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

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, SH704
SH7043, SH7044, SH704
Group

Hardware Manual

REJ09B0

Notes regarding these materials

1. These materials are intended as a reference to assist our customers in the selection of the Renesas Technology C product best suited to the customer's application; they do not convey any license under any intellectual propert 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 originating in the use of any product data, diagrams, charts, programs, algorithms, or circuit application exampl these materials.
3. All information contained in these materials, including product data, diagrams, charts, programs and algorithms information on products at the time of publication of these materials, and are subject to change by Renesas Tec Corporation without notice due to product improvements or other reasons. It is therefore recommended that cus Renesas Technology Corporation or an authorized Renesas Technology Corporation product distributor for the 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 or errors.
   Please also pay attention to information published by Renesas Technology Corporation by various means, incl 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, algorithms, please be sure to evaluate all information as a total system before making a final decision on the app information and products. Renesas Technology Corporation assumes no responsibility for any damage, liabilit resulting from the information contained herein.
5. Renesas Technology Corporation semiconductors are not designed or manufactured for use in a device or syste under circumstances in which human life is potentially at stake. Please contact Renesas Technology Corporatio authorized Renesas Technology Corporation product distributor when considering the use of a product contain specific purposes, such as apparatus or systems for transportation, vehicular, medical, aerospace, nuclear, or und use.
6. The prior written approval of Renesas Technology Corporation is necessary to reprint or reproduce in whole or materials.
7. If these products or technologies are subject to the Japanese export control restrictions, they must be exported u 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 de prohibited.
8. Please contact Renesas Technology Corporation for further details on these materials or the products contained

performance during external memory access.

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

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

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

Related Manual

SH7040 series instructions

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

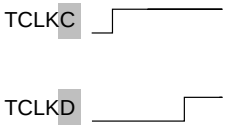
Please consult your Renesas Technology sales representative for details for developmen environment system.

RENESAS

									16 MHz	3.3 V	HD6437040AVCF16
	SH7041A	A mask	64 kB	32 bits	±4LSB (Mid-Speed)	QFP2020-144 QFP2020-144Cu*	–20°C to 75°C	28 MHz 16 MHz	5 V 3.3 V	HD6437041AF28 HD6437041AVF16	See ROM
								28 MHz 16 MHz	5 V 3.3 V	HD6437041ACF28 HD6437041AVCF16	
	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 ROM
	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	5 V 3.3 V 3.3 V	HD6437042AF28 HD6437042AVF16 HD6437042AVX16	See ROM
								28 MHz 16 MHz	5 V 3.3 V	HD6437042ACF28 HD6437042AVCF16	
	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 ROM
	SH7043A	A mask	128 kB	32 bits	±4LSB (Mid-Speed)	QFP2020-144 QFP2020-144Cu*	–20°C to 75°C	28 MHz 16 MHz	5 V 3.3 V	HD6437043AF28 HD6437043AVF16	See ROM
								28 MHz 16 MHz	5 V 3.3 V	HD6437043ACF28 HD6437043AVCF16	
	SH7044	A mask	256 kB	16 bits	±4LSB (Mid-Speed)	QFP2020-112	–20°C to 75°C	28 MHz	5 V	HD6437044F28	See ROM
	SH7045	A mask	256 kB	32bits	±4LSB (Mid-Speed)	QFP2020-144	–20°C to 75°C	28 MHz	5 V	HD6437045F28	See ROM
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	5 V 3.3 V 3.3 V 5 V	HD6417040AF28 HD6417040AVF16 HD6417040AVX16 HD6417040ACF28	
								16 MHz	3.3 V	HD6417040AVCF16	
	SH7041A	A mask		32 bits	±4LSB (Mid-Speed)	QFP2020-144 QFP2020-144Cu*	–20°C to 75°C	28 MHz 16 MHz	5 V 3.3 V	HD6417041AF28 HD6417041AVF16	
								28 MHz 16 MHz	5 V 3.3 V	HD6417041ACF28 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 m proceed while CPU is not programming or erasing the fl 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/
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

Operation Register (DMAOR)		• Bits 15–10—Reserved bits: Data are 0 when read. value always be 0.
	227	Description amended • Bits 7–3—Reserved bits: Data are 0 when read. Th 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**

14.2.8 Bit Rate Register (BRR)

491 Table amended

Table 14.3 Bit Rates and BRR Settings in Asynchronous Mode (cont)

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)

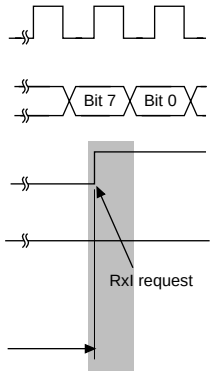
495 Table amended

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

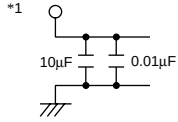
14.3.4 Clock Synchronous Operation

529 Figure amended

Figure 14.22 Example of SCI Receive Operation



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 (o

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

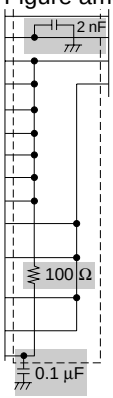
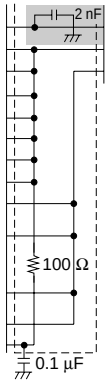
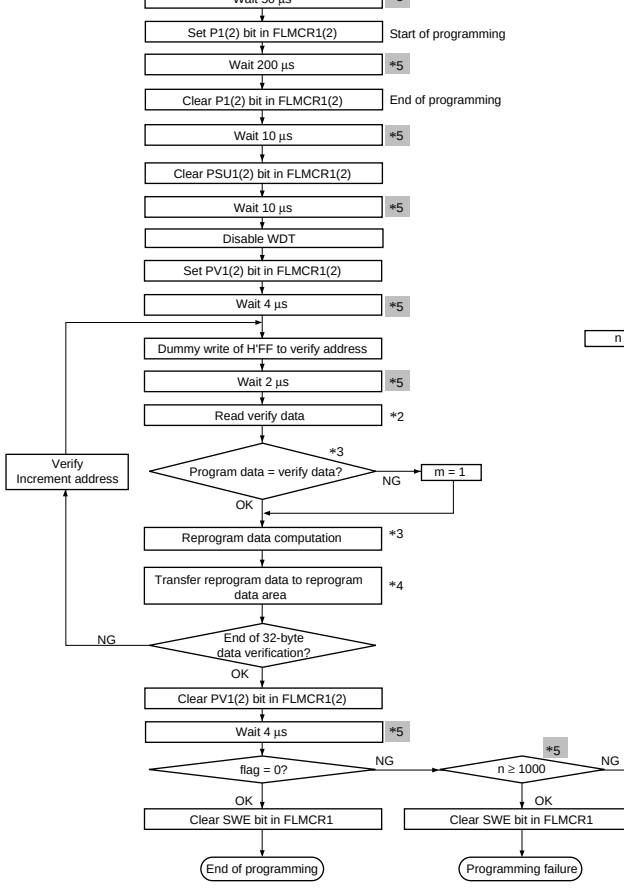
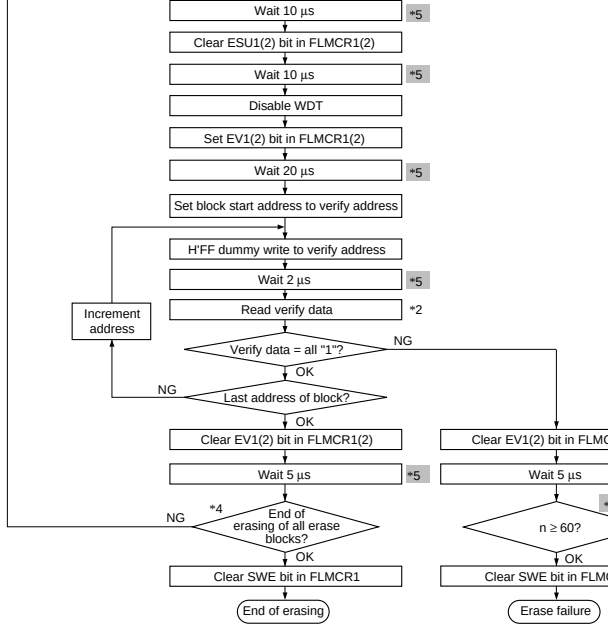


Figure 21.4 SH7043 Pin and HN27C101 Pin Correspondence (144-Pin Version)	673	Figure amended 
22.2.2 Mode Transition Diagram Figure 22.2 Flash Memory Mode Transitions	683	Note amended Execute transition between the user mode and user progr while the CPU is not programming or erasing the flash me



Note *5 added.

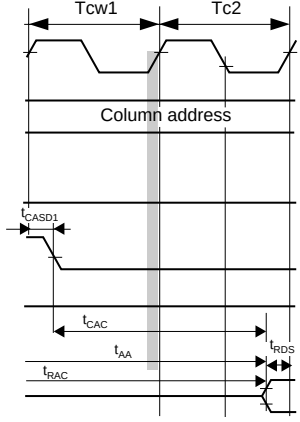
*5 Make sure to set the wait times and repetitions as specified. Programming may not complete correctly if values other than the specified ones are used.



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 tu
*5 Make sure to set the wait times and repetitions as specified. Erasing may not complete correctly i
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

Figure 25.12 DRAM Cycle (Normal Mode, 1 Wait, TPC=0, RCD=0)



764 Figure amended

Figure 25.13 DRAM Cycle (Normal Mode, 2 Waits, TPC=1, RCD=1)

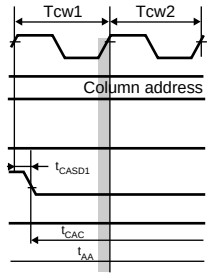
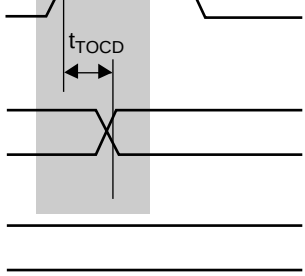
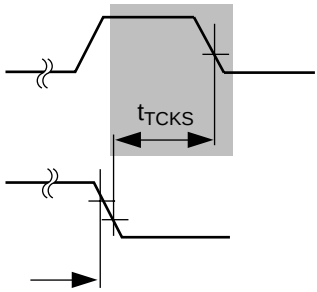


Figure 25.23 MTU I/O Timing		
Figure 25.24 MTU Clock Input Timing	770	Figure amended 
25.3.11 Measuring Conditions for AC Characteristics Figure 25.33 Output Load Circuit	778	Title amended Output Load Circuit

Analog supply current	AI_{CC}	—	4	8	mA	f = 16.7MH
	AI_{ref}	—	0.5	1 *3	mA	QFP144 ve

*3 2 mA in the A mask version of MASK products.

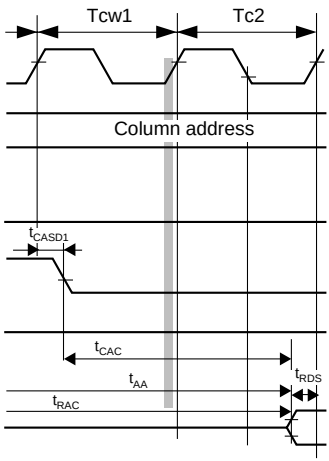
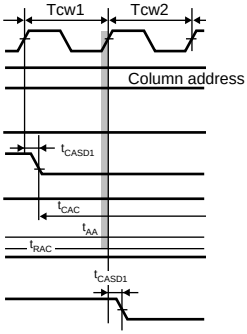
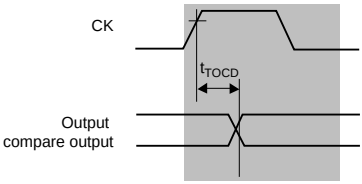
26.3.2 Control Signal Timing Table 26.5 Control Signal Timing	786	Note amended Notes: *1 SH7042/43 ZTAT (excluding A mask) are 3.2V *2 The $\overline{RES}$, $\overline{MRES}$, NMI, $\overline{BREQ}$, and $\overline{IRQ7}$–$\overline{IRQ}$ are asynchronous inputs, but when the setup ti shown here are provided, the signals are consi have produced changes at clock rise (for $\overline{RES}$, $\overline{BREQ}$) or clock fall (for NMI and $\overline{IRQ7}$–$\overline{IRQ0}$). setup times are not provided, recognition is del 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 

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

Modes During Reset, Power-Down, and Bus Right Release Modes (144 Pin)

Class	Pin Name	Power-On	Manual	Standby	Sleep	Release	Ri
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*
	$\overline{IRQOUT}$ (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*
	DACK0, DACK1 (PE14, PE15)	Z*4	O	Z	O	O	Z
	DRAK0, DRAK1	Z*4	O	O*1	O	O	O*
	$\overline{DREQ0}$, $\overline{DREQ1}$	Z*4	I	Z	I	I	Z

	RXD0–RXD1	Z*4	I	Z	I	I	Z
A/D converter	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
	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 st
occur simultaneously due to the timing between $\overline{BREQ}$ and internal oper
cases, standby mode results, but the standby state may be different.

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

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

*1 If the standby control register port high-impedance bits are set to 1, outp
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 m
them, are unstable during the $\overline{RES}$ setup time (t_{RESS}) immediately after
goes to low level.

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[*]
	TIOC3B,TIOC3D, TIOC4A–TIOC4D	Z[*4]	I/O	Z	I/O	I/O	Z
	TCLKA–TCLKD	Z[*4]	I	Z	I	I	Z

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 s
occur simultaneously due to the timing between $\overline{\text{BREQ}}$ and internal oper
cases, standby mode results, but the standby state may be different.
The initial pin states depend on the mode. See section 18, Pin Function
(PFC), for details.

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

*1 If the standby control register port high-impedance bits are set to 1, outp
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 m
them, are unstable during the $\overline{\text{RES}}$ setup time (t_{RESS}) immediately after
goes to low level.

			HD6417041AVCF16	HD6417041AVCF16	QFP2020-144Cu*1	H
SH7042	Mask ROM verion	–	HD6437042F28	HD6437042 (***)F28	QFP2020-112	H
			HD6437042VF16	HD6437042 (***)VF16	QFP2020-112	H
	Z-TAT version	–	HD6477042F28	HD6477042F28	QFP2020-112	H
			HD6477042VF16	HD6477042VF16	QFP2020-112	H
SH7042A	Mask ROM verion	A MASK	HD6437042AF28	HD6437042A(***)F28	QFP2020-112	H
			HD6437042AVF16	HD6437042A(***)VF16	QFP2020-112	H
			HD6437042AVX16	HD6437042A(***)VX16	TQFP1414-120	H
			HD6437042ACF28	HD6437042A(***)CF28	QFP2020-112Cu*1	H
			HD6437042AVCF16	HD6437042A(***)VCF16	QFP2020-112Cu*1	H

Product Type		Mask Version	Product Code	Mark Code	Package	O
SH7042A	Z-TAT version	A MASK	HD6477042AF28	HD6477042AF28	QFP2020-112	H
			HD6477042AVF16	HD6477042AVF16	QFP2020-112	H
			HD6477042AVX16	HD6477042AVX16	TQFP1414-120	H
			HD6477042ACF28	HD6477042ACF28	QFP2020-112Cu*1	H
			HD6477042AVCF16	HD6477042AVCF16	QFP2020-112Cu*1	H
SH7043	Mask ROM version	–	HD6437043F28	HD6437043(***)F28	QFP2020-144	H
			HD6437043VF16	HD6437043(***)VF16	QFP2020-144	H
	Z-TAT version	–	HD6477043F28	HD6477043F28	QFP2020-144	H
			HD6477043VF16	HD6477043VF16	QFP2020-144	H
SH7043A	Mask ROM version	A MASK	HD6437043AF28	HD6437043A(***)F28	QFP2020-144	H
			HD6437043AVF16	HD6437043A(***)VF16	QFP2020-144	H
			HD6437043ACF28	HD6437043A(***)CF28	QFP2020-144Cu*1	H
			HD6437043AVCF16	HD6437043A(***)VCF16	QFP2020-144Cu*1	H
	Z-TAT version	A MASK	HD6477043AF28	HD6477043AF28	QFP2020-144	H
			HD6477043AVF16	HD6477043AVF16	QFP2020-144	H
			HD6477043ACF28	HD6477043ACF28	QFP2020-144Cu*1	H
			HD6477043AVCF16	HD6477043AVCF16	QFP2020-144Cu*1	H
SH7044	Mask ROM version	A MASK	HD6437044F28	HD6437044(***)F28	QFP2020-112	H
	F-ZTAT version		HD64F7044F28	HD64F7044F28	QFP2020-112	H
SH7045	Mask ROM version	A MASK	HD6437045F28	HD6437045(***)F28	QFP2020-144	H
	F-ZTAT version		HD64F7045F28	HD64F7045F28	QFP2020-144	H

(***) 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 t
number (ex. E00). The letter indicates the voltage and frequency, as sho

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

RENESAS

impossible with microprocessors, such as real-time control, which demands high speeds particular, the SH7040 series has a 1-kbyte on-chip cache, which allows an improvemen performance during external memory access.

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

In addition to the masked-ROM versions of the SH7040 series, the SH7042 and SH7043 ZTAT™*1 version with user-programmable on-chip PROM and the SH7044 and SH704 F-ZTAT™*2 version with on-chip flash memory. These versions enable users to respond and flexibly to changing application specifications, growing production volumes, and ot conditions.

Notes: *1 ZTAT (Zero Turn-Around Time) is a registered trademark of Renesas Techn 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

- 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/ 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 space) with the following settable features:
  - Bus size (8, 16, or 32 bits)
  - Number of wait cycles

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 transf destination address
  - — Indirect transfer mode: regards the data at the transfer source address as an addre 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 reque
- 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 transf
- 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 processin based on 16-bit timer, 5 channels
- 16 dual-use output compare/input capture registers

- 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

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 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):

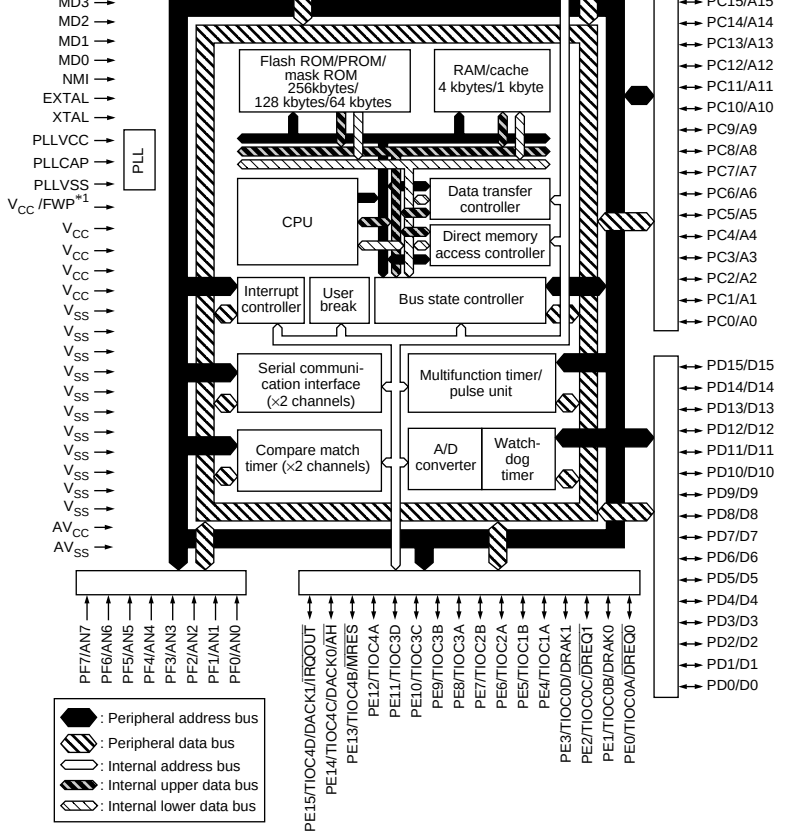
- On-chip clock pulse generator
  - On-chip clock-doubling PLL circuit

Notes on the SH7040 Series Specifications (For details, see each section in this manual)

	DTC	DMAC	MTU	A/D Converter	ROM	Electrical Characteristics
				See “High-Speed A/D Converter”	See “128 kB PROM”	See “Electrical Characteristics”
ot	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”
				See “High-Speed A/D Converter”	See “128 kB PROM”	See “Electrical Characteristics”
ot	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”
ot	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”
ot	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”
ot	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”
ot	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”
				See “High-Speed A/D Converter”	See “128 kB Mask ROM”	See “Electrical Characteristics”
ot	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”
				See “High-Speed A/D Converter”	See “128 kB Mask ROM”	See “Electrical Characteristics”
ot	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”

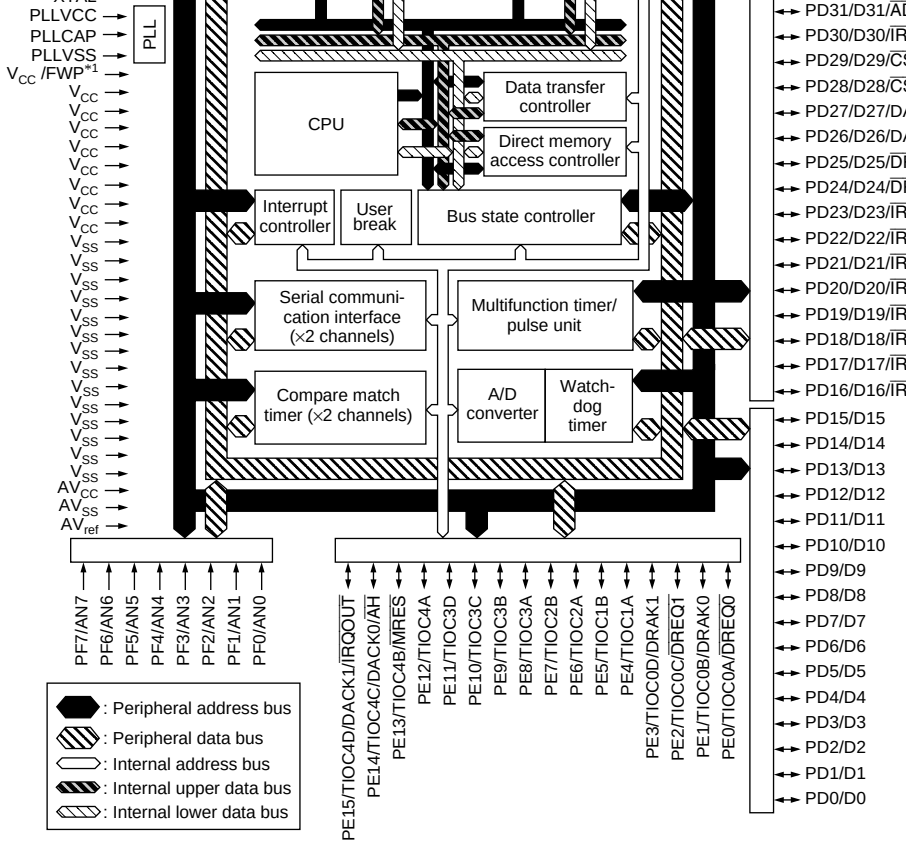
Notes on the SH7040 Series Specifications (For details, see each section in this manual)

	DTC	DMAC	MTU	A/D Converter	ROM	Electrical Characteristics
ot	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”
ot	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”
ot	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”
ot	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”



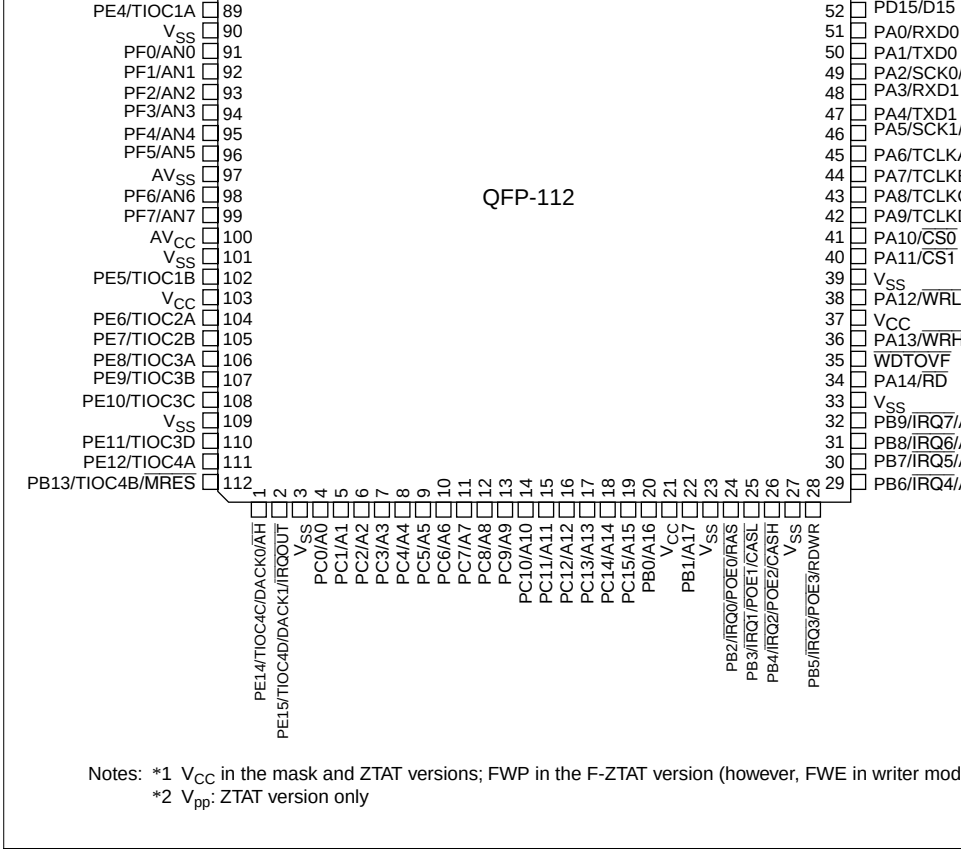
Notes: *1 V_{CC} in the mask and ZTAT versions; FWP in the F-ZTAT version (however, FWE in writer mode)
*2 V_{pp}: ZTAT version only

Figure 1.1 Block Diagram of the SH7040, SH7042, SH7044 (QFP-112 Pin), SH704 (TQFP-120 pin)



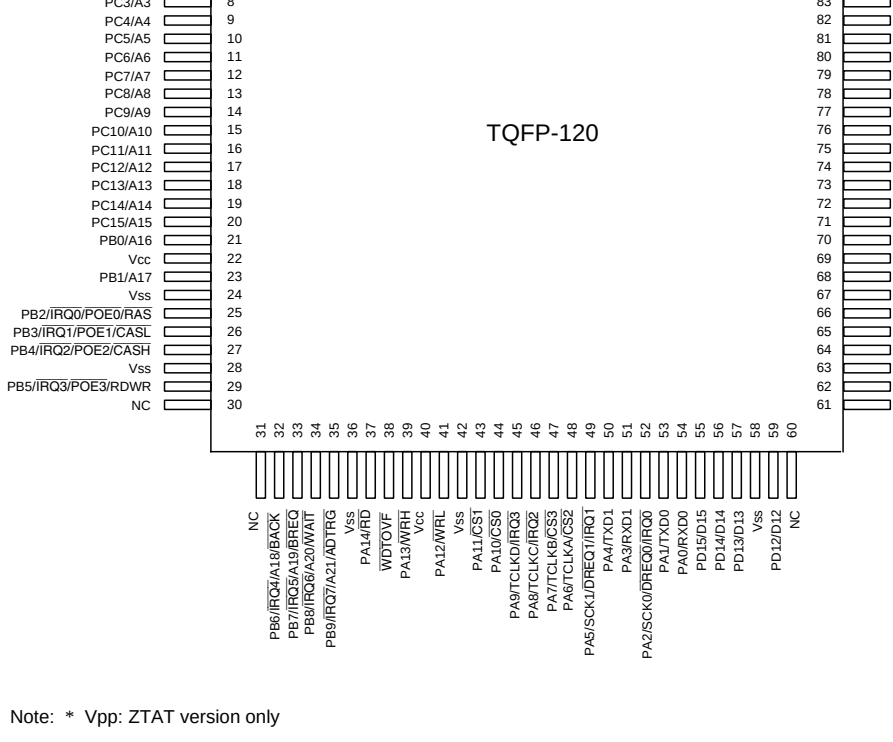
Notes: *1 V_{CC} in the mask and ZTAT versions; FWP in the F-ZTAT version (however, FWE in writer mode)
*2 V_{pp}: ZTAT version only

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



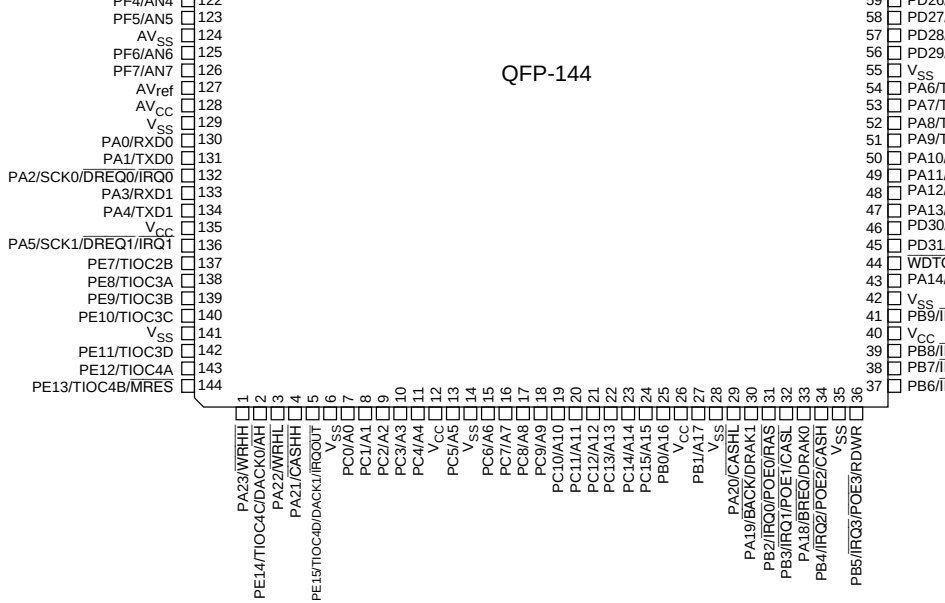
Notes: *1 V_{CC} in the mask and ZTAT versions; FWP in the F-ZTAT version (however, FWE in writer mod
*2 V_{pp}: ZTAT version only

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



Note: * Vpp: ZTAT version only

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



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

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

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/$\overline{\text{IRQ0}}$/$\overline{\text{POE0}}$/$\overline{\text{RAS}}$	NC
25	PB3/$\overline{\text{IRQ1}}$/$\overline{\text{POE1}}$/$\overline{\text{CASL}}$	$\overline{\text{OE}}$
26	PB4/$\overline{\text{IRQ2}}$/$\overline{\text{POE2}}$/$\overline{\text{CASH}}$	$\overline{\text{PGM}}$
27	V_{SS}	V_{SS}
28	PB5/$\overline{\text{IRQ3}}$/$\overline{\text{POE3}}$/RDWR	V_{CC}

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/$\overline{IRQ3}$	NC
43	PA8/TCLKC/$\overline{IRQ2}$	NC
44	PA7/TCLKB/$\overline{CS3}$	NC
45	PA6/TCLKA/$\overline{CS2}$	NC
46	PA5/SCK1/$\overline{DREQ1}$/$\overline{IRQI}$	NC
47	PA4/TXD1	NC
48	PA3 /RXD1	NC
49	PA2/SCK0/$\overline{DREQ0}$/$\overline{IRQ0}$	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

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

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

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/$\overline{\text{IRQ0}}$/$\overline{\text{POE0}}$/$\overline{\text{RAS}}$	NC
26	PB3/$\overline{\text{IRQ1}}$/$\overline{\text{POE1}}$/$\overline{\text{CASL}}$	OE
27	PB4/$\overline{\text{IRQ2}}$/$\overline{\text{POE2}}$/$\overline{\text{CASH}}$	PGM
28	V_{SS}	V_{SS}
29	PB5/$\overline{\text{IRQ3}}$/$\overline{\text{POE3}}$/RDWR	V_{CC}
30	NC	NC
31	NC	NC

40	V_{CC}	V_{CC}
41	PA12/$\overline{WRL}$	NC
42	V_{SS}	V_{SS}
43	PA11/$\overline{CS1}$	NC
44	PA10/$\overline{CS0}$	NC
45	PA9/TCLKD/$\overline{IRQ3}$	NC
46	PA8/TCLKC/$\overline{IRQ2}$	NC
47	PA7/TCLKB/$\overline{CS3}$	NC
48	PA6/TCLKA/$\overline{CS2}$	NC
49	PA5/SCK1/$\overline{DREQ1}$/$\overline{IRQ1}$	NC
50	PA4/TXD1	NC
51	PA3/RXD2	NC
52	PA2/SCK0/$\overline{DREQ0}$/$\overline{IRQ0}$	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

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

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

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

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

69	PD18/D18/$\overline{IRQ2}$	NC
70	PD17/D17/$\overline{IRQ1}$	NC
71	V_{SS}	V_{SS}
72	PD16/D16/$\overline{IRQ0}$	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

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}

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

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/$\overline{\text{IRQ0}}$/$\overline{\text{POE0}}$/$\overline{\text{RAS}}$	NC
25	PB3/$\overline{\text{IRQ1}}$/$\overline{\text{POE1}}$/$\overline{\text{CASL}}$	NC
26	PB4/$\overline{\text{IRQ2}}$/$\overline{\text{POE2}}$/$\overline{\text{CASH}}$	A17
27	V_{SS}	V_{SS}
28	PB5/$\overline{\text{IRQ3}}$/$\overline{\text{POE3}}$/RDWR	NC
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

42	PA9/TCLKD/$\overline{\text{IRQ3}}$	$\overline{\text{CE}}$
43	PA8/TCLKC/$\overline{\text{IRQ2}}$	$\overline{\text{OE}}$
44	PA7/TCLKB/$\overline{\text{CS3}}$	$\overline{\text{WE}}$
45	PA6/TCLKA/$\overline{\text{CS2}}$	NC
46	PA5/SCK1/$\overline{\text{DREQ1}}$/$\overline{\text{IRQ1}}$	V_{CC}
47	PA4/TXD1	NC
48	PA3/RXD1	NC
49	PA2/SCK0/$\overline{\text{DREQ0}}$/$\overline{\text{IRQ0}}$	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

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

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

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

46	PD30/D30/IRQOUT	NC
47	PA13/$\overline{WRH}$	NC
48	PA12/$\overline{WRL}$	NC
49	PA11/$\overline{CS1}$	NC
50	PA10/$\overline{CS0}$	NC
51	PA9/TCLKD/$\overline{IRQ3}$	$\overline{CE}$
52	PA8/TCLKC/$\overline{IRQ2}$	$\overline{OE}$
53	PA7/TCLKB/$\overline{CS3}$	$\overline{WE}$
54	PA6/TCLKA/$\overline{CS2}$	NC
55	V_{SS}	V_{SS}
56	PD29/D29/$\overline{CS3}$	NC
57	PD28/D28/$\overline{CS2}$	NC
58	PD27/D27/DACK1	NC
59	PD26/D26/DACK0	NC
60	PD25/D25/$\overline{DREQ1}$	NC
61	V_{SS}	V_{SS}
62	PD24/D24/$\overline{DREQ0}$	NC
63	V_{CC}	V_{CC}
64	PD23/D23/$\overline{IRQ7}$	NC
65	PD22/D22/$\overline{IRQ6}$	NC
66	PD21/D21/$\overline{IRQ5}$	NC
67	PD20/D20/$\overline{IRQ4}$	NC
68	PD19/D19/$\overline{IRQ3}$	NC
69	PD18/D18/$\overline{IRQ2}$	NC
70	PD17/D17/$\overline{IRQ1}$	NC
71	V_{SS}	V_{SS}
72	PD16/D16/$\overline{IRQ0}$	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 write

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

				ground. No operation will oc there are any open pins.
	V_{PP}	I	Program supply	Connects to the power supp during normal operation. When in PROM mode, apply
Clock	PLLVCC	I	PLL supply	On-chip PLL oscillator suppl
	PLLVSS	I	PLL ground	On-chip PLL oscillator groun
	PLLCAP	I	PLL capacitance	On-chip PLL oscillator extern capacitance connection pin.
	EXTAL	I	External clock	Connect a crystal oscillator. external clock can be input t 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
	$\overline{\text{BREQ}}$	I	Bus request	Goes low when external dev requests bus right release
	$\overline{\text{BACK}}$	O	Bus request acknowledge	Indicates that bus right has b released to external device. device that output the $\overline{\text{BREQ}}$ receives the $\overline{\text{BACK}}$ signal, n the device that it has obtaine right.

				edge input.
	$\overline{\text{IRQOUT}}$	O	Interrupt request output	Indicates that interrupt caus occurred. Enables notificati interrupt generation also du 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 TQ pin versions) or 32-bit (QFP version) bidirectional data b
Bus control	$\overline{\text{CS0}}$–$\overline{\text{CS3}}$	O	Chip selects 0–3	Chip select signals for exter memory or devices.
	$\overline{\text{RD}}$	O	Read	Indicates reading from an e device.
	$\overline{\text{WRH}}$	O	Upper write	Indicates writing the upper 8 (15–8) of external data.
	$\overline{\text{WRL}}$	O	Lower write	Indicates writing the lower 8 (7–0) of external data.
	$\overline{\text{WAIT}}$	I	Wait	Input causes insertion of wa into the bus cycle during ex 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 col address strobe. Output when the upper 8 bit data are accessed.

	(QFP-144)			24 of external data.
	$\overline{\text{WRHL}}$ (QFP-144)	O	HL write	Indicates the writing of bi 16 of external data.
	$\overline{\text{CASHH}}$ (QFP-144)	O	HH column address strobe	Timing signal for DRAM c address strobe. Output w 31 to 24 of data are acce
	$\overline{\text{CASHL}}$ (QFP-144)	O	HL column address strobe	Timing signal for DRAM c address strobe. Output w 23 to 16 of data are acce
Bus control multifunction timer/pulse unit	TCLKA TCLKB TCLKC TCLKD	I	MTU timer clock input	Input pins for external clo the MTU counter.
	TIOC0A TIOC0B TIOC0C TIOC0D	I/O	MTU input capture/ output compare (channel 0)	Channel 0 input capture input/output compare out output pins.
	TIOC1A TIOC1B	I/O	MTU input capture/output compare (channel 1)	Channel 1 input capture input/output compare out output pins.
	TIOC2A TIOC2B	I/O	MTU input capture/output compare (channel 2)	Channel 2 input capture input/output compare out output pins.

Direct memory access controller (DMAC)	$\overline{\text{DREQ0}}$	I	DMA transfer request (channels 0, 1)	input pin for external request DMA transfer.
	$\overline{\text{DREQ1}}$			
	DRAK0–DRAK1	O	DREQ request acknowledgment (channels 0, 1)	Output the input sampling acknowledgment of external transfer requests.
	DACK0–DACK1	O	DMA transfer strobe (channels 0, 1)	Output a strobe to the externa external DMA transfer reques
Serial communication interface (SCI)	TxD0–TxD1	O	Transmit data (channels 0, 1)	SCI0, SCI1 transmit data outp (TxD1 is used for data transfe boot mode of F-ZTAT)
	RxD0–RxD1	I	Receive data (channels 0, 1)	SCI0, SCI1 receive data inpu (RxD1 is used for data transfe boot mode of F-ZTAT)
	SCK0–SCK1	I/O	Serial clock (channels 0, 1)	SCI0, SCI1 clock input/output
A/D Converter	AV_{CC}	I	Analog supply	Analog supply; connected to
	AV_{SS}	I	Analog ground	Analog supply; connected to
	AVref (QFP-144 only)	I	Analog reference supply	Analog reference supply inpu (Connected to AV_{CC} internally 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.

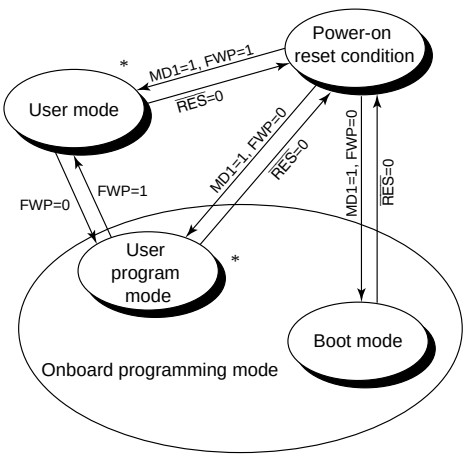
		port	pins. Each bit can be designated input/output.
PC0–PC15	I/O	General purpose port	General purpose input/outp pins. Each bit can be designated input/output.
PD0–PD15 (QFP-112) PD0–PD31 (QFP-144)	I/O	General purpose port	General purpose input/outp pins. Each bit can be designated input/output.
PE0–PE15	I/O	General purpose port	General purpose input/outp pins. Each bit can be designated input/output.
PF0–PF7	I	General purpose port	General purpose input por

Usage Notes

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

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



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

Figure 1.6 Condition Transfer for Flash Memory

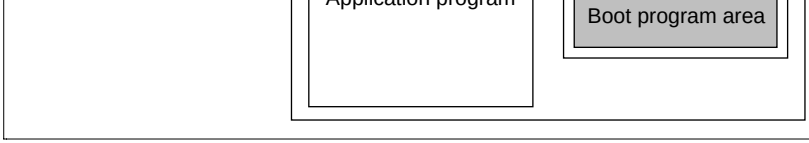
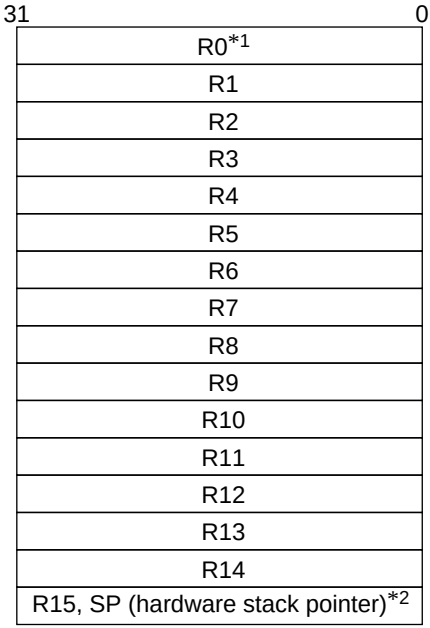


Figure. 1.7 Data Transfer during Boot Mode

and recovering the status register (SR) and program counter (PC) in exception processin
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 addres
mode and indirect indexed GBR addressing mode. In some instructions
functions as a fixed source register or destination register.
*2 R15 functions as a hardware stack pointer (SP) during exception proces

Figure 2.1 General Registers

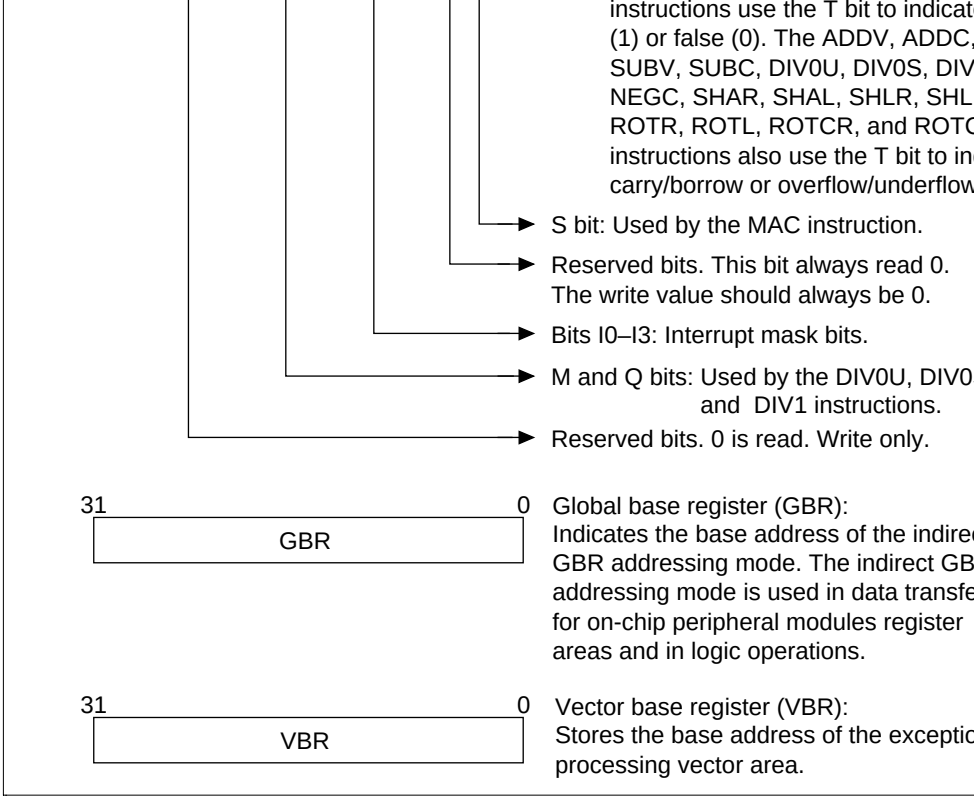
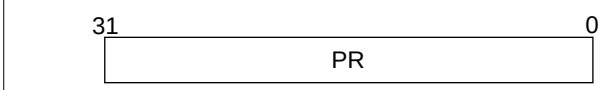
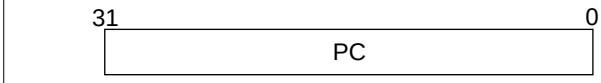


Figure 2.2 Control Registers



Procedure register (PR): Stores a return address from a subroutine procedure.



Program counter (PC): Indicates the fourth byte (second instructi after the current instruction.

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 addres
Control registers	SR	Bits I3–I0 are 1111 (H'F), reserved bits are 0, bits are undefined
	GBR	Undefined
	VBR	H'00000000
System registers	MACH, MACL, PR	Undefined
	PC	Value of the program counter in the vector ad

2.2.2 Data Format in Memory

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

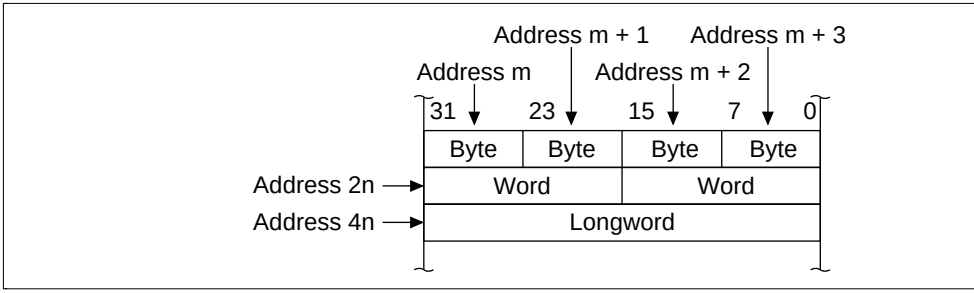


Figure 2.5 Byte, Word, and Longword Alignment

2.2.3 Immediate Data Format

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

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 a
bytes, words, or longwords. Byte or word data accessed from memory is sign-extended
handled as longword data. Immediate data is sign-extended for arithmetic operations or
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 Conventiona	
`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 operati
involve memory access, data is loaded to the registers and executed (load-store architect
Instructions such as AND that manipulate bits, however, are executed directly in memor

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

instructions that change the T bit is kept to a minimum to improve the processing speed (t
2.4).

Table 2.4 T Bit

SH7040 Series CPU		Description	Example of Convention	
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 longw
immediate data is not input via instruction codes but is stored in a memory table. An imm
data transfer instruction (MOV) accesses the memory table using the PC relative addressi
with displacement (table 2.5).

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

Table 2.6 Absolute Address Accessing

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

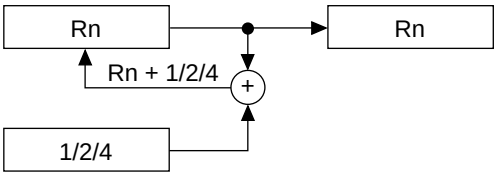
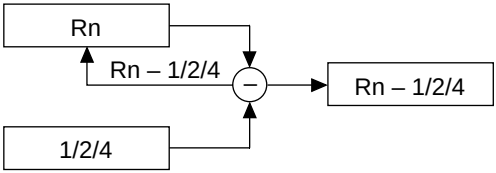
Note: @(disp,PC) accesses the immediate data.

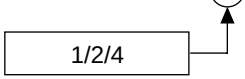
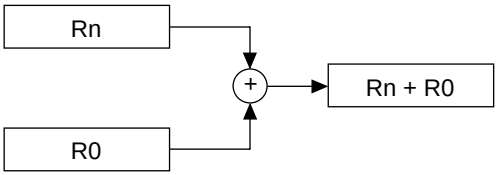
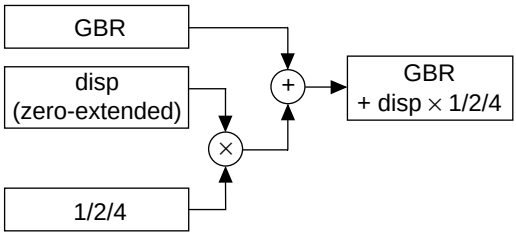
16-Bit/32-Bit Displacement: When data is accessed by 16-bit or 32-bit displacement, th existing displacement value is placed in the memory table. Loading the immediate data instruction is executed transfers that value to the register and the data is accessed in the indexed register addressing mode (table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

Table 2.7 Displacement Accessing

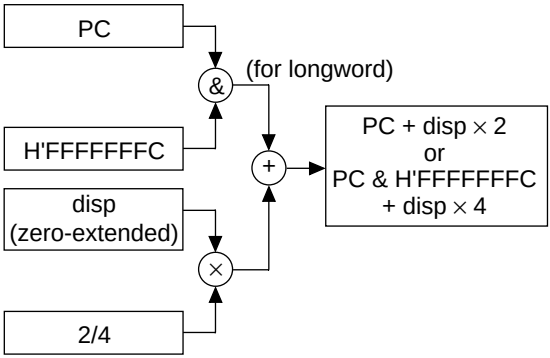
Classification	SH7040 Series CPU		Example of Conventional
16-bit displacement	MOV.W	@(disp,PC),R0	MOV.W @(H'1234,R1),R2
	MOV.W	@(R0,R1),R2	
		.....	
	.DATA.W	H'1234	

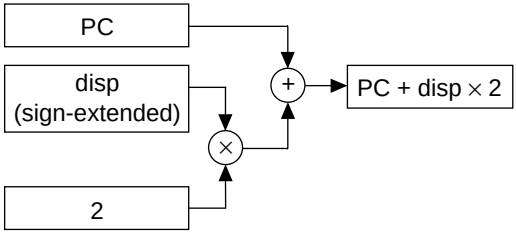
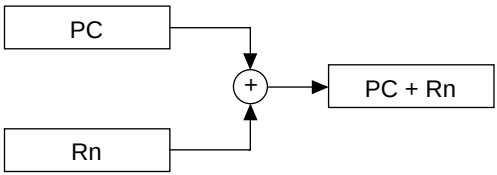
Note: @(disp,PC) accesses the immediate data.

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 (Afte instr exec Byte → R Wor → R Long 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 → R Wor → R Long Rn – (Inst exec Rn a calc

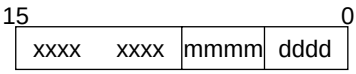
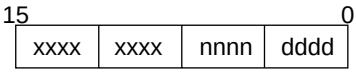
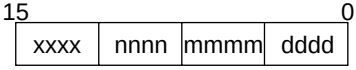
			
Indirect indexed register addressing	@(R0, Rn)	The effective address is the Rn value plus R0. 	Rn
Indirect GBR addressing with displacement	@(disp:8, GBR)	The effective address is the GBR value plus an 8-bit displacement (disp). The value of disp is zero-extended, and remains the same for a byte operation, is doubled for a word operation, and is quadrupled for a longword operation. 	Byte disp Wo disp Lon GBI 4

quadrupled for a longword operation. For a longword operation, the lowest two bits of the PC value are masked.

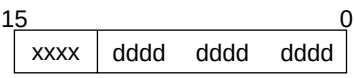
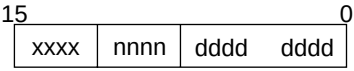
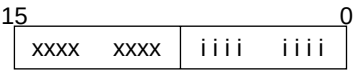
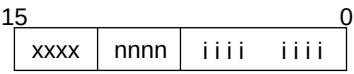


	disp:12	The effective address is the PC value sign-extended with a 12-bit displacement (disp), doubled, and added to the PC value.  	PC
	Rn	The effective address is the register PC value plus Rn.  	PC
Immediate addressing	#imm:8	The 8-bit immediate data (imm) for the TST, AND, OR, and XOR instructions 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.	—

Instruction Formats	Source Operand	Destination Operand	Example
0 format 15 0 xxxx xxxx xxxx xxxx	—	—	NOP
n format 15 0 xxxx nnnn xxxx xxxx	—	nnnn: Direct register	MOVT R
	Control register or system register	nnnn: Direct register	STS M
	Control register or system register	nnnn: Indirect pre-decrement register	STC.L
m format 15 0 xxxx mmmm xxxx xxxx	mmmm: Direct register	Control register or system register	LDC
	mmmm: Indirect post-increment register	Control register or system register	LDC.L
	mmmm: Direct register	—	JMP @R
	mmmm: PC relative using Rm	—	BRAF R

	post-increment register (multiply/ accumulate)		
	mmmm: Indirect post-increment register	nnnn: Direct register	MOV.L
	mmmm: Direct register	nnnn: Indirect pre-decrement register	MOV.L
	mmmm: Direct register	nnnn: Indirect indexed register	MOV.L Rm,@(R
md format 	mmmmdddd: indirect register with displacement	R0 (Direct register)	MOV.B @(disp
nd4 format 	R0 (Direct register)	nnnndddd: Indirect register with displacement	MOV.B R0,@(d
nmd format 	mmmm: Direct register	nnnndddd: Indirect register with displacement	MOV.L Rm,@(d
	mmmmdddd: Indirect register with displacement	nnnn: Direct register	MOV.L @(disp

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

	dddddddd: PC relative	—	BF
d12 format 	dddddddddddd: PC relative	—	BRA (label PC)
nd8 format 	dddddddd: PC relative with displacement	nnnn: Direct register	MOV.L @(disp,
i format 	iiiiiiii: Immediate	Indirect indexed GBR	AND.B #imm,@(
	iiiiiiii: Immediate	R0 (Direct register)	AND
	iiiiiiii: Immediate	—	TRAPA
ni format 	iiiiiiii: Immediate	nnnn: Direct register	ADD

		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	

		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	

		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, op
and execution states in order by classification.

		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-d in the column means no change.

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

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

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

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

CMP/EQ	Rm,Rn	0011nnnnmmmm0000	If Rn = Rm, 1 → T	1	Co res
CMP/HS	Rm,Rn	0011nnnnmmmm0010	If Rn≥Rm with unsigned data, 1 → T	1	Co res
CMP/GE	Rm,Rn	0011nnnnmmmm0011	If Rn ≥ Rm with signed data, 1 → T	1	Co res
CMP/HI	Rm,Rn	0011nnnnmmmm0110	If Rn > Rm with unsigned data, 1 → T	1	Co res
CMP/GT	Rm,Rn	0011nnnnmmmm0111	If Rn > Rm with signed data, 1 → T	1	Co res
CMP/PL	Rn	0100nnnn00010101	If Rn > 0, 1 → T	1	Co res
CMP/PZ	Rn	0100nnnn00010001	If Rn ≥ 0, 1 → T	1	Co res
CMP/STR	Rm,Rn	0010nnnnmmmm1100	If Rn and Rm have an equivalent byte, 1 → T	1	Co res
DIV1	Rm,Rn	0011nnnnmmmm0100	Single-step division (Rn/Rm)	1	Ca res
DIV0S	Rm,Rn	0010nnnnmmmm0111	MSB of Rn → Q, MSB of Rm → M, M ^ Q → T	1	Ca res
DIV0U		0000000000011001	0 → M/Q/T	1	0

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	Borr

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
TST	Rm,Rn	0010nnnnmmmm1000	Rn & Rm; if the result is 0, 1 → T	1
TST	#imm,R0	11001000iiiiiiii	R0 & imm; if the result is 0, 1 → T	1
TST.B	#imm,@(R0,GBR)	11001100iiiiiiii	(R0 + GBR) & imm; if the result is 0, 1 → T	3
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 TAS instruction execution. However, bus release due to $\overline{\text{BREQ}}$ is carried out.

SHLR	Rn	0100nnnn00000001	0 → Rn → T	1
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

			PC	
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.

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
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

destination register of the load instruction (memory → register) and the register u
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 exe
and power-down. Figure 2.6 shows the transitions between the states.

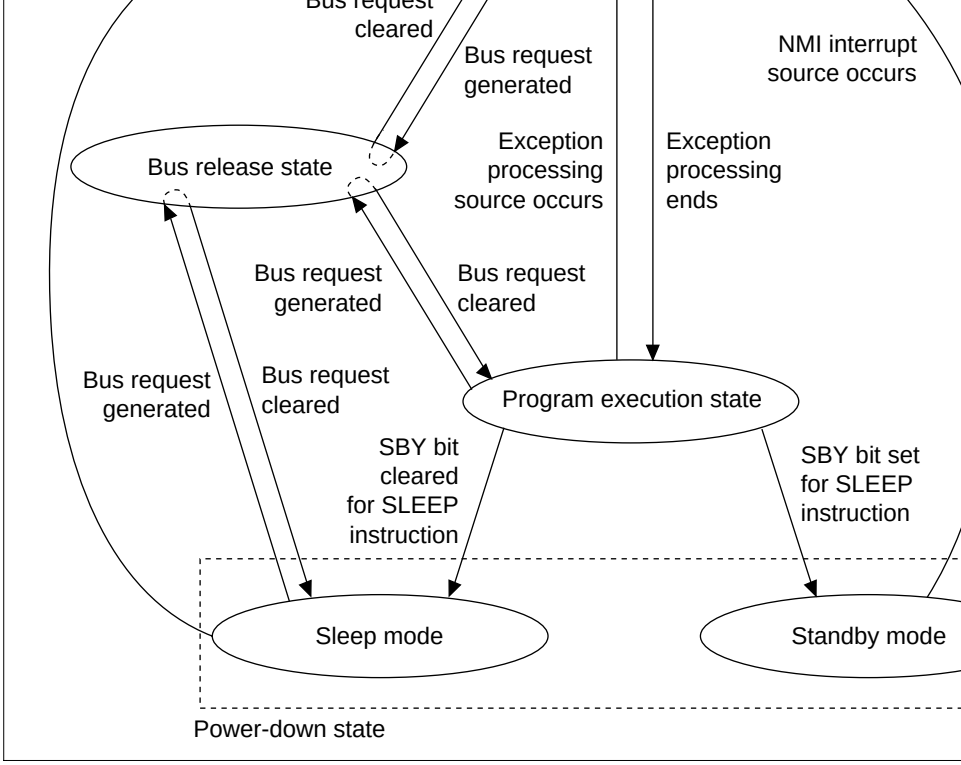


Figure 2.6 Transitions between Processing States

Reset State: The CPU resets in the reset state. When the $\overline{\text{RES}}$ pin level goes low, a pow
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 occ
exception processing sources such as resets or interrupts alter the CPU's processing stat

declines. The SLEEP instruction places the CPU in the power-down state. This state has t
modes: sleep mode and standby mode.

Bus Release State: In the bus release state, the CPU releases access rights to the bus to th
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 w
CPU operation halts, lowering power consumption. There are two power-down state mod
mode and standby mode.

Sleep Mode: When standby bit SBY (in the standby control register SBYCR) is cleared t
SLEEP instruction executed, the CPU moves from program execution state to sleep mode
sleep mode, the CPU halts and the contents of its internal registers and the data in on-chip
(or on-chip RAM) is maintained. The on-chip peripheral modules other than the CPU do
in the sleep mode.

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

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

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

	with SBY bit cleared to 0 in SBYCR							error • Powe • Manu
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 • Powe • Manu

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

Mode No.	Pin Setting					Mode Name	On-Chip ROM	CS0 Area
	FWP	MD3*1	MD2*1	MD1	MD0			112 Pin
0	1	x	x	0	0	MCU mode 0	Not Active	8-bit space
1	1	x	x	0	1	MCU mode 1	Not Active	16-bit space
2	1	x	x	1	0	MCU mode 2	Active	8/16-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
—	0	x	x	0	1			—
—	0	x	x	1	0	User programming mode*4	Active	8/16-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 3.3 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 wi 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 w 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-c space is 32 bit.
Mode 3 (single chip mode)	Any port can be used, but external addresses can not be emplo
Mode 4 (PROM mode)	On-chip ROM can be programmed using a general PROM write
Clock mode	The input waveform frequency can be used as is, doubled or qu as an internal clock in modes 0 to 3.

MD1	Input	Designates operating mode through the level applied to th
MD2	Input	Designates clock mode through the level applied to this p
MD3	Input	Designates clock mode through the level applied to this p

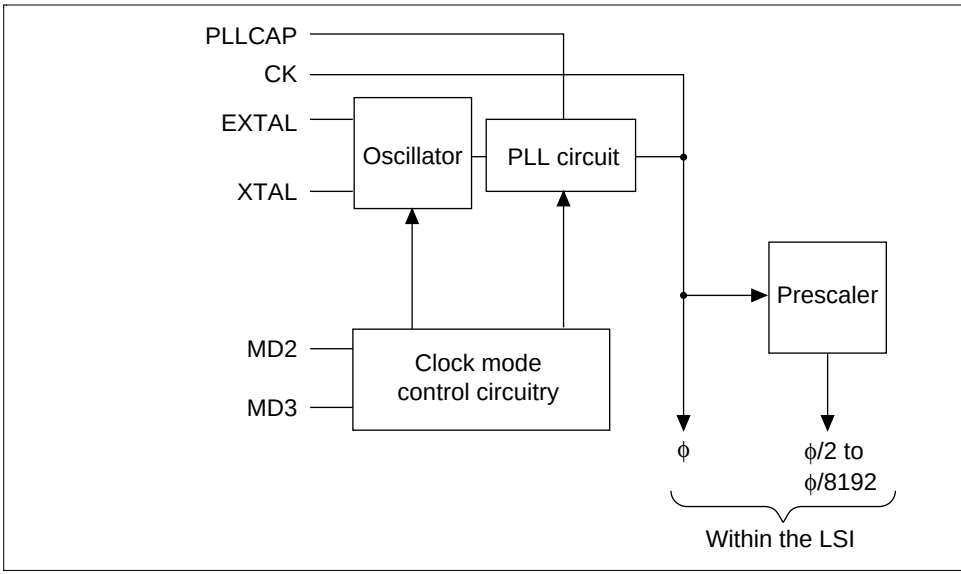


Figure 4.1 Block Diagram of the Clock Pulse Generator

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. Us
damping resistance (Rd) listed in table 4.1. Use a 4–10 MHz crystal oscillator (consult y
concerning the compatibility of the crystal oscillator and the LSI).

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 oscillator with the characteristics listed in table 4.2.

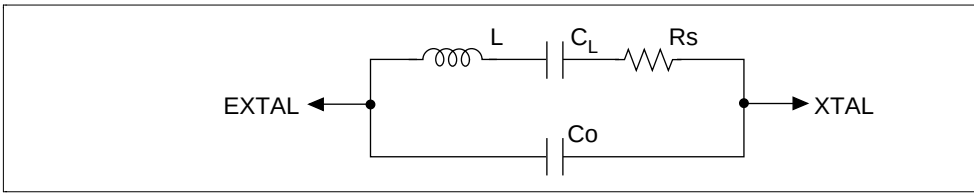


Figure 4.3 Crystal Oscillator Equivalent Circuit

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 clock high level to stop it when in standby mode. During operation, make the external inp frequency 4–10 MHz.

4.3 Prescaler

The prescaler divides the system clock (ϕ) to generate an internal clock (ϕ/2 to ϕ/8192) f
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 ch
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/DA
$\overline{\text{IRQOUT}}$) are set to high-impedance regardless of PFC setting.

Even in standby mode, these six pins become high-impedance regardless of PFC setting
pins enter the normal state after standby mode is cancelled.When abnormalities that halt
oscillator occur except in standby mode, other LSI operations become undefined. In this
operations, including these six pins, become undefined even when the oscillator operatio
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 set
user should refer to the connection examples in this section and perform a careful evalua
oscillator circuit ratings will differ depending on factors such as the oscillator used and t
capacitance of the mounted circuitry. Therefore, the oscillator manufacturer should be c
before a decision is made. Make sure that the voltage applied to the oscillator does not e
maximum rating.

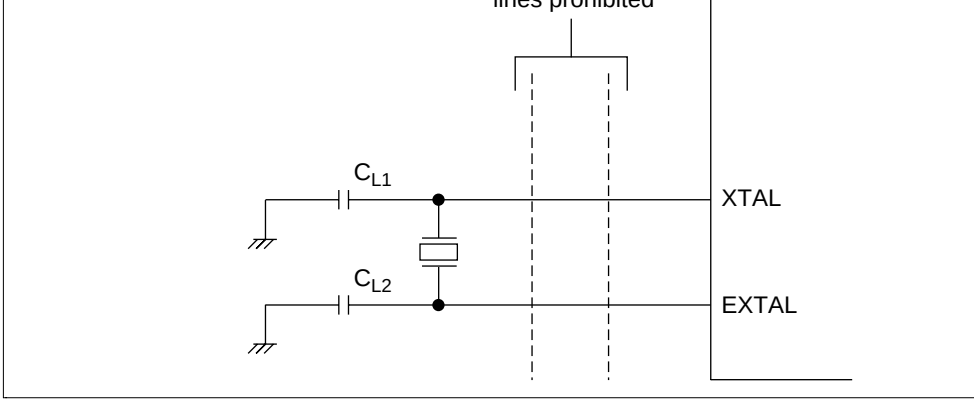


Figure 4.5 Cautions for Oscillator Circuit System Board Design

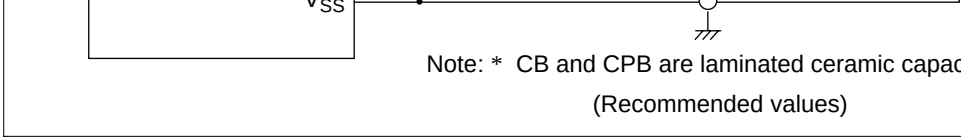


Figure 4.6 Cautions for Use of PLL Oscillator Circuit

Place oscillation stabilization capacitor C1 and resistor R1 near the PLLCAP pin, and en 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 pow 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 gene external oscillator in order to reduce radiation noise.

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

Exception	Source	
Reset	Power-on reset	
	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)	

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

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

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

Interrupts		Detected when instruction is decoded and starts when 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 ex a delayed branch instruction (delay slot).
	Illegal slot instructions	Starts from the decoding of undefined code placed in a branch instruction (delay slot) or of instructions that rew 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 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 addr manual resets). See section 5.1.3, Exception Processing Vector Table, for more inform is then written to the vector base register (VBR) and 1111 is written to the interrupt m (I3–I0) of the status register (SR). The program begins running from the PC address f 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 interrupt priority level is written to the SR's interrupt mask bits (I3–I0). For address e instruction exception processing, the I3–I0 bits are not affected. The start address is th fetched from the exception processing vector table and the program begins running fr address.

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

	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 interrupt are given in section 6, Interrupt Controller (INTC), and table 6.3, Interrup 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: m resets and power-on resets. As table 5.5 shows, both types of resets initialize the internal the CPU. In power-on resets, all registers of the on-chip peripheral modules are initialize manual resets, they are not.

power or when in standby mode (when the clock circuit is halted) or at least 20 t_{cyc} (wh
clock circuit is running). During power-on reset, CPU internal status and all registers of
peripheral modules are initialized. See Appendix C, Pin States, for the status of individu
during the power-on reset status.

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

1. The initial value (execution start address) of the program counter (PC) is fetched fro
exception processing vector table.
2. The initial value of the stack pointer (SP) is fetched from the exception processing v
3. The vector base register (VBR) is cleared to H'00000000 and the interrupt mask bits
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
(PC) and SP and the program begins executing.

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

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

Bus Cycle

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

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

5.4 Interrupts

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

Table 5.7 Interrupt Sources

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

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
Interrupt Controller (INTC), and table 6.3, Interrupt Exception Processing Vectors and
for more information on vector numbers and vector table address offsets.

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 regis through H (IPRA–IPRH).
On-chip peripheral module	0–15	Set with interrupt priority level setting regis through H (IPRA–IPRH).

5.4.2 Interrupt Exception Processing

When an interrupt occurs, its priority level is ascertained by the interrupt controller (INTC is always accepted, but other interrupts are only accepted if they have a priority level high 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 proces CPU saves SR and the program counter (PC) to the stack. The priority level value of the a interrupt is written to SR bits I3–I0. For NMI, however, the priority level is 16, but the va I3–I0 is H'F (level 15). Next, the start address of the exception service routine is fetched f 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 interr exception processing.

5.5 Exceptions Triggered by Instructions

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

5.5.1 Trap Instructions

When a TRAPA instruction is executed, trap instruction exception processing starts up. 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 addr instruction to be executed after the TRAPA instruction.
3. The exception service routine start address is fetched from the exception processing table that corresponds to the vector number specified in the TRAPA instruction. Tha is jumped to and the program starts executing. The jump that occurs is not a delayed

5.5.2 Illegal Slot Instructions

An instruction placed immediately after a delayed branch instruction is said to be placed slot. When the instruction placed in the delay slot is undefined code, illegal slot exceptio processing starts up when that undefined code is decoded. Illegal slot exception processi starts up when an instruction that rewrites the program counter (PC) is placed in a delay processing starts when the instruction is decoded. The CPU handles an illegal slot instru 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 addr delayed branch instruction immediately before the undefined code or the instruction rewrites the PC.
3. The exception service routine start address is fetched from the exception processing table that corresponds to the exception that occurred. That address is jumped to and t program starts executing. The jump that occurs is not a delayed branch.

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**	**Interr**
Immediately after a delayed branch instruction*1	Not accepted	Not ac
Immediately after an interrupt-disabled instruction*2	Accepted	Not ac

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

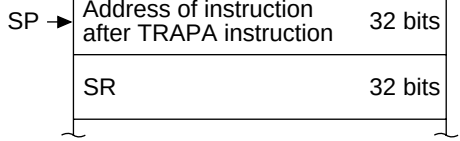
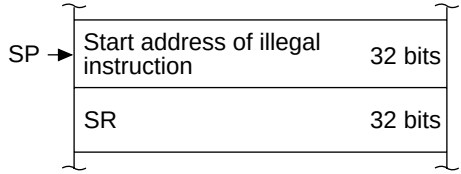
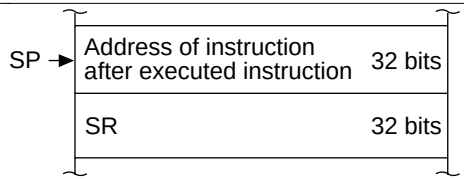
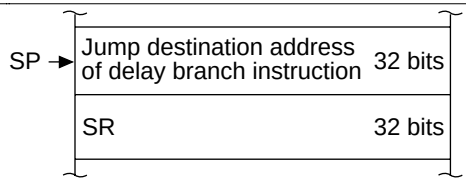
*2 Interrupt-disabled instructions: LDC, LDC.L, STC, STC.L, LDS, LDS.L, STS, S

5.6.1 Immediately after a Delayed Branch Instruction

When an instruction placed immediately after a delayed branch instruction (delay slot) is 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 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, i are not accepted. Address errors are accepted.

	
General illegal instruction	
Interrupt	
Illegal slot instruction	

5.8.3 Address Errors Caused by Stacking of Address Error Exception Processin

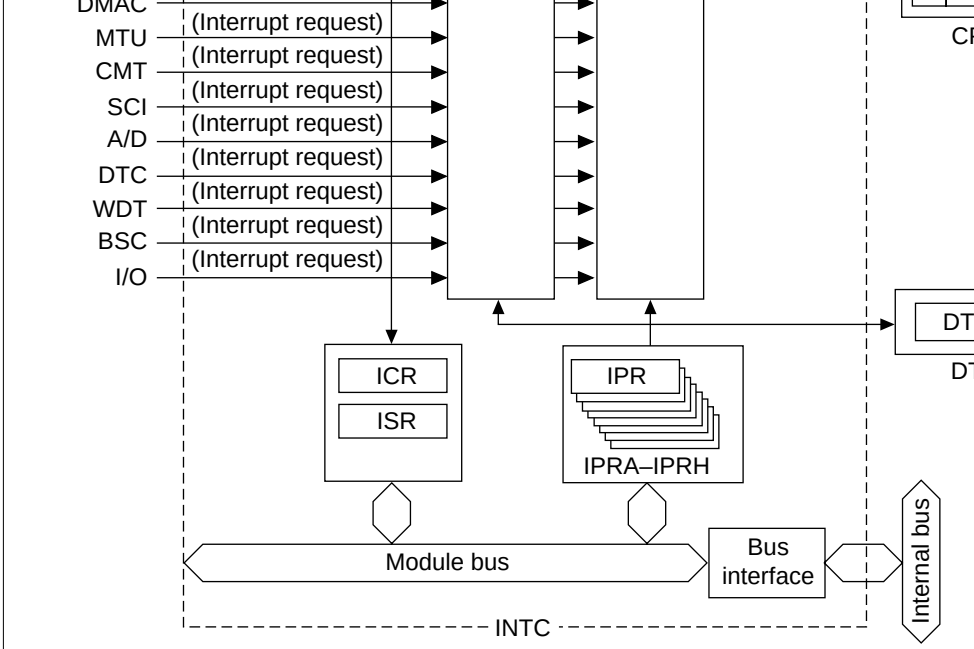
When the stack pointer is not a multiple of four, an address error will occur during stacki exception processing (interrupts, etc.) and address error exception processing will start up as the first exception processing is ended. Address errors will then also occur in the stack this address error exception processing. To ensure that address error exception processing go into an endless loop, no address errors are accepted at that point. This allows program 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 cycl is executed. During stacking of the status register (SR) and program counter (PC), the SP both, so the value of SP will not be a multiple of four after the stacking either. The addres output during stacking is the SP value, so the address where the error occurred is itself ou This means the write data stacked will be undefined.

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



UBC: User break controller
DMAC: Direct memory access controller
MTU: Multifunction timer pulse unit
CMT: Compare match timer
SCI: Serial communication interface
A/D: A/D converter
DTC: Data transfer controller
WDT: Watchdog timer
BSC: Bus state controller (DRAM refresh control section)

I/O: I/O port (port output control sec
ICR: Interrupt control register
ISR: IRQ ststus register
DTER: DTC enable register
IPRA–IPRH: Interrupt priority level setting registers A to H
SR: Status register

Figure 6.1 INTC Block Diagram

6.1.4 Register Configuration

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

Table 6.2 Register Configuration

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

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

6.2.2 User Break Interrupt

A user break interrupt has a priority of level 15, and occurs when the break condition set
user break controller (UBC) is satisfied. User break interrupt requests are detected by edg
held until accepted. User break interrupt exception processing sets the interrupt mask leve
(I3–I0) in the status register (SR) to level 15. For more information about the user break i
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 bit
(IRQ0S–IRQ7S) of the interrupt control register (ICR) to select low level detection or fal
detection for each pin. The priority level can be set from 0 to 15 for each pin using the int
priority registers A and B (IPRA–IPRB).

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

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

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

- Bus state controller (BSC)
- I/O port (I/O)

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

On-chip peripheral module interrupt exception processing sets the interrupt mask level b in the status register (SR) to the priority level value of the on-chip peripheral module int 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 an priorities.

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

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

			H'00000107		(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)	—	

			H'0000016F			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

			H'000001CF			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

			H'0000021F			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 ↕ Low
BSC	CMI	153	H'00000264– H'00000267	0–15 (0)		
I/O	OEI	156	H'00000270– H'00000273	0–15 (0)	IPRH (11–8)	—

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 ↕ Low
	ADI1	137	H'00000224– H'00000227	0–15 (0)		

Bit:	7	6	5	4	3	2	1
Initial value:	0	0	0	0	0	0	0
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	DMA
Interrupt priority register D	MTU0	MTU0	MTU1	MTU
Interrupt priority register E	MTU2	MTTU2	MTU3	MTU
Interrupt priority register F	MTU4	MTU4	SCI0	SCI1
Interrupt priority register G	A/D(A/D0, A/D1)*	DTC	CMT0	CMT
Interrupt priority register H	WDT, BSC	I/O	Reserved	Rese

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

pin NMI and $\overline{IRQ0}$ –$\overline{IRQ7}$ and indicates the input signal level to the NMI pin. A power-
initializes ICR but the standby mode does not.

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

Bit:	7	6	5	4	3	2	1
	IRQ0S	IRQ1S	IRQ2S	IRQ3S	IRQ4S	IRQ5S	IRQ6S
Initial value:	0	0	0	0	0	0	0
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.
  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

6.3.3 IRQ Status Register (ISR)

The ISR is a 16-bit register that indicates the interrupt request status of the external interr
pins $\overline{\text{IRQ0}}$–$\overline{\text{IRQ7}}$. When IRQ interrupts are set to edge detection, held interrupt requests c
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
	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R

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

		3. When a DTC transfer due to IRQn interrupt ha 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

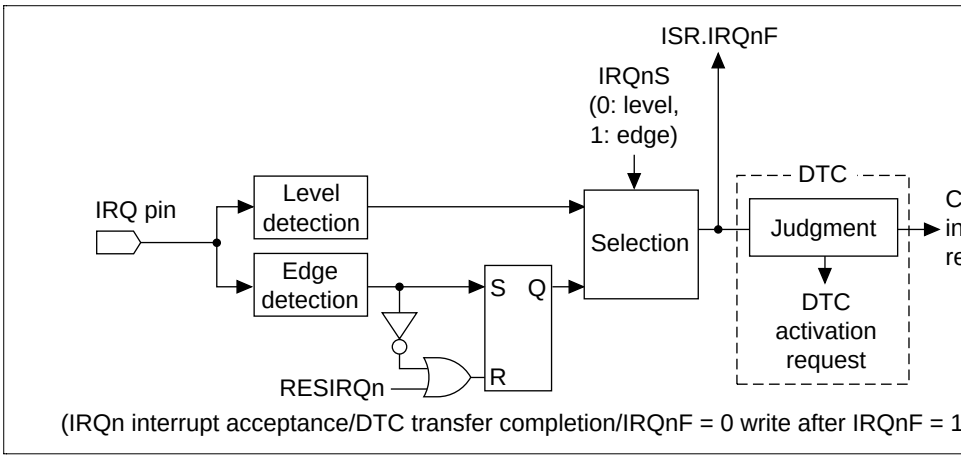
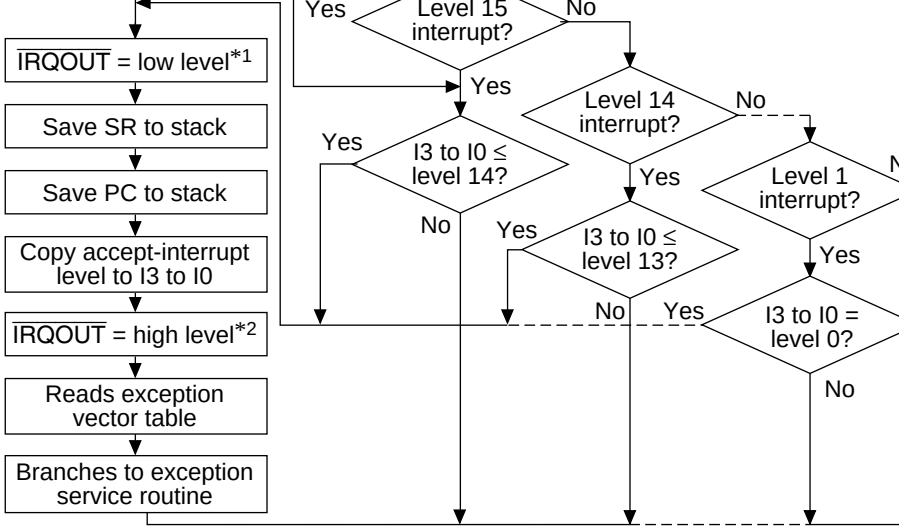


Figure 6.2 External Interrupt Process

accessing the IRQ status register (ISR). See section 6.2.5, IRQ Interrupts, for details.
held pending due to edge detection are cleared by a power-on reset or a manual reset.
these interrupts have the same priority level or if multiple interrupts occur within a sin
module, the interrupt with the highest default priority or the highest priority within its
setting range (as indicated in table 6.3) is selected.

3. The interrupt controller compares the priority level of the selected interrupt request wi
interrupt mask bits (I3–I0) in the CPU's status register (SR). If the request priority lev
equal to or less than the level set in I3–I0, the request is ignored. If the request priority
higher than the level in bits I3–I0, the interrupt controller accepts the interrupt and sen
interrupt request signal to the CPU.
4. When the interrupt controller accepts an interrupt, a low level is output from the $\overline{\text{IRQ}}$
5. The CPU detects the interrupt request sent from the interrupt controller when it decod
next instruction to be executed. Instead of executing the decoded instruction, the CPU
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 (I
the status register (SR).
8. When the accepted interrupt is sensed by level or is from an on-chip peripheral modul
level is output from the $\overline{\text{IRQOUT}}$ pin. When the accepted interrupt is sensed by edge,
level is output from the $\overline{\text{IRQOUT}}$ pin at the point when the CPU starts interrupt excep
processing instead of instruction execution as noted in (5) above. However, if the inte
controller accepts an interrupt with a higher priority than one it is in the midst of acce
$\overline{\text{IRQOUT}}$ pin will remain low level.
9. The CPU reads the start address of the exception service routine from the exception v
table for the accepted interrupt, jumps to that address, and starts executing the progra
This jump is not a delay branch.



I3 to I0: Interrupt mask bits of status register

Notes: *1 IRQOUT is the same signal as the interrupt request signal to the CPU figure 6.1). Thus, it is output when there is a higher priority interrupt r than the one in the I3 to I0 bits of the SR.

*2 When the accepted interrupt is sensed by edge, the IRQOUT pin bec high level at the point when the CPU starts interrupt exception proces instead of instruction execution (before SR is saved to the stack).
If the interrupt controller has accepted another interrupt with a higher and has output an interrupt request to the CPU, the IRQOUT pin will low level.

Figure 6.3 Interrupt Sequence Flowchart

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 a interrupt request until the interrupt exception processing starts and fetching of the first ins of the interrupt service routine begins. Figure 6.5 shows the pipeline when an IRQ interru accepted.

sequence currently being executed by CPU				interrupt or addres exception process + m1 + m2 + m3 + interrupt-masking follows, however, may be even long
Time from start of interrupt exception processing until fetch of first instruction of exception service routine starts		5 + m1 + m2 + m3		Performs the PC a saves and vector fetch.
Interrupt response time	Total:	7 + m1 + m2 + m3	9 + m1 + m2 + m3	
	Minimum:	10	12	0.35–0.42 µs at 2
	Maximum:	12 + 2 (m1 + m2 + m3) + m4	13 + 2 (m1 + m2 + m3) + m4	0.67–0.70 µs at 2

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

F: Instruction fetch (instruction fetched from memory where program is stored).
D: Instruction decoding (fetched instruction is decoded).
E: Instruction execution (data operation and address calculation is performed according to the results of decoding).
M: Memory access (data in memory is accessed).

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 r
to the INTC. The masking condition is listed below:

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

The INTC masks CPU interrupts when the corresponding DTE bit is a 1. The DTE clear c
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 trans

Where: DTECLR = DISEL + counter 0.

Figure 6.6 shows a control block diagram.

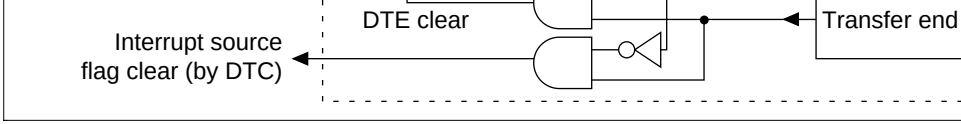


Figure 6.6 Interrupt Control Block Diagram

6.6.1 Handling DTC Activating and CPU Interrupt Sources, but Not DMAC A 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 inter request to the CPU. The activating source does not clear.
5. The CPU clears interrupt sources with its interrupt processing routine. It then confirr transfer counter value. When the transfer counter value ≠ 0, it sets the DTE bit to 1 a the next data transfer. If the transfer counter value = 0, it performs the necessary end processing in the interrupt processing routine.

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 re 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 interr 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 Sourc

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 inte processing routine.

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 use interrupt exception processing routine can be run.
- Select either to break in the CPU instruction fetch cycle before the instruction is exec after.

7.1.2 Block Diagram

Figure 7.1 shows a block diagram of the UBC.

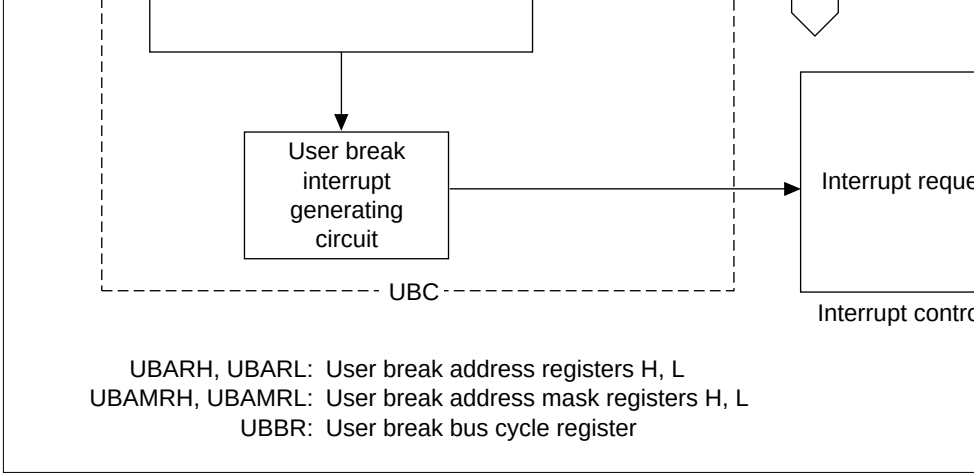


Figure 7.1 User Break Controller Block Diagram

7.1.3 Register Configuration

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

7.2.1 User Break Address Register (UBAR)

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

UBARH:

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

Bit:	7	6	5	4	3	2	1
UBARH	UBA23	UBA22	UBA21	UBA20	UBA19	UBA18	UBA17
Initial value:	0	0	0	0	0	0	0
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 t 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 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 regi (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 addre established in the UBARH, and UBAMRL designates whether to mask any of the break a bits established in the UBARL. Resets and hardware standbys initialize both UBAMRH a UBAMRL to H'0000. They are not initialized in manual reset or software standby mode.

UBAMRH:

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

Bit:	7	6	5	4	3	2	1
UBAMRH	UBM23	UBM22	UBM21	UBM20	UBM19	UBM18	UBM17
Initial value:	0	0	0	0	0	0	0
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 designate whether to mask any of the break address 31–16 bits (UBA31–UBA16) es in the UBARH.

- UBAMRL Bits 15–0—User Break Address Mask 15–0 (UBM15–UBM0): These bit designate whether to mask any of the break address 15–0 bits (UBA15–UBA0) estab the UBARL.

Bits 15–0: UBMn	Description
0	Break address UBAn is included in the break conditions (initi
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 f 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 softw standby mode.

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 b 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 whet 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 cycl

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 co to be word-size accesses (even when there are instructions in on-chip memory ar instruction fetches are done simultaneously in 1 bus cycle).

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

conditions are in agreement. When using user break interrupts, always be certain to es
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 b
fulfilled. When the set conditions are satisfied, the UBC sends a user break interrupt r
signal to the interrupt controller (INTC).
3. The interrupt controller checks the accepted user break interrupt request signal's prior
The user break interrupt has priority level 15, so it is accepted only if the interrupt ma
in bits I3–I0 in the status register (SR) is 14 or lower. When the I3–I0 bit level is 15, t
break interrupt cannot be accepted but it is held pending until user break interrupt exc
processing can be carried out. Consequently, user break interrupts within NMI except
service routines cannot be accepted, since the I3–I0 bit level is 15. However, if the I3-
level is changed to 14 or lower at the start of the NMI exception service routine, user
interrupts become acceptable thereafter. Section 6, Interrupt Controller (INTC), descr
handling of priority levels in greater detail.
4. The INTC sends the user break interrupt request signal to the CPU, which begins user
interrupt exception processing upon receipt. See Section 6.4, Interrupt Operation, for
interrupt exception processing.

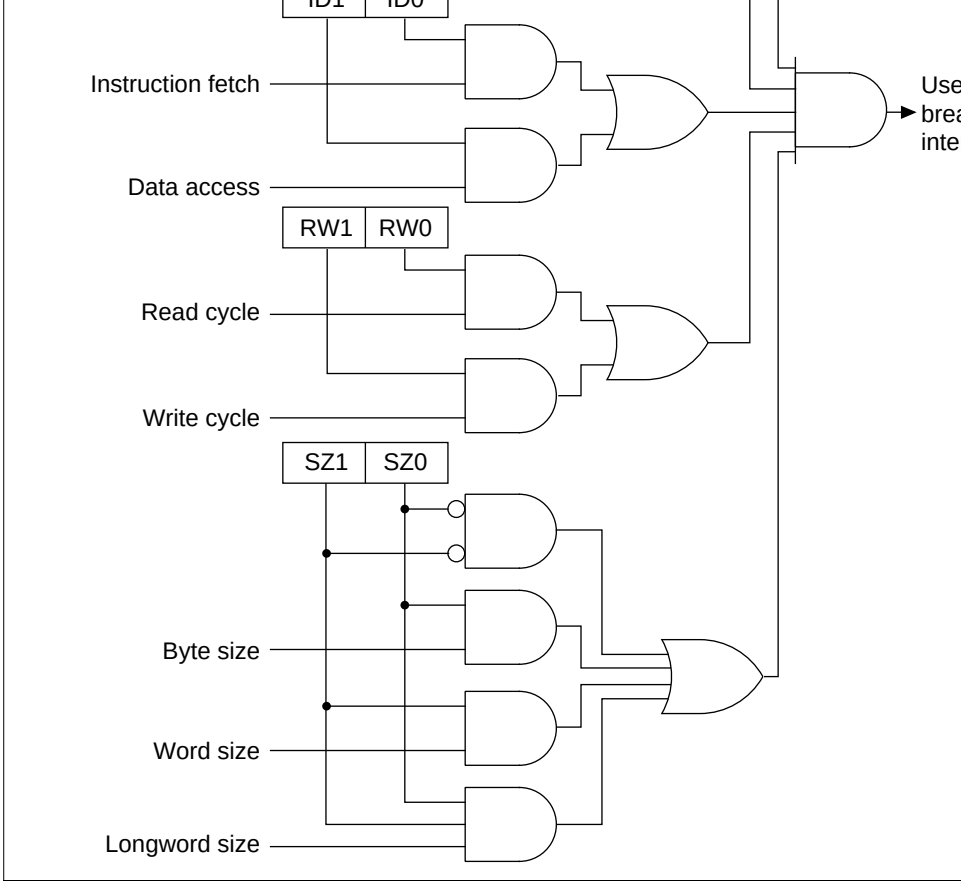


Figure 7.2 Break Condition Judgment Method

stack in user break interrupt exception processing is the address that matches the break co
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 in
(delay slot), or on an instruction that follows an interrupt-disabled instruction, however, t
break interrupt is not accepted immediately, but the break condition establishing instructi
executed. The user break interrupt is accepted after execution of the instruction that has a
the interrupt. In this case, the PC value saved is the start address of the instruction that wi
executed after the instruction that has accepted the interrupt.

Break on Data Access (CPU/Peripheral): The program counter (PC) value is the top ad
the next instruction after the last instruction executed before the user break exception pro
started. When data access (CPU/peripheral) is set as a break condition, the place where th
will occur cannot be specified exactly. The break will occur at the instruction fetched clos
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 pos
the instruction at H'00000402 to accept an interrupt, the user break exception processing
executed after execution of that instruction. The instruction at H'00000404 is not executed
PC value saved is H'00000404.

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
address. However, if the first instruction fetch address after the branch is an odd address
these conditions, user break interrupt exception processing will be done after address err
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 eve

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 DM
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 i
the second of these two instructions but the contents of the UBC break condition register
changed so as to alter the break condition immediately after the first of the two instructio
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, instruction
fetched and executed as follows:

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

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

   Instruction execution: Conditional branch instruction execution → Branch desti
   instruction execution
2. TRAPA instruction, branch taken: TRAPA

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

instruction fetch and execution, the kind of overrun fetch instructions noted above do become objects of a break. If data access breaks are also included with instruction fe as break conditions, a break occurs because the instruction overrun fetch is also rega 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 priority than a user break is in contention and is accepted in the decode stage for that ins (or the next instruction), user break exception handling may not be performed after com 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 instru following a branch (BRA, BRAF, BT, BF, BT/S, BF/S, BSR, BSRF, JMP, JSR, RTS, R exception handling), and exception handling for this branch destination instruction with 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 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 th destination instruction on execution of the branch, a user break will not be generated eve break condition has been set for the first jump destination instruction fetch.

— 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 destin
- 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 tr
- Transfers can be activated by software

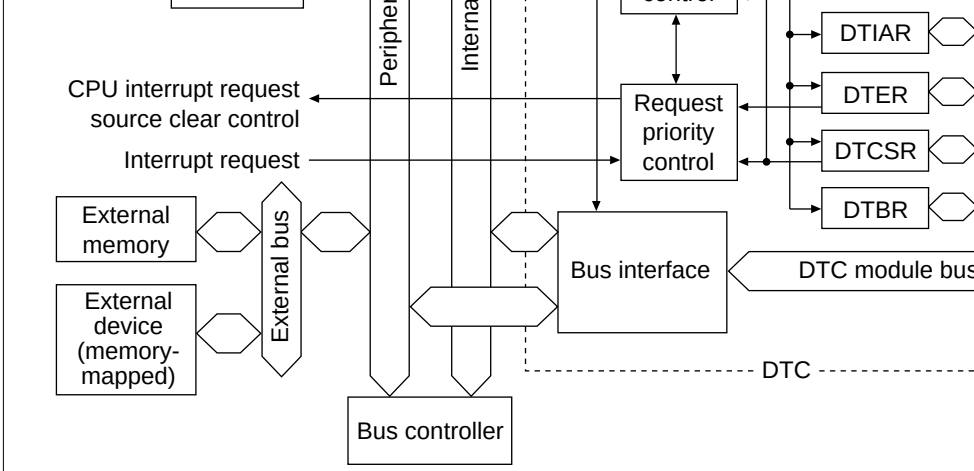


Figure 8.1 DTC Block Diagram

DTC initial address register	DTIAR	—	Undefined	—	—
DTC transfer count register A	DTCRA	—*2	Undefined	—*2	—
DTC transfer count register B	DTCRB	—*2	Undefined	—*2	—
DTC enable register A	DTEA	R/W	H'00	H'FFFF8700	8,
DTC enable register B	DTEB	R/W	H'00	H'FFFF8701	8,
DTC enable register C	DTEC	R/W	H'00	H'FFFF8702	8,
DTC enable register D	DTED	R/W	H'00	H'FFFF8703	8,
DTC enable register E	DTEE	R/W	H'00	H'FFFF8704	8,
DTC control/status register	DTCSR	R/(W)*3	H'0000	H'FFFF8706	8,
DTC information base register	DTBR	R/W	Undefined	H'FFFF8708	16

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 mo
contents of this register is located in memory.

- Bits 15–14—Source Address Mode 1, 0 (SM1, SM0): These bits designate whether to 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 longwo
1	1	DTSAR is decremented after transfer (–1 for byte unit transfer, –2 for word, –4 for longwo

- Bits 13–12—Destination Address Mode 1, 0 (DM1, DM0): These bits designate whet 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 longwo
1	1	DTDAR is decremented after transfer (–1 for byte unit transfer, –2 for word, –4 for longwo

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 mo
  bit designates whether the source side or destination side will be the repeat area or bl

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 continuo
  data transfers with the same activating source. Continued transfer information is read
  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 informatio transfer)

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 tran source address. An even address indicates that the transfer size is word; a multiple-of-fou means it is longword. The contents of this register is located in memory.

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

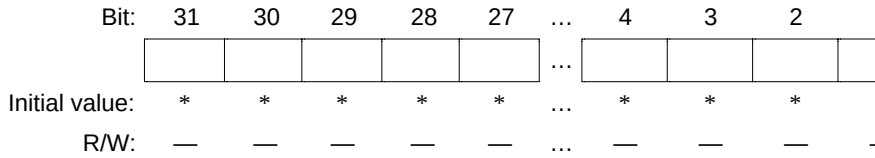
Note: * Initial value is undefined.

8.2.3 DTC Destination Address Register (DTDAR)

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

destination address in the repeat area.

The contents of this register are located in memory.



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 register are located in memory.

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

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. T
contents of this register is located in memory. The block length is 1 when the set value is
65535 when it is H'FFFF, and 65536 when it is H'0000.

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

Bit:	7	6	5	4	3	2	1
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
interrupt source that activates the DTC. They set disable/enable for DTC activation for ea
interrupt source. When a bit is 1, DTC activation by the corresponding interrupt source is
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 d
initialize DTER.

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

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

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

Bit:	7	6	5	4	3	2	1
Bit name:	DTVEC7	DTVEC6	DTVEC5	DTVEC4	DTVEC3	DTVEC2	DTVEC1
Initial value:	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

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
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 9—Address Error Flag (AE): Indicates that an address error by the DTC has occu
  When the AE bit is set, DTC transfers are not allowed even if the DTER bit is set to 1
  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 DT
  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
  vector addresses for DTC activation by software. A vector address is calculated as H'0
  DTVEC[7:0]. Always specify 0 for DTVEC0. 8 bits are available, so you can specify
  H'00 (0)–H'FE (254).

Bit:	7	6	5	4	3	2	1
Initial value:	*	*	*	*	*	*	*
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, 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 mak possible to perform data transfers for an arbitrary number of channels. Further, setting th bit to 1 makes it possible to perform multiple transfers continuously through one DTC tr request.

There are three DTC transfer modes: normal mode, repeat mode, and block transfer mod DTC transfer, the transfer source address and transfer destination address are incremente 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.

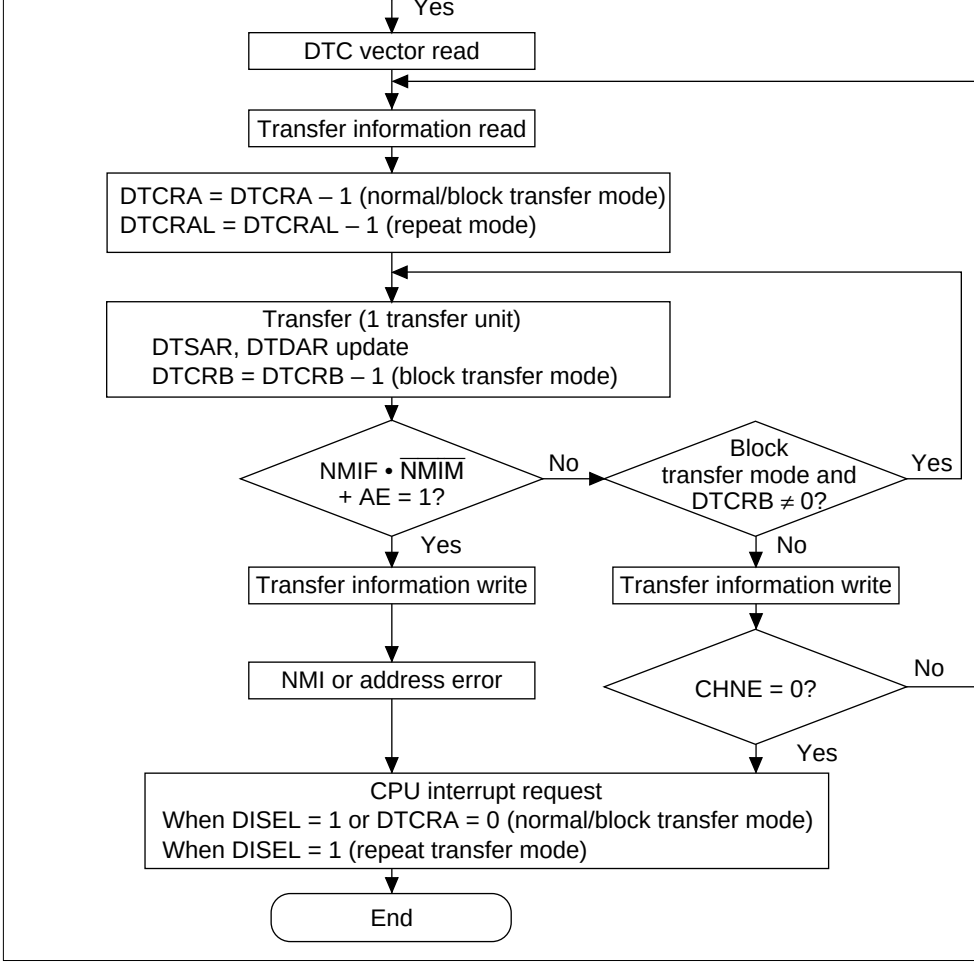


Figure 8.2 DTC Operation Flowchart

a request is made only after the completion of a designated number of data transfers. Wh
SWDTCE interrupt is requested of the CPU, the SWDTE bit of the DTCSR is automatic
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
activated in accordance with the default priority rankings shown in table 8.2.

Figure 8.3 shows a block diagram of activating source control.

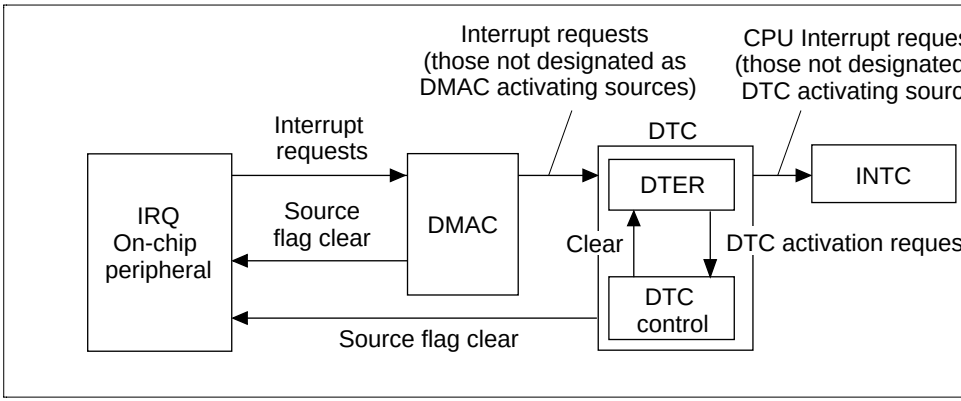


Figure 8.3 Activating Source Control Block Diagram

8.3.3 DTC Vector Table

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

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

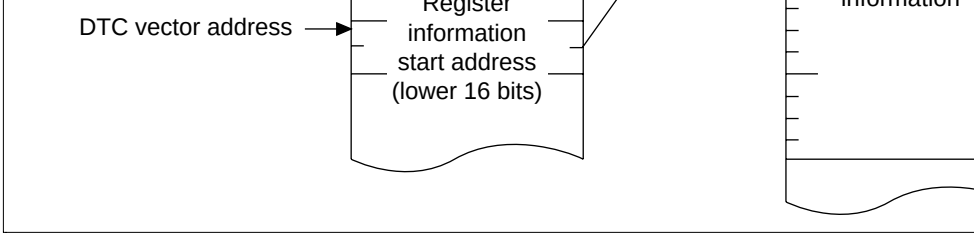


Figure 8.4 Correspondence between DTC Vector Address and Register Inform

	TGI3C	H'0000040E–H'0000040F	DTEA0	Arbitrary	Arbitrary	
	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	L

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 inf
start addresses are designated by DTBR for the upper 16 bits, and the DTC vector table fo
lower 16 bits.

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

Fundamentally, certain RAM areas are designated for addresses storing register informati

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 to
transfer count is 1 to 65536. An interrupt request is generated to the CPU when the trans
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 Transfer Information Writ	
Register	**Function**	**When DTCRA is other than 1**	**When DTCI**
DTMR	Operation mode control	DTMR	DTMR
DTCRA	Transfer count	DTCRA – 1	DTCRA – 1
DTSAR	Transfer source address	Increment/decrement/ fixed	Increment/d fixed
DTDAR	Transfer destination address	Increment/decrement/ fixed	Increment/d fixed

8.3.6 Repeat Mode

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

Register	Function	When DTCRA is other than 1	When DTCR
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 b
DTSAR	Transfer source address	Increment/decrement/ fixed	(DTS = 0) Inc decrement/ fi (DTS = 1) DT
DTDAR	Transfer destination address	Increment/decrement/ fixed	(DTS = 0) DT (DTS = 1) Inc decrement/ fi

8.3.7 Block Transfer Mode

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

The block length is specified between 1 and 65536. When a 1-block transfers ends, the ad register of the designated block area is returned to its initial state. Other address registers consecutively incremented, decremented, or remain fixed. The block transfer count is 1 to 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 pos

Table 8.5 shows the register functions for block transfer mode.

8.3.8 Operation Timing

Figure 8.6 shows a DTC operation timing example.

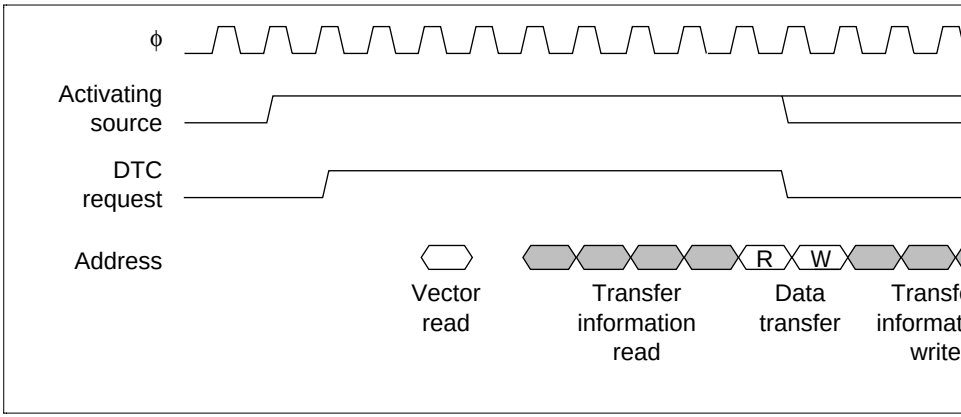


Figure 8.6 DTC Operation Timing Example (Normal Mode)

When register information is located in on-chip RAM, each mode requires 4 cycles for t
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 sh
state counts needed for execution state.

Access Objective			On-chip RAM	On-chip ROM	Internal I/O Register		External D	
Bus width			32	32	32		8	16
Access state			1	1	2*1	3*2	2	2
Execution state	Vector read	S_I	—	1	—		4	2
	Register information read/write	S_J	1	1	—		8	4
	Byte data read	S_K	1	1	2	3	2	2
	Word data read	S_K	1	1	2	3	4	2
	Long word data read	S_K	1	1	4	6	8	4
	Byte data write	S_L	1	1	2	3	2	2
	Word data write	S_L	1	1	2	3	4	2
	Long word write	S_L	1	1	4	6	8	4
	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. ∑ indicates the numb transfers by one activating source (count + 1 when CHNE bit is 1).

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

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
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 DT
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 clear
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 DT
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 increm
(DM1 = 1, DM0 = 0), normal mode (MD1 = MD0 = 0), byte size (SZ1 = SZ0 = 0), o
transfer per activating source (CHNE = 0), and a CPU interrupt request after the desi
number of data transfers (DISEL = 0). 128 (H'0080) is set in DTCRA, the RDR addr
SCI is set in DTSAR, and the start address of the RAM storing the receive data is set
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.

that DTER have ended, or disable the transfer source for each channel so that DTC tra
corresponding to that DTER will not occur. The above restrictions do not apply for A
due to change in the access method of DTER. However, take caution when changing I
mask, since modification of the program is required.

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

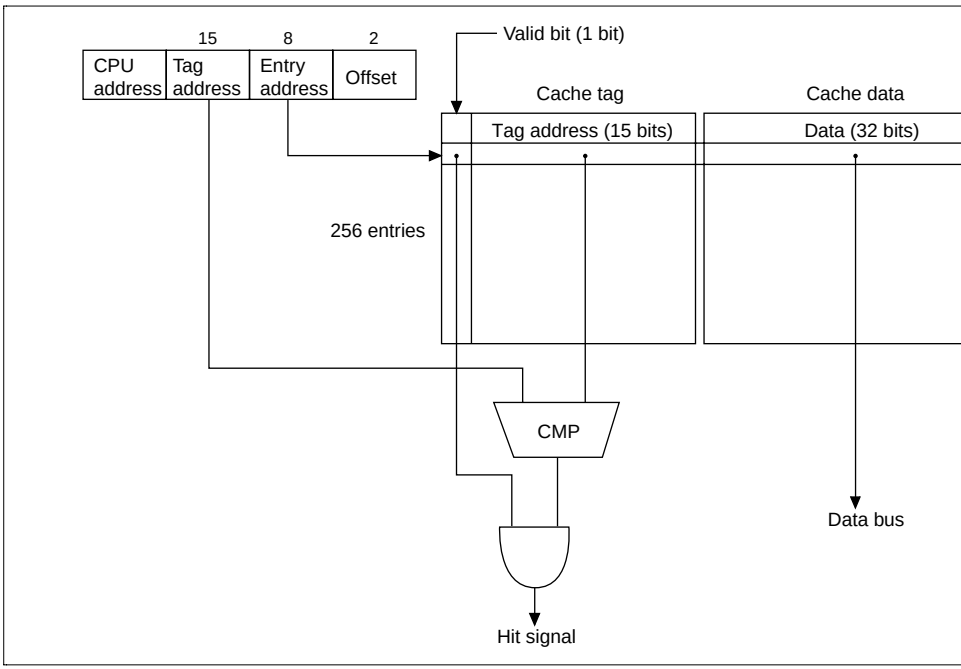


Figure 9.1 Cache Tag and Cache Data Configuration

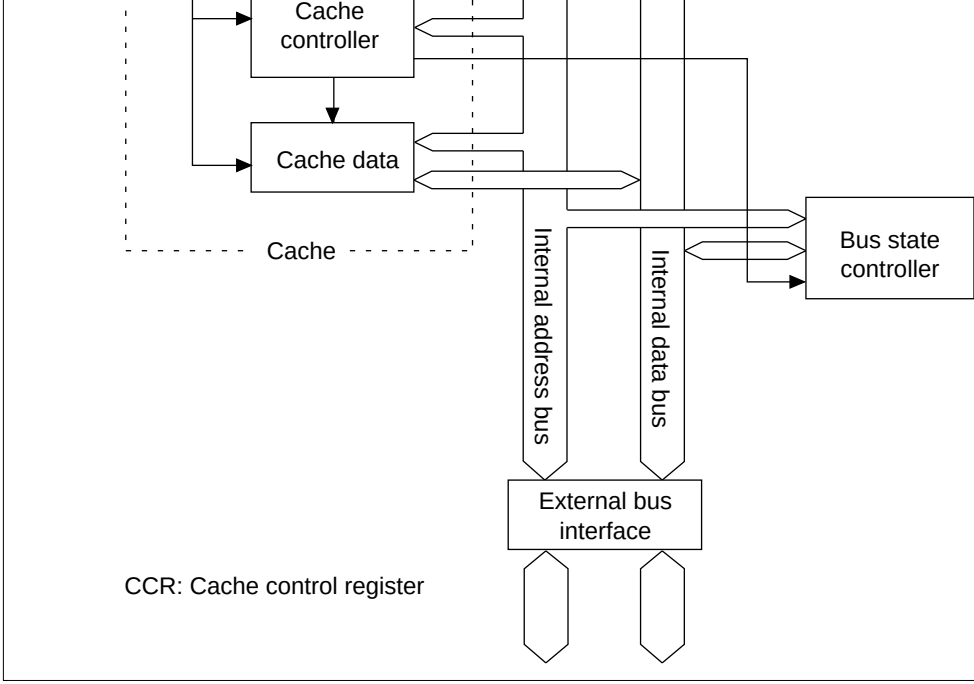


Figure 9.2 Cache Block Diagram

9.1.3 Register Configuration

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

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

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

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

- Bit 4—DRAM Space Cache Enable (CEDRAM): Selects whether to use DRAM spa cache object (enable) or to exclude it (disable). A 0 disables, and a 1 enables such us

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

1	CS2 space cache enabled

- Bit 1—CS1 Space Cache Enable (CECS1): Selects whether to use CS1 space as a cac (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 cac (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 add and a data array, where addresses (tag address, including valid bit) and data (4-byte line l cache control are recorded. The special cache space is shown in table 9.2. It can be used a RAM space when the cache is not being used.

