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

16

H8S/2655 Group

Hardware Manual

Renesas 16-Bit Single-Chip Microcomputer H8S Family/H8S/2600 Series

H8S/2655 HD6432655
HD6472655
H8S/2653 HD6432653

Renesas Electronics
www.renesas.com

Rev.5.00 2006.09

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

2. Treatment of Unused Input Pins

Note: Fix all unused input pins to high or low level.
Generally, the input pins of CMOS products are high-impedance input pins. If
are in their open states, intermediate levels are induced by noise in the vicinity
through current flows internally, and a malfunction may occur.

3. Processing before Initialization

Note: When power is first supplied, the product's state is undefined.
The states of internal circuits are undefined until full power is supplied throug
chip and a low level is input on the reset pin. During the period where the stat
undefined, the register settings and the output state of each pin are also undefi
your system so that it does not malfunction because of processing while it is in
undefined state. For those products which have a reset function, reset the LSI
after the power supply has been turned on.

4. Prohibition of Access to Undefined or Reserved Addresses

Note: Access to undefined or reserved addresses is prohibited.
The undefined or reserved addresses may be used to expand functions, or test
may have been be allocated to these addresses. Do not access these registers; t
operation is not guaranteed if they are accessed.

on the high-level language C can also be run efficiently.

The address space is divided into eight areas. The data bus width and access states can for each of these areas, and various kinds of memory can be connected fast and easily.

On-chip memory consists of large-capacity ROM and RAM. PROM (ZTAT®*) and m versions are available, providing a quick and flexible response to conditions from ram through full-scale volume production, even for applications with frequently changing specifications.

On-chip supporting functions include a 16-bit timer pulse unit (TPU), programmable p generator (PPG), 8-bit timers, watchdog timer (WDT), serial communication interface converter, D/A converter, and I/O ports.

In addition, an on-chip DMA controller (DMAC) and data transfer controller (DTC) a enabling high-speed data transfer without CPU intervention.

Use of the H8S/2655 Group enables compact, high-performance systems to be implem easily.

This manual describes the hardware of the H8S/2655 Group. Refer to the H8S/2600 S H8S/2000 Series Software Manual for a detailed description of the instruction set.

Note: * ZTAT is a registered trademark of Renesas Technology Corp.

5.5.1 External Interrupts	102	IRQ7 to IRQ0 interrupts Description amended • Using ISCR, it is possible ... , at pins $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ}}$
5.4.6 Interrupt Exception Handling Sequence Figure 5.11 Interrupt Exception Handling	122	Description amended Internal data bus
7.2.4 DMA Control Register (CMACR)	227	Bit table amended Bit 6 (Before) DTID5 → (After) DTID
9.4.3 Pin Functions Table 9.7 Port 3 Pin Functions	373	Table 9.7 amended TxD_1 output pin* TxD_0 output pin*
10.2.3 Timer I/O Control Register (TIOR)	445	Bits 7 to 4—I/O Control B3 to B0 (IOB3 to IOB0) I/O Control D3 to D0 (IOD3 to IOD0) Channel 0 description amended

	Bit 7	Bit 6	Bit 5	Bit 4			
Channel	IOD3	IOD2	IOD1	IOD0	Description		
0	1	0	0	0	TGR0D is input capture register*2	Capture input source is TIOCD0 pin	Input capt
				1			Input capt
			1	*			Input capt
		1	*	*		Capture input source is channel 1/count clock	Input capt count-up/

	446	Channel 2 description amended

	Bit 7	Bit 6	Bit 5	Bit 4			
Channel	IOB3	IOB2	IOB1	IOB0	Description		
2	1	0	0	0	TGR2B is input capture register	Capture input source is TIOCB2 pin	Input capt
				1			Input capt
			1	*			Input capt

RENESAS

		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at
			1	0			1 output at
				1			Toggle outp match
	1	0	0	0	TGR0C is input capture register*1	Capture input source is TIOCC0 pin	Input captur
				1			Input captur
			1	*			Input captur
		1	*	*		Capture input source is channel 1/count clock	Input captur count-up/co

454 Channel 3 description amended

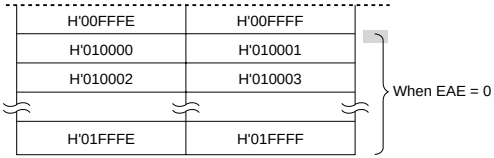
Channel	Bit 3 IOC3	Bit 2 IOC2	Bit 1 IOC1	Bit 0 IOC0	Description		
3	0	0	0	0	TGR3C is output compare register*1	Output disabled	
				1		Initial output is 0 output	0 output at
			1	0			1 output at
				1			Toggle outp match
		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at
			1	0			1 output at
				1			Toggle outp match
	1	0	0	0	TGR3C is input capture register*1	Capture input source is TIOCC3 pin	Input captur
				1			Input captur
			1	*			Input captur
		1	*	*		Capture input source is channel 4/count clock	Input captur count-up/co

14.2.6 Serial Control Register (SCR) 598 Bit 5—Overrun Error (ORER)

Description of setting condition amended

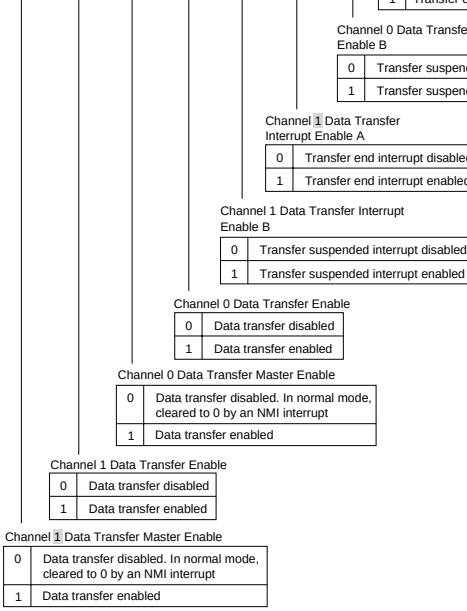
When the next serial reception is completed while RI

Bit Rate (bit/s)	n	N	Error (%)
31250	0	3	–7.84

15.2.3 Serial Mode Register (SMR)	656	Description added Bits 6 to 0—Operate in the same way as for the nor details, see section 14.2.5, Serial Mode Register (S
	657	Figure of TEND flag generation timing in transmissi deleted
16.6 Usage Notes	707	Usage notes deleted (Before) • If conversion is terminated ... 3. After terr 200 A/D reference clock cycles.) → (After) (deleted
19.1.1 Block Diagram Figure 19.1 Block Diagram of ROM (H8S/2655)	721	Figure 19.1 amended 
20.1.1 Block Diagram Figure 20.1 Block Diagram of Clock Pulse Generator	735	Figure 20.1 amended (Before) SCK1, SCK0 → (After) SCK2 to SCK0
B.1 Addresses	872	Table amended

Address (low)	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
H'FF38	SBYCR	SSBY	STS2	STS1	STS0	OPE	—	—	—
H'FF39	SYSCR	MACS	—	INTM1	INTM0	NMIEG	—	—	RAM
H'FF3A	SCKCR	PSTOP	—	—	—	—	SCK2	SCK1	SCK0
H'FF3B	MDCR	—	—	—	—	—	MDS2	MDS1	MDS
H'FF3C	MSTPCRH	MSTP15	MSTP14	MSTP13	MSTP12	MSTP11	MSTP10	MSTP9	MSTI
H'FF3D	MSTPCRL	MSTP7	MSTP6	MSTP5	MSTP4	MSTP3	MSTP2	MSTP1	MSTI

RENESAS



931 DTCERA to DTCERF H'FF30 to H'FF35 DTC

Correspondence between Interrupt Sources and DT

Table amended

	Bits					
Register	**7**	**6**	**5**	**4**	**3**	**2**
DTCERD	—	—	TGI5A	TGI5B	CMIA0	CMIB0

		Figure amended
		TGR Input Capture/Output Compare Flag A
		TGR Input Capture/Output Compare Flag B
	1010	TSR2 H'FFF5 TPU2
		Figure amended
		TGR Input Capture/Output Compare Flag A
		TGR Input Capture/Output Compare Flag B
Appendix G Package Dimensions Figure G.1 TFP-120 Package Dimensions	1058	Figure G.1 replaced
Figure G.2 FP-128 Package Dimensions	1059	Figure G.2 replaced

The H8S/2600 CPU has an internal 32-bit architecture, is provided with sixteen 16-bit registers and a concise, optimized instruction set designed for high-speed operation, a address a 16-Mbyte linear address space. The instruction set is upward-compatible wi and H8/300H CPU instructions at the object-code level, facilitating migration from th H8/300L, or H8/300H Series.

On-chip peripheral functions required for system configuration include DMA controll and data transfer controller (DTC) bus masters, ROM and RAM, a16-bit timer-pulse u programmable pulse generator (PPG), 8-bit timer, watchdog timer (WDT), serial com interface (SCI), A/D converter, D/A converter, and I/O ports.

The on-chip ROM is either PROM (ZTAT®*) or mask ROM, with a capacity of 128 ROM is connected to the CPU via a 16-bit data bus, enabling both byte and word data accessed in one state. Instruction fetching has been speeded up, and processing speed

Seven operating modes, modes 1 to 7, are provided, and there is a choice of address sp single-chip mode or external expansion mode.

The features of the H8S/2655 Group are shown in table 1.1.

Note: * ZTAT is a registered trademark of Renesas Technology Corp.

	— High-speed arithmetic operations 8/16/32-bit register-register add/subtract: 50 ns 16 × 16-bit register-register multiply: 200 ns 16 × 16 + 42-bit multiply and accumulate: 200 ns 32 ÷ 16-bit register-register divide: 1000 ns • Instruction set suitable for high-speed operation — Sixty-nine basic instructions — 8/16/32-bit move/arithmetic and logic instructions — Unsigned/signed multiply and divide instructions — Multiply-and accumulate instruction — Powerful bit-manipulation instructions • Two CPU operating modes — Normal mode: 64-kbyte address space — Advanced mode: 16-Mbyte address space
Bus controller	• Address space divided into 8 areas, with bus specifications set independently for each area • Chip select output possible for each area • Choice of 8-bit or 16-bit access space for each area • 2-state or 3-state access space can be designated for each are • Number of program wait states can be set for each area • Burst ROM directly connectable • Maximum 8-Mbyte DRAM or PSRAM directly connectable (or u interval timer possible) • External bus release function

Data transfer controller (DTC)	• Can be activated by internal interrupt or software • Multiple transfers or multiple types of transfer possible for one source • Transfer possible in repeat mode, block transfer mode, etc. • Request can be sent to CPU for interrupt that activated DTC
16-bit timer-pulse unit (TPU)	• 6-channel 16-bit timer on-chip • Pulse I/O processing capability for up to 16 pins • Automatic 2-phase encoder count capability
Programmable pulse generator (PPG)	• Maximum 16-bit pulse output possible with TPU as time base • Output trigger selectable in 4-bit groups • Non-overlap margin can be set • Direct output or inverse output setting possible
8-bit timer 2 channels	• 8-bit up-counter (external event count capability) • Two time constant registers • Two-channel connection possible
Watchdog timer	• Watchdog timer or interval timer selectable
Serial communication interface (SCI) 3 channels	• Asynchronous mode or synchronous mode selectable • Multiprocessor communication function • Smart card interface function
A/D converter	• Resolution: 10 bits • Input: 8 channels • High-speed conversion: 2.2 µs minimum conversion time (at 20-MHz operation) • Single or scan mode selectable • Sample and hold circuit • A/D conversion can be activated by external trigger or timer tr

H8S/2655	128 kbytes	4 kbytes
H8S/2653	64 kbytes	4 kbytes

Interrupt controller	• Nine external interrupt pins (NMI, $\overline{IRQ}_0$ to $\overline{IRQ}_7$) • 52 internal interrupt sources • Eight priority levels settable
Power-down state	• Medium-speed mode • Sleep mode • Module stop mode • Software standby mode • Hardware standby mode
Operating modes	Seven MCU operating modes

Mode	CPU Operating Mode	Description	On-Chip ROM	External Initial Value
1	Normal	On-chip ROM disabled expansion mode	Disabled	8 bits
2		On-chip ROM enabled expansion mode	Enabled	8 bits
3		Single-chip mode	Enabled	—
4	Advanced	On-chip ROM disabled expansion mode	Disabled	16 bits
5		On-chip ROM disabled expansion mode	Disabled	8 bits
6		On-chip ROM enabled expansion mode	Enabled	8 bits
7		Single-chip mode	Enabled	—

HD6472655TE	HD6472655VTE	PROM
HD6472655F	HD6472655VF	
HD6432655(***)TE	HD6432655(***)TE	Mask ROM
HD6432655(***)F	HD6432655(***)F	
HD6432653(***)TE	HD6432653(***)TE	
HD6432653(***)F	HD6432653(***)F	

Legend: Marked (***) is ROM code.

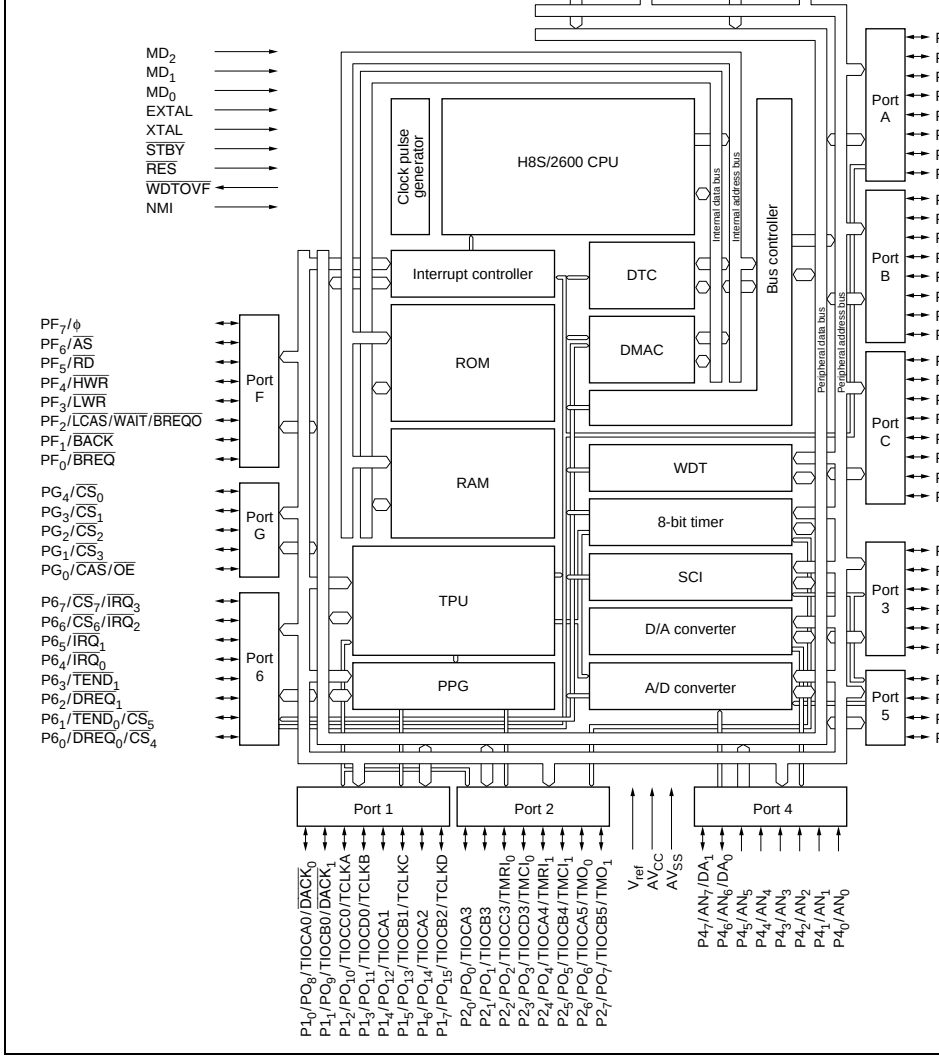


Figure 1.1 Block Diagram

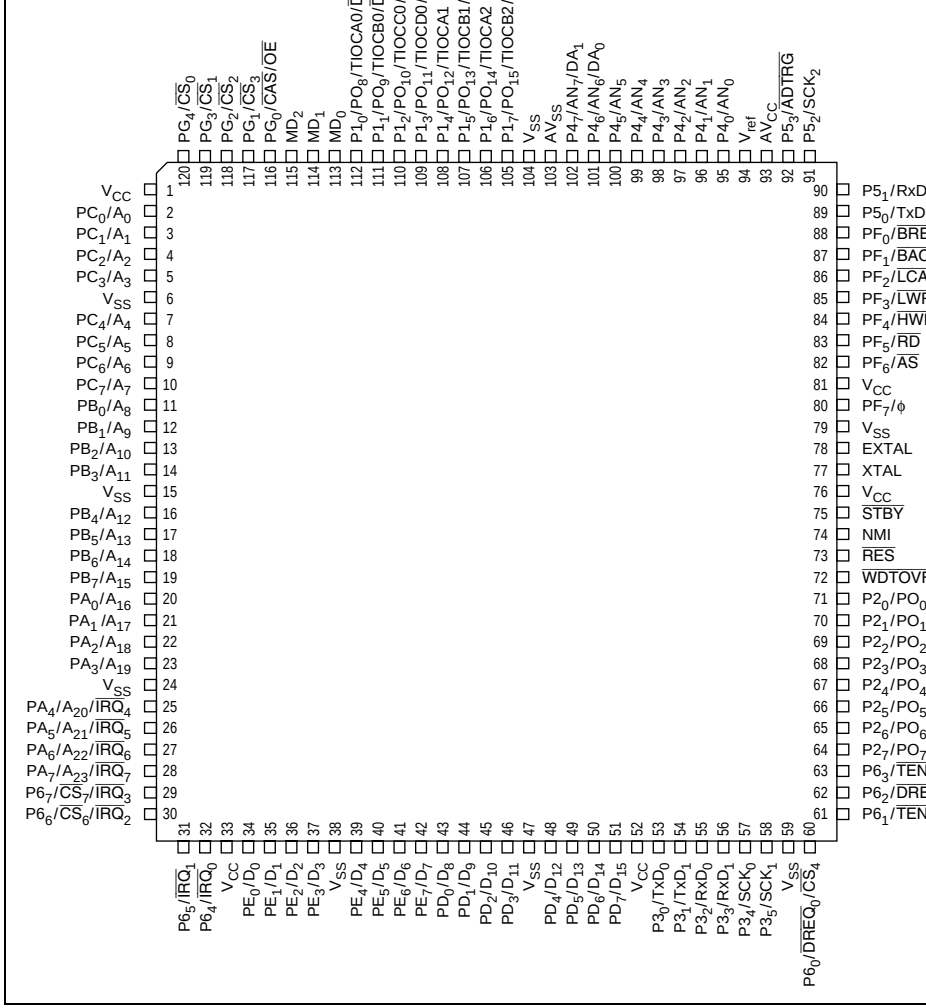


Figure 1.2 Pin Arrangement (TFP-120: Top View)

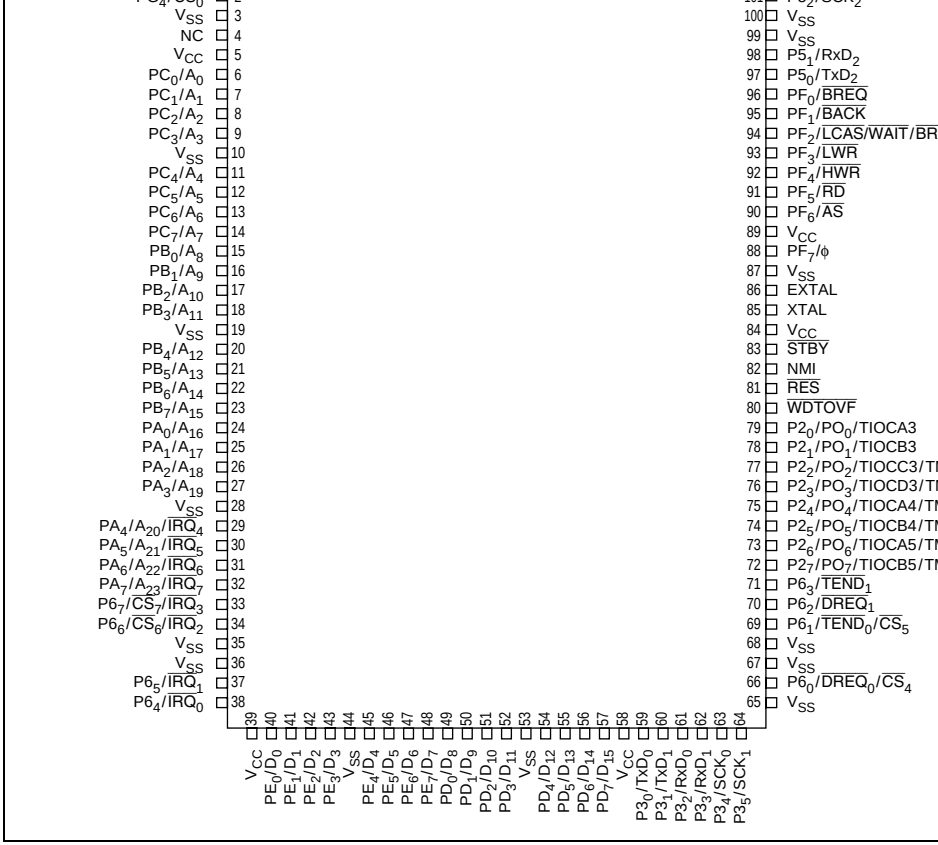


Figure 1.3 Pin Arrangement (FP-128: Top View)

2	6	A_0	PC_0/A_0	PC_0	A_0	A_0	PC_0/A_0	PC_0
3	7	A_1	PC_1/A_1	PC_1	A_1	A_1	PC_1/A_1	PC_1
4	8	A_2	PC_2/A_2	PC_2	A_2	A_2	PC_2/A_2	PC_2
5	9	A_3	PC_3/A_3	PC_3	A_3	A_3	PC_3/A_3	PC_3
6	10	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}
7	11	A_4	PC_4/A_4	PC_4	A_4	A_4	PC_4/A_4	PC_4
8	12	A_5	PC_5/A_5	PC_5	A_5	A_5	PC_5/A_5	PC_5
9	13	A_6	PC_6/A_6	PC_6	A_6	A_6	PC_6/A_6	PC_6
10	14	A_7	PC_7/A_7	PC_7	A_7	A_7	PC_7/A_7	PC_7
11	15	A_8	PB_0/A_8	PB_0	A_8	A_8	PB_0/A_8	PB_0
12	16	A_9	PB_1/A_9	PB_1	A_9	A_9	PB_1/A_9	PB_1
13	17	A_{10}	PB_2/A_{10}	PB_2	A_{10}	A_{10}	PB_2/A_{10}	PB_2
14	18	A_{11}	PB_3/A_{11}	PB_3	A_{11}	A_{11}	PB_3/A_{11}	PB_3
15	19	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}
16	20	A_{12}	PB_4/A_{12}	PB_4	A_{12}	A_{12}	PB_4/A_{12}	PB_4
17	21	A_{13}	PB_5/A_{13}	PB_5	A_{13}	A_{13}	PB_5/A_{13}	PB_5
18	22	A_{14}	PB_6/A_{14}	PB_6	A_{14}	A_{14}	PB_6/A_{14}	PB_6
19	23	A_{15}	PB_7/A_{15}	PB_7	A_{15}	A_{15}	PB_7/A_{15}	PB_7
20	24	PA_0	PA_0	PA_0	A_{16}	A_{16}	PA_0/A_{16}	PA_0
21	25	PA_1	PA_1	PA_1	A_{17}	A_{17}	PA_1/A_{17}	PA_1
22	26	PA_2	PA_2	PA_2	A_{18}	A_{18}	PA_2/A_{18}	PA_2
23	27	PA_3	PA_3	PA_3	A_{19}	A_{19}	PA_3/A_{19}	PA_3
24	28	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}
25	29	$PA_4/\overline{IRQ}_4$	$PA_4/\overline{IRQ}_4$	$PA_4/\overline{IRQ}_4$	A_{20}	A_{20}	$PA_4/A_{20}/\overline{IRQ}_4$	$PA_4/$
26	30	$PA_5/\overline{IRQ}_5$	$PA_5/\overline{IRQ}_5$	$PA_5/\overline{IRQ}_5$	$PA_5/A_{21}/\overline{IRQ}_5$	$PA_5/A_{21}/\overline{IRQ}_5$	$PA_5/A_{21}/\overline{IRQ}_5$	$PA_5/$

30	34	$P6_6/\overline{IRQ_2}$	$P6_6/\overline{IRQ_2}$	$P6_6/\overline{IRQ_2}$	$P6_6/\overline{IRQ_2}/\overline{CS_6}$	$P6_6/\overline{IRQ_2}/\overline{CS_6}$	$P6_6/\overline{IRQ_2}/\overline{CS_6}$	$P6_6/\overline{IR}$
—	35	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}
—	36	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}
31	37	$P6_5/\overline{IRQ_1}$	$P6_5/\overline{IRQ_1}$	$P6_5/\overline{IRQ_1}$	$P6_5/\overline{IRQ_1}$	$P6_5/\overline{IRQ_1}$	$P6_5/\overline{IRQ_1}$	$P6_5/\overline{IR}$
32	38	$P6_4/\overline{IRQ_0}$	$P6_4/\overline{IRQ_0}$	$P6_4/\overline{IRQ_0}$	$P6_4/\overline{IRQ_0}$	$P6_4/\overline{IRQ_0}$	$P6_4/\overline{IRQ_0}$	$P6_4/\overline{IR}$
33	39	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}
34	40	PE_0/D_0	PE_0/D_0	PE_0	PE_0/D_0	PE_0/D_0	PE_0/D_0	PE_0
35	41	PE_1/D_1	PE_1/D_1	PE_1	PE_1/D_1	PE_1/D_1	PE_1/D_1	PE_1
36	42	PE_2/D_2	PE_2/D_2	PE_2	PE_2/D_2	PE_2/D_2	PE_2/D_2	PE_2
37	43	PE_3/D_3	PE_3/D_3	PE_3	PE_3/D_3	PE_3/D_3	PE_3/D_3	PE_3
38	44	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}
39	45	PE_4/D_4	PE_4/D_4	PE_4	PE_4/D_4	PE_4/D_4	PE_4/D_4	PE_4
40	46	PE_5/D_5	PE_5/D_5	PE_5	PE_5/D_5	PE_5/D_5	PE_5/D_5	PE_5
41	47	PE_6/D_6	PE_6/D_6	PE_6	PE_6/D_6	PE_6/D_6	PE_6/D_6	PE_6
42	48	PE_7/D_7	PE_7/D_7	PE_7	PE_7/D_7	PE_7/D_7	PE_7/D_7	PE_7
43	49	D_8	D_8	PD_0	D_8	D_8	D_8	PD_0
44	50	D_9	D_9	PD_1	D_9	D_9	D_9	PD_1
45	51	D_{10}	D_{10}	PD_2	D_{10}	D_{10}	D_{10}	PD_2
46	52	D_{11}	D_{11}	PD_3	D_{11}	D_{11}	D_{11}	PD_3
47	53	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}
48	54	D_{12}	D_{12}	PD_4	D_{12}	D_{12}	D_{12}	PD_4
49	55	D_{13}	D_{13}	PD_5	D_{13}	D_{13}	D_{13}	PD_5
50	56	D_{14}	D_{14}	PD_6	D_{14}	D_{14}	D_{14}	PD_6
51	57	D_{15}	D_{15}	PD_7	D_{15}	D_{15}	D_{15}	PD_7
52	58	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}
53	59	$P3_0/TxD_0$	$P3_0/TxD_0$	$P3_0/TxD_0$	$P3_0/TxD_0$	$P3_0/TxD_0$	$P3_0/TxD_0$	$P3_0/Tx$
54	60	$P3_1/TxD_1$	$P3_1/TxD_1$	$P3_1/TxD_1$	$P3_1/TxD_1$	$P3_1/TxD_1$	$P3_1/TxD_1$	$P3_1/Tx$

60	66	P6$_0$/ $\overline{\text{DREQ}}_0$	P6$_0$/ $\overline{\text{DREQ}}_0$	P6$_0$/ $\overline{\text{DREQ}}_0$	P6$_0$/ $\overline{\text{DREQ}}_0$/ $\overline{\text{CS}}_4$	P6$_0$/ $\overline{\text{DREQ}}_0$/ $\overline{\text{CS}}_4$	P6$_0$/ $\overline{\text{DREQ}}_0$/ $\overline{\text{CS}}_4$	P6$_0$/ $\overline{\text{DRE}}$
—	67	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}
—	68	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}
61	69	P6$_1$/ $\overline{\text{TEND}}_0$	P6$_1$/ $\overline{\text{TEND}}_0$	P6$_1$/ $\overline{\text{TEND}}_0$	P6$_1$/ $\overline{\text{TEND}}_0$/ $\overline{\text{CS}}_5$	P6$_1$/ $\overline{\text{TEND}}_0$/ $\overline{\text{CS}}_5$	P6$_1$/ $\overline{\text{TEND}}_0$/ $\overline{\text{CS}}_5$	P6$_1$/ $\overline{\text{TEN}}$
62	70	P6$_2$/ $\overline{\text{DREQ}}_1$	P6$_2$/ $\overline{\text{DREQ}}_1$	P6$_2$/ $\overline{\text{DREQ}}_1$	P6$_2$/ $\overline{\text{DREQ}}_1$	P6$_2$/ $\overline{\text{DREQ}}_1$	P6$_2$/ $\overline{\text{DREQ}}_1$	P6$_2$/ $\overline{\text{DRE}}$
63	71	P6$_3$/ $\overline{\text{TEND}}_1$	P6$_3$/ $\overline{\text{TEND}}_1$	P6$_3$/ $\overline{\text{TEND}}_1$	P6$_3$/ $\overline{\text{TEND}}_1$	P6$_3$/ $\overline{\text{TEND}}_1$	P6$_3$/ $\overline{\text{TEND}}_1$	P6$_3$/ $\overline{\text{TEN}}$
64	72	P2$_7$/PO$_7$/ TIOCB5/ TMO$_1$	P2$_7$/PO$_7$/ TIOCB5/ TMO$_1$	P2$_7$/PO$_7$/ TIOCB5/ TMO$_1$	P2$_7$/PO$_7$/ TIOCB5/ TMO$_1$	P2$_7$/PO$_7$/ TIOCB5/ TMO$_1$	P2$_7$/PO$_7$/ TIOCB5/ TMO$_1$	P2$_7$/I TIOC TMC
65	73	P2$_6$/PO$_6$/ TIOCA5/ TMO$_0$	P2$_6$/PO$_6$/ TIOCA5/ TMO$_0$	P2$_6$/PO$_6$/ TIOCA5/ TMO$_0$	P2$_6$/PO$_6$/ TIOCA5/ TMO$_0$	P2$_6$/PO$_6$/ TIOCA5/ TMO$_0$	P2$_6$/PO$_6$/ TIOCA5/ TMO$_0$	P2$_6$/I TIOC TMC
66	74	P2$_5$/PO$_5$/ TIOCB4/ TMCI$_1$	P2$_5$/PO$_5$/ TIOCB4/ TMCI$_1$	P2$_5$/PO$_5$/ TIOCB4/ TMCI$_1$	P2$_5$/PO$_5$/ TIOCB4/ TMCI$_1$	P2$_5$/PO$_5$/ TIOCB4/ TMCI$_1$	P2$_5$/PO$_5$/ TIOCB4/ TMCI$_1$	P2$_5$/I TIOC TMC
67	75	P2$_4$/PO$_4$/ TIOCA4/ TMRI$_1$	P2$_4$/PO$_4$/ TIOCA4/ TMRI$_1$	P2$_4$/PO$_4$/ TIOCA4/ TMRI$_1$	P2$_4$/PO$_4$/ TIOCA4/ TMRI$_1$	P2$_4$/PO$_4$/ TIOCA4/ TMRI$_1$	P2$_4$/PO$_4$/ TIOCA4/ TMRI$_1$	P2$_4$/I TIOC TMR
68	76	P2$_3$/PO$_3$/ TIOCD3/ TMCI$_0$	P2$_3$/PO$_3$/ TIOCD3/ TMCI$_0$	P2$_3$/PO$_3$/ TIOCD3/ TMCI$_0$	P2$_3$/PO$_3$/ TIOCD3/ TMCI$_0$	P2$_3$/PO$_3$/ TIOCD3/ TMCI$_0$	P2$_3$/PO$_3$/ TIOCD3/ TMCI$_0$	P2$_3$/I TIOC TMC
69	77	P2$_2$/PO$_2$/ TIOCC3/ TMRI$_1$	P2$_2$/PO$_2$/ TIOCC3/ TMRI$_1$	P2$_2$/PO$_2$/ TIOCC3/ TMRI$_1$	P2$_2$/PO$_2$/ TIOCC3/ TMRI$_1$	P2$_2$/PO$_2$/ TIOCC3/ TMRI$_1$	P2$_2$/PO$_2$/ TIOCC3/ TMRI$_1$	P2$_2$/I TIOC TMR
70	78	P2$_1$/PO$_1$/ TIOCB3	P2$_1$/PO$_1$/ TIOCB3	P2$_1$/PO$_1$/ TIOCB3	P2$_1$/PO$_1$/ TIOCB3	P2$_1$/PO$_1$/ TIOCB3	P2$_1$/PO$_1$/ TIOCB3	P2$_1$/I TIOC

76	84	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}
77	85	XTAL	XTAL	XTAL	XTAL	XTAL	XTAL	XTAL
78	86	EXTAL	EXTAL	EXTAL	EXTAL	EXTAL	EXTAL	EXTA
79	87	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}
80	88	PF_7/ϕ	PF_7/ϕ	PF_7/ϕ	PF_7/ϕ	PF_7/ϕ	PF_7/ϕ	PF_7/ϕ
81	89	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}
82	90	$\overline{AS}$	$\overline{AS}$	PF_6	$\overline{AS}$	$\overline{AS}$	$\overline{AS}$	PF_6
83	91	$\overline{RD}$	$\overline{RD}$	PF_5	$\overline{RD}$	$\overline{RD}$	$\overline{RD}$	PF_5
84	92	$\overline{HWR}$	$\overline{HWR}$	PF_4	$\overline{HWR}$	$\overline{HWR}$	$\overline{HWR}$	PF_4
85	93	$\overline{LWR}$	$\overline{LWR}$	PF_3	$\overline{LWR}$	$\overline{LWR}$	$\overline{LWR}$	PF_3
86	94	$PF_2/\overline{WAIT}/\overline{BREQO}$	$PF_2/\overline{WAIT}/\overline{BREQO}$	PF_2	$PF_2/\overline{LCAS}/\overline{WAIT}/\overline{BREQO}$	$PF_2/\overline{LCAS}/\overline{WAIT}/\overline{BREQO}$	$PF_2/\overline{LCAS}/\overline{WAIT}/\overline{BREQO}$	PF_2
87	95	$PF_1/\overline{BACK}$	$PF_1/\overline{BACK}$	PF_1	$PF_1/\overline{BACK}$	$PF_1/\overline{BACK}$	$PF_1/\overline{BACK}$	PF_1
88	96	$PF_0/\overline{BREQ}$	$PF_0/\overline{BREQ}$	PF_0	$PF_0/\overline{BREQ}$	$PF_0/\overline{BREQ}$	$PF_0/\overline{BREQ}$	PF_0
89	97	$P5_0/TxD_2$	$P5_0/TxD_2$	$P5_0/TxD_2$	$P5_0/TxD_2$	$P5_0/TxD_2$	$P5_0/TxD_2$	$P5_0/T$
90	98	$P5_1/RxD_2$	$P5_1/RxD_2$	$P5_1/RxD_2$	$P5_1/RxD_2$	$P5_1/RxD_2$	$P5_1/RxD_2$	$P5_1/R$
—	99	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}
—	100	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}
91	101	$P5_2/SCK_2$	$P5_2/SCK_2$	$P5_2/SCK_2$	$P5_2/SCK_2$	$P5_2/SCK_2$	$P5_2/SCK_2$	$P5_2/S$
92	102	$P5_3/\overline{ADTRG}$	$P5_3/\overline{ADTRG}$	$P5_3/\overline{ADTRG}$	$P5_3/\overline{ADTRG}$	$P5_3/\overline{ADTRG}$	$P5_3/\overline{ADTRG}$	$P5_3/\overline{ADTR}$
93	103	AV_{CC}	AV_{CC}	AV_{CC}	AV_{CC}	AV_{CC}	AV_{CC}	AV_{CC}
94	104	V_{ref}	V_{ref}	V_{ref}	V_{ref}	V_{ref}	V_{ref}	V_{ref}
95	105	$P4_0/AN_0$	$P4_0/AN_0$	$P4_0/AN_0$	$P4_0/AN_0$	$P4_0/AN_0$	$P4_0/AN_0$	$P4_0/A$
96	106	$P4_1/AN_1$	$P4_1/AN_1$	$P4_1/AN_1$	$P4_1/AN_1$	$P4_1/AN_1$	$P4_1/AN_1$	$P4_1/A$
97	107	$P4_2/AN_2$	$P4_2/AN_2$	$P4_2/AN_2$	$P4_2/AN_2$	$P4_2/AN_2$	$P4_2/AN_2$	$P4_2/A$
98	108	$P4_3/AN_3$	$P4_3/AN_3$	$P4_3/AN_3$	$P4_3/AN_3$	$P4_3/AN_3$	$P4_3/AN_3$	$P4_3/A$
99	109	$P4_4/AN_4$	$P4_4/AN_4$	$P4_4/AN_4$	$P4_4/AN_4$	$P4_4/AN_4$	$P4_4/AN_4$	$P4_4/A$

104	114	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}
105	115	$P1_7/PO_{15}$/ TIOCB2/ TCLKD	$P1_7/PO_{15}$/ TIOCB2/ TCLKD	$P1_7/PO_{15}$/ TIOCB2/ TCLKD	$P1_7/PO_{15}$/ TIOCB2/ TCLKD	$P1_7/PO_{15}$/ TIOCB2/ TCLKD	$P1_7/PO_{15}$/ TIOCB2/ TCLKD	P1$_7$/ TIOC TCL
106	116	$P1_6/PO_{14}$/ TIOCA2	$P1_6/PO_{14}$/ TIOCA2	$P1_6/PO_{14}$/ TIOCA2	$P1_6/PO_{14}$/ TIOCA2	$P1_6/PO_{14}$/ TIOCA2	$P1_6/PO_{14}$/ TIOCA2	P1$_6$/ TIOC
107	117	$P1_5/PO_{13}$/ TIOCB1/ TCLKC	$P1_5/PO_{13}$/ TIOCB1/ TCLKC	$P1_5/PO_{13}$/ TIOCB1/ TCLKC	$P1_5/PO_{13}$/ TIOCB1/ TCLKC	$P1_5/PO_{13}$/ TIOCB1/ TCLKC	$P1_5/PO_{13}$/ TIOCB1/ TCLKC	P1$_5$/ TIOC TCL
108	118	$P1_4/PO_{12}$/ TIOCA1	$P1_4/PO_{12}$/ TIOCA1	$P1_4/PO_{12}$/ TIOCA1	$P1_4/PO_{12}$/ TIOCA1	$P1_4/PO_{12}$/ TIOCA1	$P1_4/PO_{12}$/ TIOCA1	P1$_4$/ TIOC
109	119	$P1_3/PO_{11}$/ TIOCD0/ TCLKB	$P1_3/PO_{11}$/ TIOCD0/ TCLKB	$P1_3/PO_{11}$/ TIOCD0/ TCLKB	$P1_3/PO_{11}$/ TIOCD0/ TCLKB	$P1_3/PO_{11}$/ TIOCD0/ TCLKB	$P1_3/PO_{11}$/ TIOCD0/ TCLKB	P1$_3$/ TIOC TCL
110	120	$P1_2/PO_{10}$/ TIOCC0/ TCLKA	$P1_2/PO_{10}$/ TIOCC0/ TCLKA	$P1_2/PO_{10}$/ TIOCC0/ TCLKA	$P1_2/PO_{10}$/ TIOCC0/ TCLKA	$P1_2/PO_{10}$/ TIOCC0/ TCLKA	$P1_2/PO_{10}$/ TIOCC0/ TCLKA	P1$_2$/ TIOC TCL
111	121	$P1_1/PO_9$/ TIOCB0/ $\overline{DACK}_1$	$P1_1/PO_9$/ TIOCB0/ $\overline{DACK}_1$	$P1_1/PO_9$/ TIOCB0/ $\overline{DACK}_1$	$P1_1/PO_9$/ TIOCB0/ $\overline{DACK}_1$	$P1_1/PO_9$/ TIOCB0/ $\overline{DACK}_1$	$P1_1/PO_9$/ TIOCB0/ $\overline{DACK}_1$	P1$_1$/ TIOC $\overline{DAC}$
112	122	$P1_0/PO_8$/ TIOCA0/ $\overline{DACK}_0$	$P1_0/PO_8$/ TIOCA0/ $\overline{DACK}_0$	$P1_0/PO_8$/ TIOCA0/ $\overline{DACK}_0$	$P1_0/PO_8$/ TIOCA0/ $\overline{DACK}_0$	$P1_0/PO_8$/ TIOCA0/ $\overline{DACK}_0$	$P1_0/PO_8$/ TIOCA0/ $\overline{DACK}_0$	P1$_0$/ TIOC $\overline{DAC}$
113	123	MD_0	MD_0	MD_0	MD_0	MD_0	MD_0	MD_0
114	124	MD_1	MD_1	MD_1	MD_1	MD_1	MD_1	MD_1
115	125	MD_2	MD_2	MD_2	MD_2	MD_2	MD_2	MD_2
116	126	PG_0	PG_0	PG_0	$PG_0/\overline{CAS}$/ $\overline{OE}$	$PG_0/\overline{CAS}$/ $\overline{OE}$	$PG_0/\overline{CAS}$/ $\overline{OE}$	PG_0
117	127	PG_1	PG_1	PG_1	$PG_1/\overline{CS}_3$	$PG_1/\overline{CS}_3$	$PG_1/\overline{CS}_3$	PG_1
118	128	PG_2	PG_2	PG_2	$PG_2/\overline{CS}_2$	$PG_2/\overline{CS}_2$	$PG_2/\overline{CS}_2$	PG_2
119	1	PG_3	PG_3	PG_3	$PG_3/\overline{CS}_1$	$PG_3/\overline{CS}_1$	$PG_3/\overline{CS}_1$	PG_3
120	2	$PG_4/\overline{CS}_0$	$PG_4/\overline{CS}_0$	PG_4	$PG_4/\overline{CS}_0$	$PG_4/\overline{CS}_0$	$PG_4/\overline{CS}_0$	PG_4
—	3	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}
—	4	NC	NC	NC	NC	NC	NC	NC

Note: NC pins should be connected to V_{SS} or left open.

	V_{CC}	52, 76, 81	58, 84, 89		power supply. All V_{CC} pins connected to the system p supply.
	V_{SS}	6, 15, 24, 38, 47, 59, 79, 104	3, 10, 19, 28, 35, 36, 44, 53, 65, 67, 68, 87, 99, 100, 114	Input	Ground: For connection to (0 V). All V_{SS} pins should b connected to the system p supply (0 V).
Clock	XTAL	77	85	Input	Connects to a crystal osci See section 20, Clock Pul Generator, for typical conn diagrams for a crystal osc external clock input.
	EXTAL	78	86	Input	Connects to a crystal osci The EXTAL pin can also i external clock. See section 20, Clock Pul Generator, for typical conn diagrams for a crystal osc external clock input.
	ϕ	80	88	Output	System clock: Supplies th clock to an external device

MD2	MD1	MD0	Op Mo
0	0	0	—
		1	Mo
	1	0	Mo
		1	Mo
1	0	0	Mo
		1	Mo
	1	0	Mo
		1	Mo

System control	$\overline{\text{RES}}$	73	81	Input	Reset input: When this p low, the chip is reset. Th reset can be selected ac the NMI input level. At po NMI pin input level shoul high.
	$\overline{\text{STBY}}$	75	83	Input	Standby: When this pin is a transition is made to ha standby mode.
	$\overline{\text{BREQ}}$	88	96	Input	Bus request: Used by an master to issue a bus req H8S/2655 Group.
	$\overline{\text{BREQO}}$	86	94	Output	Bus request output: The request signal used when bus master accesses ext in the external bus-releas
	$\overline{\text{BACK}}$	87	95	Output	Bus request acknowledg that the bus has been rel external bus master.

Address bus	A_{23} to A_0	28 to 25, 23 to 16, 14 to 7, 5 to 2	32 to 29, 27 to 20, 18 to 11, 9 to 6	Output	Address bus: These pins o address.
Data bus	D_{15} to D_0	51 to 48, 46 to 39, 37 to 34	57 to 54, 52 to 45, 43 to 40	I/O	Data bus: These pins cons bidirectional data bus.
Bus control	$\overline{CS}_7$ to $\overline{CS}_0$	29, 30, 61, 60, 117 to 120	33, 34, 69, 66, 127, 128, 1, 2	Output	Chip select: Signals for se areas 7 to 0.
	$\overline{AS}$	82	90	Output	Address strobe: When this it indicates that address o the address bus is enable
	$\overline{RD}$	83	91	Output	Read: When this pin is lov indicates that the external space can be read.
	$\overline{HWR}$	84	92	Output	High write/write enable/up enable: A strobe signal that writes space and indicates that t half (D_{15} to D_8) of the data enabled. The 2CAS type DRAM wri signal. The 2WE type DRAM upp enable signal.

					lower column address st The 2WE type DRAM lov enable signal.
	$\overline{CAS}$/ $\overline{OE}$	116	126	Output	Upper column address s column address strobe/o enable/refresh: The 2CAS type DRAM u address strobe signal. The 2WE type DRAM co address strobe signal. The PSRAM output enab
	$\overline{LCAS}$	86	94	Output	Lower column address s The 2-CAS type (LCASS lower column address st
	$\overline{WAIT}$	86	94	Input	Wait: Requests insertion state in the bus cycle wh accessing external 3-stat space.
DMA controller (DMAC)	$\overline{DREQ}_1$, $\overline{DREQ}_0$	62, 60	70, 66	Input	DMA request 1 and 0: Th request DMAC activation
	$\overline{TEND}_1$, $\overline{TEND}_0$	63, 61	71, 69	Output	DMA transfer end 1 and pins indicate the end of D transfer.
	$\overline{DACK}_1$, $\overline{DACK}_0$	111, 112	121, 122	Output	DMA transfer acknowled These are the DMAC sin transfer acknowledge pin

	TIOCA1, TIOCB1	108, 107	118, 117	I/O	Input capture/ output comp A1 and B1: The TGR1A a input capture input or outp compare output, or PWM
	TIOCA2, TIOCB2	106, 105	116, 115	I/O	Input capture/ output comp A2 and B2: The TGR2A a input capture input or outp compare output, or PWM
	TIOCA3, TIOCB3, TIOCC3, TIOCD3	71 to 68	79 to 76	I/O	Input capture/ output comp A3 to D3: The TGR3A to input capture input or outp compare output, or PWM
	TIOCA4, TIOCB4	67, 66	75, 74	I/O	Input capture/ output comp A4 and B4: The TGR4A a input capture input or outp compare output, or PWM
	TIOCA5, TIOCB5	65, 64	73, 72	I/O	Input capture/ output comp A5 and B5: The TGR5A a input capture input or outp compare output, or PWM
Programmable pulse generator (PPG)	PO_{15} to PO_0	105 to 112, 64 to 71	115 to 122, 72 to 79	Output	Pulse output 15 to 0: Puls pins.
8-bit timer	TMO_0, TMO_1	65, 64	73, 72	Output	Compare match output: The compare match outpu
	$TMCI_0$, $TMCI_1$	68, 66	76, 74	Input	Counter external clock inp pins for the external clock counter.
	$TMRI_0$, $TMRI_1$	69, 67	77, 75	Input	Counter external reset inp counter reset input pins.

Smart Card interface	RxD_2, RxD_1, RxD_0	90, 56, 55	98, 62, 61	Input	Receive data (channel 0) Data input pins.
	SCK_2, SCK_1 SCK_0	91, 58 57	101, 64, 63	I/O	Serial clock (channel 0, 1 Clock I/O pins.
A/D converter	AN_7 to AN_0	102 to 95	112 to 105	Input	Analog 7 to 0: Analog inp
	$\overline{ADTRG}$	92	102	Input	A/D conversion external Pin for input of an extern start A/D conversion.
D/A converter	DA_1, DA_0	102, 101	112, 111	Output	Analog output: D/A conv output pins.
A/D converter and D/A converter	AV_{CC}	93	103	Input	This is the power supply A/D converter and D/A c When the A/D converter converter are not used, t should be connected to t power supply (+5 V).
	AV_{SS}	103	113	Input	This is the ground pin for converter and D/A conve When the A/D converter converter are not used, t should be connected to t power supply (0 V).
	V_{ref}	94	104	Input	This is the reference volt pin for the A/D converter converter. When the A/D converter converter are not used, t should be connected to t power supply (+5 V).

				register (P2DDR).
$P3_5$ to $P3_0$	58 to 53	64 to 59	I/O	Port 3: A 6-bit I/O port. Inp output can be designated by means of the port 3 dat register (P3DDR).
$P4_7$ to $P4_0$	102 to 95	112 to 105	Input	Port 4: An 8-bit input port.
$P5_3$ to $P5_0$	92 to 89	102, 101, 98, 97	I/O	Port 5: A 4-bit I/O port. Inp output can be designated by means of the port 5 dat register (P5DDR).
$P6_7$ to $P6_0$	29 to 32, 63 to 60	33, 34, 37, 38, 71 to 69, 66	I/O	Port 6: An 8-bit I/O port. In output can be designated by means of the port 6 dat register (P6DDR).
PA_7 to PA_0	28 to 25, 23 to 20	32 to 29, 27 to 24	I/O	Port A: An 8-bit I/O port. In output can be designated by means of the port A da register (PADDR).
PB_7 to PB_0	19 to 16, 14 to 11	23 to 20, 18 to 15	I/O	Port B: An 8-bit I/O port. In output can be designated by means of the port B da register (PBDDR).
PC_7 to PC_0	10 to 7, 5 to 2	14 to 11, 9 to 6	I/O	Port C: An 8-bit I/O port. In output can be designated by means of the port C da register (PCDDR).
PD_7 to PD_0	51 to 48, 46 to 43	57 to 54, 52 to 49	I/O	Port D: An 8-bit I/O port. In output can be designated by means of the port D da register (PDDDR).

				register (PFDDR).
PG_4 to PG_0	120 to 116	2, 1, 128 to 126	I/O	Port G: A 5-bit I/O port. I output can be designated by means of the port G d register (PGDDR).

ideal for realtime control.

2.1.1 Features

The H8S/2600 CPU has the following features.

- Upward-compatible with H8/300 and H8/300H CPUs
  - Can execute H8/300 and H8/300H object programs
- General-register architecture
  - Sixteen 16-bit general registers (also usable as sixteen 8-bit registers or eight 3 registers)
- Sixty-nine basic instructions
  - 8/16/32-bit arithmetic and logic instructions
  - Multiply and divide instructions
  - Powerful bit-manipulation instructions
  - Multiply-and-accumulate instruction
- Eight addressing modes
  - Register direct [Rn]
  - Register indirect [@ERn]
  - Register indirect with displacement [@(d:16,ERn) or @(d:32,ERn)]
  - Register indirect with post-increment or pre-decrement [@ERn+ or @–ERn]
  - Absolute address [@aa:8, @aa:16, @aa:24, or @aa:32]
  - Immediate [#xx:8, #xx:16, or #xx:32]
  - Program-counter relative [@(d:8,PC) or @(d:16,PC)]
  - Memory indirect [@@aa:8]
- 16-Mbyte address space
  - Program: 16 Mbytes
  - Data: 16 Mbytes (4 Gbytes architecturally)

— 32 ÷ 16-bit register-register divide: 1000 ns

- Two CPU operating modes
  - — Normal mode
  - — Advanced mode
- Power-down state
  - — Transition to power-down state by SLEEP instruction
  - — CPU clock speed selection

2.1.2 Differences from H8/300 CPU

In comparison to the H8/300 CPU, the H8S/2600 CPU has the following enhancements

- More general registers and control registers
  - — Eight 16-bit expanded registers, and one 8-bit and two 32-bit control registers, h added.
- Expanded address space
  - — Normal mode supports the same 64-kbyte address space as the H8/300 CPU.
  - — Advanced mode supports a maximum 16-Mbyte address space.
- Enhanced addressing
  - — The addressing modes have been enhanced to make effective use of the 16-Mby space.
- Enhanced instructions
  - — Addressing modes of bit-manipulation instructions have been enhanced.
  - — Signed multiply and divide instructions have been added.
  - — A multiply-and-accumulate instruction has been added.
  - — Two-bit shift instructions have been added.
  - — Instructions for saving and restoring multiple registers have been added.
  - — A test and set instruction has been added.

— One 8-bit and two 32-bit control registers have been added.

- Enhanced instructions
  — Addressing modes of bit-manipulation instructions have been enhanced.
  — A multiply-and-accumulate instruction has been added.
  — Two-bit shift instructions have been added.
  — Instructions for saving and restoring multiple registers have been added.
  — A test and set instruction has been added.
- Higher speed
  — Basic instructions execute twice as fast.

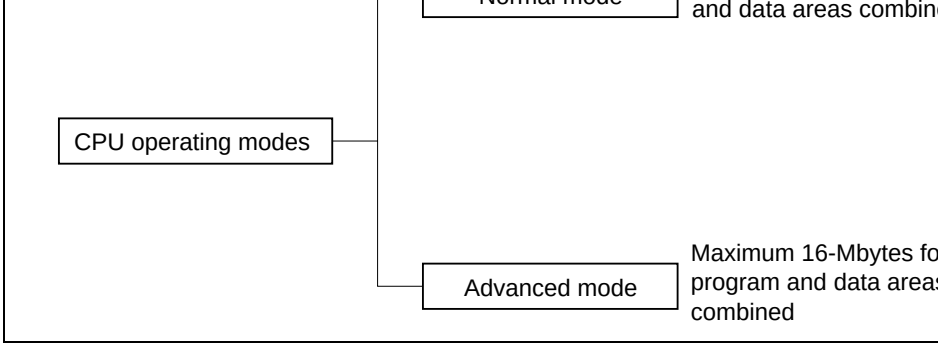


Figure 2.1 CPU Operating Modes

(1) Normal Mode

The exception vector table and stack have the same structure as in the H8/300 CPU.

Address Space: A maximum address space of 64 kbytes can be accessed.

Extended Registers (En): The extended registers (E0 to E7) can be used as 16-bit regi
the upper 16-bit segments of 32-bit registers. When En is used as a 16-bit register it can
any value, even when the corresponding general register (Rn) is used as an address regi
general register is referenced in the register indirect addressing mode with pre-decreme
or post-increment (@Rn+) and a carry or borrow occurs, however, the value in the corr
extended register (En) will be affected.

Instruction Set: All instructions and addressing modes can be used. Only the lower 16
effective addresses (EA) are valid.

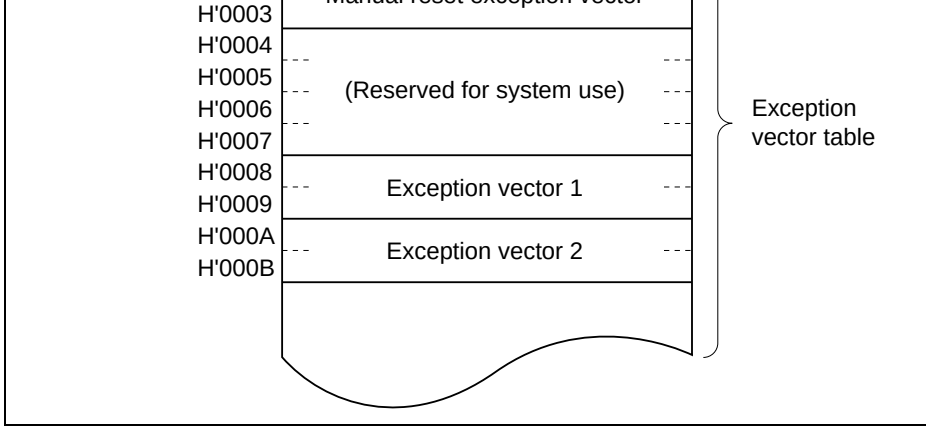
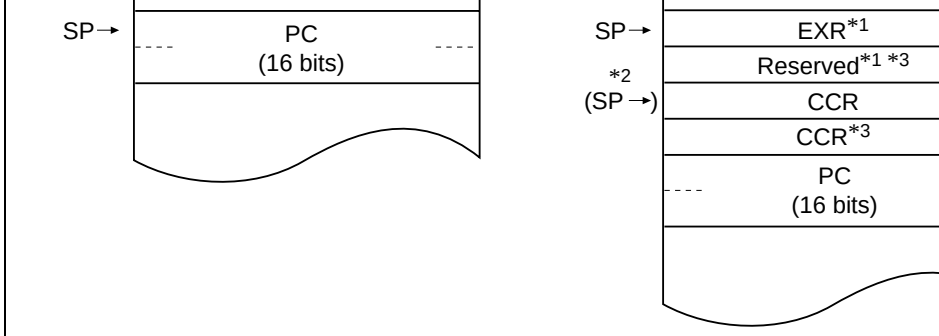


Figure 2.2 Exception Vector Table (Normal Mode)

The memory indirect addressing mode (@@aa:8) employed in the JMP and JSR instru an 8-bit absolute address included in the instruction code to specify a memory operand contains a branch address. In normal mode the operand is a 16-bit word operand, prov bit branch address. Branch addresses can be stored in the top area from H'0000 to H'00 that this area is also used for the exception vector table.



Notes: 1. When EXR is not used it is not stored on the stack.
2. SP when EXR is not used.
3. Ignored when returning.

Figure 2.3 Stack Structure in Normal Mode

(2) Advanced Mode

Address Space: Linear access is provided to a 16-Mbyte maximum address space (arch a maximum 16-Mbyte program area and a maximum 4-Gbyte data area, with a maximu Gbytes for program and data areas combined).

Extended Registers (En): The extended registers (E0 to E7) can be used as 16-bit regi the upper 16-bit segments of 32-bit registers or address registers.

Instruction Set: All instructions and addressing modes can be used.

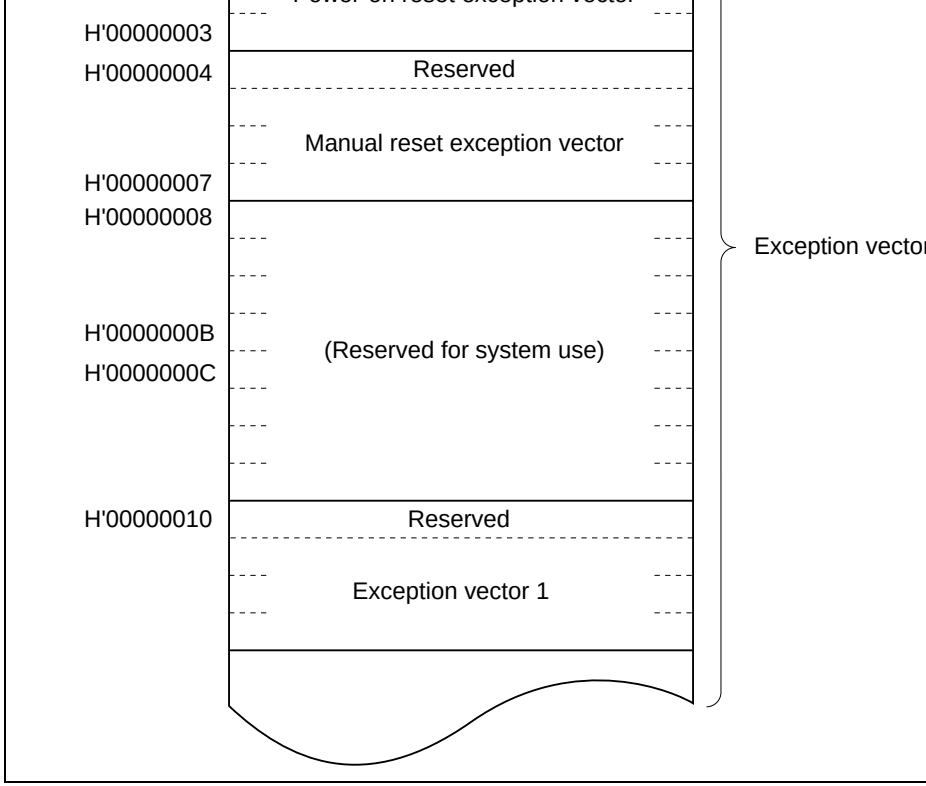
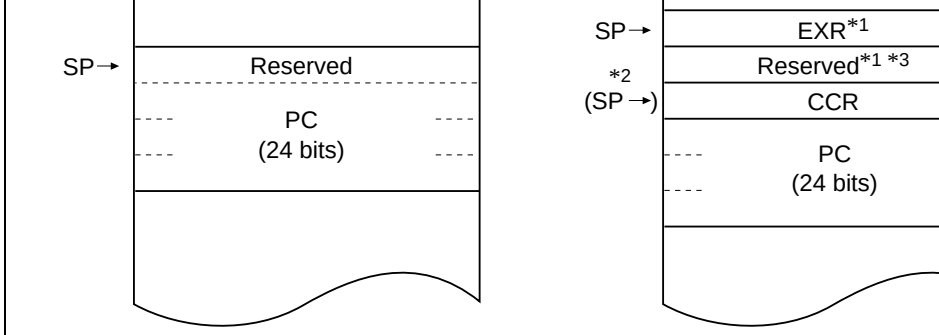


Figure 2.4 Exception Vector Table (Advanced Mode)

The memory indirect addressing mode (@@aa:8) employed in the JMP and JSR instru an 8-bit absolute address included in the instruction code to specify a memory operand contains a branch address. In advanced mode the operand is a 32-bit longword operan a 32-bit branch address. The upper 8 bits of these 32 bits are a reserved area that is reg H'00. Branch addresses can be stored in the area from H'00000000 to H'000000FF. No first part of this range is also the exception vector table.



(a) Subroutine Branch **(b) Exception Handlin**

Notes: 1. When EXR is not used it is not stored on the stack.
2. SP when EXR is not used.
3. Ignored when returning.

Figure 2.5 Stack Structure in Advanced Mode

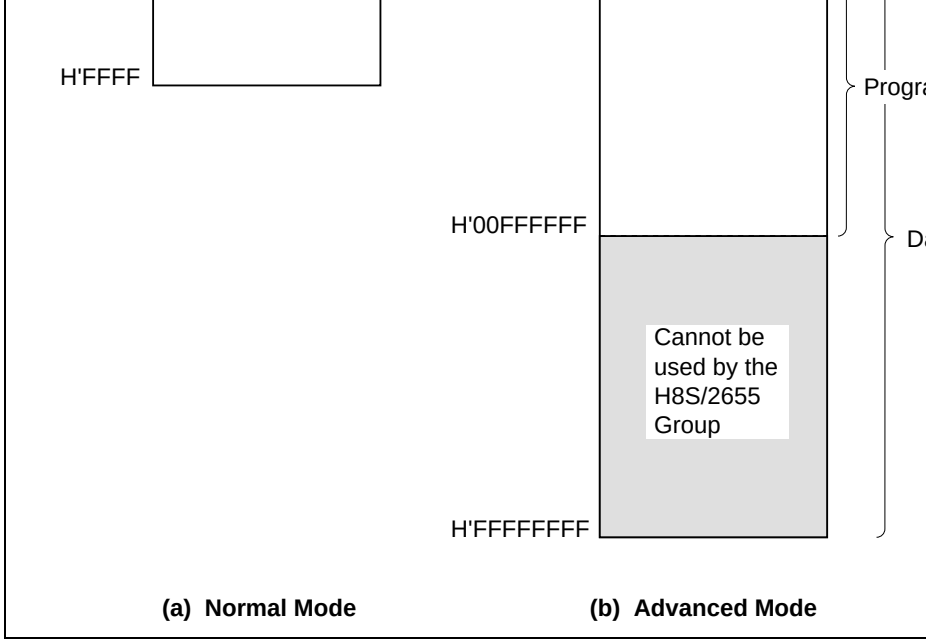
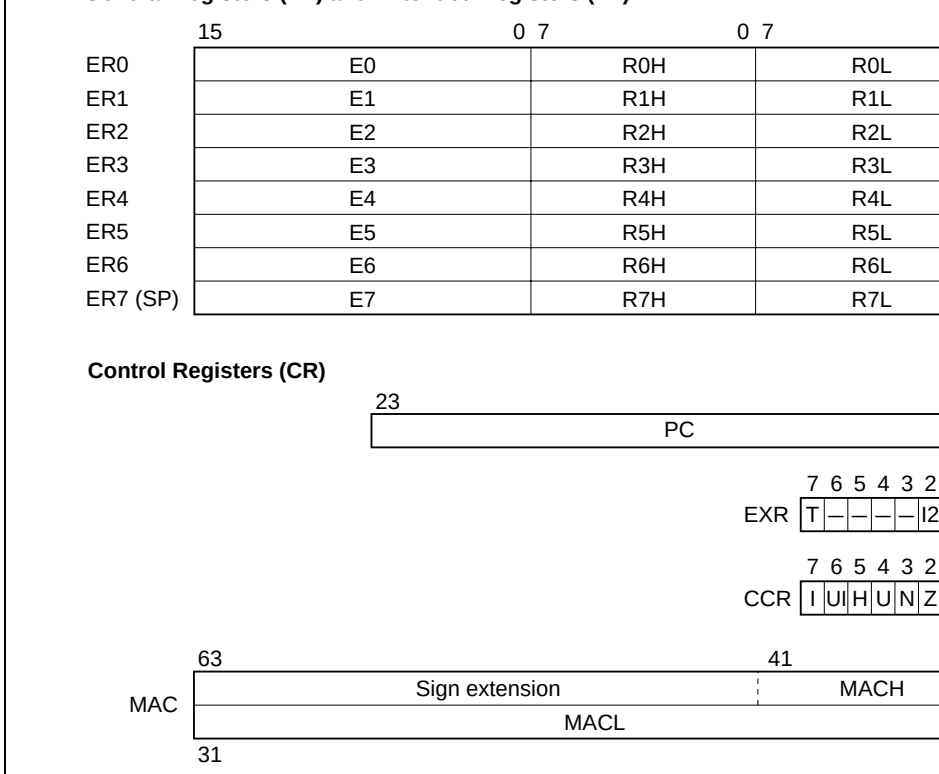


Figure 2.6 Memory Map



Legend:

SP:	Stack pointer	H:	Half-carry flag
PC:	Program counter	U:	User bit
EXR:	Extended control register	N:	Negative flag
T:	Trace bit	Z:	Zero flag
I2 to I0:	Interrupt mask bits	V:	Overflow flag
CCR:	Condition-code register	C:	Carry flag
I:	Interrupt mask bit	MAC:	Multiply-accumulate register
UI:	User bit or interrupt mask bit		

Figure 2.7 CPU Registers

registers. The E registers (E0 to E7) are also referred to as extended registers.

The R registers divide into 8-bit general registers designated by the letters RH (R0H to
RL (R0L to R7L). These registers are functionally equivalent, providing a maximum s
registers.

Figure 2.8 illustrates the usage of the general registers. The usage of each register can
independently.

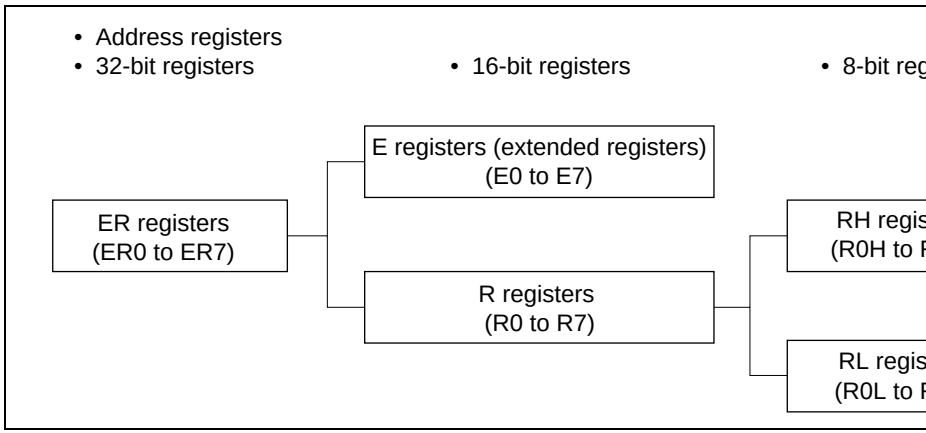


Figure 2.8 Usage of General Registers

General register ER7 has the function of stack pointer (SP) in addition to its general-r
function, and is used implicitly in exception handling and subroutine calls. Figure 2.9
stack.

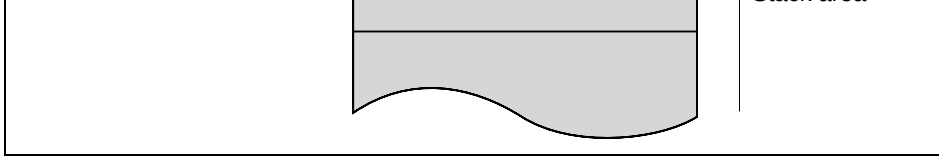


Figure 2.9 Stack

2.4.3 Control Registers

The control registers are the 24-bit program counter (PC), 8-bit extended control registe
8-bit condition-code register (CCR), and 64-bit multiply-accumulate register (MAC).

(1) Program Counter (PC)

This 24-bit counter indicates the address of the next instruction the CPU will execute. T
of all CPU instructions is 2 bytes (one word), so the least significant PC bit is ignored.
instruction is fetched, the least significant PC bit is regarded as 0.)

(2) Extended Control Register (EXR)

This 8-bit register contains the trace bit (T) and three interrupt mask bits (I2 to I0).

Bit 7—Trace Bit (T): Selects trace mode. When this bit is cleared to 0, instructions are
in sequence. When this bit is set to 1, a trace exception is generated each time an instru
executed.

Bits 6 to 3—Reserved: These bits are reserved. They are always read as 1.

Bits 2 to 0—Interrupt Mask Bits (I2 to I0): These bits designate the interrupt mask le
7). For details, refer to section 5, Interrupt Controller.

Bit 7—Interrupt Mask Bit (I): Masks interrupts other than NMI when set to 1. (NMI regardless of the I bit setting.) The I bit is set to 1 by hardware at the start of an except handling sequence. For details, refer to section 5, Interrupt Controller.

Bit 6—User Bit or Interrupt Mask Bit (UI): Can be written and read by software us LDC, STC, ANDC, ORC, and XORC instructions. This bit can also be used as an inte bit. For details, refer to section 5, Interrupt Controller.

Bit 5—Half-Carry Flag (H): When the ADD.B, ADDX.B, SUB.B, SUBX.B, CMP.B instruction is executed, this flag is set to 1 if there is a carry or borrow at bit 3, and cle otherwise. When the ADD.W, SUB.W, CMP.W, or NEG.W instruction is executed, th set to 1 if there is a carry or borrow at bit 11, and cleared to 0 otherwise. When the AD SUB.L, CMP.L, or NEG.L instruction is executed, the H flag is set to 1 if there is a ca borrow at bit 27, and cleared to 0 otherwise.

Bit 4—User Bit (U): Can be written and read by software using the LDC, STC, AND XORC instructions.

Bit 3—Negative Flag (N): Stores the value of the most significant bit (sign bit) of dat

Bit 2—Zero Flag (Z): Set to 1 to indicate zero data, and cleared to 0 to indicate non-z

Bit 1—Overflow Flag (V): Set to 1 when an arithmetic overflow occurs, and cleared times.

Bit 0—Carry Flag (C): Set to 1 when a carry occurs, and cleared to 0 otherwise. Use

- Add instructions, to indicate a carry
- Subtract instructions, to indicate a borrow
- Shift and rotate instructions, to store the value shifted out of the end bit

The carry flag is also used as a bit accumulator by bit manipulation instructions.

This 64-bit register stores the results of multiply-and-accumulate operations. It consists
bit registers denoted MACH and MACL. The lower 10 bits of MACH are valid; the up
a sign extension.

2.4.4 Initial Register Values

Reset exception handling loads the CPU's program counter (PC) from the vector table,
trace bit in EXR to 0, and sets the interrupt mask bits in CCR and EXR to 1. The other
and the general registers are not initialized. In particular, the stack pointer (ER7) is not
The stack pointer should therefore be initialized by an MOV.L instruction executed imm
after a reset.

Figure 2.10 shows the data formats in general registers.

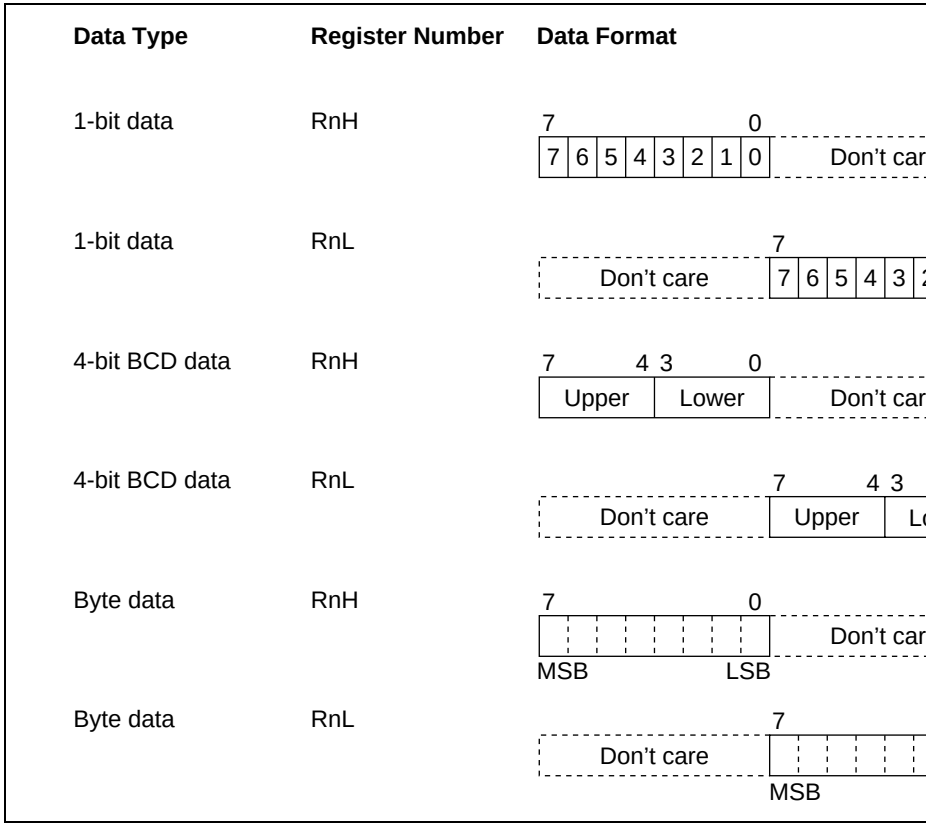
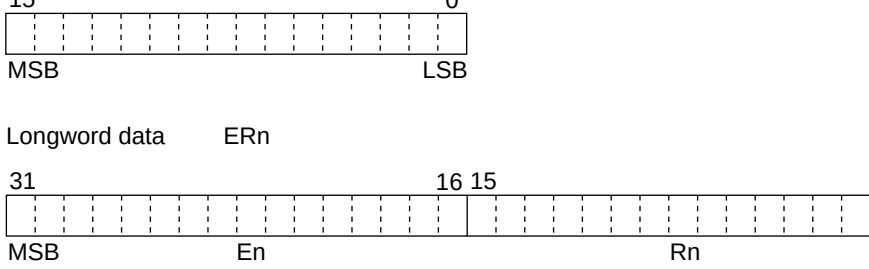


Figure 2.10 General Register Data Formats



Legend:
ERn: General register ER
En: General register E
Rn: General register R
RnH: General register RH
RnL: General register RL
MSB: Most significant bit
LSB: Least significant bit

Figure 2.10 General Register Data Formats (cont)

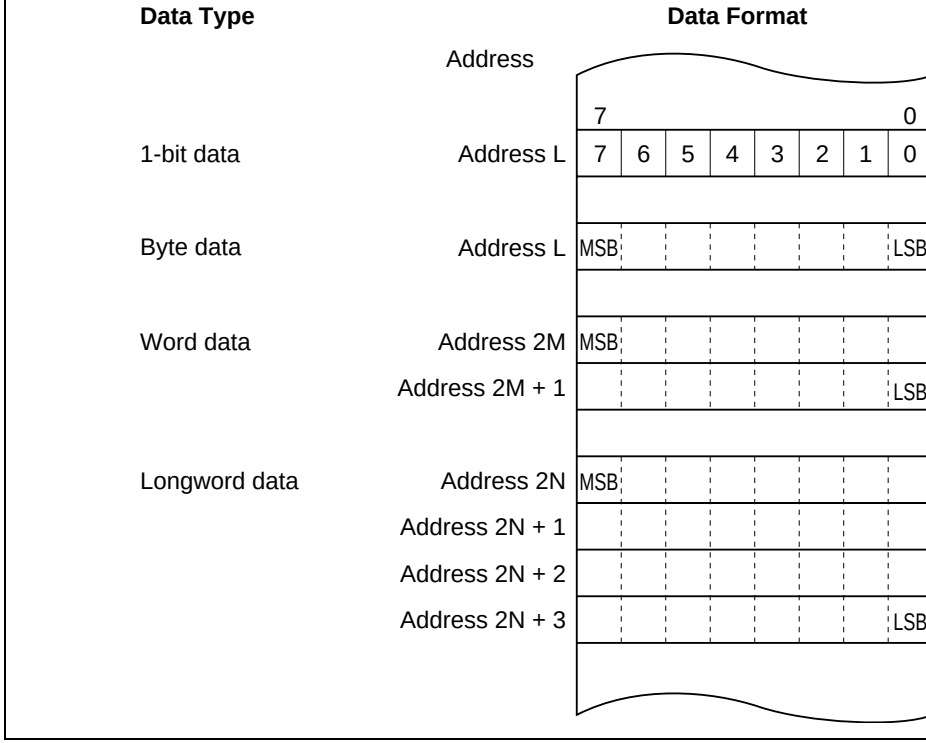


Figure 2.11 Memory Data Formats

When ER7 is used as an address register to access the stack, the operand size should b
or longword size.

Function	Instructions	Size
Data transfer	MOV	BW
	POP*1, PUSH*1	WL
	LDM, STM	L
	MOVFPE, MOVTPE	B
Arithmetic operations	ADD, SUB, CMP, NEG	BW
	ADDX, SUBX, DAA, DAS	B
	INC, DEC	BW
	ADDS, SUBS	L
	MULXU, DIVXU, MULXS, DIVXS	BW
	EXTU, EXTS	WL
	TAS	B
	MAC, LDMAC, STMAC, CLRMAC	—
Logic operations	AND, OR, XOR, NOT	BW
Shift	SHAL, SHAR, SHLL, SHLR, ROTL, ROTR, ROTXL, ROTXR	BW
Bit manipulation	BSET, BCLR, BNOT, BTST, BLD, BILD, BST, BIST, BAND, BIAND, BOR, BIOR, BXOR, BIXOR	B
Branch	Bcc*2, JMP, BSR, JSR, RTS	—
System control	TRAPA, RTE, SLEEP, LDC, STC, ANDC, ORC, XORC, NOP	—
Block data transfer	EEPMOV	—

Legend: B: Byte
W: Word
L: Longword

Notes: 1. POP.W Rn and PUSH.W Rn are identical to MOV.W @SP+, Rn and MOV.W @-SP. POP.L ERn and PUSH.L ERn are identical to MOV.L @SP+, ERn and MOV.L ERn, @-SP.
2. Bcc is the general name for conditional branch instructions.

Function	Instruction	#xx	Rn	@ERn	@(d:16,ERn	@(d:32,ERn	@–ERn/@E	@aa:8	@aa:16	@aa:24	@aa:32	@(d:8,PC)	@(d:16,PC
Data transfer	MOV	BWL	BWL	BWL	BWL	BWL	BWL	B	BWL	—	BWL	—	—
	POP, PUSH	—	—	—	—	—	—	—	—	—	—	—	—
	LDM, STM	—	—	—	—	—	—	—	—	—	—	—	—
	MOVEPE, MOVTPE	—	—	—	—	—	—	—	B	—	—	—	—
Arithmetic operations	ADD, CMP	BWL	BWL	—	—	—	—	—	—	—	—	—	—
	SUB	WL	BWL	—	—	—	—	—	—	—	—	—	—
	ADDX, SUBX	B	B	—	—	—	—	—	—	—	—	—	—
	ADDS, SUBS	—	L	—	—	—	—	—	—	—	—	—	—
	INC, DEC	—	BWL	—	—	—	—	—	—	—	—	—	—
	DAA, DAS	—	B	—	—	—	—	—	—	—	—	—	—
	MULXU, DIVXU	—	BW	—	—	—	—	—	—	—	—	—	—
	MULXS, DIVXS	—	BW	—	—	—	—	—	—	—	—	—	—
	NEG	—	BWL	—	—	—	—	—	—	—	—	—	—
	EXTU, EXTS	—	WL	—	—	—	—	—	—	—	—	—	—
	TAS	—	—	B	—	—	—	—	—	—	—	—	—
	MAC	—	—	—	—	—	○	—	—	—	—	—	—
	CLRMAC	—	—	—	—	—	—	—	—	—	—	—	—
	LDMAC, STMAC	—	L	—	—	—	—	—	—	—	—	—	—
Logic operations	AND, OR, XOR	BWL	BWL	—	—	—	—	—	—	—	—	—	—
	NOT	—	BWL	—	—	—	—	—	—	—	—	—	—
Shift		—	BWL	—	—	—	—	—	—	—	—	—	—
Bit manipulation		—	B	B	—	—	—	B	B	—	B	—	—

System control	TRAPA	—	—	—	—	—	—	—	—	—	—	—	—
	RTE	—	—	—	—	—	—	—	—	—	—	—	—
	SLEEP	—	—	—	—	—	—	—	—	—	—	—	—
	LDC	B	B	W	W	W	W	—	W	—	W	—	—
	STC	—	B	W	W	W	W	—	W	—	W	—	—
	ANDC, ORC, XORC	B	—	—	—	—	—	—	—	—	—	—	—
	NOP	—	—	—	—	—	—	—	—	—	—	—	—
Block data transfer		—	—	—	—	—	—	—	—	—	—	—	—

Legend:

B: Byte

W: Word

L: Longword

Rn	General register*
ERn	General register (32-bit register)
MAC	Multiply-accumulate register (32-bit register)
(EAd)	Destination operand
(EAs)	Source operand
EXR	Extended control register
CCR	Condition-code register
N	N (negative) flag in CCR
Z	Z (zero) flag in CCR
V	V (overflow) flag in CCR
C	C (carry) flag in CCR
PC	Program counter
SP	Stack pointer
#IMM	Immediate data
disp	Displacement
+	Addition
–	Subtraction
×	Multiplication
÷	Division
∧	Logical AND
∨	Logical OR
⊕	Logical exclusive OR
→	Move
¬	NOT (logical complement)
:8/:16/:24/:32	8-, 16-, 24-, or 32-bit length

Note: * General registers include 8-bit registers (R0H to R7H, R0L to R7L), 16-bit r
to R7, E0 to E7), and 32-bit registers (ER0 to ER7).

POP	W/L	@SP+ → Rn Pops a register from the stack. POP.W Rn is MOV.W @SP+, Rn. POP.L ERn is identical to @SP+, ERn.
PUSH	W/L	Rn → @–SP Pushes a register onto the stack. PUSH.W Rn identical to MOV.W Rn, @–SP. PUSH.L ERn to MOV.L ERn, @–SP.
LDM	L	@SP+ → Rn (register list) Pops two or more general registers from the s
STM	L	Rn (register list) → @–SP Pushes two or more general registers onto th

Note: * Size refers to the operand size.
B: Byte
W: Word
L: Longword

		byte data in two general registers, or on imm and data in a general register.
INC DEC	B/W/L	Rd ± 1 → Rd, Rd ± 2 → Rd Increments or decrements a general register (Byte operands can be incremented or decre 1 only.)
ADDS SUBS	L	Rd ± 1 → Rd, Rd ± 2 → Rd, Rd ± 4 → Rd Adds or subtracts the value 1, 2, or 4 to or fr 32-bit register.
DAA DAS	B	Rd decimal adjust → Rd Decimal-adjusts an addition or subtraction re general register by referring to the CCR to p BCD data.
MULXU	B/W	Rd × Rs → Rd Performs unsigned multiplication on data in registers: either 8 bits × 8 bits → 16 bits or 1 16 bits → 32 bits.
MULXS	B/W	Rd × Rs → Rd Performs signed multiplication on data in two registers: either 8 bits × 8 bits → 16 bits or 1 16 bits → 32 bits.
DIVXU	B/W	Rd ÷ Rs → Rd Performs unsigned division on data in two g registers: either 16 bits ÷ 8 bits → 8-bit quoti remainder or 32 bits ÷ 16 bits → 16-bit quoti bit remainder.

Note: * Size refers to the operand size.
B: Byte
W: Word
L: Longword

		bits according to the result.
NEG	B/W/L	0 – Rd → Rd Takes the two's complement (arithmetic comp data in a general register.
EXTU	W/L	Rd (zero extension) → Rd Extends the lower 8 bits of a 16-bit register to or the lower 16 bits of a 32-bit register to long by padding with zeros on the left.
EXTS	W/L	Rd (sign extension) → Rd Extends the lower 8 bits of a 16-bit register to or the lower 16 bits of a 32-bit register to long by extending the sign bit.
TAS	B	@ERd – 0, 1 → (<bit 7> of @Erd) Tests memory contents, and sets the most sig (bit 7) to 1.
MAC	—	(EAs) × (EAd) + MAC → MAC Performs signed multiplication on memory co adds the result to the multiply-accumulate reg following operations can be performed: 16 bits × 16 bits + 32 bits → 32 bits, saturatin 16 bits × 16 bits + 42 bits → 42 bits, non-satu
CLRMAC	—	0 → MAC Clears the multiply-accumulate register to zer
LDMAC STMAC	L	Rs → MAC, MAC → Rd Transfers data between a general register an multiply-accumulate register.

Note: * Size refers to the operand size.
B: Byte
W: Word
L: Longword

			register and another general register or imm
	NOT	B/W/L	¬ (Rd) → (Rd) Takes the one's complement of general regi contents.
Shift operations	SHAL SHAR	B/W/L	Rd (shift) → Rd Performs an arithmetic shift on general regis 1-bit or 2-bit shift is possible.
	SHLL SHLR	B/W/L	Rd (shift) → Rd Performs a logical shift on general register c 1-bit or 2-bit shift is possible.
	ROTL ROTR	B/W/L	Rd (rotate) → Rd Rotates general register contents. 1-bit or 2-bit rotation is possible.
	ROTXL ROTXR	B/W/L	Rd (rotate) → Rd Rotates general register contents through th 1-bit or 2-bit rotation is possible.

Note: * Size refers to the operand size.
B: Byte
W: Word
L: Longword

		immediate data or the lower three bits of a ge register.
BNOT	B	¬ (<bit-No.> of <EAd>) → (<bit-No.> of <EAd Inverts a specified bit in a general register or operand. The bit number is specified by 3-bit data or the lower three bits of a general regist
BTST	B	¬ (<bit-No.> of <EAd>) → Z Tests a specified bit in a general register or m operand and sets or clears the Z flag accordi bit number is specified by 3-bit immediate dat lower three bits of a general register.
BAND	B	C ∧ (<bit-No.> of <EAd>) → C ANDs the carry flag with a specified bit in a g register or memory operand and stores the re carry flag.
BIAND	B	C ∧ ¬ (<bit-No.> of <EAd>) → C ANDs the carry flag with the inverse of a spec a general register or memory operand and st result in the carry flag. The bit number is specified by 3-bit immediat
BOR	B	C ∨ (<bit-No.> of <EAd>) → C ORs the carry flag with a specified bit in a ge register or memory operand and stores the re carry flag.
BIOR	B	C ∨ ¬ (<bit-No.> of <EAd>) → C ORs the carry flag with the inverse of a specif general register or memory operand and store result in the carry flag. The bit number is specified by 3-bit immediat

Note: * Size refers to the operand size.
B: Byte

		specified bit in a general register or memory and stores the result in the carry flag. The bit number is specified by 3-bit immedia
BLD	B	(<bit-No.> of <EAd>) → C Transfers a specified bit in a general registe operand to the carry flag.
BILD	B	¬ (<bit-No.> of <EAd>) → C Transfers the inverse of a specified bit in a g register or memory operand to the carry flag The bit number is specified by 3-bit immedia
BST	B	C → (<bit-No.> of <EAd>) Transfers the carry flag value to a specified general register or memory operand.
BIST	B	¬ C → (<bit-No.> of <EAd>) Transfers the inverse of the carry flag value specified bit in a general register or memory The bit number is specified by 3-bit immedia

Note: * Size refers to the operand size.
B: Byte

		BLS	Low or same	C ∨ Z
		BCC(BHS)	Carry clear (high or same)	C = 0
		BCS(BLO)	Carry set (low)	C = 1
		BNE	Not equal	Z = 0
		BEQ	Equal	Z = 1
		BVC	Overflow clear	V = 0
		BVS	Overflow set	V = 1
		BPL	Plus	N = 0
		BMI	Minus	N = 1
		BGE	Greater or equal	N ⊕
		BLT	Less than	N ⊕
		BGT	Greater than	Z∨(N
		BLE	Less or equal	Z∨(N
	JMP	—	Branches unconditionally to a specified addre	
	BSR	—	Branches to a subroutine at a specified addre	
	JSR	—	Branches to a subroutine at a specified addre	
	RTS	—	Returns from a subroutine	

		them and memory. The upper 8 bits are vali
STC	B/W	CCR → (EAd), EXR → (EAd) Transfers CCR or EXR contents to a genera memory. Although CCR and EXR are 8-bit r word-size transfers are performed between t memory. The upper 8 bits are valid.
ANDC	B	CCR ∧ #IMM → CCR, EXR ∧ #IMM → EXR Logically ANDs the CCR or EXR contents w immediate data.
ORC	B	CCR ∨ #IMM → CCR, EXR ∨ #IMM → EXR Logically ORs the CCR or EXR contents wit data.
XORC	B	CCR ⊕ #IMM → CCR, EXR ⊕ #IMM → EXR Logically exclusive-ORs the CCR or EXR co immediate data.
NOP	—	PC + 2 → PC Only increments the program counter.

Note: * Size refers to the operand size.
B: Byte
W: Word

	Until R4 = 0 else next;
	Transfers a data block according to paramete general registers R4L or R4, ER5, and ER6. R4L or R4: size of block (bytes) ER5: starting source address ER6: starting destination address Execution of the next instruction begins as so transfer is completed.

2.6.4 Basic Instruction Formats

The H8S/2655 Group instructions consist of 2-byte (1-word) units. An instruction cons
operation field (op field), a register field (r field), an effective address extension (EA fi
condition field (cc).

(1) Operation Field

Indicates the function of the instruction, the addressing mode, and the operation to b
out on the operand. The operation field always includes the first four bits of the inst
Some instructions have two operation fields.

(2) Register Field

Specifies a general register. Address registers are specified by 3 bits, data registers b
4 bits. Some instructions have two register fields. Some have no register field.

(3) Effective Address Extension

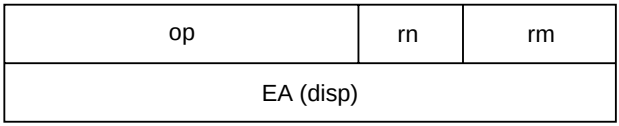
Eight, 16, or 32 bits specifying immediate data, an absolute address, or a displacem

(4) Condition Field

Specifies the branching condition of Bcc instructions.

Figure 2.12 shows examples of instruction formats.

(3) Operation field, register fields, and effective address extension



MOV.B @(d:16, Rn

(4) Operation field, effective address extension, and condition field

BRA d:16, etc

Figure 2.12 Instruction Formats (Examples)

absolute addressing mode to specify an operand, and register direct (BSET, BCLR, BN
BTST instructions) or immediate (3-bit) addressing mode to specify a bit number in the

Table 2.4 Addressing Modes

No.	Addressing Mode	Symbol
1	Register direct	Rn
2	Register indirect	@ERn
3	Register indirect with displacement	@(d:16,ERn)/@(d:32,ERn)
4	Register indirect with post-increment Register indirect with pre-decrement	@ERn+ @–ERn
5	Absolute address	@aa:8/@aa:16/@aa:24/@a
6	Immediate	#xx:8/#xx:16/#xx:32
7	Program-counter relative	@(d:8,PC)/@(d:16,PC)
8	Memory indirect	@@aa:8

(1) Register Direct—Rn

The register field of the instruction specifies an 8-, 16-, or 32-bit general register contai
operand. R0H to R7H and R0L to R7L can be specified as 8-bit registers. R0 to R7 and
can be specified as 16-bit registers. ER0 to ER7 can be specified as 32-bit registers.

(2) Register Indirect—@ERn

The register field of the instruction code specifies an address register (ERn) which cont
address of the operand on memory. If the address is a program instruction address, the
bits are valid and the upper 8 bits are all assumed to be 0 (H'00).

- Register indirect with post-increment—@ERn+

  The register field of the instruction code specifies an address register (ERn) which address of a memory operand. After the operand is accessed, 1, 2, or 4 is added to register contents and the sum is stored in the address register. The value added is 1 access, 2 for word transfer instruction, or 4 for longword transfer instruction. For v longword transfer instruction, the register value should be even.

- Register indirect with pre-decrement—@-ERn

  The value 1, 2, or 4 is subtracted from an address register (ERn) specified by the r in the instruction code, and the result becomes the address of a memory operand. T also stored in the address register. The value subtracted is 1 for byte access, 2 for v instruction, or 4 for longword transfer instruction. For word or longword transfer i the register value should be even.

(5) Absolute Address—@aa:8, @aa:16, @aa:24, or @aa:32

The instruction code contains the absolute address of a memory operand. The absolute may be 8 bits long (@aa:8), 16 bits long (@aa:16), 24 bits long (@aa:24), or 32 bits lo (@aa:32).

To access data, the absolute address should be 8 bits (@aa:8), 16 bits (@aa:16), or 32 (@aa:32) long. For an 8-bit absolute address, the upper 24 bits are all assumed to be 1 For a 16-bit absolute address the upper 16 bits are a sign extension. A 32-bit absolute access the entire address space.

A 24-bit absolute address (@aa:24) indicates the address of a program instruction. The bits are all assumed to be 0 (H'00).

Table 2.5 indicates the accessible absolute address ranges.

(6) Immediate—#xx:8, #xx:16, or #xx:32

The instruction contains 8-bit (#xx:8), 16-bit (#xx:16), or 32-bit (#xx:32) immediate da
operand.

The ADDS, SUBS, INC, and DEC instructions contain immediate data implicitly. Som
manipulation instructions contain 3-bit immediate data in the instruction code, specifyi
number. The TRAPA instruction contains 2-bit immediate data in its instruction code, s
vector address.

(7) Program-Counter Relative—@(d:8, PC) or @(d:16, PC)

This mode is used in the Bcc and BSR instructions. An 8-bit or 16-bit displacement cor
the instruction is sign-extended and added to the 24-bit PC contents to generate a branc
Only the lower 24 bits of this branch address are valid; the upper 8 bits are all assumed
(H'00). The PC value to which the displacement is added is the address of the first byte
instruction, so the possible branching range is –126 to +128 bytes (–63 to +64 words) o
+32768 bytes (–16383 to +16384 words) from the branch instruction. The resulting val
be an even number.

(8) Memory Indirect—@@aa:8

This mode can be used by the JMP and JSR instructions. The instruction code contains
absolute address specifying a memory operand. This memory operand contains a branc
The upper bits of the absolute address are all assumed to be 0, so the address range is 0
(H'0000 to H'00FF in normal mode, H'000000 to H'0000FF in advanced mode). In norm
the memory operand is a word operand and the branch address is 16 bits long. In advan
the memory operand is a longword operand, the first byte of which is assumed to be all

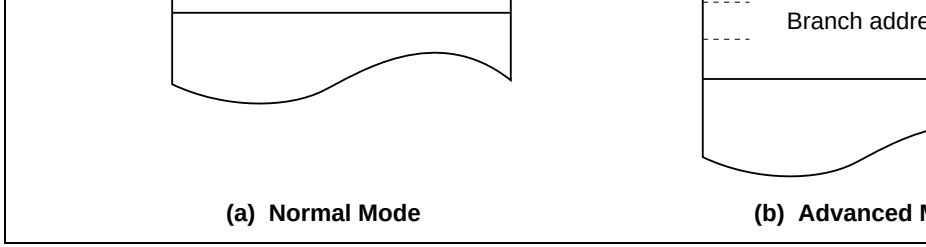


Figure 2.13 Branch Address Specification in Memory Indirect Mode

If an odd address is specified in word or longword memory access, or as a branch addr least significant bit is regarded as 0, causing data to be accessed or instruction code to at the address preceding the specified address. (For further information, see section 2.5 Data Formats.)

2.7.2 Effective Address Calculation

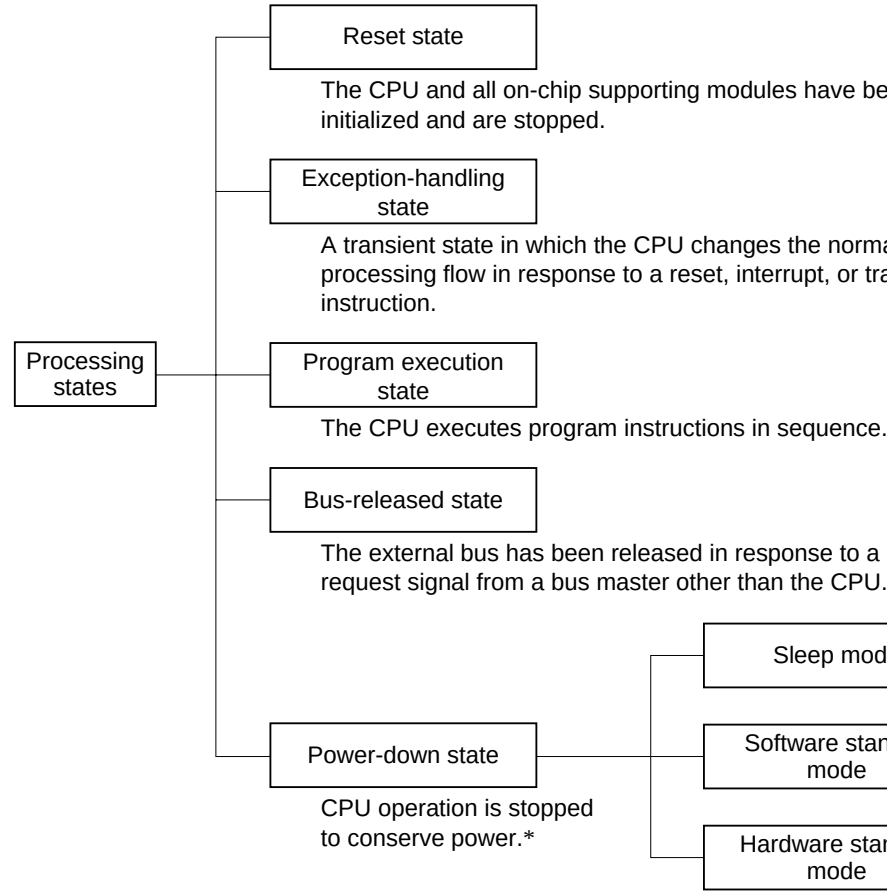
Table 2.6 indicates how effective addresses are calculated in each addressing mode. In mode the upper 8 bits of the effective address are ignored in order to generate a 16-bit

No.	Addressing Mode and Instruction Format	Effective Address Calculation	Effectiv
1	Register direct (Rn) op | rm | rm		Operand is gener
2	Register indirect (@ERn) op | r	31 General register contents 0	31 24 2 Don't care
3	Register indirect with displacement @(d:16, ERn) or @(d:32, ERn) op | r | disp	31 General register contents 0 31 Sign extension | disp 0	31 24 2 Don't care
4	Register indirect with post-increment or pre-decrement • Register indirect with post-increment @ERn+ op | r • Register indirect with pre-decrement @–ERn op | r	31 General register contents 0 1, 2, or 4 31 General register contents 0 1, 2, or 4 Operand Size | Value added B | 1	31 24 2 Don't care 31 24 2 Don't care

No.	Addressing Mode and Instruction Format	Effective Address Calculation	Effi
5	Absolute address @aa:8 op abs @aa:16 op abs @aa:24 op abs @aa:32 op abs	31 24 2 Don't care 31 24 2 Don't care ex 31 24 2 Don't care 31 24 2 Don't care	
6	Immediate #xx:8/#xx:16/#xx:32 op IMM	Operand is	

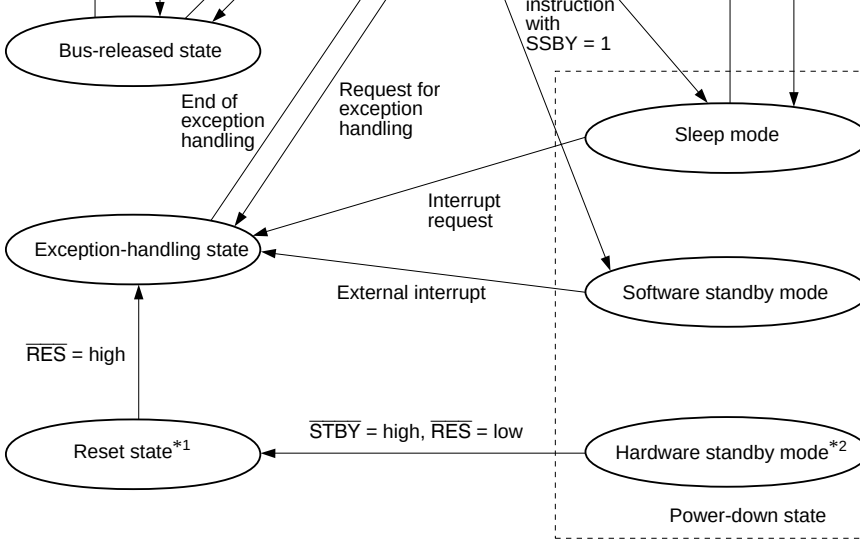
RENESAS

No.	Addressing Mode and Instruction Format	Effective Address Calculation	Eff
7	Program-counter relative @(d:8, PC)/@(d:16, PC) op | disp	23 0 PC contents 23 0 Sign extension | disp	31 24 2 Don't care
8	Memory indirect @@aa:8 • Normal mode op | abs • Advanced mode op | abs	31 8 7 0 H'000000 | abs 15 0 Memory contents 31 8 7 0 H'000000 | abs 31 0 Memory contents	31 24 2 Don't care 31 24 2 Don't care



Note: * The power-down state also includes a medium-speed mode, module stop

Figure 2.14 Processing States



Notes: 1. From any state except hardware standby mode, a transition to the reset state occurs whene goes low. A transition can also be made to the reset state when the watchdog timer overflo
2. From any state, a transition to hardware standby mode occurs when $\overline{STBY}$ goes low.

Figure 2.15 State Transitions

2.8.2 Reset State

When the $\overline{RES}$ input goes low all current processing stops and the CPU enters the reset interrupts are masked in the reset state. Reset exception handling starts when the $\overline{RES}$ s changes from low to high.

The reset state can also be entered by a watchdog timer overflow. For details, refer to s Watchdog Timer.

Exception handling is performed for traces, resets, interrupts, and trap instructions. Ta
indicates the types of exception handling and their priority. Trap instruction exception
always accepted, in the program execution state.

Exception handling and the stack structure depend on the interrupt control mode set in

Table 2.7 Exception Handling Types and Priority

Priority	Type of Exception	Detection Timing	Start of Exception H
High ▲	Reset	Synchronized with clock	Exception handling st immediately after a lo transition at the $\overline{\text{RES}}$ the watchdog timer ov
	Trace	End of instruction execution or end of exception-handling sequence*1	When the trace (T) bi the trace starts at the current instruction or exception-handling se
	Interrupt	End of instruction execution or end of exception-handling sequence*2	When an interrupt is r exception handling st end of the current inst current exception-han sequence
	Trap instruction	When TRAPA instruction is executed	Exception handling st trap (TRAPA) instruct executed*3
Low			

Notes: 1. Traces are enabled only in interrupt control modes 2 and 3. Trace exceptio not executed at the end of the RTE instruction.
2. Interrupts are not detected at the end of the ANDC, ORC, XORC, and LDC or immediately after reset exception handling.
3. Trap instruction exception handling is always accepted, in the program exe

Traces are enabled only in interrupt control modes 2 and 3. Trace mode is entered when of EXR is set to 1. When trace mode is established, trace exception handling starts at th each instruction.

At the end of a trace exception-handling sequence, the T bit of EXR is cleared to 0 and is cleared. Interrupt masks are not affected.

The T bit saved on the stack retains its value of 1, and when the RTE instruction is exec return from the trace exception-handling routine, trace mode is entered again. Trace exc handling is not executed at the end of the RTE instruction.

Trace mode is not entered in interrupt control modes 0 and 1, regardless of the state of t

(4) Interrupt Exception Handling and Trap Instruction Exception Handling

When interrupt or trap-instruction exception handling begins, the CPU references the st (ER7) and pushes the program counter and other control registers onto the stack. Next, alters the settings of the interrupt mask bits in the control registers. Then the CPU fetch address (vector) from the exception vector table and program execution starts from that address.

Figure 2.16 shows the stack after exception handling ends.

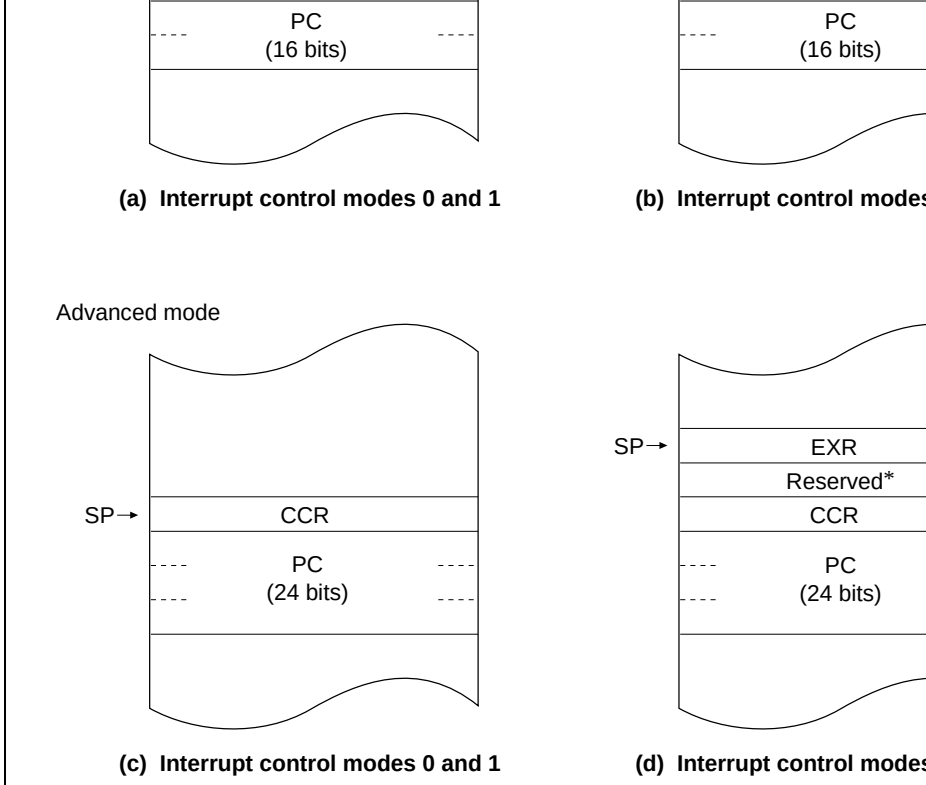


Figure 2.16 Stack Structure after Exception Handling (Examples)

2.8.4 Program Execution State

In this state the CPU executes program instructions in sequence.

2.8.6 Power-Down State

The power-down state includes both modes in which the CPU stops operating and mod
the CPU does not stop. There are three modes in which the CPU stops operating: sleep
software standby mode, and hardware standby mode. There are also two other power-d
modes: medium-speed mode, and module stop mode. In medium-speed mode the CPU
bus masters operate on a medium-speed clock. Module stop mode permits halting of th
of individual modules, other than the CPU. For details, refer to section 21, Power-Dow

(1) Sleep Mode

A transition to sleep mode is made if the SLEEP instruction is executed while the softw
standby bit (SSBY) in the standby control register (SBYCR) is cleared to 0. In sleep m
operations stop immediately after execution of the SLEEP instruction. The contents of
registers are retained.

(2) Software Standby Mode

A transition to software standby mode is made if the SLEEP instruction is executed wh
SSBY bit in SBYCR is set to 1. In software standby mode, the CPU and clock halt and
operations stop. As long as a specified voltage is supplied, the contents of CPU register
chip RAM are retained. The I/O ports also remain in their existing states.

(3) Hardware Standby Mode

A transition to hardware standby mode is made when the $\overline{\text{STBY}}$ pin goes low. In hardw
standby mode, the CPU and clock halt and all MCU operations stop. The on-chip suppo
modules are reset, but as long as a specified voltage is supplied, on-chip RAM contents
retained.

2.9.2 On-Chip Memory (ROM, RAM)

On-chip memory is accessed in one state. The data bus is 16 bits wide, permitting both
word transfer instruction. Figure 2.17 shows the on-chip memory access cycle. Figure
the pin states.

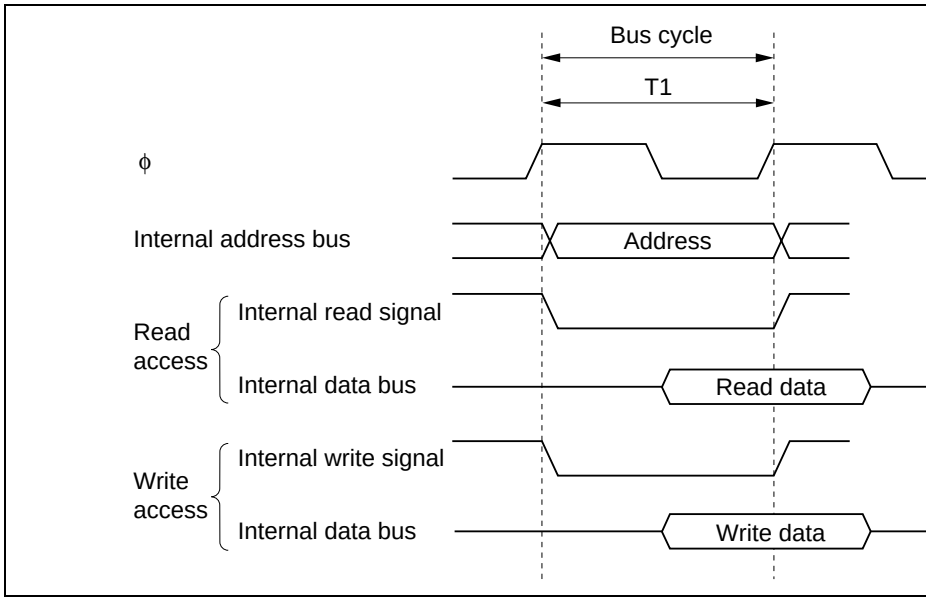


Figure 2.17 On-Chip Memory Access Cycle

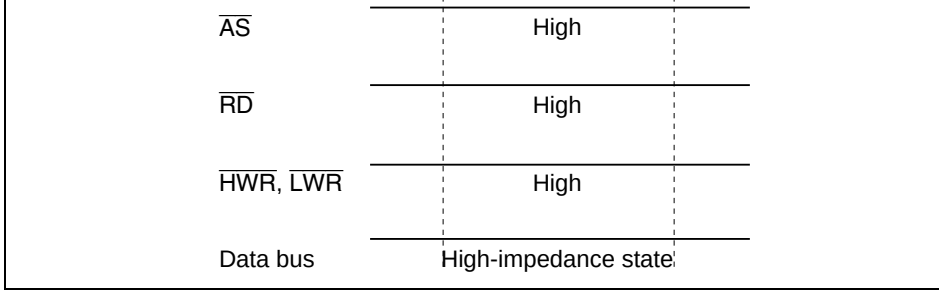


Figure 2.18 Pin States during On-Chip Memory Access

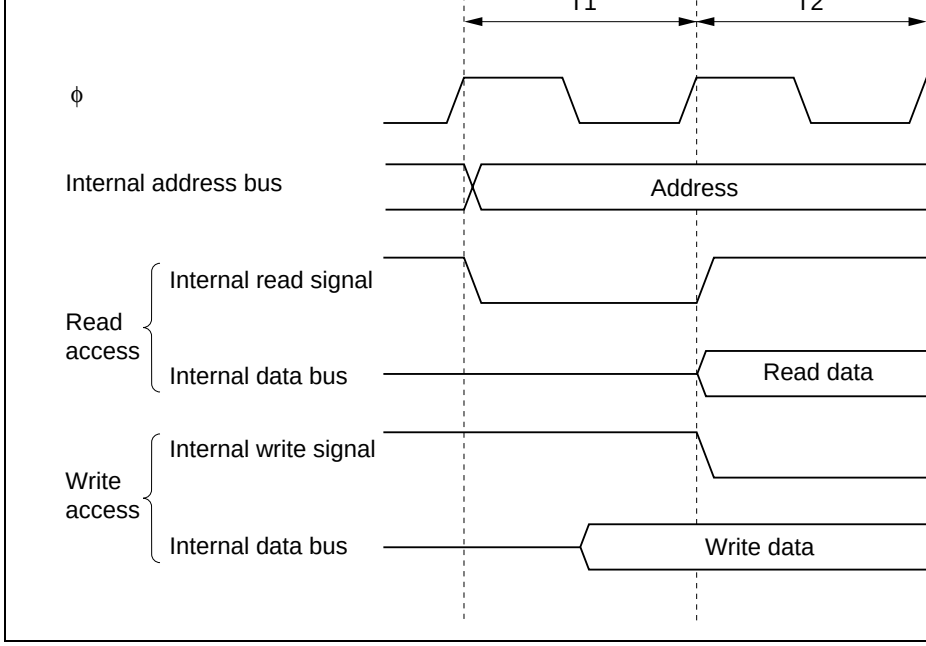


Figure 2.19 On-Chip Supporting Module Access Cycle

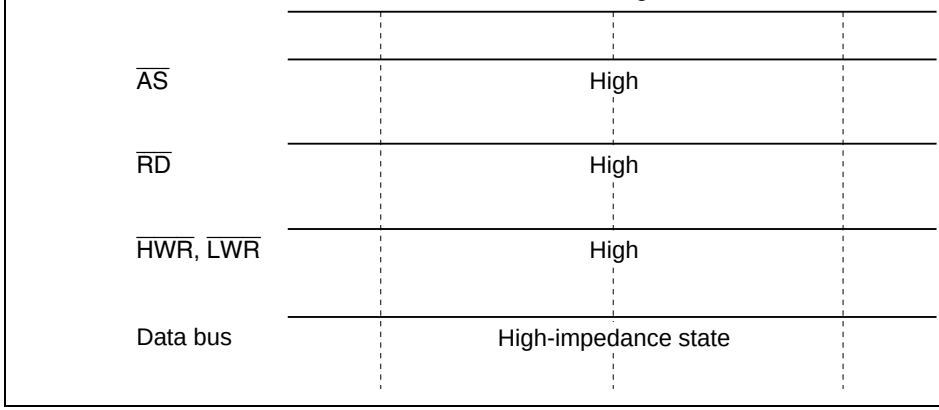


Figure 2.20 Pin States during On-Chip Supporting Module Access

2.9.4 External Address Space Access Timing

The external address space is accessed with an 8-bit or 16-bit data bus width in a two-s
three-state bus cycle. In three-state access, wait states can be inserted. For further detail
section 6, Bus Controller.

the CPU operating mode, enabling/disabling of on-chip ROM, and the initial bus widt
setting the mode pins (MD_2 to MD_0).

Table 3.1 lists the MCU operating modes.

Table 3.1 MCU Operating Mode Selection

MCU Operating Mode	MD_2	MD_1	MD_0	CPU Operating Mode	Description	On-Chip ROM	Extern Initial Width
0	0	0	0	—	—	—	—
1			1	Normal	On-chip ROM disabled, expanded mode	Disabled	8 bits
2		1	0		On-chip ROM enabled, expanded mode	Enabled	8 bits
3			1		Single-chip mode		—
4	1	0	0	Advanced	On-chip ROM disabled, expanded mode	Disabled	16 bits
5			1				8 bits
6		1	0		On-chip ROM enabled, expanded mode	Enabled	8 bits
7			1		Single-chip mode		—

The CPU's architecture allows for 4 Gbytes of address space, but the H8S/2655 Grou
accesses a maximum of 16 Mbytes.

Modes 1, 2, and 4 to 6 are externally expanded modes that allow access to external me
peripheral devices.

3.1.2 Register Configuration

The H8S/2655 Group has a mode control register (MDCR) that indicates the inputs at t pins (MD_2 to MD_0), and a system control register (SYSCR) that controls the operation o H8S/2655 Group. Table 3.2 summarizes these registers.

Table 3.2 MCU Registers

Name	Abbreviation	R/W	Initial Value	Add
Mode control register	MDCR	R	Undetermined	H'FF
System control register	SYSCR	R/W	H'01	H'FF

Note: * Lower 16 bits of the address.

R/W	:	—	—	—	—	—	R	R

Note: * Determined by pins MD_2 to MD_0.

MDCR is an 8-bit read-only register that indicates the current operating mode of the H Group.

Bit 7—Reserved: Read-only bit, always read as 1.

Bits 6 to 3—Reserved: Read-only bits, always read as 0.

Bits 2 to 0—Mode Select 2 to 0 (MDS2 to MDS0): These bits indicate the input leve MD_2 to MD_0 (the current operating mode). Bits MDS2 to MDS0 correspond to MD_2 to MDS2 to MDS0 are read-only bits-they cannot be written to. The mode pin (MD_2 to M levels are latched into these bits when MDCR is read. These latches are canceled by a reset, but are retained after a manual reset.

3.2.2 System Control Register (SYSCR)

Bit	:	7	6	5	4	3	2	1
		MACS	—	INTM1	INTM0	NMIEG	—	—
Initial value	:	0	0	0	0	0	0	0
R/W	:	R/W	—	R/W	R/W	R/W	—	—

Bit 7—MAC Saturation (MACS): Selects either saturating or non-saturating calcula MAC instruction.

Bits 5 and 4—Interrupt Control Mode 1 and 0 (INTM1, INTM0): These bits select mode of the interrupt controller. For details of the interrupt control modes, see section 5 Interrupt Control Modes and Interrupt Operation.

Bit 5	Bit 4	Interrupt	
INTM1	INTM0	Control Mode	Description
0	0	0	Control of interrupts by I bit (I
	1	1	Control of interrupts by I bit, U bit, and ICR
1	0	2	Control of interrupts by I2 to I0 bits and IPR
	1	3	Control of interrupts by I, UI, and I2 to I0 bi and ICR and IPR

Bit 3—NMI Edge Select (NMIEG): Selects the valid edge of the NMI interrupt input.

Bit 3	
NMIEG	Description
0	An interrupt is requested at the falling edge of NMI input (I
1	An interrupt is requested at the rising edge of NMI input

Bits 2 and 1—Reserved: Read-only bits, always read as 0.

Bit 0—RAM Enable (RAME): Enables or disables the on-chip RAM. The RAME bit initialized when the reset status is released. It is not initialized in software standby mod

Bit 0	
RAME	Description
0	On-chip RAM is disabled
1	On-chip RAM is enabled (I

bus control signals. However, note that if 16-bit access is designated by the bus contro
mode switches to 16 bits and port E becomes a data bus.

3.3.2 Mode 2

The CPU can access a 64-kbyte address space in normal mode. The on-chip ROM is e
8-bit bus mode is set. immediately after a reset.

Ports B and C function as input ports immediately after a reset. They can each be set t
addresses by setting the corresponding bits in the data direction register (DDR) to 1. P
functions as a data bus, and part of port F carries bus control signals. However, note th
access is designated by the bus controller, the bus mode switches to 16 bits and port E
data bus.

The amount of on-chip ROM that can be used is limited to 56 kbytes.

3.3.3 Mode 3

The CPU can access a 64-kbyte address space in normal mode. The on-chip ROM is e
external addresses cannot be accessed.

All I/O ports are available for use as input-output ports.

The amount of on-chip ROM that can be used is limited to 56 kbytes.

3.3.4 Mode 4

The CPU can access a 16-Mbyte address space in advanced mode. The on-chip ROM

Ports A, B and C function as an address bus, ports D and E function as a data bus, and
F carries bus control signals.

F carries bus control signals.

The initial bus mode after a reset is 8 bits, with 8-bit access to all areas. However, note least one area is designated for 16-bit access by the bus controller, the bus mode switch bits and port E becomes a data bus.

3.3.6 Mode 6

The CPU can access a 16-Mbyte address space in advanced mode. The on-chip ROM is

Ports A, B and C function as input ports immediately after a reset. They can each be set addresses by setting the corresponding bits in the data direction register (DDR) to 1. Po functions as a data bus, and part of port F carries bus control signals.

The initial bus mode after a reset is 8 bits, with 8-bit access to all areas.

3.3.7 Mode 7

The CPU can access a 16-Mbyte address space in advanced mode. The on-chip ROM is but external addresses cannot be accessed.

All I/O ports are available for use as input-output ports.

	PA_4 to PA_0				A	A	
Port B		A	P*/A	P	A	A	P*/A
Port C		A	P*/A	P	A	A	P*/A
Port D		D	D	P	D	D	D
Port E		P*/D	P*/D	P	P*/D	P*/D	P*/D
Port F	PF_7	P*/C*	P*/C*	P*/C	P*/C*	P*/C*	P*/C*
	PF_6 to PF_3	C	C	P	C	C	C
	PF_2 to PF_0	P*/C	P*/C		P*/C	P*/C	P*/C

Legend:

P: I/O port

A: Address bus output

D: Data bus I/O

C: Control signals, clock I/O

Note: * After reset

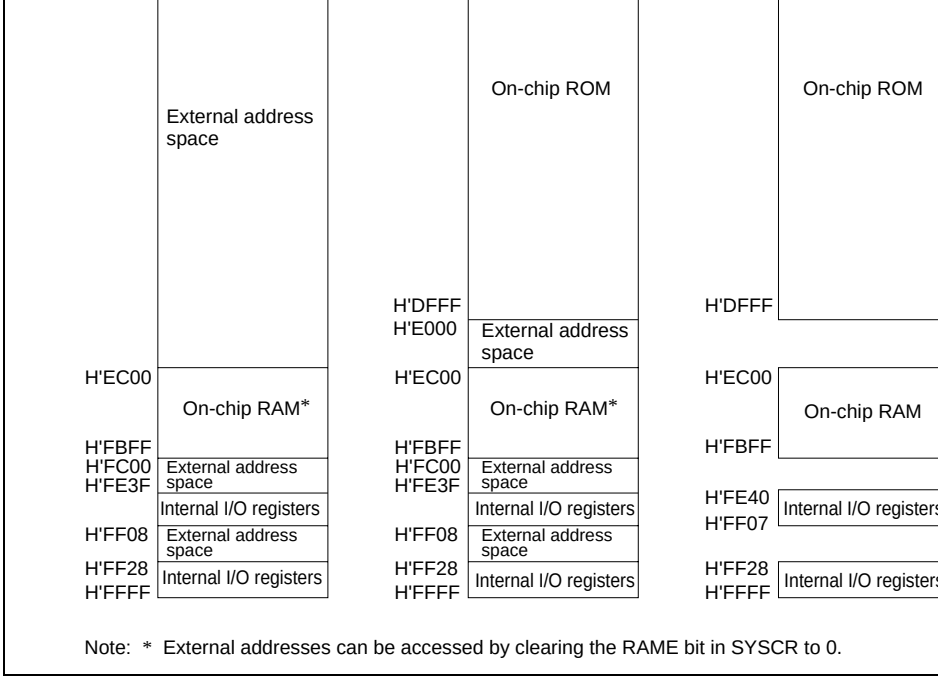
3.5 Memory Map in Each Operating Mode

Figure 3.1 shows a memory map for each of the operating modes.

The address space is 64 kbytes in modes 1 to 3 (normal modes), and 16 Mbytes in mo (advanced modes).

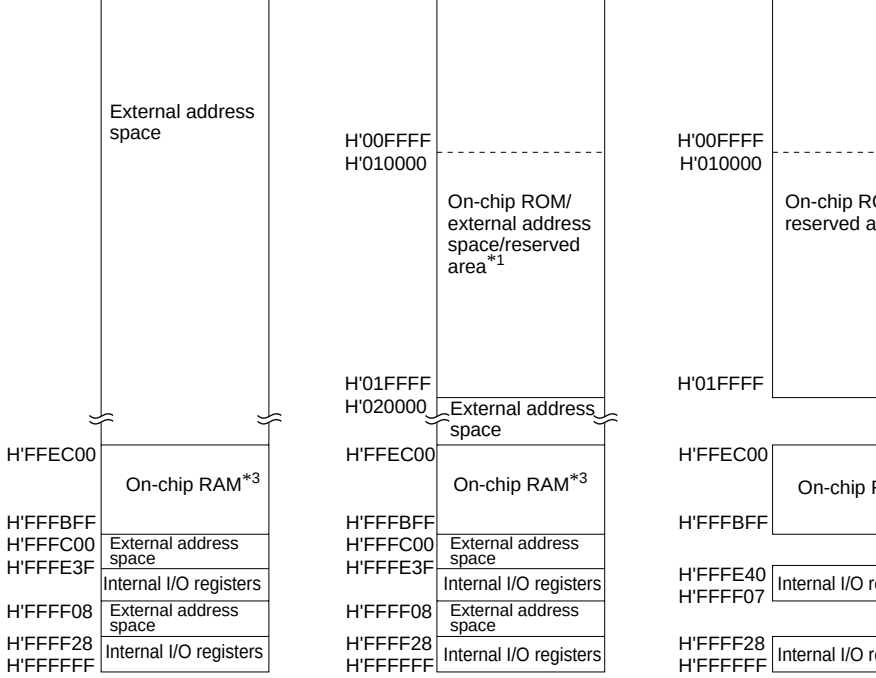
The on-chip ROM of H8S/2655 contains 128 kbytes, but only 56 kbytes are available and 3 (normal modes).

The address space is divided into eight areas for modes 4 to 7. For details, see section Controller.



Note: * External addresses can be accessed by clearing the RAME bit in SYSCR to 0.

Figure 3.1 Memory Map in Each Operating Mode



Notes: 1. When the EAE bit in BCRL is set to 1, this area is external address space. When the EAE b cleared to 0, it is on-chip ROM in the H8S/2655, and a reserved area in the H8S/2653.
2. In the H8S/2655, this area is reserved when the EAE bit in BCRL is set to 1, and on-chip R when the EAE bit is cleared to 0. In the H8S/2653, it is a reserved area.
3. External addresses can be accessed by clearing the RAME bit in SYSCR to 0.

Figure 3.1 Memory Map in Each Operating Mode (cont)

Exception handling is prioritized as shown in table 4.1. If two or more exceptions occu
simultaneously, they are accepted and processed in order of priority. Trap instruction e
are accepted at all times, in the program execution state. See appendix D.1, Port States
Mode.

Exception handling sources, the stack structure, and the operation of the CPU vary dep
the interrupt control mode set by the INTM0 and INTM1 bits of SYSCR.

Table 4.1 Exception Types and Priority

Priority	Exception Type	Start of Exception Handling
High	Reset	Starts immediately after a low-to-high transitio $\overline{\text{RES}}$ pin, or when the watchdog timer overflow
▲	Trace*1	Starts when execution of the current instructio exception handling ends, if the trace (T) bit is
	Interrupt	Starts when execution of the current instructio exception handling ends, if an interrupt reques issued*2
Low	Trap instruction (TRAPA)*3	Started by execution of a trap instruction (TRA

Notes: 1. Traces are enabled only in interrupt control modes 2 and 3. Trace exceptio
not executed after execution of an RTE instruction.
2. Interrupt detection is not performed on completion of ANDC, ORC, XORC,
instruction execution, or on completion of reset exception handling.
3. Trap instruction exception handling requests are accepted at all times in pr
execution state.

starts from that address.

For a reset exception, steps 2 and 3 above are carried out.

4.1.3 Exception Vector Table

The exception sources are classified as shown in figure 4.1. Different vector addresses
assigned to different exception sources.

Table 4.2 lists the exception sources and their vector addresses.

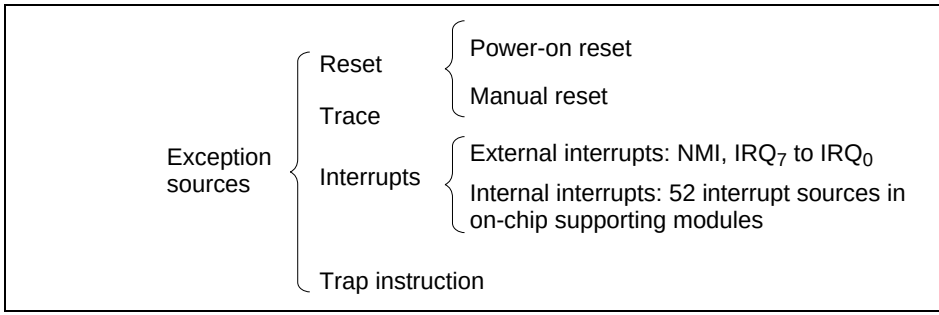


Figure 4.1 Exception Sources

In modes 6 and 7 in the H8S/2655, the on-chip ROM available for use after a power-on
64-kbyte area comprising addresses H'000000 to H'00FFFF. Care is required when sett
addresses. In this case, clearing the EAE bit in BCRL enables the 128-kbyte area comp
addresses H'000000 to H'01FFFF to be used.

		4	H'0008 to H'0009	H'0010 t
Trace		5	H'000A to H'000B	H'0014 t
Reserved for system use		6	H'000C to H'000D	H'0018 t
External interrupt	NMI	7	H'000E to H'000F	H'001C t
Trap instruction (4 sources)		8	H'0010 to H'0011	H'0020 t
		9	H'0012 to H'0013	H'0024 t
		10	H'0014 to H'0015	H'0028 t
		11	H'0016 to H'0017	H'002C t
Reserved for system use		12	H'0018 to H'0019	H'0030 t
		13	H'001A to H'001B	H'0034 t
		14	H'001C to H'001D	H'0038 t
		15	H'001E to H'001F	H'003C t
External interrupt	IRQ_0	16	H'0020 to H'0021	H'0040 t
	IRQ_1	17	H'0022 to H'0023	H'0044 t
	IRQ_2	18	H'0024 to H'0025	H'0048 t
	IRQ_3	19	H'0026 to H'0027	H'004C t
	IRQ_4	20	H'0028 to H'0029	H'0050 t
	IRQ_5	21	H'002A to H'002B	H'0054 t
	IRQ_6	22	H'002C to H'002D	H'0058 t
	IRQ_7	23	H'002E to H'002F	H'005C t
Internal interrupt*2		24 | 91	H'0030 to H'0031 | H'00B6 to H'00B7	H'0060 t H'016C t

Notes: 1. Lower 16 bits of the address.

2. For details of internal interrupt vectors, see section 5.3.3, Interrupt Exceptio Vector Table.

Immediately after a reset, interrupt control mode 0 is set.

Reset exception handling begins when the $\overline{\text{RES}}$ pin changes from low to high.

The level of the NMI pin at reset determines whether the type of reset is a power-on res
manual reset.

The H8S/2655 Group can also be reset by overflow of the watchdog timer. For details s
13, Watchdog Timer.

4.2.2 Reset Types

A reset can be of either of two types: a power-on reset or a manual reset. Reset types ar
table 4.3. A power-on reset should be used when powering on.

The internal state of the CPU is initialized by either type of reset. A power-on reset also
all the registers in the on-chip supporting modules, while a manual reset initializes all t
in the on-chip supporting modules except for the bus controller and I/O ports, which re
previous states.
With a manual reset, since the on-chip supporting modules are initialized, ports used as
supporting module I/O pins are switched to I/O ports controlled by DDR and DR.

Table 4.3 Reset Types

	Reset Transition Conditions		Internal State	
Type	**NMI**	**$\overline{\text{RES}}$**	**CPU**	**On-Chip Supporting Mo**
Power-on reset	High	Low	Initialized	Initialized
Manual reset	Low	Low	Initialized	Initialized, except for bus and I/O ports

A reset caused by the watchdog timer can also be of either of two types: a power-on res
manual reset.

1. The internal state of the CPU and the registers of the on-chip supporting modules a initialized, and the I bit is set to 1 in CCR.
2. The reset exception handling vector address is read and transferred to the PC, and execution starts from the address indicated by the PC.

Figures 4.2 and 4.3 show examples of the reset sequence.

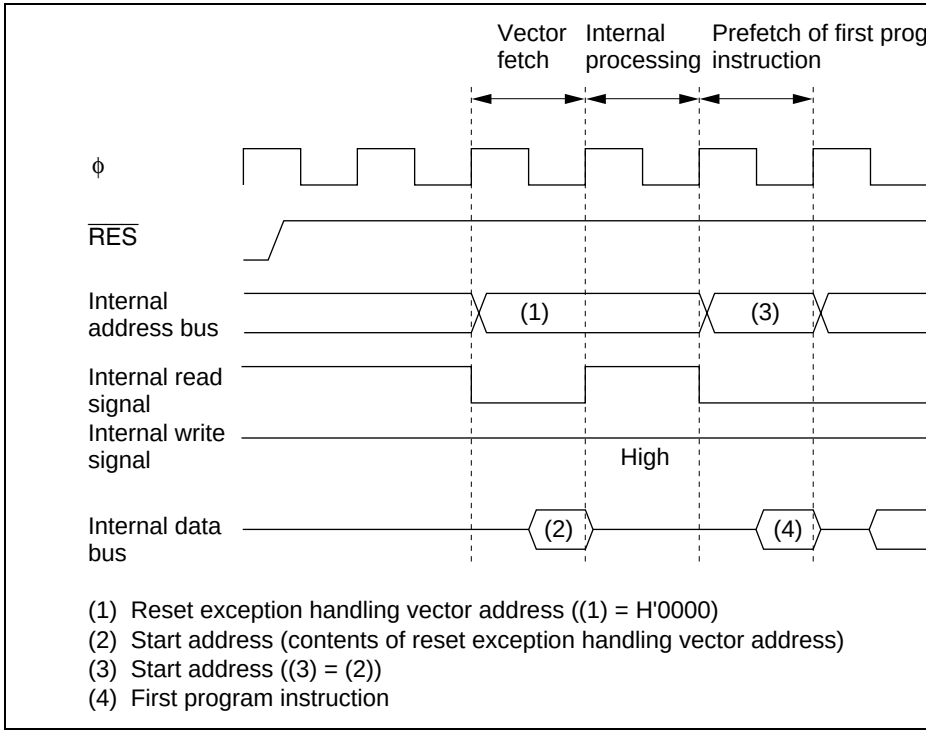
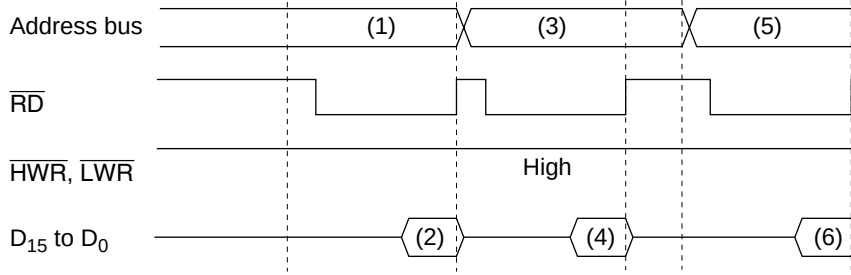


Figure 4.2 Reset Sequence (Modes 2 and 3)



(1) (3) Reset exception handling vector address ((1) = H'000000, (3) = H'00000
(2) (4) Start address (contents of reset exception handling vector address)
(5) Start address ((5) = (2) (4))
(6) First program instruction

Note: * 3 program wait states are inserted.

Figure 4.3 Reset Sequence (Mode 4)

4.2.4 Interrupts after Reset

If an interrupt is accepted after a reset but before the stack pointer (SP) is initialized, th
CCR will not be saved correctly, leading to a program crash. To prevent this, all interru
including NMI, are disabled immediately after a reset. Since the first instruction of a pr
always executed immediately after the reset state ends, make sure that this instruction i
the stack pointer (example: MOV.L #xx:32, SP).

Trace mode is canceled by clearing the T bit in EXR to 0. It is not affected by interrup

Table 4.4 shows the state of CCR and EXR after execution of trace exception handlin

Interrupts are accepted even within the trace exception handling routine.

The T bit saved on the stack retains its value of 1, and when control is returned from t
exception handling routine by the RTE instruction, trace mode resumes.

Trace exception handling is not carried out after execution of the RTE instruction.

Table 4.4 Status of CCR and EXR after Trace Exception Handling

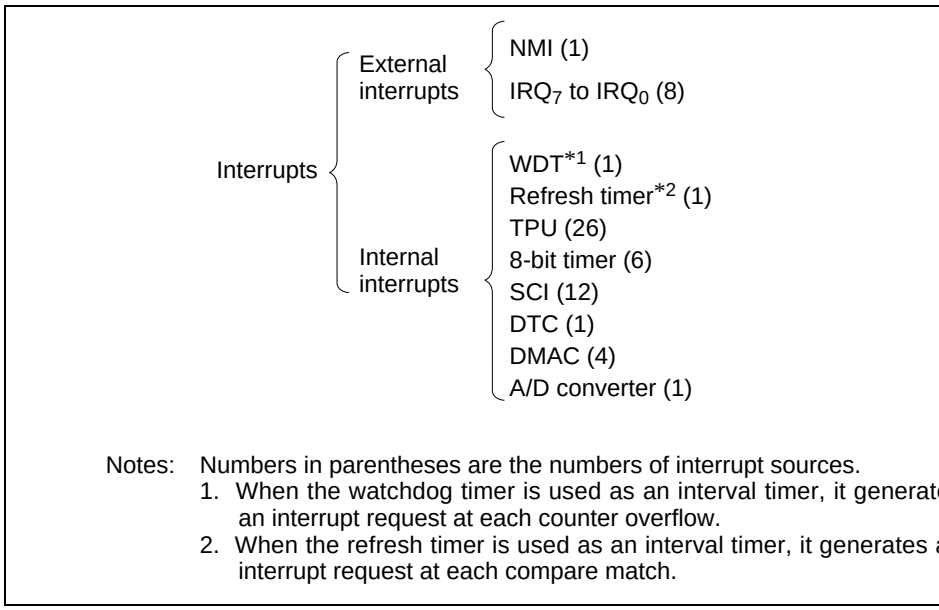
Interrupt Control Mode	CCR		EXR	
	I	UI	I2 to I0	
0	Trace exception handling cannot be used			
1				
2	1	—	—	
3	1	1	—	

Legend:
1: Set to 1
0: Cleared to 0
—: Retains value prior to execution.

source has a separate vector address.

NMI is the highest-priority interrupt, and is always accepted. Interrupts are controlled b
interrupt controller. The interrupt controller has four interrupt control modes and can as
interrupts other than NMI to either three or eight priority/mask levels to enable multiple
interrupt control.

For details of interrupts, see section 5, Interrupt Controller.



Notes: Numbers in parentheses are the numbers of interrupt sources.
1. When the watchdog timer is used as an interval timer, it generat
an interrupt request at each counter overflow.
2. When the refresh timer is used as an interval timer, it generates
interrupt request at each compare match.

Figure 4.4 Interrupt Sources and Number of Interrupts

Table 4.5 Status of CCR and EXR after Trap Instruction Exception Handling

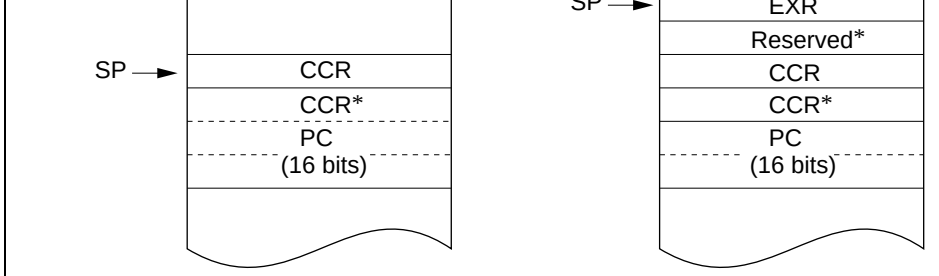
Interrupt Control Mode	CCR		EXR	
	I	UI	I2 to I0	
0	1	—	—	
1	1	1	—	
2	1	—	—	
3	1	1	—	

Legend:

1: Set to 1

0: Cleared to 0

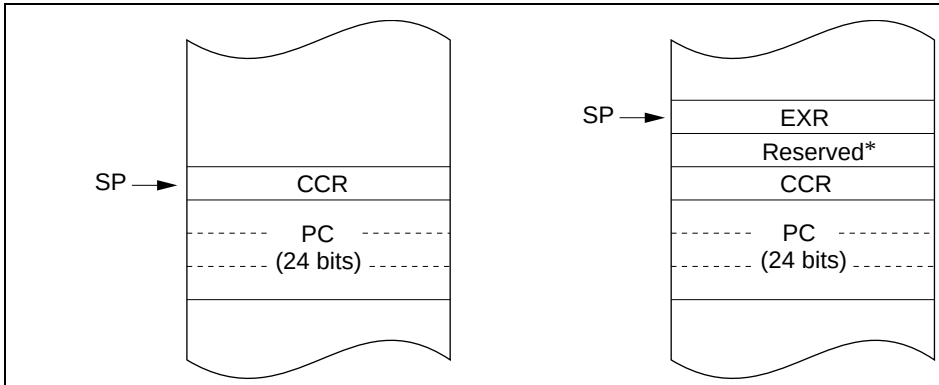
—: Retains value prior to execution.



