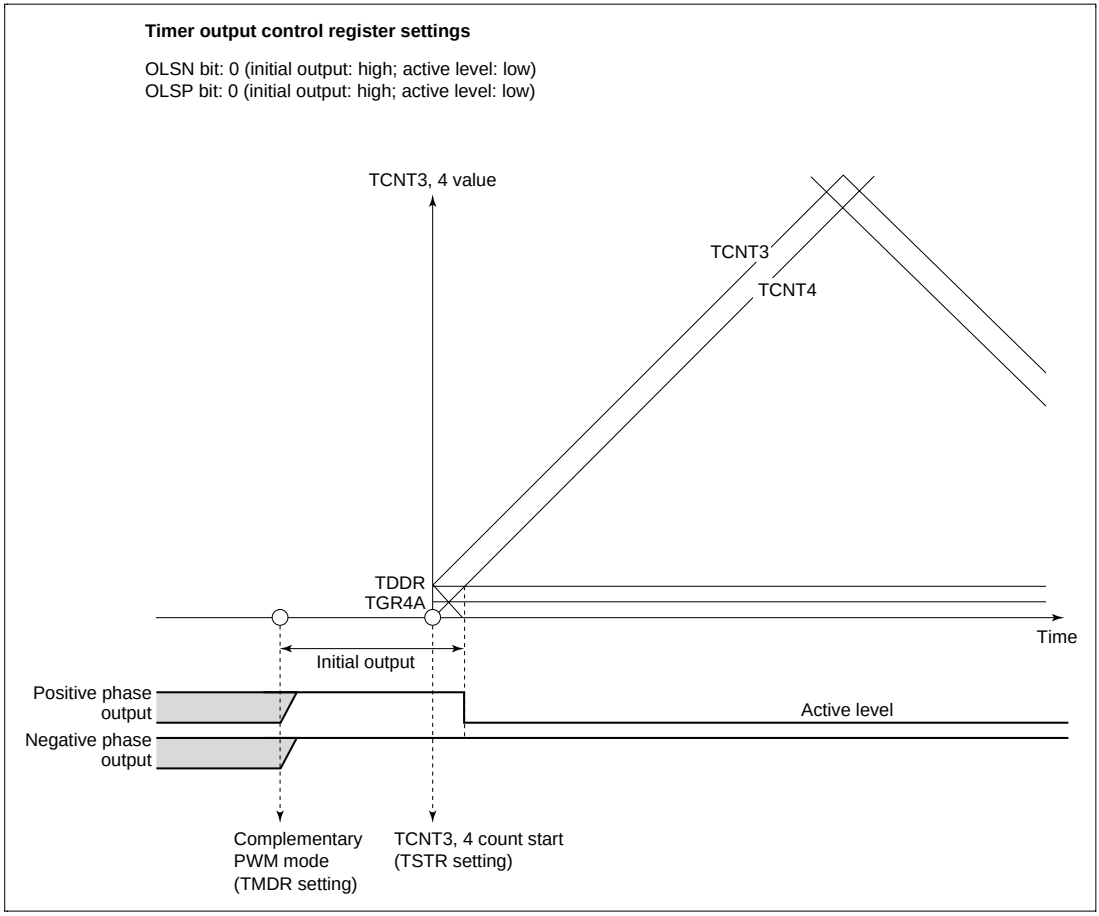


Figure 12.44 Example of Initial Output in Complementary PWM Mode (2)

- Complementary PWM mode PWM output generation method

  In complementary PWM mode, 3-phase output is performed of PWM waveforms with a non-overlap time between the positive and negative phases. This non-overlap time is called the dead time.

  A PWM waveform is generated by output of the output level selected in the timer output control register in the event of a compare-match between a counter and data register. While TCNTS is counting, data register and temporary register values are simultaneously compared to create consecutive PWM pulses from 0 to 100%. The relative timing of on and off compare-match occurrence may vary, but the compare-match that turns off each phase takes precedence to secure the dead time and ensure that the positive phase and negative phase on times do not overlap. Figures 12.45 to 12.47 show examples of waveform generation in complementary PWM mode.

  The positive phase/negative phase off timing is generated by a compare-match with the solid-line counter, and the on timing by a compare-match with the dotted-line counter operating with a delay of the dead time behind the solid-line counter. In the T1 period, compare-match **a** that turns off the negative phase has the highest priority, and compare-matches occurring prior to **a** are ignored. In the T2 period, compare-match **c** that turns off the positive phase has the highest priority, and compare-matches occurring prior to **c** are ignored.

  In normal cases, compare-matches occur in the order **a** → **b** → **c** → **d** (or **c** → **d** → **a'** → **b'**), as shown in figure 12.45.

  If compare-matches deviate from the **a** → **b** → **c** → **d** order, since the time for which the negative phase is off is less than twice the dead time, the figure shows the positive phase as not being turned on. If compare-matches deviate from the **c** → **d** → **a'** → **b'** order, since the time for which the positive phase is off is less than twice the dead time, the figure shows the negative phase as not being turned on.

  If compare-match **c** occurs first following compare-match **a**, as shown in figure 12.46, compare-match **b** is ignored, and the negative phase is turned off by compare-match **d**. This is because turning off of the positive phase has priority due to the occurrence of compare-match **c** (positive phase off timing) before compare-match **b** (positive phase on timing) (consequently, the waveform does not change since the positive phase goes from off to off).

  Similarly, in the example in figure 12.47, compare-match **a'** with the new data in the temporary register occurs before compare-match **c**, but other compare-matches occurring up to **c**, which turns of the positive phase, are ignored. As a result, the positive phase is not turned on.

  Thus, in complementary PWM mode, compare-matches at turn-off timings take precedence, and turn-on timing compare-matches that occur before a turn-off timing compare-match are ignored.

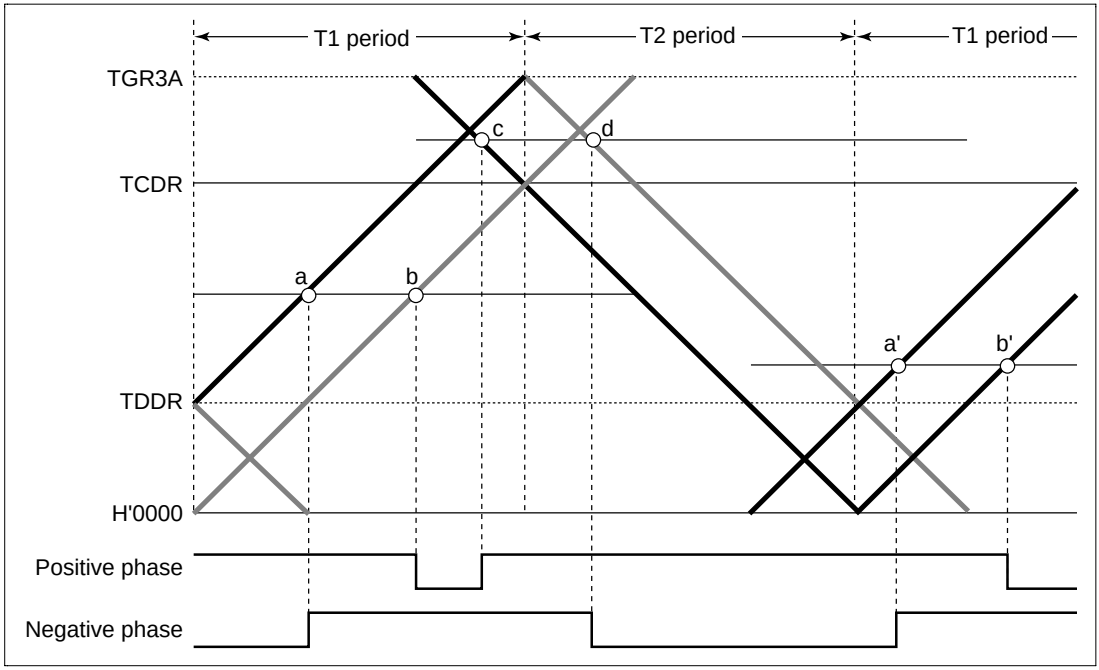


Figure 12.45 Example of Complementary PWM Mode Waveform Output (1)

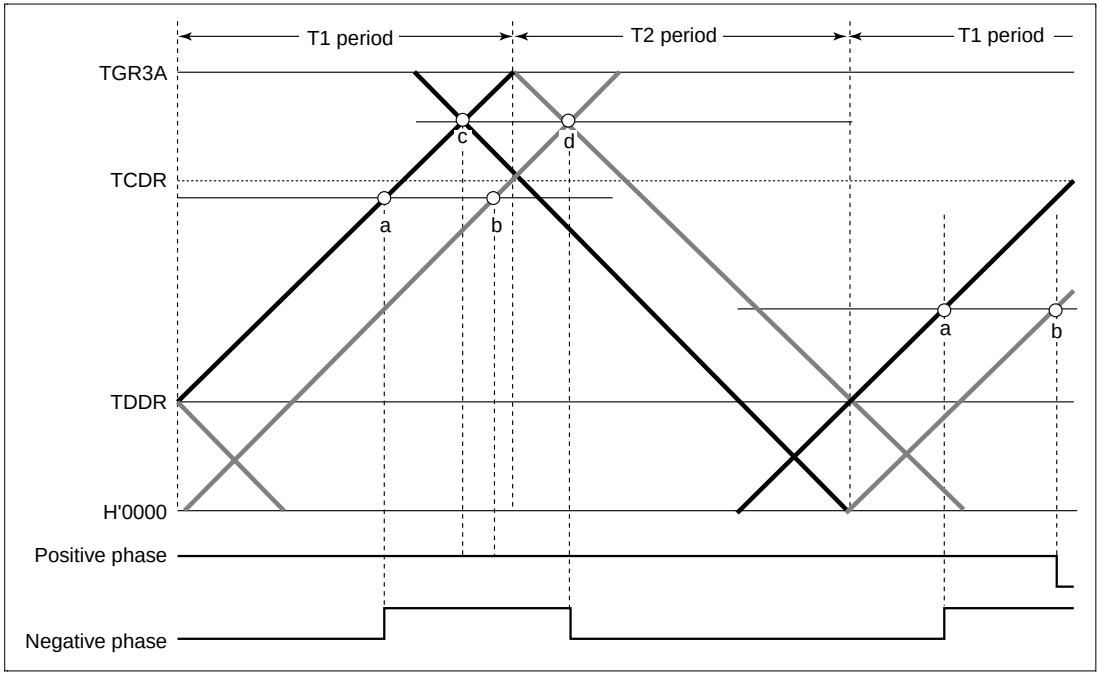


Figure 12.46 Example of Complementary PWM Mode Waveform Output (2)

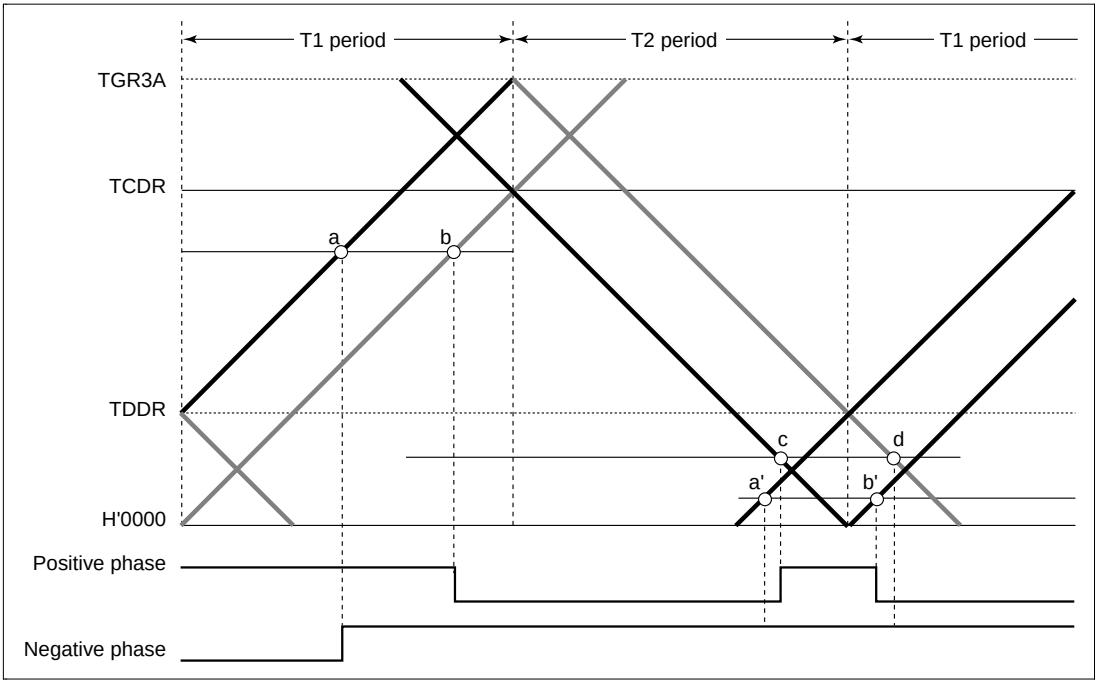


Figure 12.47 Example of Complementary PWM Mode Waveform Output (3)

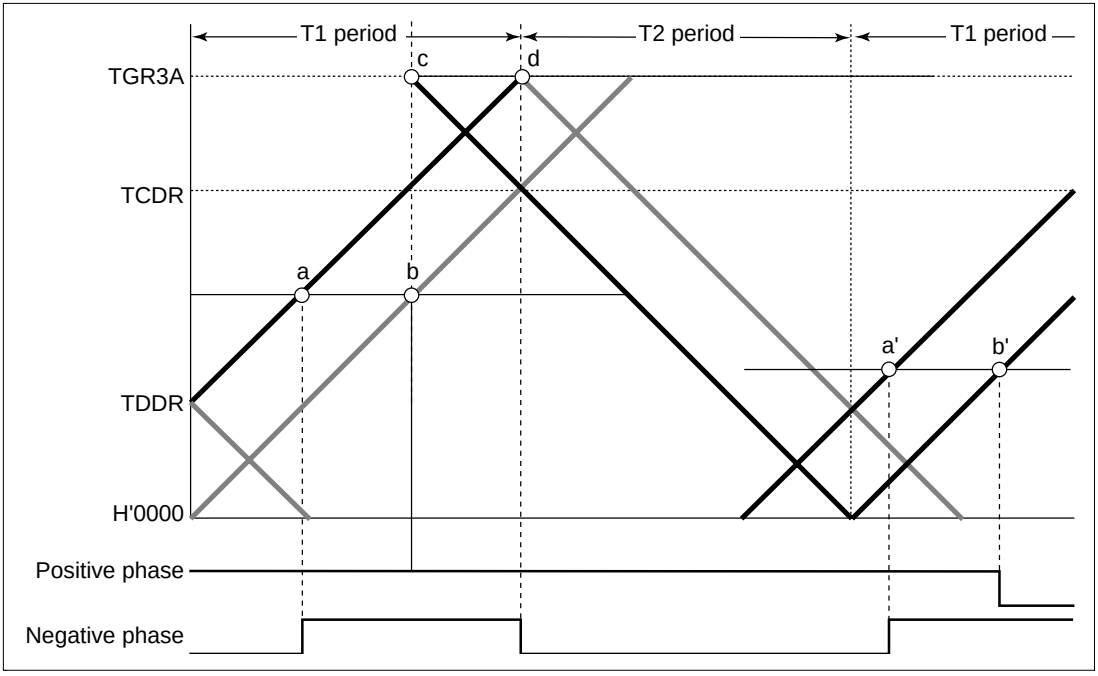


Figure 12.48 Example of Complementary PWM Mode 0% and 100% Waveform Output (1)

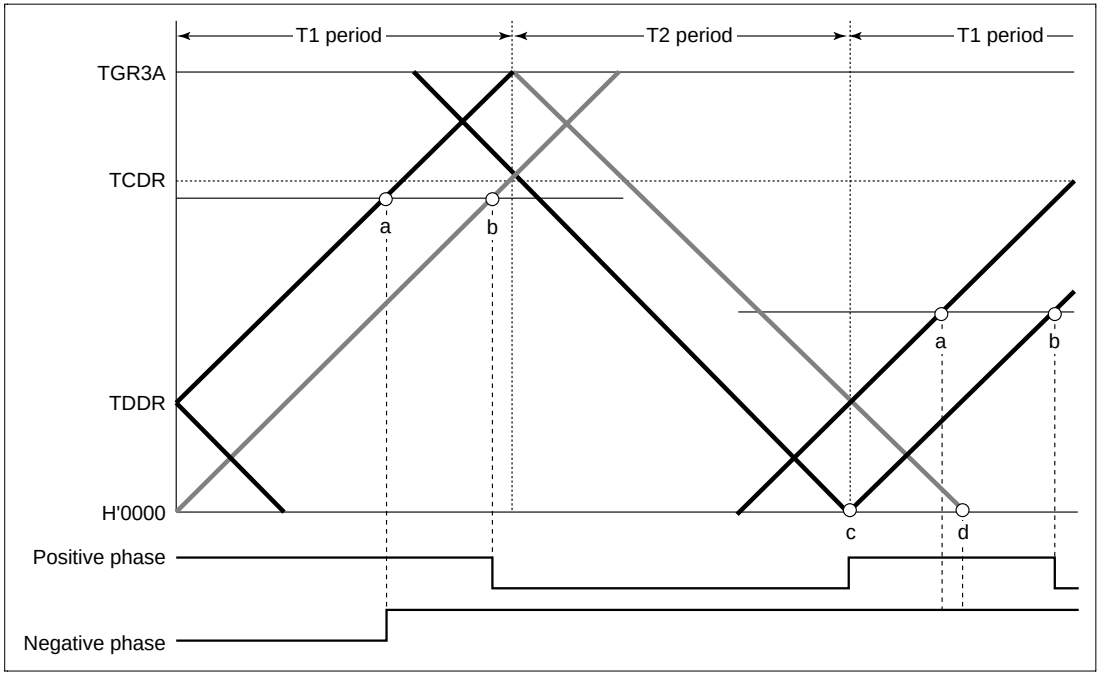


Figure 12.49 Example of Complementary PWM Mode 0% and 100% Waveform Output (2)

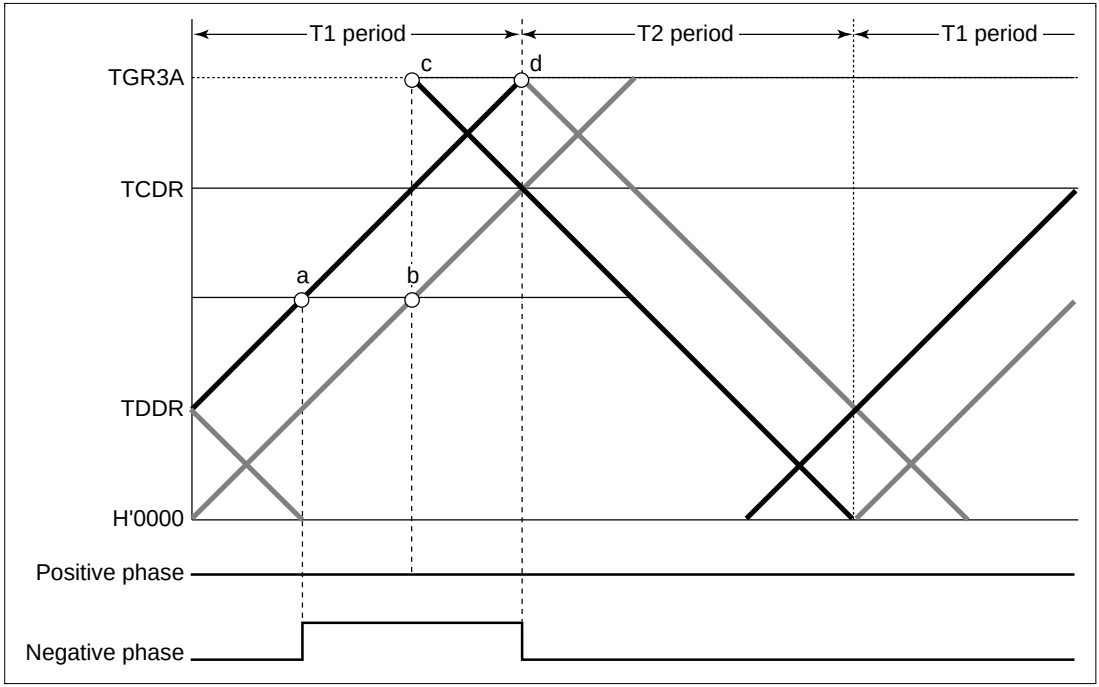


Figure 12.50 Example of Complementary PWM Mode 0% and 100% Waveform Output (3)

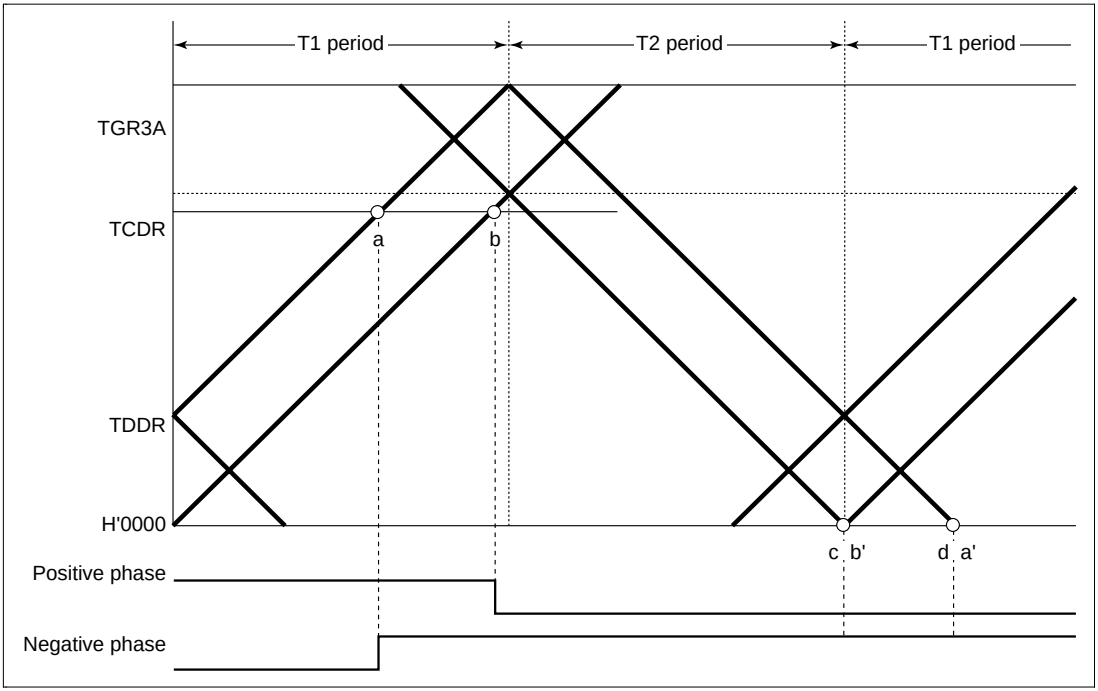


Figure 12.51 Example of Complementary PWM Mode 0% and 100% Waveform Output (4)

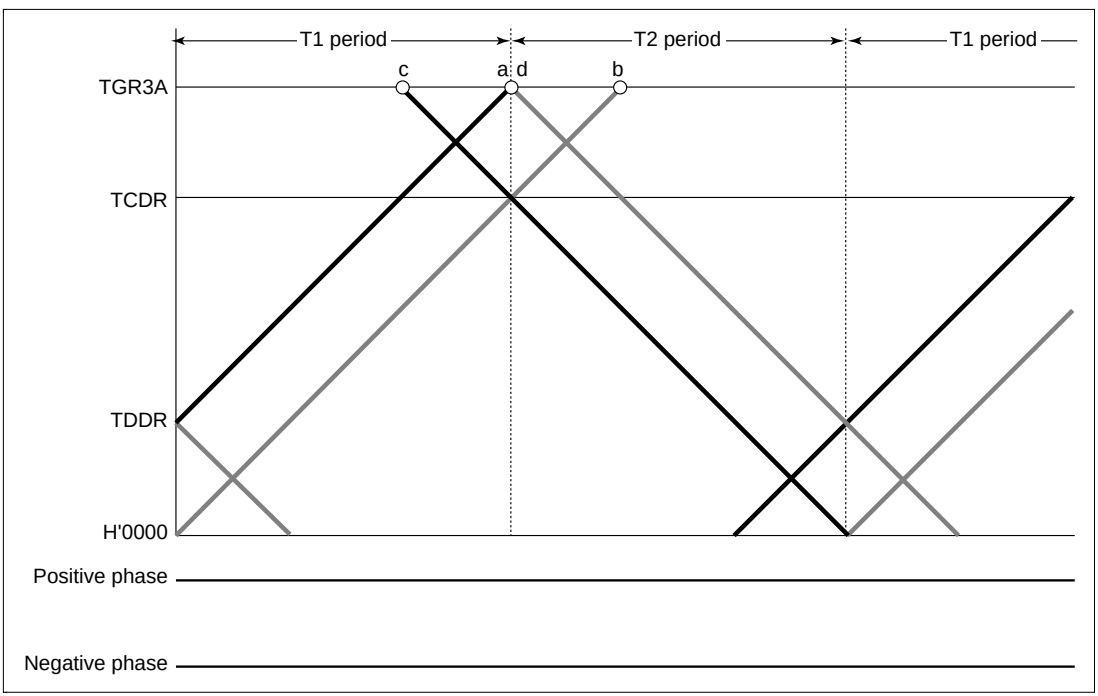


Figure 12.52 Example of Complementary PWM Mode 0% and 100% Waveform Output (5)

- Complementary PWM mode 0% and 100% duty output

  In complementary PWM mode, 0% and 100% duty cycles can be output as required. Figures 12.48 to 12.52 show output examples.

  100% duty output is performed when the data register value is set to H'0000. The waveform in this case has a positive phase with a 100% on-state. 0% duty output is performed when the data register value is set to the same value as TGR3A. The waveform in this case has a positive phase with a 100% off-state.

  On and off compare-matches occur simultaneously, but if a turn-on compare-match and turn-off compare-match for the same phase occur simultaneously, both compare-matches are ignored and the waveform does not change.

- Toggle output synchronized with PWM cycle

  In complementary PWM mode, toggle output can be performed in synchronization with the PWM carrier cycle by setting the PSYE bit to 1 in the timer output control register (TOCR). An example of a toggle output waveform is shown in figure 12.53.

  This output is toggled by a compare-match between TCNT3 and TGR3A and a compare-match between TCNT4 and H'0000.

  The output pin for this toggle output is the TIOC3A pin. The initial output is 1.

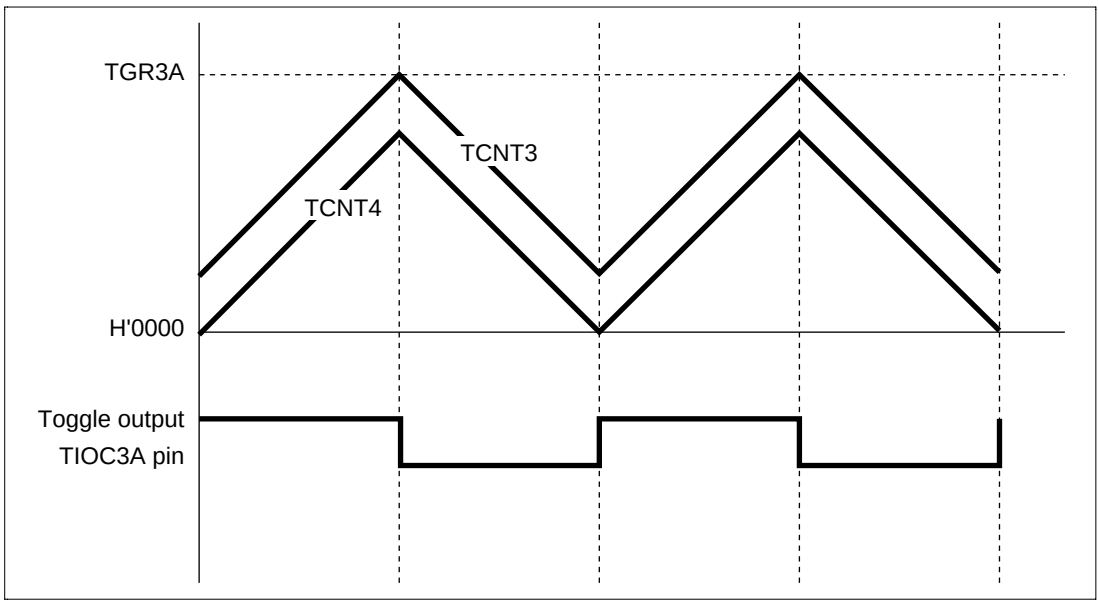


Figure 12.53 Example of Toggle Output Waveform Synchronized with PWM Output

- Counter clearing by another channel

  In complementary PWM mode, by setting a mode for synchronization with another channel by means of the timer synchro register (TSYR), and selecting synchronous clearing with bits CCLR2–CCLR0 in the timer control register (TCR), it is possible to have TCNT3, TCNT4, and TCNTS cleared by another channel.

  Figure 12.54 illustrates the operation.

  Use of this function enables counter clearing and restarting to be performed by means of an external signal.

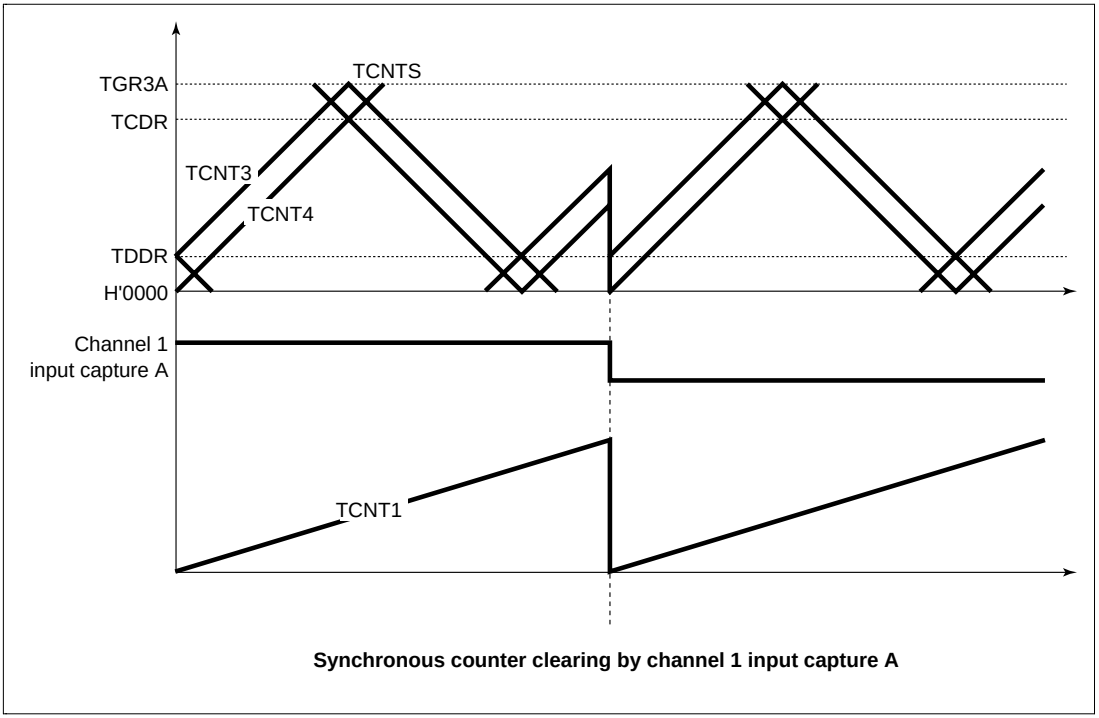


Figure 12.54 Counter Clearing Synchronized with Another Channel

- Example of AC synchronous motor (brushless DC motor) drive waveform output

  In complementary PWM mode, a brushless DC motor can easily be controlled using the timer gate control register (TGCR). Figures 12.55 to 12.58 show examples of brushless DC motor drive waveforms created using TGCR.

  When output phase switching for a 3-phase brushless DC motor is performed by means of external signals detected with a Hall element, etc., clear the FB bit in TGCR to 0. In this case, the external signals indicating the polarity position are input to channel 0 timer input pins TIOC0A, TIOC0B, and TIOC0C (set with PFC). When an edge is detected at pin TIOC0A, TIOC0B, or TIOC0C, the output on/off state is switched automatically.

  When the FB bit is 1, the output on/off state is switched when the UF, VF, or WF bit in TGCR is cleared to 0 or set to 1.

  The drive waveforms are output from the complementary PWM mode 6-phase output pins. With this 6-phase output, in the case of on output, it is possible to use complementary PWM mode output and perform chopping output by setting the N bit or P bit to 1. When the N bit or P bit is 0, level output is selected.

  The 6-phase output active level (on output level) can be set with the OLSN and OLSP bits in the timer output control register (TOCR) regardless of the setting of the N and P bits. When using this mode, set the 6-phase output waveform to High active (Low active is also permitted for A masks).

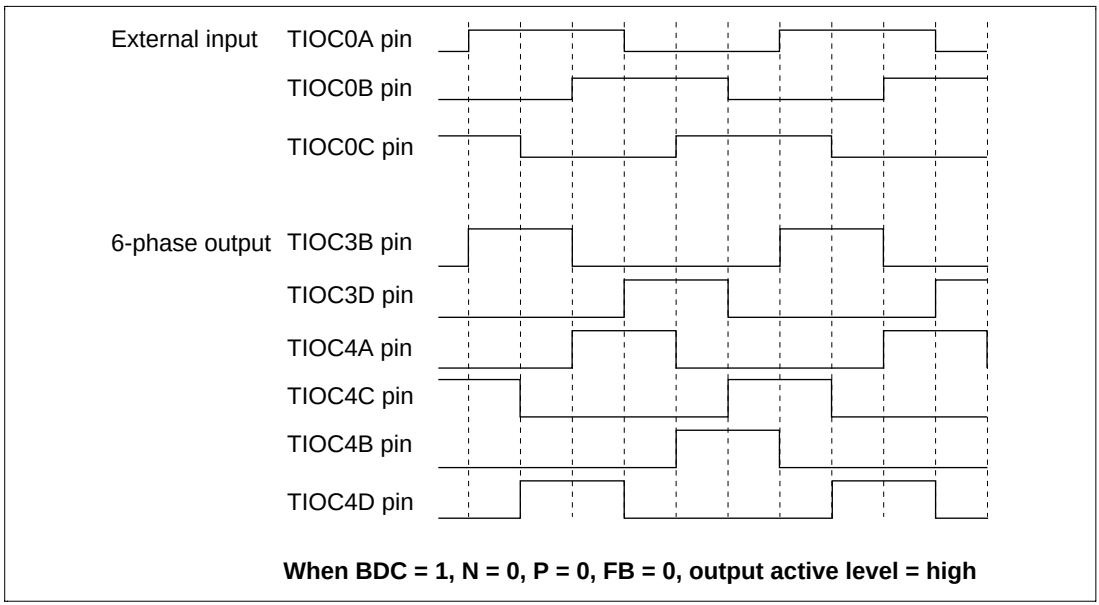


Figure 12.55 Example of Output Phase Switching by External Input (1)

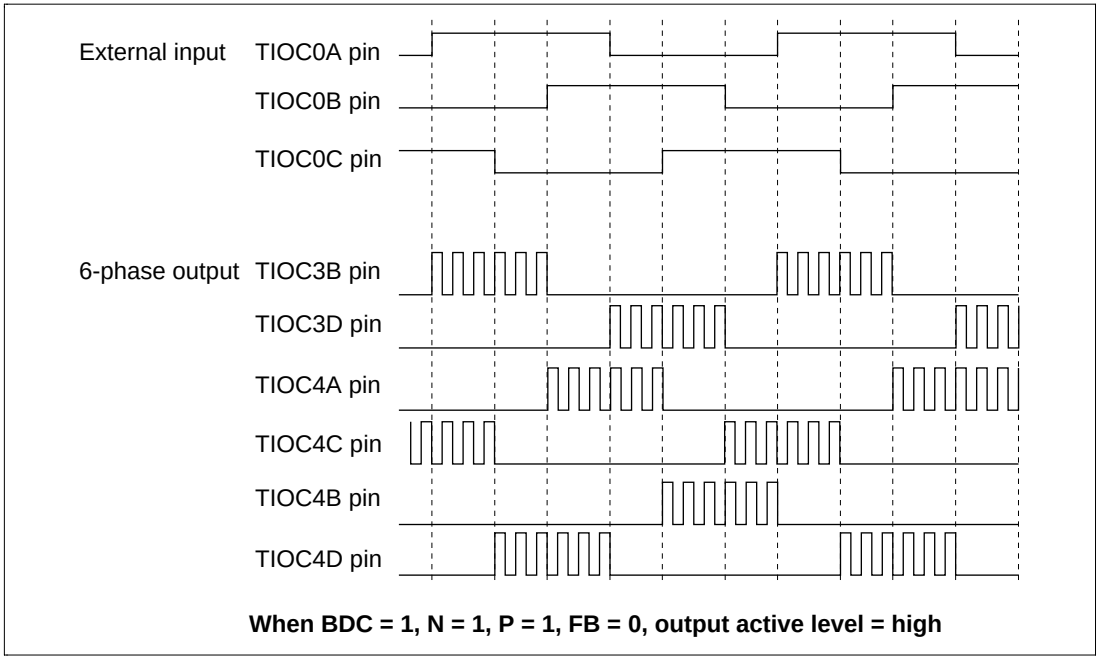


Figure 12.56 Example of Output Phase Switching by External Input (2)

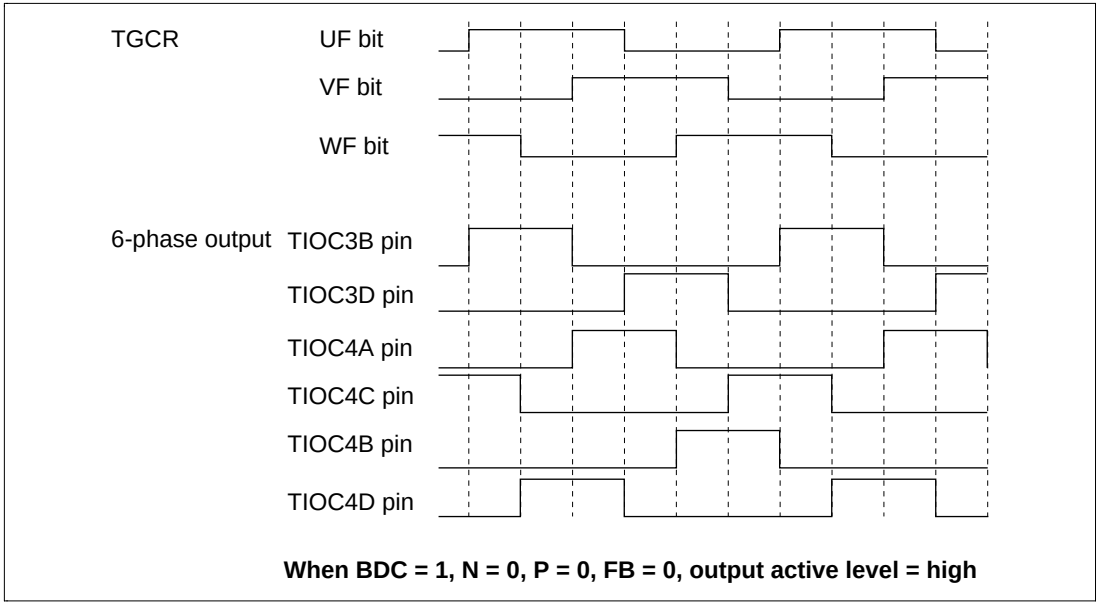


Figure 12.57 Example of Output Phase Switching by Means of UF, VF, WF Bit Settings (1)

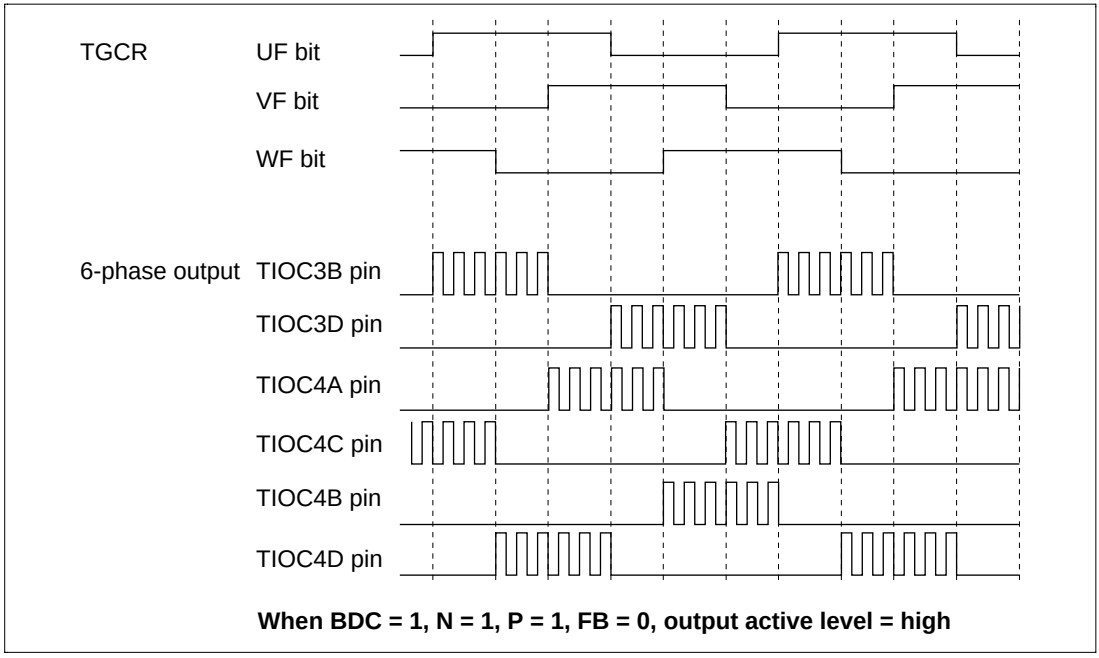


Figure 12.58 Example of Output Phase Switching by Means of UF, VF, WF Bit Settings (2)

- A/D conversion start request setting

  In complementary PWM mode, an A/D conversion start request can be issued using a TGR3A compare-match or a compare-match on a channel other than channels 3 and 4.

  When start requests using a TGR3A compare-match are set, A/D conversion can be started at the center of the PWM pulse.

  A/D conversion start requests can be set by setting the TTGE bit to 1 in the timer interrupt enable register (TIER).

Complementary PWM Mode Output Protection Function: Complementary PWM mode output has the following protection functions.

- Register and counter miswrite prevention function

  With the exception of the buffer registers, which can be rewritten at any time, access by the CPU can be enabled or disabled for the mode registers, control registers, compare registers, and counters used in complementary PWM mode by means of bit 13 in the bus controller's bus control register 1 (BCR1). The registers and counters concerned are listed in table 12.3.

  This function enables miswriting due to CPU runaway to be prevented by disabling CPU access to the mode registers, control registers, and counters.

- Halting of PWM output by external signal

  The 6-phase PWM output pins can be set automatically to the high-impedance state by inputting specified external signals. There are four external signal input pins.

  See section 12.9, Port Output Enable (POE), for details.

- Halting of PWM output when oscillator is stopped

  If it is detected that the clock input to the SH7040 chip has stopped, the 6-phase PWM output pins automatically go to the high-impedance state. The pin states are not guaranteed when the clock is restarted.

  See section 4.4, Oscillator Halt Function, for details.

12.5 Interrupts

12.5.1 Interrupt Sources and Priority Ranking

The MTU has three interrupt sources: TGR register compare-match/input captures, TCNT counter overflows and TCNT counter underflows. Because each of these three types of interrupts are allocated its own dedicated status flag and enable/disable bit, the issuing of interrupt request signals to the interrupt controller can be independently enabled or disabled.

When an interrupt source is generated, the corresponding status flag in the timer status register (TSR) is set to 1. If the corresponding enable/disable bit in the timer input enable register (TIER) is set to 1 at this time, the MTU makes an interrupt request of the interrupt controller. The interrupt request is canceled by clearing the status flag to 0.

The channel priority order can be changed with the interrupt controller. The priority ranking within a channel is fixed. For more information, see section 6, Interrupt Controller (INTC).

Table 12.17 lists the MTU interrupt sources.

Input Capture/Compare Match Interrupts: If the TGIE bit of the timer input enable register (TIER) is already set to 1 when the TGF flag in the timer status register (TSR) is set to 1 by a TGR register input capture/compare-match of any channel, an interrupt request is sent to the interrupt controller. The interrupt request is canceled by clearing the TGF flag to 0. The MTU has 16 input capture/compare-match interrupts; four each for channels 0, 3, and 4, and two each for channels 1 and 2.

Overflow Interrupts: If the TCIEV bit of the TIER is already set to 1 when the TCFV flag in the TSR is set to 1 by a TCNT counter overflow of any channel, an interrupt request is sent to the interrupt controller. The interrupt request is canceled by clearing the TCFV flag to 0. The MTU has five overflow interrupts, one for each channel.

Underflow Interrupts: If the TCIEU bit of the TIER is already set to 1 when the TCFU flag in the TSR is set to 1 by a TCNT counter underflow of any channel, an interrupt request is sent to the interrupt controller. The interrupt request is canceled by clearing the TCFU flag to 0. The MTU has two underflow interrupts, one each for channels 1 and 2.

Table 12.17 MTU Interrupt Sources

Channel	Interrupt Source	Description	DMAC Activation	DTC Activation	Priority*
0	TGI0A	TGR0A input capture/compare-match	Yes	Yes	High
	TGI0B	TGR0B input capture/compare-match	No	Yes	
	TGI0C	TGR0C input capture/compare-match	No	Yes	
	TGI0D	TGR0D input capture/compare-match	No	Yes	
	TCI0V	TCNT0 overflow	No	No	
1	TGI1A	TGR1A input capture/compare-match	Yes	Yes	
	TGI1B	TGR1B input capture/compare-match	No	Yes	
	TCI1V	TCNT1 overflow	No	No	
	TCI1U	TCNT1 underflow	No	No	
2	TGI2A	TGR2A input capture/compare-match	Yes	Yes	
	TGI2B	TGR2B input capture/compare-match	No	Yes	
	TCI2V	TCNT2 overflow	No	No	
	TCI2U	TCNT2 underflow	No	No	
3	TGI3A	TGR3A input capture/compare-match	Yes	Yes	
	TGI3B	TGR3B input capture/compare-match	No	Yes	
	TGI3C	TGR3C input capture/compare-match	No	Yes	
	TGI3D	TGR3D input capture/compare-match	No	Yes	
	TCI3V	TCNT3 overflow	No	No	
4	TGI4A	TGR4A input capture/compare-match	Yes	Yes	
	TGI4B	TGR4B input capture/compare-match	No	Yes	
	TGI4C	TGR4C input capture/compare-match	No	Yes	
	TGI4D	TGR4D input capture/compare-match	No	Yes	
	TCI4V	TCNT overflow/underflow	No	Yes	Low

Note: * Indicates the initial status following reset. The ranking of channels can be altered using the interrupt controller.

12.5.2 DTC/DMAC Activation

DTC Activation: The TGR register input capture/compare-match interrupt of any channel can be used as a source to activate the on-chip data transfer controller (DTC). For details, refer to section 8, Data Transfer Controller (DTC).

The MTU has 17 input capture/compare-match interrupts that can be used as DTC activation sources, four each for channels 0 and 3, two each for channels 1 and 2, and five for channel 4.

DMAC Activation: The TGRA register input capture/compare-match interrupt of any channel can be used as a source to activate the on-chip DMAC. For details, refer to section 11, Direct Memory Access Controller (DMAC).

The MTU has 5 TGRA register input capture/compare-match interrupts, one for any channel, that can be used as DMAC activation sources.

12.5.3 A/D Converter Activation

The TGRA register input capture/compare-match of any channel can be used to activate the on-chip A/D converter.

If the TTGE bit of the TIER is already set to 1 when the TGFA flag in the TSR is set to 1 by a TGRA register input capture/compare-match of any of the channels, an A/D conversion start request is sent to the A/D converter. If the MTU conversion start trigger is selected at such a time on the A/D converter side when this happens, the A/D conversion starts.

The MTU has 5 TGRA register input capture/compare-match interrupts, one for each channel, that can be used as A/D converter activation sources.

12.6 Operation Timing

12.6.1 Input/Output Timing

TCNT Count Timing: Count timing for the TCNT counter with internal clock operation is shown in figure 12.59. Count timing with external clock operation (normal mode) is shown in figure 12.60, and figure 12.61 shows count timing with external clock operation (phase counting mode).

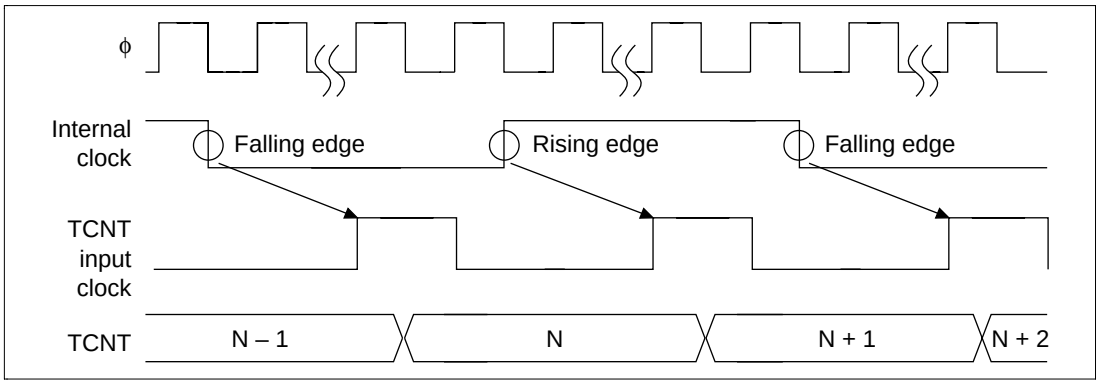


Figure 12.59 TCNT Count Timing during Internal Clock Operation

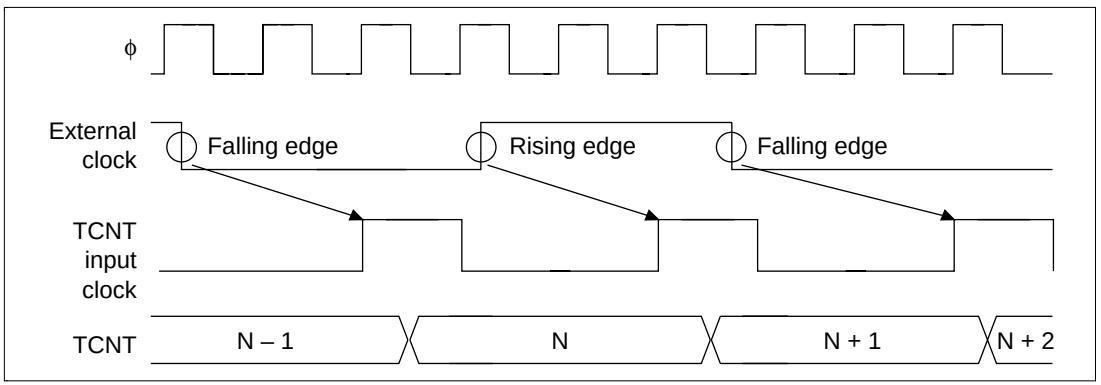


Figure 12.60 TCNT Count Timing during External Clock Operation (Normal Mode)

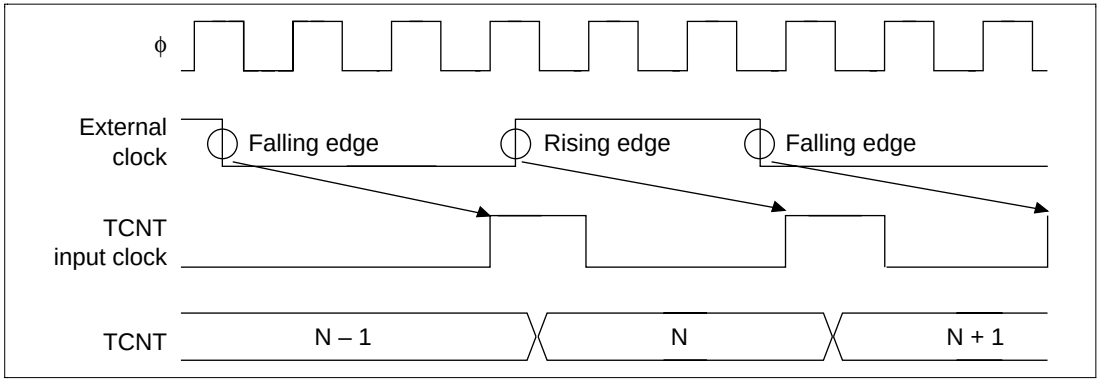


Figure 12.61 TCNT Count Timing during External Clock Operation (Phase Counting Mode)

Output Compare Output Timing: The compare-match signal is generated at the final state of TCNT and TGR matching. When a compare-match signal is issued, the output value set in TIOR or TOCR is output to the output compare output pin (TIOC pin). After TCNT and TGR matching, a compare-match signal is not issued until immediately before the TCNT input clock.

Output compare output timing (normal mode and PWM mode) is shown in figure 12.62. See figure 12.63 for output compare output timing in complementary PWM mode and reset sync PWM mode.

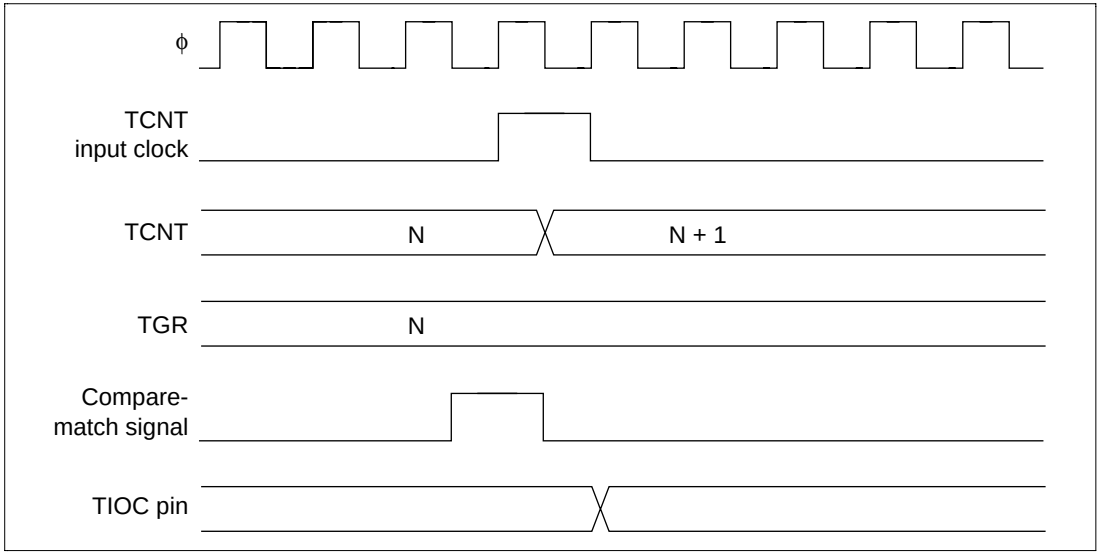


Figure 12.62 Output Compare Output Timing (Normal Mode/PWM Mode)

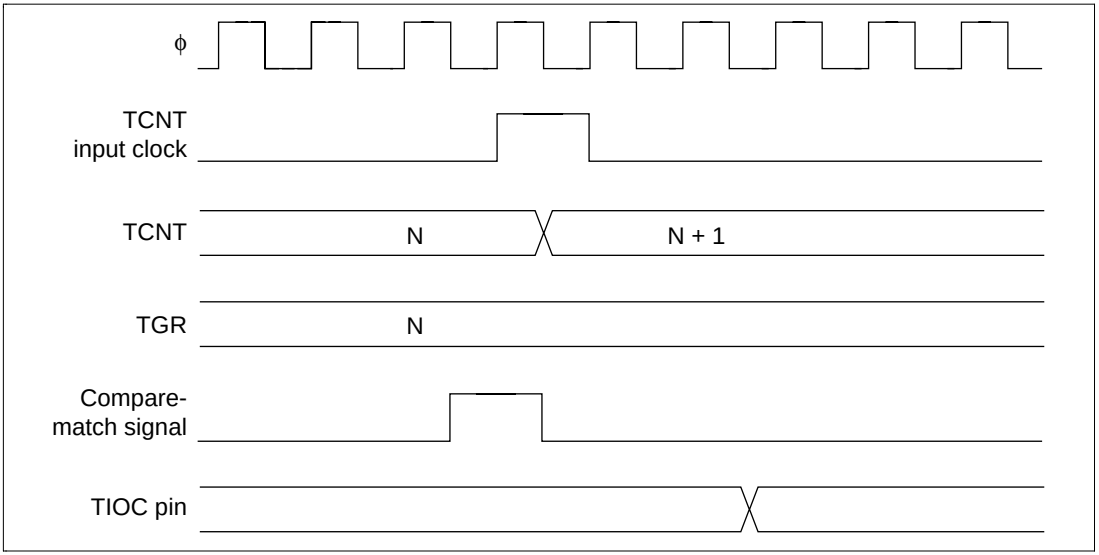


Figure 12.63 Output Compare Output Timing (Complementary PWM Mode/Reset Sync PWM Mode)

Input Capture Signal Timing: Figure 12.64 illustrates input capture timing.

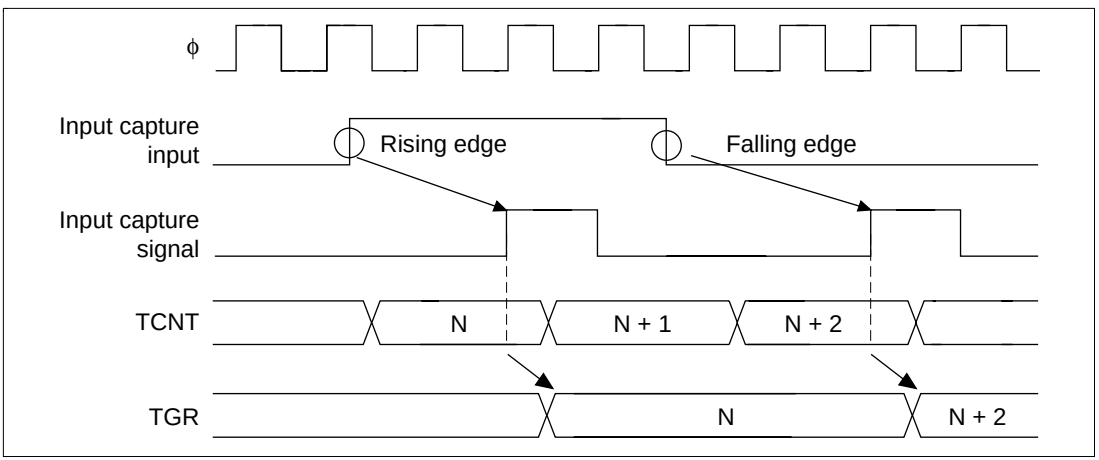


Figure 12.64 Input Capture Input Signal Timing

Counter Clearing Timing Due to Compare-Match/Input Capture: Timing for counter clearing due to compare-match is shown in figure 12.65. Figure 12.66 shows the timing for counter clearing due to input capture.

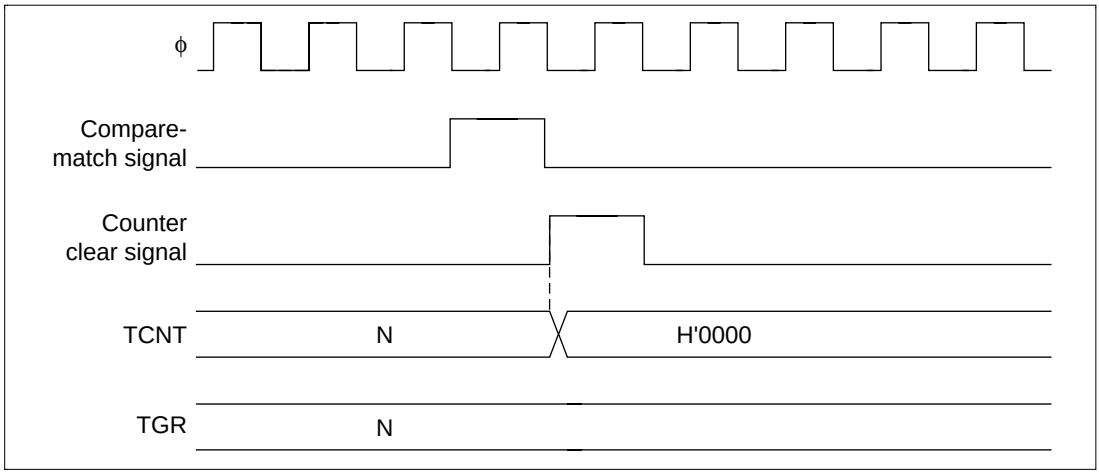


Figure 12.65 Counter Clearing Timing (Compare-Match)

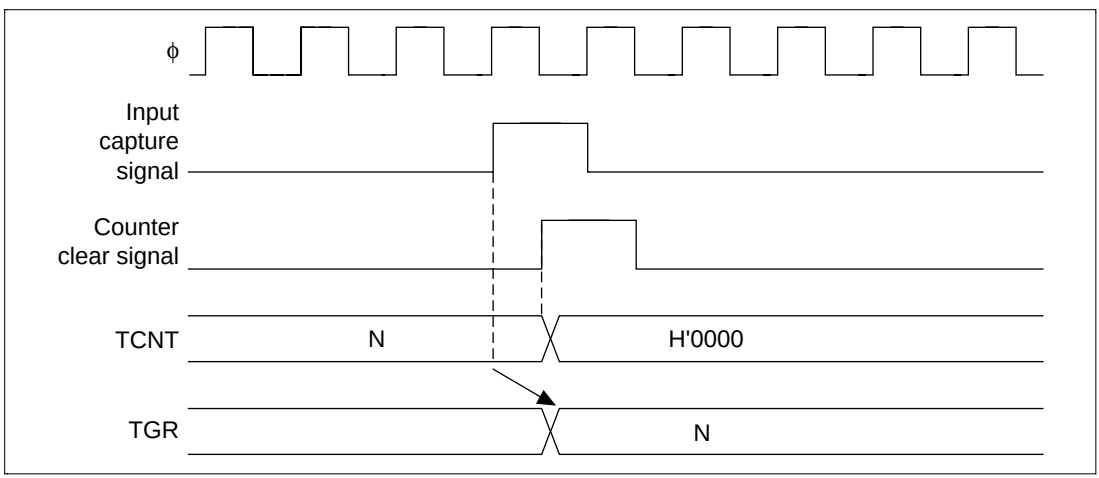


Figure 12.66 Counter Clearing Timing (Input Capture)

Buffer Operation Timing: Compare-match buffer operation timing is shown in figure 12.67. Figure 12.68 shows input capture buffer operation timing.

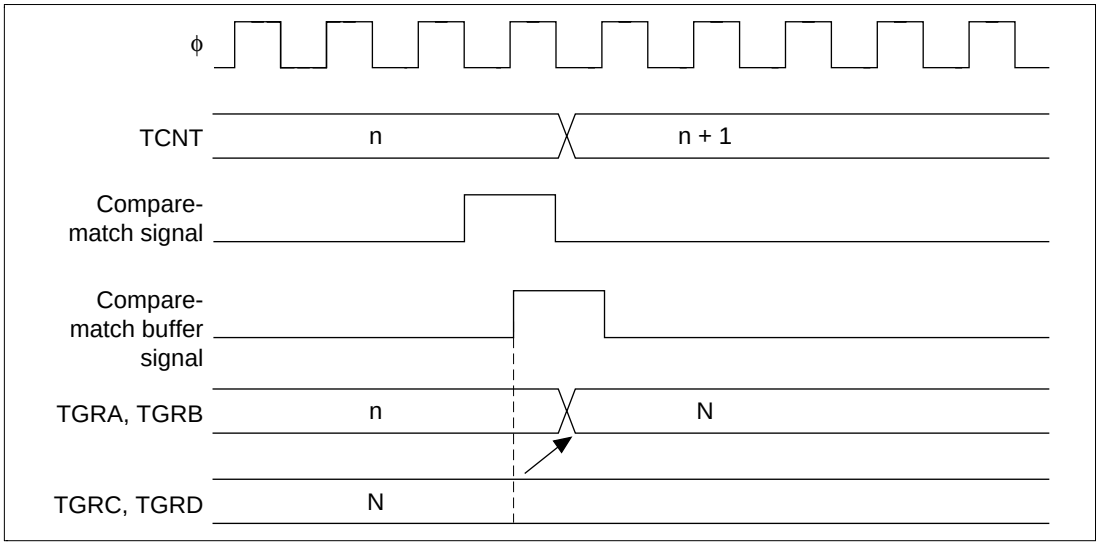


Figure 12.67 Buffer Operation Timing (Compare-Match)

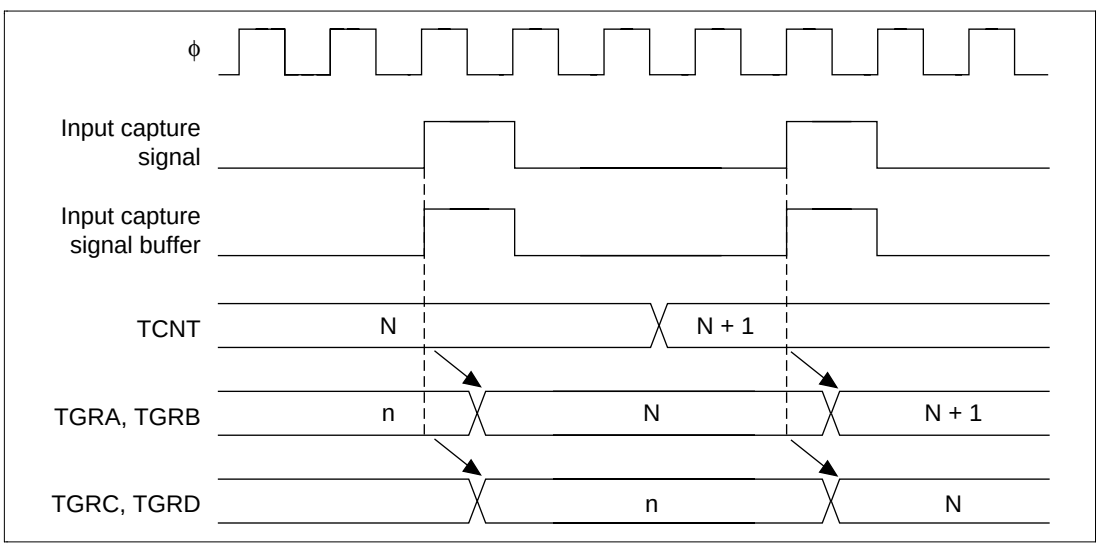


Figure 12.68 Buffer Operation Timing (Input Capture)

12.6.2 Interrupt Signal Timing

Setting TGF Flag Timing during Compare-Match: Figure 12.69 shows timing for the TGF flag of the timer status register (TSR) due to compare-match, as well as TGI interrupt request signal timing.

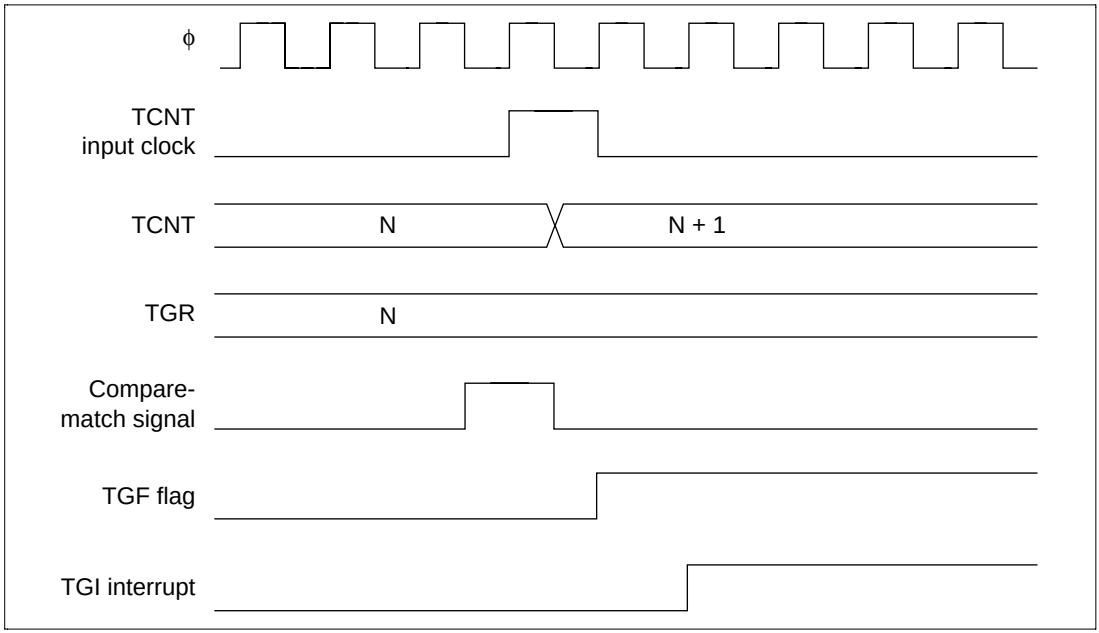


Figure 12.69 TGI Interrupt Timing (Compare Match)

Setting TGF Flag Timing during Input Capture: Figure 12.70 shows timing for the TGF flag of the timer status register (TSR) due to input capture, as well as TGI interrupt request signal timing.

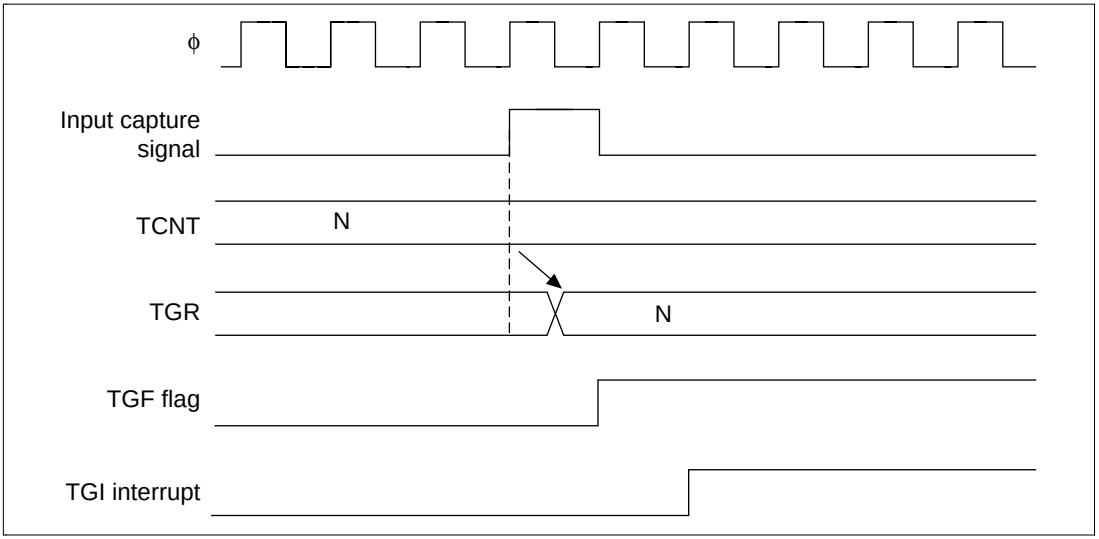


Figure 12.70 TGI Interrupt Timing (Input Capture)

Setting Timing for Overflow Flag (TCFV)/Underflow Flag (TCFU): Figure 12.71 shows timing for the TCFV flag of the timer status register (TSR) due to overflow, as well as TCIV interrupt request signal timing. Figure 12.72 shows timing for the TCFU flag of the timer status register (TSR) due to underflow, as well as TCIU interrupt request signal timing. Figure 12.73 shows timing for the TCFV flag of TSR4 due to underflow in complementary PWM mode, as well as TCIV interrupt request signal timing.

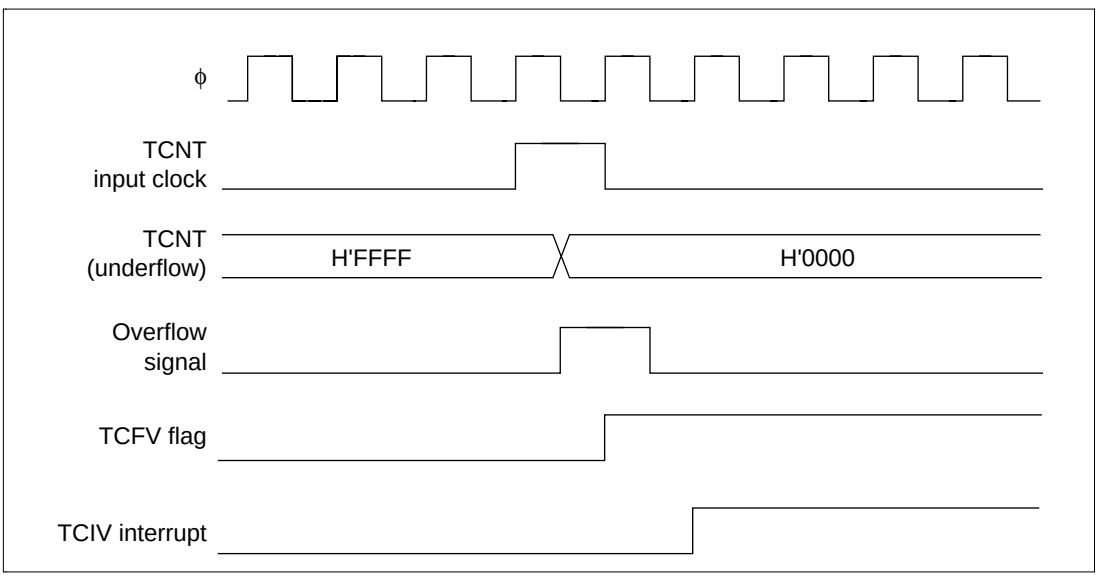


Figure 12.71 TCIV Interrupt Setting Timing

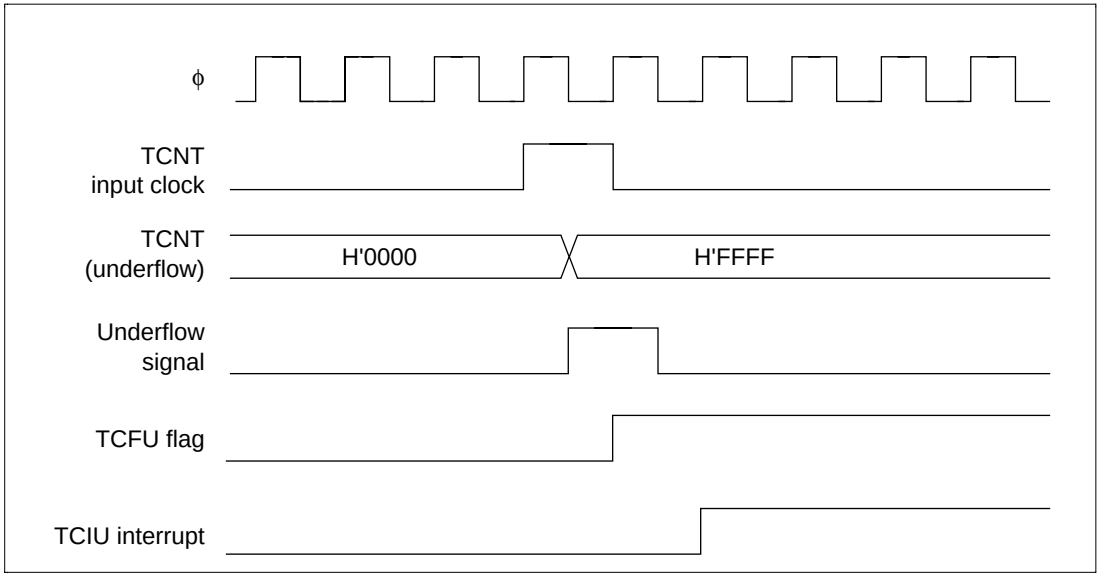


Figure 12.72 TCIU Interrupt Setting Timing

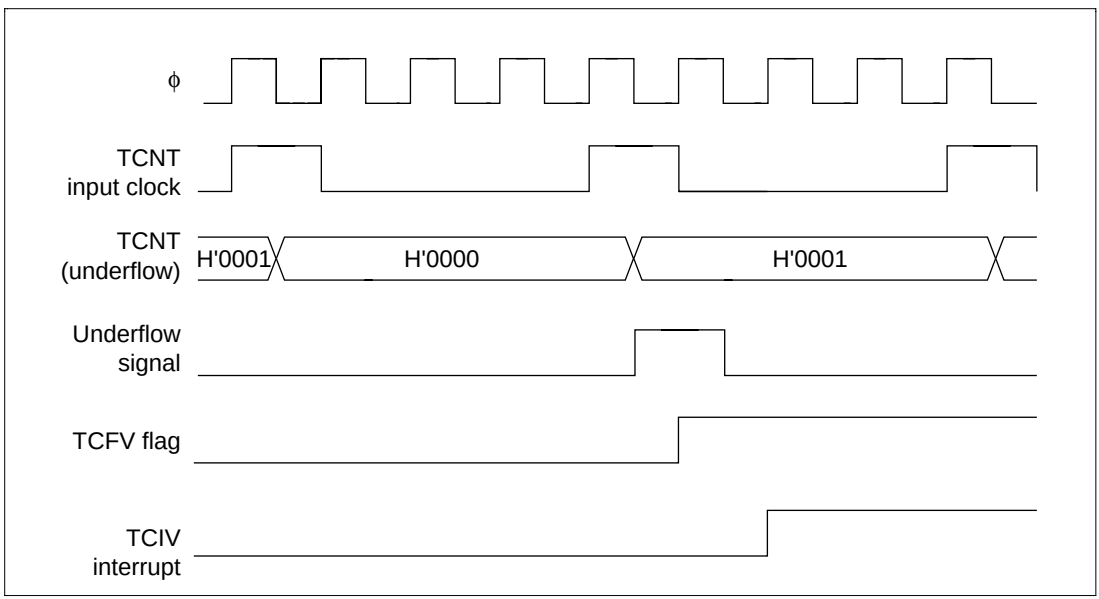


Figure 12.73 TCIV Interrupt Setting Timing (TSR4, Complementary PWM Mode)

Status Flag Clearing Timing: The status flag is cleared when the CPU reads a 1 status followed by a 0 write. For DTC/DMA controller activation, clearing can also be done automatically. Figure 12.74 shows the timing for status flag clearing by the CPU. Figure 12.75 shows timing for clearing due to the DTC/DMA controller.

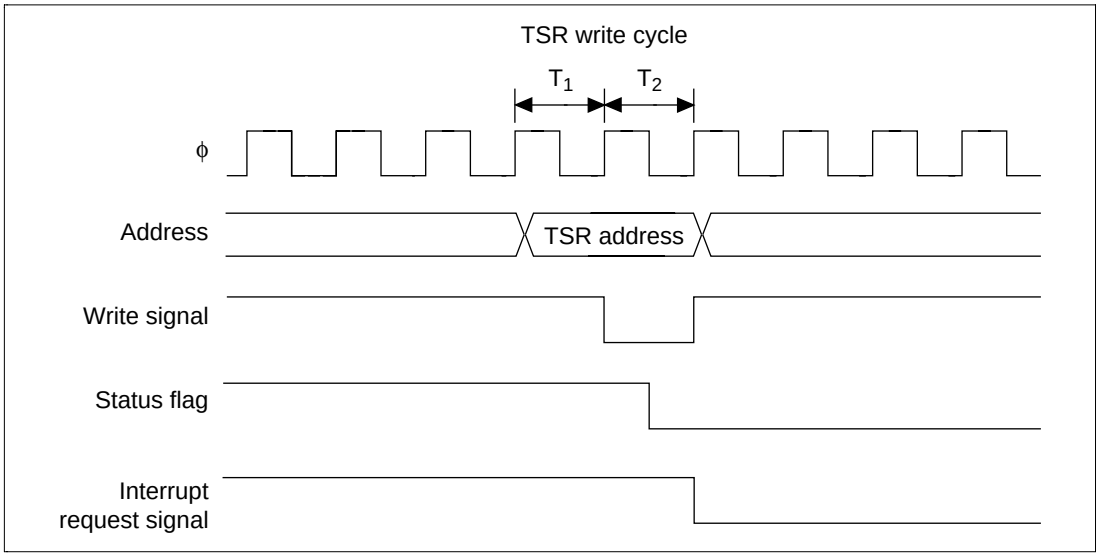


Figure 12.74 Timing of Status Flag Clearing by the CPU

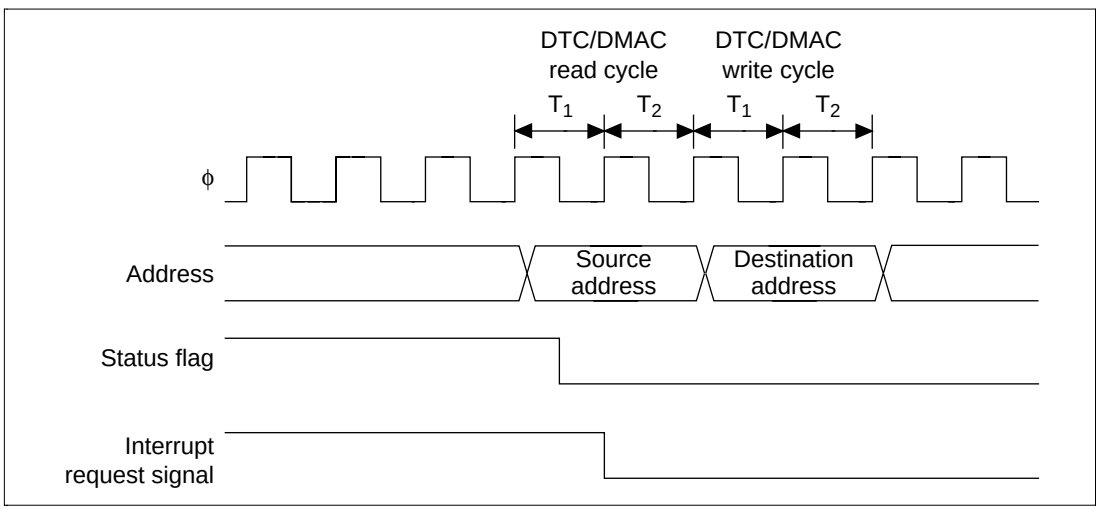


Figure 12.75 Timing of Status Flag Clearing by DTC/DMAC Activation

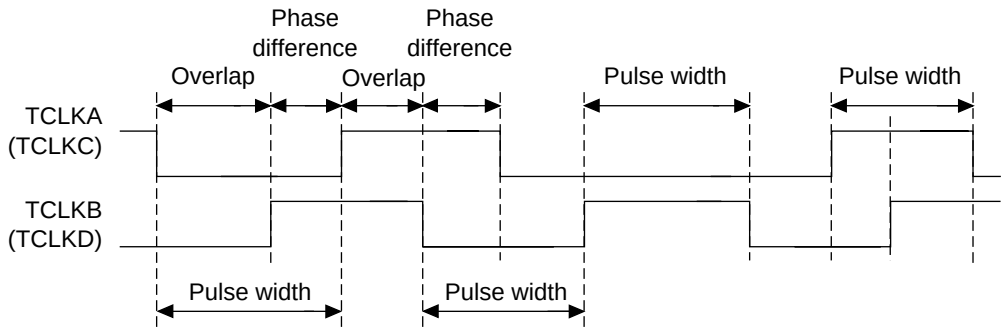
12.7 Notes and Precautions

This section describes contention and other matters requiring special attention during MTU operations.

12.7.1 Input Clock Limitations

The input clock pulse width, in the case of single edge, must be 1.5 states or greater, and 2.5 states or greater for both edges. Normal operation cannot be guaranteed with lesser pulse widths.

In phase counting mode, the phase difference between the two input clocks and the overlap must be 1.5 states or greater for each, and the pulse width must be 2.5 states or greater. Input clock conditions for phase counting mode are shown in figure 12.76.



Note: Phase difference and overlap: 1.5 states or greater
Pulse width: 2.5 states or greater

Figure 12.76 Phase Difference, Overlap, and Pulse Width in Phase Count Mode

12.7.2 Note on Cycle Setting

When setting a counter clearing by compare-match, clearing is done in the final state when TCNT matches the TGR value (update timing for count value on TCNT match). The actual number of states set in the counter is given by the following equation:

$$f = \frac{\phi}{(N+1)}$$

(f: counter frequency, ϕ: operating frequency, N: value set in the TGR)

12.7.3 Contention between TCNT Write and Clear

If a counter clear signal is issued in the T_2 state during the TCNT write cycle, TCNT clearing has priority, and TCNT write is not conducted (figure 12.77).

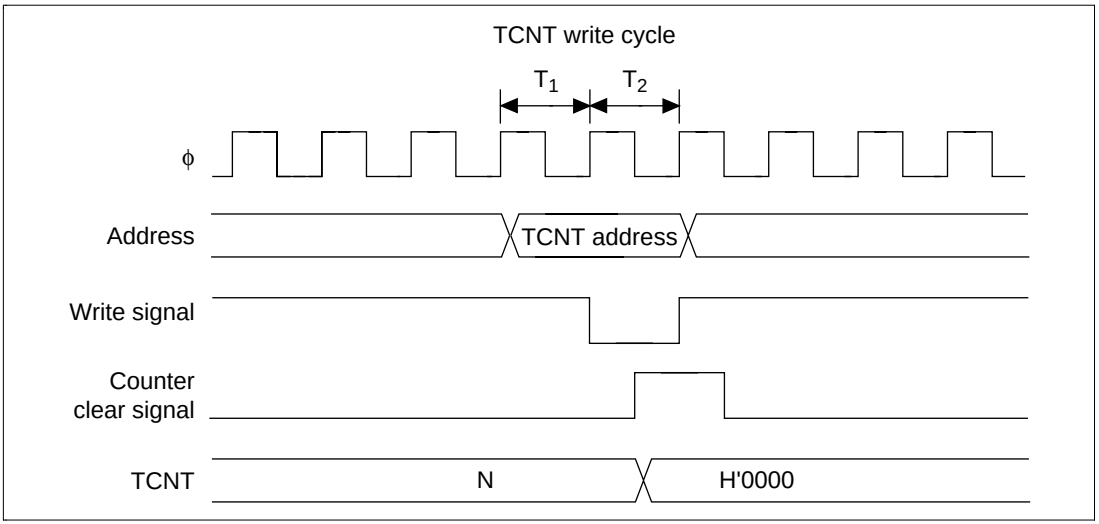


Figure 12.77 TCNT Write and Clear Contention

12.7.4 Contention between TCNT Write and Increment

If a count-up signal is issued in the T_2 state during the TCNT write cycle, TCNT write has priority, and the counter is not incremented (figure 12.78).

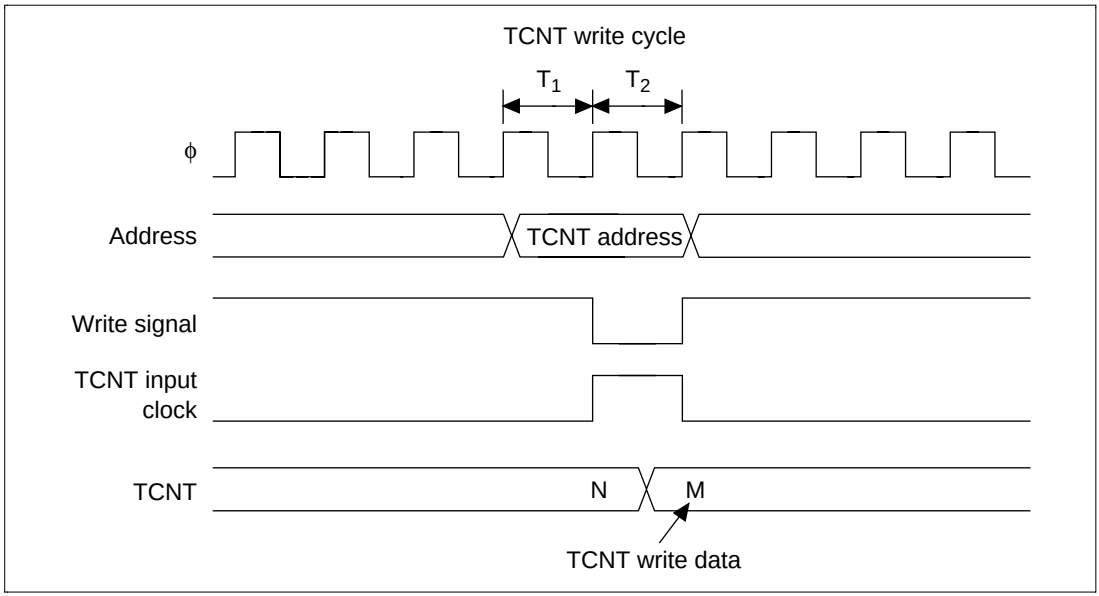


Figure 12.78 TCNT Write and Increment Contention

12.7.5 Contention between Buffer Register Write and Compare Match

If a compare-match occurs in the T_2 state of the TGR write cycle, data is transferred by the buffer operation from the buffer register to the TGR. Data to be transferred differs depending on channels 0 and 3 and 4: data on channel 0 is that after write, and on channels 3 and 4, before write (figures 12.79 and 12.80).

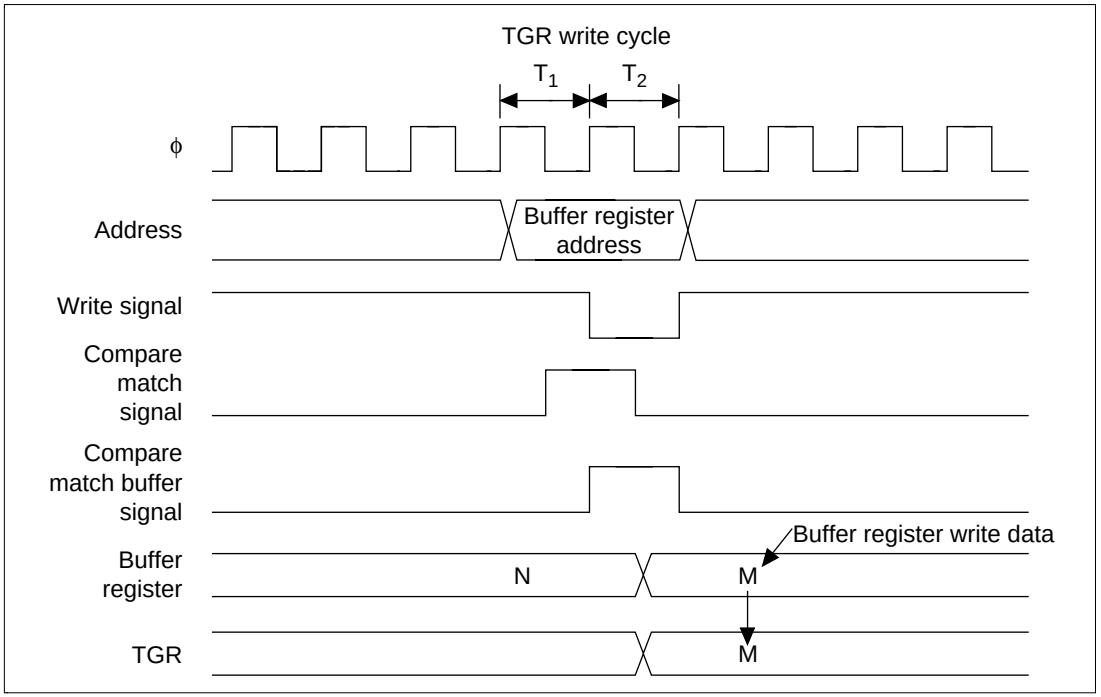


Figure 12.79 TGR Write and Compare-Match Contention (Channel 0)

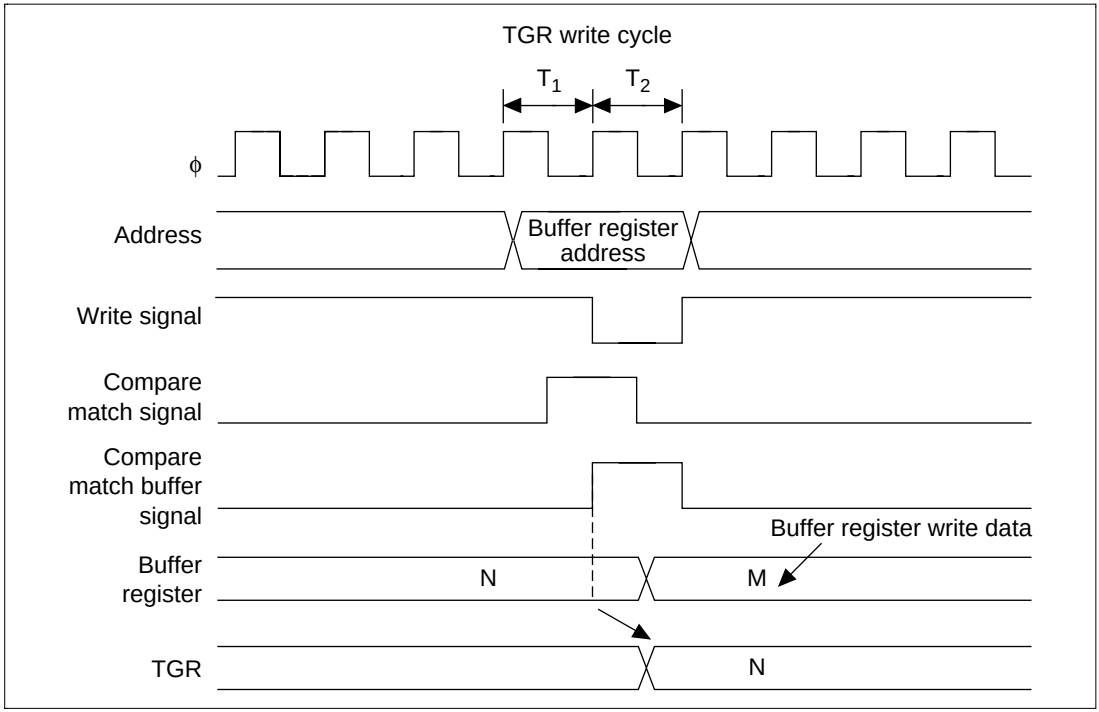


Figure 12.80 TGR Write and Compare-Match Contention (Channels 3 and 4)

12.7.6 Contention between TGR Read and Input Capture

If an input capture signal is issued in the T_1 state of the TGR read cycle, the read data is that after input capture transfer (figure 12.81).

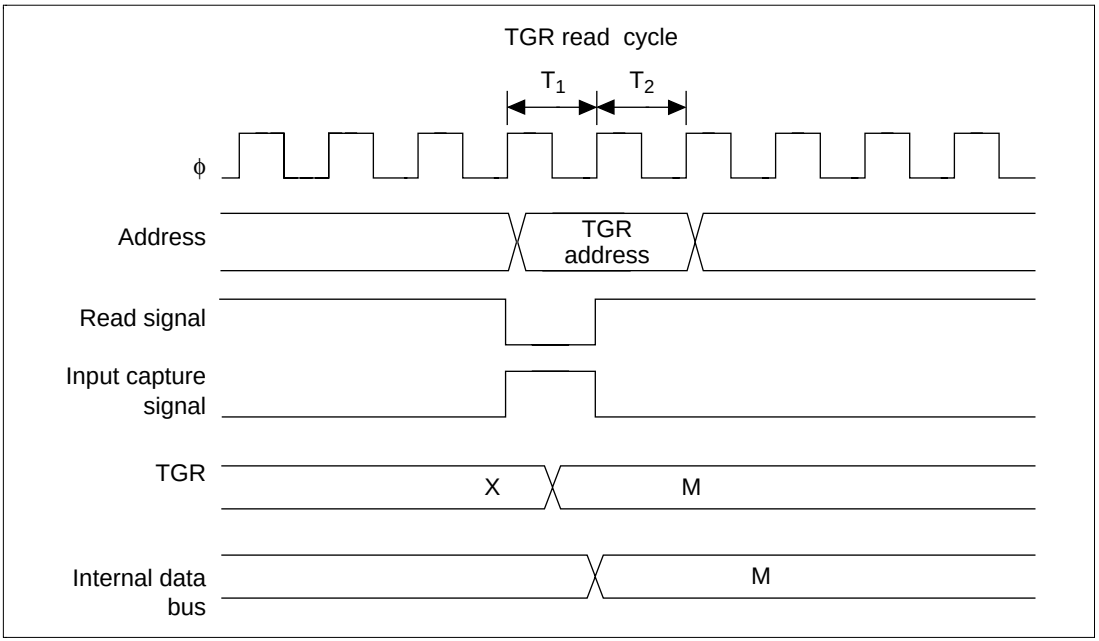


Figure 12.81 TGR Read and Input Capture Contention

12.7.7 Contention between TGR Write and Input Capture

If an input capture signal is issued in the T_2 state of the TGR read cycle, input capture has priority, and TGR write does not occur (figure 12.82).

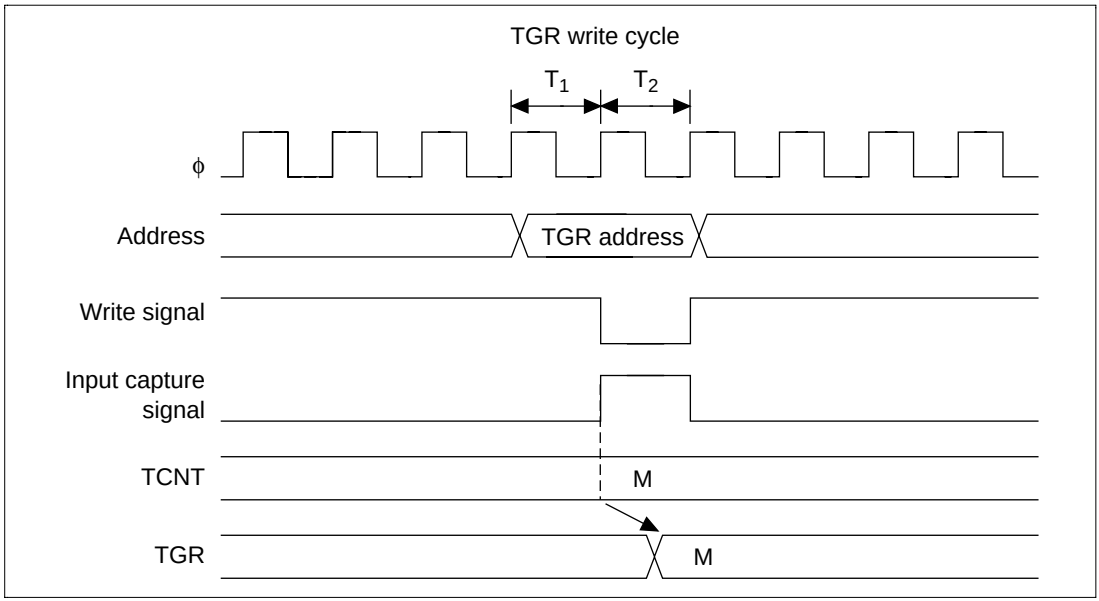


Figure 12.82 TGR Write and Input Capture Contention

12.7.8 Contention between Buffer Register Write and Input Capture

If an input capture signal is issued in the T_2 state of the buffer write cycle, write to the buffer register does not occur, and buffer operation takes priority (figure 12.83).

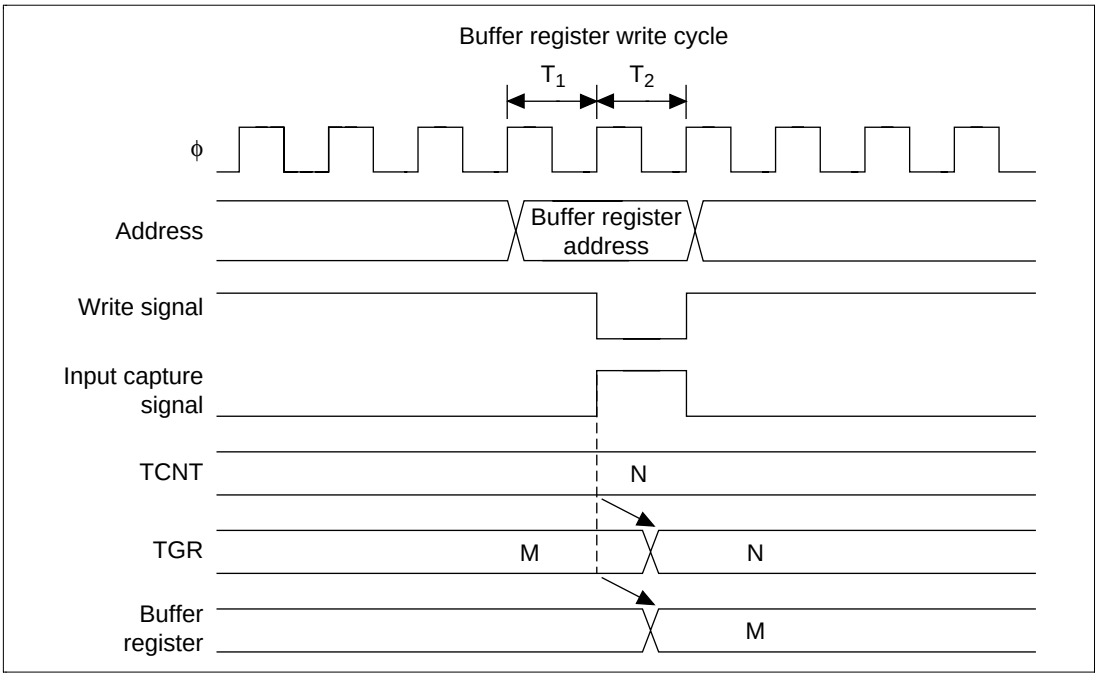


Figure 12.83 Buffer Register Write and Input Capture Contention

12.7.9 Contention between TGR Write and Compare Match

If a compare-match occurs in the T_2 state of the TGR write cycle, data is written to the TGR and a compare-match signal is issued (figure 12.84).

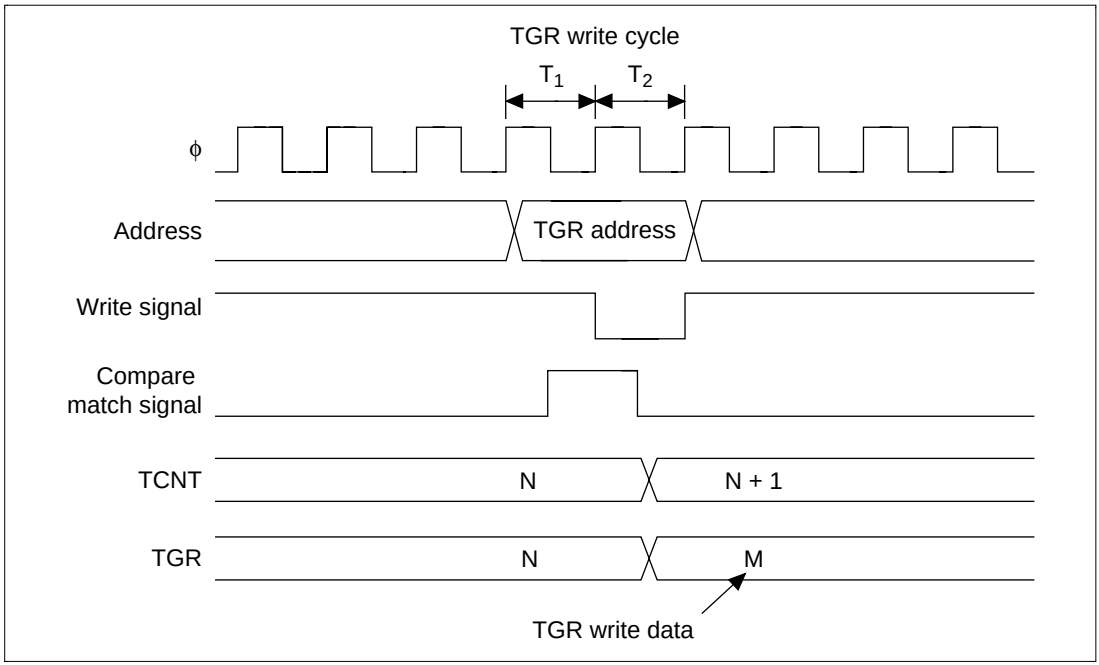


Figure 12.84 TGR Write and Compare Match Contention

12.7.10 TCNT2 Write and Overflow/Underflow Contention in Cascade Connection

With timer counters TCNT1 and TCNT2 in a cascade connection, when a contention occurs during TCNT1 count (during a TCNT2 overflow/underflow) in the T_2 state of the TCNT2 write cycle, the write to TCNT2 is conducted, and the TCNT1 count signal is prohibited. At this point, if there is match with TGR1A or TGR1B and the TCNT1 value, a compare signal is issued. The timing is shown in figure 12.85.

For cascade connections, be sure to synchronize settings for channels 1 and 2 when setting TCNT clearing.

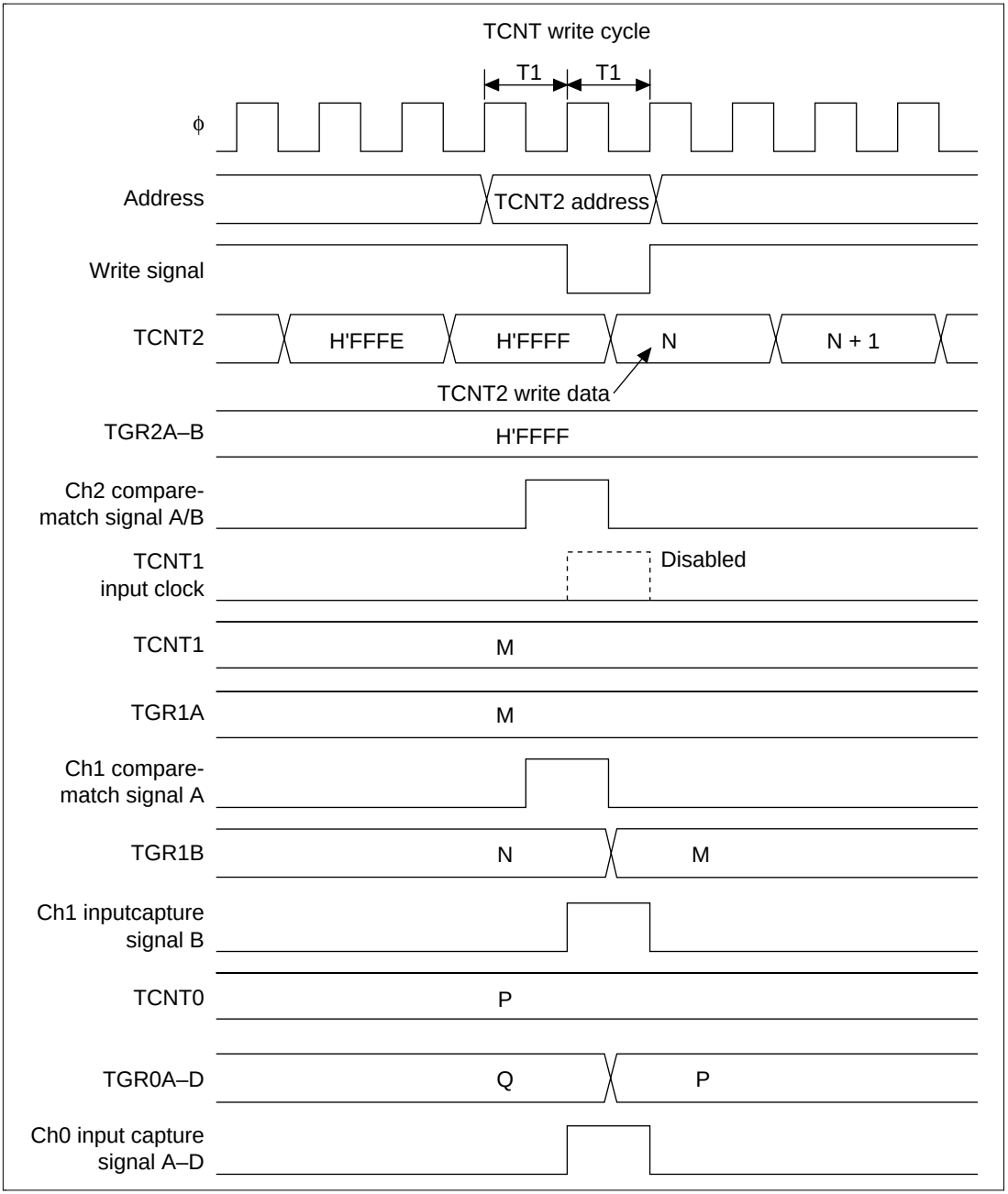


Figure 12.85 TCNT2 Write and Overflow/Underflow Contention with Cascade Connection

12.7.11 Counter Value during Complementary PWM Mode Stop

When counting operation is suspended with TCNT3 and TCNT4 in complementary PWM mode, TCNT3 has the timer dead time register (TDDR) value, and TCNT4 is held at H'0000.

When restarting complementary PWM mode, counting begins automatically from the initialized state (figure 12.86).

When counting begins in another operating mode, be sure that TCNT3 and TCNT4 are set to the initial values.

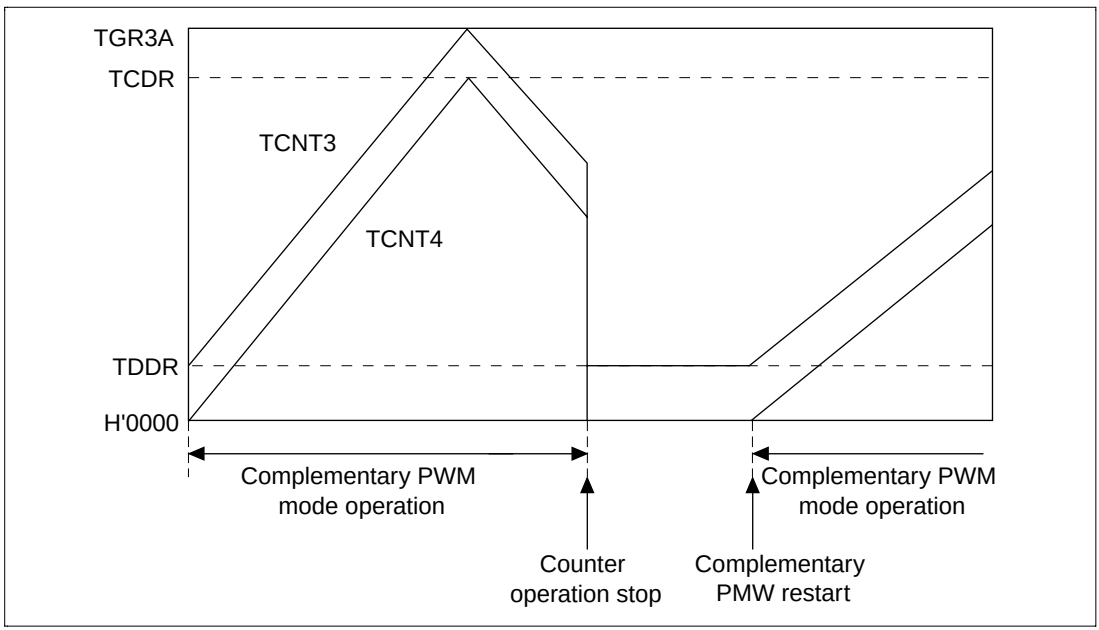


Figure 12.86 Counter Value during Complementary PWM Mode Stop

12.7.12 Buffer Operation Setting in Complementary PWM Mode

In complementary PWM mode, conduct rewrites by buffer operation for the PWM cycle setting register (TGR3A), PWM carrier cycle setting register (TCDR) and duty setting registers (TGR3B, TRG4A, and TGR4B).

In complementary PWM mode, channel 3 and channel 4 buffers operate in accordance with bit settings BFA and BFB of TMDR3. When TMDR3's BFA bit is set to 1, TGR3C functions as a buffer register for TGR3A. At the same time, TGR4C functions as the buffer register for TRG4A, while the TCBR functions as the TCDR's buffer register.

12.7.13 Reset Sync PWM Mode Buffer Operation and Compare Match Flag

When setting buffer operation for reset sync PWM mode, set the BFA and BFB bits of TMDR4 to 0. The TIOC4C pin will be unable to produce its waveform output if the BFA bit of TMDR4 is set to 1.

In reset sync PWM mode, the channel 3 and channel 4 buffers operate in accordance with the BFA and BFB bit settings of TMDR3. For example, if the BFA bit of TMDR3 is set to 1, TGR3C functions as the buffer register for TGR3A. At the same time, TGR4C functions as the buffer register for TRG4A.

When setting buffer operation for reset sync PWM mode, take particular care, since compare-match flag TGFC bit and TGFD bit operations differ with TSR3 and TSR4.

The TGFC bit and TGFD bit of TSR3 are not set when TGR3C and TGR3D are operating as buffer registers. On the other hand, TSR4's TGFC and TGFD bits are set even when TGR4C and TGR4D are operating as buffer registers.

When buffer operation has been set for reset sync PWM mode, set the timer interrupt enable register's (TIER4) TGIEC and TGIED bits to 0, to prohibit interrupt output.

Figure 12.87 shows an example of operations for TGR3, TGR4, TIOC3, and TIOC4, with TMDR3's BFA and BFB bits set to 1, and TMDR4's BFA and BFB bits set to 0.

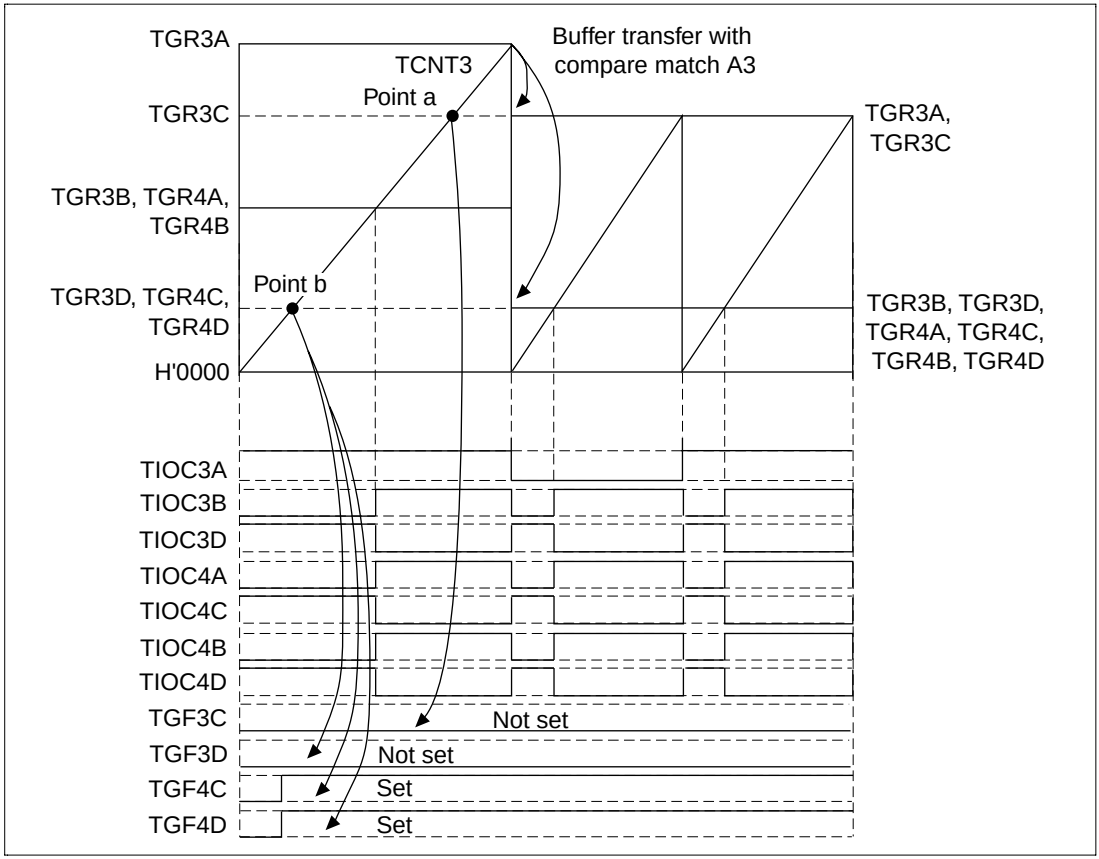


Figure 12.87 Buffer Operation and Compare-Match Flags in Reset Sync PWM Mode

- A mask operation

  For A mask, the above operation is modified as follows:

  When setting buffer operation for reset sync PWM mode, set the BFA and BFB bits of TMDR4 to 0. The TIOC4C pin will be unable to produce its waveform output if the BFA bit of TMDR4 is set to 1.

  In reset sync PWM mode, the channel 3 and channel 4 buffers operate in accordance with the BFA and BFB bit settings of TMDR3. For example, if the BFA bit of TMDR3 is set to 1, TGR3C functions as the buffer register for TGR3A. At the same time, TGR4C functions as the buffer register for TRG4A.

  When setting buffer operation for reset sync PWM mode, the compare-match flag TGFC bit and TGFD bit operations will be the same for TSR3 and TSR4.

  The TGFC bit and TGFD bit of TSR3 are not set when TGR3C and TGR3D are operating as buffer registers. The TGFC bit and TGFD bit of TSR4 are not set when TGR4C and TGR4D are operating as buffer registers.

When setting the buffer operation in the reset synchronous PWM mode, it is not necessary to set the timer interrupt enable register's (TIER4) TGIEC and TGIED bits to 0, to prohibit interrupt output.

Figure 12.88 shows an example of operations for TGR3, TGR4, TIOC3, and TIOC4, with TMDR3's BFA and BFB bits set to 1, and TMDR4's BFA and BFB bits set to 0.

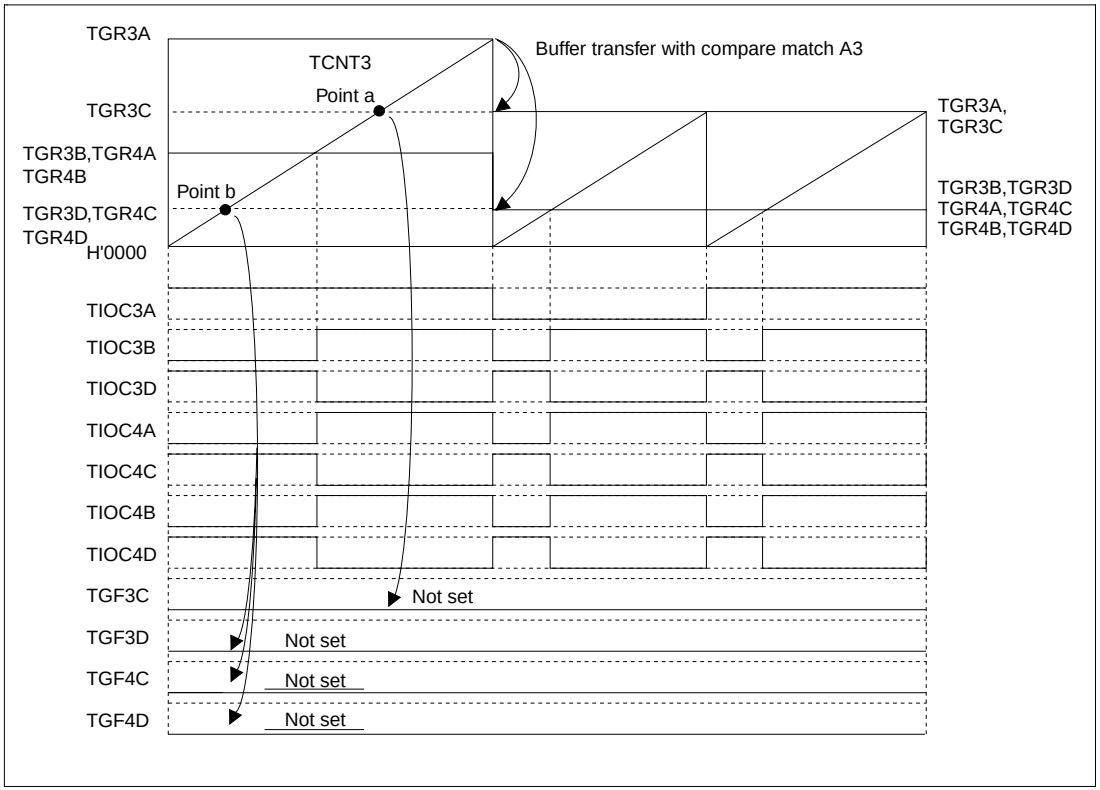


Figure 12.88 Buffer Operation and Compare-Match Flags in Reset Sync PWM Mode (for A Mask)

12.7.14 Overflow Flags in Reset Sync PWM Mode

When set to reset sync PWM mode, TCNT3 and TCNT4 start counting when the CST3 bit of TSTR is set to 1. At this point, TCNT4's count clock source and count edge obey the TCR3 setting.

In reset sync PWM mode, with cycle register TGR3A's set value at H'FFFF, take care when specifying TGR3A compare-match for the counter clear source, since the operation of the overflow flag (TCFV bit) differs with TSR3 and TSR4.

When TCNT3 and TCNT4 count up to H'FFFF, a compare-match occurs with TGR3A, and TCNT3 and TCNT4 are both cleared. At this point, TSR3's TCFV bit is not set, but the TCFV bit of TSR4 is set.

This can be avoided by sync setting for channel 3 and channel 4. Set the SYNC3 and SYNC4 bits of the timer sync register (TSYR) to 1, compare-match with TGR3A by TCR3 for the counter clear source, and sync clear with TCR4. This gives the sync setting for channels 3 and 4.

Figure 12.89 shows a TCFV bit operation example in reset sync PWM mode with a set value for cycle register TGR3A of H'FFFF, when a TGR3A compare-match has been specified without synchronous setting for the counter clear source.

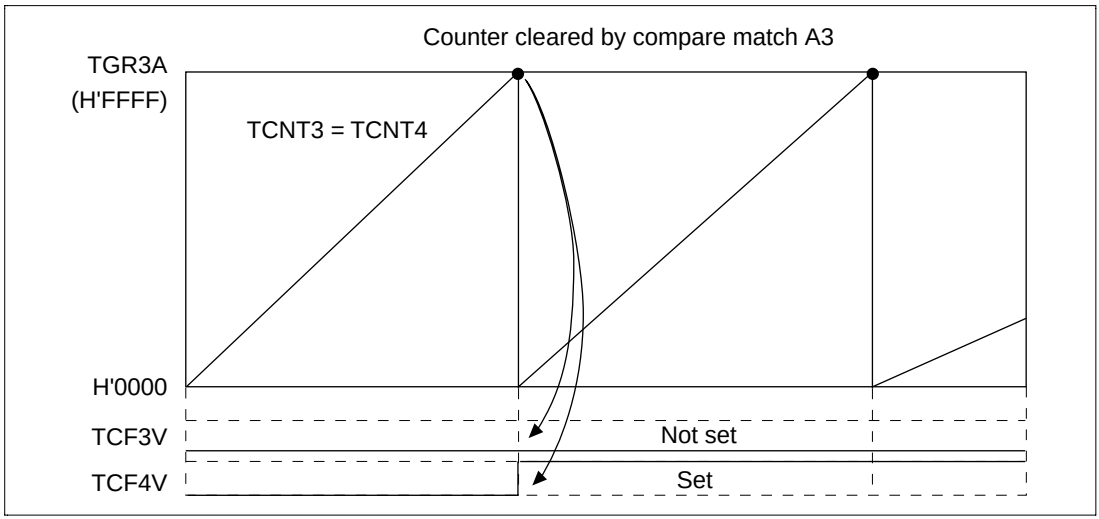


Figure 12.89 Reset Sync PWM Mode Overflow Flag

- A mask operation

  For A mask, the above operation is modified as follows:

  When set to reset sync PWM mode, TCNT3 and TCNT4 start counting when the CST3 bit of TSTR is set to 1. At this point, TCNT4's count clock source and count edge obey the TCR3 setting.

  In reset sync PWM mode, with cycle register TGR3A's set value at H'FFFF and specifying TGR3A compare-match for the counter clear source, the operation of the overflow flag TCFV bit for both TSR3 and TSR4 will be the same.

  When TCNT3 and TCNT4 count up to H'FFFF, a compare-match occurs with TGR3A, and TCNT3 and TCNT4 are both cleared. At this point, TCFV bits for TSR3 and TSR4 are not set.

  Figure 12.90 shows a TCFV bit operation example in reset sync PWM mode with a set value for cycle register TGR3A of H'FFFF, when a TGR3A compare-match has been specified without synchronous setting for the counter clear source.

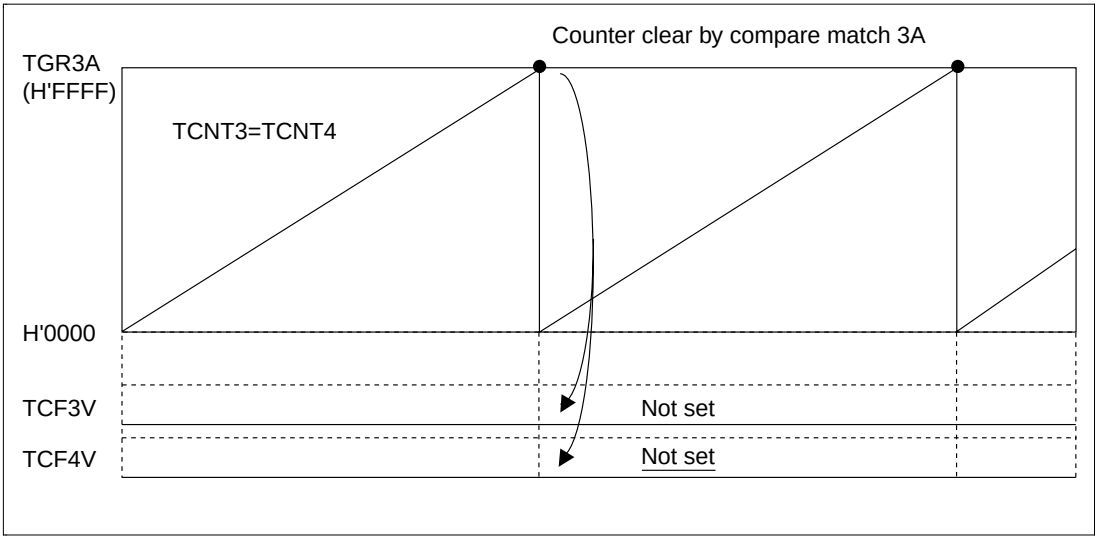


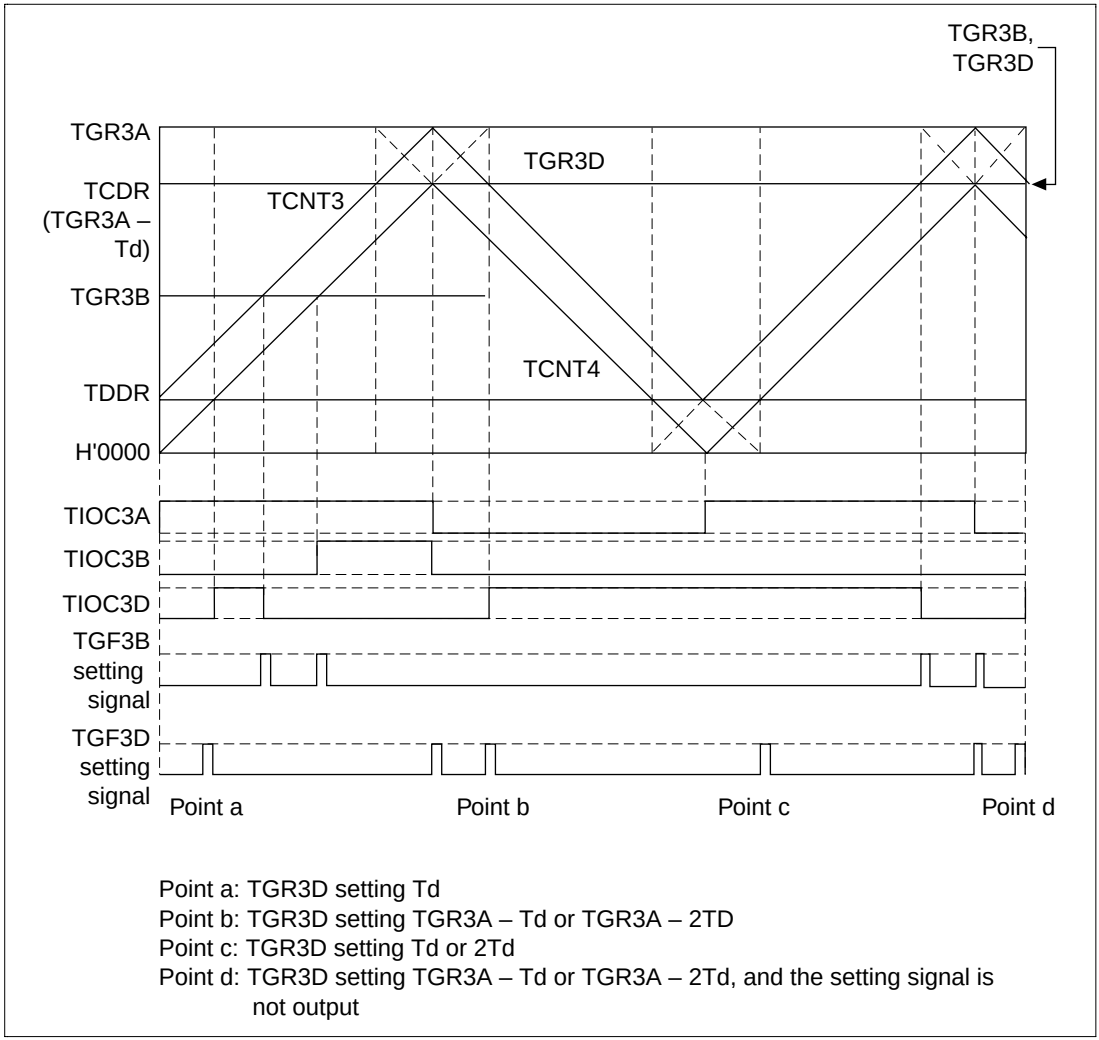
Figure. 12.90 Reset Sync PWM Mode Overflow Flag (for A Mask)

12.7.15 Notes on Compare Match Flags in Complementary PWM Mode

In complementary PWM mode, buffer register compare-match flags can be set only for compare with three counters (TCNT3, TCNT4, and TCNTS).

Note that when the buffer register set value is dead time (Td), 2Td, TGR3A – Td, or TGR3A – 2Td, the buffer register compare-match flag may not be set.

Figure 12.91 gives a description when TGR3B is the specified duty setting register, TGR3D the buffer register with TGR3A – Td as the buffer register set value.



Point a: TGR3D setting Td
Point b: TGR3D setting TGR3A – Td or TGR3A – 2TD
Point c: TGR3D setting Td or 2Td
Point d: TGR3D setting TGR3A – Td or TGR3A – 2Td, and the setting signal is not output

Figure 12.91 Special Properties of Compare Match Flag in Complementary PWM Mode

- A mask operation

  For A mask, the above operation is modified as follows:

  In complementary PWM mode, buffer register compare-match flags can be set only for compare with three counters (TCNT3, TCNT4, and TCNTS).

  Special properties of compare match flag disappear and compare match flags of buffer registers are set to all set values of buffer registers.

  Figure 12.92 shows an example when setting the duty setting register to TGR3B, buffer register to TGR3D and Buffer register to TGR3A-Td.

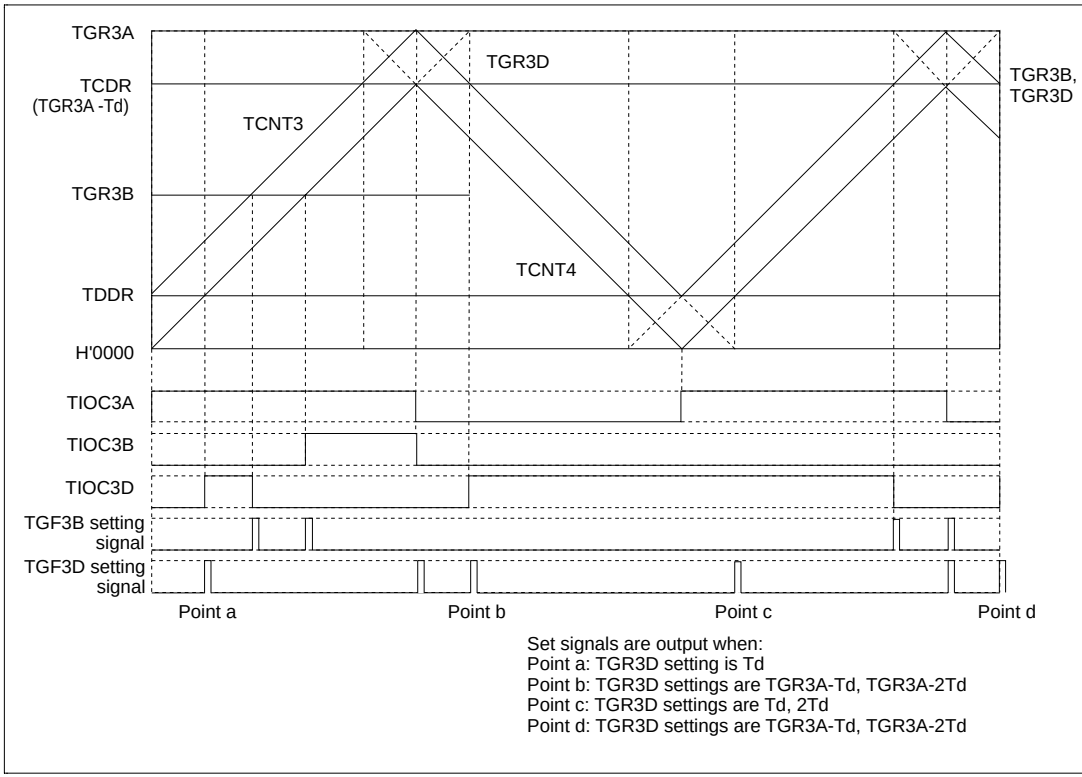


Figure. 12.92 Special Properties of Compare Match Flag in Complementary PWM Mode (for A Mask)

12.7.16 Contention between Overflow/Underflow and Counter Clearing

If overflow/underflow and counter clearing occur simultaneously, the TCFV/TCFU flag in TSR is not set and TCNT clearing takes precedence.

Figure 12.93 shows the operation timing when a TGR compare-match is specified as the clearing source, and H'FFFF is set in TGR.

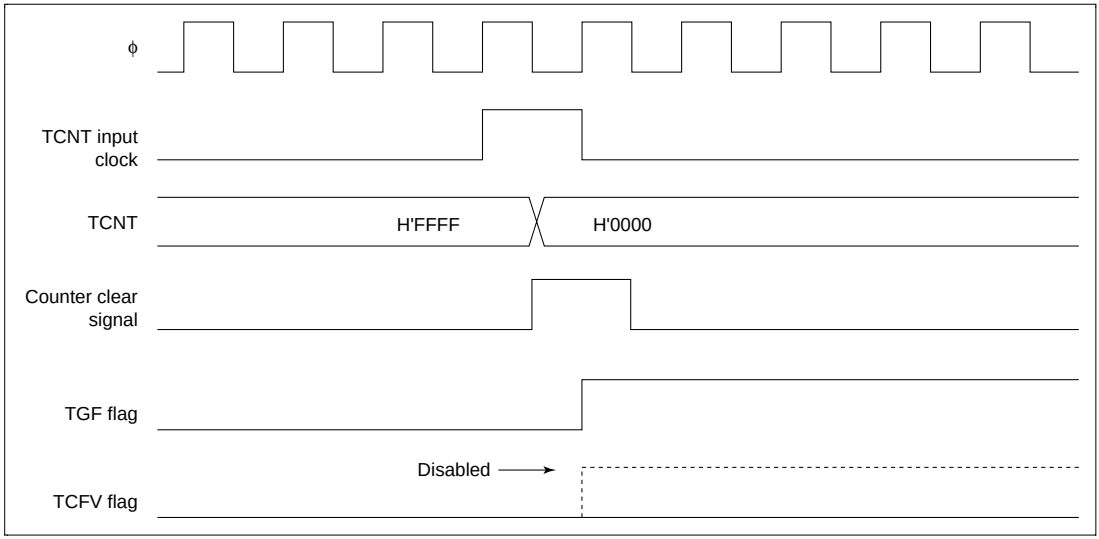


Figure 12.93 Contention between Overflow and Counter Clearing

12.7.17 Contention between TCNT Write and Overflow/Underflow

If there is an up-count or down-count in the T2 state of a TCNT write cycle, and overflow/underflow occurs, the TCNT write takes precedence and the TCFV/TCFU flag in TSR is not set .

Figure 12.94 shows the operation timing in this case.

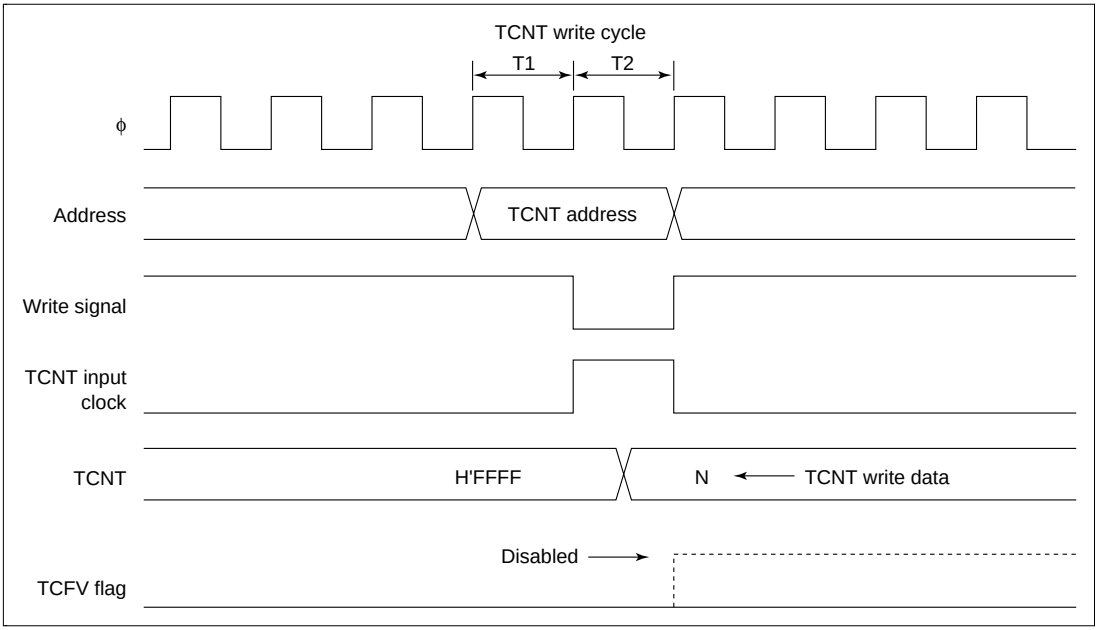


Figure 12.94 Contention between TCNT Write and Overflow

12.7.18 Cautions on Transition from Normal Operation or PWM Mode 1 to Reset-Synchronous PWM Mode

When making a transition from channel 3 or 4 normal operation or PWM mode 1 to reset-synchronous PWM mode, if the counter is halted with the output pins (TIOC3B, TIOC3D, TIOC4A, TIOC4C, TIOC4B, TIOC4D) in the high-impedance state, followed by the transition to reset-synchronous PWM mode and operation in that mode, the initial pin output will not be correct.

When making a transition from normal operation to reset-synchronous PWM mode, write H'11 to registers TIOR3H, TIOR3L, TIOR4H, and TIOR4L to initialize the output pins to low level output, then set an initial register value of H'00 before making the mode transition.

When making a transition from PWM mode 1 to reset-synchronous PWM mode, first switch to normal operation, then initialize the output pins to low level output and set an initial register value of H'00 before making the transition to reset-synchronous PWM mode.

12.7.19 Output Level in Complementary PWM Mode and Reset-Synchronous PWM Mode

When channels 3 and 4 are in complementary PWM mode or reset-synchronous PWM mode, the PWM waveform output level is set with the OLSP and OLSN bits in the timer output control register (TOCR). In the case of complementary PWM mode or reset-synchronous PWM mode, TIOR should be set to H'00.

12.7.20 Cautions on Using the Chopping Function in Complementary PWM Mode or Reset Synchronous PWM Mode (A Mask Excluded)

When channels 3 and 4 are in complementary PWM mode or reset-synchronous PWM mode and using the chopping output function, setting the PWM waveform output level to low active with the OLSP and OLSN bits in the timer output control register (TOCR) will output an incorrect gate signal or chopping output.

When channels 3 and 4 are in complementary PWM mode or reset-synchronous PWM mode and using the chopping output function, the PWM output level should be set to high active.

12.7.21 Cautions on Carrying Out Buffer Operation of Channel 0 in PWM Mode (A Mask Excluded)

In PWM mode 1, the TGRA and TGRB registers are used in pairs and PWM waveform is output to the TIOCA pin. In the same manner, the TGRC and TGRD registers are used in pairs and PWM waveform is output to the TIOCC pin. If either the TGRC or TGRD register is operating as a buffer register, the TIOCC pin cannot execute default output setting or PWM waveform output with the I/O control register (TIOR).

Note that for channel 0, the TIOCC pin allows both default output setting by TIOR and PWM output when setting buffer operation only for the TGRD register in PWM mode.

When using channel 0 in PWM mode 1 and setting buffer operation, use both the TGRC and TGRD registers as buffer registers.

12.7.22 Cautions on Restarting with Sync Clear of Another Channel in Complementary PWM Mode (A Mask Excluded)

The complementation PWM mode operates while waiting for the current, next and the following set values as values of the PWM duty. If clearing sync from another channel during operation, the PWM output will return to the default output and restart.

When restarting with sync clear, the following operations may occur:

1. When restarting with sync clear, the next set value is used for the PWM duty, however, the following set value may be used by mistake.
2. If sync clear and the setting of the value following the next value of PWM duty (write to TGR4D) occurs at the same time, the next set value may be overwritten.

How to avoid 1

When selecting the mode to transfer using the crest/trough in the complementary PWM transfer mode, set the value following the next value of the PWM duty (write to TGR4D) while the temporary register is not executing comparisons. Furthermore, set the occurrence timing of sync clear while the temporary register is not executing comparisons.

When selecting the mode to transfer using the crest in the transfer mode, set the value following the next value of the PWM duty (write to TGR4D) while the temporary register is not executing comparisons and while TCNT3 and TCNT4 are counting up. Furthermore, set the occurrence timing of sync clear while the temporary register is not executing comparisons and while TCNT3 and TCNT4 are counting up.

When selecting the mode to transfer using the trough in the transfer mode, set the value following the next value of the PWM duty (write to TGR4D) while the temporary register is not executing comparisons and while TCNT3 and TCNT4 are counting down. Furthermore, set the occurrence timing of sync clear while the temporary register is not executing comparisons and while TCNT3 and TCNT4 are counting down.

How to avoid 2

Regardless of the transfer mode, set so that the sync clear and the setting of the value following the next value (write to TGR4D) does not occur at the same time.

Figure 12.95 shows an example of the duration while the temporary register is executing comparisons. initial TB, TA, and TB indicate the duration of the temporary register comparison.

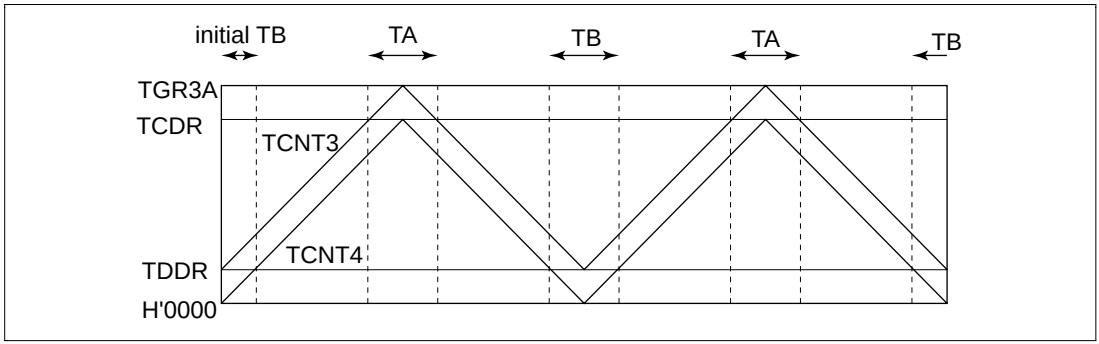


Figure. 12.95 Temporary Register Comparison Execution Time

12.8 MTU Output Pin Initialization

12.8.1 Operating Modes

The MTU has the following six operating modes. Waveform output is possible in all of these modes.

- Normal mode (channels 0 and 4)
- PWM mode 1 (channels 0 and 4)
- PWM mode 2 (channels 0 and 2)
- Phase counting modes 1–4 (channels 1 and 2)
- Complementary PWM mode (channels 3 and 4)
- Reset-synchronous PWM mode (channels 3 and 4)

The MTU output pin initialization method for each of these modes is described in this section.

12.8.2 Reset Start Operation

The MTU output pins (TIOC*) are initialized low by a reset and in standby mode. Since MTU pin function selection is performed by the pin function controller (PFC), when the PFC is set, the MTU pin states at that point are output to the ports. When MTU output is selected by the PFC immediately after a reset, the MTU output initial level, low, is output directly at the port. When the active level is low, the system will operate at this point, and therefore the PFC setting should be made after initialization of the MTU output pins is completed.

Note: Channel number and port notation are substituted for *.

12.8.3 Operation in Case of Re-Setting Due to Error During Operation, Etc.

If an error occurs during MTU operation, MTU output should be cut by the system. Cutoff is performed by switching the pin output to port output with the PFC and outputting the inverse of the active level. For large-current pins, output can also be cut by hardware, using port output enable (POE). The pin initialization procedures for re-setting due to an error during operation, etc., and the procedures for restarting in a different mode after re-setting, are shown below.

The MTU has six operating modes, as stated above. There are thus 36 mode transition combinations, but some transitions are not available with certain channel and mode combinations. Possible mode transition combinations are shown in table 12.18.

Table 12.18 Mode Transition Combinations

	After					
Before	**Normal**	**PWM1**	**PWM2**	**PCM**	**CPWM**	**RPWM**
Normal	(1)	(2)	(3)	(4)	(5)	(6)
PWM1	(7)	(8)	(9)	(10)	(11)	(12)
PWM2	(13)	(14)	(15)	(16)	None	None
PCM	(17)	(18)	(19)	(20)	None	None
CPWM	(21)	(22)	None	None	(23)	(24)
RPWM	(25)	(26)	None	None	(27)	(28)

Legend:
Normal: Normal mode
PWM1: PWM mode 1
PWM2: PWM mode 2
PCM: Phase counting modes 1–4
CPWM: Complementary PWM mode
RPWM: Reset-synchronous PWM mode

The above abbreviations are used in some places in following descriptions.

12.8.4 Overview of Initialization Procedures and Mode Transitions in Case of Error during Operation, Etc.

- When making a transition to a mode (Normal, PWM1, PWM2, PCM) in which the pin output level is selected by the timer I/O control register (TIOR) setting, initialize the pins by means of a TIOR setting.
- In PWM mode 1, since a waveform is not output to the TIOC*B (TIOC *D) pin, setting TIOR will not initialize the pins. If initialization is required, carry it out in normal mode, then switch to PWM mode 1.

- In PWM mode 2, since a waveform is not output to the cycle register pin, setting TIOR will not initialize the pins. If initialization is required, carry it out in normal mode, then switch to PWM mode 2.
- In normal mode or PWM mode 2, if TGRC and TGRD operate as buffer registers, setting TIOR will not initialize the buffer register pins. If initialization is required, clear buffer mode, carry out initialization, then set buffer mode again.
- In PWM mode 1, if either TGRC or TGRD operates as a buffer register, setting TIOR will not initialize the TGRC pin. To initialize the TGRC pin, clear buffer mode, carry out initialization, then set buffer mode again.
- When making a transition to a mode (CPWM, RPWM) in which the pin output level is selected by the timer output control register (TOCR) setting, switch to normal mode and perform initialization with TIOR, then restore TIOR to its initial value, and temporarily disable channel 3 and 4 output with the timer output master enable register (TOER). Then operate the unit in accordance with the mode setting procedure (TOCR setting, TMDR setting, TOER setting).

Pin initialization procedures are described below for the numbered combinations in table 12.19. The active level is assumed to be low.

Note: Channel number is substituted for * indicated in this article.

(1) Operation when Error Occurs during Normal Mode Operation, and Operation is Restarted in Normal Mode: Figure 12.96 shows an explanatory diagram of the case where an error occurs in normal mode and operation is restarted in normal mode after re-setting.

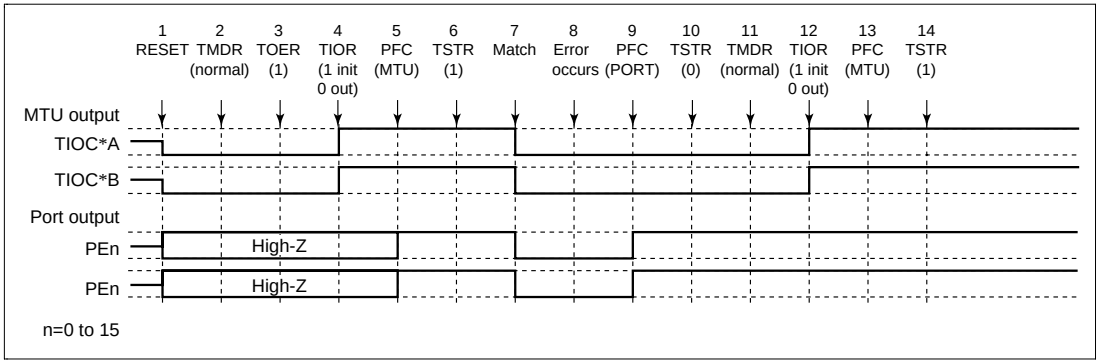


Figure 12.96 Error Occurrence in Normal Mode, Recovery in Normal Mode

1. After a reset, MTU output is low and ports are in the high-impedance state.
2. After a reset, the TMDR setting is for normal mode.
3. For channels 3 and 4, enable output with TOER before initializing the pins with TIOR.
4. Initialize the pins with TIOR. (The example shows initial high output, with low output on compare-match occurrence.)
5. Set MTU output with the PFC.
6. The count operation is started by TSTR.
7. Output goes low on compare-match occurrence.
8. An error occurs.
9. Set port output with the PFC and output the inverse of the active level.
10. The count operation is stopped by TSTR.
11. Not necessary when restarting in normal mode.
12. Initialize the pins with TIOR.
13. Set MTU output with the PFC.
14. Operation is restarted by TSTR.

(2) Operation when Error Occurs during Normal Mode Operation, and Operation is Restarted in PWM Mode 1: Figure 12.97 shows an explanatory diagram of the case where an error occurs in normal mode and operation is restarted in PWM mode 1 after re-setting.

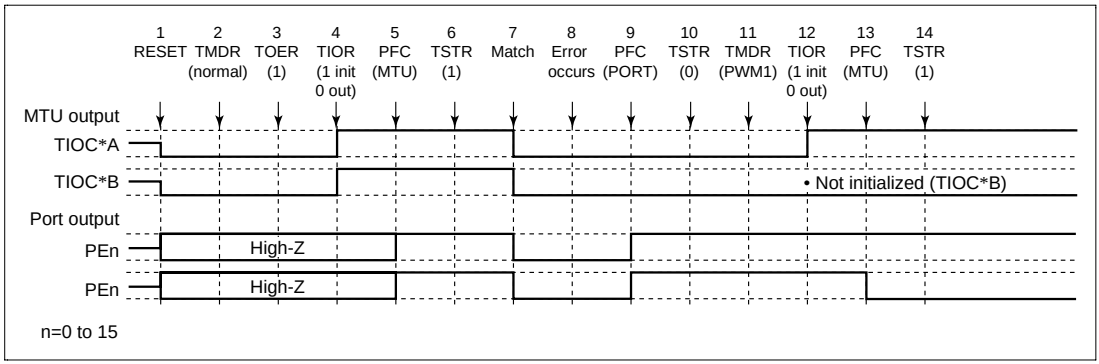


Figure 12.97 Error Occurrence in Normal Mode, Recovery in PWM Mode 1

1 to 10 are the same as in figure 12.96.

11. Set PWM mode 1.
12. Initialize the pins with TIOR. (In PWM mode 1, the TIOC*B side is not initialized. If initialization is required, initialize in normal mode, then switch to PWM mode 1.)
13. Set MTU output with the PFC.
14. Operation is restarted by TSTR.

(3) Operation when Error Occurs during Normal Mode Operation, and Operation is Restarted in PWM Mode 2: Figure 12.98 shows an explanatory diagram of the case where an error occurs in normal mode and operation is restarted in PWM mode 2 after re-setting.

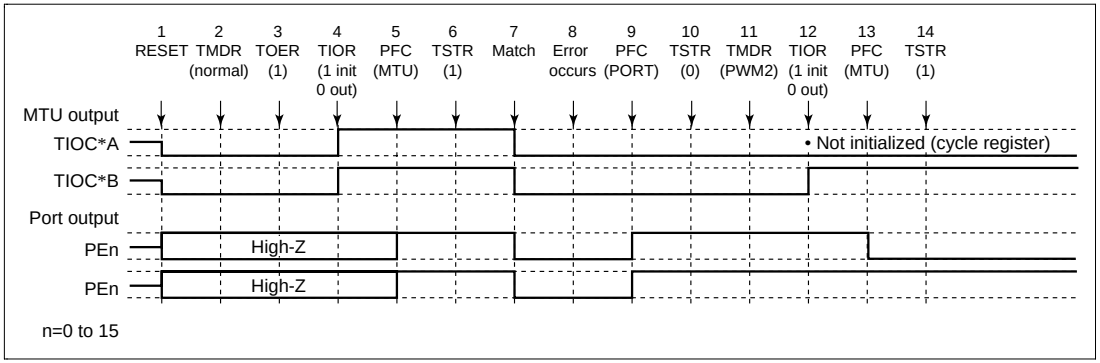


Figure 12.98 Error Occurrence in Normal Mode, Recovery in PWM Mode 2

1 to 10 are the same as in figure 12.96.

11. Set PWM mode 2.
12. Initialize the pins with TIOR. (In PWM mode 2, the cycle register pins are not initialized. If initialization is required, initialize in normal mode, then switch to PWM mode 2.)
13. Set MTU output with the PFC.
14. Operation is restarted by TSTR.

Note: PWM mode 2 can only be set for channels 0–2, and therefore TOER setting is not necessary.

(4) Operation when Error Occurs during Normal Mode Operation, and Operation is Restarted in Phase Counting Mode: Figure 12.99 shows an explanatory diagram of the case where an error occurs in normal mode and operation is restarted in phase counting mode after re-setting.

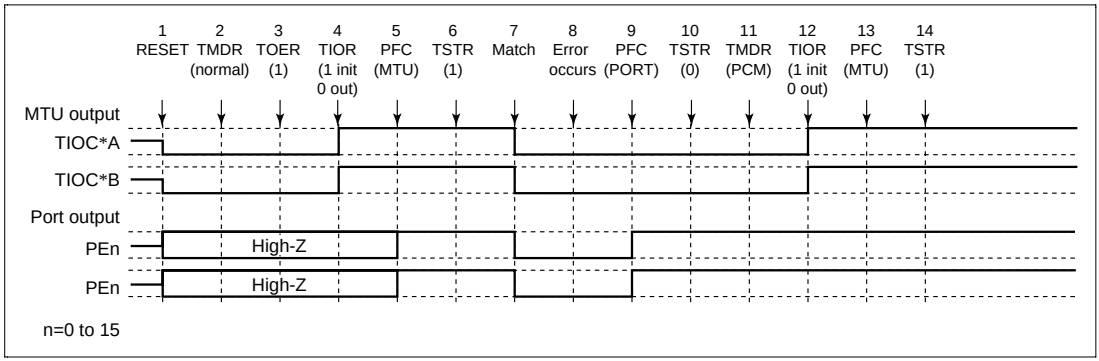


Figure 12.99 Error Occurrence in Normal Mode, Recovery in Phase Counting Mode

1 to 10 are the same as in figure 12.96.

11. Set phase counting mode.
12. Initialize the pins with TIOR.
13. Set MTU output with the PFC.
14. Operation is restarted by TSTR.

Note: Phase counting mode can only be set for channels 1 and 2, and therefore TOER setting is not necessary.

(5) Operation when Error Occurs during Normal Mode Operation, and Operation is Restarted in Complementary PWM Mode: Figure 12.100 shows an explanatory diagram of the case where an error occurs in normal mode and operation is restarted in complementary PWM mode after re-setting.

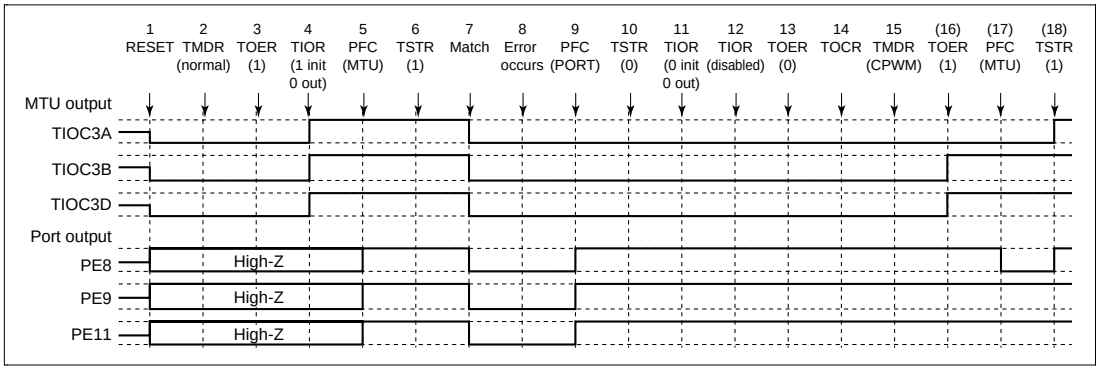


Figure 12.100 Error Occurrence in Normal Mode, Recovery in Complementary PWM Mode

1 to 10 are the same as in figure 12.96.

11. Initialize the normal mode waveform generation section with TIOR.
12. Disable operation of the normal mode waveform generation section with TIOR.
13. Disable channel 3 and 4 output with TOER.
14. Select the complementary PWM output level and cyclic output enabling/disabling with TOCR.
15. Set complementary PWM.
16. Enable channel 3 and 4 output with TOER.
17. Set MTU output with the PFC.
18. Operation is restarted by TSTR.

(6) Operation when Error Occurs during Normal Mode Operation, and Operation is Restarted in Reset-Synchronous PWM Mode: Figure 12.101 shows an explanatory diagram of the case where an error occurs in normal mode and operation is restarted in reset-synchronous PWM mode after re-setting.