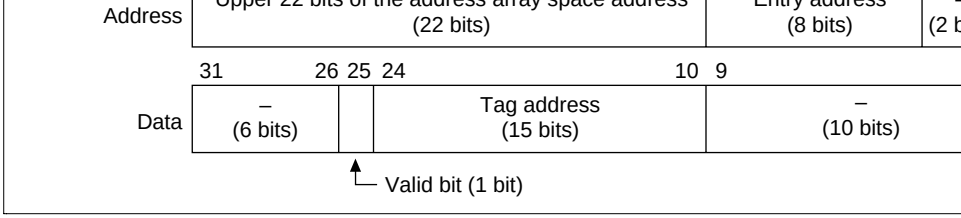


Figure 9.3 Cache Address Array

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

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

9.3.2 Cache Data Array Read/Write Space

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

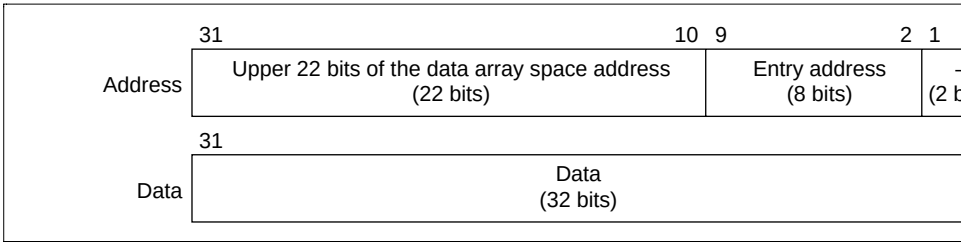


Figure 9.4 Cache Data Array

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

Data Array Write: Designates entry address and writes designated data to the correspo

While the cache is enabled, it is not possible to write to the address array or data array via CPU, DMAC, or DTC, and a read will return an undefined value. The cache must be disa 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 befor cache fill (access from external memory in the event of a cache miss), as shown in figure However, in the case of consecutive cache misses, idle cycles are not generated for the se 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 of the bus cycle (or the last bus cycle when two or four bus cycles are generated, such as 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 immedi before the end of the bus cycle is extended by an additional cycle. In RAS down mode, th 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 Norm (No Wait, No CS Assert Extension)

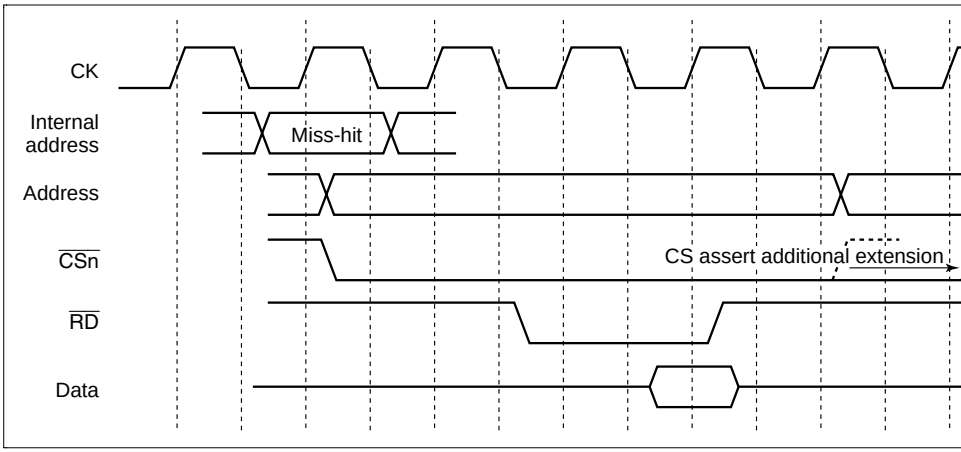


Figure 9.6 Cache Fill Timing in Case of Consecutive Cache Misses from Norma (No Wait, CS Assert Extension)

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

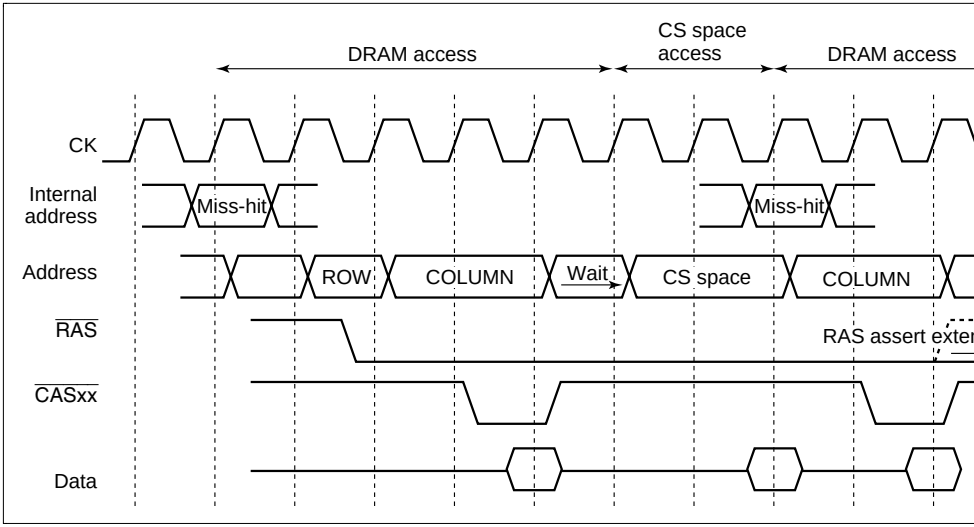


Figure 9.8 Cache Fill Timing in Case of Consecutive Cache Misses from DRAM (RAS Down Mode, TPC = 0, RCD = 0, No Wait)

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 cycle generation is performed. The next hit operates as a cache hit.

- Address space is divided into five spaces
  - A maximum linear 2 Mbytes for on-chip ROM effective mode, and a maximum l 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 connecte
- 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)

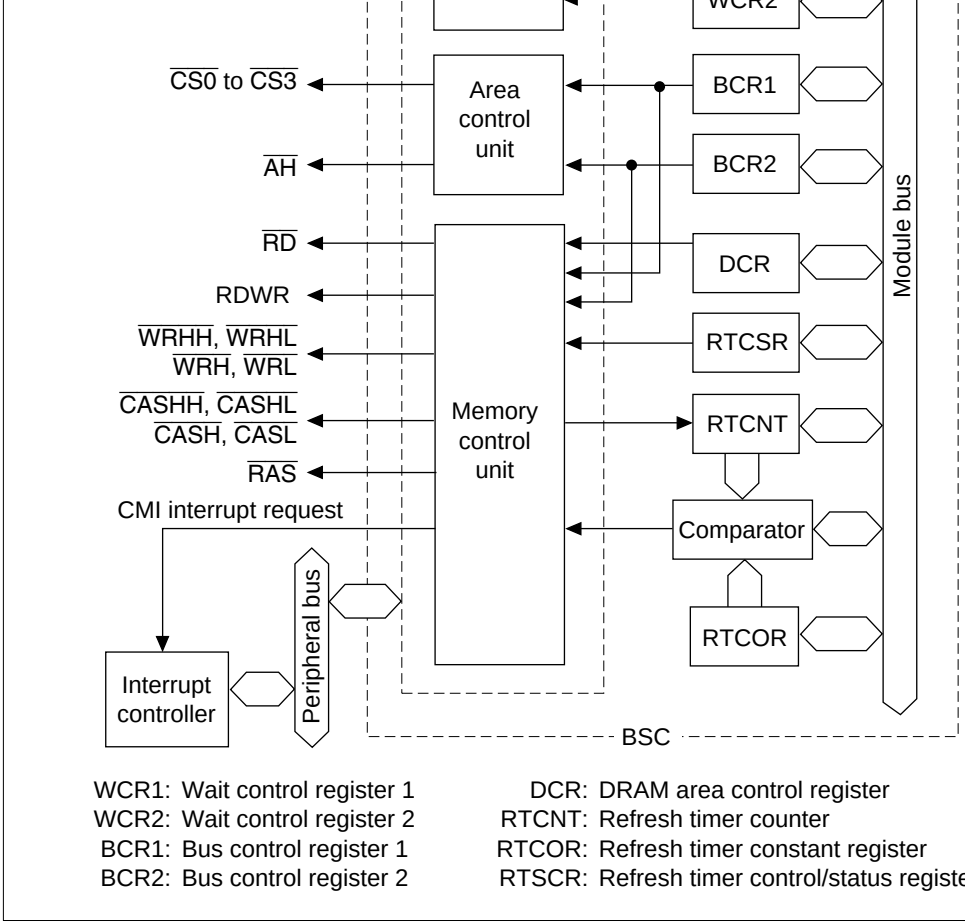


Figure 10.1 BSC Block Diagram

		output during DRAM access.
$\overline{\text{WRHH}}$	O	Strobe that indicates a write cycle to the most significant byte (D31–D2 ordinary space/multiplex I/O. Also output during DRAM access.
$\overline{\text{WRHL}}$	O	Strobe that indicates a write cycle to the 2nd byte (D23–D16) for ordina space/multiplex I/O. Also output during DRAM access.
$\overline{\text{WRH}}$	O	Strobe that indicates a write cycle to the 3rd byte (D15–D8) for ordinary space/multiplex I/O. Also output during DRAM access.
$\overline{\text{WRL}}$	O	Strobe that indicates a write cycle to the least significant byte (D7–D0) ordinary space/multiplex I/O. Also output during DRAM access.
RDWR	O	Strobe indicating a write cycle to DRAM (used for DRAM space)
$\overline{\text{RAS}}$	O	RAS signal for DRAM (used for DRAM space)
$\overline{\text{CASHH}}$	O	CAS signal when accessing the most significant byte (D31–D24) of DR (used for DRAM space)
$\overline{\text{CASHL}}$	O	CAS signal when accessing the 2nd byte (D23–D16) of DRAM (used fo space)
$\overline{\text{CASH}}$	O	CAS signal when accessing the 3rd byte (D15–D8) of DRAM (used for space)
$\overline{\text{CASL}}$	O	CAS signal when accessing the least significant byte (D7–D0) of DRAM for DRAM space)
$\overline{\text{AH}}$	O	Signal to hold the address during address/data multiplex
$\overline{\text{WAIT}}$	I	Wait state request signal
$\overline{\text{BREQ}}$	I	Bus release request input
$\overline{\text{BACK}}$	O	Bus use enable output

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

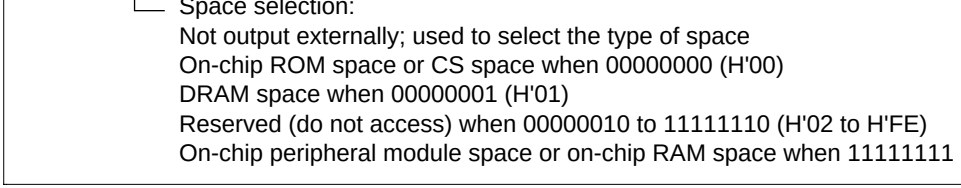


Figure 10.2 Address Format

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 a map for on-chip ROM ineffective mode.

H'02000000–H'FFFF7FFF	Reserved	Reserved		
H'FFFF8000–H'FFFF87FF	On-chip peripheral module	On-chip peripheral module	2 kbytes	8/16
H'FFFF8800–H'FFFFEFFF	Reserved	Reserved		
H'FFFFF000–H'FFFFFFFF	On-chip RAM	On-chip RAM	4 kbytes	32 bi

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

*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.

	module			
H'FFFF8800–H'FFFFEFFF	Reserved	Reserved		
H'FFFFF000–H'FFFFFFFF	On-chip RAM	On-chip RAM	4 kbytes	32

Notes: 1. Do not access reserved spaces. Operation cannot be guaranteed if they are a
2. In the single-chip mode, spaces other than on-chip ROM, on-chip RAM and o 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, sele 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 word (16 bits) or less.

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

- Bits 15, 14, 12–9—Reserved: These bits always read as 0. The write value should alw
- Bit 13—MTU Read/Write Enable (MTURWE): When this bit is 1, MTU control regis 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/da multiplex I/O space. When address/data multiplex I/O space is selected, the address a are multiplexed and output from the data bus. When CS3 space is an address/data mul 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 effective only when CS3 space is ordinary space. When CS3 space is an address/data 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 4—CS0 Space Long Size Specification (A0LG): Specifies the CS0 space bus siz

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 inef 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 0. This is effective only when CS3 space is ordinary space. When CS3 space is an ad 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 bi ordinary space).

- Bit 2—CS2 Space Size Specification (A2SZ): Specifies the CS2 space bus size when 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 bi

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 m
the CS0 space bus size is specified by the mode pin. However, even in on-chip RO
effective mode, this bit is ignored when A0LG = 1; CS0 space bus size becomes lo
(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 signa
extension of each CS space.

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

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

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

0	0	No idle cycle after accessing CS3
	1	Inserts one idle cycle after accessi space
1	0	Inserts two idle cycles after access space
	1	Inserts three idle cycles after acce space (initial value)

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

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

details.

CW3 specifies the continuous access idles for CS3 space; CW2 specifies the continuo
idles for CS2 space; CW1 specifies the continuous access idles for CS1 space and CW
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)

1	CS3 space 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 space.

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

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 nu
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)

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 valu

10.2.4 Wait Control Register 2 (WCR2)

WCR2 is a 16-bit read/write register that specifies the number of access cycles for DRA 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 reset software standbys.

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

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

- Bits 3–0—CS Space DMA Single Address Mode Access Wait Specification (DSW3, DSW1, DSW0): Specifies the number of waits for CS space access (0–15) during DM 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 valu

10.2.5 DRAM Area Control Register (DCR)

DCR is a 16-bit read/write register that selects the number of waits, operation mode, num 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 standbys.

$\overline{\text{RAS}}$ 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 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) 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

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 accessing a different external space (CS space) or when doing a DRAM write, after D 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

	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 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 ini manual resets or software standbys.

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

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

- Bit 5—Compare Match Interrupt Enable (CMIE): Enables or disables an interrupt req 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 am 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

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 interrupt requests.

RTCNT counts up with the clock selected by the CKS2–CKS0 bits of the RTCSR. RTC can always be read/written by the CPU. When RTCNT matches RTCOR, RTCNT is cle H'0000 and the CMF flag of the RTCSR is set to 1. If the RFSH bit of RTCSR is 1 and t bit is 0 at this time, a CAS-before-RAS refresh is performed. Additionally, if the CMIE 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 software standbys.

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

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

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 reset
software standbys.

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

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

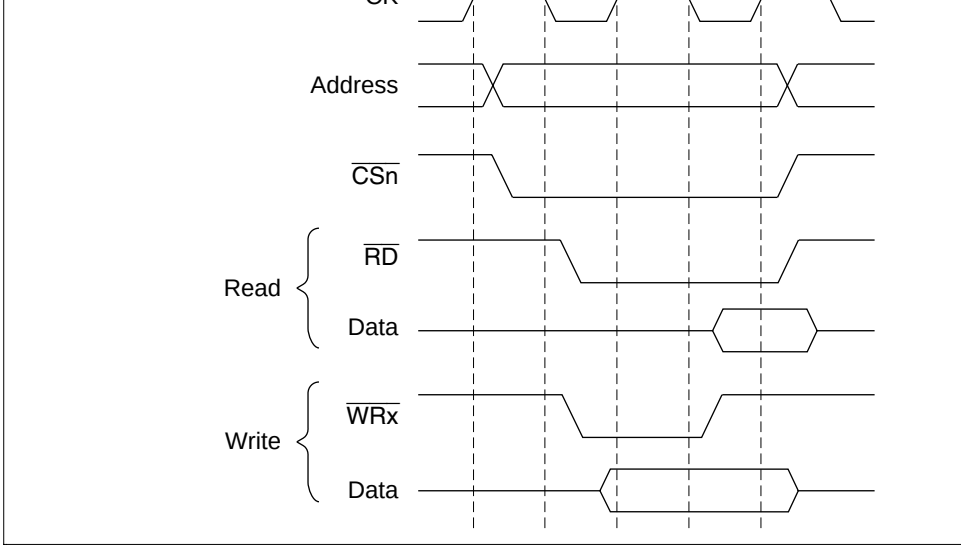


Figure 10.3 Basic Timing of Ordinary Space Access

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

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

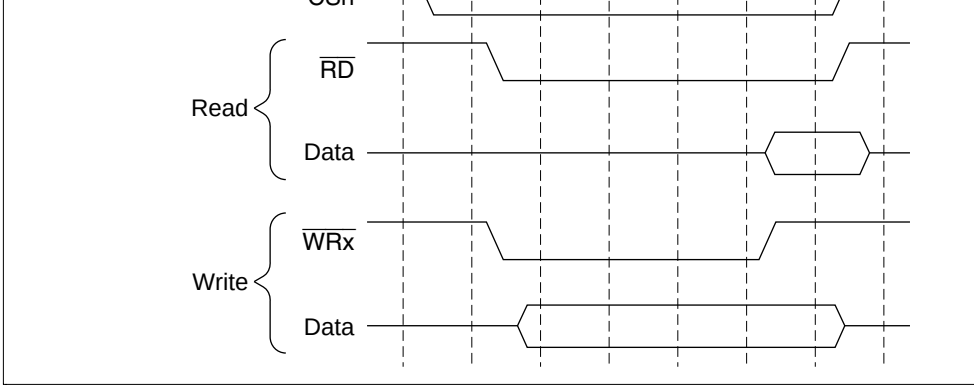


Figure 10.4 Wait Timing of Ordinary Space Access (Software Wait Only)

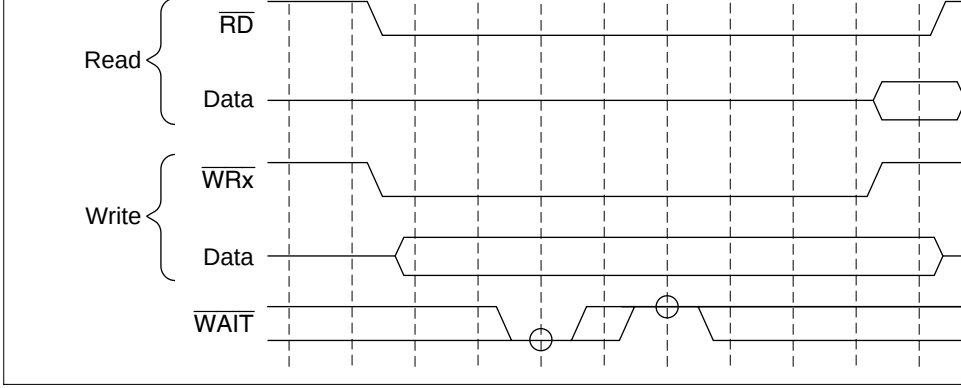


Figure 10.5 Wait State Timing of Ordinary Space Access (Wait States from Softv 2 State + $\overline{\text{WAIT}}$ Signal)

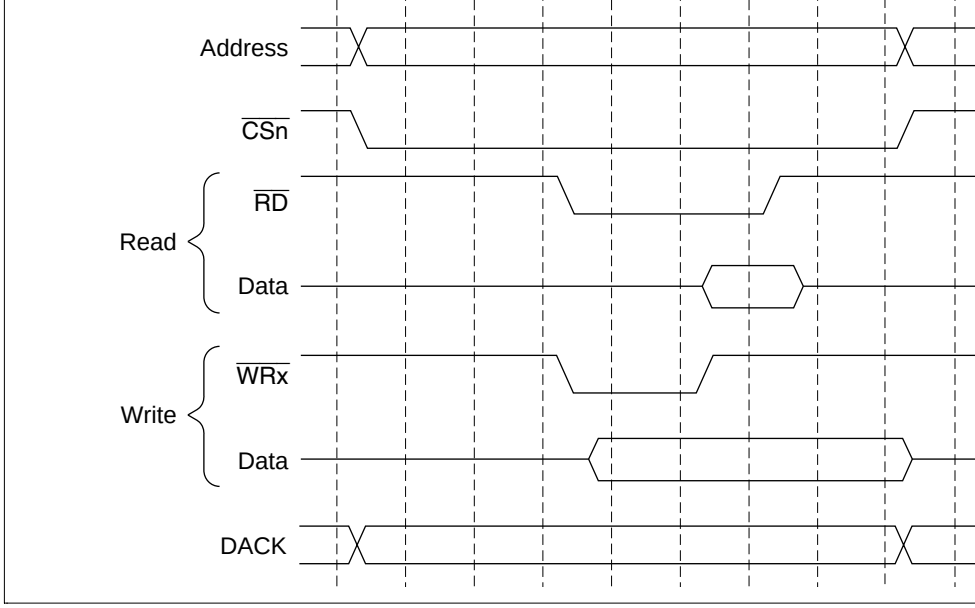


Figure 10.6 $\overline{\text{CS}}$ Assert Period Extension Function

			Row Address		Column Address	
AMX1	AMX0	Shift Amount	Output Pins	Output Address	Output Address	Ou Pi
0	0	9 bit	A21–A15	A21–A15	A21–A0	A2
			A14–A0	A23–A9		
0	1	10 bit	A21–A14	A21–A14	A21–A0	A2
			A13–A0	A23–A10		
1	0	11 bit	A21–A13	A21–A13	A21–A0	A2
			A12–A0	A23–A11		
1	1	12 bit	A21–A12	A21–A12	A21–A0	A2
			A11–A0	A23–A12		

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

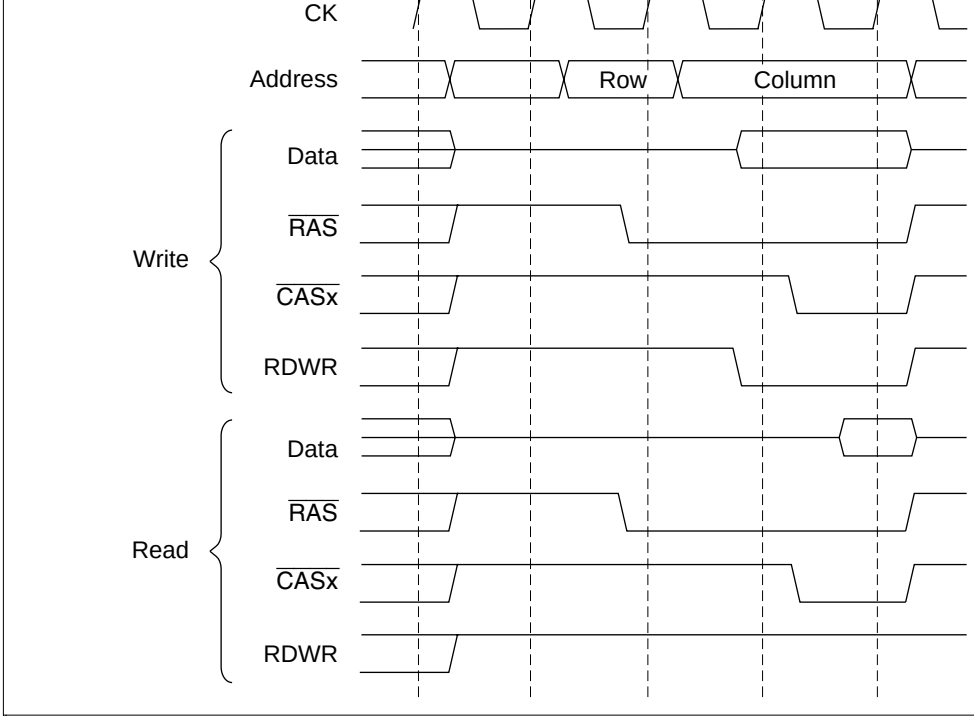


Figure 10.7 DRAM Bus Cycle (Normal Mode, TPC = 0, RCD = 0, No Waits

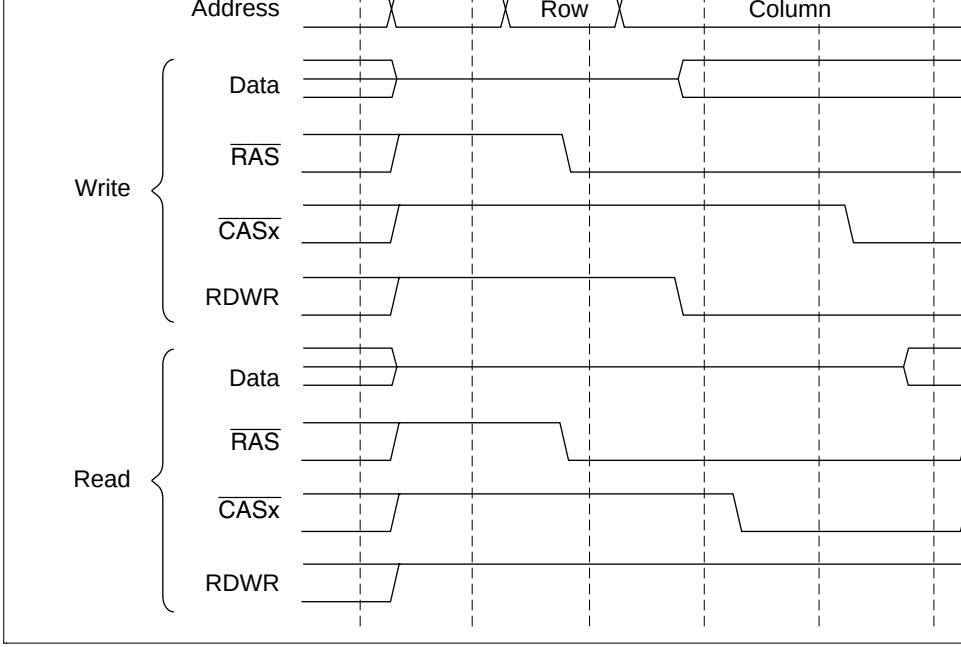


Figure 10.8 DRAM Bus Cycle (Normal Mode, TPC = 0, RCD = 0, One Wa

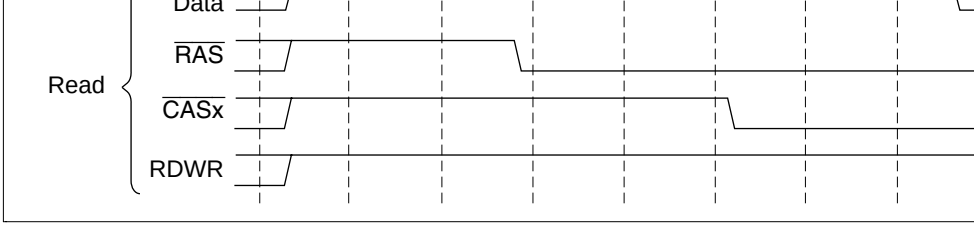


Figure 10.9 DRAM Bus Cycle (Normal Mode, TPC = 1, RCD = 1, Two Wait

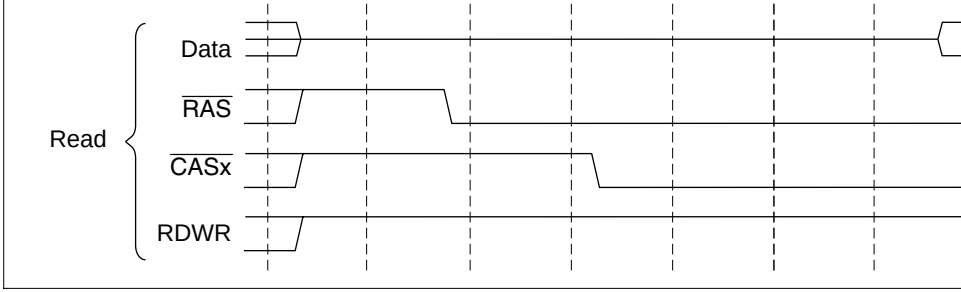


Figure 10.10 DRAM Bus Cycle (Normal Mode, TPC = 0, RCD = 0, Three W

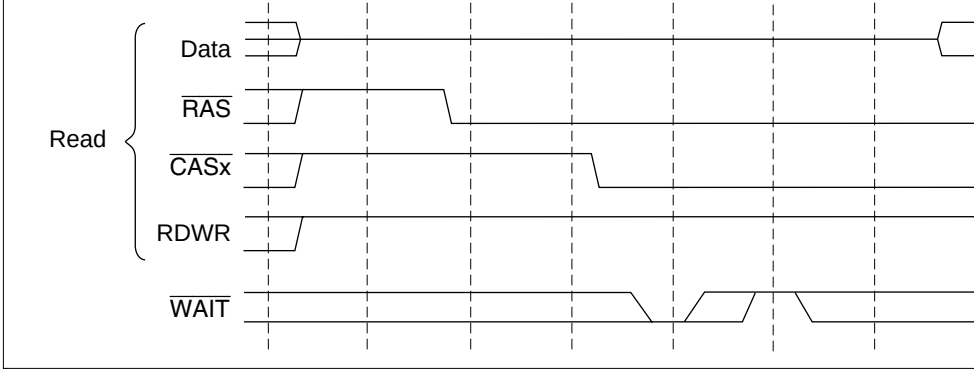


Figure 10.11 DRAM Bus Cycle (Normal Mode, TPC = 0, RCD= 0, Two Waits + W to $\overline{\text{WAIT}}$ Signal)

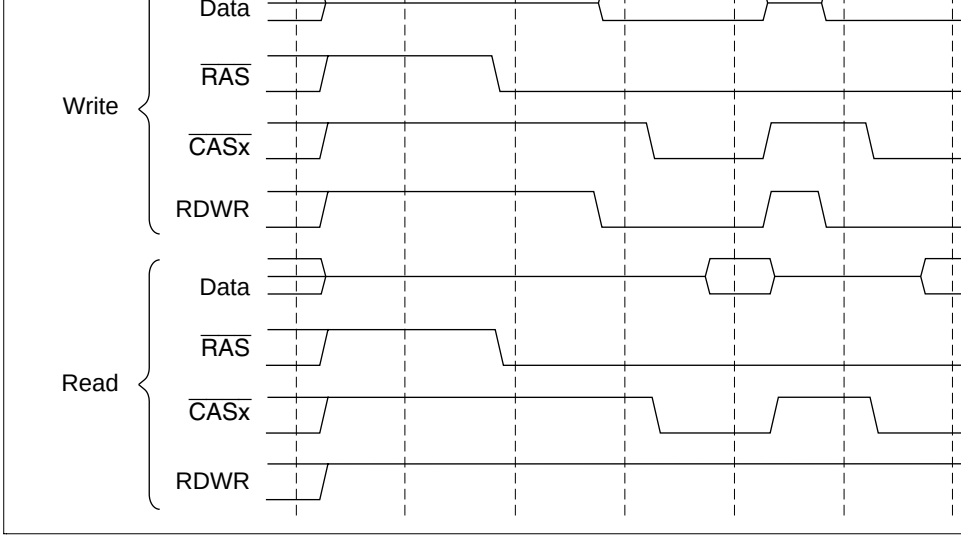


Figure 10.12 DRAM Bus Cycle (High-Speed Page Mode)

RAS Down Mode: There are some instances where even if burst operation is selected, c accesses to DRAM will not occur, but another space will be accessed instead part way th access. In such cases, if the $\overline{\text{RAS}}$ signal is maintained at low level during the time the ot is accessed, it is possible to continue burst operation at the time the next DRAM same ro 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.

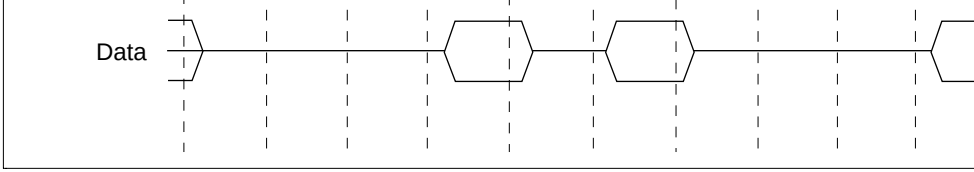


Figure 10.13 DRAM Access Normal Operation (RAS Up Mode)

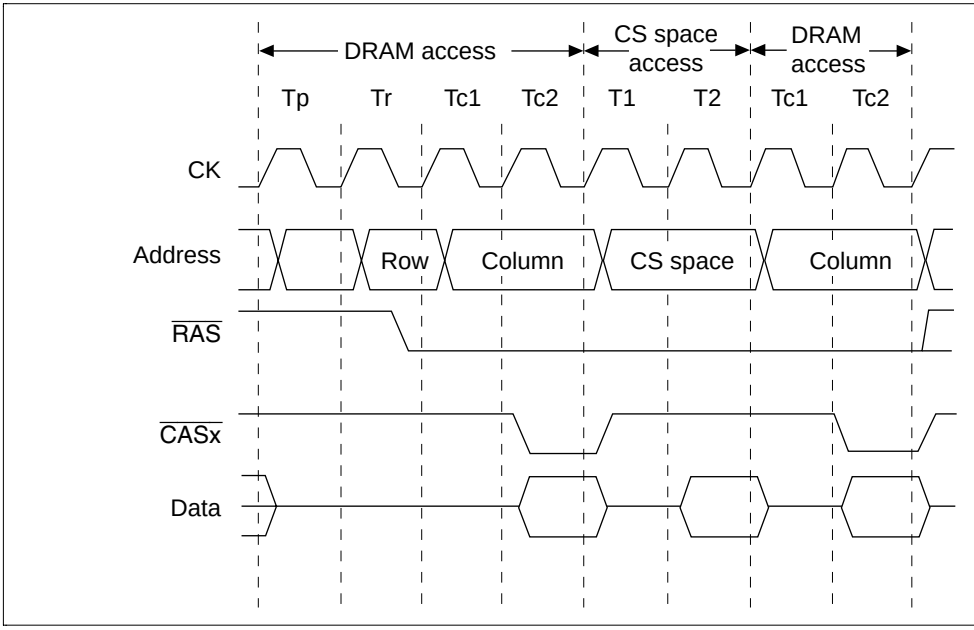


Figure 10.14 RAS Down Mode

DCR.

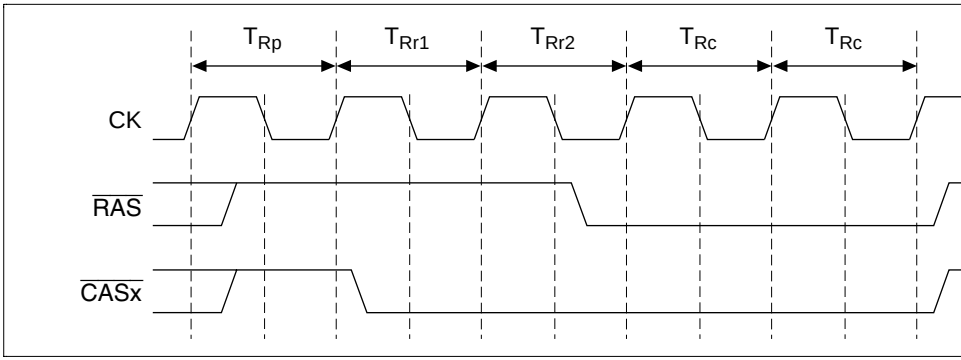


Figure 10.15 CAS-Before-RAS Refresh Timing (TRAS1, TRAS0 = 0, 0)

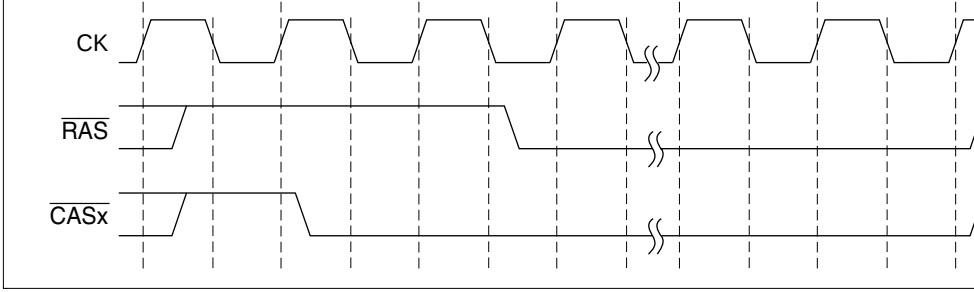


Figure 10.16 Self-Refresh Timing

the bus size becomes 8 bit and addresses and data are input and output through the D7–D
When the A14 address bit is 1, the bus size becomes 16 bit and address output and data
through the D15–D0 pins. Access for the address/data multiplex I/O space is controlled
$\overline{AH}$, $\overline{RD}$, and $\overline{WRx}$ signals.

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

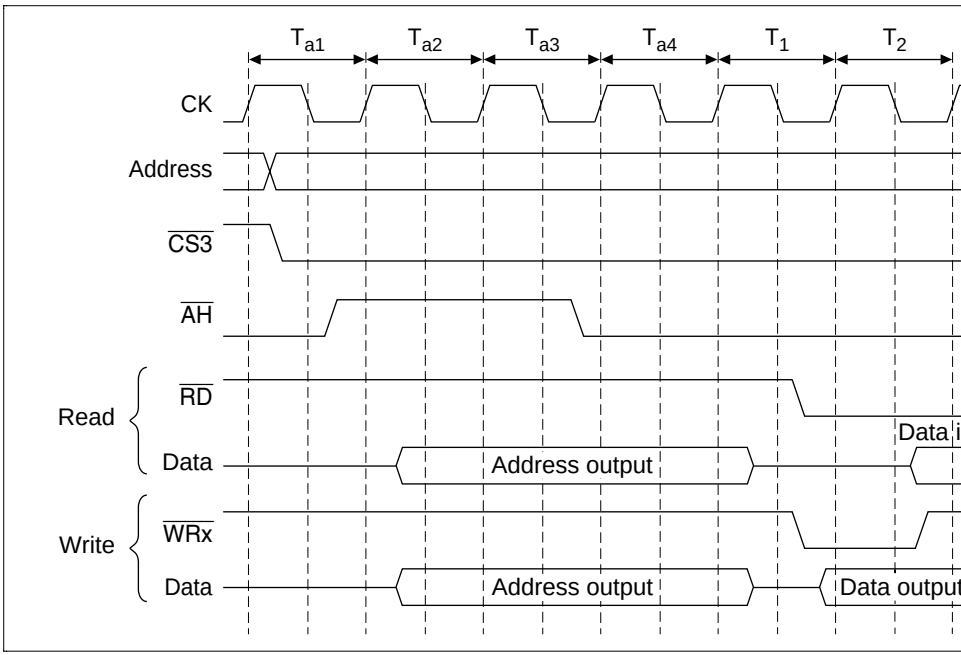


Figure 10.17 Address/Data Multiplex I/O Space Access Timing (No Wait

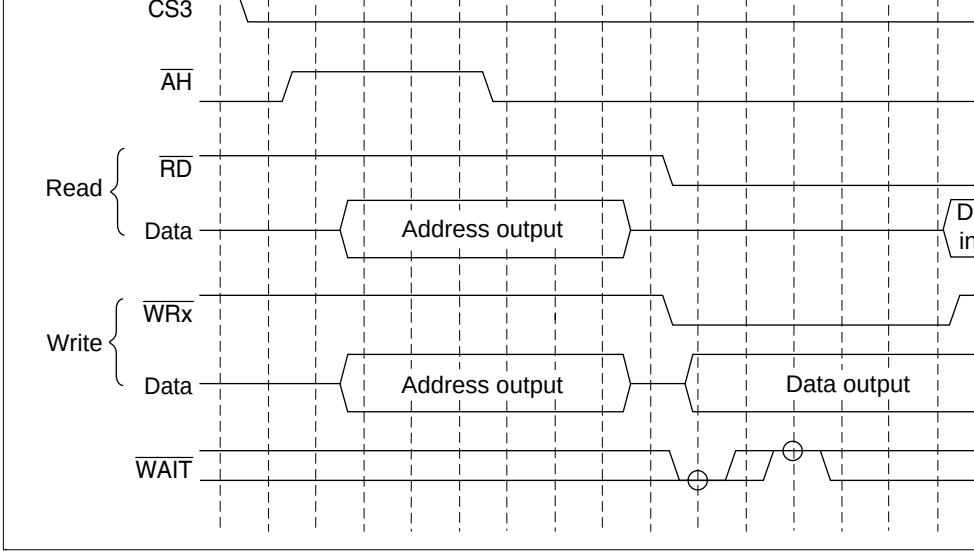


Figure 10.18 Address/Data Multiplex I/O Space Access Wait State Timing (One S Wait + One External Wait)

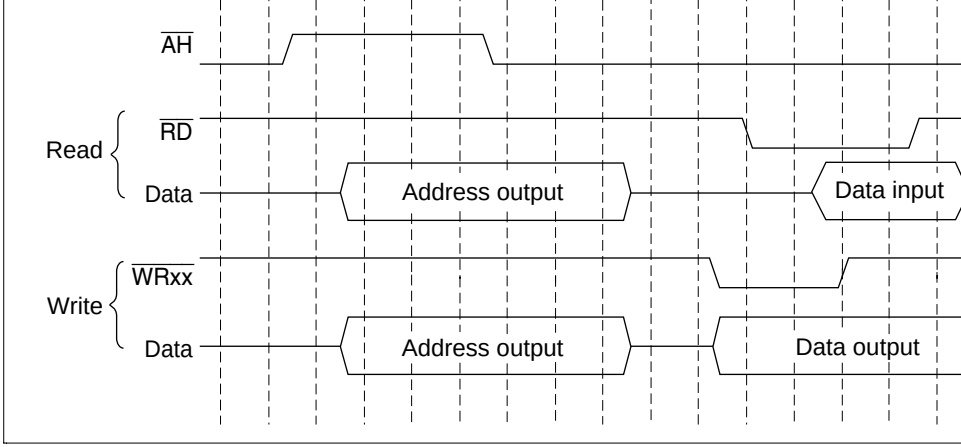


Figure 10.19 Wait Timing in Address/Data Multiplex I/O Space when CS Ass Extension is Set

10.6 Waits between Access Cycles

When a read from a slow device is completed, data buffers may not go off in time to pre conflicts with the next access. If there is a data conflict during memory access, the probl solved by inserting a wait in the access cycle.

To enable detection of bus cycle starts, waits can be inserted between access cycles duri 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 af cycles, waits are inserted so that the number of idle cycles specified by the IW31–IW00

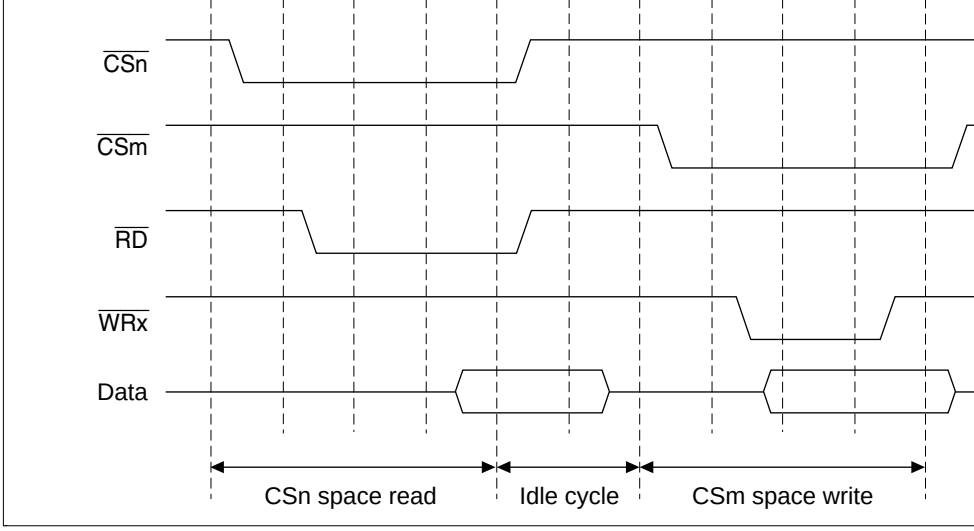


Figure 10.20 Idle Cycle Insertion Example

IW31 and IW30 specify the number of idle cycles required after a CS3 space read either t other external spaces, or for this LSI, to do write accesses. In the same manner, IW21 and specify the number of idle cycles after a CS2 space read, IW11 and IW10, the number aft 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 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.

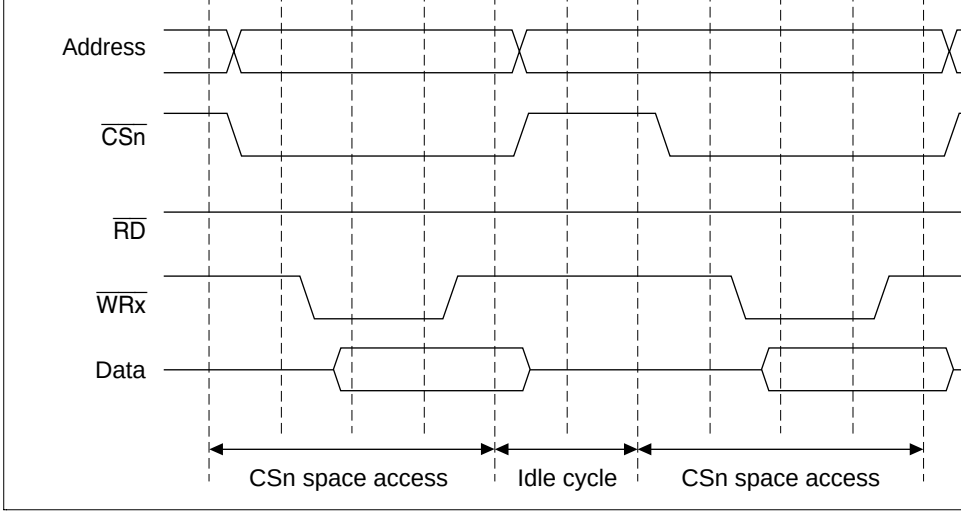


Figure 10.21 Same Space Consecutive Access Idle Cycle Insertion Examp

10.7 Bus Arbitration

The SH7040 series has a bus arbitration function that, when a bus release request is rece an external device, releases the bus to that device. It also has two internal bus masters, t and the DMAC, DTC. The priority ranking for determining bus right transfer between t 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 indire transfer mode operation, the DMAC continues operating even if a DTC request is receiv

Through port register settings, $\overline{\text{IRQOUT}}$ is asserted to indicate that a CAS-before-RAS request for DRAM has been generated during release of bus rights to an external device.

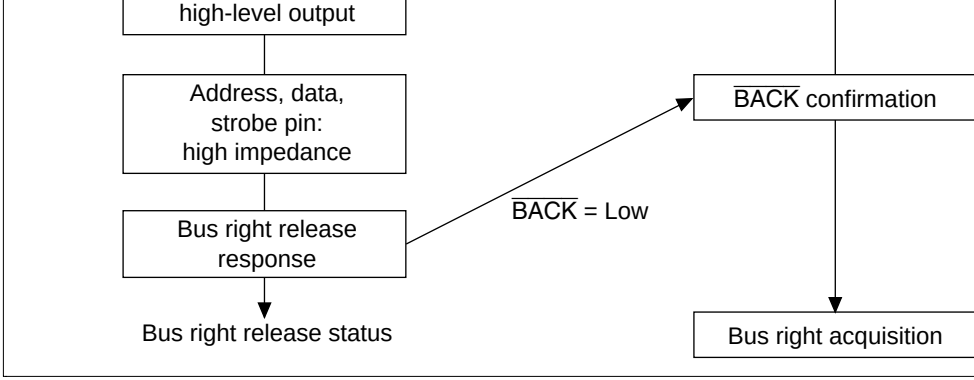


Figure 10.22 Bus Right Release Procedure

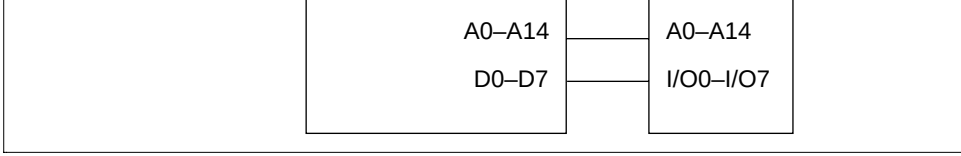


Figure 10.23 8-Bit Data Bus Width ROM Connection

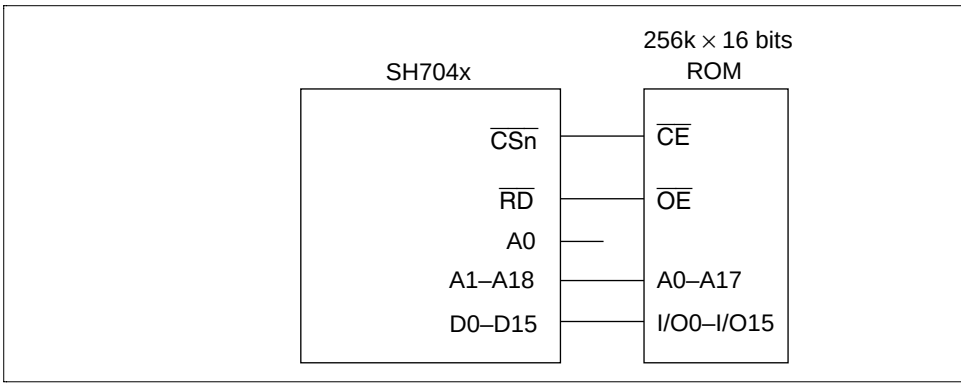


Figure 10.24 16-Bit Data Bus Width ROM Connection

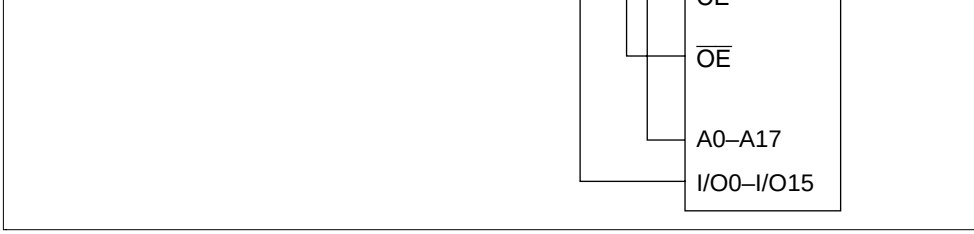


Figure 10.25 32-Bit Data Bus Width ROM Connection

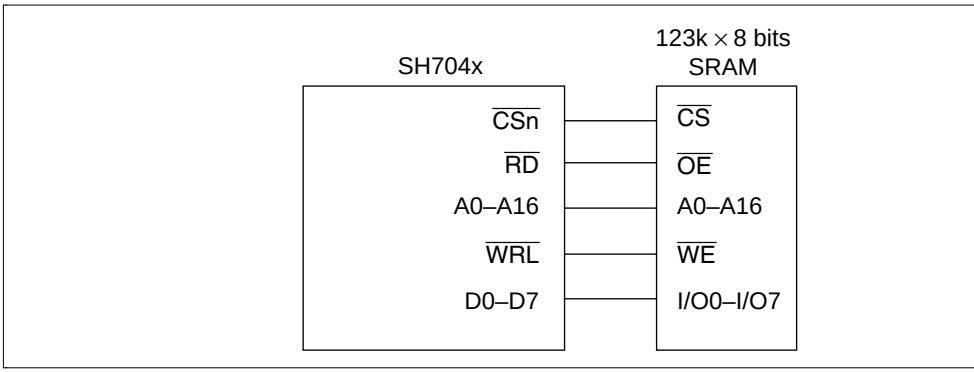


Figure 10.26 8-Bit Data Bus Width SRAM Connection

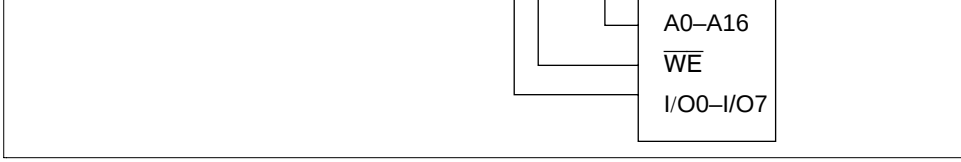


Figure 10.27 16-Bit Data Bus Width SRAM Connection

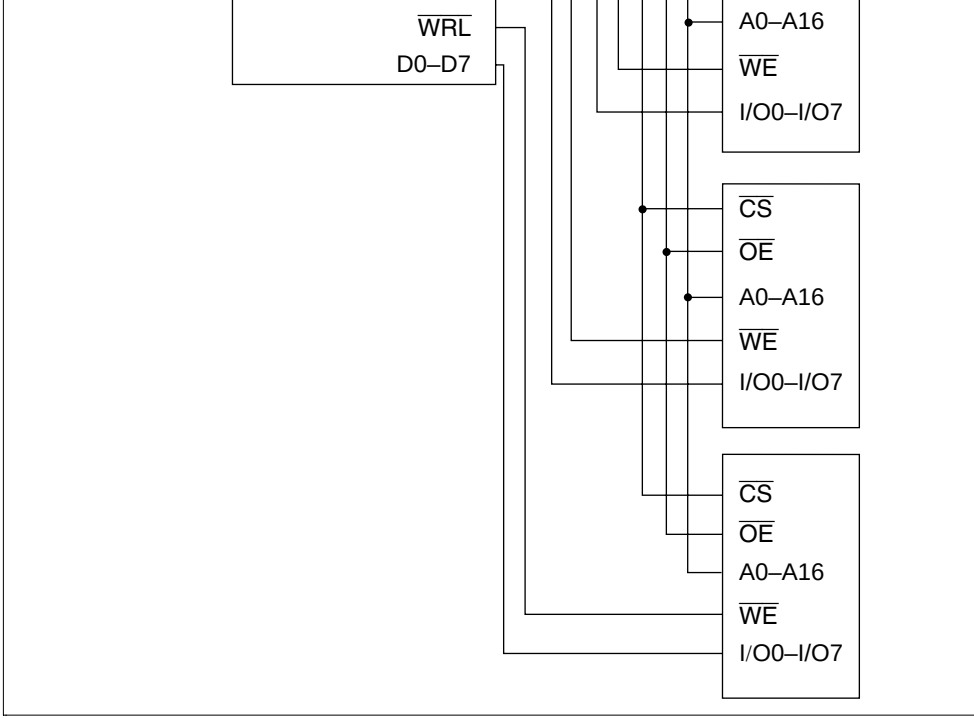


Figure 10.28 32-Bit Data Bus Width SRAM Connection

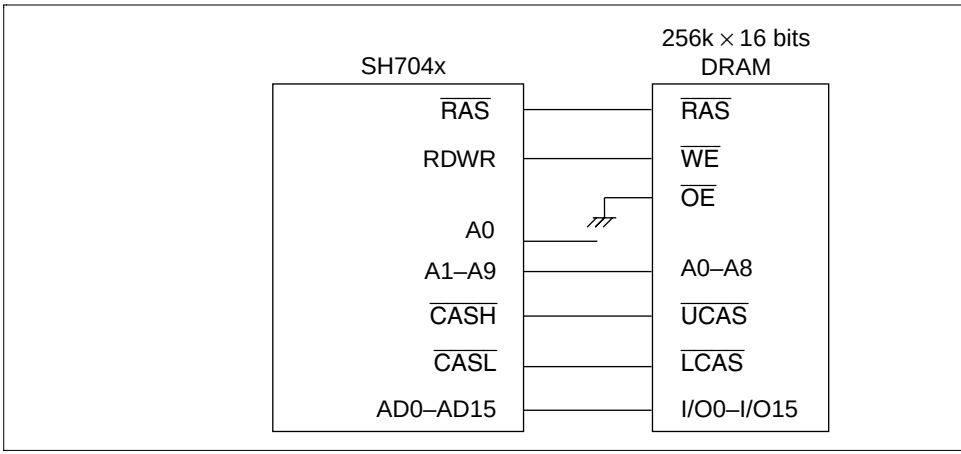


Figure 10.30 16-Bit Data Bus Width DRAM Connection

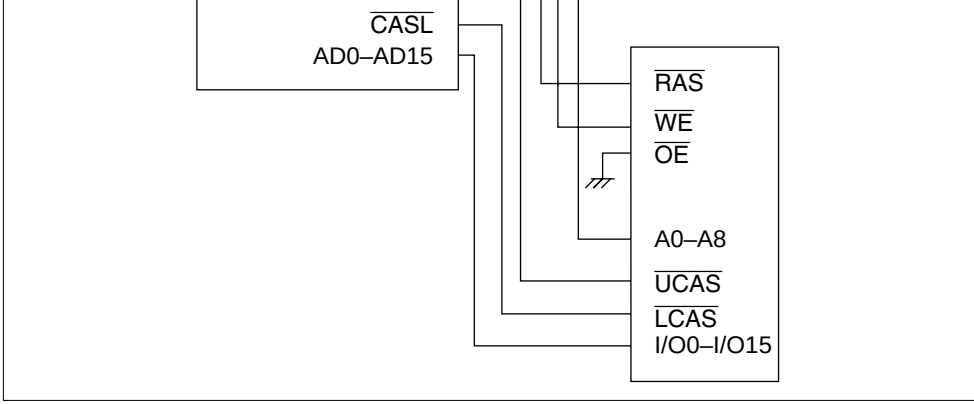


Figure 10.31 32-Bit Data Bus Width DRAM Connection

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 ta

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
Connected bus width	8bit	16bit	16bit	16bit	16bit	16bit	16bit	16bit	16bit	16bit
Access cycle	2cyc	2cyc	2cyc	2cyc	2cyc	2cyc	3cyc	3cyc	3cyc	3cyc

Note: * A/D of A mask products are accessed in 8-bit width, 3 cyc.

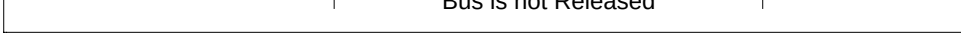


Figure 10.32 One Bus Cycle

10.10 CPU Operation when Program is in External Memory

In the SH7040 Series, two words (equivalent to two instructions) are normally fetched i
instruction fetch. This is also true when the program is located in external memory, irre
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
address, or if the program counter value immediately before the program branches is an
(2n) address, the CPU will always fetch 32 bits (equivalent to two instructions) that incl
respective word instruction.

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 tra
  - Single address mode: Either the transfer source or transfer destination (peripheral accessed by a DACK signal while the other is accessed by address. One transfer data is transferred in each bus cycle.
  - Dual address mode: Both the transfer source and transfer destination are accessed address. Dual address mode can be direct or indirect address transfer.
    - Direct access: Values set in a DMAC internal register indicate the accessed a both the transfer source and transfer destination. Two bus cycles are required data transfer.
    - Indirect access: The value stored at the location pointed to by the address set DMAC internal transfer source register is used as the address. Operation is ot the same as direct access. This function can only be set for channel 3. Four bu are required for one data transfer.
- Channel function: Transfer modes that can be set are different for each channel. (Du mode indirect access can only be set for channel 1. Only direct access is possible for 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 eve transfer.

— Fixed priority mode: Always fixed
— Round robin mode: Sets the lowest priority level for the channel that received the request last

- CPU can be interrupted when the specified number of data transfers are complete.

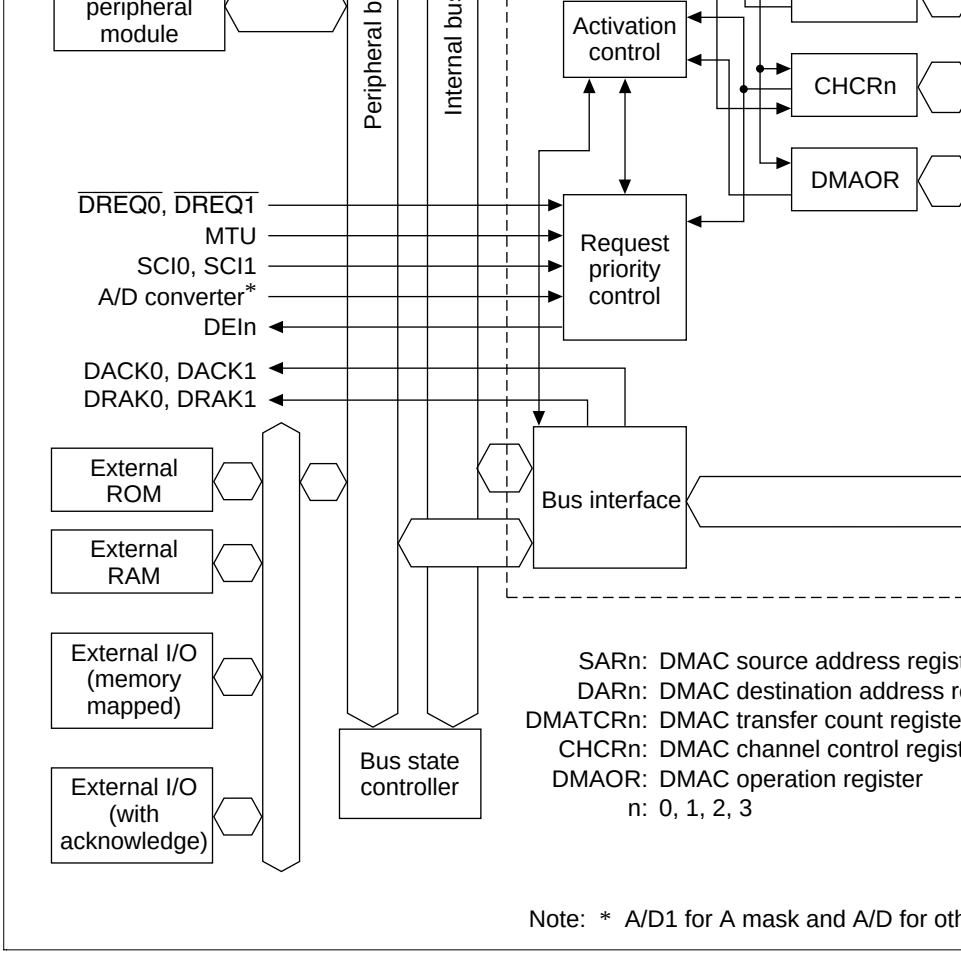


Figure 11.1 DMAC Block Diagram

				external source
1	DMA transfer request	$\overline{\text{DREQ1}}$	I	DMA transfer request input fro external device to channel 1
	DMA transfer request acknowledge	DACK1	O	DMA transfer strobe output fro channel 1 to external device
	$\overline{\text{DREQ1}}$ acceptance confirmation	DRAK1	O	Sampling receive acknowledge for DMA transfer request input external source

	DMA transfer count register 0	DMATCR0	R/W	Undefined	H'FFFF86C8	32 bit
	DMA channel control register 0	CHCR0	R/W*1	H'00000000	H'FFFF86CC	32 bit
1	DMA source address register 1	SAR1	R/W	Undefined	H'FFFF86D0	32 bit
	DMA destination address register 1	DAR1	R/W	Undefined	H'FFFF86D4	32 bit
	DMA transfer count register 1	DMATCR1	R/W	Undefined	H'FFFF86D8	32 bit
	DMA channel control register 1	CHCR1	R/W*1	H'00000000	H'FFFF86DC	32 bit
2	DMA source address register 2	SAR2	R/W	Undefined	H'FFFF86E0	32 bit
	DMA destination address register 2	DAR2	R/W	Undefined	H'FFFF86E4	32 bit

	register 3						
	DMA channel control register 3	CHCR3	R/W*1	H'00000000	H'FFFF86FC	32 bit	1
Shared	DMA operation register	DMAOR	R/W*1	H'0000	H'FFFF86B0	16 bit	1

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 DM clear flags. No other writes are allowed.

*2 For 16-bit access of SAR0–SAR3, DAR0–DAR3, and CHCR0–CHCR3, the 16 on the side not accessed is held.

*3 DMATCR has a 24-bit configuration: bits 0–23. Writing to the upper 8 bits (bits 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 spec source address of a DMA transfer. These registers have a count function, and during a DM transfer, they indicate the next source address. In single-address mode, SAR values are ig 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. Op cannot be guaranteed on any other addresses.

The initial value after power-on resets or in software standby mode is undefined. These re are not initialized with manual reset.

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 th
the destination address of a DMA transfer. These registers have a count function, and du
DMA transfer, they indicate the next destination address. In single-address mode, DAR
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. C
cannot be guaranteed on any other address. The initial value after power-on resets or in s
standby mode, is undefined. These registers are not initialized with manual reset.

Bit:	31	30	29	28	27	26	25
Initial value:	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	23	22	21	…	…	2	1
				…	…		
Initial value:	—	—	—	…	…	—	—
R/W:	R/W	R/W	R/W	…	…	R/W	R/W

Initial value:	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R

Bit:	23	22	21	20	19	18	17
Initial value:	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

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

	—	—	—	DI*2	RO*2	RL*2	AM*2
Initial value:	—	—	—	0	0	0	0
	R	R	R	(R/W)	(R/W)	(R/W)	(R/W)

Bit:	15	14	13	12	11	10	9
	DM1	DM0	SM1	SM0	RS3	RS2	RS1
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	—	DS*2	TM	TS1	TS0	IE	TE
Initial value:	—	0	0	0	0	0	0
R/W:	R	(R/W)	R/W	R/W	R/W	R/W	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 indire mode operation for channel 3 source address. This bit is valid only in CHCR3. It alw 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 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 D the data write cycle or data read cycle. In single address mode, DACK is always outpu irrespective of the setting of this bit. This bit is valid only for CHCR0 and CHCR1. It 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) sig output to active high or active low. This bit is valid only with CHCR0 and CHCR1. It 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 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 ar when transferring data from an external device to address space in single address mo

Bit 13: SM1	Bit 12: SM0	Description
0	0	Source address fixed (initial value)
0	1	Source address incremented (+1 during 8-bit transfe during 16-bit transfer, +4 during 32-bit transfer)
1	0	Source address decremented (–1 during 8-bit transfe 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 reg (SAR3) the actual storage address of the data you want to transfer as the data storage ad (indirect address).

During indirect address mode, SAR3 obeys the SM1/SM0 setting for increment/decreme case, SAR3's increment/decrement is fixed at +4/–4 or 0, irrespective of the transfer dat specified by TS1 and TS0.

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 requ
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 re
mode to either low-level detection or falling-edge detection. This bit is valid only witl
and CHCR1. For CHCR2 and CHCR3, this bit always reads as 0 and cannot be modif

  Even with channels 0 and 1, when specifying an on-chip peripheral module or auto-re
the transfer request source, this bit setting is ignored. The sampling method is fixed at
edge detection in cases other than auto-request.

Bit 6: DS	Description
0	Low-level detection (initial value)
1	Falling-edge detection

1	1	Prohibited

- Bit 2—Interrupt Enable (IE): When this bit is set to 1, interrupt requests are generate number of data transfers specified in the DMATCR (when TE = 1).

Bit 2: IE	Description
0	Interrupt request not generated after DMATCR-specified trans (initial value)
1	Interrupt request enabled on completion of DMATCR specifie of transfers

- Bit 1—Transfer End Flag (TE): This bit is set to 1 after the number of data transfers by the DMATCR. At this time, if the IE bit is set to 1, an interrupt request is generat

  If data transfer ends before TE is set to 1 (for example, due to an NMI or address err clearing of the DE bit or DME bit of the DMAOR) the TE is not set to 1. With this b 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, sta mode
1	DMATCR specified number of transfers completed

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. M
reset does not initialize DMAOR.

Bit:	15	14	13	12	11	10	9
	—	—	—	—	—	—	PR1
Initial value:	—	—	—	—	—	—	0
R/W:	R	R	R	R	R	R	R/W

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

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.

transfer. If this bit is set during a data transfer, transfers on all channels are suspende
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 w
DMAC is operating or suspended. If this bit is set during a data transfer, transfers on
channels are suspended. The CPU is unable to write a 1 to the NMIF. Clearing is eff
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

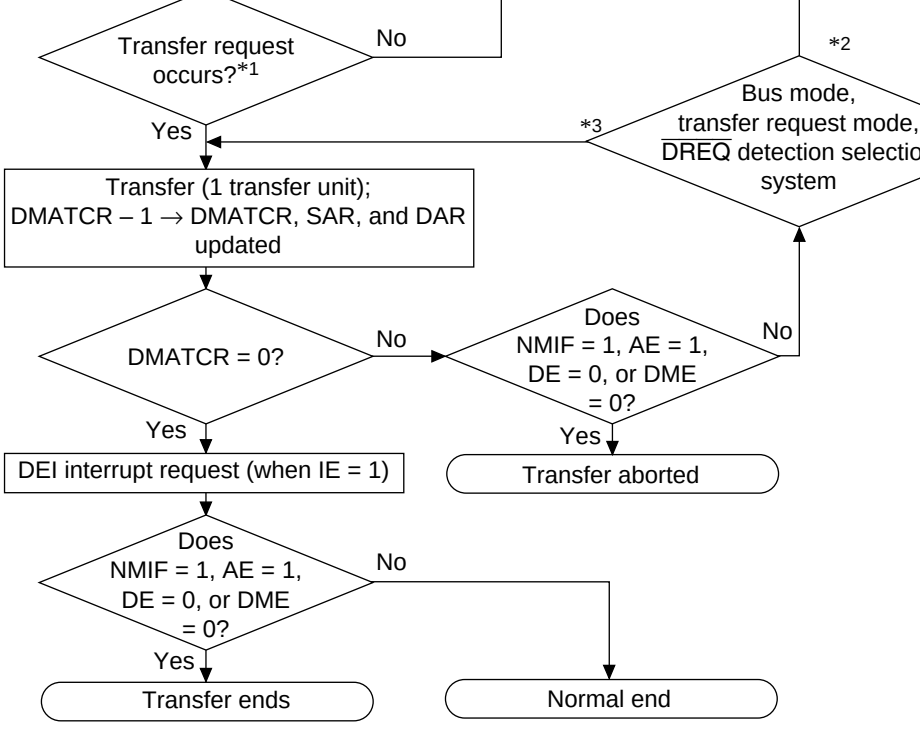
11.3 Operation

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 en transfer. Transfers can be requested in three modes: auto-request, external request, and or peripheral module request. Transfer can be in either the single address mode or the dual a mode, and dual address mode can be either direct or indirect address transfer mode. The b 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 transfer count register (DMATCR), DMA channel control registers (CHCR), and DMA o register (DMAOR) are set to the desired transfer conditions, the DMAC transfers data acc 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 t unit of data (determined by TS0 and TS1 setting). For an auto-request, the transfer beg 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 mode.
3. When the specified number of transfers have been completed (when DMATCR reache transfer ends normally. If the IE bit of the CHCR is set to 1 at this time, a DEI interru to the CPU.
4. When an address error occurs in the DMAC or an NMI interrupt is generated, the tran aborted. Transfers are also aborted when the DE bit of the CHCR or the DME bit of th DMAOR are changed to 0.



Notes: *1 In auto-request mode, transfer begins when NMIF, AE, and TE are and the DE and DME bits are set to 1.

*2 $\overline{\text{DREQ}}$ = level detection in burst mode (external request), or cycle-s mode.

*3 $\overline{\text{DREQ}}$ = edge detection in burst mode (external request), or auto-re mode in burst mode.

Figure 11.2 DMAC Transfer Flowchart

NMIF and AE bits of DMAOR are all 0).

External Request Mode: In this mode a transfer is performed at the request signal ($\overline{\text{DRE}}$ external device. Choose one of the modes shown in table 11.3 according to the applicatio When this mode is selected, if the DMA transfer is enabled (DE = 1, DME = 1, TE = 0, N AE = 0), a transfer is performed upon a request at the $\overline{\text{DREQ}}$ input. Choose to detect $\overline{\text{DRE}}$ either the falling edge or low level of the signal input with the DS bit of CHCR0–CHCR3 is level detection, DS = 1 is edge detection). The source of the transfer request does not h 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 devi DACK
0	0	1	1	Single address mode	External device with DACK	External mer memory-map external devi

Note: * External memory, memory-mapped external device, on-chip memory, on-chip pe module (excluding DMAC, DTC, BSC, UBC).

On-Chip Peripheral Module Request Mode: In this mode a transfer is performed at the request signal (interrupt request signal) of an on-chip peripheral module. As indicated in t 11.4, there are ten transfer request signals: five from the multifunction timer pulse unit (M which are compare match or input capture interrupts; the receive data full interrupts (RxI transmit data empty interrupts (TxI) of the two serial communication interfaces (SCI); an conversion end interrupt (ADI1 for A mask, ADI for others) of the A/D converter. When

0	1	1	0	MTU*2	TGI0A	Any*1	Any*1	Burst/c
0	1	1	1	MTU*2	TGI1A	Any*1	Any*1	Burst/c
1	0	0	0	MTU*2	TGI2A	Any*1	Any*1	Burst/c
1	0	0	1	MTU*2	TGI3A	Any*1	Any*1	Burst/c
1	0	1	0	MTU*2	TGI4A	Any*1	Any*1	Burst/c
1	0	1	1	A/D	ADI*5	ADDR*4	Any*1	Burst/c
1	1	0	0	SCI0*3 transmit block	TxI0	Any*1	TDR0	Burst/c
1	1	0	1	SCI0*3 transmit block	RxI0	RDR0	Any*1	Burst/c
1	1	1	0	SCI1*3 transmit block	TxI1	Any*1	TDR1	Burst/c
1	1	1	1	SCI1*3 transmit block	RxI1	RDR1	Any*1	Burst/c

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 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 signal, interrupts for the CPU are not generated.

When a DMA transfer is conducted corresponding with one of the transfer request signa 11.4, it is automatically discontinued. In cycle steal mode this occurs in the first transfer burst mode with the last transfer.

- CH2 > CH0 > CH1 > CH3

These are selected by settings of the PR1 and PR0 bits of the DMA operation register (DM

Round Robin Mode: In round robin mode, each time the transfer of one transfer unit (by or long word) ends on a given channel, that channel receives the lowest priority level (fig The priority level in round robin mode immediately after a reset is CH0 > CH1 > CH2 > C

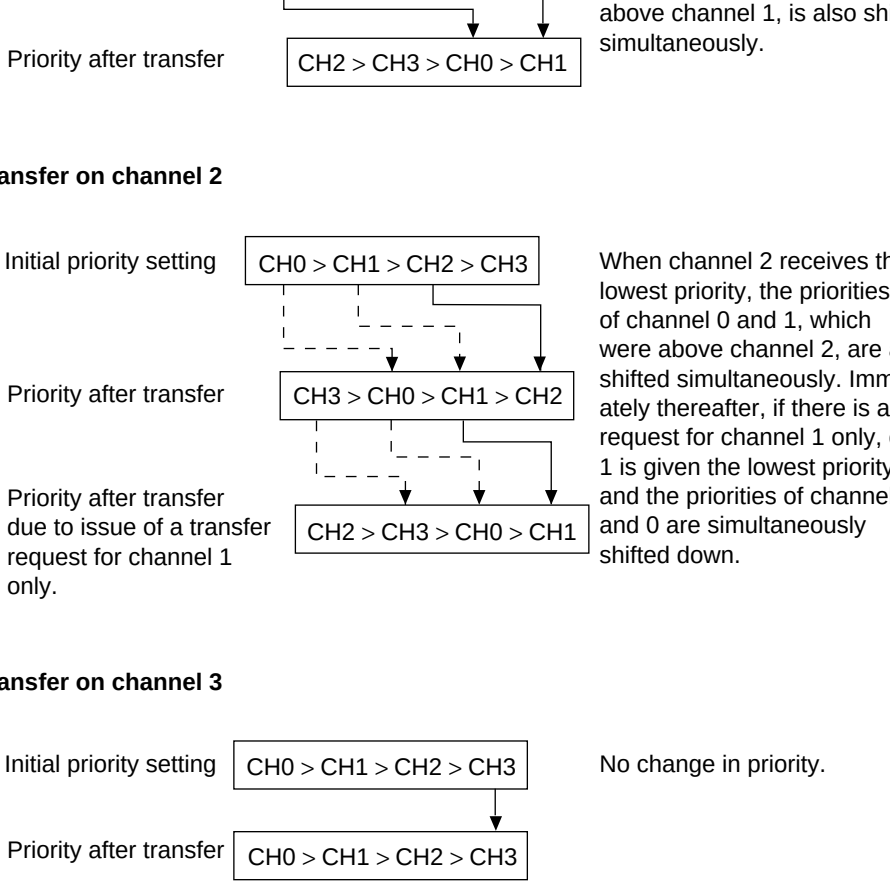


Figure 11.3 Round Robin Mode

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 channel 3 the lowest priority.

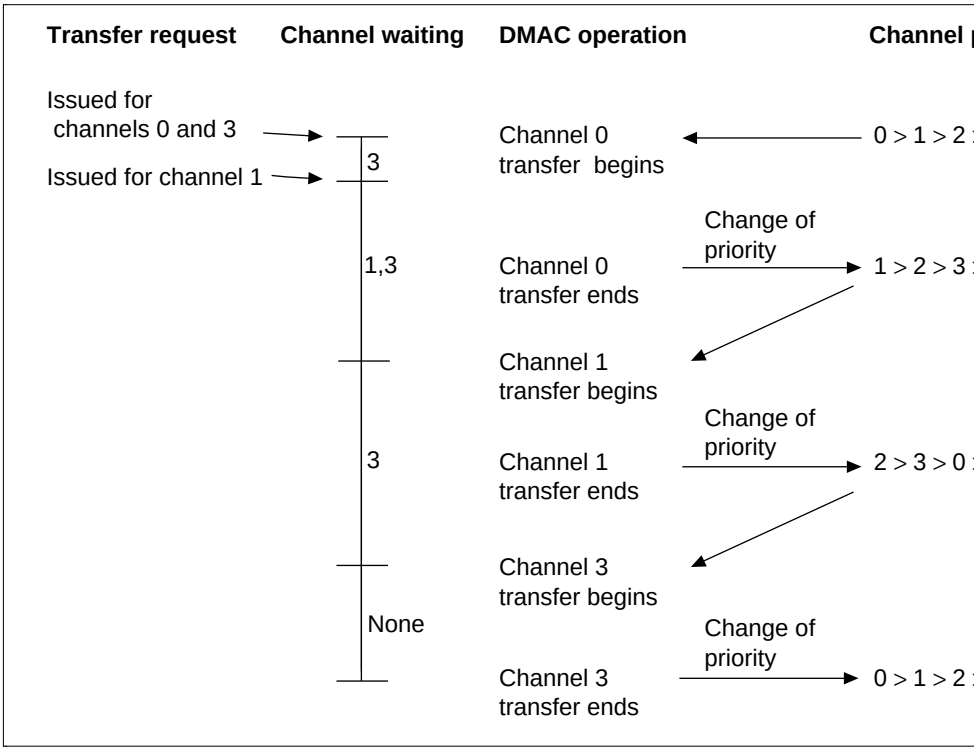


Figure 11.4 Example of Changes in Priority in Round Robin Mode

	Destination				
Source	**External Device with DACK**	**External Memory**	**Memory-Mapped External Device**	**On-Chip Memory**	**On-C Perip Modu**
External device with DACK	Not available	Single	Single	Not available	Not a
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 ad mode.

11.3.5 Address Modes

Single Address Mode: In the single address mode, both the transfer source and destinat external; one (selectable) is accessed by a DACK signal while the other is accessed by a In this mode, the DMAC performs the DMA transfer in 1 bus cycle by simultaneously o transfer request acknowledge DACK signal to one external device to access it while out address to the other end of the transfer. Figure 11.5 shows an example of a transfer betw external memory and an external device with DACK in which the external device outpu the data bus while that data is written in external memory in the same bus cycle.

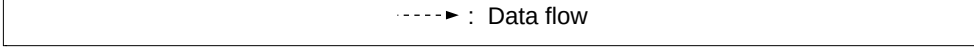
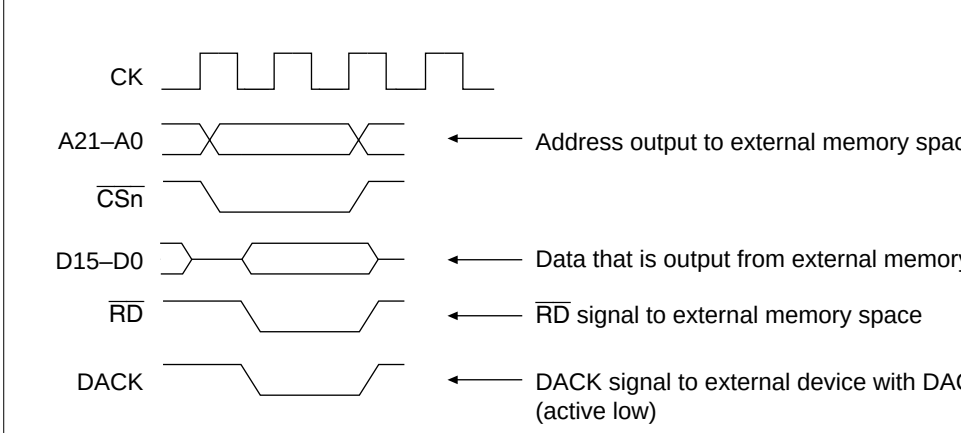


Figure 11.5 Data Flow in Single Address Mode

Two types of transfers are possible in the single address mode: (a) transfers between exte devices with DACK and memory-mapped external devices, and (b) transfers between ext devices with DACK and external memory. The only transfer requests for either of these i external request ($\overline{\text{DREQ}}$). Figure 11.6 shows the DMA transfer timing for the single addr



b. External memory space to external device with DACK

Figure 11.6 Example of DMA Transfer Timing in the Single Address Mo

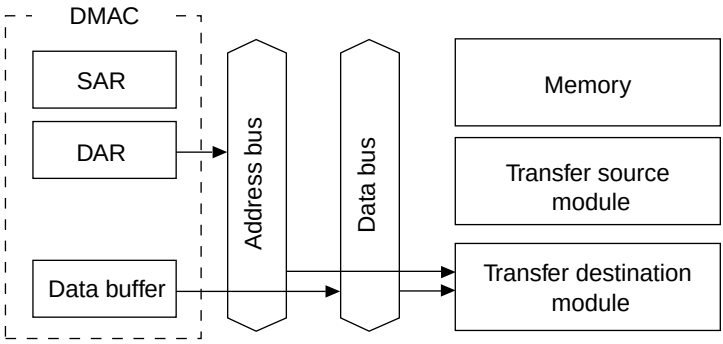
11.3.6 Dual Address Mode

Dual address mode is used for access of both the transfer source and destination by addr
Transfer source and destination can be accessed either internally or externally. Dual add
is subdivided into two other modes: direct address transfer mode and indirect address tra
mode.

Direct Address Transfer Mode: Data is read from the transfer source during the data r
and written to the transfer destination during the write cycle, so transfer is conducted in
cycles. At this time, the transfer data is temporarily stored in the DMAC. With the kind
memory transfer shown in figure 11.7, data is read from one of the memories by the DM
during a read cycle, then written to the other external memory during the subsequent wr
Figure 11.8 shows the timing for this operation.

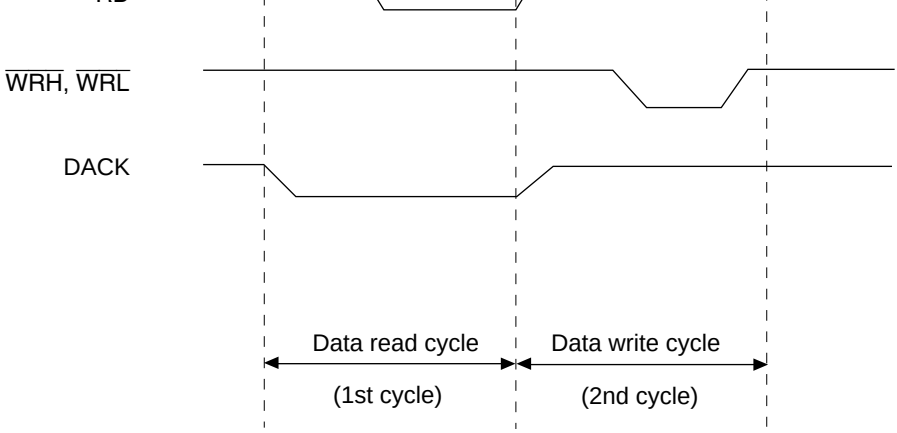
The SAR value is taken as the address, and data is read from the transfer sou
module and stored temporarily in the DMAC.

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

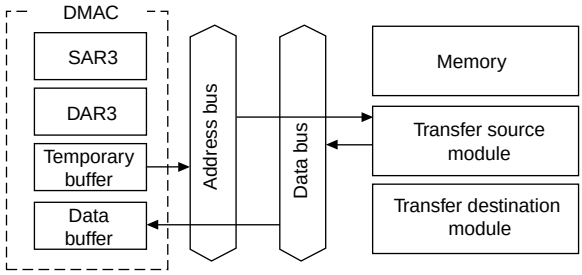


Note: Transfer between external memories with DACK are output during read cycle.

Figure 11.8 Example of Direct Address Transfer Timing in Dual Address M

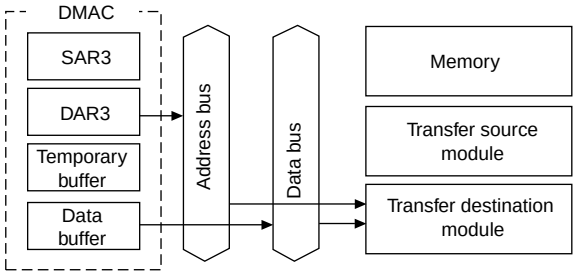
indirect address is output to the address bus. When transfer data is 32-bit, the third and fo
cycles each need to be doubled, giving a required total of six bus cycles and one NOP cyc
whole operation.

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.

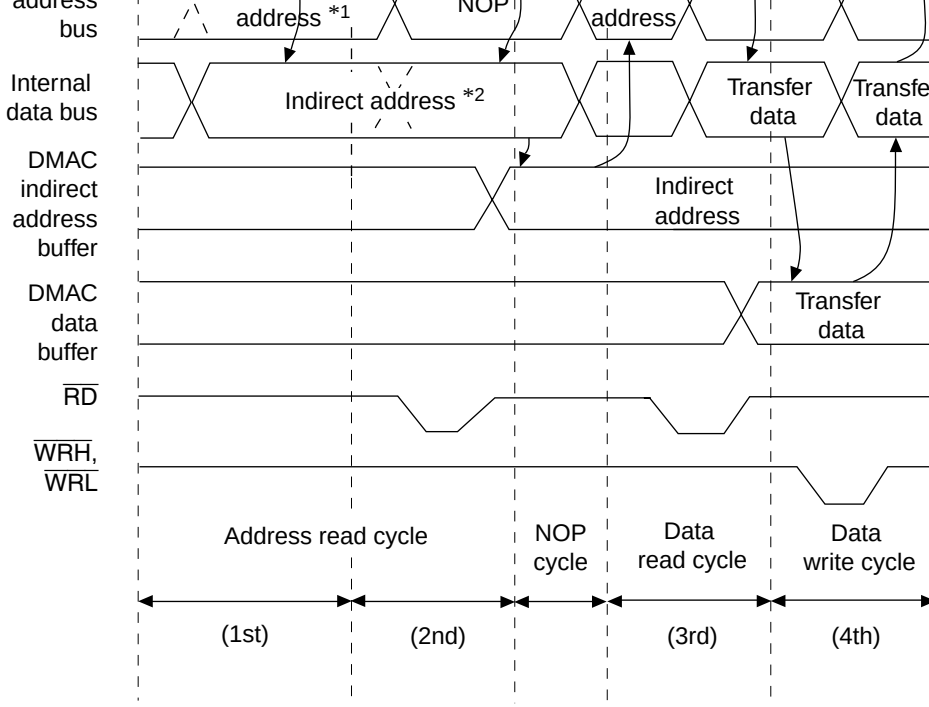
4th bus cycle



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 dat bus.

Figure 11.10 Dual Address Mode and Indirect Address Transfer Timing Exam (External Memory Space to External Memory Space)

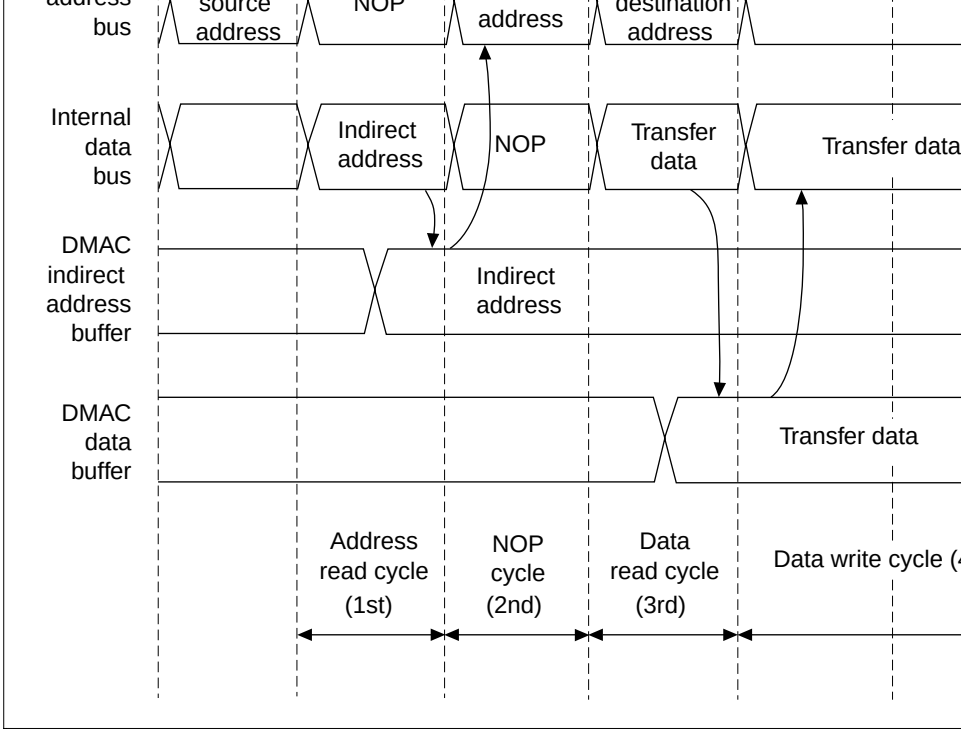


Figure 11.11 Dual Address Mode and Indirect Address Transfer Timing Exa
(On-chip Memory Space to On-chip Memory Space)

Transfer conditions are dual address mode and DREQ level detection.

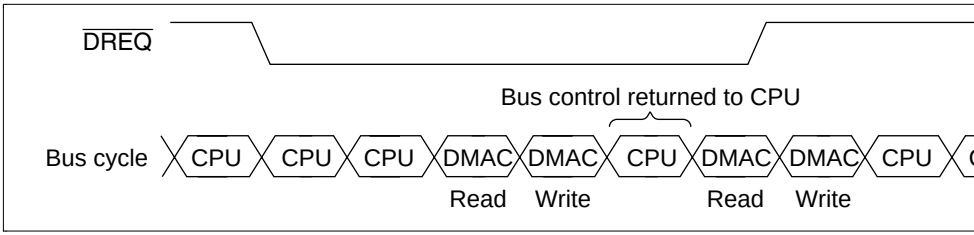


Figure 11.12 DMA Transfer Example in the Cycle-Steal Mode

Burst Mode: Once the bus right is obtained, the transfer is performed continuously until transfer end condition is satisfied. In the external request mode with low level detection o DREQ pin, however, when the DREQ pin is driven high, the bus passes to the other bus r after the bus cycle of the DMAC that currently has an acknowledged request ends, even i transfer end conditions have not been satisfied.

Figure 11.13 shows an example of DMA transfer timing in the burst mode. Transfer cond single address mode and DREQ level detection.

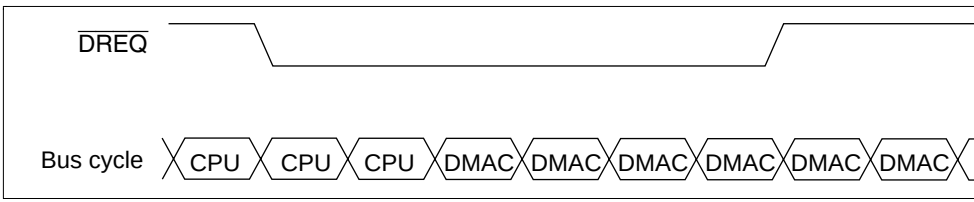


Figure 11.13 DMA Transfer Example in the Burst Mode

External memory and memory-mapped external device	Any*1	B/C	8/16/32	0
Memory-mapped external device and memory-mapped external device	Any*1	B/C	8/16/32	0
External memory and on-chip memory	Any*1	B/C	8/16/32	0
External memory and on-chip peripheral module	Any*2	B/C*3	8/16/32*4	0
Memory-mapped external device and on-chip memory	Any*1	B/C	8/16/32	0
Memory-mapped external device and on-chip peripheral module	Any*2	B/C*3	8/16/32*4	0
On-chip memory and on-chip memory	Any*1	B/C	8/16/32	0
On-chip memory and on-chip peripheral module	Any*2	B/C*3	8/16/32*4	0
On-chip peripheral module and on-chip peripheral module	Any*2	B/C*3	8/16/32*4	0

Notes: *1 External request, auto-request or on-chip peripheral module request enabled
in the case of on-chip peripheral module request, it is not possible to specify t
A/D converter for the transfer request source.

*2 External request, auto-request or on-chip peripheral module request possible
if transfer request source is also the SCI or A/D converter (A/D1 for A mask),
transfer source or transfer destination must be the SCI or A/D converter (A/D
mask). For A mask, setting A/D0 as the transfer request source is not permitt

*3 When the transfer request source is the SCI, only cycle steal mode is possibl

*4 Access size permitted by register of on-chip peripheral module that is the tran
source or transfer destination.

*5 When the transfer request is an external request, channels 0 and 1 only can

*6 B: Burst, C: Cycle steal

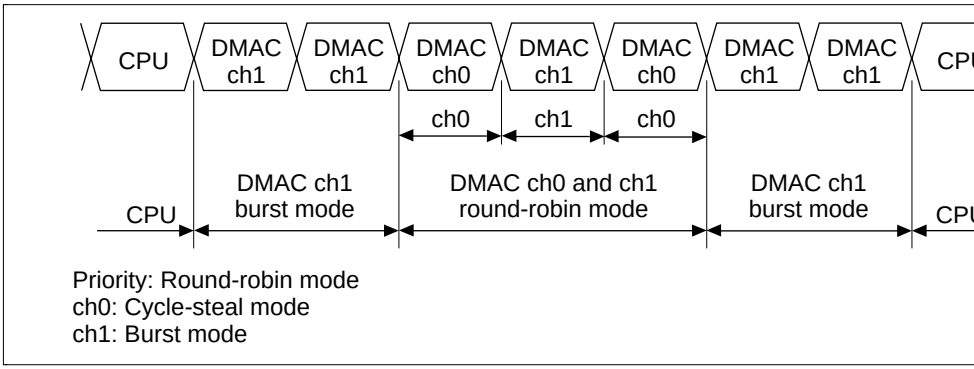


Figure 11.14 Bus Handling when Multiple Channels Are Operating

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 bus master is controlled by the bus state controller (BSC) just as it is when the CPU is the master. For details, see section 10, Bus State Controller (BSC).

$\overline{\text{DREQ}}$ Pin Sampling Timing and DRAK Signal: In external request mode, the $\overline{\text{DREQ}}$ sampled by either falling edge or low-level detection. When a $\overline{\text{DREQ}}$ input is detected, a bus cycle is issued and DMA transfer effected, at the earliest, after three states. However, mode when single address operation is specified, a dummy cycle is inserted for the first b 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 o 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 the DMAC transfer generated by the previous sampling.

As in figure 11.16, whatever cycle the CPU transfer cycle is, the next sampling begins f
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 examp
output during DACK write.

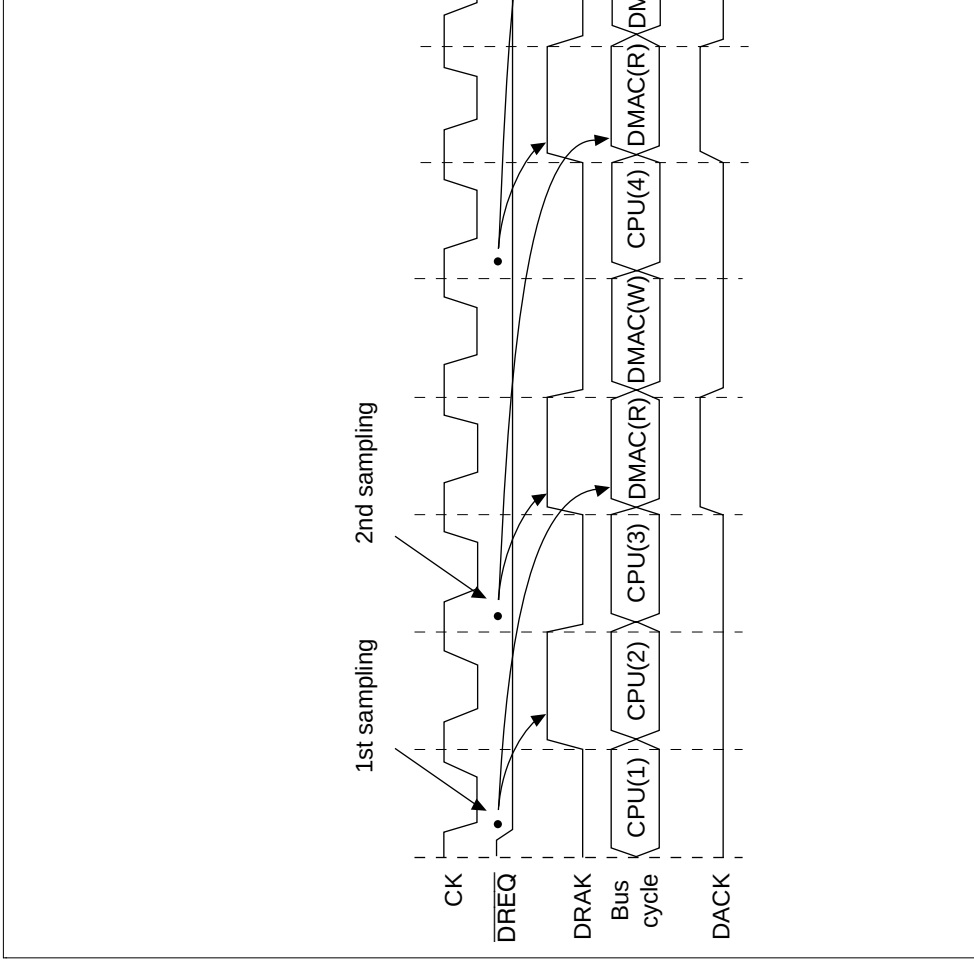


Figure 11.15 Cycle Steal, Dual Address, and Level Detection (Fastest Operati

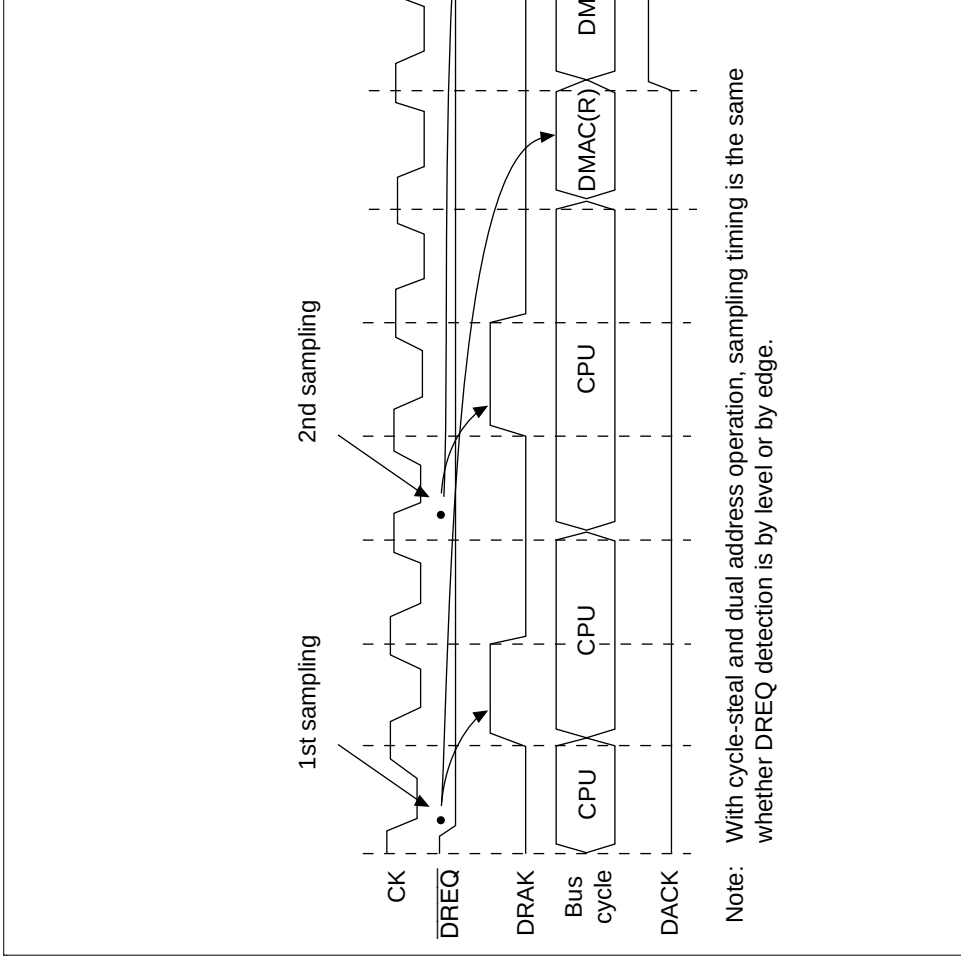


Figure 11.16 Cycle Steal, Dual Address, and Level Detection (Normal Opera

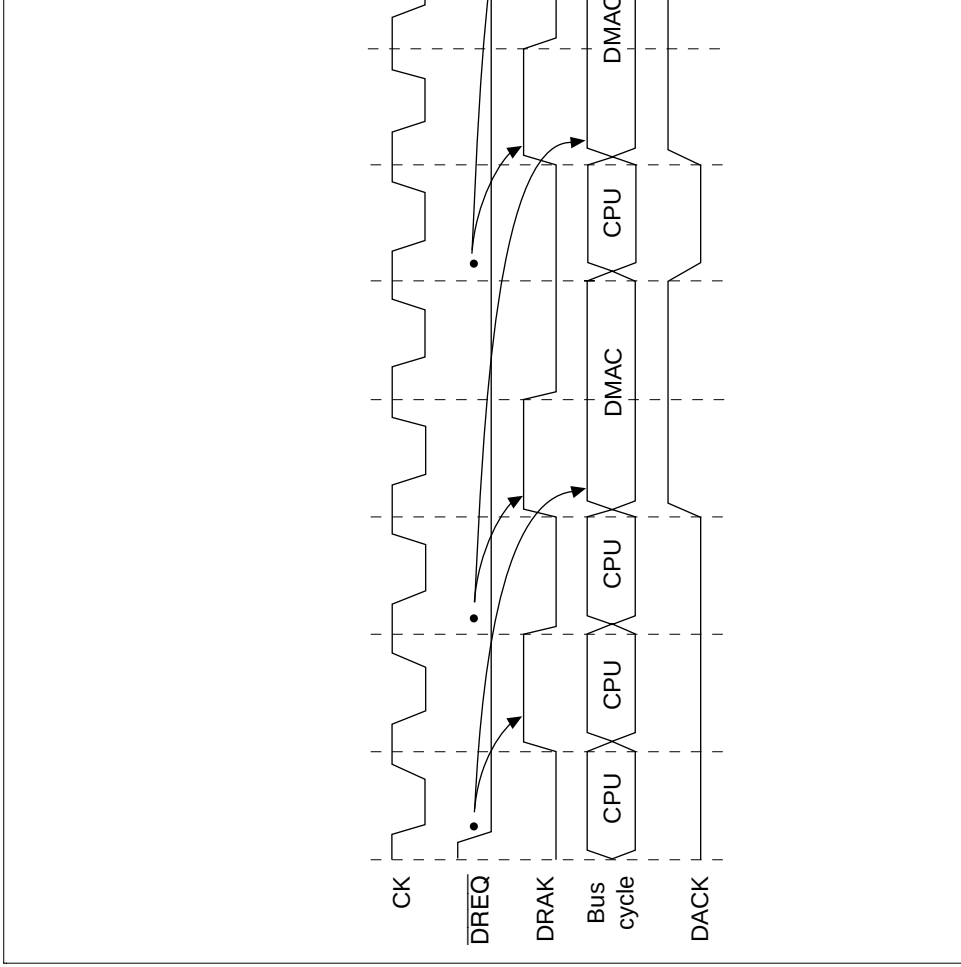


Figure 11.17 Cycle Steal, Single Address, and Level Detection (Fastest Opera

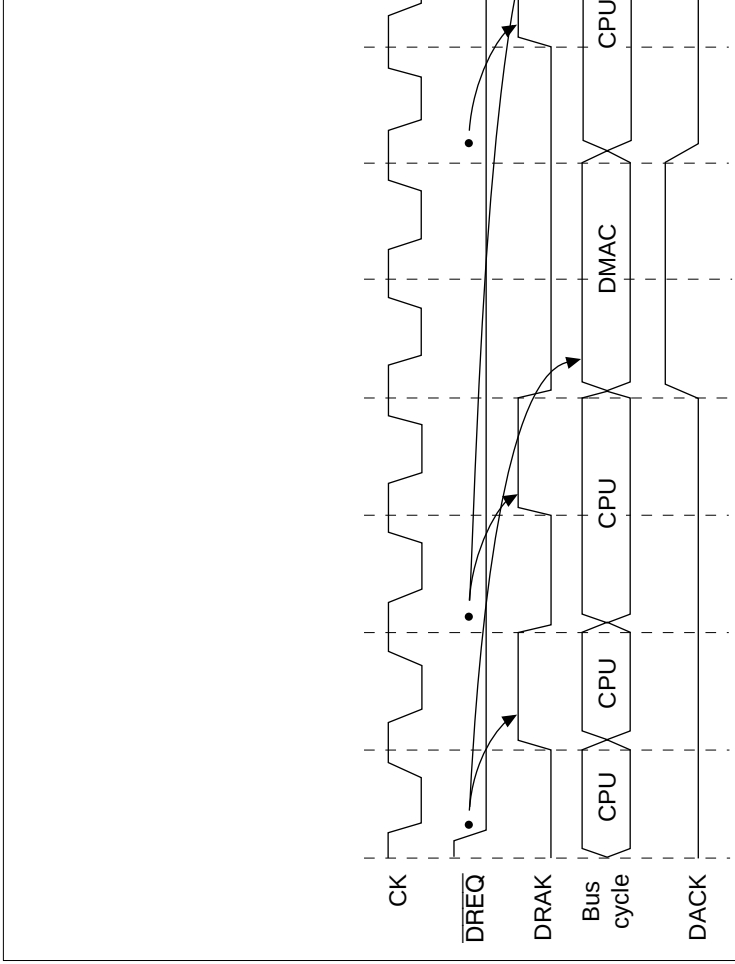


Figure 11.18 Cycle Steal, Single Address, and Level Detection (Normal Operat

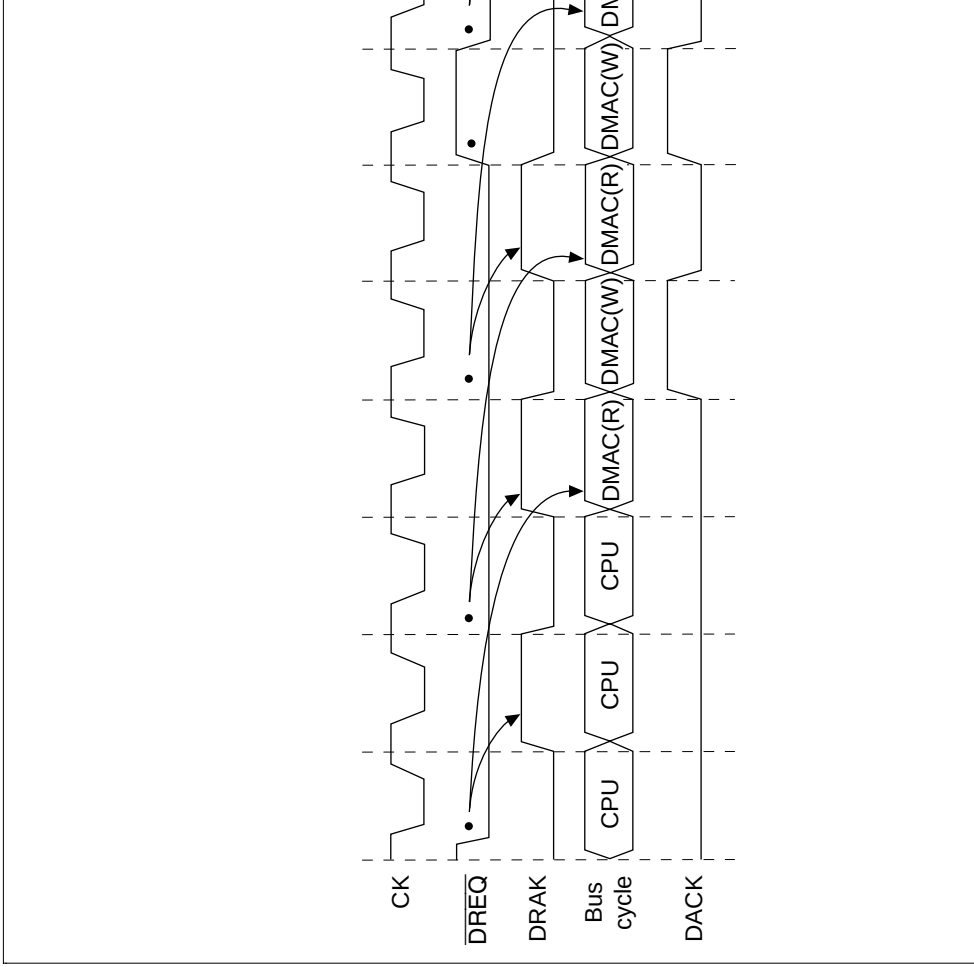


Figure 11.19 Burst Mode, Dual Address, and Level Detection (Fastest Operat

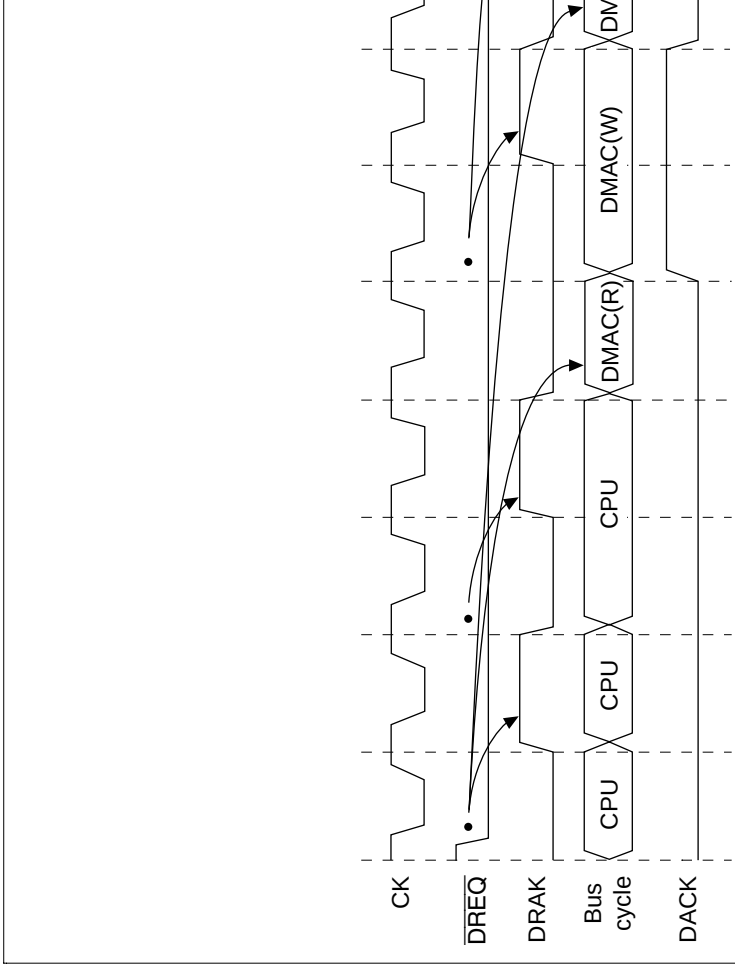


Figure 11.20 Burst Mode, Dual Address, and Level Detection (Normal Opera

sampling timing during this period begins from the start of the transfer one bus cycle before 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 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.