(a) Interrupt control modes 0 and 1 (b) Interrupt control modes 2 ar

Note: * Ignored on return.

Figure 4.5 (1) Stack Status after Exception Handling (Normal Modes)



(a) Interrupt control modes 0 and 1 (b) Interrupt control modes 2 ar

Note: * Ignored on return.

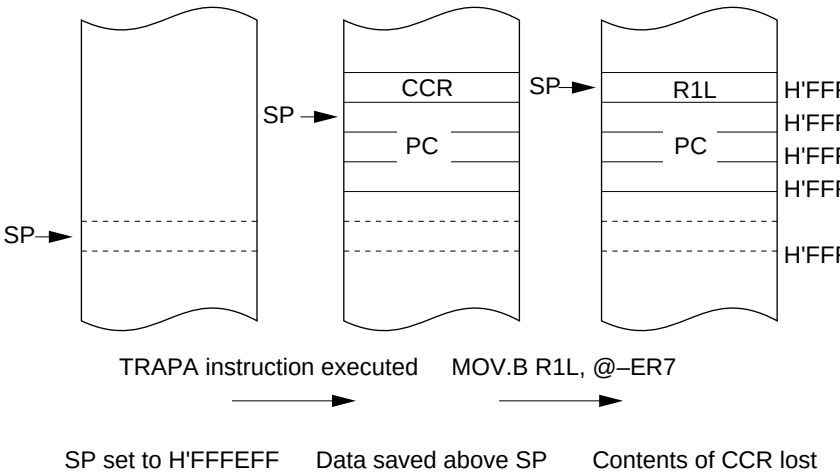
Figure 4.5 (2) Stack Status after Exception Handling (Advanced Modes

```
PUSH.L    ERn    (or MOV.L ERn, @-SP)
```

Use the following instructions to restore registers:

```
POP.W     Rn     (or MOV.W @SP+, Rn)
POP.L     ERn    (or MOV.L @SP+, ERn)
```

Setting SP to an odd value may lead to a malfunction. Figure 4.6 shows an example of
happens when the SP value is odd.



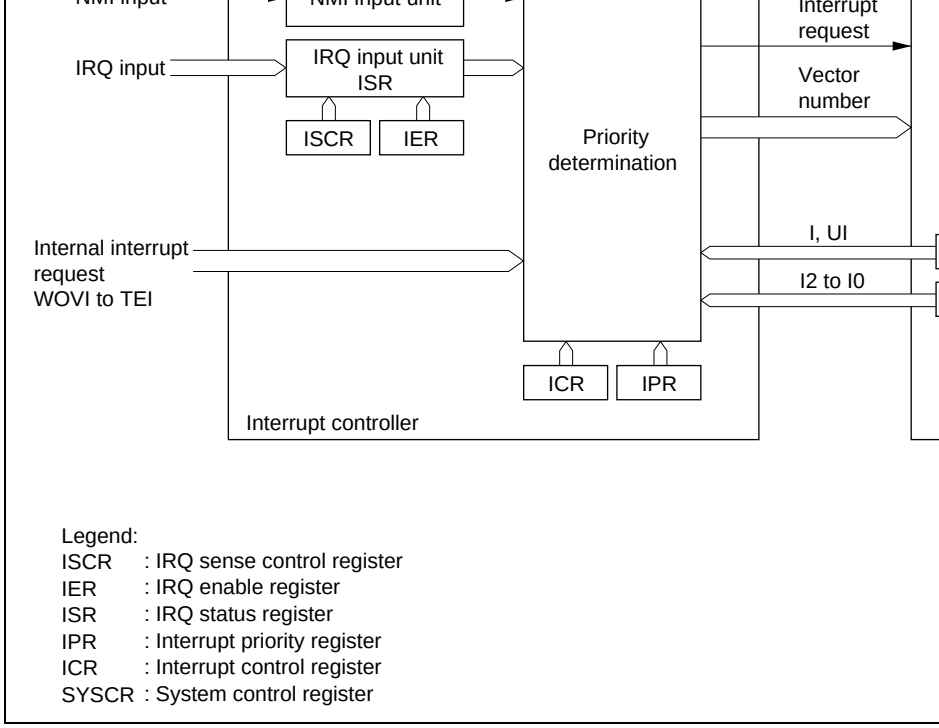
Legend: CCR: Condition code register
PC: Program counter
R1L: General register R1L
SP: Stack pointer

Note: This diagram illustrates an example in which the interrupt control mode is 0, in advanced mode.

Figure 4.6 Operation when SP Value Is Odd

controller has the following features:

- Four interrupt control modes
  - Any of four interrupt control modes can be set by means of the INTM1 and IN the system control register (SYSCR).
- Priorities settable with ICR
  - An interrupt control register (ICR) is provided for setting interrupt priorities. T levels can be set for each module for all interrupts except NMI.
- Priorities settable with IPR
  - An interrupt priority register (IPR) is provided for setting interrupt priorities. E levels can be set for each module for all interrupts except NMI.
  - NMI is assigned the highest priority level of 8, and can be accepted at all times
- Independent vector addresses
  - All interrupt sources are assigned independent vector addresses, making it unn the source to be identified in the interrupt handling routine.
- Nine external interrupts
  - NMI is the highest-priority interrupt, and is accepted at all times. Rising edge o edge can be selected for NMI.
  - Falling edge, rising edge, or both edge detection, or level sensing, can be selec to IRQ0.
- DTC and DMAC control
  - DTC and DMAC activation is performed by means of interrupts.



Legend:
ISCR : IRQ sense control register
IER : IRQ enable register
ISR : IRQ status register
IPR : Interrupt priority register
ICR : Interrupt control register
SYSCR : System control register

Figure 5.1 Block Diagram of Interrupt Controller

5.1.3 Pin Configuration

Table 5.1 summarizes the pins of the interrupt controller.

5.1.4 Register Configuration

Table 5.2 summarizes the registers of the interrupt controller.

Table 5.2 Interrupt Controller Registers

Name	Abbreviation	R/W	Initial Value	A
System control register	SYSCR	R/W	H'01	H
IRQ sense control register H	ISCRH	R/W	H'00	H
IRQ sense control register L	ISCRL	R/W	H'00	H
IRQ enable register	IER	R/W	H'00	H
IRQ status register	ISR	R/(W)*2	H'00	H
Interrupt control register A	ICRA	R/W	H'00	H
Interrupt control register B	ICRB	R/W	H'00	H
Interrupt control register C	ICRC	R/W	H'00	H
Interrupt priority register A	IPRA	R/W	H'77	H
Interrupt priority register B	IPRB	R/W	H'77	H
Interrupt priority register C	IPRC	R/W	H'77	H
Interrupt priority register D	IPRD	R/W	H'77	H
Interrupt priority register E	IPRE	R/W	H'77	H
Interrupt priority register F	IPRF	R/W	H'77	H
Interrupt priority register G	IPRG	R/W	H'77	H
Interrupt priority register H	IPRH	R/W	H'77	H
Interrupt priority register I	IPRI	R/W	H'77	H
Interrupt priority register J	IPRJ	R/W	H'77	H
Interrupt priority register K	IPRK	R/W	H'77	H

Notes: 1. Lower 16 bits of the address.
2. Can only be written with 0 for flag clearing.

R/W	.	R/W	—	R/W	R/W	R/W	—	—

SYSCR is an 8-bit readable/writable register that selects the interrupt control mode, and detected edge for NMI.

Only bits 5 to 3 are described here; for details of the other bits, see section 3.2.2, System Register (SYSCR).

SYSCR is initialized to H'01 by a reset and in hardware standby mode. It is not initializ software standby mode.

Bits 5 and 4—Interrupt Control Mode 1 and 0 (INTM1, INTM0): These bits select interrupt control modes for the interrupt controller.

Bit 5	Bit 4	Interrupt	
INTM1	**INTM0**	**Control Mode**	**Description**
0	0	0	Interrupts are controlled by I bit
	1	1	Interrupts are controlled by I and UI bits and ICR
1	0	2	Interrupts are controlled by bits I2 to I0, and IPR
	1	3	Interrupts are controlled by bits I, UI, and I2 to I0 IPR

Bit 3—NMI Edge Select (NMIEG): Selects the input edge for the NMI pin.

Bit 3	
NMIEG	**Description**
0	Interrupt request generated at falling edge of NMI input
1	Interrupt request generated at rising edge of NMI input

interrupts other than NMI.

The correspondence between ICR settings and interrupt sources is shown in table 5.3.

The ICR registers are initialized to H'00 by a reset and in hardware standby mode.

Table 5.3 Correspondence between Interrupt Sources and ICR Settings

	Bits						
Register	**7**	**6**	**5**	**4**	**3**	**2**	**1**
ICRA	IRQ_0	IRQ_1	IRQ_2 IRQ_3	IRQ_4 IRQ_5	IRQ_6 IRQ_7	DTC	Watchdo timer
ICRB	—	A/D converter	TPU channel 0	TPU channel 1	TPU channel 2	TPU channel 3	TPU channel 4
ICRC	8-bit timer channel 0	8-bit timer channel 1	DMAC	SCI channel 0	SCI channel 1	SCI channel 2	—

5.2.3 Interrupt Priority Registers A to K (IPRA to IPRK)

Bit	:	7	6	5	4	3	2	1
		—	IPR6	IPR5	IPR4	—	IPR2	IPR1
Initial value	:	0	1	1	1	0	1	1
R/W	:	—	R/W	R/W	R/W	—	R/W	R/W

The IPR registers are eleven 8-bit readable/writable registers that set priorities (levels
interrupts other than NMI.

The correspondence between IPR settings and interrupt sources is shown in table 5.4.

The IPR registers set a priority (level 7 to 0) for each interrupt source other than NMI.

The IPR registers are initialized to H'77 by a reset and in hardware standby mode.

	IRQ_3	IRQ_5
IPRC	IRQ_6 IRQ_7	DTC
IPRD	Watchdog timer	Refresh timer
IPRE	—	A/D converter
IPRF	TPU channel 0	TPU channel 1
IPRG	TPU channel 2	TPU channel 3
IPRH	TPU channel 4	TPU channel 5
IPRI	8-bit timer channel 0	8-bit timer channel 1
IPRJ	DMAC	SCI channel 0
IPRK	SCI channel 1	SCI channel 2

As shown in table 5.4, multiple interrupts are assigned to one IPR. Setting a value in th
from H'0 to H'7 in the 3-bit groups of bits 6 to 4 and 2 to 0 sets the priority of the corre
interrupt. The lowest priority level, level 0, is assigned by setting H'0, and the highest p
level, level 7, by setting H'7.

When interrupt requests are generated, the highest-priority interrupt according to the pr
levels set in the IPR registers is selected. This interrupt level is then compared with the
mask level set by the interrupt mask bits (I2 to I0) in the extend register (EXR) in the C
the priority level of the interrupt is higher than the set mask level, an interrupt request i
the CPU.

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

IER is initialized to H'00 by a reset and in hardware standby mode.

Bits 7 to 0—IRQ$_7$ to IRQ$_0$ Enable (IRQ7E to IRQ0E): These bits select whether IR are enabled or disabled.

Bit n

IRQnE	Description
0	IRQ$_n$ interrupts disabled
1	IRQ$_n$ interrupts enabled

Note: n = 7 to 0

5.2.5 IRQ Sense Control Registers H and L (ISCRH, ISCRL)

ISCRH

Bit	:	15	14	13	12	11	10	9
		IRQ7SCB	IRQ7SCA	IRQ6SCB	IRQ6SCA	IRQ5SCB	IRQ5SCA	IRQ4SCB
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

ISCRL

Bit	:	7	6	5	4	3	2	1
		IRQ3SCB	IRQ3SCA	IRQ2SCB	IRQ2SCA	IRQ1SCB	IRQ1SCA	IRQ0SCB
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

IRQ7SCB to IRQ0SCB	IRQ7SCA to IRQ0SCA	Description
0	0	Interrupt request generated at $\overline{IRQ}_7$ to $\overline{IRQ}_0$ input low le
	1	Interrupt request generated at falling edge of $\overline{IRQ}_7$ to $\overline{IF}$
1	0	Interrupt request generated at rising edge of $\overline{IRQ}_7$ to $\overline{IF}$
	1	Interrupt request generated at both falling and rising ed to $\overline{IRQ}_0$ input

5.2.6 IRQ Status Register (ISR)

Bit	:	7	6	5	4	3	2	1
		IRQ7F	IRQ6F	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F
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: * Only 0 can be written, to clear the flag.

ISR is an 8-bit readable/writable register that indicates the status of IRQ_7 to IRQ_0 interr requests.

ISR is initialized to H'00 by a reset and in hardware standby mode.

	(IRQnSCB = IRQnSCA = 0) and $\overline{IRQ}_n$ input is high • When IRQn interrupt exception handling is executed when falling, rising, edge detection is set (IRQnSCB = 1 or IRQnSCA = 1) • When the DTC is activated by an IRQ_n interrupt, and the DISEL bit in MR DTC is cleared to 0
1	[Setting conditions] • When $\overline{IRQ}_n$ input goes low when low-level detection is set (IRQnSCB = IR 0) • When a falling edge occurs in $\overline{IRQ}_n$ input when falling edge detection is s (IRQnSCB = 0, IRQnSCA = 1) • When a rising edge occurs in $\overline{IRQ}_n$ input when rising edge detection is se (IRQnSCB = 1, IRQnSCA = 0) • When a falling or rising edge occurs in $\overline{IRQ}_n$ input when both-edge detect (IRQnSCB = IRQnSCA = 1)

Note: n = 7 to 0

used to restore the H8S/2655 Group from software standby mode.

NMI Interrupt

NMI is the highest-priority interrupt, and is always accepted by the CPU regardless of t
the CPU interrupt mask bits. The NMIEG bit in SYSCR can be used to select whether a
is requested at a rising edge or a falling edge on the NMI pin.

The vector number for NMI interrupt exception handling is 7.

IRQ_7 to IRQ_0 Interrupts

Interrupts IRQ_7 to IRQ_0 are requested by an input signal at pins $\overline{IRQ}_7$ to $\overline{IRQ}_0$. Interrupts
IRQ_0 have the following features:

- Using ISCR, it is possible to select whether an interrupt is generated by a low level,
  edge, rising edge, or both edges, at pins $\overline{IRQ7}$ to $\overline{IRQ0}$.
- Enabling or disabling of interrupt requests IRQ7 to IRQ0 can be selected with IER.
- The interrupt control level can be set with ICR, and the interrupt priority level can b
  IPR.
- The status of interrupt requests IRQ_7 to IRQ_0 is indicated in ISR. ISR flags can be cl
  by software.

A block diagram of interrupts IRQ_7 to IRQ_0 is shown in figure 5.2.

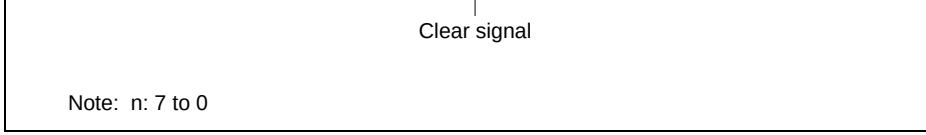


Figure 5.2 Block Diagram of Interrupts IRQ_7 to IRQ_0

Figure 5.3 shows the timing of setting IRQnF.

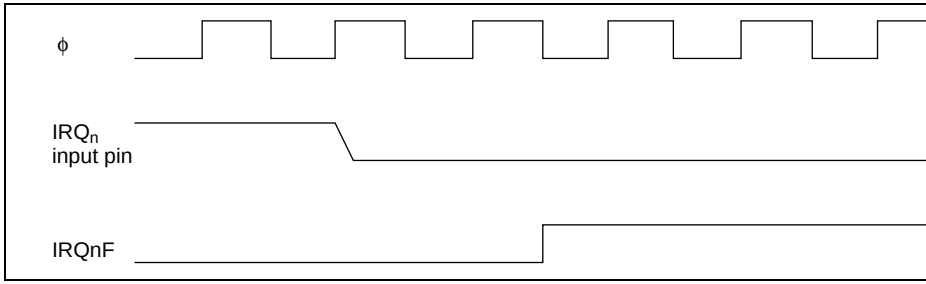


Figure 5.3 Timing of Setting IRQnF

The vector numbers for IRQ_7 to IRQ_0 interrupt exception handling are 23 to 16.

Detection of IRQ_7 to IRQ_0 interrupts does not depend on whether the relevant pin has
input or output. However, when a pin is used as an external interrupt input pin, do not
corresponding DDR to 0 and use the pin as an I/O pin for another function.

set by means of IPR.

- The DMAC and DTC can be activated by a TPU, 8-bit timer, SCI, or other interrup
  When the DMAC or DTC is activated by an interrupt, the interrupt control mode an
  mask bits are not affected.

5.3.3 Interrupt Exception Handling Vector Table

Table 5.5 shows interrupt exception handling sources, vector addresses, and interrupt p
For default priorities, the lower the vector number, the higher the priority.

Priorities among modules can be set by means of the ICR and IPR. The situation when
more modules are set to the same priority, and priorities within a module, are fixed as s
table 5.5.

IRQ_1		17	H'0022	H'0044	ICRA6	IPR 2 to
IRQ_2		18	H'0024	H'0048	ICRA5	IPR
IRQ_3		19	H'0026	H'004C		6 to
IRQ_4		20	H'0028	H'0050	ICRA4	IPR
IRQ_5		21	H'002A	H'0054		2 to
IRQ_6		22	H'002C	H'0058	ICRA3	IPR
IRQ_7		23	H'002E	H'005C		6 to
SWDTEND (software activation interrupt end)	DTC	24	H'0030	H'0060	ICRA2	IPR 2 to
WOVI (interval timer)	Watchdog timer	25	H'0032	H'0064	ICRA1	IPR 6 to
CMI (compare match)	Refresh controller	26	H'0034	H'0068	ICRA0	IPR 2 to
Reserved	—	27	H'0036	H'006C	ICRB7	IPR 6 to
ADI (A/D conversion end)	A/D	28	H'0038	H'0070	ICRB6	IPR 2 to
Reserved	—	29	H'003A	H'0074		
		30	H'003C	H'0078		
		31	H'003E	H'007C		
TGI0A (TGR0A input capture/compare match)	TPU channel 0	32	H'0040	H'0080	ICRB5	IPR 6 to
TGI0B (TGR0B input capture/compare match)		33	H'0042	H'0084		
TGI0C (TGR0C input capture/compare match)		34	H'0044	H'0088		
TGI0D (TGR0D input capture/compare match)		35	H'0046	H'008C		
TCI0V (overflow 0)		36	H'0048	H'0090		

TGI1B (TGR1B input capture/compare match)		41	H'0052	H'00A4		
TCI1V (overflow 1)		42	H'0054	H'00A8		
TCI1U (underflow 1)		43	H'0056	H'00AC		
TGI2A (TGR2A input capture/compare match)	TPU channel 2	44	H'0058	H'00B0	ICRB3	IPR 6 to
TGI2B (TGR2B input capture/compare match)		45	H'005A	H'00B4		
TCI2V (overflow 2)		46	H'005C	H'00B8		
TCI2U (underflow 2)		47	H'005E	H'00BC		
TGI3A (TGR3A input capture/compare match)	TPU channel 3	48	H'0060	H'00C0	ICRB2	IPR 2 to
TGI3B (TGR3B input capture/compare match)		49	H'0062	H'00C4		
TGI3C (TGR3C input capture/compare match)		50	H'0064	H'00C8		
TGI3D (TGR3D input capture/compare match)		51	H'0066	H'00CC		
TCI3V (overflow 1)		52	H'0068	H'00D0		
Reserved	—	53 54 55	H'006A H'006C H'006E	H'00D4 H'00D8 H'00DC		
TGI4A (TGR4A input capture/compare match)	TPU channel 4	56	H'0070	H'00E0	ICRB1	IPR 6 to
TGI4B (TGR4B input capture/compare match)		57	H'0072	H'00E4		
TCI4V (overflow 4)		58	H'0074	H'00E8		
TCI4U (underflow 4)		59	H'0076	H'00EC		

TCI5U (underflow 5)		63	H'007E	H'00FC		
CMIA0 (compare match A0)	8-bit timer channel 0	64	H'0080	H'0100	ICRC7	IPR 6 to
CMIB0 (compare match B0)		65	H'0082	H'0104		
OVI0 (overflow 0)		66	H'0084	H'0108		
Reserved	—	67	H'0086	H'010C		
CMIA1 (compare match A1)	8-bit timer channel 1	68	H'0088	H'0110	ICRC6	IPR 2 to
CMIB1 (compare match B1)		69	H'008A	H'0114		
OVI1 (overflow 1)		70	H'008C	H'0118		
Reserved	—	71	H'008E	H'011C		
DEND0A (channel 0/ channel 0A transfer end)	DMAC	72	H'0090	H'0120	ICRC5	IPR 6 to
DEND0B (channel 0B transfer end)		73	H'0092	H'0124		
DEND1A (channel 1/ channel 1A transfer end)		74	H'0094	H'0128		
DEND1B (channel 1B transfer end)		75	H'0096	H'012C		
Reserved	—	76	H'0098	H'0130		
		77	H'009A	H'0134		
		78	H'009C	H'0138		
		79	H'009E	H'013C		
ERI0 (receive error 0)	SCI channel 0	80	H'00A0	H'0140	ICRC4	IPR 2 to
RXI0 (reception completed 0)		81	H'00A2	H'0144		
TXI0 (transmit data empty 0)		82	H'00A4	H'0148		
TEI0 (transmission end 0)		83	H'00A6	H'014C		

ERI2 (receive error 2)	SCI channel 2	88	H'00B0	H'0160	ICRC2	IPRI 2 to
RXI2 (reception completed 2)		89	H'00B2	H'0164		
TXI2 (transmit data empty 2)		90	H'00B4	H'0168		
TEI2 (transmission end 2)		91	H'00B6	H'016C		

Note: * Lower 16 bits of the start address.

5.4 Interrupt Operation

5.4.1 Interrupt Control Modes and Interrupt Operation

Interrupt operations in the H8S/2655 Group differ depending on the interrupt control m

NMI interrupts are accepted at all times except in the reset state and the hardware stand the case of IRQ interrupts and on-chip supporting module interrupts, an enable bit is pr each interrupt. Clearing an enable bit to 0 disables the corresponding interrupt request. sources for which the enable bits are set to 1 are controlled by the interrupt controller.

Table 5.6 shows the interrupt control modes.

The interrupt controller performs interrupt control according to the interrupt control mo the INTM1 and INTM0 bits in SYSCR, the priorities set in ICR and IPR, and the mask indicated by the I and UI bits in the CPU's CCR, and bits I2 to I0 in EXR.

					Priority can be set wi
2	1	0	IPR	I2 to I0	8-level interrupt mask performed by bits I2 t 8 priority levels can b IPR.
3		1	ICR, IPR	I, UI, I2 to I0	Control is performed combination of interr set by the I and UI bi priority setting by ICR 8-level interrupt mask performed by bits I2 t 8-level priority setting

Figure 5.4 shows a block diagram of the priority decision circuit.

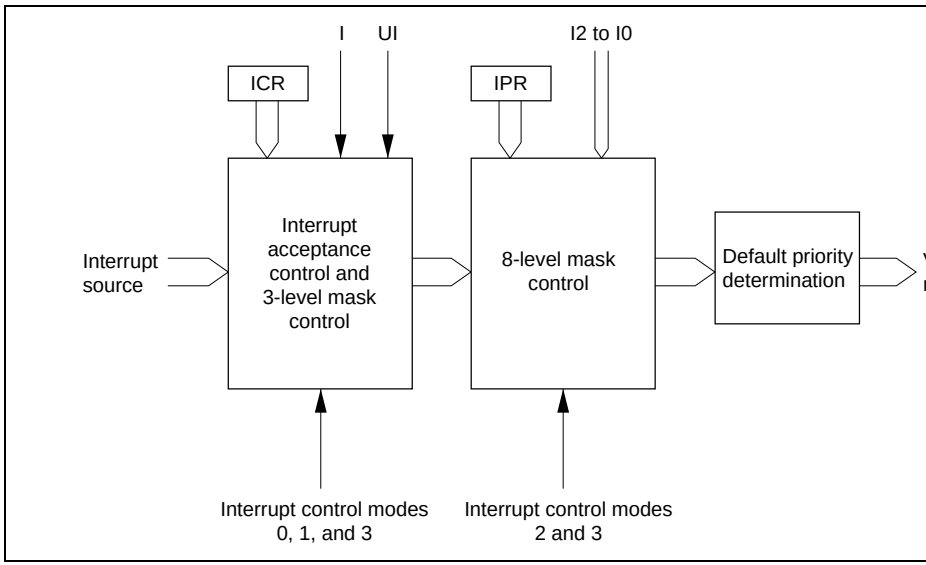


Figure 5.4 Block Diagram of Interrupt Control Operation

	Interrupt Mask Bits		
Interrupt Control Mode	**I**	**UI**	**Selected Interrupts**
0	0	*	All interrupts (control level 1 has
	1	*	NMI interrupts
1	0	*	All interrupts (control level 1 has
	1	0	NMI and control level 1 interrupt
		1	NMI interrupts
2	*	*	All interrupts
3	0	*	All interrupts
	1	0	NMI and control level 1 interrupt
		1	NMI interrupts

Legend:
*: Don't care

(2) 8-Level Control

In interrupt control modes 2 and 3, 8-level mask level determination is performed acco
interrupt priority level (IPR) for interrupts selected in interrupt acceptance control and
control.

The interrupt source selected is the interrupt with the highest priority level, and whose
level set in IPR is higher than the mask level.

Table 5.8 Interrupts Selected in Each Interrupt Control Mode (2)

Interrupt Control Mode	**Selected Interrupts**
0	All interrupts
1	
2	Highest-priority-level (IPR) interrupt whose priority level is
3	than the mask level (IPR > I2 to I0).

Interrupt sources with a lower priority than the accepted interrupt source are held pend

Table 5.9 shows operations and control signal functions in each interrupt control mod

Table 5.9 Operations and Control Signal Functions in Each Interrupt Control

Interrupt Control Mode	Setting		Interrupt Acceptance Control 3-Level Control				8-Level Control			Default Priority Determinat
	INTM1	INTM0		I	UI	ICR		I2 to I0	IPR	
0	0	0	○	IM	—	PR	X	—	—*2	○
1		1	○	IM	IM	PR	X	—	—*2	○
2	1	0	X	—*1	—	—	○	IM	PR	○
3		1	○	IM	IM	PR	○	IM	PR	○

Legend:

○ : Interrupt operation control performed

X : No operation. (All interrupts enabled)

IM : Used as interrupt mask bit

PR : Sets priority.

— : Not used.

Notes: 1. Set to 1 when interrupt is accepted.

2. Keep the initial setting.

[1] If an interrupt source occurs when the corresponding interrupt enable bit is set to 1, interrupt request is sent to the interrupt controller.

[2] When interrupt requests are sent to the interrupt controller, a control level 1 interrup according to the control level set in ICR, has priority for selection, and other interru are held pending. If a number of interrupt requests with the same control level settin generated at the same time, the interrupt request with the highest priority according priority system shown in table 5.5 is selected.

[3] The I bit is then referenced. If the I bit is cleared to 0, the interrupt request is accept bit is set to 1, only an NMI interrupt is accepted, and other interrupt requests are hel

[4] When an interrupt request is accepted, interrupt exception handling starts after exec current instruction has been completed.

[5] The PC and CCR are saved to the stack area by interrupt exception handling. The P the stack shows the address of the first instruction to be executed after returning fro interrupt handling routine.

[6] Next, the I bit in CCR is set to 1. This masks all interrupts except NMI.

[7] A vector address is generated for the accepted interrupt, and execution of the interru handling routine starts at the address indicated by the contents of that vector addres

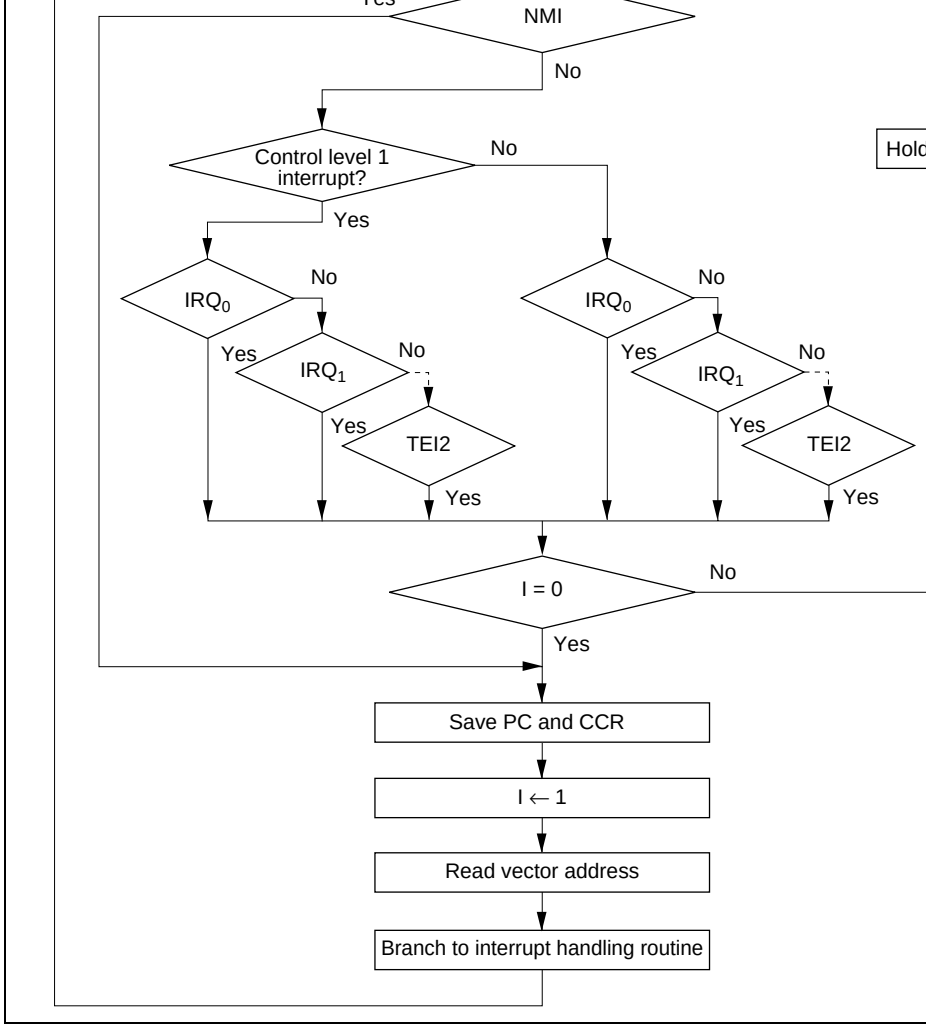


Figure 5.5 Flowchart of Procedure Up to Interrupt Acceptance in Interrupt Control Mode 0

disabled when both the I bit and the UI bit are set to 1.

For example, if the interrupt enable bit for an interrupt request is set to 1, and H'20, H'0 are set in ICRA, ICRB, and ICRC, respectively, (i.e. IRQ_2 and IRQ_3 interrupts are set to level 1 and other interrupts to control level 0), the situation is as follows:

- When I = 0, all interrupts are enabled
- (Priority order: NMI > IRQ_2 > IRQ_3 > IRQ_0 ...)
- When I = 1 and UI = 0, only NMI, IRQ2, and IRQ3 interrupts are enabled
- When I = 1 and UI = 1, only NMI interrupts are enabled

Figure 5.6 shows the state transitions in these cases.

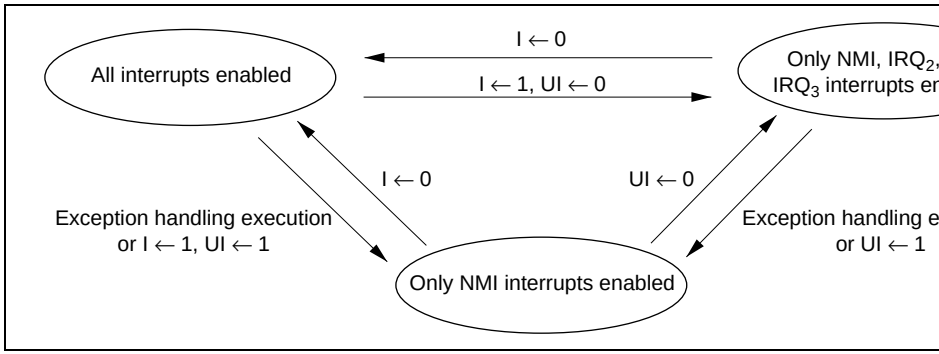


Figure 5.6 Example of State Transitions in Interrupt Control Mode 1

Figure 5.7 shows a flowchart of the interrupt acceptance operation in this case.

[3] The I bit is then referenced. If the I bit is cleared to 0, the UI bit is not affected.

An interrupt request set to interrupt control level 0 is accepted when the I bit is cle the I bit is set to 1, only an NMI interrupt is accepted, and other interrupt requests pending.

An interrupt request set to interrupt control level 1 has priority over an interrupt re interrupt control level 0, and is accepted if the I bit is cleared to 0, or if the I bits is the UI bit is cleared to 0.

When both the I bit and the UI bit are set to 1, only an NMI interrupt is accepted, a interrupt requests are held pending.

[4] When an interrupt request is accepted, interrupt exception handling starts after exe current instruction has been completed.

[5] The PC and CCR are saved to the stack area by interrupt exception handling. The the stack shows the address of the first instruction to be executed after returning fr interrupt handling routine.

[6] Next, the I and UI bits in CCR are set to 1. This masks all interrupts except NMI.

[7] A vector address is generated for the accepted interrupt, and execution of the inter handling routine starts at the address indicated by the contents of that vector addre

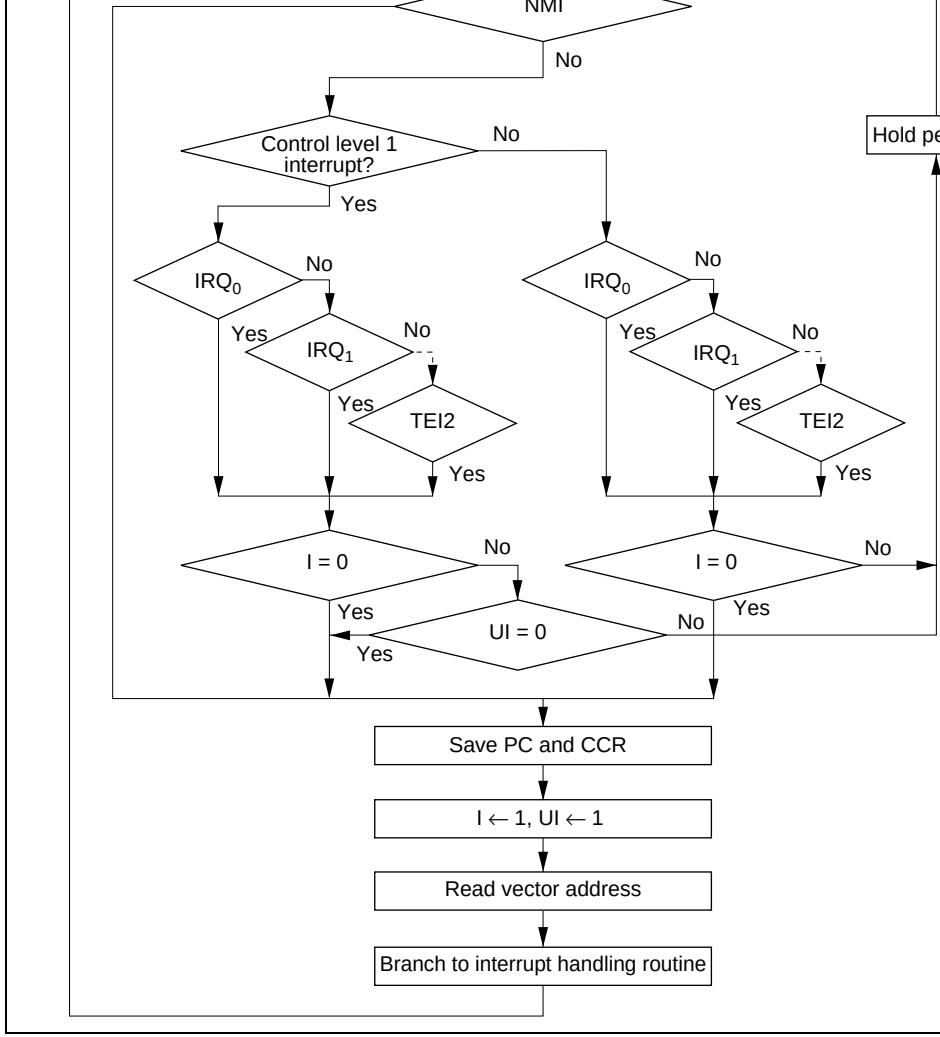


Figure 5.7 Flowchart of Procedure Up to Interrupt Acceptance in Interrupt Control Mode 1

[2] When interrupt requests are sent to the interrupt controller, the interrupt with the h
priority according to the interrupt priority levels set in IPR is selected, and lower-p
interrupt requests are held pending. If a number of interrupt requests with the same
generated at the same time, the interrupt request with the highest priority according
priority system shown in table 5.5 is selected.

[3] Next, the priority of the selected interrupt request is compared with the interrupt m
in EXR. An interrupt request with a priority no higher than the mask level set at th
held pending, and only an interrupt request with a priority higher than the interrupt
is accepted.

[4] When an interrupt request is accepted, interrupt exception handling starts after exe
current instruction has been completed.

[5] The PC, CCR, and EXR are saved to the stack area by interrupt exception handling
saved on the stack shows the address of the first instruction to be executed after re
the interrupt handling routine.

[6] The T bit in EXR is cleared to 0. The interrupt mask level is rewritten with the pri
the accepted interrupt.

If the accepted interrupt is NMI, the interrupt mask level is set to H'7.

[7] A vector address is generated for the accepted interrupt, and execution of the inter
handling routine starts at the address indicated by the contents of that vector addre

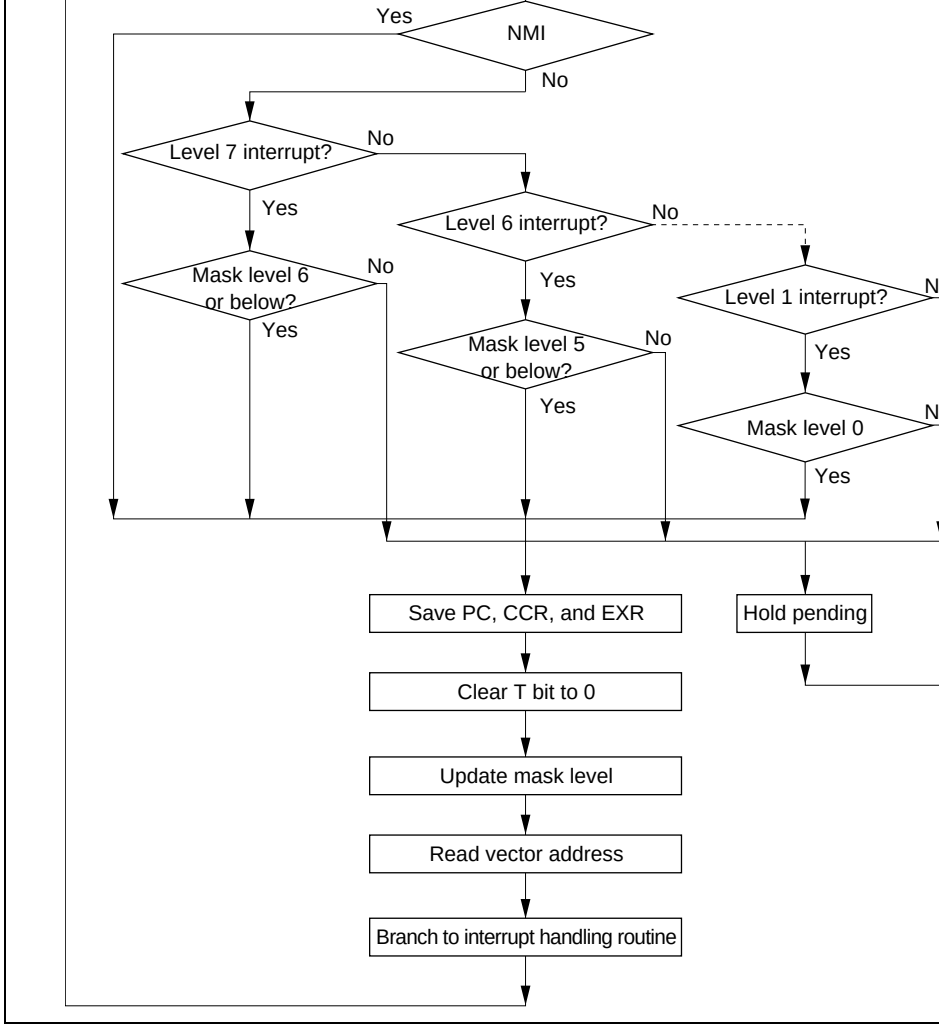


Figure 5.8 Flowchart of Procedure Up to Interrupt Acceptance in Interrupt Control Mode 2

- Control level 1 interrupt requests are enabled when the I bit or UI bit is cleared to
  disabled when both the I bit and the UI bit are set to 1.
- Eight-level priority control is performed when the I bit is cleared to 0.

For example, if the interrupt enable bit for an interrupt request is set to 1, and H'00, H
are set in ICRA, ICRB, and ICRC, respectively, (i.e. TPU channels 0 and 1 and SCI c
set to control level 1 and other interrupts to control level 0), the situation is as follows

- When I = 0, 8-level mask control is performed for all interrupts.
- The interrupt controller enables TPU0, TPU1, and SCI0 interrupts. Bits I2 to I0 ar
  and the interrupt mask level is regarded as 0.
- When I = 1 and UI = 1, only NMI interrupts are enabled.

Figure 5.9 shows the state transitions in these cases.

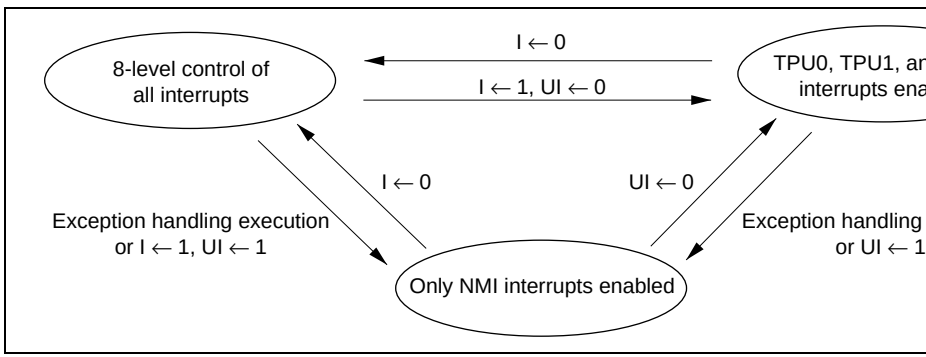


Figure 5.9 Example of State Transitions in Interrupt Control Mode

Figure 5.10 shows a flowchart of the interrupt acceptance operation in this case.

[3] The interrupt request with the highest priority according to the priority levels set in selected.

[4] If the I bit is cleared to 0, the priority level of the selected interrupt request is compa the interrupt mask level set in bits I2 to I0. An interrupt request with a priority no hi the mask level set at that time is held pending, and only an interrupt request with a p higher than the interrupt mask level is accepted.

[5] When an interrupt request is accepted, interrupt exception handling starts after exec current instruction has been completed.

[6] The PC, CCR, and EXR are saved to the stack area by interrupt exception handling. saved on the stack shows the address of the first instruction to be executed after retu the interrupt handling routine.

[7] Next, the I and UI bits in CCR are set to 1. This masks all interrupts except NMI. A to I0 are rewritten with the priority of the accepted interrupt. If the accepted interru the interrupt mask level is set to H'7.

[8] The T bit in EXR is cleared to 0.

[9] A vector address is generated for the accepted interrupt, and execution of the interru handling routine starts at the address indicated by the contents of that vector addres

[10]If interrupts are enabled again in the interrupt handling routine, the control level of interrupt to be enabled is set to 1, and the UI bit in CCR is cleared to 0. At control l interrupt with the highest priority according to the priority level is selected. Bits I2 disabled, and the interrupt mask level is regarded as 0.

When the I bit is cleared to 0, the control level is ignored and an interrupt with a pri higher than the mask level set in bits I2 to I0 is accepted.

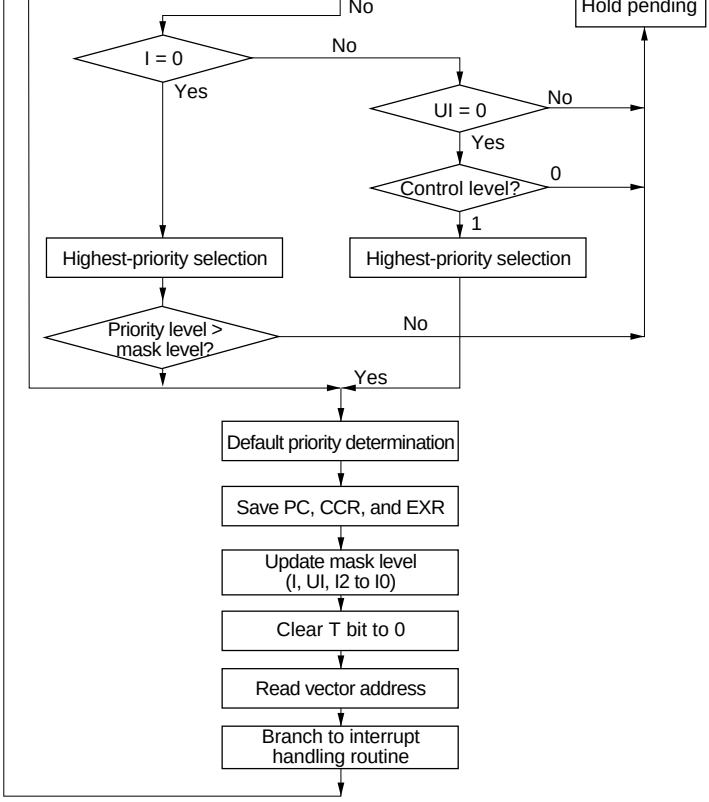


Figure 5.10 Flowchart of Procedure Up to Interrupt Acceptance in Interrupt Control Mode 3

5.4.6 Interrupt Exception Handling Sequence

Figure 5.11 shows the interrupt exception handling sequence. The example shown is f
where interrupt control mode 0 is set in advanced mode, and the program area and sta
in on-chip memory.

Figure 5.11 Interrupt Exception Handling

Table 5.10 Interrupt Response Times

No.	Execution Status	Normal Mode		Advanc
		INTM1 = 0	INTM1 = 1	INTM1 = 0
1	Interrupt priority determination*1	3	3	3
2	Number of wait states until executing instruction ends*2	1 to 19 + 2·S_I	1 to 19 + 2·S_I	1 to 19 + 2·S_I
3	PC, CCR, EXR stack save	2·S_K	3·S_K	2·S_K
4	Vector fetch	S_I	S_I	2·S_I
5	Instruction fetch*3	2·S_I	2·S_I	2·S_I
6	Internal processing*4	2	2	2
Total (using on-chip memory)		11 to 31	12 to 32	12 to 32

Notes: 1. Two states in case of internal interrupt.
2. Refers to DIVXS instruction.
3. Prefetch after interrupt acceptance and interrupt handling routine prefetch.
4. Internal processing after interrupt acceptance and internal processing after

Table 5.11 Number of States in Interrupt Handling Routine Execution Statuses

Symbol		Object of Access			
			External Device		
			8-Bit Bus		16-Bi
		Internal Memory	2-State Access	3-State Access	2-State Access
Instruction fetch	S_I	1	4	6 + 2m	2
Branch address read	S_J				
Stack manipulation	S_K				

Legend:
m: Number of wait states in an external device access.

MOV, if an interrupt is generated during execution of the instruction, the interrupt cond
still be enabled on completion of the instruction, and so interrupt exception handling fo
interrupt will be executed on completion of the instruction. However, if there is an inter
request of higher priority than that interrupt, interrupt exception handling will be execu
higher-priority interrupt, and the lower-priority interrupt will be ignored.

The same also applies when an interrupt source flag is cleared.

Figure 5.12 shows and example in which the CMIEA bit in 8-bit timer TCR is cleared t

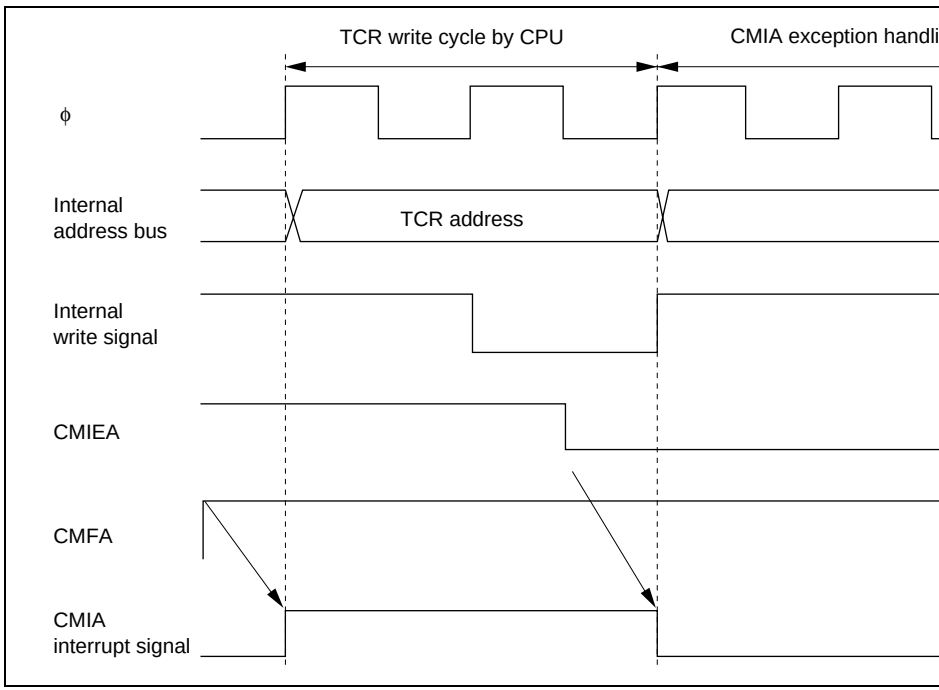


Figure 5.12 Contention between Interrupt Generation and Disabling

becomes valid two states after execution of the instruction ends.

5.5.3 Times when Interrupts Are Disabled

There are times when interrupt acceptance is disabled by the interrupt controller.

The interrupt controller disables interrupt acceptance for a 3-state period after the CPU updated the mask level with an LDC, ANDC, ORC, or XORC instruction.

5.5.4 Interrupts during Execution of EEPMOV Instruction

Interrupt operation differs between the EEPMOV.B instruction and the EEPMOV.W i

With the EEPMOV.B instruction, an interrupt request (including NMI) issued during is not accepted until the move is completed.

With the EEPMOV.W instruction, if an interrupt request is issued during the transfer, exception handling starts at a break in the transfer cycle. The PC value saved on the st case is the address of the next instruction.

Therefore, if an interrupt is generated during execution of an EEPMOV.W instruction following coding should be used.

```
L1:   EEPMOV.W
      MOV.W     R4,R4
      BNE       L1
```

- Activation request to DTC
- Activation request to DMAC
- Selection of a number of the above

For details of interrupt requests that can be used with to activate the DTC or DMAC, se
Data Transfer Controller, and section 7, DMA Controller.

5.6.2 Block Diagram

Figure 5.13 shows a block diagram of the DTC and DMAC interrupt controller.

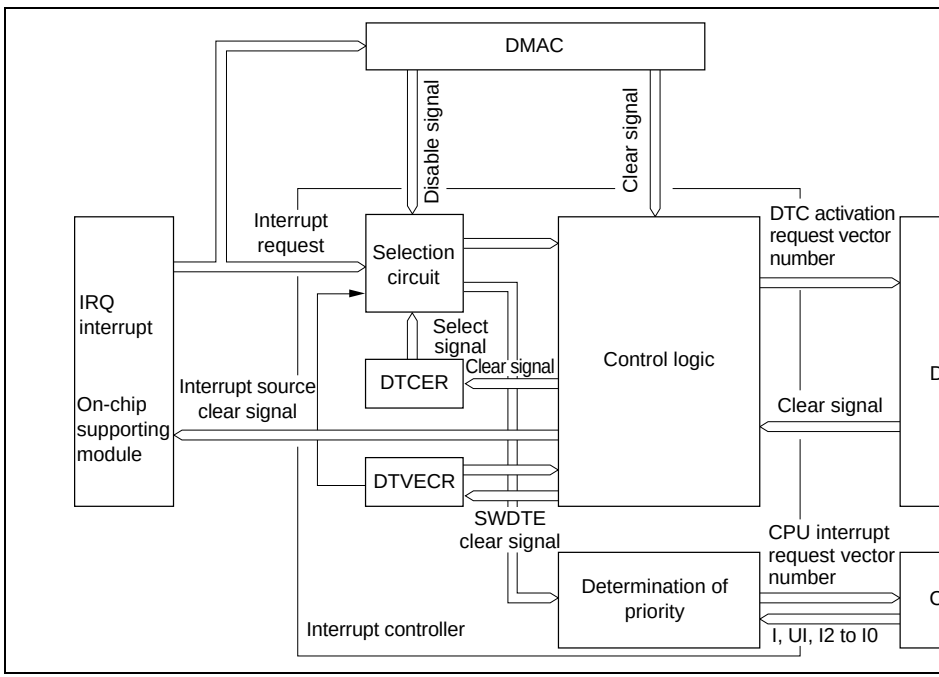


Figure 5.13 Interrupt Control for DTC and DMAC

activation source is to be managed by the DMAC can be selected with the DTA bit of
When the DTA bit is set to 1, the interrupt source constituting that DMAC activation s
a DTC activation source or CPU interrupt source.

For interrupt sources other than interrupts managed by the DMAC, it is possible to sel
activation request or CPU interrupt request with the DTCE bit of DTCEA to DTCEF i

After a DTC data transfer, the DTCE bit can be cleared to 0 and an interrupt request s
CPU in accordance with the specification of the DISEL bit of MRB in the DTC.

When the DTC has performed the specified number of data transfers and the transfer c
is zero, the DTCE bit is cleared to 0 and an interrupt request is sent to the CPU after t
transfer.

(2) Determination of Priority

The DTC activation source is selected in accordance with the default priority order, an
affected by mask or priority levels. See section 7.6, Interrupts, and section 8.3.3, DTC
Table, for the respective priorities.

With the DMAC, the activation source is input directly to each channel.

(3) Operation Order

If the same interrupt is selected as a DTC activation source and a CPU interrupt sourc
data transfer is performed first, followed by CPU interrupt exception handling.

If the same interrupt is selected as a DMAC activation source and a DTC activation so
interrupt source, operations are performed for them independently according to their r
operating statuses and bus mastership priorities.

0	0	*	○	×	
	1	0	○	◎	
		1	○	○	
1	*	*	◎	×	

Legend:

◎ : The relevant interrupt is used. Interrupt source clearing is performed.
(The CPU should clear the source flag in the interrupt handling routine.)

○ : The relevant interrupt is used. The interrupt source is not cleared.

× : The relevant bit cannot be used.

* : Don't care

(4) Notes on Use

SCI and A/D converter interrupt sources are cleared when the DMAC or DTC reads or the prescribed register, and are not dependent upon the DTA bit or DISEL bit.

The bus controller also has a bus arbitration function, and controls the operation of the masters: the CPU, DMA controller (DMAC), and data transfer controller (DTC).

6.1.1 Features

The features of the bus controller are listed below.

- Manages external address space in area units
  - In advanced mode, manages the external space as 8 areas of 128-kbytes/2-Mby
  - In normal mode, manages the external space as a single area
  - Bus specifications can be set independently for each area
  - DRAM/PSRAM/burst ROM interfaces can be set
- Basic bus interface
  - Chip select ($\overline{CS}_0$ to $\overline{CS}_7$) can be output for areas 0 to 7
  - 8-bit access or 16-bit access can be selected for each area
  - 2-state access or 3-state access can be selected for each area
  - Program wait states can be inserted for each area
- DRAM interface
  - DRAM interface can be set for areas 2 to 5 (in advanced mode)
  - Row address/column address multiplexed output (8/9/10 bits)
  - Two byte access methods (2-CAS and 2-WE)
  - Burst operation (fast page mode)
  - T_p cycle insertion to secure RAS precharging time
  - Choice of CAS-before-RAS refreshing or self-refreshing
- Pseudo-SRAM (PSRAM) direct interface
  - PSRAM interface can be set for areas 2 to 5 (in advanced mode)
  - Burst operation (static column mode)
  - T_p cycle insertion to secure RAS precharging time

external read cycle

- Write buffer functions
  - External write cycle and internal access can be executed in parallel
  - DMAC single-address mode and internal access can be executed in parallel
- Bus arbitration function
  - Includes a bus arbiter that arbitrates bus mastership among the CPU, DMAC, an
- Other features
  - Refresh counter (refresh timer) can be used as an interval timer
  - External bus release function

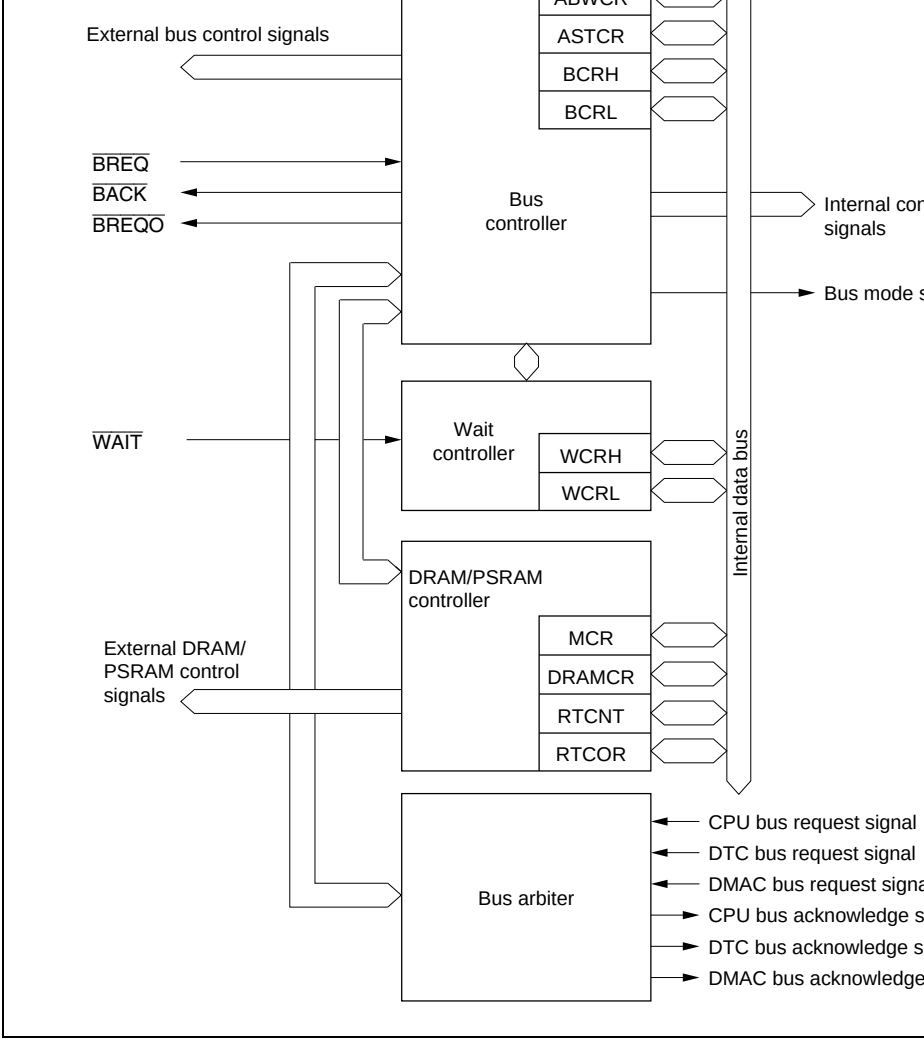


Figure 6.1 Block Diagram of Bus Controller

Read	$\overline{RD}$	Output	Strobe signal indicating that exte is being read.
High write/write enable/upper write enable	$\overline{HWR}$	Output	Strobe signal indicating that exte is to be written, and upper half (D data bus is enabled. 2-CAS DRAM write enable signa 2-WE DRAM upper write enable
Low write/lower column address strobe/lower write enable	$\overline{LWR}$	Output	Strobe signal indicating that exte is to be written, and lower half (D data bus is enabled. 2-CAS (LCASS = 1) DRAM lowe address strobe signal.* 2-WE DRAM lower write enable
Chip select 0	$\overline{CS_0}$	Output	Strobe signal indicating that area selected.
Chip select 1	$\overline{CS_1}$	Output	Strobe signal indicating that area selected.
Chip select 2/row address strobe 2	$\overline{CS_2}$	Output	Strobe signal indicating that area selected. DRAM row address strobe signa area 2 is in DRAM space.
Chip select 3/row address strobe 3	$\overline{CS_3}$	Output	Strobe signal indicating that area selected. DRAM row address strobe signa area 3 is in DRAM space.
Chip select 4/row address strobe 4	$\overline{CS_4}$	Output	Strobe signal indicating that area selected. DRAM row address strobe signa area 4 is in DRAM space.

			selected.
Upper column address strobe/ column address strobe/output enable/refresh	$\overline{CAS}$/ $\overline{OE}$	Output	2-CAS DRAM upper column ad strobe signal. 2-WE DRAM column address s PSRAM output enable signal w 2 to 5 are in PSRAM space.
Lower column strobe	$\overline{LCAS}$	Output	The 2-CAS type (LCASS = 0) D column address strobe signal.*
Wait	$\overline{WAIT}$	Input	Wait request signal when acces external 3-state access space.
Bus request	$\overline{BREQ}$	Input	Request signal that releases bu external device.
Bus request acknowledge	$\overline{BACK}$	Output	Acknowledge signal indicating been released.
Bus request output	$\overline{BREQO}$	Output	External bus request signal use internal bus master accesses e space when external bus is rele

Note: * Using the LCASS bit in BCRL, it is possible to select use of either the $\overline{LWR}$ $\overline{LCAS}$ pin for the 2-CAS type DRAM lower column strobe signal.

Bus width control register	ABWCR	R/W	H'FF/H'00*2	Retained	H'
Access state control register	ASTCR	R/W	H'FF	Retained	H'
Wait control register H	WCRH	R/W	H'FF	Retained	H'
Wait control register L	WCRL	R/W	H'FF	Retained	H'
Bus control register H	BCRH	R/W	H'D0	Retained	H'
Bus control register L	BCRL	R/W	H'3C	Retained	H'
Memory control register	MCR	R/W	H'00	Retained	H'
DRAM control register	DRAMCR	R/W	H'00	Retained	H'
Refresh timer/counter	RTCNT	R/W	H'00	Retained	H'
Refresh time constant register	RTCOR	R/W	H'FF	Retained	H'

Notes: 1. Lower 16 bits of the address.
2. Determined by the MCU operating mode.

Initial value	:	1	1	1	1	1	1	1
RW	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Mode 4								
Initial value	:	0	0	0	0	0	0	0
RW	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

ABWCR is an 8-bit readable/writable register that designates each area for either 8-bi 16-bit access.

ABWCR sets the data bus width for the external memory space. The bus width for on- memory and internal I/O registers is fixed regardless of the settings in ABWCR.

In normal mode, the settings of bits ABW7 to ABW1 have no effect on operation.

After a power-on reset and in hardware standby mode, ABWCR is initialized to H'FF to 3, and 5 to 7, and to H'00 in mode 4. It is not initialized by a manual reset or in soft standby mode.

Bits 7 to 0—Area 7 to 0 Bus Width Control (ABW7 to ABW0): These bits select w corresponding area is to be designated for 8-bit access or 16-bit access. In normal mod of area 0 is enabled, and the ABW0 bit selects whether external space is to be designa access or 16-bit access.

Bit n

ABWn	Description
0	Area n is designated for 16-bit access
1	Area n is designated for 8-bit access

Note: n = 7 to 0

space or a 3-state access space.

ASTCR sets the number of access states for the external memory space. The number of states for on-chip memory and internal I/O registers is fixed regardless of the settings in

In normal mode, the settings of bits AST7 to AST1 have no effect on operation.

ASTCR is initialized to H'FF by a power-on reset and in hardware standby mode. It is n initialized by a manual reset or in software standby mode.

Bits 7 to 0—Area 7 to 0 Access State Control (AST7 to AST0): These bits select wh corresponding area is to be designated as a 2-state access space or a 3-state access spac normal mode, only part of area 0 is enabled, and the AST0 bit selects whether external be designated for 2-state access or 3-state access.

Wait state insertion is enabled or disabled at the same time.

Bit n	
ASTn	**Description**
0	Area n is designated for 2-state access Wait state insertion in area n external space is disabled
1	Area n is designated for 3-state access (I Wait state insertion in area n external space is enabled

Note: n = 7 to 0

Program waits are not inserted in the case of on-chip memory or internal I/O registers.

WCRH and WCRL are initialized to H'FF by a power-on reset and in hardware standb
They are not initialized by a manual reset or in software standby mode.

(1) WCRH

Bit	:	7	6	5	4	3	2	1
		W71	W70	W61	W60	W51	W50	W41
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

Bits 7 and 6—Area 7 Wait Control 1 and 0 (W71, W70): These bits select the numb
program wait states when area 7 in external space is accessed while the AST7 bit in A
to 1.

Bit 7	Bit 6	
W71	**W70**	**Description**
0	0	Program wait not inserted when external space area 7 is acce
	1	1 program wait state inserted when external space area 7 is
1	0	2 program wait states inserted when external space area 7 is
	1	3 program wait states inserted when external space area 7 is

1	0	2 program wait states inserted when external space area 6 is a
	1	3 program wait states inserted when external space area 6 is a (I

Bits 3 and 2—Area 5 Wait Control 1 and 0 (W51, W50): These bits select the numb program wait states when area 5 in external space is accessed while the AST5 bit in AS to 1.

Bit 3	Bit 2	
W51	**W50**	**Description**
0	0	Program wait not inserted when external space area 5 is acces
	1	1 program wait state inserted when external space area 5 is a
1	0	2 program wait states inserted when external space area 5 is a
	1	3 program wait states inserted when external space area 5 is a (I

Bits 1 and 0—Area 4 Wait Control 1 and 0 (W41, W40): These bits select the numb program wait states when area 4 in external space is accessed while the AST4 bit in AS to 1.

Bit 1	Bit 0	
W41	**W40**	**Description**
0	0	Program wait not inserted when external space area 4 is acces
	1	1 program wait state inserted when external space area 4 is a
1	0	2 program wait states inserted when external space area 4 is a
	1	3 program wait states inserted when external space area 4 is a (I

program wait states when area 3 in external space is accessed while the AST3 bit in A
to 1.

Bit 7	Bit 6	
W31	**W30**	**Description**
0	0	Program wait not inserted when external space area 3 is acce
	1	1 program wait state inserted when external space area 3 is
1	0	2 program wait states inserted when external space area 3 is
	1	3 program wait states inserted when external space area 3 is

Bits 5 and 4—Area 2 Wait Control 1 and 0 (W21, W20): These bits select the numb
program wait states when area 2 in external space is accessed while the AST2 bit in A
to 1.

Bit 5	Bit 4	
W21	**W20**	**Description**
0	0	Program wait not inserted when external space area 2 is acce
	1	1 program wait state inserted when external space area 2 is
1	0	2 program wait states inserted when external space area 2 is
	1	3 program wait states inserted when external space area 2 is

1	0	2 program wait states inserted when external space area 1 is a
	1	3 program wait states inserted when external space area 1 is a (I

Bits 1 and 0—Area 0 Wait Control 1 and 0 (W01, W00): These bits select the numb program wait states when area 0 in external space is accessed while the AST0 bit in AS to 1.

Bit 1	Bit 0	
W01	**W00**	**Description**
0	0	Program wait not inserted when external space area 0 is acces
	1	1 program wait state inserted when external space area 0 is a
1	0	2 program wait states inserted when external space area 0 is a
	1	3 program wait states inserted when external space area 0 is a (I

6.2.4 Bus Control Register H (BCRH)

Bit	:	7	6	5	4	3	2	1	
		ICIS1	ICIS0	BRSTRM	BRSTS1	BRSTS0	RMTS2	RMTS1	R
Initial value	:	1	1	0	1	0	0	0	
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

BCRH is an 8-bit readable/writable register that selects enabling or disabling of idle cy insertion, and the memory interface for areas 2 to 5 and area 0.

BCRH is initialized to H'D0 by a power-on reset and in hardware standby mode. It is n initialized by a manual reset or in software standby mode.

Bit 6—Idle Cycle Insert 0 (ICIS0): Selects whether or not one idle cycle state is to l
between bus cycles when successive external read and external write cycles are perfor

Bit 6

ICIS0	Description
0	Idle cycle not inserted in case of successive external read and external w
1	Idle cycle inserted in case of successive external read and external write c

Bit 5—Burst ROM Enable (BRSTRM): Selects whether area 0 is used as a burst RO
interface. In normal mode, the selection can be made from the entire external space.

Burst ROM interface and PSRAM burst operation cannot be set at the same time.

Bit 5

BRSTRM	Description
0	Area 0 is basic bus interface
1	Area 0 is burst ROM interface

Bit 4—Burst Cycle Select 1 (BRSTS1): Selects the number of burst cycles for the bu
interface.

Bit 4

BRSTS1	Description
0	Burst cycle comprises 1 state
1	Burst cycle comprises 2 states

Bits 2 to 0—RAM Type Select (RMTS2 to RMTS0): These bits select the memory in areas 2 to 5 in advanced mode.

When DRAM space is selected, the relevant area is designated as DRAM interface, and PSRAM space is selected, it is designated as PSRAM interface.

Bit 2	Bit 1	Bit 0	Description			
RMTS2	RMTS1	RMTS0	Area 5	Area 4	Area 3	
0	0	0	Normal space			
		1	Normal space			DR
	1	0	Normal space		DRAM spac	
		1	DRAM space			
1	0	0	Normal space			
		1	Normal space			PSR
	1	0	Normal space		PSRAM spa	
		1	PSRAM space			

state protocol, the area partition unit, the LCAS signal, DMAC single address transfer
disabling of the write data buffer function, and enabling or disabling of $\overline{\text{WAIT}}$ pin inp

BCRL is initialized to H'3C by a power-on reset and in hardware standby mode. It is n
initialized by a manual reset or in software standby mode.

Bit 7—Bus Release Enable (BRLE): Enables or disables external bus release.

Bit 7

BRLE	Description
0	External bus release is disabled. $\overline{\text{BREQ}}$, $\overline{\text{BACK}}$, and $\overline{\text{BREQO}}$ can be used
1	External bus release is enabled.

Bit 6—BREQO Pin Enable (BREQOE): Outputs a signal that requests the external
to drop the bus request signal ($\overline{\text{BREQ}}$) in the external bus release state, when an intern
master performs an external space access, or when a refresh request is generated.

Bit 6

BREQOE	Description
0	$\overline{\text{BREQO}}$ output disabled. $\overline{\text{BREQO}}$ can be used as I/O port.
1	$\overline{\text{BREQO}}$ output enabled.

	of a reserved area* (in the H8S/2655)
1	Addresses H'010000 to H'01FFFF are external addresses (external expans or a reserved area* (single-chip mode) (

Note: * Reserved areas should not be accessed.

Bit 4—LCAS Select (LCASS): Selects use of the $\overline{\text{LWR}}$ pin or the $\overline{\text{LCAS}}$ pin for the 2-
DRAM interface $\overline{\text{LCAS}}$ signal.

Bit 4

LCASS	Description
0	$\overline{\text{LCAS}}$ pin used for 2-CAS type DRAM interface $\overline{\text{LCAS}}$ signal ($\overline{\text{BREQO}}$ output and $\overline{\text{WAIT}}$ input cannot be used when $\overline{\text{LCAS}}$ signal is used
1	$\overline{\text{LWR}}$ pin used for 2-CAS type DRAM interface $\overline{\text{LCAS}}$ signal (RAS down mode cannot be used) (

Bit 3—DACK Timing Select (DDS): Selects the DMAC single address transfer bus ti
the DRAM interface or PSRAM interface.

Bit 3

DDS	Description
0	When DMAC single address transfer is performed in DRAM/PSRAM space is always executed $\overline{\text{DACK}}$ signal goes low from T_r or T_1 cycle
1	Burst access is possible when DMAC single address transfer is performed DRAM/PSRAM space $\overline{\text{DACK}}$ signal goes low from T_{c1} or T_2 cycle (

used for an external write cycle or DMAC single address cycle.

Bit 1

WDBE	Description
0	Write data buffer function not used
1	Write data buffer function used

Bit 0—WAIT Pin Enable (WAITE): Selects enabling or disabling of wait input by t
pin.

Bit 0

WAITE	Description
0	Wait input by $\overline{\text{WAIT}}$ pin disabled. $\overline{\text{WAIT}}$ pin can be used as I/O port.
1	Wait input by $\overline{\text{WAIT}}$ pin enabled

6.2.6 Memory Control Register (MCR)

Bit	:	7	6	5	4	3	2	1
		TPC	BE	RCDM	CW2	MXC1	MXC0	RLW1
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

MCR is an 8-bit readable/writable register that selects the DRAM strobe control meth
of precharge cycles, access mode, address multiplexing shift size, and the number of w
inserted during refreshing, when areas 2 to 5 are designated as DRAM interface. Whe
5 are designated as PSRAM interface, MCR selects the number of precharge cycles an
mode for PSRAM.

0	1-state precharge cycle is inserted	(
1	2-state precharge cycle is inserted	

Bit 6—Burst Access Enable (BE): Selects enabling or disabling of burst access to area designated as DRAM space or PSRAM space. DRAM space burst access is performed mode, and PSRAM access in static column mode.

PSRAM burst operation and burst ROM interface cannot be set at the same time.

Bit 6

BE	Description	
0	Burst disabled (always full access)	(
1	• For DRAM space access Access in fast page mode • For PSRAM space access Access in static column mode	

Bit 5—RAS Down Mode (RCDM): When areas 2 to 5 are designated as DRAM space to DRAM is interrupted, RCDM selects whether the next DRAM access is waited for w $\overline{RAS}$ signal held low (RAS down mode), or the $\overline{RAS}$ signal is driven high again (RAS u RAS down mode cannot be used with the 2-CAS method (LCASS=1). When selecting mode, set the BE bit to 1.

When areas 2 to 5 are designated as PSRAM space, this bit is invalid.

Bit 5

RCDM	Description	
0	DRAM interface: RAS up mode selected	(
1	DRAM interface: RAS down mode selected	

Bits 3 and 2—Multiplex Shift Count 1 and 0 (MXC1, MXC0): These bits select the shift to the lower half of the row address in row address/column address multiplexing DRAM interface. In burst operation on the DRAM/PSRAM interface, these bits also s row address to be used for comparison.

Bit 3	Bit 2	
MXC1	**MXC0**	**Description**
0	0	8-bit shift • When 8-bit access space is designated: Row address A_{23} for comparison • When 16-bit access space is designated: Row address A_{2} for comparison
	1	9-bit shift • When 8-bit access space is designated: Row address A_{23} for comparison • When 16-bit access space is designated: Row address A_{2} for comparison
1	0	10-bit shift • When 8-bit access space is designated: Row address A_{23} for comparison • When 16-bit access space is designated: Row address A_{2} for comparison
	1	—

1	0	2 wait states inserted
	1	3 wait states inserted

6.2.7 DRAM Control Register (DRAMCR)

Bit	:	7	6	5	4	3	2	1	
		RFSHE	RCW	RMODE	CMF	CMIE	CKS2	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	

DRAMCR is an 8-bit readable/writable register that selects the DRAM refresh mode ar
counter clock, and controls the refresh timer.

DRAMCR is initialized to H'00 by a power-on reset and in hardware standby mode. It i
initialized by a manual reset or in software standby mode.

Bit 7—Refresh Control (RFSHE): Selects whether or not refresh control is performed
refresh control is not performed, the refresh timer can be used as an interval timer. Refr
is not performed in normal mode.

Bit 7

RFSHE	Description
0	Refresh control is not performed
1	Refresh control is performed

	RAS falls in T_{Rc1} cycle

Bit 5—Refresh Mode (RMODE): When refresh control is performed (RFSHE = 1), t
selects whether normal refreshing (CAS-before-RAS refreshing for the DRAM interf
refreshing for the PSRAM interface) or self-refreshing is performed.

Bit 5

RMODE	Description
0	• DRAM interface CAS-before-RAS refreshing used • PSRAM interface Auto-refreshing used (I
1	Self-refreshing used

Bit 4—Compare Match Flag (CMF): Status flag that indicates a match between the
RTCNT and RTCOR.

When refresh control is performed (RFSHE = 1), 1 should be written to the CMF bit v
to DRAMCR.

Bit 4

CMF	Description
0	[Clearing condition] Cleared by reading the CMF flag when CMF = 1, then writing 0 to the CM
1	[Setting condition] Set when RTCNT = RTCOR

1	Interrupt request (CMI) by CMF flag enabled

Bits 2 to 0—Refresh Counter Clock Select (CKS2 to CKS0): These bits select the cl
input to RTCNT from among 7 internal clocks obtained by dividing the system clock (
the input clock is selected with bits CKS2 to CKS0, RTCNT begins counting up.

Bit 2	Bit 1	Bit 0	
CKS2	**CKS1**	**CKS0**	**Description**
0	0	0	Count operation disabled
		1	Count uses ϕ/2
	1	0	Count uses ϕ/8
		1	Count uses ϕ/32
1	0	0	Count uses ϕ/128
		1	Count uses ϕ/512
	1	0	Count uses ϕ/2048
		1	Count uses ϕ/4096

RTCNT counts up using the internal clock selected by bits CKS2 to CKS0 in DRAMC

When RTCNT matches RTCOR (compare match), the CMF flag in DRAMCR is set t
RTCNT is cleared to H'00. If the RFSHE bit in DRAMCR is set to 1 at this time, a ref
started. Also, if the CMIE bit in DRAMCR is set to 1, a compare match interrupt (CM
generated.

RTCNT is initialized to H'00 by a power-on reset and in hardware standby mode. It is
initialized by a manual reset or in software standby mode.

6.2.9 Refresh Time Constant Register (RTCOR)

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

RTCOR is an 8-bit readable/writable register that sets the period for compare match o
with RTCNT.

The values of RTCOR and RTCNT are constantly compared, and if they match, the C
DRAMCR is set to 1 and RTCNT is cleared to H'00.

RTCOR is initialized to H'FF by a power-on reset and in hardware standby mode. It is
initialized by a manual reset or in software standby mode.

Chip select signals ($\overline{CS}_0$ to $\overline{CS}_7$) can be output for each area.

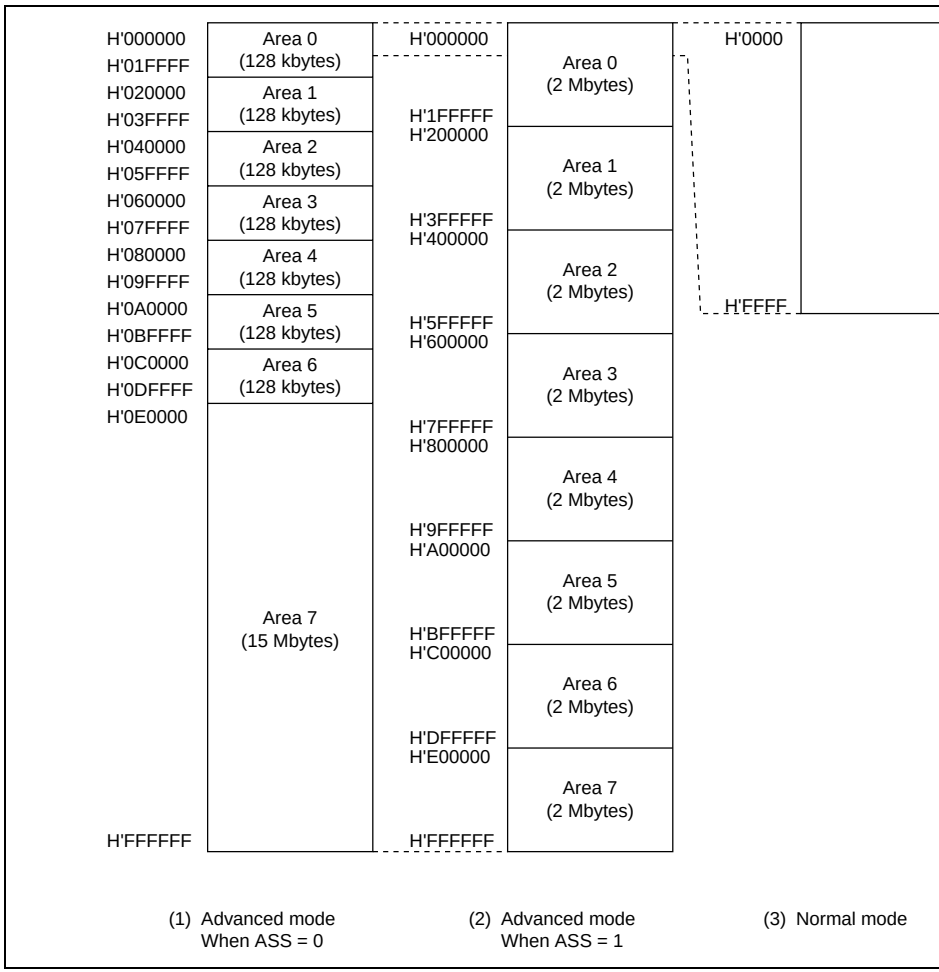


Figure 6.2 Overview of Area Partitioning

A bus width of 8 or 16 bits can be selected with ADWCR. An area for which an 8-
selected functions as an 8-bit access space, and an area for which a 16-bit bus is se
functions as a16-bit access space.

If all areas are designated for 8-bit access, 8-bit bus mode is set; if any area is desi
16-bit access, 16-bit bus mode is set. When the burst ROM interface is designated,
mode is always set.

(2) Number of Access States

Two or three access states can be selected with ASTCR. An area for which 2-state
selected functions as a 2-state access space, and an area for which 3-state access is
functions as a 3-state access space.

With the DRAM/PSRAM interface and burst ROM interface, the number of acces
be determined without regard to ASTCR.

When 2-state access space is designated, wait insertion is disabled.

(3) Number of Program Wait States

When 3-state access space is designated by ASTCR, the number of program wait s
inserted automatically is selected with WCRH and WCRL. From 0 to 3 program w
can be selected.

Table 6.3 shows the bus specifications for each basic bus interface area.

		1	0			2
			1			3
1	0	—	—	8	2	0
	1	0	0		3	0
			1			1
		1	0			2
			1			3

6.3.3 Memory Interfaces

The H8S/2655 Group memory interfaces comprise a basic bus interface that allows dire
connection of ROM, SRAM, and so on; a DRAM interface that allows direct connectio
DRAM; a PSRAM interface that allows direct connection of PSRAM; and a burst ROM
that allows direct connection of burst ROM. The interface can be selected independentl
area.

An area for which the basic bus interface is designated functions as normal space, an ar
which the DRAM interface is designated functions as DRAM space, an area for which
PSRAM interface is designated functions as PSRAM space, and an area for which the b
interface is designated functions as burst ROM space.

Area 0 includes on-chip ROM, and in ROM-disabled expansion mode, all of area 0 is
space. In ROM-enabled expansion mode, the space excluding on-chip ROM is externa

When area 0 external space is accessed, the $\overline{CS}_0$ signal can be output.

Either basic bus interface or burst ROM interface can be selected for area 0.

The size of area 0 is switched between 128 kbytes and 2 Mbytes according to the state
bit.

Areas 1 and 6

In external expansion mode, all of areas 1 and 6 is external space.

When area 1 and 6 external space is accessed, the $\overline{CS}_1$ and $\overline{CS}_6$ pin signals respectivel
output.

Only the basic bus interface can be used for areas 1 and 6.

The size of areas 1 and 6 is switched between 128 kbytes and 2 Mbytes according to t
the ASS bit.

Areas 2 to 5

In external expansion mode, all of areas 2 to 5 is external space.

When area 2 to 5 external space is accessed, signals $\overline{CS}_2$ to $\overline{CS}_5$ can be output.

Basic bus interface, DRAM interface, or PSRAM interface can be selected for areas 2
the DRAM interface, signals $\overline{CS}_2$ to $\overline{CS}_5$ are used as $\overline{RAS}$ signals.

The size of areas 2 to 5 is switched between 128 kbytes and 2 Mbytes according to the
ASS bit.

Only the basic bus interface can be used for the area 7 memory interface.

The size of area 7 is switched between 15 Mbytes and 2 Mbytes according to the state o
bit.

6.3.5 Areas in Normal Mode

In normal mode, a 64-kbyte address space comprising part of area 0 is controlled. Area
partitioning is not performed in normal mode. In ROM-disabled expansion mode, the s
excluding the on-chip RAM and internal I/O registers is external space. In ROM-enable
expansion mode the space excluding the on-chip ROM, on-chip RAM, and internal I/O
external space. The on-chip RAM is enabled when the RAME bit in the system control
(SYSCR) is set to 1; when the RAME bit is cleared to 0, the on-chip RAM is disabled a
corresponding space becomes external space.

When external space is accessed, the $\overline{CS}_0$ signal can be output.

The basic bus interface or burst ROM interface can be selected.

for the port corresponding to the particular $\overline{CS}_n$ pin.

In ROM-disabled expansion mode, the $\overline{CS}_0$ pin is placed in the output state after a pov
Pins $\overline{CS}_1$ to $\overline{CS}_7$ are placed in the input state after a power-on reset, and so the correspo
should be set to 1 when outputting signals $\overline{CS}_1$ to $\overline{CS}_7$.

In ROM-enabled expansion mode, pins $\overline{CS}_0$ to $\overline{CS}_7$ are all placed in the input state afte
reset, and so the corresponding DDR should be set to 1 when outputting signals $\overline{CS}_0$ to

For details, see section 9, I/O Ports.

When areas 2 to 5 are designated as DRAM space, outputs $\overline{CS}_2$ to $\overline{CS}_5$ are used as $\overline{RAS}$

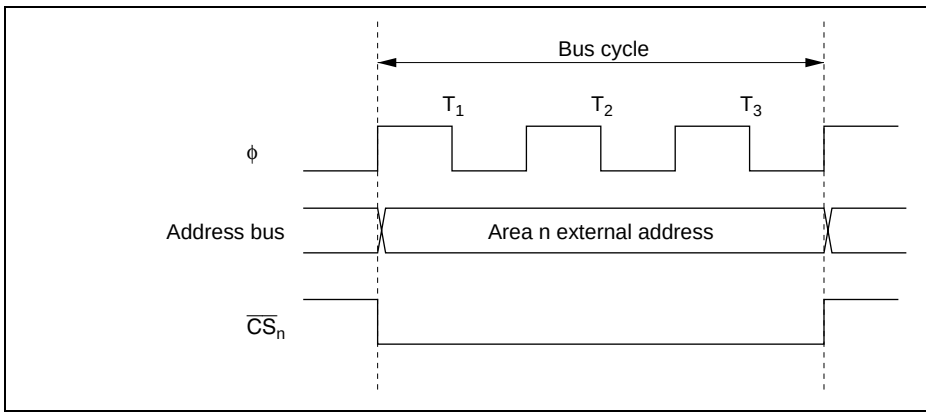


Figure 6.3 $\overline{CS}_n$ Signal Output Timing (n = 0 to 7)

6.4.2 Data Size and Data Alignment

Data sizes for the CPU and other internal bus masters are byte, word, and longword. Th
controller has a data alignment function, and when accessing external space, controls w
upper data bus (D_{15} to D_8) or lower data bus (D_7 to D_0) is used according to the bus spec
for the area being accessed (8-bit access space or 16-bit access space) and the data size.

8-Bit Access Space

Figure 6.4 illustrates data alignment control for the 8-bit access space. With the 8-bit ac
the upper data bus (D_{15} to D_8) is always used for accesses. The amount of data that can b
at one time is one byte: a word transfer instruction is performed as two byte accesses, a
longword transfer instruction, as four byte accesses.

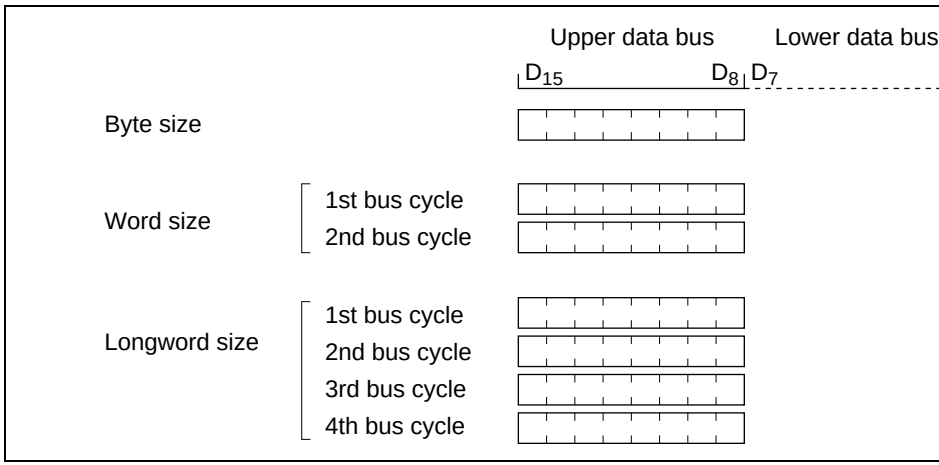


Figure 6.4 Access Sizes and Data Alignment Control (8-Bit Access Spac

address.

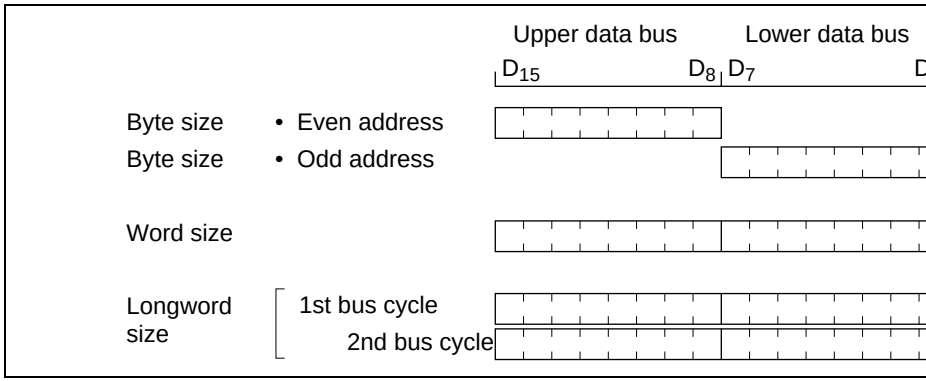


Figure 6.5 Access Sizes and Data Alignment Control (16-Bit Access Spa

Table 6.4 Data Buses Used and Valid Strobes

Area	Access Size	Read/ Write	Address	Valid Strobe	Upper Data Bus (D_{15} to D_8)	Lower D (D_7 to D
8-bit access space	Byte	Read	—	$\overline{RD}$	Valid	Invalid
		Write	—	$\overline{HWR}$		Undefin
16-bit access space	Byte	Read	Even	$\overline{RD}$	Valid	Invalid
			Odd		Invalid	Valid
		Write	Even	$\overline{HWR}$	Valid	Undefin
			Odd	$\overline{LWR}$	Undefined	Valid
	Word	Read	—	$\overline{RD}$	Valid	Valid
		Write	—	$\overline{HWR}$, $\overline{LWR}$	Valid	Valid

Note: Undefined: Undefined data is output.
Invalid: Input state; input value is ignored.

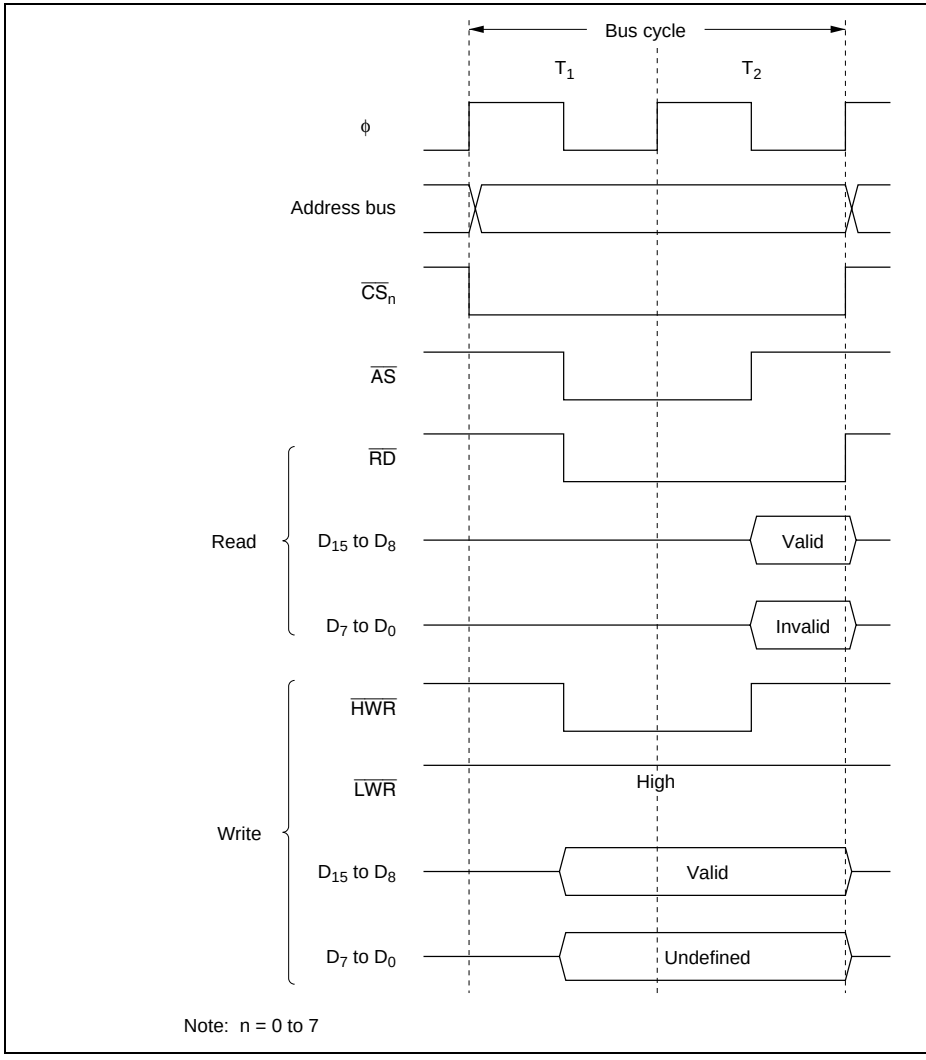


Figure 6.6 Bus Timing for 8-Bit 2-State Access Space

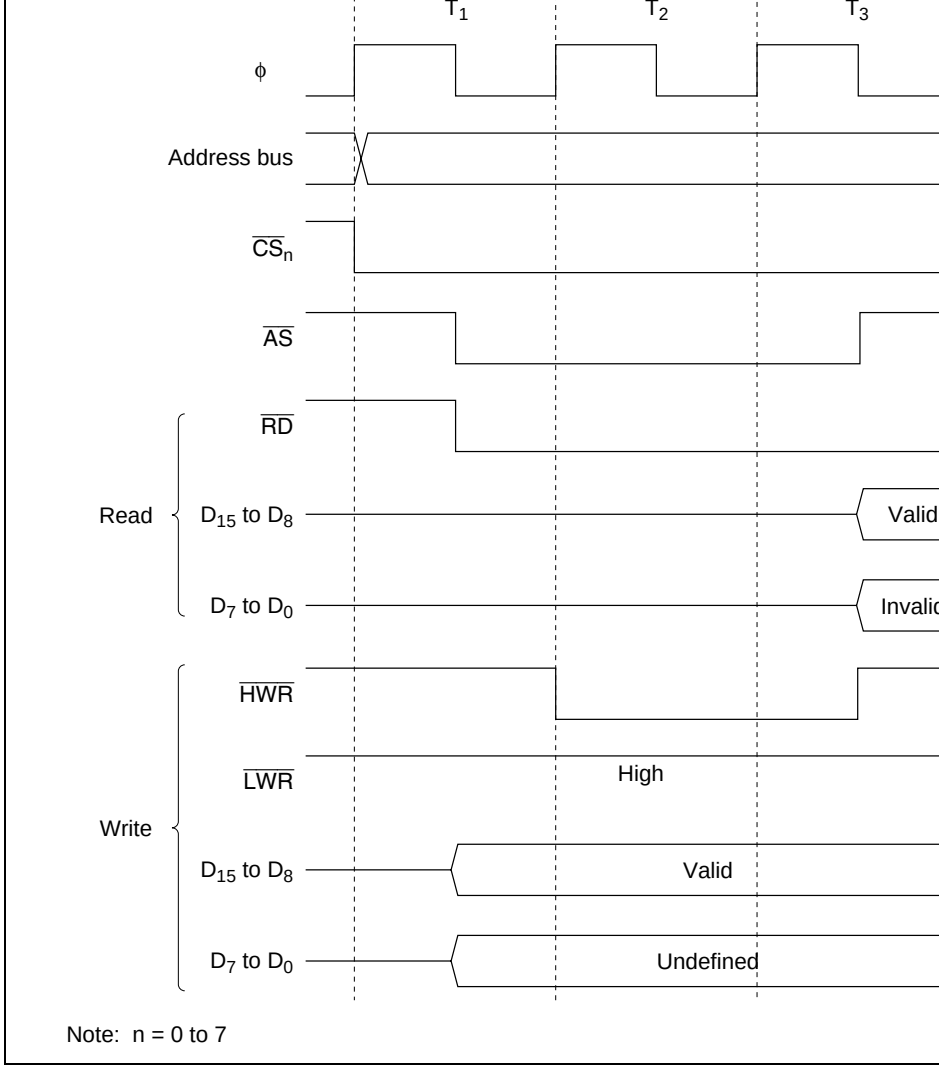


Figure 6.7 Bus Timing for 8-Bit 3-State Access Space

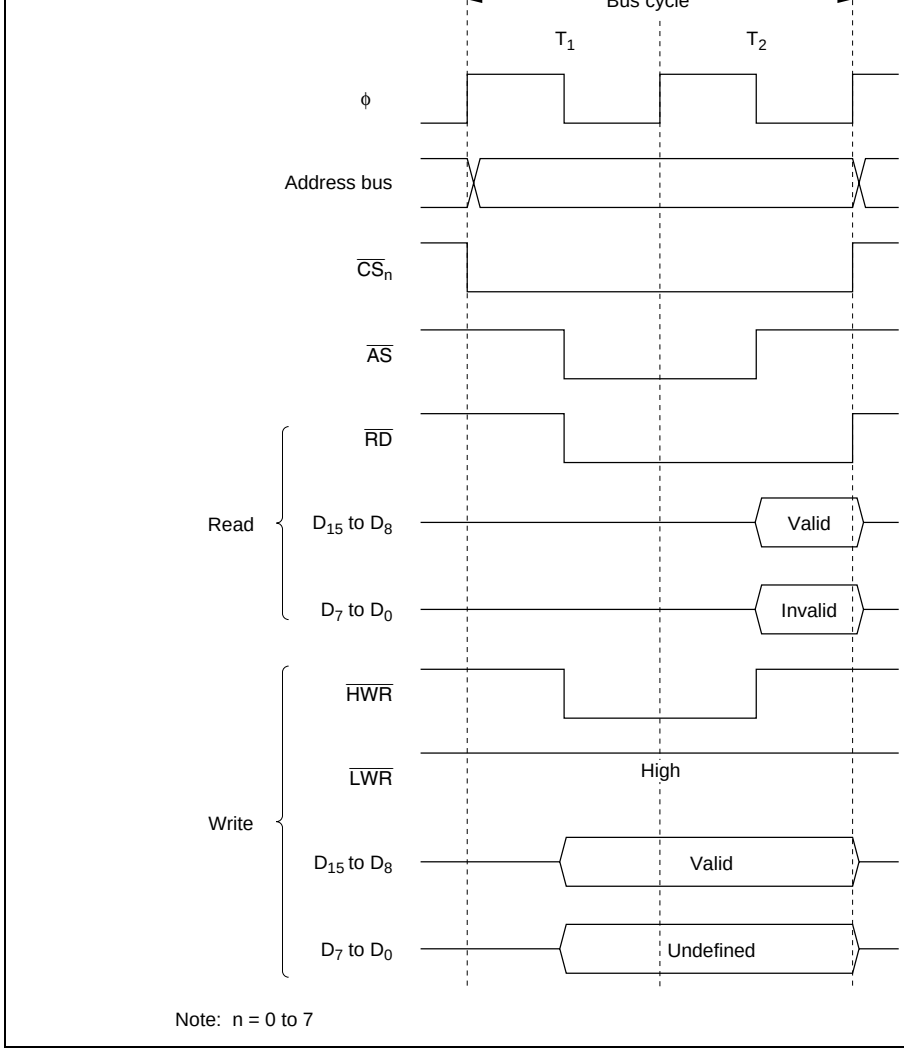


Figure 6.8 Bus Timing for 16-Bit 2-State Access Space (1) (Even Address Byt

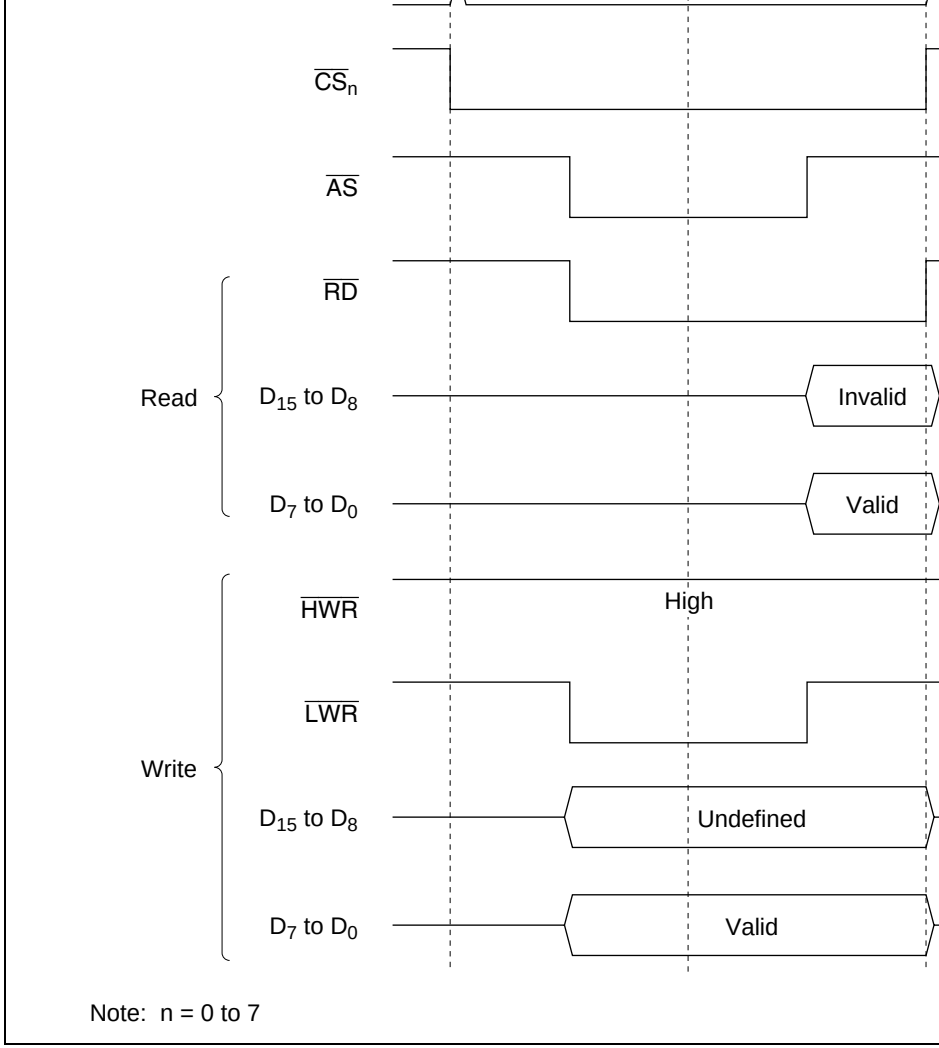
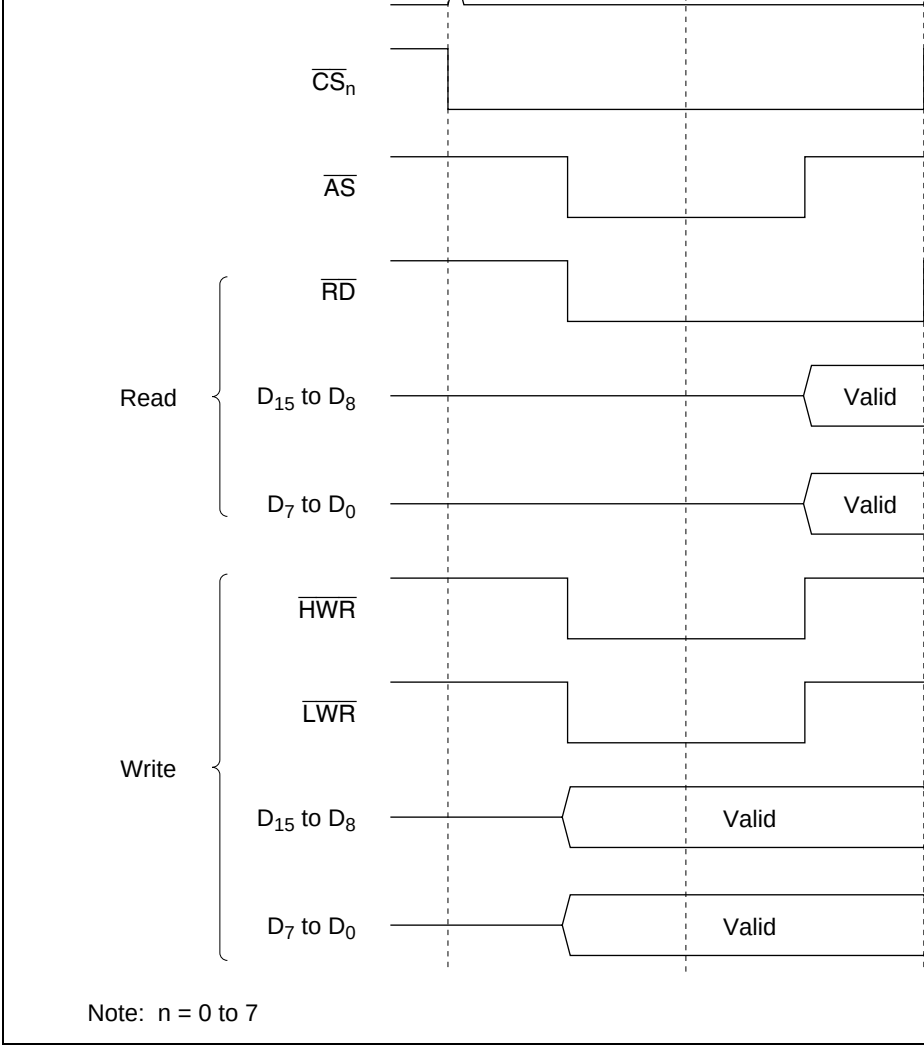


Figure 6.9 Bus Timing for 16-Bit 2-State Access Space (2) (Odd Address Byte



Note: n = 0 to 7

Figure 6.10 Bus Timing for 16-Bit 2-State Access Space (3) (Word Acc

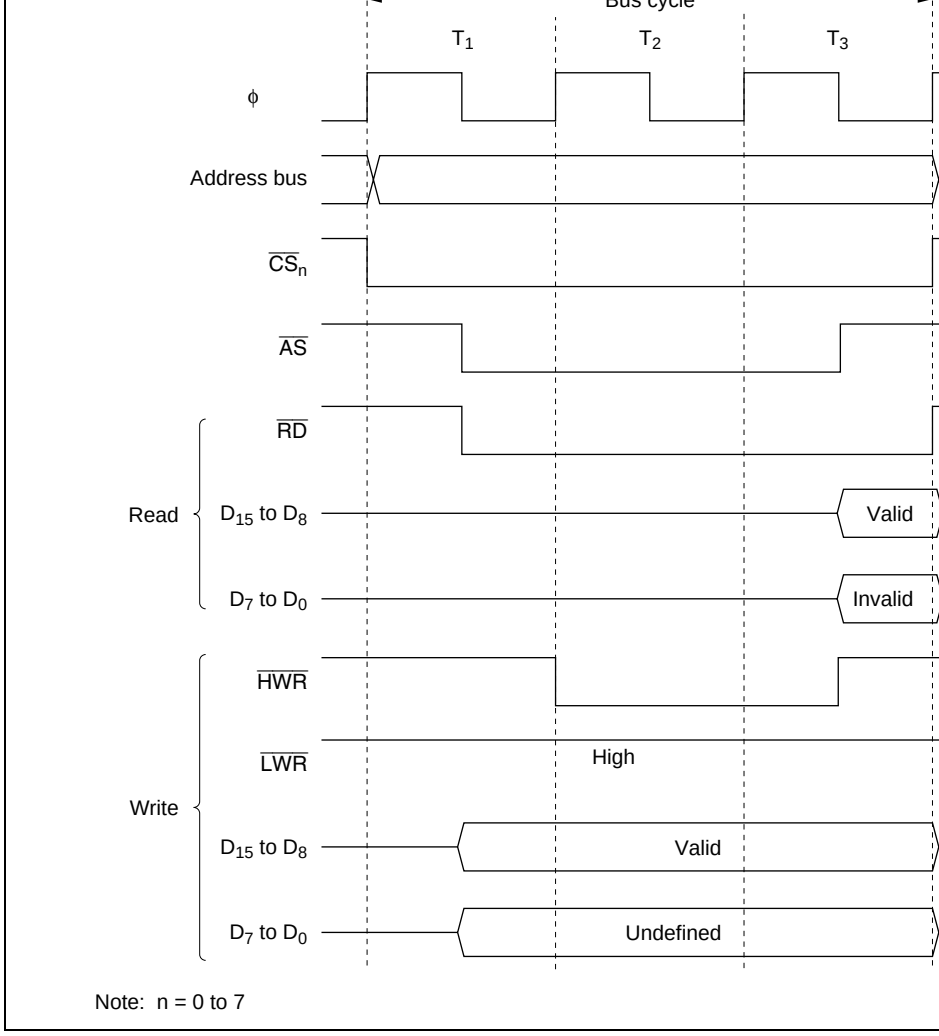


Figure 6.11 Bus Timing for 16-Bit 3-State Access Space (1) (Even Address Byt

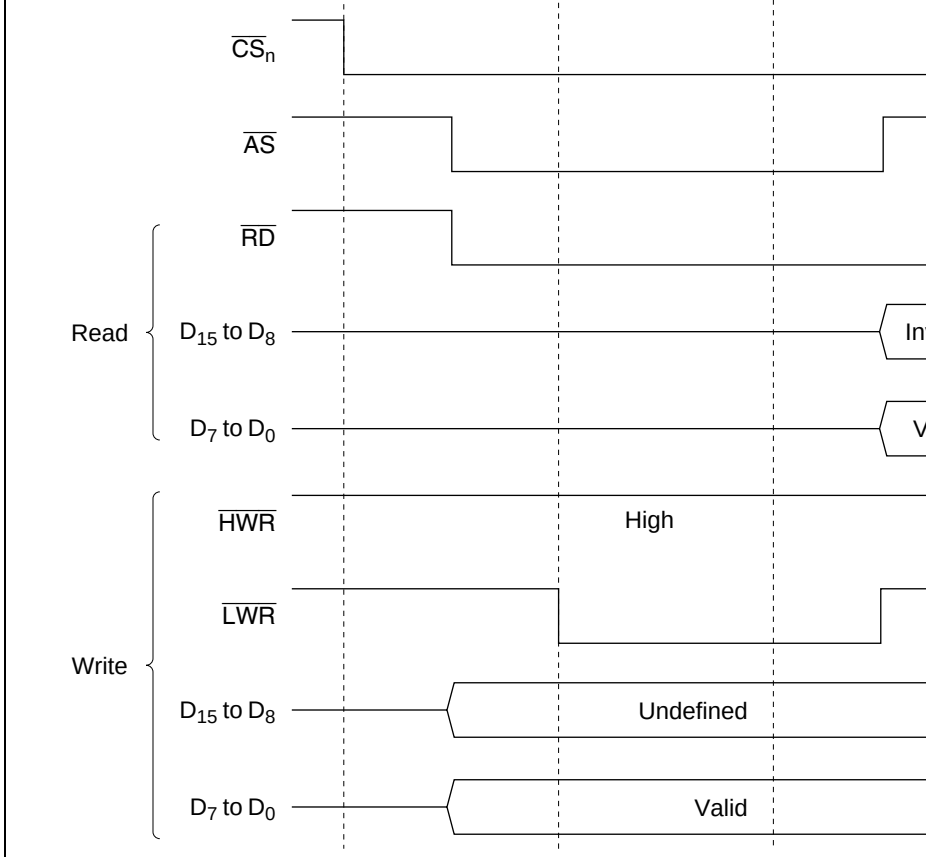


Figure 6.12 Bus Timing for 16-Bit 3-State Access Space (2) (Odd Address Byt

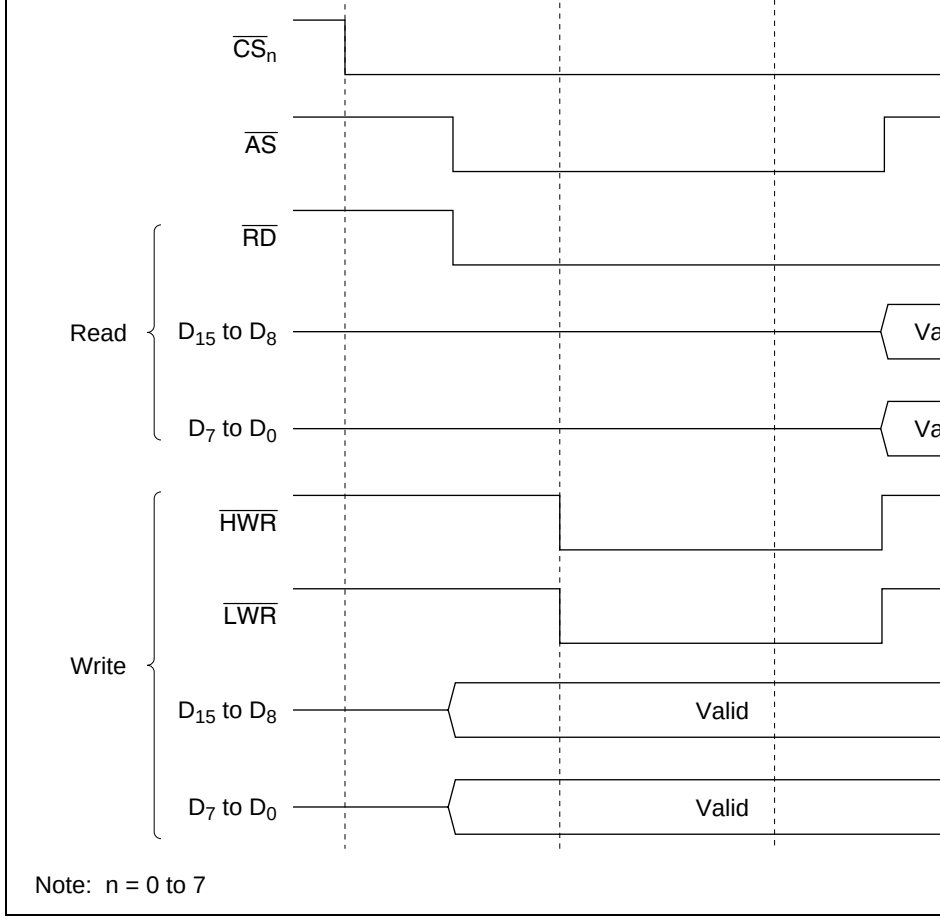


Figure 6.13 Bus Timing for 16-Bit 3-State Access Space (3) (Word Acces

individual area basis in 3-state access space, according to the settings of BWCRH and BWCRL.

- Pin Wait Insertion

  Setting the WAITE bit in BCRH to 1 enables wait insertion by means of the $\overline{\text{WAIT}}$ pin. If an external space is accessed in this state, a program wait is first inserted. If the $\overline{\text{WAIT}}$ pin is low at the falling edge of ϕ in the last T_2 or T_w state, another T_w state is inserted. If the $\overline{\text{WAIT}}$ pin is held low, T_w states are inserted until it goes high.

  This is useful when inserting four or more T_w states, or when changing the number of T_w states for different external devices.

  The WAITE bit setting applies to all areas.

Figure 6.14 shows an example of wait state insertion timing.

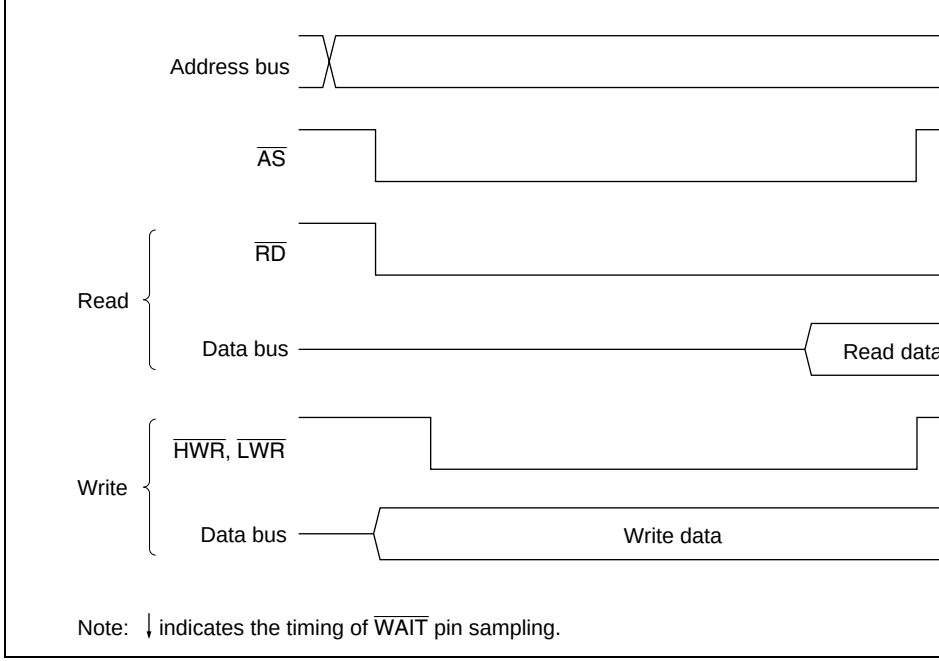


Figure 6.14 Example of Wait State Insertion Timing

The settings after a power-on reset are: 3-state access, 3 program wait state insertion, a input disabled. When a manual reset is performed, the contents of bus controller registe retained, and the wait control settings remain the same as before the reset.

possible, using fast page mode.

6.5.2 Setting DRAM Space

Areas 2 to 5 are designated as DRAM space by setting bits RMTS2 to RMTS0 in BCI relation between the settings of bits RMTS2 to RMTS0 and DRAM space is shown in Possible DRAM space settings are: one area (area 2), two areas (areas 2 and 3), and fo (areas 2 to 5).

Table 6.5 Settings of Bits RMTS2 to RMTS0 and Corresponding DRAM Spac

RMTS2	RMTS1	RMTS0	Area 5	Area 4	Area 3	
0	0	1	Normal space			D
	1	0	Normal space		DRAM sp	
		1	DRAM space			

	MXC1	MXC0	Shift Size	A_{23} to A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A
Row address	0	0	8 bits	A_{23} to A_{13}	A_{20}	A_{19}	A_{18}	A_{17}	A_{16}	A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A
		1	9 bits	A_{23} to A_{13}	A_{12}	A_{20}	A_{19}	A_{18}	A_{17}	A_{16}	A_{15}	A_{14}	A_{13}	A_{12}	A
	1	0	10 bits	A_{23} to A_{13}	A_{12}	A_{11}	A_{20}	A_{19}	A_{18}	A_{17}	A_{16}	A_{15}	A_{14}	A_{13}	A
		1	Setting prohibited	—	—	—	—	—	—	—	—	—	—	—	–
Column address	—	—	—	A_{23} to A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A

6.5.4 Data Bus

If the bit in ABWCR corresponding to an area designated as DRAM space is set to 1, th
designated as 8-bit DRAM space; if the bit is cleared to 0, the area is designated as 16-b
space. In 16-bit DRAM space, ×16-bit configuration DRAM can be connected directly.

In 8-bit DRAM space the upper half of the data bus, D_{15} to D_8, is enabled, while in 16-b
space both the upper and lower halves of the data bus, D_{15} to D_0, are enabled.

Access sizes and data alignment are the same as for the basic bus interface: see section
Size and Data Alignment.

		enable		enable for DRAM spac When 2-WE system is write enable for DRAM access.
$\overline{\text{LWR}}$	$\overline{\text{LCAS}}$/$\overline{\text{LWE}}$	Lower column address strobe/lower write enable	Output	When 2-CAS system (L is set, lower column ad for DRAM space acces When 2-WE system is write enable for DRAM access.
$\overline{\text{LCAS}}$	$\overline{\text{LCAS}}$	Lower column address strobe	Output	Lower column address access to 2-CAS type DRAM space.
$\overline{\text{CS}}_2$	$\overline{\text{RAS}}_2$	Row address strobe 2	Output	Row address strobe wh designated as DRAM s
$\overline{\text{CS}}_3$	$\overline{\text{RAS}}_3$	Row address strobe 3	Output	Row address strobe wh designated as DRAM s
$\overline{\text{CS}}_4$	$\overline{\text{RAS}}_4$	Row address strobe 4	Output	Row address strobe wh designated as DRAM s
$\overline{\text{CS}}_5$	$\overline{\text{RAS}}_5$	Row address strobe 5	Output	Row address strobe wh designated as DRAM s
$\overline{\text{CAS}}$	$\overline{\text{CAS}}$/$\overline{\text{UCAS}}$	Column address strobe/ upper column address strobe	Output	When 2-WE system is address strobe. When 2-CAS system is column address strobe
$\overline{\text{WAIT}}$	$\overline{\text{WAIT}}$	Wait	Input	Wait request signal
A_{12} to A_0	A_{12} to A_0	Address pins	Output	Row address/column a multiplexed output
D_{15} to D_0	D_{15} to D_0	Data pins	I/O	Data input/output pins

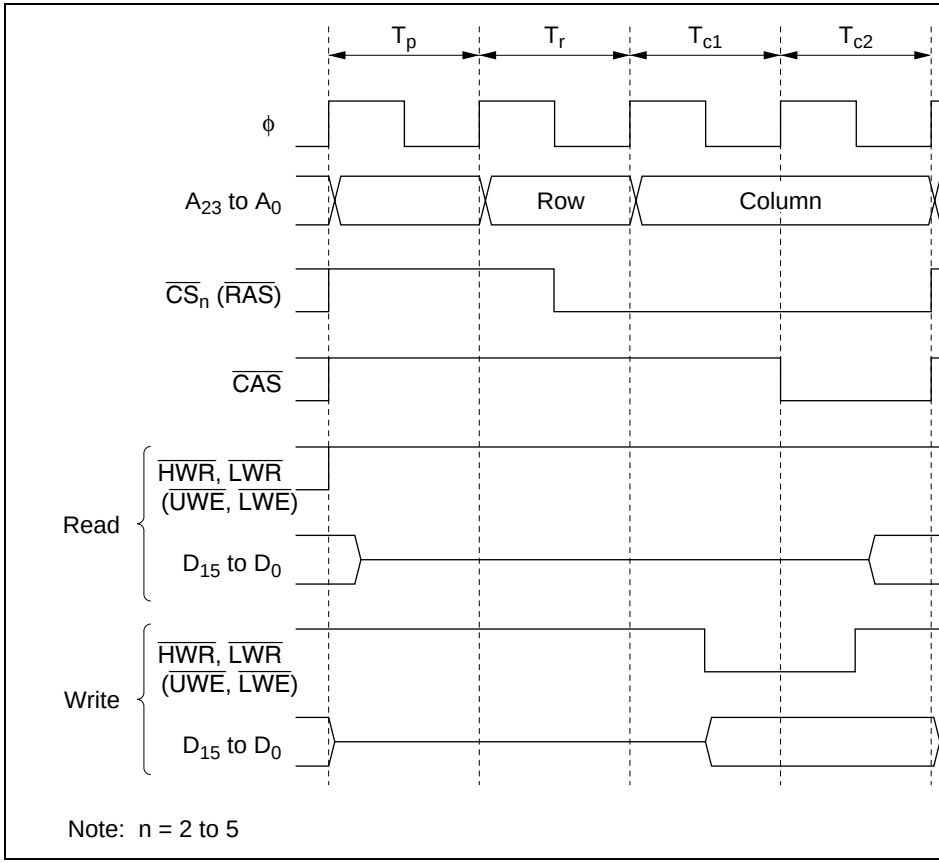


Figure 6.15 Basic Access Timing (2-WE System)

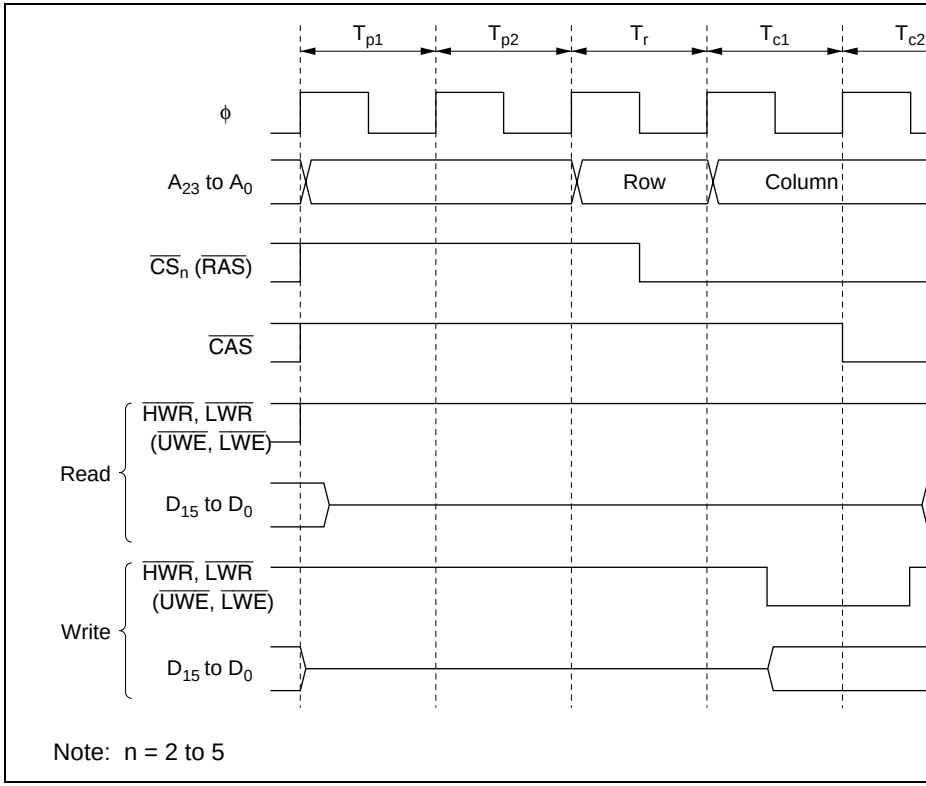


Figure 6.16 Timing with Two Precharge States (2-WE System)

the settings of WCRH and WCRL.

- Pin Wait Insertion

  When the WAITE bit in BCRH is set to 1, wait input by means of the $\overline{\text{WAIT}}$ pin is regardless of the setting of the AST bit in ASTCR. When DRAM space is accessed state, a program wait is first inserted. If the $\overline{\text{WAIT}}$ pin is low at the falling edge of ϕ T_{c1} or T_w state, another T_w state is inserted. If the $\overline{\text{WAIT}}$ pin is held low, T_w states are until it goes high.

Figure 6.17 shows an example of wait state insertion timing.

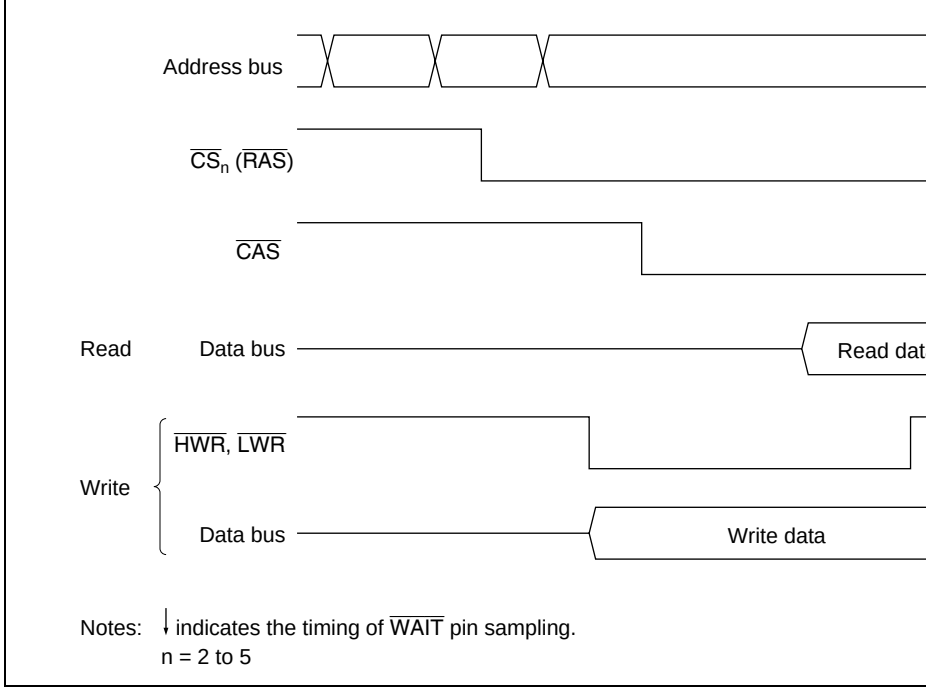


Figure 6.17 Example of Wait State Insertion Timing

UCAS, LCAS, and WE signals are output. Use of the LWR pin or the LCAS pin fo
signal can be selected by means of the LCASS bit.

(a) When LCASS = 0

Figure 6.18 (a) shows the control timing in the 2-CAS system (LCASS = 0), and
6.19 (a) shows an example of 2-CAS system (LCASS = 0) DRAM connection.

(b) When LCASS = 1

Figure 6.18 (b) shows the control timing in the 2-CAS system (LCASS = 1), and
6.19 (b) shows an example of 2-CAS system (LCASS = 1) DRAM connection.

In this case, since the $\overline{\text{LWR}}$ pin is used for the $\overline{\text{LCAS}}$ signal, RAS down mode c
used.

Regardless of the ICIS1 and ICIS0 bits, when non-DRAM space is accessed foll
DRAM space access, an idle cycle (TDI) is inserted after the DRAM space acce

Access to another space is not performed during CBR refreshing; access to anot
performed after insertion of an idle cycle (TRI).

- 2-WE System

When the CW2 bit in MCR is set to 1, the 2-WE system is selected. With this syste
$\overline{\text{UWE}}$, and $\overline{\text{LWE}}$ signals are output. Figure 6.20 shows the control timing in the 2-W
and figure 6.21 shows an example of DRAM connection using this system.

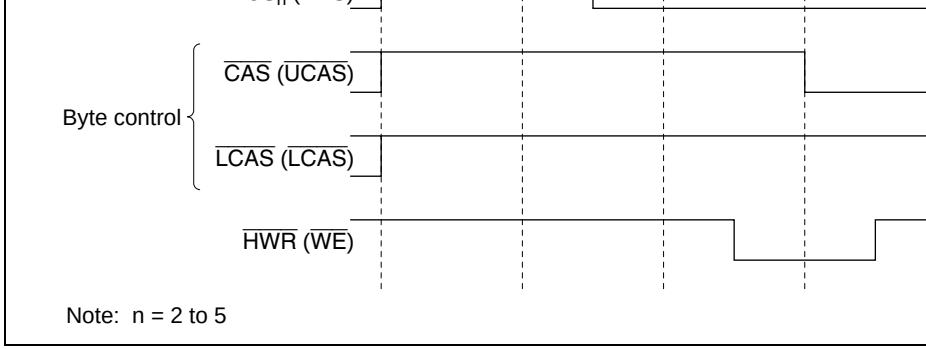


Figure 6.18 (a) 2-CAS System (LCASS = 0) Control Timing (Upper Byte Wri

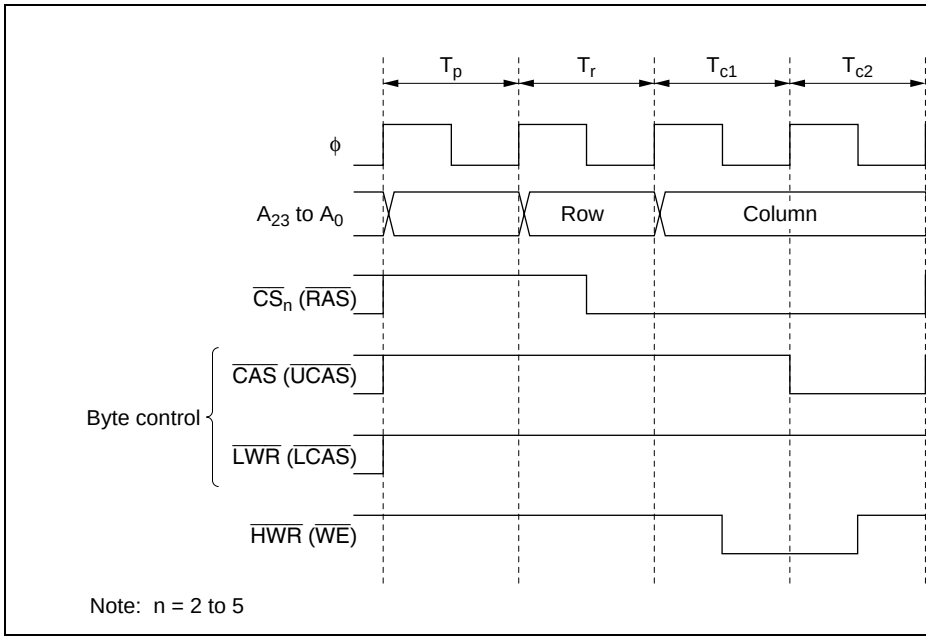


Figure 6.18 (b) 2-CAS System (LCASS = 1) Control Timing (Upper Byte Wri

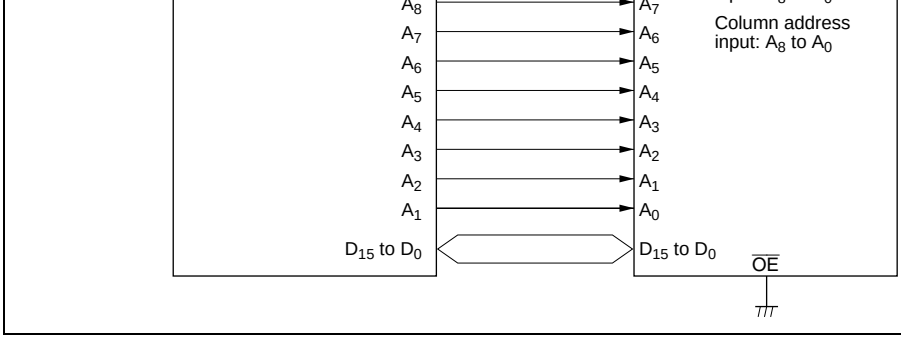


Figure 6.19 (a) Example of 2-CAS System (LCASS = 0) DRAM Connecti

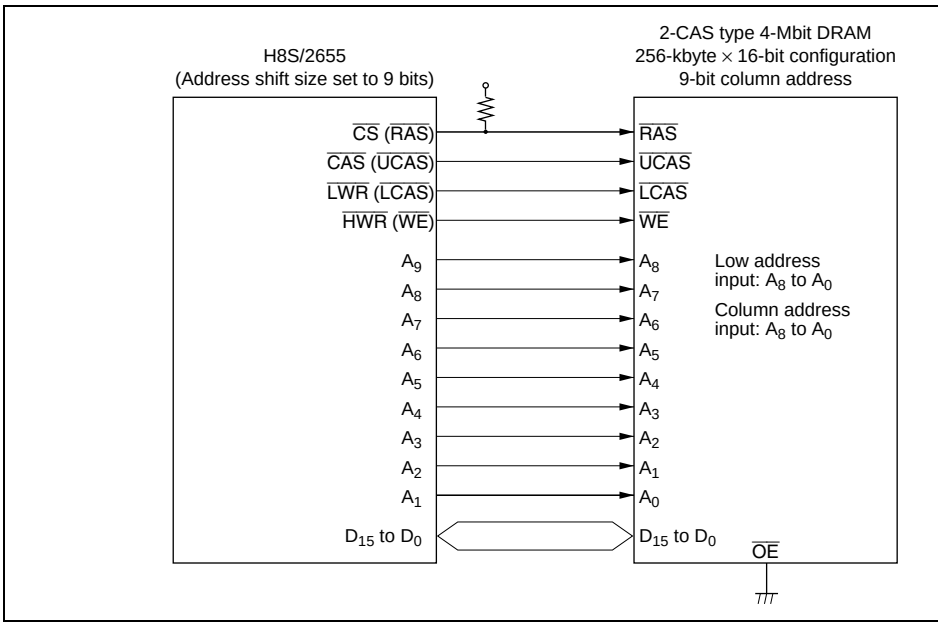


Figure 6.19 (b) Example of 2-CAS System (LCASS = 1) DRAM Connecti

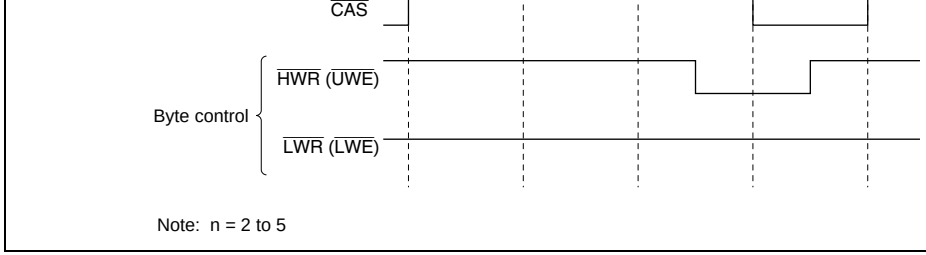


Figure 6.20 2-WE System Control Timing (Upper Byte Access)

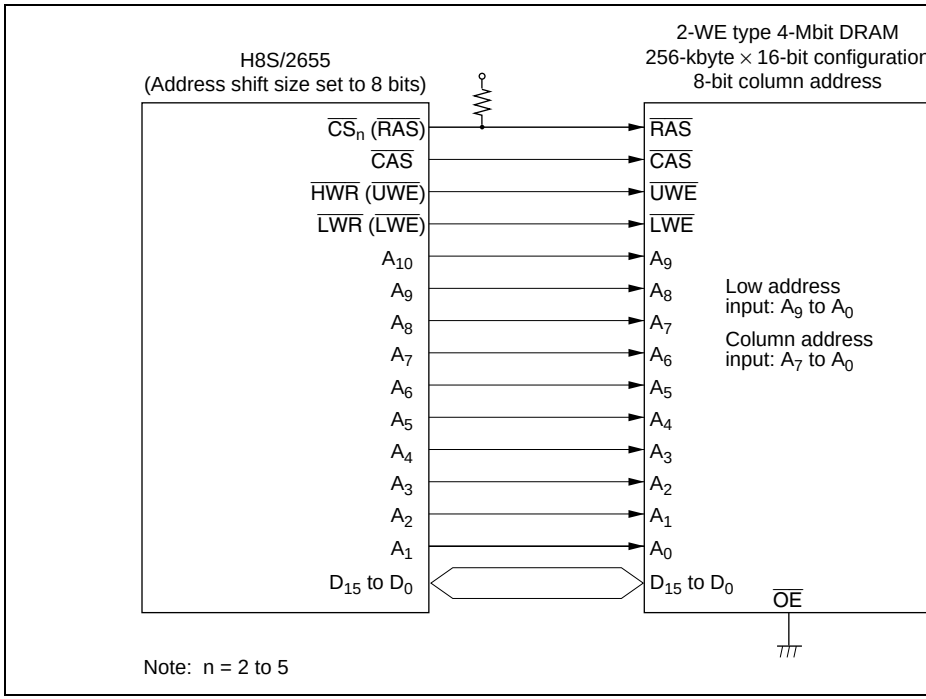


Figure 6.21 Example of 2-WE Type DRAM Connection

(1) Burst Access (Fast Page Mode) Operation Timing

Figure 6.22 shows the operation timing for burst access. When there are consecutive ac
for DRAM space, the $\overline{CAS}$ signal and column address output cycles (two states) contin
as the row address is the same for consecutive access cycles. The row address used for
comparison is set with bits MXC1 and MXC0 in MCR.