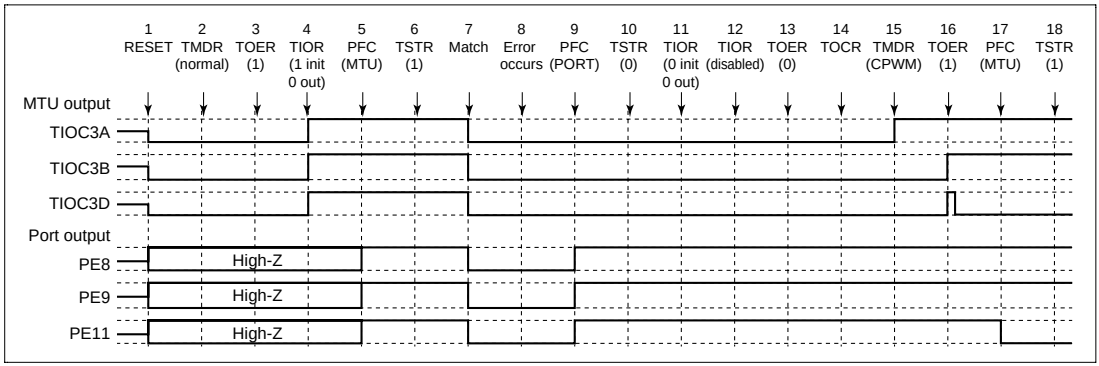


Figure 12.101 Error Occurrence in Normal Mode, Recovery in Reset-Synchronous PWM Mode

1 to 13 are the same as in figure 12.100.

14. Select the reset-synchronous PWM output level and cyclic output enabling/disabling with TOCR.
15. Set reset-synchronous PWM.
16. Enable channel 3 and 4 output with TOER.
17. Set MTU output with the PFC.
18. Operation is restarted by TSTR.

(7) Operation when Error Occurs during PWM Mode 1 Operation, and Operation is Restarted in Normal Mode: Figure 12.102 shows an explanatory diagram of the case where an error occurs in PWM mode 1 and operation is restarted in normal mode after re-setting.

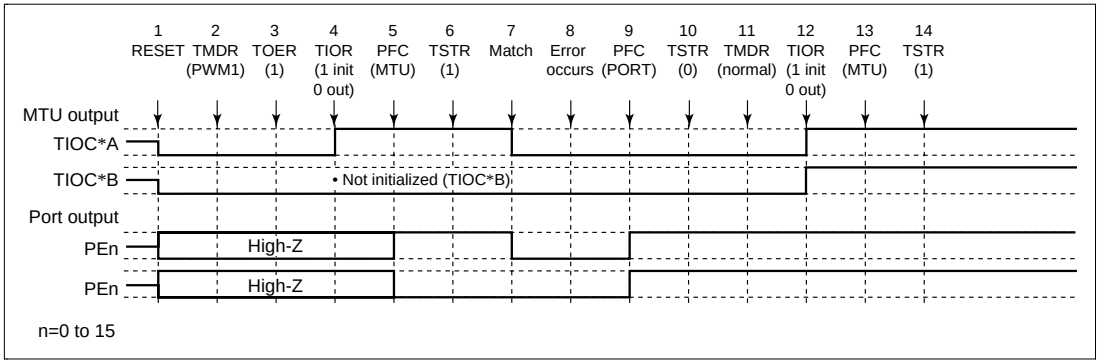


Figure 12.102 Error Occurrence in PWM Mode 1, Recovery in Normal Mode

1. After a reset, MTU output is low and ports are in the high-impedance state.
2. Set PWM mode 1.
3. For channels 3 and 4, enable output with TOER before initializing the pins with TIOR.
4. Initialize the pins with TIOR. (The example shows initial high output, with low output on compare-match occurrence. In PWM mode 1, the TIOC*B side is not initialized.)
5. Set MTU output with the PFC.
6. The count operation is started by TSTR.
7. Output goes low on compare-match occurrence.
8. An error occurs.
9. Set port output with the PFC and output the inverse of the active level.
10. The count operation is stopped by TSTR.
11. Set normal mode.
12. Initialize the pins with TIOR.
13. Set MTU output with the PFC.
14. Operation is restarted by TSTR.

(8) Operation when Error Occurs during PWM Mode 1 Operation, and Operation is Restarted in PWM Mode 1: Figure 12.103 shows an explanatory diagram of the case where an error occurs in PWM mode 1 and operation is restarted in PWM mode 1 after re-setting.

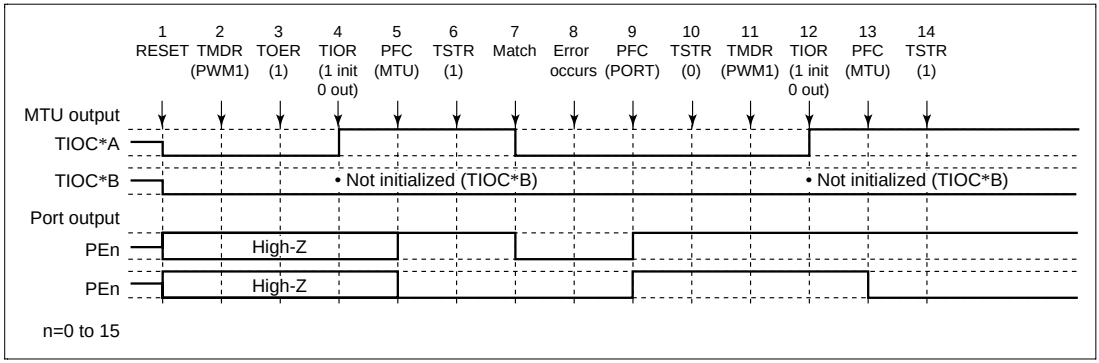


Figure 12.103 Error Occurrence in PWM Mode 1, Recovery in PWM Mode 1

1 to 10 are the same as in figure 12.102.

11. Not necessary when restarting in PWM mode 1.
12. Initialize the pins with TIOR. (In PWM mode 1, the TIOC*B side is not initialized.)
13. Set MTU output with the PFC.
14. Operation is restarted by TSTR.

(9) Operation when Rrror Occurs during PWM Mode 1 Operation, and Operation is Restarted in PWM Mode 2: Figure 12.104 shows an explanatory diagram of the case where an error occurs in PWM mode 1 and operation is restarted in PWM mode 2 after re-setting.

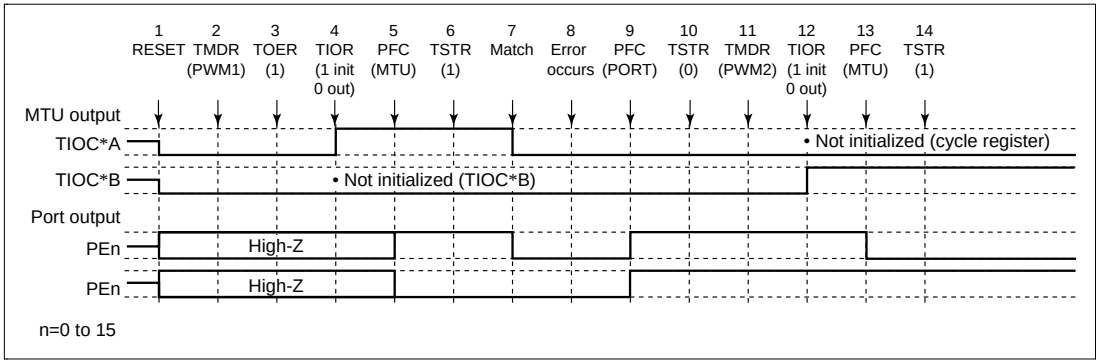


Figure 12.104 Error Occurrence in PWM Mode 1, Recovery in PWM Mode 2

1 to 10 are the same as in figure 12.102.

11. Set PWM mode 2.
12. Initialize the pins with TIOR. (In PWM mode 2, the cycle register pins are not initialized.)
13. Set MTU output with the PFC.
14. Operation is restarted by TSTR.

Note: PWM mode 2 can only be set for channels 0–2, and therefore TOER setting is not necessary.

(10) Operation when Error Occurs during PWM Mode 1 Operation, and Operation is Restarted in Phase Counting Mode: Figure 12.105 shows an explanatory diagram of the case where an error occurs in PWM mode 1 and operation is restarted in phase counting mode after re-setting.

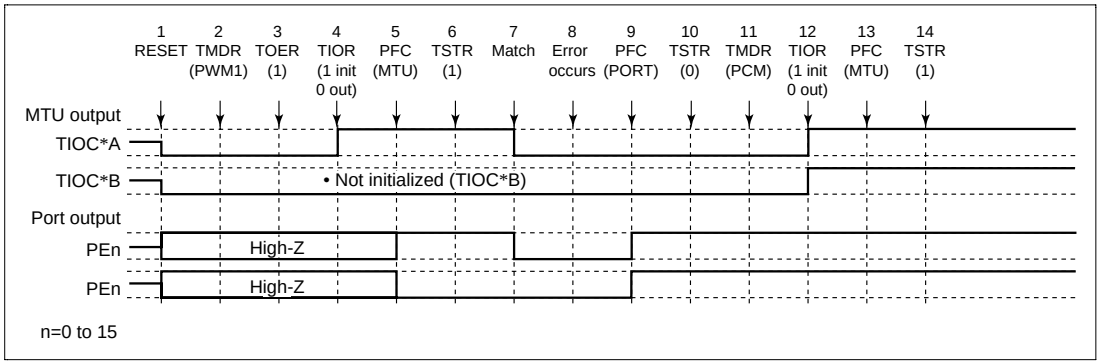


Figure 12.105 Error Occurrence in PWM Mode 1, Recovery in Phase Counting Mode

1 to 10 are the same as in figure 12.102.

11. Set phase counting mode.
12. Initialize the pins with TIOR.
13. Set MTU output with the PFC.
14. Operation is restarted by TSTR.

Note: Phase counting mode can only be set for channels 1 and 2, and therefore TOER setting is not necessary.

(11) Operation when Error Occurs during PWM Mode 1 Operation, and Operation is Restarted in Complementary PWM Mode: Figure 12.106 shows an explanatory diagram of the case where an error occurs in PWM mode 1 and operation is restarted in complementary PWM mode after re-setting.

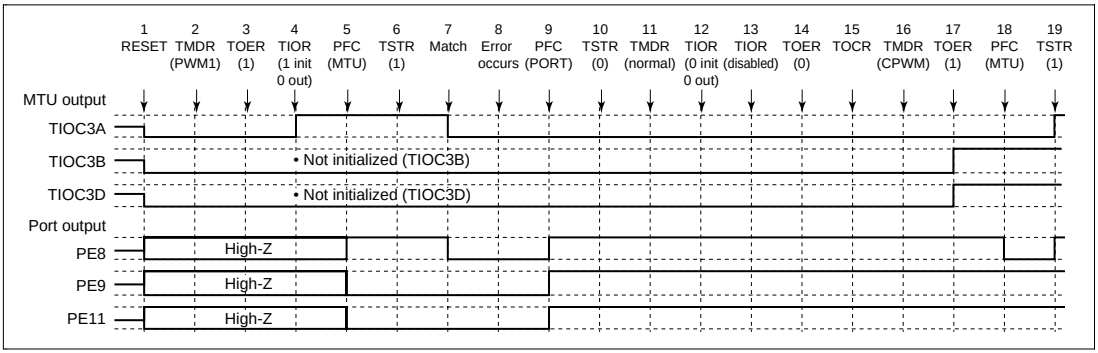


Figure 12.106 Error Occurrence in PWM Mode 1, Recovery in Complementary PWM Mode

1 to 10 are the same as in figure 12.102.

11. Set normal mode for initialization of the normal mode waveform generation section.
12. Initialize the PWM mode 1 waveform generation section with TIOR.
13. Disable operation of the PWM mode 1 waveform generation section with TIOR.
14. Disable channel 3 and 4 output with TOER.
15. Select the complementary PWM output level and cyclic output enabling/disabling with TOCR.
16. Set complementary PWM.
17. Enable channel 3 and 4 output with TOER.
18. Set MTU output with the PFC.
19. Operation is restarted by TSTR.

(12) Operation when Error Occurs during PWM Mode 1 Operation, and Operation is Restarted in Reset-Synchronous PWM Mode: Figure 12.107 shows an explanatory diagram of the case where an error occurs in PWM mode 1 and operation is restarted in reset-synchronous PWM mode after re-setting.

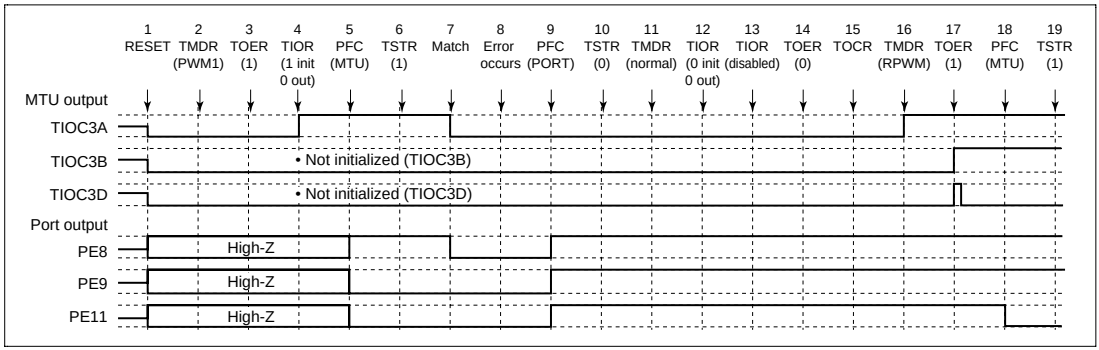


Figure 12.107 Error Occurrence in PWM Mode 1, Recovery in Reset-Synchronous PWM Mode

1 to 14 are the same as in figure 12.106.

15. Select the reset-synchronous PWM output level and cyclic output enabling/disabling with TOCR.
16. Set reset-synchronous PWM.
17. Enable channel 3 and 4 output with TOER.
18. Set MTU output with the PFC.
19. Operation is restarted by TSTR.

(13) Operation when Error Occurs during PWM Mode 2 Operation, and Operation is Restarted in Normal Mode: Figure 12.108 shows an explanatory diagram of the case where an error occurs in PWM mode 2 and operation is restarted in normal mode after re-setting.

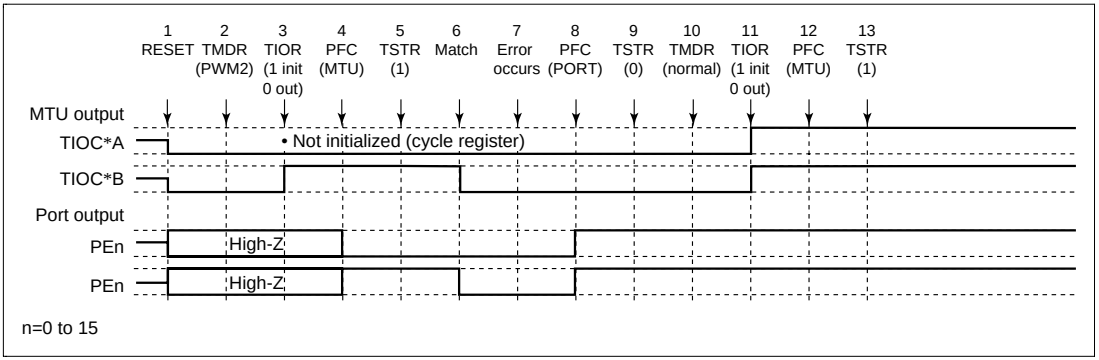


Figure 12.108 Error Occurrence in PWM Mode 2, Recovery in Normal Mode

1. After a reset, MTU output is low and ports are in the high-impedance state.
2. Set PWM mode 2.
3. Initialize the pins with TIOR. (The example shows initial high output, with low output on compare-match occurrence. In PWM mode 2, the cycle register pins are not initialized. In the example, TIOC *A is the cycle register.)
4. Set MTU output with the PFC.
5. The count operation is started by TSTR.
6. Output goes low on compare-match occurrence.
7. An error occurs.
8. Set port output with the PFC and output the inverse of the active level.
9. The count operation is stopped by TSTR.
10. Set normal mode.
11. Initialize the pins with TIOR.
12. Set MTU output with the PFC.
13. Operation is restarted by TSTR.

(14) Operation when Error Occurs during PWM Mode 2 Operation, and Operation is Restarted in PWM Mode 1: Figure 12.109 shows an explanatory diagram of the case where an error occurs in PWM mode 2 and operation is restarted in PWM mode 1 after re-setting.

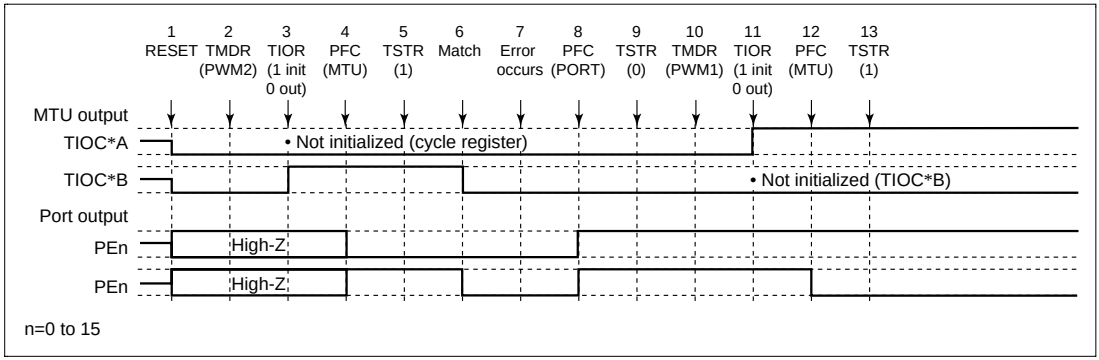


Figure 12.109 Error Occurrence in PWM Mode 2, Recovery in PWM Mode 1

1 to 9 are the same as in figure 12.108.

10. Set PWM mode 1.
11. Initialize the pins with TIOR. (In PWM mode 1, the TIOC*B side is not initialized.)
12. Set MTU output with the PFC.
13. Operation is restarted by TSTR.

(15) Operation when Error Occurs during PWM Mode 2 Operation, and Operation is Restarted in PWM Mode 2: Figure 12.110 shows an explanatory diagram of the case where an error occurs in PWM mode 2 and operation is restarted in PWM mode 2 after re-setting.

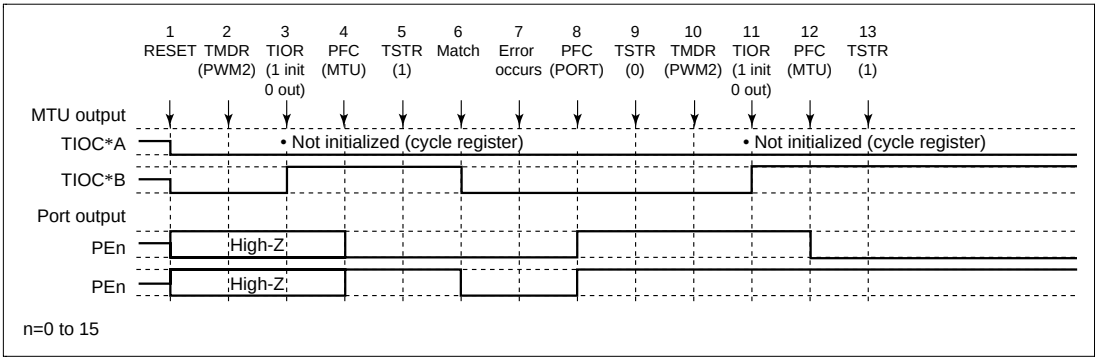


Figure 12.110 Error Occurrence in PWM Mode 2, Recovery in PWM Mode 2

1 to 9 are the same as in figure 12.108.

10. Not necessary when restarting in PWM mode 2.
11. Initialize the pins with TIOR. (In PWM mode 2, the cycle register pins are not initialized.)
12. Set MTU output with the PFC.
13. Operation is restarted by TSTR.

(16) Operation when Error Occurs during PWM Mode 2 Operation, and Operation is Restarted in Phase Counting Mode: Figure 12.111 shows an explanatory diagram of the case where an error occurs in PWM mode 2 and operation is restarted in phase counting mode after re-setting.

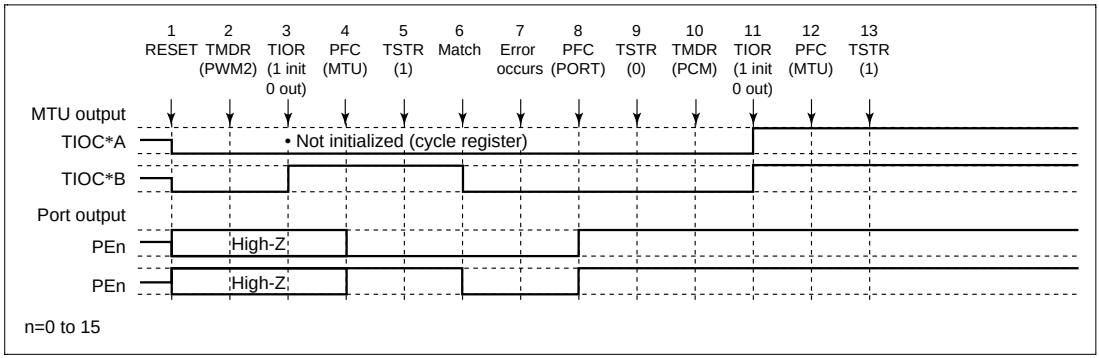


Figure 12.111 Error Occurrence in PWM Mode 2, Recovery in Phase Counting Mode

1 to 9 are the same as in figure 12.108.

10. Set phase counting mode.
11. Initialize the pins with TIOR.
12. Set MTU output with the PFC.
13. Operation is restarted by TSTR.

(17) Operation when Error Occurs during Phase Counting Mode Operation, and Operation is Restarted in Normal Mode: Figure 12.112 shows an explanatory diagram of the case where an error occurs in phase counting mode and operation is restarted in normal mode after re-setting.

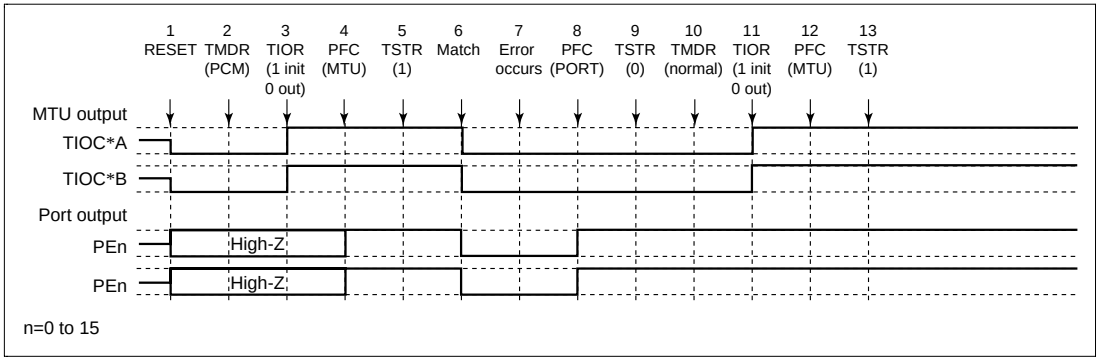


Figure 12.112 Error Occurrence in Phase Counting Mode, Recovery in Normal Mode

1. After a reset, MTU output is low and ports are in the high-impedance state.
2. Set phase counting mode.
3. Initialize the pins with TIOR. (The example shows initial high output, with low output on compare-match occurrence.)
4. Set MTU output with the PFC.
5. The count operation is started by TSTR.
6. Output goes low on compare-match occurrence.
7. An error occurs.
8. Set port output with the PFC and output the inverse of the active level.
9. The count operation is stopped by TSTR.
10. Set in normal mode.
11. Initialize the pins with TIOR.
12. Set MTU output with the PFC.
13. Operation is restarted by TSTR.

(18) Operation when Error Occurs during Phase Counting Mode Operation, and Operation is Restarted in PWM Mode 1: Figure 12.113 shows an explanatory diagram of the case where an error occurs in phase counting mode and operation is restarted in PWM mode 1 after re-setting.

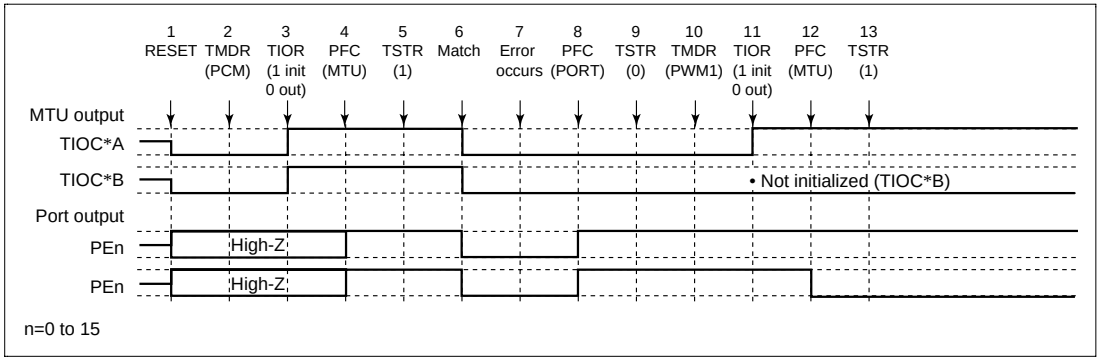


Figure 12.113 Error Occurrence in Phase Counting Mode, Recovery in PWM Mode 1

1 to 9 are the same as in figure 12.112.

10. Set PWM mode 1.
11. Initialize the pins with TIOR. (In PWM mode 1, the TIOC *B side is not initialized.)
12. Set MTU output with the PFC.
13. Operation is restarted by TSTR.

(19) Operation when Error Occurs during Phase Counting Mode Operation, and Operation is Restarted in PWM Mode 2: Figure 12.114 shows an explanatory diagram of the case where an error occurs in phase counting mode and operation is restarted in PWM mode 2 after re-setting.

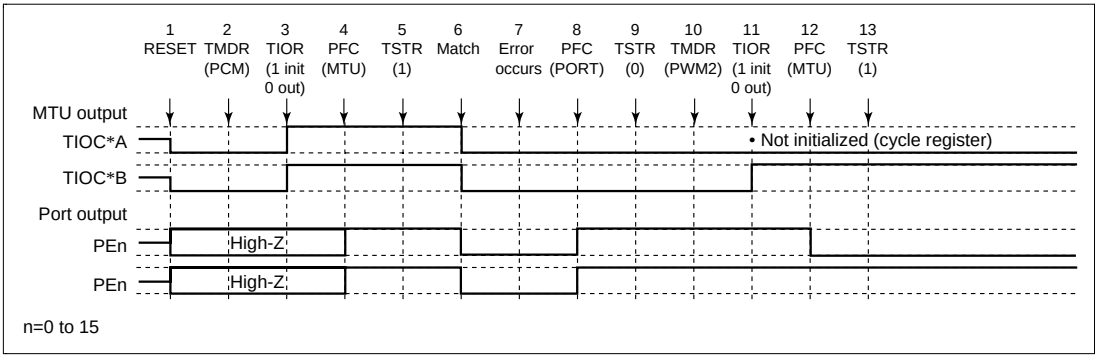


Figure 12.114 Error Occurrence in Phase Counting Mode, Recovery in PWM Mode 2

1 to 9 are the same as in figure 12.112.

10. Set PWM mode 2.
11. Initialize the pins with TIOR. (In PWM mode 2, the cycle register pins are not initialized.)
12. Set MTU output with the PFC.
13. Operation is restarted by TSTR.

(20) Operation when Error Occurs during Phase Counting Mode Operation, and Operation is Restarted in Phase Counting Mode: Figure 12.115 shows an explanatory diagram of the case where an error occurs in phase counting mode and operation is restarted in phase counting mode after re-setting.

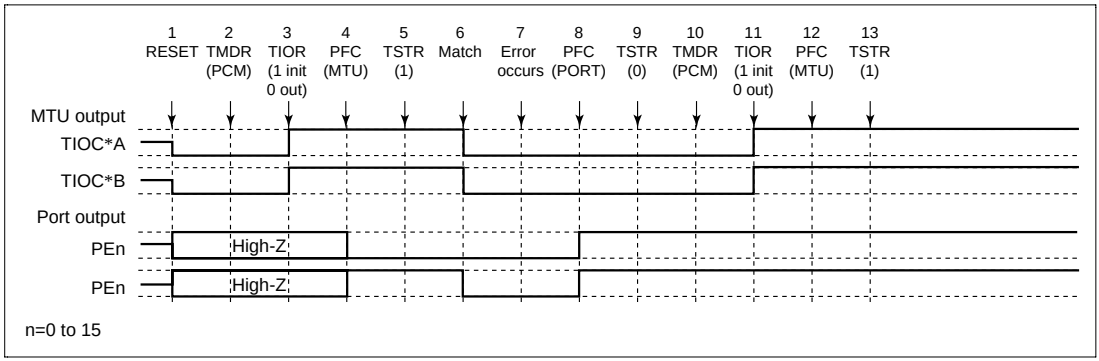


Figure 12.115 Error Occurrence in Phase Counting Mode, Recovery in Phase Counting Mode

1 to 9 are the same as in figure 12.112.

10. Not necessary when restarting in phase counting mode.
11. Initialize the pins with TIOR.
12. Set MTU output with the PFC.
13. Operation is restarted by TSTR.

(21) Operation when Error Occurs during Complementary PWM Mode Operation, and Operation is Restarted in Normal Mode: Figure 12.116 shows an explanatory diagram of the case where an error occurs in complementary PWM mode and operation is restarted in normal mode after re-setting.

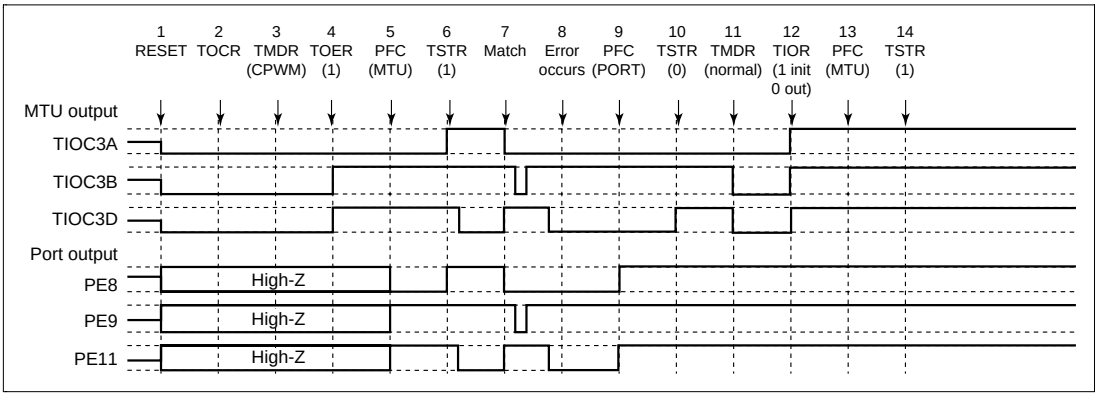


Figure 12.116 Error Occurrence in Complementary PWM Mode, Recovery in Normal Mode

1. After a reset, MTU output is low and ports are in the high-impedance state.
2. Select the complementary PWM output level and cyclic output enabling/disabling with TOCR.
3. Set complementary PWM.
4. Enable channel 3 and 4 output with TOER.
5. Set MTU output with the PFC.
6. The count operation is started by TSTR.
7. The complementary PWM waveform is output on compare-match occurrence.
8. An error occurs.
9. Set port output with the PFC and output the inverse of the active level.
10. The count operation is stopped by TSTR. (MTU output becomes the complementary PWM output initial value.)
11. Set normal mode. (MTU output goes low.)
12. Initialize the pins with TIOR.
13. Set MTU output with the PFC.
14. Operation is restarted by TSTR.

(22) Operation when Error Occurs during Complementary PWM Mode Operation, and Operation is Restarted in PWM Mode 1: Figure 12.117 shows an explanatory diagram of the case where an error occurs in complementary PWM mode and operation is restarted in PWM mode 1 after re-setting.

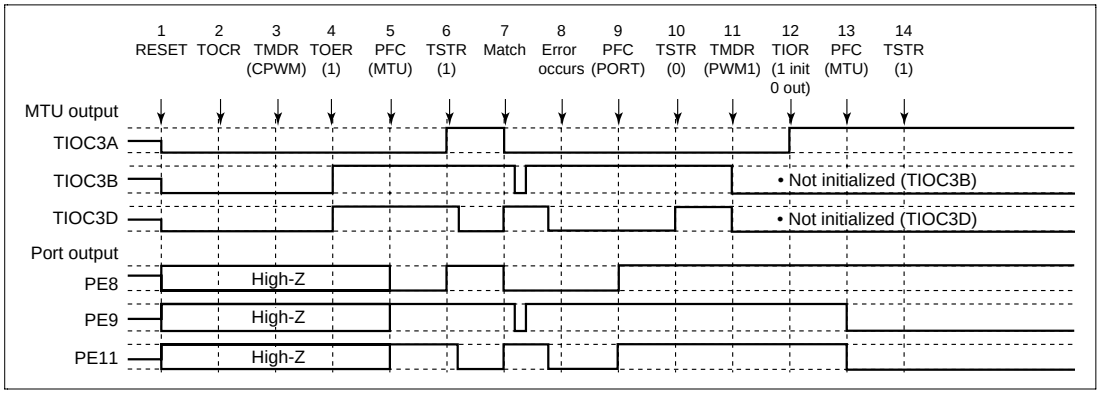


Figure 12.117 Error Occurrence in Complementary PWM Mode, Recovery in PWM Mode 1

1 to 10 are the same as in figure 12.116.

11. Set PWM mode 1. (MTU output goes low.)
12. Initialize the pins with TIOR. (In PWM mode 1, the TIOC *B side is not initialized.)
13. Set MTU output with the PFC.
14. Operation is restarted by TSTR.

(23a) Operation when Error Occurs during Complementary PWM Mode Operation, and Operation is Restarted in Complementary PWM Mode: Figure 12.118 shows an explanatory diagram of the case where an error occurs in complementary PWM mode and operation is restarted in complementary PWM mode after re-setting (when operation is restarted using the cycle and duty settings at the time the counter was stopped).

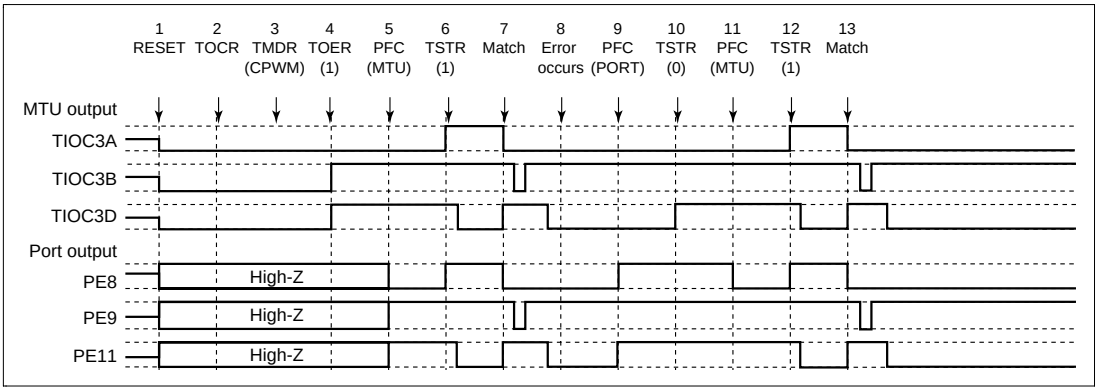


Figure 12.118 Error Occurrence in Complementary PWM Mode, Recovery in Complementary PWM Mode

1 to 10 are the same as in figure 12.116.

11. Set MTU output with the PFC.
12. Operation is restarted by TSTR.
13. The complementary PWM waveform is output on compare-match occurrence.

(23b) Operation when Error Occurs during Complementary PWM Mode Operation, and Operation is Restarted in Complementary PWM Mode: Figure 12.119 shows an explanatory diagram of the case where an error occurs in complementary PWM mode and operation is restarted in complementary PWM mode after re-setting (when operation is restarted using completely new cycle and duty settings).

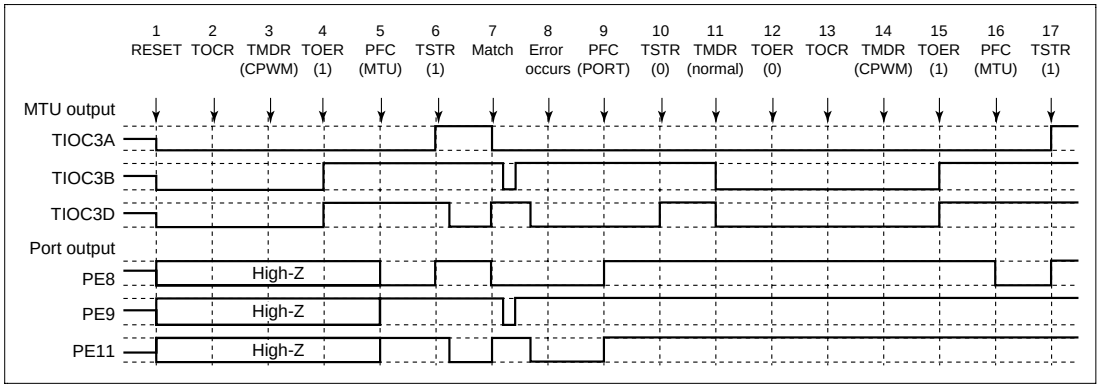


Figure 12.119 Error Occurrence in Complementary PWM Mode, Recovery in Complementary PWM Mode

1 to 10 are the same as in figure 12.116.

11. Set normal mode and make new settings. (MTU output goes low.)
12. Disable channel 3 and 4 output with TOER.
13. Select the complementary PWM mode output level and cyclic output enabling/disabling with TOCR.
14. Set complementary PWM.
15. Enable channel 3 and 4 output with TOER.
16. Set MTU output with the PFC.
17. Operation is restarted by TSTR.

(24) Operation when Error Occurs during Complementary PWM Mode Operation, and Operation is Restarted in Reset-Synchronous PWM Mode: Figure 12.120 shows an explanatory diagram of the case where an error occurs in complementary PWM mode and operation is restarted in reset-synchronous PWM mode.

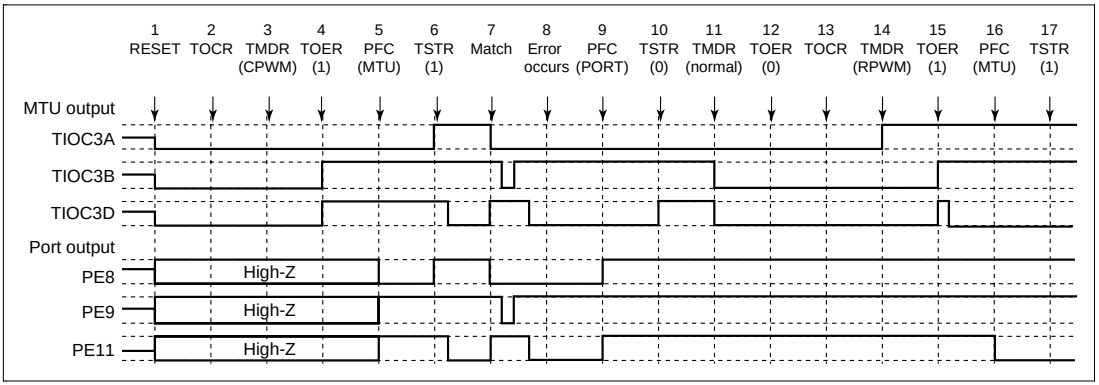


Figure 12.120 Error Occurrence in Complementary PWM Mode, Recovery in Reset-Synchronous PWM Mode

1 to 10 are the same as in figure 12.116.

11. Set normal mode. (MTU output goes low.)
12. Disable channel 3 and 4 output with TOER.
13. Select the reset-synchronous PWM mode output level and cyclic output enabling/disabling with TOCR.
14. Set reset-synchronous PWM.
15. Enable channel 3 and 4 output with TOER.
16. Set MTU output with the PFC.
17. Operation is restarted by TSTR.

(25) Operation when Error Occurs during Reset-Synchronous PWM Mode Operation, and Operation is Restarted in Normal Mode: Figure 12.121 shows an explanatory diagram of the case where an error occurs in reset-synchronous PWM mode and operation is restarted in normal mode after re-setting.

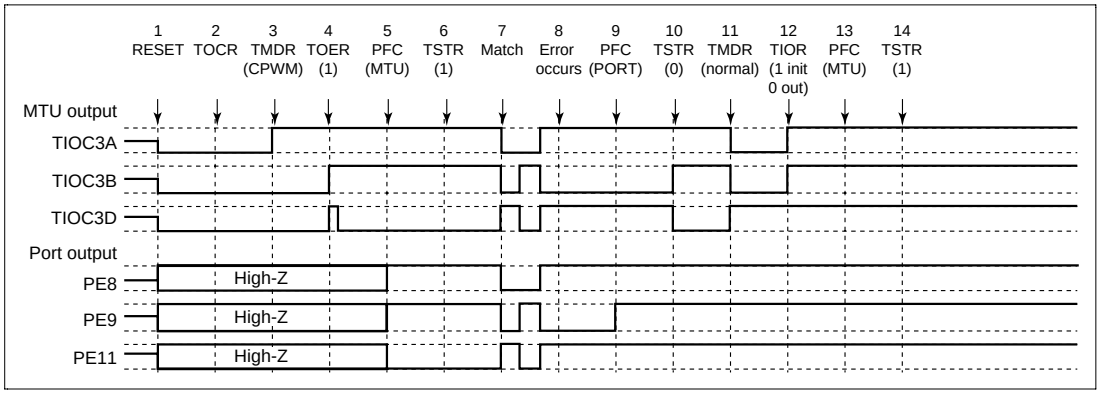


Figure 12.121 Error Occurrence in Reset-Synchronous PWM Mode, Recovery in Normal Mode

1. After a reset, MTU output is low and ports are in the high-impedance state.
2. Select the reset-synchronous PWM output level and cyclic output enabling/disabling with TOCR.
3. Set reset-synchronous PWM.
4. Enable channel 3 and 4 output with TOER.
5. Set MTU output with the PFC.
6. The count operation is started by TSTR.
7. The reset-synchronous PWM waveform is output on compare-match occurrence.
8. An error occurs.
9. Set port output with the PFC and output the inverse of the active level.
10. The count operation is stopped by TSTR. (MTU output becomes the reset-synchronous PWM output initial value.)
11. Set normal operating mode. (MTU positive phase output is low, and negative phase output is high.)
12. Initialize the pins with TIOR.
13. Set MTU output with the PFC.
14. Operation is restarted by TSTR.

(26) Operation when Error Occurs during Reset-Synchronous PWM Mode Operation, and Operation is Restarted in PWM Mode 1: Figure 12.122 shows an explanatory diagram of the case where an error occurs in reset-synchronous PWM mode and operation is restarted in PWM mode 1 after re-setting.

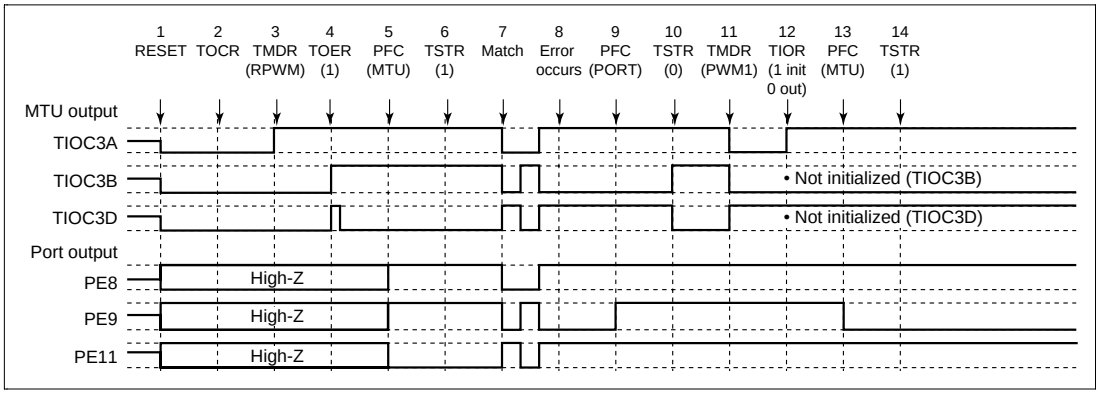


Figure 12.122 Error Occurrence in Reset-Synchronous PWM Mode, Recovery in PWM Mode 1

1 to 10 are the same as in figure 12.121.

11. Set PWM mode 1. (MTU positive phase output is low, and negative phase output is high.)
12. Initialize the pins with TIOR. (In PWM mode 1, the TIOC *B side is not initialized.)
13. Set MTU output with the PFC.
14. Operation is restarted by TSTR.

(27) Operation when Error Occurs during Reset-Synchronous PWM Mode Operation, and Operation is Restarted in Complementary PWM Mode: Figure 12.123 shows an explanatory diagram of the case where an error occurs in reset-synchronous PWM mode and operation is restarted in complementary PWM mode after re-setting.

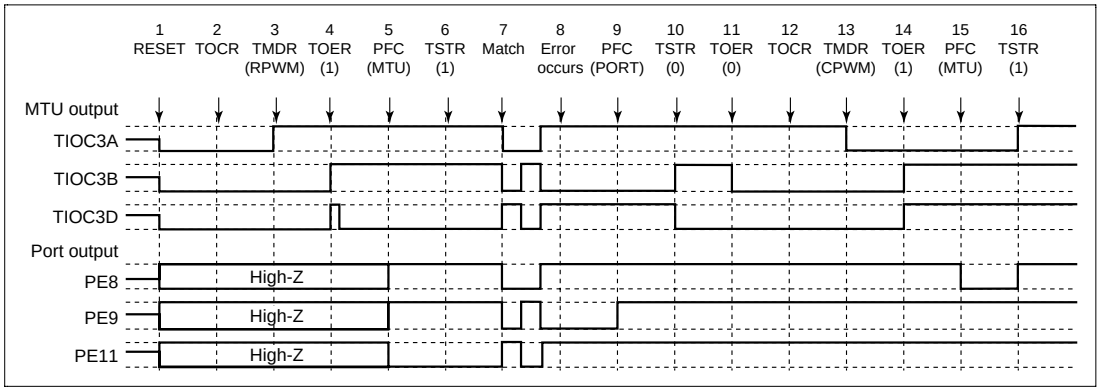


Figure 12.123 Error Occurrence in Reset-Synchronous PWM Mode, Recovery in Complementary PWM Mode

1 to 10 are the same as in figure 12.121.

11. Disable channel 3 and 4 output with TOER.
12. Select the complementary PWM output level and cyclic output enabling/disabling with TOCR.
13. Set complementary PWM. (The MTU cyclic output pin goes low.)
14. Enable channel 3 and 4 output with TOER.
15. Set MTU output with the PFC.
16. Operation is restarted by TSTR.

(28) Operation when Error Occurs during Reset-Synchronous PWM Mode Operation, and Operation is Restarted in Reset-Synchronous PWM Mode: Figure 12.124 shows an explanatory diagram of the case where an error occurs in reset-synchronous PWM mode and operation is restarted in reset-synchronous PWM mode after re-setting.

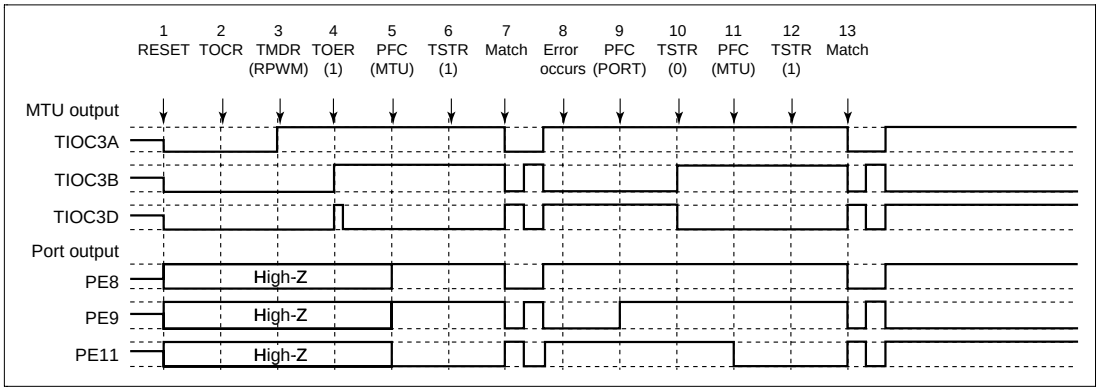


Figure 12.124 Error Occurrence in Reset-Synchronous PWM Mode, Recovery in Reset-Synchronous PWM Mode

1 to 10 are the same as in figure 12.121.

11. Set MTU output with the PFC.
12. Operation is restarted by TSTR.
13. The reset-synchronous PWM waveform is output on compare-match occurrence.

12.9 Port Output Enable (POE)

The port output enable (POE) can be used to establish a high-impedance state for high-current pins, by changing the $\overline{\text{POE0}}$–$\overline{\text{POE3}}$ pin input, depending on the output status of the high-current pins (PE09/TIOC3B, PE11/TIOC3D, PE12/TIOC4A, PE13/TIOC4B/$\overline{\text{MRES}}$, PE14/TIOC4C/DACK0/$\overline{\text{AH}}$, PE15/TIOC4D/DACK1/$\overline{\text{IRQOUT}}$). It can also simultaneously generate interrupt requests.

The high-current pins also become high-impedance regardless of whether these pin functions are selected in cases such as when the oscillator stops or in standby mode. Refer to section 4, Clock Pulse Generator (CPG), for details.

12.9.1 Features

- Each of the $\overline{\text{POE0}}$–$\overline{\text{POE3}}$ input pins can be set for falling edge, $\phi/8 \times 16$, $\phi/16 \times 16$, or $\phi/128 \times 16$ low-level sampling.
- High-current pins can be set to high-impedance state by $\overline{\text{POE0}}$–$\overline{\text{POE3}}$ pin falling-edge or low-level sampling.
- High-current pins can be set to high-impedance state when the high-current pin output levels are compared and simultaneous low-level output continues for one cycle or more (except in the 33.3 MHz version).
- Interrupts can be generated by input-level sampling or output-level comparison results.

12.9.2 Block Diagram

The POE has input-level detection circuitry and output-level detection circuitry, as shown in the block diagram of figure 12.125.

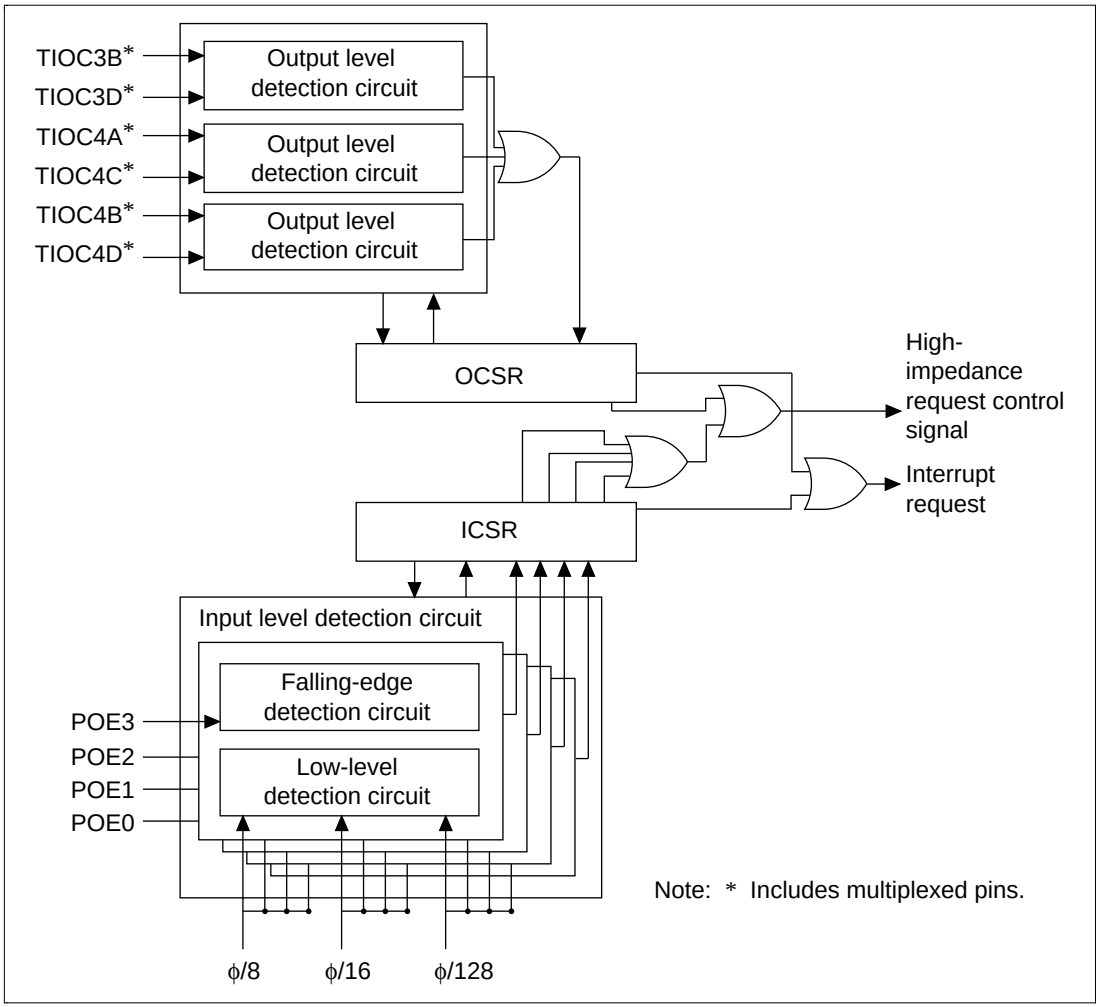


Figure 12.125 POE Block Diagram

12.9.3 Pin Configuration

Table 12.18 shows the POE pins.

Table 12.18 Pin Configuration

Name	Abbreviation	I/O	Description
Port output enable input pins	$\overline{\text{POE0}}$–$\overline{\text{POE3}}$	Input	Input request signals to make high-current pins high-impedance state

Table 12.19 shows output-level comparisons with pin combinations.

Table 12.19 Output Level Comparisons

Pin Combination	I/O	Description
PE09/TIOC3B and PE11/TIOC3D	Output	All high-current pins are made high-impedance state when the pins simultaneously output low-level for longer than 1 cycle.
PE12/TIOC4A and PE14/TIOC4C/DACK0/$\overline{\text{AH}}$	Output	All high-current pins are made high-impedance state when the pins simultaneously output low-level for longer than 1 cycle.
PE13/TIOC4B/$\overline{\text{MRES}}$ and PE15/TIOC4D/DACK1/$\overline{\text{IRQOUT}}$	Output	All high-current pins are made high-impedance state when the pins simultaneously output low-level for longer than 1 cycle.

12.9.4 Register Configuration

The POE has the two registers shown in table 12.20. The input level control/status register (ICSR) controls both $\overline{\text{POE0}}$–$\overline{\text{POE3}}$ pin input signal detection and interrupts. The output level control/status register (OCSR) controls both the enable/disable of output comparison and interrupts.

Table 12.20 Input Level Control/Status Register Configuration

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Input level control/status register	ICSR	R/(W)*1	H'0000	H'FFFF83C0 H'FFFF83C1	8, 16, 32
Output level control/status register	OCSR	R/(W)*2	H'0000	H'FFFF83C2 H'FFFF83C3	8, 16, 32

Notes: *1 Only 0 writes to bits 15–12 are possible to clear the flags.
*2 Only 0 writes to bits 15 are possible to clear the flags.

12.10 POE Register Descriptions

12.10.1 Input Level Control/Status Register (ICSR)

The input level control/status register (ICSR) is a 16-bit read/write register that selects the $\overline{\text{POE0}}$–$\overline{\text{POE3}}$ pin input modes, controls the enable/prohibit of interrupts, and indicates status. If any of the POE3F–POE0F bits are set to 1, the high current pins become high impedance state.

ICSR is initialized to H'0000 by power-on resets; however, it is not initialized for manual resets, standby mode, or sleep mode, so the previous data is maintained.

Bit:	15	14	13	12	11	10	9	8
	POE3F	POE2F	POE1F	POE0F	—	—	—	PIE
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R	R/W

Bit:	7	6	5	4	3	2	1	0
	POE3M1	POE3M0	POE2M1	POE2M0	POE1M1	POE1M0	POE0M1	POE0M0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: * Only 0 writes are possible to clear the flags.

- Bit 15—POE3 Flag (POE3F): This flag indicates that a high impedance request has been input to the $\overline{\text{POE3}}$ pin.

Bit 15: POE3F	Description
0	Clear condition: By writing 0 to POE3F after reading a POE3F = 1 (initial value)
1	Set condition: When the input set by ICSR bits 7 and 6 occurs at the $\overline{\text{POE3}}$ pin

- Bit 14—POE2 Flag (POE2F): This flag indicates that a high impedance request has been input to the $\overline{\text{POE2}}$ pin.

Bit 14: POE2F	Description
0	Clear condition: By writing 0 to POE2F after reading a POE2F = 1 (initial value)
1	Set condition: When the input set by ICSR bits 5 and 4 occurs at the $\overline{\text{POE2}}$ pin

- Bit 13—POE1 Flag (POE1F): This flag indicates that a high impedance request has been input to the $\overline{\text{POE1}}$ pin.

Bit 13: POE1F	Description
0	Clear condition: By writing 0 to POE1F after reading a POE1F = 1 (initial value)
1	Set condition: When the input set by ICSR bits 3 and 2 occurs at the $\overline{\text{POE1}}$ pin

- Bit 12—POE0 Flag (POE0F): This flag indicates that a high impedance request has been input to the $\overline{\text{POE0}}$ pin.

Bit 12: POE0F	Description
0	Clear condition: By writing 0 to POE0F after reading a POE0F = 1 (initial value)
1	Set condition: When the input set by ICSR bits 1 and 0 occurs at the $\overline{\text{POE0}}$ pin

- Bits 11–9—Reserved: These bits always read as 0. The write value should always be 0.
- Bit 8—Port Interrupt Enable (PIE): Enables or disables interrupt requests when any of the POE0F–POE3F bits of the ICSR are set to 1.

Bit 8: PIE	Description
0	Interrupt requests disabled (initial value)
1	Interrupt requests enabled

- Bits 7 and 6—POE3 Mode 1, 0 (POE3M1 and POE3M0): These bits select the input mode of the $\overline{\text{POE3}}$ pin.

Bit 7: POE3M1	Bit 6: POE3M0	Description
0	0	Accept request on falling edge of $\overline{\text{POE3}}$ input. (initial value)
	1	Accept request when $\overline{\text{POE3}}$ input has been sampled for 16 ϕ/8 clock pulses, and all are low level.
1	0	Accept request when $\overline{\text{POE3}}$ input has been sampled for 16 ϕ/16 clock pulses, and all are low level.
	1	Accept request when $\overline{\text{POE3}}$ input has been sampled for 16 ϕ/128 clock pulses, and all are low level.

- Bits 5 and 4—POE2 Mode 1, 0 (POE2M1 and POE2M0): These bits select the input mode of the $\overline{\text{POE2}}$ pin.

Bit 5: POE2M1	Bit 4: POE2M0	Description
0	0	Accept request on falling edge of $\overline{\text{POE2}}$ input. (initial value)
	1	Accept request when $\overline{\text{POE2}}$ input has been sampled for 16 ϕ/8 clock pulses, and all are low level.
1	0	Accept request when $\overline{\text{POE2}}$ input has been sampled for 16 ϕ/16 clock pulses, and all are low level.
	1	Accept request when $\overline{\text{POE2}}$ input has been sampled for 16 ϕ/128 clock pulses, and all are low level.

- Bits 3 and 2—POE1 Mode 1, 0 (POE1M1 and POE1M0): These bits select the input mode of the $\overline{\text{POE1}}$ pin.

Bit 3: POE1M1	Bit 2: POE1M0	Description
0	0	Accept request on falling edge of $\overline{\text{POE1}}$ input. (initial value)
	1	Accept request when $\overline{\text{POE1}}$ input has been sampled for 16 ϕ/8 clock pulses, and all are low level.
1	0	Accept request when $\overline{\text{POE1}}$ input has been sampled for 16 ϕ/16 clock pulses, and all are low level.
	1	Accept request when $\overline{\text{POE1}}$ input has been sampled for 16 ϕ/128 clock pulses, and all are low level.

- Bits 1 and 0—POE0 Mode 1, 0 (POE0M1 and POE0M0): These bits select the input mode of the $\overline{\text{POE0}}$ pin.

Bit 1: POE0M1	Bit 0: POE0M0	Description
0	0	Accept request on falling edge of $\overline{\text{POE0}}$ input. (initial value)
	1	Accept request when $\overline{\text{POE0}}$ input has been sampled for 16 ϕ/8 clock pulses, and all are low level.
1	0	Accept request when $\overline{\text{POE0}}$ input has been sampled for 16 ϕ/16 clock pulses, and all are low level.
	1	Accept request when $\overline{\text{POE0}}$ input has been sampled for 16 ϕ/128 clock pulses, and all are low level.

12.10.2 Output Level Control/Status Register (OCSR)

The output level control/status register (OCSR) is a 16-bit read/write register that controls the enable/disable of both output level comparison and interrupts, and indicates status. If the OSF bit is set to 1, the high current pins become high impedance.

OCSR is initialized to H'0000 by an external power-on reset; however, it is not initialized for manual resets, reset by WDT standby mode, or sleep mode, so the previous data is maintained.

Bit:	15	14	13	12	11	10	9	8
	OSF	—	—	—	—	—	OCE	OIE
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R	R	R	R	R	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Note: * Only 0 writes are possible to clear the flag.

- Bit 15—Output Short Flag (OSF): This flag indicates that among the three pairs of 2 phase outputs compared, the outputs of at least one pair have simultaneously become Low level output.

Bit 15: OSF	Description
0	Clear condition: By writing 0 to OSF after reading an OSF = 1 (initial value)
1	Set condition: When any one pair of the 2-phase outputs simultaneously become Low level

- Bits 14–10—Reserved: These bits always read as 0. The write value should always be 0.

- Bit 9—Output Level Compare Enable (OCE): This bit enables the start of output level comparisons. When setting this bit, pay special attention to the output pin combinations shown in table 12.19. When 0 is output, the OSF bit is set to 1 at the same time this bit is set, and output goes to high impedance. Accordingly, bits 15–11 and bit 9 of the port E data register (PEDR) are set to 1. For the MTU output comparison, set the bit to 1 after setting the MTU's output pins with the PFC. Set this bit only when using pins as outputs.

  When the OCE bit is set to 1, if OIE = 0 a high-impedance request will not be issued even if OSF is set to 1. Therefore, in order to have a high-impedance request issued according to the result of the output comparison, the OIE bit must be set to 1. When OCE = 1 and OIE = 1, an interrupt request will be generated at the same time as the high-impedance request; however, this interrupt can be masked by means of an interrupt controller (INTC) setting.

Bit 9: OCE	Description
0	Output level compare disabled (initial value)
1	Output level compare enabled; makes an output high impedance request when OSF = 1.

- Bit 8—Output Short Interrupt Enable (OIE): Makes interrupt requests when the OSF bit of the OCSR is set.

Bit 8: OIE	Description
0	Interrupt requests disabled (initial value)
1	Interrupt requests enabled

- Bits 7–0—Reserved: Always read as 0, and cannot be modified.

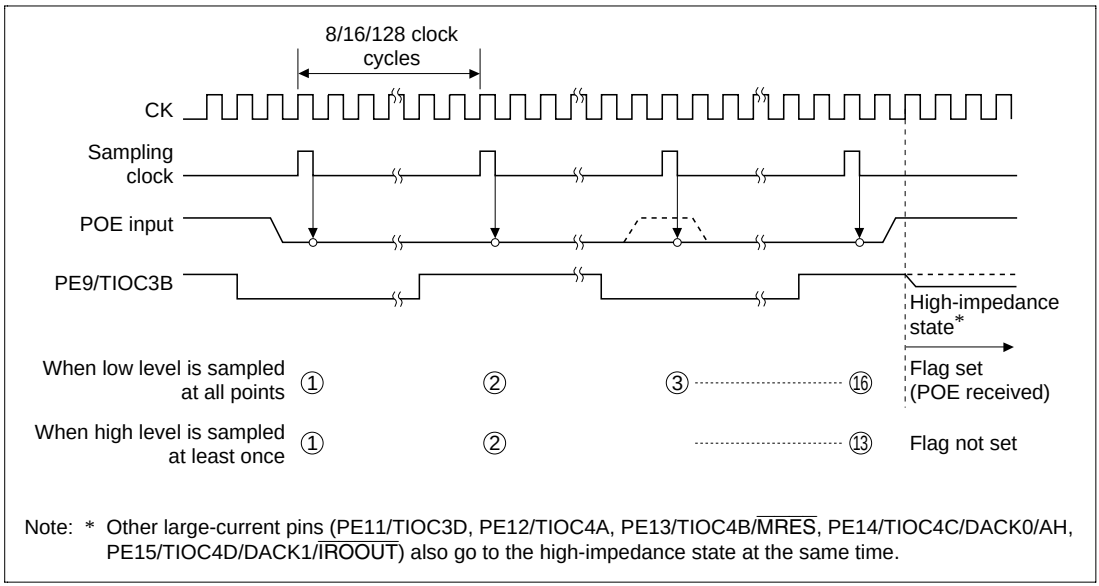
12.11 Operation

12.11.1 Input Level Detection Operation

If the input conditions set by the ICSR occur on any of the $\overline{\text{POE}}$ pins, all high-current pins become high-impedance state.

Falling Edge Detection: When a change from high to low level is input to the $\overline{\text{POE}}$ pins.

Low-Level Detection: Figure 12.126 shows the low-level detection operation. Sixteen continuous low levels are sampled with the sampling clock established by the ICSR. If even one high level is detected during this interval, the low level is not accepted.



Note: * Other large-current pins (PE11/TIOC3D, PE12/TIOC4A, PE13/TIOC4B/$\overline{\text{MRES}}$, PE14/TIOC4C/DACK0/AH, PE15/TIOC4D/DACK1/$\overline{\text{IROOUT}}$) also go to the high-impedance state at the same time.

Figure 12.126 Low-Level Detection Operation

12.11.2 Output-Level Compare Operation

Figure 12.127 shows an example of the output-level compare operation for the combination of PE09/TIOC3B and PE11/TIOC3D. The operation is the same for the other pin combinations.

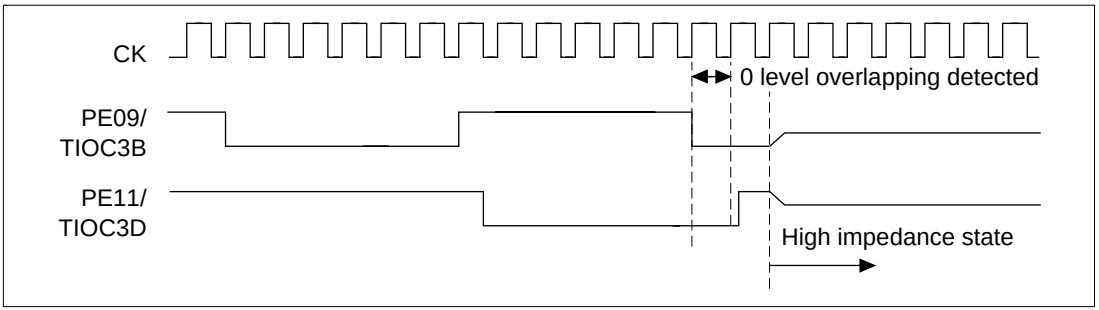


Figure 12.127 Output-Level Detection Operation

12.11.3 Release from High-Impedance State

High-current pins that have entered high-impedance state due to input-level detection can be released either by returning them to their initial state with a power-on reset, or by clearing all of the bit 12–15 (POE0F–POE3F) flags of the ICSR. High-current pins that have become high-impedance due to output-level detection can be released either by returning them to their initial state with a power-on reset, or by first clearing bit 9 (OCE) of the OCSR to disable output-level compares, then clearing the bit 15 (OSF) flag. However, when returning from high-impedance state by clearing the OSF flag, always do so only after outputting a high level from the high-current pins (TIOC3B, TIOC3D, TIOC4A, TIOC4B, TIOC4C, and TIOC4D). High-level outputs can be achieved by setting the MTU internal registers. See section 12.2, MTU Register Descriptions, for details.

12.11.4 POE timing

Figure 12.128 shows an example of timing from $\overline{\text{POE}}$ input to high impedance of pin.

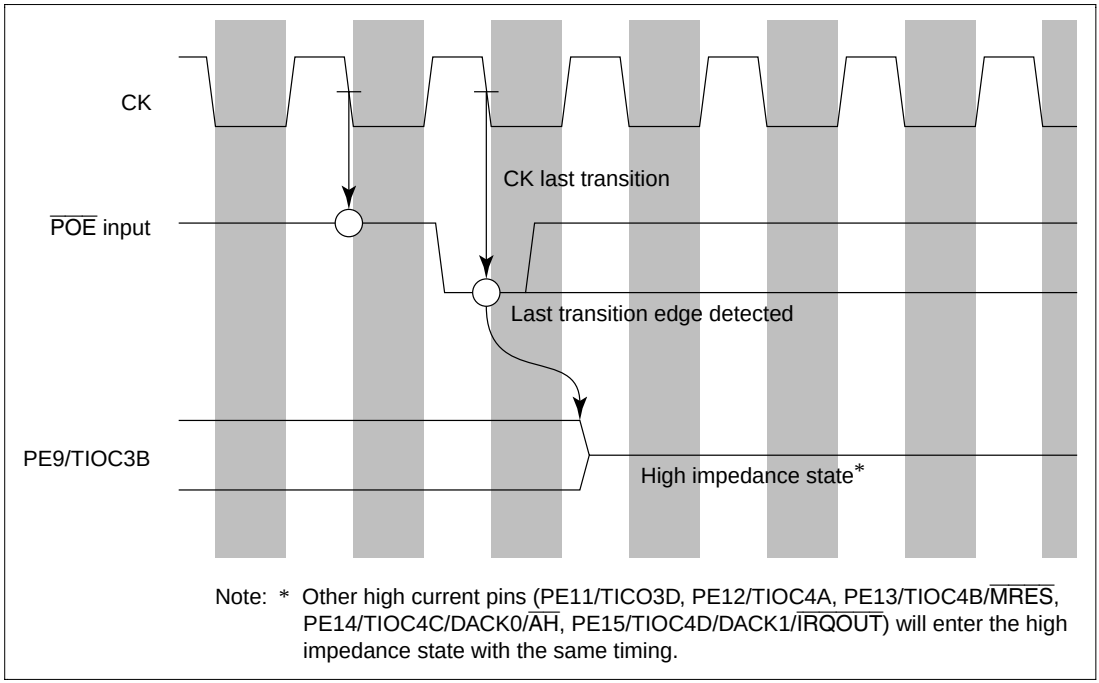


Figure 12.128 Last Transition Edge Detection Operation

12.11.5 Usage Notes

To perform POE level detection, first set POE input to high level.

Section 13 Watchdog Timer (WDT)

13.1 Overview

The watchdog timer (WDT) is a 1-channel timer for monitoring system operations. If a system encounters a problem (crashes, for example) and the timer counter overflows without being rewritten correctly by the CPU, an overflow signal ($\overline{\text{WDTOVF}}$) is output externally. The WDT can simultaneously generate an internal reset signal for the entire chip.

When the watchdog function is not needed, the WDT can be used as an interval timer. In the interval timer operation, an interval timer interrupt is generated at each counter overflow. The WDT is also used in recovering from the standby mode.

13.1.1 Features

- Works in watchdog timer mode or interval timer mode.
- Outputs $\overline{\text{WDTOVF}}$ in the watchdog timer mode. When the counter overflows in the watchdog timer mode, overflow signal $\overline{\text{WDTOVF}}$ is output externally. You can select whether to reset the chip internally when this happens. Either the power-on reset or manual reset signal can be selected as the internal reset signal.
- Generates interrupts in the interval timer mode. When the counter overflows, it generates an interval timer interrupt.
- Clears standby mode.
- Works with eight counter input clocks.

13.1.2 Block Diagram

Figure 13.1 is the block diagram of the WDT.

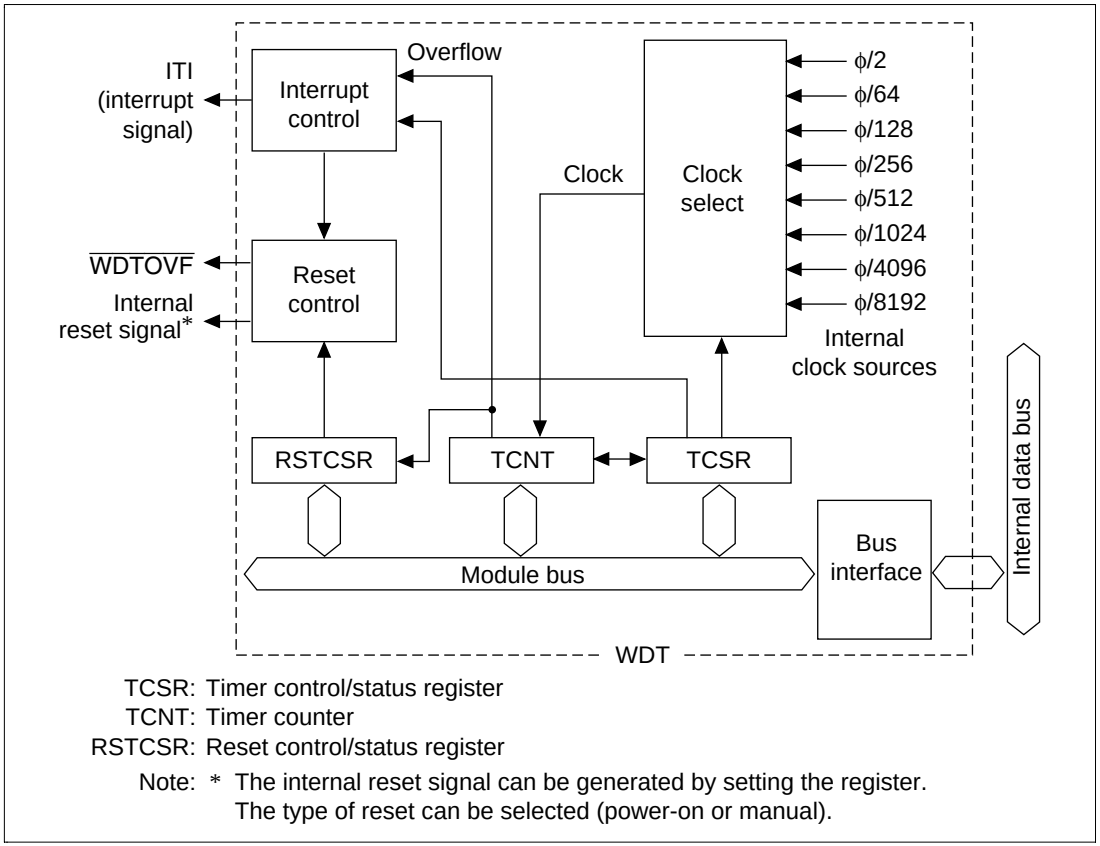


Figure 13.1 WDT Block Diagram

13.1.3 Pin Configuration

Table 13.1 shows the pin configuration.

Table 13.1 Pin Configuration

Pin	Abbreviation	I/O	Function
Watchdog timer overflow	$\overline{\text{WDTOVF}}$	O	Outputs the counter overflow signal in the watchdog timer mode

13.1.4 Register Configuration

Table 13.2 summarizes the three WDT registers. They are used to select the clock, switch the WDT mode, and control the reset signal.

Table 13.2 WDT Registers

				Address	
Name	**Abbreviation**	**R/W**	**Initial Value**	**Write*1**	**Read*2**
Timer control/status register	TCSR	R/(W)*3	H'18	H'FFFF8610	H'FFFF8610
Timer counter	TCNT	R/W	H'00		H'FFFF8611
Reset control/status register	RSTCSR	R/(W)*3	H'1F	H'FFFF8612	H'FFFF8613

Notes: *1 Write by word transfer. It cannot be written in byte or longword.
*2 Read by byte transfer. It cannot be read in word or longword.
*3 Only 0 can be written in bit 7 to clear the flag.

13.2 Register Descriptions

13.2.1 Timer Counter (TCNT)

The TCNT is an 8-bit read/write upcounter. (The TCNT differs from other registers in that it is more difficult to write to. See section 13.2.4, Register Access, for details.) When the timer enable bit (TME) in the timer control/status register (TCSR) is set to 1, the watchdog timer counter starts counting pulses of an internal clock selected by clock select bits 2–0 (CKS2–CKS0) in the TCSR. When the value of the TCNT overflows (changes from H'FF to H'00), a watchdog timer overflow signal ($\overline{\text{WDTOVF}}$) or interval timer interrupt (ITI) is generated, depending on the mode selected in the WT/$\overline{\text{IT}}$ bit of the TCSR.

The TCNT is initialized to H'00 by a power-on reset and when the TME bit is cleared to 0. It is not initialized in the standby mode. The TCNT is not initialized by a manual reset from an external source ($\overline{\text{MRES}}$), but is initialized by a manual reset from the WDT.

Bit:	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

13.2.2 Timer Control/Status Register (TCSR)

The timer control/status register (TCSR) is an 8-bit read/write register. (The TCSR differs from other registers in that it is more difficult to write to. See section 13.2.4, Register Access, for details.) Its functions include selecting the timer mode and clock source.

Bits 7–5 are initialized to 000 by a power-on reset or in standby mode. Bits 2–0 are initialized to 000 by a power-on reset, but retain their values in the standby mode. These bits are not initialized by a manual reset from an external source ($\overline{\text{MRES}}$), but are initialized by a manual reset from the WDT.

Bit:	7	6	5	4	3	2	1	0
	OVF	WT/$\overline{\text{IT}}$	TME	—	—	CKS2	CKS1	CKS0
Initial value:	0	0	0	1	1	0	0	0
R/W:	R/(W)	R/W	R/W	R	R	R/W	R/W	R/W

- Bit 7—Overflow Flag (OVF): Indicates that the TCNT has overflowed from H'FF to H'00 in the interval timer mode. It is not set in the watchdog timer mode.

Bit 7: OVF	Description
0	No overflow of TCNT in interval timer mode (initial value) Cleared by reading OVF, then writing 0 in OVF
1	TCNT overflow in the interval timer mode

- Bit 6—Timer Mode Select (WT/$\overline{\text{IT}}$): Selects whether to use the WDT as a watchdog timer or interval timer. When the TCNT overflows, the WDT either generates an interval timer interrupt (ITI) or generates a $\overline{\text{WDTOVF}}$ signal, depending on the mode selected.

Bit 6: WT/$\overline{\text{IT}}$	Description
0	Interval timer mode: interval timer interrupt request to the CPU when TCNT overflows (initial value)
1	Watchdog timer mode: $\overline{\text{WDTOVF}}$ signal output externally when TCNT overflows. (Section 13.2.3, Reset Control/Status Register (RSTCSR), describes in detail what happens when TCNT overflows in the watchdog timer mode.)

- Bit 5—Timer Enable (TME): Enables or disables the timer.

Bit 5: TME	Description
0	Timer disabled: TCNT is initialized to H'00 and count-up stops (initial value)
1	Timer enabled: TCNT starts counting. A $\overline{\text{WDTOVF}}$ signal or interrupt is generated when TCNT overflows.

- Bits 4 and 3—Reserved: These bits always read as 1. The write value should always be 1.
- Bits 2–0: Clock Select 2–0 (CKS2–CKS0): These bits select one of eight internal clock sources for input to the TCNT. The clock signals are obtained by dividing the frequency of the system clock (ϕ).

			Description	
Bit 2: CKS2	Bit 1: CKS1	Bit 0: CKS0	Clock Source	Overflow Interval* (ϕ = 28.7 MHz)
0	0	0	ϕ/2 (initial value)	17.9 µs
0	0	1	ϕ/64	573.4 µs
0	1	0	ϕ/128	1.1 ms
0	1	1	ϕ/256	2.3 ms
1	0	0	ϕ/512	4.6 ms
1	0	1	ϕ/1024	9.2 ms
1	1	0	ϕ/4096	36.7 ms
1	1	1	ϕ/8192	73.4 ms

Note: * The overflow interval listed is the time from when the TCNT begins counting at H'00 until an overflow occurs.

13.2.3 Reset Control/Status Register (RSTCSR)

The RSTCSR is an 8-bit readable and writable register. (The RSTCSR differs from other registers in that it is more difficult to write. See section 13.2.4, Register Access, for details.) It controls output of the internal reset signal generated by timer counter (TCNT) overflow and selects the internal reset signal type. RSTCR is initialized to H'1F by input of a reset signal from the $\overline{\text{RES}}$ pin, but is not initialized by the internal reset signal generated by the overflow of the WDT. It is initialized to H'1F in standby mode.

Bit:	7	6	5	4	3	2	1	0
	WOVF	RSTE	RSTS	—	—	—	—	—
Initial value:	0	0	0	1	1	1	1	1
R/W:	R/(W)*	R/W	R/W	R	R	R	R	R

Note: * Only 0 can be written in bit 7 to clear the flag.

- Bit 7—Watchdog Timer Overflow Flag (WOVF): Indicates that the TCNT has overflowed (H'FF–H'00) in the watchdog timer mode. It is not set in the interval timer mode.

Bit 7: WOVF	Description
0	No TCNT overflow in watchdog timer mode (initial value) Cleared when software reads WOVF, then writes 0 in WOVF
1	Set by TCNT overflow in watchdog timer mode

- Bit 6—Reset Enable (RSTE): Selects whether to reset the chip internally if the TCNT overflows in the watchdog timer mode.

Bit 6: RSTE	Description
0	Not reset when TCNT overflows (initial value). LSI not reset internally, but TCNT and TCSR reset within WDT.
1	Reset when TCNT overflows

- Bit 5—Reset Select (RSTS): Selects the type of internal reset generated if the TCNT overflows in the watchdog timer mode.

Bit 5: RSTS	Description
0	Power-on reset (initial value)
1	Manual reset

- Bits 4–0—Reserved: These bits always read as 1. The write value should always be 1.

13.2.4 Register Access

The watchdog timer's TCNT, TCSR, and RSTCSR registers differ from other registers in that they are more difficult to write to. The procedures for writing and reading these registers are given below.

Writing to the TCNT and TCSR: These registers must be written by a word transfer instruction. They cannot be written by byte transfer instructions.

The TCNT and TCSR both have the same write address. The write data must be contained in the lower byte of the written word. The upper byte must be H'5A (for the TCNT) or H'A5 (for the TCSR) (figure 13.2). This transfers the write data from the lower byte to the TCNT or TCSR.

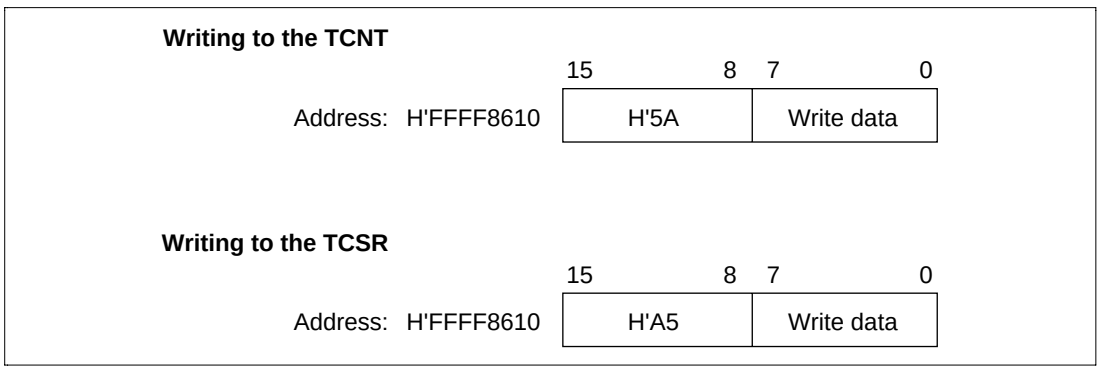


Figure 13.2 Writing to the TCNT and TCSR

Writing to the RSTCSR: The RSTCSR must be written by a word access to address H'FFFF8612. It cannot be written by byte transfer instructions.

Procedures for writing 0 in WOVF (bit 7) and for writing to RSTE (bit 6) and RSTS (bit 5) are different, as shown in figure 13.3.

To write 0 in the WOVF bit, the write data must be H'A5 in the upper byte and H'00 in the lower byte. This clears the WOVF bit to 0. The RSTE and RSTS bits are not affected. To write to the RSTE and RSTS bits, the upper byte must be H'5A and the lower byte must be the write data. The values of bits 6 and 5 of the lower byte are transferred to the RSTE and RSTS bits, respectively. The WOVF bit is not affected.

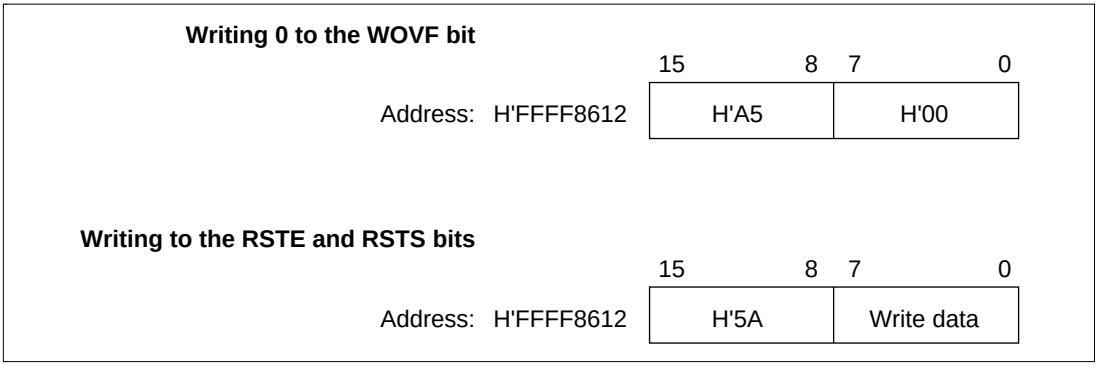


Figure 13.3 Writing to the RSTCSR

Reading from the TCNT, TCSR, and RSTCSR: TCNT, TCSR, and RSTCSR are read like other registers. Use byte transfer instructions. The read addresses are H'FFFF8610 for the TCSR, H'FFFF8611 for the TCNT, and H'FFFF8613 for the RSTCSR.

13.3 Operation

13.3.1 Watchdog Timer Mode

To use the WDT as a watchdog timer, set the WT/$\overline{IT}$ and TME bits of the TCSR to 1. Software must prevent TCNT overflow by rewriting the TCNT value (normally by writing H'00) before overflow occurs. No TCNT overflows will occur while the system is operating normally, but if the TCNT fails to be rewritten and overflows occur due to a system crash or the like, a $\overline{WDTOVF}$ signal is output externally (figure 13.4). The $\overline{WDTOVF}$ signal can be used to reset the system. The $\overline{WDTOVF}$ signal is output for 128 ϕ clock cycles.

If the RSTE bit in the RSTCSR is set to 1, a signal to reset the chip will be generated internally simultaneous to the $\overline{WDTOVF}$ signal when TCNT overflows. Either a power-on reset or a manual reset can be selected by the RSTS bit. The internal reset signal is output for 512 ϕ clock cycles.

When a watchdog overflow reset is generated simultaneously with a reset input at the $\overline{RES}$ pin, the $\overline{RES}$ reset takes priority, and the WOVF bit is cleared to 0.

The following are not initialized a WDT reset signal:

- The MTU’s POE (Port Output Enable) function register
- PFC (Pin Function Controller) function register
- I/O port register

Initializing is only possible by external power-on reset.

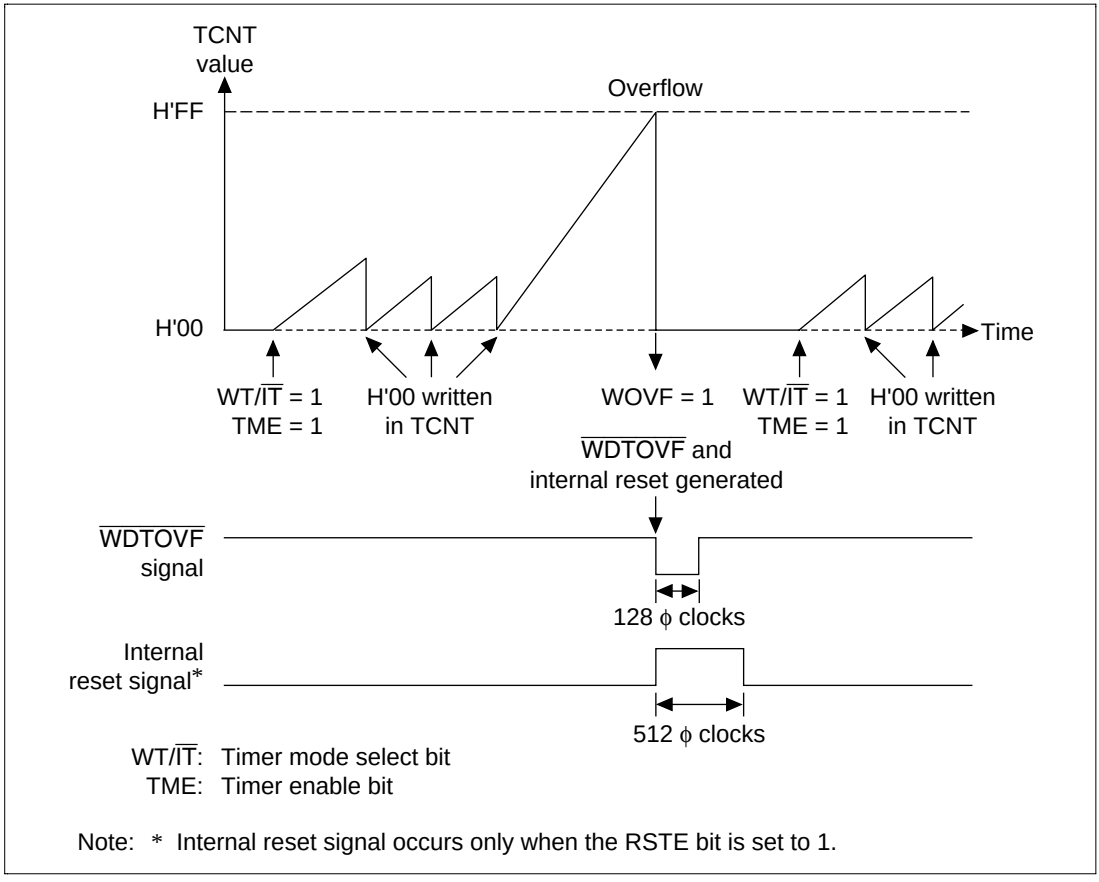


Figure 13.4 Operation in the Watchdog Timer Mode

13.3.2 Interval Timer Mode

To use the WDT as an interval timer, clear WT/$\overline{IT}$ to 0 and set TME to 1. An interval timer interrupt (ITI) is generated each time the timer counter overflows. This function can be used to generate interval timer interrupts at regular intervals (figure 13.5).

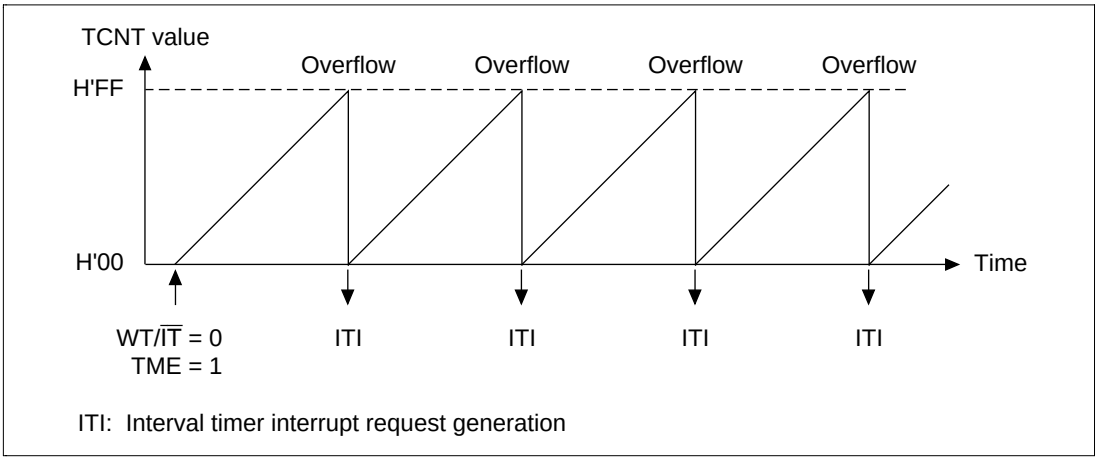


Figure 13.5 Operation in the Interval Timer Mode

13.3.3 Clearing the Standby Mode

The watchdog timer has a special function to clear the standby mode with an NMI interrupt. When using the standby mode, set the WDT as described below.

Before Transition to the Standby Mode: The TME bit in the TCSR must be cleared to 0 to stop the watchdog timer counter before it enters the standby mode. The chip cannot enter the standby mode while the TME bit is set to 1. Set bits CKS2–CKS0 so that the counter overflow interval is equal to or longer than the oscillation settling time. See sections 25.3, and 26.3, AC Characteristics, for the oscillation settling time.

Recovery from the Standby Mode: When an NMI request signal is received in standby mode, the clock oscillator starts running and the watchdog timer starts incrementing at the rate selected by bits CKS2–CKS0 before the standby mode was entered. When the TCNT overflows (changes from H'FF to H'00), the clock is presumed to be stable and usable; clock signals are supplied to the entire chip and the standby mode ends.

For details on the standby mode, see section 24, Power-Down State.

13.3.4 Timing of Setting the Overflow Flag (OVF)

In the interval timer mode, when the TCNT overflows, the OVF flag of the TCSR is set to 1 and an interval timer interrupt is simultaneously requested (figure 13.6).

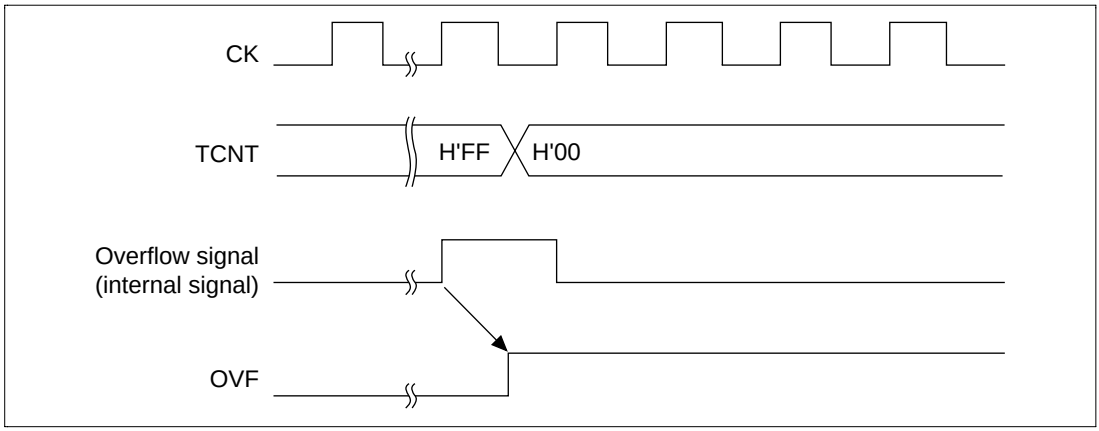


Figure 13.6 Timing of Setting the OVF

13.3.5 Timing of Setting the Watchdog Timer Overflow Flag (WOVF)

When the TCNT overflows in the watchdog timer mode, the WOVF bit of the RSTCSR is set to 1 and a $\overline{\text{WDTOVF}}$ signal is output. When the RSTE bit is set to 1, TCNT overflow enables an internal reset signal to be generated for the entire chip (figure 13.7).

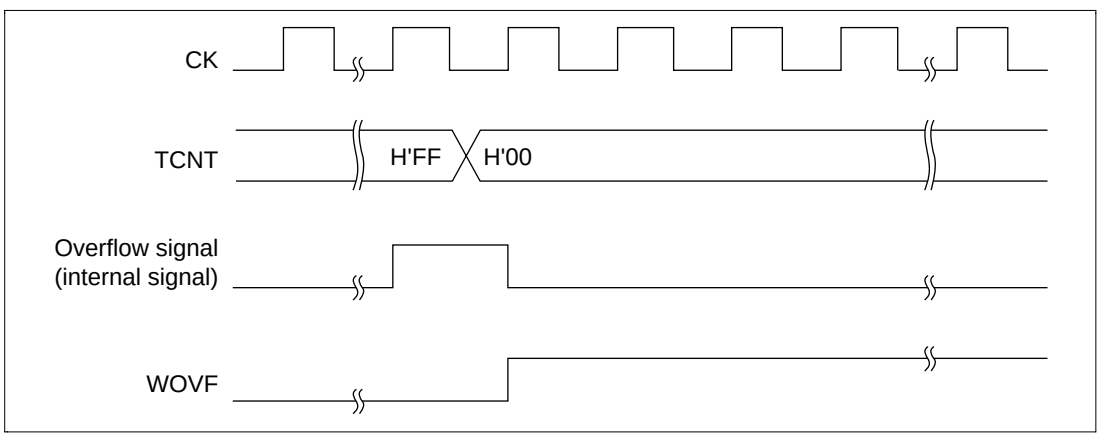


Figure 13.7 Timing of Setting the WOVF Bit

13.4 Notes on Use

13.4.1 TCNT Write and Increment Contention

If a timer counter (TCNT) increment clock pulse is generated during the T_3 state of a write cycle to the TCNT, the write takes priority and the timer counter is not incremented (figure 13.8).

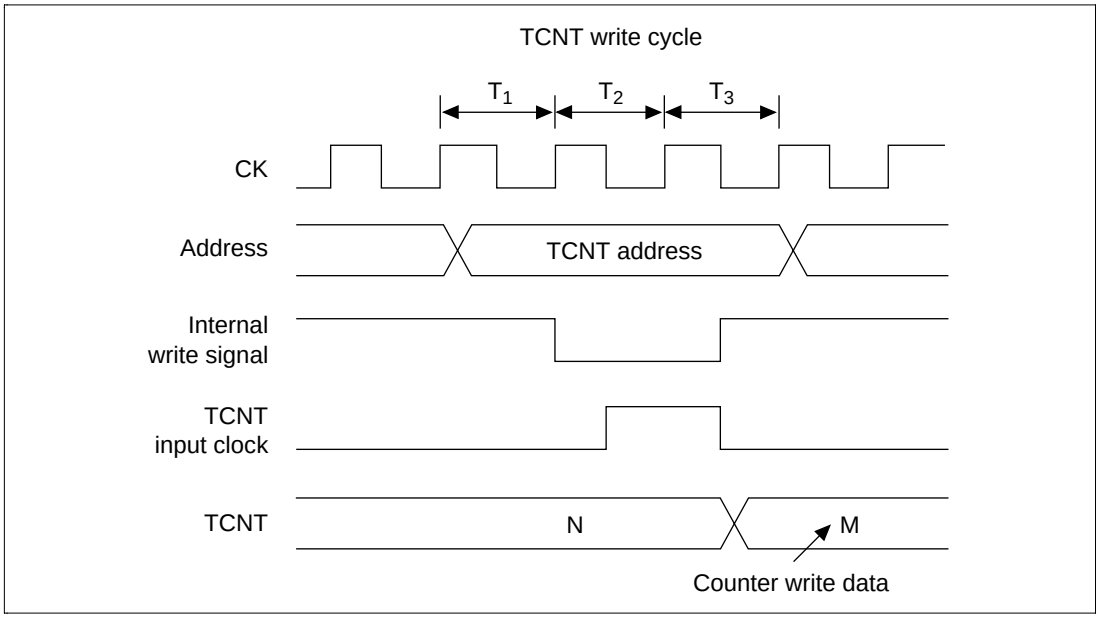


Figure 13.8 Contention between TCNT Write and Increment

13.4.2 Changing CKS2–CKS0 Bit Values

If the values of bits CKS2–CKS0 are altered while the WDT is running, the count may increment incorrectly. Always stop the watchdog timer (by clearing the TME bit to 0) before changing the values of bits CKS2–CKS0.

13.4.3 Changing between Watchdog Timer/Interval Timer Modes

To prevent incorrect operation, always stop the watchdog timer (by clearing the TME bit to 0) before switching between interval timer mode and watchdog timer mode.

13.4.4 System Reset With $\overline{\text{WDTOVF}}$

If a $\overline{\text{WDTOVF}}$ signal is input to the $\overline{\text{RES}}$ pin, the LSI cannot initialize correctly.

Avoid logical input of the $\overline{\text{WDTOVF}}$ output signal to the $\overline{\text{RES}}$ input pin. To reset the entire system with the $\overline{\text{WDTOVF}}$ signal, use the circuit shown in figure 13.9.

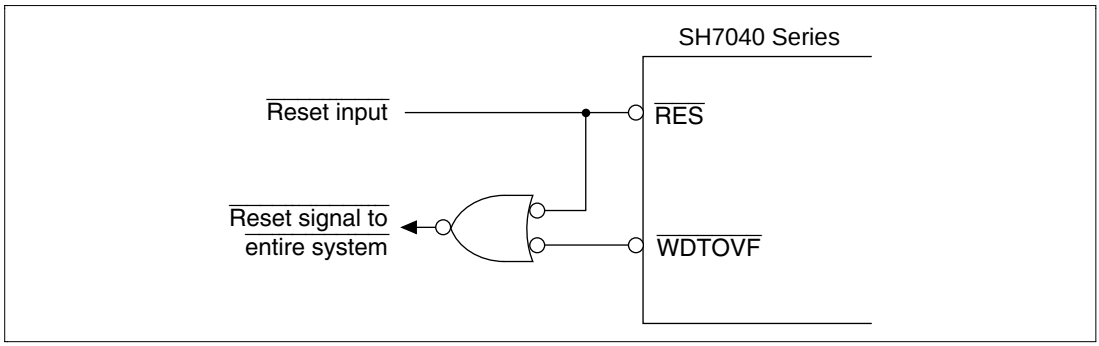


Figure 13.9 Example of a System Reset Circuit with a $\overline{\text{WDTOVF}}$ Signal

13.4.5 Internal Reset with the Watchdog Timer

If the RSTE bit is cleared to 0 in the watchdog timer mode, the LSI will not reset internally when a TCNT overflow occurs, but the TCNT and TCSR in the WDT will reset.

Section 14 Serial Communication Interface (SCI)

14.1 Overview

The SH7040 Series has a serial communication interface (SCI) with two independent channels, both of which possess the same functions.

The SCI supports both asynchronous and clock synchronous serial communication. It also has a multiprocessor communication function for serial communication among two or more processors.

14.1.1 Features

- Select asynchronous or clock synchronous as the serial communications mode.
  - — Asynchronous mode: Serial data communications are synched by start-stop in character units. The SCI can communicate with a universal asynchronous receiver/transmitter (UART), an asynchronous communication interface adapter (ACIA), or any other chip that employs a standard asynchronous serial communication. It can also communicate with two or more other processors using the multiprocessor communication function. There are twelve selectable serial data communication formats.
    - Data length: seven or eight bits
    - Stop bit length: one or two bits
    - Parity: even, odd, or none
    - Multiprocessor bit: one or none
    - Receive error detection: parity, overrun, and framing errors
    - Break detection: by reading the RxD level directly when a framing error occurs
  - — Clocked synchronous mode: Serial data communication is synchronized with a clock signal. The SCI can communicate with other chips having a clock synchronous communication function. There is one serial data communication format.
    - Data length: eight bits
    - Receive error detection: overrun errors
- Full duplex communication: The transmitting and receiving sections are independent, so the SCI can transmit and receive simultaneously. Both sections use double buffering, so continuous data transfer is possible in both the transmit and receive directions.
- On-chip baud rate generator with selectable bit rates.
- Internal or external transmit/receive clock source: baud rate generator (internal) or SCK pin (external).
- Four types of interrupts: Transmit-data-empty, transmit-end, receive-data-full, and receive-error interrupts are requested independently. The transmit-data-empty and receive-data-full interrupts can start the direct memory access controller (DMAC)/data transfer controller (DTC) to transfer data.

14.1.2 Block Diagram

Figure 14.1 shows a block diagram of the SCI.

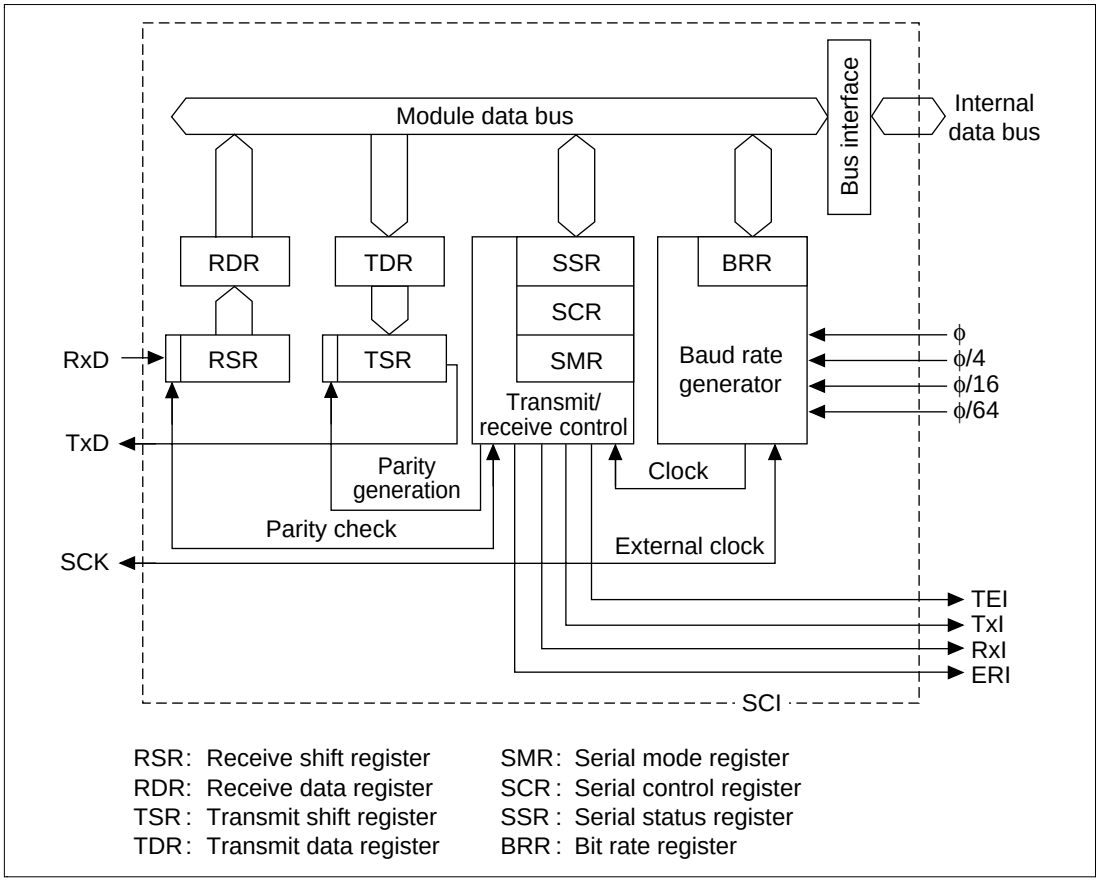


Figure 14.1 SCI Block Diagram

14.1.3 Pin Configuration

Table 14.1 summarizes the SCI pins by channel.

Table 14.1 SCI Pins

Channel	Pin Name	Abbreviation	Input/Output	Function
0	Serial clock pin	SCK0	Input/output	SCI0 clock input/output
	Receive data pin	RxD0	Input	SCI0 receive data input
	Transmit data pin	TxD0	Output	SCI0 transmit data output
1	Serial clock pin	SCK1	Input/output	SCI1 clock input/output
	Receive data pin	RxD1	Input	SCI1 receive data input
	Transmit data pin	TxD1	Output	SCI1 transmit data output

14.1.4 Register Configuration

Table 14.2 summarizes the SCI internal registers. These registers select the communication mode (asynchronous or clock synchronous), specify the data format and bit rate, and control the transmitter and receiver sections.

Table 14.2 Registers

Channel	Name	Abbreviation	R/W	Initial Value	Address*2	Access Size
0	Serial mode register	SMR0	R/W	H'00	H'FFFF81A0	8, 16
	Bit rate register	BRR0	R/W	H'FF	H'FFFF81A1	8, 16
	Serial control register	SCR0	R/W	H'00	H'FFFF81A2	8, 16
	Transmit data register	TDR0	R/W	H'FF	H'FFFF81A3	8, 16
	Serial status register	SSR0	R/(W)*1	H'84	H'FFFF81A4	8, 16
	Receive data register	RDR0	R	H'00	H'FFFF81A5	8, 16
1	Serial mode register	SMR1	R/W	H'00	H'FFFF81B0	8, 16
	Bit rate register	BRR1	R/W	H'FF	H'FFFF81B1	8, 16
	Serial control register	SCR1	R/W	H'00	H'FFFF81B2	8, 16
	Transmit data register	TDR1	R/W	H'FF	H'FFFF81B3	8, 16
	Serial status register	SSR1	R/(W)*1	H'84	H'FFFF81B4	8, 16
	Receive data register	RDR1	R	H'00	H'FFFF81B5	8, 16

Notes: *1 The only value that can be written is a 0 to clear the flags.
*2 Do not access empty addresses.

14.2 Register Descriptions

14.2.1 Receive Shift Register (RSR)

The receive shift register (RSR) receives serial data. Data input at the RxD pin is loaded into the RSR in the order received, LSB (bit 0) first, converting the data to parallel form. When one byte has been received, it is automatically transferred to the RDR.

The CPU cannot read or write the RSR directly.

Bit:	7	6	5	4	3	2	1	0
R/W:	—	—	—	—	—	—	—	—

14.2.2 Receive Data Register (RDR)

The receive data register (RDR) stores serial receive data. The SCI completes the reception of one byte of serial data by moving the received data from the receive shift register (RSR) into the RDR for storage. The RSR is then ready to receive the next data. This double buffering allows the SCI to receive data continuously.

The CPU can read but not write the RDR. The RDR is initialized to H'00 by a power-on reset or in standby mode. Manual reset does not initialize RDR.

Bit:	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

14.2.3 Transmit Shift Register (TSR)

The transmit shift register (TSR) transmits serial data. The SCI loads transmit data from the transmit data register (TDR) into the TSR, then transmits the data serially from the TxD pin, LSB (bit 0) first. After transmitting one data byte, the SCI automatically loads the next transmit data from the TDR into the TSR and starts transmitting again. If the TDRE bit of the SSR is 1, however, the SCI does not load the TDR contents into the TSR.

The CPU cannot read or write the TSR directly.

Bit:	7	6	5	4	3	2	1	0
R/W:	—	—	—	—	—	—	—	—

14.2.4 Transmit Data Register (TDR)

The transmit data register (TDR) is an 8-bit register that stores data for serial transmission. When the SCI detects that the transmit shift register (TSR) is empty, it moves transmit data written in the TDR into the TSR and starts serial transmission. Continuous serial transmission is possible by writing the next transmit data in the TDR during serial transmission from the TSR.

The CPU can always read and write the TDR. The TDR is initialized to H'FF by a power-on reset or in standby mode. Manual reset does not initialize TDR.

Bit:	7	6	5	4	3	2	1	0
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

14.2.5 Serial Mode Register (SMR)

The serial mode register (SMR) is an 8-bit register that specifies the SCI serial communication format and selects the clock source for the baud rate generator.

The CPU can always read and write the SMR. The SMR is initialized to H'00 by a power-on reset or in standby mode. Manual reset does not initialize SMR.

Bit:	7	6	5	4	3	2	1	0
	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- Bit 7—Communication Mode (C/$\overline{A}$): Selects whether the SCI operates in the asynchronous or clock synchronous mode.

Bit 7: C/$\overline{A}$	Description
0	Asynchronous mode (initial value)
1	Clocked synchronous mode

- Bit 6—Character Length (CHR): Selects 7-bit or 8-bit data in the asynchronous mode. In the clock synchronous mode, the data length is always eight bits, regardless of the CHR setting.

Bit 6: CHR	Description
0	Eight-bit data (initial value)
1	Seven-bit data. (When 7-bit data is selected, the MSB (bit 7) of the transmit data register is not transmitted.)

- Bit 5—Parity Enable (PE): Selects whether to add a parity bit to transmit data and to check the parity of receive data, in the asynchronous mode. In the clock synchronous mode, a parity bit is neither added nor checked, regardless of the PE setting.

Bit 5: PE	Description
0	Parity bit not added or checked (initial value)
1	Parity bit added and checked. When PE is set to 1, an even or odd parity bit is added to transmit data, depending on the parity mode (O/$\overline{E}$) setting. Receive data parity is checked according to the even/odd (O/$\overline{E}$) mode setting.

- Bit 4—Parity Mode (O/$\overline{E}$): Selects even or odd parity when parity bits are added and checked. The O/$\overline{E}$ setting is used only in asynchronous mode and only when the parity enable bit (PE) is set to 1 to enable parity addition and check. The O/$\overline{E}$ setting is ignored in the clock synchronous mode, or in the asynchronous mode when parity addition and check is disabled.

Bit 4: O/$\overline{E}$	Description
0	Even parity (initial value). If even parity is selected, the parity bit is added to transmit data to make an even number of 1s in the transmitted character and parity bit combined. Receive data is checked to see if it has an even number of 1s in the received character and parity bit combined.
1	Odd parity. If odd parity is selected, the parity bit is added to transmit data to make an odd number of 1s in the transmitted character and parity bit combined. Receive data is checked to see if it has an odd number of 1s in the received character and parity bit combined.

- Bit 3—Stop Bit Length (STOP): Selects one or two bits as the stop bit length in the asynchronous mode. This setting is used only in the asynchronous mode. It is ignored in the clock synchronous mode because no stop bits are added.

  In receiving, only the first stop bit is checked, regardless of the STOP bit setting. If the second stop bit is 1, it is treated as a stop bit, but if the second stop bit is 0, it is treated as the start bit of the next incoming character.

Bit 3: STOP	Description
0	One stop bit (initial value). In transmitting, a single bit of 1 is added at the end of each transmitted character.
1	Two stop bits. In transmitting, two bits of 1 are added at the end of each transmitted character.

- Bit 2—Multiprocessor Mode (MP): Selects multiprocessor format. When multiprocessor format is selected, settings of the parity enable (PE) and parity mode (O/$\overline{E}$) bits are ignored. The MP bit setting is used only in the asynchronous mode; it is ignored in the clock synchronous mode. For the multiprocessor communication function, see section 14.3.3, Multiprocessor Communication.

Bit 2: MP	Description
0	Multiprocessor function disabled (initial value)
1	Multiprocessor format selected

- Bits 1 and 0—Clock Select 1 and 0 (CKS1 and CKS0): These bits select the internal clock source of the on-chip baud rate generator. Four clock sources are available; ϕ, $\phi/4$, $\phi/16$, or $\phi/64$. For further information on the clock source, bit rate register settings, and baud rate, see section 14.2.8, Bit Rate Register (BRR).

Bit 1: CKS1	Bit 0: CKS0	Description
0	0	ϕ (initial value)
	1	$\phi/4$
1	0	$\phi/16$
	1	$\phi/64$

14.2.6 Serial Control Register (SCR)

The serial control register (SCR) operates the SCI transmitter/receiver, selects the serial clock output in the asynchronous mode, enables/disables interrupt requests, and selects the transmit/receive clock source. The CPU can always read and write the SCR. The SCR is initialized to H'00 by a power-on reset or in standby mode. Manual reset does not initialize SCR.

Bit:	7	6	5	4	3	2	1	0
	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- Bit 7—Transmit Interrupt Enable (TIE): Enables or disables the transmit-data-empty interrupt (TxI) requested when the transmit data register empty bit (TDRE) in the serial status register (SSR) is set to 1 by transfer of serial transmit data from the TDR to the TSR.

Bit 7: TIE	Description
0	Transmit-data-empty interrupt request (TxI) is disabled (initial value). The TxI interrupt request can be cleared by reading TDRE after it has been set to 1, then clearing TDRE to 0, or by clearing TIE to 0.
1	Transmit-data-empty interrupt request (TxI) is enabled

- Bit 6—Receive Interrupt Enable (RIE): Enables or disables the receive-data-full interrupt (RxI) requested when the receive data register full bit (RDRF) in the serial status register (SSR) is set to 1 by transfer of serial receive data from the RSR to the RDR. It also enables or disables receive-error interrupt (ERI) requests.

Bit 6: RIE	Description
0	Receive-data-full interrupt (RxI) and receive-error interrupt (ERI) requests are disabled (initial value). RxI and ERI interrupt requests can be cleared by reading the RDRF flag or error flag (FER, PER, or ORER) after it has been set to 1, then clearing the flag to 0, or by clearing RIE to 0.
1	Receive-data-full interrupt (RxI) and receive-error interrupt (ERI) requests are enabled.

- Bit 5—Transmit Enable (TE): Enables or disables the SCI serial transmitter.

Bit 5: TE	Description
0	Transmitter disabled (initial value). The transmit data register empty bit (TDRE) in the serial status register (SSR) is locked at 1.
1	Transmitter enabled. Serial transmission starts when the transmit data register empty (TDRE) bit in the serial status register (SSR) is cleared to 0 after writing of transmit data into the TDR. Select the transmit format in the SMR before setting TE to 1.

- Bit 4—Receive Enable (RE): Enables or disables the SCI serial receiver.

Bit 4: RE	Description
0	Receiver disabled (initial value). Clearing RE to 0 does not affect the receive flags (RDRF, FER, PER, ORER). These flags retain their previous values.
1	Receiver enabled. Serial reception starts when a start bit is detected in the asynchronous mode, or synchronous clock input is detected in the clock synchronous mode. Select the receive format in the SMR before setting RE to 1.

- Bit 3—Multiprocessor Interrupt Enable (MPIE): Enables or disables multiprocessor interrupts. The MPIE setting is used only in the asynchronous mode, and only if the multiprocessor mode bit (MP) in the serial mode register (SMR) is set to 1 during reception. The MPIE setting is ignored in the clock synchronous mode or when the MP bit is cleared to 0.

Bit 3: MPIE	Description
0	Multiprocessor interrupts are disabled (normal receive operation) (initial value). MPIE is cleared when the MPIE bit is cleared to 0, or the multiprocessor bit (MPB) is set to 1 in receive data.
1	Multiprocessor interrupts are enabled. Receive-data-full interrupt requests (RxI), receive-error interrupt requests (ERI), and setting of the RDRF, FER, and ORER status flags in the serial status register (SSR) are disabled until data with the multiprocessor bit set to 1 is received. The SCI does not transfer receive data from the RSR to the RDR, does not detect receive errors, and does not set the RDRF, FER, and ORER flags in the serial status register (SSR). When it receives data that includes MPB = 1, MPB is set to 1, and the SCI automatically clears MPIE to 0, generates RxI and ERI interrupts (if the TIE and RIE bits in the SCR are set to 1), and allows the FER and ORER bits to be set.

- Bit 2—Transmit-End Interrupt Enable (TEIE): Enables or disables the transmit-end interrupt (TEI) requested if TDR does not contain valid transmit data when the MSB is transmitted.

Bit 2: TEIE	Description
0	Transmit-end interrupt (TEI) requests are disabled.* (initial value)
1	Transmit-end interrupt (TEI) requests are enabled.*

Note: * The TEI request can be cleared by reading the TDRE bit in the serial status register (SSR) after it has been set to 1, then clearing TDRE to 0 and clearing the transmit end (TEND) bit to 0; or by clearing the TEIE bit to 0.

- Bits 1 and 0—Clock Enable 1 and 0 (CKE1 and CKE0): These bits select the SCI clock source and enable or disable clock output from the SCK pin. Depending on the combination of CKE1 and CKE0, the SCK pin can be used for serial clock output, or serial clock input. Select the SCK pin function by using the pin function controller (PFC).

  The CKE0 setting is valid only in the asynchronous mode, and only when the SCI is internally clocked (CKE1 = 0). The CKE0 setting is ignored in the clock synchronous mode, or when an external clock source is selected (CKE1 = 1). Select the SCI operating mode in the serial mode register (SMR) before setting CKE1 and CKE0. For further details on selection of the SCI clock source, see table 14.9 in section 14.3, Operation.

Bit 1: CKE1	Bit 0: CKE0	Description*1	
0	0	Asynchronous mode	Internal clock, SCK pin used for input pin (input signal is ignored) or output pin (output level is undefined)*2
		Clock synchronous mode	Internal clock, SCK pin used for synchronous clock output*2
0	1	Asynchronous mode	Internal clock, SCK pin used for clock output*3
		Clock synchronous mode	Internal clock, SCK pin used for synchronous clock output
1	0	Asynchronous mode	External clock, SCK pin used for clock input*4
		Clock synchronous mode	External clock, SCK pin used for synchronous clock input
1	1	Asynchronous mode	External clock, SCK pin used for clock input*4
		Clock synchronous mode	External clock, SCK pin used for synchronous clock input

Notes: *1 The SCK pin is multiplexed with other functions. Use the pin function controller (PFC) to select the SCK function for this pin, as well as the I/O direction.

*2 Initial value.

*3 The output clock frequency is the same as the bit rate.

*4 The input clock frequency is 16 times the bit rate.

14.2.7 Serial Status Register (SSR)

The serial status register (SSR) is an 8-bit register containing multiprocessor bit values, and status flags that indicate SCI operating status.

The CPU can always read and write the SSR, but cannot write 1 in the status flags (TDRE, RDRF, ORER, PER, and FER). These flags can be cleared to 0 only if they have first been read (after being set to 1). Bits 2 (TEND) and 1 (MPB) are read-only bits that cannot be written. The SSR is initialized to H'84 by a power-on reset or in standby mode. Manual reset does not initialize SSR.

Bit:	7	6	5	4	3	2	1	0
	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT
Initial value:	1	0	0	0	0	1	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R/W

Note: * The only value that can be written is a 0 to clear the flag.

- Bit 7—Transmit Data Register Empty (TDRE): Indicates that the SCI has loaded transmit data from the TDR into the TSR and new serial transmit data can be written in the TDR.

Bit 7: TDRE	Description
0	TDR contains valid transmit data TDRE is cleared to 0 when software reads TDRE after it has been set to 1, then writes 0 in TDRE or the DMAC or DTC writes data in TDR
1	TDR does not contain valid transmit data (initial value) TDRE is set to 1 when the chip is power-on reset or enters standby mode, the TE bit in the serial control register (SCR) is cleared to 0, or TDR contents are loaded into TSR, so new data can be written in TDR

- Bit 6—Receive Data Register Full (RDRF): Indicates that RDR contains received data.

Bit 6: RDRF	Description
0	RDR does not contain valid received data (initial value) RDRF is cleared to 0 when the chip is power-on reset or enters standby mode, software reads RDRF after it has been set to 1, then writes 0 in RDRF, or the DMAC or DTC reads data from RDR
1	RDR contains valid received data RDRF is set to 1 when serial data is received normally and transferred from RSR to RDR

Note: The RDR and RDRF are not affected by detection of receive errors or by clearing of the RE bit to 0 in the serial control register. They retain their previous contents. If RDRF is still set to 1 when reception of the next data ends, an overrun error (ORER) occurs and the received data is lost.

- Bit 5—Overrun Error (ORER): Indicates that data reception ended abnormally due to an overrun error.

Bit 5: ORER	Description
0	Receiving is in progress or has ended normally (initial value). Clearing the RE bit to 0 in the serial control register does not affect the ORER bit, which retains its previous value. ORER is cleared to 0 when the chip is power-on reset or enters standby mode or software reads ORER after it has been set to 1, then writes 0 in ORER
1	A receive overrun error occurred. RDR continues to hold the data received before the overrun error, so subsequent receive data is lost. Serial receiving cannot continue while ORER is set to 1. In the clock synchronous mode, serial transmitting is disabled. ORER is set to 1 if reception of the next serial data ends when RDRF is set to 1

- Bit 4—Framing Error (FER): Indicates that data reception ended abnormally due to a framing error in the asynchronous mode.

Bit 4: FER	Description
0	Receiving is in progress or has ended normally (initial value). Clearing the RE bit to 0 in the serial control register does not affect the FER bit, which retains its previous value. FER is cleared to 0 when the chip is power-on reset or enters standby mode or software reads FER after it has been set to 1, then writes 0 in FER
1	A receive framing error occurred. When the stop bit length is two bits, only the first bit is checked to see if it is a 1. The second stop bit is not checked. When a framing error occurs, the SCI transfers the receive data into the RDR but does not set RDRF. Serial receiving cannot continue while FER is set to 1. In the clock synchronous mode, serial transmitting is also disabled. FER is set to 1 if the stop bit at the end of receive data is checked and found to be 0

- Bit 3—Parity Error (PER): Indicates that data reception (with parity) ended abnormally due to a parity error in the asynchronous mode.

Bit 3: PER	Description
0	Receiving is in progress or has ended normally (initial value). Clearing the RE bit to 0 in the serial control register does not affect the PER bit, which retains its previous value. PER is cleared to 0 when the chip is power-on reset or enters standby mode or software reads PER after it has been set to 1, then writes 0 in PER
1	A receive parity error occurred. When a parity error occurs, the SCI transfers the receive data into the RDR but does not set RDRF. Serial receiving cannot continue while PER is set to 1. In the clock synchronous mode, serial transmitting is also disabled. PER is set to 1 if the number of 1s in receive data, including the parity bit, does not match the even or odd parity setting of the parity mode bit (O/$\overline{E}$) in the serial mode register (SMR)

- Bit 2—Transmit End (TEND): Indicates that when the last bit of a serial character was transmitted, the TDR did not contain valid data, so transmission has ended. TEND is a read-only bit and cannot be written.

Bit 2: TEND	Description
0	Transmission is in progress TEND is cleared to 0 when software reads TDRE after it has been set to 1, then writes 0 in TDRE, or the DMAC or DTC writes data in TDR
1	End of transmission (initial value) TEND is set to 1 when the chip is power-on reset or enters standby mode, TE is cleared to 0 in the serial control register (SCR), or TDRE is 1 when the last bit of a one-byte serial character is transmitted.

- Bit 1—Multiprocessor Bit (MPB): Stores the value of the multiprocessor bit in receive data when a multiprocessor format is selected for receiving in the asynchronous mode. The MPB is a read-only bit and cannot be written.

Bit 1: MPB	Description
0	Multiprocessor bit value in receive data is 0 (initial value). If RE is cleared to 0 when a multiprocessor format is selected, the MPB retains its previous value.
1	Multiprocessor bit value in receive data is 1

- Bit 0—Multiprocessor Bit Transfer (MPBT): Stores the value of the multiprocessor bit added to transmit data when a multiprocessor format is selected for transmitting in the asynchronous mode. The MPBT setting is ignored in the clock synchronous mode, when a multiprocessor format is not selected, or when the SCI is not transmitting.

Bit 0: MPBT	Description
0	Multiprocessor bit value in transmit data is 0 (initial value)
1	Multiprocessor bit value in transmit data is 1

14.2.8 Bit Rate Register (BRR)

The bit rate register (BRR) is an 8-bit register that, together with the baud rate generator clock source selected by the CKS1 and CKS0 bits in the serial mode register (SMR), determines the serial transmit/receive bit rate.

The CPU can always read and write the BRR. The BRR is initialized to H'FF by a power-on reset or in standby mode. Each channel has independent baud rate generator control, so different values can be set in the two channels. Manual reset does not initialize BRR.

Bit:	7	6	5	4	3	2	1	0
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 14.3 lists examples of BRR settings in the asynchronous mode; table 14.4 lists examples of BBR settings in the clock synchronous mode.

Table 14.3 Bit Rates and BRR Settings in Asynchronous Mode

	φ (MHz)								
Bit Rate	**4**			**4.9152**			**6**		
(Bits/s)	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**
110	2	70	0.03	2	86	0.31	2	106	–0.44
150	1	207	0.16	1	255	0.00	2	77	0.16
300	1	103	0.16	1	127	0.00	1	155	0.16
600	0	207	0.16	0	255	0.00	1	77	0.16
1200	0	103	0.16	0	127	0.00	0	155	0.16
2400	0	51	0.16	0	63	0.00	0	77	0.16
4800	0	25	0.16	0	31	0.00	0	38	0.16
9600	0	12	0.16	0	15	0.00	0	19	–2.34
14400	0	8	–3.55	0	10	–3.03	0	12	0.16
19200	0	6	–6.99	0	7	0.00	0	9	–2.34
28800	0	3	8.51	0	4	6.67	0	6	–6.99
31250	0	3	0.00	0	4	–1.70	0	5	0.00
38400	0	2	8.51	0	3	0.00	0	4	–2.34

Table 14.3 Bit Rates and BRR Settings in Asynchronous Mode (cont)

Bit Rate (Bits/s)	φ (MHz)								
	7.3728			8			9.8304		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	130	–0.07	2	141	0.03	2	174	–0.26
150	2	95	0.00	2	103	0.16	2	127	0.00
300	1	191	0.00	1	207	0.16	1	255	0.00
600	1	95	0.00	1	103	0.16	1	127	0.00
1200	0	191	0.00	0	207	0.16	0	255	0.00
2400	0	95	0.00	0	103	0.16	0	127	0.00
4800	0	47	0.00	0	51	0.16	0	63	0.00
9600	0	23	0.00	0	25	0.16	0	31	0.00
14400	0	15	0.00	0	16	2.12	0	20	1.59
19200	0	11	0.00	0	12	0.16	0	15	0.00
28800	0	7	0.00	0	8	–3.55	0	10	–3.03
31250	0	6	5.33	0	7	0.00	0	9	–1.70
38400	0	5	0.00	0	6	–6.99	0	7	0.00

Table 14.3 Bit Rates and BRR Settings in Asynchronous Mode (cont)

Bit Rate (Bits/s)	φ (MHz)								
	10			11.0592			12		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	177	–0.25	2	195	0.19	2	212	0.03
150	2	129	0.16	2	143	0.00	2	155	0.16
300	2	64	0.16	2	71	0.00	2	77	0.16
600	1	129	0.16	1	143	0.00	1	155	0.16
1200	1	64	0.16	1	71	0.00	1	77	0.16
2400	0	129	0.16	0	143	0.00	0	155	0.16
4800	0	64	0.16	0	71	0.00	0	77	0.16
9600	0	32	–1.36	0	35	0.00	0	38	0.16
14400	0	21	–1.36	0	23	0.00	0	25	0.16
19200	0	15	1.73	0	17	0.00	0	19	–2.34
28800	0	10	–1.36	0	11	0.00	0	12	0.16
31250	0	9	0.00	0	10	0.54	0	11	0.00
38400	0	7	1.73	0	8	0.00	0	9	–2.34

Table 14.3 Bit Rates and BRR Settings in Asynchronous Mode (cont)

Bit Rate (Bits/s)	φ (MHz)								
	12.288			14			14.7456		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	217	0.08	2	248	–0.17	3	64	0.70
150	2	159	0.00	2	181	0.16	2	191	0.00
300	2	79	0.00	2	90	0.16	2	95	0.00
600	1	159	0.00	1	181	0.16	1	191	0.00
1200	1	79	0.00	1	90	0.16	1	95	0.00
2400	0	159	0.00	0	181	0.16	0	191	0.00
4800	0	79	0.00	0	90	0.16	0	95	0.00
9600	0	39	0.00	0	45	–0.93	0	47	0.00
14400	0	26	–1.23	0	29	1.27	0	31	0.00
19200	0	19	0.00	0	22	–0.93	0	23	0.00
28800	0	12	2.56	0	14	1.27	0	15	0.00
31250	0	11	2.40	0	13	0.00	0	14	–1.70
38400	0	9	0.00	0	10	3.57	0	11	0.00

Table 14.3 Bit Rates and BRR Settings in Asynchronous Mode (cont)

Bit Rate (Bits/s)	φ (MHz)								
	16			17.2032			18		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	70	0.03	3	75	0.48	3	79	–0.12
150	2	207	0.16	2	223	0.00	2	233	0.16
300	2	103	0.16	2	111	0.00	2	116	0.16
600	1	207	0.16	1	223	0.00	1	233	0.16
1200	1	103	0.16	1	111	0.00	1	116	0.16
2400	0	207	0.16	0	223	0.00	0	233	0.16
4800	0	103	0.16	0	111	0.00	0	116	0.16
9600	0	51	0.16	0	55	0.00	0	58	–0.69
14400	0	34	–0.79	0	36	0.90	0	38	0.16
19200	0	25	0.16	0	27	0.00	0	28	1.02
28800	0	16	2.12	0	18	–1.75	0	19	–2.34
31250	0	15	0.00	0	16	1.20	0	17	0.00
38400	0	12	0.16	0	13	0.00	0	14	–2.34

Table 14.3 Bit Rates and BRR Settings in Asynchronous Mode (cont)

Bit Rate (Bits/s)	φ (MHz)								
	18.432			19.6608			20		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	81	–0.22	3	86	0.31	3	88	–0.25
150	2	239	0.00	2	255	0.00	3	64	0.16
300	2	119	0.00	2	127	0.00	2	129	0.16
600	1	239	0.00	1	255	0.00	2	64	0.16
1200	1	119	0.00	1	127	0.00	1	129	0.16
2400	0	239	0.00	0	255	0.00	1	64	0.16
4800	0	119	0.00	0	127	0.00	0	129	0.16
9600	0	59	0.00	0	63	0.00	0	64	0.16
14400	0	39	0.00	0	42	–0.78	0	42	0.94
19200	0	29	0.00	0	31	0.00	0	32	–1.36
28800	0	19	0.00	0	20	1.59	0	21	–1.36
31250	0	17	2.40	0	19	–1.70	0	19	0.00
38400	0	14	0.00	0	15	0.00	0	15	1.73

Table 14.3 Bit Rates and BRR Settings in Asynchronous Mode (cont)

Bit Rate (Bits/s)	φ (MHz)								
	22			22.1184			24		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	97	–0.35	3	97	0.19	3	106	–0.44
150	3	71	–0.54	3	71	0.00	3	77	0.16
300	2	142	0.16	2	143	0.00	2	155	0.16
600	2	71	–0.54	2	71	0.00	2	77	0.16
1200	1	142	0.16	1	143	0.00	1	155	0.16
2400	1	71	–0.54	1	71	0.00	1	77	0.16
4800	0	142	0.16	0	143	0.00	0	155	0.16
9600	0	71	–0.54	0	71	0.00	0	77	0.16
14400	0	47	–0.54	0	47	0.00	0	51	0.16
19200	0	35	–0.54	0	35	0.00	0	38	0.16
28800	0	23	–0.54	0	23	0.00	0	25	0.16
31250	0	21	0.00	0	21	0.54	0	23	0.00
38400	0	17	–0.54	0	17	0.00	0	19	–2.34

Table 14.3 Bit Rates and BRR Settings in Asynchronous Mode (cont)

Bit Rate (Bits/s)	φ (MHz)								
	24.576			25.8048			26		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	108	0.08	3	114	–0.40	3	114	0.36
150	3	79	0.00	3	83	0.00	3	84	–0.43
300	2	159	0.00	2	167	0.00	2	168	0.16
600	2	79	0.00	2	83	0.00	2	84	–0.43
1200	1	159	0.00	1	167	0.00	1	168	0.16
2400	1	79	0.00	1	83	0.00	1	84	–0.43
4800	0	159	0.00	0	167	0.00	0	168	0.16
9600	0	79	0.00	0	83	0.00	0	84	–0.43
14400	0	52	0.63	0	55	0.00	0	55	0.76
19200	0	39	0.00	0	41	0.00	0	41	0.76
28800	0	26	–1.23	0	27	0.00	0	27	0.76
31250	0	24	–1.70	0	25	–0.75	0	25	0.00
38400	0	19	0.00	0	20	0.00	0	20	0.76

Table 14.3 Bit Rates and BRR Settings in Asynchronous Mode (cont)

Bit Rate (Bits/s)	ϕ (MHz)								
	27.0336			28			29.4912		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	119	0.00	3	123	0.23	3	130	-0.07
150	3	87	0.00	3	90	0.16	3	95	0.00
300	2	175	0.00	2	181	0.16	2	191	0.00
600	1	87	0.00	2	90	0.16	2	95	0.00
1200	1	175	0.00	1	181	0.16	1	191	0.00
2400	1	87	0.00	1	90	0.16	1	95	0.00
4800	0	175	0.00	0	181	0.16	0	191	0.00
9600	0	87	0.00	0	90	0.16	0	95	0.00
14400	0	58	–0.56	0	60	–0.39	0	63	0.00
19200	0	43	0.00	0	45	0.93	0	47	0.00
28800	0	28	1.15	0	29	1.27	0	31	0.00
31250	0	26	0.12	0	27	0.00	0	28	1.69
38400	0	21	0.00	0	22	–0.93	0	23	0.00

Table 14.3 Bit Rates and BRR Settings in Asynchronous Mode (cont)

Bit Rate (Bits/s)	φ (MHz)								
	30			31.9488			32		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	132	0.13	3	141	-0.13	3	141	0.03
150	3	97	-0.35	3	103	0.00	3	103	0.16
300	2	194	0.16	2	207	0.00	2	207	0.16
600	2	97	-0.35	2	103	0.00	2	103	0.16
1200	1	194	0.16	1	207	0.00	1	207	0.16
2400	1	97	-0.35	1	103	0.00	1	103	0.16
4800	0	194	0.16	0	207	0.00	0	207	0.16
9600	0	97	-0.35	0	103	0.00	0	103	0.16
14400	0	64	0.16	0	68	0.48	0	68	0.64
19200	0	48	-0.35	0	51	0.00	0	51	0.16
28800	0	32	-1.36	0	34	-0.95	0	34	-0.79
31250	0	29	0.00	0	31	-0.16	0	31	0.00
38400	0	23	1.73	0	25	0.00	0	25	0.16

Table 14.3 Bit Rates and BRR Settings in Asynchronous Mode (cont)

Bit Rate (Bits/s)	ϕ (MHz)								
	33			33.1776			33.3333		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	145	0.33	3	146	0.19	3	147	-0.02
150	3	106	0.39	3	107	0.00	3	108	-0.45
300	2	214	-0.07	2	215	0.00	2	216	0.01
600	2	106	0.39	2	107	0.00	2	108	-0.45
1200	1	214	-0.07	1	215	0.00	1	216	0.01
2400	1	106	0.39	1	107	0.00	1	108	-0.45
4800	0	214	-0.07	0	215	0.00	0	216	0.01
9600	0	106	0.39	0	107	0.00	0	108	-0.45
14400	0	71	–0.54	0	91	0.00	0	91	0.47
19200	0	53	–0.54	0	53	0.00	0	53	0.47
28800	0	35	–0.54	0	35	0.00	0	35	0.47
31250	0	32	0.00	0	32	0.54	0	32	1.01
38400	0	26	–0.54	0	26	0.00	0	26	0.47

Table 14.4 Bit Rates and BRR Settings in Clocked Synchronous Mode

Bit Rate (Bits/s)	φ (MHz)							
	4		8		10		12	
	n	N	n	N	n	N	n	N
110	3	141						
250	2	249	3	124	3	155	3	187
500	2	124	2	249	3	77	3	93
1k	1	249	2	124	2	155	2	187
2.5k	1	99	1	199	1	249	2	74
5k	0	199	1	99	1	124	1	149
10k	0	99	0	199	0	249	1	74
25k	0	39	0	79	0	99	0	119
50k	0	19	0	39	0	49	0	59
100k	0	9	0	19	0	24	0	29
250k	0	3	0	7	0	9	0	11
500k	0	1	0	3	0	4	0	5
1M	0	0*	0	1	—	—	0	2
2.5M					0	0*	0	0*
5M								

Table 14.4 Bit Rates and BRR Settings in Clocked Synchronous Mode (cont)

Bit Rate (Bits/s)	φ (MHz)							
	16		20		24		28	
	n	N	n	N	n	N	n	N
110								
250	3	249						
500	3	124	3	155	3	187	3	218
1k	2	249	3	77	3	93	3	108
2.5k	2	99	2	124	2	149	2	174
5k	1	199	1	249	2	74	2	87
10k	1	99	1	124	1	149	1	174
25k	0	159	0	199	0	239	1	69
50k	0	79	0	99	0	119	0	139
100k	0	39	0	49	0	59	0	69
250k	0	15	0	19	0	23	0	27
500k	0	7	0	9	0	11	0	13
1M	0	3	0	4	0	5	0	6
2.5M	—	—	0	1	—	—	0	2
3.5M			—	—	—	—	0	1
5M			0	0*	—	—	—	—
7M					—	—	0	0*

Table 14.4 Bit Rates and BRR Settings in Clocked Synchronous Mode (cont)

Bit Rate (Bits/s)	φ (MHz)							
	30		32		33		33.3333	
	n	N	n	N	n	N	n	N
110								
250								
500	3	233	3	249				
1k	3	116	3	124	3	128	3	129
2.5k	2	187	2	199	2	205	2	207
5k	2	93	2	99	2	102	2	103
10k	1	187	1	199	1	205	1	207
25k	1	74	1	79	1	82	1	82
50k	0	149	0	159	0	164	0	166
100k	0	74	0	79	0	82	0	82
250k	0	29	0	31	0	32	0	32
500k	0	14	0	15	0	16	0	16
1M	0	7	0	7	0	7	0	7
2.5M	0	2	0	2	0	2	—	—
5M	—	—	—	—	—	—	—	—
7M	—	—	—	—	—	—	—	—

Note: Settings with an error of 1% or less are recommended.

Legend

Blank: No setting available

—: Setting possible, but error occurs

*: Continuous transmission/reception is not possible.

The BRR setting is calculated as follows:

Asynchronous mode:

$$N = \frac{\phi}{64 \times 2^{2n-1} \times B} \times 10^6 - 1$$

Synchronous mode:

$$N = \frac{\phi}{64 \times 2^{2n-1} \times B} \times 10^6 - 1$$

B: Bit rate (bit/s)
N: Baud rate generator BRR setting ($0 \leq N \leq 255$)
ϕ: Operating frequency (MHz)
n: Baud rate generator input clock (n = 0 to 3)
(See the following table for the clock sources and value of n.)

		SMR Settings	
n	**Clock Source**	**CKS1**	**CKS2**
0	ϕ	0	0
1	$\phi/4$	0	1
2	$\phi/16$	1	0
3	$\phi/64$	1	1

The bit rate error in asynchronous mode is calculated as follows:

$$\text{Error (\%)} = \left\{ \frac{\phi \times 10^6}{(N+1) \times B \times 64 \times 2^{2n-1}} - 1 \right\} \times 100$$

Table 14.5 indicates the maximum bit rates in the asynchronous mode when the baud rate generator is being used for various frequencies. Tables 14.6 and 14.7 show the maximum rates for external clock input.

Table 14.5 Maximum Bit Rates for Various Frequencies with Baud Rate Generator (Asynchronous Mode)

		Settings	
ϕ (MHz)	Maximum Bit Rate (Bits/s)	n	N
4	125000	0	0
4.9152	153600	0	0
6	187500	0	0
7.3728	230400	0	0
8	250000	0	0
9.8304	307200	0	0
10	312500	0	0
11.0592	345600	0	0
12	375000	0	0
12.288	384000	0	0
14	437500	0	0
14.7456	460800	0	0
16	500000	0	0
17.2032	537600	0	0
18	562500	0	0
18.432	576000	0	0
19.6608	614400	0	0
20	625000	0	0
22	687500	0	0
22.1184	691200	0	0
24	750000	0	0
24.576	768000	0	0
25.8048	806400	0	0
26	812500	0	0
27.0336	844800	0	0
28	875000	0	0
29.4912	921600	0	0
30	937500	0	0
31.9488	998400	0	0
32	1000000	0	0
33	1031250	0	0
33.1776	1036800	0	0
33.3333	1041666	0	0

Table 14.6 Maximum Bit Rates during External Clock Input (Asynchronous Mode)

φ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (Bits/s)
4	1.0000	62500
4.9152	1.2288	76800
6	1.5000	93750
7.3728	1.8432	115200
8	2.0000	125000
9.8304	2.4576	153600
10	2.5000	156250
11.0592	2.7648	172800
12	3.0000	187500
12.288	3.0720	192000
14	3.5000	218750
14.7456	3.6864	230400
16	4.0000	250000
17.2032	4.3008	268800
18	4.5000	281250
18.432	4.6080	288000
19.6608	4.9152	307200
20	5.0000	312500
22	5.5000	343750
22.1184	5.5296	345600
24	6.0000	375000
24.576	6.1440	384000
25.8048	6.4512	403200
26	6.5000	406250
27.0336	6.7584	422400
28	7.0000	437500
29.4912	7.3728	460800
30	7.5000	468750
31.9488	7.9872	499200
32	8.0000	500000
33	8.2500	515625
33.1776	8.2944	518400
33.3333	8.3333	520832.8125

Table 14.7 Maximum Bit Rates during External Clock Input (Clock Synchronous Mode)

φ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (Bits/s)
4	0.6667	666666.7
6	1.0000	1000000.0
8	1.3333	1333333.3
10	1.6667	1666666.7
12	2.0000	2000000.0
14	2.3333	2333333.3
16	2.6667	2666666.7
18	3.0000	3000000.0
20	3.3333	3333333.3
22	3.6667	3666666.7
24	4.0000	4000000.0
26	4.3333	4333333.3
28	4.6667	4666666.7
30	5.0000	5000000.0
32	5.3333	5333333.3
33.3333	5.5556	5555550.0

14.3 Operation

14.3.1 Overview

For serial communication, the SCI has an asynchronous mode in which characters are synchronized individually, and a clock synchronous mode in which communication is synchronized with clock pulses. Asynchronous/clock synchronous mode and the transmission format are selected in the serial mode register (SMR), as shown in table 14.8. The SCI clock source is selected by the C/$\overline{A}$ bit in the serial mode register (SMR) and the CKE1 and CKE0 bits in the serial control register (SCR), as shown in table 14.9.

Asynchronous Mode:

- Data length is selectable: seven or eight bits.
- Parity and multiprocessor bits are selectable, as well as the stop bit length (one or two bits). These selections determine the transmit/receive format and character length.
- In receiving, it is possible to detect framing errors (FER), parity errors (PER), overrun errors (ORER), and the break state.
- An internal or external clock can be selected as the SCI clock source.
  - — When an internal clock is selected, the SCI operates using the on-chip baud rate generator clock, and can output a clock with a frequency matching the bit rate.
  - — When an external clock is selected, the external clock input must have a frequency 16 times the bit rate. (The on-chip baud rate generator is not used.)

Clock Synchronous Mode:

- The communication format has a fixed 8-bit data length.
- In receiving, it is possible to detect overrun errors (ORER).
- An internal or external clock can be selected as the SCI clock source.
  - — When an internal clock is selected, the SCI operates using the on-chip baud rate generator clock, and outputs a synchronous clock signal to external devices.
  - — When an external clock is selected, the SCI operates on the input synchronous clock. The on-chip baud rate generator is not used.

Table 14.8 Serial Mode Register Settings and SCI Communication Formats

	SMR Settings					SCI Communication Format			
Mode	Bit 7 C/$\overline{A}$	Bit 6 CHR	Bit 5 PE	Bit 2 MP	Bit 3 STOP	Data Length	Parity Bit	Multipro-cessor Bit	Stop Bit Length
Asynchronous	0	0	0	0	0	8-bit	Not set	Not set	1 bit
					1				2 bits
			1		0		Set		1 bit
					1				2 bits
		1	0		0	7-bit	Not set		1 bit
					1				2 bits
			1		0		Set		1 bit
					1				2 bits
Asynchronous (multiprocessor format)		0	*	1	0	8-bit	Not set	Set	1 bit
			*		1				2 bits
		1	*		0	7-bit			1 bit
			*		1				2 bits
Clock synchronous	1	*	*	*	*	8-bit		Not set	None

Note: Asterisks (*) in the table indicate don't-care bits.

Table 14.9 SMR and SCR Settings and SCI Clock Source Selection

	SMR	SCR Settings		SCI Transmit/Receive Clock	
Mode	Bit 7 C/$\overline{A}$	Bit 1 CKE1	Bit 0 CKE0	Clock Source	SCK Pin Function*
Asynchronous	0	0	0	Internal	SCI does not use the SCK pin
			1		Outputs a clock with frequency matching the bit rate
		1	0	External	Inputs a clock with frequency 16 times the bit rate
			1		
Clock synch-ronous	1	0	0	Internal	Outputs the synchronous clock
			1		
		1	0	External	Inputs the synchronous clock
			1		

Note: * Select the function in combination with the pin function controller (PFC).

14.3.2 Operation in Asynchronous Mode

In the asynchronous mode, each transmitted or received character begins with a start bit and ends with a stop bit. Serial communication is synchronized one character at a time.

The transmitting and receiving sections of the SCI are independent, so full duplex communication is possible. The transmitter and receiver are both double buffered, so data can be written and read while transmitting and receiving are in progress, enabling continuous transmitting and receiving.

Figure 14.2 shows the general format of asynchronous serial communication. In asynchronous serial communication, the communication line is normally held in the marking (high) state. The SCI monitors the line and starts serial communication when the line goes to the space (low) state, indicating a start bit. One serial character consists of a start bit (low), data (LSB first), parity bit (high or low), and stop bit (high), in that order.

When receiving in the asynchronous mode, the SCI synchronizes on the falling edge of the start bit. The SCI samples each data bit on the eighth pulse of a clock with a frequency 16 times the bit rate. Receive data is latched at the center of each bit.

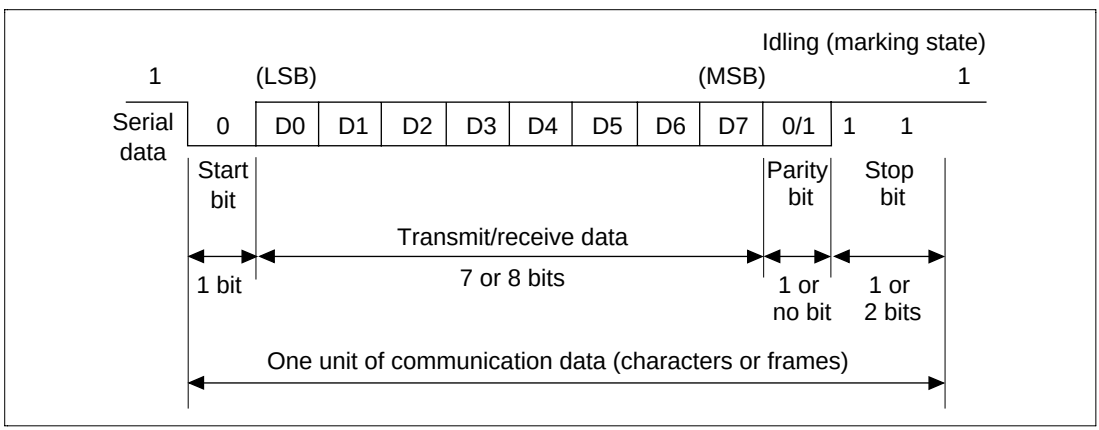


Figure 14.2 Data Format in Asynchronous Communication (Example: 8-bit Data with Parity and Two Stop Bits)

Transmit/Receive Formats: Table 14.10 shows the 11 communication formats that can be selected in the asynchronous mode. The format is selected by settings in the serial mode register (SMR).

Table 14.10 Serial Communication Formats (Asynchronous Mode)

SMR Bits				Serial Transmit/Receive Format and Frame Length											
CHR	PE	MP	STOP	1	2	3	4	5	6	7	8	9	10	11	12
0	0	0	0	START	8-Bit data								STOP		
0	0	0	1	START	8-Bit data								STOP	STOP	
0	1	0	0	START	8-Bit data								P	STOP	
0	1	0	1	START	8-Bit data								P	STOP	STOP
1	0	0	0	START	7-Bit data							STOP			
1	0	0	1	START	7-Bit data							STOP	STOP		
1	1	0	0	START	7-Bit data							P	STOP		
1	1	0	1	START	7-Bit data							P	STOP	STOP	
0	—	1	0	START	8-Bit data								MPB	STOP	
0	—	1	1	START	8-Bit data								MPB	STOP	STOP
1	—	1	0	START	7-Bit data							MPB	STOP		
1	—	1	1	START	7-Bit data							MPB	STOP	STOP	

—: Don't care bits.

Note: START: Start bit
STOP: Stop bit
P: Parity bit
MPB: Multiprocessor bit

Clock: An internal clock generated by the on-chip baud rate generator or an external clock input from the SCK pin can be selected as the SCI transmit/receive clock. The clock source is selected by the C/$\overline{A}$ bit in the serial mode register (SMR) and bits CKE1 and CKE0 in the serial control register (SCR) (table 14.9).

When an external clock is input at the SCK pin, it must have a frequency equal to 16 times the desired bit rate.

When the SCI operates on an internal clock, it can output a clock signal at the SCK pin. The frequency of this output clock is equal to the bit rate. The phase is aligned as in figure 14.3 so that the rising edge of the clock occurs at the center of each transmit data bit.

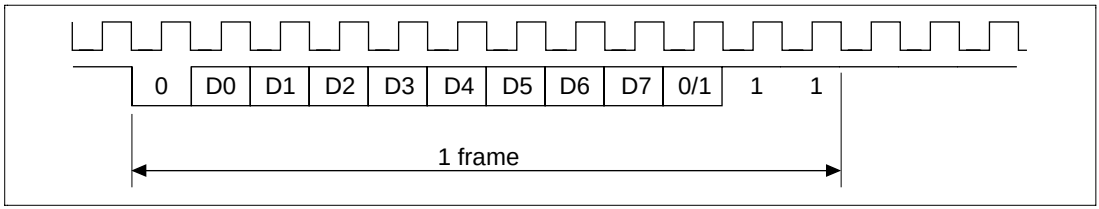


Figure 14.3 Output Clock and Communication Data Phase Relationship (Asynchronous Mode)

SCI Initialization (Asynchronous Mode): Before transmitting or receiving, clear the TE and RE bits to 0 in the serial control register (SCR), then initialize the SCI as follows.

When changing the operation mode or communication format, always clear the TE and RE bits to 0 before following the procedure given below. Clearing TE to 0 sets TDRE to 1 and initializes the transmit shift register (TSR). Clearing RE to 0, however, does not initialize the RDRF, PER, FER, and ORER flags and receive data register (RDR), which retain their previous contents.

When an external clock is used, the clock should not be stopped during initialization or subsequent operation. SCI operation becomes unreliable if the clock is stopped.

Figure 14.4 is a sample flowchart for initializing the SCI. The procedure is as follows (the steps correspond to the numbers in the flowchart):

1. Select the clock source in the serial control register (SCR). Leave RIE, TIE, TEIE, MPIE, TE, and RE cleared to 0. If clock output is selected in asynchronous mode, clock output starts immediately after the setting is made to SCR.
2. Select the communication format in the serial mode register (SMR).
3. Write the value corresponding to the bit rate in the bit rate register (BRR) unless an external clock is used.
4. Wait for at least the interval required to transmit or receive one bit, then set TE or RE in the serial control register (SCR) to 1. Also set RIE, TIE, TEIE, and MPIE as necessary. Setting TE or RE enables the SCI to use the TxD or RxD pin. The initial states are the marking transmit state, and the idle receive state (waiting for a start bit).

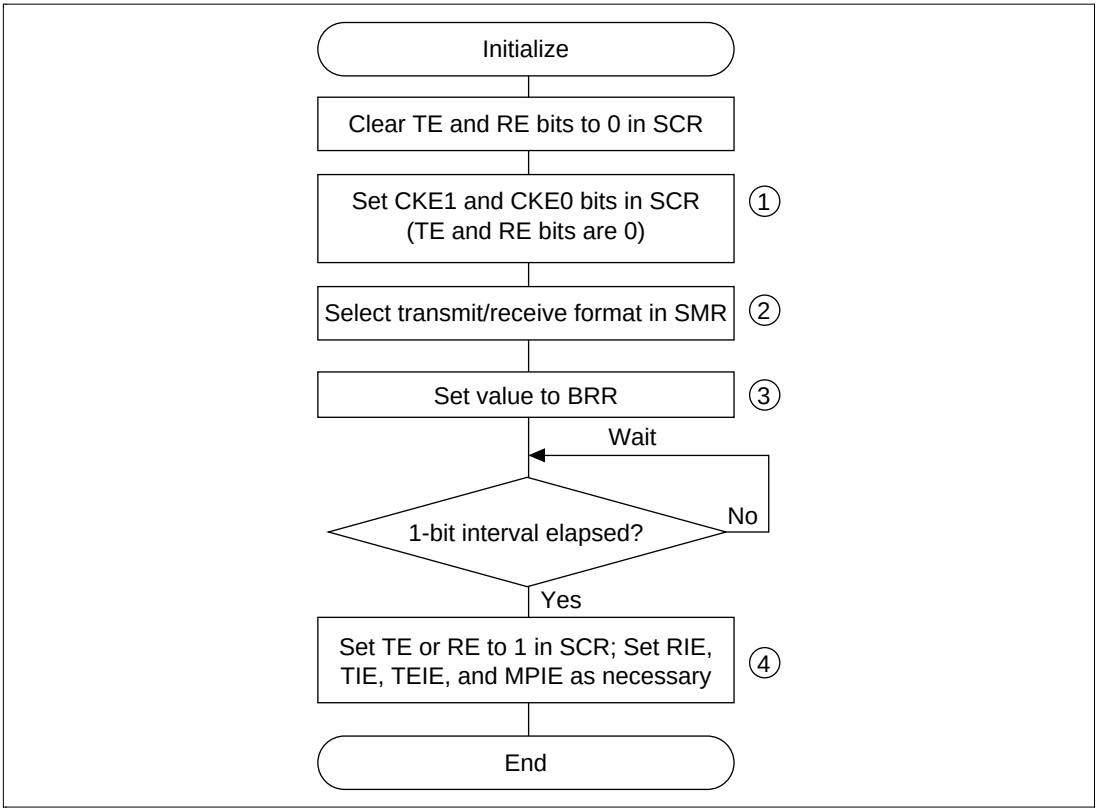


Figure 14.4 Sample Flowchart for SCI Initialization

Transmitting Serial Data (Asynchronous Mode): Figure 14.5 shows a sample flowchart for transmitting serial data. The procedure is as follows (the steps correspond to the numbers in the flowchart):

1. SCI initialization: Set the TxD pin using the PFC.
2. SCI status check and transmit data write: Read the serial status register (SSR), check that the TDRE bit is 1, then write transmit data in the transmit data register (TDR) and clear TDRE to 0.
3. Continue transmitting serial data: Read the TDRE bit to check whether it is safe to write (if it reads 1); if so, write data in TDR, then clear TDRE to 0. When the DMAC or the DTC is started by a transmit-data-empty interrupt request (TxI) in order to write data in TDR, the TDRE bit is checked and cleared automatically.
4. To output a break at the end of serial transmission, first clear the port data register (DR) to 0, then clear the TE to 0 in SCR and use the PFC to establish the TxD pin as an output port.