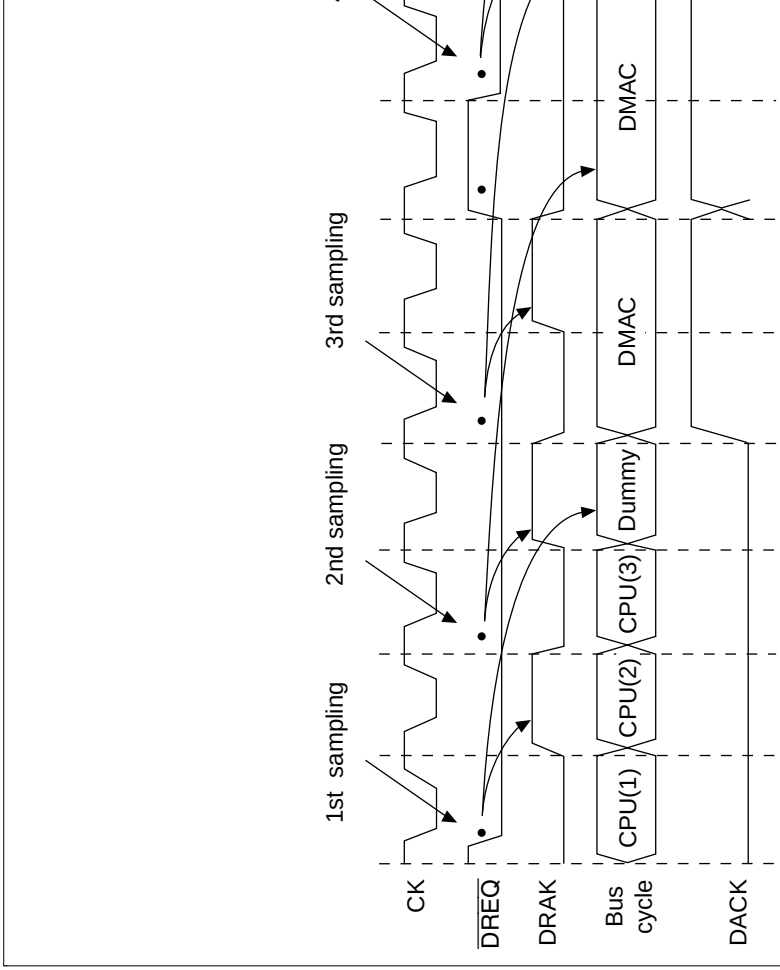


Figure 11.21 Burst Mode, Single Address, and Level Detection (Fastest Opera

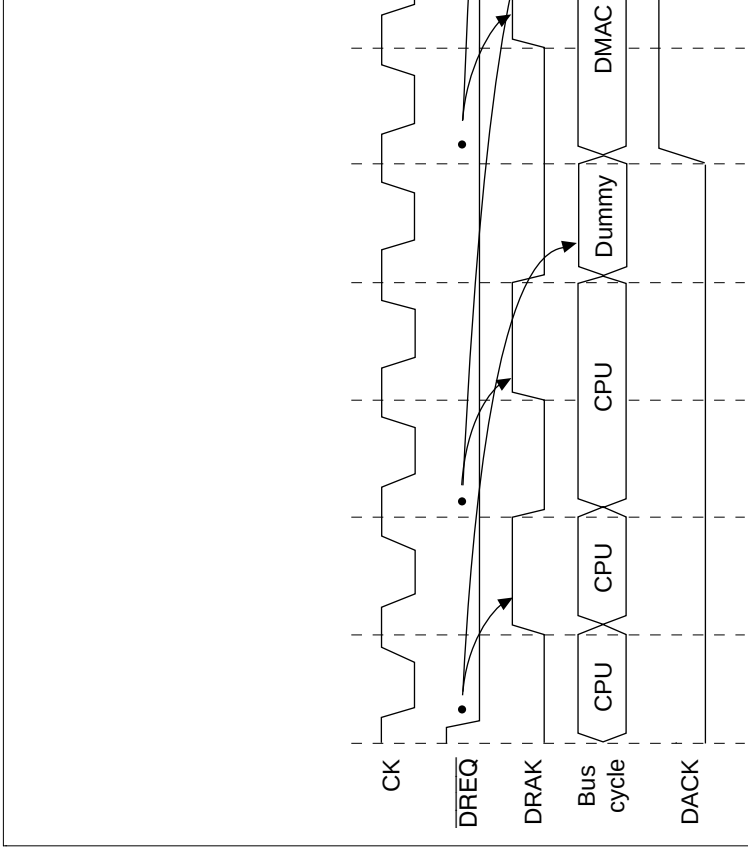


Figure 11.22 Burst Mode, Single Address, and Level Detection (Normal Opera

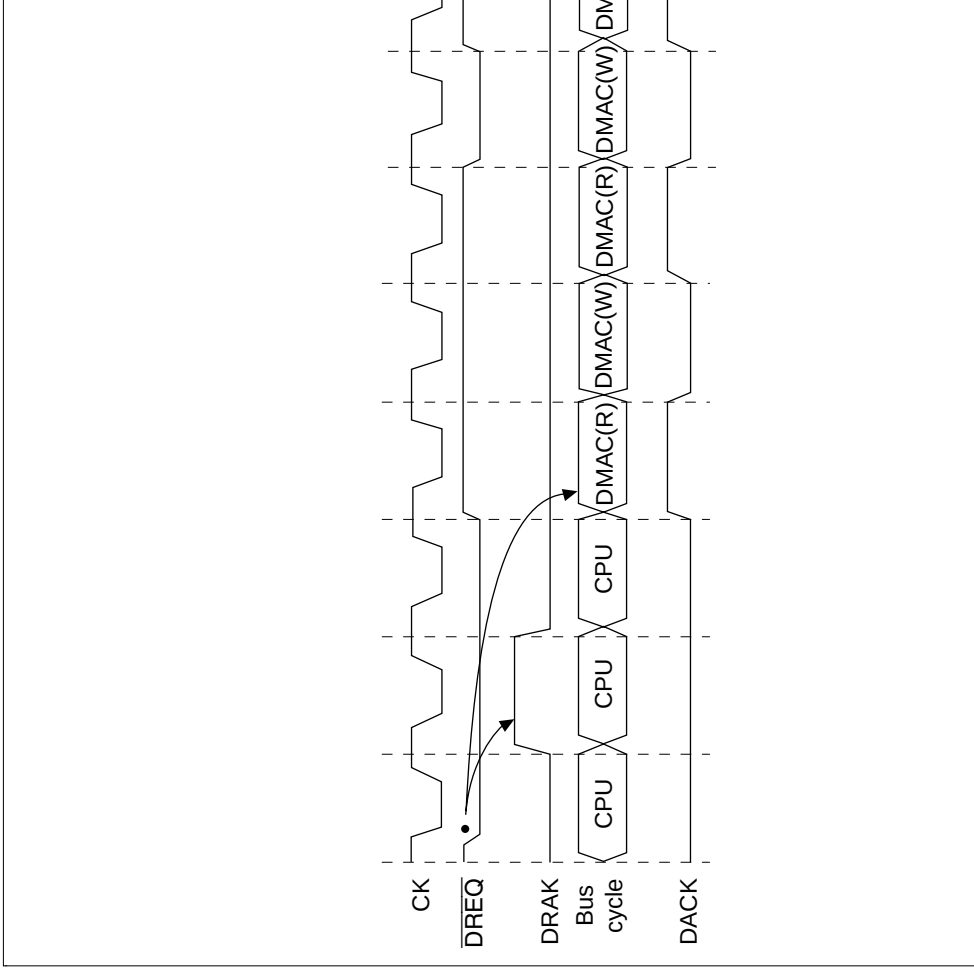


Figure 11.23 Burst Mode, Dual Address, and Edge Detection

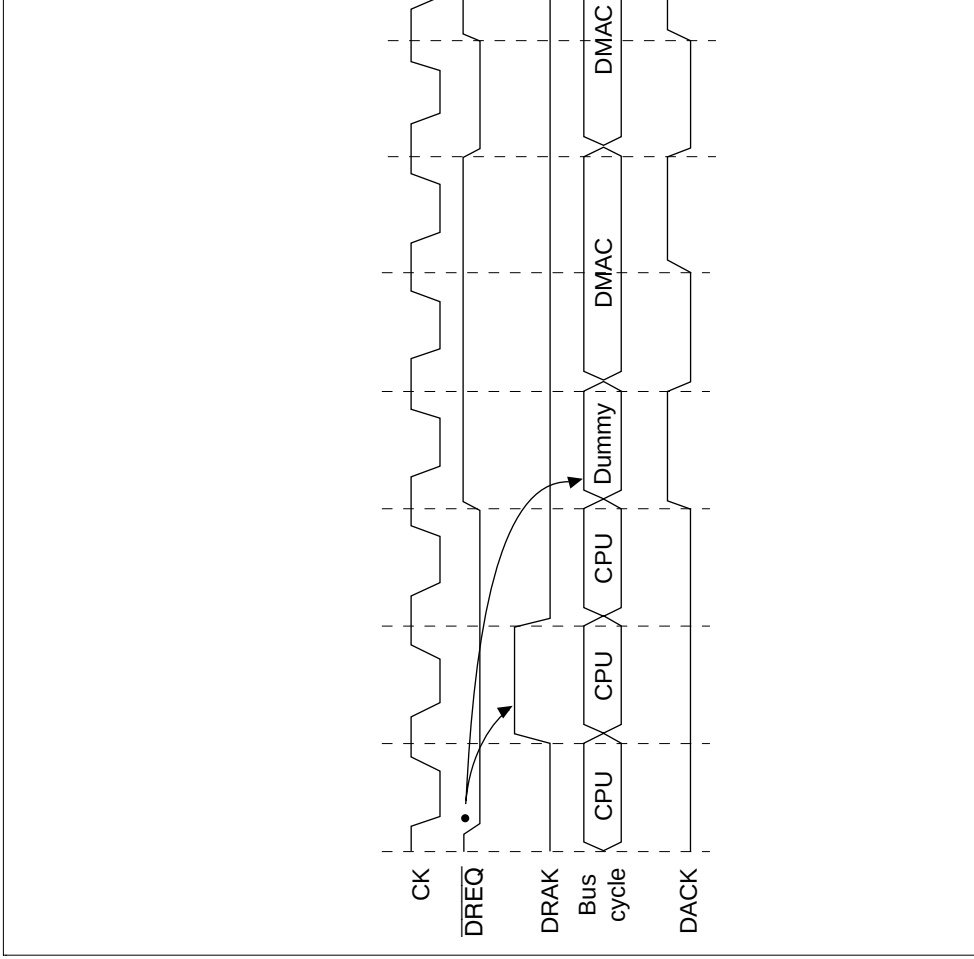


Figure 11.24 Burst Mode, Single Address and Edge Detection

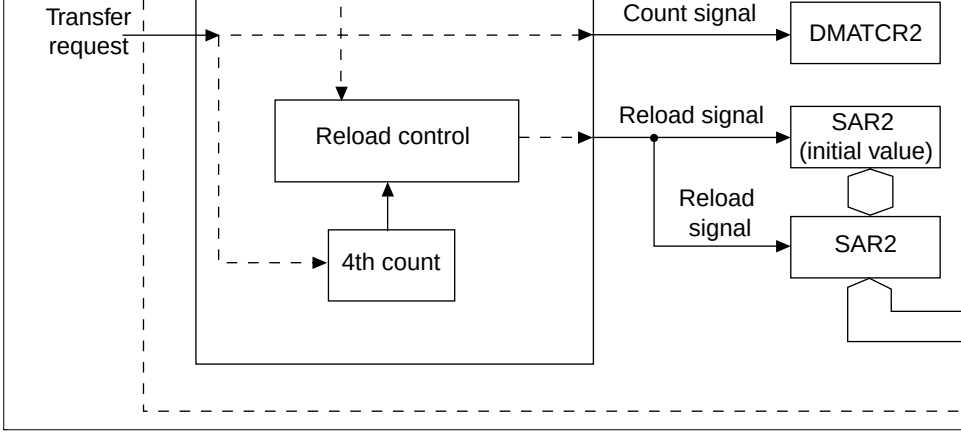


Figure 11.25 Source Address Reload Function

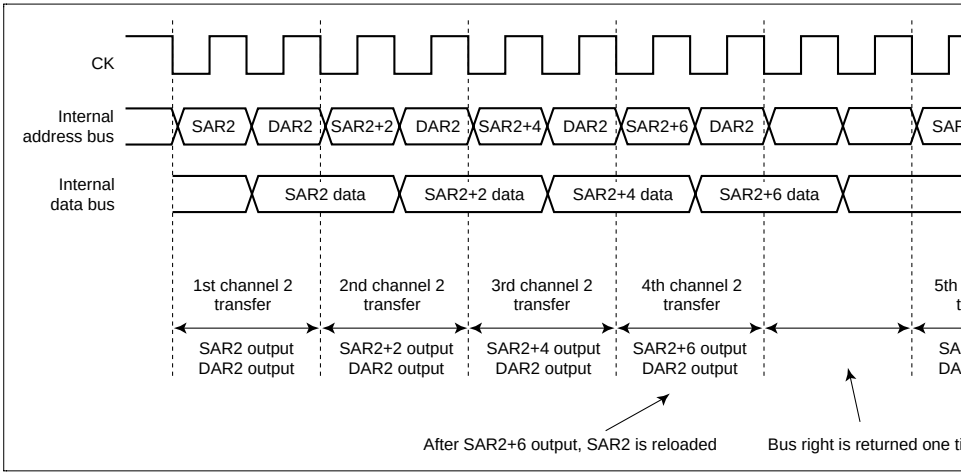


Figure 11.26 Source Address Reload Function Timing Chart

the address reload function, SAR, DAR2, and DMATCR2 settings must be carried out be
execution.

11.3.12 DMA Transfer Ending Conditions

The DMA transfer ending conditions vary for individual channels ending and for all chan
ending together.

Individual Channel Ending Conditions: There are two ending conditions. A transfer en
the value of the channel's DMA transfer count register (DMATCR) is 0, or when the DE
channel's CHCR is cleared to 0.

- When DMATCR is 0: When the DMATCR value becomes 0 and the corresponding c
DMA transfer ends, the transfer end flag bit (TE) is set in the CHCR. If the IE (interru
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 t
channel's CHCR. The TE bit is not set when this happens.

Conditions for Ending All Channels Simultaneously: Transfers on all channels end wh
NMIF (NMI flag) bit or AE (address error flag) bit is set to 1 in the DMAOR, or when th
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
error occurs, the NMIF or AE bit is set to 1 in the DMAOR and all channels stop thei
transfers. The DMAC obtains the bus rights, and if these flags are set to 1 during exec
a transfer, DMAC halts operation when the transfer processing currently being execut
and transfers the bus right to the other bus master. Consequently, even if the NMIF or
are set to 1 during a transfer, the DMA source address register (SAR), designation add
register (DAR), and transfer count register (TCR) are all updated. The TE bit is not se
resume the transfers after NMI interrupt or address error processing, clear the appropr
bit to 0. To avoid restarting a transfer on a particular channel, clear its DE bit to 0.

one bus cycle. Also, since the DMAC is located in word space, while a word-size access
DMAC is completed in one bus cycle, a longword-size access is automatically divided i
word accesses, requiring two bus cycles (six basic clock cycles). These two bus cycles a
executed consecutively; a different bus cycle is never inserted between the two word acc
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
transferred to external memory using the DMAC channel 3.

Table 11.7 indicates the transfer conditions and the setting values of each of the register

Table 11.7 Transfer Conditions and Register Set Values for Transfer between On SCI and External Memory

Transfer Conditions	Register	Value
Transfer source: RDR0 of on-chip SCI0	SAR3	H'FFFF81A
Transfer destination: external memory	DAR3	H'00400000
Transfer count: 64 times	DMATCR3	H'0000004(
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

Transfer source: external RAM	SAR1	H'00400000
Transfer destination: external device with DACK	DAR1	(access by D
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 m 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.

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 transfers. In the above example, when a transfer request is input from the A/D converter size data is first read in from the H'FFFF83F0 register of AD0 and that data is written to chip memory address H'FFFFF001. Because a byte size transfer was performed, the SA DAR values at this point are H'FFFF83F1 and H'FFFFF001, respectively. Also, because 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 and sixth transfers and the SAR value continues to increment from H'FFFF83F3 to H'FF H'FFFF83F5 and so on. However, when the address reload is on, the DMAC transfer is upon completion of the fourth one and the bus right request signal to the CPU is cleared time, the value stored in SAR is not H'FFFF83F3–H'FFFF83F4, but H'FFFF83F3–H'FF return to the initially established address. The DAR value always continues to be decren regardless of whether the address reload is on or off.

The DMAC internal status, due to the above operation after completion of the fourth tra indicated in table 11.10 for both address reload on and off.

CHCR is set to 1, are issued regardless of whether the address reload is on or

2. If transfer request source flag clears are executed until the DMATCR value bec 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 case 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 f There are cases where abnormal operation will result if anything else is design

To execute transfers after the fifth one when the address reload is on, make the transfer re source issue another transfer request signal.

11.4.4 Example of DMA Transfer between A/D Converter and Internal Memory (Reload On) (A Mask)

In this example the on-chip A/D converter (A/D1) is the transfer source and the internal m the transfer destination, and the address reload on.

Table 11.11 indicates the transfer conditions and the setting values of each of the register

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
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'FFFF8408 register and that data is written to
chip memory address H'FFFFF001. Because a byte size transfer was performed, the SA
DAR values at this point are H'FFFF8409 and H'FFFFF001, respectively. Also, because
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
and sixth transfers and the SAR value continues to increment from H'FFFF840B to H'FF
to H'FFFF840D and so on. However, when the address reload is on, the DMAC transfer
upon completion of the fourth transfer and the bus right request signal to the CPU is clea
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 contin
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 tra
indicated in table 11.12 for both address reload on and off.

CHCR is set to 1, are issued regardless of whether the address reload is on or
2. If transfer request source flag clears are executed until the DMATCR value bec
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 case
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 f
There are cases where abnormal operation will result if anything else is design

To execute more than four transfers with the address reload on, make the transfer request
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 memor
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 register

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 n
the transfer source data. In the case of indirect addressing, the value stored in the SAR a
read, then that value is used as the address and the data read from that address is used as
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
the address located at H'00400000, which is the value set in SAR3, is performed first. T
H'00450000 is stored at this H'00400000 address, and the DMAC first reads this H'0045
value. It then uses this read value of H'00450000 as an address and reads the value of H'
stored in the H'00450000 address. It then writes the value H'55 to the address H'FFFF81
designated by DAR3 to complete one indirect address transfer.

With indirect addressing, the first executed data read from the address established in SA
results in a longword size transfer regardless of the TS0, TS1 bit designations for transfe
size. However, the transfer source address fixed and increment or decrement designation
according to the SM0, SM1 bits. Consequently, despite the fact that the transfer data siz
designation is byte in this example, the SAR3 value at the end of one transfer is H'00400
write operation is exactly the same as an ordinary dual address transfer write operation.

instances where abnormal operation will result if any other registers are established la

7. After the DMATCR count becomes 0 and the DMA transfer ends normally, always w
the DMATCR, even when executing the maximum number of transfers on the same c
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. Th
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 fu
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
level when performing the DMAC establishment.
11. When operating in single address mode, establish an external address as the address. T
instances where abnormal operation will result if an internal address is established.
12. Do not access DMAC register empty addresses (H'FFFF86B2–H'FFFF86BF). Operati
cannot be guaranteed when empty addresses are accessed.

channels 1 and 2 that can be set to function independently as output compare or input
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 fal detection.
  - — Counter clearing function: Counters can be cleared by a compare-match or input
  - — Synchronizing mode: Two or more timer counters (TCNT) can be written to simultaneously. Two or more timer counters can be simultaneously cleared by a match or input capture. Counter synchronization functions enable synchronized r input/output.
  - — PWM mode: PWM output can be provided with any duty cycle. When combined counter synchronizing function, enables up to twelve-phase PWM output. (With 0–2 set to PWM mode 2, channels 3–4, and channels 0–4 synchronized with TG 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 channe 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 wa comparator type six-phase PWM waveform can be output.

— Channels 0–4 compare-match/input capture signals can be used as A/D converter conversion start triggers.

		TIOC0B	TIOC1B	TIOC2B	TIOC3B	TIOC
		TIOC0C			TIOC3C	TIOC
		TIOC0D			TIOC3D	TIOC
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 matc capt
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	TGR pare inpu

capture 0A	capture 1A	capture 2A	capture 3A	captu
Compare match/input capture 0B	Compare match/input capture 1B	Compare match/input capture 2B	Compare match/input capture 3B	Comp match captu
Compare match/input capture 0C	Overflow	Overflow	Compare match/input capture 3C	Comp match captu
Compare match/input capture 0D	Underflow	Underflow	Compare match/input capture 3D	Comp match captu
Overflow	—	—	Overflow	Overfl under

12.1.2 Block Diagram

Figure 12.1 is the block diagram of the MTU.

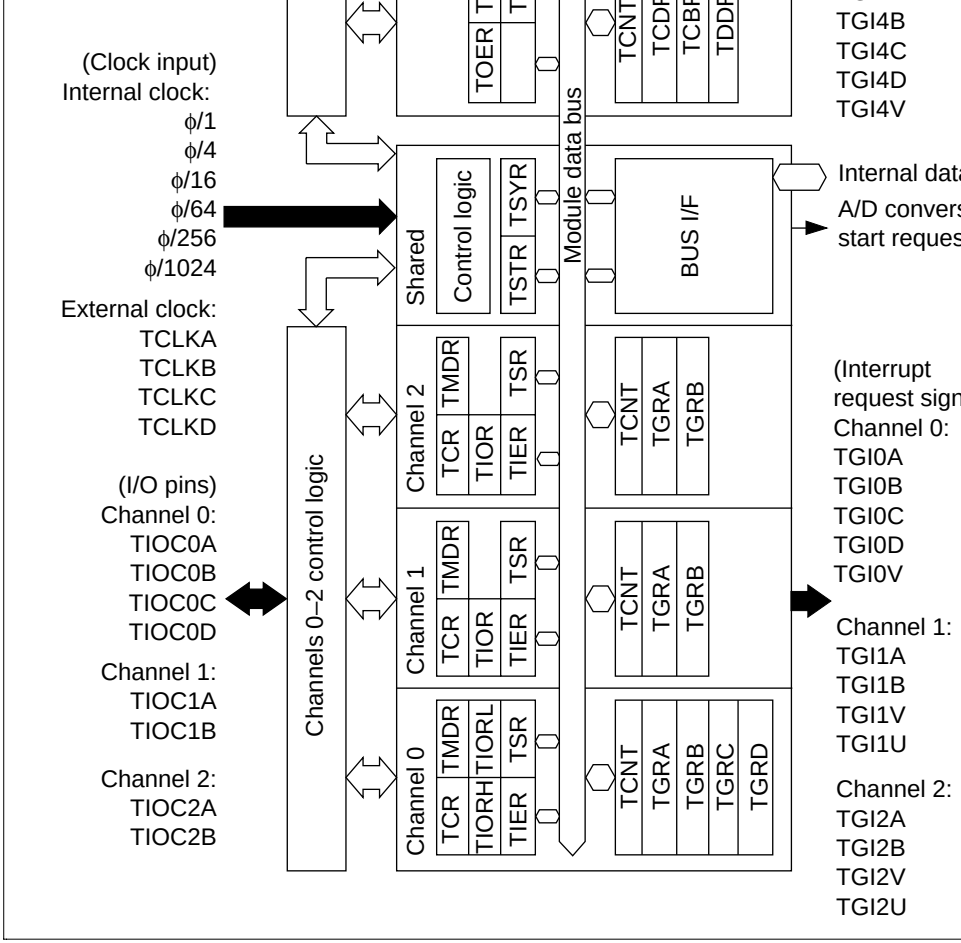


Figure 12.1 MTU Block Diagram

	Clock input D	TCLKD	I	Clock D input pin (B-phase input pin in cha 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

	compare-match 3C			In complementary PWM/reset synchronou mode
	Input capture/output compare-match 3D	TIOC3D	I/O	TGR3D input capture input/output compar pin In complementary PWM/reset synchronou mode, PWM output/U phase output pin
4	Input capture/output compare-match 4A	TIOC4A	I/O	TGR4A input capture input/output compar output/PWM output pin In complementary PWM/reset synchronou mode, PWM output/V phase output pin
	Input capture/output compare-match 4B	TIOC4B	I/O	TGR4B input capture input/output compar pin In complementary PWM/reset synchronou mode, PWM output/W phase output pin
	Input capture/output compare-match 4C	TIOC4C	I/O	TGR4C input capture input/output compar output/PWM output pin In complementary PWM/reset synchronou mode, PWM output/V phase output pin
	Input capture/output compare-match 4D	TIOC4D	I/O	TGR4D input capture input/output compar pin In complementary PWM/reset synchronou mode, PWM output/W phase output pin

Note: The TIOC pins output undefined values when they are set to input capture and tir by the pin function controller (PFC).

	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, 3
	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
	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, 3
	General register 1A	TGR1A	R/W	H'FFFF	H'FFFF8288	
	General register 1B	TGR1B	R/W	H'FFFF	H'FFFF828A	

	General register 2B	TGR2B	R/W	H'FFFF	H'FFFF82AA	
3	Timer control register 3	TCR3	R/W*3	H'00	H'FFFF8200	8, 1
	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, 1
	Timer counter 3	TCNT3	R/W*3	H'0000	H'FFFF8210	16,
	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,
	General register 3D	TGR3D	R/W	H'FFFF	H'FFFF8226	
4	Timer control register 4	TCR4	R/W*3	H'00	H'FFFF8201	8, 1
	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, 1

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, 3
Timer dead time data register	TDDR	R/W*3	H'FFFF	H'FFFF8216	
Timer subcounter	TCNTS	R	H'0000	H'FFFF8220	16, 3
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 (B cleared, access becomes impossible (undefined read/write disabled).

Initial value:	0	0	0	0	0	0	0
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
	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1
Initial value:	0	0	0	0	0	0	0
R/W:	R	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 the TCNT counter.

	1	Synchronizing clear: TCNT is cleared in synchronization with 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 the buffer registers have priority and compare-match/input captures do not occ

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 ca
	1	0	TCNT is cleared by TGRB compare-match or input ca
		1	Synchronizing clear: TCNT is cleared in synchronizat 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 mo
*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 inp edges. When counting is done on both edges of the internal clock the input clock frequ becomes 1/2 (Example: both edges of $\phi/4$ = rising edge of $\phi/2$). When phase count mo used with channels 1, 2, these settings are ignored, as the phase count mode settings h priority.

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

Channel	Internal Clock						Other Channel Overflow/ Underflow	External Clock		
	φ/1	φ/4	φ/16	φ/64	φ/256	φ/1024		TCLKA	TCLKB	TC KC
0	O	O	O	O	X	X	X	O	O	O
1	O	O	O	O	O	X	O	O	O	X
2	O	O	O	O	X	O	X	O	O	O
3	O	O	O	O	O	O	X	O	O	X
4	O	O	O	O	O	O	X	O	O	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

		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.

		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

Channels 1, 2: TMDR1, TMDR2:

Bit:	7	6	5	4	3	2	1
	—	—	—	—	MD3	MD2	MD1
Initial value:	1	1	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W

- Bits 7, 6—Reserved: These bits are reserved. They always read as 1, and cannot be m
- Bit 5—Buffer Operation B (BFB): Designates whether to use the TGRB register for n operation, or buffer operation in combination with the TGRD register. When using TG 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 re and cannot be modified.

Bit 5: BFB	Description
0	TGRB operates normally (initial value)
1	TGRB and TGRD buffer operation

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 an

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 channel 3. When channel 3 is set to reset synchronous PWM mode or compl PWM mode, the channel 4 settings become ineffective and automatically con channel 3 settings. However, do not set channel 4 to reset synchronous PWM complementary PWM mode. Reset synchronous PWM mode and complemen mode can not be set for channels 0, 1, 2.

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 functi
- Bits 3–0—I/O Control A3–B0 (IOA3–IOA0): These bits set the TGRA register functi

Channels 0, 3, 4: TIOR0L, TIOR3L, TIOR4L:

Bit:	7	6	5	4	3	2	1
	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1
Initial value:	0	0	0	0	0	0	0
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 bec 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 functi
- Bits 3–0—I/O Control C3–C0 (IOC3–IOC0): These bits set the TGRC register functi

		1	0		output	Output 1 on compare-m
			1		is 1	Toggle output on compa
1	0	0	0	TGR0B	Capture	Input capture on rising e
			1	is an	input source	Input capture on falling e
		1	0	input	is the	Input capture on both ed
			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	

			1		is 1	Toggle output on compar
1	0	0	0	TGR0A	Capture	Input capture on rising ed
			1	is an	input source	Input capture on falling ed
		1	0	input	is the	Input capture on both edg
			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	

		1	0		output	Output 1 on compare-m
			1		is 1	Toggle output on compa
1	0	0	0	TGR0D	Capture	Input capture on rising e
			1	is an	input source	Input capture on falling e
		1	0	input	is the	Input capture on both ed
			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 reg settings become ineffective and input capture/output compares do not occur.

			1		is 1	Toggle output on compar
1	0	0	0	TGR0C	Capture	Input capture on rising ed
			1	is an	input source	Input capture on falling ed
		1	0	input	is the	Input capture on both edg
			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 regis
settings become ineffective and input capture/output compares do not occur.

		1	0		output	Output 1 on compare-m
			1		is 1	Toggle output on compa
1	0	0	0	TGR1B	Capture	Input capture on rising e
			1	is an	input source	Input capture on falling e
		1	0	input	is the	Input capture on both ed
			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

			1		is 1	Toggle output on compar
1	0	0	0	TGR1A	Capture	Input capture on rising ed
			1	is an	input source	Input capture on falling ed
		1	0	input	is the	Input capture on both edg
			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 cap generation
			1		capture	

		1	0		output	Output 1 on compare-m
			1		is 1	Toggle output on compa
1	0	0	0	TGR2B	Capture	Input capture on rising e
			1	is an	input source	Input capture on falling e
		1	0	input	is the	Input capture on both ed
			1	capture	TIOC2B pin	
	1	0	0	register		Input capture on rising e
			1			Input capture on falling e
		1	0			Input capture on both ed
			1			

			1		is 1	Toggle output on compar
1	0	0	0	TGR2A	Capture	Input capture on rising ed
			1	is an	input source	Input capture on falling ed
		1	0	input	is the	Input capture on both edg
			1	capture	TIOC2A pin	
	1	0	0	register		Input capture on rising ed
			1			Input capture on falling ed
		1	0			Input capture on both edg
			1			

		1	0		output	Output 1 on compare-m
			1		is 1	Toggle output on compa
1	0	0	0	TGR3B	Capture	Input capture on rising e
			1	is an	input source	Input capture on falling e
		1	0	input	is the	Input capture on both ed
			1	capture	TIOC3B pin	
	1	0	0	register		Input capture on rising e
			1			Input capture on falling e
		1	0			Input capture on both ed
			1			

			1		is 1	Toggle output on compar
1	0	0	0	TGR3A	Capture	Input capture on rising ed
			1	is an	input source	Input capture on falling ed
		1	0	input	is the	Input capture on both edg
			1	capture	TIOC3A pin	
	1	0	0	register		Input capture on rising ed
			1			Input capture on falling ed
		1	0			Input capture on both edg
			1			

		1	0		output	Output 1 on compare-m
			1		is 1	Toggle output on compa
1	0	0	0	TGR3D	Capture	Input capture on rising e
			1	is an	input source	Input capture on falling e
		1	0	input	is the	Input capture on both ed
			1	capture	TIOC3D pin	
	1	0	0	register		Input capture on rising e
			1			Input capture on falling e
		1	0			Input capture on both ed
			1			

Note: When the BFB bit of TMDR3 is set to 1 and TGR3D is being used as a buffer reg settings become ineffective and input capture/output compares do not occur.

			1		is 1	Toggle output on compar
1	0	0	0	TGR3C	Capture	Input capture on rising ed
			1	is an	input source	Input capture on falling ed
		1	0	input	is the	Input capture on both edg
			1	capture	TIOC3C pin	
	1	0	0	register		Input capture on rising ed
			1			Input capture on falling ed
		1	0			Input capture on both edg
			1			

Note: When the BFA bit of TMDR3 is set to 1 and TGR3C is being used as a buffer regis settings become ineffective and input capture/output compares do not occur.

		1	0		output	Output 1 on compare-m
			1		is 1	Toggle output on compa
1	0	0	0	TGR4B	Capture	Input capture on rising e
			1	is an	input source	Input capture on falling e
		1	0	input	is the	Input capture on both ed
			1	capture	TIOC4B pin	
	1	0	0	register		Input capture on rising e
			1			Input capture on falling e
		1	0			Input capture on both ed
			1			

			1		is 1	Toggle output on compar
1	0	0	0	TGR4A	Capture	Input capture on rising ed
			1	is an	input source	Input capture on falling ed
		1	0	input	is the	Input capture on both edg
			1	capture	TIOC4A pin	
	1	0	0	register		Input capture on rising ed
			1			Input capture on falling ed
		1	0			Input capture on both edg
			1			

		1	0		output	Output 1 on compare-m
			1		is 1	Toggle output on compa
1	0	0	0	TGR4D	Capture	Input capture on rising e
			1	is an	input source	Input capture on falling e
		1	0	input	is the	Input capture on both ed
			1	capture	TIOC4D pin	
	1	0	0	register		Input capture on rising e
			1			Input capture on falling e
		1	0			Input capture on both ed
			1			

Note: When the BFB bit of TMDR4 is set to 1 and TGR4D is being used as a buffer reg settings become ineffective and input capture/output compares do not occur.

			1		is 1	Toggle output on compar
1	0	0	0	TGR4C	Capture	Input capture on rising ed
			1	is an	input source	Input capture on falling ed
		1	0	input	is the	Input capture on both edg
			1	capture	TIOC4C pin	
	1	0	0	register		Input capture on rising ed
			1			Input capture on falling ed
		1	0			Input capture on both edg
			1			

Note: When the BFA bit of TMDR4 is set to 1 and TGR4C is being used as a buffer regis 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 The MTU has five TIER registers, one each for channel. TIER is initialized to H'40 by a r by standby mode.

Channel 0: TIER0:

Bit:	7	6	5	4	3	2	1
	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB
Initial value:	0	1	0	0	0	0	0
R/W:	R/W	R	R	R/W	R/W	R/W	R/W

R/W:	R/W	R	R	R/W	R/W	R/W	R/W

- Bit 7—A/D Conversion Start Request Enable (TTGE): Enables or disables generatio 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 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 sho 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 w 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)

be 1.

Bit 2: TGIEC	Description
0	Disable interrupt requests (TGIC) due to the TGFC bit (initial va
1	Enable interrupt requests (TGIC) due to the TGFC bit

- Bit 1—TGR Interrupt Enable B (TGIEB): Enables or disables TGFB interrupt request 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 va
1	Enable interrupt requests (TGIB) due to the TGFB bit

- Bit 0—TGR Interrupt Enable A (TGIEA): Enables or disables TGFA interrupt reques 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 va
1	Enable interrupt requests (TGIA) due to the TGFA bit

Channels 1, 2: TSR1, TSR2:

Bit:	7	6	5	4	3	2	1
	TCFD	—	TCFU	TCFV	—	—	TGFB
Initial value:	1	1	0	0	0	0	0
R/W:	R	R	R/(W)*	R/(W)*	R	R	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
	TCFD	—	—	TCFV	TGFD	TGFC	TGFB
Initial value:	1	1	0	0	0	0	0
R/W:	R	R	R	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 o channel 1, 2, 3, 4 TCNT counters.

  This bit is reserved in channel 0. This bit always reads as 1. The write value should a 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 4: TCFV	Description
0	Clear condition: With TCFV =1, a 0 write to TCFV after reading (initial value)
1	Set condition: When the TCNT value overflows (H'FFFF → H'0

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 – 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 a 0.

Bit 3: TGFD	Description
0	Clear condition: With TGFD = 1, a 0 write to TGFD following a (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 valu to TGRD by the input capture signal)

	• When TGRC is functioning as input capture (the TCNT va 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 (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 val 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 (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 val to TGRA by the input capture signal)

3	TCNT3	Increment/decrement counter
4	TCNT4	Increment/decrement counter*2

Notes: *1 Can only be used as an increment/decrement counter in phase counting mode other channel overflow/underflow counting. It becomes an increment counter in cases.

*2 Can only be used as an increment counter in complementary PWM mode. It be an increment counter in all other cases.

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

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

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

Bit:	7	6	5	4	3	2	1
Initial value:	1	1	1	1	1	1	1
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 tim counters (TCNT) of channels 0–4. TSTR is initialized to H'00 upon power-on reset or st mode. Manual reset does not initialize TSTR.

Bit:	7	6	5	4	3	2	1
	CST4	CST3	—	—	—	CST2	CST1
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R	R	R/W	R/W

- Bits 7, 6, 2–0—Counter Start 4–0 (CST4–CST0): Select the start and stop of the tim (TCNT). The counter start to channel and bit to channel correspondence are indicate tables below.

If 0 is written to the CST bit during operation with the TIOC pin in output status, the stops, but the TIOC pin output compare output level is maintained. If a write is don TIOR register while the CST bit is a 0, the pin output level is updated to the establi initial output value. In complementary PWM mode or reset sync PWM mode, when written to the CST bit of a TIOC pin in output mode during operation, it returns to th 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 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 standby mode. Manual reset does not initialize TSYR.

Bit:	7	6	5	4	3	2	1
	SYNC4	SYNC3	—	—	—	SYNC2	SYNC1
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R	R	R/W	R/W

- Bits 7, 6, 2–0—Timer Synchronization 4–0 (SYNC4–SYNC0): Selects operation inde of, or synchronized to, other channels. Synchronous operation allows synchronous cle to multiple TCNT synchronous presets and other channel counter clears. A minimum channels must have SYNC bits set to 1 for synchronous operation. For synchronizatio clearing, it is necessary to set the TCNT counter clear sources (the CCLR2–CCLR0 b TCR register), in addition to the SYNC bit. The counter start to channel and bit-to-cha correspondence are indicated in the tables below.

	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 operati

*2 TCNT counter clear sources (CCLR2–CCLR0 bits of the TCR register) must l 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

12.2.10 Timer Output Master Enable Register (TOER)

The timer output master enable register (TOER) enables/disables output settings for outp TIOC4D, TIOC4C, TIOC3D, TIOC4B, TIOC4A, and TIOC3B. These pins do not outpu if the TOER bits have not been set. Set TOER of CH3 and CH2 prior to setting TIOR of CH4. The TOER is an 8-bit read/write register. The register is initialized to H'C0 by a p reset or in standby mode. Manual reset does not initialize TOER.

Bit:	7	6	5	4	3	2	1
	—	—	OE4D	OE4C	OE3D	OE4B	OE4A
Initial value:	1	1	0	0	0	0	0
R/W:	R	R	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

- Bit 3—Master Enable TIOC3D (OE3D): Enables or disables the TIOC3D pin MTU o

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 ou

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 o

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 ou

Bit 0: OE3B	Description
0	Disable TIOC3B pin MTU output (initial value)
1	Enable TIOC3B pin MTU output

- Bits 7, 5–2—Reserved: These bits always read as 1. The write value should always b
- Bit 6—PWM Synchronous Output Enable (PSYE): Selects the enable/disable of togg synchronized with the PWM period.

Bit 6: PSYE	Description
0	Toggle output synchronous with PWM period disabled (initial
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**	**Decremen**
0	High level*	Low level	High level	Low level (value)
1	Low level*	High level	Low level	High level

Note: * The reverse phase waveform initial output value changes to active level after ela 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 C**
0	High level	Low level	Low level	High level (ini
1	Low level	High level	High level	Low level

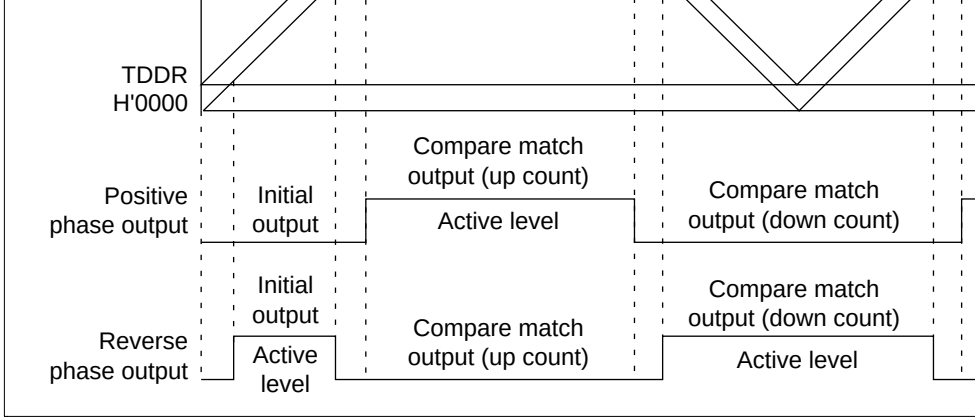


Figure 12.2 Complementary PWM Mode Output Level Example

12.2.12 Timer Gate Control Register (TGCR)

The timer gate control register (TGCR) is an 8-bit read/write register that controls the wa 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 mode. Manual reset does not initialize TGCR. These register settings are ineffective for a other than complementary PWM mode/reset-synchronized PWM mode.

Bit:	7	6	5	4	3	2	1
	—	BDC	N	P	FB	WF	VF
Initial value:	1	0	0	0	0	0	0
R/W:	R	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.

0	Output gate signals directly to reverse phase pin output (initial v
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 t phase pin (TIOC3B, TIOC4A, and TIOC4B) output, or to output by chopping the ga and the complementary PWM/reset-synchronized PWM output.

Bit 4: P	Description
0	Output gate signals directly to positive phase pin output (initial v
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 f 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 captur
1	Feedback input is register input (TGCR's UF, VF, WF settings)

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 complem PWM mode. The TCNTS counter is initialized to H'00 by a power-on reset or in standby 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
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R

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

R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
Initial value:	1	1	1	1	1	1	1
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 P mode. Set the PWM carrier sync value as the TCDR register value. This register is cons compared with the TCNTS counter in complementary PWM mode, and when a match o TCNTS counter switches direction (decrement to increment).

The TCDR register is initialized to H'FFFF by a reset or in standby mode. Manual reset initialize TCDR. Accessing the TCDR in 8-bit units is prohibited. Always access in 16-b

Bit:	15	14	13	12	11	10	9
Initial value:	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

Bit:	7	6	5	4	3	2	1	
Initial value:	1	1	1	1	1	1	1	
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
the bus master enables 16-bit read/writes. 8-bit read/write is not possible. Always access
units. Figure 12.3 shows an example of 16-bit register access operation.

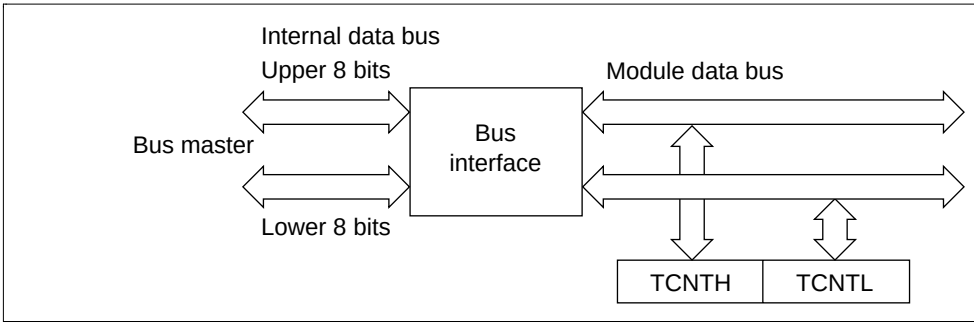


Figure 12.3 16-Bit Register Access Operation (Bus Master ↔ TCNT (16 Bit

Figure 12.4 8-Bit Register Access Operation (Bus Master ↔ TCR (Upper 8 I

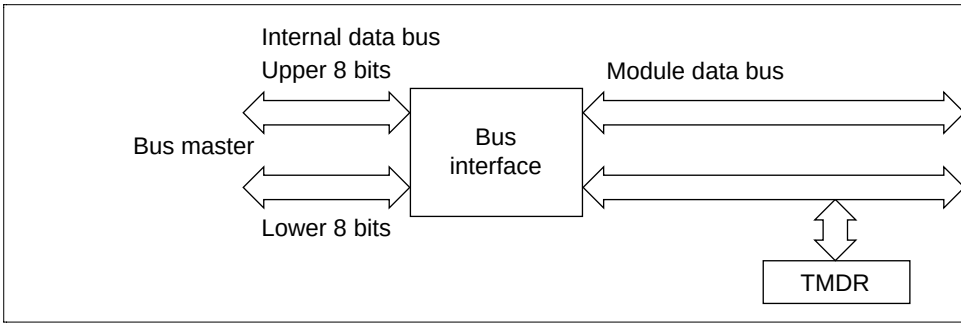


Figure 12.5 8-Bit Register Access Operation (Bus Master ↔ TMDR (Lower 8

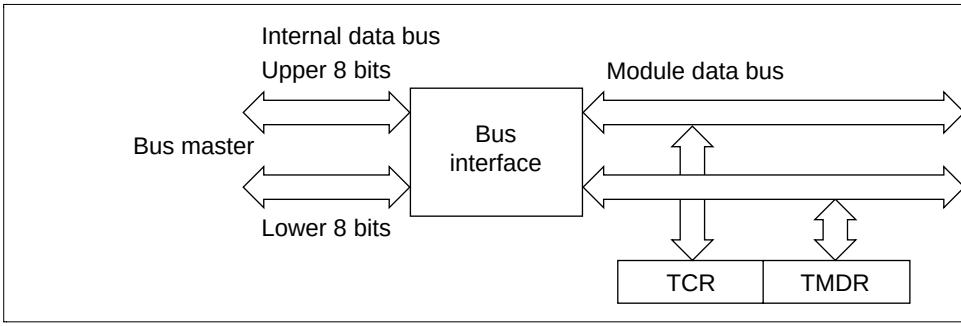


Figure 12.6 8-Bit Register Access Operation (Bus Master ↔ TCR, TMDR (16

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 opera be set to clear the TCNTs simultaneously.

Buffer Operation: When TGR is an output compare register, the buffer register value of corresponding channel is transferred to the TGR when a compare-match occurs. When TG input capture register, the TCNT counter value is transferred to the TGR when an input ca occur simultaneously the value previously stored in the TGR is transferred to the buffer re

Cascade Connection Operation: The channel 1 and channel 2 counters (TCNT1 and TC 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 TIOR register. Each TGR can be set for PWM waveform output with a duty cycle betwee and 100%.

Phase Counting Mode: In phase counting mode, the phase differential between two cloc from the channel 1 and channel 2 external clock input pins is detected and the TCNT cou operates as an up/down counter. In phase counting mode, the corresponding TCLK pins b clock inputs and TCNT functions as an up/down counter. It can be used as a two-phase en pulse input.

Reset-Synchronized PWM Mode: Three-phase positive and negative PWM waveforms obtained using channels 3 and 4 (the three phases of the PWM waveform share a transitio on one side). When set for reset-synchronized PWM mode, TGR3A, TGR3B, TGR4A, an TGR4B automatically become output compare registers. The TIOC3A, TIOC3B, TIOC4A TIOC4B, TIOC4C, and TIOC4D pins also become PWM output pins, and TCNT3 and TC become upcounters. TCNT4, TGR4A, and TGR4B are isolated from TCNT4.

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 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 T clearing source for the TCNT.
3. Set the TGR selected in step 2 as an output compare register using the timer I/O cont 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.

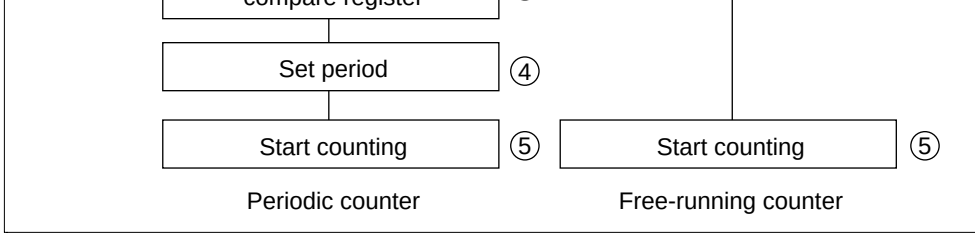


Figure 12.7 Procedure for Selecting the Counting Operation

Free-Running Counter Operation Example: A reset of the MTU timer counters (TCNT them all in the free-running mode. When a bit in the TSTR is set to 1, the corresponding t counter operates as a free-running counter and begins to increment. When the count overf from H'FFFF–H'0000, the TCFV bit in the timer status register (TSR) is set to 1. If the TC in the timer's corresponding timer interrupt enable register (TIER) is set to 1, the MTU w an interrupt request to the interrupt controller. After the TCNT overflows, counting contin H'0000. Figure 12.8 shows an example of free-running counter operation.

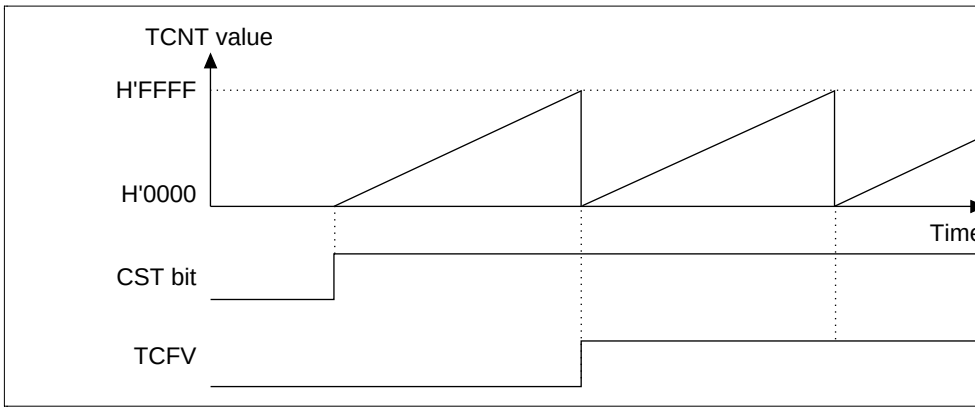


Figure 12.8 Free-Running Counter Operation

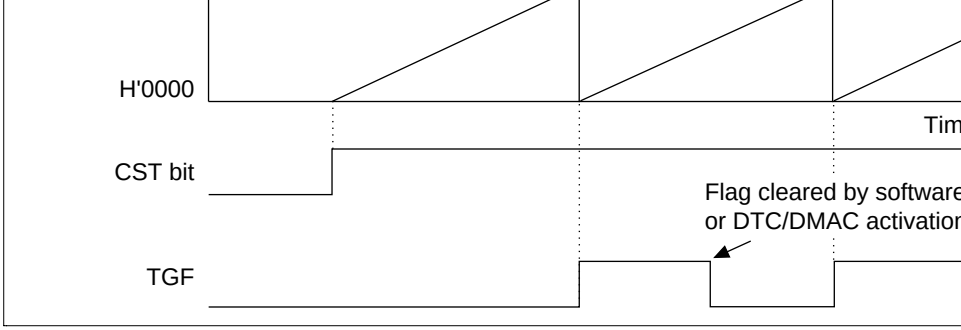


Figure 12.9 Periodic Counter Operation

Compare-Match Waveform Output Function: The MTU can output 0 level, 1 level, o 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 outpu toggle output for compare-match output. The TIOC pin will output the set initial valu 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 Oper

Waveform Output Operation (0 Output/1 Output): Figure 12.11 shows 0 output/1 out the example, TCNT is a free-running counter, 1 is output upon compare-match A and 0 is upon compare-match B. When the pin level matches the set level, the pin level does not c

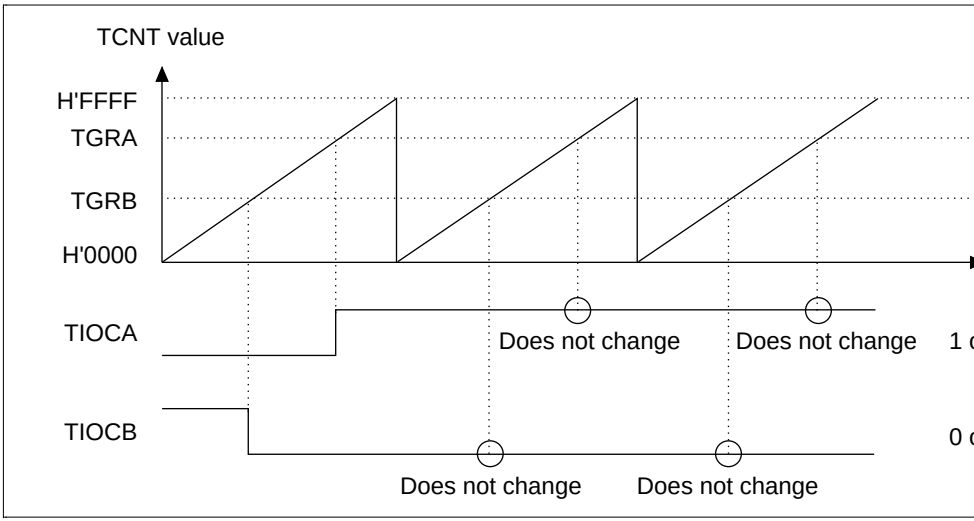


Figure 12.11 Example of 0 Output/1 Output

Waveform Output Operation (Toggle Output): Figure 12.12 shows the toggle output. example, the TCNT operates as a periodic counter cleared by compare-match B, with tog 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
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
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 ca
   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.

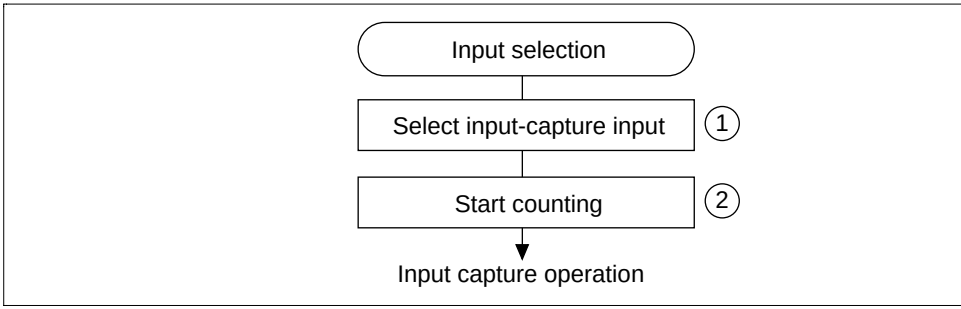


Figure 12.13 Procedure for Selecting Input Capture Operation

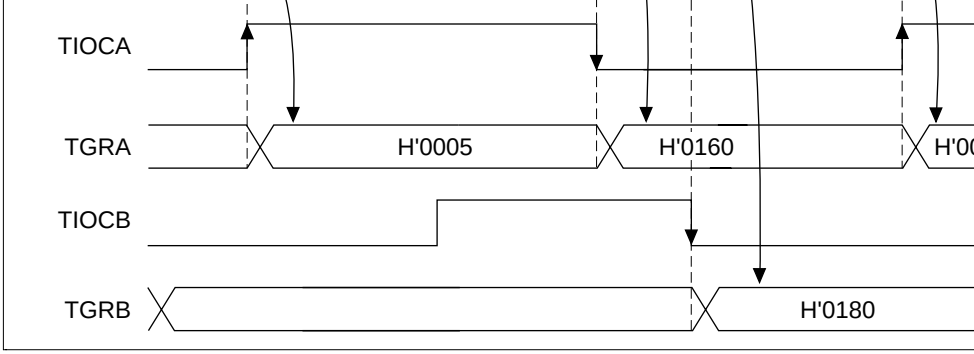


Figure 12.14 Input Capture Operation

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 T settings (synchronized clear).

The synchronizing mode can increase the number of TGR registers for a single time base channels can be set for synchronous operation.

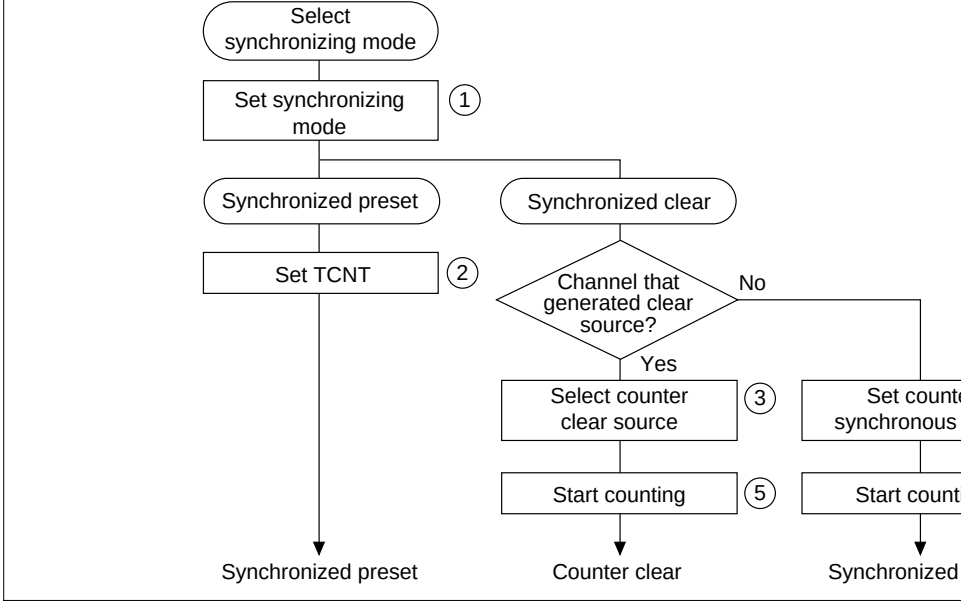


Figure 12.15 Procedure for Selecting Synchronizing Operation

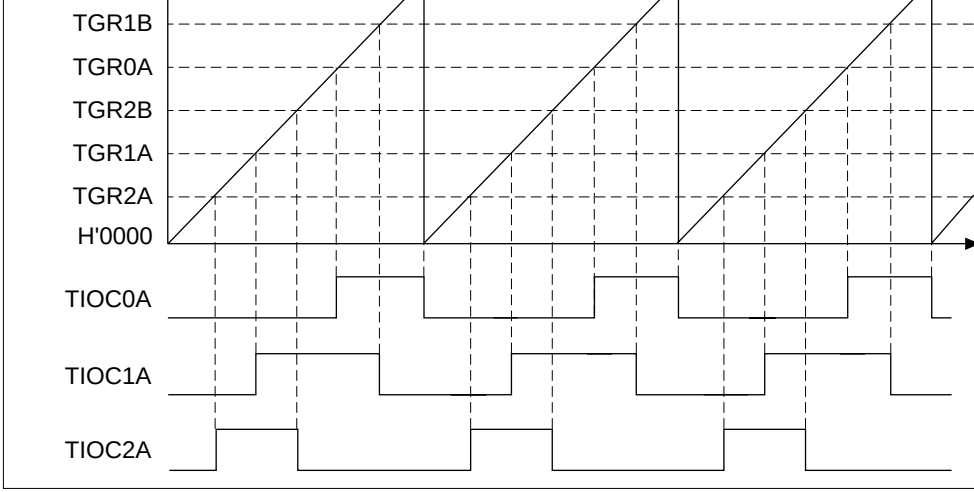


Figure 12.16 Synchronized Operation Example

4	TGR4A	TGR4C
	TGR4B	TGR4D

The buffer operation differs, depending on whether the TGR has been set as an input cap
register or an output compare register.

When TGR Is an Output Compare Register: When a compare-match occurs, the corr
channel buffer register value is transferred to the general register. Figure 12.17 shows an

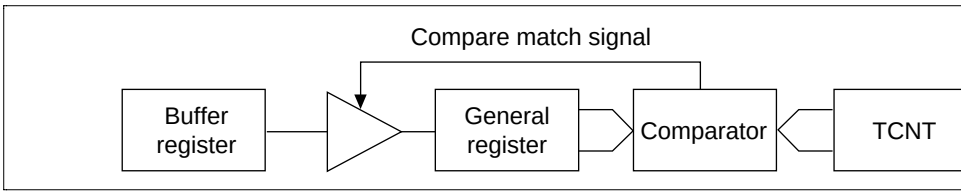


Figure 12.17 Compare Match Buffer Operation

When TGR Is an Input Capture Register: When an input capture occurs, the timer co
(TCNT) value is transferred to the general register (TGR), and the value that had been h
that time in the TGR is transferred to the buffer register (figure 12.18).

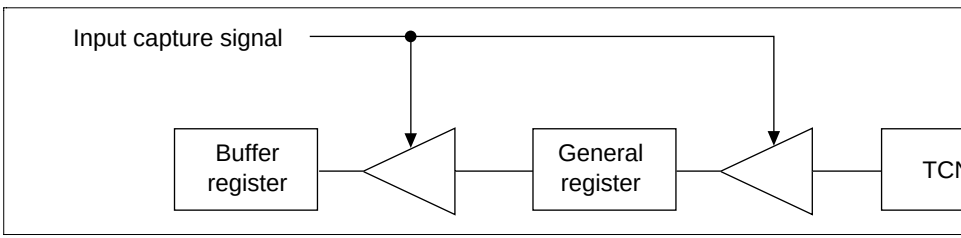


Figure 12.18 Input Capture Buffer Operation

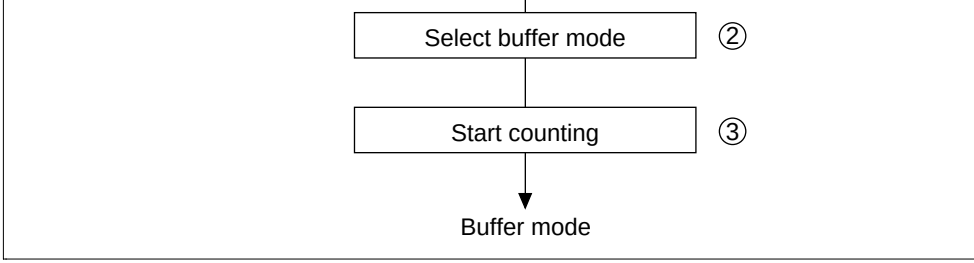


Figure 12.19 Buffer Operation Setting Procedure

Buffer Operation Examples—when TGR Is an Output Compare Register: Figure 12
shows an example of channel 0 set to PWM mode 1, and the TGRA and TGRC registers
buffer operation.

The TCNT counter is cleared by a compare-match B, and the output is a 1 upon compare-
and 0 output upon compare-match B. Because buffer mode is selected, a compare-match
changes the output, and the buffer register TGRC value is simultaneously transferred to t
general register TGRA. This operation is repeated with each occurrence of a compare-ma

See section 12.4.6, PWM Mode, for details on the PWM mode.

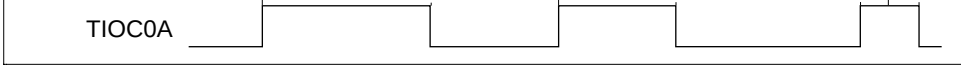


Figure 12.20 Buffer Operation Example (Output Compare Register)

Buffer Operation Examples—when TGR Is an Input Capture Register: Figure 12.2 an example of TGRA set as an input capture register with the TGRA and TGRB register buffer operation.

The TCNT counter is cleared by a TGRA register input capture, and the TIOCA pin inp input edge is selected as both rising and falling edge. Because buffer mode is selected, a capture A causes the TCNT counter value to be stored in the TGRA register, and the val was stored in the TGRA up until that time is simultaneously transferred to the TGRC reg

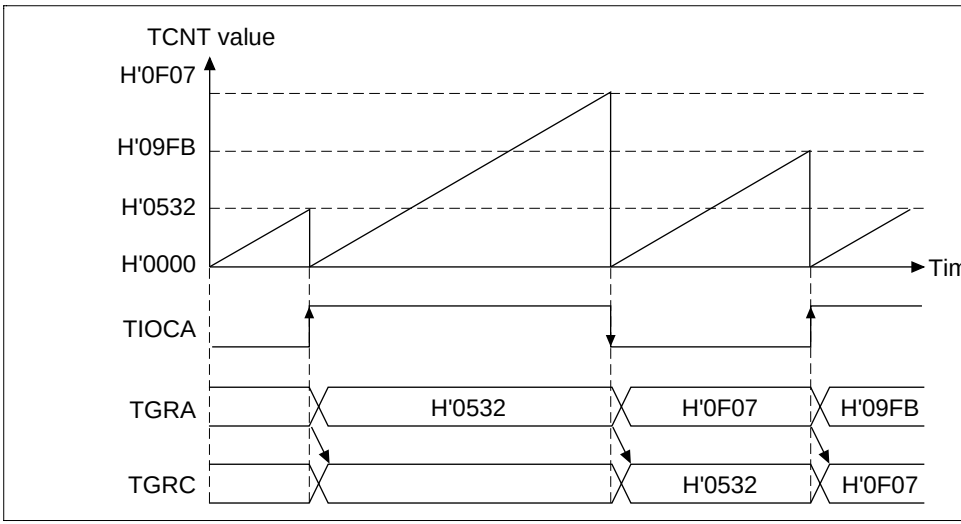


Figure 12.21 Buffer Operation Example (Input Capture Register)

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
   "count by TCNT2 overflow/underflow."
2. Set the CST bits corresponding to the upper and lower 16 bits in the TSTR to 1 to star
   count operation.

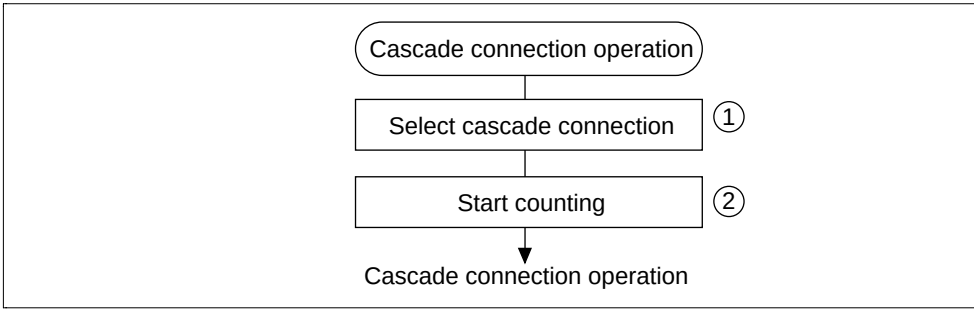


Figure 12.22 Procedure for Selecting Cascade Connection Mode

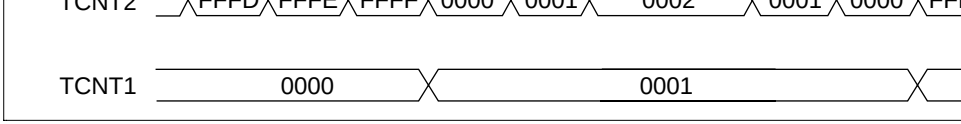


Figure 12.23 Cascade Connection Operation Example (Phase Counting Mo

12.4.6 PWM Mode

PWM mode outputs the various PWM waveforms from output pins. Output levels of 0 c
output, or toggle output can be selected as the output level for the compare-match of eac

A period can be set for a register by using the TGR compare-match as a counter clear so
five channels can be independently set to PWM mode. Synchronous operation is also po

There are two PWM modes:

- PWM mode 1

  Generates PWM output using the TGRA and TGRB registers, and TGRC and TGRD
  as pairs. The initial output values are those established in the TGRA and TGRC regis
  When the values set in TGR registers being used as a pair are equal, output values w
  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
  register. The output value of each pin upon a counter clear is the initial value establis
  TIOR register. When the values set in the period register and duty register are equal,
  values will not change even if a compare-match occurs. PWM mode 2 can be set onl
  channels 0, 1, and 2.

	TGR2B		TIOC 2B
3 (AB pair)	TGR3A TGR3B	TIOC3A	Setting not pos
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 tin 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 so
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 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.

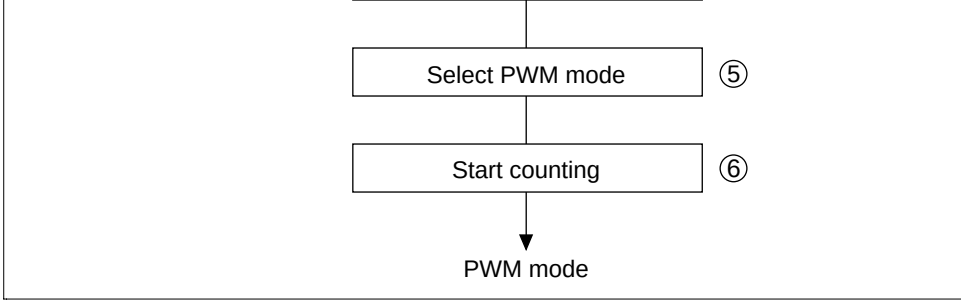


Figure 12.24 Procedure for Selecting the PWM Mode

PWM Mode Operation Examples—PWM Mode 1 (Figure 12.25): A TGRA register match is used as a TCNT counter clear source, the TGRA register initial output value an compare output value are both 0, and the TGRB register output compare output value is example, the value established in the TGRA register becomes the period and the value e in the TGRB register becomes the duty cycle.

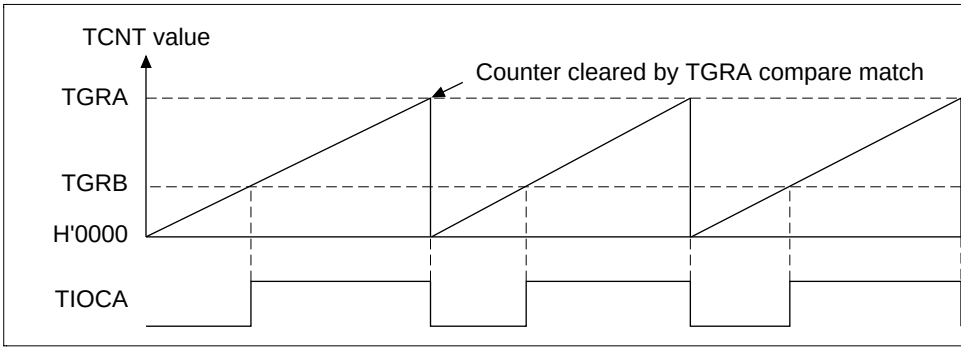


Figure 12.25 PWM Mode Operation Example (Mode 1)

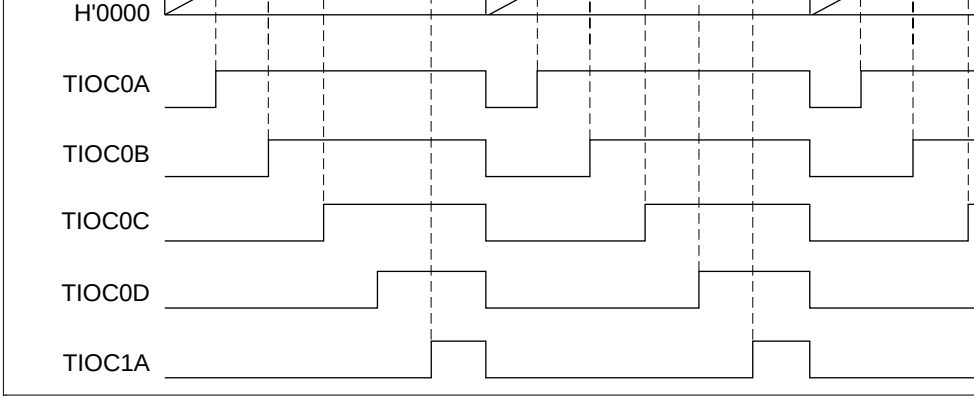


Figure 12.26 PWM Mode Operation Example (Mode 2)

0% Duty Cycle: Figure 12.27 shows an example of a 0% duty cycle PWM waveform ou
PWM mode.

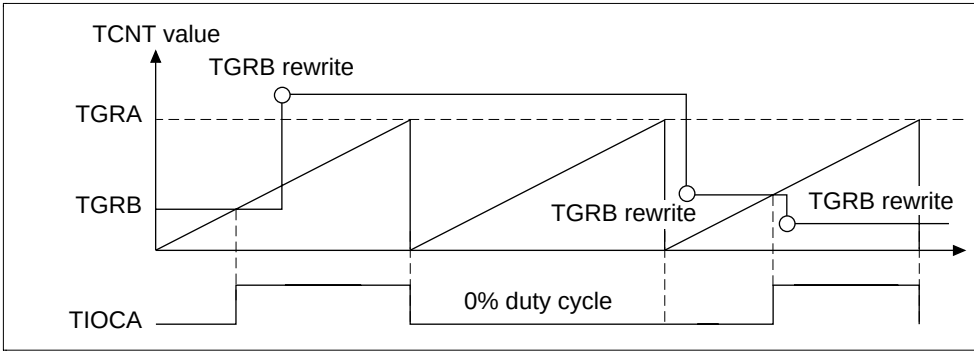


Figure 12.27 PWM Mode Operation Example (0% Duty Cycle)

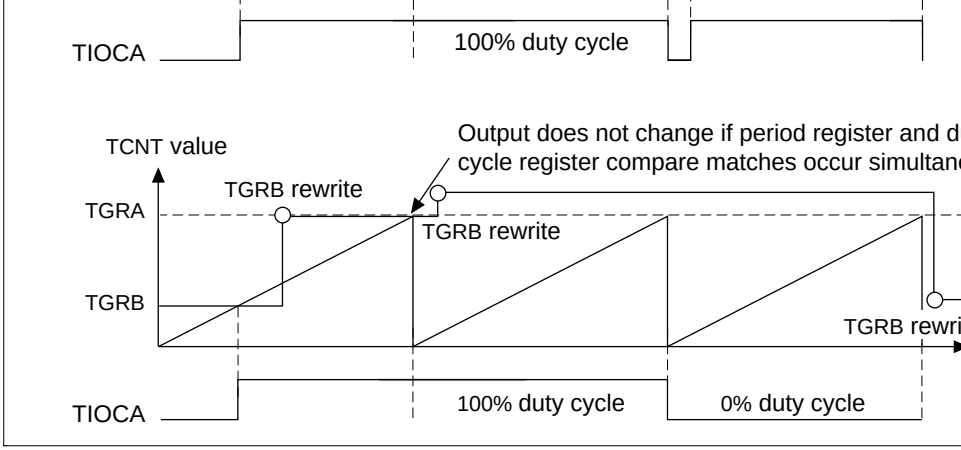


Figure 12.28 PWM Mode Operation Example (100% Duty Cycle)

12.4.7 Phase Counting Mode

The phase counting mode detects the phase differential of two external clock inputs and 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 regardless of the settings of the TPSC2–TPSC0 bits of TCR or the CKEG1 and CKEG0 TCNT also becomes an up/down counter. Since the TCR CCLR1/CCLR0 bits, TIOR, T TGR functions are all enabled, input capture and compare-match functions and interrupt can be used.

When the TCNT counter is incrementing, an overflow sets the TSR register TCFV (over flag). When it is decrementing, an underflow sets the TCFU (underflow flag).

1. Set the MD3–MD0 bits of the timer mode register (TMDR) to select the phase counti
2. Set the CST bit of the timer start register (TSTR) to 1 to start the count.

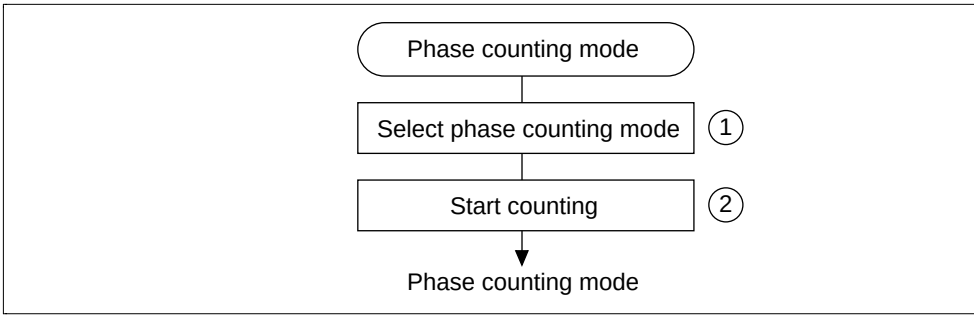


Figure 12.29 Procedure for Selecting the Phase Counting Mode

Phase Counting Operation Examples: The phase counting mode uses the phase differe between two external clocks to increment/decrement the TCNT counter. There are 4 mod depending on the count conditions.

Phase Counting Mode 1: Figure 12.30 shows an example of phase counting mode 1 ope Table 12.9 lists the up counting and down counting conditions for the TCNT.

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)	

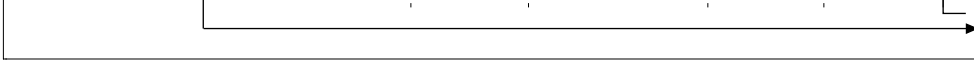


Figure 12.31 Phase Counting Mode 2 Operation

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
0 (low level)	Falling edge	Does not count (don't
Rising edge	0 (low level)	Does not count (don't
Falling edge	1 (high level)	Increment
1 (high level)	Falling edge	Does not count (don't
0 (low level)	Rising edge	Does not count (don't
Rising edge	1 (high level)	Does not count (don't
Falling edge	0 (low level)	Decrement

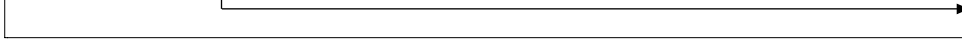


Figure 12.32 Phase Counting Mode 3 Operation

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
0 (low level)	Falling edge	Does not count (don
Rising edge	0 (low level)	Does not count (don
Falling edge	1 (high level)	Increment
1 (high level)	Falling edge	Decrement
0 (low level)	Rising edge	Does not count (don
Rising edge	1 (high level)	Does not count (don
Falling edge	0 (low level)	Does not count (don

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
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
Falling edge	0 (low level)	

Phase Counting Mode Application Example: Figure 12.34 shows an example where ch is set to phase counting mode and is teamed with channel 0 to input a two-phase encoder 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 ar the TCLKA and TCLKB pins.

Channel 0 is set so that the TCNT counter is cleared on a TGR0C register compare-match TGR0A and TGR0C registers are used with the compare-match function to establish the s control and position control periods. The TGR0B register is used with the input capture fu and the TGR0B and TGR0D registers are employed for buffer operation. The channel 1 c

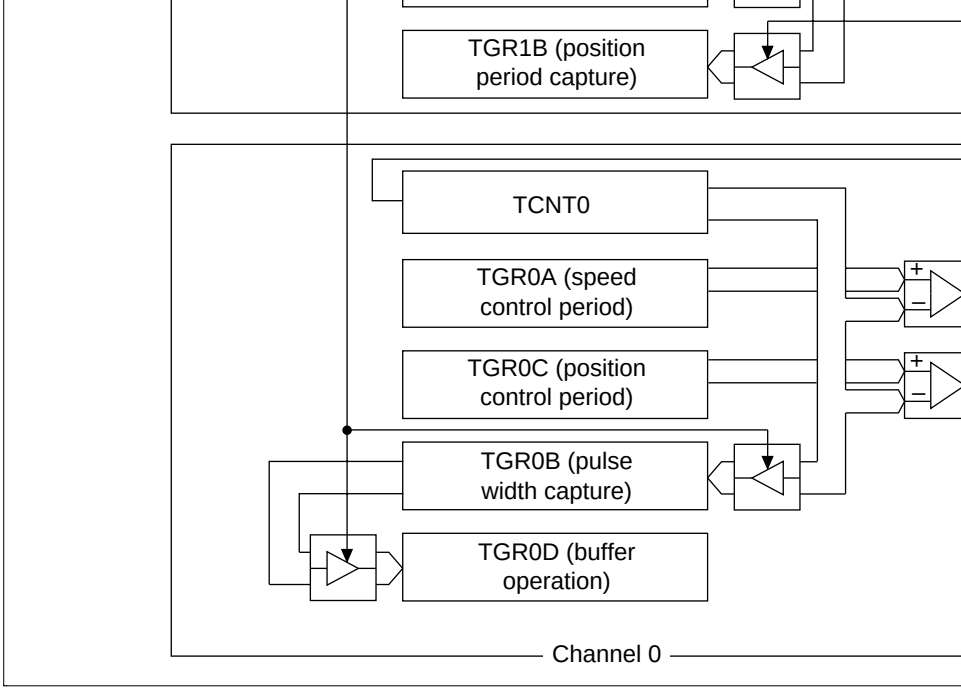


Figure 12.34 Phase Count Mode Application Example

	TIOC3D	PWM output 1' (negative-phase waveform of PWM ou
4	TIOC4A	PWM output 2
	TIOC4C	PWM output 2' (negative-phase waveform of PWM ou
	TIOC4B	PWM output 3
	TIOC4D	PWM output 3' (negative-phase waveform of PWM ou

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 p
TGR4A	Sets the turning point for PWM waveform output by the TIOC4A and TIOC4C p
TGR4B	Sets the turning point for PWM waveform output by the TIOC4B and TIOC4D p

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 rese 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 clo clock edge for channel 3.
3. Set bits CCLR2–CCLR0 in the TCR3 to select TGRA compare-match as a counter cl source.

8. Set bits MD3–MD0 in TMDR3 to B'1000 to select the reset-synchronized PWM mo
   TIOC3B, TIOC3D, TIOC4A, TIOC4B, TIOC4C, and TIOC4D become PWM outpu
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.

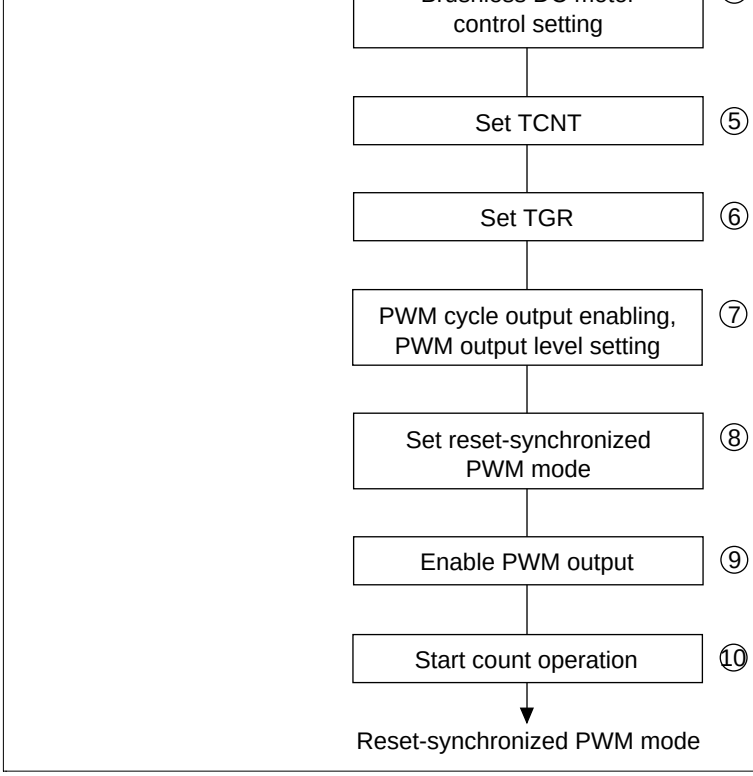


Figure 12.35 Procedure for Selecting the Reset-Synchronized PWM Mode

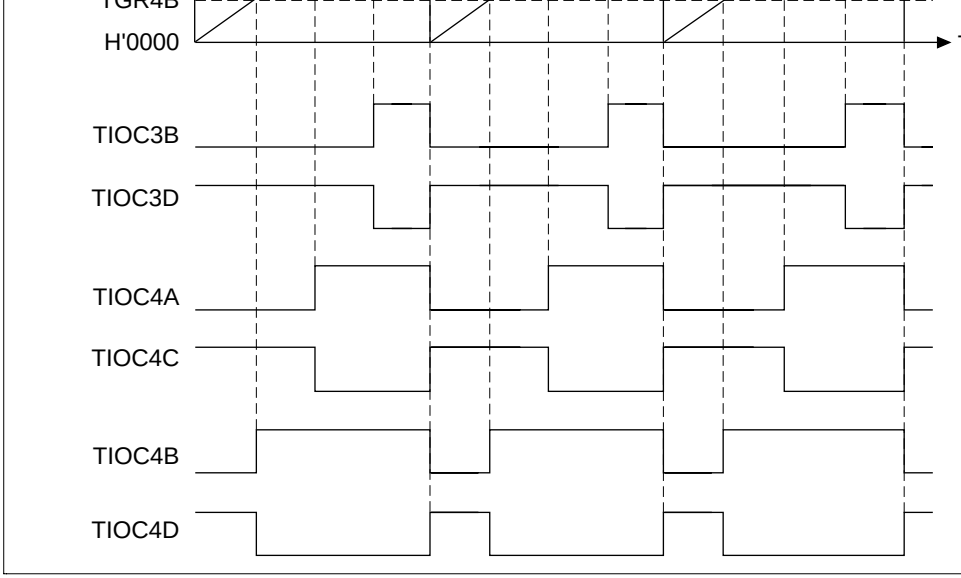


Figure 12.36 Reset-Synchronized PWM Mode Operation Example (When the T
OLSN = 1 and OLSP = 1)

Table 12.15 Output Pins for Complementary PWM Mode

Channel	Output Pin	Description
3	TIOC3A	Toggle output synchronized with PWM period (or I/O p
	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 wave PWM output 1)
4	TIOC4A	PWM output 2
	TIOC4B	PWM output 3
	TIOC4C	$\overline{\text{PWM output 2}}$ (non-overlapping negative-phase wave PWM output 2)
	TIOC4D	$\overline{\text{PWM output 3}}$ (non-overlapping negative-phase wave PWM output 3)

4	TCNT4	Up-count start, initialized to H'0000	Maskable by BSC setting*
	TGR4A	PWM output 2 compare register	Maskable by BSC setting*
	TGR4B	PWM output 3 compare register	Maskable by BSC setting*
	TGR4C	PWM output 2/TGR4A buffer register	Always readable/
	TGR4D	PWM output 3/TGR4B buffer register	Always readable/
Timer dead time data register (TDDR)		Set TCNT4 and TCNT3 offset value (dead time value)	Maskable by BSC setting*
Timer cycle data register (TCDR)		Set TCNT4 upper limit value (1/2 carrier cycle)	Maskable by BSC setting*
Timer cycle buffer register (TCBR)		TCDR buffer register	Always readable/
Subcounter (TCNTS)		Subcounter for dead time generation	Read-only
Temporary register 1 (TEMP1)		PWM output 1/TGR3B temporary register	Not readable/writ
Temporary register 2 (TEMP2)		PWM output 2/TGR4A temporary register	Not readable/writ
Temporary register 3 (TEMP3)		PWM output 3/TGR4B temporary register	Not readable/writ

Note: * Access can be enabled or disabled according to the setting of bit 13 (MTURWE BSC/BCR1 (bus controller/bus control register 1).

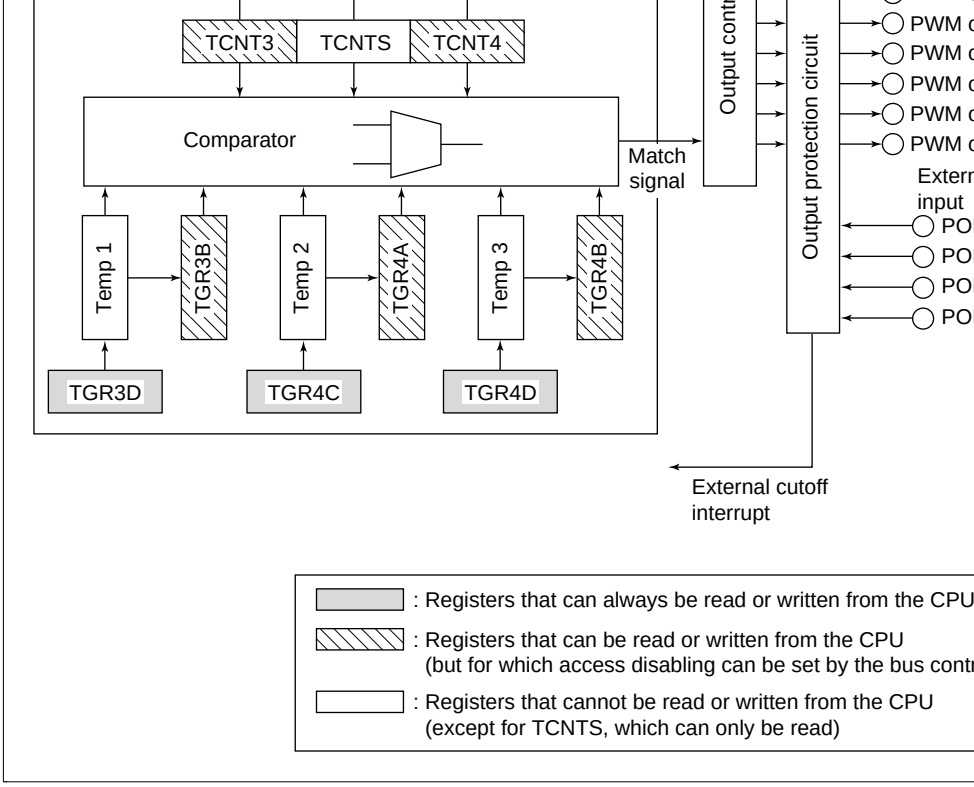


Figure 12.37 Block Diagram of Channels 3 and 4 in Complementary PWM M

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 comple PWM mode operation. In this case, synchronize the channel generating the synchron 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 buffe (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 carrie data register (TCDR) and carrier cycle buffer register (TCBR), and 1/2 the carrier cy the dead time in TGR3A and TGR3C.
8. Select enabling/disabling of toggle output synchronized with the PWM cycle using b in the timer output control register (TOCR), and set the PWM output level with bits OLSN.
9. Select complementary PWM mode in timer mode register 3 (TMDR3). Pins TIOC3A TIOC3B, TIOC3D, TIOC4A, TIOC4B, TIOC4C, and TIOC4D function as output pi set in TMDR4.
10. Set enabling/disabling of PWM waveform output pin output in the timer output mast register (TOER).
11. Set bits CST3 and CST4 in TSTR to 1 simultaneously to start the count operation.

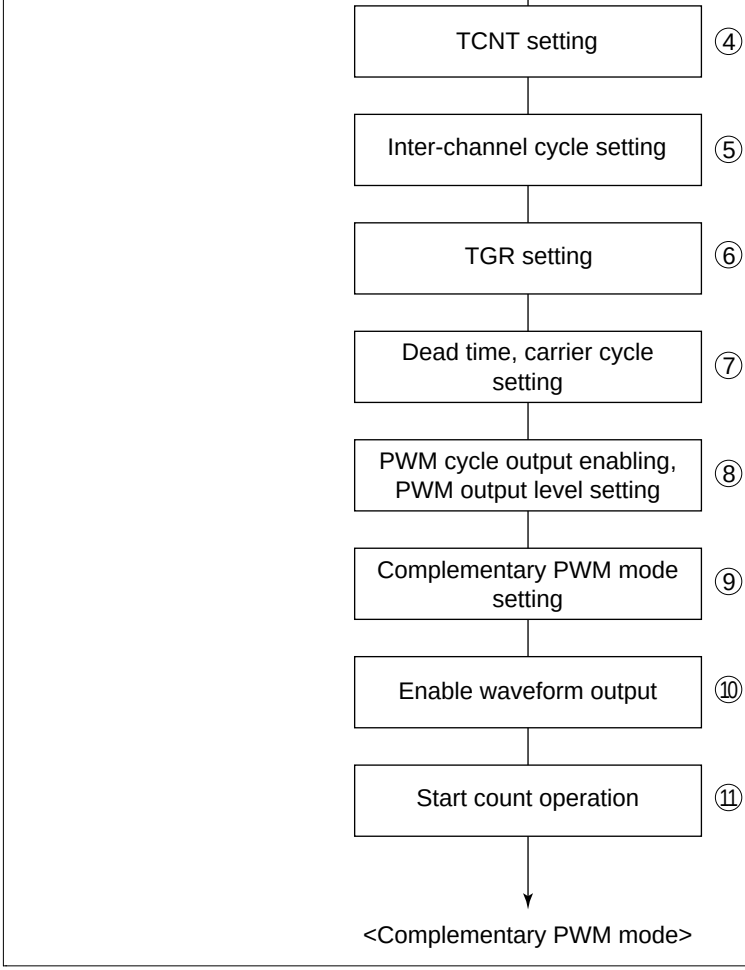


Figure 12.38 Example of Complementary PWM Mode Setting Procedure

TCNT4 is initialized to H'0000.

When the CST bit is set to 1, TCNT4 counts up in synchronization with TCNT3, and to down-counting when it matches TCDR . On reaching H'0000, TCNT4 switches to 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 is started, and when TCNTS matches TCDR, the operation switches to up-counting. TCNTS matches TGR3A, it is cleared to H'0000.

When TCNT4 matches TDDR during TCNT3 and TCNT4 down-counting, up-count started, and when TCNTS matches TDDR, the operation switches to down-counting 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 F is set during the count operation only.

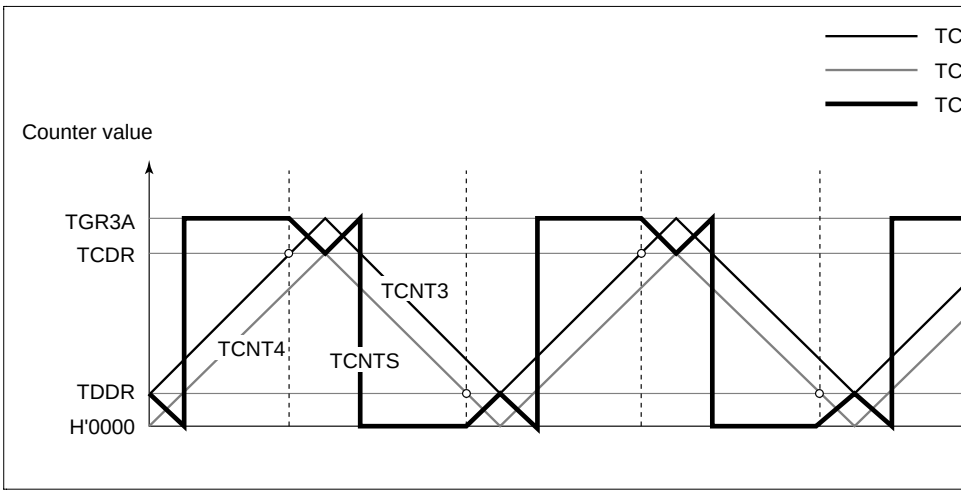


Figure 12.39 Complementary PWM Mode Counter Operation

The data written to a buffer register is constantly transferred to the temporary register interval. Data is not transferred to the temporary register in the Tb interval. Data writt buffer register in this interval is transferred to the temporary register at the end of the interval.

The value transferred to a temporary register is transferred to the compare register wh TCNTS for which the Tb interval ends matches TGR3A when counting up, or H'0000 counting down. The timing for transfer from the temporary register to the compare reg be selected with bits MD3–MD0 in the timer mode register (TMDR). Figure 12.40 sh 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 performed, the temporary register has the same function as the compare register, and i compared with the counter. In this interval, therefore, there are two compare registers phase output, with the compare register containing the pre-change data, and the tempo register containing the new data. In this interval, the three counters—TCNT3, TCNT4 TCNTS—and two registers—compare register and temporary register—are compared PWM output controlled accordingly.

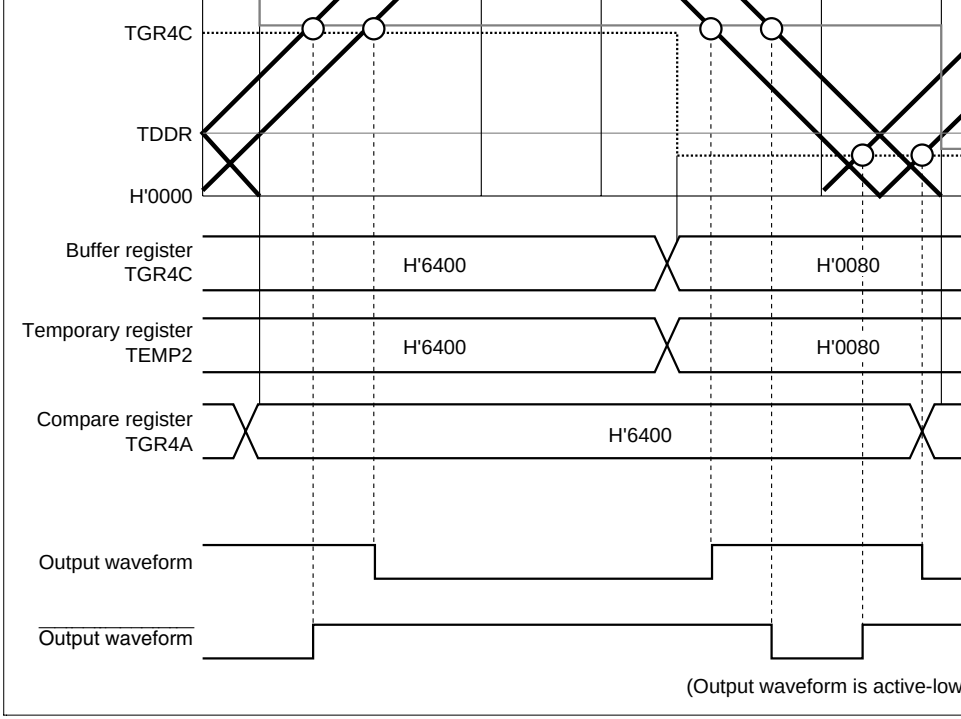


Figure 12.40 Example of Complementary PWM Mode Operation

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 a time Td set in TDDR.

- PWM output level setting

  In complementary PWM mode, the PWM pulse output level is set with bits OLSN an in the timer output control register (TOCR).

  The output level can be set for each of the three positive phases and three negative ph phase output.

  Complementary PWM mode should be cleared before setting or changing output leve

- Dead time setting

  In complementary PWM mode, PWM pulses are output with a non-overlapping relati between the positive and negative phases. This non-overlap time is called the dead tim

  The non-overlap time is set in the timer dead time data register (TDDR). The value se TDDR is used as the TCNT3 counter start value, and creates non-overlap between TC TCNT4. Complementary PWM mode should be cleared before changing the contents TDDR.

the crest, and from the current cycle when performed in the trough. Figure 12.41 illu
operation when the PWM cycle is updated at the crest.

See the following section, Register data updating, for the method of updating the dat
buffer register.

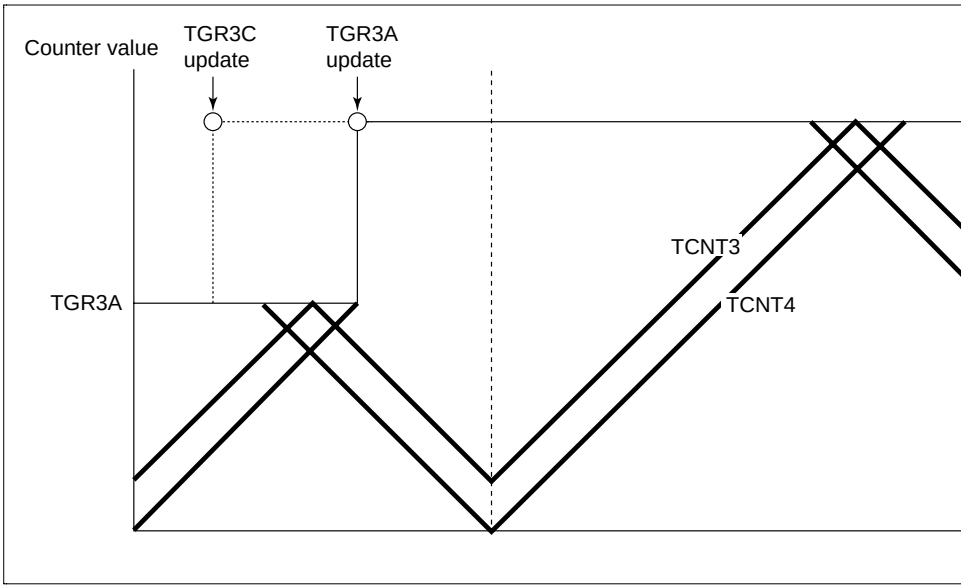


Figure 12.41 Example of PWM Cycle Updating

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 o
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
when not updating all five registers, or when updating the TGR4D data. In this case, t
written to TGR4D should be the same as the data prior to the write operation.

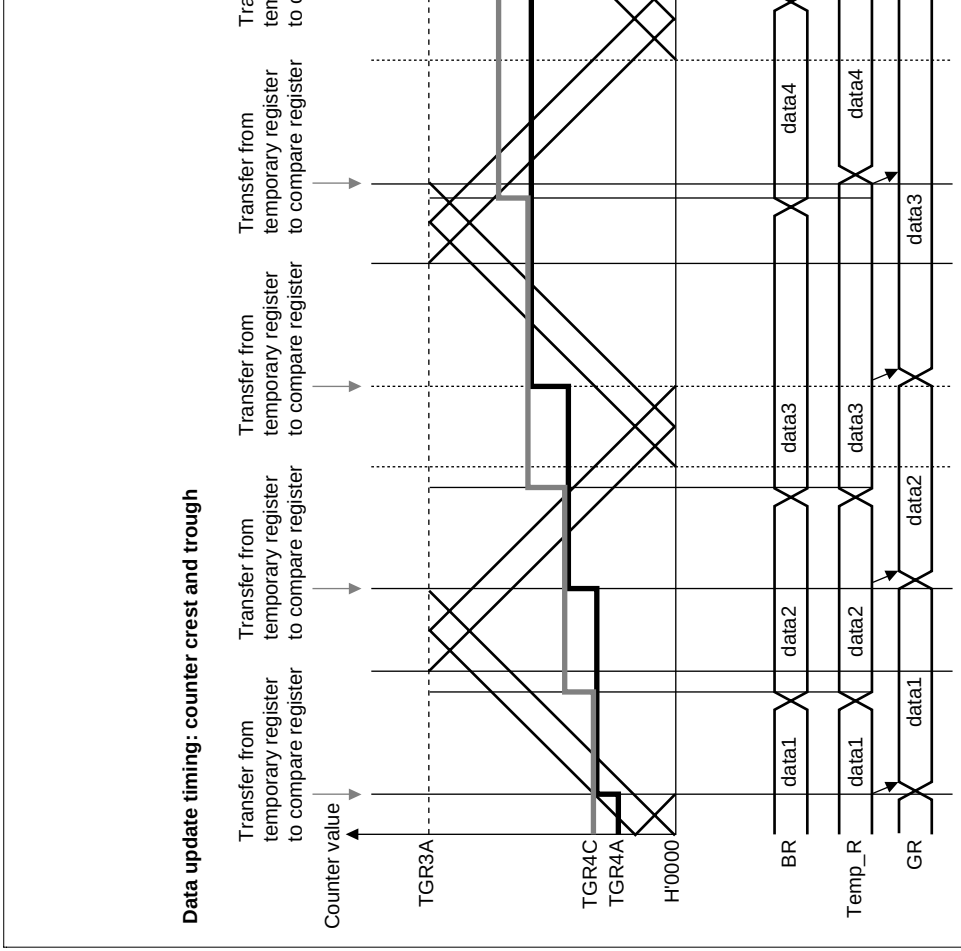


Figure 12.42 Example of Data Update in Complementary PWM Mode

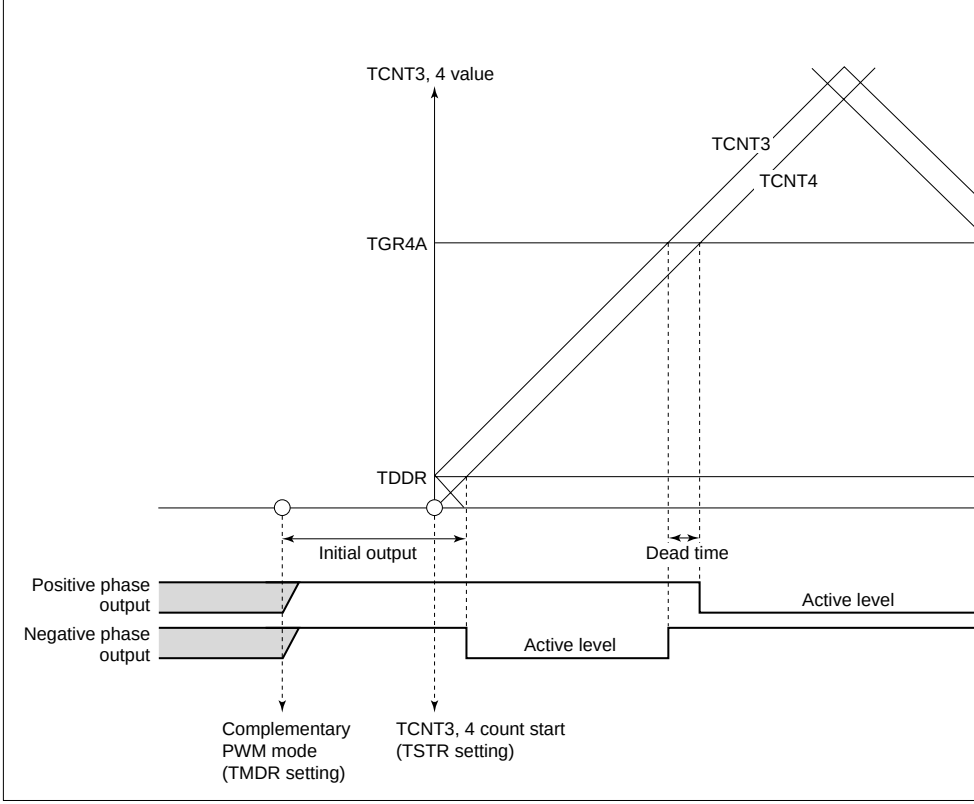


Figure 12.43 Example of Initial Output in Complementary PWM Mode (1)

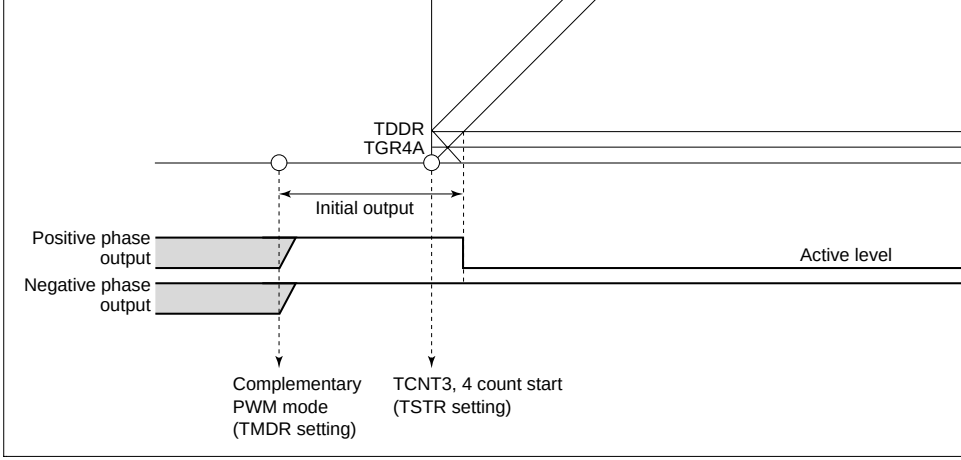


Figure 12.44 Example of Initial Output in Complementary PWM Mode (

The positive phase/negative phase off timing is generated by a compare-match with th
line counter, and the on timing by a compare-match with the dotted-line counter opera
a delay of the dead time behind the solid-line counter. In the T1 period, compare-matc
turns off the negative phase has the highest priority, and compare-matches occurring p
are ignored. In the T2 period, compare-match **c** that turns off the positive phase has th
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**
as shown in figure 12.45.

If compare-matches deviate from the **a** → **b** → **c** → **d** order, since the time for which
negative phase is off is less than twice the dead time, the figure shows the positive pha
being turned on. If compare-matches deviate from the **c** → **d** → **a'** → **b'** order, since
for which the positive phase is off is less than twice the dead time, the figure shows th
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 **c**
because turning off of the positive phase has priority due to the occurrence of compare
(positive phase off timing) before compare-match **b** (positive phase on timing) (conse
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 occurr
c, which turns of the positive phase, are ignored. As a result, the positive phase is not
on.

Thus, in complementary PWM mode, compare-matches at turn-off timings take prece
and turn-on timing compare-matches that occur before a turn-off timing compare-mat
ignored.