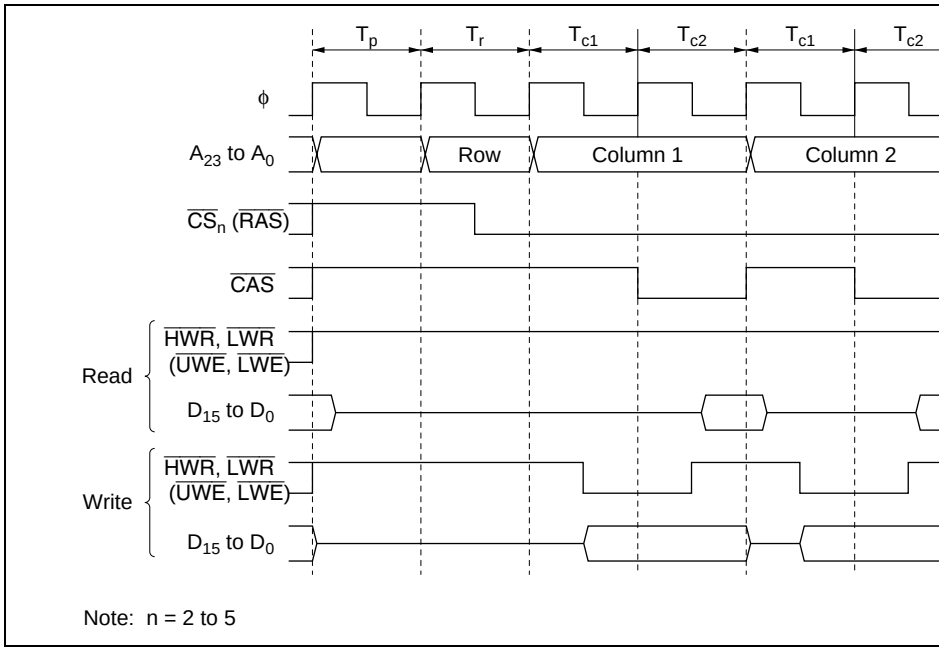


Figure 6.22 Operation Timing in Fast Page Mode (2-WE System)

The bus cycle can also be extended in burst access by inserting wait states. The wait sta
method and timing are the same as for full access. For details, see section 6.5.8, Wait C

RAS down mode: To select RAS down mode, set the RCDM bit in MCR to 1. If acce
DRAM space is interrupted and another space is accessed, the $\overline{RAS}$ signal is held low
access to the other space, and burst access is performed if the row address of the next
space access is the same as the row address of the previous DRAM space access. Figu
shows an example of the timing in RAS down mode.

Note, however, that the $\overline{RAS}$ signal will go high if a refresh operation interrupts RAS

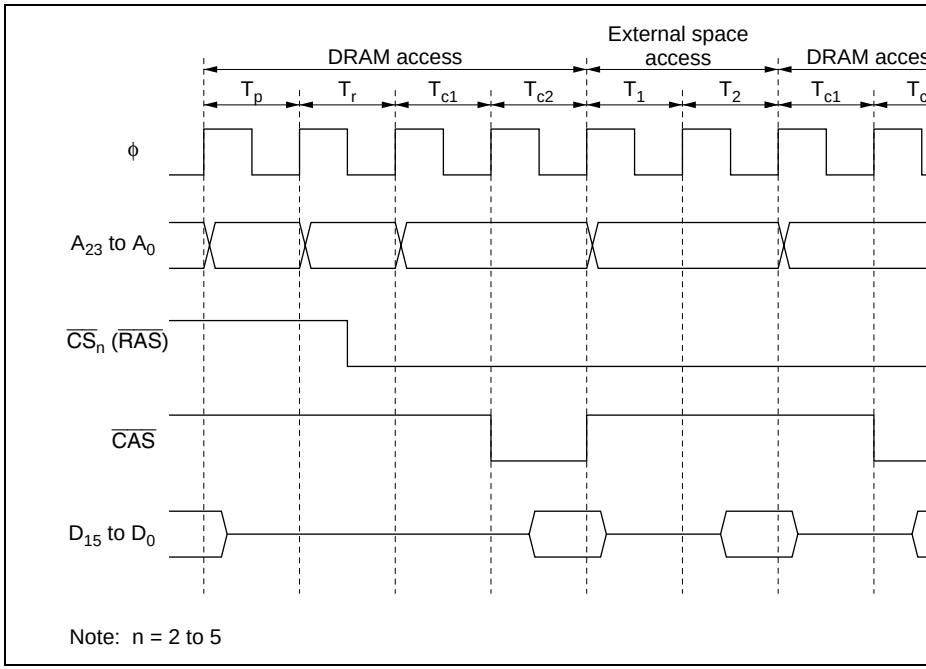


Figure 6.23 Example of Operation Timing in RAS Down Mode

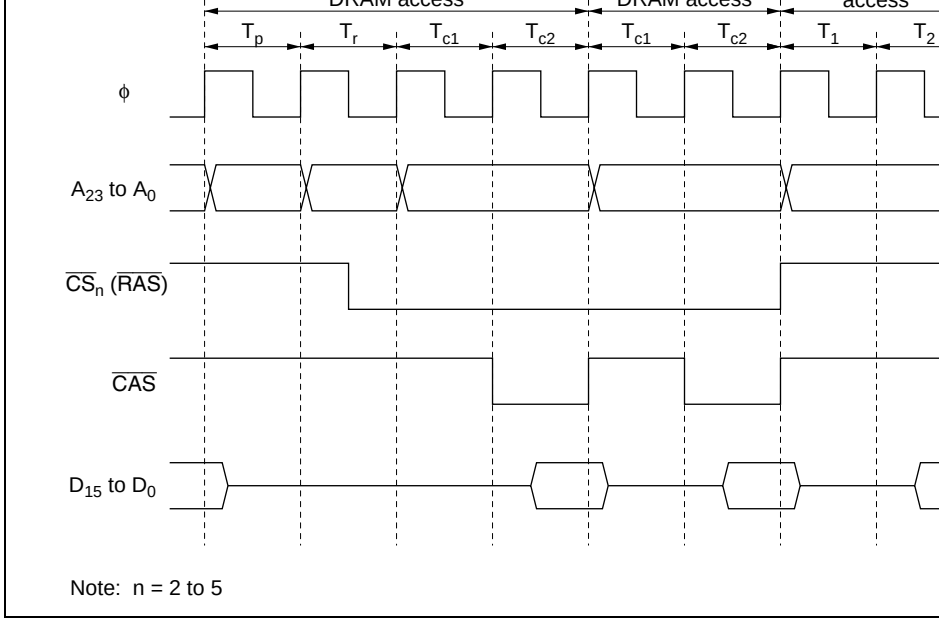


Figure 6.24 Example of Operation Timing in RAS Up Mode

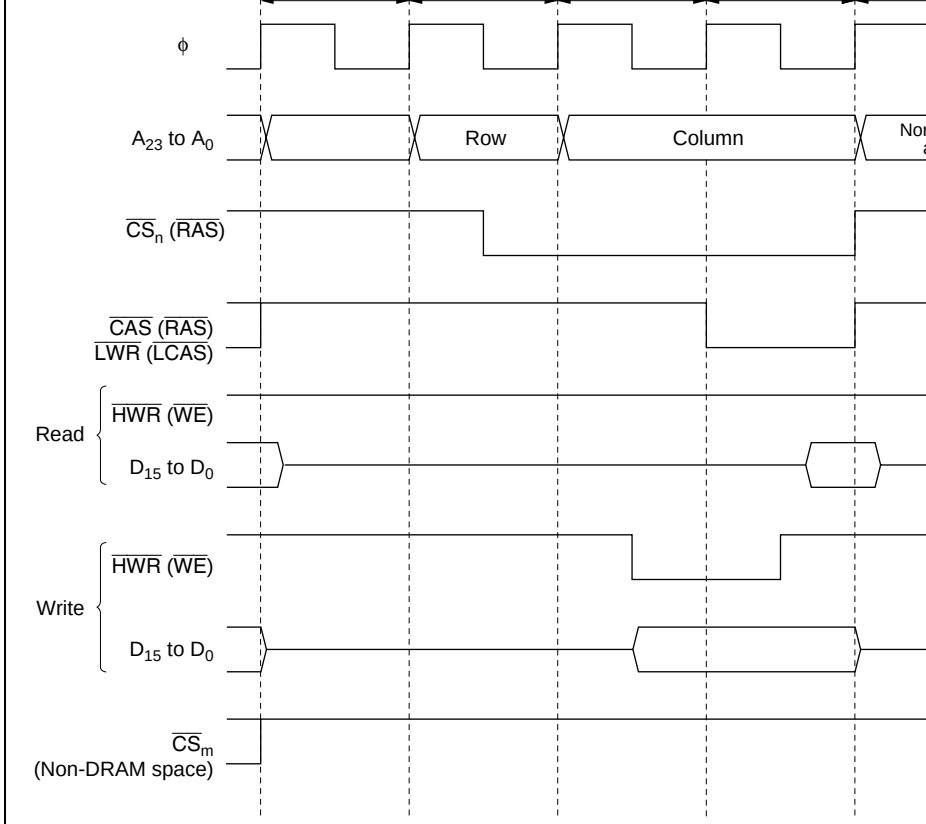


Figure 6.25 Idle Cycle Insertion with 2-CAS System (LCASS = 1)

With CBR refreshing, RTCNT counts up using the input clock selected by bits CKS2 to
DRAMCR, and when the count matches the value set in RTCOR (compare match), refr
is performed. At the same time, RTCNT is reset and starts counting again from H'00. R
thus repeated at fixed intervals determined by RTCOR and bits CKS2 to CKS0. Set a v
RTCOR and bits CKS2 to CKS0 that will meet the refreshing interval specification for
used.

When bits CKS2 to CKS0 are set, RTCNT starts counting up. RTCNT and RTCOR set
should therefore be completed before setting bits CKS2 to CKS0.

Do not clear the CMF flag when refresh control is being performed (RFSHE = 1).

RTCNT operation is shown in figure 6.26, compare match timing in figure 6.27, and 2-
CAS system CBR refresh timings in figures 6.28 and 29.

An access to another normal space is performed during the 2-WE system or 2-CAS sys
(LCASS = 0) refresh period. An access to another normal space is not performed during
2-CAS system (LCASS = 1) refresh period, but following insertion of an idle cycle (T_R
refreshing is completed. An idle cycle (T_{RI}) is not inserted when an on-chip memory ac
DRAM space access follows.

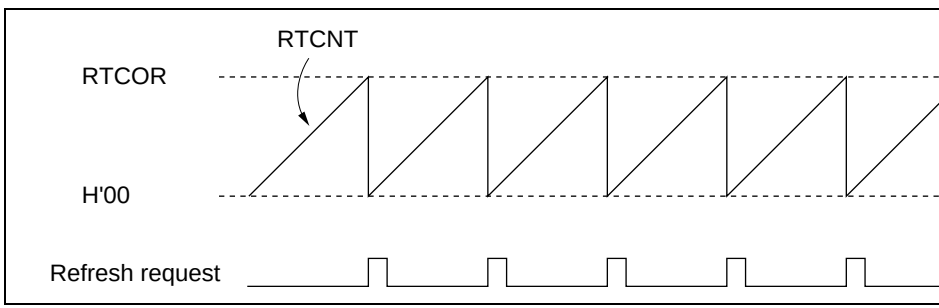


Figure 6.26 RTCNT Operation

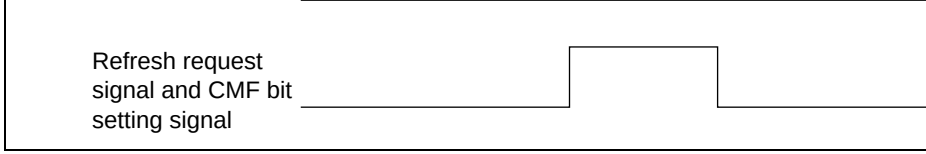


Figure 6.27 Compare Match Timing

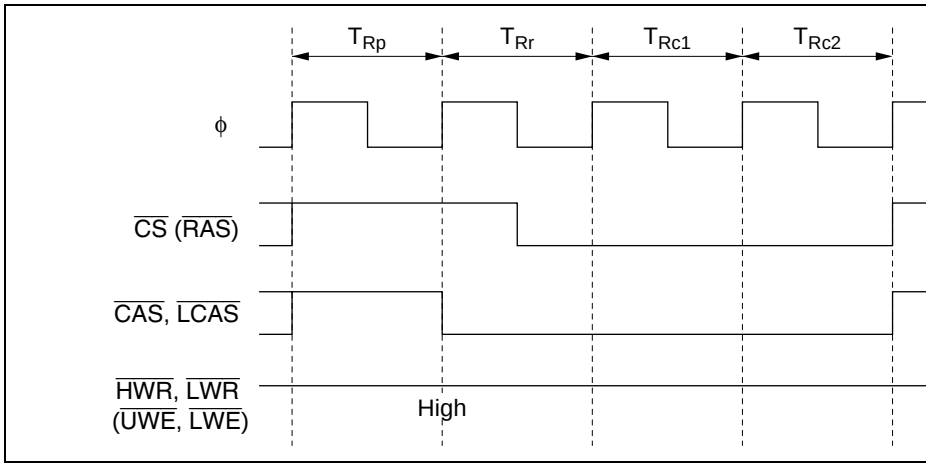


Figure 6.28 2-WE System or 2-CAS System (LCASS = 0) CBR Refresh T
(When RCW = 0 and CW2 = 1; or RCW = 0, CWZ = 0, and LCASS =

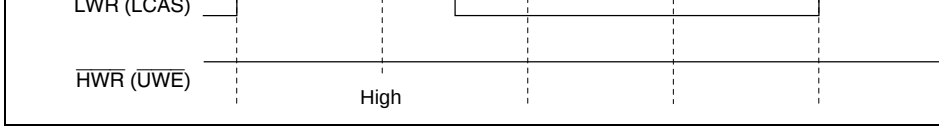


Figure 6.29 2-CAS System (LCASS = 1) CBR Refresh Timing (When RCW = 0 and CW2 = 0, LCASS = 1)

When the RCW bit is set to 1, $\overline{RAS}$ signal output is delayed by one cycle. The width of signal should be adjusted with bits RLW1 and RLW0. These bits are only enabled in re operations.

Figure 6.30 shows the timing when the RCW bit is set to 1.

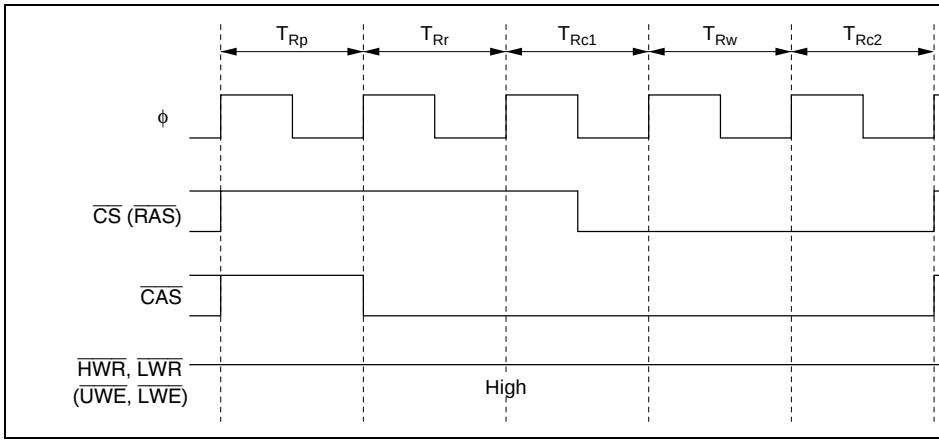


Figure 6.30 CBR Refresh Timing (When RCW = 1, RLW1 = 0, RLW0 = 1, C

(2) Self-Refreshing

A self-refresh mode (battery backup mode) is provided for DRAM as a kind of standby this mode, refresh timing and refresh addresses are generated within the DRAM.

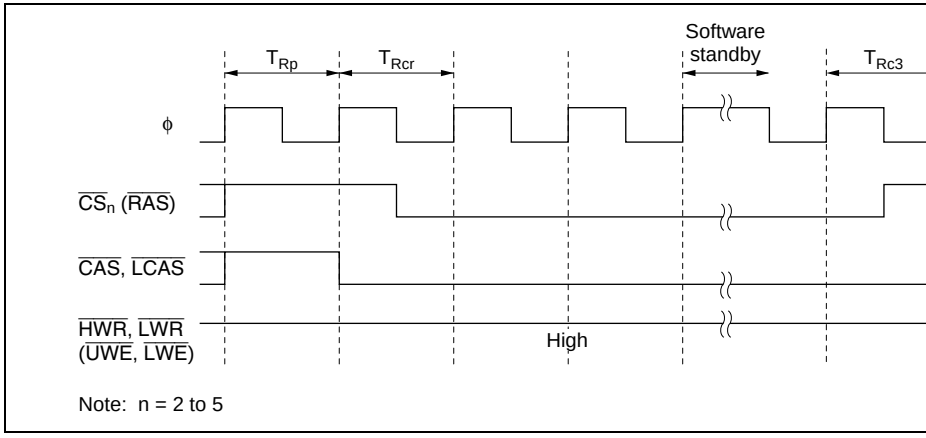


Figure 6.31 (a) Self-Refresh Timing (When CW2 = 1, or CWZ = 1 and LCA

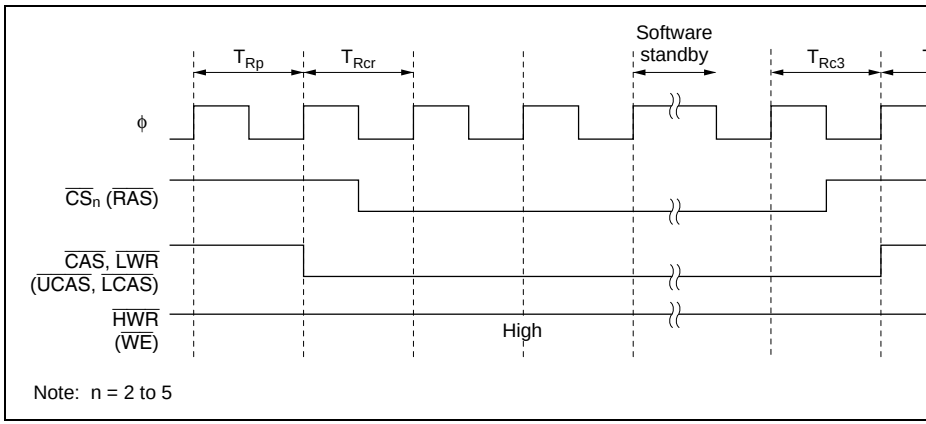


Figure 6.31 (b) Self-Refresh Timing (When CW2 = 0, LCASS = 1)

In the directly connected PSRAM, the refresh signal ($\overline{RFSH}$) and output enable signal multiplexed. Burst operation is also possible, using static column mode.

6.6.2 Setting PSRAM Space

Areas 2 to 5 are designated as PSRAM space by setting bits RMTS2 to RMTS0 in BC relation between the settings of bits RMTS2 to RMTS0 and PSRAM space is shown in Possible PSRAM space settings are: one area (area 2), two areas (areas 2 and 3), and fo (areas 2 to 5).

Table 6.8 Settings of Bits RMTS2 to RMTS0 and Corresponding PSRAM Spac

RMTS2	RMTS1	RMTS0	Area 5	Area 4	Area 3	
1	0	1	Normal space			PSF
	1	0	Normal space		PSRAM spa	
		1	PSRAM space			

6.6.3 Data Bus

If the bit in ABWCR corresponding to an area designated as PSRAM space is set to 1, designated as 8-bit PSRAM space; if the bit is cleared to 0, the area is designated as 16- PSRAM space.

In 8-bit PSRAM space the upper half of the data bus, D_{15} to D_8, is enabled, while in 16- space both the upper and lower halves of the data bus, D_{15} to D_0, are enabled.

Access sizes and data alignment are the same as for the basic bus interface: see section Size and Data Alignment.

		enable		PSRAM space is de 16-bit access, or wri when designated for 8-bit access.
$\overline{LWR}$	$\overline{LWE}$	Lower write enable	Output	Lower write enable v PSRAM space is de 16-bit access.
$\overline{CS_2}$	$\overline{CE_2}$	Chip enable 2	Output	Chip enable signal v is designated as PS
$\overline{CS_3}$	$\overline{CE_3}$	Chip enable 3	Output	Chip enable signal v is designated as PS
$\overline{CS_4}$	$\overline{CE_4}$	Chip enable 4	Output	Chip enable signal v is designated as PS
$\overline{CS_5}$	$\overline{CE_5}$	Chip enable 5	Output	Chip enable signal v is designated as PS
$\overline{CAS}$	$\overline{OE}/\overline{RFSH}$	Output enable/refresh	Output	Connected to PSRA enable/refresh dual-
$\overline{WAIT}$	$\overline{WAIT}$	Wait	Input	Wait request signal
A_{20} to A_0	A_{20} to A_0	Address pins	Output	Address output pins
D_{15} to D_0	D_{15} to D_0	Data pins	I/O	Data input/output pi

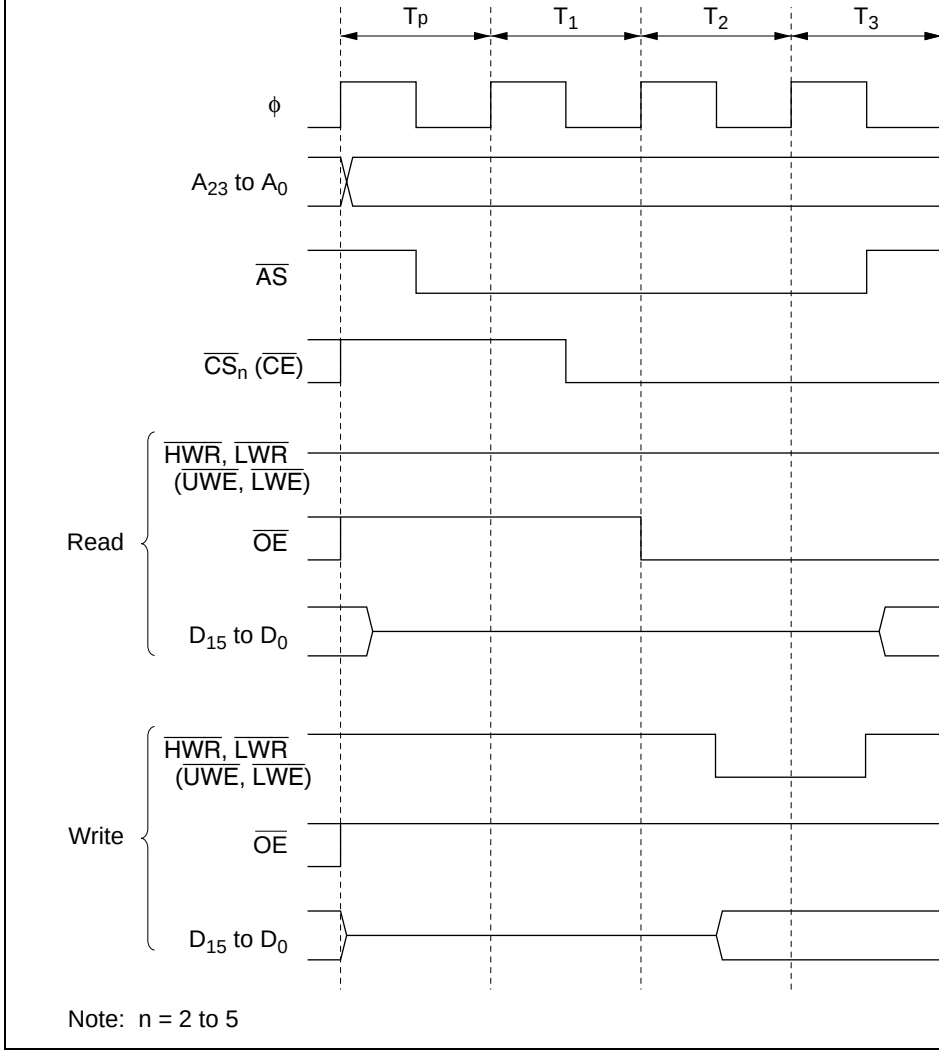


Figure 6.32 Basic Access Timing

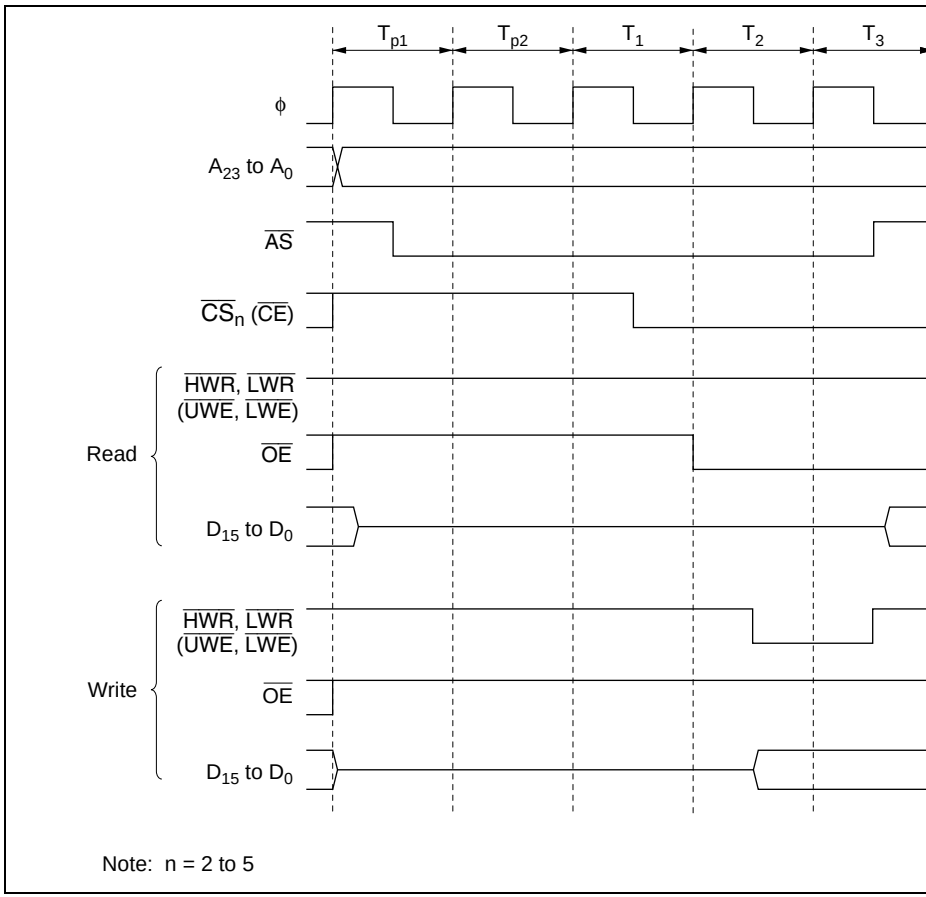


Figure 6.33 Timing with Two Precharge States

the settings of WCRH and WCRL.

(2) Pin Wait Insertion

When the WAITE bit in BCRL is set to 1, wait input by means of the $\overline{\text{WAIT}}$ pin is
regardless of the setting of the AST bit in ASTCR. When PSRAM space is accesse
state, a program wait is first inserted. If the $\overline{\text{WAIT}}$ pin is low at the falling edge of ¢
T_2 or T_w state, another T_w state is inserted. If the $\overline{\text{WAIT}}$ pin is held low, T_w states are
until it goes high.

Figure 6.34 shows an example of wait state insertion timing.

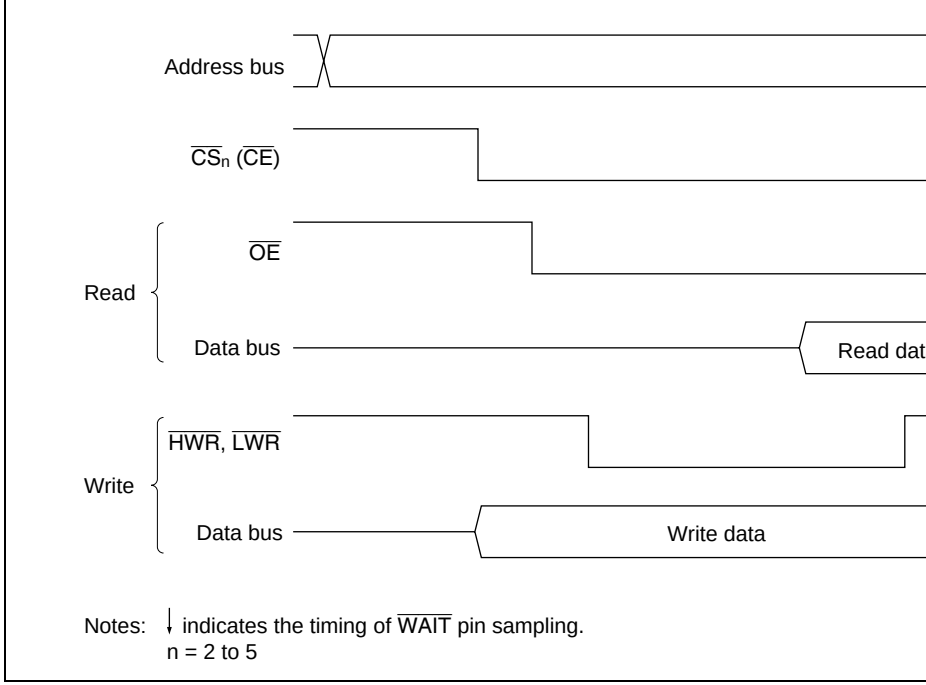


Figure 6.34 Example of Wait State Insertion Timing

for PSRAM space, column address output cycles (two states) continue as long as the ro
is the same for consecutive access cycles. The row address used for the comparison is s
MXC1 and MXC0 in MCR.

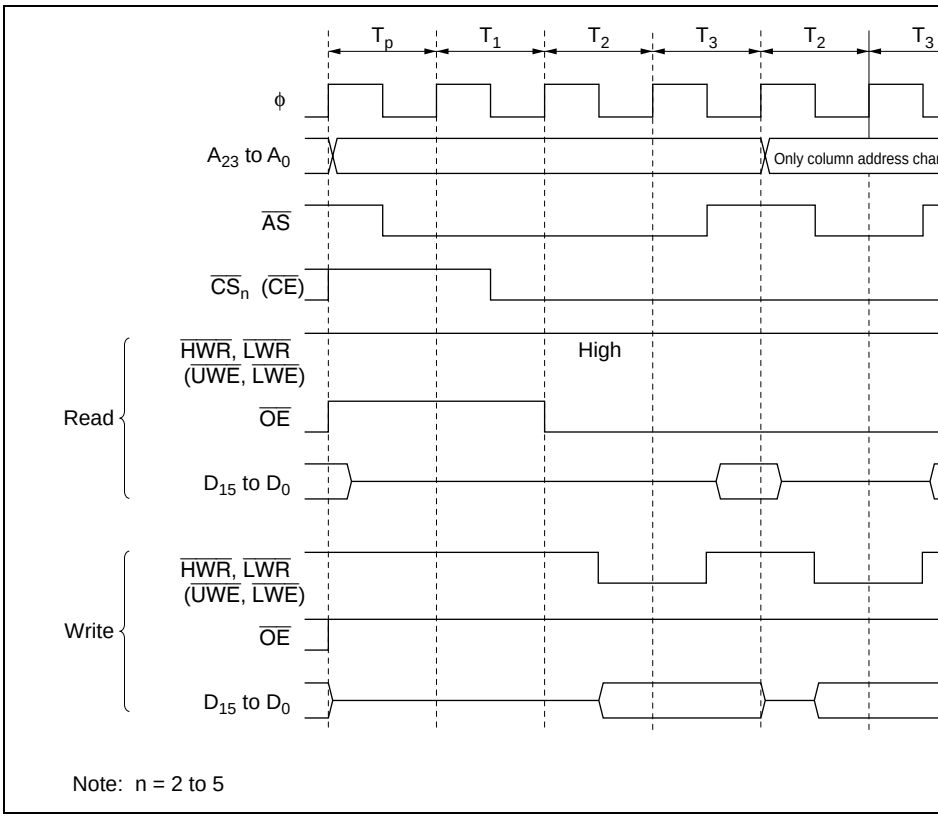


Figure 6.35 Operation Timing for Burst Access

The bus cycle can also be extended in burst access by inserting wait states. The wait sta
method and timing are the same as for normal mode. For details, see section 6.6.7, Wai

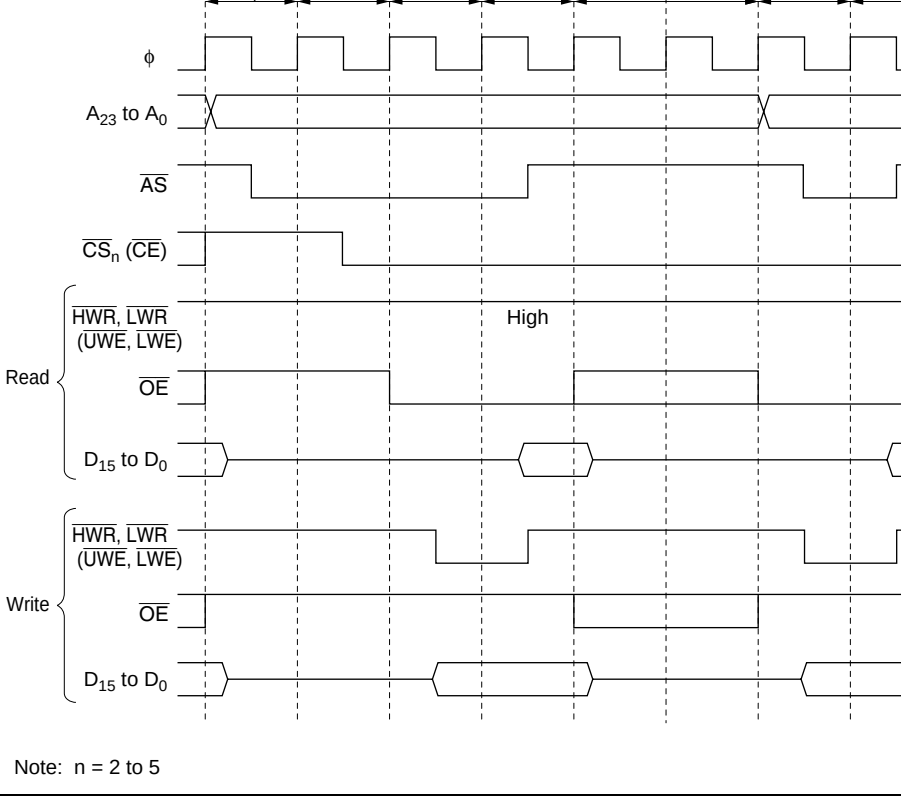


Figure 6.36 Example of Operation Timing in Burst Access

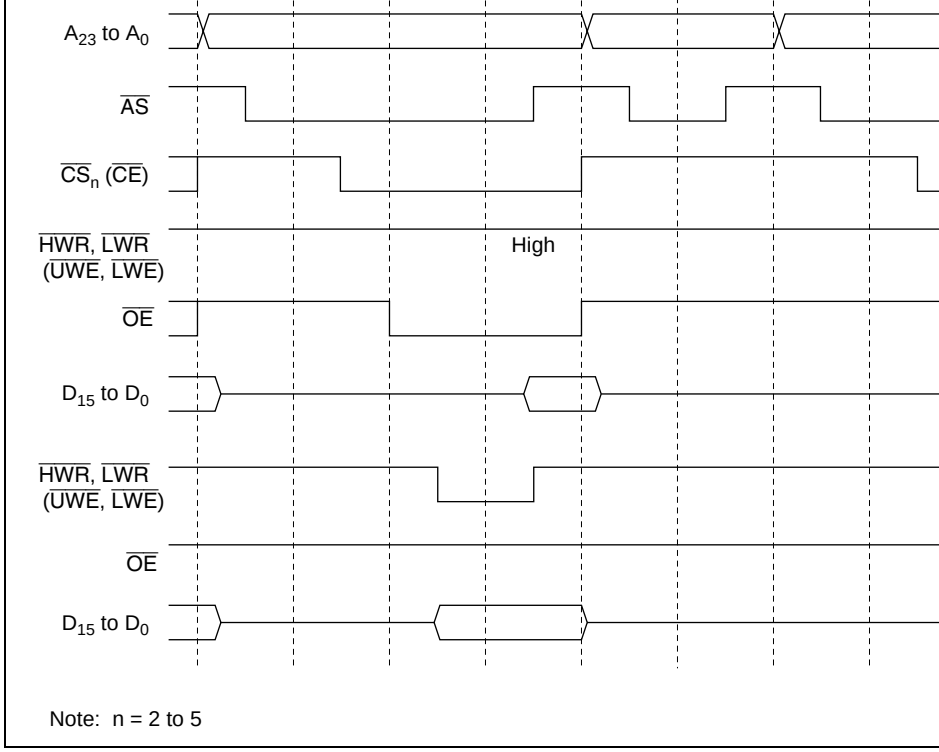


Figure 6.37 Example of Interrupted Operation Timing in Burst Access

With auto-refreshing, RTCNT counts up using the input clock selected by bits CKS2 t
DRAMCR, and when the count matches the value set in RTCOR (compare match), re
is performed. At the same time, RTCNT is reset and starts counting again from H'00.
thus repeated at fixed intervals determined by RTCOR and bits CKS2 to CKS0. Set a
RTCOR and bits CKS2 to CKS0 that will meet the refreshing interval specification fo
PSRAM used.

When bits CKS2 to CKS0 are set, RTCNT starts counting up. RTCNT and RTCOR se
should therefore be completed before setting bits CKS2 to CKS0.

Auto-refresh timing is shown in figure 6.38.

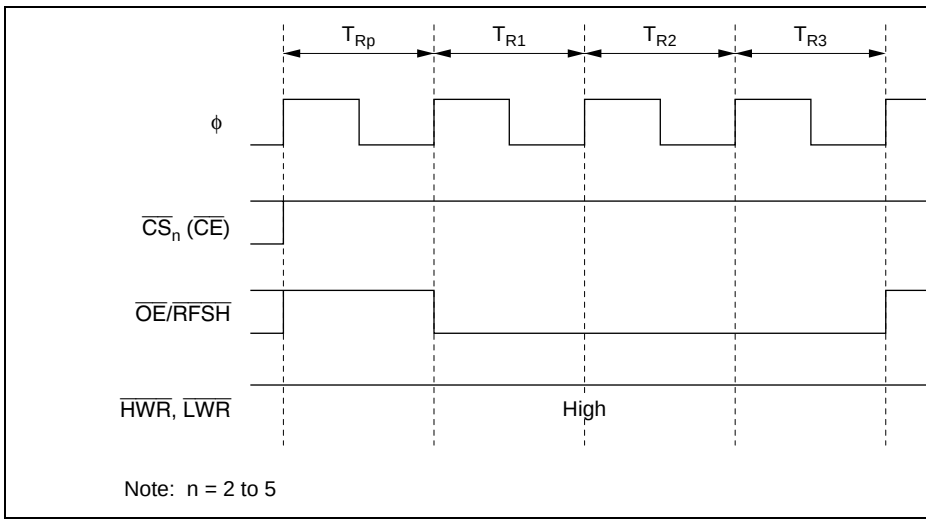


Figure 6.38 Auto-Refresh Timing

Figure 6.39 shows self-refresh timing.

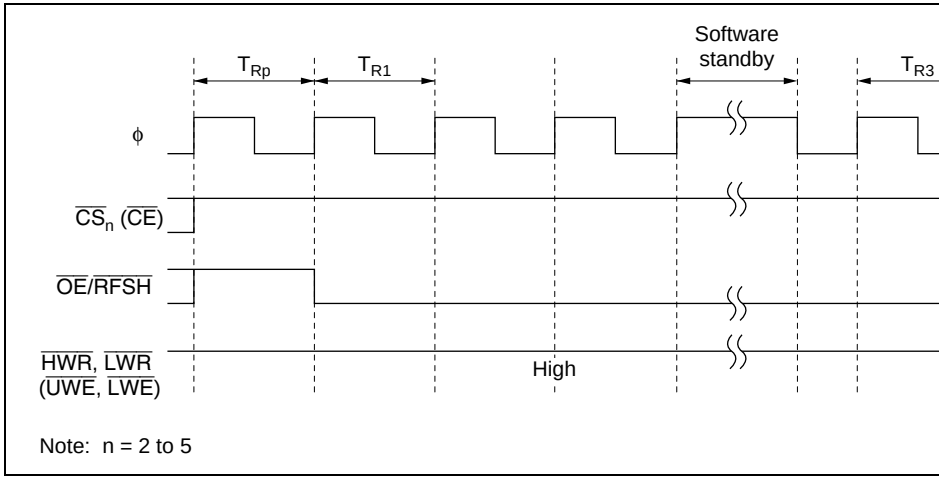


Figure 6.39 Self-Refresh Timing

6.6.10 Power-On Sequence

A power-on reset initializes the bus controller. If PSRAM is connected, you should che
characteristics and perform the necessary processing.

Burst access is performed by determining the address only, irrespective of the bus mas
$\overline{\text{DACK}}$ output goes low from the T_{c1} state in the case of the DRAM interface, and fron
in the case of the PSRAM interface.

Figure 6.40 shows the $\overline{\text{DACK}}$ output timing for the DRAM interface when DDS = 1.

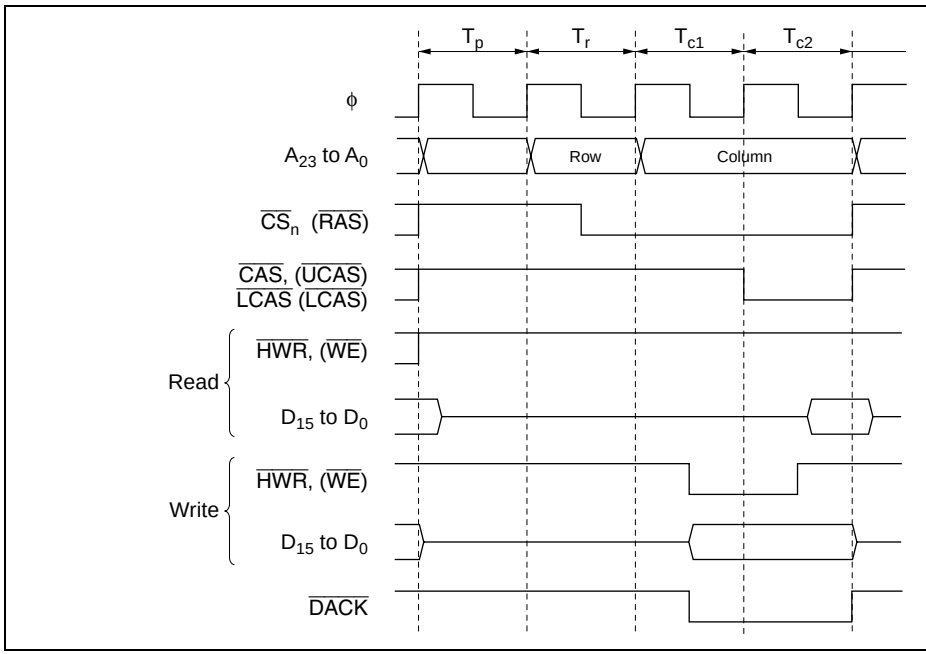


Figure 6.40 $\overline{\text{DACK}}$ Output Timing when DDS = 1 (Example of DRAM Ac

Figure 6.41 shows the $\overline{\text{DACK}}$ output timing for the DRAM interface when DDS = 0.

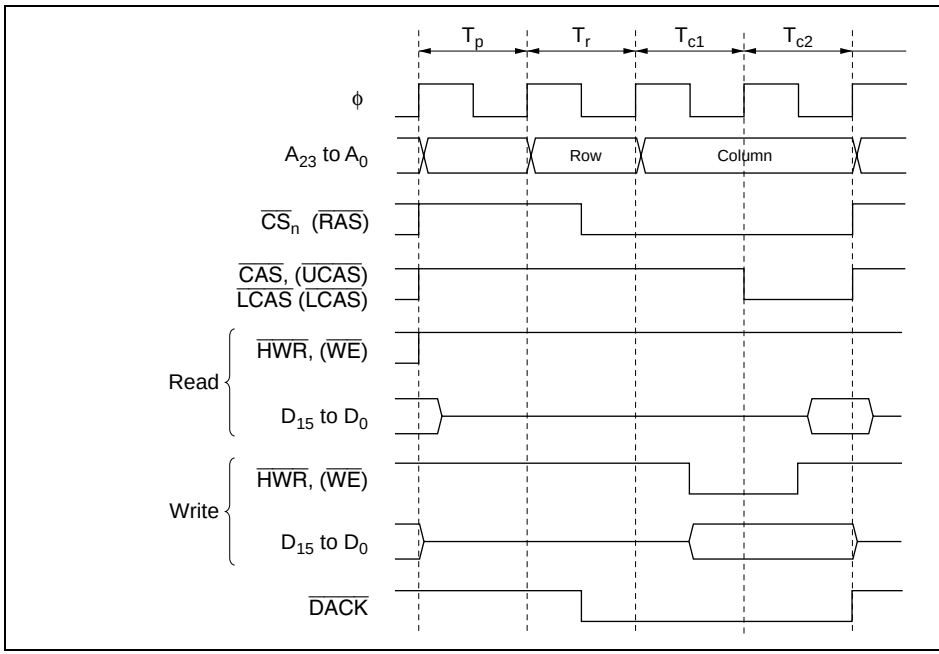


Figure 6.41 $\overline{\text{DACK}}$ Output Timing when DDS = 0 (Example of DRAM Acc

Area 0 can be designated as burst ROM space by means of the BRSTRM bit in BCRH. Consecutive burst accesses of a maximum of 4 words or 8 words can be performed for instruction fetches only. One or two states can be selected for burst access.

Do not select the burst ROM interface and pseudo-SRAM burst operation at the same

6.8.2 Basic Timing

The number of states in the initial cycle (full access) of the burst ROM interface is in with the setting of the AST0 bit in ASTCR. Also, when the AST0 bit is set to 1, wait s insertion is possible. One or two states can be selected for the burst cycle, according to of the BRSTS1 bit in BCRH. Wait states cannot be inserted. When area 0 is designate ROM space, it becomes 16-bit access space regardless of the setting of the ABW0 bit

When the BRSTS0 bit in BCRH is cleared to 0, burst access of up to 4 words is perfor the BRSTS0 bit is set to 1, burst access of up to 8 words is performed.

The basic access timing for burst ROM space is shown in figure 6.42 (a) and (b). The shown in figure 6.42 (a) is for the case where the AST0 and BRSTS1 bits are both set that in figure 6.42 (b) is for the case where both these bits are cleared to 0.

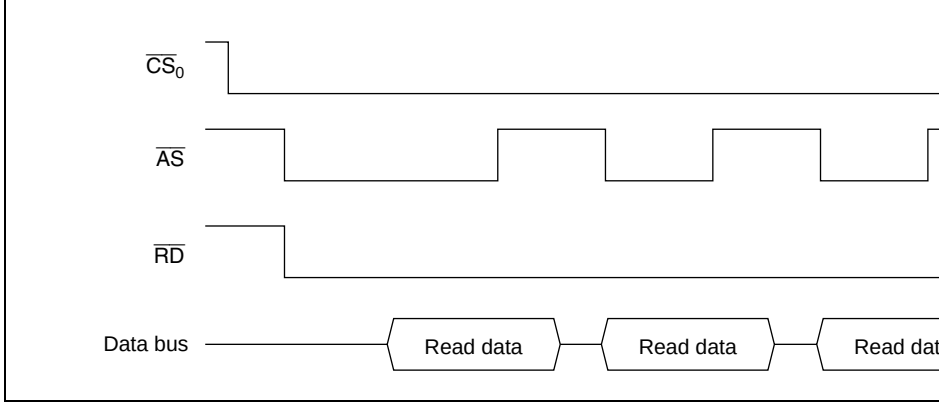


Figure 6.42 (a) Example of Burst ROM Access Timing (When AST0 = BRST

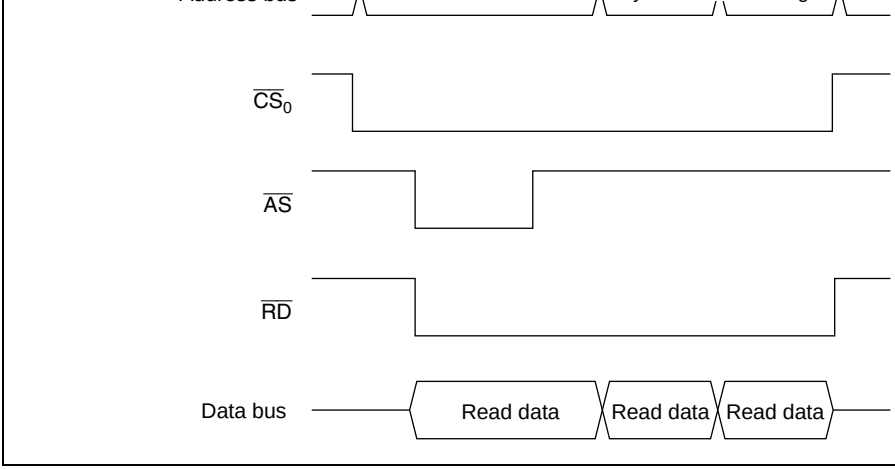


Figure 6.42 (b) Example of Burst ROM Access Timing (When AST0 = BRS

6.8.3 Wait Control

As with the basic bus interface, either program wait insertion or pin wait insertion usi
pin can be used in the initial cycle (full access) of the burst ROM interface. See sectio
Control.

Wait states cannot be inserted in a burst cycle.

floating time, and high-speed memory, I/O interfaces, and so on.

(1) Consecutive Reads between Different Areas

If consecutive reads between different areas occur while the ICIS1 bit in BCRH is set t
cycle is inserted at the start of the second read cycle. This is enabled in advanced mode

Figure 6.43 shows an example of the operation in this case. In this example, bus cycle A
cycle from ROM with a long output floating time, and bus cycle B is a read cycle from
each being located in a different area. In (a), an idle cycle is not inserted, and a collision
cycle B between the read data from ROM and that from SRAM. In (b), an idle cycle is
and a data collision is prevented.

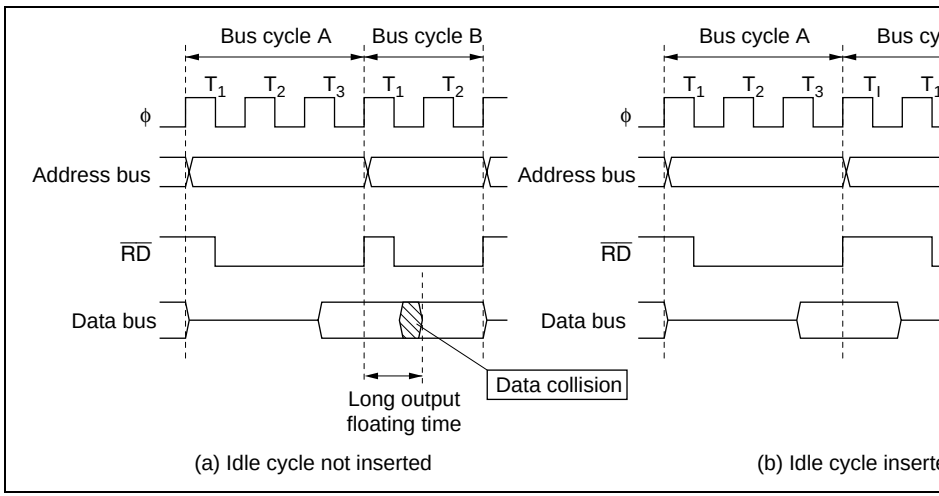


Figure 6.43 Example of Idle Cycle Operation (1) (When ICIS1 = 1)

the CPU write data. In (b), an idle cycle is inserted, and a data collision is prevented.

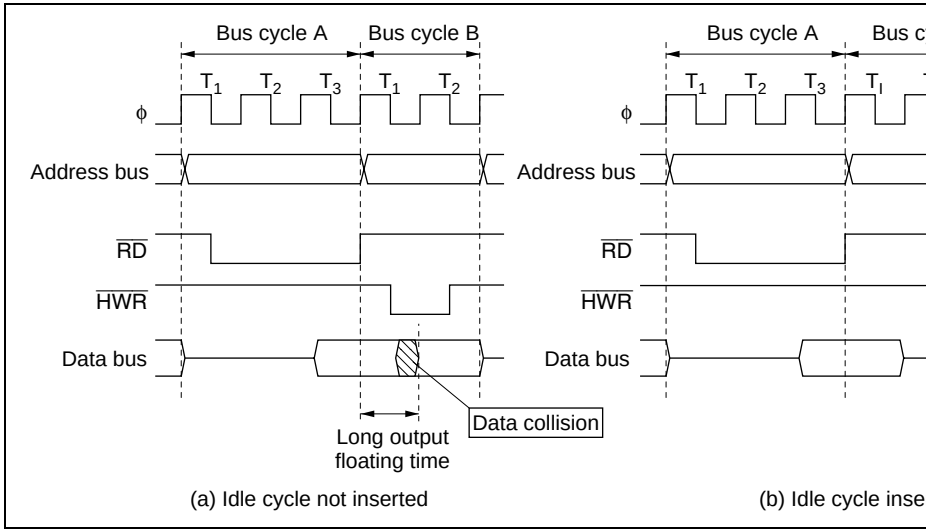


Figure 6.44 Example of Idle Cycle Operation (2) (When ICIS0 = 1)

(3) Usage Notes

When DRAM space is accessed, the ICIS0 and ICIS1 bit settings are disabled. In the c
consecutive reads between different areas, for example, if the second access is a DRA
only a T_p cycle is inserted, and a T_I cycle is not. The timing in this case is shown in fig

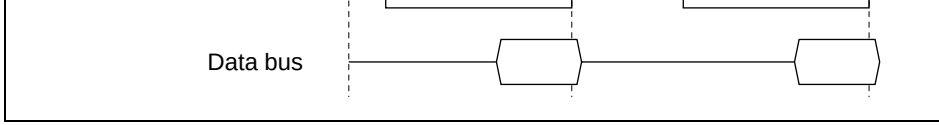


Figure 6.45 Example of DRAM Access after External Read

6.9.2 Pin States in Idle Cycle

Table 6.10 shows pin states in an idle cycle.

Table 6.10 Pin States in Idle Cycle

Pins	Pin State
A_{23} to A_0	Contents of next bus cycle
D_{15} to D_0	High impedance
$\overline{CS}_n$	High*1
$\overline{CAS}/\overline{OE}$	High*2
$\overline{AS}$	High
$\overline{RD}$	High
$\overline{HWR}$	High
$\overline{LWR}$	High
$\overline{DACK}_n$	High

Notes: 1. Remains low in PSRAM space CS down mode. Also remains low in DRAM s down mode or a refresh cycle.
2. Remains low in PSRAM space CS down mode or a refresh cycle.

longer, and there is an internal access next, only an external write is executed in the fi
from the next state onward an internal access (on-chip memory or internal I/O register
is executed in parallel with the external write rather than waiting until it ends.

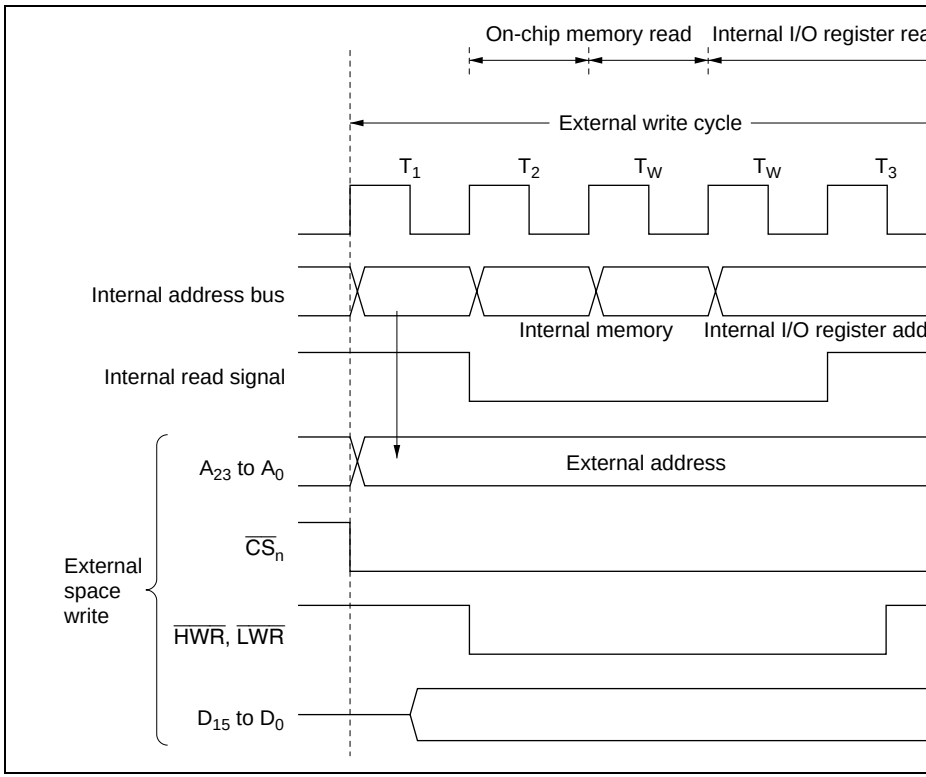


Figure 6.46 Example of Timing when Write Data Buffer Function is U

If an internal bus master wants to make an external access in the external bus released s
refresh request is generated, it can issue a bus request off-chip.

6.11.2 Operation

In external expansion mode, the bus can be released to an external device by setting the
in BCRL to 1. Driving the $\overline{\text{BREQ}}$ pin low issues an external bus request to the H8S/265
When the $\overline{\text{BREQ}}$ pin is sampled, at the prescribed timing the $\overline{\text{BACK}}$ pin is driven low,
address bus, data bus, and bus control signals are placed in the high-impedance state, es
the external bus-released state.

In the external bus released state, an internal bus master can perform accesses using the
bus. When an internal bus master wants to make an external access, it temporarily defe
activation of the bus cycle, and waits for the bus request from the external bus master to
dropped. Even if a refresh request is generated in the external bus released state, refres
deferred until the external bus master drops the bus request.

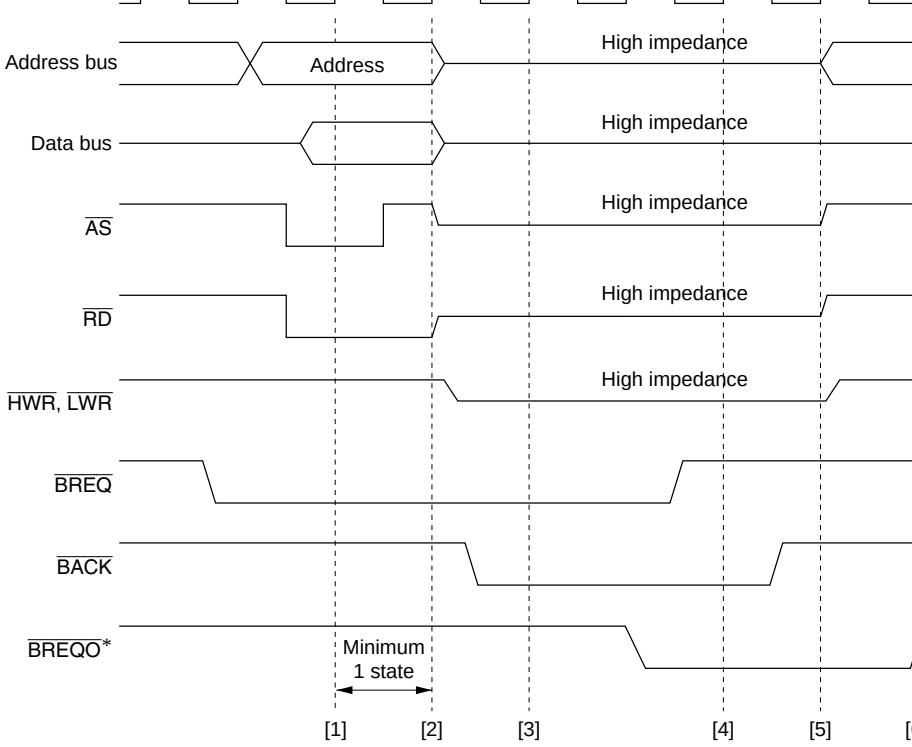
If the BREQOE bit in BCRL is set to 1, when an internal bus master wants to make an
access in the external bus released state, or when a refresh request is generated, the $\overline{\text{BR}}$
driven low and a request can be made off-chip to drop the bus request.

When the $\overline{\text{BREQ}}$ pin is driven high, the $\overline{\text{BACK}}$ pin is driven high at the prescribed timi
external bus released state is terminated.

In the event of simultaneous external bus release request, refresh request, and external a
request generation, the order of priority is as follows:

(High) Refresh > External bus release > Internal bus master external access (Lo

D_{15} to D_0	High impedance
$\overline{CS_n}$	High impedance
$\overline{CAS}$	High impedance
$\overline{AS}$	High impedance
$\overline{RD}$	High impedance
$\overline{HWR}$	High impedance
$\overline{LWR}$	High impedance
$\overline{DACK_n}$	High



[1] Low level of $\overline{\text{BREQ}}$ pin is sampled at rise of T_2 state.
[2] $\overline{\text{BACK}}$ pin is driven low at end of CPU read cycle, releasing bus to external bus master.
[3] $\overline{\text{BREQ}}$ pin state is still sampled in external bus released state.
[4] High level of $\overline{\text{BREQ}}$ pin is sampled.
[5] $\overline{\text{BACK}}$ pin is driven high, ending bus release cycle.
[6] $\overline{\text{BREQO}}$ signal goes high 1.5 clocks after $\overline{\text{BACK}}$ signal goes high.

Note: * Output only when BREQOE is set to 1.

Figure 6.47 Bus-Released State Transition Timing

signal. The bus arbiter determines priorities at the prescribed timing, and permits use o
means of a bus request acknowledge signal. The selected bus master then takes posses
bus and begins its operation.

6.12.2 Operation

The bus arbiter detects the bus masters' bus request signals, and if the bus is requested
request acknowledge signal to the bus master making the request. If there are bus requ
more than one bus master, the bus request acknowledge signal is sent to the one with t
priority. When a bus master receives the bus request acknowledge signal, it takes poss
bus until that signal is canceled.

The order of priority of the bus masters is as follows:

(High) DMAC > DTC > CPU (Low)

An internal bus access by an internal bus master, external bus release, and refreshing,
executed in parallel.

In the event of simultaneous external bus release request, refresh request, and internal
external access request generation, the order of priority is as follows:

(High) Refresh > External bus release > Internal bus master external access

The CPU is the lowest-priority bus master, and if a bus request is received from the DT
DMAC, the bus arbiter transfers the bus to the bus master that issued the request. The t
transfer of the bus is as follows:

- The bus is transferred at a break between bus cycles. However, if a bus cycle is exe
  discrete operations, as in the case of a longword-size access, the bus is not transferr
  the operations. See appendix A.5, Bus States during Instruction Execution, for timi
  which the bus is not transferred.
- If the CPU is in sleep mode, it transfers the bus immediately.

DTC

The DTC sends the bus arbiter a request for the bus when an activation request is gener

The DTC can release the bus after a vector read, a register information read (3 states), a
transfer, or a register information write (3 states). It does not release the bus during a re
information read (3 states), a single data transfer, or a register information write (3 state

DMAC

The DMAC sends the bus arbiter a request for the bus when an activation request is ge

In the case of an external request in short address mode or normal mode, and in cycle s
the DMAC releases the bus after a single transfer.

In block transfer mode, it releases the bus after transfer of one block, and in burst mode
completion of a transfer.

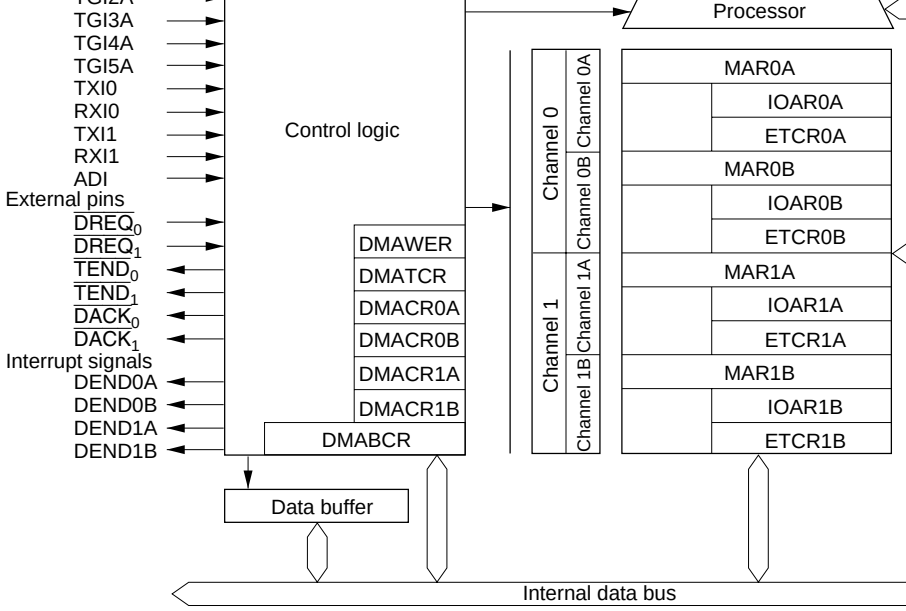
In a power-on reset, the H8S/2655, including the bus controller, enters the reset state a
and an executing bus cycle is discontinued.

In a manual reset, the bus controller's registers and internal state are maintained, and a
external bus cycle is completed. In this case, $\overline{\text{WAIT}}$ input is ignored and write data is
guaranteed. Also, since the DMAC is initialized by a manual reset, $\overline{\text{DACK}}$ and $\overline{\text{TEND}}$
disabled and these pins become I/O ports controlled by DDR and DR.

7.1.1 Features

The features of the DMAC are listed below.

- Choice of short address mode or full address mode
  Short address mode:
  — Maximum of 4 channels can be used
  — Choice of dual address mode or single address mode
  — In dual address mode, one of the two addresses, transfer source and transfer de specified as 24 bits and the other as16 bits
  — In single address mode, transfer source or transfer destination address only is s 24 bits
  — In single address mode, transfer can be performed in one bus cycle
  — Choice of sequential mode, idle mode, or repeat mode for dual address mode a address mode
  Full address mode:
  — Maximum of 2 channels can be used
  — Transfer source and transfer destination address specified as 24 bits
  — Choice of normal mode or block transfer mode
- 16-Mbyte address space can be specified directly
- Byte or word can be set as the transfer unit
- Activation sources: internal interrupt, external request, auto-request (depending on mode)
  — Six 16-bit timer-pulse unit (TPU) compare match/input capture interrupts
  — Serial communication interface (SCI0, SCI1) transmission complete interrupt, complete interrupt
  — A/D converter conversion end interrupt
  — External request
  — Auto-request



Legend:
DMAWER : DMA write enable register
DMATCR : DMA terminal control register
DMABCR : DMA band control register (for all channels)
DMACR : DMA control register
MAR : Memory address register
IOAR : I/O address register
ETCR : Executive transfer counter register

Figure 7.1 Block Diagram of DMAC

Dual address mode • Sequential mode — 1-byte or 1-word transfer executed for one transfer request — Memory address incremented/ decremented by 1 or 2 — 1 to 65536 transfers • Idle mode — 1-byte or 1-word transfer executed for one transfer request — Memory address fixed — 1 to 65536 transfers • Repeat mode — 1-byte or 1-word transfer executed for one transfer request — Memory address incremented/ decremented by 1 or 2 — After specified number of transfers (1 to 256), initial state is restored and operation continues	• TPU channel 0 to 5 compare match/input capture A interrupt • SCI transmission complete interrupt • SCI reception complete interrupt • A/D converter conversion end interrupt • External request	24/16	16/
Single address mode • 1-byte or 1-word transfer executed for one transfer request • Transfer in 1 bus cycle using $\overline{\text{DACK}}$ pin in place of address specifying I/O • Specifiable for modes (1) to (3)	• External request	24/$\overline{\text{DACK}}$	$\overline{\text{DA}}$

— Transfers continue for the specified number of times (1 to 65536) — Choice of burst or cycle steal transfer External request — 1-byte or 1-word transfer executed for one transfer request — 1 to 65536 transfers	• External request		
• Block transfer mode — Specified block size transfer executed for one transfer request — 1 to 65536 transfers — Either source or destination specifiable as block area — Block size: 1 to 256 bytes or words	• TPU channel 0 to 5 compare match/input capture A interrupt • SCI transmission complete interrupt • SCI reception complete interrupt • External request • A/D converter conversion end interrupt	24	24

When the $\overline{\text{DREQ}}$ pin is used, do not designate the corresponding port for output.

With regard to the $\overline{\text{DACK}}$ pins, setting single address transfer automatically sets the c
port to output, functioning as a $\overline{\text{DACK}}$ pin.

With regard to the $\overline{\text{TEND}}$ pins, whether or not the corresponding port is used as a $\overline{\text{TEN}}$
be specified by means of a register setting.

Table 7.2 DMAC Pins

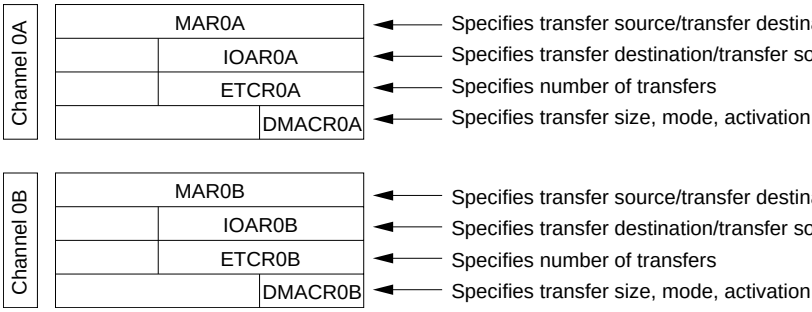
Channel	Pin Name	Symbol	I/O	Function
0	DMA request 0	$\overline{\text{DREQ}}_0$	Input	DMAC channel 0 ex request
	DMA transfer acknowledge 0	$\overline{\text{DACK}}_0$	Output	DMAC channel 0 si transfer acknowledg
	DMA transfer end 0	$\overline{\text{TEND}}_0$	Output	DMAC channel 0 tra
1	DMA request 1	$\overline{\text{DREQ}}_1$	Input	DMAC channel 1 ex request
	DMA transfer acknowledge 1	$\overline{\text{DACK}}_1$	Output	DMAC channel 1 si transfer acknowledg
	DMA transfer end 1	$\overline{\text{TEND}}_1$	Output	DMAC channel 1 tra

	H'FEE4	16 bits	I/O address register 0A	IOAR0A	R/W
	H'FEE6	16 bits	Transfer count register 0A	ETCR0A	R/W
	H'FEE8	16 bits	Memory address register 0B	MAR0B	R/W
	H'FEEC	16 bits	I/O address register 0B	IOAR0B	R/W
	H'FEEE	16 bits	Transfer count register 0B	ETCR0B	R/W
1	H'FEF0	16 bits	Memory address register 1A	MAR1A	R/W
	H'FEF4	16 bits	I/O address register 1A	IOAR1A	R/W
	H'FEF6	16 bits	Transfer count register 1A	ETCR1A	R/W
	H'FEF8	16 bits	Memory address register 1B	MAR1B	R/W
	H'FEFC	16 bits	I/O address register 1B	IOAR1B	R/W
	H'FEFE	16 bits	Transfer count register 1B	ETCR1B	R/W
0, 1	H'FF00	8 bits	DMA write enable register	DMAWER	R/W
	H'FF01	8 bits	DMA terminal control register	DMATCR	R/W
	H'FF02	16 bits	DMA control register 0A	DMACR0A	R/W
	H'FF03	16 bits	DMA control register 0B	DMACR0B	R/W
	H'FF04	16 bits	DMA control register 1A	DMACR1A	R/W
	H'FF05	16 bits	DMA control register 1B	DMACR1B	R/W
	H'FF06	16 bits	DMA band control register	DMABCR	R/W
	H'FF3C	8 bits	Module stop control register	MSTPCR	R/W

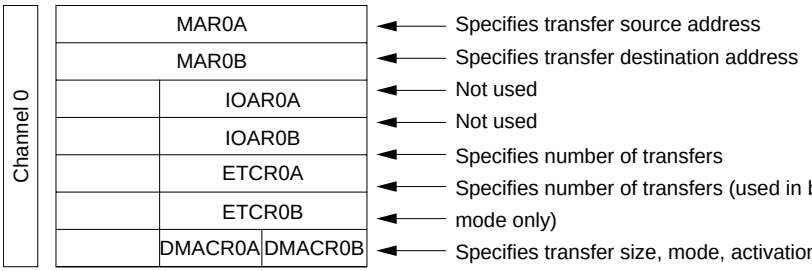
Note: * Lower 16 bits of the address.

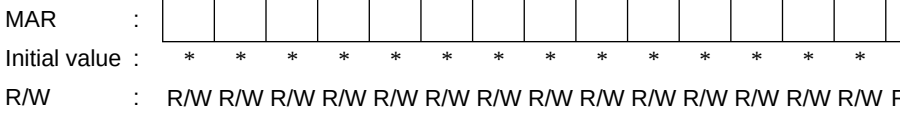
Channel 0)

FAE0	Description
0	Short address mode specified (channels A and B operate independently)



1	Full address mode specified (channels A and B operate in combination)





Legend: *: Undefined

MAR is a 32-bit readable/writable register that specifies the transfer source address or d
address.

The upper 8 bits of MAR are reserved: they are always read as 0, and cannot be modifi

Whether MAR functions as the source address register or as the destination address reg
selected by means of the DTDIR bit in DMACR.

MAR is incremented or decremented each time a byte or word transfer is executed, so t
address specified by MAR is constantly updated. For details, see section 7.2.4, DMA C
Register (DMACR).

MAR is not initialized by a reset or in standby mode.

IOAR is a 16-bit readable/writable register that specifies the lower 16 bits of the trans
address or destination address. The upper 8 bits of the transfer address are automatical
H'FF.

Whether IOAR functions as the source address register or as the destination address re
be selected by means of the DTDIR bit in DMACR.

IOAR is invalid in single address mode.

IOAR is not incremented or decremented each time a transfer is executed, so that the a
specified by IOAR is fixed.

IOAR is not initialized by a reset or in standby mode.

Transfer Counter

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
ETCR	:														
Initial value	:	*	*	*	*	*	*	*	*	*	*	*	*	*	*
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

Legend: *: Undefined

In sequential mode and idle mode, ETCR functions as a 16-bit transfer counter (with a
of 1 to 65536). ETCR is decremented by 1 each time a transfer is performed, and when
reaches H'0000, the DTE bit in DMABCR is cleared, and transfer ends.

(2) Repeat Mode

Transfer Number Storage

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

Transfer Counter

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

Legend: *: Undefined

In repeat mode, ETCR functions as transfer counter ETCRL (with a count range of 1 to
transfer number storage register ETCRH. ETCRL is decremented by 1 each time a tran
performed, and when the count reaches H'00, ETCRL is loaded with the value in ETCR
point, MAR is automatically restored to the value it had when the count was started. Th

DMACR	:	DTSZ	DTID	RPE	DTDIR	DTF3	DTF2	DTF1
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

DMACR is an 8-bit readable/writable register that controls the operation of each DMA

DMACR is initialized to H'00 by a reset, and in standby mode.

Bit 7—Data Transfer Size (DTSZ): Selects the size of data to be transferred at one t

Bit 7

DTSZ	Description
0	Byte-size transfer
1	Word-size transfer

Bit 6—Data Transfer Increment/Decrement (DTID): Selects incrementing or decre
MAR every data transfer in sequential mode or repeat mode.

In idle mode, MAR is neither incremented nor decremented.

Bit 6

DTID	Description
0	MAR is incremented after a data transfer • When DTSZ = 0, MAR is incremented by 1 after a transfer • When DTSZ = 1, MAR is incremented by 2 after a transfer
1	MAR is decremented after a data transfer • When DTSZ = 0, MAR is decremented by 1 after a transfer • When DTSZ = 1, MAR is decremented by 2 after a transfer

	1	Transfer in idle mode (with transfer end interrupt)

For details of operation in sequential, idle, and repeat mode, see section 7.5.2, Sequenti section 7.5.3, Idle Mode, and section 7.5.4, Repeat Mode.

Bit 4—Data Transfer Direction (DTDIR): Used in combination with the SAE bit in I to specify the data transfer direction (source or destination). The function of this bit is t different in dual address mode and single address mode.

DMABCR	Bit 4	
SAE	**DTDIR**	**Description**
0	0	Transfer with MAR as source address and IOAR as destinatior address (
	1	Transfer with IOAR as source address and MAR as destinatior
1	0	Transfer with MAR as source address and $\overline{\text{DACK}}$ pin as write s
	1	Transfer with $\overline{\text{DACK}}$ pin as read strobe and MAR as destinatio

0	0	0	0	—
			1	Activated by A/D converter conversion end int
		1	0	—
			1	—
	1	0	0	Activated by SCI channel 0 transmission comp
			1	Activated by SCI channel 0 reception complete
		1	0	Activated by SCI channel 1 transmission comp
			1	Activated by SCI channel 1 reception complete
1	0	0	0	Activated by TPU channel 0 compare match/in A interrupt
			1	Activated by TPU channel 1 compare match/in A interrupt
		1	0	Activated by TPU channel 2 compare match/in A interrupt
			1	Activated by TPU channel 3 compare match/in A interrupt
	1	0	0	Activated by TPU channel 4 compare match/in A interrupt
			1	Activated by TPU channel 5 compare match/in A interrupt
		1	0	—
			1	—

	1	0	0	Activated by SCI channel 0 transmission compl
			1	Activated by SCI channel 0 reception complete
		1	0	Activated by SCI channel 1 transmission compl
			1	Activated by SCI channel 1 reception complete
1	0	0	0	Activated by TPU channel 0 compare match/inp A interrupt
			1	Activated by TPU channel 1 compare match/inp A interrupt
		1	0	Activated by TPU channel 2 compare match/inp A interrupt
			1	Activated by TPU channel 3 compare match/inp A interrupt
	1	0	0	Activated by TPU channel 4 compare match/inp A interrupt
			1	Activated by TPU channel 5 compare match/inp A interrupt
		1	0	—
			1	—

Note: * Detected as a low level in the first transfer after transfer is enabled.

The same factor can be selected for more than one channel. In this case, activation start highest-priority channel according to the relative channel priorities. For relative channe see section 7.5.13, DMAC Multi-Channel Operation.

Bit	:	7	6	5	4	3	2	1
DMABCRL	:	DTME1	DTE1	DTME0	DTE0	DTIE1B	DTIE1A	DTIE0B
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

DMABCR is a 16-bit readable/writable register that controls the operation of each DM channel.

DMABCR is initialized to H'0000 by a reset, and in standby mode.

Bit 15—Full Address Enable 1 (FAE1): Specifies whether channel 1 is to be used in address mode or full address mode.

In short address mode, channels 1A and 1B are used as independent channels.

Bit 15

FAE1	Description
0	Short address mode
1	Full address mode

Bit 14—Full Address Enable 0 (FAE0): Specifies whether channel 0 is to be used in address mode or full address mode.

In short address mode, channels 0A and 0B are used as independent channels.

Bit 14

FAE0	Description
0	Short address mode
1	Full address mode

This bit is invalid in full address mode.

Bit 12—Single Address Enable 0 (SAE0): Specifies whether channel 0B is to be used transfer in dual address mode or single address mode.

Bit 12

SAE0	Description
0	Transfer in dual address mode
1	Transfer in single address mode

This bit is invalid in full address mode.

Bits 11 to 8—Data Transfer Acknowledge (DTA): These bits enable or disable clearing DMA transfer is performed, of the internal interrupt source selected by the data transfer setting.

When DTE = 1 and DTA = 1, the internal interrupt source selected by the data transfer setting is cleared automatically by DMA transfer. When DTE = 1 and DTA = 1, the interrupt source selected by the data transfer factor setting does not issue an interrupt request to CPU or DTC.

When DTE = 1 and DTA = 0, the internal interrupt source selected by the data transfer setting is not cleared when a transfer is performed, and can issue an interrupt request to or DTC in parallel. In this case, the interrupt source should be cleared by the CPU or DTC transfer.

When DTE = 0, the internal interrupt source selected by the data transfer factor setting interrupt request to the CPU or DTC regardless of the DTA bit setting.

1	Clearing of selected internal interrupt source at time of DMA transfer is en

Bit 10—Data Transfer Acknowledge 1A (DTA1A): Enables or disables clearing, wi
transfer is performed, of the internal interrupt source selected by the channel 1A data t
factor setting.

Bit 10

DTA1A	Description
0	Clearing of selected internal interrupt source at time of DMA transfer is dis
1	Clearing of selected internal interrupt source at time of DMA transfer is en

Bit 9—Data Transfer Acknowledge 0B (DTA0B): Enables or disables clearing, whe
transfer is performed, of the internal interrupt source selected by the channel 0B data t
factor setting.

Bit 9

DTA0B	Description
0	Clearing of selected internal interrupt source at time of DMA transfer is dis
1	Clearing of selected internal interrupt source at time of DMA transfer is en

1	Clearing of selected internal interrupt source at time of DMA transfer is ena

Bits 7 to 4—Data Transfer Enable (DTE): When DTE = 0, data transfer is disabled a activation source selected by the data transfer factor setting is ignored. If the activation an internal interrupt, an interrupt request is issued to the CPU or DTC. If the DTIE bit i when DTE = 0, the DMAC regards this as indicating the end of a transfer, and issues a end interrupt request to the CPU or DTC.

The conditions for the DTE bit being cleared to 0 are as follows:

- When initialization is performed
- When the specified number of transfers have been completed in a transfer mode oth repeat mode
- When 0 is written to the DTE bit to forcibly abort the transfer, or for a similar reaso

When DTE = 1, data transfer is enabled and the DMAC waits for a request by the activ source selected by the data transfer factor setting. When a request is issued by the activ source, DMA transfer is executed.

The condition for the DTE bit being set to 1 is as follows:

- When 1 is written to the DTE bit after the DTE bit is read as 0

Bit 7—Data Transfer Enable 1B (DTE1B): Enables or disables data transfer on chan

Bit 7

DTE1B	Description	
0	Data transfer disabled	(
1	Data transfer enabled	

Bit 5	
DTE0B	**Description**
0	Data transfer disabled
1	Data transfer enabled

Bit 4—Data Transfer Enable 0A (DTE0A): Enables or disables data transfer on cha

Bit 4	
DTE0A	**Description**
0	Data transfer disabled
1	Data transfer enabled

Bits 3 to 0—Data Transfer End Interrupt Enable (DTIE): These bits enable or dis interrupt to the CPU or DTC when transfer ends. If the DTIE bit is set to 1 when DTE DMAC regards this as indicating the end of a transfer, and issues a transfer end interru the CPU or DTC.

A transfer end interrupt can be canceled either by clearing the DTIE bit to 0 in the inte handling routine, or by performing processing to continue transfer by setting the trans and address register again, and then setting the DTE bit to 1.

Bit 3—Data Transfer Interrupt Enable 1B (DTIE1B): Enables or disables the char transfer end interrupt.

Bit 3	
DTIE1B	**Description**
0	Transfer end interrupt disabled
1	Transfer end interrupt enabled

Bit 1—Data Transfer Interrupt Enable 0B (DTIE0B): Enables or disables the chann
transfer end interrupt.

Bit 1

DTIE0B	Description
0	Transfer end interrupt disabled (I
1	Transfer end interrupt enabled

Bit 0—Data Transfer Interrupt Enable 0A (DTIE0A): Enables or disables the chann
transfer end interrupt.

Bit 0

DTIE0A	Description
0	Transfer end interrupt disabled (I
1	Transfer end interrupt enabled

MAR	:	—	—	—	—	—	—	—	—						
Initial value	:	0	0	0	0	0	0	0	0	*	*	*	*	*	*
R/W	:	—	—	—	—	—	—	—	—	R/W	R/W	R/W	R/W	R/W	R/W

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
MAR	:														
Initial value	:	*	*	*	*	*	*	*	*	*	*	*	*	*	*
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

Legend: *: Undefined

MAR is a 32-bit readable/writable register; MARA functions as the transfer source ad
register, and MARB as the destination address register.

MAR is composed of two 16-bit registers, MARH and MARL. The upper 8 bits of MA
reserved: they are always read as 0, and cannot be modified.

MAR is incremented or decremented each time a byte or word transfer is executed, so
source or destination memory address can be updated automatically. For details, see s
DMA Control Register (DMACR).

MAR is not initialized by a reset or in standby mode.

ETCR is not initialized by a reset or in standby mode.

(1) Normal Mode

ETCRA:

Transfer Counter

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	
ETCR	:															
Initial value	:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	
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

Legend: *: Undefined

In normal mode, ETCRA functions as a 16-bit transfer counter. ETCRA is decremented time a transfer is performed, and transfer ends when the count reaches H'0000. ETCRB at this time.

ETCRB:
ETCRB is not used in normal mode.

Block size counter

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

Legend: *: Undefined

ETCRB:

Block Transfer Counter

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
ETCRB	:														
Initial value	:	*	*	*	*	*	*	*	*	*	*	*	*	*	*
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

In block transfer mode, ETCRAL functions as an 8-bit block size counter and ETCRA
block size. ETCRAL is decremented each time a 1-byte or 1-word transfer is performe
the count reaches H'00, ETCRAL is loaded with the value in ETCRAH. So by setting
size in ETCRAH and ETCRAL, it is possible to repeatedly transfer blocks consisting
desired number of bytes or words.

ETCRB functions in block transfer mode, as a 16-bit block transfer counter. ETCRB i
decremented by 1 each time a block is transferred, and transfer ends when the count re
H'0000.

Bit	:	15	14	13	12	11	10	9
DMACRA	:	DTSZ	SAID	SAIDE	BLKDIR	BLKE	—	—
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

DMACRB:

Bit	:	7	6	5	4	3	2	1
DMACRB	:	—	DAID	DAIDE	—	DTF3	DTF2	DTF1
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—Data Transfer Size (DTSZ): Selects the size of data to be transferred at one t

Bit 15

DTSZ	Description
0	Byte-size transfer
1	Word-size transfer

Bit 14—Source Address Increment/Decrement (SAID)

		• When DTSZ = 0, MARA is incremented by 1 after a transf
		• When DTSZ = 1, MARA is incremented by 2 after a transf
1	0	MARA is fixed
	1	MARA is decremented after a data transfer
		• When DTSZ = 0, MARA is decremented by 1 after a trans
		• When DTSZ = 1, MARA is decremented by 2 after a trans

Bit 12—Block Direction (BLKDIR)

Bit 11—Block Enable (BLKE): These bits specify whether normal mode or block tra
is to be used. If block transfer mode is specified, the BLKDIR bit specifies whether th
or the destination side is to be the block area.

Bit 12	Bit 11	
BLKDIR	**BLKE**	**Description**
0	0	Transfer in normal mode
	1	Transfer in block transfer mode, destination side is block area
1	0	Transfer in normal mode
	1	Transfer in block transfer mode, source side is block area

For operation in normal mode and block transfer mode, see section 7.5, Operation.

Bits 10 to 7—Reserved: Can be read or written to.

Bit 6—Destination Address Increment/Decrement (DAID)

		• When DTSZ = 0, MARB is incremented by 1 after a transfe • When DTSZ = 1, MARB is incremented by 2 after a transfe
1	0	MARB is fixed
	1	MARB is decremented after a data transfer • When DTSZ = 0, MARB is decremented by 1 after a transf • When DTSZ = 1, MARB is decremented by 2 after a transf

Bit 4—Reserved: Can be read or written to.

Bits 3 to 0—Data Transfer Factor (DTF3 to DTF0): These bits select the data transf (activation source). The factors that can be specified differ between normal mode and b transfer mode.

- Normal Mode

Bit 3	Bit 2	Bit 1	Bit 0	
DTF3	**DTF2**	**DTF1**	**DTF0**	**Description**
0	0	0	0	— (I
			1	—
		1	0	Activated by $\overline{\text{DREQ}}$ pin falling edge input
			1	Activated by $\overline{\text{DREQ}}$ pin low-level input
	1	0	*	—
		1	0	Auto-request (cycle steal)
			1	Auto-request (burst)
1	*	*	*	—

Legend: *: Don't care

	1	0	0	Activated by SCI channel 0 transmission comp
			1	Activated by SCI channel 0 reception complete
		1	0	Activated by SCI channel 1 transmission comp
			1	Activated by SCI channel 1 reception complete
1	0	0	0	Activated by TPU channel 0 compare match/in A interrupt
			1	Activated by TPU channel 1 compare match/in A interrupt
		1	0	Activated by TPU channel 2 compare match/in A interrupt
			1	Activated by TPU channel 3 compare match/in A interrupt
	1	0	0	Activated by TPU channel 4 compare match/in A interrupt
			1	Activated by TPU channel 5 compare match/in A interrupt
		1	0	—
			1	—

Note: * Detected as a low level in the first transfer after transfer is enabled.

The same factor can be selected for more than one channel. In this case, activation sta highest-priority channel according to the relative channel priorities. For relative chann see section 7.5.13, DMAC Multi-Channel Operation.

Bit	:	7	6	5	4	3	2	1
DMABCRL	:	DTME1	DTE1	DTME0	DTE0	DTIE1B	DTIE1A	DTIE0B
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

DMABCR is a 16-bit readable/writable register that controls the operation of each DM channel.

DMABCR is initialized to H'0000 by a reset, and in standby mode.

Bit 15—Full Address Enable 1 (FAE1): Specifies whether channel 1 is to be used in address mode or full address mode.

In full address mode, channels 1A and 1B are used together as a single channel.

Bit 15

FAE1	Description
0	Short address mode
1	Full address mode

Bit 14—Full Address Enable 0 (FAE0): Specifies whether channel 0 is to be used in address mode or full address mode.

In full address mode, channels 0A and 0B are used together as a single channel.

Bit 14

FAE0	Description
0	Short address mode
1	Full address mode

CPU or DTC.

When the DTE = 1 and the DTA = 0, the internal interrupt source selected by the data
factor setting is not cleared when a transfer is performed, and can issue an interrupt re
CPU or DTC in parallel. In this case, the interrupt source should be cleared by the CP
transfer.

When the DTE = 0, the internal interrupt source selected by the data transfer factor se
an interrupt request to the CPU or DTC regardless of the DTA bit setting.

The state of the DTME bit does not affect the above operations.

Bit 11—Data Transfer Acknowledge 1 (DTA1): Enables or disables clearing, when
transfer is performed, of the internal interrupt source selected by the channel 1 data tra
setting.

Bit 11

DTA1	Description
0	Clearing of selected internal interrupt source at time of DMA transfer is dis
1	Clearing of selected internal interrupt source at time of DMA transfer is en

Bit 9—Data Transfer Acknowledge 0 (DTA0): Enables or disables clearing, when I
transfer is performed, of the internal interrupt source selected by the channel 0 data tra
setting.

Bit 9

DTA0	Description
0	Clearing of selected internal interrupt source at time of DMA transfer is dis
1	Clearing of selected internal interrupt source at time of DMA transfer is en

block transfer mode, however, the DTME bit is not cleared by an NMI interrupt, and tr
not interrupted.

The conditions for the DTME bit being cleared to 0 are as follows:

- When initialization is performed
- When NMI is input in burst mode
- When 0 is written to the DTME bit

The condition for DTME being set to 1 is as follows:

- When 1 is written to DTME after DTME is read as 0

Bit 7—Data Transfer Master Enable 1 (DTME1): Enables or disables data transfer c
1.

Bit 7

DTME1	Description
0	Data transfer disabled. In burst mode, cleared to 0 by an NMI interrupt (I
1	Data transfer enabled

Bit 5—Data Transfer Master Enable 0 (DTME0): Enables or disables data transfer c
0.

Bit 5

DTME0	Description
0	Data transfer disabled. In normal mode, cleared to 0 by an NMI interrupt (I
1	Data transfer enabled

- When the specified number of transfers have been completed
- When 0 is written to the DTE bit to forcibly abort the transfer, or for a similar reas

When DTE = 1 and DTME = 1, data transfer is enabled and the DMAC waits for a req activation source selected by the data transfer factor setting. When a request is issued activation source, DMA transfer is executed.

The condition for the DTE bit being set to 1 is as follows:

- When 1 is written to the DTE bit after the DTE bit is read as 0

Bit 6—Data Transfer Enable 1 (DTE1): Enables or disables data transfer on channe

Bit 6

DTE1	Description
0	Data transfer disabled
1	Data transfer enabled

Bit 4—Data Transfer Enable 0 (DTE0): Enables or disables data transfer on channe

Bit 4

DTE0	Description
0	Data transfer disabled
1	Data transfer enabled

transfer break interrupt.

Bit 3

DTIE1B	Description
0	Transfer break interrupt disabled
1	Transfer break interrupt enabled

Bit 1—Data Transfer Interrupt Enable 0B (DTIE0B): Enables or disables the chann
transfer break interrupt.

Bit 1

DTIE0B	Description
0	Transfer break interrupt disabled
1	Transfer break interrupt enabled

Bit 2—Data Transfer Interrupt Enable 1A (DTIE1A): Enables or disables the char
transfer end interrupt.

Bit 2

DTIE1A	Description
0	Transfer end interrupt disabled
1	Transfer end interrupt enabled

Bit 0—Data Transfer Interrupt Enable 0A (DTIE0A): Enables or disables the char
transfer end interrupt.

Bit 0

DTIE0A	Description
0	Transfer end interrupt disabled
1	Transfer end interrupt enabled

concerned, The restrictions applied by DMAWER are valid for the DTC.

Figure 7.2 shows the transfer areas for activating the DTC with a channel 0A transfer e
interrupt, and reactivating channel 0A. The address register and count register area is re
first DTC transfer, then the control register area is re-set by the second DTC chain trans

When re-setting the control register area, perform masking by setting bits in DMAWER
modification of the contents of the other channels.

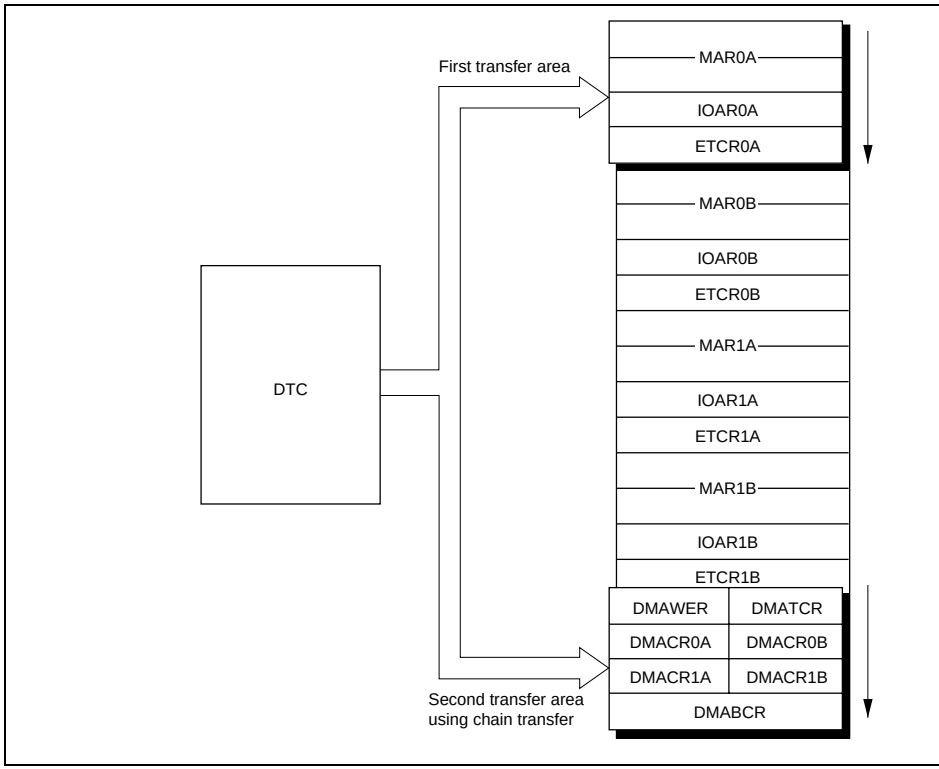


Figure 7.2 Areas for Register Re-Setting by DTC (Example: Channel 0A

DMAWER is initialized to H'00 by a reset, and in standby mode.

Bits 7 to 4—Reserved: Read-only bits, always read as 0.

Bit 3—Write Enable 1B (WE1B): Enables or disables writes to all bits in DMACR1 and 3 in DMABCR, and bit 5 in DMATCR by the DTC.

Bit 3

WE1B	Description
0	Writes to all bits in DMACR1B, bits 11, 7, and 3 in DMABCR, and bit 5 in D are disabled
1	Writes to all bits in DMACR1B, bits 11, 7, and 3 in DMABCR, and bit 5 in D are enabled

Bit 2—Write Enable 1A (WE1A): Enables or disables writes to all bits in DMACR1 10, 6, and 2 in DMABCR by the DTC.

Bit 2

WE1A	Description
0	Writes to all bits in DMACR1A, and bits 10, 6, and 2 in DMABCR are disal
1	Writes to all bits in DMACR1A, and bits 10, 6, and 2 in DMABCR are enab

	are enabled

Bit 0—Write Enable 0A (WE0A): Enables or disables writes to all bits in DMACR0A 8, 4, and 0 in DMABCR.

Bit 0	
WE0A	**Description**
0	Writes to all bits in DMACR0A, and bits 8, 4, and 0 in DMABCR are disable
1	Writes to all bits in DMACR0A, and bits 8, 4, and 0 in DMABCR are enable

Writes by the DTC to bits 15 to 12 (FAE and SAE) in DMABCR are invalid regardless DMAWER settings. These bits should be changed, if necessary, by CPU processing.

In writes by the DTC to bits 7 to 4 (DTE) in DMABCR, 1 can be written without first r To reactivate a channel set to full address mode, write 1 to both Write Enable A and W B for the channel to be reactivated.

MAR, IOAR, and ETCR are always write-enabled regardless of the DMAWER setting modifying these registers, the channel for which the modification is to be made should

transfer end pin output. A port can be set for output automatically, and a transfer end s
by setting the appropriate bit.

DMATCR is initialized to H'00 by a reset, and in standby mode.

Bits 7 and 6—Reserved: Read-only bits, always read as 0.

Bit 5—Transfer End Enable 1 (TEE1): Enables or disables transfer end pin 1 ($\overline{\text{TEN}}$

Bit 5

TEE1	Description
0	$\overline{\text{TEND}}_1$ pin output disabled
1	$\overline{\text{TEND}}_1$ pin output enabled

Bit 4—Transfer End Enable 0 (TEE0): Enables or disables transfer end pin 0 ($\overline{\text{TEN}}$

Bit 4

TEE0	Description
0	$\overline{\text{TEND}}_0$ pin output disabled
1	$\overline{\text{TEND}}_0$ pin output enabled

The $\overline{\text{TEND}}$ pins are assigned only to channel B in short address mode.

The transfer end signal indicates the transfer cycle in which the transfer counter reach
regardless of the transfer source. An exception is block transfer mode, in which the tra
signal indicates the transfer cycle in which the block counter reached 0.

Bits 3 to 0—Reserved: Read-only bits, always read as 0.

MSTPCR is a 16-bit readable/writable register that performs module stop mode control

When the MSTP15 bit in MSTPCR is set to 1, the DMAC operation stops at the end of cycle and a transition is made to module stop mode. For details, see section 21.5, Modu Mode.

MSTPCR is initialized to H'3FFF by a reset and in hardware standby mode. It is not ini software standby mode.

Bit 15—Module Stop (MSTP15): Specifies the DMAC module stop mode.

Bits 15

MSTP15	Description	
0	DMAC module stop mode cleared	(
1	DMAC module stop mode set	

Short address mode	Dual address mode	(1) Sequential mode (2) Idle mode (3) repeat mode	• TPU channel 0 to 5 compare match/input capture A interrupt • SCI transmission complete interrupt • SCI reception complete interrupt • A/D converter conversion end interrupt • External request	• Up to 4 chan operate inde • External requ to channel B • Single addre applies to ch only • Modes (1), (2 can also be s single addres
		(4) Single address mode		
Full address mode		(5) Normal mode	• External request • Auto-request	• Max. 2-chan operation, co channels A a • With auto-re mode transfe steal transfe selected
		(6) Block transfer mode	• TPU channel 0 to 5 compare match/input capture A interrupt • SCI transmission complete interrupt • SCI reception complete interrupt • A/D converter conversion end interrupt • External request	

(2) Idle mode

In response to a single transfer request, the specified number of transfers are carried ou
or one word at a time. An interrupt request can be sent to the CPU or DTC when the sp
number of transfers have been completed. One address is specified as 24 bits, and the o
bits. The transfer source address and transfer destination address are fixed. The transfer
is programmable.

(3) Repeat mode

In response to a single transfer request, the specified number of transfers are carried ou
or one word at a time. When the specified number of transfers have been completed, th
and transfer counter are restored to their original settings, and operation is continued. N
request is sent to the CPU or DTC. One address is specified as 24 bits, and the other as
The transfer direction is programmable.

(4) Single address mode

In response to a single transfer request, the specified number of transfers are carried ou
external memory and an external device, one byte or one word at a time. Unlike dual ad
mode, source and destination accesses are performed in parallel. Therefore, either the s
destination is an external device which can be accessed with a strobe alone, using the $\overline{D}$
One address is specified as 24 bits, and for the other, the pin is set automatically. The tr
direction is programmable.

Modes (1), (2) and (3) can also be specified for single address mode.

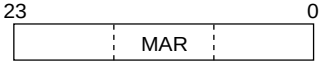
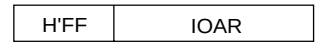
External request: In response to a single transfer request, the specified number of tra
carried out, one byte or one word at a time. An interrupt request can be sent to the CP
when the specified number of transfers have been completed. Both addresses are spec
bits.

(6) Block transfer mode

In response to a single transfer request, a block transfer of the specified block size is c
This is repeated the specified number of times, once each time there is a transfer reque
end of each single block transfer, one address is restored to its original setting. An inte
request can be sent to the CPU or DTC when the specified number of block transfers h
completed. Both addresses are specified as 24 bits.

Table 7.6 summarizes register functions in sequential mode.

Table 7.6 Register Functions in Sequential Mode

Register	Function		Initial Setting	Oper
	DTDIR = 0	DTDIR = 1		
	Source address register	Destination address register	Start address of transfer destination or transfer source	Incre decre every
	Destination address register	Source address register	Start address of transfer source or transfer destination	Fixed
	Transfer counter		Number of transfers	Decre every transf when reach

Legend:
MAR : Memory address register
IOAR : I/O address register
ETCR : Transfer count register
DTDIR : Data transfer direction bit

MAR specifies the start address of the transfer source or transfer destination as 24 bits. incremented or decremented by 1 or 2 each time a byte or word is transferred.

IOAR specifies the lower 16 bits of the other address. The 8 bits above IOAR have a va H'FF.

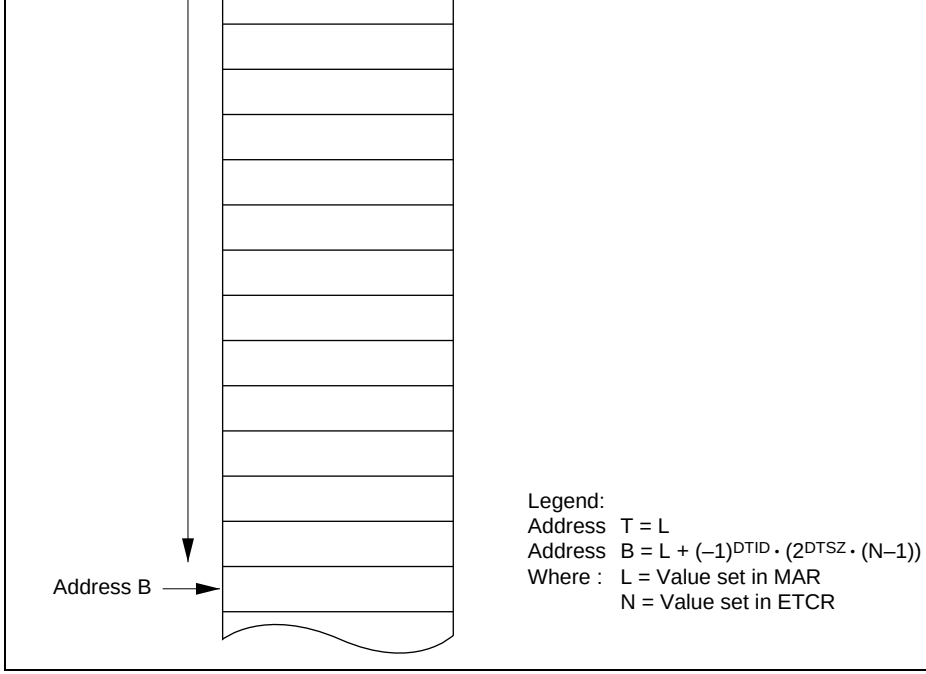
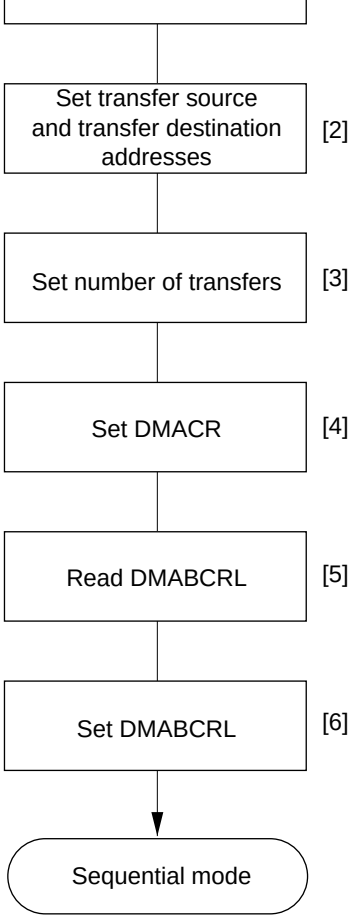


Figure 7.3 Operation in Sequential Mode

The number of transfers is specified as 16 bits in ETCR. ETCR is decremented by 1 e transfer is executed, and when its value reaches H'0000, the DTE bit is cleared and tra If the DTIE bit is set to 1 at this time, an interrupt request is sent to the CPU or DTC.

The maximum number of transfers, when H'0000 is set in ETCR, is 65,536.

Transfer requests (activation sources) consist of A/D converter conversion end interru requests, SCI transmission complete and reception complete interrupts, and TPU chan compare match/input capture A interrupts. External requests can be set for channel B



destination address in MAR and IOAR.

[3] Set the number of transfers in ETCR.

[4] Set each bit in DMACR.
- Set the transfer data size with the DTSZ
- Specify whether MAR is to be incremen decremented with the DTID bit.
- Clear the RPE bit to 0 to select sequent mode.
- Specify the transfer direction with the D bit.
- Select the activation source with bits DT DTF0.

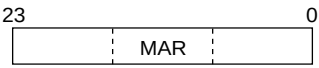
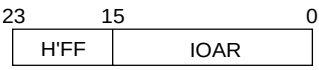
[5] Read the DTE bit in DMABCRL as 0.

[6] Set each bit in DMABCRL.
- Specify enabling or disabling of transfer interrupts with the DTIE bit.
- Set the DTE bit to 1 to enable transfer.

Figure 7.4 Example of Sequential Mode Setting Procedure

Table 7.7 summarizes register functions in idle mode.

Table 7.7 Register Functions in Idle Mode

Register	Function		Initial Setting	Op
	DTDIR = 0	DTDIR = 1		
	Source address register	Destination address register	Start address of transfer destination or transfer source	Fix
	Destination address register	Source address register	Start address of transfer source or transfer destination	Fix
	Transfer counter		Number of transfers	Dec eve tra whe rea

Legend:
MAR : Memory address register
IOAR : I/O address register
ETCR : Transfer count register
DTDIR : Data transfer direction bit

MAR specifies the start address of the transfer source or transfer destination as 24 bits neither incremented nor decremented each time a byte or word is transferred.

IOAR specifies the lower 16 bits of the other address. The 8 bits above IOAR have a v H'FF.

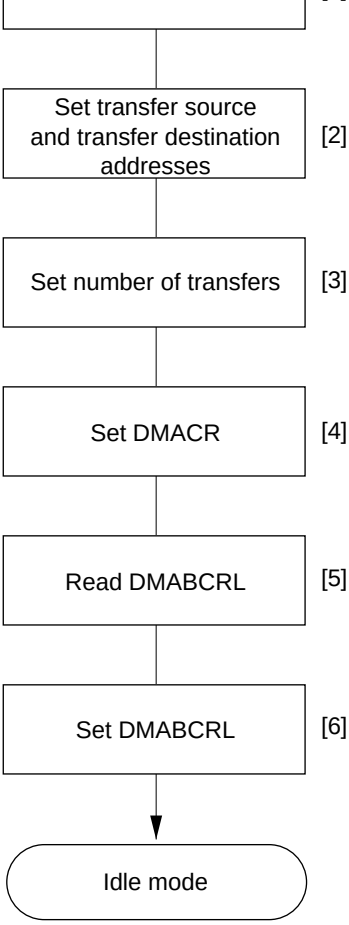
Figure 7.5 Operation in Idle Mode

The number of transfers is specified as 16 bits in ETCR. ETCR is decremented by 1 ea transfer is executed, and when its value reaches H'0000, the DTE bit is cleared and tran If the DTIE bit is set to 1 at this time, an interrupt request is sent to the CPU or DTC.

The maximum number of transfers, when H'0000 is set in ETCR, is 65,536.

Transfer requests (activation sources) consist of A/D converter conversion end interrup requests, SCI transmission complete and reception complete interrupts, and TPU chann compare match/input capture A interrupts. External requests can be set for channel B o

When the DMAC is used in single address mode, only channel B can be set.



[2] Set the transfer source address and trans
destination address in MAR and IOAR.

[3] Set the number of transfers in ETCR.

[4] Set each bit in DMACR.
- Set the transfer data size with the DTSZ
- Specify whether MAR is to be incremen
decremented with the DTID bit.
- Set the RPE bit to 1.
- Specify the transfer direction with the D
bit.
- Select the activation source with bits DT
DTF0.

[5] Read the DTE bit in DMABCRL as 0.

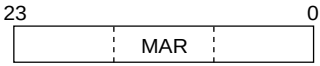
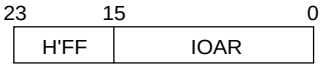
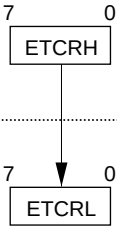
[6] Set each bit in DMABCRL.
- Set the DTIE bit to 1.
- Set the DTE bit to 1 to enable transfer.

Figure 7.6 Example of Idle Mode Setting Procedure

by the DTDIR bit in DMACR.

Table 7.8 summarizes register functions in repeat mode.

Table 7.8 Register Functions in Repeat Mode

Register	Function		Initial Setting	Opera
	DTDIR = 0	DTDIR = 1		
	Source address register	Destination address register	Start address of transfer destination or transfer source	Increm decre every Initial restor value H'000
	Destination address register	Source address register	Start address of transfer source or transfer destination	Fixed
	Holds number of transfers Transfer counter		Number of transfers Number of transfers	Fixed Decre every Loade ETCR when reach

Legend:
MAR : Memory address register
IOAR : I/O address register
ETCR : Transfer count register
DTDIR : Data transfer direction bit

In repeat mode, ETCRL functions as the transfer counter, and ETCRH is used to hold of transfers. ETCRL is decremented by 1 each time a transfer is executed, and when it reaches H'00, it is loaded with the value in ETCRH. At the same time, the value set in restored in accordance with the values of the DTSZ and DTID bits in DMACR. The M restoration operation is as shown below.

$$MAR = MAR - (-1)^{DTID \cdot 2DTSZ} \cdot ETCRH$$

The same value should be set in ETCRH and ETCRL.

In repeat mode, operation continues until the DTE bit is cleared. To end the transfer o therefore, you should clear the DTE bit to 0. A transfer end interrupt request is not ser or DTC.

By setting the DTE bit to 1 again after it has been cleared, the operation can be restart transfer after that terminated when the DTE bit was cleared.

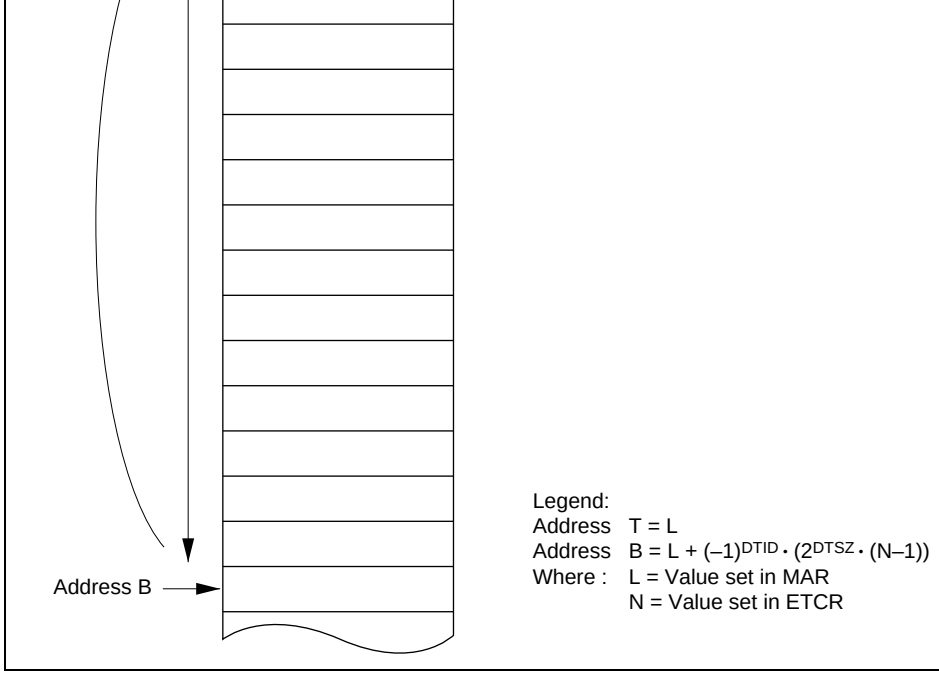
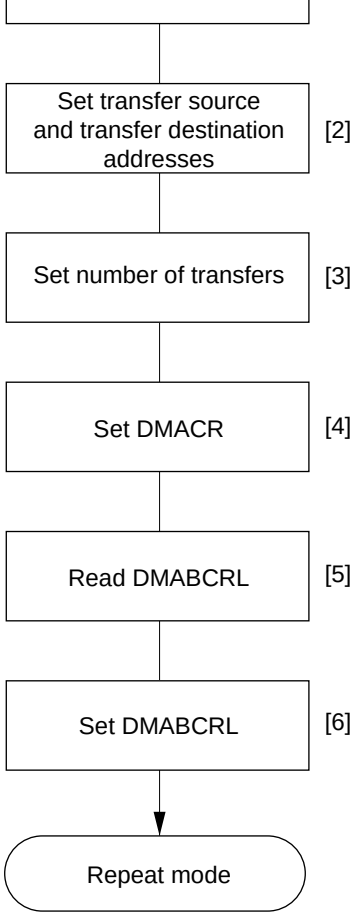


Figure 7.7 Operation in Repeat mode

Transfer requests (activation sources) consist of A/D converter conversion end interrup
requests, SCI transmission complete and reception complete interrupts, and TPU chann
compare match/input capture A interrupts. External requests can be set for channel B o



destination address in MAR and IOAR.

[3] Set the number of transfers in both ETCR ETCRL.

[4] Set each bit in DMACR.
- Set the transfer data size with the DTS
- Specify whether MAR is to be increme decremented with the DTID bit.
- Set the RPE bit to 1.
- Specify the transfer direction with the bit.
- Select the activation source with bits D DTF0.

[5] Read the DTE bit in DMABCRL as 0.

[6] Set each bit in DMABCRL.
- Clear the DTIE bit to 0.
- Set the DTE bit to 1 to enable transfer

Figure 7.8 Example of Repeat Mode Setting Procedure

Table 7.9 Register Functions in Single Address Mode

Register	Function		Initial Setting	Ope
	DTDIR = 0	DTDIR = 1		
	Source address register	Destination address register	Start address of transfer destination or transfer source	*
$\overline{\text{DACK}}$ pin	Write strobe	Read strobe	(Set automatically by SAE bit; IOAR is invalid)	Strol exte
	Transfer counter		Number of transfers	*

Legend:

MAR : Memory address register

IOAR : I/O address register

ETCR : Transfer count register

DTDIR : Data transfer direction bit

$\overline{\text{DACK}}$: Data transfer acknowledge

Note: * See the operation descriptions in sections 7.5.2, Sequential Mode, 7.5.3, Idl and 7.5.4, Repeat Mode.

MAR specifies the start address of the transfer source or transfer destination as 24 bits.

IOAR is invalid; in its place the strobe for external devices ($\overline{\text{DACK}}$) is output.

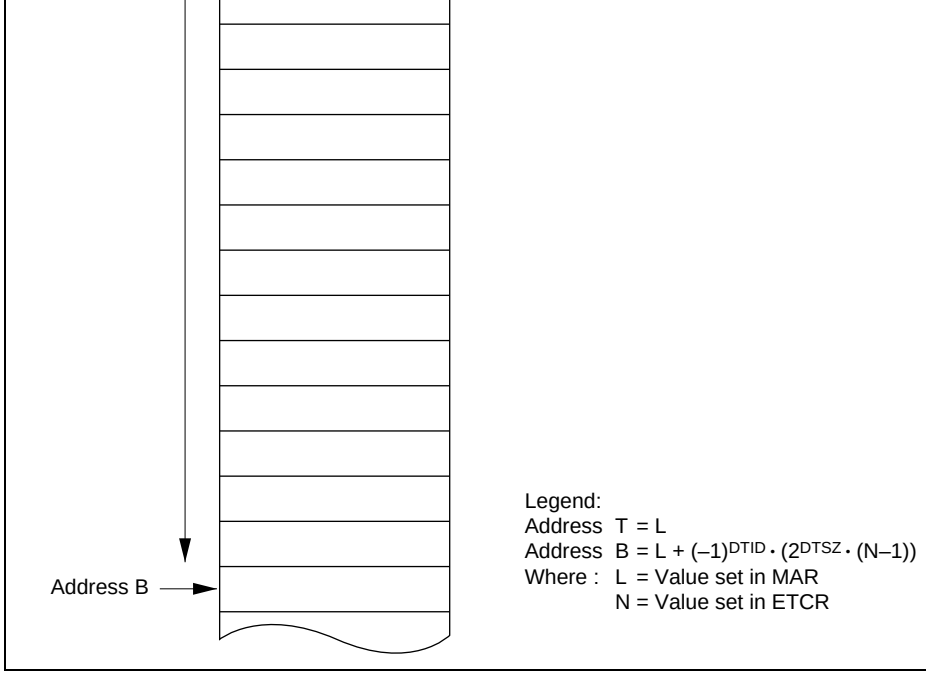
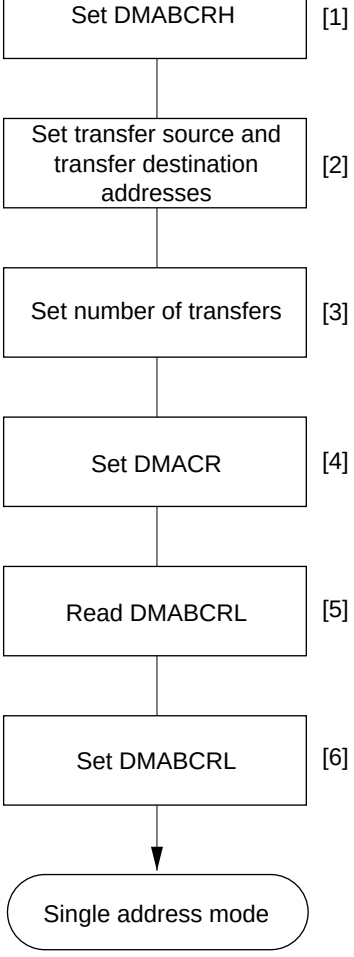


Figure 7.9 Operation in Single Address Mode (When Sequential Mode Is Sp



[1] Set DMABCRH.
- Specify enabling or disabling of interna interrupt clearing with the DTA bit.

[2] Set the transfer source address/transfer destination address in MAR.

[3] Set the number of transfers in ETCR.

[4] Set each bit in DMACR.
- Set the transfer data size with the DTS
- Specify whether MAR is to be incremer decremented with the DTID bit.
- Clear the RPE bit to 0 to select sequen mode.
- Specify the transfer direction with the D bit.
- Select the activation source with bits D DTF0.

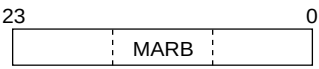
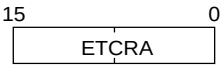
[5] Read the DTE bit in DMABCRL as 0.

[6] Set each bit in DMABCRL.
- Specify enabling or disabling of transfe interrupts with the DTIE bit.
- Set the DTE bit to 1 to enable transfer.

Figure 7.10 Example of Single Address Mode Setting Procedure (When Sequential Mode Is Specified)

Table 7.10 summarizes register functions in normal mode.

Table 7.10 Register Functions in Normal Mode

Register	Function	Initial Setting	Operation
	Source address register	Start address of transfer source	Incremented/d every transfer
	Destination address register	Start address of transfer destination	Incremented/d every transfer
	Transfer counter	Number of transfers	Decremented transfer; trans when count re H'0000

Legend:
MARA : Memory address register A
MARB : Memory address register B
ETCRA : Transfer count register A

MARA and MARB specify the start addresses of the transfer source and transfer desti respectively, as 24 bits. MAR can be incremented or decremented by 1 or 2 each time word is transferred, or can be fixed.

Incrementing, decrementing, or holding a fixed value can be set separately for MARA MARB.

The number of transfers is specified by ETCRA as 16 bits. ETCRA is decremented ea transfer is performed, and when its value reaches H'0000 the DTE bit is cleared and tr If the DTIE bit is set to 1 at this time, an interrupt request is sent to the CPU or DTC.

The maximum number of transfers, when H'0000 is set in ETCRA, is 65,536.

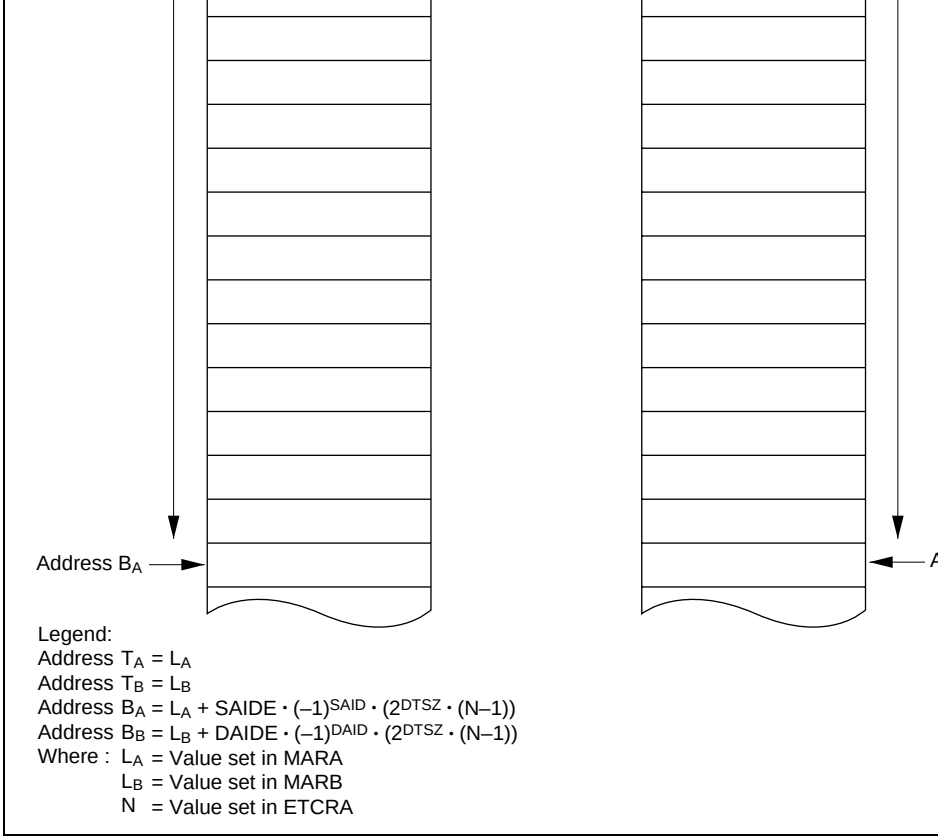
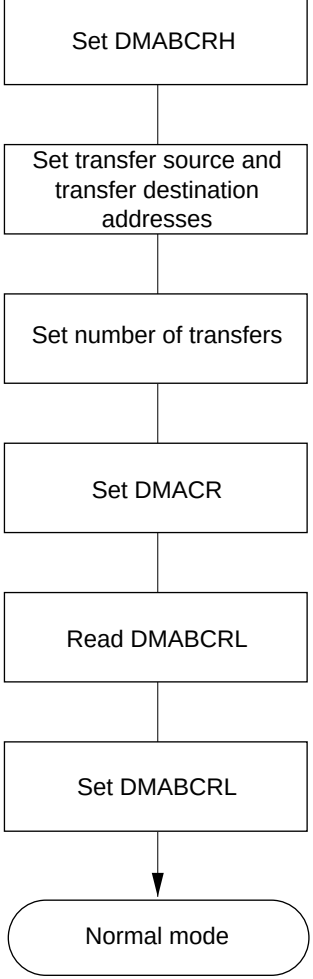


Figure 7.11 Operation in Normal Mode

Transfer requests (activation sources) are external requests and auto-requests.

With auto-request, the DMAC is only activated by register setting, and the specified nu transfers are performed automatically. With auto-request, cycle steal mode or burst mo selected. In cycle steal mode, the bus is released to another bus master each time a tran performed. In burst mode, the bus is held continuously until transfer ends.



[1] Set DMABCRH.
Interrupt clearing with the DTA bit.

[2] Set the transfer source address in MARA
the transfer destination address in MARB

[3] Set the number of transfers in ETCRA.

[4] Set each bit in DMACRA and DMACRB.
- Set the transfer data size with the DTS
- Specify whether MARA is to be increm
decremented, or fixed, with the SAID a
SAIDE bits.
- Clear the BLKE bit to 0 to select norm
mode.
- Specify whether MARB is to be increm
decremented, or fixed, with the DAID a
DAIDE bits.
- Select the activation source with bits D
DTF0.

[5] Read DTE = 0 and DTME = 0 in DMABC

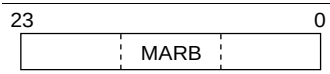
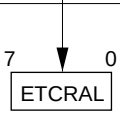
[6] Set each bit in DMABCRL.
- Specify enabling or disabling of transf
interrupts with the DTIE bit.
- Set both the DTME bit and the DTE bi
enable transfer.

Figure 7.12 Example of Normal Mode Setting Procedure

transfer destination can be selected as a block area (an area composed of a number of b
words).

Table 7.11 summarizes register functions in block transfer mode.

Table 7.11 Register Functions in Block Transfer Mode

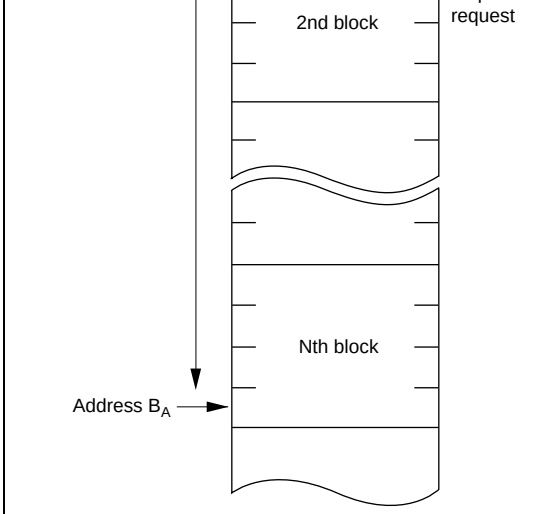
Register	Function	Initial Setting	Operation
	Source address register	Start address of transfer source	Incremented/de every transfer,
	Destination address register	Start address of transfer destination	Incremented/de every transfer,
	Holds block size	Block size	Fixed
	Block size counter	Block size	Decremented e transfer; ETCR copied when co reaches H'00
	Block transfer counter	Number of block transfers	Decremented e transfer; transfe when count rea H'0000

Legend:

MARA : Memory address register A
MARB : Memory address register B
ETCRA : Transfer count register A
ETCRB : Transfer count register B

To specify the number of transfers, if M is the size of one block (where M = 1 to 256) transfers are to be performed (where N = 1 to 65,536), M is set in both ETCRAH and and N in ETCRB.

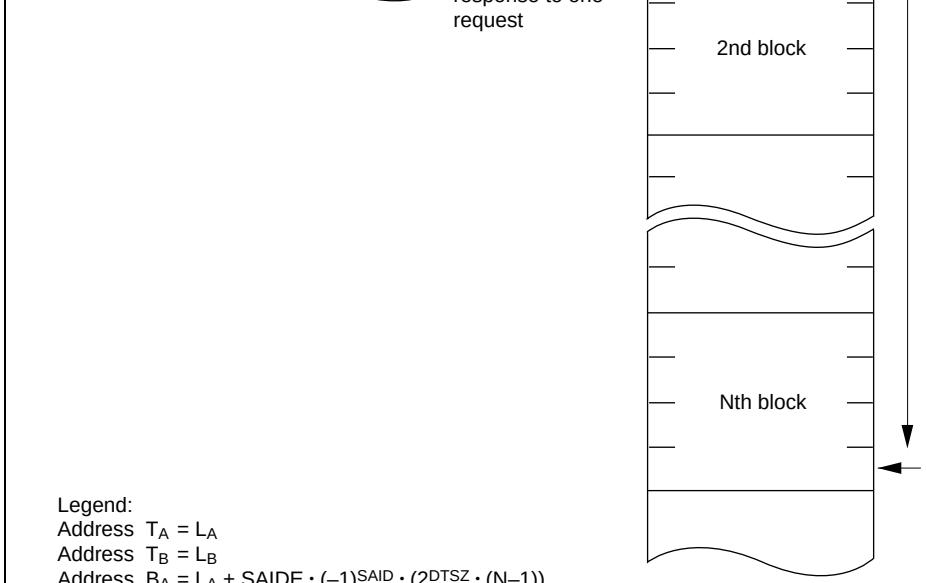
Figure 7.13 illustrates operation in block transfer mode when MARB is designated as



Legend:
Address $T_A = L_A$
Address $T_B = L_B$
Address $B_A = L_A + SAIDE \cdot (-1)^{SAID} \cdot (2^{DTSZ} \cdot (M \cdot N - 1))$
Address $B_B = L_B + DAIDE \cdot (-1)^{DAID} \cdot (2^{DTSZ} \cdot (N - 1))$
Where : L_A = Value set in MARA
L_B = Value set in MARB
N = Value set in ETCRB
M = Value set in ETCRAH and ETCRAL

Figure 7.13 Operation in Block Transfer Mode (BLKDIR = 0)

Figure 7.14 illustrates operation in block transfer mode when MARA is designated as a



Legend:
Address $T_A = L_A$
Address $T_B = L_B$
Address $B_A = L_A + SAIDE \cdot (-1)^{SAID} \cdot (2^{DTSZ} \cdot (N-1))$
Address $B_B = L_B + DAIDE \cdot (-1)^{DAID} \cdot (2^{DTSZ} \cdot (M \cdot N-1))$
Where : L_A = Value set in MARA
L_B = Value set in MARB
N = Value set in ETCRB
M = Value set in ETCRAH and ETCRAL

Figure 7.14 Operation in Block Transfer Mode (BLKDIR = 1)

ETCRAL is decremented by 1 each time a byte or word transfer is performed. In respo
single transfer request, burst transfer is performed until the value in ETCRAL reaches
ETCRAL is then loaded with the value in ETCRAH. At this time, the value in the MA
for which a block designation has been given by the BLKDIR bit in DMACRA is rest
accordance with the DTSZ, SAID/DAID, and SAIDE/DAIDE bits in DMACR.

ETCRB is decremented by 1 every block transfer, and when the count reaches H'0000
is cleared and transfer ends. If the DTIE bit is set to 1 at this point, an interrupt reques
the CPU or DTC.

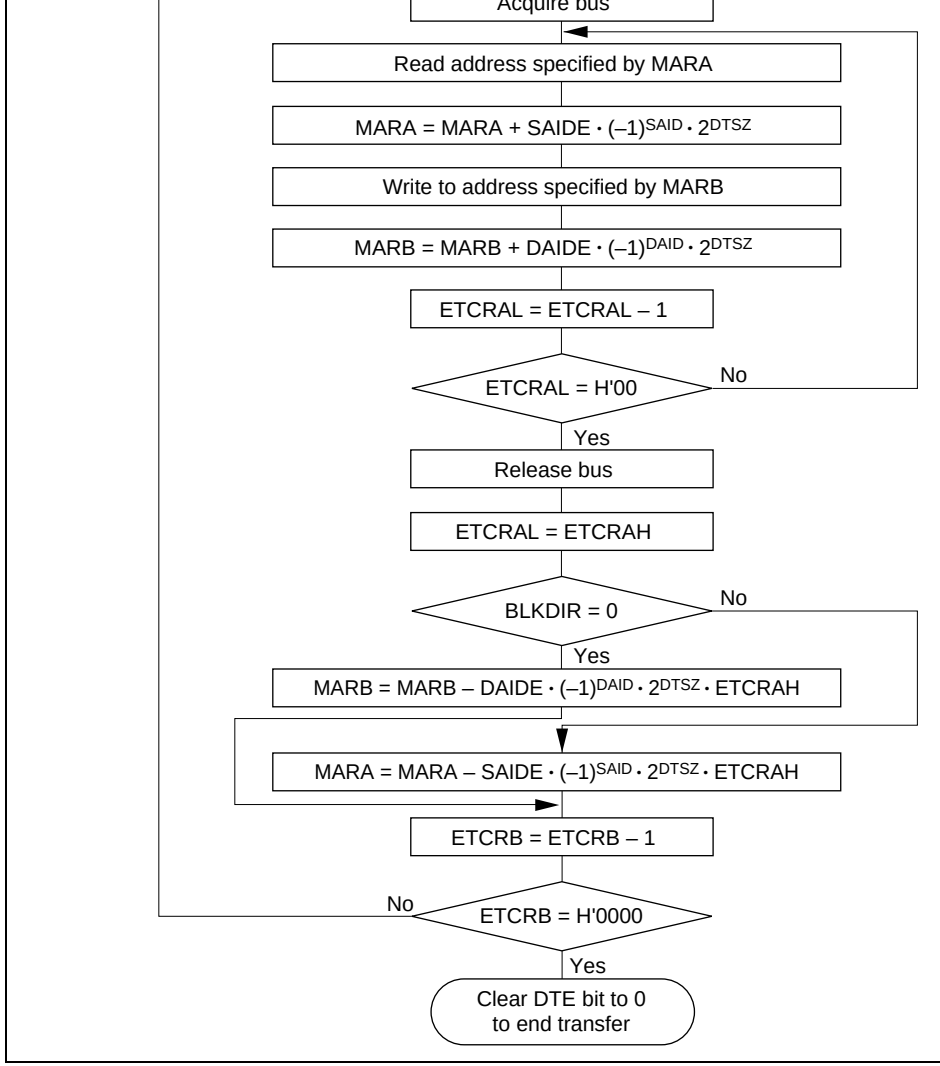
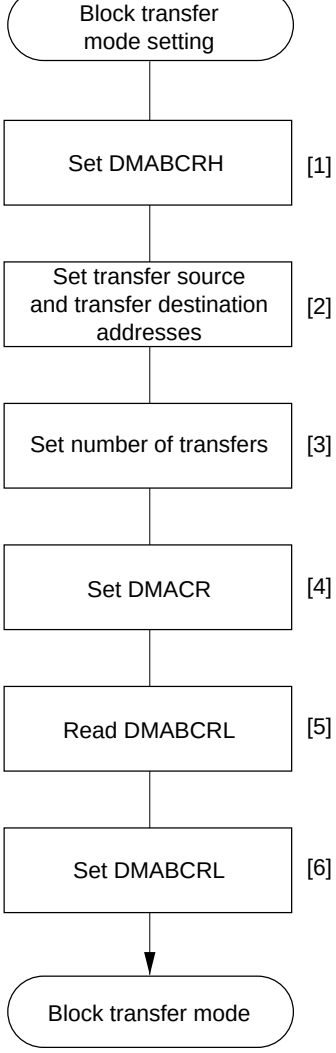


Figure 7.15 Operation Flow in Block Transfer Mode



[1] Set each bit in DMABCRH.
- Set the FAE bit to 1 to select full addres mode.
- Specify enabling or disabling of internal interrupt clearing with the DTA bit.

[2] Set the transfer source address in MARA, the transfer destination address in MARB.

[3] Set the block size in both ETCRAH and ETCRAL. Set the number of transfers in ETCRB.

[4] Set each bit in DMACRA and DMACRB.
- Set the transfer data size with the DTSZ
- Specify whether MARA is to be increme decremented, or fixed, with the SAID ar SAIDE bits.
- Set the BLKE bit to 1 to select block tra mode.
- Specify whether the transfer source or t transfer destination is a block area with BLKDIR bit.
- Specify whether MARB is to be increme decremented, or fixed, with the DAID ar DAIDE bits.
- Select the activation source with bits DT DTF0.

[5] Read DTE = 0 and DTME = 0 in DMABCR

[6] Set each bit in DMABCRL.
- Specify enabling or disabling of transfer interrupts to the CPU with the DTIE bit.
- Set both the DTME bit and the DTE bit t enable transfer.

Figure 7.16 Example of Block Transfer Mode Setting Procedure

Activation Source		Channels 0A and 1A	Channels 0B and 1B	Normal Mode	Bl Tr M
Internal Interrupts	ADI	○	○	X	○
	TXI0	○	○	X	○
	RXI0	○	○	X	○
	TXI1	○	○	X	○
	RXI1	○	○	X	○
	TGI0A	○	○	X	○
	TGI1A	○	○	X	○
	TGI2A	○	○	X	○
	TGI3A	○	○	X	○
	TGI4A	○	○	X	○
	TGI5A	○	○	X	○
External Requests	$\overline{\text{DREQ}}$ pin falling edge input	X	○	○	○
	$\overline{\text{DREQ}}$ pin low-level input	X	○	○	○
Auto-request		X	X	○	X

Legend:

○ : Can be specified

X : Cannot be specified

DTC activation source (DTA = 1), the interrupt source flag is cleared automatically by transfer. With ADI, TXI, and RXI interrupts, however, the interrupt source flag is not unless the prescribed register is accessed in a DMA transfer. If the same interrupt is us activation source for more than one channel, the interrupt request flag is cleared when priority channel is activated first. Transfer requests for other channels are held pendin DMAC, and activation is carried out in order of priority.

When DTE = 0, such as after completion of a transfer, a request from the selected acti source is not sent to the DMAC, regardless of the DTA bit. In this case, the relevant in request is sent to the CPU or DTC.

In case of overlap with a CPU interrupt source or DTC activation source (DTA = 0), t request flag is not cleared by the DMAC.

Activation by External Request

If an external request ($\overline{\text{DREQ}}$ pin) is specified as an activation source, the relevant por set to input mode in advance.

Level sensing or edge sensing can be used for external requests.

External request operation in normal mode (short address mode or full address mode) below.

When edge sensing is selected, a 1-byte or 1-word transfer is executed each time a hig transition is detected on the $\overline{\text{DREQ}}$ pin. The next transfer may not be performed if the input before transfer is completed.

When level sensing is selected, the DMAC stands by for a transfer request while the $\overline{\text{D}}$ held high. While the $\overline{\text{DREQ}}$ pin is held low, transfers continue in succession, with the released each time a byte or word is transferred. If the $\overline{\text{DREQ}}$ pin goes high in the mid transfer, the transfer is interrupted and the DMAC stands by for a transfer request.

In burst mode, the DMAC keeps possession of the bus until the end of the transfer, and
performed continuously.

Single Address Mode

The DMAC can operate in dual address mode in which read cycles and write cycles are
cycles, or single address mode in which read and write cycles are executed in parallel.

In dual address mode, transfer is performed with the source address and destination add
specified separately.

In single address mode, on the other hand, transfer is performed between external space
either the transfer source or the transfer destination is specified by an address, and an e
device for which selection is performed by means of the $\overline{\text{DACK}}$ strobe, without regard
address. Figure 7.17 shows the data bus in single address mode.