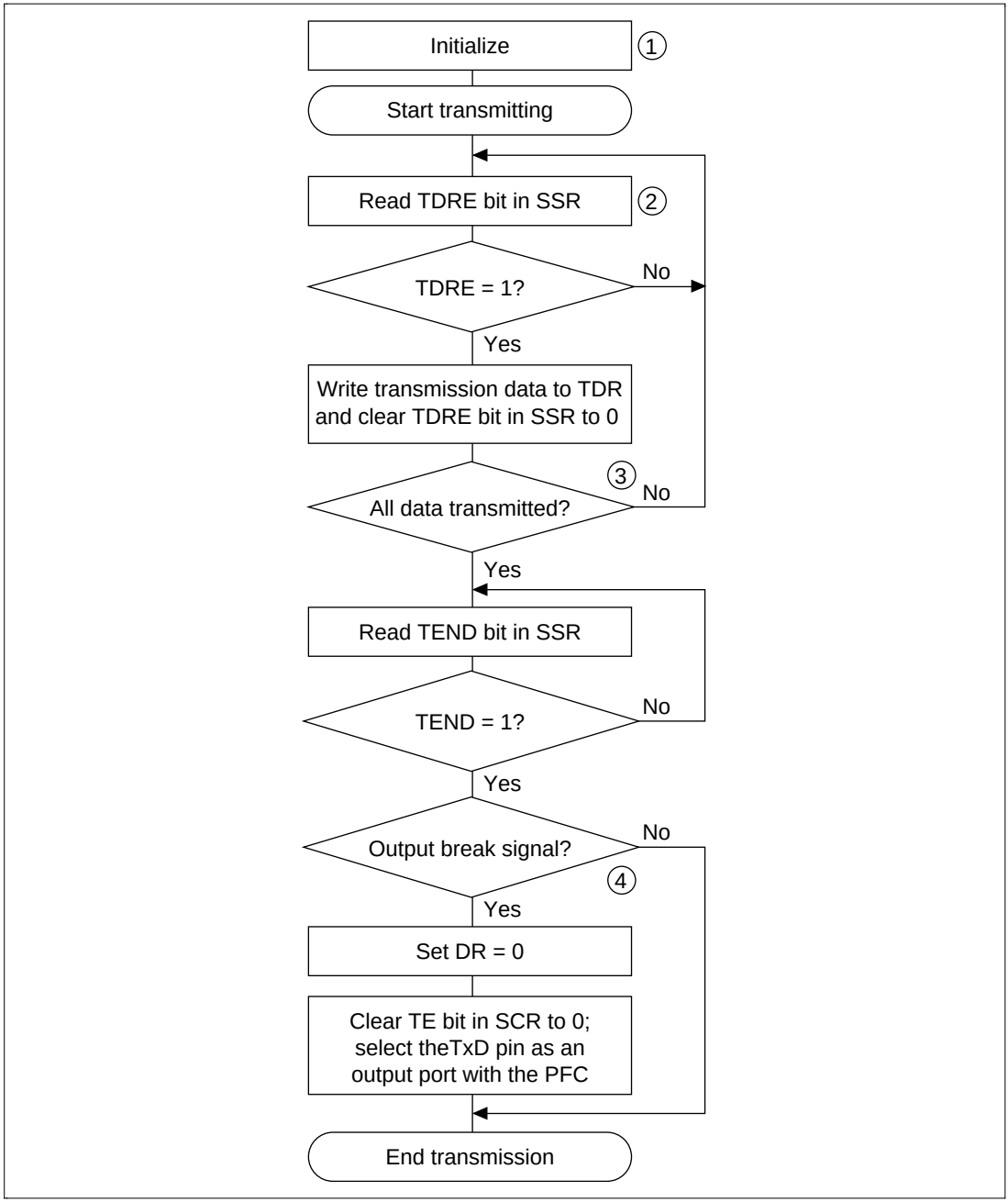


Figure 14.5 Sample Flowchart for Transmitting Serial Data

In transmitting serial data, the SCI operates as follows:

1. The SCI monitors the TDRE bit in the SSR. When TDRE is cleared to 0, the SCI recognizes that the transmit data register (TDR) contains new data, and loads this data from the TDR into the transmit shift register (TSR).
2. After loading the data from the TDR into the TSR, the SCI sets the TDRE bit to 1 and starts transmitting. If the transmit-data-empty interrupt enable bit (TIE) is set to 1 in the SCR, the SCI requests a transmit-data-empty interrupt (TxI) at this time.

   Serial transmit data is transmitted in the following order from the TxD pin:

   a. Start bit: one 0 bit is output.
   b. Transmit data: seven or eight bits of data are output, LSB first.
   c. Parity bit or multiprocessor bit: one parity bit (even or odd parity) or one multiprocessor bit is output. Formats in which neither a parity bit nor a multiprocessor bit is output can also be selected.
   d. Stop bit: one or two 1 bits (stop bits) are output.
   e. Marking: output of 1 bits continues until the start bit of the next transmit data.

3. The SCI checks the TDRE bit when it outputs the stop bit. If TDRE is 0, the SCI loads new data from the TDR into the TSR, outputs the stop bit, then begins serial transmission of the next frame. If TDRE is 1, the SCI sets the TEND bit to 1 in the SSR, outputs the stop bit, then continues output of 1 bits (marking). If the transmit-end interrupt enable bit (TEIE) in the SCR is set to 1, a transmit-end interrupt (TEI) is requested.

Figure 14.6 shows an example of SCI transmit operation in the asynchronous mode.

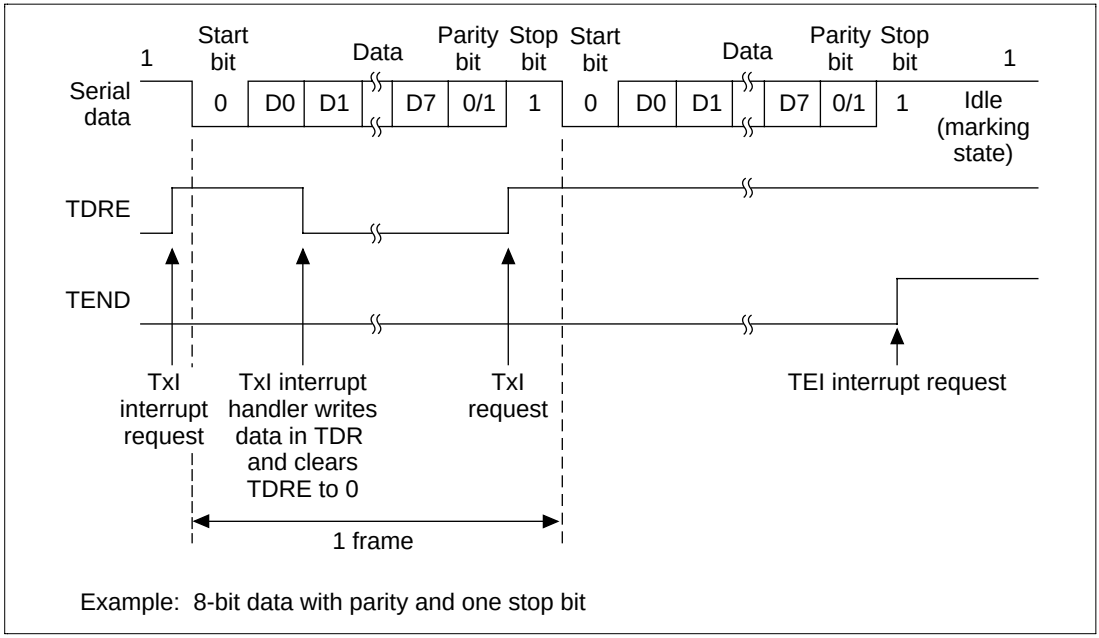


Figure 14.6 SCI Transmit Operation in Asynchronous Mode

Receiving Serial Data (Asynchronous Mode): Figures 14.7 and 14.8 show a sample flowchart for receiving serial data. The procedure is as follows (the steps correspond to the numbers in the flowchart).

1. SCI initialization: Set the RxD pin using the PFC.
2. Receive error handling and break detection: If a receive error occurs, read the ORER, PER, and FER bits of the SSR to identify the error. After executing the necessary error handling, clear ORER, PER, and FER all to 0. Receiving cannot resume if ORER, PER, or FER remain set to 1. When a framing error occurs, the RxD pin can be read to detect the break state.
3. SCI status check and receive-data read: Read the serial status register (SSR), check that RDRF is set to 1, then read receive data from the receive data register (RDR) and clear RDRF to 0. The RxI interrupt can also be used to determine if the RDRF bit has changed from 0 to 1.
4. Continue receiving serial data: Read the RDR and RDRF bit and clear RDRF to 0 before the stop bit of the current frame is received. If the DMAC or the DTC is started by a receive-data-full interrupt (RxI) to read RDR, the RDRF bit is cleared automatically so this step is unnecessary.

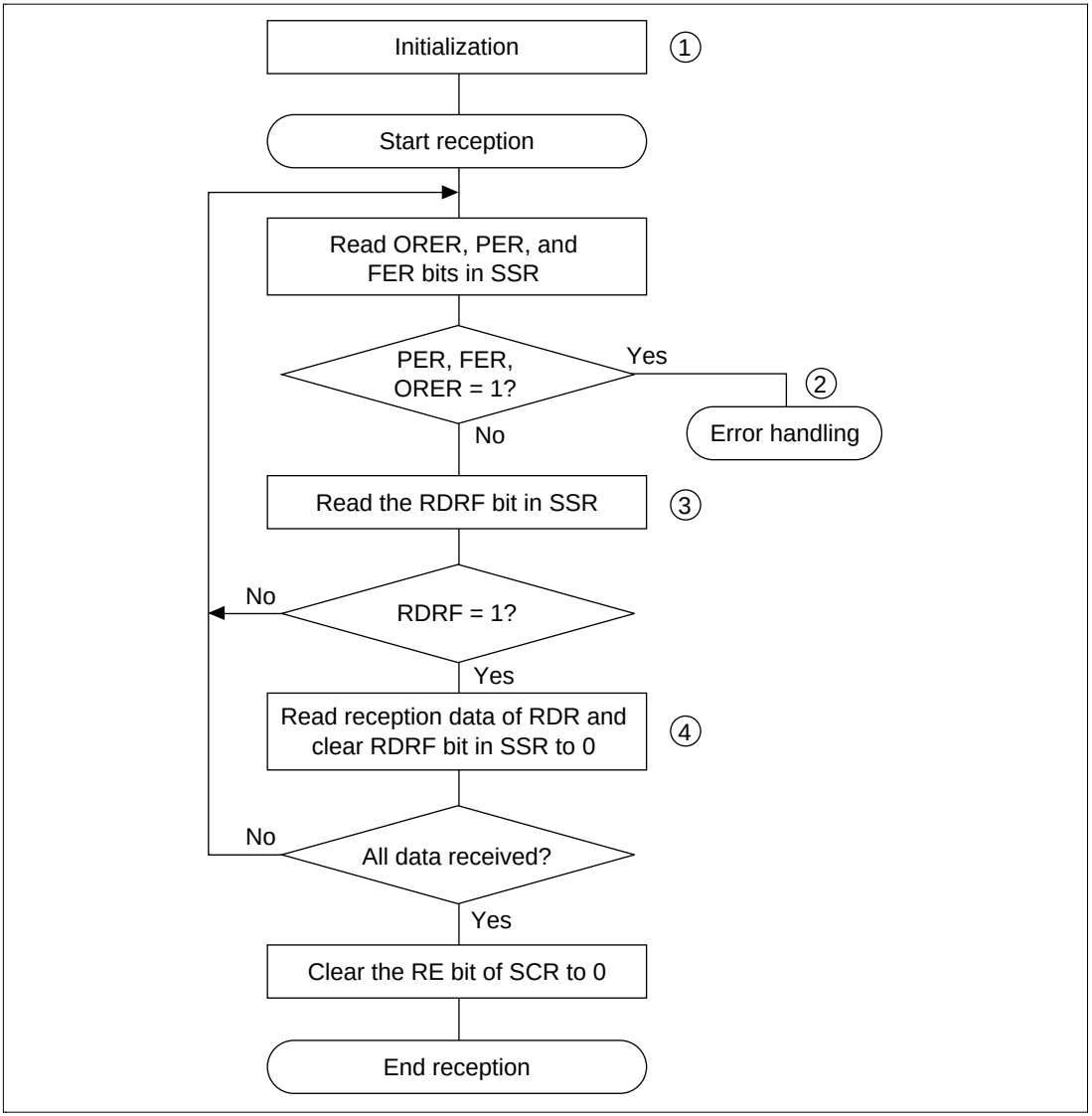


Figure 14.7 Sample Flowchart for Receiving Serial Data (1)

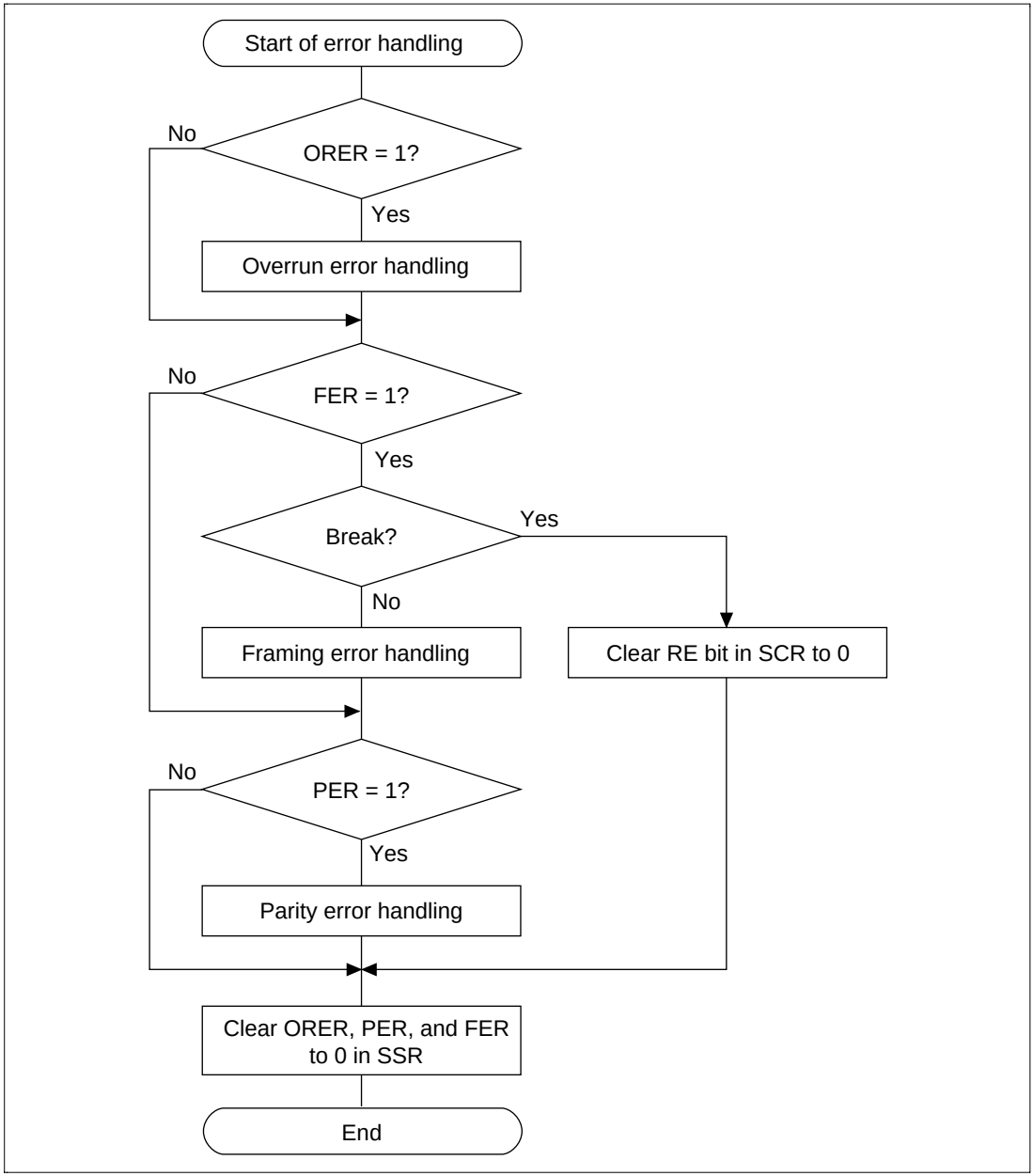


Figure 14.8 Sample Flowchart for Receiving Serial Data (2)

In receiving, the SCI operates as follows:

1. The SCI monitors the communication line. When it detects a start bit (0), the SCI synchronizes internally and starts receiving.
2. Receive data is shifted into the RSR in order from the LSB to the MSB.
3. The parity bit and stop bit are received. After receiving these bits, the SCI makes the following checks:
   a. Parity check. The number of 1s in the receive data must match the even or odd parity setting of the O/E bit in the SMR.
   b. Stop bit check. The stop bit value must be 1. If there are two stop bits, only the first stop bit is checked.
   c. Status check. RDRF must be 0 so that receive data can be loaded from the RSR into the RDR.

   If the data passes these checks, the SCI sets RDRF to 1 and stores the received data in the RDR. If one of the checks fails (receive error), the SCI operates as indicated in table 14.11.

   Note: When a receive error occurs, further receiving is disabled. While receiving, the RDRF bit is not set to 1, so be sure to clear the error flags.
4. After setting RDRF to 1, if the receive-data-full interrupt enable bit (RIE) is set to 1 in the SCR, the SCI requests a receive-data-full interrupt (RxI). If one of the error flags (ORER, PER, or FER) is set to 1 and the receive-data-full interrupt enable bit (RIE) in the SCR is also set to 1, the SCI requests a receive-error interrupt (ERI).

Figure 14.9 shows an example of SCI receive operation in the asynchronous mode.

Table 14.11 Receive Error Conditions and SCI Operation

Receive Error	Abbreviation	Condition	Data Transfer
Overrun error	ORER	Receiving of next data ends while RDRF is still set to 1 in SSR	Receive data not loaded from RSR into RDR
Framing error	FER	Stop bit is 0	Receive data loaded from RSR into RDR
Parity error	PER	Parity of receive data differs from even/odd parity setting in SMR	Receive data loaded from RSR into RDR

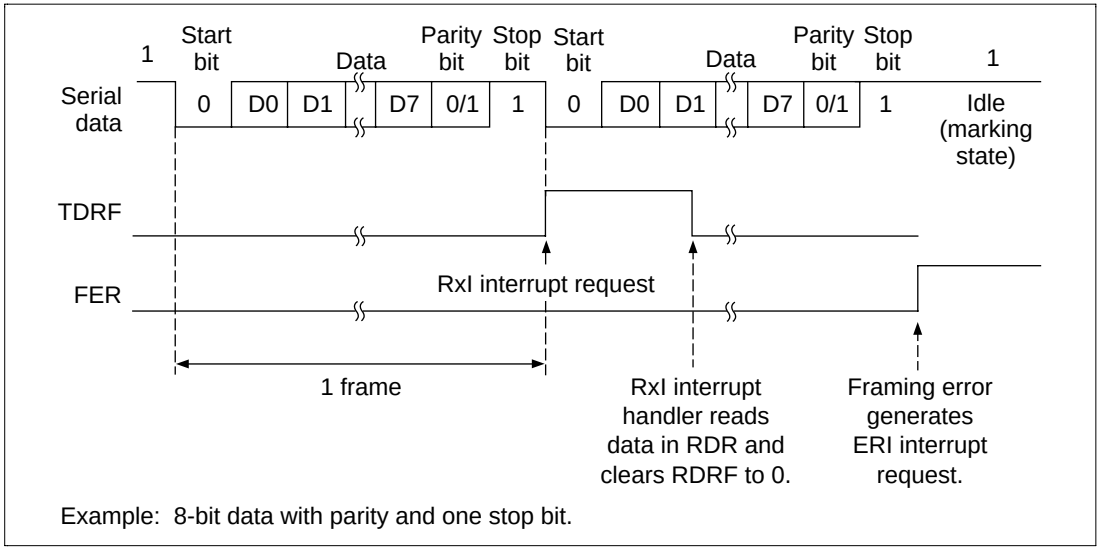


Figure 14.9 SCI Receive Operation

14.3.3 Multiprocessor Communication

The multiprocessor communication function enables several processors to share a single serial communication line for sending and receiving data. The processors communicate in the asynchronous mode using a format with an additional multiprocessor bit (multiprocessor format).

In multiprocessor communication, each receiving processor is addressed by a unique ID. A serial communication cycle consists of an ID-sending cycle that identifies the receiving processor, and a data-sending cycle. The multiprocessor bit distinguishes ID-sending cycles from data-sending cycles. The transmitting processor starts by sending the ID of the receiving processor with which it wants to communicate as data with the multiprocessor bit set to 1. Next the transmitting processor sends transmit data with the multiprocessor bit cleared to 0.

Receiving processors skip incoming data until they receive data with the multiprocessor bit set to 1. When they receive data with the multiprocessor bit set to 1, receiving processors compare the data with their IDs. The receiving processor with a matching ID continues to receive further incoming data. Processors with IDs not matching the received data skip further incoming data until they again receive data with the multiprocessor bit set to 1. Multiple processors can send and receive data in this way.

Figure 14.10 shows the example of communication among processors using the multiprocessor format.

Communication Formats: Four formats are available. Parity-bit settings are ignored when the multiprocessor format is selected. For details see table 14.8.

Clock: See the description in the asynchronous mode section.

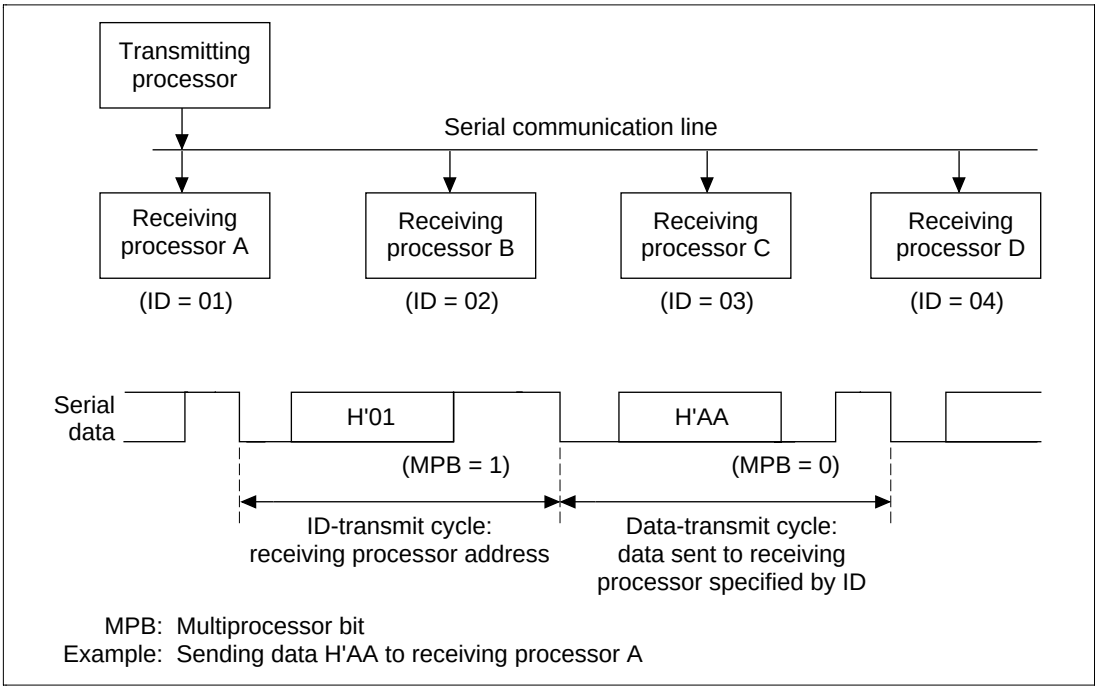


Figure 14.10 Communication among Processors Using Multiprocessor Format

Transmitting Multiprocessor Serial Data: Figure 14.11 shows a sample flowchart for transmitting multiprocessor serial data. The procedure is as follows (the steps correspond to the numbers in the flowchart):

1. SCI initialization: Set the TxD pin using the PFC.
2. SCI status check and transmit data write: Read the serial status register (SSR), check that the TDRE bit is 1, then write transmit data in the transmit data register (TDR). Also set MPBT (multiprocessor bit transfer) to 0 or 1 in SSR. Finally, clear TDRE to 0.
3. Continue transmitting serial data: Read the TDRE bit to check whether it is safe to write (if it reads 1); if so, write data in TDR, then clear TDRE to 0. When the DMAC or the DTC is started by a transmit-data-empty interrupt request (TxI) to write data in TDR, the TDRE bit is checked and cleared automatically.
4. Output a break at the end of serial transmission: Set the data register (DR) of the port to 0, then clear TE to 0 in SCR and set the TxD pin function as output port with the PFC.

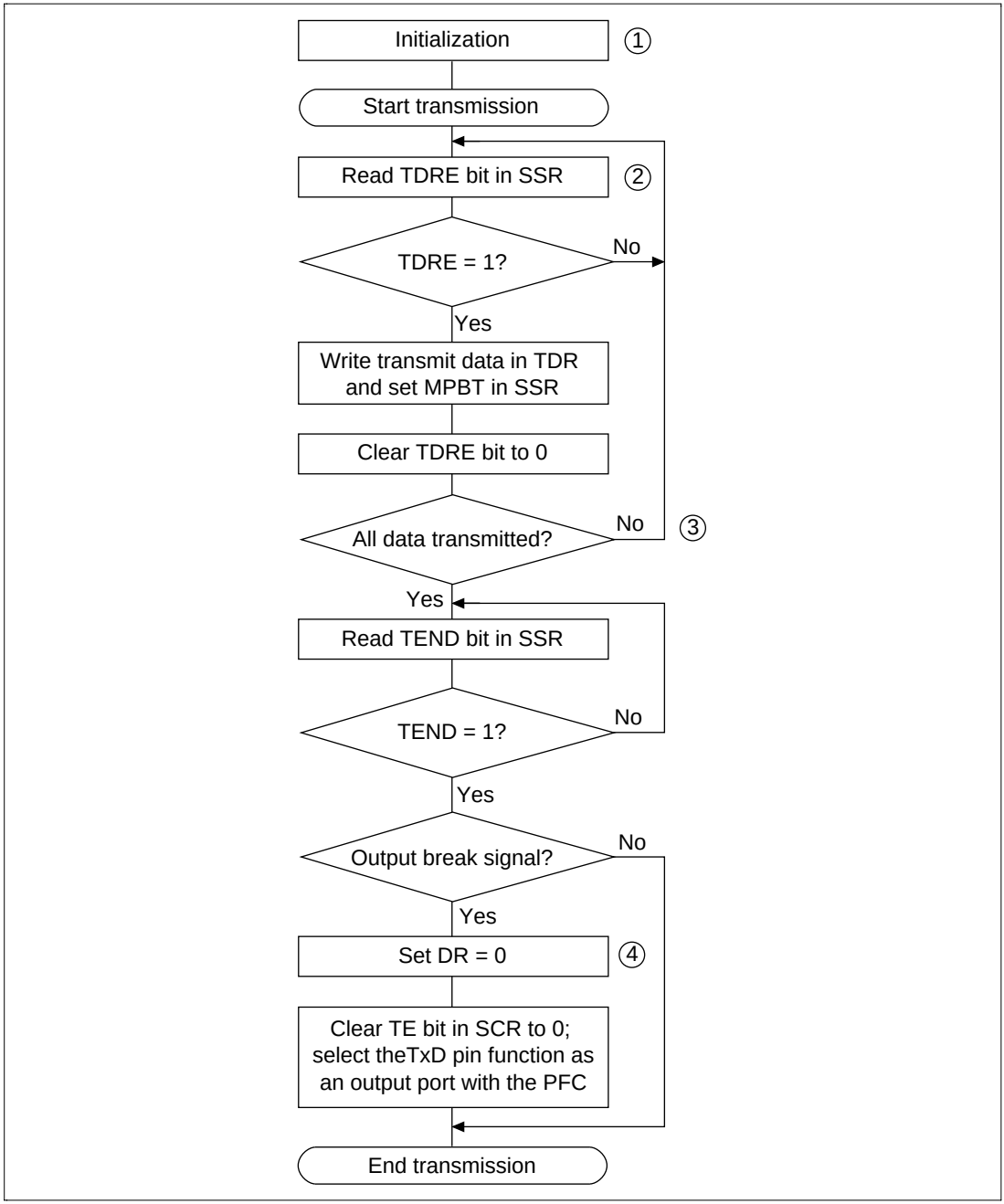


Figure 14.11 Sample Flowchart for Transmitting Multiprocessor Serial Data

In transmitting serial data, the SCI operates as follows:

1. The SCI monitors the TDRE bit in the SSR. When TDRE is cleared to 0 the SCI recognizes that the transmit data register (TDR) contains new data, and loads this data from the TDR into the transmit shift register (TSR).
2. After loading the data from the TDR into the TSR, the SCI sets the TDRE bit to 1 and starts transmitting. If the transmit-data-empty interrupt enable bit (TIE) in the SCR is set to 1, the SCI requests a transmit-data-empty interrupt (TxI) at this time.

   Serial transmit data is transmitted in the following order from the TxD pin:

   a. Start bit: one 0 bit is output.
   b. Transmit data: seven or eight bits are output, LSB first.
   c. Multiprocessor bit: one multiprocessor bit (MPBT value) is output.
   d. Stop bit: one or two 1 bits (stop bits) are output.
   e. Marking: output of 1 bits continues until the start bit of the next transmit data.

3. The SCI checks the TDRE bit when it outputs the stop bit. If TDRE is 0, the SCI loads data from the TDR into the TSR, outputs the stop bit, then begins serial transmission of the next frame. If TDRE is 1, the SCI sets the TEND bit in the SSR to 1, outputs the stop bit, then continues output of 1 bits in the marking state. If the transmit-end interrupt enable bit (TEIE) in the SCR is set to 1, a transmit-end interrupt (TEI) is requested at this time.

Figure 14.12 shows an example of SCI receive operation in the multiprocessor format.

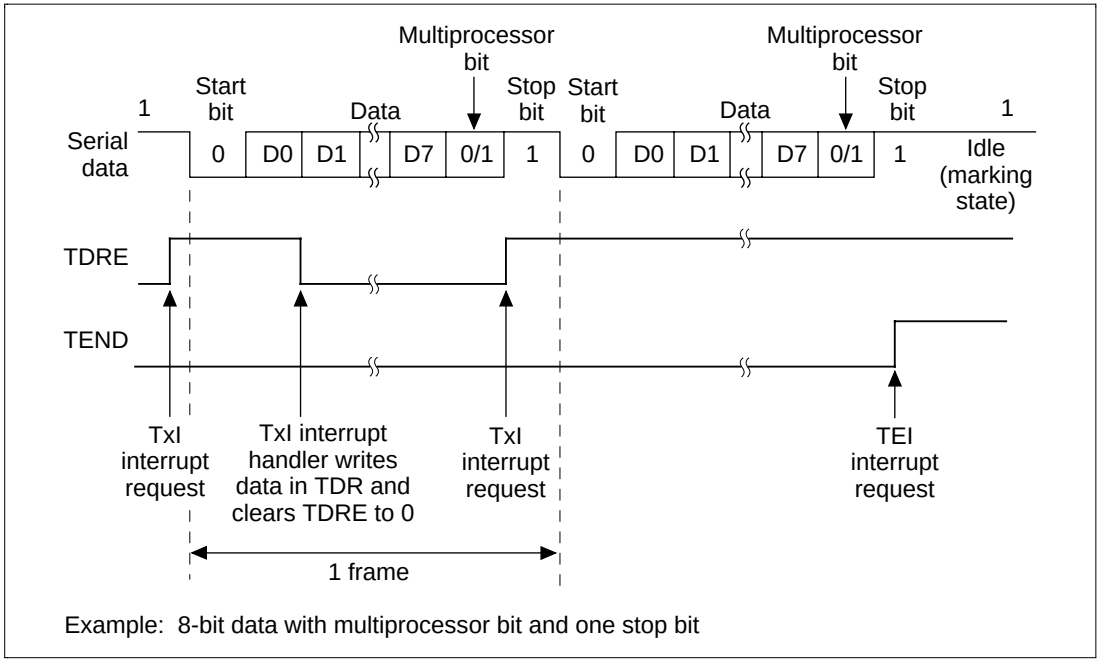


Figure 14.12 SCI Multiprocessor Transmit Operation

Receiving Multiprocessor Serial Data: Figure 14.13 shows a sample flowchart for receiving multiprocessor serial data. The procedure for receiving multiprocessor serial data is listed below.

1. SCI initialization: Set the RxD pin using the PFC.
2. ID receive cycle: Set the MPIE bit in the serial control register (SCR) to 1.
3. SCI status check and compare to ID reception: Read the serial status register (SSR), check that RDRF is set to 1, then read data from the receive data register (RDR) and compare with the processor's own ID. If the ID does not match the receive data, set MPIE to 1 again and clear RDRF to 0. If the ID matches the receive data, clear RDRF to 0.
4. Receive error handling and break detection: If a receive error occurs, read the ORER and FER bits in SSR to identify the error. After executing the necessary error processing, clear both ORER and FER to 0. Receiving cannot resume if ORER or FER remain set to 1. When a framing error occurs, the RxD pin can be read to detect the break state.
5. SCI status check and data receiving: Read SSR, check that RDRF is set to 1, then read data from the receive data register (RDR).

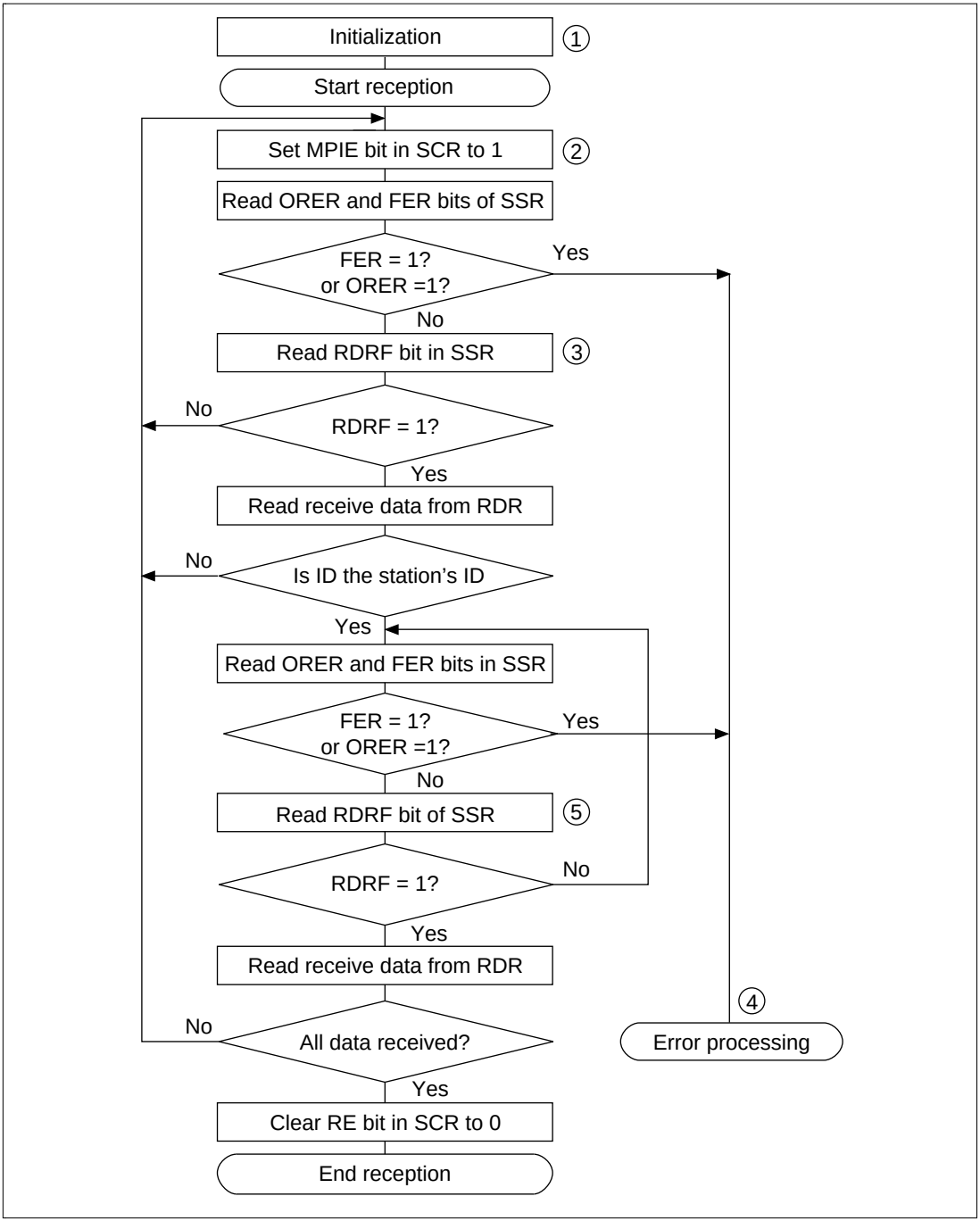


Figure 14.13 Sample Flowchart for Receiving Multiprocessor Serial Data

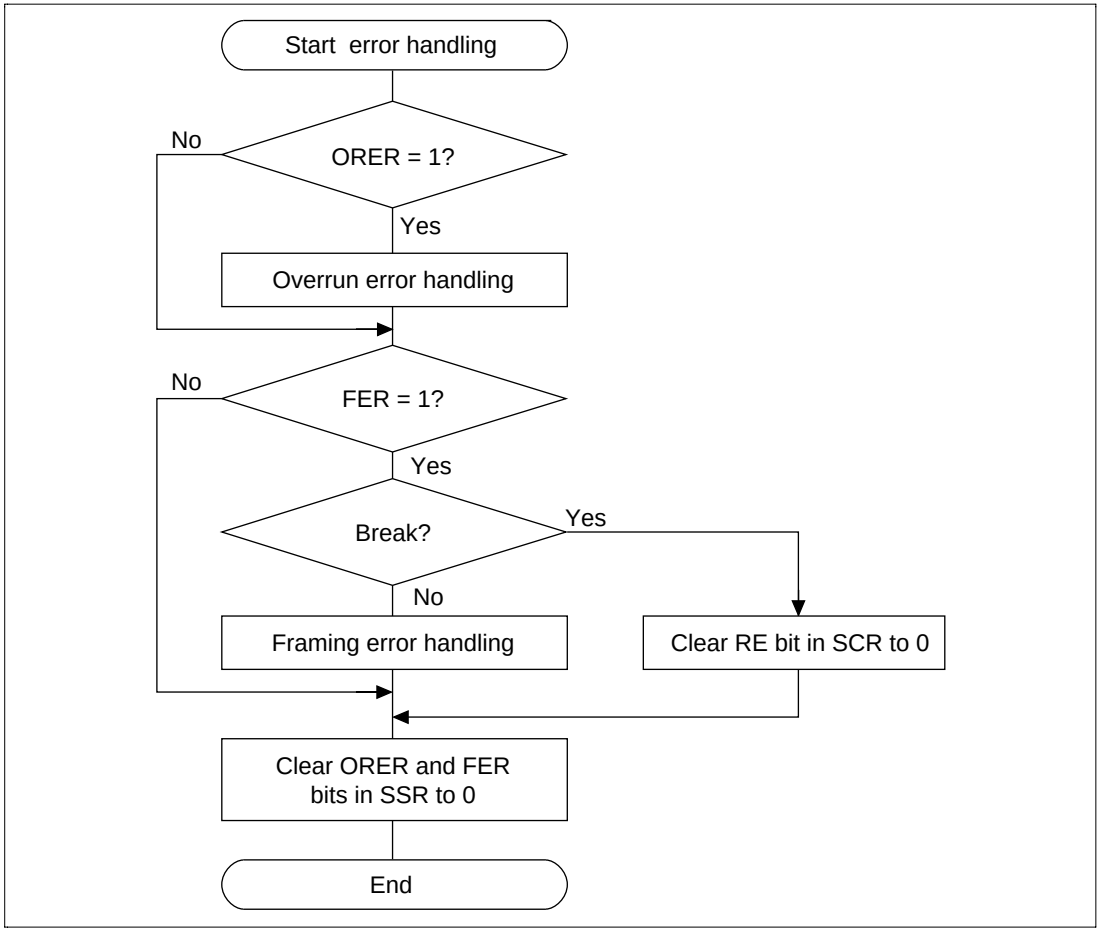


Figure 14.13 Sample Flowchart for Receiving Multiprocessor Serial Data (cont)

Figures 14.14 and 14.15 show examples of SCI receive operation using a multiprocessor format.

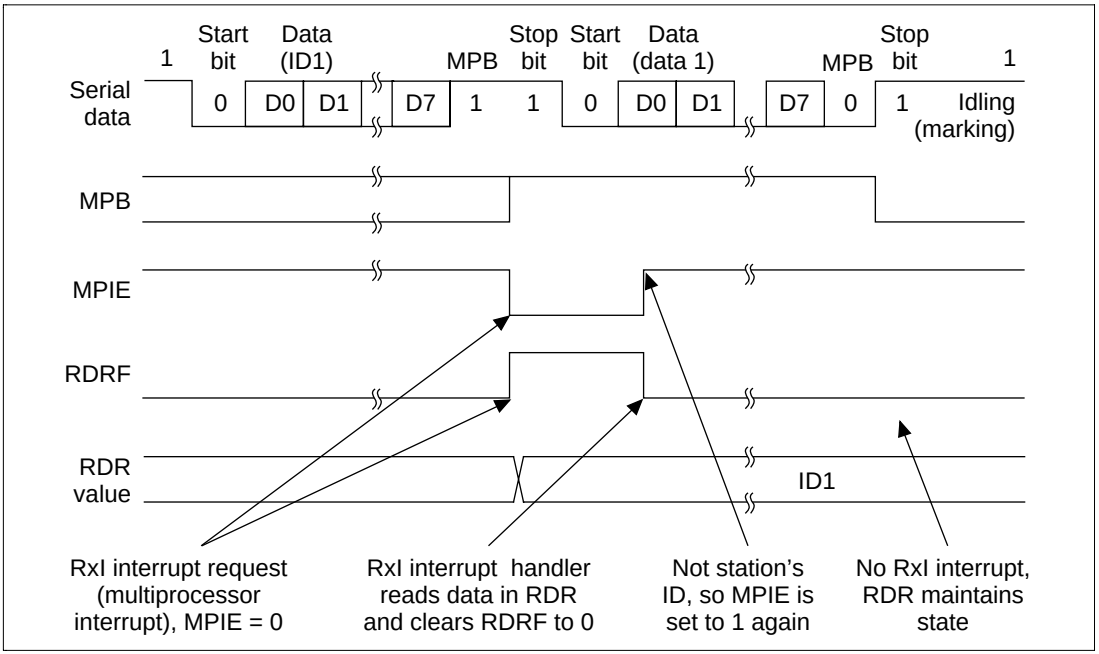


Figure 14.14 SCI Receive Operation (ID Does Not Match)

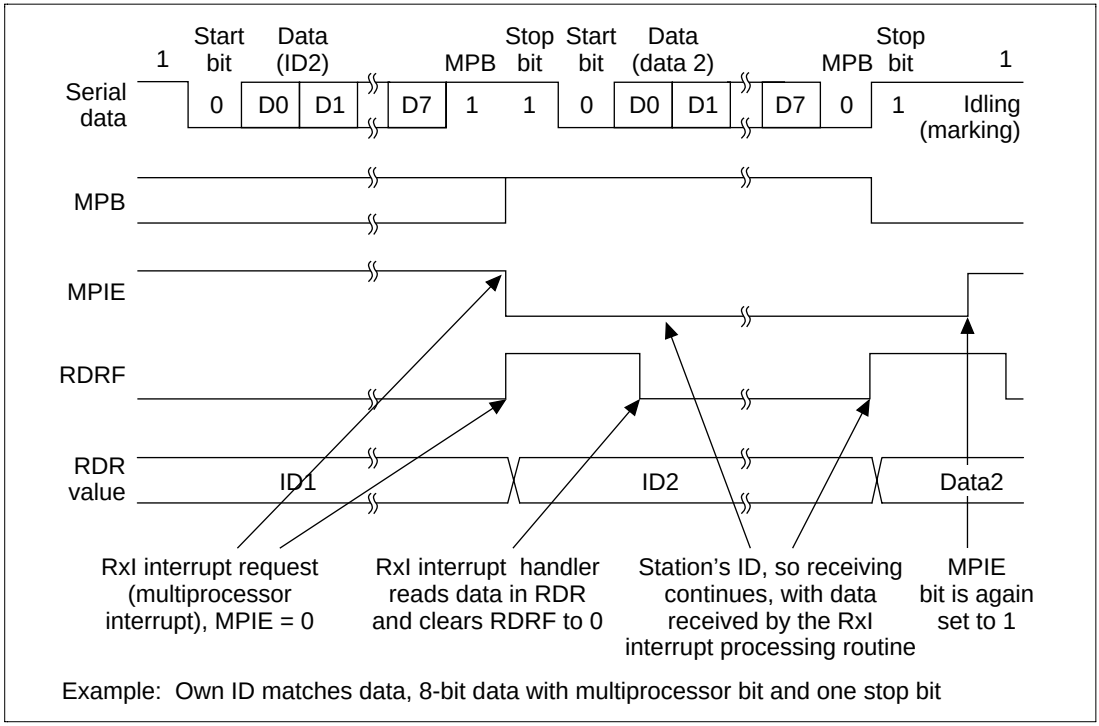


Figure 14.15 Example of SCI Receive Operation (ID Matches)

14.3.4 Clock Synchronous Operation

In the clock synchronous mode, the SCI transmits and receives data in synchronization with clock pulses. This mode is suitable for high-speed serial communication.

The SCI transmitter and receiver are independent, so full duplex communication is possible while sharing the same clock. The transmitter and receiver are also double buffered, so continuous transmitting or receiving is possible by reading or writing data while transmitting or receiving is in progress.

Figure 14.16 shows the general format in clock synchronous serial communication.

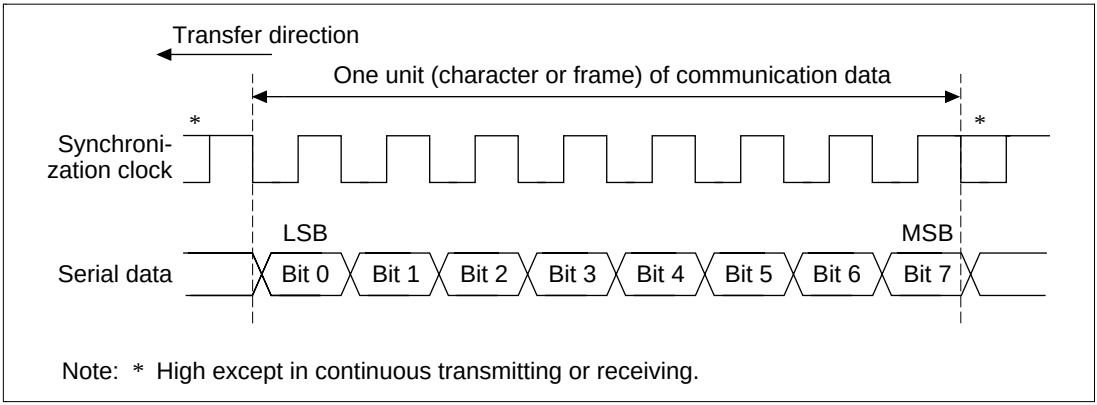


Figure 14.16 Data Format in Clock Synchronous Communication

In clock synchronous serial communication, each data bit is output on the communication line from one falling edge of the serial clock to the next. Data are guaranteed valid at the rising edge of the serial clock. In each character, the serial data bits are transmitted in order from the LSB (first) to the MSB (last). After output of the MSB, the communication line remains in the state of the MSB. In the clock synchronous mode, the SCI transmits or receives data by synchronizing with the falling edge of the synchronization clock.

Communication Format: The data length is fixed at eight bits. No parity bit or multiprocessor bit can be added.

Clock: An internal clock generated by the on-chip baud rate generator or an external clock input from the SCK pin can be selected as the SCI transmit/receive clock. The clock source is selected by the C/$\overline{A}$ bit in the serial mode register (SMR) and bits CKE1 and CKE0 in the serial control register (SCR). See table 14.9.

When the SCI operates on an internal clock, it outputs the clock signal at the SCK pin. Eight clock pulses are output per transmitted or received character. When the SCI is not transmitting or receiving, the clock signal remains in the high state.

Note: An overrun error occurs only during the receive operation, and the sync clock is output until the RE bit is cleared to 0. When you want to perform a receive operation in one-character units, select external clock for the clock source.

SCI Initialization (Clock Synchronous Mode): Before transmitting or receiving, software must clear the TE and RE bits to 0 in the serial control register (SCR), then initialize the SCI as follows.

When changing the mode or communication format, always clear the TE and RE bits to 0 before following the procedure given below. Clearing TE to 0 sets TDRE to 1 and initializes the transmit shift register (TSR). Clearing RE to 0, however, does not initialize the RDRF, PER, FER, and ORER flags and receive data register (RDR), which retain their previous contents.

Figure 14.17 is a sample flowchart for initializing the SCI.

1. Select the clock source in the serial control register (SCR). Leave RIE, TIE, TEIE, MPIE, TE, and RE cleared to 0.
2. Select the communication format in the serial mode register (SMR).
3. Write the value corresponding to the bit rate in the bit rate register (BRR) unless an external clock is used.
4. Wait for at least the interval required to transmit or receive one bit, then set TE or RE in the serial control register (SCR) to 1. Also set RIE, TIE, TEIE, and MPIE. The TxD, RxD pins becomes usable in response to the PFC corresponding bits and the TE, RE bit settings.

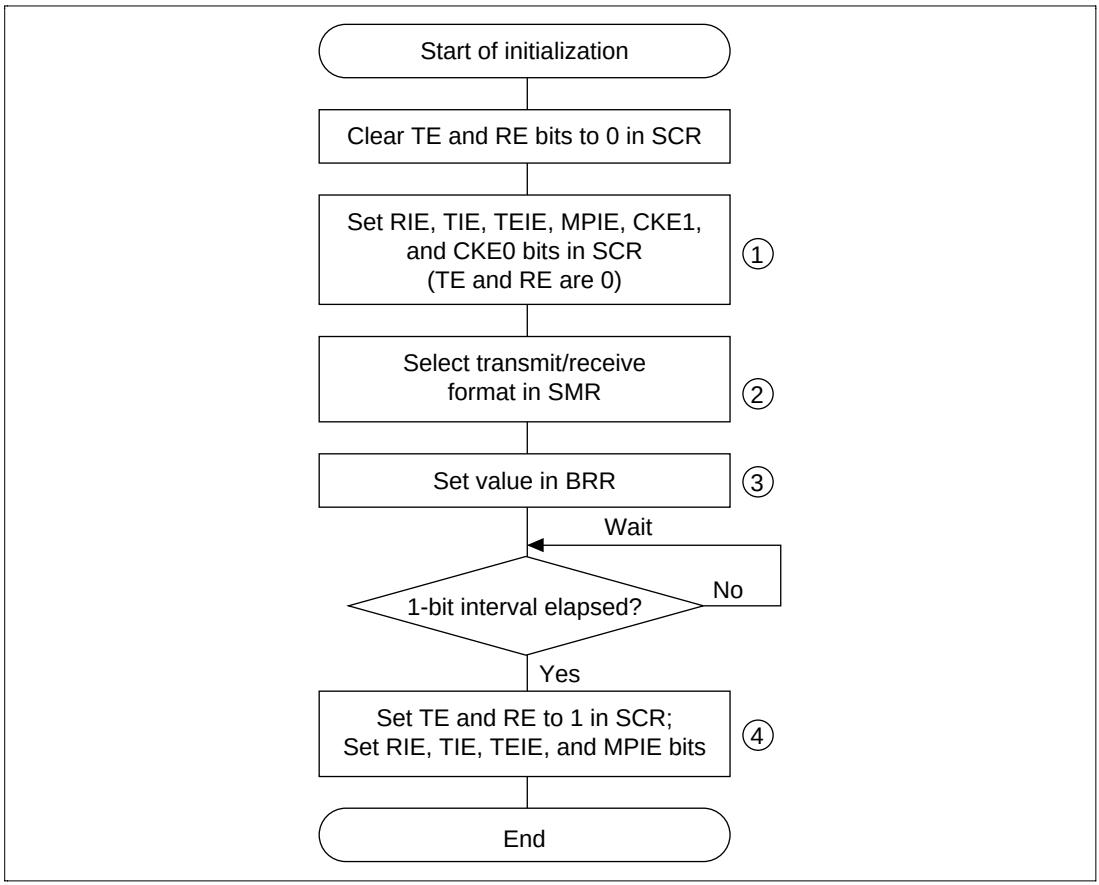


Figure 14.17 Sample Flowchart for SCI Initialization

Transmitting Serial Data (Synchronous Mode): Figure 14.18 shows a sample flowchart for transmitting serial data and indicates the procedure to follow.

1. SCI initialization: Set the TxD pin function with the PFC.
2. SCI status check and transmit data write: Read SSR, check that the TDRE flag is 1, then write transmit data in TDR and clear the TDRE flag to 0.
3. To continue transmitting serial data: After checking that the TDRE flag is 1, indicating that data can be written, write data in TDR, then clear the TDRE flag to 0. When the DMAC or DTC is activated by a transmit-data-empty interrupt request (TxI) to write data in TDR, the TDRE flag is checked and cleared automatically.

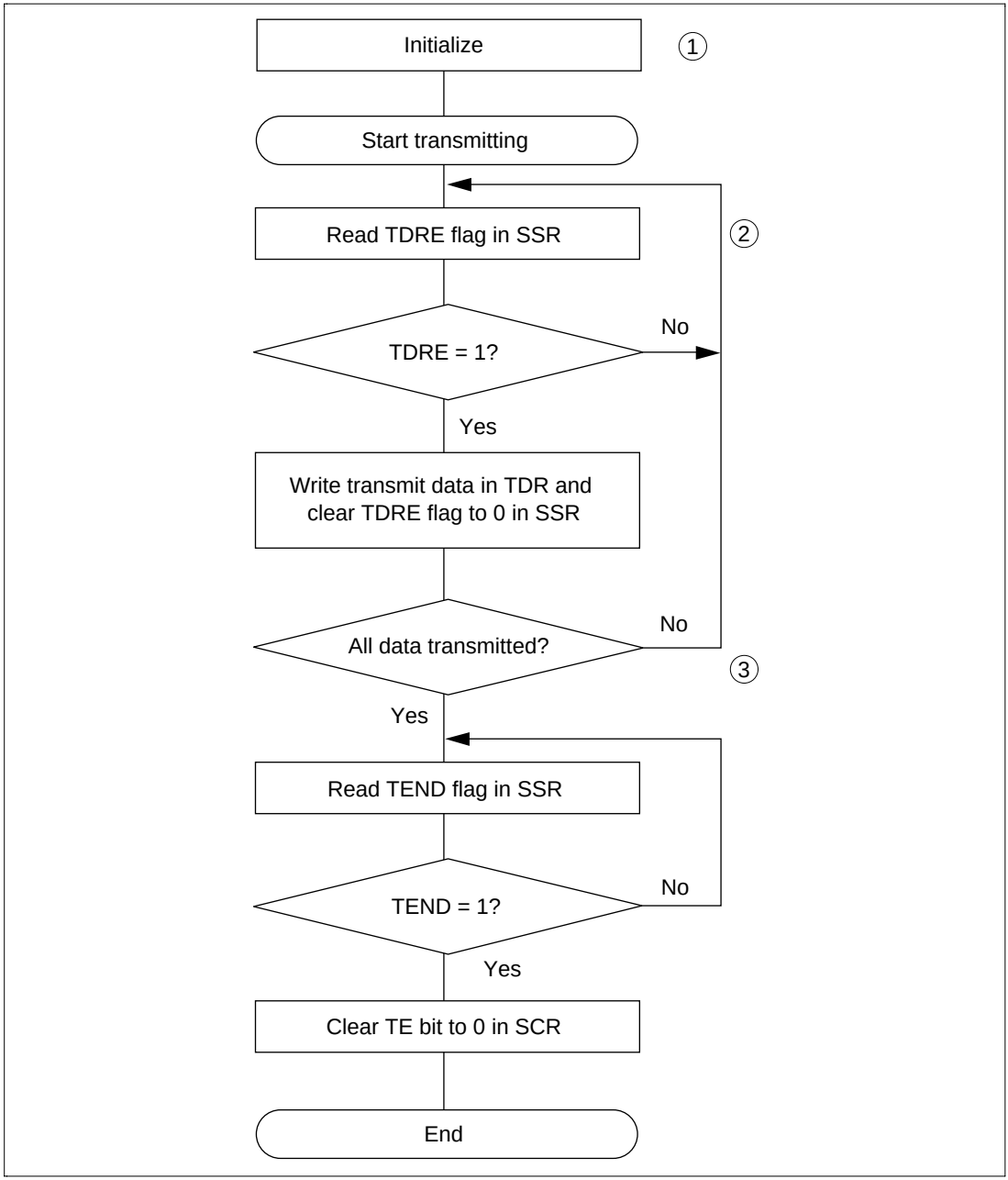


Figure 14.18 Sample Flowchart for Serial Transmitting

Figure 14.19 shows an example of SCI transmit operation.

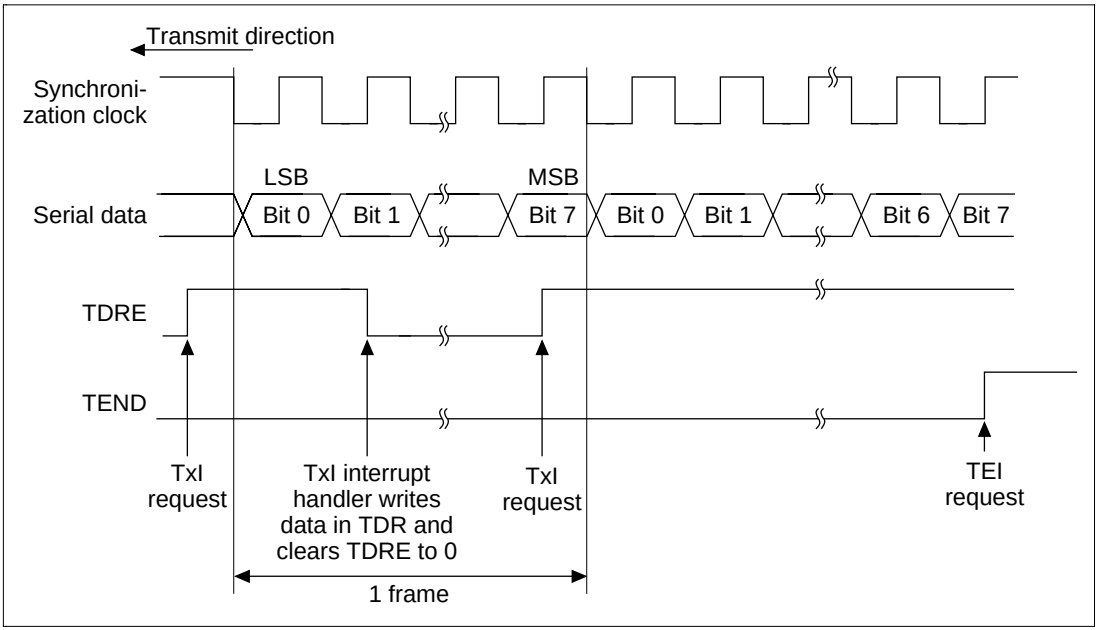


Figure 14.19 Example of SCI Transmit Operation

SCI serial transmission operates as follows.

1. The SCI monitors the TDRE bit in the SSR. When TDRE is cleared to 0 the SCI recognizes that the transmit data register (TDR) contains new data and loads this data from the TDR into the transmit shift register (TSR).
2. After loading the data from the TDR into the TSR, the SCI sets the TDRE bit to 1 and starts transmitting. If the transmit-data-empty interrupt enable bit (TIE) in the SCR is set to 1, the SCI requests a transmit-data-empty interrupt (TxI) at this time.

   If clock output mode is selected, the SCI outputs eight synchronous clock pulses. If an external clock source is selected, the SCI outputs data in synchronization with the input clock. Data are output from the TxD pin in order from the LSB (bit 0) to the MSB (bit 7).
3. The SCI checks the TDRE bit when it outputs the MSB (bit 7). If TDRE is 0, the SCI loads data from the TDR into the TSR, then begins serial transmission of the next frame. If TDRE is 1, the SCI sets the TEND bit in the SSR to 1, transmits the MSB, then holds the transmit data pin (TxD) in the MSB state. If the transmit-end interrupt enable bit (TEIE) in the SCR is set to 1, a transmit-end interrupt (TEI) is requested at this time.
4. After the end of serial transmission, the SCK pin is held in the high state.

Receiving Serial Data (Clock Synchronous Mode): Figures 14.20 and 14.21 shows a sample flowchart for receiving serial data. When switching from the asynchronous mode to the clock synchronous mode, make sure that ORER, PER, and FER are cleared to 0. If PER or FER is set to 1, the RDRF bit will not be set and both transmitting and receiving will be disabled.

The procedure for receiving serial data is listed below:

1. SCI initialization: Set the RxD pin using the PFC.
2. Receive error handling: If a receive error occurs, read the ORER bit in SSR to identify the error. After executing the necessary error handling, clear ORER to 0. Transmitting/receiving cannot resume if ORER remains set to 1.
3. SCI status check and receive data read: Read the serial status register (SSR), check that RDRF is set to 1, then read receive data from the receive data register (RDR) and clear RDRF to 0. The RxI interrupt can also be used to determine if the RDRF bit has changed from 0 to 1.
4. Continue receiving serial data: Read RDR, and clear RDRF to 0 before the frame MSB (bit 7) of the current frame is received. If the DMAC or the DTC is started by a receive-data-full interrupt (RxI) to read RDR, the RDRF bit is cleared automatically so this step is unnecessary.

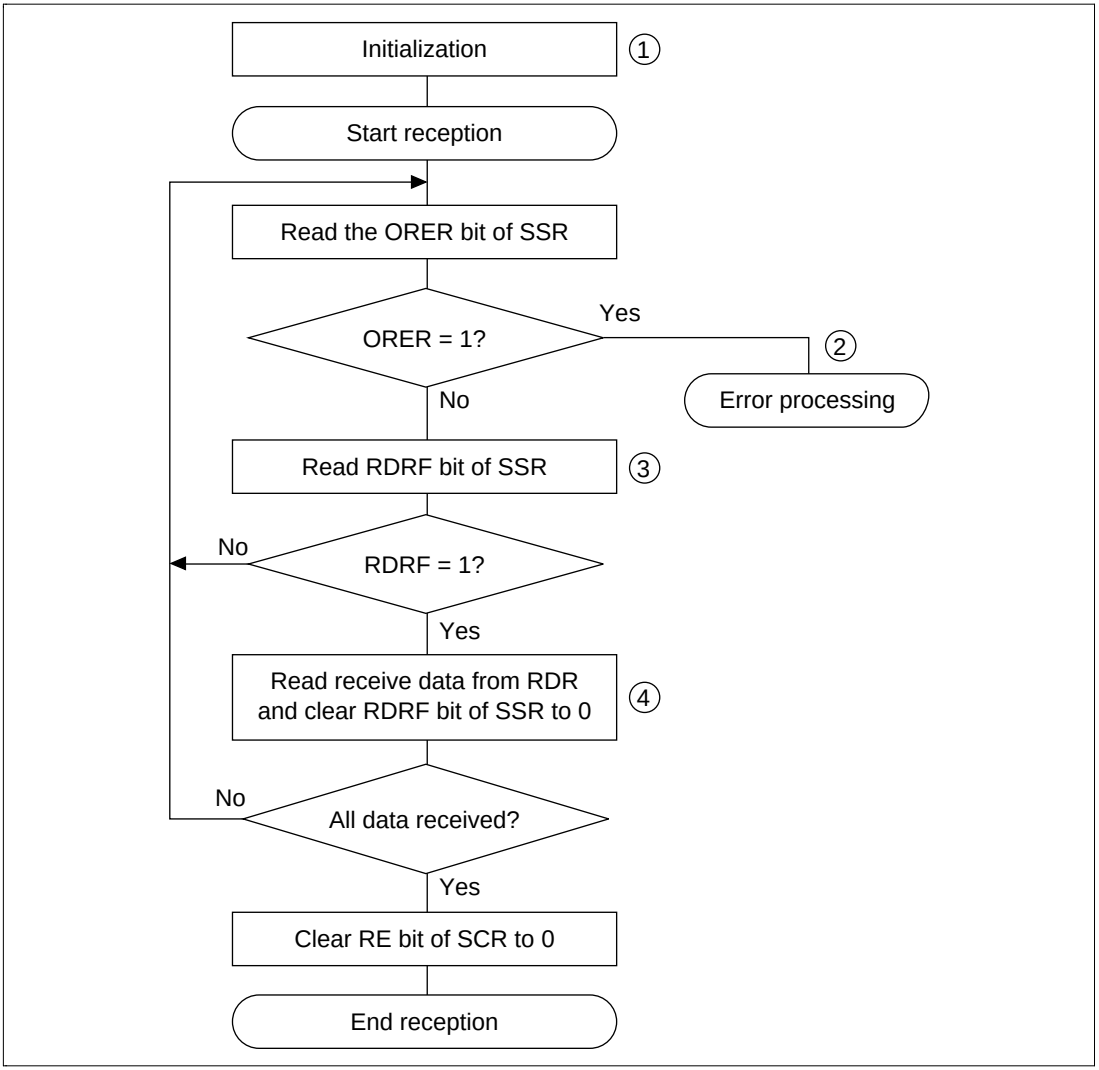


Figure 14.20 Sample Flowchart for Serial Receiving (1)

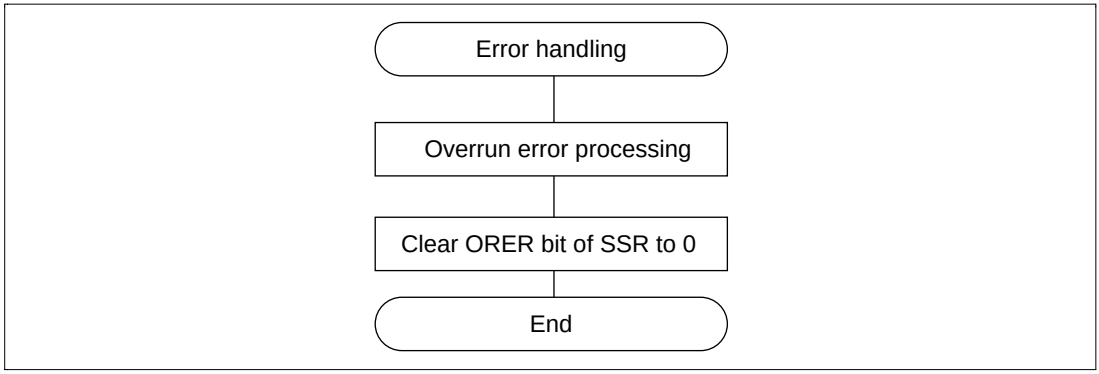


Figure 14.21 Sample Flowchart for Serial Receiving (2)

Figure 14.22 shows an example of the SCI receive operation.

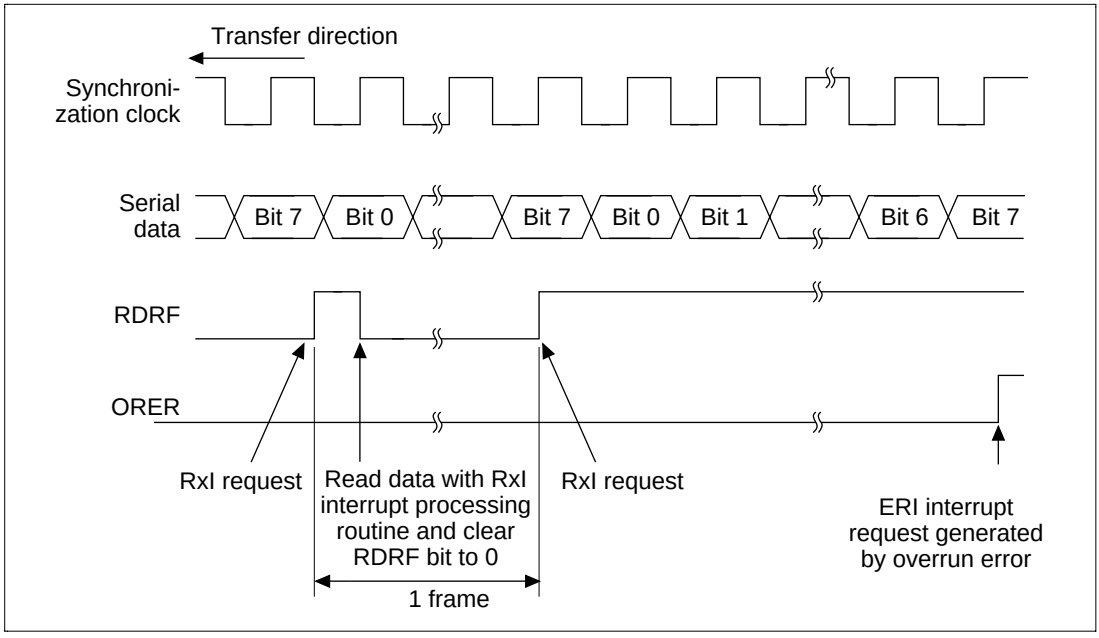


Figure 14.22 Example of SCI Receive Operation

In receiving, the SCI operates as follows:

1. The SCI synchronizes with serial clock input or output and initializes internally.
2. Receive data is shifted into the RSR in order from the LSB to the MSB. After receiving the data, the SCI checks that RDRF is 0 so that receive data can be loaded from the RSR into the RDR. If this check passes, the SCI sets RDRF to 1 and stores the received data in the RDR. If the check does not pass (receive error), the SCI operates as indicated in table 14.11 and no further transmission or reception is possible. If the error flag is set to 1, the RDRF bit is not set

to 1 during reception, even if the RDRF bit is 0 cleared. When restarting reception, be sure to clear the error flag.

3. After setting RDRF to 1, if the receive-data-full interrupt enable bit (RIE) is set to 1 in the SCR, the SCI requests a receive-data-full interrupt (RxI). If the ORER bit is set to 1 and the receive-data-full interrupt enable bit (RIE) in the SCR is also set to 1, the SCI requests a receive-error interrupt (ERI).

Transmitting and Receiving Serial Data Simultaneously (Clock Synchronous Mode): Figure 14.23 shows a sample flowchart for transmitting and receiving serial data simultaneously. The procedure is as follows (the steps correspond to the numbers in the flowchart):

1. SCI initialization: Set the TxD and RxD pins using the PFC.
2. SCI status check and transmit data write: Read the serial status register (SSR), check that the TDRE bit is 1, then write transmit data in the transmit data register (TDR) and clear TDRE to 0. The TxI interrupt can also be used to determine if the TDRE bit has changed from 0 to 1.
3. Receive error handling: If a receive error occurs, read the ORER bit in SSR to identify the error. After executing the necessary error processing, clear ORER to 0. Transmitting/receiving cannot resume if ORER remains set to 1.
4. SCI status check and receive data read: Read the serial status register (SSR), check that RDRF is set to 1, then read receive data from the receive data register (RDR) and clear RDRF to 0. The RxI interrupt can also be used to determine if the RDRF bit has changed from 0 to 1.
5. Continue transmitting and receiving serial data: Read the RDRF bit and RDR, and clear RDRF to 0 before the frame MSB (bit 7) of the current frame is received. Also read the TDRE bit to check whether it is safe to write (if it reads 1); if so, write data in TDR, then clear TDRE to 0 before the MSB (bit 7) of the current frame is transmitted. When the DMAC or the DTC is started by a transmit-data-empty interrupt request (TxI) to write data in TDR, the TDRE bit is checked and cleared automatically. When the DMAC or the DTC is started by a receive-data-full interrupt (RxI) to read RDR, the RDRF bit is cleared automatically.

Note: In switching from transmitting or receiving to simultaneous transmitting and receiving, simultaneously clear both TE and RE to 0, then simultaneously set both TE and RE to 1.

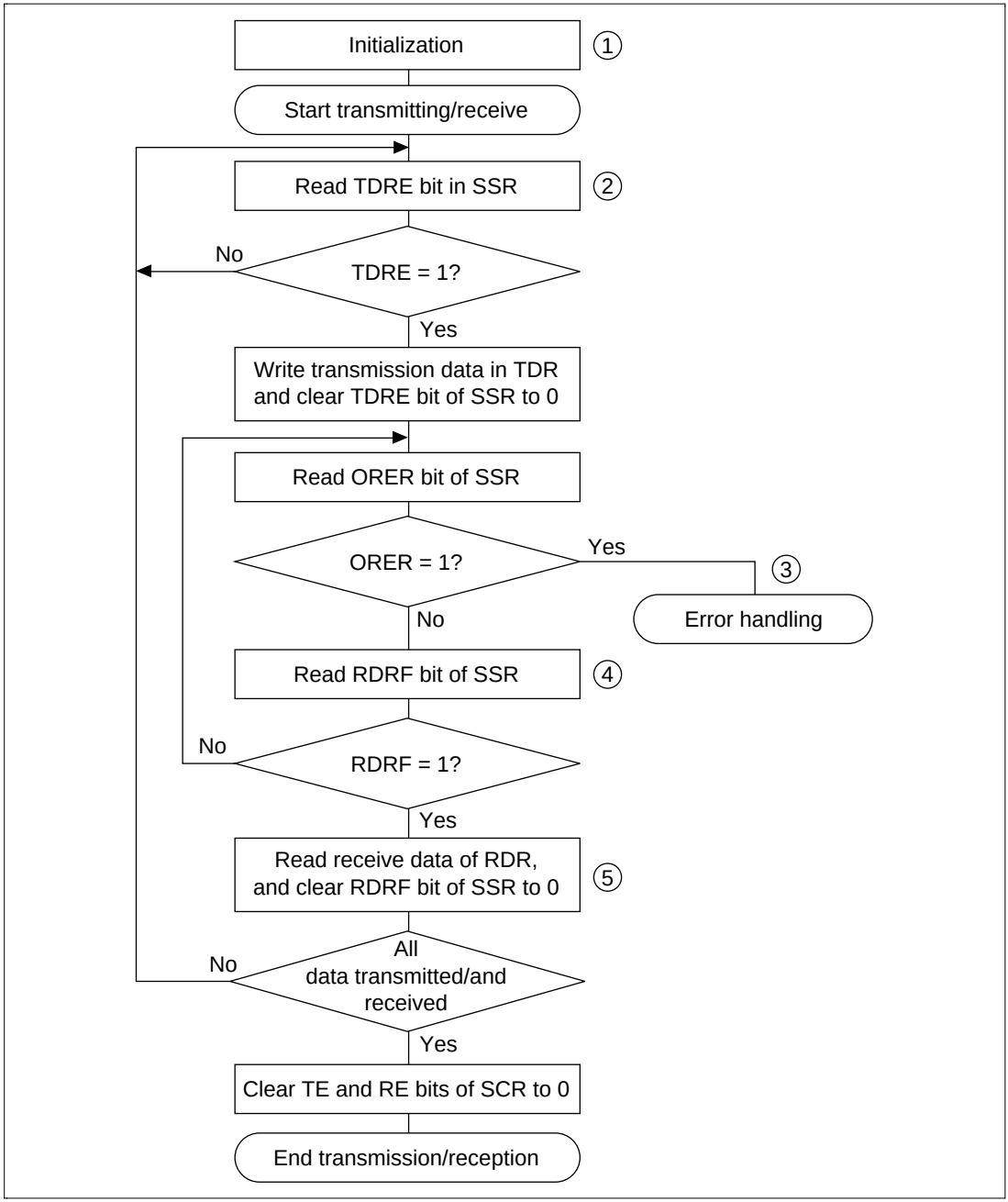


Figure 14.23 Sample Flowchart for Serial Transmission

14.4 SCI Interrupt Sources and the DMAC/DTC

The SCI has four interrupt sources: transmit-end (TEI), receive-error (ERI), receive-data-full (RxI), and transmit-data-empty (TxI). Table 14.12 lists the interrupt sources and indicates their priority. These interrupts can be enabled and disabled by the TIE, RIE, and TEIE bits in the serial control register (SCR). Each interrupt request is sent separately to the interrupt controller.

TxI is requested when the TDRE bit in the SSR is set to 1. TxI can start the direct memory access controller (DMAC) or the data transfer controller (DTC) to transfer data. TDRE is automatically cleared to 0 when the DMAC or the DTC writes data in the transmit data register (TDR).

RxI is requested when the RDRF bit in the SSR is set to 1. RxI can start the DMAC or the DTC to transfer data. RDRF is automatically cleared to 0 when the DMAC or the DTC reads the receive data register (RDR).

ERI is requested when the ORER, PER, or FER bit in the SSR is set to 1. ERI cannot start the DMAC or the DTC.

TEI is requested when the TEND bit in the SSR is set to 1. TEI cannot start the DMAC or the DTC. Where the TxI interrupt indicates that transmit data writing is enabled, the TEI interrupt indicates that the transmit operation is complete.

Table 14.12 SCI Interrupt Sources

Interrupt Source	Description	DMAC/DTC Activation	Priority
ERI	Receive error (ORER, PER, or FER)	No	High
RxI	Receive data full (RDRF)	Yes	
TxI	Transmit data empty (TDRE)	Yes	
TEI	Transmit end (TEND)	No	Low

14.5 Notes on Use

Sections 14.5.1 through 14.5.9 provide information for using the SCI.

14.5.1 TDR Write and TDRE Flags

The TDRE bit in the serial status register (SSR) is a status flag indicating loading of transmit data from TDR into TSR. The SCI sets TDRE to 1 when it transfers data from TDR to TSR. Data can be written to TDR regardless of the TDRE bit status. If new data is written in TDR when TDRE is 0, however, the old data stored in TDR will be lost because the data has not yet been transferred to the TSR. Before writing transmit data to the TDR, be sure to check that TDRE is set to 1.

14.5.2 Simultaneous Multiple Receive Errors

Table 14.13 indicates the state of the SSR status flags when multiple receive errors occur simultaneously. When an overrun error occurs, the RSR contents cannot be transferred to the RDR, so receive data is lost.

Table 14.13 SSR Status Flags and Transfer of Receive Data

	SSR Status Flags				Receive Data Transfer
Receive Error Status	**RDRF**	**ORER**	**FER**	**PER**	**RSR → RDR**
Overrun error	1	1	0	0	X
Framing error	0	0	1	0	O
Parity error	0	0	0	1	O
Overrun error + framing error	1	1	1	0	X
Overrun error + parity error	1	1	0	1	X
Framing error + parity error	0	0	1	1	O
Overrun error + framing error + parity error	1	1	1	1	X

Notes: O = Receive data is transferred from RSR to RDR.
X = Receive data is not transferred from RSR to RDR.

14.5.3 Break Detection and Processing

Break signals can be detected by reading the RxD pin directly when a framing error (FER) is detected. In the break state, the input from the RxD pin consists of all 0s, so FER is set and the parity error flag (PER) may also be set. In the break state, the SCI receiver continues to operate, so if the FER bit is cleared to 0, it will be set to 1 again.

14.5.4 Sending a Break Signal

The TxD pin becomes a general I/O pin with the I/O direction and level determined by the I/O port data register (DR) and pin function controller (PFC) control register (CR). These conditions allow break signals to be sent. The DR value is substituted for the marking status until the PFC is set. Consequently, the output port is set to initially output a 1. To send a break in serial transmission, first clear the DR to 0, then establish the TxD pin as an output port using the PFC. When TE is cleared to 0, the transmission section is initialized regardless of the present transmission status.

14.5.5 Receive Error Flags and Transmitter Operation (Clock Synchronous Mode Only)

When a receive error flag (ORER, PER, or FER) is set to 1, the SCI will not start transmitting even if TDRE is set to 1. Be sure to clear the receive error flags to 0 before starting to transmit. Note that clearing RE to 0 does not clear the receive error flags.

14.5.6 Receive Data Sampling Timing and Receive Margin in the Asynchronous Mode

In the asynchronous mode, the SCI operates on a base clock of 16 times the bit rate frequency. In receiving, the SCI synchronizes internally with the falling edge of the start bit, which it samples on the base clock. Receive data is latched on the rising edge of the eighth base clock pulse (figure 14.24).

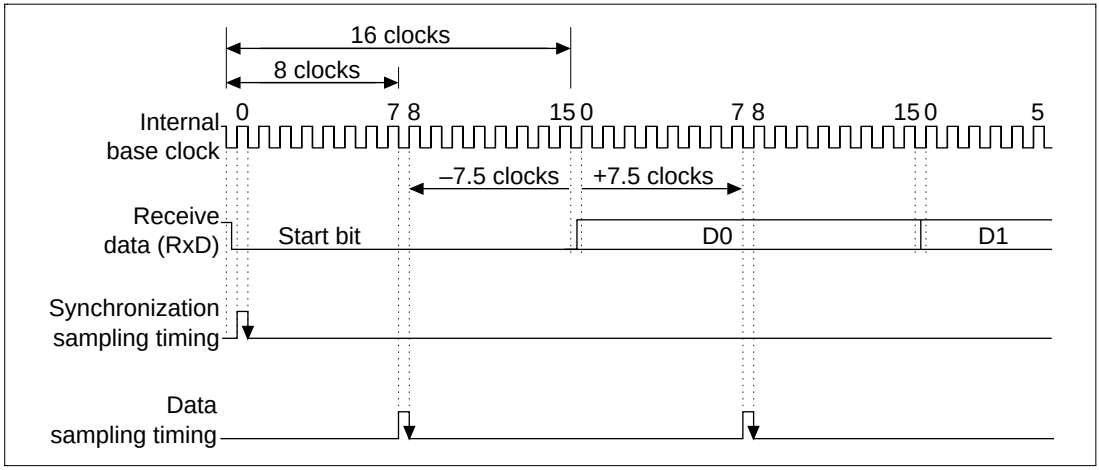


Figure 14.24 Receive Data Sampling Timing in Asynchronous Mode

The receive margin in the asynchronous mode can therefore be expressed as:

$$M = \left| \left(0.5 - \frac{1}{2N}\right) - (L - 0.5)F - \frac{|D - 0.5|}{N}(1 + F) \right| \times 100\%$$

M : Receive margin (%)
N : Ratio of clock frequency to bit rate (N = 16)
D : Clock duty cycle (D = 0–1.0)
L : Frame length (L = 9–12)
F : Absolute deviation of clock frequency

From the equation above, if F = 0 and D = 0.5 the receive margin is 46.875%:

D = 0.5, F = 0
M = $(0.5 - 1/(2 \times 16)) \times 100\%$
= 46.875%

This is a theoretical value. A reasonable margin to allow in system designs is 20–30%.

14.5.7 Constraints on DMAC/DTC Use

- When using an external clock source for the synchronization clock, update the TDR with the DMAC or the DTC, and then after five system clocks or more elapse, input a transmit clock. If a transmit clock is input in the first four system clocks after the TDR is written, an error may occur (figure 14.25).
- Before reading the receive data register (RDR) with the DMAC/DTC, select the receive-data-full interrupt of the SCI as a start-up source.

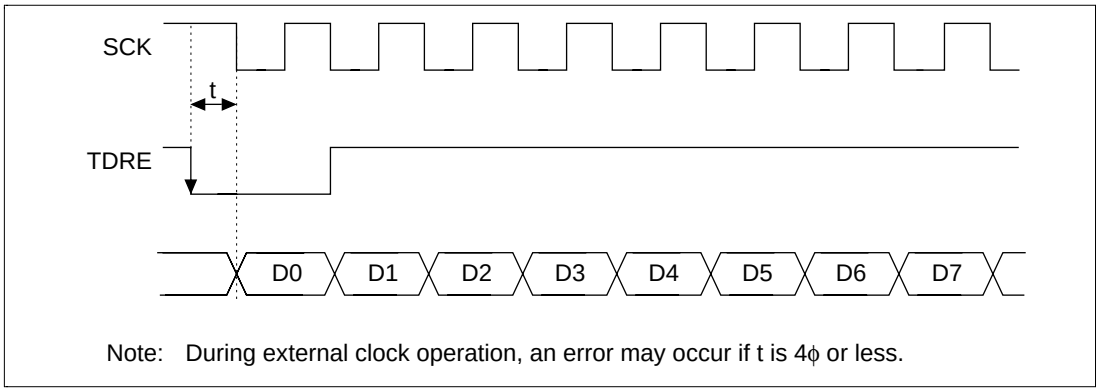


Figure 14.25 Example of Clock Synchronous Transmission with DMAC

14.5.8 Cautions for Clock Synchronous External Clock Mode

- Set TE = RE = 1 only when the external clock SCK is 1.
- Do not set TE = RE = 1 until at least four clocks after the external clock SCK has changed from 0 to 1.
- When receiving, RDRF is 1 when RE is set to zero 2.5–3.5 clocks after the rising edge of the RxD D7 bit SCK input, but it cannot be copied to RDR.

14.5.9 Caution for Clock Synchronous Internal Clock Mode

When receiving, RDRF is 1 when RE is set to zero 1.5 clocks after the rising edge of the RxD D7 bit SCK output, but it cannot be copied to RDR.

Section 15 High Speed A/D Converter (Excluding A Mask)

15.1 Overview

The high speed A/D converter has 10-bit resolution, and can select from a maximum of eight channels of analog inputs.

15.1.1 Features

The high speed A/D converter has the following features:

- 10-bit resolution
- Eight input channels
- Analog conversion voltage range setting is selectable
  - Using the reference voltage pin (AVref) as an analog standard voltage (Vref), conversion of analog input from 0 to Vref (only with SH7043).
- High-speed conversion
  - Minimum conversion time: 2.9 µs per channel (for 28-MHz operation)
  - 1.4 µs per channel during continuous conversion
- Multiple conversion modes
  - Select mode/group mode
  - Single mode/scan mode
  - Buffered operation possible
  - 2 channel simultaneous sampling possible
- Three types of conversion start
  - Software, timer conversion start trigger (MTU), or $\overline{\text{ADTRG}}$ pin can be selected.
- Eight data registers
  - Conversion results stored in 16-bit data registers corresponding to each channel.
- Sample and hold function
- A/D conversion end interrupt generation
  - An A/D conversion end interrupt (ADI) request can be generated on completion of A/D conversions

15.1.2 Block Diagram

Figure 15.1 is the block diagram of the high speed A/D converter.

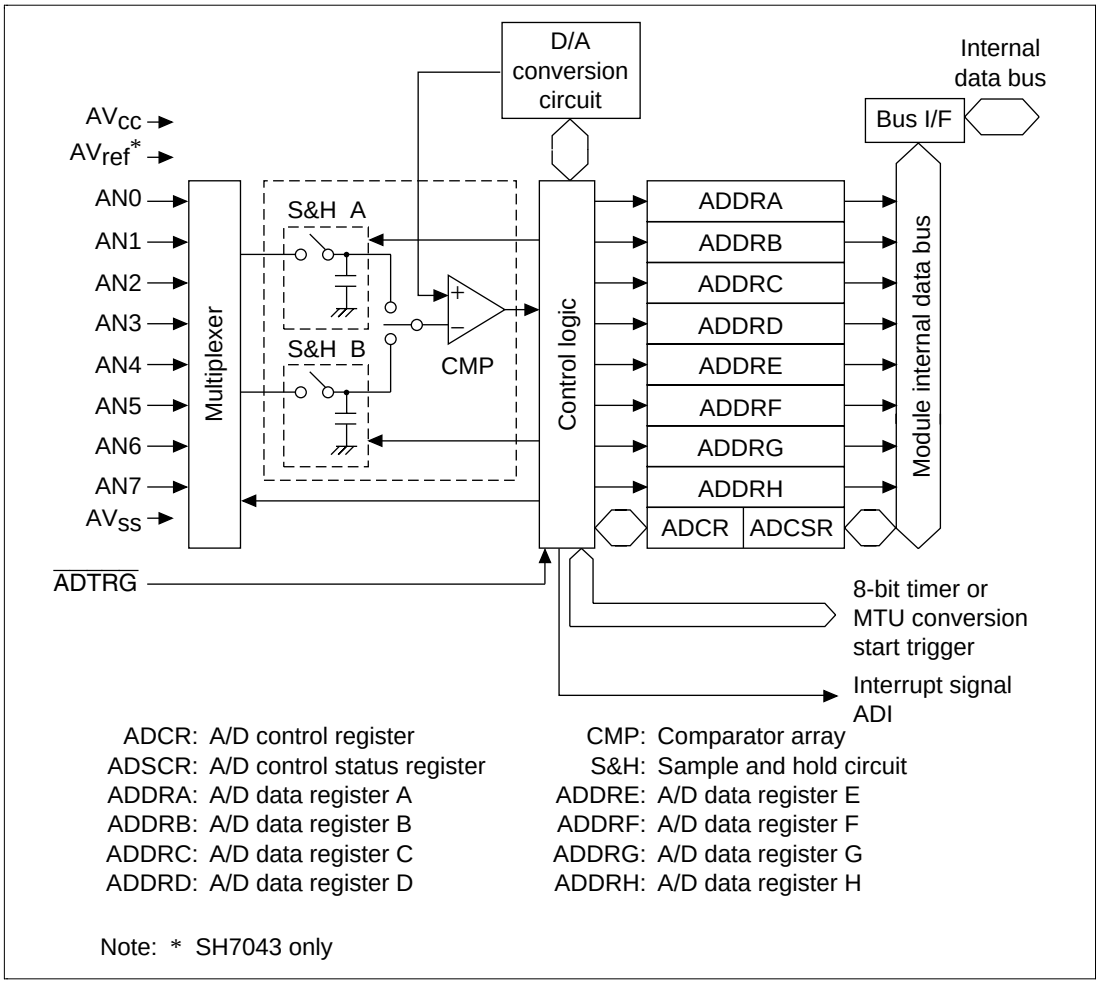


Figure 15.1 High Speed A/D Converter Block Diagram

15.1.3 Pin Configuration

Table 15.1 shows the input pins used by the high speed A/D converter.

The AV_{cc} and AV_{ss} pins are for the A/D converter internal analog section power supply. The AV_{ref} pin is for the A/D conversion standard voltage.

Table 15.1 Pin Configuration

Pin	Abbreviation	I/O	Function
Analog supply	AV_{CC}	I	Analog section power supply
Analog ground	AV_{SS}	I	Analog section ground and A/D conversion reference voltage
Reference voltage	AV_{ref}	I	A/D conversion standard voltage (SH7043 only)
Analog input 0	AN0	I	Analog input channel 0
Analog input 1	AN1	I	Analog input channel 1
Analog input 2	AN2	I	Analog input channel 2
Analog input 3	AN3	I	Analog input channel 3
Analog input 4	AN4	I	Analog input channel 4
Analog input 5	AN5	I	Analog input channel 5
Analog input 6	AN6	I	Analog input channel 6
Analog input 7	AN7	I	Analog input channel 7
A/D external trigger input	$\overline{\text{ADTRG}}$	I	External trigger for A/D conversion start

15.1.4 Register Configuration

Table 15.2 shows the configuration of the high speed A/D converter registers.

Table 15.2 Register Configuration

Name	Abbreviation	R/W	Initial Value	Address	Access Size
A/D data register A	ADDRA	R	H'0000	H'FFFF83F0	8,16
A/D data register B	ADDRB	R	H'0000	H'FFFF83F2	
A/D data register C	ADDRC	R	H'0000	H'FFFF83F4	
A/D data register D	ADDRD	R	H'0000	H'FFFF83F6	
A/D data register E	ADDRE	R	H'0000	H'FFFF83F8	
A/D data register F	ADDRF	R	H'0000	H'FFFF83FA	
A/D data register G	ADDRG	R	H'0000	H'FFFF83FC	
A/D data register H	ADDRH	R	H'0000	H'FFFF83FE	
A/D control/status register	ADCSR	R/(W)*	H'00	H'FFFF83E0	
A/D control register	ADCR	R/W	H'00	H'FFFF83E1	

Note: * Only 0 can be written to bit 7 to clear the flag.

15.2 Register Descriptions

15.2.1 A/D Data Registers A–H (ADDRA–ADDRH)

The ADDR are 16-bit read only registers for storing A/D conversion results. There are eight of these registers, ADDRA through ADDRH.

The A/D converted data is 10-bit data which is sent to the ADDR for the corresponding converted channel for storage. The lower 8 bits of the A/D converted data are transferred to and stored in the lower byte (bits 7–0) of the ADDR, and the upper 2 bits are stored into the upper byte (bits 9, 8). Bits 15–10 always read as 0. Data reads can be either byte or word. The upper 8 bits of the converted data are transferred upon byte data reads. Additionally, buffered operation is possible by combining ADDRA–ADDRD.

Table 15.3 shows the correspondence between the analog input channels and the ADDR.

The ADDR are initialized to H'0000 by power-on reset or in standby mode. Manual reset does not initialize ADDR.

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	AD9	AD8
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Table 15.3 Analog Input Channel and ADDR Correspondence

Analog Input Channel	A/D Data Register
AN0	ADDRA*
AN1	ADDRB*
AN2	ADDRC*
AN3	ADDRD*
AN4	ADDRE
AN5	ADDRF
AN6	ADDRG
AN7	ADDRH

Note: * Except during buffer operation

15.2.2 A/D Control/Status Register (ADCSR)

The ADCSR is an 8-bit read/write register used for A/D conversion operation control and to indicate status.

The ADCSR is initialized to H'00 by power-on reset or in standby mode. Manual reset does not initialize ADCSR.

Bit:	7	6	5	4	3	2	1	0
	ADF	ADIE	ADST	CKS	GRP	CH2	CH1	CH0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: * The only value that can be written is a 0 to clear the flag.

- Bit 7—A/D End Flag (ADF): This status flag indicates that A/D conversion has ended.

Bit 7: ADF	Description
0	Clear conditions (initial value) • With ADF = 1, by reading the ADF flag then writing 0 in ADF • When the DTC or DMAC are activated by an ADI interrupt
1	Set conditions • Single mode: When A/D conversion ends after conversion for all designated channels (during buffer operation, this is not set until operation of the specified buffer has ended) • Scan mode: After one round of A/D conversion for all specified channels

- Bit 6—A/D Interrupt Enable (ADIE): Enables or disables interrupt requests (ADI) after A/D conversion ends. Set the ADIE bit while conversion is suspended.

Bit 6: ADIE	Description
0	Disables interrupt requests (ADI) after A/D conversion ends (initial value)
1	Enables interrupt requests (ADI) after A/D conversion ends

- Bit 5—A/D Start (ADST): Selects start or stop for A/D conversion. A 1 is maintained during A/D conversions.

  The ADST bit can be set to 1 by software, timer conversion start triggers, or an A/D external trigger input pin ($\overline{\text{ADTRG}}$).

Bit 5: ADST	Description
0	A/D conversion halted (initial value)
1	Single mode: Start A/D conversion. Automatically cleared to 0 after conversion for the designated channel ends. Scan mode: Start A/D conversion. Continuous conversion until 0 cleared by software.

- Bit 4—Clock Select (CKS): Sets the A/D conversion time. Set, according to the operating frequency, to give a conversion time of at least 2 µs (5 V version) or 4 µs (3.3 V version).

  Make conversion time changes only while conversion is halted.

Bit 4: CKS	Description
0	Conversion time = 40 states (A/D converter standard clock = ϕ/2) (initial value)
1	Conversion time = 80 states (when ϕ/4 is selected)

- Bit 3—Group Mode (GRP): Designates either select mode or group mode for the A/D conversion channel selection.

  Set the GRP bit only while conversion is halted.

Bit 3: GRP	Description
0	Select mode (initial value)
1	Group mode

- Bits 2–0—Channel Select 2–0 (CH2–CH0): These bits, along with the GRP bit, select the analog input channel.

  Set the input channel only while conversion is halted.

			Description	
Bit 2: CH2	**Bit 1: CH1**	**Bit 0: CH0**	**Select Mode (GRP = 0)**	**Group Mode (GRP = 1)**
0	0	0	AN0 (initial value)	AN0
0	0	1	AN1	AN0–AN1
0	1	0	AN2	AN0–AN2
0	1	1	AN3	AN0–AN3
1	0	0	AN4	AN0–AN4
1	0	1	AN5	AN0–AN5
1	1	0	AN6	AN0–AN6
1	1	1	AN7	AN0–AN7

15.2.3 A/D Control Register (ADCR)

The ADCR is an 8-bit read/write register used for A/D conversion operation control. The ADCR is initialized to H'00 by power-on reset or in standby mode. Manual reset does not initialize.

Bit:	7	6	5	4	3	2	1	0
	—	PWR	TRGS1	TRGS0	SCAN	DSMP	BUFE1	BUFE0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- Bit 7—Reserved: This bit always reads as 0. The write value should always be 0.
- Bit 6—Power (PWR): Designates the conversion start mode for the high speed A/D converter. Setting the PWR bit to 1 sets high speed start mode, and a 0 sets to low power conversion mode. See section 15.4.7, Conversion Start Modes, for details on the conversion start operation.

  Set the PWR bit only while conversion is halted.

Bit 6: PWR	Description
0	Low power conversion mode (initial value)
1	High speed start mode

- Bits 5 and 4—Timer Trigger Select 1, 0 (TRGS1, TRGS0): These bits enable or prohibit A/D conversion starts by trigger signals.

  Set the TRGS1, TRGS0 bits only while conversion is halted.

Bit 5: TRGS1	Bit 4: TRGS0	Description
0	0	Enable A/D conversion start by software (initial value)
0	1	Enables A/D conversion start by MTU conversion start trigger
1	0	Reserved
1	1	Enables A/D conversion start by external trigger pin ($\overline{\text{ADTRG}}$)

- Bit 3—Scan Mode (SCAN): Selects either single mode or scan mode for the A/D conversion operation mode. See section 15.4, Operation, for details on single mode and scan mode operation.

  Set the SCAN bit only while conversion is halted.

Bit 3: SCAN	Description
0	Single mode (initial value)
1	Scan mode

- Bit 2—Simultaneous Sampling (DSMP): Enables or disables the simultaneous sampling of two channels. See section 15.4.6, Simultaneous Sampling Operation, for details on simultaneous sampling.

  Set the DSMP bit only while conversion is halted.

Bit 2: DSMP	Description
0	Normal sampling operation (initial value)
1	Simultaneous sampling operation

- Bits 1–0—Buffer Enable 1, 0 (BUFE1, BUFE0): These bits select whether to use the ADDRB–ADDRD as buffer registers.

  Set the BUFE1 and BUFE0 bits only while conversion is halted.

Bit 1: BUFE1	Bit 0: BUFE0	Description
0	0	Normal operation (initial value)
0	1	ADDRA and ADDRB buffer operation: conversion result → ADDRA → ADDRB (ADDRB is the buffer register)
1	0	ADDRA and ADDRC, also ADDRB and ADDRD buffer operation: conversion result 1 → ADDRA → ADDRC, conversion result 2 → ADDRB → ADDRD (ADDRC and ADDRD are buffer registers)
1	1	ADDRA–ADDRD buffer operation: conversion result → ADDRA → ADDRB → ADDRC → ADDRD (ADDRB–ADDRD are buffer registers)

15.3 Bus Master Interface

The ADDRA–ADDRH are 16-bit registers with a 16-bit width data bus to the bus master. The bus master can read from ADDRA–ADDRH in either word or byte units.

When an ADDR is read in word units, the ADDR contents are transferred to the bus master 16 bits at a time. In byte unit reads, the contents of the most significant eight bits (AD9–AD2) of the converted data (AD9–AD0) are transferred to the bus master.

Figures 15.2 and 15.3 shows an example of the ADDR read operation.

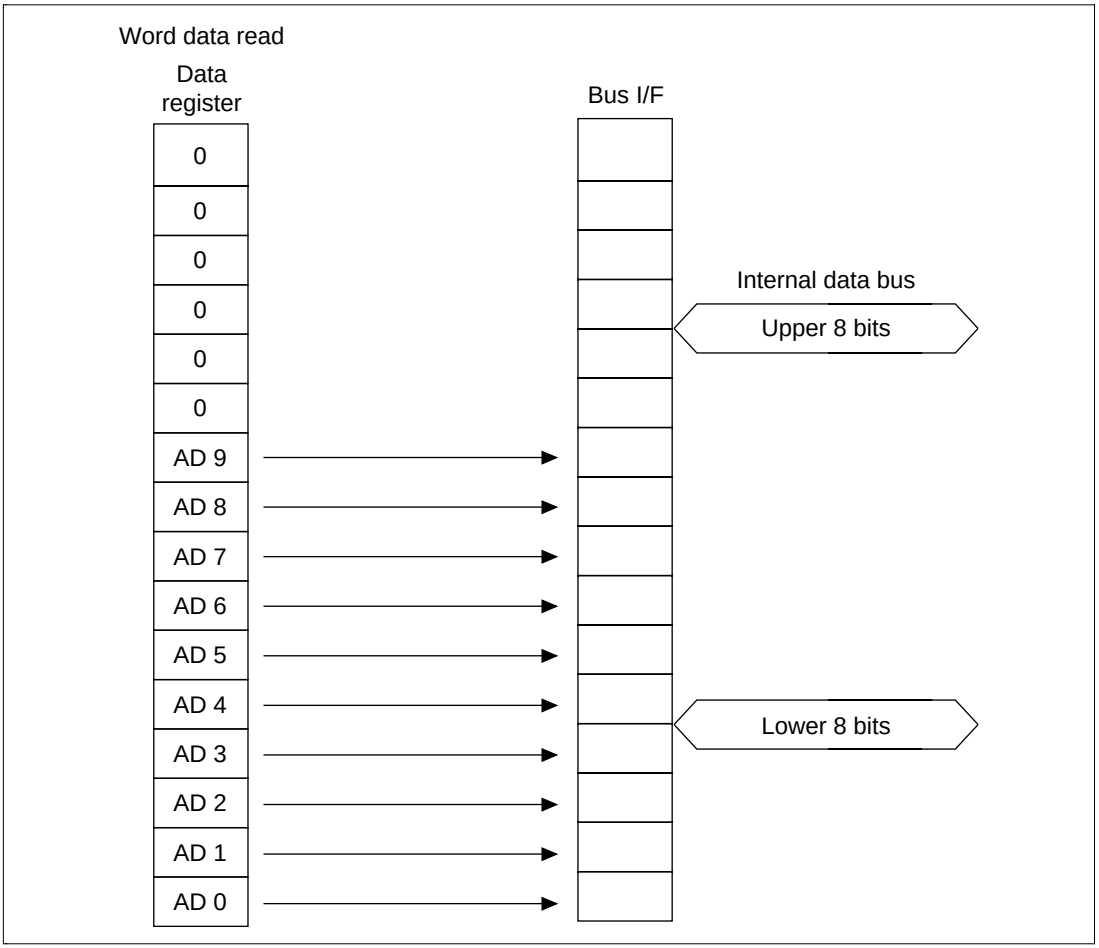


Figure 15.2 ADDR Read Operation (1)

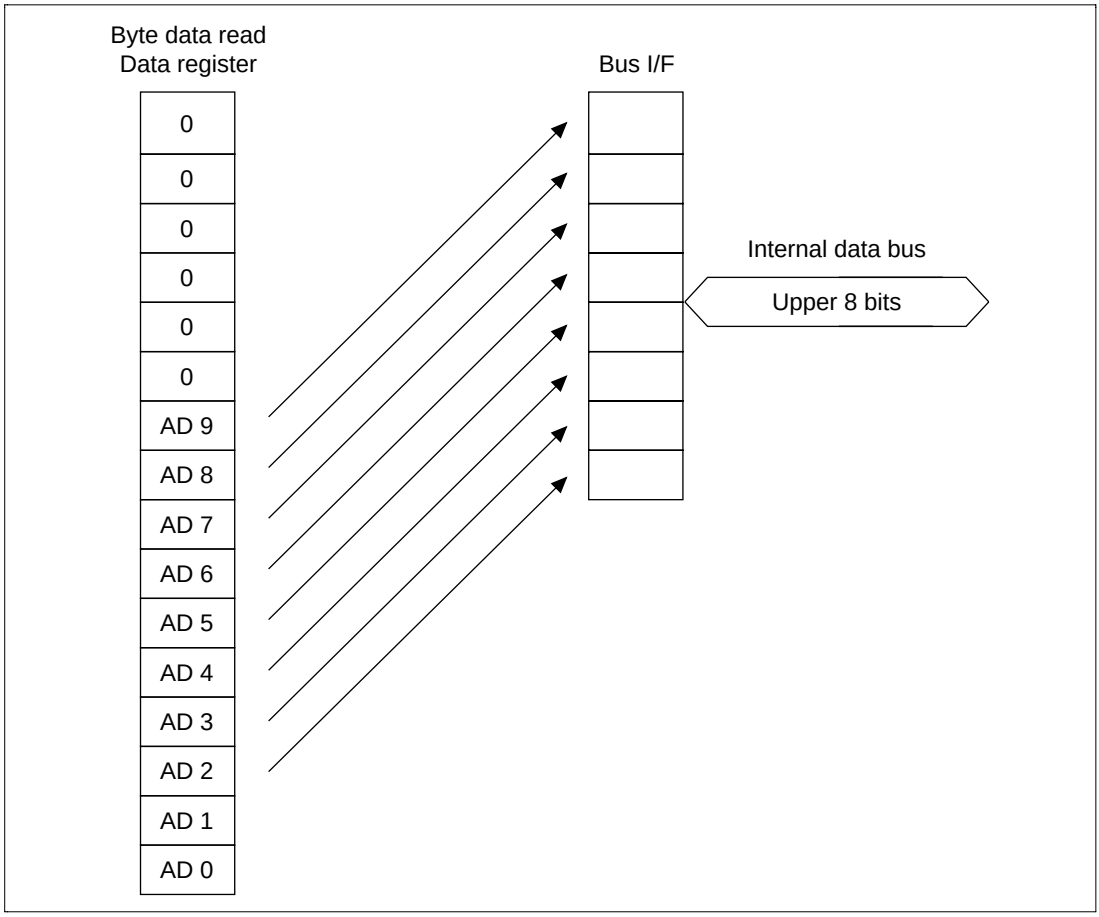


Figure 15.3 ADDR Read Operation (2)

15.4 Operation

- The high speed A/D converter has 10-bit resolution.
- In addition to the four operating modes of select or group, and single or scan can be set in combination with buffer operation and simultaneous sampling operation.
- Select mode uses one channel and group mode selects multiple channels.
- One start in the single mode performs conversions on all selected channels, and one start in the scan mode performs repeated conversions until stopped by software.
- In buffer operation, the previous conversion result is saved in a buffer register at the end of a conversion for the relevant channel.
- In simultaneous sampling operation, the analog input voltages of two channels are sampled simultaneously then converted in order.
- Software, a timer conversion start trigger (MTU), or an ADTRG input can be selected as the conversion start condition.
- High speed start mode or low power conversion mode can be selected for A/D conversion using the PWR bit setting.
- When changing the operation mode or input channel, rewrite the ADCSR, ADCR while the ADST bit is cleared to 0. After rewriting the ADCSR, ADCR, A/D conversion will be restarted when the ADST bit is set to 1. Operation mode or input channel changes can be made simultaneously with ADST bit setting. When stopping an A/D conversion before completion, 0 clear the ADST bit.

15.4.1 Select-Single Mode

Choose select-single mode when doing A/D conversions for one channel only.

When the ADST bit is set to 1, A/D conversion is started according to the designated conversion start conditions. The ADST bit is held to 1 during the A/D conversion and is automatically cleared to 0 upon completion.

The ADF flag is also set to 1 at the end of conversion. If the ADIE bit is set to 1 at this time, an ADI interrupt request is generated. The ADF flag is cleared by reading the ADCSR, then writing a 0.

Figure 15.4 shows an example of operation in the select-single mode when AN1 is selected.

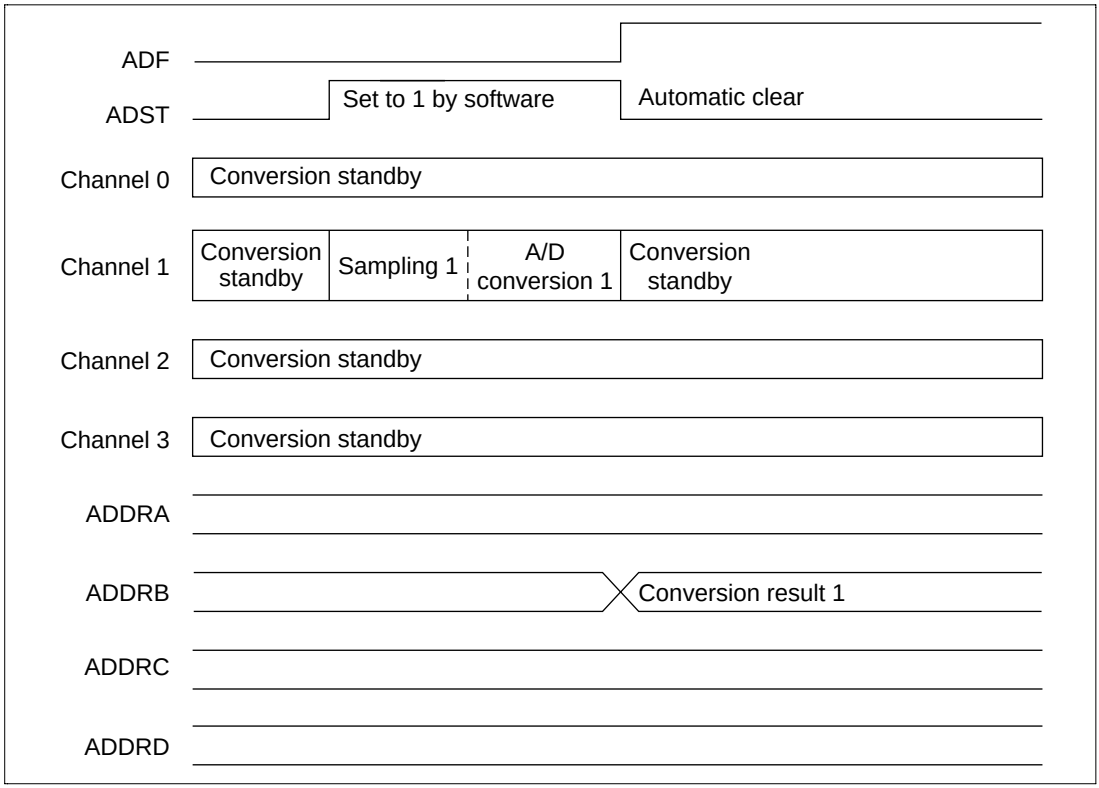


Figure 15.4 A/D Converter Operation Example (Select-Single Mode)

15.4.2 Select-Scan Mode

Choose select-scan mode when doing repeated A/D conversions for one channel. This is useful when doing continuous monitoring of the analog input of one channel.

When the ADST bit is set to 1, A/D conversion is started according to the designated conversion start conditions. The ADST bit is held to 1 until 0 cleared by software. A/D conversion for the selected input channel is repeated during that interval.

The ADF flag is set to 1 at the end of the first conversion. At this point, if the ADIE bit is set, an ADI interrupt request is issued, and the A/D converter is halted. With the A/D converter in stop mode due to an ADI interrupt request, conversion is restarted when the ADF flag is cleared to 0. The ADF flag is cleared by reading the ADCSR then writing a 0.

Figure 15.5 shows an example of operation in the select-scan mode when AN1 is selected.

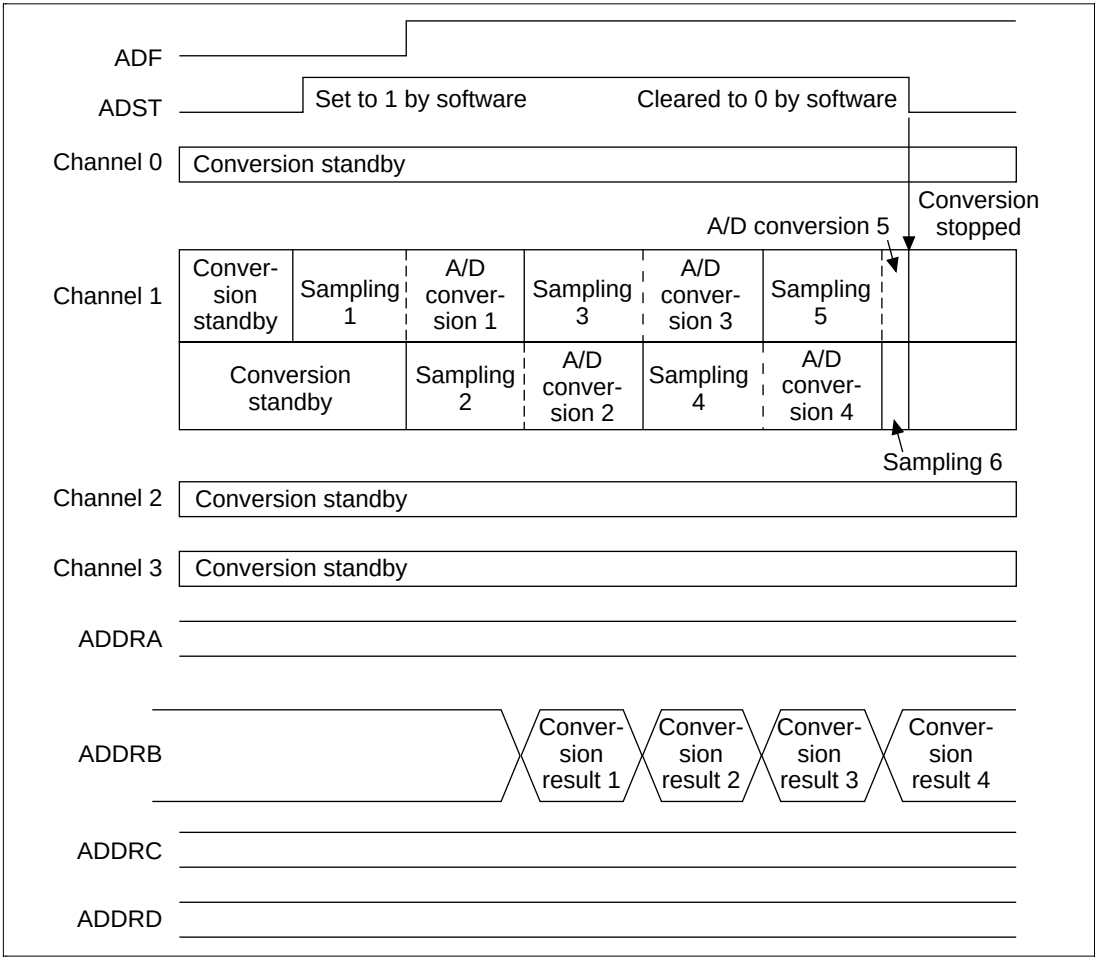


Figure 15.5 A/D Converter Operation Example (Select-Scan Mode)

15.4.3 Group-Single Mode

Choose group-single mode when doing A/D conversions for multiple channels.

When the ADST bit is set to 1, A/D conversion is started according to the designated conversion start conditions. The ADST bit is held to 1 during A/D conversion and is automatically cleared to 0 when all conversions for the designated input channels are completed.

The ADF flag is set to 1 when all conversions for the designated input channels are completed. If the ADIE bit is set to 1 at this time, an ADI interrupt request is generated. The ADF flag is cleared by reading the ADCSR then writing a 0.

Figure 15.6 shows an example of operation in the group-single mode when AN0–AN2 are selected.

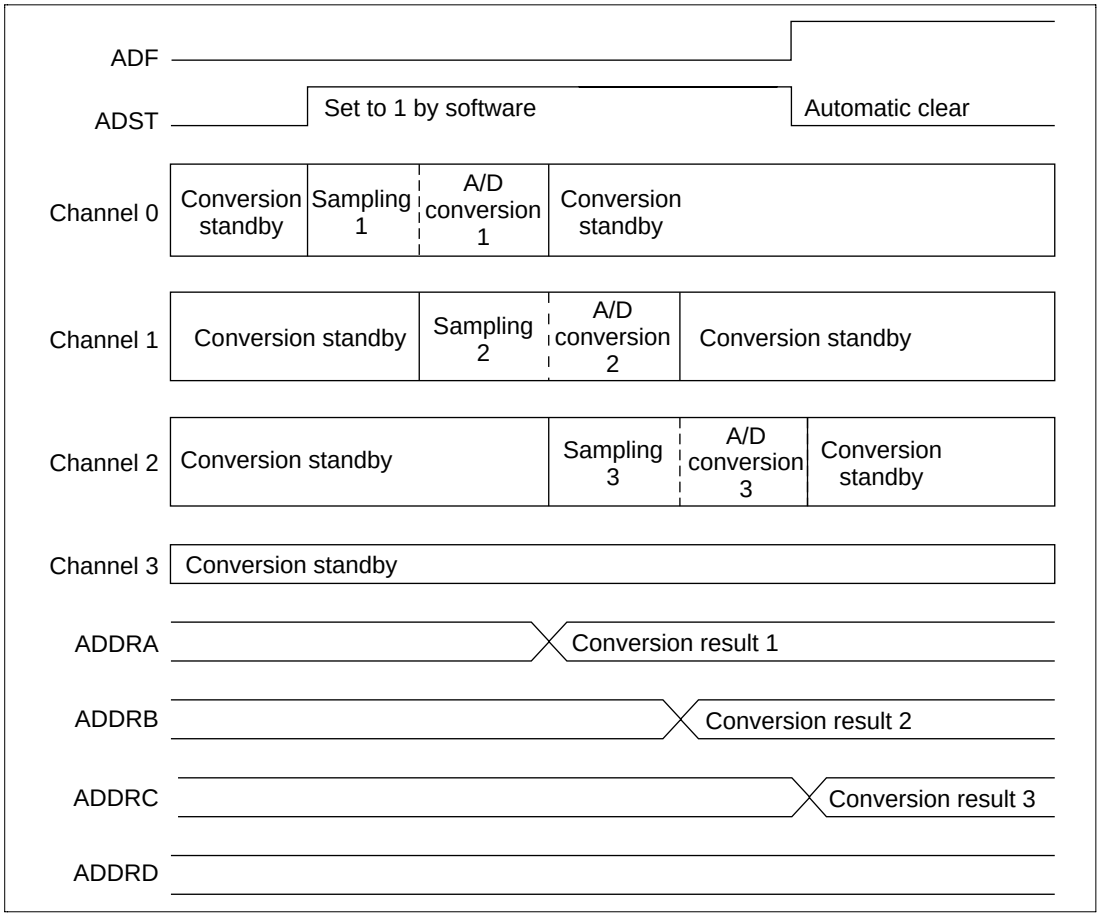


Figure 15.6 A/D Converter Operation Example (Group-Single Mode)

15.4.4 Group-Scan Mode

Choose group-scan mode when doing repeated A/D conversions for multiple channels. This is useful when doing continuous monitoring of the analog inputs of multiple channels.

When the ADST bit is set to 1, A/D conversion is started according to the designated conversion start conditions. The ADST bit is held to 1 until 0 cleared by software. A/D conversion for the selected input channels is repeated during that interval.

The ADF flag is set to 1 at the completion of the first conversions of all the designated input channels. At this point, if the ADIE bit is set to 1, an ADI interrupt request is issued, and the A/D converter is temporarily halted. With the A/D converter in stop mode due to an ADI interrupt request, conversion is restarted when the ADF flag is cleared to 0. The ADF flag is cleared by reading the ADCSR, then writing a 0.

Figure 15.7 shows an example of operation in the group-scan mode when AN0–AN2 are selected.

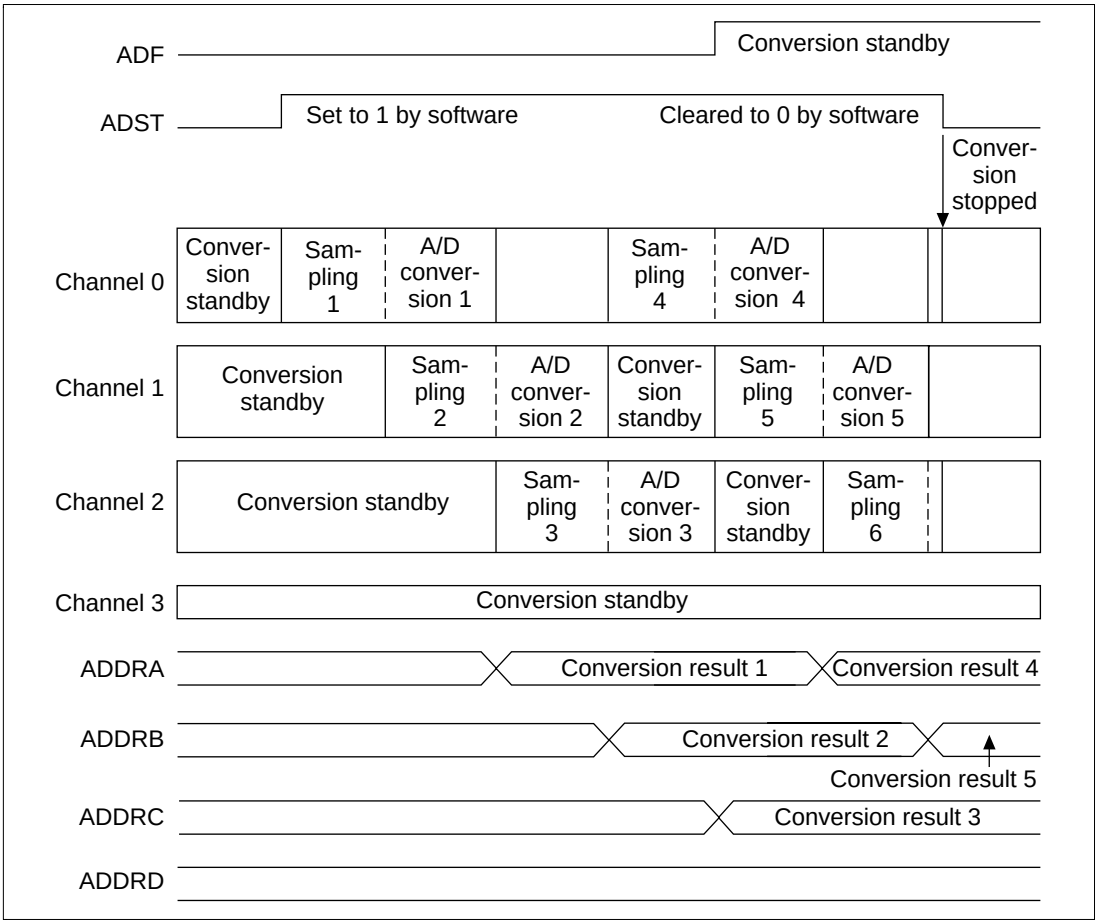


Figure 15.7 A/D Converter Operation Example (Group-Scan Mode)

15.4.5 Buffer Operation

When conversion ends on the relevant channel, the conversion result is stored in the ADDR, and simultaneously, the previously stored result is transferred to another ADDR. Buffer operation can be selected from the following:

- AN0 → ADDRA → ADDRB (Two-stage, one-group operation)
- AN0 → ADDRA → ADDRC, AN1 → ADDRB → ADDRD (Two-stage, two-group operation)
- AN0 → ADDRA → ADDRB → ADDRC → ADDRD (Four-stage, one-group operation)

To use in combination with simultaneous sampling, set GRP = 1, BUFE1, BUFE0 = B'10, and CH2 = 0. Buffer operation timing is shown in figure 15.8.

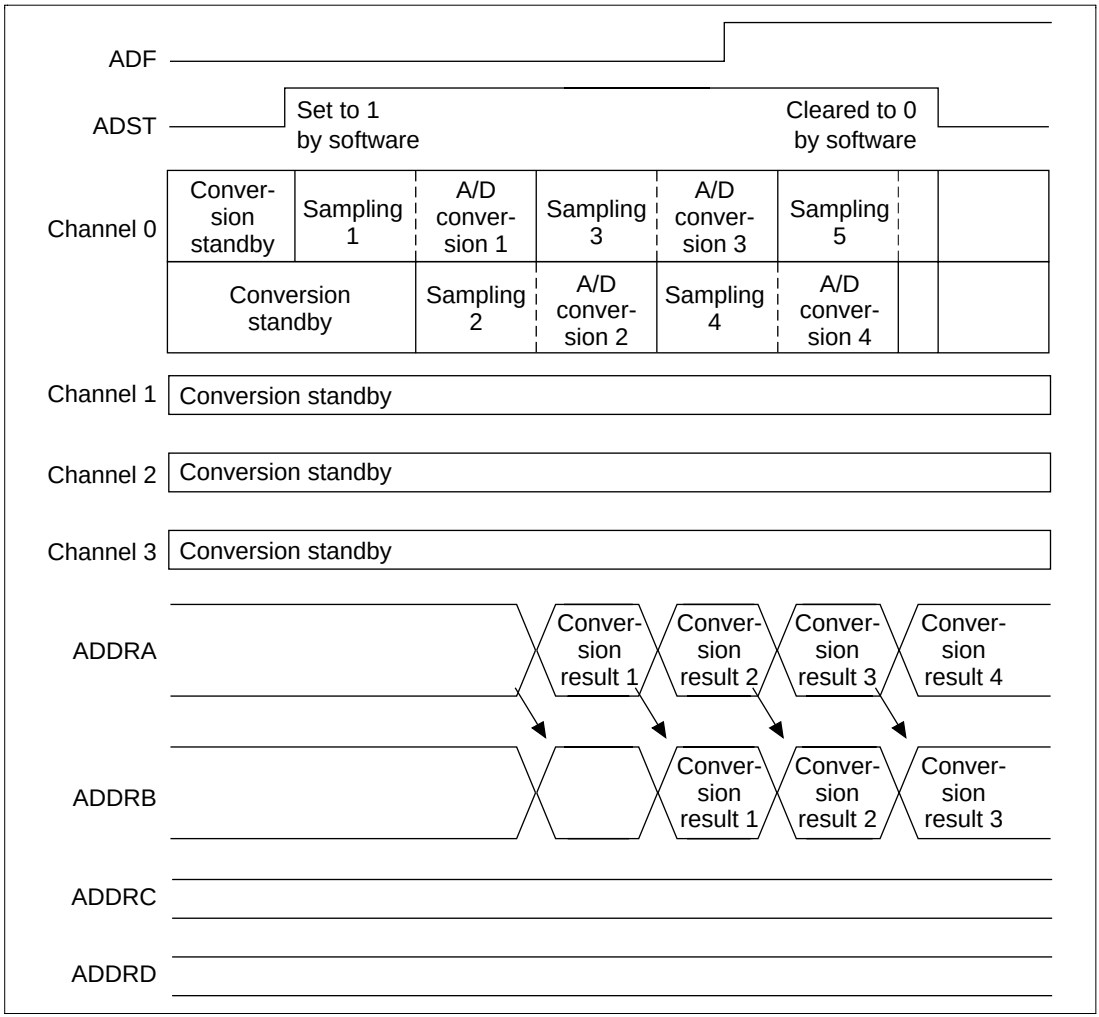


Figure 15.8 Buffer Operation Example (Select Scan Mode: Two-Stage One-Group Operation, When CH2–CH0 = B'001)

Buffer-Only Operation: When performing conversion only on the analog input channels specified by the BUFE1 and BUFE0 bits, select group mode, and you can select the ADF flag setting conditions with the CH2–CH0 bits.

Table 15.4 shows conversion during buffer operation and ADF flag setting conditions. The ADF flag is set at the point in the table when the final conversion has ended. In single mode, conversion is halted after the ADF flag is set to 1. In scan mode, conversion continues, and the converted data is stored in sequence in the buffer registers specified by the BUFE1 and BUFE0 bits.

When the ADF flag is set to 1, if the ADIE bit is also set to 1, an ADI interrupt is issued. After the ADCSR is read, the ADF flag is cleared by a 0 write.

With select single mode, the A/D converter goes into standby mode at the end of every conversion cycle. The A/D converter is restarted by software, a timer trigger, or external trigger. When the number of conversion cycles shown in table 15.4 have ended, the ADF flag is set to 1.

Table 15.4 Conversion Channel and ADF Flag Setting/Clearing Conditions during Buffer Operation 1

Channel Setting			Sampling Channel		
CH2	**CH1**	**CH0**	**BUFE1, BUFE0 = B'01**	**BUFE1, BUFE0 = B'10**	**BUFE1, BUFE0 = B'11**
0	0	0	AN0 1 time (ADDRA)	AN0, AN1 1 time (ADDRB)	AN0 1 time (ADDRA)
		1	AN0 2 times (ADDRB)		AN0 2 times (ADDRB)
	1	0	*	AN0, AN1 2 times (ADDRD)	AN0 3 times (ADDRC)
		1	*		AN0 4 times (ADDRD)
1	—	—	*	*	*

Note: * See table 15.5.

Combined Group Mode and Buffer Operation: Continuous conversion is possible on analog input channels (AN0 and AN1) specified by bits BUFE1 and BUFE0 as well as AN4–AN7 due to setting of bits CH2–CH0.

Table 15.5 shows conversion during buffer operation and ADF flag setting conditions. The ADF flag is set at the point in the table when the final conversion has ended. In this case, conversion is performed on the analog input corresponding with the ADDR specified in the buffer register. For example, when BUFE1 and BUFE0 = B'11 and CH2–CH0 = B'110, conversion results are stored in ADDRA and ADDRE–ADDRG. Also, contents of ADDRA–ADDRC before the start of conversion are transferred to ADDRB–ADDRD.

In single mode, conversion is halted after the ADF flag has been set to 1. Conversion continues in scan mode.

Table 15.5 Conversion Channel and ADF Flag Setting/Clearing Conditions During Buffer Operation 2

Channel Setting			Sampling Channel		
CH2	**CH1**	**CH0**	**BUFE1, BUFE0 = B'01**	**BUFE1, BUFE0 = B'10**	**BUFE1, BUFE0 = B'11**
0	0	—	*	*	*
	1	0	AN0, AN2 (ADDRC)		
		1	AN0, AN2, AN3 (ADDRD)		
1	0	0	AN0, AN2–AN4 (ADDRE)	AN0, AN1, AN4 (ADDRE)	AN0, AN4 (ADDRE)
		1	AN0, AN2–AN5 (ADDRF)	AN0, AN1, AN4, AN5 (ADDRF)	AN0, AN4, AN5 (ADDRF)
	1	0	AN0, AN2–AN6 (ADDRG)	AN0, AN1, AN4–AN6 (ADDRG)	AN0, AN4–AN6 (ADDRG)
		1	AN0, AN2–AN7 (ADDRH)	AN0, AN1, AN4–AN7 (ADDRH)	AN0, AN4–AN7 (ADDRH)

Note: * See table 15.4.

ADF Flag Clearing: When the DTC and DMAC are started up due to an A/D conversion end interrupt, the ADF flag is cleared when the ADDR specified in table 15.4 or 15.5 has been read.

Resetting the Number of Buffer Operations: Clear the BUFE1 and BUFE0 bits to B'00 in conversion standby mode or when the converter has been halted. The number of buffer operations is cleared to 0.

Updating Buffer Operations: Clear the BUFE1 and BUFE0 bits to B'00 in conversion standby mode or when the converter has been halted. Thereafter, set BUFE1 and BUFE0, and the buffer operations shown in tables 15.4 and 15.5 are performed when conversion is resumed.

15.4.6 Simultaneous Sampling Operation

With simultaneous sampling, continuous conversion is conducted with sampling of the input voltages on two channels at the same time. Simultaneous sampling is valid in group mode. Channels for sampling are determined by the CH2 and CH1 bits of the RDSCR. The combinations are shown in table 15.6. For example, if GRP = 1 when CH2 and CH1 = B'11, sampling occurs in order in the following pairs: AN0, AN1→AN2, AN3→AN4, AN5→AN6, AN7. Sampling timing is shown in figure 15.9.

Table 15.6 Simultaneous Sampling Channels

Channel Setting		
CH2	**CH1**	**Sampling Channels, GRP 1**
0	0	AN0, AN1
	1	AN0, AN1→AN2, AN3
1	0	AN0, AN1→AN2, AN3→AN4, AN5
	1	AN0, AN1→AN2, AN3→AN4, AN5→AN6, AN7

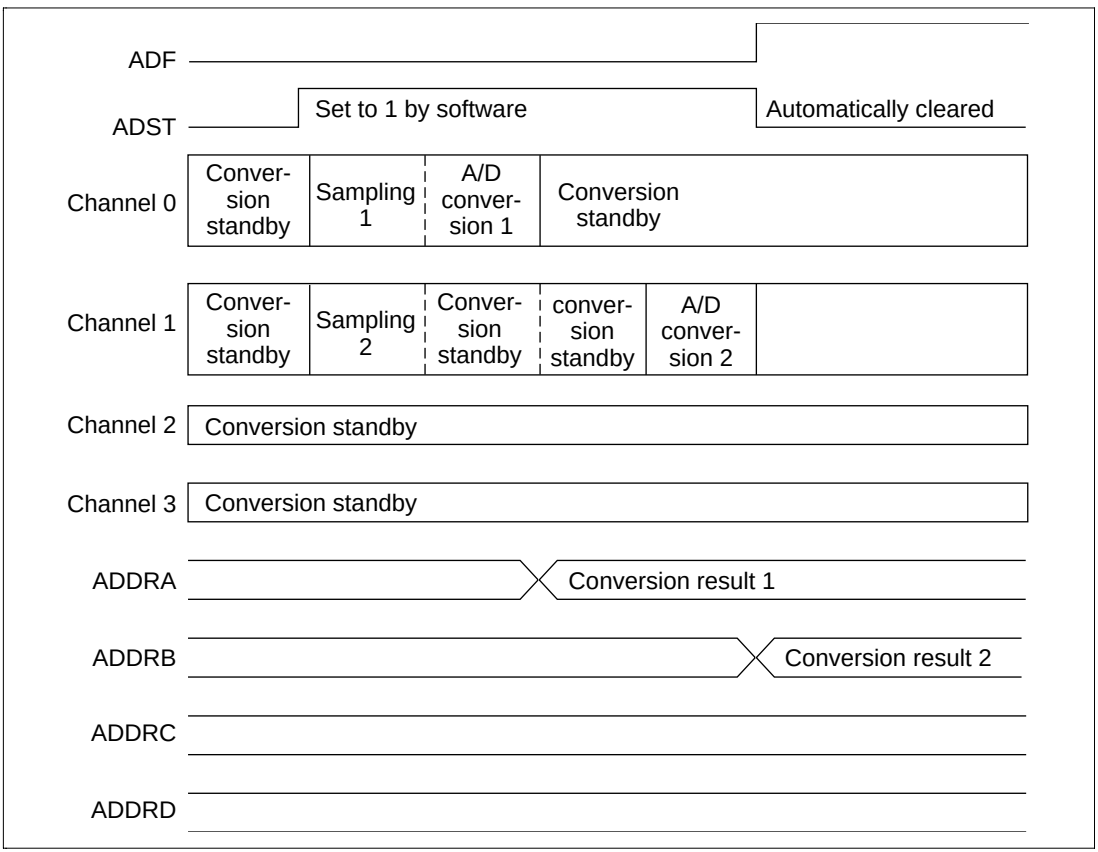


Figure 15.9 Simultaneous Sampling Operation (Group Single Mode)

15.4.7 Conversion Start Modes

The conversion start mode of the high speed A/D converter is set by the PWR bit of the ADCSR. When the PWR bit is cleared to 0, low-power conversion mode is set and the internal analog circuit becomes inactive. High-speed start mode is set by setting the PWR bit to 1, and the analog circuit becomes active.

In the low-power conversion mode, power is applied to the analog circuitry simultaneous to the conversion start (ADST set). When 200 cycles of the reference clock have elapsed, conversion becomes possible for the analog circuit and the first A/D conversion begins. When performing consecutive conversions, the second and later conversions are executed in 10 cycles. Select the basic clock with the CKS bit of the ADCSR. When the A/D conversion ends, ADST is cleared to 0 and the analog circuit power supply is automatically cut off. Because the analog circuit is only active during the A/D conversion operation period in this mode, current consumption can be reduced.

In high-speed start mode, ADST is cleared to 0 when A/D conversion ends. Power continues to be supplied to the analog circuitry, and conversion-ready status is maintained. Conversion is restarted immediately by resetting ADST to 1. However, the first conversion after power-on begins 200 cycles after setting ADST. Clear the PWR bit to 0 to switch off the analog power supply. When performing consecutive conversions, the second and later conversions are executed in 20 cycles. Because the analog circuit is always active in this mode, A/D conversion can be executed at high speed.

Figures 15.10 and 15.11 show examples of conversion start operation timing.

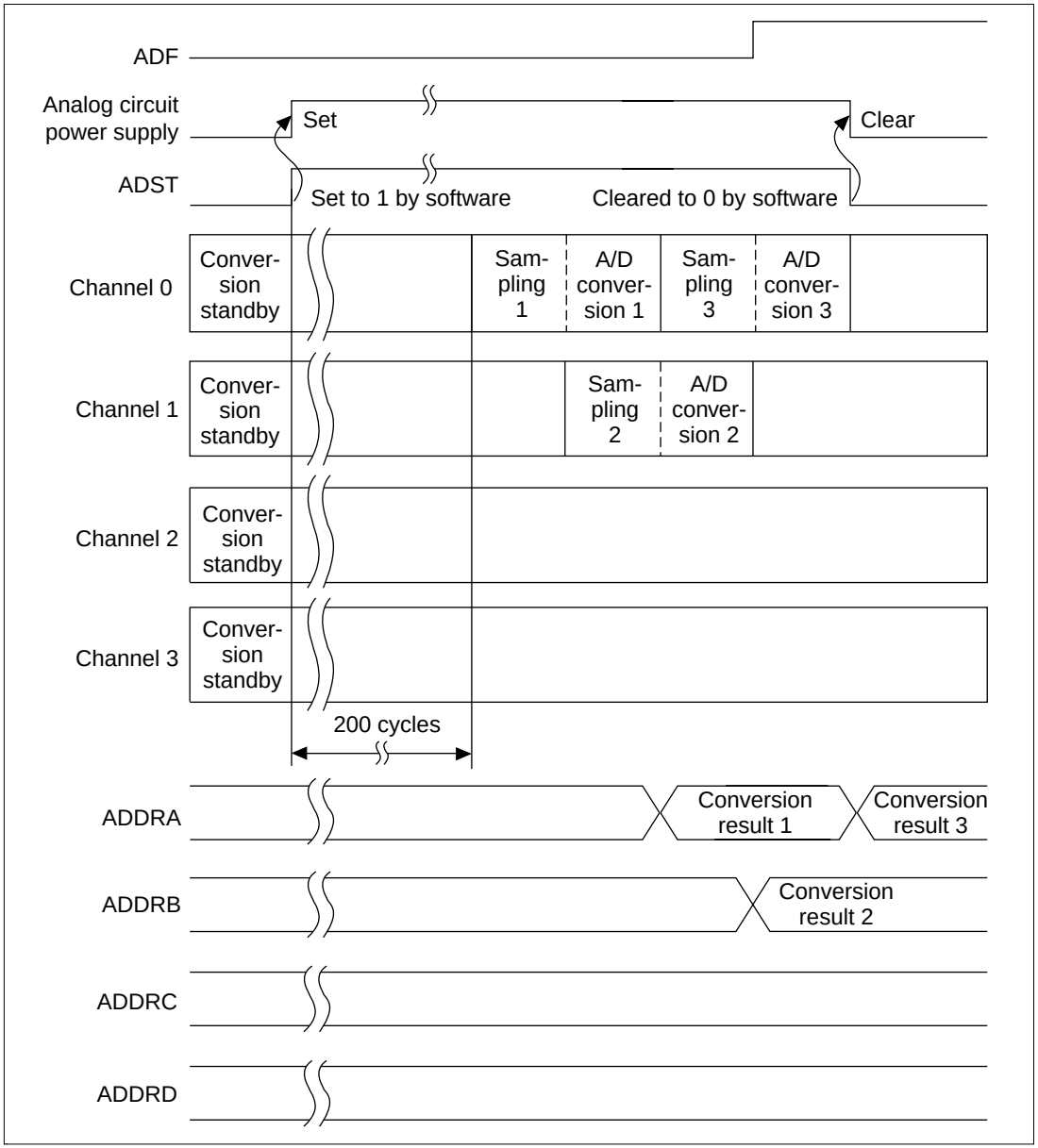


Figure 15.10 Conversion Start Operation (Low-Power Conversion Mode)

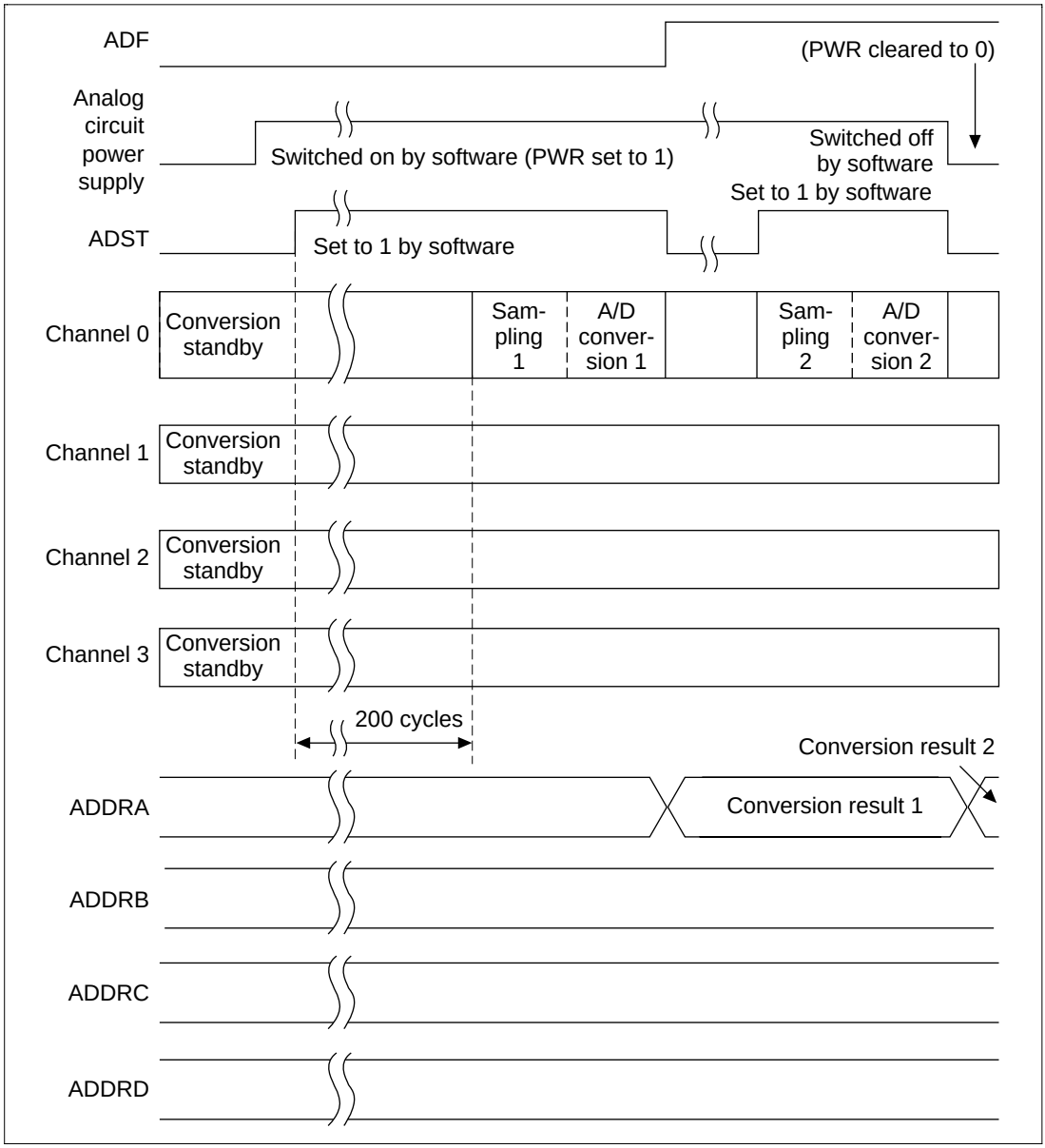


Figure 15.11 Conversion Start Operation (High-Speed Start Mode)

15.4.8 Conversion Start by External Input

A/D conversions can be started by trigger signals generated by timer conversion start triggers or $\overline{\text{ADTRG}}$ inputs. When a trigger signal designated by the TRGS1 and TRGS0 bits of the ADCR occurs, the ADST bit of the ADCSR is set to 1 and A/D conversion is started.

The other operations are the same as when the ADST bit is set to 1 by software. Figure 15.12 shows an example of the timing when the ADST bit is set by an external input.

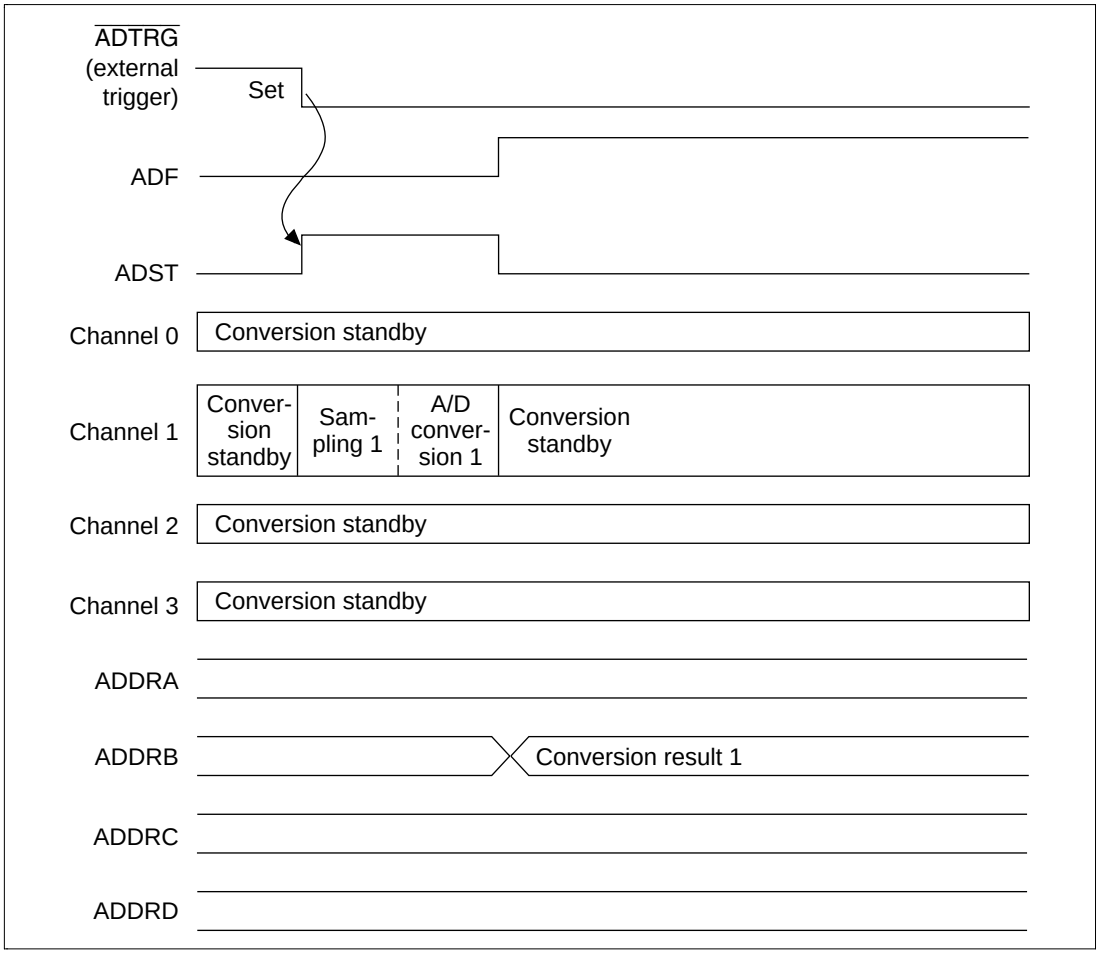


Figure 15.12 Conversion Start by $\overline{\text{ADTRG}}$ Conversion Start Trigger

15.4.9 A/D Conversion Time

The high speed A/D converter has an on-chip sample and hold circuit. The high speed A/D converter samples the input at time t_D after the ADST bit is set to 1, and then starts the conversion.

The A/D conversion time t_{CONV} is the sum of the conversion start delay time t_D, the input sampling time t_{SPL}, and the operating time t_{CP}. This conversion time is not a set value, but is decided by the t_D ADCSR write timing, or the timer conversion start trigger generation timing.

Figure 15.13 shows an example of A/D conversion timing. Table 15.7 lists A/D conversion times.

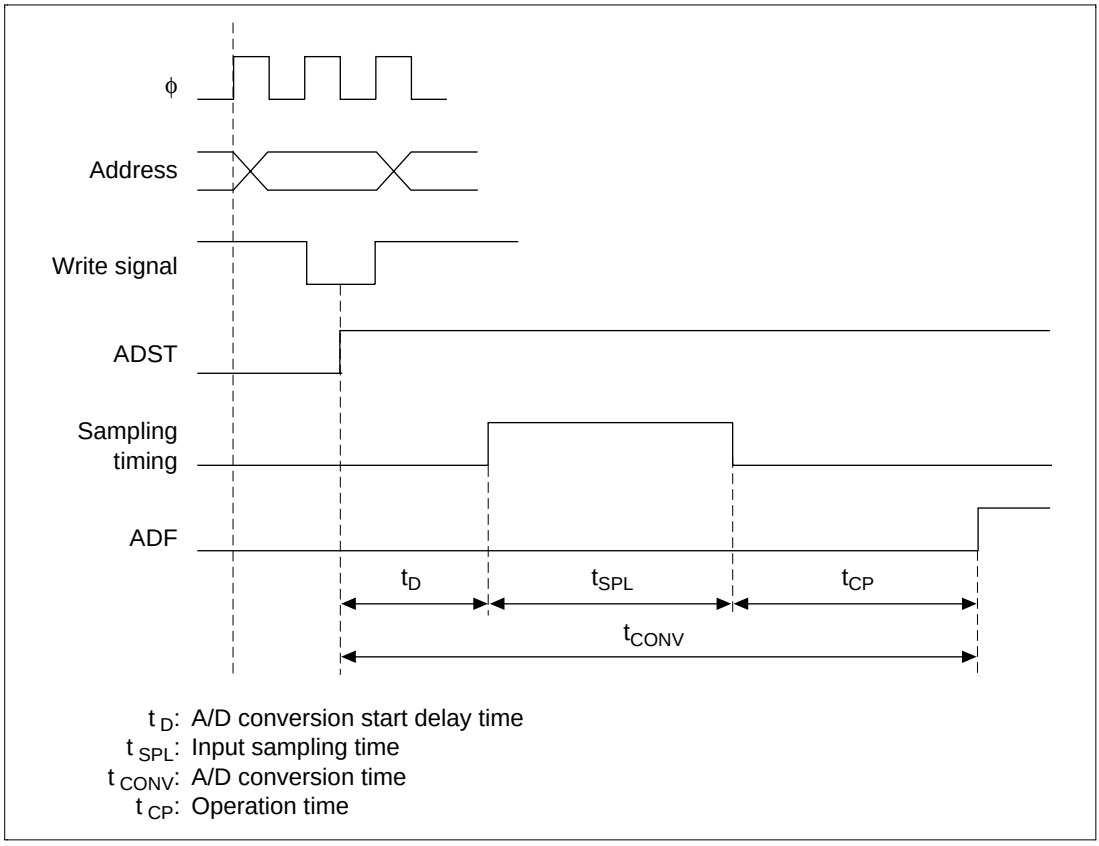


Figure 15.13 A/D Conversion Timing

Table 15.7 A/D Conversion Times

		CKS = 0			CKS = 1		
Time	**Symbol**	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**
A/D conversion start delay time	t_D	1.5	1.5	1.5	1.5	1.5	1.5
Input sampling time	t_{SPL}	20	20	20	40	40	40
A/D conversion time	t_{CONV}	42.5	42.5	42.5	82.5	82.5	82.5

Notes: 1. Unit: states

2. Table entries are for when ADST = 1. If 200 states have not elapsed since the PWR bit has been set, no conversions are done until after those 200 states have occurred. When PWR = 0, add 200 states to the first A/D conversion start delay time. When continuously executing conversion, tcp for the second time and following is 20 cycle when CKS=0 and 40 cycle when CKS=1.

The CKS bit of the ADCSR is the operation time t_{CONV}, but set so that this is 2 µs or greater. Table 15.8 shows the operating frequency and CKS bit settings.

Table 15.8 Operating Frequency and CKS Bit Settings

		Minimum Conversion Time (µs)				
CKS	**Conversion Time (States)**	**28 MHz**	**20 MHz**	**16 MHz**	**10 MHz**	**8 MHz**
0	42.5	—	2.1	2.6	4.3	5.3
1	82.5	2.9	4.2	5.0	8.3	10.3

Note: The indication "—" means the setting is not available.

15.5 Interrupts

The high speed A/D converter generates an A/D conversion end interrupt (ADI) upon completion of A/D conversions. The ADI interrupt request can be enabled or disabled by the ADIE bit of the ADCSR.

The DTC or DMAC can be activated by ADI interrupts. When converted data is read by the DTC or DMAC upon an ADI interrupt, consecutive conversions can be done without software responsibility.

Table 15.9 lists the high speed A/D converter interrupt sources.

During scan mode, if the ADIE bit is set to 1, A/D conversion is temporarily suspended immediately when the ADF flag is set to 1. A/D conversion is restarted when the ADF flag is cleared to 0.

When the DTC or DMAC are activated by an ADI interrupt, the ADF flag is cleared to 0 when the final specified data register is read.

Table 15.9 High Speed A/D Converter Interrupt Sources

Interrupt Source	Description	DTC, DMAC Activation
ADI	Interrupt caused by conversion end	Possible

15.6 Notes on Use

Take note of the following for the A/D converter.

1. Analog input voltage range

   During A/D conversions, see that the voltage applied to the analog input pins AN0–AN7 is within the range Avss ≤ AN0–AN7 ≤ AVcc.

2. AVcc and AVss input voltages

   The AVcc and AVss input voltage must be AVcc = Vcc ± 10%, AVss = Vss. When not using the A/D converter, use AVcc = Vcc, AVss = Vss. During the standby mode, use V_{RAM} ≤ Avcc ≤ 5.5V, AVss = Vss. V_{RAM} is the RAM standby voltage.

3. AVref input voltage

   The analog standard voltage AVref (AV_{ref}) must be Avref ≤ AVcc. When not using the A/D converter, use AV_{ref} = Vcc. During the standby mode, use V_{RAM} ≤ AVref ≤ AVcc. V_{RAM} is the RAM standby voltage.

4. Input ports

   The time constant for the circuit connecting to the input port must be shorter than the sampling time of the A/D converter. Input voltage may not be sampled sufficiently when the time constant of the circuit is long.

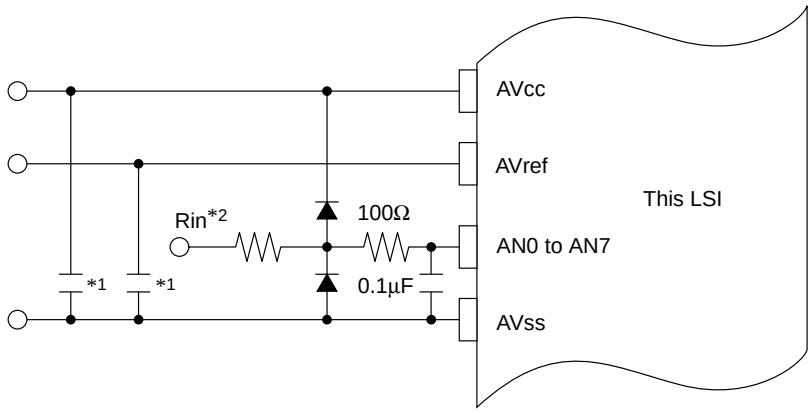
5. Conversion start modes

   Depending on the PWR bit setting, the demand for A/D conversion will differ for the high-speed start mode and low-demand conversion mode.

6. Analog input pins handling

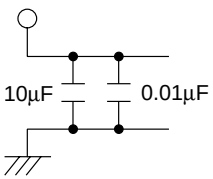
   Connect a protection circuit as shown in figure 15.14 to prevent analog input pins (AN0–AN7) from being destroyed due to abnormal voltage from surge, etc. This circuit is also equipped with a CR filter to control errors due to noise. The circuit shown in the diagram is only an example and the number of circuits is to be determined by considering the actual condition of use.

   Figure 15.15 shows an equivalent circuit of analog input pins and table 15.10 shows the specification of the analog input pins.



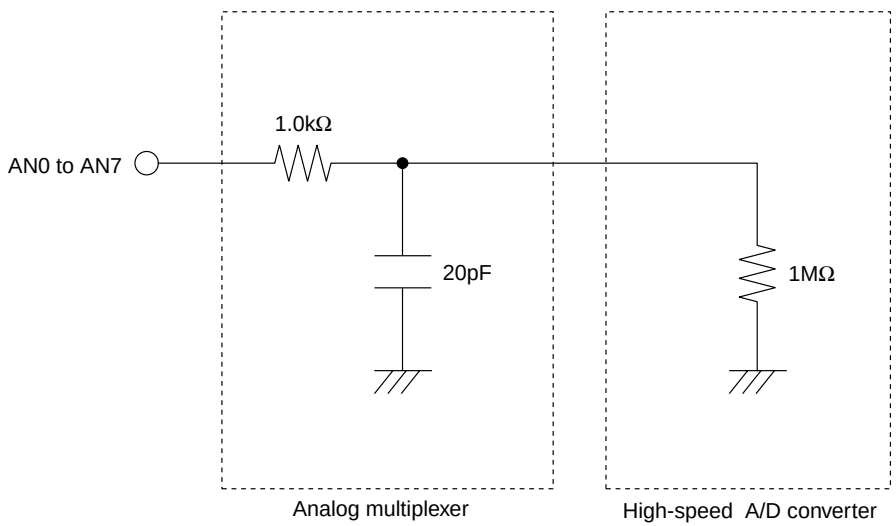
Notes: Numbers are only to be noted as reference value

*1



*2 Rin: Input impedance

Figure 15.14 Example of a Protection Circuit for the Analog Input Pins



Note: Numbers are only to be noted as reference value

Figure 15.15 Equivalent Circuit of Analog Input Pins

Table 15.10 Analog Input Pin Specification

Item	Min	Max	Unit
Analog input capacity	—	20	pF
Permitted source impedance	—	1	kΩ

Section 16 Mid-Speed A/D Converter (A Mask)

16.1 Overview

The mid-speed A/D converter has 10 bit resolution, and can select from a maximum of eight channels of analog input.