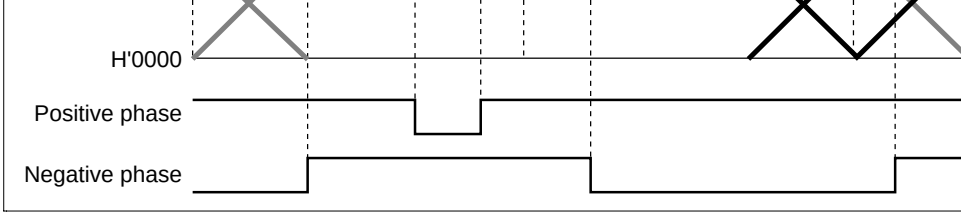


Figure 12.45 Example of Complementary PWM Mode Waveform Output

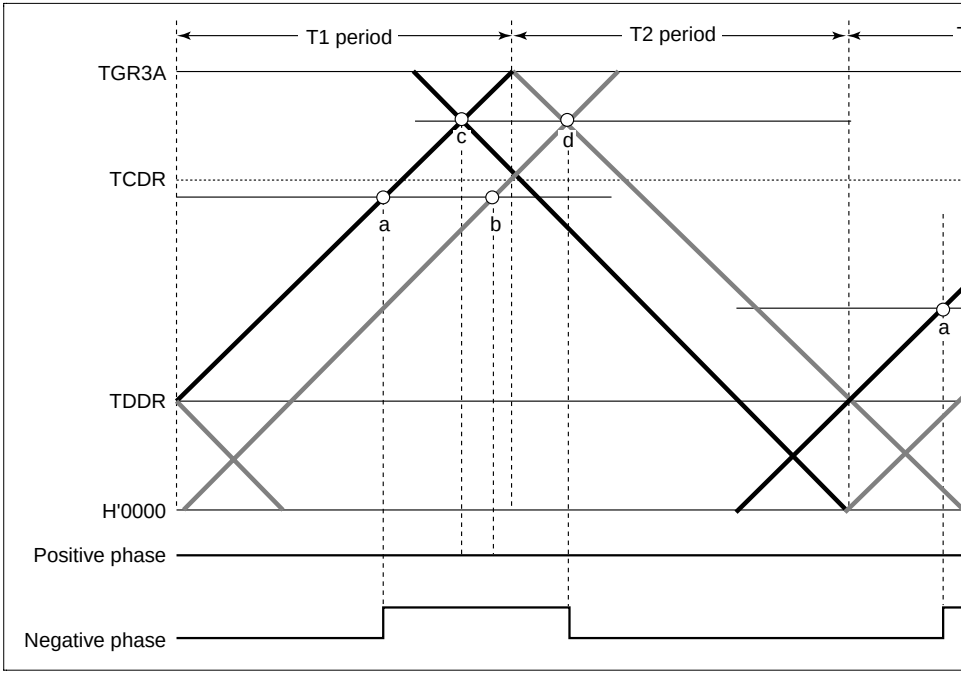


Figure 12.46 Example of Complementary PWM Mode Waveform Output

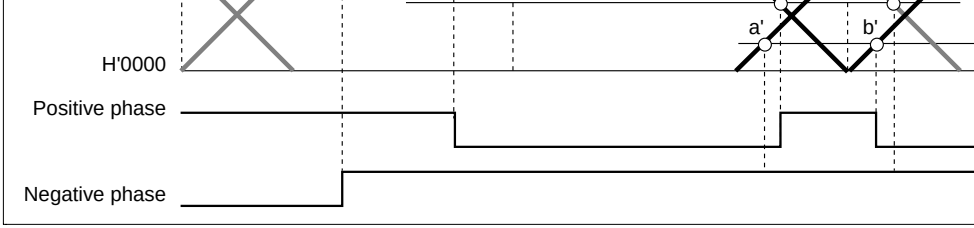


Figure 12.47 Example of Complementary PWM Mode Waveform Output (

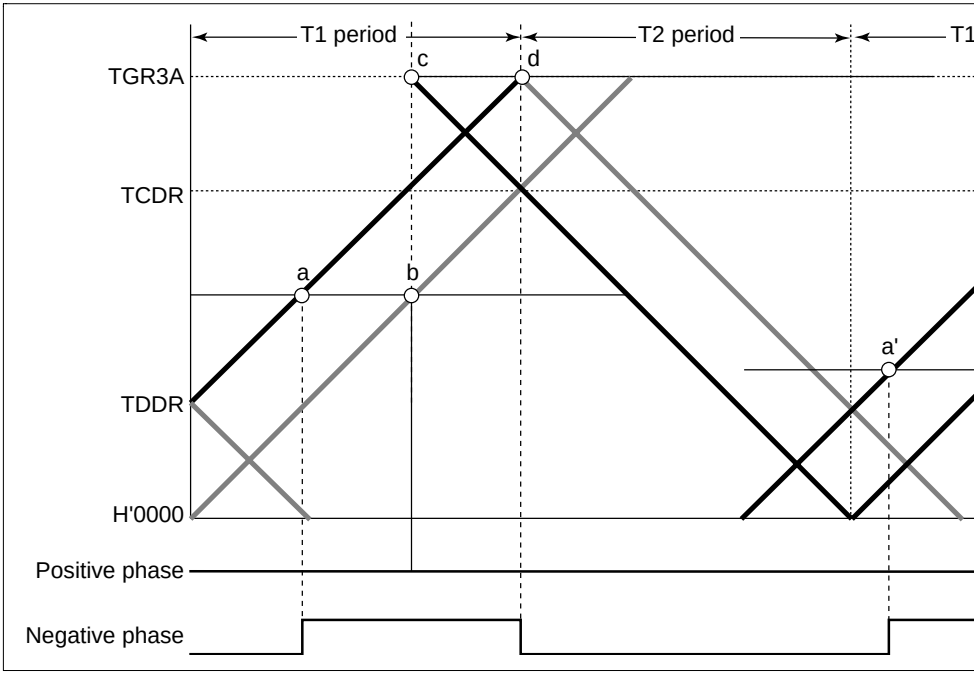


Figure 12.48 Example of Complementary PWM Mode 0% and 100% Waveform O

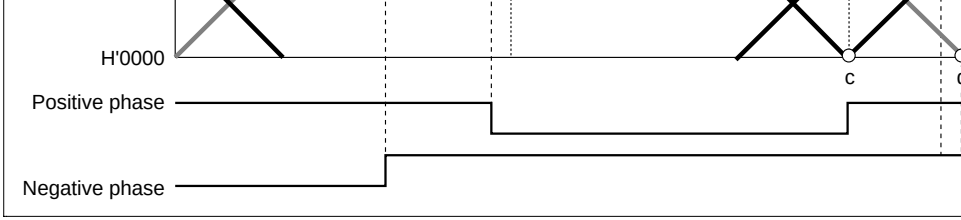


Figure 12.49 Example of Complementary PWM Mode 0% and 100% Waveform (

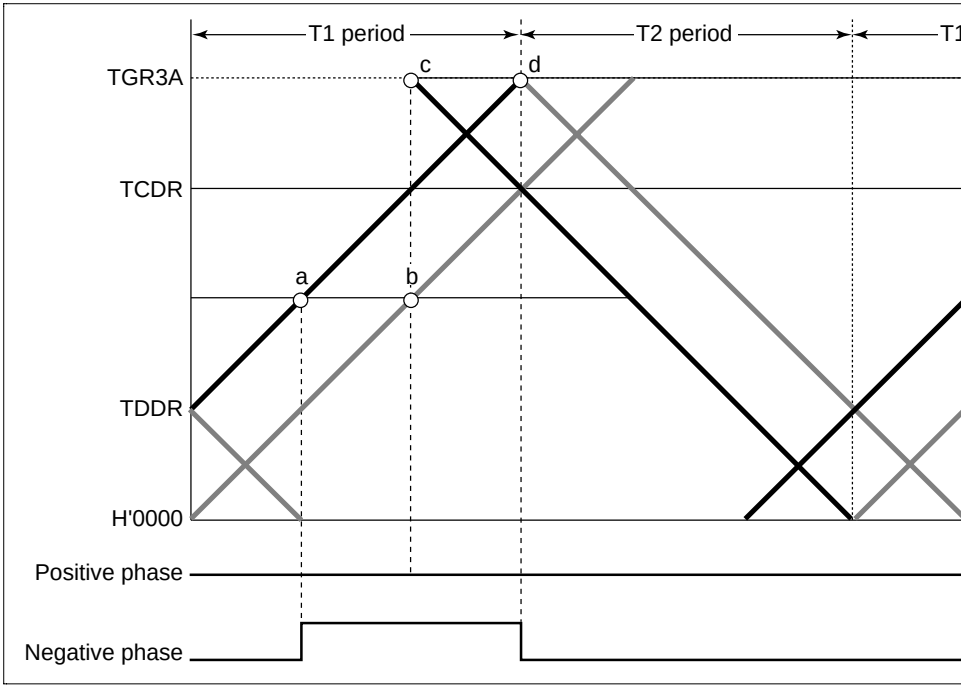


Figure 12.50 Example of Complementary PWM Mode 0% and 100% Waveform (

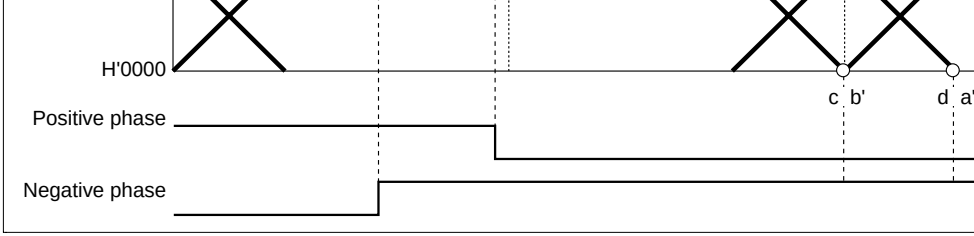


Figure 12.51 Example of Complementary PWM Mode 0% and 100% Waveform O

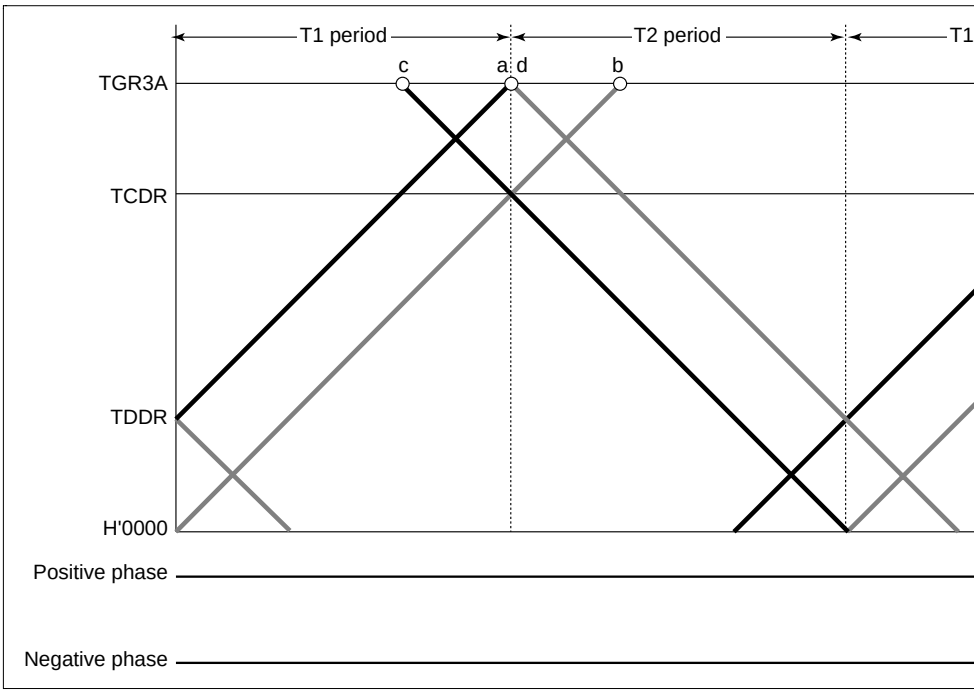


Figure 12.52 Example of Complementary PWM Mode 0% and 100% Waveform O

In complementary PWM mode, toggle output can be performed in synchronization w
PWM carrier cycle by setting the PSYE bit to 1 in the timer output control register (T
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 comp
between TCNT4 and H'0000.

The output pin for this toggle output is the TIOC3A pin. The initial output is 1.

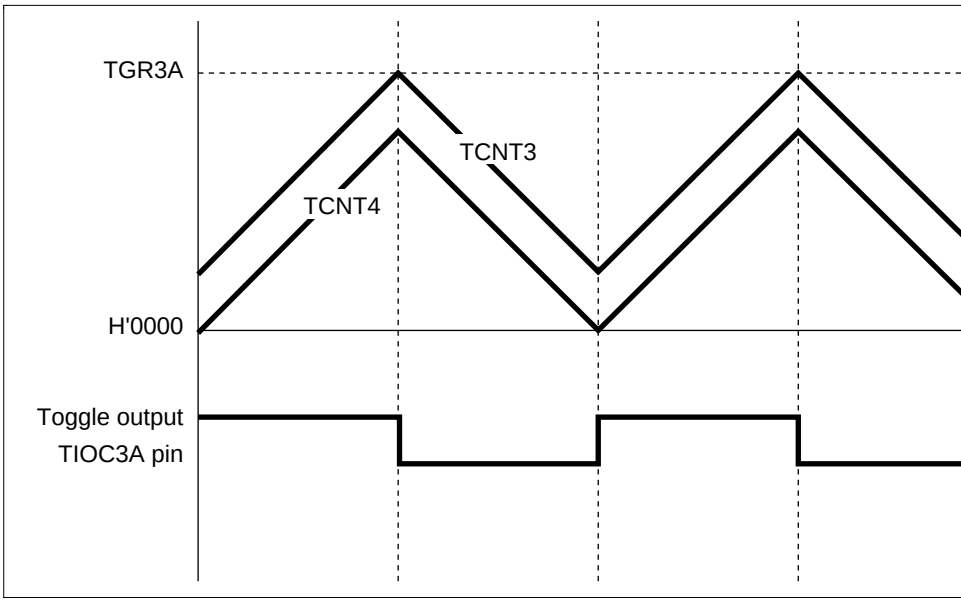


Figure 12.53 Example of Toggle Output Waveform Synchronized with PWM C

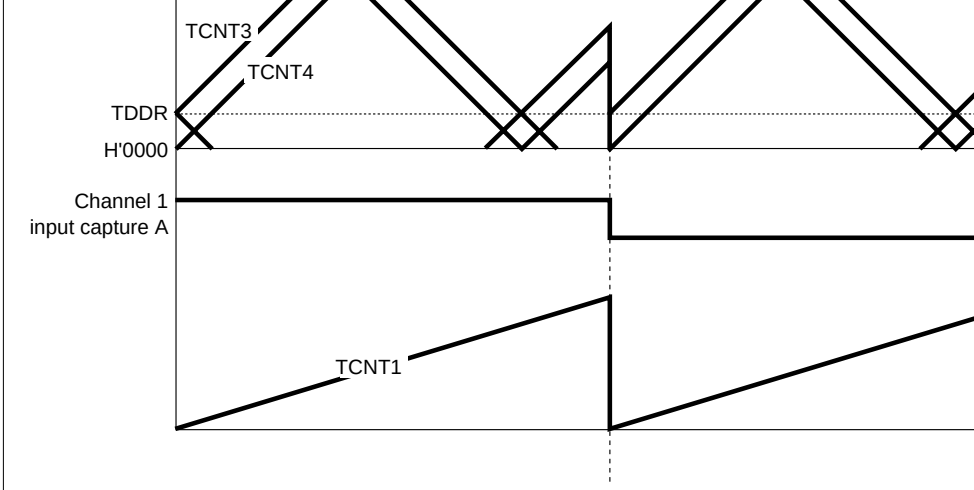


Figure 12.54 Counter Clearing Synchronized with Another Channel

With this 6-phase output, in the case of on output, it is possible to use complementar mode output and perform chopping output by setting the N bit or P bit to 1. When th P bit is 0, level output is selected.

The 6-phase output active level (on output level) can be set with the OLSN and OLS the timer output control register (TOCR) regardless of the setting of the N and P bits using this mode, set the 6-phase output waveform to High active (Low active is also for A masks).

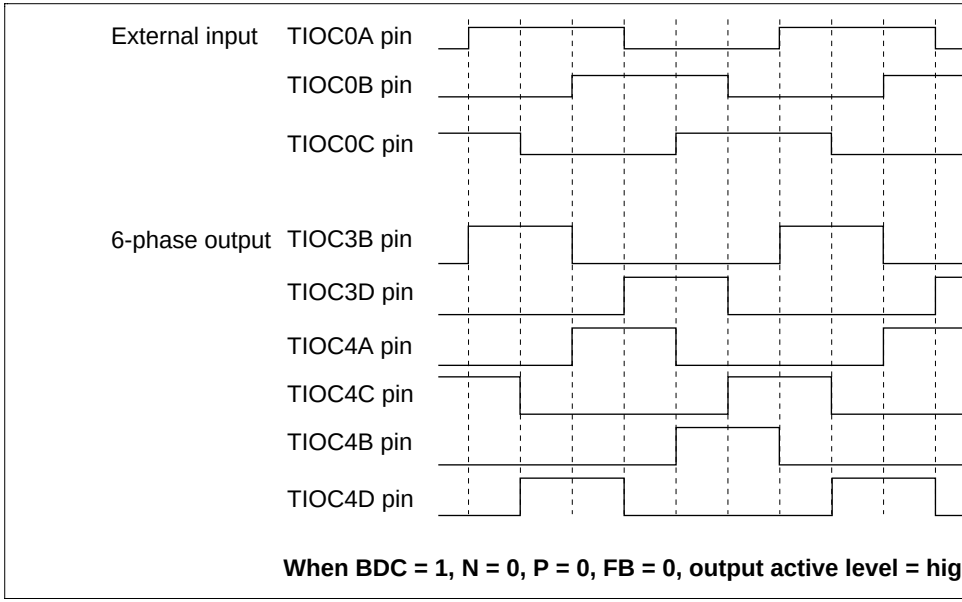


Figure 12.55 Example of Output Phase Switching by External Input (1)

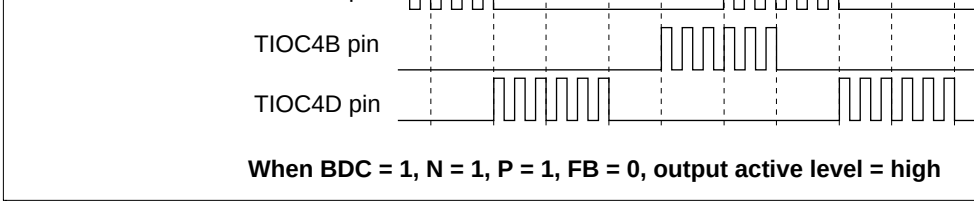


Figure 12.56 Example of Output Phase Switching by External Input (2)

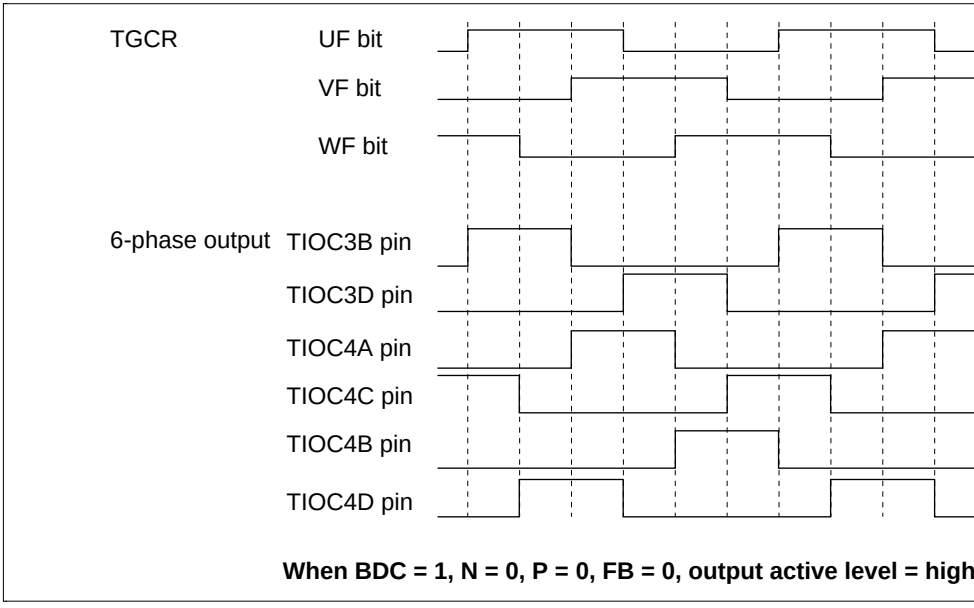
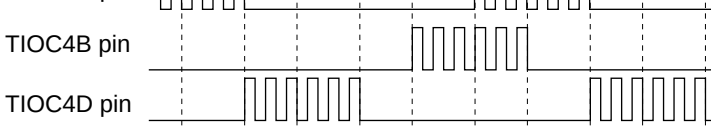


Figure 12.57 Example of Output Phase Switching by Means of UF, VF, WF Bit Set



When BDC = 1, N = 1, P = 1, FB = 0, output active level = high

Figure 12.58 Example of Output Phase Switching by Means of UF, VF, WF Bit S

- A/D conversion start request setting

  In complementary PWM mode, an A/D conversion start request can be issued using 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 the center of the PWM pulse.

  A/D conversion start requests can be set by setting the TTGE bit to 1 in the timer int enable register (TIER).

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 pins automatically go to the high-impedance state. The pin states are not guaranteed w clock is restarted.

  See section 4.4, Oscillator Halt Function, for details.

interrupt request is canceled by clearing the status flag to 0.

The channel priority order can be changed with the interrupt controller. The priority ranl
within a channel is fixed. For more information, see section 6, Interrupt Controller (INT

Table 12.17 lists the MTU interrupt sources.

Input Capture/Compare Match Interrupts: If the TGIE bit of the timer input enable r
(TIER) is already set to 1 when the TGF flag in the timer status register (TSR) is set to 1
register input capture/compare-match of any channel, an interrupt request is sent to the i
controller. The interrupt request is canceled by clearing the TGF flag to 0. The MTU has
capture/compare-match interrupts; four each for channels 0, 3, and 4, and two each for c
and 2.

Overflow Interrupts: If the TCIEV bit of the TIER is already set to 1 when the TCFV
TSR is set to 1 by a TCNT counter overflow of any channel, an interrupt request is sent
interrupt controller. The interrupt request is canceled by clearing the TCFV flag to 0. Th
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
the TSR is set to 1 by a TCNT counter underflow of any channel, an interrupt request is
interrupt controller. The interrupt request is canceled by clearing the TCFU flag to 0. Th
has two underflow interrupts, one each for channels 1 and 2.

	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	L

Note: * Indicates the initial status following reset. The ranking of channels can be altered the interrupt controller.

The MTU has 5 TGRA register input capture/compare-match interrupts, one for any cha
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
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
TGRA register input capture/compare-match of any of the channels, an A/D conversion
request is sent to the A/D converter. If the MTU conversion start trigger is selected at su
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 ch
can be used as A/D converter activation sources.

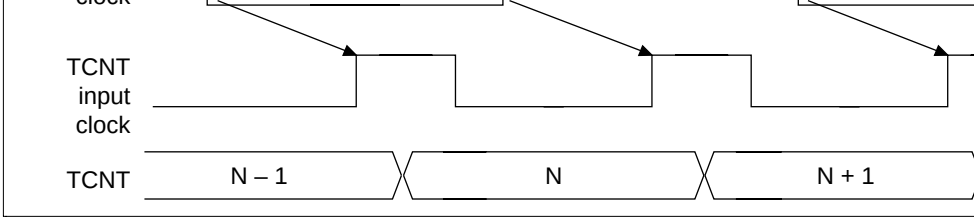


Figure 12.59 TCNT Count Timing during Internal Clock Operation

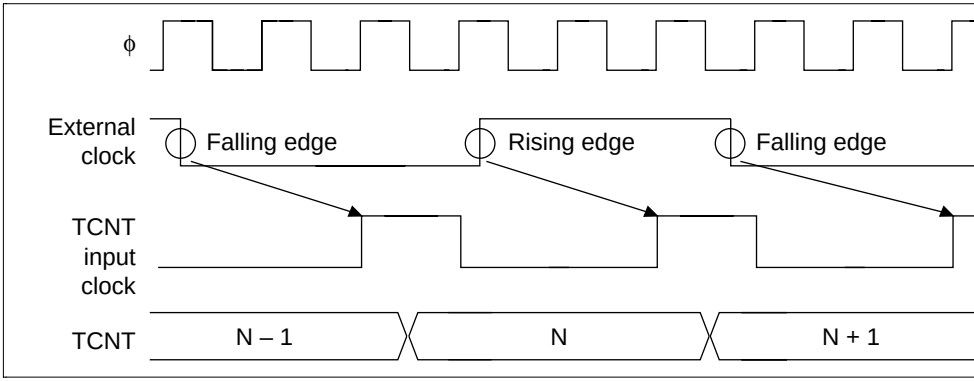


Figure 12.60 TCNT Count Timing during External Clock Operation (Normal M

Output Compare Output Timing: The compare-match signal is generated at the final
TCNT and TGR matching. When a compare-match signal is issued, the output value set
or TOCR is output to the output compare output pin (TIOC pin). After TCNT and TGR
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
figure 12.63 for output compare output timing in complementary PWM mode and reset
PWM mode.

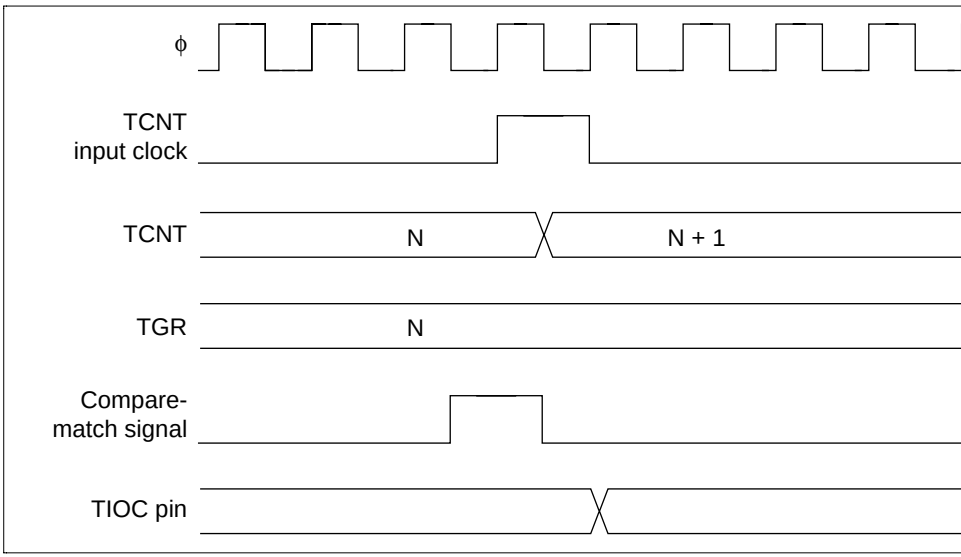


Figure 12.62 Output Compare Output Timing (Normal Mode/PWM Mod

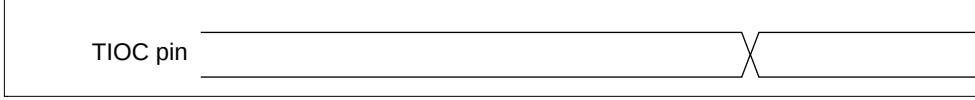


Figure 12.63 Output Compare Output Timing (Complementary PWM Mode/Res PWM Mode)

Input Capture Signal Timing: Figure 12.64 illustrates input capture timing.

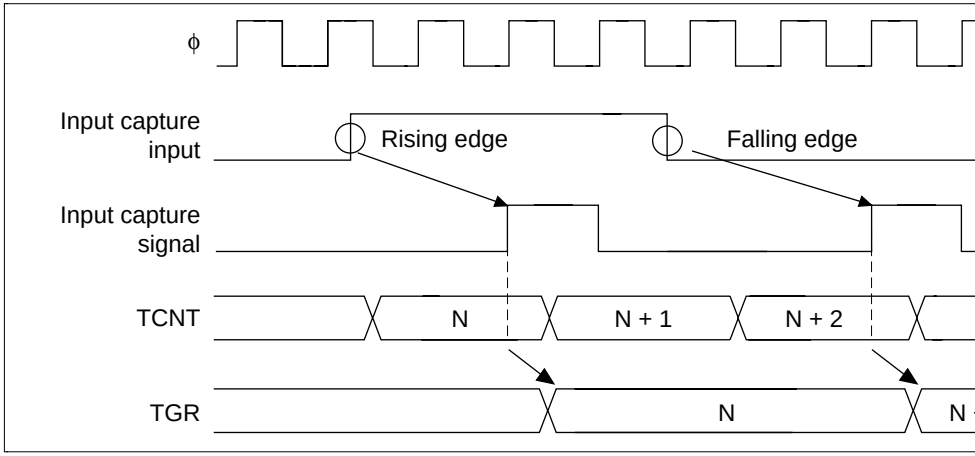


Figure 12.64 Input Capture Input Signal Timing

Figure 12.65 Counter Clearing Timing (Compare-Match)

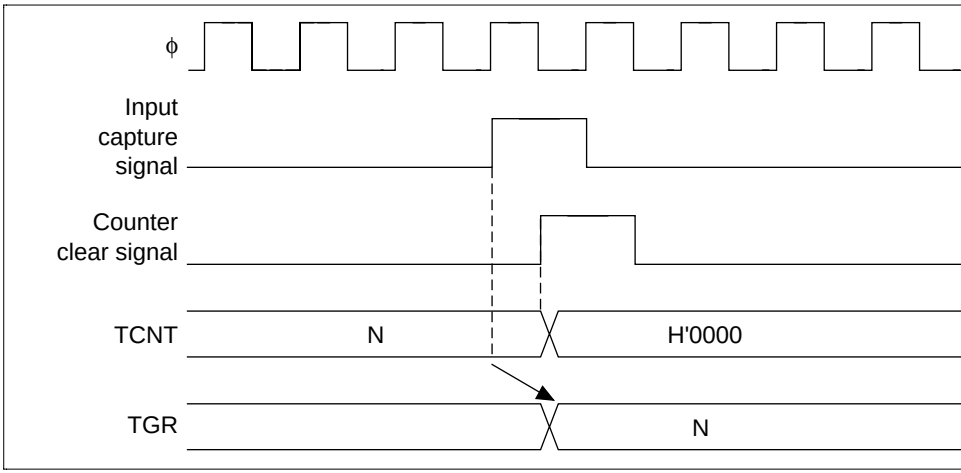


Figure 12.66 Counter Clearing Timing (Input Capture)

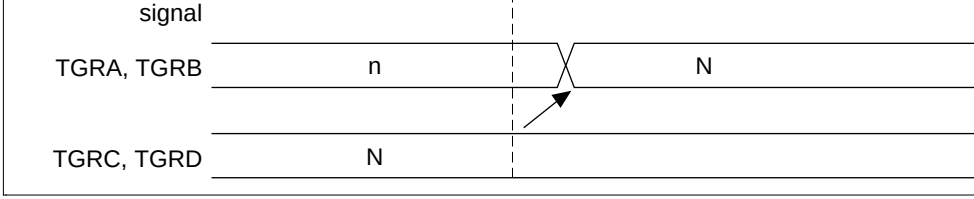


Figure 12.67 Buffer Operation Timing (Compare-Match)

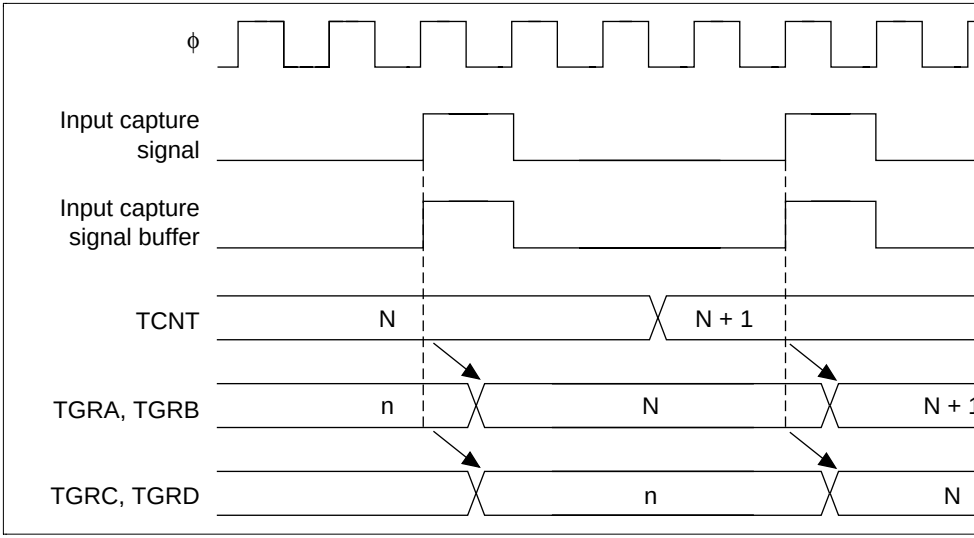


Figure 12.68 Buffer Operation Timing (Input Capture)

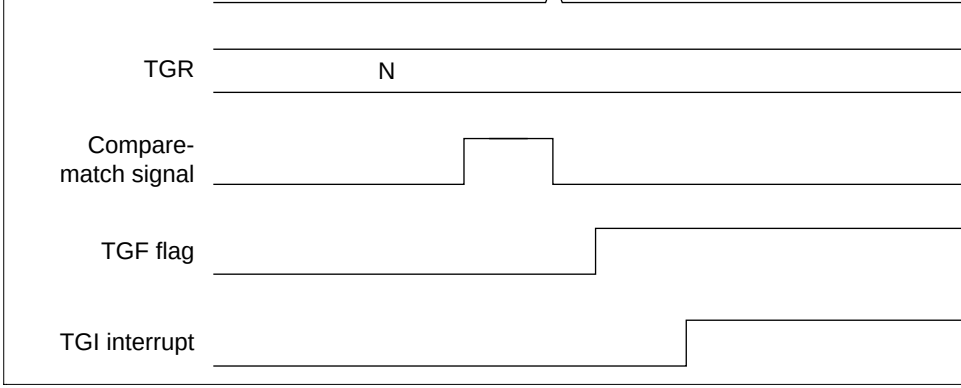


Figure 12.69 TGI Interrupt Timing (Compare Match)

Setting TGF Flag Timing during Input Capture: Figure 12.70 shows timing for the T
of the timer status register (TSR) due to input capture, as well as TGI interrupt request s
timing.

Figure 12.70 TGI Interrupt Timing (Input Capture)

Setting Timing for Overflow Flag (TCFV)/Underflow Flag (TCFU): Figure 12.71 sho timing for the TCFV flag of the timer status register (TSR) due to overflow, as well as TC interrupt request signal timing. Figure 12.72 shows timing for the TCFU flag of the timer register (TSR) due to underflow, as well as TCIU interrupt request signal timing. Figure 1 shows timing for the TCFV flag of TSR4 due to underflow in complementary PWM mod as TCIV interrupt request signal timing.

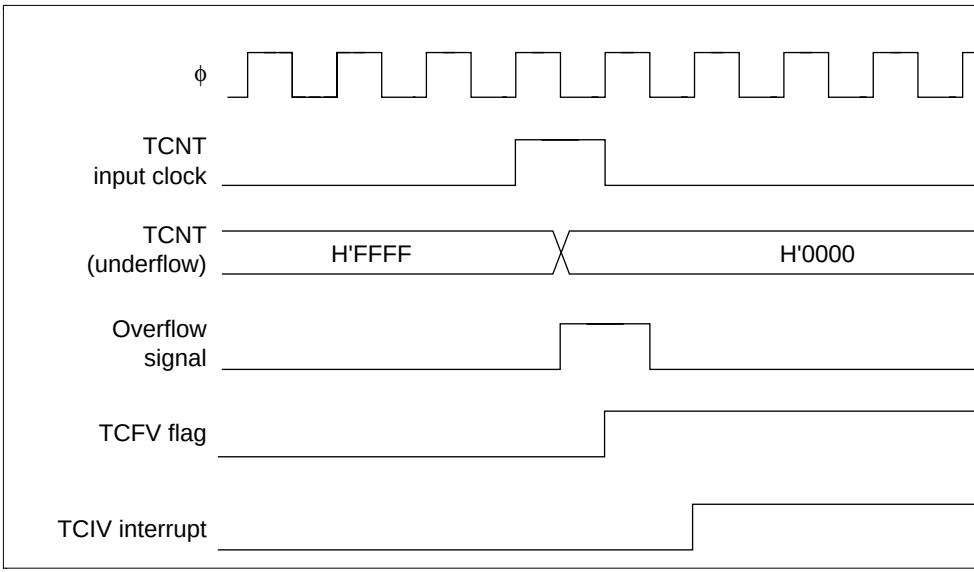


Figure 12.71 TCIV Interrupt Setting Timing

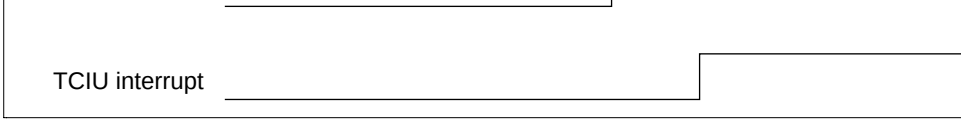


Figure 12.72 TCIU Interrupt Setting Timing

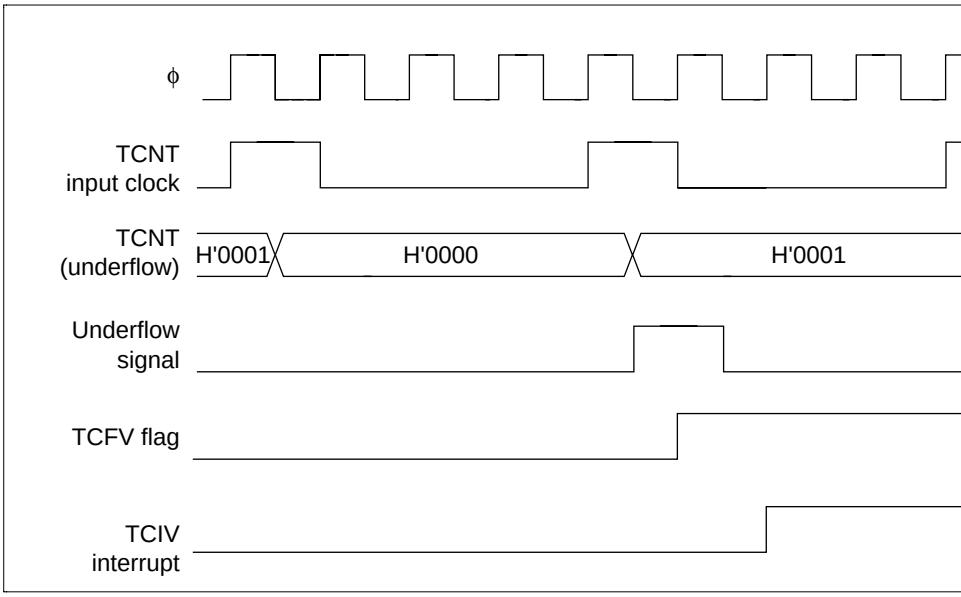


Figure 12.73 TCIV Interrupt Setting Timing (TSR4, Complementary PWM M

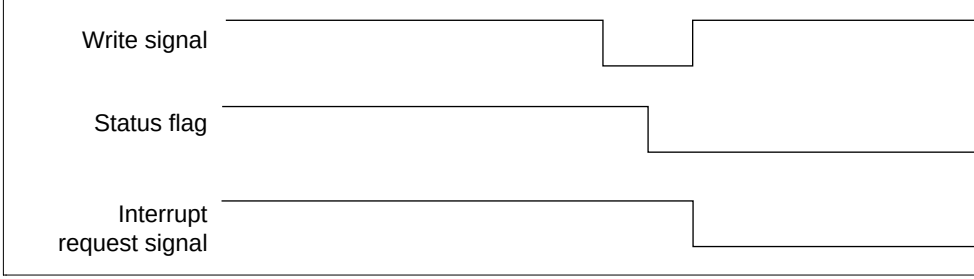


Figure 12.74 Timing of Status Flag Clearing by the CPU

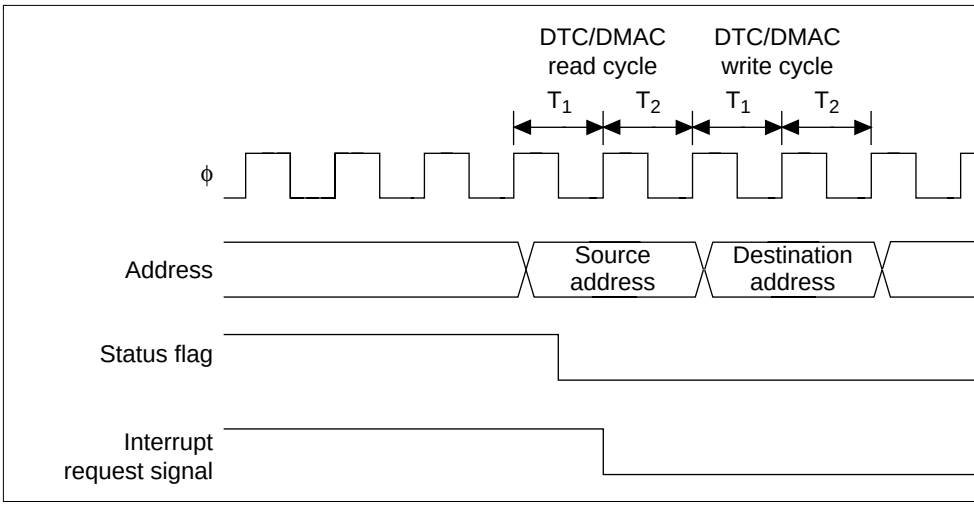
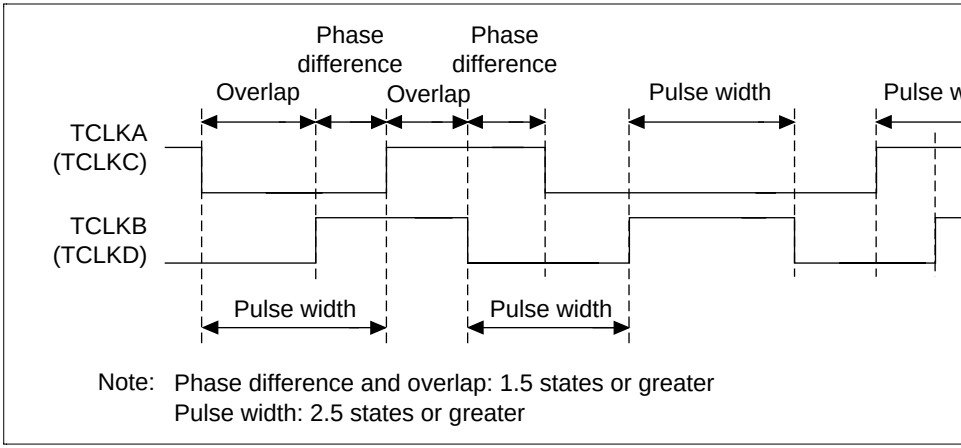


Figure 12.75 Timing of Status Flag Clearing by DTC/DMAC Activation



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 M

12.7.2 Note on Cycle Setting

When setting a counter clearing by compare-match, clearing is done in the final state wh
matches the TGR value (update timing for count value on TCNT match). The actual nur
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)

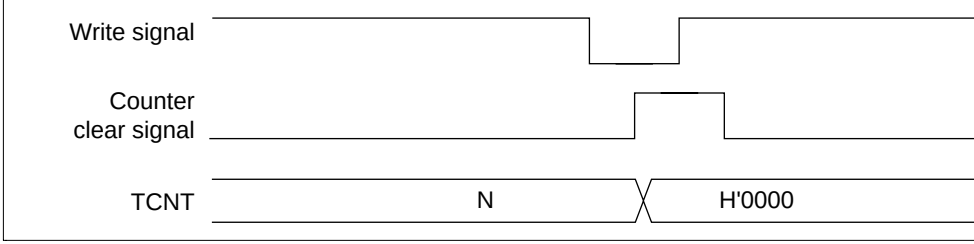


Figure 12.77 TCNT Write and Clear Contention

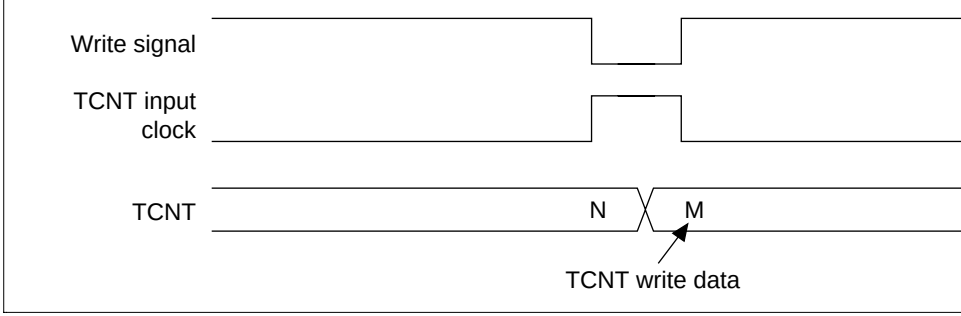


Figure 12.78 TCNT Write and Increment Contention

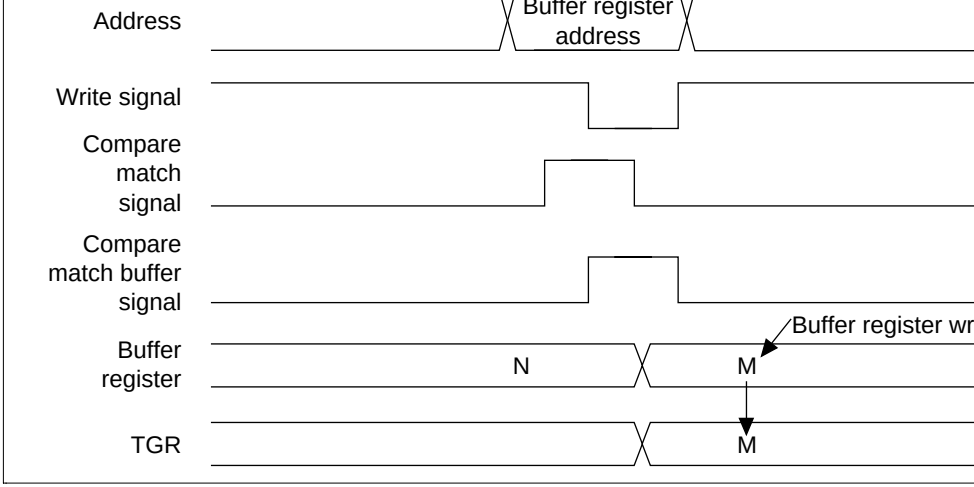


Figure 12.79 TGR Write and Compare-Match Contention (Channel 0)

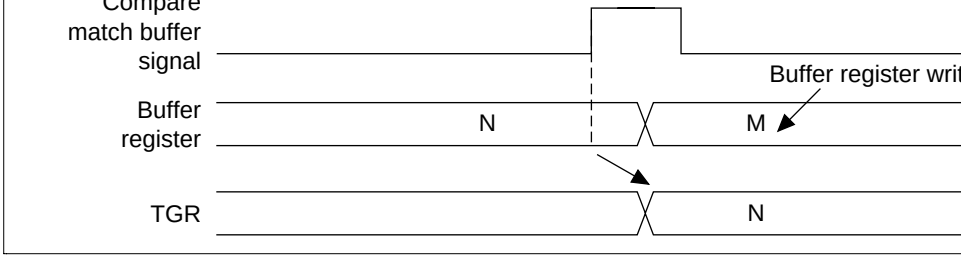


Figure 12.80 TGR Write and Compare-Match Contention (Channels 3 and

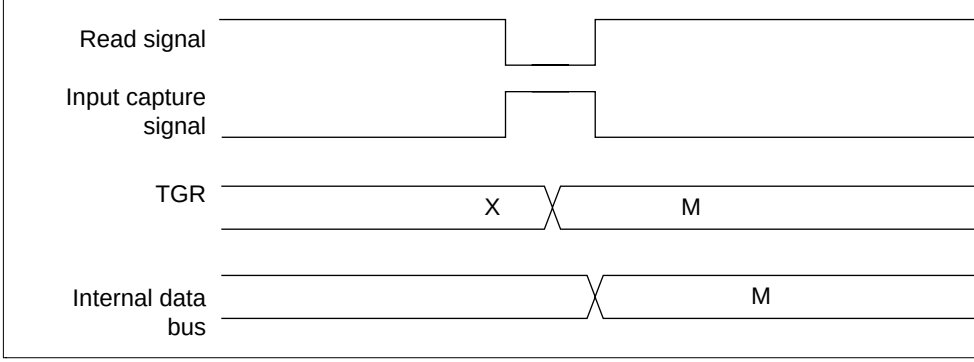


Figure 12.81 TGR Read and Input Capture Contention

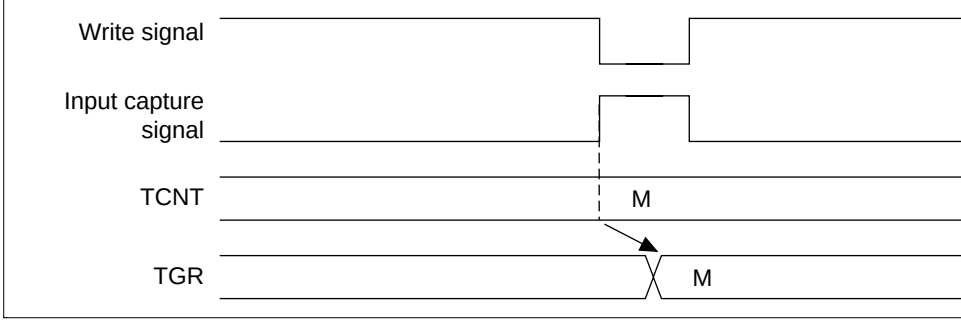


Figure 12.82 TGR Write and Input Capture Contention

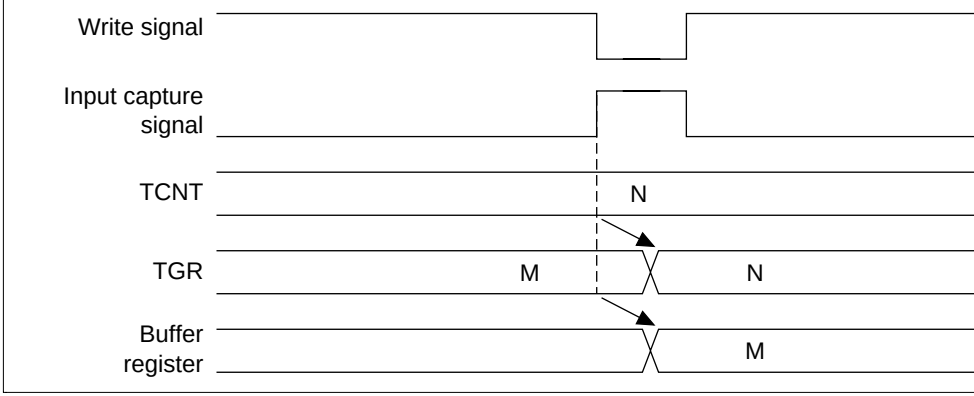


Figure 12.83 Buffer Register Write and Input Capture Contention

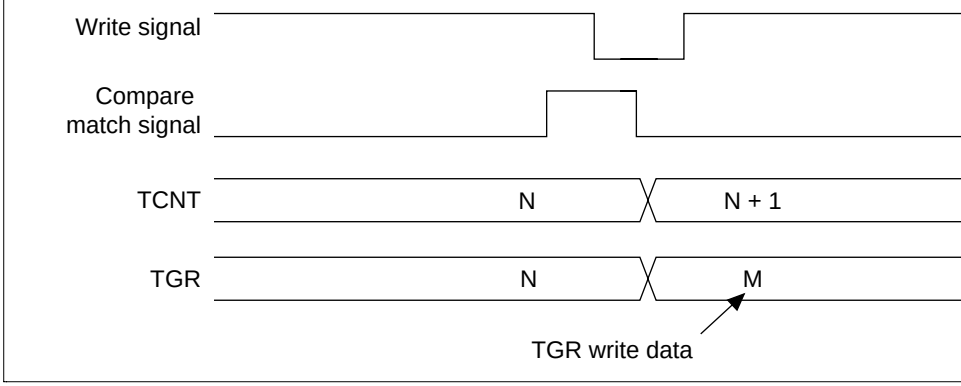


Figure 12.84 TGR Write and Compare Match Contention

12.7.10 TCNT2 Write and Overflow/Underflow Contention in Cascade Connectio

With timer counters TCNT1 and TCNT2 in a cascade connection, when a contention oc during TCNT1 count (during a TCNT2 overflow/underflow) in the T_2 state of the TCNT cycle, the write to TCNT2 is conducted, and the TCNT1 count signal is prohibited. At th there is match with TGR1A or TGR1B and the TCNT1 value, a compare signal is issue timing is shown in figure 12.85.

For cascade connections, be sure to synchronize settings for channels 1 and 2 when setti clearing.

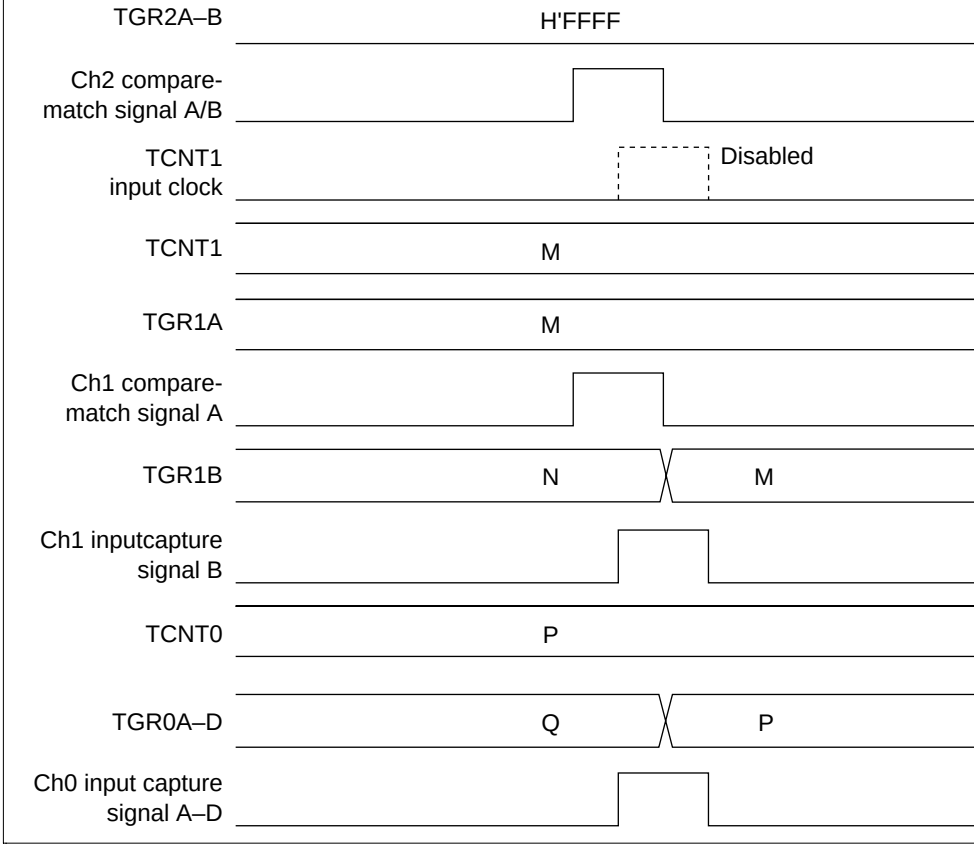


Figure 12.85 TCNT2 Write and Overflow/Underflow Contention with Cascade Co

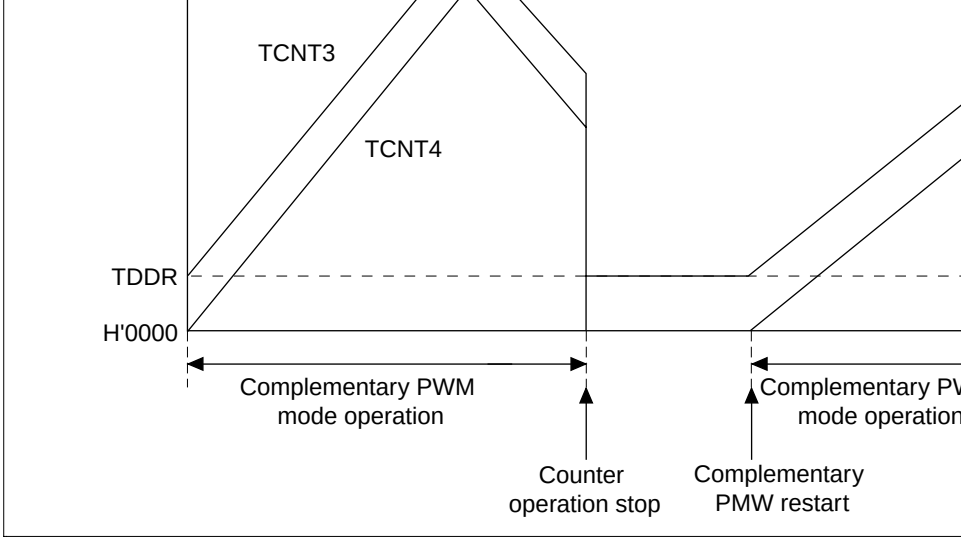


Figure 12.86 Counter Value during Complementary PWM Mode Stop

12.7.12 Buffer Operation Setting in Complementary PWM Mode

In complementary PWM mode, conduct rewrites by buffer operation for the PWM cycle register (TGR3A), PWM carrier cycle setting register (TCDR) and duty setting registers TRG4A, and TGR4B).

In complementary PWM mode, channel 3 and channel 4 buffers operate in accordance v settings BFA and BFB of TMDR3. When TMDR3's BFA bit is set to 1, TGR3C functio buffer register for TGR3A. At the same time, TGR4C functions as the buffer register fo while the TCBR functions as the TCDR's buffer register.

The TGFC bit and TGFD bit of TSR3 are not set when TGR3C and TGR3D are operating buffer registers. On the other hand, TSR4's TGFC and TGFD bits are set even when TGR 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.

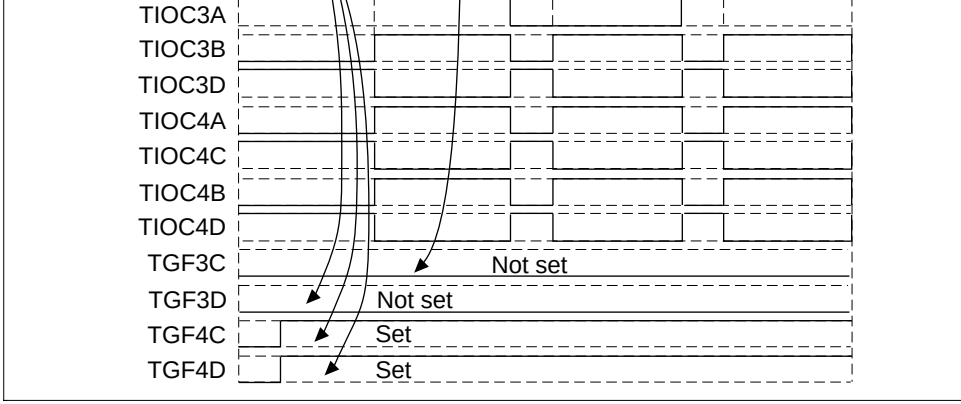


Figure 12.87 Buffer Operation and Compare-Match Flags in Reset Sync PWM

- 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 TMDR4 to 0. The TIOC4C pin will be unable to produce its waveform output if the TMDR4 is set to 1.

  In reset sync PWM mode, the channel 3 and channel 4 buffers operate in accordance BFA and BFB bit settings of TMDR3. For example, if the BFA bit of TMDR3 is set TGR3C functions as the buffer register for TGR3A. At the same time, TGR4C funct buffer register for TRG4A.

  When setting buffer operation for reset sync PWM mode, the compare-match flag T 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 ope buffer registers. The TGFC bit and TGFD bit of TSR4 are not set when TGR4C and are operating as buffer registers.

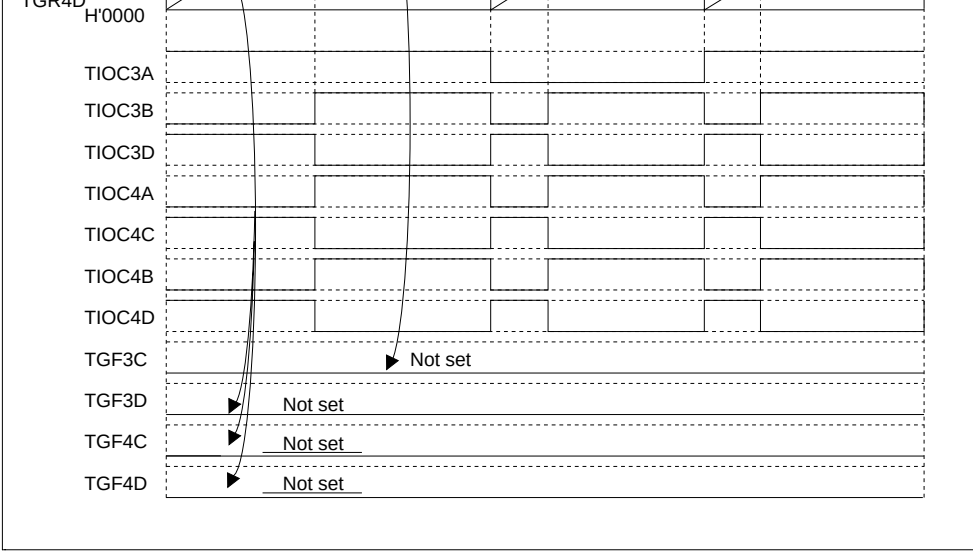


Figure 12.88 Buffer Operation and Compare-Match Flags in Reset Sync PWM M (for A Mask)

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 b TSTR is set to 1. At this point, TCNT4's count clock source and count edge obey the TC setting.

In reset sync PWM mode, with cycle register TGR3A's set value at H'FFFF, take care wh specifying TGR3A compare-match for the counter clear source, since the operation of the overflow flag (TCFV bit) differs with TSR3 and TSR4.

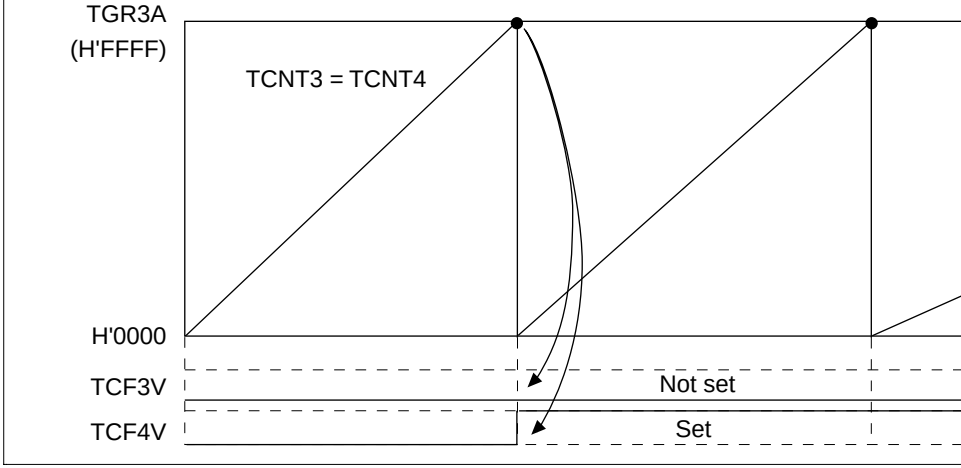


Figure 12.89 Reset Sync PWM Mode Overflow Flag

- 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 CS
  TSTR is set to 1. At this point, TCNT4's count clock source and count edge obey the
  setting.

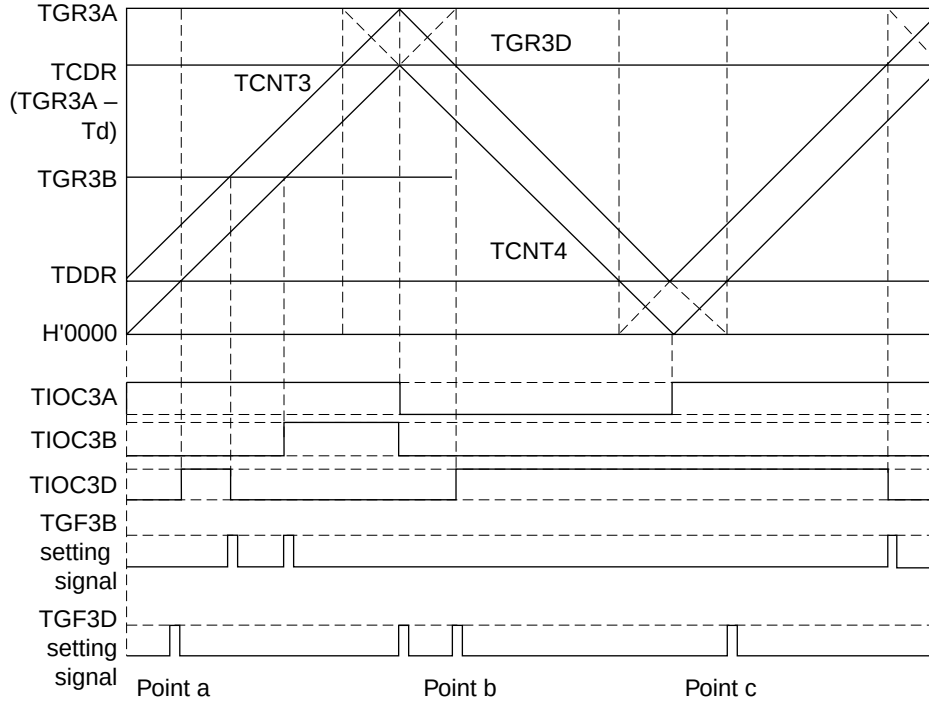
  In reset sync PWM mode, with cycle register TGR3A's set value at H'FFFF and spec
  TGR3A compare-match for the counter clear source, the operation of the overflow fl
  bit for both TSR3 and TSR4 will be the same.

  When TCNT3 and TCNT4 count up to H'FFFF, a compare-match occurs with TGR3
  TCNT3 and TCNT4 are both cleared. At this point, TCFV bits for TSR3 and TSR4 a

  Figure 12.90 shows a TCFV bit operation example in reset sync PWM mode with a s
  for cycle register TGR3A of H'FFFF, when a TGR3A compare-match has been spec
  without synchronous setting for the counter clear source.

TCF4V		▶	Not set

Figure. 12.90 Reset Sync PWM Mode Overflow Flag (for A Mask)



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 signa
not output

Figure 12.91 Special Properties of Compare Match Flag in Complementary PW

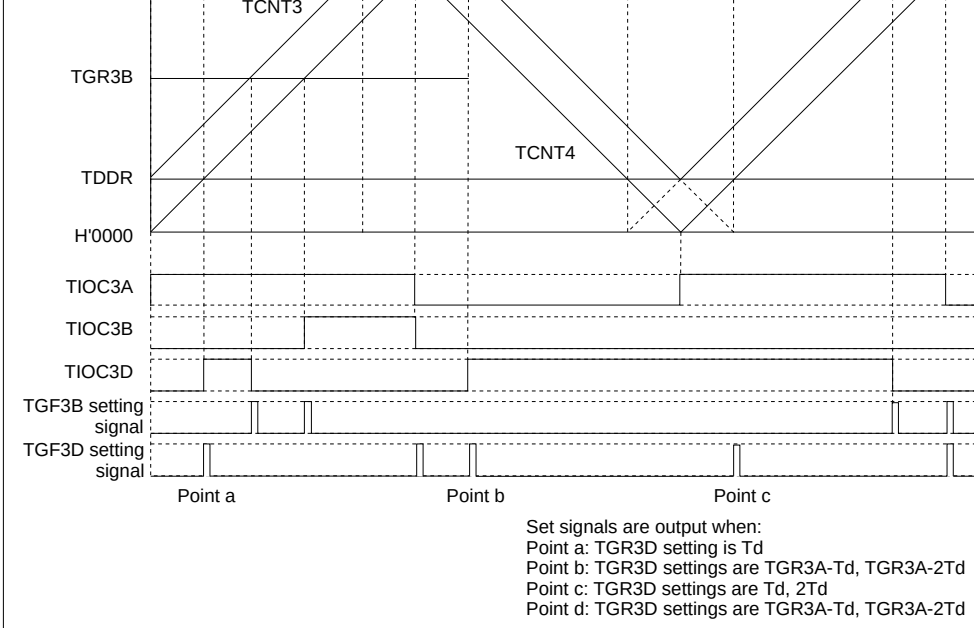


Figure. 12.92 Special Properties of Compare Match Flag in Complementary PWM (for A Mask)

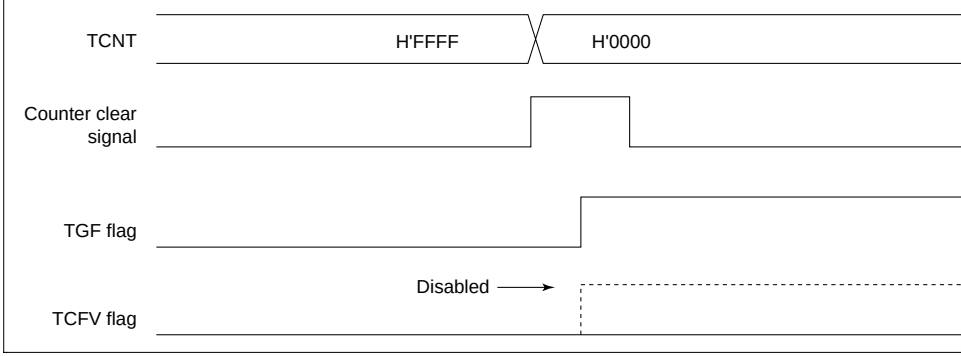


Figure 12.93 Contention between Overflow and Counter Clearing

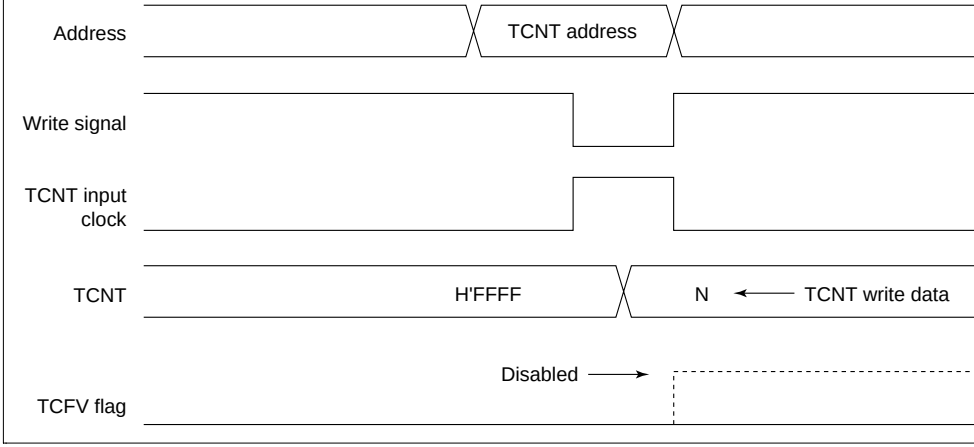


Figure 12.94 Contention between TCNT Write and Overflow

When making a transition from PWM mode 1 to reset-synchronous PWM mode, first sw
normal operation, then initialize the output pins to low level output and set an initial reg
of H'00 before making the transition to reset-synchronous PWM mode.

12.7.19 Output Level in Complementary PWM Mode and Reset-Synchronous PW

When channels 3 and 4 are in complementary PWM mode or reset-synchronous PWM m
PWM waveform output level is set with the OLSP and OLSN bits in the timer output co
register (TOCR). In the case of complementary PWM mode or reset-synchronous PWM
TIOR should be set to H'00.

12.7.20 Cautions on Using the Chopping Function in Complementary PWM Mod Synchronous PWM Mode (A Mask Excluded)

When channels 3 and 4 are in complementary PWM mode or reset-synchronous PWM m
using the chopping output function, setting the PWM waveform output level to low activ
OLSP and OLSN bits in the timer output control register (TOCR) will output an incorre
signal or chopping output.

When channels 3 and 4 are in complementary PWM mode or reset-synchronous PWM m
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 Excluded)

In PWM mode 1, the TGRA and TGRB registers are used in pairs and PWM waveform
to the TIOCA pin. In the same manner, the TGRC and TGRD registers are used in pairs
waveform is output to the TIOCC pin. If either the TGRC or TGRD register is operating
buffer register, the TIOCC pin cannot execute default output setting or PWM waveform
with the I/O control register (TIOR).

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, howeve 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 (writ 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 t mode, set the value following the next value of the PWM duty (write to TGR4D) while th temporary register is not executing comparisons. Furthermore, set the occurrence timing 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 fol the next value of the PWM duty (write to TGR4D) while the temporary register is not exe comparisons and while TCNT3 and TCNT4 are counting up. Furthermore, set the occurr timing of sync clear while the temporary register is not executing comparisons and while and TCNT4 are counting up.

When selecting the mode to transfer using the trough in the transfer mode, set the value f the next value of the PWM duty (write to TGR4D) while the temporary register is not exe comparisons and while TCNT3 and TCNT4 are counting down. Furthermore, set the occ timing of sync clear while the temporary register is not executing comparisons and while 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 foll the next value (write to TGR4D) does not occur at the same time.

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 t
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 se

12.8.2 Reset Start Operation

The MTU output pins (TIOC*) are initialized low by a reset and in standby mode. Since
function selection is performed by the pin function controller (PFC), when the PFC is se
MTU pin states at that point are output to the ports. When MTU output is selected by th
immediately after a reset, the MTU output initial level, low, is output directly at the port
the active level is low, the system will operate at this point, and therefore the PFC settin
be made after initialization of the MTU output pins is completed.

Note: Channel number and port notation are substituted for *.

	After					
Before	**Normal**	**PWM1**	**PWM2**	**PCM**	**CPWM**	**RPW**
Normal	(1)	(2)	(3)	(4)	(5)	(6)
PWM1	(7)	(8)	(9)	(10)	(11)	(12)
PWM2	(13)	(14)	(15)	(16)	None	Non
PCM	(17)	(18)	(19)	(20)	None	Non
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 Er during Operation, Etc.

- When making a transition to a mode (Normal, PWM1, PWM2, PCM) in which the pi level is selected by the timer I/O control register (TIOR) setting, initialize the pins by a TIOR setting.
- In PWM mode 1, since a waveform is not output to the TIOC*B (TIOC *D) pin, setti will not initialize the pins. If initialization is required, carry it out in normal mode, the to PWM mode 1.

3 and 4 output with the timer output master enable register (TOER). Then operate th
accordance with the mode setting procedure (TOCR setting, TMDR setting, TOER s

Pin initialization procedures are described below for the numbered combinations in tabl
The active level is assumed to be low.

Note: Channel number is substituted for * indicated in this article.

n=0 to 15

Figure 12.96 Error Occurrence in Normal Mode, Recovery in Normal Mod

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 TIO
4. Initialize the pins with TIOR. (The example shows initial high output, with low outp 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.

n=0 to 15

Figure 12.97 Error Occurrence in Normal Mode, Recovery in PWM Mod

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. 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.

n=0 to 15

Figure 12.98 Error Occurrence in Normal Mode, Recovery in PWM Mode

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 initial 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 no necessary.

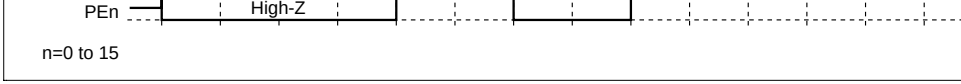


Figure 12.99 Error Occurrence in Normal Mode, Recovery in Phase Counting

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 not necessary.

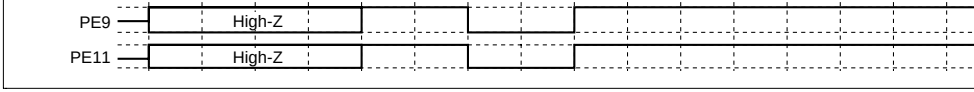


Figure 12.100 Error Occurrence in Normal Mode, Recovery in Complementary Mode

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 wi 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.

Figure 12.101 Error Occurrence in Normal Mode, Recovery in Reset-Synchr 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/disablin 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.

n=0 to 15

Figure 12.102 Error Occurrence in PWM Mode 1, Recovery in Normal Mod

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 TIO
4. Initialize the pins with TIOR. (The example shows initial high output, with low outp 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.

n=0 to 15

Figure 12.103 Error Occurrence in PWM Mode 1, Recovery in PWM Mod

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.

n=0 to 15

Figure 12.104 Error Occurrence in PWM Mode 1, Recovery in PWM Mode

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 initial
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 no necessary.

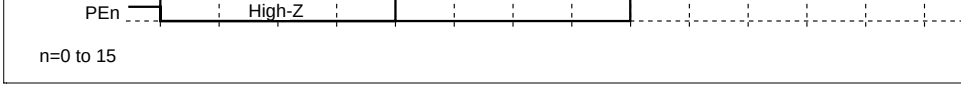


Figure 12.105 Error Occurrence in PWM Mode 1, Recovery in Phase Counting

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 not necessary.

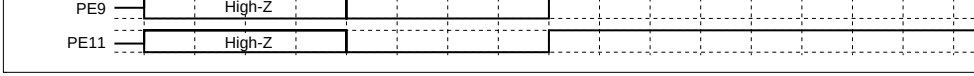


Figure 12.106 Error Occurrence in PWM Mode 1, Recovery in Complementa PWM Mode

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 wi 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.

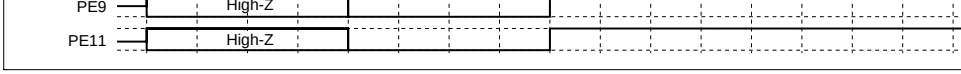


Figure 12.107 Error Occurrence in PWM Mode 1, Recovery in Reset-Synchr
PWM Mode

1 to 14 are the same as in figure 12.106.

15. Select the reset-synchronous PWM output level and cyclic output enabling/disablin 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.

n=0 to 15

Figure 12.108 Error Occurrence in PWM Mode 2, Recovery in Normal Mod

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 outp compare-match occurrence. In PWM mode 2, the cycle register pins are not initialize 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.

n=0 to 15

Figure 12.109 Error Occurrence in PWM Mode 2, Recovery in PWM Mod

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.

n=0 to 15

Figure 12.110 Error Occurrence in PWM Mode 2, Recovery in PWM Mode

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 initial
12. Set MTU output with the PFC.
13. Operation is restarted by TSTR.

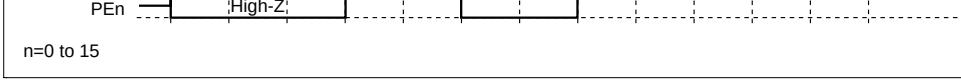


Figure 12.111 Error Occurrence in PWM Mode 2, Recovery in Phase Counting

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.

n=0 to 15

Figure 12.112 Error Occurrence in Phase Counting Mode, Recovery in Normal

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 outp 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.

Figure 12.113 Error Occurrence in Phase Counting Mode, Recovery in PWM

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.

Figure 12.114 Error Occurrence in Phase Counting Mode, Recovery in PWM M

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 initial
12. Set MTU output with the PFC.
13. Operation is restarted by TSTR.

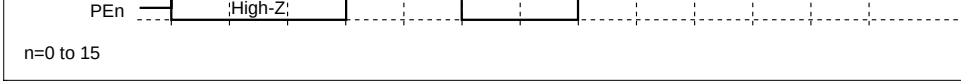


Figure 12.115 Error Occurrence in Phase Counting Mode, Recovery in Ph Counting Mode

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.

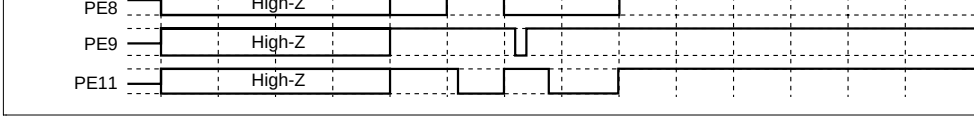


Figure 12.116 Error Occurrence in Complementary PWM Mode, Recovery Normal Mode

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 wi 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 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.

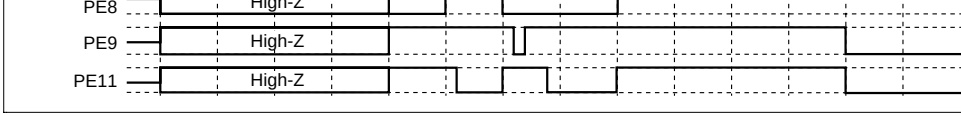


Figure 12.117 Error Occurrence in Complementary PWM Mode, Recovery in Mode 1

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.

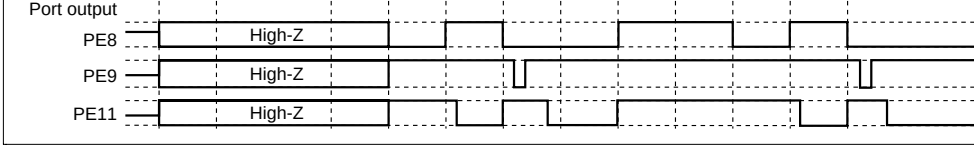


Figure 12.118 Error Occurrence in Complementary PWM Mode, Recovery Complementary PWM Mode

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.

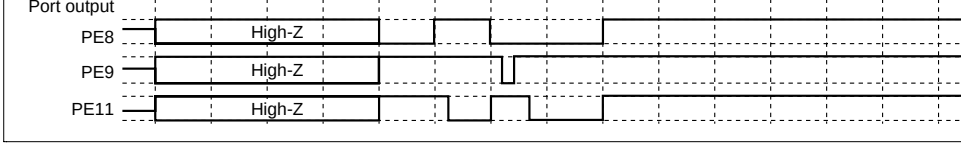


Figure 12.119 Error Occurrence in Complementary PWM Mode, Recovery Complementary PWM Mode

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/disal 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.

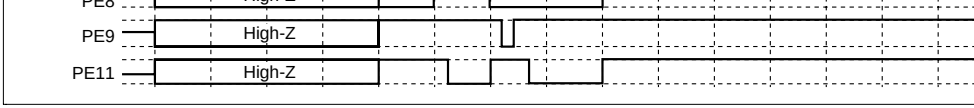


Figure 12.120 Error Occurrence in Complementary PWM Mode, Recovery in Reset-Synchronous PWM Mode

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/dis 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.

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/disablin 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-synchron output initial value.)
11. Set normal operating mode. (MTU positive phase output is low, and negative phase high.)
12. Initialize the pins with TIOR.
13. Set MTU output with the PFC.
14. Operation is restarted by TSTR.

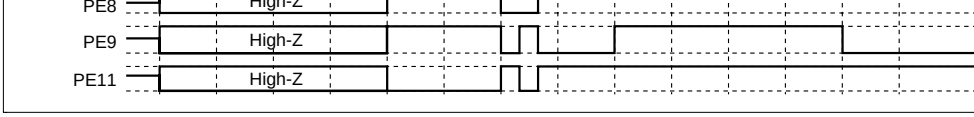


Figure 12.122 Error Occurrence in Reset-Synchronous PWM Mode, Recovery in PWM Mode 1

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 l
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.

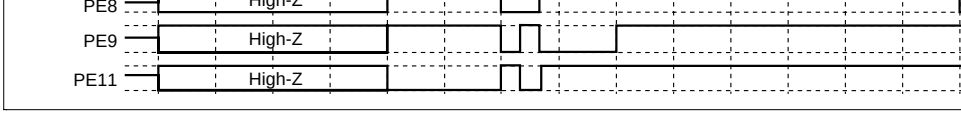


Figure 12.123 Error Occurrence in Reset-Synchronous PWM Mode, Recovery in Complementary PWM Mode

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 v 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.

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.

- Each of the $\overline{\text{POE0}}$–$\overline{\text{POE3}}$ input pins can be set for falling edge, $\phi/8 \times 16$, $\phi/16 \times 16$, o
  16 low-level sampling.
- High-current pins can be set to high-impedance state by $\overline{\text{POE0}}$–$\overline{\text{POE3}}$ pin falling-edg
  level sampling.
- High-current pins can be set to high-impedance state when the high-current pin outpu
  are compared and simultaneous low-level output continues for one cycle or more (ex
  33.3 MHz version).
- Interrupts can be generated by input-level sampling or output-level comparison resul

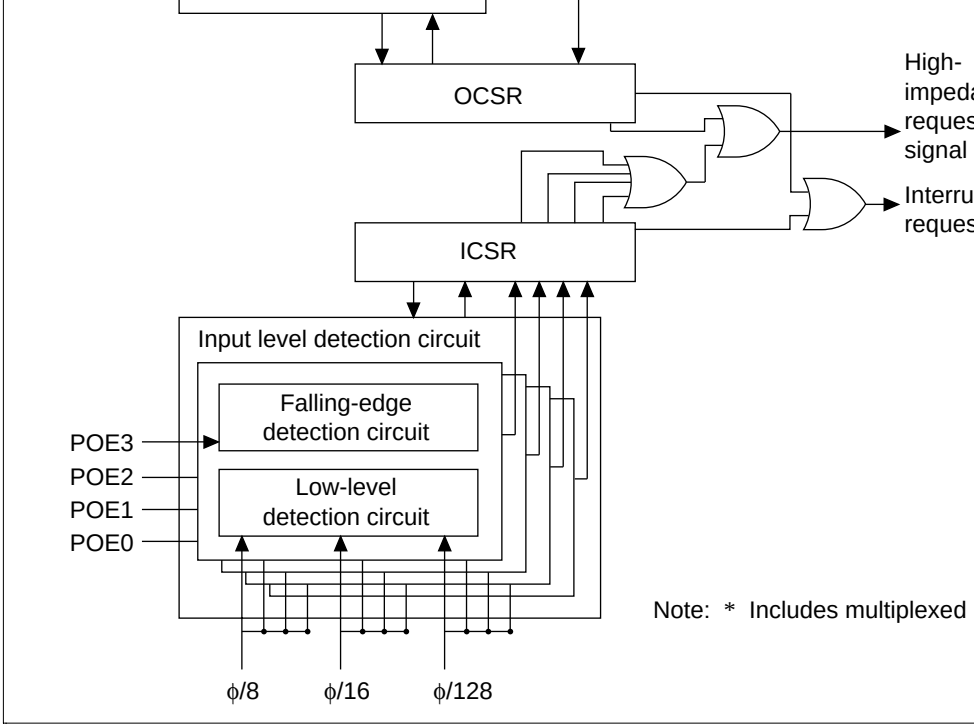


Figure 12.125 POE Block Diagram

Pin Combination	I/O	Description
PE09/TIOC3B and PE11/TIOC3D	Output	All high-current pins are made high-impec state when the pins simultaneously outpu for longer than 1 cycle.
PE12/TIOC4A and PE14/TIOC4C/DACK0/$\overline{AH}$	Output	All high-current pins are made high-impec state when the pins simultaneously outpu for longer than 1 cycle.
PE13/TIOC4B/$\overline{MRES}$ and PE15/TIOC4D/DACK1/$\overline{IRQOUT}$	Output	All high-current pins are made high-impec state when the pins simultaneously outpu 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 regis controls both $\overline{POE0}$–$\overline{POE3}$ pin input signal detection and interrupts. The output level control/status register (OCSR) controls both the enable/disable of output comparison an interrupts.

Table 12.20 Input Level Control/Status Register Configuration

Name	Abbreviation	R/W	Initial Value	Address	Acces
Input level control/status register	ICSR	R/(W)*1	H'0000	H'FFFF83C0 H'FFFF83C1	8, 16,
Output level control/status register	OCSR	R/(W)*2	H'0000	H'FFFF83C2 H'FFFF83C3	8, 16,

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.

Bit:	7	6	5	4	3	2	1	0
	POE3F	POE2F	POE1F	POE0F	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R	R

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 b to the $\overline{\text{POE3}}$ pin.

Bit 15: POE3F	Description
0	Clear condition: By writing 0 to POE3F after reading a PO (initial value)
1	Set condition: When the input set by ICSR bits 7 and 6 oc the $\overline{\text{POE3}}$ pin

- Bit 14—POE2 Flag (POE2F): This flag indicates that a high impedance request has b to the $\overline{\text{POE2}}$ pin.

Bit 14: POE2F	Description
0	Clear condition: By writing 0 to POE2F after reading a PO (initial value)
1	Set condition: When the input set by ICSR bits 5 and 4 oc the $\overline{\text{POE2}}$ pin

0	Clear condition: By writing 0 to POE0F after reading a P (initial value)
1	Set condition: When the input set by ICSR bits 1 and 0 c the $\overline{\text{POE0}}$ pin

- Bits 11–9—Reserved: These bits always read as 0. The write value should always be
- Bit 8—Port Interrupt Enable (PIE): Enables or disables interrupt requests when any o 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 the $\overline{\text{POE3}}$ pin.

Bit 7: POE3M1	Bit 6: POE3M0	Description
0	0	Accept request on falling edge of $\overline{\text{POE3}}$ input. (initia
	1	Accept request when $\overline{\text{POE3}}$ input has been sample ϕ/8 clock pulses, and all are low level.
1	0	Accept request when $\overline{\text{POE3}}$ input has been sample ϕ/16 clock pulses, and all are low level.
	1	Accept request when $\overline{\text{POE3}}$ input has been sample 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 m
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
	1	Accept request when $\overline{\text{POE1}}$ input has been sampled ϕ/8 clock pulses, and all are low level.
1	0	Accept request when $\overline{\text{POE1}}$ input has been sampled ϕ/16 clock pulses, and all are low level.
	1	Accept request when $\overline{\text{POE1}}$ input has been sampled 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 m
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
	1	Accept request when $\overline{\text{POE0}}$ input has been sampled ϕ/8 clock pulses, and all are low level.
1	0	Accept request when $\overline{\text{POE0}}$ input has been sampled ϕ/16 clock pulses, and all are low level.
	1	Accept request when $\overline{\text{POE0}}$ input has been sampled ϕ/128 clock pulses, and all are low level.

Bit:	7	6	5	4	3	2	1
	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0
R/W:	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 outputs compared, the outputs of at least one pair have simultaneously become Low output.

Bit 15: OSF	Description
0	Clear condition: By writing 0 to OSF after reading an OSF = 1 (init
1	Set condition: When any one pair of the 2-phase outputs simultane become Low level

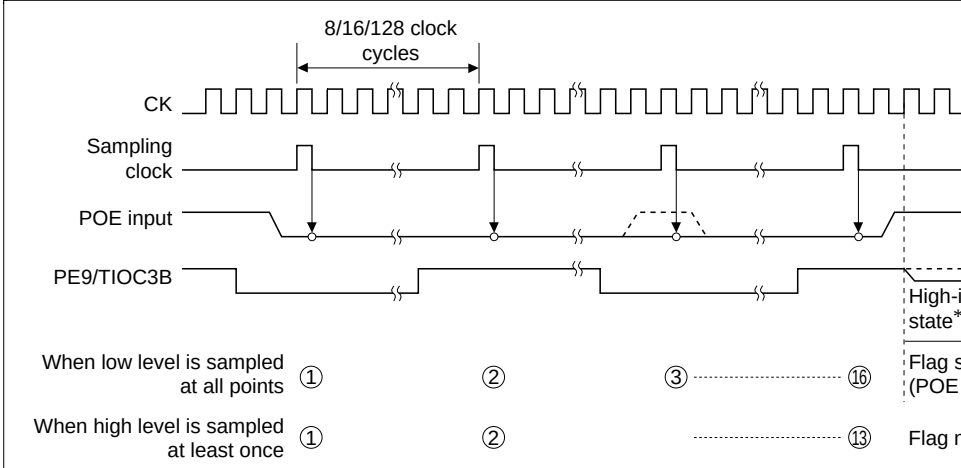
- Bits 14–10—Reserved: These bits always read as 0. The write value should always b

0	Output level compare disabled (initial value)
1	Output level compare enabled; makes an output high impedance re when OSF = 1.

- Bit 8—Output Short Interrupt Enable (OIE): Makes interrupt requests when the OSF l 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.



Note: * Other large-current pins (PE11/TIOC3D, PE12/TIOC4A, PE13/TIOC4B/$\overline{\text{MRES}}$, PE14/TIOC4C/
PE15/TIOC4D/DACK1/$\overline{\text{IROOUT}}$) also go to the high-impedance state at the same time.

Figure 12.126 Low-Level Detection Operation

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 released either by returning them to their initial state with a power-on reset, or by clearing the bit 12–15 (POE0F–POE3F) flags of the ICSR. High-current pins that have become hi impedance due to output-level detection can be released either by returning them to their state with a power-on reset, or by first clearing bit 9 (OCE) of the OCSR to disable outpu compares, then clearing the bit 15 (OSF) flag. However, when returning from high-impec state by clearing the OSF flag, always do so only after outputting a high level from the hi current pins (TIOC3B, TIOC3D, TIOC4A, TIOC4B, TIOC4C, and TIOC4D). High-level can be achieved by setting the MTU internal registers. See section 12.2, MTU Register Descriptions, for details.

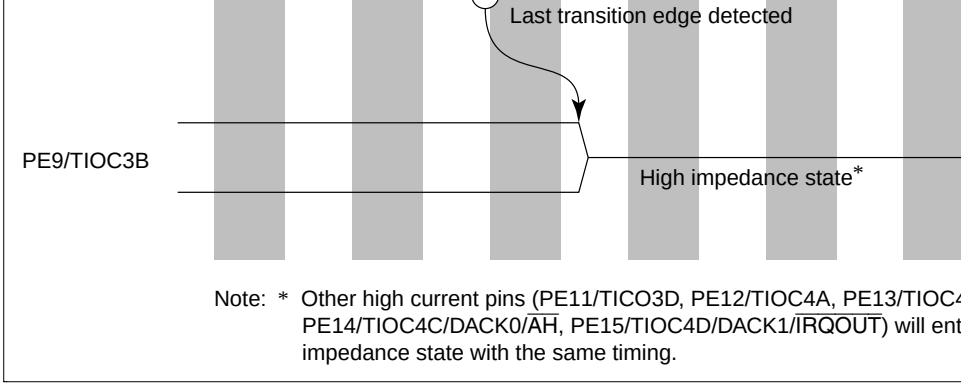


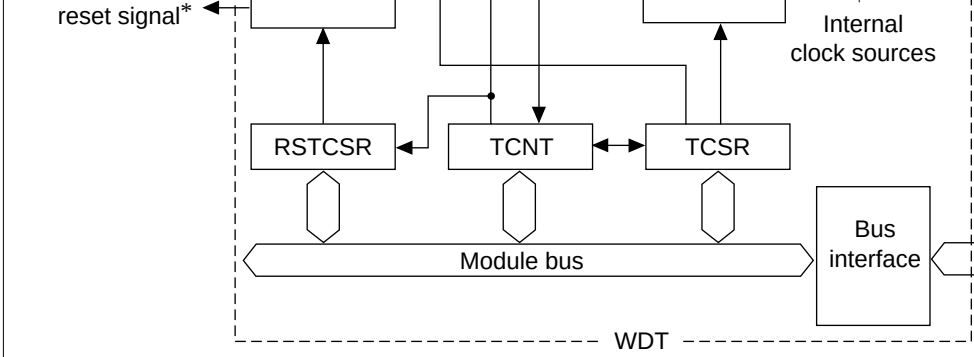
Figure 12.128 Last Transition Edge Detection Operation

12.11.5 Usage Notes

To perform POE level detection, first set POE input to high level.

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 timer mode, overflow signal $\overline{\text{WDTOVF}}$ is output externally. You can select whether the chip internally when this happens. Either the power-on reset or manual reset sign selected as the internal reset signal.
- Generates interrupts in the interval timer mode. When the counter overflows, it gene interval timer interrupt.
- Clears standby mode.
- Works with eight counter input clocks.



TCSR: Timer control/status register
TCNT: Timer counter
RSTCSR: Reset control/status register

Note: * The internal reset signal can be generated by setting the register. The type of reset can be selected (power-on or manual).

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 sign watchdog timer mode

register

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 th
more difficult to write to. See section 13.2.4, Register Access, for details.) When the tim
bit (TME) in the timer control/status register (TCSR) is set to 1, the watchdog timer cou
counting pulses of an internal clock selected by clock select bits 2–0 (CKS2–CKS0) in t
When the value of the TCNT overflows (changes from H'FF to H'00), a watchdog timer
signal ($\overline{\text{WDTOVF}}$) or interval timer interrupt (ITI) is generated, depending on the mode
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
initialized in the standby mode. The TCNT is not initialized by a manual reset from an e
source ($\overline{\text{MRES}}$), but is initialized by a manual reset from the WDT.

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

Initial value:	0	0	0	1	1	0	0
R/W:	R/(W)	R/W	R/W	R	R	R/W	R/W

- Bit 7—Overflow Flag (OVF): Indicates that the TCNT has overflowed from H'FF to H
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/$\overline{IT}$): Selects whether to use the WDT as a watchdog t
interval timer. When the TCNT overflows, the WDT either generates an interval timer
interrupt (ITI) or generates a $\overline{WDTOVF}$ signal, depending on the mode selected.

Bit 6: WT/$\overline{IT}$	Description
0	Interval timer mode: interval timer interrupt request to the CPU TCNT overflows (initial value)
1	Watchdog timer mode: $\overline{WDTOVF}$ signal output externally when overflows. (Section 13.2.3, Reset Control/Status Register (RST describes in detail what happens when TCNT overflows in the timer mode.)

			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
an overflow occurs.

R/W: R/(W) R/W R/W 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 overfl (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 int 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 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.

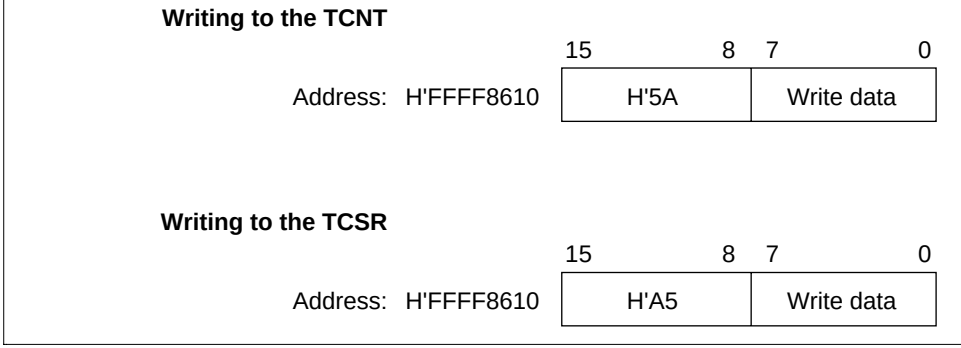


Figure 13.2 Writing to the TCNT and TCSR

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 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 t byte. This clears the WOVF bit to 0. The RSTE and RSTS bits are not affected. To write RSTE and RSTS bits, the upper byte must be H'5A and the lower byte must be the write values of bits 6 and 5 of the lower byte are transferred to the RSTE and RSTS bits, respe The WOVF bit is not affected.

Reading from the TCNT, TCSR, and RSTCSR: TCNT, TCSR, and RSTCSR are read other registers. Use byte transfer instructions. The read addresses are H'FFFF8610 for the 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. Sof must prevent TCNT overflow by rewriting the TCNT value (normally by writing H'00) b overflow occurs. No TCNT overflows will occur while the system is operating normally, TCNT fails to be rewritten and overflows occur due to a system crash or the like, a $\overline{WDT}$ signal is output externally (figure 13.4). The $\overline{WDTOVF}$ signal can be used to reset the sy $\overline{WDTOVF}$ 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 inte simultaneous to the $\overline{WDTOVF}$ signal when TCNT overflows. Either a power-on reset or reset can be selected by the RSTS bit. The internal reset signal is output for 512 ϕ clock c

When a watchdog overflow reset is generated simultaneously with a reset input at the $\overline{RE}$ $\overline{RES}$ 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.

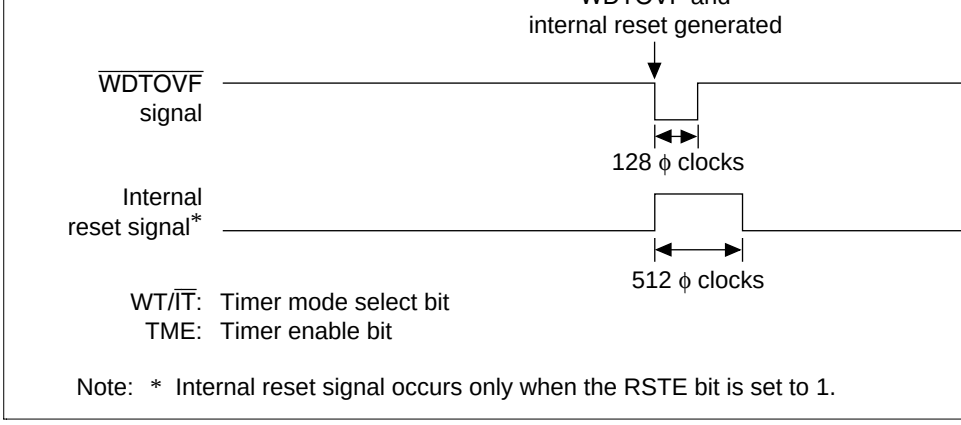


Figure 13.4 Operation in the Watchdog Timer Mode

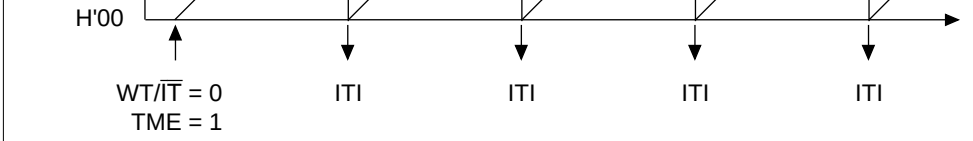


Figure 13.5 Operation in the Interval Timer Mode

13.3.3 Clearing the Standby Mode

The watchdog timer has a special function to clear the standby mode with an NMI interru 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 the watchdog timer counter before it enters the standby mode. The chip cannot enter the s mode while the TME bit is set to 1. Set bits CKS2–CKS0 so that the counter overflow int 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 the clock oscillator starts running and the watchdog timer starts incrementing at the rate s by bits CKS2–CKS0 before the standby mode was entered. When the TCNT overflows (c from H'FF to H'00), the clock is presumed to be stable and usable; clock signals are suppl entire chip and the standby mode ends.

For details on the standby mode, see section 24, Power-Down State.

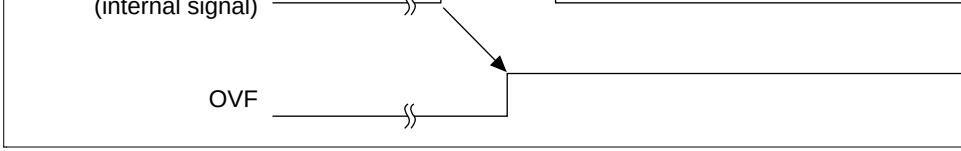


Figure 13.6 Timing of Setting the OVF

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 and a $\overline{\text{WDTOVF}}$ signal is output. When the RSTE bit is set to 1, TCNT overflow enable internal reset signal to be generated for the entire chip (figure 13.7).

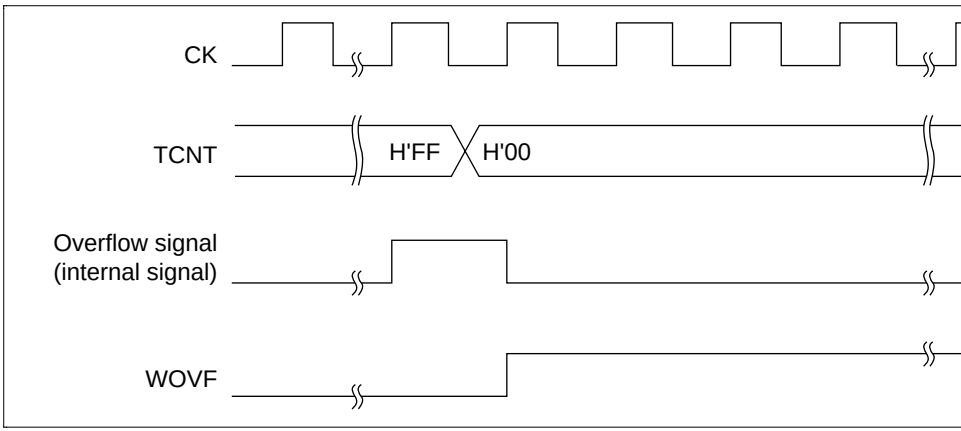


Figure 13.7 Timing of Setting the WOVF Bit

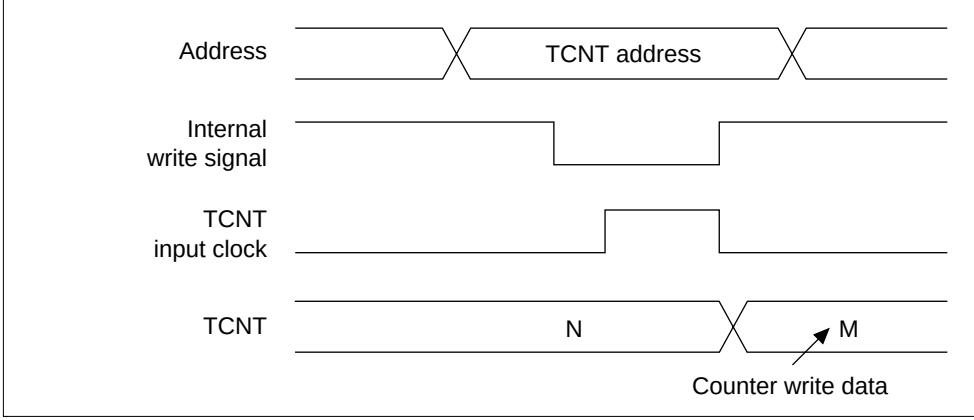


Figure 13.8 Contention between TCNT Write and Increment

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 in
incorrectly. Always stop the watchdog timer (by clearing the TME bit to 0) before changi
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
before switching between interval timer mode and watchdog timer mode.

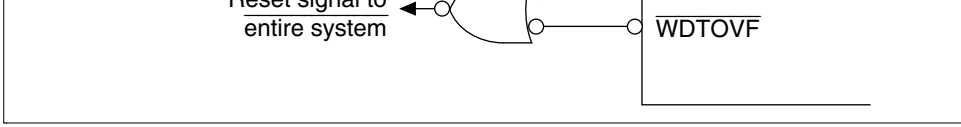


Figure 13.9 Example of a System Reset Circuit with a $\overline{\text{WDTOVF}}$ Signal

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 interna
TCNT overflow occurs, but the TCNT and TCSR in the WDT will reset.

- Select asynchronous or clock synchronous as the serial communications mode.
  - Asynchronous mode: Serial data communications are synched by start-stop in ch units. The SCI can communicate with a universal asynchronous receiver/transmit (UART), an asynchronous communication interface adapter (ACIA), or any othe employs a standard asynchronous serial communication. It can also communicate or more other processors using the multiprocessor communication function. Ther 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 cl 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 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 S (external).
- Four types of interrupts: Transmit-data-empty, transmit-end, receive-data-full, and re error interrupts are requested independently. The transmit-data-empty and receive-da interrupts can start the direct memory access controller (DMAC)/data transfer contro to transfer data.

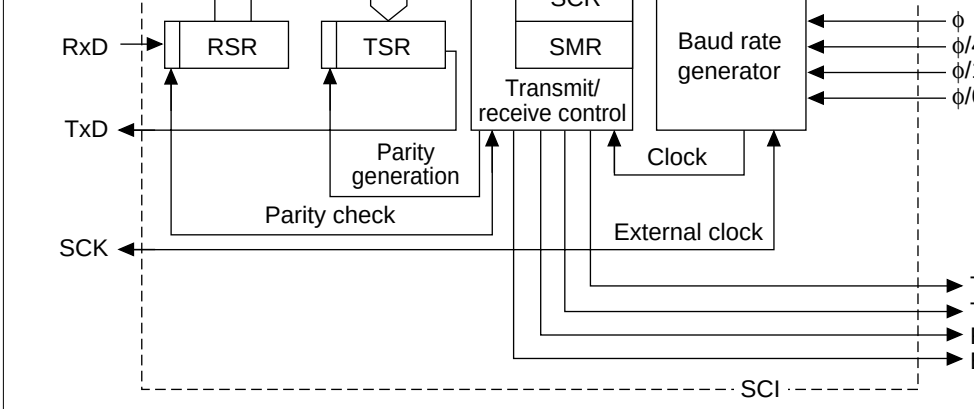


Figure 14.1 SCI Block Diagram

	Transmit data pin	TxD1	Output	SCI1 transmit da

14.1.4 Register Configuration

Table 14.2 summarizes the SCI internal registers. These registers select the communicat (asynchronous or clock synchronous), specify the data format and bit rate, and control th transmitter and receiver sections.

Table 14.2 Registers

Channel	Name	Abbreviation	R/W	Initial Value	Address*2
0	Serial mode register	SMR0	R/W	H'00	H'FFFF81A0
	Bit rate register	BRR0	R/W	H'FF	H'FFFF81A1
	Serial control register	SCR0	R/W	H'00	H'FFFF81A2
	Transmit data register	TDR0	R/W	H'FF	H'FFFF81A3
	Serial status register	SSR0	R/(W)*1	H'84	H'FFFF81A4
	Receive data register	RDR0	R	H'00	H'FFFF81A5
1	Serial mode register	SMR1	R/W	H'00	H'FFFF81B0
	Bit rate register	BRR1	R/W	H'FF	H'FFFF81B1
	Serial control register	SCR1	R/W	H'00	H'FFFF81B2
	Transmit data register	TDR1	R/W	H'FF	H'FFFF81B3
	Serial status register	SSR1	R/(W)*1	H'84	H'FFFF81B4
	Receive data register	RDR1	R	H'00	H'FFFF81B5

Notes: *1 The only value that can be written is a 0 to clear the flags.
*2 Do not access empty addresses.

R/W:

14.2.2 Receive Data Register (RDR)

The receive data register (RDR) stores serial receive data. The SCI completes the receptio byte of serial data by moving the received data from the receive shift register (RSR) into for storage. The RSR is then ready to receive the next data. This double buffering allows to receive data continuously.

The CPU can read but not write the RDR. The RDR is initialized to H'00 by a power-on r standby mode. Manual reset does not initialize RDR.

Bit:	7	6	5	4	3	2	1
Initial value:	0	0	0	0	0	0	0
R/W:	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 t transmit data register (TDR) into the TSR, then transmits the data serially from the TxD p (bit 0) first. After transmitting one data byte, the SCI automatically loads the next transmi 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 can always read and write the TDR. The TDR is initialized to H'FF by a powe
or in standby mode. Manual reset does not initialize TDR.