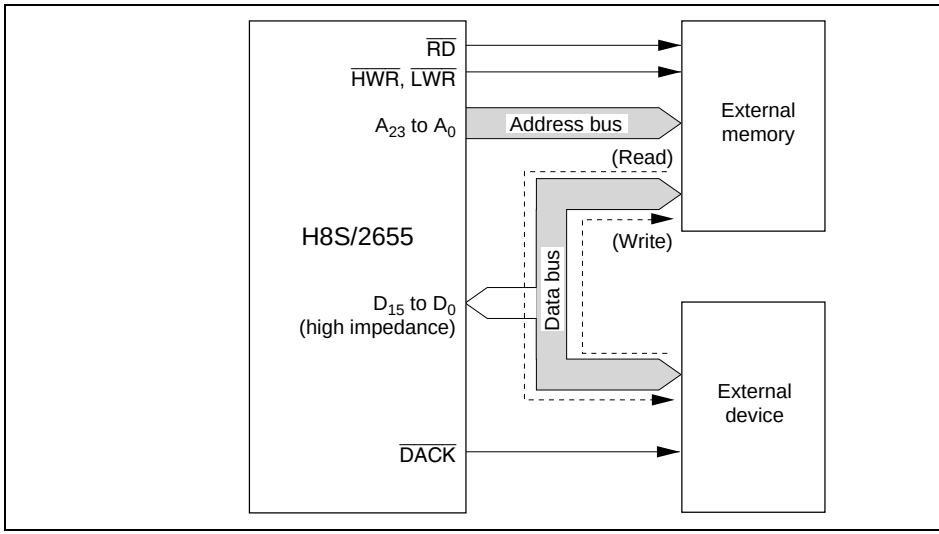


Figure 7.17 Data Bus in Single Address Mode

external memory area. On the external device side, DACK is output in synchronization
address strobe. For details of bus cycles, see section 7.5.11, DMAC Bus Cycles (Singl
Mode).

Do not specify internal space for transfer addresses in single address mode.

7.5.9 Basic DMAC Bus Cycles

An example of the basic DMAC bus cycle timing is shown in figure 7.18. In this exam
size transfer is performed from 16-bit, 2-state access space to 8-bit, 3-state access spac
bus is transferred from the CPU to the DMAC, a source address read and destination a
are performed. The bus is not released in response to another bus request, etc., betwee
and write operations. As with CPU cycles, DMA cycles conform to the bus controller

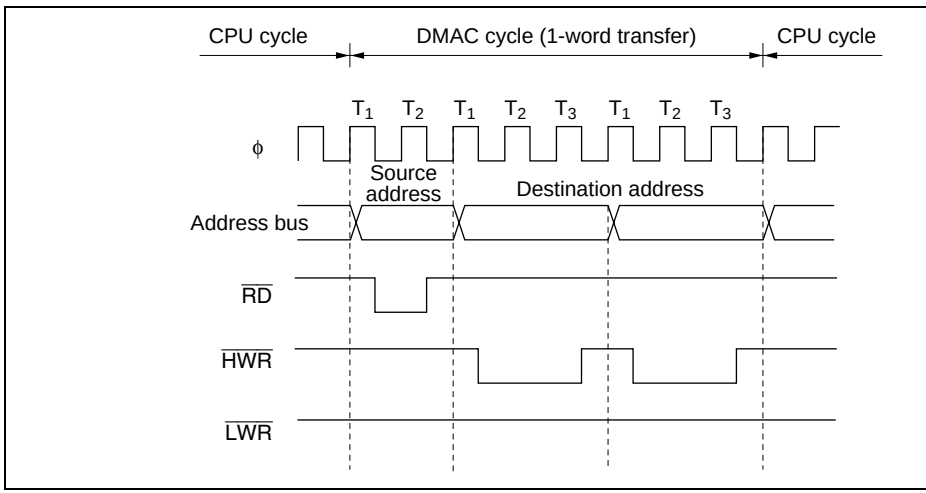


Figure 7.18 Example of DMA Transfer Bus Timing

The address is not output to the external address bus in an access to on-chip memory o
I/O register.

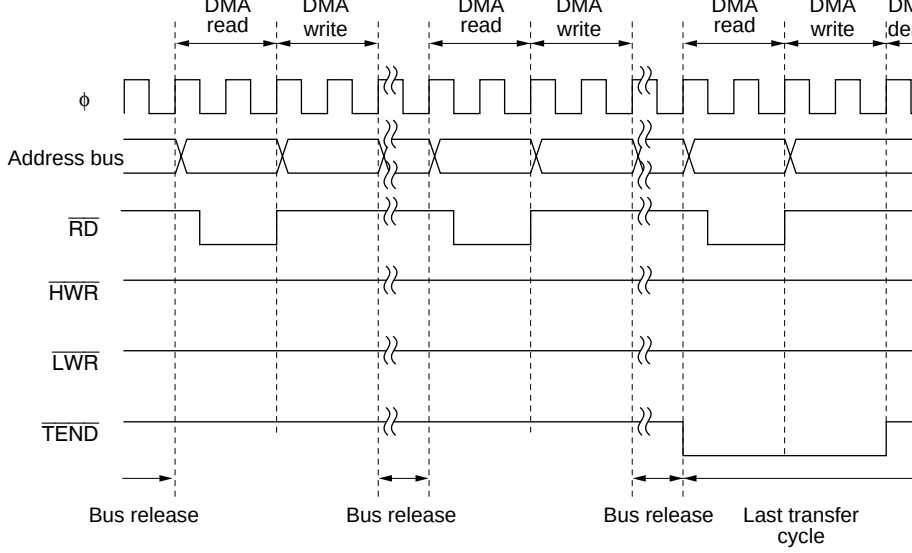


Figure 7.19 Example of Short Address Mode Transfer

A 1-byte or 1-word transfer is performed for one transfer request, and after the transfer released. While the bus is released one or more bus cycles are inserted by the CPU or D

In the transfer end cycle (the cycle in which the transfer counter reaches 0), a one-state cycle is inserted after the DMA write cycle.

In repeat mode, when $\overline{\text{TEND}}$ output is enabled, $\overline{\text{TEND}}$ output goes low in the transfer c which the transfer counter reaches 0.

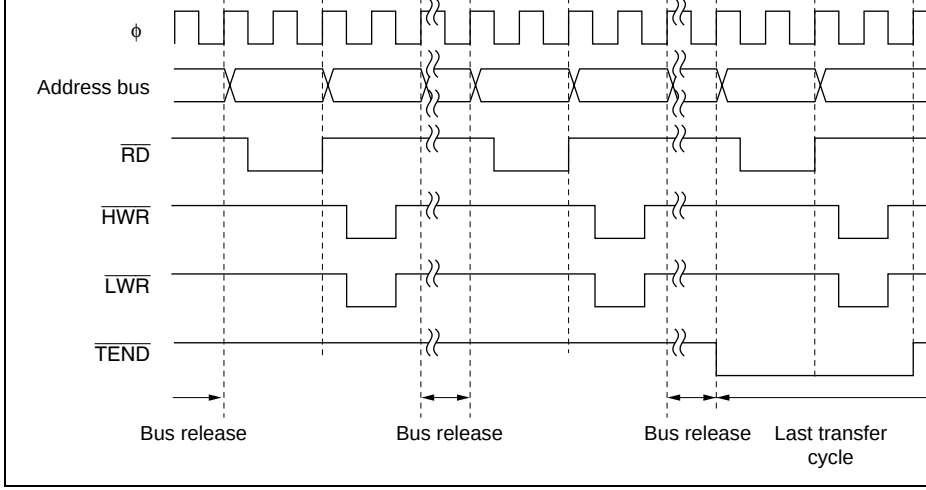


Figure 7.20 Example of Full Address Mode (Cycle Steal) Transfer

A one-byte or one-word transfer is performed, and after the transfer the bus is released bus is released one bus cycle is inserted by the CPU or DTC.

In the transfer end cycle (the cycle in which the transfer counter reaches 0), a one-state cycle is inserted after the DMA write cycle.

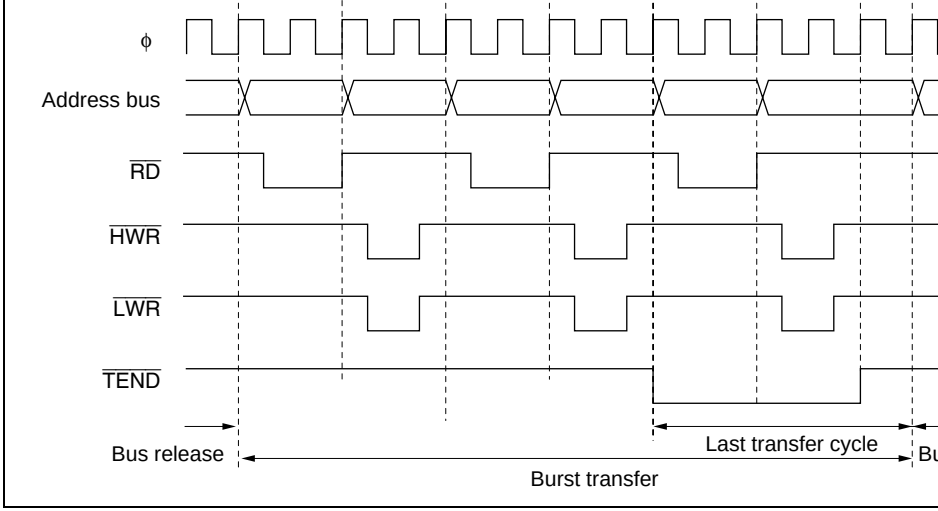


Figure 7.21 Example of Full Address Mode (Burst Mode) Transfer

In burst mode, one-byte or one-word transfers are executed consecutively until transfer

In the transfer end cycle (the cycle in which the transfer counter reaches 0), a one-state cycle is inserted after the DMA write cycle.

If a request from another higher-priority channel is generated after burst transfer starts, channel has to wait until the burst transfer ends.

If an NMI is generated while a channel designated for burst transfer is in the transfer en the DTME bit is cleared and the channel is placed in the transfer disabled state. If burst has already been activated inside the DMAC, the bus is released on completion of a one one-word transfer within the burst transfer, and burst transfer is suspended. If the last tr cycle of the burst transfer has already been activated inside the DMAC, execution conti end of the transfer even if the DTME bit is cleared.

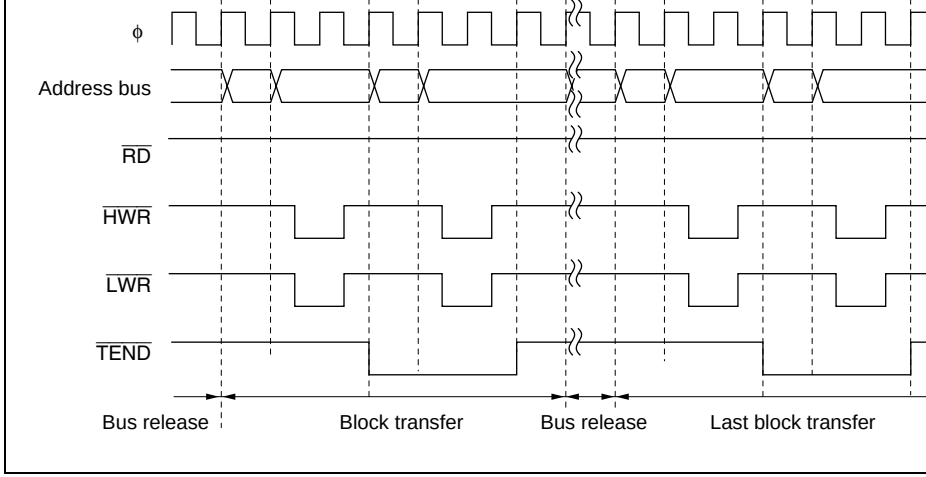
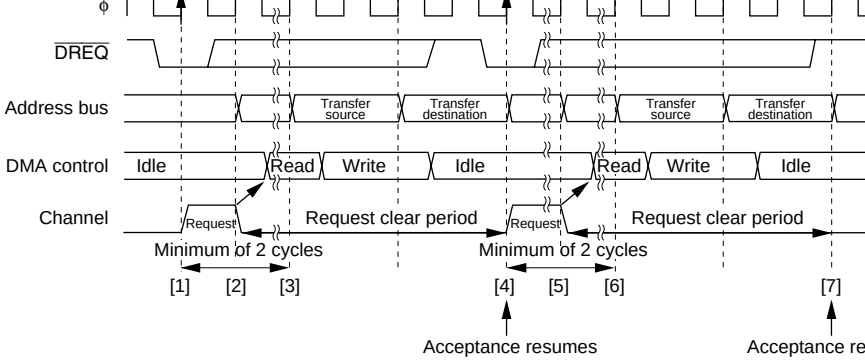


Figure 7.22 Example of Full Address Mode (Block Transfer Mode) Tran

A one-block transfer is performed for one transfer request, and after the transfer the bu released. While the bus is released, one or more bus cycles are inserted by the CPU or

In the transfer end cycle of each block (the cycle in which the transfer counter reaches state DMA dead cycle is inserted after the DMA write cycle.

One block is transmitted without interruption. NMI generation does not affect block tr operation.



[1] Acceptance after transfer enabling; the DREQ pin low level is sampled on the risi
edge of φ, and the request is held.

[2] [5] The request is cleared at the next bus break, and activation is started in the DMA

[3] [6] Start of DMA cycle; DREQ pin high level sampling on the rising edge of φ starts.

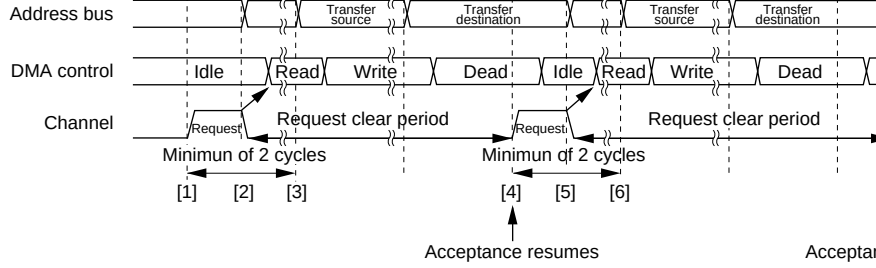
[4] [7] When the DREQ pin high level has been sampled, acceptance is resumed after t
write cycle is completed.
(As in [1], the DREQ pin low level is sampled on the rising edge of φ, and the requ
is held.)

Note: In write data buffer mode, bus breaks from [2] to [7] may be hidden, and not visibl

Figure 7.23 Example of DREQ Pin Falling Edge Activated Normal Mode Tr

DREQ pin sampling is performed every cycle, with the rising edge of the next φ cycle a
end of the DMABCR write cycle for setting the transfer enabled state as the starting po

When the DREQ pin low level is sampled while acceptance by means of the DREQ pin
possible, the request is held in the DMAC. Then, when activation is initiated in the DM
request is cleared, and DREQ pin high level sampling for edge detection is started. If D
high level sampling has been completed by the time the DMA write cycle ends, accepta
resumes after the end of the write cycle, DREQ pin low level sampling is performed ag
this operation is repeated until the transfer ends.



[1] Acceptance after transfer enabling; the $\overline{\text{DREQ}}$ pin low level is sampled on the rising ed and the request is held.

[2] [5] The request is cleared at the next bus break, and activation is started in the DMAC.

[3] [6] Start of DMA cycle; $\overline{\text{DREQ}}$ pin high level sampling on the rising edge of ϕ starts.

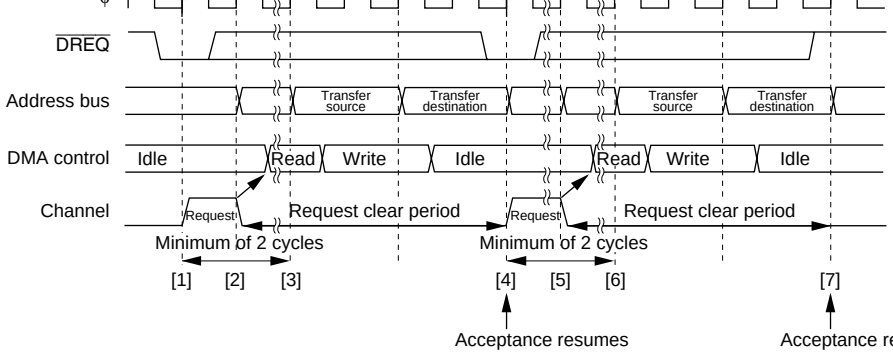
[4] [7] When the $\overline{\text{DREQ}}$ pin high level has been sampled, acceptance is resumed after the de is completed.
(As in [1], the $\overline{\text{DREQ}}$ pin low level is sampled on the rising edge of ϕ, and the request is

Note: In write data buffer mode, bus breaks from [2] to [7] may be hidden, and not visible.

Figure 7.24 Example of $\overline{\text{DREQ}}$ Pin Falling Edge Activated Block Transfer Mod

$\overline{\text{DREQ}}$ pin sampling is performed every cycle, with the rising edge of the next ϕ cycle end of the DMABCR write cycle for setting the transfer enabled state as the starting p

When the $\overline{\text{DREQ}}$ pin low level is sampled while acceptance by means of the $\overline{\text{DREQ}}$ pi possible, the request is held in the DMAC. Then, when activation is initiated in the DM request is cleared, and $\overline{\text{DREQ}}$ pin high level sampling for edge detection is started. If high level sampling has been completed by the time the DMA dead cycle ends, accept resumes after the end of the dead cycle, $\overline{\text{DREQ}}$ pin low level sampling is performed ag operation is repeated until the transfer ends.



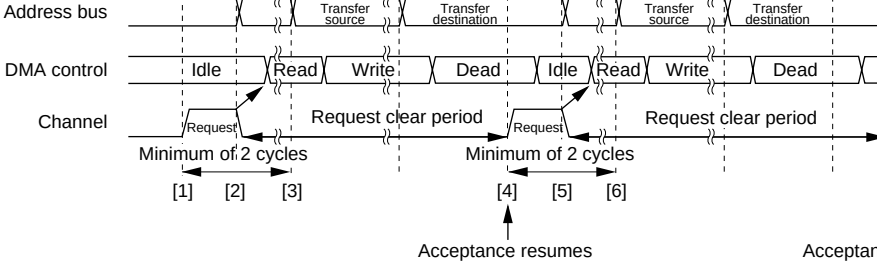
[1] Acceptance after transfer enabling; the $\overline{\text{DREQ}}$ pin low level is sampled on the rising edge of φ, and the request is held.
[2] [5] The request is cleared at the next bus break, and activation is started in the DMAC.
[3] [6] The DMA cycle is started.
[4] [7] Acceptance is resumed after the write cycle is completed.
(As in [1], the $\overline{\text{DREQ}}$ pin low level is sampled on the rising edge of φ, and the request

Note: In write data buffer mode, bus breaks from [2] to [7] may be hidden, and not visible.

Figure 7.25 Example of $\overline{\text{DREQ}}$ Level Activated Normal Mode Transfer

$\overline{\text{DREQ}}$ pin sampling is performed every cycle, with the rising edge of the next φ cycle a end of the DMABCR write cycle for setting the transfer enabled state as the starting po

When the $\overline{\text{DREQ}}$ pin low level is sampled while acceptance by means of the $\overline{\text{DREQ}}$ pin possible, the request is held in the DMAC. Then, when activation is initiated in the DM request is cleared. After the end of the write cycle, acceptance resumes, $\overline{\text{DREQ}}$ pin low sampling is performed again, and this operation is repeated until the transfer ends.



[1] Acceptance after transfer enabling; the $\overline{\text{DREQ}}$ pin low level is sampled on the rising edge of ϕ, and the request is held.

[2] [5] The request is cleared at the next bus break, and activation is started in the DMAC.

[3] [6] The DMA cycle is started.

[4] [7] Acceptance is resumed after the dead cycle is completed.
(As in [1], the $\overline{\text{DREQ}}$ pin low level is sampled on the rising edge of ϕ, and the request i

Note: In write data buffer mode, bus breaks from [2] to [7] may be hidden, and not visible.

Figure 7.26 Example of $\overline{\text{DREQ}}$ Level Activated Block Transfer Mode Tra

$\overline{\text{DREQ}}$ pin sampling is performed every cycle, with the rising edge of the next ϕ cycle end of the DMABCR write cycle for setting the transfer enabled state as the starting p

When the $\overline{\text{DREQ}}$ pin low level is sampled while acceptance by means of the $\overline{\text{DREQ}}$ pi possible, the request is held in the DMAC. Then, when activation is initiated in the DM request is cleared. After the end of the dead cycle, acceptance resumes, $\overline{\text{DREQ}}$ pin low sampling is performed again, and this operation is repeated until the transfer ends.

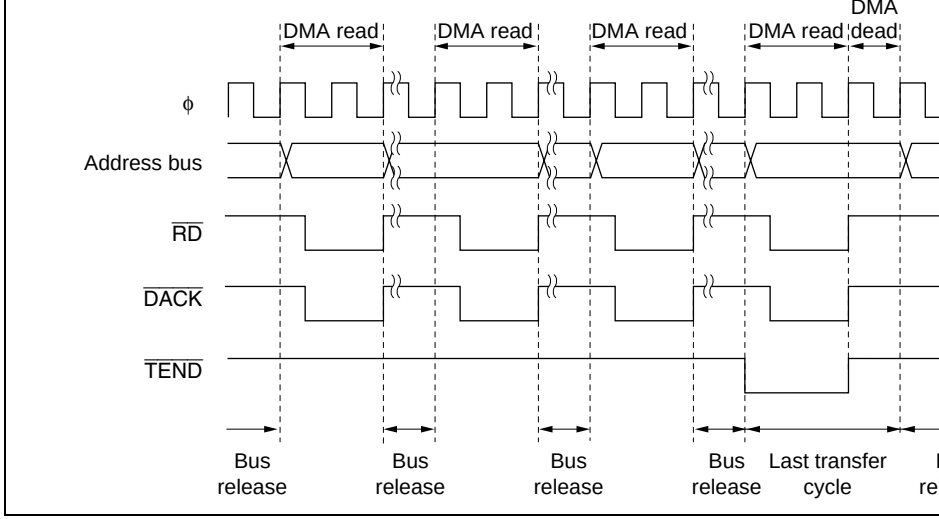


Figure 7.27 Example of Single Address Mode (Byte Read) Transfer

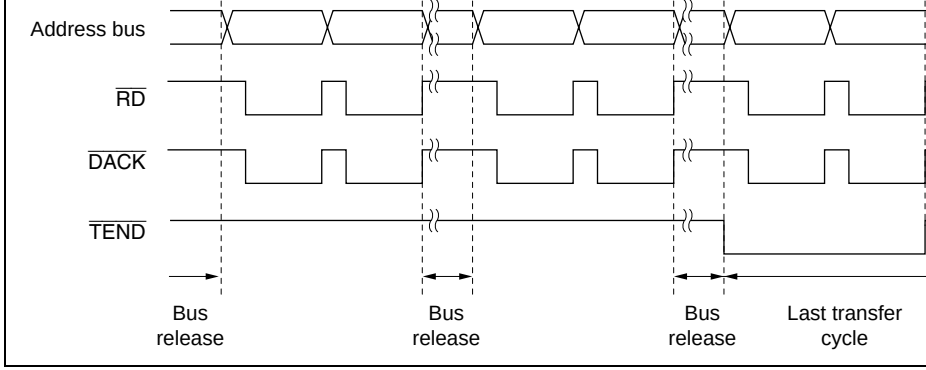


Figure 7.28 Example of Single Address Mode (Word Read) Transfer

A one-byte or one-word transfer is performed for one transfer request, and after the tra bus is released. While the bus is released, one or more bus cycles are inserted by the C

In the transfer end cycle (the cycle in which the transfer counter reaches 0), a one-state cycle is inserted after the DMA write cycle.

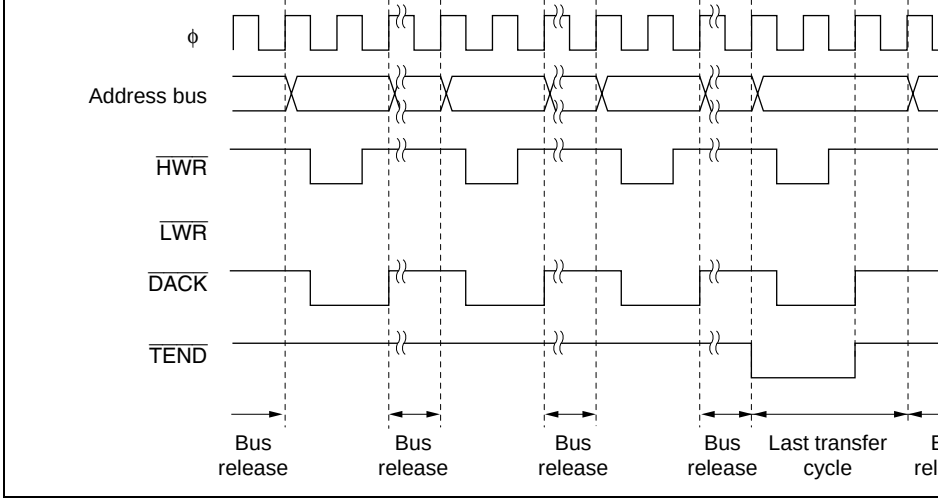


Figure 7.29 Example of Single Address Mode (Byte Write) Transfer

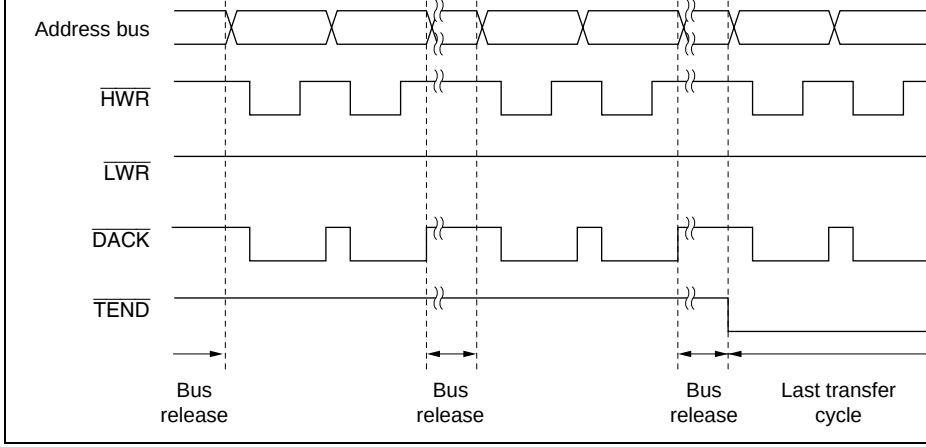
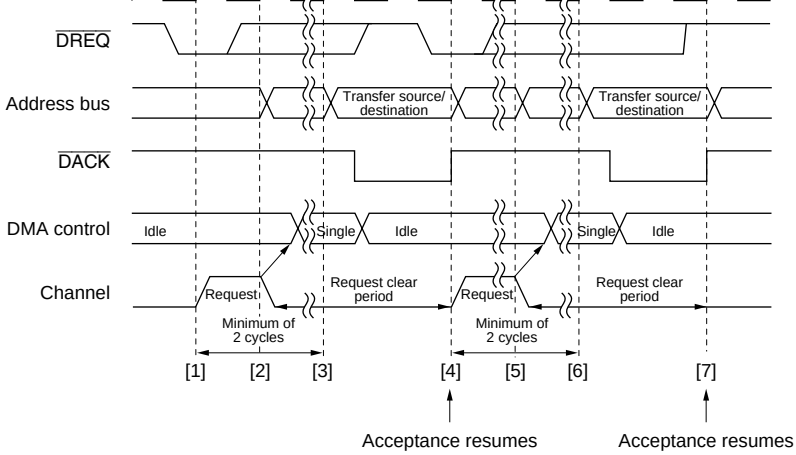


Figure 7.30 Example of Single Address Mode (Word Write) Transfe

A one-byte or one-word transfer is performed for one transfer request, and after the tra bus is released. While the bus is released one or more bus cycles are inserted by the C

In the transfer end cycle (the cycle in which the transfer counter reaches 0), a one-stat cycle is inserted after the DMA write cycle.



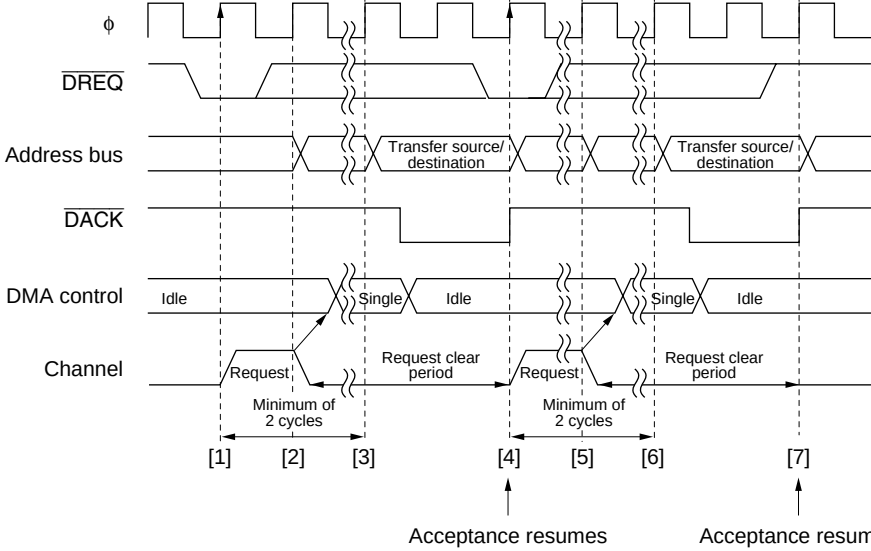
[1] Acceptance after transfer enabling; the $\overline{\text{DREQ}}$ pin low level is sampled on the rising edge of ϕ, and the request is held.
[2] [5] The request is cleared at the next bus break, and activation is started in the DMAC.
[3] [6] Start of DMA cycle; $\overline{\text{DREQ}}$ pin high level sampling on the rising edge of ϕ starts.
[4] [7] When the $\overline{\text{DREQ}}$ pin high level has been sampled, acceptance is resumed after the single cycle is completed.
(As in [1], the $\overline{\text{DREQ}}$ pin low level is sampled on the rising edge of ϕ, and the request is held

Note: In write data buffer mode, bus breaks from [2] to [7] may be hidden, and not visible.

Figure 7.31 Example of $\overline{\text{DREQ}}$ Pin Falling Edge Activated Single Address Mode

$\overline{\text{DREQ}}$ pin sampling is performed every cycle, with the rising edge of the next ϕ cycle a end of the DMABCR write cycle for setting the transfer enabled state as the starting po

When the $\overline{\text{DREQ}}$ pin low level is sampled while acceptance by means of the $\overline{\text{DREQ}}$ pin possible, the request is held in the DMAC. Then, when activation is initiated in the DM request is cleared, and $\overline{\text{DREQ}}$ pin high level sampling for edge detection is started. If $\overline{\text{D}}$ high level sampling has been completed by the time the DMA single cycle ends, accept resumes after the end of the single cycle, $\overline{\text{DREQ}}$ pin low level sampling is performed ag this operation is repeated until the transfer ends.



[1] Acceptance after transfer enabling; the $\overline{\text{DREQ}}$ pin low level is sampled on the rising edge of φ, and the request is held.
[2] [5] The request is cleared at the next bus break, and activation is started in the DMAC.
[3] [6] The DMAC cycle is started.
[4] [7] Acceptance is resumed after the single cycle is completed.
(As in [1], the $\overline{\text{DREQ}}$ pin low level is sampled on the rising edge of φ, and the reques

Note: In write data buffer mode, bus breaks from [2] to [7] may be hidden, and not visible.

Figure 7.32 Example of $\overline{\text{DREQ}}$ Pin Low Level Activated Single Address Mode

$\overline{\text{DREQ}}$ pin sampling is performed every cycle, with the rising edge of the next φ cycle end of the DMABCR write cycle for setting the transfer enabled state as the starting p

When the $\overline{\text{DREQ}}$ pin low level is sampled while acceptance by means of the $\overline{\text{DREQ}}$ pi possible, the request is held in the DMAC. Then, when activation is initiated in the DM

When the WDBE bit of BCRL in the bus controller is set to 1, enabling the write data b
function, dual address transfer external write cycles or single address transfers and inter
accesses (on-chip memory or internal I/O registers) are executed in parallel. Internal ac
independent of the bus master, and DMAC dead cycles are regarded as internal accesse

A low level can always be output from the $\overline{\text{TEND}}$ pin if the bus cycle in which a low le
output is an external bus cycle. However, a low level is not output from the $\overline{\text{TEND}}$ pin
cycle in which a low level is to be output from the $\overline{\text{TEND}}$ pin is an internal bus cycle, a
external write cycle is executed in parallel with this cycle.

Figure 7.33 shows an example of burst mode transfer from on-chip RAM to external m
using the write data buffer function.

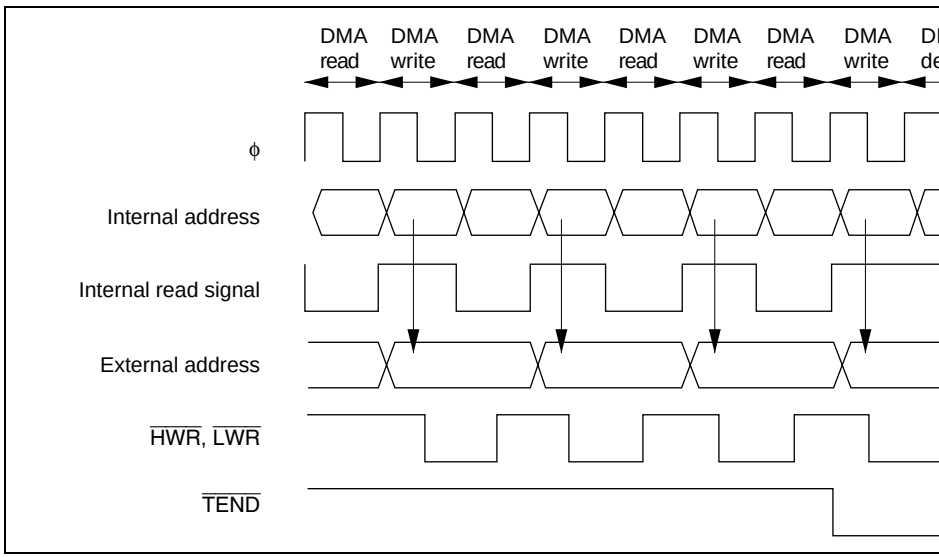


Figure 7.33 Example of Dual Address Transfer Using Write Data Buffer Fu

Figure 7.34 shows an example of single address transfer using the write data buffer fun
this example, the CPU program area is in on-chip memory.

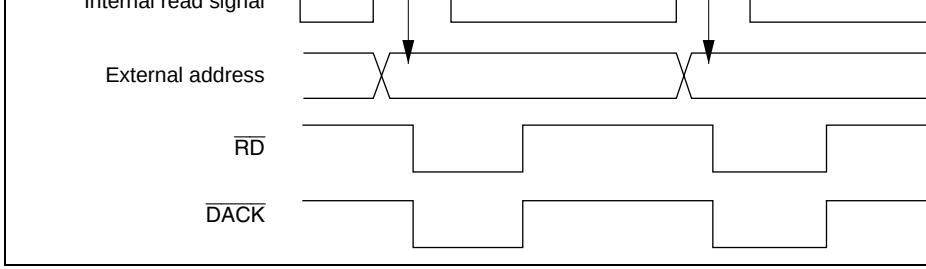


Figure 7.34 Example of Single Address Transfer Using Write Data Buffer F

When the write data buffer function is activated, the DMAC recognizes that the bus cy concerned has ended, and starts the next operation. Therefore, $\overline{\text{DREQ}}$ pin sampling is state after the start of the DMA write cycle or single address transfer.

7.5.13 DMAC Multi-Channel Operation

The DMAC channel priority order is: channel 0 > channel 1, and channel A > channel 7.13 summarizes the priority order for DMAC channels.

Table 7.13 DMAC Channel Priority Order

Short Address Mode	Full Address Mode	Priority
Channel 0A	Channel 0	High
Channel 0B		↑
Channel 1A	Channel 1	↓
Channel 1B		Low

If transfer requests are issued simultaneously for more than one channel, or if a transfe another channel is issued during a transfer, when the bus is released the DMAC select highest-priority channel from among those issuing a request according to the priority o in table 7.13.

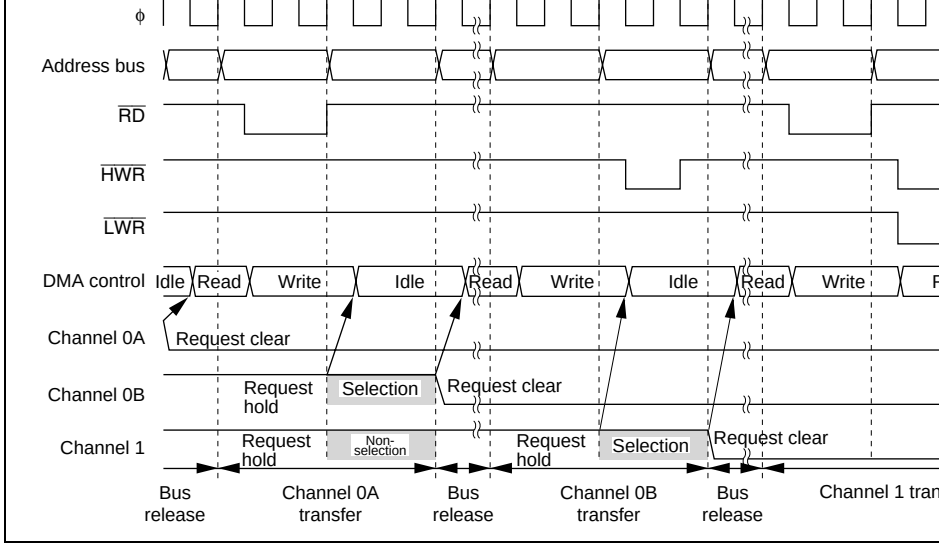


Figure 7.35 Example of Multi-Channel Transfer

7.5.14 Relation between External Bus Requests, Refresh Cycles, the DTC, and t

There can be no break between a DMA cycle read and a DMA cycle write. This means refresh cycle, external bus release cycle, or DTC cycle is not generated between the ext and external write in a DMA cycle.

In the case of successive read and write cycles, such as in burst transfer or block transfe or external bus released state may be inserted after a write cycle. Since the DTC has a l priority than the DMAC, the DTC does not operate until the DMAC releases the bus.

When DMA cycle reads or writes are accesses to on-chip memory or internal I/O regist DMA cycles may be executed at the same time as refresh cycles or external bus release

completion of the 1-byte or 1-word transfer in progress, then releases the bus, which p
CPU.

The channel on which transfer was interrupted can be restarted by setting the DTME b
Figure 7.36 shows the procedure for continuing transfer when it has been interrupted b
interrupt on a channel designated for burst mode transfer.

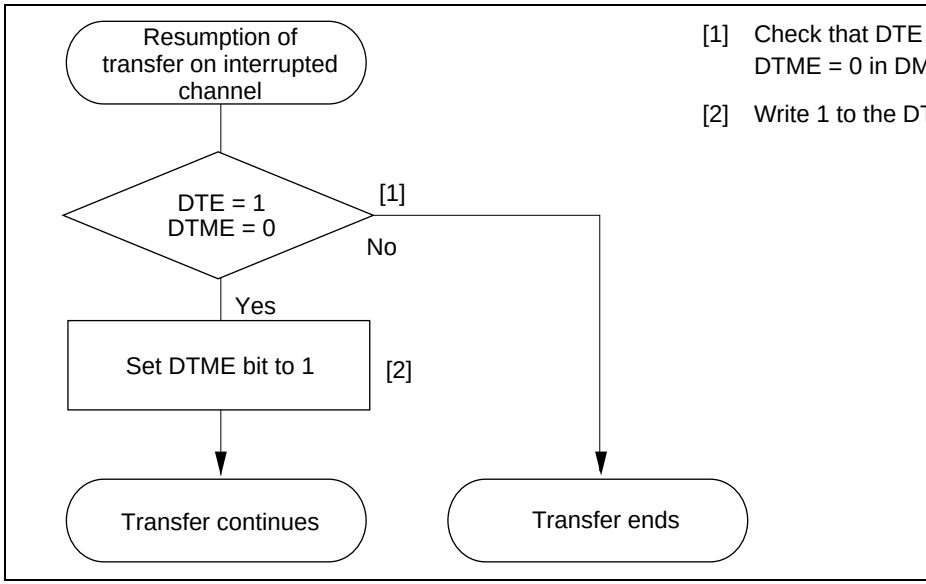
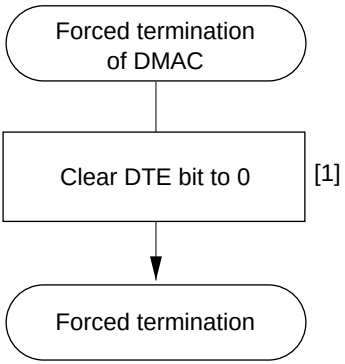
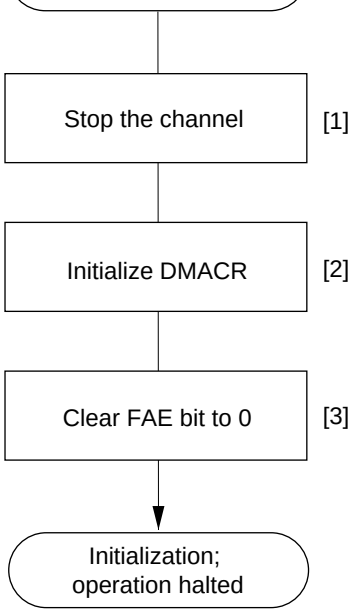


**Figure 7.36 Example of Procedure for Continuing Transfer on Channel Inter
NMI Interrupt**



[1] Clear the DTE bit in DMABCRL to 0.
If you want to prevent interrupt generatio
forced termination of DMAC operation, cl
DTIE bit to 0 at the same time.

Figure 7.37 Example of Procedure for Forcibly Terminating DMAC Opera



and the DTE bit is cleared to 0, then clea
DTME bit to 0.
Also clear the corresponding DTIE bit to 0
same time.

[2] Clear all bits in DMACRA and DMACRB t

[3] Clear the FAE bit in DMABCRH to 0.

Figure 7.38 Example of Procedure for Clearing Full Address Mode

DEND0A	Interrupt due to end of transfer on channel 0A	Interrupt due to end of transfer on channel 0	High
DEND0B	Interrupt due to end of transfer on channel 0B	Interrupt due to break in transfer on channel 0	
DEND1A	Interrupt due to end of transfer on channel 1A	Interrupt due to end of transfer on channel 1	
DEND1B	Interrupt due to end of transfer on channel 1B	Interrupt due to break in transfer on channel 1	Low

Enabling or disabling of each interrupt source is set by means of the DTIE bit for the corresponding channel in DMABCR, and interrupts from each source are sent to the int controller independently.

The relative priority of transfer end interrupts on each channel is decided by the interru controller, as shown in table 7.13.

Figure 7.39 shows a block diagram of a transfer end/transfer break interrupt. An interru always generated when the DTIE bit is set to 1 while DTE bit is cleared to 0.

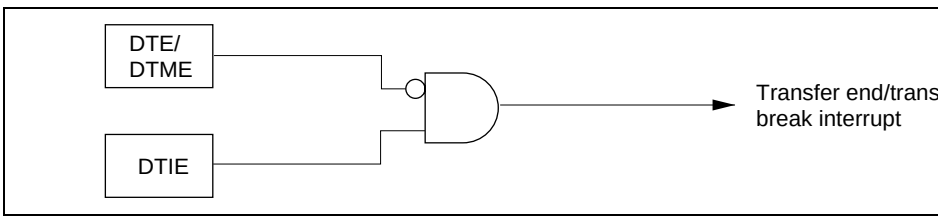


Figure 7.39 Block Diagram of Transfer End/Transfer Break Interrupt

In full address mode, a transfer break interrupt is generated when the DTME bit is clear while DTIEB bit is set to 1.

In both short address mode and full address mode, DMABCR should be set so as to pre occurrence of a combination that constitutes a condition for interrupt generation during

Module Stop

When the MSTP15 bit in MSTPCR is set to 1, the DMAC clock stops, and the module entered. However, 1 cannot be written to the MSTP15 bit if any of the DMAC channe This setting should therefore be made when DMAC operation is stopped.

When the DMAC clock stops, DMAC register accesses can no longer be made. Since following DMAC register settings are valid even in the module stop state, they should invalidated, if necessary, before a module stop.

- Transfer end/suspend interrupt (DTE = 0 and DTIE = 1)
- TEND pin enable (TEE = 1)
- $\overline{\text{DACK}}$ pin enable (FAE = 0 and SAE = 1)

Medium-Speed Mode

When the DTA bit is 0, internal interrupt signals specified as DMAC transfer sources detected.

In medium-speed mode, the DMAC operates on a medium-speed clock, while on-chip modules operate on a high-speed clock. Consequently, if the period in which the relev source is cleared by the CPU, DTC, or another DMAC channel, and the next interrupt generated, is less than one state with respect to the DMAC clock (bus master clock), e detection may not be possible and the interrupt may be ignored.

Also, in medium-speed mode, $\overline{\text{DREQ}}$ pin sampling is performed on the rising edge of speed clock.

external accesses should only be manipulated when external reads, etc., are used with D
operation disabled, and the operation is not performed in parallel with external access.

(b) Write Data Buffer Function and DMAC Operation Timing

The DMAC can start its next operation during external access using the write data buff
Consequently, the $\overline{\text{DREQ}}$ pin sampling timing, $\overline{\text{TEND}}$ output timing, etc., are different
case in which the write data buffer function is disabled. Also, internal bus cycles maybe
and not visible.

(c) Write Data Buffer Function and $\overline{\text{TEND}}$ Output

A low level is not output from the $\overline{\text{TEND}}$ pin if the bus cycle in which a low level is to
from the $\overline{\text{TEND}}$ pin is an internal bus cycle, and an external write cycle is executed in p
this cycle. Note, for example, that a low level may not be output from the $\overline{\text{TEND}}$ pin if
data buffer function is used when data transfer is performed between an internal I/O reg
on-chip memory.

If at least one of the DMAC transfer addresses is an external address, a low level is out
the $\overline{\text{TEND}}$ pin.

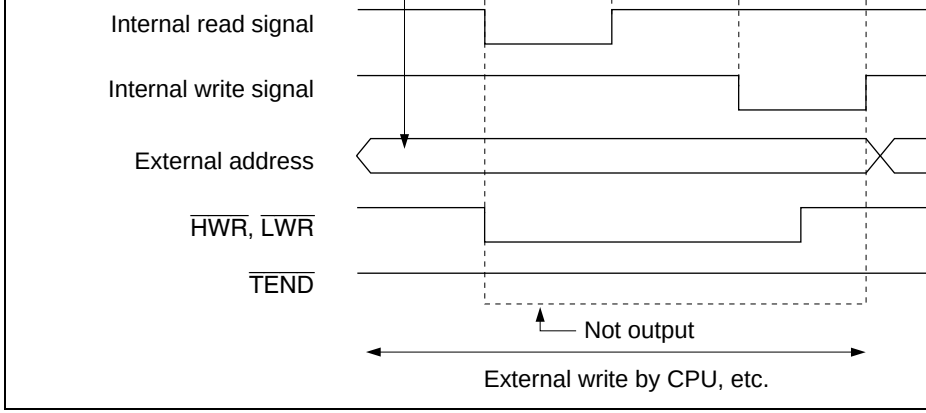


Figure 7.40 Example in Which Low Level Is Not Output at TEND Pi

Activation by Falling Edge on DREQ Pin

DREQ pin falling edge detection is performed in synchronization with DMAC interna
The operation is as follows:

[1] Activation request wait state: Waits for detection of a low level on the DREQ pin,
switches to [2].

[2] Transfer wait state: Waits for DMAC data transfer to become possible, and switch

[3] Activation request disabled state: Waits for detection of a high level on the DREQ
switches to [1].

After DMAC transfer is enabled, a transition is made to [1]. Thus, initial activation af
enabled is performed by detection of a low level.

Internal Interrupt after End of Transfer

When the DTE bit is cleared to 0 by the end of transfer or an abort, the selected interna request will be sent to the CPU or DTC even if DTA is set to 1.

Also, if internal DMAC activation has already been initiated when operation is aborted transfer is executed but flag clearing is not performed for the selected internal interrupt DTA is set to 1.

An internal interrupt request following the end of transfer or an abort should be handle CPU as necessary.

Channel Re-Setting

To reactivate a number of channels when multiple channels are enabled, use exclusive transfer end interrupts, and perform DMABCR control bit operations exclusively.

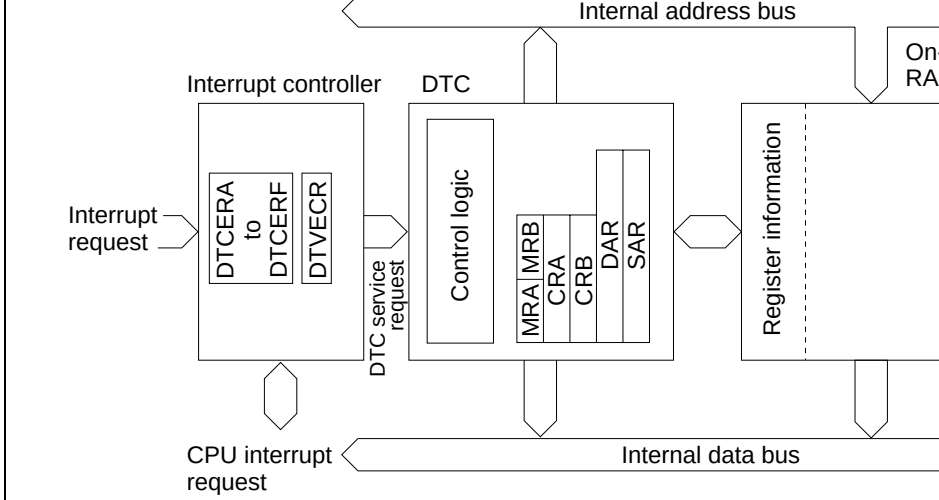
Note, in particular, that in cases where multiple interrupts are generated between readi writing of DMABCR, and a DMABCR operation is performed during new interrupt ha DMABCR write data in the original interrupt handling routine will be incorrect, and th invalidate the results of the operations by the multiple interrupts. Ensure that overlappi DMABCR operations are not performed by multiple interrupts, and that there is no sep between read and write operations by the use of a bit-manipulation instruction.

Also, when the DTE and DTME bits are cleared by the DMAC or are written with 0, th first be read while cleared to 0 before the CPU can write a 1 to them.

8.1.1 Features

The features of the DTC are:

- Transfer possible over any number of channels
  - Transfer information is stored in memory
  - One activation source can trigger a number of data transfers (chain transfer)
- Wide range of transfer modes
  - Normal, repeat, and block transfer modes available
  - Incrementing, decrementing, and fixing of source and destination addresses ca
- Direct specification of 16-Mbyte address space possible
  - 24-bit transfer source and destination addresses can be specified
- Transfer can be set in byte or word units
- A CPU interrupt can be requested for the interrupt that activated the DTC
  - An interrupt request can be issued to the CPU after one data transfer ends
  - An interrupt request can be issued to the CPU after the specified data transfers completely ended
- Activation by software is possible



Legend:

MRA, MRB	: DTC mode registers A and B
CRA, CRB	: DTC transfer count registers A and B
SAR	: DTC source address register
DAR	: DTC destination address register
DTCERA to DTCERF	: DTC enable registers A to F
DTVECR	: DTC vector register

Figure 8.1 Block Diagram of DTC

DTC source address register	SAR	—*2	Undefined	—*3
DTC destination address register	DAR	—*2	Undefined	—*3
DTC transfer count register A	CRA	—*2	Undefined	—*3
DTC transfer count register B	CRB	—*2	Undefined	—*3
DTC enable registers	DTCER	R/W	H'00	H'FF30
DTC vector register	DTVECR	R/W	H'00	H'FF37
Module stop control register	MSTPCR	R/W	H'3FFF	H'FF3C

Notes: 1. Lower 16 bits of the address.
2. Registers within the DTC cannot be read or written to directly.
3. Addresses H'F800 to H'FBFF contain register information. When the DTC i not clear the RAME bit in SYSCR to 0.

Initial value :	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined
R/W :	—	—	—	—	—	—	—

Bits 7 and 6—Source Address Mode 1 and 0 (SM1, SM0): These bits specify whethe to be incremented, decremented, or left fixed after a data transfer.

Bit 7	Bit 6	
SM1	**SM0**	**Description**
0	—	SAR is fixed
1	0	SAR is incremented after a transfer (by +1 when Sz = 0; by +2 when Sz = 1)
	1	SAR is decremented after a transfer (by –1 when Sz = 0; by –2 when Sz = 1)

Bits 5 and 4—Destination Address Mode 1 and 0 (DM1, DM0): These bits specify v DAR is to be incremented, decremented, or left fixed after a data transfer.

Bit 5	Bit 4	
DM1	**DM0**	**Description**
0	—	DAR is fixed
1	0	DAR is incremented after a transfer (by +1 when Sz = 0; by +2 when Sz = 1)
	1	DAR is decremented after a transfer (by –1 when Sz = 0; by –2 when Sz = 1)

Bit 1—DTC Transfer Mode Select (DTS): Specifies whether the source side or the d
side is set to be a repeat area or block area, in repeat mode or block transfer mode.

Bit 1

DTS	Description
0	Destination side is repeat area or block area
1	Source side is repeat area or block area

Bit 0—DTC Data Transfer Size (Sz): Specifies the size of data to be transferred.

Bit 0

Sz	Description
0	Byte-size transfer
1	Word-size transfer

MRB is an 8-bit register that controls the DTC operating mode.

Bit 7—DTC Chain Transfer Enable (CHNE): Specifies chain transfer. With chain tr
number of data transfers can be performed consecutively in response to a single transfe

In data transfer with CHNE set to 1, determination of the end of the specified number o
clearing of the interrupt source flag, and clearing of DTCER is not performed.

Bit 7

CHNE	Description
0	End of DTC data transfer (activation waiting state is entered)
1	DTC chain transfer (new register information is read, then data is transferre

Bit 6—DTC Interrupt Select (DISEL): Specifies whether interrupt requests to the CP
disabled or enabled after a data transfer.

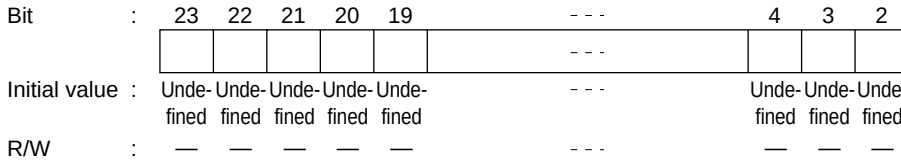
Bit 6

DISEL	Description
0	After a data transfer ends, the CPU interrupt is disabled unless the transfer 0 (the DTC clears the interrupt source flag of the activating interrupt to 0)
1	After a data transfer ends, the CPU interrupt is enabled (the DTC does not interrupt source flag of the activating interrupt to 0)

Bits 5 to 0—Reserved: These bits have no effect on DTC operation in the H8S/2655 G
should always be written with 0.

SAR is a 24-bit register that designates the source address of data to be transferred by
For word-size transfer, specify an even source address.

8.2.4 DTC Destination Address Register (DAR)



DAR is a 24-bit register that designates the destination address of data to be transferre
DTC. For word-size transfer, specify an even destination address.

CRA is a 16-bit register that designates the number of times data is to be transferred by

In normal mode, the entire CRA functions as a 16-bit transfer counter (1 to 65536). It is decremented by 1 every time data is transferred, and transfer ends when the count reach

In repeat mode or block transfer mode, the CRA is divided into two parts: the upper 8 b (CRAH) and the lower 8 bits (CRAL). CRAH holds the number of transfers while CRA functions as an 8-bit transfer counter (1 to 256). CRAL is decremented by 1 every time transferred, and the contents of CRAH are sent when the count reaches H'00. This oper repeated.

8.2.6 DTC Transfer Count Register B (CRB)

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	
Initial value:		Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	U f
R/W	:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	

CRB is a 16-bit register that designates the number of times data is to be transferred by block transfer mode. It functions as a 16-bit transfer counter (1 to 65536) that is decrem every time data is transferred, and transfer ends when the count reaches H'0000.

with bits corresponding to the interrupt sources that can activate the DTC. These bits e
disable DTC service for the corresponding interrupt sources.

The DTC enable registers are initialized to H'00 by a reset and in hardware standby m

Bit n—DTC Activation Enable (DTCEn)

Bit n	
DTCEn	**Description**
0	DTC activation by this interrupt is disabled [Clearing conditions] • When the DISEL bit is 1 and the data transfer has ended • When the specified number of transfers have ended
1	DTC activation by this interrupt is enabled [Holding condition] When the DISEL bit is 0 and the specified number of transfers have not en

Note: n = 7 to 0

A DTCE bit can be set for each interrupt source that can activate the DTC. The corres
between interrupt sources and DTCE bits is shown in table 8.4, together with the vecto
generated for each interrupt controller.

For DTCE bit setting, read/write operations must be performed using bit-manipulation
such as BSET and BCLR. For the initial setting only, however, when multiple activati
are set at one time, it is possible to disable interrups and write after executing a dumm
relevant register.

is read.

DTVECR is an 8-bit readable/writable register that enables or disables DTC activation software, and sets a vector number for the software activation interrupt.

DTVECR is initialized to H'00 by a reset and in hardware standby mode.

Bit 7—DTC Software Activation Enable (SWDTE): Enables or disables DTC activa software.

When clearing the SWDTE bit to 0 by software, write 0 to SWDTE after reading SWD

Bit 7

SWDTE	Description
0	DTC software activation is disabled ([Clearing condition] When the DISEL bit is 0 and the specified number of transfers have not en
1	DTC software activation is enabled [Holding conditions] • When the DISEL bit is 1 and data transfer has ended • When the specified number of transfers have ended • During data transfer due to software activation

Bits 6 to 0—DTC Software Activation Vectors 6 to 0 (DTVEC6 to DTVEC0): Thes specify a vector number for DTC software activation.

The vector address is expressed as H'0400 + ((vector number) << 1). <<1 indicates a or shift. For example, when DTVEC6 to DTVEC0 = H'10, the vector address is H'0420.

MSTPCR is a 16-bit readable/writable register that performs module stop mode contro

When the MSTP14 bit in MSTPCR is set to 1, the DTC operation stops at the end of t and a transition is made to module stop mode. However, 1 cannot be written in the MS while the DTC is operating. For details, see section 21.5, Module Stop Mode.

MSTPCR is initialized to H'3FFF by a reset and in hardware standby mode. It is not in software standby mode.

Bit 14—Module Stop (MSTP14): Specifies the DTC module stop mode.

Bit 14

MSTP14	Description
0	DTC module stop mode cleared
1	DTC module stop mode set

8.3 Operation

8.3.1 Overview

When activated, the DTC reads register information that is already stored in memory a data on the basis of that register information. After the data transfer, it writes updated information back to memory. Pre-storage of register information in memory makes it transfer data over any required number of channels. Setting the CHNE bit to 1 makes perform a number of transfers with a single activation.

Figure 8.2 shows a flowchart of DTC operation.

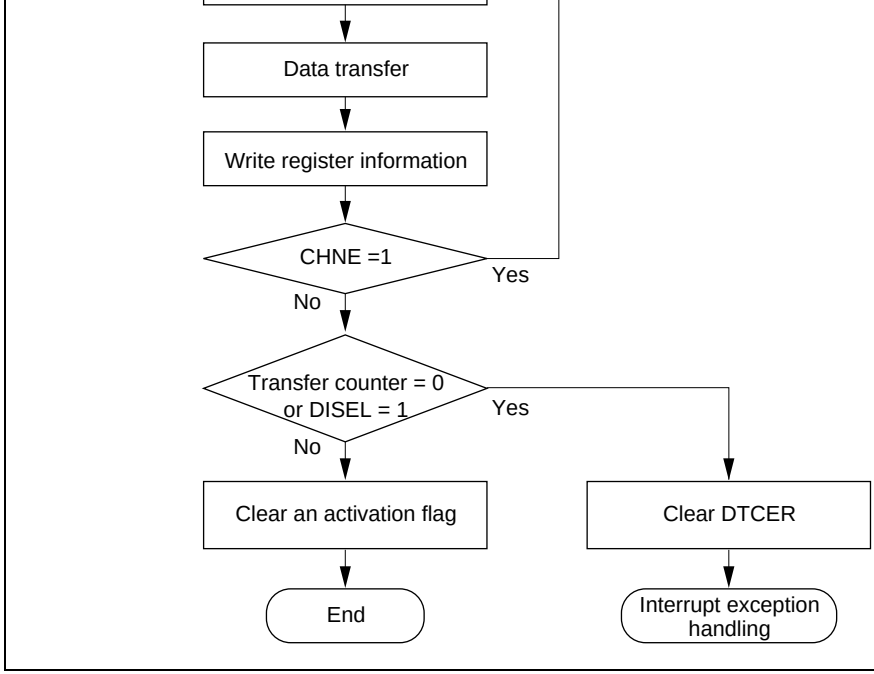


Figure 8.2 Flowchart of DTC Operation

The DTC transfer mode can be normal mode, repeat mode, or block transfer mode.

The 24-bit SAR designates the DTC transfer source address and the 24-bit DAR design transfer destination address. After each transfer, SAR and DAR are independently incre decremented, or left fixed.

Table 8.2 outlines the functions of the DTC.

— Memory addresses are incremented or decremented by 1 or 2
— Up to 65,536 transfers possible

- Repeat mode
  — One transfer request transfers one byte or one word
  — Memory addresses are incremented or decremented by 1 or 2
  — After the specified number of transfers (1 to 256), the initial state resumes and operation continues
- Block transfer mode
  — One transfer request transfers a block of the specified size
  — Block size is from 1 to 256 bytes or words
  — Up to 65,536 transfers possible
  — A block area can be designated at either the source or destination

- SCI TXI or RXI
- A/D converter ADI
- DMAC DEND
- Software

8.3.2 Activation Sources

The DTC operates when activated by an interrupt or by a write to DTVECR by softwa
interrupt request can be directed to the CPU or DTC, as designated by the correspondi
bit. An interrupt becomes a DTC activation source when the corresponding bit is set to
CPU interrupt source when the bit is cleared to 0.

At the end of a data transfer (or the last consecutive transfer in the case of chain transf
activation source or corresponding DTCER bit is cleared. Table 8.3 shows activation s
DTCER clearance. The activation source flag, in the case of RXI0, for example, is the
of SCI0.

The activation source flag is cleared to 0	The activation source flag rema A request is issued to the CPU f activation source interrupt

Figure 8.3 shows a block diagram of activation source control. For details see section 5 Controller.

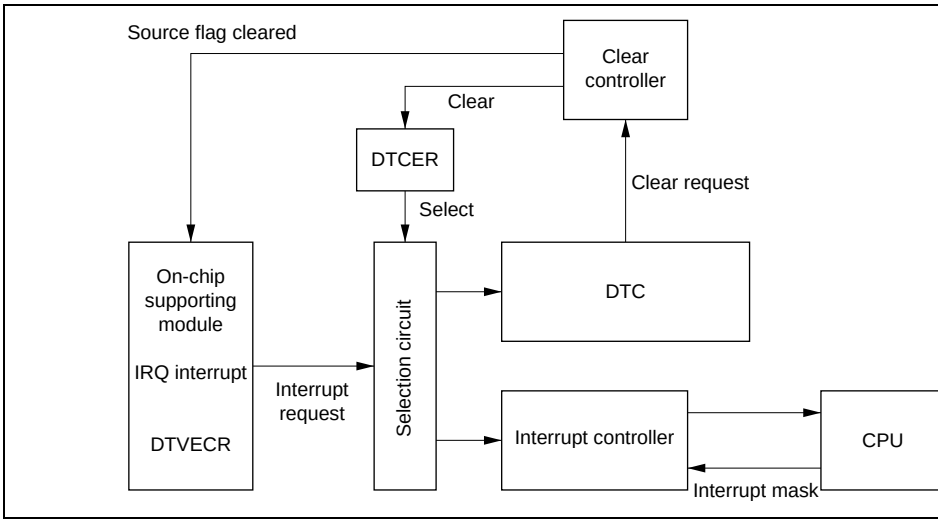


Figure 8.3 Block Diagram of DTC Activation Source Control

When an interrupt has been designated a DTC activation source, existing CPU mask le interrupt controller priorities have no effect. If there is more than one activation source time, the DTC operates in accordance with the default priorities.

The DTC reads the start address of the register information from the vector address se
activation source, and then reads the register information from that start address. The r
information can be placed at predetermined addresses in the on-chip RAM. The start a
the register information should be an integral multiple of four.

The configuration of the vector address is the same in both normal and advanced mod
unit being used in both cases. These two bytes specify the lower bits of the address in
RAM.

IRQ1		17	H'0422	DTCEA6
IRQ2		18	H'0424	DTCEA5
IRQ3		19	H'0426	DTCEA4
IRQ4		20	H'0428	DTCEA3
IRQ5		21	H'042A	DTCEA2
IRQ6		22	H'042C	DTCEA1
IRQ7		23	H'042E	DTCEA0
ADI (A/D conversion end)	A/D	28	H'0438	DTCEB6
TGI0A (GR0A compare match/ input capture)	TPU channel 0	32	H'0440	DTCEB5
TGI0B (GR0B compare match/ input capture)		33	H'0442	DTCEB4
TGI0C (GR0C compare match/ input capture)		34	H'0444	DTCEB3
TGI0D (GR0D compare match/ input capture)		35	H'0446	DTCEB2
TGI1A (GR1A compare match/ input capture)	TPU channel 1	40	H'0450	DTCEB1
TGI1B (GR1B compare match/ input capture)		41	H'0452	DTCEB0
TGI2A (GR2A compare match/ input capture)	TPU channel 2	44	H'0458	DTCEC7
TGI2B (GR2B compare match/ input capture)		45	H'045A	DTCEC6

TGI3D (GR3D compare match/ input capture)		51	H'0466	DTCEC2
TGI4A (GR4A compare match/ input capture)	TPU channel 4	56	H'0470	DTCEC1
TGI4B (GR4B compare match/ input capture)		57	H'0472	DTCEC0
TGI5A (GR5A compare match/ input capture)	TPU channel 5	60	H'0478	DTCED5
TGI5B (GR5B compare match/ input capture)		61	H'047A	DTCED4
CMIA0	8-bit timer channel 0	64	H'0480	DTCED3
CMIB0		65	H'0482	DTCED2
CMIA1	8-bit timer channel 1	68	H'0488	DTCED1
CMIB1		69	H'048A	DTCED0
DMTEND0A (DMAC transfer end 0)	DMAC	72	H'0490	DTCEE7
DMTEND0B (DMAC transfer end 1)		73	H'0492	DTCEE6
DMTEND1A (DMAC transfer end 2)		74	H'0494	DTCEE5
DMTEND1B (DMAC transfer end 3)		75	H'0496	DTCEE4
RXI0 (reception complete 0)	SCI channel 0	81	H'04A2	DTCEE3
TXI0 (transmit data empty 0)		82	H'04A4	DTCEE2
RXI1 (reception complete 1)	SCI channel 1	85	H'04AA	DTCEE1
TXI1 (transmit data empty 1)		86	H'04AC	DTCEE0
RXI2 (reception complete 2)	SCI channel 2	89	H'04B2	DTCEF7
TXI2 (transmit data empty 2)		90	H'04B4	DTCEF6

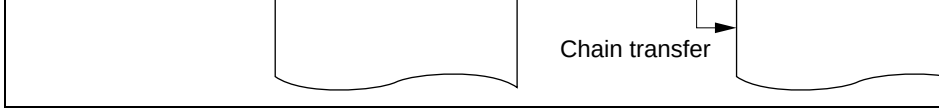


Figure 8.4 Correspondence between DTC Vector Address and Register Infor

8.3.4 Location of Register Information in Address Space

Figure 8.5 shows how the register information should be located in the address space.

Locate the MRA, SAR, MRB, DAR, CRA, and CRB registers, in that order, from the s of the register information (contents of the vector address). In the case of chain transfer information should be located in consecutive areas.

Locate the register information in the on-chip RAM (addresses: H'FFF800 to H'FFFBF

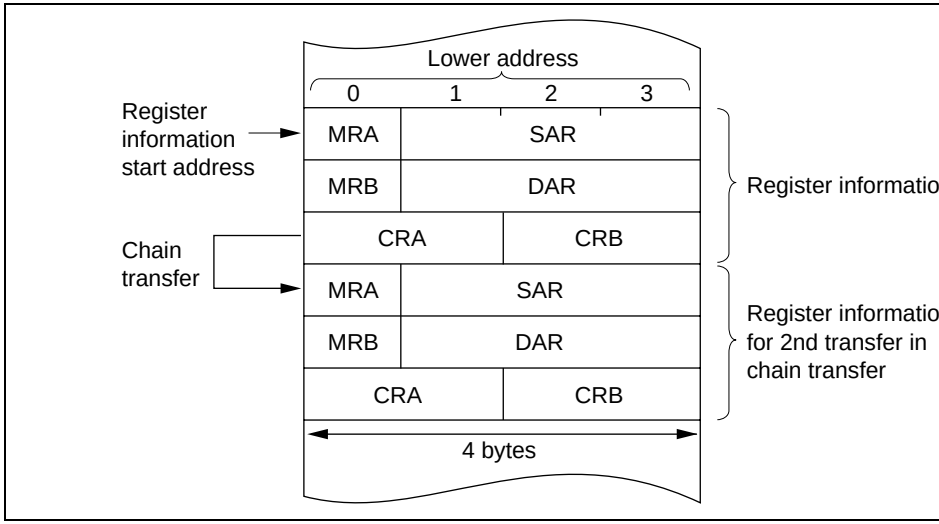


Figure 8.5 Location of Register Information in Address Space

Table 8.5 Register Information in Normal Mode

Name	Abbreviation	Function
DTC source address register	SAR	Designates source addre
DTC destination address register	DAR	Designates destination a
DTC transfer count register A	CRA	Designates transfer coun
DTC transfer count register B	CRB	Not used

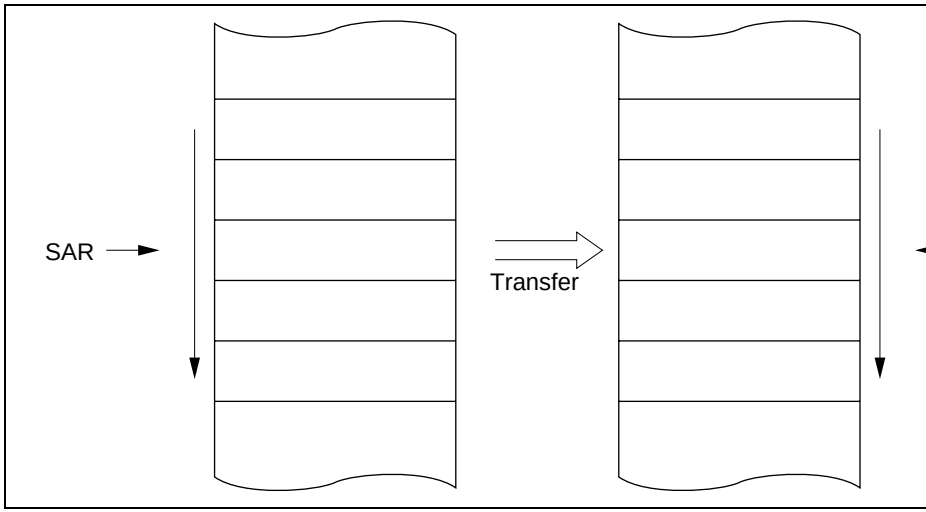


Figure 8.6 Memory Mapping in Normal Mode

Table 8.6 lists the register information in repeat mode and figure 8.7 shows memory ma
repeat mode.

Table 8.6 Register Information in Repeat Mode

Name	Abbreviation	Function
DTC source address register	SAR	Designates source addres
DTC destination address register	DAR	Designates destination ad
DTC transfer count register AH	CRAH	Holds number of transfers
DTC transfer count register AL	CRAL	Designates transfer count
DTC transfer count register B	CRB	Not used

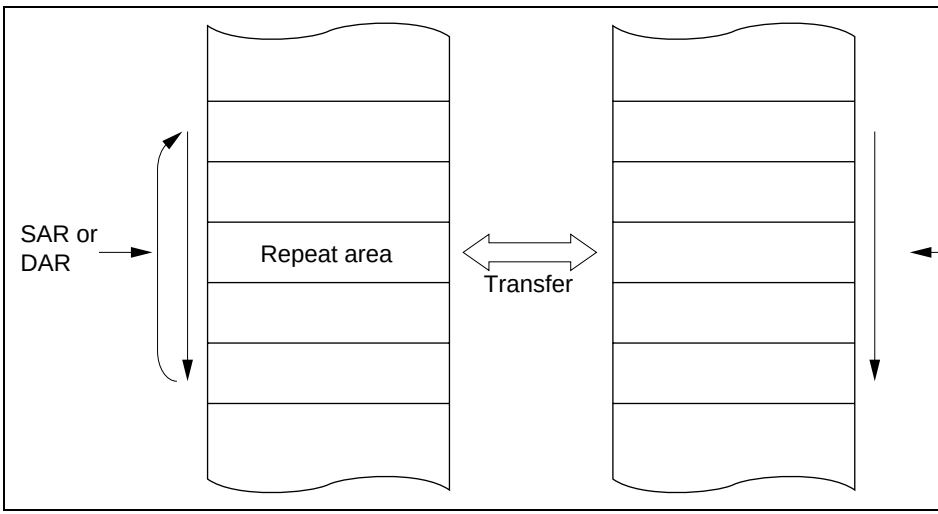


Figure 8.7 Memory Mapping in Repeat Mode

CPU interrupt is requested.

Table 8.7 lists the register information in block transfer mode and figure 8.8 shows me
mapping in block transfer mode.

Table 8.7 Register Information in Block Transfer Mode

Name	Abbreviation	Function
DTC source address register	SAR	Designates transfer sour
DTC destination address register	DAR	Designates destination a
DTC transfer count register AH	CRAH	Holds block size
DTC transfer count register AL	CRAL	Designates block size co
DTC transfer count register B	CRB	Transfer count

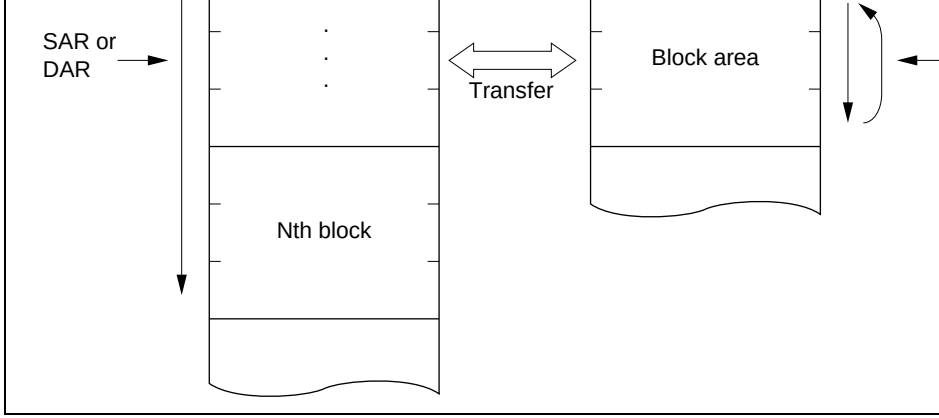


Figure 8.8 Memory Mapping in Block Transfer Mode

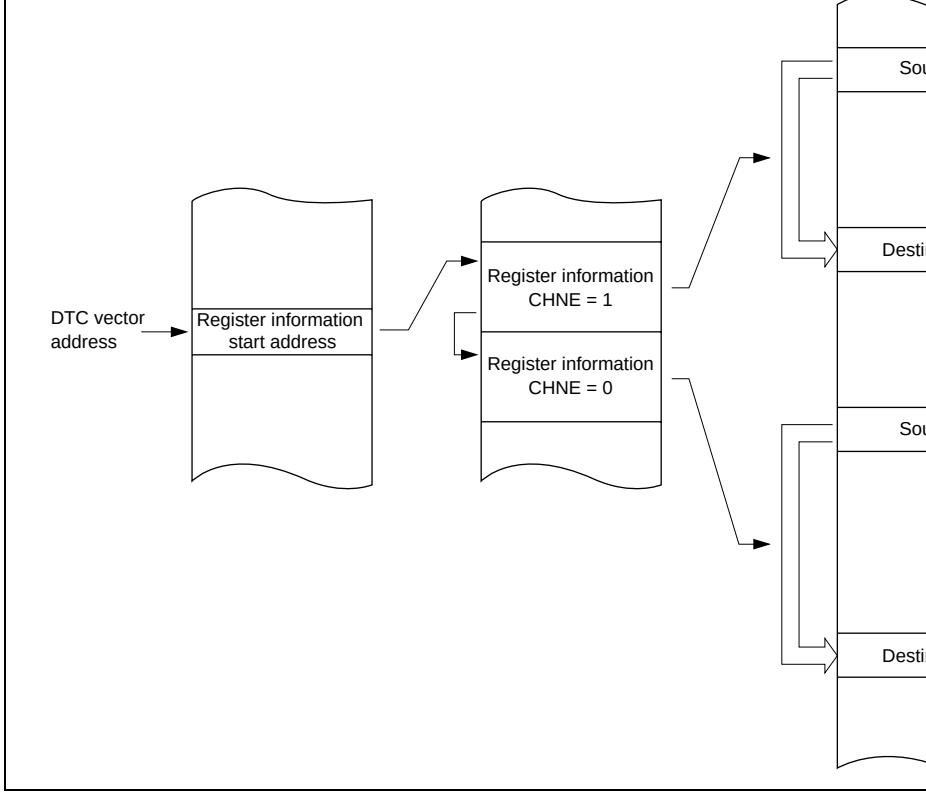


Figure 8.9 Chain Transfer Memory Map

In the case of transfer with CHNE set to 1, an interrupt request to the CPU is not gene end of the specified number of transfers or by setting of the DISEL bit to 1, and the in source flag for the activation source is not affected.

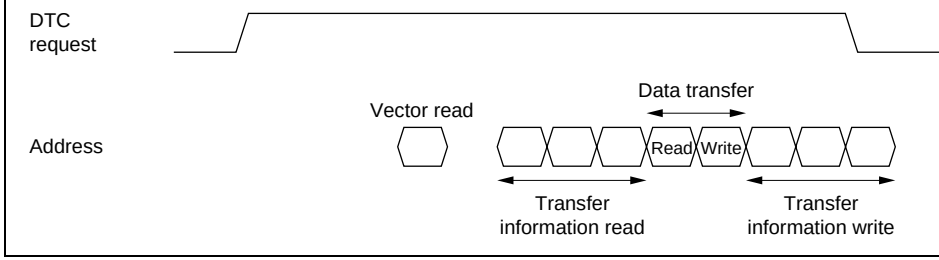


Figure 8.10 DTC Operation Timing (Example in Normal Mode or Repeat M

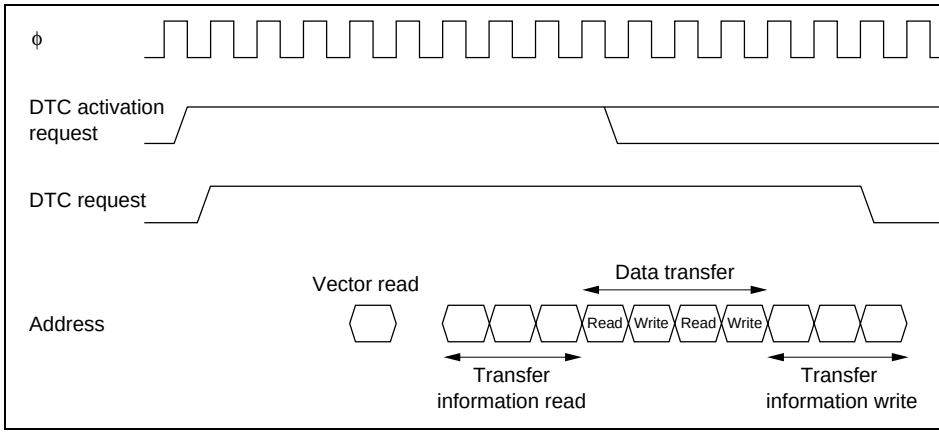


Figure 8.11 DTC Operation Timing (Example of Block Transfer Mode with Block Size of 2)

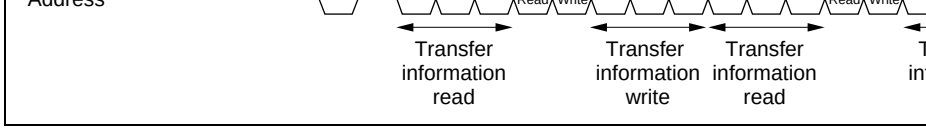


Figure 8.12 DTC Operation Timing (Example of Chain Transfer)

8.3.10 Number of DTC Execution States

Table 8.8 lists execution statuses for a single DTC data transfer, and table 8.9 shows t
states required for each execution status.

Table 8.8 DTC Execution Statuses

Mode	Vector Read I	Register Information Read/Write J	Data Read K	Data Write L	I C M
Normal	1	6	1	1	3
Repeat	1	6	1	1	3
Block transfer	1	6	N	N	3

N: Block size (initial setting of CRA

	Register information read/write	S_J	1	—	—	—	—	—	—
	Byte data read	S_K	1	1	2	2	2	3 + m	2
	Word data read	S_K	1	1	4	2	4	6 + 2m	2
	Byte data write	S_L	1	1	2	2	2	3 + m	2
	Word data write	S_L	1	1	4	2	4	6 + 2m	2
	Internal operation	S_M	1						

The number of execution states is calculated from the formula below. Note that Σ mean of all transfers activated by one activation event (the number in which the CHNE bit is plus 1).

$$\text{Number of execution states} = I \cdot S_I + \Sigma (J \cdot S_J + K \cdot S_K + L \cdot S_L) + M \cdot S_M$$

For example, when the DTC vector address table is located in on-chip ROM, normal m and data is transferred from the on-chip ROM to an internal I/O register, the time requi DTC operation is 13 states. The time from activation to the end of the data write is 10 s

[3] Set the corresponding bit in DTCER to 1.

[4] Set the enable bits for the interrupt sources to be used as the activation sources to 1
is activated when an interrupt used as an activation source is generated.

[5] After the end of one data transfer, or after the specified number of data transfers ha
the DTCE bit is cleared to 0 and a CPU interrupt is requested. If the DTC is to con
transferring data, set the DTCE bit to 1.

Activation by Software

The procedure for using the DTC with software activation is as follows:

[1] Set the MRA, MRB, SAR, DAR, CRA, and CRB register information in the on-ch

[2] Set the start address of the register information in the DTC vector address.

[3] Check that the SWDTE bit is 0.

[4] Write 1 to SWDTE bit and the vector number to DTVECR.

[5] Check the vector number written to DTVECR.

[6] After the end of one data transfer, if the DISEL bit is 0 and a CPU interrupt is not
the SWDTE bit is cleared to 0. If the DTC is to continue transferring data, set the S
to 1. When the DISEL bit is 1, or after the specified number of data transfers have
SWDTE bit is held at 1 and a CPU interrupt is requested.

SCI RDR address in SAR, the start address of the RAM area where the data will be
DAR, and 128 (H'0080) in CRA. CRB can be set to any value.

[2] Set the start address of the register information at the DTC vector address.

[3] Set the corresponding bit in DTCER to 1.

[4] Set the SCI to the appropriate receive mode. Set the RIE bit in SCR to 1 to enable t
complete (RXI) interrupt. Since the generation of a receive error during the SCI rec
operation will disable subsequent reception, the CPU should be enabled to accept re
interrupts.

[5] Each time reception of one byte of data ends on the SCI, the RDRF flag in SSR is s
RXI interrupt is generated, and the DTC is activated. The receive data is transferred
to RAM by the DTC. DAR is incremented and CRA is decremented. The RDRF fla
automatically cleared to 0.

[6] When CRA becomes 0 after the 128 data transfers have ended, the RDRF flag is he
DTCE bit is cleared to 0, and an RXI interrupt request is sent to the CPU. The inter
handling routine should perform wrap-up processing.

[1] Set transfer to the PPG's NDR. Set MRA to incrementing source address (SM1 = 1
fixed destination address (DM1 = DM0 = 0), repeat mode (MD1 = 0, MD0 = 1), a
(Sz = 1). Set the source side as a repeat area (DTS = 1). Set MRB to chain mode (C
DISEL = 0). Set the data table start address in SAR, the NDRH address in DAR, a
table size in CRAH and CRAL. CRB can be set to any value.

[2] Set transfer to the TPU's TGR. Set MRA to incrementing source address (SM1 = 1
fixed destination address (DM1 = DM0 = 0), normal mode (MD1 = MD0 = 0), and
(Sz = 1). Set the data table start address in SAR, the TGRA address in DAR, and t
size in CRA. CRB can be set to any value.

[3] Locate the TPU transfer register information consecutively after the NDR transfer
information.

[4] Set the start address of the NDR transfer register information at the DTC vector ad

[5] Set the bit corresponding to TGIA in DTCER to 1.

[6] Set TGRA as an output compare register (output disabled) with TIOR, and enable
interrupt with TIER.

[7] Set the initial output value in PODR, and the next output value in NDR. Set bits in
NDER for which output is to be performed to 1. Using PCR, select the TPU compa
be used as the output trigger.

[8] Set the CST bit in TSTR to 1, and start the TCNT count operation.

[9] Each time a TGRA compare match occurs, the next output value is transferred to N
set value of the next output trigger period is transferred to TGRA. The activation s
flag is cleared.

[10] When the specified number of transfers are completed (the TPU transfer CRA va
TGFA flag is held at 1, the DTCE bit is cleared to 0, and a TGIA interrupt request
CPU. The interrupt handling routine should perform wrap-up processing.

0). Set the transfer source address (H'1000) in SAR, the destination address (H'2000
and 128 (H'8080) in CRA. Set 1 (H'0001) in CRB.

[2] Set the start address of the register information at the DTC vector address (H'04C0)

[3] Check that the SWDTE bit in DTVECR is 0. Check that there is currently no transf
by software.

[4] Write 1 to the SWDTE bit and the vector number (H'60) to DTVECR. The write d

[5] Read DTVECR again and check that it is set to the vector number (H'60). If it is no
indicates that the write failed. This is presumably because an interrupt occurred betv
3 and 4 and led to a different software activation. To activate this transfer, go back t

[6] If the write was successful, the DTC is activated and a block of 128 bytes of data is

[7] After the transfer, an SWDTEND interrupt occurs. The interrupt handling routine sh
the SWDTE bit to 0 and perform other wrap-up processing.

When the DISEL bit is 1 and one data transfer has ended, or the specified number of t
have ended, after data transfer ends, the SWDTE bit is held at 1 and an SWDTEND in
generated. The interrupt handling routine should clear the SWDTE bit to 0.

When the DTC is activated by software, an SWDTEND interrupt is not generated dur
transfer wait or during data transfer even if the SWDTE bit is set to 1.

8.5 Usage Notes

Module Stop

When the MSTP14 bit in MSTPCR is set to 1, the DTC clock stops, and the DTC ente
module stop state. However, 1 cannot be written in the MSTP14 bit while the DTC is

On-Chip RAM

The MRA, MRB, SAR, DAR, CRA, and CRB registers are all located in on-chip RAM
DTC is used, the RAME bit in SYSCR must not be cleared to 0.

DMAC Transfer End Interrupt

When DTC transfer is activated by a DMAC transfer end interrupt, regardless of the t
counter and DISEL bit, the DMAC's DTE bit is not subject to DTC control, and the w
priority. Consequently, an interrupt request is not sent to the CPU when the DTC trans
reaches 0.

Table 9.1 summarizes the port functions. The pins of each port also have other functio

Each port includes a data direction register (DDR) that controls input/output (not prov
input-only port), a data register (DR) that stores output data, and a port register (PORT
read the pin states.

Ports A to E have a built-in MOS input pull-up function, and in addition to DR and DD
MOS input pull-up control register (PCR) to control the on/off state of MOS input pul

Ports 3 and A include an open-drain control register (ODR) that controls the on/off sta
output buffer PMOS.

Ports A to E can drive a single TTL load and 90 pF capacitive load, and ports 1 to 3, 5
can drive a single TTL load and 30 pF capacitive load. All the I/O ports can drive a D
transistor when in output mode. Ports 1, and A to C can drive an LED (10 mA sink cu

Port 2, and ports 6_4 to 6_7 and A_4 to A_7, are Schmitt-triggered inputs.

For block diagrams of the ports see appendix C, I/O Port Block Diagrams.

Port	Description	Pins	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5
Port 1	• 8-bit I/O port	$P1_7/PO_{15}/TIOCB_2/TCLKD$ $P1_6/PO_{14}/TIOCA_2$ $P1_5/PO_{13}/TIOCB_1/TCLKC$ $P1_4/PO_{12}/TIOCA_1$ $P1_3/PO_{11}/TIOCD_0/TCLKB$ $P1_2/PO_{10}/TIOCC_0/TCLKA$ $P1_1/PO_9/TIOCB_0/\overline{DACK}_1$ $P1_0/PO_8/TIOCA_0/\overline{DACK}_0$	8-bit I/O port also functioning as DMA controller ($\overline{DACK}_0$ and $\overline{DACK}_1$), TPU I/O pins (TCLKA, TCLKB, TCLKC, $TIOCA_0$, $TIOCB_0$, $TIOCC_0$, $TIOCD_0$, $TIOCA_1$, $TIOCB_1$, $TIOCA_2$, $TIOCB_2$) and PPG output pins (PO_{15} to PO_8)				
Port 2	• 8-bit I/O port • Schmitt-triggered input	$P2_7/PO_7/TIOCB_5/TMO_1$ $P2_6/PO_6/TIOCA_5/TMO_0$ $P25/PO_5/TIOCB_4/TMCI_1$ $P2_4/PO_4/TIOCA_4/TMRI_1$ $P2_3/PO_3/TIOCD_3/TMCI_0$ $P2_2/PO_2/TIOCC_3/TMRI_0$ $P2_1/PO_1/TIOCB_3$ $P2_0/PO_0/TIOCA_3$	8-bit I/O port also functioning as TPU I/O pins ($TIOCC_3$, $TIOCD_3$, $TIOCA_4$, $TIOCB_4$, $TIOCA_5$, $TIOCB_5$ (channels 0 and 1) I/O pins ($TMRI_0$, $TMCI_0$, TMO_0, TMO_1) and PPG output pins (PO_7 to PO_0)				
Port 3	• 6-bit I/O port • Open-drain output capability	$P3_5/SCK_1$ $P3_4/SCK_0$ $P3_3/RxD_1$ $P3_2/RxD_0$ $P3_1/TxD_1$ $P3_0/TxD_0$	6-bit I/O port also functioning as SCI (channels 0 and 1) (TxD_0, RxD_0, SCK_0, TxD_1, RxD_1, SCK_1)				
Port 4	• 8-bit input port	$P4_7/AN_7/DA_1$ $P4_6/AN_6/DA_0$ $P4_5/AN_5$ $P4_4/AN_4$ $P4_3/AN_3$ $P4_2/AN_2$ $P4_1/AN_1$ $P4_0/AN_0$	8-bit input port also functioning as A/D converter (AN_0 to AN_7) and D/A converter analog outputs (DA_1 a				

Port	Description	Pins	Mode 1	Mode 2	Mode 3	Mode 4
Port 5	• 4-bit I/O port	$P5_3/\overline{ADTRG}$ $P5_2/SCK_2$ $P5_1/RxD_2$ $P5_0/TxD_2$	4-bit I/O port also functioning as SCI (cha RxD_2, SCK_2) and A/D converter input pin			
Port 6	• 8-bit I/O port • Schmitt-triggered input ($P6_4$ to $P6_7$)	$P6_7/\overline{IRQ_3}/\overline{CS_7}$ $P6_6/\overline{IRQ_2}/\overline{CS_6}$ $P6_5/\overline{IRQ_1}$ $P6_4/\overline{IRQ_0}$ $P6_3/\overline{TEND_1}$ $P6_2/\overline{DREQ_1}$ $P6_1/\overline{TEND_0}/\overline{CS_5}$ $P6_0/\overline{DREQ_0}/\overline{CS_4}$	8-bit I/O port also functioning as DMA controller I/O pins ($\overline{DREQ_0}$, $\overline{TEND_0}$, $\overline{DREQ_1}$, $\overline{TEND_1}$) and interrupt input pins ($\overline{IRQ_0}$ to $\overline{IRQ_3}$)			8-bit I/O po as DMA co ($\overline{DREQ_0}$, $\overline{T}$ $\overline{TEND_1}$), b pins ($\overline{CS_4}$ interrupt in ($\overline{IRQ_0}$ to $\overline{IR}$

Port	Description	Pins	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5
Port A	• 8-bit I/O port • Built-in MOS input pull-up • Open-drain output capability • Schmitt-triggered input (PA_4 to PA_7)	$PA_7/A_{23}/\overline{IRQ_7}$ $PA_6/A_{22}/\overline{IRQ_6}$ $PA_5/A_{21}/\overline{IRQ_5}$	Dual function as I/O ports and interrupt input pins ($\overline{IRQ_7}$ to $\overline{IRQ_4}$)			When DDR = 0 (after reset): dual function as input ports and interrupt input pins ($\overline{IRQ_7}$ to $\overline{IRQ_5}$) When DDR = 1: address output	
		$PA_4/A_{20}/\overline{IRQ_4}$				Address output	
		PA_3/A_{19} to PA_0/A_{16}	I/O ports			Address output	
Port B	• 8-bit I/O port • Built-in MOS input pull-up	PB_7/A_{15} to PB_0/A_8	Address output	When DDR = 0 (after reset): input port When DDR = 1: address output	I/O port	Address output	

Port	Description	Pins	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5
Port C	• 8-bit I/O port • Built-in MOS input pull-up	PC_7/A_7 to PC_0/A_0	Address output	When DDR = 0 (after reset): input port When DDR = 1: address output	I/O port	Address output	
Port D	• 8-bit I/O port • Built-in MOS input pull-up	PD_7/D_{15} to PD_0/D_8	Data bus input/output		I/O port	Data bus input/output	
Port E	• 8-bit I/O port • Built-in MOS input pull-up	PE_7/D_7 to PE_0/D_0	In 8-bit bus mode: I/O port In 16-bit bus mode: data bus input/output		I/O port	In 8-bit bus mode: I/O port In 16-bit bus mode: data bus input/output	
Port F	• 8-bit I/O port	PF_7/ϕ	When DDR = 0: input port When DDR = 1 (after reset): ϕ output		When DDR = 0 (after reset): input port When DDR = 1: ϕ output	When DDR = 0: input port When DDR = 1 (after reset)	
		$PF_6/\overline{AS}$ $PF_5/\overline{RD}$ $PF_4/\overline{HWR}$ $PF_3/\overline{LWR}$	$\overline{AS}$, $\overline{RD}$, $\overline{HWR}$, $\overline{LWR}$ output	$\overline{AS}$, $\overline{RD}$, $\overline{HWR}$, $\overline{LWR}$	I/O port	$\overline{AS}$, $\overline{RD}$, $\overline{HWR}$, $\overline{LWR}$ output	

Port	Description	Pins	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5
Port F	• 8-bit I/O port	PF_2/$\overline{LCAS}$/$\overline{WAIT}$/$\overline{BREQO}$	When WAITE = 0 and BREQOE = 0 (after reset): I/O port When WAITE = 1 and BREQOE = 0: $\overline{WAIT}$ input When WAITE = 0 and BREQOE = 1: $\overline{BREQO}$ input		I/O port	When WAITE = 0 and BR (after reset): I/O port When WAITE = 1 and BR $\overline{WAIT}$ input When WAITE = 0 and BR $\overline{BREQO}$ output When RMTS2 to RMTS0= CW2= 0, and LCASS= 0:	
		PF_1/$\overline{BACK}$ PF_0/$\overline{BREQ}$	When BRLE = 0 (after reset): I/O port When BRLE = 1: $\overline{BREQ}$ input, $\overline{BACK}$ output			When BRLE = 0 (after res When BRLE = 1: $\overline{BREQ}$ i output	
Port G	• 5-bit I/O port	PG_4/$\overline{CS_0}$	When DDR= 0*1: input port When DDR= 1*2: $\overline{CS_0}$ output		I/O port	When DDR = 0*1: input p When DDR = 1*2: $\overline{CS_0}$ ou	
		PG_3/$\overline{CS_1}$ PG_2/$\overline{CS_2}$ PG_1/$\overline{CS_3}$	I/O port			When DDR = 0 (after rese When DDR = 1: $\overline{CS_1}$, $\overline{CS_2}$	
		PG_0/$\overline{CAS}$/$\overline{OE}$				DRAM space set: $\overline{CAS}$ ou PSRAM space set: $\overline{OE}$ ou Otherwise (after reset): I/O	

Notes: 1. After a reset in mode 2 or 6
2. After a reset in mode 1, 4 or 5

Figure 9.1 shows the port 1 pin configuration.

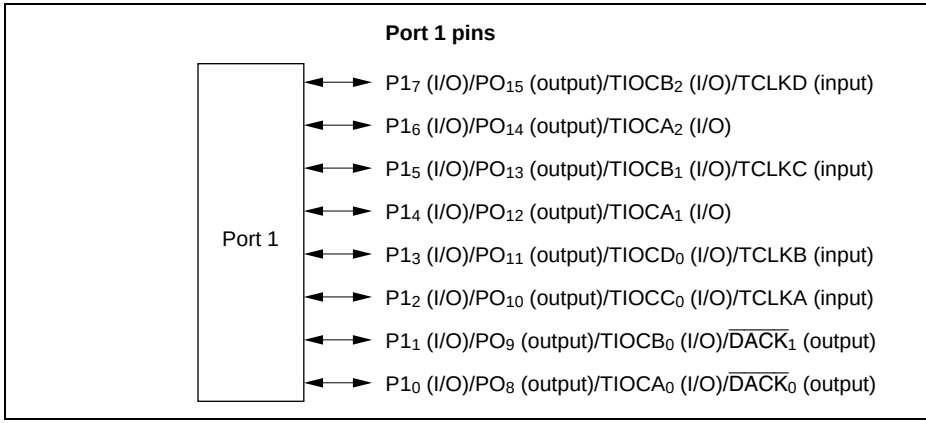


Figure 9.1 Port 1 Pin Functions

9.2.2 Register Configuration

Table 9.2 shows the port 1 register configuration.

Table 9.2 Port 1 Registers

Name	Abbreviation	R/W	Initial Value	Ad
Port 1 data direction register	P1DDR	W	H'00	H'
Port 1 data register	P1DR	R/W	H'00	H'
Port 1 register	PORT1	R	Undefined	H'

Note: * Lower 16 bits of the address.

pins of port 1. P1DDR cannot be read; if it is, an undefined value will be read.

Setting a P1DDR bit to 1 makes the corresponding port 1 pin an output pin, while clear to 0 makes the pin an input pin.

P1DDR is initialized to H'00 by a power-on reset, and in hardware standby mode. It ret prior state after a manual reset, and in software standby mode. As the PPG, TPU, and D initialized by a manual reset, the pin states are determined by the P1DDR and P1DR specifications.

Port 1 Data Register (P1DR)

Bit	:	7	6	5	4	3	2	1
		P17DR	P16DR	P15DR	P14DR	P13DR	P12DR	P11DR
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

P1DR is an 8-bit readable/writable register that stores output data for the port 1 pins (P

P1DR is initialized to H'00 by a power-on reset, and in hardware standby mode. It retai state after a manual reset, and in software standby mode.

PORT1 is an 8-bit read-only register that shows the pin states. It cannot be written to. output data for the port 1 pins ($P1_7$ to $P1_0$) must always be performed on P1DR.

If a port 1 read is performed while P1DDR bits are set to 1, the P1DR values are read. read is performed while P1DDR bits are cleared to 0, the pin states are read.

After a power-on reset and in hardware standby mode, PORT1 contents are determine states, as P1DDR and P1DR are initialized. PORT1 retains its prior state after a manu in software standby mode.

Pin	Selection Method and Pin Functions
$P1_7/PO_{15}/TIOCB_2/$ TCLKD	The pin function is switched as shown below according to the comb the TPU channel 2 setting by bits MD3 to MD0 in TMDR2, bits IOB3 TIOR2, bits CCLR1 and CCLR0 in TCR2, bits TPSC2 to TPSC0 in T TCR5, bit NDER15 in NDERH, and bit P17DDR.

TPU Channel 2 Setting	Table Below (1)	Table Below	
P17DDR	—	0	1
NDER15	—	—	0
Pin function	$TIOCB_2$ output	$P1_7$ input	$P1_7$ output
		$TIOCB_2$ inpu	
	TCLKD input*2		

Notes: 1. $TIOCB_2$ input when MD3 to MD0 = B'0000, B'01xx, and
2. TCLKD input when the setting for either TCR0 or TCR5 to TPSC0 = B'111.
TCLKD input when channels 2 and 4 are set to phase co mode.

TPU Channel 2 Setting	(2)	(1)	(2)	(2)	(1)
MD3 to MD0	B'0000, B'01xx		B'0010		B'0011
IOB3 to IOB0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	—	B'xx00	Other th
CCLR1, CCLR0	—	—	—	—	Other than B'10
Output function	—	Output compare output	—	—	PWM mode 2 output

x

NDER14	—		—	0	
Pin function	$TIOCA_2$ output		$P1_6$ input	$P1_6$ output	
			$TIOCA_2$ inp		

Note: 1. $TIOCA_2$ input when MD3 to MD0 = B'0000, B'01xx, and

TPU Channel 2 Setting	(2)	(1)	(2)	(1)	(1)
MD3 to MD0	B'0000, B'01xx		B'001x	B'0010	B
IOA3 to IOA0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	B'xx00	Other than B	
CCLR2 to CCLR0	—	—	—	—	Other than B'0
Output function	—	Output compare output	—	PWM mode 1 output*2	PWM mode output

Note: 2. $TIOCB_2$ output is disabled.

NDER13	—	—	0
Pin function	$TIOCB_1$ output	$P1_5$ input	$P1_5$ output
		$TIOCB_1$ inpu	
	TCLKC input*2		

Notes: 1. $TIOCB_1$ input when MD3 to MD0 = B'0000, B'01xx and I
IOB0 = B'10xx.

2. TCLKC input when the setting for either TCR0 or TCR2
to TPSC0 = B'110; or when the setting for either TCR4 c
TPSC2 to TPSC0 = B'101.
TCLKC input when channels 2 and 4 are set to phase co
mode.

TPU Channel 1 Setting	(2)	(1)	(2)	(2)	(1)
MD3 to MD0	B'0000, B'01xx		B'0010		B'0011
IOB3 to IOB0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	—	B'xx00	Other th
CCLR1, CCLR0	—	—	—	—	Other than B'10
Output function	—	Output compare output	—	—	PWM mode 2 output

x

NDER12	—		—	0	
Pin function	$TIOCA_1$ output		$P1_4$ input	$P1_4$ output	
			$TIOCA_1$ inp		

Note: 1. $TIOCA_1$ input when MD3 to MD0 = B'0000, B'01xx, IOA
B'10xx.

TPU Channel 1 Setting	(2)	(1)	(2)	(1)	(1)
MD3 to MD0	B'0000, B'01xx		B'001x	B'0010	B
IOA3 to IOA0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	B'xx00	Other than B'xx00	Other t
CCLR1, CCLR0	—	—	—	—	Other than B'0
Output function	—	Output compare output	—	PWM mode 1 output*2	PWM mode 1 output

Note: 2. $TIOCB_1$ output is disabled.

NDER11	—	—	0
Pin function	$TIOCD_0$ output	$P1_3$ input	$P1_3$ output
		$TIOCD_0$ inp	
	TCLKB input*2		

Notes: 1. $TIOCD_0$ input when MD3 to MD0 = B'0000, IOD3 to IOD

2. TCLKB input when the setting for TCR0 to TCR2 is: TPS
TPSC0 = B'101;
TCLKB input when channels 1 and 5 are set to phase co
mode.

TPU Channel 0 Setting	(2)	(1)	(2)	(2)	(1)
MD3 to MD0	B'0000		B'0010	B'0011	
IOD3 to IOD0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	—	B'xx00	Other th
CCLR2 to CCLR0	—	—	—	—	Other than B'110
Output function	—	Output compare output	—	—	PWM mode 2 output

x

NDER10	—		—	0	
Pin function	$TIOCC_0$ output		$P1_2$ input	$P1_2$ output	
			$TIOCC_0$ inp		
	TCLKA input*2				

Notes: 1. $TIOCC_0$ input when MD3 to MD0 = B'0000, and IOC3 t
B'10xx.

2. TCLKA input when the setting for TCR0 to TCR5 is: TP
TPSC0 = B'100; TCLKA input when channels 1 and 5 a
phase counting mode.

TPU Channel 0 Setting	(2)	(1)	(2)	(1)	(1)
MD3 to MD0	B'0000		B'001x	B'0010	B
IOC3 to IOC0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	B'xx00	Other than B	
CCLR2 to CCLR0	—	—	—	—	Other than B'101
Output function	—	Output compare output	—	PWM mode 1 output*3	PWM mode 1 output

Note: 3. $TIOCD_0$ output is disabled.
When BFA = 1 or BFB = 1 in TMDR0, output is disable
setting (2) applies.

P11DDR	—	0	1	1	
NDER9	—	—	0	1	
Pin function	$TIOCB_0$ output	$P1_1$ input	$P1_1$ output	PO_9 output	
		TIOCB0 input*1			

Note: 1. $TIOCB_0$ input when MD3 to MD0 = B'0000, and IOB3 to B'10xx.

TPU Channel 0 Setting	(2)	(1)	(2)	(2)	(1)
MD3 to MD0	B'0000		B'0010		B'0011
IOB3 to IOB0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	—	B'xx00	Other th
CCLR2 to CCLR0	—	—	—	—	Other than B'010
Output function	—	Output compare output	—	—	PWM mode 2 output

x

P10DDR	—	0	1	1
NDER8	—	—	0	1
Pin function	$TIOCA_0$ output	$P1_0$ input	$P1_0$ output	PO_8 output
		$TIOCA_0$ input*1		

Note: 1. $TIOCA_0$ input when MD3 to MD0 = B'0000, and IOA3 to
B'10xx.

TPU Channel 0 Setting	(2)	(1)	(2)	(1)	(1)
MD3 to MD0	B'0000		B'001x	B'0010	B
IOA3 to IOA0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	B'xx00	Other than B	
CCLR2 to CCLR0	—	—	—	—	Other than B'001
Output function	—	Output compare output	—	PWM mode 1 output*2	PWM mode output

Note: 2. $TIOCB_0$ output is disabled.

Figure 9.2 shows the port 2 pin configuration.

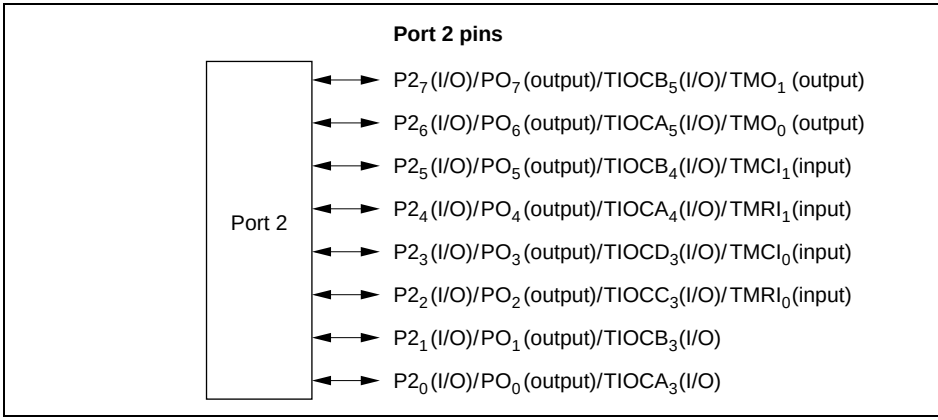


Figure 9.2 Port 2 Pin Functions

9.3.2 Register Configuration

Table 9.4 shows the port 2 register configuration.

Table 9.4 Port 2 Registers

Name	Abbreviation	R/W	Initial Value	Ad
Port 2 data direction register	P2DDR	W	H'00	H'
Port 2 data register	P2DR	R/W	H'00	H'
Port 2 register	PORT2	R	Undefined	H'

Note: * Lower 16 bits of the address.

pins of port 2. P2DDR cannot be read; if it is, an undefined value will be read.

Setting a P2DDR bit to 1 makes the corresponding port 2 pin an output pin, while clea
to 0 makes the pin an input pin.

P2DDR is initialized to H'00 by a power-on reset, and in hardware standby mode. It re
prior state after a manual reset, and in software standby mode. As the PPG, TPU, and
are initialized by a manual reset, the pin states are determined by the P2DDR and P2D
specifications.

Port 2 Data Register (P2DR)

Bit	:	7	6	5	4	3	2	1
		P27DR	P26DR	P25DR	P24DR	P23DR	P22DR	P21DR
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

P2DR is an 8-bit readable/writable register that stores output data for the port 2 pins (

P2DR is initialized to H'00 by a power-on reset, and in hardware standby mode. It reta
state after a manual reset, and in software standby mode.

PORT2 is an 8-bit read-only register that shows the pin states. It cannot be written to. V
output data for the port 2 pins ($P2_7$ to $P2_0$) must always be performed on P2DR.

If a port 2 read is performed while P2DDR bits are set to 1, the P2DR values are read. I
read is performed while P2DDR bits are cleared to 0, the pin states are read.

After a power-on reset and in hardware standby mode, PORT2 contents are determined
states, as P2DDR and P2DR are initialized. PORT2 retains its prior state after a manual
in software standby mode.

$P2_7/PO_7/TIOCB_5/TMO_1$

The pin function is switched as shown below according to the com
the TPU channel 5 setting by bits MD3 to MD0 in TMDR5, bits IOB
TIOR5, bits CCLR1 and CCLR0 in TCR5, bit NDER7 in NDERL, b
OS0 in TCSR1, and bit P27DDR.

OS3 to OS0	All 0			
TPU Channel 5 Setting	Table Below (1)	Table Below (2)		
P27DDR	—	0	1	1
NDER7	—	—	0	1
Pin function	$TIOCB_5$ output	$P2_7$ input	$P2_7$ output	PO_7 output
		$TIOCB_5$ input*		

Note: * $TIOCB_5$ input when MD3 to MD0 = B'0000, B'01xx, and

TPU Channel 5 Setting	(2)	(1)	(2)	(2)	(1)
MD3 to MD0	B'0000, B'01xx		B'0010		B'0011
IOB3 to IOB0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	—	B'xx00	Other t
CCLR1, CCLR0	—	—	—	—	Other than B'1
Output function	—	Output compare output	—	—	PWM mode output

P26DDR	—	0	1	1	
NDER6	—	—	0	1	
Pin function	$TIOCA_5$ output	$P2_6$ input	$P2_6$ output	PO_6 output	
		$TIOCA_5$ input*1			

Note: 1. $TIOCA_5$ input when MD3 to MD0 = B'0000, B'01xx, and

TPU Channel 5 Setting	(2)	(1)	(2)	(1)	(1)
MD3 to MD0	B'0000, B'01xx		B'001x	B'0010	B'
IOA3 to IOA0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	B'xx00	Other than B'	
CCLR1, CCLR0	—	—	—	—	Other than B'01
Output function	—	Output compare output	—	PWM mode 1 output*2	PWM mode 2 output

Note: 2. $TIOCB_5$ output is disabled.

P25DDR	—	0	1
NDER5	—	—	0
Pin function	$TIOCB_4$ output	$P2_5$ input	$P2_5$ output
		$TIOCB_4$ inp	
	$TMCI_1$ input		

Note: 1. $TIOCB_4$ input when MD3 to MD0 = B'0000, B'01xx, and IOB0 = B'10xx.

TPU Channel 4 Setting	(2)	(1)	(2)	(2)	(1)
MD3 to MD0	B'0000, B'01xx		B'0010		B'0011
IOB3 to IOB0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	—	B'xx00	Other t
CCLR1, CCLR0	—	—	—	—	Other than B'1
Output function	—	Output compare output	—	—	PWM mode output

P24DDR	—	0	1
NDER4	—	—	0
Pin function	$TIOCA_4$ output	$P2_4$ input	$P2_4$ output
		$TIOCA_4$ inp	
	$TMRI_1$ input		

Note: 1. $TIOCA_4$ input when MD3 to MD0 = B'0000, B'01xx, and
IOA0 = B'10xx.

TPU Channel 4 Setting	(2)	(1)	(2)	(1)	(1)
MD3 to MD0	B'0000, B'01xx		B'001x	B'0010	B'
IOA3 to IOA0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	B'xx00	Other than B'	
CCLR1, CCLR0	—	—	—	—	Other than B'01
Output function	—	Output compare output	—	PWM mode 1 output*2	PWM mode 2 output

x

Note: 2. $TIOCB_4$ output is disabled.

P23DDR	—	0	1
NDER3	—	—	0
Pin function	$TIOCD_3$ output	$P2_3$ input	$P2_3$ output
		$TIOCD_3$ inp	
	$TMCI_0$ input		

Note: 1. $TIOCD_3$ input when MD3 to MD0 = B'0000, and IOD3 to
B'10xx.

TPU Channel 3 Setting	(2)	(1)	(2)	(2)	(1)
MD3 to MD0	B'0000		B'0010		B'0011
IOD3 to IOD0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	—	B'xx00	Other t
CCLR2 to CCLR0	—	—	—	—	Other than B'110
Output function	—	Output compare output	—	—	PWM mode output

P22DDR	—	0	1
NDER2	—	—	0
Pin function	$TIOCC_3$ output	$P2_2$ input	$P2_2$ output
		$TIOCC_3$ inp	
	$TMRI_0$ input		

Note: 1. $TIOCC_3$ input when MD3 to MD0 = B'0000, and IOC3 to B'10xx.

TPU Channel 3 Setting	(2)	(1)	(2)	(1)	(1)
MD3 to MD0	B'0000		B'001x	B'0010	B'(
IOC3 to IOC0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	B'xx00	Other than B'	
CCLR2 to CCLR0	—	—	—	—	Other than B'101
Output function	—	Output compare output	—	PWM mode 1 output*2	PWM mode 2 output

x

Note: 2. $TIOCD_3$ output is disabled.
When BFA = 1 or BFB = 1 in TMDR3, output is disabled setting (2) applies.

NDER1	—		—	0	
Pin function	$TIOCB_3$ output		$P2_1$ input	$P2_1$ output	
			$TIOCB_3$ inp		

Note: 1. $TIOCB_3$ input when MD3 to MD0 = B'0000, and IOB3 to
B'10xx.

TPU Channel 3 Setting	(2)	(1)	(2)	(2)	(1)
MD3 to MD0	B'0000		B'0010	B'0011	
IOB3 to IOB0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	—	B'xx00	Other t
CCLR2 to CCLR0	—	—	—	—	Other than B'010
Output function	—	Output compare output	—	—	PWM mode output

NDER0	—	—	0
Pin function	$TIOCA_3$ output	$P2_0$ input	$P2_0$ output
		$TIOCA_3$ inpu	

Note: 1. $TIOCA_3$ input when MD3 to MD0 = B'0000, and IOA3 to
B'10xx.

TPU Channel 3 Setting	(2)	(1)	(2)	(1)	(1)
MD3 to MD0	B'0000		B'001x	B'0010	B'(
IOA3 to IOA0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	B'xx00	Other than B'	
CCLR2 to CCLR0	—	—	—	—	Other than B'001
Output function	—	Output compare output	—	PWM mode 1 output*2	PWM mode 2 output

x

Note: 2. $TIOCB_3$ output is disabled.

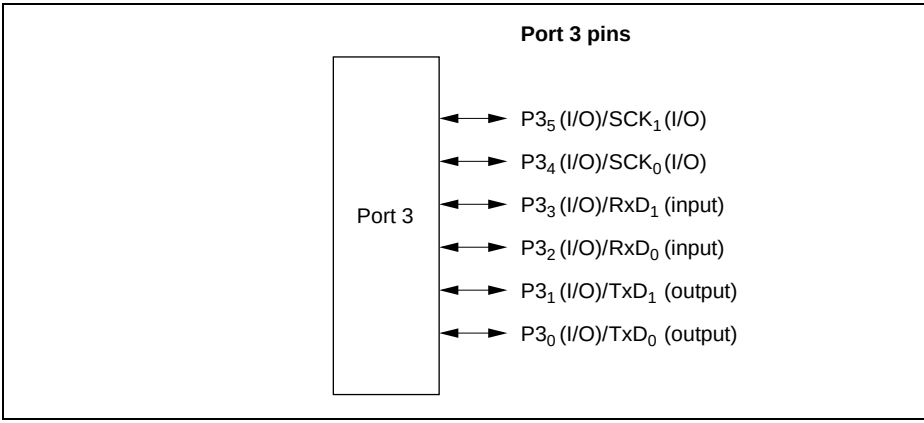


Figure 9.3 Port 3 Pin Functions

9.4.2 Register Configuration

Table 9.6 shows the port 3 register configuration.

Table 9.6 Port 3 Registers

Name	Abbreviation	R/W	Initial Value	A
Port 3 data direction register	P3DDR	W	H'C0	H
Port 3 data register	P3DR	R/W	H'C0	H
Port 3 register	PORT3	R	Undefined	H
Port 3 open drain control register	P3ODR	R/W	H'C0	H

Note: * Lower 16 bits of the address.

pins of port 3. Bits 7 and 6 are reserved. P3DDR cannot be read; if it is, an undefined v
read.

Setting a P3DDR bit to 1 makes the corresponding port 3 pin an output pin, while clear
to 0 makes the pin an input pin.

P3DDR is initialized to H'C0 by a power-on reset, and in hardware standby mode. It re
prior state after a manual reset, and in software standby mode. As the SCI is initialized,
states are determined by the P3DDR and P3DR specifications.

Port 3 Data Register (P3DR)

Bit	:	7	6	5	4	3	2	1
		—	—	P35DR	P34DR	P33DR	P32DR	P31DR
Initial value	:	1	1	0	0	0	0	0
R/W	:	—	—	R/W	R/W	R/W	R/W	R/W

P3DR is an 8-bit readable/writable register that stores output data for the port 3 pins (P

Bits 7 and 6 are reserved; they return an undetermined value if read, and cannot be mod

P3DR is initialized to H'C0 by a power-on reset, and in hardware standby mode. It reta
state after a manual reset, and in software standby mode.

PORT3 is an 8-bit read-only register that shows the pin states. Writing of output data f
pins ($P3_5$ to $P3_0$) must always be performed on P3DR.

Bits 7 and 6 are reserved; they return an undetermined value if read, and cannot be mo

If a port 3 read is performed while P3DDR bits are set to 1, the P3DR values are read.
read is performed while P3DDR bits are cleared to 0, the pin states are read.

After a power-on reset and in hardware standby mode, PORT3 contents are determine
states, as P3DDR and P3DR are initialized. PORT3 retains its prior state after a manu
in software standby mode.

Port 3 Open Drain Control Register (P3ODR)

Bit	:	7	6	5	4	3	2	1
		—	—	P35ODR	P34ODR	P33ODR	P32ODR	P31ODR
Initial value	:	1	1	0	0	0	0	0
R/W	:	—	—	R/W	R/W	R/W	R/W	R/W

P3ODR is an 8-bit readable/writable register that controls the PMOS on/off status for
pin ($P3_5$ to $P3_0$).

Bits 7 and 6 are reserved; they return an undetermined value if read, and cannot be mo

Setting a P3ODR bit to 1 makes the corresponding port 3 pin an NMOS open-drain ou
while clearing the bit to 0 makes the pin a CMOS output pin.

P3ODR is initialized to H'C0 by a power-on reset, and in hardware standby mode. It r
prior state after a manual reset, and in software standby mode.

bit C/$\overline{A}$ in the SCI1 SMR, bits CKE0 and CKE1 in SCR, and bit P35

CKE1	0			
C/$\overline{A}$	0			1
CKE0	0		1	—
P35DDR	0	1	—	—
Pin function	$P3_5$ input pin	$P3_5$ output pin*	SCK_1 output pin*	SCK_1 output pin*

Note: * When P35ODR = 1, the pin becomes an NMOS open-d

$P3_4/SCK_0$

The pin function is switched as shown below according to the comb
bit C/$\overline{A}$ in the SCI0 SMR, bits CKE0 and CKE1 in SCR, and bit P34

CKE1	0			
C/$\overline{A}$	0			1
CKE0	0		1	—
P34DDR	0	1	—	—
Pin function	$P3_4$ input pin	$P3_4$ output pin*	SCK_0 output pin*	SCK_0 output pin*

Note: * When P34ODR = 1, the pin becomes an NMOS open-d

$P3_2/RxD_0$

The pin function is switched as shown below according to the com
bit RE in the SCI0 SCR, and bit P32DDR.

RE	0		
P32DDR	0	1	
Pin function	$P3_2$ input pin	$P3_2$ output pin*	$RxD_$

Note: * When P32ODR = 1, the pin becomes an NMOS open-

$P3_1/TxD_1$

The pin function is switched as shown below according to the com
bit TE in the SCI1 SCR, and bit P31DDR.

TE	0		
P31DDR	0	1	
Pin function	$P3_1$ input pin	$P3_1$ output pin*	TxD_1

Note: * When P31ODR = 1, the pin becomes an NMOS open-

$P3_0/TxD_0$

The pin function is switched as shown below according to the com
bit TE in the SCI0 SCR, and bit P30DDR.

TE	0		
P30DDR	0	1	
Pin function	$P3_0$ input pin	$P3_0$ output pin*	TxD_0

Note: * When P30ODR = 1, the pin becomes an NMOS open-

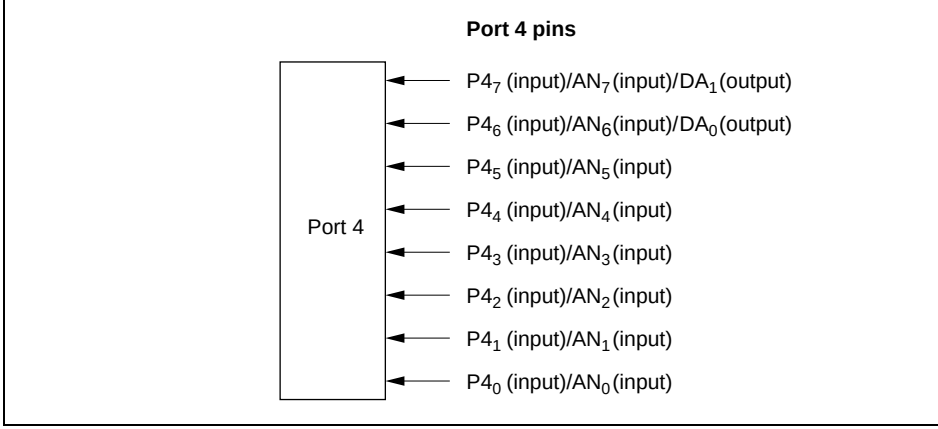


Figure 9.4 Port 4 Pin Functions

9.5.2 Register Configuration

Table 9.8 shows the port 4 register configuration. Port 4 is an input-only port, and does
data direction register or data register.

Table 9.8 Port 4 Registers

Name	Abbreviation	R/W	Initial Value	A
Port 4 register	PORT4	R	Undefined	H'

Note: * Lower 16 bits of the address.

Note: * Determined by state of pins $P4_7$ to $P4_0$.

9.5.3 Pin Functions

Port 4 pins also function as A/D converter analog input pins (AN_0 to AN_7) and D/A co
analog output pins (DA_0 and DA_1).

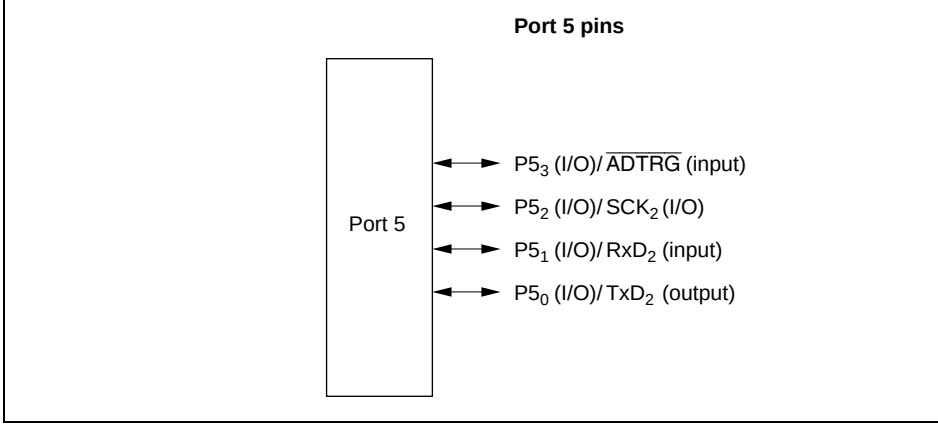


Figure 9.5 Port 5 Pin Functions

9.6.2 Register Configuration

Table 9.9 shows the port 5 register configuration.

Table 9.9 Port 5 Registers

Name	Abbreviation	R/W	Initial Value	Ad
Port 5 data direction register	P5DDR	W	H'F0	H'
Port 5 data register	P5DR	R/W	H'F0	H'
Port 5 register	PORT5	R	Undefined	H'

Note: * Lower 16 bits of the address.

pins of port 5. Bits 7 to 4 are reserved. P5DDR cannot be read; if it is, an undefined va
read.

Setting a P5DDR bit to 1 makes the corresponding port 5 pin an output pin, while clea
to 0 makes the pin an input pin.

P5DDR is initialized to H'F0 by a power-on reset, and in hardware standby mode. It re
prior state after a manual reset, and in software standby mode. As the SCI is initialized
states are determined by the P5DDR and P5DR specifications.

Port 5 Data Register (P5DR)

Bit	:	7	6	5	4	3	2	1
		—	—	—	—	P53DR	P52DR	P51DR
Initial value	:	1	1	1	1	0	0	0
R/W	:	—	—	—	—	R/W	R/W	R/W

P5DR is an 8-bit readable/writable register that stores output data for the port 5 pins (

Bits 7 to 4 are reserved; they return an undetermined value if read, and cannot be mod

P5DR is initialized to H'F0 by a power-on reset, and in hardware standby mode. It reta
state after a manual reset, and in software standby mode.

PORT5 is an 8-bit read-only register that shows the pin states. It cannot be written to. W
output data for the port 5 pins ($P5_3$ to $P5_0$) must always be performed on P5DR.

Bits 7 to 4 are reserved; they return an undetermined value if read, and cannot be modif

If a port 5 read is performed while P5DDR bits are set to 1, the P5DR values are read. I
read is performed while P5DDR bits are cleared to 0, the pin states are read.

After a power-on reset and in hardware standby mode, PORT5 contents are determined
states, as P5DDR and P5DR are initialized. PORT5 retains its prior state after a manual
in software standby mode.

bits TRGS1 and TRGS0 in the A/D converter ADCR, and bit P53D

P53DDR	0	1
Pin function	$P5_3$ input pin	$P5_3$ outpu
	$\overline{\text{ADTRG}}$ input pin*	

Note: * $\overline{\text{ADTRG}}$ input when TRGS0 = TRGS1 = 1.

$P5_2/SCK_2$

The pin function is switched as shown below according to the com
bit C/$\overline{\text{A}}$ in the SCI2 SMR, bits CKE0 and CKE1 in SCR, and bit P5

CKE1	0			
C/$\overline{\text{A}}$	0			1
CKE0	0		1	—
P52DDR	0	1	—	—
Pin function	$P5_2$ input pin	$P5_2$ output pin	SCK_2 output pin	SCK_2 output pin

$P5_1/RxD_2$

The pin function is switched as shown below according to the com
bit RE in the SCI2 SCR, and bit P51DDR.

RE	0		
P51DDR	0	1	
Pin function	$P5_1$ input pin	$P5_1$ output pin	RxD_2

$P5_0/TxD_2$

The pin function is switched as shown below according to the com
bit TE in the SCI2 SCR, and bit P50DDR.

TE	0		
P50DDR	0	1	
Pin function	$P5_0$ input pin	$P5_0$ output pin	TxD_2

inputs. Figure 9.6 shows the port 6 pin configuration.

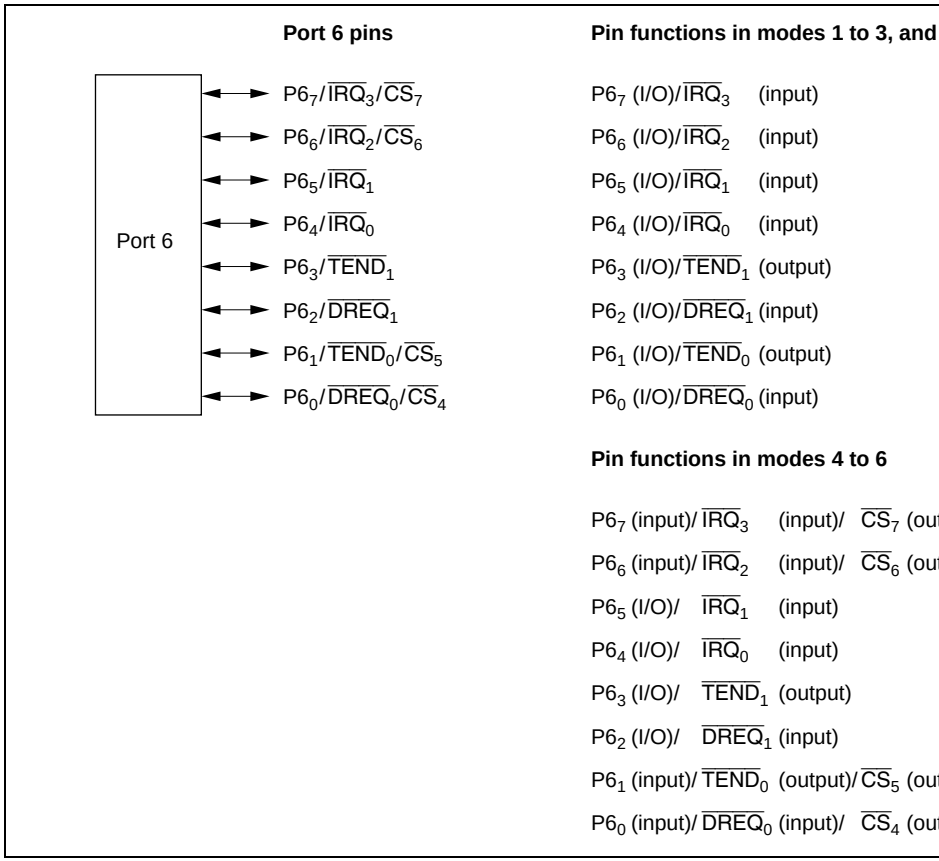


Figure 9.6 Port 6 Pin Functions

Port 6 register	PORT6	R	Undefined

Note: * Lower 16 bits of the address.

Port 6 Data Direction Register (P6DDR)

Bit	:	7	6	5	4	3	2	1
		P67DDR	P66DDR	P65DDR	P64DDR	P63DDR	P62DDR	P61DDR
Initial value	:	0	0	0	0	0	0	0
R/W	:	W	W	W	W	W	W	W

P6DDR is an 8-bit write-only register, the individual bits of which specify input or ou pins of port 6. P6DDR cannot be read; if it is, an undefined value will be read.

Setting a P6DDR bit to 1 makes the corresponding port 6 pin an output pin, while clea to 0 makes the pin an input pin.

P6DDR is initialized to H'00 by a power-on reset, and in hardware standby mode. It re prior state after a manual reset, and in software standby mode. As the DMAC is initial manual reset, the pin states are determined by the P6DDR and P6DR specifications.

P6DR is initialized to H'00 by a power-on reset, and in hardware standby mode. It retai
state after a manual reset, and in software standby mode.

Port 6 Register (PORT6)

Bit	:	7	6	5	4	3	2	1
		P67	P66	P65	P64	P63	P62	P61
Initial value	:	—*	—*	—*	—*	—*	—*	—*
RW	:	R	R	R	R	R	R	R

Note: * Determined by state of pins $P6_7$ to $P6_0$.

PORT6 is an 8-bit read-only register that shows the pin states. It cannot be written to. W
output data for the port 6 pins ($P6_7$ to $P6_0$) must always be performed on P6DR.

If a port 6 read is performed while P6DDR bits are set to 1, the P6DR values are read. I
read is performed while P6DDR bits are cleared to 0, the pin states are read.

After a power-on reset and in hardware standby mode, PORT6 contents are determined
states, as P6DDR and P6DR are initialized. PORT6 retains its prior state after a manual
in software standby mode.

$P6_7/\overline{IRQ}_3/\overline{CS}_7$ The pin function is switched as shown below according to bit P67D

Mode	Modes 1 to 3, 7		Modes 4	
P67DDR	0	1	0	
Pin function	$P6_7$ input pin	$P6_7$ output pin	$P6_7$ input pin	$\overline{CS}$
	$\overline{IRQ}_3$ interrupt input pin			

$P6_6/\overline{IRQ}_2/\overline{CS}_6$ The pin function is switched as shown below according to bit P66D

Mode	Modes 1 to 3, 7		Modes 4	
P66DDR	0	1	0	
Pin function	$P6_6$ input pin	$P6_6$ output pin	$P6_6$ input pin	$\overline{CS}$
	$\overline{IRQ}_2$ interrupt input pin			

$P6_5/\overline{IRQ}_1$ The pin function is switched as shown below according to bit P65D

P65DDR	0	1
Pin function	$P6_5$ input pin	$P6_5$ outpu
	$\overline{IRQ}_1$ interrupt input pin	

$P6_4/\overline{IRQ}_0$ The pin function is switched as shown below according to bit P64D

P64DDR	0	1
Pin function	$P6_4$ input pin	$P6_4$ outpu
	$\overline{IRQ}_0$ interrupt input pin	

$P6_2/\overline{DREQ}_1$ The pin function is switched as shown below according to bit P62D

P62DDR	0	1
Pin function	$P6_2$ input pin	$P6_2$ output
	$\overline{DERQ}_1$ input	

$P6_1/\overline{TEND}_0/\overline{CS}_5$ The pin function is switched as shown below according to the comb bit TEE0 in the DMAC DMATCR, and bit P61DDR.

Modes 1 to 3, 7

TEE0	0		
P61DDR	0	1	
Pin function	$P6_1$ input pin	$P6_1$ output pin	$\overline{TEND}$

Modes 4 to 6

TEE0	0		
P61DDR	0	1	
Pin function	$P6_1$ input pin	$\overline{CS}_5$ output pin	$\overline{TEND}$

$P6_0/\overline{DREQ}_0/\overline{CS}_4$ The pin function is switched as shown below according to bit P60D

Mode	Modes 1 to 3, 7		Modes 4 to	
P60DDR	0	1	0	
Pin function	$P6_0$ input pin	$P6_0$ output pin	$P6_0$ input pin	$\overline{CS}$
	$\overline{DREQ}_0$ input			

PA_4 are schmitt-triggered inputs.

Figure 9.7 shows the port A pin configuration.

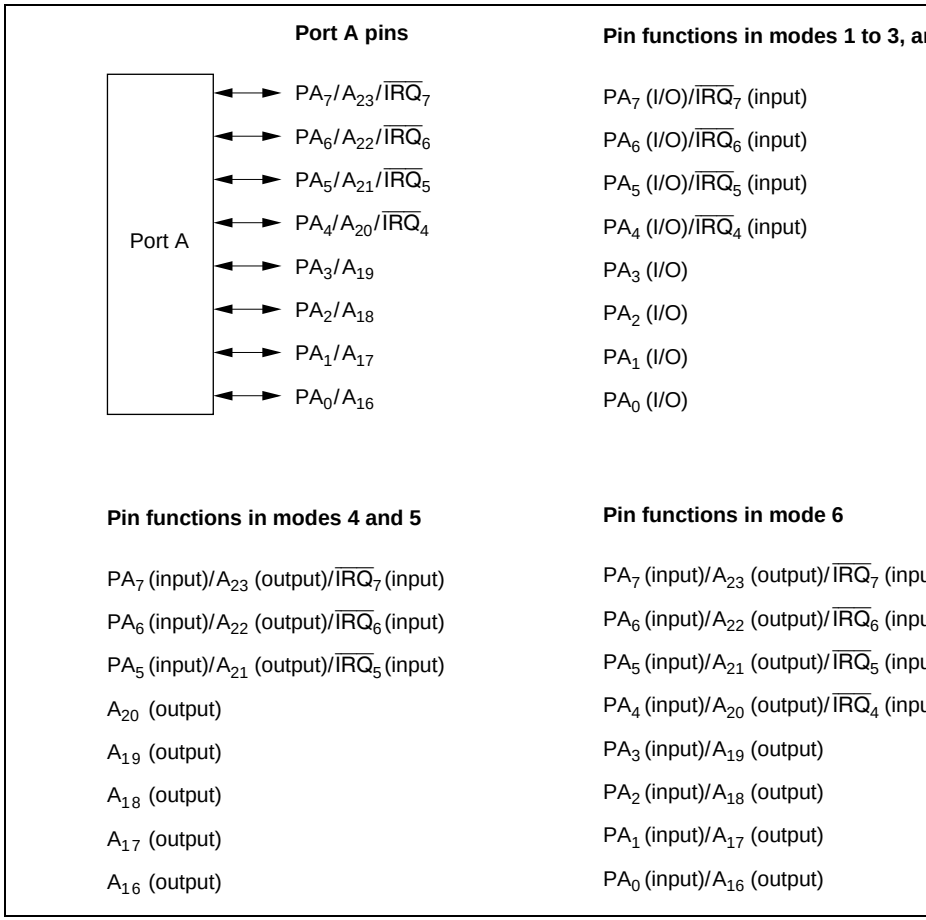


Figure 9.7 Port A Pin Functions

Port A register	PORTA	R	Undefined	H'
Port A MOS pull-up control register	PAPCR	R/W	H'00	H'
Port A open-drain control register	PAODR	R/W	H'00	H'

Note: * Lower 16 bits of the address.

Port A Data Direction Register (PADDR)

Bit	:	7	6	5	4	3	2	1
		PA7DDR	PA6DDR	PA5DDR	PA4DDR	PA3DDR	PA2DDR	PA1DDR
Initial value	:	0	0	0	0	0	0	0
R/W	:	W	W	W	W	W	W	W

PADDR is an 8-bit write-only register, the individual bits of which specify input or out pins of port A. PADDR cannot be read; if it is, an undefined value will be read.

PADDR is initialized to H'00 by a power-on reset and in hardware standby mode. It ret prior state after a manual reset, and in software standby mode. The OPE bit in SBYCR select whether the address output pins retain their output state or become high-impedan transition is made to software standby mode.

- Modes 1 to 3, and 7

  Setting a PADDR bit to 1 makes the corresponding port A pin an output port, while the bit to 0 makes the pin an input port.

- Modes 4 and 5

  The corresponding port A pins are address outputs irrespective of the value of bits P PA0DDR.

  Setting one of bits PA7DDR to PA5DDR to 1 makes the corresponding port A pin a output, while clearing the bit to 0 makes the pin an input port.

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

PADR is an 8-bit readable/writable register that stores output data for the port A pins (

PADR is initialized to H'00 by a power-on reset, and in hardware standby mode. It ret
state after a manual reset, and in software standby mode.

Port A Register (PORTA)

Bit	:	7	6	5	4	3	2	1
		PA7	PA6	PA5	PA4	PA3	PA2	PA1
Initial value	:	—*	—*	—*	—*	—*	—*	—*
R/W	:	R	R	R	R	R	R	R

Note: * Determined by state of pins PA_7 to PA_0.

PORTA is an 8-bit read-only register that shows the pin states. It cannot be written to.
output data for the port A pins (PA_7 to PA_0) must always be performed on PADR.

If a port A read is performed while PADDR bits are set to 1, the PADR values are rea
read is performed while PADDR bits are cleared to 0, the pin states are read.

After a power-on reset and in hardware standby mode, PORTA contents are determine
states, as PADDR and PADR are initialized. PORTA retains its prior state after a man
and in software standby mode.

incorporated into port A on an individual bit basis.

All the bits are valid in modes 1 to 3, 6, and 7, and bits 7 to 5 are valid in modes 4 and 5
PADDR bit is cleared to 0 (input port setting), setting the corresponding PAPCR bit to 1
the MOS input pull-up for the corresponding pin.

PAPCR is initialized to H'00 by a power-on reset, and in hardware standby mode. It ret
prior state after a manual reset, and in software standby mode.

Port A Open Drain Control Register (PAODR)

Bit	:	7	6	5	4	3	2	1	
		PA7ODR	PA6ODR	PA5ODR	PA4ODR	PA3ODR	PA2ODR	PA1ODR	P
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	

PAODR is an 8-bit readable/writable register that controls whether PMOS is on or off f
port A pin (PA_7 to PA_0).

All bits are valid in modes 1 to 3, and 7.

Setting a PAODR bit to 1 makes the corresponding port A pin an NMOS open-drain ou
clearing the bit to 0 makes the pin a CMOS output.

PAODR is initialized to H'00 by a power-on reset, and in hardware standby mode. It re
prior state after a manual reset, and in software standby mode.