The mid-speed A/D converter is structured by two independent modules (A/D0 and A/D1)

16.1.1 Features

The mid-speed A/D converter has the following features:

- 10-bit resolution
- Eight input channels (four channels times two)
- Analog conversion voltage range setting is selectable
  - — Using the standard voltage pin (AVref) as an analog standard voltage (Vref), conversion of analog input from 0V to Vref (only with SH7041A, SH7043A, and SH7045).
    (Connected to AV_{CC} internally in the SH7040A, SH7042A, and SH7044.)
- High speed conversion
  - — Minimum conversion time: per channel
  - — Operation frequency: f≤20MHz, CKS=0, 1
    6.7µs (20MHz, CKS=1)
  - — Operation frequency: f>20MHz, CKS=0
    9.3µs (28.7MHz, CKS=0)
- Multiple conversion modes
  - — Single mode/scan mode
  - — 2 channel simultaneous conversion
- Three types of conversion start
  - — Software, timer conversion start trigger (MTU), or ADTRG pin can be selected.
- Eight data registers
  - — Conversion results stored in 16-bit data registers corresponding to each channel.
- Sample and hold function
- A/D conversion end interrupt generation
  - — An A/D conversion end interrupt (ADI) can be generated on completion of A/D conversion.
- Furthermore, ADI0 (A/D0 interrupt request) can activate DTC and ADI1 (A/D1 interrupt request) can activate DMAC.

16.1.2 Block Diagram

Figure 16.1 is the block diagram of the mid-speed A/D converter.

AV_{CC}, AVref and AV_{SS} pins of both A/D are common in LSI.

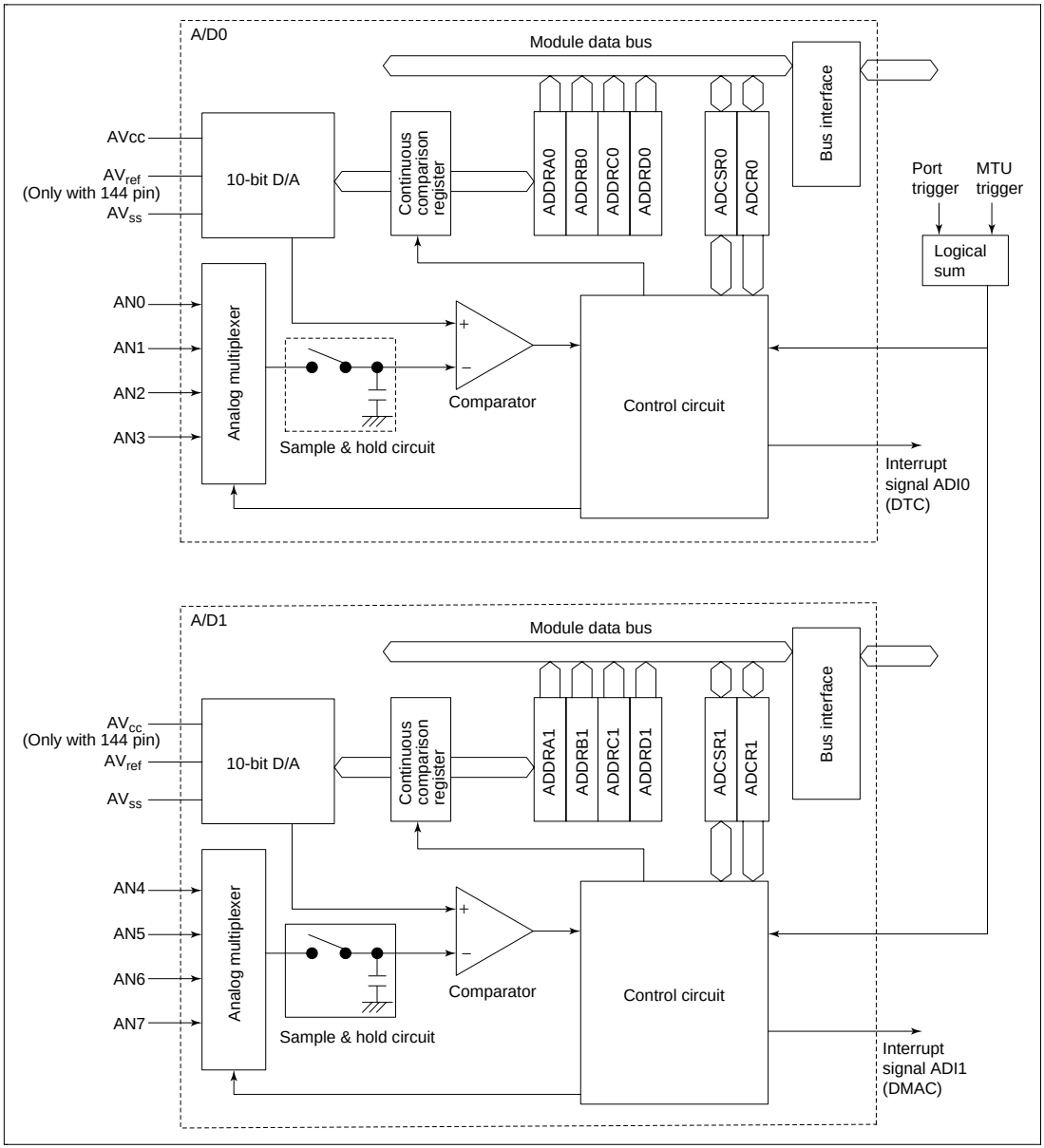


Figure 16.1 Mid-Speed A/D Converter Block Diagram

16.1.3 Pin Configuration

Table 16.1 shows the input pins used with the mid-speed A/D converter.

The AV_{CC} and AV_{SS} pins are for the mid-speed A/D converter internal analog section power supply. AVref pin is the A/D conversion standard voltage.

Table 16.1 Pin Configuration

Pin		Abbreviation	I/O	Function
Analog supply		Av_{CC}	I	Analog section power supply
Analog ground		AV_{SS}	I	Analog section ground and A/D conversion standard voltage
Standard voltage		AVref*	I	A/D conversion standard voltage (SH7041A, SH7043A, and SH7045 only)
A/D0	Analog input 0	AND	I	Analog input channel 0
	Analog input 1	AN1	I	Analog input channel 1
	Analog input 2	AN2	I	Analog input channel 2
	Analog input 3	AN3	I	Analog input channel 3
A/D1	Analog input 4	AN4	I	Analog input channel 4
	Analog input 5	AN5	I	Analog input channel 5
	Analog input 6	AN6	I	Analog input channel 6
	Analog input 7	AN7	I	Analog input channel 7
A/D external trigger input		$\overline{\text{ADTRG}}$	I	External trigger for A/D conversion start

Note: * In the SH7040A, SH7042A, and SH7044, AV_{ref} is connected to AV_{CC} internally.

16.1.4 Register Configuration

Table 16.2 shows the register configuration of the mid-speed A/D converter.

Table 16.2 Register Configuration

Name	Abbreviation	R/W	Initial Value	Address	Access Size
A/D0 data register AH	ADDRA0H	R	H'00	H'FFFF8400	8, 16
A/D0 data register AL	ADDRA0L	R	H'00	H'FFFF8401	8
A/D0 data register BH	ADDRB0H	R	H'00	H'FFFF8402	8, 16
A/D0 data register BL	ADDRB0L	R	H'00	H'FFFF8403	8
A/D0 data register CH	ADDRC0H	R	H'00	H'FFFF8404	8, 16
A/D0 data register CL	ADDRC0L	R	H'00	H'FFFF8405	8
A/D0 data register DH	ADDRD0H	R	H'00	H'FFFF8406	8, 16
A/D0 data register DL	ADDRD0L	R	H'00	H'FFFF8407	8
A/D0 control/status register	ADCSR0	R/(W)*	H'00	H'FFFF8410	8, 16
A/D0 control register	ADCR0	R/W	H'7F	H'FFFF8412	8, 16
A/D1 data register AH	ADDRA1H	R	H'00	H'FFFF8408	8, 16
A/D1 data register AL	ADDRA1L	R	H'00	H'FFFF8409	8
A/D1 data register BH	ADDRB1H	R	H'00	H'FFFF840A	8, 16
A/D1 data register BL	ADDRB1L	R	H'00	H'FFFF840B	8
A/D1 data register CH	ADDRC1H	R	H'00	H'FFFF840C	8, 16
A/D1 data register CL	ADDRC1L	R	H'00	H'FFFF840D	8
A/D1 data register DH	ADDRD1H	R	H'00	H'FFFF840E	8, 16
A/D1 data register DL	ADDRD1L	R	H'00	H'FFFF840F	8
A/D1 control/status register	ADCSR1	R/(W)*	H'00	H'FFFF8411	8
A/D1 control register	ADCR1	R/W	H'7F	H'FFFF8413	8

Note: * Only 0 can be written to bit 7 to clear the flag.

16.2 Register Descriptions

16.2.1 A/D Data Register A–D (ADDRA0–ADDRD0, ADDRA1–ADDRD1)

A/D registers are special registers that read stored results of A/D conversion in 16 bits. There are eight registers: ADDRA0–ADDRD0 (A/D0) and ADDRA1–ADDRD1 (A/D1).

The A/D converted data is 10 bit data which is to the ADDR of the corresponding converted channel for storage. The upper 8 bits of the A/D converted data correspond to the upper byte of the ADDR and the lower 2 bits correspond to the lower byte. Bits 5–0 of the lower byte of ADDR are reserved and always read 0. Analog input channels and correspondence to ADDR are shown in table 16.3.

ADDR can always be read from the CPU. The upper byte may be read directly. The lower byte is transferred through the temporary register (TEMP). For details, see section 16.3, Interface with CPU.

ADDR is initialized to H'0000 during power-on reset or standby mode. ADDR will not be initialized by manual reset.

Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDRn :	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	—	—	—	—	—	—
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W :	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

(n=A to D)

Table 16.3 Analog Input Channel and ADDRA–ADDRD Correspondence

Analog Input Channel	A/D Data Register	Module
AN0	ADDRA0	A/D0
AN1	ADDRB0	
AN2	ADDRC0	
AN3	ADDRD0	
AN4	ADDRA1	A/D1
AN5	ADDRB1	
AN6	ADDRC1	
AN7	ADDRD1	

16.2.2 A/D Control/Status Register (ADCSR0, ADCSR1)

The A/D control/status registers (ADCSR0, 1) are registers that can read/write in 8 bits and control A/D converter operations such as mode selection. There are the ADCSR0 (A/D0) and ADCSR1 (A/D1).

The ADCSR is initialized to H'00 during power-on reset or standby mode. Manual reset does not initialize ADCSR.

Bit :	7	6	5	4	3	2	1	0
	ADF	ADIE	ADST	SCAN	CKS	—	CH1	CH0
Initial value :	0	0	0	0	0	0	1	0
R/W :	R/(W)*	R/W	R/W	R/W	R/W	R	R/W	R/W

Note: * Only 0 can be written to clear the flag.

- Bit 7—A/D End Flag (ADF): Status flag that indicates end of A/D conversion.

Bit 7:

ADF	Description	
0	[Clear conditions]	(Initial value)
	1. Writing 0 to ADF after reading ADF with ADF=1	
	2. When registers of the mid-speed converter are accessed after the DMAC and DTC are activated by ADI interrupt.	
1	[Set conditions]	
	1. Single mode: When A/D conversion is complete	
	2. Scan mode: When A/D conversion of all designated channels are complete	

- Bit 6—A/D Interrupt Enable (ADIE): Enables or disables interrupt request (ADI) due to completion of A/D conversion.

Bit 6:

ADIE	Description	
0	Disables interrupt request (ADI) due to completion of A/D conversion	(Initial value)
1	Enables interrupt request (ADI) due to completion of A/D conversion	

- Bit 5—A/D Start (ADST): Selects start/end of A/D conversion. A1 is maintained during A/D conversion start. It is also possible to set a 1 by the A/D conversion trigger input pin (ADTRG).

Bit 5:

ADST	Description	
0	A/D conversion halted	(Initial value)
1	1. Single mode: Starts A/D conversion. Automatically clears to 0 when conversion of the designated channel is complete 2. Scan mode: Starts A/D conversion. Continuous conversion until cleared to 0 by the software	

- Bit 4—Scan Mode (SCAN): Selects the A/D conversion mode from single mode and scan mode. For operations during single/scan mode, see section 16.4, Operation. When switching modes, proceed while ADST=0.

Bit 4:

SCAN	Description	
0	Single mode	(Initial value)
1	Scan mode	

- Bit 3—Clock Select (CKS): Sets the A/D conversion time. Proceed conversion time switch while adst=0. Always set CKS=0 when operating frequency exceeds 20MHz.

Bit 3:

CKS	Description	
0	Conversion time = 266 states (max)	(Initial value)
1	Conversion time = 134 states (max)	

- Bit 2—Reserved bit: Bit 2 always reads 0. Furthermore, always write 0.
- Bits 1, 0—Channel select 1, 0 (CH1, CH0): Selects the analog input channel along with the SCAN bit. Switch channels while ADST=0.

Channel Selection		Description			
		Single mode		Scan mode	
CH1	CH0	A/D0	A/D1	A/D0	A/D1
0	0	AN0 (Initial value)	AN4 (Initial value)	AN0	AN4
	1	AN1	AN5	AN0, AN1	AN4, AN5
1	0	AN2	AN6	AN0, AN1	AN4–AN6
	1	AN3	AN7	AN0–AN3	AN4–AN7

16.2.3 A/D Control Register (ADCR0, ADCR1)

A/D control registers (ADCR0, 1) are registers that can read/write in 8 bits and enables or disables A/D conversion start of the external trigger input. There are the ADCR0 (A/D0) and ADCR1 (A/D1).

ADCR is initialized to H'7F during power-on reset and standby mode. Manual reset does not initialize ADCR.

Bit :	7	6	5	4	3	2	1	0
Initial value :	TRGE	—	—	—	—	—	—	—
	0	1	1	1	1	1	1	1
R/W :	R/W	R	R	R	R	R	R	R

- Bit 7—Trigger Enable (TRGE): Enables or disables A/D conversion start of input from external or MTU trigger.

Bit 7:

TRGE	Description
0	Disables A/D conversion start of external or MTU trigger (Initial value)
1	Starts A/D conversion on last transition edge of A/D conversion trigger input pin ($\overline{\text{ADTRG}}$) or MTU trigger.

A/D0 and A/D1 are common for external trigger pin and MTU trigger.

A/D0 and A/D1 settings are of logical sum.

- Bits 6–0—Reserved bits: These bits always read as 1. The write value should always be 1.

16.3 Interface with CPU

Although A/D data register ADDR (ADDRA0–ADDRD0, ADDRA1–ADDRD1) are 16-bit registers, the bus width within the chip that integrates with the CPU is 8-bits. So, upper and lower data of the ADDR must be read separately.

To avoid change in data while reading the upper/lower 2 bytes of ADDR, the lower byte data is read through the temporary register (TEMP). The upper byte data can be read directly.

The procedure for reading data from ADDR is as follows: First, read the upper byte data from ADDR. At this time, the upper byte data is read directly into the CPU and the lower byte data is transferred to TEMP of the mid-speed A/D converter. Next, read the lower byte to read the TEMP contents into the CPU.

When reading the ADDR in byte size, read the upper byte before the lower byte. Furthermore, it is possible to read only the upper byte, however, please note that contents are not guaranteed when reading only the lower byte. In addition, when reading ADDR in word size, upper byte is automatically read before the lower byte.

Figure 16.2 shows the data flow when reading from ADDR.

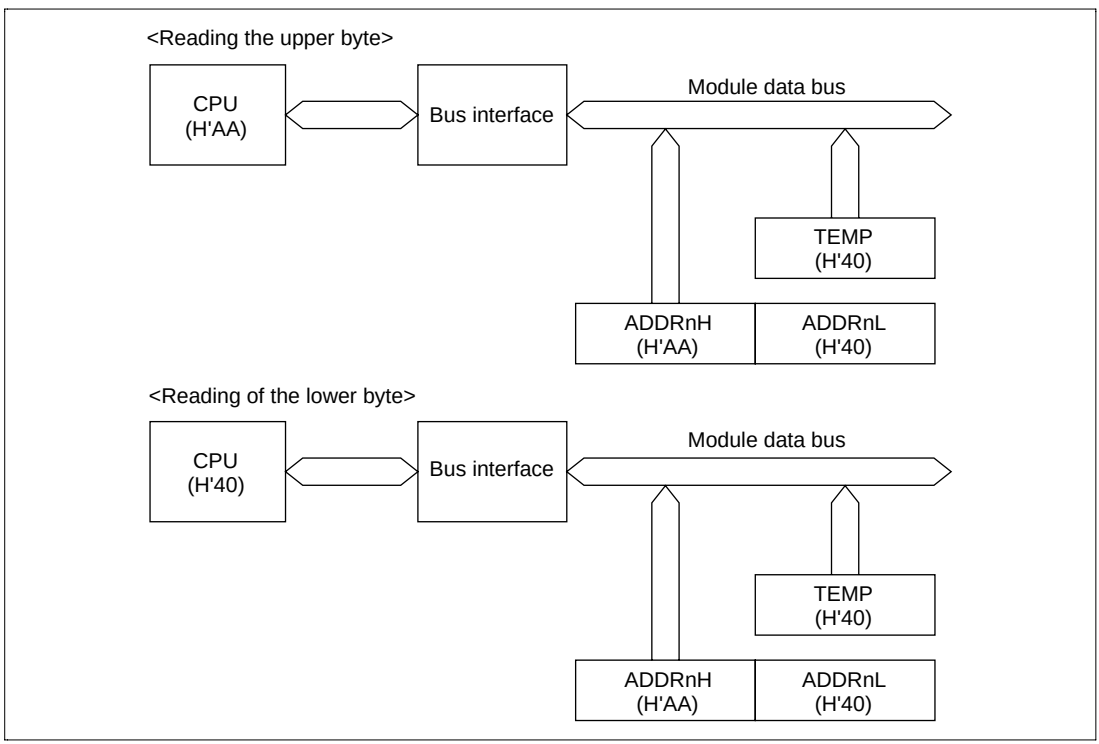


Figure 16.2 ADDR Access Operation (During Reading of (H'AA40))

16.4 Operation

The mid-speed converter operates using the continuous comparison method and is equipped with 10-bit resolution. Operations for the single and scan modes are explained below.

16.4.1 Single Mode (SCAN=0)

The single mode is selected when executing A/D conversion for one channel only. A/D conversion is initiated when the ADST bit of the A/D control/status register is set to 1 by the software or external trigger input. The ADST bit is held to 1 during the A/D conversion and is automatically cleared to 0 upon completion.

When conversion is complete, the ADF bit of ADCSR is set to 1. At this time, if the ADIE bit of ADCSR is 1, ADI interrupt request occurs.

The ADF bit can be cleared by writing 0 after reading ADF=1.

To switch modes or analog input channels during A/D conversion, clear the ADST bit to 0 and stop A/D conversion to avoid malfunction. After switching (mode/channel change and ADST bit setting can be made at the same time), set the ADST bit to 1 to restart A/D conversion.

An example of operation when channel 1 (AN1) is selected in the single mode is shown in figure 16.3 (the bit specification in the example is the ADCSR0 register).

1. Set operation mode to single mode (SCAN=0), input channel to AN1 (CH1=0, CH0=1) and A/D interrupt request to enable (ADIE) then start A/D conversion (ADST=1).
2. When A/D conversion is complete, A/D conversion result is transferred to ADDRB0. At the same time, ADF=1 will become ADF=0 and the mid-speed converter will standby for conversion.
3. Since ADF=1 and ADIE=1, ADI interrupt request will occur.
4. The A/D interrupt process routine will start.
5. After reading ADF=1, write 0 to ADF.
6. Read the A/D conversion result (ADDRB0) and process.
7. End A/D interrupt process routine execution. When ADST bit is set to 1, A/D conversion starts, following steps (2) to (7) above.

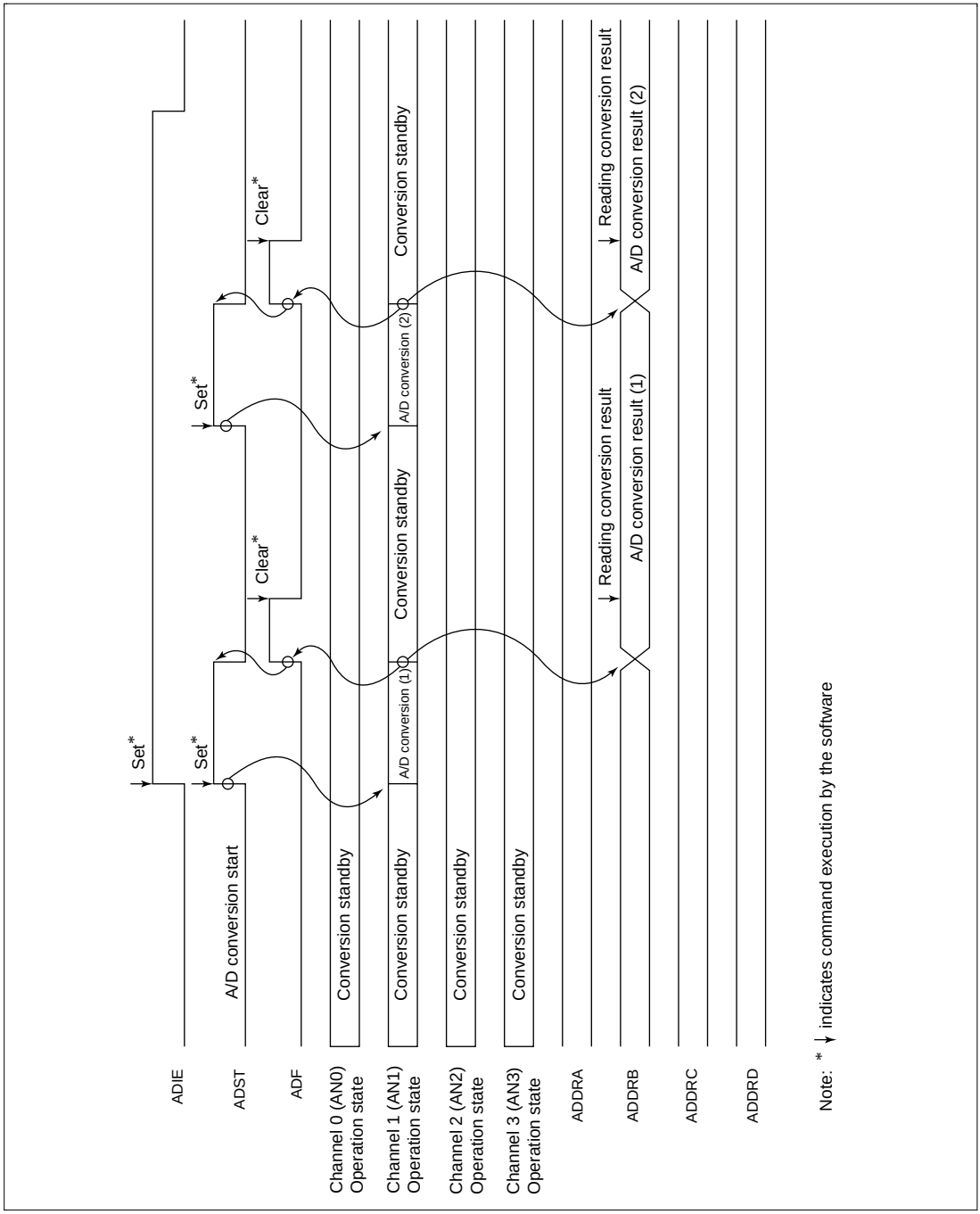


Figure 16.3 Operation Example of Mid-speed A/D Converter (Single Mode, Channel 1 Selected)

16.4.2 Scan Mode (SCAN=1)

The scan mode is optimal for monitoring analog input of multiple channels (including channel 1). A/D conversion is started from channel 1 (AN0 for A/D0 and AN4 for A/D1) of the group when the ADST bit of the A/D control/status register (ADCSR) is set to 1 by the software or external trigger input.

When multiple channels are selected, A/D conversion of channel 2 (AN1 or AN5) is initiated immediately after completion of the channel 1 conversion.

To switch modes or analog input channels during A/D conversion, clear the ADST bit to 0 and stop A/D conversion to avoid malfunction. After switching (mode/channel change and ADST bit setting can be made at the same time), set ADST bit to 1 to restart A/D conversion from channel 1.

An example of operation when three channels of A/D0 (AN0–2) are selected for A/D conversion is shown in figure 16.4 (the bit specification in the example is the ADCSR0 register).

1. Set operation mode to scan mode (SCAN=1), set analog channels to AN0–2 (CH1=1, CH0=0) then start A/D conversion (ADST=1).
2. When A/D conversion for channel 1 is complete, A/D conversion result is transferred to ADDRA0.

   Next, channel 2 (AN1) will automatically be selected and conversion will begin.
3. In the same manner, channel 3 will be converted (AN2).
4. When conversion of all of the selected channels (AN0–AN2) are complete, ADF will become 1 and channel 1 (AN0) will again be selected and conversion will begin.

   At this time, if the ADIE bit is set to 1, ADI interrupt request will occur after completing A/D conversion.
5. Steps (2) to (4) will be repeated while ADST bit is set to 1.

   A/D conversion will stop when setting the ADST bit to 0. When setting the ADST bit to 1, A/D conversion will start again from channel 1 (AN0).

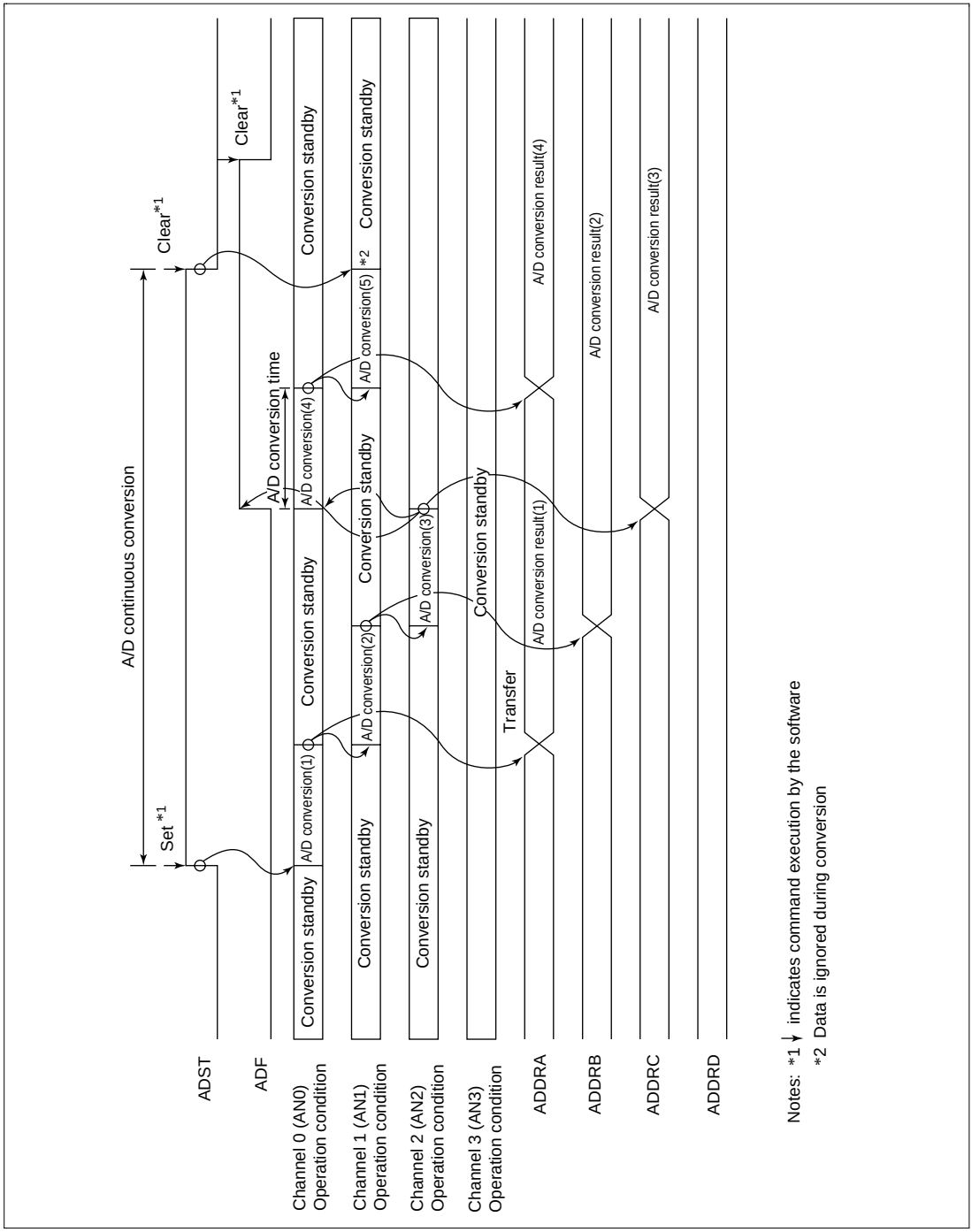


Figure 16.4 Operation Example of Mid-speed A/D Converter (Scan Mode, Three Channels Selected) (AN0–AN2)

16.4.3 Input Sampling and A/D Conversion Time

The mid-speed A/D converter is equipped with a sample and hold circuit. The mid-speed A/D converter samples input after t_D hours has elapsed since setting the ADST bit of the A/D control/status register (ADCSR) to 1, then begins conversion. The A/D conversion timing is shown in table 16.4.

The A/D conversion time, as shown in figure 16.5, includes both t_D and input sampling time. Here, t_D is determined by the write timing to ADCSR and is not constant. Thus the conversion time changes in the range shown in table 16.4.

The conversion time shown in table 16.4 is the time for the first conversion. For the second conversion and after, the time will be 256 state (fixed) for CKS=0 and 128 state (fixed) for CKS=1.

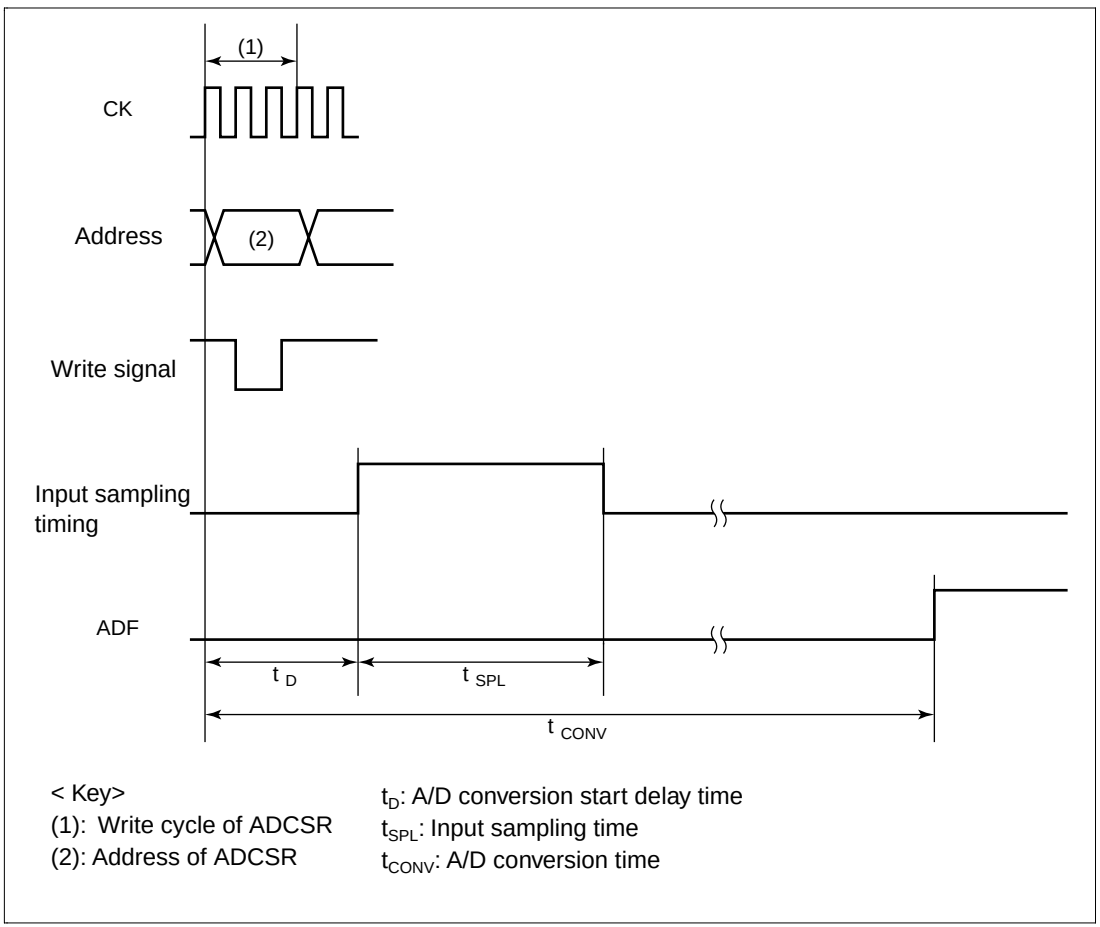


Figure 16.5 A/D Conversion Timing

Table 16.4 A/D Conversion Time (Single Mode)

		CKS=0			CKS=1		
	Notation	Min	Typ	Max	Min	Typ	Max
A/D conversion start delay time	t_D	10	—	17	6	—	9
Input sampling time	t_{SPL}	—	64	—	—	32	—
A/D conversion time	t_{CCNV}	259	—	266	131	—	134

Note: Numbers in the table are in states (t_{cyc}).

16.4.4 External Trigger Input Timing

It is possible to start A/D conversion from an external trigger input. External trigger input is input from the $\overline{\text{ADTRG}}$ pin or MTU when the TRGE bit of the A/D control register (ADCR) is set to 1.

A/D conversion is started when the ADST bit of the A/D control/status register (ADCSR) is set to 1 by the $\overline{\text{ADTRG}}$ input pin last transition edge or MTU trigger. Other operations, regardless of whether in the single or scan mode, are the same as when setting the ADST bit to 1 with the software.

Figure 16.6 shows an example of external trigger input timing.

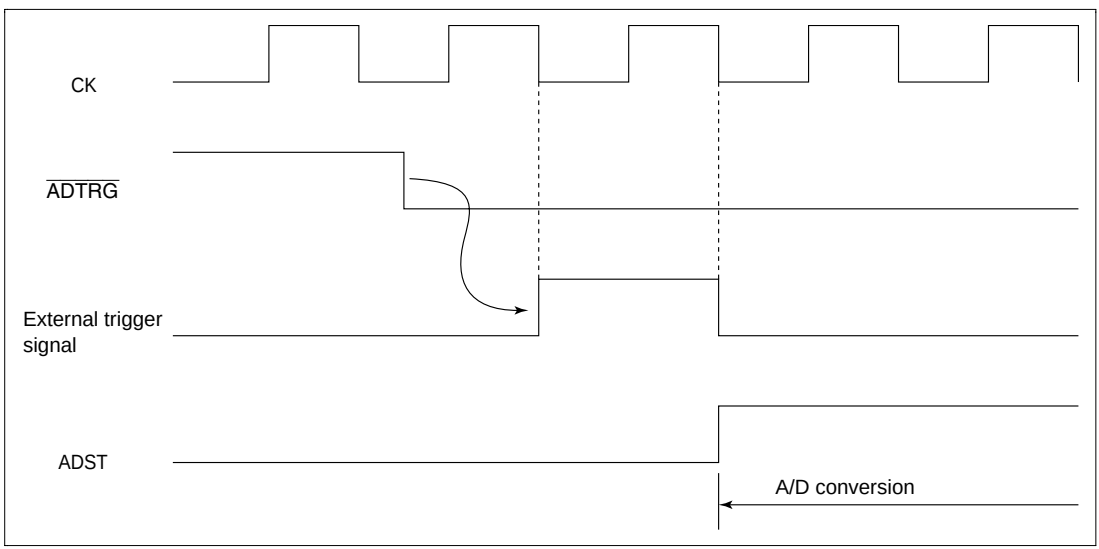


Figure 16.6 External Trigger Input Timing

16.5 Interrupt and DMA, DTC Transfer Requests

The mid-speed A/D converter generates A/D conversion complete interrupt when completing A/D conversion.

The ADI interrupt request can be enabled or disabled by the ADIE bit of ADCSR. It is also possible to activate DMA or DTC transfer by the ADI interrupt request. It is possible to activate DTC with ADI0 interrupt of A/D0 and activate DMAC with ADI1 interrupt of A/D1. Table 16.5 shows the interrupt factors of the mid-speed A/D converter.

Table 16.5 Mid-speed A/D Converter Interrupt Factors

Mid-speed A/D converter	Interrupt Factor	Content	DTC	DMAC
A/D0	ADI0	Interrupt by conversion complete	O	×
A/D1	ADI1		×	O

O: activation enabled
×: activation disabled

When accessing the A/D0 register with DTC activated by ADI0 interrupt, the ADF bit of the A/D0 control/status register (ADCSR0) will automatically be cleared to 0. Furthermore, it is possible to automatically clear the ADF bit of ADCSR1 by register access of A/D1 with activated DMAC of ADI1 interrupt. For details on the automatic clearing operation of this interrupt factor, see section 8, Data Transfer Controller (DTC).

16.6 A/D Conversion Precision Definitions

The medium-speed A/D converter converts analog values input from analog input channels to 10-bit digital values by comparing them with an analog reference voltage. In this operation, the absolute precision of the A/D conversion (i.e. the deviation between the input analog value and the output digital value) includes the following kinds of error.

(1) Offset error
(2) Full-scale error
(3) Quantization error
(4) Nonlinearity error

The above four kinds of error are described below with reference to figure 16.7. For the sake of clarity, this figure shows 3-bit medium-speed A/D conversion rather than 10-bit medium-speed A/D conversion. Offset error (see figure 16.7 (1)) is the deviation between the actual A/D conversion characteristic and the ideal A/D conversion characteristic when the digital output value changes from the minimum value (zero voltage) of 0000000000 (000 in the figure) to 0000000001 (001 in the figure). Full-scale error (see figure 16.7 (2)) is the deviation between the actual A/D conversion characteristic and the ideal A/D conversion characteristic when the digital output value changes from 1111111110 (110 in the figure) to the maximum value (full-scale voltage) of 111111111 (111 in the figure). Quantization error is the deviation inherent in the medium-speed A/D converter, given by 1/2 LSB (see figure 16.7 (3)). Nonlinearity error is the deviation between the actual A/D conversion characteristic and the ideal A/D conversion characteristic from zero voltage to full-scale voltage (see figure 16.7 (4)). This does not include offset error, full-scale error, and quantization error.

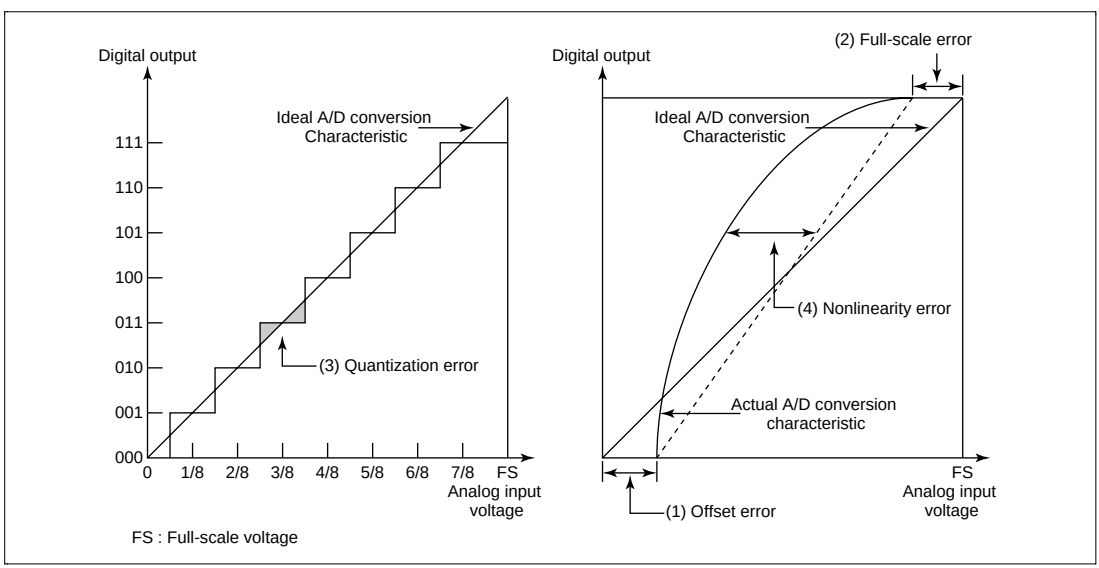


Figure 16.7 A/D Conversion Precision Definitions

16.7 Usage Notes

The following points should be noted when using the mid-speed A/D converter.

16.7.1 Analog Voltage Settings

(1) Analog input voltage range

The voltage applied to analog input pins during A/D conversion should be in the range AVSS ≤ ANn ≤ AV_{ref} (n = 0 to 7).

(2) AV_{CC} and AV_{SS} input voltages

For the AV_{CC} and AV_{SS} input voltages, set $AV_{CC} = V_{CC} \pm 10\%$ and $AV_{SS} = V_{SS}$. When the medium-speed A/D converter is not used, set $AV_{CC} = V_{CC}$ and $AV_{SS} = V_{SS}$.

(3) AVref input voltage

For the AV_{ref} pin input voltage analog reference, set $AV_{ref} \leq AV_{CC}$. When the medium-speed A/D converter is not used, set $AV_{ref} = AV_{CC}$.

(4) AV_{CC} and AV_{ref} must be connected to the power supply (V_{CC}) even if the medium-speed A/D converter is not used or is in standby mode.

16.7.2 Handling of Analog Input Pins

To prevent damage from surges and other abnormal voltages at the analog input pins (AN0-AN7), connect a protection circuit such as that shown in figure 16.8. This circuit also includes a CR filter function that suppresses error due to noise. The circuit shown here is only a design example; circuit constants must be decided on the basis of the actual operating conditions.

Figure 16.9 shows an equivalent circuit for the analog input pins, and table 16.6 summarizes the analog input pin specifications.

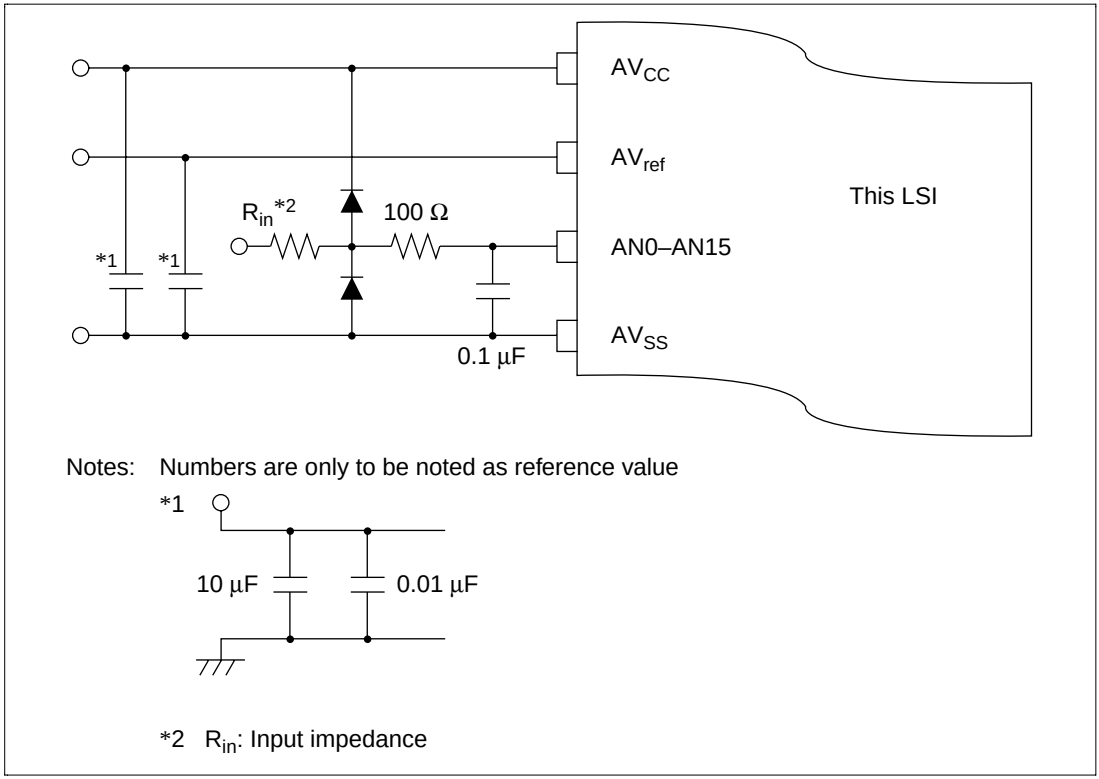


Figure 16.8 Example of Analog Input Pin Protection Circuit

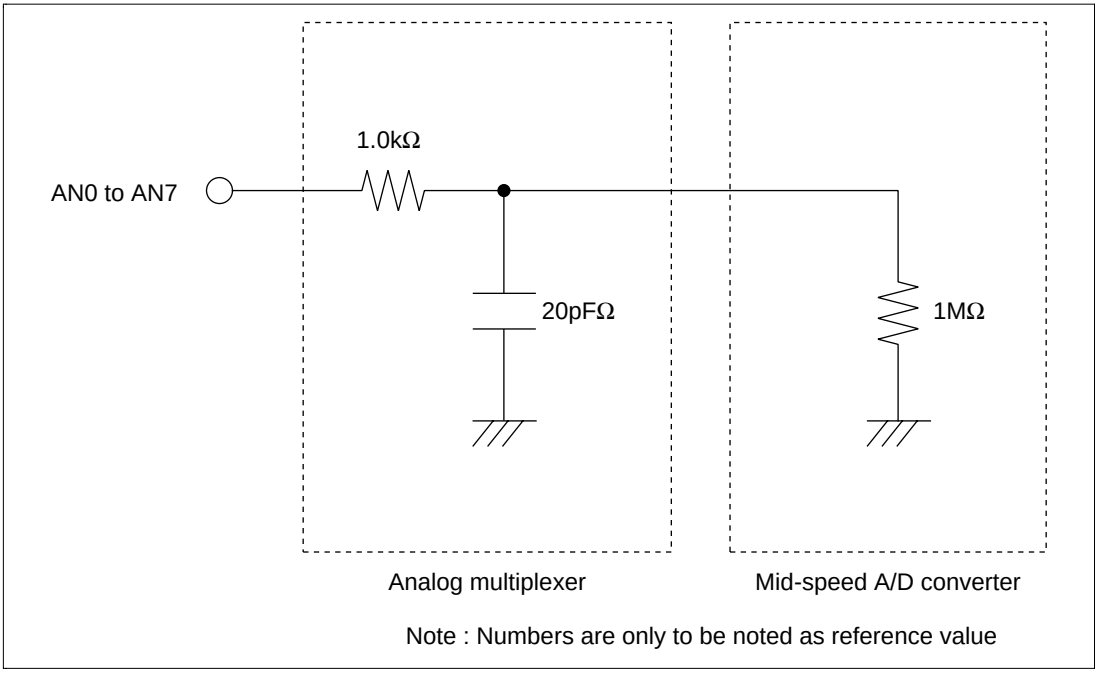


Figure 16.9 Equivalent Circuit for the Analog Input Pins

Table 16.6 Analog Pin Specifications

Item	Min	Max	Unit
Analog input capacitance	—	20	pF
Permissible signal source impedance	—	1	kΩ

Section 17 Compare Match Timer (CMT)

17.1 Overview

The SH7040 series has an on-chip compare match timer (CMT) configured of 16-bit timers for two channels. The CMT has 16-bit counters and can generate interrupts at set intervals.

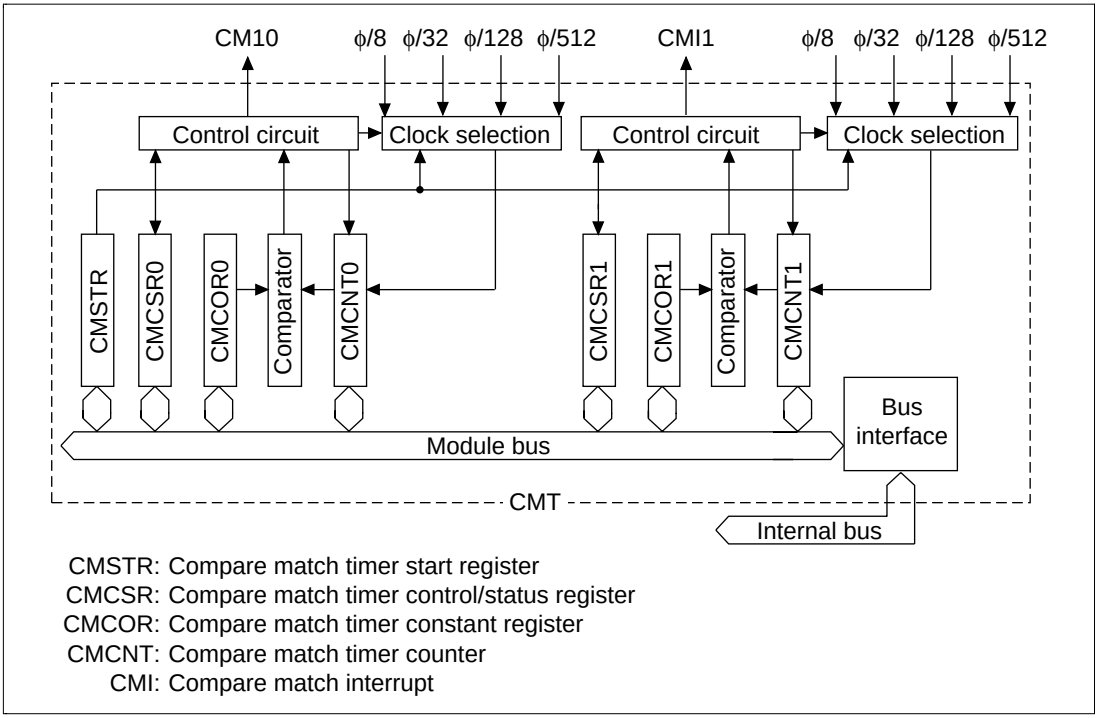
17.1.1 Features

The CMT has the following features:

- Four types of counter input clock can be selected
  - One of four internal clocks (ϕ/8, ϕ/32, ϕ/128, ϕ/512) can be selected independently for each channel.
- Interrupt sources
  - A compare match interrupt can be requested independently for each channel.

17.1.2 Block Diagram

Figure 17.1 shows a block diagram of the CMT.



CMSTR: Compare match timer start register
CMCSR: Compare match timer control/status register
CMCOR: Compare match timer constant register
CMCNT: Compare match timer counter
CMI: Compare match interrupt

Figure 17.1 CMT Block Diagram

17.1.3 Register Configuration

Table 17.1 summarizes the CMT register configuration.

Table 17.1 Register Configuration

Channel	Name	Abbreviation	R/W	Initial Value	Address	Access Size (Bits)
Shared	Compare match timer start register	CMSTR	R/W	H'0000	H'FFFF83D0	8, 16, 32
0	Compare match timer control/status register 0	CMCSR0	R/(W)*	H'0000	H'FFFF83D2	8, 16, 32
	Compare match timer counter 0	CMCNT0	R/W	H'0000	H'FFFF83D4	8, 16, 32
	Compare match timer constant register 0	CMCOR0	R/W	H'FFFF	H'FFFF83D6	8, 16, 32
1	Compare match timer control/status register 1	CMCSR1	R/(W)*	H'0000	H'FFFF83D8	8, 16, 32
	Compare match timer counter 1	CMCNT1	R/W	H'0000	H'FFFF83DA	8, 16, 32
	Compare match timer constant register 1	CMCOR1	R/W	H'FFFF	H'FFFF83DC	8, 16, 32

Note: * The only value that can be written to the CMCSR0 and CMCSR1 CMF bits is a 0 to clear the flags.

17.2 Register Descriptions

17.2.1 Compare Match Timer Start Register (CMSTR)

The compare match timer start register (CMSTR) is a 16-bit register that selects whether to operate or halt the channel 0 and channel 1 counters (CMCNT). It is initialized to H'0000 by power-on resets and by standby mode. Manual reset does not initialize CMSTR.

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	STR1	STR0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W

- Bits 15–2—Reserved: These bits always read as 0. The write value should always be 0.
- Bit 1—Count Start 1 (STR1): Selects whether to operate or halt compare match timer counter 1.

Bit 1: STR1	Description
0	CMCNT1 count operation halted (initial value)
1	CMCNT1 count operation

- Bit 0—Count Start 0 (STR0): Selects whether to operate or halt compare match timer counter 0.

Bit 0: STR0	Description
0	CMCNT0 count operation halted (initial value)
1	CMCNT0 count operation

17.2.2 Compare Match Timer Control/Status Register (CMCSR)

The compare match timer control/status register (CMCSR) is a 16-bit register that indicates the occurrence of compare matches, sets the enable/disable of interrupts, and establishes the clock used for incrementation. It is initialized to H'0000 by power-on resets and by standby mode. Manual reset does not initialize CMCSR.

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	CMF	CMIE	—	—	—	—	CKS1	CKS0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/W	R	R	R	R	R/W	R/W

Note: * The only value that can be written is a 0 to clear the flag.

- Bits 15–8 and 5–2—Reserved: These bits always read as 0. The write value should always be 0.

- Bit 7—Compare Match Flag (CMF): This flag indicates whether or not the CMCNT and CMCOR values have matched.

Bit 7: CMF	Description
0	CMCNT and CMCOR values have not matched (initial status) Clear condition: Write a 0 to CMF after reading a 1 from it
1	CMCNT and CMCOR values have matched

- Bit 6—Compare Match Interrupt Enable (CMIE): Selects whether to enable or disable a compare match interrupt (CMI) when the CMCNT and CMCOR values have matched (CMF = 1).

Bit 6: CMIE	Description
0	Compare match interrupts (CMI) disabled (initial status)
1	Compare match interrupts (CMI) enabled

- Bits 1, 0—Clock Select 1, 0 (CKS1, CKS0): These bits select the clock input to the CMCNT from among the four internal clocks obtained by dividing the system clock (ϕ). When the STR bit of the CMSTR is set to 1, the CMCNT begins incrementing with the clock selected by CKS1 and CKS0.

Bit 1: CKS1	Bit 0: CKS0	Description
0	0	ϕ/8 (initial status)
	1	ϕ/32
1	0	ϕ/128
	1	ϕ/512

17.2.3 Compare Match Timer Counter (CMCNT)

The compare match timer counter (CMCNT) is a 16-bit register used as an upcounter for generating interrupt requests.

When an internal clock is selected with the CKS1, CKS0 bits of the CMCSR register and the STR bit of the CMSTR is set to 1, the CMCNT begins incrementing with that clock. When the CMCNT value matches that of the compare match timer constant register (CMCOR), the CMCNT is cleared to H'0000 and the CMF flag of the CMCSR is set to 1. If the CMIE bit of the CMCSR is set to 1 at this time, a compare match interrupt (CMI) is requested.

The CMCNT is initialized to H'0000 by power-on resets and by standby mode. Manual reset does not initialize CMCNT.

Bit:	15	14	13	12	11	10	9	8
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

17.2.4 Compare Match Timer Constant Register (CMCOR)

The compare match timer constant register (CMCOR) is a 16-bit register that sets the compare match period with the CMCNT.

The CMCOR is initialized to H'FFFF by power-on resets and by standby mode. There is no initializing with manual reset.

Bit:	15	14	13	12	11	10	9	8
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

17.3 Operation

17.3.1 Period Count Operation

When an internal clock is selected with the CKS1, CKS0 bits of the CMCSR register and the STR bit of the CMSTR is set to 1, the CMCNT begins incrementing with the selected clock. When the CMCNT counter value matches that of the compare match constant register (CMCOR), the CMCNT counter is cleared to H'0000 and the CMF flag of the CMCSR register is set to 1. If the CMIE bit of the CMCSR register is set to 1 at this time, a compare match interrupt (CMI) is requested. The CMCNT counter begins counting up again from H'0000.

Figure 17.2 shows the compare match counter operation.

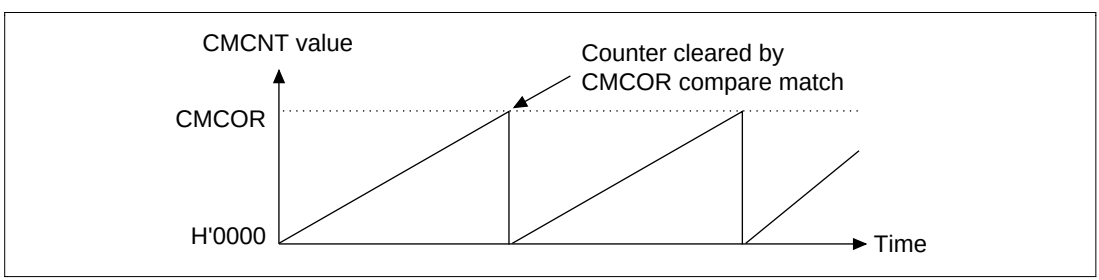


Figure 17.2 Counter Operation

17.3.2 CMCNT Count Timing

One of four clocks (ϕ/8, ϕ/32, ϕ/128, ϕ/512) obtained by dividing the system clock (CK) can be selected by the CKS1, CKS0 bits of the CMCSR. Figure 17.3 shows the timing.

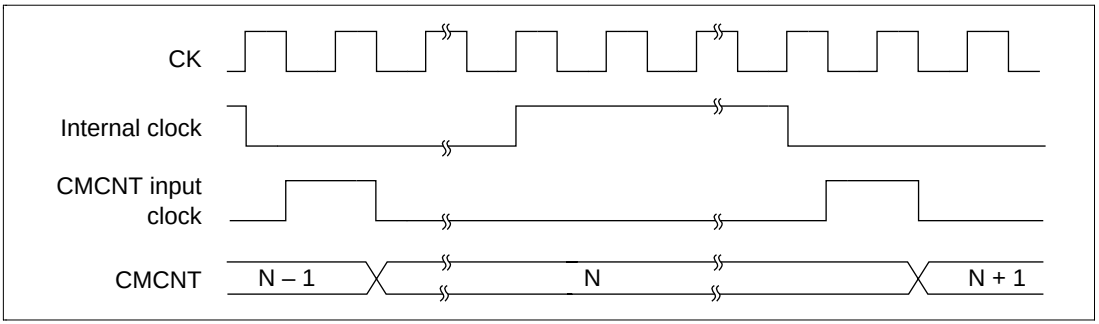


Figure 17.3 Count Timing

17.4 Interrupts

17.4.1 Interrupt Sources and DTC Activation

The CMT has a compare match interrupt for each channel, with independent vector addresses allocated to each of them. The corresponding interrupt request is output when the interrupt request flag CMF is set to 1 and the interrupt enable bit CMIE has also been set to 1.

When activating CPU interrupts by interrupt request, the priority between the channels can be changed by using the interrupt controller settings. See section 6, Interrupt Controller (INTC), for details.

Interrupt requests can also be used as data transfer controller (DTC) activating sources. In this case, channel priorities are fixed. See section 8, Data Transfer Controller (DTC), for details.

17.4.2 Compare Match Flag Set Timing

The CMF bit of the CMCSR register is set to 1 by the compare match signal generated when the CMCOR register and the CMCNT counter match. The compare match signal is generated upon the final state of the match (timing at which the CMCNT counter matching count value is updated). Consequently, after the CMCOR register and the CMCNT counter match, a compare match signal will not be generated until a CMCNT counter input clock occurs. Figure 17.4 shows the CMF bit set timing.

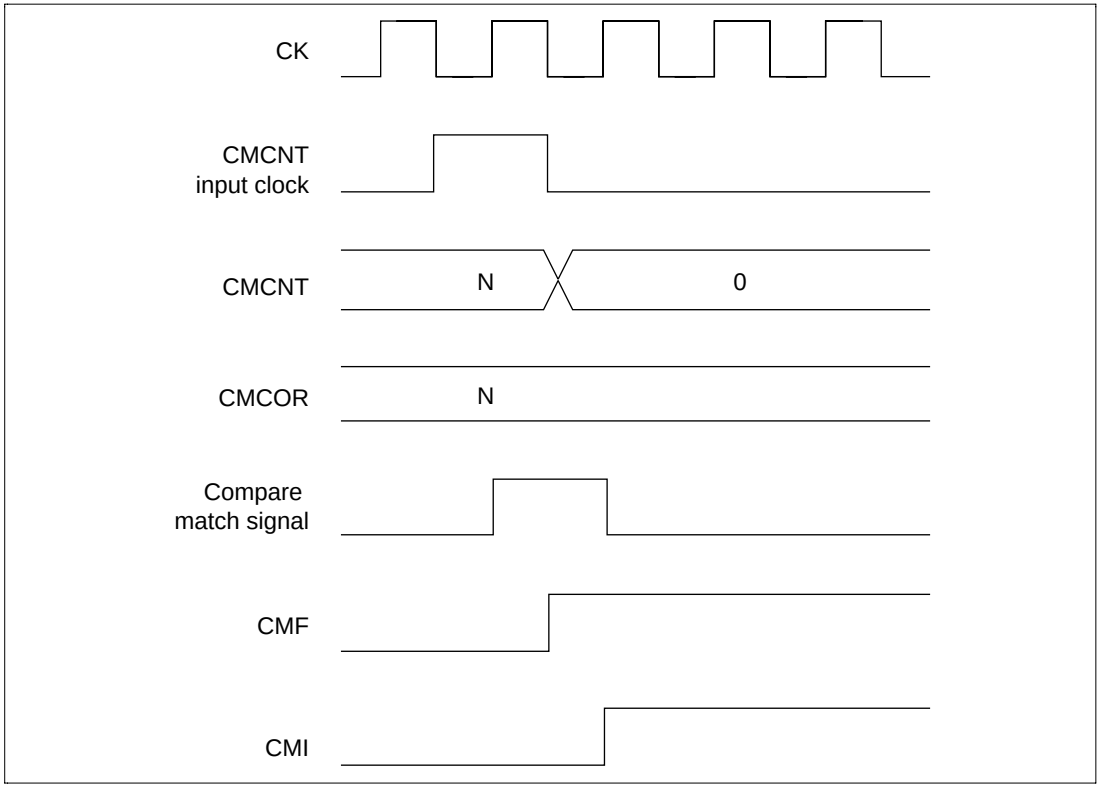


Figure 17.4 CMF Set Timing

17.4.3 Compare Match Flag Clear Timing

The CMF bit of the CMCSR register is cleared either by writing a 0 to it after reading a 1, or by a clear signal after a DTC transfer. Figure 17.5 shows the timing when the CMF bit is cleared by the CPU.

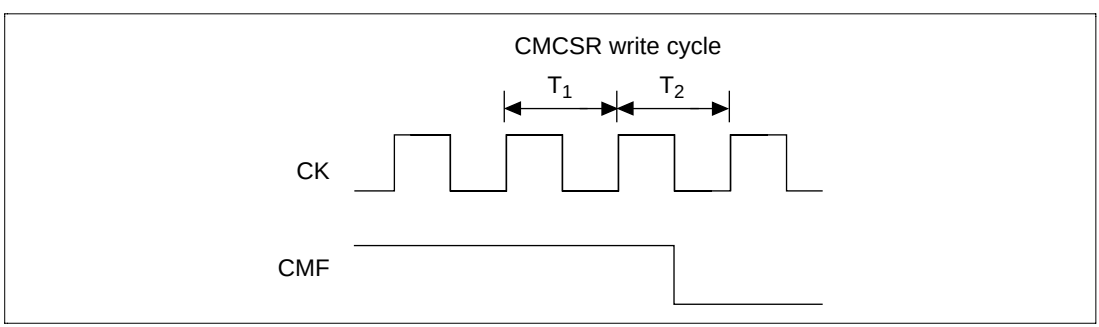


Figure 17.5 Timing of CMF Clear by the CPU

17.5 Notes on Use

Take care that the contentions described in sections 17.5.1–17.5.3 do not arise during CMT operation.

17.5.1 Contention between CMCNT Write and Compare Match

If a compare match signal is generated during the T_2 state of the CMCNT counter write cycle, the CMCNT counter clear has priority, so the write to the CMCNT counter is not performed. Figure 17.6 shows the timing.

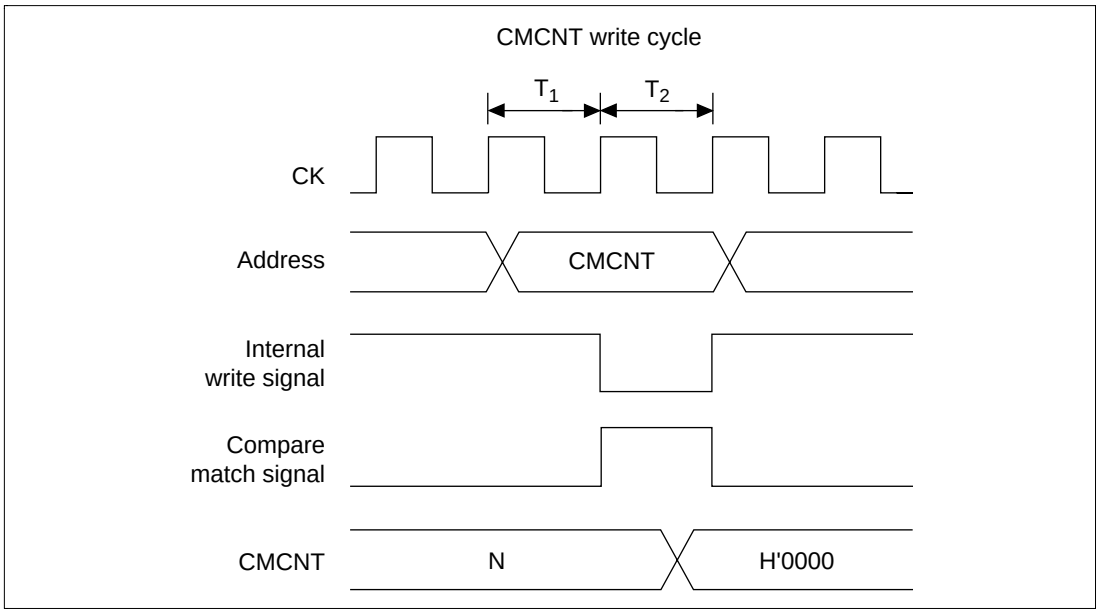


Figure 17.6 CMCNT Write and Compare Match Contention

17.5.2 Contention between CMCNT Word Write and Incrementation

If an increment occurs during the T_2 state of the CMCNT counter word write cycle, the counter write has priority, so no increment occurs. Figure 17.7 shows the timing.

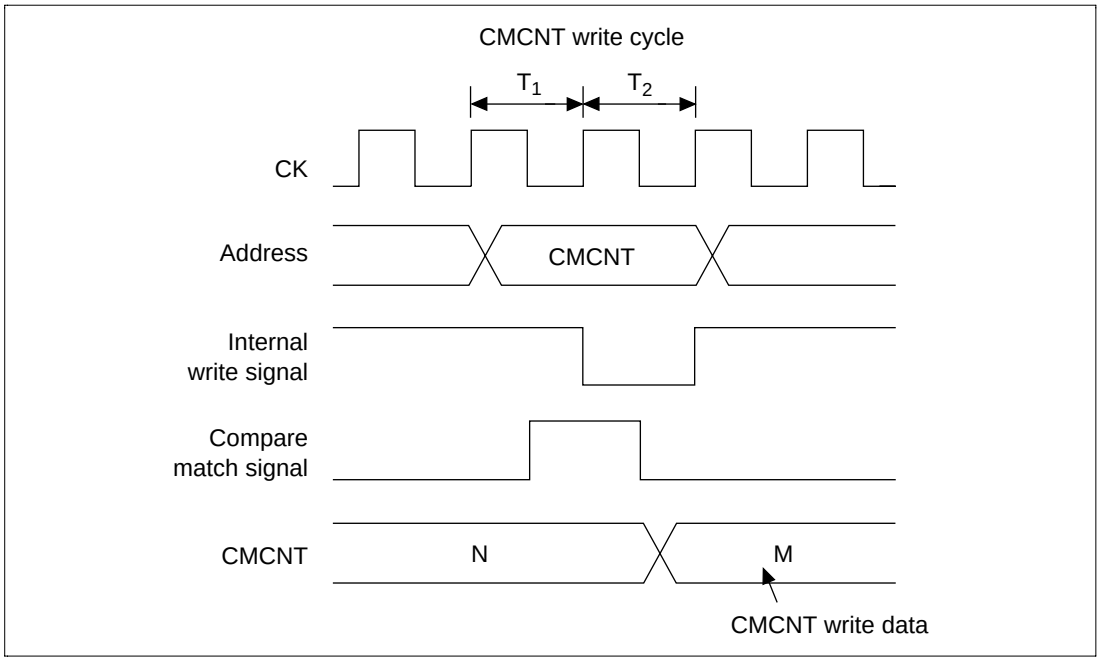


Figure 17.7 CMCNT Word Write and Increment Contention

17.5.3 Contention between CMCNT Byte Write and Incrementation

If an increment occurs during the T_2 state of the CMCNT byte write cycle, the counter write has priority, so no increment of the write data results on the writing side. The byte data on the side not performing the writing is also not incremented, so the contents are those before the write.

Figure 17.8 shows the timing when an increment occurs during the T_2 state of the CMCNTH write cycle.

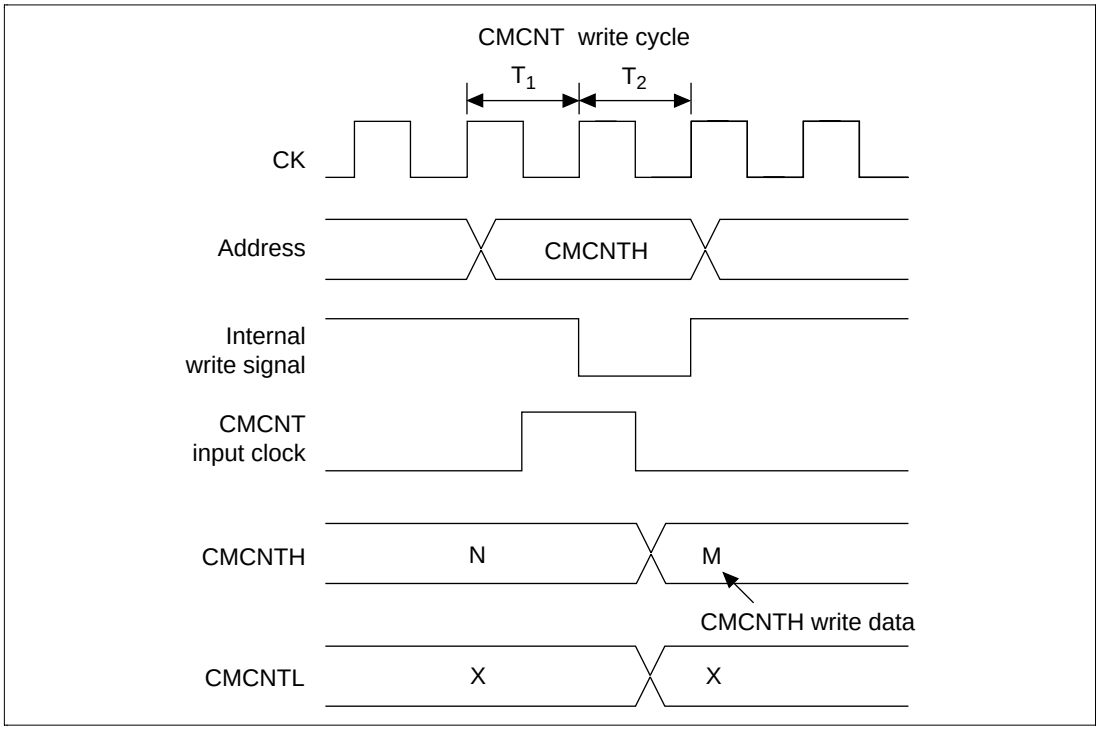


Figure 17.8 CMCNT Byte Write and Increment Contention

18.1 Overview

The pin function controller (PFC) is composed of registers for selecting the function of multiplexed pins and the direction of input/output. Table 18.1 lists the SH7040 Series's multiplexed pins. The multiplex pin functions have restrictions dependent on the operating mode. Table 18.2 lists the pin functions and initial values for each operating mode.

Table 18.1 Multiplexed Pins

Port	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)	FP-112	FP-144	TFP-120
A	PA23 I/O (port)	$\overline{\text{WRHH}}$ output (BSC)	—	—	—	1	—
	PA22 I/O (port)	$\overline{\text{WRHL}}$ output (BSC)	—	—	—	3	—
	PA21 I/O (port)	$\overline{\text{CASHH}}$ output (BSC)	—	—	—	4	—
	PA20 I/O (port)	$\overline{\text{CASHL}}$ output (BSC)	—	—	—	29	—
	PA19 I/O (port)	$\overline{\text{BACK}}$ output (BSC)	DRAK1 output (DMAC)	—	—	30	—
	PA18 I/O (port)	$\overline{\text{BREQ}}$ input (BSC)	DRAK0 output (DMAC)	—	—	33	—
	PA17 /O (port)	$\overline{\text{WAIT}}$ input (BSC)	—	—	—	101	—
	PA16 I/O (port)	$\overline{\text{AH}}$ output (BSC)	—	—	—	100	—
	PA15 I/O (port)	CK output (CPG)	—	—	83	107	88
	PA14 I/O (port)	$\overline{\text{RD}}$ output (BSC)	—	—	34	43	37
	PA13 I/O (port)	$\overline{\text{WRH}}$ output (BSC)	—	—	36	47	39
	PA12 I/O (port)	$\overline{\text{WRL}}$ output (BSC)	—	—	38	48	41
	PA11 I/O (port)	$\overline{\text{CS1}}$ output (BSC)	—	—	40	49	43
	PA10 I/O (port)	$\overline{\text{CS0}}$ output (BSC)	—	—	41	50	44
	PA9 I/O (port)	TCLKD input (MTU)	$\overline{\text{IRQ3}}$ (INTC)	—	42	51	45
	PA8 I/O (port)	TCLKC input (MTU)	$\overline{\text{IRQ2}}$ (INTC)	—	43	52	46
	PA7 I/O (port)	TCLKB input (MTU)	$\overline{\text{CS3}}$ output (BSC)	—	44	53	47
	PA6 I/O (port)	TCLKA input (MTU)	$\overline{\text{CS2}}$ output (BSC)	—	45	54	48
	PA5 I/O (port)	SCK1 I/O (SCI)	$\overline{\text{DREQ1}}$ input (DMAC)	$\overline{\text{IRQ1}}$ input (INTC)	46	136	49

Table 18.1 Multiplexed Pins (cont)

Port	Function 1 (Re-lated Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)	FP-112	FP-144	TFP-120
A	PA4 I/O (port)	TxD1 output (SCI)	—	—	47	134	50
	PA3 I/O (port)	RxD1 input (SCI)	—	—	48	133	51
	PA2 I/O (port)	SCK0 I/O (SCI)	$\overline{\text{DREQ0}}$ input (DMAC)	$\overline{\text{IRQ0}}$ input (INTC)	49	132	52
	PA1 I/O (port)	TxD0 output (SCI)	—	—	50	131	53
	PA0 I/O (port)	RxD0 input (SCI)	—	—	51	130	54
B	PB9 I/O (port)	$\overline{\text{IRQ7}}$ input (INTC)	A21 output (BSC)	$\overline{\text{ADTRG}}$ input (A/D)	32	41	35
	PB8 I/O (port)	$\overline{\text{IRQ6}}$ input (INTC)	A20 output (BSC)	$\overline{\text{WAIT}}$ input (BSC)	31	39	34
	PB7 I/O (port)	$\overline{\text{IRQ5}}$ input (INTC)	A19 output (BSC)	$\overline{\text{BREQ}}$ input (BSC)	30	38	33
	PB6 I/O (port)	$\overline{\text{IRQ4}}$ input (INTC)	A18 output (BSC)	$\overline{\text{BACK}}$ output (BSC)	29	37	32
	PB5 I/O (port)	$\overline{\text{IRQ3}}$ input (INTC)	$\overline{\text{POE3}}$ input (port)	RDWR output (BSC)	28	36	29
	PB4 I/O (port)	$\overline{\text{IRQ2}}$ input (INTC)	$\overline{\text{POE2}}$ input (port)	$\overline{\text{CASH}}$ output (BSC)	26	34	27
	PB3 I/O (port)	$\overline{\text{IRQ1}}$ input (INTC)	$\overline{\text{POE1}}$ input (port)	$\overline{\text{CASL}}$ output (BSC)	25	32	26
	PB2 I/O (port)	$\overline{\text{IRQ0}}$ input (INTC)	$\overline{\text{POE0}}$ input (port)	$\overline{\text{RAS}}$ output (BSC)	24	31	25
	PB1 I/O (port)	A17 input (BSC)	—	—	22	27	23
	PB0 I/O (port)	A16 output (BSC)	—	—	20	25	21
C	PC15 I/O (port)	A15 output (BSC)	—	—	19	24	20
	PC14 I/O (port)	A14 output (BSC)	—	—	18	23	19
	PC13 I/O (port)	A13 output (BSC)	—	—	17	22	18
	PC12 I/O (port)	A12 output (BSC)	—	—	16	21	17
	PC11 I/O (port)	A11 output (BSC)	—	—	15	20	16
	PC10 I/O (port)	A10 output (BSC)	—	—	14	19	15
	PC9 I/O (port)	A9 output (BSC)	—	—	13	18	14
	PC8 I/O (port)	A8 output (BSC)	—	—	12	17	13
	PC7 I/O (port)	A7 output (BSC)	—	—	11	16	12

Table 18.1 Multiplexed Pins (cont)

Port	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)	FP-112	FP-144	TFP-120
C	PC6 I/O (port)	A6 output (BSC)	—	—	10	15	11
	PC5 I/O (port)	A5 output (BSC)	—	—	9	13	10
	PC4 I/O (port)	A4 output (BSC)	—	—	8	11	9
	PC3 I/O (port)	A3 output (BSC)	—	—	7	10	8
	PC2 I/O (port)	A2 output (BSC)	—	—	6	9	7
	PC1 I/O (port)	A1 output (BSC)	—	—	5	8	6
	PC0 I/O (port)	A0 output (BSC)	—	—	4	7	5
D	PD31 I/O (port)	D31 I/O (BSC)	$\overline{\text{ADTRG}}$ input (A/D)	—	—	45	—
	PD30 I/O (port)	D30 I/O (BSC)	$\overline{\text{IRQOUT}}$ output (INTC)	—	—	46	—
	PD29 I/O (port)	D29 I/O (BSC)	$\overline{\text{CS3}}$ output (BSC)	—	—	56	—
	PD28 I/O (port)	D28 I/O (BSC)	$\overline{\text{CS2}}$ output (BSC)	—	—	57	—
	PD27 I/O (port)	D27 I/O (BSC)	DACK1 output (DMAC)	—	—	58	—
	PD26 I/O (port)	D26 I/O (BSC)	DACK0 output (DMAC)	—	—	59	—
	PD25 I/O (port)	D25 I/O (BSC)	$\overline{\text{DREQ1}}$ input (DMAC)	—	—	60	—
	PD24 I/O (port)	D24 I/O (BSC)	$\overline{\text{DREQ0}}$ input (DMAC)	—	—	62	—
	PD23 I/O (port)	D23 I/O (BSC)	$\overline{\text{IRQ7}}$ input (INTC)	—	—	64	—
	PD22 I/O (port)	D22 I/O (BSC)	$\overline{\text{IRQ6}}$ input (INTC)	—	—	65	—
	PD21 I/O (port)	D21 I/O (BSC)	$\overline{\text{IRQ5}}$ input (INTC)	—	—	66	—
	PD20 I/O (port)	D20 I/O (BSC)	$\overline{\text{IRQ4}}$ input (INTC)	—	—	67	—
	PD19 I/O (port)	D19 I/O (BSC)	$\overline{\text{IRQ3}}$ input (INTC)	—	—	68	—
	PD18 I/O (port)	D18 I/O (BSC)	$\overline{\text{IRQ2}}$ input (INTC)	—	—	69	—
	PD17 I/O (port)	D17 I/O (BSC)	$\overline{\text{IRQ1}}$ input (INTC)	—	—	70	—
	PD16 I/O (port)	D16 I/O (BSC)	$\overline{\text{IRQ0}}$ input (INTC)	—	—	72	—
	PD15 I/O (port)	D15 I/O (BSC)	—	—	52	73	55
	PD14 I/O (port)	D14 I/O (BSC)	—	—	53	74	56

Table 18.1 Multiplexed Pins (cont)

Port	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)	FP-112	FP-144	TFP-120
D	PD13 I/O (port)	D13 I/O (BSC)	—	—	54	75	57
	PD12 I/O (port)	D12 I/O (BSC)	—	—	56	76	59
	PD11 I/O (port)	D11 I/O (BSC)	—	—	57	78	62
	PD10 I/O (port)	D10 I/O (BSC)	—	—	58	80	63
	PD9 I/O (port)	D9 I/O (BSC)	—	—	59	81	64
	PD8 I/O (port)	D8 I/O (BSC)	—	—	60	82	65
	PD7 I/O (port)	D7 I/O (BSC)	—	—	62	83	67
	PD6 I/O (port)	D6 I/O (BSC)	—	—	63	84	68
	PD5 I/O (port)	D5 I/O (BSC)	—	—	64	86	69
	PD4 I/O (port)	D4 I/O (BSC)	—	—	66	88	71
	PD3 I/O (port)	D3 I/O (BSC)	—	—	67	89	72
	PD2 I/O (port)	D2 I/O (BSC)	—	—	68	90	73
	PD1 I/O (port)	D1 I/O (BSC)	—	—	69	91	74
	PD0 I/O (port)	D0 I/O (BSC)	—	—	70	92	75
E	PE15 I/O (port)	TIOC4D I/O (MTU)	DACK1 output (DMAC)	$\overline{\text{IRQOUT}}$ output (INTC)	2	5	3
	PE14 I/O (port)	TIOC4C I/O (MTU)	DACK0 output (DMAC)	$\overline{\text{AH}}$ output (BSC)	1	2	2
	PE13 I/O (port)	TIOC4B I/O (MTU)	$\overline{\text{MRES}}$ input (INTC)	—	112	144	120
	PE12 I/O (port)	TIOC4A I/O (MTU)	—	—	111	143	119
	PE11 I/O (port)	TIOC3D I/O (MTU)	—	—	110	142	118
	PE10 I/O (port)	TIOC3C I/O (MTU)	—	—	108	140	116
	PE9 I/O (port)	TIOC3B I/O (MTU)	—	—	107	139	115
	PE8 I/O (port)	TIOC3A I/O (MTU)	—	—	106	138	114
	PE7 I/O (port)	TIOC2B I/O (MTU)	—	—	105	137	113
	PE6 I/O (port)	TIOC2A I/O (MTU)	—	—	104	116	112
	PE5 I/O (port)	TIOC1B I/O (MTU)	—	—	102	115	109
	PE4 I/O (port)	TIOC1A I/O (MTU)	—	—	89	114	96

Table 18.1 Multiplexed Pins (cont)

Port	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)	FP-112	FP-144	TFP-120
E	PE3 I/O (port)	TIOC0D I/O (MTU)	DRAK1 output (DMAC)	—	88	113	95
	PE2 I/O (port)	TIOC0C I/O (MTU)	$\overline{\text{DREQ1}}$ input (DMAC)	—	87	111	94
	PE1 I/O (port)	TIOC0B I/O (MTU)	DRAK0 output (DMAC)	—	86	110	93
	PE0 I/O (port)	TIOC0A I/O (MTU)	$\overline{\text{DREQ0}}$ input (DMAC)	—	85	109	92
F	PF7 input (port)	AN7 input (A/D)	—	—	99	126	106
	PF6 input (port)	AN6 input (A/D)	—	—	98	125	105
	PF5 input (port)	AN5 input (A/D)	—	—	96	123	103
	PF4 input (port)	AN4 input (A/D)	—	—	95	122	102
	PF3 input (port)	AN3 input (A/D)	—	—	94	121	101
	PF2 input (port)	AN2 input (A/D)	—	—	93	120	100
	PF1 input (port)	AN1 input (A/D)	—	—	92	119	99
	PF0 input (port)	AN0 input (A/D)	—	—	91	118	98

Table 18.2 Pin Arrangement by Mode

Pin NO.			Pin Name								
			On-Chip ROM Disabled				On-Chip ROM Enabled				
			MPU Mode0		MPU Mode 1		MPU Mode2		Single Chip Mode		
TFP120	FP144	FP112	Initial Function	PFC Selected Function Possibilities	Initial Function	PFC Selected Function Possibilities	Initial Function	PFC Selected Function Possibilities	Initial Function	PFC Selected Function Possibilities	PROM Mode
22, 40, 70, 82, 111	12,26,40,63 77,85,112 135	21,37,65 103	Vcc	Vcc	Vcc	Vcc	Vcc	Vcc	Vcc	Vcc	Vcc
4, 24, 28, 36, 42, 58, 66, 76, 97, 108, 117	6,14,28,35 42,55,61,71 79,87,93 117,129,141	3,23,27,33 39,55,61,71 90,101,109	Vss	Vss	Vss	Vss	Vss	Vss	Vss	Vss	Vss
75	92	70	D0	D0	D0	D0	PD0	PD0/D0	PD0	PD0	D0
74	91	69	D1	D1	D1	D1	PD1	PD1/D1	PD1	PD1	D1
73	90	68	D2	D2	D2	D2	PD2	PD2/D2	PD2	PD2	D2
72	89	67	D3	D3	D3	D3	PD3	PD3/D3	PD3	PD3	D3
71	88	66	D4	D4	D4	D4	PD4	PD4/D4	PD4	PD4	D4
69	86	64	D5	D5	D5	D5	PD5	PD5/D5	PD5	PD5	D5
68	84	63	D6	D6	D6	D6	PD6	PD6/D6	PD6	PD6	D6
67	83	62	D7	D7	D7	D7	PD7	PD7/D7	PD7	PD7	D7
65	82	60	D8	D8	D8	D8	PD8	PD8/D8	PD8	PD8	NC
64	81	59	D9	D9	D9	D9	PD9	PD9/D9	PD9	PD9	NC
63	80	58	D10	D10	D10	D10	PD10	PD10/D10	PD10	PD10	NC
62	78	57	D11	D11	D11	D11	PD11	PD11/D11	PD11	PD11	NC
59	76	56	D12	D12	D12	D12	PD12	PD12/D12	PD12	PD12	NC
57	75	54	D13	D13	D13	D13	PD13	PD13/D13	PD13	PD13	NC
56	74	53	D14	D14	D14	D14	PD14	PD14/D14	PD14	PD14	NC
55	73	52	D15	D15	D15	D15	PD15	PD15/D15	PD15	PD15	NC
—	72	—	PD16	PD16/D16/$\overline{\text{IRQ0}}$	D16	D16	PD16	PD16/D16/$\overline{\text{IRQ0}}$	PD16	PD16/$\overline{\text{IRQ0}}$	NC
—	70	—	PD17	PD17/D17/$\overline{\text{IRQ1}}$	D17	D17	PD17	PD17/D17/$\overline{\text{IRQ1}}$	PD17	PD17/$\overline{\text{IRQ1}}$	NC
—	69	—	PD18	PD18/D18/$\overline{\text{IRQ2}}$	D18	D18	PD18	PD18/D18/$\overline{\text{IRQ2}}$	PD18	PD18/$\overline{\text{IRQ2}}$	NC
—	68	—	PD19	PD19/D19/$\overline{\text{IRQ3}}$	D19	D19	PD19	PD19/D19/$\overline{\text{IRQ3}}$	PD19	PD19/$\overline{\text{IRQ3}}$	NC
—	67	—	PD20	PD20/D20/$\overline{\text{IRQ4}}$	D20	D20	PD20	PD20/D20/$\overline{\text{IRQ4}}$	PD20	PD20/$\overline{\text{IRQ4}}$	NC
—	66	—	PD21	PD21/D21/$\overline{\text{IRQ5}}$	D21	D21	PD21	PD21/D21/$\overline{\text{IRQ5}}$	PD21	PD21/$\overline{\text{IRQ5}}$	NC
—	65	—	PD22	PD22/D22/$\overline{\text{IRQ6}}$	D22	D22	PD22	PD22/D22/$\overline{\text{IRQ6}}$	PD22	PD22/$\overline{\text{IRQ6}}$	NC
—	64	—	PD23	PD23/D23/$\overline{\text{IRQ7}}$	D23	D23	PD23	PD23/D23/$\overline{\text{IRQ7}}$	PD23	PD23/$\overline{\text{IRQ7}}$	NC
—	62	—	PD24	PD24/D24/$\overline{\text{DREQ0}}$	D24	D24	PD24	PD24/D24/$\overline{\text{DREQ0}}$	PD24	PD24	NC
—	60	—	PD25	PD25/D25/$\overline{\text{DREQ1}}$	D25	D25	PD25	PD25/D25/$\overline{\text{DREQ1}}$	PD25	PD25	NC
—	59	—	PD26	PD26/D26/DACK0	D26	D26	PD26	PD26/D26/DACK0	PD26	PD26	NC
—	58	—	PD27	PD27/D27/DACK1	D27	D27	PD27	PD27/D27/DACK1	PD27	PD27	NC
—	57	—	PD28	PD28/D28/$\overline{\text{CS2}}$	D28	D28	PD28	PD28/D28/$\overline{\text{CS2}}$	PD28	PD28	NC
—	56	—	PD29	PD29/D29/$\overline{\text{CS3}}$	D29	D29	PD29	PD29/D29/$\overline{\text{CS3}}$	PD29	PD29	NC
—	46	—	PD30	PD30/D30/$\overline{\text{IRQOUT}}$	D30	D30	PD30	PD30/D30/$\overline{\text{IRQOUT}}$	PD30	PD30	NC
—	45	—	PD31	PD31/D31/$\overline{\text{ADTRG}}$	D31	D31	PD31	PD31/D31/$\overline{\text{ADTRG}}$	PD31	PD31/$\overline{\text{ADTRG}}$	NC
5	7	4	A0	A0	A0	A0	PC0	PC0/A0	PC0	PC0	A0
6	8	5	A1	A1	A1	A1	PC1	PC1/A1	PC1	PC1	A1
7	9	6	A2	A2	A2	A2	PC2	PC2/A2	PC2	PC2	A2
8	10	7	A3	A3	A3	A3	PC3	PC3/A3	PC3	PC3	A3
9	11	8	A4	A4	A4	A4	PC4	PC4/A4	PC4	PC4	A4
10	13	9	A5	A5	A5	A5	PC5	PC5/A5	PC5	PC5	A5
11	15	10	A6	A6	A6	A6	PC6	PC6/A6	PC6	PC6	A6
12	16	11	A7	A7	A7	A7	PC7	PC7/A7	PC7	PC7	A7
13	17	12	A8	A8	A8	A8	PC8	PC8/A8	PC8	PC8	A8
14	18	13	A9	A9	A9	A9	PC9	PC9/A9	PC9	PC9	NC
15	19	14	A10	A10	A10	A10	PC10	PC10/A10	PC10	PC10	A10
16	20	15	A11	A11	A11	A11	PC11	PC11/A11	PC11	PC11	A11
17	21	16	A12	A12	A12	A12	PC12	PC12/A12	PC12	PC12	A12

Table 18.2 Pin Arrangement by Mode (cont)

Pin NO.			Pin Name								
			On-Chip ROM Disabled				On-Chip ROM Enabled				
			MPU Mode0		MPU Mode 1		MPU Mode2		Single Chip Mode		
TFP120	FP144	FP112	Initial Function	PFC Selected Function Possiblilities	Initial Function	PFC Selected Function Possiblilities	Initial Function	PFC Selected Function Possiblilities	Initial Function	PFC Selected Function Possiblilities	PROM Mode
18	22	17	A13	A13	A13	A13	PC13	PC13/A13	PC13	PC13	A13
19	23	18	A14	A14	A14	A14	PC14	PC14/A14	PC14	PC14	A14
20	24	19	A15	A15	A15	A15	PC15	PC15/A15	PC15	PC15	A15
21	25	20	A16	A16	A16	A16	PB0	PB0/A16	PB0	PB0	A16
23	27	22	A17	A17	A17	A17	PB1	PB1/A17	PB1	PB1	NC
25	31	24	PB2	PB2/IRQ0/POE0/RAS	PB2	PB2/IRQ0/POE0/RAS	PB2	PB2/IRQ0/POE0/RAS	PB2	PB2/IRQ0/POE0	NC
26	32	25	PB3	PB3/IRQ1/POE1/CASL	PB3	PB3/IRQ1/POE1/CASL	PB3	PB3/IRQ1/POE1/CASL	PB3	PB3/IRQ1/POE1	OE
27	34	26	PB4	PB4/IRQ2/POE2/CASH	PB4	PB4/IRQ2/POE2/CASH	PB4	PB4/IRQ2/POE2/CASH	PB4	PB4/IRQ2/POE2	PGM
29	36	28	PB5	PB5/IRQ3/POE3/RDWR	PB5	PB5/IRQ3/POE3/RDWR	PB5	PB5/IRQ3/POE3/RDWR	PB5	PB5/IRQ3/POE3	NC
32	37	29	PB6	PB6/IRQ4/A18/BACK	PB6	PB6/IRQ4/A18/BACK	PB6	PB6/IRQ4/A18/BACK	PB6	PB6/IRQ4	NC
33	38	30	PB7	PB7/IRQ5/A19/BREQ	PB7	PB7/IRQ5/A19/BREQ	PB7	PB7/IRQ5/A19/BREQ	PB7	PB7/IRQ5	NC
34	39	31	PB8	PB8/IRQ6/A20/WAIT	PB8	PB8/IRQ6/A20/WAIT	PB8	PB8/IRQ6/A20/WAIT	PB8	PB8/IRQ6	NC
35	41	32	PB9	PB9/IRQ7/A21/ADTRG	PB9	PB9/IRQ7/A21/ADTRG	PB9	PB9/IRQ7/A21/ADTRG	PB9	PB9/IRQ7/ADTRG	NC
54	130	51	PA0	PA0/RXD0	PA0	PA0/RXD0	PA0	PA0/RXD0	PA0	PA0/RXD0	NC
53	131	50	PA1	PA1/TXD0	PA1	PA1/TXD0	PA1	PA1/TXD0	PA1	PA1/TXD0	NC
52	132	49	PA2	PA2/SCK0/DREQ0/IRQ0	PA2	PA2/SCK0/DREQ0/IRQ0	PA2	PA2/SCK0/DREQ0/IRQ0	PA2	PA2/SCK0/IRQ0	NC
51	133	48	PA3	PA3/RXD1	PA3	PA3/RXD1	PA3	PA3/RXD1	PA3	PA3/RXD1	NC
50	134	47	PA4	PA4/TXD1	PA4	PA4/TXD1	PA4	PA4/TXD1	PA4	PA4/TXD1	NC
49	136	46	PA5	PA5/SCK1/DREQ1/IRQ1	PA5	PA5/SCK1/DREQ1/IRQ1	PA5	PA5/SCK1/DREQ1/IRQ1	PA5	PA5/SCK1/IRQ1	NC
48	54	45	PA6	PA6/TCLKA/CS2	PA6	PA6/TCLKA/CS2	PA6	PA6/TCLKA/CS2	PA6	PA6/TCLKA	NC
47	53	44	PA7	PA7/TCLKB/CS3	PA7	PA7/TCLKB/CS3	PA7	PA7/TCLKB/CS3	PA7	PA7/TCLKB	NC
46	52	43	PA8	PA8/TCLKC/IRQ2	PA8	PA8/TCLKC/IRQ2	PA8	PA8/TCLKC/IRQ2	PA8	PA8/TCLKC/IRQ2	NC
45	51	42	PA9	PA9/TCLKD/IRQ3	PA9	PA9/TCLKD/IRQ3	PA9	PA9/TCLKD/IRQ3	PA9	PA9/TCLKD/IRQ3	NC
44	50	41	CS0	CS0	CS0	CS0	PA10	PA10/CS0	PA10	PA10	NC
43	49	40	CS1	CS1	CS1	CS1	PA11	PA11/CS1	PA11	PA11	NC
41	48	38	WRL	WRL	WRL	WRL	PA12	PA12/WRL	PA12	PA12	NC
39	47	36	WRH	WRH	WRH	WRH	PA13	PA13/WRH	PA13	PA13	NC
37	43	34	RD	RD	RD	RD	PA14	PA14/RD	PA14	PA14	NC
88	107	83	CK	PA15/CK	CK	PA15/CK	CK	PA15/CK	PA15	PA15/CK	NC
—	100	—	PA16	PA16/AH	PA16	PA16/AH	PA16	PA16/AH	PA16	PA16	NC
—	101	—	PA17	PA17/WAIT	PA17	PA17/WAIT	PA17	PA17/WAIT	PA17	PA17	NC
—	33	—	PA18	PA18/BREQ/DRAK0	PA18	PA18/BREQ/DRAK0	PA18	PA18/BREQ/DRAK0	PA18	PA18	NC
—	30	—	PA19	PA19/BACK/DRAK1	PA19	PA19/BACK/DRAK1	PA19	PA19/BACK/DRAK1	PA19	PA19	NC
—	29	—	PA20	PA20/CASHL	PA20	PA20/CASHL	PA20	PA20/CASHL	PA20	PA20	NC
—	4	—	PA21	PA21/CASHH	PA21	PA21/CASHH	PA21	PA21/CASHH	PA21	PA21	NC
—	3	—	WRHL	WRHL	WRHL	WRHL	PA22	PA22/WRHL	PA22	PA22	NC
—	1	—	WRHH	WRHH	WRHH	WRHH	PA23	PA23/WRHH	PA23	PA23	NC
85	104	80	PLLVCC	PLLVCC	PLLVCC	PLLVCC	PLLVCC	PLLVCC	PLLVCC	PLLVCC	VCC
87	106	82	PLLVSS	PLLVSS	PLLVSS	PLLVSS	PLLVSS	PLLVSS	PLLVSS	PLLVSS	VSS
79	96	74	EXTAL	EXTAL	EXTAL	EXTAL	EXTAL	EXTAL	EXTAL	EXTAL	NC
77	94	72	XTAL	XTAL	XTAL	XTAL	XTAL	XTAL	XTAL	XTAL	NC
86	105	81	PLLCAP	PLLCAP	PLLCAP	PLLCAP	PLLCAP	PLLCAP	PLLCAP	PLLCAP	NC
81	98	76	NMI	NMI	NMI	NMI	NMI	NMI	NMI	NMI	A9
89	108	84	RES	RES	RES	RES	RES	RES	RES	RES	VPP
88	44	35	WDTOVF	WDTOVF	WDTOVF	WDTOVF	WDTOVF	WDTOVF	WDTOVF	WDTOVF	NC
84	103	79	MD0	MD0	MD0	MD0	MD0	MD0	MD0	MD0	VCC
83	102	78	MD1	MD1	MD1	MD1	MD1	MD1	MD1	MD1	VCC
80	97	75	MD2	MD2	MD2	MD2	MD2	MD2	MD2	MD2	VCC
78	95	73	MD3	MD3	MD3	MD3	MD3	MD3	MD3	MD3	VCC
82	99	77	VCC	VCC	VCC	VCC	VCC	VCC	VCC	VCC	VCC
107	128	100	AVCC	AVCC	AVCC	AVCC	AVCC	AVCC	AVCC	AVCC	VCC
104	124	97	AVSS	AVSS	AVSS	AVSS	AVSS	AVSS	AVSS	AVSS	VSS

Table 18.2 Pin Arrangement by Mode (cont)

Pin NO.			Pin Name								
			On-Chip ROM Disabled				On-Chip ROM Enabled				
			MPU Mode0		MPU Mode 1		MPU Mode2		Single Chip Mode		
TFP120	FP144	FP112	Initial Function	PFC Selected Function Possiblilities	Initial Function	PFC Selected Function Possiblilities	Initial Function	PFC Selected Function Possiblilities	Initial Function	PFC Selected Function Possiblilities	PROM Mode
—	127	—	AVREF	AVREF	AVREF	AVREF	AVREF	AVREF	AVREF	AVREF	
98	118	91	PF0/AN0	PF0/AN0	PF0/AN0	PF0/AN0	PF0/AN0	PF0/AN0	PF0/AN0	PF0/AN0	VSS
99	119	92	PF1/AN1	PF1/AN1	PF1/AN1	PF1/AN1	PF1/AN1	PF1/AN1	PF1/AN1	PF1/AN1	VSS
100	120	93	PF2/AN2	PF2/AN2	PF2/AN2	PF2/AN2	PF2/AN2	PF2/AN2	PF2/AN2	PF2/AN2	VSS
101	121	94	PF3/AN3	PF3/AN3	PF3/AN3	PF3/AN3	PF3/AN3	PF3/AN3	PF3/AN3	PF3/AN3	VSS
102	122	95	PF4/AN4	PF4/AN4	PF4/AN4	PF4/AN4	PF4/AN4	PF4/AN4	PF4/AN4	PF4/AN4	VSS
103	123	96	PF5/AN5	PF5/AN5	PF5/AN5	PF5/AN5	PF5/AN5	PF5/AN5	PF5/AN5	PF5/AN5	VSS
105	125	98	PF6/AN6	PF6/AN6	PF6/AN6	PF6/AN6	PF6/AN6	PF6/AN6	PF6/AN6	PF6/AN6	VSS
106	126	99	PF7/AN7	PF7/AN7	PF7/AN7	PF7/AN7	PF7/AN7	PF7/AN7	PF7/AN7	PF7/AN7	VSS
92	109	85	PE0	PE0/TIOC0A/$\overline{\text{DREQ0}}$	PE0	PE0/TIOC0A/$\overline{\text{DREQ0}}$	PE0	PE0/TIOC0A/$\overline{\text{DREQ0}}$	PE0	PE0/TIOC0A	NC
93	110	86	PE1	PE1/TIOC0B/DRAK0	PE1	PE1/TIOC0B/DRAK0	PE1	PE1/TIOC0B/DRAK0	PE1	PE1/TIOC0B	NC
94	111	87	PE2	PE2/TIOC0C/$\overline{\text{DREQ1}}$	PE2	PE2/TIOC0C/$\overline{\text{DREQ1}}$	PE2	PE2/TIOC0C/$\overline{\text{DREQ1}}$	PE2	PE2/TIOC0C	NC
95	113	88	PE3	PE3/TIOC0D/DRAK1	PE3	PE3/TIOC0D/DRAK1	PE3	PE3/TIOC0D/DRAK1	PE3	PE3/TIOC0D	NC
96	114	89	PE4	PE4/TIOC1A	PE4	PE4/TIOC1A	PE4	PE4/TIOC1A	PE4	PE4/TIOC1A	NC
109	115	102	PE5	PE5/TIOC1B	PE5	PE5/TIOC1B	PE5	PE5/TIOC1B	PE5	PE5/TIOC1B	NC
112	116	104	PE6	PE6/TIOC2A	PE6	PE6/TIOC2A	PE6	PE6/TIOC2A	PE6	PE6/TIOC2A	NC
113	137	105	PE7	PE7/TIOC2B	PE7	PE7/TIOC2B	PE7	PE7/TIOC2B	PE7	PE7/TIOC2B	NC
114	138	106	PE8	PE8/TIOC3A	PE8	PE8/TIOC3A	PE8	PE8/TIOC3A	PE8	PE8/TIOC3A	NC
115	139	107	PE9	PE9/TIOC3B	PE9	PE9/TIOC3B	PE9	PE9/TIOC3B	PE9	PE9/TIOC3B	NC
116	140	108	PE10	PE10/TIOC3C	PE10	PE10/TIOC3C	PE10	PE10/TIOC3C	PE10	PE10/TIOC3C	NC
118	142	110	PE11	PE11/TIOC3D	PE11	PE11/TIOC3D	PE11	PE11/TIOC3D	PE11	PE11/TIOC3D	NC
119	143	111	PE12	PE12/TIOC4A	PE12	PE12/TIOC4A	PE12	PE12/TIOC4A	PE12	PE12/TIOC4A	NC
120	144	112	PE13	PE13/TIOC4B/$\overline{\text{MRES}}$	PE13	PE13/TIOC4B/$\overline{\text{MRES}}$	PE13	PE13/TIOC4B/$\overline{\text{MRES}}$	PE13	PE13/TIOC4B/$\overline{\text{MRES}}$	NC
2	2	1	PE14	PE14/TIOC4C/DACK0/$\overline{\text{AH}}$	PE14	PE14/TIOC4C/DACK0/$\overline{\text{AH}}$	PE14	PE14/TIOC4C/DACK0/$\overline{\text{AH}}$	PE14	PE14/TIOC4C	NC
3	5	2	PE15	PE15/TIOC4D/DACK1/$\overline{\text{IRQOUT}}$	PE15	PE15/TIOC4D/DACK1/$\overline{\text{IRQOUT}}$	PE15	PE15/TIOC4D/DACK1/$\overline{\text{IRQOUT}}$	PE15	PE15/TIOC4D	$\overline{\text{CE}}$

18.2 Register Configuration

Table 18.3 summarizes the registers of the pin function controller.

Table 18.3 Pin Function Controller Registers

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port A I/O register H	PAIORH	R/W	H'0000	H'FFFF8384 H'FFFF8385	8, 16, 32
Port A I/O register L	PAIORL	R/W	H'0000	H'FFFF8386 H'FFFF8387	8, 16, 32
Port A control register H	PACRH	R/W	H'0000	H'FFFF8388 H'FFFF8389	8, 16, 32
Port A control register L1	PACRL1	R/W	H'0000* H'4000	H'FFFF838C H'FFFF838D	8, 16, 32
Port A control register L2	PACRL2	R/W	H'0000	H'FFFF838E H'FFFF838F	8, 16, 32
Port B I/O register	PBIOR	R/W	H'0000	H'FFFF8394 H'FFFF8395	8, 16, 32
Port B control register 1	PBCR1	R/W	H'0000	H'FFFF8398 H'FFFF8399	8, 16, 32
Port B control register 2	PBCR2	R/W	H'0000	H'FFFF839A H'FFFF839B	8, 16, 32
Port C I/O register	PCIOR	R/W	H'0000	H'FFFF8396 H'FFFF8397	8, 16, 32
Port C control register	PCCR	R/W	H'0000	H'FFFF839C H'FFFF839D	8, 16, 32
Port D I/O register H	PDIORH	R/W	H'0000	H'FFFF83A4 H'FFFF83A5	8, 16, 32
Port D I/O register L	PDIORL	R/W	H'0000	H'FFFF83A6 H'FFFF83A7	8, 16, 32
Port D control register H1	PDCRH1	R/W	H'0000	H'FFFF83A8 H'FFFF83A9	8, 16, 32
Port D control register H2	PDCRH2	R/W	H'0000	H'FFFF83AA H'FFFF83AB	8, 16, 32
Port D control register L	PDCRL	R/W	H'0000	H'FFFF83AC H'FFFF83AD	8, 16, 32
Port E I/O register	PEIOR	R/W	H'0000	H'FFFF83B4 H'FFFF83B5	8, 16, 32
Port E control register 1	PECR1	R/W	H'0000	H'FFFF83B8 H'FFFF83B9	8, 16, 32
Port E control register 2	PECR2	R/W	H'0000	H'FFFF83BA H'FFFF83BB	8, 16, 32
IRQOUT function control register	IFCR	R/W	H'0000	H'FFFF83C8 H'FFFF83C9	8, 16, 32

Note: * The port A control register L1 initial value varies depending on the operating mode.

18.3 Register Descriptions

18.3.1 Port A I/O Register H (PAIORH)

The port A I/O register H (PAIORH) is a 16-bit read/write register that selects input or output for the most significant 8 pins of port A. Bits PA23IOR–PA16IOR correspond to pins PA23/$\overline{\text{WRHH}}$–PA16/$\overline{\text{AH}}$. PAIORH is enabled when the port A pins function as general input/outputs (PA23–PA16). For other functions, it is disabled.

For port A pin functions PA23–PA16, a given pin in port A is an output pin if its corresponding PAIORH bit is set to 1, and an input pin if the bit is cleared to 0.

PAIORH is initialized to H'0000 by external power-on reset; however, it is not initialized for manual resets, reset by WDT, standby mode, or sleep mode, so the previous data is maintained.

The settings for this register are effective only for the 144-pin version. There are no corresponding pins for this register in the 112-pin and 120-pin versions. However, read/writes are possible.

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	PA23 IOR	PA22 IOR	PA21 IOR	PA20 IOR	PA19 IOR	PA18 IOR	PA17 IOR	PA16 IOR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

18.3.2 Port A I/O Register L (PAIORL)

The port A I/O register L (PAIORL) is a 16-bit read/write register that selects input or output for the least significant 16 pins of port A. Bits PA15IOR–PA0IOR correspond to pins PA15/CK–PA0/RXD0. PAIORL is enabled when the port A pins function as general input/outputs (PA15–PA0), or with the serial clock (SCK1, SCK0). For other functions, it is disabled.

When the port A pin functions PA15–PA0 are SCK1, SCK0, a given pin in port A is an output pin if its corresponding PAIORL bit is set to 1, and an input pin if the bit is cleared to 0.

PAIORL is initialized to H'0000 by external power-on reset; however, it is not initialized for manual resets, reset by WDT, standby mode, or sleep mode, so the previous data is maintained.

Bit:	15	14	13	12	11	10	9	8
	PA15 IOR	PA14 IOR	PA13 IOR	PA12 IOR	PA11 IOR	PA10 IOR	PA9 IOR	PA8 IOR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	PA7 IOR	PA6 IOR	PA5 IOR	PA4 IOR	PA3 IOR	PA2 IOR	PA1 IOR	PA0 IOR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

18.3.3 Port A Control Register H (PACRH)

PACRH is a 16-bit read/write register that selects the multiplex pin function for the eight most significant pins of port A. PACRH selects the PA23/$\overline{\text{WRHH}}$–PA16/$\overline{\text{AH}}$ pin functions.

The eight most significant pins of port A have bus control signals ($\overline{\text{WRHH}}$, $\overline{\text{WRHL}}$, $\overline{\text{CASHH}}$, $\overline{\text{CASHL}}$, $\overline{\text{BACK}}$, $\overline{\text{BREQ}}$, $\overline{\text{WAIT}}$, $\overline{\text{AH}}$) and DMAC control signals (DRAK1, DRAK0), but there are instances when the register settings that select these pin functions will be ignored. Refer to table 18.2, Pin Arrangement by Mode.

PACRH is initialized to H'0000 by external power-on reset but is not initialized for manual resets, reset by WDT, standby mode, or sleep mode, so the previous data is maintained.

The settings for this register are effective only for the 144-pin version. There are no corresponding pins for this register in the 112-pin and 120-pin versions. However, read/writes are possible.

Bit:	15	14	13	12	11	10	9	8
	—	PA23 MD	—	PA22 MD	—	PA21 MD	—	PA20 MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R/W	R	R/W	R	R/W

Bit:	7	6	5	4	3	2	1	0
	PA19 MD1	PA19 MD0	PA18 MD1	PA18 MD0	—	PA17 MD	—	PA16 MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R/W	R	R/W

- Bit 15—Reserved: This bit always reads as 0. The write value should always be 0.
- Bit 14—PA23 Mode (PA23MD): Selects the function of the PA23/$\overline{\text{WRHH}}$ pin.

Bit 14: PA23MD	Description
0	General input/output (PA23) (initial value) ($\overline{\text{WRHH}}$ in on-chip ROM invalid mode)
1	Most significant byte write output ($\overline{\text{WRHH}}$) (PA23 in single chip mode)

- Bit 13—Reserved: This bit always reads as 0. The write value should always be 0.
- Bit 12—PA22 Mode (PA22MD): Selects the function of the PA22/$\overline{\text{WRHL}}$ pin.

Bit 12: PA22MD	Description
0	General input/output (PA22) (initial value) ($\overline{\text{WRHL}}$ in on-chip ROM invalid mode)
1	Write output ($\overline{\text{WRHL}}$) (PA22 in single chip mode)

- Bit 11—Reserved: This bit always reads as 0. The write value should always be 0.
- Bit 10—PA21 Mode (PA21MD): Selects the function of the PA21/$\overline{\text{CASHH}}$ pin.

Bit 10: PA21MD	Description
0	General input/output (PA21) (initial value)
1	Column address output ($\overline{\text{CASHH}}$) (PA21 in single chip mode)

- Bit 9—Reserved: Always reads as 0. The write values should always be 0.

- Bit 8—PA20 Mode (PA20MD): Selects the function of the PA20/$\overline{\text{CASHL}}$ pin.

Bit 8: PA20MD	Description
0	General input/output (PA20) (initial value)
1	Column address output ($\overline{\text{CASHL}}$) (PA20 in single chip mode)

- Bits 7 and 6—PA19 Mode 1, 0 (PA19MD1 and PA19MD0): These bits select the function of the PA19/$\overline{\text{BACK}}$/DRAK1 pin.

Bit 7: PA19MD1	Bit 6: PA19MD0	Description
0	0	General input/output (PA19) (initial value)
	1	Bus right request acknowledge ($\overline{\text{BACK}}$) (PA19 in single chip mode)
1	0	DREQ1 request received output (DRAK1) (PA19 in single chip mode)
	1	Reserved

- Bits 5 and 4—PA18 Mode 1, 0 (PA18MD1 and PA18MD0): These bits select the function of the PA18/$\overline{\text{BREQ}}$/DRAK0 pin.

Bit 5: PA18MD1	Bit 4: PA18MD0	Description
0	0	General input/output (PA18) (initial value)
	1	Bus right request input ($\overline{\text{BREQ}}$) (PA18 in single chip mode)
1	0	DREQ0 request received output (DRAK0) (PA18 in single chip mode)
	1	Reserved

- Bit 3—Reserved: This bit always reads as 0. The write value should always be 0.
- Bit 2—PA17 Mode (PA17MD): Selects the function of the PA17/$\overline{\text{WAIT}}$ pin.

Bit 2: PA17MD	Description
0	General input/output (PA17) (initial value)
1	Wait state request input ($\overline{\text{WAIT}}$) (PA17 in single chip mode)

- Bit 1—Reserved: This bit always reads as 0. The write value should always be 0.

- Bit 0—PA16 Mode (PA16MD): Selects the function of the PA16/$\overline{AH}$ pin.

Bit 0: PA16MD	Description
0	General input/output (PA16) (initial value)
1	Address hold output ($\overline{AH}$) (PA16 in single chip mode)

18.3.4 Port A Control Registers L1, L2 (PACRL1 and PACRL2)

PACRL1 and PACRL2 are 16-bit read/write registers that select the functions of the least significant sixteen multiplexed pins of port A. PACRL1 selects the function of the PA15/CK–PA8/TCLKC/$\overline{IRQ2}$ pins of port A; PACRL2 selects the function of the PA7/TCLKB/$\overline{CS3}$–PA0/RXD0 pins of port A.

Port A has bus control signals ($\overline{RD}$, $\overline{WRH}$, $\overline{WRL}$, $\overline{CS0}$–$\overline{CS3}$, $\overline{AH}$) and DMAC control signals (DREQ0–DREQ1), but there are instances when the register settings that select these pin functions will be ignored, depending on the operation mode. Refer to table 18.2, Pin Arrangement by Mode, for details.

PACRL1 is initialized by external power-on reset to H'4000 in extended mode, and to H'0000 in single chip mode. PACRL2 is initialized by external power-on reset to H'0000. Neither register is initialized by manual resets, reset by WDT, standby mode, or sleep mode, so the previous data is maintained.

Port A Control Register L1 (PACRL1):

Bit:	15	14	13	12	11	10	9	8
	—	PA15MD	—	PA14MD	—	PA13MD	—	PA12MD
Initial value:	0	0(1)*	0	0	0	0	0	0
R/W:	R	R/W	R	R/W	R	R/W	R	R/W

Bit:	7	6	5	4	3	2	1	0
	—	PA11MD	—	PA10MD	PA9MD1	PA9MD0	PA8MD1	PA8MD0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R/W	R/W	R/W	R/W	R/W

Note: * Bit 14 is initialized to 1 in extended mode.

- Bit 15—Reserved: This bit always reads as 0. The write value should always be 0.

- Bit 14—PA15 Mode (PA15MD): Selects the function of the PA15/CK pin.

Bit 14: PA15MD	Description
0	General input/output (PA15) (single chip mode initial value)
1	Clock output (CK) (extended mode initial value)

- Bit 13—Reserved: This bit always reads as 0. The write value should always be 0.
- Bit 12—PA14 Mode (PA14MD): Selects the function of the PA14/$\overline{RD}$ pin.

Bit 12: PA14MD	Description
0	General input/output (PA14) (initial value) ($\overline{RD}$ in on-chip ROM invalid mode)
1	Read output ($\overline{RD}$) (PA14 in single chip mode)

- Bit 11—Reserved: This bit always reads as 0. The write value should always be 0.
- Bit 10—PA13 Mode (PA13MD): Selects the function of the PA13/$\overline{WRH}$ pin.

Bit 10: PA13MD	Description
0	General input/output (PA13) (initial value) ($\overline{WRH}$ in on-chip ROM invalid mode)
1	Most significant side write output ($\overline{WRH}$) (PA13 in single chip mode)

- Bit 9—Reserved: This bit always reads as 0. The write value should always be 0.
- Bit 8—PA12 Mode (PA12MD): Selects the function of the PA12/$\overline{WRL}$ pin.

Bit 8: PA12MD	Description
0	General input/output (PA12) (initial value) ($\overline{WRL}$ in on-chip ROM invalid mode)
1	Least significant side write output ($\overline{WRL}$) (PA12 in single chip mode)

- Bit 7—Reserved: This bit always reads as 0. The write value should always be 0.
- Bit 6—PA11 Mode (PA11MD): Selects the function of the PA11/$\overline{CS1}$ pin.

Bit 6: PA11MD	Description
0	General input/output (PA11) (initial value) ($\overline{CS1}$ in on-chip ROM invalid mode)
1	Chip select output ($\overline{CS1}$) (PA11 in single chip mode)

- Bit 5—Reserved: This bit always reads as 0. The write value should always be 0.

- Bit 4—PA10 Mode (PA10MD): Selects the function of the PA10/$\overline{CS0}$ pin.

Bit 4: PA10MD	Description
0	General input/output (PA10) (initial value) ($\overline{CS0}$ in on-chip ROM invalid mode)
1	Chip select output ($\overline{CS0}$) (PA10 in single chip mode)

- Bits 3 and 2—PA9 Mode 1, 0 (PA9MD1 and PA9MD0): These bits select the function of the PA9/TCLKD/$\overline{IRQ3}$ pin.

Bit 3: PA9MD1	Bit 2: PA9MD0	Description
0	0	General input/output (PA9) (initial value)
	1	MTU timer clock input (TCLKD)
1	0	Interrupt request input ($\overline{IRQ3}$)
	1	Reserved

- Bits 1 and 0—PA8 Mode 1, 0 (PA8MD1 and PA8MD0): These bits select the function of the PA8/TCLKC/$\overline{IRQ2}$ pin.

Bit 1: PA8MD1	Bit 0: PA8MD0	Description
0	0	General input/output (PA8) (initial value)
	1	MTU timer clock input (TCLKC)
1	0	Interrupt request input ($\overline{IRQ2}$)
	1	Reserved

Port A Control Register L2 (PACRL2):

Bit:	15	14	13	12	11	10	9	8
	PA7 MD1	PA7 MD0	PA6 MD1	PA6 MD0	PA5 MD1	PA5 MD0	—	PA4MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W

Bit:	7	6	5	4	3	2	1	0
	—	PA3MD	PA2 MD1	PA2 MD0	—	PA1MD	—	PA0MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R	R/W

- Bits 15 and 14—PA7 Mode 1, 0 (PA7MD1 and PA7MD0): These bits select the function of the PA7/TCLKB/$\overline{CS3}$ pin.

Bit 15: PA7MD1	Bit 14: PA7MD0	Description
0	0	General input/output (PA7) (initial value)
	1	MTU timer clock input (TCLKB)
1	0	Chip select output ($\overline{CS3}$) (PA7 in single chip mode)
	1	Reserved

- Bits 13 and 12—PA6 Mode 1, 0 (PA6MD1 and PA6MD0): These bits select the function of the PA6/TCLKA/$\overline{CS2}$ pin.

Bit 13: PA6MD1	Bit 12: PA6MD0	Description
0	0	General input/output (PA6) (initial value)
	1	MTU timer clock input (TCLKA)
1	0	Chip select output ($\overline{CS2}$) (PA6 in single chip mode)
	1	Reserved

- Bits 11 and 10—PA5 Mode 1, 0 (PA5MD1 and PA5MD0): These bits select the function of the PA5/SCK1/$\overline{\text{DREQ1}}$/$\overline{\text{IRQ1}}$ pin.

Bit 11: PA5MD1	Bit 10: PA5MD0	Description
0	0	General input/output (PA5) (initial value)
	1	Serial clock input/output (SCK1)
1	0	DMA transfer request received input ($\overline{\text{DREQ1}}$) (PA5 in single chip mode)
	1	Interrupt request input ($\overline{\text{IRQ1}}$)

- Bit 9—Reserved: This bit always reads as 0. The write value should always be 0.
- Bit 8—PA4 Mode (PA4MD): Selects the function of the PA4/TxD1 pin.

Bit 8: PA4MD	Description
0	General input/output (PA4) (initial value)
1	Transmit data output (TxD1)

- Bit 7—Reserved: This bit always reads as 0. The write value should always be 0.
- Bit 6—PA3 Mode (PA3MD): Selects the function of the PA3/RxD1 pin.

Bit 6: PA3MD	Description
0	General input/output (PA3) (initial value)
1	Receive data input (RxD1)

- Bits 5 and 4—PA2 Mode 1, 0 (PA2MD1 and PA2MD0): These bits select the function of the PA2/SCK0/$\overline{\text{DREQ0}}$/$\overline{\text{IRQ0}}$ pin.

Bit 5: PA2MD1	Bit 4: PA2MD0	Description
0	0	General input/output (PA2) (initial value)
	1	Serial clock input/output (SCK0)
1	0	DMA transfer request received input ($\overline{\text{DREQ0}}$) (PA2 in single chip mode)
	1	Interrupt request input ($\overline{\text{IRQ0}}$)

- Bit 3—Reserved: This bit always reads as 0. The write value should always be 0.

- Bit 2—PA1 Mode (PA1MD): Selects the function of the PA1/TxD0 pin.

Bit 2: PA1MD	Description
0	General input/output (PA1) (initial value)
1	Transmit data output (TxD0)

- Bit 1—Reserved: This bit always reads as 0. The write value should always be 0.

- Bit 0—PA0 Mode (PA0MD): Selects the function of the PA0/RxD0 pin.

Bit 0: PA0MD	Description
0	General input/output (PA0) (initial value)
1	Receive data input (RxD0)

18.3.5 Port B I/O Register (PBIOR)

The port B I/O register (PBIOR) is a 16-bit read/write register that selects input or output for the ten port B pins. Bits PB9IOR–PB0IOR correspond to the PB9/$\overline{\text{IRQ7}}$/A21/$\overline{\text{ADTRG}}$ pin to PB0/A16 pin. PBIOR is enabled when the port B pins function as input/outputs (PB9–PB0). For other functions, it is disabled.

For port B pin functions PB9–PB0, a given pin in port B is an output pin if its corresponding PBIOR bit is set to 1, and an input pin if the bit is cleared to 0.

PBIOR is initialized to H'0000 by external power-on reset; however, it is not initialized for manual resets, reset by WDT, standby mode, or sleep mode, so the previous data is maintained.

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	PB9 IOR	PB8 IOR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	PB7 IOR	PB6 IOR	PB5 IOR	PB4 IOR	PB3 IOR	PB2 IOR	PB1 IOR	PB0 IOR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

18.3.6 Port B Control Registers (PBCR1 and PBCR2)

PBCR1 and PBCR2 are 16-bit read/write registers that select the functions of the ten multiplexed pins of port B. PBCR1 selects the functions of the top two bits of port B; PBCR2 selects the functions of the bottom eight bits of port B.

Port B has bus control signals (RDWR, $\overline{\text{RAS}}$, $\overline{\text{CASH}}$, $\overline{\text{CASL}}$, $\overline{\text{WAIT}}$, $\overline{\text{BREQ}}$, $\overline{\text{BACK}}$) and address outputs (A21, A20, A19, A18, A17, A16), but there are instances when the register settings that select these pin functions will be ignored, depending on the operation mode. Refer to table 18.2, Pin Arrangement by Mode, for details.

PBCR1 and PBCR2 are both initialized to H'0000 by external power-on reset but are not initialized for manual resets, reset by WDT, standby mode, or sleep mode, so the previous data is maintained.

Port B Control Register 1 (PBCR1):

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	PB9 MD1	PB9 MD0	PB8 MD1	PB8 MD0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W

- Bits 15-4—Reserved: These bits always read as 0. The write value should always be 0.
- Bits 3 and 2—PB9 Mode (PB9MD1 and PB9MD0): PB9MD1 and PB9MD0 select the function of the PB9/$\overline{\text{IRQ7}}$/A21/$\overline{\text{ADTRG}}$ pin.

Bit 3: PB9MD1	Bit 2: PB9MD0	Description
0	0	General input/output (PB9) (initial value)
	1	Interrupt request input ($\overline{\text{IRQ7}}$)
1	0	Address output (A21) (PB9 in single chip mode)
	1	A/D conversion trigger input ($\overline{\text{ADTRG}}$)

- Bits 1 and 0—PB8 Mode (PB8MD1 and PB8MD0): PB8MD1 and PB8MD0 select the function of the PB8/$\overline{\text{IRQ6}}$/A20/$\overline{\text{WAIT}}$ pin.

Bit 1: PB8MD1	Bit 0: PB8MD0	Description
0	0	General input/output (PB8) (initial value)
	1	Interrupt request input ($\overline{\text{IRQ6}}$)
1	0	Address output (A20) (PB8 in single chip mode)
	1	Wait state request input ($\overline{\text{WAIT}}$) (PB8 in single chip mode)

Port B Control Register 2 (PBCR2):

Bit:	15	14	13	12	11	10	9	8
	PB7MD1	PB7MD0	PB6MD1	PB6MD0	PB5MD1	PB5MD0	PB4MD1	PB4MD0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	PB3MD1	PB3MD0	PB2MD1	PB2MD0	—	PB1MD	—	PB0MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R/W	R	R/W

- Bits 15 and 14—PB7 Mode (PB7MD1 and PB7MD0): PB7MD1 and PB7MD0 select the function of the PB7/$\overline{\text{IRQ5}}$/A19/$\overline{\text{BREQ}}$ pin.

Bit 15: PB7MD1	Bit 14: PB7MD0	Description
0	0	General input/output (PB7) (initial value)
	1	Interrupt request input ($\overline{\text{IRQ5}}$)
1	0	Address output (A19) (PB7 in single chip mode)
	1	Bus right request input ($\overline{\text{BREQ}}$) (PB7 in single chip mode)

- Bits 13 and 12—PB6 Mode (PB6MD1 and PB6MD0): PB6MD1 and PB6MD0 select the function of the PB6/$\overline{\text{IRQ4}}$/A18/$\overline{\text{BACK}}$ pin.

Bit 13: PB6MD1	Bit 12: PB6MD0	Description
0	0	General input/output (PB6) (initial value)
	1	Interrupt request input ($\overline{\text{IRQ4}}$)
1	0	Address output (A18) (PB6 in single chip mode)
	1	Bus right request output ($\overline{\text{BACK}}$) (PB6 in single chip mode)

- Bits 11 and 10—PB5 Mode (PB5MD1 and PB5MD0): PB5MD1 and PB5MD0 select the function of the PB5/$\overline{\text{IRQ3}}$/$\overline{\text{POE3}}$/RDWR pin.

Bit 11: PB5MD1	Bit 10: PB5MD0	Description
0	0	General input/output (PB5) (initial value)
	1	Interrupt request input ($\overline{\text{IRQ3}}$)
1	0	Port output enable ($\overline{\text{POE3}}$)
	1	Read/write output (RDWR)

- Bits 9 and 8—PB4 Mode (PB4MD1 and PB4MD0): PB4MD1 and PB4MD0 select the function of the PB4/$\overline{\text{IRQ2}}$/$\overline{\text{POE2}}$/$\overline{\text{CASH}}$ pin.

Bit 9: PB4MD1	Bit 8: PB4MD0	Description
0	0	General input/output (PB4) (initial value)
	1	Interrupt request input ($\overline{\text{IRQ2}}$)
1	0	Port output enable ($\overline{\text{POE2}}$)
	1	Column address strobe ($\overline{\text{CASH}}$) (PB4 in single chip mode)

- Bits 7 and 6—PB3 Mode (PB3MD1 and PB3MD0): PB3MD1 and PB3MD0 select the function of the PB3/$\overline{\text{IRQ1}}$/$\overline{\text{POE1}}$/$\overline{\text{CASL}}$ pin.

Bit 7: PB3MD1	Bit 6: PB3MD0	Description
0	0	General input/output (PB3) (initial value)
	1	Interrupt request input ($\overline{\text{IRQ1}}$)
1	0	Port output enable ($\overline{\text{POE1}}$)
	1	Column address strobe ($\overline{\text{CASL}}$) (PB3 in single chip mode)

- Bits 5 and 4—PB2 Mode (PB2MD1 and PB2MD0): PB2MD1 and PB2MD0 select the function of the PB2/$\overline{\text{IRQ0}}$/$\overline{\text{POE0}}$/$\overline{\text{RAS}}$ pin.

Bit 5: PB2MD1	Bit 4: PB2MD0	Description
0	0	General input/output (PB2) (initial value)
	1	Interrupt request input ($\overline{\text{IRQ0}}$)
1	0	Port output enable ($\overline{\text{POE0}}$)
	1	Row address strobe ($\overline{\text{RAS}}$) (PB2 in single chip mode)

- Bit 3—Reserved: This bit always reads as 0. The write value should always be 0.
- Bit 2—PB1 Mode (PB1MD): Selects the function of the PB1/A17 pin.

Bit 2: PB1MD	Description
0	General input/output (PB1) (initial value) (A17 in on-chip ROM invalid mode)
1	Address output (A17) (PB1 in single chip mode)

- Bit 1—Reserved: This bit always reads as 0. The write value should always be 0.
- Bit 0—PB0 Mode (PB0MD): Selects the function of the PB0/A16 pin.

Bit 0: PA0MD	Description
0	General input/output (PB0) (initial value) (A16 in on-chip ROM invalid mode)
1	Address output (A16) (PB0 in single chip mode)

18.3.7 Port C I/O Register (PCIOR)

The port C I/O register (PCIOR) is a 16-bit read/write register that selects input or output for the 16 port C pins. Bits PC15IOR–PC0IOR correspond to pins PC15/A15 to PC0/A0. PCIOR is enabled when the port C pins function as general input/outputs (PC15–PC0). For other functions, it is disabled.

When the port C pin functions are as PC15–PC0, a given pin in port C is an output pin if its corresponding PCIOR bit is set to 1, and an input pin if the bit is cleared to 0.

PCIOR is initialized to H'0000 by external power-on reset; however, it is not initialized for manual resets, reset by WDT, standby mode, or sleep mode, so the previous values are maintained.

Bit:	15	14	13	12	11	10	9	8
	PC15 IOR	PC14 IOR	PC13 IOR	PC12 IOR	PC11 IOR	PC10 IOR	PC9 IOR	PC8 IOR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	PC7 IOR	PC6 IOR	PC5 IOR	PC4 IOR	PC3 IOR	PC2 IOR	PC1 IOR	PC0 IOR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

18.3.8 Port C Control Register (PCCR)

PCCR is a 16-bit read/write register that selects the functions for the sixteen port C multiplexed pins. There are instances when these register settings will be ignored, depending on the operation mode. Refer to table 18.2, Pin Arrangement by Mode, for details.

PCCR is initialized to H'0000 by power-on resets but is not initialized for manual resets, reset by WDT, standby mode, or sleep mode, so the previous data is maintained.

Bit:	15	14	13	12	11	10	9	8
	PC15 MD	PC14 MD	PC13 MD	PC12 MD	PC11 MD	PC10 MD	PC9 MD	PC8 MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	PC7 MD	PC6 MD	PC5 MD	PC4 MD	PC3 MD	PC2 MD	PC1 MD	PC0 MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- Bit 15—PC15 Mode (PC15MD): Selects the function of the PC15/A15 pin.

Bit 15: PC15MD	Description
0	General input/output (PC15) (initial value) (A15 in on-chip ROM invalid mode)
1	Address output (A15) (PC15 in single chip mode)

- Bit 14—PC14 Mode (PC14MD): Selects the function of the PC14/A14 pin.

Bit 14: PC14MD	Description
0	General input/output (PC14) (initial value) (A14 in on-chip ROM invalid mode)
1	Address output (A14) (PC14 in single chip mode)

- Bit 13—PC13 Mode (PC13MD): Selects the function of the PC13/A13 pin.

Bit 13: PC13MD	Description
0	General input/output (PC13) (initial value) (A13 in on-chip ROM invalid mode)
1	Address output (A13) (PC13 in single chip mode)

- Bit 12—PC12 Mode (PC12MD): Selects the function of the PC12/A12 pin.

Bit 12: PC12MD	Description
0	General input/output (PC12) (initial value) (A12 in on-chip ROM invalid mode)
1	Address output (A12) (PC12 in single chip mode)

- Bit 11—PC11 Mode (PC11MD): Selects the function of the PC11/A11 pin.

Bit 11: PC11MD	Description
0	General input/output (PC11) (initial value) (A11 in on-chip ROM invalid mode)
1	Address output (A11) (PC11 in single chip mode)

- Bit 10—PC10 Mode (PC10MD): Selects the function of the PC10/A10 pin.

Bit 10: PC10MD	Description
0	General input/output (PC10) (initial value) (A10 in on-chip ROM invalid mode)
1	Address output (A10) (PC10 in single chip mode)

- Bit 9—PC9 Mode (PC9MD): Selects the function of the PC9/A9 pin.

Bit 9: PC9MD	Description
0	General input/output (PC9) (initial value) (A9 in on-chip ROM invalid mode)
1	Address output (A9) (PC9 in single chip mode)

- Bit 8—PC8 Mode (PC8MD): Selects the function of the PC8/A8 pin.

Bit 8: PC8MD	Description
0	General input/output (PC8) (initial value) (A8 in on-chip ROM invalid mode)
1	Address output (A8) (PC8 in single chip mode)

- Bit 7—PC7 Mode (PC7MD): Selects the function of the PC7/A7 pin.

Bit 7: PC7MD	Description
0	General input/output (PC7) (initial value) (A7 in on-chip ROM invalid mode)
1	Address output (A7) (PC7 in single chip mode)

- Bit 6—PC6 Mode (PC6MD): Selects the function of the PC6/A6 pin.

Bit 6: PC6MD	Description
0	General input/output (PC6) (initial value) (A6 in on-chip ROM invalid mode)
1	Address output (A6) (PC6 in single chip mode)

- Bit 5—PC5 Mode (PC5MD): Selects the function of the PC5/A5 pin.

Bit 5: PC5MD	Description
0	General input/output (PC5) (initial value) (A5 in on-chip ROM invalid mode)
1	Address output (A5) (PC5 in single chip mode)

- Bit 4—PC4 Mode (PC4MD): Selects the function of the PC4/A4 pin.

Bit 4: PC4MD	Description
0	General input/output (PC4) (initial value) (A4 in on-chip ROM invalid mode)
1	Address output (A4) (PC4 in single chip mode)

- Bit 3—PC3 Mode (PC3MD): Selects the function of the PC3/A3 pin.

Bit 3: PC3MD	Description
0	General input/output (PC3) (initial value) (A3 in on-chip ROM invalid mode)
1	Address output (A3) (PC3 in single chip mode)

- Bit 2—PC2 Mode (PC2MD): Selects the function of the PC2/A2 pin.

Bit 2: PC2MD	Description
0	General input/output (PC2) (initial value) (A2 in on-chip ROM invalid mode)
1	Address output (A2) (PC2 in single chip mode)

- Bit 1—PC1 Mode (PC1MD): Selects the function of the PC1/A1 pin.

Bit 1: PC1MD	Description
0	General input/output (PC1) (initial value) (A1 in on-chip ROM invalid mode)
1	Address output (A1) (PC1 in single chip mode)

- Bit 0—PC0 Mode (PC0MD): Selects the function of the PC0/A0 pin.

Bit 0: PC0MD	Description
0	General input/output (PC0) (initial value) (A0 in on-chip ROM invalid mode)
1	Address output (A0) (PC0 in single chip mode)

18.3.9 Port D I/O Register H (PDIORH)

The port D I/O register H (PDIORH) is a 16-bit read/write register that selects input or output for the most significant sixteen port D pins. Bits PD31IOR–PD16IOR correspond to the PD31/D31/$\overline{\text{ADTRG}}$ pin to PD16/D16/$\overline{\text{IRQ0}}$ pin. PDIORH is enabled when the port D pins function as general input/outputs (PD31–PD16). For other functions, it is disabled.

For port D pin functions PD31–PD16, a given pin in port D is an output pin if its corresponding PDIORH bit is set to 1, and an input pin if the bit is cleared to 0.

PDIORH is initialized to H'0000 by external power-on reset; however, it is not initialized for manual resets, reset by WDT, standby mode, or sleep mode, so the previous data is maintained.

The settings for this register are effective only for the 144-pin version. There are no corresponding pins for this register in the 112-pin and 120-pin versions. However, read/writes are possible.

Bit:	15	14	13	12	11	10	9	8
	PD31 IOR	PD30 IOR	PD29 IOR	PD28 IOR	PD27 IOR	PD26 IOR	PD25 IOR	PD24 IOR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	PD23 IOR	PD22 IOR	PD21 IOR	PD20 IOR	PD19 IOR	PD18 IOR	PD17 IOR	PD16 IOR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

18.3.10 Port D I/O Register L (PDIORL)

The port D I/O register L (PDIORL) is a 16-bit read/write register that selects input or output for the least significant sixteen port D pins. Bits PD15IOR–PD0IOR correspond to the PD15/D15 pin to PD0/D0 pin. PDIORL is enabled when the port D pins function as general input/outputs (PD15–PD0). For other functions, it is disabled.

For port D pin functions PD15–PD0, a given pin in port D is an output pin if its corresponding PDIORL bit is set to 1, and an input pin if the bit is cleared to 0.

PDIORL is initialized to H'0000 by external power-on reset; however, it is not initialized for manual resets, reset by WDT, standby mode, or sleep mode, so the previous data is maintained.

Bit:	15	14	13	12	11	10	9	8
	PD15 IOR	PD14 IOR	PD13 IOR	PD12 IOR	PD11 IOR	PD10 IOR	PD9 IOR	PD8 IOR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	PD7 IOR	PD6 IOR	PD5 IOR	PD4 IOR	PD3 IOR	PD2 IOR	PD1 IOR	PD0 IOR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

18.3.11 Port D Control Registers H1, H2 (PDCRH1 and PDCRH2)

PDCRH1 and PDCRH2 are 16-bit read/write registers that select the functions of the most significant sixteen multiplexed pins of port D. PDCRH1 selects the functions of the PD31/D31/$\overline{\text{ADTRG}}$–PD24/D24/$\overline{\text{DREQ0}}$ pins of port D; PDCRH2 selects the functions of the PD23/D23/$\overline{\text{IRQ7}}$–PD16/D16/$\overline{\text{IRQ0}}$ pins of port D. There are instances when these register settings will be ignored, depending on the operation mode. Refer to table 18.2, Pin Arrangement by Mode, for details.

The settings for this register are effective only for the 144-pin version. There are no corresponding pins for this register in the 112-pin and 120-pin versions. However, read/writes are possible.

PDCRH1 and PDCRH2 are both initialized to H'0000 by external power-on reset but are not initialized for manual resets, reset by WDT, standby mode, or sleep mode, so the previous data is maintained.

Port D Control Register H1 (PDCRH1):

Bit:	15	14	13	12	11	10	9	8
	PD31 MD1	PD31 MD0	PD30 MD1	PD30 MD0	PD29 MD1	PD29 MD0	PD28 MD1	PD28 MD0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	PD27 MD1	PD27 MD0	PD26 MD1	PD26 MD0	PD25 MD1	PD25 MD0	PD24 MD1	PD24 MD0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- Bits 15 and 14—PD31 Mode 1, 0 (PD31MD1 and PD31MD0): These bits select the function of the PD31/D31/$\overline{\text{ADTRG}}$ pin.

Bit 15: PD31MD1	Bit 14: PD31MD0	Description
0	0	General input/output (PD31) (initial value) (No ROM, D31 with CS0 = 32 bit width)
	1	Data input/output (D31) (PD31 in single chip mode)
1	0	A/D conversion trigger input ($\overline{\text{ADTRG}}$) (No ROM, D31 with CS0 = 32 bit width)
	1	Reserved

- Bits 13 and 12—PD30 Mode 1, 0 (PD30MD1 and PD30MD0): These bits select the function of the PD30/D30/$\overline{\text{IRQOUT}}$ pin.

Bit 13: PD30MD1	Bit 12: PD30MD0	Description
0	0	General input/output (PD30) (initial value) (No ROM, D30 with CS0 = 32 bit width)
	1	Data input/output (D30) (PD30 in single chip mode)
1	0	Interrupt request received output ($\overline{\text{IRQOUT}}$) (No ROM, D30 with CS0 = 32 bit width. Reserved in single chip mode)
	1	Reserved

- Bits 11 and 10—PD29 Mode 1, 0 (PD29MD1 and PD29MD0): These bits select the function of the PD29/D29/$\overline{CS3}$ pin.

Bit 11: PD29MD1	Bit 10: PD29MD0	Description
0	0	General input/output (PD29) (initial value) (D29 with no ROM and CS0 = 32 bit width)
	1	Data input/output (D29) (PD29 in single chip mode)
1	0	Chip select output ($\overline{CS3}$) (PD29 in single chip mode, D29 with no ROM and CS0 = 32 bit width)
	1	Reserved

- Bits 9 and 8—PD28 Mode 1, 0 (PD28MD1 and PD28MD0): These bits select the function of the PD28/D28/$\overline{CS2}$ pin.

Bit 9: PD28MD1	Bit 8: PD28MD0	Description
0	0	General input/output (PD28) (initial value) (D28 with no ROM and CS0 = 32 bit width)
	1	Data input/output (D28) (PD28 in single chip mode)
1	0	Chip select output ($\overline{CS2}$) (PD28 in single chip mode, and D28 with no ROM and CS0 = 32 bit width)
	1	Reserved

- Bits 7 and 6—PD27 Mode 1, 0 (PD27MD1 and PD27MD0): These bits select the function of the PD27/D27/DACK1 pin.

Bit 7: PD27MD1	Bit 6: PD27MD0	Description
0	0	General input/output (PD27) (initial value) (D27 with no ROM and CS0 = 32 bit width)
	1	Data input/output (D27) (PD27 in single chip mode)
1	0	DMA transfer request received output (DACK1) (PD27 in single chip mode, and D27 with no ROM and CS0 = 32 bit width)
	1	Reserved

- Bits 5 and 4—PD26 Mode 1, 0 (PD26MD1 and PD26MD0): These bits select the function of the PD26/D26/DACK0 pin.

Bit 5: PD26MD1	Bit 4: PD26MD0	Description
0	0	General input/output (PD26) (initial value) (D26 with no ROM and CS0 = 32 bit width)
	1	Data input/output (D26) (PD26 in single chip mode)
1	0	DMA transfer request received output (DACK0) (PD26 in single chip mode, and D26 with no ROM and CS0 = 32 bit width)
	1	Reserved

- Bits 3 and 2—PD25 Mode 1, 0 (PD25MD1 and PD25MD0): These bits select the function of the PD25/D25/$\overline{\text{DREQ1}}$ pin.

Bit 3: PD25MD1	Bit 2: PD25MD0	Description
0	0	General input/output (PD25) (initial value) (D25 with no ROM and CS0 = 32 bit width)
	1	Data input/output (D25) (PD25 in single chip mode)
1	0	DMA transfer request input ($\overline{\text{DREQ1}}$) (PD25 in single chip mode, and D25 with no ROM and CS0 = 32 bit width)
	1	Reserved

- Bits 1 and 0—PD24 Mode 1, 0 (PD24MD1 and PD24MD0): These bits select the function of the PD24/D24/$\overline{\text{DREQ0}}$ pin.

Bit 1: PD24MD1	Bit 0: PD24MD0	Description
0	0	General input/output (PD24) (initial value) (D24 with no ROM and CS0 = 32 bit width)
	1	Data input/output (D24) (PD24 in single chip mode)
1	0	DMA transfer request input ($\overline{\text{DREQ0}}$) (PD24 in single chip mode, and D24 with no ROM and CS0 = 32 bit width)
	1	Reserved

Port D Control Register H2 (PDCRH2):

Bit:	15	14	13	12	11	10	9	8
	PD23 MD1	PD23 MD0	PD22 MD1	PD22 MD0	PD21 MD1	PD21 MD0	PD20 MD1	PD20 MD0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	PD19 MD1	PD19 MD0	PD18 MD1	PD18 MD0	PD17 MD1	PD17 MD0	PD16 MD1	PD16 MD0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- Bits 15 and 14—PD23 Mode 1, 0 (PD23MD1 and PD23MD0): These bits select the function of the PD23/D23/$\overline{\text{IRQ7}}$ pin.

Bit 15: PD23MD1	Bit 14: PD23MD0	Description
0	0	General input/output (PD23) (initial value) (D23 with no ROM and CS0 = 32 bit width)
	1	Data input/output (D23) (PD23 in single chip mode)
1	0	Interrupt request input ($\overline{\text{IRQ7}}$)
	1	Reserved

- Bits 13 and 12—PD22 Mode 1, 0 (PD22MD1 and PD22MD0): These bits select the function of the PD22/D22/$\overline{\text{IRQ6}}$ pin.

Bit 13: PD22MD1	Bit 12: PD22MD0	Description
0	0	General input/output (PD22) (initial value) (D22 with no ROM and CS0 = 32 bit width)
	1	Data input/output (D22) (PD22 in single chip mode)
1	0	Interrupt request input ($\overline{\text{IRQ6}}$)
	1	Reserved

- Bits 11 and 10—PD21 Mode 1, 0 (PD21MD1 and PD21MD0): These bits select the function of the PD21/D21/$\overline{\text{IRQ5}}$ pin.

Bit 11: PD21MD1	Bit 10: PD21MD0	Description
0	0	General input/output (PD21) (initial value) (D21 with no ROM and CS0 = 32 bit width)
	1	Data input/output (D21) (PD21 in single chip mode)
1	0	Interrupt request input ($\overline{\text{IRQ5}}$)
	1	Reserved

- Bits 9 and 8—PD20 Mode 1, 0 (PD20MD1 and PD20MD0): These bits select the function of the PD20/D20/$\overline{\text{IRQ4}}$ pin.

Bit 9: PD20MD1	Bit 8: PD20MD0	Description
0	0	General input/output (PD20) (initial value) (D20 with no ROM and CS0 = 32 bit width)
	1	Data input/output (D20) (PD20 in single chip mode)
1	0	Interrupt request input ($\overline{\text{IRQ4}}$)
	1	Reserved

- Bits 7 and 6—PD19 Mode 1, 0 (PD19MD1 and PD19MD0): These bits select the function of the PD19/D19/$\overline{\text{IRQ3}}$ pin.

Bit 7: PD19MD1	Bit 6: PD19MD0	Description
0	0	General input/output (PD19) (initial value) (D19 with no ROM and CS0 = 32 bit width)
	1	Data input/output (D19) (PD19 in single chip mode)
1	0	Interrupt request input ($\overline{\text{IRQ3}}$)
	1	Reserved

- Bits 5 and 4—PD18 Mode 1, 0 (PD18MD1 and PD18MD0): These bits select the function of the PD18/D18/$\overline{\text{IRQ2}}$ pin.

Bit 5: PD18MD1	Bit 4: PD18MD0	Description
0	0	General input/output (PD18) (initial value) (D18 with no ROM and CS0 = 32 bit width
	1	Data input/output (D18) (PD18 in single chip mode)
1	0	Interrupt request input ($\overline{\text{IRQ2}}$)
	1	Reserved

- Bits 3 and 2—PD17 Mode 1, 0 (PD17MD1 and PD17MD0): These bits select the function of the PD17/D17/$\overline{\text{IRQ1}}$ pin.

Bit 3: PD17MD1	Bit 2: PD17MD0	Description
0	0	General input/output (PD17) (initial value) (D17 with no ROM and CS0 = 32 bit width)
	1	Data input/output (D17) (PD17 in single chip mode)
1	0	Interrupt request input ($\overline{\text{IRQ1}}$)
	1	Reserved

- Bits 1 and 0—PD16 Mode 1, 0 (PD16MD1 and PD16MD0): These bits select the function of the PD16/D16/$\overline{\text{IRQ0}}$ pin.

Bit 1: PD16MD1	Bit 0: PD16MD0	Description
0	0	General input/output (PD16) (initial value) (D16 with no ROM and CS0 = 32 bit width)
	1	Data input/output (D16) (PD16 in single chip mode)
1	0	Interrupt request input ($\overline{\text{IRQ0}}$)
	1	Reserved

18.3.12 Port D Control Register L (PDCRL)

PDCRL is a 16-bit read/write register that selects the multiplexed pin functions for the least significant sixteen port D pins. There are instances when these register settings will be ignored, depending on the operation mode.

On-Chip ROM-Disabled Extended Mode:

- 144-pin version:
  - Mode 0 (16-bit bus): Port D pins are data I/O pins; PDCRL settings are disabled.
  - Mode 1 (32-bit bus): Port D pins are data I/O pins; PDCRL settings are disabled.
- 112-pin and 120-pin versions:
  - Mode 0 (8-bit bus): Port D pins are data I/O pins; PDCRL settings are disabled.
  - Mode 1 (16-bit bus): Port D pins are data I/O pins; PDCRL settings are disabled.

On-Chip ROM-Enabled Extended Mode: The port D pins are shared as data I/O pins and general I/O pins; PDCRL settings are enabled.

Single Chip Mode: The port D pins are general I/O pins; PDCRL settings are disabled.

PDCRL is initialized to H'0000 by external power-on reset but is not initialized for manual resets, reset by WDT, standby mode, or sleep mode, so the previous data is maintained.

Port D Control Register L (PDCRL)

Bit:	15	14	13	12	11	10	9	8
	PD15 MD	PD14 MD	PD13 MD	PD12 MD	PD11 MD	PD10 MD	PD9 MD	PD8 MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	PD7 MD	PD6 MD	PD5 MD	PD4 MD	PD3 MD	PD2 MD	PD1 MD	PD0 MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- Bit 15—PD15 Mode (PD15MD): Selects the function of the PD15/D15 pin.

Bit 15: PD15MD	Description
0	General input/output (PD15) (initial value) (D15 in on-chip ROM invalid mode)
1	Data input/output (D15) (PD15 in single chip mode)

- Bit 14—PD14 Mode (PD14MD): Selects the function of the PD14/D14 pin.

Bit 14: PD14MD	Description
0	General input/output (PD14) (initial value) (D14 in on-chip ROM invalid mode)
1	Data input/output (D14) (PD14 in single chip mode)

- Bit 13—PD13 Mode (PD13MD): Selects the function of the PD13/D13 pin.

Bit 13: PD13MD	Description
0	General input/output (PD13) (initial value) (D13 in on-chip ROM invalid mode)
1	Data input/output (D13) (PD13 in single chip mode)

- Bit 12—PD12 Mode (PD12MD): Selects the function of the PD12/D12 pin.

Bit 12: PD12MD	Description
0	General input/output (PD12) (initial value) (D12 in on-chip ROM invalid mode)
1	Data input/output (D12) (PD12 in single chip mode)

- Bit 11—PD11 Mode (PD11MD): Selects the function of the PD11/D11 pin.

Bit 11: PD11MD	Description
0	General input/output (PD11) (initial value) (D11 in on-chip ROM invalid mode)
1	Data input/output (D11) (PD11 in single chip mode)

- Bit 10—PD10 Mode (PD10MD): Selects the function of the PD10/D10 pin.

Bit 10: PD10MD	Description
0	General input/output (PD10) (initial value) (D10 in on-chip ROM invalid mode)
1	Data input/output (D10) (PD10 in single chip mode)

- Bit 9—PD9 Mode (PD9MD): Selects the function of the PD9/D9 pin.

Bit 9: PD9MD	Description
0	General input/output (PD9) (initial value) (D9 in on-chip ROM invalid mode)
1	Data input/output (D9) (PD9 in single chip mode)

- Bit 8—PD8 Mode (PD8MD): Selects the function of the PD8/D8 pin.

Bit 8: PD8MD	Description
0	General input/output (PD8) (initial value) (D8 in on-chip ROM invalid mode)
1	Data input/output (D8) (PD8 in single chip mode)

- Bit 7—PD7 Mode (PD7MD): Selects the function of the PD7/D7 pin.

Bit 7: PD7MD	Description
0	General input/output (PD7) (initial value) (D7 in on-chip ROM invalid mode)
1	Data input/output (D7) (PD7 in single chip mode)

- Bit 6—PD6 Mode (PD6MD): Selects the function of the PD6/D6 pin.

Bit 6: PD6MD	Description
0	General input/output (PD6) (initial value) (D6 in on-chip ROM invalid mode)
1	Data input/output (D6) (PD6 in single chip mode)

- Bit 5—PD5 Mode (PD5MD): Selects the function of the PD5/D5 pin.

Bit 5: PD5MD	Description
0	General input/output (PD5) (initial value) (D5 in on-chip ROM invalid mode)
1	Data input/output (D5) (PD5 in single chip mode)

- Bit 4—PD4 Mode (PD4MD): Selects the function of the PD4/D4 pin.

Bit 4: PD4MD	Description
0	General input/output (PD4) (initial value) (D4 in on-chip ROM invalid mode)
1	Data input/output (D4) (PD4 in single chip mode)

- Bit 3—PD3 Mode (PD3MD): Selects the function of the PD3/D3 pin.

Bit 3: PD3MD	Description
0	General input/output (PD3) (initial value) (D3 in on-chip ROM invalid mode)
1	Data input/output (D3) (PD3 in single chip mode)

- Bit 2—PD2 Mode (PD2MD): Selects the function of the PD2/D2 pin.

Bit 2: PD2MD	Description
0	General input/output (PD2) (initial value) (D2 in on-chip ROM invalid mode)
1	Data input/output (D2) (PD2 in single chip mode)

- Bit 1—PD1 Mode (PD1MD): Selects the function of the PD1/D1 pin.

Bit 1: PD1MD	Description
0	General input/output (PD1) (initial value) (D1 in on-chip ROM invalid mode)
1	Data input/output (D1) (PD1 in single chip mode)

- Bit 0—PD0 Mode (PD0MD): Selects the function of the PD0/D0 pin.

Bit 0: PD0MD	Description
0	General input/output (PD0) (initial value) (D0 in on-chip ROM invalid mode)
1	Data input/output (D0) (PD0 in single chip mode)

18.3.13 Port E I/O Register (PEIOR)

The port E I/O register (PEIOR) is a 16-bit read/write register that selects input or output for the 16 port E pins. Bits PE15IOR–PE0IOR correspond to pins PE15/TIOC4D/DACK1/$\overline{\text{IRQOUT}}$–PE0/TIOC0A/$\overline{\text{DREQ0}}$. PEIOR is enabled when the port E pins function as general input/outputs (PE15–PE0) or TIOC pin of the MTU. For other functions, it is disabled.

When the port E pin functions are as PE15–PE0, or TIOC pin of the MTU, a given pin in port E is an output pin if its corresponding PEIOR bit is set to 1, and an input pin if the bit is cleared to 0.

PEIOR is initialized to H'0000 by external power-on reset; however, it is not initialized for manual resets, reset by WDT, standby mode, or sleep mode, so the previous data is maintained.

Bit:	15	14	13	12	11	10	9	8
	PE15 IOR	PE14 IOR	PE13 IOR	PE12 IOR	PE11 IOR	PE10 IOR	PE9 IOR	PE8 IOR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	PE7 IOR	PE6 IOR	PE5 IOR	PE4 IOR	PE3 IOR	PE2 IOR	PE1 IOR	PE0 IOR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

18.3.14 Port E Control Registers 1, 2 (PECR1 and PECR2)

PECR1 and PECR2 are 16-bit read/write registers that select the functions of the sixteen multiplexed pins of port E. PECR1 selects the functions of the upper eight bit pins of port E; PECR2 selects the function of the lower eight bit pins of port E.

Port E has a bus control signal ($\overline{\text{AH}}$) and DMAC control signals (DACK1, DACK0, DRAK1, DRAK0), but there are instances when the register settings that select these pin functions will be ignored, depending on the operation mode. Refer to table 18.2, Pin Arrangement by Mode, for details.

PECR1 and PECR2 are both initialized to H'0000 by external power-on reset but are not initialized for manual resets, reset by WDT, standby mode, or sleep mode, so the previous data is maintained.

Port E Control Register 1 (PECR1):

Bit:	15	14	13	12	11	10	9	8
	PE15 MD1	PE15 MD0	PE14 MD1	PE14 MD0	PE13 MD1	PE13 MD0	—	PE12MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W

Bit:	7	6	5	4	3	2	1	0
	—	PE11MD	—	PE10MD	—	PE9MD	—	PE8MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R/W	R	R/W	R	R/W

- Bits 15 and 14—PE15 Mode 1, 0 (PE15MD1 and PE15MD0): These bits select the function of the PE15/TIOC4D/DACK1/$\overline{\text{IRQOUT}}$ pin.

Bit 15: PE15MD1	Bit 14: PE15MD0	Description
0	0	Input/output (PE15) (initial value)
	1	MTU input capture input/output compare output (TIOC4D)
1	0	DMAC request received output (DACK1) (PE15 in single chip mode)
	1	Interrupt request output ($\overline{\text{IRQOUT}}$) (Reserved in single chip mode)

- Bits 13 and 12—PE14 Mode 1, 0 (PE14MD1 and PE14MD0): These bits select the function of the PE14/TIOC4C/DACK0/$\overline{\text{AH}}$ pin.

Bit 13: PE14MD1	Bit 12: PE14MD0	Description
0	0	Input/output (PE14) (initial value)
	1	MTU input capture input/output compare output (TIOC4C)
1	0	DMAC request received output (DACK0) (PE14 in single chip mode)
	1	Address hold output ($\overline{\text{AH}}$) (PE14 in single chip mode)

- Bits 11 and 10—PE13 Mode 1, 0 (PE13MD1 and PE13MD0): These bits select the function of the PE13/TIOC4B/$\overline{\text{MRES}}$ pin.

Bit 11: PE13MD1	Bit 10: PE13MD0	Description
0	0	General input/output (PE13) (initial value)
	1	MTU input capture input/output compare output (TIOC4B)
1	0	Manual reset input ($\overline{\text{MRES}}$)
	1	Reserved

- Bit 9—Reserved: This bit always reads as 0. The write values should always be 0.
- Bit 8—PE12 Mode (PE12MD): Selects the function of the PE12/TIOC4A pin.

Bit 8: PE12MD	Description
0	General input/output (PE12) (initial value)
1	MTU input capture input/output compare output (TIOC4A)

- Bit 7—Reserved: This bit always reads as 0. The write values should always be 0.
- Bit 6—PE11 Mode (PE11MD): Selects the function of the PE11/TIOC3D pin.