Bit:	7	6	5	4	3	2	1
Initial value:	1	1	1	1	1	1	1
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 commun
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 powe
or in standby mode. Manual reset does not initialize SMR.

Bit:	7	6	5	4	3	2	1
	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

1	Seven-bit data. (When 7-bit data is selected, the MSB (bit 7) of transmit data register is not transmitted.)

- Bit 5—Parity Enable (PE): Selects whether to add a parity bit to transmit data and to c parity of receive data, in the asynchronous mode. In the clock synchronous mode, a pa 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 parity bit is added to transmit data, depending on the parity mo setting. Receive data parity is checked according to the even/o mode setting.

- Bit 4—Parity Mode (O/$\overline{E}$): Selects even or odd parity when parity bits are added and c The O/$\overline{E}$ setting is used only in asynchronous mode and only when the parity enable b 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 di

Bit 4: O/$\overline{E}$	Description
0	Even parity (initial value). If even parity is selected, the parity b added to transmit data to make an even number of 1s in the tra character and parity bit combined. Receive data is checked to s has an even number of 1s in the received character and parity combined.
1	Odd parity. If odd parity is selected, the parity bit is added to tra data to make an odd number of 1s in the transmitted character parity bit combined. Receive data is checked to see if it has an number of 1s in the received character and parity bit combined

- Bit 2—Multiprocessor Mode (MP): Selects multiprocessor format. When multiproce format is selected, settings of the parity enable (PE) and parity mode (O/$\overline{E}$) bits are ig 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 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 source of the on-chip baud rate generator. Four clock sources are available; ϕ, ϕ/4, ϕ/ ϕ/64. For further information on the clock source, bit rate register settings, and baud section 14.2.8, Bit Rate Register (BRR).

Bit 1: CKS1	Bit 0: CKS0	Description
0	0	ϕ (initial value)
	1	ϕ/4
1	0	ϕ/16
	1	ϕ/64

- Bit 7—Transmit Interrupt Enable (TIE): Enables or disables the transmit-data-empty i (TxI) requested when the transmit data register empty bit (TDRE) in the serial status r (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 v The TxI interrupt request can be cleared by reading TDRE afte 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 interr requested when the receive data register full bit (RDRF) in the serial status register (S to 1 by transfer of serial receive data from the RSR to the RDR. It also enables or disa receive-error interrupt (ERI) requests.

Bit 6: RIE	Description
0	Receive-data-full interrupt (RxI) and receive-error interrupt (ER requests are disabled (initial value). RxI and ERI interrupt requ be cleared by reading the RDRF flag or error flag (FER, PER, c after it has been set to 1, then clearing the flag to 0, or by clear to 0.
1	Receive-data-full interrupt (RxI) and receive-error interrupt (ER requests are enabled.

0	Receiver disabled (initial value). Clearing RE to 0 does not aff receive flags (RDRF, FER, PER, ORER). These flags retain t previous values.
1	Receiver enabled. Serial reception starts when a start bit is de the asynchronous mode, or synchronous clock input is detect clock synchronous mode. Select the receive format in the SM setting RE to 1.

- Bit 3—Multiprocessor Interrupt Enable (MPIE): Enables or disables multiprocessor The MPIE setting is used only in the asynchronous mode, and only if the multiproce bit (MP) in the serial mode register (SMR) is set to 1 during reception. The MPIE se 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 operati value). MPIE is cleared when the MPIE bit is cleared to 0, or t multiprocessor bit (MPB) is set to 1 in receive data.
1	Multiprocessor interrupts are enabled. Receive-data-full interr requests (RxI), receive-error interrupt requests (ERI), and set RDRF, FER, and ORER status flags in the serial status regist are disabled until data with the multiprocessor bit set to 1 is re The SCI does not transfer receive data from the RSR to the R not detect receive errors, and does not set the RDRF, FER, a flags in the serial status register (SSR). When it receives data includes MPB = 1, MPB is set to 1, and the SCI automatically MPIE to 0, generates RxI and ERI interrupts (if the TIE and R the SCR are set to 1), and allows the FER and ORER bits to

and CKE0, the SCK pin can be used for serial clock output, or serial clock input. Sele
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 i
clocked (CKE1 = 0). The CKE0 setting is ignored in the clock synchronous mode, or
external clock source is selected (CKE1 = 1). Select the SCI operating mode in the se
register (SMR) before setting CKE1 and CKE0. For further details on selection of the
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 (inp is ignored) or output pin (output level is unde
		Clock synchronous mode	Internal clock, SCK pin used for synchronous output*2
0	1	Asynchronous mode	Internal clock, SCK pin used for clock output
		Clock synchronous mode	Internal clock, SCK pin used for synchronous output
1	0	Asynchronous mode	External clock, SCK pin used for clock input*
		Clock synchronous mode	External clock, SCK pin used for synchronou input
1	1	Asynchronous mode	External clock, SCK pin used for clock input*
		Clock synchronous mode	External clock, SCK pin used for synchronou input

Notes: *1 The SCK pin is multiplexed with other functions. Use the pin function controller 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.

R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R

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 tra 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 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 mo bit in the serial control register (SCR) is cleared to 0, or TDR contents into TSR, so new data can be written in TDR

received data is lost.

- Bit 5—Overrun Error (ORER): Indicates that data reception ended abnormally due to overrun error.

Bit 5: ORER	Description
0	Receiving is in progress or has ended normally (initial value). Clearing th to 0 in the serial control register does not affect the ORER bit, which reta previous value. ORER is cleared to 0 when the chip is power-on reset or enters standby 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 receiv before the overrun error, so subsequent receive data is lost. Serial recei cannot continue while ORER is set to 1. In the clock synchronous mode transmitting is disabled. ORER is set to 1 if reception of the next serial data ends when RDRF is

	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 be 0

- Bit 3—Parity Error (PER): Indicates that data reception (with parity) ended abnorma a parity error in the asynchronous mode.

Bit 3: PER	Description
0	Receiving is in progress or has ended normally (initial value). Clearing to 0 in the serial control register does not affect the PER bit, which reta previous value. PER is cleared to 0 when the chip is power-on reset or enters standby 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 tra receive data into the RDR but does not set RDRF. Serial receiving can continue while PER is set to 1. In the clock synchronous mode, serial t is also disabled. PER is set to 1 if the number of 1s in receive data, including the parity not match the even or odd parity setting of the parity mode bit (O/$\overline{E}$) in mode register (SMR)

- Bit 1—Multiprocessor Bit (MPB): Stores the value of the multiprocessor bit in receiv when a multiprocessor format is selected for receiving in the asynchronous mode. The 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 cleare when a multiprocessor format is selected, the MPB retains its previous v
1	Multiprocessor bit value in receive data is 1

- Bit 0—Multiprocessor Bit Transfer (MPBT): Stores the value of the multiprocessor bi to transmit data when a multiprocessor format is selected for transmitting in the asyn mode. The MPBT setting is ignored in the clock synchronous mode, when a multipro 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

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 exa
BBR settings in the clock synchronous mode.

Table 14.3 Bit Rates and BRR Settings in Asynchronous Mode

Bit Rate (Bits/s)	ϕ (MHz)								
	4			4.9152			6		
	n	N	Error (%)	n	N	Error (%)	n	N	E
110	2	70	0.03	2	86	0.31	2	106	–
150	1	207	0.16	1	255	0.00	2	77	0
300	1	103	0.16	1	127	0.00	1	155	0
600	0	207	0.16	0	255	0.00	1	77	0
1200	0	103	0.16	0	127	0.00	0	155	0
2400	0	51	0.16	0	63	0.00	0	77	0
4800	0	25	0.16	0	31	0.00	0	38	0
9600	0	12	0.16	0	15	0.00	0	19	–
14400	0	8	–3.55	0	10	–3.03	0	12	0
19200	0	6	–6.99	0	7	0.00	0	9	–
28800	0	3	8.51	0	4	6.67	0	6	–
31250	0	3	0.00	0	4	–1.70	0	5	0
38400	0	2	8.51	0	3	0.00	0	4	–

4800	0	47	0.00	0	51	0.16	0	63	0.
9600	0	23	0.00	0	25	0.16	0	31	0.
14400	0	15	0.00	0	16	2.12	0	20	1.
19200	0	11	0.00	0	12	0.16	0	15	0.
28800	0	7	0.00	0	8	–3.55	0	10	–3
31250	0	6	5.33	0	7	0.00	0	9	–1
38400	0	5	0.00	0	6	–6.99	0	7	0.

4800	0	64	0.16	0	71	0.00	0	77	0
9600	0	32	–1.36	0	35	0.00	0	38	0
14400	0	21	–1.36	0	23	0.00	0	25	0
19200	0	15	1.73	0	17	0.00	0	19	-
28800	0	10	–1.36	0	11	0.00	0	12	0
31250	0	9	0.00	0	10	0.54	0	11	0
38400	0	7	1.73	0	8	0.00	0	9	-

4800	0	79	0.00	0	90	0.16	0	95	0.
9600	0	39	0.00	0	45	–0.93	0	47	0.
14400	0	26	–1.23	0	29	1.27	0	31	0.
19200	0	19	0.00	0	22	–0.93	0	23	0.
28800	0	12	2.56	0	14	1.27	0	15	0.
31250	0	11	2.40	0	13	0.00	0	14	–1
38400	0	9	0.00	0	10	3.57	0	11	0.

4800	0	103	0.16	0	111	0.00	0	116	0
9600	0	51	0.16	0	55	0.00	0	58	-
14400	0	34	–0.79	0	36	0.90	0	38	0
19200	0	25	0.16	0	27	0.00	0	28	1
28800	0	16	2.12	0	18	–1.75	0	19	-
31250	0	15	0.00	0	16	1.20	0	17	0
38400	0	12	0.16	0	13	0.00	0	14	-

4800	0	119	0.00	0	127	0.00	0	129	0.
9600	0	59	0.00	0	63	0.00	0	64	0.
14400	0	39	0.00	0	42	–0.78	0	42	0.
19200	0	29	0.00	0	31	0.00	0	32	–1
28800	0	19	0.00	0	20	1.59	0	21	–1
31250	0	17	2.40	0	19	–1.70	0	19	0.
38400	0	14	0.00	0	15	0.00	0	15	1.

4800	0	142	0.16	0	143	0.00	0	155	0
9600	0	71	–0.54	0	71	0.00	0	77	0
14400	0	47	–0.54	0	47	0.00	0	51	0
19200	0	35	–0.54	0	35	0.00	0	38	0
28800	0	23	–0.54	0	23	0.00	0	25	0
31250	0	21	0.00	0	21	0.54	0	23	0
38400	0	17	–0.54	0	17	0.00	0	19	-

4800	0	159	0.00	0	167	0.00	0	168	0.
9600	0	79	0.00	0	83	0.00	0	84	–0
14400	0	52	0.63	0	55	0.00	0	55	0.
19200	0	39	0.00	0	41	0.00	0	41	0.
28800	0	26	–1.23	0	27	0.00	0	27	0.
31250	0	24	–1.70	0	25	–0.75	0	25	0.
38400	0	19	0.00	0	20	0.00	0	20	0.

4800	0	175	0.00	0	181	0.16	0	191
9600	0	87	0.00	0	90	0.16	0	95
14400	0	58	–0.56	0	60	–0.39	0	63
19200	0	43	0.00	0	45	0.93	0	47
28800	0	28	1.15	0	29	1.27	0	31
31250	0	26	0.12	0	27	0.00	0	28
38400	0	21	0.00	0	22	–0.93	0	23

4800	0	194	0.16	0	207	0.00	0	207	0
9600	0	97	-0.35	0	103	0.00	0	103	0
14400	0	64	0.16	0	68	0.48	0	68	0
19200	0	48	-0.35	0	51	0.00	0	51	0
28800	0	32	-1.36	0	34	-0.95	0	34	-0
31250	0	29	0.00	0	31	-0.16	0	31	0
38400	0	23	1.73	0	25	0.00	0	25	0

4800	0	214	-0.07	0	215	0.00	0	216
9600	0	106	0.39	0	107	0.00	0	108
14400	0	71	–0.54	0	91	0.00	0	91
19200	0	53	–0.54	0	53	0.00	0	53
28800	0	35	–0.54	0	35	0.00	0	35
31250	0	32	0.00	0	32	0.54	0	32
38400	0	26	–0.54	0	26	0.00	0	26

10k	0	99	0	199	0	249	1	7
25k	0	39	0	79	0	99	0	1
50k	0	19	0	39	0	49	0	5
100k	0	9	0	19	0	24	0	2
250k	0	3	0	7	0	9	0	1
500k	0	1	0	3	0	4	0	5
1M	0	0*	0	1	—	—	0	2
2.5M					0	0*	0	0
5M								

10k	1	99	1	124	1	149	1
25k	0	159	0	199	0	239	1
50k	0	79	0	99	0	119	0
100k	0	39	0	49	0	59	0
250k	0	15	0	19	0	23	0
500k	0	7	0	9	0	11	0
1M	0	3	0	4	0	5	0
2.5M	—	—	0	1	—	—	0
3.5M			—	—	—	—	0
5M			0	0*	—	—	—
7M					—	—	0

10k	1	187	1	199	1	205	1	2
25k	1	74	1	79	1	82	1	8
50k	0	149	0	159	0	164	0	1
100k	0	74	0	79	0	82	0	8
250k	0	29	0	31	0	32	0	3
500k	0	14	0	15	0	16	0	1
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:

$$N = \frac{\phi}{64 \times 2^{2n-1} \times B} \times 10^6 - 1$$

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 ra
generator is being used for various frequencies. Tables 14.6 and 14.7 show the maximur
external clock input.

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

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

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

- Data length is selectable: seven or eight bits.
- Parity and multiprocessor bits are selectable, as well as the stop bit length (one or tw
  These selections determine the transmit/receive format and character length.
- In receiving, it is possible to detect framing errors (FER), parity errors (PER), overru
  (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 g
    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 frequenc
    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 g
    clock, and outputs a synchronous clock signal to external devices.
  - When an external clock is selected, the SCI operates on the input synchronous cl
    on-chip baud rate generator is not used.

			1		0		Set	
					1			
Asynchronous (multiprocessor format)		0	*	1	0	8-bit	Not set	Set
			*		1			
		1	*		0	7-bit		
			*		1			
Clock synchronous	1	*	*	*	*	8-bit		Not set

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 frequenc matching the bit rate
		1	0	External	Inputs a clock with frequency the bit rate
			1		
Clock synch-ronous	1	0	0	Internal	Outputs the synchronous cloc
			1		
		1	0	External	Inputs the synchronous clock
			1		

Note: * Select the function in combination with the pin function controller (PFC).

(high or low), and stop bit (high), in that order.

When receiving in the asynchronous mode, the SCI synchronizes on the falling edge of t
bit. The SCI samples each data bit on the eighth pulse of a clock with a frequency 16 tim
rate. Receive data is latched at the center of each bit.

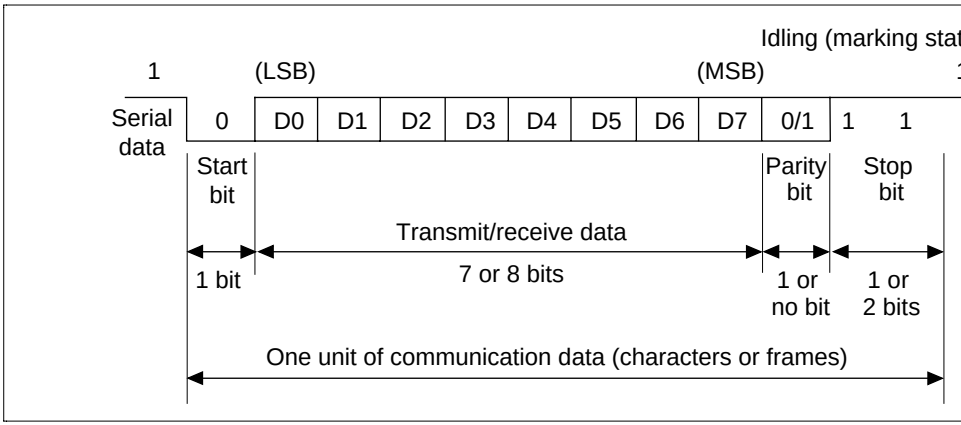


**Figure 14.2 Data Format in Asynchronous Communication (Example: 8-bit Da
Parity and Two Stop Bits)**

0	1	0	1	START	8-Bit data		P	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
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 cloc from the SCK pin can be selected as the SCI transmit/receive clock. The clock source is s by the C/$\overline{A}$ bit in the serial mode register (SMR) and bits CKE1 and CKE0 in the serial c register (SCR) (table 14.9).

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 numbers 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).

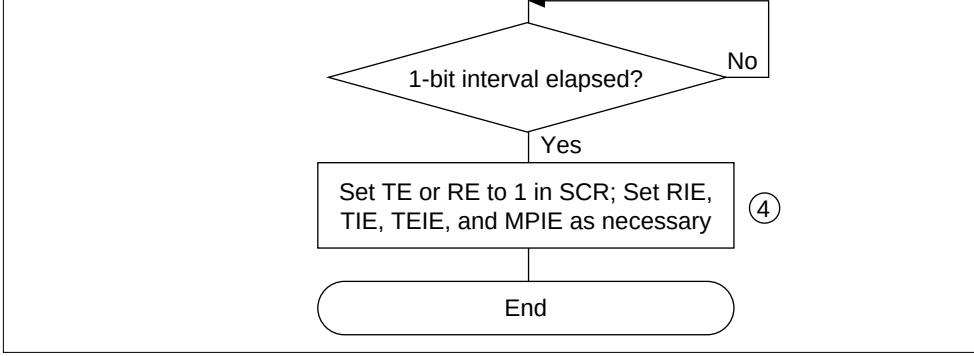


Figure 14.4 Sample Flowchart for SCI Initialization

Transmitting Serial Data (Asynchronous Mode): Figure 14.5 shows a sample flowcha transmitting serial data. The procedure is as follows (the steps correspond to the numbers 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 t TDRE bit is 1, then write transmit data in the transmit data register (TDR) and clear T to 0.
3. Continue transmitting serial data: Read the TDRE bit to check whether it is safe to wr reads 1); if so, write data in TDR, then clear TDRE to 0. When the DMAC or the DT started by a transmit-data-empty interrupt request (TxI) in order to write data in TDR, TDRE bit is checked and cleared automatically.
4. To output a break at the end of serial transmission, first clear the port data register (D then clear the TE to 0 in SCR and use the PFC to establish the TxD pin as an output p

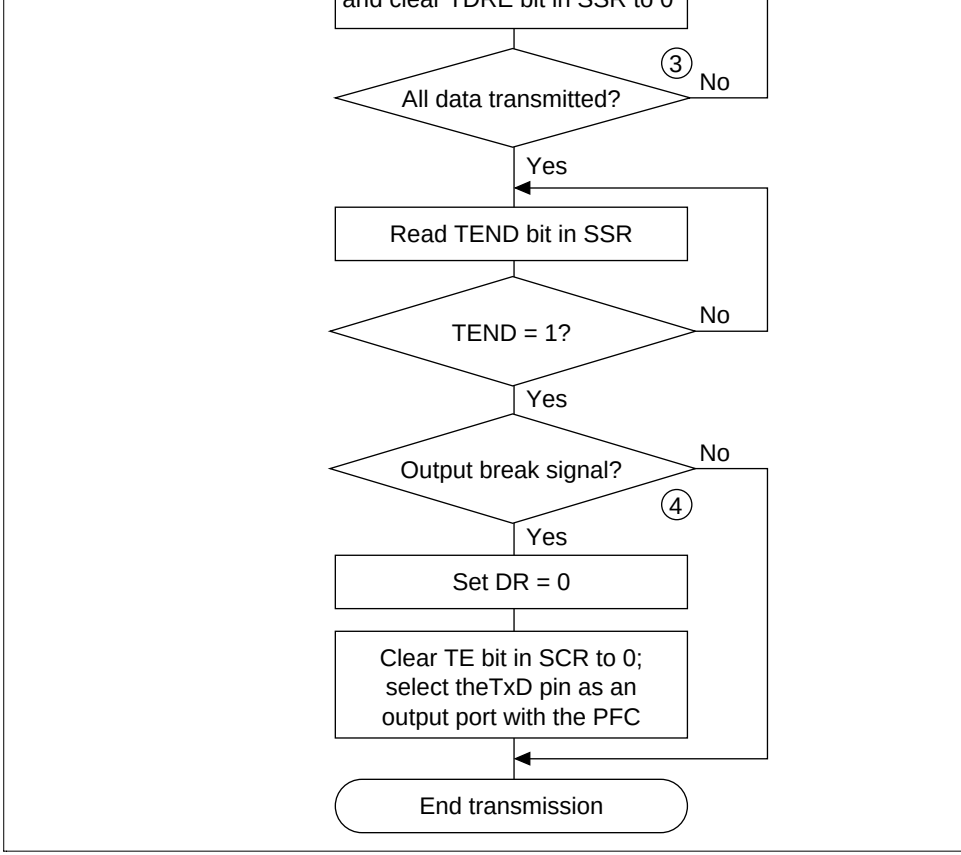


Figure 14.5 Sample Flowchart for Transmitting Serial Data

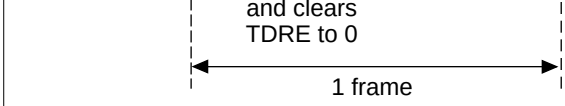
is output. Formats in which neither a parity bit nor a multiprocessor bit is output c
be selected.

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 load
data from the TDR into the TSR, outputs the stop bit, then begins serial transmission o
next frame. If TDRE is 1, the SCI sets the TEND bit to 1 in the SSR, outputs the stop
continues output of 1 bits (marking). If the transmit-end interrupt enable bit (TEIE) in
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.



Example: 8-bit data with parity and one stop bit

Figure 14.6 SCI Transmit Operation in Asynchronous Mode

Receiving Serial Data (Asynchronous Mode): Figures 14.7 and 14.8 show a sample fl for receiving serial data. The procedure is as follows (the steps correspond to the numbe 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 and FER bits of the SSR to identify the error. After executing the necessary error ha clear ORER, PER, and FER all to 0. Receiving cannot resume if ORER, PER, or FE set to 1. When a framing error occurs, the RxD pin can be read to detect the break st
3. SCI status check and receive-data read: Read the serial status register (SSR), check t is set to 1, then read receive data from the receive data register (RDR) and clear RDF The RxI interrupt can also be used to determine if the RDRF bit has changed from 0
4. Continue receiving serial data: Read the RDR and RDRF bit and clear RDRF to 0 be stop bit of the current frame is received. If the DMAC or the DTC is started by a rec full interrupt (RxI) to read RDR, the RDRF bit is cleared automatically so this step i unnecessary.

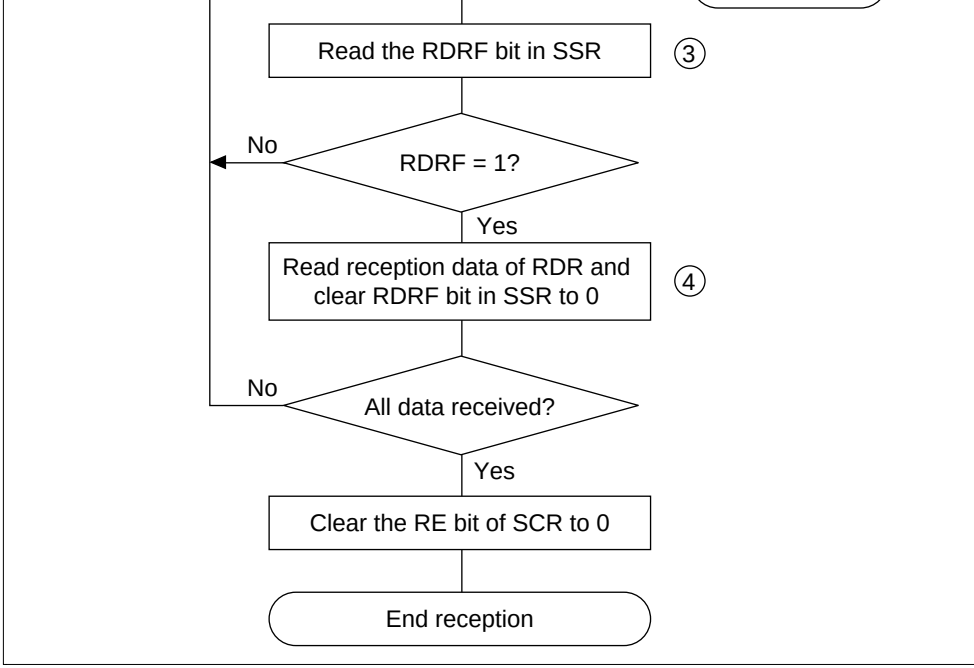


Figure 14.7 Sample Flowchart for Receiving Serial Data (1)

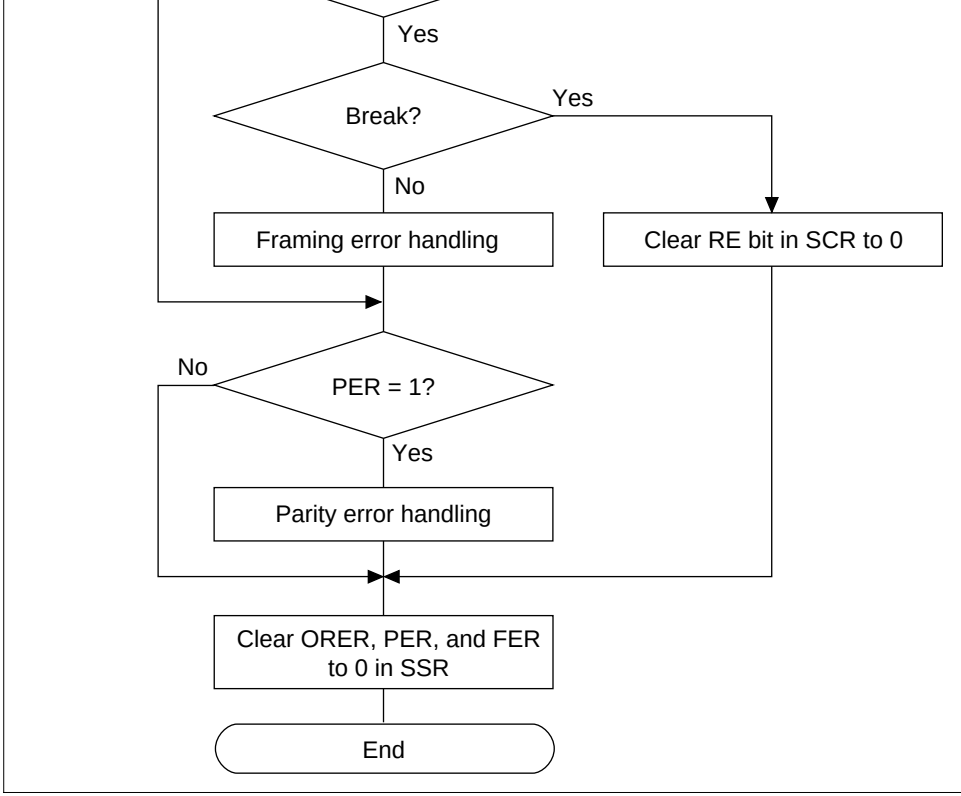


Figure 14.8 Sample Flowchart for Receiving Serial Data (2)

RDR.

If the data passes these checks, the SCI sets RDRF to 1 and stores the received data in RDR. If one of the checks fails (receive error), the SCI operates as indicated in table 1

Note: When a receive error occurs, further receiving is disabled. While receiving, the 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 SCR, the SCI requests a receive-data-full interrupt (RxI). If one of the error flags (OR PER, or FER) is set to 1 and the receive-data-full interrupt enable bit (RIE) in the SCI 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 l from RSR into RDI
Framing error	FER	Stop bit is 0	Receive data loade RSR into RDR
Parity error	PER	Parity of receive data differs from even/odd parity setting in SMR	Receive data loade RSR into RDR

clears RDRF to 0. request.

Example: 8-bit data with parity and one stop bit.

Figure 14.9 SCI Receive Operation

14.3.3 Multiprocessor Communication

The multiprocessor communication function enables several processors to share a single communication line for sending and receiving data. The processors communicate in the asynchronous mode using a format with an additional multiprocessor bit (multiprocesso

In multiprocessor communication, each receiving processor is addressed by a unique ID communication cycle consists of an ID-sending cycle that identifies the receiving proces data-sending cycle. The multiprocessor bit distinguishes ID-sending cycles from data-se cycles. The transmitting processor starts by sending the ID of the receiving processor wi wants to communicate as data with the multiprocessor bit set to 1. Next the transmitting sends transmit data with the multiprocessor bit cleared to 0.

Receiving processors skip incoming data until they receive data with the multiprocessor 1. When they receive data with the multiprocessor bit set to 1, receiving processors com data with their IDs. The receiving processor with a matching ID continues to receive fur incoming data. Processors with IDs not matching the received data skip further incomin until they again receive data with the multiprocessor bit set to 1. Multiple processors can receive data in this way.

Figure 14.10 shows the example of communication among processors using the multipr format.

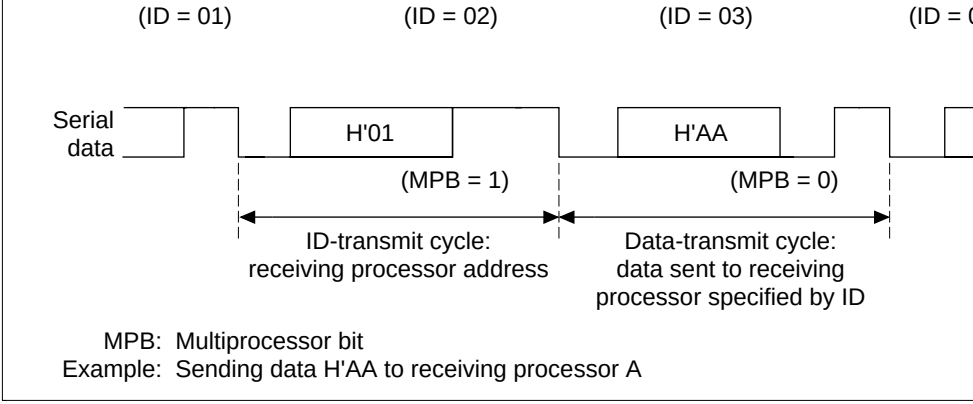


Figure 14.10 Communication among Processors Using Multiprocessor Form

Transmitting Multiprocessor Serial Data: Figure 14.11 shows a sample flowchart for transmitting multiprocessor serial data. The procedure is as follows (the steps correspond 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 t TDRE bit is 1, then write transmit data in the transmit data register (TDR). Also set M (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 wr reads 1); if so, write data in TDR, then clear TDRE to 0. When the DMAC or the DT started by a transmit-data-empty interrupt request (TxI) to write data in TDR, the TDR checked and cleared automatically.
4. Output a break at the end of serial transmission: Set the data register (DR) of the port clear TE to 0 in SCR and set the TxD pin function as output port with the PFC.

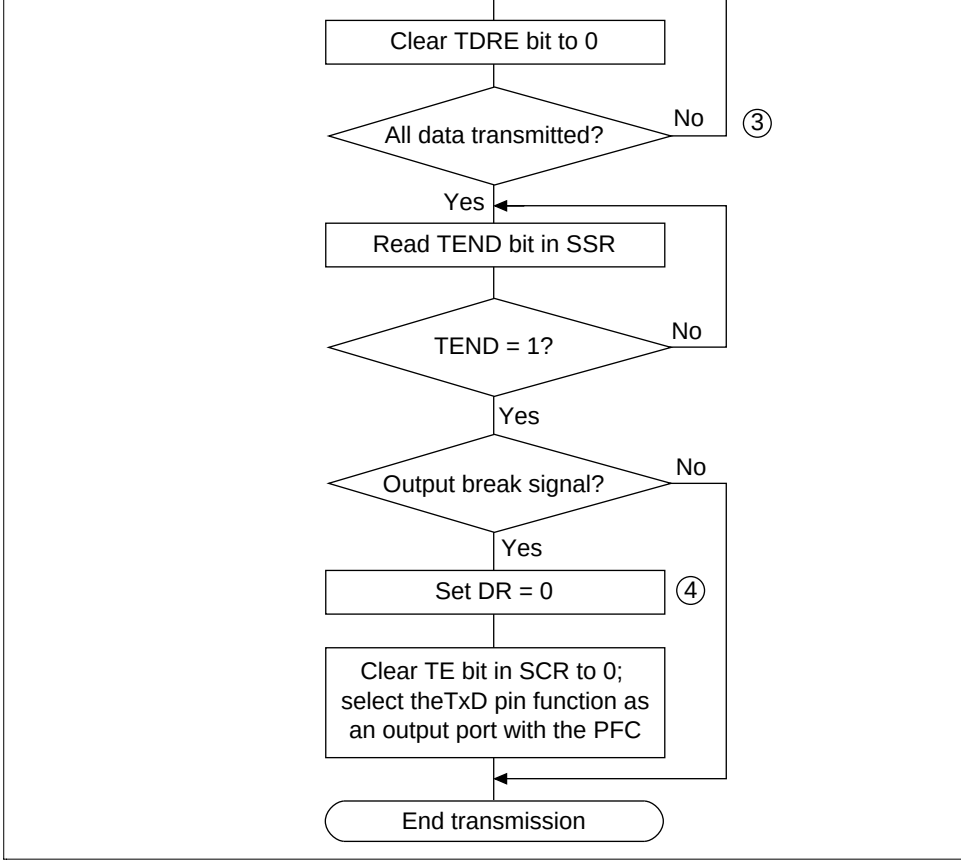


Figure 14.11 Sample Flowchart for Transmitting Multiprocessor Serial Da

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 load from the TDR into the TSR, outputs the stop bit, then begins serial transmission of the frame. If TDRE is 1, the SCI sets the TEND bit in the SSR to 1, outputs the stop bit, t continues output of 1 bits in the marking state. If the transmit-end interrupt enable bit 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.

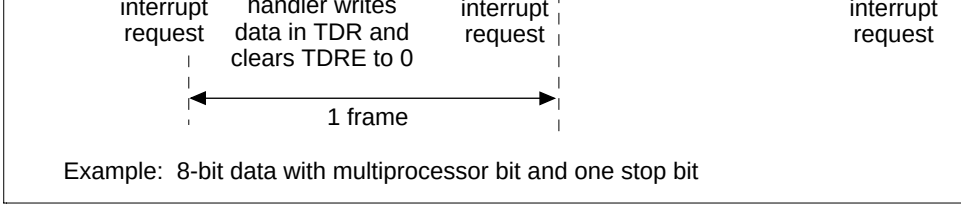


Figure 14.12 SCI Multiprocessor Transmit Operation

Receiving Multiprocessor Serial Data: Figure 14.13 shows a sample flowchart for rec multiprocessor serial data. The procedure for receiving multiprocessor serial data is liste

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), RDRF is set to 1, then read data from the receive data register (RDR) and compare w processor's own ID. If the ID does not match the receive data, set MPIE to 1 again a 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 bits in SSR to identify the error. After executing the necessary error processing, clea ORER and FER to 0. Receiving cannot resume if ORER or FER remain set to 1. Wh 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 rea from the receive data register (RDR).

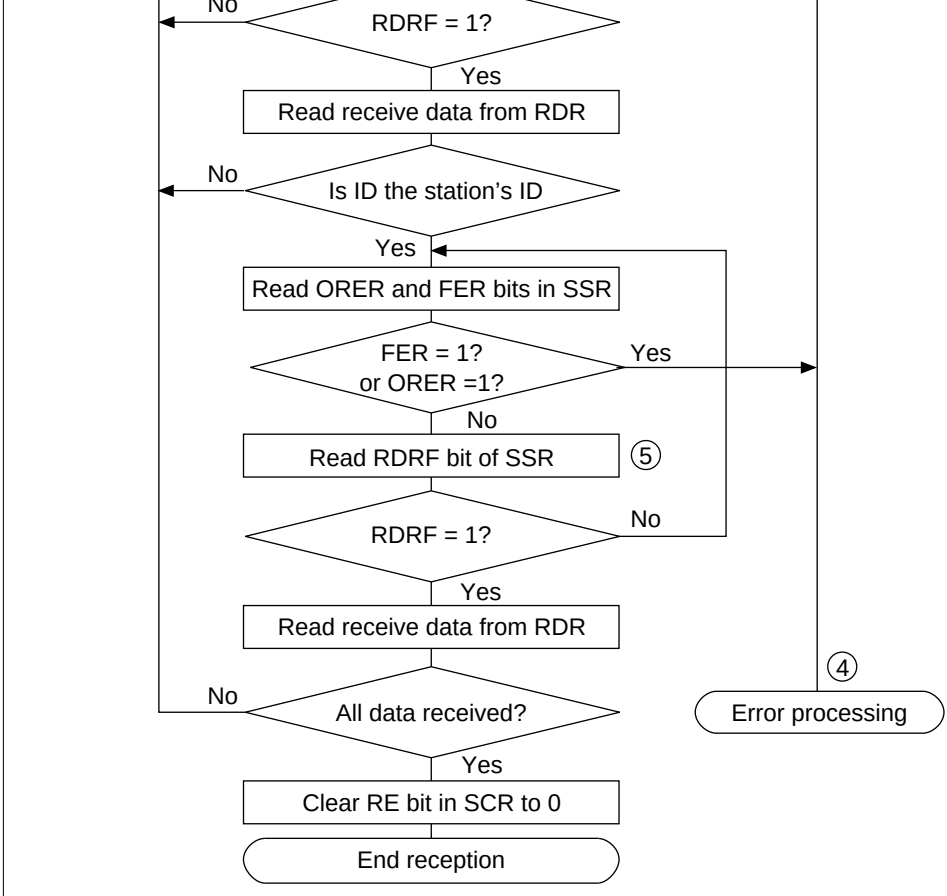


Figure 14.13 Sample Flowchart for Receiving Multiprocessor Serial Data

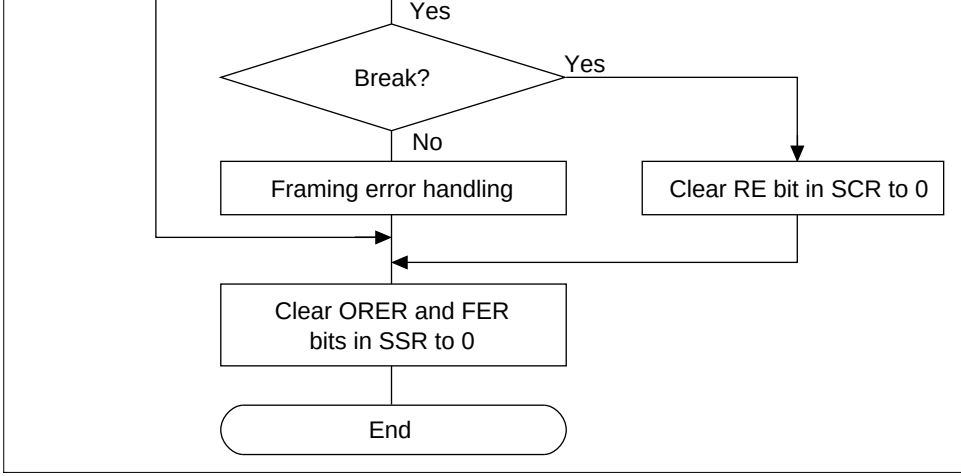


Figure 14.13 Sample Flowchart for Receiving Multiprocessor Serial Data (c

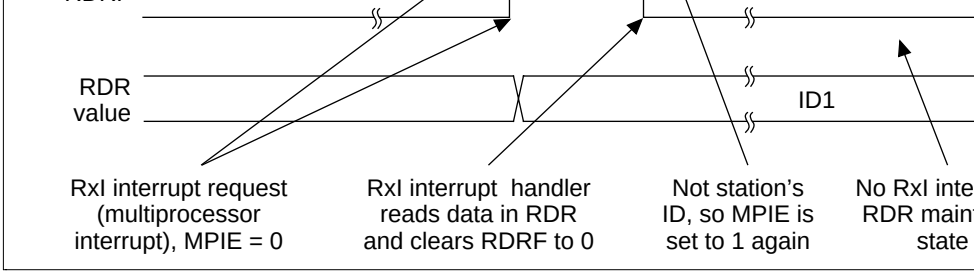


Figure 14.14 SCI Receive Operation (ID Does Not Match)

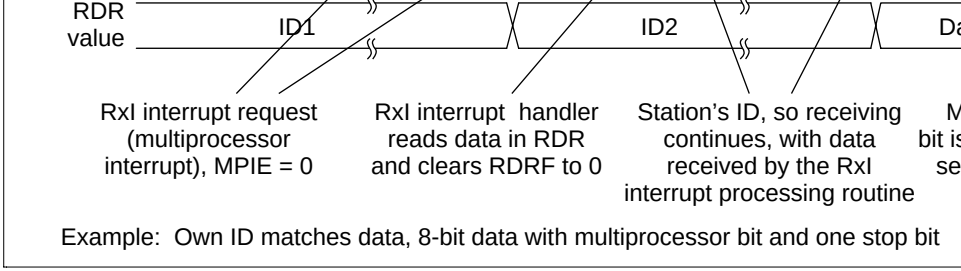


Figure 14.15 Example of SCI Receive Operation (ID Matches)

14.3.4 Clock Synchronous Operation

In the clock synchronous mode, the SCI transmits and receives data in synchronization w
pulses. This mode is suitable for high-speed serial communication.

The SCI transmitter and receiver are independent, so full duplex communication is possi
sharing the same clock. The transmitter and receiver are also double buffered, so continu
transmitting or receiving is possible by reading or writing data while transmitting or rec
progress.

Figure 14.16 shows the general format in clock synchronous serial communication.

In clock synchronous serial communication, each data bit is output on the communication from one falling edge of the serial clock to the next. Data are guaranteed valid at the risin the serial clock. In each character, the serial data bits are transmitted in order from the LS to the MSB (last). After output of the MSB, the communication line remains in the state o MSB. In the clock synchronous mode, the SCI transmits or receives data by synchronizin the falling edge of the synchronization clock.

Communication Format: The data length is fixed at eight bits. No parity bit or multipro can be added.

Clock: An internal clock generated by the on-chip baud rate generator or an external cloc from the SCK pin can be selected as the SCI transmit/receive clock. The clock source is s by the C/$\overline{A}$ bit in the serial mode register (SMR) and bits CKE1 and CKE0 in the serial co register (SCR). See table 14.9.

When the SCI operates on an internal clock, it outputs the clock signal at the SCK pin. Ei pulses are output per transmitted or received character. When the SCI is not transmitting receiving, the clock signal remains in the high state.

Note: An overrun error occurs only during the receive operation, and the sync clock is o until the RE bit is cleared to 0. When you want to perform a receive operation in character units, select external clock for the clock source.

SCI Initialization (Clock Synchronous Mode): Before transmitting or receiving, softwa clear the TE and RE bits to 0 in the serial control register (SCR), then initialize the SCI as

When changing the mode or communication format, always clear the TE and RE bits to 0 following the procedure given below. Clearing TE to 0 sets TDRE to 1 and initializes the shift register (TSR). Clearing RE to 0, however, does not initialize the RDRF, PER, FER, ORER flags and receive data register (RDR), which retain their previous contents.

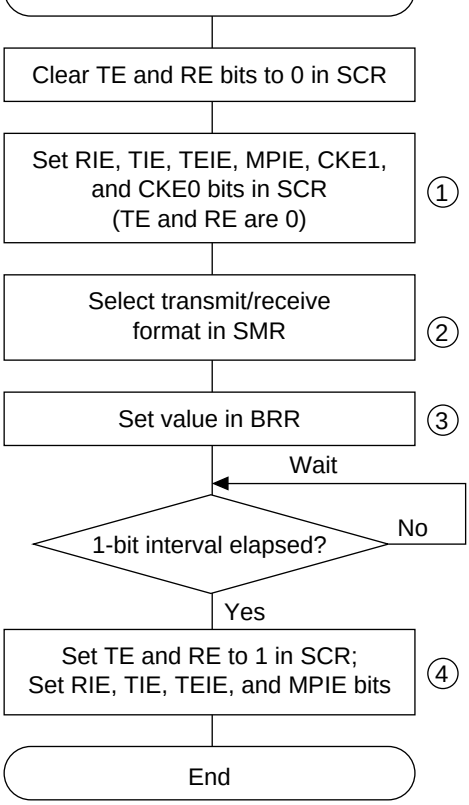


Figure 14.17 Sample Flowchart for SCI Initialization

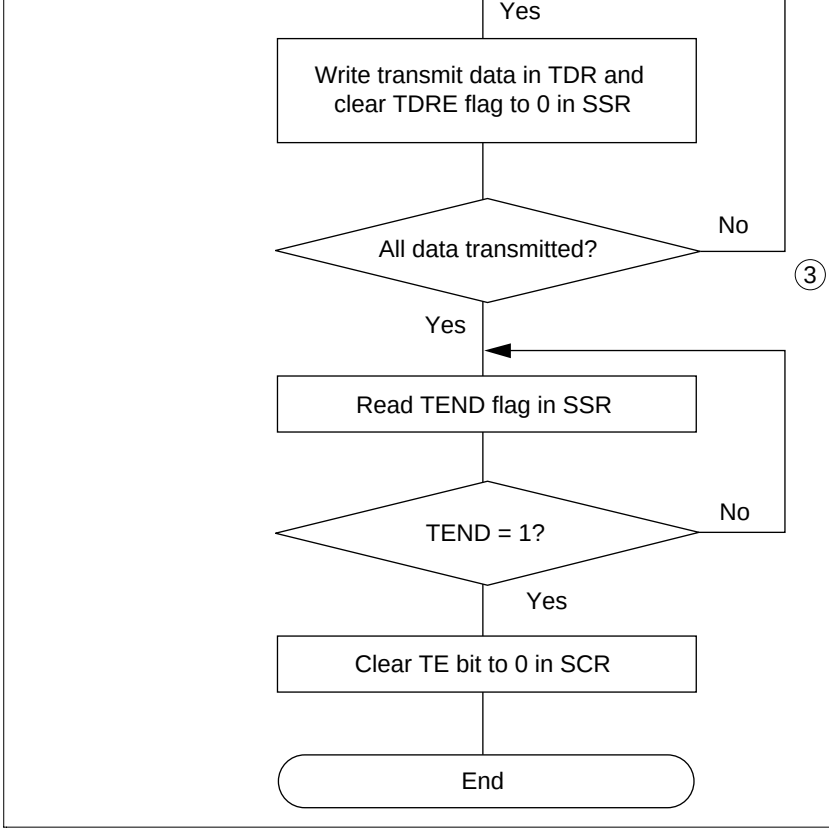


Figure 14.18 Sample Flowchart for Serial Transmitting

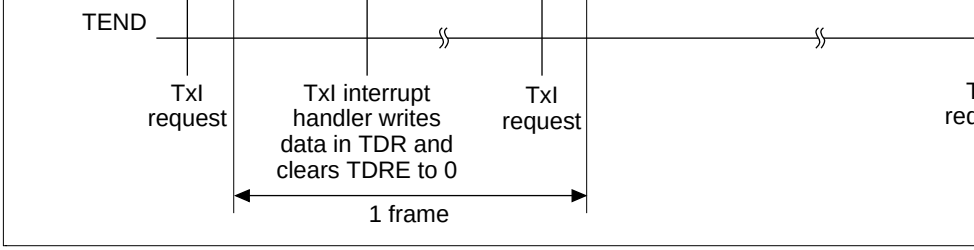


Figure 14.19 Example of SCI Transmit Operation

SCI serial transmission operates as follows.

1. The SCI monitors the TDRE bit in the SSR. When TDRE is cleared to 0 the SCI recog that the transmit data register (TDR) contains new data and loads this data from the T 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 transmitting. If the transmit-data-empty interrupt enable bit (TIE) in the SCR is set to 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 clock source is selected, the SCI outputs data in synchronization with the input clock. 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 data from the TDR into the TSR, then begins serial transmission of the next frame. If 1, the SCI sets the TEND bit in the SSR to 1, transmits the MSB, then holds the transm pin (TxD) in the MSB state. If the transmit-end interrupt enable bit (TEIE) in the SCR 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.

The RxI interrupt can also be used to determine if the RDRF bit has changed from 0

4. Continue receiving serial data: Read RDR, and clear RDRF to 0 before the frame MS
of the current frame is received. If the DMAC or the DTC is started by a receive-data
interrupt (RxI) to read RDR, the RDRF bit is cleared automatically so this step is unn

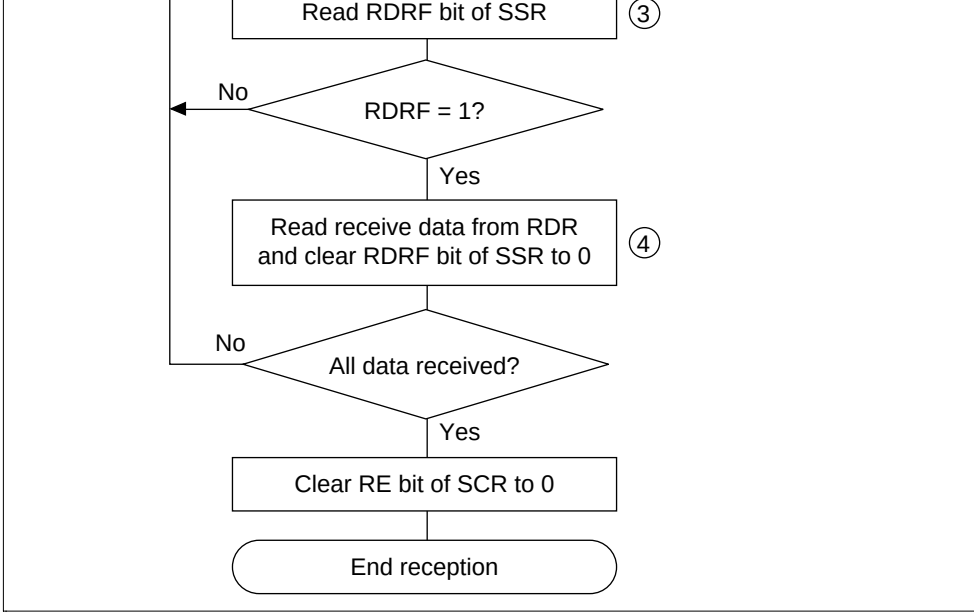


Figure 14.20 Sample Flowchart for Serial Receiving (1)

Figure 14.22 shows an example of the SCI receive operation.

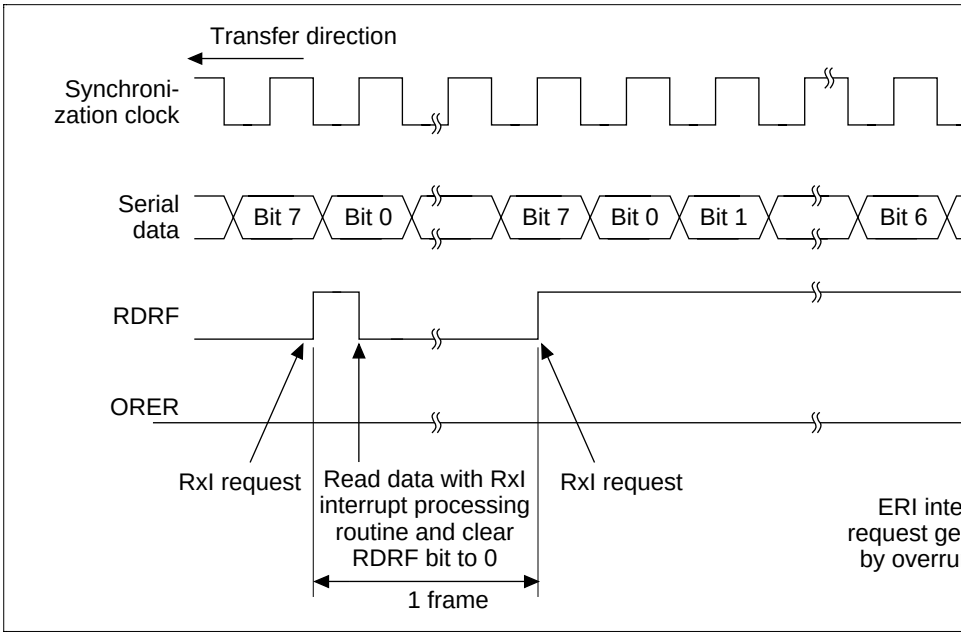


Figure 14.22 Example of SCI Receive Operation

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 receiv data, the SCI checks that RDRF is 0 so that receive data can be loaded from the RSR RDR. If this check passes, the SCI sets RDRF to 1 and stores the received data in the the check does not pass (receive error), the SCI operates as indicated in table 14.11 a further transmission or reception is possible. If the error flag is set to 1, the RDRF bi

TDRE bit is 1, then write transmit data in the transmit data register (TDR) and clear T
0. The TxI interrupt can also be used to determine if the TDRE bit has changed from 0

3. Receive error handling: If a receive error occurs, read the ORER bit in SSR to identify
error. After executing the necessary error processing, clear ORER to 0. Transmitting/r
cannot resume if ORER remains set to 1.
4. SCI status check and receive data read: Read the serial status register (SSR), check tha
is set to 1, then read receive data from the receive data register (RDR) and clear RDR
The RxI interrupt can also be used to determine if the RDRF bit has changed from 0 to
5. Continue transmitting and receiving serial data: Read the RDRF bit and RDR, and cle
to 0 before the frame MSB (bit 7) of the current frame is received. Also read the TDR
check whether it is safe to write (if it reads 1); if so, write data in TDR, then clear TDI
before the MSB (bit 7) of the current frame is transmitted. When the DMAC or the DT
started by a transmit-data-empty interrupt request (TxI) to write data in TDR, the TDR
checked and cleared automatically. When the DMAC or the DTC is started by a receiv
full interrupt (RxI) to read RDR, the RDRF bit is cleared automatically.

Note: In switching from transmitting or receiving to simultaneous transmitting and rece
simultaneously clear both TE and RE to 0, then simultaneously set both TE and R

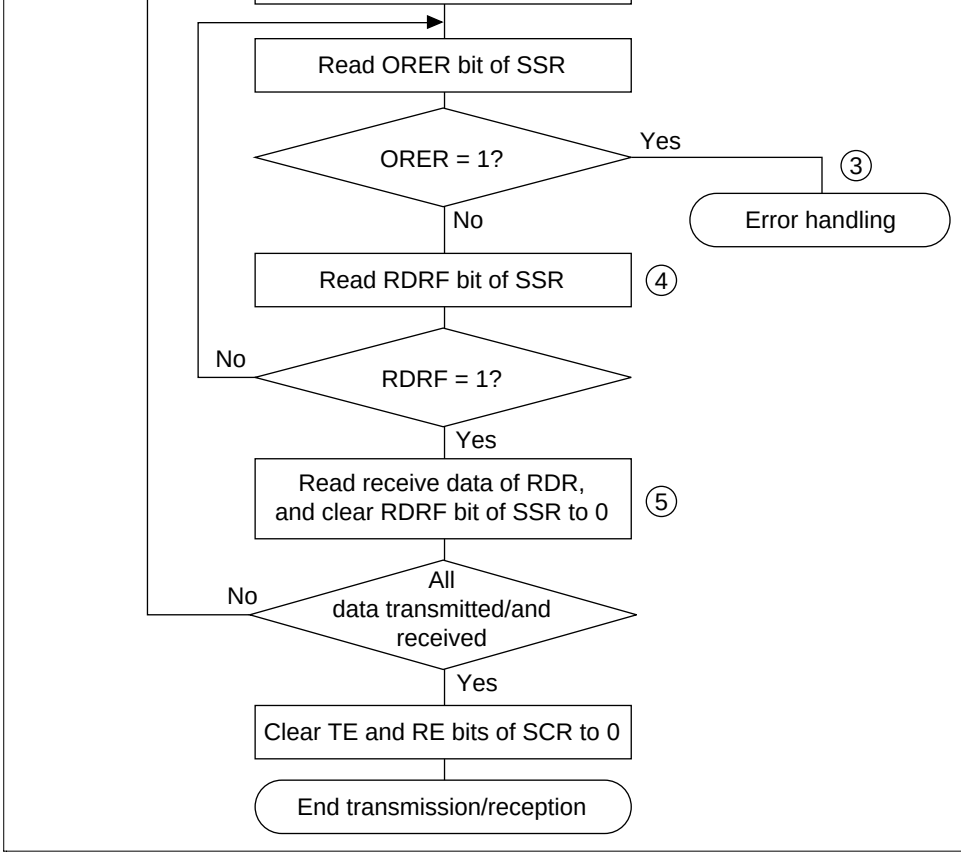


Figure 14.23 Sample Flowchart for Serial Transmission

data register (RDR).

ERI is requested when the ORER, PER, or FER bit in the SSR is set to 1. ERI cannot star
DMAC or the DTC.

TEI is requested when the TEND bit in the SSR is set to 1. TEI cannot start the DMAC o
DTC. Where the TxI interrupt indicates that transmit data writing is enabled, the TEI inte
indicates that the transmit operation is complete.

Table 14.12 SCI Interrupt Sources

Interrupt Source	Description	DMAC/DTC Activation
ERI	Receive error (ORER, PER, or FER)	No
RxI	Receive data full (RDRF)	Yes
TxI	Transmit data empty (TDRE)	Yes
TEI	Transmit end (TEND)	No

14.5.2 Simultaneous Multiple Receive Errors

Table 14.13 indicates the state of the SSR status flags when multiple receive errors occu
simultaneously. When an overrun error occurs, the RSR contents cannot be transferred t
RDR, so receive data is lost.

Table 14.13 SSR Status Flags and Transfer of Receive Data

	SSR Status Flags				Receiv Transf
Receive Error Status	**RDRF**	**ORER**	**FER**	**PER**	**RSR →**
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.

first clear the DR to 0, then establish the TxD pin as an output port using the PFC. When
cleared to 0, the transmission section is initialized regardless of the present transmission s

14.5.5 Receive Error Flags and Transmitter Operation (Clock Synchronous Mod

When a receive error flag (ORER, PER, or FER) is set to 1, the SCI will not start transmi
even if TDRE is set to 1. Be sure to clear the receive error flags to 0 before starting to tra
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 M

In the asynchronous mode, the SCI operates on a base clock of 16 times the bit rate frequ
receiving, the SCI synchronizes internally with the falling edge of the start bit, which it sa
the base clock. Receive data is latched on the rising edge of the eighth base clock pulse (f
14.24).

Figure 14.24 Receive Data Sampling Timing in Asynchronous Mode

The receive margin in the asynchronous mode can therefore be expressed as:

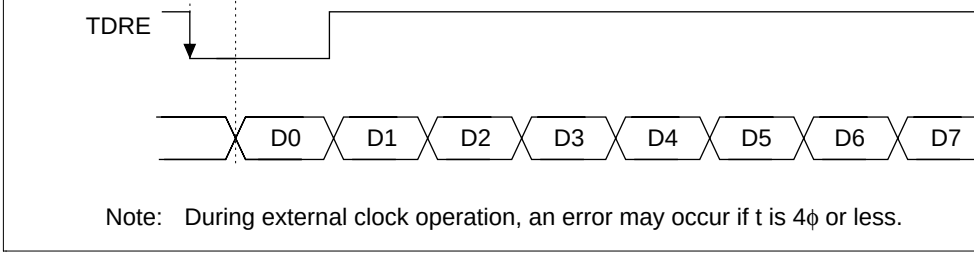
$$M = \left| \left(0.5 - \frac{1}{2N}\right) - (L - 0.5)F - \frac{|D - 0.5|}{N}(1 + F) \right| \times 100\%$$

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%.



Note: During external clock operation, an error may occur if t is 4ϕ or less.

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 cha from 0 to 1.
- When receiving, RDRF is 1 when RE is set to zero 2.5–3.5 clocks after the rising edg 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 bit SCK output, but it cannot be copied to RDR.

- Eight input channels
- Analog conversion voltage range setting is selectable
  — Using the reference voltage pin (AVref) as an analog standard voltage (Vref), co
  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
  conversions

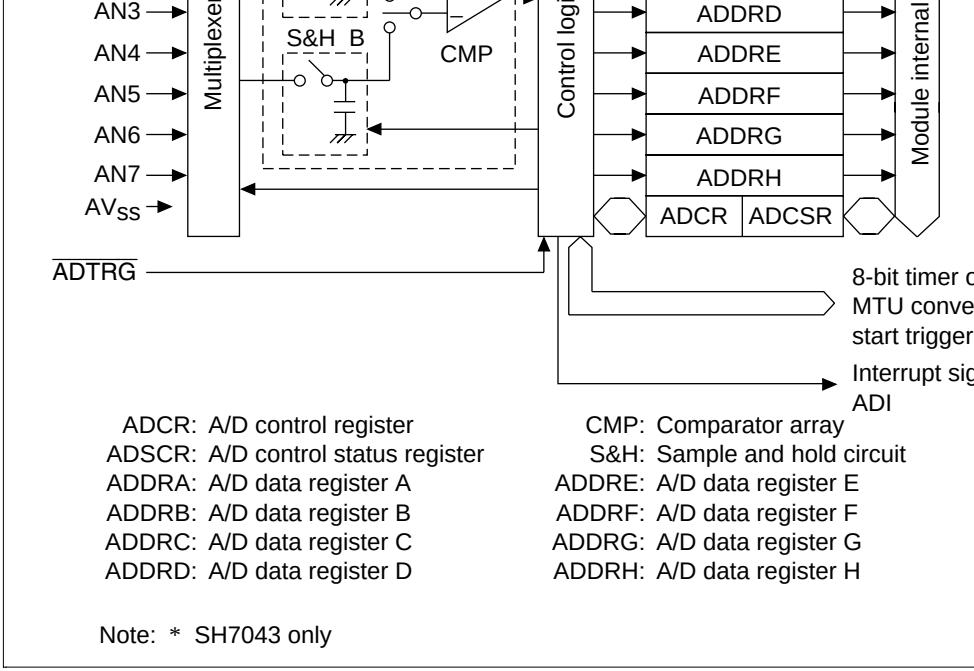


Figure 15.1 High Speed A/D Converter Block Diagram

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. T
AV_{ref} pin is for the A/D conversion standard voltage.

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 conversi

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	Acce
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.

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 initialize ADDR.

Bit:	15	14	13	12	11	10	9
	—	—	—	—	—	—	AD9
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1
	AD7	AD6	AD5	AD4	AD3	AD2	AD1
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R

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 an
indicate status.

The ADCSR is initialized to H'00 by power-on reset or in standby mode. Manual reset d
initialize ADCSR.

Bit:	7	6	5	4	3	2	1
	ADF	ADIE	ADST	CKS	GRP	CH2	CH1
Initial value:	0	0	0	0	0	0	0
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 6—A/D Interrupt Enable (ADIE): Enables or disables interrupt requests (ADI) aft
  conversion ends. Set the ADIE bit while conversion is suspended.

Bit 6: ADIE	Description
0	Disables interrupt requests (ADI) after A/D conversion ends (in 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
  A/D conversions.

  The ADST bit can be set to 1 by software, timer conversion start triggers, or an A/D e
  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 conversion for the designated channel ends. Scan mode: Start A/D conversion. Continuous conversion until cleared by software.

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, sele 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 (GR
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

Setting the PWR bit to 1 sets high speed start mode, and a 0 sets to low power conversi 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 prohi 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 star
1	0	Reserved
1	1	Enables A/D conversion start by external trigger pin ($\overline{A}$

- Bit 3—Scan Mode (SCAN): Selects either single mode or scan mode for the A/D con operation mode. See section 15.4, Operation, for details on single mode and scan mod operation.

  Set the SCAN bit only while conversion is halted.

Bit 3: SCAN	Description
0	Single mode (initial value)
1	Scan mode

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 re ADDRA → ADDRB (ADDRB is the buffer register)
1	0	ADDRA and ADDRC, also ADDRB and ADDRD buff operation: conversion result 1 → ADDRA → ADDRC conversion result 2 → ADDRB → ADDRD (ADDRC a ADDRD are buffer registers)
1	1	ADDRA–ADDRD buffer operation: conversion result ADDRA → ADDRB → ADDRC → ADDRD (ADDRB– 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 maste
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 ma
at a time. In byte unit reads, the contents of the most significant eight bits (AD9–AD2) c
converted data (AD9–AD0) are transferred to the bus master.

Figures 15.2 and 15.3 shows an example of the ADDR read operation.

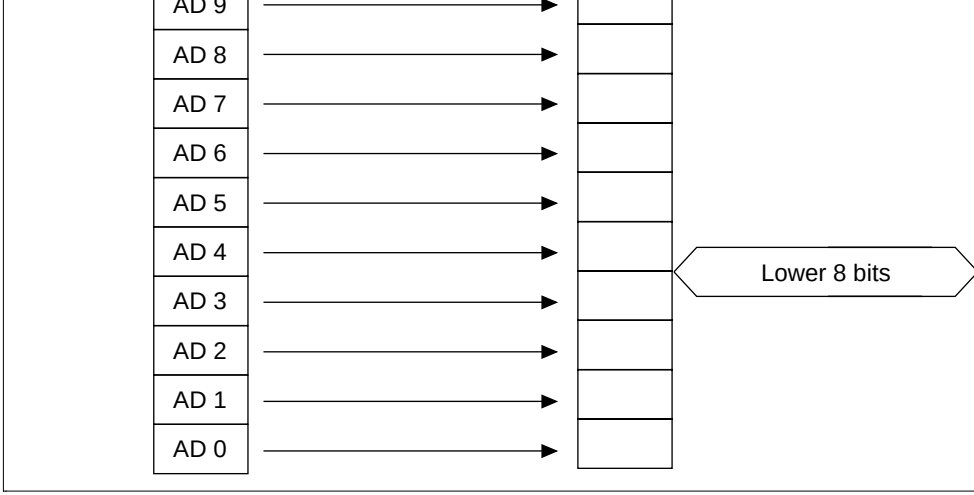


Figure 15.2 ADDR Read Operation (1)

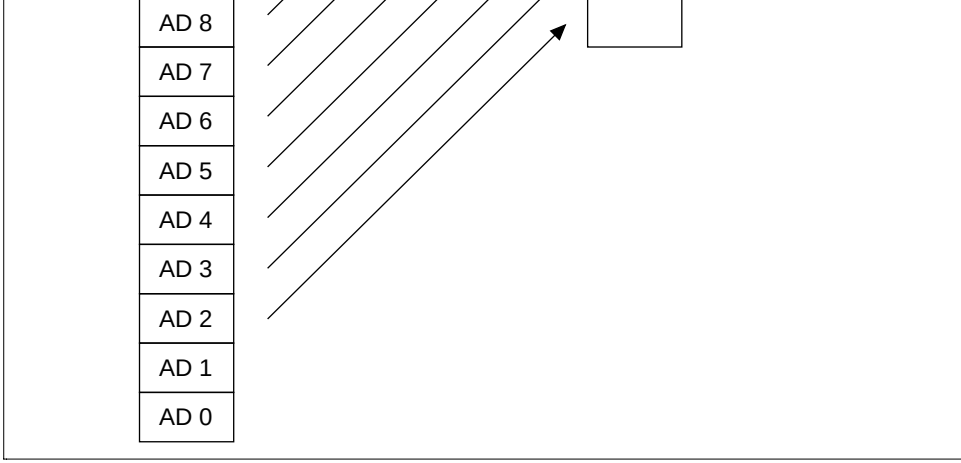


Figure 15.3 ADDR Read Operation (2)

- Software, a timer conversion start trigger (MTU), or an ADTRG input can be selected conversion start condition.
- High speed start mode or low power conversion mode can be selected for A/D conver using the PWR bit setting.
- When changing the operation mode or input channel, rewrite the ADCSR, ADCR whi ADST bit is cleared to 0. After rewriting the ADCSR, ADCR, A/D conversion will be 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 com 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 con start conditions. The ADST bit is held to 1 during the A/D conversion and is automaticall 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 ti ADI interrupt request is generated. The ADF flag is cleared by reading the ADCSR, then 0.

Figure 15.4 shows an example of operation in the select-single mode when AN1 is selecte

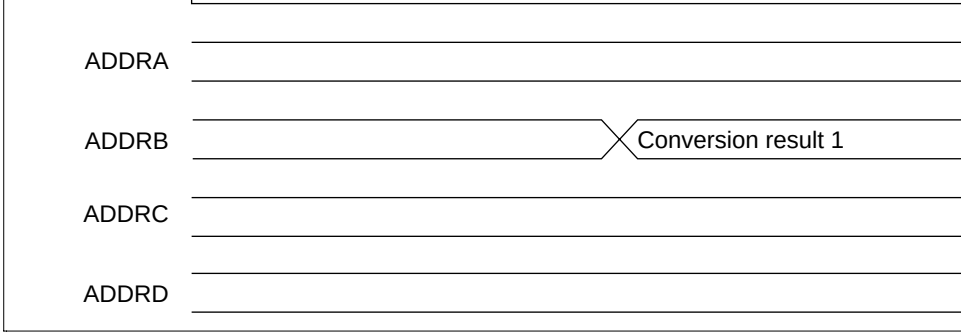


Figure 15.4 A/D Converter Operation Example (Select-Single Mode)

15.4.2 Select-Scan Mode

Choose select-scan mode when doing repeated A/D conversions for one channel. This is 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 co start conditions. The ADST bit is held to 1 until 0 cleared by software. A/D conversion f 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 ADI interrupt request is issued, and the A/D converter is halted. With the A/D converter mode due to an ADI interrupt request, conversion is restarted when the ADF flag is clea 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 selecte

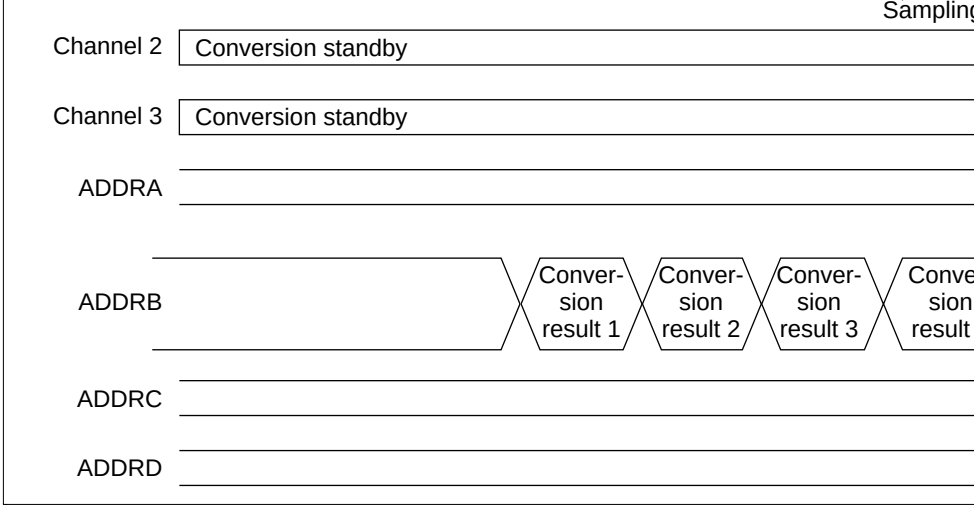


Figure 15.5 A/D Converter Operation Example (Select-Scan Mode)

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 con start conditions. The ADST bit is held to 1 during A/D conversion and is automatically cl 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 comp the ADIE bit is set to 1 at this time, an ADI interrupt request is generated. The ADF flag by reading the ADCSR then writing a 0.

Figure 15.6 shows an example of operation in the group-single mode when AN0–AN2 ar selected.

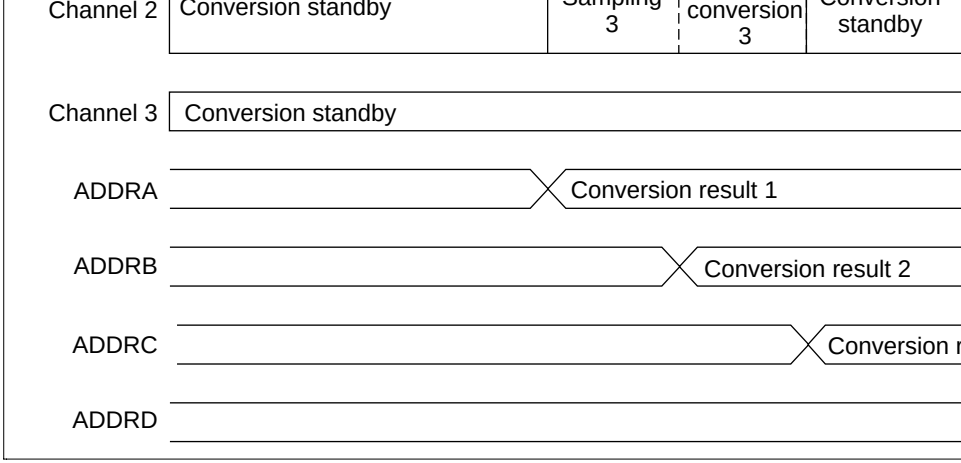


Figure 15.6 A/D Converter Operation Example (Group-Single Mode)

15.4.4 Group-Scan Mode

Choose group-scan mode when doing repeated A/D conversions for multiple channels. T
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 co
start conditions. The ADST bit is held to 1 until 0 cleared by software. A/D conversion f
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 i
channels. At this point, if the ADIE bit is set to 1, an ADI interrupt request is issued, and
converter is temporarily halted. With the A/D converter in stop mode due to an ADI inte
request, conversion is restarted when the ADF flag is cleared to 0. The ADF flag is clear
reading the ADCSR, then writing a 0.

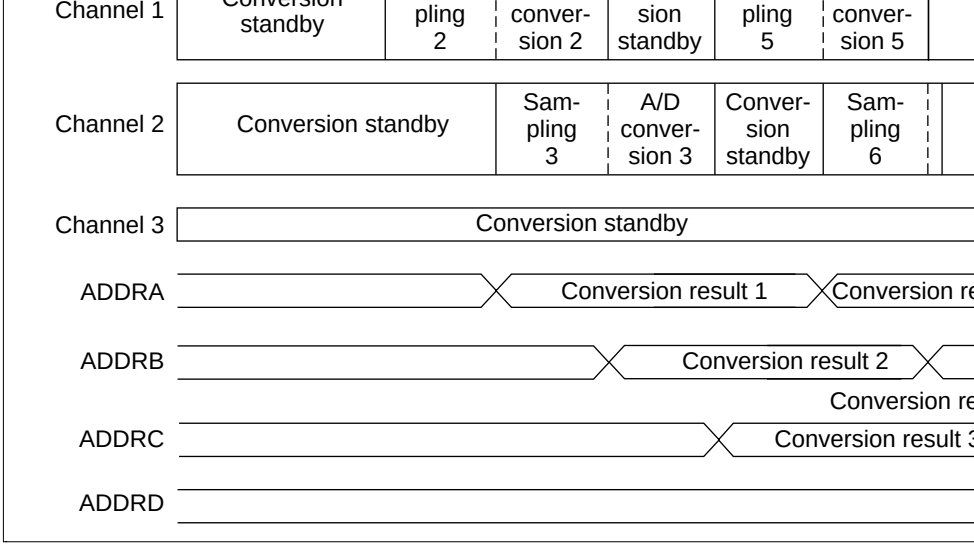


Figure 15.7 A/D Converter Operation Example (Group-Scan Mode)

15.4.5 Buffer Operation

When conversion ends on the relevant channel, the conversion result is stored in the ADD simultaneously, the previously stored result is transferred to another ADDR. Buffer opera 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 operat

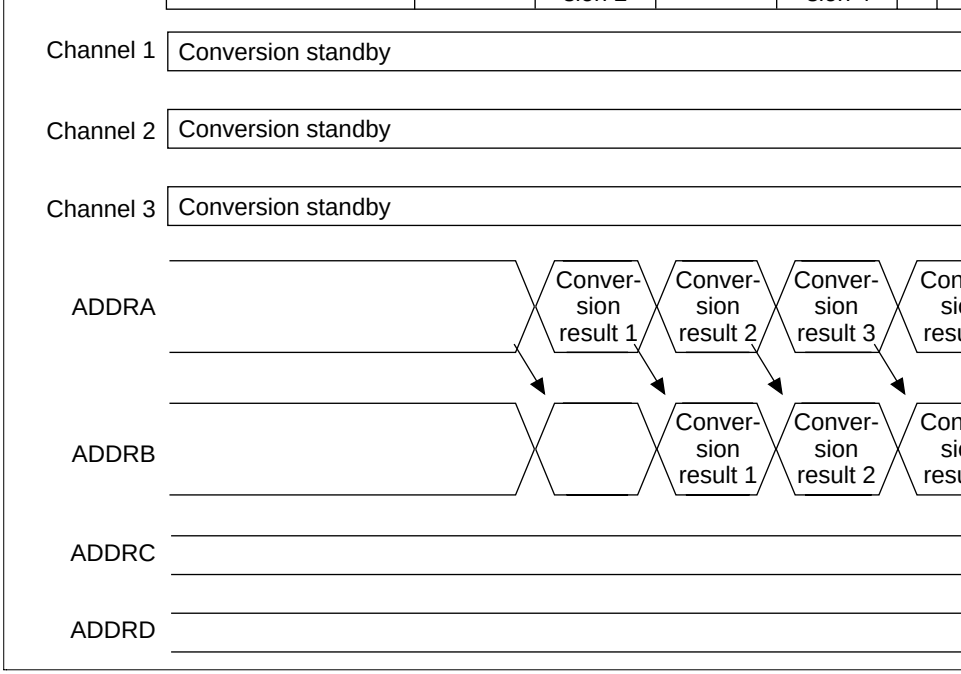


Figure 15.8 Buffer Operation Example (Select Scan Mode: Two-Stage One-C Operation, When CH2–CH0 = B'001)

Buffer-Only Operation: When performing conversion only on the analog input channe specified by the BUFE1 and BUFE0 bits, select group mode, and you can select the AD setting conditions with the CH2–CH0 bits.

Table 15.4 shows conversion during buffer operation and ADF flag setting conditions. T flag is set at the point in the table when the final conversion has ended. In single mode, is halted after the ADF flag is set to 1. In scan mode, conversion continues, and the conv is stored in sequence in the buffer registers specified by the BUFE1 and BUFE0 bits.

				(ADDRB)	
		1	AN0 2 times (ADDRB)		AN0 2 times (AD
	1	0	*	AN0, AN1 2 times (ADDRD)	AN0 3 times (AD
		1	*		AN0 4 times (AD
1	—	—	*	*	*

Note: * See table 15.5.

Combined Group Mode and Buffer Operation: Continuous conversion is possible on a input channels (AN0 and AN1) specified by bits BUFE1 and BUFE0 as well as AN4–AN setting of bits CH2–CH0.

Table 15.5 shows conversion during buffer operation and ADF flag setting conditions. Th flag is set at the point in the table when the final conversion has ended. In this case, conve performed on the analog input corresponding with the ADDR specified in the buffer regis example, when BUFE1 and BUFE0 = B'11 and CH2–CH0 = B'110, conversion results ar in ADDRA and ADDRE–ADDRG. Also, contents of ADDRA–ADDRC before the start conversion are transferred to ADDRB–ADDRD.

In single mode, conversion is halted after the ADF flag has been set to 1. Conversion con scan mode.

		(ADDRF)	(ADDRF)	(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 conversio interrupt, the ADF flag is cleared when the ADDR specified in table 15.4 or 15.5 has be

Resetting the Number of Buffer Operations: Clear the BUFE1 and BUFE0 bits to B'0 conversion standby mode or when the converter has been halted. The number of buffer o is cleared to 0.

Updating Buffer Operations: Clear the BUFE1 and BUFE0 bits to B'00 in conversion mode or when the converter has been halted. Thereafter, set BUFE1 and BUFE0, and th 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 i voltages on two channels at the same time. Simultaneous sampling is valid in group mo Channels for sampling are determined by the CH2 and CH1 bits of the RDSCR. The co are shown in table 15.6. For example, if GRP = 1 when CH2 and CH1 = B'11, sampling order in the following pairs: AN0, AN1→AN2, AN3→AN4, AN5→AN6, AN7. Sampl is shown in figure 15.9.

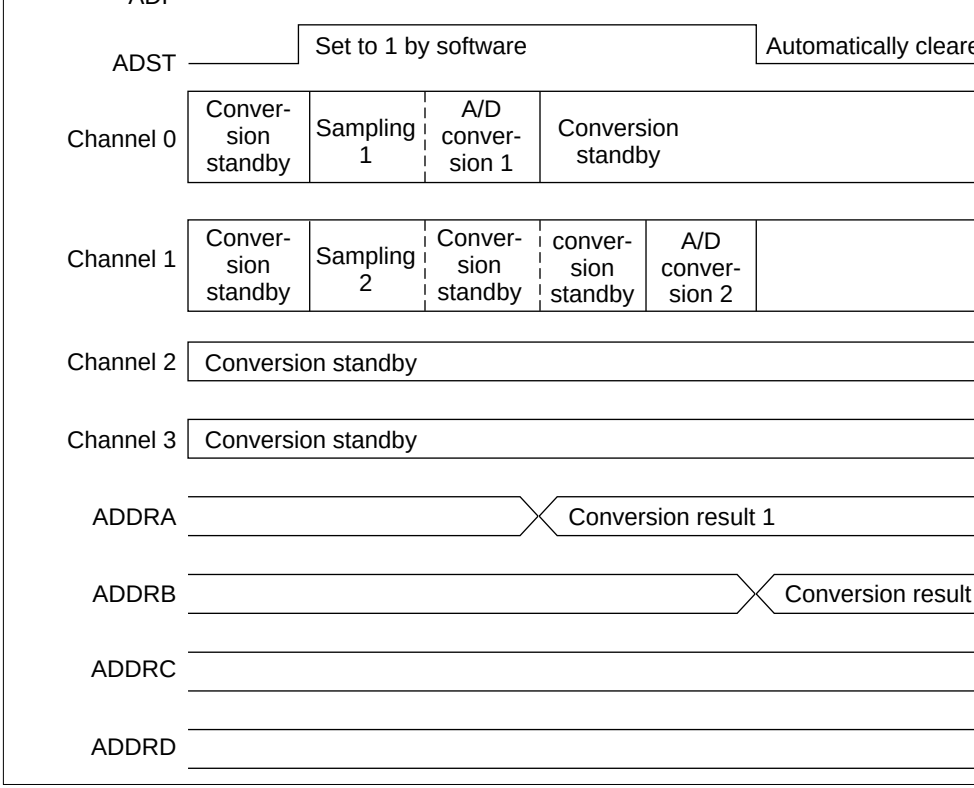


Figure 15.9 Simultaneous Sampling Operation (Group Single Mode)

active during the A/D conversion operation period in this mode, current consumption ca
reduced.

In high-speed start mode, ADST is cleared to 0 when A/D conversion ends. Power conti
supplied to the analog circuitry, and conversion-ready status is maintained. Conversion i
immediately by resetting ADST to 1. However, the first conversion after power-on begi
cycles after setting ADST. Clear the PWR bit to 0 to switch off the analog power supply
performing consecutive conversions, the second and later conversions are executed in 2
Because the analog circuit is always active in this mode, A/D conversion can be execute
speed.

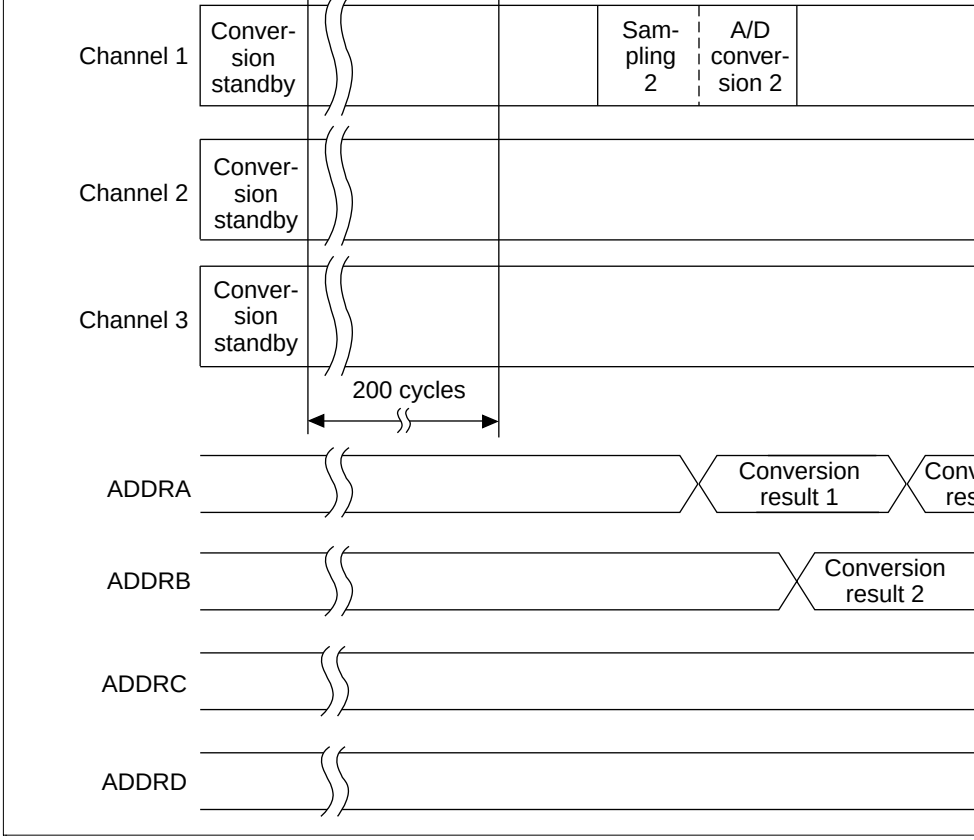


Figure 15.10 Conversion Start Operation (Low-Power Conversion Mode)

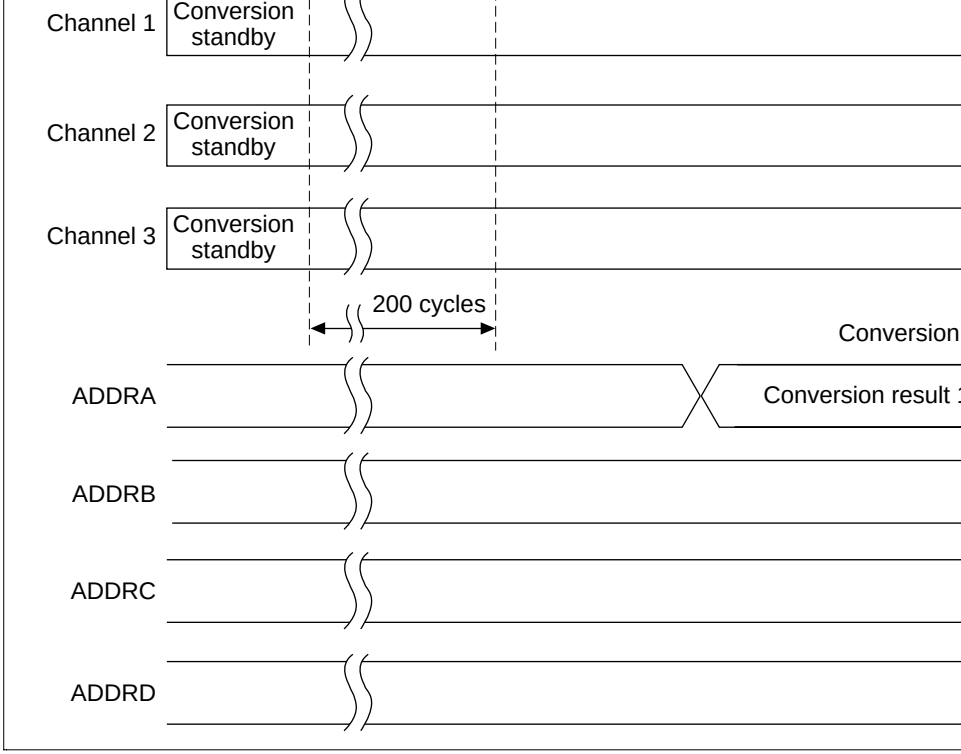


Figure 15.11 Conversion Start Operation (High-Speed Start Mode)

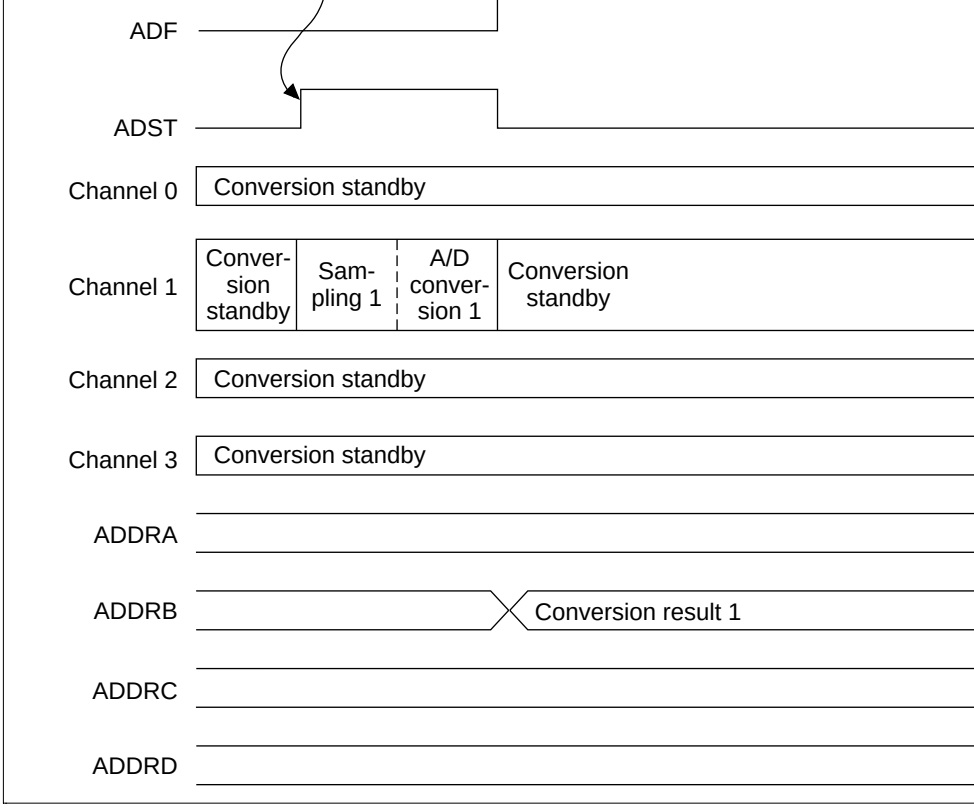
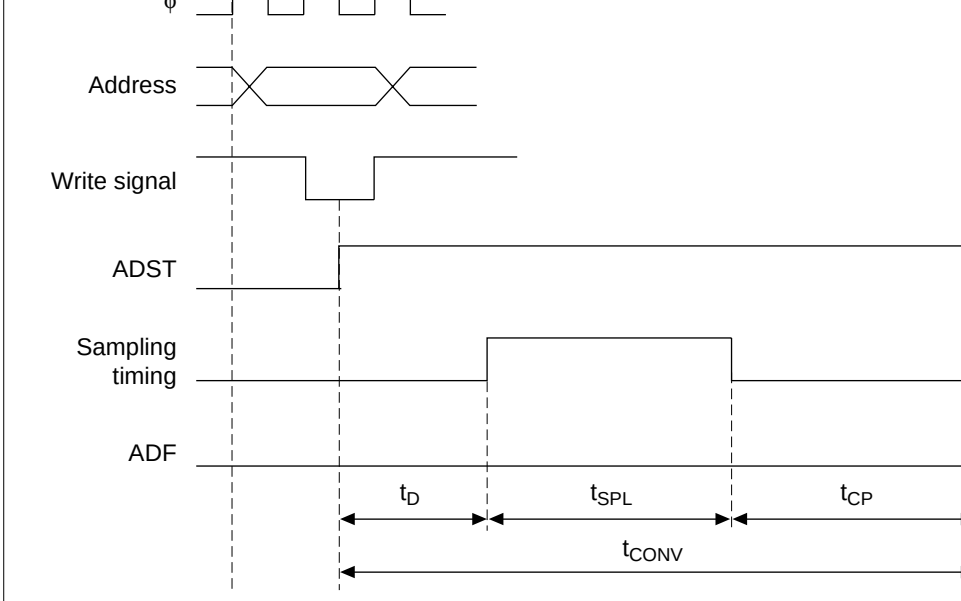


Figure 15.12 Conversion Start by $\overline{\text{ADTRG}}$ Conversion Start Trigger



t_D: A/D conversion start delay time
t_{SPL}: Input sampling time
t_{CONV}: A/D conversion time
t_{CP}: Operation time

Figure 15.13 A/D Conversion Timing

continuously executing conversion, top for the second time and following is 2
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 great
15.8 shows the operating frequency and CKS bit settings.

Table 15.8 Operating Frequency and CKS Bit Settings

CKS	Conversion Time (States)	Minimum Conversion Time (µs)				
		28 MHz	20 MHz	16 MHz	10 MHz	8
0	42.5	—	2.1	2.6	4.3	5
1	82.5	2.9	4.2	5.0	8.3	1

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 cor
of A/D conversions. The ADI interrupt request can be enabled or disabled by the ADIE b
ADCSR.

The DTC or DMAC can be activated by ADI interrupts. When converted data is read by t
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 fla
cleared to 0.

1. Analog input voltage range

   During A/D conversions, see that the voltage applied to the analog input pins AN0–A 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 the A/D converter, use AVcc = Vcc, AVss = Vss. During the standby mode, use V_{RA} ≤ 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 t converter, use AV_{ref} = Vcc. During the standby mode, use V_{RAM} ≤ AVref ≤ AVcc. V_{R} RAM standby voltage.

4. Input ports

   The time constant for the circuit connecting to the input port must be shorter than the time of the A/D converter. Input voltage may not be sampled sufficiently when the ti 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 speed start mode and low-demand conversion mode.

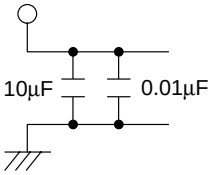
6. Analog input pins handling

   Connect a protection circuit as shown in figure 15.14 to prevent analog input pins (A from being destroyed due to abnormal voltage from surge, etc. This circuit is also eq with a CR filter to control errors due to noise. The circuit shown in the diagram is on example and the number of circuits is to be determined by considering the actual con use.

   Figure 15.15 shows an equivalent circuit of analog input pins and table 15.10 shows specification of the analog input pins.

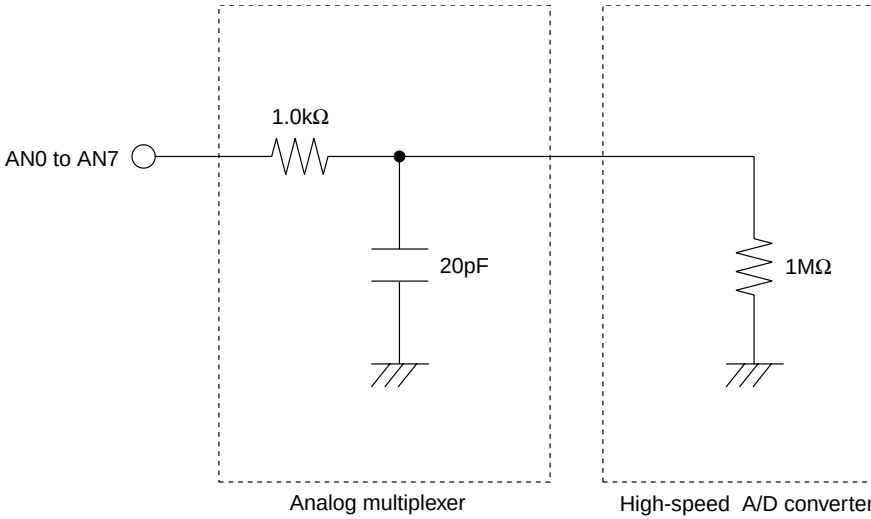
Notes: Numbers are only to be noted as reference value

*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

- 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), con 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 inte request) can activate DMAC.

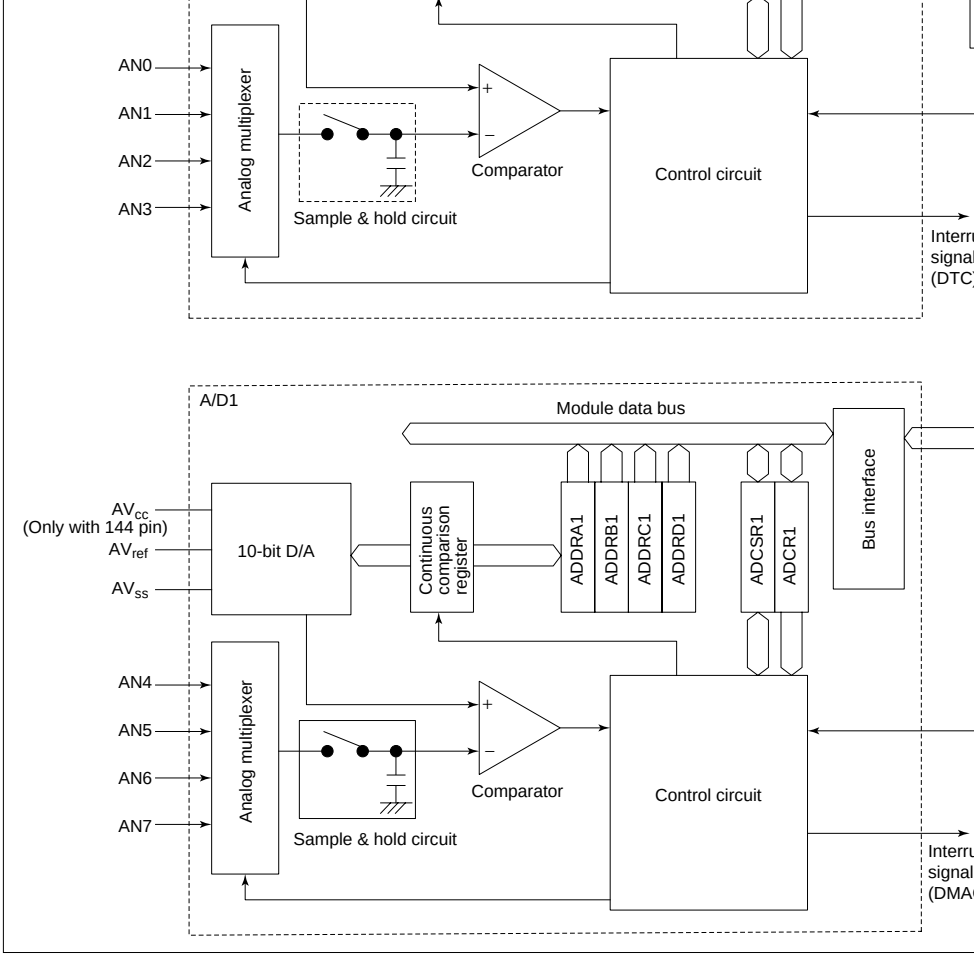


Figure 16.1 Mid-Speed A/D Converter Block Diagram

Standard voltage		AVref*	I	A/D conversion standard voltage (SH7041A, SH7043A, and SH704
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 conversior

Note: * In the SH7040A, SH7042A, and SH7044, AV_{ref} is connected to AV_{CC} internally.

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.

ADDR can always be read from the CPU. The upper byte may be read directly. The low
transferred through the temporary register (TEMP). For details, see section 16.3, Interfa
CPU.

ADDR is initialized to H'0000 during power-on reset or standby mode. ADDR will not
initialized by manual reset.

Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2
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
R/W :	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	

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] (Ini 1. Writing 0 to ADF after reading ADF with ADF=1 2. When registers of the mid-speed converter are accessed after the DMAC a 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
completion of A/D conversion.

Bit 6:

ADIE	Description
0	Disables interrupt request (ADI) due to completion of A/D conversion (Ini
1	Enables interrupt request (ADI) due to completion of A/D conversion

- Bit 4—Scan Mode (SCAN): Selects the A/D conversion mode from single mode and
  mode. For operations during single/scan mode, see section 16.4, Operation. When sw
  modes, proceed while ADST=0.

Bit 4:

SCAN	Description
0	Single mode (I
1	Scan mode

- Bit 3—Clock Select (CKS): Sets the A/D conversion time. Proceed conversion time
  while adst=0. Always set CKS=0 when operating frequency exceeds 20MHz.

Bit 3:

CKS	Description
0	Conversion time = 266 states (max) (I
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 w
  SCAN bit. Switch channels while ADST=0.

A/D control registers (ADCR0, 1) are registers that can read/write in 8 bits and enables or
A/D conversion start of the external trigger input. There are the ADCR0 (A/D0) and ADC
(A/D1).

ADCR is initialized to H'7F during power-on reset and standby mode. Manual reset does
initialize ADCR.

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

- Bit 7—Trigger Enable (TRGE): Enables or disables A/D conversion start of input fro
external or MTU trigger.

Bit 7:

TRGE	Description
0	Disables A/D conversion start of external or MTU trigger (Ini
1	Starts A/D conversion on last transition edge of A/D conversion trigger input pi (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

When reading the ADDR in byte size, read the upper byte before the lower byte. Further possible to read only the upper byte, however, please note that contents are not guarante reading only the lower byte. In addition, when reading ADDR in word size, upper byte i automatically read before the lower byte.

Figure 16.2 shows the data flow when reading from ADDR.

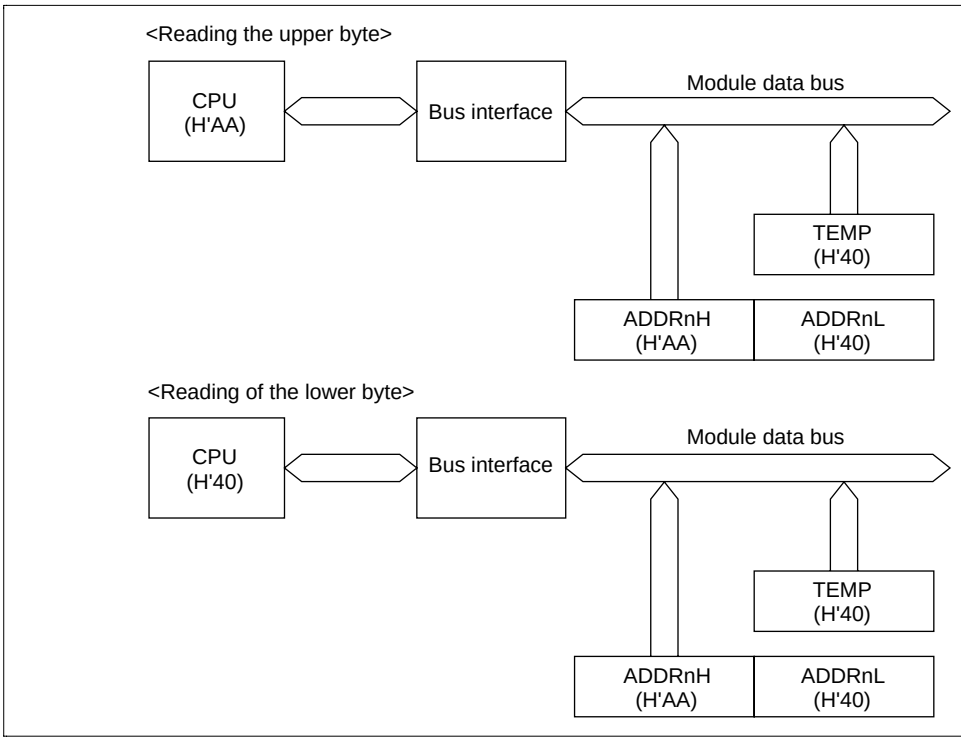


Figure 16.2 ADDR Access Operation (During Reading of (H'AA40))

When conversion is complete, the ADF bit of ADCSR is set to 1. At this time, if the ADI 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 stop A/D conversion to avoid malfunction. After switching (mode/channel change and AI 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 i 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= 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. 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 conver starts, following steps (2) to (7) above.

Figure 16.3 Operation Example of Mid-speed A/D Converter (Single Mode, Ch Selected)

An example of operation when three channels of A/D0 (AN0–2) are selected for A/D co
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
   then start A/D conversion (ADST=1).
2. When A/D conversion for channel 1 is complete, A/D conversion result is transferred
   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
   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 complet
   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
   A/D conversion will start again from channel 1 (AN0).

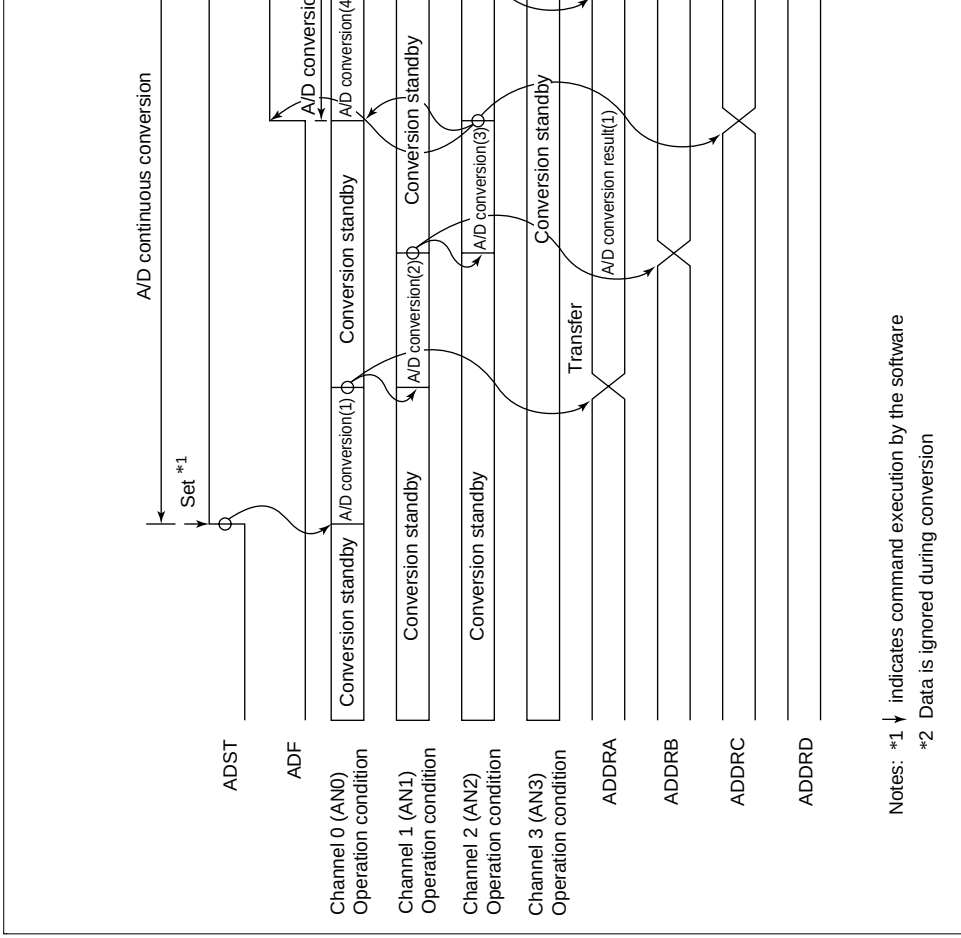
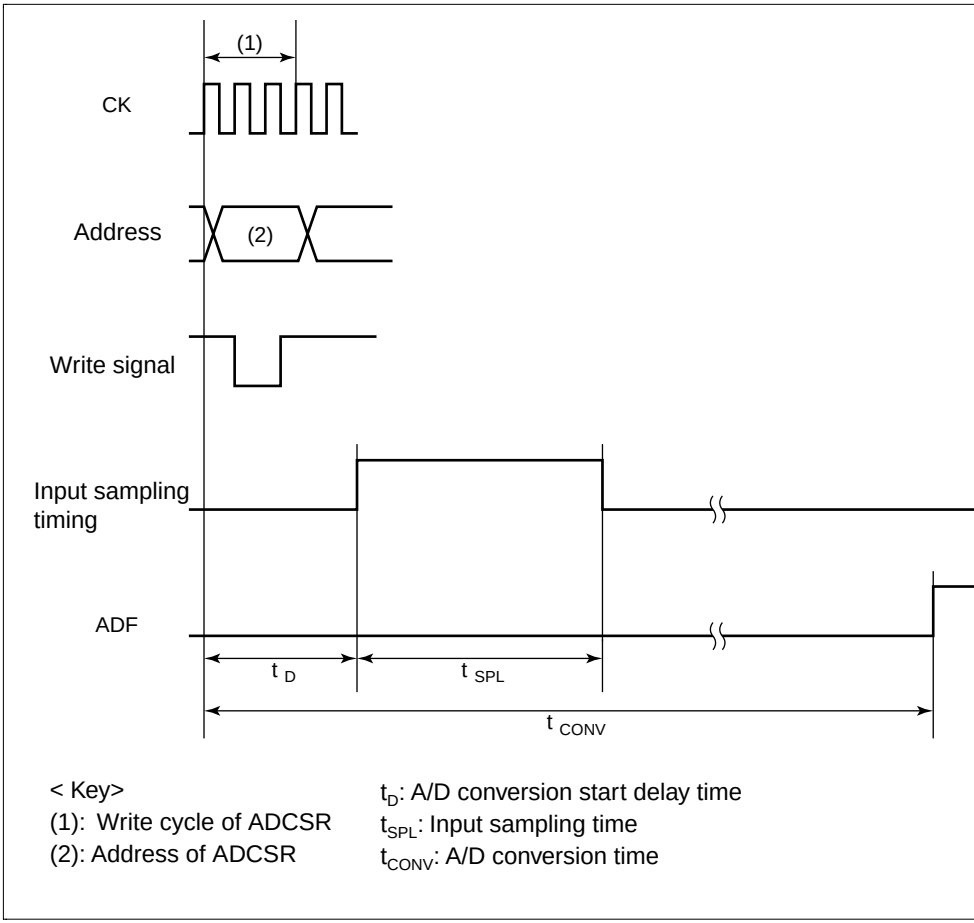


Figure 16.4 Operation Example of Mid-speed A/D Converter (Scan Mode, Three Selected) (AN0–AN2)

CKS=1.



< Key>
(1): Write cycle of ADCSR
(2): Address of ADCSR

t_D: A/D conversion start delay time
t_{SPL}: Input sampling time
t_{CONV}: A/D conversion time

Figure 16.5 A/D Conversion Timing

It is possible to start A/D conversion from an external trigger input. External trigger inpu
from the $\overline{\text{ADTRG}}$ pin or MTU when the TRGE bit of the A/D control register (ADCR)

A/D conversion is started when the ADST bit of the A/D control/status register (ADCSR
1 by the $\overline{\text{ADTRG}}$ input pin last transition edge or MTU trigger. Other operations, regard
whether in the single or scan mode, are the same as when setting the ADST bit to 1 with
software.

Figure 16.6 shows an example of external trigger input timing.