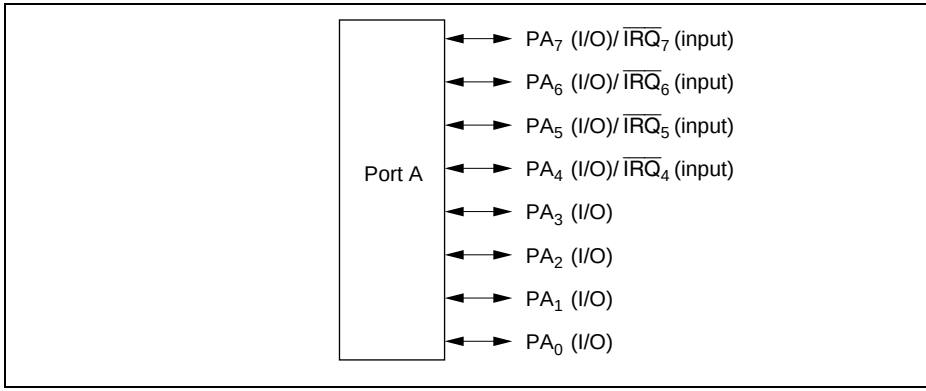


Figure 9.8 Port A Pin Functions (Modes 1 to 3, and 7)

Modes 4 and 5

In modes 4 and 5, the lower 5 bits of port A are designated as address outputs automat
the upper 3 bits function as address outputs or input ports and interrupt input pins. Inp
can be specified individually for the upper 3 bits. Setting one of bits PA7DDR to PA5
makes the corresponding port A pin an address output, while clearing the bit to 0 mak
input port.

Port A pin functions in modes 4 and 5 are shown in figure 9.9.

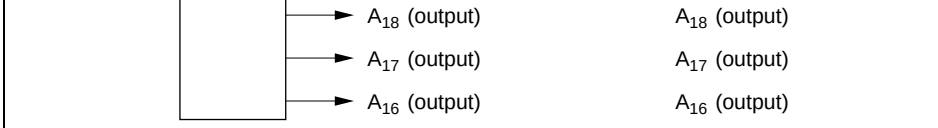


Figure 9.9 Port A Pin Functions (Modes 4 and 5)

Mode 6

In mode 6, port A pins function as address outputs or input ports and interrupt input pin
output can be specified on an individual bit basis. Setting a PADDR bit to 1 makes the
corresponding port A pin an address output, while clearing the bit to 0 makes the pin an

Port A pin functions in mode 6 are shown in figure 9.10.

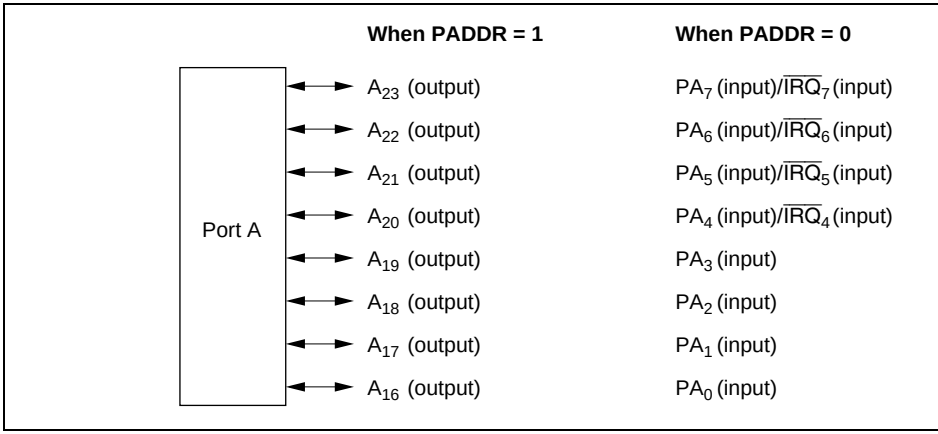


Figure 9.10 Port A Pin Functions (Mode 6)

The MOS input pull-up function is in the off state after a power-on reset, and in hardw
mode. The prior state is retained after a manual reset, and in software standby mode.

Table 9.14 summarizes the MOS input pull-up states.

Table 9.14 MOS Input Pull-Up States (Port A)

Modes		Power-On Reset	Hardware Standby Mode	Manual Reset	Software Standby Mode
1 to 3, 6, 7	PA_7 to PA_0	OFF		ON/OFF	
4, 5	PA_7 to PA_5			ON/OFF	
	PA_4 to PA_0			OFF	

Legend:
OFF: MOS input pull-up is always off.
ON/OFF: On when PADDR = 0 and PAPCR = 1; otherwise off.

Figure 9.11 shows the port B pin configuration.

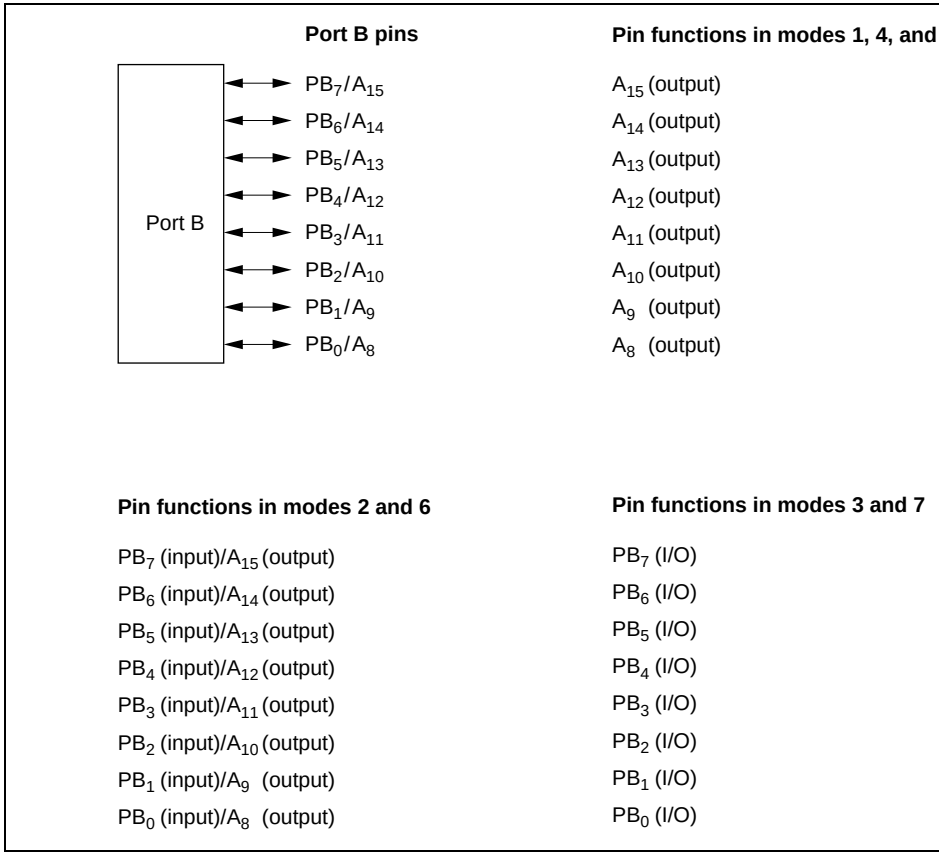


Figure 9.11 Port B Pin Functions

Port B register	PORTB	R	Undefined	H
Port B MOS pull-up control register	PBPCR	R/W	H'00	H

Note: * Lower 16 bits of the address.

Port B Data Direction Register (PBDDR)

Bit	:	7	6	5	4	3	2	1
		PB7DDR	PB6DDR	PB5DDR	PB4DDR	PB3DDR	PB2DDR	PB1DDR
Initial value	:	0	0	0	0	0	0	0
R/W	:	W	W	W	W	W	W	W

PBDDR is an 8-bit write-only register, the individual bits of which specify input or ou pins of port B. PBDDR cannot be read; if it is, an undefined value will be read.

PBDDR is initialized to H'00 by a power-on reset and in hardware standby mode. It re prior state after a manual reset, and in software standby mode. The OPE bit in SBYCR select whether the address output pins retain their output state or become high-impeda transition is made to software standby mode.

- Modes 1, 4, and 5

  The corresponding port B pins are address outputs irrespective of the value of the

- Modes 2 and 6

  Setting a PBDDR bit to 1 makes the corresponding port B pin an address output, clearing the bit to 0 makes the pin an input port.

- Modes 3 and 7

  Setting a PBDDR bit to 1 makes the corresponding port B pin an output port, whil the bit to 0 makes the pin an input port.

PBDR is initialized to H'00 by a power-on reset, and in hardware standby mode. It reta
state after a manual reset, and in software standby mode.

Port B Register (PORTB)

Bit	:	7	6	5	4	3	2	1
		PB7	PB6	PB5	PB4	PB3	PB2	PB1
Initial value	:	—*	—*	—*	—*	—*	—*	—*
R/W	:	R	R	R	R	R	R	R

Note: * Determined by state of pins PB_7 to PB_0.

PORTB is an 8-bit read-only register that shows the pin states. It cannot be written to. V
output data for the port B pins (PB_7 to PB_0) must always be performed on PBDR.

If a port B read is performed while PBDDR bits are set to 1, the PBDR values are read.
read is performed while PBDDR bits are cleared to 0, the pin states are read.

After a power-on reset and in hardware standby mode, PORTB contents are determined
states, as PBDDR and PBDR are initialized. PORTB retains its prior state after a manu
in software standby mode.

incorporated into port B on an individual bit basis.

When a PBDDR bit is cleared to 0 (input port setting) in mode 2, 3, 6, or 7, setting the corresponding PBPCR bit to 1 turns on the MOS input pull-up for the corresponding p

PBPCR is initialized to H'00 by a power-on reset, and in hardware standby mode. It re prior state after a manual reset, and in software standby mode.

9.9.3 Pin Functions

Modes 1, 4, and 5

In modes 1, 4, and 5, port B pins are automatically designated as address outputs.

Port B pin functions in modes 1, 4, and 5 are shown in figure 9.12.

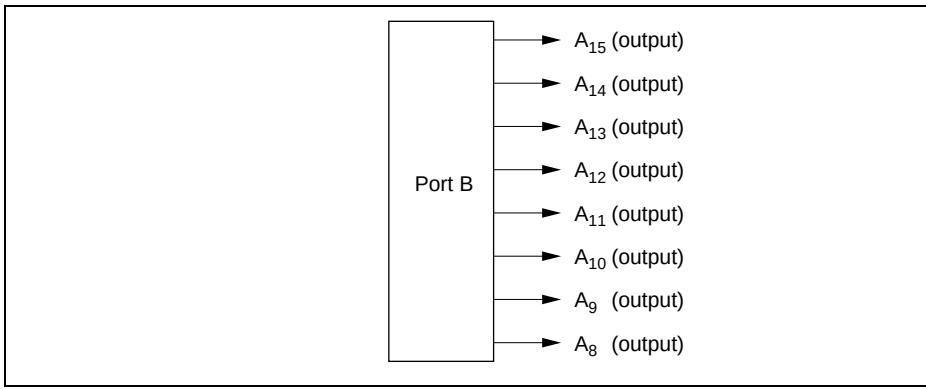


Figure 9.12 Port B Pin Functions (Modes 1, 4, and 5)

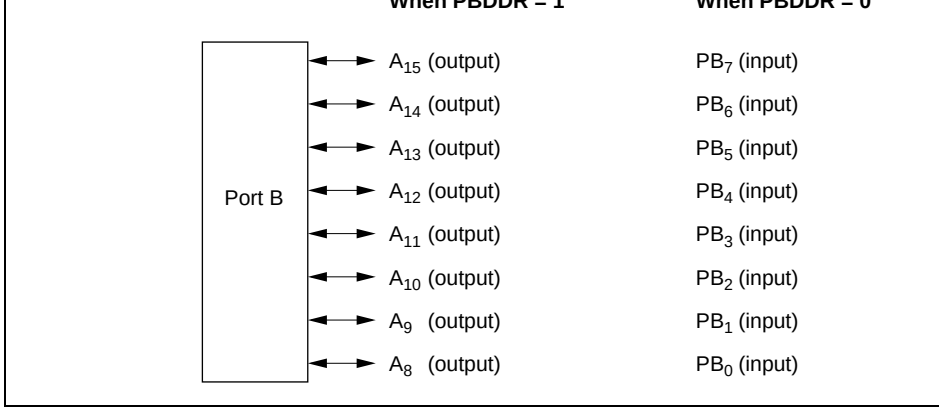


Figure 9.13 Port B Pin Functions (Modes 2 and 6)

Modes 3 and 7

In modes 3 and 7, port B pins function as I/O ports. Input or output can be specified for on an individual bit basis. Setting a PBDDR bit to 1 makes the corresponding port B pi port, while clearing the bit to 0 makes the pin an input port.

Port B pin functions in modes 3 and 7 are shown in figure 9.14.

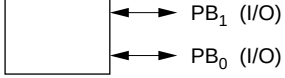


Figure 9.14 Port B Pin Functions (Modes 3 and 7)

9.9.4 MOS Input Pull-Up Function

Port B has a built-in MOS input pull-up function that can be controlled by software. T
input pull-up function can be used in modes 2, 3, 6, and 7, and can be specified as on
individual bit basis.

When a PBDDR bit is cleared to 0 in mode 2, 3, 6, or 7, setting the corresponding PB
turns on the MOS input pull-up for that pin.

The MOS input pull-up function is in the off state after a power-on reset, and in hardw
mode. The prior state is retained after a manual reset, and in software standby mode.

Table 9.16 summarizes the MOS input pull-up states.

Table 9.16 MOS Input Pull-Up States (Port B)

Modes	Power-On Reset	Hardware Standby Mode	Manual Reset	Software Standby Mode
1, 4, 5	OFF		OFF	
2, 3, 6, 7			ON/OFF	

Legend:
OFF: MOS input pull-up is always off.
ON/OFF: On when PBDDR = 0 and PBPCR = 1; otherwise off.

Figure 9.15 shows the port C pin configuration.

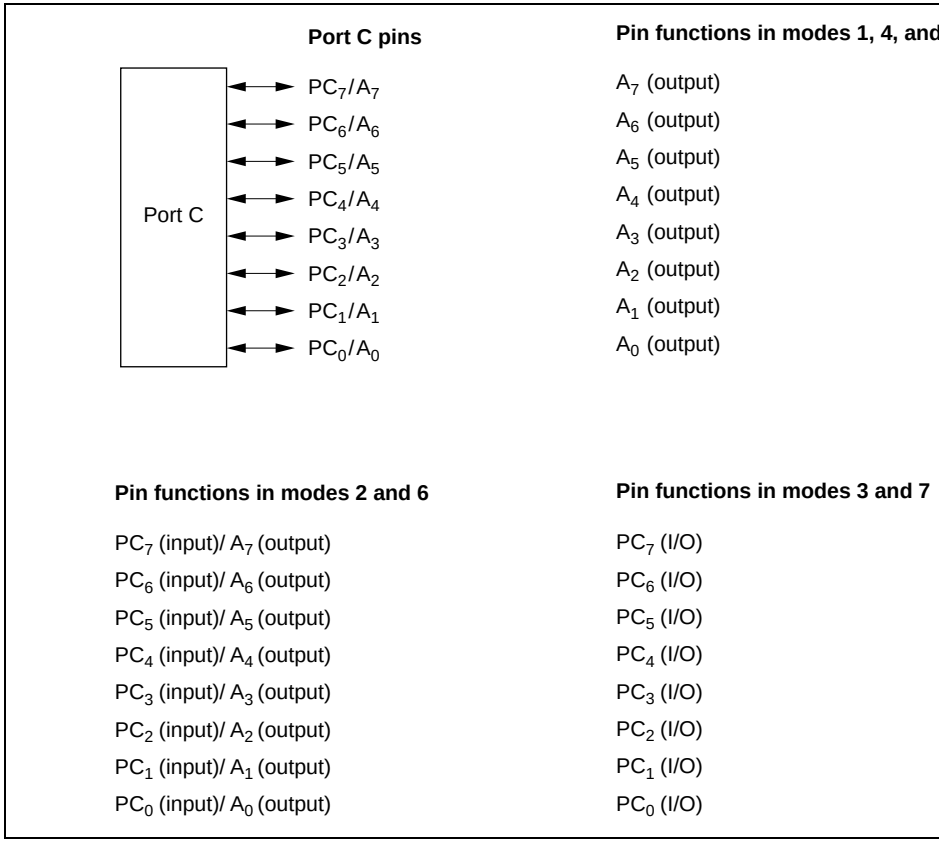


Figure 9.15 Port C Pin Functions

Port C register	PORTC	R	Undefined	H
Port C MOS pull-up control register	PCPCR	R/W	H'00	H

Note: * Lower 16 bits of the address.

Port C Data Direction Register (PCDDR)

Bit	:	7	6	5	4	3	2	1
		PC7DDR	PC6DDR	PC5DDR	PC4DDR	PC3DDR	PC2DDR	PC1DD
Initial value	:	0	0	0	0	0	0	0
R/W	:	W	W	W	W	W	W	W

PCDDR is an 8-bit write-only register, the individual bits of which specify input or ou pins of port C. PCDDR cannot be read; if it is, an undefined value will be read.

PCDDR is initialized to H'00 by a power-on reset and in hardware standby mode. It re prior state after a manual reset, and in software standby mode. The OPE bit in SBYCR select whether the address output pins retain their output state or become high-impeda transition is made to software standby mode.

- Modes 1, 4, and 5

  The corresponding port C pins are address outputs irrespective of the value of the

- Modes 2 and 6

  Setting a PCDDR bit to 1 makes the corresponding port C pin an address output, clearing the bit to 0 makes the pin an input port.

- Modes 3 and 7

  Setting a PCDDR bit to 1 makes the corresponding port C pin an output port, whil the bit to 0 makes the pin an input port.

PCDR is initialized to H'00 by a power-on reset, and in hardware standby mode. It reta
state after a manual reset, and in software standby mode.

Port C Register (PORTC)

Bit	:	7	6	5	4	3	2	1
		PC7	PC6	PC5	PC4	PC3	PC2	PC1
Initial value	:	—*	—*	—*	—*	—*	—*	—*
R/W	:	R	R	R	R	R	R	R

Note: * Determined by state of pins PC_7 to PC_0.

PORTC is an 8-bit read-only register that shows the pin states. It cannot be written to. V
output data for the port C pins (PC_7 to PC_0) must always be performed on PCDR.

If a port C read is performed while PCDDR bits are set to 1, the PCDR values are read.
read is performed while PCDDR bits are cleared to 0, the pin states are read.

After a power-on reset and in hardware standby mode, PORTC contents are determined
states, as PCDDR and PCDR are initialized. PORTC retains its prior state after a manu
in software standby mode.

incorporated into port C on an individual bit basis.

When a PCDDR bit is cleared to 0 (input port setting) in mode 2, 3, 6, or 7, setting the corresponding PCPCR bit to 1 turns on the MOS input pull-up for the corresponding p

PCPCR is initialized to H'00 by a power-on reset, and in hardware standby mode. It re prior state after a manual reset, and in software standby mode.

9.10.3 Pin Functions

Modes 1, 4, and 5

In modes 1, 4, and 5, port C pins are automatically designated as address outputs.

Port C pin functions in modes 1, 4, and 5 are shown in figure 9.16.

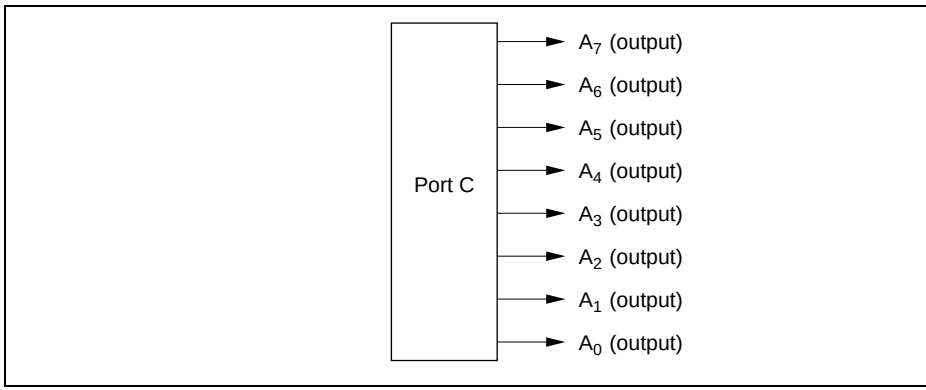


Figure 9.16 Port C Pin Functions (Modes 1, 4, and 5)

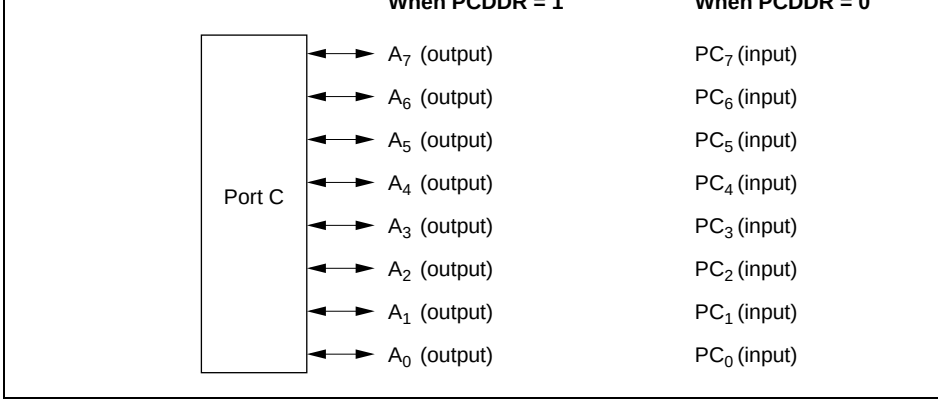


Figure 9.17 Port C Pin Functions (Modes 2 and 6)

Modes 3 and 7

In modes 3 and 7, port C pins function as I/O ports. Input or output can be specified for on an individual bit basis. Setting a PCDDR bit to 1 makes the corresponding port C pi port, while clearing the bit to 0 makes the pin an input port.

Port C pin functions in modes 3 and 7 are shown in figure 9.18.

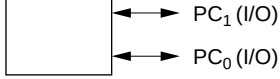


Figure 9.18 Port C Pin Functions (Modes 3 and 7)

9.10.4 MOS Input Pull-Up Function

Port C has a built-in MOS input pull-up function that can be controlled by software. T
input pull-up function can be used in modes 2, 3, 6, and 7, and can be specified as on
individual bit basis.

When a PCDDR bit is cleared to 0 in mode 2, 3, 6, or 7, setting the corresponding PC
turns on the MOS input pull-up for that pin.

The MOS input pull-up function is in the off state after a power-on reset, and in hardw
mode. The prior state is retained after a manual reset, and in software standby mode.

Table 9.18 summarizes the MOS input pull-up states.

Table 9.18 MOS Input Pull-Up States (Port C)

Modes	Power-On Reset	Hardware Standby Mode	Manual Reset	Software Standby Mode
1, 4, 5	OFF		OFF	
2, 3, 6, 7			ON/OFF	

Legend:
OFF: MOS input pull-up is always off.
ON/OFF: On when PCDDR = 0 and PCPCR = 1; otherwise off.

Figure 9.19 shows the port D pin configuration.

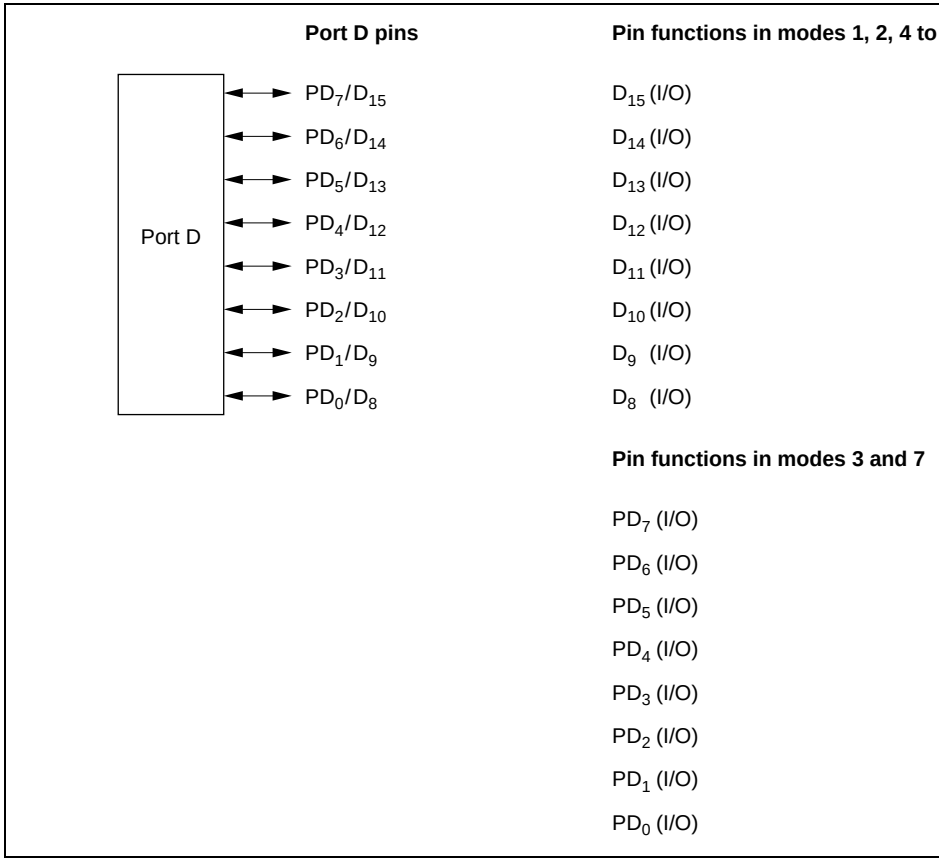


Figure 9.19 Port D Pin Functions

Port D register	PORTD	R	Undefined	H
Port D MOS pull-up control register	PDPCR	R/W	H'00	H

Note: * Lower 16 bits of the address.

Port D Data Direction Register (PDDDR)

Bit	:	7	6	5	4	3	2	1
		PD7DDR	PD6DDR	PD5DDR	PD4DDR	PD3DDR	PD2DDR	PD1DDR
Initial value	:	0	0	0	0	0	0	0
R/W	:	W	W	W	W	W	W	W

PDDDR is an 8-bit write-only register, the individual bits of which specify input or ou pins of port D. PDDDR cannot be read; if it is, an undefined value will be read..

PDDDR is initialized to H'00 by a power-on reset, and in hardware standby mode. It r prior state after a manual reset, and in software standby mode.

- Modes 1, 2, 4 to 6

  The input/output direction specification by PDDDR is ignored, and port D is autor designated for data I/O.

- Modes 3 and 7

  Setting a PDDDR bit to 1 makes the corresponding port D pin an output port, whil the bit to 0 makes the pin an input port.

PDDR is initialized to H'00 by a power-on reset, and in hardware standby mode. It reta
state after a manual reset, and in software standby mode.

Port D Register (PORTD)

Bit	:	7	6	5	4	3	2	1
		PD7	PD6	PD5	PD4	PD3	PD2	PD1
Initial value	:	—*	—*	—*	—*	—*	—*	—*
R/W	:	R	R	R	R	R	R	R

Note: * Determined by state of pins PD_7 to PD_0.

PORTD is an 8-bit read-only register that shows the pin states. It cannot be written to. V
output data for the port D pins (PD_7 to PD_0) must always be performed on PDDR.

If a port D read is performed while PDDDR bits are set to 1, the PDDR values are read
read is performed while PDDDR bits are cleared to 0, the pin states are read.

After a power-on reset and in hardware standby mode, PORTD contents are determined
states, as PDDDR and PDDR are initialized. PORTD retains its prior state after a manu
and in software standby mode.

incorporated into port D on an individual bit basis.

When a PDDDR bit is cleared to 0 (input port setting) in mode 3 or 7, setting the corre
PDPCR bit to 1 turns on the MOS input pull-up for the corresponding pin.

PDPCR is initialized to H'00 by a power-on reset, and in hardware standby mode. It re
prior state after a manual reset, and in software standby mode.

9.11.3 Pin Functions

Modes 1, 2, 4 to 6

In modes 1, 2, 4 to 6, port D pins are automatically designated as data I/O pins.

Port D pin functions in modes 1, 2, 4 to 6 are shown in figure 9.20.

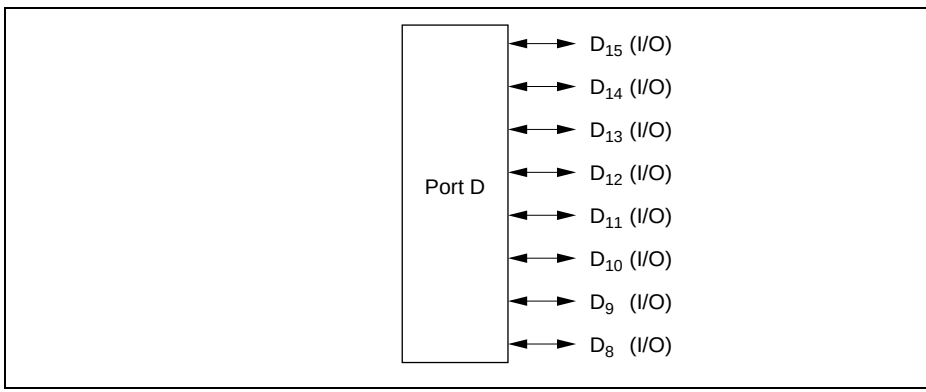


Figure 9.20 Port D Pin Functions (Modes 1, 2, 4 to 6)

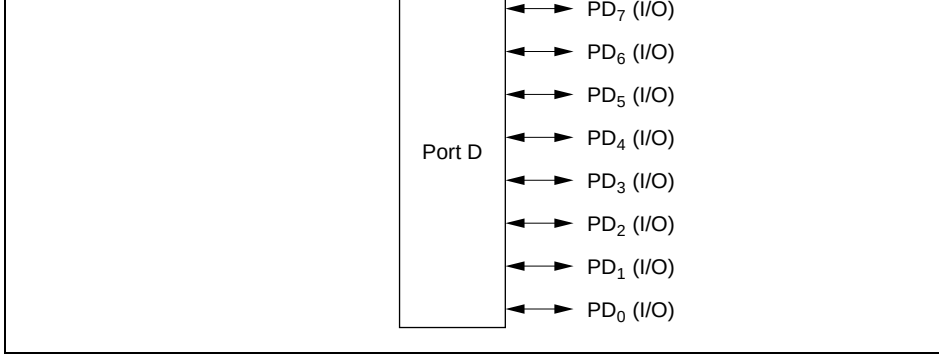


Figure 9.21 Port D Pin Functions (Modes 3 and 7)

The MOS input pull-up function is in the off state after a power-on reset, and in hardw
mode. The prior state is retained after a manual reset, and in software standby mode.

Table 9.20 summarizes the MOS input pull-up states.

Table 9.20 MOS Input Pull-Up States (Port D)

Modes	Power-On Reset	Hardware Standby Mode	Manual Reset	Software Standby Mode
1, 2, 4 to 6	OFF		OFF	
3, 7			ON/OFF	

Legend:
OFF: MOS input pull-up is always off.
ON/OFF: On when PDDDR = 0 and PDPCR = 1; otherwise off.

Figure 9.22 shows the port E pin configuration.

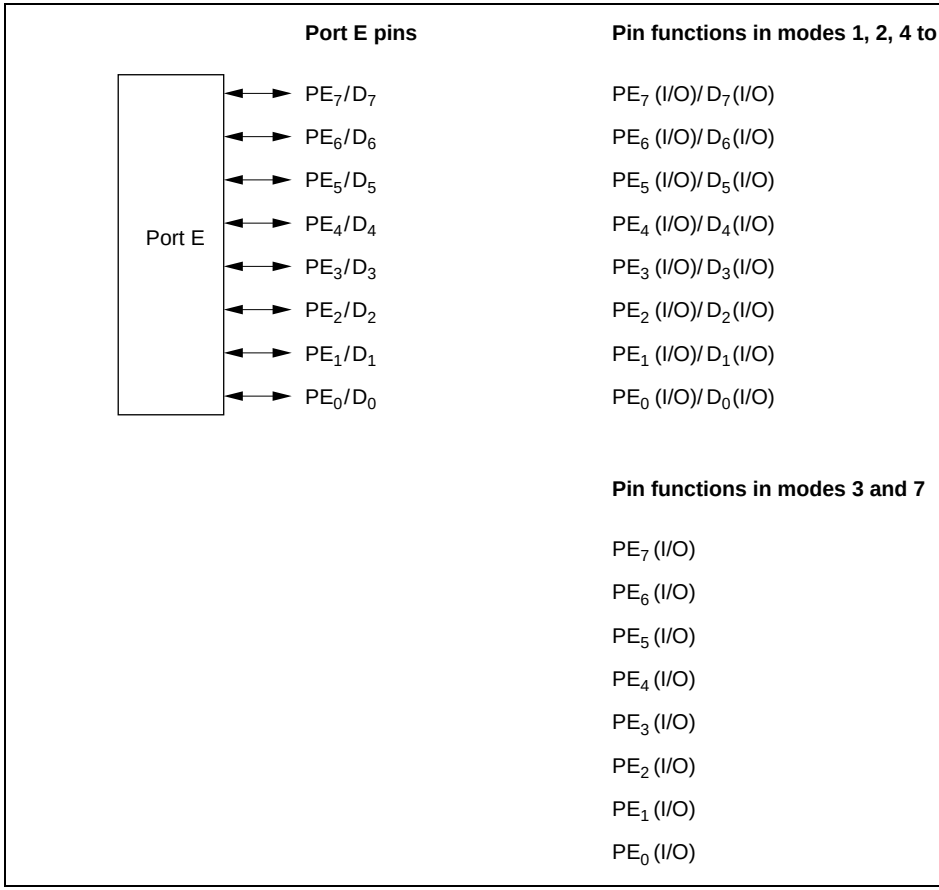


Figure 9.22 Port E Pin Functions

Port E register	PORTE	R	Undefined	H
Port E MOS pull-up control register	PEPCR	R/W	H'00	H

Note: * Lower 16 bits of the address.

Port E Data Direction Register (PEDDR)

Bit	:	7	6	5	4	3	2	1
		PE7DDR	PE6DDR	PE5DDR	PE4DDR	PE3DDR	PE2DDR	PE1DDR
Initial value	:	0	0	0	0	0	0	0
R/W	:	W	W	W	W	W	W	W

PEDDR is an 8-bit write-only register, the individual bits of which specify input or ou pins of port E. PEDDR cannot be read; if it is, an undefined value will be read.

PEDDR is initialized to H'00 by a power-on reset, and in hardware standby mode. It r prior state after a manual reset, and in software standby mode.

- Modes 1, 2, 4 to 6

  When 8-bit bus mode has been selected, port E pins function as I/O ports. Setting a to 1 makes the corresponding port E pin an output port, while clearing the bit to 0 pin an input port.

  When 16-bit bus mode has been selected, the input/output direction specification b ignored, and port E is designated for data I/O.

  For details of 8-bit and 16-bit bus modes, see section 6, Bus Controller.

- Modes 3 and 7

  Setting a PEDDR bit to 1 makes the corresponding port E pin an output port, while bit to 0 makes the pin an input port.

PEDR is initialized to H'00 by a power-on reset, and in hardware standby mode. It retai
state after a manual reset, and in software standby mode.

Port E Register (PORTE)

Bit	:	7	6	5	4	3	2	1
		PE7	PE6	PE5	PE4	PE3	PE2	PE1
Initial value	:	—*	—*	—*	—*	—*	—*	—*
R/W	:	R	R	R	R	R	R	R

Note: * Determined by state of pins PE_7 to PE_0.

PORTE is an 8-bit read-only register that shows the pin states. It cannot be written to. V
output data for the port E pins (PE_7 to PE_0) must always be performed on PEDR.

If a port E read is performed while PEDDR bits are set to 1, the PEDR values are read.
read is performed while PEDDR bits are cleared to 0, the pin states are read.

After a power-on reset and in hardware standby mode, PORTE contents are determined
states, as PEDDR and PEDR are initialized. PORTE retains its prior state after a manua
in software standby mode.

incorporated into port E on an individual bit basis.

When a PEDDR bit is cleared to 0 (input port setting) when 8-bit bus mode is selected 2, 4 to 6, or in mode 3 or 7, setting the corresponding PEPCR bit to 1 turns on the MO up for the corresponding pin.

PEPCR is initialized to H'00 by a power-on reset, and in hardware standby mode. It re prior state after a manual reset, and in software standby mode.

9.12.3 Pin Functions

Modes 1, 2, 4 to 6

In modes 1, 2, 4 to 6, when 8-bit access is designated and 8-bit bus mode is selected, p are automatically designated as I/O ports. Setting a PEDDR bit to 1 makes the corresp E pin an output port, while clearing the bit to 0 makes the pin an input port.

When 16-bit bus mode is selected, the input/output direction specification by PEDDR and port E is designated for data I/O.

Port E pin functions in modes 1, 2, 4 to 6 are shown in figure 9.23.

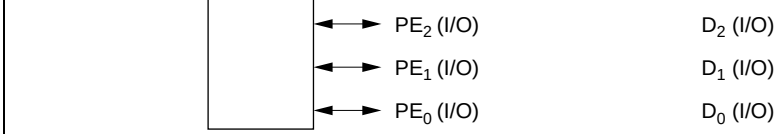


Figure 9.23 Port E Pin Functions (Modes 1, 2, 4 to 6)

Modes 3 and 7

In modes 3 and 7, port E pins function as I/O ports. Input or output can be specified for on a bit-by-bit basis. Setting a PEDDR bit to 1 makes the corresponding port E pin an o while clearing the bit to 0 makes the pin an input port.

Port E pin functions in modes 3 and 7 are shown in figure 9.24.

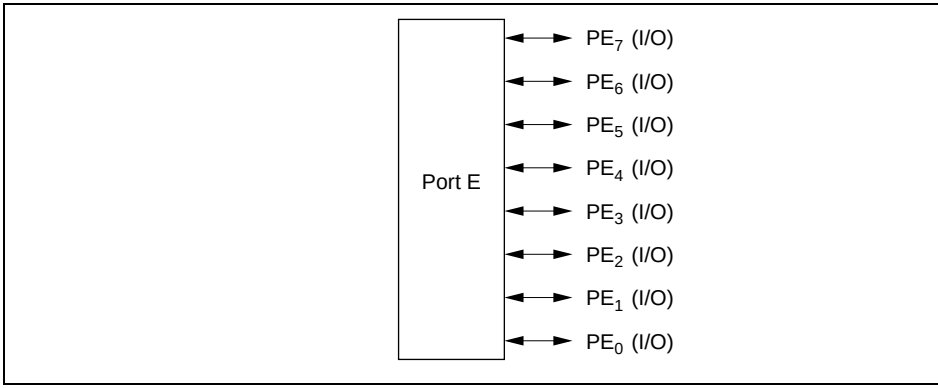


Figure 9.24 Port E Pin Functions (Modes 3 and 7)

The MOS input pull-up function is in the off state after a power-on reset, and in hardw
mode. The prior state is retained after a manual reset, and in software standby mode.

Table 9.22 summarizes the MOS input pull-up states.

Table 9.22 MOS Input Pull-Up States (Port E)

Modes		Power-On Reset	Hardware Standby Mode	Manual Reset	Software Standby Mode
3, 7		OFF		ON/OFF	
1, 2, 4 to 6	8-bit bus				
	16-bit bus			OFF	

Legend:
OFF: MOS input pull-up is always off.
ON/OFF: On when PEDDR = 0 and PEPCR = 1; otherwise off.

Figure 9.25 shows the port F pin configuration.

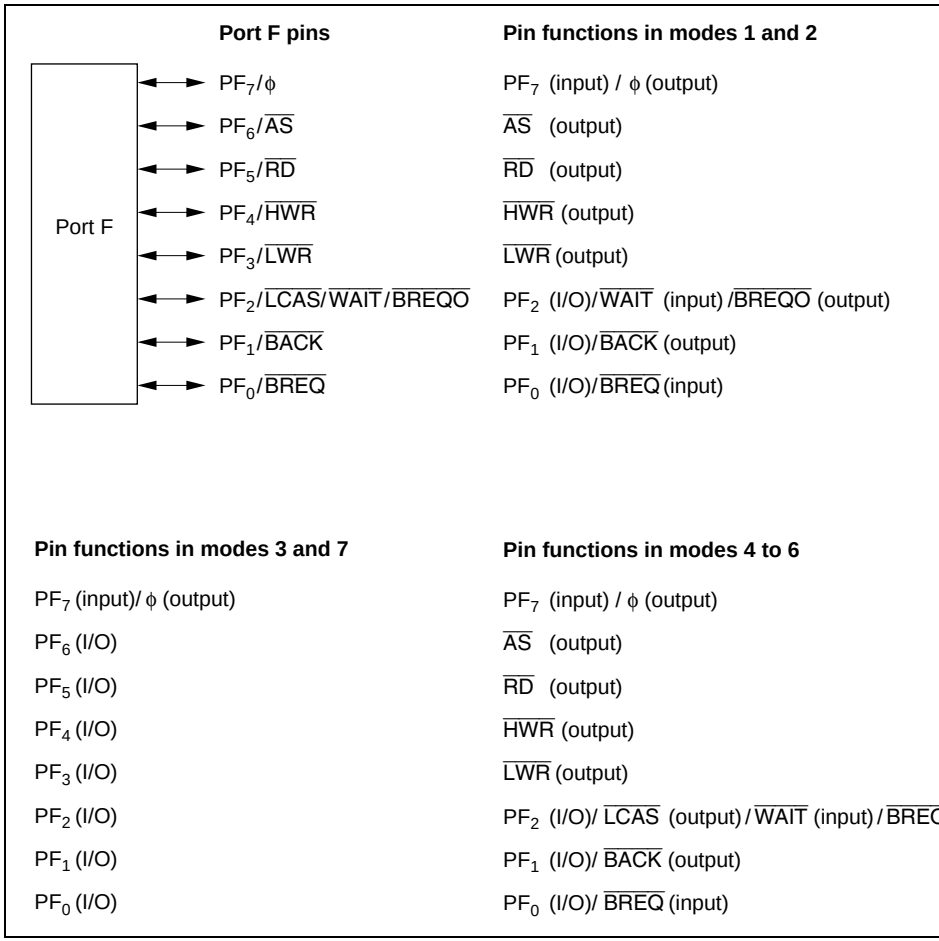


Figure 9.25 Port F Pin Functions

Port F register	PORTF	R	Undefined	H

Notes: 1. Lower 16 bits of the address.
2. Initial value depends on the mode.

Port F Data Direction Register (PFDDR)

Bit	:	7	6	5	4	3	2	1
		PF7DDR	PF6DDR	PF5DDR	PF4DDR	PF3DDR	PF2DDR	PF1DDR
Modes 1, 2, 4 to 6								
Initial value	:	1	0	0	0	0	0	0
R/W	:	W	W	W	W	W	W	W
Modes 3 and 7								
Initial value	:	0	0	0	0	0	0	0
R/W	:	W	W	W	W	W	W	W

PFDDR is an 8-bit write-only register, the individual bits of which specify input or ou
pins of port F. PFDDR cannot be read; if it is, an undefined value will be read.

PFDDR is initialized by a power-on reset, and in hardware standby mode, to H'80 in
4 to 6, and to H'00 in modes 3 and 7. It retains its prior state after a manual reset, and
standby mode. The OPE bit in SBYCR is used to select whether the bus control outpu
their output state or become high-impedance when a transition is made to software sta

- Modes 1, 2, 4 to 6

  Pin PF_7 functions as the ϕ output pin when the corresponding PFDDR bit is set to
  input port when the bit is cleared to 0.

  The input/output direction specified by PFDDR is ignored for pins PF_6 to PF_3, whi
  automatically designated as bus control outputs ($\overline{AS}$, $\overline{RD}$, $\overline{HWR}$, and $\overline{LWR}$).

  Pins PF_2 to PF_0 are designated as bus control input/output pins ($\overline{LCAS}$, $\overline{WAIT}$, $\overline{BR}$
  $\overline{BACK}$, $\overline{BREQ}$) by means of bus controller settings. At other times, setting a PFDI

Bit	:	7	6	5	4	3	2	1
		PF7DR	PF6DR	PF5DR	PF4DR	PF3DR	PF2DR	PF1DR
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

PFDR is an 8-bit readable/writable register that stores output data for the port F pins (P

PFDR is initialized to H'00 by a power-on reset, and in hardware standby mode. It retai
state after a manual reset, and in software standby mode.

Port F Register (PORTF)

Bit	:	7	6	5	4	3	2	1
		PF7	PF6	PF5	PF4	PF3	PF2	PF1
Initial value	:	—*	—*	—*	—*	—*	—*	—*
R/W	:	R	R	R	R	R	R	R

Note: * Determined by state of pins PF_7 to PF_0.

PORTF is an 8-bit read-only register that shows the pin states. Writing of output data fo
F pins (PF_7 to PF_0) must always be performed on PFDR.

If a port F read is performed while PFDDR bits are set to 1, the PFDR values are read.
read is performed while PFDDR bits are cleared to 0, the pin states are read.

After a power-on reset and in hardware standby mode, PORTF contents are determined
states, as PFDDR and PFDR are initialized. PORTF retains its prior state after a manua
in software standby mode.

PF_7/ϕ

The pin function is switched as shown below according to bit PF7D

PF7DDR	0	1
Pin function	PF_7 input pin	ϕ output

$PF_6/\overline{AS}$

The pin function is switched as shown below according to the oper and bit PF6DDR.

Operating Mode	Modes 1, 2, 4 to 6	Modes 3 and 7	
PF6DDR	—	0	
Pin function	$\overline{AS}$ output pin	PF_6 input pin	PF_6 o

$PF_5/\overline{RD}$

The pin function is switched as shown below according to the oper and bit PF5DDR.

Operating Mode	Modes 1, 2, 4 to 6	Modes 3 and 7	
PF5DDR	—	0	
Pin function	$\overline{RD}$ output pin	PF_5 input pin	PF_5 o

$PF_4/\overline{HWR}$

The pin function is switched as shown below according to the oper and bit PF4DDR.

Operating Mode	Modes 1, 2, 4 to 6	Modes 3 and 7	
PF4DDR	—	0	
Pin function	$\overline{HWR}$ output pin	PF_4 input pin	PF_4 o

PF_2/LCAS/WAIT/ $\overline{BREQO}$

The pin function is switched as shown below according to the comb
the operating mode, and bits RMTS2 to RMTS0, LCASS, BREQOE
and PF2DDR.

Operating Mode	Modes 1, 2, 4 to 6					Mode
LCASS	0				*	
BREQOE	0			1	—	
WAITE	0		1	—	—	
PF2DDR	0	1	—	—	—	0
Pin function	PF_2 input pin	PF_2 output pin	$\overline{WAIT}$ input pin	$\overline{BREQO}$ output pin	$\overline{LCAS}$ output pin	PF_2 inpu pin

Note: * Only when RMTS2 to RMTS0 = B'001 to B'011 and CW
modes 4 to 6.

PF_1/$\overline{BACK}$

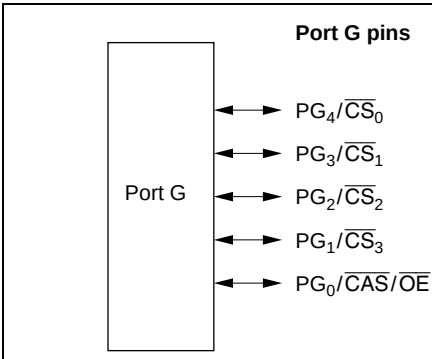
The pin function is switched as shown below according to the comb
the operating mode, and bits BRLE and PF1DDR.

Operating Mode	Modes 1, 2, 4 to 6			Modes 3
BRLE	0		1	—
PF1DDR	0	1	—	0
Pin function	PF_1 input pin	PF_1 output pin	$\overline{BACK}$ output pin	PF_1 input pin

PF_0/$\overline{BREQ}$

The pin function is switched as shown below according to the comb
the operating mode, and bits BRLE and PF0DDR.

Operating Mode	Modes 1, 2, 4 to 6			Modes 3
BRLE	0		1	—
PF0DDR	0	1	—	0
Pin function	PF_0 input pin	PF_0 output pin	$\overline{BREQ}$ input pin	PF_0 input pin



Pin functions in modes 1 and 2	
PG_4 (input)/$\overline{CS}_0$ (output)	
PG_3 (I/O)	
PG_2 (I/O)	
PG_1 (I/O)	
PG_0 (I/O)	

Pin functions in modes 3 and 7	Pin functions in modes 4 to 6
PG_4 (I/O)	PG_4 (input)/$\overline{CS}_0$ (output)
PG_3 (I/O)	PG_3 (input)/$\overline{CS}_1$ (output)
PG_2 (I/O)	PG_2 (input)/$\overline{CS}_2$ (output)
PG_1 (I/O)	PG_1 (input)/$\overline{CS}_3$ (output)
PG_0 (I/O)	PG_0 (I/O)/ $\overline{CAS}$ (output)/$\overline{OE}$ (out

Figure 9.26 Port G Pin Functions

Port G register	PORTG	R	Undefined	H'

Notes: 1. Lower 16 bits of the address.
2. Initial value depends on the mode.

Port G Data Direction Register (PGDDR)

Bit	:	7	6	5	4	3	2	1
		—	—	—	PG4DDR	PG3DDR	PG2DDR	PG1DDR
Modes 1, 4, 5								
Initial value	:	1	1	1	1	0	0	0
R/W	:	—	—	—	W	W	W	W
Modes 2, 3, 6, 7								
Initial value	:	1	1	1	0	0	0	0
R/W	:	—	—	—	W	W	W	W

PGDDR is an 8-bit write-only register, the individual bits of which specify input or out
pins of port G. PGDDR cannot be read, and bits 7 to 5 are reserved. If PGDDR is read,
undefined value will be read.

PGDDR is initialized by a power-on reset, and in hardware standby mode, to H'F0 in m
and 5, and to H'E0 in modes 2, 3, 6, and 7. It retains its prior state after a manual reset,
software standby mode. The OPE bit in SBYCR is used to select whether the bus contr
pins retain their output state or become high-impedance when a transition is made to so
standby mode.

the bit to 0 makes the pin an input port.

- Modes 4 to 6

  Pins PG_4 to PG_1 function as bus control output pins ($\overline{CS}_0$ to $\overline{CS}_3$) when the correspo
  PGDDR bits are set to 1, and as input ports when the bits are cleared to 0.

  Pin PG_0 functions as the $\overline{CAS}$ output pin when DRAM interface is designated, and
  output pin when PSRAM interface is designated. Otherwise, setting the correspon
  PGDDR bit to 1 makes the pin an output port, while clearing the bit to 0 makes the
  input port.

  For details of the DRAM and PSRAM interfaces, see section 6, Bus Controller.

Port G Data Register (PGDR)

Bit	:	7	6	5	4	3	2	1
		—	—	—	PG4DR	PG3DR	PG2DR	PG1DR
Initial value	:	1	1	1	0	0	0	0
R/W	:	—	—	—	R/W	R/W	R/W	R/W

PGDR is an 8-bit readable/writable register that stores output data for the port G pins

Bits 7 to 5 are reserved; they return an undetermined value if read, and cannot be mod

PGDR is initialized to H'E0 by a power-on reset, and in hardware standby mode. It re
state after a manual reset, and in software standby mode.

PORTG is an 8-bit read-only register that shows the pin states. It cannot be written to. V
output data for the port G pins (PG_4 to PG_0) must always be performed on PGDR.

Bits 7 to 5 are reserved; they return an undetermined value if read, and cannot be modif

If a port G read is performed while PGDDR bits are set to 1, the PGDR values are read
read is performed while PGDDR bits are cleared to 0, the pin states are read.

After a power-on reset and in hardware standby mode, PORTG contents are determined
states, as PGDDR and PGDR are initialized. PORTG retains its prior state after a manu
and in software standby mode.

$PG_4/\overline{CS}_0$ The pin function is switched as shown below according to the oper
and bit PG4DDR.

Operating Mode	Modes 1, 2, 4 to 6		Modes 3 a	
PG4DDR	0	1	0	
Pin function	PG_4 input pin	$\overline{CS}_0$ output pin	PG_4 input pin	PG

$PG_3/\overline{CS}_1$ The pin function is switched as shown below according to the oper
and bit PG3DDR.

Operating Mode	Modes 1 to 3, 7		Modes 4	
PG3DDR	0	1	0	
Pin function	PG_3 input pin	PG_3 output pin	PG_3 input pin	$\overline{CS}$

$PG_2/\overline{CS}_2$ The pin function is switched as shown below according to the oper
and bit PG2DDR.

Operating Mode	Modes 1 to 3, 7		Modes 4	
PG2DDR	0	1	0	
Pin function	PG_2 input pin	PG_2 output pin	PG_2 input pin	$\overline{CS}$

PG_0/$\overline{CAS}$/$\overline{OE}$ The pin function is switched as shown below according to the comb the operating mode and bits RMTS2 to RMTS0 and PG0DDR.

Operating Mode	Modes 1 to 3, 7		Modes 4 to 6		
RMTS2 to RMTS0	—		B'000, B'100		B'0XX
PG0DDR	0	1	0	1	
Pin function	PG_0 input pin	PG_0 output pin	PG_0 input pin	PG_0 output pin	$\overline{CAS}$ output

10.1.1 Features

- Maximum 16-pulse input/output
- A total of 16 timer general registers (TGRs) are provided (four each for channels 0
  two each for channels 1, 2, 4, and 5), each of which can be set independently as an
  compare/input capture register

  TGRC and TGRD for channels 0 and 3 can also be used as buffer registers
- Selection of 8 counter input clocks for each channel
- The following operations can be set for each channel:
  - Waveform output at compare match: Selection of 0, 1, or toggle output
  - Input capture function: Selection of rising edge, falling edge, or both edge dete
  - Counter clear operation: Counter clearing possible by compare match or input
  - Synchronous operation: Multiple timer counters (TCNT) can be written to simu

    Simultaneous clearing by compare match and input capture possible

    Register simultaneous input/output possible by counter synchronous operation
  - PWM mode: Any PWM output duty can be set

    Maximum of 15-phase PWM output possible by combination with synchronou
- Buffer operation settable for channels 0 and 3
  - Input capture register double-buffering possible
  - Automatic rewriting of output compare register possible
- Phase counting mode settable independently for each of channels 1, 2, 4, and 5
  - Two-phase encoder pulse up/down-count possible
- Cascaded operation
  - Channel 2 (channel 5) input clock operates as 32-bit counter by setting channel
    4) overflow/underflow
- Fast access via internal 16-bit bus
  - Fast access is possible via a 16-bit interface

- Programmable pulse generator (PPG) output trigger can be generated
  - Channel 0 to 3 compare match/input capture signals can be used as PPG output
- A/D converter conversion start trigger can be generated
  - Channel 0 to 5 compare match A/input capture A signals can be used as A/D co conversion start trigger

Table 10.1 lists the functions of the TPU.

		TCLKD		TCLKC	TCLKA	
General registers		TGR0A TGR0B	TGR1A TGR1B	TGR2A TGR2B	TGR3A TGR3B	TGR4A TGR4B
General registers/ buffer registers		TGR0C TGR0D	—	—	TGR3C TGR3D	—
I/O pins		TIOCA0 TIOCB0 TIOCC0 TIOCD0	TIOCA1 TIOCB1	TIOCA2 TIOCB2	TIOCA3 TIOCB3 TIOCC3 TIOCD3	TIOCA4 TIOCB4
Counter clear function		TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture
Compare match output	0 output	○	○	○	○	○
	1 output	○	○	○	○	○
	Toggle output	○	○	○	○	○
Input capture function		○	○	○	○	○
Synchronous operation		○	○	○	○	○
PWM mode		○	○	○	○	○
Phase counting mode		—	○	○	—	○
Buffer operation		○	—	—	○	—

	input capture	input capture	input capture	input capture	input capture	i c
A/D converter trigger	TGR0A compare match or input capture	TGR1A compare match or input capture	TGR2A compare match or input capture	TGR3A compare match or input capture	TGR4A compare match or input capture	T c m i c
PPG trigger	TGR0A/ TGR0B compare match or input capture	TGR1A/ TGR1B compare match or input capture	TGR2A/ TGR2B compare match or input capture	TGR3A/ TGR3B compare match or input capture	—	–
Interrupt sources	5 sources • Compare match or input capture 0A • Compare match or input capture 0B • Compare match or input capture 0C • Compare match or input capture 0D • Overflow	4 sources • Compare match or input capture 1A • Compare match or input capture 1B • Overflow • Underflow	4 sources • Compare match or input capture 2A • Compare match or input capture 2B • Overflow • Underflow	5 sources • Compare match or input capture 3A • Compare match or input capture 3B • Compare match or input capture 3C • Compare match or input capture 3D • Overflow	4 sources • Compare match or input capture 4A • Compare match or input capture 4B • Overflow • Underflow	4 • • • •

Legend:

○ : Possible

— : Not possible

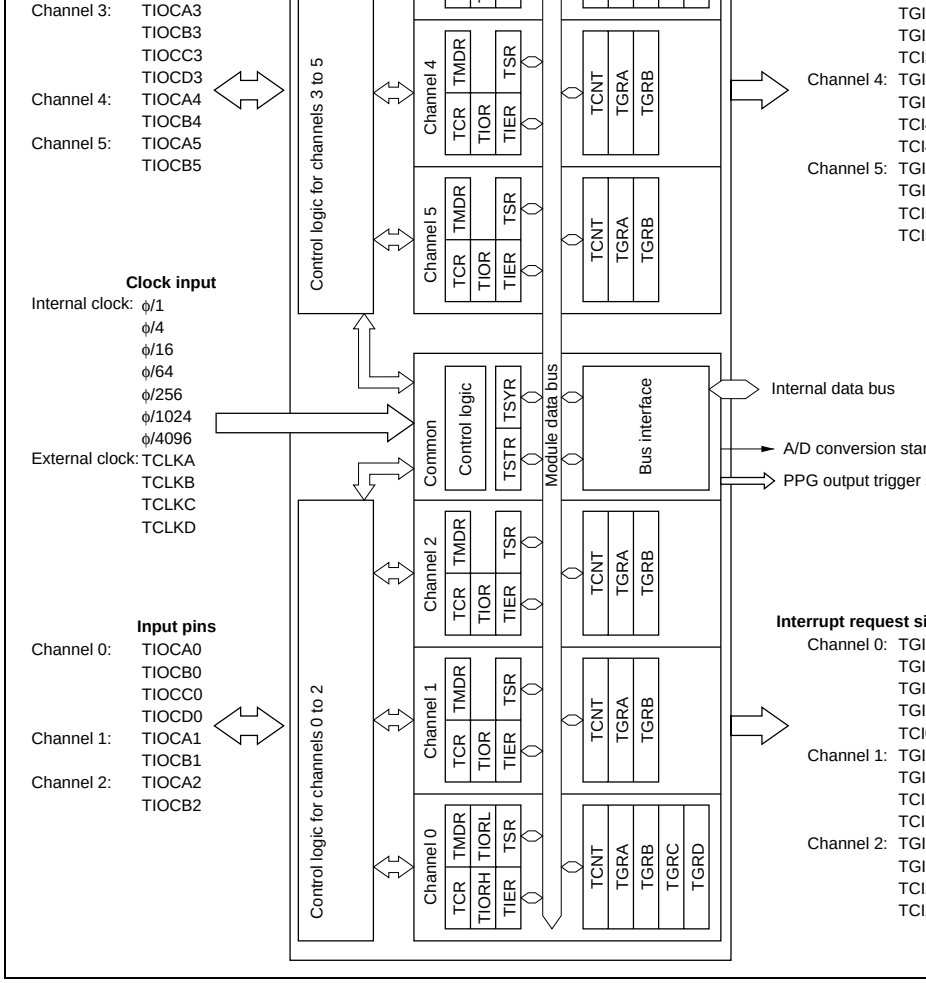


Figure 10.1 Block Diagram of TPU

				phase input)
	Clock input B	TCLKB	Input	External clock B input pin (Channel 1 and 5 phase counting m phase input)
	Clock input C	TCLKC	Input	External clock C input pin (Channel 2 and 4 phase counting m phase input)
	Clock input D	TCLKD	Input	External clock D input pin (Channel 2 and 4 phase counting m phase input)
0	Input capture/out compare match A0	TIOCA0	I/O	TGR0A input capture input/output c output/PWM output pin
	Input capture/out compare match B0	TIOCB0	I/O	TGR0B input capture input/output c output/PWM output pin
	Input capture/out compare match C0	TIOCC0	I/O	TGR0C input capture input/output c output/PWM output pin
	Input capture/out compare match D0	TIOCD0	I/O	TGR0D input capture input/output c output/PWM output pin
1	Input capture/out compare match A1	TIOCA1	I/O	TGR1A input capture input/output c output/PWM output pin
	Input capture/out compare match B1	TIOCB1	I/O	TGR1B input capture input/output c output/PWM output pin
2	Input capture/out compare match A2	TIOCA2	I/O	TGR2A input capture input/output c output/PWM output pin
	Input capture/out compare match B2	TIOCB2	I/O	TGR2B input capture input/output c output/PWM output pin

	compare match D3			output/PWM output pin
4	Input capture/out compare match A4	TIOCA4	I/O	TGR4A input capture input/output output/PWM output pin
	Input capture/out compare match B4	TIOCB4	I/O	TGR4B input capture input/output output/PWM output pin
5	Input capture/out compare match A5	TIOCA5	I/O	TGR5A input capture input/output output/PWM output pin
	Input capture/out compare match B5	TIOCB5	I/O	TGR5B input capture input/output output/PWM output pin

	Timer I/O control register 0H	TIOR0H	R/W	H'00	H
	Timer I/O control register 0L	TIOR0L	R/W	H'00	H
	Timer interrupt enable register 0	TIER0	R/W	H'40	H
	Timer status register 0	TSR0	R/(W)*2	H'C0	H
	Timer counter 0	TCNT0	R/W	H'0000	H
	Timer general register 0A	TGR0A	R/W	H'FFFF	H
	Timer general register 0B	TGR0B	R/W	H'FFFF	H
	Timer general register 0C	TGR0C	R/W	H'FFFF	H
	Timer general register 0D	TGR0D	R/W	H'FFFF	H
1	Timer control register 1	TCR1	R/W	H'00	H
	Timer mode register 1	TMDR1	R/W	H'C0	H
	Timer I/O control register 1	TIOR1	R/W	H'00	H
	Timer interrupt enable register 1	TIER1	R/W	H'40	H
	Timer status register 1	TSR1	R/(W)*2	H'C0	H
	Timer counter 1	TCNT1	R/W	H'0000	H
	Timer general register 1A	TGR1A	R/W	H'FFFF	H
	Timer general register 1B	TGR1B	R/W	H'FFFF	H
2	Timer control register 2	TCR2	R/W	H'00	H
	Timer mode register 2	TMDR2	R/W	H'C0	H
	Timer I/O control register 2	TIOR2	R/W	H'00	H
	Timer interrupt enable register 2	TIER2	R/W	H'40	H
	Timer status register 2	TSR2	R/(W)*2	H'C0	H
	Timer counter 2	TCNT2	R/W	H'0000	H
	Timer general register 2A	TGR2A	R/W	H'FFFF	H
	Timer general register 2B	TGR2B	R/W	H'FFFF	H

	Timer counter 3	TCNT3	R/W	H'0000
	Timer general register 3A	TGR3A	R/W	H'FFFF
	Timer general register 3B	TGR3B	R/W	H'FFFF
	Timer general register 3C	TGR3C	R/W	H'FFFF
	Timer general register 3D	TGR3D	R/W	H'FFFF
4	Timer control register 4	TCR4	R/W	H'00
	Timer mode register 4	TMDR4	R/W	H'C0
	Timer I/O control register 4	TIOR4	R/W	H'00
	Timer interrupt enable register 4	TIER4	R/W	H'40
	Timer status register 4	TSR4	R/(W)*2	H'C0
	Timer counter 4	TCNT4	R/W	H'0000
	Timer general register 4A	TGR4A	R/W	H'FFFF
	Timer general register 4B	TGR4B	R/W	H'FFFF
5	Timer control register 5	TCR5	R/W	H'00
	Timer mode register 5	TMDR5	R/W	H'C0
	Timer I/O control register 5	TIOR5	R/W	H'00
	Timer interrupt enable register 5	TIER5	R/W	H'40
	Timer status register 5	TSR5	R/(W)*2	H'C0
	Timer counter 5	TCNT5	R/W	H'0000
	Timer general register 5A	TGR5A	R/W	H'FFFF
	Timer general register 5B	TGR5B	R/W	H'FFFF
All	Timer start register	TSTR	R/W	H'00
	Timer synchro register	TSYR	R/W	H'00
	Module stop control register	MSTPCR	R/W	H'3FFF

Notes: 1. Lower 16 bits of the address.
2. Can only be written with 0 for flag clearing.

Channel 0: TCR0
Channel 3: TCR3

Bit	:	7	6	5	4	3	2	1
		CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1
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

Channel 1: TCR1
Channel 2: TCR2
Channel 4: TCR4
Channel 5: TCR5

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/W	R/W	R/W	R/W	R/W	R/W

		1	0	TCNT cleared by TGRB compare m capture
			1	TCNT cleared by counter clearing fo channel performing synchronous cle synchronous operation*1
	1	0	0	TCNT clearing disabled
			1	TCNT cleared by TGRC compare m capture*2
		1	0	TCNT cleared by TGRD compare m capture*2
			1	TCNT cleared by counter clearing fo channel performing synchronous cle synchronous operation*1

Channel	Bit 7 Reserved*3	Bit 6 CCLR1	Bit 5 CCLR0	Description
1, 2, 4, 5	0	0	0	TCNT clearing disabled
			1	TCNT cleared by TGRA compare m capture
		1	0	TCNT cleared by TGRB compare m capture
			1	TCNT cleared by counter clearing fo channel performing synchronous cle synchronous operation*1

Notes: 1. Synchronous operation setting is performed by setting the SYNC bit in TSY
2. When TGRC or TGRD is used as a buffer register, TCNT is not cleared bec buffer register setting has priority, and compare match/input capture does n
3. Bit 7 is reserved in channels 1, 2, 4, and 5. It is always read as 0 and canno modified.

	1	Count at falling edge
1	*	Count at both edges

Legend: *: Don't care

Note: Internal clock edge selection is valid when the input clock is φ/4 or slower. This s
ignored if the input clock is φ/1, or when overflow/underflow of another channel is

Bits 2, 1, and 0—Time Prescaler 2, 1, and 0 (TPSC2 to TPSC0): These bits select th
counter clock. The clock source can be selected independently for each channel. Table
the clock sources that can be set for each channel.

Table 10.4 TPU Clock Sources

	Internal Clock							External Clock			
Channel	**φ/1**	**φ/4**	**φ/16**	**φ/64**	**φ/256**	**φ/1024**	**φ/4096**	**TCLKA**	**TCLKB**	**TCLKC**	**TCLKD**
0	○	○	○	○				○	○	○	○
1	○	○	○	○	○			○	○		
2	○	○	○	○		○		○	○	○	
3	○	○	○	○	○	○	○	○			
4	○	○	○	○		○		○		○	
5	○	○	○	○	○			○		○	○

Legend:
○: Setting
Blank: No setting

			1	External clock: counts on TCLKB pi
		1	0	External clock: counts on TCLKC pi
			1	External clock: counts on TCLKD pi

	Bit 2	Bit 1	Bit 0	
Channel	**TPSC2**	**TPSC1**	**TPSC0**	**Description**
1	0	0	0	Internal clock: counts on φ/1
			1	Internal clock: counts on φ/4
		1	0	Internal clock: counts on φ/16
			1	Internal clock: counts on φ/64
	1	0	0	External clock: counts on TCLKA pi
			1	External clock: counts on TCLKB pi
		1	0	Internal clock: counts on φ/256
			1	Counts on TCNT2 overflow/underflo

Note: This setting is ignored when channel 1 is in phase counting mode.

	Bit 2	Bit 1	Bit 0	
Channel	**TPSC2**	**TPSC1**	**TPSC0**	**Description**
2	0	0	0	Internal clock: counts on φ/1
			1	Internal clock: counts on φ/4
		1	0	Internal clock: counts on φ/16
			1	Internal clock: counts on φ/64
	1	0	0	External clock: counts on TCLKA pi
			1	External clock: counts on TCLKB pi
		1	0	External clock: counts on TCLKC pi
			1	Internal clock: counts on φ/1024

Note: This setting is ignored when channel 2 is in phase counting mode.

			1	Internal clock: counts on φ/1024
		1	0	Internal clock: counts on φ/256
			1	Internal clock: counts on φ/4096

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
4	0	0	0	Internal clock: counts on φ/1
			1	Internal clock: counts on φ/4
		1	0	Internal clock: counts on φ/16
			1	Internal clock: counts on φ/64
	1	0	0	External clock: counts on TCLKA pin
			1	External clock: counts on TCLKC pin
		1	0	Internal clock: counts on φ/1024
			1	Counts on TCNT5 overflow/underflow

Note: This setting is ignored when channel 4 is in phase counting mode.

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
5	0	0	0	Internal clock: counts on φ/1
			1	Internal clock: counts on φ/4
		1	0	Internal clock: counts on φ/16
			1	Internal clock: counts on φ/64
	1	0	0	External clock: counts on TCLKA pin
			1	External clock: counts on TCLKC pin
		1	0	Internal clock: counts on φ/256
			1	External clock: counts on TCLKD pin

Note: This setting is ignored when channel 5 is in phase counting mode.

Channel 3: TMDR3

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

Channel 1: TMDR1
Channel 2: TMDR2
Channel 4: TMDR4
Channel 5: TMDR5

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

Bits 7 and 6—Reserved: Read-only bits, always read as 1.

Bit 5—Buffer Operation B (BFB): Specifies whether TGRB is to operate in the norm
TGRB and TGRD are to be used together for buffer operation. When TGRD is used a
register, TGRD input capture/output compare is not generated.

In channels 1, 2, 4, and 5, which have no TGRD, bit 5 is reserved. It is always read as
cannot be modified.

Bit 5

BFB	Description
0	TGRB operates normally
1	TGRB and TGRD used together for buffer operation

0	TGRA operates normally	(
1	TGRA and TGRC used together for buffer operation	

Bits 3 to 0—Modes 3 to 0 (MD3 to MD0): These bits are used to set the timer operati

Bit 3	Bit 2	Bit 1	Bit 0		
MD3[*1]	**MD2**[*2]	**MD1**	**MD0**	**Description**	
0	0	0	0	Normal operation	(
			1	Reserved	
		1	0	PWM mode 1	
			1	PWM mode 2	
	1	0	0	Phase counting mode 1	
			1	Phase counting mode 2	
		1	0	Phase counting mode 3	
			1	Phase counting mode 4	
1	*	*	*	—	

Legend: *: Don’t care

Notes: 1. MD3 is a reserved bit. In a write, it should always be written with 0.
2. Phase counting mode cannot be set for channels 0 and 3. In this case, 0 sho be written to MD2.

Bit	:	7	6	5	4	3	2	1
		IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1
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

Channel 0: TIOR0L
Channel 3: TIOR3L

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 TGRC or TGRD is designated for buffer operation, this setting is invalid register operates as a buffer register.

The TIOR registers are 8-bit registers that control the TGR registers. The TPU has eig registers, two each for channels 0 and 3, and one each for channels 1, 2, 4, and 5. The registers are initialized to H'00 by a reset, and in hardware standby mode.

Care is required since TIOR is affected by the TMDR setting. The initial output specif is valid when the counter is stopped (the CST bit in TSTR is cleared to 0). Note also th mode 2, the output at the point at which the counter is cleared to 0 is specified.

			1	is output compare register	Initial output is 0 output	0 output at compa
		1	0			1 output at compa
			1			Toggle output at c match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compa
		1	0			1 output at compa
			1			Toggle output at c match
1	0	0	0	TGR0B is input capture register	Capture input source is TIOCB0 pin	Input capture at ri
			1			Input capture at fa
		1	*			Input capture at b
	1	*	*		Capture input source is channel 1/count clock	Input capture at T count- up/count-d

Legend: *: Don't care

Note: 1. When bits TPSC2 to TPSC0 in TCR1 are set to B'000 and ϕ/1 is used as the count clock, this setting is invalid and input capture is not generated.

			1		Initial output is 1 output	0 output at comp
		1	0			1 output at comp
			1			Toggle output at match
1	0	0	0	TGR0D is input capture register*2	Capture input source is TIOCD0 pin	Input capture at r
			1			Input capture at f
		1	*			Input capture at b
	1	*	*		Capture input source is channel 1/count clock	Input capture at T count-up/count-d

Legend: *: Don’t care

Notes: 1. When bits TPSC2 to TPSC0 in TCR1 are set to B'000 and φ/1 is used as th count clock, this setting is invalid and input capture is not generated.

2. When the BFB bit in TMDR0 is set to 1 and TGR0D is used as a buffer regi setting is invalid and input capture/output compare is not generated.

				1		Initial output is 1 output	0 output at compa
			1	0			1 output at compa
				1			Toggle output at c match
	1	0	0	0	TGR1B is input capture register	Capture input source is TIOCB1 pin	Input capture at ri
				1			Input capture at fa
			1	*			Input capture at b
		1	*	*		Capture input source is TGR0C compare match/ input capture	Input capture at g TGR0C compare capture

Legend: *: Don't care

Channel	Bit 7 IOB3	Bit 6 IOB2	Bit 5 IOB1	Bit 4 IOB0	Description		
2	0	0	0	0	TGR2B is output compare register	Output disabled	(
				1		Initial output is 0 output	0 output at compa
			1	0			1 output at compa
				1			Toggle output at c match
		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at compa
			1	0			1 output at compa
				1			Toggle output at c match
	1	0	0	0	TGR2B is input capture register	Capture input source is TIOCB2 pin	Input capture at ri
				1			Input capture at fa
			1	*			Input capture at b

Legend: *: Don't care

			1		Initial output is 1 output	0 output at comp
		1	0			1 output at comp
			1			Toggle output at match
1	0	0	0	TGR3B is input capture register	Capture input source is TIOCB3 pin	Input capture at r
			1			Input capture at f
		1	*			Input capture at b
	1	*	*		Capture input source is channel 4/count clock	Input capture at T count-up/count-d

Legend: *: Don’t care

Note: 1. When bits TPSC2 to TPSC0 in TCR4 are set to B'000 and ϕ/1 is used as th count clock, this setting is invalid and input capture is not generated.

			1		Initial output is 1 output	0 output at compa
		1	0			1 output at compa
			1			Toggle output at c match
1	0	0	0	TGR3D is input capture register*2	Capture input source is TIOCD3 pin	Input capture at ri
			1			Input capture at fa
		1	*			Input capture at b
	1	*	*		Capture input source is channel 4/count clock	Input capture at T count-up/count-do

Legend: *: Don't care

Notes: 1. When bits TPSC2 to TPSC0 in TCR4 are set to B'000 and ϕ/1 is used as the count clock, this setting is invalid and input capture is not generated.

2. When the BFB bit in TMDR3 is set to 1 and TGR3D is used as a buffer regis setting is invalid and input capture/output compare is not generated.

				1		Initial output is 1 output	0 output at comp
			1	0			1 output at comp
				1			Toggle output at match
	1	0	0	0	TGR4B is input capture register	Capture input source is TIOCB4 pin	Input capture at r
				1			Input capture at f
			1	*			Input capture at l
		1	*	*		Capture input source is TGR3C compare match/ input capture	Input capture at g TGR3C compare input capture

Legend: *: Don't care

Channel	Bit 7 IOB3	Bit 6 IOB2	Bit 5 IOB1	Bit 4 IOB0	Description		
5	0	0	0	0	TGR5B is output compare register	Output disabled	
				1		Initial output is 0 output	0 output at comp
			1	0			1 output at comp
				1			Toggle output at match
		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at comp
			1	0			1 output at comp
				1			Toggle output at match
	1	*	0	0	TGR5B is input capture register	Capture input source is TIOCB5 pin	Input capture at r
				1			Input capture at f
			1	*			Input capture at l

Legend: *: Don't care

			1	is output compare register	Initial output is 0 output	0 output at compa
		1	0			1 output at compa
			1			Toggle output at c match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compa
		1	0			1 output at compa
			1			Toggle output at c match
1	0	0	0	TGR0A is input capture register	Capture input source is TIOCA0 pin	Input capture at ri
			1			Input capture at fa
		1	*			Input capture at b
	1	*	*		Capture input source is channel 1/ count clock	Input capture at T count-up/count-do

Legend: *: Don't care

			1		Initial output is 1 output	0 output at comp
		1	0			1 output at comp
			1			Toggle output at match
1	0	0	0	TGR0C is input capture register*1	Capture input source is TIOCC0 pin	Input capture at r
			1			Input capture at f
		1	*			Input capture at b
	1	*	*		Capture input source is channel 1/count clock	Input capture at T count-up/count-d

Legend: *: Don't care

Note: 1. When the BFA bit in TMDR0 is set to 1 and TGR0C is used as a buffer regi setting is invalid and input capture/output compare is not generated.

				1		Initial output is 1 output	0 output at compa
			1	0			1 output at compa
				1			Toggle output at c match
	1	0	0	0	TGR1A is input capture register	Capture input source is TIOCA1 pin	Input capture at ri
				1			Input capture at fa
			1	*			Input capture at b
		1	*	*		Capture input source is TGR0A compare match/ input capture	Input capture at g channel 0/TGR0A match/input captu

Legend: *: Don't care

Channel	Bit 3 IOA3	Bit 2 IOA2	Bit 1 IOA1	Bit 0 IOA0	Description		
2	0	0	0	0	TGR2A is output compare register	Output disabled	(
				1		Initial output is 0 output	0 output at compa
			1	0			1 output at compa
				1			Toggle output at c match
		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at compa
			1	0			1 output at compa
				1			Toggle output at c match
	1	*	0	0	TGR2A is input capture register	Capture input source is TIOCA2 pin	Input capture at ri
				1			Input capture at fa
			1	*			Input capture at b

Legend: *: Don't care

						match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at comp
		1	0			1 output at comp
			1			Toggle output at match
1	0	0	0	TGR3A is input capture register	Capture input source is TIOCA3 pin	Input capture at
			1			Input capture at f
		1	*			Input capture at
	1	*	*		Capture input source is channel 4/count clock	Input capture at count-up/count-d

Legend: *: Don’t care

			1		Initial output is 1 output	0 output at compa
		1	0			1 output at compa
			1			Toggle output at c match
1	0	0	0	TGR3C is input capture register*1	Capture input source is TIOCC3 pin	Input capture at ri
			1			Input capture at fa
		1	*			Input capture at b
	1	*	*		Capture input source is channel 4/count clock	Input capture at T count-up/count-do

Legend: *: Don't care

Note: 1. When the BFA bit in TMDR3 is set to 1 and TGR3C is used as a buffer regis setting is invalid and input capture/output compare is not generated.

				1		Initial output is 1 output	0 output at comp
			1	0			1 output at comp
				1			Toggle output at match
	1	0	0	0	TGR4A is input capture register	Capture input source is TIOCA4 pin	Input capture at r
				1			Input capture at f
			1	*			Input capture at b
		1	*	*		Capture input source is TGR3A compare match/ input capture	Input capture at g TGR3A compare capture

Legend: *: Don't care

Channel	Bit 3 IOA3	Bit 2 IOA2	Bit 1 IOA1	Bit 0 IOA0	Description		
5	0	0	0	0	TGR5A is output compare register	Output disabled	(
				1		Initial output is 0 output	0 output at comp
			1	0			1 output at comp
				1			Toggle output at match
		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at comp
			1	0			1 output at comp
				1			Toggle output at match
	1	*	0	0	TGR5A is input capture register	Capture input source is TIOCA5 pin	Input capture at r
				1			Input capture at f
			1	*			Input capture at b

Legend: *: Don't care

R/W	:	R/W	—	—	R/W	R/W	R/W	R/W

Channel 1: TIER1
Channel 2: TIER2
Channel 4: TIER4
Channel 5: TIER5

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

The TIER registers are 8-bit registers that control enabling or disabling of interrupt req each channel. The TPU has six TIER registers, one for each channel. The TIER registe initialized to H'40 by a reset, and in hardware standby mode.

Bit 7—A/D Conversion Start Request Enable (TTGE): Enables or disables generati conversion start requests by TGRA input capture/compare match.

Bit 7

TTGE	Description
0	A/D conversion start request generation disabled
1	A/D conversion start request generation enabled

Bit 6—Reserved: Read-only bit, always read as 1.

1	Interrupt requests (TCIU) by TCFU enabled

Bit 4—Overflow Interrupt Enable (TCIEV): Enables or disables interrupt requests the TCFV flag when the TCFV flag in TSR is set to 1.

Bit 4

TCIEV	Description
0	Interrupt requests (TCIV) by TCFV disabled
1	Interrupt requests (TCIV) by TCFV enabled

Bit 3—TGR Interrupt Enable D (TGIED): Enables or disables interrupt requests (T TGFD bit when the TGFD bit in TSR is set to 1 in channels 0 and 3.

In channels 1, 2, 4, and 5, bit 3 is reserved. It is always read as 0 and cannot be modifi

Bit 3

TGIED	Description
0	Interrupt requests (TGID) by TGFD bit disabled
1	Interrupt requests (TGID) by TGFD bit enabled

Bit 2—TGR Interrupt Enable C (TGIEC): Enables or disables interrupt requests (T TGFC bit when the TGFC bit in TSR is set to 1 in channels 0 and 3.

In channels 1, 2, 4, and 5, bit 2 is reserved. It is always read as 0 and cannot be modifi

Bit 2

TGIEC	Description
0	Interrupt requests (TGIC) by TGFC bit disabled
1	Interrupt requests (TGIC) by TGFC bit enabled

Bit 0—TGR Interrupt Enable A (TGIEA): Enables or disables interrupt requests (TG
TGFA bit when the TGFA bit in TSR is set to 1.

Bit 1

TGIEA	Description	
0	Interrupt requests (TGIA) by TGFA bit disabled	(
1	Interrupt requests (TGIA) by TGFA bit enabled	

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

Note: * Can only be written with 0 for flag clearing.

Channel 1: TSR1
Channel 2: TSR2
Channel 4: TSR4
Channel 5: TSR5

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/(W)*	R/(W)*	—	—	R/(W)*

Note: * Can only be written with 0 for flag clearing.

The TSR registers are 8-bit registers that indicate the status of each channel. The TPU registers, one for each channel. The TSR registers are initialized to H'C0 by a reset, ar hardware standby mode.

Bit 7—Count Direction Flag (TCFD): Status flag that shows the direction in which counts in channels 1, 2, 4, and 5.

In channels 0 and 3, bit 7 is reserved. It is always read as 1 and cannot be modified.

Bit 7

TCFD	Description
0	TCNT counts down
1	TCNT counts up

Bit 6—Reserved: Read-only bit, always read as 1.

	When 0 is written to TCFU after reading TCFU = 1
1	[Setting condition] When the TCNT value underflows (changes from H'0000 to H'FFFF)

Bit 4—Overflow Flag (TCFV): Status flag that indicates that TCNT overflow has occ

Bit 4

TCFV	Description
0	[Clearing condition] (When 0 is written to TCFV after reading TCFV = 1
1	[Setting condition] When the TCNT value overflows (changes from H'FFFF to H'0000)

Bit 3—Input Capture/Output Compare Flag D (TGFD): Status flag that indicates th occurrence of TGRD input capture or compare match in channels 0 and 3.

In channels 1, 2, 4, and 5, bit 3 is reserved. It is always read as 0 and cannot be modifie

Bit 3

TGFD	Description
0	[Clearing conditions] (• When DTC is activated by TGID interrupt while DISEL bit of MRB in DT • When 0 is written to TGFD after reading TGFD = 1
1	[Setting conditions] • When TCNT = TGRD while TGRD is functioning as output compare reg • When TCNT value is transferred to TGRD by input capture signal while functioning as input capture register

	• When DTC is activated by TGIC interrupt while DISEL bit of MRB in D
	• When 0 is written to TGFC after reading TGFC = 1
1	[Setting conditions]
	• When TCNT = TGRC while TGRC is functioning as output compare re
	• When TCNT value is transferred to TGRC by input capture signal while functioning as input capture register

Bit 1—Input Capture/Output Compare Flag B (TGFB): Status flag that indicates t occurrence of TGRB input capture or compare match.

Bit 1

TGFB	Description
0	[Clearing conditions]
	• When DTC is activated by TGIB interrupt while DISEL bit of MRB in D
	• When 0 is written to TGFB after reading TGFB = 1
1	[Setting conditions]
	• When TCNT = TGRB while TGRB is functioning as output compare re
	• When TCNT value is transferred to TGRB by input capture signal while functioning as input capture register

	• When 0 is written to TGFA after reading TGFA = 1
1	[Setting conditions] • When TCNT = TGRA while TGRA is functioning as output compare regis • When TCNT value is transferred to TGRA by input capture signal while T functioning as input capture register

10.2.6 Timer Counter (TCNT)

Channel 0: TCNT0 (up-counter)
Channel 1: TCNT1 (up/down-counter*)
Channel 2: TCNT2 (up/down-counter*)
Channel 3: TCNT3 (up-counter)
Channel 4: TCNT4 (up/down-counter*)
Channel 5: TCNT5 (up/down-counter*)

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
Initial value	:	0	0	0	0	0	0	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	R/W	R/W	R/W	R/W	R/W	R/W

Note: * These counters can be used as up/down-counters only in phase counting m when counting overflow/underflow on another channel. In other cases they f as up-counters.

The TCNT registers are 16-bit counters. The TPU has six TCNT counters, one for each

The TCNT counters are initialized to H'0000 by a reset, and in hardware standby mode

The TCNT counters cannot be accessed in 8-bit units; they must always be accessed as unit.

The TGR registers are 16-bit registers with a dual function as output compare and inp
registers. The TPU has 16 TGR registers, four each for channels 0 and 3 and two each
1, 2, 4, and 5. TGRC and TGRD for channels 0 and 3 can also be designated for opera
buffer registers*. The TGR registers are initialized to H'FFFF by a reset, and in hardw
mode.

The TGR registers cannot be accessed in 8-bit units; they must always be accessed as

Note: * TGR buffer register combinations are TGRA—TGRC and TGRB—TGRD

TSTR is initialized to H'00 by a reset, and in hardware standby mode.

When setting the operating mode in TMDR or the TCNT count clock in TCR, TCNT c
operation should first be stopped.

Bits 7 and 6—Reserved: Should always be written with 0.

Bits 5 to 0—Counter Start 5 to 0 (CST5 to CST0): These bits select operation or stop
TCNT.

Bit n

CSTn	Description
0	TCNTn count operation is stopped (I
1	TCNTn performs count operation

Notes: 1. n = 5 to 0

2. If 0 is written to the CST bit during operation with the TIOC pin designated fo
the counter stops but the TIOC pin output compare output level is retained. I
written to when the CST bit is cleared to 0, the pin output level will be chang
set initial output value.

operation for the channel 0 to 5 TCNT counters. A channel performs synchronous ope
the corresponding bit in TSYR is set to 1.

TSYR is initialized to H'00 by a reset, and in hardware standby mode.

Bits 7 and 6—Reserved: Should always be written with 0..

Bits 5 to 0—Timer Synchro 5 to 0 (SYNC5 to SYNC0): These bits select whether o
independent of or synchronized with other channels.

When synchronous operation is selected, synchronous presetting of multiple channels[1] and synchronous clearing through counter clearing on another channel[*2] are possible.

Notes: 1. To set synchronous operation, the SYNC bits for at least two channels mus
2. To set synchronous clearing, in addition to the SYNC bit, the TCNT cleari must also be set by means of bits CCLR2 to CCLR0 in TCR.

Bit n

SYNCn	Description
0	TCNTn operates independently (TCNT presetting/clearing is unrelated to other channels)
1	TCNTn performs synchronous operation TCNT synchronous presetting/synchronous clearing is possible

Note: n = 5 to 0

MSTPCR is a 16-bit readable/writable register that performs module stop mode control

When the MSTP13 bit in MSTPCR is set to 1, TPU operation stops at the end of the bu
a transition is made to module stop mode. Registers cannot be read or written to in mod
mode. For details, see section 21.5, Module Stop Mode.

MSTPCR is initialized to H'3FFF by a reset and in hardware standby mode. It is not ini
software standby mode.

Bit 13—Module Stop (MSTP13): Specifies the TPU module stop mode.

Bit 13

MSTP13	Description
0	TPU module stop mode cleared
1	TPU module stop mode set (I

An example of 16-bit register access operation is shown in figure 10.2.

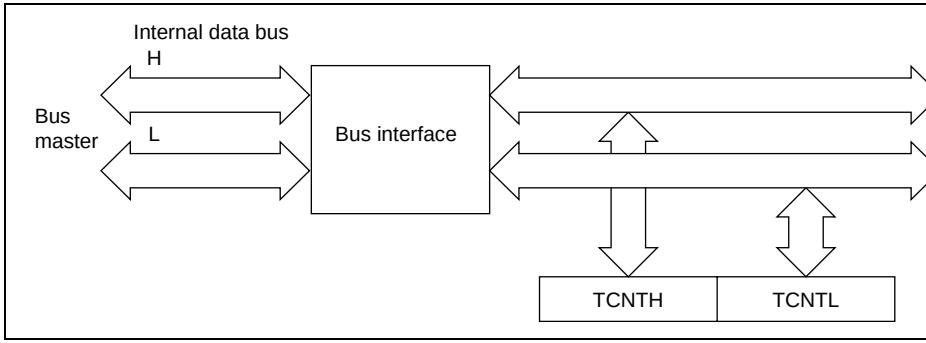


Figure 10.2 16-Bit Register Access Operation [Bus Master ↔ TCNT (16

10.3.2 8-Bit Registers

Registers other than TCNT and TGR are 8-bit. As the data bus to the CPU is 16 bits w
registers can be read and written to in 16-bit units. They can also be read and written t
units.

Examples of 8-bit register access operation are shown in figures 10.3, 10.4, and 10.5.

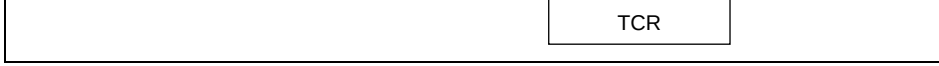


Figure 10.3 8-Bit Register Access Operation [Bus Master ↔ TCR (Upper 8

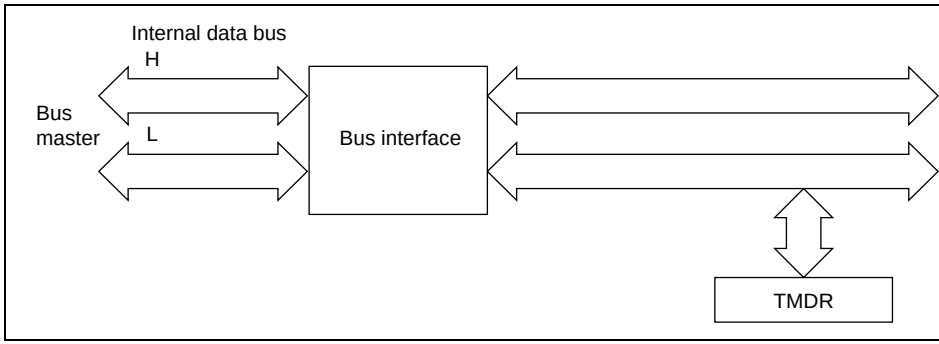


Figure 10.4 8-Bit Register Access Operation [Bus Master ↔ TMDR (Lower 8

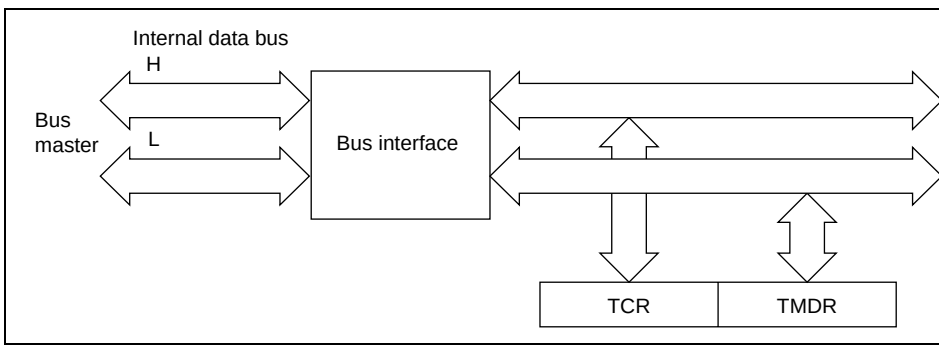


Figure 10.5 8-Bit Register Access Operation [Bus Master ↔ TCR and TMDR (

Each channel has a TCNT and TGR register. TCNT performs up-counting, and is also free-running operation, synchronous counting, and external event counting.

Each TGR can be used as an input capture register or output compare register.

Synchronous Operation

When synchronous operation is designated for a channel, TCNT for that channel perfo synchronous presetting. That is, when TCNT for a channel designated for synchronou is rewritten, the TCNT counters for the other channels are also rewritten at the same ti Synchronous clearing of the TCNT counters is also possible by setting the timer synch bits in TSYR for channels designated for synchronous operation.

Buffer Operation

When TGR is an output compare register: When a compare match occurs, the valu buffer register for the relevant channel is transferred to TGR.

When TGR is an input capture register: When input capture occurs, the value in TC transfer to TGR and the value previously held in TGR is transferred to the buffer regis

Cascaded Operation

The channel 1 counter (TCNT1), channel 2 counter (TCNT2), channel 4 counter (TCN channel 5 counter (TCNT5) can be connected together to operate as a 32-bit counter.

PWM Mode

In this mode, a PWM waveform is output. The output level can be set by means of TIO waveform with a duty of between 0% and 100% can be output, according to the settin TGR register.

Counter Operation

When one of bits CST0 to CST5 is set to 1 in TSTR, the TCNT counter for the corresp
channel starts counting. TCNT can operate as a free-running counter, periodic counter,

Example of count operation setting procedure: Figure 10.6 shows an example of the
operation setting procedure.

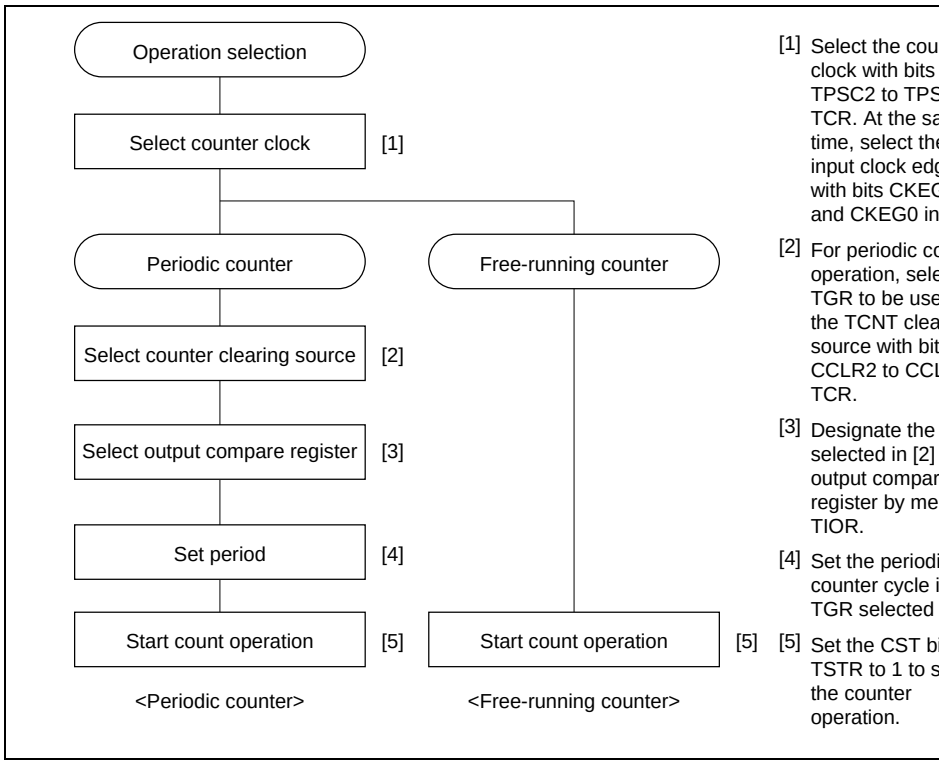


Figure 10.6 Example of Counter Operation Setting Procedure

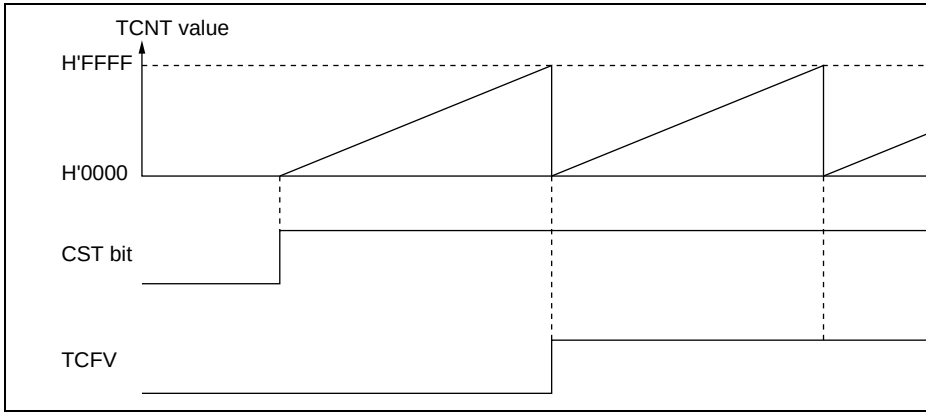


Figure 10.7 Free-Running Counter Operation

When compare match is selected as the TCNT clearing source, the TCNT counter for channel performs periodic count operation. The TGR register for setting the period is as an output compare register, and counter clearing by compare match is selected by n CCLR2 to CCLR0 in TCR. After the settings have been made, TCNT starts up-count periodic counter when the corresponding bit in TSTR is set to 1. When the count valu the value in TGR, the TGF bit in TSR is set to 1 and TCNT is cleared to H'0000.

If the value of the corresponding TGIE bit in TIER is 1 at this point, the TPU requests After a compare match, TCNT starts counting up again from H'0000.

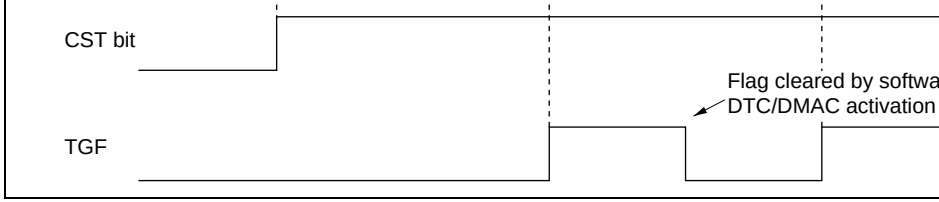
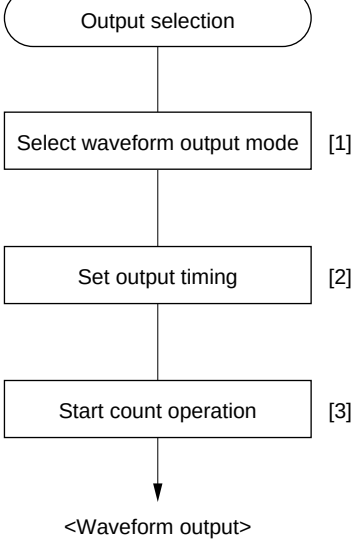


Figure 10.8 Periodic Counter Operation



[1] Select initial value 0 output or 1 output, a compare match output value 0 output, 1 or toggle output, by means of TIOR. The initial value is output at the TIOC pin unti first compare match occurs.

[2] Set the timing for compare match genera TGR.

[3] Set the CST bit in TSTR to 1 to start the operation.

Figure 10.9 Example of Setting Procedure for Waveform Output by Compar

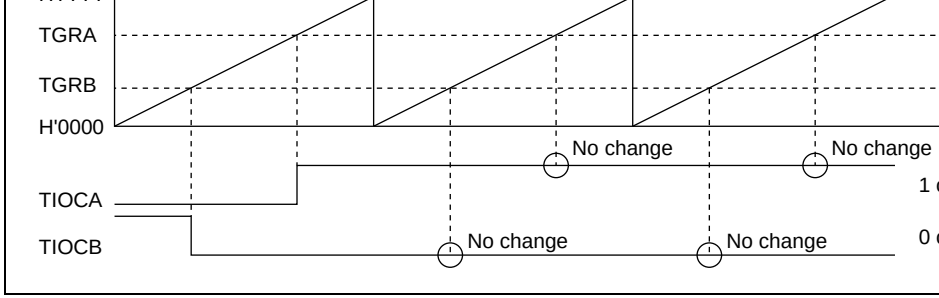


Figure 10.10 Example of 0 Output/1 Output Operation

Figure 10.11 shows an example of toggle output.

In this example TCNT has been designated as a periodic counter (with counter clearing by compare match B), and settings have been made so that output is toggled by both co match A and compare match B.

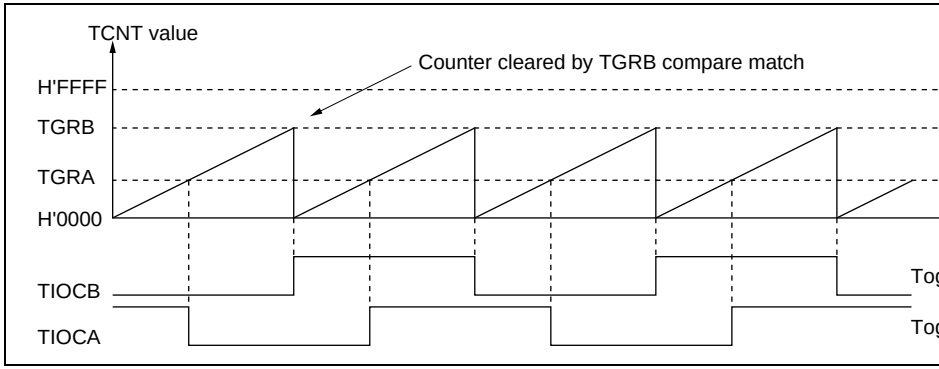


Figure 10.11 Example of Toggle Output Operation

0 and 3, ϕ/1 should not be selected as the counter input clock used for input ca
Input capture will not be generated if ϕ/1 is selected.

Example of input capture operation setting procedure: Figure 10.12 shows an exa
input capture operation setting procedure.

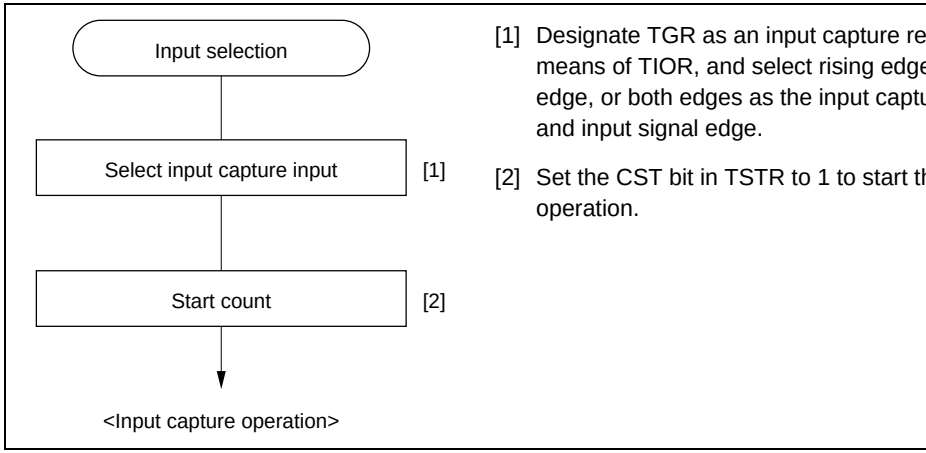


Figure 10.12 Example of Input Capture Operation Setting Procedur

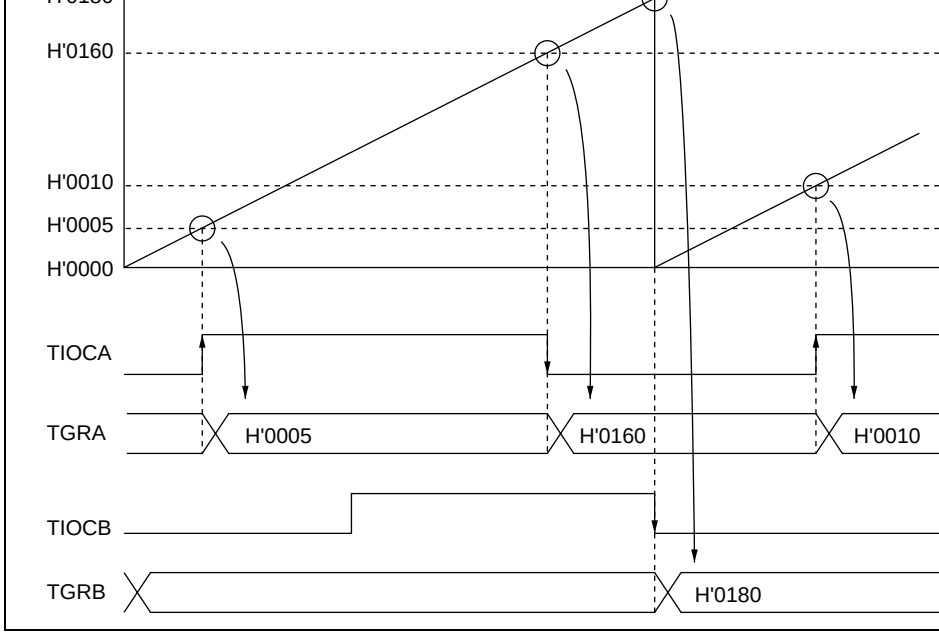


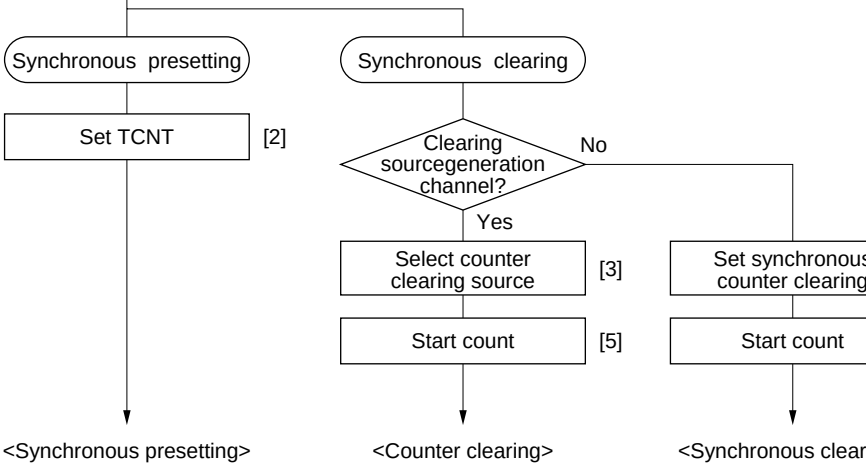
Figure 10.13 Example of Input Capture Operation

10.4.3 Synchronous Operation

In synchronous operation, the values in a number of TCNT counters can be rewritten simultaneously (synchronous presetting). Also, a number of TCNT counters can be clea simultaneously by making the appropriate setting in TCR (synchronous clearing).

Synchronous operation enables TGR to be incremented with respect to a single time ba

Channels 0 to 5 can all be designated for synchronous operation.



[1] Set to 1 the SYNC bits in TSYR corresponding to the channels to be designated for synchron operation.

[2] When the TCNT counter of any of the channels designated for synchronous operation is w the same value is simultaneously written to the other TCNT counters.

[3] Use bits CCLR2 to CCLR0 in TCR to specify TCNT clearing by input capture/output compa

[4] Use bits CCLR2 to CCLR0 in TCR to designate synchronous clearing for the counter clear

[5] Set to 1 the CST bits in TSTR for the relevant channels, to start the count operation.

Figure 10.14 Example of Synchronous Operation Setting Procedure

time, synchronous presetting, and synchronous clearing by TGR0B compare match, is p
for channel 0 to 2 TCNT counters, and the data set in TGR0B is used as the PWM cycl

For details of PWM modes, see section 10.4.6, PWM Modes.

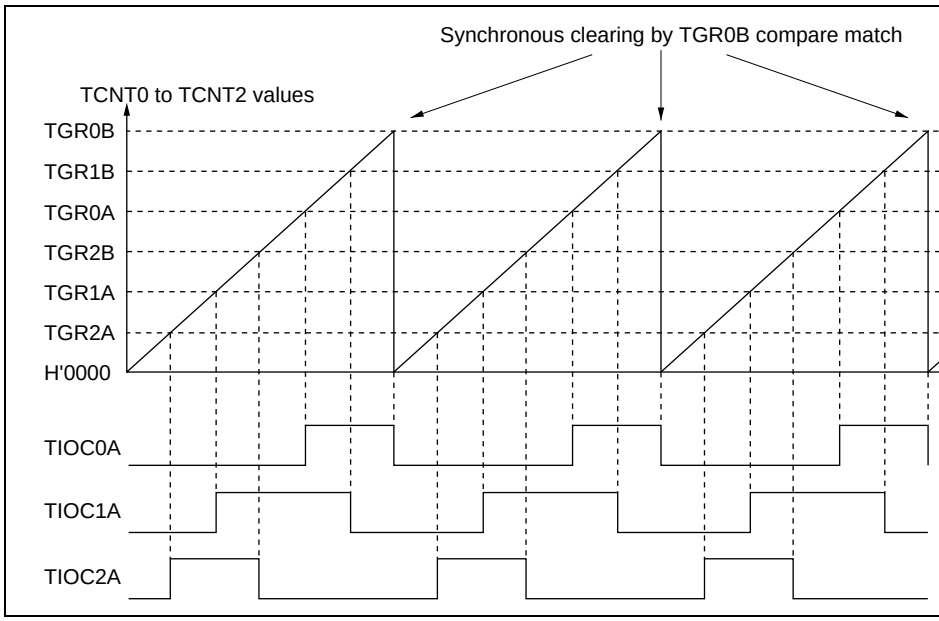


Figure 10.15 Example of Synchronous Operation

Table 10.5 Register Combinations in Buffer Operation

Channel	Timer General Register	Buffer Register
0	TGR0A	TGR0C
	TGR0B	TGR0D
3	TGR3A	TGR3C
	TGR3B	TGR3D

- When TGR is an output compare register

  When a compare match occurs, the value in the buffer register for the correspondi
  transferred to the timer general register.

  This operation is illustrated in figure 10.16.

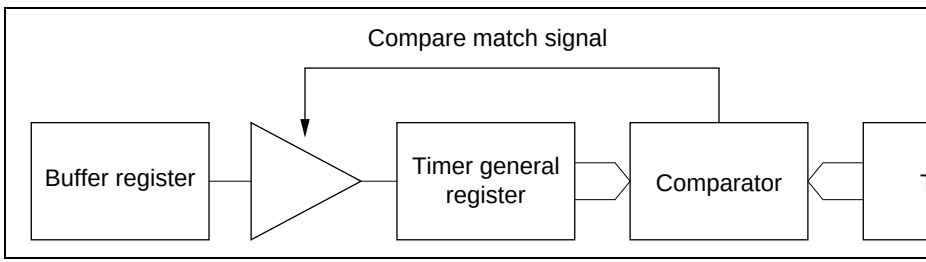


Figure 10.16 Compare Match Buffer Operation

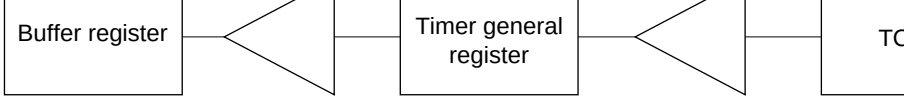
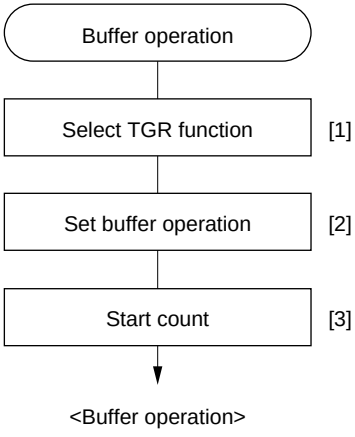


Figure 10.17 Input Capture Buffer Operation

Example of Buffer Operation Setting Procedure

Figure 10.18 shows an example of the buffer operation setting procedure.



[1] Designate TGR as an input capture regist
output compare register by means of TIO

[2] Designate TGR for buffer operation with b
BFA and BFB in TMDR.

[3] Set the CST bit in TSTR to 1 to start the c
operation.

Figure 10.18 Example of Buffer Operation Setting Procedure

operation is repeated each time compare match A occurs.

For details of PWM modes, see section 10.4.6, PWM Modes.

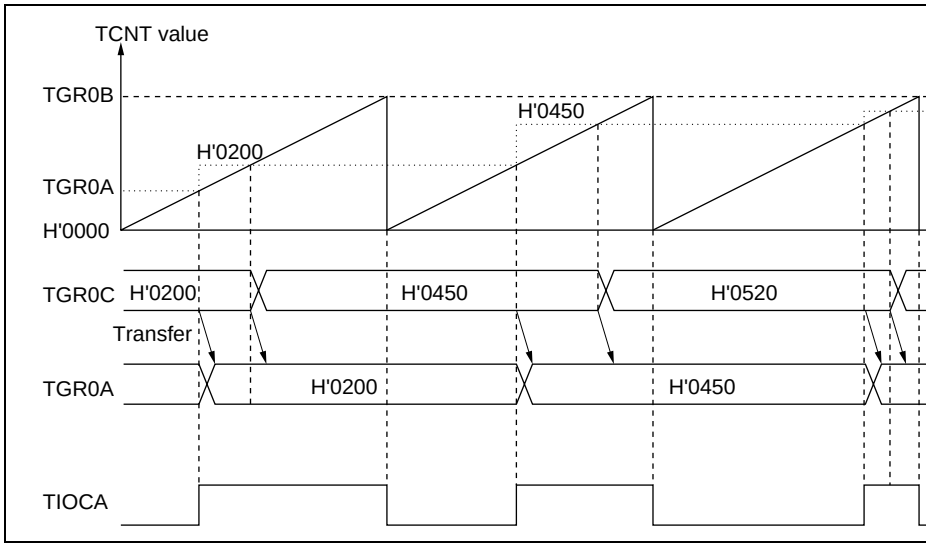


Figure 10.19 Example of Buffer Operation (1)

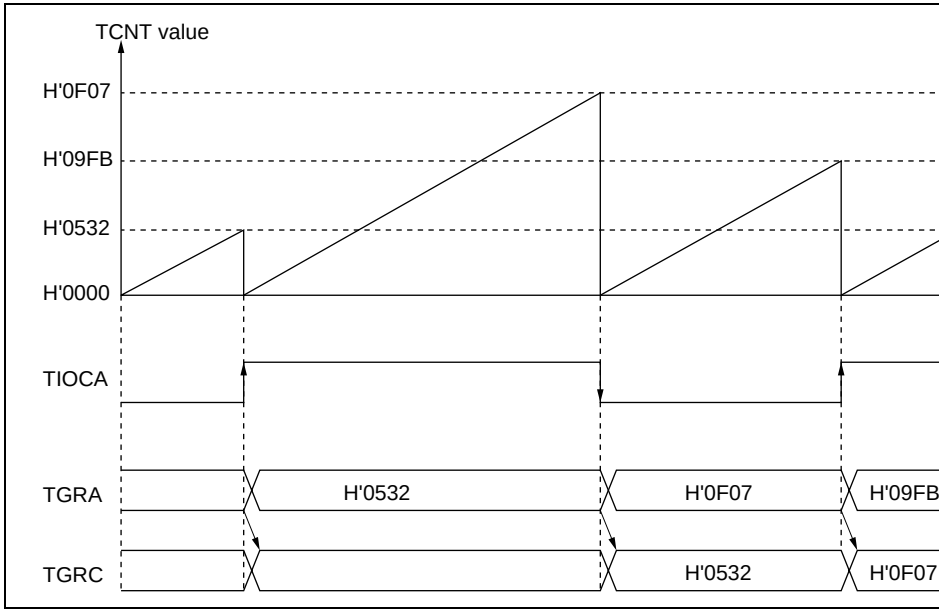


Figure 10.20 Example of Buffer Operation (2)

Table 10.6 shows the register combinations used in cascaded operation.

Note: When phase counting mode is set for channel 1 or 4, the counter clock setting and the counter operates independently in phase counting mode.

Table 10.6 Cascaded Combinations

Combination	Upper 16 Bits	Lower 16 Bits
Channels 1 and 2	TCNT1	TCNT2
Channels 4 and 5	TCNT4	TCNT5

Example of Cascaded Operation Setting Procedure

Figure 10.21 shows an example of the setting procedure for cascaded operation.

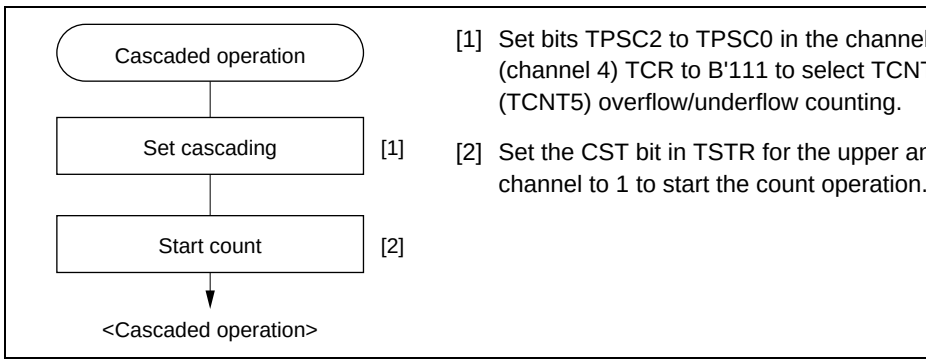


Figure 10.21 Cascaded Operation Setting Procedure

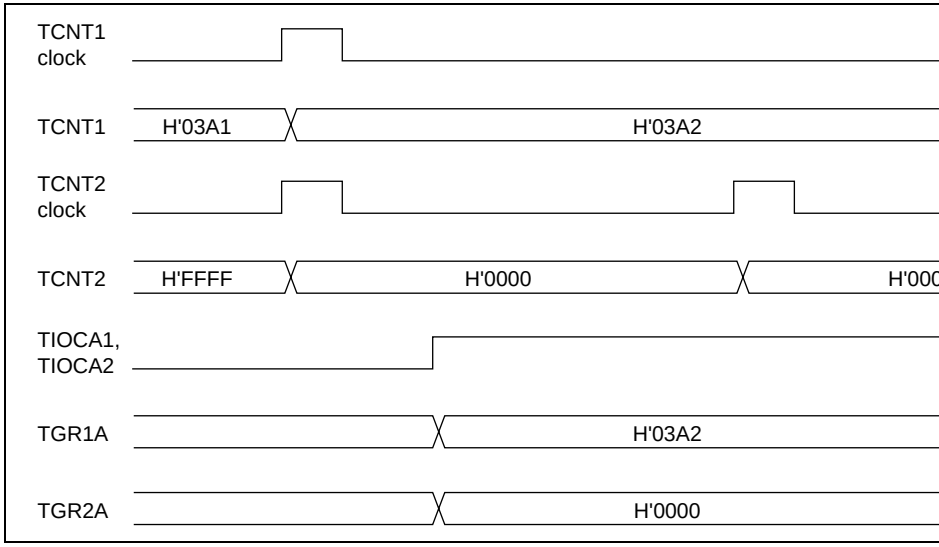


Figure 10.22 Example of Cascaded Operation (1)

Figure 10.23 illustrates the operation when counting upon TCNT2 overflow/underflow set for TCNT1, and phase counting mode has been designated for channel 2.

TCNT1 is incremented by TCNT2 overflow and decremented by TCNT2 underflow.

Figure 10.23 Example of Cascaded Operation (2)

10.4.6 PWM Modes

In PWM mode, PWM waveforms are output from the output pins. 0, 1, or toggle outp
selected as the output level in response to compare match of each TGR.

Designating TGR compare match as the counter clearing source enables the period to
register. All channels can be designated for PWM mode independently. Synchronous
also possible.

There are two PWM modes, as described below.

- PWM mode 1

  PWM output is generated from the TIOCA and TIOCC pins by pairing TGRA wit
  TGRC with TGRD. The output specified by bits IOA3 to IOA0 and IOC3 to IOC0
  output from the TIOCA and TIOCC pins at compare matches A and C, and the ou
  specified by bits IOB3 to IOB0 and IOD3 to IOD0 in TIOR is output at compare m
  and D. The initial output value is the value set in TGRA or TGRC. If the set value
  TGRs are identical, the output value does not change when a compare match occu

  In PWM mode 1, a maximum 8-phase PWM output is possible.

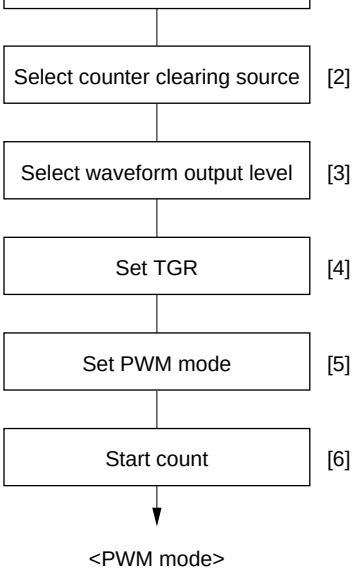
- PWM mode 2

  PWM output is generated using one TGR as the cycle register and the others as du
  The output specified in TIOR is performed by means of compare matches. Upon c
  clearing by a synchronization register compare match, the output value of each pin
  value set in TIOR. If the set values of the cycle and duty registers are identical, the
  value does not change when a compare match occurs.

  In PWM mode 2, a maximum 15-phase PWM output is possible by combined use
  synchronous operation.

	TGR0C	TIOCC0	TIOCC0
	TGR0D		TIOCD0
1	TGR1A	TIOCA1	TIOCA1
	TGR1B		TIOCB1
2	TGR2A	TIOCA2	TIOCA2
	TGR2B		TIOCB2
3	TGR3A	TIOCA3	TIOCA3
	TGR3B		TIOCB3
	TGR3C	TIOCC3	TIOCC3
	TGR3D		TIOCD3
4	TGR4A	TIOCA4	TIOCA4
	TGR4B		TIOCB4
5	TGR5A	TIOCA5	TIOCA5
	TGR5B		TIOCB5

Note: In PWM mode 2, PWM output is not possible for the TGR register in which the p



[2] Use bits CCLR2 to CCLR0 in TCR to sel
TGR to be used as the TCNT clearing so

[3] Use TIOR to designate the TGR as an o
compare register, and select the initial va
output value.

[4] Set the cycle in the TGR selected in [2],
the duty in the other the TGR.

[5] Select the PWM mode with bits MD3 to M
TMDR.

[6] Set the CST bit in TSTR to 1 to start the
operation.

Figure 10.24 Example of PWM Mode Setting Procedure

Examples of PWM Mode Operation

Figure 10.25 shows an example of PWM mode 1 operation.

In this example, TGRA compare match is set as the TCNT clearing source, 0 is set for
initial output value and output value, and 1 is set as the TGRB output value.

In this case, the value set in TGRA is used as the period, and the values set in TGRB r
the duty.

Figure 10.25 Example of PWM Mode Operation (1)

Figure 10.26 shows an example of PWM mode 2 operation.

In this example, synchronous operation is designated for channels 0 and 1, TGR1B con is set as the TCNT clearing source, and 0 is set for the initial output value and 1 for the value of the other TGR registers (TGR0A to TGR0D, TGR1A), to output a 5-phase PW waveform.

In this case, the value set in TGR1B is used as the cycle, and the values set in the other the duty.

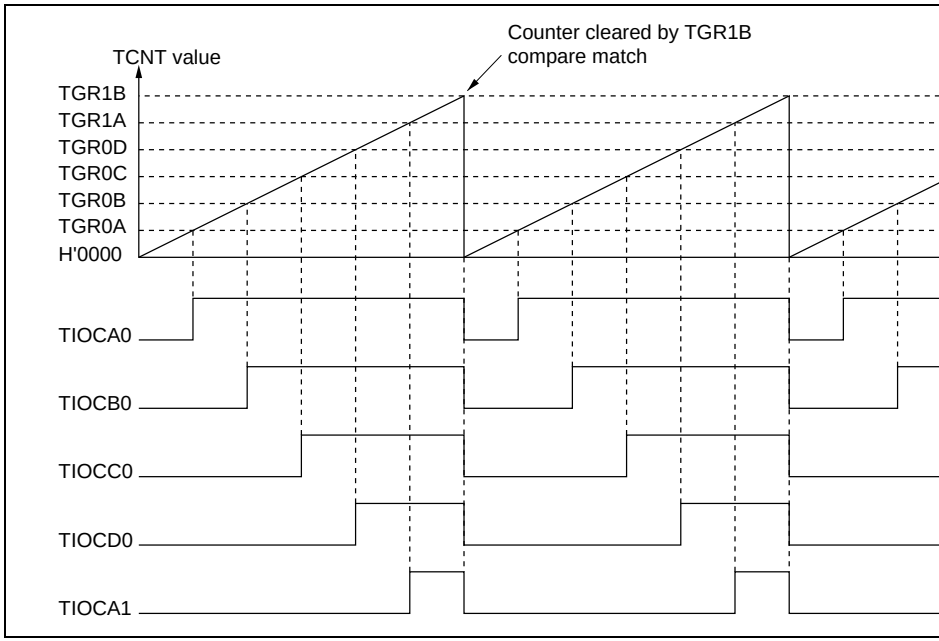


Figure 10.26 Example of PWM Mode Operation (2)

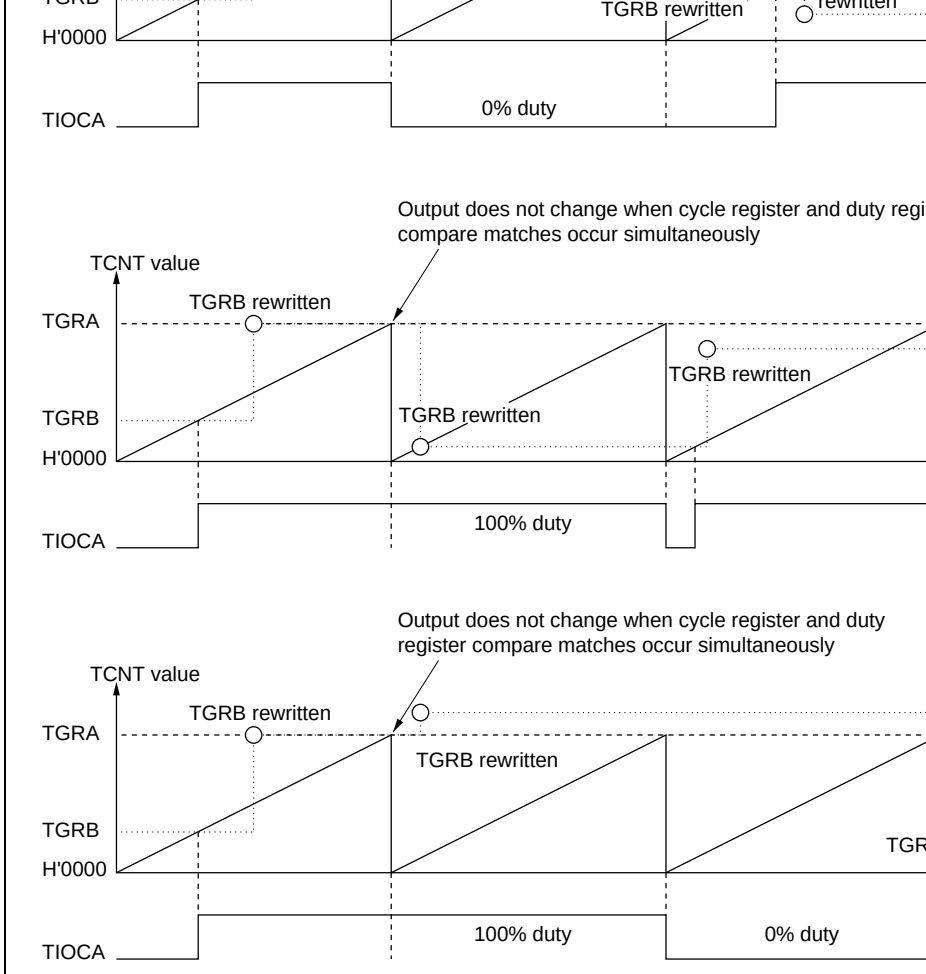


Figure 10.27 Example of PWM Mode Operation (3)

used.

When overflow occurs while TCNT is counting up, the TCFV flag in TSR is set; when
occurs while TCNT is counting down, the TCFU flag is set.

The TCFD bit in TSR is the count direction flag. Reading the TCFD flag provides an in
whether TCNT is counting up or down.

Table 10.8 shows the correspondence between external clock pins and channels.

Table 10.8 Phase Counting Mode Clock Input Pins

	External Clock Pin	
Channels	**A-Phase**	**B-Phase**
When channel 1 or 5 is set to phase counting mode	TCLKA	TCLKB
When channel 2 or 4 is set to phase counting mode	TCLKC	TCLKD

Example of Phase Counting Mode Setting Procedure

Figure 10.28 shows an example of the phase counting mode setting procedure.

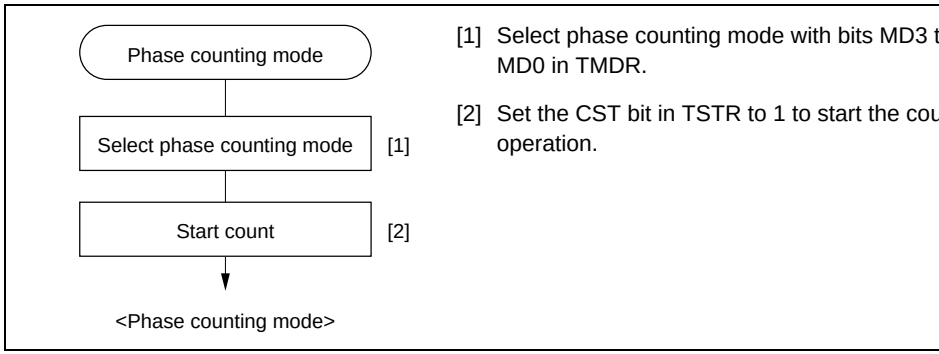


Figure 10.28 Example of Phase Counting Mode Setting Procedure

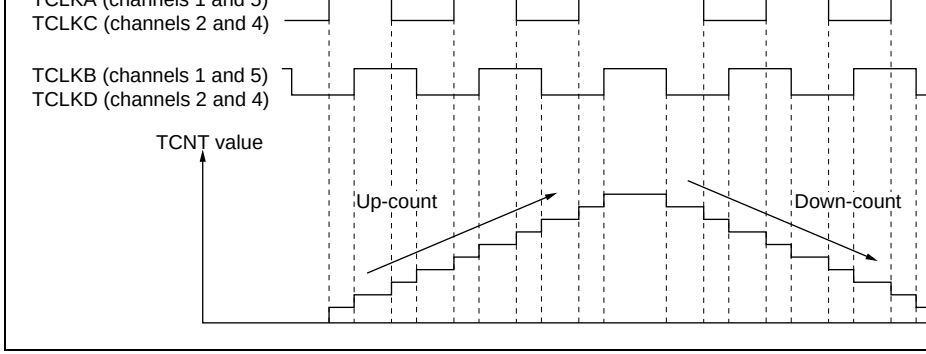


Figure 10.29 Example of Phase Counting Mode 1 Operation

Table 10.9 Up/Down-Count Conditions in Phase Counting Mode 1

TCLKA (Channels 1 and 5) TCLKC (Channels 2 and 4)	TCLKB (Channels 1 and 5) TCLKD (Channels 2 and 4)	Operation
High level	↑	Up-count
Low level	↓	
↑	Low level	
↓	High level	
High level	↓	Down-coun
Low level	↑	
↑	High level	
↓	Low level	

Legend:
↑: Rising edge
↓: Falling edge

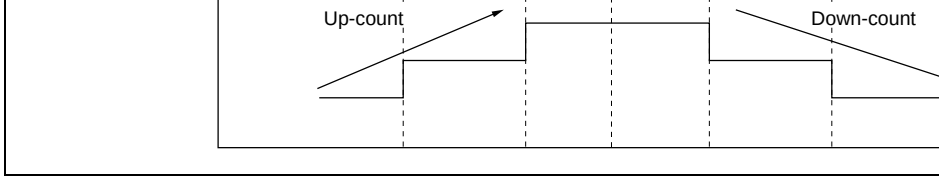


Figure 10.30 Example of Phase Counting Mode 2 Operation

Table 10.10 Up/Down-Count Conditions in Phase Counting Mode 2

TCLKA (Channels 1 and 5) TCLKC (Channels 2 and 4)	TCLKB (Channels 1 and 5) TCLKD (Channels 2 and 4)	Operation
High level	↑	Don't care
Low level	↓	
↑	Low level	
↓	High level	Up-count
High level	↓	Don't care
Low level	↑	
↑	High level	
↓	Low level	Down-count

Legend:
↑: Rising edge
↓: Falling edge

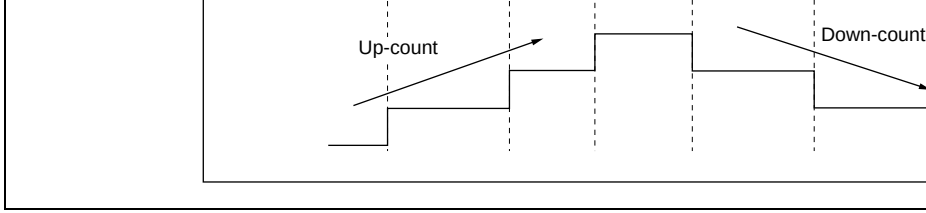


Figure 10.31 Example of Phase Counting Mode 3 Operation

Table 10.11 Up/Down-Count Conditions in Phase Counting Mode 3

TCLKA (Channels 1 and 5) TCLKC (Channels 2 and 4)	TCLKB (Channels 1 and 5) TCLKD (Channels 2 and 4)	Operation
High level	↑	Don't care
Low level	↓	
↑	Low level	
↓	High level	Up-count
High level	↓	Down-coun
Low level	↑	Don't care
↑	High level	
↓	Low level	

Legend:
↑: Rising edge
↓: Falling edge

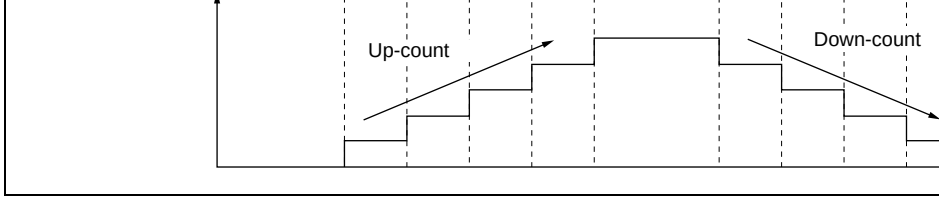


Figure 10.32 Example of Phase Counting Mode 4 Operation

Table 10.12 Up/Down-Count Conditions in Phase Counting Mode 4

TCLKA (Channels 1 and 5) TCLKC (Channels 2 and 4)	TCLKB (Channels 1 and 5) TCLKD (Channels 2 and 4)	Operation
High level	↑	Up-count
Low level	↓	
↑	Low level	Don't care
↓	High level	
High level	↓	Down-count
Low level	↑	
↑	High level	Don't care
↓	Low level	

Legend:
↑: Rising edge
↓: Falling edge

Channel 0 operates with TCNT counter clearing by TGR0C compare match; TGR0A
are used for the compare match function, and are set with the speed control period and
control period. TGR0B is used for input capture, with TGR0B and TGR0D operating
mode. The channel 1 counter input clock is designated as the TGR0B input capture so
detection of the pulse width of 2-phase encoder 4-multiplication pulses is performed.

TGR1A and TGR1B for channel 1 are designated for input capture, channel 0 TGR0A
TGR0C compare matches are selected as the input capture source, and store the up/do
values for the control periods.

This procedure enables accurate position/speed detection to be achieved.

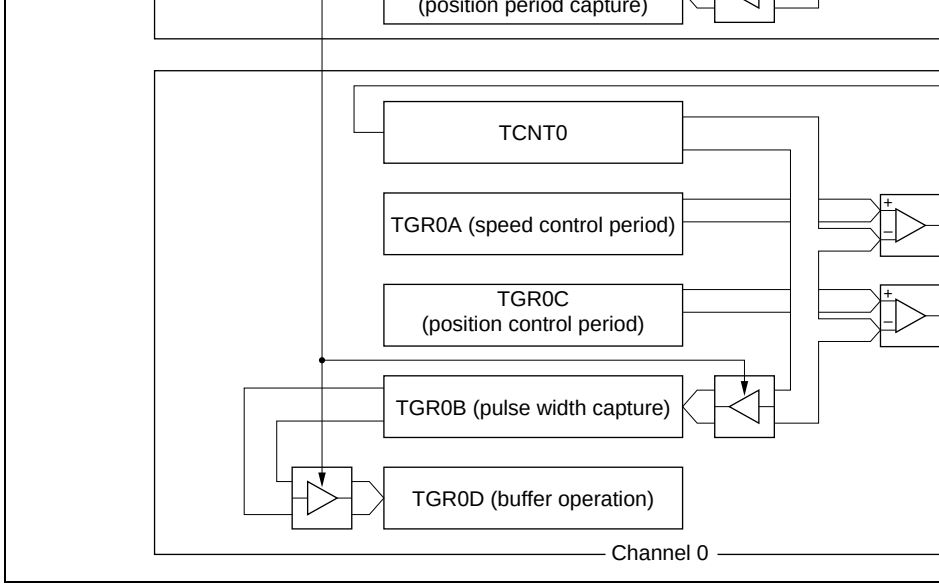


Figure 10.33 Phase Counting Mode Application Example

When an interrupt request is generated, the corresponding status flag in TSR is set to 1
corresponding enable/disable bit in TIER is set to 1 at this time, an interrupt is request
interrupt request is cleared by clearing the status flag to 0.

Relative channel priorities can be changed by the interrupt controller, but the priority o
a channel is fixed. For details, see section 5, Interrupt Controller.

Table 10.13 lists the TPU interrupt sources.

	TCI0V	TCNT0 overflow	Not possible	Not possib
1	TGI1A	TGR1A input capture/compare match	Possible	Possible
	TGI1B	TGR1B input capture/compare match	Not possible	Possible
	TCI1V	TCNT1 overflow	Not possible	Not possib
	TCI1U	TCNT1 underflow	Not possible	Not possib
2	TGI2A	TGR2A input capture/compare match	Possible	Possible
	TGI2B	TGR2B input capture/compare match	Not possible	Possible
	TCI2V	TCNT2 overflow	Not possible	Not possib
	TCI2U	TCNT2 underflow	Not possible	Not possib
3	TGI3A	TGR3A input capture/compare match	Possible	Possible
	TGI3B	TGR3B input capture/compare match	Not possible	Possible
	TGI3C	TGR3C input capture/compare match	Not possible	Possible
	TGI3D	TGR3D input capture/compare match	Not possible	Possible
	TCI3V	TCNT3 overflow	Not possible	Not possib
4	TGI4A	TGR4A input capture/compare match	Possible	Possible
	TGI4B	TGR4B input capture/compare match	Not possible	Possible
	TCI4V	TCNT4 overflow	Not possible	Not possib
	TCI4U	TCNT4 underflow	Not possible	Not possib
5	TGI5A	TGR5A input capture/compare match	Possible	Possible
	TGI5B	TGR5B input capture/compare match	Not possible	Possible
	TCI5V	TCNT5 overflow	Not possible	Not possib
	TCI5U	TCNT5 underflow	Not possible	Not possib

Note: This table shows the initial state immediately after a reset. The relative channel p
can be changed by the interrupt controller.

An interrupt is requested if the TCIEV bit in TIER is set to 1 when the TCFV flag in T
1 by the occurrence of TCNT overflow on a channel. The interrupt request is cleared b
the TCFV flag to 0. The TPU has six overflow interrupts, one for each channel.

Underflow Interrupt

An interrupt is requested if the TCIEU bit in TIER is set to 1 when the TCFU flag in T
1 by the occurrence of TCNT underflow on a channel. The interrupt request is cleared
the TCFU flag to 0. The TPU has four underflow interrupts, one each for channels 1, 2

10.5.2 DTC/DMAC Activation

DTC Activation

The DTC can be activated by the TGR input capture/compare match interrupt for a ch
details, see section 8, Data Transfer Controller.

A total of 16 TPU input capture/compare match interrupts can be used as DTC activat
four each for channels 0 and 3, and two each for channels 1, 2, 4, and 5.

DMAC Activation

The DMAC can be activated by the TGRA input capture/compare match interrupt for
For details, see section 7, DMA Controller.

In the TPU, a total of six TGRA input capture/compare match interrupts can be used a
activation sources, one for each channel.

In the TPU, a total of six TGRA input capture/compare match interrupts can be used as converter conversion start sources, one for each channel.