Bit 6: PE11MD	Description
0	General input/output (PE11) (initial value)
1	MTU input capture input/output compare output (TIOC3D)

- Bit 5—Reserved: This bit always reads as 0. The write values should always be 0.
- Bit 4—PE10 Mode (PE10MD): Selects the function of the PE10/TIOC3C pin.

Bit 4: PE10MD	Description
0	General input/output (PE10) (initial value)
1	MTU input capture input/output compare output (TIOC3C)

- Bit 3—Reserved: This bit always reads as 0. The write values should always be 0.

- Bit 2—PE9 Mode (PE9MD): Selects the function of the PE9/TIOC3B pin.

Bit 2: PE9MD	Description
0	General input/output (PE9) (initial value)
1	MTU input capture input/output compare output (TIOC3B)

- Bit 1—Reserved: This bit always reads as 0. The write values should always be 0.

- Bit 0—PE8 Mode (PE8MD): Selects the function of the PE8/TIOC3A pin.

Bit 0: PE8MD	Description
0	General input/output (PE8) (initial value)
1	MTU input capture input/output compare output (TIOC3A)

Port E Control Register 2 (PECR2):

Bit:	15	14	13	12	11	10	9	8
	—	PE7MD	—	PE6MD	—	PE5MD	—	PE4MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R/W	R	R/W	R	R/W

Bit:	7	6	5	4	3	2	1	0
	PE3 MD1	PE3 MD0	PE2 MD1	PE2 MD0	PE1 MD1	PE1 MD0	PE0 MD1	PE0 MD0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- Bit 15—Reserved: This bit always reads as 0. The write value should always be 0.

- Bit 14—PE7 Mode (PE7MD): Selects the function of the PE7/TIOC2B pin.

Bit 14: PE7MD	Description
0	General input/output (PE7) (initial value)
1	MTU input capture input/output compare output (TIOC2B)

- Bit 13 —Reserved: This bit always reads as 0. The write value should always be 0.

- Bit 12—PE6 Mode (PE6MD): Selects the function of the PE6/TIOC2A pin.

Bit 12: PE6MD	Description
0	General input/output (PE6) (initial value)
1	MTU input capture input/output compare output (TIOC2A)

- Bit 11—Reserved: This bit always reads as 0. The write value should always be 0.
- Bit 10—PE5 Mode (PE5MD): Selects the function of the PE5/TIOC1B pin.

Bit 10: PE5MD	Description
0	General input/output (PE5) (initial value)
1	MTU input capture input/output compare output (TIOC1B)

- Bit 9—Reserved: This bit always reads as 0. The write value should always be 0.
- Bit 8—PE4 Mode (PE4MD): Selects the function of the PE4/TIOC1A pin.

Bit 8: PE4MD	Description
0	General input/output (PE4) (initial value)
1	MTU input capture input/output compare output (TIOC1A)

- Bits 7 and 6—PE3 Mode 1, 0 (PE3MD1 and PE3MD0): These bits select the function of the PE3/TIOC0D/DRAK1 pin.

Bit 7: PE3MD1	Bit 6: PE3MD0	Description
0	0	General input/output (PE3) (initial value)
	1	MTU input capture input/output compare output (TIOC0D)
1	0	$\overline{\text{DREQ1}}$ request received output (DRAK1) (PE3 in single chip mode)
	1	Reserved

- Bits 5 and 4—PE2 Mode 1, 0 (PE2MD1 and PE2MD0): These bits select the function of the PE2/TIOC0C/$\overline{\text{DREQ1}}$ pin.

Bit 5: PE2MD1	Bit 4: PE2MD0	Description
0	0	General input/output (PE2) (initial value)
	1	MTU input capture input/output compare output (TIOC0C)
1	0	$\overline{\text{DREQ1}}$ request receive input (PE2 in single chip mode)
	1	Reserved

- Bits 3 and 2—PE1 Mode 1, 0 (PE1MD1 and PE1MD0): These bits select the function of the PE1/TIOC0B/DRAK0 pin.

Bit 3: PE1MD1	Bit 2: PE1MD0	Description
0	0	General input/output (PE1) (initial value)
	1	MTU input capture input/output compare output (TIOC0B)
1	0	$\overline{\text{DREQ0}}$ request received output (DRAK0) (PE1 in single chip mode)
	1	Reserved

- Bits 1 and 0—PE0 Mode 1, 0 (PE0MD1 and PE0MD0): These bits select the function of the PE0/TIOC0A/$\overline{\text{DREQ0}}$ pin.

Bit 1: PE0MD1	Bit 0: PE0MD0	Description
0	0	General input/output (PE0) (initial value)
	1	MTU input capture input/output compare output (TIOC0A)
1	0	$\overline{\text{DREQ0}}$ request receive input (PE0 in single chip mode)
	1	Reserved

18.3.15 IRQOUT Function Control Register (IFCR)

The IFCR is a 16-bit read/write register used to control output when the multiplexed pins are established as $\overline{\text{IRQOUT}}$ outputs by the port D control register (PDCRH1) or port E control register (PECR1). When PDCRH1 or PECR1 are set for any other function, the settings of this register have no effect on the pin functions.

The IFCR is initialized to H'0000 by external power-on reset but is not initialized for manual resets, reset by WDT, standby mode, or sleep mode, so the previous data is maintained.

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	IRQ MD3	IRQ MD2	IRQ MD1	IRQ MD0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W

- Bits 3 and 2—IRQOUT Mode 3, 2 (IRQMD3 and IRQMD2): These bits select the $\overline{\text{IRQOUT}}$ pin function when the PDCRH1 bits 13 and 12 (PD30MD1, PD30MD0) are set to (1, 0). These bit settings are effective only for the 144 pin version. Reads and writes are also possible in the 112-pin and 120-pin versions, but they have no effect on the pin functions.

Bit 3: IRQMD3	Bit 2: IRQMD2	Description
0	0	Interrupt request received output (initial value)
	1	Refresh signal output
1	0	Interrupt request received, or refresh signal output (which of the two is output depends on the operation status at the time)
	1	Always high level output

- Bits 1 and 0—IRQOUT Mode 1, 0 (IRQMD1 and IRQMD0): These bits select the $\overline{\text{IRQOUT}}$ pin function when the PECR1 bits 1 and 0 (PE15MD1, PE15MD0) are set to (1, 1).

Bit 1: IRQMD1	Bit 0: IRQMD0	Description
0	0	Interrupt request received output (initial value)
	1	Refresh signal output
1	0	Interrupt request received, or refresh signal output (which of the two is output depends on the operation status at the time)
	1	Always high level output

18.4 Cautions on Use

For the I/O ports and pins with multiplexing of DREQ or IRQ, switching from the port input Low level condition to IRQ or DREQ edge detection will detect the concerned edge.

Section 19 I/O Ports (I/O)

19.1 Overview

There are six ports, A, B, C, D, E, and F. The pins of the ports are multiplexed for use as general-purpose I/Os (the port F pins are general input) or for other functions. Use the pin function controller (PFC) to select the function of multiplexed pins. The ports each have one data register for storing pin data. The initialize function after power-on reset differs depending on the operating mode of each pin. See table 18.2, Pin Arrangement by Mode, for details.

19.2 Port A

There are two versions of port A:

- FP-112/TFP-120
- FP-144

In the FP-112 and TFP-120 versions, port A is a 16-pin input/output port, as listed in table 19.1.

Table 19.1 Port A, FP-112/TFP-120 Version

ROM Disabled Extended Mode (Modes 0, 1)	ROM Enabled Extended Mode (Mode 2)	Single Chip Mode
PA15 (I/O)/CK (output)	PA15 (I/O)/CK (output)	PA15 (I/O)/CK (output)
$\overline{RD}$ (output)	PA14 (I/O)/$\overline{RD}$ (output)	PA14 (I/O)
$\overline{WRH}$ (output)	PA13 (I/O)/$\overline{WRH}$ (output)	PA13 (I/O)
$\overline{WRL}$ (output)	PA12 (I/O)/$\overline{WRL}$ (output)	PA12 (I/O)
$\overline{CS1}$ (output)	PA11 (I/O)/$\overline{CS1}$ (output)	PA11 (I/O)
$\overline{CS0}$ (output)	PA10 (I/O)/$\overline{CS0}$ (output)	PA10 (I/O)
PA9 (I/O)/TCLKD (input)/$\overline{IRQ3}$ (input)	PA9 (I/O)/TCLKD (input)/$\overline{IRQ3}$ (input)	PA9 (I/O)/TCLKD (input)/$\overline{IRQ3}$ (input)
PA8 (I/O)/TCLKC (input)/$\overline{IRQ2}$ (input)	PA8 (I/O)/TCLKC (input)/$\overline{IRQ2}$ (input)	PA8 (I/O)/TCLKC (input)/$\overline{IRQ2}$ (input)
PA7 (I/O)/TCLKB (input)/$\overline{CS3}$ (output)	PA7 (I/O)/TCLKB (input)/$\overline{CS3}$ (output)	PA7 (I/O)/TCLKB (input)
PA6 (I/O)/TCLKA (input)/$\overline{CS2}$ (output)	PA6 (I/O)/TCLKA (input)/$\overline{CS2}$ (output)	PA6 (I/O)/TCLKA (input)
PA5 (I/O)/SCK1 (I/O)/$\overline{DREQ1}$ (input)/$\overline{IRQ1}$ (input)	PA5 (I/O)/SCK1 (I/O)/$\overline{DREQ1}$ (input)/$\overline{IRQ1}$ (input)	PA5 (I/O)/SCK1 (I/O)/$\overline{IRQ1}$ (input)
PA4 (I/O)/TXD1 (output)	PA4 (I/O)/TXD1 (output)	PA4 (I/O)/TXD1 (output)
PA3 (I/O)/RXD1 (input)	PA3 (I/O)/RXD1 (input)	PA3 (I/O)/RXD1 (input)
PA2 (I/O)/SCK0 (I/O)/$\overline{DREQ0}$ (input)/$\overline{IRQ0}$ (input)	PA2 (I/O)/SCK0 (I/O)/$\overline{DREQ0}$ (input)/$\overline{IRQ0}$ (input)	PA2 (I/O)/SCK0 (I/O)/$\overline{IRQ0}$ (input)
PA1 (I/O)/TXD0 (output)	PA1 (I/O)/TXD0 (output)	PA1 (I/O)/TXD0 (output)
PA0 (I/O)/RXD0 (input)	PA0 (I/O)/RXD0 (input)	PA0 (I/O)/RXD0 (input)

In the FP-144 version, port A is a 24-pin input/output port, as listed in table 19.2.

Table 19.2 Port A, FP-144 Version

ROM Disabled Extended Mode (Modes 0, 1)	ROM Enabled Extended Mode (Mode 2)	Single Chip Mode
$\overline{\text{WRHH}}$ (output)	PA23 (I/O)/$\overline{\text{WRHH}}$ (output)	PA23 (I/O)
$\overline{\text{WRHL}}$ (output)	PA22 (I/O)/$\overline{\text{WRHL}}$ (output)	PA22 (I/O)
PA21 (I/O)/$\overline{\text{CASHH}}$ (output)	PA21 (I/O)/$\overline{\text{CASHH}}$ (output)	PA21 (I/O)
PA20 (I/O)/$\overline{\text{CASHL}}$ (output)	PA20 (I/O)/$\overline{\text{CASHL}}$ (output)	PA20 (I/O)
PA19 (I/O)/$\overline{\text{BACK}}$ (output)/ DRAK1 (output)	PA19 (I/O)/$\overline{\text{BACK}}$ (output)/ DRAK1 (output)	PA19 (I/O)
PA18 (I/O)/$\overline{\text{BREQ}}$ (input)/ DRAK0 (output)	PA18 (I/O)/$\overline{\text{BREQ}}$ (input)/ DRAK0 (output)	PA18 (I/O)
PA17 (I/O)/$\overline{\text{WAIT}}$ (input)	PA17 (I/O)/$\overline{\text{WAIT}}$ (input)	PA17 (I/O)
PA16 (I/O)/$\overline{\text{AH}}$ (output)	PA16 (I/O)/$\overline{\text{AH}}$ (output)	PA16 (I/O)
PA15 (I/O)/CK (output)	PA15 (I/O)/CK (output)	PA15 (I/O)/CK (output)
$\overline{\text{RD}}$ (output)	PA14 (I/O)/$\overline{\text{RD}}$ (output)	PA14 (I/O)
$\overline{\text{WRH}}$ (output)	PA13 (I/O)/$\overline{\text{WRH}}$ (output)	PA13 (I/O)
$\overline{\text{WRL}}$ (output)	PA12 (I/O)/$\overline{\text{WRL}}$ (output)	PA12 (I/O)
$\overline{\text{CS1}}$ (output)	PA11 (I/O)/$\overline{\text{CS1}}$ (output)	PA11 (I/O)
$\overline{\text{CS0}}$ (output)	PA10 (I/O)/$\overline{\text{CS0}}$ (output)	PA10 (I/O)
PA9 (I/O)/TCLKD (input)/$\overline{\text{IRQ3}}$ (input)	PA9 (I/O)/TCLKD (input)/$\overline{\text{IRQ3}}$ (input)	PA9 (I/O)/TCLKD (input)/$\overline{\text{IRQ3}}$ (input)
PA8 (I/O)/TCLKC (input)/$\overline{\text{IRQ2}}$ (input)	PA8 (I/O)/TCLKC (input)/$\overline{\text{IRQ2}}$ (input)	PA8 (I/O)/TCLKC (input)/$\overline{\text{IRQ2}}$ (input)
PA7 (I/O)/TCLKB (input)/$\overline{\text{CS3}}$ (output)	PA7 (I/O)/TCLKB (input)/$\overline{\text{CS3}}$ (output)	PA7 (I/O)/TCLKB (input)
PA6 (I/O)/TCLKA (input)/$\overline{\text{CS2}}$ (output)	PA6 (I/O)/TCLKA (input)/$\overline{\text{CS2}}$ (output)	PA6 (I/O)/TCLKA (input)
PA5 (I/O)/SCK1 (I/O)/$\overline{\text{DREQ1}}$ (input)/$\overline{\text{IRQ1}}$ (input)	PA5 (I/O)/SCK1 (I/O)/$\overline{\text{DREQ1}}$ (input)/$\overline{\text{IRQ1}}$ (input)	PA5 (I/O)/SCK1 (I/O)/$\overline{\text{IRQ1}}$ (input)
PA4 (I/O)/TXD1 (output)	PA4 (I/O)/TXD1 (output)	PA4 (I/O)/TXD1 (output)
PA3 (I/O)/RXD1 (input)	PA3 (I/O)/RXD1 (input)	PA3 (I/O)/RXD1 (input)
PA2 (I/O)/SCK0 (I/O)/$\overline{\text{DREQ0}}$ (input)/$\overline{\text{IRQ0}}$ (input)	PA2 (I/O)/SCK0 (I/O)/$\overline{\text{DREQ0}}$ (input)/$\overline{\text{IRQ0}}$ (input)	PA2 (I/O)/SCK0 (I/O)/$\overline{\text{IRQ0}}$ (input)
PA1 (I/O)/TXD0 (output)	PA1 (I/O)/TXD0 (output)	PA1 (I/O)/TXD0 (output)
PA0 (I/O)/RXD0 (input)	PA0 (I/O)/RXD0 (input)	PA0 (I/O)/RXD0 (input)

19.2.1 Register Configuration

Table 19.3 summarizes the port A register.

Table 19.3 Port A Register

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port A data register H	PADRH	R/W	H'0000	H'FFFF8380 H'FFFF8381	8, 16, 32
Port A data register L	PADRL	R/W	H'0000	H'FFFF8382 H'FFFF8383	8, 16, 32

19.2.2 Port A Data Register H (PADRH)

PADRH is a 16-bit read/write register that stores data for port A. The bits PA23DR–PA16DR correspond to the PA23/$\overline{\text{WRHH}}$–PA16/$\overline{\text{AH}}$ pins. When the pins are used as ordinary outputs, they will output whatever value is written in the PADRH; when PADRH is read, the register value will be output regardless of the pin status. When the pins are used as ordinary inputs, the pin status rather than the register value is read directly when PADRH is read. When a value is written to PADRH, that value can be written into PADRH, but it will not affect the pin status. Table 19.4 shows the read/write operations of the port A data register.

PADRH is initialized by an external power-on reset. However, PADRH is not initialized for manual reset, reset by WDT, standby mode, or sleep mode.

These register settings function only for the 144-pin version. There are no pins corresponding to this register in the 112-pin version. However, read/writes are possible.

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	PA23DR	PA22DR	PA21DR	PA20DR	PA19DR	PA18DR	PA17DR	PA16DR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

19.2.3 Port A Data Register L (PADRL)

PADRL is a 16-bit read/write register that stores data for port A. The bits PA15DR–PA0DR correspond to the PA15/CK–PA0/RXD0 pins. When the pins are used as ordinary outputs, they will output whatever value is written in the PADRL; when PADRL is read, the register value will be output regardless of the pin status. When the pins are used as ordinary inputs, the pin status rather than the register value is read directly when PADRL is read. When a value is written to PADRL, that value can be written into PADRL, but it will not affect the pin status. Table 19.4 shows the read/write operations of the port A data register.

PADRL is initialized by an external power-on reset. However, PADRL is not initialized for manual reset, reset by WDT, standby mode, or sleep mode.

Bit:	15	14	13	12	11	10	9	8
	PA15DR	PA14DR	PA13DR	PA12DR	PA11DR	PA10DR	PA9DR	PA8DR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	PA7DR	PA6DR	PA5DR	PA4DR	PA3DR	PA2DR	PA1DR	PA0DR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 19.4 Read/Write Operation of the Port A Data Register (PADR)

PAIOR	Pin Status	Read	Write
0	Ordinary input	Pin status	Can write to PADR, but it has no effect on pin status
	Other function	Pin status	Can write to PADR, but it has no effect on pin status
1	Ordinary output	PADR value	Value written is output by pin
	Other function	PADR value	Can write to PADR, but it has no effect on pin status

19.3 Port B

Port B is a 10-pin input/output port as listed in table 19.5.

Table 19.5 Port B

ROM Disabled Extended Mode (Modes 0, 1)	ROM Enabled Extended Mode (Mode 2)	Single Chip Mode
PB9 (I/O)/$\overline{\text{IRQ7}}$ (input)/A21 (output)/$\overline{\text{ADTRG}}$ (input)	PB9 (I/O)/$\overline{\text{IRQ7}}$ (input)/A21 (output)/$\overline{\text{ADTRG}}$ (input)	PB9 (I/O)/$\overline{\text{IRQ7}}$ (input)/$\overline{\text{ADTRG}}$ (input)
PB8 (I/O)/$\overline{\text{IRQ6}}$ (input)/A20 (output)/$\overline{\text{WAIT}}$ (input)	PB8 (I/O)/$\overline{\text{IRQ6}}$ (input)/A20 (output)/$\overline{\text{WAIT}}$ (input)	PB8 (I/O)/$\overline{\text{IRQ6}}$ (input)
PB7 (I/O)/$\overline{\text{IRQ5}}$ (input)/A19 (output)/$\overline{\text{BREQ}}$ (input)	PB7 (I/O)/$\overline{\text{IRQ5}}$ (input)/A19 (output)/$\overline{\text{BREQ}}$ (input)	PB7 (I/O)/$\overline{\text{IRQ5}}$ (input)
PB6 (I/O)/$\overline{\text{IRQ4}}$ (input)/A18 (output)/$\overline{\text{BACK}}$ (output)	PB6 (I/O)/$\overline{\text{IRQ4}}$ (input)/A18 (output)/$\overline{\text{BACK}}$ (input)	PB6 (I/O)/$\overline{\text{IRQ4}}$ (input)
PB5 (I/O)/$\overline{\text{IRQ3}}$ (input)/$\overline{\text{POE3}}$ (input)/RDWR (output)	PB5 (I/O)/$\overline{\text{IRQ3}}$ (input)/$\overline{\text{POE3}}$ (input)/RDWR (output)	PB5 (I/O)/$\overline{\text{IRQ3}}$ (input)/$\overline{\text{POE3}}$ (input)
PB4 (I/O)/$\overline{\text{IRQ2}}$ (input)/$\overline{\text{POE2}}$ (input)/$\overline{\text{CASH}}$ (output)	PB4 (I/O)/$\overline{\text{IRQ2}}$ (input)/$\overline{\text{POE2}}$ (input)/$\overline{\text{CASH}}$ (output)	PB4 (I/O)/$\overline{\text{IRQ2}}$ (input)/$\overline{\text{POE2}}$ (input)
PB3 (I/O)/$\overline{\text{IRQ1}}$ (input)/$\overline{\text{POE1}}$ (input)/$\overline{\text{CASL}}$ (output)	PB3 (I/O)/$\overline{\text{IRQ1}}$ (input)/$\overline{\text{POE1}}$ (input)/$\overline{\text{CASL}}$ (output)	PB3 (I/O)/$\overline{\text{IRQ1}}$ (input)/$\overline{\text{POE1}}$ (input)
PB2 (I/O)/$\overline{\text{IRQ0}}$ (input)/$\overline{\text{POE0}}$ (input)/$\overline{\text{RAS}}$ (output)	PB2 (I/O)/$\overline{\text{IRQ0}}$ (input)/$\overline{\text{POE0}}$ (input)/$\overline{\text{RAS}}$ (output)	PB2 (I/O)/$\overline{\text{IRQ0}}$ (input)/$\overline{\text{POE0}}$ (input)
A17 (output)	PB1 (I/O)/A17 (output)	PB1 (I/O)
A16 (output)	PB0 (I/O)/A16 (output)	PB0 (I/O)

19.3.1 Register Configuration

Table 19.6 summarizes the port B register.

Table 19.6 Port B Register

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port B data register	PBDR	R/W	H'0000	H'FFFF8390 H'FFFF8391	8, 16, 32

19.3.2 Port B Data Register (PBDR)

PBDR is a 16-bit read/write register that stores data for port B. The bits PB9DR–PB0DR correspond to the PB9/$\overline{\text{IRQ7}}$/A21/$\overline{\text{ADTRG}}$–PB0/A16 pins. When the pins are used as ordinary outputs, they will output whatever value is written in the PBDR; when PBDR is read, the register value will be read regardless of the pin status. When the pins are used as ordinary inputs, the pin status rather than the register value is read directly when PBDR is read. When a value is written to PBDR, that value can be written into PBDR, but it will not affect the pin status. Table 19.7 shows the read/write operations of the port B data register.

PBDR is initialized by an external power-on reset. However, PBDR is not initialized for a manual reset, reset by WDT, standby mode, or sleep mode.

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	PB9DR	PB8DR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	PB7DR	PB6DR	PB5DR	PB4DR	PB3DR	PB2DR	PB1DR	PB0DR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 19.7 Read/Write Operation of the Port B Data Register (PBDR)

PBIOR	Pin Status	Read	Write
0	Ordinary input	Pin status	Can write to PBDR, but it has no effect on pin status
	Other function	Pin status	Can write to PBDR, but it has no effect on pin status
1	Ordinary output	PBDR value	Value written is output by pin
	Other function	PBDR value	Can write to PBDR, but it has no effect on pin status

19.4 Port C

Port C is a 16 pin input/output port as listed in table 19.8.

Table 19.8 Port C

ROM Disabled Extended Mode (Modes 0, 1)	ROM Enabled Extended Mode (Mode 2)	Single Chip Mode
A15 (output)	PC15 (I/O)/A15 (output)	PC15 (I/O)
A14 (output)	PC14 (I/O)/A14 (output)	PC14 (I/O)
A13 (output)	PC13 (I/O)/A13 (output)	PC13 (I/O)
A12 (output)	PC12 (I/O)/A12 (output)	PC12 (I/O)
A11 (output)	PC11 (I/O)/A11 (output)	PC11 (I/O)
A10 (output)	PC10 (I/O)/A10 (output)	PC10 (I/O)
A9 (output)	PC9 (I/O)/A9 (output)	PC9 (I/O)
A8 (output)	PC8 (I/O)/A8 (output)	PC8 (I/O)
A7 (output)	PC7 (I/O)/A7 (output)	PC7 (I/O)
A6 (output)	PC6 (I/O)/A6 (output)	PC6 (I/O)
A5 (output)	PC5 (I/O)/A5 (output)	PC5 (I/O)
A4 (output)	PC4 (I/O)/A4 (output)	PC4 (I/O)
A3 (output)	PC3 (I/O)/A3 (output)	PC3 (I/O)
A2 (output)	PC2 (I/O)/A2 (output)	PC2 (I/O)
A1 (output)	PC1 (I/O)/A1 (output)	PC1 (I/O)
A0 (output)	PC0 (I/O)/A0 (output)	PC0 (I/O)

19.4.1 Register Configuration

Table 19.9 summarizes the port C register.

Table 19.9 Port C Register

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port C data register	PCDR	R/W	H'0000	H'FFFF8392 H'FFFF8393	8, 16, 32

19.4.2 Port C Data Register (PCDR)

PCDR is a 16-bit read/write register that stores data for port C. The bits PC15DR–PC0DR correspond to the PC15/A15–PC0/A0 pins. When the pins are used as ordinary outputs, they will output whatever value is written in the PCDR; when PCDR is read, the register value will be read regardless of the pin status. When the pins are used as ordinary inputs, the pin status rather than the register value is read directly when PCDR is read. When a value is written to PCDR, that value can be written into PCDR, but it will not affect the pin status. Table 19.10 shows the read/write operations of the port C data register.

PCDR is initialized by an external power-on reset. However, PCDR is not initialized for a manual reset, reset by WDT, standby mode, or sleep mode.

Bit:	15	14	13	12	11	10	9	8
	PC15DR	PC14DR	PC13DR	PC12DR	PC11DR	PC10DR	PC9DR	PC8DR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	PC7DR	PC6DR	PC5DR	PC4DR	PC3DR	PC2DR	PC1DR	PC0DR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 19.10 Read/Write Operation of the Port C Data Register (PCDR)

PCIOR	Pin Status	Read	Write
0	Ordinary input	Pin status	Can write to PCDR, but it has no effect on pin status
	Other function	Pin status	Can write to PCDR, but it has no effect on pin status
1	Ordinary output	PCDR value	Value written is output by pin
	Other function	PCDR value	Can write to PCDR, but it has no effect on pin status

19.5 Port D

There are two versions of port D:

- FP-112
- FP-144

In the FP-112 version, port D is a 16-pin input/output port, as shown in table 19.11.

Table 19.11 Port D, FP-112 Version

Extended Mode Without ROM (Mode 0)	Extended Mode Without ROM (Mode 1)	Extended Mode With ROM (Mode 2)	Single Chip Mode
D15 (I/O)	D15 (I/O)	PD15 (I/O)/D15 (I/O)	PD15 (I/O)
D14 (I/O)	D14 (I/O)	PD14 (I/O)/D14 (I/O)	PD14 (I/O)
D13 (I/O)	D13 (I/O)	PD13 (I/O)/D13 (I/O)	PD13 (I/O)
D12 (I/O)	D12 (I/O)	PD12 (I/O)/D12 (I/O)	PD12 (I/O)
D11 (I/O)	D11 (I/O)	PD11 (I/O)/D11 (I/O)	PD11 (I/O)
D10 (I/O)	D10 (I/O)	PD10 (I/O)/D10 (I/O)	PD10 (I/O)
D9 (I/O)	D9 (I/O)	PD9 (I/O)/D9 (I/O)	PD9 (I/O)
D8 (I/O)	D8 (I/O)	PD8 (I/O)/D8 (I/O)	PD8 (I/O)
D7 (I/O)	D7 (I/O)	PD7 (I/O)/D7 (I/O)	PD7 (I/O)
D6 (I/O)	D6 (I/O)	PD6 (I/O)/D6 (I/O)	PD6 (I/O)
D5 (I/O)	D5 (I/O)	PD5 (I/O)/D5 (I/O)	PD5 (I/O)
D4 (I/O)	D4 (I/O)	PD4 (I/O)/D4 (I/O)	PD4 (I/O)
D3 (I/O)	D3 (I/O)	PD3 (I/O)/D3 (I/O)	PD3 (I/O)
D2 (I/O)	D2 (I/O)	PD2 (I/O)/D2 (I/O)	PD2 (I/O)
D1 (I/O)	D1 (I/O)	PD1 (I/O)/D1 (I/O)	PD1 (I/O)
D0 (I/O)	D0 (I/O)	PD0 (I/O)/D0 (I/O)	PD0 (I/O)

In the FP-144 version, port D is a 32-pin input/output port, as listed in table 19.12.

Table 19.12 Port D, FP-144 Version

Extended Mode Without ROM (Mode 0)	Extended Mode Without ROM (Mode 1)	Extended Mode With ROM (Mode 2)	Single Chip Mode
PD31 (I/O)/D31 (I/O)/ $\overline{\text{ADTRG}}$ (input)	D31 (I/O)	PD31 (I/O)/D31 (I/O)/ $\overline{\text{ADTRG}}$ (input)	PD31 (I/O)/$\overline{\text{ADTRG}}$ (input)
PD30 (I/O)/D30 (I/O)/ $\overline{\text{IRQOUT}}$ (output)	D30 (I/O)	PD30 (I/O)/D30 (I/O)/ $\overline{\text{IRQOUT}}$ (output)	PD30 (I/O)/$\overline{\text{IRQOUT}}$ (output)
PD29 (I/O)/D29 (I/O)/ $\overline{\text{CS3}}$ (output)	D29 (I/O)	PD29 (I/O)/D29 (I/O)/ $\overline{\text{CS3}}$ (output)	PD29 (I/O)
PD28 (I/O)/D28 (I/O)/ $\overline{\text{CS2}}$ (output)	D28 (I/O)	PD28 (I/O)/D28 (I/O)/ $\overline{\text{CS2}}$ (output)	PD28 (I/O)
PD27 (I/O)/D27 (I/O)/ DACK1 (output)	D27 (I/O)	PD27 (I/O)/D27 (I/O)/ DACK1 (output)	PD27 (I/O)
PD26 (I/O)/D26 (I/O)/ DACK0 (output)	D26 (I/O)	PD26 (I/O)/D26 (I/O)/ DACK0 (output)	PD26 (I/O)
PD25 (I/O)/D25 (I/O)/ $\overline{\text{DREQ1}}$ (input)	D25 (I/O)	PD25 (I/O)/D25 (I/O)/ $\overline{\text{DREQ1}}$ (input)	PD25 (I/O)
PD24 (I/O)/D24 (I/O)/ $\overline{\text{DREQ0}}$ (input)	D24 (I/O)	PD24 (I/O)/D24 (I/O)/ $\overline{\text{DREQ0}}$ (input)	PD24 (I/O)
PD23 (I/O)/D23 (I/O)/ $\overline{\text{IRQ7}}$ (input)	D23 (I/O)	PD23 (I/O)/D23 (I/O)/ $\overline{\text{IRQ7}}$ (input)	PD23 (I/O)/$\overline{\text{IRQ7}}$ (input)
PD22 (I/O)/D22 (I/O)/ $\overline{\text{IRQ6}}$ (input)	D22 (I/O)	PD22 (I/O)/D22 (I/O)/ $\overline{\text{IRQ6}}$ (input)	PD22 (I/O)/$\overline{\text{IRQ6}}$ (input)
PD21 (I/O)/D21 (I/O)/ $\overline{\text{IRQ5}}$ (input)	D21 (I/O)	PD21 (I/O)/D21 (I/O)/ $\overline{\text{IRQ5}}$ (input)	PD21 (I/O)/$\overline{\text{IRQ5}}$ (input)
PD20 (I/O)/D20 (I/O)/ $\overline{\text{IRQ4}}$ (input)	D20 (I/O)	PD20 (I/O)/D20 (I/O)/ $\overline{\text{IRQ4}}$ (input)	PD20 (I/O)/$\overline{\text{IRQ4}}$ (input)
PD19 (I/O)/D19 (I/O)/ $\overline{\text{IRQ3}}$ (input)	D19 (I/O)	PD19 (I/O)/D19 (I/O)/ $\overline{\text{IRQ3}}$ (input)	PD19 (I/O)/$\overline{\text{IRQ3}}$ (input)
PD18 (I/O)/D18 (I/O)/ $\overline{\text{IRQ2}}$ (input)	D18 (I/O)	PD18 (I/O)/D18 (I/O)/ $\overline{\text{IRQ2}}$ (input)	PD18 (I/O)/$\overline{\text{IRQ2}}$ (input)
PD17 (I/O)/D17 (I/O)/ $\overline{\text{IRQ1}}$ (input)	D17 (I/O)	PD17 (I/O)/D17 (I/O)/ $\overline{\text{IRQ1}}$ (input)	PD17 (I/O)/$\overline{\text{IRQ1}}$ (input)
PD16 (I/O)/D16 (I/O)/ $\overline{\text{IRQ0}}$ (input)	D16 (I/O)	PD16 (I/O)/D16 (I/O)/ $\overline{\text{IRQ0}}$ (input)	PD16 (I/O)/$\overline{\text{IRQ0}}$ (input)

Table 19.12 Port D, FP-144 Version (cont)

Extended Mode Without ROM (Mode 0)	Extended Mode Without ROM (Mode 1)	Extended Mode With ROM (Mode 2)	Single Chip Mode
D15 (I/O)	D15 (I/O)	PD15 (I/O)/D15 (I/O)	PD15 (I/O)
D14 (I/O)	D14 (I/O)	PD14 (I/O)/D14 (I/O)	PD14 (I/O)
D13 (I/O)	D13 (I/O)	PD13 (I/O)/D13 (I/O)	PD13 (I/O)
D12 (I/O)	D12 (I/O)	PD12 (I/O)/D12 (I/O)	PD12 (I/O)
D11 (I/O)	D11 (I/O)	PD11 (I/O)/D11 (I/O)	PD11 (I/O)
D10 (I/O)	D10 (I/O)	PD10 (I/O)/D10 (I/O)	PD10 (I/O)
D9 (I/O)	D9 (I/O)	PD9 (I/O)/D9 (I/O)	PD9 (I/O)
D8 (I/O)	D8 (I/O)	PD8 (I/O)/D8 (I/O)	PD8 (I/O)
D7 (I/O)	D7 (I/O)	PD7 (I/O)/D7 (I/O)	PD7 (I/O)
D6 (I/O)	D6 (I/O)	PD6 (I/O)/D6 (I/O)	PD6 (I/O)
D5 (I/O)	D5 (I/O)	PD5 (I/O)/D5 (I/O)	PD5 (I/O)
D4 (I/O)	D4 (I/O)	PD4 (I/O)/D4 (I/O)	PD4 (I/O)
D3 (I/O)	D3 (I/O)	PD3 (I/O)/D3 (I/O)	PD3 (I/O)
D2 (I/O)	D2 (I/O)	PD2 (I/O)/D2 (I/O)	PD2 (I/O)
D1 (I/O)	D1 (I/O)	PD1 (I/O)/D1 (I/O)	PD1 (I/O)
D0 (I/O)	D0 (I/O)	PD0 (I/O)/D0 (I/O)	PD0 (I/O)

19.5.1 Register Configuration

Table 19.13 summarizes the port D register.

Table 19.13 Port D Register

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port D data register H	PDDRH	R/W	H'0000	H'FFFF83A0 H'FFFF83A1	8, 16, 32
Port D data register L	PDDRL	R/W	H'0000	H'FFFF83A2 H'FFFF83A3	8, 16, 32

19.5.2 Port D Data Register H (PDDRH)

PDDRH is a 16-bit read/write register that stores data for port D. The bits PD31DR–PD16DR correspond to the PD31/D31/$\overline{\text{ADTRG}}$–PD16/D16/$\overline{\text{IRQ0}}$ pins. When the pins are used as ordinary outputs, they will output whatever value is written in the PDDRH; when PDDRH is read, the register value will be read regardless of the pin status. When the pins are used as ordinary inputs, the pin status rather than the register value is read directly when PDDRH is read. When a value is written to PDDRH, that value can be written into PDDRH, but it will not affect the pin status. Table 19.14 shows the read/write operations of the port D data register.

PDDRH is initialized by an external power-on reset. However, PDDRH is not initialized for a manual reset, reset by WDT, standby mode, or sleep mode.

These register settings function only for the 144-pin version. There are no pins corresponding to this register in the 112-pin version. However, read/writes are possible.

Bit:	15	14	13	12	11	10	9	8
	PD31DR	PD30DR	PD29DR	PD28DR	PD27DR	PD26DR	PD25DR	PD24DR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	PD23DR	PD22DR	PD21DR	PD20DR	PD19DR	PD18DR	PD17DR	PD16DR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

19.5.3 Port D Data Register L (PDDRL)

PDDRL is a 16-bit read/write register that stores data for port D. The bits PD15DR–PD0DR correspond to the PD15/D15–PD0/D0 pins. When the pins are used as ordinary outputs, they will output whatever value is written in the PDDRL; when PDDRL is read, the register value will be read regardless of the pin status. When the pins are used as ordinary inputs, the pin status rather than the register value is read directly when PDDRL is read. When a value is written to PDDRL, that value can be written into PDDRL, but it will not affect the pin status. Table 19.14 shows the read/write operations of the port D data register.

PDDRL is initialized by an external power-on reset. However, PDDRL is not initialized for a manual reset, reset by WDT, standby mode, or sleep mode.

Bit:	15	14	13	12	11	10	9	8
	PD15DR	PD14DR	PD13DR	PD12DR	PD11DR	PD10DR	PD9DR	PD8DR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	PD7DR	PD6DR	PD5DR	PD4DR	PD3DR	PD2DR	PD1DR	PD0DR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 19.14 Read/Write Operation of the Port D Data Register (PDDR)

PDIOR	Pin Status	Read	Write
0	Ordinary input	Pin status	Can write to PDDR, but it has no effect on pin status
	Other function	Pin status	Can write to PDDR, but it has no effect on pin status
1	Ordinary output	PDDR value	Value written is output by pin
	Other function	PDDR value	Can write to PDDR, but it has no effect on pin status

19.6 Port E

Port E is a 16-pin input/output port, as listed in table 19.15.

Table 19.15 Port E

Extended Modes (Modes 0, 1, 2)	Single Chip Mode
PE15 (I/O)/TIOC4D (I/O)/DACK1 (output)/$\overline{\text{IRQOUT}}$ (output)	PE15 (I/O)/TIOC4D (I/O)/$\overline{\text{IRQOUT}}$ (output)
PE14 (I/O)/TIOC4C (I/O)/DACK0 (output)/$\overline{\text{AH}}$ (output)	PE14 (I/O)/TIOC4C (I/O)
PE13 (I/O)/TIOC4B (I/O)/$\overline{\text{MRES}}$ (input)	PE13 (I/O)/TIOC4B (I/O)/$\overline{\text{MRES}}$ (input)
PE12 (I/O)/TIOC4A (I/O)	PE12 (I/O)/TIOC4A (I/O)
PE11 (I/O)/TIOC3D (I/O)	PE11 (I/O)/TIOC3D (I/O)
PE10 (I/O)/TIOC3C (I/O)	PE10 (I/O)/TIOC3C (I/O)
PE9 (I/O)/TIOC3B (I/O)	PE9 (I/O)/TIOC3B (I/O)
PE8 (I/O)/TIOC3A (I/O)	PE8 (I/O)/TIOC3A (I/O)
PE7 (I/O)/TIOC2B (I/O)	PE7 (I/O)/TIOC2B (I/O)
PE6 (I/O)/TIOC2A (I/O)	PE6 (I/O)/TIOC2A (I/O)
PE5 (I/O)/TIOC1B (I/O)	PE5 (I/O)/TIOC1B (I/O)
PE4 (I/O)/TIOC1A (I/O)	PE4 (I/O)/TIOC1A (I/O)
PE3 (I/O)/TIOC0D (I/O)/DRAK1 (output)	PE3 (I/O)/TIOC0D (I/O)
PE2 (I/O)/TIOC0C (I/O)/$\overline{\text{DREQ1}}$ (input)	PE2 (I/O)/TIOC0C (I/O)
PE1 (I/O)/TIOC0B (I/O)/DRAK0 (output)	PE1 (I/O)/TIOC0B (I/O)
PE0 (I/O)/TIOC0A (I/O)/$\overline{\text{DREQ0}}$ (input)	PE0 (I/O)/TIOC0A (I/O)

19.6.1 Register Configuration

Table 19.16 summarizes the port E register.

Table 19.16 Port E Register

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port E data register	PEDR	R/W	H'0000	H'FFFF83B0 H'FFFF83B1	8, 16, 32

19.6.2 Port E Data Register (PEDR)

PEDR is a 16-bit read/write register that stores data for port E. The bits PE15DR–PE0DR correspond to the PE15/TIOC4D/DACK1/$\overline{\text{IRQOUT}}$–PE0/TIOC0A/$\overline{\text{DREQ0}}$ pins. When the pins are used as ordinary outputs, they will output whatever value is written in the PEDR; when PEDR is read, the register value will be read regardless of the pin status. When the pins are used as ordinary inputs, the pin status rather than the register value is read directly when PEDR is read. When a value is written to PEDR, that value can be written into PEDR, but it will not affect the pin status. Table 19.17 shows the read/write operations of the port E data register.

PEDR is initialized by a external power-on reset. However, PEDR is not initialized for a manual reset, reset by WDT, standby mode, or sleep mode, so the previous data is retained.

Bit:	15	14	13	12	11	10	9	8
	PE15DR	PE14DR	PE13DR	PE12DR	PE11DR	PE10DR	PE9DR	PE8DR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	PE7DR	PE6DR	PE5DR	PE4DR	PE3DR	PE2DR	PE1DR	PE0DR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 19.17 Read/Write Operation of the Port E Data Register (PEDR)

PEIOR	Pin Status	Read	Write
0	Ordinary input	Pin status	Can write to PEDR, but it has no effect on pin status
	Other function	Pin status	Can write to PEDR, but it has no effect on pin status
1	Ordinary output	PEDR value	Value written is output by pin
	Other function	PEDR value	Can write to PEDR, but it has no effect on pin status

19.7 Port F

Port F is an 8-pin input port. All modes are configured in the following way:

- PF7 (input)/AN7 (input)
- PF6 (input)/AN6 (input)
- PF5 (input)/AN5 (input)
- PF4 (input)/AN4 (input)
- PF3 (input)/AN3 (input)
- PF2 (input)/AN2 (input)
- PF1 (input)/AN1 (input)
- PF0 (input)/AN0 (input)

19.7.1 Register Configuration

Table 19.18 summarizes the port F register.

Table 19.18 Port F Register

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port F data register	PFDR	R	External pin dependent	H'FFFF83B3	8

19.7.2 Port F Data Register (PFDR)

PFDR is an 8-bit read-only register that stores data for port F. The bits PF7DR–PF0DR correspond to the PF7/AN7–PF0/AN0 pins. There are no bits 15–8, so always access as eight bits. Any value written into these bits is ignored, and there is no effect on the status of the pins. When any of the bits are read, the pin status rather than the bit value is read directly. However, when an A/D converter analog input is being sampled, values of 1 are read out. Table 19.19 shows the read/write operations of the port F data register.

PFDR is not initialized by power-on resets, manual resets, standby mode, or sleep mode (the bits always reflect the pin status).

Bit:	7	6	5	4	3	2	1	0
	PF7DR	PF6DR	PF5DR	PF4DR	PF3DR	PF2DR	PF1DR	PF0DR
Initial value:	*	*	*	*	*	*	*	*
R/W:	R	R	R	R	R	R	R	R

Note: * Initial values are dependent on the status of the pins at the time of the reads.

Table 19.19 Read/Write Operation of the Port F Data Register (PFDR)

Pin I/O	Pin Function	Read	Write
Input	Ordinary	Pin status is read	Ignored (no effect on pin status)
	ANn: analog input	1 is read	Ignored (no effect on pin status)

n=7-0

20.1 Overview

This LSI is available with 64 kbytes, 128 kbytes, or 256 kbytes of on-chip ROM. The on-chip ROM is connected to the CPU, direct memory access controller (DMAC) and data transfer controller (DTC) through a 32-bit data bus (figures 20.1, 20.2, and 20.3). The CPU, DMAC, and DTC can access the on-chip ROM in 8, 16, and 32-bit widths. Data in the on-chip ROM can always be accessed in one cycle.

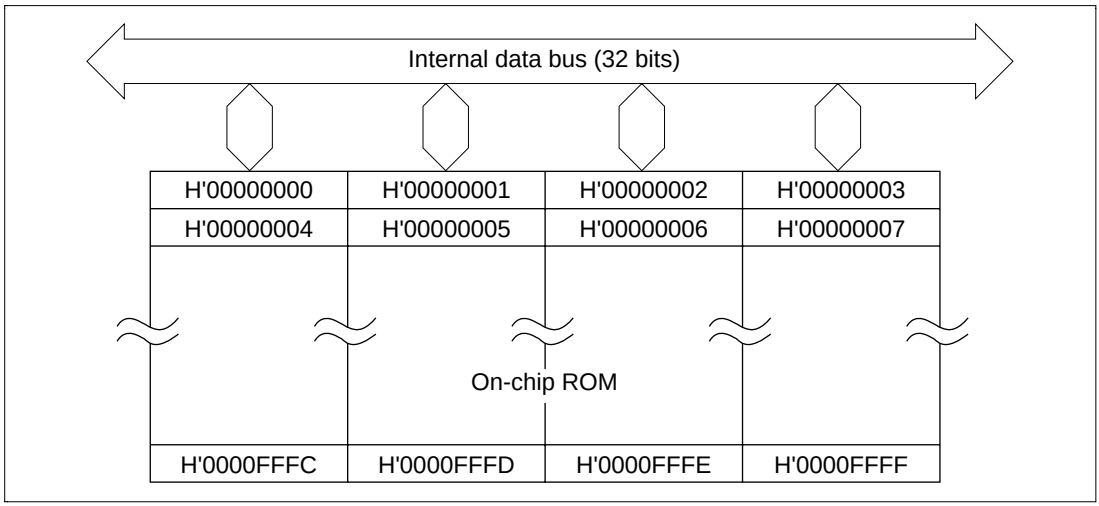


Figure 20.1 Mask ROM Block Diagram (64-kbyte Version)

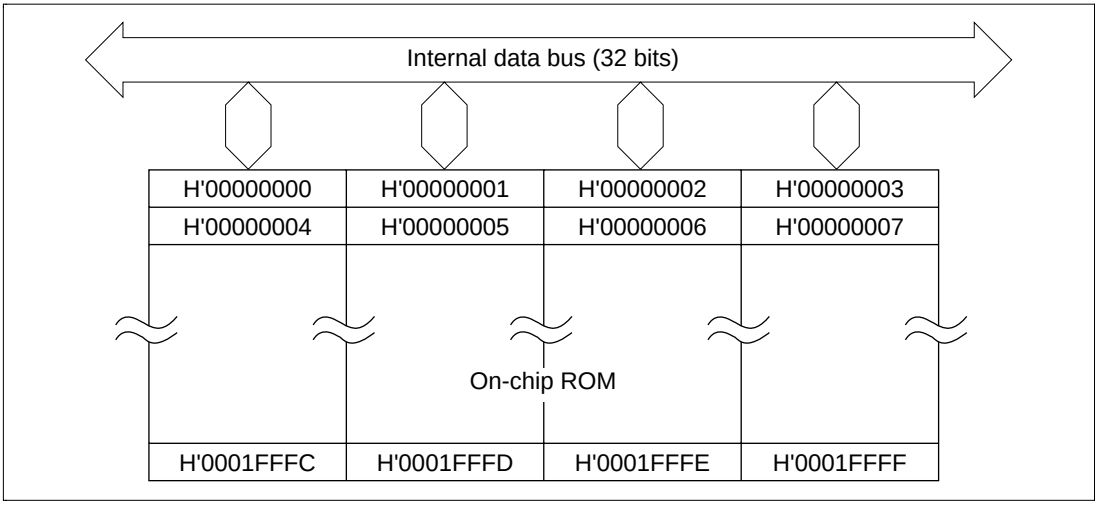


Figure 20.2 Mask ROM Block Diagram (128-kbyte Version)

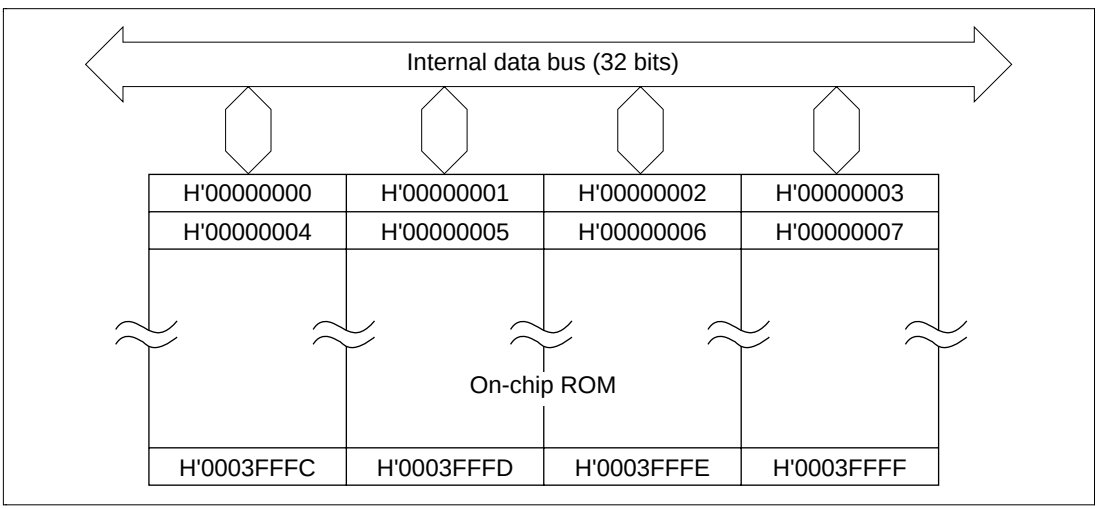


Figure 20.3 Mask ROM Block Diagram (256-kbyte Version)

The operating mode determines whether the on-chip ROM is valid or not. The operating mode is selected using mode-setting pins MD3–MD0 as shown in table 20.1. If you are using the on-chip ROM, select mode 2 or mode 3; if you are not, select mode 0 or 1. The on-chip ROM is allocated to addresses H'00000000–H'0000FFFF of memory area 0 for the 64-kbyte version, H'00000000–H'0001FFFF of memory area 0 for the 128-kbyte version and H'00000000–H'0003FFFF of memory area 0 for the 256-kbyte version.

Table 20.1 Operation Modes and ROM

	Mode Setting Pin				
Operation Mode	**MD3**	**MD2**	**MD1**	**MD0**	**Area 0**
Mode 0 (MCU mode 0)	*	*	0	0	On-chip ROM invalid, external 8-bit space (112 pin and 120 pin), external 16-bit space (144 pin)
Mode 1 (MCU mode 1)	*	*	0	1	On-chip ROM invalid, external 16-bit space (112 pin and 120 pin), external 32-bit space (144 pin)
Mode 2 (MCU mode 2)	*	*	1	0	On-chip ROM valid, external space (bus width set with bus state controller)
Mode 3 (MCU mode 3)	*	*	1	1	On-chip ROM valid, single-chip mode

0: Low

1: High

*: Refer to section 3, Operating Modes.

21.1 Overview

This LSI has 128 kbytes of on-chip PROM.The on-chip ROM is connected to the CPU, the direct memory access controller (DMAC) and the data transfer controller (DTC) through a 32-bit data bus (figures 21.1). The CPU, DMAC, and DTC can access the on-chip ROM in 8, 16, and 32-bit widths. Data in the on-chip ROM can always be accessed in one cycle.

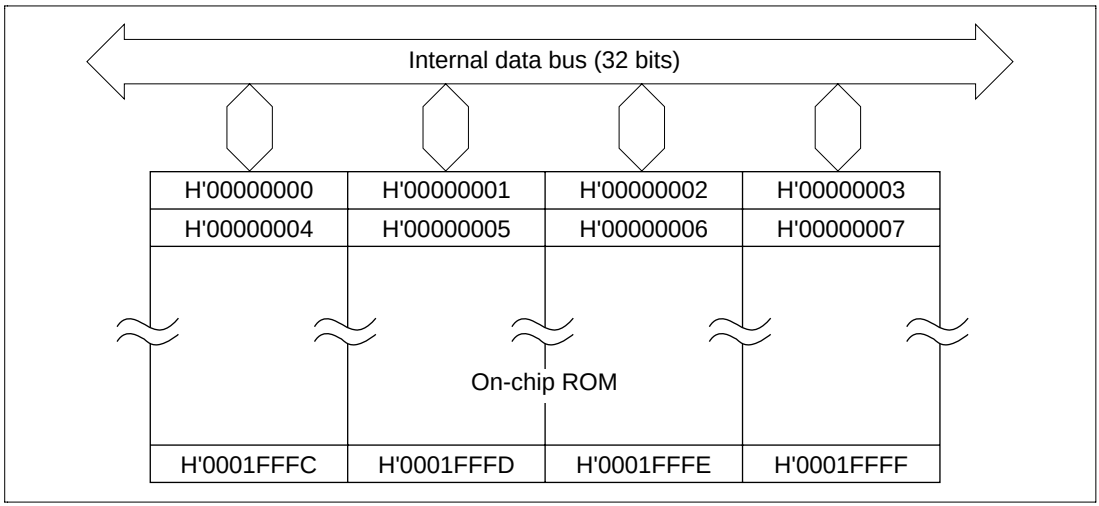


Figure 21.1 PROM Block Diagram

The operating mode determines whether the on-chip ROM is valid or not. The operating mode is selected using mode-setting pins MD3–MD0 as shown in table 21.1. If you are using the on-chip ROM, select mode 2 or mode 3; if you are not, select mode 0 or 1. The on-chip ROM is allocated to addresses H'00000000–H'0001FFFF of memory area 0.

Table 21.1 Operating Modes and ROM

	Mode Setting Pin				
Operating Mode	**MD3**	**MD2**	**MD1**	**MD0**	**Area 0**
Mode 0 (MCU mode 0)	*	*	0	0	On-chip ROM invalid, external 8-bit space (112 pin and 120 pin), external 16-bit space (144 pin)
Mode 1 (MCU mode 1)	*	*	0	1	On-chip ROM invalid, external 16-bit space (112 pin and 120 pin), external 32-bit space (144 pin)
Mode 2 (MCU mode 2)	*	*	1	0	On-chip ROM valid, with an external space (bus width setting is performed by the bus state controller)
Mode 3 (MCU mode 3)	*	*	1	1	On-chip ROM valid, single chip mode
Mode 7 (PROM mode)	1	1	1	1	—

0: Low

1: High

*: Refer to section 3, Operating Modes.

With the PROM version, programs can be written in the same manner as with an ordinary EPROM by setting the LSI to PROM mode and using a standard EPROM writer.

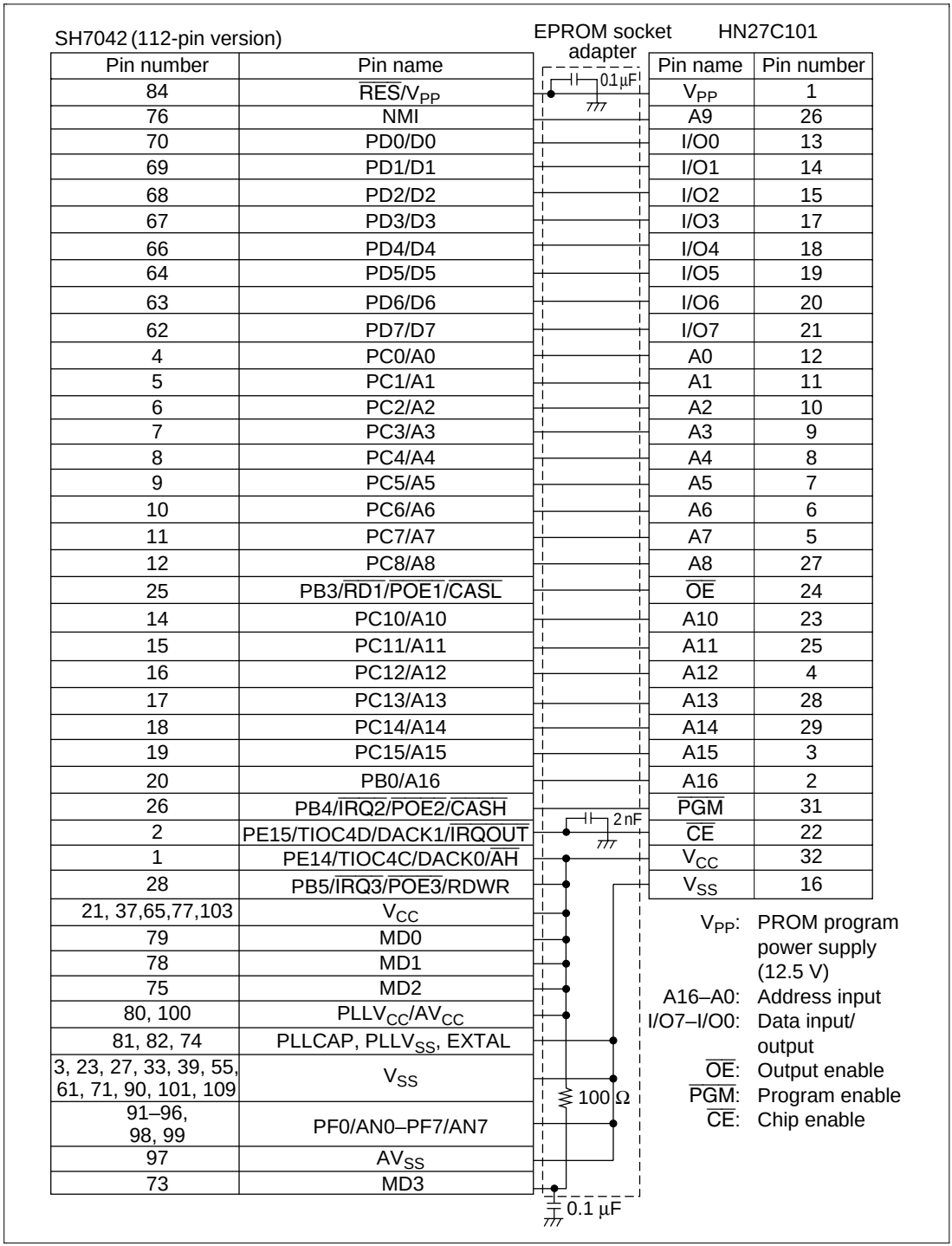
21.2 PROM Mode

21.2.1 PROM Mode Settings

When programming the on-chip PROM, set the pins as shown in figure 21.2, 21.3, or 21.4, and perform the programming in PROM mode.

21.2.2 Socket Adapter Pin Correspondence and Memory Map

Connect the socket adapter to the SH7040 series chip as shown in figure 21.2 or 21.3. This will allow the on-chip PROM to be programmed in the same manner as an ordinary 32-pin EPROM (HN27C101). Figures 21.2, 21.3, and 21.4 show the correspondence between the SH7040 Series pins and HN27C101 pins. Figure 21.5 is a memory map of the on-chip ROM.



Pin number	Pin name
84	$\overline{RES}/V_{PP}$
76	NMI
70	PD0/D0
69	PD1/D1
68	PD2/D2
67	PD3/D3
66	PD4/D4
64	PD5/D5
63	PD6/D6
62	PD7/D7
4	PC0/A0
5	PC1/A1
6	PC2/A2
7	PC3/A3
8	PC4/A4
9	PC5/A5
10	PC6/A6
11	PC7/A7
12	PC8/A8
25	PB3/$\overline{RD}$1/$\overline{POE}$1/$\overline{CASL}$
14	PC10/A10
15	PC11/A11
16	PC12/A12
17	PC13/A13
18	PC14/A14
19	PC15/A15
20	PB0/A16
26	PB4/$\overline{IRQ2}$/$\overline{POE2}$/$\overline{CASH}$
2	PE15/TIOC4D/DACK1/$\overline{IRQOUT}$
1	PE14/TIOC4C/DACK0/$\overline{AH}$
28	PB5/$\overline{IRQ3}$/$\overline{POE3}$/RDWR
21, 37,65,77,103	V_{CC}
79	MD0
78	MD1
75	MD2
80, 100	$PLLV_{CC}/AV_{CC}$
81, 82, 74	PLLCAP, $PLLV_{SS}$, EXTAL
3, 23, 27, 33, 39, 55, 61, 71, 90, 101, 109	V_{SS}
91–96, 98, 99	PF0/AN0–PF7/AN7
97	AV_{SS}
73	MD3

Pin name	Pin number
V_{PP}	1
A9	26
I/O0	13
I/O1	14
I/O2	15
I/O3	17
I/O4	18
I/O5	19
I/O6	20
I/O7	21
A0	12
A1	11
A2	10
A3	9
A4	8
A5	7
A6	6
A7	5
A8	27
$\overline{OE}$	24
A10	23
A11	25
A12	4
A13	28
A14	29
A15	3
A16	2
$\overline{PGM}$	31
$\overline{CE}$	22
V_{CC}	32
V_{SS}	16

Figure 21.2 SH7042 Pin and HN27C101 Pin Correspondence (112-Pin Version)

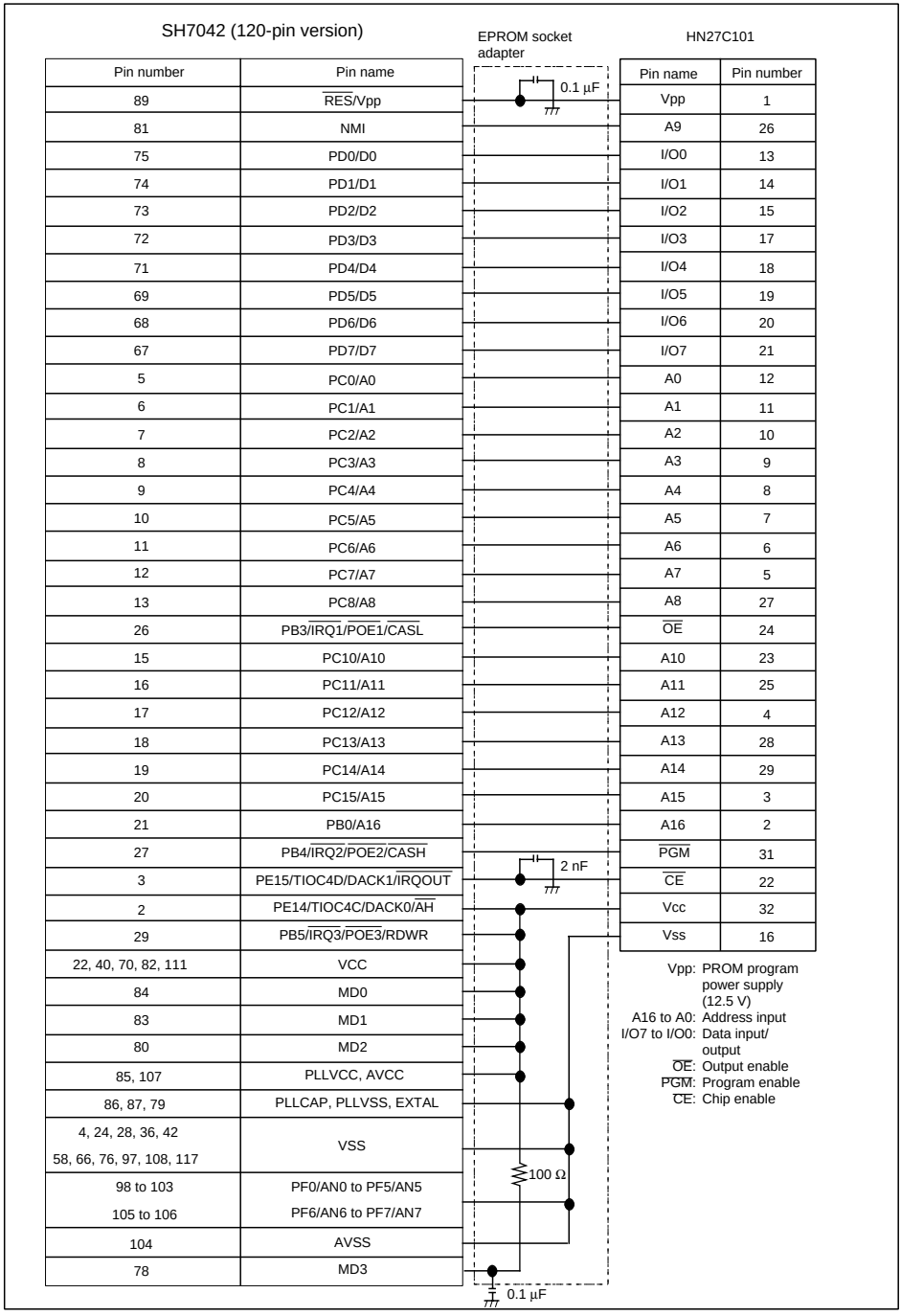
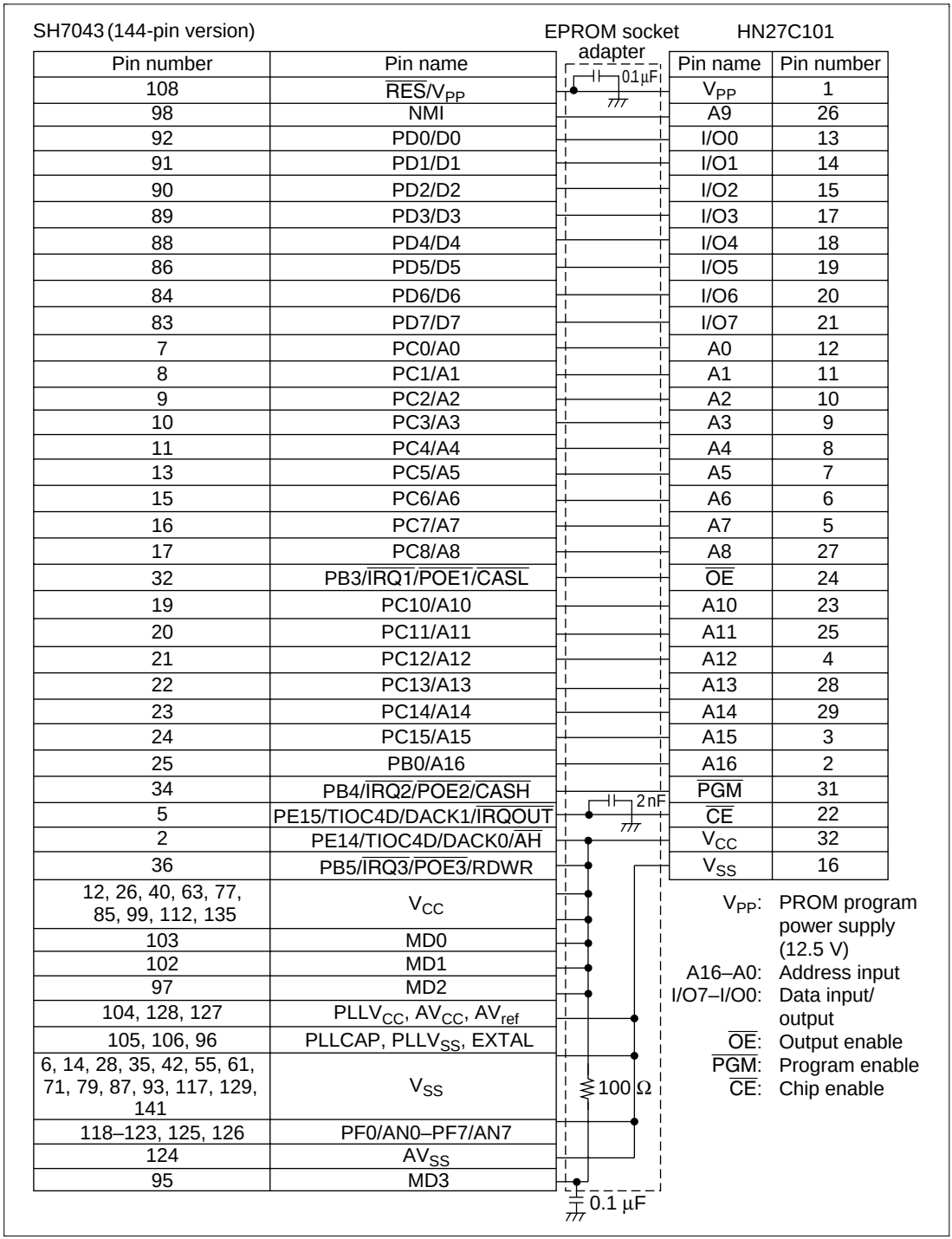


Figure 21.3 SH7042 Pin and HN27C101 Pin Correspondence (120-Pin Version)



Pin number	Pin name
108	$\overline{RES}/V_{PP}$
98	NMI
92	PD0/D0
91	PD1/D1
90	PD2/D2
89	PD3/D3
88	PD4/D4
86	PD5/D5
84	PD6/D6
83	PD7/D7
7	PC0/A0
8	PC1/A1
9	PC2/A2
10	PC3/A3
11	PC4/A4
13	PC5/A5
15	PC6/A6
16	PC7/A7
17	PC8/A8
32	PB3/$\overline{IRQ1}$/$\overline{POE1}$/$\overline{CASL}$
19	PC10/A10
20	PC11/A11
21	PC12/A12
22	PC13/A13
23	PC14/A14
24	PC15/A15
25	PB0/A16
34	PB4/$\overline{IRQ2}$/$\overline{POE2}$/$\overline{CASH}$
5	PE15/TIOC4D/DACK1/$\overline{IRQOUT}$
2	PE14/TIOC4D/DACK0/$\overline{AH}$
36	PB5/$\overline{IRQ3}$/$\overline{POE3}$/RDWR
12, 26, 40, 63, 77, 85, 99, 112, 135	V_{CC}
103	MD0
102	MD1
97	MD2
104, 128, 127	$PLLV_{CC}$, AV_{CC}, AV_{ref}
105, 106, 96	PLLCAP, $PLLV_{SS}$, EXTAL
6, 14, 28, 35, 42, 55, 61, 71, 79, 87, 93, 117, 129, 141	V_{SS}
118–123, 125, 126	PF0/AN0–PF7/AN7
124	AV_{SS}
95	MD3

Pin name	Pin number
V_{PP}	1
A9	26
I/O0	13
I/O1	14
I/O2	15
I/O3	17
I/O4	18
I/O5	19
I/O6	20
I/O7	21
A0	12
A1	11
A2	10
A3	9
A4	8
A5	7
A6	6
A7	5
A8	27
$\overline{OE}$	24
A10	23
A11	25
A12	4
A13	28
A14	29
A15	3
A16	2
$\overline{PGM}$	31
$\overline{CE}$	22
V_{CC}	32
V_{SS}	16

Figure 21.4 SH7043 Pin and HN27C101 Pin Correspondence (144-Pin Version)

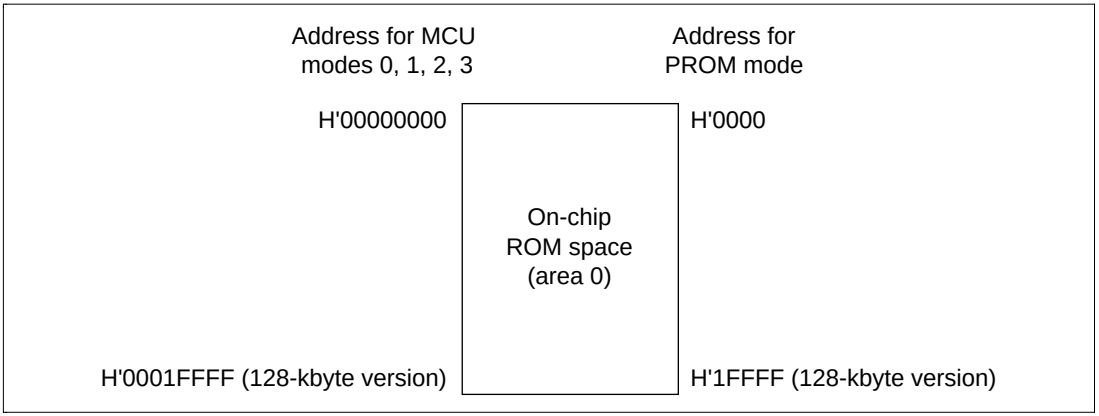


Figure 21.5 On-Chip ROM Memory Map

21.3 PROM Programming

The PROM mode write/verify specifications are the same as those of the standard EPROM HN27C101. However, because the page program format is not supported, do not set the PROM writer to the page programming mode. PROM writers that only support page programming mode cannot be used. When selecting a PROM writer, confirm that it supports the byte-by-byte high-speed, high-reliability programming format.

21.3.1 Programming Mode Selection

There are two on-chip PROM programming modes: write and verify (reads and confirms written data). The mode is selected by using the pins (table 21.2).

Table 21.2 PROM Programming Mode Selection

	Pin						
Mode	**$\overline{CE}$**	**$\overline{OE}$**	**$\overline{PGM}$**	**V_{PP}**	**V_{CC}**	**I/O7–I/O0**	**A16–A0**
Write	0	1	0	V_{PP}	V_{CC}	Data input	Address input
Verify	0	0	1			Data output	
Programming Prohibited	0	0	0			High impedance	
	0	1	1				
	1	0	0				
	1	1	1				

Note: 0: low level, 1: high level, V_{PP}: V_{PP} level, V_{CC}: V_{CC} level.

21.3.2 Write/Verify and Electrical Characteristics

Write/Verify: Writing and verification can be done using an efficient high speed, high reliability programming format. This format allows data writing that is both fast and reliable without applying voltage stress to the device. Figure 21.6 shows the basic flow of the high speed, high reliability programming format.

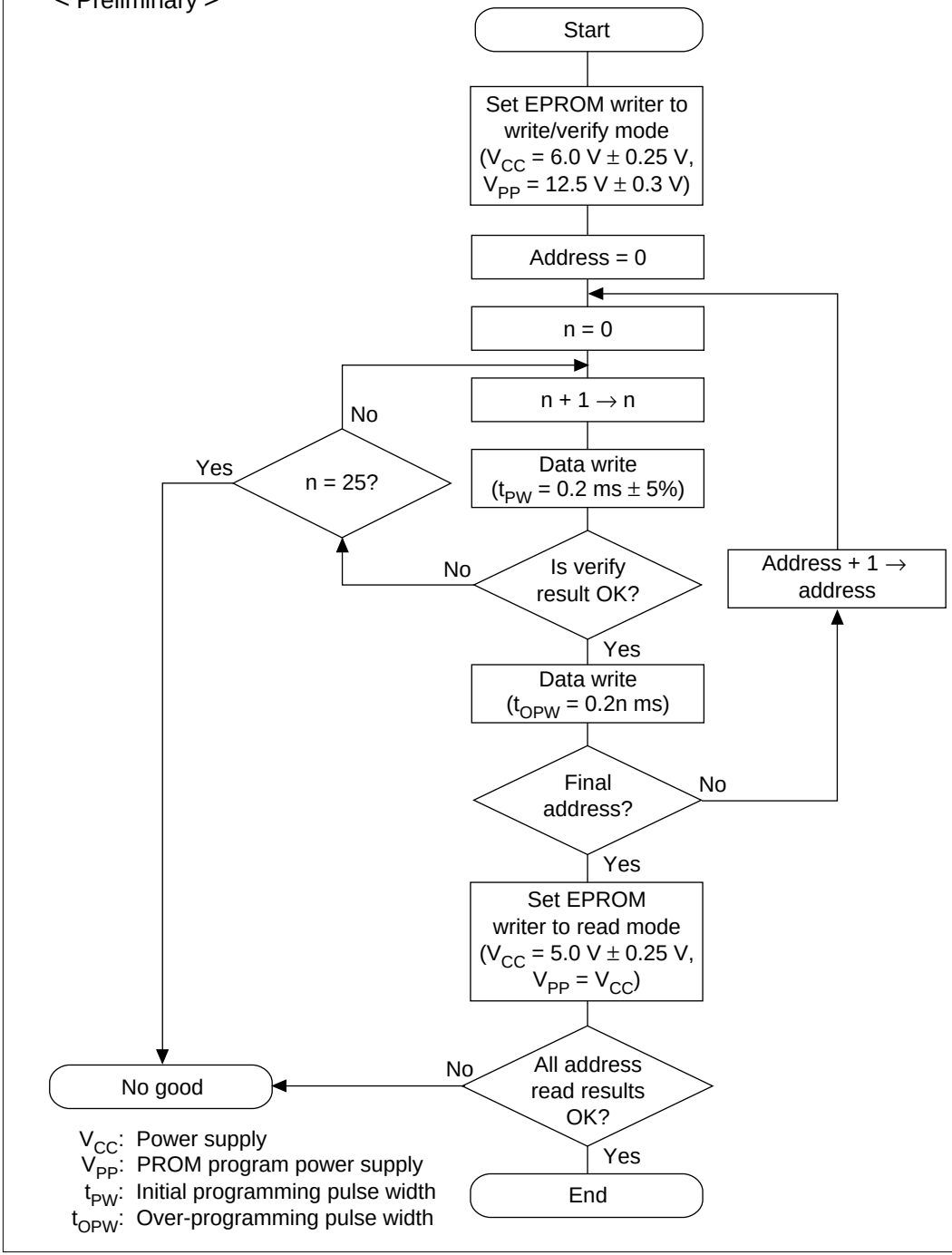


Figure 21.6 High-Speed, High-Reliability Programming Basic Flow

Electrical Characteristics: Tables 21.3 and 21.4 show the electrical characteristics for programming. Figure 21.7 shows the timing.

Table 21.3 DC Characteristics (V_{CC} = 6.0 V ± 0.25 V, V_{PP} = 12.5 V ± 0.3 V, V_{SS} = 0 V, Ta = 25°C ± 5°C)

Item	Pin	Symbol	Min	Typ	Max	Unit	Measurement Conditions
Input high-level voltage	I/O7–I/O0, A16–A0, $\overline{OE}$, $\overline{CE}$, $\overline{PGM}$	V_{IH}	2.4	—	V_{CC} + 0.3	V	
Input low-level voltage	I/O7–I/O0, A16–A0, $\overline{OE}$, $\overline{CE}$, $\overline{PGM}$	V_{IL}	–0.3	—	0.8	V	
Output high-level voltage	I/O7–I/O0	V_{OH}	2.4	—	—	V	I_{OH} = –200 µA
Output low-level voltage	I/O7–I/O0	V_{OL}	—	—	0.45	V	I_{OL} = 1.6 mA
Input leak current	I/O7–I/O0, A16–A0, $\overline{OE}$, $\overline{CE}$, $\overline{PGM}$	$\lvert I_{LI} \rvert$	—	—	2	µA	V_{IN} = 5.25 V/0.5 V
V_{CC} current		I_{CC}	—	—	80	mA	
V_{PP} current		I_{PP}	—	—	80	mA	

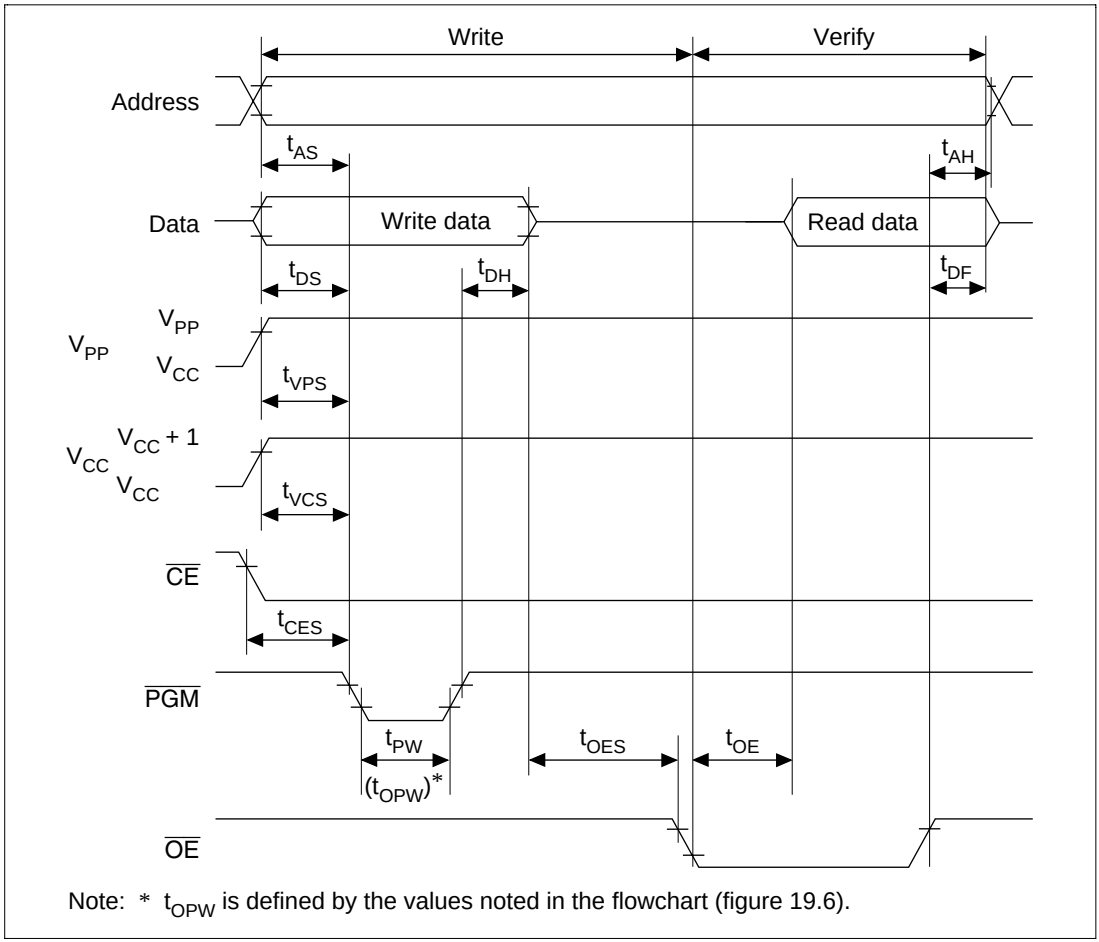
Table 21.4 AC Characteristics (V_{CC} = 6.0 V ± 0.25 V, V_{PP} = 12.5 V ± 0.3 V, V_{SS} = 0 V, Ta = 25°C ± 5°C)

Item	Symbol	Min	Typ	Max	Unit	Measurement Conditions
Address setup time	t_{AS}	2	—	—	µs	Figure 21.6*1
$\overline{OE}$ setup time	t_{OES}	2	—	—	µs	
Data setup time	t_{DS}	2	—	—	µs	
Address hold time	t_{AH}	0	—	—	µs	
Data hold time	t_{DH}	2	—	—	µs	
Data output disable time	t_{DF}*2	—	—	130	ns	
Vpp setup time	t_{VPS}	2	—	—	µs	
$\overline{PGM}$ pulse width during initial programming	t_{PW}	0.19	0.20	0.21	ms	
$\overline{PGM}$ pulse width during over-programming	t_{OPW}*3	0.19	—	5.25	ms	
Vcc setup time	t_{VCS}	2	—	—	µs	
$\overline{CE}$ setup time	t_{CES}	2	—	—	µs	
Data output delay time	t_{OE}	0	—	150	ns	

Notes: *1 Input pulse level: 0.45 V to 2.4 V; input rise, fall times ≤ 20 ns; input timing reference levels: 0.8 V, 2.0 V; output timing reference levels: 0.8 V, 2.0 V.

*2 t_{DF} is defined as when the output becomes open state and referencing the output level is no longer possible.

*3 t_{OPW} is defined by the values noted in the flowchart (figure 21.6).



Note: * t_{OPW} is defined by the values noted in the flowchart (figure 19.6).

Figure 21.7 Write/Verify Timing

21.3.3 Cautions on Writing

1. Writes must always be done with the established voltage and timing. The write voltage (programming voltage) V_{PP} is 12.5 V (when the EPROM writer is set for the HN27C101 Hitachi specifications, V_{PP} becomes 12.5 V). Devices will sometimes be destroyed if a voltage higher than the rated one is applied. Pay particular attention to such phenomena as EPROM writer overshoot.
2. Always confirm that the indices of the EPROM writer socket, socket adapter, and device are in agreement before programming. Devices will sometimes be destroyed due to excessive current flow if these are not connected in the proper locations.
3. Do not touch the socket adapter or device during writing. Contact faults can sometimes cause devices to be improperly written.
4. Page programming mode writes are not possible. Always set to byte programming mode.

5. Terminate the writing if a write malfunction occurs in consecutive addresses. In such cases, check for problems in the EPROM writer and/or socket adapter. There are some cases where write/verify will not be possible if using an EPROM writer with a high impedance power supply system.
6. Use a EPROM writer that conforms to the socket adapter supported by this LSI.

21.3.4 Post-Write Reliability

High temperature biasing (or burn-in) of devices is recommended after writing in order to improve the data retention characteristics. High temperature biasing is a method of screening that eliminates parts with faulty initial data retention by on-chip PROM memory cells within a short period of time. Figure 21.8 shows the flow from the on-chip PROM programming including screening to the installation of the device on a board.

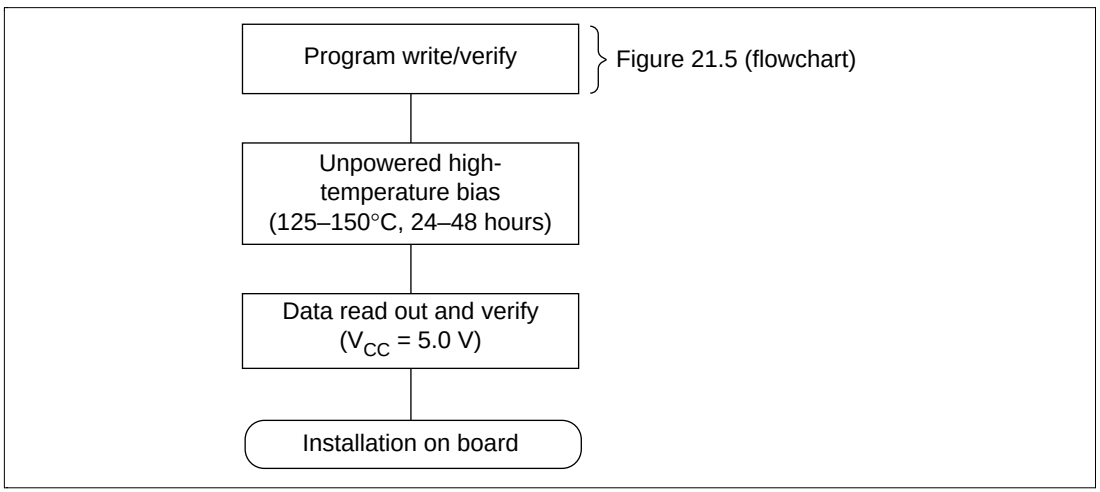


Figure 21.8 Screening Flow

If there are any abnormalities in program write/verify or program read-out verification after high temperature biasing, please contact a Renesas Technology technical representative.

Section 22 256kB Flash Memory (F-ZTAT)

22.1 Features

This LSI has 256 kbytes of on-chip flash memory. The features of the flash memory are summarized below.

- Four flash memory operating modes
  - Program mode
  - Erase mode
  - Program-verify mode
  - Erase-verify mode
- Programming/erase methods

  The flash memory is programmed 32 bytes at a time. Block erase (in single-block units) can be performed. Block erasing can be performed as required on 1 kbyte, 28 kbyte, and 32 kbyte blocks.
- Programming/erase times

  The flash memory programming time is 10 ms (typ.) for simultaneous 32-byte programming, equivalent to 300 µs (typ.) per byte, and the erase time is 100 ms (typ.) per block.
- Reprogramming capability

  The flash memory can be reprogrammed up to 100 times.
- On-board programming modes

  There are two modes in which flash memory can be programmed/erased/verified on-board:
  - Boot mode
  - User program mode
- Automatic bit rate adjustment

  With data transfer in boot mode, this LSI's bit rate can be automatically adjusted to match the transfer bit rate of the host.
- Flash memory emulation in RAM

  Flash memory programming can be emulated in real time by overlapping a part of RAM onto flash memory.
- Protect modes

  There are two protect modes, hardware and software, which allow protected status to be designated for flash memory program/erase/verify operations
- Programmer mode

  Flash memory can be programmed/erased in programmer mode, using a PROM programmer, as well as in on-board programming mode.

22.2 Overview

22.2.1 Block Diagram

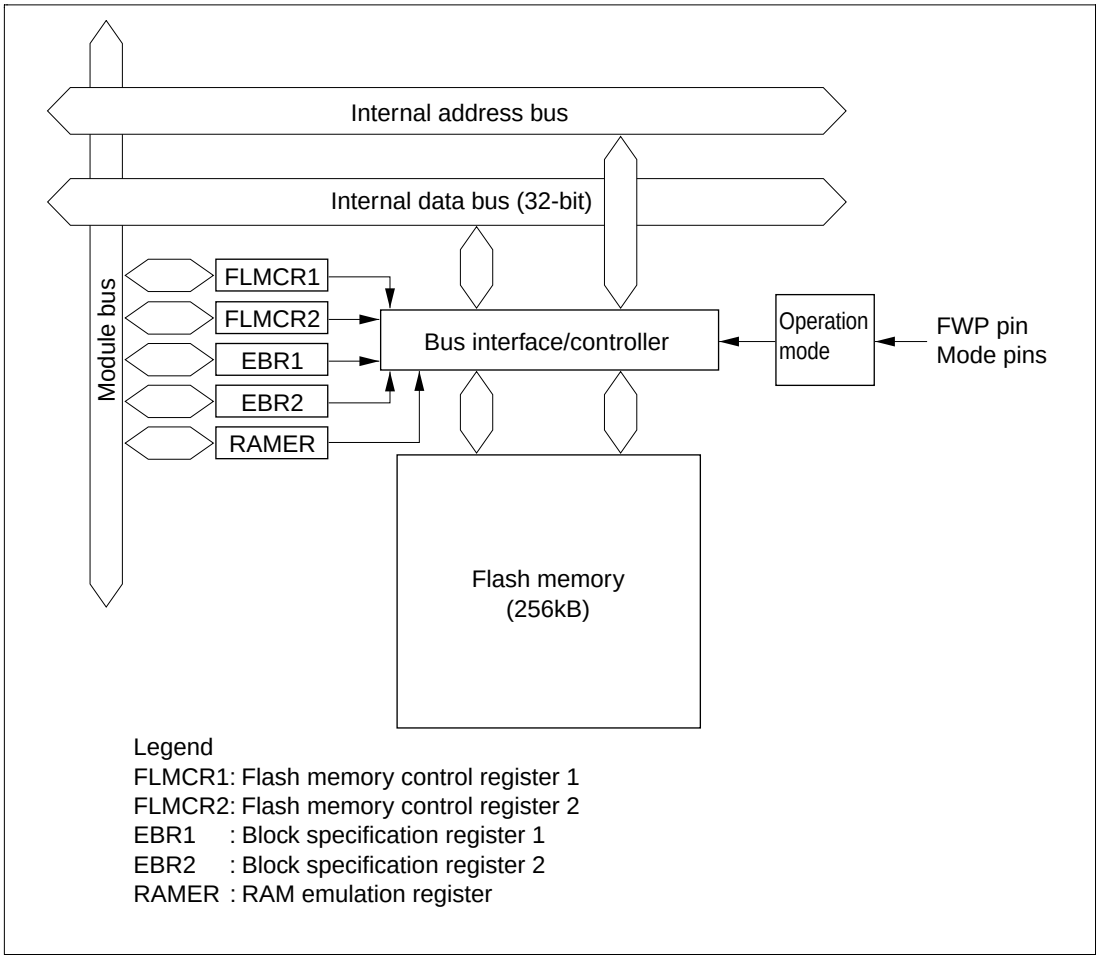


Figure 22.1 Flash Memory Block Diagram

22.2.2 Mode Transition Diagram

When the mode pins and the FWP pin are set in the reset state and a reset start is executed, the microcomputer enters one of the operating modes shown in figure 22.2. In user mode, flash memory can be read but not programmed or erased.

Flash memory can be programmed and erased in boot mode, user program mode, and programmer mode.

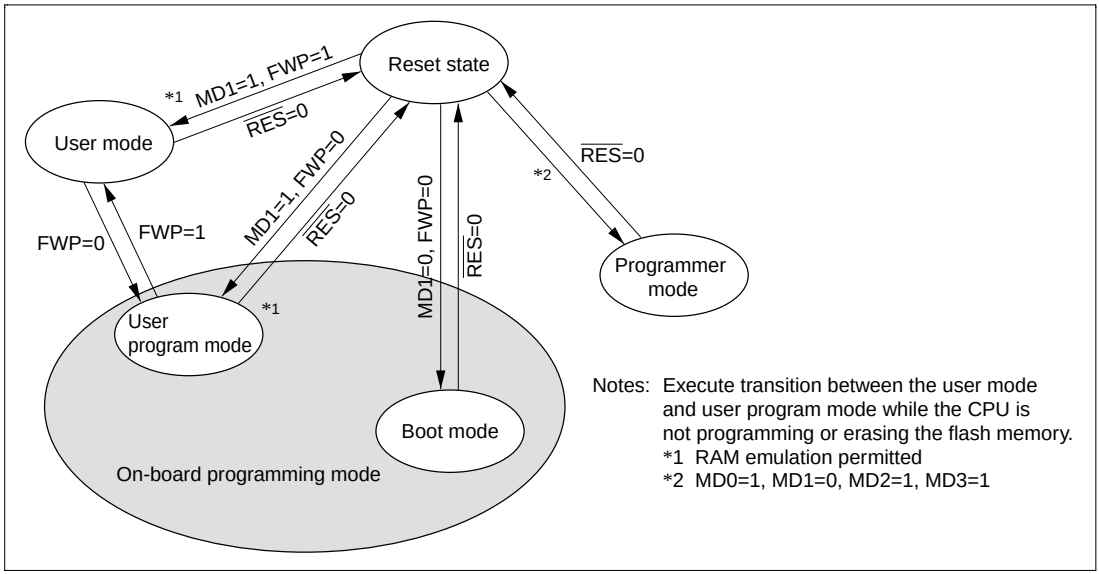
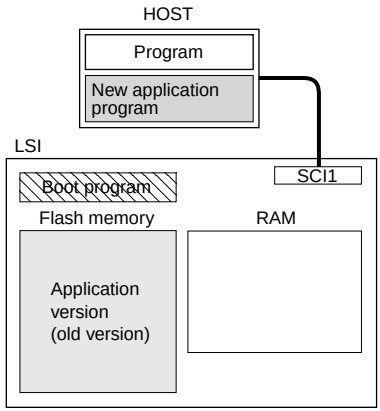


Figure 22.2 Flash Memory Mode Transitions

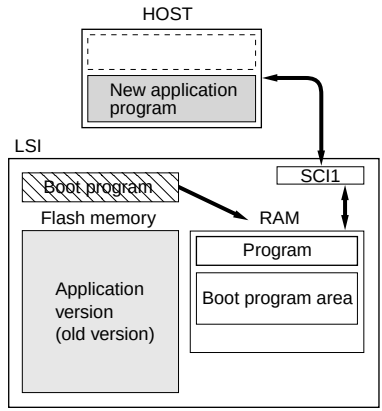
22.2.3 Onboard Program Mode

Boot mode

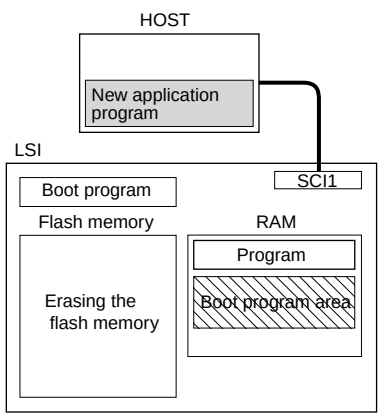
1. Initial state
   The old program version or data remains written in the flash memory. The user should prepare the programming control program and new application program beforehand in the host.



2. Programming control program transfer
   When boot mode is entered, the boot program in the LSI (originally incorporated in the chip) is started and the programming control program in the host is transferred to RAM via SCI communication. The boot program required for flash memory erasing is automatically transferred to the RAM boot program area.



3. Initializing the flash memory
   To initialize (to H'FF) the flash memory, execute the erase program located in the boot program area (within RAM). During the boot mode, the entire flash memory is erased, regardless of blocks.



4. Writing the new application program
   Execute the program transferred to RAM from the host and write the new application program located at the transfer destination to the flash memory.

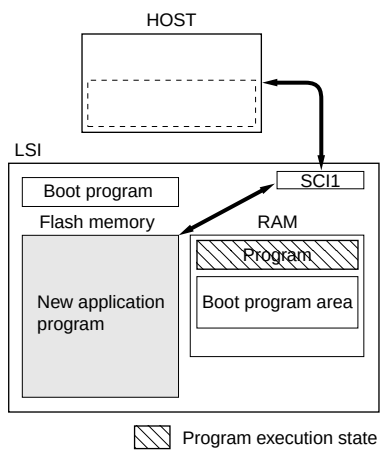
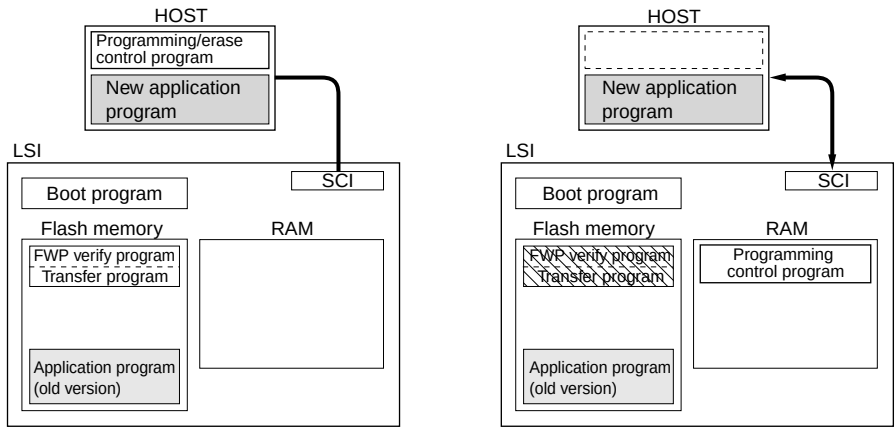


Figure 22.3 Boot Mode

1. Initial state
   The FWP assessment program that confirms that user program mode has been entered, and the program that will transfer the programming/erase control program from flash memory to on-chip RAM should be written into the flash memory by the user beforehand. The programming/erase control program should be prepared in the host or in the flash memory.

2. Programming/erase control program transfer
   When user program mode is entered, user software confirms this fact, executes transfer program in the flash memory, and transfers the programming/erase control program to RAM.



3. Initializing the flash memory
   Execute the Programming/erase program in RAM to initialize (to H'FF) the flash memory. Erase is executed in block units, but cannot be executed in byte units.

4. Writing new application program
   Next, the new application program in the host is written into the erased flash memory blocks. Do not write to unerased blocks.

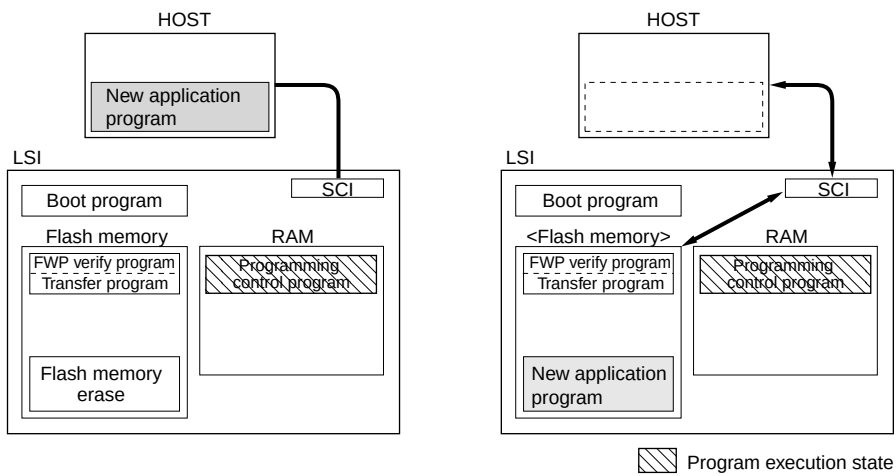


Figure 22.4 User Program Mode

22.2.4 Flash Memory Emulation in RAM

Emulation should be performed in user mode or user program mode. When the emulation block set in RAMER is accessed while the emulation function is being executed, data written in the overlap RAM is read.

- User Mode
- User Program Mode

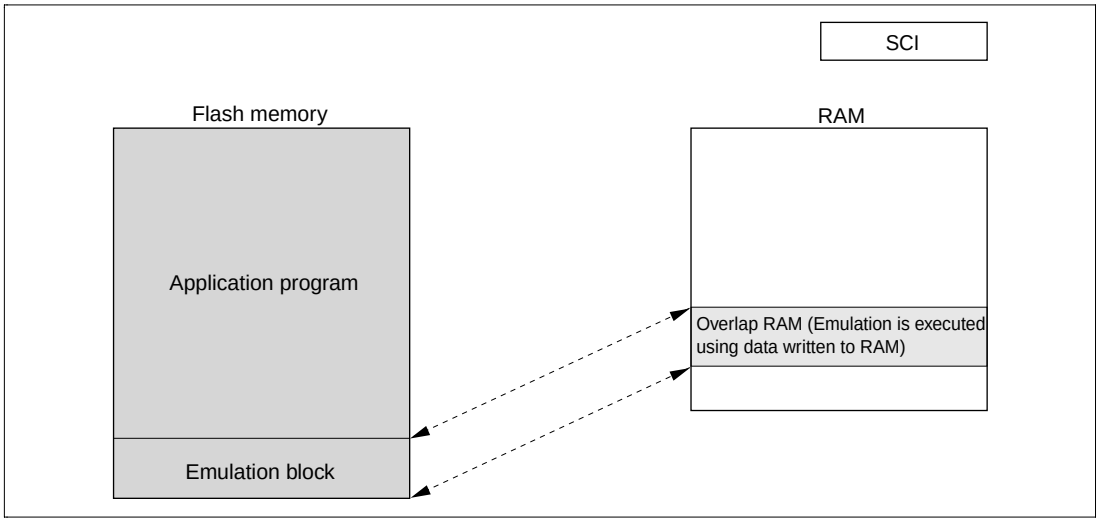


Figure 22.5 Emulation

When overlap RAM data is confirmed, the RAMS bit is cleared, RAM overlap is released, and writes should actually be performed to the flash memory.

When the programming control program is transferred to RAM, ensure that the transfer destination and the overlap RAM do not overlap, as this will cause data in the overlap RAM to be rewritten.

- User Program Mode

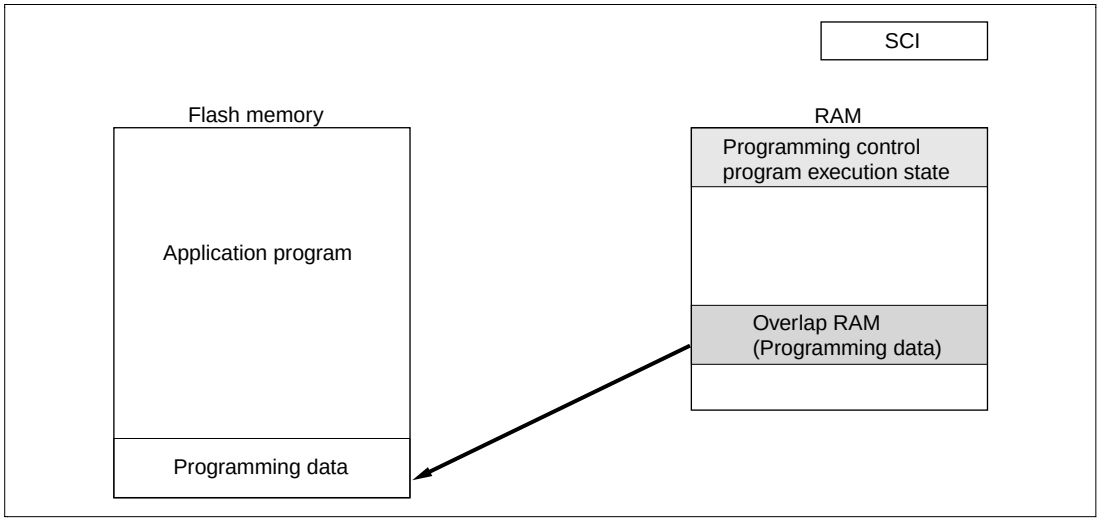


Figure 22.6 Programming to the Flash Memory

22.2.5 Differences between Boot Mode and User Program Mode

Table 22.1 Differences between Boot Mode and User Program Mode

	Boot Mode	User Program Mode
Total erase	Yes	Yes
Block erase	No	Yes
Programming control program*	(2)	(1) (2) (3)

(1) Erase/erase-verify

(2) Program/program-verify

(3) Emulation

Note: * To be prepared by the user according to the recommended algorithm.

22.2.6 Block Configuration

The flash memory is divided into seven 32 kbyte blocks, one 28 kbyte blocks, and four 1 kbyte blocks.

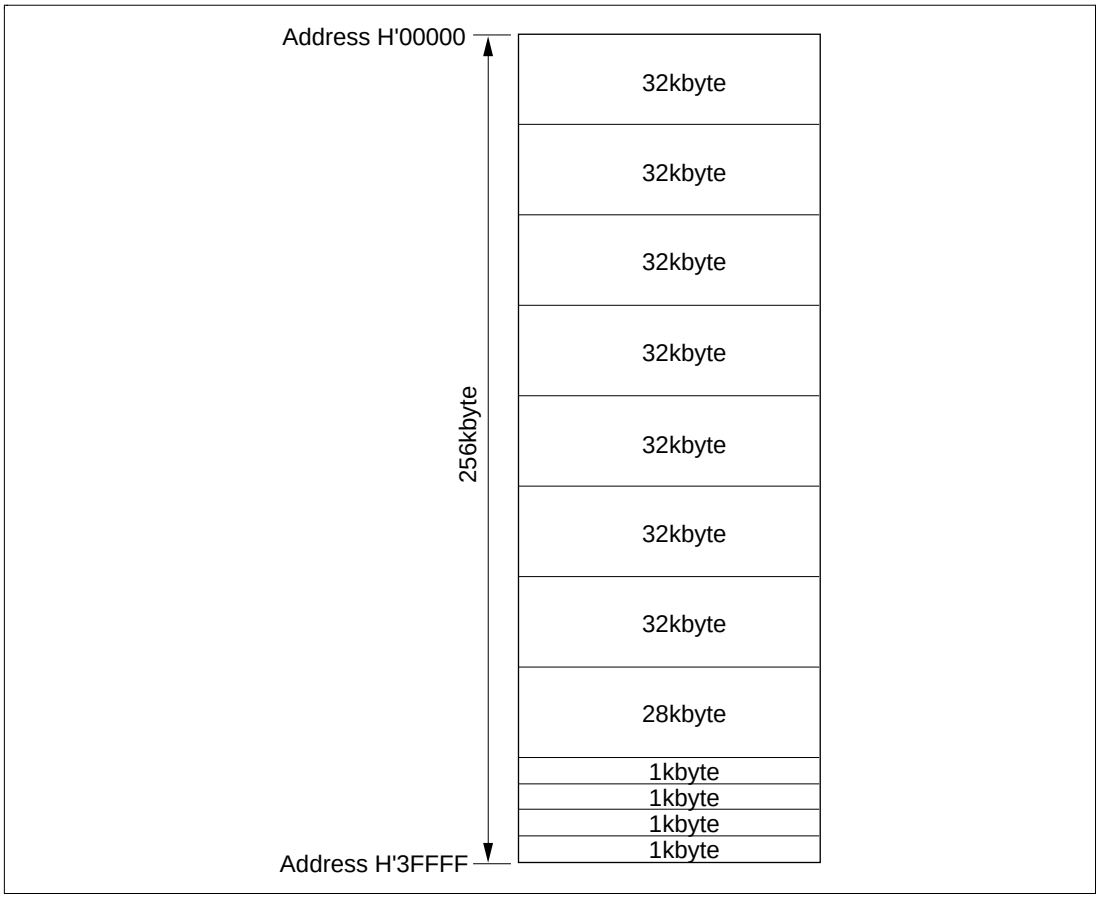


Figure 22.7 Block Configuration

22.3 Pin Configuration

The flash memory is controlled by the pins shown in table 22.2.

Table 22.2 Pin Configuration

Pin Name	Abbreviation	I/O	Function
Power-on reset	$\overline{\text{RES}}$	Input	Power-on reset
Flash write protect	FWP	Input	Flash program/erase protection by hardware
Mode 3	MD3	Input	Set operation mode of LSI
Mode 2	MD2	Input	Set operation mode of LSI
Mode 1	MD1	Input	Set operation mode of LSI
Mode 0	MD0	Input	Set operation mode of LSI
Transmit data	TxD1	Output	Serial send data output
Receive data	RxD1	Input	Serial receive data input

22.4 Register Configuration

Registers that control the flash memory when the on-chip flash memory is valid are shown in table 22.3.

Table 22.3 Register Configuration

Name	Abbre-viation	R/W	Initial Value	Address	Access Size
Flash memory control register 1	FLMCR1	R/W*1	H'00*2	H'FFFF8580	8
Flash memory control register 2	FLMCR2	R/W*1	H'00*3	H'FFFF8581	8
Erase block register 1	EBR1	R/W*1	H'00*3	H'FFFF8582	8
Erase block register 2	EBR2	R/W*1	H'00*3	H'FFFF8583	8
RAM emulation register	RAMER	R/W	H'0000	H'FFFF8628	8, 16, 32

Notes: 1. FLMCR1, FLMCR2, EBR1, and EBR2 are 8-bit registers, and RAMER is a 16-bit register.

2. Only byte accesses are valid for FLMCR1, FLMCR2, EBR1, and EBR2, the access requiring 3 cycles. Three cycles are required for a byte or word access to RAMER, and 6 cycles for a longword access.

3. When a longword write is performed on RAMER, 0 must always be written to the lower word (address H'FFFF8630). Operation is not guaranteed if any other value is written.

*1 In modes in which the on-chip flash memory is disabled, a read will return H'00, and writes are invalid. Writes are also disabled when the FWE bit is set to 1 in FLMCR1.

*2 When a low level is input to the FWP pin, the initial value is H'80.

*3 When a high level is input to the FWP pin, or if a low level is input and the SWE bit in FLMCR1 is not set, these registers are initialized to H'00.

22.5 Description of Registers

22.5.1 Flash Memory Control Register 1 (FLMCR1)

FLMCR1 is an 8-bit register used for flash memory operating mode control. Program-verify mode or erase-verify mode for addresses H'00000–H'1FFFF is entered by setting SWE to 1 when FWE = 1, then setting the EV1 or PV1 bit. Program mode for addresses H'00000–H'1FFFF is entered by setting SWE to 1 when FWE = 1, then setting the PSU1 bit, and finally setting the P1 bit. Erase mode for addresses H'00000–H'1FFFF is entered by setting SWE to 1 when FWE = 1, then setting the ESU1 bit, and finally setting the E1 bit. FLMCR1 is initialized in the standby mode or with power-on reset. Its initial value is H'80 when a low level is input to the FWP pin, and H'00 when a high level is input. When on-chip flash memory is disabled, a read will return H'80, and writes are invalid.

Writes to bits SWE, ESU1, PSU1, EV1, and PV1 are enabled only when FWE = 1 and SWE = 1; writes to the E1 bit only when FWE = 1, SWE = 1, and ESU1 = 1; and writes to the P1 bit only when FWE = 1, SWE = 1, and PSU1 = 1.

Bit:	7	6	5	4	3	2	1	0
	FWE	SWE	ESU1	PSU1	EV1	PV1	E1	P1
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- Bit 7—Flash Write Enable Bit (FWE): Displays the state of the FWP pin which sets hardware protection against flash memory programming/erasing.

Bit 7: FWE	Description
0	When high level is input to the FWP pin (hardware-protect state)
1	When low level is input to the FWP pin

- Bit 6—Software Write Enable Bit (SWE): Enables or disables the flash memory. This bit should be set when setting bits 5–0, FLMCR2 bits 5–0, EBR1 bits 3–0, and EBR2 bits 7–0.

Bit 6: SWE	Description	
0	Writes disabled	(Initial value)
1	Writes enabled [Setting condition] When FWE=1	

- Bit 5—Erase Setup Bit 1 (ESU1): Prepares for a transition to erase mode (applicable addresses: H'00000–H'1FFFF). Do not set the SWE, PSU1, EV1, PV1, E1, or P1 bit at the same time.

Bit 5: ESU1	Description
0	Erase setup release (Initial value)
1	Erase setup [Setting condition] When FWE=1 and SWE=1

- Bit 4—Program Setup Bit 1 (PSU1): Prepares for a transition to program mode (applicable addresses: H'00000–H'1FFFFF). Do not set the SWE, ESU1, EV1, PV1, E1, or P1 bit at the same time.

Bit 4: PSU1	Description
0	Program setup release (Initial value)
1	Program setup [Setting condition] When FWE=1 and SWE=1

- Bit 3—Erase-Verify 1 (EV1): Selects erase-verify mode transition or release (applicable addresses: H'00000–H'1FFFF). Do not set the SWE, ESU1, PSU1, PV1, E1, or P1 bit at the same time.

Bit 3: EV1	Description
0	Erase verify mode release (Initial value)
1	Transition to erase verify mode [Setting condition] When FWE=1 and SWE=1

- Bit 2—Program-Verify 1 (PV1): Selects program-verify mode transition or release (applicable addresses: H'00000–H'1FFFF). Do not set the SWE, ESU1, PSU1, EV1, E1, or P1 bit at the same time.

Bit 2: PV1	Description
0	Program verify mode release (Initial value)
1	Transition to program verify mode [Setting condition] When FWE=1 and SWE=1

- Bit 1—Erase 1 (E1): Selects erase mode transition or release (applicable addresses: H'00000–H'1FFFF). Do not set the SWE, ESU1, PSU1, EV1, PV1, or P1 bit at the same time.

Bit 1: E1	Description
0	Erase mode release (Initial value)
1	Transition to erase mode [Setting condition] When FWE=1, SWE=1, and ESU1=1

- Bit 0—Program 1 (P1): Selects program mode transition or release (applicable addresses: H'00000–H'1FFFF). Do not set the SWE, PSU1, ESU1, EV1, PV1, or E1 bit at the same time.

Bit 0: P1	Description
0	Program setup mode release (Initial value)
1	Program setup [Setting condition] When FWE=1, SWE=1, and PSU1=1

22.5.2 Flash Memory Control Register 2 (FLMCR2)

FLMCR2 is an 8-bit register used for flash memory operating mode control. Program-verify mode or erase-verify mode for addresses H'20000–H'3FFFF is entered by setting SWE (FLMCR1) to 1 when FWE (FLMCR1) = 1, then setting the EV2 or PV2 bit. Program mode for addresses H'20000–H'3FFFF is entered by setting SWE (FLMCR1) to 1 when FWE (FLMCR1) = 1, then setting the PSU2 bit, and finally setting the P2 bit. Erase mode for addresses H'20000–H'3FFFF is entered by setting SWE (FLMCR1) to 1 when FWE (FLMCR1) = 1, then setting the ESU2 bit, and finally setting the E2 bit. FLMCR2 is initialized to H'00 by a power-on reset, in standby mode, when a high level is input to the FWP pin, and when a low level is input to the FWP pin and the SWE bit in FLMCR1 is not set (the exception is the FLER bit, which is initialized only by a power-on reset). When on-chip flash memory is disabled, a read will return H'00, and writes are invalid.

Writes to bits ESU2, PSU2, EV2, and PV2 in FLMCR2 are enabled only when FWE (FLMCR1) = 1 and SWE (FLMCR1) = 1; writes to the E2 bit only when FWE (FLMCR1) = 1, SWE (FLMCR1) = 1, and ESU2 = 1; and writes to the P2 bit only when FWE (FLMCR1) = 1, SWE (FLMCR1) = 1, and PSU2 = 1.

Bit:	7	6	5	4	3	2	1	0
	FLER	—	ESU2	PSU2	EV2	PV2	E2	P2
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W

- Bit 7—Flash Memory Error (FLER): Indicates that an error has occurred during an operation on flash memory (programming or erasing). When FLER is set to 1, flash memory goes to the error-protection state.

Bit 7: FLER	Description
0	Flash memory is operating normally. (Initial value) Flash memory program/erase protect (error protect) disabled
1	Indicates error during flash memory program/erase. Flash memory program/erase protect (error protect) enabled [Setting condition] See section 22.8.3, Error protection

- Bit 6—Reserved bit: This bit is always read as 0.

- Bit 5—Erase Setup Bit 2 (ESU2): Prepares for a transition to erase mode (applicable addresses: H'20000–H'3FFFF). Do not set the PSU2, EV2, PV2, E2, or P2 bit at the same time.

Bit 5: ESU2	Description
0	Erase setup release (Initial value)
1	Erase setup [Setting condition] When FWE=1 and SWE=1

- Bit 4—Program Setup Bit 2 (PSU2): Prepares for a transition to program mode (applicable addresses: H'20000–H'3FFFF). Do not set the ESU2, EV2, PV2, E2, or P2 bit at the same time.

Bit 4: PSU2	Description
0	Program setup release (Initial value)
1	Program setup [Setting condition] When FWE=1 and SWE=1

- Bit 3—Erase-Verify 2 (EV2): Selects erase-verify mode transition or release (applicable addresses: H'20000–H'3FFFF). Do not set the ESU2, PSU2, PV2, E2, or P2 bit at the same time.

Bit 3: EV2	Description
0	Erase verify mode release (Initial value)
1	Transition to the erase verify mode [Setting condition] When FWE=1 and SWE=1

- Bit 2—Program-Verify 2 (PV2): Selects program-verify mode transition or release (applicable addresses: H'20000–H'3FFFF). Do not set the ESU2, PSU2, EV2, E2, or P2 bit at the same time.

Bit 2: PV2	Description
0	Program verify mode release (Initial value)
1	Transition to the program verify mode [Setting condition] When FWE=1, and SWE=1

- Bit 1—Erase 2 (E2): Selects erase mode transition or release (applicable addresses: H'20000–H'3FFFF). Do not set the ESU2, PSU2, EV2, PV2, or P2 bit at the same time.

Bit 1: E2	Description
0	Erase mode release (Initial value)
1	Transition to the erase mode [Setting condition] When FWE=1, SWE=1, and ESU2=1

- Bit 0—Program 2 (P2): Selects program mode transition or release (applicable addresses: H'20000–H'3FFFF). Do not set the ESU2, PSU2, EV2, PV2, or E2 bit at the same time.

Bit 0: P2	Description
0	Program mode release(Initial value)
1	Transition to the program mode [Setting condition] When FWE=1, SWE=1, and PSU2=1

22.5.3 Erase Block Register 1 (EBR1)

EBR1 is an 8-bit register that specifies the flash memory erase area block by block. EBR1 is initialized to H'00 by a power-on reset and standby mode, when a high level is input to the FWP pin, and when a low level is input to the FWP pin and the SWE bit in FLMCR1 is not set. When a bit in EBR1 is set to 1, the corresponding block can be erased. Other blocks are erase-protected. Only one of the bits of EBR1 and EBR2 combined can be set. Do not set more than one bit. If more than one bit is set, writes to bits ESU1, ESU2, E1, and E2 will be invalid. When on-chip flash memory is disabled, a read will return H'00, and writes are invalid.

The flash memory block configuration is shown in table 22.4.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	EB3	EB2	EB1	EB0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W

22.5.4 Erase Block Register 2 (EBR2)

EBR2 is an 8-bit register that specifies the flash memory erase area block by block. EBR2 is initialized to H'00 by a power-on reset and standby mode, when a high level is input to the FWP pin, and when a low level is input to the FWP pin and the SWE bit in FLMCR1 is not set. When a bit in EBR2 is set to 1, the corresponding block can be erased. Other blocks are erase-protected. When on-chip flash memory is disabled, a read will return H'00, and writes are invalid.

The flash memory block configuration is shown in table 22.4.

Bit:	7	6	5	4	3	2	1	0
	EB11	EB10	EB9	EB8	EB7	EB6	EB5	EB4
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 22.4 Flash Memory Erase Blocks

Block (size)	Addresses
EB0 (32kB)	H'000000–H'007FFF
EB1 (32kB)	H'008000–H'00FFFF
EB2 (32kB)	H'010000–H'017FFF
EB3 (32kB)	H'018000–H'01FFFF
EB4 (32kB)	H'020000–H'027FFF
EB5 (32kB)	H'028000–H'02FFFF
EB6 (32kB)	H'030000–H'037FFF
EB7 (28kB)	H'038000–H'03EFFF
EB8 (1kB)	H'03F000–H'03F3FF
EB9 (1kB)	H'03F400–H'03F7FF
EB10 (1kB)	H'03F800–H'03FBFF
EB11 (1kB)	H'03FC00–H'03FFFF

22.5.5 RAM Emulation Register (RAMER)

RAMER specifies the area of flash memory to be overlapped with part of RAM when emulating real-time flash memory programming. RAMER is initialized to H'0000 by a power-on reset. It is not initialized in software standby mode. RAMER settings should be made in user mode or user program mode. (For details, see the description of the BSC.)

Flash memory area divisions are shown in table 22.5. To ensure correct operation of the emulation function, the ROM for which RAM emulation is performed should not be accessed immediately after this register has been modified. Normal execution of an access immediately after register modification is not guaranteed.

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	RAMS	RAM1	RAM0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

- Bits 15–3—Reserved bits: These bits are always read as 0.

- Bit 2—RAM Select (RAMS): Specifies selection or non-selection of flash memory emulation in RAM. When RAMS = 1, all flash memory block are program/erase-protected. This bit is ignored when the on-chip ROM is disabled.

Bit 2: RAMS	Description
0	Emulation not selected Program/erase protect of all flash memory blocks is disabled (Initial value)
1	Emulation selected Program/erase protect of all flash memory blocks is enabled

- Bits 1 and 0—Flash Memory Area Selection (RAM1, RAM0): These bits are used together with bit 2 to select the flash memory area to be overlapped with RAM. (See table 22.5.)

Table 22.5 Separation of the Flash Memory Area

Addresses	Block Name	RAMS	RAM1	RAM0
H'FFF800–H'FFFBFF	RAM area 1kB	0	*	*
H'03F000–H'03F3FF	EB8 (1kB)	1	0	0
H'03F400–H'03F7FF	EB9 (1kB)	1	0	1
H'03F800–H'03FBFF	EB10(1kB)	1	1	0
H'03FC00–H'03FFFF	EB11(1kB)	1	1	1

22.6 On-Board Programming Mode

When pins are set to on-board programming mode and a power-on reset is executed, a transition is made to the on-board programming state in which program/erase/verify operations can be performed on the on-chip flash memory. There are two on-board programming modes: boot mode and user program mode. The pin settings for transition to each of these modes are shown in table 22.6. For a diagram of the transitions to the various flash memory modes, see figure 22.2.

Table 22.6 Setting On-Board Programming Modes

Mode		PLL Multiple	FWP	MD3	MD2	MD1	MD0
Boot mode	Expanded Mode	×1	0	0	0	0	0
	Single-chip Mode			0	0	0	1
	Expanded Mode	×2		0	1	0	0
	Single-chip Mode			0	1	0	1
	Expanded Mode	×4		1	0	0	0
	Single-chip Mode			1	0	0	1
User program mode	Expanded Mode	×1	0	0	0	1	0
	Single-chip Mode			0	0	1	1
	Expanded Mode	×2		0	1	1	0
	Single-chip Mode			0	1	1	1
	Expanded Mode	×4		1	0	1	0
	Single-chip Mode			1	0	1	1

22.6.1 Boot Mode

When boot mode is used, the flash memory programming control program must be prepared in the host beforehand. The SCI to be used is set to channel asynchronous mode.

When a reset start is executed after the LSI pins have been set to boot mode in the power-on reset state, the boot program built into the LSI is started and the programming control program prepared in the host is serially transmitted to the LSI via SCI channel 1. In the LSI, the programming control program received via SCI channel 1 is written into the programming control program area in on-chip RAM. After the transfer is completed, control branches to the start address of the programming control program area and the programming control program execution state is entered (flash memory programming is performed).

The transferred programming control program must therefore include coding that follows the programming algorithm given later.

The system configuration in boot mode is shown in figure 22.8, and the boot mode execution procedure in figure 22.9.

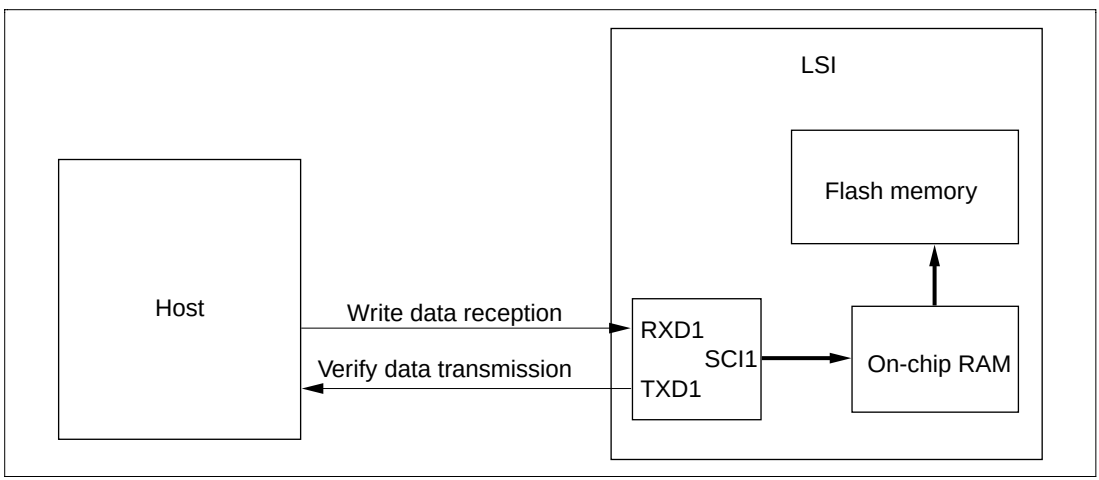
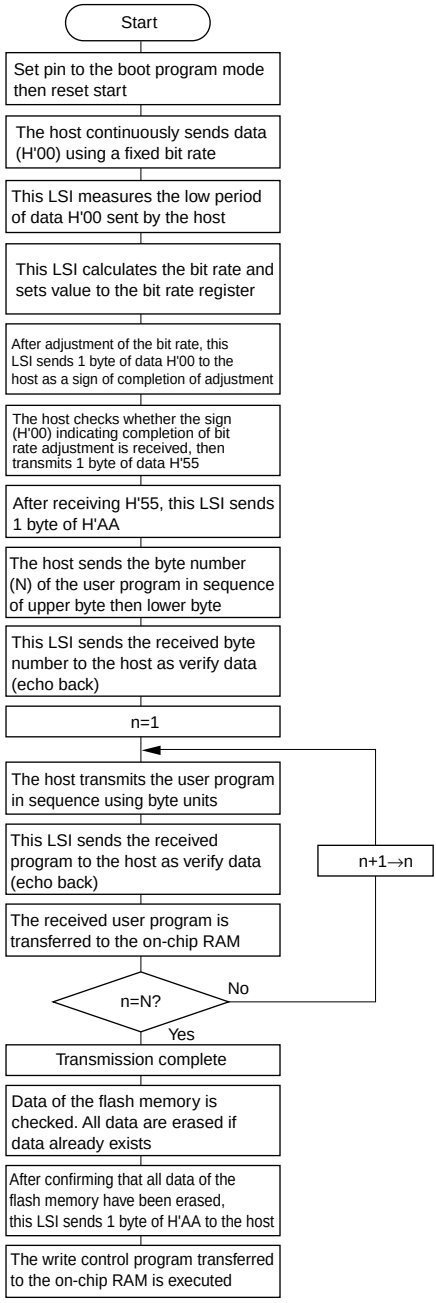


Figure 22.8 System Configuration in Boot Mode



Note: If a memory cell does not operate normally and cannot be erased, one H'FF byte is transmitted as an erase error, and the erase operation and subsequent operations are halted.

Figure 22.9 Boot Mode Execution Procedure

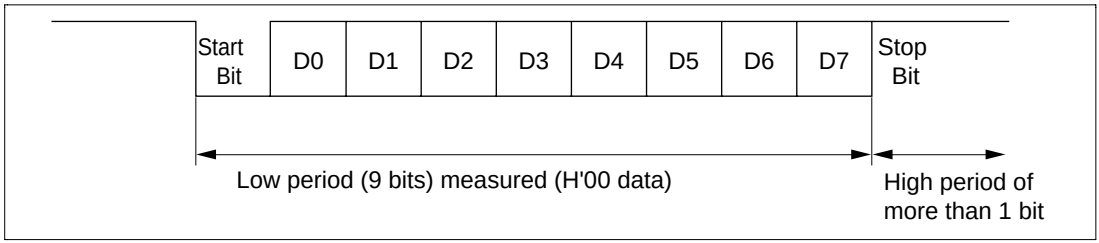


Figure 22.10 Automatic SCI Bit Rate Adjustment

When boot mode is initiated, the LSI measures the low period of the asynchronous SCI communication data (H'00) transmitted continuously from the host. The SCI transmit/receive format should be set as follows: 8-bit data, 1 stop bit, no parity. The LSI calculates the bit rate of the transmission from the host from the measured low period, and transmits one H'00 byte to the host to indicate the end of bit rate adjustment. The host should confirm that this adjustment end indication (H'00) has been received normally, and transmit one H'55 byte to the LSI. If reception cannot be performed normally, initiate boot mode again (reset), and repeat the above operations. Depending on the host's transmission bit rate and the LSI's system clock frequency, there will be a discrepancy between the bit rates of the host and the LSI. To ensure correct SCI operation, the host's transfer bit rate should be set to 9,600 or 4,800 bps.

Table 22.7 shows host transfer bit rates and system clock frequencies for which automatic adjustment of the LSI bit rate is possible. The boot program should be executed within this system clock range.

Table 22.7 System Clock Frequencies for which Automatic Adjustment of LSI Bit Rate is Possible

Host Bit Rate	System Clock Frequency for which Automatic Adjustment of LSI Bit Rate is Possible
9,600 bps	8 to 28.7 MHz
4,800 bps	4 to 20 MHz

On-Chip RAM Area Divisions in Boot Mode: In boot mode, the RAM area is divided into an area used by the boot program and an area to which the programming control program is transferred via the SCI, as shown in figure 22.11. The boot program area cannot be used until the execution state in boot mode switches to the programming control program transferred from the host.

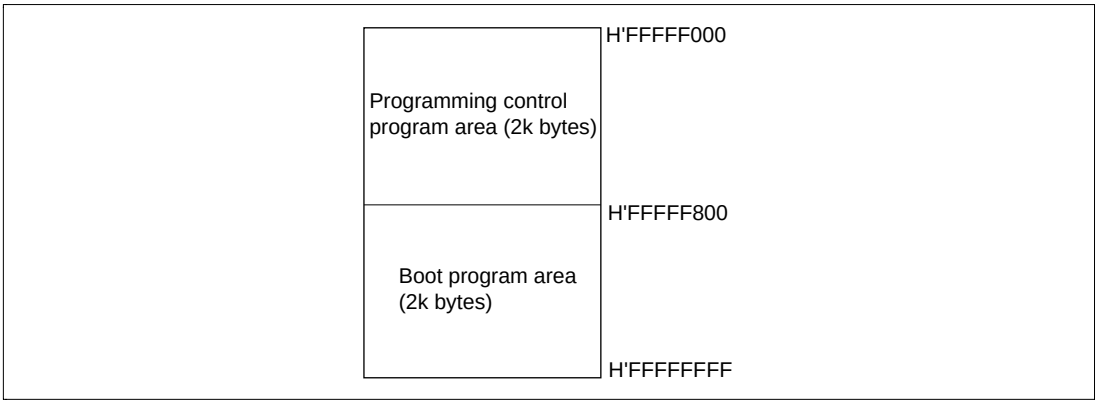


Figure 22.11 RAM Areas in Boot Mode

Note: The boot program area cannot be used until a transition is made to the execution state for the programming control program transferred to RAM. Note also that the boot program remains in this area of the on-chip RAM even after control branches to the programming control program.

22.6.2 User Program Mode

After setting FWP, the user should branch to, and execute, the previously prepared programming/erase control program.

As the flash memory itself cannot be read while flash memory programming/erasing is being executed, the control program that performs programming and erasing should be run in on-chip RAM or external memory.

Use the following procedure (figure 22.12) to execute the programming control program that writes to flash memory (when transferred to RAM).

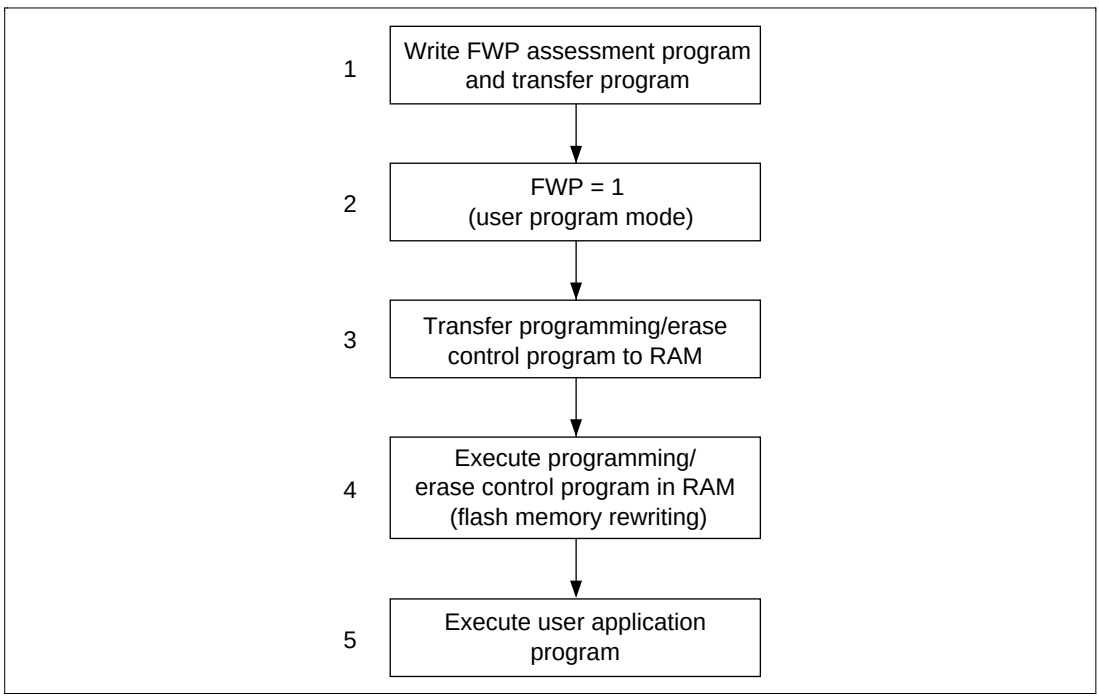


Figure 22.12 User Program Mode Execution Procedure

Notes: 1. When programming and erasing, start the watchdog timer so that measures can be taken to prevent program runaway, etc. Memory cells may not operate normally if overprogrammed or overerased due to program runaway.

2. If an address at which a flash memory register resides is read in the mask ROM or ZTAT version, the value will be undefined. When a flash memory version program is used in the mask ROM or ZTAT version, the state of the FWP pin cannot be determined. A modification must therefore be made to prevent operation of the flash memory rewrite program.

22.7 Programming/Erasing Flash Memory

A software method, using the CPU, is employed to program and erase flash memory in the on-board programming modes. There are four flash memory operating modes: program mode, erase mode, program-verify mode, and erase-verify mode. Transitions to these modes are made by setting the PSU1, ESU1, P1, E1, PV1, and EV1 bits in FLMCR1 for addresses H'00000–H'1FFFF, or the PSU2, ESU2, P2, E2, PV2, and EV2 bits in FLMCR2 for addresses H'20000–H'3FFFF.

The flash memory cannot be read while being programmed or erased. Therefore, the program (programming control program) that controls flash memory programming/erasing should be located and executed in on-chip RAM or external memory.

Notes: 1. Operation is not guaranteed if setting/resetting of the SWE, ESU1, PSU1, EV1, PV1, E1, and P1 bits in FLMCR1, or the ESU2, PSU2, EV2, PV2, E2, and P2 bits in FLMCR2, is executed by a program in flash memory.
2. When programming or erasing, set FWP to low level (programming/erasing will not be executed if FWP is set to high level).
3. Programming should be performed in the erased state. Do not perform additional programming on previously programmed addresses.
4. Do not program addresses H'00000–H'1FFFF and H'20000–H'3FFFF simultaneously. Operation is not guaranteed if this is done.

22.7.1 Program Mode (n = 1 for Addresses H'0000–H'1FFFF, n = 2 for Addresses H'20000–H'3FFFF)

When writing data or programs to flash memory, the program/program-verify flowchart shown in figure 22.13 should be followed. Performing program operations according to this flowchart will enable data or programs to be written to flash memory without subjecting the device to voltage stress or sacrificing program data reliability. Programming should be carried out 32 bytes at a time.

Following the elapse of 10 µs or more after the SWE bit is set to 1 in flash memory control register 1 (FLMCR1), 32-byte program data is stored in the program data area and reprogram data area, and the 32-byte data in the program data area in RAM is written consecutively to the program address (the lower 8 bits of the first address written to must be H'00, H'20, H'40, H'60, H'80, H'A0, H'C0, or H'E0). Thirty-two consecutive byte data transfers are performed. The program address and program data are latched in the flash memory. A 32-byte data transfer must be performed even if writing fewer than 32 bytes; in this case, H'FF data must be written to the extra addresses.

Next, the watchdog timer is set to prevent overprogramming in the event of program runaway, etc. Set a minimum value of 300 µs or more as the WDT overflow period. After this, preparation for program mode (program setup) is carried out by setting the PSUn bit in FLMCRn, and after the elapse of 50 µs or more, the operating mode is switched to program mode by setting the Pn bit in

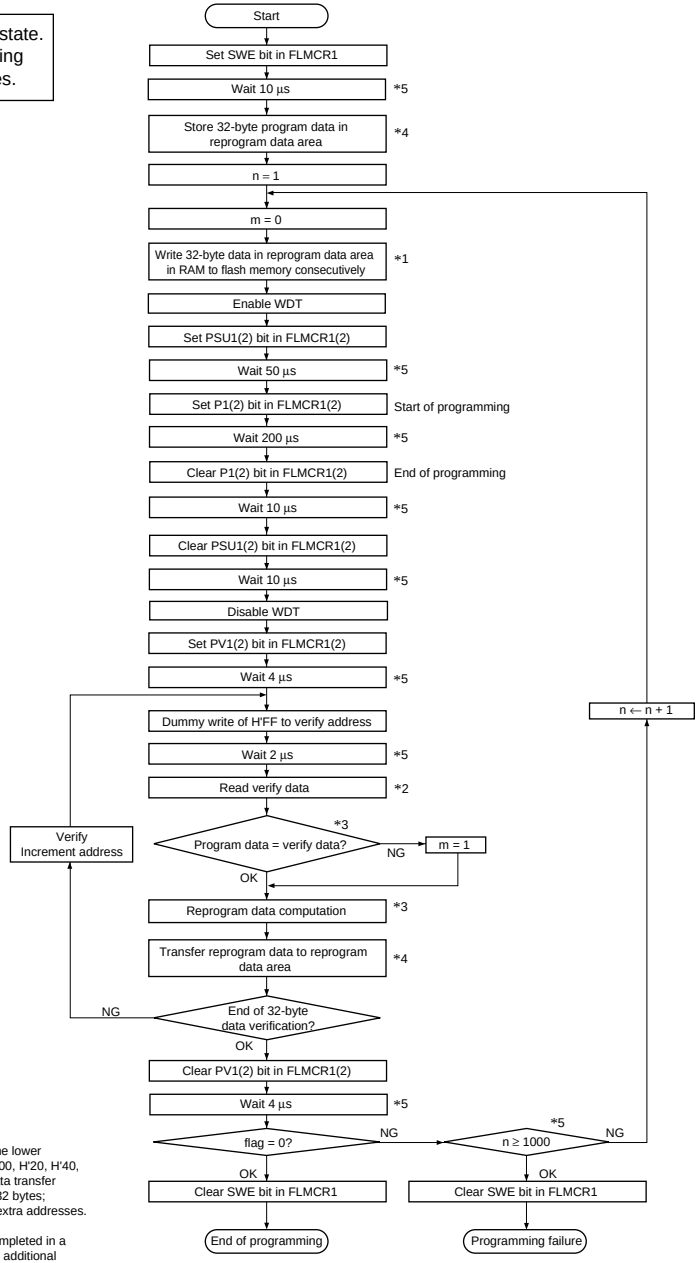
FLMCRn. The time during which the Pn bit is set is the flash memory programming time. Set 200 µs as the time for one programming operation.

22.7.2 Program-Verify Mode (n = 1 for Addresses H'0000–H'1FFFF, n = 2 for Addresses H'20000–H'3FFFF)

In program-verify mode, the data written in program mode is read to check whether it has been correctly written in the flash memory.

After the elapse of a given programming time, the programming mode is exited (the Pn bit in FLMCRn is released, then the PSUn bit is released at least 10 µs later). The watchdog timer is released after the elapse of 10 µs or more, and the operating mode is switched to program-verify mode by setting the PVn bit in FLMCRn. Before reading in program-verify mode, a dummy write of H'FF data should be made to the addresses to be read. The dummy write should be executed after the elapse of 4 µs or more. When the flash memory is read in this state (verify data is read in 32-bit units), the data at the latched address is read. Wait at least 2 µs after the dummy write before performing this read operation. Next, the written data is compared with the verify data, and reprogram data is computed (see figure 22.13) and transferred to the reprogram data area. After 32 bytes of data have been verified, exit program-verify mode, wait for at least 4 µs, then release the SWE bit in FLMCR1. If reprogramming is necessary, set program mode again, and repeat the program/program-verify sequence as before. However, ensure that the program/program-verify sequence is not repeated more than 1,000 times on the same bits.

Perform programming in the erased state.
Do not perform additional programming
on previously programmed addresses.



Notes: *1 Data transfer is performed by byte transfer. The lower 8 bits of the first address written to must be H'00, H'20, H'40, H'60, H'80, H'A0, H'C0, or H'E0. A 32-byte data transfer must be performed even if writing fewer than 32 bytes; in this case, H'FF data must be written to the extra addresses.
*2 Verify data is read in 32-bit (longword) units.
*3 Even bits for which programming has been completed in a 32-byte programming loop will be subjected to additional programming if they fail the subsequent verify operation.
*4 A 32-byte area for storing program data and a 32-byte area for storing reprogram data are required in RAM. The contents of the latter are rewritten according to the progress of the programming operation.
*5 Make sure to set the wait times and repetitions as specified. Programming may not complete correctly if values other than the specified ones are used.

Note: The memory erased state is 1. Programming is performed on 0 data.

Write data (D)	Verify data (V)	Rewrite data (X)	Comment
0	0	1	Programming completed
0	1	0	Programming incomplete; reprogram
1	0	1	
1	1	1	Still in erased state; no action

Figure 22.13 Program/Program Verify Flow

- Sample 32-byte programming program

  The wait time set values (number of loops) are for the case where f = 28.7 MHz. For other frequencies, the set value is given by the following expression:

  Wait time (μs) × f (MHz) ÷ 4

Registers Used

R4 (input): Program data storage address
R5 (input): Programming destination address
R7 (output): OK (normal) or NG (error)
R0-3, 8-13: Work registers

```
FLMCR1          .EQU    H'80
FLMCR2          .EQU    H'81
OK              .EQU    H'0
NG              .EQU    H'1
Wait10u         .EQU    72
Wait50u         .EQU    359
Wait4u          .EQU    29
Wait2u          .EQU    14
Wait200u        .EQU    1435
WDT_TCSR        .EQU    H'FFFF8610
WDT_573u        .EQU    H'A579
SWESET          .EQU    B'01000000
PSU1SET         .EQU    B'00010000
P1SET           .EQU    B'00000001
P1CLEAR         .EQU    B'11111110
PSU1CLEAR       .EQU    B'11101111
PVSET           .EQU    B'00000100
PVCLEAR         .EQU    B'11111011
SWECLEAR        .EQU    B'10111111
MAXVerify       .EQU    1000
;
FlashProgram    .EQU    $
        MOV     #H'01,R2            ; R2 work register (1)
        MOV.L   #PdataBuff,R0       ; Save program data to work area
        MOV     R4,R12
        MOV     #8,R13
COPY_LOOP       .EQU    $
```

```
        MOV.L   @R12+,R1
        MOV.L   R1,@R0
        ADD.L   #4,R0
        ADD.L   #-1,R13
        CMP/PL  R13
        BT      COPY_LOOP
        MOV.L   #H'FFFF8500,R0          ; Initialize GBR
        LDC     R0,GBR
;
        MOV.L   #Wait10u,R3
        MOV.L   #FLMCR1,R0              ; Initialize R0 to FLCMR1 address
        OR.B    #SWESET,@(R0,GBR)       ; Set SWE
Wait_1  SUBC    R2,R3                   ; Wait 10 µs
        BF      Wait_1
;
        MOV.L   #H'20000,R9
        CMP/GT  R5,R9
        BT      Program_Start
        MOV.L   #FLMCR2,R0
Program_Start   .EQU    $
        MOV.L   #0,R9                   ; Initialize n (R9) to 0
;
Program_loop    .EQU    $
        MOV.L   #0,R10                  ; Initialize m (R10) to 0
        MOV.L   #32,R3                  ; Write 32-byte data consecutively
        MOV.L   #PdataBuff,R12
        MOV.L   R5,R13
Write_Loop      .EQU    $
        MOV.B   @R12+,R1
        MOV.B   R1,@R13
        ADD.L   #1,R13
        ADD.L   #-1,R3
        CMP/PL  R3
        BT      Write_Loop
;
        MOV.L   #WDT_TCSR,R1            ; Enable WDT
        MOV.W   #WDT_573u,R3            ; 573.4 µs cycle
```

```
        MOV.W     R3,@R1
;
        MOV.L     #Wait50u,R3
        OR.B      #PSU1SET,@(R0,GBR)      ; Set PSU
Wait_2  SUBC      R2,R3                   ; Wait 50 µs
        BF
;
        MOV.L     #Wait200u,R3
        OR.B      #P1SET,@(R0,GBR)        ; Set P
Wait_3  SUBC      R2,R3                   ; Wait 200 µs
        BF        Wait_3
;
        MOV.L     #Wait10u,R3
        AND.B     #P1CLEAR,@(R0,GBR)      ; Clear P
Wait_4  SUBC      R2,R3                   ; Wait 10 µs
        BF        Wait_4
;
        MOV.L     #Wait10u,R3
        AND.B     #PSU1CLEAR,@(R0,GBR)    ; Clear PSU
Wait_5  SUBC      R2,R3                   ; Wait 10 µs
        BF        Wait_5
;
        MOV.L     #WDT_TCSR,R1            ; Disable WDT
        MOV.W     #H'A55F,R3
        MOV.W     R3,@R1
;
        MOV.L     #Wait4u,R3
        OR.B      #PVSET,@(R0,GBR)        ; Set PV
Wait_6  SUBC      R2,R3                   ; Wait 4 µs
        BF        Wait_6
;
        MOV.L     PdataBuff,R3
        MOV.L     R4,R1
        MOV.L     R5,R12
        MOV.L     #8,R13
        MOV.L     #H'FFFFFFFF,R11
;
```

```
VerifyLoop      .EQU    $
        MOV.L   R11,@R12            ; Write H'FF to verify address
        MOV.L   R11,@R3             ; Reprogram data RAM (PdataBuff) initialization
        MOV.L   #Wait2u,R7
Wait_7  SUBC    R2,R7               ; Wait 2 µs
        BF      Wait_7
;
        MOV.L   @R12+,R7
        MOV.L   @R1+,R8
        CMP/EQ  R7,R8               ; Verify
        BT      Verify_OK
        MOV.L   #1,R10              ; Verify NG, m <- 1
        XOR     R8,R7               ; Program data computation
        NOT     R7,R7
        OR      R7,R8
        MOV.L   R8,@R3              ; Store in reprogram data RAM (PdataBuff)
Verify_OK       .EQU    $
        ADD.L   #4,R3
        ADD.L   #-1,R13
        CMP/PL  R13
        BT      VerifyLoop
;
        MOV.L   #Wait4u,R7
        AND.B   #PVCLEAR,@(R0,GBR)  ; Clear PV
Wait_8  SUBC    R2,R7               ; Wait 4 µs
        BF      Wait_8
;
        CMP/PL  R10 ; if m=0 then GOTO Program_OK
        BF      Program_OK
        ADD     #1,R9
        MOV.L   #NG,R7              ; R7 <- NG (return value)
        MOV.L   #MAXVerify,R12      ; if n>=MAXVerify then Program NG
        CMP/EQ  R9,R12
        BT      Program_end
        BRA     Program_loop
        NOP
Program_OK      .EQU    $
```

```
        MOV.L     #OK,R7                ; R7 <- OK (return value)
Program_end       .EQU      $
        MOV.B     #H'00,R0
        MOV.B     R0,@(FLMCR1,GBR)      ; Clear SWE
;
        RTS
        NOP
;
        .ALIGN    4
PdataBuff         .RES.B  32
```

22.7.3 Erase Mode (n = 1 for Addresses H'0000–H'1FFFF, n = 2 for Addresses H'20000–H'3FFFF)

When erasing flash memory, the erase/erase-verify flowchart shown in figure 22.14 should be followed.

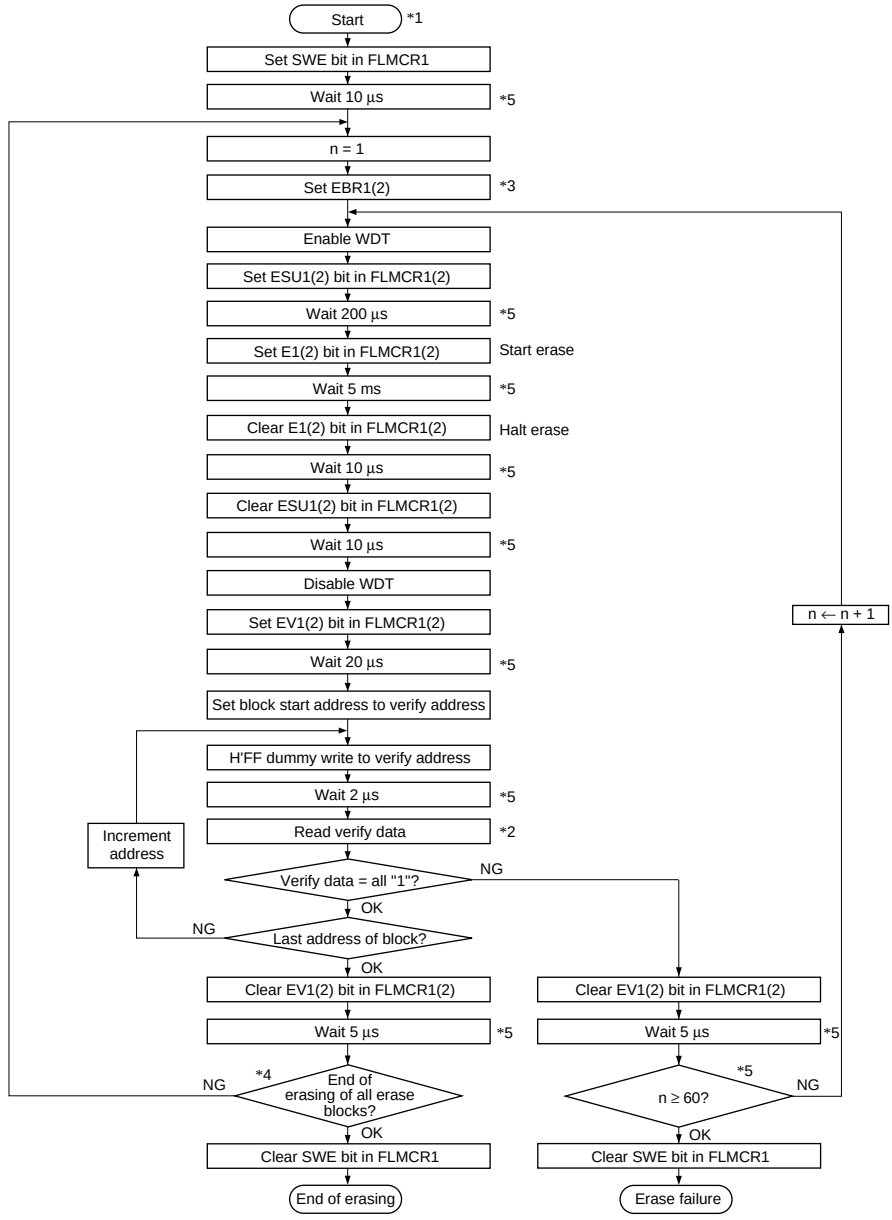
To perform data or program erasure, set the flash memory area to be erased in erase block register n (EBRn) at least 10 µs after setting the SWE bit to 1 in flash memory control register 1 (FLMCR1). Next, the watchdog timer is set to prevent overerasing in the event of program runaway, etc. Set 5.3 µs as the WDT overflow period. After this, preparation for erase mode (erase setup) is carried out by setting the ESUn bit in FLMCRn, and after the elapse of 200 µs or more, the operating mode is switched to erase mode by setting the En bit in FLMCRn. The time during which the En bit is set is the flash memory erase time. Set an erase time of 5 ms.

Note: With flash memory erasing, preprogramming (setting all memory data in the memory to be erased to all “0”) is not necessary before starting the erase procedure.

22.7.4 Erase-Verify Mode (n = 1 for Addresses H'00000–H'1FFFF, n = 2 for Addresses H'20000–H'3FFFF)

In erase-verify mode, data is read after memory has been erased to check whether it has been correctly erased.

After the elapse of the erase time, erase mode is exited (the En bit in FLMCRn is released, then the ESUn bit is released at least 10 µs later), the watchdog timer is released after the elapse of 10 µs or more, and the operating mode is switched to erase-verify mode by setting the EVn bit in FLMCRn. Before reading in erase-verify mode, a dummy write of H'FF data should be made to the addresses to be read. The dummy write should be executed after the elapse of 20 µs or more. When the flash memory is read in this state (verify data is read in 32-bit units), the data at the latched address is read. Wait at least 2 µs after the dummy write before performing this read operation. If the read data has been erased (all “1”), a dummy write is performed to the next address, and erase-verify is performed. If the read data is unerased, set erase mode again and repeat the erase/erase-verify sequence in the same way. However, ensure that the erase/erase-verify sequence is not repeated more than 60 times. When verification is completed, exit erase-verify mode, and wait for at least 5 µs. If erasure has been completed on all the erase blocks after completing erase-verify operations on all these blocks, release the SWE bit in FLMCR1. If there are any unerased blocks, set erase mode again, and repeat the erase/erase-verify sequence as before. However, ensure that the erase/erase-verify sequence is not repeated more than 60 times.



Notes: *1 Preprogramming (setting erase block data to all “0”) is not necessary.
*2 Verify data is read in 32-bit (longword) units.
*3 Set only one bit in EBR1(2). More than one bit cannot be set.
*4 Erasing is performed in block units. To erase a number of blocks, each block must be erased in turn.
*5 Make sure to set the wait times and repetitions as specified. Erasing may not complete correctly if values other than the specified ones are used.

Figure 22.14 Erase/Erase-Verify Flowchart (Single Block Erase)

- Sample one-block erase program

  The wait time set values (number of loops) are for the case where f = 28.7 MHz. For other frequencies, the set value is given by the following expression:

  Wait time (μS) × f (MHz) ÷ 4

  The WDT overflow cycle set value is for the case where f = 28.7 MHz. For other frequencies, ensure that the overflow cycle is a minimum of 5.3 ms.

Registers Used

R5 (input): Memory block table pointer
R7 (output): OK (normal) or NG (error)
R0-3, 6, 8-9: Work registers

```
FLMCR1          .EQU    H'80
FLMCR2          .EQU    H'81
EBR1            .EQU    H'82
EBR2            .EQU    H'83
Wait10u         .EQU    72
Wait2u          .EQU    14
Wait200u        .EQU    1435
Wait5m          .EQU    35875
Wait20u         .EQU    144
Wait5u          .EQU    36
WDT_TCSR        .EQU    H'FFFF8610
WDT_9m          .EQU    H'A57D
SWESET          .EQU    B'01000000
ESUSET          .EQU    B'00100000
ESET            .EQU    B'00000010
ECLEAR          .EQU    B'11111101
ESUCLEAR        .EQU    B'11011111
EVSET           .EQU    B'00001000
EVCLEAR         .EQU    B'11110111
SWECLEAR        .EQU    B'10111111
MAXErase        .EQU    60
;
FlashErase      .EQU    $
        MOV.L   #H'FFFF8500,R0
        LDC     R0,GBR                  ; Initialize GBR
        MOV.L   #1,R2
```

```
;
        MOV.L     #Wait10u,R3
        MOV.L     #FLMCR1,R0
        OR.B      #SWESET,@(R0,GBR)       ; Set SWE
EWait_1 SUBC      R2,R3                   ; Wait 10 µs
        BF        EWait_1
;
        MOV.L     #0,R9                   ; Initialize n (R9) to 0
;
        MOV.B     @(6,R5),R0
        MOV.B     R0,@(EBR1,GBR)          ; Erase memory block (EBR1) setting
        MOV.B     @(7,R5),R0
        MOV.B     R0,@(EBR2,GBR)          ; Erase memory block (EBR2) setting
;
        MOV.L     #FLMCR1,R0
        MOV.L     @R5,R6                  ; Erase memory block start address -> R6
        MOV.L     #H'020000,R7
        CMP/GT    R6,R7
        BT        EraseLoop
        MOV.L     #FLMCR2,R0
;
EraseLoop         .EQU     $
        MOV.L     #WDT_TCSR,R1            ; Enable WDT
        MOV.W     #WDT_9m,R3              ; 9.2 ms cycle
        MOV.W     R3,@R1
;
        MOV.L     #Wait200u,R3
        OR.B      #ESUSET,@(R0,GBR)       ; Set ESU
EWait_2 SUBC      R2,R3                   ; Wait 200 µs
        BF        EWait_2
;
        MOV.L     #Wait5m,R3
        OR.B      #ESET,@(R0,GBR)         ; Set E
EWait_3 SUBC      R2,R3                   ; Wait 5 ms
        BF        EWait_3
;
        MOV.L     #Wait10u,R3
```

```
        AND.B   #ECLEAR,@(R0,GBR)       ; Clear E
EWait_4 SUBC    R2,R3                   ; Wait 10 µs
        BF      EWait_4
;
        MOV.L   #Wait10u,R3
        AND.B   #ESUCLEAR,@(R0,GBR)     ; Clear ESU
EWait_5 SUBC    R2,R3                   ; Wait 10 µs
        BF      EWait_5
;
        MOV.L   #WDT_TCSR,R1            ; Disable WDT
        MOV.W   #H'A55F,R3
        MOV.W   R3,@R1
;
        MOV.L   #Wait20u,R3
        OR.B    #EVSET,@(R0,GBR)        ; Set EV
EWait_6 SUBC    R2,R3                   ; Wait 20 µs
        BF      EWait_6
;
        MOV.L   @R5,R6                  ; Erase memory block start address -> R6
BlockVerify_1   .EQU    $               ; Erase-verify
        MOV.L   #H'FFFFFFFF,R8
        MOV.L   R8,@R6                  ; H'FF dummy write
        MOV.L   #Wait2u,R3
EWait_7 SUBC    R2,R3
        BF      EWait_7
;
        MOV.L   @R6+,R1                 ; Read verify data
        CMP/EQ  R8,R1
        BF      BlockVerify_NG
        MOV.L   @(8,R5),R7
        CMP/EQ  R6,R7                   ; Check for last address of memory block
        BF      BlockVerify_1
        MOV.L   #Wait5u,R3
        AND.B   #EVCLEAR,@(R0,GBR)      ; Clear EV
EWait_8 SUBC    R2,R3                   ; Wait 5 µs
        BF      EWait_8
;
```

```
        MOV.L   #OK,R7                  ; R7 <- OK (return value)
        BRA     FlashErase_end          ; Verify OK
        NOP
;
BlockVerify_NG  .EQU    $
        ADD.L   #1,R9                   ; Verify NG, n <- n + 1
        MOV.L   #Wait5u,R3
        AND.B   #EVCLEAR,@(R0,GBR)      ; Clear EV
EWait_9 SUBC    R2,R3                   ; Wait 5 µs
        BF      EWait_9
        MOV.L   #MAXErase,R7            ; If n > MAXErase then erase NG
        CMP/EQ  R7,R9
        BF      EraseLoop
        MOV.L   #NG,R7                  ; R7 <- NG (return value)
FlashErase_end  .EQU    $
        MOV.L   #FLMCR1,R0
        AND.B   #SWECLEAR,@(R0,GBR)     ; Clear SWE
;
        RTS
        NOP
;
; Memory block table    Memory block start address: EBR value
        .ALIGN  4
Flash_BlockData .EQU    $
EB0     .DATA.L H'00000000,H'00000100
EB1     .DATA.L H'00008000,H'00000200
EB2     .DATA.L H'00010000,H'00000400
EB3     .DATA.L H'00018000,H'00000800
EB4     .DATA.L H'00020000,H'00000001
EB5     .DATA.L H'00028000,H'00000002
EB6     .DATA.L H'00030000,H'00000004
EB7     .DATA.L H'00038000,H'00000008
EB8     .DATA.L H'0003F000,H'00000010
EB9     .DATA.L H'0003F400,H'00000020
EB10    .DATA.L H'0003F800,H'00000040
EB11    .DATA.L H'0003FC00,H'00000080

Dummy   .DATA.L H'00040000
```

22.8 Protection

There are two kinds of flash memory program/erase protection, hardware protection and software protection.