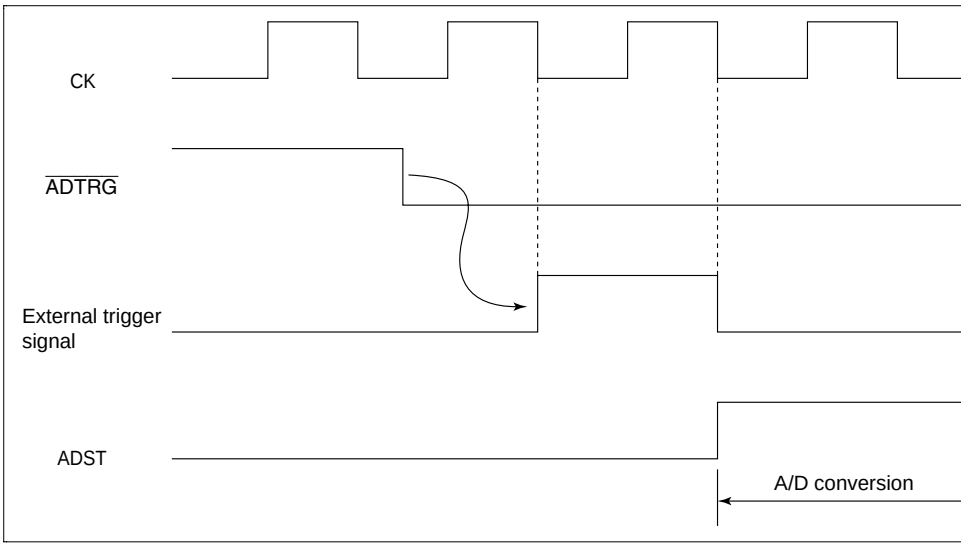


Figure 16.6 External Trigger Input Timing

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 control/status register (ADCSR0) will automatically be cleared to 0. Furthermore, it is po automatically clear the ADF bit of ADCSR1 by register access of A/D1 with activated DI ADI1 interrupt. For details on the automatic clearing operation of this interrupt factor, see 8, Data Transfer Controller (DTC).

clarity, this figure shows 3-bit medium-speed A/D conversion rather than 10-bit medium
A/D conversion. Offset error (see figure 16.7 (1)) is the deviation between the actual A/
conversion characteristic and the ideal A/D conversion characteristic when the digital ou
changes from the minimum value (zero voltage) of 0000000000 (000 in the figure) to 00
(001 in the figure). Full-scale error (see figure 16.7 (2)) is the deviation between the ac
conversion characteristic and the ideal A/D conversion characteristic when the digital ou
changes from 1111111110 (110 in the figure) to the maximum value (full-scale voltage)
111111111 (111 in the figure). Quantization error is the deviation inherent in the mediu
A/D converter, given by 1/2 LSB (see figure 16.7 (3)). Nonlinearity error is the deviatio
the actual A/D conversion characteristic and the ideal A/D conversion characteristic fro
voltage to full-scale voltage (see figure 16.7 (4)). This does not include offset error, full
error, and quantization error.

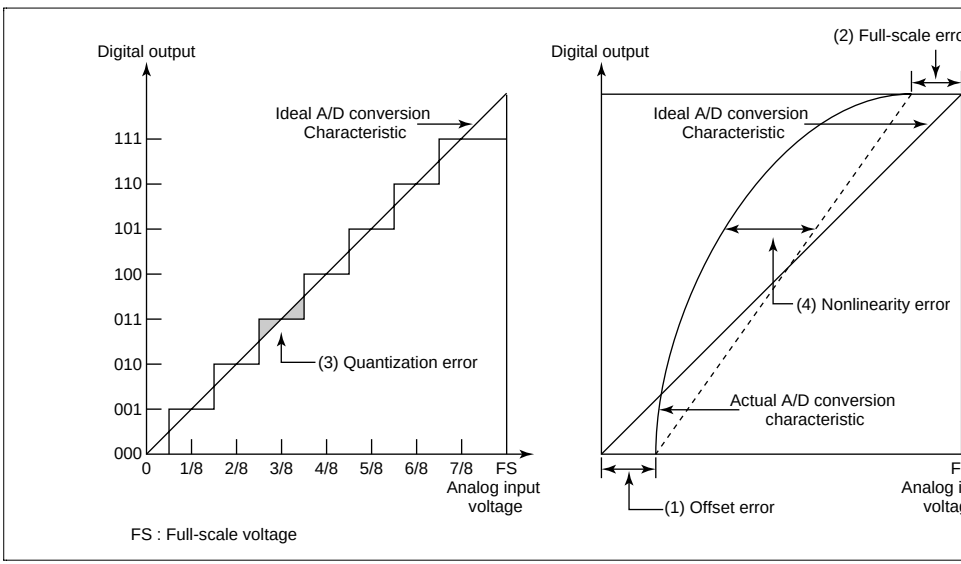


Figure 16.7 A/D Conversion Precision Definitions

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 mediu
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-sp
converter is not used or is in standby mode.

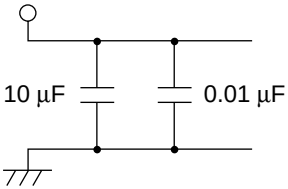
16.7.2 Handling of Analog Input Pins

To prevent damage from surges and other abnormal voltages at the analog input pins (AN
connect a protection circuit such as that shown in figure 16.8. This circuit also includes a
function that suppresses error due to noise. The circuit shown here is only a design exam
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 summari
analog input pin specifications.

Notes: Numbers are only to be noted as reference value

*1



*2 R_{in}: Input impedance

Figure 16.8 Example of Analog Input Pin Protection Circuit

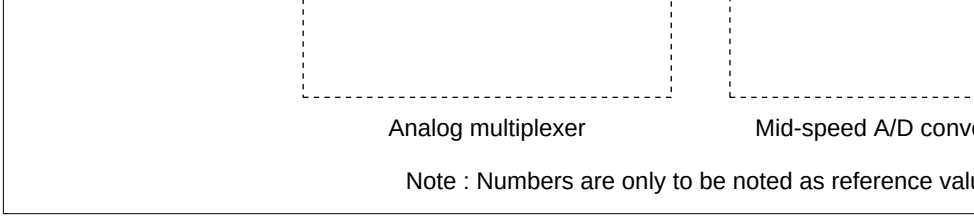


Figure 16.9 Equivalent Circuit for the Analog Input Pins

Table 16.6 Analog Pin Specifications

Item	Min	Max	Unit
Analog input capacitance	—	20	pF
Permissible signal source impedance	—	1	kΩ

— One of four internal clocks (ϕ/8, ϕ/32, ϕ/128, ϕ/512) can be selected independent 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.

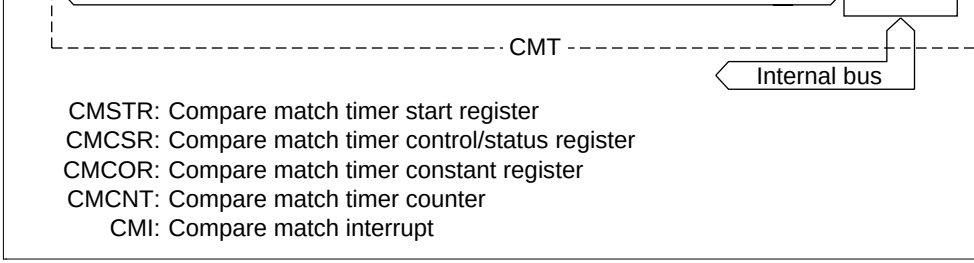


Figure 17.1 CMT Block Diagram

	counter 0					
	Compare match timer constant register 0	CMCOR0	R/W	H'FFFF	H'FFFF83D6	8, 1
1	Compare match timer control/status register 1	CMCSR1	R/(W)*	H'0000	H'FFFF83D8	8, 1
	Compare match timer counter 1	CMCNT1	R/W	H'0000	H'FFFF83DA	8, 1
	Compare match timer constant register 1	CMCOR1	R/W	H'FFFF	H'FFFF83DC	8, 1

Note: * The only value that can be written to the CMCSR0 and CMCSR1 CMF bits is a
the flags.

Bit:	7	6	5	4	3	2	1
	—	—	—	—	—	—	STR1
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	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 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 0.

Bit 0: STR0	Description
0	CMCNT0 count operation halted (initial value)
1	CMCNT0 count operation

Bit:	7	6	5	4	3	2	1
	CMF	CMIE	—	—	—	—	CKS1
Initial value:	0	0	0	0	0	0	0
R/W:	R/(W)*	R/W	R	R	R	R	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 a 0.

- Bit 7—Compare Match Flag (CMF): This flag indicates whether or not the CMCNT 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 disab compare match interrupt (CMI) when the CMCNT and CMCOR values have matche 1).

Bit 6: CMIE	Description
0	Compare match interrupts (CMI) disabled (initial status)
1	Compare match interrupts (CMI) enabled

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 bit of the CMSTR is set to 1, the CMCNT begins incrementing with that clock. When the value matches that of the compare match timer constant register (CMCOR), the CMCNT cleared to H'0000 and the CMF flag of the CMCSR is set to 1. If the CMIE bit of the CM 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 re not initialize CMCNT.

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

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

Bit:	7	6	5	4	3	2	1
Initial value:	1	1	1	1	1	1	1
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 an bit of the CMSTR is set to 1, the CMCNT begins incrementing with the selected clock. CMCNT counter value matches that of the compare match constant register (CMCOR), CMCNT counter is cleared to H'0000 and the CMF flag of the CMCSR register is set to CMIE bit of the CMCSR register is set to 1 at this time, a compare match interrupt (CM requested. The CMCNT counter begins counting up again from H'0000.

Figure 17.2 shows the compare match counter operation.

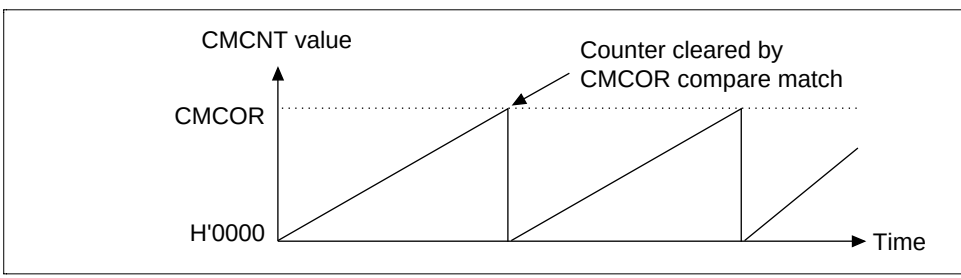


Figure 17.2 Counter Operation

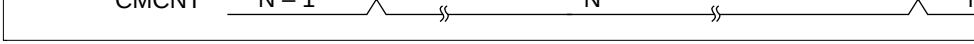


Figure 17.3 Count Timing

17.4 Interrupts

17.4.1 Interrupt Sources and DTC Activation

The CMT has a compare match interrupt for each channel, with independent vector addre
allocated to each of them. The corresponding interrupt request is output when the interrup
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 ca
changed by using the interrupt controller settings. See section 6, Interrupt Controller (INT
details.

Interrupt requests can also be used as data transfer controller (DTC) activating sources. In
case, channel priorities are fixed. See section 8, Data Transfer Controller (DTC), for detai

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 wh
CMCOR register and the CMCNT counter match. The compare match signal is generated
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 com
match signal will not be generated until a CMCNT counter input clock occurs. Figure 17.
the CMF bit set timing.

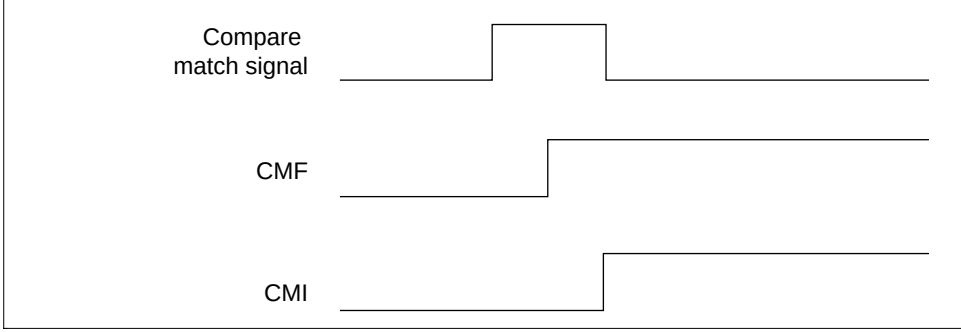


Figure 17.4 CMF Set Timing

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
clear signal after a DTC transfer. Figure 17.5 shows the timing when the CMF bit is clea
CPU.

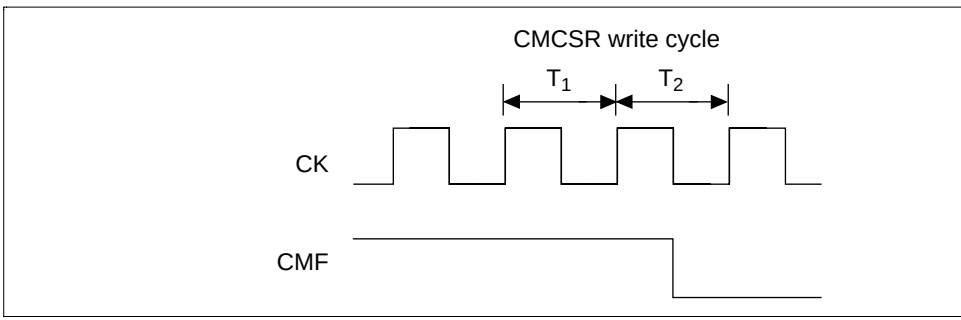


Figure 17.5 Timing of CMF Clear by the CPU

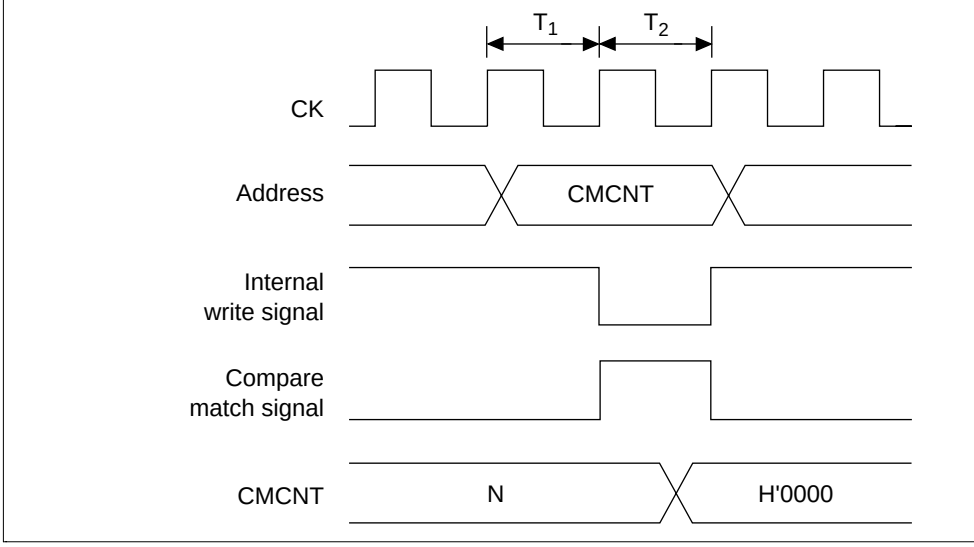


Figure 17.6 CMCNT Write and Compare Match Contention

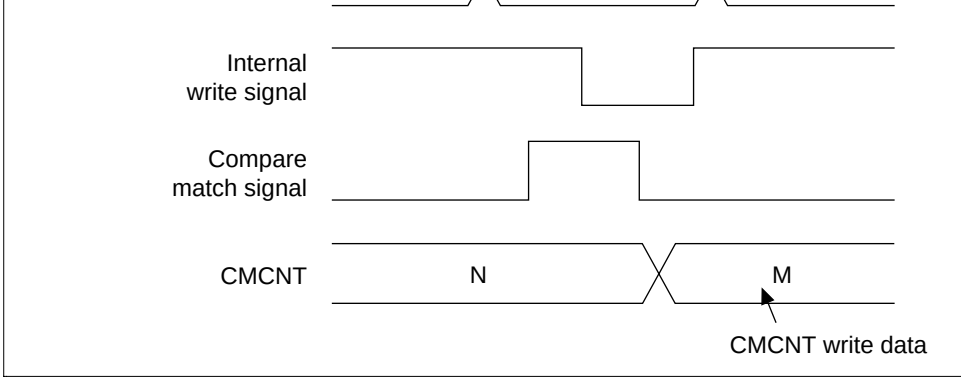


Figure 17.7 CMCNT Word Write and Increment Contention

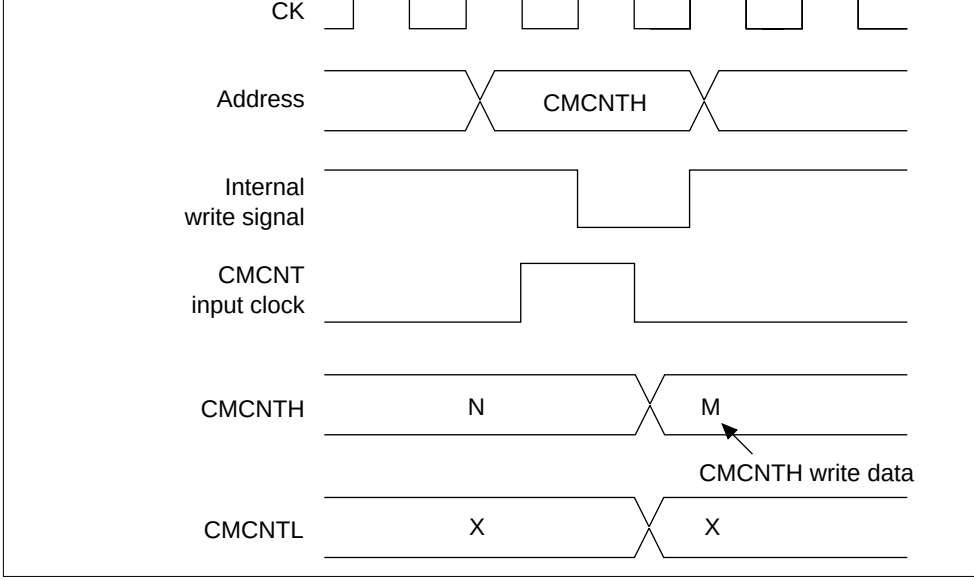


Figure 17.8 CMCNT Byte Write and Increment Contention

Port	(Related Module)	(Related Module)	(Related Module)	(Related Module)	112
A	PA23 I/O (port)	$\overline{\text{WRHH}}$ output (BSC)	—	—	—
	PA22 I/O (port)	$\overline{\text{WRHL}}$ output (BSC)	—	—	—
	PA21 I/O (port)	$\overline{\text{CASHH}}$ output (BSC)	—	—	—
	PA20 I/O (port)	$\overline{\text{CASHL}}$ output (BSC)	—	—	—
	PA19 I/O (port)	$\overline{\text{BACK}}$ output (BSC)	DRAK1 output (DMAC)	—	—
	PA18 I/O (port)	$\overline{\text{BREQ}}$ input (BSC)	DRAK0 output (DMAC)	—	—
	PA17 /O (port)	$\overline{\text{WAIT}}$ input (BSC)	—	—	—
	PA16 I/O (port)	$\overline{\text{AH}}$ output (BSC)	—	—	—
	PA15 I/O (port)	CK output (CPG)	—	—	83
	PA14 I/O (port)	$\overline{\text{RD}}$ output (BSC)	—	—	34
	PA13 I/O (port)	$\overline{\text{WRH}}$ output (BSC)	—	—	36
	PA12 I/O (port)	$\overline{\text{WRL}}$ output (BSC)	—	—	38
	PA11 I/O (port)	$\overline{\text{CS1}}$ output (BSC)	—	—	40
	PA10 I/O (port)	$\overline{\text{CS0}}$ output (BSC)	—	—	41
	PA9 I/O (port)	TCLKD input (MTU)	$\overline{\text{IRQ3}}$ (INTC)	—	42
	PA8 I/O (port)	TCLKC input (MTU)	$\overline{\text{IRQ2}}$ (INTC)	—	43
	PA7 I/O (port)	TCLKB input (MTU)	$\overline{\text{CS3}}$ output (BSC)	—	44
	PA6 I/O (port)	TCLKA input (MTU)	$\overline{\text{CS2}}$ output (BSC)	—	45
	PA5 I/O (port)	SCK1 I/O (SCI)	$\overline{\text{DREQ1}}$ input (DMAC)	$\overline{\text{IRQ1}}$ input (INTC)	46

	PB7 I/O (port)	$\overline{\text{IRQ5}}$ input (INTC)	A19 output (BSC)	$\overline{\text{BREQ}}$ input (BSC)	30
	PB6 I/O (port)	$\overline{\text{IRQ4}}$ input (INTC)	A18 output (BSC)	$\overline{\text{BACK}}$ output (BSC)	29
	PB5 I/O (port)	$\overline{\text{IRQ3}}$ input (INTC)	$\overline{\text{POE3}}$ input (port)	RDWR output (BSC)	28
	PB4 I/O (port)	$\overline{\text{IRQ2}}$ input (INTC)	$\overline{\text{POE2}}$ input (port)	$\overline{\text{CASH}}$ output (BSC)	26
	PB3 I/O (port)	$\overline{\text{IRQ1}}$ input (INTC)	$\overline{\text{POE1}}$ input (port)	$\overline{\text{CASL}}$ output (BSC)	25
	PB2 I/O (port)	$\overline{\text{IRQ0}}$ input (INTC)	$\overline{\text{POE0}}$ input (port)	$\overline{\text{RAS}}$ output (BSC)	24
	PB1 I/O (port)	A17 input (BSC)	—	—	22
	PB0 I/O (port)	A16 output (BSC)	—	—	20
C	PC15 I/O (port)	A15 output (BSC)	—	—	19
	PC14 I/O (port)	A14 output (BSC)	—	—	18
	PC13 I/O (port)	A13 output (BSC)	—	—	17
	PC12 I/O (port)	A12 output (BSC)	—	—	16
	PC11 I/O (port)	A11 output (BSC)	—	—	15
	PC10 I/O (port)	A10 output (BSC)	—	—	14
	PC9 I/O (port)	A9 output (BSC)	—	—	13
	PC8 I/O (port)	A8 output (BSC)	—	—	12
	PC7 I/O (port)	A7 output (BSC)	—	—	11

D	PD31 I/O (port)	D31 I/O (BSC)	ADTRG input (A/D)	—	—
	PD30 I/O (port)	D30 I/O (BSC)	$\overline{\text{IRQOUT}}$ output (INTC)	—	—
	PD29 I/O (port)	D29 I/O (BSC)	$\overline{\text{CS3}}$ output (BSC)	—	—
	PD28 I/O (port)	D28 I/O (BSC)	$\overline{\text{CS2}}$ output (BSC)	—	—
	PD27 I/O (port)	D27 I/O (BSC)	DACK1 output (DMAC)	—	—
	PD26 I/O (port)	D26 I/O (BSC)	DACK0 output (DMAC)	—	—
	PD25 I/O (port)	D25 I/O (BSC)	$\overline{\text{DREQ1}}$ input (DMAC)	—	—
	PD24 I/O (port)	D24 I/O (BSC)	$\overline{\text{DREQ0}}$ input (DMAC)	—	—
	PD23 I/O (port)	D23 I/O (BSC)	$\overline{\text{IRQ7}}$ input (INTC)	—	—
	PD22 I/O (port)	D22 I/O (BSC)	$\overline{\text{IRQ6}}$ input (INTC)	—	—
	PD21 I/O (port)	D21 I/O (BSC)	$\overline{\text{IRQ5}}$ input (INTC)	—	—
	PD20 I/O (port)	D20 I/O (BSC)	$\overline{\text{IRQ4}}$ input (INTC)	—	—
	PD19 I/O (port)	D19 I/O (BSC)	$\overline{\text{IRQ3}}$ input (INTC)	—	—
	PD18 I/O (port)	D18 I/O (BSC)	$\overline{\text{IRQ2}}$ input (INTC)	—	—
	PD17 I/O (port)	D17 I/O (BSC)	$\overline{\text{IRQ1}}$ input (INTC)	—	—
	PD16 I/O (port)	D16 I/O (BSC)	$\overline{\text{IRQ0}}$ input (INTC)	—	—
	PD15 I/O (port)	D15 I/O (BSC)	—	—	52
	PD14 I/O (port)	D14 I/O (BSC)	—	—	53

	PD6 I/O (port)	D6 I/O (BSC)	—	—	63
	PD5 I/O (port)	D5 I/O (BSC)	—	—	64
	PD4 I/O (port)	D4 I/O (BSC)	—	—	66
	PD3 I/O (port)	D3 I/O (BSC)	—	—	67
	PD2 I/O (port)	D2 I/O (BSC)	—	—	68
	PD1 I/O (port)	D1 I/O (BSC)	—	—	69
	PD0 I/O (port)	D0 I/O (BSC)	—	—	70
E	PE15 I/O (port)	TIOC4D I/O (MTU)	DACK1 output (DMAC)	$\overline{\text{IRQOUT}}$ output (INTC)	2
	PE14 I/O (port)	TIOC4C I/O (MTU)	DACK0 output (DMAC)	$\overline{\text{AH}}$ output (BSC)	1
	PE13 I/O (port)	TIOC4B I/O (MTU)	$\overline{\text{MRES}}$ input (INTC)	—	112
	PE12 I/O (port)	TIOC4A I/O (MTU)	—	—	111
	PE11 I/O (port)	TIOC3D I/O (MTU)	—	—	110
	PE10 I/O (port)	TIOC3C I/O (MTU)	—	—	108
	PE9 I/O (port)	TIOC3B I/O (MTU)	—	—	107
	PE8 I/O (port)	TIOC3A I/O (MTU)	—	—	106
	PE7 I/O (port)	TIOC2B I/O (MTU)	—	—	105
	PE6 I/O (port)	TIOC2A I/O (MTU)	—	—	104
	PE5 I/O (port)	TIOC1B I/O (MTU)	—	—	102
	PE4 I/O (port)	TIOC1A I/O (MTU)	—	—	89

PF6 input (port)	AN6 input (A/D)	—	—	98
PF5 input (port)	AN5 input (A/D)	—	—	96
PF4 input (port)	AN4 input (A/D)	—	—	95
PF3 input (port)	AN3 input (A/D)	—	—	94
PF2 input (port)	AN2 input (A/D)	—	—	93
PF1 input (port)	AN1 input (A/D)	—	—	92
PF0 input (port)	AN0 input (A/D)	—	—	91

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	
TFP120	FP144	FP112	Initial Function	PFC Selected Function Possiblilities	Initial Function	PFC Selected Function Possiblilities	Initial Function	PFC Selected Fun
22, 40, 70, 82, 111	12,26,40,63 77,85,112 135	21,37,65 103	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
75	92	70	D0	D0	D0	D0	PD0	PD0/D0
74	91	69	D1	D1	D1	D1	PD1	PD1/D1
73	90	68	D2	D2	D2	D2	PD2	PD2/D2
72	89	67	D3	D3	D3	D3	PD3	PD3/D3
71	88	66	D4	D4	D4	D4	PD4	PD4/D4
69	86	64	D5	D5	D5	D5	PD5	PD5/D5
68	84	63	D6	D6	D6	D6	PD6	PD6/D6
67	83	62	D7	D7	D7	D7	PD7	PD7/D7
65	82	60	D8	D8	D8	D8	PD8	PD8/D8
64	81	59	D9	D9	D9	D9	PD9	PD9/D9
63	80	58	D10	D10	D10	D10	PD10	PD10/D10
62	78	57	D11	D11	D11	D11	PD11	PD11/D11
59	76	56	D12	D12	D12	D12	PD12	PD12/D12
57	75	54	D13	D13	D13	D13	PD13	PD13/D13
56	74	53	D14	D14	D14	D14	PD14	PD14/D14
55	73	52	D15	D15	D15	D15	PD15	PD15/D15
—	72	—	PD16	PD16/D16/$\overline{\text{IRQ0}}$	D16	D16	PD16	PD16/D16/$\overline{\text{IRQ}}$
—	70	—	PD17	PD17/D17/$\overline{\text{IRQ1}}$	D17	D17	PD17	PD17/D17/$\overline{\text{IRQ}}$
—	69	—	PD18	PD18/D18/$\overline{\text{IRQ2}}$	D18	D18	PD18	PD18/D18/$\overline{\text{IRQ}}$
—	68	—	PD19	PD19/D19/$\overline{\text{IRQ3}}$	D19	D19	PD19	PD19/D19/$\overline{\text{IRQ}}$
—	67	—	PD20	PD20/D20/$\overline{\text{IRQ4}}$	D20	D20	PD20	PD20/D20/$\overline{\text{IRQ}}$
—	66	—	PD21	PD21/D21/$\overline{\text{IRQ5}}$	D21	D21	PD21	PD21/D21/$\overline{\text{IRQ}}$
—	65	—	PD22	PD22/D22/$\overline{\text{IRQ6}}$	D22	D22	PD22	PD22/D22/$\overline{\text{IRQ}}$
—	64	—	PD23	PD23/D23/$\overline{\text{IRQ7}}$	D23	D23	PD23	PD23/D23/$\overline{\text{IRQ}}$
—	62	—	PD24	PD24/D24/$\overline{\text{DREQ0}}$	D24	D24	PD24	PD24/D24/$\overline{\text{DRE}}$
—	60	—	PD25	PD25/D25/$\overline{\text{DREQ1}}$	D25	D25	PD25	PD25/D25/$\overline{\text{DRE}}$
—	59	—	PD26	PD26/D26/DACK0	D26	D26	PD26	PD26/D26/DA0
—	58	—	PD27	PD27/D27/DACK1	D27	D27	PD27	PD27/D27/DA0
—	57	—	PD28	PD28/D28/$\overline{\text{CS2}}$	D28	D28	PD28	PD28/D28/$\overline{\text{CS2}}$
—	56	—	PD29	PD29/D29/$\overline{\text{CS3}}$	D29	D29	PD29	PD29/D29/$\overline{\text{CS3}}$
—	46	—	PD30	PD30/D30/$\overline{\text{IRQOUT}}$	D30	D30	PD30	PD30/D30/$\overline{\text{IRQ}}$
—	45	—	PD31	PD31/D31/$\overline{\text{ADTRG}}$	D31	D31	PD31	PD31/D31/$\overline{\text{ADT}}$
5	7	4	A0	A0	A0	A0	PC0	PC0/A0
6	8	5	A1	A1	A1	A1	PC1	PC1/A1
7	9	6	A2	A2	A2	A2	PC2	PC2/A2
8	10	7	A3	A3	A3	A3	PC3	PC3/A3
9	11	8	A4	A4	A4	A4	PC4	PC4/A4
10	13	9	A5	A5	A5	A5	PC5	PC5/A5
11	15	10	A6	A6	A6	A6	PC6	PC6/A6
12	16	11	A7	A7	A7	A7	PC7	PC7/A7

Table 18.2 Pin Arrangement by Mo

Pin NO.			Pin Name					
			On-Chip ROM Disabled				On-Chip ROM Enabled	
			MPU Mode0		MPU Mode 1		MPU Mode2	
TFP120	FP144	FP112	Initial Function	PFC Selected Function Possiblilities	Initial Function	PFC Selected Function Possiblilities	Initial Function	PFC Selected Fu
18	22	17	A13	A13	A13	A13	PC13	PC13/A13
19	23	18	A14	A14	A14	A14	PC14	PC14/A14
20	24	19	A15	A15	A15	A15	PC15	PC15/A15
21	25	20	A16	A16	A16	A16	PB0	PB0/A16
23	27	22	A17	A17	A17	A17	PB1	PB1/A17
25	31	24	PB2	PB2/IRQ0/POE0/RAS	PB2	PB2/IRQ0/POE0/RAS	PB2	PB2/IRQ0/PO
26	32	25	PB3	PB3/IRQ1/POE1/CASL	PB3	PB3/IRQ1/POE1/CASL	PB3	PB3/IRQ1/PO
27	34	26	PB4	PB4/IRQ2/POE2/CASH	PB4	PB4/IRQ2/POE2/CASH	PB4	PB4/IRQ2/PO
29	36	28	PB5	PB5/IRQ3/POE3/RDWR	PB5	PB5/IRQ3/POE3/RDWR	PB5	PB5/IRQ3/PO
32	37	29	PB6	PB6/IRQ4/A18/BACK	PB6	PB6/IRQ4/A18/BACK	PB6	PB6/IRQ4/A1
33	38	30	PB7	PB7/IRQ5/A19/BREQ	PB7	PB7/IRQ5/A19/BREQ	PB7	PB7/IRQ5/A1
34	39	31	PB8	PB8/IRQ6/A20/WAIT	PB8	PB8/IRQ6/A20/WAIT	PB8	PB8/IRQ6/A2
35	41	32	PB9	PB9/IRQ7/A21/ADTRG	PB9	PB9/IRQ7/A21/ADTRG	PB9	PB9/IRQ7/A2
54	130	51	PA0	PA0/RXD0	PA0	PA0/RXD0	PA0	PA0/RXD0
53	131	50	PA1	PA1/TXD0	PA1	PA1/TXD0	PA1	PA1/TXD0
52	132	49	PA2	PA2/SCK0/DREQ0/IRQ0	PA2	PA2/SCK0/DREQ0/IRQ0	PA2	PA2/SCK0/DR
51	133	48	PA3	PA3/RXD1	PA3	PA3/RXD1	PA3	PA3/RXD1
50	134	47	PA4	PA4/TXD1	PA4	PA4/TXD1	PA4	PA4/TXD1
49	136	46	PA5	PA5/SCK1/DREQ1/IRQ1	PA5	PA5/SCK1/DREQ1/IRQ1	PA5	PA5/SCK1/DR
48	54	45	PA6	PA6/TCLKA/CS2	PA6	PA6/TCLKA/CS2	PA6	PA6/TCLKA/C
47	53	44	PA7	PA7/TCLKB/CS3	PA7	PA7/TCLKB/CS3	PA7	PA7/TCLKB/C
46	52	43	PA8	PA8/TCLKC/IRQ2	PA8	PA8/TCLKC/IRQ2	PA8	PA8/TCLKC/I
45	51	42	PA9	PA9/TCLKD/IRQ3	PA9	PA9/TCLKD/IRQ3	PA9	PA9/TCLKD/I
44	50	41	CS0	CS0	CS0	CS0	PA10	PA10/CS0
43	49	40	CS1	CS1	CS1	CS1	PA11	PA11/CS1
41	48	38	WRL	WRL	WRL	WRL	PA12	PA12/WRL
39	47	36	WRH	WRH	WRH	WRH	PA13	PA13/WRH
37	43	34	RD	RD	RD	RD	PA14	PA14/RD
88	107	83	CK	PA15/CK	CK	PA15/CK	CK	PA15/CK
—	100	—	PA16	PA16/AH	PA16	PA16/AH	PA16	PA16/AH
—	101	—	PA17	PA17/WAIT	PA17	PA17/WAIT	PA17	PA17/WAIT
—	33	—	PA18	PA18/BREQ/DRAK0	PA18	PA18/BREQ/DRAK0	PA18	PA18/BREQ/I
—	30	—	PA19	PA19/BACK/DRAK1	PA19	PA19/BACK/DRAK1	PA19	PA19/BACK/I
—	29	—	PA20	PA20/CASHL	PA20	PA20/CASHL	PA20	PA20/CASHL
—	4	—	PA21	PA21/CASHH	PA21	PA21/CASHH	PA21	PA21/CASHH
—	3	—	WRHL	WRHL	WRHL	WRHL	PA22	PA22/WRHL
—	1	—	WRHH	WRHH	WRHH	WRHH	PA23	PA23/WRHH
85	104	80	PLLVCC	PLLVCC	PLLVCC	PLLVCC	PLLVCC	PLLVCC
87	106	82	PLLVSS	PLLVSS	PLLVSS	PLLVSS	PLLVSS	PLLVSS
79	96	74	EXTAL	EXTAL	EXTAL	EXTAL	EXTAL	EXTAL
77	94	72	XTAL	XTAL	XTAL	XTAL	XTAL	XTAL
86	105	81	PLLCAP	PLLCAP	PLLCAP	PLLCAP	PLLCAP	PLLCAP
81	98	76	NMI	NMI	NMI	NMI	NMI	NMI
89	108	84	RES	RES	RES	RES	RES	RES
88	44	35	WDTOVF	WDTOVF	WDTOVF	WDTOVF	WDTOVF	WDTOVF

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	
TFP120	FP144	FP112	Initial Function	PFC Selected Function	Possibilities	Initial Function	PFC Selected Function	Possibilities	Initial Function	PFC Selected Fun
—	127	—	AVREF	AVREF		AVREF	AVREF		AVREF	AVREF
98	118	91	PF0/AN0	PF0/AN0		PF0/AN0	PF0/AN0		PF0/AN0	PF0/AN0
99	119	92	PF1/AN1	PF1/AN1		PF1/AN1	PF1/AN1		PF1/AN1	PF1/AN1
100	120	93	PF2/AN2	PF2/AN2		PF2/AN2	PF2/AN2		PF2/AN2	PF2/AN2
101	121	94	PF3/AN3	PF3/AN3		PF3/AN3	PF3/AN3		PF3/AN3	PF3/AN3
102	122	95	PF4/AN4	PF4/AN4		PF4/AN4	PF4/AN4		PF4/AN4	PF4/AN4
103	123	96	PF5/AN5	PF5/AN5		PF5/AN5	PF5/AN5		PF5/AN5	PF5/AN5
105	125	98	PF6/AN6	PF6/AN6		PF6/AN6	PF6/AN6		PF6/AN6	PF6/AN6
106	126	99	PF7/AN7	PF7/AN7		PF7/AN7	PF7/AN7		PF7/AN7	PF7/AN7
92	109	85	PE0	PE0/TIOC0A/DREQ0		PE0	PE0/TIOC0A/DREQ0		PE0	PE0/TIOC0A/D
93	110	86	PE1	PE1/TIOC0B/DRAK0		PE1	PE1/TIOC0B/DRAK0		PE1	PE1/TIOC0B/D
94	111	87	PE2	PE2/TIOC0C/DREQ1		PE2	PE2/TIOC0C/DREQ1		PE2	PE2/TIOC0C/D
95	113	88	PE3	PE3/TIOC0D/DRAK1		PE3	PE3/TIOC0D/DRAK1		PE3	PE3/TIOC0D/D
96	114	89	PE4	PE4/TIOC1A		PE4	PE4/TIOC1A		PE4	PE4/TIOC1A
109	115	102	PE5	PE5/TIOC1B		PE5	PE5/TIOC1B		PE5	PE5/TIOC1B
112	116	104	PE6	PE6/TIOC2A		PE6	PE6/TIOC2A		PE6	PE6/TIOC2A
113	137	105	PE7	PE7/TIOC2B		PE7	PE7/TIOC2B		PE7	PE7/TIOC2B
114	138	106	PE8	PE8/TIOC3A		PE8	PE8/TIOC3A		PE8	PE8/TIOC3A
115	139	107	PE9	PE9/TIOC3B		PE9	PE9/TIOC3B		PE9	PE9/TIOC3B
116	140	108	PE10	PE10/TIOC3C		PE10	PE10/TIOC3C		PE10	PE10/TIOC3C
118	142	110	PE11	PE11/TIOC3D		PE11	PE11/TIOC3D		PE11	PE11/TIOC3D
119	143	111	PE12	PE12/TIOC4A		PE12	PE12/TIOC4A		PE12	PE12/TIOC4A
120	144	112	PE13	PE13/TIOC4B/$\overline{\text{MRES}}$		PE13	PE13/TIOC4B/$\overline{\text{MRES}}$		PE13	PE13/TIOC4B/
2	2	1	PE14	PE14/TIOC4C/DACK0/$\overline{\text{AH}}$		PE14	PE14/TIOC4C/DACK0/$\overline{\text{AH}}$		PE14	PE14/TIOC4C/
3	5	2	PE15	PE15/TIOC4D/DACK1/$\overline{\text{IRQOUT}}$		PE15	PE15/TIOC4D/DACK1/$\overline{\text{IRQOUT}}$		PE15	PE15/TIOC4D/

Port A control register L1	PACRL1	R/W	H'0000 H'4000	H'FFFF838C H'FFFF838D	8, 16, 3
Port A control register L2	PACRL2	R/W	H'0000	H'FFFF838E H'FFFF838F	8, 16, 3
Port B I/O register	PBIOR	R/W	H'0000	H'FFFF8394 H'FFFF8395	8, 16, 3
Port B control register 1	PBCR1	R/W	H'0000	H'FFFF8398 H'FFFF8399	8, 16, 3
Port B control register 2	PBCR2	R/W	H'0000	H'FFFF839A H'FFFF839B	8, 16, 3
Port C I/O register	PCIOR	R/W	H'0000	H'FFFF8396 H'FFFF8397	8, 16, 3
Port C control register	PCCR	R/W	H'0000	H'FFFF839C H'FFFF839D	8, 16, 3
Port D I/O register H	PDIORH	R/W	H'0000	H'FFFF83A4 H'FFFF83A5	8, 16, 3
Port D I/O register L	PDIORL	R/W	H'0000	H'FFFF83A6 H'FFFF83A7	8, 16, 3
Port D control register H1	PDCRH1	R/W	H'0000	H'FFFF83A8 H'FFFF83A9	8, 16, 3
Port D control register H2	PDCRH2	R/W	H'0000	H'FFFF83AA H'FFFF83AB	8, 16, 3
Port D control register L	PDCRL	R/W	H'0000	H'FFFF83AC H'FFFF83AD	8, 16, 3
Port E I/O register	PEIOR	R/W	H'0000	H'FFFF83B4 H'FFFF83B5	8, 16, 3
Port E control register 1	PECR1	R/W	H'0000	H'FFFF83B8 H'FFFF83B9	8, 16, 3
Port E control register 2	PECR2	R/W	H'0000	H'FFFF83BA H'FFFF83BB	8, 16, 3
IRQOUT function control register	IFCR	R/W	H'0000	H'FFFF83C8 H'FFFF83C9	8, 16, 3

Note: * The port A control register L1 initial value varies depending on the operating mo

manual resets, reset by WDT, standby mode, or sleep mode, so the previous data is maint

The settings for this register are effective only for the 144-pin version. There are no corre
pins for this register in the 112-pin and 120-pin versions. However, read/writes are possib

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

Bit:	7	6	5	4	3	2	1
	PA23 IOR	PA22 IOR	PA21 IOR	PA20 IOR	PA19 IOR	PA18 IOR	PA17 IOR
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9
	PA15 IOR	PA14 IOR	PA13 IOR	PA12 IOR	PA11 IOR	PA10 IOR	PA9 IOR
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	PA7 IOR	PA6 IOR	PA5 IOR	PA4 IOR	PA3 IOR	PA2 IOR	PA1 IOR
Initial value:	0	0	0	0	0	0	0
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 eigh
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 ($\overline{\text{WRHH}}$, $\overline{\text{WRHL}}$, $\overline{\text{CA}}$
$\overline{\text{CASHL}}$, $\overline{\text{BACK}}$, $\overline{\text{BREQ}}$, $\overline{\text{WAIT}}$, $\overline{\text{AH}}$) and DMAC control signals (DRAK1, DRAK0), b
are instances when the register settings that select these pin functions will be ignored. R
table 18.2, Pin Arrangement by Mode.

PACRH is initialized to H'0000 by external power-on reset but is not initialized for man
reset by WDT, standby mode, or sleep mode, so the previous data is maintained.

Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R/W	R

- 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 inva
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 inva
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.

	1	Bus right request acknowledge (BACK) (PA19 in si mode)
1	0	DREQ1 request received output (DRAK1) (PA19 in chip mode)
	1	Reserved

- Bits 5 and 4—PA18 Mode 1, 0 (PA18MD1 and PA18MD0): These bits select the fu 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
1	0	DREQ0 request received output (DRAK0) (PA18 in 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.

Port A has bus control signals ($\overline{RD}$, $\overline{WRH}$, $\overline{WRL}$, $\overline{CS0}$–$\overline{CS3}$, $\overline{AH}$) and DMAC control sig
(DREQ0–DREQ1), but there are instances when the register settings that select these pin
will be ignored, depending on the operation mode. Refer to table 18.2, Pin Arrangement b
for details.

PACRL1 is initialized by external power-on reset to H'4000 in extended mode, and to H'0
single chip mode. PACRL2 is initialized by external power-on reset to H'0000. Neither re
initialized by manual resets, reset by WDT, standby mode, or sleep mode, so the previous
maintained.

Port A Control Register L1 (PACRL1):

Bit:	15	14	13	12	11	10	9
	—	PA15MD	—	PA14MD	—	PA13MD	—
Initial value:	0	0(1)*	0	0	0	0	0
R/W:	R	R/W	R	R/W	R	R/W	R

Bit:	7	6	5	4	3	2	1
	—	PA11MD	—	PA10MD	PA9MD1	PA9MD0	PA8MD1
Initial value:	0	0	0	0	0	0	0
R/W:	R	R/W	R	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.

1	Read output (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/$\overline{\text{WRH}}$ pin.

Bit 10: PA13MD	Description
0	General input/output (PA13) (initial value) ($\overline{\text{WRH}}$ in on-chip ROM inva
1	Most significant side write output ($\overline{\text{WRH}}$) (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/$\overline{\text{WRL}}$ pin.

Bit 8: PA12MD	Description
0	General input/output (PA12) (initial value) ($\overline{\text{WRL}}$ in on-chip ROM inval
1	Least significant side write output ($\overline{\text{WRL}}$) (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/$\overline{\text{CS1}}$ pin.

Bit 6: PA11MD	Description
0	General input/output (PA11) (initial value) ($\overline{\text{CS1}}$ in on-chip ROM invali
1	Chip select output ($\overline{\text{CS1}}$) (PA11 in single chip mode)

- Bit 5—Reserved: This bit always reads as 0. The write value should always be 0.

	1	MTU timer clock input (TCLKD)
1	0	Interrupt request input ($\overline{\text{IRQ3}}$)
	1	Reserved

- Bits 1 and 0—PA8 Mode 1, 0 (PA8MD1 and PA8MD0): These bits select the functio PA8/TCLKC/$\overline{\text{IRQ2}}$ 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 ($\overline{\text{IRQ2}}$)
	1	Reserved

R/W:	R	R/W	R/W	R/W	R	R/W	R

- Bits 15 and 14—PA7 Mode 1, 0 (PA7MD1 and PA7MD0): These bits select the fun
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 fun
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

- 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 functio 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 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.

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 outpu
ten port B pins. Bits PB9IOR–PB0IOR correspond to the PB9/$\overline{\text{IRQ7}}$/A21/$\overline{\text{ADTRG}}$ pin t
pin. PBIOR is enabled when the port B pins function as input/outputs (PB9–PB0). For o
functions, it is disabled.

For port B pin functions PB9–PB0, a given pin in port B is an output pin if its correspon
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
resets, reset by WDT, standby mode, or sleep mode, so the previous data is maintained.

Bit:	15	14	13	12	11	10	9
	—	—	—	—	—	—	PB9 IOR
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W

Bit:	7	6	5	4	3	2	1
	PB7 IOR	PB6 IOR	PB5 IOR	PB4 IOR	PB3 IOR	PB2 IOR	PB1 IOR
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

maintained.

Port B Control Register 1 (PBCR1):

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

Bit:	7	6	5	4	3	2	1
	—	—	—	—	PB9 MD1	PB9 MD0	PB8 MD1
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	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 th 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 ($\overline{\text{ADTRG}}$)

Bit:	15	14	13	12	11	10	9
	PB7MD1	PB7MD0	PB6MD1	PB6MD0	PB5MD1	PB5MD0	PB4MD1
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	PB3MD1	PB3MD0	PB2MD1	PB2MD0	—	PB1MD	—
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R/W	R

- Bits 15 and 14—PB7 Mode (PB7MD1 and PB7MD0): PB7MD1 and PB7MD0 selec 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 m

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 th 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 m

- Bits 7 and 6—PB3 Mode (PB3MD1 and PB3MD0): PB3MD1 and PB3MD0 select th 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 m

Bit 2: PB1MD	Description
0	General input/output (PB1) (initial value) (A17 in on-chip ROM invalid m
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 m
1	Address output (A16) (PB0 in single chip mode)

	PC15 IOR	PC14 IOR	PC13 IOR	PC12 IOR	PC11 IOR	PC10 IOR	PC9 IOR
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	PC7 IOR	PC6 IOR	PC5 IOR	PC4 IOR	PC3 IOR	PC2 IOR	PC1 IOR
Initial value:	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:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	PC7 MD	PC6 MD	PC5 MD	PC4 MD	PC3 MD	PC2 MD	PC1 MD
Initial value:	0	0	0	0	0	0	0
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 inval
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 inval
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 inval
1	Address output (A13) (PC13 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
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 m
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 m
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 m
1	Address output (A7) (PC7 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
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
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
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
1	Address output (A1) (PC1 in single chip mode)

For port D pin functions PD31–PD16, a given pin in port D is an output pin if its corresp
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
manual resets, reset by WDT, standby mode, or sleep mode, so the previous data is maint

The settings for this register are effective only for the 144-pin version. There are no corre
pins for this register in the 112-pin and 120-pin versions. However, read/writes are possib

Bit:	15	14	13	12	11	10	9
	PD31 IOR	PD30 IOR	PD29 IOR	PD28 IOR	PD27 IOR	PD26 IOR	PD25 IOR
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	PD23 IOR	PD22 IOR	PD21 IOR	PD20 IOR	PD19 IOR	PD18 IOR	PD17 IOR
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9
	PD15 IOR	PD14 IOR	PD13 IOR	PD12 IOR	PD11 IOR	PD10 IOR	PD9 IOR
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	PD7 IOR	PD6 IOR	PD5 IOR	PD4 IOR	PD3 IOR	PD2 IOR	PD1 IOR
Initial value:	0	0	0	0	0	0	0
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 mo 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 PD23/D23/$\overline{\text{IRQ7}}$–PD16/D16/$\overline{\text{IRQ0}}$ pins of port D. There are instances when these regist will be ignored, depending on the operation mode. Refer to table 18.2, Pin Arrangement for details.

The settings for this register are effective only for the 144-pin version. There are no corr pins for this register in the 112-pin and 120-pin versions. However, read/writes are possi

PDCRH1 and PDCRH2 are both initialized to H'0000 by external power-on reset but are initialized for manual resets, reset by WDT, standby mode, or sleep mode, so the previo maintained.

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 f 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, D 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 CS0 = 32 bit width)
	1	Reserved

- Bits 13 and 12—PD30 Mode 1, 0 (PD30MD1 and PD30MD0): These bits select the f 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, D 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 with CS0 = 32 bit width. Reserved in single chip mode
	1	Reserved

- Bits 9 and 8—PD28 Mode 1, 0 (PD28MD1 and PD28MD0): These bits select the fu
the PD28/D28/$\overline{CS2}$ pin.

Bit 9: PD28MD1	Bit 8: PD28MD0	Description
0	0	General input/output (PD28) (initial value) (D28 with 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, 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 fu
the PD27/D27/DACK1 pin.

Bit 7: PD27MD1	Bit 6: PD27MD0	Description
0	0	General input/output (PD27) (initial value) (D27 with and CS0 = 32 bit width)
	1	Data input/output (D27) (PD27 in single chip mode)
1	0	DMA transfer request received output (DACK1) (PD2 chip mode, and D27 with no ROM and CS0 = 32 bit
	1	Reserved

- Bits 3 and 2—PD25 Mode 1, 0 (PD25MD1 and PD25MD0): These bits select the fun
the PD25/D25/$\overline{\text{DREQ1}}$ pin.

Bit 3: PD25MD1	Bit 2: PD25MD0	Description
0	0	General input/output (PD25) (initial value) (D25 with n 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 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 fun
the PD24/D24/$\overline{\text{DREQ0}}$ pin.

Bit 1: PD24MD1	Bit 0: PD24MD0	Description
0	0	General input/output (PD24) (initial value) (D24 with n 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 mode, and D24 with no ROM and CS0 = 32 bit width)
	1	Reserved

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 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 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 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 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

the PD20/D20/IRQ4 pin.

Bit 9: PD20MD1	Bit 8: PD20MD0	Description
0	0	General input/output (PD20) (initial value) (D20 with n 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 fun the PD19/D19/$\overline{\text{IRQ3}}$ pin.

Bit 7: PD19MD1	Bit 6: PD19MD0	Description
0	0	General input/output (PD19) (initial value) (D19 with n 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

the PD17/D17/$\overline{\text{IRQ1}}$ pin.

Bit 3: PD17MD1	Bit 2: PD17MD0	Description
0	0	General input/output (PD17) (initial value) (D17 with 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 fu
the PD16/D16/$\overline{\text{IRQ0}}$ pin.

Bit 1: PD16MD1	Bit 0: PD16MD0	Description
0	0	General input/output (PD16) (initial value) (D16 with 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

— 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 a 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 manu 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
	PD15 MD	PD14 MD	PD13 MD	PD12 MD	PD11 MD	PD10 MD	PD9 MD
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	PD7 MD	PD6 MD	PD5 MD	PD4 MD	PD3 MD	PD2 MD	PD1 MD
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- 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 invali
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 invali
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 invali
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 invali
1	Data input/output (D10) (PD10 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 mo
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 mo
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 mo
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 mo
1	Data input/output (D4) (PD4 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 m
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 m
1	Data input/output (D0) (PD0 in single chip mode)

	15	14	13	12	11	10	9
	PE15 IOR	PE14 IOR	PE13 IOR	PE12 IOR	PE11 IOR	PE10 IOR	PE9 IOR
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	PE7 IOR	PE6 IOR	PE5 IOR	PE4 IOR	PE3 IOR	PE2 IOR	PE1 IOR
Initial value:	0	0	0	0	0	0	0
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 PECR2 selects the function of the lower eight bit pins of port E.

Port E has a bus control signal ($\overline{AH}$) and DMAC control signals (DACK1, DACK0, DRA DRAK0), but there are instances when the register settings that select these pin functions ignored, depending on the operation mode. Refer to table 18.2, Pin Arrangement by Mod details.

PECR1 and PECR2 are both initialized to H'0000 by external power-on reset but are not i for manual resets, reset by WDT, standby mode, or sleep mode, so the previous data is m

- Bits 15 and 14—PE15 Mode 1, 0 (PE15MD1 and PE15MD0): These bits select the f
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 (TI
1	0	DMAC request received output (DACK1) (PE15 in s 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 f
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 (TI
1	0	DMAC request received output (DACK0) (PE14 in s mode)
	1	Address hold output ($\overline{\text{AH}}$) (PE14 in single chip mod

- 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.

1	MTU input capture input/output compare output (TIOC3A)

Port E Control Register 2 (PECR2):

Bit:	15	14	13	12	11	10	9
	—	PE7MD	—	PE6MD	—	PE5MD	—
Initial value:	0	0	0	0	0	0	0
R/W:	R	R/W	R	R/W	R	R/W	R

Bit:	7	6	5	4	3	2	1
	PE3 MD1	PE3 MD0	PE2 MD1	PE2 MD0	PE1 MD1	PE1 MD0	PE0 MD1
Initial value:	0	0	0	0	0	0	0
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.

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 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 (TIO
1	0	$\overline{\text{DREQ1}}$ request received output (DRAK1) (PE3 in sir mode)
	1	Reserved

Bit 3: PE1MD1	Bit 2: PE1MD0	Description
0	0	General input/output (PE1) (initial value)
	1	MTU input capture input/output compare output (TI
1	0	$\overline{\text{DREQ0}}$ request received output (DRAK0) (PE1 in s mode)
	1	Reserved

- Bits 1 and 0—PE0 Mode 1, 0 (PE0MD1 and PE0MD0): These bits select the functio 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 (TI
1	0	$\overline{\text{DREQ0}}$ request receive input (PE0 in single chip m
	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 pin established as $\overline{\text{IRQOUT}}$ outputs by the port D control register (PDCRH1) or port E cont (PECR1). When PDCRH1 or PECR1 are set for any other function, the settings of this r have no effect on the pin functions.

R/W:	R	R	R	R	R/W	R/W	R/W

- Bits 3 and 2—IRQOUT Mode 3, 2 (IRQMD3 and IRQMD2): These bits select the $\overline{IR}$ pin function when the PDCRH1 bits 13 and 12 (PD30MD1, PD30MD0) are set to (1, bit settings are effective only for the 144 pin version. Reads and writes are also possib 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 outp (which of the two is output depends on the oper status at the time)
	1	Always high level output

- Bits 1 and 0—IRQOUT Mode 1, 0 (IRQMD1 and IRQMD0): These bits select the $\overline{IR}$ 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 outp (which of the two is output depends on the oper status at the time)
	1	Always high level output

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 tab

PA8 (I/O)/TCLKC (input)/$\overline{\text{IRQ2}}$ (input)	PA8 (I/O)/TCLKC (input)/$\overline{\text{IRQ2}}$ (input)	PA8 (I/O)/TCLKC (inp (input)
PA7 (I/O)/TCLKB (input)/$\overline{\text{CS3}}$ (output)	PA7 (I/O)/TCLKB (input)/$\overline{\text{CS3}}$ (output)	PA7 (I/O)/TCLKB (inp
PA6 (I/O)/TCLKA (input)/$\overline{\text{CS2}}$ (output)	PA6 (I/O)/TCLKA (input)/$\overline{\text{CS2}}$ (output)	PA6 (I/O)/TCLKA (inp
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)/ (input)
PA4 (I/O)/TXD1 (output)	PA4 (I/O)/TXD1 (output)	PA4 (I/O)/TXD1 (outp
PA3 (I/O)/RXD1 (input)	PA3 (I/O)/RXD1 (input)	PA3 (I/O)/RXD1 (inpu
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)/ (input)
PA1 (I/O)/TXD0 (output)	PA1 (I/O)/TXD0 (output)	PA1 (I/O)/TXD0 (outp
PA0 (I/O)/RXD0 (input)	PA0 (I/O)/RXD0 (input)	PA0 (I/O)/RXD0 (inpu

In the FP-144 version, port A is a 24-pin input/output port, as listed in table 19.2.

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 (outp
$\overline{\text{RD}}$ (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 (in (input)
PA8 (I/O)/TCLKC (input)/$\overline{\text{IRQ2}}$ (input)	PA8 (I/O)/TCLKC (input)/$\overline{\text{IRQ2}}$ (input)	PA8 (I/O)/TCLKC (in (input)
PA7 (I/O)/TCLKB (input)/$\overline{\text{CS3}}$ (output)	PA7 (I/O)/TCLKB (input)/$\overline{\text{CS3}}$ (output)	PA7 (I/O)/TCLKB (in
PA6 (I/O)/TCLKA (input)/$\overline{\text{CS2}}$ (output)	PA6 (I/O)/TCLKA (input)/$\overline{\text{CS2}}$ (output)	PA6 (I/O)/TCLKA (in
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 (input)
PA4 (I/O)/TXD1 (output)	PA4 (I/O)/TXD1 (output)	PA4 (I/O)/TXD1 (out
PA3 (I/O)/RXD1 (input)	PA3 (I/O)/RXD1 (input)	PA3 (I/O)/RXD1 (inp
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 (input)
PA1 (I/O)/TXD0 (output)	PA1 (I/O)/TXD0 (output)	PA1 (I/O)/TXD0 (out
PA0 (I/O)/RXD0 (input)	PA0 (I/O)/RXD0 (input)	PA0 (I/O)/RXD0 (inp

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–PA1(correspond to the PA23/$\overline{\text{WRHH}}$–PA16/$\overline{\text{AH}}$ pins. When the pins are used as ordinary outp will output whatever value is written in the PADRH; when PADRH is read, the register v be output regardless of the pin status. When the pins are used as ordinary inputs, the pin s rather than the register value is read directly when PADRH is read. When a value is writt PADRH, that value can be written into PADRH, but it will not affect the pin status. Table 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 manual reset, reset by WDT, standby mode, or sleep mode.

These register settings function only for the 144-pin version. There are no pins correspon this register in the 112-pin version. However, read/writes are possible.

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

Bit:	7	6	5	4	3	2	1
	PA23DR	PA22DR	PA21DR	PA20DR	PA19DR	PA18DR	PA17DR
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9
	PA15DR	PA14DR	PA13DR	PA12DR	PA11DR	PA10DR	PA9DR
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	PA7DR	PA6DR	PA5DR	PA4DR	PA3DR	PA2DR	PA1DR
Initial value:	0	0	0	0	0	0	0
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
	Other function	Pin status	Can write to PADR, but it has no effect on
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

(output)/BREQ (input)	(output)/BREQ (input)	
PB6 (I/O)/IRQ4 (input)/A18 (output)/BACK (output)	PB6 (I/O)/IRQ4 (input)/A18 (output)/BACK (input)	PB6 (I/O)/IRQ4 (input
PB5 (I/O)/IRQ3 (input)/POE3 (input)/RDWR (output)	PB5 (I/O)/IRQ3 (input)/POE3 (input)/RDWR (output)	PB5 (I/O)/IRQ3 (input (input)
PB4 (I/O)/IRQ2 (input)/POE2 (input)/CASH (output)	PB4 (I/O)/IRQ2 (input)/POE2 (input)/CASH (output)	PB4 (I/O)/IRQ2 (input (input)
PB3 (I/O)/IRQ1 (input)/POE1 (input)/CASL (output)	PB3 (I/O)/IRQ1 (input)/POE1 (input)/CASL (output)	PB3 (I/O)/IRQ1 (input (input)
PB2 (I/O)/IRQ0 (input)/POE0 (input)/RAS (output)	PB2 (I/O)/IRQ0 (input)/POE0 (input)/RAS (output)	PB2 (I/O)/IRQ0 (input (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
Port B data register	PBDR	R/W	H'0000	H'FFFF8390 H'FFFF8391	8, 16, 32

Bit:	15	14	13	12	11	10	9
	—	—	—	—	—	—	PB9DR
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W

Bit:	7	6	5	4	3	2	1
	PB7DR	PB6DR	PB5DR	PB4DR	PB3DR	PB2DR	PB1DR
Initial value:	0	0	0	0	0	0	0
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 pi
	Other function	Pin status	Can write to PBDR, but it has no effect on pi
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 pi

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
Port C data register	PCDR	R/W	H'0000	H'FFFF8392 H'FFFF8393	8, 16, 32

Bit:	15	14	13	12	11	10	9
	PC15DR	PC14DR	PC13DR	PC12DR	PC11DR	PC10DR	PC9DR
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	PC7DR	PC6DR	PC5DR	PC4DR	PC3DR	PC2DR	PC1DR
Initial value:	0	0	0	0	0	0	0
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 p
	Other function	Pin status	Can write to PCDR, but it has no effect on p
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 p

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.

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)/$\overline{\text{IR}}$ (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)/$\overline{\text{IR}}$ (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)/$\overline{\text{IR}}$ (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)/$\overline{\text{IR}}$ (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)/$\overline{\text{IR}}$ (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)/$\overline{\text{IR}}$ (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)/$\overline{\text{IR}}$ (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)/$\overline{\text{IR}}$ (input)

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
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

These register settings function only for the 144-pin version. There are no pins correspo
this register in the 112-pin version. However, read/writes are possible.

Bit:	15	14	13	12	11	10	9
	PD31DR	PD30DR	PD29DR	PD28DR	PD27DR	PD26DR	PD25DR
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	PD23DR	PD22DR	PD21DR	PD20DR	PD19DR	PD18DR	PD17DR
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9
	PD15DR	PD14DR	PD13DR	PD12DR	PD11DR	PD10DR	PD9DR
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	PD7DR	PD6DR	PD5DR	PD4DR	PD3DR	PD2DR	PD1DR
Initial value:	0	0	0	0	0	0	0
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 pi
	Other function	Pin status	Can write to PDDR, but it has no effect on pi
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 pi

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	Acces
Port E data register	PEDR	R/W	H'0000	H'FFFF83B0 H'FFFF83B1	8, 16, 3

Bit:	15	14	13	12	11	10	9
	PE15DR	PE14DR	PE13DR	PE12DR	PE11DR	PE10DR	PE9DR
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	PE7DR	PE6DR	PE5DR	PE4DR	PE3DR	PE2DR	PE1DR
Initial value:	0	0	0	0	0	0	0
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 pi
	Other function	Pin status	Can write to PEDR, but it has no effect on pi
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 pi

- 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	Acces
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 Any value written into these bits is ignored, and there is no effect on the status of the pi any of the bits are read, the pin status rather than the bit value is read directly. However, A/D converter analog input is being sampled, values of 1 are read out. Table 19.19 show read/write operations of the port F data register.

PFDR is not initialized by power-on resets, manual resets, standby mode, or sleep mode always reflect the pin status).

Bit:	7	6	5	4	3	2	1
	PF7DR	PF6DR	PF5DR	PF4DR	PF3DR	PF2DR	PF1DR
Initial value:	*	*	*	*	*	*	*
R/W:	R	R	R	R	R	R	R

Note: * Initial values are dependent on the status of the pins at the time of the reads.

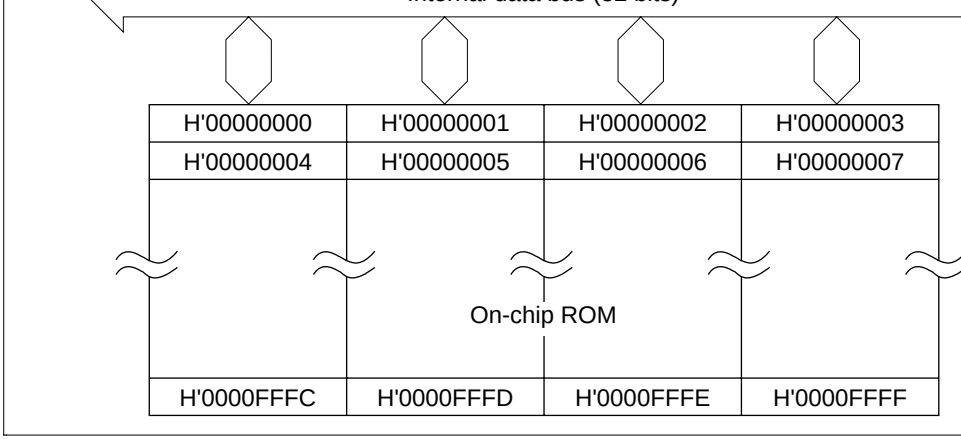


Figure 20.1 Mask ROM Block Diagram (64-kbyte Version)

H'0001FFFC	H'0001FFFD	H'0001FFFE	H'0001FFFF

Figure 20.2 Mask ROM Block Diagram (128-kbyte Version)

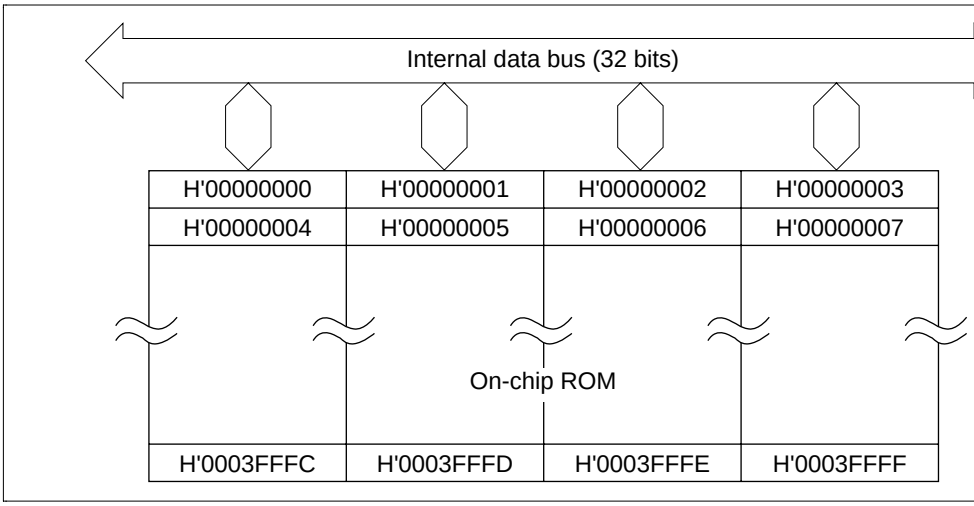


Figure 20.3 Mask ROM Block Diagram (256-kbyte Version)

Mode 1 (MCU mode 1)	*	*	0	1	On-chip ROM invalid, external 16-bit sp pin and 120 pin), external 32-bit space
Mode 2 (MCU mode 2)	*	*	1	0	On-chip ROM valid, external space (bu 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.

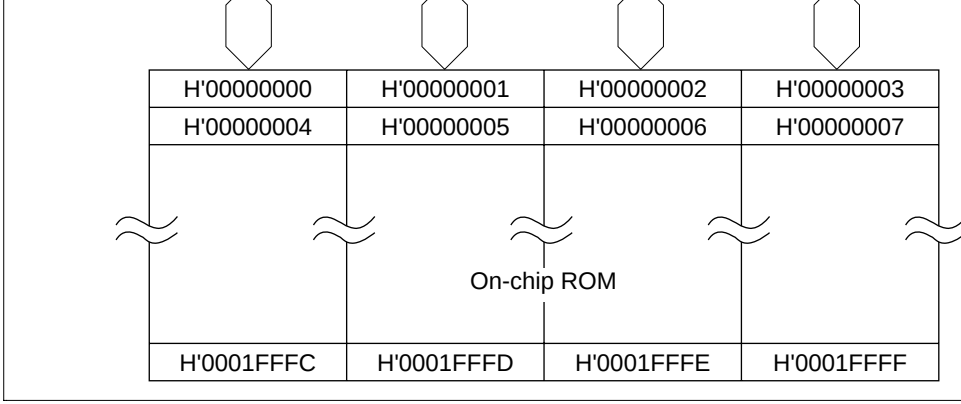


Figure 21.1 PROM Block Diagram

The operating mode determines whether the on-chip ROM is valid or not. The operating
selected using mode-setting pins MD3–MD0 as shown in table 21.1. If you are using the
ROM, select mode 2 or mode 3; if you are not, select mode 0 or 1. The on-chip ROM is
to addresses H'00000000–H'0001FFFF of memory area 0.

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

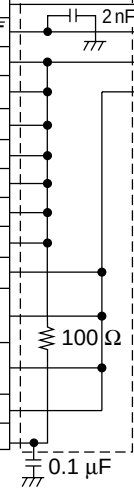
When programming the on-chip PROM, set the pins as shown in figure 21.2, 21.3, or 21.4 perform the programming in PROM mode.

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 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 pins and HN27C101 pins. Figure 21.5 is a memory map of the on-chip ROM.

6	PC2/A2
7	PC3/A3
8	PC4/A4
9	PC5/A5
10	PC6/A6
11	PC7/A7
12	PC8/A8
25	PB3/$\overline{\text{RD1}}$/$\overline{\text{POE1}}$/$\overline{\text{CASL}}$
14	PC10/A10
15	PC11/A11
16	PC12/A12
17	PC13/A13
18	PC14/A14
19	PC15/A15
20	PB0/A16
26	PB4/$\overline{\text{IRQ2}}$/$\overline{\text{POE2}}$/$\overline{\text{CASH}}$
2	PE15/TIOC4D/DACK1/$\overline{\text{IRQOUT}}$
1	PE14/TIOC4C/DACK0/$\overline{\text{AH}}$
28	PB5/$\overline{\text{IRQ3}}$/$\overline{\text{POE3}}$/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

A2	10
A3	9
A4	8
A5	7
A6	6
A7	5
A8	27
$\overline{\text{OE}}$	24
A10	23
A11	25
A12	4
A13	28
A14	29
A15	3
A16	2
$\overline{\text{PGM}}$	31
$\overline{\text{CE}}$	22
V_{CC}	32
V_{SS}	16



V_{PP}: PROM p
power su
(12.5 V)
A16–A0: Address
I/O7–I/O0: Data inp
output
$\overline{\text{OE}}$: Output e
$\overline{\text{PGM}}$: Program
$\overline{\text{CE}}$: Chip ena

Figure 21.2 SH7042 Pin and HN27C101 Pin Correspondence (112-Pin Vers

7	PC2/A2	A2	10
8	PC3/A3	A3	9
9	PC4/A4	A4	8
10	PC5/A5	A5	7
11	PC6/A6	A6	6
12	PC7/A7	A7	5
13	PC8/A8	A8	27
26	PB3/$\overline{\text{IRQ1}}$/POE1/$\overline{\text{CASL}}$	$\overline{\text{OE}}$	24
15	PC10/A10	A10	23
16	PC11/A11	A11	25
17	PC12/A12	A12	4
18	PC13/A13	A13	28
19	PC14/A14	A14	29
20	PC15/A15	A15	3
21	PB0/A16	A16	2
27	PB4/$\overline{\text{IRQ2}}$/POE2/$\overline{\text{CASH}}$	$\overline{\text{PGM}}$	31
3	PE15/TIOC4D/DACK1/$\overline{\text{IRQOUT}}$	$\overline{\text{CE}}$	22
2	PE14/TIOC4C/DACK0/$\overline{\text{AH}}$	Vcc	32
29	PB5/$\overline{\text{IRQ3}}$/$\overline{\text{POE3}}$/RDWR	Vss	16
22, 40, 70, 82, 111	VCC		
84	MD0		
83	MD1		
80	MD2		
85, 107	PLLVCC, AVCC		
86, 87, 79	PLLCAP, PLLVSS, EXTAL		
4, 24, 28, 36, 42 58, 66, 76, 97, 108, 117	VSS		
98 to 103 105 to 106	PF0/AN0 to PF5/AN5 PF6/AN6 to PF7/AN7		
104	AVSS		
78	MD3		

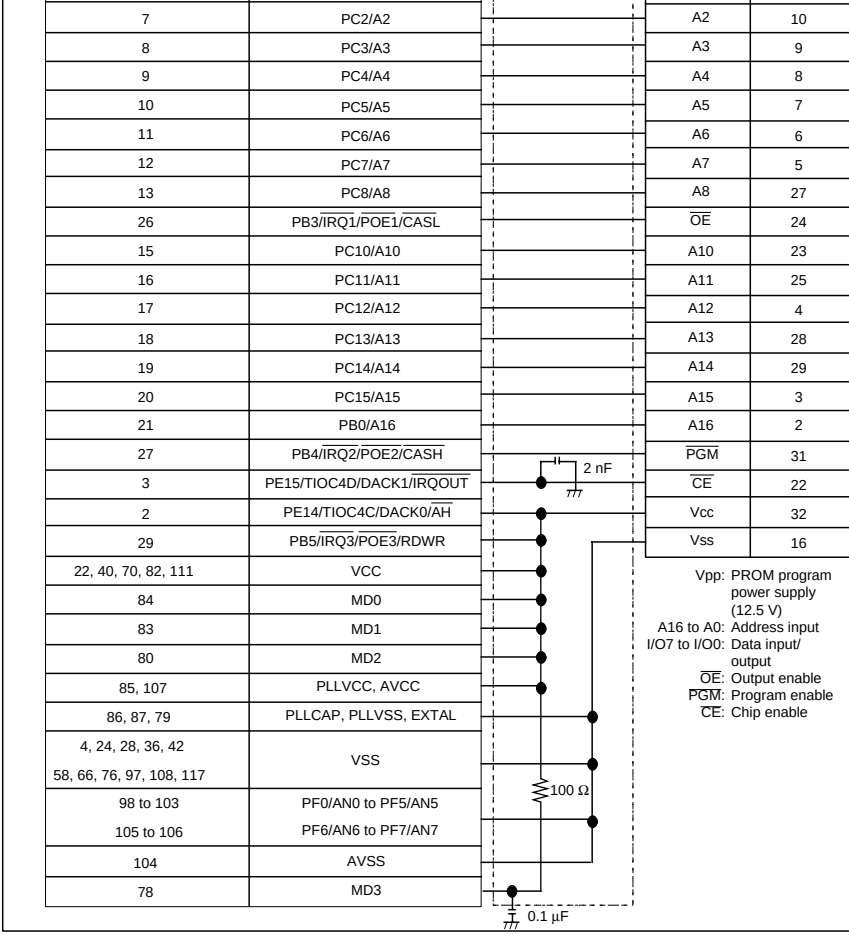
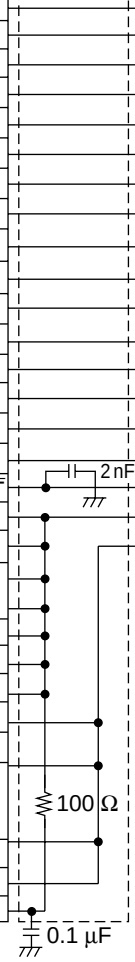


Figure 21.3 SH7042 Pin and HN27C101 Pin Correspondence (120-Pin Versio

9	PC2/A2	A2	10
10	PC3/A3	A3	9
11	PC4/A4	A4	8
13	PC5/A5	A5	7
15	PC6/A6	A6	6
16	PC7/A7	A7	5
17	PC8/A8	A8	27
32	PB3/$\overline{\text{IRQ1}}$/$\overline{\text{POE1}}$/$\overline{\text{CASL}}$	$\overline{\text{OE}}$	24
19	PC10/A10	A10	23
20	PC11/A11	A11	25
21	PC12/A12	A12	4
22	PC13/A13	A13	28
23	PC14/A14	A14	29
24	PC15/A15	A15	3
25	PB0/A16	A16	2
34	PB4/$\overline{\text{IRQ2}}$/$\overline{\text{POE2}}$/$\overline{\text{CASH}}$	$\overline{\text{PGM}}$	31
5	PE15/TIOC4D/DACK1/$\overline{\text{IRQOUT}}$	$\overline{\text{CE}}$	22
2	PE14/TIOC4D/DACK0/$\overline{\text{AH}}$	V_{CC}	32
36	PB5/$\overline{\text{IRQ3}}$/$\overline{\text{POE3}}$/RDWR	V_{SS}	16
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		



V_{PP}: PROM
power
(12.5 V
A16–A0: Addres
I/O7–I/O0: Data i
output
$\overline{\text{OE}}$: Outpu
$\overline{\text{PGM}}$: Progra
$\overline{\text{CE}}$: Chip e

Figure 21.4 SH7043 Pin and HN27C101 Pin Correspondence (144-Pin Vers

21.3 PROM Programming

The PROM mode write/verify specifications are the same as those of the standard EPROI HN27C101. However, because the page program format is not supported, do not set the P writer to the page programming mode. PROM writers that only support page programmin cannot be used. When selecting a PROM writer, confirm that it supports the byte-by-byte 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 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–A**
Write	0	1	0	V_{PP}	V_{CC}	Data input	Addres 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.

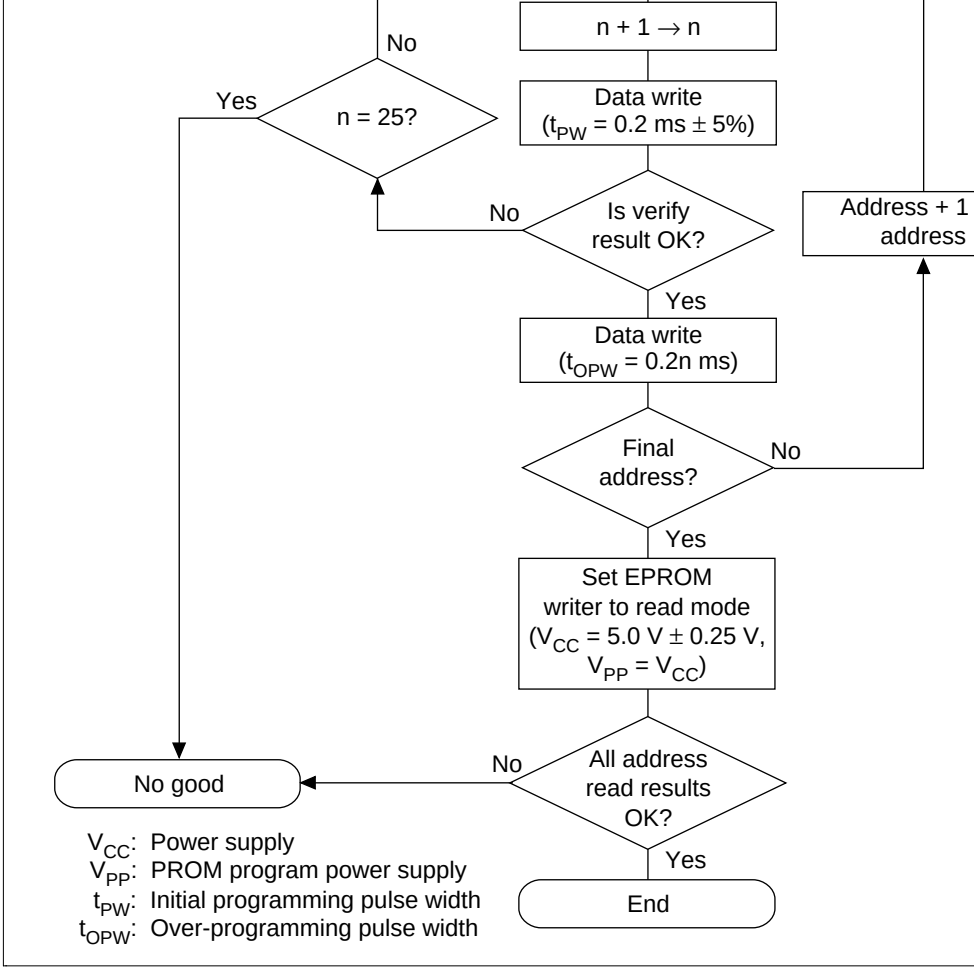


Figure 21.6 High-Speed, High-Reliability Programming Basic Flow

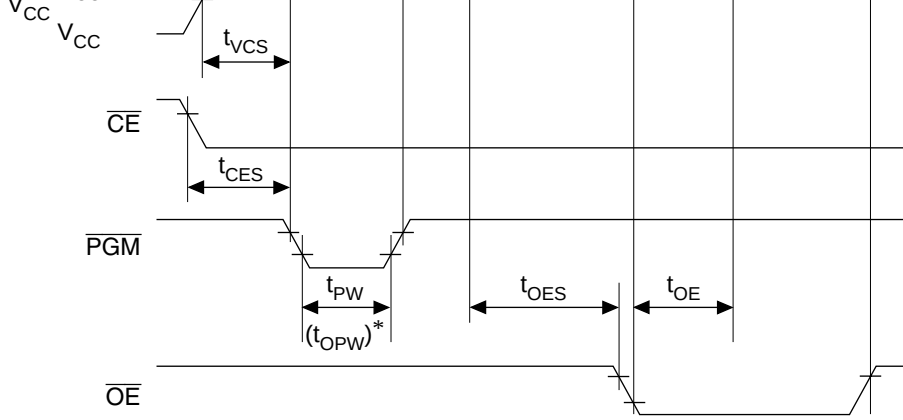
voltage							
Output low-level voltage	I/O7–I/O0	V_{OL}	—	—	0.45	V	I_{OL} = 1.6
Input leak current	I/O7–I/O0, A16–A0, $\overline{OE}$, $\overline{CE}$, $\overline{PGM}$	$\lvert I_{LI} \rvert$	—	—	2	µA	V_{IN} = 5.2
V_{CC} current		I_{CC}	—	—	80	mA	
V_{PP} current		I_{PP}	—	—	80	mA	

$\overline{PGM}$ 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 refe
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 outp
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 volta (programming voltage) V_{PP} is 12.5 V (when the EPROM writer is set for the HN27C Hitachi specifications, V_{PP} becomes 12.5 V). Devices will sometimes be destroyed if higher than the rated one is applied. Pay particular attention to such phenomena as E writer overshoot.
2. Always confirm that the indices of the EPROM writer socket, socket adapter, and de agreement before programming. Devices will sometimes be destroyed due to excessi flow if these are not connected in the proper locations.
3. Do not touch the socket adapter or device during writing. Contact faults can sometim devices to be improperly written.
4. Page programming mode writes are not possible. Always set to byte programming m

screening to the installation of the device on a board.

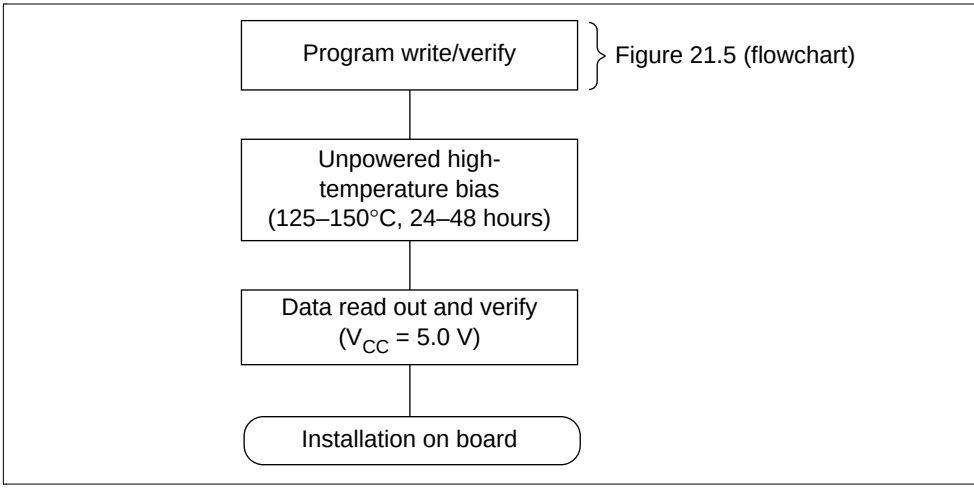


Figure 21.8 Screening Flow

If there are any abnormalities in program write/verify or program read-out verification af
temperature biasing, please contact a Renesas Technology technical representative.

— Erase-verify mode

- Programming/erase methods

  The flash memory is programmed 32 bytes at a time. Block erase (in single-block un
  performed. Block erasing can be performed as required on 1 kbyte, 28 kbyte, and 32
  blocks.

- Programming/erase times

  The flash memory programming time is 10 ms (typ.) for simultaneous 32-byte progr
  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-

  — 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 m
  transfer bit rate of the host.

- Flash memory emulation in RAM

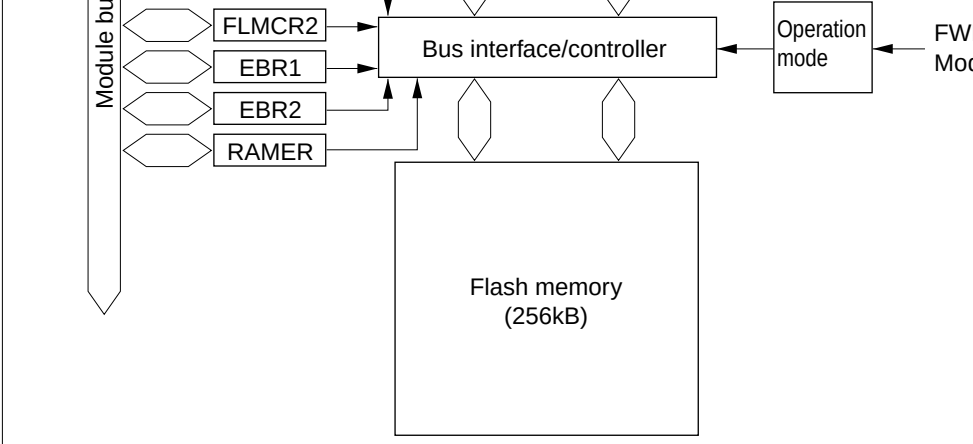
  Flash memory programming can be emulated in real time by overlapping a part of R
  flash memory.

- Protect modes

  There are two protect modes, hardware and software, which allow protected status to
  designated for flash memory program/erase/verify operations

- Programmer mode

  Flash memory can be programmed/erased in programmer mode, using a PROM prog
  as well as in on-board programming mode.



Legend
FLMCR1: Flash memory control register 1
FLMCR2: Flash memory control register 2
EBR1 : Block specification register 1
EBR2 : Block specification register 2
RAMER : RAM emulation register

Figure 22.1 Flash Memory Block Diagram

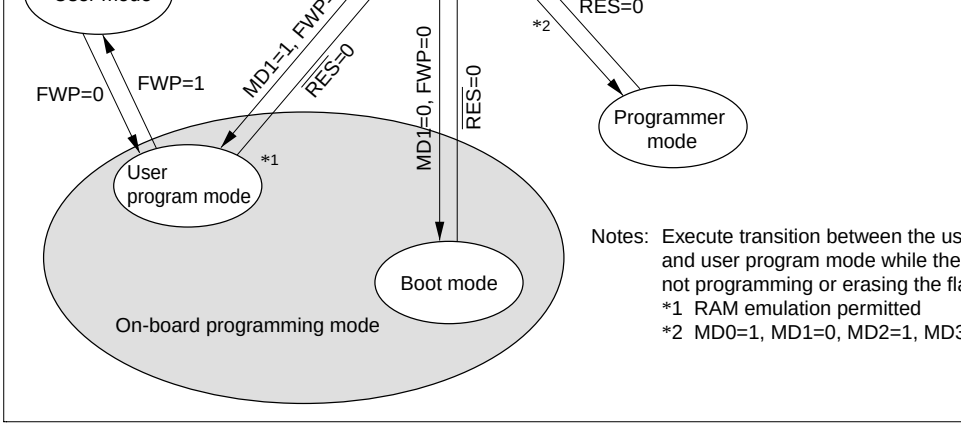


Figure 22.2 Flash Memory Mode Transitions

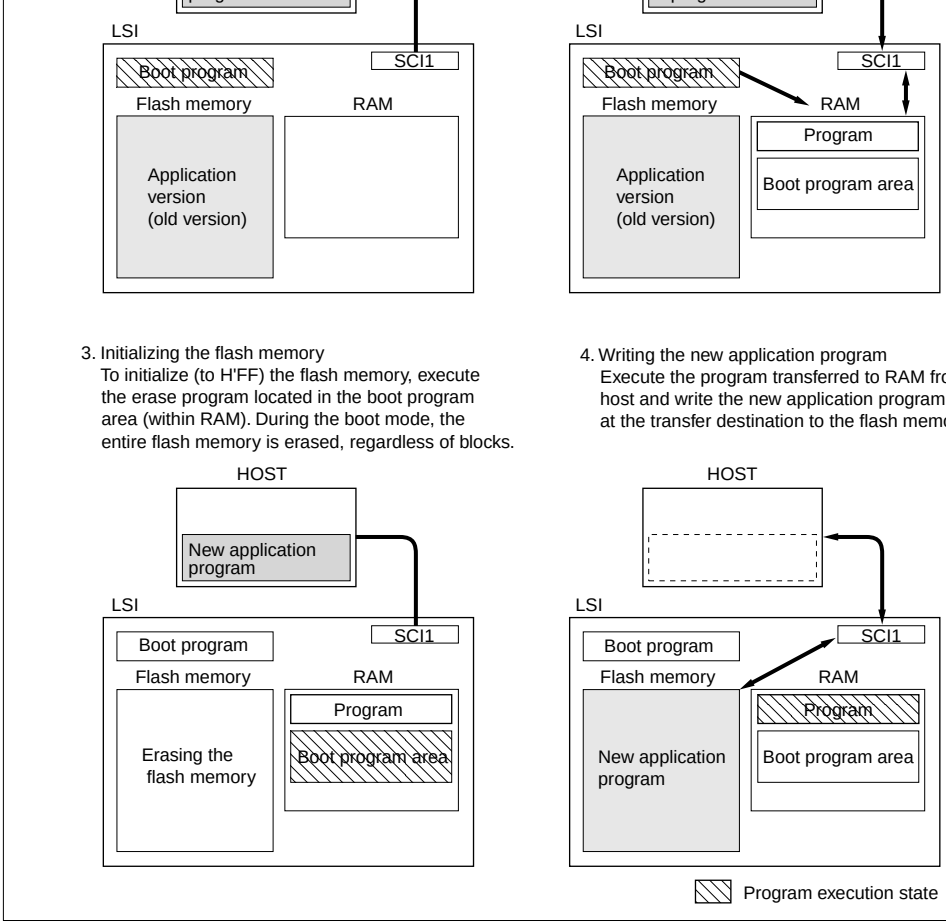


Figure 22.3 Boot Mode

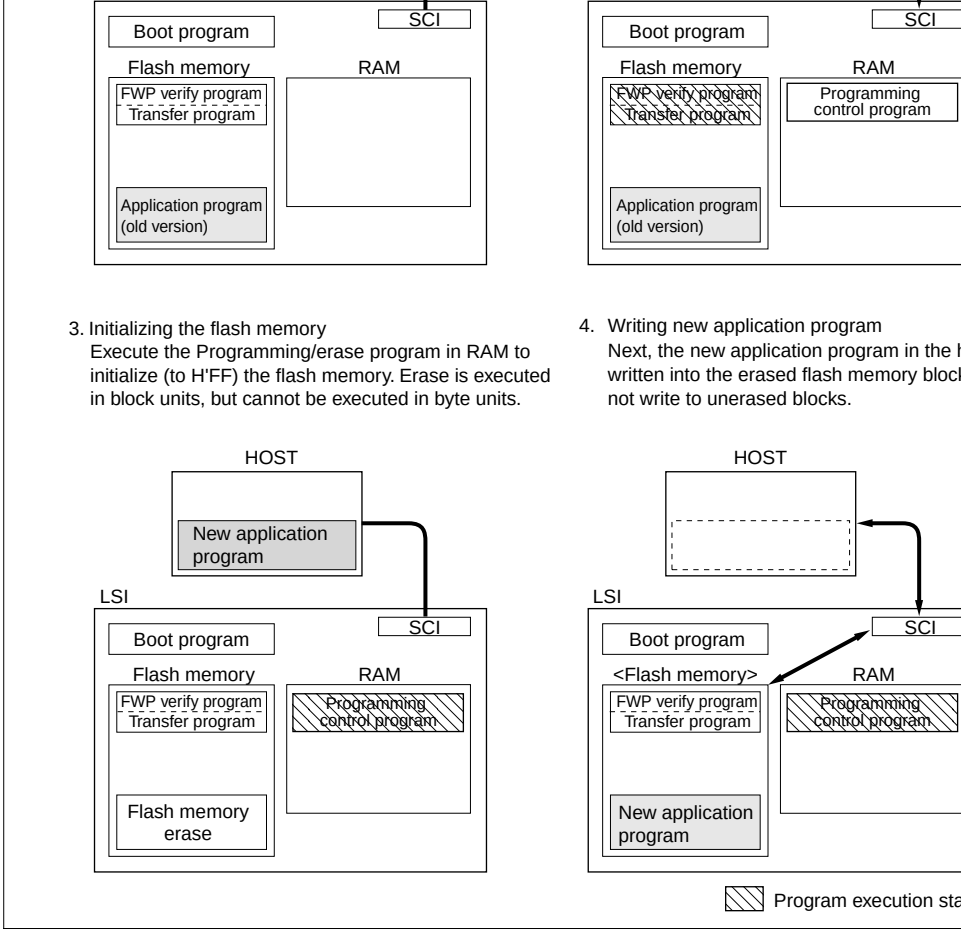


Figure 22.4 User Program Mode

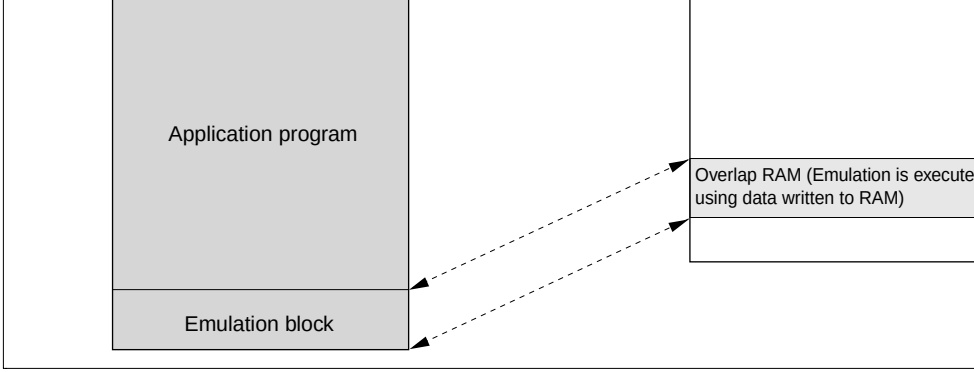


Figure 22.5 Emulation

When overlap RAM data is confirmed, the RAMS bit is cleared, RAM overlap is rele
writes should actually be performed to the flash memory.

When the programming control program is transferred to RAM, ensure that the transf
destination and the overlap RAM do not overlap, as this will cause data in the overlap
be rewritten.

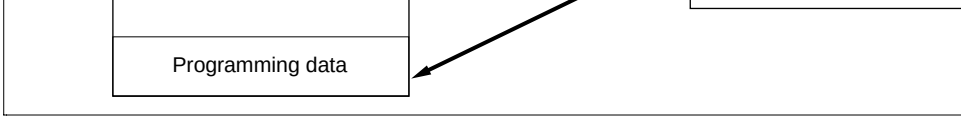


Figure 22.6 Programming to the Flash Memory

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 Mo
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.

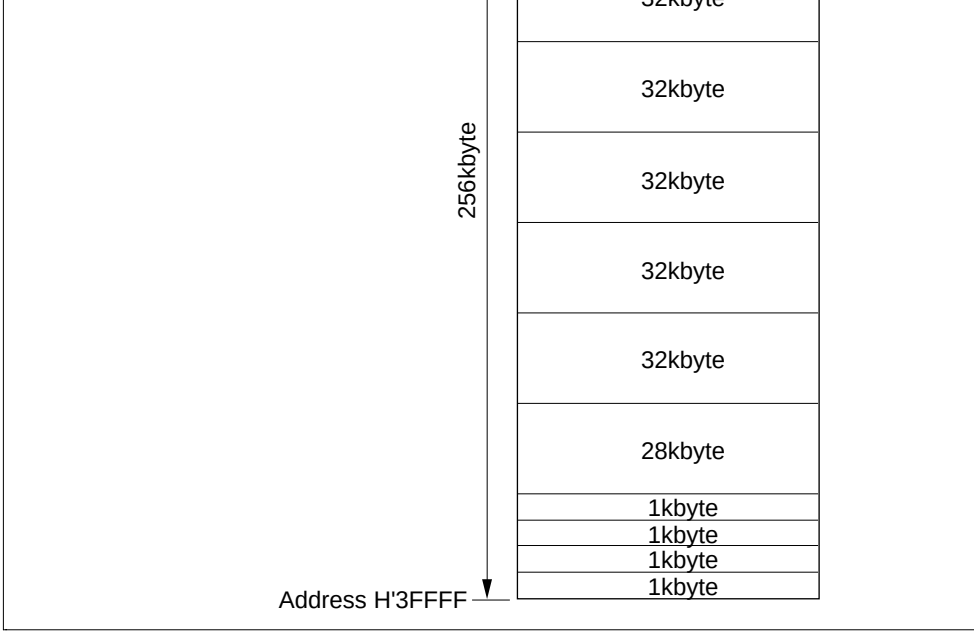


Figure 22.7 Block Configuration

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 sho
22.3.

Table 22.3 Register Configuration

Name	Abbre-viation	R/W	Initial Value	Address	Ac
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,

Notes: 1. FLMCR1, FLMCR2, EBR1, and EBR2 are 8-bit registers, and RAMER is a 16
register.

2. Only byte accesses are valid for FLMCR1, FLMCR2, EBR1, and EBR2, the a
requiring 3 cycles. Three cycles are required for a byte or word access to RA
6 cycles for a longword access.

3. When a longword write is performed on RAMER, 0 must always be written to
word (address H'FFFF8630). Operation is not guaranteed if any other value i

*1 In modes in which the on-chip flash memory is disabled, a read will return H'0
writes are invalid. Writes are also disabled when the FWE bit is set to 1 in FL

*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 SW
FLMCR1 is not set, these registers are initialized to H'00.

invalid.

Writes to bits SWE, ESU1, PSU1, EV1, and PV1 are enabled only when FWE = 1 and SW writes to the E1 bit only when FWE = 1, SWE = 1, and ESU1 = 1; and writes to the P1 bi when FWE = 1, SWE = 1, and PSU1 = 1.

Bit:	7	6	5	4	3	2	1
	FWE	SWE	ESU1	PSU1	EV1	PV1	E1
Initial value:	0	0	0	0	0	0	0
R/W:	R	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 h 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 should be set when setting bits 5–0, FLMCR2 bits 5–0, EBR1 bits 3–0, and EBR2 bits

Bit 6: SWE	Description
0	Writes disabled (Initial v
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 a same time.

	[Setting condition] When FWE=1 and SWE=1

- Bit 3—Erase-Verify 1 (EV1): Selects erase-verify mode transition or release (applica addresses: H'00000–H'1FFFF). Do not set the SWE, ESU1, PSU1, PV1, E1, or P1 bi 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 (addresses: H'00000–H'1FFFF). Do not set the SWE, ESU1, PSU1, EV1, E1, or P1 bi 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

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-ver or erase-verify mode for addresses H'20000–H'3FFFF is entered by setting SWE (FLMCI 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 setting the PSU2 bit, and finally setting the P2 bit. Erase mode for addresses H'20000–H' entered by setting SWE (FLMCR1) to 1 when FWE (FLMCR1) = 1, then setting the ESU and finally setting the E2 bit. FLMCR2 is initialized to H'00 by a power-on reset, in stand mode, when a high level is input to the FWP pin, and when a low level is input to the FW and the SWE bit in FLMCR1 is not set (the exception is the FLER bit, which is initialized a power-on reset). When on-chip flash memory is disabled, a read will return H'00, and w invalid.

Writes to bits ESU2, PSU2, EV2, and PV2 in FLMCR2 are enabled only when FWE (FL 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, (FLMCR1) = 1, and PSU2 = 1.

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

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 (appl addresses: H'20000–H'3FFFF). Do not set the ESU2, EV2, PV2, E2, or P2 bit at the

Bit 4: PSU2	Description
0	Program setup release (Initial value)
1	Program setup [Setting condition] When FWE=1 and SWE=1

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 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 addres H'20000–H'3FFFF). Do not set the ESU2, PSU2, EV2, PV2, or E2 bit at the same tim

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

	—	—	—	—	EB3	EB2	EB1
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	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. EBR initialized to H'00 by a power-on reset and standby mode, when a high level is input to t pin, and when a low level is input to the FWP pin and the SWE bit in FLMCR1 is not se bit in EBR2 is set to 1, the corresponding block can be erased. Other blocks are erase-pr 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
	EB11	EB10	EB9	EB8	EB7	EB6	EB5
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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 em
real-time flash memory programming. RAMER is initialized to H'0000 by a power-on res
not initialized in software standby mode. RAMER settings should be made in user mode
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 e
function, the ROM for which RAM emulation is performed should not be accessed imme
after this register has been modified. Normal execution of an access immediately after reg
modification is not guaranteed.

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

Bit:	7	6	5	4	3	2	1
	—	—	—	—	—	RAMS	RAM1
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W

- Bits 15–3—Reserved bits: These bits are always read as 0.

Table 22.5 Separation of the Flash Memory Area

Addresses	Block Name	RAMS	RAM1	RAM
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

	Expanded Mode	×2		0	1	0
	Single-chip Mode			0	1	0
	Expanded Mode	×4		1	0	0
	Single-chip Mode			1	0	0
User program mode	Expanded Mode	×1	0	0	0	1
	Single-chip Mode			0	0	1
	Expanded Mode	×2		0	1	1
	Single-chip Mode			0	1	1
	Expanded Mode	×4		1	0	1
	Single-chip Mode			1	0	1

The transferred programming control program must therefore include coding that follow
programming algorithm given later.

The system configuration in boot mode is shown in figure 22.8, and the boot mode exec
procedure in figure 22.9.

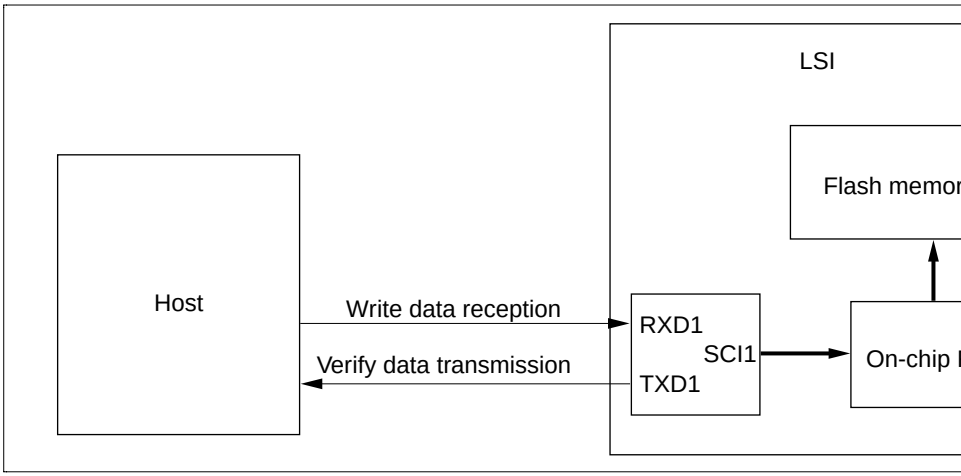
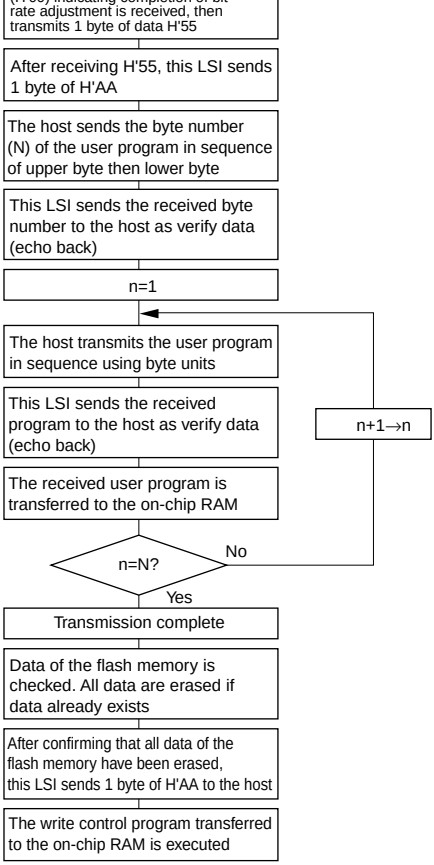


Figure 22.8 System Configuration in Boot Mode



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

communication data (H'00) transmitted continuously from the host. The SCI transmit/re
format should be set as follows: 8-bit data, 1 stop bit, no parity. The LSI calculates the b
the transmission from the host from the measured low period, and transmits one H'00 by
host to indicate the end of bit rate adjustment. The host should confirm that this adjustm
indication (H'00) has been received normally, and transmit one H'55 byte to the LSI. If r
cannot be performed normally, initiate boot mode again (reset), and repeat the above op
Depending on the host's transmission bit rate and the LSI's system clock frequency, the
a discrepancy between the bit rates of the host and the LSI. To ensure correct SCI opera
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 automat
adjustment of the LSI bit rate is possible. The boot program should be executed within t
clock range.

**Table 22.7 System Clock Frequencies for which Automatic Adjustment of LSI Bi
Possible**

Host Bit Rate	System Clock Frequency for which Automatic Adjustme of LSI Bit Rate is Possible
9,600 bps	8 to 28.7 MHz
4,800 bps	4 to 20 MHz

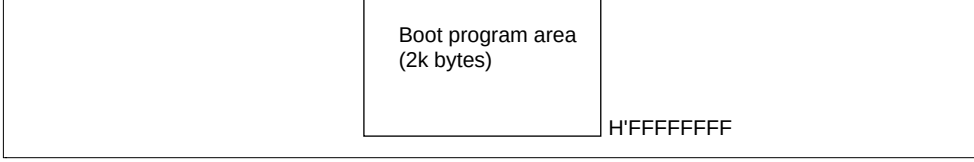


Figure 22.11 RAM Areas in Boot Mode

Note: The boot program area cannot be used until a transition is made to the execution s
the programming control program transferred to RAM. Note also that the boot pr
remains in this area of the on-chip RAM even after control branches to the progra
control program.

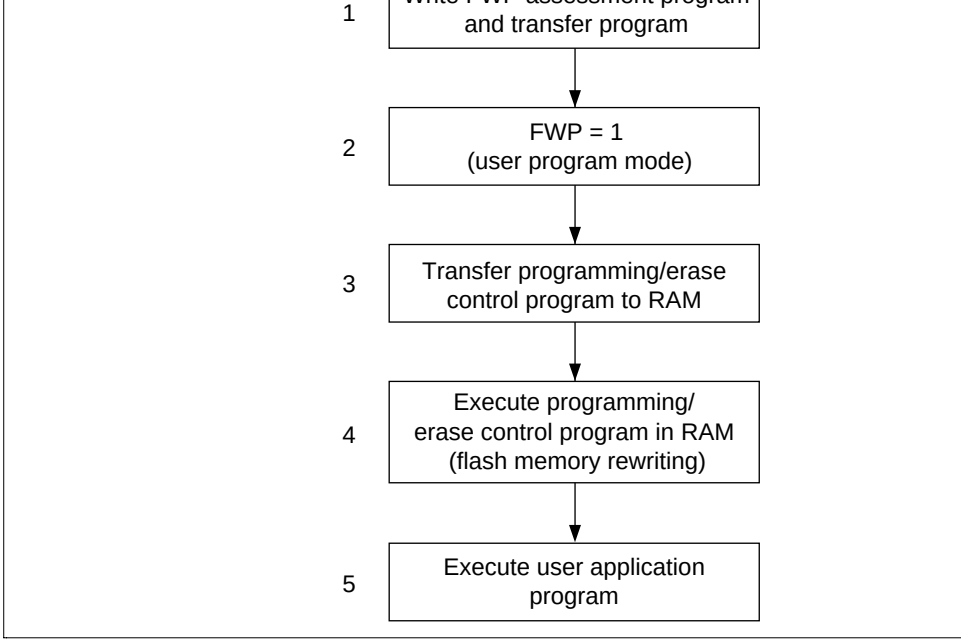


Figure 22.12 User Program Mode Execution Procedure

Notes: 1. When programming and erasing, start the watchdog timer so that measures c
taken to prevent program runaway, etc. Memory cells may not operate norma
overprogrammed or overerased due to program runaway.

2. If an address at which a flash memory register resides is read in the mask RO
ZTAT version, the value will be undefined. When a flash memory version p
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
memory rewrite program.

E1, and P1 bits in FLMCR1, or the ESU2, PSU2, EV2, PV2, E2, and P2 bits i
FLMCR2, is executed by a program in flash memory.

2. When programming or erasing, set FWP to low level (programming/erasing w
executed if FWP is set to high level).
3. Programming should be performed in the erased state. Do not perform additio
programming on previously programmed addresses.
4. Do not program addresses H'00000–H'1FFFF and H'20000–H'3FFFF simultan
Operation is not guaranteed if this is done.