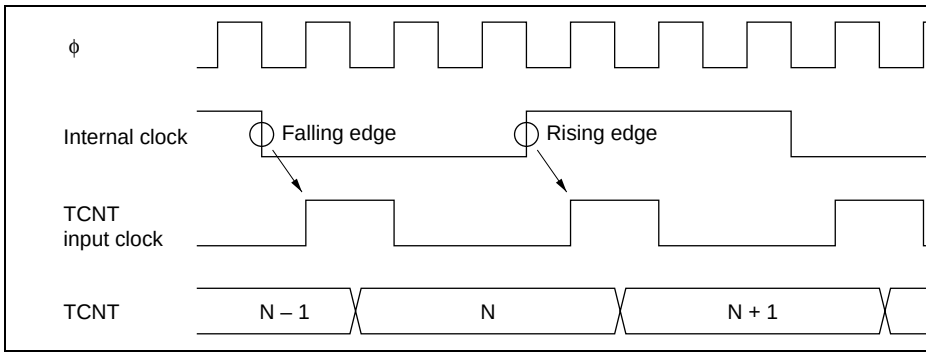


Figure 10.34 Count Timing in Internal Clock Operation

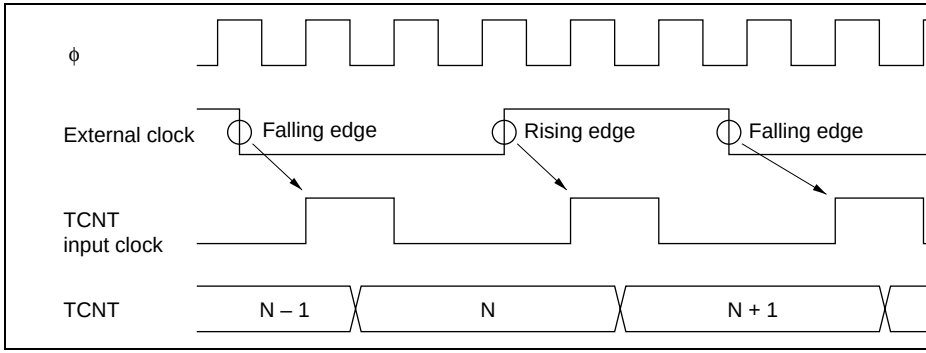


Figure 10.35 Count Timing in External Clock Operation

Output Compare Output Timing

A compare match signal is generated in the final state in which TCNT and TGR matcl at which the count value matched by TCNT is updated). When a compare match signa generated, the output value set in TIOR is output at the output compare output pin. Af between TCNT and TGR, the compare match signal is not generated until the TCNT i generated.

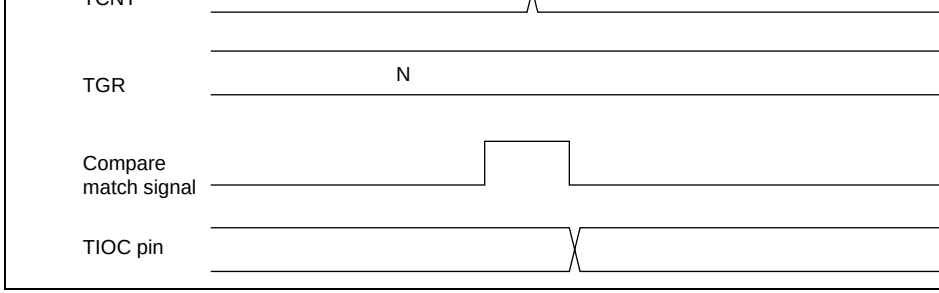


Figure 10.36 Output Compare Output Timing

Input Capture Signal Timing

Figure 10.37 shows input capture signal timing.

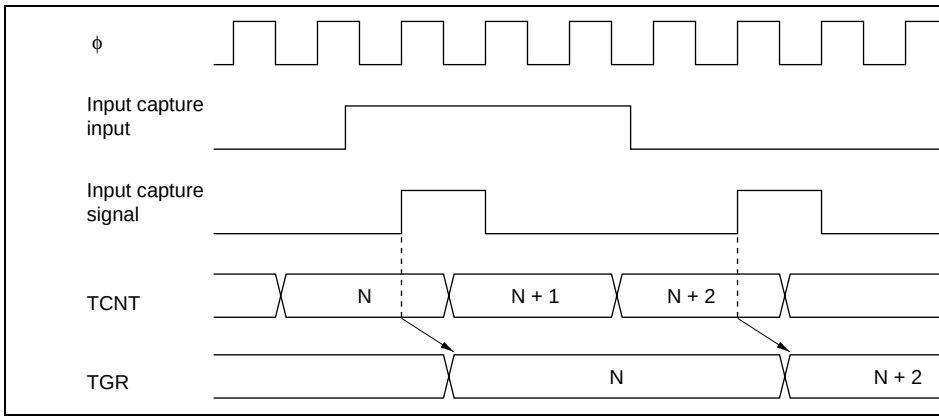


Figure 10.37 Input Capture Input Signal Timing

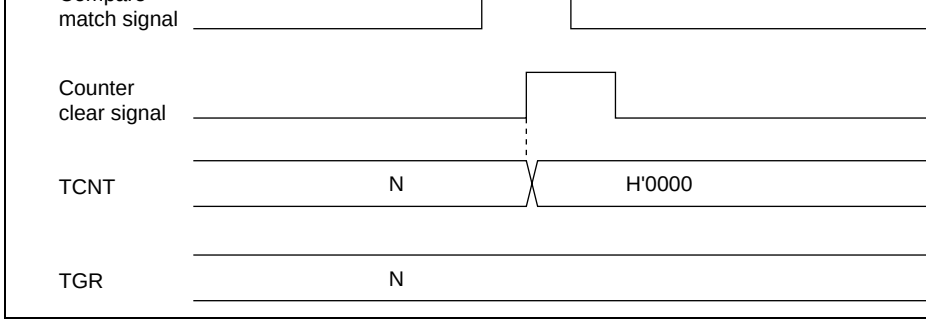


Figure 10.38 Counter Clear Timing (Compare Match)

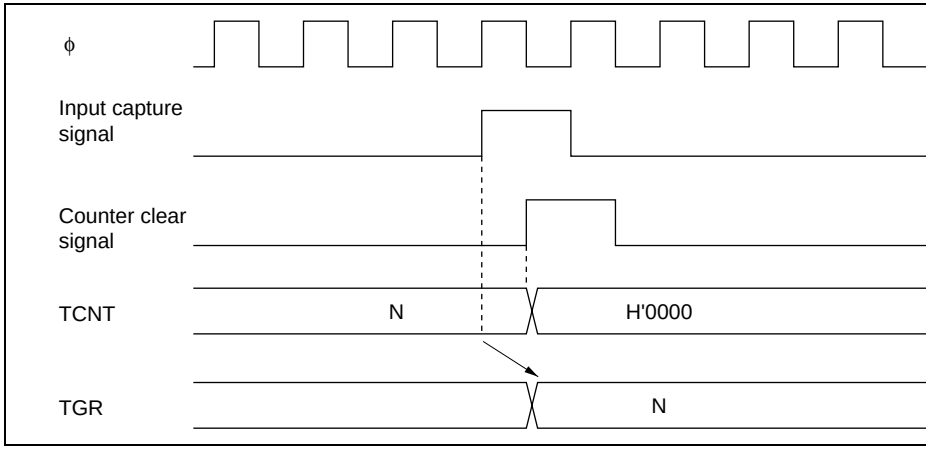


Figure 10.39 Counter Clear Timing (Input Capture)

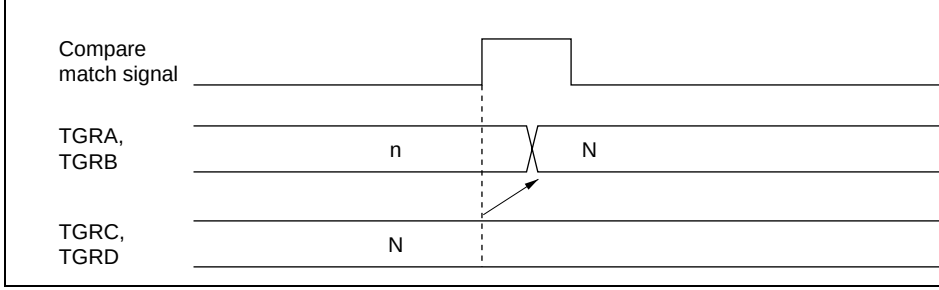


Figure 10.40 Buffer Operation Timing (Compare Match)

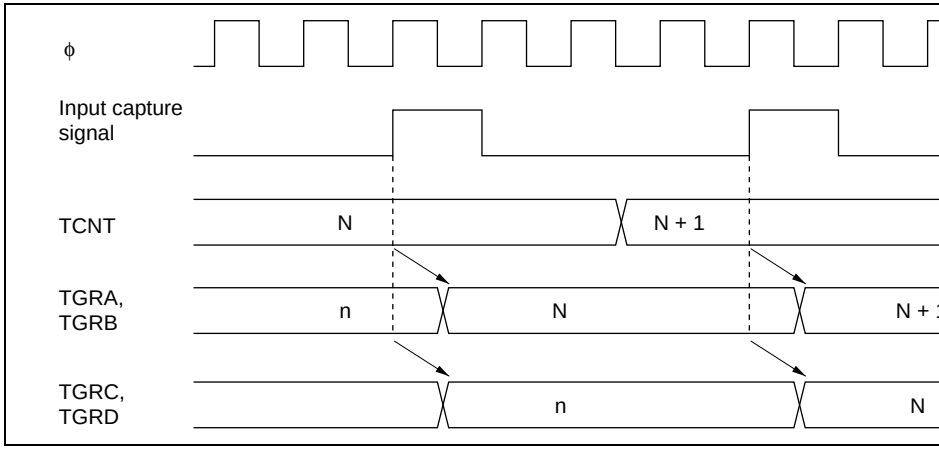


Figure 10.41 Buffer Operation Timing (Input Capture)

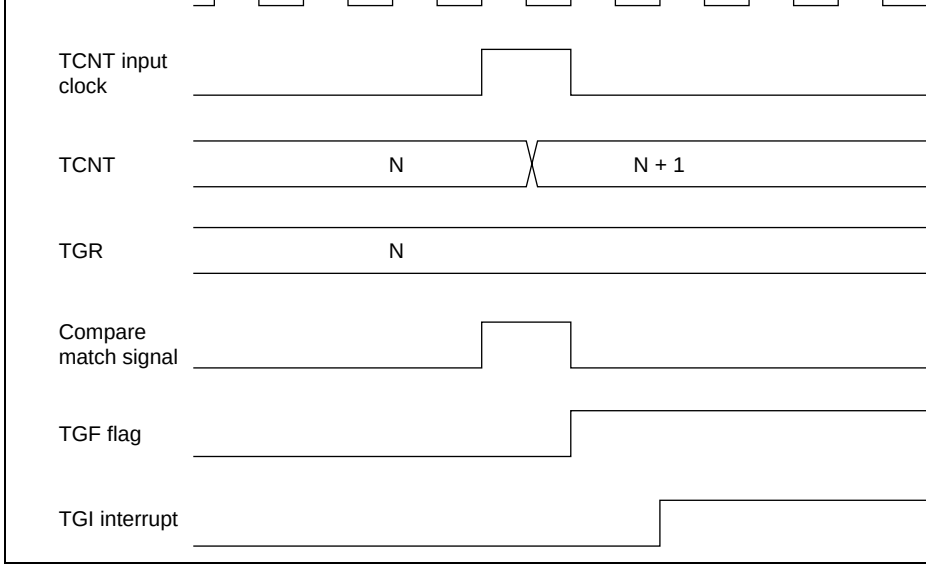


Figure 10.42 TGI Interrupt Timing (Compare Match)

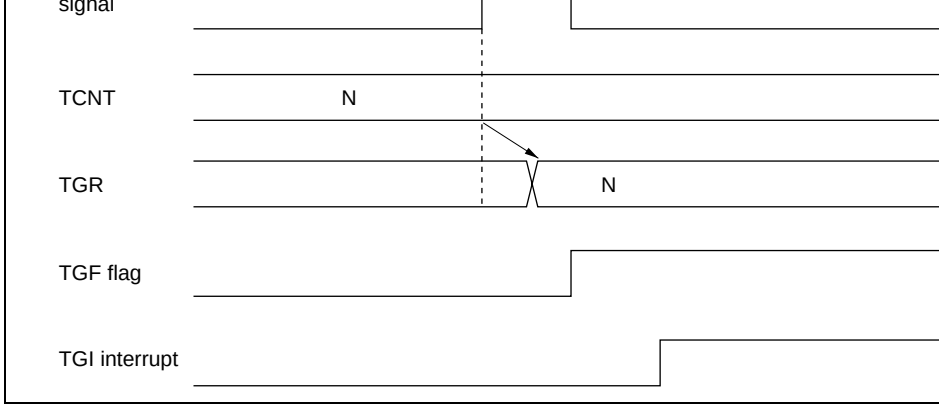


Figure 10.43 TGI Interrupt Timing (Input Capture)

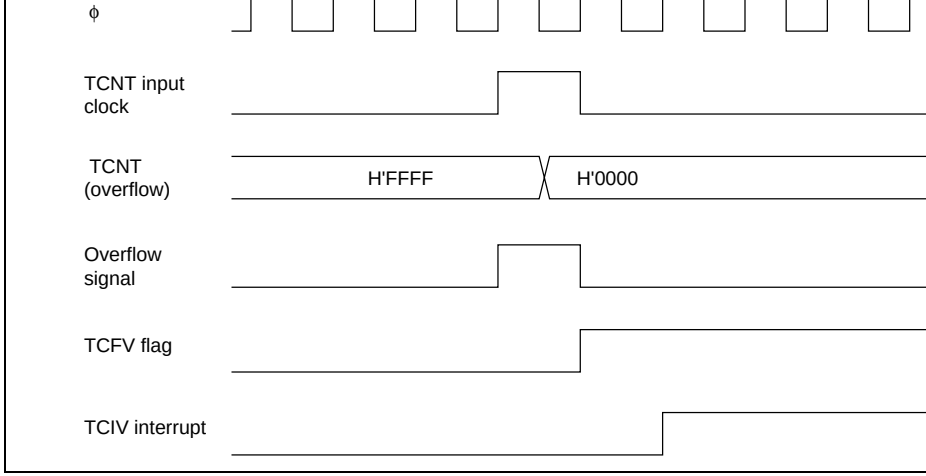


Figure 10.44 TCIV Interrupt Setting Timing

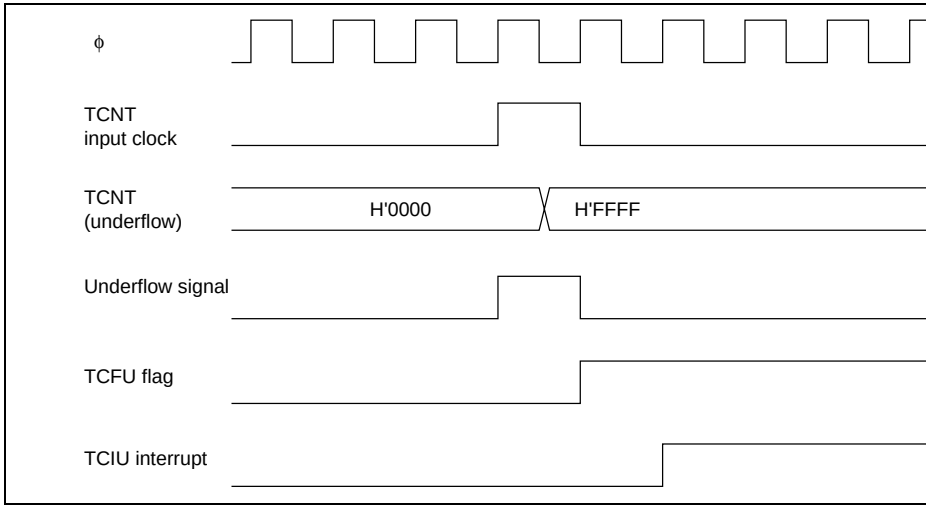


Figure 10.45 TCIU Interrupt Setting Timing

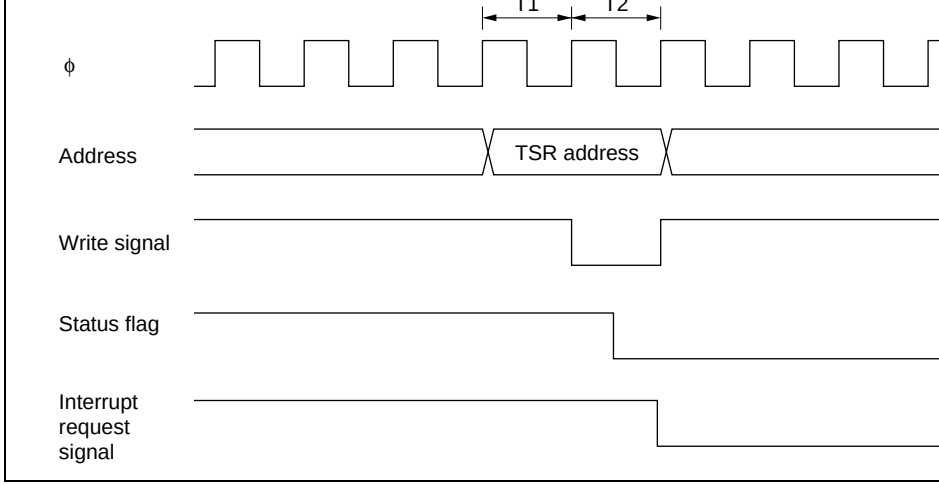


Figure 10.46 Timing for Status Flag Clearing by CPU

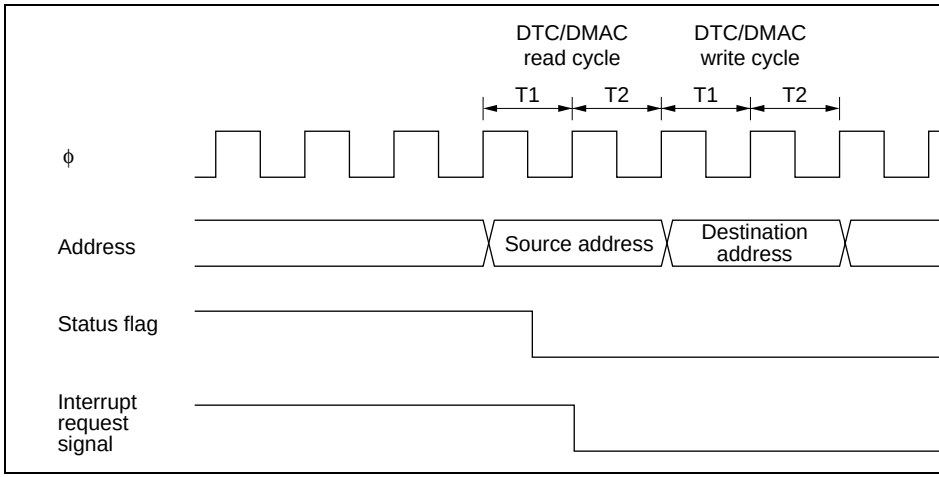
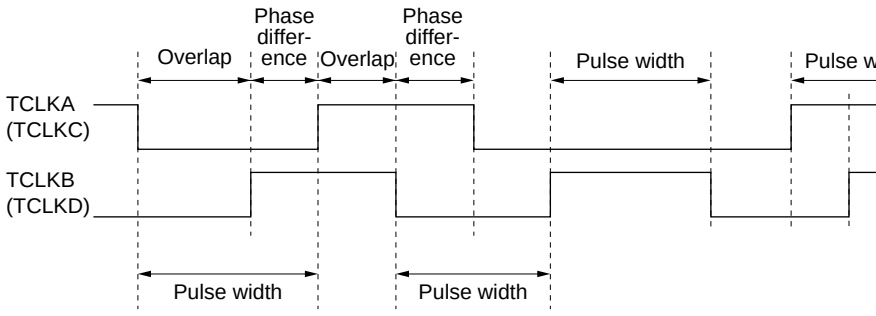


Figure 10.47 Timing for Status Flag Clearing by DTC/DMAC Activatio

narrower pulse width.

In phase counting mode, the phase difference and overlap between the two input clock least 1.5 states, and the pulse width must be at least 2.5 states. Figure 10.48 shows the conditions in phase counting mode.



Notes: Phase difference and overlap: 1.5 states or more
Pulse width : 2.5 states or more

Figure 10.48 Phase Difference, Overlap, and Pulse Width in Phase Countin

Caution on Period Setting

When counter clearing by compare match is set, TCNT is cleared in the final state in w matches the TGR value (the point at which the count value matched by TCNT is upda Consequently, the actual counter frequency is given by the following formula:

$$f = \frac{\phi}{(N + 1)}$$

Where f: Counter frequency
ϕ: Operating frequency
N: TGR set value

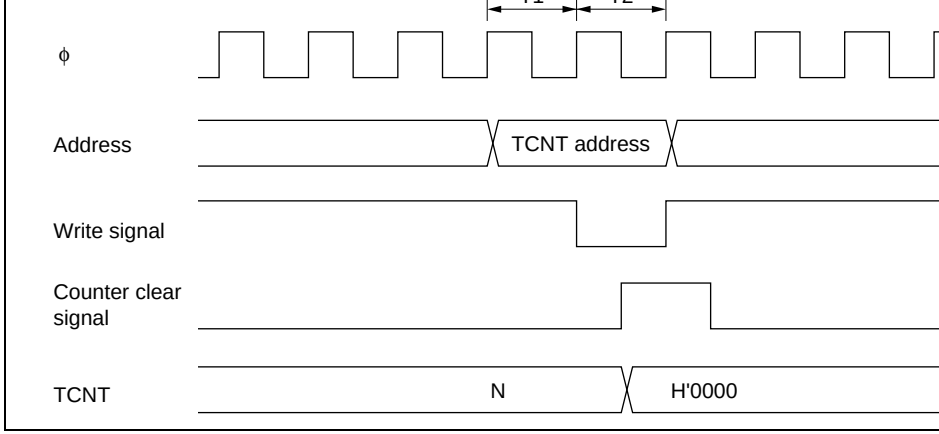


Figure 10.49 Contention between TCNT Write and Clear Operations

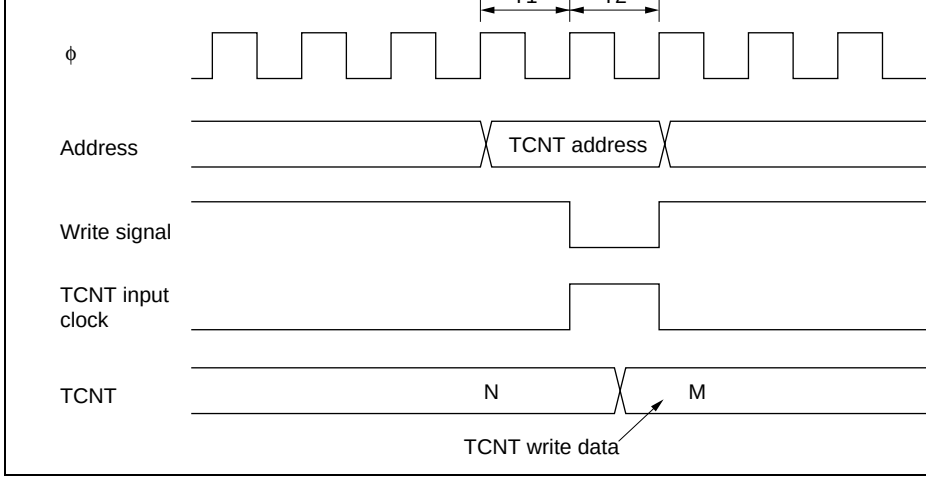


Figure 10.50 Contention between TCNT Write and Increment Operati

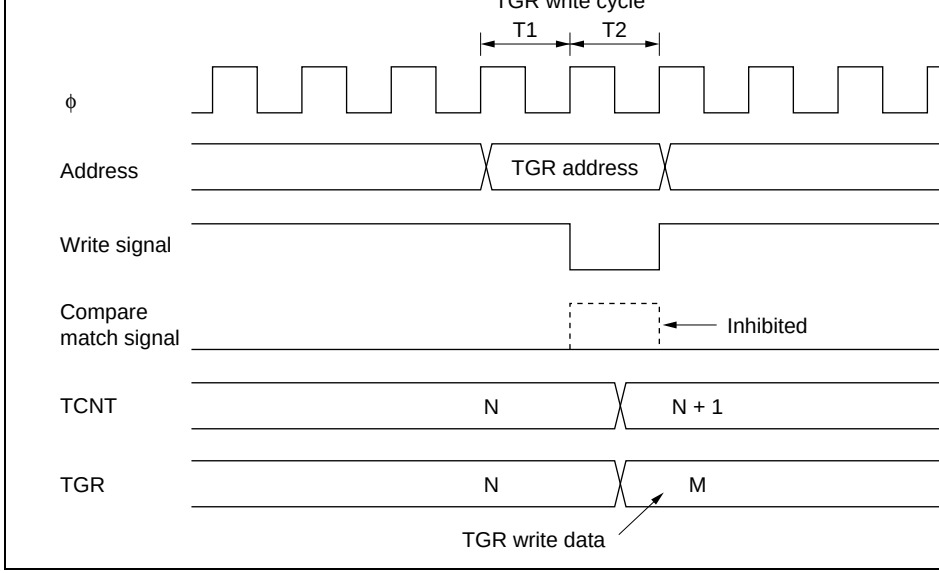


Figure 10.51 Contention between TGR Write and Compare Match

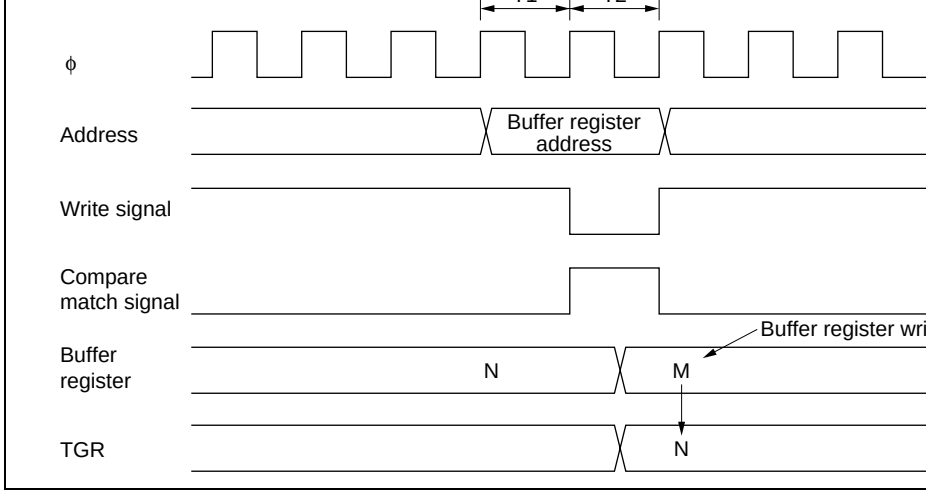


Figure 10.52 Contention between Buffer Register Write and Compare M

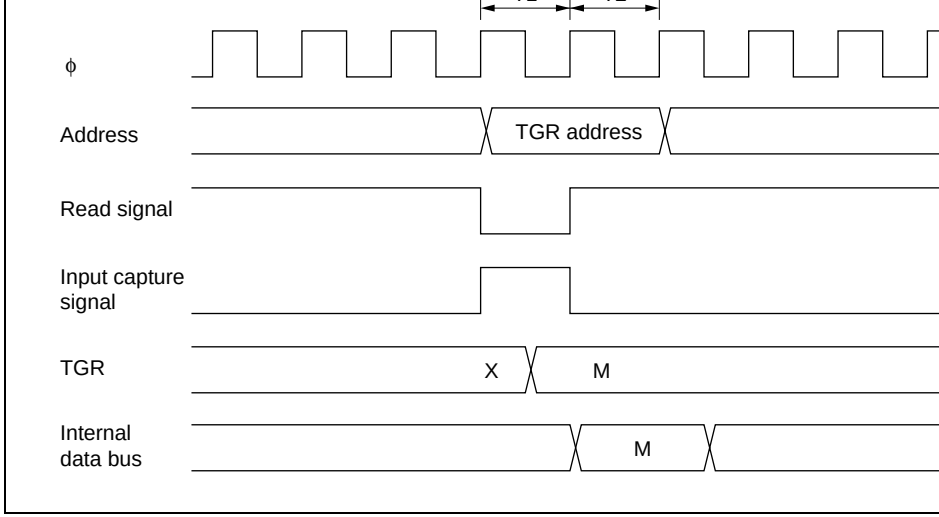


Figure 10.53 Contention between TGR Read and Input Capture

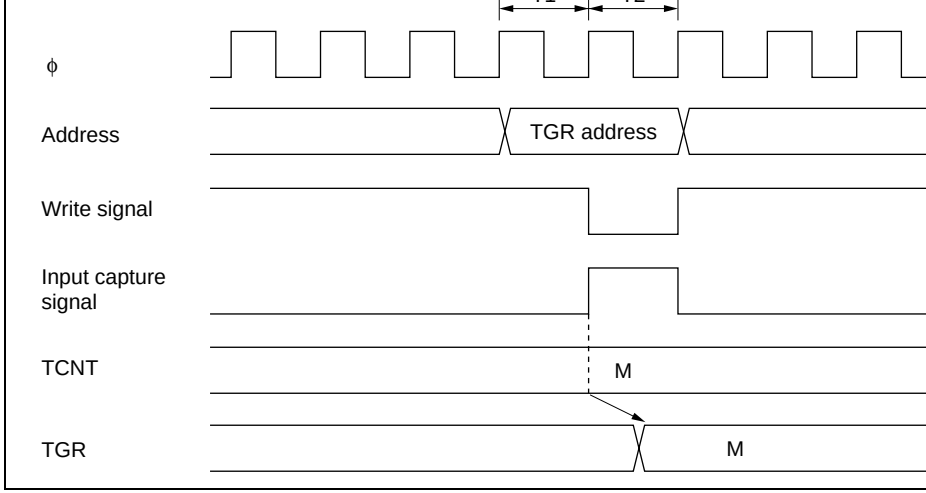


Figure 10.54 Contention between TGR Write and Input Capture

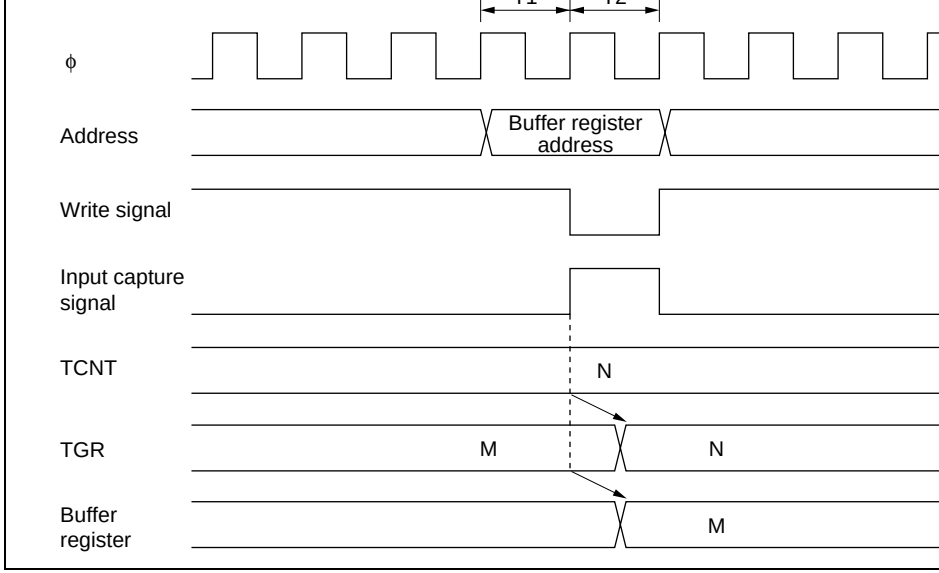


Figure 10.55 Contention between Buffer Register Write and Input Captu

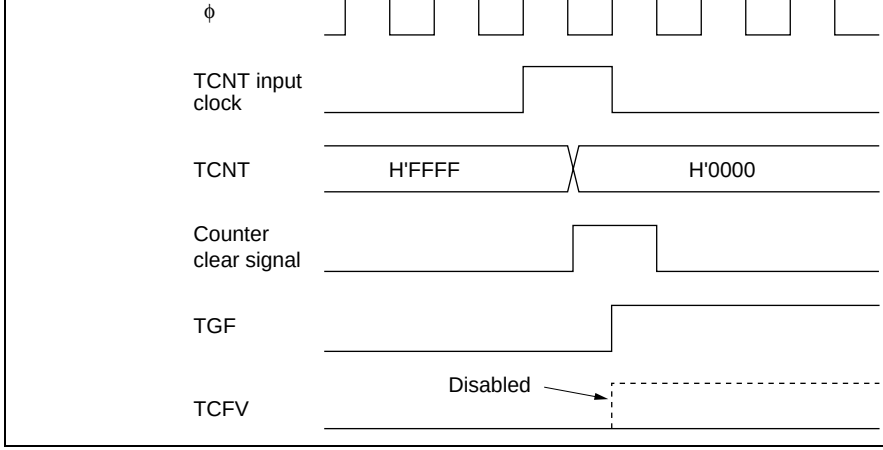


Figure 10.56 Contention between Overflow and Counter Clearing

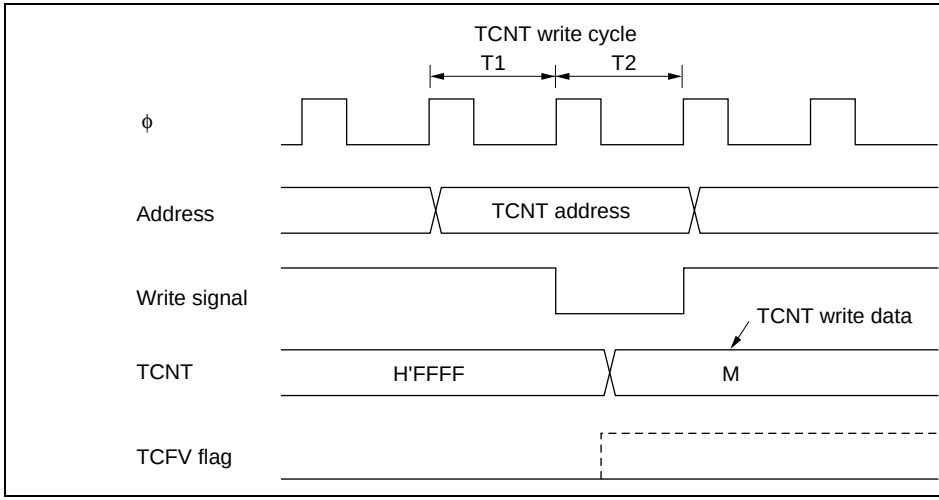


Figure 10.57 Contention between TCNT Write and Overflow

Multiplexing of I/O Pins

In the H8S/2655 Group, the TCLKA input pin is multiplexed with the TIOCC0 I/O pin TCLKB input pin with the TIOCD0 I/O pin, the TCLKC input pin with the TIOCB1 I/ the TCLKD input pin with the TIOCB2 I/O pin. When an external clock is input, comp output should not be performed from a multiplexed pin.

Interrupts and Module Stop Mode

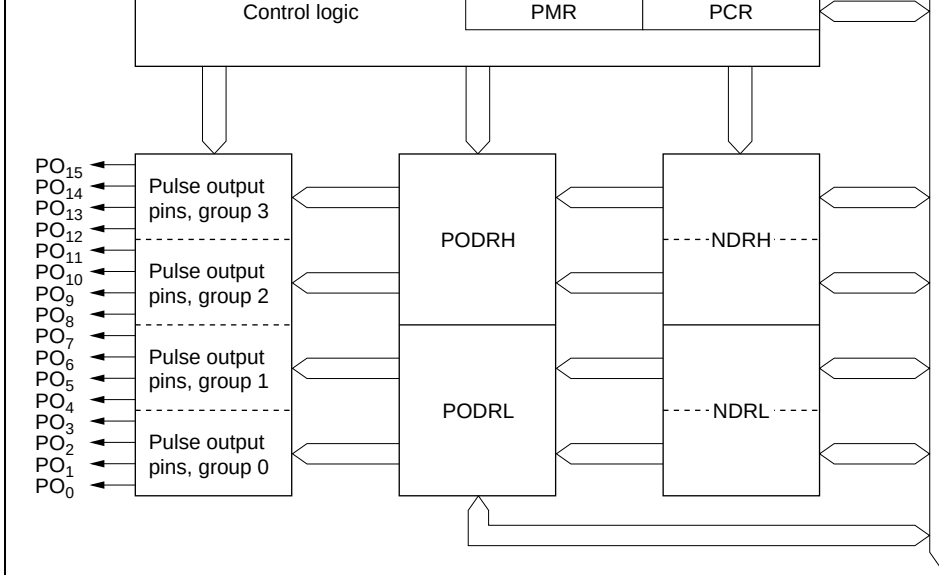
If module stop mode is set when an interrupt has been requested, the CPU interrupt sou DMAC/DTC activation source cannot be cleared. Interrupts should therefore be disable setting module stop mode.

independently.

11.1.1 Features

PPG features are listed below.

- 16-bit output data
  Maximum 16-bit data can be output, and output can be enabled on a bit-by-bit bas
- Four output groups
  Output trigger signals can be selected in 4-bit groups to provide up to four differen
  outputs.
- Selectable output trigger signals
  Output trigger signals can be selected for each group from the compare match sign
  TPU channels.
- Non-overlap mode
  A non-overlap margin can be provided between pulse outputs.
- Can operate together with the data transfer controller (DTC) and DMA controller (
  The compare match signals selected as output trigger signals can activate the DTC
  for sequential output of data without CPU intervention.
- Settable inverted output
  Inverted data can be output for each group.



Legend:

PMR	: PPG output mode register
PCR	: PPG output control register
NDERH	: Next data enable register H
NDERL	: Next data enable register L
NDRH	: Next data register H
NDRL	: Next data register L
PODRH	: Output data register H
PODRL	: Output data register L

Figure 11.1 Block Diagram of PPG

Pulse output 2	PO_2	Output	
Pulse output 3	PO_3	Output	
Pulse output 4	PO_4	Output	Group 1 pulse output
Pulse output 5	PO_5	Output	
Pulse output 6	PO_6	Output	
Pulse output 7	PO_7	Output	
Pulse output 8	PO_8	Output	Group 2 pulse output
Pulse output 9	PO_9	Output	
Pulse output 10	PO_{10}	Output	
Pulse output 11	PO_{11}	Output	
Pulse output 12	PO_{12}	Output	Group 3 pulse output
Pulse output 13	PO_{13}	Output	
Pulse output 14	PO_{14}	Output	
Pulse output 15	PO_{15}	Output	

Next data enable register H	NDERH	R/W	H'00	H'
Next data enable register L	NDERL	R/W	H'00	H'
Output data register H	PODRH	R/(W)*2	H'00	H'
Output data register L	PODRL	R/(W)*2	H'00	H'
Next data register H	NDRH	R/W	H'00	H' H'
Next data register L	NDRL	R/W	H'00	H' H'
Port 1 data direction register	P1DDR	W	H'00	H'
Port 2 data direction register	P2DDR	W	H'00	H'
Module stop control register	MSTPCR	R/W	H'3FFF	H'

Notes: 1. Lower 16 bits of the address.

2. Bits used for pulse output cannot be written to.

3. When the same output trigger is selected for pulse output groups 2 and 3 by setting, the NDRH address is H'FF4C. When the output triggers are different address is H'FF4E for group 2 and H'FF4C for group 3.

   Similarly, when the same output trigger is selected for pulse output groups 0 the PCR setting, the NDRL address is H'FF4D. When the output triggers are the NDRL address is H'FF4F for group 0 and H'FF4D for group 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

NDERL

Bit	:	7	6	5	4	3	2	1
		NDER7	NDER6	NDER5	NDER4	NDER3	NDER2	NDER1
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

NDERH and NDERL are 8-bit readable/writable registers that enable or disable pulse
bit-by-bit basis.

If a bit is enabled for pulse output by NDERH or NDERL, the NDR value is automati
transferred to the corresponding PODR bit when the TPU compare match event specif
occurs, updating the output value. If pulse output is disabled, the bit value is not transf
NDR to PODR and the output value does not change.

NDERH and NDERL are each initialized to H'00 by a reset and in hardware standby m
are not initialized in software standby mode.

NDERH Bits 7 to 0—Next Data Enable 15 to 8 (NDER15 to NDER8): These bits e
disable pulse output on a bit-by-bit basis.

Bits 7 to 0

NDER15 to NDER8	Description
0	Pulse outputs PO_{15} to PO_8 are disabled (NDR15 to NDR8 are transferred to POD15 to POD8)
1	Pulse outputs PO_{15} to PO_8 are enabled (NDR15 to NDR8 are to POD15 to POD8)

POD7 to POD0)

11.2.2 Output Data Registers H and L (PODRH, PODRL)

PODRH

Bit	:	7	6	5	4	3	2	1
		POD15	POD14	POD13	POD12	POD11	POD10	POD9
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)*

PODRL

Bit	:	7	6	5	4	3	2	1
		POD7	POD6	POD5	POD4	POD3	POD2	POD1
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: * A bit that has been set for pulse output by NDER is read-only.

PODRH and PODRL are 8-bit readable/writable registers that store output data for use output.

not initialized in software standby mode.

11.2.4 Notes on NDR Access

The NDRH and NDRL addresses differ depending on whether pulse output groups ha
output trigger or different output triggers.

	NDR15	NDR14	NDR13	NDR12	NDR11	NDR10	NDR9
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

- Address H'FF4E

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

If pulse output groups 0 and 1 are triggered by the same compare match event, the NDR is H'FF4D. The upper 4 bits belong to group 1 and the lower 4 bits to group 0. Address consists entirely of reserved bits that cannot be modified and are always read as 1.

- Address H'FF4D

Bit :	7	6	5	4	3	2	1
	NDR7	NDR6	NDR5	NDR4	NDR3	NDR2	NDR1
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

- Address H'FF4F

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

Bit	:	7	6	5	4	3	2	1
		NDR15	NDR14	NDR13	NDR12	—	—	—
Initial value	:	0	0	0	0	1	1	1
R/W	:	R/W	R/W	R/W	R/W	—	—	—

- Address H'FF4E

Bit	:	7	6	5	4	3	2	1
		—	—	—	—	NDR11	NDR10	NDR9
Initial value	:	1	1	1	1	0	0	0
R/W	:	—	—	—	—	R/W	R/W	R/W

If pulse output groups 0 and 1 are triggered by different compare match event, the add
upper 4 bits in NDRL (group 1) is H'FF4D and the address of the lower 4 bits (group
Bits 3 to 0 of address H'FF4D and bits 7 to 4 of address H'FF4F are reserved bits that
modified and are always read as 1.

- Address H'FF4D

Bit	:	7	6	5	4	3	2	1
		NDR7	NDR6	NDR5	NDR4	—	—	—
Initial value	:	0	0	0	0	1	1	1
R/W	:	R/W	R/W	R/W	R/W	—	—	—

- Address H'FF4F

Bit	:	7	6	5	4	3	2	1
		—	—	—	—	NDR3	NDR2	NDR1
Initial value	:	1	1	1	1	0	0	0
R/W	:	—	—	—	—	R/W	R/W	R/W

group-by-group basis.

PCR is initialized to H'FF by a reset and in hardware standby mode. It is not initialized
standby mode.

Bits 7 and 6—Group 3 Compare Match Select 1 and 0 (G3CMS1, G3CMS0): Thes
select the compare match that triggers pulse output group 3 (pins PO_{15} to PO_{12}).

Bit 7	Bit 6	Description
G3CMS1	**G3CMS0**	**Output Trigger for Pulse Output Group 3**
0	0	Compare match in TPU channel 0
	1	Compare match in TPU channel 1
1	0	Compare match in TPU channel 2
	1	Compare match in TPU channel 3 (I

Bits 5 and 4—Group 2 Compare Match Select 1 and 0 (G2CMS1, G2CMS0): Thes
select the compare match that triggers pulse output group 2 (pins PO_{11} to PO_{8}).

Bit 5	Bit 4	Description
G2CMS1	**G2CMS0**	**Output Trigger for Pulse Output Group 2**
0	0	Compare match in TPU channel 0
	1	Compare match in TPU channel 1
1	0	Compare match in TPU channel 2
	1	Compare match in TPU channel 3 (I

	1	Compare match in TPU channel 3

Bits 1 and 0—Group 0 Compare Match Select 1 and 0 (G0CMS1, G0CMS0): The select the compare match that triggers pulse output group 0 (pins PO_3 to PO_0).

Bit 1	Bit 0	Description
G0CMS1	**G0CMS0**	**Output Trigger for Pulse Output Group 0**
0	0	Compare match in TPU channel 0
	1	Compare match in TPU channel 1
1	0	Compare match in TPU channel 2
	1	Compare match in TPU channel 3

11.2.6 PPG Output Mode Register (PMR)

Bit	:	7	6	5	4	3	2	1
		G3INV	G2INV	G1INV	G0INV	G3NOV	G2NOV	G1NOV
Initial value	:	1	1	1	1	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PMR is an 8-bit readable/writable register that selects pulse output inversion and non-operation for each group.

The output trigger period of a non-overlapping operation PPG output waveform is set and the non-overlap margin is set in TGRA. The output values change at compare mat

For details, see section 11.3.4, Non-Overlapping Pulse Output.

PMR is initialized to H'F0 by a reset and in hardware standby mode. It is not initialize software standby mode.

Bit 6—Group 2 Inversion (G2INV): Selects direct output or inverted output for pulse group 2 (pins PO_{11} to PO_8).

Bit 6

G2INV	Description
0	Inverted output for pulse output group 2 (low-level output at pin for a 1 in P
1	Direct output for pulse output group 2 (high-level output at pin for a 1 in PO (I

Bit 5—Group 1 Inversion (G1INV): Selects direct output or inverted output for pulse group 1 (pins PO_7 to PO_4).

Bit 5

G1INV	Description
0	Inverted output for pulse output group 1 (low-level output at pin for a 1 in P
1	Direct output for pulse output group 1 (high-level output at pin for a 1 in PO (I

Bit 4—Group 0 Inversion (G0INV): Selects direct output or inverted output for pulse group 0 (pins PO_3 to PO_0).

Bit 4

G0INV	Description
0	Inverted output for pulse output group 0 (low-level output at pin for a 1 in P
1	Direct output for pulse output group 0 (high-level output at pin for a 1 in PO (I

Bit 3—Group 3 Non-Overlap (G3NOV): Selects normal or non-overlapping operatio output group 3 (pins PO_{15} to PO_{12}).

output group 2 (pins PO_{11} to PO_8).

Bit 2

G2NOV	Description
0	Normal operation in pulse output group 2 (output values updated at compa in the selected TPU channel)
1	Non-overlapping operation in pulse output group 2 (independent 1 and 0 c compare match A or B in the selected TPU channel)

Bit 1—Group 1 Non-Overlap (G1NOV): Selects normal or non-overlapping operati output group 1 (pins PO_7 to PO_4).

Bit 1

G1NOV	Description
0	Normal operation in pulse output group 1 (output values updated at compa in the selected TPU channel)
1	Non-overlapping operation in pulse output group 1 (independent 1 and 0 c compare match A or B in the selected TPU channel)

Bit 0—Group 0 Non-Overlap (G0NOV): Selects normal or non-overlapping operati output group 0 (pins PO_3 to PO_0).

Bit 0

G0NOV	Description
0	Normal operation in pulse output group 0 (output values updated at compa in the selected TPU channel)
1	Non-overlapping operation in pulse output group 0 (independent 1 and 0 c compare match A or B in the selected TPU channel)

pins of port 1.

Port 1 is multiplexed with pins PO_{15} to PO_8. Bits corresponding to pins used for PPG ou
be set to 1. For further information about P1DDR, see section 9.2, Port 1.

11.2.8 Port 2 Data Direction Register (P2DDR)

Bit	:	7	6	5	4	3	2	1
		P27DDR	P26DDR	P25DDR	P24DDR	P23DDR	P22DDR	P21DDR
Initial value	:	0	0	0	0	0	0	0
R/W	:	W	W	W	W	W	W	W

P2DDR is an 8-bit write-only register, the individual bits of which specify input or outp
pins of port 2.

Port 2 is multiplexed with pins PO_7 to PO_0. Bits corresponding to pins used for PPG ou
be set to 1. For further information about P2DDR, see section 9.3, Port 2.

MSTPCR is a 16-bit readable/writable register that performs module stop mode contro

When the MSTP11 bit in MSTPCR is set to 1, PPG operation stops at the end of the b a transition is made to module stop mode. Registers cannot be read or written to in mo mode. For details, see section 21.5, Module Stop Mode.

MSTPCR is initialized to H'3FFF by a reset and in hardware standby mode. It is not in software standby mode.

Bit 11—Module Stop (MSTP11): Specifies the PPG module stop mode.

Bit 11

MSTP11	Description
0	PPG module stop mode cleared
1	PPG module stop mode set

transferred to PODR to update the output values.

Figure 11.2 illustrates the PPG output operation and table 11.3 summarizes the PPG op
conditions.

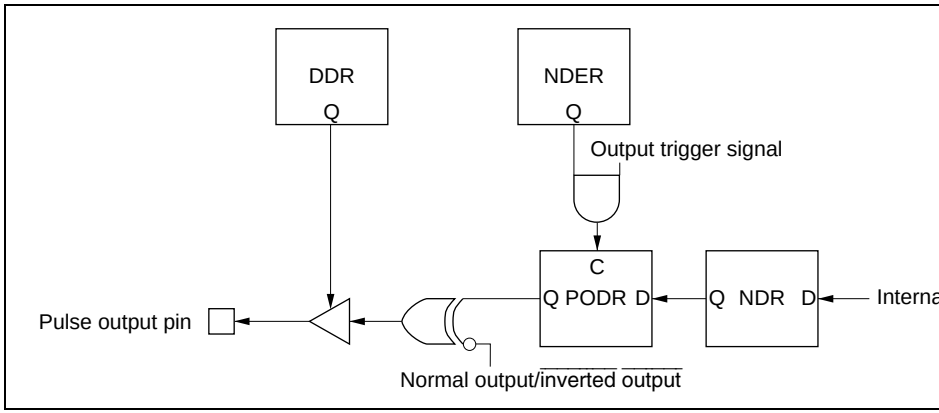


Figure 11.2 PPG Output Operation

Table 11.3 PPG Operating Conditions

NDER	DDR	Pin Function
0	0	Generic input port
	1	Generic output port
1	0	Generic input port (but the PODR bit is a read-only bit, and wh compare match occurs, the NDR bit value is transferred to the
	1	PPG pulse output

Sequential output of data of up to 16 bits is possible by writing new output data to ND the next compare match. For details of non-overlapping operation, see section 11.3.4, N Overlapping Pulse Output.

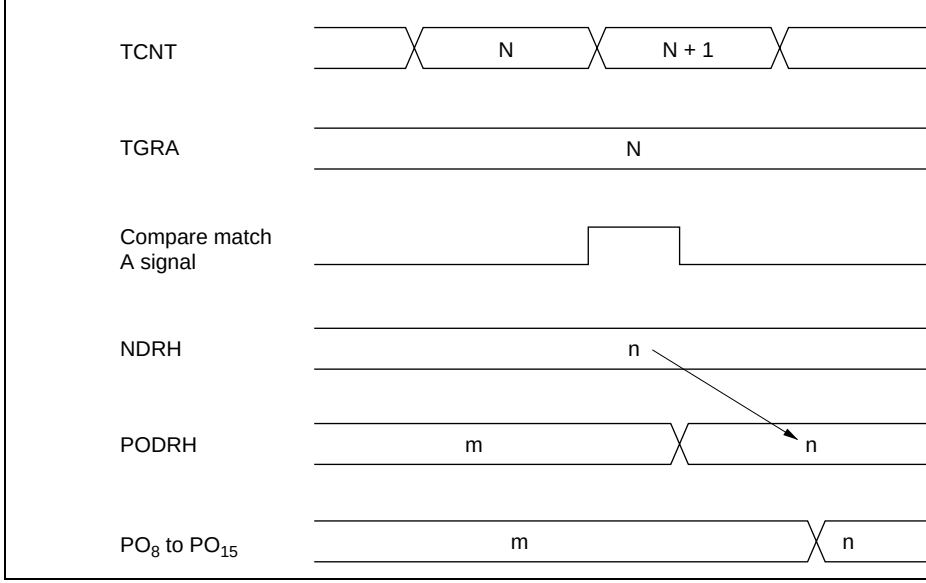
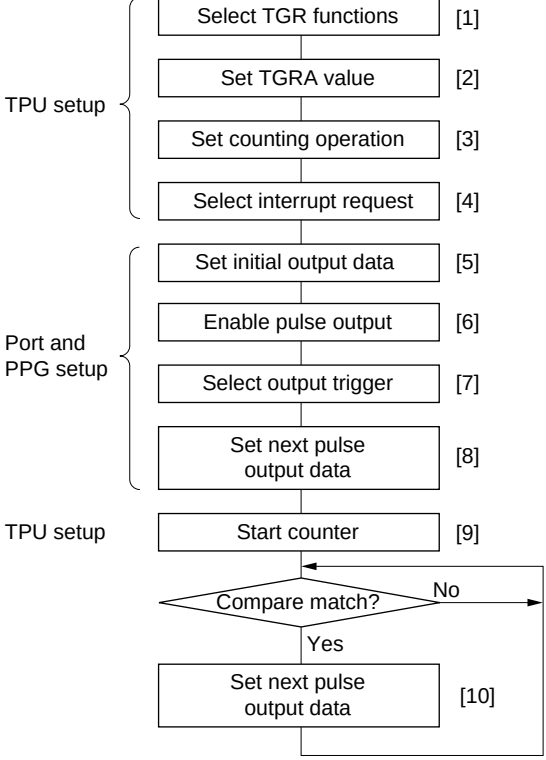


Figure 11.3 Timing of Transfer and Output of NDR Contents (Examp



disabled)

[2] Set the PPG output trigger p

[3] Select the counter clock sou with bits TPSC2 to TPSC0 i Select the counter clear sou with bits CCLR1 and CCLR

[4] Enable the TGIA interrupt in The DTC or DMAC can also up to transfer data to NDR.

[5] Set the initial output values PODR.

[6] Set the DDR and NDER bits pins to be used for pulse out

[7] Select the TPU compare ma event to be used as the outp trigger in PCR.

[8] Set the next pulse output va NDR.

[9] Set the CST bit in TSTR to start the TCNT counter.

[10] At each TGIA interrupt, set output values in NDR.

Figure 11.4 Setup Procedure for Normal Pulse Output (Example)

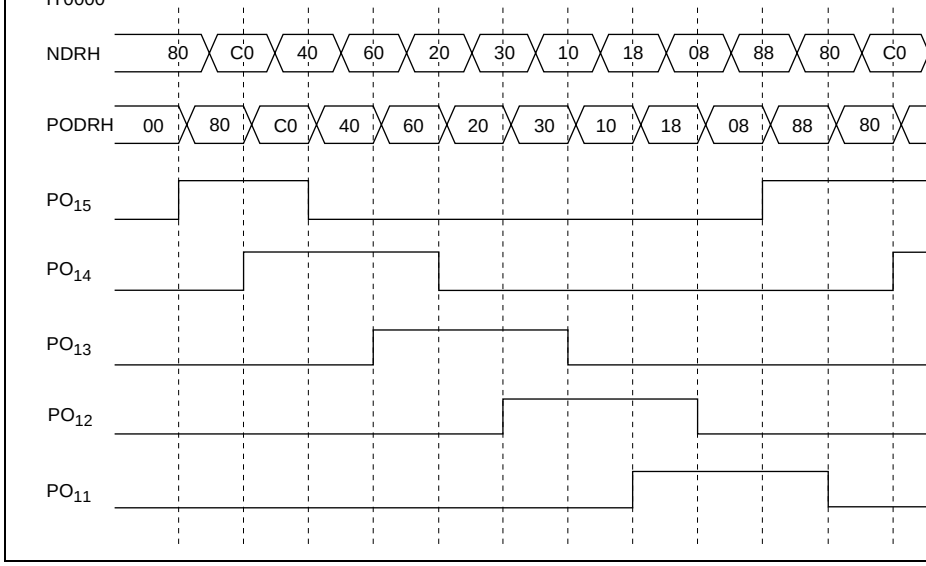


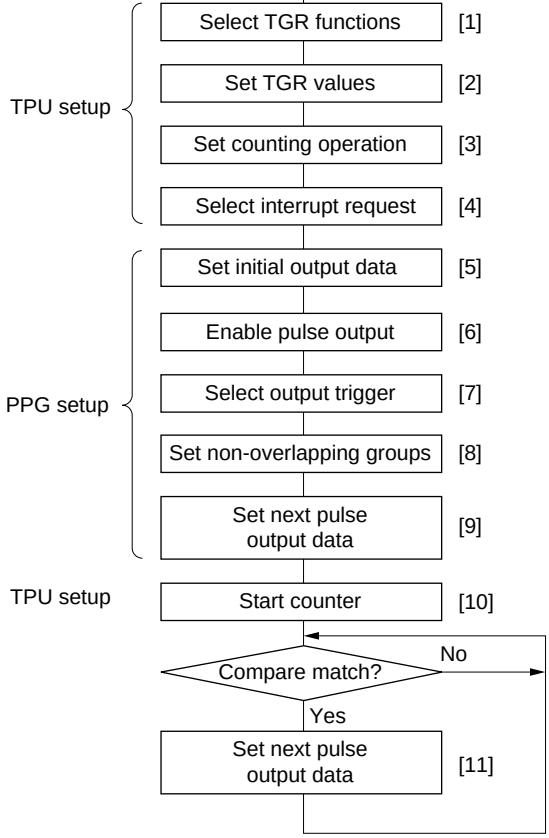
Figure 11.5 Normal Pulse Output Example (Five-Phase Pulse Output

[1] Set up the TPU channel to be used as the output trigger channel so that TGRA is a compare register and the counter will be cleared by compare match A. Set the trigg TGRA and set the TGIEA bit in TIER to 1 to enable the compare match A (TGIA)

[2] Write H'F8 in P1DDR and NDERH, and set the G3CMS1, G3CMS0, G2CMS1, a bits in PCR to select compare match in the TPU channel set up in the previous ste output trigger. Write output data H'80 in NDRH.

[3] The timer counter in the TPU channel starts. When compare match A occurs, the N contents are transferred to PODRH and output. The TGIA interrupt handling routi next output data (H'C0) in NDRH.

[4] Five-phase overlapping pulse output (one or two phases active at a time) can be ob subsequently by writing H'40, H'60, H'20, H'30. H'10, H'18, H'08, H'88... at succes interrupts. If the DTC or DMAC is set for activation by this interrupt, pulse output obtained without imposing a load on the CPU.



(with output disabled)

[2] Set the pulse output trigger in TGRB and the non-overla margin in TGRA.

[3] Select the counter clock sou with bits TPSC2 to TPSC0 i Select the counter clear sou with bits CCLR1 and CCLR

[4] Enable the TGIA interrupt in The DTC or DMAC can also up to transfer data to NDR.

[5] Set the initial output values PODR.

[6] Set the DDR and NDER bits pins to be used for pulse ou

[7] Select the TPU compare ma event to be used as the puls output trigger in PCR.

[8] In PMR, select the groups th operate in non-overlap mod

[9] Set the next pulse output va NDR.

[10] Set the CST bit in TSTR to start the TCNT counter.

[11] At each TGIA interrupt, set output values in NDR.

Figure 11.6 Setup Procedure for Non-Overlapping Pulse Output (Examp

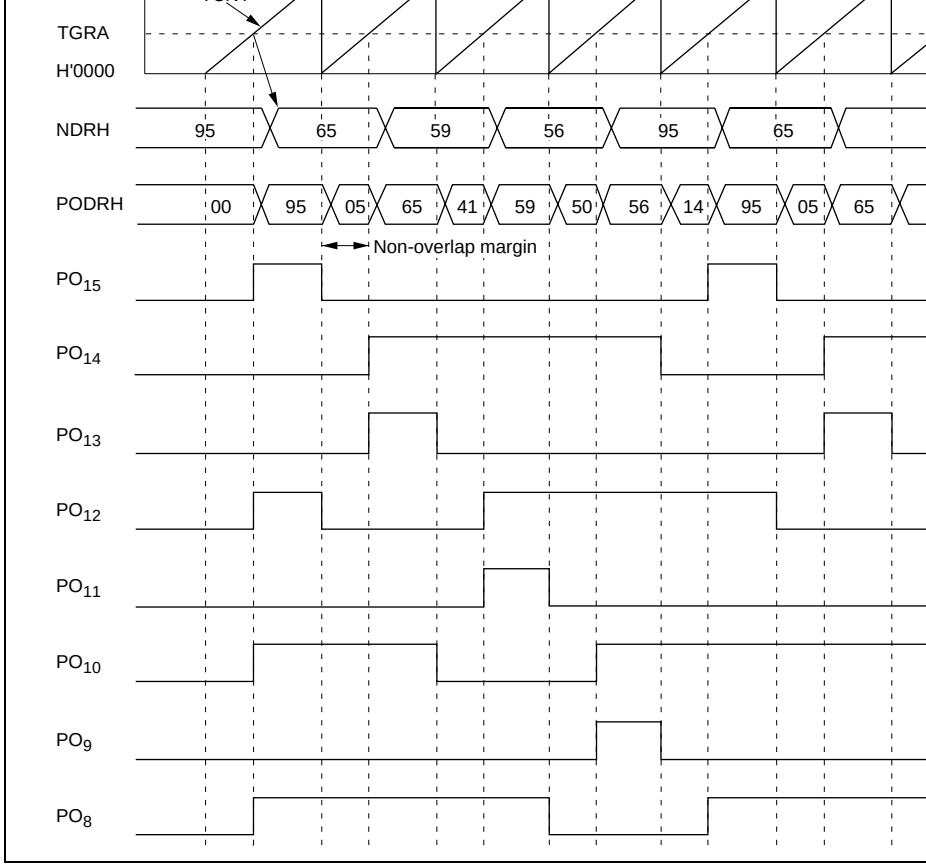


Figure 11.7 Non-Overlapping Pulse Output Example (Four-Phase Complen

Write output data H'55 in NDRH.

[3] The timer counter in the TPU channel starts. When a compare match with TGRB oc
outputs change from 1 to 0. When a compare match with TGRA occurs, outputs cha
to 1 (the change from 0 to 1 is delayed by the value set in TGRA). The TGIA interr
handling routine writes the next output data (H'65) in NDRH.

[4] Four-phase complementary non-overlapping pulse output can be obtained subseque
writing H'59, H'56, H'95... at successive TGIA interrupts. If the DTC or DMAC is s
activation by this interrupt, pulse output can be obtained without imposing a load or

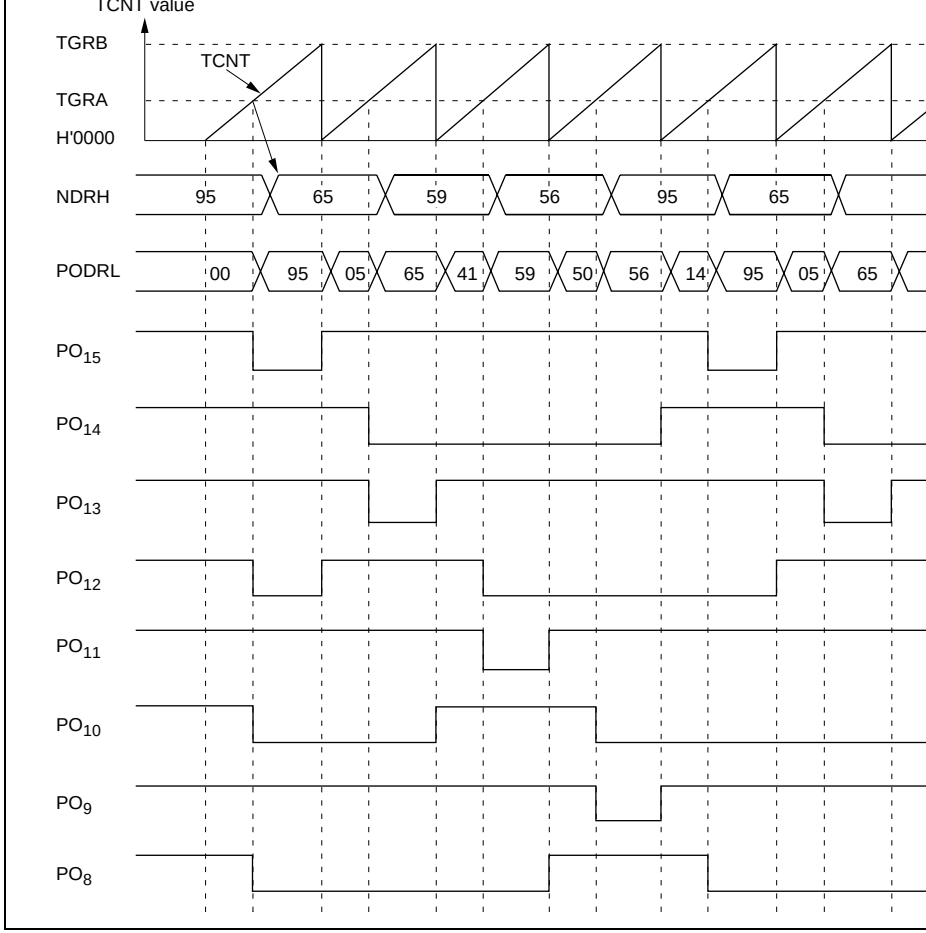


Figure 11.8 Inverted Pulse Output (Example)

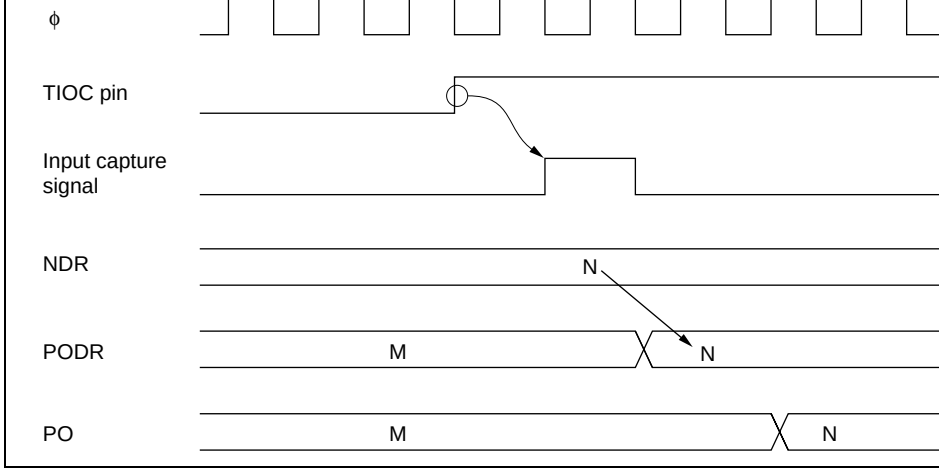


Figure 11.9 Pulse Output Triggered by Input Capture (Example)

Pin functions should be changed only under conditions in which the output trigger eve
occur.

11.4.2 Note on Non-Overlapping Output

During non-overlapping operation, the transfer of NDR bit values to PODR bits takes
follows.

- NDR bits are always transferred to PODR bits at compare match A.
- At compare match B, NDR bits are transferred only if their value is 0. Bits are not
  if their value is 1.

Figure 11.10 illustrates the non-overlapping pulse output operation.

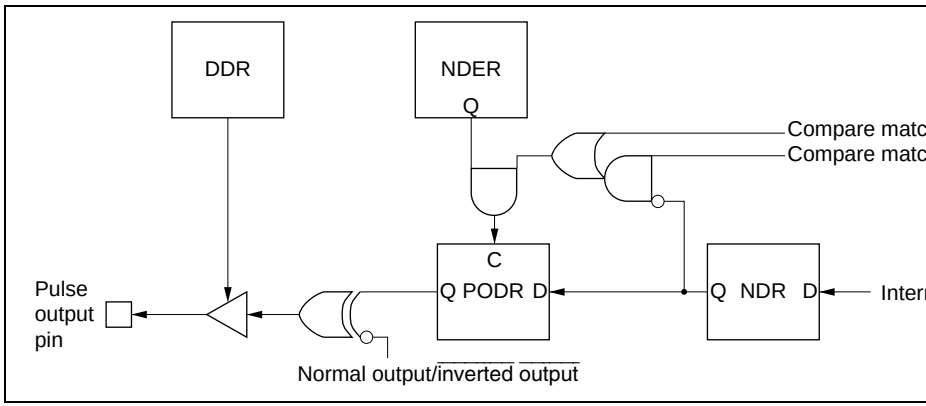


Figure 11.10 Non-Overlapping Pulse Output

Therefore, 0 data can be transferred ahead of 1 data by making compare match B occu
compare match A. The NDR contents should not be altered during the interval from c
match B to compare match A (the non-overlap margin).

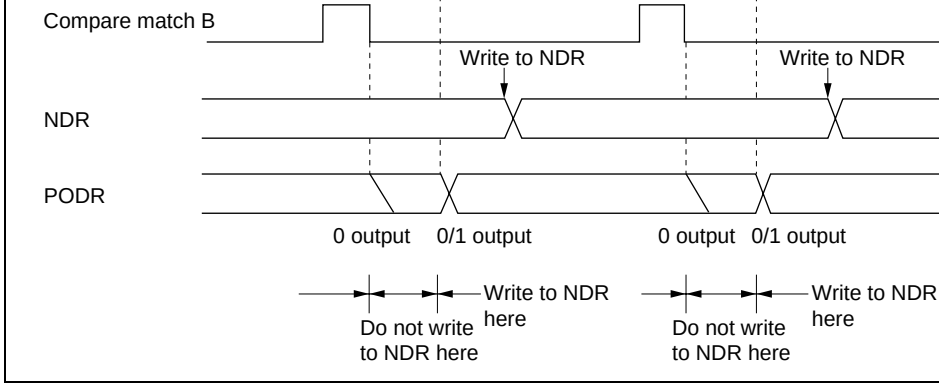


Figure 11.11 Non-Overlapping Operation and NDR Write Timing

module can thus be used for a variety of functions, including pulse output with an arbi
cycle.

12.1.1 Features

The features of the 8-bit timer module are listed below.

- Selection of four clock sources

  The counters can be driven by one of three internal clock signals (ϕ/8, ϕ/64, or ϕ/8
  external clock input (enabling use as an external event counter).
- Selection of three ways to clear the counters

  The counters can be cleared on compare match A or B, or by an external reset sign
- Timer output control by a combination of two compare match signals

  The timer output signal in each channel is controlled by a combination of two inde
  compare match signals, enabling the timer to generate output waveforms with an a
  cycle or PWM output.
- Provision for cascading of two channels
  - Operation as a 16-bit timer is possible, using channel 0 for the upper 8 bits and
    for the lower 8 bits (16-bit count mode).
  - Channel 1 can be used to count channel 0 compare matches (compare match co
- Three independent interrupts

  Compare match A and B and overflow interrupts can be requested independently.

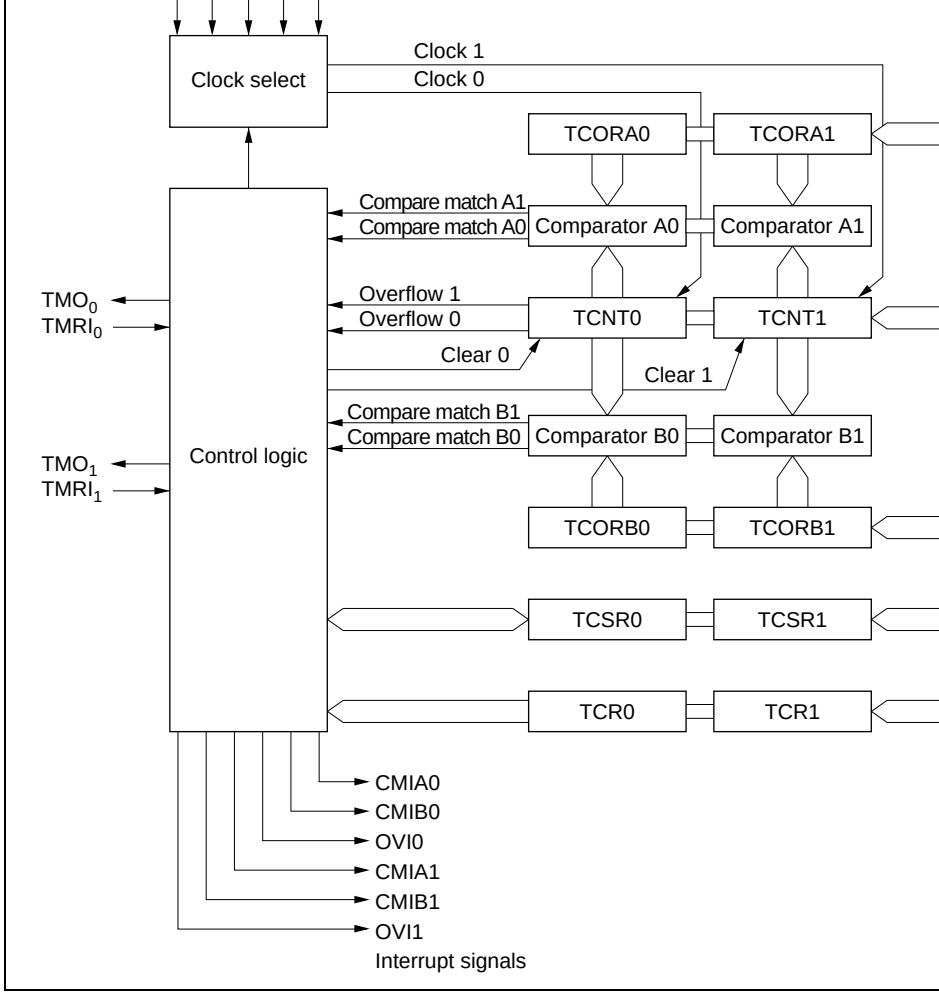


Figure 12.1 Block Diagram of 8-Bit Timer

	Timer reset input pin 0	$TMRI_0$	Input	Inputs external reset to c
1	Timer output pin 1	TMO_1	Output	Outputs at compare mat
	Timer clock input pin 1	$TMCI_1$	Input	Inputs external clock for
	Timer reset input pin 1	$TMRI_1$	Input	Inputs external reset to c

12.1.4 Register Configuration

Table 12.2 summarizes the registers of the 8-bit timer module.

Table 12.2 8-Bit Timer Registers

Channel	Name	Abbreviation	R/W	Initial value	A
0	Timer control register 0	TCR0	R/W	H'00	H
	Timer control/status register 0	TCSR0	R/(W)*2	H'00	H
	Time constant register A0	TCORA0	R/W	H'FF	H
	Time constant register B0	TCORB0	R/W	H'FF	H
	Timer counter 0	TCNT0	R/W	H'00	H
1	Timer control register 1	TCR1	R/W	H'00	H
	Timer control/status register 1	TCSR1	R/(W)*2	H'10	H
	Time constant register A1	TCORA1	R/W	H'FF	H
	Time constant register B1	TCORB1	R/W	H'FF	H
	Timer counter 1	TCNT1	R/W	H'00	H
All	Module stop control register	MSTPCR	R/W	H'3FFF	H

Notes: 1. Lower 16 bits of the address
2. Only 0 can be written to bits 7 to 5, to clear these flags.

Each pair of registers for channel 0 and channel 1 is a 16-bit register with the upper 8 channel 0 and the lower 8 bits for channel 1, so they can be accessed together by word instruction.

Initial value	:	0	0	0	0	0	0	0	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	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TCNT0 and TCNT1 are 8-bit readable/writable up-counters that increment on pulses ge
from an internal or external clock source. This clock source is selected by clock select b
to CKS0 of TCR. The CPU can read or write to TCNT0 and TCNT1 at all times.

TCNT0 and TCNT1 comprise a single 16-bit register, so they can be accessed together
transfer instruction.

TCNT0 and TCNT1 can be cleared by an external reset input or by a compare match si
Which signal is to be used for clearing is selected by clock clear bits CCLR1 and CCLE

When a timer counter overflows from H'FF to H'00, OVF in TCSR is set to 1.

TCNT0 and TCNT1 are each initialized to H'00 by a reset and in hardware standby mo

12.2.2 Time Constant Registers A0 and A1 (TCORA0, TCORA1)

		TCORA0								TCORA1						
Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
Initial value	:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TCORA0 and TCORA1 are 8-bit readable/writable registers. TCORA0 and TCORA1 c
single 16-bit register so they can be accessed together by word transfer instruction.

TCORA is continually compared with the value in TCNT. When a match is detected, th
corresponding CMFA flag of TCSR is set. Note, however, that comparison is disabled
T2 state of a TCOR write cycle.

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
Initial value	:	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TCORB0 and TCORB1 are 8-bit readable/writable registers. TCORB0 and TCORB1 single 16-bit register so they can be accessed together by word transfer instruction.

TCORB is continually compared with the value in TCNT. When a match is detected, corresponding CMFB flag of TCSR is set. Note, however, that comparison is disabled T2 state of a TCOR write cycle.

The timer output can be freely controlled by these compare match signals and the setti output select bits OS3 and OS2 of TCSR.

TCORB0 and TCORB1 are each initialized to H'FF by a reset and in hardware standb

12.2.4 Time Control Registers 0 and 1 (TCR0, TCR1)

Bit	:	7	6	5	4	3	2	1
		CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	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

TCR0 and TCR1 are 8-bit readable/writable registers that select the clock source and t which TCNT is cleared, and enable interrupts.

TCR0 and TCR1 are each initialized to H'00 by a reset and in hardware standby mode

For details of this timing, see section 12.3, Operation.

Bit 6—Compare Match Interrupt Enable A (CMIEA): Selects whether CMFA inter
requests (CMIA) are enabled or disabled when the CMFA flag of TCSR is set to 1.

Bit 6	
CMIEA	**Description**
0	CMFA interrupt requests (CMIA) are disabled
1	CMFA interrupt requests (CMIA) are enabled

Bit 5—Timer Overflow Interrupt Enable (OVIE): Selects whether OVF interrupt re
(OVI) are enabled or disabled when the OVF flag of TCSR is set to 1.

Bit 5	
OVIE	**Description**
0	OVF interrupt requests (OVI) are disabled
1	OVF interrupt requests (OVI) are enabled

Bits 4 and 3—Counter Clear 1 and 0 (CCLR1 and CCLR0): These bits select the m
which TCNT is cleared: by compare match A or B, or by an external reset input.

Bit 4	Bit 3	
CCLR1	**CCLR0**	**Description**
0	0	Clear is disabled
	1	Clear by compare match A
1	0	Clear by compare match B
	1	Clear by rising edge of external reset input

Some functions differ between channel 0 and channel 1.

Bit 2	Bit 1	Bit 0	
CKS2	**CKS1**	**CKS0**	**Description**
0	0	0	Clock input disabled
		1	Internal clock, counted at falling edge of ϕ/8
	1	0	Internal clock, counted at falling edge of ϕ/64
		1	Internal clock, counted at falling edge of ϕ/8192
1	0	0	For channel 0: count at TCNT1 overflow signal*
			For channel 1: count at TCNT0 compare match A*
		1	External clock, counted at rising edge
	1	0	External clock, counted at falling edge
		1	External clock, counted at both rising and falling edges

Note: * If the count input of channel 0 is the TCNT1 overflow signal and that of cha
TCNT0 compare match signal, no incrementing clock is generated. Do not
setting.

TCSR1

Bit	:	7	6	5	4	3	2	1	
		CMFB	CMFA	OVF	—	OS3	OS2	OS1	O
Initial value	:	0	0	0	1	0	0	0	
R/W	:	R/(W)*	R/(W)*	R/(W)*	—	R/W	R/W	R/W	R

Note: * Only 0 can be written to bits 7 to 5, to clear these flags.

TCSR0 and TCSR1 are 8-bit registers that display compare match and overflow statuse control compare match output.

TCSR0 is initialized to H'00, and TCSR1 to H'10, by a reset and in hardware standby m

Bit 7—Compare Match Flag B (CMFB): Status flag indicating whether the values of TCORB match.

Bit 7

CMFB	Description
0	[Clearing conditions] (• Cleared by reading CMFB when CMFB = 1, then writing 0 to CMFB • When DTC is activated by CMIB interrupt while DISEL bit of MRB in DT
1	[Setting condition] Set when TCNT matches TCORB

1	[Setting condition] Set when TCNT matches TCORA

Bit 5—Timer Overflow Flag (OVF): Status flag indicating that TCNT has overflow
from H'FF to H'00).

Bit 5

OVF	Description
0	[Clearing condition] Cleared by reading OVF when OVF = 1, then writing 0 to OVF
1	[Setting condition] Set when TCNT overflows from H'FF to H'00

Bit 4—A/D Trigger Enable (ADTE) (TCSR0 Only): Selects enabling or disabling o
converter start requests by compare-match A.

In TCSR1, this bit is reserved: it is always read as 1 and cannot be modified.

Bit 4

ADTE	Description
0	A/D converter start requests by compare match A are disabled
1	A/D converter start requests by compare match A are enabled

priority.

Timer output is disabled when bits OS3 to OS0 are all 0.

After a reset, the timer output is 0 until the first compare match event occurs.

Bit 3	Bit 2	
OS3	**OS2**	**Description**
0	0	No change when compare match B occurs
	1	0 is output when compare match B occurs
1	0	1 is output when compare match B occurs
	1	Output is inverted when compare match B occurs (toggle output)

Bit 1	Bit 0	
OS1	**OS0**	**Description**
0	0	No change when compare match A occurs
	1	0 is output when compare match A occurs
1	0	1 is output when compare match A occurs
	1	Output is inverted when compare match A occurs (toggle output)

MSTPCR is a 16-bit readable/writable register that performs module stop mode contro

When the MSTP12 bit in MSTPCR is set to 1, the 8-bit timer operation stops at the en cycle and a transition is made to module stop mode. Registers cannot be read or writte module stop mode. For details, see section 21.5, Module Stop Mode.

MSTPCR is initialized to H'3FFF by a reset and in hardware standby mode. It is not in software standby mode.

Bit 12—Module Stop (MSTP12): Specifies the 8-bit timer stop mode.

Bit 12

MSTP12	Description
0	8-bit timer module stop mode cleared
1	8-bit timer module stop mode set

Three different internal clock signals (φ/8, φ/64, or φ/8192) divided from the system clo
be selected, by setting bits CKS2 to CKS0 in TCR. Figure 12.2 shows the count timing

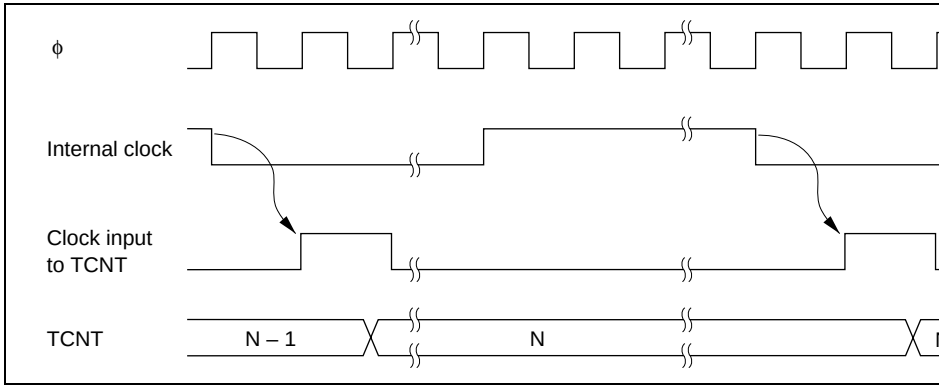


Figure 12.2 Count Timing for Internal Clock Input

External Clock

Three incrementation methods can be selected by setting bits CKS2 to CKS0 in TCR: a
edge, the falling edge, and both rising and falling edges.

Note that the external clock pulse width must be at least 1.5 states for incrementation at
edge, and at least 2.5 states for incrementation at both edges. The counter will not incr
correctly if the pulse width is less than these values.

Figure 12.3 shows the timing of incrementation at both edges of an external clock signa

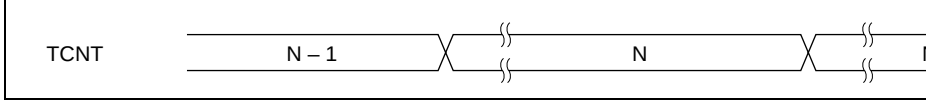


Figure 12.3 Count Timing for External Clock Input

12.3.2 Compare Match Timing

Setting of Compare Match Flags A and B (CMFA, CMFB)

The CMFA and CMFB flags in TCSR are set to 1 by a compare match signal generate
TCOR and TCNT values match. The compare match signal is generated at the last sta
the match is true, just before the timer counter is updated.

Therefore, when TCOR and TCNT match, the compare match signal is not generated
next incrementation clock input. Figure 12.4 shows this timing.

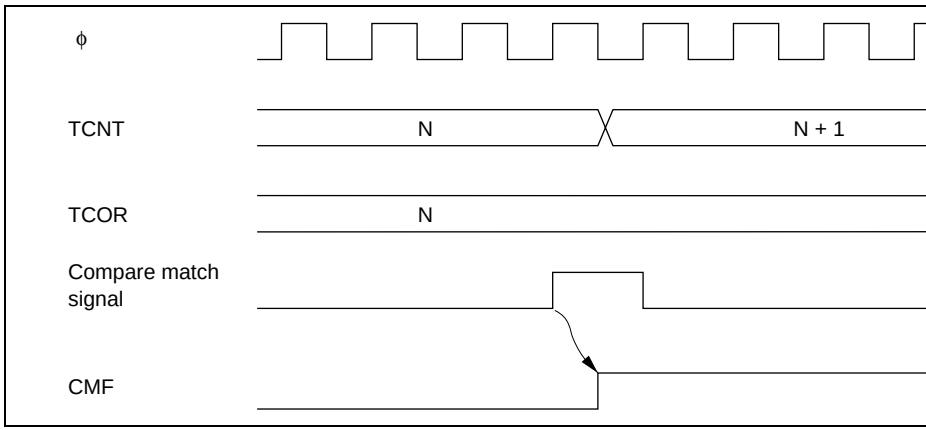


Figure 12.4 Timing of CMF Setting

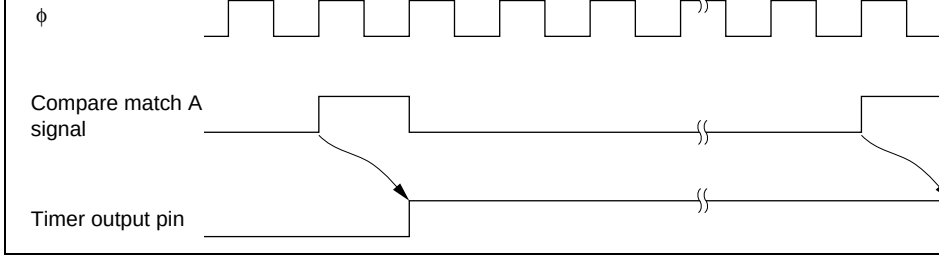


Figure 12.5 Timing of Timer Output

Timing of Compare Match Clear

The timer counter is cleared when compare match A or B occurs, depending on the sett
CCLR1 and CCLR0 bits in TCR. Figure 12.6 shows the timing of this operation.

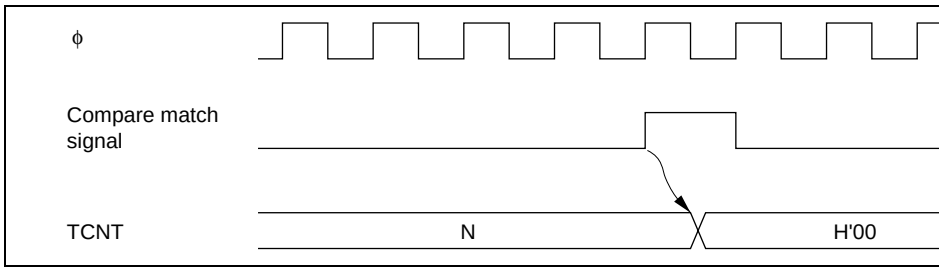


Figure 12.6 Timing of Compare Match Clear

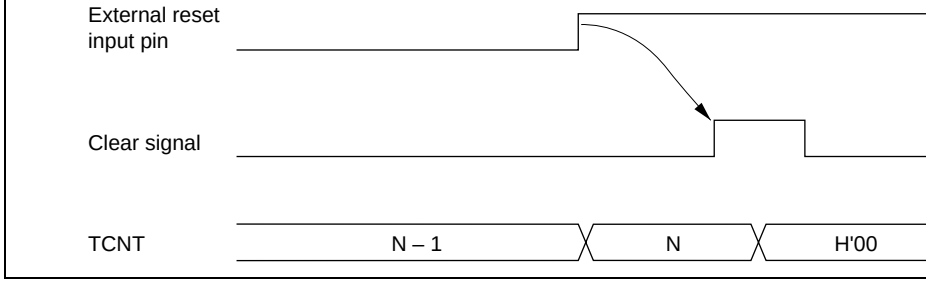


Figure 12.7 Timing of External Reset

12.3.4 Timing of Overflow Flag (OVF) Setting

The OVF in TCSR is set to 1 when the timer count overflows (changes from H'FF to H
12.8 shows the timing of this operation.

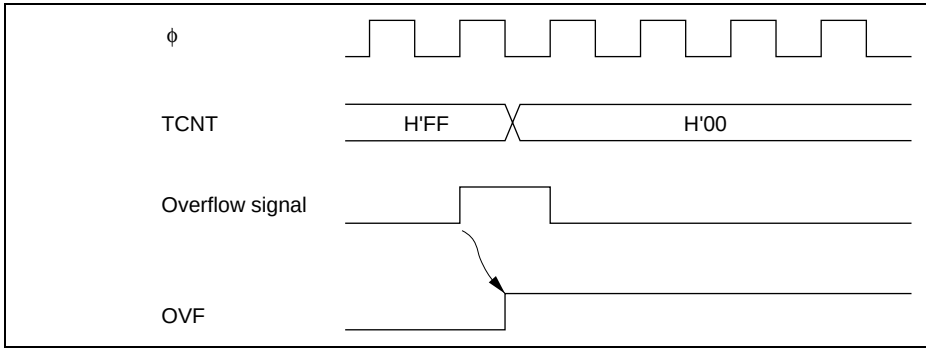


Figure 12.8 Timing of OVF Setting

When bits CKS2 to CKS0 in TCR0 are set to B'100, the timer functions as a single 16-
with channel 0 occupying the upper 8 bits and channel 1 occupying the lower 8 bits.

Setting of compare match flags:

- The CMF flag in TCSR0 is set to 1 when a 16-bit compare match event occurs.
- The CMF flag in TCSR1 is set to 1 when a lower 8-bit compare match event occurs

Counter clear specification:

- If the CCLR1 and CCLR0 bits in TCR0 have been set for counter clear at compare
  16-bit counter (TCNT0 and TCNT1 together) is cleared when a 16-bit compare mat
  occurs. The 16-bit counter (TCNT0 and TCNT1 together) is cleared even if counter
  the $TMRI_0$ pin has also been set.
- The settings of the CCLR1 and CCLR0 bits in TCR1 are ignored. The lower 8 bits
  cleared independently.

Pin output:

- Control of output from the TMO_0 pin by bits OS3 to OS0 in TCSR0 is in accordanc
  16-bit compare match conditions.
- Control of output from the TMO_1 pin by bits OS3 to OS0 in TCSR1 is in accordanc
  lower 8-bit compare match conditions.

Compare Match Counter Mode

When bits CKS2 to CKS0 in TCR1 are B'100, TCNT1 counts compare match A's for c

Channels 1 and 0 are controlled independently. Conditions such as setting of the CMF
generation of interrupts, output from the TMO pin, and counter clear are in accordance
settings for each channel.

There are three 8-bit timer interrupt sources: CMIA, CMIB, and OVI. Their relative p
shown in table 12.3. Each interrupt source is set as enabled or disabled by the corresp
interrupt enable bit in TCR, and independent interrupt requests are sent for each to the
controller. It is also possible to activate the DTC by means of CMIA and CMIB interr

Table 12.3 8-Bit Timer Interrupt Sources

Interrupt Source	Description	DTC Activation	Pr
CMIA	Interrupt by CMFA	Possible	Hi
CMIB	Interrupt by CMFB	Possible	
OVI	Interrupt by OVF	Not possible	Lo

compare match and to 0 at a TCORB compare match.

With these settings, the 8-bit timer provides output of pulses at a rate determined by TC
a pulse width determined by TCORB. No software intervention is required.

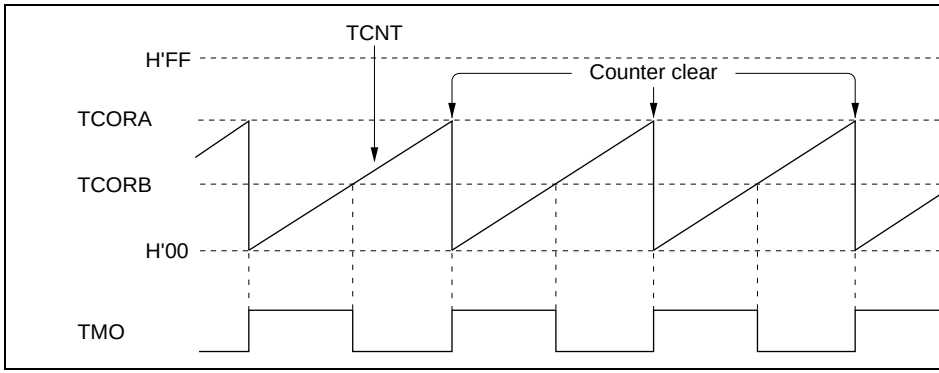


Figure 12.9 Example of Pulse Output

takes priority, so that the counter is cleared and the write is not performed.

Figure 12.10 shows this operation.

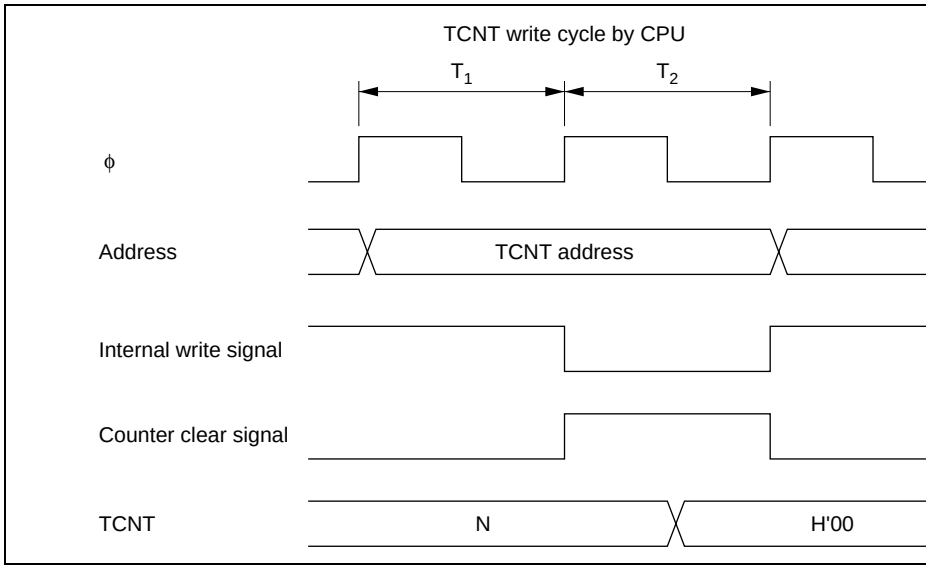


Figure 12.10 Contention between TCNT Write and Clear

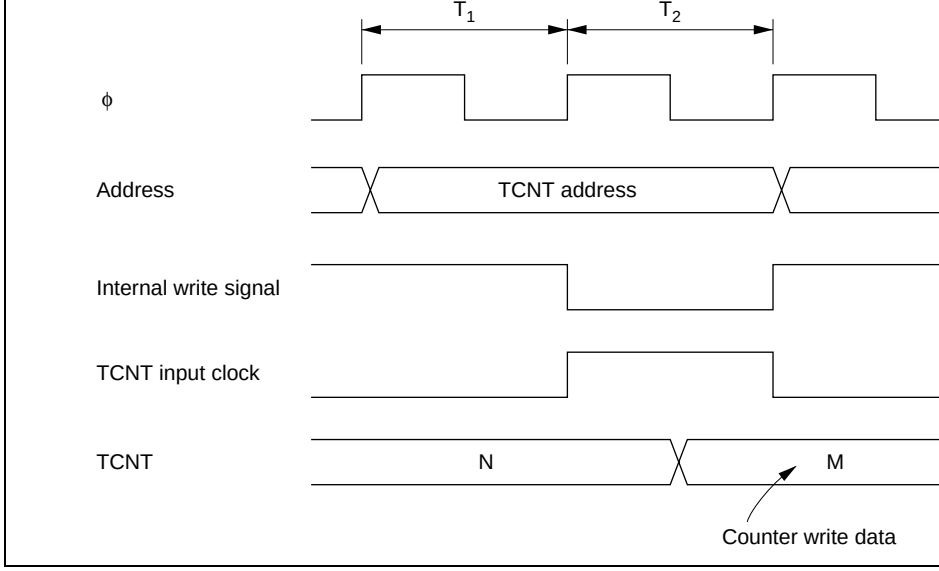


Figure 12.11 Contention between TCNT Write and Increment

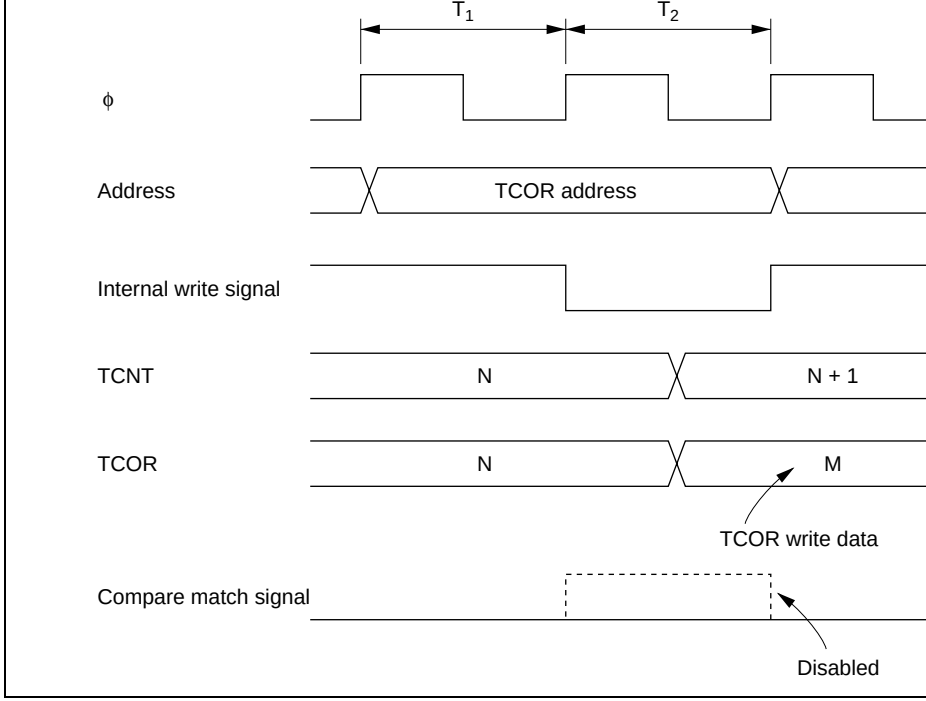


Figure 12.12 Contention between TCOR Write and Compare Match

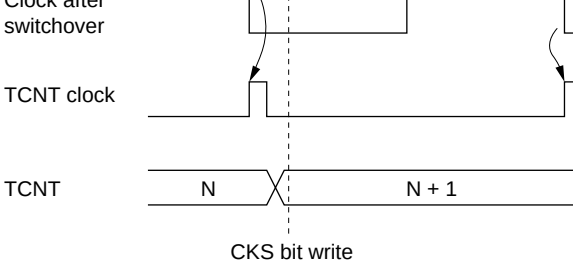
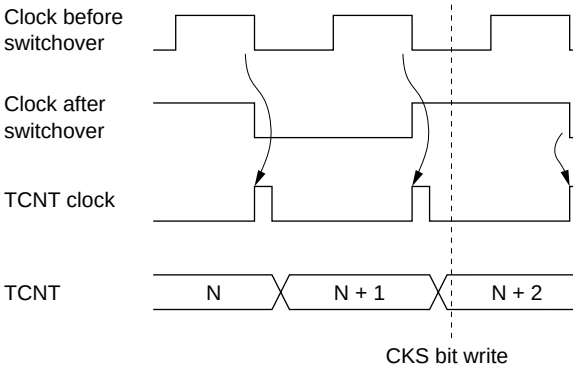
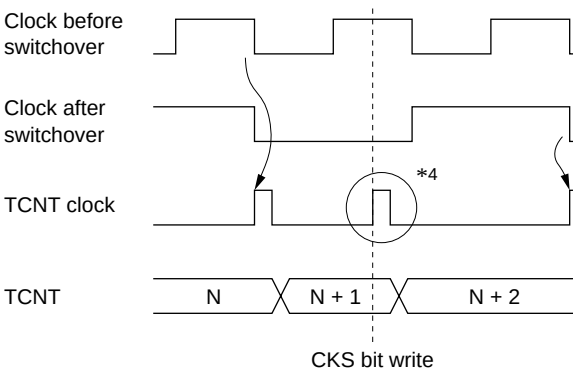
Toggle output	High
1 output	
0 output	
No change	Low

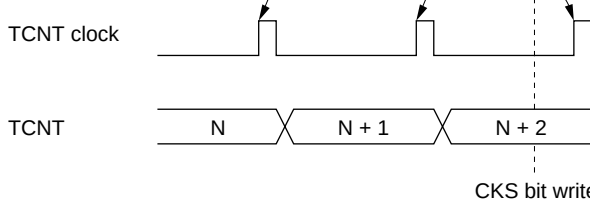
12.6.5 Switching of Internal Clocks and TCNT Operation

TCNT may increment erroneously when the internal clock is switched over. Table 12.5 relationship between the timing at which the internal clock is switched (by writing to th and CKS0 bits) and the TCNT operation.

When the TCNT clock is generated from an internal clock, the falling edge of the intern pulse is detected. If clock switching causes a change from high to low level, as shown i table 12.5, a TCNT clock pulse is generated on the assumption that the switchover is a edge. This increments TCNT.

The erroneous incrementation can also happen when switching between internal and ex clocks.

		
2	Switching from low to high*2	
3	Switching from high to low*3	



Notes: 1. Includes switching from low to stop, and from stop to low.
2. Includes switching from stop to high.
3. Includes switching from high to stop.
4. Generated on the assumption that the switchover is a falling edge; TCNT is incremented.

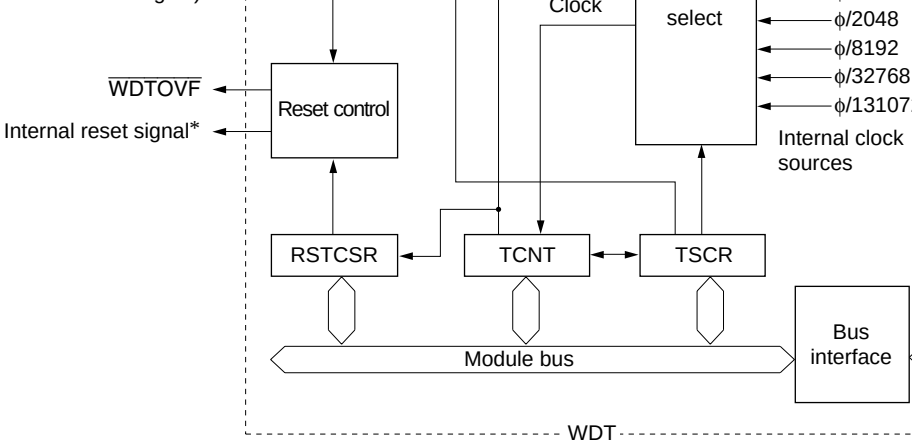
generate an internal reset signal for the H8S/2655 Group.

When this watchdog function is not needed, the WDT can be used as an interval timer
timer operation, an interval timer interrupt is generated each time the counter overflow

13.1.1 Features

WDT features are listed below.

- Switchable between watchdog timer mode and interval timer mode
- $\overline{\text{WDTOVF}}$ output when in watchdog timer mode
  If the counter overflows, the WDT outputs $\overline{\text{WDTOVF}}$. It is possible to select whet
  the entire H8S/2655 Group is reset at the same time. This internal reset can be a p
  reset or a manual reset.
- Interrupt generation when in interval timer mode
  If the counter overflows, the WDT generates an interval timer interrupt.
- Choice of eight counter clock sources.



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

Note: * The type of internal reset signal depends on a register setting. Either power-on reset o
reset can be selected.

Figure 13.1 Block Diagram of WDT

13.1.4 Register Configuration

The WDT has three registers, as summarized in table 13.2. These registers control clo
WDT mode switching, and the reset signal.

Table 13.2 WDT Registers

Name	Abbreviation	R/W	Initial Value	Ad Write*2
Timer control/status register	TCSR	R/(W)*3	H'18	H'FFBC
Timer counter	TCNT	R/W	H'00	H'FFBC
Reset control/status register	RSTCSR	R/(W)*3	H'1F	H'FFBE

Notes: 1. Lower 16 bits of the address.
2. For details of write operations, see section 13.2.4, Notes on Register Acces
3. Only a write of 0 is permitted to bit 7, to clear the flag.

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

TCNT is an 8-bit readable/writable* up-counter.

When the TME bit is set to 1 in TCSR, TCNT starts counting pulses generated from the clock source selected by bits CKS2 to CKS0 in TCSR. When the count overflows (chan H'FF to H'00), either the watchdog timer overflow signal ($\overline{\text{WDTOVF}}$) or an interval tim (WOVI) is generated, depending on the mode selected by the WT/$\overline{\text{IT}}$ bit in TCSR.

TCNT is initialized to H'00 by a reset, in hardware standby mode, or when the TME bit to 0. It is not initialized in software standby mode.

Note: * The method for writing to TCNT is different from that for general registers inadvertent overwriting. For details see section 13.2.4, Notes on Register Ac

13.2.2 Timer Control/Status Register (TCSR)

Bit	:	7	6	5	4	3	2	1
		OVF	WT/$\overline{\text{IT}}$	TME	—	—	CKS2	CKS1
Initial value	:	0	0	0	1	1	0	0
R/W	:	R/(W)*	R/W	R/W	—	—	R/W	R/W

Note: * Can only be written with 0 for flag clearing.

TCSR is an 8-bit readable/writable* register. Its functions include selecting the clock so input to TCNT, and the timer mode.

TCR is initialized to H'18 by a reset and in hardware standby mode. It is not initialized standby mode.

Note: * The method for writing to TCSR is different from that for general registers inadvertent overwriting. For details see section 13.2.4, Notes on Register Ac

Bit 6—Timer Mode Select (WT/$\overline{IT}$): Selects whether the WDT is used as a watchdo interval timer. If used as an interval timer, the WDT generates an interval timer interru (WOVI) when TCNT overflows. If used as a watchdog timer, the WDT generates the signal when TCNT overflows.

Bit 6

WT/$\overline{IT}$	Description
0	Interval timer: Sends the CPU an interval timer interrupt request (WOVI) when TCNT overflows
1	Watchdog timer: Generates the $\overline{WDTOVF}$ signal when TCNT overflows*

Note: * For details of the case where TCNT overflows in watchdog timer mode, see 13.2.3, Reset Control/Status Register (RSTCSR).

Bit 5—Timer Enable (TME): Selects whether TCNT runs or is halted.

Bit 5

TME	Description
0	TCNT is initialized to H'00 and halted
1	TCNT counts

Bits 4 and 3—Reserved: Read-only bits, always read as 1.

		1	φ/512	6.6 ms
1	0	0	φ/2048	26.2 ms
		1	φ/8192	104.9 ms
	1	0	φ/32768	419.4 ms
		1	φ/131072	1.68 s

Note: * The overflow period is the time from when TCNT starts counting up from H'0 overflow occurs.

13.2.3 Reset Control/Status Register (RSTCSR)

Bit	:	7	6	5	4	3	2	1
		WOVF	RSTE	RSTS	—	—	—	—
Initial value	:	0	0	0	1	1	1	1
R/W	:	R/(W)*	R/W	R/W	—	—	—	—

Note: * Can only be written with 0 for flag clearing.

RSTCSR is an 8-bit readable/writable* register that controls the generation of the inter signal when TCNT overflows, and selects the type of internal reset signal.

RSTCSR is initialized to H'1F by a reset signal from the $\overline{\text{RES}}$ pin, but not by the WDT reset signal caused by overflows.

Note: * The method for writing to RSTCSR is different from that for general registe prevent inadvertent overwriting. For details see section 13.2.4, Notes on Re Access.

	Set when TCNT overflows (changed from H'FF to H'00) during watchdog t operation

Bit 6—Reset Enable (RSTE): Specifies whether or not a reset signal is generated in t H8S/2655 Group if TCNT overflows during watchdog timer operation.

Bit 6

RSTE	Description
0	Reset signal is not generated if TCNT overflows*
1	Reset signal is generated if TCNT overflows

Note: * The modules within the H8S/2655 Group are not reset, but TCNT and TCS WDT are reset.

Bit 5—Reset Select (RSTS): Selects the type of internal reset generated if TCNT ove during watchdog timer operation.

For details of the types of resets, see section 4, Exception Handling.

Bit 5

RSTS	Description
0	Power-on reset
1	Manual reset

Bits 4 to 0—Reserved: Read-only bits, always read as 1.

These registers must be written to by a word transfer instruction. They cannot be writte
byte instructions.

Figure 13.2 shows the format of data written to TCNT and TCSR. TCNT and TCSR bo
same write address. For a write to TCNT, the upper byte of the written word must cont
and the lower byte must contain the write data. For a write to TCSR, the upper byte of
word must contain H'A5 and the lower byte must contain the write data. This transfers
data from the lower byte to TCNT or TCSR.

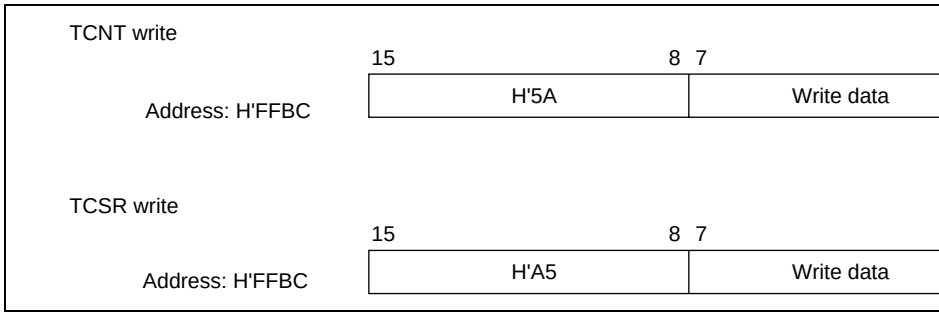


Figure 13.2 Format of Data Written to TCNT and TCSR

lower byte. This clears the WOVF bit to 0, but has no effect on the RSTE and RSTS b
to the RSTE and RSTS bits, the upper byte must contain H'5A and the lower byte mus
write data. This writes the values in bits 6 and 5 of the lower byte into the RSTE and
but has no effect on the WOVF bit.

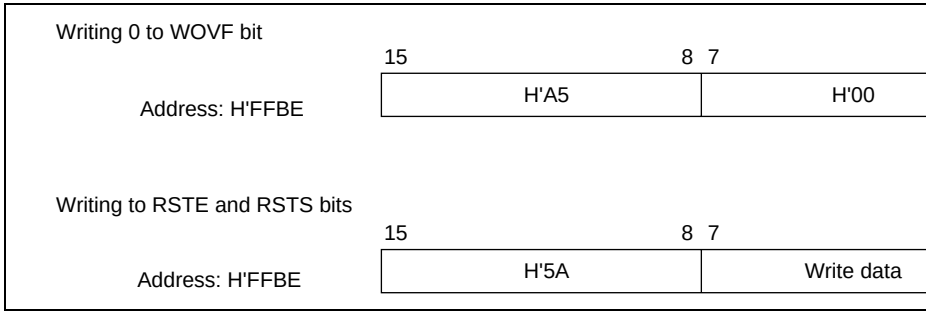


Figure 13.3 Format of Data Written to RSTCSR

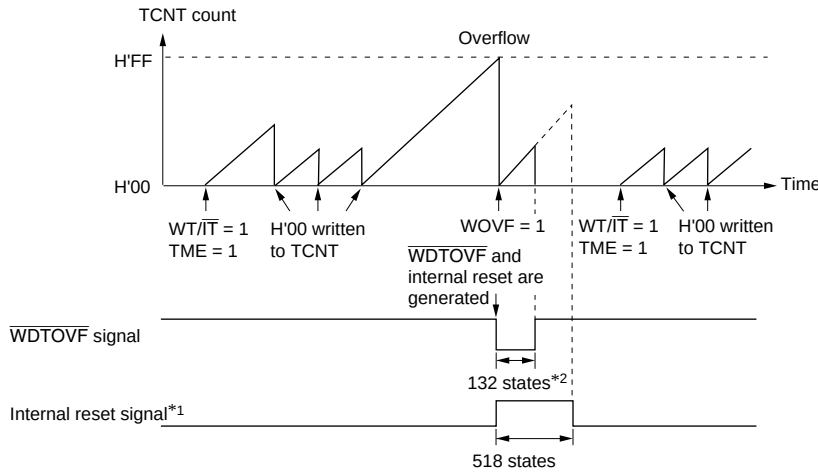
Reading TCNT, TCSR, and RSTCSR

These registers are read in the same way as other registers. The read addresses are H'F
TCSR, H'FFBD for TCNT, and H'FFBF for RSTCSR.

signal is output. This is shown in figure 13.4. This WDTOVF signal can be used to rese system. The $\overline{\text{WDTOVF}}$ signal is output for 132 states when RSTE = 1, and for 130 stat RSTE = 0.

If TCNT overflows when 1 is set in the RSTE bit in RSTCSR, a signal that resets the H Group internally is generated at the same time as the $\overline{\text{WDTOVF}}$ signal. This reset can b as a power-on reset or a manual reset, depending on the setting of the RSTS bit in RST internal reset signal is output for 518 states.

If a reset caused by a signal input to the $\overline{\text{RES}}$ pin occurs at the same time as a reset caus WDT overflow, the $\overline{\text{RES}}$ pin reset has priority and the WOVF bit in RSTCSR is cleared



Legend:
WT/$\overline{\text{IT}}$: Timer mode select bit
TME : Timer enable bit

Notes: 1. The internal reset signal is generated only if the RSTE bit is set to 1.
2. 130 states when the RSTE bit is cleared to 0.

Figure 13.4 Watchdog Timer Operation

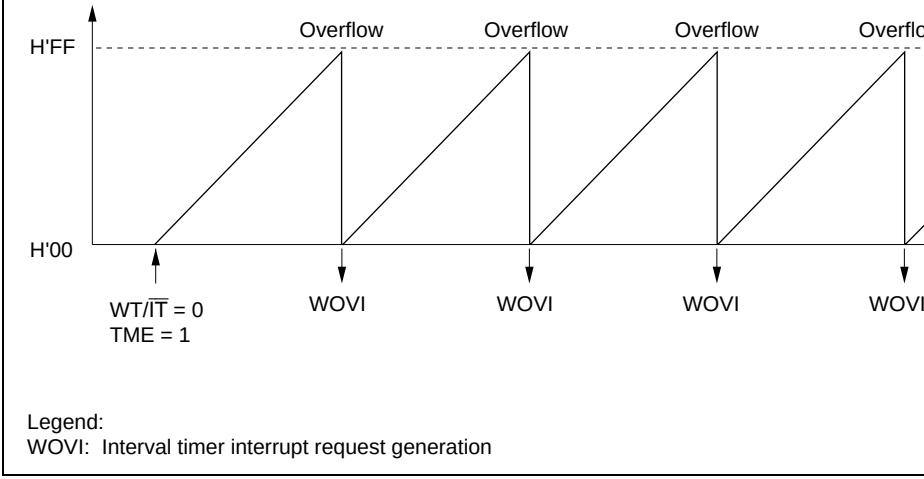


Figure 13.5 Interval Timer Operation

13.3.3 Timing of Setting Overflow Flag (OVF)

The OVF flag is set to 1 if TCNT overflows during interval timer operation. At the sa
interval timer interrupt (WOVI) is requested. This timing is shown in figure 13.6.

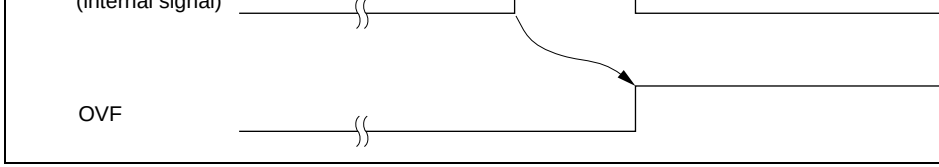


Figure 13.6 Timing of Setting of OVF

13.3.4 Timing of Setting of Watchdog Timer Overflow Flag (WOVF)

The WOVF flag is set to 1 if TCNT overflows during watchdog timer operation. At the
the $\overline{\text{WDTOVF}}$ signal goes low. If TCNT overflows while the RSTE bit in RSTCSR is s
internal reset signal is generated for the entire H8S/2655 Group chip. Figure 13.7 show
in this case.

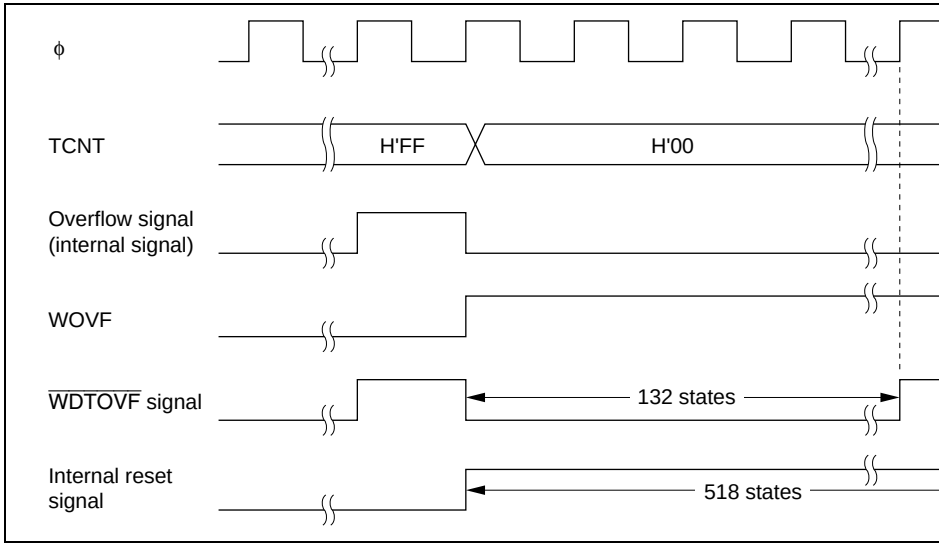


Figure 13.7 Timing of Setting of WOVF

If a timer counter clock pulse is generated during the T2 state of a TCNT write cycle, takes priority and the timer counter is not incremented. Figure 13.8 shows this operati

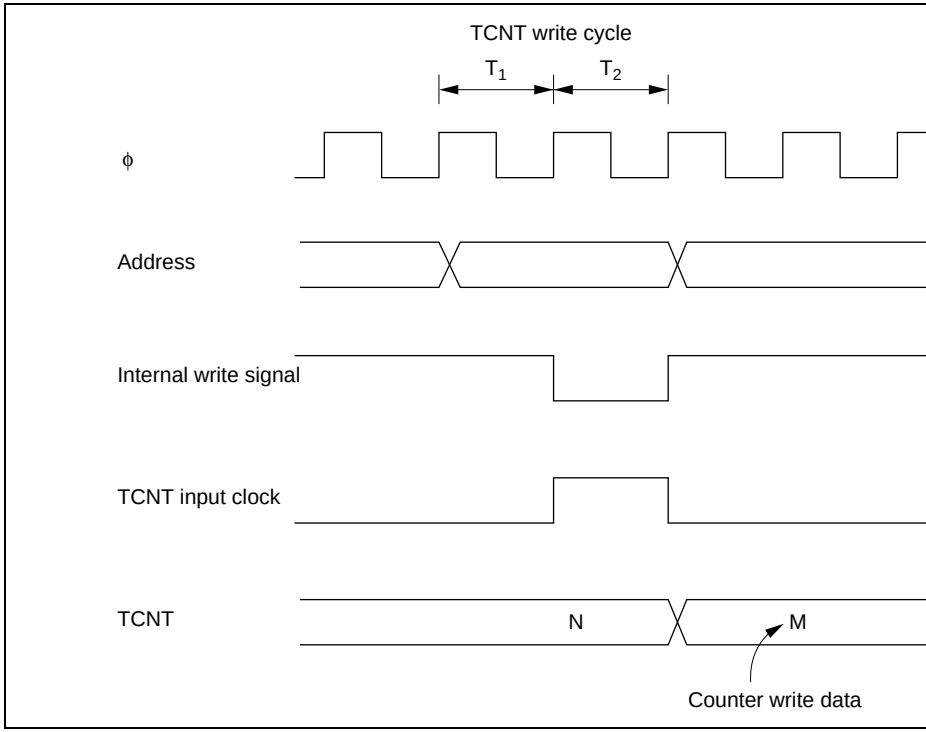


Figure 13.8 Contention between TCNT Write and Increment

If the mode is switched from watchdog timer to interval timer, or vice versa, while the
operating, errors could occur in the incrementation. Software must stop the watchdog ti
clearing the TME bit to 0) before switching the mode.

13.5.4 System Reset by $\overline{\text{WDTOVF}}$ Signal

If the $\overline{\text{WDTOVF}}$ output signal is input to the $\overline{\text{RES}}$ pin of the H8S/2655 Group, the H8S
Group will not be initialized correctly. Make sure that the $\overline{\text{WDTOVF}}$ signal is not input
to the $\overline{\text{RES}}$ pin. To reset the entire system by means of the $\overline{\text{WDTOVF}}$ signal, use the cir
in figure 13.9.

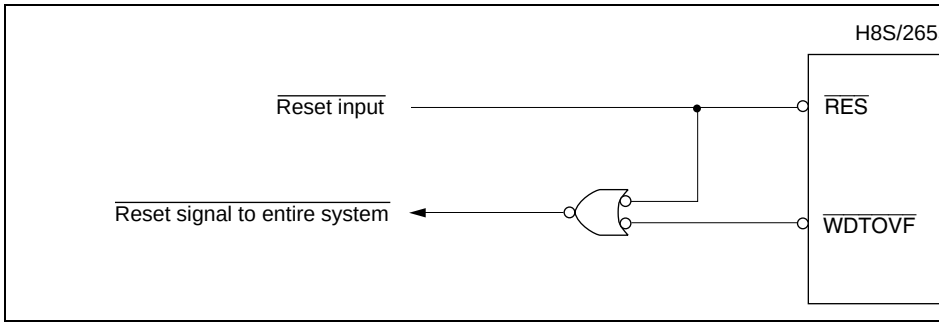


Figure 13.9 Circuit for System Reset by $\overline{\text{WDTOVF}}$ Signal (Example)

13.5.5 Internal Reset in Watchdog Timer Mode

The H8S/2655 Group is not reset internally if TCNT overflows while the RSTE bit is c
during watchdog timer operation, but TCNT and TSCR of the WDT are reset.

TCNT, TCSR, and RSTCR cannot be written to while the $\overline{\text{WDTOVF}}$ signal is low. Als
a read of the WOVF flag is not recognized during this period. To clear the WOVF falg,
read TCSR after the $\overline{\text{WDTOVF}}$ signal goes high, then write 0 to the WOVF flag.

processors (multiprocessor communication function).

14.1.1 Features

SCI features are listed below.

- Choice of asynchronous or clocked synchronous serial communication mode

  Asynchronous mode:

  — Serial data communication executed using asynchronous system in which sync is achieved character by character

    Serial data communication can be carried out with standard asynchronous com chips such as a Universal Asynchronous Receiver/Transmitter (UART) or Asy Communication Interface Adapter (ACIA)

  — A multiprocessor communication function is provided that enables serial data communication with a number of processors

  — Choice of 12 serial data transfer formats

Data length:	7 or 8 bits
Stop bit length:	1 or 2 bits
Parity:	Even, odd, or none
Multiprocessor bit:	1 or 0

  — Receive error detection: Parity, overrun, and framing errors

  — Break detection: Break can be detected by reading the RxD pin level directly in case of a framing error

  Clocked Synchronous mode:

  — Serial data communication synchronized with a clock

    Serial data communication can be carried out with other chips that have a sync communication function

  — One serial data transfer format

Data length:	8 bits

  — Receive error detection: Overrun errors detected

Note: * Descriptions in this section refer to LSB-first transfer.

- On-chip baud rate generator allows any bit rate to be selected
- Choice of serial clock source: internal clock from baud rate generator or external cl
  SCK pin
- Four interrupt sources
  — Four interrupt sources — transmit-data-empty, transmit-end, receive-data-full, a
  error — that can issue requests independently
  — The transmit-data-empty interrupt and receive data full interrupts can activate th
  controller (DMAC) or data transfer controller (DTC) to execute data transfer

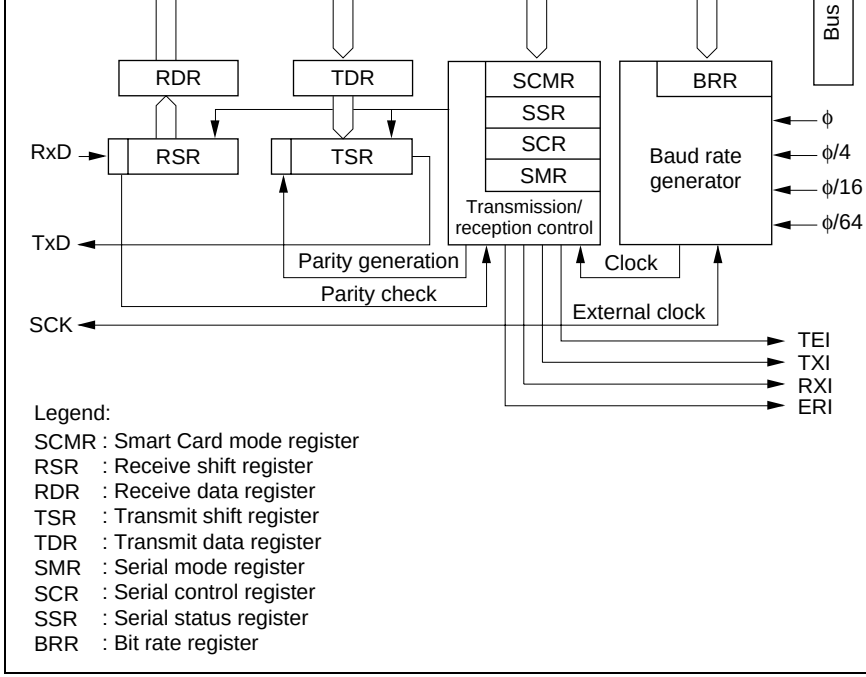


Figure 14.1 Block Diagram of SCI

	Transmit data pin 0	TxD_0	Output	SCI0 transmit data output
1	Serial clock pin 1	SCK_1	I/O	SCI1 clock input/output
	Receive data pin 1	RxD_1	Input	SCI1 receive data input
	Transmit data pin 1	TxD_1	Output	SCI1 transmit data output
2	Serial clock pin 2	SCK_2	I/O	SCI2 clock input/output
	Receive data pin 2	RxD_2	Input	SCI2 receive data input
	Transmit data pin 2	TxD_2	Output	SCI2 transmit data output

0	Serial mode register 0	SMR0	R/W	H'00
	Bit rate register 0	BRR0	R/W	H'FF
	Serial control register 0	SCR0	R/W	H'00
	Transmit data register 0	TDR0	R/W	H'FF
	Serial status register 0	SSR0	R/(W)*2	H'84
	Receive data register 0	RDR0	R	H'00
	Smart card mode register 0	SCMR0	R/W	H'F2
1	Serial mode register 1	SMR1	R/W	H'00
	Bit rate register 1	BRR1	R/W	H'FF
	Serial control register 1	SCR1	R/W	H'00
	Transmit data register 1	TDR1	R/W	H'FF
	Serial status register 1	SSR1	R/(W)*2	H'84
	Receive data register 1	RDR1	R	H'00
	Smart card mode register 1	SCMR1	R/W	H'F2
2	Serial mode register 2	SMR2	R/W	H'00
	Bit rate register 2	BRR2	R/W	H'FF
	Serial control register 2	SCR2	R/W	H'00
	Transmit data register 2	TDR2	R/W	H'FF
	Serial status register 2	SSR2	R/(W)*2	H'84
	Receive data register 2	RDR2	R	H'00
	Smart card mode register 2	SCMR2	R/W	H'F2
All	Module stop control register	MSTPCR	R/W	H'3FFF

Notes: 1. Lower 16 bits of the address.
2. Can only be written with 0 for flag clearing.

RSR is a register used to receive serial data.

The SCI sets serial data input from the RxD pin in RSR in the order received, starting w
LSB (bit 0), and converts it to parallel data. When one byte of data has been received, i
transferred to RDR automatically.

RSR cannot be directly read or written to by the CPU.

14.2.2 Receive Data Register (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

RDR is a register that stores received serial data.

When the SCI has received one byte of serial data, it transfers the received serial data f
RDR where it is stored, and completes the receive operation. After this, RSR is receive

Since RSR and RDR function as a double buffer in this way, enables continuous receiv
operations to be performed.

RDR is a read-only register, and cannot be written to by the CPU.

RDR is initialized to H'00 by a reset, and in standby mode or module stop mode.

To perform serial data transmission, the SCI first transfers transmit data from TDR to
sends the data to the TxD pin starting with the LSB (bit 0).

When transmission of one byte is completed, the next transmit data is transferred from
TSR, and transmission started, automatically. However, data transfer from TDR to TS
performed if the TDRE bit in SSR is set to 1.

TSR cannot be directly read or written to by the CPU.

14.2.4 Transmit Data Register (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

TDR is an 8-bit register that stores data for serial transmission.

When the SCI detects that TSR is empty, it transfers the transmit data written in TDR
starts serial transmission. Continuous serial transmission can be carried out by writing
transmit data to TDR during serial transmission of the data in TSR.

TDR can be read or written to by the CPU at all times.

TDR is initialized to H'FF by a reset, and in standby mode or module stop mode.

generator clock source.

SMR can be read or written to by the CPU at all times.

SMR is initialized to H'00 by a reset, and in standby mode or module stop mode.

Bit 7—Communication Mode (C/$\overline{A}$): Selects asynchronous mode or clocked synchro
as the SCI operating mode.

Bit 7

C/$\overline{A}$	Description	
0	Asynchronous mode	(
1	Clocked synchronous mode	

Bit 6—Character Length (CHR): Selects 7 or 8 bits as the data length in asynchronou
clocked synchronous mode, a fixed data length of 8 bits is used regardless of the CHR s

Bit 6

CHR	Description	
0	8-bit data	(
1	7-bit data*	

Note: * When 7-bit data is selected, the MSB (bit 7) of TDR is not transmitted, and it
possible to choose between LSB-first or MSB-first transfer.

Note: * When the PE bit is set to 1, the parity (even or odd) specified by the O/$\overline{E}$ bit transmit data before transmission. In reception, the parity bit is checked for (even or odd) specified by the O/$\overline{E}$ bit.

Bit 4—Parity Mode (O/$\overline{E}$): Selects either even or odd parity for use in parity addition checking.

The O/$\overline{E}$ bit setting is only valid when the PE bit is set to 1, enabling parity bit additio checking, in asynchronous mode. The O/$\overline{E}$ bit setting is invalid in clocked synchronou when parity addition and checking is disabled in asynchronous mode.

Bit 4

O/$\overline{E}$	Description
0	Even parity*1
1	Odd parity*2

Notes: 1. When even parity is set, parity bit addition is performed in transmission so t number of 1 bits in the transmit character plus the parity bit is even.
In reception, a check is performed to see if the total number of 1 bits in the character plus the parity bit is even.
2. When odd parity is set, parity bit addition is performed in transmission so th number of 1 bits in the transmit character plus the parity bit is odd.
In reception, a check is performed to see if the total number of 1 bits in the character plus the parity bit is odd.

Notes: 1. In transmission, a single 1 bit (stop bit) is added to the end of a transmit cha
before it is sent.
2. In transmission, two 1 bits (stop bits) are added to the end of a transmit char
before it is sent.

In reception, only the first stop bit is checked, regardless of the STOP bit setting. If the
stop bit is 1, it is treated as a stop bit; if it is 0, it is treated as the start bit of the next tra
character.

Bit 2—Multiprocessor Mode (MP): Selects multiprocessor format. When multiproces
is selected, the PE bit and O/$\overline{E}$ bit parity settings are invalid. The MP bit setting is only
asynchronous mode; it is invalid in clocked synchronous mode.

For details of the multiprocessor communication function, see section 14.3.3, Multipro
Communication Function.

Bit 2

MP	Description	
0	Multiprocessor function disabled	(
1	Multiprocessor format selected	

0	0	φ clock
	1	φ/4 clock
1	0	φ/16 clock
	1	φ/64 clock

14.2.6 Serial Control Register (SCR)

Bit	:	7	6	5	4	3	2	1
		TIE	RIE	TE	RE	MPIE	TEIE	CKE1
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

SCR is a register that performs enabling or disabling of SCI transfer operations, serial in asynchronous mode, and interrupt requests, and selection of the serial clock source.

SCR can be read or written to by the CPU at all times.

SCR is initialized to H'00 by a reset, and in standby mode or module stop mode.

Bit 7—Transmit Interrupt Enable (TIE): Enables or disables transmit data empty i (TXI) request generation when serial transmit data is transferred from TDR to TSR an flag in SSR is set to 1.

Bit 7

TIE	Description
0	Transmit data empty interrupt (TXI) requests disabled*
1	Transmit data empty interrupt (TXI) requests enabled

Note: * TXI interrupt request cancellation can be performed by reading 1 from the T then clearing it to 0, or clearing the TIE bit to 0.

1	Receive data full interrupt (RXI) request and receive error interrupt (ERI) re enabled

Note: * RXI and ERI interrupt request cancellation can be performed by reading 1 fr RDRF flag, or the FER, PER, or ORER flag, then clearing the flag to 0, or cl RIE bit to 0.

Bit 5—Transmit Enable (TE): Enables or disables the start of serial transmission by t

Bit 5

TE	Description
0	Transmission disabled*1 (
1	Transmission enabled*2

Notes: 1. The TDRE flag in SSR is fixed at 1.
2. In this state, serial transmission is started when transmit data is written to TD TDRE flag in SSR is cleared to 0.
SMR setting must be performed to decide the transfer format before setting to 1.

Bit 4—Receive Enable (RE): Enables or disables the start of serial reception by the SC

Bit 4

RE	Description
0	Reception disabled*1 (
1	Reception enabled*2

Notes: 1. Clearing the RE bit to 0 does not affect the RDRF, FER, PER, and ORER fla retain their states.
2. Serial reception is started in this state when a start bit is detected in asynchr mode or serial clock input is detected in clocked synchronous mode.
SMR setting must be performed to decide the transfer format before setting to 1.

[Clearing conditions]

- When the MPIE bit is cleared to 0
- When MPB= 1 data is received

1	Multiprocessor interrupts enabled*
	Receive interrupt (RXI) requests, receive error interrupt (ERI) requests, ar of the RDRF, FER, and ORER flags in SSR are disabled until data with th multiprocessor bit set to 1 is received.

Note: * When receive data including MPB = 0 is received, receive data transfer fro RDR, receive error detection, and setting of the RDRF, FER, and ORER fla is not performed. When receive data including MPB = 1 is received, the MP is set to 1, the MPIE bit is cleared to 0 automatically, and generation of RXI interrupts (when the TIE and RIE bits in SCR are set to 1) and FER and OR setting is enabled.

Bit 2—Transmit End Interrupt Enable (TEIE): Enables or disables transmit end in (TEI) request generation when there is no valid transmit data in TDR in MSB data tra

Bit 2

TEIE	Description
0	Transmit end interrupt (TEI) request disabled*
1	Transmit end interrupt (TEI) request enabled*

Note: * TEI cancellation can be performed by reading 1 from the TDRE flag in SSR clearing it to 0 and clearing the TEND flag to 0, or clearing the TEIE bit to 0

SMR before setting the CKE1 and CKE0 bits.

For details of clock source selection, see table 14.9.

Bit 1	Bit 0		
CKE1	**CKE0**	**Description**	
0	0	Asynchronous mode	Internal clock/SCK pin functions as I/O p
		Clocked synchronous mode	Internal clock/SCK pin functions as seria output
	1	Asynchronous mode	Internal clock/SCK pin functions as cloc
		Clocked synchronous mode	Internal clock/SCK pin functions as seria output
1	0	Asynchronous mode	External clock/SCK pin functions as cloc
		Clocked synchronous mode	External clock/SCK pin functions as seri input
	1	Asynchronous mode	External clock/SCK pin functions as cloc
		Clocked synchronous mode	External clock/SCK pin functions as seri input

Notes: 1. Initial value
2. Outputs a clock of the same frequency as the bit rate.
3. Inputs a clock with a frequency 16 times the bit rate.

14.2.7 Serial Status Register (SSR)

Bit	:	7	6	5	4	3	2	1	
		TDRE	RDRF	ORER	FER	PER	TEND	MPB	
Initial value	:	1	0	0	0	0	1	0	
R/W	:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	

Note: * Only 0 can be written, to clear the flag.

Bit 7—Transmit Data Register Empty (TDRE): Indicates that data has been transfe
TDR to TSR and the next serial data can be written to TDR.

Bit 7

TDRE	Description
0	[Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When the DMAC or DTC is activated by a TXI interrupt and write data
1	[Setting conditions] • When the TE bit in SCR is 0 • When data in transferred from TDR to TSR and data can be written to

Bit 6—Receive Data Register Full (RDRF): Indicates that the received data is stored

Bit 6

RDRF	Description
0	[Clearing conditions] • When 0 is written to RDRF after reading RDRF = 1 • When the DMAC or DTC is activated by an RXI interrupt and read data
1	[Setting condition] When serial reception ends normally and receive data is transferred from

Note: RDR and the RDRF flag are not affected and retain their previous values when detected during reception or when the RE bit in SCR is cleared to 0.
If reception of the next data is completed while the RDRF flag is still set to 1, an error will occur and the receive data will be lost.

When the next serial reception is completed while RDRF = 1*2

Notes: 1. The ORER flag is not affected and retains its previous state when the RE bit
cleared to 0.
2. The receive data prior to the overrun error is retained in RDR, and the data r
subsequently is lost. Also, subsequent serial reception cannot be continued
ORER flag is set to 1. In clocked synchronous mode, serial transmission can
continued, either.

Bit 4—Framing Error (FER): Indicates that a framing error occurred during reception
asynchronous mode, causing abnormal termination.

Bit 4

FER	Description
0	[Clearing condition] (In When 0 is written to FER after reading FER = 1
1	[Setting condition] When the SCI checks whether the stop bit at the end of the receive data wh
reception ends, and the stop bit is 0*2

Notes: 1. The FER flag is not affected and retains its previous state when the RE bit in
cleared to 0.
2. In 2-stop-bit mode, only the first stop bit is checked for a value of 0; the secc
is not checked. If a framing error occurs, the receive data is transferred to RI
RDRF flag is not set. Also, subsequent serial reception cannot be continued
FER flag is set to 1. In clocked synchronous mode, serial transmission canno
continued, either.

	When, in reception, the number of 1 bits in the receive data plus the parity match the parity setting (even or odd) specified by the O/$\overline{E}$ bit in SMR[*2]

Notes: 1. The PER flag is not affected and retains its previous state when the RE bit cleared to 0.

2. If a parity error occurs, the receive data is transferred to RDR but the RDRF set. Also, subsequent serial reception cannot be continued while the PER f

1. In clocked synchronous mode, serial transmission cannot be continued,

Bit 2—Transmit End (TEND): Indicates that there is no valid data in TDR when the the transmit character is sent, and transmission has been ended.

The TEND flag is read-only and cannot be modified.

Bit 2

TEND	Description
0	[Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When the DMAC or DTC is activated by a TXI interrupt and write data
1	[Setting conditions] • When the TE bit in SCR is 0 • When TDRE = 1 at transmission of the last bit of a 1-byte serial transm

	When data with a 0 multiprocessor bit is received
1	[Setting condition] When data with a 1 multiprocessor bit is received

Note: * Retains its previous state when the RE bit in SCR is cleared to 0 with multipr format.

Bit 0—Multiprocessor Bit Transfer (MPBT): When transmission is performed using multiprocessor format in asynchronous mode, MPBT stores the multiprocessor bit to be the transmit data.

The MPBT bit setting is invalid when multiprocessor format is not used, when not trans and in clocked synchronous mode.

Bit 0

MPBT	Description
0	Data with a 0 multiprocessor bit is transmitted (I
1	Data with a 1 multiprocessor bit is transmitted

generator operating clock selected by bits CKS1 and CKS0 in SMR.

BRR can be read or written to by the CPU at all times.

BRR is initialized to H'FF by a reset, and in standby mode or module stop mode.

As baud rate generator control is performed independently for each channel, different
be set for each channel.

Table 14.3 shows sample BRR settings in asynchronous mode, and table 14.4 shows s
settings in clocked synchronous mode.