22.8.1 Hardware Protection

Hardware protection refers to a state in which programming/erasing of flash memory is forcibly disabled or aborted. Hardware protection is reset by settings in flash memory control register 1 (FLMCR1), flash memory control register 2 (FLMCR2), erase block register 1 (EBR1), and erase block register 2 (EBR2). The FLMCR1, FLMCR2, EBR1, and EBR2 settings are retained in the error-protected state. (See table 22.8.)

Table 22.8 Hardware Protection

		Function	
Item	**Description**	**Program**	**Erase**
FWP pin protection	• When a high level is input to the FWP pin, FLMCR1, FLMCR2, EBR1, and EBR2 are initialized, and the program/erase-protected state is entered.	Yes	Yes
Reset/standby protection	• In a reset (including a WDT overflow reset) and in standby mode, FLMCR1, FLMCR2, EBR1, and EBR2 are initialized, and the program/erase-protected state is entered. • In a reset via the $\overline{\text{RES}}$ pin, the reset state is not entered unless the $\overline{\text{RES}}$ pin is held low until oscillation stabilizes after powering on. In the case of a reset during operation, hold the $\overline{\text{RES}}$ pin low for the $\overline{\text{RES}}$ pulse width specified in the AC Characteristics section.	Yes	Yes

22.8.2 Software Protection

Software protection can be implemented by setting the SWE bit in FLMCR1, erase block register 1 (EBR1), erase block register 2 (EBR2), and the RAMS bit in the RAM emulation register (RAMER). When software protection is in effect, setting the P1 or E1 bit in flash memory control register 1 (FLMCR1), or the P2 or E2 bit in flash memory control register 2 (FLMCR2), does not cause a transition to program mode or erase mode. (See table 22.9.)

Software protect can be enabled by setting the SWE bit of FLMCR1, block specification register 1 (EBR1), block specification register 2 (EBR2) and the RAMS bit of the RAM emulation register. During software protect, transition cannot be made to the program mode or the erase mode even when setting P1 or E1 bits of the flash memory control register 1 (FLMCR1), or P2 or E2 bits of flash memory control register 2 (FLMCR2). (See table 22.9.)

Table 22.9 Software Protection

Item	Description	Function	
		Program	Erase
SWE bit protection	• Clearing the SWE bit to 0 in FLMCR1 sets the program/erase-protected state for all blocks. (Execute in on-chip RAM or external memory.)	Yes	Yes
Block specification protection	• Erase protection can be set for individual blocks by settings in erase block register 1 (EBR1) and erase block register 2 (EBR2). • Setting EBR1 and EBR2 to H'00 places all blocks in the erase-protected state.	—	Yes
Emulation protection	• Setting the RAMS bit to 1 in the RAM emulation register (RAMER) places all blocks in the program/erase-protected state.	Yes	Yes

22.8.3 Error Protection

In error protection, an error is detected when microcomputer runaway occurs during flash memory programming/erasing, or operation is not performed in accordance with the program/erase algorithm, and the program/erase operation is aborted. Aborting the program/erase operation prevents damage to the flash memory due to overprogramming or overerasing.

If the SH7051 malfunctions during flash memory programming/erasing, the FLER bit is set to 1 in FLMCR2 and the error protection state is entered. The FLMCR1, FLMCR2, EBR1, and EBR2 settings are retained, but program mode or erase mode is aborted at the point at which the error occurred. Program mode or erase mode cannot be re-entered by re-setting the P1, P2, E1, or E2 bit. However, PV1, PV2, EV1, and EV2 bit setting is enabled, and a transition can be made to verify mode.

FLER bit setting conditions are as follows:

1. When flash memory is read during programming/erasing (including a vector read or instruction fetch)
2. Immediately after exception handling (excluding a reset) during programming/erasing
3. When a SLEEP instruction (including software standby) is executed during programming/erasing
4. When the bus is released during programming/erasing

Error protection is released only by a reset and in hardware standby mode.

Figure 22.15 shows the flash memory state transition diagram.

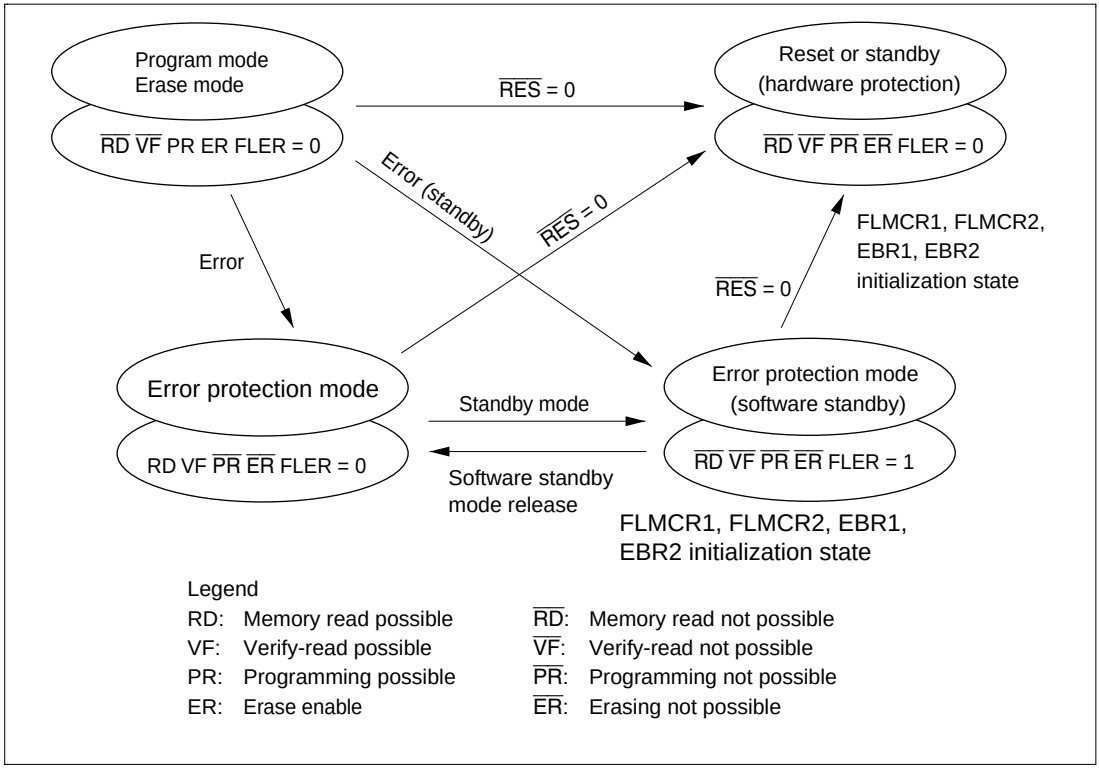


Figure 22.15 Flash Memory State Transitions

22.9 Flash Memory Emulation in RAM

Making a setting in the RAM emulation register (RAMER) enables part of RAM to be overlapped onto the flash memory area so that data to be written to flash memory can be emulated in RAM in real time. After the RAMER setting has been made, accesses can be made from the flash memory area or the RAM area overlapping flash memory. Emulation can be performed in user mode and user program mode. Figure 22.16 shows an example of emulation of real-time flash memory programming.

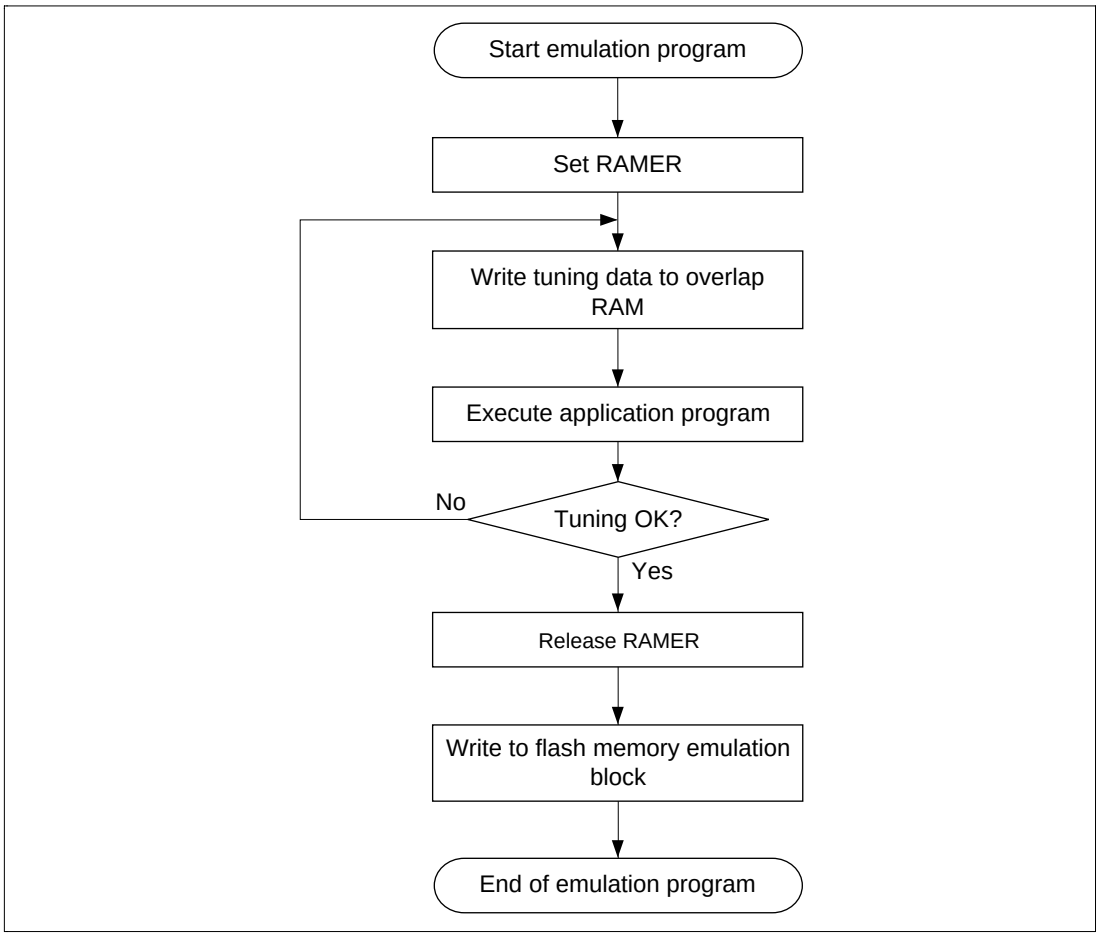


Figure 22.16 Flowchart for Flash Memory Emulation in RAM

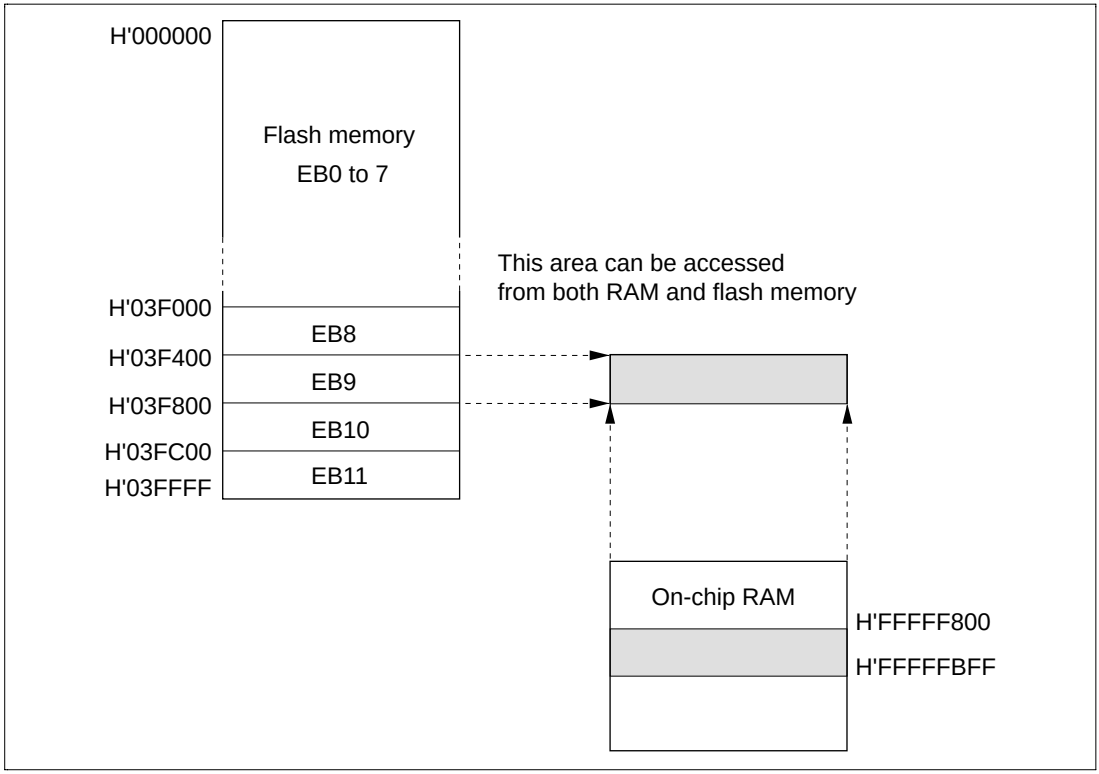


Figure 22.17 Example of RAM Overlap Operation

Example in which Flash Memory Block Area (EB8) is Overlapped

1. Set bits RAMS, RAM1, and RAM0 in RAMER to 1, 0, 1, to overlap part of RAM onto the area (EB8) for which real-time programming is required.
2. Real-time programming is performed using the overlapping RAM.
3. After the program data has been confirmed, the RAMS bit is cleared, releasing RAM overlap.
4. The data written in the overlapping RAM is written into the flash memory space (EB8).

Notes: 1. When the RAMS bit is set to 1, program/erase protection is enabled for all blocks regardless of the value of RAM1 and RAM0 (emulation protection). In this state, setting the P1 or E1 bit in flash memory control register 1 (FLMCR1), or the P2 or E2 bit in flash memory control register 2 (FLMCR2), will not cause a transition to program mode or erase mode. When actually programming or erasing a flash memory area, the RAMS bit should be cleared to 0.

2. A RAM area cannot be erased by execution of software in accordance with the erase algorithm while flash memory emulation in RAM is being used.

22.10 Note on Flash Memory Programming/Erasing

In the on-board programming modes (user mode and user program mode), NMI input should be disabled to give top priority to the program/erase operations (including RAM emulation).

22.11 Flash Memory Programmer Mode

Programs and data can be written and erased in programmer mode as well as in the on-board programming modes. In programmer mode, flash memory read mode, auto-program mode, auto-erase mode, and status read mode are supported. In auto-program mode, auto-erase mode, and status read mode, a status polling procedure is used, and in status read mode, detailed internal signals are output after execution of an auto-program or auto-erase operation.

In programmer mode, set the mode pins to PLL x2 mode (see table 22.10) and use a 6 MHz input clock. The LSI will then operate at 12 MHz.

Table 22.10 shows the pin settings for programmer mode. For the pin names in programmer mode, see section 1.3.2, Pin Arrangement by Mode).

Table 22.10 Programming Mode Pin Settings

Pin Names	Settings
Mode pin: MD3, MD2, MD1, MD0	1101 (PLL × 2)
FWE pin	High level input (in auto-program and auto-erase modes)
$\overline{\text{RES}}$ pin	Power-on reset circuit
XTAL, EXTAL, PLLVcc, PLLCAP, and PLLVss pins	Oscillator circuit

Note: During the programming mode, polarity of the FWP pin is inverted and becomes the FWE (flash write enable) pin.

22.11.1 Socket Adapter Pin Correspondence Diagrams

Connect the socket adapter to the chip as shown in figures 22.19 and 22.20. This will enable conversion to a 32-pin arrangement. The on-chip ROM memory map is shown in figure 22.18, and socket adapter pin correspondence diagrams in figures 22.19 and 22.20.

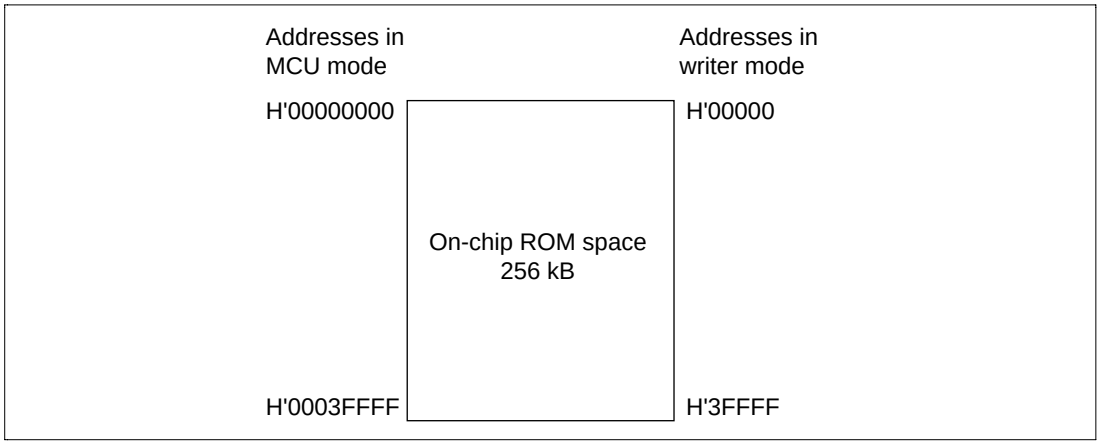


Figure 22.18 On-Chip ROM Memory Map

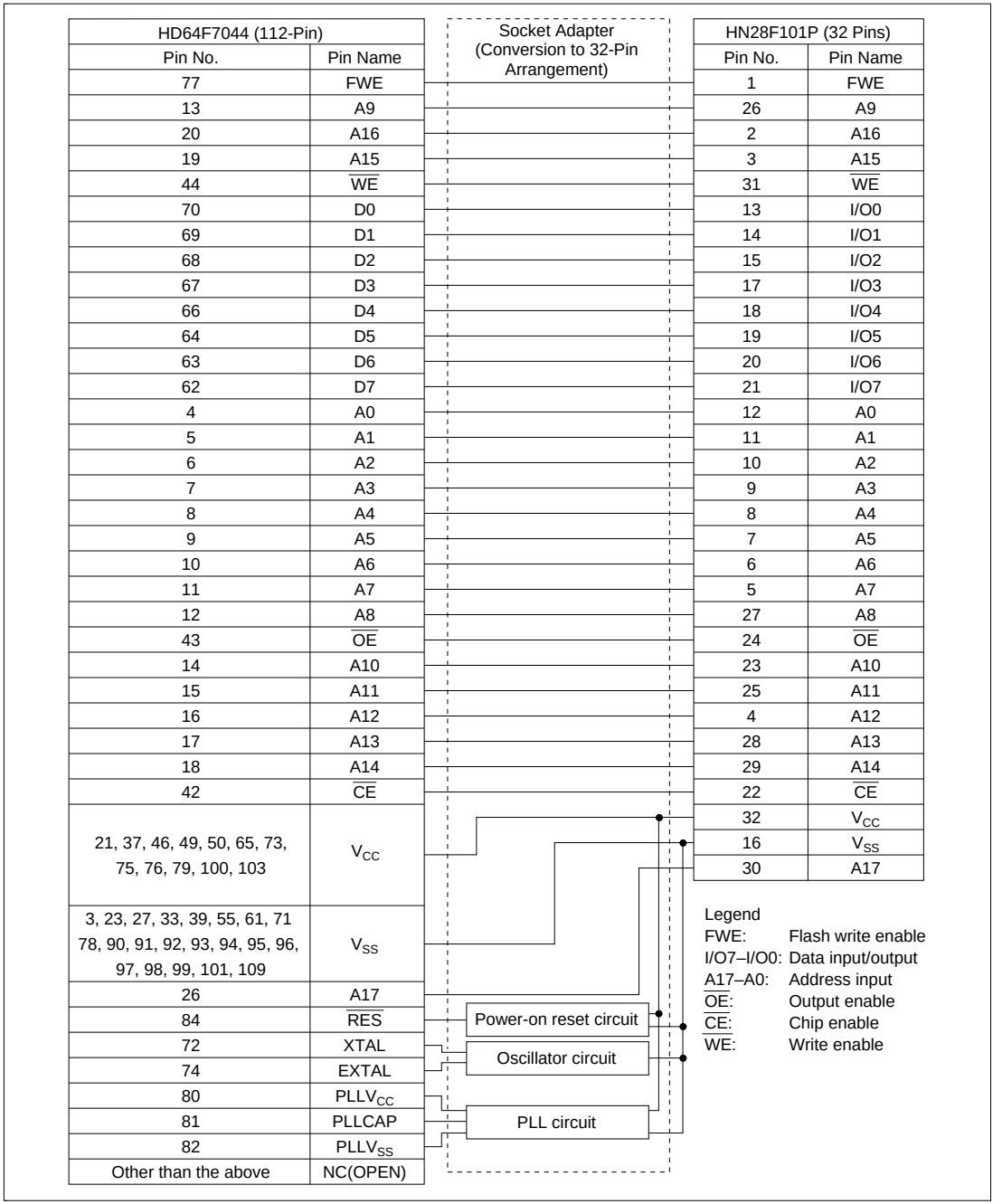


Figure 22.19 Socket Adapter Pin Correspondence Diagram (SH7044)

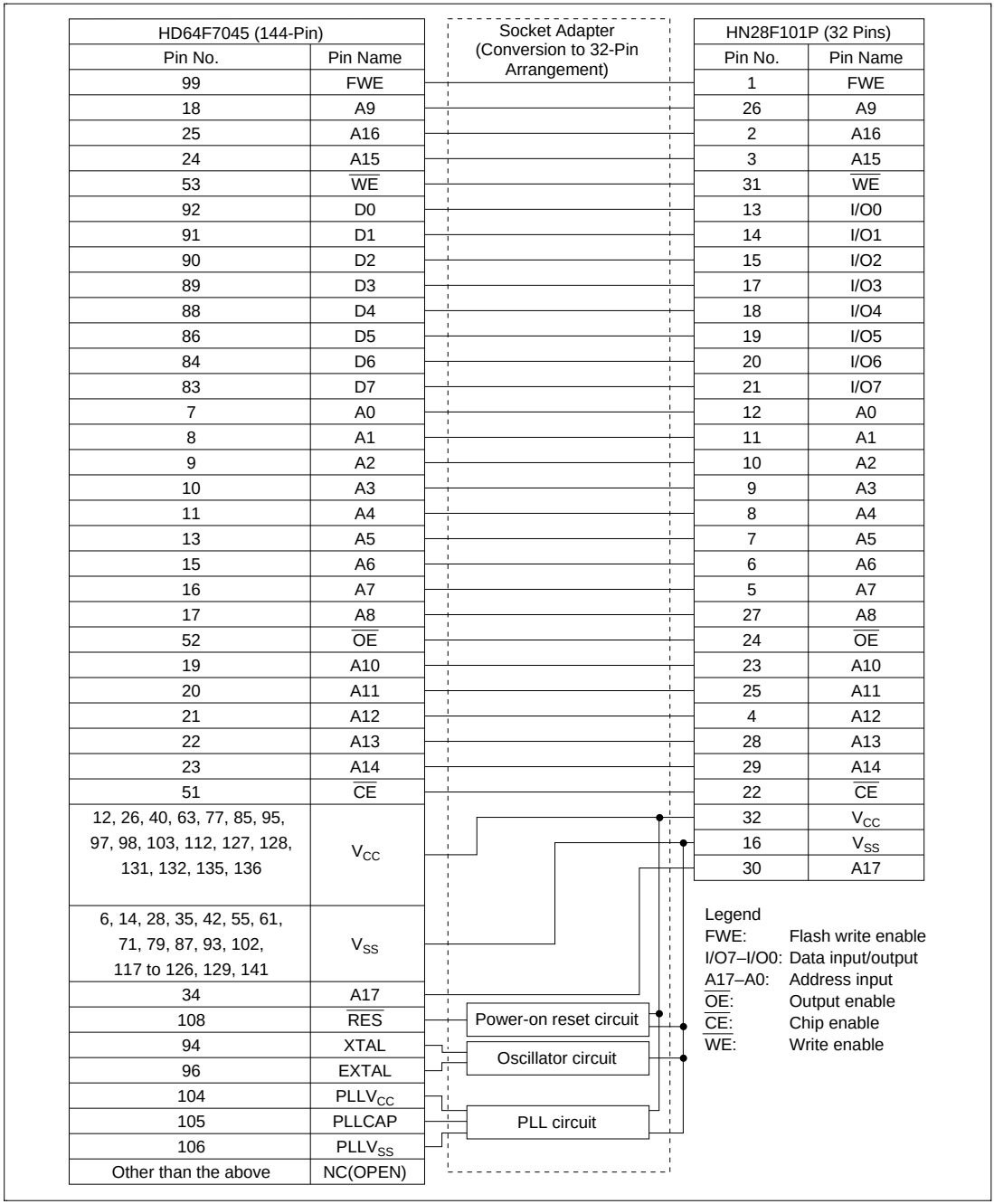


Figure 22.20 Socket Adapter Pin Correspondence Diagram (SH7045)

22.11.2 Programmer Mode Operation

Table 22.11 shows how the different operating modes are set when using programmer mode, and table 22.12 lists the commands used in programmer mode. Details of each mode are given below.

- Memory Read Mode

  Memory read mode supports byte reads.
- Auto-Program Mode

  Auto-program mode supports programming of 128 bytes at a time. Status polling is used to confirm the end of auto-programming.
- Auto-Erase Mode

  Auto-erase mode supports automatic erasing of the entire flash memory. Status polling is used to confirm the end of auto-programming.
- Status Read Mode

  Status polling is used for auto-programming and auto-erasing, and normal termination can be confirmed by reading the I/O6 signal. In status read mode, error information is output if an error occurs.

Table 22.11 Settings for Various Operating Modes In Programmer Mode

	Pin names					
Mode	**FWE**	**$\overline{CE}$**	**$\overline{OE}$**	**$\overline{WE}$**	**I/O7–0**	**A17–0**
Read	H or L	L	L	H	Data output	Ain
Output disable	H or L	L	H	H	Hi-z	Ain
Command write	H or L	L	H	L	Data input	*Ain
Chip disable	H or L	H	X	X	Hi-z	Ain

Notes: *Ain indicates that there is also address input in auto-program mode.

1. Chip disable is not a standby state; internally, it is an operation state.
2. For command writes in auto-program and auto-erase modes, input a high level to the FWE pin.

Table 22.12 Commands of the Programmer Mode

Command Name	Number of Cycles	First Cycle			Second Cycle		
		Mode	Address	Data	Mode	Address	Data
Memory read mode	1+n	write	X	H'00	read	RA	Dout
Auto-program mode	129	write	X	H'40	write	WA	Din
Auto-erase mode	2	write	X	H'20	write	X	H'20
Status read mode	2	write	X	H'71	write	X	H'71

Notes: 1. In auto-program mode, 129 cycles are required for command writing by a simultaneous 128-byte write.
2. In memory read mode, the number of cycles depends on the number of address write cycles (n).

22.11.3 Memory Read Mode

Table 22.13 AC Characteristics in Transition to Memory Read Mode (Conditions: V_{CC} = 5.0 V ±10%, V_{SS} = 0 V, T_a = 25°C ±5°C)

Item	Symbol	Min	Max	Unit	Notes
Command write cycle	t_{nxtc}	20		µs	
$\overline{CE}$ hold time	t_{ceh}	0		ns	
$\overline{CE}$ setup time	t_{ces}	0		ns	
Data hold time	t_{dh}	50		ns	
Data setup time	t_{ds}	50		ns	
Write pulse width	t_{wep}	70		ns	
$\overline{WE}$ rise time	t_r		30	ns	
$\overline{WE}$ fall time	t_f		30	ns	

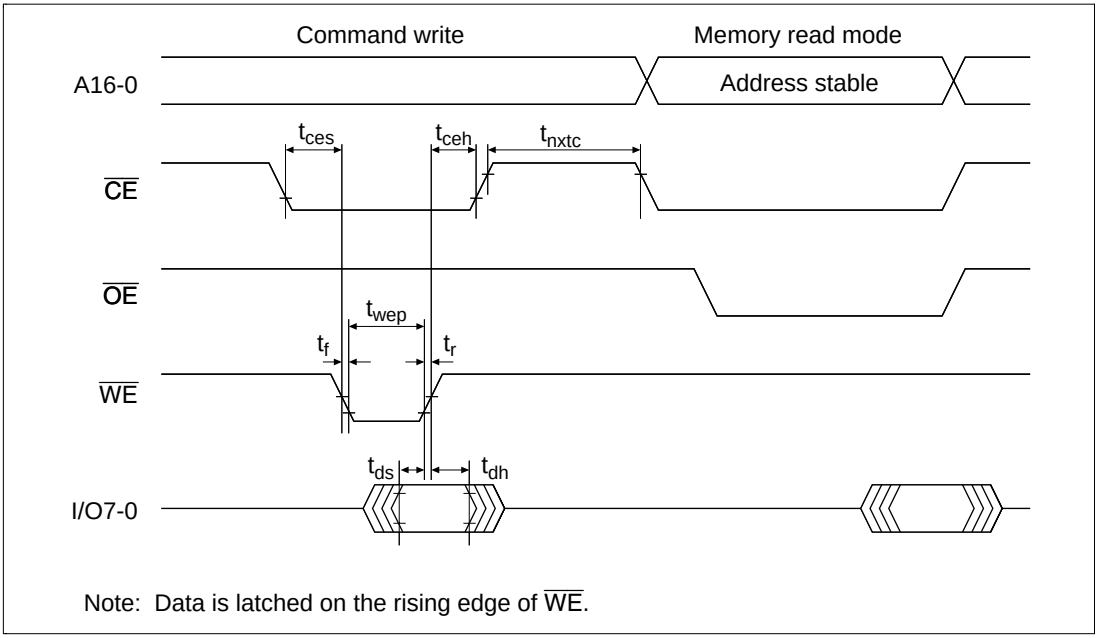
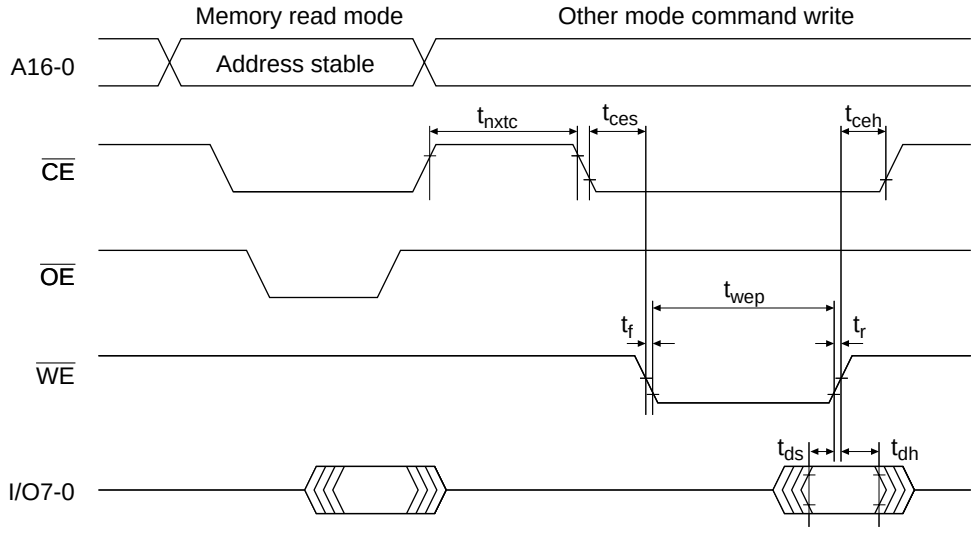


Figure 22.21 Timing Waveforms for Memory Read after Memory Write

Table 22.14 AC Characteristics in Transition from Memory Read Mode to Another Mode (Conditions: V_{CC} = 5.0 V ±10%, V_{SS} = 0 V, T_a = 25°C ±5°C)

Item	Symbol	Min	Max	Unit	Notes
Command write cycle	t_{nxtc}	20		µs	
$\overline{CE}$ hold time	t_{ceh}	0		ns	
$\overline{CE}$ setup time	t_{ces}	0		ns	
Data hold time	t_{dh}	50		ns	
Data setup time	t_{ds}	50		ns	
Write pulse width	t_{wep}	70		ns	
$\overline{WE}$ rise time	t_r		30	ns	
$\overline{WE}$ fall time	t_f		30	ns	



Note: Do not enable $\overline{WE}$ and $\overline{OE}$ at the same time.

Figure 22.22 Timing Waveforms in Transition from Memory Read Mode to Another Mode

Table 22.15 AC Characteristics in Memory Read Mode (Conditions: V_{CC} = 5.0 V ±10%, V_{SS} = 0 V, T_a = 25°C ±5°C)

Item	Symbol	Min	Max	Unit	Notes
Access time	t_{acc}		20	µs	
$\overline{CE}$ output delay time	t_{ce}		150	ns	
$\overline{OE}$ output delay time	t_{oe}		150	ns	
Output disable delay time	t_{df}		100	ns	
Data output hold time	t_{oh}	5		ns	

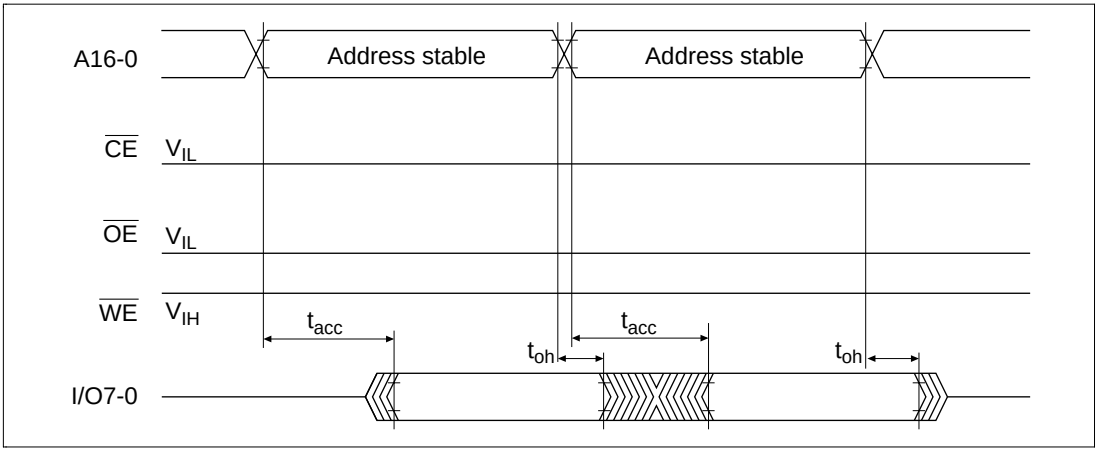


Figure 22.23 $\overline{CE}$ and $\overline{OE}$ Enable State Read Timing Waveforms

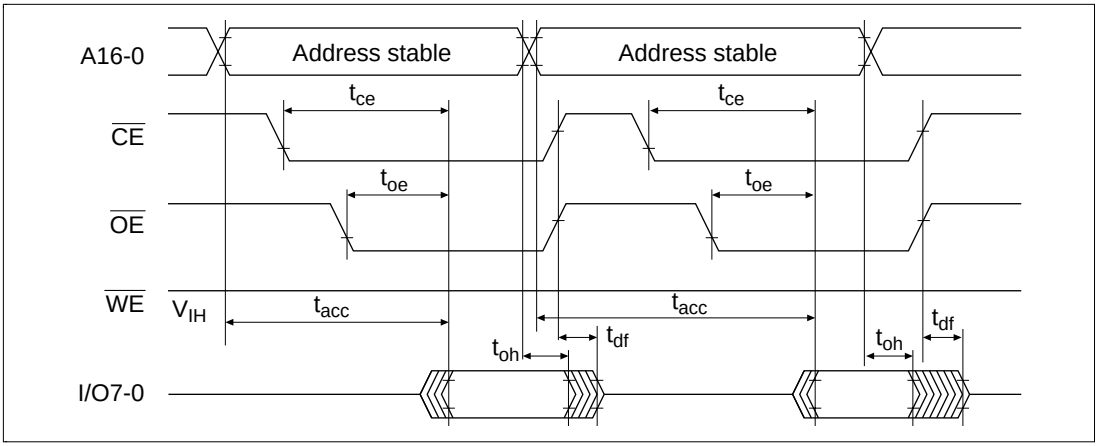


Figure 22.24 $\overline{CE}$ and $\overline{OE}$ Clock System Read Timing Waveforms

22.11.4 Auto-Program Mode

1. In auto-program mode, 128 bytes are programmed simultaneously. This should be carried out by executing 128 consecutive byte transfers.
2. A 128-byte data transfer is necessary even when programming fewer than 128 bytes. In this case, H'FF data must be written to the extra addresses.
3. The lower 8 bits of the transfer address must be H'00 or H'80. If a value other than an effective address is input, processing will switch to a memory write operation but a write error will be flagged.
4. Memory address transfer is performed in the second cycle (figure 22.24). Do not perform transfer after the second cycle.
5. Do not perform a command write during a programming operation.
6. Perform one auto-program operation for a 128-byte block for each address. Characteristics are not guaranteed for two or more additional programming operations.
7. Confirm normal end of auto-programming by checking I/O6. Alternatively, status read mode can also be used for this purpose (I/O7 status output uses the auto-program operation end identification pin).
8. Status polling I/O6 and I/O7 pin information is retained until the next command write. As long as the next command write has not been performed, reading is possible by enabling $\overline{CE}$ and $\overline{OE}$.

Table 22.16 AC Characteristics in Auto-Program Mode (Conditions: V_{CC} = 5.0 V ±10%, V_{SS} = 0 V, T_a = 25°C ±5°C)

Item	Symbol	Min	Max	Unit	Notes
Command write cycle	t_{nxtc}	20		µs	
$\overline{CE}$ hold time	t_{ceh}	0		ns	
$\overline{CE}$ setup time	t_{ces}	0		ns	
Data hold time	t_{dh}	50		ns	
Data setup time	t_{ds}	50		ns	
Write pulse width	t_{wep}	70		ns	
Status polling start time	t_{wsts}	1		ms	
Status polling access time	t_{spa}		150	ns	
Address setup time	t_{as}	0		ns	
Address hold time	t_{ah}	60		ns	
Memory write time	t_{write}	1	3000	ms	
Write setup time	t_{pns}	100		ns	
Write end setup time	t_{pnh}	100		ns	
$\overline{WE}$ rise time	t_r		30	ns	
$\overline{WE}$ fall time	t_f		30	ns	

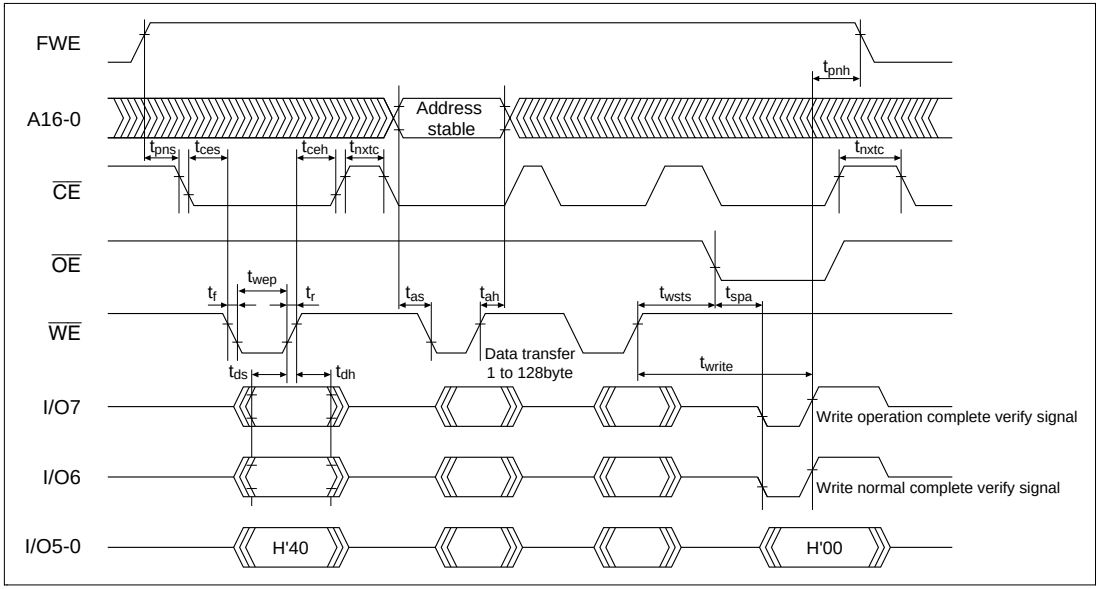


Figure 22.25 Auto-Program Mode Timing Waveforms

22.11.5 Auto-Erase Mode

1. Auto-erase mode supports only entire memory erasing.
2. Do not perform a command write during auto-erasing..
3. Confirm normal end of auto-erasing by checking I/O6. Alternatively, status read mode can also be used for this purpose (I/O7 status output uses the auto-erase operation end identification pin).
4. Status polling I/O6 and I/O7 pin information is retained until the next command write. As long as the next command write has not been performed, reading is possible by enabling $\overline{CE}$ and $\overline{OE}$.

Table 22.17 AC Characteristics in Auto-Erase Mode (Conditions: V_{CC} = 5.0 V ±10%, V_{SS} = 0 V, T_a = 25°C ±5°C)

Item	Symbol	Min	Max	Unit	Notes
Command write cycle	t_{nxtc}	20		µs	
$\overline{CE}$ hold time	t_{ceh}	0		ns	
$\overline{CE}$ setup time	t_{ces}	0		ns	
Data hold time	t_{dh}	50		ns	
Data setup time	t_{ds}	50		ns	
Write pulse width	t_{wep}	70		ns	
Status polling start time	t_{ests}	1		ms	
Status polling access time	t_{spa}		150	ns	
Memory erase time	t_{erase}	100	40000	ms	
Erase setup time	t_{ens}	100		ns	
Erase end setup time	t_{enh}	100		ns	
$\overline{WE}$ rise time	t_r		30	ns	
$\overline{WE}$ fall time	t_f		30	ns	

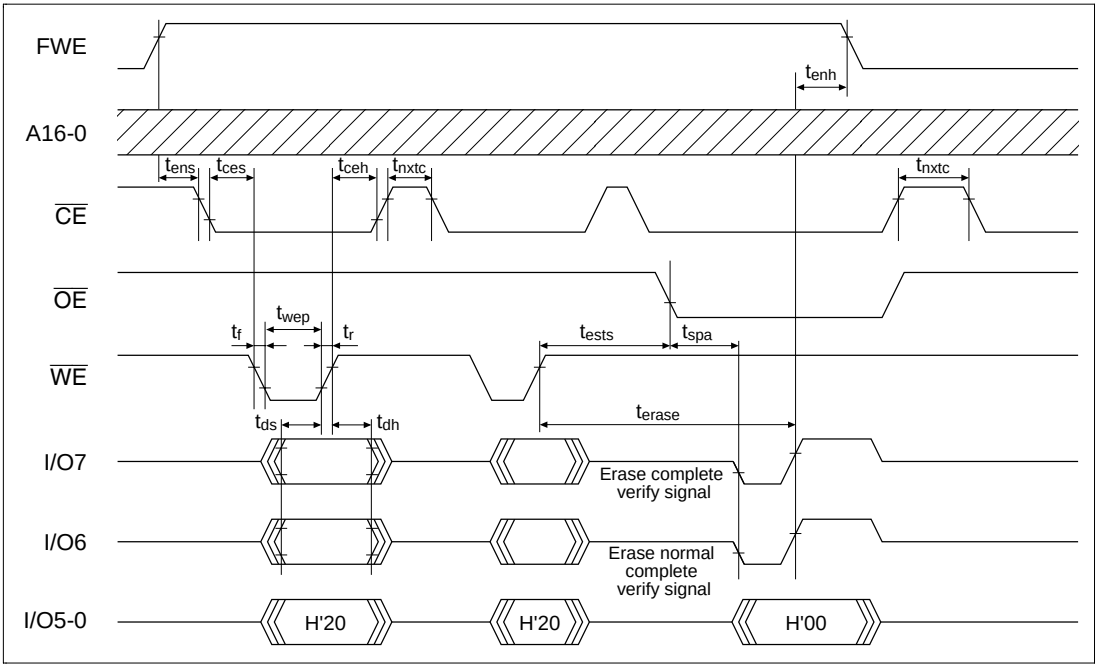


Figure 22.26 Auto-Erase Mode Timing Waveforms

22.11.6 Status Read Mode

Table 22.18 AC Characteristics in Status Read Mode (Conditions: V_{CC} = 5.0 V ±10%, V_{SS} = 0 V, T_a = 25°C ±5°C)

Item	Symbol	Min	Max	Unit	Notes
Read time after command write	t_{std}	20		µs	
$\overline{CE}$ hold time	t_{ceh}	0		ns	
$\overline{CE}$ setup time	t_{ces}	0		ns	
Data hold time	t_{dh}	50		ns	
Data setup time	t_{ds}	50		ns	
Write pulse width	t_{wep}	70		ns	
$\overline{OE}$ output delay time	t_{oe}		150	ns	
Disable delay time	t_{df}		100	ns	
$\overline{CE}$ output delay time	t_{ce}		150	ns	
$\overline{WE}$ rise time	t_r		30	ns	
$\overline{WE}$ fall time	t_f		30	ns	

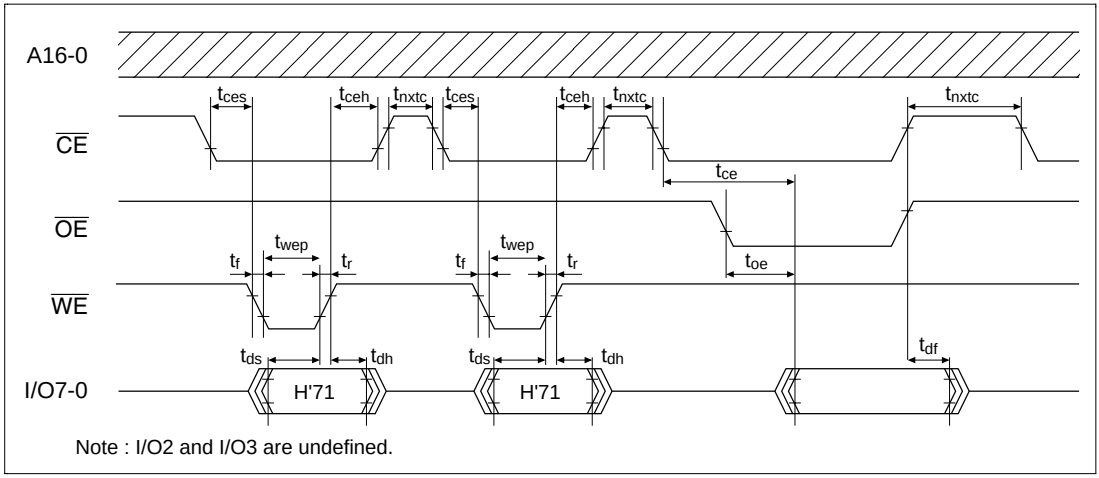


Figure 22.27 Status Read Mode Timing Waveforms

Table 22.19 Return Commands for the Status Read Mode

Pin Name	I/O7	I/O6	I/O5	I/O4	I/O3	I/O2	I/O1	I/O0
Attribute	Normal end identification	Command error	Program-ming error	Erase error	—	—	Program-ming or erase count exceeded	Effective address error
Initial value	0	0	0	0	0	0	0	0
Indications	Normal end: 0 Abnormal end: 1	Command Error: 1 Otherwise: 0	Program-ming Error: 1 Otherwise: 0	Erasing Error: 1 Otherwise: 0	—	—	Count exceeded: 1 Otherwise: 0	Effective address Error: 1 Otherwise: 0

Note: D2 and D3 are undefined at present.

22.11.7 Status Polling

1. I/O7 status polling is a flag that indicates the operating status in auto-program/auto-erase mode.
2. I/O6 status polling is a flag that indicates a normal or abnormal end in auto-program/auto-erase mode.

Table 22.20 Status Polling Output Truth Table

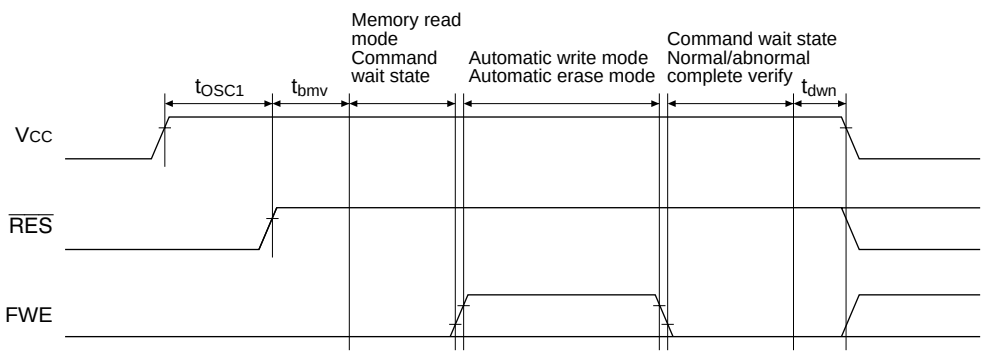
Pin Name	During Internal Operation	Abnormal End		Normal End
I/O7	0	1	0	1
I/O6	0	0	1	1
I/O5–0	0	0	0	0

22.11.8 Programmer Mode Transition Time

Commands cannot be accepted during the oscillation stabilization period or the programmer mode setup period. After the programmer mode setup time, a transition is made to memory read mode.

Table 22.21 Stipulated Transition Times to Command Wait State

Item	Symbol	Min	Max	Unit	Notes
Standby release (oscillation stabilization time)	t_{OSC1}	10		ms	
Programmer mode setup time	t_{bmv}	10		ms	
V_{CC} hold time	t_{dwn}	0		ms	



Note : For the level of FWE input pin, set V_{IL} when using other than the automatic write mode and automatic erase mode.

Figure 22.28 Oscillation Stabilization Time and Boot Program Transfer Time

22.11.9 Cautions Concerning Memory Programming

1. When programming addresses which have previously been programmed, carry out auto-erasing before auto-programming.
2. When performing programming using PROM mode on a chip that has been programmed/erased in an on-board programming mode, auto-erasing is recommended before carrying out auto-programming.

Notes: 1. The flash memory is initially in the erased state when the device is shipped by Renesas Technology. For other chips for which the erasure history is unknown, it is recommended that auto-erasing be executed to check and supplement the initialization (erase) level.
2. Auto-programming should be performed once only on the same address block. Additional programming cannot be performed on previously programmed address blocks.

23.1 Overview

The SH7040 series has 4 kbytes of on-chip RAM. The on-chip RAM is linked to the CPU and direct memory access controller (DMAC)/data transfer controller (DTC) with a 32-bit data bus (figure 23.1). The CPU can access data in the on-chip RAM in 8, 16, or 32 bit widths. The DMAC can access 8 or 16 bit widths. On-chip RAM data can always be accessed in one state, making the RAM ideal for use as a program area, stack area, or data area, which require high-speed access. The contents of the on-chip RAM are held in both the sleep and standby modes. Memory area 0 addresses H'FFFFF000–H'FFFFFFFF are allocated to the on-chip RAM.

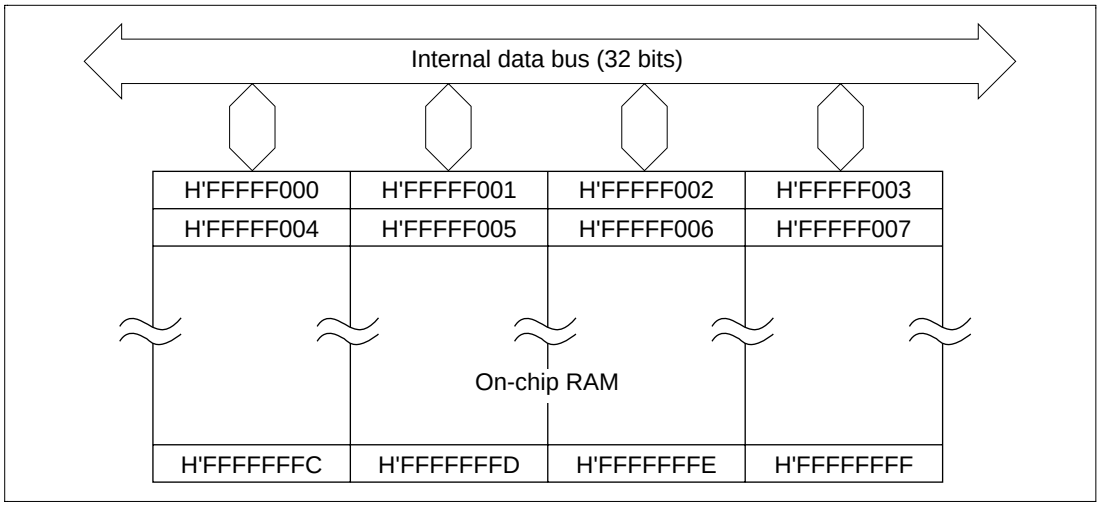


Figure 23.1 Block Diagram of RAM

23.2 Operation

The on-chip RAM is accessed by accessing addresses H'FFFFF000–H'FFFFFFFF. On-chip RAM is also used as cache memory. There are 2 kbytes of on-chip RAM space during cache use. See section 9, Cache Memory (CAC), for details.

Section 24 Power-Down State

24.1 Overview

In the power-down state, the CPU functions are halted. This enables a great reduction in power consumption.

24.1.1 Power-Down States

The power-down state is effected by the following two modes:

- Sleep mode
- Standby mode

Table 24.1 describes the transition conditions for entering the modes from the program execution state as well as the CPU and peripheral function status in each mode and the procedures for canceling each mode.

Table 24.1 Power-Down State Conditions

		State						
Mode	**Entering Procedure**	**Clock**	**CPU**	**On-Chip Peripheral Modules**	**CPU Registers**	**RAM**	**I/O Ports**	**Canceling Procedure**
Sleep	Execute SLEEP instruction with SBY bit set to 0 in SBYCR	Run	Halt	Run	Held	Held	Held	• Interrupt • DMAC/DTC address error • Power-on reset • Manual reset
Stand-by	Execute SLEEP instruction with SBY bit set to 1 in SBYCR	Halt	Halt	Halt*1	Held	Held	Held or high impe-dance*2	• NMI interrupt • Power-on reset • Manual reset

Notes: SBYCR: standby control register. SBY: standby bit

*1 Some bits within on-chip peripheral module registers are initialized by the standby mode; some are not. Refer to table 24.3, Register States in the Standby Mode, in section 24.4.1, Transition to Standby Mode. Also refer to the register descriptions for each peripheral module.

*2 The status of the I/O port in standby mode is set by the port high impedance bit (HIZ) of the SBYCR. Refer to section 24.2, Standby Control Register (SBYCR). For pin status other than for the I/O port, refer to Appendix C, Pin Status.

24.1.2 Related Register

Table 24.2 shows the register used for power-down state control.

Table 24.2 Related Register

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Standby control register	SBYCR	R/W	H'1F	H'FFFF8614	8, 16, 32

24.2 Standby Control Register (SBYCR)

The standby control register (SBYCR) is a read/write 8-bit register that sets the transition to standby mode, and the port status in standby mode. The SBYCR is initialized to H'1F when reset.

Bit:	7	6	5	4	3	2	1	0
	SBY	HIZ	—	—	—	—	—	—
Initial value:	0	0	0	1	1	1	1	1
R/W:	R/W	R/W	R	R	R	R	R	R

- Bit 7—Standby (SBY): Specifies transition to the standby mode. The SBY bit cannot be set to 1 while the watchdog timer is running (when the timer enable bit (TME) of the WDT timer control/status register (TCSR) is set to 1). To enter the standby mode, always halt the WDT by 0 clearing the TME bit, then set the SBY bit.

Bit 7: SBY	Description
0	Executing SLEEP instruction puts the LSI into sleep mode (initial value)
1	Executing SLEEP instruction puts the LSI into standby mode

- Bit 6—Port High Impedance (HIZ): In the standby mode, this bit selects whether to set the I/O port pin to high impedance or hold the pin status. The HIZ bit cannot be set to 1 when the TME bit of the WDT timer control/status register (TCSR) is set to 1. When making the I/O port pin status high impedance, always clear the TME bit to 0 before setting the HIZ bit.

Bit 6: HIZ	Description
0	Holds pin status while in standby mode (initial value)
1	Keeps pin at high impedance while in standby mode

- Bits 5–0—Reserved: Bit 5 always reads as 0. Always write 0 to bit 5. Bits 4–0 always read as 1. Always write 1 to these bits.

24.3 Sleep Mode

24.3.1 Transition to Sleep Mode

Executing the SLEEP instruction when the SBY bit of SBYCR is 0 causes a transition from the program execution state to the sleep mode. Although the CPU halts immediately after executing the SLEEP instruction, the contents of its internal registers remain unchanged. The on-chip peripheral modules continue to run during the sleep mode.

24.3.2 Canceling Sleep Mode

Sleep mode is canceled by an interrupt, DMAC/DTC address error, power-on reset, or manual reset.

Cancellation by an Interrupt: When an interrupt occurs, the sleep mode is canceled and interrupt exception processing is executed. The sleep mode is not canceled if the interrupt cannot be accepted because its priority level is equal to or less than the mask level set in the CPU's status register (SR) or if an interrupt by an on-chip peripheral module is disabled at the peripheral module.

Cancellation by a DMAC/DTC Address Error: If a DMAC/DTC address error occurs, the sleep mode is canceled and DMAC/DTC address error exception processing is executed.

Cancellation by a Power-On Reset: A power-on reset resulting from setting the $\overline{\text{RES}}$ pin to low level cancels the sleep mode.

Cancellation by a Manual Reset: When the $\overline{\text{MRES}}$ pin is set to low level while the $\overline{\text{RES}}$ pin is at high level, a manual reset occurs and the sleep mode is canceled.

24.4 Standby Mode

24.4.1 Transition to Standby Mode

To enter the standby mode, set the SBY bit to 1 in SBYCR, then execute the SLEEP instruction. The LSI moves from the program execution state to the standby mode. In the standby mode, power consumption is greatly reduced by halting not only the CPU, but the clock and on-chip peripheral modules as well. CPU register contents and on-chip RAM data are held as long as the prescribed voltages are applied. The register contents of some on-chip peripheral modules are initialized, but some are not (table 24.3). The I/O port status can be selected as held or high impedance by the port high impedance bit (HIZ) of the SBYCR. For pin status other than for the I/O port, refer to Appendix C, Pin States.

Table 24.3 Register States in the Standby Mode

Module	Registers Initialized	Registers that Retain Data	Registers with Undefined Contents
Interrupt controller (INTC)	—	All registers	—
User break controller (UBC)	—	All registers	—
Data transfer controller (DTC)	All registers (excluding transfer data in memory and DTDR)	—	—
Cache memory (CAC)	—	All registers	—
Bus state controller (BSC)	—	All registers	—
Direct memory access controller (DMAC)	• DMA channel control registers 0–3 (CHCR0–CHCR3) • DMA operation register (DMAOR)	—	• DMA source address registers 0–3 (SAR0–SAR3) • DMA destination address registers 0–3 (DAR0–DAR3) • DMA transfer count registers 0–3 (DMATCR0–DMATCR3)
Multifunction timer pulse unit (MTU)	MTU associated registers	POE associated registers	—
Watchdog timer (WDT)	• Bits 7–5 (OVF, WT/$\overline{IT}$, TME) of the timer control status register (TCSR) • Reset control/status register (RSTCSR)	• Bits 2–0 (CKS2–CKS0) of the TCSR • Timer counter (TCNT)	—
Serial communication interface (SCI)	• Receive data register (RDR) • Transmit data register (TDR) • Serial mode register (SMR) • Serial control register (SCR) • Serial status register (SSR) • Bit rate register (BBR)	—	—
A/D converter (A/D)	All registers	—	—
Compare match timer (CMT)	All registers	—	—

Table 24.3 Register States in the Standby Mode (cont)

Module	Registers Initialized	Registers that Retain Data	Registers with Undefined Contents
Pin function controller (PFC)	—	All registers	—
I/O port (I/O)	—	All registers	—
Power-down state related	—	Standby control register (SBYCR)	—

24.4.2 Canceling the Standby Mode

The standby mode is canceled by an NMI interrupt, a power-on reset, or a manual reset.

Cancellation by an NMI: Clock oscillation starts when a rising edge or falling edge (selected by the NMI edge select bit (NMIE) of the interrupt control register (ICR) of the INTC) is detected in the NMI signal. This clock is supplied only to the watchdog timer (WDT). A WDT overflow occurs if the time established by the clock select bits (CKS2–CKS0) in the TCSR of the WDT elapses before transition to the standby mode. The occurrence of this overflow is used to indicate that the clock has stabilized, so the clock is supplied to the entire chip, the standby mode is canceled, and NMI exception processing begins.

When canceling standby mode with NMI interrupts, set the CKS2–CKS0 bits so that the WDT overflow period is longer than the oscillation stabilization time.

When canceling standby mode with an NMI pin set for falling edge, be sure that the NMI pin level upon entering standby (when the clock is halted) is high level, and that the NMI pin level upon returning from standby (when the clock starts after oscillation stabilization) is low level. When canceling standby mode with an NMI pin set for rising edge, be sure that the NMI pin level upon entering standby (when the clock is halted) is low level, and that the NMI pin level upon returning from standby (when the clock starts after oscillation stabilization) is high level.

Cancellation by a Power-On Reset: A power-on reset caused by setting the $\overline{\text{RES}}$ pin to low level cancels the standby mode.

24.4.3 Standby Mode Application Example

This example describes a transition to standby mode on the falling edge of an NMI signal, and a cancellation on the rising edge of the NMI signal. The timing is shown in figure 24.1.

When the NMI pin is changed from high to low level while the NMI edge select bit (NMIE) of the ICR is set to 0 (falling edge detection), the NMI interrupt is accepted. When the NMIE bit is set to 1 (rising edge detection) by an NMI exception service routine, the standby bit (SBY) of the SBYCR is set to 1, and a SLEEP instruction is executed, standby mode is entered. Thereafter, standby mode is canceled when the NMI pin is changed from low to high level.

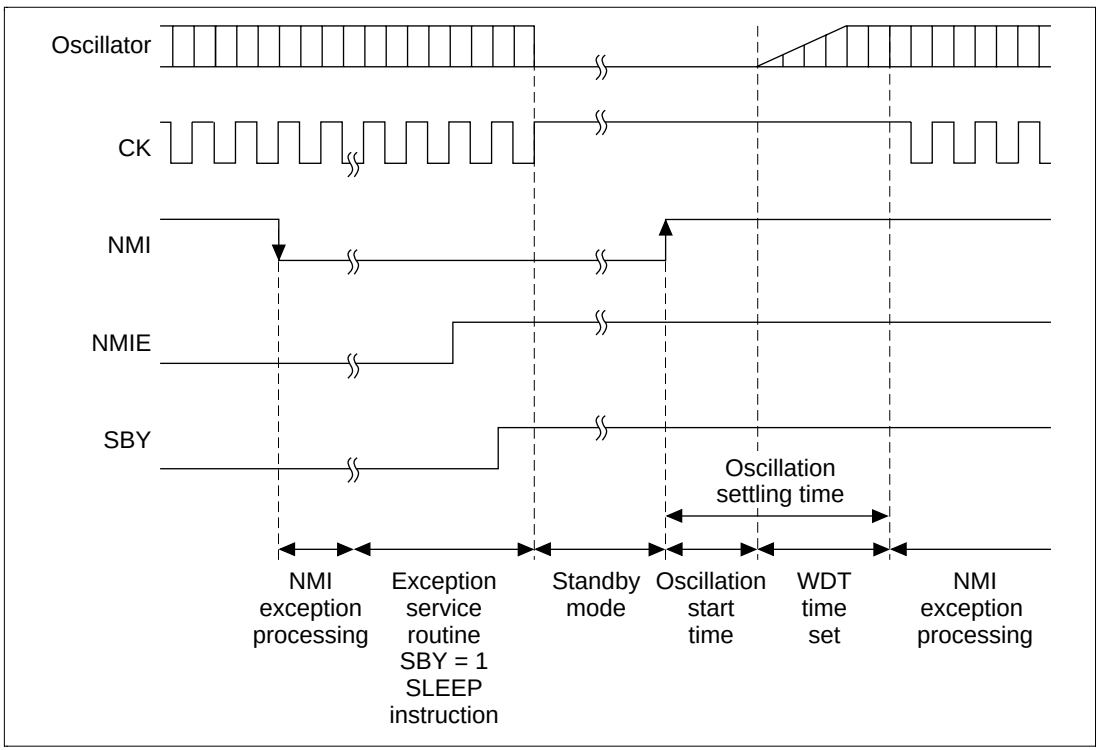


Figure 24.1 Standby Mode NMI Timing (Application Example)

Section 25 Electrical Characteristics (5V, 28.7 MHz Version)

25.1 Absolute Maximum Ratings

Table 25.1 shows the absolute maximum ratings.

Table 25.1 Absolute Maximum Ratings

Item	Symbol	Rating	Unit
Power supply voltage	V_{CC}	–0.3 to +7.0	V
Programmable voltage (ZTAT version only)	V_{PP}	–0.3 to +13.5	V
Input voltage (other than A/D ports)	V_{in}	–0.3 to V_{CC} + 0.3	V
Input voltage (A/D ports)	V_{in}	–0.3 to AV_{CC} + 0.3	V
Analog supply voltage	AV_{CC}	–0.3 to +7.0	V
Analog reference voltage (QFP-144 only)	AV_{ref}	–0.3 to AV_{CC} + 0.3	V
Analog input voltage	V_{AN}	–0.3 to AV_{CC} + 0.3	V
Operating temperature	T_{opr}	–20 to +75*1	°C
Programming temperature (ZTAT version only)	T_{we}	–20 to +75*2	°C
Storage temperature	T_{stg}	–55 to +125	°C

Notes: Operating the LSI in excess of the absolute maximum ratings may result in permanent damage.

*1 Normal Products : T_{OPR} = –40 to + 85°C for wide-temperature range products.

*2 Normal Products: T_{we} = –20 to +85°C for wide-temperature range products.

25.2 DC Characteristics

Table 25.2 DC Characteristics (Conditions: V_{CC} = 5.0 V ± 10%, AV_{CC} = 5.0 V ± 10%, AV_{CC} = V_{CC} ± 10%, AV_{ref} = 4.5 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, Ta = –20 to +75° C)

Item	Pin	Symbol	Min	Typ	Max	Unit	Measurement Conditions
Input high-level voltage	$\overline{RES}$, NMI, MD3–MD0, PA2, PA5, PA6–PA9, PE0–PE15, FWP	V_{IH}	V_{CC} – 0.7	—	V_{CC} + 0.3	V	—
	EXTAL		V_{CC} × 0.7	—	V_{CC} + 0.3	V	—
	A/D port		2.2	—	AV_{CC} + 0.3	V	—
	Other input pins		2.2	—	V_{CC} + 0.3	V	—
Input low-level voltage	$\overline{RES}$, NMI, MD3–MD0, PA2, PA5, PA6–PA9, PE0–PE15, FWP	V_{IL}	–0.3	—	0.5	V	—
	Other input pins		–0.3	—	0.8	V	—
Schmitt trigger input voltage	PA2, PA5, PA6–PA9, PE0–PE15	$VT^+ – VT^-$	0.4	—	—	V	$VT^+ \geq V_{CC}$ – 0.7 V (min)
							$VT^- \leq$ 0.5 V (max)
Input leak current	$\overline{RES}$, NMI, MD3–MD0, PA2, PA5, PA6–PA9, PE0–PE15,FWP	| Iin |	—	—	1.0	µA	Vin = 0.5 to V_{CC} – 0.5 V
	A/D port		—	—	1.0	µA	Vin = 0.5 to AV_{CC} – 0.5 V
	Other input pins (except EXTAL pin)		—	—	1.0	µA	Vin = 0.5 to V_{CC} – 0.5 V
Three-state leak current (while off)	A21–A0, D31–D0, $\overline{CS3}$–$\overline{CS0}$, RDWR, $\overline{RAS}$, $\overline{CASxx}$, $\overline{WRxx}$, $\overline{RD}$, ports A, B, C, D, E	| I_{TSI} |	—	—	1.0	µA	Vin = 0.5 to V_{CC} – 0.5 V

Table 25.2 DC Characteristics (Conditions: V_{CC} = 5.0 V ± 10%, AV_{CC} = 5.0 V ± 10%, AV_{CC} = V_{CC} ± 10%, AV_{ref} = 4.5 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, Ta = –20 to +75°C) (cont)

Item	Pin	Symbol	Min	Typ	Max	Unit	Measurement Conditions
Output high-level voltage	All output pins	V_{OH}	V_{CC} – 0.5	—	—	V	I_{OH} = –200 µA
			3.5	—	—	V	I_{OH} = –1 mA
Output low-level voltage	All output pins	V_{OL}	—	—	0.4	V	I_{OL} = 1.6 mA
	PE9, PE11–PE15		—	—	1.5	V	I_{OL} = 15 mA
Input capacitance	$\overline{RES}$	Cin	—	—	80*1	pF	Vin = 0 V, f = 1 MHz, Ta = 25°C
	NMI		—	—	50	pF	
	All other input pins		—	—	20	pF	
Current consumption	Ordinary operation	I_{CC}	—	160	230	mA	f = 28 MHz
	Sleep		—	140	200	mA	f = 28 MHz
	Standby		—	0.01	5	µA	Ta ≤ 50°C
			—	—	20	µA	Ta > 50°C
Analog supply current		AI_{CC}	—	5	10	mA	
		AI_{ref}	—	0.5	1*2	mA	QFP144 version only
RAM standby voltage		V_{RAM}	2.0	—	—	V	

Notes:
1. When the A/D converter is not used (including during standby), do not release the AV_{CC}, AV_{SS}, and AV_{ref} (SH7041,SH7043,SH7045 only) pins. Connect the AV_{CC} and AV_{ref} (SH7041,SH7043,SH7045 only) pins to V_{CC} and the AV_{SS} pin to V_{SS}.
2. The current consumption is measured when V_{IH}min = V_{CC} – 0.5 V, V_{IL} max = 0.5 V, with all output pins unloaded.
3. The ZTAT and mask versions as well as F-ZTAT and mask versions have the same functions, and the electrical characteristics of both are within specification, but characteristic-related performance values, operating margins, noise margins, noise emission, etc., are different. Caution is therefore required in carrying out system design, when switching between ZTAT and mask versions, and when switching between F-ZTAT and mask versions.
4. When the SH7040 chip is used for high-speed operation, the package surface temperature rises. Appropriate measures (such as heat dissipation) to ensure overall system reliability and safety should therefore be investigated.

*1 110pF for A mask

*2 5 mA in the A mask version, except for F-ZTAT products.

Table 25.3 Permitted Output Current Values (Conditions: V_{CC} = 5.0 V ± 10%, AV_{CC} = 5.0 V ± 10%, AV_{CC} = V_{CC} ± 10%, AV_{ref} = 4.5 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, Ta = –20 to +75° C)

Item	Symbol	Min	Typ	Max	Unit
Output low-level permissible current (per pin)	I_{OL}	—	—	2.0*	mA
Output low-level permissible current (total)	$\sum I_{OL}$	—	—	80	mA
Output high-level permissible current (per pin)	$-I_{OH}$	—	—	2.0	mA
Output high-level permissible current (total)	$\sum (-I_{OH})$	—	—	25	mA

Notes: To assure LSI reliability, do not exceed the output values listed in this table.

* PE9, PE11–PE15 are I_{OL} = 15 mA (max). Make sure that no more than three of these pins exceed an I_{OL} of 2.0 mA simultaneously.

25.3 AC Characteristics

25.3.1 Clock Timing

Table 25.4 Clock Timing (Conditions: V_{CC} = 5.0 V ± 10%, AV_{CC} = 5.0 V ± 10%, AV_{CC} = V_{CC} ± 10%, AV_{ref} = 4.5 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, Ta = –20 to +75° C)

Item	Symbol	Min	Max	Unit	Figures
Operating frequency	f_{OP}	4	28.7	MHz	25.1
Clock cycle time	t_{cyc}	34.8	250	ns	
Clock low-level pulse width	t_{CL}	10	—	ns	
Clock high-level pulse width	t_{CH}	10	—	ns	
Clock rise time	t_{CR}	—	5	ns	
Clock fall time	t_{CF}	—	5	ns	
EXTAL clock input frequency	f_{EX}	4	10	MHz	25.2
EXTAL clock input cycle time	t_{EXcyc}	100	250	ns	
EXTAL clock low-level input pulse width	t_{EXL}	40	—	ns	
EXTAL clock high-level input pulse width	t_{EXH}	40	—	ns	
EXTAL clock input rise time	t_{EXR}	—	5	ns	
EXTAL clock input fall time	t_{EXF}	—	5	ns	
Reset oscillation settling time	t_{OSC1}	10	—	ms	25.3
Standby return clock settling time	t_{OSC2}	10	—	ms	

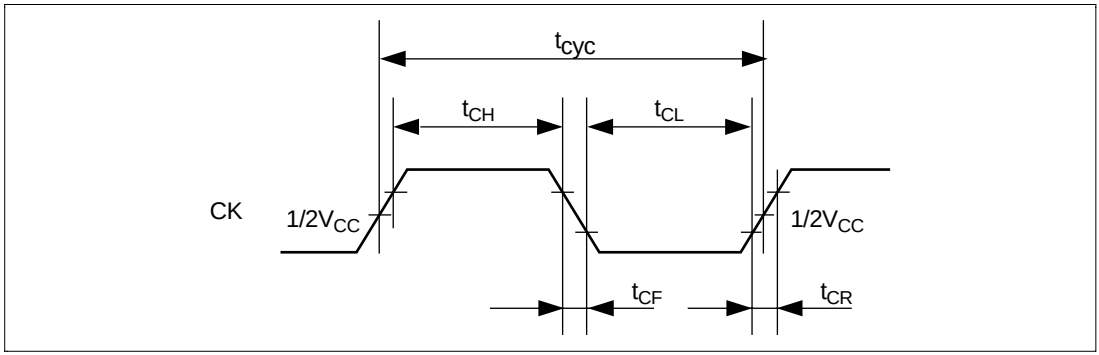


Figure 25.1 System Clock Timing

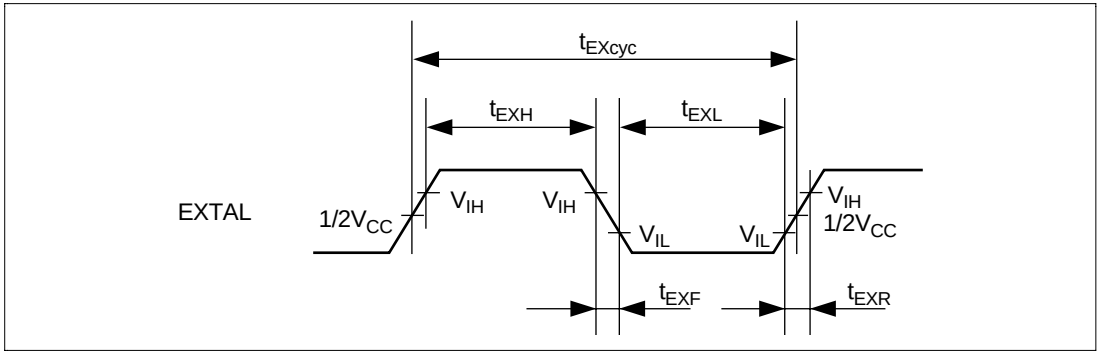


Figure 25.2 EXTAL Clock Input Timing

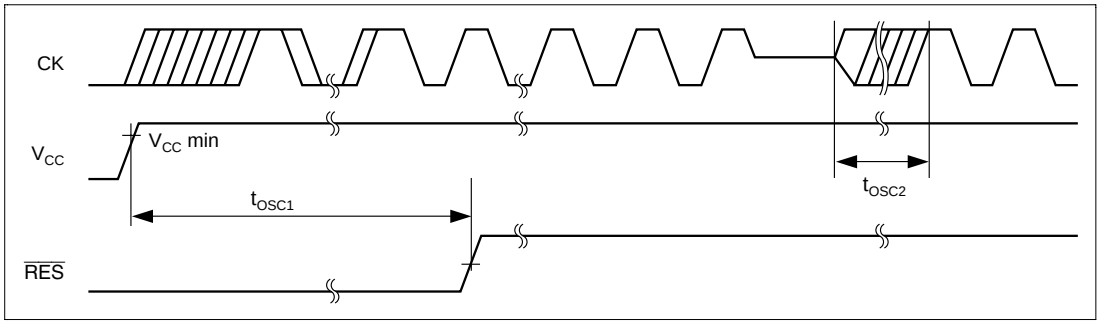


Figure 25.3 Oscillation Settling Time

25.3.2 Control Signal Timing

Table 25.5 Control Signal Timing (Conditions: V_{CC} = 5.0 V ± 10%, AV_{CC} = 5.0 V ± 10%, AV_{CC} = V_{CC} ± 10%, AV_{ref} = 4.5 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, Ta = –20 to +75° C)

Item	Symbol	Min	Max	Unit	Figure
$\overline{RES}$ rise/fall	t_{RESr}, t_{RESf}	—	200	ns	25.4
$\overline{RES}$ pulse width	t_{RESW}	20	—	t_{cyc}	
$\overline{MRES}$ pulse width	t_{MRESW}	20	—	t_{cyc}	
NMI rise/fall	t_{NMIr}, t_{NMIf}	—	200	ns	25.5
$\overline{RES}$ setup time*	t_{RESS}	35	—	ns	25.4, 25.5
$\overline{MRES}$ setup time*	t_{MRESS}	35	—	ns	
NMI setup time*	t_{NMIS}	35	—	ns	
$\overline{IRQ7}$–$\overline{IRQ0}$ setup time (edge detection)	t_{IRQES}	35	—	ns	
$\overline{IRQ7}$–$\overline{IRQ0}$ setup time (level detection)	t_{IRQLS}	35	—	ns	
NMI hold time	t_{NMIH}	35	—	ns	25.5
$\overline{IRQ7}$–$\overline{IRQ0}$ hold time	t_{IRQEH}	35	—	ns	
$\overline{IRQOUT}$ output delay time	t_{IRQOD}	—	35	ns	25.6
Bus request setup time	t_{BRQS}	35	—	ns	25.7
Bus acknowledge delay time 1	t_{BACKD1}	—	35	ns	
Bus acknowledge delay time 2	t_{BACKD2}	—	35	ns	
Bus three-state delay time	t_{BZD}	—	35	ns	

Note: * The $\overline{RES}$, $\overline{MRES}$, NMI, $\overline{BREQ}$, and $\overline{IRQ7}$–$\overline{IRQ0}$ signals are asynchronous inputs, but when thesetup times shown here are provided, the signals are considered to have produced changes at clock rise (for $\overline{RES}$, $\overline{MRES}$, $\overline{BREQ}$) or clock fall (for NMI and $\overline{IRQ7}$–$\overline{IRQ0}$). If the setup times are not provided, recognition is delayed until the next clock rise or fall.

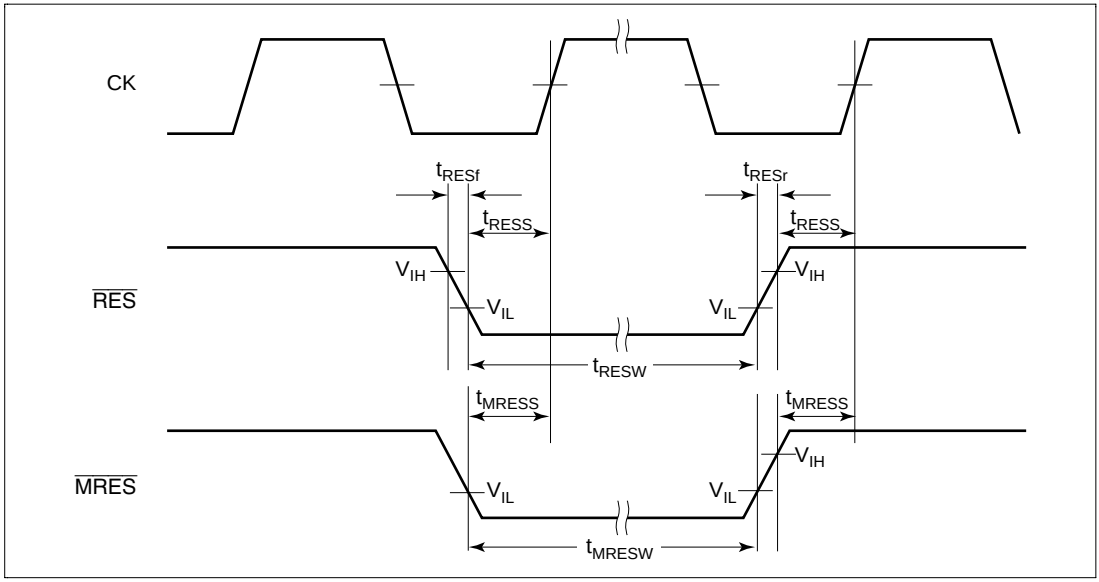


Figure 25.4 Reset Input Timing

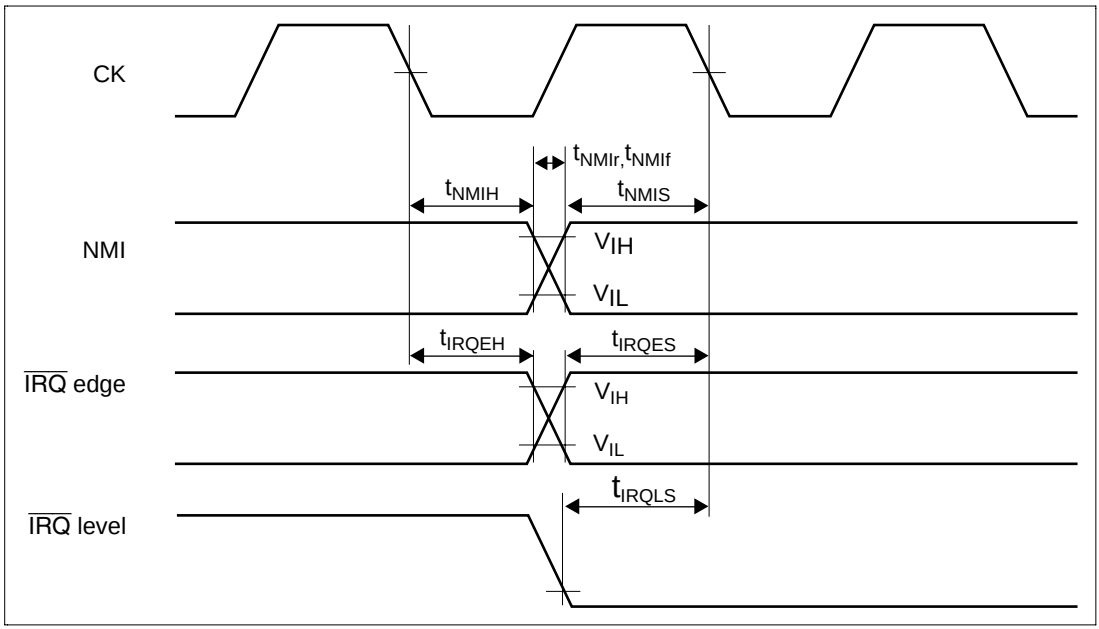


Figure 25.5 Interrupt Signal Input Timing

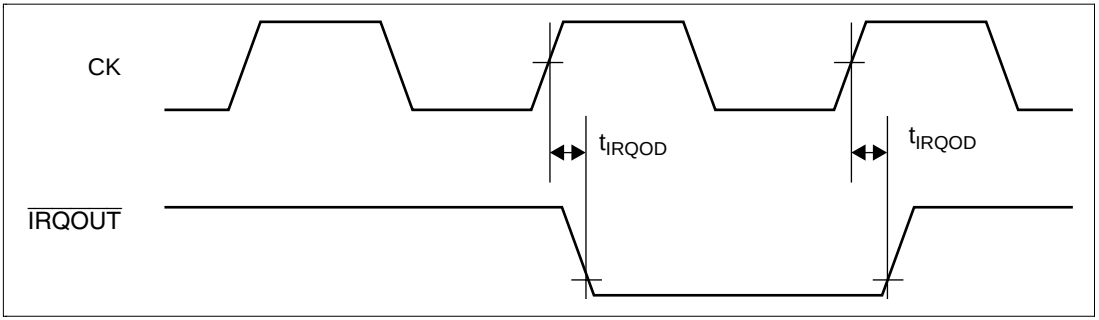
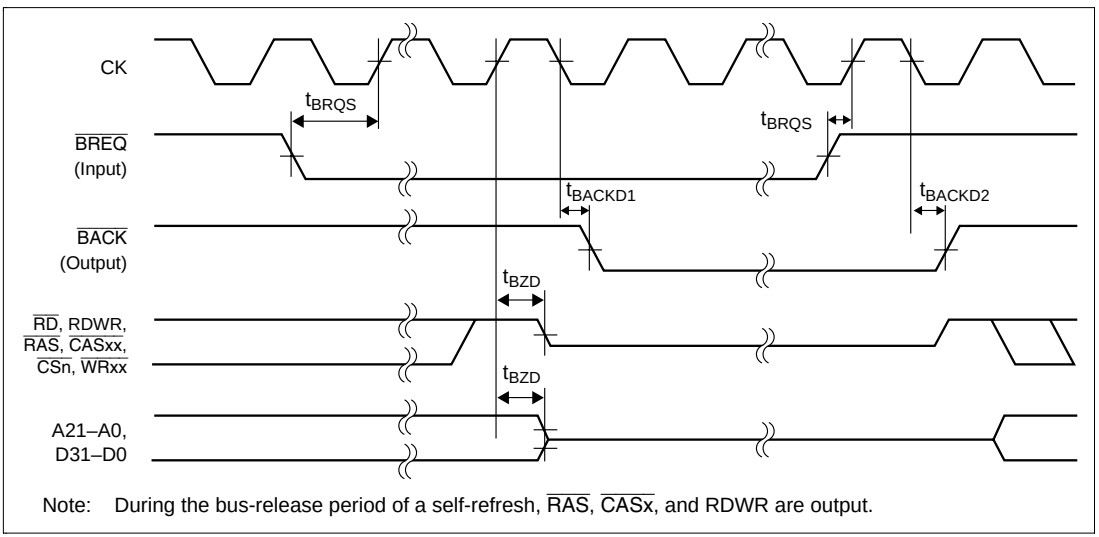


Figure 25.6 Interrupt Signal Output Timing



Note: During the bus-release period of a self-refresh, $\overline{RAS}$, $\overline{CASx}$, and RDWR are output.

Figure 25.7 Bus Right Release Timing

25.3.3 Bus Timing

Table 25.6 Bus Timing (Conditions: V_{CC} = 5.0 V ± 10%, AV_{CC} = 5.0 V ± 10%, AV_{CC} = V_{CC} ± 10%, AV_{ref} = 4.5 V – AV_{CC}, V_{SS} = AV_{SS} = 0 V, Ta = – 20 to +75°C)

Item	Symbol	Min	Max	Unit	Figure
Address delay time	t_{AD}	2*3	18	ns	25.8, 25.9, 25.11–25.16, 25.19
$\overline{CS}$ delay time 1	t_{CSD1}	2*3	21	ns	25.8, 25.9, 25.19
$\overline{CS}$ delay time 2	t_{CSD2}	2*3	21	ns	
Read strobe delay time 1	t_{RSD1}	2*3	18	ns	25.8, 25.9, 25.11–25.16, 25.19
Read strobe delay time 2	t_{RSD2}	2*3	18	ns	
Read data setup time	t_{RDS}*4	15	—	ns	
Read data hold time	t_{RDH}	0	—	ns	
Write strobe delay time 1	t_{WSD1}	2*3	18	ns	
Write strobe delay time 2	t_{WSD2}	2*3	18	ns	
Write data delay time	t_{WDD}	—	35	ns	
Write data hold time	t_{WDH}	0	10*2	ns	
$\overline{WAIT}$ setup time	t_{WTS}	15	—	ns	25.10, 25.15, 25.19
$\overline{WAIT}$ hold time	t_{WTH}	0	—	ns	
$\overline{RAS}$ delay time 1	t_{RASD1}	2*3	18	ns	25.11–25.18
$\overline{RAS}$ delay time 2	t_{RASD2}	2*3	18	ns	
$\overline{CAS}$ delay time 1	t_{CASD1}	2*3	18	ns	
$\overline{CAS}$ delay time 2	t_{CASD2}	2*3	18	ns	
Read data access time	t_{ACC}*1	$t_{cyc} \times (n + 2) - 40$	—	ns	25.8, 25.9
Access time from read strobe	t_{OE}*1	$t_{cyc} \times (n + 1.5) - 40$	—	ns	
Access time from column address	t_{AA}*1	$t_{cyc} \times (n + 2) - 40$	—	ns	25.11–25.16
Access time from $\overline{RAS}$	t_{RAC}*1	$t_{cyc} \times (n + RCD + 2.5) - 40$	—	ns	
Access time from $\overline{CAS}$	t_{CAC}*1	$t_{cyc} \times (n + 1) - 40$	—	ns	
Row address hold time	t_{RAH}	$t_{cyc} \times (RCD + 0.5) - 15$	—	ns	
Row address setup time	t_{ASR}*5	$t_{cyc} \times 0.5–17.5$	—	ns	
Data input setup time	t_{DS}	$t_{cyc} \times (m + 0.5) - 25$	—	ns	
Data input hold time	t_{DH}	20	—	ns	

Table 25.7 Bus Timing (Conditions: V_{CC} = 5.0 V ± 10%, AV_{CC} = 5.0 V ± 10%, AV_{CC} = V_{CC} ± 10%, AV_{ref} = 4.5 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, Ta = – 20 to +75°C)

Item	Symbol	Min	Max	Unit	Figure
Write address setup time	t_{AS}	0	—	ns	25.8–25.9
Write address hold time	t_{WR}	5	—	ns	25.8, 25.9, 25.19
Write data hold time	t_{WRH}	0	—	ns	
Read/write strobe delay time 1	t_{RWD1}	2*3	18	ns	25.11–25.16
Read/write strobe delay time 2	t_{RWD2}	2*3	18	ns	
High-speed page mode CAS precharge time	t_{CP}	t_{cyc} – 25	—	ns	25.16
RAS precharge time	t_{RP}	$t_{cyc} \times (TPC + 1.5) - 15$	—	ns	25.11–25.16
CAS setup time	t_{CSR}	10	—	ns	25.17, 25.18
AH delay time 1	t_{AHD1}	2*3	18	ns	25.19
AH delay time 2	t_{AHD2}	2*3	18	ns	
Multiplex address delay time	t_{MAD}	2*3	18	ns	
Multiplex address hold time	t_{MAH}	0	—	ns	
DACK delay time	t_{DACKD1}	2*3	21	ns	25.8, 25.9, 25.11–25.16, 25.19

Notes: n is the number of waits. m is 0 when the number of DRAM write cycle waits is 0, and 1 otherwise. RCD is the set value of the RCD bit in DCR. TPC is the set value of the TPC bit in DCR.

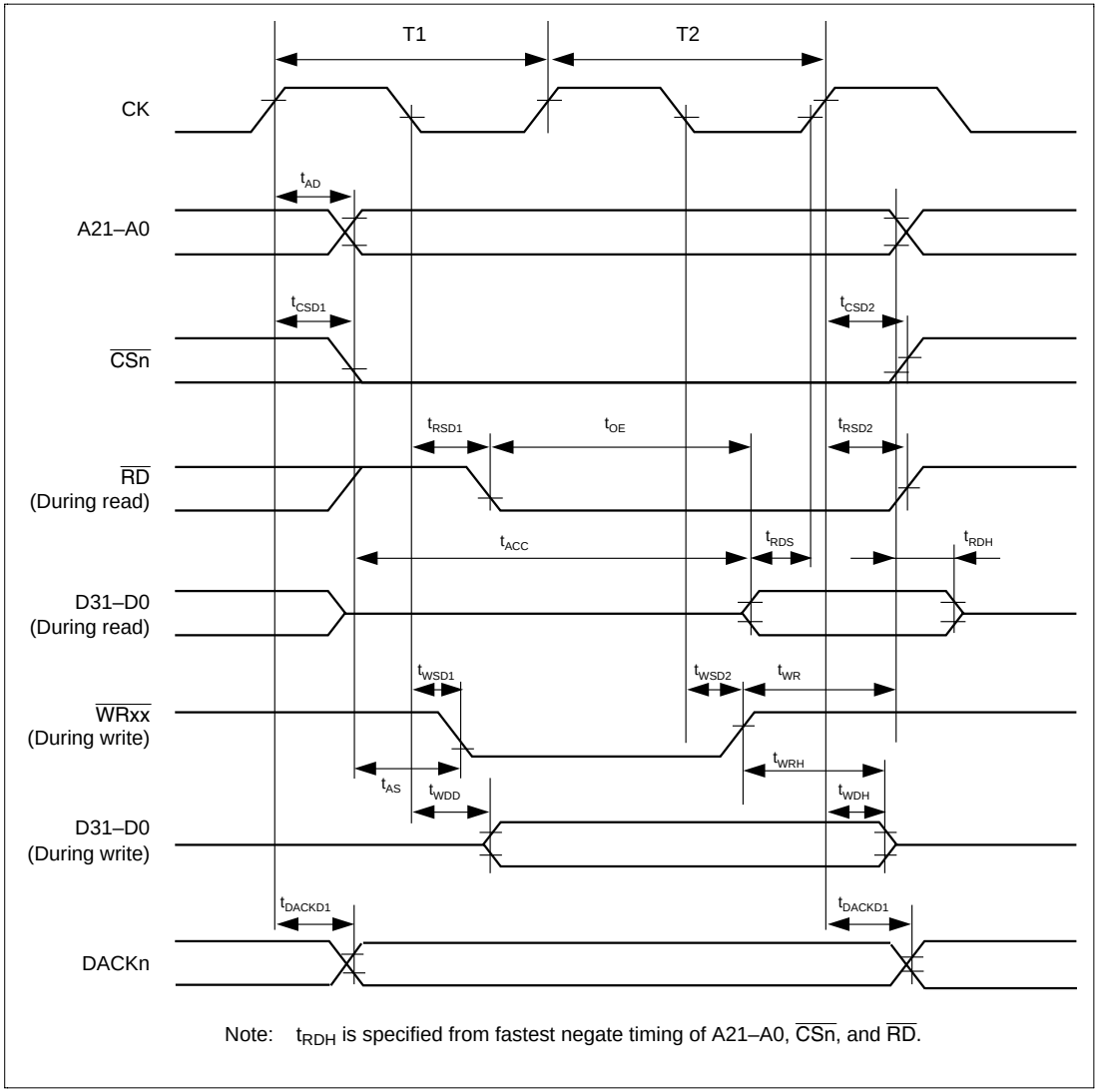
*1 If the access time is satisfied, t_{RDS} need not be satisfied.

*2 t_{WDH} (max) is a reference value.

*3 The delay time Min values are reference values (typ).

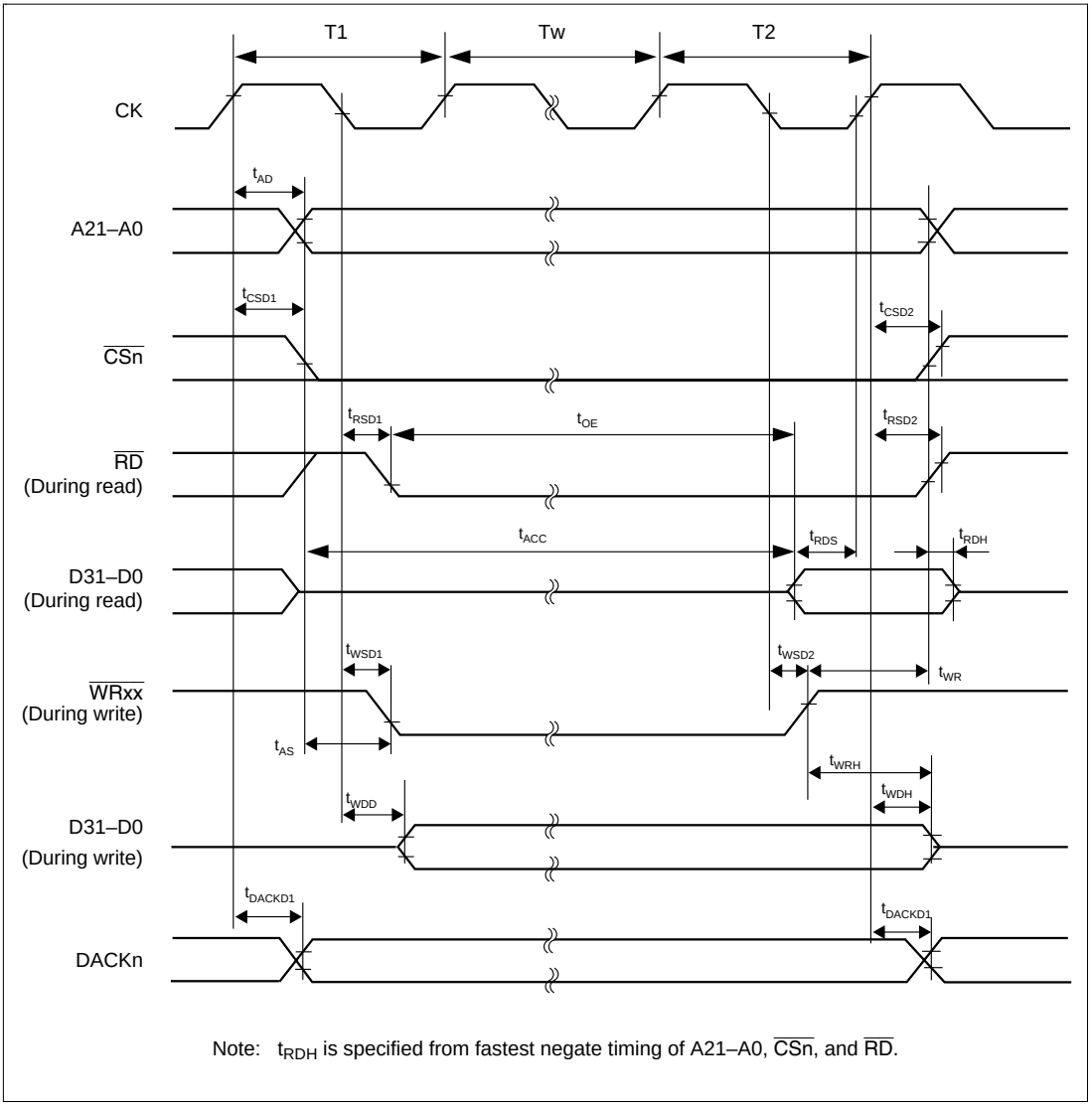
*4 t_{RDS} is a reference value.

*5 When 28.7MHz, tASR=0ns (min)



Note: t_{RDH} is specified from fastest negate timing of A21–A0, $\overline{CSn}$, and $\overline{RD}$.

Figure 25.8 Basic Cycle (No Waits)



Note: t_{RDH} is specified from fastest negate timing of A21–A0, $\overline{CSn}$, and $\overline{RD}$.

Figure 25.9 Basic Cycle (Software Waits)

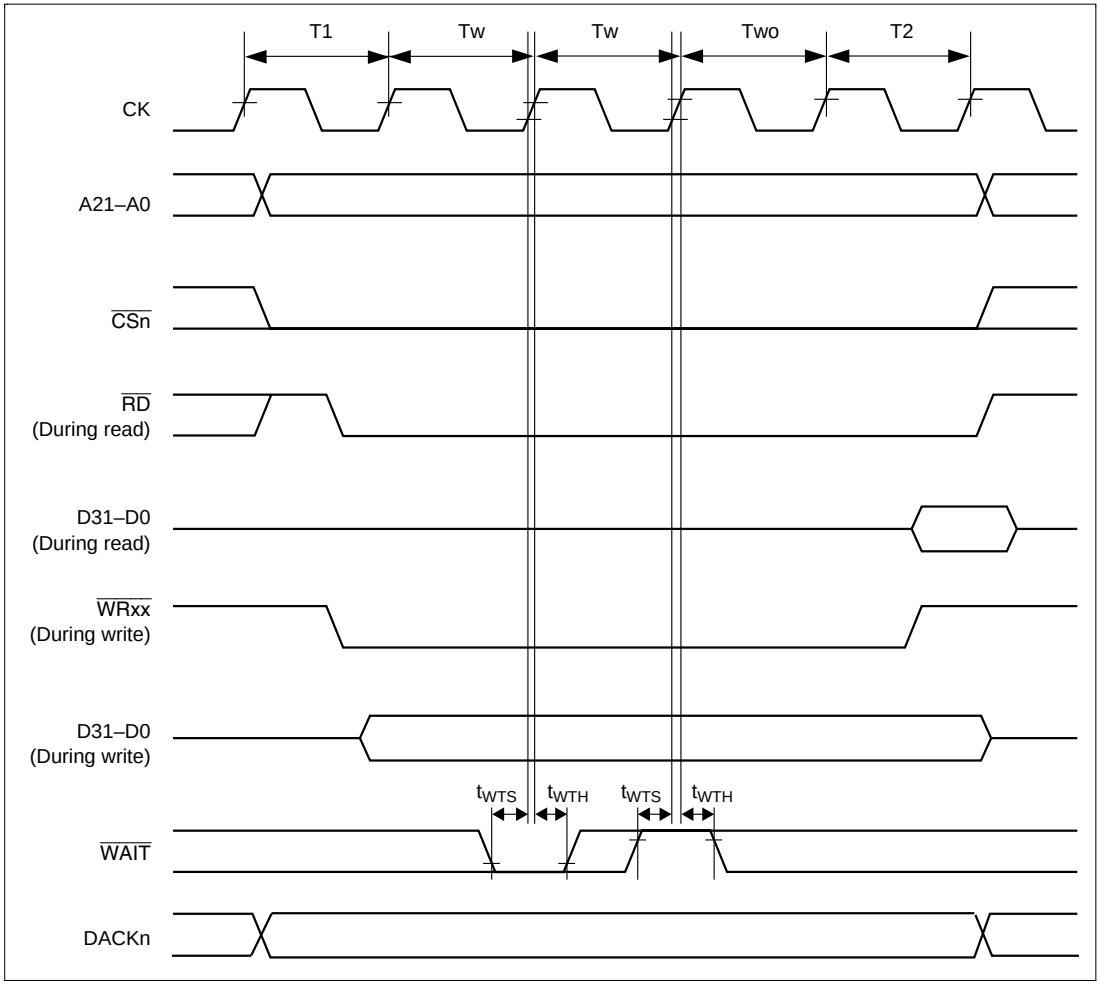
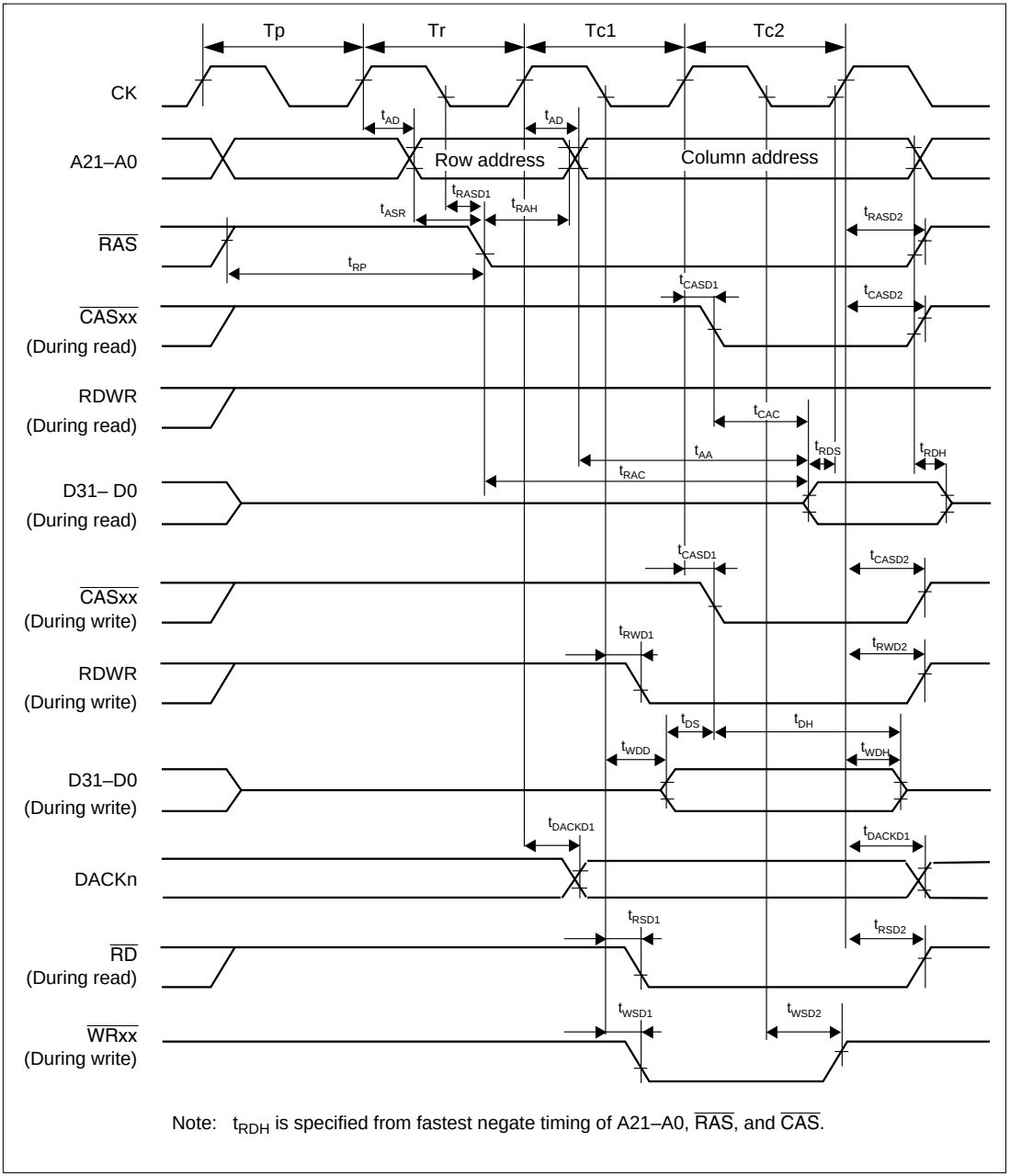
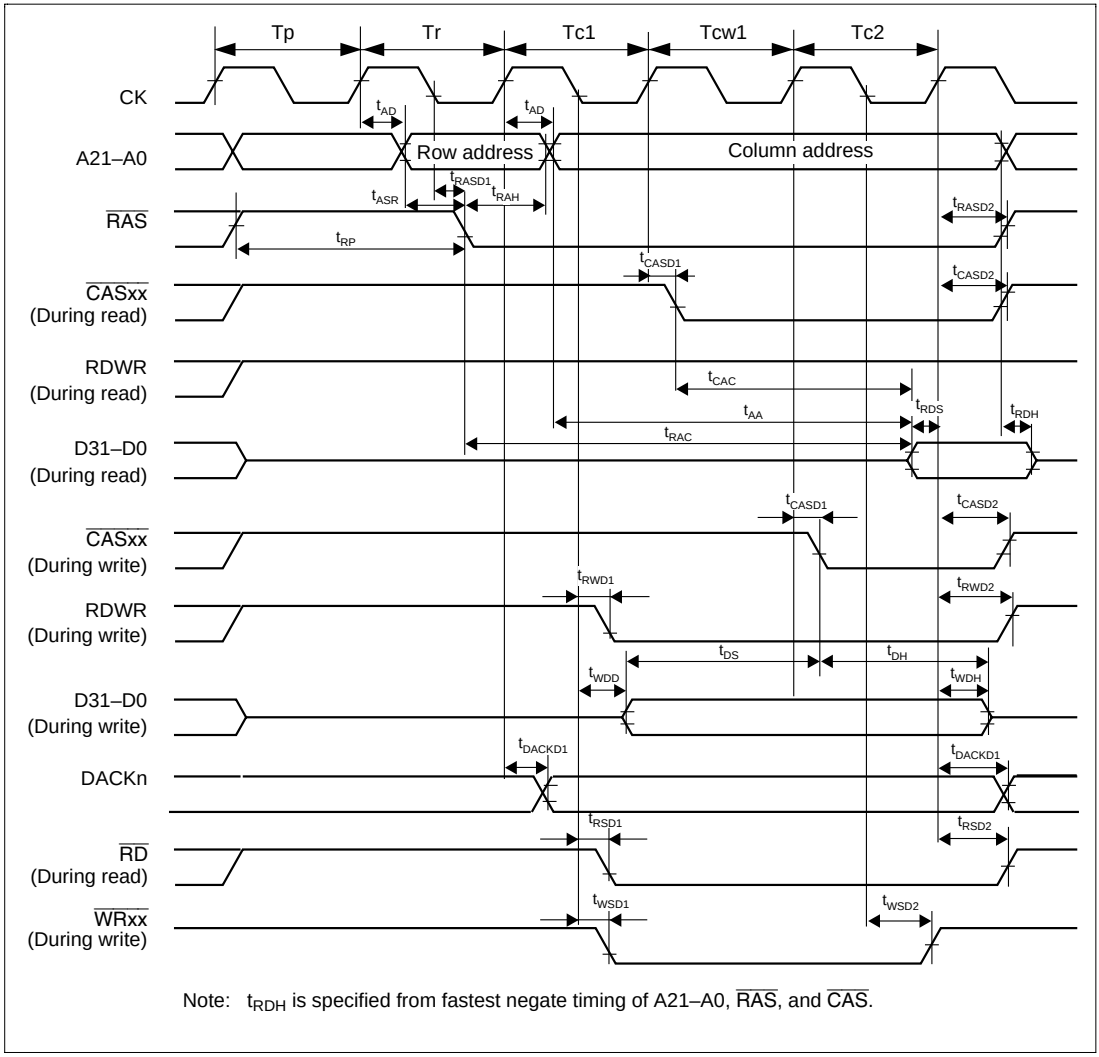


Figure 25.10 Basic Cycle (2 Software Waits + Wait due to $\overline{\text{WAIT}}$ Signal)



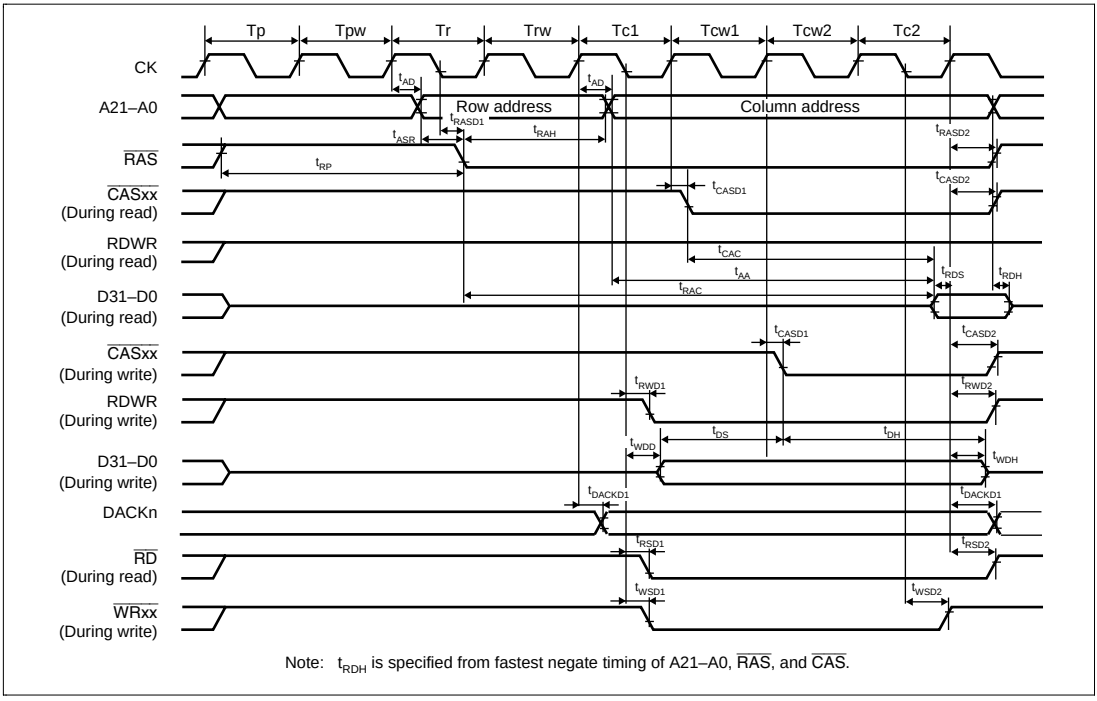
Note: t_{RDH} is specified from fastest negate timing of A21–A0, $\overline{RAS}$, and $\overline{CAS}$.

Figure 25.11 DRAM Cycle (Normal Mode, No Waits, TPC = 0, RCD = 0)



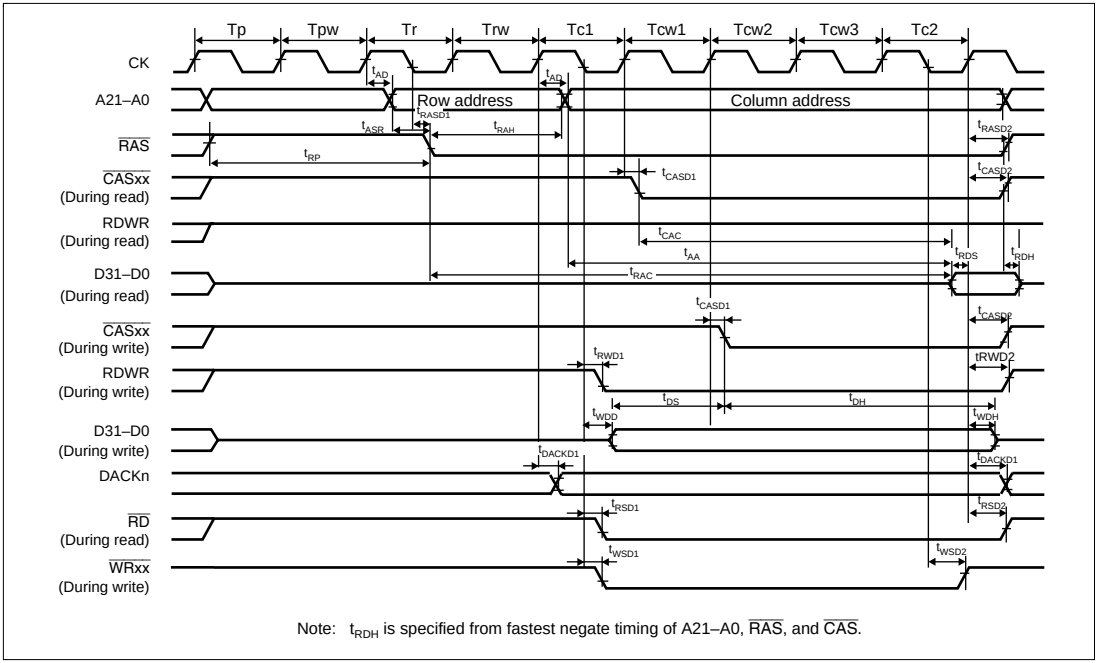
Note: t_{RDH} is specified from fastest negate timing of A21–A0, $\overline{RAS}$, and $\overline{CAS}$.

Figure 25.12 DRAM Cycle (Normal Mode, 1 Wait, TPC = 0, RCD = 0)



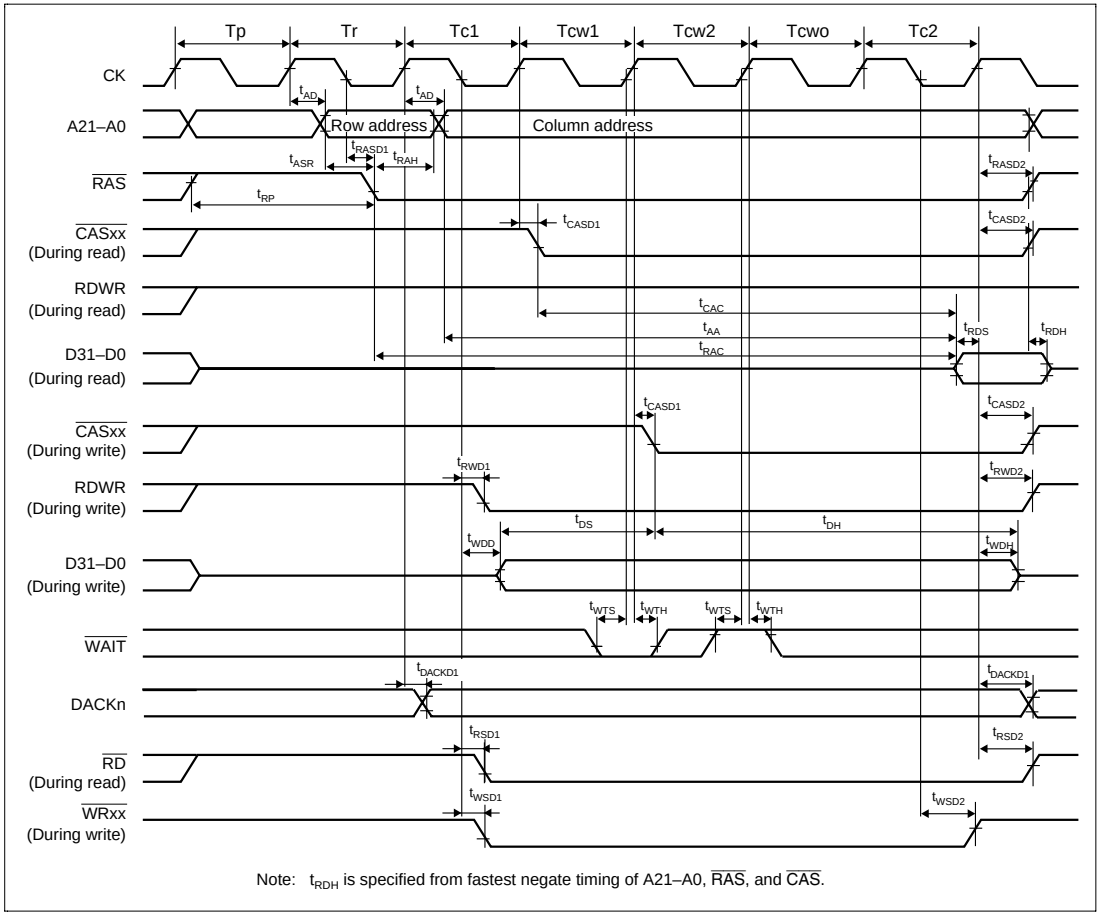
Note: t_{RDH} is specified from fastest negate timing of A21–A0, $\overline{RAS}$, and $\overline{CAS}$.

Figure 25.13 DRAM Cycle (Normal Mode, 2 Waits, TPC = 1, RCD = 1)



Note: t_{RDH} is specified from fastest negate timing of A21–A0, $\overline{RAS}$, and $\overline{CAS}$.

Figure 25.14 DRAM Cycle (Normal Mode, 3 Waits, TPC = 1, RCD = 1)



Note: t_{RDH} is specified from fastest negate timing of A21–A0, $\overline{RAS}$, and $\overline{CAS}$.

Figure 25.15 DRAM Cycle (Normal Mode, 2 Waits + Wait due to $\overline{WAIT}$ Signal)

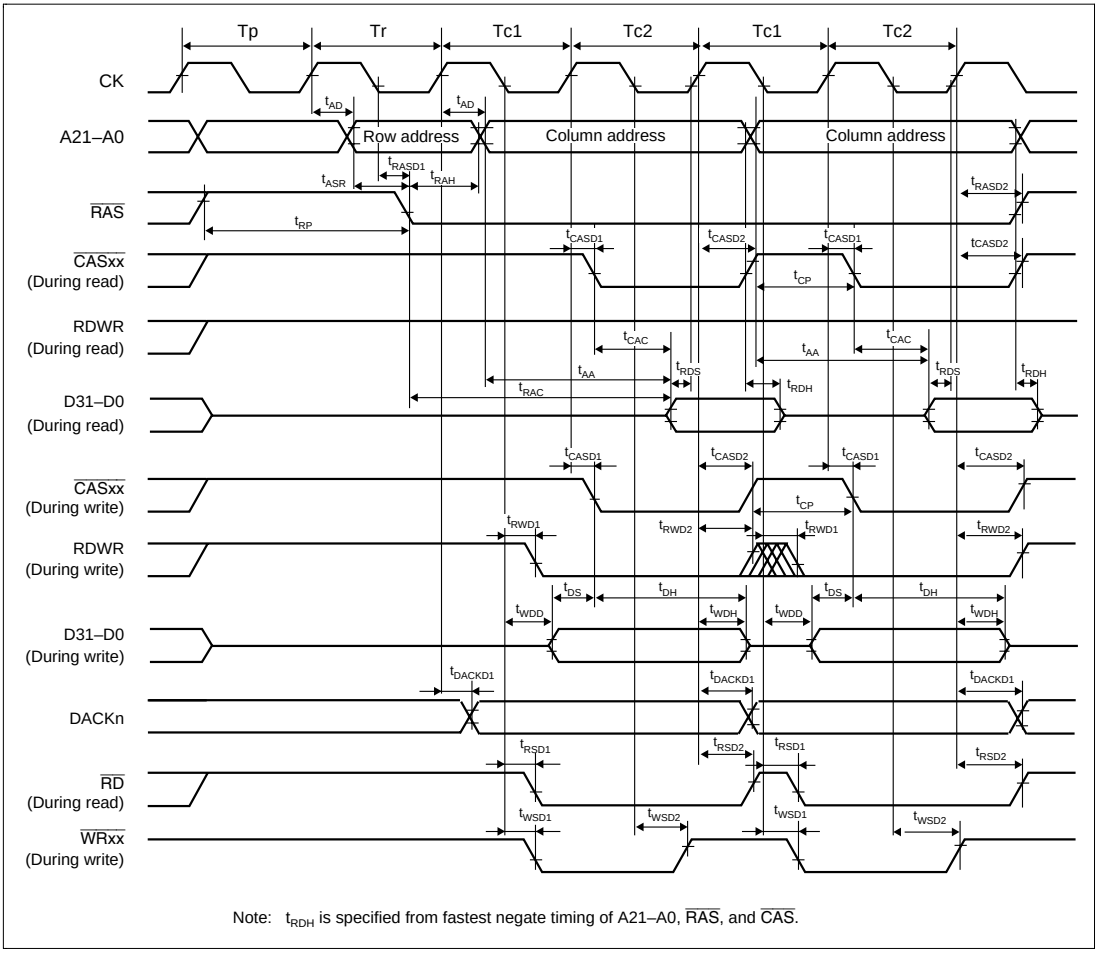


Figure 25.16 DRAM Cycle (High-Speed Page Mode)

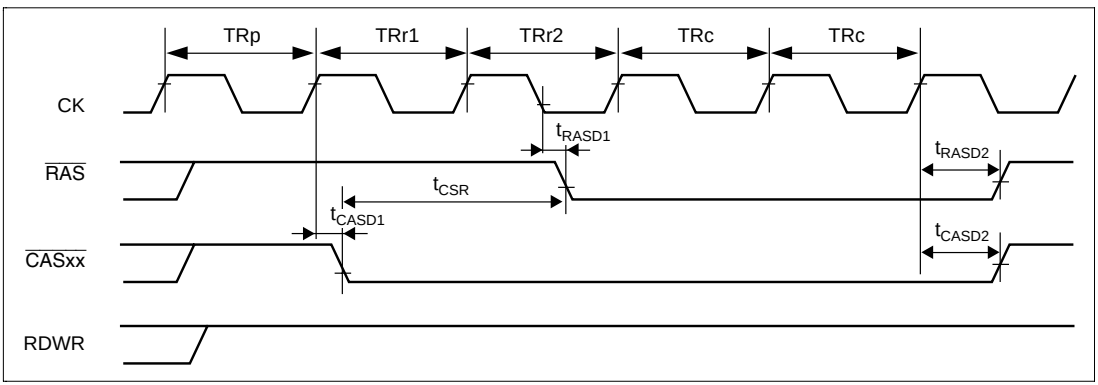


Figure 25.17 CAS Before RAS Refresh (TRAS1 = 0, TRAS0 = 0)

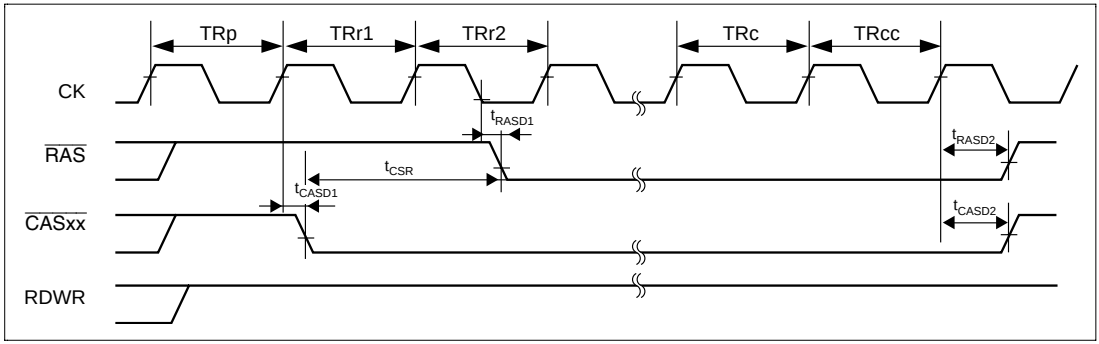


Figure 25.18 Self Refresh

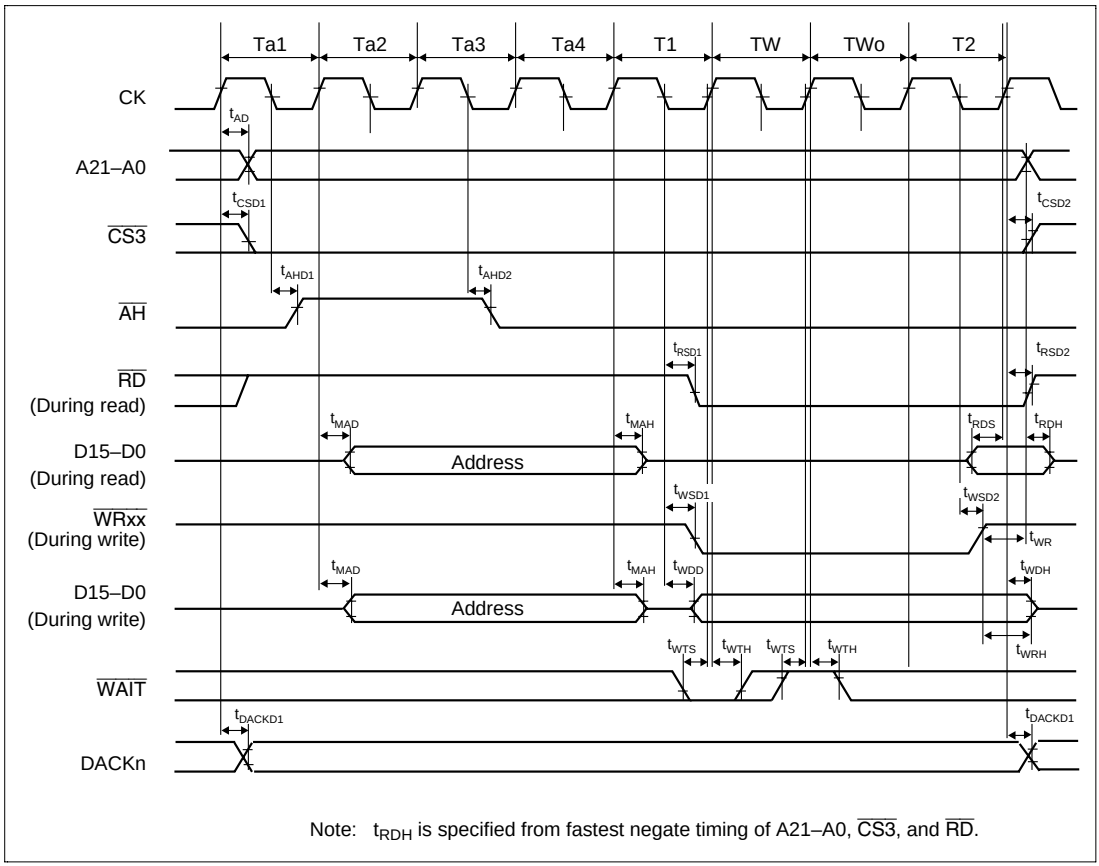


Figure 25.19 Address Data Multiplex I/O Space Cycle (1 Software Wait + External Wait)

25.3.4 Direct Memory Access Controller Timing

Table 25.8 Direct Memory Access Controller Timing (Conditions: V_{CC} = 5.0 V ± 10%, AV_{CC} = 5.0 V ± 10%, AV_{CC} = V_{CC} ± 10%, AV_{ref} = 4.5 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, Ta = – 20 to +75°C)

Item	Symbol	Min	Max	Unit	Figure
$\overline{\text{DREQ0}}$ and $\overline{\text{DREQ1}}$ setup time	t_{DRQS}	18	—	ns	25.20
$\overline{\text{DREQ0}}$ and $\overline{\text{DREQ1}}$ hold time	t_{DRQH}	18	—	ns	
$\overline{\text{DREQ0}}$ and $\overline{\text{DREQ1}}$ pulse width	t_{DRQW}	1.5	—	t_{cyc}	25.21
DRAK output delay time	t_{DRAKD}	—	18	ns	25.22

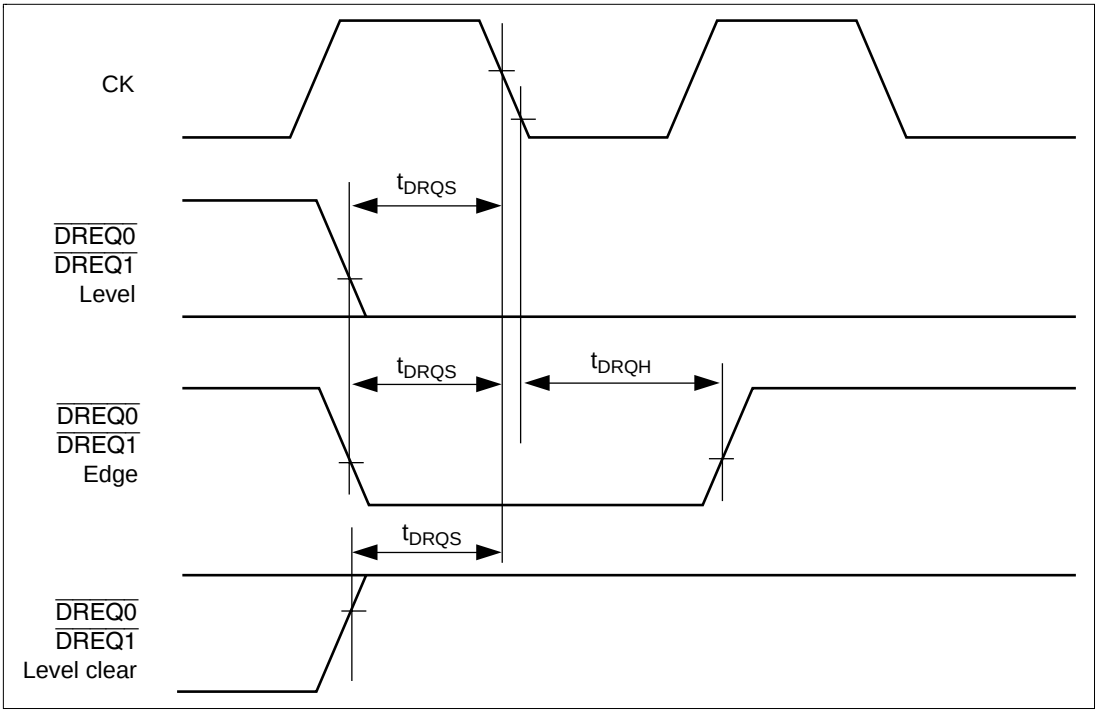


Figure 25.20 $\overline{\text{DREQ0}}$ and $\overline{\text{DREQ1}}$ Input Timing (1)

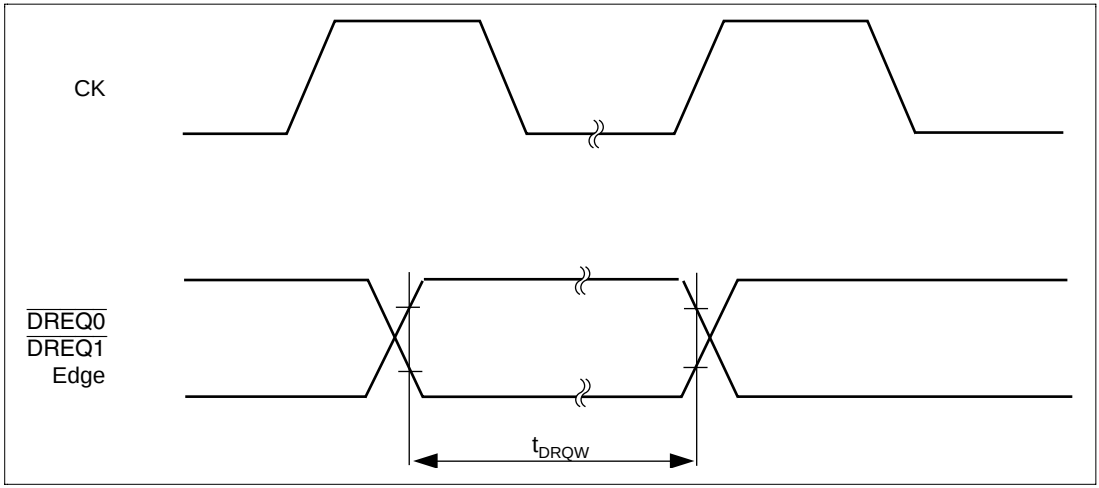


Figure 25.21 $\overline{\text{DREQ0}}$ and $\overline{\text{DREQ1}}$ Input Timing (2)

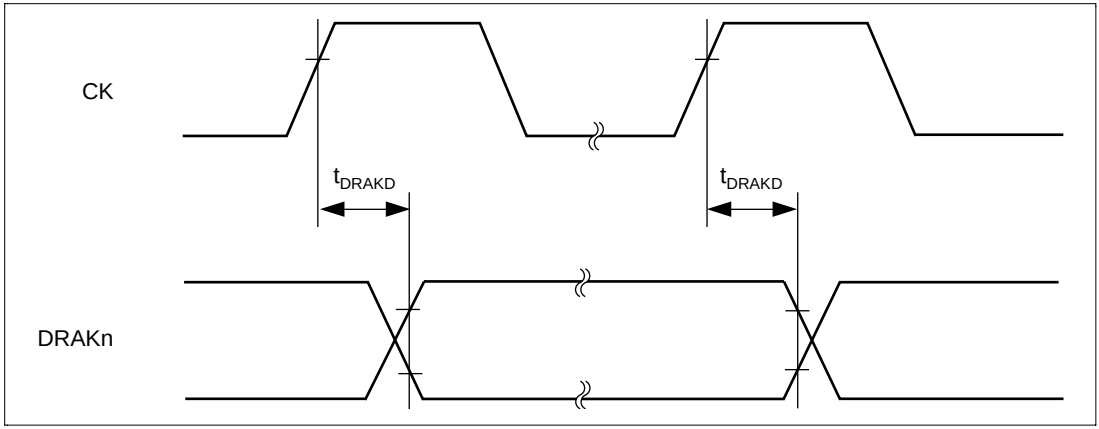


Figure 25.22 DRAK Output Delay Time

25.3.5 Multifunction Timer Pulse Unit Timing

Table 25.9 Multifunction Timer Pulse Unit Timing (Conditions: V_{CC} = 5.0 V ± 10%, AV_{CC} = 5.0 V ± 10%, AV_{CC} = V_{CC} ± 10%, AV_{ref} = 4.5 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, Ta = – 20 to +75°C)

Item	Symbol	Min	Max	Unit	Figure
Output compare output delay time	t_{TOCD}	—	100	ns	25.23
Input capture input setup time	t_{TICS}	30	—	ns	
Timer input setup time	t_{TCKS}	35	—	ns	
Timer clock pulse width (single edge specification)	$t_{TCKWH/L}$	1.5	—	t_{cyc}	25.24
Timer clock pulse width (both edges specified)	$t_{TCKWH/L}$	2.5	—	t_{cyc}	
Timer clock pulse width (phase measurement mode)	$t_{TCKWH/L}$	2.5	—	t_{cyc}	

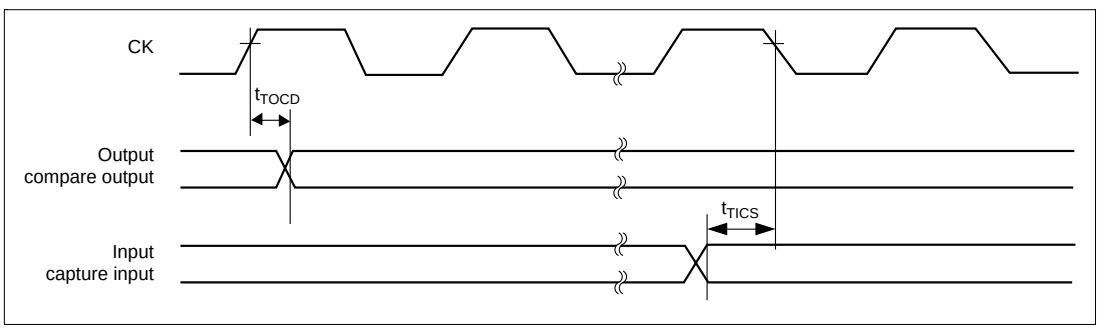


Figure 25.23 MTU I/O Timing

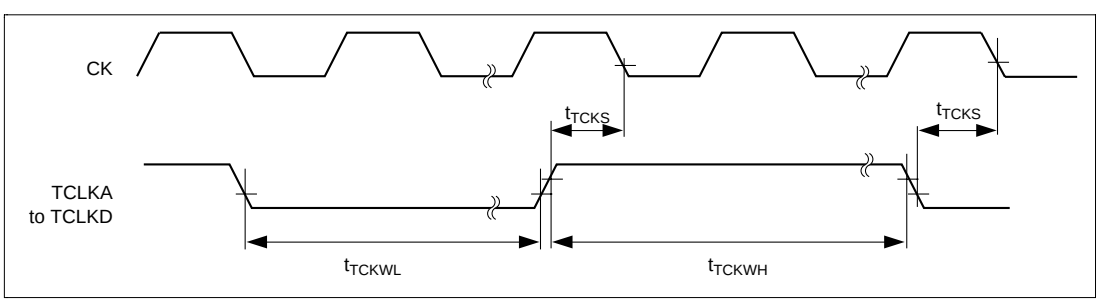


Figure 25.24 MTU Clock Input Timing

25.3.6 I/O Port Timing

Table 25.10 I/O Port Timing (Conditions: V_{CC} = 5.0 V ± 10%, AV_{CC} = 5.0 V ± 10%, AV_{CC} = V_{CC} ± 10%, AV_{ref} = 4.5 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, Ta = – 20 to +75°C)

Item	Symbol	Min	Max	Unit	Figure
Port output data delay time	t_{PWD}	—	100	ns	25.25
Port input hold time	t_{PRH}	35	—	ns	
Port input setup time	t_{PRS}	35	—	ns	

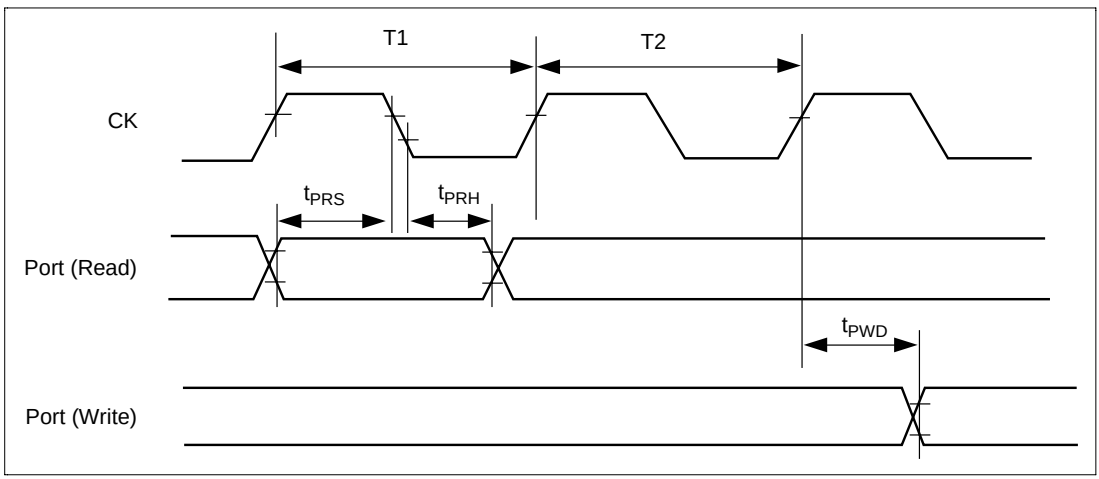


Figure 25.25 I/O Port I/O Timing

25.3.7 Watchdog Timer Timing

Table 25.11 Watchdog Timer Timing (Conditions: V_{CC} = 5.0 V ± 10%, AV_{CC} = 5.0 V ± 10%, AV_{CC} = V_{CC} ± 10%, AV_{ref} = 4.5 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, Ta = – 20 to +75°C)

Item	Symbol	Min	Max	Unit	Figure
$\overline{\text{WDTOVF}}$ delay time	t_{WOVD}	—	100	ns	25.26

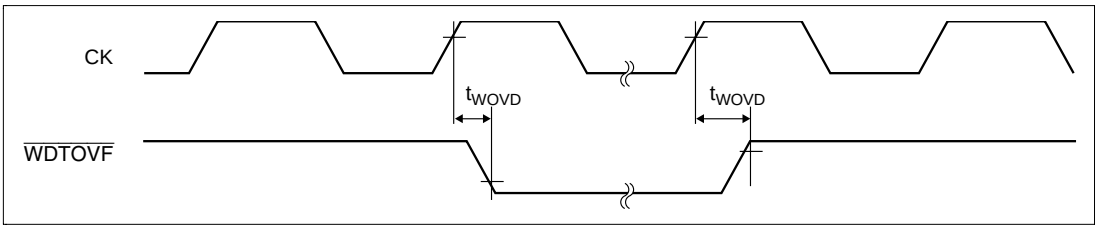


Figure 25.26 Watchdog Timer Timing

25.3.8 Serial Communication Interface Timing

Table 25.12 Serial Communication Interface Timing (Conditions: V_{CC} = 5.0 V ± 10%, AV_{CC} = 5.0 V ± 10%, AV_{CC} = V_{CC} ± 10%, AV_{ref} = 4.5 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, Ta = – 20 to +75°C)

Item	Symbol	Min	Max	Unit	Figure
Input clock cycle	t_{scyc}	4	—	t_{cyc}	25.27
Input clock cycle (clock sync)	t_{scyc}	6	—	t_{cyc}	
Input clock pulse width	t_{sckw}	0.4	0.6	t_{scyc}	
Input clock rise time	t_{sckr}	—	1.5	t_{cyc}	
Input clock fall time	t_{sckf}	—	1.5	t_{cyc}	
Transmit data delay time (clock sync)	t_{TXD}	—	100	ns	25.28
Receive data setup time (clock sync)	t_{RXS}	100	—	ns	
Receive data hold time (clock sync)	t_{RXH}	100	—	ns	

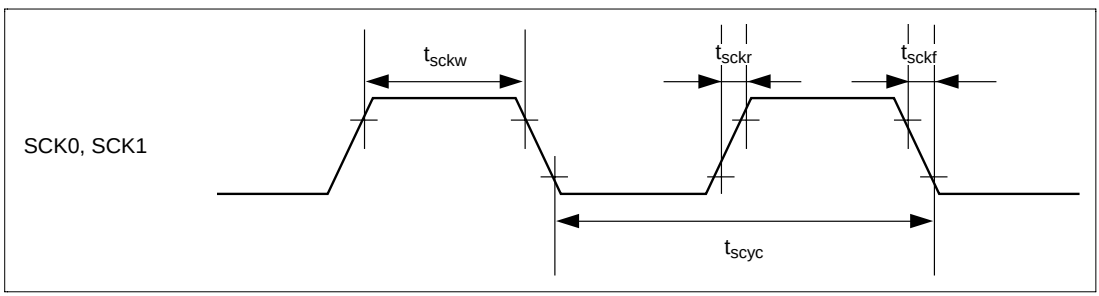


Figure 25.27 Input Clock Timing

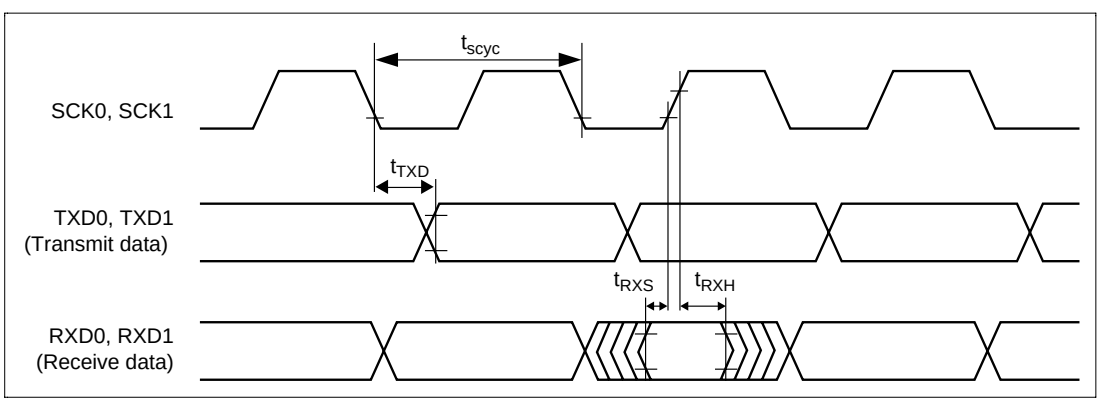


Figure 25.28 SCI I/O Timing (Clock Sync Mode)

25.3.9 High-speed A/D Converter Timing (excluding A mask)

Table 25.13 High-speed A/D Converter Timing (Conditions: V_{CC} = 5.0 V ± 10%, AV_{CC} = 5.0 V ± 10%, AV_{CC} = V_{CC} ± 10%, AV_{ref} = 4.5 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, Ta = – 20 to +75°C)

Item		Symbol	Min	Typ	Max	Unit	Figure
External trigger input pulse width		t_{TRGW}	2	—	—	t_{cyc}	25.29
External trigger input start delay time		t_{TRGS}	50	—	—	ns	
A/D conversion start delay time	CKS = 0	t_D	1.5	1.5	1.5	t_{cyc}	25.30
	CKS = 1		1.5	1.5	1.5		
Input sampling time	CKS = 0	t_{SPL}	20	20	20		
	CKS = 1		40	40	40		
A/D conversion time	CKS = 0	t_{CONV}	42.5	42.5	42.5		
	CKS = 1		82.5	82.5	82.5		

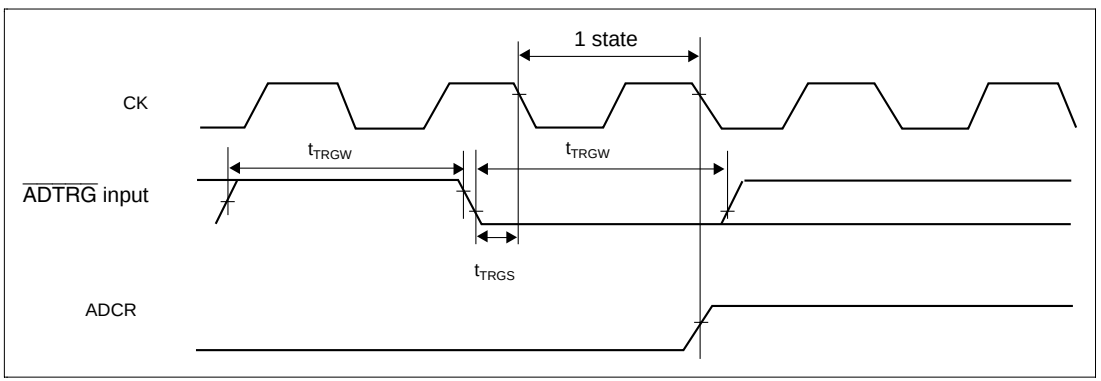


Figure 25.29 External Trigger Input Timing

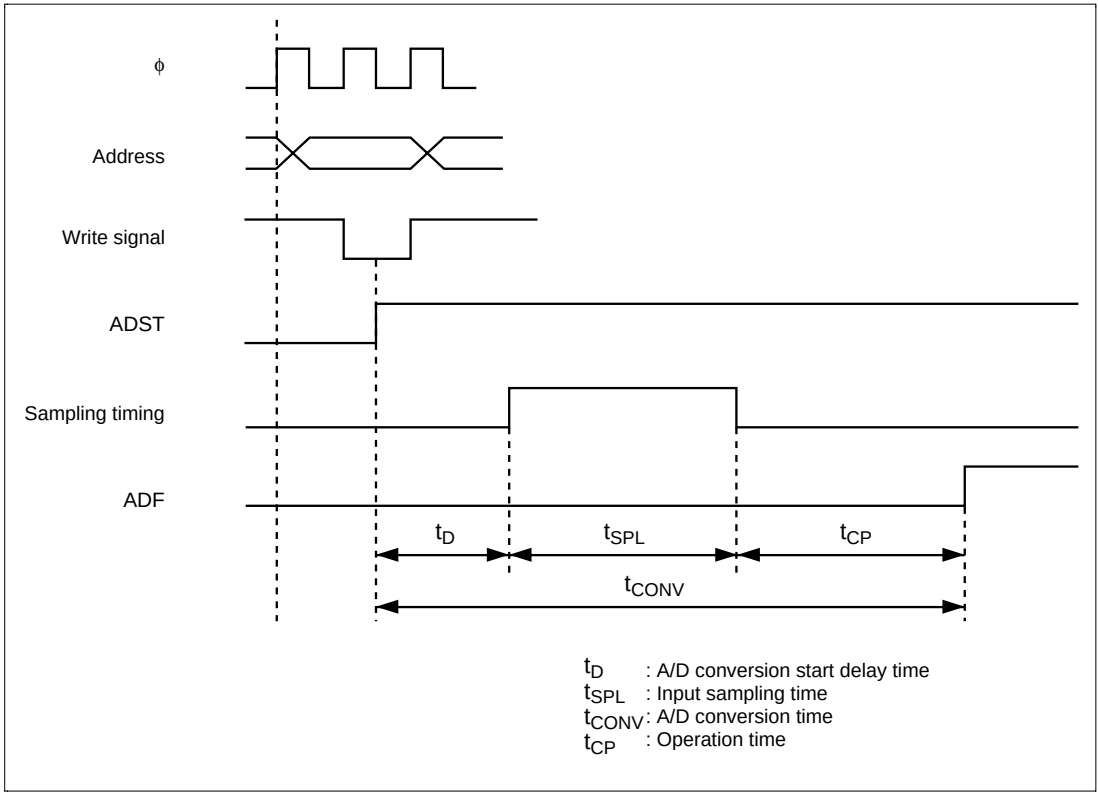


Figure 25.30 Analog Conversion Timing

25.3.10 Mid-speed Converter Timing (A mask)

Table 25.14 shows Mid-speed converter timing

Table 25.14 Mid-speed Converter Timing (Conditions:Vcc=5.0V ± 10%, AVcc=5.0V ± 10%, AVcc=Vcc ± 10%, AVref=4.5V to Avcc, Vss=AVss=0V, Ta=-20 to ± 75°C)

Item		Symbol	Min	Typ	Max	Unit	Figure
External trigger input pulse width		t_{TRGW}	2	—	—	t_{cyc}	25.31
External trigger input start delay time		t_{TRGS}	50	—	—	ns	
A/D conversion start delay time	CKS = 0	t_D	10	—	17	t_{cyc}	25.32
	CKS = 1		6	—	9		
Input sampling time	CKS = 0	t_{SPL}	—	64	—		
	CKS = 1		—	32	—		
A/D conversion time	CKS = 0	t_{CONV}	259	—	266		
	CKS = 1		131	—	134		

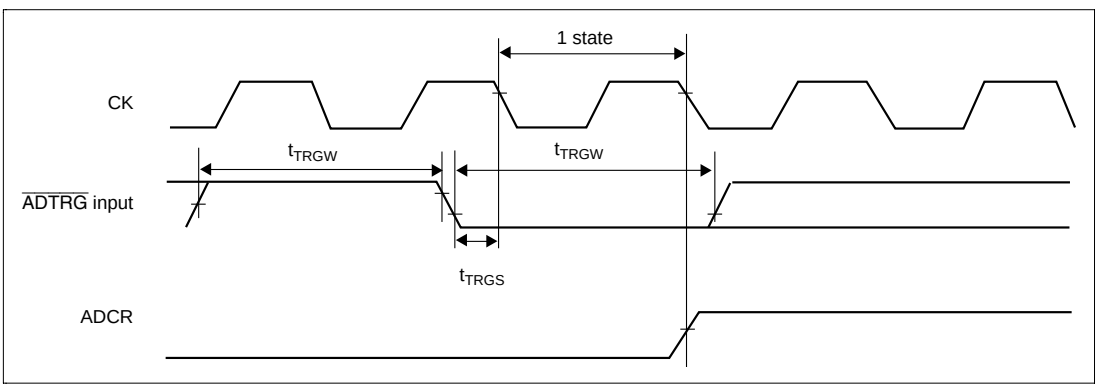
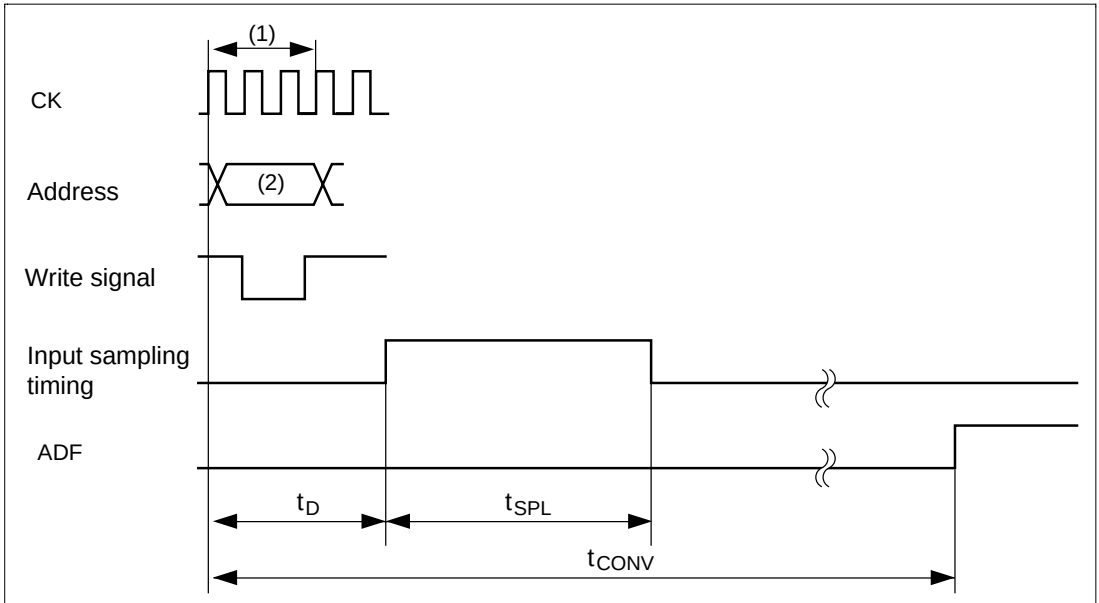


Figure 25.31 External Trigger Input Timing



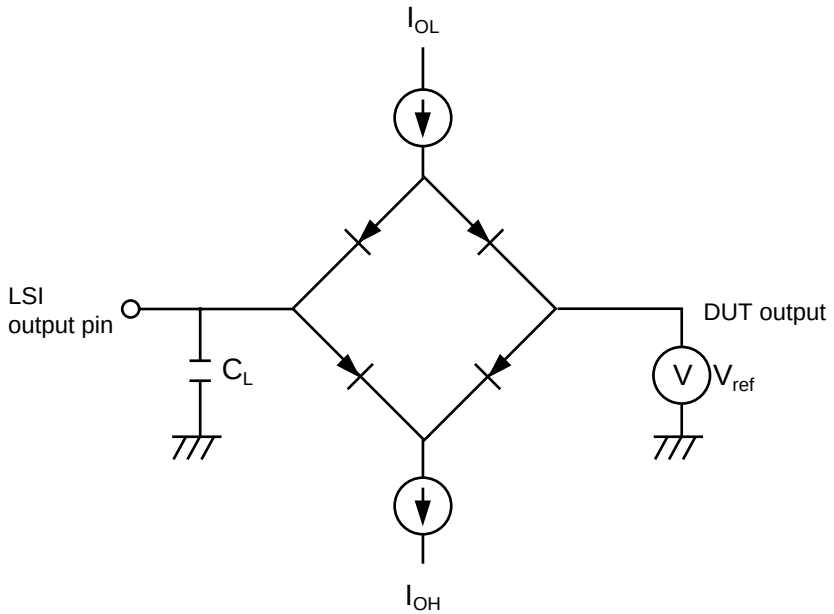
Legend:

(1) : ADCSR write cycle
(2) : ADCSR address
t_D : A/D conversion start delay time
t_{SPL} : Input sampling time
t_{CONV} : A/D conversion time

Figure 25.32 Analog Conversion Timing

25.3.11 Measuring Conditions for AC Characteristics

- Input reference levels:
  - — High level: 2.2 V
  - — Low level: 0.8 V
- Output reference levels:
  - — High level: 2.0 V
  - — Low level: 0.8 V



Note: C_L is set with the following pins, including the total capacitance of the measurement equipment etc:

30 pF: CK, $\overline{RAS}$, $\overline{CASxx}$, RDWR, $\overline{CS0}$–$\overline{CS3}$, $\overline{AH}$, $\overline{BREQ}$, $\overline{BACK}$, DACK0, DACK1, and $\overline{IRQOUT}$

50 pF: A21–A0, D31–D0, $\overline{RD}$, $\overline{WRxx}$

70 pF: Port output and peripheral module output pins other than the above.