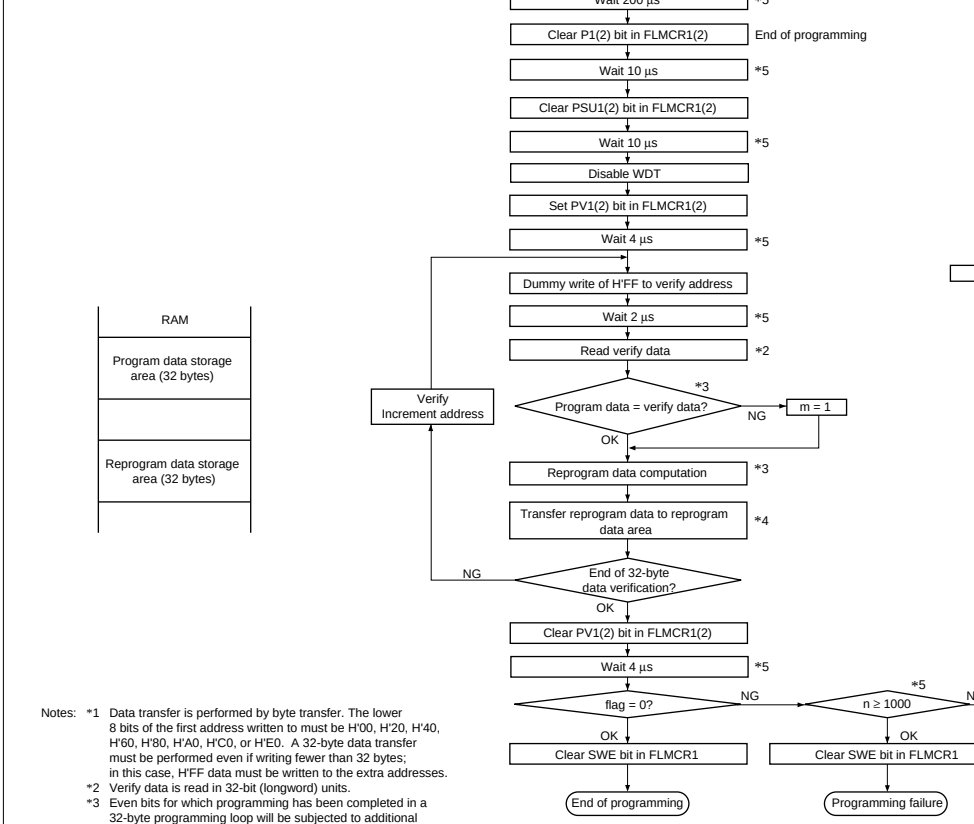
22.7.1 Program Mode (n = 1 for Addresses H'0000–H'1FFFF, n = 2 for Addresses H'20000–H'3FFFF)

When writing data or programs to flash memory, the program/program-verify flowchart s
figure 22.13 should be followed. Performing program operations according to this flowch
enable data or programs to be written to flash memory without subjecting the device to vo
stress or sacrificing program data reliability. Programming should be carried out 32 bytes
time.

Following the elapse of 10 µs or more after the SWE bit is set to 1 in flash memory contr
register 1 (FLMCR1), 32-byte program data is stored in the program data area and reprog
area, and the 32-byte data in the program data area in RAM is written consecutively to th
program address (the lower 8 bits of the first address written to must be H'00, H'20, H'40,
H'80, H'A0, H'C0, or H'E0). Thirty-two consecutive byte data transfers are performed. Th
program address and program data are latched in the flash memory. A 32-byte data transf
be performed even if writing fewer than 32 bytes; in this case, H'FF data must be written
extra addresses.

Next, the watchdog timer is set to prevent overprogramming in the event of program runa
Set a minimum value of 300 µs or more as the WDT overflow period. After this, prepara
program mode (program setup) is carried out by setting the PSUn bit in FLMCRn, and af
elapse of 50 µs or more, the operating mode is switched to program mode by setting the P

of H'FF data should be made to the addresses to be read. The dummy write should be ex
after the elapse of 4 µs or more. When the flash memory is read in this state (verify data
32-bit units), the data at the latched address is read. Wait at least 2 µs after the dummy w
before performing this read operation. Next, the written data is compared with the verify
reprogram data is computed (see figure 22.13) and transferred to the reprogram data area
bytes of data have been verified, exit program-verify mode, wait for at least 4 µs, then re
SWE bit in FLMCR1. If reprogramming is necessary, set program mode again, and repe
program/program-verify sequence as before. However, ensure that the program/program
sequence is not repeated more than 1,000 times on the same bits.



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; repr
1	0	1	
1	1	1	Still in erased state; no action

Figure 22.13 Program/Program Verify Flow

```
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    $
```

```
        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.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
;
```

```
        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 (Pdatal
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    $
```

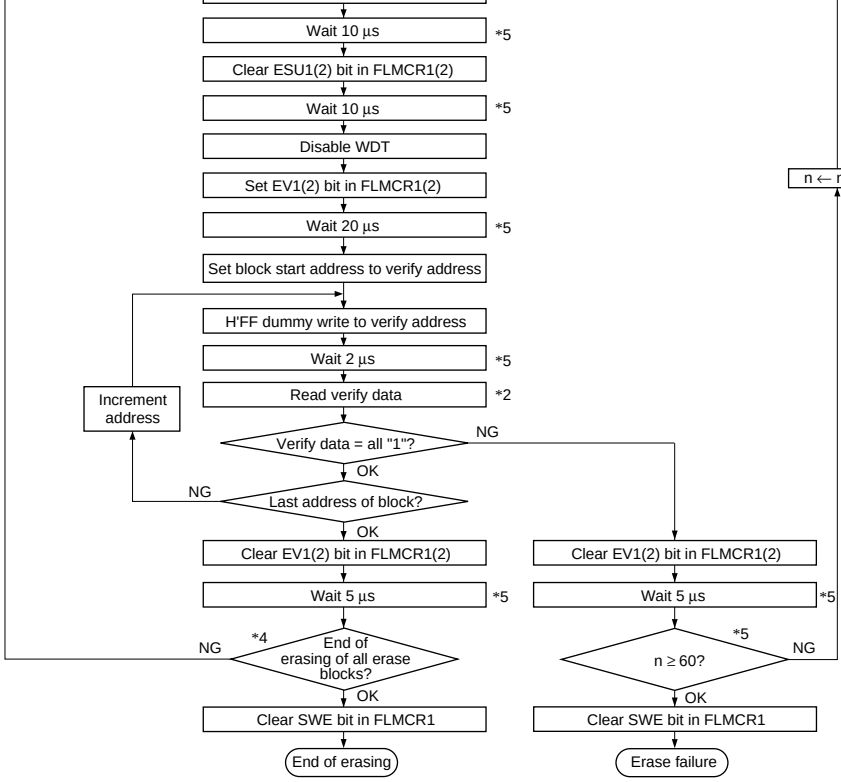
22.7.3 Erase Mode (n = 1 for Addresses H'0000–H'1FFFF, n = 2 for Addresses H'3FFFF)

When erasing flash memory, the erase/erase-verify flowchart shown in figure 22.14 sho followed.

To perform data or program erasure, set the flash memory area to be erased in erase bloc 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 progr runaway, etc. Set 5.3 µs as the WDT overflow period. After this, preparation for erase m setup) is carried out by setting the ESUn bit in FLMCRn, and after the elapse of 200 µs the operating mode is switched to erase mode by setting the En bit in FLMCRn. The tim 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 me be erased to all “0”) is not necessary before starting the erase procedure.

latched address is read. Wait at least 2 µs after the dummy write before performing this re
operation. If the read data has been erased (all “1”), a dummy write is performed to the ne
address, and erase-verify is performed. If the read data is unerased, set erase mode again a
repeat the erase/erase-verify sequence in the same way. However, ensure that the erase/e
verify sequence is not repeated more than 60 times. When verification is completed, exit
verify mode, and wait for at least 5 µs. If erasure has been completed on all the erase bloc
completing erase-verify operations on all these blocks, release the SWE bit in FLMCR1.
are any unerased blocks, set erase mode again, and repeat the erase/erase-verify sequence
before. However, ensure that the erase/erase-verify sequence is not repeated more than 60



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)

```
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.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 ->
        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
```

```
        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 ->
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 blo
        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
;
```

```
        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
```

Table 22.8 Hardware Protection

Item	Description	Func Program
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
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

Table 22.9 Software Protection

Item	Description	Fur Program
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
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.	—
Emulation protection	• Setting the RAMS bit to 1 in the RAM emulation register (RAMER) places all blocks in the program/erase-protected state.	Yes

FLER bit setting conditions are as follows:

1. When flash memory is read during programming/erasing (including a vector read or i 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.

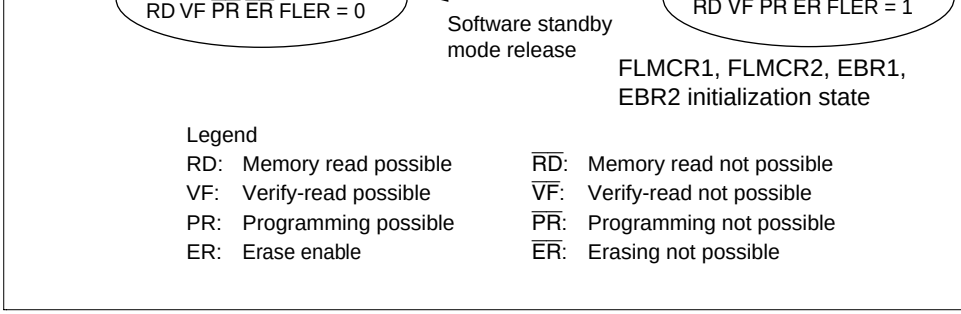


Figure 22.15 Flash Memory State Transitions

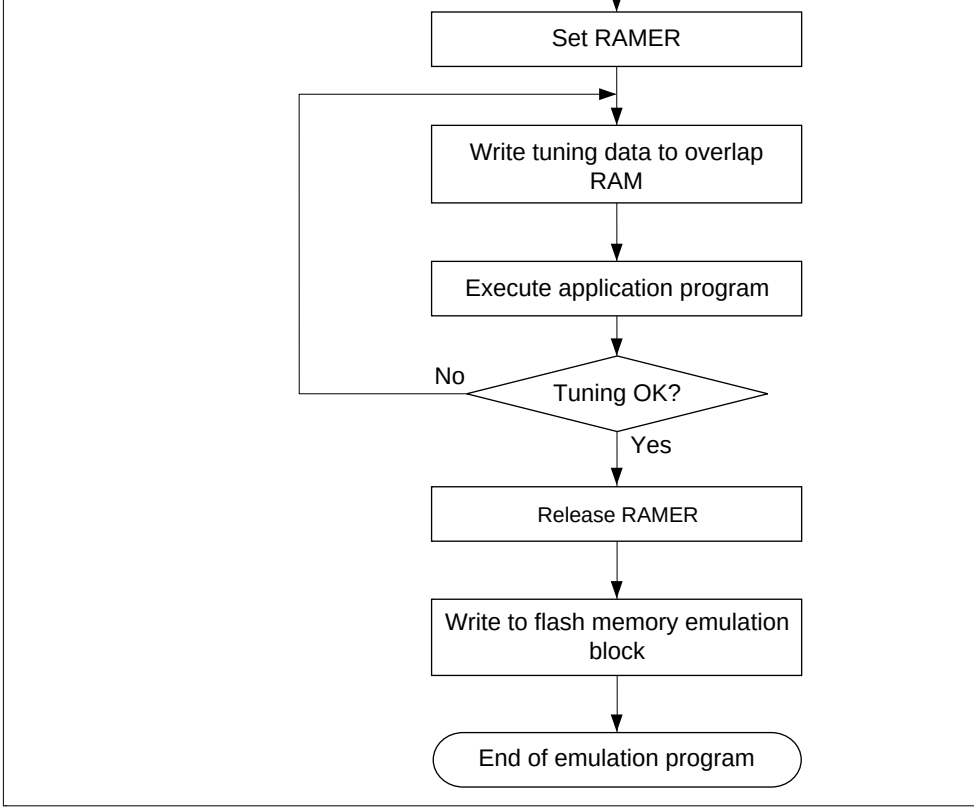


Figure 22.16 Flowchart for Flash Memory Emulation in RAM

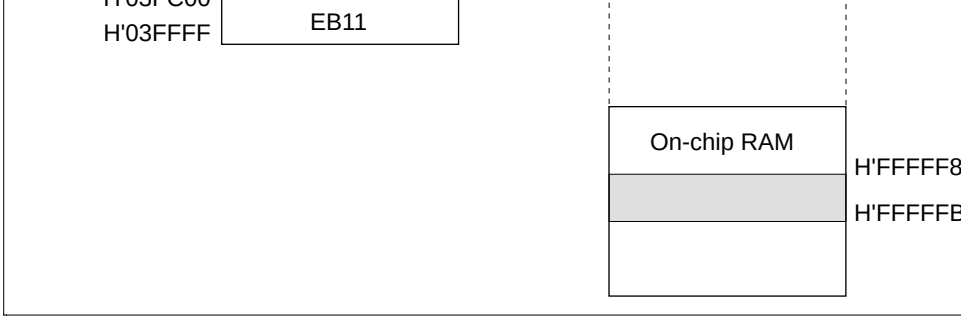


Figure 22.17 Example of RAM Overlap Operation

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 or
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
4. The data written in the overlapping RAM is written into the flash memory space (EB

Notes: 1. When the RAMS bit is set to 1, program/erase protection is enabled for all bl
regardless of the value of RAM1 and RAM0 (emulation protection). In this s
setting the P1 or E1 bit in flash memory control register 1 (FLMCR1), or the
bit in flash memory control register 2 (FLMCR2), will not cause a transition
program mode or erase mode. When actually programming or erasing a flash
area, the RAMS bit should be cleared to 0.

2. A RAM area cannot be erased by execution of software in accordance with th
algorithm while flash memory emulation in RAM is being used.

In programmer mode, set the mode pins to PLL x2 mode (see table 22.10) and use a 6 MH clock. The LSI will then operate at 12 MHz.

Table 22.10 shows the pin settings for programmer mode. For the pin names in programm 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 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 th (flash write enable) pin.

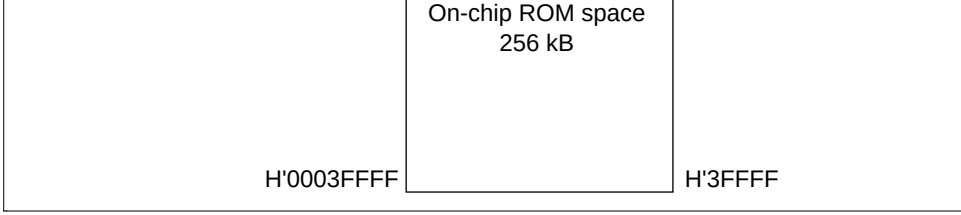
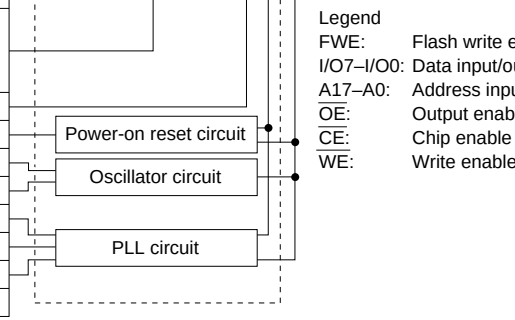


Figure 22.18 On-Chip ROM Memory Map

4	A0
5	A1
6	A2
7	A3
8	A4
9	A5
10	A6
11	A7
12	A8
43	$\overline{OE}$
14	A10
15	A11
16	A12
17	A13
18	A14
42	$\overline{CE}$
21, 37, 46, 49, 50, 65, 73, 75, 76, 79, 100, 103	V_{CC}
3, 23, 27, 33, 39, 55, 61, 71 78, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 101, 109	V_{SS}
26	A17
84	$\overline{RES}$
72	XTAL
74	EXTAL
80	$PLLV_{CC}$
81	PLLCAP
82	$PLLV_{SS}$
Other than the above	NC(OPEN)

12	A0
11	A1
10	A2
9	A3
8	A4
7	A5
6	A6
5	A7
27	A8
24	$\overline{OE}$
23	A10
25	A11
4	A12
28	A13
29	A14
22	$\overline{CE}$
32	V_{CC}
16	V_{SS}
30	A17



Legend
FWE: Flash write e
I/O7–I/O0: Data input/o
A17–A0: Address inpu
$\overline{OE}$: Output enab
$\overline{CE}$: Chip enable
$\overline{WE}$: Write enable

Figure 22.19 Socket Adapter Pin Correspondence Diagram (SH7044)

7	A0
8	A1
9	A2
10	A3
11	A4
13	A5
15	A6
16	A7
17	A8
52	$\overline{OE}$
19	A10
20	A11
21	A12
22	A13
23	A14
51	$\overline{CE}$
12, 26, 40, 63, 77, 85, 95, 97, 98, 103, 112, 127, 128, 131, 132, 135, 136	V_{CC}
6, 14, 28, 35, 42, 55, 61, 71, 79, 87, 93, 102, 117 to 126, 129, 141	V_{SS}
34	A17
108	$\overline{RES}$
94	XTAL
96	EXTAL
104	$PLLV_{CC}$
105	PLLCAP
106	$PLLV_{SS}$
Other than the above	NC(OPEN)

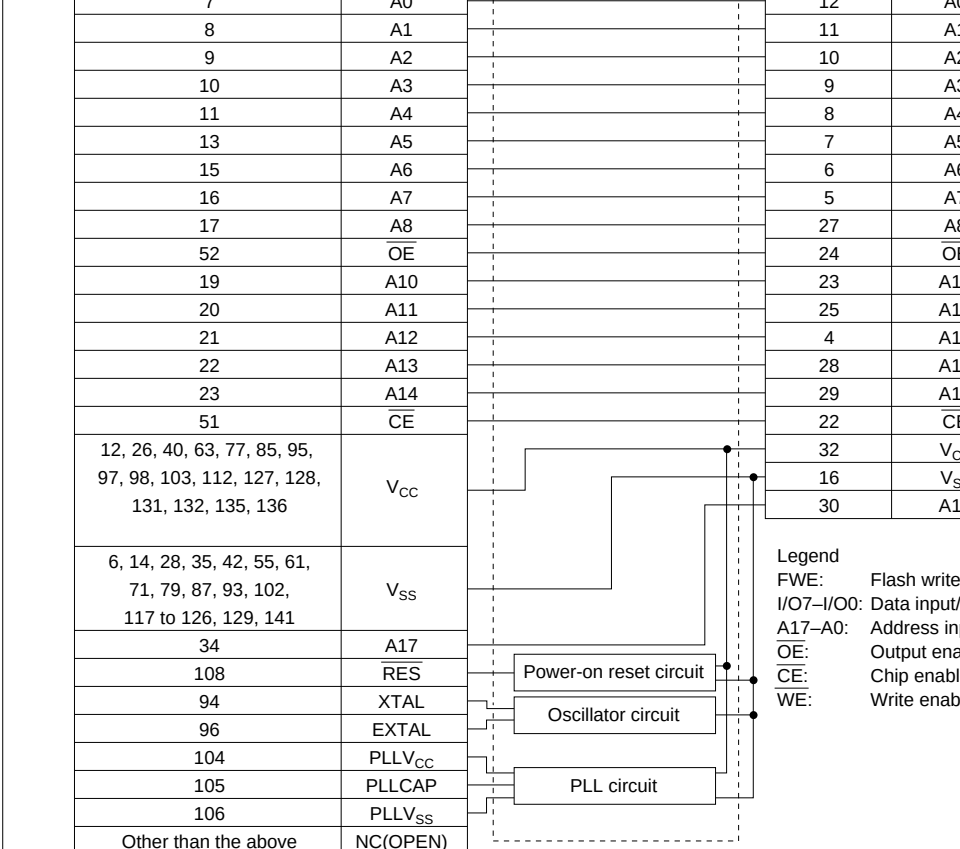


Figure 22.20 Socket Adapter Pin Correspondence Diagram (SH7045)

to confirm the end of auto-programming.

- Status Read Mode

  Status polling is used for auto-programming and auto-erasing, and normal termination
  confirmed by reading the I/O6 signal. In status read mode, error information is output
  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**
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	*Ai
Chip disable	H or L	H	X	X	Hi-z	Ain

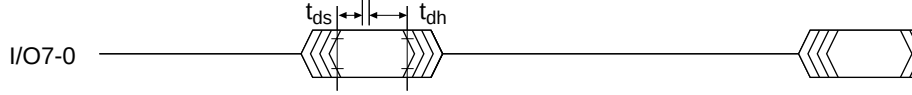
Notes: *Ain indicates that there is also address input in auto-program mode.

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
   FWE pin.

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	

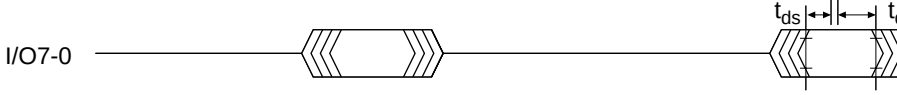


Note: Data is latched on the rising edge of $\overline{WE}$.

Figure 22.21 Timing Waveforms for Memory Read after Memory Write

Table 22.14 AC Characteristics in Transition from Memory Read Mode to Another (Conditions: V_{CC} = 5.0 V ±10%, V_{SS} = 0 V, T_a = 25°C ±5°C)

Item	Symbol	Min	Max	Unit	Note
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 Anot

Table 22.15 AC Characteristics in Memory Read Mode (Conditions: V_{CC} = 5.0 V ± V_{SS} = 0 V, T_a = 25°C ±5°C)

Item	Symbol	Min	Max	Unit	No
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

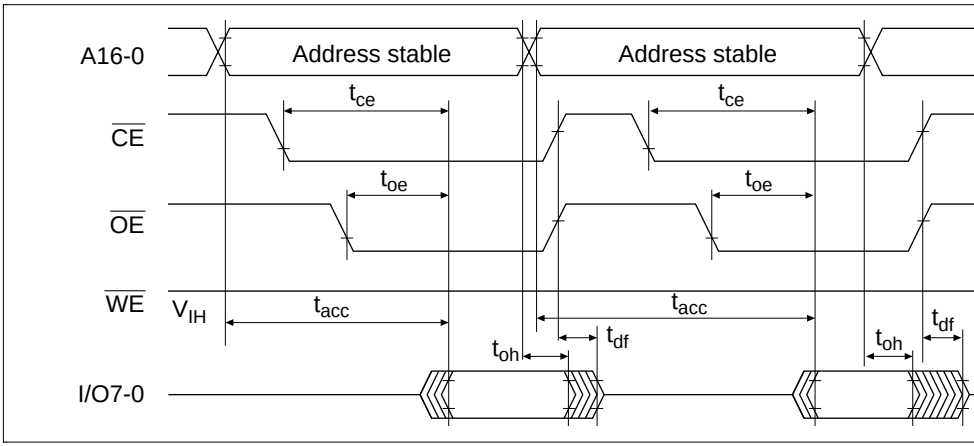


Figure 22.24 $\overline{CE}$ and $\overline{OE}$ Clock System Read Timing Waveforms

6. Perform one auto-program operation for a 128-byte block for each address. Characte not guaranteed for two or more additional programming operations.
7. Confirm normal end of auto-programming by checking I/O6. Alternatively, status re can also be used for this purpose (I/O7 status output uses the auto-program operation identification pin).
8. Status polling I/O6 and I/O7 pin information is retained until the next command writ as the next command write has not been performed, reading is possible by enabling $\overline{OE}$.

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

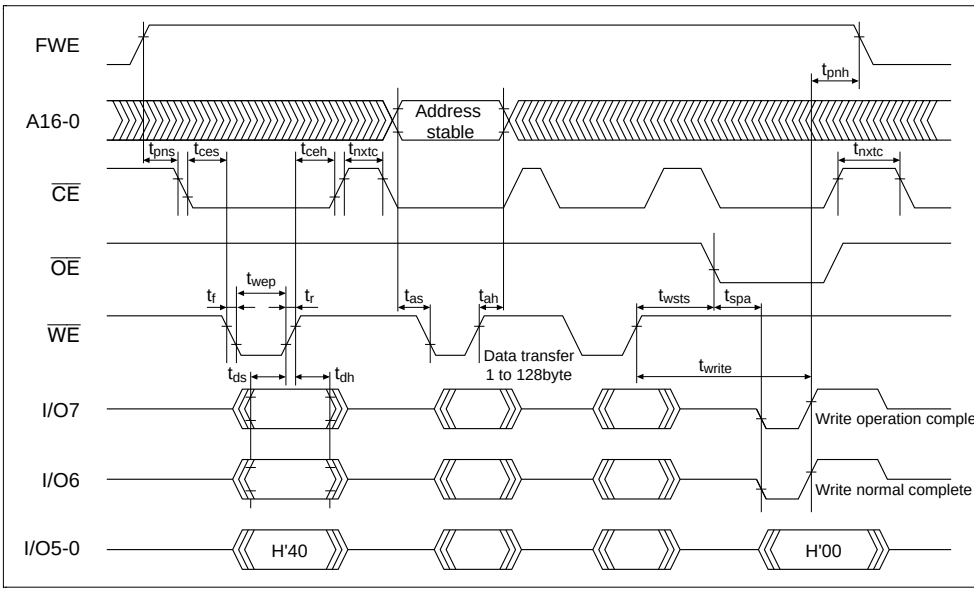


Figure 22.25 Auto-Program Mode Timing Waveforms

(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	

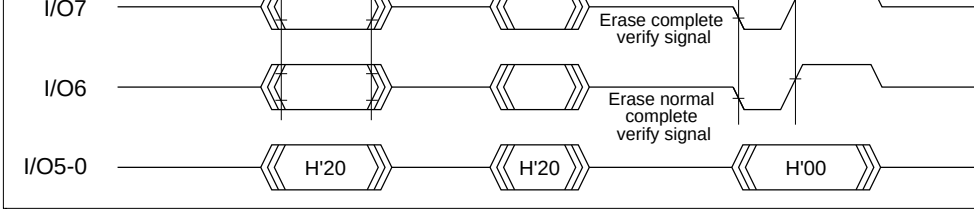


Figure 22.26 Auto-Erase Mode Timing Waveforms

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
Attribute	Normal end identification	Command error	Program-ming error	Erase error	—	—	Program-ming or erase count exceeded
Initial value	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

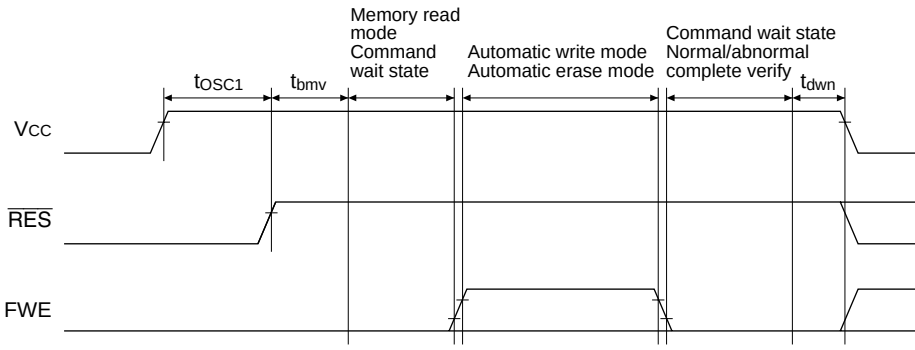
Note: D2 and D3 are undefined at present.

22.11.7 Status Polling

1. I/O7 status polling is a flag that indicates the operating status in auto-program/auto-e mode.
2. I/O6 status polling is a flag that indicates a normal or abnormal end in auto-program/ mode.

Table 22.21 Stipulated Transition Times to Command Wait State

Item	Symbol	Min	Max	Unit	Not
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 Tim

Additional programming cannot be performed on previously programmed ad
blocks.

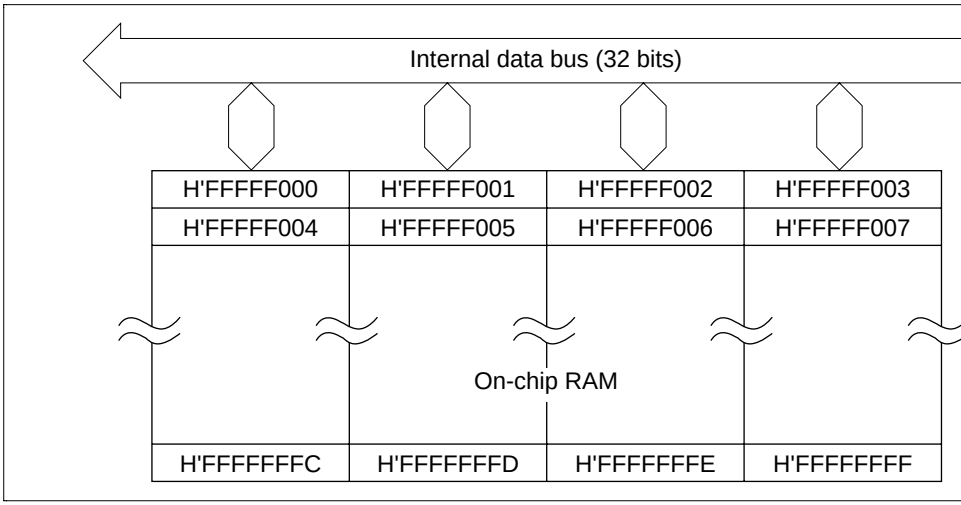


Figure 23.1 Block Diagram of RAM

23.2 Operation

The on-chip RAM is accessed by accessing addresses H'FFFFF000–H'FFFFFFFF. On-c
is also used as cache memory. There are 2 kbytes of on-chip RAM space during cache u
section 9, Cache Memory (CAC), for details.

- Standby mode

Table 24.1 describes the transition conditions for entering the modes from the program e state as well as the CPU and peripheral function status in each mode and the procedures 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**	**Cancel Proced**
Sleep	Execute SLEEP instruction with SBY bit set to 0 in SBYCR	Run	Halt	Run	Held	Held	Held	• Inte • DMA addr • Pow rese • Man
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 • Pow rese • Man

Notes: SBYCR: standby control register. SBY: standby bit

*1 Some bits within on-chip peripheral module registers are initialized by the sta mode; some are not. Refer to table 24.3, Register States in the Standby Mod section 24.4.1, Transition to Standby Mode. Also refer to the register descript each peripheral module.

*2 The status of the I/O port in standby mode is set by the port high impedance the SBYCR. Refer to section 24.2, Standby Control Register (SBYCR). For pi other than for the I/O port, refer to Appendix C, Pin Status.

standby mode, and the port status in standby mode. The SBYCR is initialized to H'1F wh

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

- Bit 7—Standby (SBY): Specifies transition to the standby mode. The SBY bit cannot 1 while the watchdog timer is running (when the timer enable bit (TME) of the WDT control/status register (TCSR) is set to 1). To enter the standby mode, always halt the 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 se port pin to high impedance or hold the pin status. The HIZ bit cannot be set to 1 when bit of the WDT timer control/status register (TCSR) is set to 1. When making the I/O 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 1. Always write 1 to these bits.

reset.

Cancellation by an Interrupt: When an interrupt occurs, the sleep mode is canceled an exception processing is executed. The sleep mode is not canceled if the interrupt cannot accepted because its priority level is equal to or less than the mask level set in the CPU's register (SR) or if an interrupt by an on-chip peripheral module is disabled at the periphe module.

Cancellation by a DMAC/DTC Address Error: If a DMAC/DTC address error occurs 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 $\overline{\text{RES}}$ p level cancels the sleep mode.

Cancellation by a Manual Reset: When the $\overline{\text{MRES}}$ pin is set to low level while the $\overline{\text{RE}}$ 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 ins The LSI moves from the program execution state to the standby mode. In the standby m power consumption is greatly reduced by halting not only the CPU, but the clock and on peripheral modules as well. CPU register contents and on-chip RAM data are held as lon prescribed voltages are applied. The register contents of some on-chip peripheral modul initialized, but some are not (table 24.3). The I/O port status can be selected as held or h impedance by the port high impedance bit (HIZ) of the SBYCR. For pin status other tha I/O port, refer to Appendix C, Pin States.

Direct memory access controller (DMAC)	• DMA channel control registers 0–3 (CHCR0–CHCR3) • DMA operation register (DMAOR)	—	• DMA so address 0–3 (SA SAR3) • DMA de address 0–3 (DA DAR3) • DMA tra count re 0–3 (DM DMATC
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	—	—

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 (se
the NMI edge select bit (NMIE) of the interrupt control register (ICR) of the INTC) is d
the NMI signal. This clock is supplied only to the watchdog timer (WDT). A WDT over
occurs if the time established by the clock select bits (CKS2–CKS0) in the TCSR of the
elapses before transition to the standby mode. The occurrence of this overflow is used to
that the clock has stabilized, so the clock is supplied to the entire chip, the standby mode
canceled, and NMI exception processing begins.

When canceling standby mode with NMI interrupts, set the CKS2–CKS0 bits so that the
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 NM
upon entering standby (when the clock is halted) is high level, and that the NMI pin leve
returning from standby (when the clock starts after oscillation stabilization) is low level.
canceling standby mode with an NMI pin set for rising edge, be sure that the NMI pin le
entering standby (when the clock is halted) is low level, and that the NMI pin level upon
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
cancels the standby mode.

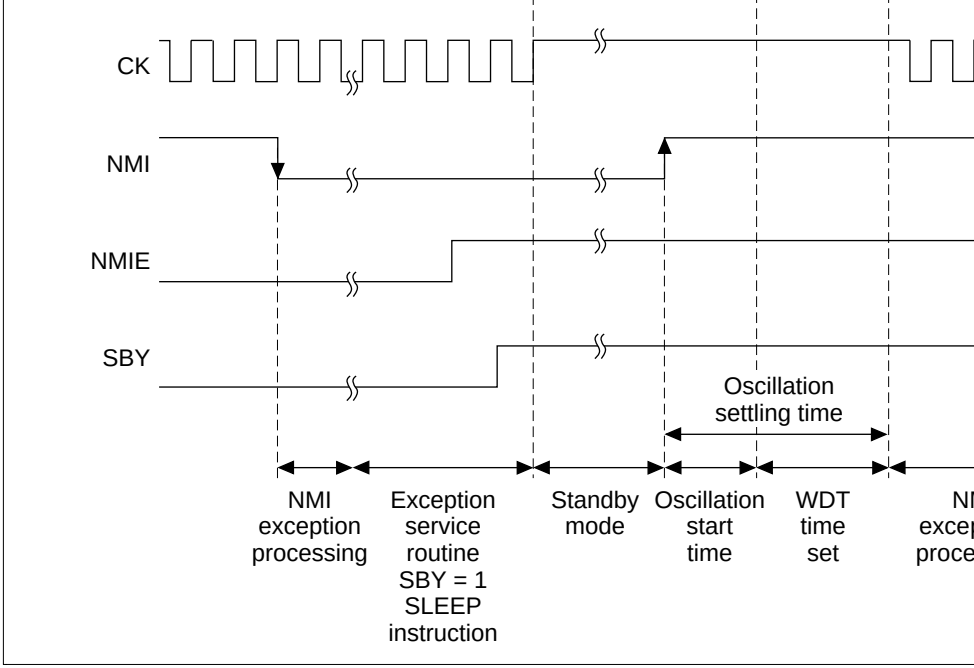


Figure 24.1 Standby Mode NMI Timing (Application Example)

Programmable voltage (ZTAT version only)	V_{PP}	–0.3 to +13.5	
Input voltage (other than A/D ports)	V_{in}	–0.3 to V_{CC} + 0.3	
Input voltage (A/D ports)	V_{in}	–0.3 to AV_{CC} + 0.3	
Analog supply voltage	AV_{CC}	–0.3 to +7.0	
Analog reference voltage (QFP-144 only)	AV_{ref}	–0.3 to AV_{CC} + 0.3	
Analog input voltage	V_{AN}	–0.3 to AV_{CC} + 0.3	
Operating temperature	T_{opr}	–20 to +75*1	
Programming temperature (ZTAT version only)	T_{we}	–20 to +75*2	
Storage temperature	T_{stg}	–55 to +125	

Notes: Operating the LSI in excess of the absolute maximum ratings may result in perm 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.

	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	$VT^+ \geq V_{CC} - 0.7$
							$VT^- \leq 0.5$ V (ma
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}
	A/D port		—	—	1.0	µA	Vin = 0.5 to AV
	Other input pins (except EXTAL pin)		—	—	1.0	µA	Vin = 0.5 to V_{CC}
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}

capacitance	NMI		—	—	50	pF	Ta = 25°C
	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 versio
RAM standby voltage		V_{RAM}	2.0	—	—	V	

Notes: 1. When the A/D converter is not used (including during standby), do not releas
AV_{SS}, and AV_{ref} (SH7041,SH7043,SH7045 only) pins. Connect the AV_{CC} and
(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
all output pins unloaded.

3. The ZTAT and mask versions as well as F-ZTAT and mask versions have th
functions, and the electrical characteristics of both are within specification, bu
characteristic-related performance values, operating margins, noise margins,
emission, etc., are different. Caution is therefore required in carrying out syst
when switching between ZTAT and mask versions, and when switching betw
ZTAT and mask versions.

4. When the SH7040 chip is used for high-speed operation, the package surfac
temperature rises. Appropriate measures (such as heat dissipation) to ensur
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.

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 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	Fig
Operating frequency	f_{OP}	4	28.7	MHz	25.
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.
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.
Standby return clock settling time	t_{OSC2}	10	—	ms	

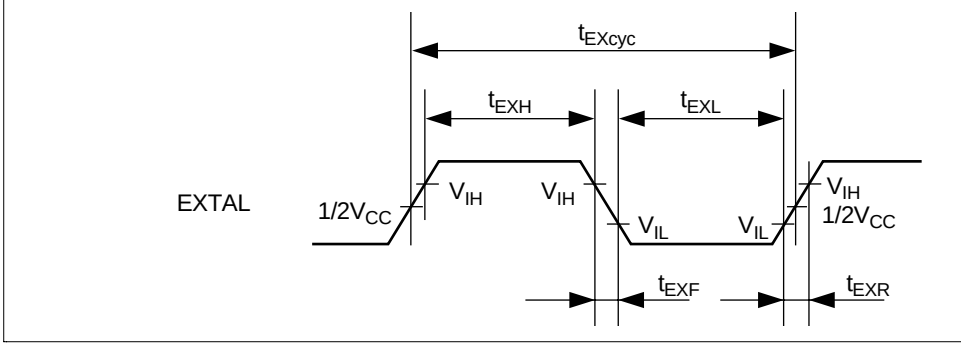


Figure 25.2 EXTAL Clock Input Timing

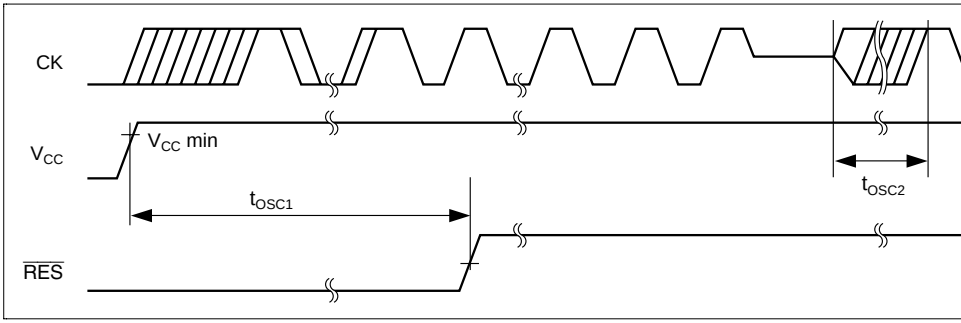


Figure 25.3 Oscillation Settling Time

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	2
$\overline{IRQ7}$–$\overline{IRQ0}$ hold time	t_{IRQEH}	35	—	ns	
$\overline{IRQOUT}$ output delay time	t_{IRQOD}	—	35	ns	2
Bus request setup time	t_{BRQS}	35	—	ns	2
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 when thesetup times shown here are provided, the signals are considered to hav produced changes at clock rise (for $\overline{RES}$, $\overline{MRES}$, $\overline{BREQ}$) or clock fall (for NMI an $\overline{IRQ0}$). If the setup times are not provided, recognition is delayed until the next cl or fall.

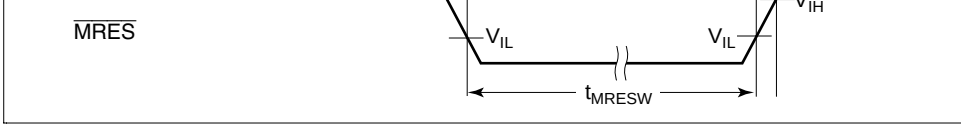


Figure 25.4 Reset Input Timing

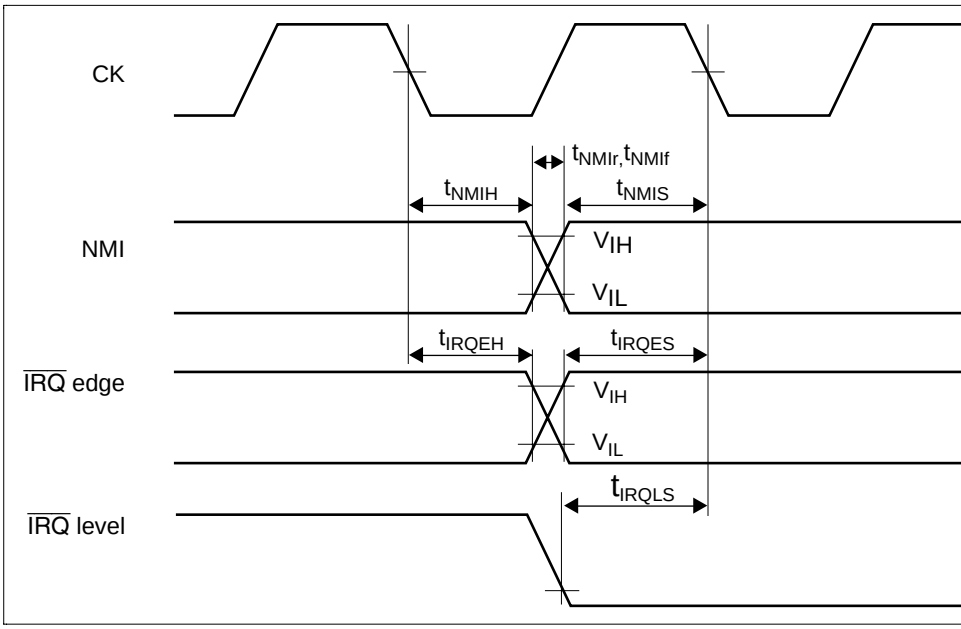
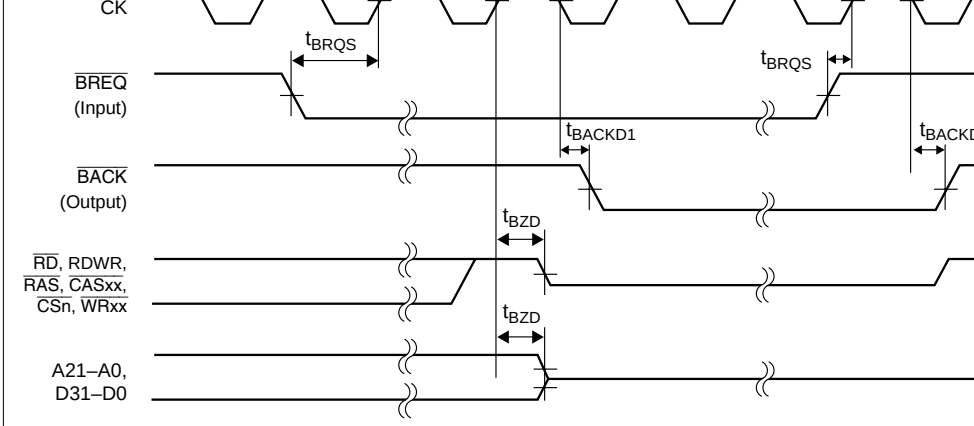


Figure 25.5 Interrupt Signal Input Timing



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

Read strobe delay time 2	t_{RSD2}	2*3	18	ns	25.19
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, 2
$\overline{WAIT}$ hold time	t_{WTH}	0	—	ns	25.19
$\overline{RAS}$ delay time 1	t_{RASD1}	2*3	18	ns	25.11–2
$\overline{RAS}$ 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
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	25.11–2
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	

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, 2 25.16, 25.19

Notes: n is the number of waits. m is 0 when the number of DRAM write cycle waits is 0, a otherwise. RCD is the set value of the RCD bit in DCR. TPC is the set value of the in DCR.

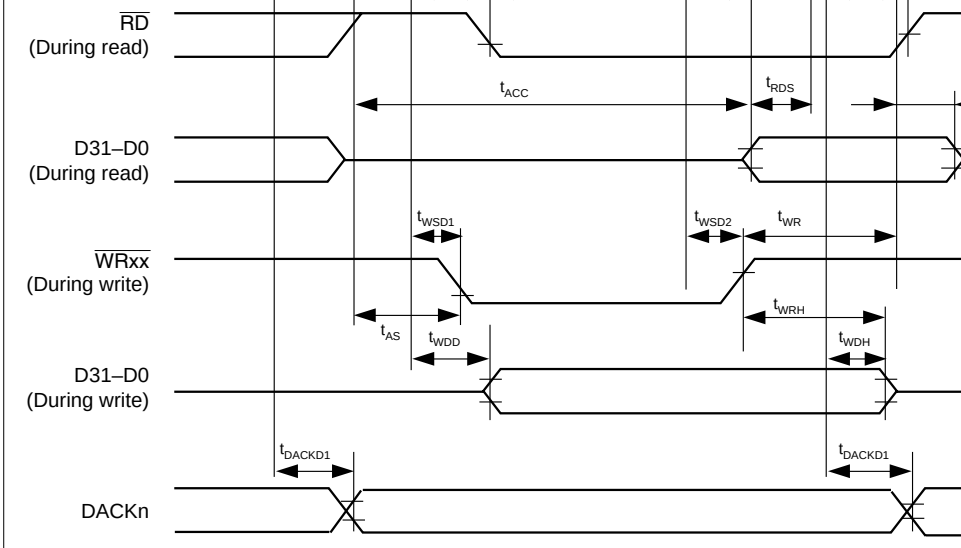
*1 If the access time is satisfied, t_{RDS} need not be satisfied.

*2 t_{WDH} (max) is a reference value.

*3 The delay time Min values are reference values (typ).

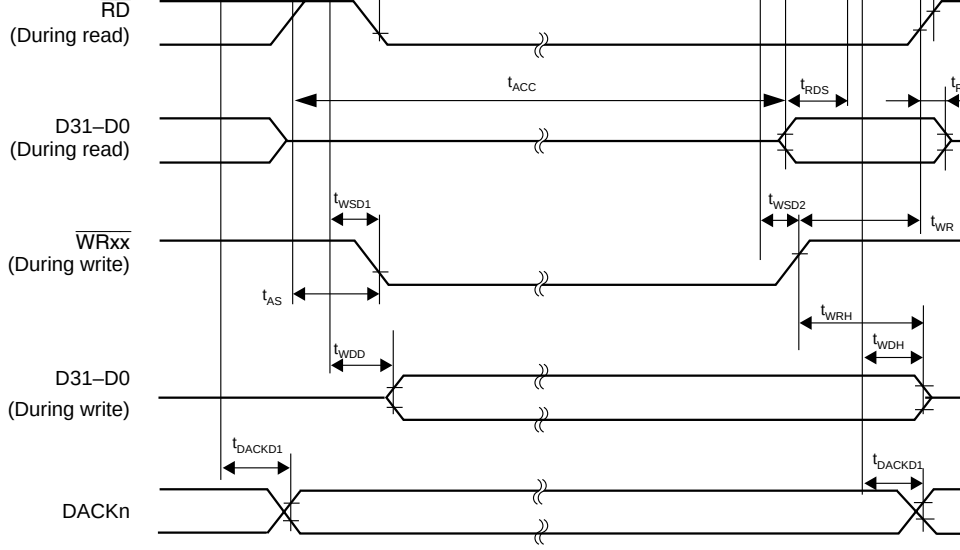
*4 t_{RDS} is a reference value.

*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)

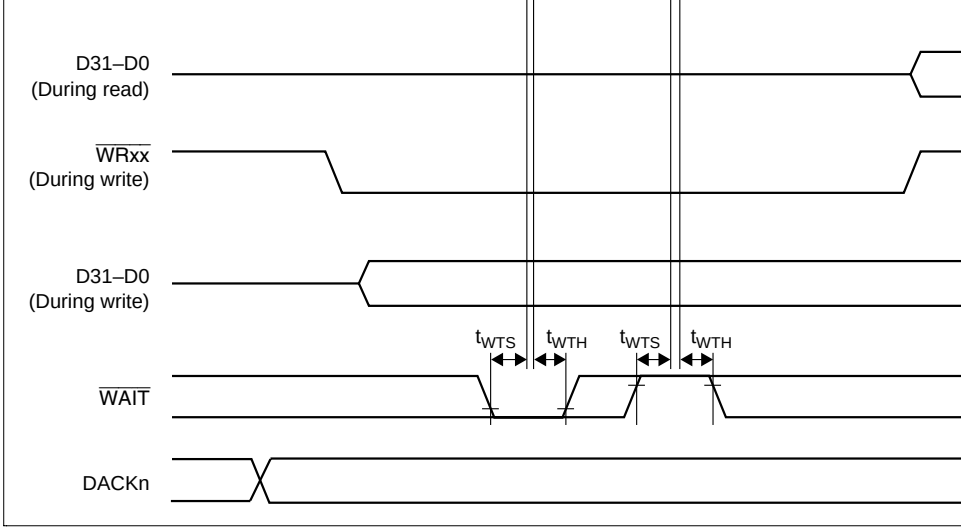
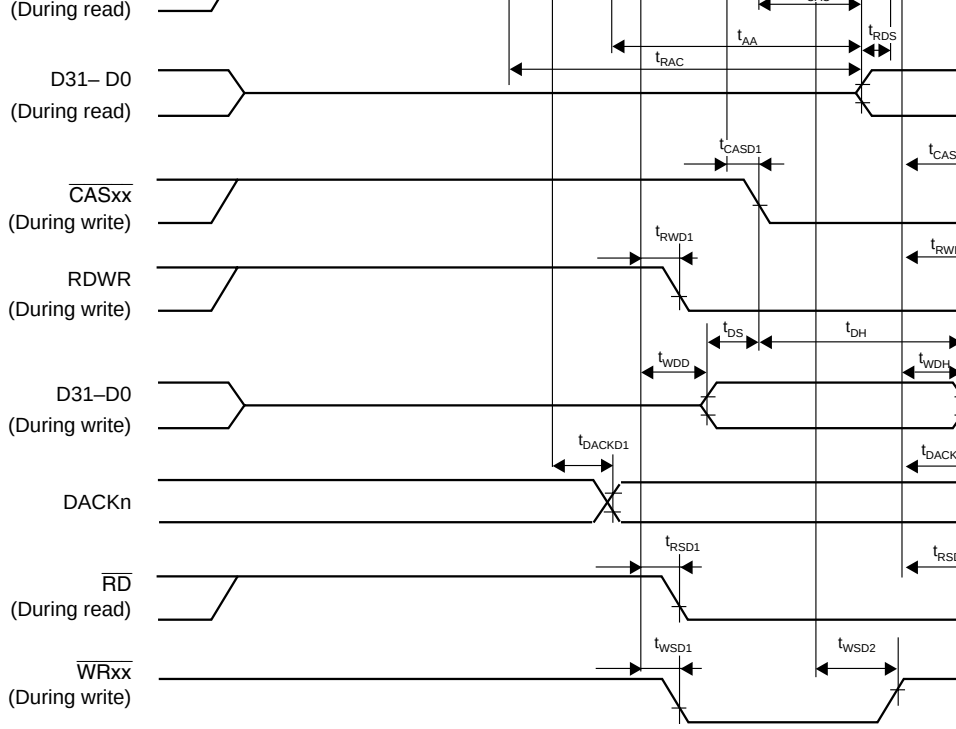
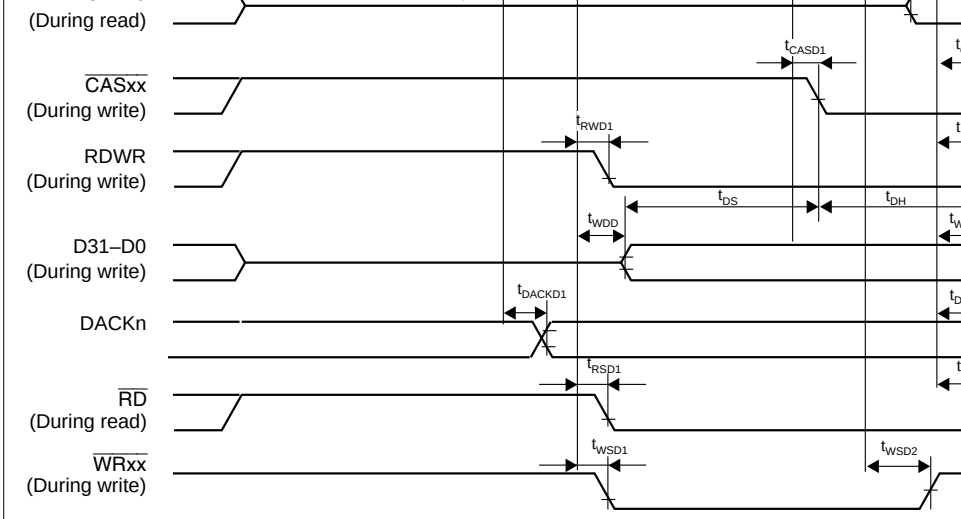


Figure 25.10 Basic Cycle (2 Software Waits + Wait due to $\overline{\text{WAIT}}$ Signal)



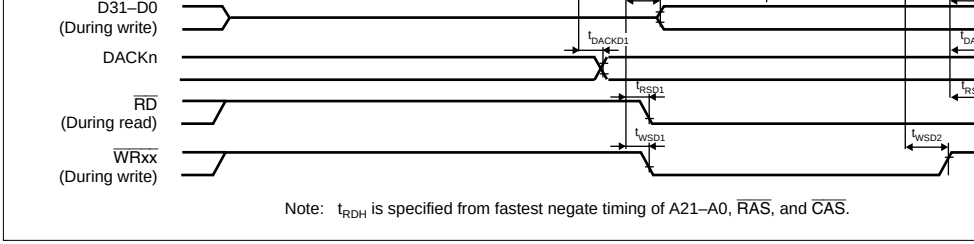
Note: t_{RDH} is specified from fastest negate timing of A21–A0, $\overline{RAS}$, and $\overline{CAS}$.

Figure 25.11 DRAM Cycle (Normal Mode, No Waits, TPC = 0, RCD = 0)



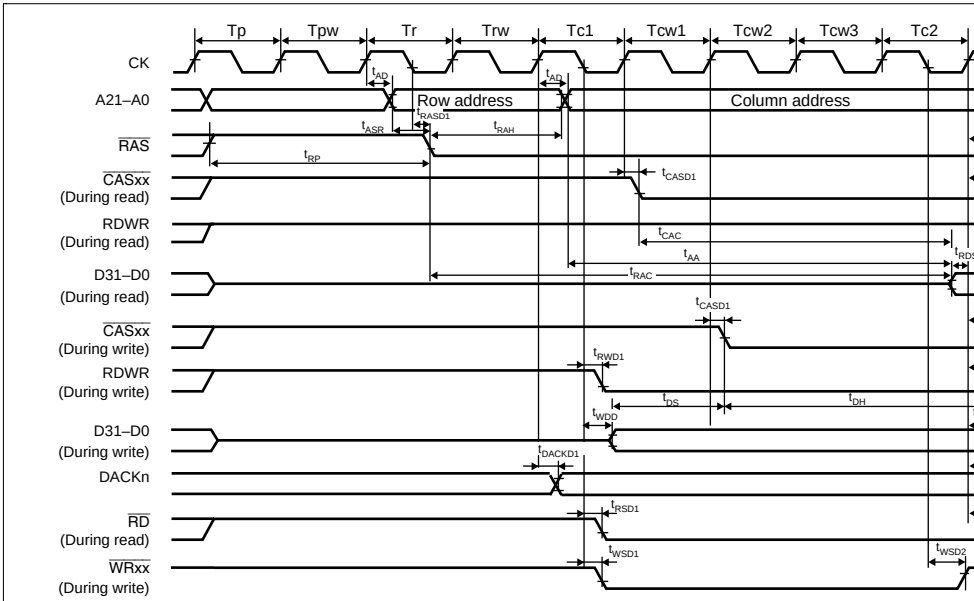
Note: t_{RDH} is specified from fastest negate timing of A21–A0, $\overline{RAS}$, and $\overline{CAS}$.

Figure 25.12 DRAM Cycle (Normal Mode, 1 Wait, TPC = 0, RCD = 0)



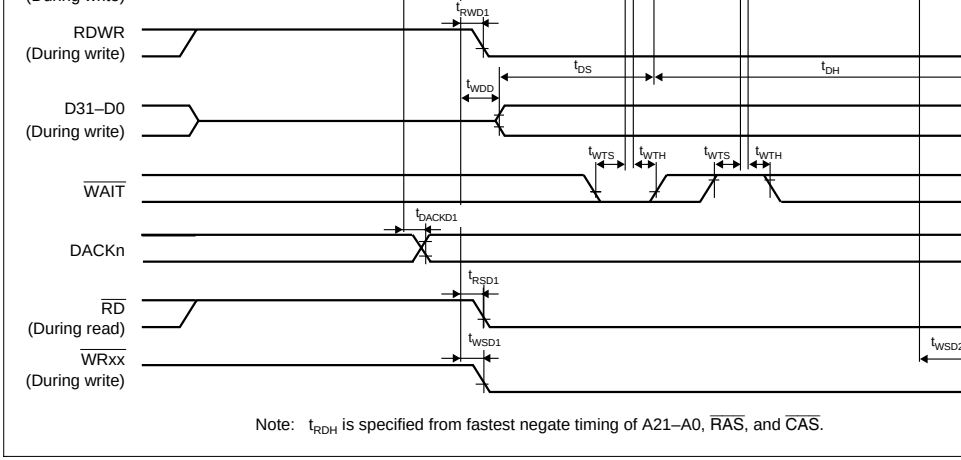
Note: t_{RDH} is specified from fastest negate timing of A21–A0, $\overline{RAS}$, and $\overline{CAS}$.

Figure 25.13 DRAM Cycle (Normal Mode, 2 Waits, TPC = 1, RCD = 1)



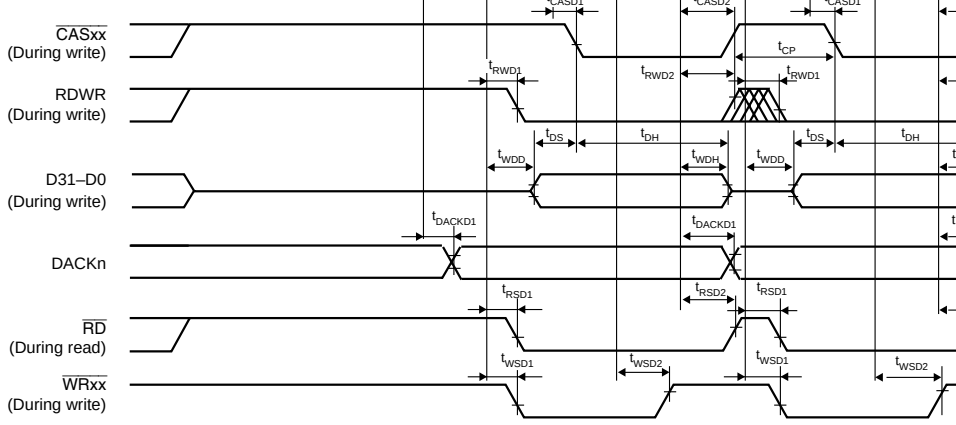
Note: t_{RDH} is specified from fastest negate timing of A21–A0, $\overline{RAS}$, and $\overline{CAS}$.

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 $\overline{WAIT}$ Sig



Note: t_{RDH} is specified from fastest negate timing of A21–A0, $\overline{RAS}$, and $\overline{CAS}$.

Figure 25.16 DRAM Cycle (High-Speed Page Mode)

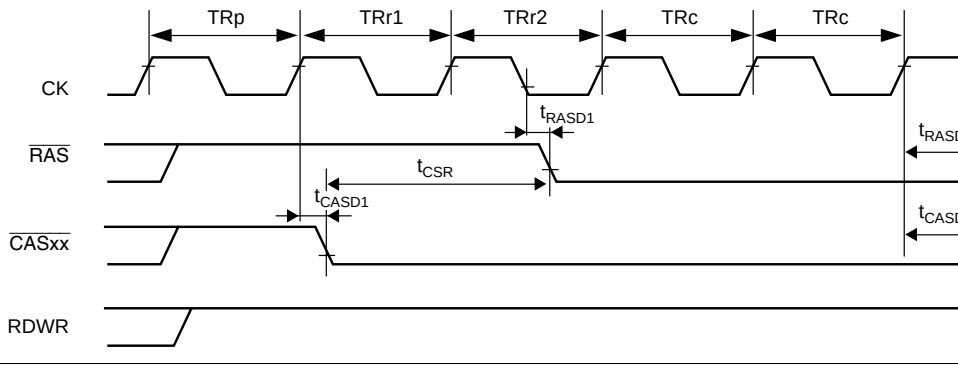
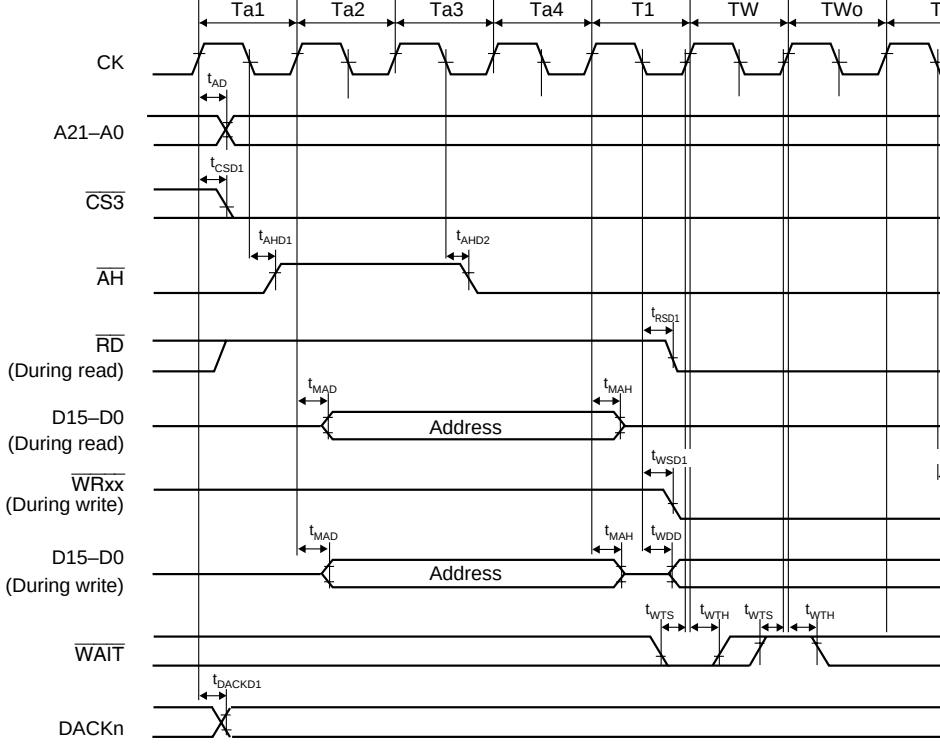


Figure 25.17 CAS Before RAS Refresh (TRAS1 = 0, TRAS0 = 0)



Note: t_{RDH} is specified from fastest negate timing of A21–A0, $\overline{CS3}$, and $\overline{R}$

Figure 25.19 Address Data Multiplex I/O Space Cycle (1 Software Wait + Extern

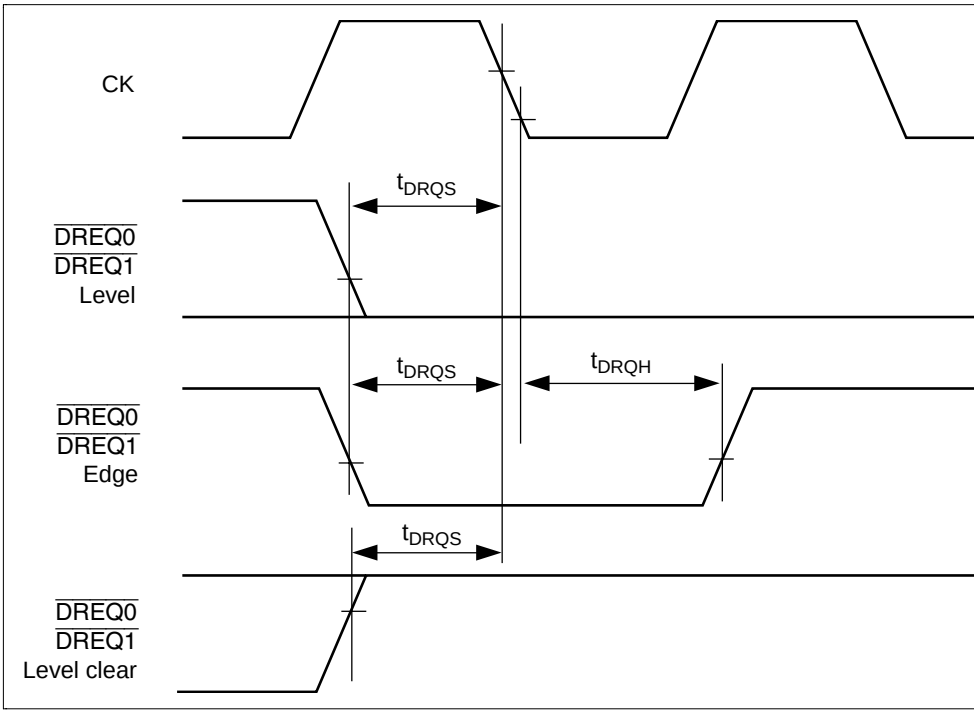


Figure 25.20 $\overline{\text{DREQ0}}$ and $\overline{\text{DREQ1}}$ Input Timing (1)

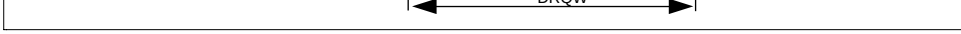


Figure 25.21 $\overline{\text{DREQ0}}$ and $\overline{\text{DREQ1}}$ Input Timing (2)

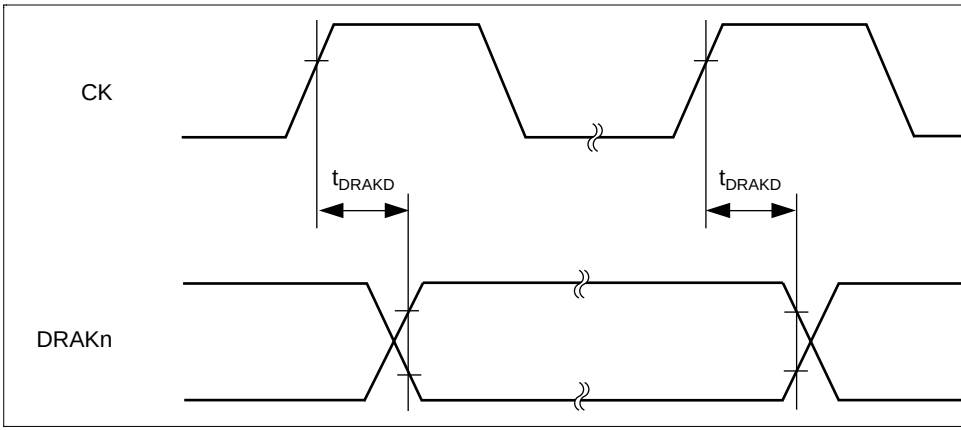


Figure 25.22 DRAK Output Delay Time

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}

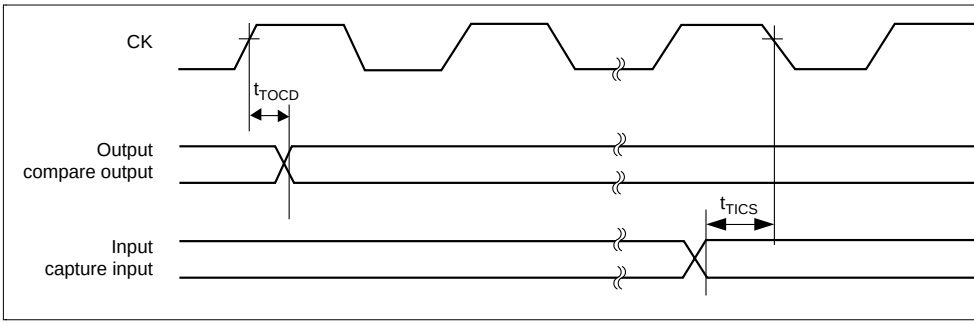


Figure 25.23 MTU I/O Timing

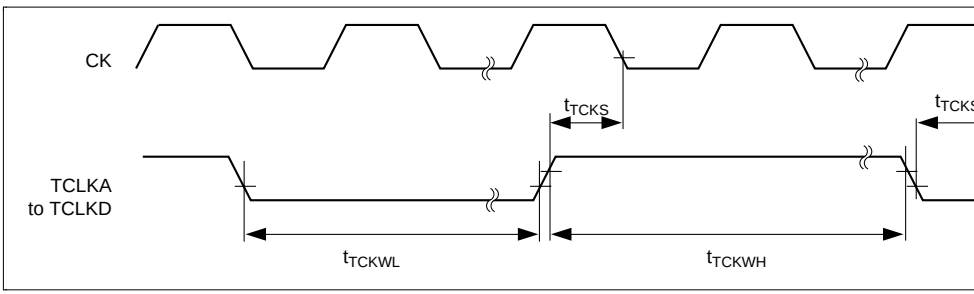


Figure 25.24 MTU Clock Input Timing

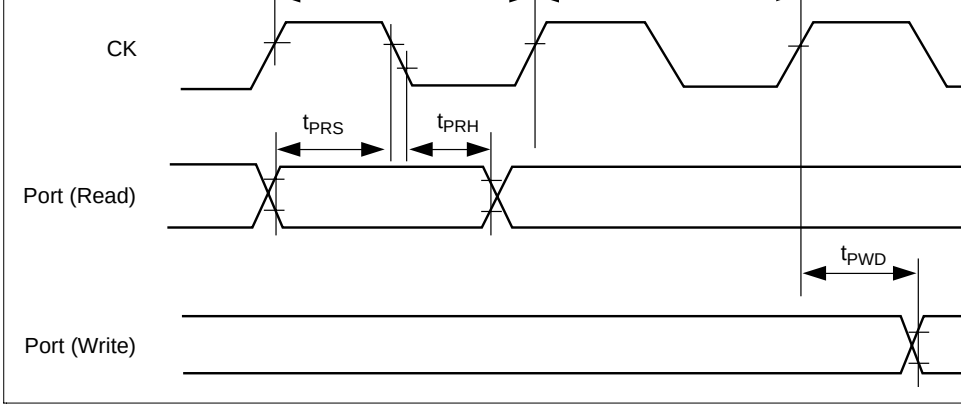


Figure 25.25 I/O Port I/O Timing

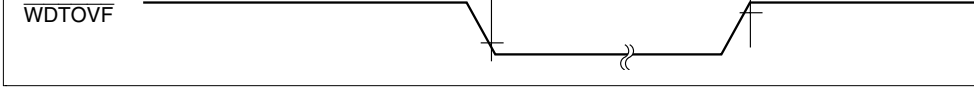


Figure 25.26 Watchdog Timer Timing

Input clock fall time	t_{sckf}		1.5	t_{cyc}
Transmit data delay time (clock sync)	t_{TXD}	—	100	ns
Receive data setup time (clock sync)	t_{RXS}	100	—	ns
Receive data hold time (clock sync)	t_{RXH}	100	—	ns

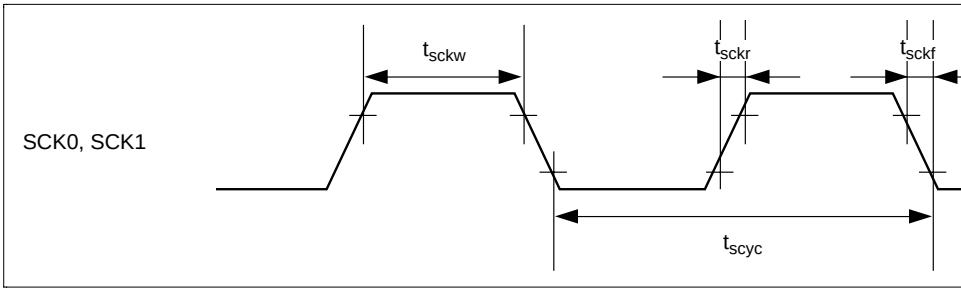


Figure 25.27 Input Clock Timing

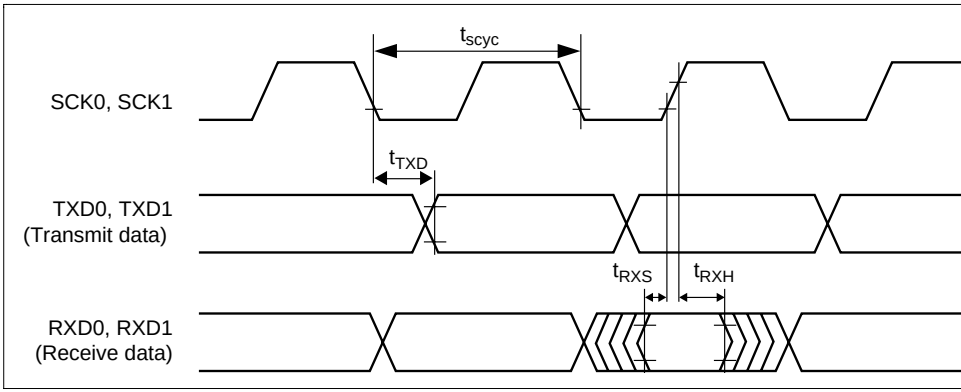


Figure 25.28 SCI I/O Timing (Clock Sync Mode)

	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

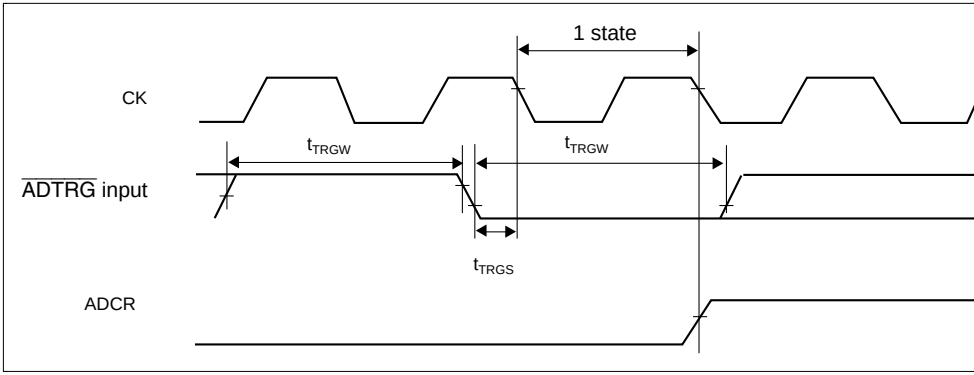


Figure 25.29 External Trigger Input Timing

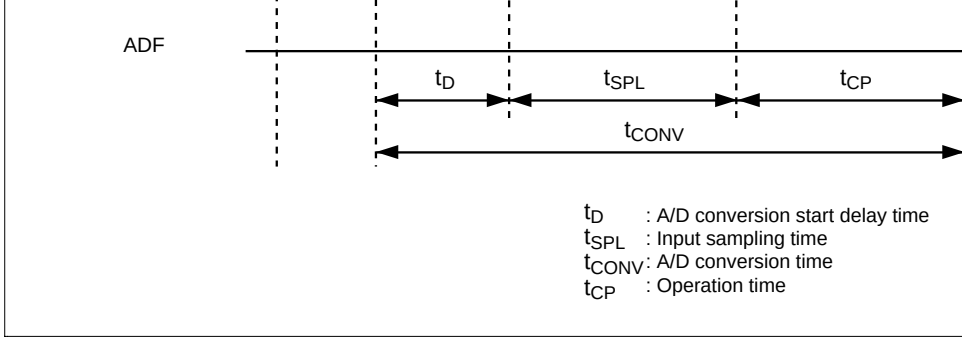


Figure 25.30 Analog Conversion Timing

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

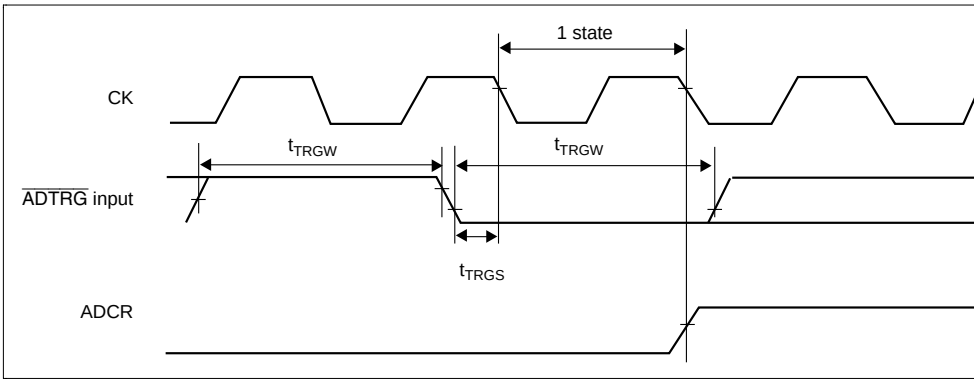
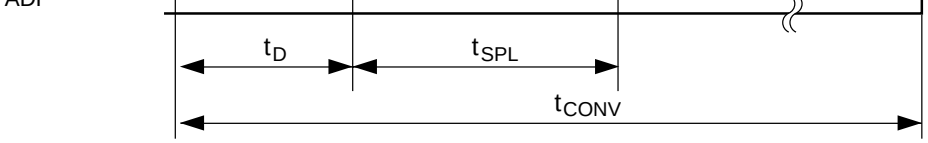


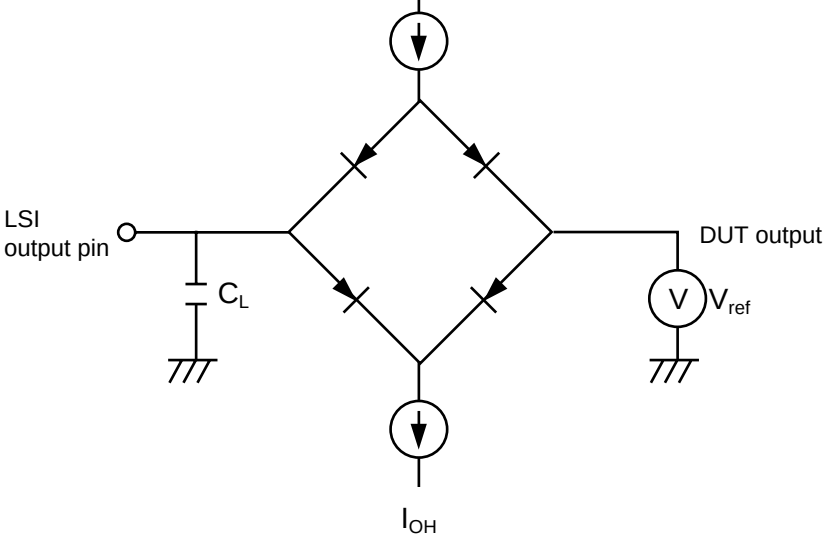
Figure 25.31 External Trigger Input Timing



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



Note: C_L is set with the following pins, including the total capacitance of the measurement equipment etc:

30 pF: CK, $\overline{\text{RAS}}$, $\overline{\text{CASxx}}$, RDWR, $\overline{\text{CS0}}$–$\overline{\text{CS3}}$, $\overline{\text{AH}}$, $\overline{\text{BREQ}}$, $\overline{\text{BACK}}$, DACK DACK1, and $\overline{\text{IRQOUT}}$

50 pF: A21–A0, D31–D0, $\overline{\text{RD}}$, $\overline{\text{WRxx}}$

70 pF: Port output and peripheral module output pins other than the ab

I_{OL}, I_{OH}: See table 25.3, Permitted Output Current Values.

Figure 25.33 Output Load Circuit

impedance				
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.
AVcc=Vcc ± 10%, AVref=4.5V to AVcc, Vss=AVss=0V, Ta=-20 to +75

	28.7 MHz			20 MHz		
Item	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**
Resolution	10	10	10	10	10	10
Conversion time (when CKS=0)	—	—	9.3	—	—	13.4
Analog input capacity	—	—	20	—	—	20
Permission signal source impedance	—	—	1	—	—	1
Non-linearity error*	—	—	±3	—	—	±3
Offset error*	—	—	±3	—	—	±3
Full scale error*	—	—	±3	—	—	±3
Quantize error*	—	—	±0.5	—	—	±0.5
Absolute error	—	—	±4	—	—	±4

Note: * Reference value

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 perm
damage.

	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	$VT^+ \geq V_{CC}$× 0.9 $VT^- \leq V_{CC}$× 0.2
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}
	A/D port		—	—	1.0	µA	V_{in}= 0.5 to AV_{C}
	Other input pins (except EXTAL pin)		—	—	1.0	µA	V_{in}= 0.5 to V_{CC}
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}

tance							
	All other input pins		—	—	20	pF	Ta = 25°C
Current consump-tion	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 versio
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) pin using the A/D converter (including standby).

Connect AV_{CC} (SH7041,SH7043,SH7045 only) and AV_{ref} (SH7041, SH7043 a 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 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 perfo values, operating margins, noise margins, noise emission, etc., are different. therefore required in carrying out system design, and when switching betwee 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.

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} 10%, AV_{ref} = 3.0* to AV_{CC}, V_{SS} = AV_{SS} = 0V, T_a = –20 to +75°C)

Item	Symbol	Min	Max	Unit	Fig
Operating frequency	f_{OP}	4	16.7	MHz	26.
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.
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.
Standby return clock settling time	t_{OSC2}	10	—	ms	

Note: * SH7042/43 ZTAT (excluding A mask) are 3.2V.

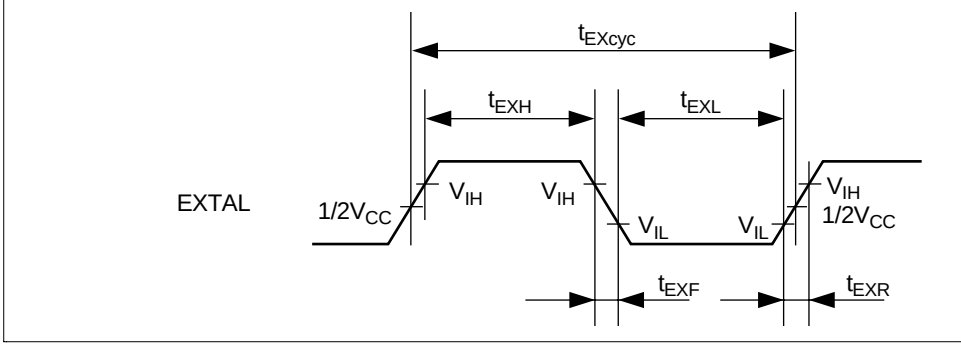


Figure 26.2 EXTAL Clock Input Timing

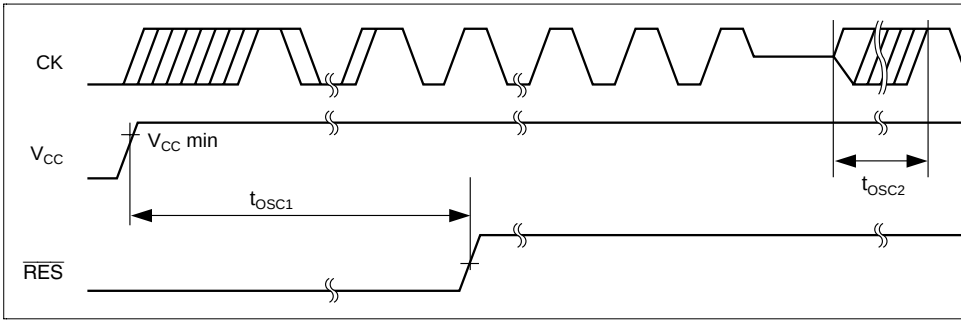


Figure 26.3 Oscillation Settling Time

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.
$\overline{IRQ7}$–$\overline{IRQ0}$ hold time	t_{IRQEH}	50	—	ns	
$\overline{IRQOUT}$ output delay time	t_{IRQOD}	—	50	ns	26.
Bus request setup time	t_{BRQS}	35	—	ns	26.
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 inpu
when the setup times shown here are provided, the signals are considered to h
produced changes at clock rise (for $\overline{RES}$, $\overline{MRES}$, $\overline{BREQ}$) or clock fall (for NMI
$\overline{IRQ7}$–$\overline{IRQ0}$). If the setup times are not provided, recognition is delayed until th
clock rise or fall.

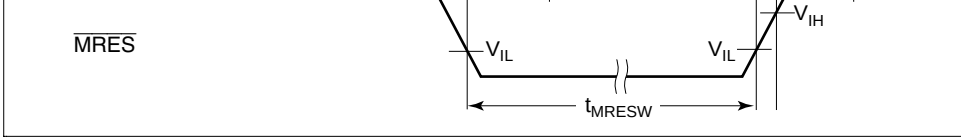


Figure 26.4 Reset Input Timing

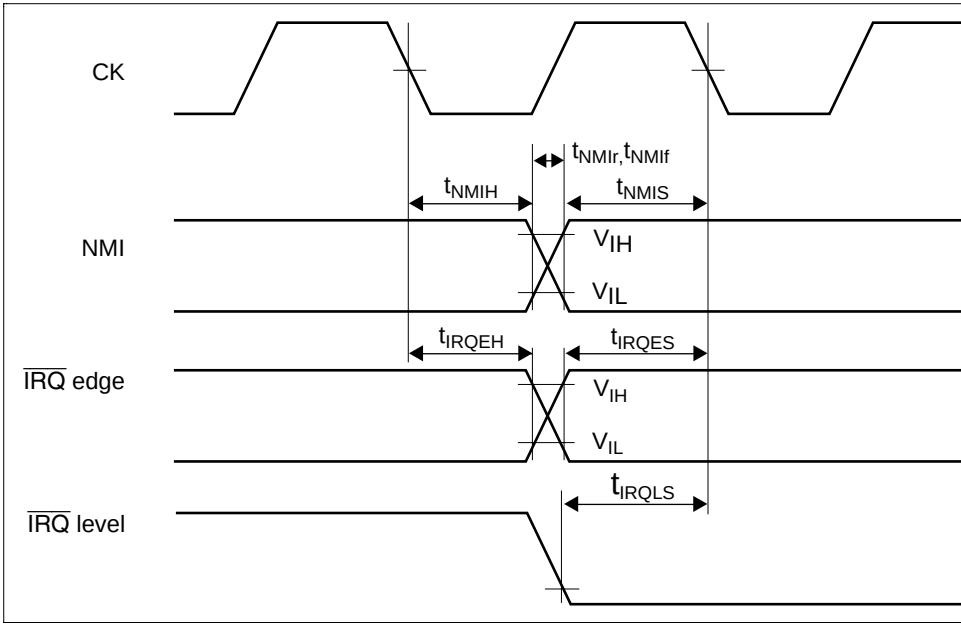
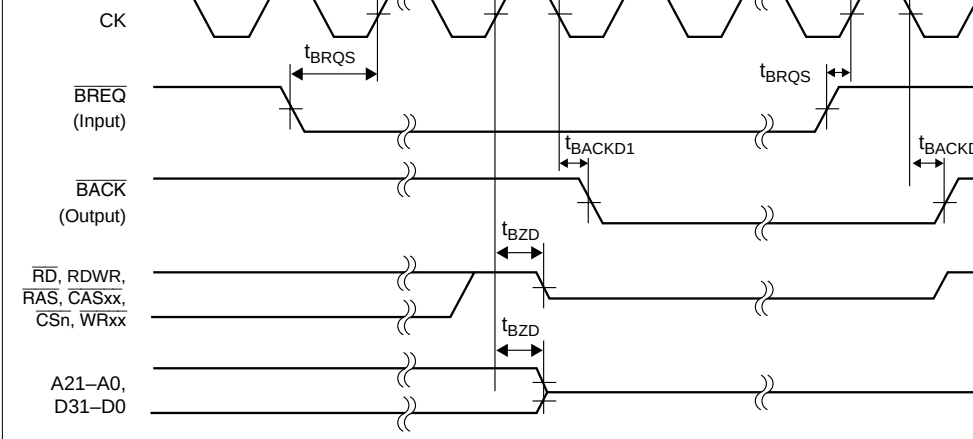


Figure 26.5 Interrupt Signal Input Timing



Note: During the bus-release period of a self-refresh, $\overline{RAS}$, $\overline{CASx}$, and RDWR are output.

Figure 26.7 Bus Right Release Timing

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	$t_{cyc}\times$ (n+2) – 45	—	ns	26.8, 9
Access time from read strobe	t_{OE}*2	$t_{cyc}\times$ (n+1.5) – 40	—	ns	
Access time from column address	t_{AA}*2	$t_{cyc}\times$ (n+2) – 45	—	ns	26.11–16
Access time from $\overline{RAS}$	t_{RAC}*2	$t_{cyc}\times$ (n+RCD+2.5) – 40	—	ns	
Access time from $\overline{CAS}$	t_{CAC}*2	$t_{cyc}\times$ (n+1) – 40	—	ns	
Row address hold time	t_{RAH}	$t_{cyc}\times$ (RCD+0.5) – 15	—	ns	
Row address setup time	t_{ASR}	0	—	ns	
Data input setup time	t_{DS}	$t_{cyc}\times$ (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
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).

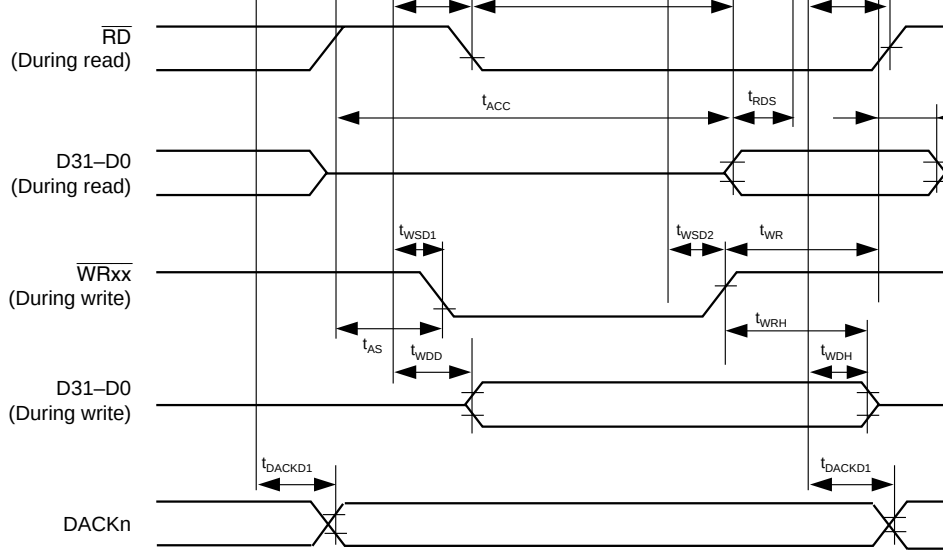
*5 t_{RDS} is a reference value.

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

Notes: TPC is the set value of the TPC bit in DCR.

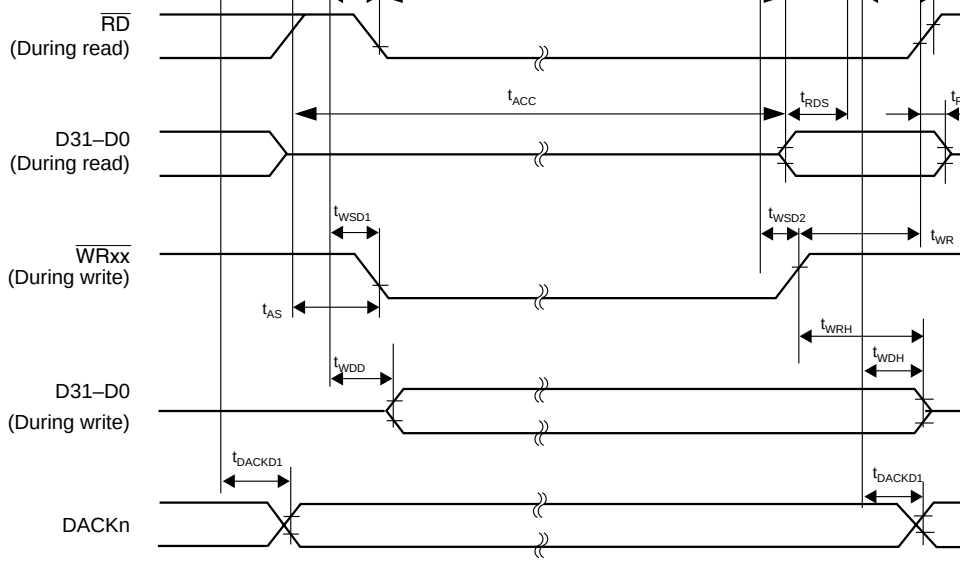
*1 SH7042/43 ZTAT (excluding A mask) are 3.2V

*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)

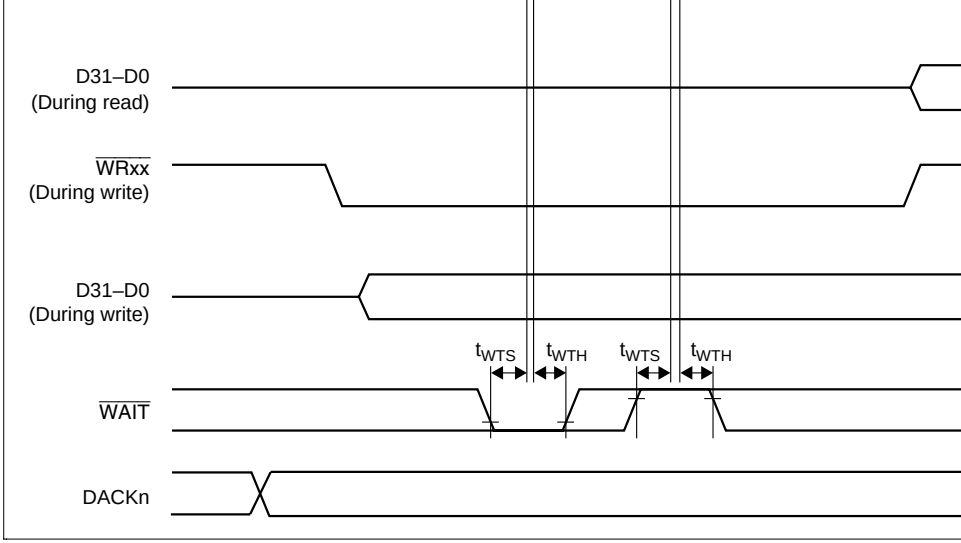
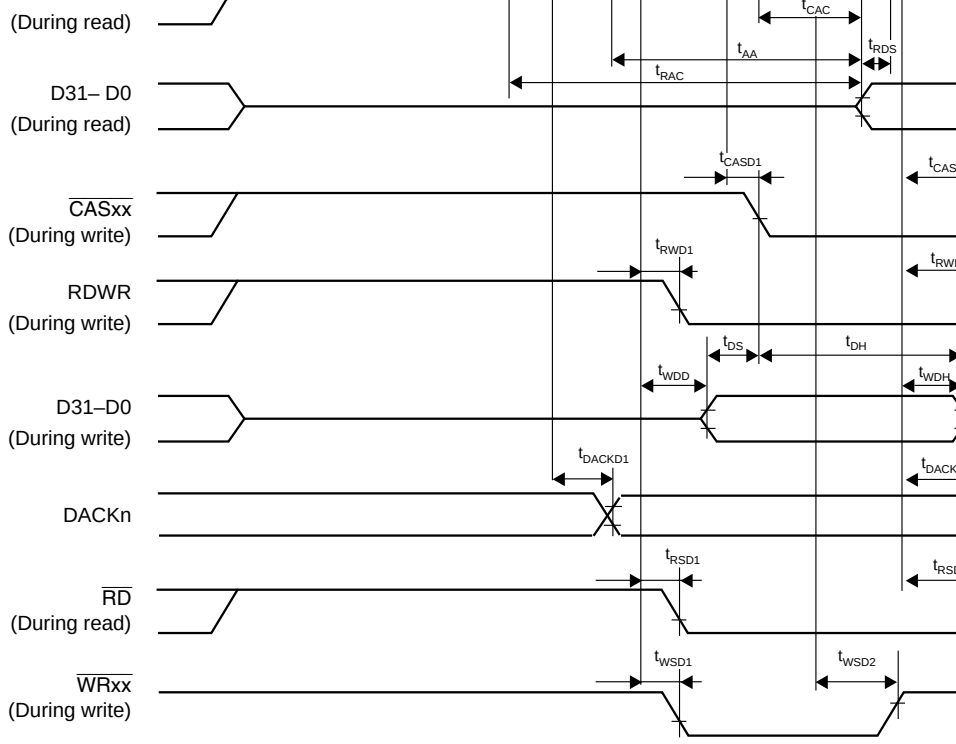
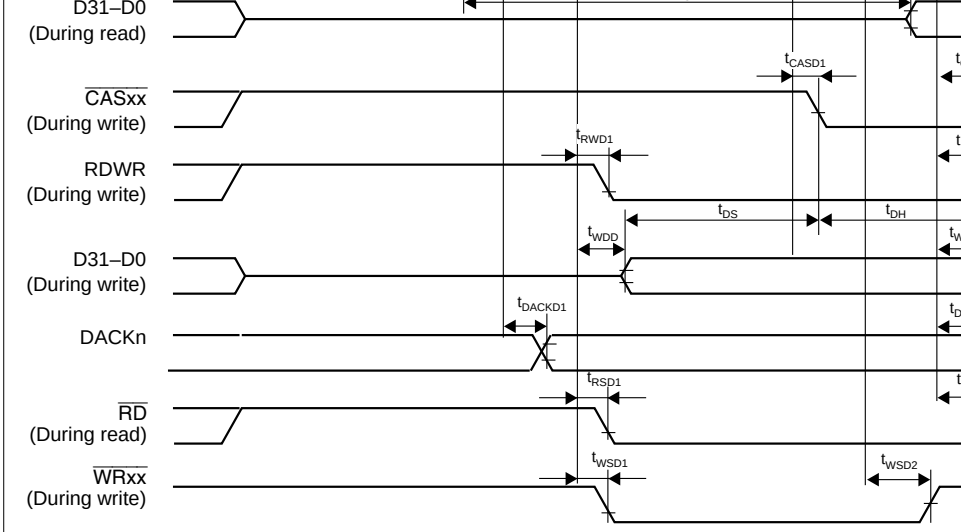


Figure 26.10 Basic Cycle (2 Software Waits + Wait due to $\overline{\text{WAIT}}$ Signal)



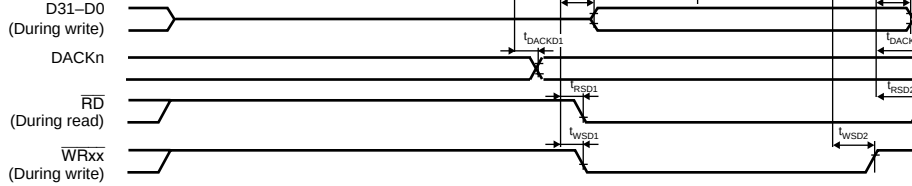
Note: t_{RDH} is specified from fastest negate timing of A21–A0, $\overline{RAS}$, and $\overline{CAS}$.

Figure 26.11 DRAM Cycle (Normal Mode, No Wait, TPC = 0, RCD = 0)



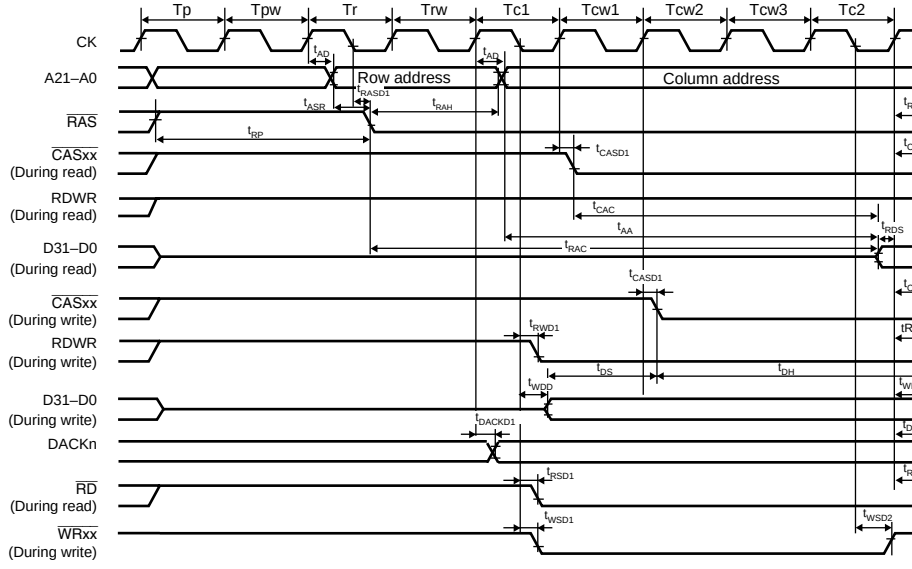
Note: t_{RDH} is specified from fastest negate timing of A21–A0, $\overline{RAS}$, and $\overline{CAS}$.

Figure 26.12 DRAM Cycle (Normal Mode, 1 Wait, TPC = 0, RCD = 0)



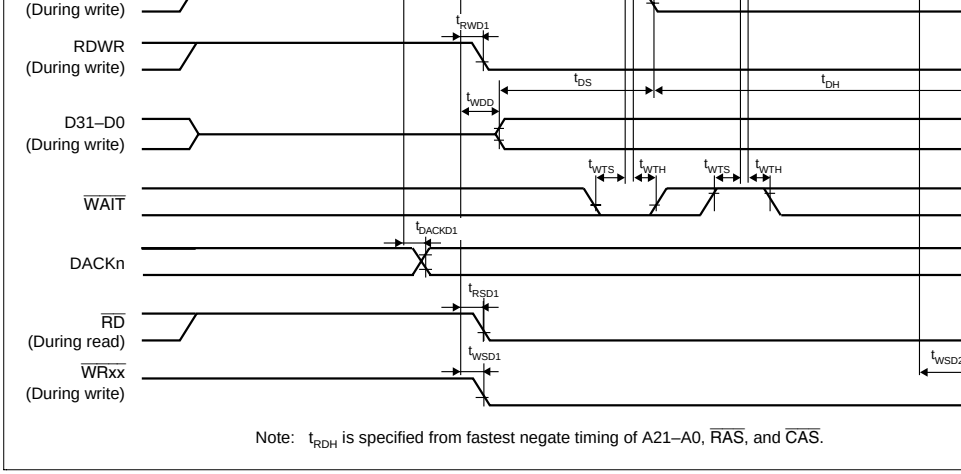
Note: t_{RDH} is specified from fastest negate timing of A21–A0, $\overline{RAS}$, and $\overline{CAS}$.

Figure 26.13 DRAM Cycle (Normal Mode, 2 Waits, TPC = 1, RCD = 1)



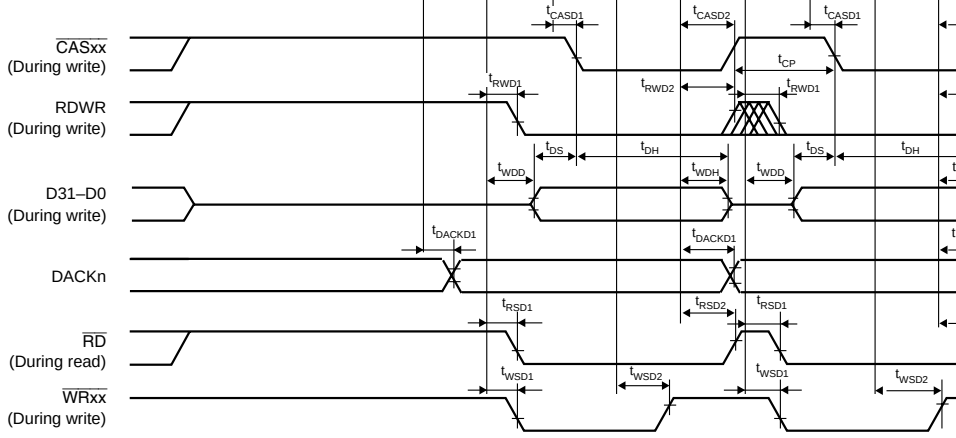
Note: t_{RDH} is specified from fastest negate timing of A21–A0, $\overline{RAS}$, and $\overline{CAS}$.

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 $\overline{WAIT}$ Sig



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)

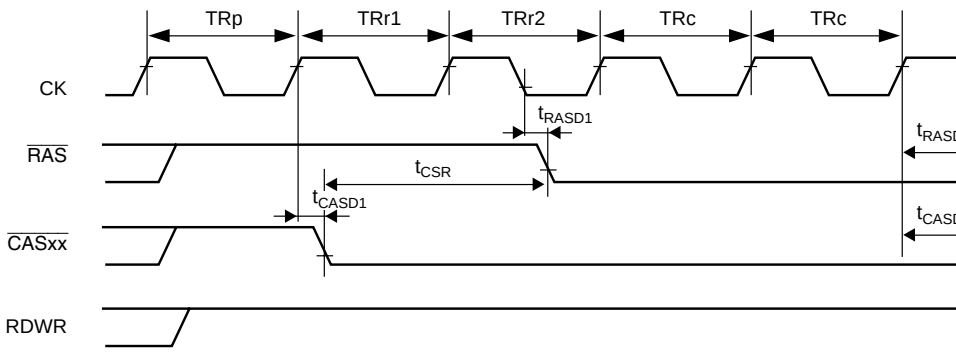
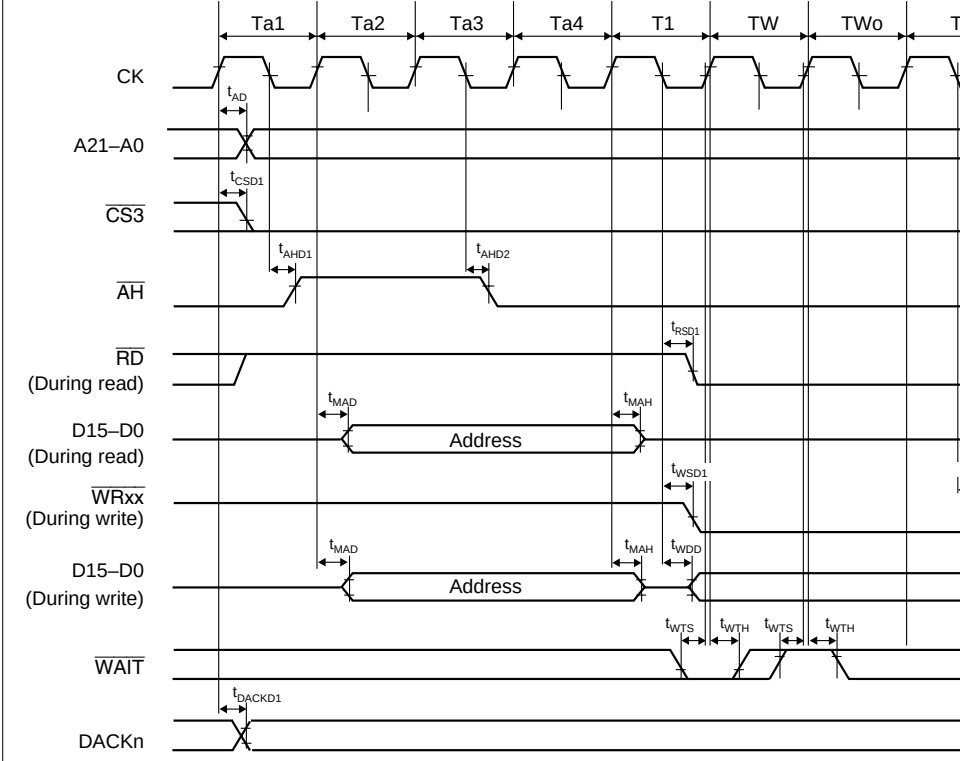


Figure 26.17 CAS Before RAS Refresh (TRAS1 = 0, TRAS0 = 0)



Note: t_{RDH} is specified from fastest negate timing of A21–A0, $\overline{CS3}$, and $\overline{R}$

Figure 26.19 Address Data Multiplex I/O Space Cycle (1 Software Wait + Extern

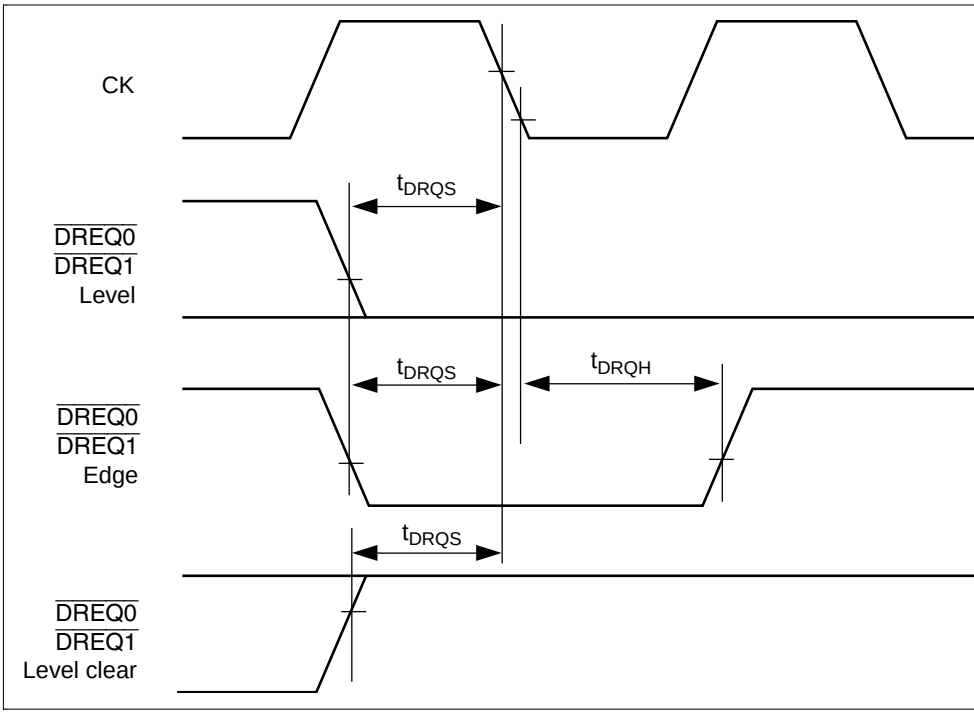


Figure 26.20 $\overline{\text{DREQ0}}$ and $\overline{\text{DREQ1}}$ Input Timing (1)

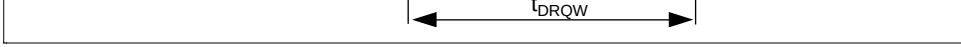


Figure 26.21 $\overline{\text{DREQ0}}$ and $\overline{\text{DREQ1}}$ Input Timing (2)

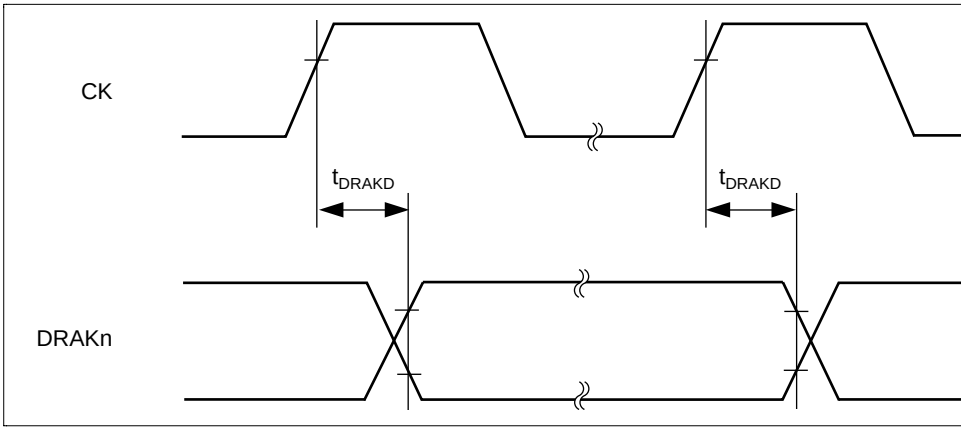


Figure 26.22 DRAK Output Delay Time

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.