Table 14.3 BRR Settings for Various Bit Rates (Asynchronous Mode)

	ϕ = 2 MHz			ϕ = 2.097152 MHz			ϕ = 2.4576 MHz			ϕ =
Bit Rate (bit/s)	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**
110	1	141	0.03	1	148	–0.04	1	174	–0.26	1	2
150	1	103	0.16	1	108	0.21	1	127	0.00	1	1
300	0	207	0.16	0	217	0.21	0	255	0.00	1	7
600	0	103	0.16	0	108	0.21	0	127	0.00	0	1
1200	0	51	0.16	0	54	–0.70	0	63	0.00	0	7
2400	0	25	0.16	0	26	1.14	0	31	0.00	0	3
4800	0	12	0.16	0	13	–2.48	0	15	0.00	0	1
9600	0	6	–6.99	0	6	–2.48	0	7	0.00	0	9
19200	0	2	8.51	0	2	13.78	0	3	0.00	0	4
31250	0	1	0.00	0	1	4.86	0	1	22.88	0	2
38400	0	1	–18.62	0	1	–14.67	0	1	0.00	0	1

1200	0	95	0.00	0	103	0.16	0	127	0.00	0	12
2400	0	47	0.00	0	51	0.16	0	63	0.00	0	64
4800	0	23	0.00	0	25	0.16	0	31	0.00	0	32
9600	0	11	0.00	0	12	0.16	0	15	0.00	0	15
19200	0	5	0.00	0	6	–6.99	0	7	0.00	0	7
31250	0	3	–7.84	0	3	0.00	0	4	–1.70	0	4
38400	0	2	0.00	0	2	8.51	0	3	0.00	0	3

	ϕ = 6 MHz			**ϕ = 6.144 MHz**			**ϕ = 7.3728 MHz**			**ϕ =**	
Bit Rate (bit/s)	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**
110	2	106	–0.44	2	108	0.08	2	130	–0.07	2	14
150	2	77	0.16	2	79	0.00	2	95	0.00	2	10
300	1	155	0.16	1	159	0.00	1	191	0.00	1	20
600	1	77	0.16	1	79	0.00	1	95	0.00	1	10
1200	0	155	0.16	0	159	0.00	0	191	0.00	0	20
2400	0	77	0.16	0	79	0.00	0	95	0.00	0	10
4800	0	38	0.16	0	39	0.00	0	47	0.00	0	51
9600	0	19	–2.34	0	19	0.00	0	23	0.00	0	25
19200	0	9	–2.34	0	9	0.00	0	11	0.00	0	12
31250	0	5	0.00	0	5	2.40	0	6	5.33	0	7
38400	0	4	–2.34	0	4	0.00	0	5	0.00	0	6

1200	0	255	0.00	1	64	0.16	1	77	0.16	1	7
2400	0	127	0.00	0	129	0.16	0	155	0.16	0	1
4800	0	63	0.00	0	64	0.16	0	77	0.16	0	7
9600	0	31	0.00	0	32	–1.36	0	38	0.16	0	3
19200	0	15	0.00	0	15	1.73	0	19	–2.34	0	1
31250	0	9	–1.70	0	9	0.00	0	11	0.00	0	1
38400	0	7	0.00	0	7	1.73	0	9	–2.34	0	9

	ϕ = 14 MHz			**ϕ = 14.7456 MHz**			**ϕ = 16 MHz**			**ϕ = 17**
Bit Rate (bit/s)	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**
110	2	248	–0.17	3	64	0.70	3	70	0.03	3	7
150	2	181	0.16	2	191	0.00	2	207	0.16	2	2
300	2	90	0.16	2	95	0.00	2	103	0.16	2	1
600	1	181	0.16	1	191	0.00	1	207	0.16	1	2
1200	1	90	0.16	1	95	0.00	1	103	0.16	1	1
2400	0	181	0.16	0	191	0.00	0	207	0.16	0	2
4800	0	90	0.16	0	95	0.00	0	103	0.16	0	1
9600	0	45	–0.93	0	47	0.00	0	51	0.16	0	5
19200	0	22	–0.93	0	23	0.00	0	25	0.16	0	2
31250	0	13	0.00	0	14	–1.70	0	15	0.00	0	1
38400	0	10	3.57	0	11	0.00	0	12	0.16	0	1

1200	1	116	0.16	1	127	0.00	1	129	0.16
2400	0	233	0.16	0	255	0.00	1	64	0.16
4800	0	116	0.16	0	127	0.00	0	129	0.16
9600	0	58	–0.69	0	63	0.00	0	64	0.16
19200	0	28	1.02	0	31	0.00	0	32	–1.36
31250	0	17	0.00	0	19	–1.70	0	19	0.00
38400	0	14	–2.34	0	15	0.00	0	15	1.73

2.5 k	0	199	1	99	1	199	1	249	2	99	2
5 k	0	99	0	199	1	99	1	124	1	199	1
10 k	0	49	0	99	0	199	0	249	1	99	1
25 k	0	19	0	39	0	79	0	99	0	159	0
50 k	0	9	0	19	0	39	0	49	0	79	0
100 k	0	4	0	9	0	19	0	24	0	39	0
250 k	0	1	0	3	0	7	0	9	0	15	0
500 k	0	0*	0	1	0	3	0	4	0	7	0
1 M			0	0*	0	1	—	—	0	3	0
2.5 M					—	—	0	0*	—	—	0
5 M									—	—	0

Legend:

Blank: Cannot be set.

—: Can be set, but there will be a degree of error.

*: Continuous transfer is not possible.

Note: As far as possible, the setting should be made so that the error is no more than

$$N = \frac{\phi}{8 \times 2^{2n-1} \times B} \times 10^6 - 1$$

Where B: Bit rate (bit/s)
N: BRR setting for baud rate generator ($0 \leq N \leq 255$)
ϕ: Operating frequency (MHz)
n: Baud rate generator input clock (n = 0 to 3)
(See the table below for the relation between n and the clock.)

		SMR Setting	
n	**Clock**	**CKS1**	**CKS0**
0	ϕ	0	0
1	$\phi/4$	0	1
2	$\phi/16$	1	0
3	$\phi/64$	1	1

The bit rate error in asynchronous mode is found from the following formula:

$$\text{Error (\%)} = \left\{ \frac{\phi \times 10^6}{(N + 1) \times B \times 64 \times 2^{2n-1}} - 1 \right\} \times 100$$

2.4576	76800	0	0
3	93750	0	0
3.6864	115200	0	0
4	125000	0	0
4.9152	153600	0	0
5	156250	0	0
6	187500	0	0
6.144	192000	0	0
7.3728	230400	0	0
8	250000	0	0
9.8304	307200	0	0
10	312500	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
19.6608	614400	0	0
20	625000	0	0

4	1.0000	62500
4.9152	1.2288	76800
5	1.2500	78125
6	1.5000	93750
6.144	1.5360	96000
7.3728	1.8432	115200
8	2.0000	125000
9.8304	2.4576	153600
10	2.5000	156250
12	3.0000	187500
12.288	3.0720	192000
14	3.5000	218750
14.7456	3.6864	230400
16	4.0000	250000
17.2032	4.3008	268800
18	4.5000	281250
19.6608	4.9152	307200
20	5.0000	312500

12	2.0000	2000000.0
14	2.3333	2333333.3
16	2.6667	2666666.7
18	3.0000	3000000.0
20	3.3333	3333333.3

mode 7-bit data, LSB-first or MSB-first can be selected regardless of the serial commu
mode. The descriptions in this chapter refer to LSB-first transfer.

For details of the other bits in SCMR, see 15.2.1, Smart Card Mode Register (SCMR).

SCMR is initialized to H'F2 by a reset, and in standby mode or module stop mode.

Bits 7 to 4—Reserved: Read-only bits, always read as 1.

Bit 3—Smart Card Data Transfer Direction (SDIR): Selects the serial/parallel conv
format. This bit is valid when 8-bit data is used as the transmit/receive format.

Bit 3

SDIR	Description	
0	TDR contents are transmitted LSB-first	(
	Receive data is stored in RDR LSB-first	
1	TDR contents are transmitted MSB-first	
	Receive data is stored in RDR MSB-first	

Bit 2—Smart Card Data Invert (SINV): When the smart card interface operates as a
SCI, 0 should be written in this bit.

Bit 1—Reserved: Read-only bit, always read as 1.

Bit 0—Smart Card Interface Mode Select (SMIF): When the smart card interface op
normal SCI, 0 should be written in this bit.

MSTPCR is a 16-bit readable/writable register that performs module stop mode contro

When the corresponding bit of bits MSTP7 to MSTP5 is set to 1, SCI operation stops the bus cycle and a transition is made to module stop mode. Registers cannot be read o in module stop mode. For details, see section 21.5, Module Stop Mode.

MSTPCR is initialized to H'3FFF by a reset and in hardware standby mode. It is not in software standby mode.

Bit 7—Module Stop (MSTP7): Specifies the SCI channel 2 module stop mode.

Bit 7

MSTP7	Description
0	SCI channel 2 module stop mode cleared
1	SCI channel 2 module stop mode set

Bit 6—Module Stop (MSTP6): Specifies the SCI channel 1 module stop mode.

Bit 6

MSTP6	Description
0	SCI channel 1 module stop mode cleared
1	SCI channel 1 module stop mode set

Bit 5—Module Stop (MSTP5): Specifies the SCI channel 0 module stop mode.

Bit 5

MSTP5	Description
0	SCI channel 0 module stop mode cleared
1	SCI channel 0 module stop mode set

Selection of asynchronous or clocked synchronous mode and the transmission format is using SMR as shown in table 14.8. The SCI clock is determined by a combination of th in SMR and the CKE1 and CKE0 bits in SCR, as shown in table 14.9.

Asynchronous Mode

- Data length: Choice of 7 or 8 bits
- Choice of parity addition, multiprocessor bit addition, and addition of 1 or 2 stop bi combination of these parameters determines the transfer format and character lengt
- Detection of framing, parity, and overrun errors, and breaks, during reception
- Choice of internal or external clock as SCI clock source
  - When internal clock is selected:

    The SCI operates on the baud rate generator clock and a clock with the same fre the bit rate can be output
  - When external clock is selected:

    A clock with a frequency of 16 times the bit rate must be input (the on-chip bau generator is not used)

Clocked Synchronous Mode

- Transfer format: Fixed 8-bit data
- Detection of overrun errors during reception
- Choice of internal or external clock as SCI clock source
  - When internal clock is selected:

    The SCI operates on the baud rate generator clock and a serial clock is output of
  - When external clock is selected:

    The on-chip baud rate generator is not used, and the SCI operates on the input s

			1	0				Yes
				1				
	1		0	0		7-bit data		No
				1				
			1	0				Yes
				1				
	0	1	—	0	Asynchronous mode (multi-processor format)	8-bit data	Yes	No
			—	1				
	1		—	0		7-bit data		
			—	1				
1	—	—	—	—	Clocked synchronous mode	8-bit data	No	

Table 14.9 SMR and SCR Settings and SCI Clock Source Selection

SMR	SCR Setting			SCI Transmit/Receive Clock	
Bit 7	Bit 1	Bit 0		Clock Source	SCK Pin Function
C/$\overline{A}$	CKE1	CKE0	Mode		
0	0	0	Asynchronous mode	Internal	SCI does not use SCK pin
		1			Outputs clock with same frequ rate
	1	0		External	Inputs clock with frequency of the bit rate
		1			
1	0	0	Clocked synchronous mode	Internal	Outputs serial clock
		1			
	1	0		External	Inputs serial clock
		1			

that data can be read or written during transmission or reception, enabling continuous d
transfer.

Figure 14.2 shows the general format for asynchronous serial communication.

In asynchronous serial communication, the transmission line is usually held in the mark
level). The SCI monitors the transmission line, and when it goes to the space state (low
recognizes a start bit and starts serial communication.

One serial communication character consists of a start bit (low level), followed by data
first order), a parity bit (high or low level), and finally one or two stop bits (high level).

In asynchronous mode, the SCI performs synchronization at the falling edge of the star
reception. The SCI samples the data on the 8th pulse of a clock with a frequency of 16
length of one bit, so that the transfer data is latched at the center of each bit.

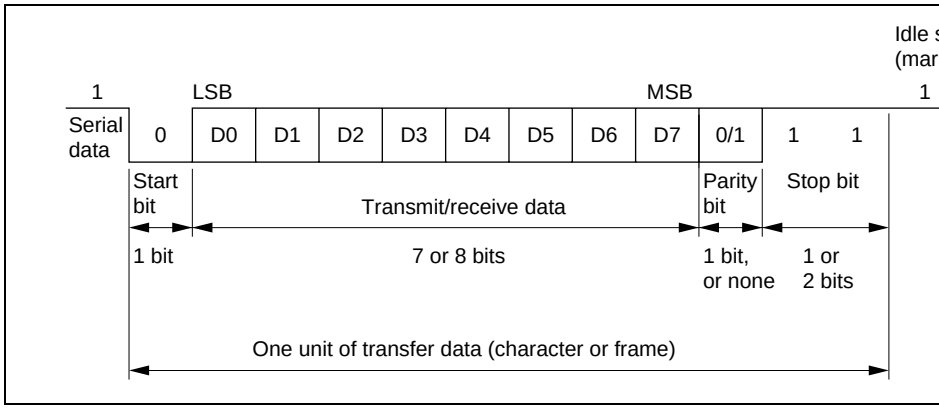


Figure 14.2 Data Format in Asynchronous Communication (Example with 8-Bit Data, Parity, Two Stop Bits)

0	0	0	0	S 8-bit data STOP
0	0	0	1	S 8-bit data STOP STO
0	1	0	0	S 8-bit data P STO
0	1	0	1	S 8-bit data P STO
1	0	0	0	S 7-bit data STOP
1	0	0	1	S 7-bit data STOP STOP
1	1	0	0	S 7-bit data P STOP
1	1	0	1	S 7-bit data P STOP STO
0	—	1	0	S 8-bit data MPB STO
0	—	1	1	S 8-bit data MPB STO
1	—	1	0	S 7-bit data MPB STOP
1	—	1	1	S 7-bit data MPB STOP STO

Legend:

S : Start bit

STOP : Stop bit

P : Parity bit

MPB : Multiprocessor bit

When the SCI is operated on an internal clock, the clock can be output from the SCK p
frequency of the clock output in this case is equal to the bit rate, and the phase is such t
rising edge of the clock is in the middle of the transmit data, as shown in figure 14.3.

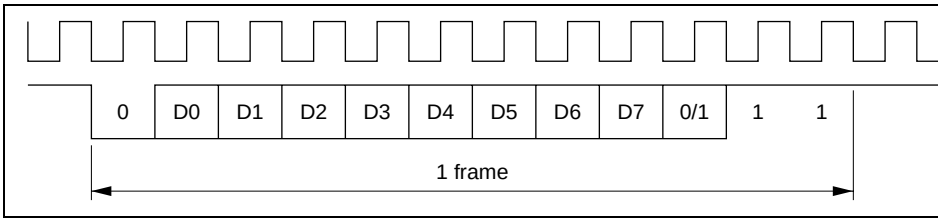


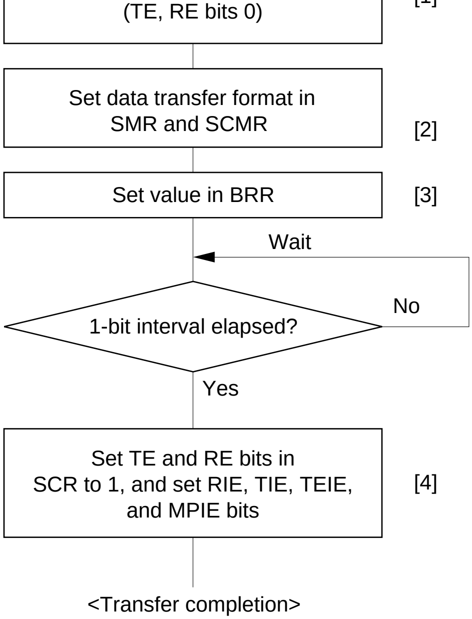
Figure 14.3 Relation between Output Clock and Transfer Data Phase (Asynchronous Mode)

Data Transfer Operations

SCI initialization (asynchronous mode): Before transmitting and receiving data, you clear the TE and RE bits in SCR to 0, then initialize the SCI as described below.

When the operating mode, transfer format, etc., is changed, the TE and RE bits must be 0 before making the change using the following procedure. When the TE bit is cleared TDRE flag is set to 1 and TSR is initialized. Note that clearing the RE bit to 0 does not contents of the RDRF, PER, FER, and ORER flags, or the contents of RDR.

When an external clock is used the clock should not be stopped during operation, inclu initialization, since operation is uncertain.



immediately after SCR setting made.

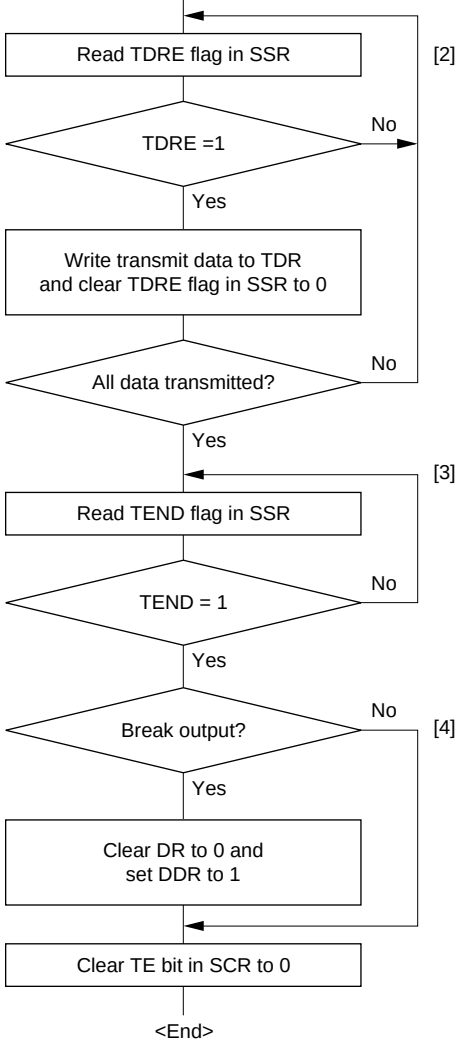
[2] Set the data transfer format i and SCMR.

[3] Write a value corresponding bit rate to BRR. Not necessa external clock is used.

[4] Wait at least one bit interval, set the TE bit or RE bit in SC Also set the RIE, TIE, TEIE, a MPIE bits.

Setting the TE and RE bits e the TxD and RxD pins to be u

Figure 14.4 Sample SCI Initialization Flowchart



After the TE bit is set to 1, a fram
of 1s is output, and transmission
enabled.

[2] SCI status check and transmit da
write:
Read SSR and check that the
TDRE flag is set to 1, then write
transmit data to TDR and clear t
TDRE flag to 0.

[3] Serial transmission continuation
procedure:
To continue serial transmission,
read 1 from the TDRE flag to
confirm that writing is possible,
then write data to TDR, and then
clear the TDRE flag to 0. Checki
and clearing of the TDRE flag is
automatic when the DMAC or DT
is activated by a transmit data
empty interrupt (TXI) request, an
data is written to TDR.

[4] Break output at the end of serial
transmission:
To output a break in serial
transmission, set DDR for the po
corresponding to the TxD pin to
clear DR to 0, then clear the TE
in SCR to 0.

Figure 14.5 Sample Serial Transmission Flowchart

The serial transmit data is sent from the TXD pin in the following order.

[a] Start bit:

One 0-bit is output.

[b] Transmit data:

8-bit or 7-bit data is output in LSB-first order.

[c] Parity bit or multiprocessor bit:

One parity bit (even or odd parity), or one multiprocessor bit is output.

A format in which neither a parity bit nor a multiprocessor bit is output can als selected.

[d] Stop bit(s):

One or two 1-bits (stop bits) are output.

[e] Mark state:

1 is output continuously until the start bit that starts the next transmission is se

[3] The SCI checks the TDRE flag at the timing for sending the stop bit.

If the TDRE flag is cleared to 0, the data is transferred from TDR to TSR, the stop and then serial transmission of the next frame is started.

If the TDRE flag is set to 1, the TEND flag in SSR is set to 1, the stop bit is sent, a "mark state" is entered in which 1 is output continuously. If the TEIE bit in SCR is this time, a TEI interrupt request is generated.

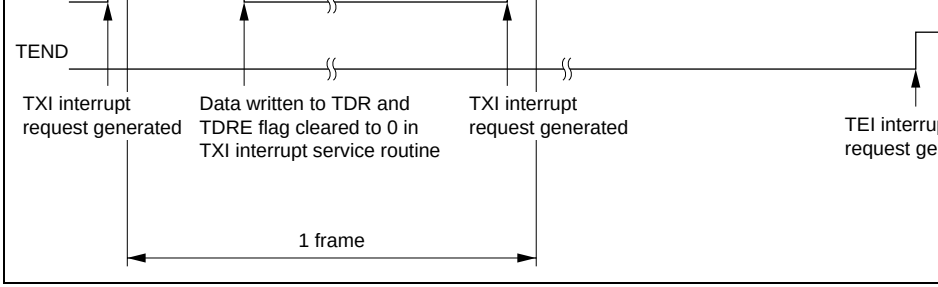
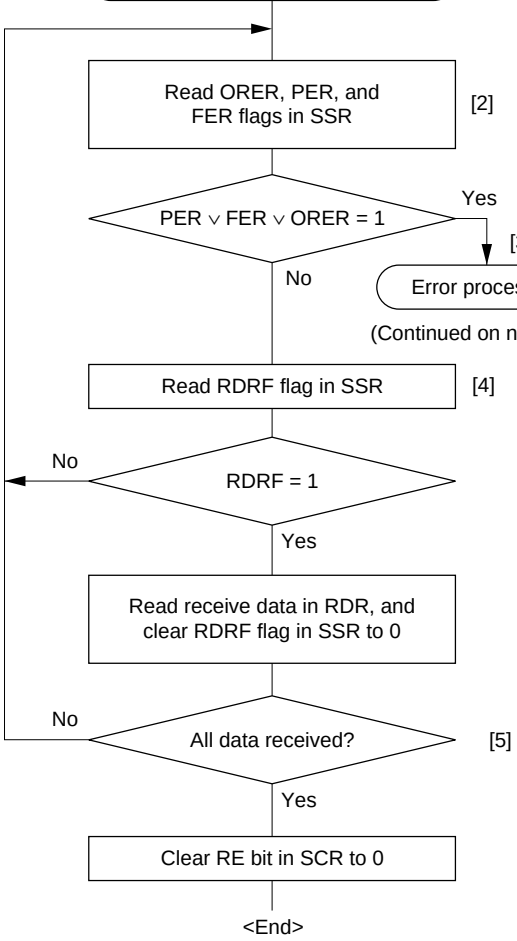


Figure 14.6 Example of Operation in Transmission in Asynchronous Mo
(Example with 8-Bit Data, Parity, One Stop Bit)



[2] [3] Receive error process
break detection:
If a receive error occurs,
ORER, PER, and FER fl
SSR to identify the error.
performing the appropria
processing, ensure that t
ORER, PER, and FER fl
all cleared to 0. Receptio
be resumed if any of thes
are set to 1. In the case
framing error, a break ca
detected by reading the
the input port correspond
the RxD pin.

[4] SCI status check and rec
data read :
Read SSR and check tha
= 1, then read the receiv
RDR and clear the RDR
0. Transition of the RDR
from 0 to 1 can also be i
by an RXI interrupt.

[5] Serial reception continua
procedure:
To continue serial recept
before the stop bit for the
frame is received, read t
RDRF flag, read RDR, a
the RDRF flag to 0. The
flag is cleared automatic
when the DMAC or DTC
activated by an RXI inter
the RDR value is read.

Figure 14.7 Sample Serial Reception Data Flowchart

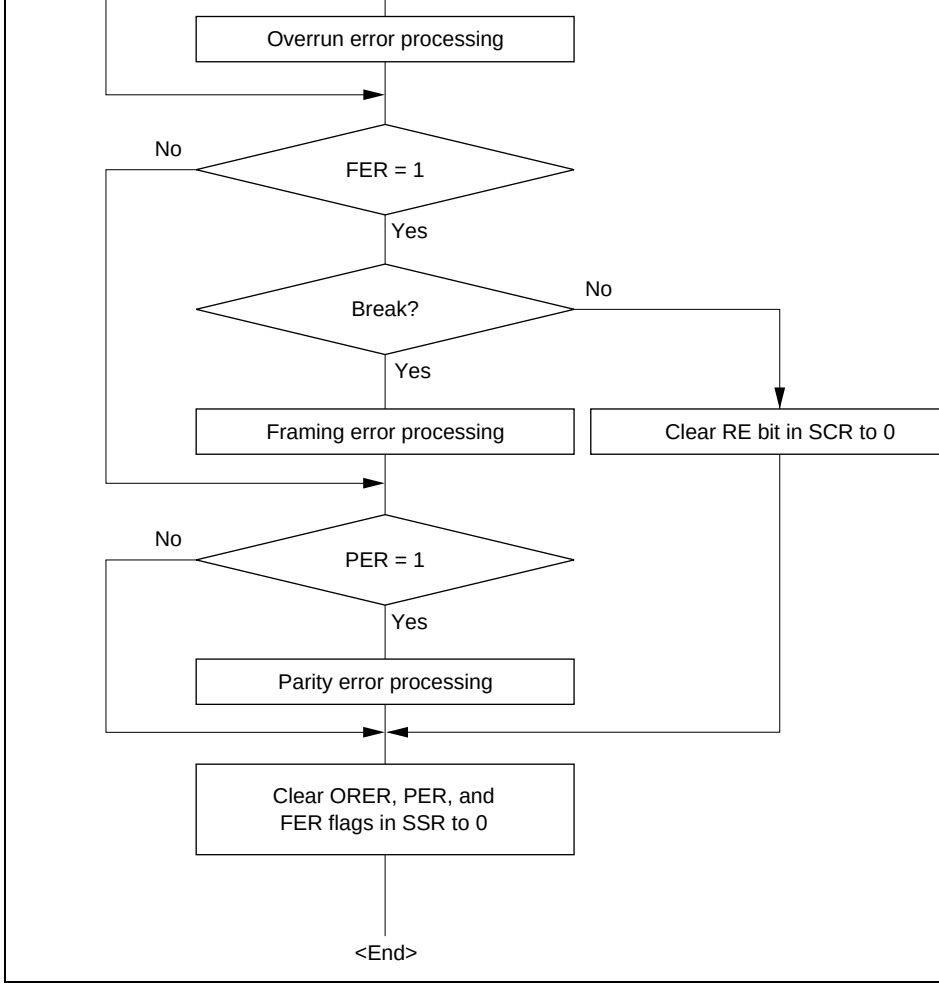


Figure 14.7 Sample Serial Reception Data Flowchart (cont)

[a] Parity check:

The SCI checks whether the number of 1 bits in the receive data agrees with th (even or odd) set in the O/$\overline{E}$ bit in SMR.

[b] Stop bit check:

The SCI checks whether the stop bit is 1.

If there are two stop bits, only the first is checked.

[c] Status check:

The SCI checks whether the RDRF flag is 0, indicating that the receive data ca transferred from RSR to RDR.

If all the above checks are passed, the RDRF flag is set to 1, and the receive da in RDR.

If a receive error* is detected in the error check, the operation is as shown in ta

Note: * Subsequent receive operations cannot be performed when a receive error h Also note that the RDRF flag is not set to 1 in reception, and so the error fl cleared to 0.

[4] If the RIE bit in SCR is set to 1 when the RDRF flag changes to 1, a receive data f (RXI) request is generated.

Also, if the RIE bit in SCR is set to 1 when the ORER, PER, or FER flag changes receive error interrupt (ERI) request is generated.

Figure 14.8 shows an example of the operation for reception in asynchronous mode.

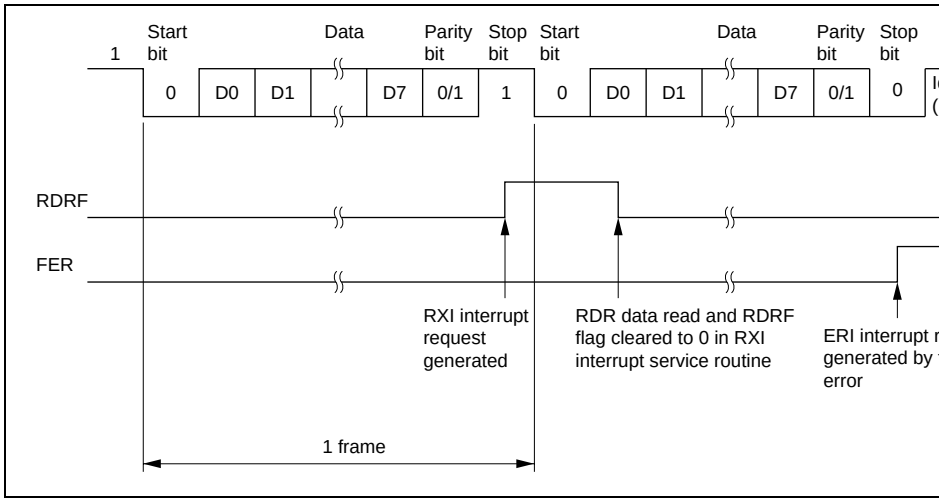


Figure 14.8 Example of SCI Operation in Reception (Example with 8-Bit Data, Parity, One Stop Bit)

unique ID code.

The serial communication cycle consists of two component cycles: an ID transmission which specifies the receiving station, and a data transmission cycle. The multiprocesso to differentiate between the ID transmission cycle and the data transmission cycle.

The transmitting station first sends the ID of the receiving station with which it wants serial communication as data with a 1 multiprocessor bit added. It then sends transmit with a 0 multiprocessor bit added.

The receiving station skips the data until data with a 1 multiprocessor bit is sent.

When data with a 1 multiprocessor bit is received, the receiving station compares that own ID. The station whose ID matches then receives the data sent next. Stations whos not match continue to skip the data until data with a 1 multiprocessor bit is again recei way, data communication is carried out among a number of processors.

Figure 14.9 shows an example of inter-processor communication using the multiproce

Data Transfer Format

There are four data transfer formats.

When the multiprocessor format is specified, the parity bit specification is invalid.

For details, see table 14.10.

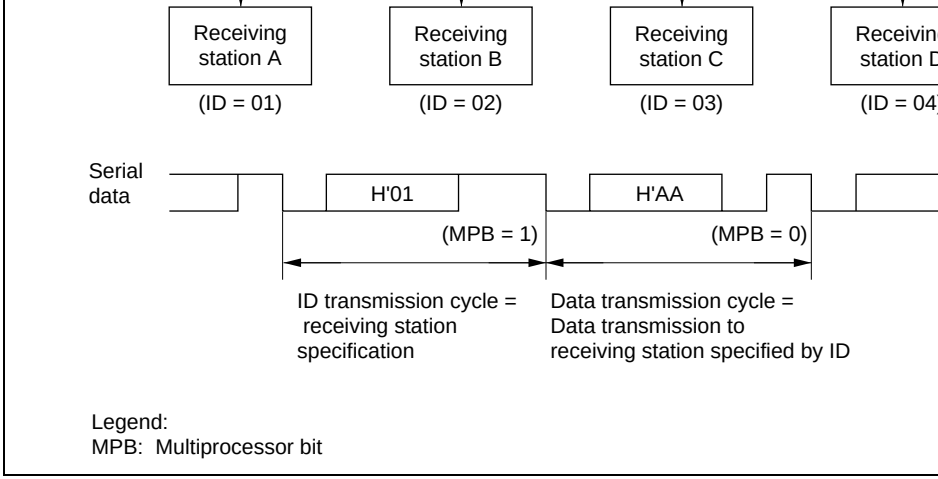
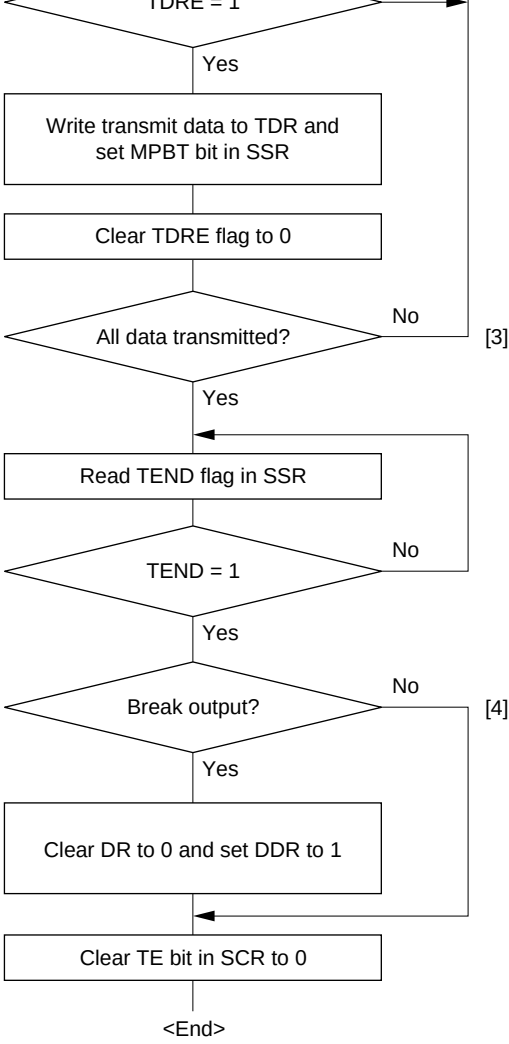


Figure 14.9 Example of Inter-Processor Communication Using Multiprocessor (Transmission of Data H'AA to Receiving Station A)

Data Transfer Operations

Multiprocessor serial data transmission: Figure 14.10 shows a sample flowchart for multiprocessor serial data transmission.

The following procedure should be used for multiprocessor serial data transmission.



data write:
Read SSR and check that t
TDRE flag is set to 1, then w
transmit data to TDR. Set th
MPBT bit in SSR to 0 or 1.
Finally, clear the TDRE flag

[3] Serial transmission continua
procedure:
To continue serial transmiss
be sure to read 1 from the T
flag to confirm that writing is
possible, then write data to
and then clear the TDRE fla
0. Checking and clearing of
TDRE flag is automatic whe
DMAC or DTC is activated b
transmit data empty interrup
(TXI) request, and data is w
to TDR.

[4] Break output at the end of s
transmission:
To output a break in serial
transmission, set the port D
1, clear DR to 0, then clear
TE bit in SCR to 0.

Figure 14.10 Sample Multiprocessor Serial Transmission Flowchart

The serial transmit data is sent from the TXD pin in the following order.

[a] Start bit:

One 0-bit is output.

[b] Transmit data:

8-bit or 7-bit data is output in LSB-first order.

[c] Multiprocessor bit

One multiprocessor bit (MPBT value) is output.

[d] Stop bit(s):

One or two 1-bits (stop bits) are output.

[e] Mark state:

1 is output continuously until the start bit that starts the next transmission is sent

[3] The SCI checks the TDRE flag at the timing for sending the stop bit.

If the TDRE flag is cleared to 0, data is transferred from TDR to TSR, the stop bit i then serial transmission of the next frame is started.

If the TDRE flag is set to 1, the TEND flag in SSR is set to 1, the stop bit is sent, ar mark state is entered in which 1 is output continuously. If the TEIE bit in SCR is se time, a transmission end interrupt (TEI) request is generated.

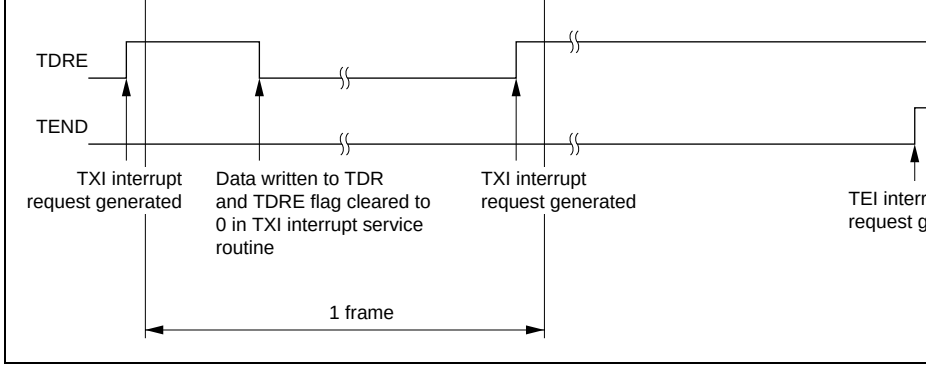
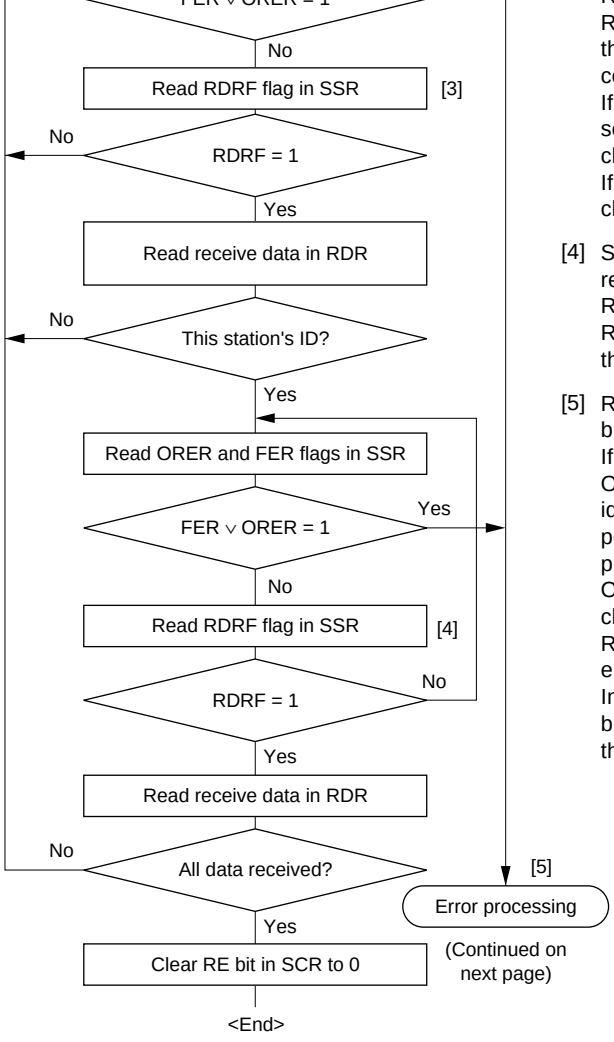


Figure 14.11 Example of SCI Operation in Transmission (Example with 8-Bit Data, Multiprocessor Bit, One Stop Bit)

Multiprocessor serial data reception: Figure 14.12 shows a sample flowchart for mu serial reception.

The following procedure should be used for multiprocessor serial data reception.



[4] SCI status check and data reception:
Read SSR and check that th
RDRF flag is set to 1, then r
the data in RDR.

[5] Receive error processing an
break detection:
If a receive error occurs, rea
ORER and FER flags in SSI
identify the error. After
performing the appropriate e
processing, ensure that the
ORER and FER flags are all
cleared to 0.
Reception cannot be resume
either of these flags is set to
In the case of a framing erro
break can be detected by re
the RxD pin value.

Figure 14.12 Sample Multiprocessor Serial Reception Flowchart

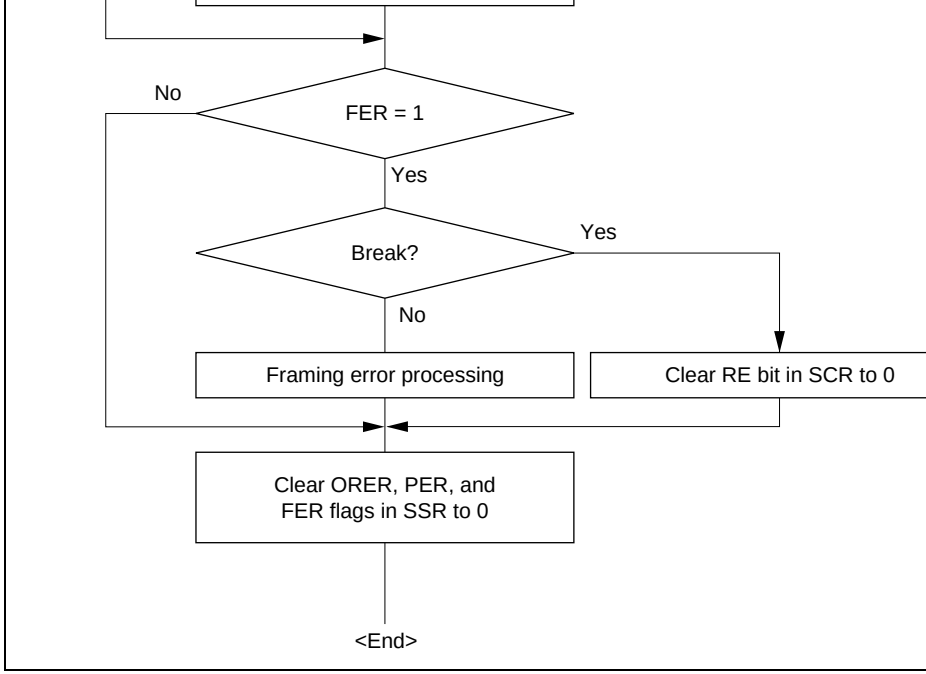


Figure 14.12 Sample Multiprocessor Serial Reception Flowchart (con

Figure 14.13 shows an example of SCI operation for multiprocessor format reception.

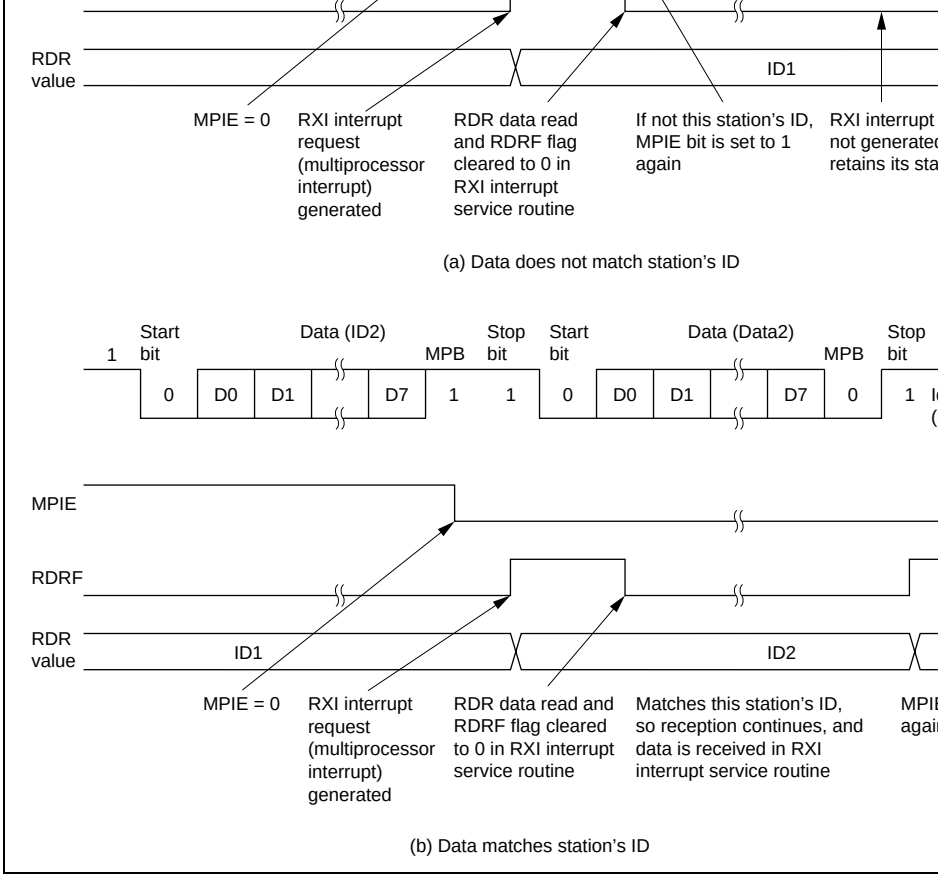


Figure 14.13 Example of SCI Operation in Reception (Example with 8-Bit Data, Multiprocessor Bit, One Stop Bit)

Figure 14.14 shows the general format for clocked synchronous serial communication

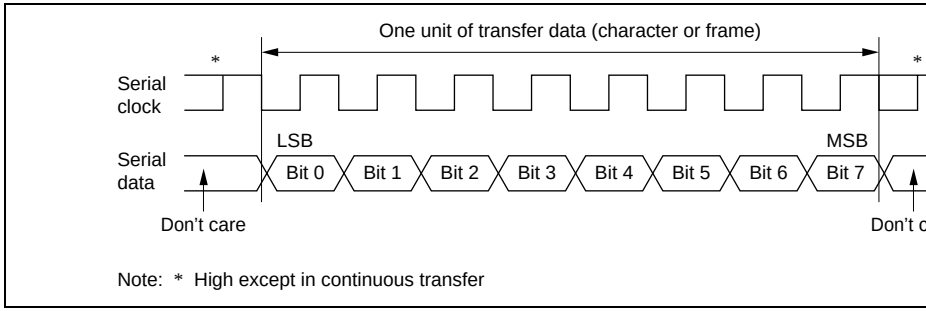


Figure 14.14 Data Format in Synchronous Communication

In clocked synchronous serial communication, data on the transmission line is output
falling edge of the serial clock to the next. Data confirmation is guaranteed at the risin
the serial clock.

In clocked serial communication, one character consists of data output starting with th
ending with the MSB. After the MSB is output, the transmission line holds the MSB s

In clocked synchronous mode, the SCI receives data in synchronization with the rising
serial clock.

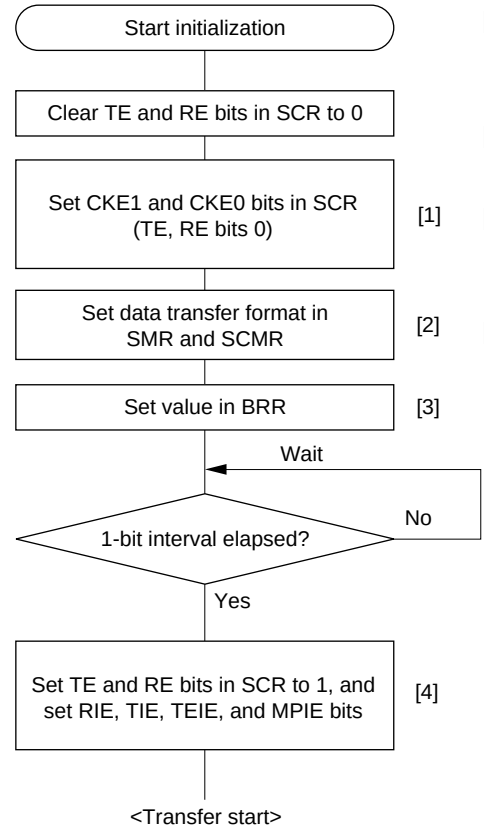
Data Transfer Format

A fixed 8-bit data format is used.

No parity or multiprocessor bits are added.

performed the clock is fixed high. When only receive operations are performed, howev
serial clock is output until an overrun error occurs or the RE bit is cleared to 0. If you w
perform receive operations in units of one character, you should select an external cloc
clock source.

Figure 14.15 shows a sample SCI initialization flowchart.



[1] Set the clock selection in SCR. B
to clear bits RIE, TIE, TEIE, and
TE and RE, to 0.

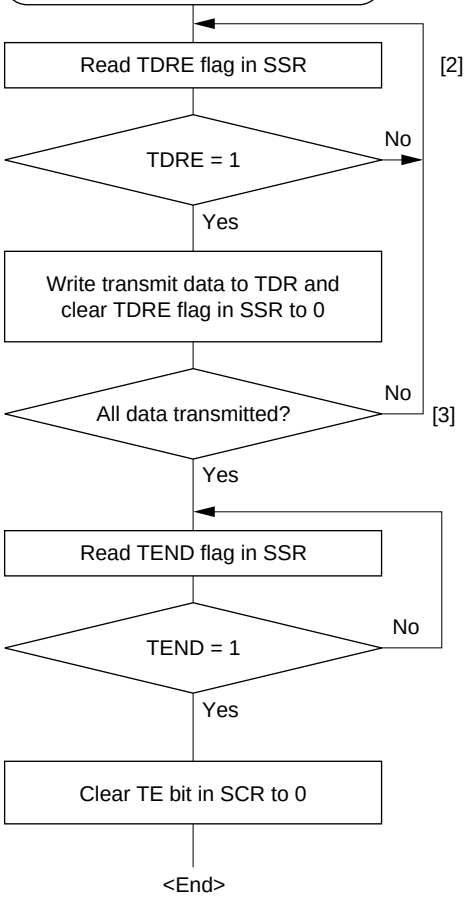
[2] Set the data transfer format in SM
and SCMR.

[3] Write a value corresponding to th
rate to BRR. Not necessary if an
external clock is used.

[4] Wait at least one bit interval, then
the TE bit or RE bit in SCR to 1.
Also set the RIE, TIE, TEIE, and
bits.
Setting the TE and RE bits enable
TxD and RxD pins to be used.

Note: In simultaneous transmit and receive operations, the TE and RE bits should
cleared to 0 or set to 1 simultaneously.

Figure 14.15 Sample SCI Initialization Flowchart



pin.

[2] SCI status check and transmit d
write:
Read SSR and check that the T
flag is set to 1, then write transm
to TDR and clear the TDRE flag

[3] Serial transmission continuation
procedure:
To continue serial transmission,
sure to read 1 from the TDRE fl
confirm that writing is possible,
write data to TDR, and then clea
TDRE flag to 0.
Checking and clearing of the TD
flag is automatic when the DMA
DTC is activated by a transmit d
empty interrupt (TXI) request ar
is written to TDR.

Figure 14.16 Sample Serial Transmission Flowchart

external clock has been specified, data is output synchronized with the input clock

The serial transmit data is sent from the TxD pin starting with the LSB (bit 0) and the MSB (bit 7).

[3] The SCI checks the TDRE flag at the timing for sending the MSB (bit 7).

If the TDRE flag is cleared to 0, data is transferred from TDR to TSR, and serial tr of the next frame is started.

If the TDRE flag is set to 1, the TEND flag in SSR is set to 1, the MSB (bit 7) is se TxD pin maintains its state.

If the TEIE bit in SCR is set to 1 at this time, a TEI interrupt request is generated.

[4] After completion of serial transmission, the SCK pin is fixed.

Figure 14.17 shows an example of SCI operation in transmission.

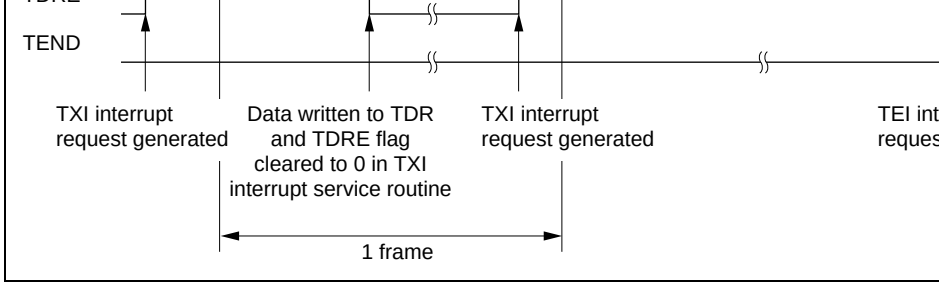


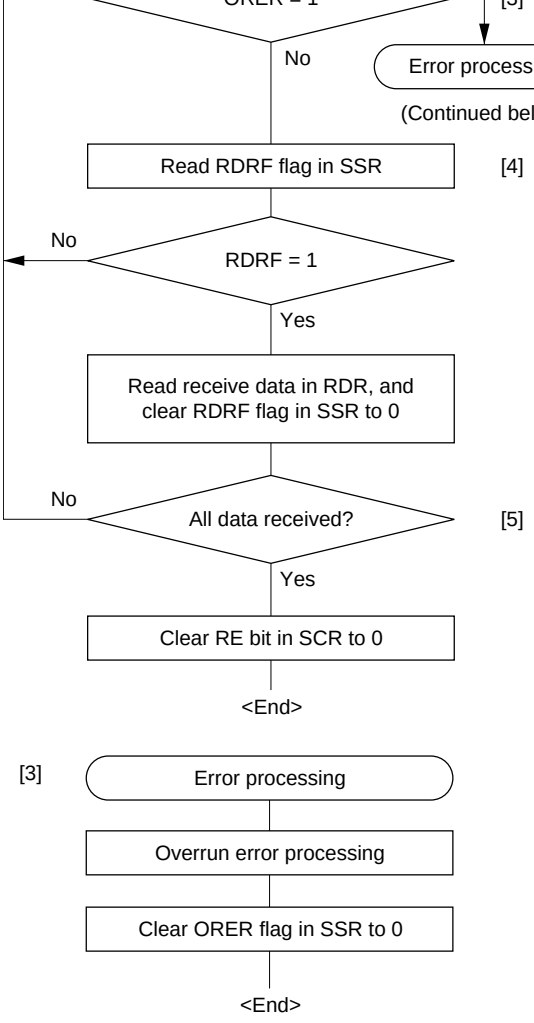
Figure 14.17 Example of SCI Operation in Transmission

Serial data reception (clocked synchronous mode): Figure 14.18 shows a sample flo
serial reception.

The following procedure should be used for serial data reception.

When changing the operating mode from asynchronous to clocked synchronous, be sur
that the ORER, PER, and FER flags are all cleared to 0.

The RDRF flag will not be set if the FER or PER flag is set to 1, and neither transmit n
operations will be possible.



[4] SCI status check and re
data read:
Read SSR and check th
RDRF flag is set to 1, th
the receive data in RDR
clear the RDRF flag to 0
Transition of the RDRF
0 to 1 can also be identi
an RXI interrupt.

[5] Serial reception continu
procedure:
To continue serial recep
before the MSB (bit 7) o
current frame is receive
reading the RDRF flag,
RDR, and clearing the R
to 0. The RDRF flag is c
automatically when the
DTC is activated by a re
data full interrupt (RXI) r
and the RDR value is re

Figure 14.18 Sample Serial Reception Flowchart

receive error is detected in the error check, the operation is as shown in table 14.11.

Neither transmit nor receive operations can be performed subsequently when a rece has been found in the error check.

[3] If the RIE bit in SCR is set to 1 when the RDRF flag changes to 1, a receive data fu (RXI) request is generated.

Also, if the RIE bit in SCR is set to 1 when the ORER flag changes to 1, a receive e interrupt (ERI) request is generated.

Figure 14.19 shows an example of SCI operation in reception.

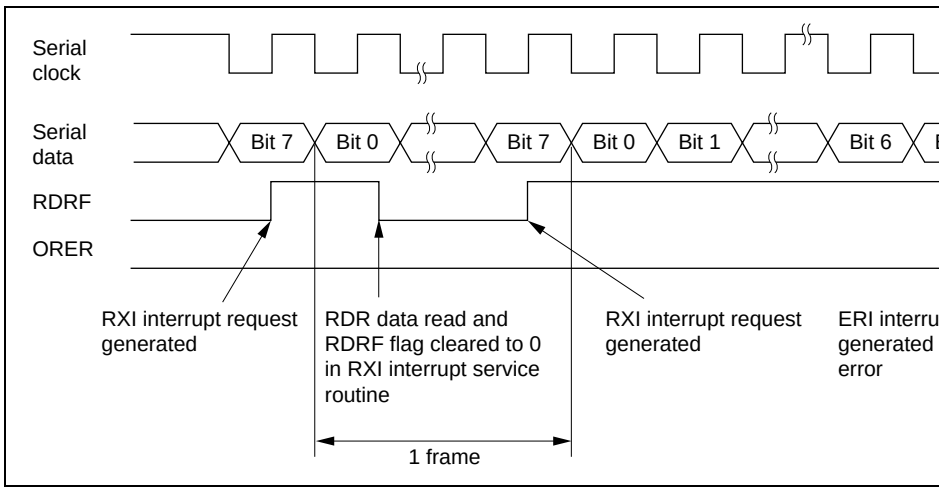
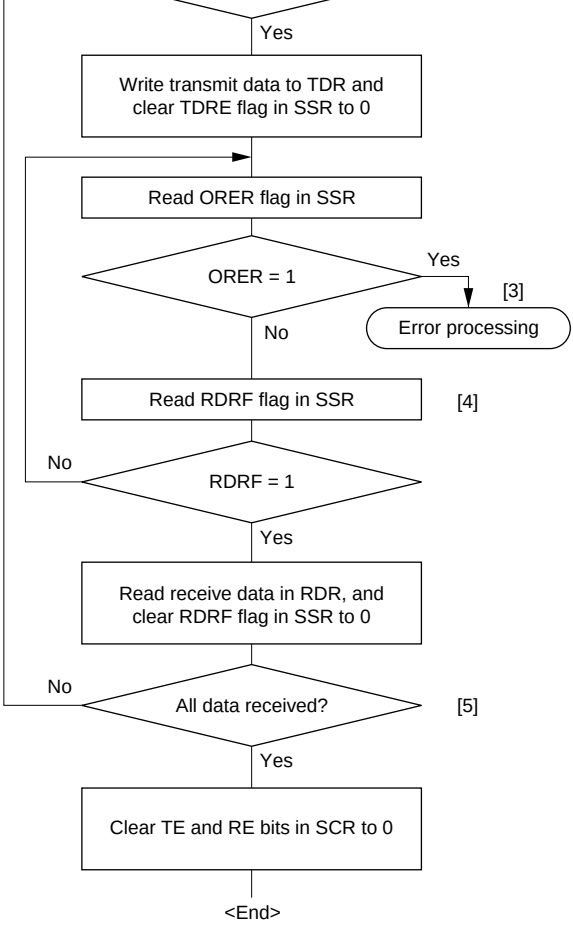


Figure 14.19 Example of SCI Operation in Reception

Simultaneous serial data transmission and reception (clocked synchronous mode): 14.20 shows a sample flowchart for simultaneous serial transmit and receive operations

The following procedure should be used for simultaneous serial data transmit and recei operations.



Note: When switching from transmit or receive operation to simultaneous transmit and receive operations, first clear the TE and RE bits to 0, then set both these bits to 1 simultaneously.

[3] Receive error processing:
If a receive error occurs, re
ORER flag in SSR , and af
performing the appropriate
processing, clear the ORE
0. Transmission/reception
resumed if the ORER flag
1.

[4] SCI status check and rece
read:
Read SSR and check that
RDRF flag is set to 1, then
receive data in RDR and c
RDRF flag to 0. Transition
RDRF flag from 0 to 1 can
identified by an RXI interru

[5] Serial transmission/recepti
continuation procedure:
To continue serial transmis
reception, before the MSB
the current frame is receiv
reading the RDRF flag, rea
RDR, and clearing the RDI
0. Also, before the MSB (b
the current frame is transm
read 1 from the TDRE flag
confirm that writing is poss
Then write data to TDR an
the TDRE flag to 0.
Checking and clearing of th
flag is automatic when the
DTC is activated by a trans
empty interrupt (TXI) reque
data is written to TDR. Also
RDRF flag is cleared autom
when the DMAC or DTC is
by a receive data full interr
request and the RDR value

Figure 14.20 Sample Flowchart of Simultaneous Serial Transmit and Receive

in SSR is set to 1, a TEI interrupt request is generated. A TXI interrupt can activate the DTC to perform data transfer. The TDRE flag is cleared to 0 automatically when data t performed by the DMAC or DTC. The DMAC and DTC cannot be activated by a TEI i request.

When the RDRF flag in SSR is set to 1, an RXI interrupt request is generated. When th PER, or FER flag in SSR is set to 1, an ERI interrupt request is generated. An RXI inte activate the DMAC or DTC to perform data transfer. The RDRF flag is cleared to 0 aut when data transfer is performed by the DMAC or DTC. The DMAC and DTC cannot b by an ERI interrupt request.

Also note that the DMAC cannot be activated by an SCI channel 2 interrupt.

		state (TDRE)		
	TEI	Interrupt due to transmission end (TEND)	Not possible	Not possible
1	ERI	Interrupt due to receive error (ORER, FER, or PER)	Not possible	Not possible
	RXI	Interrupt due to receive data full state (RDRF)	Possible	Possible
	TXI	Interrupt due to transmit data empty state (TDRE)	Possible	Possible
	TEI	Interrupt due to transmission end (TEND)	Not possible	Not possible
2	ERI	Interrupt due to receive error (ORER, FER, or PER)	Not possible	Not possible
	RXI	Interrupt due to receive data full state (RDRF)	Possible	Not possible
	TXI	Interrupt due to transmit data empty state (TDRE)	Possible	Not possible
	TEI	Interrupt due to transmission end (TEND)	Not possible	Not possible

Note: * This table shows the initial state immediately after a reset. Relative prioritie channels can be changed by means of the interrupt controller.

A TEI interrupt is requested when the TEND flag is set to 1 while the TEIE bit is set t TEND flag is cleared at the same time as the TDRE flag. Consequently, if a TEI inter TXI interrupt are requested simultaneously, the TXI interrupt may be accepted first, v result that the TDRE and TEND flags are cleared. Note that the TEI interrupt will not in this case.

Data can be written to TDR regardless of the state of the TDRE flag. However, if new d written to TDR when the TDRE flag is cleared to 0, the data stored in TDR will be lost not yet been transferred to TSR. It is therefore essential to check that the TDRE flag is before writing transmit data to TDR.

Operation when Multiple Receive Errors Occur Simultaneously

If a number of receive errors occur at the same time, the state of the status flags in SSR shown in table 14.13. If there is an overrun error, data is not transferred from RSR to R the receive data is lost.

Table 14.13 State of SSR Status Flags and Transfer of Receive Data

SSR Status Flags				Receive Data Transfer	
RDRF	**ORER**	**FER**	**PER**	**RSR to RDR**	**Receive Error Statu**
1	1	0	0	X	Overrun error
0	0	1	0	○	Framing error
0	0	0	1	○	Parity error
1	1	1	0	X	Overrun error + frami
1	1	0	1	X	Overrun error + parity
0	0	1	1	○	Framing error + parity
1	1	1	1	X	Overrun error + frami parity error

Legend:

○: Receive data is transferred from RSR to RDR.

X: Receive data is not transferred from RSR to RDR.

Sending a Break (Asynchronous Mode Only)

The TxD pin has a dual function as an I/O port whose direction (input or output) is de
DR and DDR. This can be used to send a break.

Between serial transmission initialization and setting of the TE bit to 1, the mark state
by the value of DR (the pin does not function as the TxD pin until the TE bit is set to 1
Consequently, DDR and DR for the port corresponding to the TxD pin are first set to 1

To send a break during serial transmission, first clear DR to 0, then clear the TE bit to

When the TE bit is cleared to 0, the transmitter is initialized regardless of the current t
state, the TxD pin becomes an I/O port, and 0 is output from the TxD pin.

Receive Error Flags and Transmit Operations (Clocked Synchronous Mode Only

Transmission cannot be started when a receive error flag (ORER, PER, or FER) is set
the TDRE flag is cleared to 0. Be sure to clear the receive error flags to 0 before starti
transmission.

Note also that receive error flags cannot be cleared to 0 even if the RE bit is cleared to

Receive Data Sampling Timing and Reception Margin in Asynchronous Mode

In asynchronous mode, the SCI operates on a basic clock with a frequency of 16 times
rate.

In reception, the SCI samples the falling edge of the start bit using the basic clock, and
internal synchronization. Receive data is latched internally at the rising edge of the 8th
basic clock. This is illustrated in figure 14.21.

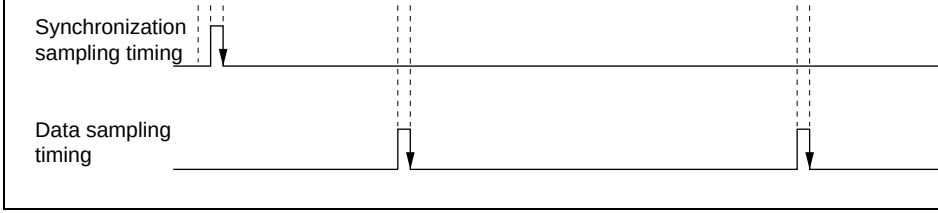


Figure 14.21 Receive Data Sampling Timing in Asynchronous Mode

Thus the reception margin in asynchronous mode is given by formula (1) below.

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

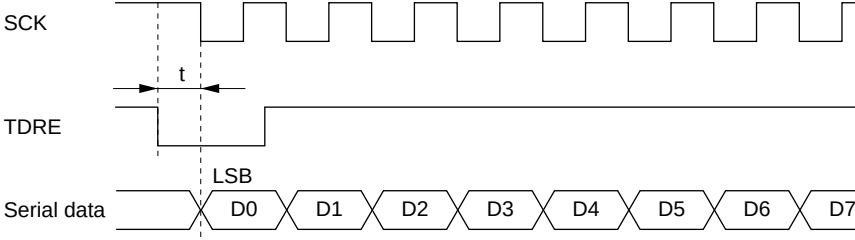
Where M: Reception margin (%)
N: Ratio of bit rate to clock (N = 16)
D: Clock duty (D = 0 to 1.0)
L: Frame length (L = 9 to 12)
F: Absolute value of clock rate deviation

Assuming values of F = 0 and D = 0.5 in formula (1), a reception margin of 46.875% is formula (2) below.

When D = 0.5 and F = 0,

$$M = (0.5 - \frac{1}{2 \times 16}) \times 100\%$$
$$= 46.875\% \quad \ldots\ldots\ldots \text{Form}$$

However, this is only the computed value, and a margin of 20% to 30% should be allow system design.



Note: When operating on an external clock, set t >4 clocks.

Figure 14.22 Example of Clocked Synchronous Transmission by DT

Switching between the normal serial communication interface and the Smart Card inte
carried out by means of a register setting.

15.1.1 Features

Features of the Smart Card interface supported by the H8S/2655 Group are as follows

- Asynchronous mode
  - Data length: 8 bits
  - Parity bit generation and checking
  - Transmission of error signal (parity error) in receive mode
  - Error signal detection and automatic data retransmission in transmit mode
  - Direct convention and inverse convention both supported
- On-chip baud rate generator allows any bit rate to be selected
- Three interrupt sources
  - Three interrupt sources (transmit data empty, receive data full, and transmit/rec that can issue requests independently
  - The transmit data empty interrupt and receive data full interrupt can activate th controller (DMAC) or data transfer controller (DTC) to execute data transfer

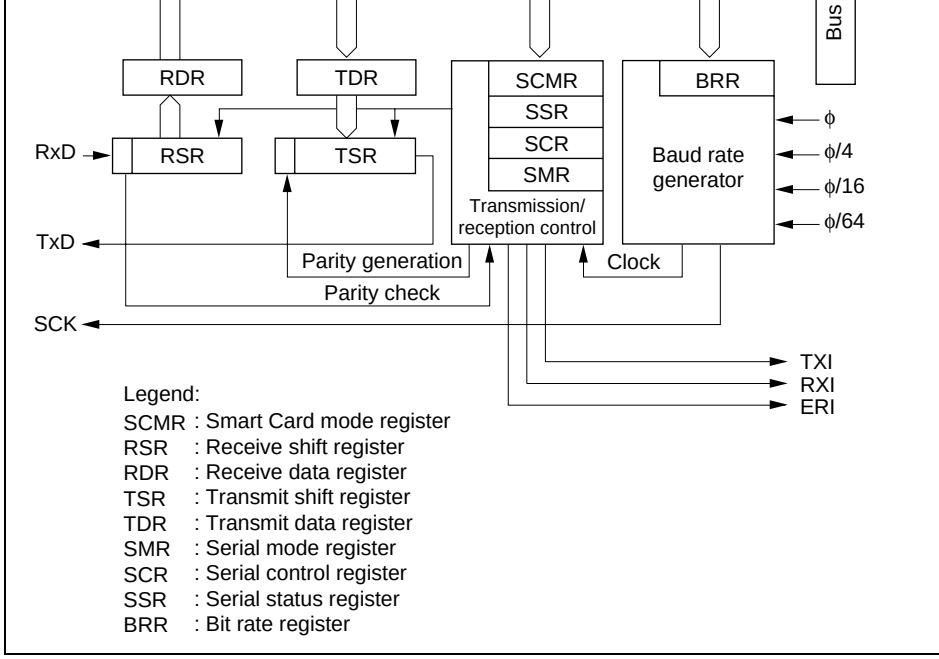


Figure 15.1 Block Diagram of Smart Card Interface

	Transmit data pin 0	TxD_0	Output	SCI0 transmit data outpu
1	Serial clock pin 1	SCK_1	I/O	SCI1 clock input/output
	Receive data pin 1	RxD_1	Input	SCI1 receive data input
	Transmit data pin 1	TxD_1	Output	SCI1 transmit data outpu
2	Serial clock pin 2	SCK_2	I/O	SCI2 clock input/output
	Receive data pin 2	RxD_2	Input	SCI2 receive data input
	Transmit data pin 2	TxD_2	Output	SCI2 transmit data outpu

0	Serial mode register 0	SMR0	R/W	H'00	H'
	Bit rate register 0	BRR0	R/W	H'FF	H'
	Serial control register 0	SCR0	R/W	H'00	H'
	Transmit data register 0	TDR0	R/W	H'FF	H'
	Serial status register 0	SSR0	R/(W)*2	H'84	H'
	Receive data register 0	RDR0	R	H'00	H'
	Smart card mode register 0	SCMR0	R/W	H'F2	H'
1	Serial mode register 1	SMR1	R/W	H'00	H'
	Bit rate register 1	BRR1	R/W	H'FF	H'
	Serial control register 1	SCR1	R/W	H'00	H'
	Transmit data register 1	TDR1	R/W	H'FF	H'
	Serial status register 1	SSR1	R/(W)*2	H'84	H'
	Receive data register 1	RDR1	R	H'00	H'
	Smart card mode register 1	SCMR1	R/W	H'F2	H'
2	Serial mode register 2	SMR2	R/W	H'00	H'
	Bit rate register 2	BRR2	R/W	H'FF	H'
	Serial control register 2	SCR2	R/W	H'00	H'
	Transmit data register 2	TDR2	R/W	H'FF	H'
	Serial status register 2	SSR2	R/(W)*2	H'84	H'
	Receive data register 2	RDR2	R	H'00	H'
	Smart card mode register 2	SCMR2	R/W	H'F2	H'
All	Module stop control register	MSTPCR	R/W	H'3FFF	H'

Notes: 1. Lower 16 bits of the address.
2. Can only be written with 0 for flag clearing.

	—	—	—	—	SDIR	SINV	—
Initial value :	1	1	1	1	0	0	1
R/W :	—	—	—	—	R/W	R/W	—

SCMR is an 8-bit readable/writable register that selects the Smart Card interface funct

SCMR is initialized to H'F2 by a reset, and in standby mode or module stop mode.

Bits 7 to 4—Reserved: Read-only bits, always read as 1.

Bit 3—Smart Card Data Transfer Direction (SDIR): Selects the serial/parallel con format.

Bit 3

SDIR	Description
0	TDR contents are transmitted LSB-first
	Receive data is stored in RDR LSB-first
1	TDR contents are transmitted MSB-first
	Receive data is stored in RDR MSB-first

	Receive data is stored as it is in RDR
1	TDR contents are inverted before being transmitted Receive data is stored in inverted form in RDR

Bit 1—Reserved: Read-only bit, always read as 1.

Bit 0—Smart Card Interface Mode Select (SMIF): Enables or disables the Smart Ca function.

Bit 0	
SMIF	**Description**
0	Smart Card interface function is disabled
1	Smart Card interface function is enabled

15.2.2 Serial Status Register (SSR)

Bit	:	7	6	5	4	3	2	1
		TDRE	RDRF	ORER	ERS	PER	TEND	MPB
Initial value	:	1	0	0	0	0	1	0
R/W	:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R

Note: * Only 0 can be written to bits 7 to 3, to clear these flags.

Bit 4 of SSR has a different function in Smart Card interface mode. Coupled with this, conditions for bit 2, TEND, are also different.

Bits 7 to 5—Operate in the same way as for the normal SCI. For details, see section 14 Status Register (SSR).

	• When 0 is written to ERS after reading ERS = 1
1	[Setting condition] When the low level of the error signal is sampled

Note: Clearing the TE bit in SCR to 0 does not affect the ERS flag, which retains its p state.

Bits 3 to 0—Operate in the same way as for the normal SCI. For details, see section 1 Status Register (SSR).

However, the setting conditions for the TEND bit, are as shown below.

Bit 2

TEND	Description
0	[Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When data is written to TDR by the DMAC or DTC
1	[Setting conditions] • Upon reset, and in standby mode or module stop mode • When the TE bit in SCR is 0 and the ERS bit is also 0 • When TDRE = 1 and ERS = 0 (normal transmission) 2.5 etu after trans 1-byte serial character

Note: etu: Elementary Time Unit (time for transfer of 1 bit)

Note: * When the smart card interface is used, be sure to make the 0 or 1 setting sho
bits 6, 5, 3, and 2.

The function of bit 7 of SMR changes in smart card interface mode.

Bit 7—GSM Mode (GM): Sets the smart card interface function to GSM mode.

This bit is cleared to 0 when the normal smart card interface is used. In GSM mode, thi
to 1, the timing of setting of the TEND flag that indicates transmission completion is ad
and clock output control mode addition is performed. The contents of the clock output
mode addition are specified by bits 1 and 0 of the serial control register (SCR).

Bit 7

GM	Description
0	Normal smart card interface mode operation (In • TEND flag generation 12.5 etu after beginning of start bit • Clock output ON/OFF control only
1	GSM mode smart card interface mode operation • TEND flag generation 11.0 etu after beginning of start bit • High/low fixing control possible in addition to clock output ON/OFF cont SCR)

Note: etu: Elementary time unit (time for transfer of 1 bit)

Bits 6 to 0—Operate in the same way as for the normal SCI. For details, see section 14
Mode Register (SMR).

mode register (SMR) is set to 1.

Bits 7 to 2—Operate in the same way as for the normal SCI. For details, see section 1 Control Register (SCR).

Bits 1 and 0—Clock Enable 1 and 0 (CKE1, CKE0): These bits are used to select th source and enable or disable clock output from the SCK pin.

In smart card interface mode, in addition to the normal switching between clock outpu and disabling, the clock output can be specified as to be fixed high or low.

SCMR	SMR	SCR Setting		SCK Pin Function
SMIF	C/$\overline{A}$, GM	CKE1	CKE0	
0	See the SCI			
1	0	0	0	Operates as port I/O pin
1	0	0	1	Outputs clock as SCK output pin
1	1	0	0	Operates as SCK output pin, with output
1	1	0	1	Outputs clock as SCK output pin
1	1	1	0	Operates as SCK output pin, with output
1	1	1	1	Outputs clock as SCK output pin

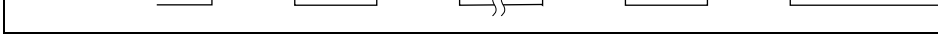


Figure 15.2 Clock Output Waveform Control

15.3 Operation

15.3.1 Overview

The main functions of the Smart Card interface are as follows.

- One frame consists of 8-bit data plus a parity bit.
- In transmission, a guard time of at least 2 etu (Elementary Time Unit: the time for t one bit) is left between the end of the parity bit and the start of the next frame.
- If a parity error is detected during reception, a low error signal level is output for or period, 10.5 etu after the start bit.
- If the error signal is sampled during transmission, the same data is transmitted auto after the elapse of 2 etu or longer.
- Only asynchronous communication is supported; there is no clocked synchronous communication function.

input to the CLK pin of the IC card. No connection is needed if the IC card uses an int

LSI port output is used as the reset signal.

Other pins must normally be connected to the power supply or ground.

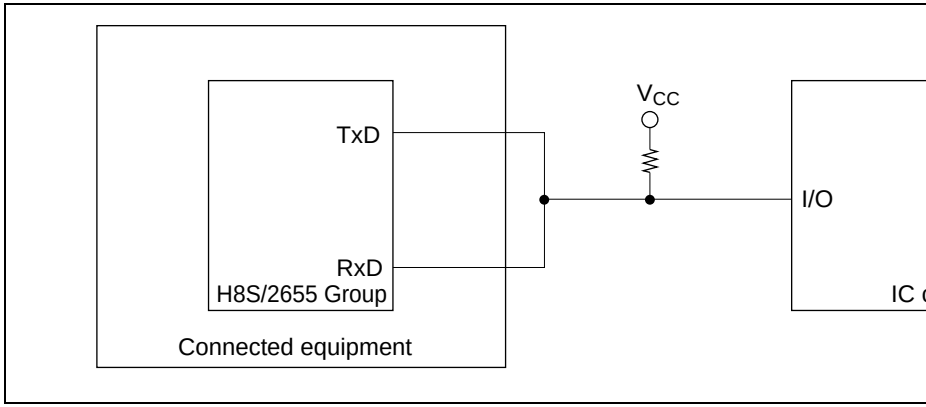


Figure 15.3 Schematic Diagram of Smart Card Interface Pin Connecti

Note: If an IC card is not connected, and the TE and RE bits are both set to 1, closed transmission/reception is possible, enabling self-diagnosis to be carried out.

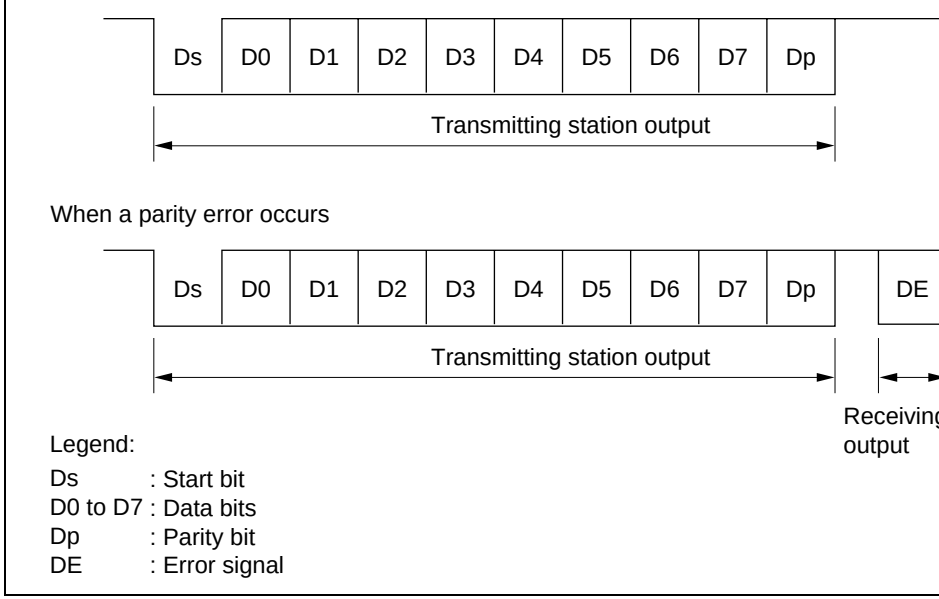


Figure 15.4 Smart Card Interface Data Format

The operation sequence is as follows.

[1] When the data line is not in use it is in the high-impedance state, and is fixed high v
up resistor.

[2] The transmitting station starts transfer of one frame of data. The data frame starts w
bit (Ds, low-level), followed by 8 data bits (D0 to D7) and a parity bit (Dp).

[3] With the Smart Card interface, the data line then returns to the high-impedance stat
line is pulled high with a pull-up resistor.

[5] If the transmitting station does not receive an error signal, it proceeds to transmit t
frame.

If it does receive an error signal, however, it returns to step [2] and retransmits the
data.

15.3.4 Register Settings

Table 15.3 shows a bit map of the registers used by the smart card interface.

Bits indicated as 0 or 1 must be set to the value shown. The setting of other bits is des
below.

Table 15.3 Smart Card Interface Register Settings

	Bit						
Register	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**
SMR	GM	0	1	O/$\overline{E}$	1	0	CKS1
BRR	BRR7	BRR6	BRR5	BRR4	BRR3	BRR2	BRR1
SCR	TIE	RIE	TE	RE	0	0	CKE1
TDR	TDR7	TDR6	TDR5	TDR4	TDR3	TDR2	TDR1
SSR	TDRE	RDRF	ORER	ERS	PER	TEND	0
RDR	RDR7	RDR6	RDR5	RDR4	RDR3	RDR2	RDR1
SCMR	—	—	—	—	SDIR	SINV	—

Legend:

—: Unused bit.

BRR Setting

BRR is used to set the bit rate. See section 15.3.5, Clock, for the method of calculating
to be set.

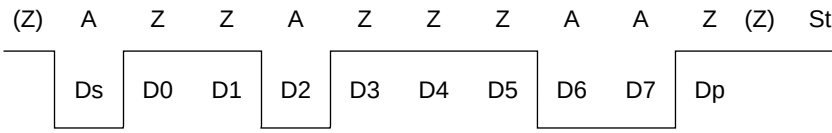
SCR Setting

The function of the TIE, RIE, TE, and RE bits is the same as for the normal SCI. For d
section 14, Serial Communication Interface (SCI).

Bits CKE1 and CKE0 specify the clock output. When the GM bit in SMR is cleared to
bits to B'00 if a clock is not to be output, or to B'01 if a clock is to be output. When the
SMR is set to 1, clock output is performed. The clock output can also be fixed high or l

Examples of register settings and the waveform of the start character are shown below types of IC card (direct convention and inverse convention).

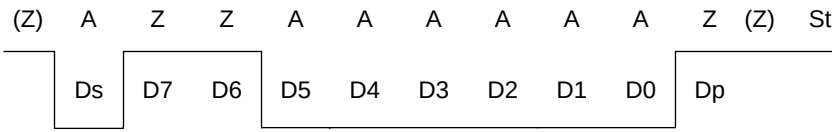
- Direct convention (SDIR = SINV = O/$\overline{E}$ = 0)



With the direct convention type, the logic 1 level corresponds to state Z and the lo state A, and transfer is performed in LSB-first order. The start character data abov

The parity bit is 1 since even parity is stipulated for the Smart Card.

- Inverse convention (SDIR = SINV = O/$\overline{E}$ = 1)



With the inverse convention type, the logic 1 level corresponds to state A and the l to state Z, and transfer is performed in MSB-first order. The start character data ab

The parity bit is 0, corresponding to state Z, since even parity is stipulated for the S

With the H8S/2655 Group, inversion specified by the SINV bit applies only to the D7 to D0. For parity bit inversion, the O/$\overline{E}$ bit in SMR is set to odd parity mode (th applies to both transmission and reception).

$$B = \frac{\phi}{1488 \times 2^{2n-1} \times (N + 1)} \times 10^6$$

Where: N = Value set in BRR ($0 \leq N \leq 255$)
B = Bit rate (bit/s)
ϕ = Operating frequency (MHz)
n = See table 15.4

Table 15.4 Correspondence between n and CKS1, CKS0

n	CKS1	CKS0
0	0	0
1		1
2	1	0
3		1

Table 15.5 Examples of Bit Rate B (bit/s) for Various BRR Settings (When n = 0

	ϕ (MHz)					
N	**10.00**	**10.714**	**13.00**	**14.285**	**16.00**	**1**
0	13441	14400	17473	19200	21505	2
1	6720	7200	8737	9600	10753	1
2	4480	4800	5824	6400	7168	8

Note: Bit rates are rounded to the nearest whole number.

	φ (MHz)											
	7.1424		10.00		10.7136		13.00		14.2848		16.00	
bit/s	N	Error	N	Error	N	Error	N	Error	N	Error	N	Error
9600	0	0.00	1	30	1	25	1	8.99	1	0.00	1	12.01

Table 15.7 Maximum Bit Rate at Various Frequencies (Smart Card Interface M

φ (MHz)	Maximum Bit Rate (bit/s)	N	n
7.1424	9600	0	0
10.00	13441	0	0
10.7136	14400	0	0
13.00	17473	0	0
14.2848	19200	0	0
16.00	21505	0	0
18.00	24194	0	0
20.00	26882	0	0

The bit rate error is given by the following formula:

$$\text{Error (\%)} = \left(\frac{\phi}{1488 \times 2^{2n-1} \times B \times (N + 1)} \times 10^6 - 1 \right) \times 100$$

[2] Clear the error flags ERS, PER, and ORER in SSR to 0.

[3] Set the O/$\overline{E}$ bit and CKS1 and CKS0 bits in SMR. Clear the C/$\overline{A}$, CHR, and MP bit
set the STOP and PE bits to 1.

[4] Set the SMIF, SDIR, and SINV bits in SCMR.
When the SMIF bit is set to 1, the TxD and RxD pins are both switched from ports
and are placed in the high-impedance state.

[5] Set the value corresponding to the bit rate in BRR.

[6] Set the CKE0 bit in SCR. Clear the TIE, RIE, TE, RE, MPIE, TEIE and CKE1 bits
If the CKE0 bit is set to 1, the clock is output from the SCK pin.

[7] Wait at least one bit interval, then set the TIE, RIE, TE, and RE bits in SCR. Do no
bit and RE bit at the same time, except for self-diagnosis.

[3] Repeat steps [2] and [3] until it can be confirmed that the TEND flag in SSR is set

[4] Write the transmit data to TDR, clear the TDRE flag to 0, and perform the transmi
The TEND flag is cleared to 0.

[5] When transmitting data continuously, go back to step [2].

[6] To end transmission, clear the TE bit to 0.

With the above processing, interrupt servicing or data transfer by the DMAC or DTC

If transmission ends and the TEND flag is set to 1 while the TIE bit is set to 1 and inte
requests are enabled, a transmit data empty interrupt (TXI) request will be generated.
occurs in transmission and the ERS flag is set to 1 while the RIE bit is set to 1 and int
requests are enabled, a transfer error interrupt (ERI) request will be generated.

The timing for setting the TEND flag depends on the value of the GM bit in SMR. Th
set timing is shown in figure 15.5.

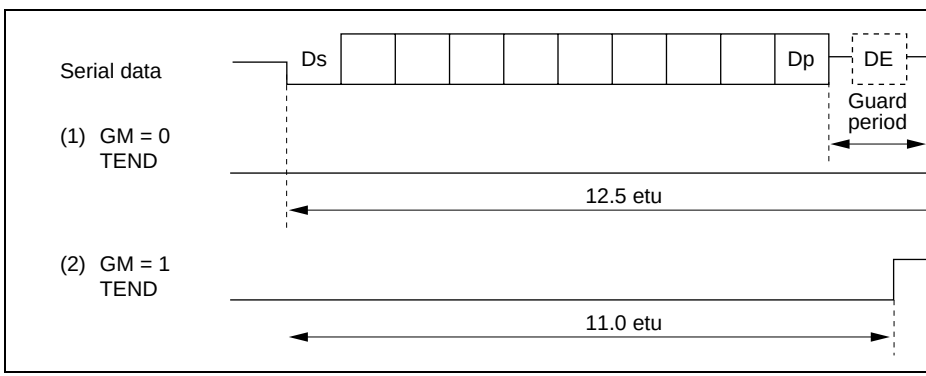


Figure 15.5 TEND Flag Set Timing

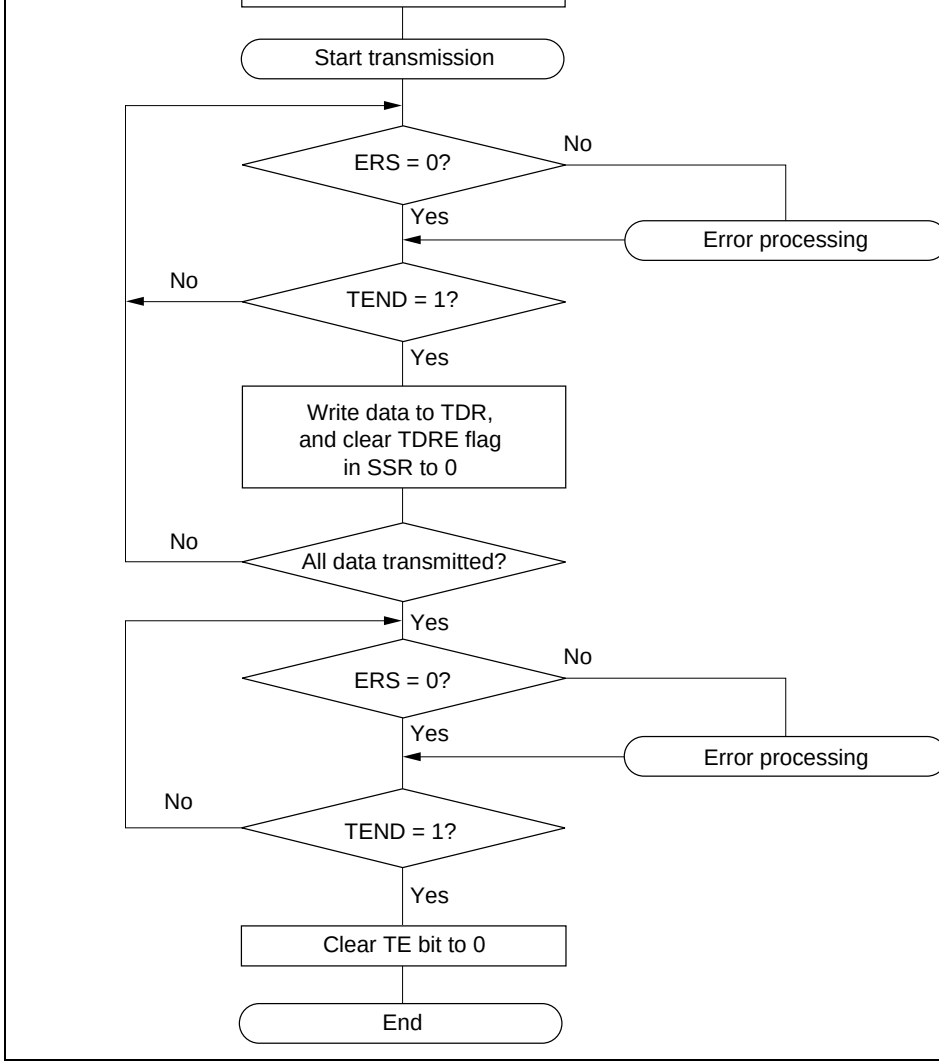


Figure 15.6 Example of Transmission Processing Flow

[3] Repeat steps [2] and [3] until it can be confirmed that the RDRF flag is set to 1.

[4] Read the receive data from RDR.

[5] When receiving data continuously, clear the RDRF flag to 0 and go back to step [2

[6] To end reception, clear the RE bit to 0.

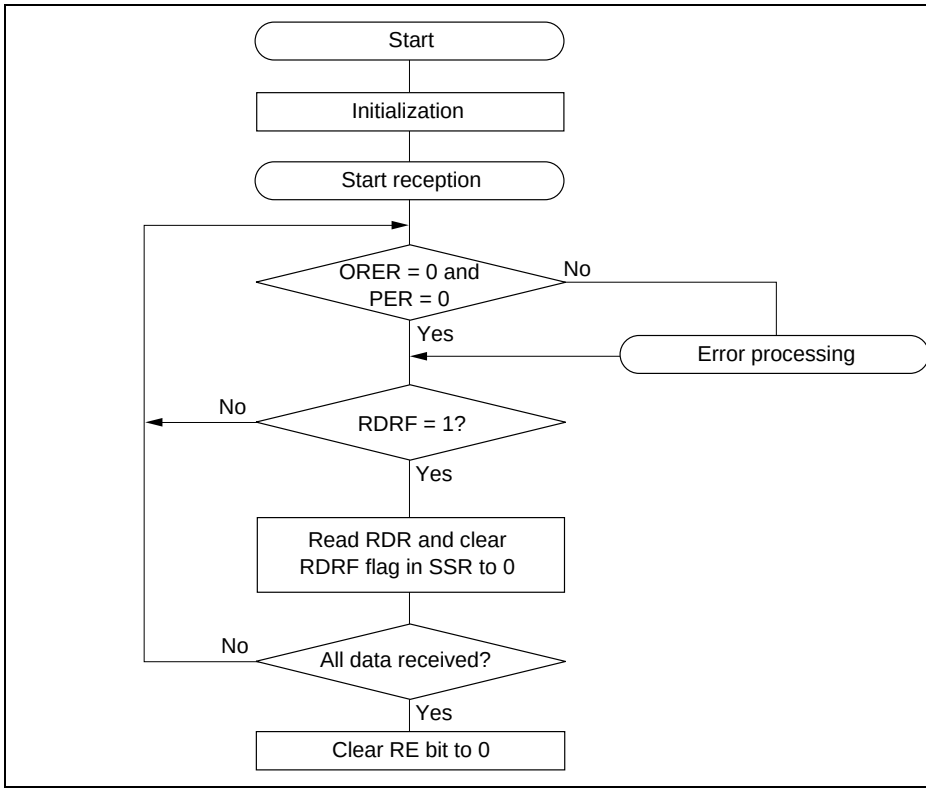


Figure 15.7 Example of Reception Processing Flow

For details, see Interrupt Operation and Data Transfer Operation by DMAC or DTC be

If a parity error occurs during reception and the PER is set to 1, the received data is stil transferred to RDR, and therefore this data can be read.

Mode Switching Operation

When switching from receive mode to transmit mode, first confirm that the receive ope been completed, then start from initialization, clearing RE bit to 0 and setting TE bit to RDRF flag or the PER and ORER flags can be used to check that the receive operation completed.

When switching from transmit mode to receive mode, first confirm that the transmit op been completed, then start from initialization, clearing TE bit to 0 and setting RE bit to TEND flag can be used to check that the transmit operation has been completed.

Fixing Clock Output Level

When the GSM bit in SMR is set to 1, the clock output level can be fixed with bits CK CKE0 in SCR. At this time, the minimum clock pulse width can be made the specified

Figure 15.8 shows the timing for fixing the clock output level. In this example, GSM is CKE1 is cleared to 0, and the CKE0 bit is controlled.

Figure 15.8 Timing for Fixing Clock Output Level

Interrupt Operation

There are three interrupt sources in smart card interface mode: transmit data empty int
requests, transfer error interrupt (ERI) requests, and receive data full interrupt (RXI) r
transmit end interrupt (TEI) request is not used in this mode.

When the TEND flag in SSR is set to 1, a TXI interrupt request is generated.

When the RDRF flag in SSR is set to 1, an RXI interrupt request is generated.

When any of flags ORER, PER, and ERS in SSR is set to 1, an ERI interrupt request i
The relationship between the operating states and interrupt sources is shown in table 1

Table 15.8 Smart Card Mode Operating States and Interrupt Sources

Operating State		Flag	Enable Bit	Interrupt Source	DMAC Activation	D A
Transmit Mode	Normal operation	TEND	TIE	TXI	Possible	P
	Error	ERS	RIE	ERI	Not possible	N
Receive Mode	Normal operation	RDRF	RIE	RXI	Possible	P
	Error	PER, ORER	RIE	ERI	Not possible	N

same data automatically. The TEND flag remains cleared to 0 during this time, and the
not activated. Thus, the number of bytes specified by the SCI and DMAC are transmitte
automatically even in retransmission following an error. However, the ERS flag is not c
automatically when an error occurs, and so the RIE bit should be set to 1 beforehand so
ERI request will be generated in the event of an error, and the ERS flag will be cleared.

When performing transfer using the DMAC or DTC, it is essential to set and enable the
DTC before carrying out SCI setting. For details of the DMAC and DTC setting proced
section 7, DMA Controller, and section 8, Data Transfer Controller.

In a receive operation, an RXI interrupt request is generated when the RDRF flag in SS
1. If the RXI request is designated beforehand as a DMAC or DTC activation source, th
or DTC will be activated by the RXI request, and transfer of the receive data will be ca
The RDRF flag is cleared to 0 automatically when data transfer is performed by the DM
DTC. If an error occurs, an error flag is set but the RDRF flag is not. For this reason, th
or DTC is not activated, but instead, an ERI interrupt request is sent to the CPU. There
error flag should be cleared.

[1] Set DR and DDR of the I/O port corresponding to the serial clock to the value for t
output state in software standby mode.

[2] Write 0 to the TE bit and RE bit in SCR to halt transmit/receive operation. At the s
set the CKE1 bit to the value for the fixed output state in software standby mode.

[3] Write 0 to the CKE0 bit in SCR to halt the clock.

[4] Wait for one serial clock period. During this interval, serial clock output is fixed a
specified level, with the pulse width maintained.

[5] Write H'00 to SMR and SCMR.

[6] Make the transition to software standby mode.

(2) Exiting software standby mode

[7] Exit software standby mode by means of an external interrupt.

[8] Set the CKE1 bit in SCR to the value for the fixed output state (corresponding I/O
software standby mode.

[9] Set smart card interface mode and output the clock. The clock is output with the sp
pulse width.

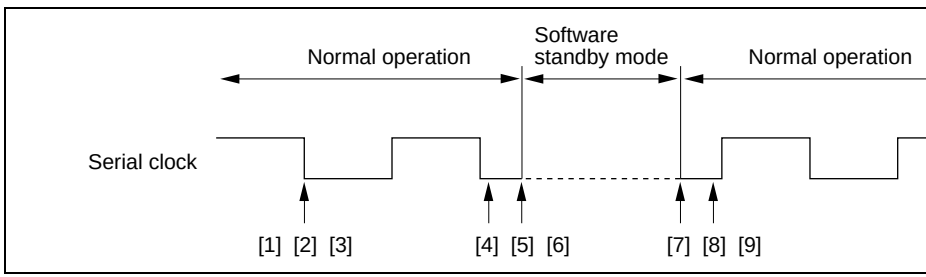


Figure 15.9 Entering and Exiting Software Standby Mode

[4] Set the CKE0 bit in SCR to 1 to start the serial clock output.

15.4 Usage Notes

The following points should be noted when using the SCI as a Smart Card interface.

Receive Data Sampling Timing and Reception Margin in Smart Card Interface M

In Smart Card Interface mode, the SCI operates on a basic clock with a frequency of 37 the transfer rate.

In reception, the SCI samples the falling edge of the start bit using the basic clock, and internal synchronization. Receive data is latched internally at the rising edge of the 186 the basic clock. This is illustrated in figure 15.10.

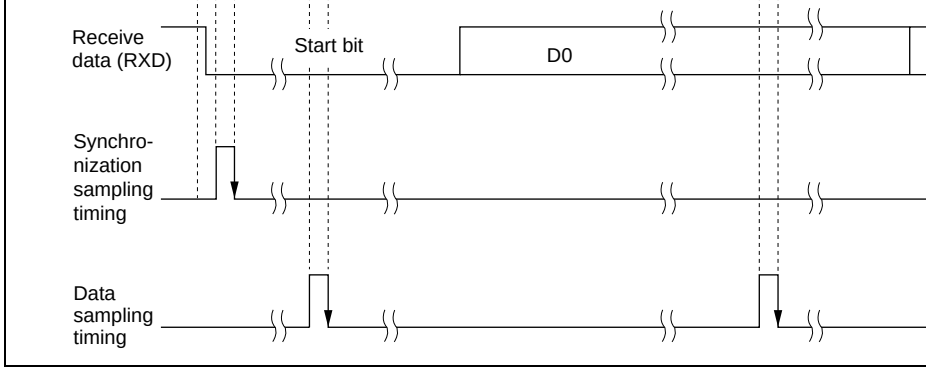


Figure 15.10 Receive Data Sampling Timing in Smart Card Mode

Thus the reception margin in asynchronous mode is given by the following formula.

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

Where M: Reception margin (%)
N: Ratio of bit rate to clock (N = 372)
D: Clock duty (D = 0 to 1.0)
L: Frame length (L = 10)
F: Absolute value of clock frequency deviation

Assuming values of F = 0 and D = 0.5 in the above formula, the reception margin forn
follows.

When D = 0.5 and F = 0,

$$M = (0.5 - 1/2 \times 372) \times 100\%$$
$$= 49.866\%$$

automatically set to 1. If the RIE bit in SCR is enabled at this time, an ERI interrupt
generated. The PER bit in SSR should be kept cleared to 0 until the next parity bit is

[2] The RDRF bit in SSR is not set for a frame in which an error has occurred.

[3] If no error is found when the received parity bit is checked, the PER bit in SSR is n

[4] If no error is found when the received parity bit is checked, the receive operation is
have been completed normally, and the RDRF flag in SSR is automatically set to 1.
bit in SCR is enabled at this time, an RXI interrupt request is generated.

If DMAC or DTC data transfer by an RXI source is enabled, the contents of RDR c
automatically. When the RDR data is read by the DMAC or DTC, the RDRF flag is
automatically cleared to 0.

[5] When a normal frame is received, the pin retains the high-impedance state at the tim
error signal transmission.

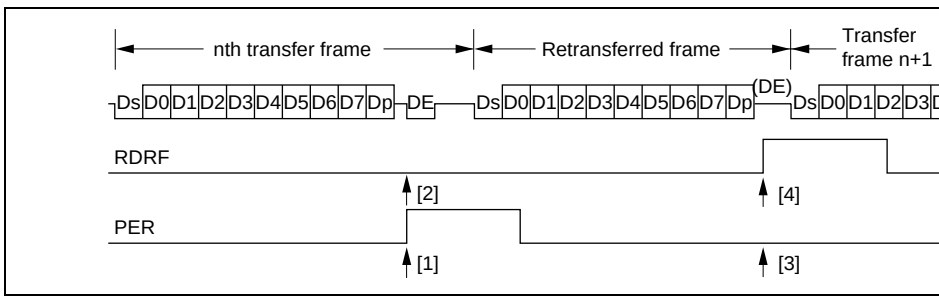


Figure 15.11 Retransfer Operation in SCI Receive Mode

is received.

[8] If an error signal is not sent back from the receiving end, the ERS bit in SSR is not

[9] If an error signal is not sent back from the receiving end, transmission of one fram a retransfer, is judged to have been completed, and the TEND bit in SSR is set to 1 bit in SCR is enabled at this time, a TXI interrupt request is generated.

If data transfer by the DMAC or DTC by means of the TXI source is enabled, the n be written to TDR automatically. When data is written to TDR by the DMAC or D TDRE bit is automatically cleared to 0.

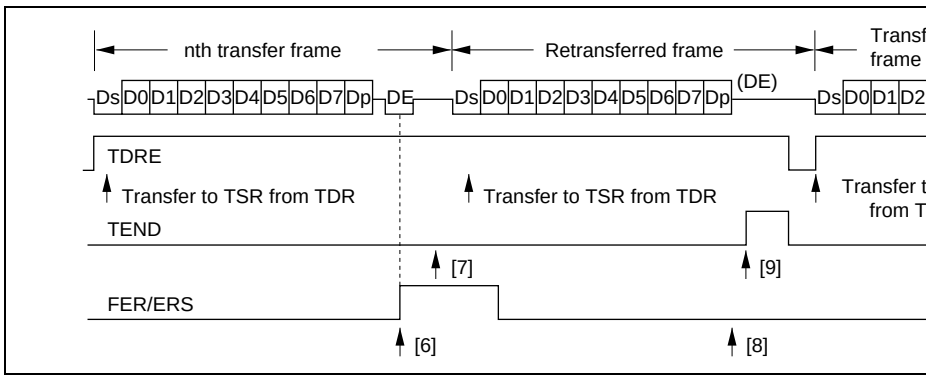
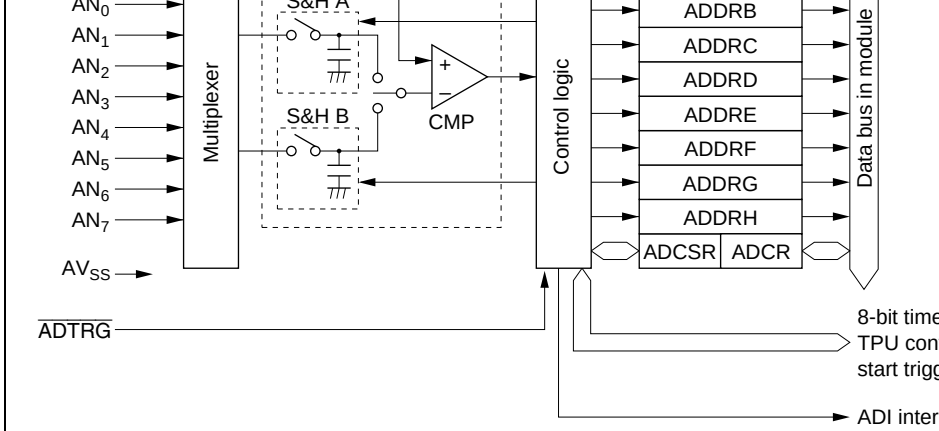


Figure 15.12 Retransfer Operation in SCI Transmit Mode

A/D converter features are listed below.

- 10-bit resolution
- Eight input channels
- Settable analog conversion voltage range
  - Conversion of analog voltages from 0 V to V_{ref}, with the reference voltage pin (analog reference voltage
- High-speed conversion
  - Minimum conversion time: 2.2 µs per channel (at 20-MHz operation)
    1.0 µs per channel in continuous conversion
- Variety of conversion modes
  - Choice of select mode or group mode
  - Choice of single mode or scan mode
  - Buffer operation possible
  - Simultaneous 2-channel sampling possible
- Three kinds of conversion start
  - Choice of software or timer conversion start trigger (TPU or 8-bit timer), or $\overline{AD}$
- Eight data registers
  - Conversion results are held in a 16-bit data register for each channel
- Sample and hold function
- A/D conversion end interrupt generation
  - A/D conversion end interrupt (ADI) request can be generated at the end of A/D



Legend:

ADCR : A/D control register	CMP : Comparator array
ADCSR : A/D control status register	S&H : Sample and hold circuit
ADDRA : A/D data register A	ADDRE : A/D data register E
ADDRB : A/D data register B	ADDRF : A/D data register F
ADDRC : A/D data register C	ADDRG : A/D data register G
ADDRD : A/D data register D	ADDRH : A/D data register H

Figure 16.1 Block Diagram of A/D Converter

16.1.3 Pin Configuration

Table 16.1 summarizes the input pins used by the A/D converter.

The AV_{CC} and AV_{SS} pins are the power supply pins for the analog block in the A/D conv
V_{ref} pin is the A/D conversion reference voltage pin.

Analog input 1	AN_1	Input	Analog input channel 1
Analog input 2	AN_2	Input	Analog input channel 2
Analog input 3	AN_3	Input	Analog input channel 3
Analog input 4	AN_4	Input	Analog input channel 4
Analog input 5	AN_5	Input	Analog input channel 5
Analog input 6	AN_6	Input	Analog input channel 6
Analog input 7	AN_7	Input	Analog input channel 7
A/D external trigger input	$\overline{\text{ADTRG}}$	Input	External trigger for starting A/D

16.1.4 Register Configuration

Table 16.2 summarizes the registers of the A/D converter.

Table 16.2 A/D Converter Registers

Name	Abbreviation	R/W	Initial Value	A
A/D data register A	ADDRA	R	H'0000	H
A/D data register B	ADDRB	R	H'0000	H
A/D data register C	ADDRC	R	H'0000	H
A/D data register D	ADDRD	R	H'0000	H
A/D data register E	ADDRE	R	H'0000	H
A/D data register F	ADDRF	R	H'0000	H
A/D data register G	ADDRG	R	H'0000	H
A/D data register H	ADDRH	R	H'0000	H
A/D control/status register	ADCSR	R/(W)*2	H'00	H
A/D control register	ADCR	R/W	H'00	H
Module stop control register	MSTPCR	R/W	H'3FFF	H

Notes: 1. Lower 16 bits of the address.
2. Bit 7 can only be written with 0 for flag clearing.

There are eight 16-bit read-only ADDR registers, ADDRA to ADDRH, used to store th
A/D conversion.

The 10-bit data resulting from A/D conversion is transferred to the ADDR register for t
on which conversion was performed, and stored there. The lower 8 bits of the converte
transferred to the lower byte (bits 7 to 0) of ADDR, and the upper 2 bits are transferred
upper byte (bits 9 and 8). Bits 15 to 10 are always read as 0.

Byte or word length can be selected for data reads. In a byte data read, the upper 8 bits
converted data are transferred. Buffer operation is also possible by using ADDRA to A
combination.

The correspondence between the analog input channels and ADDR registers is shown i
16.3.

The ADDR registers are initialized to H'0000 by a reset, and in hardware standby mode

Table 16.3 Analog Input Channels and Corresponding ADDR Registers

Analog Input Channel	A/D Data Register
AN_0	ADDRA*
AN_1	ADDRB*
AN_2	ADDRC*
AN_3	ADDRD*
AN_4	ADDRE
AN_5	ADDRF
AN_6	ADDRG
AN_7	ADDRH

Note: * Except when buffer operation is used.

Note: * Only 0 can be written to bit 7, to clear this flag.

ADCSR is an 8-bit readable/writable register that controls A/D conversion operations the status of the operation.

ADCSR is initialized to H'00 by a reset, and in hardware standby mode.

Bit 7—A/D End Flag (ADF): Status flag that indicates the end of A/D conversion.

Bit 7	
ADF	**Description**
0	[Clearing conditions] • When 0 is written to the ADF flag after reading ADF = 1 • When the DTC or DMAC is activated by an ADI interrupt and the presc register is read
1	[Setting conditions] • Single mode: When conversion ends for all specified channels, and A/ conversion ends* • Scan mode: When one round of conversion has been performed on al channels

Note: * In buffer operation, the ADF flag is not set until completion of the specified operation.

1	A/D conversion end interrupt (ADI) request enabled

Bit 5—A/D Start (ADST): Selects starting or stopping on A/D conversion. Holds a va
during A/D conversion.

The ADST bit can be set to 1 by software, a timer conversion start trigger, or the A/D e
trigger input pin ($\overline{\text{ADTRG}}$).

Bit 5

ADST	Description
0	A/D conversion stopped (In
1	• Single mode: A/D conversion is started. Cleared to 0 automatically whe conversion on the specified channel ends • Scan mode: A/D conversion is started. Conversion continues until ADS cleared to 0 by software

Bit 4—Clock Select (CKS): Sets the A/D conversion time. Set the CKS bit according
operating frequency so that the conversion time is at least 2 µs.

Only change the conversion time while conversion is stopped.

Bit 4

CKS	Description
0	Conversion time = 24 states (A/D converter reference clock = ϕ) (
1	Conversion time = 44 states (A/D converter reference clock = ϕ/2)

1	Group mode

Bits 2 to 0—Channel Select 2 to 0 (CH2 to CH0): Together with the GRP bit, these the analog input channel(s).

Only set the input channel while conversion is stopped.

Bit 2	Bit 1	Bit 0	Description	
CH2	**CH1**	**CH0**	**Select Mode (GRP = 0)**	**Group Mode (GRP =**
0	0	0	AN_0 (Initial value)	AN_0
		1	AN_1	AN_0 to AN_1
	1	0	AN_2	AN_0 to AN_2
		1	AN_3	AN_0 to AN_3
1	0	0	AN_4	AN_0 to AN_4
		1	AN_5	AN_0 to AN_5
	1	0	AN_6	AN_0 to AN_6
		1	AN_7	AN_0 to AN_7

16.2.3 A/D Control Register (ADCR)

Bit :	7	6	5	4	3	2	1
	—	PWR	TRGS1	TRGS0	SCAN	DSMP	BUFE1
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

ADCR is an 8-bit readable/writable register that controls A/D conversion operations.

ADCR is initialized to H'00 by a reset, and in hardware standby mode.

PWR	Description
0	Low-power conversion mode
1	High-speed start mode

Bits 5 and 4—Timer Trigger Select 1 and 0 (TRGS1, TRGS0): These bits select ena
disabling of A/D conversion start by a trigger signal.

Only set bits TRGS1 and TRGS0 while conversion is stopped.

Bit 5	Bit 4	
TRGS1	TRGS0	Description
0	0	A/D conversion start by software is enabled
	1	A/D conversion start by TPU conversion start trigger is enable
1	0	A/D conversion start by 8-bit timer conversion start trigger is e
	1	A/D conversion start by external trigger pin ($\overline{\text{ADTRG}}$) is enable

Bit 3—Scan Mode (SCAN): Selects single mode or scan mode as the A/D conversion
mode. For operation in single mode and scan mode, see section 16.4, Operation.

Only set the SCAN bit while conversion is stopped.

Bit 3	
SCAN	Description
0	Single mode
1	Scan mode

0	Normal sampling operation
1	Simultaneous sampling operation

Bits 1 and 0—Buffer Enable 1 and 0 (BUFE1, BUFE0): These bits specify whether registers ADDRB to ADDRD are to be used as buffer registers.

For setting and clearing of the ADF flag in the case of buffer operation, see section 16 Operation.

Only set the BUFE1 and BUFE0 bits while conversion is stopped.

Bit 1	Bit 0	
BUFE1	**BUFE0**	**Description**
0	0	Normal operation
	1	ADDRA and ADDRB are used for buffer operation (conversio result → ADDRA → ADDRB) (ADDRB is the buffer register)
1	0	ADDRA and ADDRC, and ADDRB and ADDRD, are used for operation (conversion result 1 → ADDRA → ADDRC; convers result 2 → ADDRB → ADDRD) (ADDRC and ADDRD are the buffer registers)
	1	ADDRA to ADDRD are used for buffer operation (conversion result → ADDRA → ADDRB → ADDRC → ADDR (ADDRB to ADDRD are the buffer registers)

MSTPCR is a 16-bit readable/writable register that performs module stop mode control

When the MSTP9 bit in MSTPCR is set to 1, A/D converter operation stops at the end cycle and a transition is made to module stop mode. Registers cannot be read or written module stop mode. For details, see section 21.5, Module Stop Mode.

MSTPCR is initialized to H'3FFF by a reset and in hardware standby mode. It is not ini software standby mode.

Bit 9—Module Stop (MSTP9): Specifies the A/D converter module stop mode.

Bit 9	
MSTP9	**Description**
0	A/D converter module stop mode cleared
1	A/D converter module stop mode set (I

16.3 Interface to Bus Master

ADDRA to ADDRH are 16-bit registers, and the data bus to the bus master is 16 bits w bus master can perform either word-size or byte-size reads on ADDRA to ADDRH.

In a word-size read of an ADDR register, all 16 bits of the ADDR contents are transfer bus master in one go. In a byte-size read of the upper byte only, the contents of the upp (AD9 to AD2) of the transferred data (AD9 to AD0) are transferred to the bus master.

Figure 16.2 illustrates the operation when reading an ADDR register.

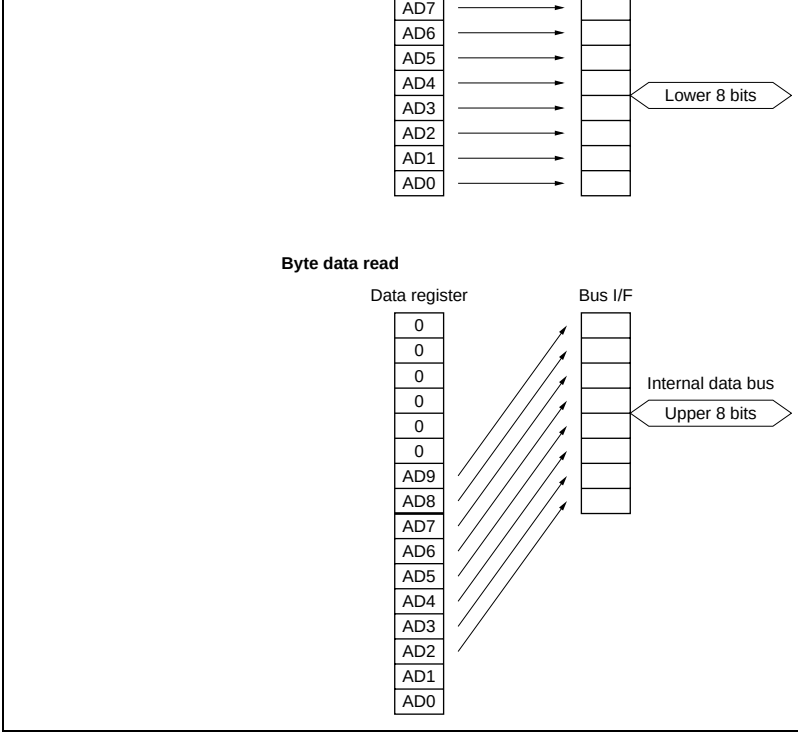


Figure 16.2 ADDR Read Operation

single activation results in conversion repeated until stopped by software. In buffer ope
when conversion ends for the channel concerned, the previous conversion results are s
buffer register. In simultaneous sampling operation, analog input voltages are sampled
channels simultaneously, and converted sequentially.

A software or timer conversion start trigger (TPU or 8-bit timer), or $\overline{\text{ADTRG}}$ input, can
selected as the conversion start condition.

Either high-speed start mode or low-power conversion mode can be selected for A/D c
by means of the PWR bit.

The operating mode or input channel can be changed by rewriting ADCSR and ADCR
ADST bit is cleared to 0. After ADCSR and ADCR have been rewritten, A/D conversio
again when the ADST bit is set to 1. A change of operating mode or input channel and
setting can be carried out simultaneously. A/D conversion can be stopped midway by c
ADST bit to 0.

ADI interrupt request is generated. The ADF flag is cleared by writing 0 after reading can be cleared with the BCLR instruction.)

Figure 16.3 shows an example of A/D converter operation when AN_1 is selected in sel mode.

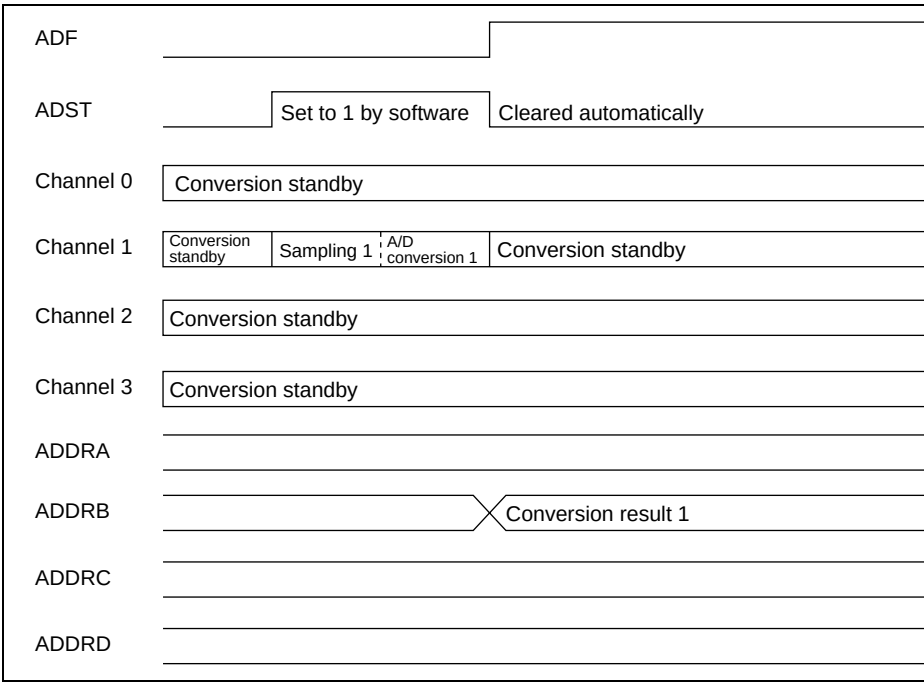


Figure 16.3 Example of A/D Converter Operation (Select Single Mod

When the first conversion operation ends, the ADF flag is set to 1. If the ADIE bit is se
this time, an ADI interrupt request is generated and A/D conversion is temporarily halt
the ADF flag to 0 when conversion has been halted by an ADI interrupt request will res
conversion. The ADF flag is cleared by writing 0 after reading ADCSR. (It can be clea
BCLR instruction.)

Figure 16.4 shows an example of A/D converter operation when AN_1 is selected in sele
mode.

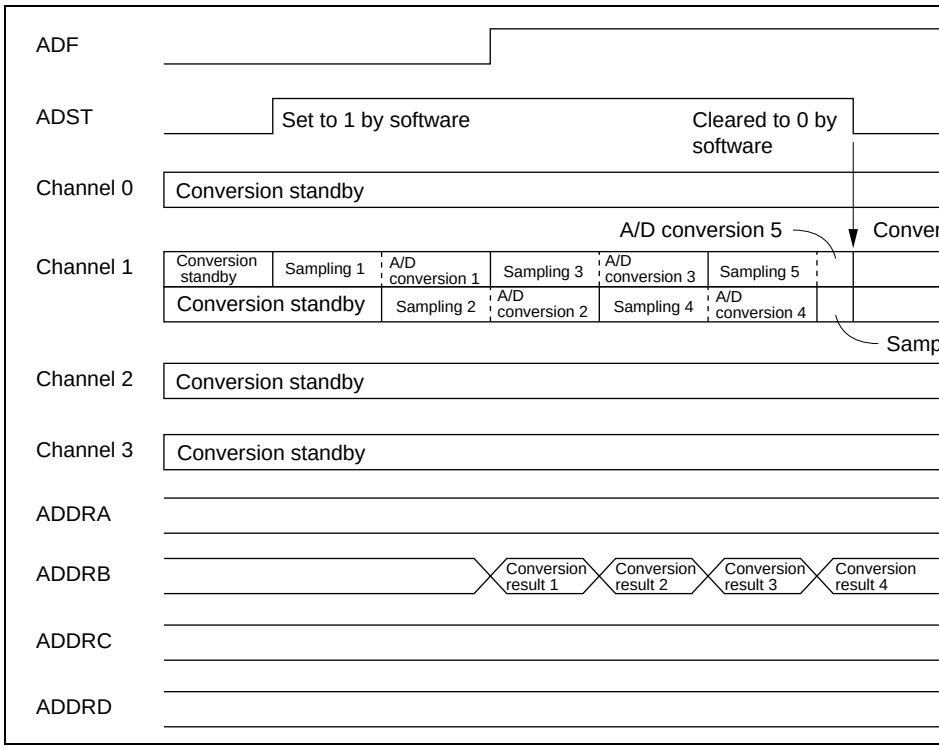


Figure 16.4 Example of A/D Converter Operation (Select Scan Mode)

ADIE bit is set to 1 at this time, an ADI interrupt request is generated. The ADF flag i
writing 0 after reading ADCSR. (It can be cleared with the BCLR instruction.)

Figure 16.5 shows an example of A/D converter operation when AN_0 to AN_2 are selec
single mode.

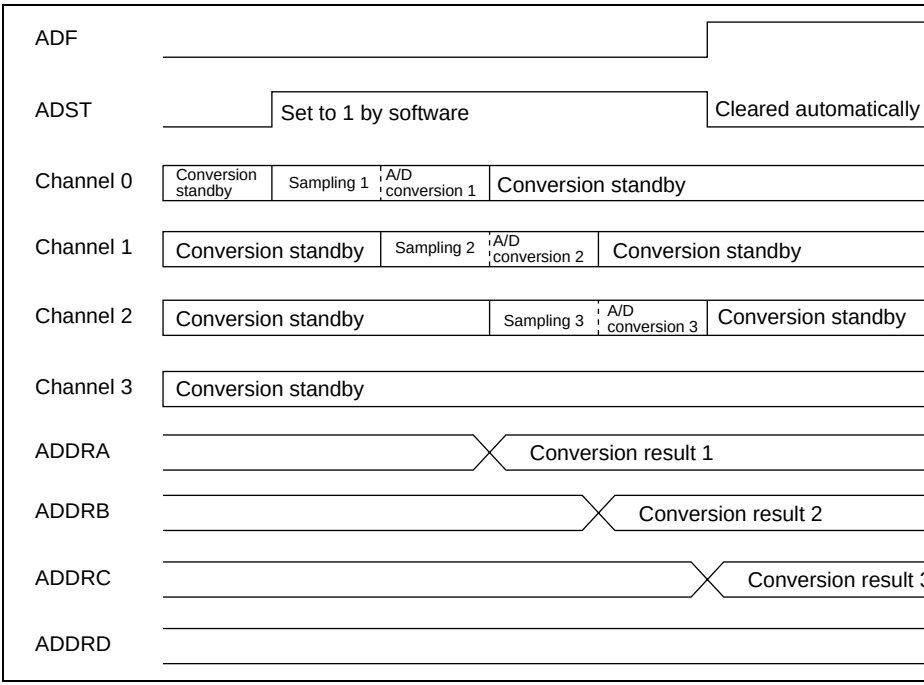


Figure 16.5 Example of A/D Converter Operation (Group Single Mod

When the first conversion operation ends for all the selected input channels, the ADF fl
1. If the ADIE bit is set to 1 at this time, an ADI interrupt request is generated and A/D
is temporarily halted. Clearing the ADF flag to 0 when conversion has been halted by a
interrupt will restart conversion. The ADF flag is cleared by writing 0 after reading AD
can be cleared with the BCLR instruction.)

Figure 16.6 shows an example of A/D converter operation when AN_0 to AN_2 are selecte
scan mode.

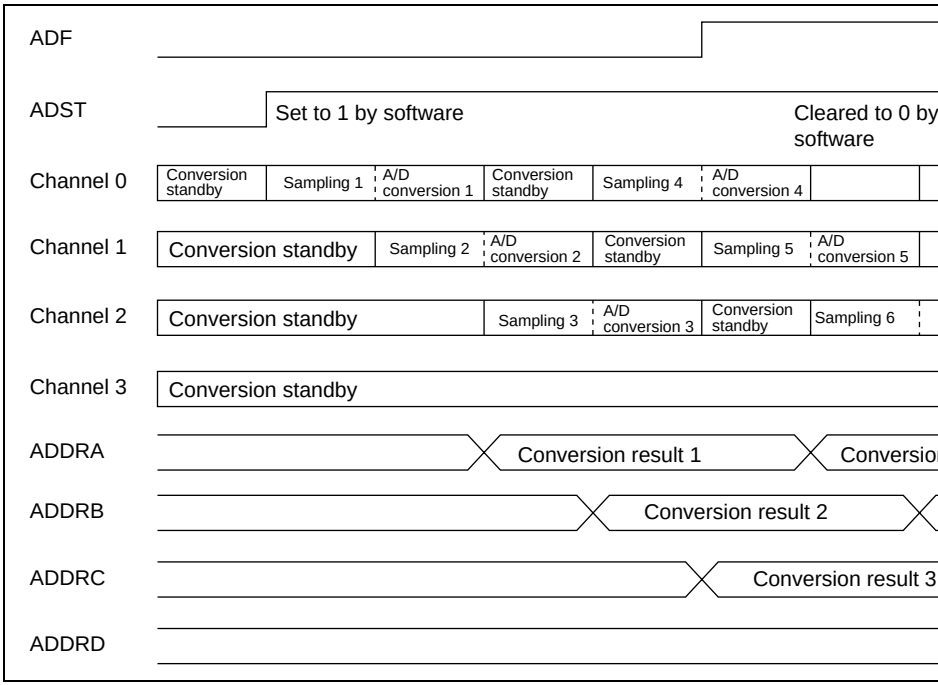


Figure 16.6 Example of A/D Converter Operation (Group Scan Mode)

When using buffer operation in combination with simultaneous sampling operation, se
BUFE1, BUFE0=B'10, and CH2=0.

Figure 16.7 shows buffer operation timing.

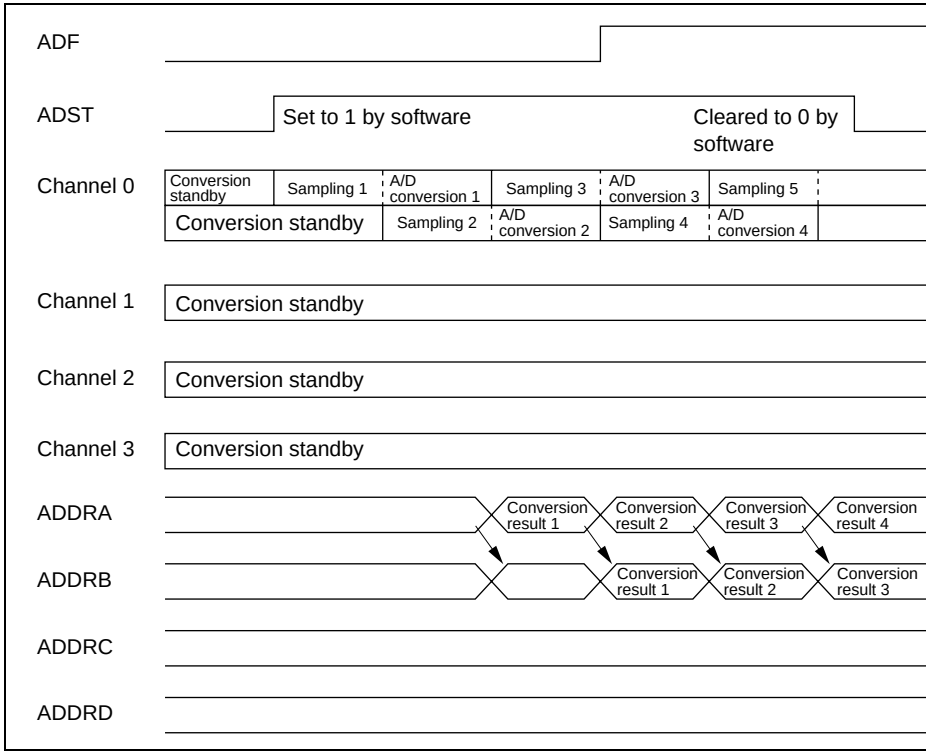


Figure 16.7 Example of Buffer Operation (Select Scan Mode: Two-Stage Operation, CH2 to CH0 = B'001)

the conversion data is stored in order in the buffer register specified by the BUFE1 and
bits.

If the ADIE bit is set to 1 while the ADF flag is set to 1, an ADI interrupt is generated.
flag is cleared to 0 by writing 0 after reading ADCSR. (It can be cleared to 0 with the B
instruction.)

In select single mode, the conversion wait state is entered on completion of each conve
When A/D conversion is restarted by software, a timer trigger, or an external trigger, ar
number of conversions shown in table 16.4 (1) have been completed, the ADF flag is se

Table 16.4 (1) Conversion Channels and ADF Flag Setting/Clearing Conditions in Operation

Setting of CH2 to CH0			Buffer Operation Selection		
CH2	**CH1**	**CH0**	**BUFE1, 0=B'01**	**BUFE1, 0=B'10**	**BUFE1, 0=B'11**
0	0	0	AN_0 once (ADDRA)	AN_0 and AN_1 once each (ADDRB)	AN_0 once (ADD
		1	AN_0 twice (ADDRB)		AN_0 twice (ADD
	1	0	See table 16.4 (2)	AN0 and AN1 twice each (ADDRD)	AN_0 three times
		1			AN_0 four times
1	—	—	See table 16.4 (2)		

Combining Group Mode with Buffer Operation

Bits CH2 to CH0 can be set to perform continuous conversion on the analog input chan
AN_1) specified by bits BUFE1 and BUFE0, and AN_4 to AN_7.

Table 16.4 (2) shows the conversion operation and ADF flag setting conditions in buffe
operation. The ADF flag is set on completion of the last conversion shown in the table.
case, the analog input corresponding to the ADDR specified in the buffer register is not
For example, if BUFE1 and BUFE0 = B'11, and CH2 to CH0 = B'110, the conversion r

Setting of CH2 to CH0			Buffer Operation Selection		
CH2	**CH1**	**CH0**	**BUFE1, 0 = B'01**	**BUFE1, 0 = B'10**	**BUFE1, 0**
0	0	—	See table 16.4 (1)		
	1	0	AN_0, AN_2 (ADDRC)	See table 16.4 (1)	
		1	AN_0, AN_2, AN_3 (ADDRD)		
1	0	0	AN_0, AN_2 to AN_4 (ADDRE)	AN_0, AN_1, AN_4 (ADDRE)	AN_0, AN_4 (ADDRE)
		1	AN_0, AN_2 to AN_5 (ADDRF)	AN_0, AN_1, AN_4, AN_5 (ADDRF)	AN_0, AN_4, (ADDRF)
	1	0	AN_0, AN_2 to AN_6 (ADDRG)	AN_0, AN_1, AN_4 to AN_6 (ADDRG)	AN_0, AN_4 t (ADDRG)
		1	AN_0, AN_2 to AN_7 (ADDRH)	AN_0, AN_1, AN_4 to AN_7 (ADDRH)	AN_0, AN_4 t (ADDRH)

Clearing the ADF Flag

If the DTC or DMAC is activated by an A/D conversion end interrupt, the ADF flag is
when the ADDR specified in table 16.4 is read.

To Reset the Number of Buffer Operations

Suspend the conversion wait state or conversion, and clear the BUFE1 and BUFE0 bit
The buffer count will be cleared to 0.

In simultaneous sampling operation, the input voltages of two channels are sampled simultaneously, and continuous conversion is performed. Simultaneous sampling opera enabled in group mode. The channels involved in simultaneous sampling operation are by bits CH2 and CH1. The combinations of these bits are shown in table 16.5.

For example, simultaneous sampling will be performed when CH2 and CH1 = B'11, on pairs AN_0, $AN_1 \rightarrow AN_2$, $AN_3 \rightarrow AN_4$, $AN_5 \rightarrow AN_6$, AN_7 in that order if GRP=1. Simulta sampling timing is shown in figure 16.8.

Table 16.5 Simultaneous Sampling Channels

Channel Setting		Sampled Channels
CH2	**CH1**	**GRP = 1**
0	0	AN_0, AN_1
	1	AN_0, $AN_1 \rightarrow AN_2$, AN_3
1	0	AN_0, $AN_1 \rightarrow AN_2$, $AN_3 \rightarrow AN_4$, AN_5
	1	AN_0, $AN_1 \rightarrow AN_2$, $AN_3 \rightarrow AN_4$, $AN_5 \rightarrow AN_6$, AN_7

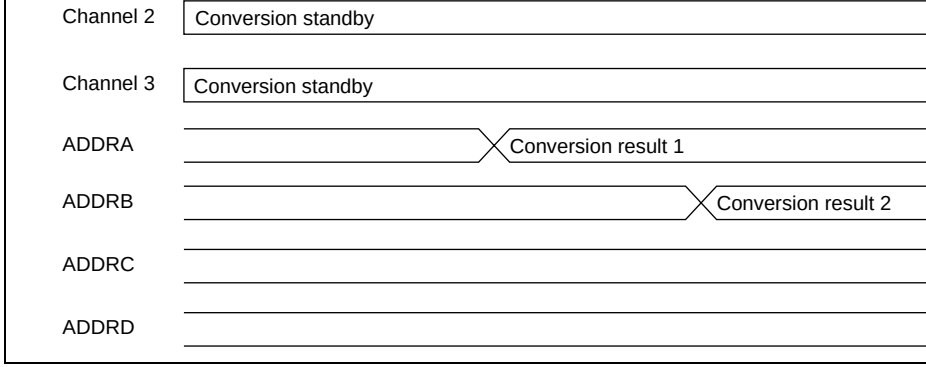


Figure 16.8 Example of Simultaneous Sampling Operation (Group Single

16.4.7 Conversion Start Modes

The A/D converter's conversion start mode is set by means of the PWR bit in ADCSR PWR bit is cleared to 0, low-power conversion mode is set, and the internal analog cir made inactive. When the PWR bit is set to 1, high-speed start mode is set, and the ana is made active.

In low-power conversion mode, the analog circuit power is turned on simultaneously w of conversion (ADST setting), and after 200 cycles of the reference clock the analog c changes to the ready state, and the first A/D conversion operation is started. The refere selected by the CKS bit in ADCSR. When conversion is carried out continuously, the subsequent A/D conversion operations are performed every 10 cycles. When A/D con ends, ADST is cleared to 0 and the analog circuit power is cut automatically. Since the circuitry is only active during A/D conversion in this mode, current dissipation can be

In high-speed start mode, even when A/D conversion ends and ADST is cleared to 0, continues to be supplied to the analog circuitry and conversion can still be carried out. is started as soon as ADST is set to 1 again. Only in the case of the first conversion aft analog power supply is turned on, conversion does not begin until 200 cycles after AD

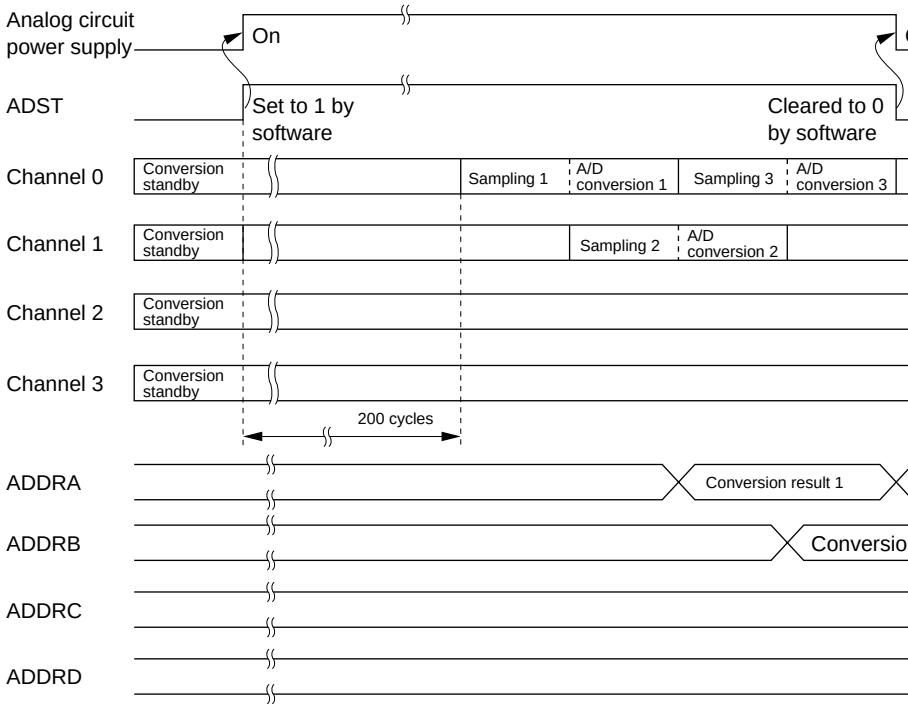


Figure 16.9 Conversion Start Operation (Low-Power Conversion Mode

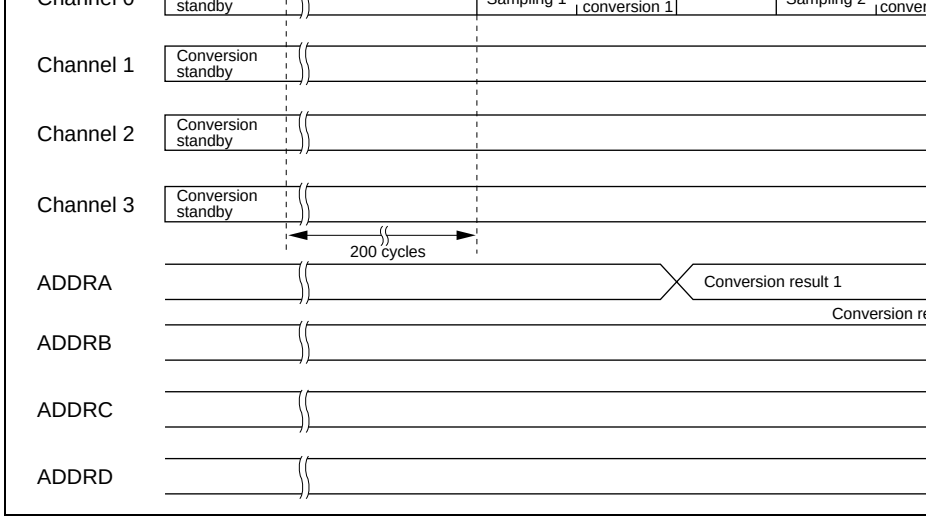


Figure 16.10 Conversion Start Operation (High-Speed Start Mode)

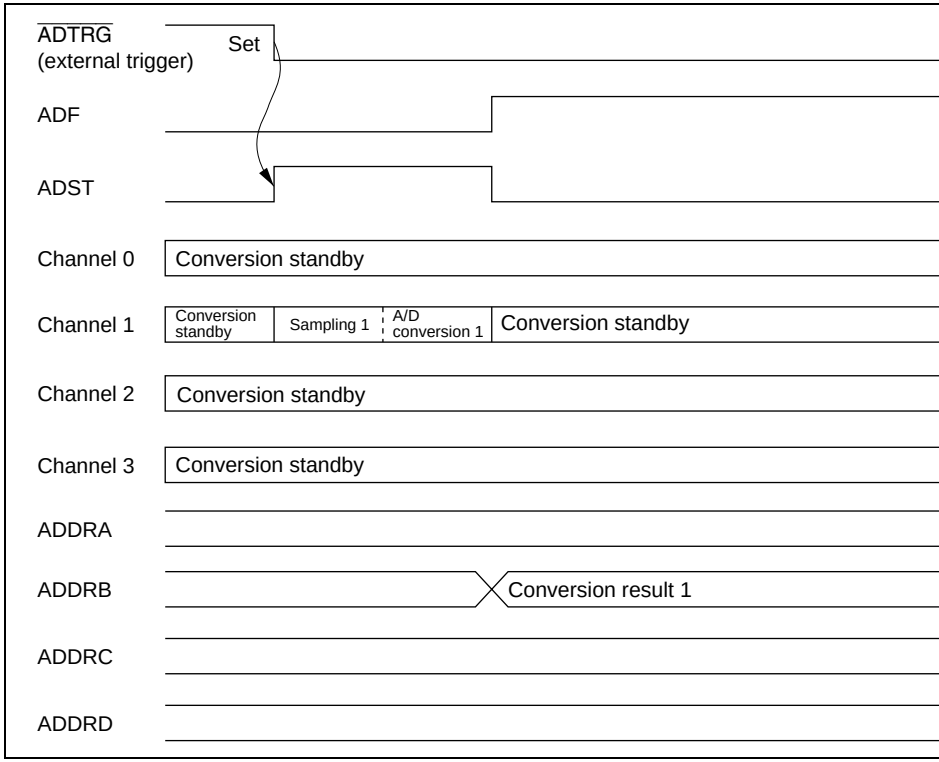
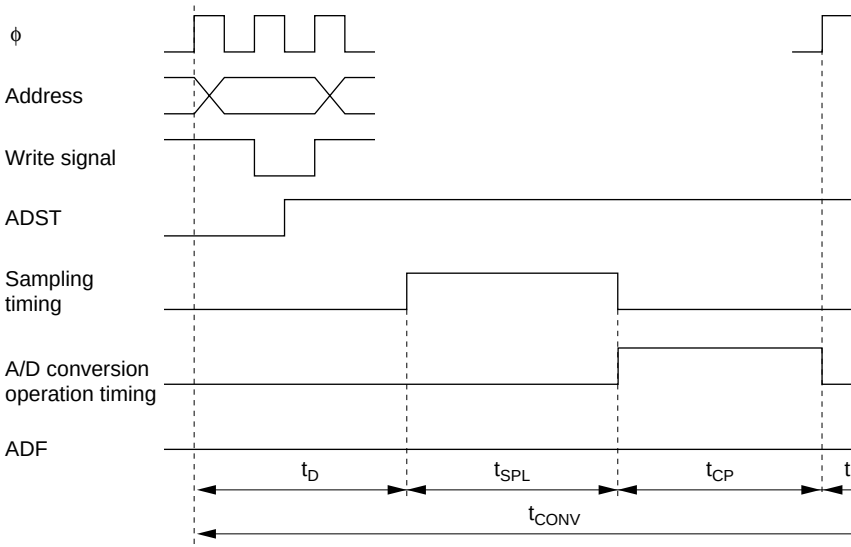


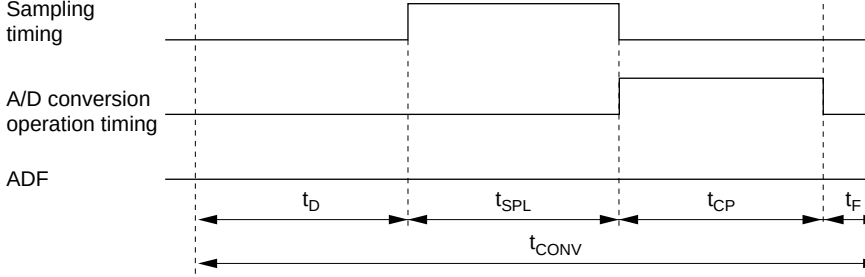
Figure 16.11 Start of Conversion by $\overline{\text{ADTRG}}$ Conversion Start Trigger

A/D conversion timing is shown in figures 16.12 (1) and (2), and A/D conversion time
16.6.