I_{OL}, I_{OH}: See table 25.3, Permitted Output Current Values.

Figure 25.33 Output Load Circuit

25.4 A/D Converter Characteristics

Table 25.15 A/D Converter Timing (excluding A mask) (Conditions: V_{CC} = 5.0 V ± 10%, AV_{CC} = 5.0 V ± 10%, AV_{CC} = V_{CC} ± 10%, AV_{ref} = 4.5 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, Ta = – 20 to +75°C)

	28.7 MHz			
Item	**Min**	**Typ**	**Max**	**Unit**
Resolution	10	10	10	Bits
Conversion time (when CKS = 1)	—	—	2.9	µs
Analog input capacitance	—	—	20	pF
Permitted signal source impedance	—	—	1	kΩ
Non-linear error*	—	—	±8	LSB
Offset error*	—	—	±8	LSB
Full-scale error*	—	—	±8	LSB
Quantization error*	—	—	±0.5	LSB
Absolute error (when CKS = 1)	—	—	±15	LSB

Note: * Reference values

Table 25.16 A/D Converter Timing (A mask) (Condition: Vcc=5.0 ± 10%, AVcc=5.0 ± 10%, AVcc=Vcc ± 10%, AVref=4.5V to AVcc, Vss=AVss=0V, Ta=-20 to +75°C)

	28.7 MHz			20 MHz			
Item	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Unit**
Resolution	10	10	10	10	10	10	Bits
Conversion time (when CKS=0)	—	—	9.3	—	—	13.4	µs
Analog input capacity	—	—	20	—	—	20	pF
Permission signal source impedance	—	—	1	—	—	1	kΩ
Non-linearity error*	—	—	±3	—	—	±3	LSB
Offset error*	—	—	±3	—	—	±3	LSB
Full scale error*	—	—	±3	—	—	±3	LSB
Quantize error*	—	—	±0.5	—	—	±0.5	LSB
Absolute error	—	—	±4	—	—	±4	LSB

Note: * Reference value

Section 26 Electrical Characteristics (3.3V, 16.7 MHz Version)

26.1 Absolute Maximum Ratings

Table 26.1 Absolute Maximum Ratings

Item	Symbol	Rating	Unit
Power supply voltage	V_{CC}	–0.3 to +7.0	V
Programmable voltage (ZTAT version only)	V_{PP}	–0.3 to +13.5	V
Input voltage (other than A/D ports)	V_{in}	–0.3 to V_{CC} + 0.3	V
Input voltage (A/D ports)	V_{in}	–0.3 to AV_{CC} + 0.3	V
Analog supply voltage	AV_{CC}	–0.3 to +7.0	V
Analog reference voltage (QFP-144 only)	AV_{ref}	–0.3 to AV_{CC} + 0.3	V
Analog input voltage	V_{AN}	–0.3 to AV_{CC} + 0.3	V
Operating temperature	T_{opr}	–20 to +75	°C
Programming temperature (ZTAT version only)	T_{we}	–20 to +75	°C
Storage temperature	T_{stg}	–55 to +125	°C

Note: Operating the LSI in excess of the absolute maximum ratings may result in permanent damage.

26.2 DC Characteristics

Table 26.2 DC Characteristics (Conditions: V_{CC} = 3.0[*1] to 3.6V, AV_{CC} = 3.0[*1] to 3.6V, AV_{CC} = V_{CC} ± 10%, AV_{ref} = 3.0[*1] to AV_{CC}, V_{SS} = AV_{SS} = 0V, T_a = –20 to +75°C)

Item	Pin	Symbol	Min	Typ	Max	Unit	Measurement Conditions
Input high-level voltage	$\overline{RES}$, NMI, MD3–0, PA2, PA5, PA6–PA9, PA0–PE15, FWP	V_{IH}	V_{CC} × 0.9	—	V_{CC} + 0.3	V	
	EXTAL		V_{CC} × 0.9	—	V_{CC} + 0.3	V	
	A/D port		V_{CC} × 0.7	—	AV_{CC} + 0.3	V	
	Other input pins		V_{CC} × 0.7	—	V_{CC} + 0.3	V	
Input low-level voltage	$\overline{RES}$, NMI, MD3–0, PA2, PA5, PA6–PA9, PA0–PE15, FWP	V_{IL}	–0.3	—	V_{CC} × 0.1	V	
	Other input pins		–0.3	—	V_{CC} × 0.2	V	
Schmitt trigger input voltage	PA2, PA5, PA6–PA9, PE0–PE15	$V_T^+ - V_T^-$	V_{CC} × 0.07	—	—	V	$VT^+ \geq V_{CC} \times 0.9V$ (min) $VT^- \leq V_{CC} \times 0.2V$ (max)
Input leak current	$\overline{RES}$, NMI, MD3–0, PA2, PA5, PA6–PA9, PE0–PE15,FWP	$\lvert I_{in} \rvert$	—	—	1.0	µA	V_{in} = 0.5 to V_{CC} – 0.5V
	A/D port		—	—	1.0	µA	V_{in} = 0.5 to AV_{CC} – 0.5V
	Other input pins (except EXTAL pin)		—	—	1.0	µA	V_{in} = 0.5 to V_{CC} – 0.5V
Three–state leak current (while off)	A21–A0, D31–D0, $\overline{CS3}$–$\overline{CS0}$, RDWR, $\overline{RAS}$, $\overline{CAS}$xx, $\overline{WR}$xx, $\overline{RD}$, Ports A, B, C, D, E	$\lvert I_{TS} \rvert$	—	—	1.0	µA	V_{in} = 0.5 to V_{CC} – 0.5V

Table 26.2 DC Characteristics (Conditions: V_{CC} = 3.0*1 to 3.6V, AV_{CC} = 3.0*1 to 3.6V, AV_{CC} = V_{CC} ± 10%, AV_{ref} = 3.0*1 to AV_{CC}, V_{SS} = AV_{SS} = 0V, T_a = –20 to +75°C) (cont)

Item	Pin	Symbol	Min	Typ	Max	Unit	Measurement Conditions
Output high-level voltage	All output pins	V_{OH}	V_{CC}– 0.5	—	—	V	I_{OH}= –200 µA
			V_{CC}– 1.0	—	—	V	I_{OH}= –1mA
Output low-level voltage	All output pins	V_{OL}	—	—	0.4	V	I_{OL}= 1.6mA
Input capacitance	$\overline{RES}$	C_{in}	—	—	80*2	pF	Vin= 0V
	NMI		—	—	50	pF	f = 1 MHz
	All other input pins		—	—	20	pF	Ta = 25°C
Current consumption	During normal operations	I_{CC}	—	80	130	mA	f = 16.7MHz
	During sleep mode		—	70	110	mA	f = 16.7MHz
	During standby mode		—	0.01	5	µA	$T_a \leq 50°C$
			—	—	20	µA	50°C < Ta
Analog supply current		AI_{CC}	—	4	8	mA	f = 16.7MHz
		AI_{ref}	—	0.5	1*3	mA	QFP144 version only
RAM standby voltage		V_{RAM}	2.0	—	—	V	

Notes: 1. Do not release AV_{CC}, AV_{SS} and AV_{ref} (SH7041, SH7043 and SH7045 only) pins when not using the A/D converter (including standby).

Connect AV_{CC} (SH7041,SH7043,SH7045 only) and AV_{ref} (SH7041, SH7043 and SH7045 only) pins to V_{CC} and AV_{SS} pin to V_{SS}.

2. The value for consumed current is with conditions of V_{IH}min = V_{CC} – 0.5V and V_{IL}max = 0.5V, with no burden on any of the output pins.
3. The ZTAT and mask versions have the same functions, and the electrical characteristics of both are within specification, but characteristic-related performance values, operating margins, noise margins, noise emission, etc., are different. Caution is therefore required in carrying out system design, and when switching between ZTAT and mask versions.

*1 SH7042/43 ZTAT (excluding A mask) are 3.2 V.

*2 110pF for A mask

*3 2 mA in the A mask version of MASK products.

Table 26.3 Permitted Output Current Values (Conditions: V_{CC} = 3.0* to 3.6V, AV_{CC} = 3.0* to 3.6V, AV_{CC} = V_{CC} ± 10%, AV_{ref} = 3.0* to AV_{CC}, V_{SS} = AV_{SS} = 0V, T_a = –20 to +75°C)

Item	Symbol	Min	Typ	Max	Unit
Output low-level permissible current (per pin)	I_{OL}	—	—	2.0	mA
Output low-level permissible current (total)	$\sum I_{OL}$	—	—	80	mA
Output high-level permissible current (per pin)	$-I_{OH}$	—	—	2.0	mA
Output high-level permissible current (total)	$\sum (-I_{OH})$	—	—	25	mA

Notes: To assure LSI reliability, do not exceed the output values listed in this table.

* SH7042/43 ZTAT (excluding A mask) are 3.2V.

26.3 AC Characteristics

26.3.1 Clock Timing

Table 26.4 Clock Timing (Conditions: V_{CC} = 3.0* to 3.6V, AV_{CC} = 3.0* to 3.6V, AV_{CC} = V_{CC} ± 10%, AV_{ref} = 3.0* to AV_{CC}, V_{SS} = AV_{SS} = 0V, T_a = –20 to +75°C)

Item	Symbol	Min	Max	Unit	Figures
Operating frequency	f_{OP}	4	16.7	MHz	26.1
Clock cycle time	t_{cyc}	60	250	ns	
Clock low-level pulse width	t_{CL}	10	—	ns	
Clock high-level pulse width	t_{CH}	10	—	ns	
Clock rise time	t_{CR}	—	5	ns	
Clock fall time	t_{CF}	—	5	ns	
EXTAL clock input frequency	f_{EX}	4	10	MHz	26.2
EXTAL clock input cycle time	t_{EXcyc}	100	250	ns	
EXTAL clock low-level input pulse width	t_{EXL}	40	—	ns	
EXTAL clock high-level input pulse width	t_{EXH}	40	—	ns	
EXTAL clock input rise time	t_{EXR}	—	5	ns	
EXTAL clock input fall time	t_{EXF}	—	5	ns	
Reset oscillation settling time	t_{OSC1}	10	—	ms	26.3
Standby return clock settling time	t_{OSC2}	10	—	ms	

Note: * SH7042/43 ZTAT (excluding A mask) are 3.2V.

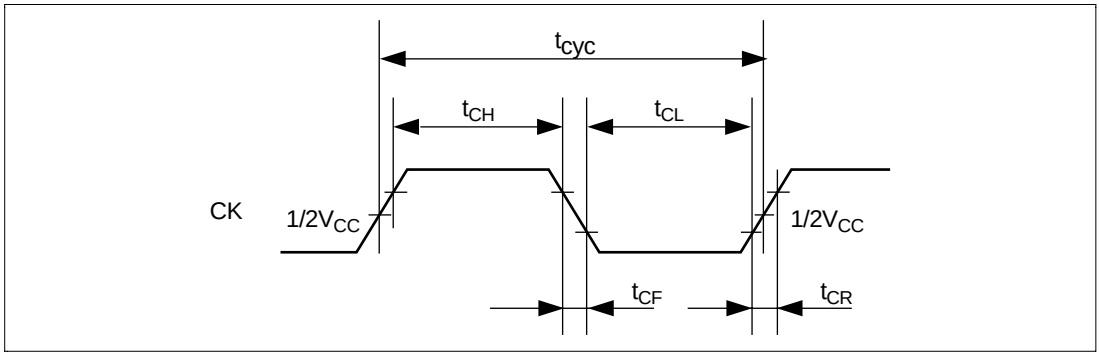


Figure 26.1 System Clock Timing

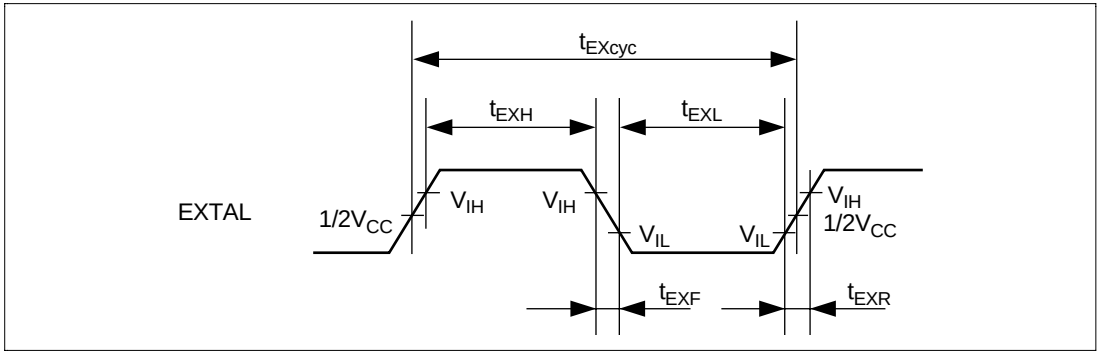


Figure 26.2 EXTAL Clock Input Timing

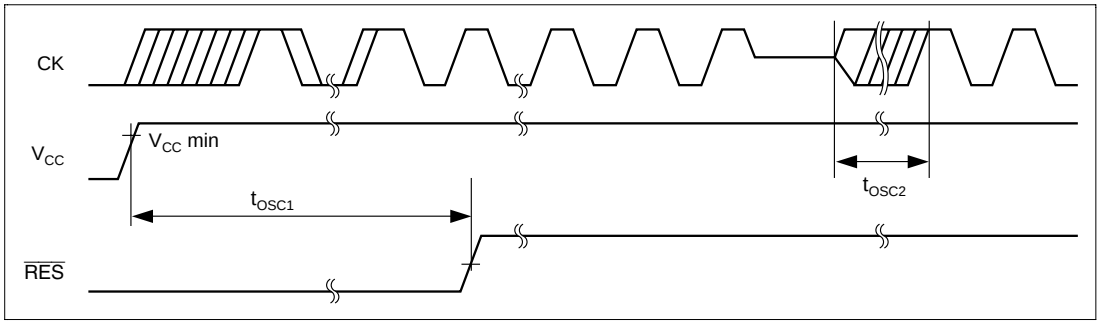


Figure 26.3 Oscillation Settling Time

26.3.2 Control Signal Timing

Table 26.5 Control Signal Timing (Conditions: V_{CC} = 3.0*1 to 3.6V, AV_{CC} = 3.0*1 to 3.6V, AV_{CC} = V_{CC} ± 10%, AV_{ref} = 3.0*1 to AV_{CC}, V_{SS} = AV_{SS} = 0V, T_a = –20 to +75°C)

Item	Symbol	Min	Max	Unit	Figure
$\overline{RES}$ rise/fall	t_{RESr}, t_{RESf}	—	200	ns	26.4
$\overline{RES}$ pulse width	t_{RESW}	20	—	t_{cyc}	
$\overline{MRES}$ pulse width	t_{MRESW}	20	—	t_{cyc}	
NMI rise/fall	t_{NMIr}, t_{NMIf}	—	200	ns	26.5
$\overline{RES}$ setup time*1	t_{RESS}	100	—	ns	26.4
$\overline{MRES}$ setup time*1	t_{MRESS}	100	—	ns	26.5
NMI setup time (during edge detection)	t_{NMIS}	100	—	ns	
$\overline{IRQ7}$–$\overline{IRQ0}$ setup time (edge detection)*2	t_{IRQES}	100	—	ns	
$\overline{IRQ7}$–$\overline{IRQ0}$ setup time (level detection)*2	t_{IRQLS}	100	—	ns	
NMI hold time	t_{NMIH}	50	—	ns	26.5
$\overline{IRQ7}$–$\overline{IRQ0}$ hold time	t_{IRQEH}	50	—	ns	
$\overline{IRQOUT}$ output delay time	t_{IRQOD}	—	50	ns	26.6
Bus request setup time	t_{BRQS}	35	—	ns	26.7
Bus acknowledge delay time 1	t_{BACKD1}	—	35	ns	
Bus acknowledge delay time 2	t_{BACKD2}	—	35	ns	
Bus three state delay time	t_{BZD}	—	35	ns	

Notes: *1 SH7042/43 ZTAT (excluding A mask) are 3.2V.

*2 The $\overline{RES}$, $\overline{MRES}$, NMI, $\overline{BREQ}$, and $\overline{IRQ7}$–$\overline{IRQ0}$ signals are asynchronous inputs, but when the setup times shown here are provided, the signals are considered to have produced changes at clock rise (for $\overline{RES}$, $\overline{MRES}$, $\overline{BREQ}$) or clock fall (for NMI and $\overline{IRQ7}$–$\overline{IRQ0}$). If the setup times are not provided, recognition is delayed until the next clock rise or fall.

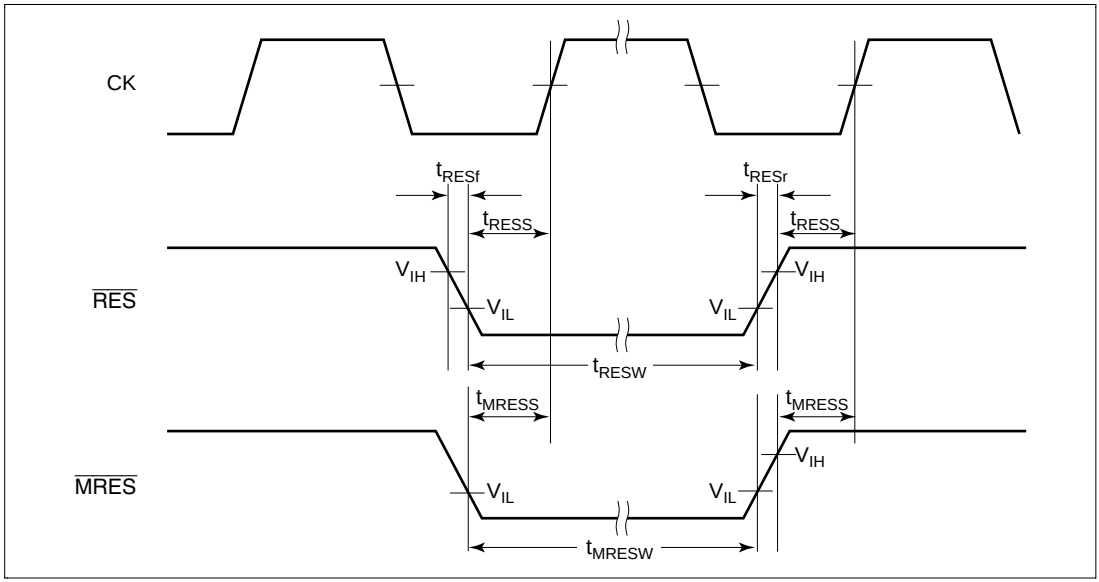


Figure 26.4 Reset Input Timing

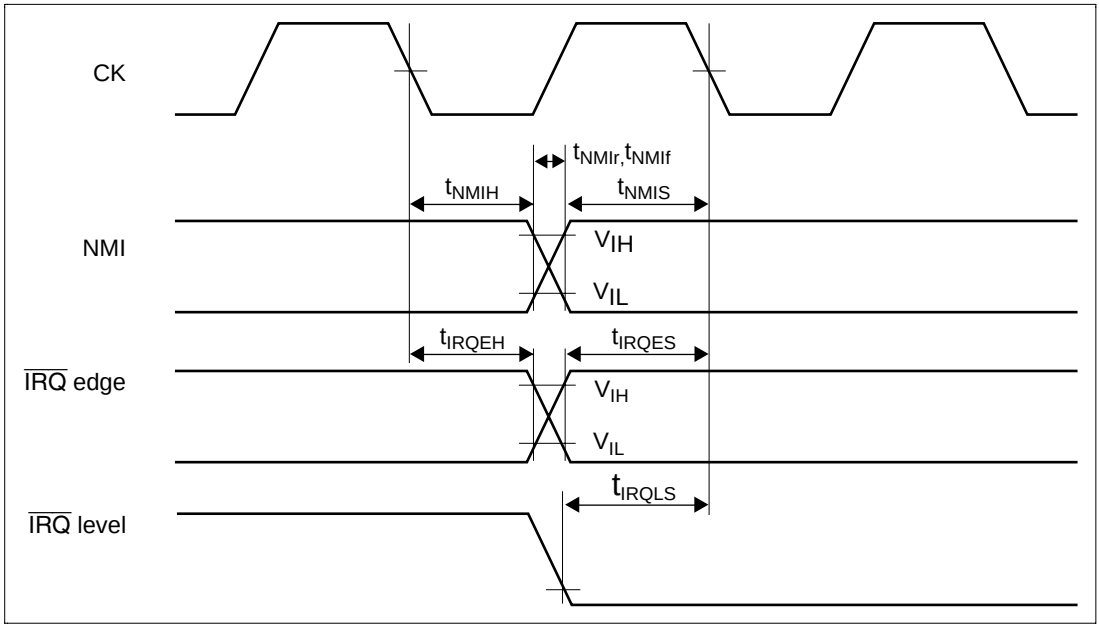


Figure 26.5 Interrupt Signal Input Timing

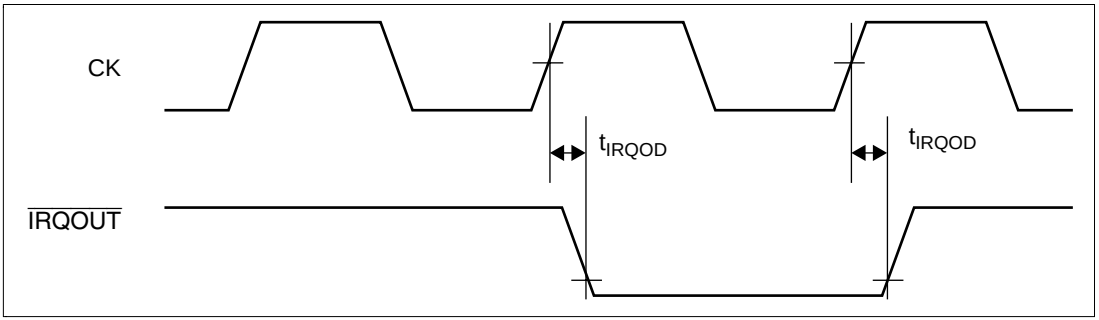
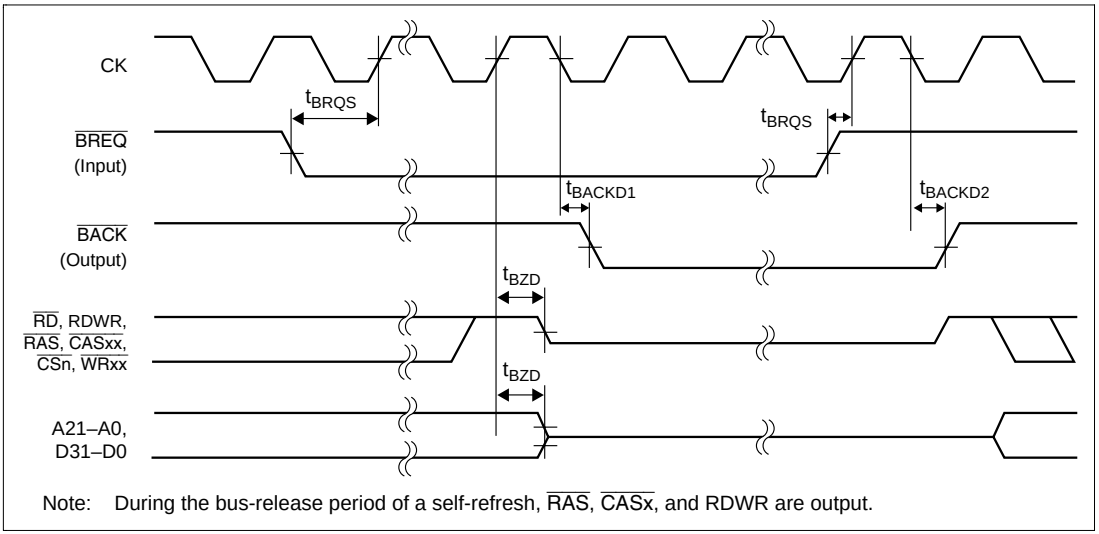


Figure 26.6 Interrupt Signal Output Timing



Note: During the bus-release period of a self-refresh, RAS, CASx, and RDWR are output.

Figure 26.7 Bus Right Release Timing

26.3.3 Bus Timing

Table 26.6 Bus Timing (Conditions: V_{CC} = 3.0*1 to 3.6V, AV_{CC} = 3.0*1 to 3.6V, AV_{CC} = V_{CC} ± 10%, AV_{ref} = 3.0*1 to AV_{CC}, V_{SS} = AV_{SS} = 0V, T_a = –20 to +75°C)

Item	Symbol	Min	Max	Unit	Figure
Address delay time	t_{AD}	3*4	35	ns	26.8, 9, 11–16, 19
$\overline{CS}$ delay time 1	t_{CSD1}	3*4	35	ns	26.8, 9, 19
$\overline{CS}$ delay time 2	t_{CSD2}	3*4	35	ns	
Read strobe delay time 1	t_{RSD1}	3*4	35	ns	26.8, 9, 11–16, 19
Read strobe delay time 2	t_{RSD2}	3*4	35	ns	
Read data setup time	t_{RDS}*5	25	—	ns	
Read data hold time	t_{RDH}	0	—	ns	
Write strobe delay time 1	t_{WSD1}	3*4	35	ns	
Write strobe delay time 2	t_{WSD2}	3*5	35	ns	
Write data delay time	t_{WDD}	—	45	ns	
Write data hold time	t_{WDH}	0	25*3	ns	
$\overline{WAIT}$ setup time	t_{WTS}	15	—	ns	26.10,15, 19
$\overline{WAIT}$ hold time	t_{WTH}	0	—	ns	
$\overline{RAS}$ delay time 1	t_{RASD1}	3*4	35	ns	26.11–18
$\overline{RAS}$ delay time 2	t_{RASD2}	3*4	35	ns	
$\overline{CAS}$ delay time 1	t_{CASD1}	3*4	35	ns	
$\overline{CAS}$ delay time 2	t_{CASD2}	3*4	35	ns	
Read data access time	t_{ACC}*2	t_{cyc}× (n+2) – 45	—	ns	26.8, 9
Access time from read strobe	t_{OE}*2	t_{cyc}× (n+1.5) – 40	—	ns	
Access time from column address	t_{AA}*2	t_{cyc}× (n+2) – 45	—	ns	26.11–16
Access time from $\overline{RAS}$	t_{RAC}*2	t_{cyc}× (n+RCD+2.5) – 40	—	ns	
Access time from $\overline{CAS}$	t_{CAC}*2	t_{cyc}× (n+1) – 40	—	ns	
Row address hold time	t_{RAH}	t_{cyc}× (RCD+0.5) – 15	—	ns	
Row address setup time	t_{ASR}	0	—	ns	
Data input setup time	t_{DS}	t_{cyc}× (m+0.5) – 27	—	ns	
Data input hold time	t_{DH}	20	—	ns	

Notes: n is the wait number. m is 1 unless the DRAM write cycle wait number is 0, then m is 0. RCD is the set value of the RCD bit of DCR.

*1 SH7042/43 ZTAT (excluding A mask) are 3.2V.

*2 If the access time is satisfied, then the t_{RDS} need not be satisfied.

*3 t_{WDH} (max) is a reference value.

*4 The delay time min values are reference values (typ).

*5 t_{RDS} is a reference value.

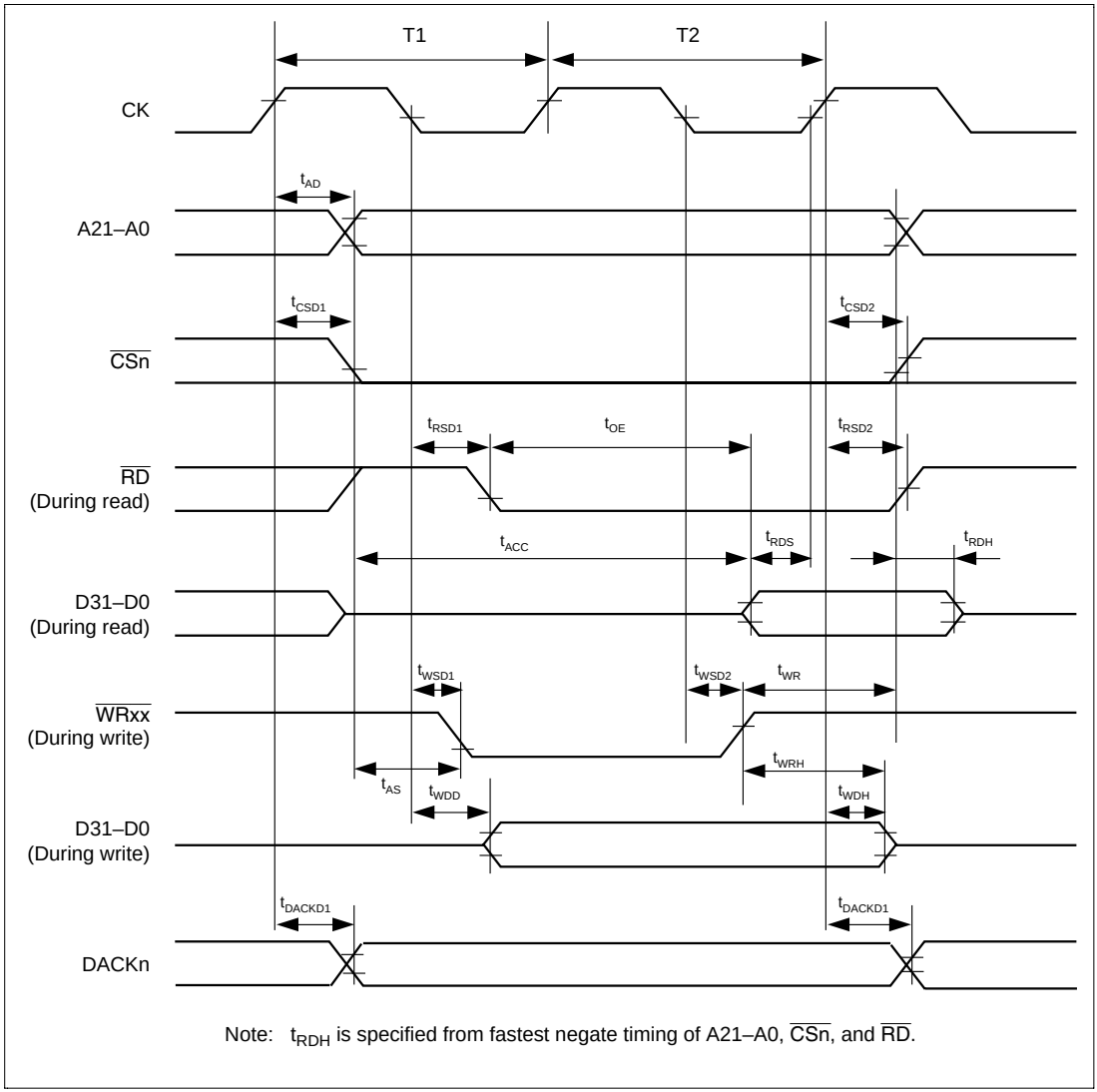
Table 26.7 Bus Timing (Conditions: V_{CC} = 3.0*1 to 3.6V, AV_{CC} = 3.0*1 to 3.6V, AV_{CC} = V_{CC} ± 10%, AV_{ref} = 3.0*1 to AV_{CC}, V_{SS} = AV_{SS} = 0V, T_a = –20 to +75°C)

Item	Symbol	Min	Max	Unit	Figure
Write address setup time	t_{AS}	0	—	ns	26.8, 9
Write address hold time	t_{WR}	5	—	ns	26.8, 9, 19
Write data hold time	t_{WRH}	0	—	ns	
Read/write strobe delay time 1	t_{RWD1}	3*2	27	ns	26.11–16
Read/write strobe delay time 2	t_{RWD2}	3*2	27	ns	
High-speed page mode $\overline{CAS}$	t_{CP}	t_{cyc}–35	—	ns	26.16
$\overline{RAS}$ precharge time	t_{RP}	t_{cyc}× (TPC+1.5) –20	—	ns	26.11–16
$\overline{CAS}$ setup time	t_{CSR}	10	—	ns	26.17, 18
$\overline{AH}$ delay time 1	t_{AHD1}	3*2	40	ns	26.19
$\overline{AH}$ delay time 2	t_{AHD2}	3*2	40	ns	
Multiplex address delay time	t_{MAD}	3*2	35	ns	
Multiplex address hold time	t_{MAH}	0	—	ns	
DACK delay time 1	t_{DACKD1}	3*2	45	ns	26.8, 9, 11–16,19

Notes: TPC is the set value of the TPC bit in DCR.

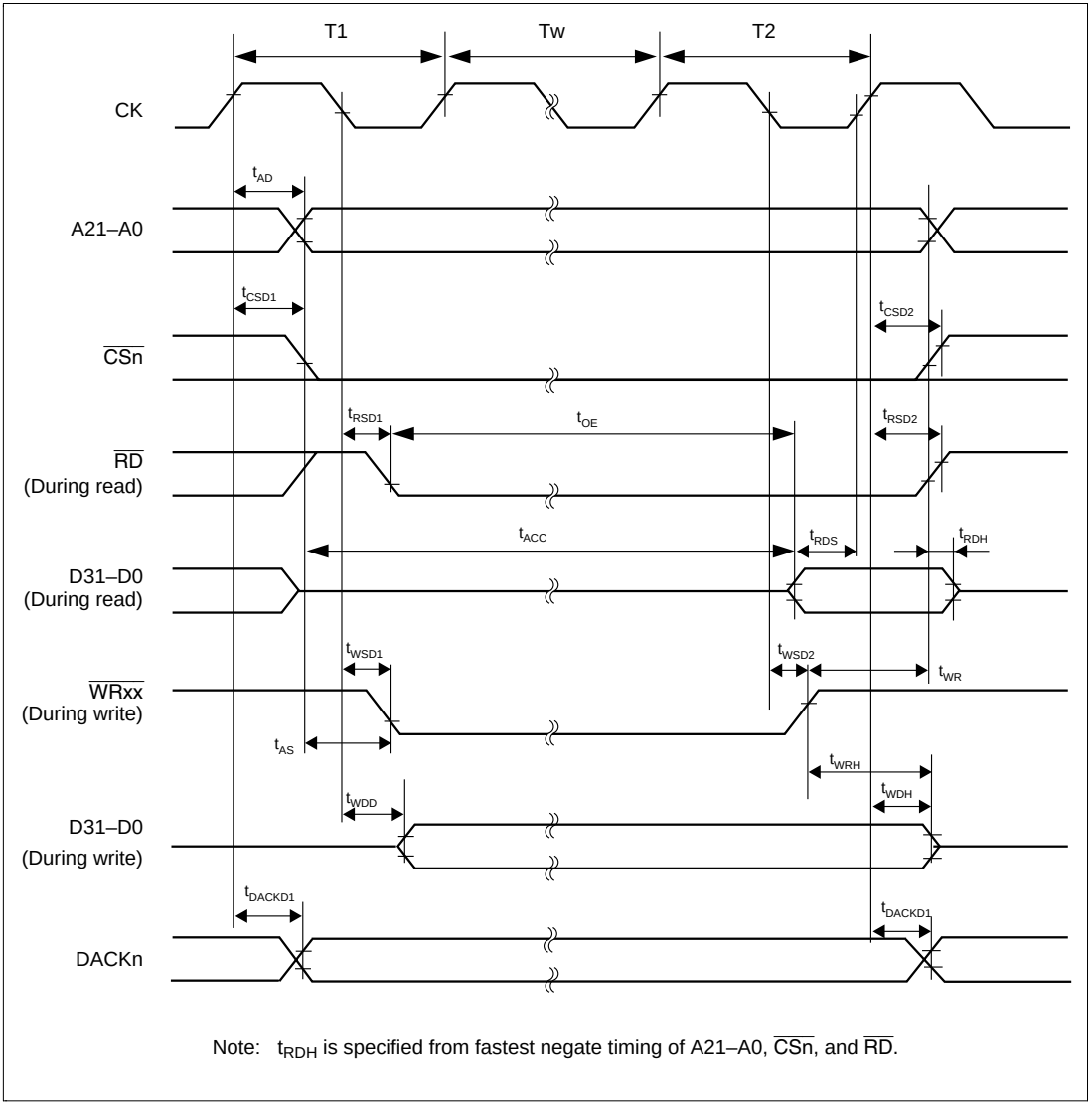
*1 SH7042/43 ZTAT (excluding A mask) are 3.2V

*2 Min values for delay time are reference values (typ)



Note: t_{RDH} is specified from fastest negate timing of A21–A0, $\overline{CSn}$, and $\overline{RD}$.

Figure 26.8 Basic Cycle (No Waits)



Note: t_{RDH} is specified from fastest negate timing of A21–A0, $\overline{CSn}$, and $\overline{RD}$.

Figure 26.9 Basic Cycle (Software Waits)

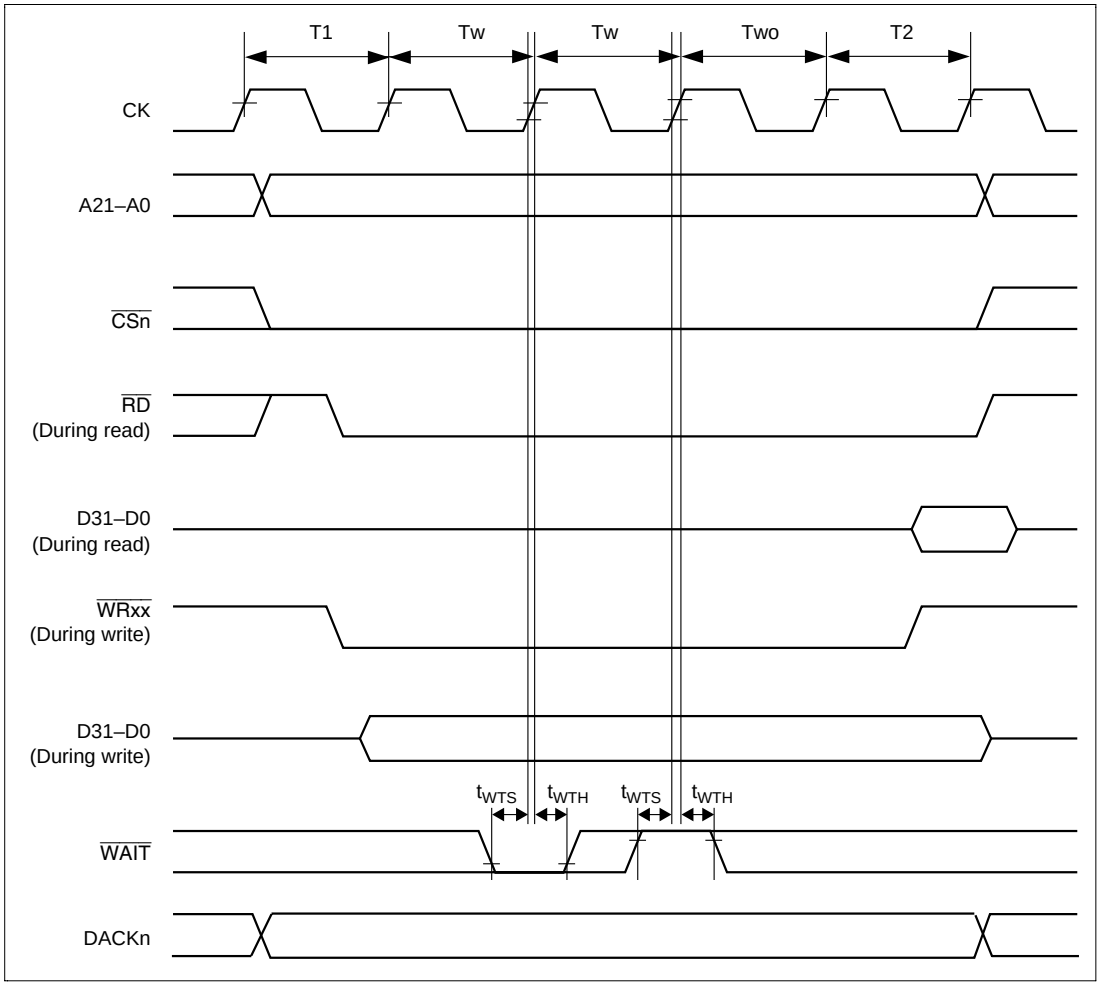
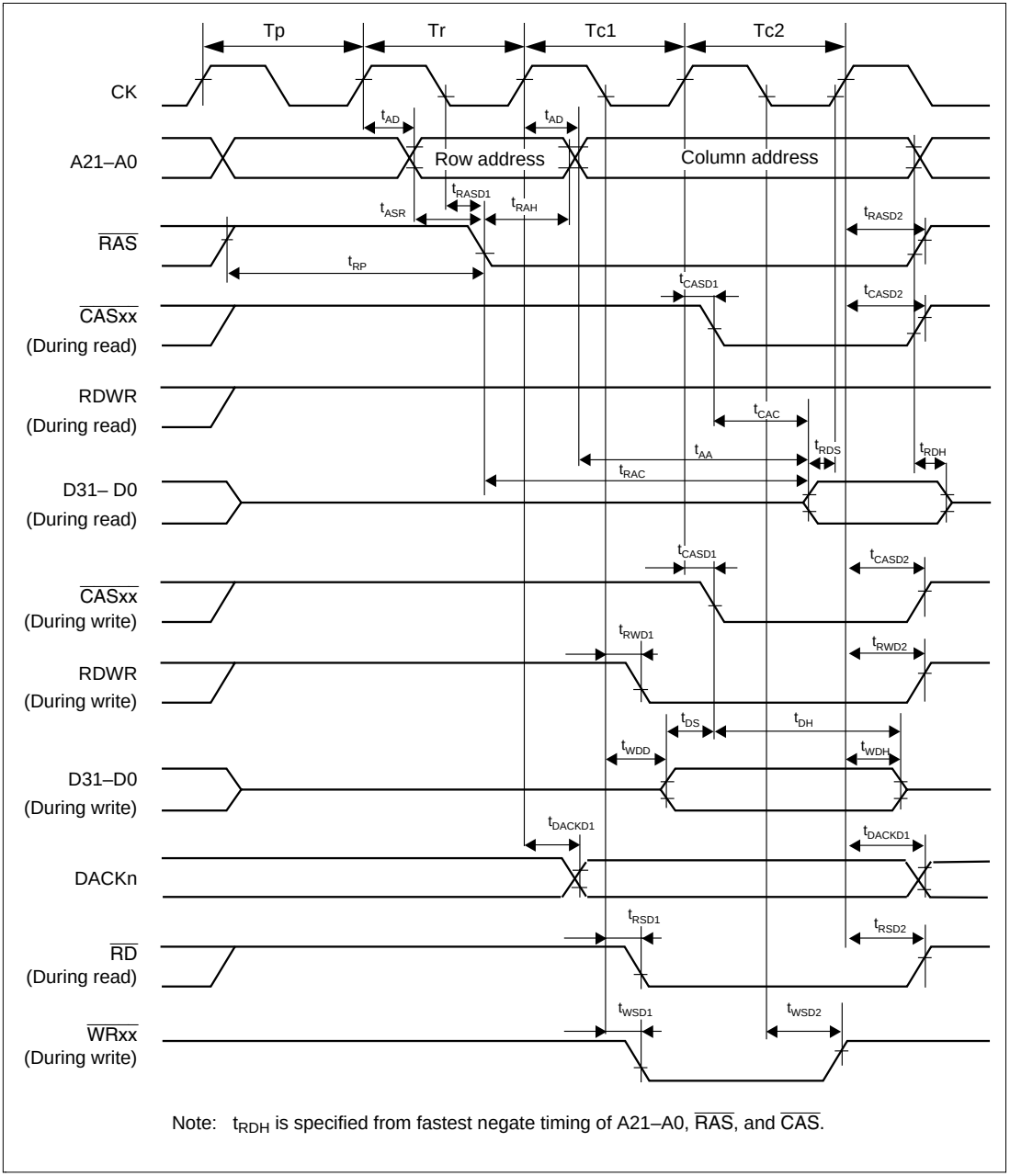
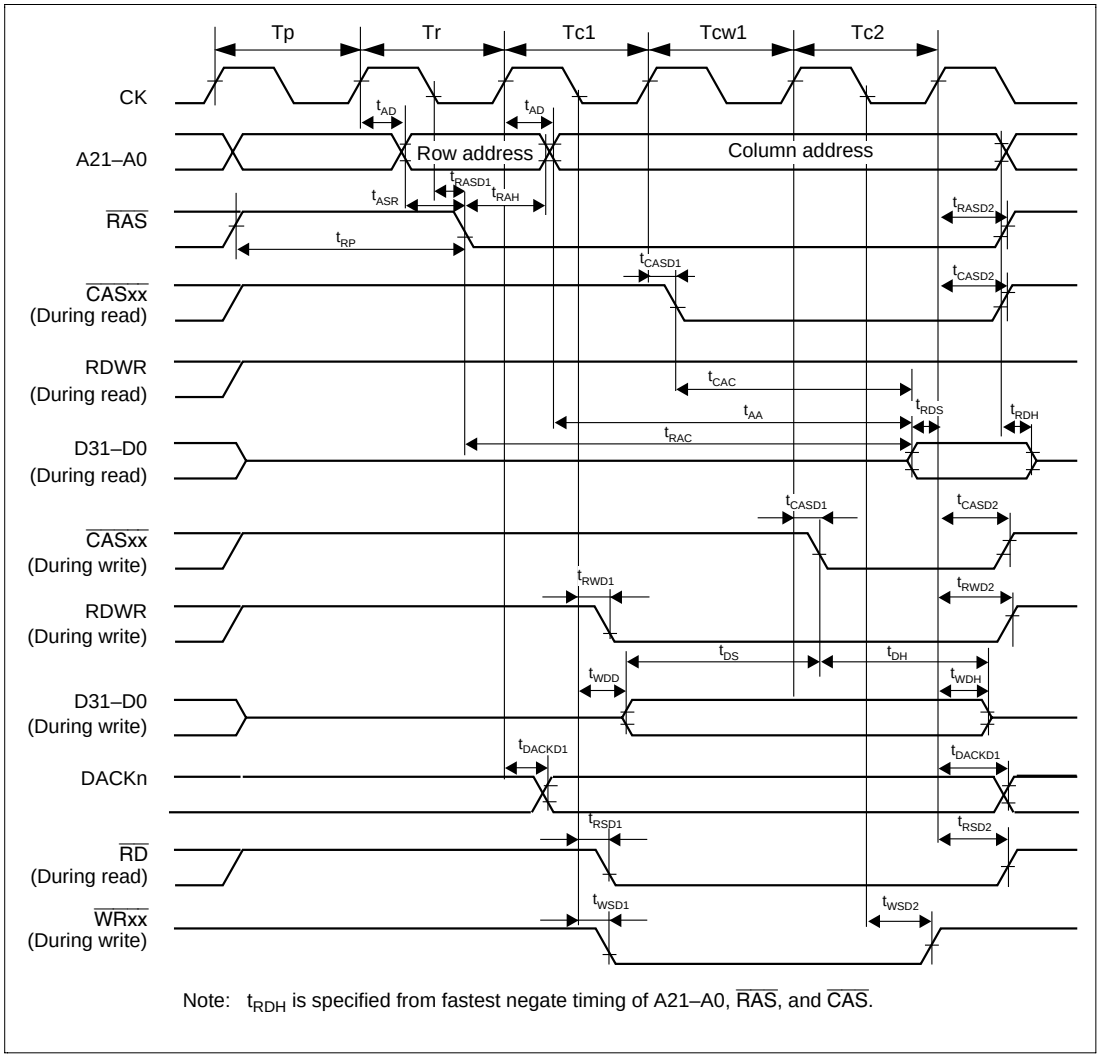


Figure 26.10 Basic Cycle (2 Software Waits + Wait due to $\overline{\text{WAIT}}$ Signal)



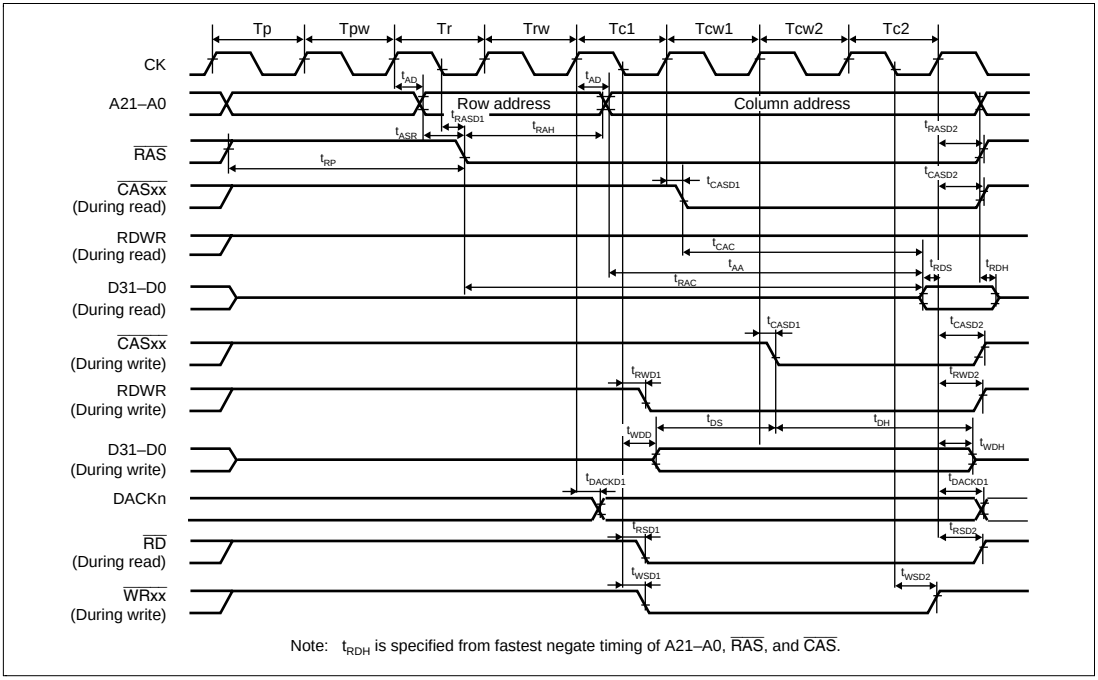
Note: t_{RDH} is specified from fastest negate timing of A21–A0, $\overline{RAS}$, and $\overline{CAS}$.

Figure 26.11 DRAM Cycle (Normal Mode, No Wait, TPC = 0, RCD = 0)



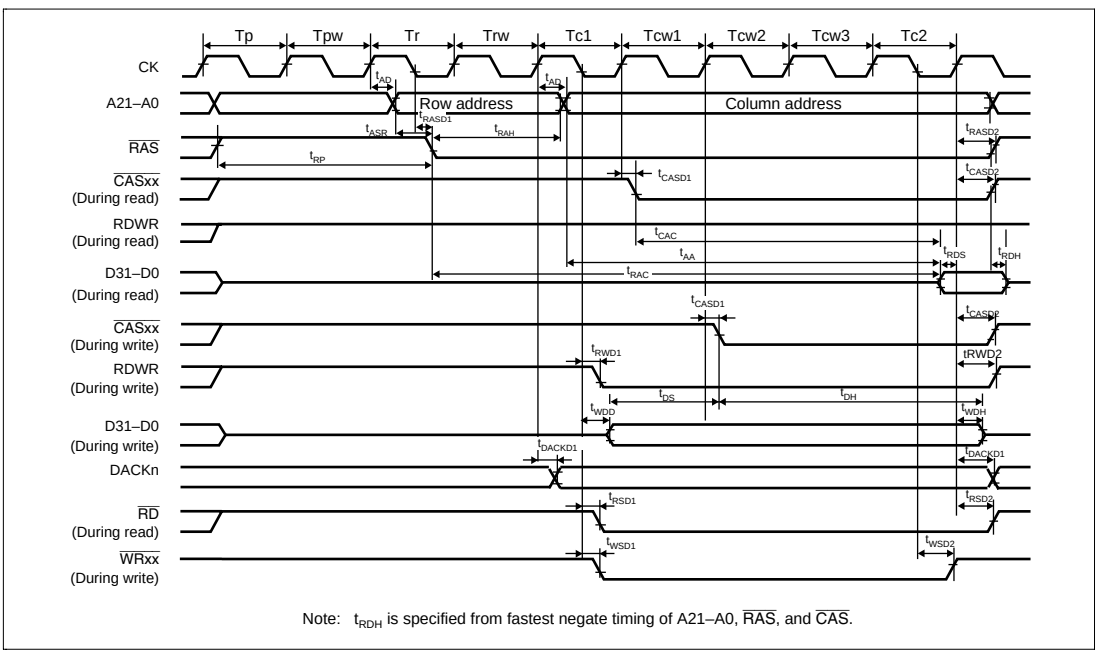
Note: t_{RDH} is specified from fastest negate timing of A21–A0, $\overline{RAS}$, and $\overline{CAS}$.

Figure 26.12 DRAM Cycle (Normal Mode, 1 Wait, TPC = 0, RCD = 0)



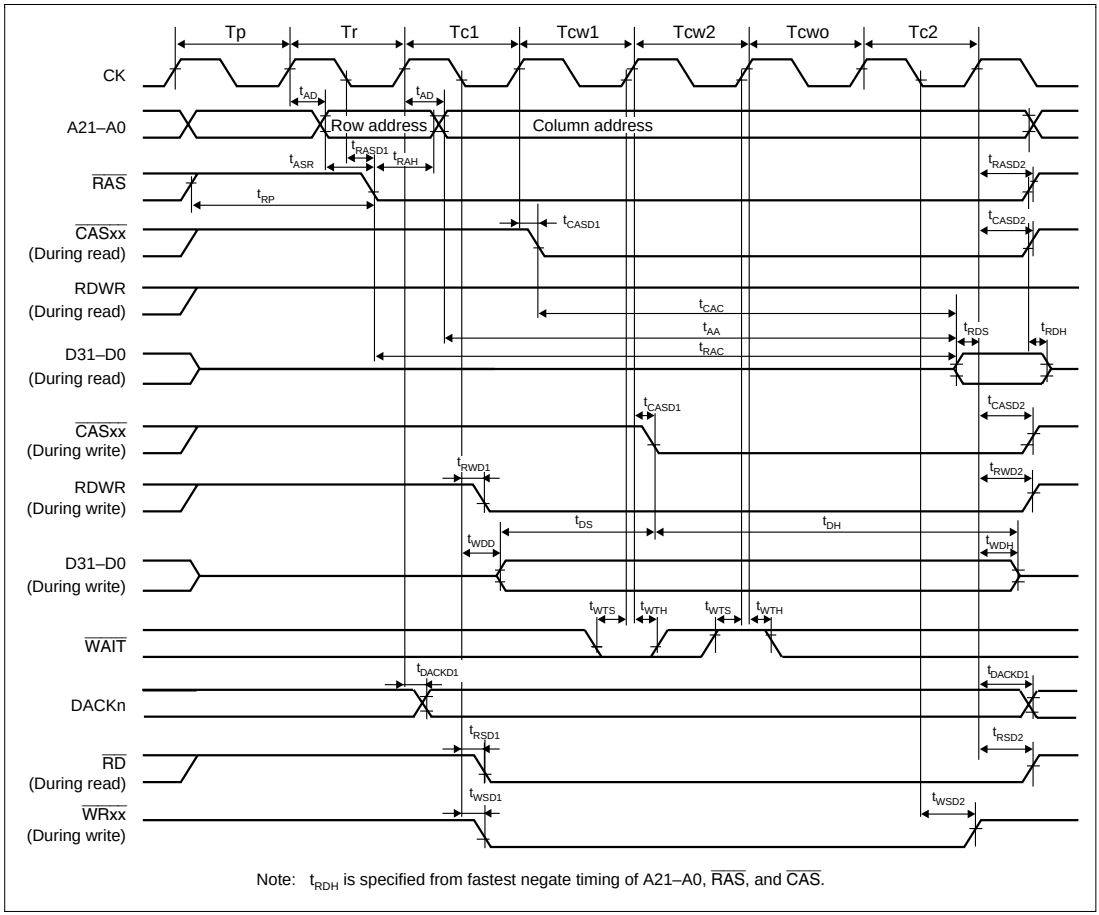
Note: t_{RDH} is specified from fastest negate timing of A21–A0, $\overline{RAS}$, and $\overline{CAS}$.

Figure 26.13 DRAM Cycle (Normal Mode, 2 Waits, TPC = 1, RCD = 1)



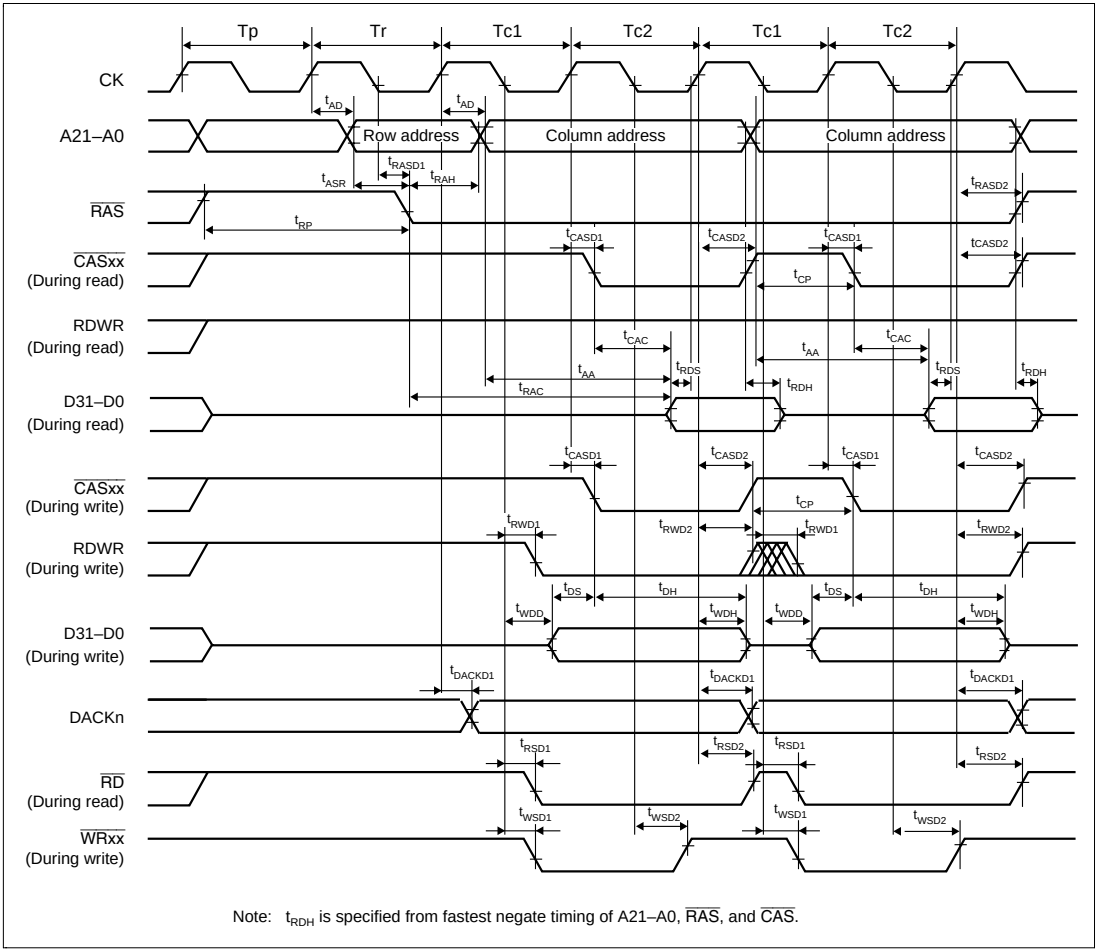
Note: t_{RDH} is specified from fastest negate timing of A21–A0, $\overline{RAS}$, and $\overline{CAS}$.

Figure 26.14 DRAM Cycle (Normal Mode, 3 Waits, TPC = 1, RCD = 1)



Note: t_{RDH} is specified from fastest negate timing of A21–A0, $\overline{RAS}$, and $\overline{CAS}$.

Figure 26.15 DRAM Cycle (Normal Mode, 2 Waits + Wait due to $\overline{WAIT}$ Signal)



Note: t_{RDH} is specified from fastest negate timing of A21–A0, $\overline{RAS}$, and $\overline{CAS}$.

Figure 26.16 DRAM Cycle (High-Speed Page Mode)

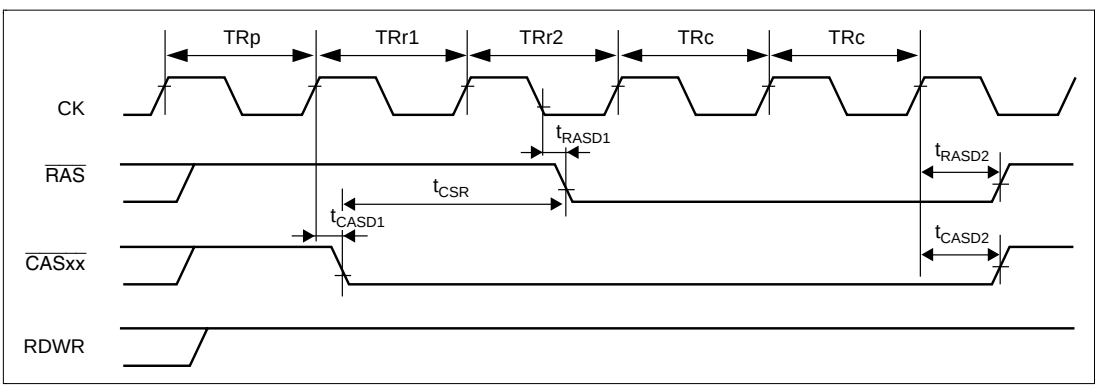


Figure 26.17 CAS Before RAS Refresh (TRAS1 = 0, TRAS0 = 0)

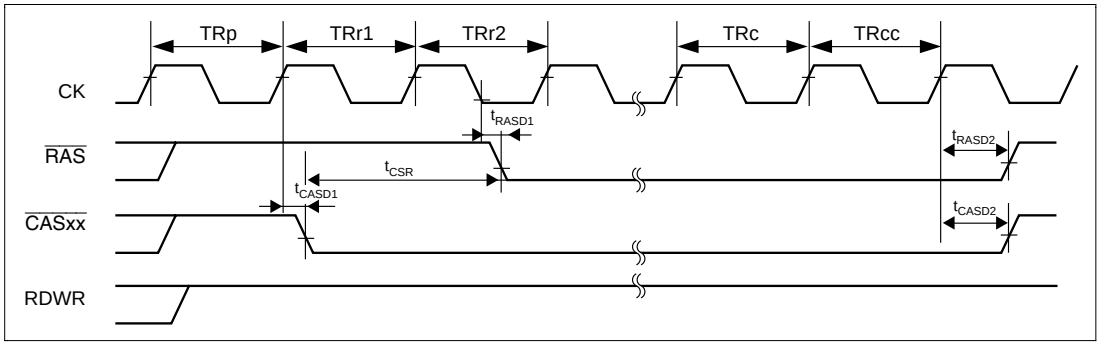


Figure 26.18 Self Refresh

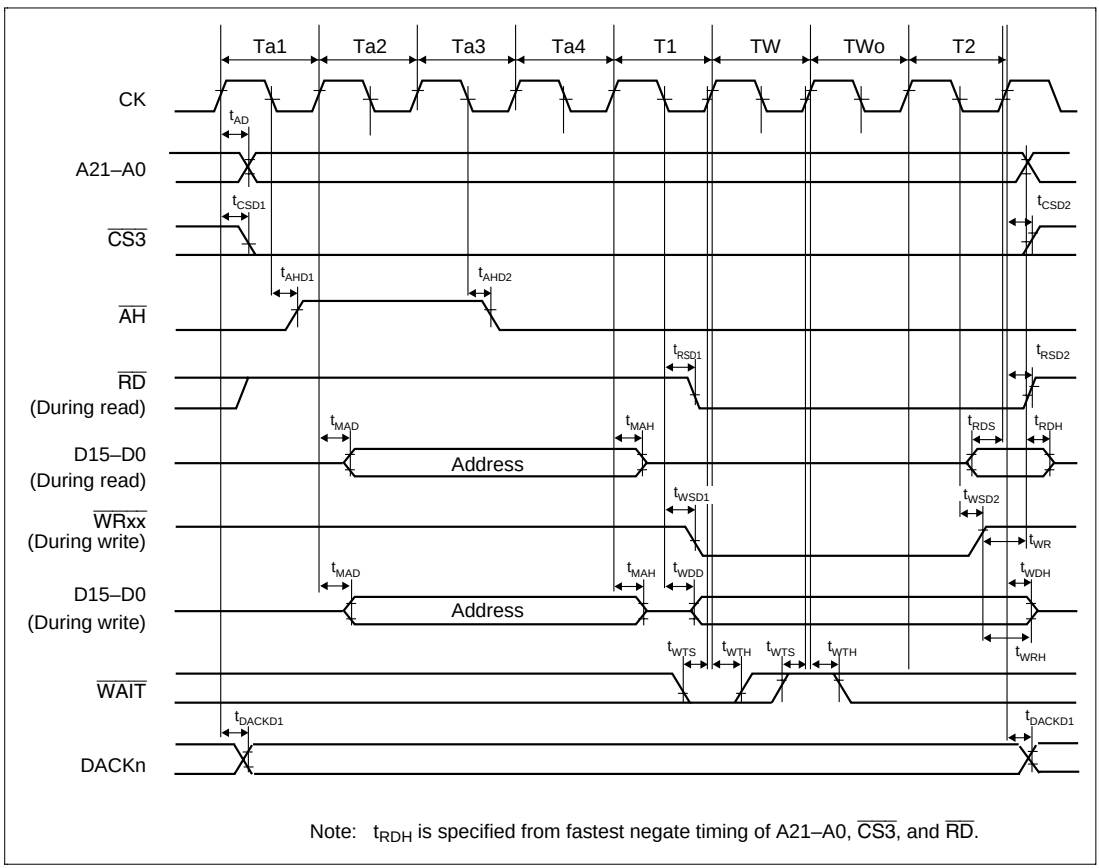


Figure 26.19 Address Data Multiplex I/O Space Cycle (1 Software Wait + External Wait)

26.3.4 Direct Memory Access Controller Timing

Table 26.8 Direct Memory Access Controller Timing (Conditions: V_{CC} = 3.0* to 3.6V, AV_{CC} = 3.0* to 3.6V, AV_{CC} = V_{CC} ± 10%, AV_{ref} = 3.0* to AV_{CC}, V_{SS} = AV_{SS} = 0V, T_a = –20 to +75°C)

Item	Symbol	Min	Max	Unit	Figure
$\overline{\text{DREQ0}}$, $\overline{\text{DREQ1}}$ setup time	t_{DRQS}	35	—	ns	26.20
$\overline{\text{DREQ0}}$, $\overline{\text{DREQ1}}$ hold time	t_{DRQH}	35	—	ns	
$\overline{\text{DREQ0}}$, $\overline{\text{DREQ1}}$ pulse width	t_{DRQW}	1.5	—	t_{cyc}	26.21
DRAK output delay time	t_{DRAKD}	—	35	ns	26.22

Note: * SH7042/43 ZTAT (excluding A mask) are 3.2V.

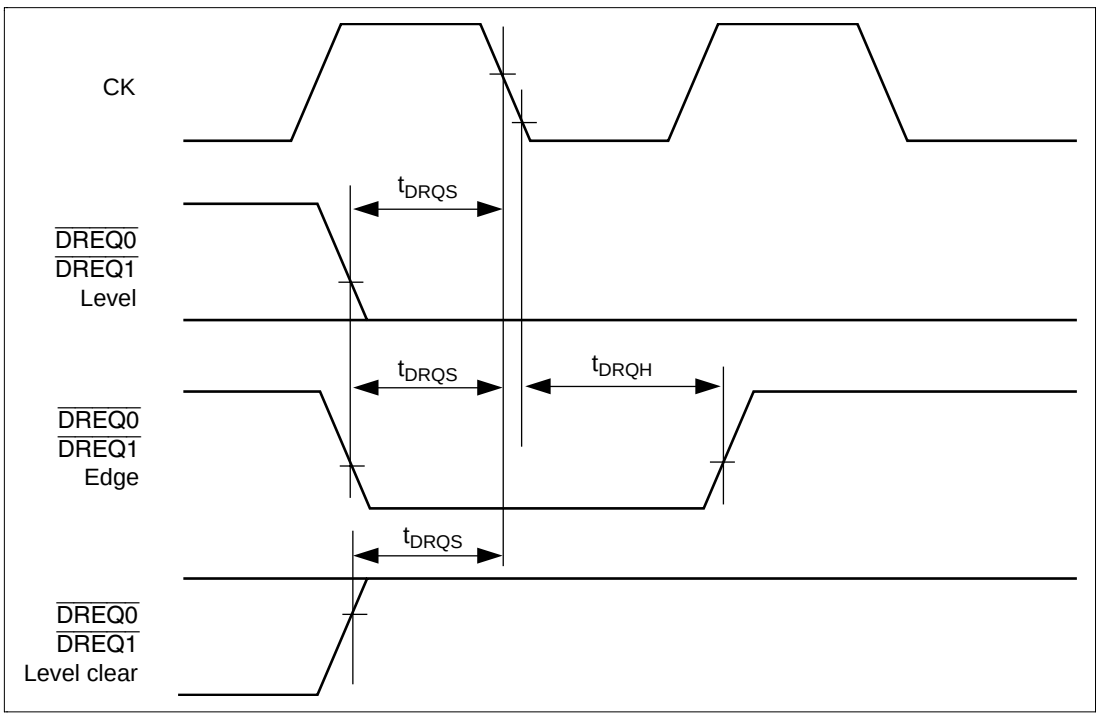


Figure 26.20 $\overline{\text{DREQ0}}$ and $\overline{\text{DREQ1}}$ Input Timing (1)

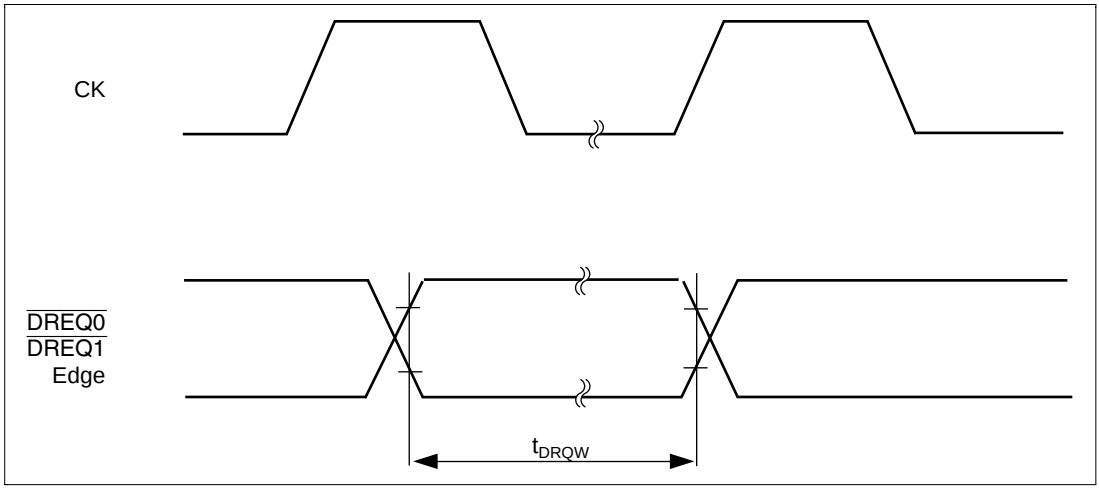


Figure 26.21 $\overline{\text{DREQ0}}$ and $\overline{\text{DREQ1}}$ Input Timing (2)

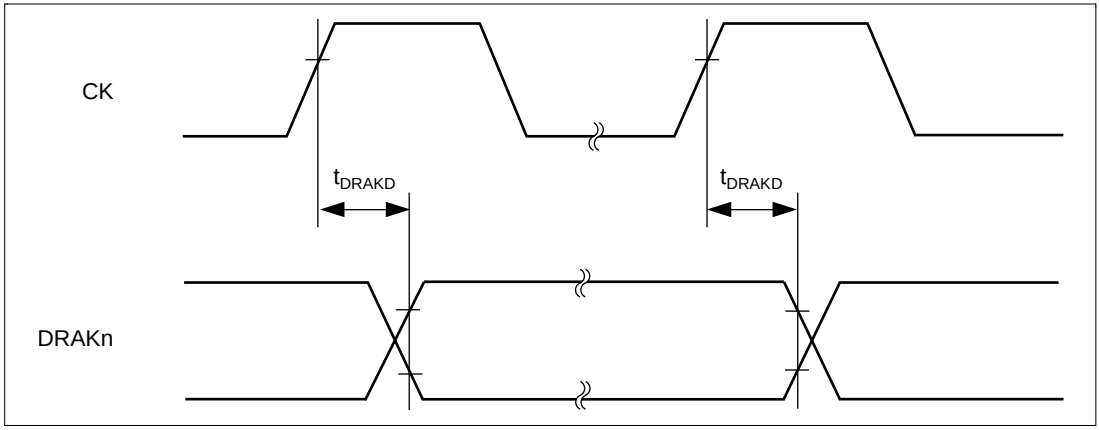


Figure 26.22 DRAK Output Delay Time

26.3.5 Multifunction Timer Pulse Unit Timing

Table 26.9 Multifunction Timer Pulse Unit Timing (Conditions: V_{CC} = 3.0* to 3.6V, AV_{CC} = 3.0* to 3.6V, AV_{CC} = V_{CC} ± 10%, AV_{ref} = 3.0* to AV_{CC}, V_{SS} = AV_{SS} = 0V, T_a = –20 to +75°C)

Item	Symbol	Min	Max	Unit	Figure
Output compare output delay time	t_{TOCD}	—	100	ns	26.23
Input capture input setup time	t_{TICS}	100	—	ns	
Timer input setup time	t_{TCKS}	100	—	ns	26.24
Timer clock pulse width (single edge specification)	$t_{TCKWH/L}$	1.5	—	t_{cyc}	
Timer clock pulse width (both edges specified)	$t_{TCKWH/L}$	2.5	—	t_{cyc}	
Timer clock pulse width (phase measurement mode)	$t_{TCKWH/L}$	2.5	—	t_{cyc}	

Note: * SH7042/43 ZTAT (excluding A mask) are 3.2V.

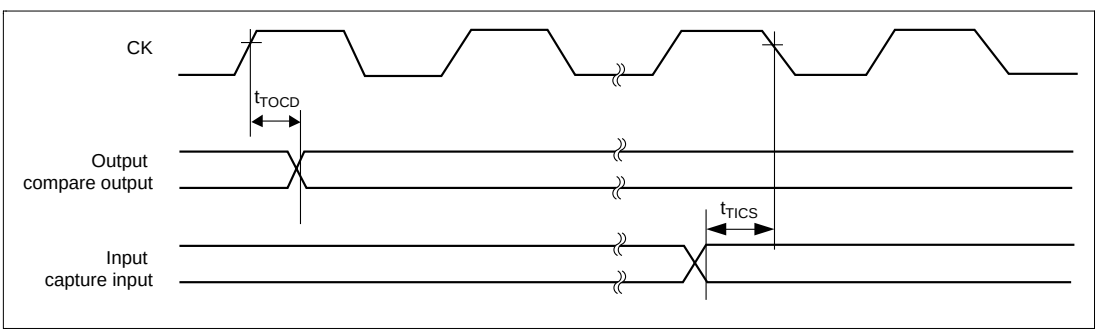


Figure 26.23 MTU I/O Timing

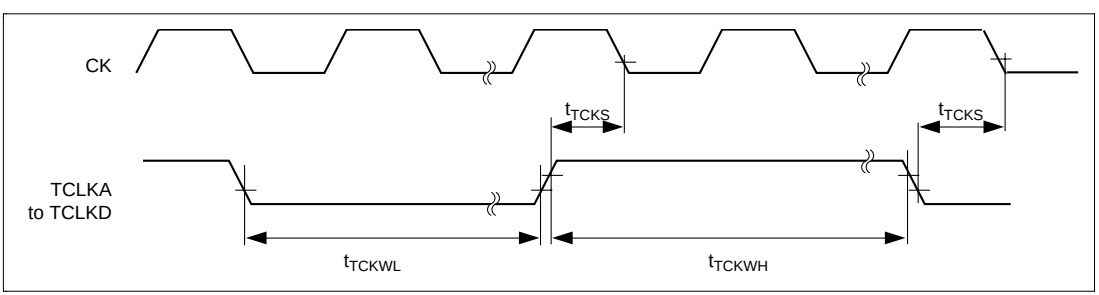


Figure 26.24 MTU Clock Input Timing

26.3.6 I/O Port Timing

Table 26.10 I/O Port Timing (Conditions: V_{CC} = 3.0* to 3.6V, AV_{CC} = 3.0* to 3.6V, AV_{CC} = V_{CC} ± 10%, AV_{ref} = 3.0* to AV_{CC}, V_{SS} = AV_{SS} = 0V, T_a = –20 to +75°C)

Item	Symbol	Min	Max	Unit	Figure
Port output data delay time	t_{PWD}	—	100	ns	26.25
Port input hold time	t_{PRH}	100	—	ns	
Port input setup time	t_{PRS}	100	—	ns	

Note: * SH7042/43 ZTAT (excluding A mask) are 3.2V.

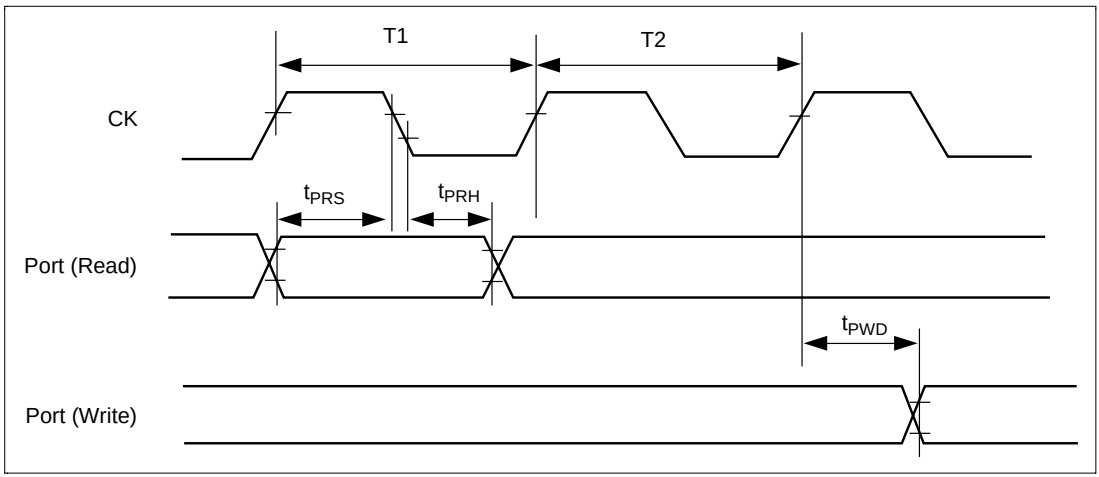


Figure 26.25 I/O Port I/O Timing

26.3.7 Watchdog Timer Timing

Table 26.11 Watchdog Timer Timing (Conditions: V_{CC} = 3.0* to 3.6V, AV_{CC} = 3.0* to 3.6V, AV_{CC} = V_{CC} ± 10%, AV_{ref} = 3.0* to AV_{CC}, V_{SS} = AV_{SS} = 0V, T_a = –20 to +75°C)

Item	Symbol	Min	Max	Unit	Figure
$\overline{\text{WDTOVF}}$ delay time	t_{WOVD}	—	100	ns	26.26

Note: * SH7042/43 ZTAT (excluding A mask) are 3.2V.

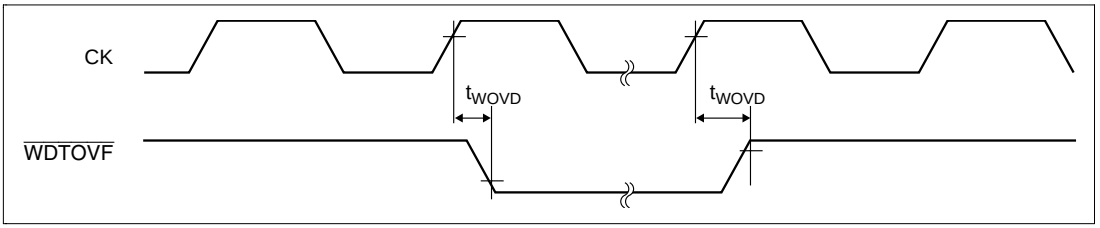


Figure 26.26 Watchdog Timer Timing

26.3.8 Serial Communication Interface Timing

Table 26.12 Serial Communication Interface Timing (Conditions: V_{CC} = 3.0* to 3.6V, AV_{CC} = 3.0* to 3.6V, AV_{CC} = V_{CC} ± 10%, AV_{ref} = 3.0* to AV_{CC}, V_{SS} = AV_{SS} = 0V, T_a = –20 to +75°C)

Item	Symbol	Min	Max	Unit	Figure
Input clock cycle	t_{scyc}	4	—	t_{cyc}	26.27
Input clock cycle (clock sync)	t_{scyc}	6	—	t_{cyc}	
Input clock pulse width	t_{sckw}	0.5	0.6	t_{scyc}	
Input clock rise time	t_{sckr}	—	1.5	t_{cyc}	
Input clock fall time	t_{sckf}	—	1.5	t_{cyc}	
Transmit data delay time (clock sync)	t_{TXD}	—	100	ns	26.28
Receive data setup time (clock sync)	t_{RXS}	100	—	ns	
Receive data hold time (clock sync)	t_{RXH}	100	—	ns	

Note: * SH7042/43 ZTAT (excluding A mask) are 3.2V.

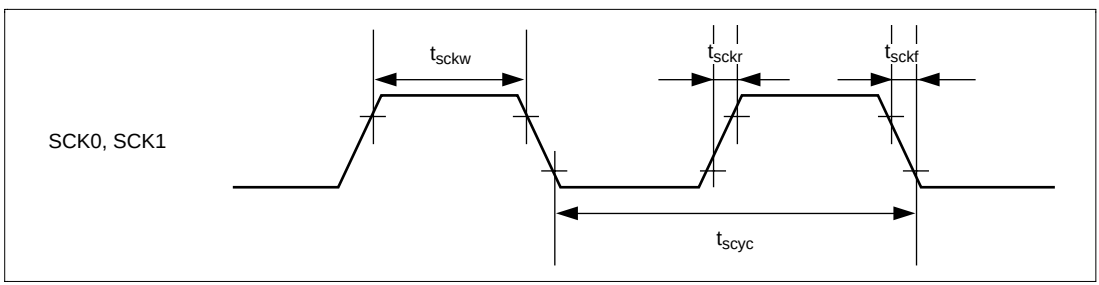


Figure 26.27 Input Clock Timing

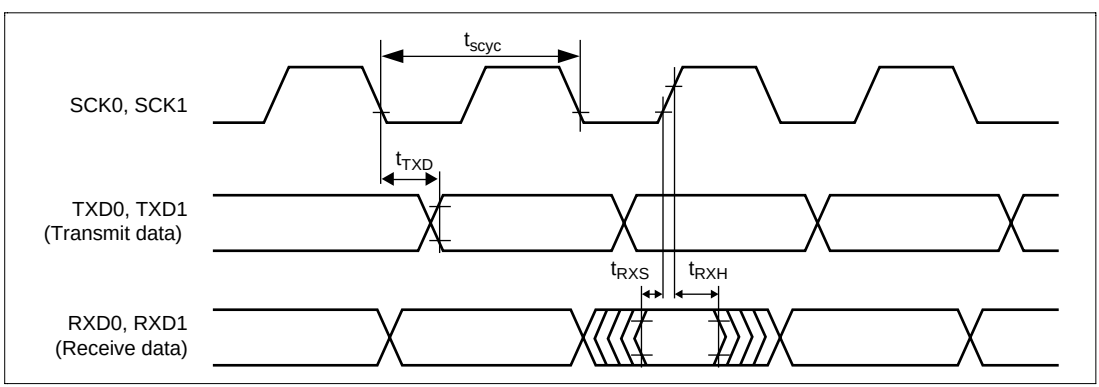


Figure 26.28 SCI I/O Timing (Clock Sync Mode)

26.3.9 High-speed A/D Converter Timing (excluding A mask)

Table 26.13 High-speed A/D Converter Timing (Conditions: V_{CC} = 3.0* to 3.6V, AV_{CC} = 3.0* to 3.6V, AV_{CC} = V_{CC} ± 10%, AV_{ref} = 3.0* to AV_{CC}, V_{SS} = AV_{SS} = 0V, T_a = –20 to +75°C)

Item		Symbol	Min	Typ	Max	Unit	Figure
External trigger input pulse width		t_{TRGW}	2	—	—	t_{cyc}	26.29
External trigger input start delay time		t_{TRGS}	50	—	—	ns	
A/D conversion start delay time	CKS = 0	t_D	1.5	1.5	1.5	t_{cyc}	26.30
	CKS = 1		1.5	1.5	1.5		
Input sampling time	CKS = 0	t_{SPL}	20	20	20		
	CKS = 1		40	40	40		
A/D conversion time	CKS = 0	t_{CONV}	42.5	42.5	42.5		
	CKS = 1		82.5	82.5	82.5		

Note: * SH7042/43 ZTAT (excluding A mask) are 3.2V.

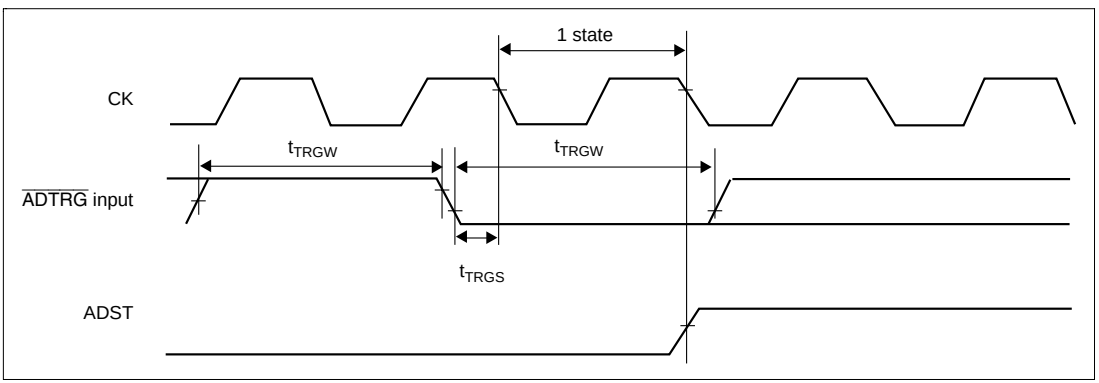


Figure 26.29 External Trigger Input Timing

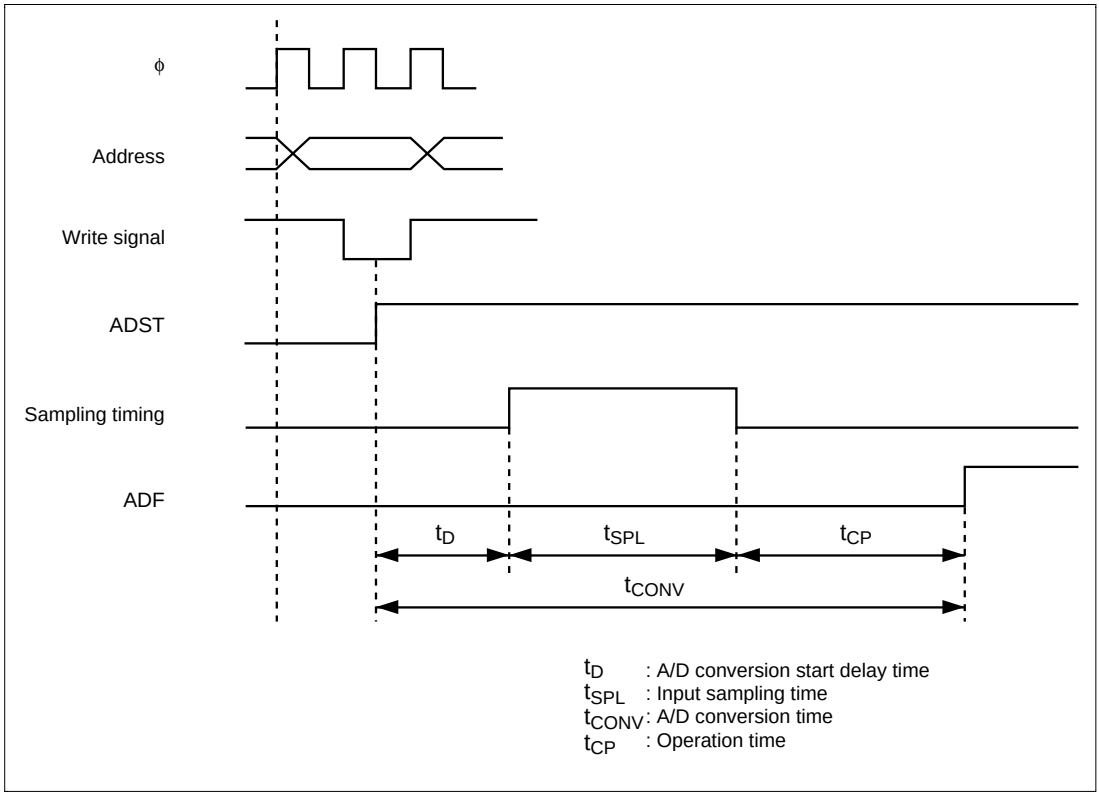


Figure 26.30 Analog Conversion Timing

26.3.10 Mid-speed Converter Timing (A mask)

Table 26.14 Mid-speed A/D Converter Timing (Conditions: V_{CC} = 3.0* to 3.6V, AV_{CC} = 3.0* to 3.6V, AV_{CC} = V_{CC} ± 10%, AV_{ref} = 3.0* to AV_{CC}, V_{SS} = AV_{SS} = 0V, T_a = –20 to +75°C)

Item		Symbol	Min	Typ	Max	Unit	Figure
External trigger input pulse width		t_{TRGW}	2	—	—	t_{cyc}	26.31
External trigger input start delay time		t_{TRGS}	50	—	—	ns	
A/D conversion start delay time	CKS = 0	t_D	10	—	17	t_{cyc}	26.32
	CKS = 1		6	—	9		
Input sampling time	CKS = 0	t_{SPL}	—	64	—		
	CKS = 1		—	32	—		
A/D conversion time	CKS = 0	t_{CONV}	259	—	266		
	CKS = 1		131	—	134		

Note: * SH7042/43 ZTAT (excluding A mask) are 3.2V.

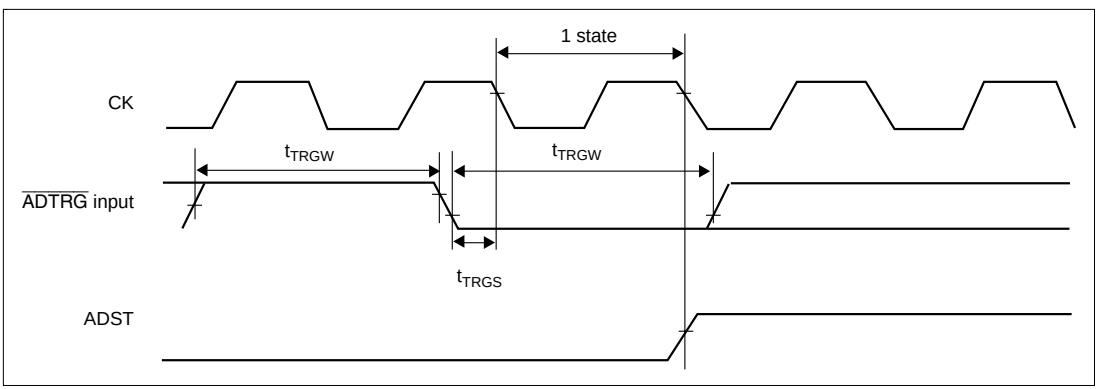
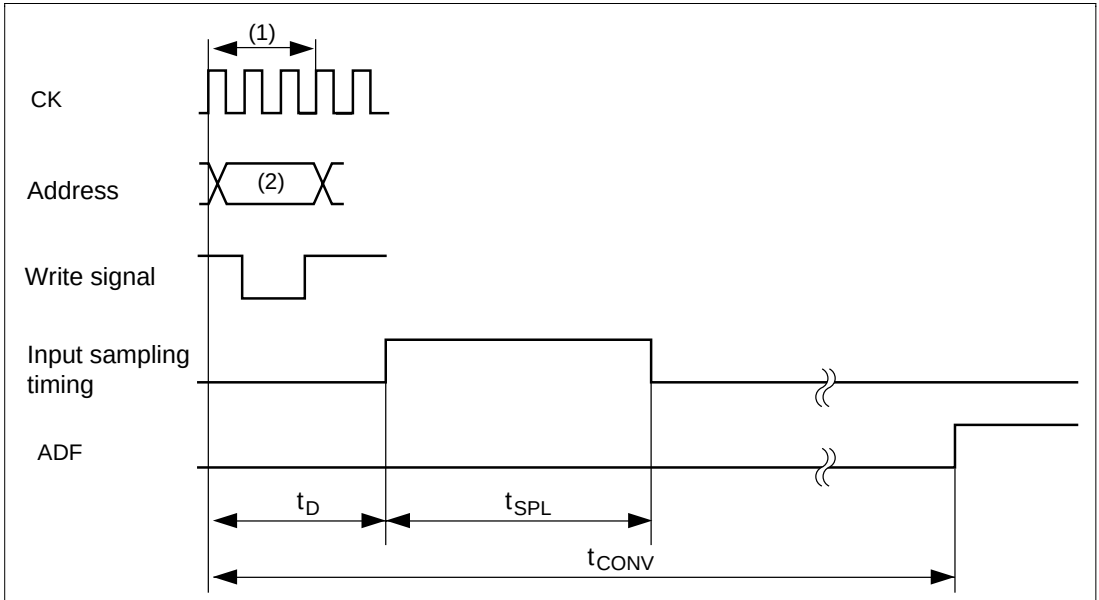


Figure 26.31 External Trigger Input Timing



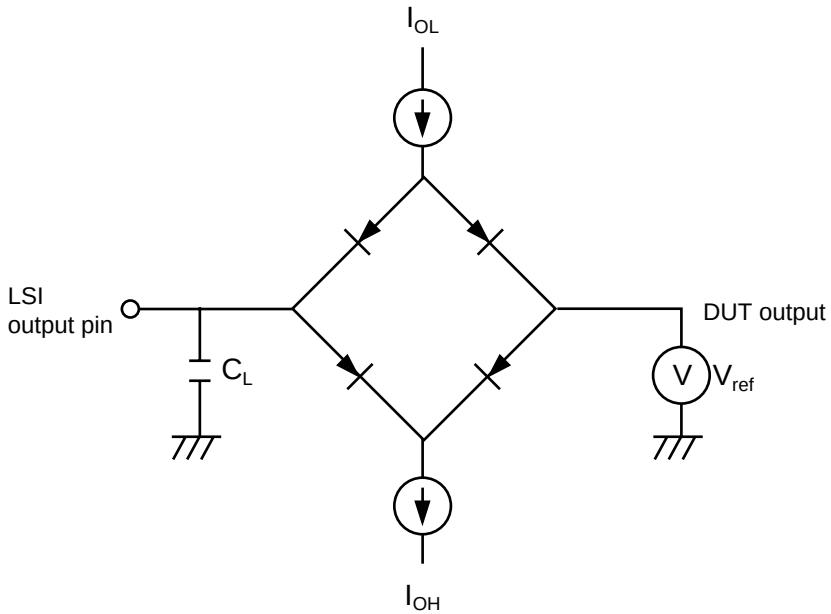
Legend:

(1) : ADCSR write cycle
(2) : ADCSR address
t_D : A/D conversion start delay time
t_{SPL} : Input sampling time
t_{CONV} : A/D conversion time

Figure 26.32 Analog Conversion Timing

26.3.11 Measurement Conditions for AC Characteristic

- Input reference levels:
  - — High level: 2.2 V
  - — Low level: 0.8 V
- Output reference levels:
  - — High level: 2.0 V
  - — Low level: 0.8 V



Note: C_L is set with the following pins, including the total capacitance of the measurement equipment etc:

30 pF:	CK, $\overline{RAS}$, $\overline{CASxx}$, RDWR, $\overline{CS0}$–$\overline{CS3}$, $\overline{AH}$, $\overline{BREQ}$, $\overline{BACK}$, DACK0, DACK1, and $\overline{IRQOUT}$
50 pF:	A21–A0, D31–D0, $\overline{RD}$, $\overline{WRxx}$
70 pF:	Port output and peripheral module output pins other than the above.
I_{OL}, I_{OH}:	See table 26.3, Permitted Output Current Values.

Figure 26.33 Output Load Circuit

26.4 A/D Converter Characteristics

Table 26.15 A/D Converter Characteristics (excluding A mask) (Conditions: V_{CC} = 3.0*1 to 3.6V, AV_{CC} = 3.0*1 to 3.6V, AV_{CC} = V_{CC} ± 10%, AV_{ref} = 3.0*1 to AV_{CC}, V_{SS} = AV_{SS} = 0V, T_a = –20 to +75°C)

	16.7MHz			
Item	**Min**	**Typ**	**Max**	**Unit**
Resolution	10	10	10	bit
Conversion time (when CKS = 1)	—	—	5	µs
Analog input capacity	—	—	20	pF
Permission signal source impedance	—	—	1	kΩ
Non-linearity error*2	—	—	± 15	LSB
Offset error*2	—	—	± 15	LSB
Full scale error*2	—	—	± 15	LSB
Quantize error*2	—	—	± 0.5	LSB
Absolute error	—	—	± 31	LSB

Notes: *1 SH7042/43 ZTAT (excluding A mask) are 3.2V.
*2 Reference values

Table 26.16 A/D Converter Characteristics (A mask) (Conditions: V_{CC} = 3.0*1 to 3.6V, AV_{CC} = 3.0*1 to 3.6V, AV_{CC} = V_{CC} ± 10%, AV_{ref} = 3.0*1 to AV_{CC}, V_{SS} = AV_{SS} = 0V, T_a = –20 to +75°C)

	16.7MHz			
Item	**min**	**typ**	**max**	**Unit**
Resolution	10	10	10	bit
Conversion time (when CKS = 0)	—	—	16.0	µs
Analog input capacity	—	—	20	pF
Permission signal source impedance	—	—	1	kΩ
Non-linearity error*2	—	—	±4	LSB
Offset error*2	—	—	±4	LSB
Full scale error*2	—	—	±4	LSB
Quantize error*2	—	—	±0.5	LSB
Absolute error	—	—	±6	LSB

Notes: *1 SH7042/43 ZTAT (excluding A mask) are 3.2V.
*2 Reference values

Appendix A On-Chip Supporting Module Registers

A.1 Addresses

Table A.1 On-Chip I/O Register Addresses

Address	Register Abbr.	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
—	DTMR	SM1	SM0	DM1	DM0	MD1	MD0	SZ1	SZ0	DTC
		DTS	CHNE	DISEL	NMIM	—	—	—	—	
—	DTSAR									
—	DTDAR									
—	DTIAR									
—	DTCRA									
—	DTCRB									
H'FFFF81A0	SMR0	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0	SCI
H'FFFF81A1	BRR0									
H'FFFF81A2	SCR0	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
H'FFFF81A3	TDR0									
H'FFFF81A4	SSR0	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT	
H'FFFF81A5	RDR0									
H'FFFF81A6 to H'FFFF81AF	—	—	—	—	—	—	—	—	—	

Table A.1 On-Chip I/O Register Addresses (cont)

Address	Register Abbr.	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFF81B0	SMR1	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0	SCI
H'FFFF81B1	BRR1									
H'FFFF81B2	SCR1	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
H'FFFF81B3	TDR1									
H'FFFF81B4	SSR1	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT	
H'FFFF81B5	RDR1									
H'FFFF81B6 to H'FFFF81FF	—	—	—	—	—	—	—	—	—	
H'FFFF8200	TCR3	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	MTU
H'FFFF8201	TCR4	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	
H'FFFF8202	TMDR3	—	—	BFB	BFA	MD3	MD2	MD1	MD0	
H'FFFF8203	TMDR4	—	—	BFB	BFA	MD3	MD2	MD1	MD0	
H'FFFF8204	TIOR3H	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
H'FFFF8205	TIOR3L	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0	
H'FFFF8206	TIOR4H	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
H'FFFF8207	TIOR4L	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0	
H'FFFF8208	TIER3	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA	
H'FFFF8209	TIER4	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA	
H'FFFF820A	TOER	—	—	OE4D	OE4C	OE3D	OE4B	OE4A	OE3B	
H'FFFF820B	TOCR	—	PSYE	—	—	—	—	OLSN	OLSP	
H'FFFF820C	—	—	—	—	—	—	—	—	—	
H'FFFF820D	TGCR	—	BDC	N	P	FB	WF	VF	UF	
H'FFFF820E	—	—	—	—	—	—	—	—	—	
H'FFFF820F	—	—	—	—	—	—	—	—	—	
H'FFFF8210	TCNT3									
H'FFFF8211										
H'FFFF8212	TCNT4									
H'FFFF8213										
H'FFFF8214	TCDR									
H'FFFF8215										
H'FFFF8216	TDDR									
H'FFFF8217										
H'FFFF8218	TGR3A									
H'FFFF8219										

Table A.1 On-Chip I/O Register Addresses (cont)

Address	Register Abbr.	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFF821A	TGR3B									MTU
H'FFFF821B										
H'FFFF821C	TGR4A									
H'FFFF821D										
H'FFFF821E	TGR4B									
H'FFFF821F										
H'FFFF8220	TCNTS									
H'FFFF8221										
H'FFFF8222	TCBR									
H'FFFF8223										
H'FFFF8224	TGR3C									
H'FFFF8225										
H'FFFF8226	TGR3D									
H'FFFF8227										
H'FFFF8228	TGR4C									
H'FFFF8229										
H'FFFF822A	TGR4D									
H'FFFF822B										
H'FFFF822C	TSR3	TCFD	—	—	TCFV	TGFD	TGFC	TGFB	TGFA	
H'FFFF822D	TSR4	TCFD	—	—	TCFV	TGFD	TGFC	TGFB	TGFA	
H'FFFF822E	—	—	—	—	—	—	—	—	—	
H'FFFF822F	—	—	—	—	—	—	—	—	—	
H'FFFF8230 to H'FFFF823F	—	—	—	—	—	—	—	—	—	
H'FFFF8240	TSTR	CST4	CST3	—	—	—	CST2	CST1	CST0	
H'FFFF8241	TSYR	SYNC4	SYNC3	—	—	—	SYNC2	SYNC1	SYNC0	
H'FFFF8242 to H'FFFF825F	—	—	—	—	—	—	—	—	—	
H'FFFF8260	TCR0	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	
H'FFFF8261	TMDR0	—	—	BFB	BFA	MD3	MD2	MD1	MD0	
H'FFFF8262	TIOR0H	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
H'FFFF8263	TIOR0L	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0	
H'FFFF8264	TIER0	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA	
H'FFFF8265	TSR0	—	—	—	TCFV	TGFD	TGFC	TGFB	TGFA	
H'FFFF8266	TCNT0									
H'FFFF8267										

Table A.1 On-Chip I/O Register Addresses (cont)

Address	Register Abbr.	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFF8268	TGR0A									MTU
H'FFFF8269										
H'FFFF826A	TGR0B									
H'FFFF826B										
H'FFFF826C	TGR0C									
H'FFFF826D										
H'FFFF826E	TGR0D									
H'FFFF826F										
H'FFFF8270 to H'FFFF827F	—	—	—	—	—	—	—	—	—	
H'FFFF8280	TCR1	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	
H'FFFF8281	TMDR1	—	—	—	—	MD3	MD2	MD1	MD0	
H'FFFF8282	TIOR1	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
H'FFFF8283	—	—	—	—	—	—	—	—	—	
H'FFFF8284	TIER1	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA	
H'FFFF8285	TSR1	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA	
H'FFFF8286	TCNT1									
H'FFFF8287										
H'FFFF8288	TGR1A									
H'FFFF8289										
H'FFFF828A	TGR1B									
H'FFFF828B										
H'FFFF828C to H'FFFF829F	—	—	—	—	—	—	—	—	—	
H'FFFF82A0	TCR2	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	
H'FFFF82A1	TMDR2	—	—	—	—	MD3	MD2	MD1	MD0	
H'FFFF82A2	TIOR2	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
H'FFFF82A3	—	—	—	—	—	—	—	—	—	
H'FFFF82A4	TIER2	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA	
H'FFFF82A5	TSR2	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA	
H'FFFF82A6	TCNT2									
H'FFFF82A7										
H'FFFF82A8	TGR2A									
H'FFFF82A9										
H'FFFF82AA	TGR2B									
H'FFFF82AB										

Table A.1 On-Chip I/O Register Addresses (cont)

Address	Register Abbr.	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFF82AC to H'FFFF8347	—	—	—	—	—	—	—	—	—	MTU
H'FFFF8348	IPRA									INTC
H'FFFF8349										
H'FFFF834A	IPRB									
H'FFFF834B										
H'FFFF834C	IPRC									
H'FFFF834D										
H'FFFF834E	IPRD									
H'FFFF834F										
H'FFFF8350	IPRE									
H'FFFF8351										
H'FFFF8352	IPRF									
H'FFFF8353										
H'FFFF8354	IPRG									
H'FFFF8355										
H'FFFF8356	IPRH									
H'FFFF8357										
H'FFFF8358	ICR	NMIL	—	—	—	—	—	—	NMIE	
H'FFFF8359		IRQ0S	IRQ1S	IRQ2S	IRQ3S	IRQ4S	IRQ5S	IRQ6S	IRQ7S	
H'FFFF835A	ISR	—	—	—	—	—	—	—	—	
H'FFFF835B		IRQ0F	IRQ1F	IRQ2F	IRQ3F	IRQ4F	IRQ5F	IRQ6F	IRQ7F	
H'FFFF835C to H'FFFF837F	—	—	—	—	—	—	—	—	—	
H'FFFF8380	PADRH	—	—	—	—	—	—	—	—	I/O
H'FFFF8381		PA23DR	PA22DR	PA21DR	PA20DR	PA19DR	PA18DR	PA17DR	PA16DR	
H'FFFF8382	PADRL	PA15DR	PA14DR	PA13DR	PA12DR	PA11DR	PA10DR	PA9DR	PA8DR	
H'FFFF8383		PA7DR	PA6DR	PA5DR	PA4DR	PA3DR	PA2DR	PA1DR	PA0DR	
H'FFFF8384	PAIORH	—	—	—	—	—	—	—	—	PFC
H'FFFF8385		PA23IOR	PA22IOR	PA21IOR	PA20IOR	PA19IOR	PA18IOR	PA17IOR	PA16IOR	
H'FFFF8386	PAIORL	PA15IOR	PA14IOR	PA13IOR	PA12IOR	PA11IOR	PA10IOR	PA9IOR	PA8IOR	
H'FFFF8387		PA7IOR	PA6IOR	PA5IOR	PA4IOR	PA3IOR	PA2IOR	PA1IOR	PA0IOR	
H'FFFF8388	PACRH	—	PA23MD	—	PA22MD	—	PA21MD	—	PA20MD	
H'FFFF8389		PA19MD1	PA19MD0	PA18MD1	PA18MD0	—	PA17MD	—	PA16MD	

Table A.1 On-Chip I/O Register Addresses (cont)

Address	Register Abbr.	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFF838A	—	—	—	—	—	—	—	—	—	PFC
H'FFFF838B	—	—	—	—	—	—	—	—	—	
H'FFFF838C	PACRL1	—	PA15MD	—	PA14MD	—	PA13MD	—	PA12MD	
H'FFFF838D		—	PA11MD	—	PA10MD	PA9MD1	PA9MD0	PA8MD1	PA8MD0	
H'FFFF838E	PACRL2	PA7MD1	PA7MD0	PA6MD1	PA6MD0	PA5MD1	PA5MD0	—	PA4MD	
H'FFFF838F		—	PA3MD	PA2MD1	PA2MD0	—	PA1MD	—	PA0MD	
H'FFFF8390	PBDR	—	—	—	—	—	—	PB9DR	PB8DR	I/O
H'FFFF8391		PB7DR	PB6DR	PB5DR	PB4DR	PB3DR	PB2DR	PB1DR	PB0DR	
H'FFFF8392	PCDR	PC15DR	PC14DR	PC13DR	PC12DR	PC11DR	PC10DR	PC9DR	PC8DR	
H'FFFF8393		PC7DR	PC6DR	PC5DR	PC4DR	PC3DR	PC2DR	PC1DR	PC0DR	
H'FFFF8394	PBIOR	—	—	—	—	—	—	PB9IOR	PB8IOR	PFC
H'FFFF8395		PB7IOR	PB6IOR	PB5IOR	PB4IOR	PB3IOR	PB2IOR	PB1IOR	PB0IOR	
H'FFFF8396	PCIOR	PC15IOR	PC14IOR	PC13IOR	PC12IOR	PC11IOR	PC10IOR	PC9IOR	PC8IOR	
H'FFFF8397		PC7IOR	PC6IOR	PC5IOR	PC4IOR	PC3IOR	PC2IOR	PC1IOR	PC0IOR	
H'FFFF8398	PBCR1	—	—	—	—	—	—	—	—	
H'FFFF8399		—	—	—	—	PB9MD1	PB9MD0	PB8MD1	PB8MD0	
H'FFFF839A	PBCR2	PB7MD1	PB7MD0	PB6MD1	PB6MD0	PB5MD1	PB5MD0	PB4MD1	PB4MD0	
H'FFFF839B		PB3MD1	PB3MD0	PB2MD1	PB2MD0	—	PB1MD	—	PB0MD	
H'FFFF839C	PCCR	PC15MD	PC14MD	PC13MD	PC12MD	PC11MD	PC10MD	PC9MD	PC8MD	
H'FFFF839D		PC7MD	PC6MD	PC5MD	PC4MD	PC3MD	PC2MD	PC1MD	PC0MD	
H'FFFF839E	—	—	—	—	—	—	—	—	—	
H'FFFF839F	—	—	—	—	—	—	—	—	—	
H'FFFF83A0	PDDRH	PD31DR	PD30DR	PD29DR	PD28DR	PD27DR	PD26DR	PD25DR	PD24DR	I/O
H'FFFF83A1		PD23DR	PD22DR	PD21DR	PD20DR	PD19DR	PD18DR	PD17DR	PD16DR	
H'FFFF83A2	PDDRL	PD15DR	PD14DR	PD13DR	PD12DR	PD11DR	PD10DR	PD9DR	PD8DR	
H'FFFF83A3		PD7DR	PD6DR	PD5DR	PD4DR	PD3DR	PD2DR	PD1DR	PD0DR	
H'FFFF83A4	PDIORH	PD31IOR	PD30IOR	PD29IOR	PD28IOR	PD27IOR	PD26IOR	PD25IOR	PD24IOR	PFC
H'FFFF83A5		PD23IOR	PD22IOR	PD21IOR	PD20IOR	PD19IOR	PD18IOR	PD17IOR	PD16IOR	
H'FFFF83A6	PDIORL	PD15IOR	PD14IOR	PD13IOR	PD12IOR	PD11IOR	PD10IOR	PD9IOR	PD8IOR	
H'FFFF83A7		PD7IOR	PD6IOR	PD5IOR	PD4IOR	PD3IOR	PD2IOR	PD1IOR	PD0IOR	
H'FFFF83A8	PDCRH1	PD31MD1	PD31MD0	PD30MD1	PD30MD0	PD29MD1	PD29MD0	PD28MD1	PD28MD0	
H'FFFF83A9		PD27MD1	PD27MD0	PD26MD1	PD26MD0	PD25MD1	PD25MD0	PD24MD1	PD24MD0	
H'FFFF83AA	PDCRH2	PD23MD1	PD23MD0	PD22MD1	PD22MD0	PD21MD1	PD21MD0	PD20MD1	PD20MD0	
H'FFFF83AB		PD19MD1	PD19MD0	PD18MD1	PD18MD0	PD17MD1	PD17MD0	PD16MD1	PD16MD0	

Table A.1 On-Chip I/O Register Addresses (cont)

Address	Register Abbr.	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFF83AC	PDCRL	PD15MD	PD14MD	PD13MD	PD12MD	PD11MD	PD10MD	PD9MD	PD8MD	PFC
H'FFFF83AD		PD7MD	PD6MD	PD5MD	PD4MD	PD3MD	PD2MD	PD1MD	PD0MD	
H'FFFF83AE	—	—	—	—	—	—	—	—	—	
H'FFFF83AF	—	—	—	—	—	—	—	—	—	
H'FFFF83B0	PEDR	PE15DR	PE14DR	PE13DR	PE12DR	PE11DR	PE10DR	PE9DR	PE8DR	I/O
H'FFFF83B1		PE7DR	PE6DR	PE5DR	PE4DR	PE3DR	PE2DR	PE1DR	PE0DR	
H'FFFF83B2	PFDR	—	—	—	—	—	—	—	—	
H'FFFF83B3		PF7DR	PF6DR	PF5DR	PF4DR	PF3DR	PF2DR	PF1DR	PF0DR	
H'FFFF83B4	PEIOR	PE15IOR	PE14IOR	PE13IOR	PE12IOR	PE11IOR	PE10IOR	PE9IOR	PE8IOR	PFC
H'FFFF83B5		PE7IOR	PE6IOR	PE5IOR	PE4IOR	PE3IOR	PE2IOR	PE1IOR	PE0IOR	
H'FFFF83B6	—	—	—	—	—	—	—	—	—	
H'FFFF83B7	—	—	—	—	—	—	—	—	—	
H'FFFF83B8	PECR1	PE15MD1	PE15MD0	PE14MD1	PE14MD0	PE13MD1	PE13MD0	—	PE12MD	
H'FFFF83B9		—	PE11MD	—	PE10MD	—	PE9MD	—	PE8MD	
H'FFFF83BA	PECR2	—	PE7MD	—	PE6MD	—	PE5MD	—	PE4MD	
H'FFFF83BB		PE3MD1	PE3MD0	PE2MD1	PE2MD0	PE1MD1	PE1MD0	PE0MD1	PE0MD0	
H'FFFF83BC to H'FFFF83BF	—	—	—	—	—	—	—	—	—	—
H'FFFF83C0	ICSR	POE3F	POE2F	POE1F	POE0F	—	—	—	PIE	MTU
H'FFFF83C1		POE3M1	POE3M0	POE2M1	POE2M0	POE1M1	POE1M0	POE0M1	POE0M0	
H'FFFF83C2	OCSR	OSF	—	—	—	—	—	OCE	OIE	
H'FFFF83C3		—	—	—	—	—	—	—	—	
H'FFFF83C4 to H'FFFF83C7	—	—	—	—	—	—	—	—	—	—
H'FFFF83C8	IFCR	—	—	—	—	—	—	—	—	PFC
H'FFFF83C9		—	—	—	—	IRQMD3	IRQMD2	IRQMD1	IRQMD0	
H'FFFF83CA to H'FFFF83CF	—	—	—	—	—	—	—	—	—	—
H'FFFF83D0	CMSTR	—	—	—	—	—	—	—	—	CMT
H'FFFF83D1		—	—	—	—	—	—	STR1	STR0	
H'FFFF83D2	CMCSR0	—	—	—	—	—	—	—	—	
H'FFFF83D3		CMF	CMIE	—	—	—	—	CKS1	CKS0	
H'FFFF83D4	CMCNT0									
H'FFFF83D5										

Table A.1 On-Chip I/O Register Addresses (cont)

Address	Register Abbr.	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFF83D6	CMCOR0									CMT
H'FFFF83D7										
H'FFFF83D8	CMCSR1	—	—	—	—	—	—	—	—	
H'FFFF83D9		CMF	CMIE	—	—	—	—	CKS1	CKS0	
H'FFFF83DA	CMCNT1									
H'FFFF83DB										
H'FFFF83DC	CMCOR1									
H'FFFF83DD										
H'FFFF83DE	—	—	—	—	—	—	—	—	—	
H'FFFF83DF	—	—	—	—	—	—	—	—	—	
H'FFFF83E0	ADCSR	ADF	ADIE	ADST	CKS	GRP	CH2	CH1	CH0	A/D
H'FFFF83E1	ADCR	—	PWR	TRGS1	TRGS0	SCAN	DSMP	BUFE1	BUFE0	(High speed) (Excl. A mask)
H'FFFF83E2 to H'FFFF83EF	—	—	—	—	—	—	—	—	—	
H'FFFF83F0	ADDRA	—	—	—	—	—	—	AD9	AD8	
H'FFFF83F1		AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	
H'FFFF83F2	ADDRB	—	—	—	—	—	—	AD9	AD8	
H'FFFF83F3		AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	
H'FFFF83F4	ADDRC	—	—	—	—	—	—	AD9	AD8	
H'FFFF83F5		AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	
H'FFFF83F6	ADDRD	—	—	—	—	—	—	AD9	AD8	
H'FFFF83F7		AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	
H'FFFF83F8	ADDRE	—	—	—	—	—	—	AD9	AD8	
H'FFFF83F9		AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	
H'FFFF83FA	ADDRF	—	—	—	—	—	—	AD9	AD8	
H'FFFF83FB		AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	
H'FFFF83FC	ADDRG	—	—	—	—	—	—	AD9	AD8	
H'FFFF83FD		AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	
H'FFFF83FE	ADDRH	—	—	—	—	—	—	AD9	AD8	
H'FFFF83FF		AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	
H'FFFF8400	ADDRA0	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	A/D(Mid-
H'FFFF8401		AD1	AD0	—	—	—	—	—	—	speed)
H'FFFF8402	ADDRB0	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	(A mask
H'FFFF8403		AD1	AD0	—	—	—	—	—	—	only)
H'FFFF8404	ADDRC0	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
H'FFFF8405		AD1	AD0	—	—	—	—	—	—	

Table A.1 On-Chip I/O Register Addresses (cont)

Address	Register Abbr.	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFF8406	ADDRD0	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	A/D(Mid-
H'FFFF8407		AD1	AD0	—	—	—	—	—	—	speed)
H'FFFF8408	ADDRA1	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	(A mask
H'FFFF8409		AD1	AD0	—	—	—	—	—	—	only)
H'FFFF840A	ADDRB1	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
H'FFFF840B		AD1	AD0	—	—	—	—	—	—	
H'FFFF840C	ADDRC1	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
H'FFFF840D		AD1	AD0	—	—	—	—	—	—	
H'FFFF840E	ADDRD1	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
H'FFFF840F		AD1	AD0	—	—	—	—	—	—	
H'FFFF8410	ADCSR0	ADF	ADIE	ADST	SCAN	CKS	CH2	CH1	CH0	
H'FFFF8411	ADCSR1	ADF	ADIE	ADST	SCAN	CKS	CH2	CH1	CH0	
H'FFFF8412	AADCR0	TRGE	—	—	—	—	—	—	—	
H'FFFF8413	AADCR1	TRGE	—	—	—	—	—	—	—	
H'FFFF8414 to H'FFFF857F	—	—	—	—	—	—	—	—	—	
H'FFFF8580	FLMCR1	FWE	SWE	ESU1	PSU1	EV1	PV1	E1	P1	FLASH
H'FFFF8581	FLMCR2	FLER	—	ESU2	PSU2	EV2	PV2	E2	P2	(F-ZTAT
H'FFFF8582	EBR1	—	—	—	—	EB3	EB2	EB1	EB0	version-
H'FFFF8583	EBR2	EB11	EB10	EB9	EB8	EB7	EB6	EB5	EB4	only)
H'FFFF8584 to H'FFFF859F	—	—	—	—	—	—	—	—	—	
H'FFFF8600	UBARH	UBA31	UBA30	UBA29	UBA28	UBA27	UBA26	UBA25	UBA24	UBC
H'FFFF8601		UBA23	UBA22	UBA21	UBA20	UBA19	UBA18	UBA17	UBA16	
H'FFFF8602	UBARL	UBA15	UBA14	UBA13	UBA12	UBA11	UBA10	UBA9	UBA8	
H'FFFF8603		UBA7	UBA6	UBA5	UBA4	UBA3	UBA2	UBA1	UBA0	
H'FFFF8604	UBAMRH	UBM31	UBM30	UBM29	UBM28	UBM27	UBM26	UBM25	UBM24	
H'FFFF8605		UBM23	UBM22	UBM21	UBM20	UBM19	UBM18	UBM17	UBM16	
H'FFFF8606	UBAMRL	UBM15	UBM14	UBM13	UBM12	UBM11	UBM10	UBM9	UBM8	
H'FFFF8607		UBM7	UBM6	UBM5	UBM4	UBM3	UBM2	UBM1	UBM0	
H'FFFF8608	UBBR	—	—	—	—	—	—	—	—	
H'FFFF8609		CP1	CP0	ID1	ID0	RW1	RW0	SZ1	SZ0	
H'FFFF860A to H'FFFF860F	—	—	—	—	—	—	—	—	—	
H'FFFF8610	TCSR	OVF	WT/IT	TME	—	—	CKS2	CKS1	CKS0	WDT

Table A.1 On-Chip I/O Register Addresses (cont)

Address	Register Abbr.	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFF8610	TCNT*1									WDT
H'FFFF8611	TCNT*2									
H'FFFF8612	RSTCSR*1	WOVF	RSTE	RSTS	—	—	—	—	—	
H'FFFF8613	RSTCSR*2	WOVF	RSTE	RSTS	—	—	—	—	—	
H'FFFF8614	SBYCR	SBY	HIZ	—	—	—	—	—	—	Power-down state
H'FFFF8615 to H'FFFF861F	—	—	—	—	—	—	—	—	—	BSC
H'FFFF8620	BCR1	—	—	MTURWE	—	—	—	—	IOE	
H'FFFF8621		A3LG	A2LG	A1LG	A0LG	A3SZ	A2SZ	A1SZ	A0SZ	
H'FFFF8622	BCR2	IW31	IW30	IW21	IW20	IW11	IW10	IW01	IW00	
H'FFFF8623		CW3	CW2	CW1	CW0	SW3	SW2	SW1	SW0	
H'FFFF8624	WCR1	W33	W32	W31	W30	W23	W22	W21	W20	
H'FFFF8625		W13	W12	W11	W10	W03	W02	W01	W00	
H'FFFF8626	WCR2	—	—	—	—	—	—	—	—	
H'FFFF8627		—	—	DDW1	DDW0	DSW3	DSW2	DSW1	DSW0	
H'FFFF8628	RAMER	—	—	—	—	—	—	—	—	FLASH (F-ZTAT
H'FFFF8629		—	—	—	—	—	RAMS	RAM1	RAM0	version only)
H'FFFF862A	DCR	TPC	RCD	TRAS1	TRAS0	DWW1	DWW0	DWR1	DWR0	BSC
H'FFFF862B		DIW	—	BE	RASD	SZ1	SZ0	AMX1	AMX0	
H'FFFF862C	RTCSR	—	—	—	—	—	—	—	—	
H'FFFF862D		—	CMF	CMIE	CKS2	CKS1	CKS0	RFSH	RMD	
H'FFFF862E	RTCNT	—	—	—	—	—	—	—	—	
H'FFFF862F										
H'FFFF8630	RTCOR	—	—	—	—	—	—	—	—	
H'FFFF8631										

Notes: *1 Write address.
*2 Read address. For details, see section 13.2.4, Register Access, in section 13, Watchdog Timer (WDT).

Table A.1 On-Chip I/O Register Addresses (cont)

Address	Register Abbr.	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFF8632 to H'FFFF86AF	—	—	—	—	—	—	—	—	—	BSC
H'FFFF86B0	DMAOR	—	—	—	—	—	—	PR1	PR0	DMAC
H'FFFF86B1		—	—	—	—	—	AE	NMIF	DME	
H'FFFF86B2 to H'FFFF86BF	—	—	—	—	—	—	—	—	—	
H'FFFF86C0	SAR0									
H'FFFF86C1										
H'FFFF86C2										
H'FFFF86C3										
H'FFFF86C4	DAR0									
H'FFFF86C5										
H'FFFF86C6										
H'FFFF86C7										
H'FFFF86C8	DMATCR0	—	—	—	—	—	—	—	—	
H'FFFF86C9										
H'FFFF86CA										
H'FFFF86CB										
H'FFFF86CC	CHCR0	—	—	—	—	—	—	—	—	
H'FFFF86CD		—	—	—	DI	RO	RL	AM	AL	
H'FFFF86CE		DM1	DM0	SM1	SM0	RS3	RS2	RS1	RS0	
H'FFFF86CF		—	DS	TM	TS1	TS0	IE	TE	DE	
H'FFFF86D0	SAR1									
H'FFFF86D1										
H'FFFF86D2										
H'FFFF86D3										
H'FFFF86D4	DAR1									
H'FFFF86D5										
H'FFFF86D6										
H'FFFF86D7										
H'FFFF86D8	DMATCR1	—	—	—	—	—	—	—	—	
H'FFFF86D9										
H'FFFF86DA										
H'FFFF86DB										

Table A.1 On-Chip I/O Register Addresses (cont)

Address	Register Abbr.	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFF86DC	CHCR1	—	—	—	—	—	—	—	—	DMAC
H'FFFF86DD		—	—	—	DI	RO	RL	AM	AL	
H'FFFF86DE		DM1	DM0	SM1	SM0	RS3	RS2	RS1	RS0	
H'FFFF86DF		—	DS	TM	TS1	TS0	IE	TE	DE	
H'FFFF86E0	SAR2									
H'FFFF86E1										
H'FFFF86E2										
H'FFFF86E3										
H'FFFF86E4	DAR2									
H'FFFF86E5										
H'FFFF86E6										
H'FFFF86E7										
H'FFFF86E8	DMATCR2	—	—	—	—	—	—	—	—	
H'FFFF86E9										
H'FFFF86EA										
H'FFFF86EB										
H'FFFF86EC	CHCR2	—	—	—	—	—	—	—	—	
H'FFFF86ED		—	—	—	DI	RO	RL	AM	AL	
H'FFFF86EE		DM1	DM0	SM1	SM0	RS3	RS2	RS1	RS0	
H'FFFF86EF		—	DS	TM	TS1	TS0	IE	TE	DE	
H'FFFF86F0	SAR3									
H'FFFF86F1										
H'FFFF86F2										
H'FFFF86F3										
H'FFFF86F4	DAR3									
H'FFFF86F5										
H'FFFF86F6										
H'FFFF86F7										
H'FFFF86F8	DMATCR3	—	—	—	—	—	—	—	—	
H'FFFF86F9										
H'FFFF86FA										
H'FFFF86FB										
H'FFFF86FC	CHCR3	—	—	—	—	—	—	—	—	
H'FFFF86FD		—	—	—	DI	RO	RL	AM	AL	
H'FFFF86FE		DM1	DM0	SM1	SM0	RS3	RS2	RS1	RS0	
H'FFFF86FF		—	DS	TM	TS1	TS0	IE	TE	DE	

Table A.1 On-Chip I/O Register Addresses (cont)

Address	Register Abbr.	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFF8700	DTEA	DTE7	DTE6	DTE5	DTE4	DTE3	DTE2	DTE1	DTE0	DTC
H'FFFF8701	DTEB	DTE7	DTE6	DTE5	DTE4	DTE3	DTE2	DTE1	DTE0	
H'FFFF8702	DTEC	DTE7	DTE6	DTE5	DTE4	DTE3	DTE2	DTE1	DTE0	
H'FFFF8703	DTED	DTE7	DTE6	DTE5	DTE4	DTE3	DTE2	DTE1	DTE0	
H'FFFF8704	DTEE	DTE7	DTE6	DTE5	DTE4	DTE3	DTE2	DTE1	DTE0	
H'FFFF8705	—	—	—	—	—	—	—	—	—	
H'FFFF8706	DTCSR	—	—	—	—	—	NMIF	AE	SWDTE	
H'FFFF8707		DTVEC7	DTVEC6	DTVEC5	DTVEC4	DTVEC3	DTVEC2	DTVEC1	DTVEC0	
H'FFFF8708	DTBR									
H'FFFF8709										
H'FFFF870A to H'FFFF873F	—	—	—	—	—	—	—	—	—	
H'FFFF8740	CCR	—	—	—	—	—	—	—	—	CAC
H'FFFF8741		—	—	—	CEDRAM	CECS3	CECS2	CECS1	CECS0	
H'FFFF8742 to H'FFFF87FF	—	—	—	—	—	—	—	—	—	

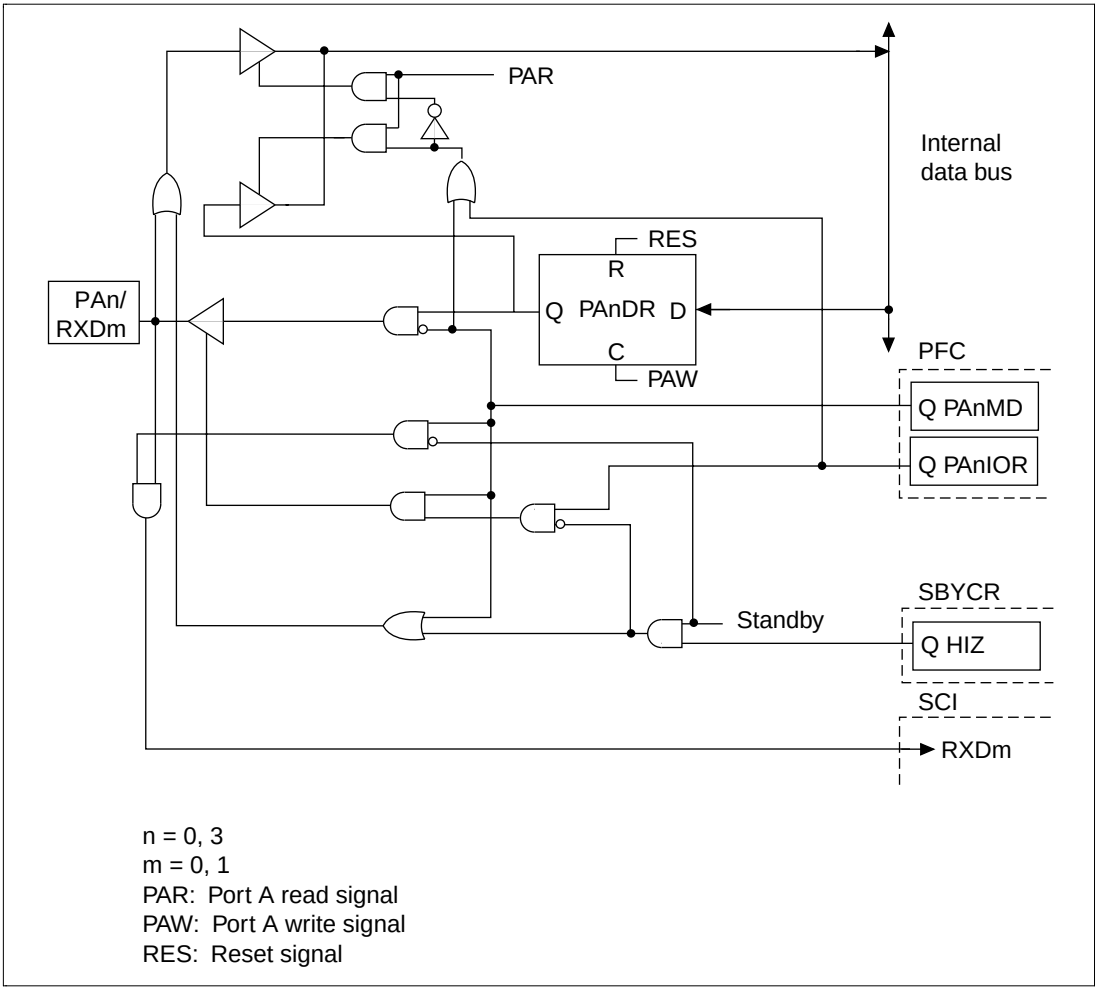


Figure B.1 PAn/RXDm Block Diagram

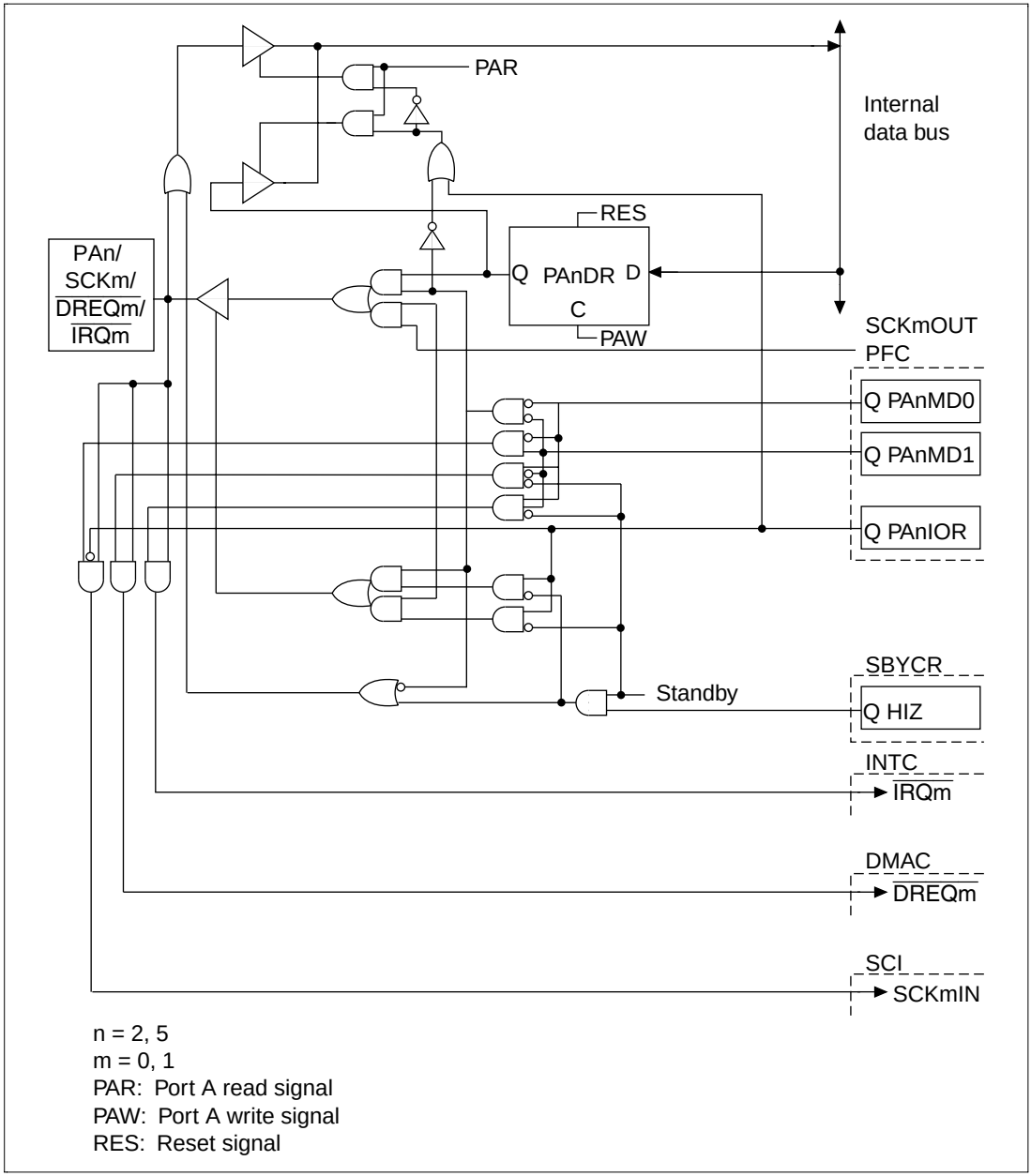


Figure B.2 PAn/SCKm/DREQm/IRQm Block Diagram

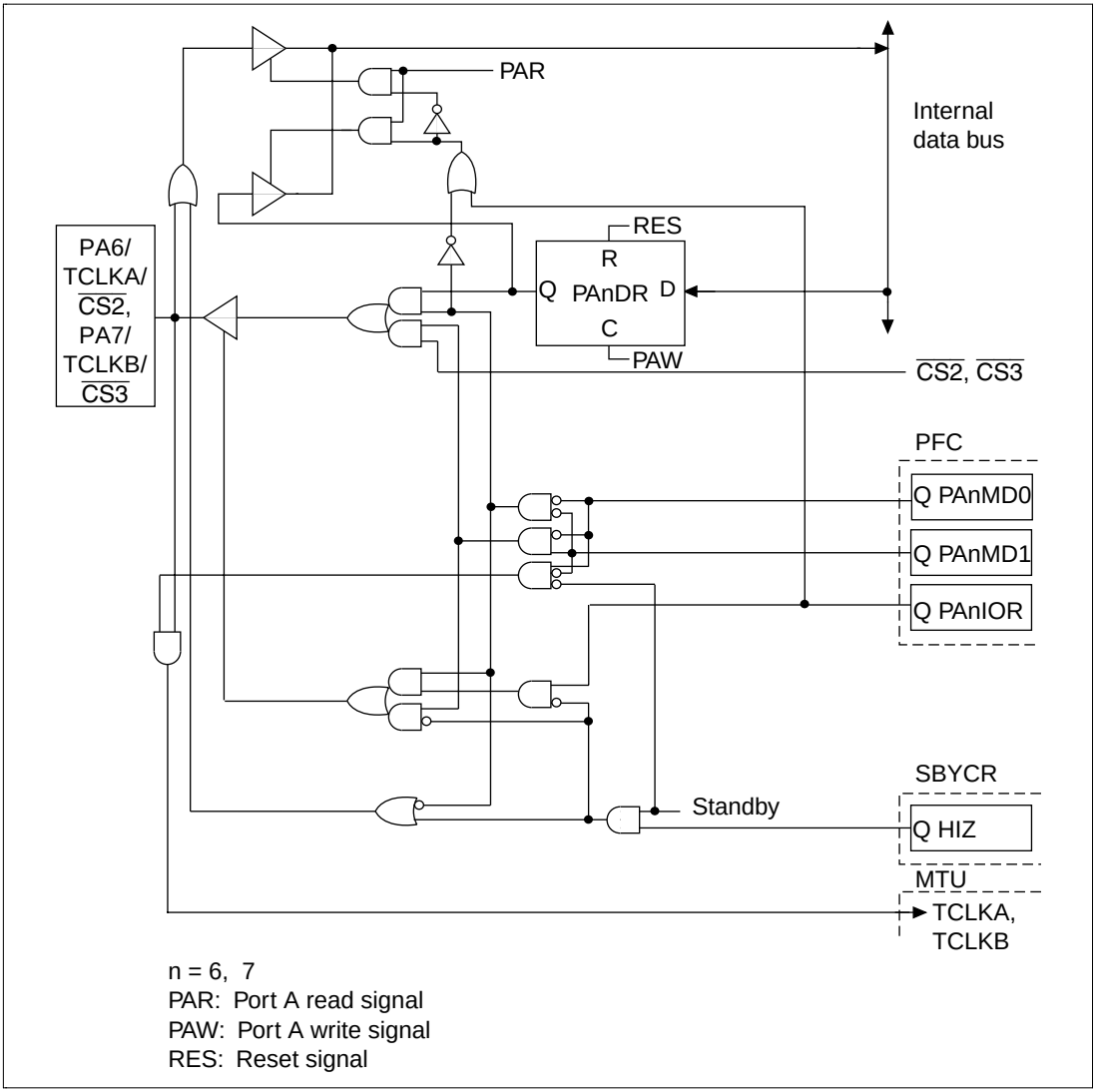


Figure B.3 PA6/TCLKA/$\overline{CS2}$, PA7/TCLKB/$\overline{CS3}$ (ZTAT, Mask) Block Diagram

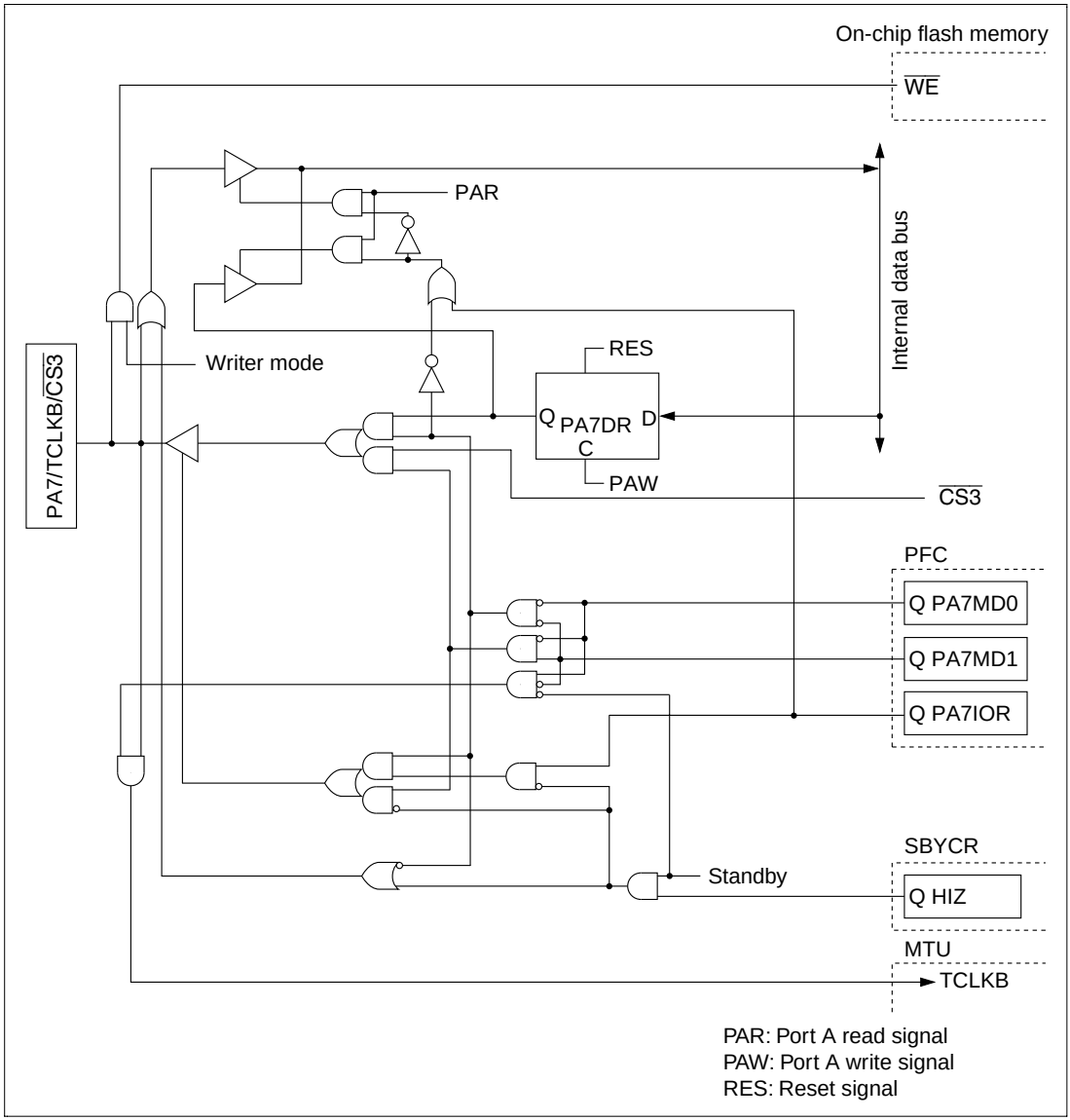


Figure B.4 PA7/TCLKB/$\overline{\text{CS3}}$ Block Diagram (F-ZTAT Version)

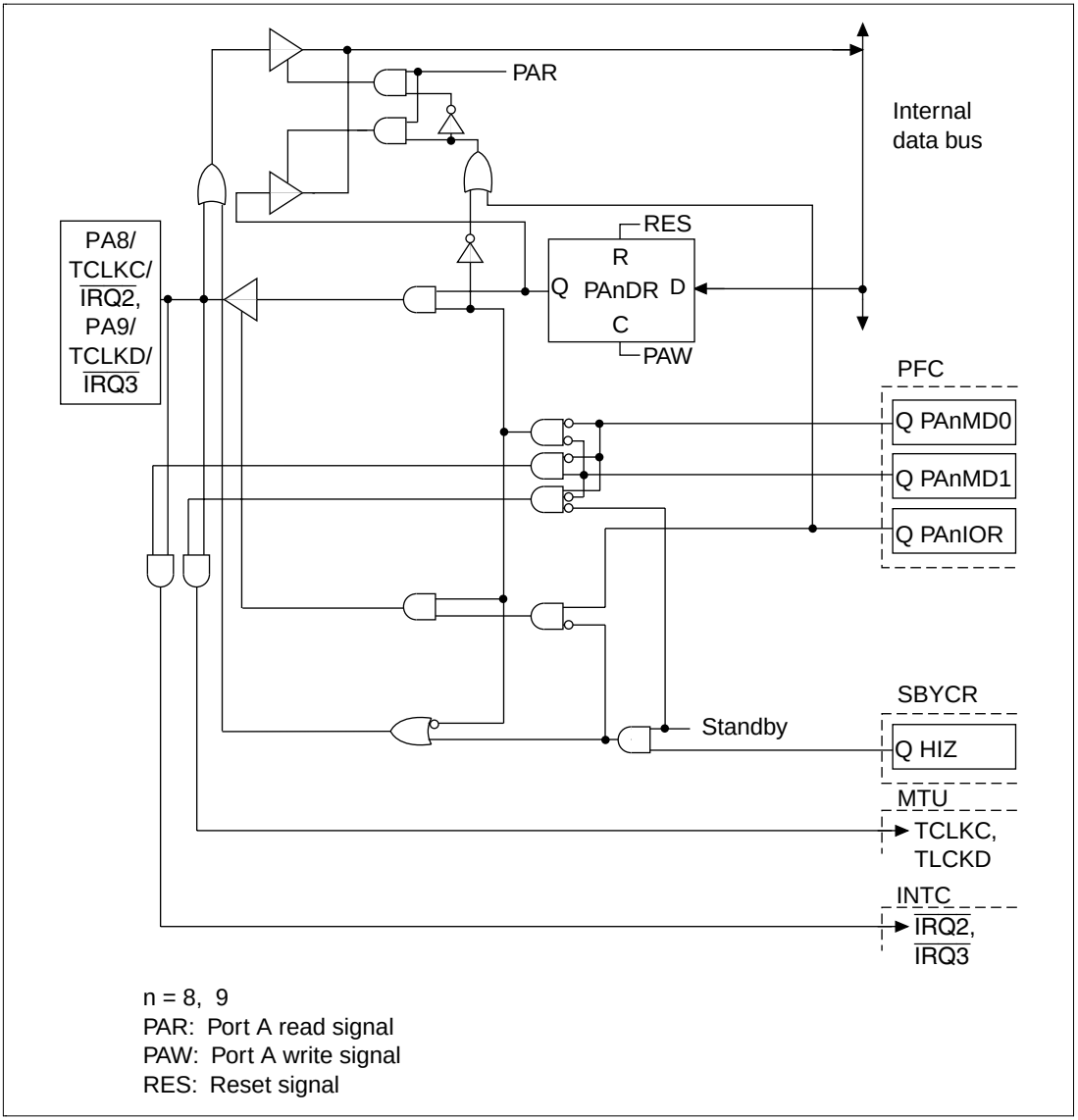


Figure B.5 PAn/TCLKm/$\overline{\text{IRQx}}$ Block Diagram (ZTAT, Mask)

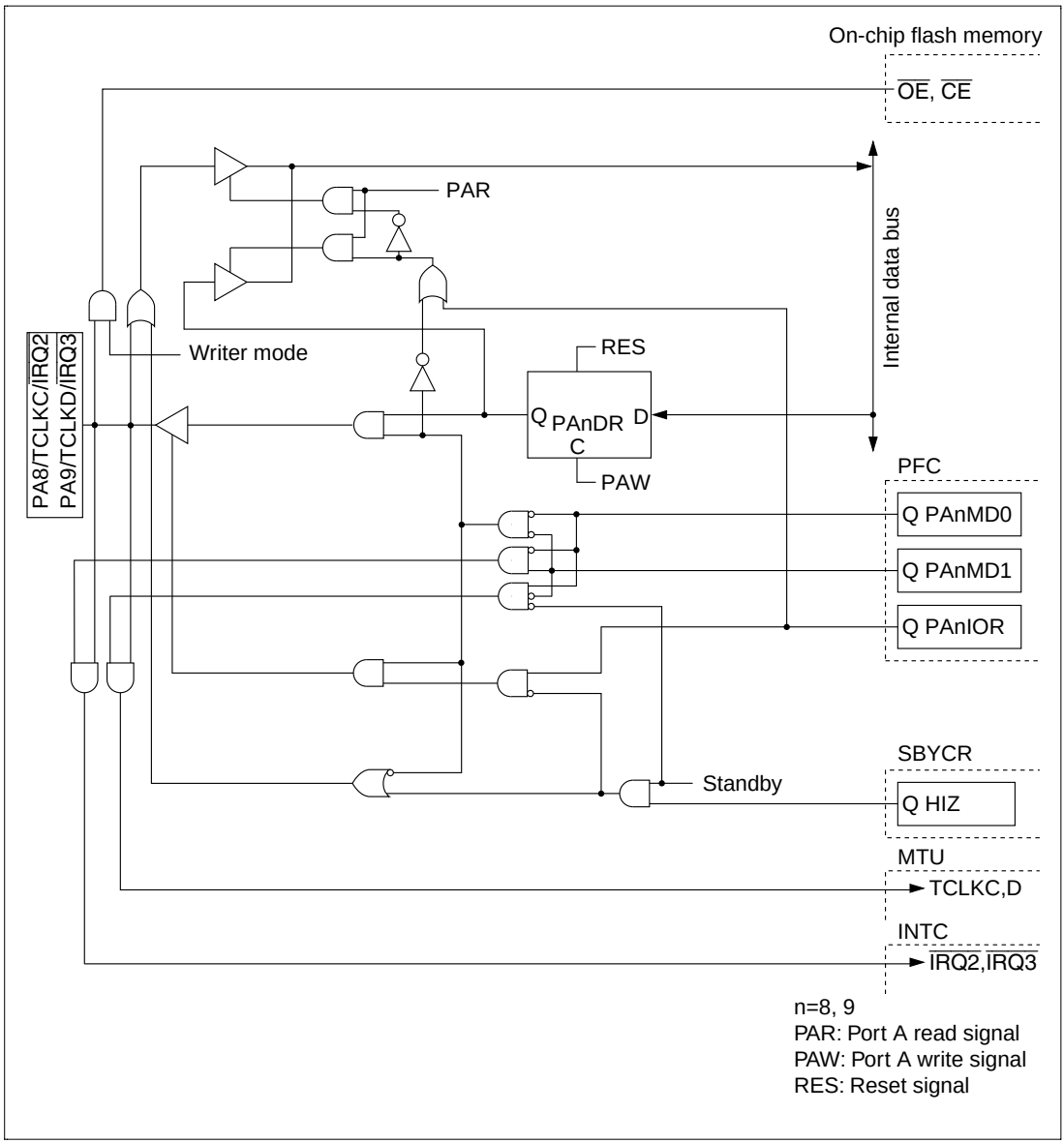


Figure B.6 PAn/TCLKm/$\overline{\text{IRQx}}$ Block Diagram (F-ZTAT Version)