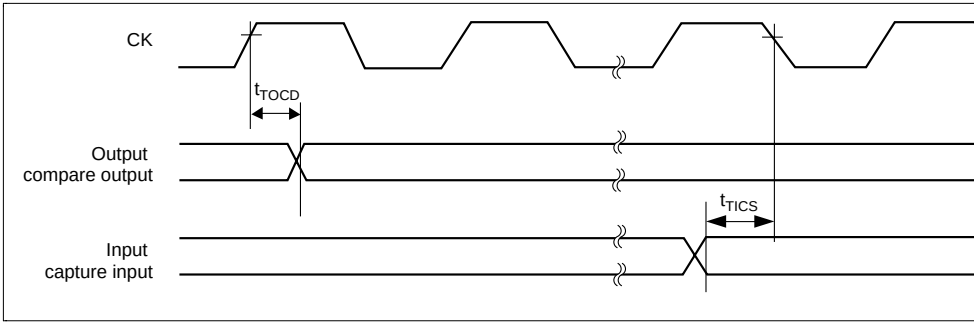


Figure 26.23 MTU I/O Timing

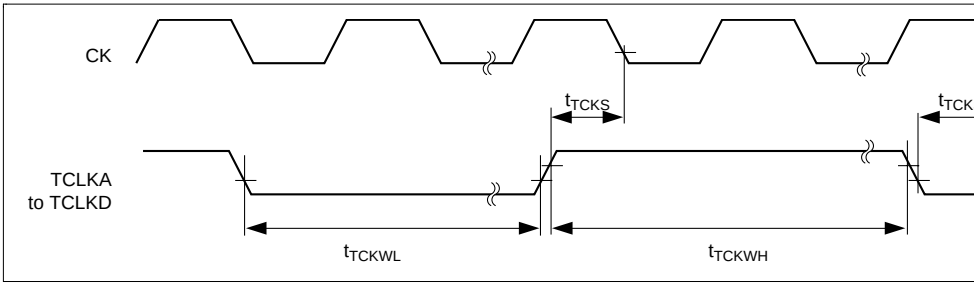


Figure 26.24 MTU Clock Input Timing

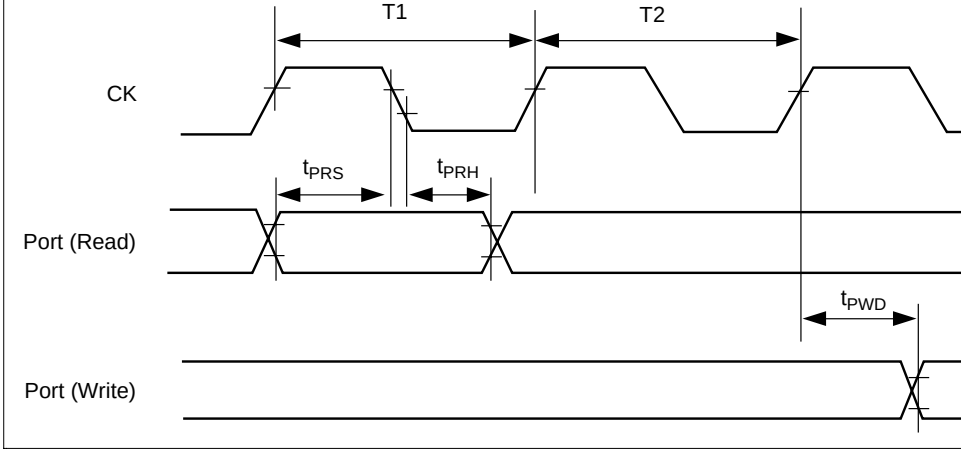


Figure 26.25 I/O Port I/O Timing

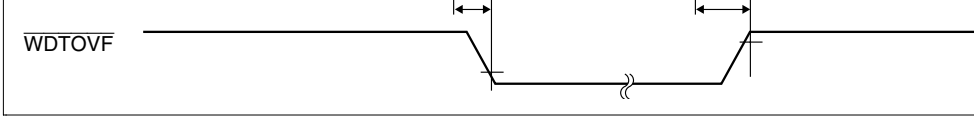


Figure 26.26 Watchdog Timer Timing

Input clock fall time	t_{sckf}		1.5	t_{cyc}	
Transmit data delay time (clock sync)	t_{TXD}	—	100	ns	26
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.

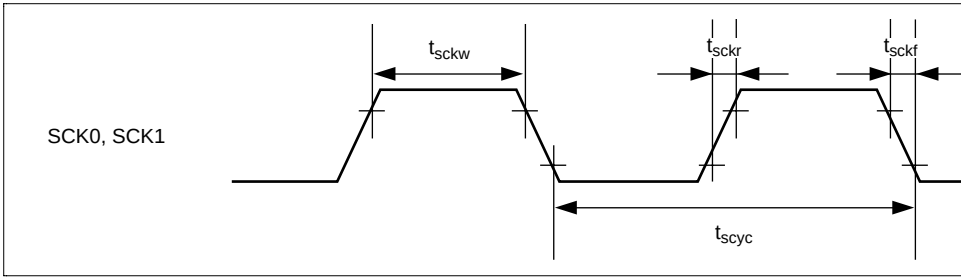


Figure 26.27 Input Clock Timing

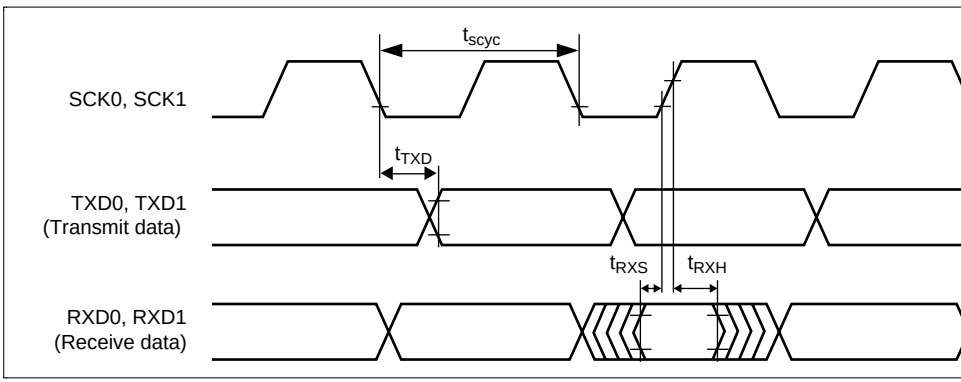


Figure 26.28 SCI I/O Timing (Clock Sync Mode)

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.

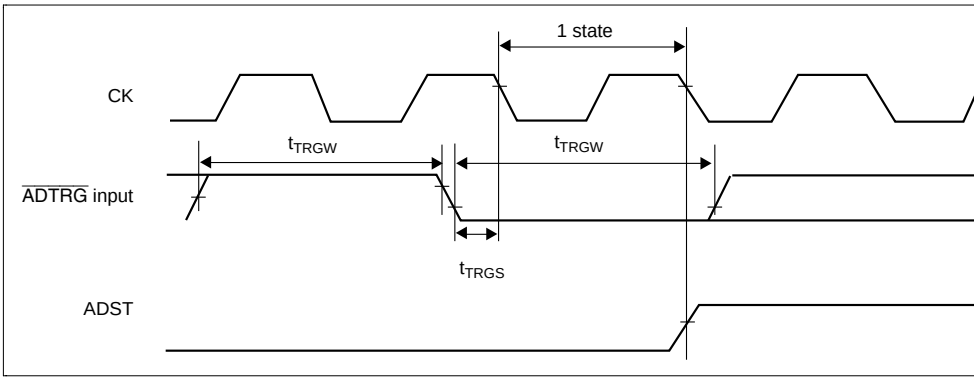


Figure 26.29 External Trigger Input Timing

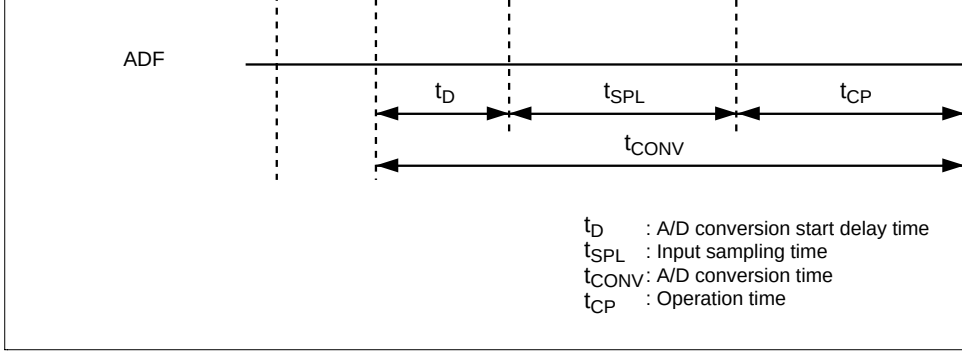


Figure 26.30 Analog Conversion Timing

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.

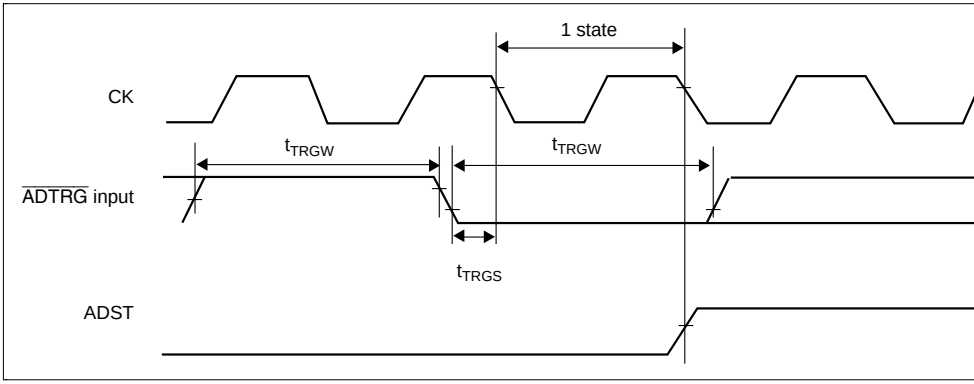
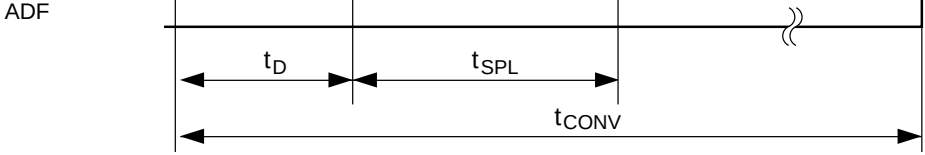


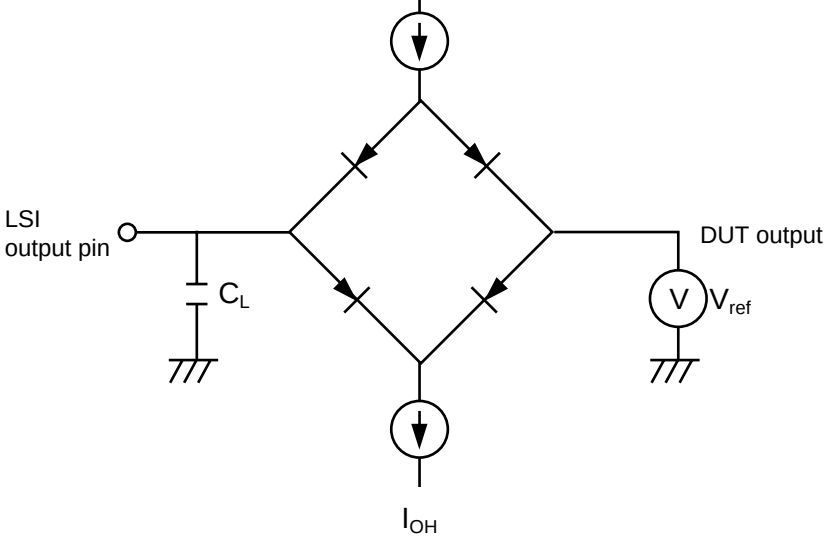
Figure 26.31 External Trigger Input Timing



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



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, $\overline{CS0}$–$\overline{CS3}$, $\overline{AH}$, $\overline{BREQ}$, $\overline{BACK}$, DACK 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 ab

I_{OL}, I_{OH}: See table 26.3, Permitted Output Current Values.

Figure 26.33 Output Load Circuit

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 3.6 = 3.0*1 to 3.6V, AV_{CC} = V_{CC} ± 10%, AV_{ref} = 3.0*1 to AV_{CC}, V_{SS} = AV_{SS} = –20 to +75°C)

	16.7MHz			
Item	**min**	**typ**	**max**	**Uni**
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	LSE
Offset error*2	—	—	±4	LSE
Full scale error*2	—	—	±4	LSE
Quantize error*2	—	—	±0.5	LSE
Absolute error	—	—	±6	LSE

Notes: *1 SH7042/43 ZTAT (excluding A mask) are 3.2V.
*2 Reference values

—	DTDAR								
—	DTIAR								
—	DTCRA								
—	DTCRB								
H'FFFF81A0	SMR0	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0
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	—	—	—	—	—	—	—	—	—

H'FFFF8200	TCR3	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
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									

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	SYNC
H'FFFF8242 to H'FFFF825F	—	—	—	—	—	—	—	—	—
H'FFFF8260	TCR0	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC
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									

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									

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	—	—	—	—	—	—	—	—
H'FFFF8381		PA23DR	PA22DR	PA21DR	PA20DR	PA19DR	PA18DR	PA17DR	PA16D
H'FFFF8382	PADRL	PA15DR	PA14DR	PA13DR	PA12DR	PA11DR	PA10DR	PA9DR	PA8DR
H'FFFF8383		PA7DR	PA6DR	PA5DR	PA4DR	PA3DR	PA2DR	PA1DR	PA0DR
H'FFFF8384	PAIORH	—	—	—	—	—	—	—	—
H'FFFF8385		PA23IOR	PA22IOR	PA21IOR	PA20IOR	PA19IOR	PA18IOR	PA17IOR	PA16IO
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	—	PA20D
H'FFFF8389		PA19MD1	PA19MD0	PA18MD1	PA18MD0	—	PA17MD	—	PA16D

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
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	PB8MD
H'FFFF839A	PBCR2	PB7MD1	PB7MD0	PB6MD1	PB6MD0	PB5MD1	PB5MD0	PB4MD1	PB4MD
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	PD24D
H'FFFF83A1		PD23DR	PD22DR	PD21DR	PD20DR	PD19DR	PD18DR	PD17DR	PD16D
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	PD24IO
H'FFFF83A5		PD23IOR	PD22IOR	PD21IOR	PD20IOR	PD19IOR	PD18IOR	PD17IOR	PD16IO
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	PD28M
H'FFFF83A9		PD27MD1	PD27MD0	PD26MD1	PD26MD0	PD25MD1	PD25MD0	PD24MD1	PD24M
H'FFFF83AA	PDCRH2	PD23MD1	PD23MD0	PD22MD1	PD22MD0	PD21MD1	PD21MD0	PD20MD1	PD20M
H'FFFF83AB		PD19MD1	PD19MD0	PD18MD1	PD18MD0	PD17MD1	PD17MD0	PD16MD1	PD16M

H'FFFF83B4	PEIOR	PE15IOR	PE14IOR	PE13IOR	PE12IOR	PE11IOR	PE10IOR	PE9IOR	PE8IO
H'FFFF83B5		PE7IOR	PE6IOR	PE5IOR	PE4IOR	PE3IOR	PE2IOR	PE1IOR	PE0IO
H'FFFF83B6	—	—	—	—	—	—	—	—	—
H'FFFF83B7	—	—	—	—	—	—	—	—	—
H'FFFF83B8	PECR1	PE15MD1	PE15MD0	PE14MD1	PE14MD0	PE13MD1	PE13MD0	—	PE12I
H'FFFF83B9		—	PE11MD	—	PE10MD	—	PE9MD	—	PE8M
H'FFFF83BA	PECR2	—	PE7MD	—	PE6MD	—	PE5MD	—	PE4M
H'FFFF83BB		PE3MD1	PE3MD0	PE2MD1	PE2MD0	PE1MD1	PE1MD0	PE0MD1	PE0M
H'FFFF83BC to H'FFFF83BF	—	—	—	—	—	—	—	—	—
H'FFFF83C0	ICSR	POE3F	POE2F	POE1F	POE0F	—	—	—	PIE
H'FFFF83C1		POE3M1	POE3M0	POE2M1	POE2M0	POE1M1	POE1M0	POE0M1	POE0
H'FFFF83C2	OCSR	OSF	—	—	—	—	—	OCE	OIE
H'FFFF83C3		—	—	—	—	—	—	—	—
H'FFFF83C4 to H'FFFF83C7	—	—	—	—	—	—	—	—	—
H'FFFF83C8	IFCR	—	—	—	—	—	—	—	—
H'FFFF83C9		—	—	—	—	IRQMD3	IRQMD2	IRQMD1	IRQM
H'FFFF83CA to H'FFFF83CF	—	—	—	—	—	—	—	—	—
H'FFFF83D0	CMSTR	—	—	—	—	—	—	—	—
H'FFFF83D1		—	—	—	—	—	—	STR1	STR0
H'FFFF83D2	CMCSR0	—	—	—	—	—	—	—	—
H'FFFF83D3		CMF	CMIE	—	—	—	—	CKS1	CKS0
H'FFFF83D4	CMCNT0								
H'FFFF83D5									

H'FFFF83DE	—	—	—	—	—	—	—	—	—
H'FFFF83DF	—	—	—	—	—	—	—	—	—
H'FFFF83E0	ADCSR	ADF	ADIE	ADST	CKS	GRP	CH2	CH1	CH0
H'FFFF83E1	ADCR	—	PWR	TRGS1	TRGS0	SCAN	DSMP	BUFE1	BUFE0
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
H'FFFF8401		AD1	AD0	—	—	—	—	—	—
H'FFFF8402	ADDRB0	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2
H'FFFF8403		AD1	AD0	—	—	—	—	—	—
H'FFFF8404	ADDRC0	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2
H'FFFF8405		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
H'FFFF8581	FLMCR2	FLER	—	ESU2	PSU2	EV2	PV2	E2	P2
H'FFFF8582	EBR1	—	—	—	—	EB3	EB2	EB1	EB0
H'FFFF8583	EBR2	EB11	EB10	EB9	EB8	EB7	EB6	EB5	EB4
H'FFFF8584 to H'FFFF859F	—	—	—	—	—	—	—	—	—
H'FFFF8600	UBARH	UBA31	UBA30	UBA29	UBA28	UBA27	UBA26	UBA25	UBA2
H'FFFF8601		UBA23	UBA22	UBA21	UBA20	UBA19	UBA18	UBA17	UBA1
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	UBM2
H'FFFF8605		UBM23	UBM22	UBM21	UBM20	UBM19	UBM18	UBM17	UBM1
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

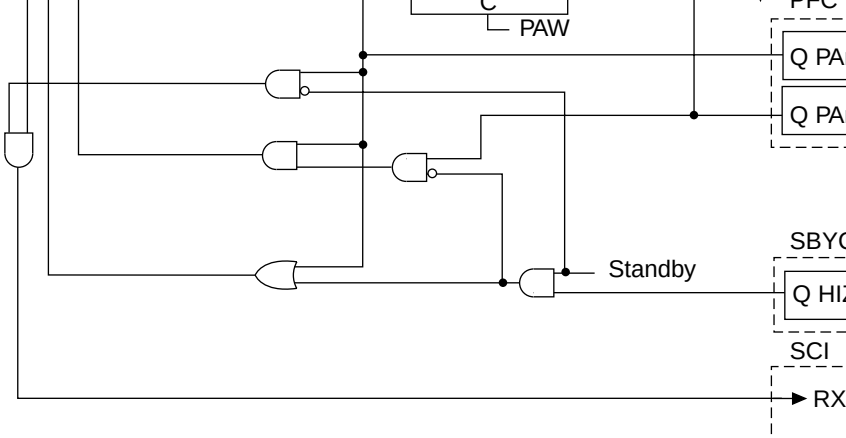
H'FFFF8620	BCR1	—	—	MTURWE	—	—	—	—	IOE
H'FFFF8621		A3LG	A2LG	A1LG	A0LG	A3SZ	A2SZ	A1SZ	A0SZ
H'FFFF8622	BCR2	IW31	IW30	IW21	IW20	IW11	IW10	IW01	IW00
H'FFFF8623		CW3	CW2	CW1	CW0	SW3	SW2	SW1	SW0
H'FFFF8624	WCR1	W33	W32	W31	W30	W23	W22	W21	W20
H'FFFF8625		W13	W12	W11	W10	W03	W02	W01	W00
H'FFFF8626	WCR2	—	—	—	—	—	—	—	—
H'FFFF8627		—	—	DDW1	DDW0	DSW3	DSW2	DSW1	DSW0
H'FFFF8628	RAMER	—	—	—	—	—	—	—	—
H'FFFF8629		—	—	—	—	—	RAMS	RAM1	RAM0
H'FFFF862A	DCR	TPC	RCD	TRAS1	TRAS0	DWW1	DWW0	DWR1	DWR0
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).

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									

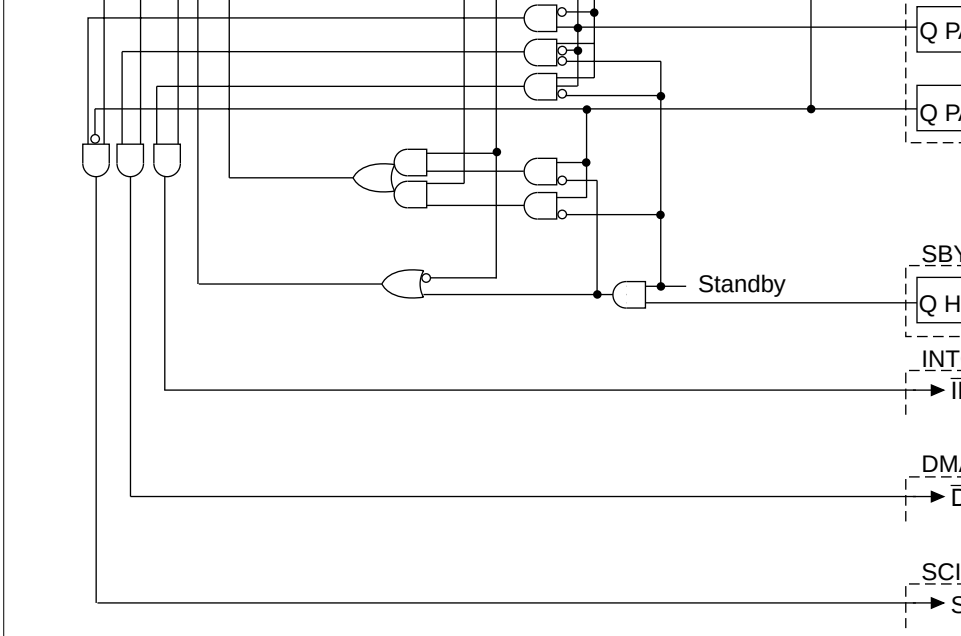
H'FFFF86E4	DAR2								
H'FFFF86E5									
H'FFFF86E6									
H'FFFF86E7									
H'FFFF86E8	DMATCR2	—	—	—	—	—	—	—	—
H'FFFF86E9									
H'FFFF86EA									
H'FFFF86EB									
H'FFFF86EC	CHCR2	—	—	—	—	—	—	—	—
H'FFFF86ED		—	—	—	DI	RO	RL	AM	AL
H'FFFF86EE		DM1	DM0	SM1	SM0	RS3	RS2	RS1	RS0
H'FFFF86EF		—	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

H'FFFF8708	DTBR								
H'FFFF8709									
H'FFFF870A to H'FFFF873F	—	—	—	—	—	—	—	—	—
H'FFFF8740	CCR	—	—	—	—	—	—	—	—
H'FFFF8741		—	—	—	CEDRAM	CECS3	CECS2	CECS1	CECS
H'FFFF8742 to H'FFFF87FF	—	—	—	—	—	—	—	—	—



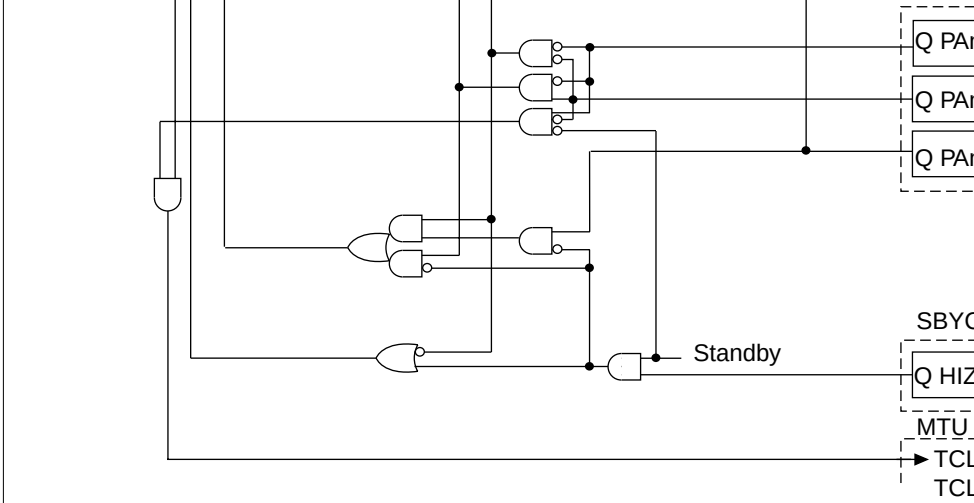
n = 0, 3
m = 0, 1
PAR: Port A read signal
PAW: Port A write signal
RES: Reset signal

Figure B.1 PAn/RXDm Block Diagram



n = 2, 5
m = 0, 1
PAR: Port A read signal
PAW: Port A write signal
RES: Reset signal

Figure B.2 PAn/SCKm/$\overline{\text{DREQm}}$/$\overline{\text{IRQm}}$ Block Diagram



n = 6, 7
PAR: Port A read signal
PAW: Port A write signal
RES: Reset signal

Figure B.3 PA6/TCLKA/$\overline{CS2}$, PA7/TCLKB/$\overline{CS3}$ (ZTAT, Mask) Block Diagr

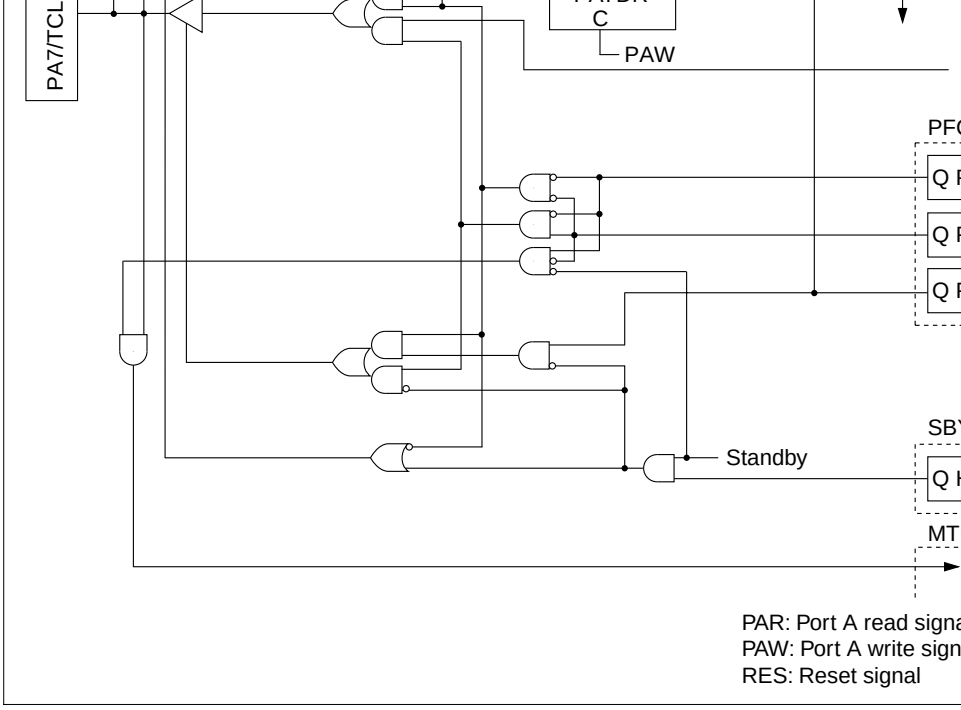
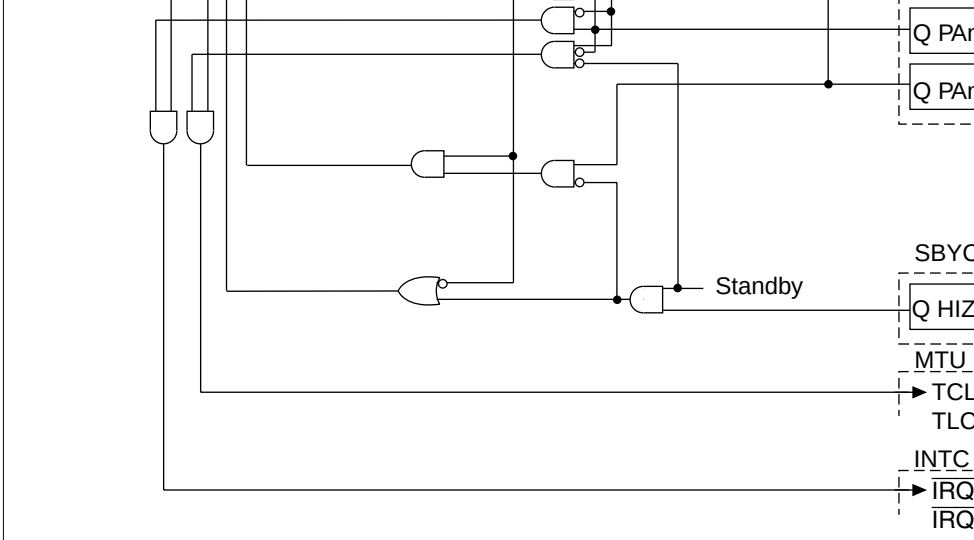


Figure B.4 PA7/TCLKB/$\overline{CS3}$ Block Diagram (F-ZTAT Version)



n = 8, 9
PAR: Port A read signal
PAW: Port A write signal
RES: Reset signal

Figure B.5 PAn/TCLKm/$\overline{\text{IRQx}}$ Block Diagram (ZTAT, Mask)

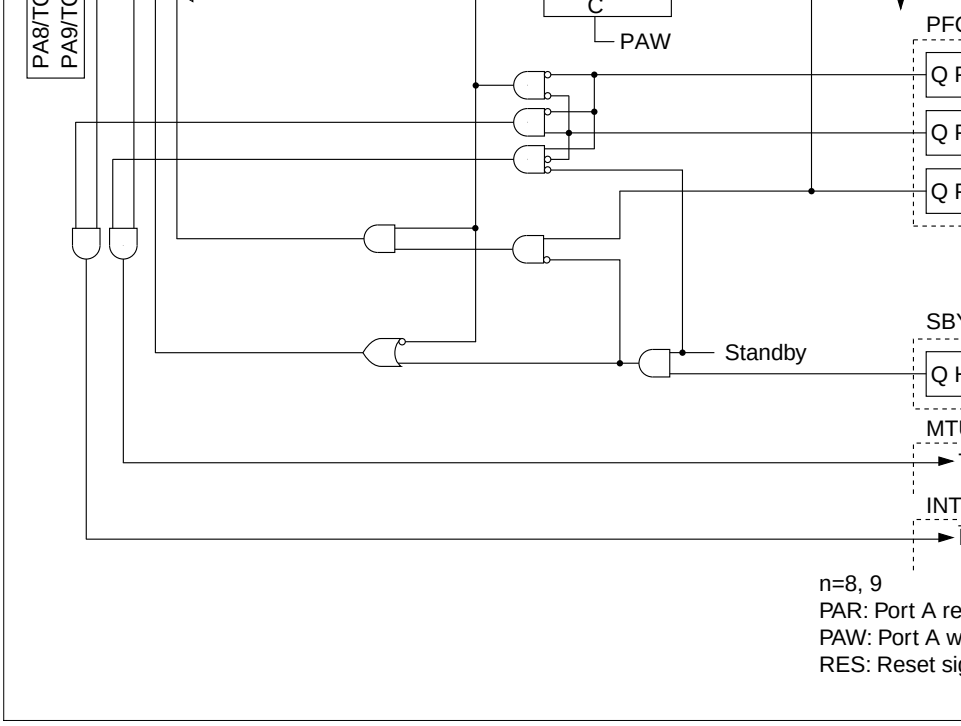
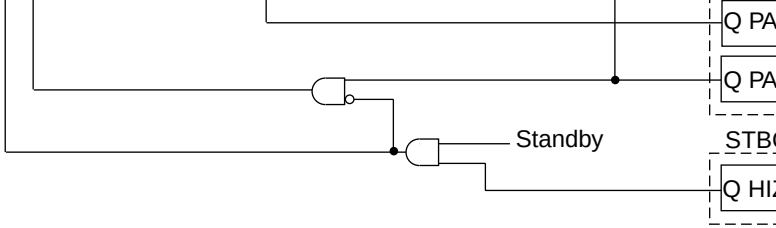
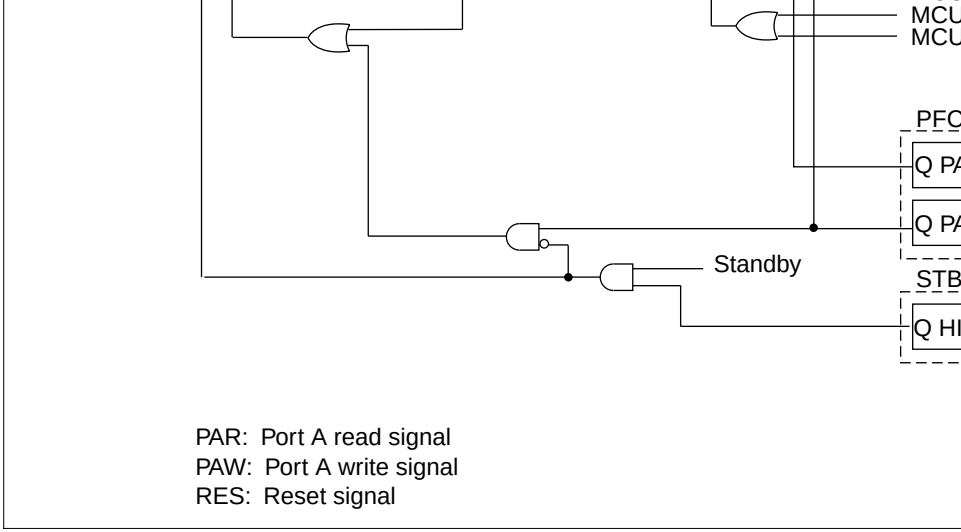


Figure B.6 PAn/TCLKm/$\overline{\text{IRQx}}$ Block Diagram (F-ZTAT Version)



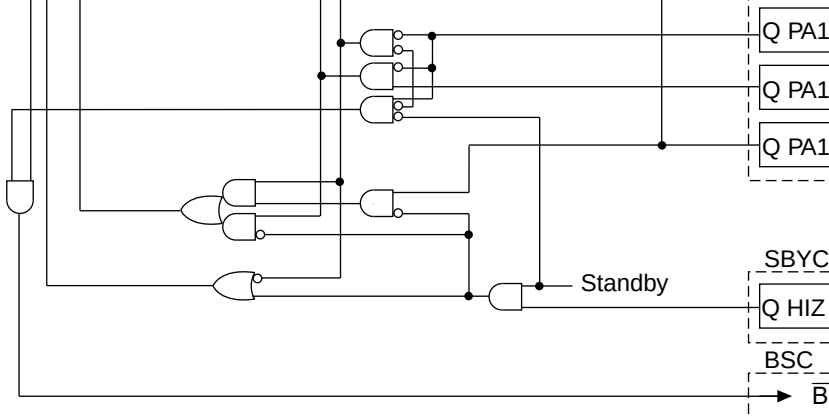
n = 1, 4
m = 0, 1
PAR: Port A read signal
PAW: Port A write signal
RES: Reset signal

Figure B.7 PAn/TXDm Block Diagram



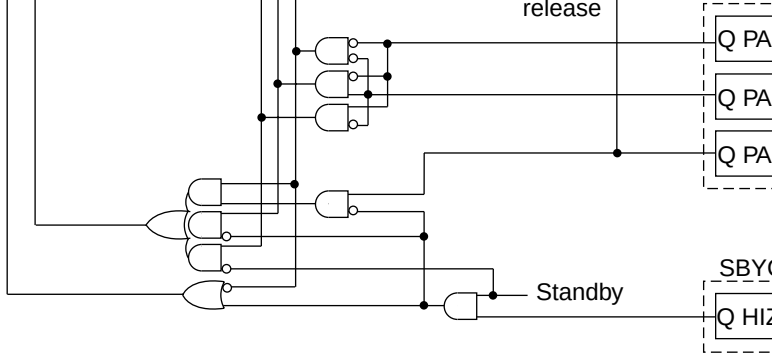
PAR: Port A read signal
PAW: Port A write signal
RES: Reset signal

Figure B.8 PA15/CK Block Diagram



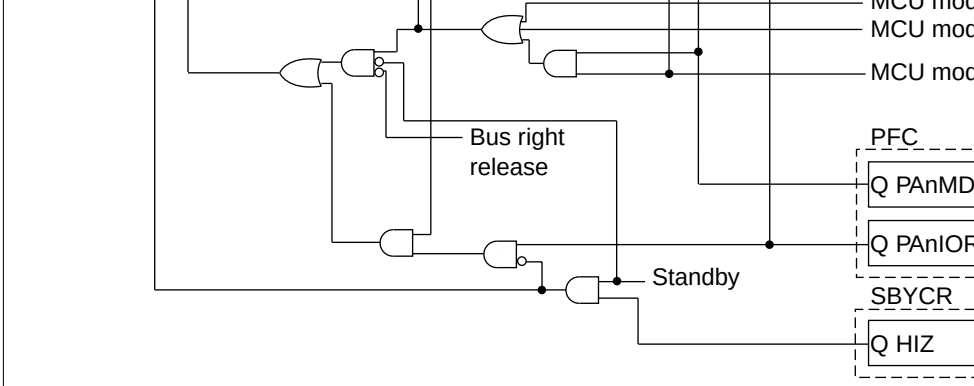
PAR: Port A read signal
PAW: Port A write signal
RES: Reset signal

Figure B.9 PA18/DRAK0/$\overline{\text{BREQ}}$ Block Diagram



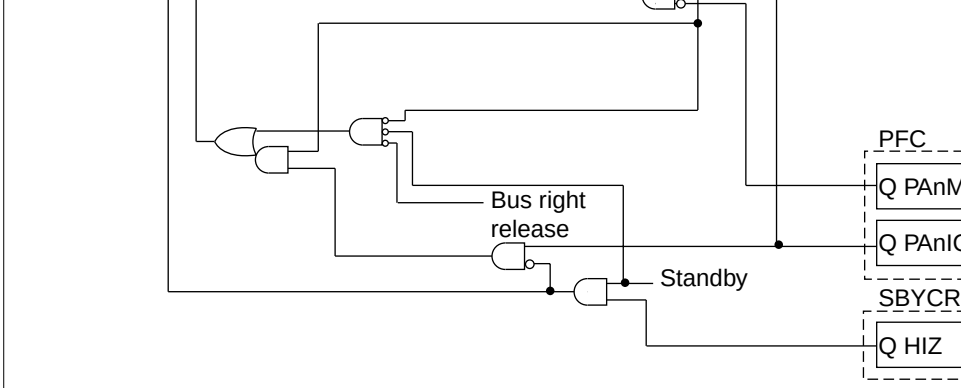
PAR: Port A read signal
PAW: Port A write signal
RES: Reset signal

Figure B.10 PA19/DRAQ1/$\overline{\text{BACK}}$ Block Diagram



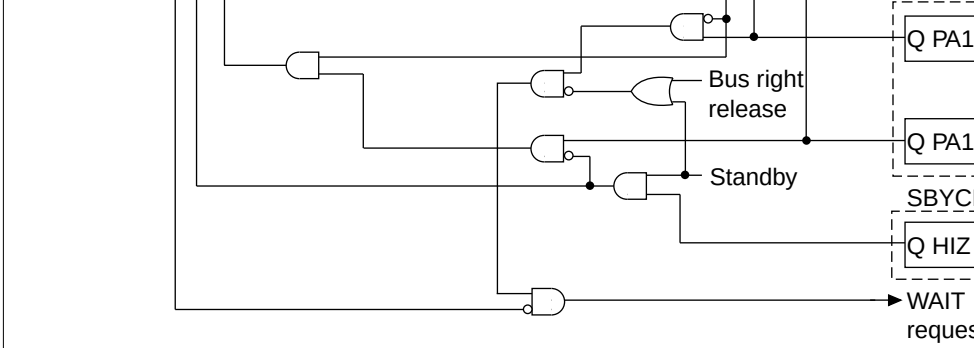
n = 10-14
PAR: Port A read signal
PAW: Port A write signal
RES: Reset signal

Figure B.11 PAn/XXX Block Diagram



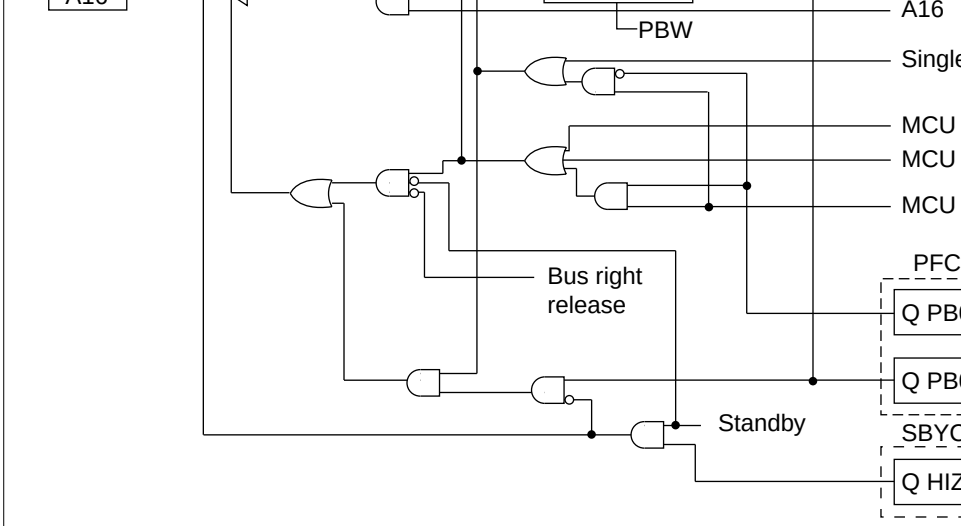
n = 16,20-23
PAR: Port A read signal
PAW: Port A write signal
RES: Reset signal

Figure B.12 PAn/XXXX Block Diagram



PAR: Port A read signal
PAW: Port A write signal
RES: Reset signal

Figure B.13 PA17/$\overline{\text{WAIT}}$ Block Diagram



PBR: Port B read signal
PBW: Port B write signal
RES: Reset signal

Note: * Not available with the mask versions.

Figure B.14 PB0/A16 Block Diagram

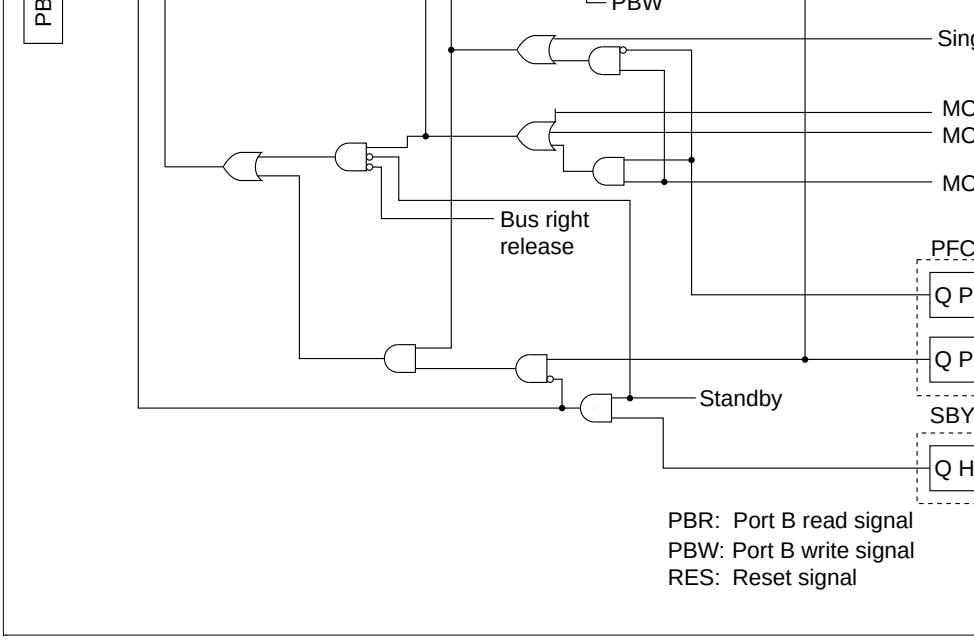
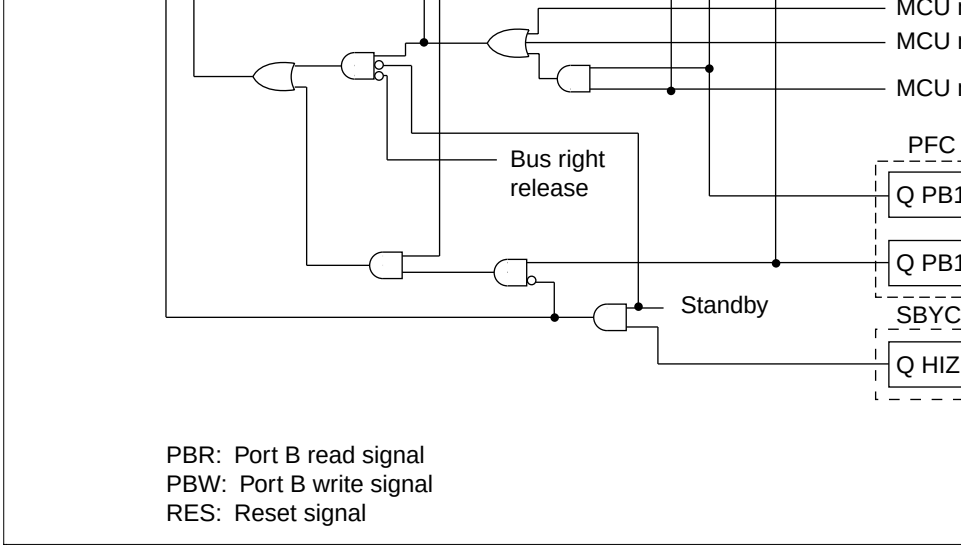
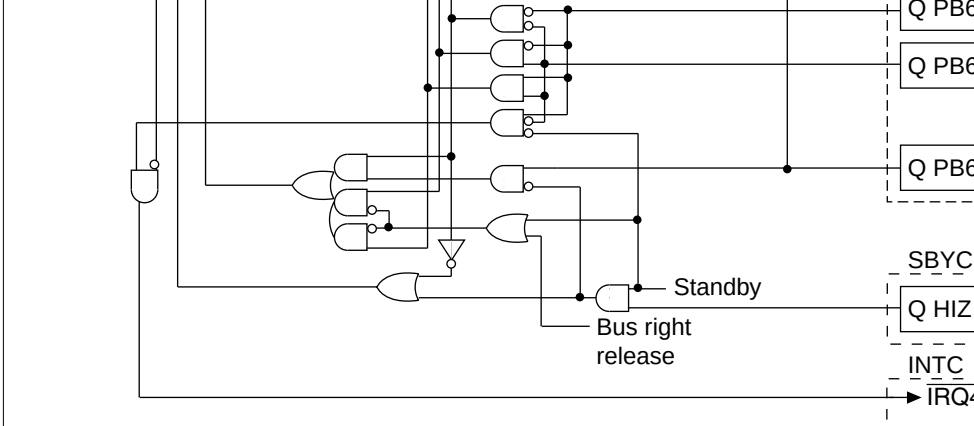


Figure B.15 PB0/A16 Block Diagram (F-ZTAT Version)



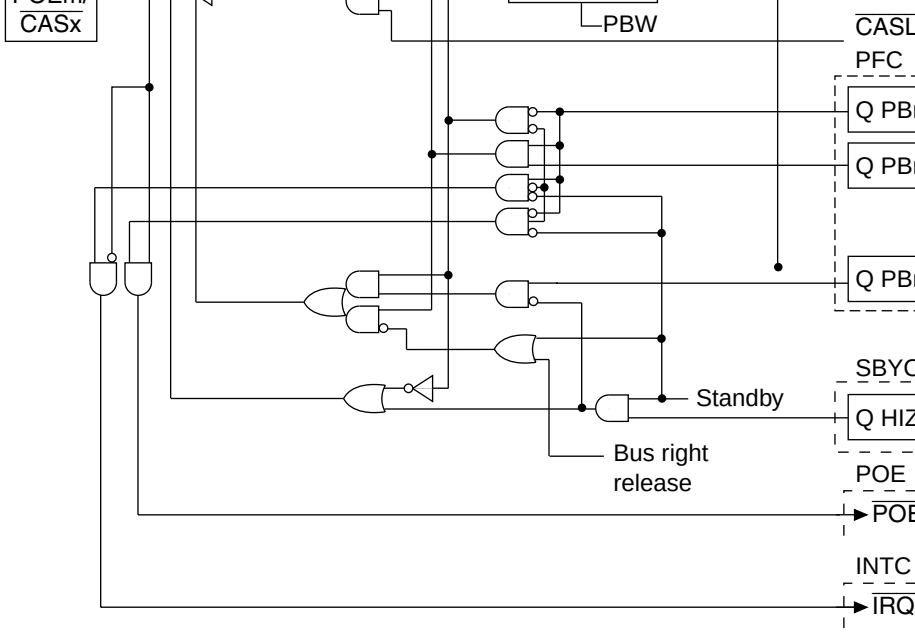
PBR: Port B read signal
PBW: Port B write signal
RES: Reset signal

Figure B.16 PB1/A17 Block Diagram



PBR: Port B read signal
PBW: Port B write signal
RES: Reset signal

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/$\overline{\text{IRQm}}$/$\overline{\text{POEm}}$/$\overline{\text{CASx}}$ Block Diagram

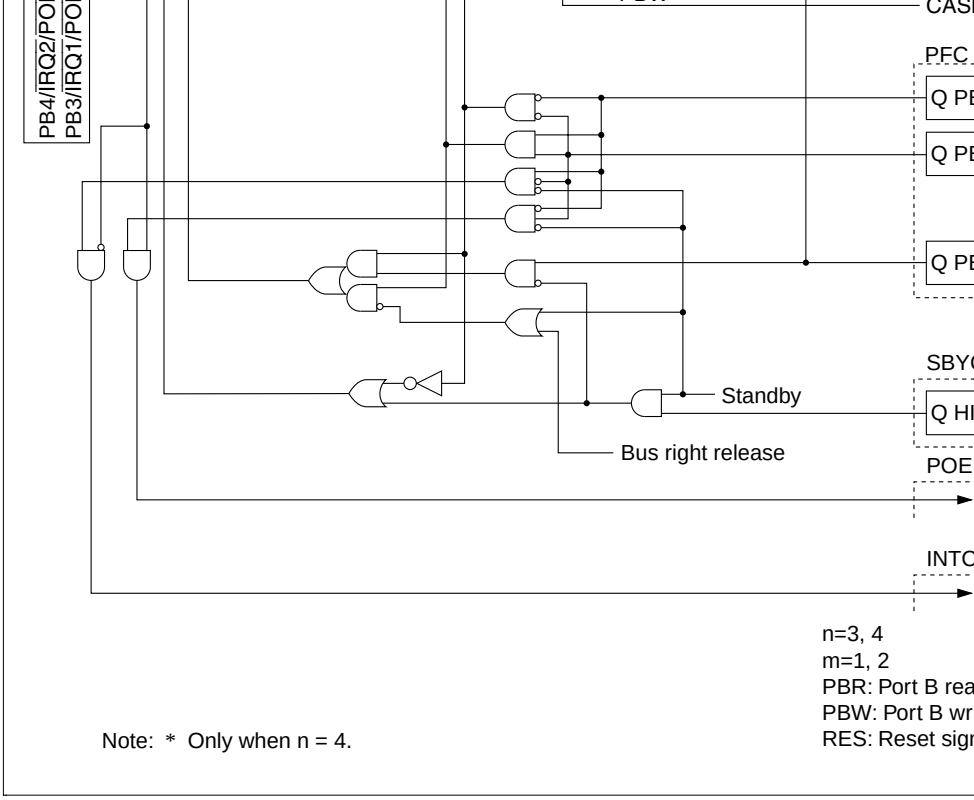
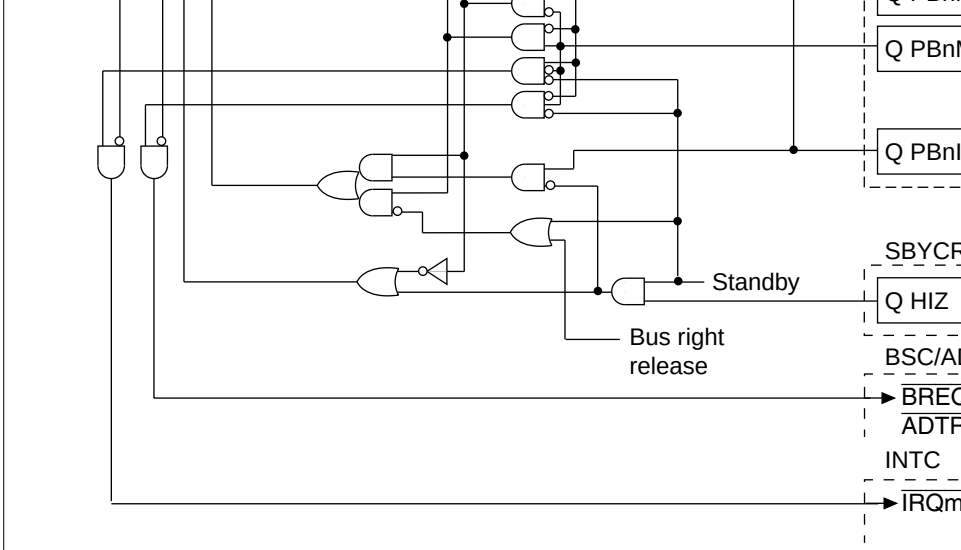
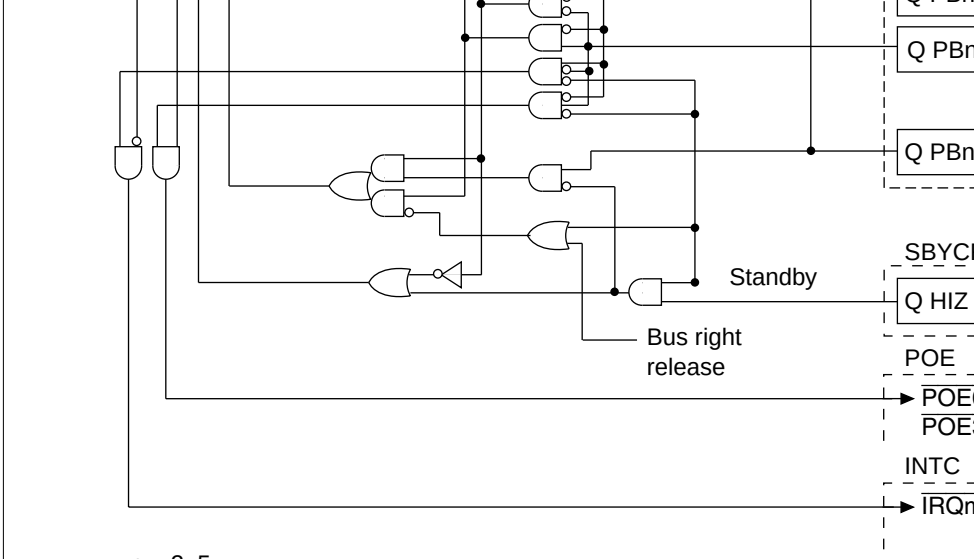


Figure B.19 PB4/$\overline{\text{IRQ2}}$/$\overline{\text{POE2}}$/$\overline{\text{CASH}}$,PB3/$\overline{\text{IRQ1}}$/$\overline{\text{POE1}}$/$\overline{\text{CASL}}$ Block Diagram (F-ZTAT Version)



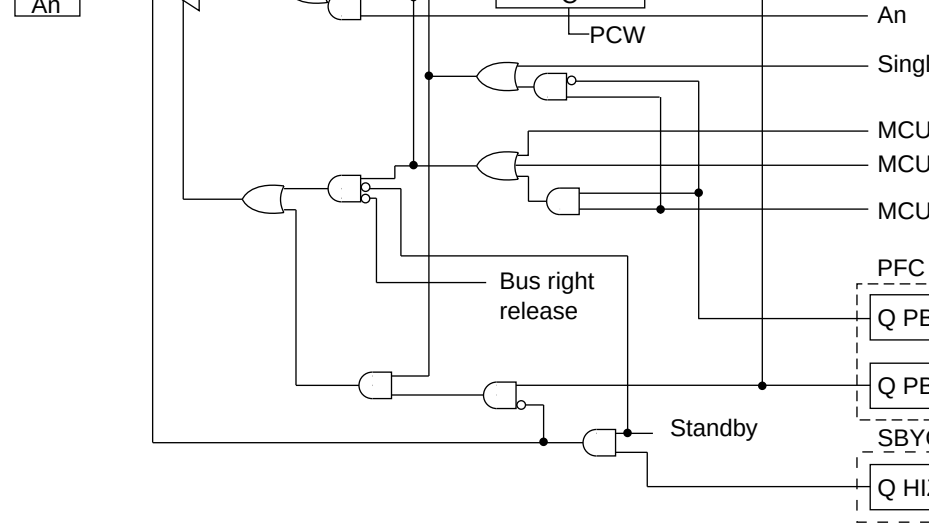
n = 7-9
m = 5-7
PBR: Port B read signal
PBW: Port B write signal
RES: Reset signal

Figure B.20 PBn/$\overline{\text{IRQm}}$/XXX/YYY Block Diagram



n = 2, 5
m = 0, 3
PBR: Port B read signal
PBW: Port B write signal
RES: Reset signal

Figure B.21 PBn/$\overline{\text{IRQm}}$/XXXX/YYYY Block Diagram



n = 0–15
PCR: Port C read signal
PCW: Port C write signal
RES: Reset signal

Note: * Not available with the mask versions.

Figure B.22 PCn/An Block Diagram

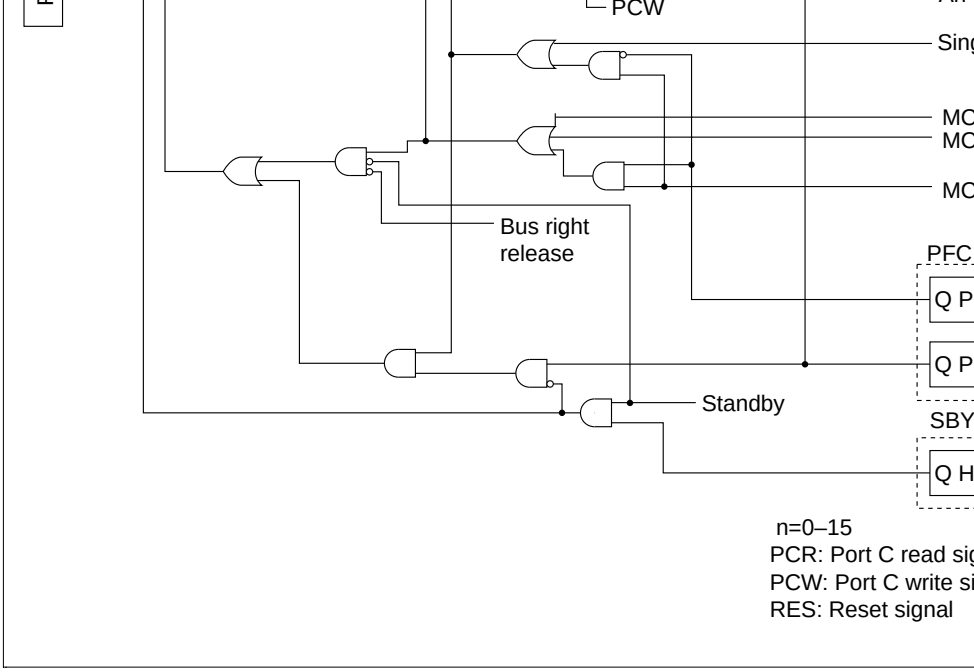
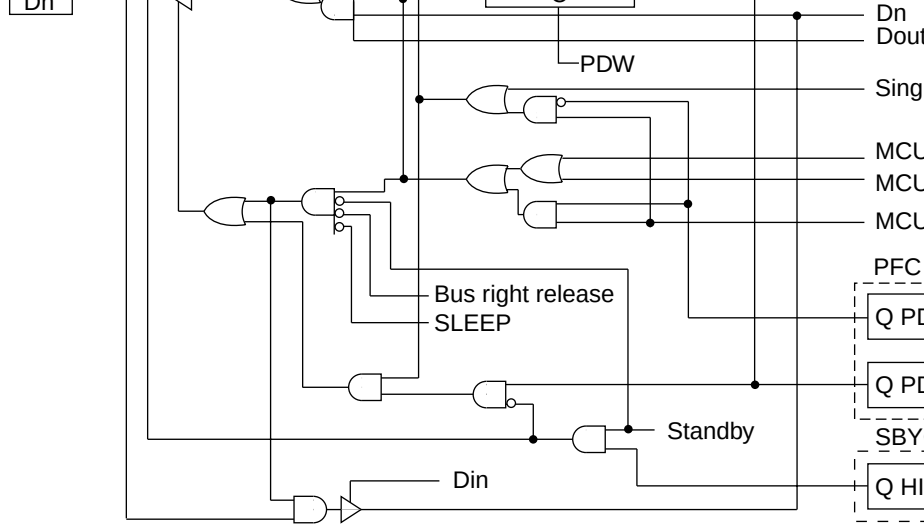


Figure B.23 PCn/An Block Diagram (F-ZTAT Version)



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

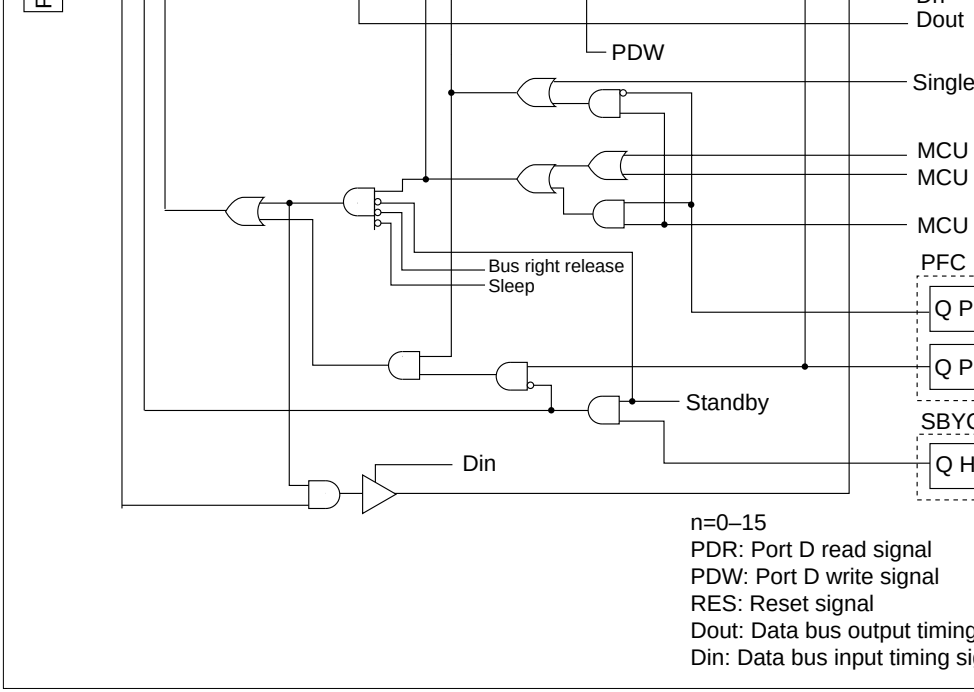
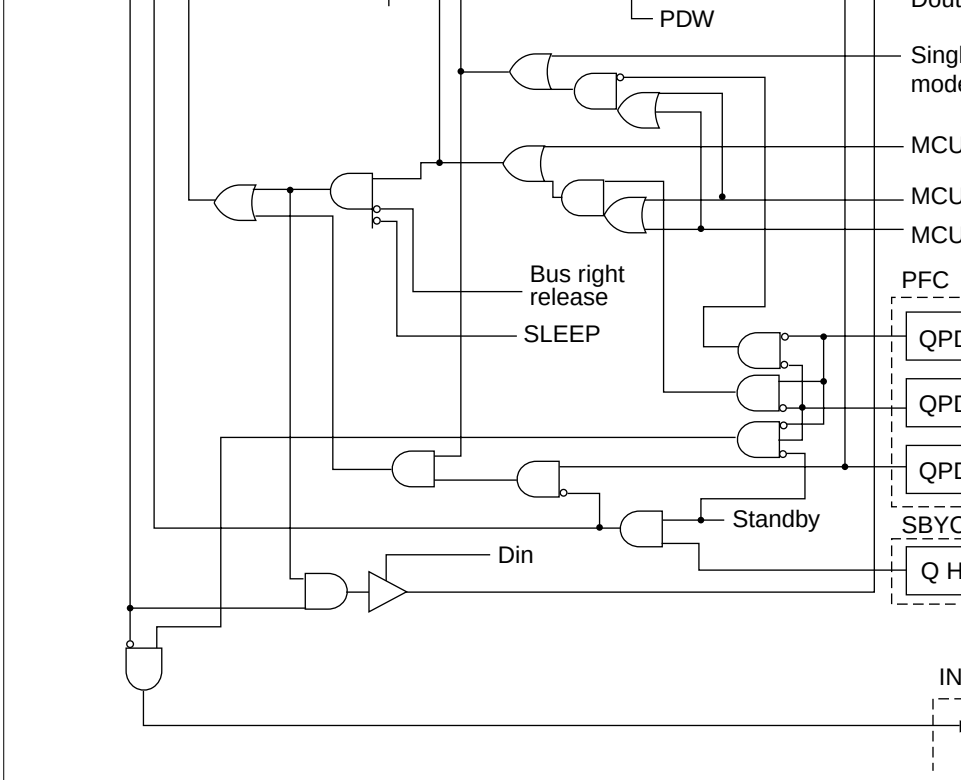


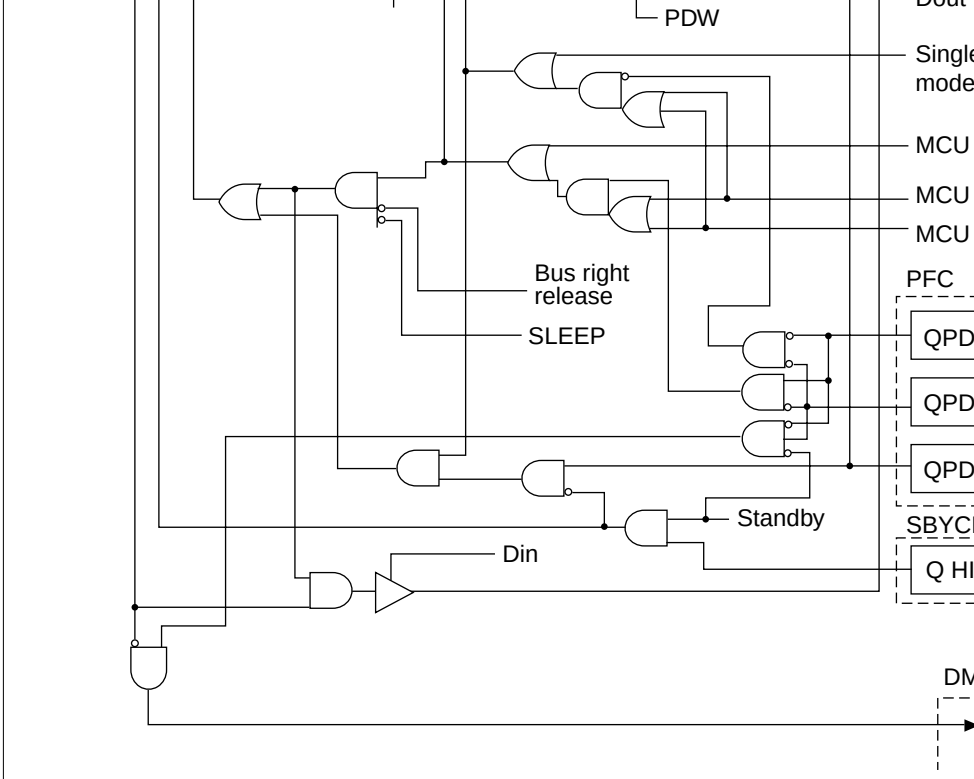
Figure B.25 PDn/Dn Block Diagram (F-ZTAT Version)



n = 16–23
m = 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

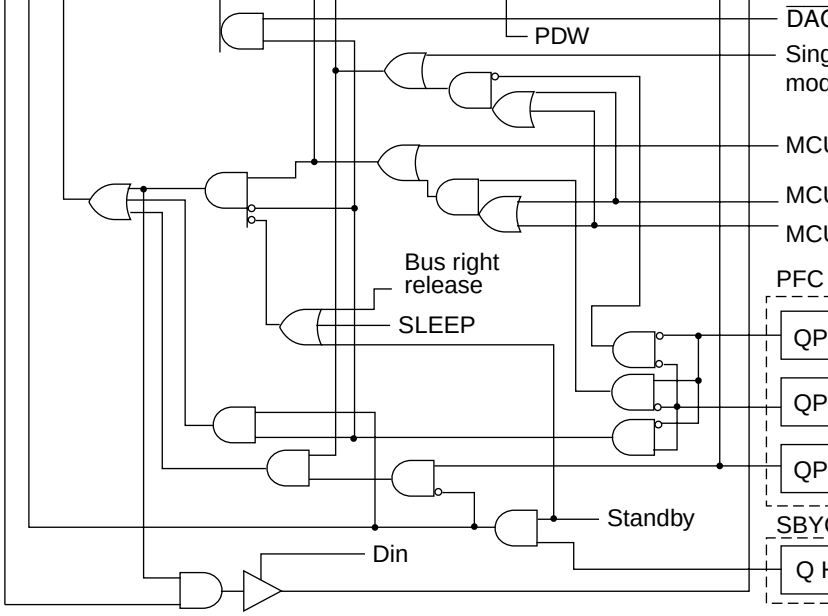
Figure B.26 PDn/Dn/$\overline{\text{IRQm}}$ Block Diagram (n = 16–23)



n = 24,25
m = 0,1
PDR: Port D read signal
PDW: Port D write signal

RES: Reset signal
Dout: Data output timing signal
Din: Data bus input timing signal

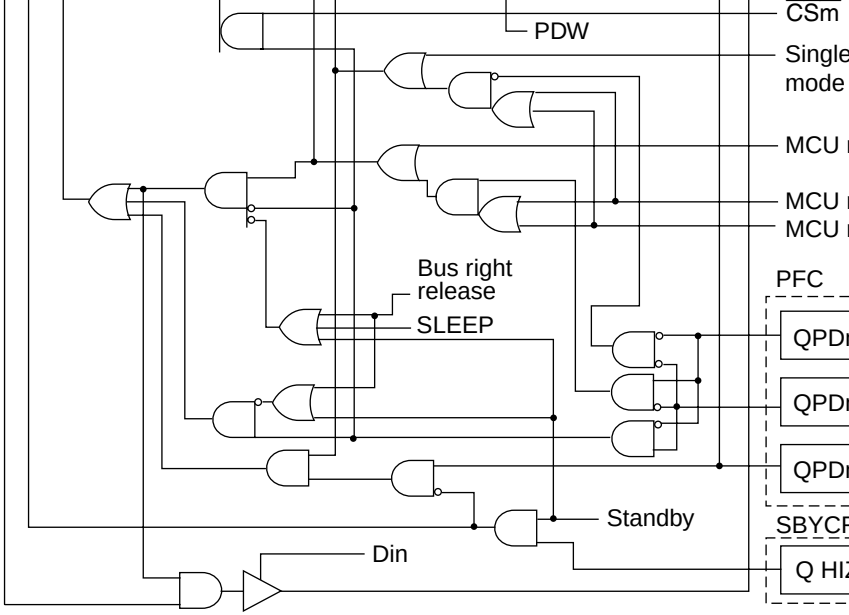
Figure B.27 PDn/Dn/$\overline{\text{DREQm}}$ Block Diagram (n = 24, 25)



n = 26,27
m = 0,1
PDR: Port D read signal
PDW: Port D write signal

RES: Reset signal
Dout: Data output timing signal
Din: Data bus input timing signal

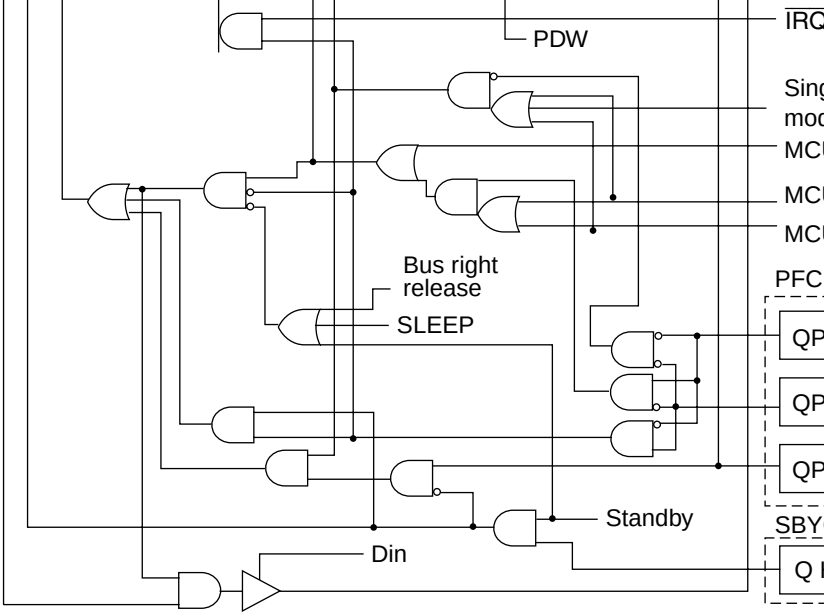
Figure B.28 PDn/Dn/$\overline{\text{DACKm}}$ Block Diagram (n = 26, 27)



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)



n = 30
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.30 PDn/Dn/$\overline{\text{IRQOUT}}$ Block Diagram (n = 30)

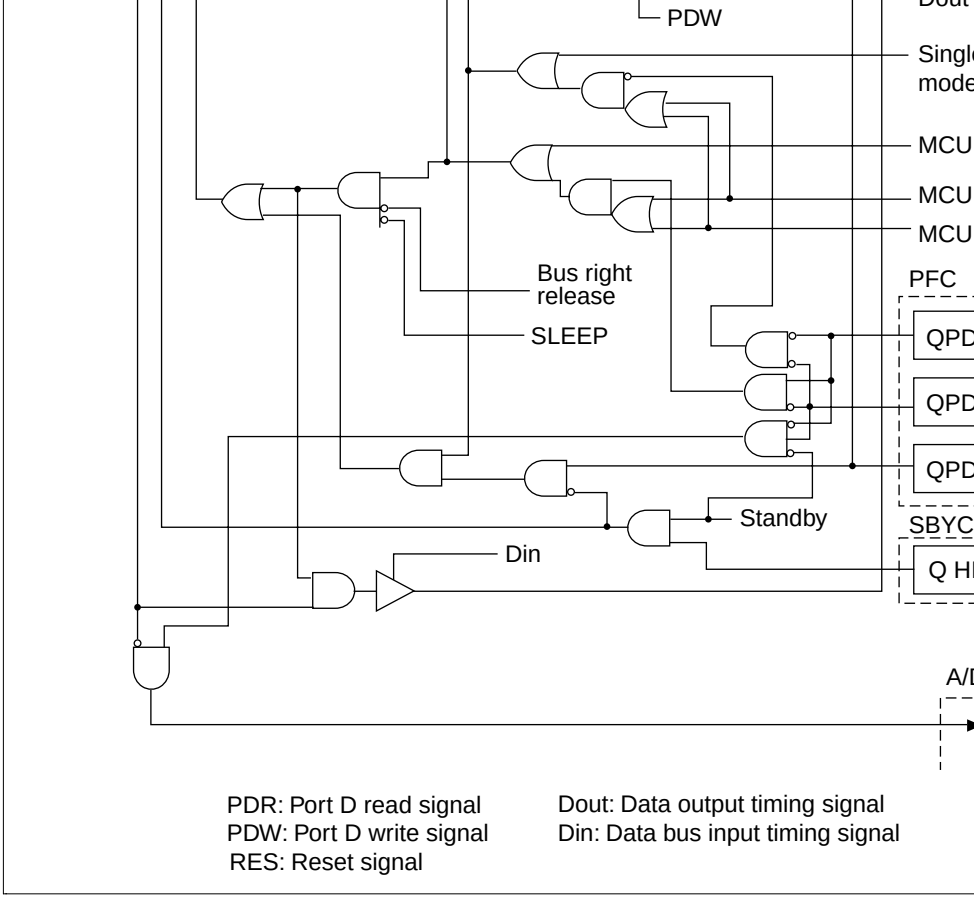
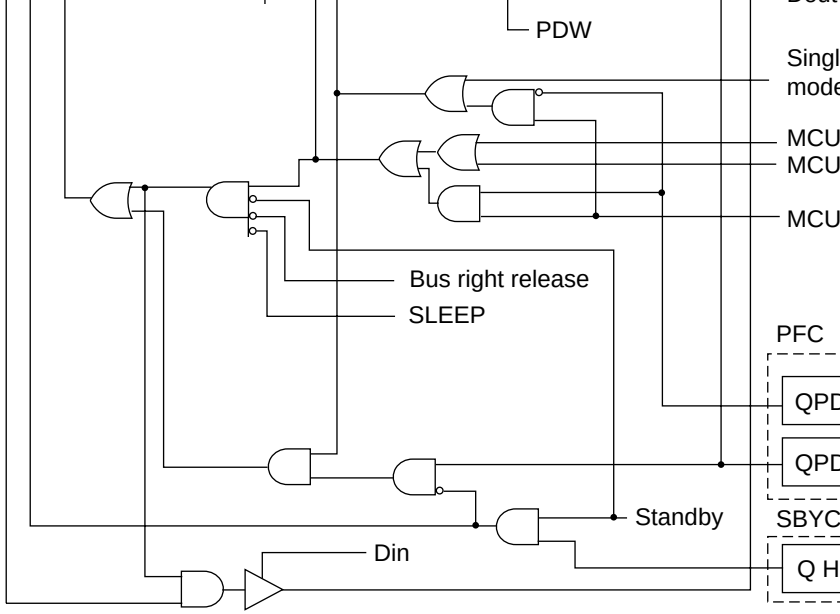


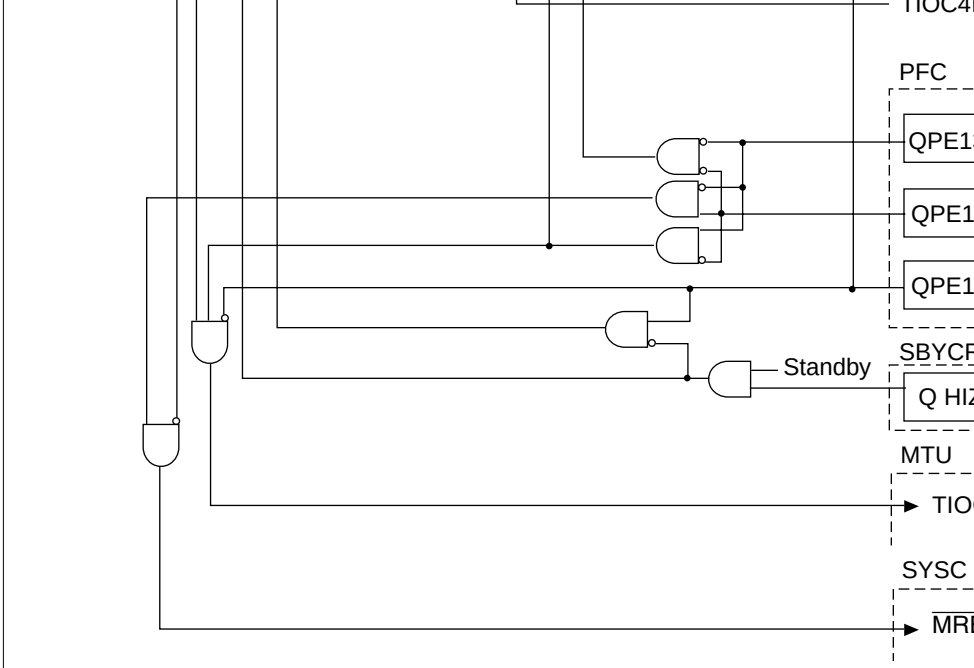
Figure B.31 PD31/D31/$\overline{\text{ADTRG}}$ Block Diagram



n = 8–15
PDR: Port D read signal
PDW: Port D write signal

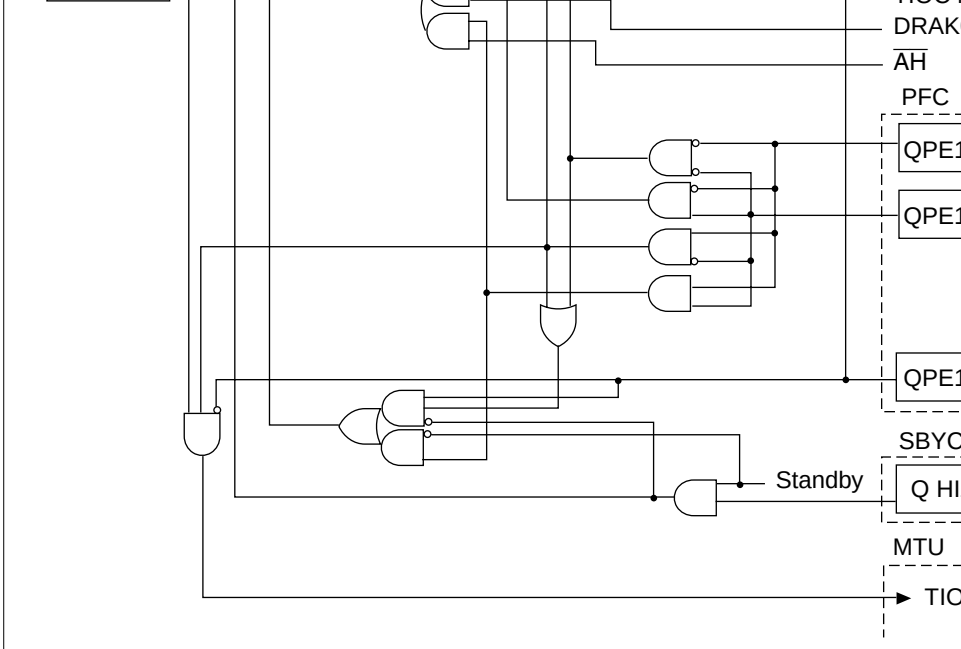
RES: Reset signal
Dout: Data output timing signal
Din: Data bus input timing signal

Figure B.32 PDn/Dn Block Diagram



PER: Port E read signal
PEW: Port E write signal
RES: Reset signal

Figure B.33 PE13/TIOC4B/$\overline{\text{MRES}}$ Block Diagram



PER: Port E read signal
PEW: Port E write signal
RES: Reset signal

Figure B.34 PE14/TIOC4C/DACK0/$\overline{\text{AH}}$ Block Diagram

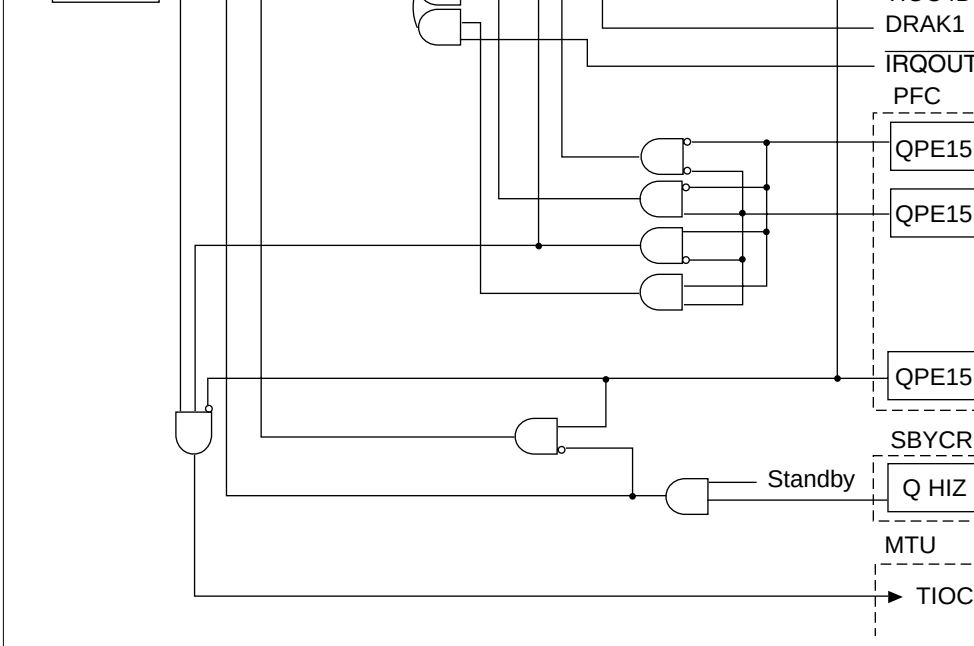


Figure B.35 PEn/TIOC4D/DACK1/$\overline{\text{IRQOUT}}$ Block Diagram

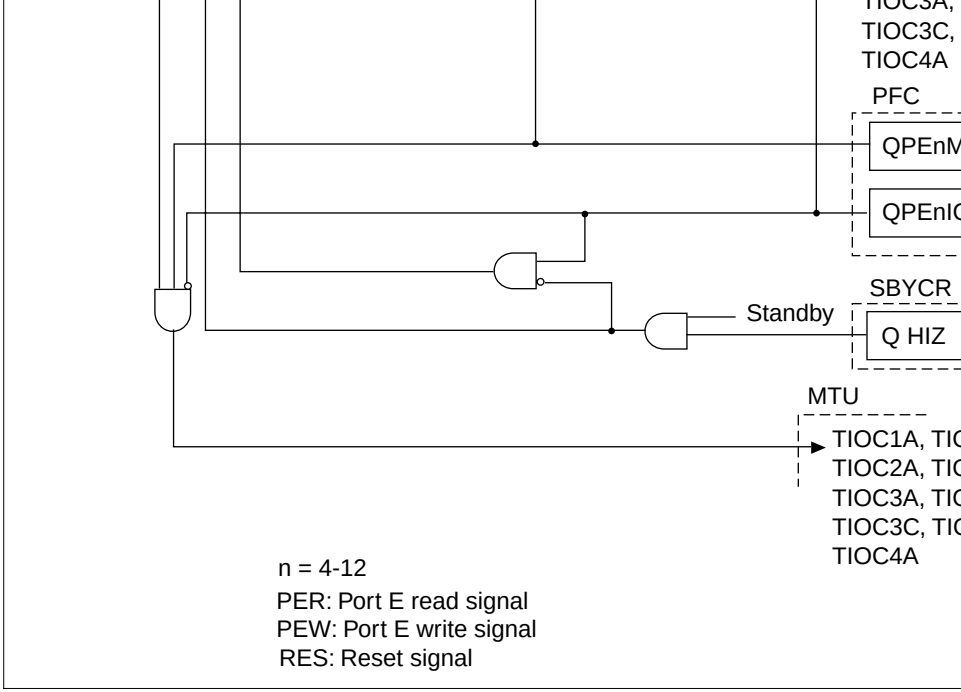
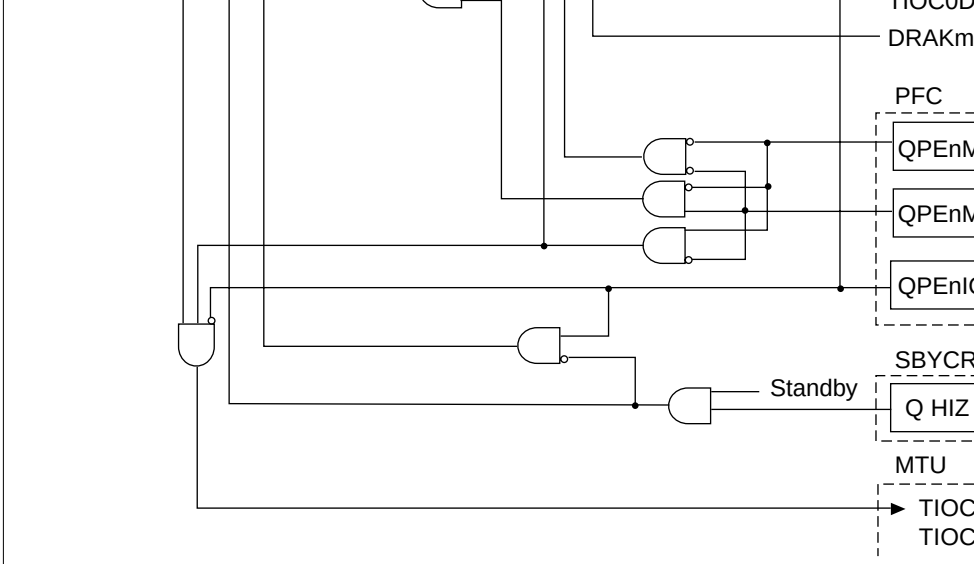


Figure B.36 PEn/TIOCXX Block Diagram



n = 1, 3
m = 0, 1
PER: Port E read signal
PEW: Port E write signal
RES: Reset signal

Figure B.37 PEn/TIOCXX/DRAKm Block Diagram

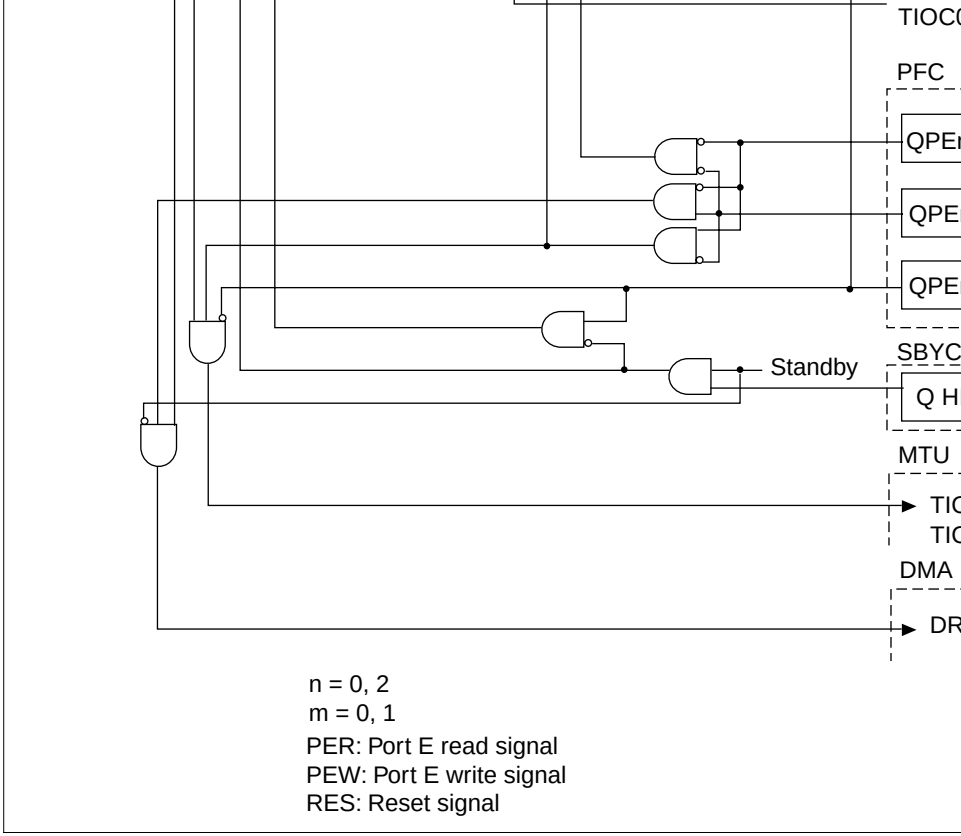


Figure B.38 PEn/TIOCXX/$\overline{\text{DREQm}}$ Block Diagram

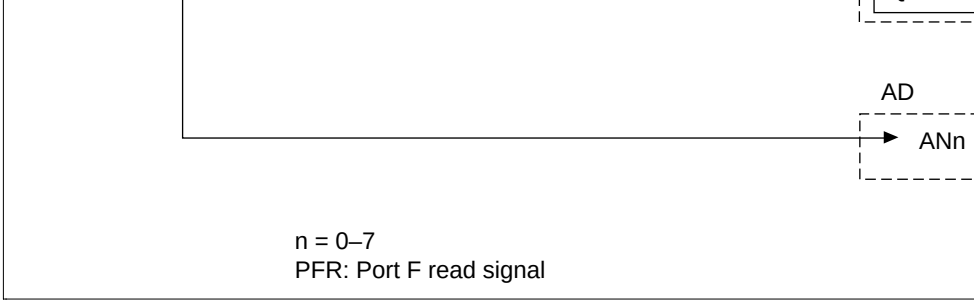


Figure B.39 PFn/ANn Block Diagram

	$\overline{\text{BREQ}}$	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

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	$\overline{ADTRG}$	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 occur simultaneously due to the timing between $\overline{BREQ}$ and internal operations cases, standby mode results, but the standby state may be different.

The initial pin states depend on the mode. See section 18, Pin Function Contr details.

2. I: Input, O: Output, H: High-level output, L: Low-level output, Z: High impedanc 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 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 multiplex them, are unstable during the $\overline{RES}$ setup time (t_{RESS}) immediately after the $\overline{RES}$ goes to low level.

	$\overline{\text{BACK}}$	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

converter control	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 occur simultaneously due to the timing between $\overline{\text{BREQ}}$ and internal operations cases, standby mode results, but the standby state may be different.

The initial pin states depend on the mode. See section 18, Pin Function Contr details.

2. I: Input, O: Output, H: High-level output, L: Low-level output, Z: High impedanc 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 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 multiplex them, are unstable during the $\overline{\text{RES}}$ setup time (t_{RESS}) immediately after the $\overline{\text{RES}}$ goes to low level.

$\overline{CASLL}$		H	H	H	H	H
RD/WR		H	H	H	H	H
$\overline{AH}$		L	L	L	L	L
$\overline{RD}$	R	H	H	H	H	H
	W	—	H	H	H	H
$\overline{WRHH}$	R	H	H	H	H	H
	W	—	H	H	H	H
$\overline{WRHL}$	R	H	H	H	H	H
	W	—	H	H	H	H
$\overline{WRLH}$	R	H	H	H	H	H
	W	—	H	H	H	H
$\overline{WRLL}$	R	H	H	H	H	H
	W	—	H	H	H	H
A21–A0		Address	Address	Address	Address	Address
D31–D24		High-Z	High-Z	High-Z	High-Z	High-Z
D23–D16		High-Z	High-Z	High-Z	High-Z	High-Z
D15–D8		High-Z	High-Z	High-Z	High-Z	High-Z
D7–D0		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.

$\overline{AH}$		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

RD/$\overline{WR}$		H	H	H	H	H	H
$\overline{AH}$		L	L	L	L	L	L
$\overline{RD}$	R	L	L	L	L	L	L
	W	H	H	H	H	H	H
$\overline{WRHH}$	R	H	H	H	H	H	H
	W	L	H	H	H	L	H
$\overline{WRHL}$	R	H	H	H	H	H	H
	W	H	L	H	H	L	H
$\overline{WRH}$	R	H	H	H	H	H	H
	W	H	H	L	H	H	L
$\overline{WRL}$	R	H	H	H	H	H	H
	W	H	H	H	L	H	L
A21–A0		Address	Address	Address	Address	Address	Address
D31–D24		Data	High-Z	High-Z	High-Z	Data	High-Z
D23–D16		High-Z	Data	High-Z	High-Z	Data	High-Z
D15–D8		High-Z	High-Z	Data	High-Z	High-Z	Data
D7–D0		High-Z	High-Z	High-Z	Data	High-Z	Data

Notes: 1. R: Read, W: Write

2. Valid: Chip select signal corresponding with accessed area is low; chip select other cases is high.

*1 L asserted in RAS down mode or refresh mode.

*2 L asserted in refresh mode.

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/I
D7–D0		Address/Data	Address	Address/Data	Address/I

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.

	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

$\overline{RD}/\overline{WR}$	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	A
D31–D24		Data	High-Z	High-Z	High-Z	Data	High-Z	D
D23–D16		High-Z	Data	High-Z	High-Z	Data	High-Z	D
D15–D8		High-Z	High-Z	Data	High-Z	High-Z	Data	D
D7–D0		High-Z	High-Z	High-Z	Data	High-Z	Data	D

Notes: 1. R: Read, W: Write

2. Valid: Chip select signal corresponding with accessed area is low; chip select s other cases is high.

*1 Asserted in RAS down mode or refresh mode.

*2 Asserted in refresh mode.

	running	
	1: Programming	1: Applicatio running

Note: This difference applies to all the F-ZTAT versions and all the mask-ROM versions different ROM size.

	ROM less version	A MASK	HD6417040AF28	HD6417040AF28	QFP2020-112	HD641
			HD6417040AVF16	HD6417040AVF16	QFP2020-112	HD641
			HD6417040AVX16	HD6417040AVX16	TQFP1414-120	HD641
			HD6417040ACF28	HD6417040ACF28	QFP2020-112Cu*1	HD641
			HD6417040AVCF16	HD6417040AVCF16	QFP2020-112Cu*1	HD641
SH7041A	Mask ROM version	A MASK	HD6437041AF28	HD6437041A(***)F28	QFP2020-144	HD643
			HD6437041AVF16	HD6437041A(***)VF16	QFP2020-144	HD643
			HD6437041ACF28	HD6437041A(***)CF28	QFP2020-144Cu*1	HD643
			HD6437041AVCF16	HD6437041A(***)VCF16	QFP2020-144Cu*1	HD643
	ROM less verion	A MASK	HD6417041AF28	HD6417041AF28	QFP2020-144	HD641
			HD6417041AVF16	HD6417041AVF16	QFP2020-144	HD641
			HD6417041ACF28	HD6417041ACF28	QFP2020-144Cu*1	HD641
			HD6417041AVCF16	HD6417041AVCF16	QFP2020-144Cu*1	HD641
SH7042	Mask ROM version	–	HD6437042F28	HD6437042 (***)F28	QFP2020-112	HD643
			HD6437042VF16	HD6437042 (***)VF16	QFP2020-112	HD643
	Z-TAT version	–	HD6477042F28	HD6477042F28	QFP2020-112	HD647
			HD6477042VF16	HD6477042VF16	QFP2020-112	HD647
SH7042A	Mask ROM version	A MASK	HD6437042AF28	HD6437042A(***)F28	QFP2020-112	HD643
			HD6437042AVF16	HD6437042A(***)VF16	QFP2020-112	HD643
			HD6437042AVX16	HD6437042A(***)VX16	TQFP1414-120	HD643
			HD6437042ACF28	HD6437042A(***)CF28	QFP2020-112Cu*1	HD643
			HD6437042AVCF16	HD6437042A(***)VCF16	QFP2020-112Cu*1	HD643

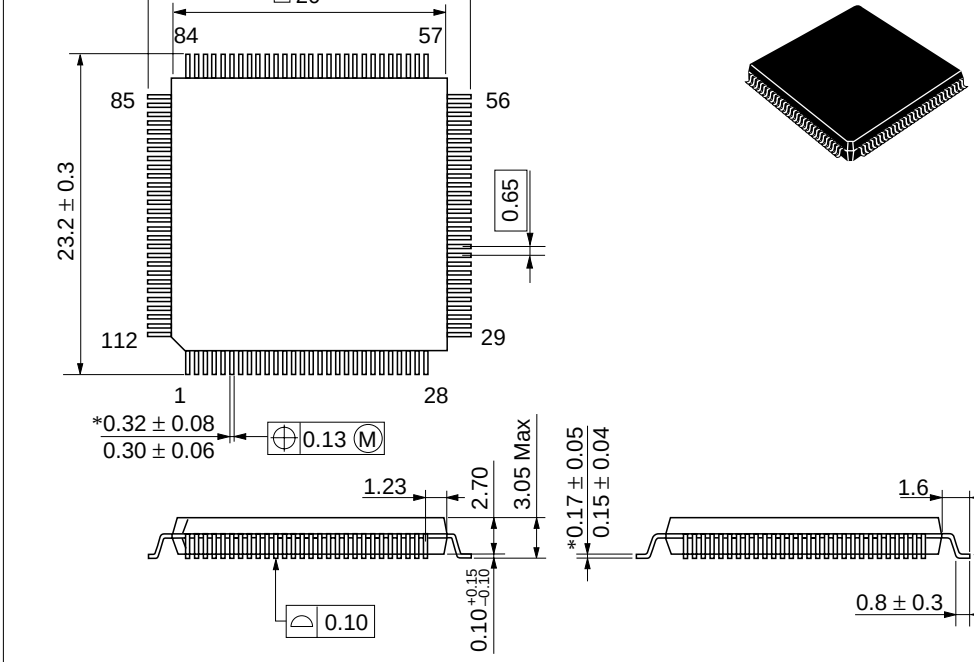
	version		HD6477043VF16	HD6477043VF16	QFP2020-144	HD647
SH7043A	Mask ROM version	A MASK	HD6437043AF28	HD6437043A(***)F28	QFP2020-144	HD643
			HD6437043AVF16	HD6437043A(***)VF16	QFP2020-144	HD643
			HD6437043ACF28	HD6437043A(***)CF28	QFP2020-144Cu*1	HD643
			HD6437043AVCF16	HD6437043A(***)VCF16	QFP2020-144Cu*1	HD643
	Z-TAT version	A MASK	HD6477043AF28	HD6477043AF28	QFP2020-144	HD647
			HD6477043AVF16	HD6477043AVF16	QFP2020-144	HD647
			HD6477043ACF28	HD6477043ACF28	QFP2020-144Cu*1	HD647
			HD6477043AVCF16	HD6477043AVCF16	QFP2020-144Cu*1	HD647
SH7044	Mask ROM version	A MASK	HD6437044F28	HD6437044(***)F28	QFP2020-112	HD643
	F-ZTAT version		HD64F7044F28	HD64F7044F28	QFP2020-112	HD64F
SH7045	Mask ROM version	A MASK	HD6437045F28	HD6437045(***)F28	QFP2020-144	HD643
	F-ZTAT version		HD64F7045F28	HD64F7045F28	QFP2020-144	HD64F

(***) 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-
number (ex. E00). The letter indicates the voltage and frequency, as shown l

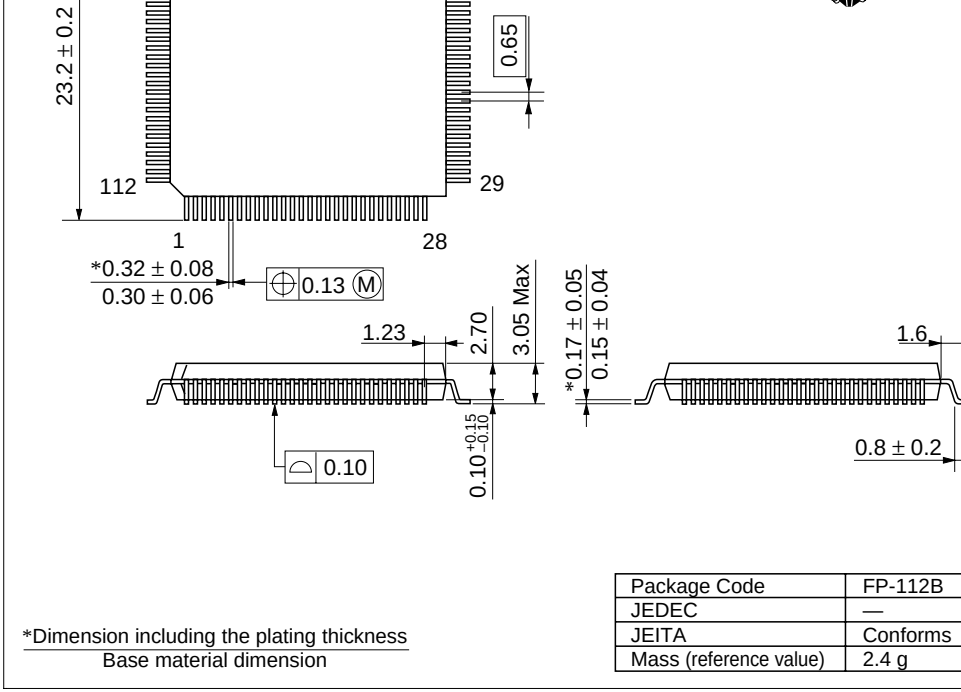
- E, F, G, H: 5.0 V, 28 MHz
- P, Q, R: 3.3 V, 16 MHz



Package Code	FP-112
JEDEC	—
JEITA	Conforms
Mass (reference value)	2.4 g

*Dimension including the plating thickness
Base material dimension

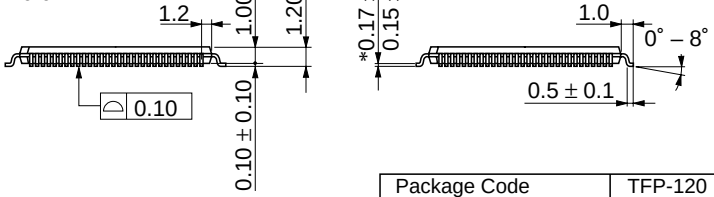
Figure F.1 Package Dimensions (FP-112)



Package Code	FP-112B
JEDEC	—
JEITA	Conforms
Mass (reference value)	2.4 g

*Dimension including the plating thickness
Base material dimension

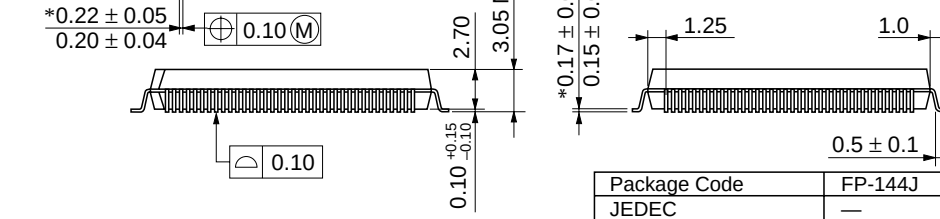
Figure F.2 Package Dimensions (FP-112B)



Package Code	TFP-120
JEDEC	—
JEITA	Conforms
Mass (reference value)	0.5 g

*Dimension including the plating thickness
Base material dimension

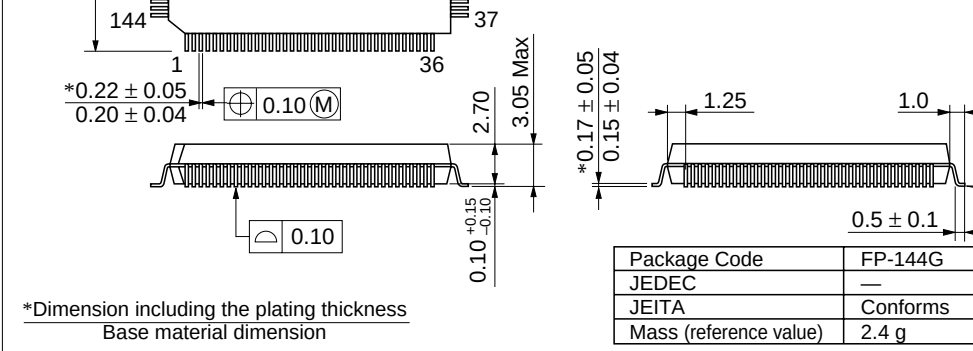
Figure F.3 Package Dimensions (TFP-120)



Package Code	FP-144J
JEDEC	—
JEITA	Conforms
Mass (reference value)	2.4 g

*Dimension including the plating thickness
Base material dimension

Figure F.4 Package Dimensions (FP-144J)



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 Gro
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 100-000

http://www.r

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

Hardware Manual

Renesas Electronics Corporation
1753, Shimonumabe, Nakahara-ku, Kawasaki-shi, Kanagawa 211-8668 Japan

REJ09B0044-0600O

X-ON Electronics

Largest Supplier of Electrical and Electronic Components

Click to view similar products for Digital Signal Processors & Controllers - DSP, DSC ***category:***

Click to view products by Renesas ***manufacturer:***

Other Similar products are found below :

0W633-001-XTP 0W888-002-XTP 42634-3 646890G 66AK2E05XABD25 66AK2G12ABY100 66AK2G12ABY60 66AK2G12ABYA100 66AK2G12ABYA100E 66AK2G12ABYA60 66AK2G12ABYA60E 66AK2G12ABYT100 66AK2H06DAAW2 66AK2H12BAAW2 66AK2H14DXAAWA24 AD21477WYCPZ1A02 AD21477WYSWZ1A02 AD21479WYSWZ2A02 AD21488WBSWZ2A02 AD21489WBCPZ402 AD21489WBSWZ402 AD21489WBSWZ4B02 AD21571WCSWZ400 AD21583WCBCZ4A10 AD21584WCBCZ4A10 ADBF592WYCPZ402 ADBF608WCBCZ502 ADBF608WCBCZ502RL ADBF700WCCPZ211 ADBF702WCCPZ411 ADBF703WCBCZ311 ADBF703WCBCZ411 ADBF704WCCPZ311 ADBF706WCCPZ411 ADBF707WCBCZ411 ADN8835CP-EVALZ ADSC571WCSWZ300 ADSC571WCSWZ400 ADSC571WCSWZ500 ADSC572WCBCZ302 ADSC572WCBCZ400 ADSC572WCBCZ402 ADSC572WCBCZ4200 ADSC572WCBCZ4202 ADSC573WCBCZ300 ADSC573WCBCZ400 ADSC573WCBCZ500 ADSC582WCBCZ4A10 ADSC583WCBCZ4A10 ADSC584WCBCZ3A10