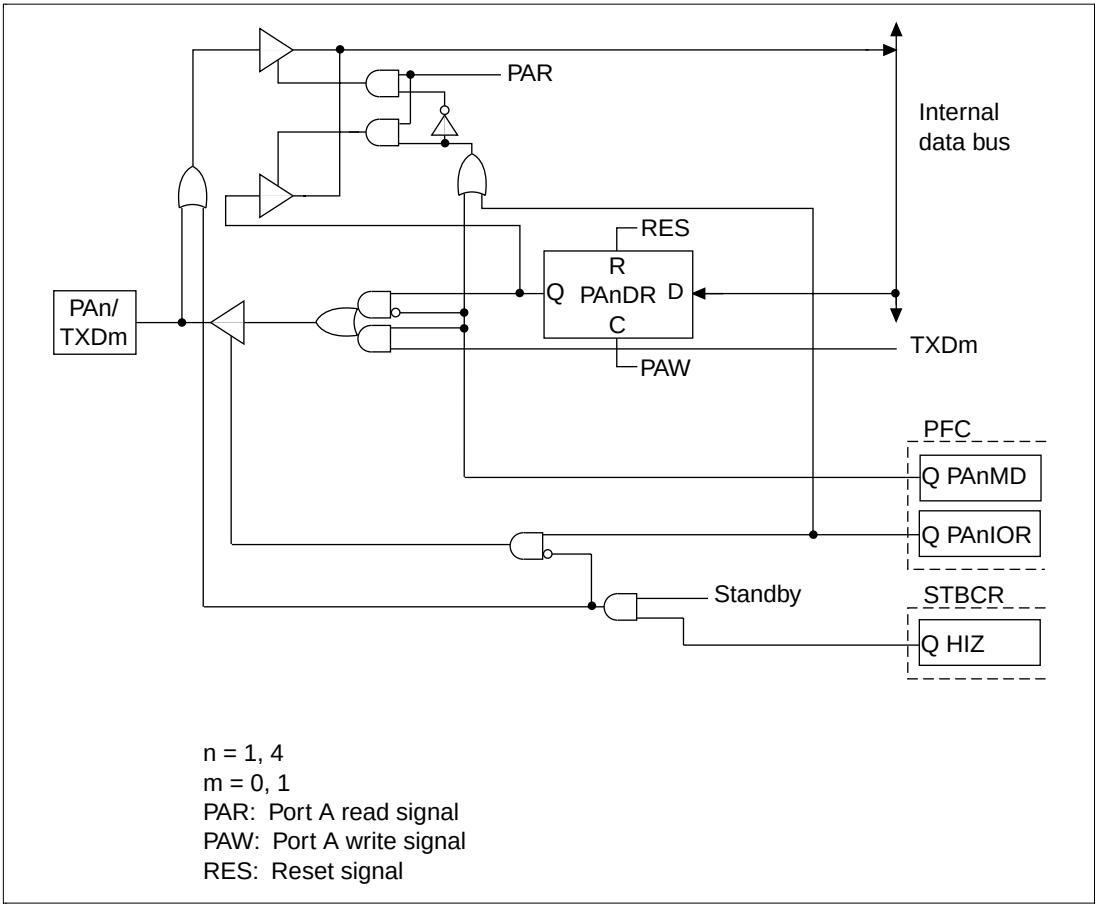


Figure B.7 PAn/TXDm Block Diagram

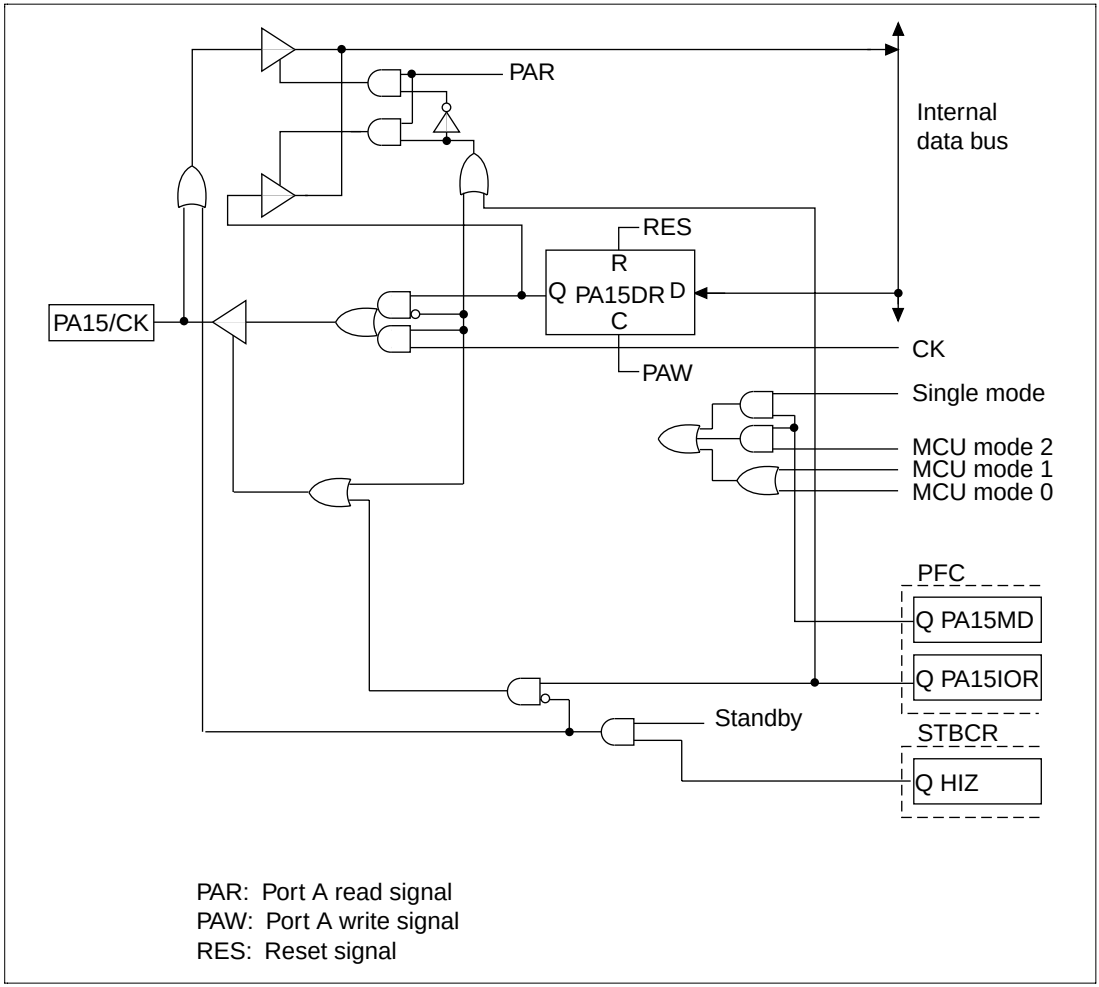


Figure B.8 PA15/CK Block Diagram

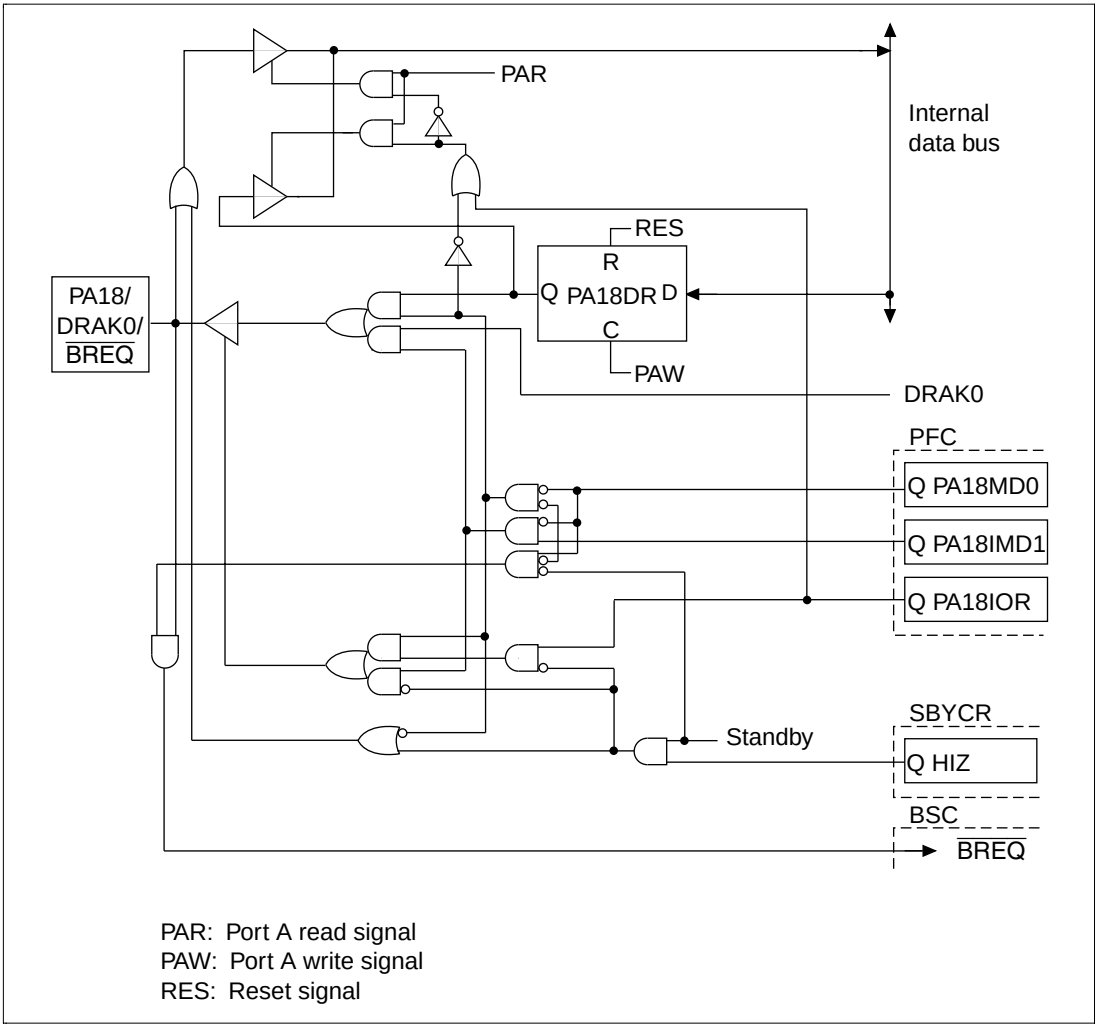


Figure B.9 PA18/DRAK0/$\overline{\text{BREQ}}$ Block Diagram

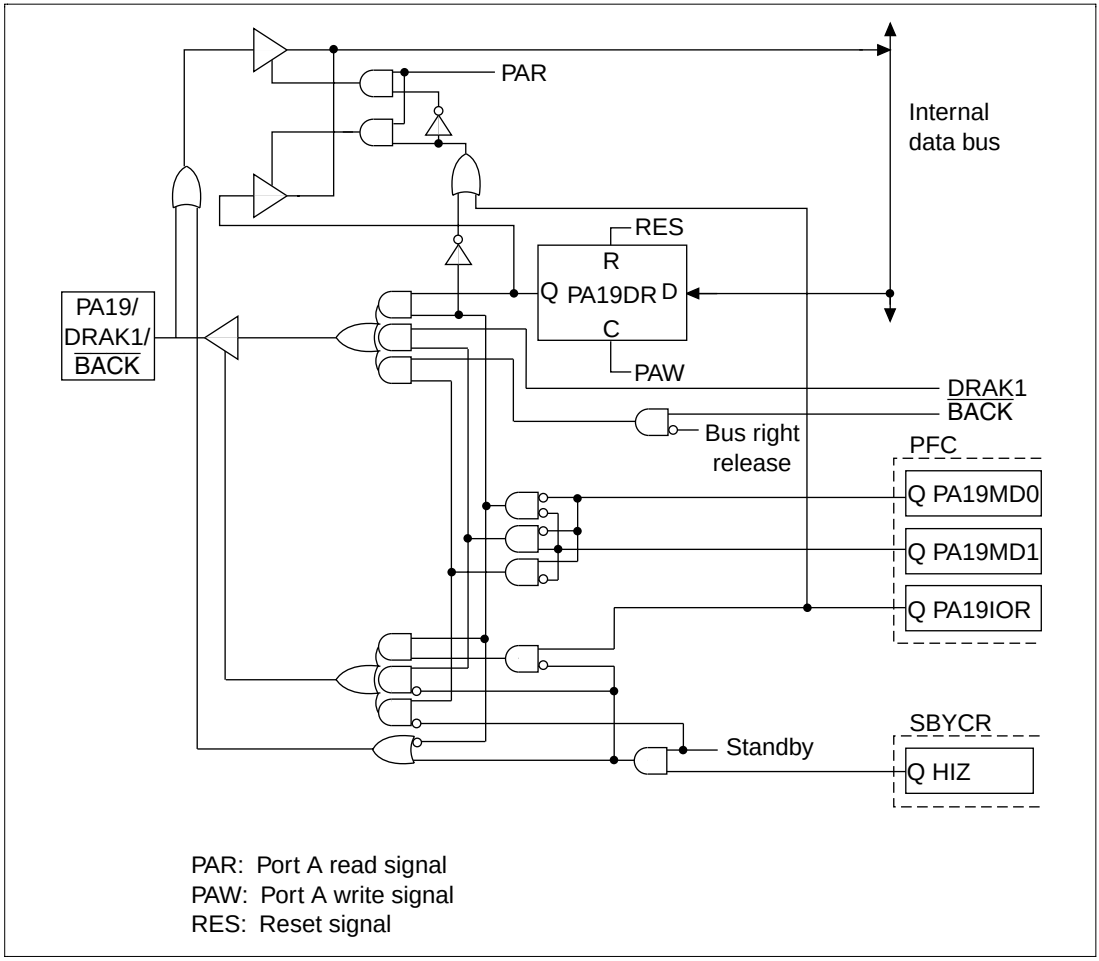


Figure B.10 PA19/DRAQ1/$\overline{\text{BACK}}$ Block Diagram

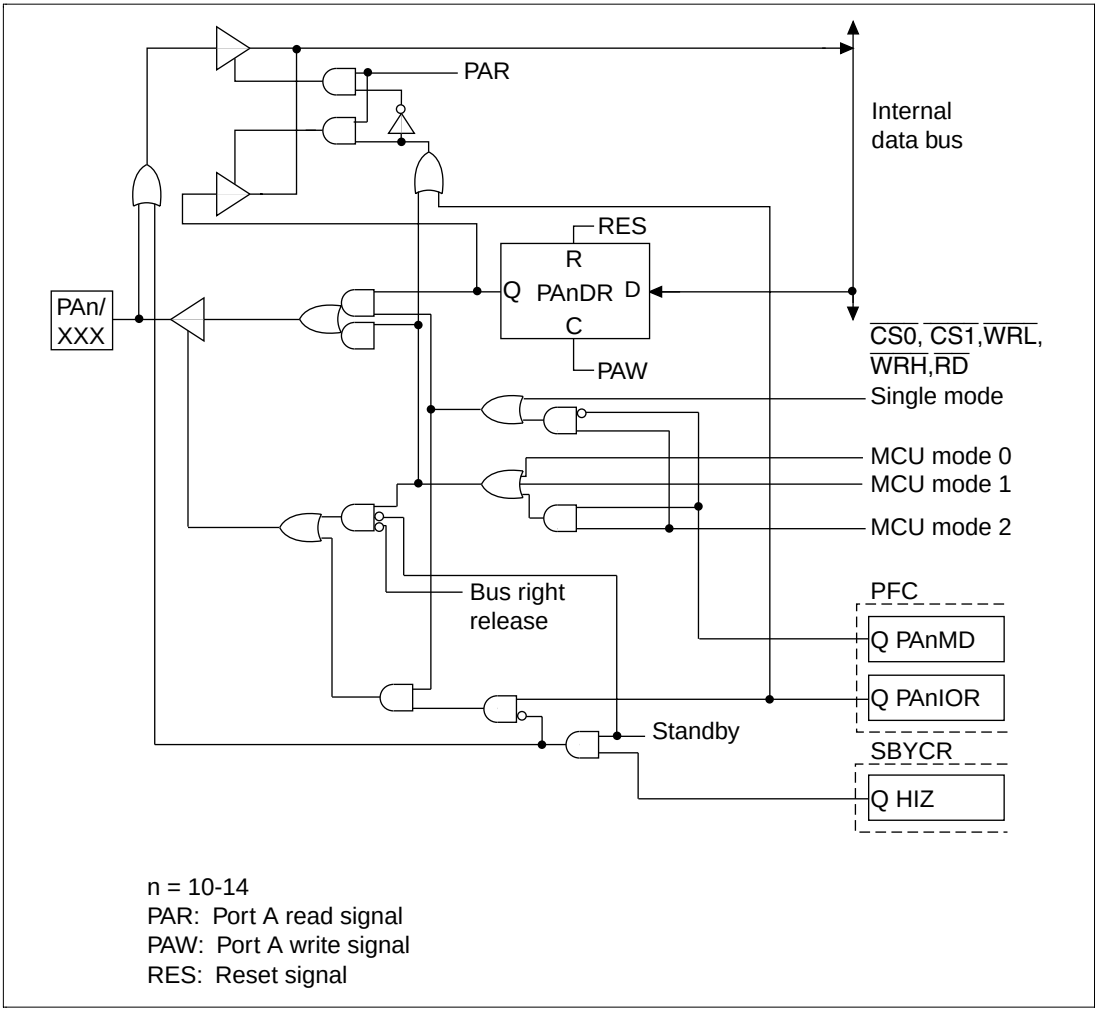


Figure B.11 PAn/XXX Block Diagram

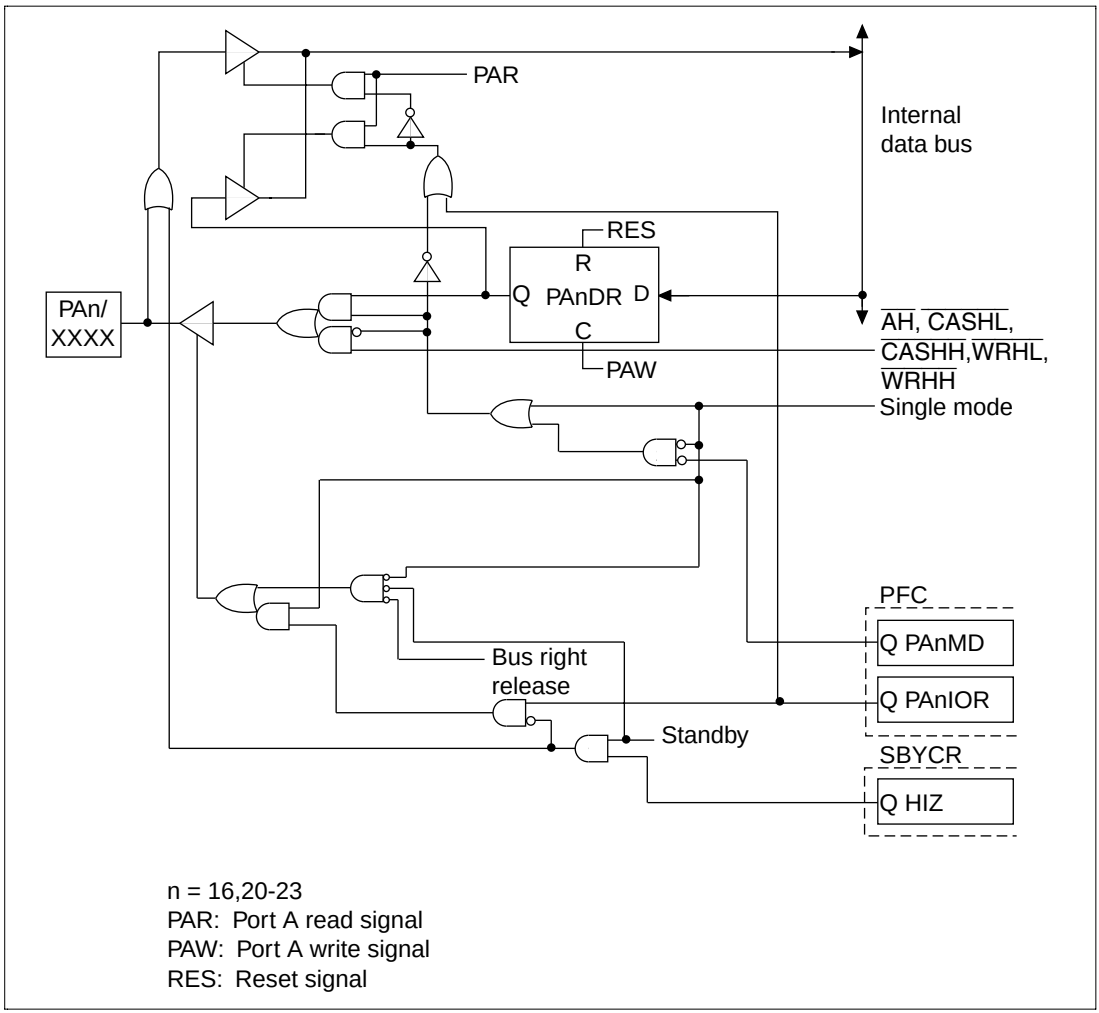


Figure B.12 PAn/XXXX Block Diagram

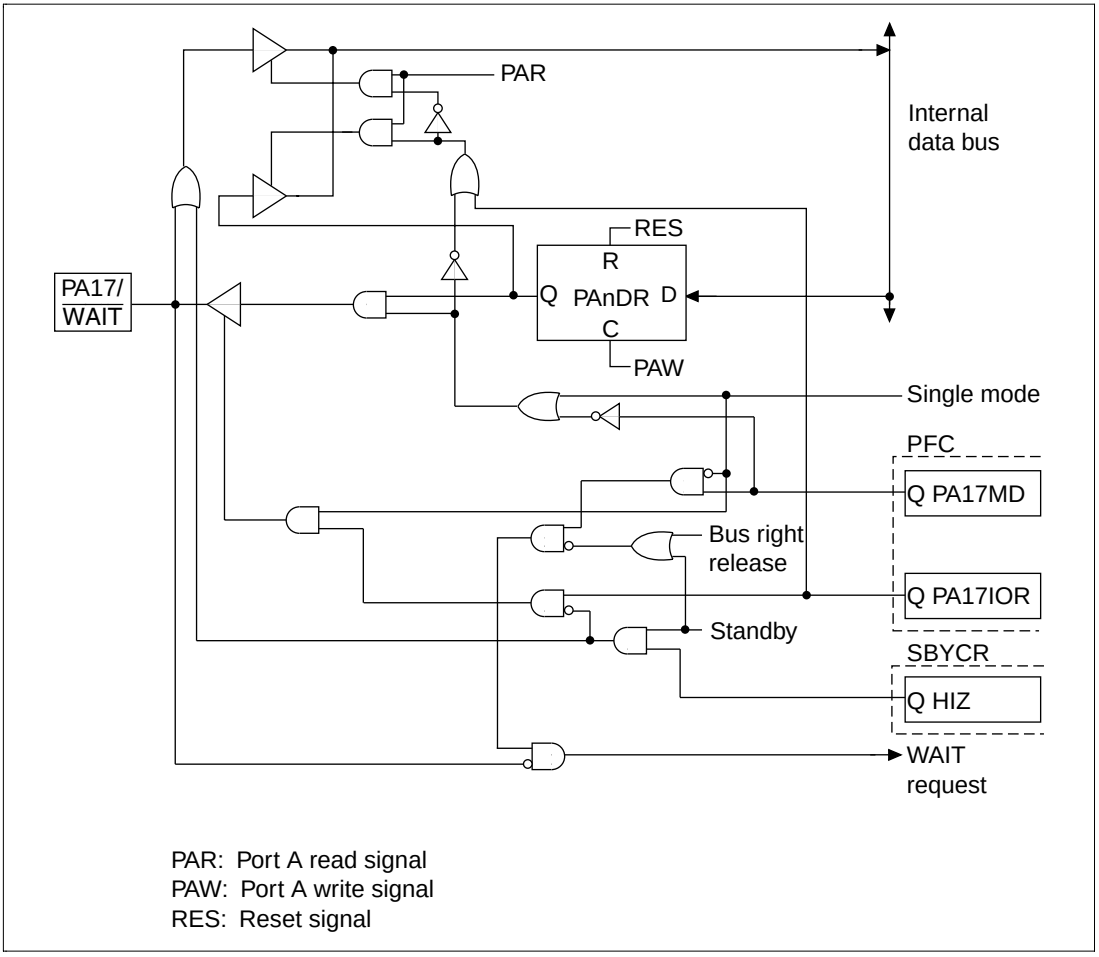


Figure B.13 PA17/$\overline{\text{WAIT}}$ Block Diagram

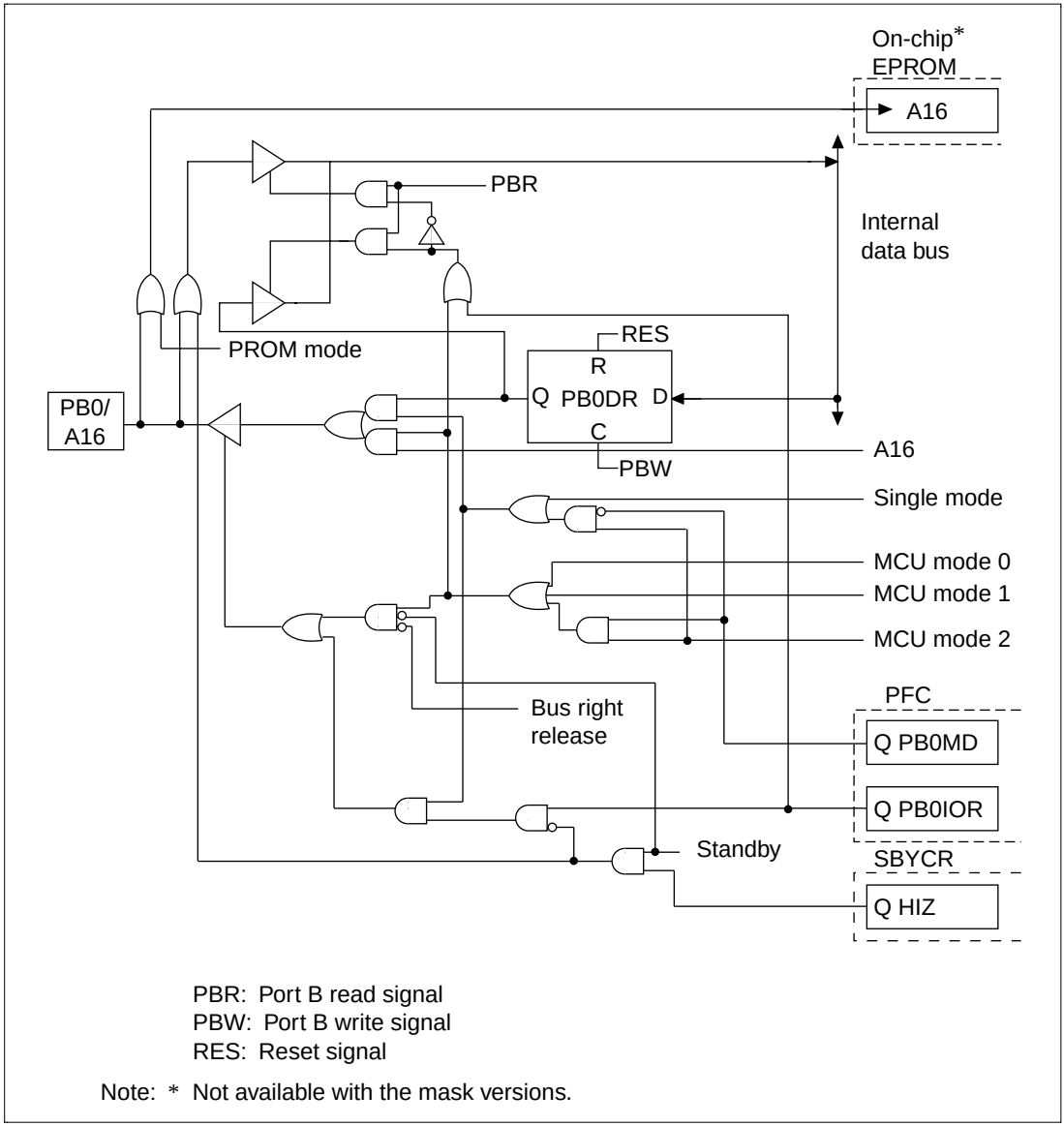


Figure B.14 PB0/A16 Block Diagram

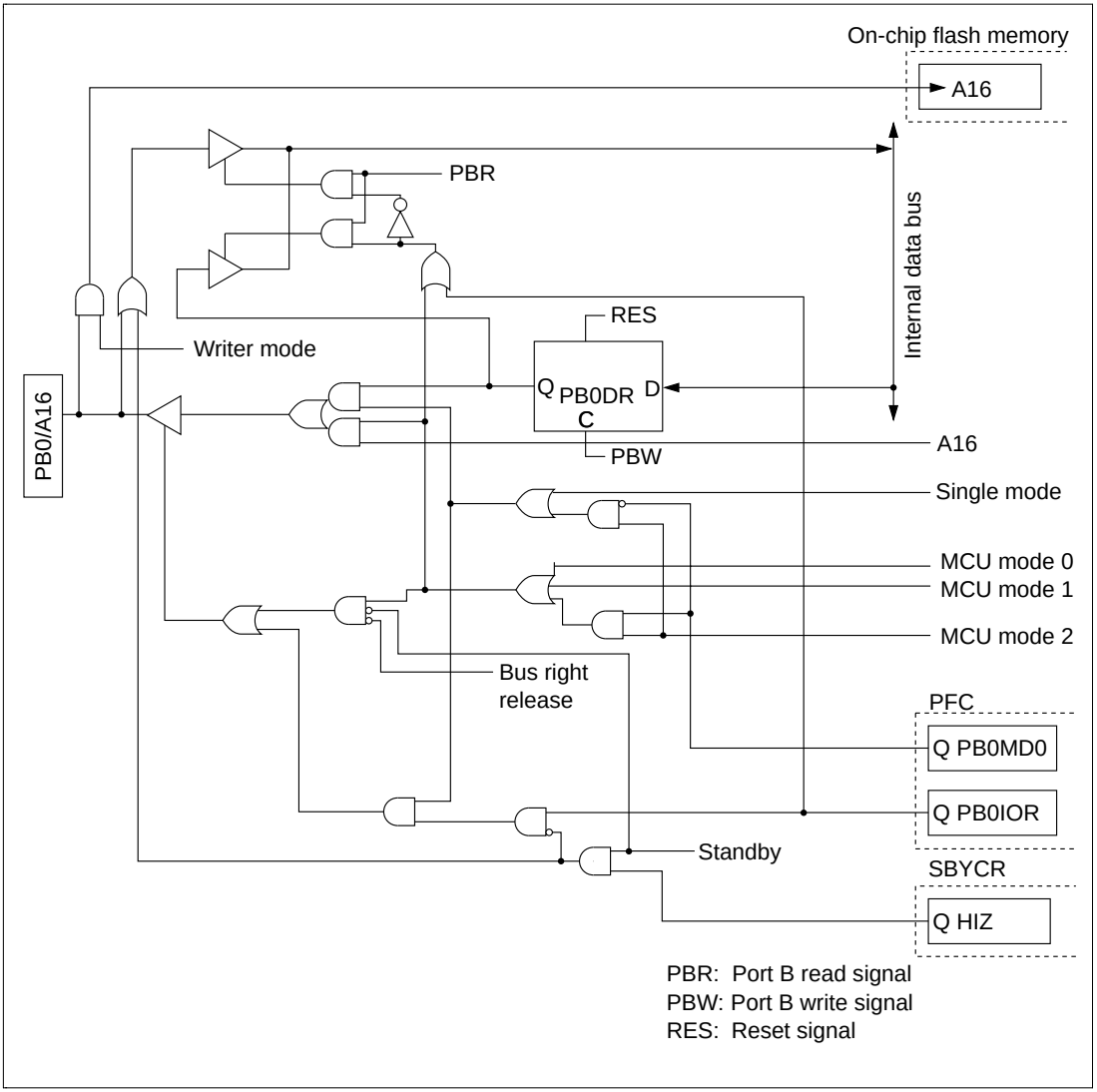


Figure B.15 PB0/A16 Block Diagram (F-ZTAT Version)

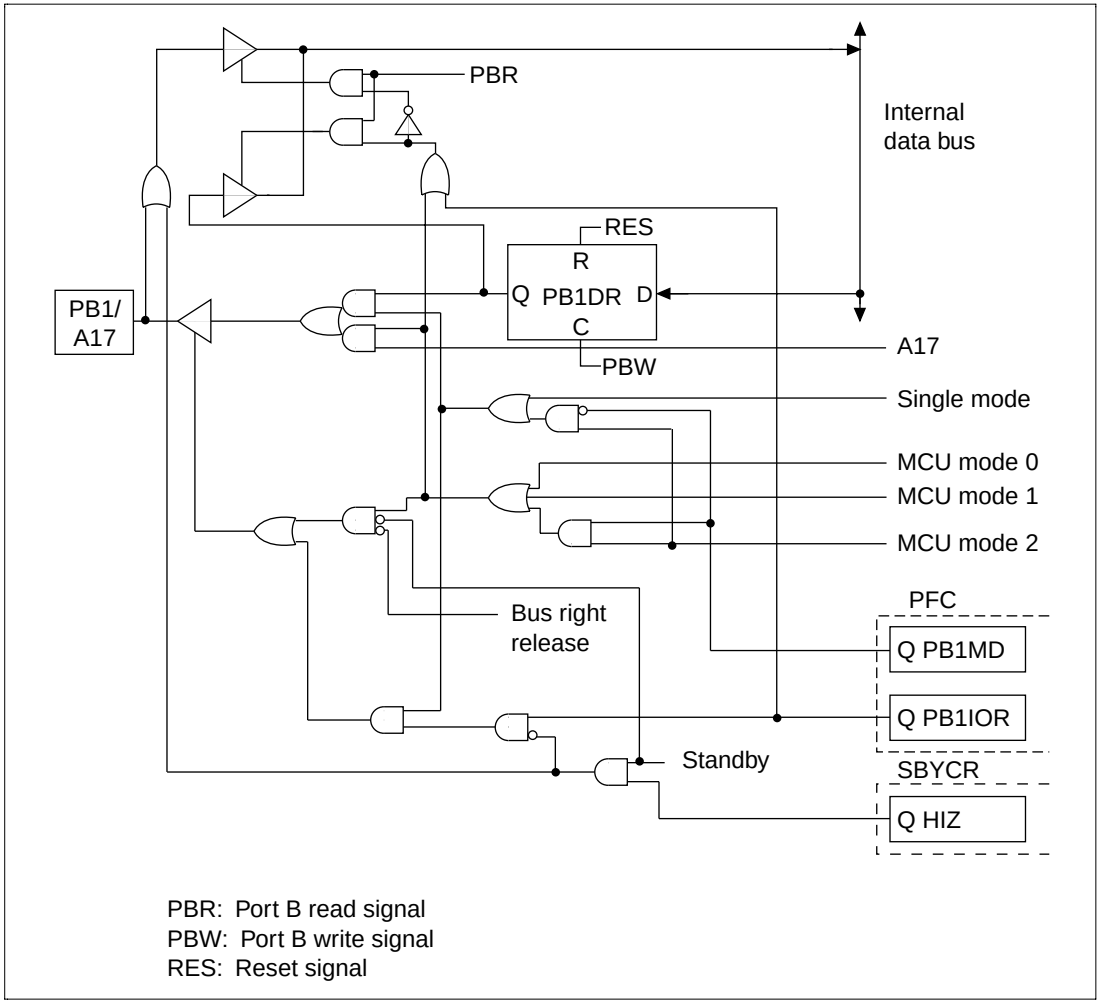


Figure B.16 PB1/A17 Block Diagram

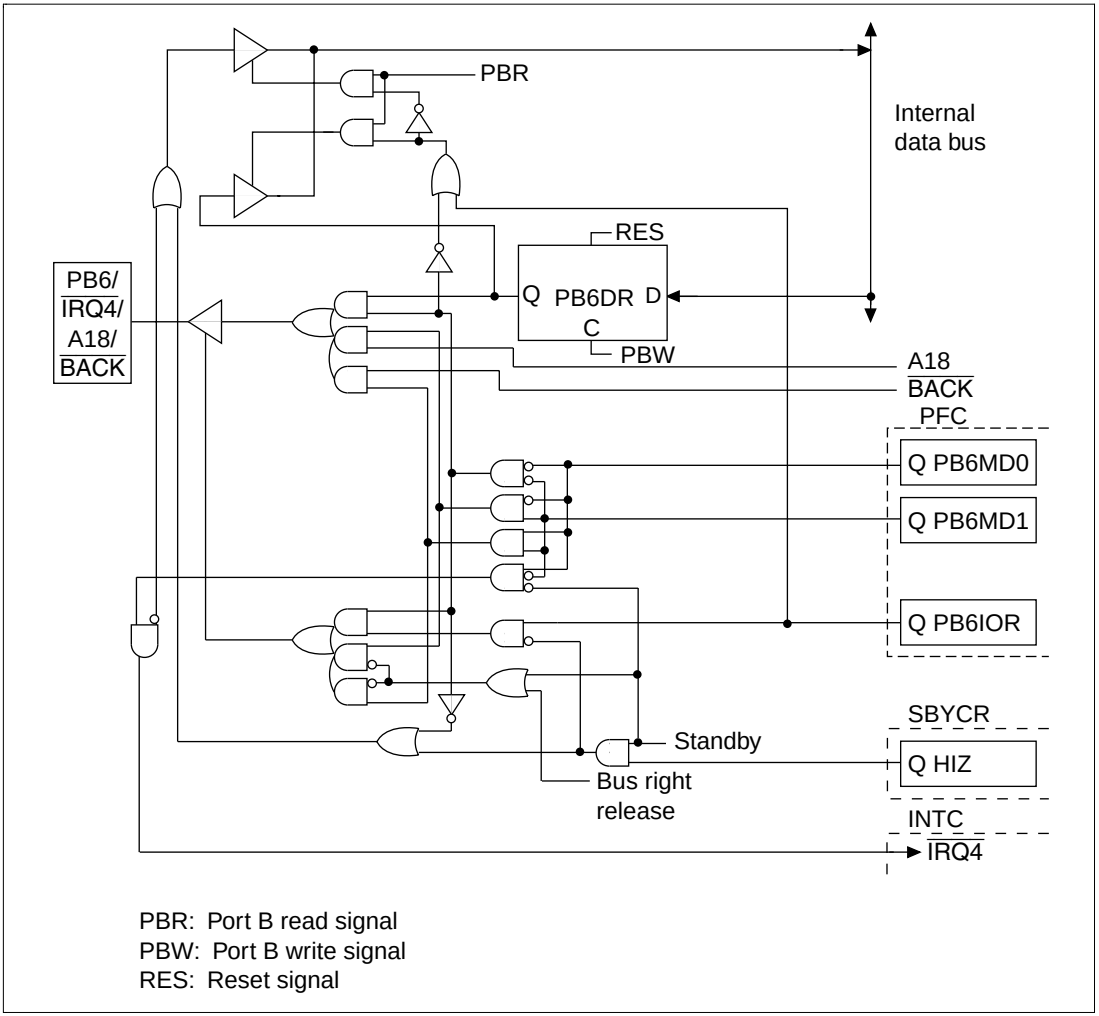
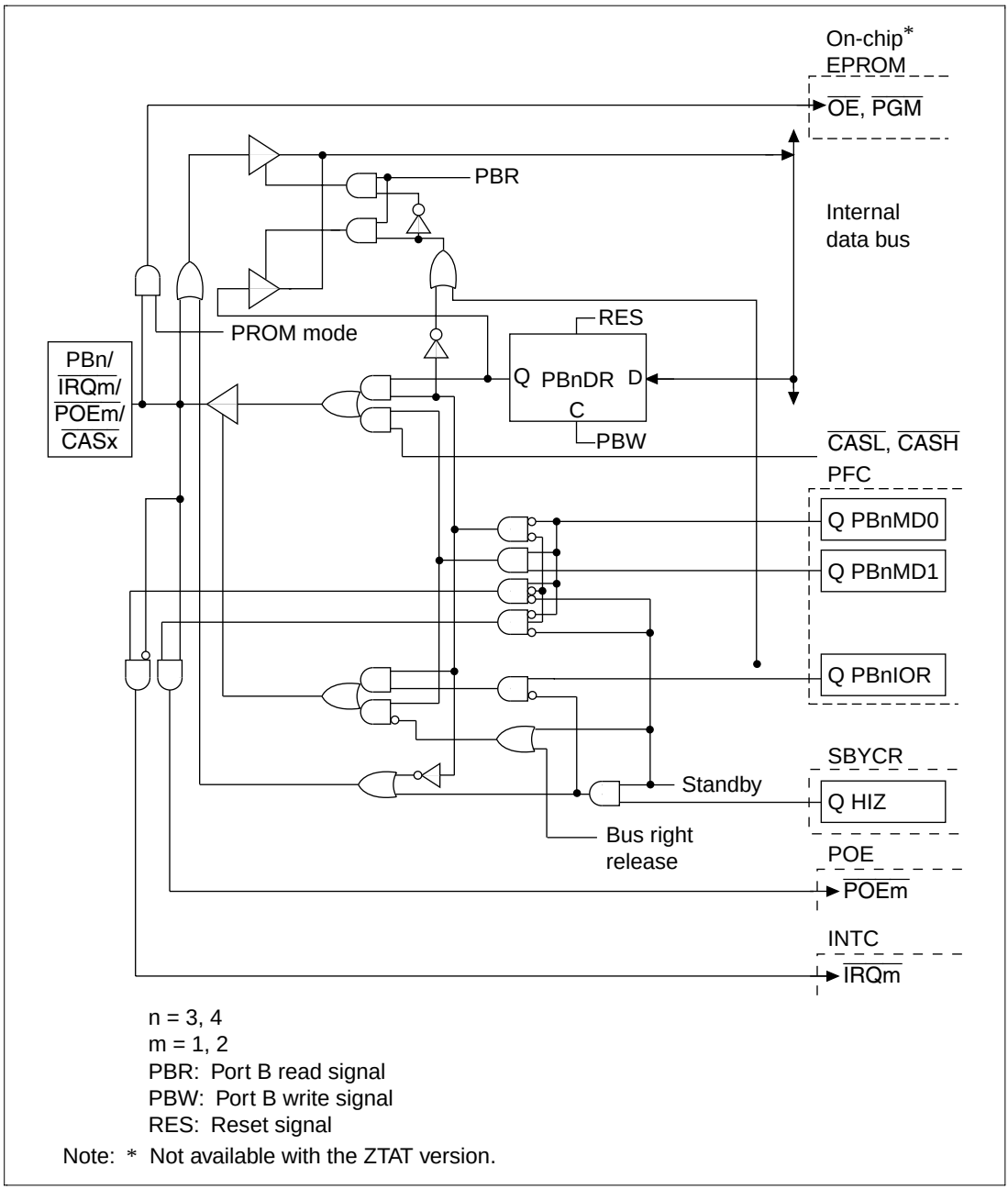


Figure B.17 PB6/$\overline{\text{IRQ4}}$/A18/$\overline{\text{BACK}}$ Block Diagram



n = 3, 4
m = 1, 2
PBR: Port B read signal
PBW: Port B write signal
RES: Reset signal

Note: * Not available with the ZTAT version.

Figure B.18 PBn/IRQm/POEm/CASx Block Diagram

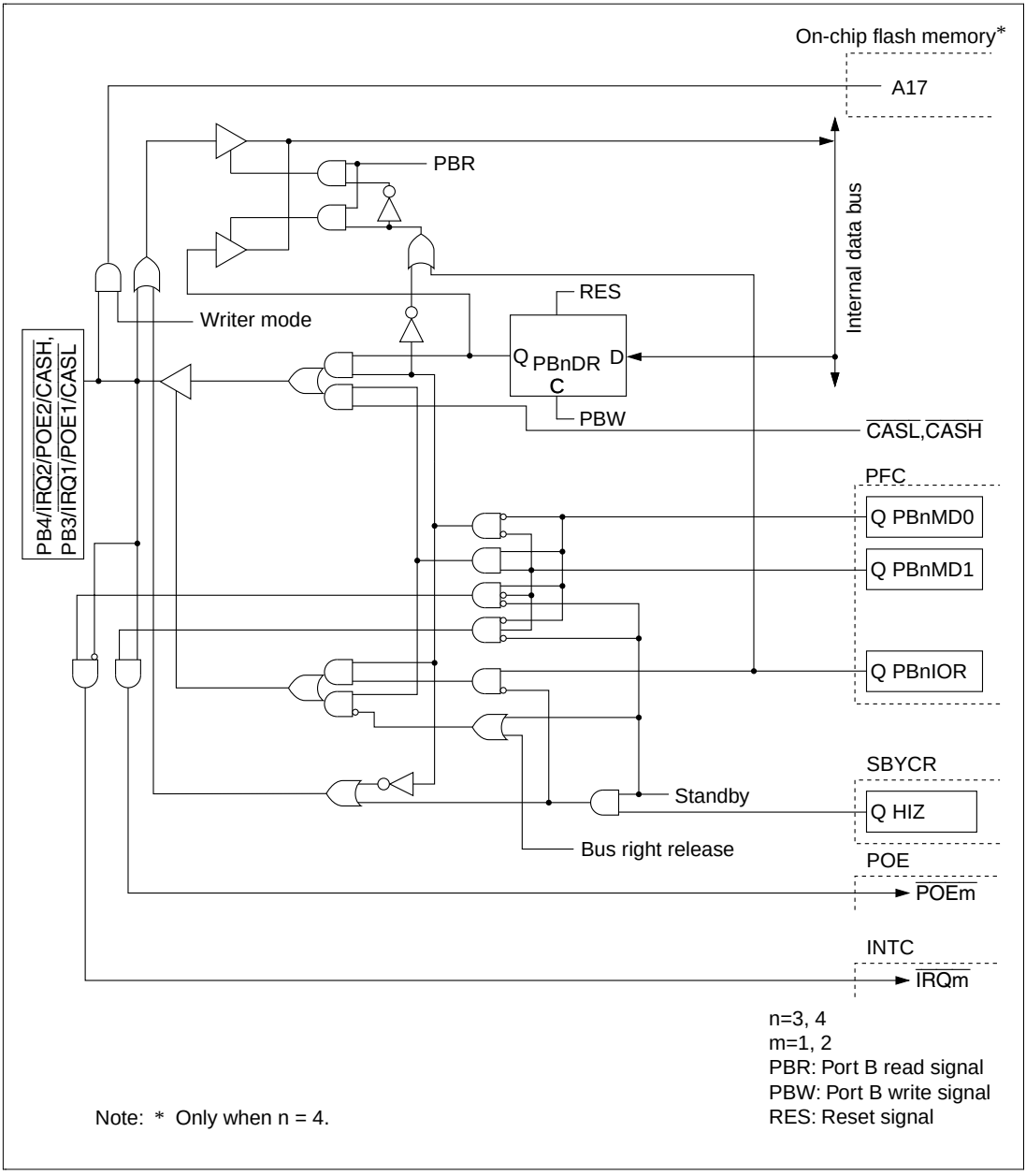


Figure B.19 PB4/IRQ2/POE2/CASH,PB3/IRQ1/POE1/CASL Block Diagram (F-ZTAT Version)

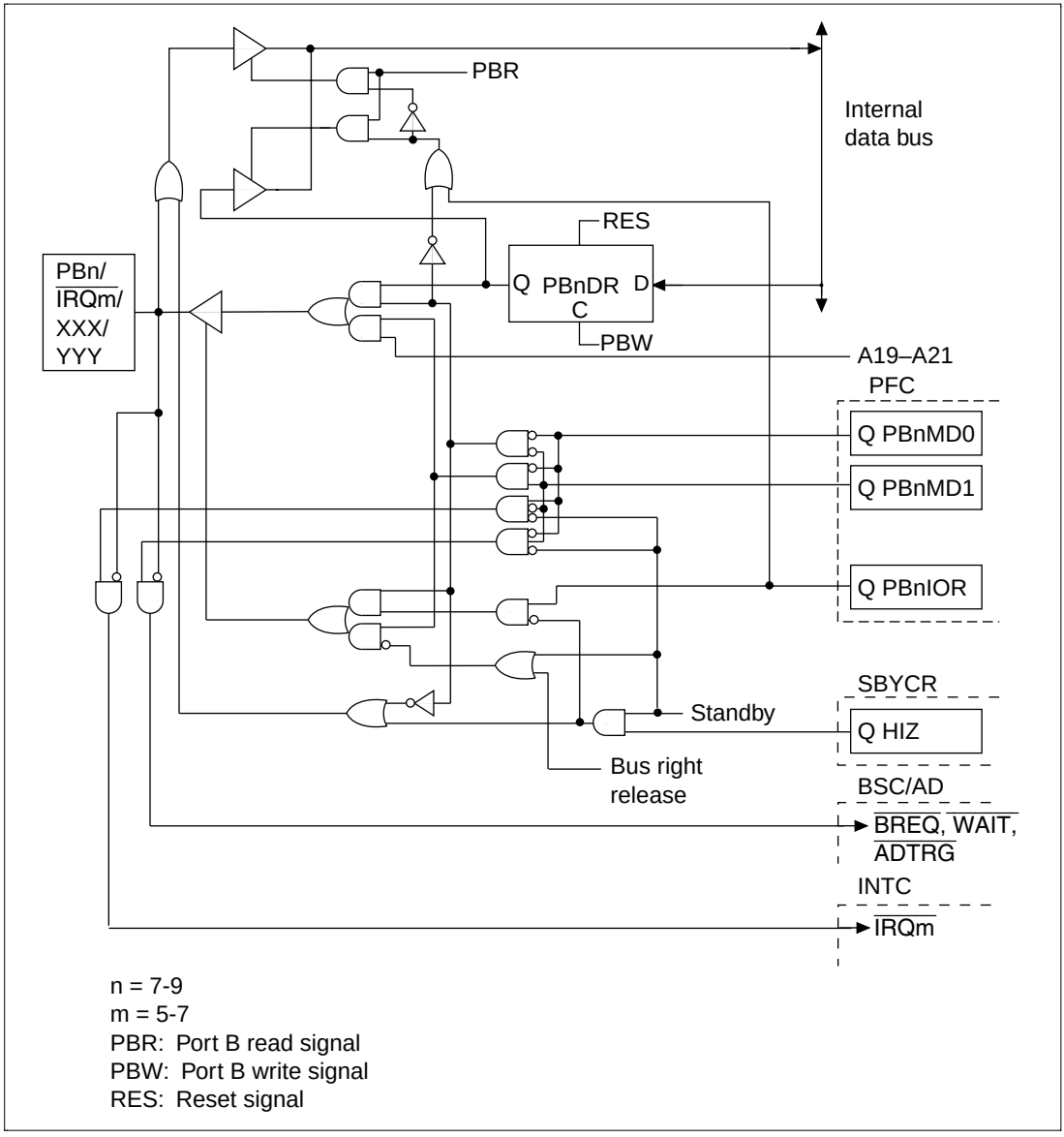


Figure B.20 PBn/IRQm/XXX/YYY Block Diagram

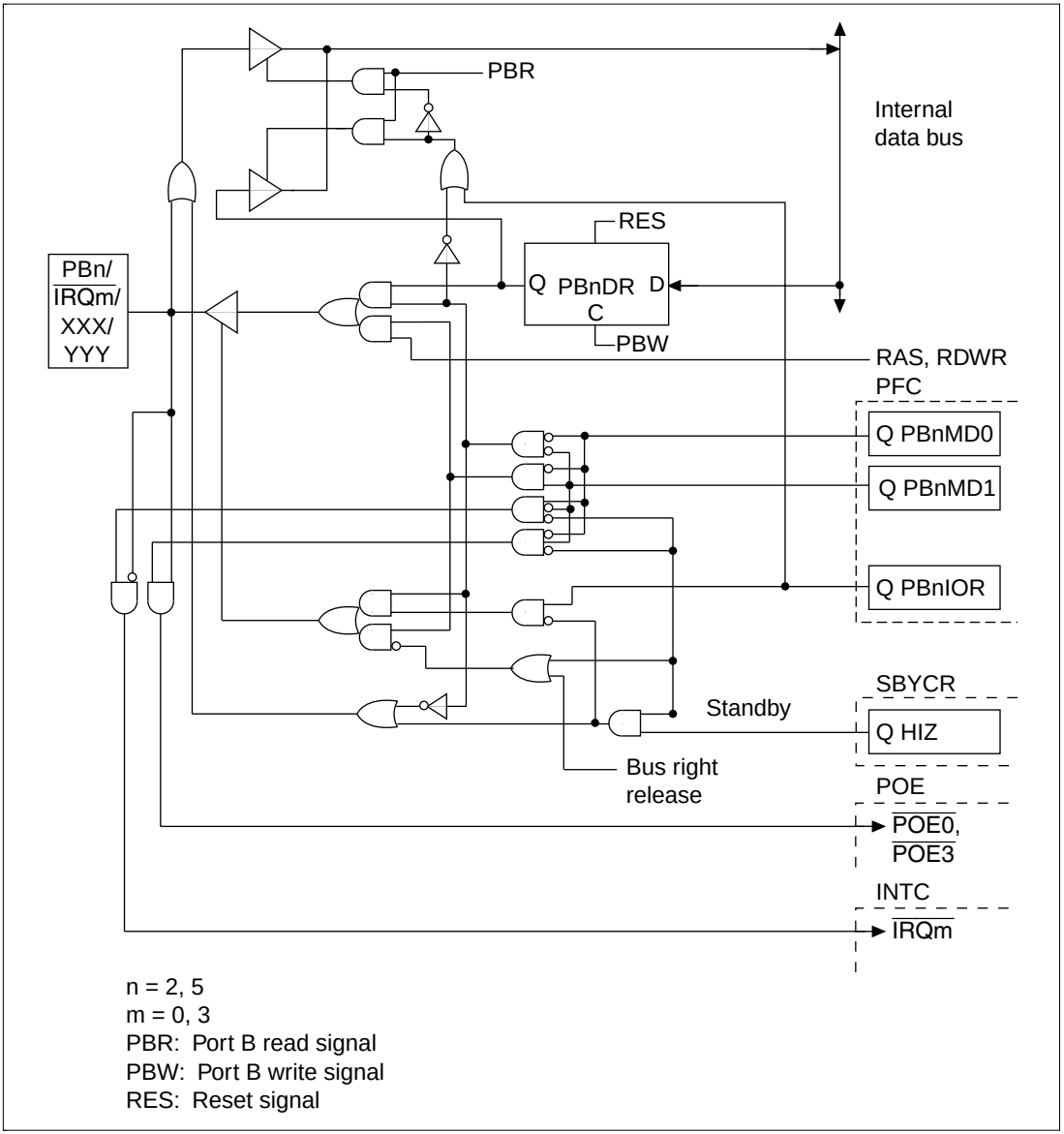


Figure B.21 PBn/IRQm/XXXX/YYYY Block Diagram

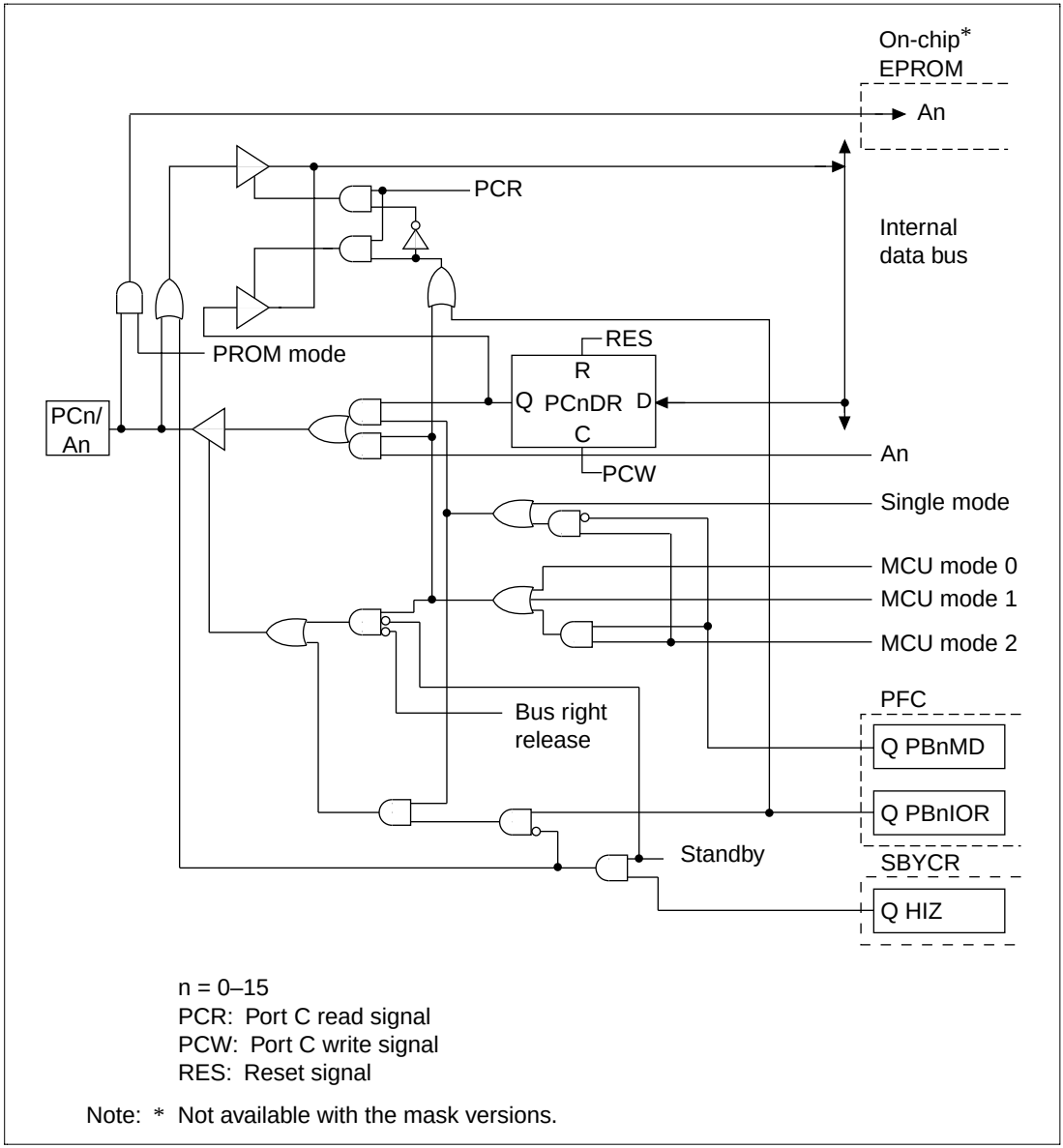


Figure B.22 PCn/An Block Diagram

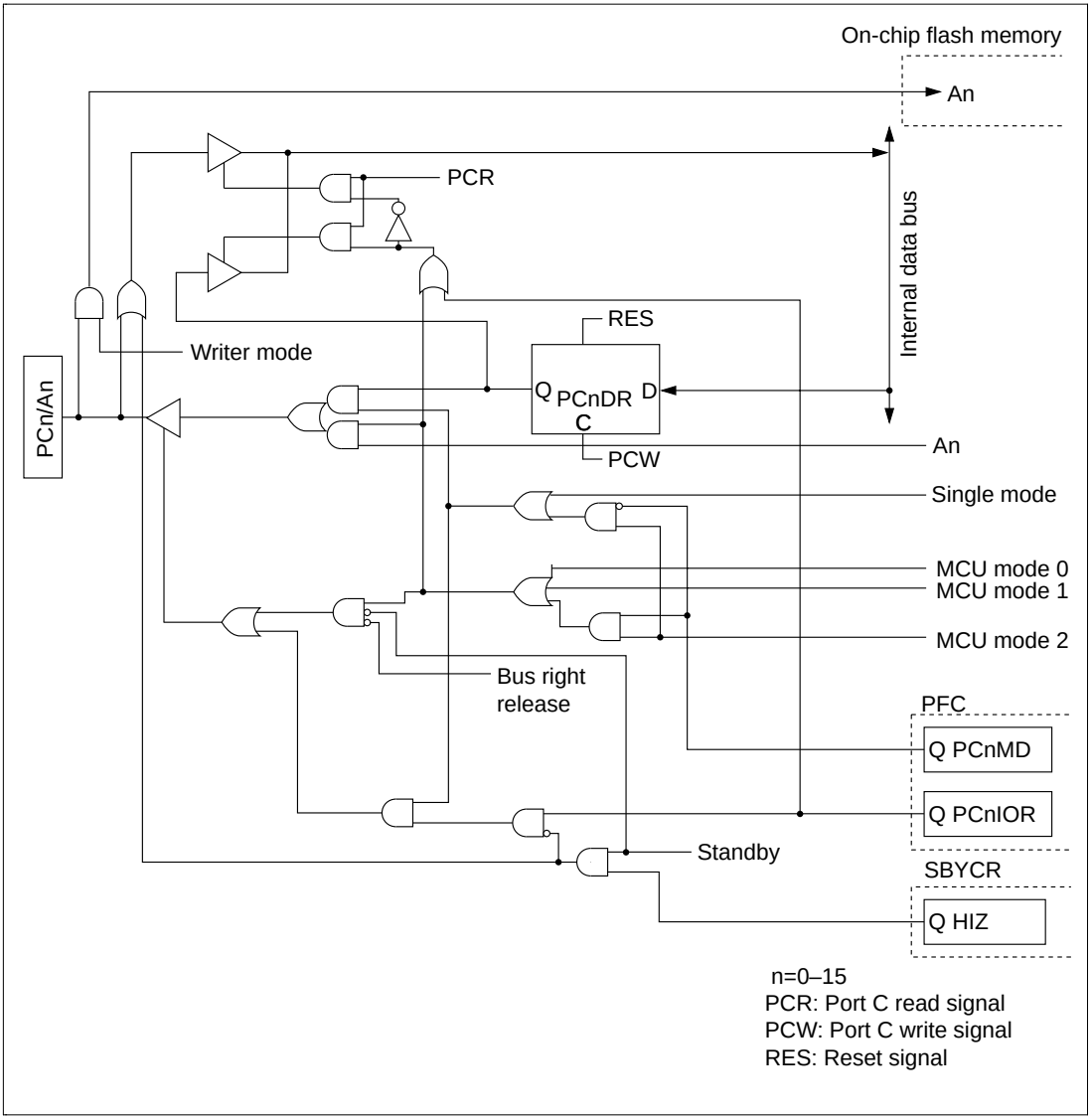
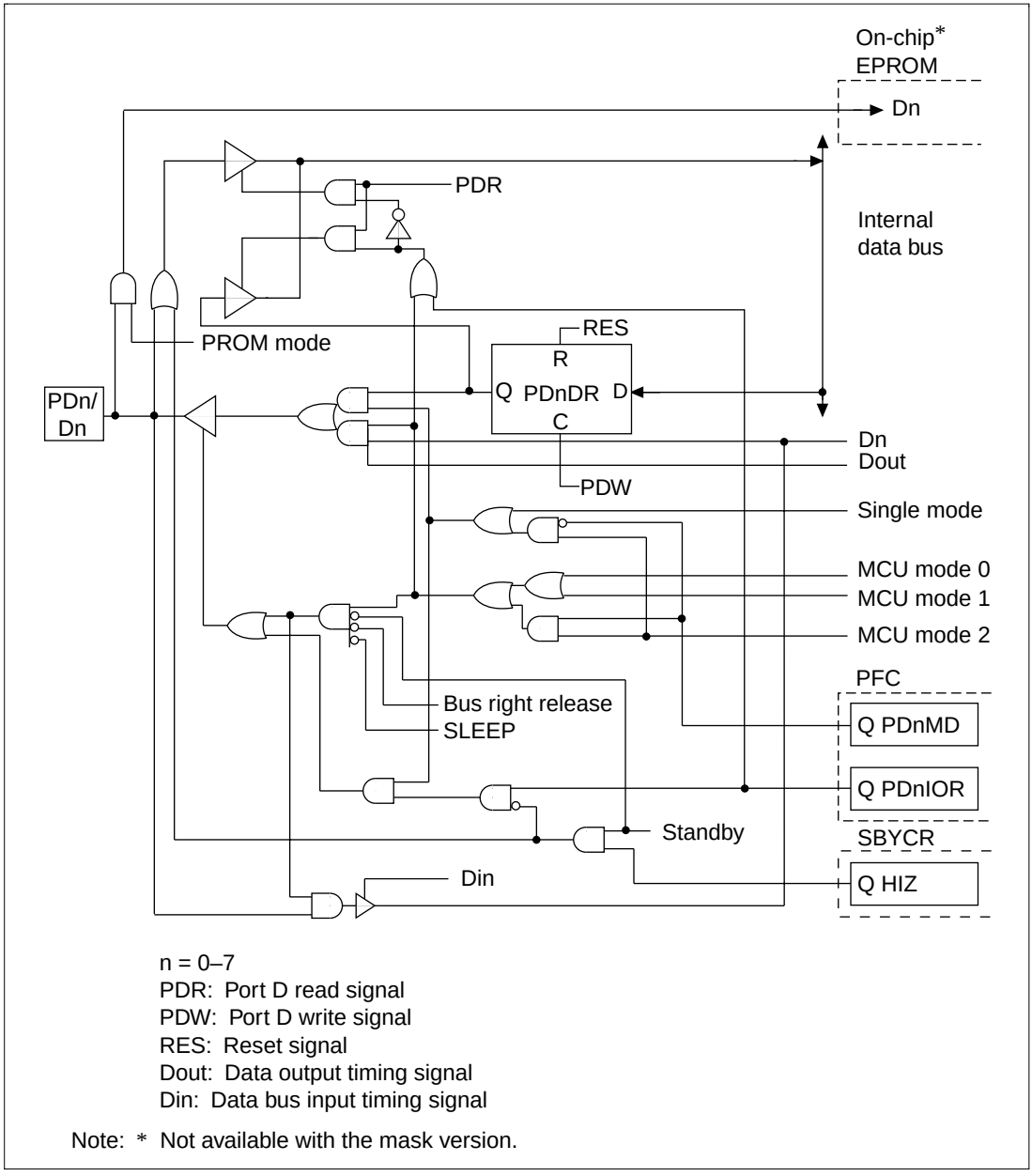


Figure B.23 PCn/An Block Diagram (F-ZTAT Version)



n = 0–7
PDR: Port D read signal
PDW: Port D write signal
RES: Reset signal
Dout: Data output timing signal
Din: Data bus input timing signal

Note: * Not available with the mask version.

Figure B.24 PDn/Dn Block Diagram

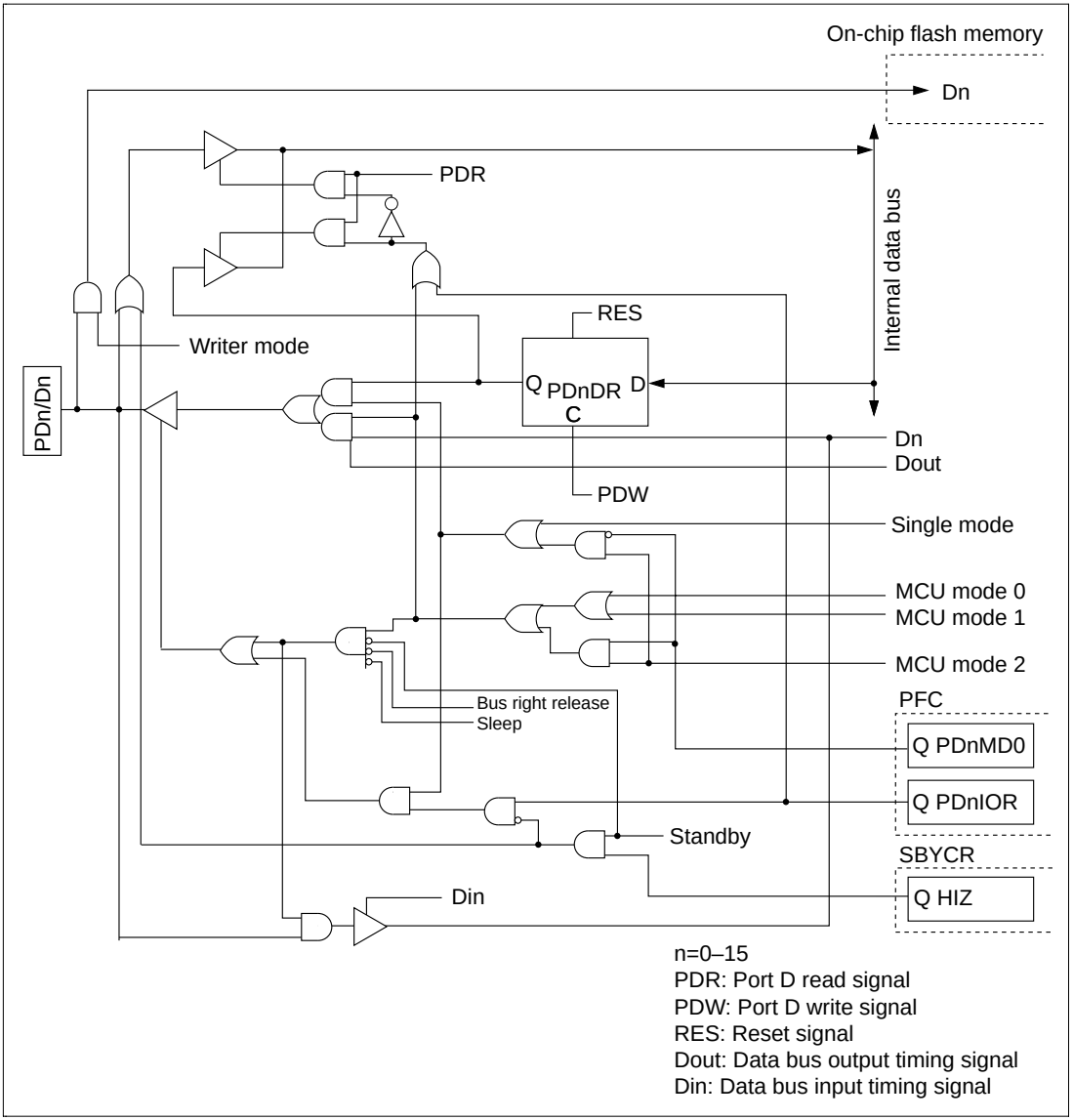


Figure B.25 PDn/Dn Block Diagram (F-ZTAT Version)

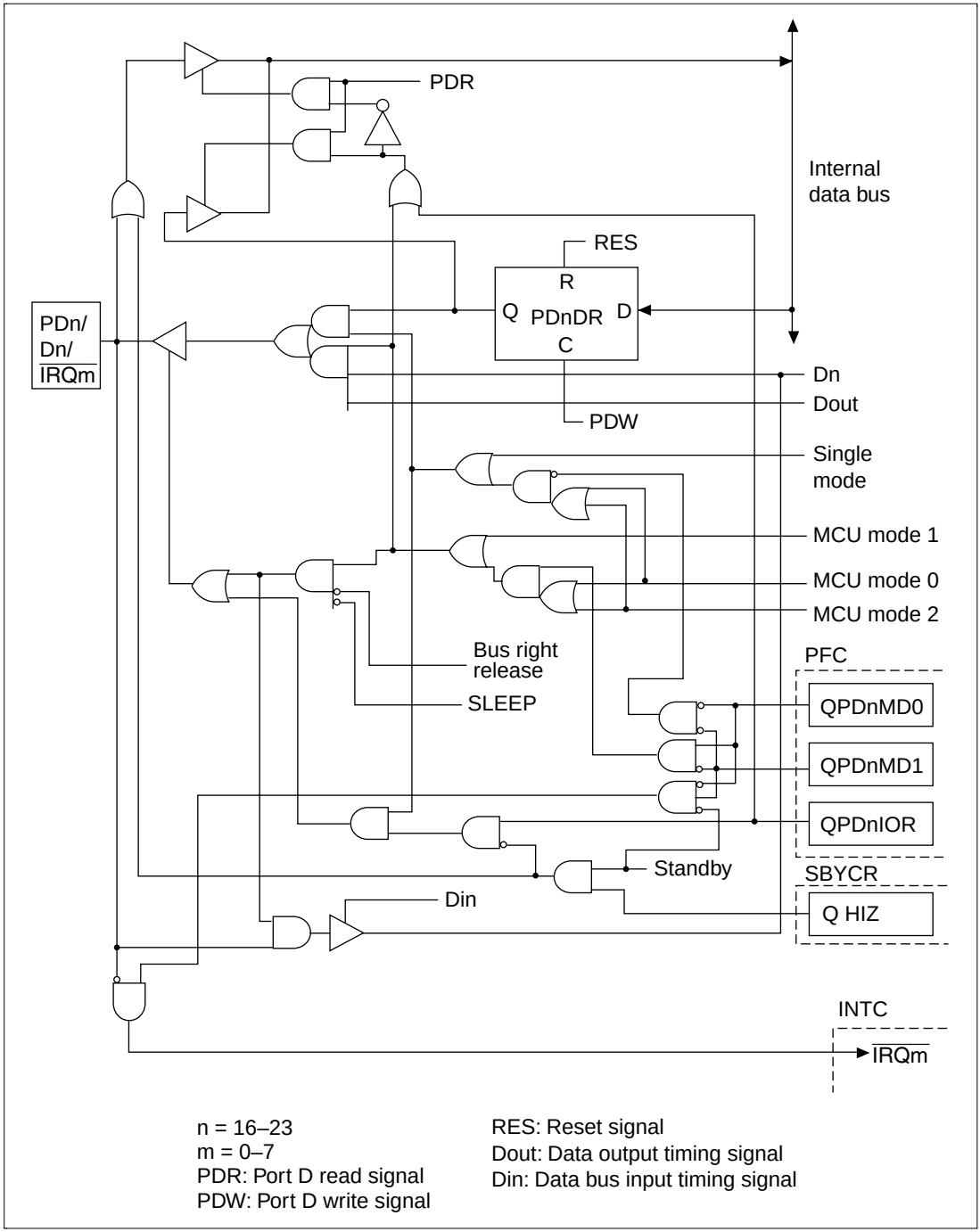


Figure B.26 PDn/Dn/$\overline{\text{IRQm}}$ Block Diagram (n = 16–23)

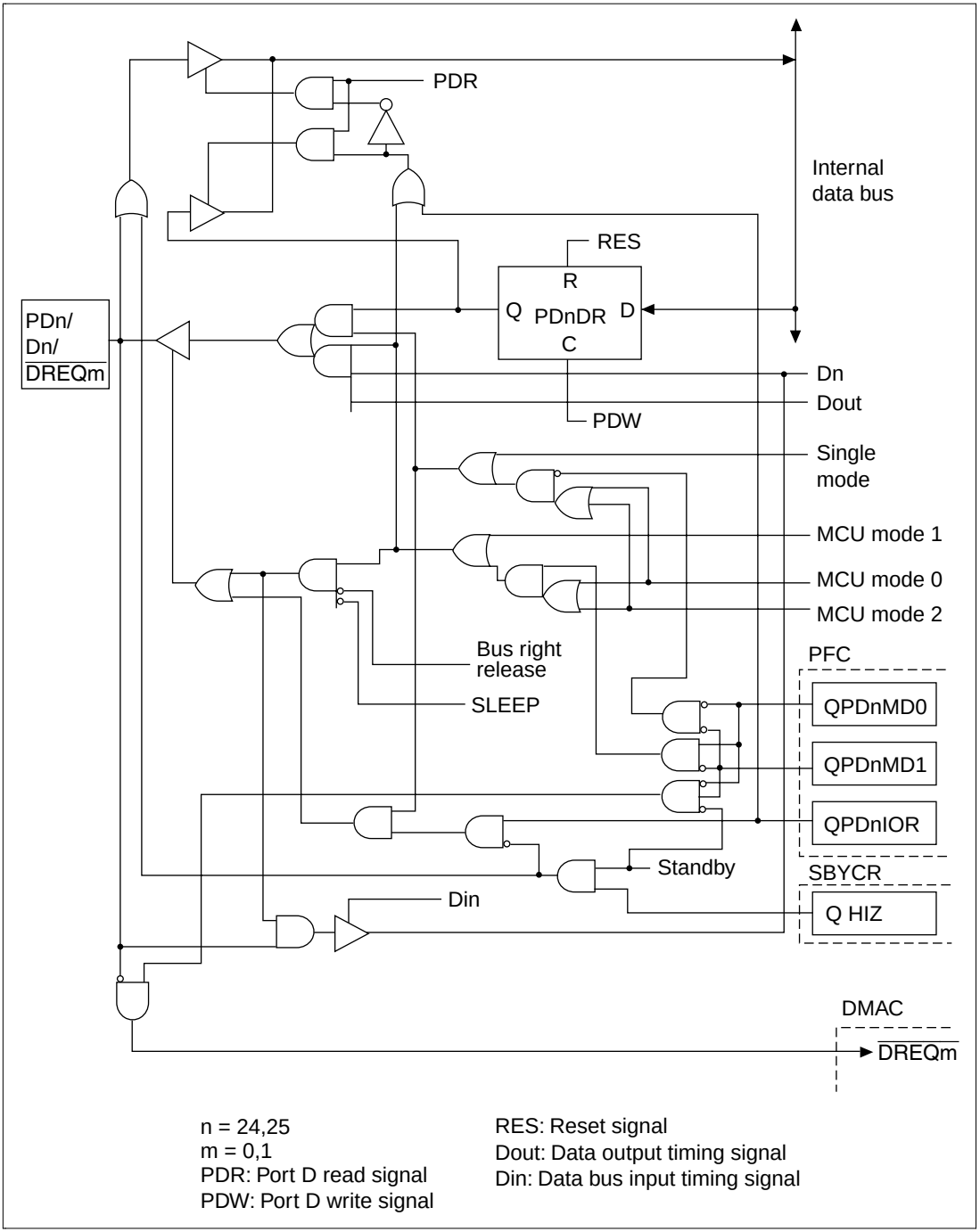


Figure B.27 PDn/Dn/$\overline{\text{DREQm}}$ Block Diagram (n = 24, 25)

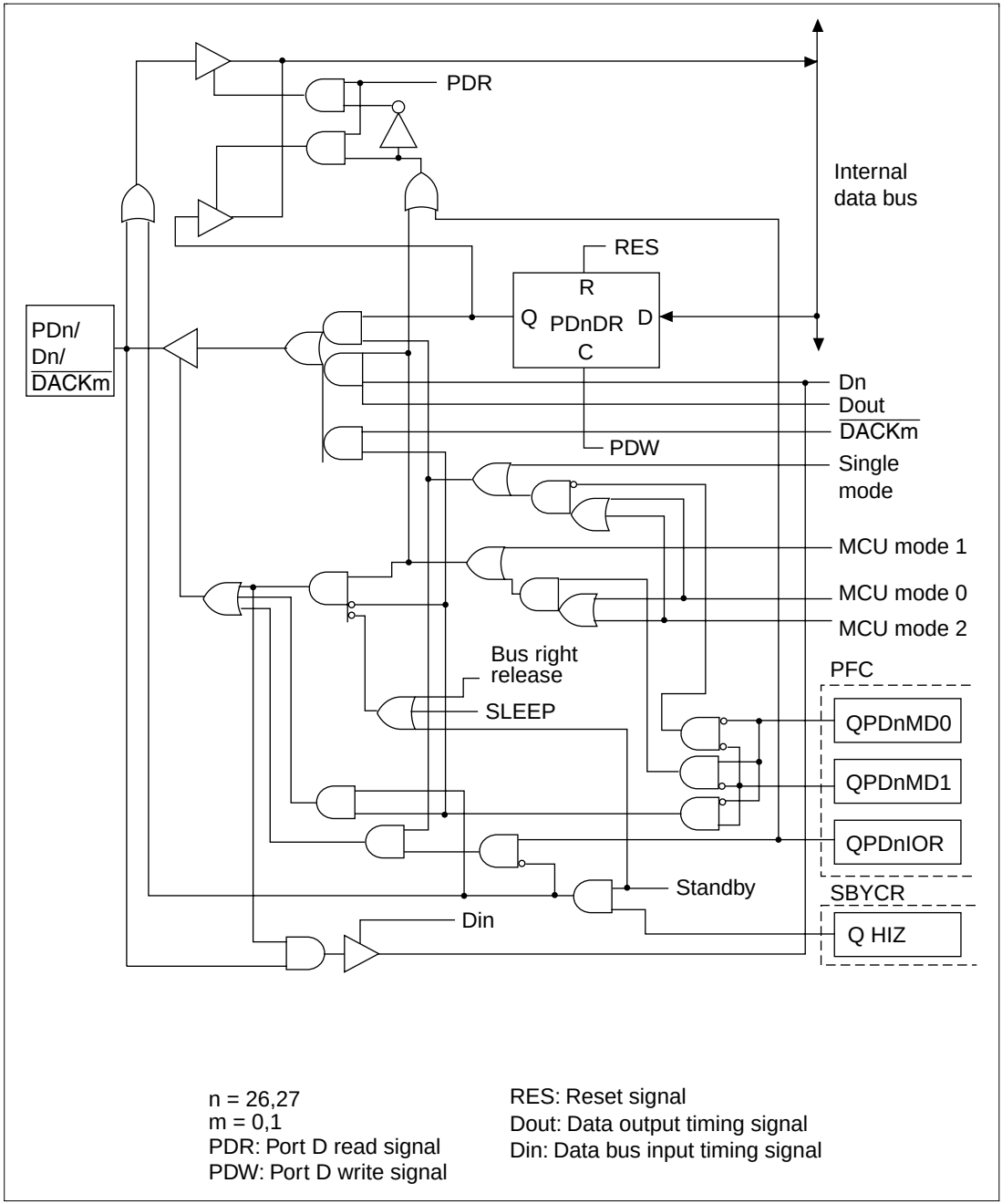
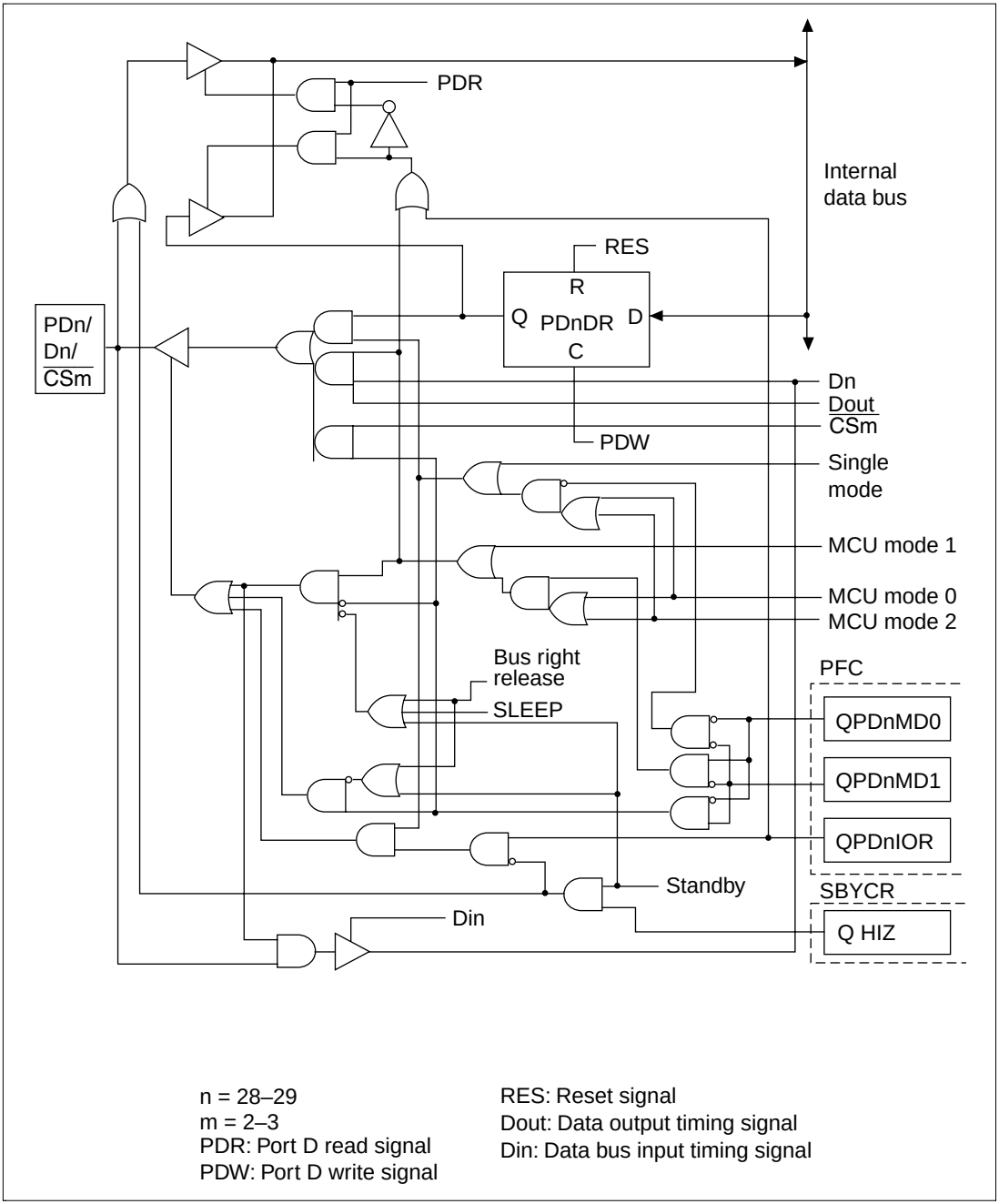


Figure B.28 PDn/Dn/$\overline{\text{DACKm}}$ Block Diagram (n = 26, 27)



n = 28–29
m = 2–3
PDR: Port D read signal
PDW: Port D write signal

RES: Reset signal
Dout: Data output timing signal
Din: Data bus input timing signal

Figure B.29 PDn/Dn/$\overline{\text{CSm}}$ Block Diagram (n = 28, 29)

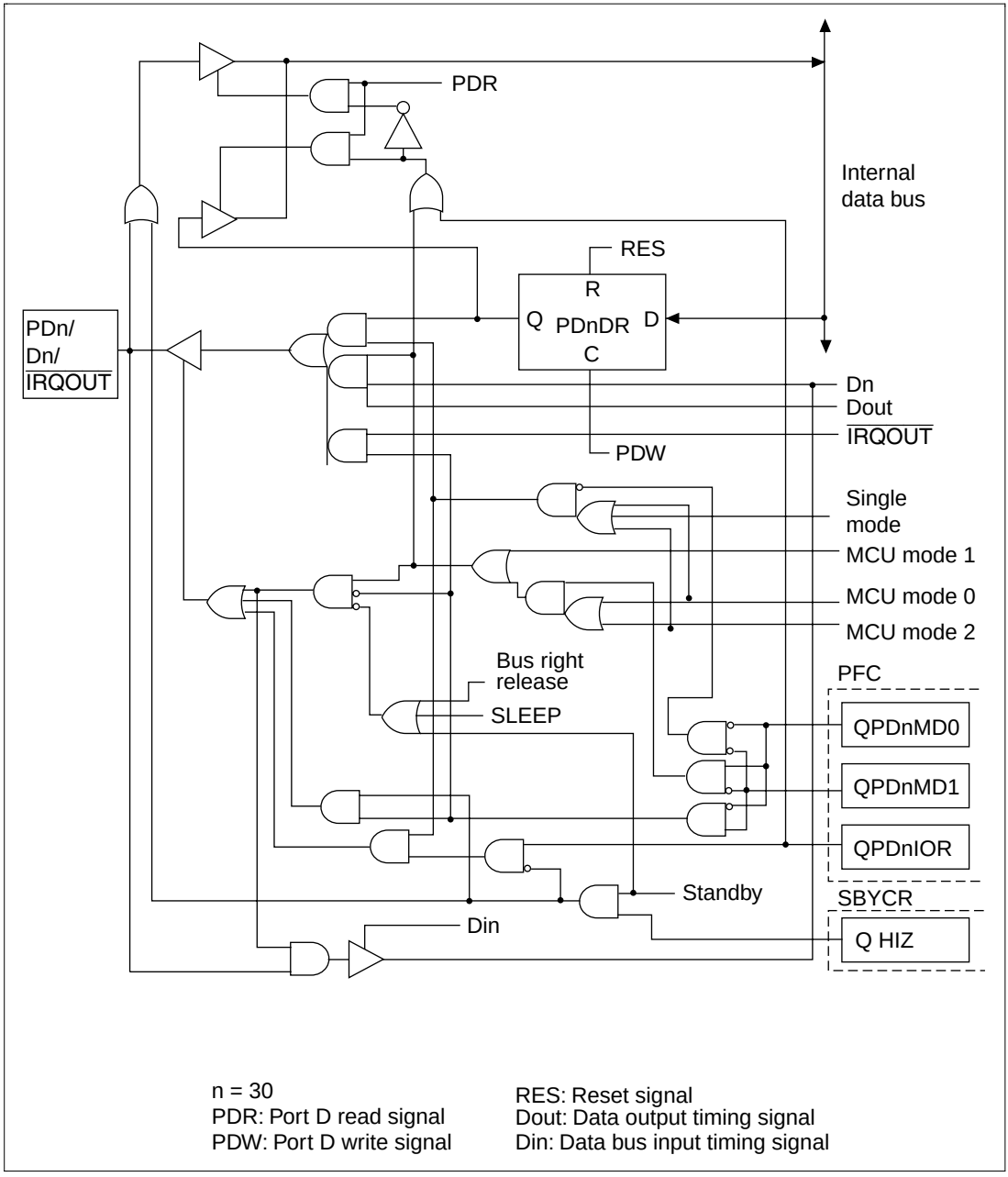


Figure B.30 PDn/Dn/$\overline{\text{IRQOUT}}$ Block Diagram (n = 30)

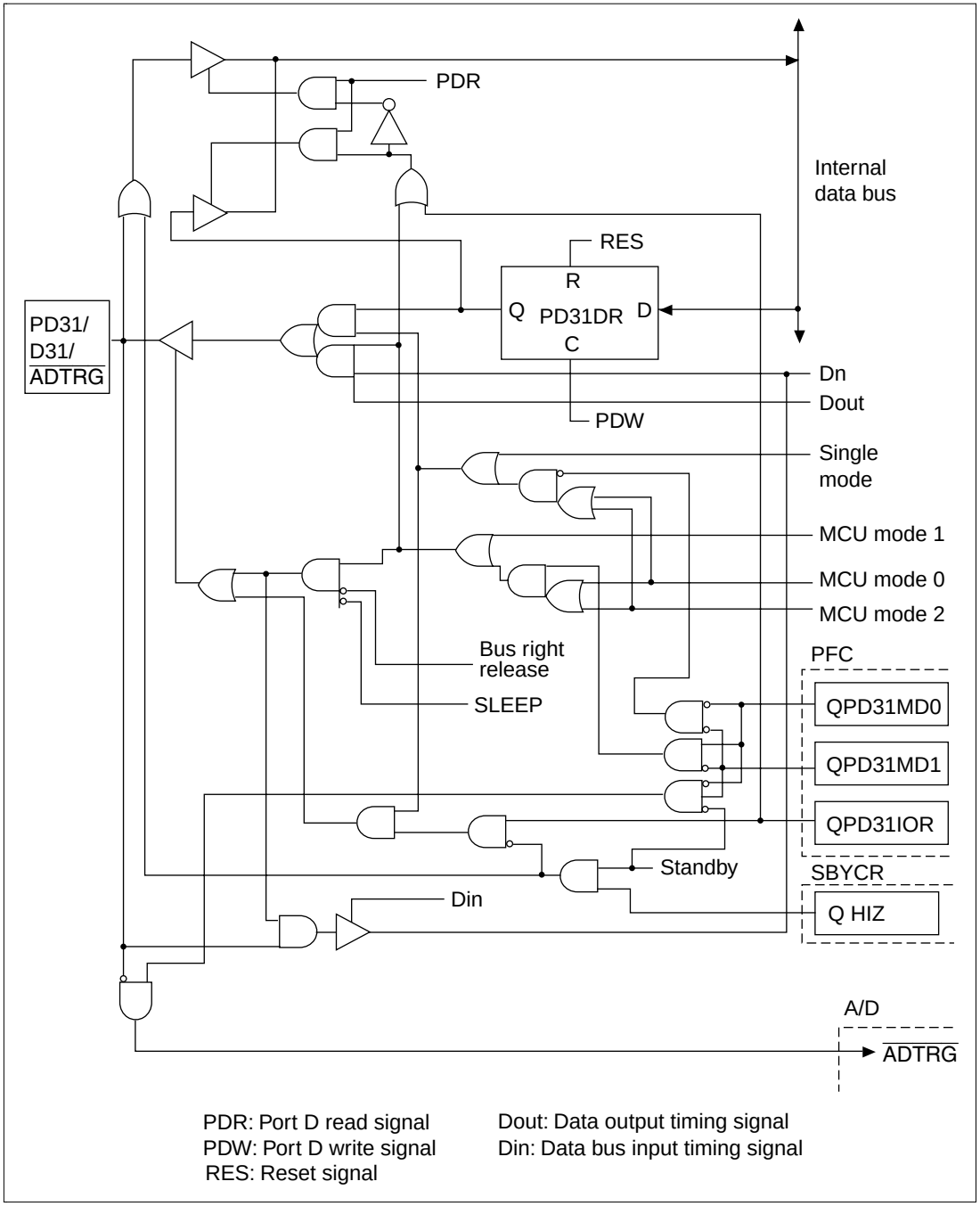


Figure B.31 PD31/D31/$\overline{\text{ADTRG}}$ Block Diagram

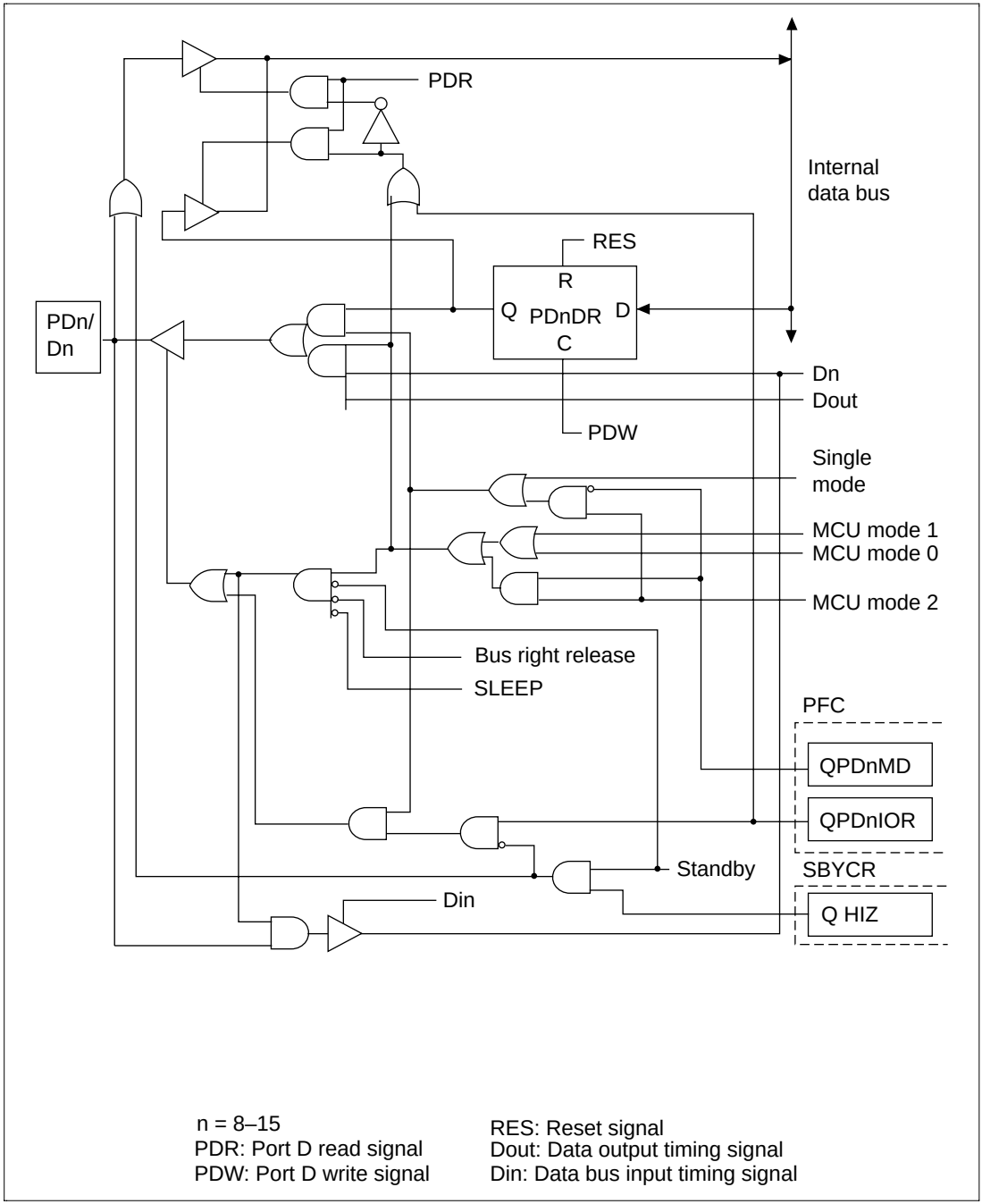


Figure B.32 PDn/Dn Block Diagram

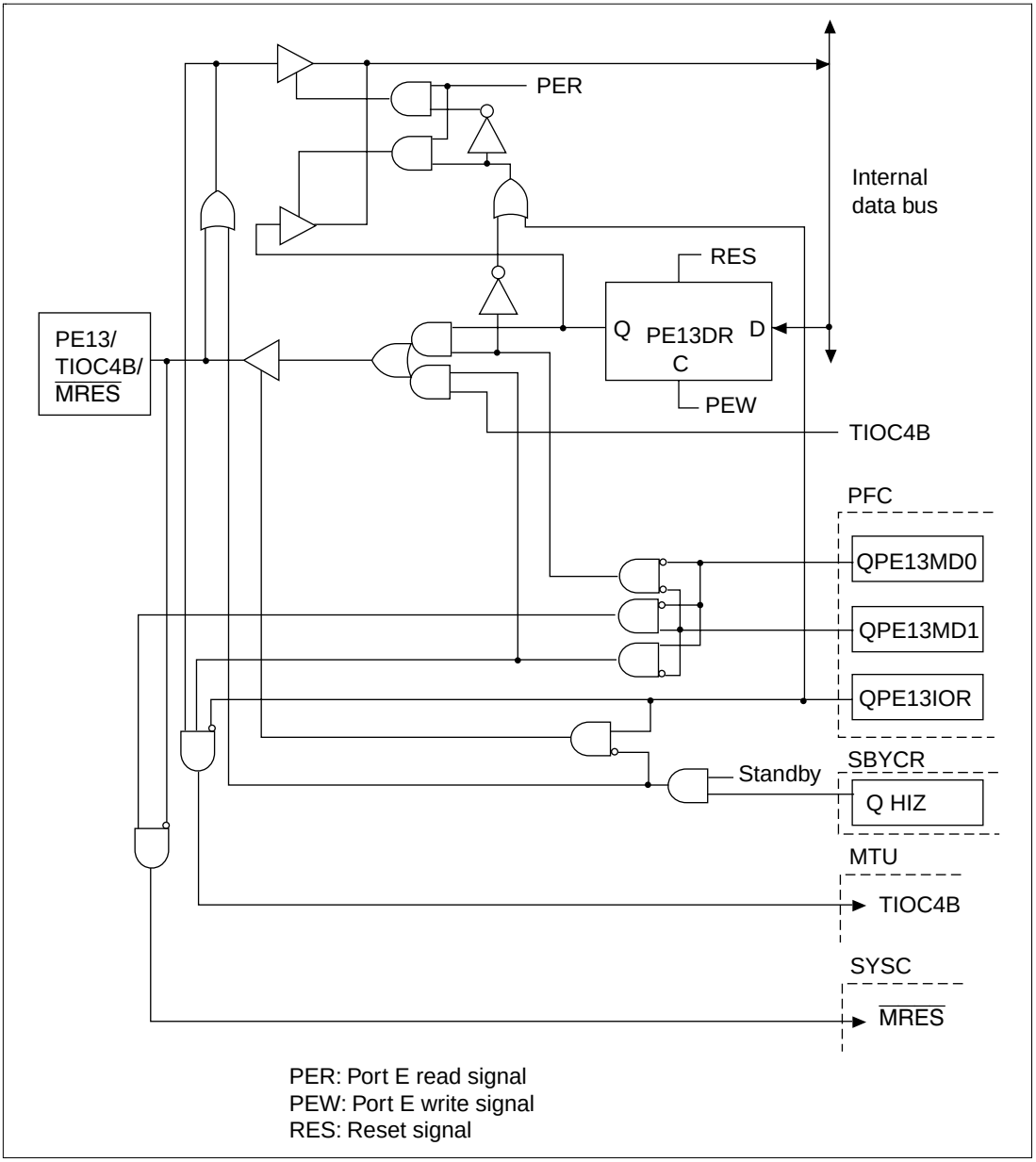


Figure B.33 PE13/TIOC4B/$\overline{\text{MRES}}$ Block Diagram

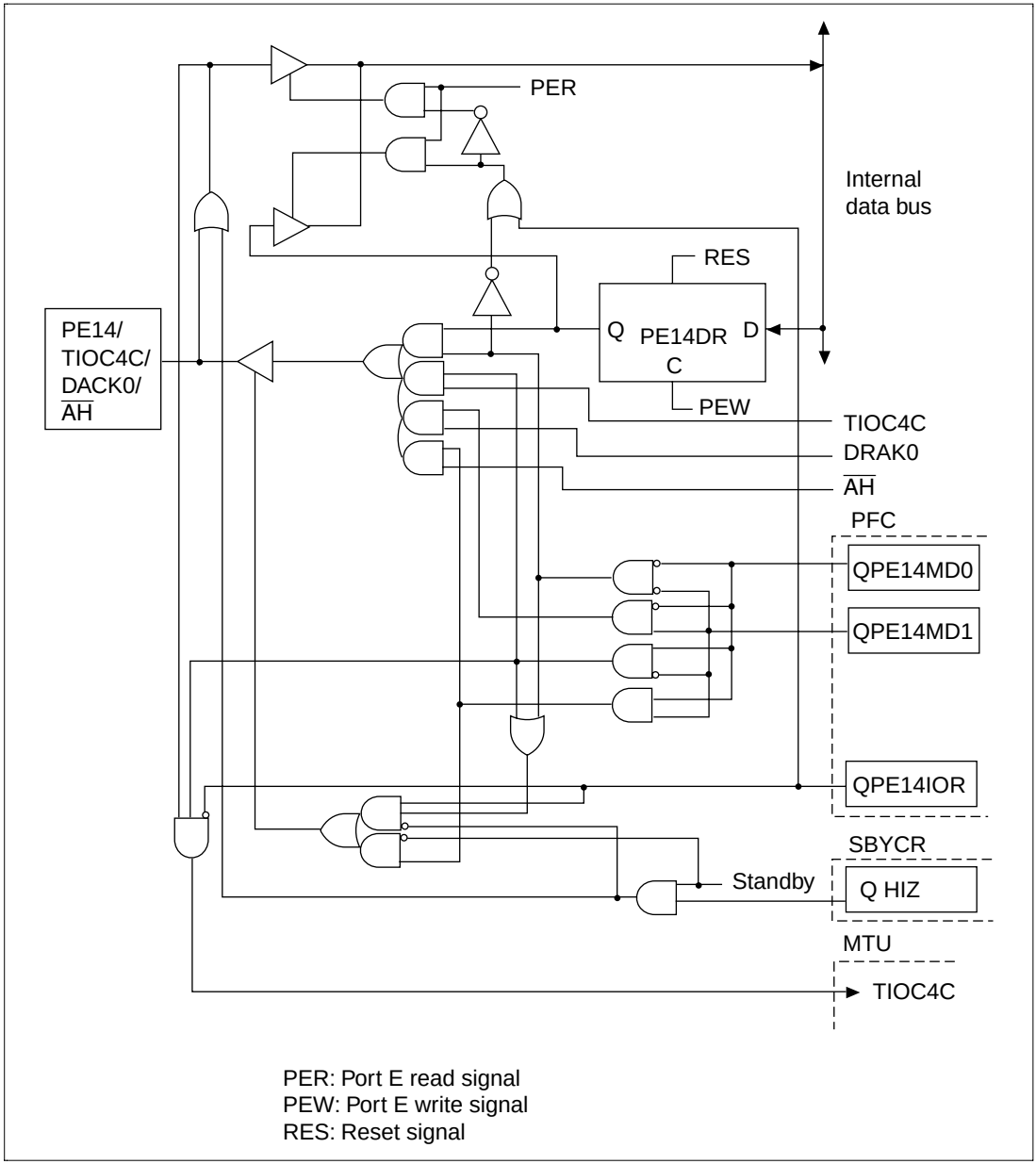


Figure B.34 PE14/TIOC4C/DACK0/$\overline{AH}$ Block Diagram

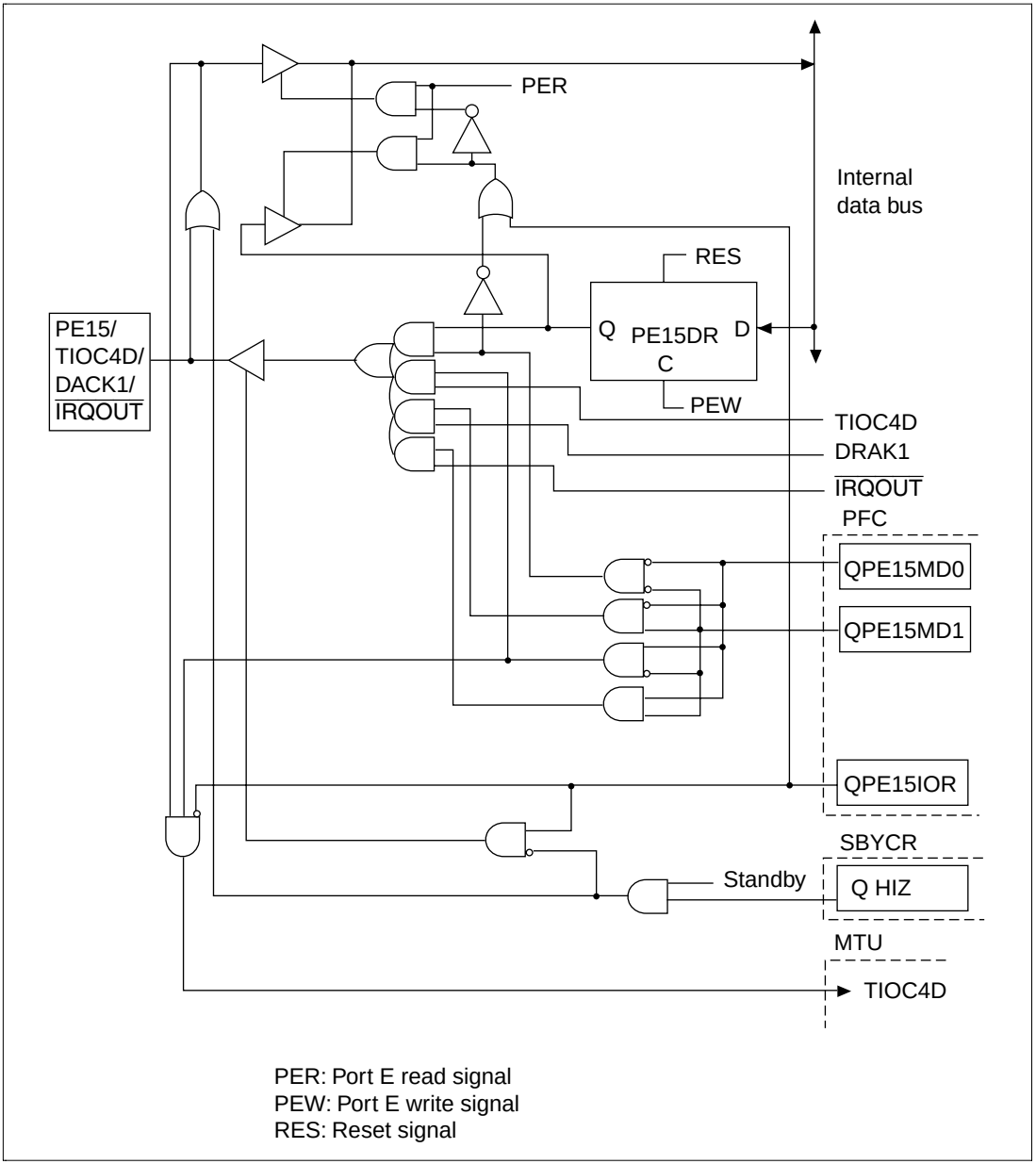


Figure B.35 PEn/TIOC4D/DACK1/$\overline{\text{IRQOUT}}$ Block Diagram

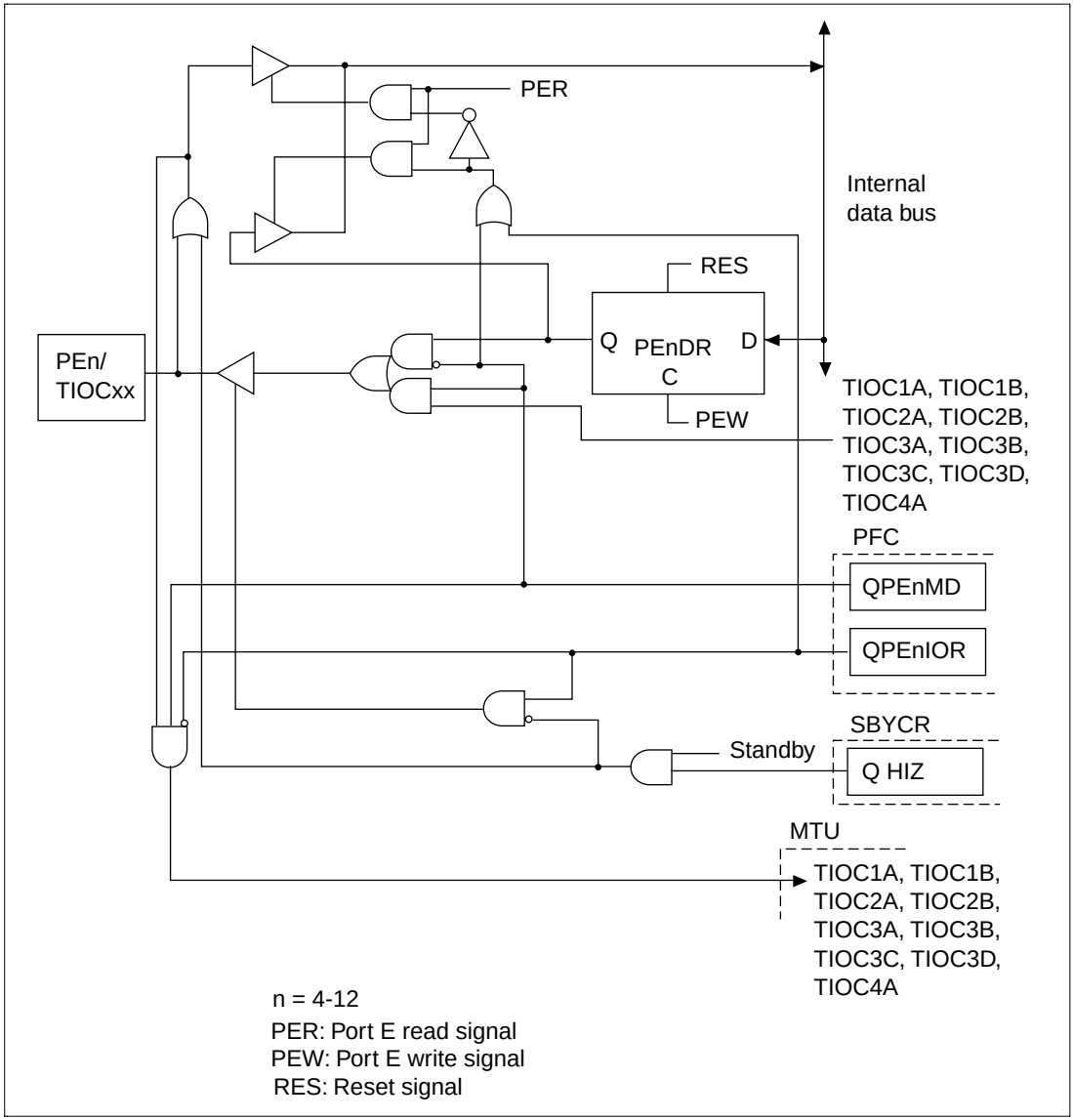


Figure B.36 PEn/TIOCXX Block Diagram

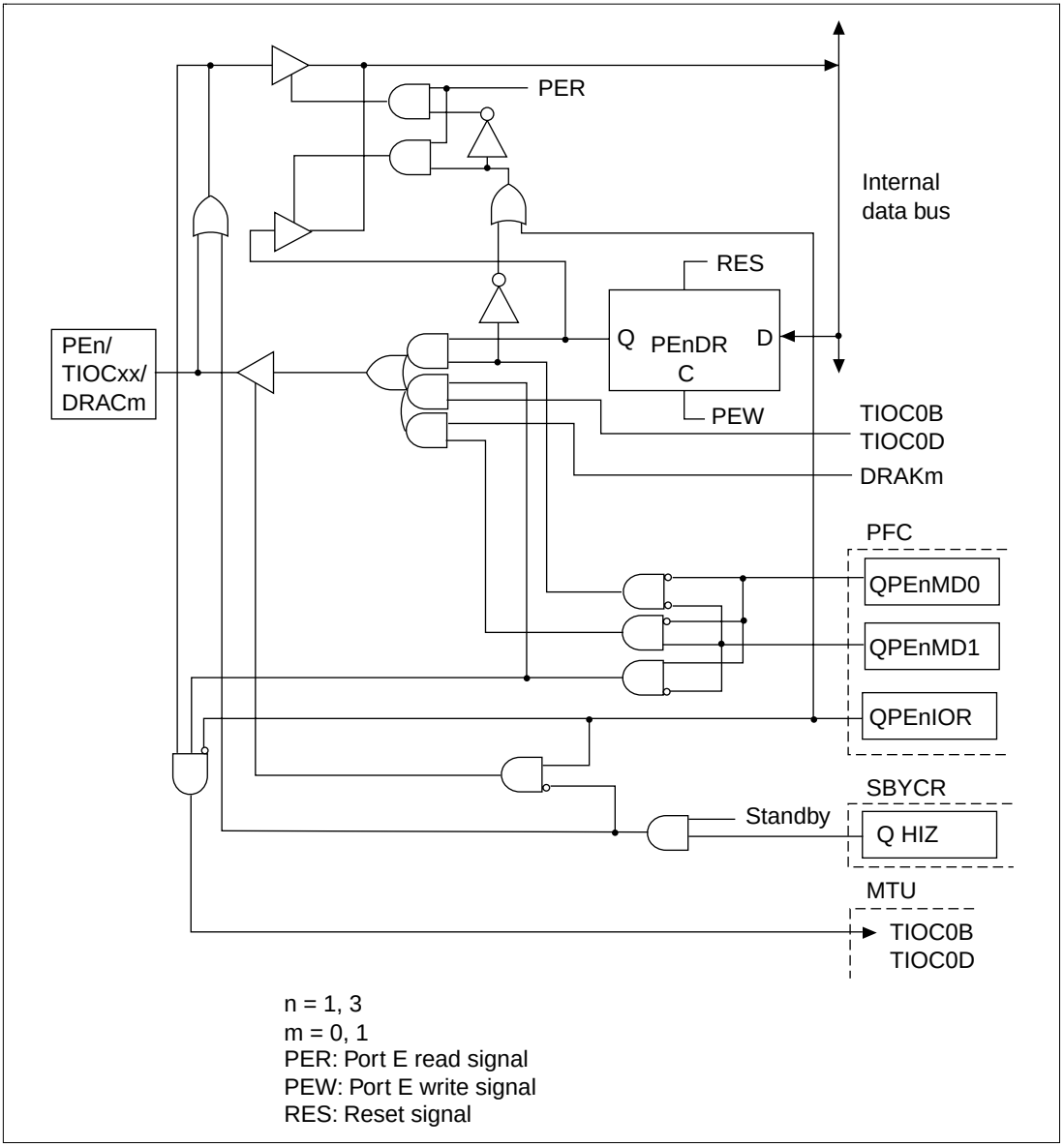


Figure B.37 PEn/TIOCXX/DRAKm Block Diagram

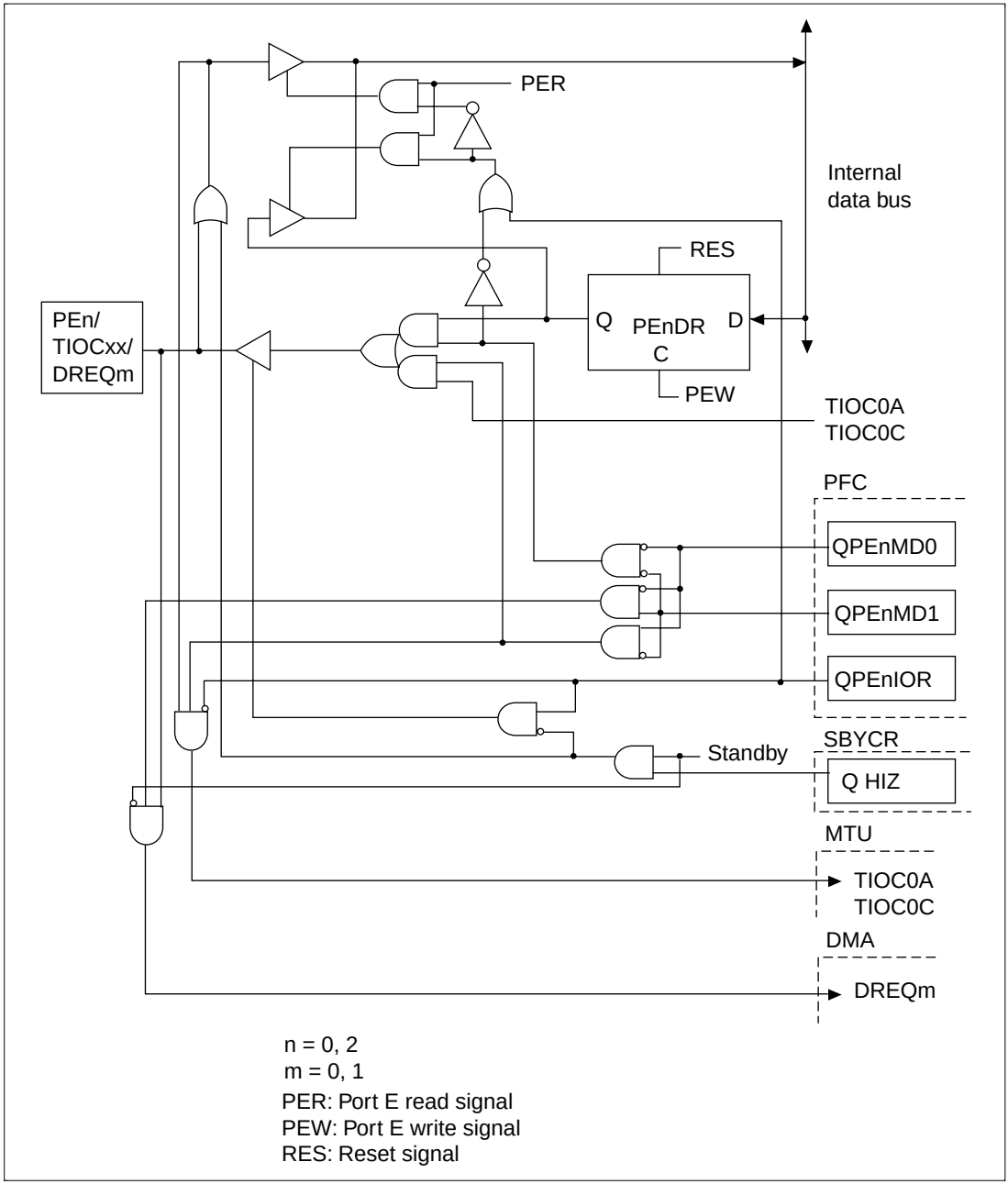


Figure B.38 PEn/TIOCXX/$\overline{\text{DREQm}}$ Block Diagram

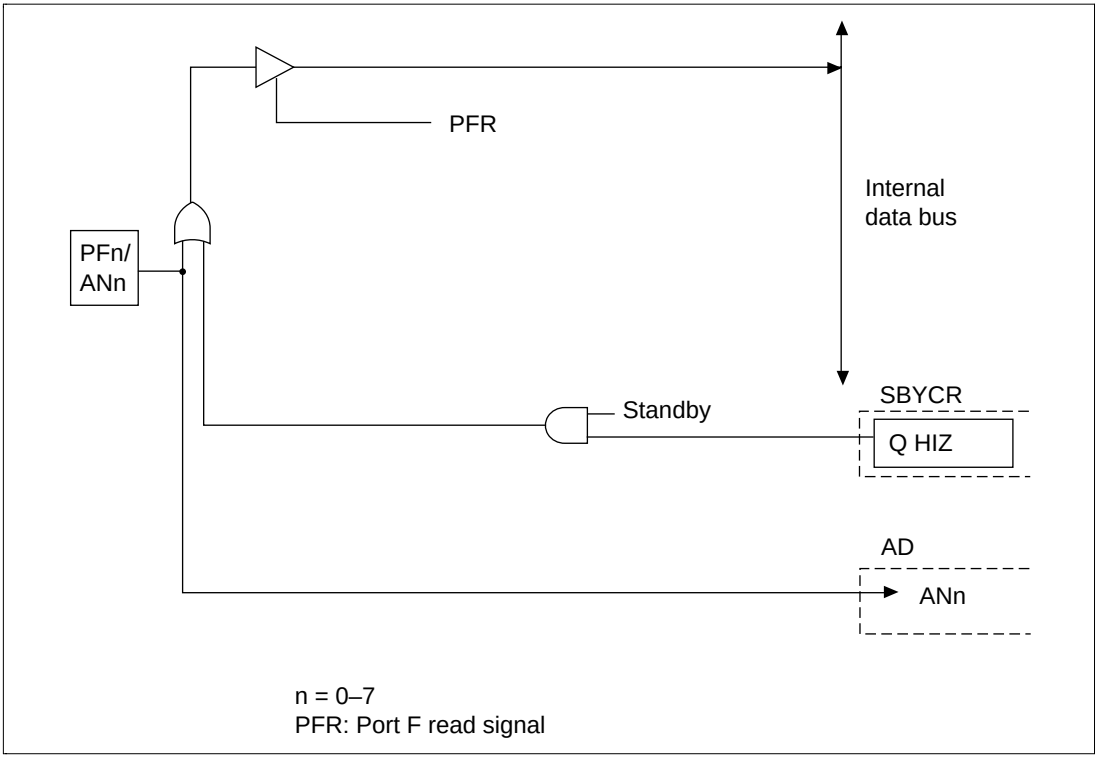


Figure B.39 PFn/ANn Block Diagram

Table C.1 Pin Modes During Reset, Power-Down, and Bus Right Release Modes (144 Pin)

Pin Function		Pin modes					
		Reset		Power-Down		Bus Right Release	Standby in Bus Right Release
Class	Pin Name	Power-On	Manual	Standby	Sleep		
Clock	CK	O	O	H*1	O	O	O
System control	$\overline{\text{RES}}$	I	I	I	I	I	I
	$\overline{\text{MRES}}$	Z*4	I	Z	I	I	Z
	$\overline{\text{WDTOVF}}$	O*3	O*3	O	O	O	O
	$\overline{\text{BREQ}}$	Z*4	I	Z	I	I	I
	$\overline{\text{BACK}}$	Z*4	O	Z	O	L	L
Interrupt	NMI	I	I	I	I	I	I
	$\overline{\text{IRQ0}}$–$\overline{\text{IRQ7}}$	Z*4	I	Z	I	I	Z
	$\overline{\text{IRQOUT}}$ (PD30)	Z*4	O	H*1	H	O	H*1
	$\overline{\text{IRQOUT}}$ (PE15)	Z*4	O	Z	H	O	Z
Address bus	A0–A21	O*2	O	Z	O	Z	Z
Data bus	D0–D31	Z*4	I/O	Z	I/O	Z	Z
Bus control	$\overline{\text{WAIT}}$	Z*4	I	Z	I	Z	Z
	RD/WR, $\overline{\text{RAS}}$	Z*4	O	O	O	Z	Z
	$\overline{\text{CASH}}$, $\overline{\text{CASL}}$, $\overline{\text{CASLH}}$, $\overline{\text{CASLL}}$	Z*4	O	O	O	Z	Z
	$\overline{\text{RD}}$	H	O	Z	O	Z	Z
	$\overline{\text{CS0}}$, $\overline{\text{CS1}}$	H	O	Z	O	Z	Z
	$\overline{\text{CS2}}$, $\overline{\text{CS3}}$	Z*4	O	Z	O	Z	Z
	$\overline{\text{WRHH}}$, $\overline{\text{WRHL}}$, $\overline{\text{WRH}}$, $\overline{\text{WRL}}$	H	O	Z	O	Z	Z
	$\overline{\text{AH}}$	Z*4	O	Z	O	Z	Z
DMAC	DACK0, DACK1 (PD26, PD27)	Z*4	O	O*1	O	O	O*1
	DACK0, DACK1 (PE14, PE15)	Z*4	O	Z	O	O	Z
	DRAK0, DRAK1	Z*4	O	O*1	O	O	O*1
	$\overline{\text{DREQ0}}$, $\overline{\text{DREQ1}}$	Z*4	I	Z	I	I	Z

Table C.1 Pin Modes During Reset, Power-Down, and Bus Right Release Modes (144 Pin) (cont)

Pin Function		Pin modes					
		Reset		Power-Down		Bus Right	Standby in Bus
Class	Pin Name	Power-On	Manual	Standby	Sleep	Release	Right Release
MTU	TIOC0A–TIOC0D, TIOC1A–TIOC1D, TIOC2A–TIOC2D, TIOC3A, TIOC3C	Z*4	I/O	K*1	I/O	I/O	K*1
	TIOC3B,TIOC3D, TIOC4A–TIOC4D	Z*4	I/O	Z	I/O	I/O	Z
	TCLKA–TCLKD	Z*4	I	Z	I	I	Z
Port control	$\overline{\text{POE0}}$–$\overline{\text{POE3}}$	Z*4	I	Z	I	I	Z
SCI	SCK0–SCK1	Z*4	I/O	Z	I/O	I/O	Z
	TXD0–TCD1	Z*4	O	O*1	O	O	O*1
	RXD0–RXD1	Z*4	I	Z	I	I	Z
A/D converter	$\overline{\text{ADTRG}}$	Z*4	I	Z	I	I	Z
	AN0–AN7	Z	I	Z	I	I	Z
I/O Port	PA0–PA23	Z*4	I/O	K*1	K	I/O	K*1
	PB0–PB9						
	PC0–PC15						
	PD0–PD31						
	PE0–PE8,PE10						
	PE9,PE11–PE15	Z*4	I/O	Z	K	I/O	Z
	PF0–PF17	Z	I	Z	I	I	Z

Notes: 1. There are instances where bus right release and transition to software standby mode occur simultaneously due to the timing between $\overline{\text{BREQ}}$ and internal operations. In such cases, standby mode results, but the standby state may be different.

The initial pin states depend on the mode. See section 18, Pin Function Controller, for details.

2. I: Input, O: Output, H: High-level output, L: Low-level output, Z: High impedance, K: Input pin with high impedance, output pin mode maintained.

*1 If the standby control register port high-impedance bits are set to 1, output pins become high impedance.

*2 A21–A18 will become input ports after power-on reset.

*3 Input in the SH7044/SH7045 F-ZTAT version.

*4 General use I/O ports PAn, PBn, PCn, PDn, and PEn, as well as pins multiplexed with them, are unstable during the $\overline{\text{RES}}$ setup time (t_{RESS}) immediately after the $\overline{\text{RES}}$ pin goes to low level.

Table C.2 Pin Modes During Reset, Power-Down, and Bus Right Release Modes (112 Pin, 120 Pin)

Pin Function		Pin modes					
		Reset		Power-Down		Bus Right	Standby in Bus
Class	Pin Name	Power-On	Manual	Standby	Sleep	Release	Right Release
Clock	CK	O	O	H*1	O	O	O
System control	$\overline{\text{RES}}$	I	I	I	I	I	I
	$\overline{\text{MRES}}$	Z*4	I	Z	I	I	Z
	$\overline{\text{WDTOVF}}$	O*3	O*3	O	O	O	O
	$\overline{\text{BREQ}}$	Z*4	I	Z	I	I	I
	$\overline{\text{BACK}}$	Z*4	O	Z	O	L	L
Interrupt	NMI	I	I	I	I	I	I
	$\overline{\text{IRQ0}}$–$\overline{\text{IRQ7}}$	Z*4	I	Z	I	I	Z
	$\overline{\text{IRQOUT}}$	Z*4	O	Z	H	O	Z
Address bus	A0–A21	O*2	O	Z	O	Z	Z
Data bus	D0–D31	Z*4	I/O	Z	I/O	Z	Z
Bus control	$\overline{\text{WAIT}}$	Z*4	I	Z	I	Z	Z
	RDWR, $\overline{\text{RAS}}$	Z*4	O	O	O	Z	Z
	$\overline{\text{CASH}}$, $\overline{\text{CASL}}$	Z*4	O	O	O	Z	Z
	$\overline{\text{RD}}$	H	O	Z	O	Z	Z
	$\overline{\text{CS0}}$, $\overline{\text{CS1}}$	H	O	Z	O	Z	Z
	$\overline{\text{CS2}}$, $\overline{\text{CS3}}$	Z*4	O	Z	O	Z	Z
	$\overline{\text{WRH}}$, $\overline{\text{WRL}}$	H	O	Z	O	Z	Z
	$\overline{\text{AH}}$	Z*4	O	Z	O	Z	Z
DMAC	DACK0–DACK1	Z*4	O	Z	O	O	Z
	DRAK0–DRAK1	Z*4	O	Z	O	O	Z
	$\overline{\text{DREQ0}}$–$\overline{\text{DREQ1}}$	Z*4	I	Z	I	I	Z
MTU	TIOC0A–TIOC0D, TIOC1A–TIOC1D, TIOC2A–TIOC2D, TIOC3A, TIOC3C	Z*4	I/O	K*1	I/O	I/O	K*1
	TIOC3B,TIOC3D, TIOC4A–TIOC4D	Z*4	I/O	Z	I/O	I/O	Z
	TCLKA–TCLKD	Z*4	I	Z	I	I	Z

Table C.2 Pin Modes During Reset, Power-Down, and Bus Right Release Modes (112 Pin, 120 Pin) (cont)

Pin Function		Pin modes					
		Reset		Power-Down		Bus Right	Standby in Bus
Class	Pin Name	Power-On	Manual	Standby	Sleep	Release	Right Release
Port control	$\overline{\text{POE0}}$–$\overline{\text{POE3}}$	Z*4	I	Z	I	I	Z
SCI	SCK0–SCK1	Z*4	I/O	Z	I/O	I/O	Z
	TXD0–TCD1	Z*4	O	O*1	O	O	O*1
	RXD0–RXD1	Z*4	I	Z	I	I	Z
A/D converter control	$\overline{\text{ADTRG}}$	Z*4	I	Z	I	I	Z
	AN0–AN7	Z	I	Z	I	I	Z
I/O Port	PA0–PA15	Z*4	I/O	K*1	K	I/O	K*1
	PB0–PB9						
	PC0–PC15						
	PD0–PD15						
	PE0–PE8–PE10						
	PE9,PE11–PE15	Z*4	I/O	Z	K	I/O	Z
	PF0–PF7	Z	I	Z	I	I	Z

Notes: 1. There are instances where bus right release and transition to software standby mode occur simultaneously due to the timing between $\overline{\text{BREQ}}$ and internal operations. In such cases, standby mode results, but the standby state may be different.

The initial pin states depend on the mode. See section 18, Pin Function Controller, for details.

2. I: Input, O: Output, H: High-level output, L: Low-level output, Z: High impedance, K: Input pin with high impedance, output pin mode maintained.

*1 If the standby control register port high-impedance bits are set to 1, output pins become high impedance.

*2 A21–A18 will become input ports after power-on reset.

*3 Input in the SH7044/SH7045 F-ZTAT version.

*4 General use I/O ports PAn, PBn, PCn, PDn, and PEn, as well as pins multiplexed with them, are unstable during the $\overline{\text{RES}}$ setup time (t_{RESS}) immediately after the $\overline{\text{RES}}$ pin goes to low level.

Table C.3 Pin Settings for On-Chip Peripheral Modules

		On-Chip Peripheral Module					
					16-Bit Space		
Pin Name		On-Chip ROM	On-Chip RAM	8-Bit Space	Upper Byte	Lower Byte	Word/ Longword
$\overline{CS0}$–$\overline{CS3}$		H	H	H	H	H	H
$\overline{RAS}$*1		H	H	H	H	H	H
$\overline{CASHH}$*2		H	H	H	H	H	H
$\overline{CASHL}$*2		H	H	H	H	H	H
$\overline{CASLH}$*2		H	H	H	H	H	H
$\overline{CASLL}$*2		H	H	H	H	H	H
RD/WR		H	H	H	H	H	H
$\overline{AH}$		L	L	L	L	L	L
$\overline{RD}$	R	H	H	H	H	H	H
	W	—	H	H	H	H	H
$\overline{WRHH}$	R	H	H	H	H	H	H
	W	—	H	H	H	H	H
$\overline{WRHL}$	R	H	H	H	H	H	H
	W	—	H	H	H	H	H
$\overline{WRLH}$	R	H	H	H	H	H	H
	W	—	H	H	H	H	H
$\overline{WRLL}$	R	H	H	H	H	H	H
	W	—	H	H	H	H	H
A21–A0		Address	Address	Address	Address	Address	Address
D31–D24		High-Z	High-Z	High-Z	High-Z	High-Z	High-Z
D23–D16		High-Z	High-Z	High-Z	High-Z	High-Z	High-Z
D15–D8		High-Z	High-Z	High-Z	High-Z	High-Z	High-Z
D7–D0		High-Z	High-Z	High-Z	High-Z	High-Z	High-Z

Notes: R: Read, W: Write

*1 L asserted in RAS down state or refresh state.

*2 L asserted in refresh state.

Table C.4 Pin Settings for Normal External Space

			16-Bit Space		
Pin Name		**8-Bit Space**	**Upper Word**	**Lower Word**	**Word/Longword**
$\overline{CS0}$–$\overline{CS3}$		Valid	Valid	Valid	Valid
$\overline{RAS}$*1		H	H	H	H
$\overline{CASHH}$*2		H	H	H	H
$\overline{CASHL}$*2		H	H	H	H
$\overline{CASLH}$*2		H	H	H	H
$\overline{CASLL}$*2		H	H	H	H
RD/WR		H	H	H	H
$\overline{AH}$		L	L	L	L
$\overline{RD}$	R	L	L	L	L
	W	H	H	H	H
$\overline{WRHH}$	R	H	H	H	H
	W	H	H	H	H
$\overline{WRHL}$	R	H	H	H	H
	W	H	H	H	H
$\overline{WRLH}$	R	H	H	H	H
	W	H	L	H	L
$\overline{WRLL}$	R	H	H	H	H
	W	L	H	L	L
A21–A0		Address	Address	Address	Address
D31–D24		High-Z	High-Z	High-Z	High-Z
D23–D16		High-Z	High-Z	High-Z	High-Z
D15–D8		High-Z	Data	High-Z	Data
D7–D0		Data	High-Z	Data	Data

Table C.4 Pin Settings for Normal External Space (cont)

		32-Bit Space						
Pin Name		**MSB**	**2nd Byte**	**3rd Byte**	**LSB**	**Upper Word**	**Lower Word**	**Longword**
$\overline{CS0}$–$\overline{CS3}$		Valid	Valid	Valid	Valid	Valid	Valid	Valid
$\overline{RAS}$*1		H	H	H	H	H	H	H
$\overline{CASHH}$*2		H	H	H	H	H	H	H
$\overline{CASHL}$*2		H	H	H	H	H	H	H
$\overline{CASLH}$*2		H	H	H	H	H	H	H
$\overline{CASLL}$*2		H	H	H	H	H	H	H
RD/WR		H	H	H	H	H	H	H
$\overline{AH}$		L	L	L	L	L	L	L
$\overline{RD}$	R	L	L	L	L	L	L	L
	W	H	H	H	H	H	H	H
$\overline{WRHH}$	R	H	H	H	H	H	H	H
	W	L	H	H	H	L	H	L
$\overline{WRHL}$	R	H	H	H	H	H	H	H
	W	H	L	H	H	L	H	L
$\overline{WRH}$	R	H	H	H	H	H	H	H
	W	H	H	L	H	H	L	L
$\overline{WRL}$	R	H	H	H	H	H	H	H
	W	H	H	H	L	H	L	L
A21–A0		Address	Address	Address	Address	Address	Address	Address
D31–D24		Data	High-Z	High-Z	High-Z	Data	High-Z	Data
D23–D16		High-Z	Data	High-Z	High-Z	Data	High-Z	Data
D15–D8		High-Z	High-Z	Data	High-Z	High-Z	Data	Data
D7–D0		High-Z	High-Z	High-Z	Data	High-Z	Data	Data

Notes: 1. R: Read, W: Write

2. Valid: Chip select signal corresponding with accessed area is low; chip select signal in other cases is high.

*1 L asserted in RAS down mode or refresh mode.

*2 L asserted in refresh mode.

Table C.5 Pin Settings for Multiplex I/O Space

			16-Bit Space		
Pin Name		**8-Bit Space**	**Upper Byte**	**Lower Byte**	**Word/Longword**
$\overline{CS0}$–$\overline{CS2}$		H	H	H	H
$\overline{CS3}$		L	L	L	L
$\overline{RAS}$*1		H	H	H	H
$\overline{CASHH}$*2		H	H	H	H
$\overline{CASHL}$*2		H	H	H	H
$\overline{CASLH}$*2		H	H	H	H
$\overline{CASLL}$*2		H	H	H	H
RD/WR		H	H	H	H
$\overline{AH}$		Valid	Valid	Valid	Valid
$\overline{RD}$	R	L	L	L	L
	W	H	H	H	H
$\overline{WRHH}$	R	H	H	H	H
	W	H	H	H	H
$\overline{WRHL}$	R	H	H	H	H
	W	H	H	H	H
$\overline{WRH}$	R	H	H	H	H
	W	H	L	H	L
$\overline{WRL}$	R	H	H	H	H
	W	L	H	L	L
A21–A0		Address	Address	Address	Address
D31–D24		High-Z	High-Z	High-Z	High-Z
D23–D16		High-Z	High-Z	High-Z	High-Z
D15–D8		High-Z	Address/Data	Address	Address/Data
D7–D0		Address/Data	Address	Address/Data	Address/Data

Notes: 1. R: Read, W: Write

2. Valid: High output in accordance with $\overline{AH}$ timing.

*1 L asserted in RAS down mode or refresh mode.

*2 L asserted in refresh mode.

Table C.6 Pin Settings for DRAM Space

			16-Bit Space		
Pin Name		**8-Bit Space**	**Upper Word**	**Lower Word**	**Word/Longword**
$\overline{CS0}$–$\overline{CS3}$		H	H	H	H
$\overline{RAS}$*1		Valid	Valid	Valid	Valid
$\overline{CASHH}$*2		H	H	H	H
$\overline{CASHL}$*2		H	H	H	H
$\overline{CASLH}$*2		H	Valid	H	Valid
$\overline{CASLL}$*2		Valid	H	Valid	Valid
RD/WR	R	H	H	H	H
	W	L	L	L	L
$\overline{AH}$		L	L	L	L
$\overline{RD}$	R	L	L	L	L
	W	H	H	H	H
$\overline{WRHH}$	R	H	H	H	H
	W	H	H	H	H
$\overline{WRHL}$	R	H	H	H	H
	W	H	H	H	H
$\overline{WRH}$	R	H	H	H	H
	W	H	L	H	L
$\overline{WRL}$	R	H	H	H	H
	W	L	H	L	L
A21–A0		Address	Address	Address	Address
D31–D24		High-Z	High-Z	High-Z	High-Z
D23–D16		High-Z	High-Z	High-Z	High-Z
D15–D8		High-Z	Data	High-Z	Data
D7–D0		Data	High-Z	Data	Data

Table C.6 Pin Settings for DRAM Space (cont)

		32-Bit Space						
Pin Name		**MSB**	**2nd Byte**	**3rd Byte**	**LSB**	**Upper Word**	**Lower Word**	**Longword**
$\overline{CS0}$–$\overline{CS3}$		H	H	H	H	H	H	H
$\overline{RAS}$*1		Valid	Valid	Valid	Valid	Valid	Valid	Valid
$\overline{CASHH}$*2		Valid	H	H	H	Valid	H	Valid
$\overline{CASHL}$*2		H	Valid	H	H	Valid	H	Valid
$\overline{CASLH}$*2		H	H	Valid	H	H	Valid	Valid
$\overline{CASLL}$*2		H	H	H	Valid	H	Valid	Valid
RD/WR	R	H	H	H	H	H	H	H
	W	L	L	L	L	L	L	L
$\overline{AH}$		L	L	L	L	L	L	L
$\overline{RD}$	R	L	L	L	L	L	L	L
	W	H	H	H	H	H	H	H
$\overline{WRHH}$	R	H	H	H	H	H	H	H
	W	L	H	H	H	L	H	L
$\overline{WRHL}$	R	H	H	H	H	H	H	H
	W	H	L	H	H	L	H	L
$\overline{WRH}$	R	H	H	H	H	H	H	H
	W	H	H	L	H	H	L	L
$\overline{WRL}$	R	H	H	H	H	H	H	H
	W	H	H	H	L	H	L	L
A21–A0		Address	Address	Address	Address	Address	Address	Address
D31–D24		Data	High-Z	High-Z	High-Z	Data	High-Z	Data
D23–D16		High-Z	Data	High-Z	High-Z	Data	High-Z	Data
D15–D8		High-Z	High-Z	Data	High-Z	High-Z	Data	Data
D7–D0		High-Z	High-Z	High-Z	Data	High-Z	Data	Data

Notes: 1. R: Read, W: Write

2. Valid: Chip select signal corresponding with accessed area is low; chip select signal in other cases is high.

*1 Asserted in RAS down mode or refresh mode.

*2 Asserted in refresh mode.

Appendix D Notes when Converting the F–ZTAT Application Software to the Mask-ROM Versions

Please note the following when converting the F-ZTAT application software to the mask-ROM versions.

The values read from the internal registers for the flash ROM of the mask-ROM version and F–ZTAT version differ as follows.

		Status	
Register	**Bit**	**F–ZTAT Version**	**Mask-ROM Version**
FLMCR1	FWE	0: Application software running	0: Is not read out
		1: Programming	1: Application software running

Note: This difference applies to all the F-ZTAT versions and all the mask-ROM versions that have different ROM size.

Appendix E Product Code Lineup

Table E.1 SH7040, SH7041, SH7042, SH7043, SH7044, and SH7045 Product Lineup

Product Type		Mask Version	Product Code	Mark Code	Package	Order Model No.*2
SH7040A	Mask ROM verion	A MASK	HD6437040AF28	HD6437040A (***)F28	QFP2020-112	HD6437040A***F
			HD6437040AVF16	HD6437040A(***)VF16	QFP2020-112	HD6437040A***F
			HD6437040AVX16	HD6437040A(***)VX16	TQFP1414-120	HD6437040A***X
			HD6437040ACF28	HD6437040A(***)CF28	QFP2020-112Cu*1	HD6437040A***CF
			HD6437040AVCF16	HD6437040A(***)VCF16	QFP2020-112Cu*1	HD6437040A***CF
	ROM less version	A MASK	HD6417040AF28	HD6417040AF28	QFP2020-112	HD6417040AF28
			HD6417040AVF16	HD6417040AVF16	QFP2020-112	HD6417040AVF16
			HD6417040AVX16	HD6417040AVX16	TQFP1414-120	HD6417040AVX16
			HD6417040ACF28	HD6417040ACF28	QFP2020-112Cu*1	HD6417040ACF28
			HD6417040AVCF16	HD6417040AVCF16	QFP2020-112Cu*1	HD6417040AVCF16
SH7041A	Mask ROM version	A MASK	HD6437041AF28	HD6437041A(***)F28	QFP2020-144	HD6437041A***F
			HD6437041AVF16	HD6437041A(***)VF16	QFP2020-144	HD6437041A***F
			HD6437041ACF28	HD6437041A(***)CF28	QFP2020-144Cu*1	HD6437041A***CF
			HD6437041AVCF16	HD6437041A(***)VCF16	QFP2020-144Cu*1	HD6437041A***CF
	ROM less verion	A MASK	HD6417041AF28	HD6417041AF28	QFP2020-144	HD6417041AF28
			HD6417041AVF16	HD6417041AVF16	QFP2020-144	HD6417041AVF16
			HD6417041ACF28	HD6417041ACF28	QFP2020-144Cu*1	HD6417041ACF28
			HD6417041AVCF16	HD6417041AVCF16	QFP2020-144Cu*1	HD6417041AVCF16
SH7042	Mask ROM version	–	HD6437042F28	HD6437042 (***)F28	QFP2020-112	HD6437042***F
			HD6437042VF16	HD6437042 (***)VF16	QFP2020-112	HD6437042***F
	Z-TAT version	–	HD6477042F28	HD6477042F28	QFP2020-112	HD6477042F28
			HD6477042VF16	HD6477042VF16	QFP2020-112	HD6477042VF16
SH7042A	Mask ROM version	A MASK	HD6437042AF28	HD6437042A(***)F28	QFP2020-112	HD6437042A***F
			HD6437042AVF16	HD6437042A(***)VF16	QFP2020-112	HD6437042A***F
			HD6437042AVX16	HD6437042A(***)VX16	TQFP1414-120	HD6437042A***X
			HD6437042ACF28	HD6437042A(***)CF28	QFP2020-112Cu*1	HD6437042A***CF
			HD6437042AVCF16	HD6437042A(***)VCF16	QFP2020-112Cu*1	HD6437042A***CF

Table E.1 SH7040, SH7041, SH7042, SH7043, SH7044, and SH7045 Product Lineup (cont)

Product Type		Mask Version	Product Code	Mark Code	Package	Order Model No.*2
SH7042A	Z-TAT version	A MASK	HD6477042AF28	HD6477042AF28	QFP2020-112	HD6477042AF28
			HD6477042AVF16	HD6477042AVF16	QFP2020-112	HD6477042AVF16
			HD6477042AVX16	HD6477042AVX16	TQFP1414-120	HD6477042AVX16
			HD6477042ACF28	HD6477042ACF28	QFP2020-112Cu*1	HD6477042ACF28
			HD6477042AVCF16	HD6477042AVCF16	QFP2020-112Cu*1	HD6477042AVCF16
SH7043	Mask ROM version	–	HD6437043F28	HD6437043(***)F28	QFP2020-144	HD6437043***F
			HD6437043VF16	HD6437043(***)VF16	QFP2020-144	HD6437043***F
	Z-TAT version	–	HD6477043F28	HD6477043F28	QFP2020-144	HD6477043F28
			HD6477043VF16	HD6477043VF16	QFP2020-144	HD6477043VF16
SH7043A	Mask ROM version	A MASK	HD6437043AF28	HD6437043A(***)F28	QFP2020-144	HD6437043A***F
			HD6437043AVF16	HD6437043A(***)VF16	QFP2020-144	HD6437043A***F
			HD6437043ACF28	HD6437043A(***)CF28	QFP2020-144Cu*1	HD6437043A***CF
			HD6437043AVCF16	HD6437043A(***)VCF16	QFP2020-144Cu*1	HD6437043A***CF
	Z-TAT version	A MASK	HD6477043AF28	HD6477043AF28	QFP2020-144	HD6477043AF28
			HD6477043AVF16	HD6477043AVF16	QFP2020-144	HD6477043AVF16
			HD6477043ACF28	HD6477043ACF28	QFP2020-144Cu*1	HD6477043ACF28
			HD6477043AVCF16	HD6477043AVCF16	QFP2020-144Cu*1	HD6477043AVCF16
SH7044	Mask ROM version	A MASK	HD6437044F28	HD6437044(***)F28	QFP2020-112	HD6437044***F
	F-ZTAT version		HD64F7044F28	HD64F7044F28	QFP2020-112	HD64F7044F28
SH7045	Mask ROM version	A MASK	HD6437045F28	HD6437045(***)F28	QFP2020-144	HD6437045***F
	F-ZTAT version		HD64F7045F28	HD64F7045F28	QFP2020-144	HD64F7045F28

(***) is the ROM code.

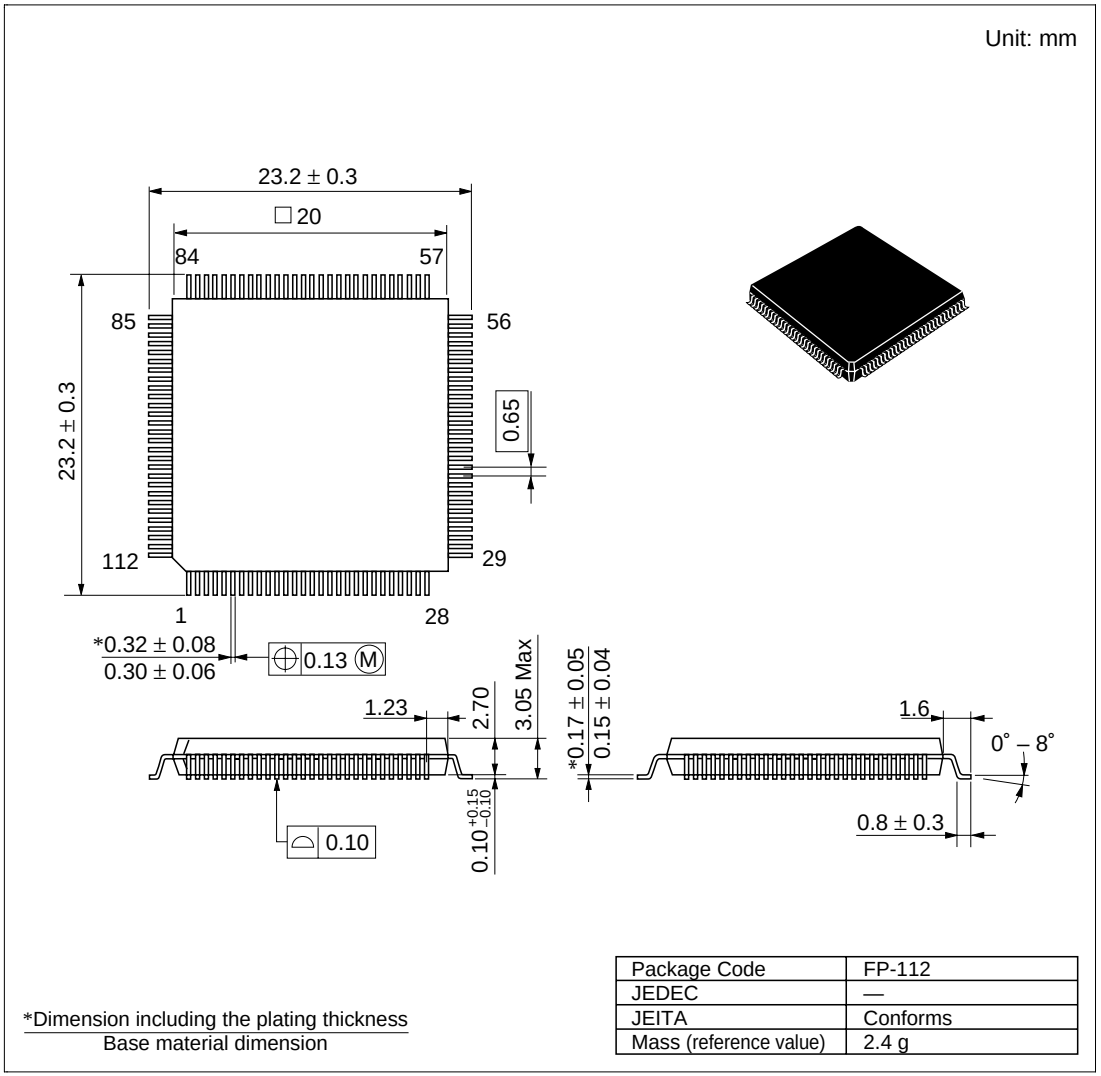
Notes: *1 Package with Copper used as the lead material.

*2 *** in the Order Model No. is the ROM code, consisting of a letter and a two-digit number (ex. E00). The letter indicates the voltage and frequency, as shown below.

- E, F, G, H: 5.0 V, 28 MHz
- P, Q, R: 3.3 V, 16 MHz

Appendix F Package Dimensions

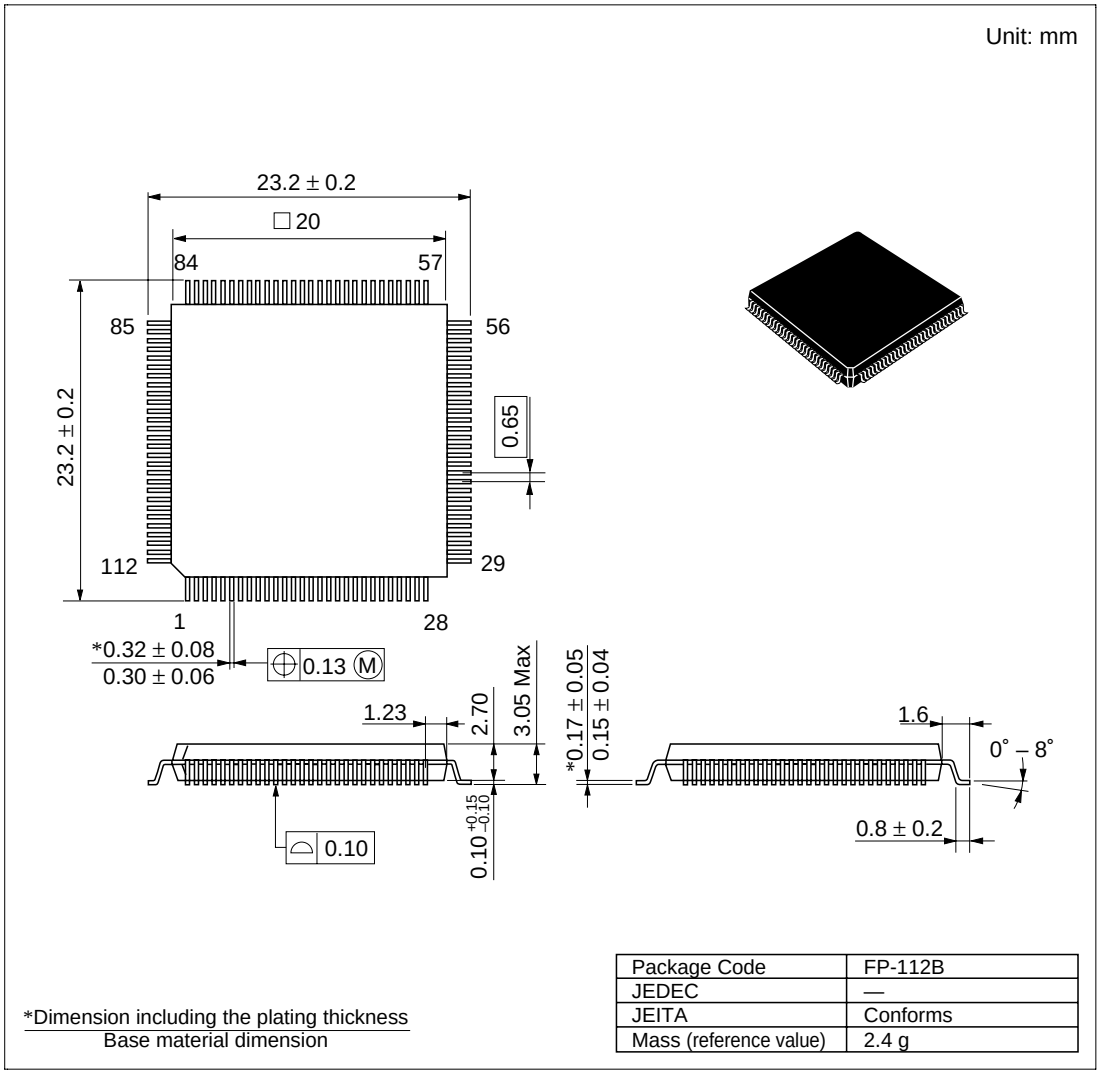
Package dimensions of the SH7040, SH7042, SH7044 (FP-112) are shown in figures F.1 and F.2.
Package dimensions of the SH7040, SH7042 (TFP-120) are shown in figure F.3.
Package dimensions of the SH7041, SH7043, SH7045 (FP-144) are shown in figures F.4 and F.5.



Package Code	FP-112
JEDEC	—
JEITA	Conforms
Mass (reference value)	2.4 g

Figure F.1 Package Dimensions (FP-112)

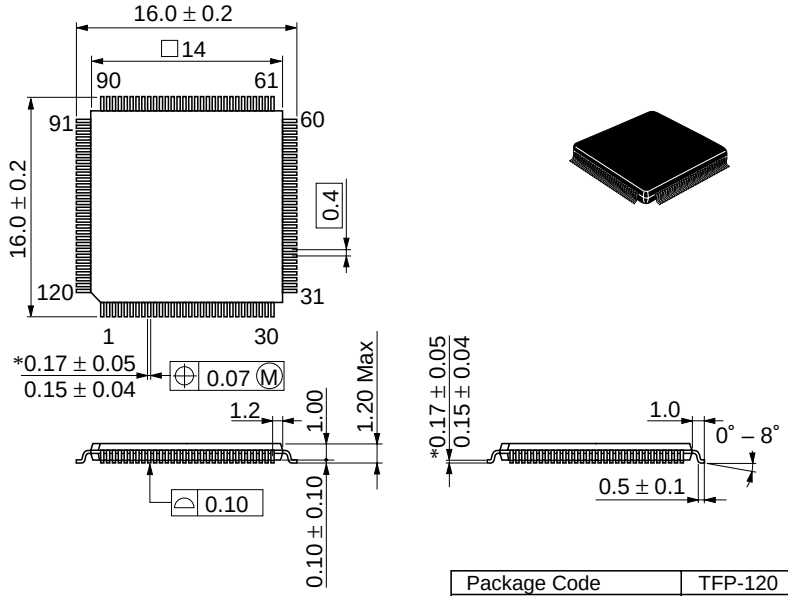
This package (FP-112B) uses copper leads.



Package Code	FP-112B
JEDEC	—
JEITA	Conforms
Mass (reference value)	2.4 g

Figure F.2 Package Dimensions (FP-112B)

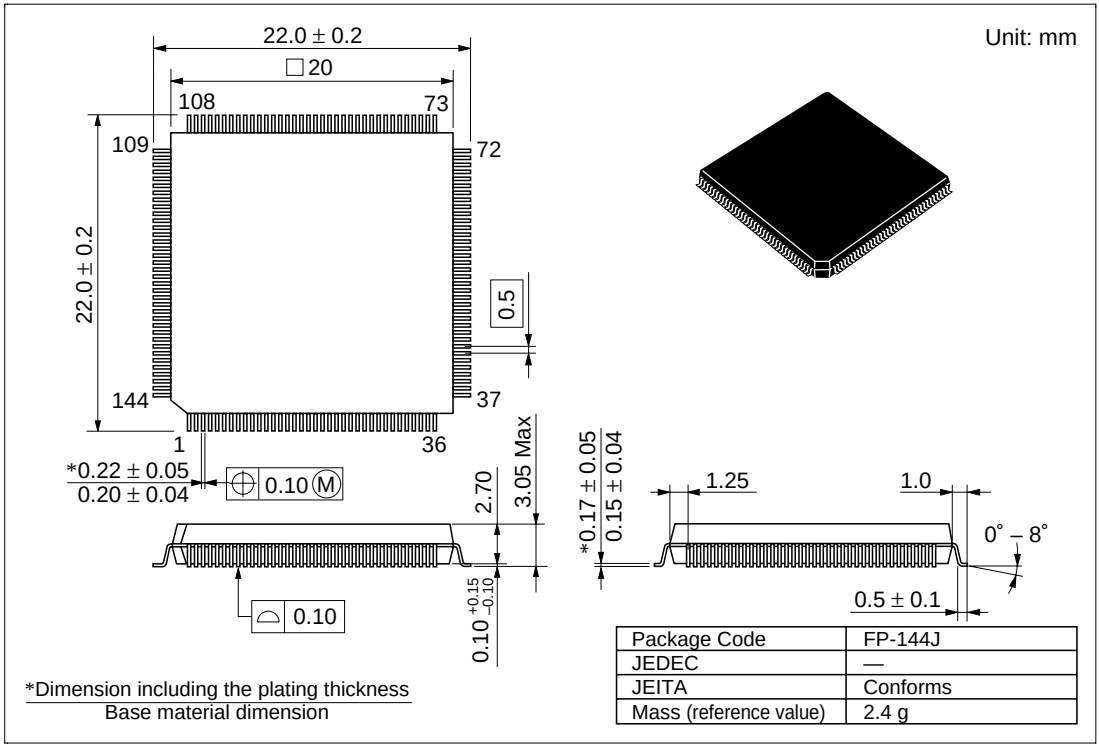
Unit: mm



Package Code	TFP-120
JEDEC	—
JEITA	Conforms
Mass (reference value)	0.5 g

*Dimension including the plating thickness
Base material dimension

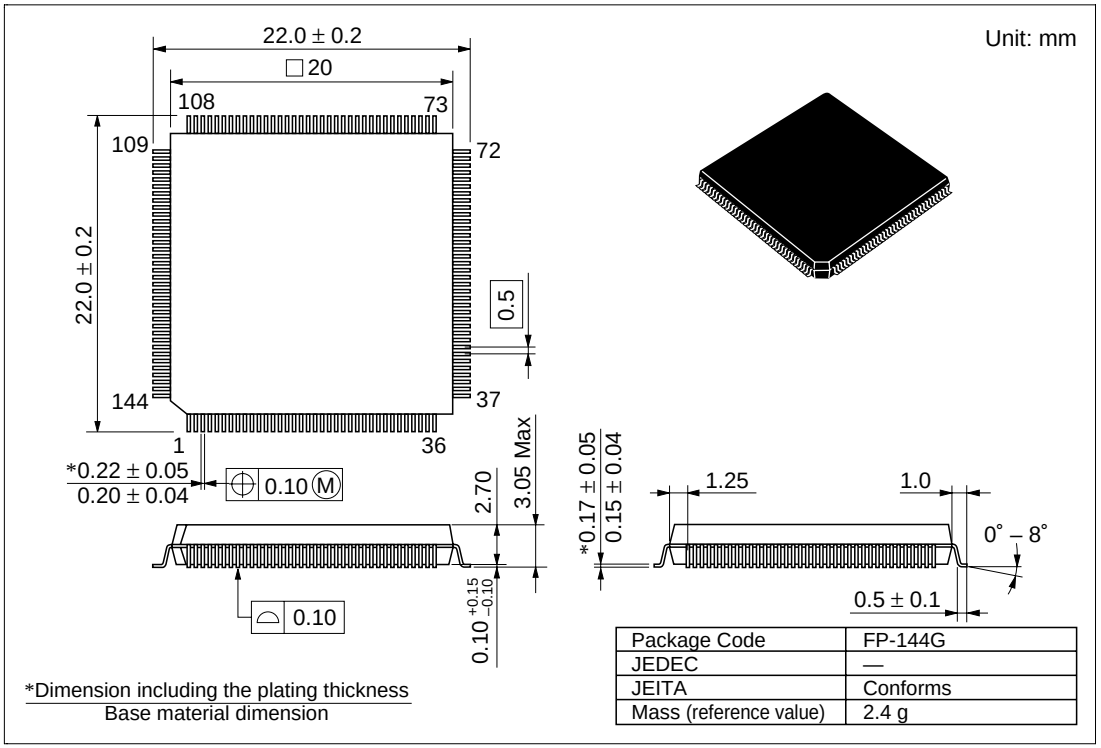
Figure F.3 Package Dimensions (TFP-120)



Package Code	FP-144J
JEDEC	—
JEITA	Conforms
Mass (reference value)	2.4 g

Figure F.4 Package Dimensions (FP-144J)

This package (FP-144G) uses copper leads.



Package Code	FP-144G
JEDEC	—
JEITA	Conforms
Mass (reference value)	2.4 g

Figure F.5 Package Dimensions (FP-144G)

SH7040, SH7041, SH7042, SH7043, SH7044, SH7045 Group Hardware Manual

Publication Date:	1st Edition, February, 1997
	Rev.6.00, May 26, 2003
Published by:	Sales Strategic Planning Div.
	Renesas Technology Corp.
Edited by:	Technical Documentation & Information Department
	Renesas Kodaira Semiconductor Co., Ltd.

Renesas Technology Corp. Sales Strategic Planning Div. Nippon Bldg., 2-6-2, Ohte-machi, Chiyoda-ku, Tokyo 100-0004, Japan

http://www.renesas.com

SH7040, SH7041, SH7042, SH7043, SH7044, SH7045 Group

Hardware Manual

Renesas Electronics Corporation
1753, Shimonumabe, Nakahara-ku, Kawasaki-shi, Kanagawa 211-8668 Japan

REJ09B0044-0600O