Legend:
t_D : A/D conversion start delay time
t_{SPL} : Input sampling time
t_{CONV} : A/D conversion time
t_{CP} : A/D conversion operation time
t_F : ADF flag set delay time

Figure 16.12 (1) A/D Conversion Timing



Legend:
t_D : A/D conversion start delay time
t_{SPL} : Input sampling time
t_{CONV} : A/D conversion time
t_{CP} : A/D conversion operation time
t_F : ADF flag set delay time

Figure 16.12 (2) A/D Conversion Timing

Table 16.6 A/D Conversion Times

	Symbol	CKS = 0	CKS
A/D conversion start delay time	t_D	3	3
Input sampling time	t_{SPL}	10	20
A/D conversion operation time	t_{CP}	10	20
ADF flag set delay time	t_F	1	1
A/D conversion time	t_{CONV}	24	44

Note: Units: States

The figures in the table are for when PWR = 1. If 200 states have not elapsed si of the PWR bit, conversion is not performed until 200 states have elapsed. Whe 200 states should be added to the first A/D conversion start delay time. When co carried out continuously, the second and subsequent t_{CONV} values are obtained b subtracting t_{SPL}.

CKS	Conversion Time (States)	Minimum Conversion Time (µs)				
		20 MHz	16 MHz	10 MHz	8 MHz	2
0	24	—	—	2.4	2.8	1
1	44	2.2	2.8	4.4	5.5	2

Legend:
—: Cannot be set.

(2) When AV_{CC} < 4.5 V, t_{CONV} ≥ 4 µs

Condition: V_{CC} = 2.7 to 5.5 V, AV_{CC} = 2.7 V to 4.5 V, V_{ref} = 2.7 V to AV_{CC}, V_{ss} = AV_{ss}

Table 16.7 (2) Operating Frequencies and CKS Bit Settings

CKS	Conversion Time (States)	Minimum Conversion Time (µs)				
		10 MHz	8 MHz	5 MHz	4 MHz	2
0	24	—	—	4.8	6.0	1
1	44	4.4	5.5	8.8	11.0	2

Legend:
—: Cannot be set.

The A/D converter interrupt source is shown in table 16.8.

If the ADIE bit is set to 1 in scan mode, setting the ADF flag to 1 will temporarily halt conversion. A/D conversion is restarted when the ADF flag is cleared to 0.

When the DTC or DMAC is activated by an ADI interrupt and the last of the specified registers is read, the ADF flag is cleared to 0.

Table 16.8 A/D Converter Interrupt Source

Interrupt Source	Description	DTC or DMAC Activa
ADI	Interrupt due to end of conversion	Possible

The AV_{CC} and AV_{SS} input voltages should be set as follows: $AV_{CC} = V_{CC} \pm 10\%$, AV
When the A/D converter is not used, set $AV_{CC} = V_{CC}$, $AV_{SS} = V_{SS}$. In standby mode,
$AV_{CC} \leq 5.5$ V, $AV_{SS} = V_{SS}$ (where V_{RAM} is the RAM standby voltage).

- V_{ref} input voltage

  The analog reference voltage V_{ref} should be set as follows: $V_{ref} \leq AV_{CC}$. When the A
  converter is not used, set $V_{ref} = V_{CC}$. In standby mode, set $V_{RAM} \leq V_{ref} \leq AV_{CC}$ (where
  RAM standby voltage).

- Input ports

  When a circuit is connected to an input port, the constant should be set to a value l
  A/D converter sampling time. If the constant is large in the case of a circuit, the in
  may not be sampled properly.

- Conversion start mode

  There is a difference in the current dissipation between high-speed start mode and
  conversion mode selected for A/D conversion operation according to the PWR bit

D/A converter features are listed below.

- 8-bit resolution
- Two output channels
- Maximum conversion time of 10 µs (with 20 pF load)
- Output voltage of 0 V to Vref
- D/A output hold function in software standby mode

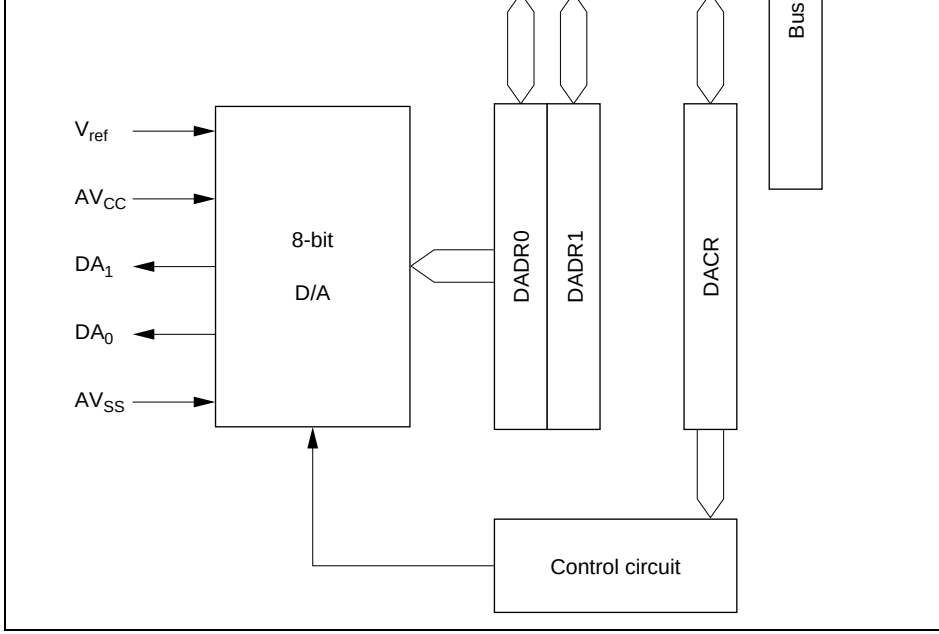


Figure 17.1 Block Diagram of D/A Converter

Analog output pin 0	DA_0	Output	Channel 0 analog output
Analog output pin 1	DA_1	Output	Channel 1 analog output
Reference voltage pin	V_{ref}	Input	Analog reference voltage

17.1.4 Register Configuration

Table 17.2 summarizes the registers of the D/A converter.

Table 17.2 D/A Converter Registers

Name	Abbreviation	R/W	Initial Value	A
D/A data register 0	DADR0	R/W	H'00	H
D/A data register 1	DADR1	R/W	H'00	H
D/A control register	DACR	R/W	H'1F	H
Module stop control register	MSTPCR	R/W	H'3FFF	H

Note: * Lower 16 bits of the address.

DADR0 and DADR1 are 8-bit readable/writable registers that store data for conversion

Whenever output is enabled, the values in DADR0 and DADR1 are converted and outp
analog output pins.

DADR0 and DADR1 are each initialized to H'00 by a reset and in hardware standby m

17.2.2 D/A Control Register (DACR)

Bit	:	7	6	5	4	3	2	1	0
		DAOE1	DAOE0	DAE	—	—	—	—	—
Initial value	:	0	0	0	1	1	1	1	1
R/W	:	R/W	R/W	R/W	—	—	—	—	—

DACR is an 8-bit readable/writable register that controls the operation of the D/A conv

DACR is initialized to H'1F by a reset and in hardware standby mode.

Bit 7—D/A Output Enable 1 (DAOE1): Controls D/A conversion and analog output f
1.

Bit 7

DAOE1	Description
0	Analog output DA_1 is disabled
1	Channel 1 D/A conversion is enabled; analog output DA_1 is enabled

Bit 5—D/A Enable (DAE): The DAOE0 and DAOE1 bits both control D/A conversi
the DAE bit is cleared to 0, the channel 0 and 1 D/A conversions are controlled indepe
When the DAE bit is set to 1, the channel 0 and 1 D/A conversions are controlled toge

Output of resultant conversions is always controlled independently by the DAOE0 and
bits.

Bit 7	Bit 6	Bit 5	
DAOE1	**DAOE0**	**DAE**	**Description**
0	0	*	Channel 0 and 1 D/A conversions disabled
	1	0	Channel 0 D/A conversion enabled Channel 1 D/A conversion disabled
		1	Channel 0 and 1 D/A conversions enabled
1	0	0	Channel 0 D/A conversion disabled Channel 1 D/A conversion enabled
		1	Channel 0 and 1 D/A conversions enabled
	1	*	Channel 0 and 1 D/A conversions enabled

Legend: *: Don't care

If the H8S/2655 Group enters software standby mode when D/A conversion is enable
output is held and the analog power current is the same as during D/A conversion. Wh
necessary to reduce the analog power current in software standby mode, clear both the
and DAOE1 bits to 0 to disable D/A output.

Bits 4 to 0—Reserved: Read-only bits, always read as 1.

MSTPCR is a 16-bit readable/writable register that performs module stop mode control

When the MSTP10 bit in MSTPCR is set to 1, D/A converter operation stops at the end cycle and a transition is made to module stop mode. Registers cannot be read or written module stop mode. For details, see section 21.5, Module Stop Mode.

MSTPCR is initialized to H'3FFF by a reset and in hardware standby mode. It is not ini software standby mode.

Bit 10—Module Stop (MSTP10): Specifies the D/A converter module stop mode.

Bit 10

MSTP10	Description	
0	D/A converter module stop mode cleared	
1	D/A converter module stop mode set	(

The operation example described in this section concerns D/A conversion on channel
17.2 shows the timing of this operation.

[1] Write the conversion data to DADR0.

[2] Set the DAOE0 bit in DACR to 1. D/A conversion is started and the DA_0 pin beco
output pin. The conversion result is output after the conversion time has elapsed. T
value is expressed by the following formula:

$$\frac{\text{DADR contents}}{256} \times V_{ref}$$

The conversion results are output continuously until DADR0 is written to again or
bit is cleared to 0.

[3] If DADR0 is written to again, the new data is immediately converted. The new co
result is output after the conversion time has elapsed.

[4] If the DAOE0 bit is cleared to 0, the DA_0 pin becomes an input pin.

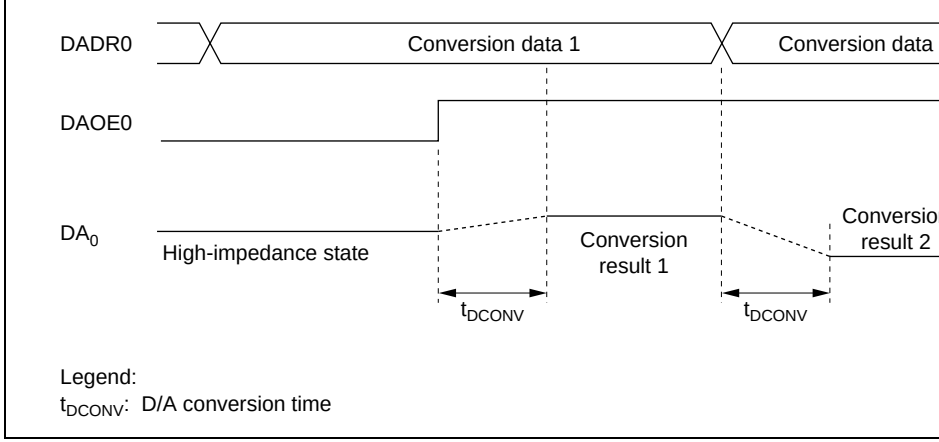


Figure 17.2 Example of D/A Converter Operation

The on-chip RAM can be enabled or disabled by means of the RAM enable bit (RAM system control register (SYSCR).

18.1.1 Block Diagram

Figure 18.1 shows a block diagram of the on-chip RAM.

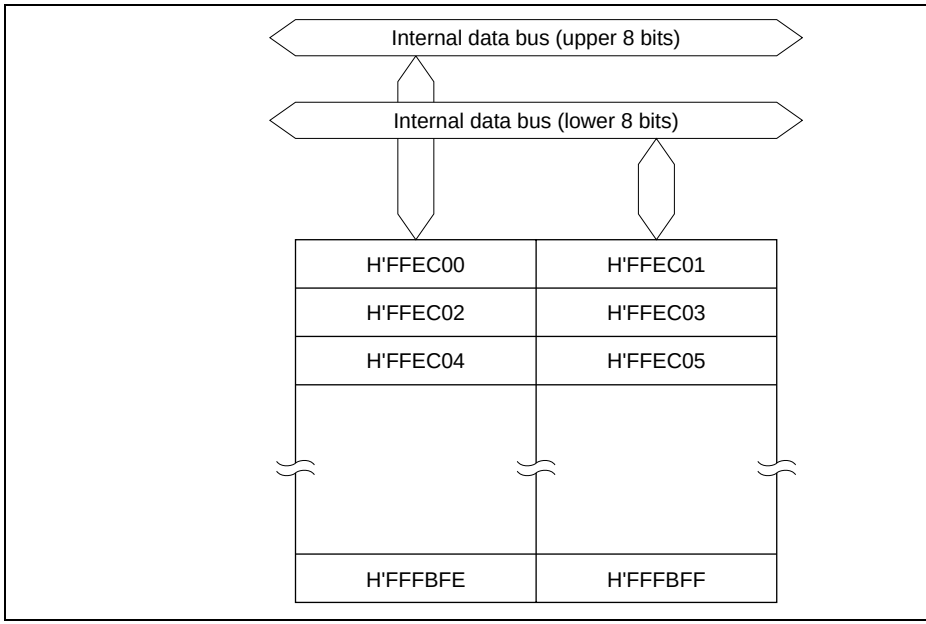


Figure 18.1 Block Diagram of RAM

Note: * Lower 16 bits of the address.

18.2 Register Descriptions

18.2.1 System Control Register (SYSCR)

Bit	:	7	6	5	4	3	2	1
		MACS	—	INTM1	INTM0	NMIEG	—	—
Initial value	:	0	0	0	0	0	0	0
R/W	:	R/W	—	R/W	R/W	R/W	—	—

The on-chip RAM is enabled or disabled by the RAME bit in SYSCR. For details of ot
SYSCR, see section 3.2.2, System Control Register (SYSCR).

Bit 0—RAM Enable (RAME): Enables or disables the on-chip RAM. The RAME bit
initialized when the reset state is released. It is not initialized in software standby mode

Bit 0

RAME	Description
0	On-chip RAM is disabled
1	On-chip RAM is enabled (I

at an even address.

18.4 Usage Notes

DTC register information can be located in addresses H'FFF800 to H'FFFBFF. When used, the RAME bit must not be cleared to 0.

processing.

The on-chip ROM is enabled or disabled by setting the mode pins (MD_2, MD_1, and M
EAE in BCRL.

The PROM version of the H8S/2655 Group can be programmed with a general-purpos
programmer, by setting PROM mode.

19.1.1 Block Diagram

Figure 19.1 shows a block diagram of the on-chip ROM.

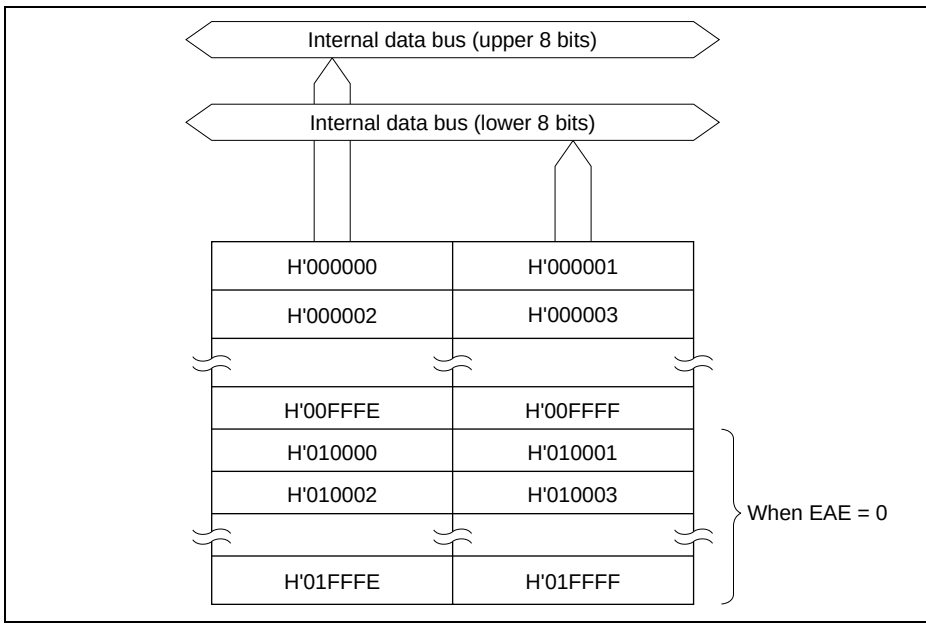


Figure 19.1 Block Diagram of ROM (H8S/2655)

Bus control register L	BCRL	R/W	H'3C	Retained	H

Note: * Lower 16 bits of the address.

19.2 Register Descriptions

19.2.1 Bus Control Register L (BCRL)

Bit	:	7	6	5	4	3	2	1
		BRLE	BREQOE	EAE	LCASS	DDS	ASS	WDBE
Initial value	:	0	0	1	1	1	1	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Enabling or disabling of part of the H8S/2655's on-chip ROM area can be selected by m
the EAE bit in BCRL. For details of the other bits in BCRL, see 6.2.5, Bus Control Reg
(BCRL).

Bit 5—External Address Enable (EAE): Selects whether addresses H'010000 to H'01
to be internal addresses or external addresses.

This setting is invalid in normal mode.

Bit 5

EAE	Description
0	Addresses H'010000 to H'01FFFF are in on-chip ROM (in the H8S/2655) or a reserved area* (in the H8S/2653).
1	Addresses H'010000 to H'01FFFF are external addresses (external expans or a reserved area* (single-chip mode). (

Note: * Reserved areas should not be accessed.

In normal mode, a maximum of 56 kbytes of ROM can be used.

Table 19.2 Operating Modes and ROM Area

Operating Mode		Mode Pin			BCRL	
		MD2	MD1	MD0	EAE	On-Chi
Mode 1	Normal expanded mode with on-chip ROM disabled	0	0	1	—	Disable
Mode 2	Normal expanded mode with on-chip ROM enabled		1	0	—	Enabled (56 kbyt
Mode 3	Normal single-chip mode			1		
Mode 4	Advanced expanded mode with on-chip ROM disabled	1	0	0	—	Disable
Mode 5	Advanced expanded mode with on-chip ROM disabled			1		
Mode 6	Advanced expanded mode with on-chip ROM enabled		1	0	0	Enabled
					1	Enabled
Mode 7	Advanced single-chip mode			1	0	Enabled
					1	Enabled

Note: * 128 kbytes in the H8S/2655, 64 kbytes in the H8S/2653
In H8/2655 modes 6 and 7, the on-chip ROM available after a power-on res
kbyte area comprising addresses H'000000 to H'00FFFF.

Use of a 120-pin/32-pin socket adapter enables programming with a commercial PROM programmer.

Note that the PROM programmer should not be set to page mode as the H8S/2655 Group does not support page programming.

Table 19.3 shows how PROM mode is selected.

Table 19.3 Selecting PROM Mode

Pin Names	Setting
MD_2, MD_1, MD_0	Low
$\overline{STBY}$	
PA_2, PA_1	High

19.4.2 Socket Adapter and Memory Map

Programs can be written and verified by attaching a 120-pin/32-pin socket adapter to the PROM programmer. Table 19.4 gives ordering information for the socket adapter, and figure 19.2 shows the wiring of the socket adapter. Figure 19.3 shows the memory map in PROM mode.

50	56	PD_6	EO_6	20
51	57	PD_7	EO_7	21
2	6	PC_0	EA_0	12
3	7	PC_1	EA_1	11
4	8	PC_2	EA_2	10
5	9	PC_3	EA_3	9
7	11	PC_4	EA_4	8
8	12	PC_5	EA_5	7
9	13	PC_6	EA_6	6
10	14	PC_7	EA_7	5
11	15	PB_0	EA_8	27
74	82	NMI	EA_9	26
13	17	PB_2	EA_{10}	23
14	18	PB_3	EA_{11}	25
16	20	PB_4	EA_{12}	4
17	21	PB_5	EA_{13}	28
18	22	PB_6	EA_{14}	29
19	23	PB_7	EA_{15}	3
20	24	PA_0	EA_{16}	2
86	94	PF_2	$\overline{CE}$	22
12	16	PB_1	$\overline{OE}$	24
87	95	PF_1	$\overline{PGM}$	31
1, 33, 52, 76, 81	1, 39, 58, 84, 89	V_{CC}	V_{CC}	32
93	103	AV_{CC}		
94	104	V_{ref}		
21	25	PA_1		
22	26	PA_2		
6, 15, 24, 38, 47, 59, 79, 104	3, 10, 19, 28, 35, 36, 44, 53, 65, 67, 68, 87, 99, 100,114	V_{SS}	V_{SS}	16
103	113	AV_{SS}		
75	83	$\overline{STBY}$		
113	123	MD_0		
114	124	MD_1		
115	125	MD_2		

Legend:

V_{PP}	: Programming power supply (12.5 V)
EO_7 to EO_0	: Data input/output
EA_{16} to EA_0	: Address input
$\overline{OE}$	: Output enable
$\overline{CE}$	: Chip enable
$\overline{PGM}$	: Program

Note: Pins not shown in this figure should be left open.

Figure 19.2 Wiring of 120-Pin/32-Pin Socket Adapter

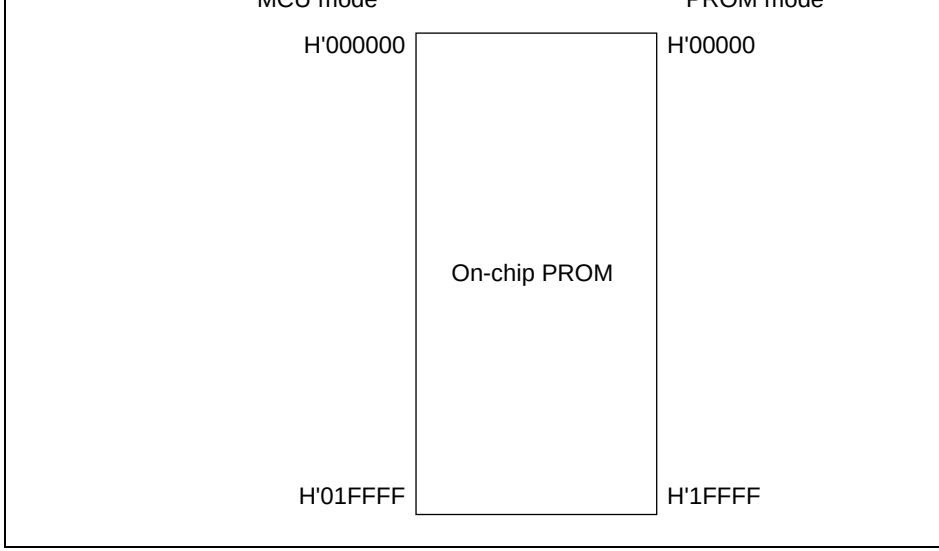


Figure 19.3 Memory Map in PROM Mode

Mode	$\overline{CE}$	$\overline{OE}$	$\overline{PGM}$	V_{PP}	V_{CC}	EO_7 to EO_0	EA_{16}
Program	L	H	L	V_{PP}	V_{CC}	Data input	Addr
Verify	L	L	H	V_{PP}	V_{CC}	Data output	Addr
Program-inhibit	L	L	L	V_{PP}	V_{CC}	High impedance	Addr
	L	H	H				
	H	L	L				
	H	H	H				

Legend:
L : Low voltage level
H : High voltage level
V_{PP} : V_{PP} voltage level
V_{CC} : V_{CC} voltage level

Programming and verification should be carried out using the same specifications as f
standard HN27C101 EPROM.

However, do not set the PROM programmer to page mode does not support page prog
PROM programmer that only supports page programming cannot be used. When choo
PROM programmer, check that it supports high-speed programming in byte units. Alv
addresses within the range H'00000 to H'1FFFF.

19.5.2 Programming and Verification

An efficient, high-speed programming procedure can be used to program and verify P
This procedure writes data quickly without subjecting the chip to voltage stress or sac
reliability. It leaves the data H'FF in unused addresses. Figure 19.4 shows the basic hi
programming flowchart. Tables 19.6 and 19.7 list the electrical characteristics of the c
programming. Figure 19.5 shows a timing chart.

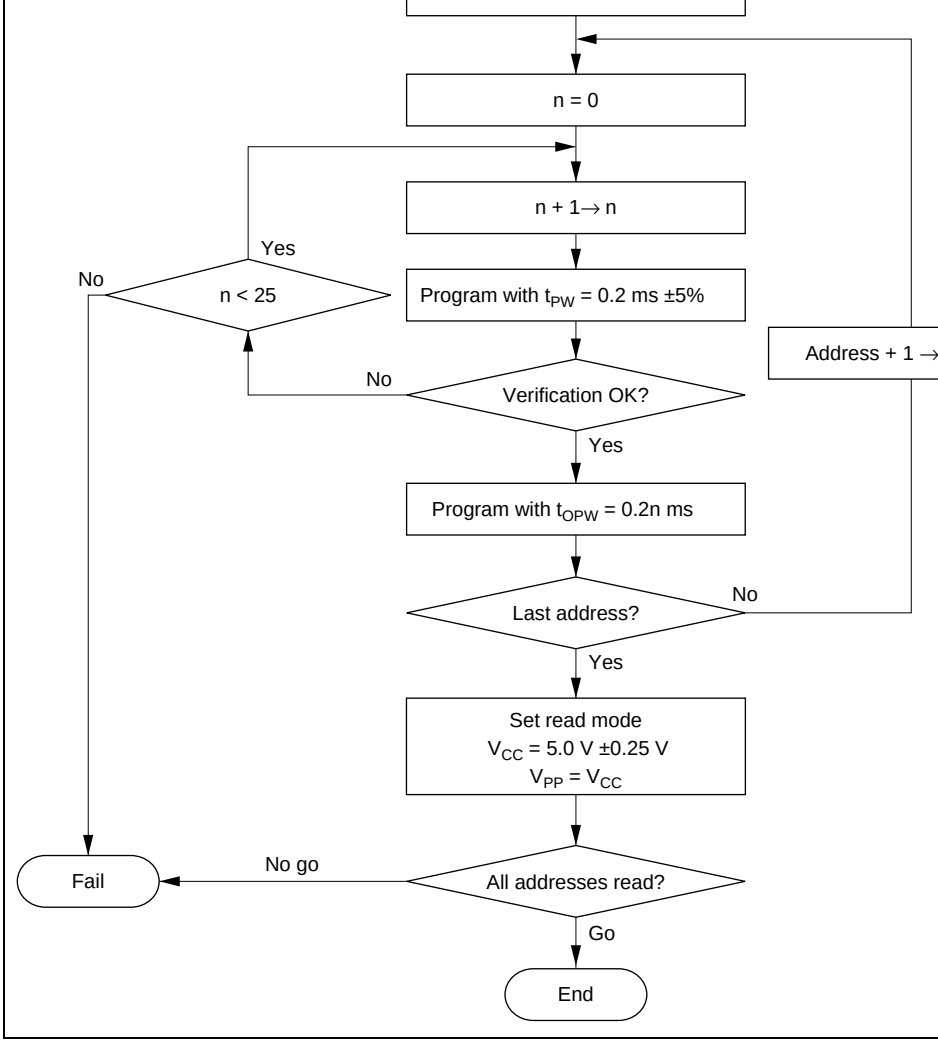


Figure 19.4 High-Speed Programming Flowchart

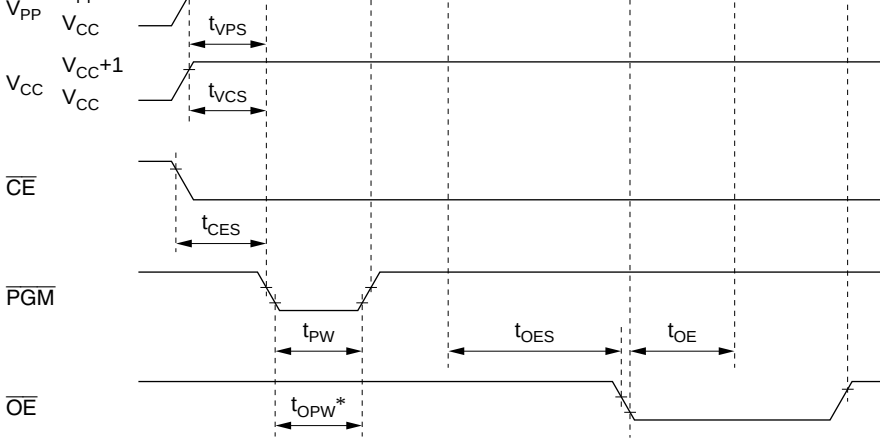
Input low voltage	EO_7 to EO_0, EA_{16} to EA_0, $\overline{OE}$, $\overline{CE}$, $\overline{PGM}$	V_{IL}	–0.3	—	0.8	V	
Output high voltage	EO_7 to EO_0	V_{OH}	2.4	—	—	V	I_O
Output low voltage	EO_7 to EO_0	V_{OL}	—	—	0.45	V	I_O
Input leakage current	EO_7 to EO_0, EA_{16} to EA_0, $\overline{OE}$, $\overline{CE}$, $\overline{PGM}$	$\lvert I_{LI} \rvert$	—	—	2	µA	V 5.
V_{CC} current		I_{CC}	—	—	40	mA	
V_{PP} current		I_{PP}	—	—	40	mA	

Data setup time	t_{DS}	2	—	—	µs
Address hold time	t_{AH}	0	—	—	µs
Data hold time	t_{DH}	2	—	—	µs
Data output disable time	t_{DF} *2	—	—	130	ns
V_{PP} setup time	t_{VPS}	2	—	—	µs
Programming pulse width	t_{PW}	0.19	0.20	0.21	ms
$\overline{PGM}$ pulse width for overwrite programming	t_{OPW} *3	0.19	—	5.25	ms
V_{CC} 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: 0.8 V to 2.2 V
Input rise time and fall time ≤ 20 ns
Timing reference levels: Input: 1.0 V, 2.0 V
Output: 0.8 V, 2.0 V

2. t_{DF} is defined to be when output has reached the open state, and the output l longer be referenced.

3. t_{OPW} is defined by the value shown in the flowchart.



Note: * t_{OPW} is defined by the value shown in the flowchart.

Figure 19.5 PROM Programming/Verification Timing

- Before programming, check that the MCU is correctly mounted in the PROM progr
  Overcurrent damage to the MCU can result if the index marks on the PROM progra
  socket adapter, and MCU are not correctly aligned.

- Do not touch the socket adapter or MCU while programming. Touching either of th
  cause contact faults and programming errors.

- The MCU cannot be programmed in page programming mode. Select the programm
  carefully.

- The size of the H8S/2655 PROM is 128 kbytes. Always set addresses within the ran
  H'00000 to H'1FFFF. During programming, write H'FF to unused addresses to avoi
  verification errors.

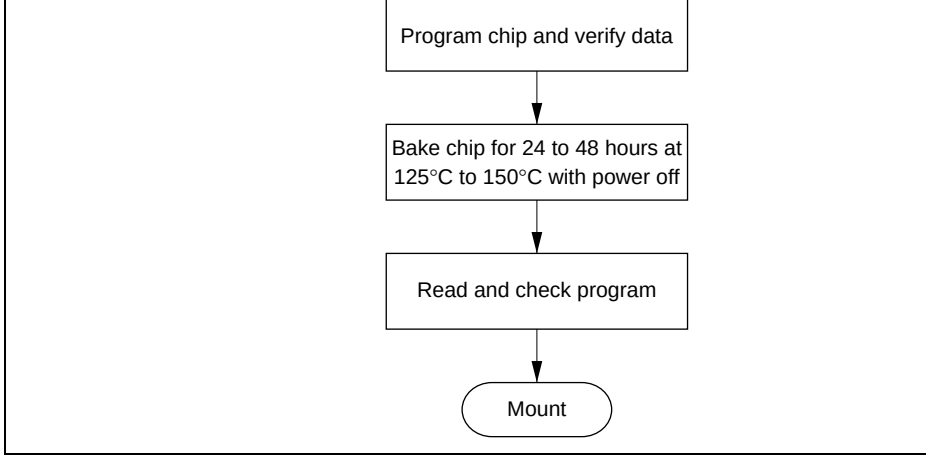


Figure 19.6 Recommended Screening Procedure

If a series of programming errors occurs while the same PROM programmer is being u
programming and check the PROM programmer and socket adapter for defects.

Please inform Renesas of any abnormal conditions noted during or after programming
screening of program data after high-temperature baking.

The clock pulse generator consists of an oscillator circuit, a duty adjustment circuit, a speed clock divider, and a bus master clock selection circuit.

20.1.1 Block Diagram

Figure 20.1 shows a block diagram of the clock pulse generator.

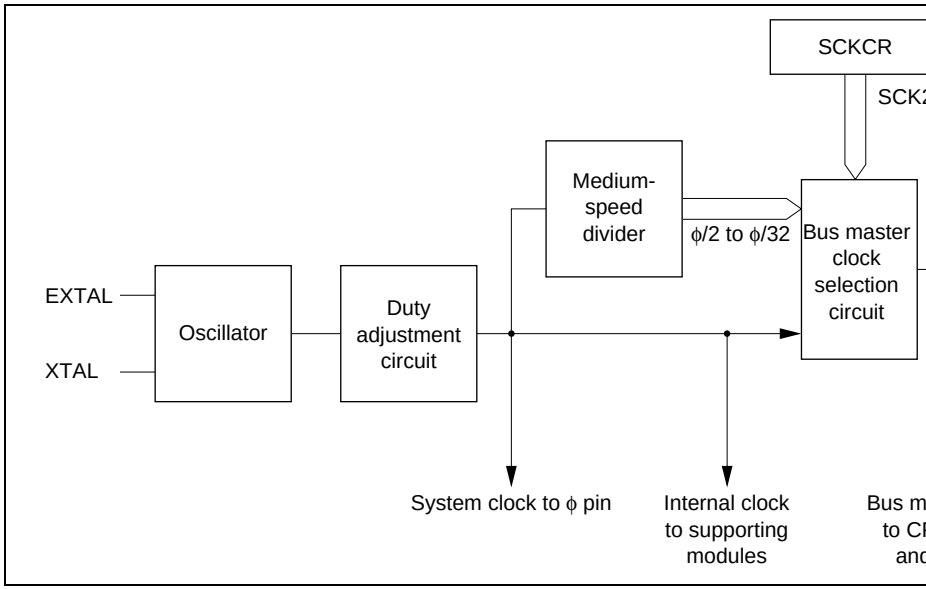


Figure 20.1 Block Diagram of Clock Pulse Generator

20.2 Register Descriptions

20.2.1 System Clock Control Register (SCKCR)

Bit	:	7	6	5	4	3	2	1
		PSTOP	—	—	—	—	SCK2	SCK1
Initial value	:	0	0	0	0	0	0	0
R/W	:	R/W	—	—	—	—	R/W	R/W

SCKCR is an 8-bit readable/writable register that performs ϕ clock output control and r
speed mode control.

SCKCR is initialized to H'00 by a reset and in hardware standby mode. It is not initializ
software standby mode.

Bit 7—ϕ Clock Output Disable (PSTOP): Controls ϕ output.

	Description			
Bit 7				
PSTOP	**Normal Operation**	**Sleep Mode**	**Software Standby Mode**	**Hardwa Standby**
0	ϕ output (initial value)	ϕ output	Fixed high	High imp
1	Fixed high	Fixed high	Fixed high	High imp

Bit 6—Reserved: This bit can be read or written to, but only 0 should be written.

Bits 5 to 3—Reserved: Read-only bits, always read as 0.

		1	Medium-speed clock is ϕ/8
1	0	0	Medium-speed clock is ϕ/16
		1	Medium-speed clock is ϕ/32
	1	—	—

20.3 Oscillator

Clock pulses can be supplied by connecting a crystal resonator, or by input of an exter

20.3.1 Connecting a Crystal Resonator

Circuit Configuration

A crystal resonator can be connected as shown in the example in figure 20.2. Select th
resistance R_d according to table 20.2. An AT-cut parallel-resonance crystal should be u

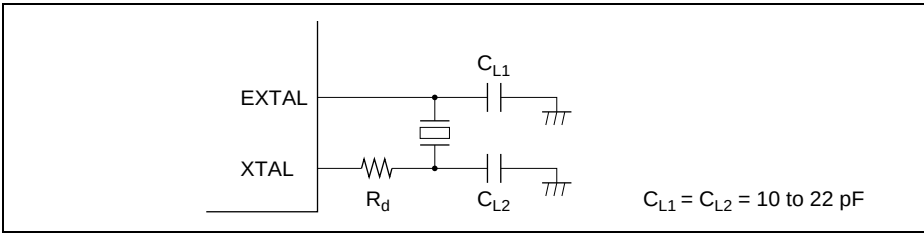


Figure 20.2 Connection of Crystal Resonator (Example)

Table 20.2 Damping Resistance Value

Frequency (MHz)	2	4	8	12	16
R_d (Ω)	1 k	500	200	0	0

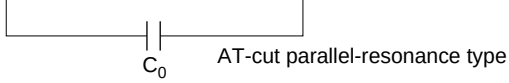


Figure 20.3 Crystal Resonator Equivalent Circuit

Table 20.3 Crystal Resonator Parameters

Frequency (MHz)	2	4	8	12	16
R_S max (Ω)	500	120	80	60	50
C_0 max (pF)	7	7	7	7	7

Note on Board Design

When a crystal resonator is connected, the following points should be noted:

Other signal lines should be routed away from the oscillator circuit to prevent induction interfering with correct oscillation. See figure 20.4.

When designing the board, place the crystal resonator and its load capacitors as close as to the XTAL and EXTAL pins.

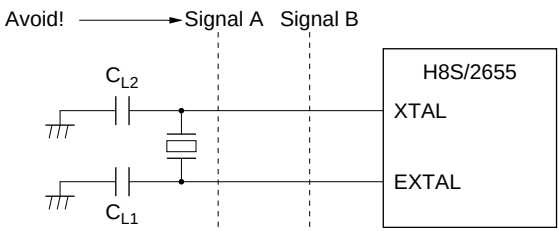


Figure 20.4 Example of Incorrect Board Design

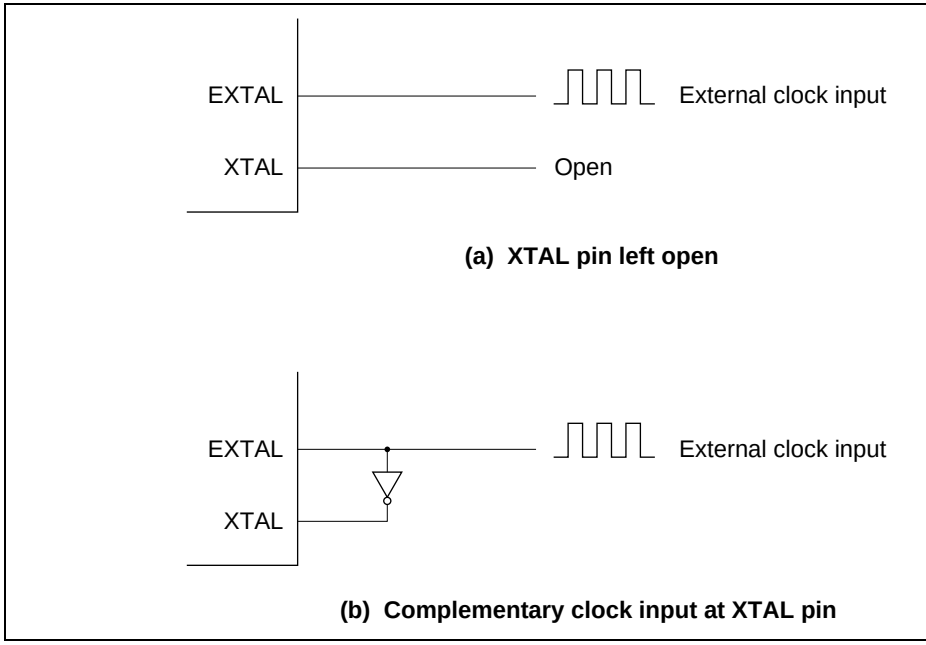


Figure 20.5 External Clock Input (Examples)

Item	Symbol	Min	Max	Min	Max	Unit	Test Conditions
External clock input low pulse width	t_{EXL}	40	—	20	—	ns	Figure 20.6
External clock input high pulse width	t_{EXH}	40	—	20	—	ns	
External clock rise time	t_{EXr}	—	10	—	5	ns	
External clock fall time	t_{EXf}	—	10	—	5	ns	
Clock low pulse width level	t_{CL}	0.4	0.6	0.4	0.6	t_{cyc}	$\phi \geq 5$ MHz
		80	—	80	—	ns	$\phi < 5$ MHz
Clock high pulse width level	t_{CH}	0.4	0.6	0.4	0.6	t_{cyc}	$\phi \geq 5$ MHz
		80	—	80	—	ns	$\phi < 5$ MHz

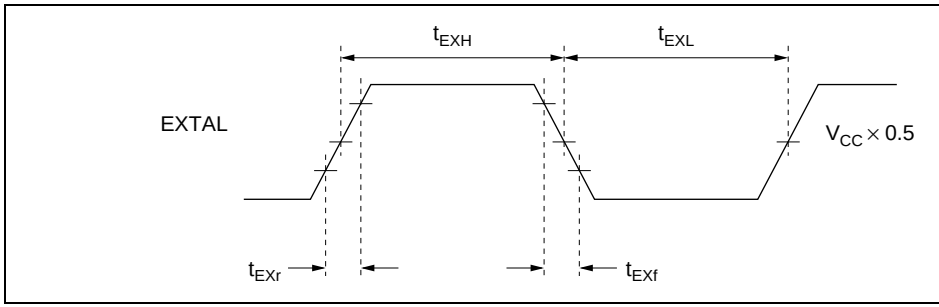


Figure 20.6 External Clock Input Timing

20.6 Bus Master Clock Selection Circuit

The bus master clock selection circuit selects the system clock (ϕ) or one of the mediu
clocks (ϕ/2, ϕ/4, or ϕ/8, ϕ/16, and ϕ/32) to be supplied to the bus master, according to
of the SCK2 to SCK0 bits in SCKCR.

modules, and so on.

The H8S/2655 Group operating modes are as follows:

(1) High-speed mode
(2) Medium-speed mode
(3) Sleep mode
(4) Module stop mode
(5) Software standby mode
(6) Hardware standby mode

Of these, (2) to (6) are power-down modes. Sleep mode is a CPU mode, medium-spee
CPU and bus master mode, and module stop mode is an on-chip supporting module m
(including bus masters other than the CPU). A combination of these modes can be set.

After a reset, the H8S/2655 Group is in high-speed mode.

Table 21.1 shows the conditions for transition to the various modes, the status of the C
supporting modules, etc., and the method of clearing each mode.

Sleep mode	Instruction	Interrupt	Functions	Halted	Retained	High speed	Functions
Module stop mode	Control register		Functions	High/ medium speed	Functions	Halted	Retained/ reset*2
Software standby mode	Instruction	External interrupt	Halted	Halted	Retained	Halted	Retained/ reset*2
Hardware standby mode	Pin		Halted	Halted	Undefined	Halted	Reset

Notes: 1. The bus master operates on the medium-speed clock, and other on-chip sup modules on the high-speed clock.
2. The SCI is reset, and other on-chip supporting modules retain their state.

21.1.1 Register Configuration

Power-down modes are controlled by the SBYCR, SCKCR, and MSTPCR registers. Ta summarizes these registers.

Table 21.2 Power-Down Mode Registers

Name	Abbreviation	R/W	Initial Value	Ad
Standby control register	SBYCR	R/W	H'08	H'
System clock control register	SCKCR	R/W	H'00	H'
Module stop control register H	MSTPCRH	R/W	H'3F	H'
Module stop control register L	MSTPCRL	R/W	H'FF	H'

Note: * Lower 16 bits of the address.

R/W	.	R/W	R/W	R/W	R/W	R/W	—	—

SBYCR is an 8-bit readable/writable register that performs software standby mode co

SBYCR is initialized to H'08 by a reset and in hardware standby mode. It is not initial
software standby mode.

Bit 7—Software Standby (SSBY): Specifies a transition to software standby mode. I
to 1 when software standby mode is released by an external interrupt, and a transition
normal operation. The SSBY bit should be cleared by writing 0 to it.

Bit 7

SSBY	Description
0	Transition to sleep mode after execution of SLEEP instruction
1	Transition to software standby mode after execution of SLEEP instruction

0	0	0	Standby time = 8192 states
		1	Standby time = 16384 states
	1	0	Standby time = 32768 states
		1	Standby time = 65536 states
1	0	0	Standby time = 131072 states
		1	Standby time = 262144 states
	1	0	Reserved
		1	Standby time = 16 states

Bit 3—Output Port Enable (OPE): Specifies whether the output of the address bus an control signals ($\overline{CS}_0$ to $\overline{CS}_7$, $\overline{AS}$, $\overline{RD}$, $\overline{HWR}$, $\overline{LWR}$, $\overline{CAS}$, $\overline{OE}$) is retained or set to the hi impedance state in software standby mode.

Bit 3

OPE	Description
0	In software standby mode, address bus and bus control signals are high-im
1	In software standby mode, address bus and bus control signals retain outpu

Bits 2 to 0—Reserved: Read-only bits, always read as 0.

speed mode control.

SCKCR is initialized to H'00 by a reset and in hardware standby mode. It is not initial
software standby mode.

Bit 7—ϕ Clock Output Disable (PSTOP): Controls ϕ output.

Bit 7	Description			
PSTOP	Normal Operating Mode	Sleep Mode	Software Standby Mode	Hardware Mode
0	ϕ output (initial value)	ϕ output	Fixed high	High impe
1	Fixed high	Fixed high	Fixed high	High impe

Bits 6—Reserved: This bit can be read or written to, but only 0 should be written.

Bits 5 to 3—Reserved: Read-only bits, always read as 0.

Bits 2 to 0—System Clock Select (SCK2 to SCK0): These bits select the clock for t
master.

Bit 2	Bit 1	Bit 0	
SCK2	SCK1	SCK0	Description
0	0	0	Bus master in high-speed mode
		1	Medium-speed clock is ϕ/2
	1	0	Medium-speed clock is ϕ/4
		1	Medium-speed clock is ϕ/8
1	0	0	Medium-speed clock is ϕ/16
		1	Medium-speed clock is ϕ/32
	1	—	—

MSTPCR is a 16-bit readable/writable register that performs module stop mode control

MSTPCR is initialized to H'3FFF by a reset and in hardware standby mode. It is not ini
software standby mode.

Bits 15 to 0—Module Stop (MSTP 15 to MSTP 0): These bits specify module stop m
table 21.3 for the method of selecting on-chip supporting modules.

Bits 15 to 0

MSTP15 to MSTP0	Description
0	Module stop mode cleared
1	Module stop mode set

21.3 Medium-Speed Mode

When the SCK2 to SCK0 bits in SCKCR are set to 1, the operating mode changes to m
speed mode as soon as the current bus cycle ends. In medium-speed mode, the CPU op
the operating clock (ϕ/2, ϕ/4, ϕ/8, ϕ/16, or ϕ/32) specified by the SCK2 to SCK0 bits. T
masters other than the CPU (the DMAC and DTC) also operate in medium-speed mode
supporting modules other than the bus masters always operate on the high-speed clock

In medium-speed mode, a bus access is executed in the specified number of states with
the bus master operating clock. For example, if ϕ/4 is selected as the operating clock, o
memory is accessed in 4 states, and internal I/O registers in 8 states.

Medium-speed mode is cleared by clearing all of bits SCK2 to SCK0 to 0. A transition
high-speed mode and medium-speed mode is cleared at the end of the current bus cycle

If a SLEEP instruction is executed when the SSBY bit in SBYCR is cleared to 0, a tran
made to sleep mode. When sleep mode is cleared by an interrupt, medium-speed mode

Figure 21.1 shows the timing for transition to and clearance of medium-speed mode.

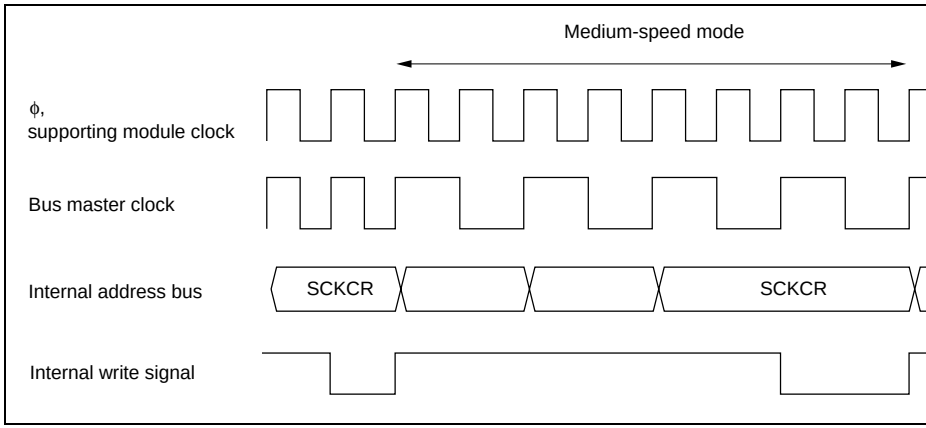


Figure 21.1 Medium-Speed Mode Transition and Clearance Timing

21.4 Sleep Mode

If a SLEEP instruction is executed when the SSBY bit in SBYCR is cleared to 0, the C sleep mode. In sleep mode, CPU operation stops but the contents of the CPU's interna are retained. Other supporting modules do not stop.

Sleep mode is cleared by a reset or any interrupt, and the CPU returns to the normal pr execution state via the exception handling state. Sleep mode is not cleared if interrupts disabled, or if interrupts other than NMI are masked by the CPU.

When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode.

independently.

Table 21.3 shows MSTP bits and the corresponding on-chip supporting modules.

When the corresponding MSTP bit is cleared to 0, module stop mode is cleared and th
starts operating at the end of the bus cycle. In module stop mode, the internal states of
other than the SCI are retained.

After reset clearance, all modules other than DMAC and DTC are in module stop mode

When an on-chip supporting module is in module stop mode, read/write access to its re
disabled.

	MSTP10	D/A converter
	MSTP9	A/D converter
	MSTP8	—
MSTPCRL	MSTP7	Serial communication interface (SCI) channel 2
	MSTP6	Serial communication interface (SCI) channel 1
	MSTP5	Serial communication interface (SCI) channel 0
	MSTP4	—
	MSTP3	—
	MSTP2	—
	MSTP1	—
	MSTP0	—

Note: Bit 8 and bits 4 to 0 can be read or written to, but do not affect operation.

For details, refer to section 7, DMA Controller, and section 8, Data Transfer Controller

On-Chip Supporting Module Interrupt

Relevant interrupt operations cannot be performed in module stop mode. Consequently stop mode is entered when an interrupt has been requested, it will not be possible to cle interrupt source or the DMAC or DTC activation source. Interrupts should therefore be before entering module stop mode.

Writing to MSTPCR

MSTPCR should only be written to by the CPU.

bus control signals are placed in the high-impedance state or retain the output state ca
specified by the OPE bit in SBYCR.

In this mode the oscillator stops, and therefore power dissipation is significantly reduc

21.6.2 Clearing Software Standby Mode

Software standby mode is cleared by an external interrupt (NMI pin, or pins $\overline{IRQ_0}$ to $\overline{I}$
means of the $\overline{RES}$ pin or $\overline{STBY}$ pin.

Clearing with an interrupt

When an NMI or IRQ_0 to IRQ_2 interrupt request signal is input, clock oscillation starts
the elapse of the time set in bits STS2 to STS0 in SYSCR, stable clocks are supplied t
H8S/2655 Group chip, software standby mode is cleared, and interrupt exception hand
started.

When clearing software standby mode with an IRQ_0 to IRQ_2 interrupt, set the correspo
enable bit to 1 and ensure that no interrupt with a higher priority than interrupts IRQ_0 t
generated. Software standby mode cannot be cleared if the interrupt has been masked
side or has been designated as a DTC activation source.

Clearing with the $\overline{RES}$ pin

When the $\overline{RES}$ pin is driven low, clock oscillation is started. At the same time as clock
starts, clocks are supplied to the entire H8S/2655 Group chip. Note that the $\overline{RES}$ pin m
low until clock oscillation stabilizes. When the $\overline{RES}$ pin goes high, the CPU begins re
handling.

Clearing with the $\overline{STBY}$ pin

When the $\overline{STBY}$ pin is driven low, a transition is made to hardware standby mode.

Table 21.4 shows the standby times for different operating frequencies and settings of t
STS0.

Table 21.4 Oscillation Stabilization Time Settings

STS2	STS1	STS0	Standby Time	20 MHz	16 MHz	12 MHz	10 MHz	8 MHz	6 MHz	4 MHz
0	0	0	8192 states	0.41	0.51	0.68	0.8	1.0	1.3	2.0
		1	16384 states	0.82	1.0	1.3	1.6	2.0	2.7	4.1
	1	0	32768 states	1.6	2.0	2.7	3.3	4.1	5.5	8.2
		1	65536 states	3.3	4.1	5.5	6.6	8.2	10.9	16.4
1	0	0	131072 states	6.6	8.2	10.9	13.1	16.4	21.8	32.8
		1	262144 states	13.1	16.4	21.8	26.2	32.8	43.6	65.6
	1	0	—*	—	—	—	—	—	—	—
		1	16 states	0.8	1.0	1.3	1.6	2.0	2.7	4.0

Legend:
—: Don't care
: Recommended time setting

Note: * Reserved. If set, the standby time will be 16 states.

Using an External Clock

Any value can be set. Normally, use of the minimum time is recommended.

Software standby mode is then cleared at the rising edge on the NMI pin.

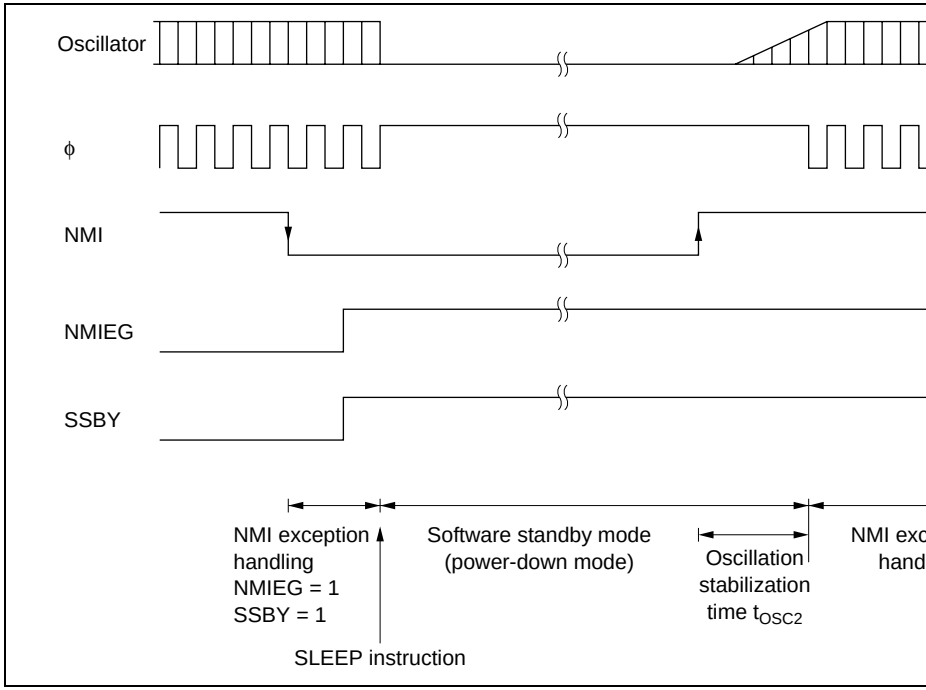


Figure 21.2 Software Standby Mode Application Example

Current Dissipation during Oscillation Stabilization Wait Period

Current dissipation increases during the oscillation stabilization wait period.

Write Data Buffer Function

The write data buffer function and software standby mode cannot be used at the same ti
the write data buffer function is used, the WDBE bit in BCRL should be cleared to 0 to
write data buffer function before entering software standby mode. Also check that exter
have finished, by reading external addresses, etc., before executing a SLEEP instructio
software standby mode. See section 6.10, Write Data Buffer Function, for details of the
buffer function.

21.7 Hardware Standby Mode

21.7.1 Hardware Standby Mode

When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode from

In hardware standby mode, all functions enter the reset state and stop operation, resulti
significant reduction in power dissipation. As long as the prescribed voltage is supplied
RAM data is retained. I/O ports are set to the high-impedance state.

In order to retain on-chip RAM data, the RAME bit in SYSCR should be cleared to 0 b
driving the $\overline{\text{STBY}}$ pin low.

Do not change the state of the mode pins (MD_2 to MD_0) while the H8S/2655 Group is i
standby mode.

Hardware standby mode is cleared by means of the $\overline{\text{STBY}}$ pin and the $\overline{\text{RES}}$ pin. When t
pin is driven high while the $\overline{\text{RES}}$ pin is low, the reset state is set and clock oscillation is
Ensure that the $\overline{\text{RES}}$ pin is held low until the clock oscillator stabilizes (at least 8 ms—t
oscillation stabilization time—when using a crystal oscillator). When the $\overline{\text{RES}}$ pin is su

hardware standby mode. Hardware standby mode is cleared by driving the STBY pin
for the oscillation stabilization time, then changing the $\overline{\text{RES}}$ pin from low to high.

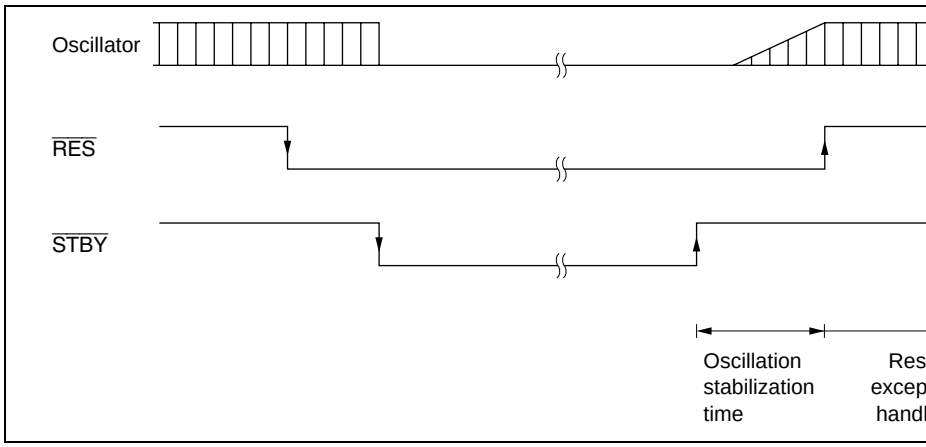


Figure 21.3 Hardware Standby Mode Timing (Example)

DDR	0	1	1
PSTOP	—	0	1
Hardware standby mode	High impedance	High impedance	High imp
Software standby mode	High impedance	Fixed high	Fixed hig
Sleep mode	High impedance	ϕ output	Fixed hig
Normal operating state	High impedance	ϕ output	Fixed hig

Item	Symbol	Value
Power supply voltage	V_{CC}	–0.3 to +7.0
Programming voltage	V_{PP}	–0.3 to +13.5
Input voltage (except port 4)	V_{in}	–0.3 to V_{CC} +0.3
Input voltage (port 4)	V_{in}	–0.3 to AV_{CC} +0.3
Reference voltage	V_{ref}	–0.3 to AV_{CC} +0.3
Analog power supply voltage	AV_{CC}	–0.3 to +7.0
Analog input voltage	V_{AN}	–0.3 to AV_{CC} +0.3
Operating temperature	T_{opr}	Regular specifications: –20 to +75
		Wide-range specifications: –40 to +85
Storage temperature	T_{stg}	–55 to +125

Caution: Permanent damage to the chip may result if absolute maximum rating are ex

specifications)

Item		Symbol	Min	Typ	Max	Unit	Test C
Schmitt trigger input voltage	Port 2, $P6_4$ to $P6_7$, PA_4 to PA_7	V_T^-	1.0	—	—	V	
		V_T^+	—	—	$V_{CC} \times 0.7$	V	
		$V_T^+ - V_T^-$	0.4	—	—	V	
Input high voltage	$\overline{RES}$, $\overline{STBY}$, NMI, MD_2 to MD_0	V_{IH}	V_{CC} –0.7	—	V_{CC} +0.3	V	
	EXTAL		$V_{CC} \times 0.7$	—	V_{CC} +0.3	V	
	Port 1, 3, 5, B to G, $P6_0$ to $P6_3$, PA_0 to PA_3		2.0	—	V_{CC} +0.3	V	
	Port4		2.0	—	AV_{CC} +0.3	V	
Input low voltage	$\overline{RES}$, $\overline{STBY}$, MD_2 to MD_0	V_{IL}	–0.3	—	0.5	V	
	NMI, EXTAL, Port 1, 3 to 5, B to G, $P6_0$ to $P6_3$, PA_0 to PA_3		–0.3	—	0.8	V	
Output high voltage	All output pins	V_{OH}	V_{CC} –0.5	—	—	V	I_{OH} = –2
			3.5	—	—	V	I_{OH} = –1
Output low voltage	All output pins	V_{OL}	—	—	0.4	V	I_{OL} = 1.6
	Port 1, A to C		—	—	1.0	V	I_{OL} = 10
Input leakage current	$\overline{RES}$	$\lvert I_{in} \rvert$	—	—	10.0	µA	V_{in} = 0.5 to V
	$\overline{STBY}$, NMI, MD_2 to MD_0		—	—	1.0	µA	
	Port 4		—	—	1.0	µA	V_{in} = 0.5 to A

	NMI		—	—	50	pF	T_a = 2
	All input pins except $\overline{RES}$ and NMI		—	—	15	pF	
Current dissipation*2	Normal operation	I_{CC}*4	—	80 (5.0 V)	122	mA	f = 20
	Sleep mode		—	60 (5.0 V)	84	mA	f = 20
	Standby mode*3		—	0.01	5.0	µA	$T_a \leq 5$
			—	—	20		50°C <
Analog power supply current	During A/D and D/A conversion	AI_{CC}	—	16 (5.0 V)	24	mA	
	Idle		—	0.01	5.0	µA	
Reference current	During A/D and D/A conversion	AI_{CC}	—	2	3.0	mA	
	Idle		—	0.01	5.0	µA	
RAM standby voltage		V_{RAM}	2.0	—	—	V	

Notes: 1. If the A/D and D/A converters are not used, do not leave the AV_{CC}, AV_{SS}, an open.
Connect AV_{CC} and V_{ref} to V_{CC}, and connect AV_{SS} to V_{SS}.
2. Current dissipation values are for V_{IH} min = V_{CC} –0.5 V and V_{IL} max = 0.5 V output pins unloaded and the on-chip pull-up transistors in the off state.
3. The values are for $V_{RAM} \leq V_{CC} < 4.5V$, V_{IH} min = $V_{CC} \times 0.9$, and V_{IL} max = 0.3
4. I_{CC} depends on V_{CC} and f as follows:
I_{CC} max = 1.0 (mA) + 1.1 (mA/(MHz × V)) × V_{CC} × f [normal mode]
I_{CC} max = 1.0 (mA) + 0.75 (mA/(MHz × V)) × V_{CC} × f [sleep mode]

voltage	PA_4 to PA_7						
		$V_T^+ - V_T^-$	$V_{CC} \times 0.07$	—	—	V	
Input high voltage	$\overline{RES}$, $\overline{STBY}$, NMI, MD_2 to MD_0	V_{IH}	$V_{CC} \times 0.9$	—	V_{CC} +0.3	V	
	EXTAL		$V_{CC} \times 0.7$	—	V_{CC} +0.3	V	
	Port 1, 3, 5, B to G, $P6_0$ to $P6_3$, PA_0 to PA_3		$V_{CC} \times 0.7$	—	V_{CC} +0.3	V	
	Port 4		$V_{CC} \times 0.7$	—	AV_{CC} +0.3	V	
Input low voltage	$\overline{RES}$, $\overline{STBY}$, MD_2 to MD_0	V_{IL}	–0.3	—	$V_{CC} \times 0.1$	V	
	NMI, EXTAL, Port 1, 3 to 5, B to G, $P6_0$ to $P6_3$, PA_0 to PA_3		–0.3	—	$V_{CC} \times 0.2$	V	V_{CC} < 4.
					0.8		V_{CC} = 4.
Output high voltage	All output pins	V_{OH}	V_{CC} –0.5	—	—	V	I_{OH} = –2
			V_{CC} –1.0	—	—	V	I_{OH} = –1
Output low voltage	All output pins	V_{OL}	—	—	0.4	V	I_{OL} = 1.(
	Port 1, A to C		—	—	1.0	V	$V_{CC} \le 4$ I_{OL} = 5 m 4.0 < V I_{OL} = 10
Input leakage current	$\overline{RES}$	$\lvert I_{in} \rvert$	—	—	10.0	µA	V_{in} = 0.5 to V
	$\overline{STBY}$, NMI, MD_2 to MD_0		—	—	1.0	µA	
	Port 4		—	—	1.0	µA	V_{in} = 0.5 to A

	NMI		—	—	50	pF	Ta = 2
	All input pins except $\overline{RES}$ and NMI		—	—	15	pF	
Current dissipation*2	Normal operation	I_{CC}*4	—	25 (3.0 V)	62	mA	f = 10
	Sleep mode		—	18 (3.0 V)	42	mA	f = 10
	Standby mode*3		—	0.01	5.0	µA	$T_a \leq 5$
			—	—	20		50°C <
Analog power supply current	During A/D and D/A conversion	AI_{CC}	—	12 (3.0 V)	22	mA	
	Idle		—	0.01	5.0	µA	
Reference current	During A/D and D/A conversion	AI_{CC}	—	1.5 (3.0 V)	2.5	mA	
	Idle		—	0.01	5.0	µA	
RAM standby voltage		V_{RAM}	2.0	—	—	V	

Notes: 1. If the A/D and D/A converters are not used,do not leave the AV_{CC}, AV_{SS}, and open.
Connect AV_{CC} and V_{ref} to V_{CC}, and connect AV_{SS} to V_{SS}.
2. Current dissipation values are for V_{IH} min = V_{CC} –0.5 V and V_{IL} max = 0.5 V output pins unloaded and the on-chip pull-up transistors in the off state.
3. The values are for $V_{RAM} \leq V_{CC} < 2.7$ V, V_{IH} min = $V_{CC} \times 0.9$, and V_{IL} max = 0.3
4. I_{CC} depends on V_{CC} and f as follows:
I_{CC} max = 1.0 (mA) + 1.1 (mA/(MHz × V)) × V_{CC} × f [normal mode]
I_{CC} max = 1.0 (mA) + 0.75 (mA/(MHz × V)) × V_{CC} × f [sleep mode]

Permissible output low current (total)	Total of 32 pins including port 1 and A to C	$\sum I_{OL}$	—	—	80
	Total of all output pins, including the above		—	—	120
Permissible output high current (per pin)	All output pins	$-I_{OH}$	—	—	2.0
Permissible output high current (total)	Total of all output pins	$\sum -I_{OH}$	—	—	40

Notes: 1. To protect chip reliability, do not exceed the output current values in table 22

2. When driving a darlington pair or LED directly, always insert a current-limitin the output line, as show in figures 22.1 and 22.2.

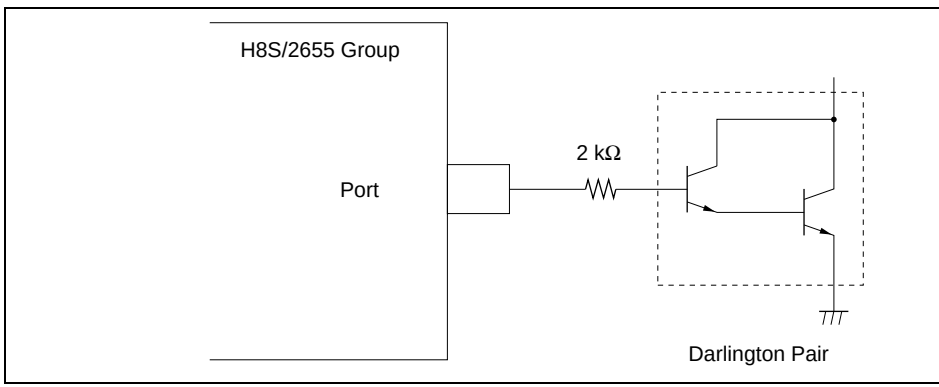


Figure 22.1 Darlington Pair Drive Circuit (Example)

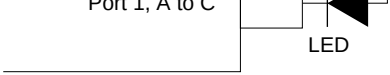


Figure 22.2 LED Drive Circuit (Example)

22.3 AC Characteristics

Figure 22.3 show, the test conditions for the AC characteristics.

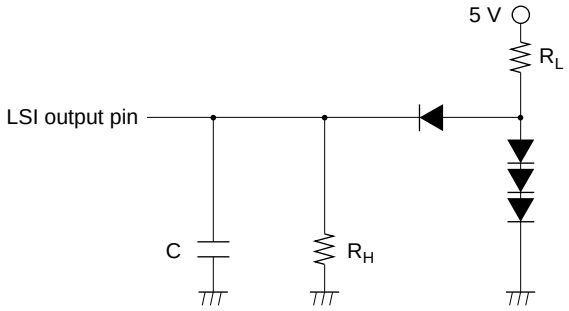


Figure 22.3 Output Load Circuit

Condition B: V_{CC} = 5.0 V ±10%, AV_{CC} = 5.0 V ±10%, V_{ref} = 4.5 V to AV_{CC},
V_{SS} = AV_{SS} = 0 V, φ = 2 to 20 MHz, T_a = –20 to +75°C (regular specifica
T_a = –40 to +85°C (wide-range specifications)

		Condition A		Condition B			
Item	**Symbol**	**Min**	**Max**	**Min**	**Max**	**Unit**	**Test Co**
Clock cycle time	t_{cyc}	100	500	50	500	ns	Figure 2
Clock high pulse width	t_{CH}	35	—	20	—	ns	Figure 2
Clock low pulse width	t_{CL}	35	—	20	—	ns	
Clock rise time	t_{Cr}	—	15	—	5	ns	
Clock fall time	t_{Cf}	—	15	—	5	ns	
Clock oscillator setting time at reset (crystal)	t_{OSC1}	20	—	10	—	ms	Figure 2
Clock oscillator setting time in software standby (crystal)	t_{OSC2}	20	—	10	—	ms	Figure 2
External clock output stabilization delay time	t_{DEXT}	500	—	500	—	µs	Figure 2

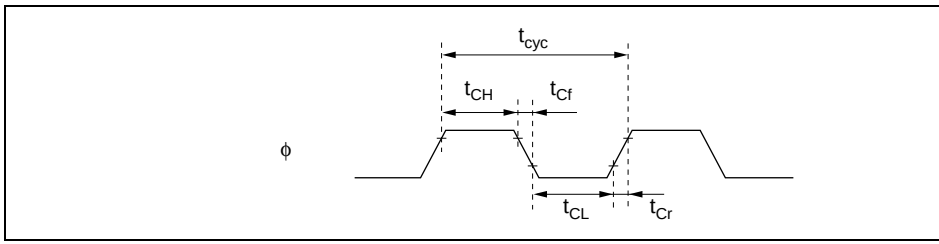


Figure 22.4 System Clock Timing

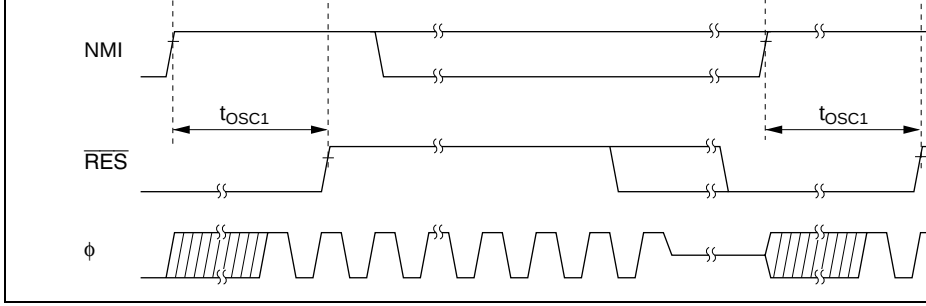


Figure 22.5 Oscillator Settling Timing

Condition B: V_{CC} = 5.0 V ±10%, AV_{CC} = 5.0 V ±10%, V_{ref} = 4.5 V to AV_{CC},
V_{SS} = AV_{SS} = 0 V, ϕ = 2 to 20 MHz, T_a = –20 to +75°C (regular specifica
T_a = –40 to +85°C (wide-range specifications)

		Condition A		Condition B			
Item	**Symbol**	**Min**	**Max**	**Min**	**Max**	**Unit**	**Test Co**
$\overline{RES}$ setup time	t_{RESS}	200	—	200	—	ns	Figure 2
$\overline{RES}$ pulse width	t_{RESW}	20	—	20	—	t_{cyc}	
NMI reset setup time	t_{NMIRS}	250	—	200	—	ns	
NMI reset hold time	t_{NMIRH}	200	—	200	—		
NMI setup time	t_{NMIS}	250	—	150	—	ns	Figure 2
NMI hold time	t_{NMIH}	10	—	10	—		
NMI pulse width (exiting software standby mode)	t_{NMIW}	200	—	200	—	ns	
$\overline{IRQ}$ setup time	t_{IRQS}	250	—	150	—	ns	
$\overline{IRQ}$ hold time	t_{IRQH}	10	—	10	—	ns	
$\overline{IRQ}$ pulse width (exiting software standby mode)	t_{IRQW}	200	—	200	—	ns	

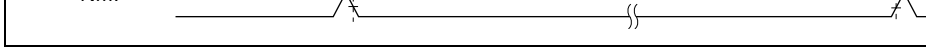


Figure 22.6 Reset Input Timing

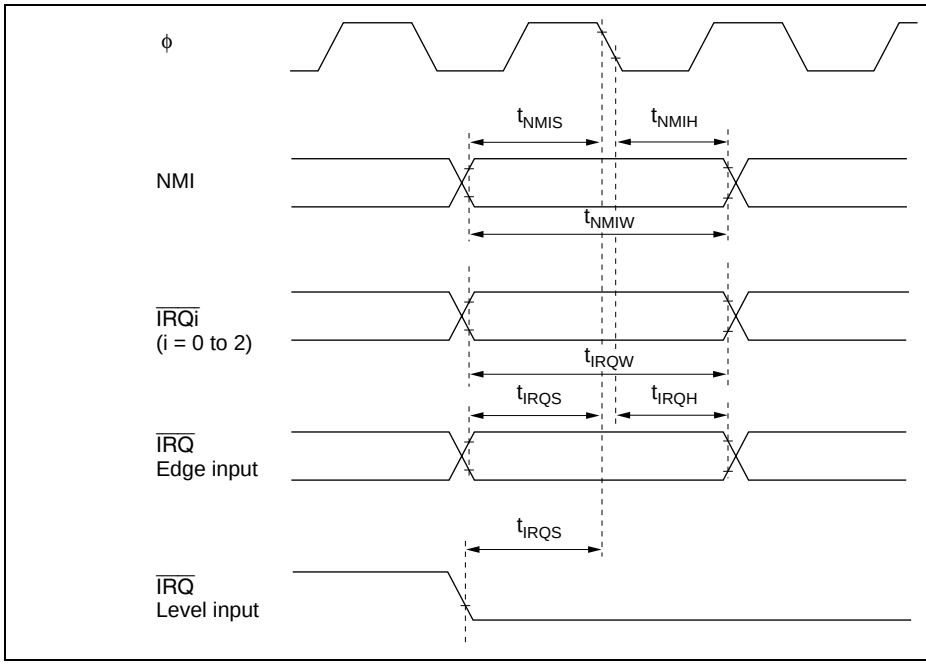


Figure 22.7 Interrupt Input Timing

Condition B: V_{CC} = 5.0 V ±10%, AV_{CC} = 5.0 V ±10%, V_{ref} = 4.5 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ϕ = 2 to 20 MHz, T_a = –20 to +75°C (regular specifica
T_a = –40 to +85°C (wide-range specifications)

		Condition A		Condition B			Test
Item	**Symbol**	**Min**	**Max**	**Min**	**Max**	**Unit**	**Cond**
Address delay time	t_{AD}	—	40	—	20	ns	Figur
Address setup time	t_{AS}	0.5 × t_{cyc} –30	—	0.5 × t_{cyc} –15	—	ns	Figur
Address hold time	t_{AH}	0.5 × t_{cyc} –20	—	0.5 × t_{cyc} –10	—	ns	
Precharge time	t_{PCH}	1.5 × t_{cyc} –40	—	1.5 × t_{cyc} –20	—	ns	
$\overline{CS}$ delay time 1	t_{CSD1}	—	40	—	20	ns	
$\overline{CS}$ delay time 2	t_{CSD2}	—	40	—	20	ns	
$\overline{CS}$ pulse width	t_{CSW}	2.5 × t_{cyc} –40	—	2.5 × t_{cyc} –20	—	ns	
$\overline{AS}$ delay time	t_{ASD}	—	40	—	20	ns	
$\overline{RD}$ delay time 1	t_{RSD1}	—	40	—	20	ns	
$\overline{RD}$ delay time 2	t_{RSD2}	—	40	—	20	ns	
$\overline{CAS}$ delay time	t_{CASD}	—	40	—	20	ns	
Read data setup time	t_{RDS}	30	—	15	—	ns	
Read data hold time	t_{RDH}	0	—	0	—	ns	
Read data access time1	t_{ACC1}	—	1.0 × t_{cyc} –50	—	1.0 × t_{cyc} –25	ns	
Read data access time2	t_{ACC2}	—	1.5 × t_{cyc} –50	—	1.5 × t_{cyc} –25	ns	

$\overline{WR}$ delay time 1	t_{WRD1}	—	40	—	20	ns	
$\overline{WR}$ delay time 2	t_{WRD2}	—	40	—	20	ns	
$\overline{WR}$ pulse width 1	t_{WSW1}	1.0 × t_{cyc} –40	—	1.0 × t_{cyc} –20	—	ns	
$\overline{WR}$ pulse width 2	t_{WSW2}	1.5 × t_{cyc} –40	—	1.5 × t_{cyc} –20	—	ns	
Write data delay time	t_{WDD}	—	60	—	30	ns	
Write data setup time	t_{WDS}	0.5 × t_{cyc} –40	—	0.5 × t_{cyc} –20	—	ns	
Write data hold time	t_{WDH}	0.5 × t_{cyc} –20	—	0.5 × t_{cyc} –10	—	ns	
$\overline{WR}$ setup time	t_{WCS}	0.5 × t_{cyc} –20	—	0.5 × t_{cyc} –10	—	ns	
$\overline{WR}$ hold time	t_{WCH}	0.5 × t_{cyc} –20	—	0.5 × t_{cyc} –10	—	ns	
$\overline{CAS}$ setup time	t_{CSR}	0.5 × t_{cyc} –20	—	0.5 × t_{cyc} –10	—	ns	Figu
$\overline{WAIT}$ setup time	t_{WTS}	60	—	30	—	ns	Figu
$\overline{WAIT}$ hold time	t_{WTH}	10	—	5	—	ns	
$\overline{BREQ}$ setup time	t_{BRQS}	60	—	30	—	ns	Figu
$\overline{BACK}$ delay time	t_{BACD}	—	30	—	15	ns	
Bus-floating time	t_{BZD}	—	100	—	50	ns	
$\overline{BREQO}$ delay time	t_{BRQOD}	—	60	—	30	ns	Figu

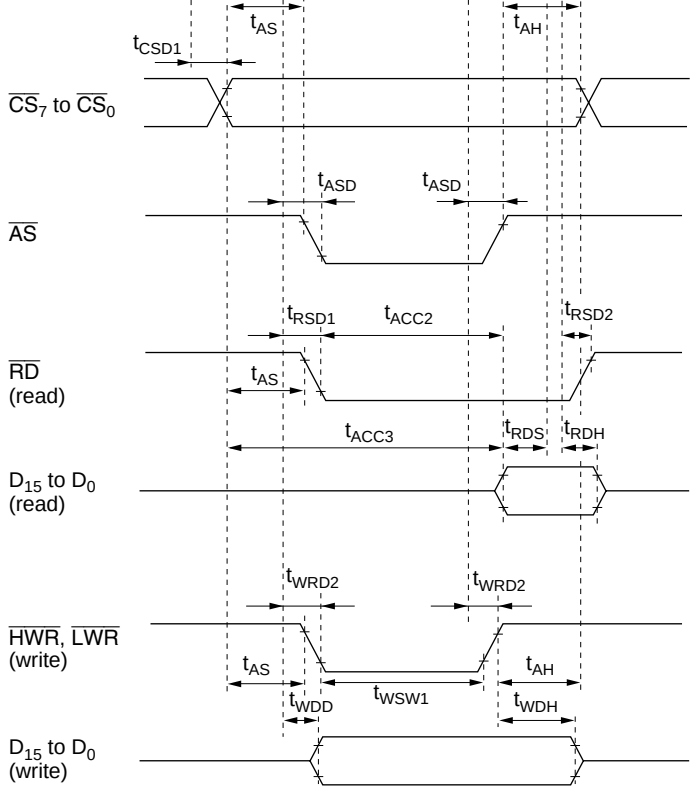


Figure 22.8 Basic Bus Timing (Two-State Access)

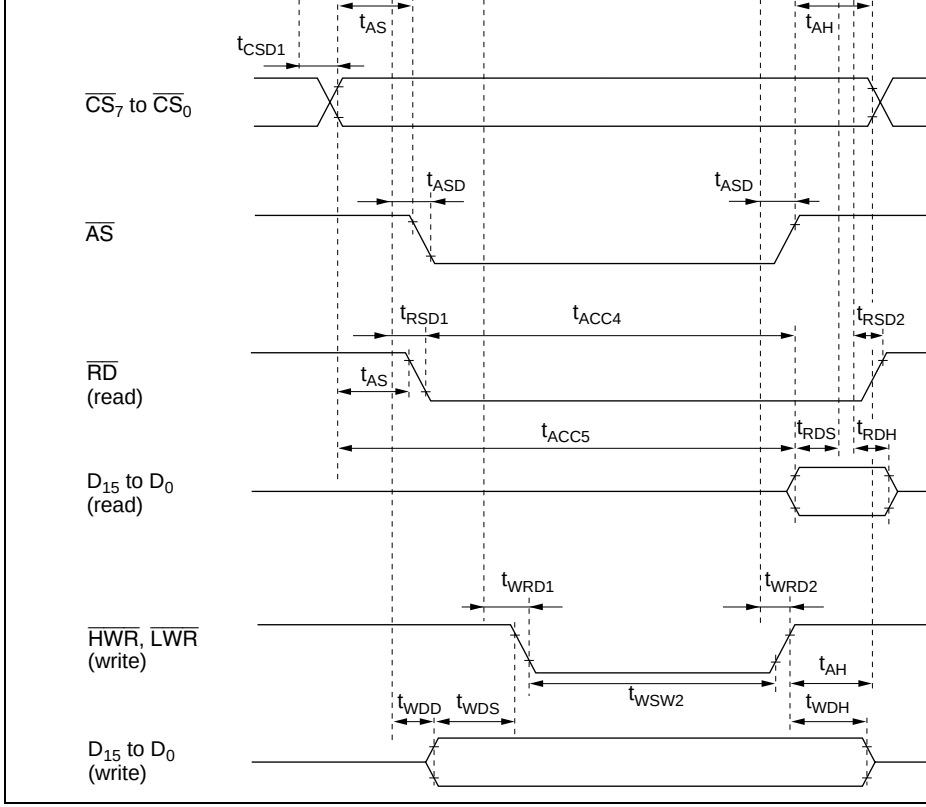


Figure 22.9 Basic Bus Timing (Three-State Access)

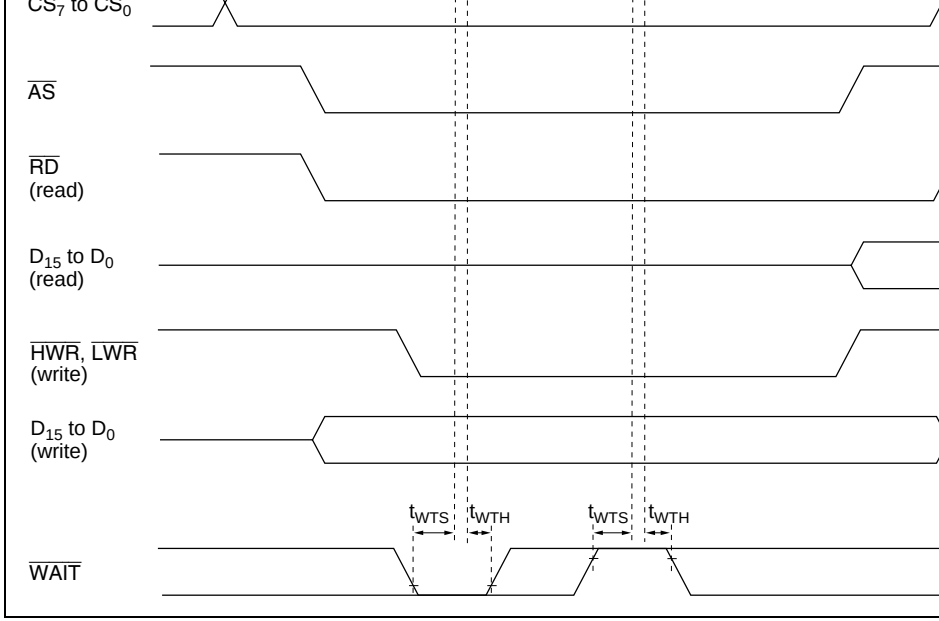


Figure 22.10 Basic Bus Timing (Three-State Access with One Wait Stat

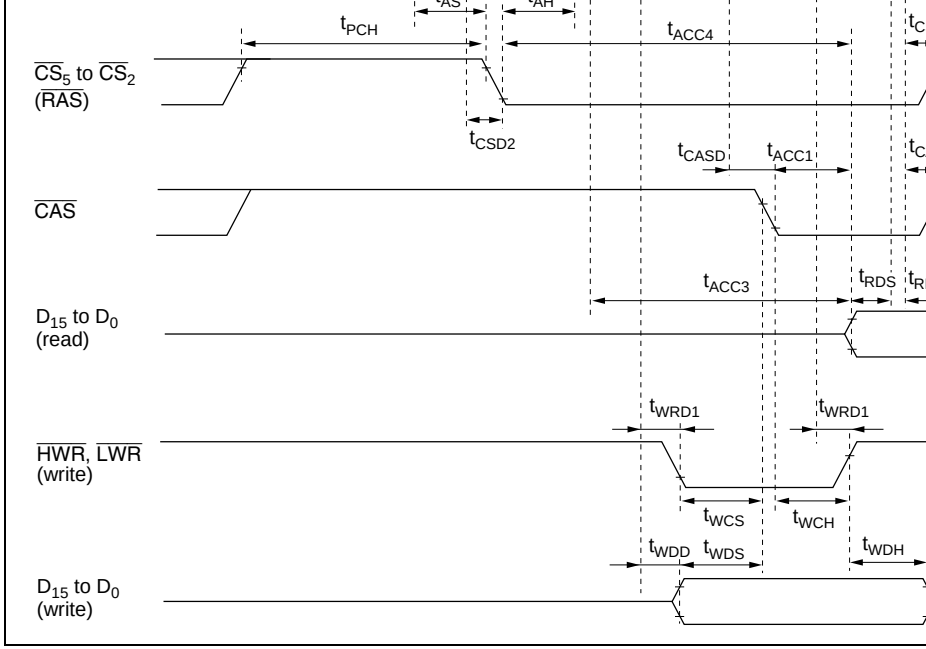


Figure 22.11 DRAM Bus Timing

Figure 22.12 CAS-Before-RAS Refresh Timing

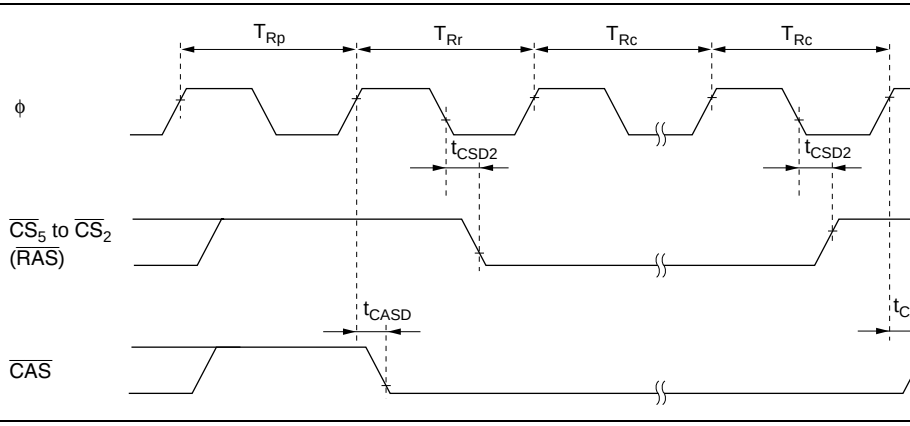


Figure 22.13 Self-Refresh Timing

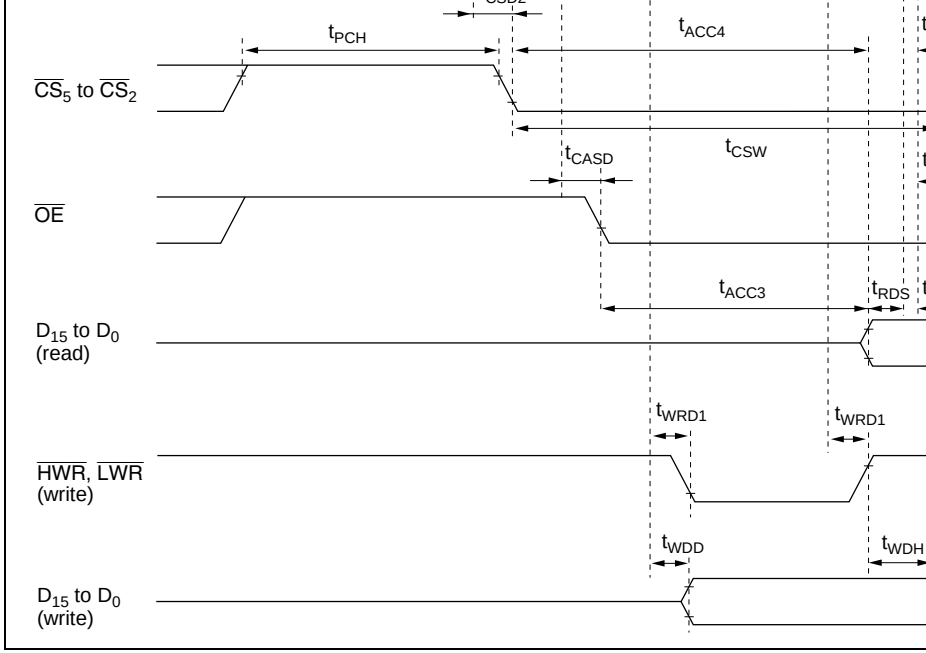


Figure 22.14 PSRAM Bus Timing

Figure 22.15 Auto Refresh Timing

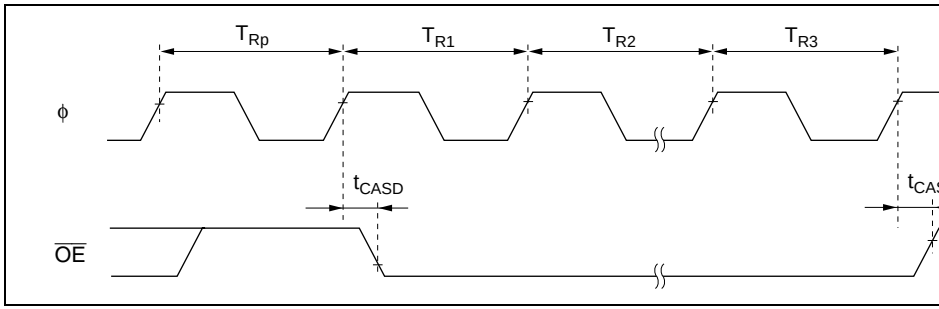


Figure 22.16 Self-Refresh Timing

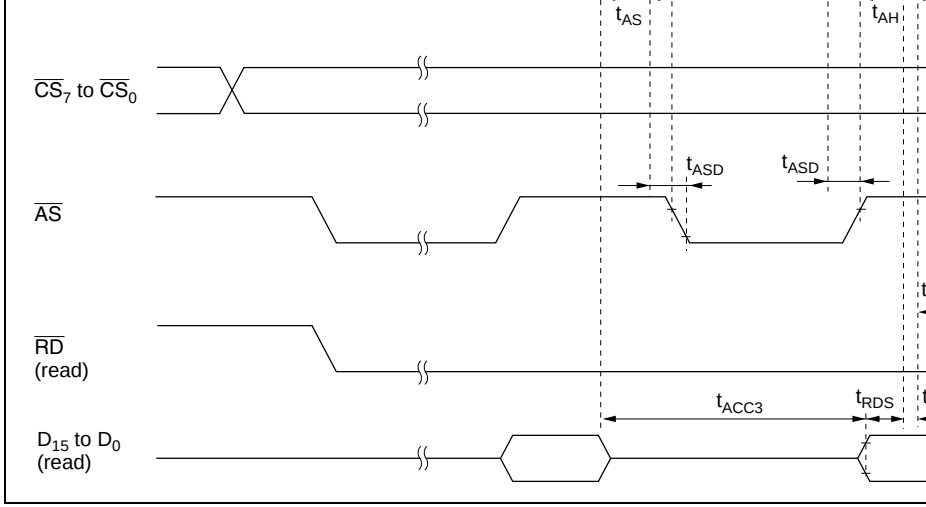


Figure 22.17 Burst ROM Access Timing (Two-State Access)

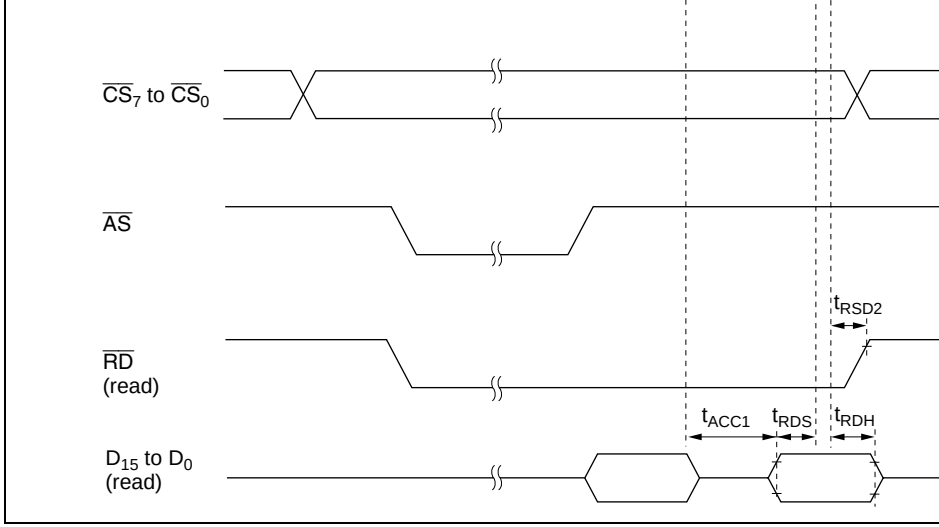


Figure 22.18 Burst ROM Access Timing (One-State Access)

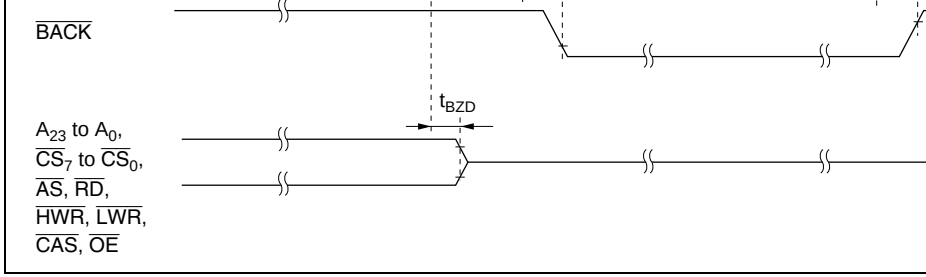


Figure 22.19 External Bus Release Timing

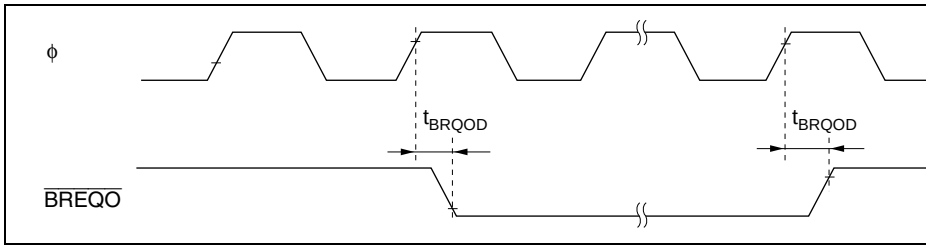


Figure 22.20 External Bus Request Output Timing

Condition B: V_{CC} = 5.0 V ±10%, AV_{CC} = 5.0 V ±10%, V_{ref} = 4.5 V to AV_{CC},
V_{SS} = AV_{SS} = 0 V, ϕ = 2 to 20 MHz, T_a = –20 to +75°C (regular specifica
T_a = –40 to +85°C (wide-range specifications)

		Condition A		Condition B			
Item	**Symbol**	**Min**	**Max**	**Min**	**Max**	**Unit**	**Test Co**
$\overline{\text{DREQ}}$ setup time	t_{DRQS}	40	—	30	—	ns	Figure 2
$\overline{\text{DREQ}}$ hold time	t_{DRQH}	10	—	10	—		
$\overline{\text{TEND}}$ delay time	t_{TED}	—	40	—	20		Figure 2
$\overline{\text{DACK}}$ delay time 1	t_{DACD1}	—	40	—	20	ns	Figure 2
$\overline{\text{DACK}}$ delay time 2	t_{DACD2}	—	40	—	20		Figure 2

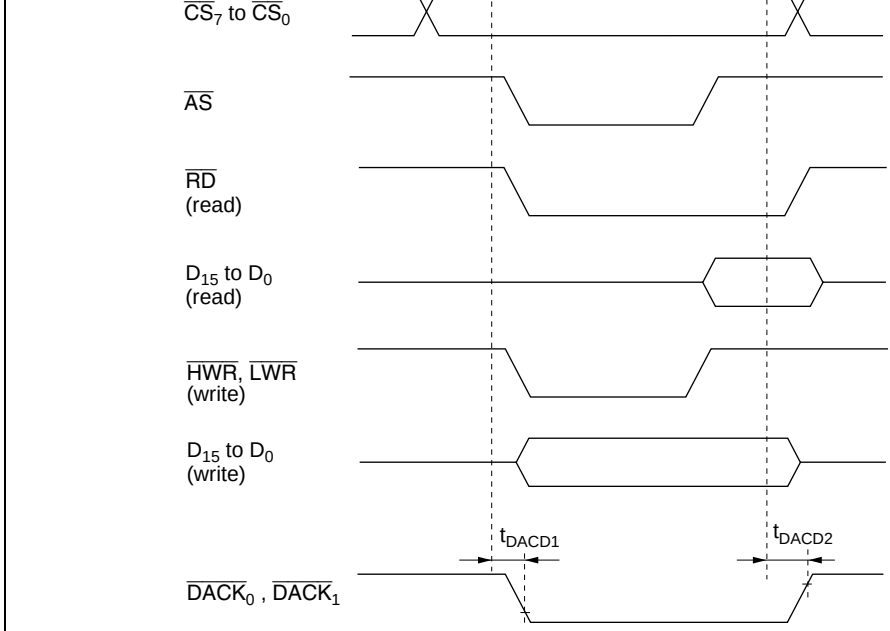


Figure 22.21 DMAC Single Address Transfer Timing (Two-State Acce

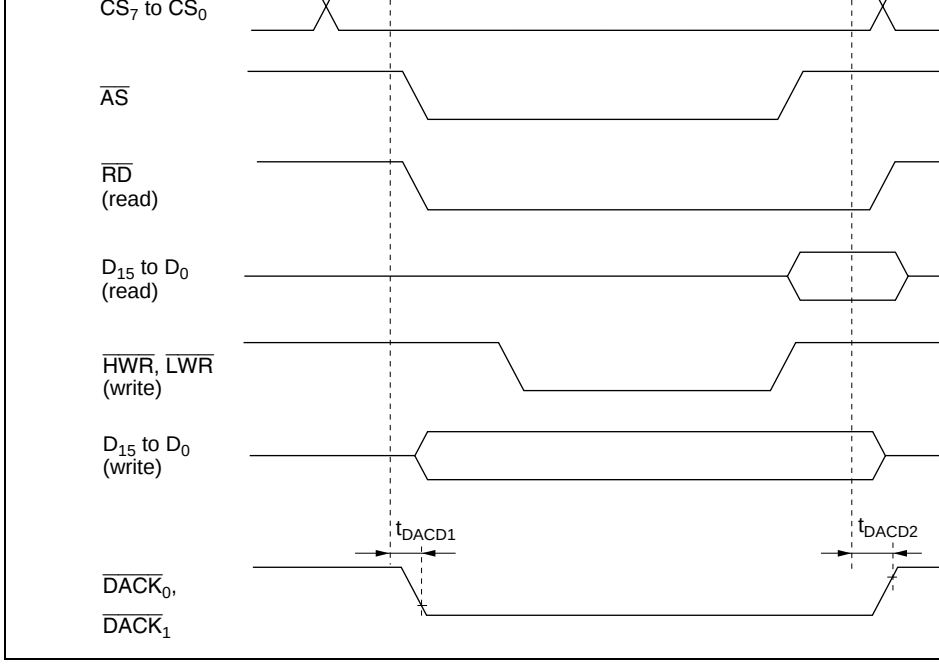


Figure 22.22 DMAC Single Address Transfer Timing (Three-State Acce

Figure 22.23 DMAC $\overline{\text{TEND}}$ Output Timing

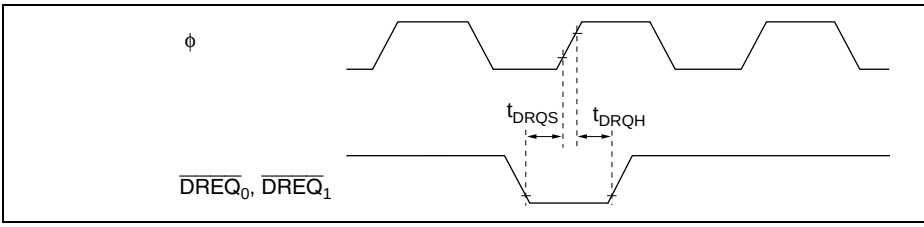


Figure 22.24 DMAC $\overline{\text{DREQ}}$ Intput Timing

Condition B: V_{CC} = 5.0 V ±10%, AV_{CC} = 5.0 V ±10%, V_{ref} = 4.5 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ϕ = 2 to 20 MHz, T_a = –20 to +75°C (regular specifica
T_a = –40 to +85°C (wide-range specifications)

				Condition A		Condition B			
Item			**Symbol**	**Min**	**Max**	**Min**	**Max**	**Unit**	**Test C**
I/O PORTS	Output data delay time		t_{PWD}	—	100	—	50	ns	Figure
	Input data setup time		t_{PRS}	50	—	30	—		
	Input data hold time		t_{PRH}	50	—	30	—		
PPG	Pulse output delay time		t_{POD}	—	100	—	50	ns	Figure
TPU	Timer output delay time		t_{TOCD}	—	100	—	50	ns	Figure
	Timer input setup time		t_{TICS}	50	—	30	—		
	Timer clock input setup time		t_{TCKS}	50	—	30	—	ns	Figure
	Timer clock pulse width	Single edge	t_{TCKWH}	1.5	—	1.5	—	t_{cyc}	
		Both edges	t_{TCKWL}	2.5	—	2.5	—		

	Timer clock pulse width	Single edge	t_{TMCWH}	1.5	—	1.5	—	t_{cyc}	Figu
		Both edges	t_{TMCWL}	2.5	—	2.5	—		
WDT	Overflow output delay time		t_{WOVD}	—	100	—	50	ns	Figu
SCI	Input clock cycle	Asynchro-nous	t_{Scyc}	4	—	4	—	t_{cyc}	Figu
		Synchro-nous		6	—	6	—		
	Input clock pulse width		t_{SCKW}	0.4	0.6	0.4	0.6	t_{Scyc}	
	Input clock rise time		t_{SCKr}	—	1.5	—	1.5	t_{cyc}	
	Input clock fall time		t_{SCKf}	—	1.5	—	1.5		
	Transmit data delay time		t_{TXD}	—	100	—	50	ns	Figu
	Receive data setup time (synchronous)		t_{RXS}	100	—	50	—	ns	
	Receive data hold time (synchronous)		t_{RXH}	100	—	50	—	ns	
A/D converter	Trigger input setup time		t_{TRGS}	50	—	30	—	ns	Figu

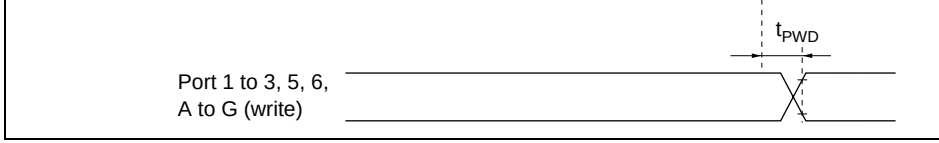


Figure 22.25 I/O Port Input/Output Timing

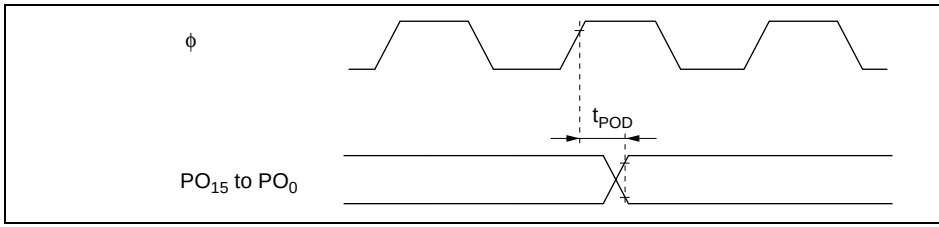


Figure 22.26 PPG Output Timing

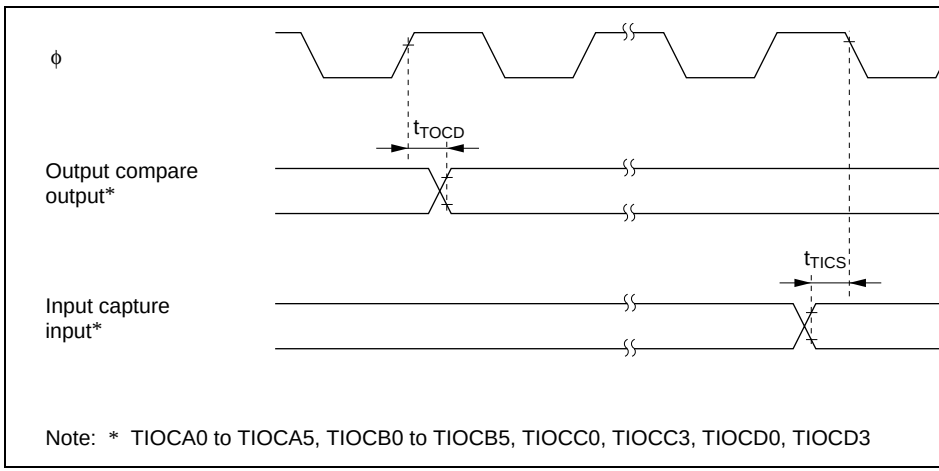


Figure 22.27 TPU Input/Output Timing

Figure 22.28 TPU Clock Input Timing

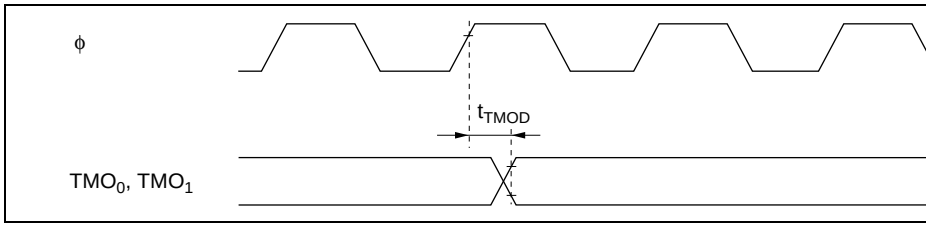


Figure 22.29 8-Bit Timer Output Timing

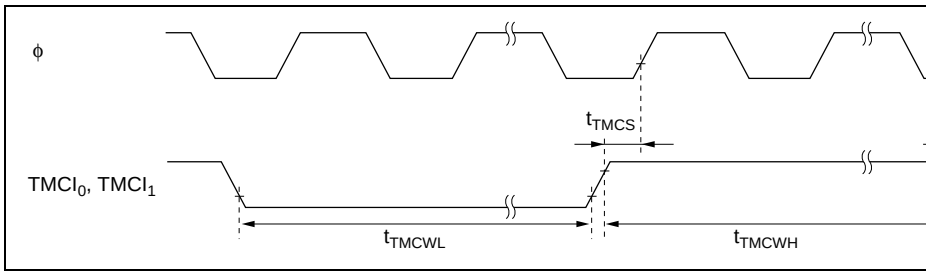


Figure 22.30 8-Bit Timer Clock Input Timing

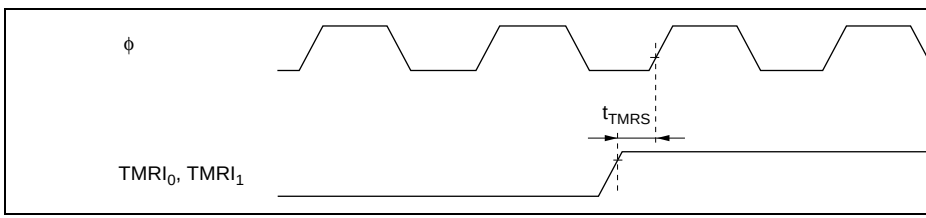


Figure 22.31 8-Bit Timer Reset Input Timing

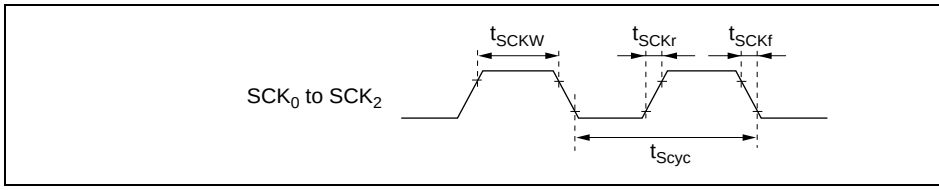


Figure 22.33 SCK Clock Input Timing

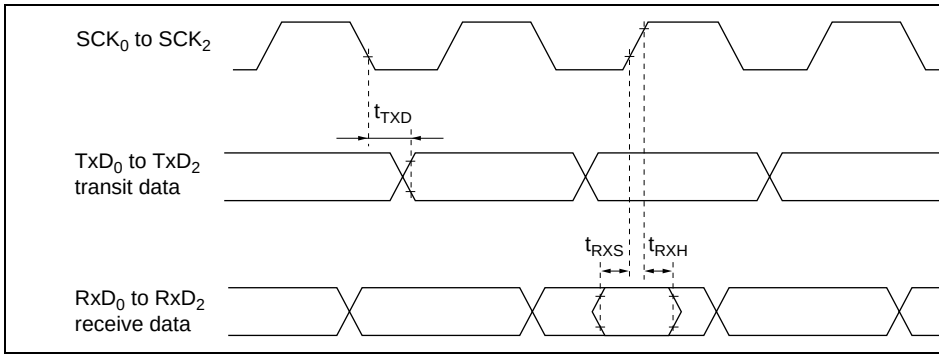


Figure 22.34 SCI Input/Output Timing (Clock Synchronous Mode)

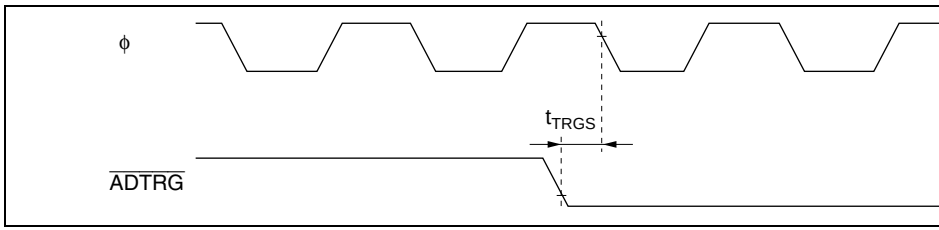


Figure 22.35 A/D Converter External Trigger Input Timing

Condition B: V_{CC} = AV_{CC} = 5.0 V ±10%, V_{ref} = 4.5 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ϕ = 2 to 20 MHz, T_a = –20 to +75°C (regular specifications), T_a = –40 to +85°C (wide-range specifications)

	Condition A			Condition B		
Item	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**
Resolution	10	10	10	10	10	10
Conversion time	—	—	44	—	—	44
Analog input capacitance	—	—	20	—	—	20
Permissible signal-source impedance	—	—	5	—	—	5
Nonlinearity error	—	—	±12.0	—	—	±8.0
Offset error	—	—	±12.0	—	—	±8.0
Full-scale error	—	—	±12.0	—	—	±8.0
Quantization	—	—	±0.5	—	—	±0.5
Absolute accuracy	—	—	±12.0	—	—	±8.0

Condition B: V_{CC} = AV_{CC} = 5.0 V ±10%, V_{ref} = 4.5 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V,
ϕ = 2 to 20 MHz, T_a = –20 to +75°C (regular specifications),
T_a = –40 to +85°C (wide-range specifications)

	Condition A			Condition B				
Item	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Unit**	**Test Co**
Resolution	8	8	8	8	8	8	bit	
Conversion time	—	—	10	—	—	10	µs	20-pF c load
Absolute accuracy	—	±2.0	±3.0	—	±1.0	±1.5	LSB	2-MΩ re load
	—	—	±2.0	—	—	±1.0	LSB	4-MΩ re load

22.6 Usage Notes

Although both the ZTAT and mask ROM versions fully meet the electrical specificatio this manual, due to differences in the fabrication process, the on-chip ROM, and the lay patterns, there will be differences in the actual values of the electrical characteristics, th margins, the noise margins, and other aspects.

Therefore, if a system is evaluated using the ZTAT version, a similar evaluation should performed using the mask ROM version.

Rs	General register (source)
Rn	General register*
ERn	General register (32-bit register)
MAC	Multiply-and-accumulate register (32-bit register)
(EAd)	Destination operand
(EAs)	Source operand
EXR	Extended control register
CCR	Condition-code register
N	N (negative) flag in CCR
Z	Z (zero) flag in CCR
V	V (overflow) flag in CCR
C	C (carry) flag in CCR
PC	Program counter
SP	Stack pointer
#IMM	Immediate data
disp	Displacement
+	Add
–	Subtract
×	Multiply
÷	Divide
∧	Logical AND
∨	Logical OR
⊕	Logical exclusive OR
→	Transfer from the operand on the left to the operand on the ri transition from the state on the left to the state on the right
¬	Logical NOT (logical complement)
( ) < >	Contents of operand
:8/:16/:24/:32	8-, 16-, 24-, or 32-bit length

Note: * General registers include 8-bit registers (R0H to R7H, R0L to R7L), 16-bit r
to R7, E0 to E7), and 32-bit registers (ER0 to ER7).

	Mnemonic	Operand Size	Addressing Mode/ Instruction Length (Bytes)								Operation	Condition Code				
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa		I	H	N	Z	V	C
MOV	MOV.B #xx:8,Rd	B	2								#xx:8→Rd8	—	—	↕	↕	0	—
	MOV.B Rs,Rd	B		2							Rs8→Rd8	—	—	↕	↕	0	—
	MOV.B @ERs,Rd	B			2						@ERs→Rd8	—	—	↕	↕	0	—
	MOV.B @(d:16,ERs),Rd	B				4					@(d:16,ERs)→Rd8	—	—	↕	↕	0	—
	MOV.B @(d:32,ERs),Rd	B				8					@(d:32,ERs)→Rd8	—	—	↕	↕	0	—
	MOV.B @ERs+,Rd	B					2				@ERs→Rd8,ERs32+1→ERs32	—	—	↕	↕	0	—
	MOV.B @aa:8,Rd	B						2			@aa:8→Rd8	—	—	↕	↕	0	—
	MOV.B @aa:16,Rd	B						4			@aa:16→Rd8	—	—	↕	↕	0	—
	MOV.B @aa:32,Rd	B						6			@aa:32→Rd8	—	—	↕	↕	0	—
	MOV.B Rs,@ERd	B			2						Rs8→@ERd	—	—	↕	↕	0	—
	MOV.B Rs,@(d:16,ERd)	B				4					Rs8→@(d:16,ERd)	—	—	↕	↕	0	—
	MOV.B Rs,@(d:32,ERd)	B				8					Rs8→@(d:32,ERd)	—	—	↕	↕	0	—
	MOV.B Rs,@-ERd	B					2				ERd32-1→ERd32,Rs8→@ERd	—	—	↕	↕	0	—
	MOV.B Rs,@aa:8	B						2			Rs8→@aa:8	—	—	↕	↕	0	—
	MOV.B Rs,@aa:16	B						4			Rs8→@aa:16	—	—	↕	↕	0	—
	MOV.B Rs,@aa:32	B						6			Rs8→@aa:32	—	—	↕	↕	0	—
	MOV.W #xx:16,Rd	W	4								#xx:16→Rd16	—	—	↕	↕	0	—
	MOV.W Rs,Rd	W		2							Rs16→Rd16	—	—	↕	↕	0	—
	MOV.W @ERs,Rd	W			2						@ERs→Rd16	—	—	↕	↕	0	—

RENESAS

Mnemonic	Operand Size	Addressing Mode/ Instruction Length (Bytes)								Operation	Condition Code		
		#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa		I	H	N
MOV	MOV.W @(d:16,ERs),Rd	W			4					@(d:16,ERs)→Rd16	—	—	↕
	MOV.W @(d:32,ERs),Rd	W			8					@(d:32,ERs)→Rd16	—	—	↕
	MOV.W @ERs+,Rd	W				2				@ERs→Rd16,ERs32+2→ERs32	—	—	↕
	MOV.W @aa:16,Rd	W					4			@aa:16→Rd16	—	—	↕
	MOV.W @aa:32,Rd	W					6			@aa:32→Rd16	—	—	↕
	MOV.W Rs,@ERd	W		2						Rs16→@ERd	—	—	↕
	MOV.W Rs,@(d:16,ERd)	W			4					Rs16→@(d:16,ERd)	—	—	↕
	MOV.W Rs,@(d:32,ERd)	W			8					Rs16→@(d:32,ERd)	—	—	↕
	MOV.W Rs,@-ERd	W				2				ERd32-2→ERd32,Rs16→@ERd	—	—	↕
	MOV.W Rs,@aa:16	W					4			Rs16→@aa:16	—	—	↕
	MOV.W Rs,@aa:32	W					6			Rs16→@aa:32	—	—	↕
	MOV.L #xx:32,ERd	L	6							#xx:32→ERd32	—	—	↕
	MOV.L ERs,ERd	L		2						ERs32→ERd32	—	—	↕
	MOV.L @ERs,ERd	L			4					@ERs→ERd32	—	—	↕
	MOV.L @(d:16,ERs),ERd	L				6				@(d:16,ERs)→ERd32	—	—	↕
	MOV.L @(d:32,ERs),ERd	L				10				@(d:32,ERs)→ERd32	—	—	↕
	MOV.L @ERs+,ERd	L					4			@ERs→ERd32,ERs32+4→ERs32	—	—	↕
	MOV.L @aa:16,ERd	L						6		@aa:16→ERd32	—	—	↕
	MOV.L @aa:32,ERd	L						8		@aa:32→ERd32	—	—	↕

Mnemonic	Operand Size	Addressing Mode/ Instruction Length (Bytes)								Operation	Condition Code					
		#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa		I	H	N	Z	V	C
MOV	MOV.L ERs,@ERd	L		4						ERs32→@ERd	—	—	↕	↕	0	—
	MOV.L ERs,@(d:16,ERd)	L			6					ERs32→@(d:16,ERd)	—	—	↕	↕	0	—
	MOV.L ERs,@(d:32,ERd)	L			10					ERs32→@(d:32,ERd)	—	—	↕	↕	0	—
	MOV.L ERs,@-ERd	L				4				ERd32-4→ERd32, ERs32→@ERd	—	—	↕	↕	0	—
	MOV.L ERs,@aa:16	L					6			ERs32→@aa:16	—	—	↕	↕	0	—
	MOV.L ERs,@aa:32	L					8			ERs32→@aa:32	—	—	↕	↕	0	—
POP	POP.W Rn	W							2	@SP→Rn16,SP+2→SP	—	—	↕	↕	0	—
	POP.L ERn	L							4	@SP→ERn32,SP+4→SP	—	—	↕	↕	0	—
PUSH	PUSH.W Rn	W							2	SP-2→SP,Rn16→@SP	—	—	↕	↕	0	—
	PUSH.L ERn	L							4	SP-4→SP,ERn32→@SP	—	—	↕	↕	0	—
LDM	LDM @SP+,(ERm-ERn)	L							4	(@SP→ERn32,SP+4→SP) Repeated for each register restored	—	—	—	—	—	—
STM	STM (ERm-ERn),@-SP	L							4	(SP-4→SP, ERn32→@SP) Repeated for each register saved	—	—	—	—	—	—
MOVFPE	MOVFPE @aa:16,Rd	Cannot be used in the H8S/2655 Group														
MOVTPE	MOVTPE Rs,@aa:16	Cannot be used in the H8S/2655 Group														

Mnemonic	Operand Size	Addressing Mode/ Instruction Length (Bytes)								Operation	Condition Code				
		#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa		I	H	N	Z	V
ADD	ADD.B #xx:8,Rd	B 2								Rd8+#xx:8→Rd8	—	↕	↕	↕	↕
	ADD.B Rs,Rd	B	2							Rd8+Rs8→Rd8	—	↕	↕	↕	↕
	ADD.W #xx:16,Rd	W 4								Rd16+#xx:16→Rd16	—	[3]	↕	↕	↕
	ADD.W Rs,Rd	W	2							Rd16+Rs16→Rd16	—	[3]	↕	↕	↕
	ADD.L #xx:32,ERd	L 6								ERd32+#xx:32→ERd32	—	[4]	↕	↕	↕
	ADD.L ERs,ERd	L	2							ERd32+ERs32→ERd32	—	[4]	↕	↕	↕
ADDX	ADDX #xx:8,Rd	B 2								Rd8+#xx:8+C→Rd8	—	↕	↕	[5]	↕
	ADDX Rs,Rd	B	2							Rd8+Rs8+C→Rd8	—	↕	↕	[5]	↕
ADDS	ADDS #1,ERd	L	2							ERd32+1→ERd32	—	—	—	—	—
	ADDS #2,ERd	L	2							ERd32+2→ERd32	—	—	—	—	—
	ADDS #4,ERd	L	2							ERd32+4→ERd32	—	—	—	—	—
INC	INC.B Rd	B	2							Rd8+1→Rd8	—	—	↕	↕	↕
	INC.W #1,Rd	W	2							Rd16+1→Rd16	—	—	↕	↕	↕
	INC.W #2,Rd	W	2							Rd16+2→Rd16	—	—	↕	↕	↕
	INC.L #1,ERd	L	2							ERd32+1→ERd32	—	—	↕	↕	↕
	INC.L #2,ERd	L	2							ERd32+2→ERd32	—	—	↕	↕	↕
DAA	DAA Rd	B	2							Rd8 decimal adjust→Rd8	—	*	↕	↕	*
SUB	SUB.B Rs,Rd	B	2							Rd8-Rs8→Rd8	—	↕	↕	↕	↕
	SUB.W #xx:16,Rd	W 4								Rd16-#xx:16→Rd16	—	[3]	↕	↕	↕

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)										Operation	Condition Code		
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa				I	H	N
SUB	SUB.W Rs,Rd	W		2									Rd16-Rs16→Rd16	—	[3]	↕
	SUB.L #xx:32,ERd	L	6										ERd32-#xx:32→ERd32	—	[4]	↕
	SUB.L ERs,ERd	L		2									ERd32-ERs32→ERd32	—	[4]	↕
SUBX	SUBX #xx:8,Rd	B	2										Rd8-#xx:8-C→Rd8	—	↕	↕
	SUBX Rs,Rd	B		2									Rd8-Rs8-C→Rd8	—	↕	↕
SUBS	SUBS #1,ERd	L		2									ERd32-1→ERd32	—	—	—
	SUBS #2,ERd	L		2									ERd32-2→ERd32	—	—	—
	SUBS #4,ERd	L		2									ERd32-4→ERd32	—	—	—
DEC	DEC.B Rd	B		2									Rd8-1→Rd8	—	—	↕
	DEC.W #1,Rd	W		2									Rd16-1→Rd16	—	—	↕
	DEC.W #2,Rd	W		2									Rd16-2→Rd16	—	—	↕
	DEC.L #1,ERd	L		2									ERd32-1→ERd32	—	—	↕
	DEC.L #2,ERd	L		2									ERd32-2→ERd32	—	—	↕
DAS	DAS Rd	B		2									Rd8 decimal adjust→Rd8	—	*	↕
MULXU	MULXU.B Rs,Rd	B		2									Rd8×Rs8→Rd16 (unsigned multiplication)	—	—	—
	MULXU.W Rs,ERd	W		2									Rd16×Rs16→ERd32 (unsigned multiplication)	—	—	—
MULXS	MULXS.B Rs,Rd	B		4									Rd8×Rs8→Rd16 (signed multiplication)	—	—	↕
	MULXS.W Rs,ERd	W		4									Rd16×Rs16→ERd32 (signed multiplication)	—	—	↕

RENESAS

Mnemonic	Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code	
		#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	|		I	H
DIVXU	DIVXU.B Rs,Rd	B	2								Rd16÷Rs8→Rd16 (RdH: remainder, RdL: quotient) (unsigned division)	—	—
	DIVXU.W Rs,ERd	W	2								ERd32÷Rs16→ERd32 (Ed: remainder, Rd: quotient) (unsigned division)	—	—
DIVXS	DIVXS.B Rs,Rd	B	4								Rd16÷Rs8→Rd16 (RdH: remainder, RdL: quotient) (signed division)	—	—
	DIVXS.W Rs,ERd	W	4								ERd32÷Rs16→ERd32 (Ed: remainder, Rd: quotient) (signed division)	—	—
CMP	CMP.B #xx:8,Rd	B	2								Rd8-#xx:8	↔	↔
	CMP.B Rs,Rd	B		2							Rd8-Rs8	↔	↔
	CMP.W #xx:16,Rd	W	4								Rd16-#xx:16	—	[3]
	CMP.W Rs,Rd	W		2							Rd16-Rs16	—	[3]
	CMP.L #xx:32,ERd	L	6								ERd32-#xx:32	—	[4]
	CMP.L ERs,ERd	L		2							ERd32-ERs32	—	[4]
NEG	NEG.B Rd	B		2							0-Rd8→Rd8	↔	↔
	NEG.W Rd	W		2							0-Rd16→Rd16	↔	↔
	NEG.L ERd	L		2							0-ERd32→ERd32	↔	↔
EXTU	EXTU.W Rd	W		2							0→(<bit 15 to 8> of Rd16)	—	—
	EXTU.L ERd	L		2							0→(<bit 31 to 16> of ERd32)	—	—

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condi	
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	—		I	H
EXTS	EXTS.W Rd	W		2								(<bit 7> of Rd16)→ (<bit 15 to 8> of Rd16)	—	—
	EXTS.L ERd	L		2								(<bit 15> of ERd32)→ (<bit 31 to 16> of ERd32)	—	—
TAS	TAS @ERd	B			4							@ERd-0→CCR set, (1)→ (<bit 7> of @ERd)	—	—
MAC	MAC @ERn+,@ERm+	—					4					@ERn×@ERm+MAC→MAC (signed multiplication) ERn+2→ERn,ERm+2→ERm	—	—
CLRMAC	CLRMAC	—									2	0→MACH,MACL	—	—
LDMAC	LDMAC ERs,MACH	L		2								ERs→MACH	—	—
	LDMAC ERs,MACL	L		2								ERs→MACL	—	—
STMAC	STMAC MACH,ERd	L		2								MACH→ERd	—	—
	STMAC MACL,ERd	L		2								MACL→ERd	—	—

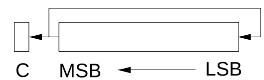
RENESAS

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code				
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	—		I	H	N	Z	V
AND	AND.B #xx:8,Rd	B	2									Rd8∧#xx:8→Rd8	—	—	↕	↕	0
	AND.B Rs,Rd	B		2								Rd8∧Rs8→Rd8	—	—	↕	↕	0
	AND.W #xx:16,Rd	W	4									Rd16∧#xx:16→Rd16	—	—	↕	↕	0
	AND.W Rs,Rd	W		2								Rd16∧Rs16→Rd16	—	—	↕	↕	0
	AND.L #xx:32,ERd	L	6									ERd32∧#xx:32→ERd32	—	—	↕	↕	0
	AND.L ERs,ERd	L		4								ERd32∧ERs32→ERd32	—	—	↕	↕	0
OR	OR.B #xx:8,Rd	B	2									Rd8∨#xx:8→Rd8	—	—	↕	↕	0
	OR.B Rs,Rd	B		2								Rd8∨Rs8→Rd8	—	—	↕	↕	0
	OR.W #xx:16,Rd	W	4									Rd16∨#xx:16→Rd16	—	—	↕	↕	0
	OR.W Rs,Rd	W		2								Rd16∨Rs16→Rd16	—	—	↕	↕	0
	OR.L #xx:32,ERd	L	6									ERd32∨#xx:32→ERd32	—	—	↕	↕	0
	OR.L ERs,ERd	L		4								ERd32∨ERs32→ERd32	—	—	↕	↕	0
XOR	XOR.B #xx:8,Rd	B	2									Rd8⊕#xx:8→Rd8	—	—	↕	↕	0
	XOR.B Rs,Rd	B		2								Rd8⊕Rs8→Rd8	—	—	↕	↕	0
	XOR.W #xx:16,Rd	W	4									Rd16⊕#xx:16→Rd16	—	—	↕	↕	0
	XOR.W Rs,Rd	W		2								Rd16⊕Rs16→Rd16	—	—	↕	↕	0
	XOR.L #xx:32,ERd	L	6									ERd32⊕#xx:32→ERd32	—	—	↕	↕	0
	XOR.L ERs,ERd	L		4								ERd32⊕ERs32→ERd32	—	—	↕	↕	0
NOT	NOT.B Rd	B		2								¬ Rd8→Rd8	—	—	↕	↕	0
	NOT.W Rd	W		2								¬ Rd16→Rd16	—	—	↕	↕	0

Mnemonic	Operand Size	Addressing Mode/ Instruction Length (Bytes)										Operation	Condition Code				
		#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	—			I	H	N	Z	V
SHAL	SHAL.B Rd	B	2										—	—	↕	↕	↕
	SHAL.B #2,Rd	B	2										—	—	↕	↕	↕
	SHAL.W Rd	W	2										—	—	↕	↕	↕
	SHAL.W #2,Rd	W	2										—	—	↕	↕	↕
	SHAL.L ERd	L	2										—	—	↕	↕	↕
	SHAL.L #2,ERd	L	2										—	—	↕	↕	↕
SHAR	SHAR.B Rd	B	2										—	—	↕	↕	0
	SHAR.B #2,Rd	B	2										—	—	↕	↕	0
	SHAR.W Rd	W	2										—	—	↕	↕	0
	SHAR.W #2,Rd	W	2										—	—	↕	↕	0
	SHAR.L ERd	L	2										—	—	↕	↕	0
	SHAR.L #2,ERd	L	2										—	—	↕	↕	0
SHLL	SHLL.B Rd	B	2										—	—	↕	↕	0
	SHLL.B #2,Rd	B	2										—	—	↕	↕	0
	SHLL.W Rd	W	2										—	—	↕	↕	0
	SHLL.W #2,Rd	W	2										—	—	↕	↕	0
	SHLL.L ERd	L	2										—	—	↕	↕	0
	SHLL.L #2,ERd	L	2										—	—	↕	↕	0

RENESAS

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)										Operation	Condition Code				
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	—			I	H	N	Z	V	C
SHLR	SHLR.B Rd	B		2									0 → MSB → LSB → C	—	—	↕	↕	0	↕
	SHLR.B #2,Rd	B		2										—	—	↕	↕	0	↕
	SHLR.W Rd	W		2										—	—	↕	↕	0	↕
	SHLR.W #2,Rd	W		2										—	—	↕	↕	0	↕
	SHLR.L ERd	L		2										—	—	↕	↕	0	↕
	SHLR.L #2,ERd	L		2										—	—	↕	↕	0	↕
ROTXL	ROTXL.B Rd	B		2									C MSB → LSB	—	—	↕	↕	0	↕
	ROTXL.B #2,Rd	B		2										—	—	↕	↕	0	↕
	ROTXL.W Rd	W		2										—	—	↕	↕	0	↕
	ROTXL.W #2,Rd	W		2										—	—	↕	↕	0	↕
	ROTXL.L ERd	L		2										—	—	↕	↕	0	↕
	ROTXL.L #2,ERd	L		2										—	—	↕	↕	0	↕
ROTXR	ROTXR.B Rd	B		2									MSB → LSB → C	—	—	↕	↕	0	↕
	ROTXR.B #2,Rd	B		2										—	—	↕	↕	0	↕
	ROTXR.W Rd	W		2										—	—	↕	↕	0	↕
	ROTXR.W #2,Rd	W		2										—	—	↕	↕	0	↕
	ROTXR.L ERd	L		2										—	—	↕	↕	0	↕
	ROTXR.L #2,ERd	L		2										—	—	↕	↕	0	↕

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condi…	
			#xx	Rn	@ERn	@(d,ERn)	@−ERn/@ERn+	@aa	@(d,PC)	@@aa	—		I	H
ROTL	ROTL.B Rd	B		2									—	—
	ROTL.B #2,Rd	B		2									—	—
	ROTL.W Rd	W		2									—	—
	ROTL.W #2,Rd	W		2									—	—
	ROTL.L ERd	L		2									—	—
	ROTL.L #2,ERd	L		2									—	—
ROTR	ROTR.B Rd	B		2									—	—
	ROTR.B #2,Rd	B		2									—	—
	ROTR.W Rd	W		2									—	—
	ROTR.W #2,Rd	W		2									—	—
	ROTR.L ERd	L		2									—	—
	ROTR.L #2,ERd	L		2									—	—

Mnemonic	Operand Size	Addressing Mode/ Instruction Length (Bytes)								Operation	Condition Co				
		#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa		I	H	N	Z	V
BSET	BSET #xx:3,Rd	B	2							(#xx:3 of Rd8)←1	—	—	—	—	—
	BSET #xx:3,@ERd	B		4						(#xx:3 of @ERd)←1	—	—	—	—	—
	BSET #xx:3,@aa:8	B					4			(#xx:3 of @aa:8)←1	—	—	—	—	—
	BSET #xx:3,@aa:16	B					6			(#xx:3 of @aa:16)←1	—	—	—	—	—
	BSET #xx:3,@aa:32	B					8			(#xx:3 of @aa:32)←1	—	—	—	—	—
	BSET Rn,Rd	B	2							(Rn8 of Rd8)←1	—	—	—	—	—
	BSET Rn,@ERd	B		4						(Rn8 of @ERd)←1	—	—	—	—	—
	BSET Rn,@aa:8	B					4			(Rn8 of @aa:8)←1	—	—	—	—	—
	BSET Rn,@aa:16	B					6			(Rn8 of @aa:16)←1	—	—	—	—	—
	BSET Rn,@aa:32	B					8			(Rn8 of @aa:32)←1	—	—	—	—	—
BCLR	BCLR #xx:3,Rd	B	2							(#xx:3 of Rd8)←0	—	—	—	—	—
	BCLR #xx:3,@ERd	B		4						(#xx:3 of @ERd)←0	—	—	—	—	—
	BCLR #xx:3,@aa:8	B					4			(#xx:3 of @aa:8)←0	—	—	—	—	—
	BCLR #xx:3,@aa:16	B					6			(#xx:3 of @aa:16)←0	—	—	—	—	—
	BCLR #xx:3,@aa:32	B					8			(#xx:3 of @aa:32)←0	—	—	—	—	—
	BCLR Rn,Rd	B	2							(Rn8 of Rd8)←0	—	—	—	—	—
	BCLR Rn,@ERd	B		4						(Rn8 of @ERd)←0	—	—	—	—	—
	BCLR Rn,@aa:8	B					4			(Rn8 of @aa:8)←0	—	—	—	—	—
	BCLR Rn,@aa:16	B					6			(Rn8 of @aa:16)←0	—	—	—	—	—

Mnemonic	Operand Size	Addressing Mode/ Instruction Length (Bytes)										Operation	Condition Code		
		#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa				I	H	N
BCLR	BCLR Rn,@aa:32	B					8					(Rn8 of @aa:32)←0	—	—	—
BNOT	BNOT #xx:3,Rd	B	2									(#xx:3 of Rd8)←[¬ (#xx:3 of Rd8)]	—	—	—
	BNOT #xx:3,@ERd	B		4								(#xx:3 of @ERd)←[¬ (#xx:3 of @ERd)]	—	—	—
	BNOT #xx:3,@aa:8	B					4					(#xx:3 of @aa:8)←[¬ (#xx:3 of @aa:8)]	—	—	—
	BNOT #xx:3,@aa:16	B					6					(#xx:3 of @aa:16)←[¬ (#xx:3 of @aa:16)]	—	—	—
	BNOT #xx:3,@aa:32	B					8					(#xx:3 of @aa:32)←[¬ (#xx:3 of @aa:32)]	—	—	—
	BNOT Rn,Rd	B	2									(Rn8 of Rd8)←[¬ (Rn8 of Rd8)]	—	—	—
	BNOT Rn,@ERd	B		4								(Rn8 of @ERd)←[¬ (Rn8 of @ERd)]	—	—	—
	BNOT Rn,@aa:8	B					4					(Rn8 of @aa:8)←[¬ (Rn8 of @aa:8)]	—	—	—
	BNOT Rn,@aa:16	B					6					(Rn8 of @aa:16)←[¬ (Rn8 of @aa:16)]	—	—	—
	BNOT Rn,@aa:32	B					8					(Rn8 of @aa:32)←[¬ (Rn8 of @aa:32)]	—	—	—
BTST	BTST #xx:3,Rd	B	2									¬ (#xx:3 of Rd8)→Z	—	—	↕
	BTST #xx:3,@ERd	B		4								¬ (#xx:3 of @ERd)→Z	—	—	↕
	BTST #xx:3,@aa:8	B					4					¬ (#xx:3 of @aa:8)→Z	—	—	↕

RENESAS

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)								Operation	Condition Code					
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa		I	H	N	Z	V	C
BTST	BTST #xx:3,@aa:32	B						8			¬ (#xx:3 of @aa:32)→Z	—	—	—	↕	—	—
	BTST Rn,Rd	B	2								¬ (Rn8 of Rd8)→Z	—	—	—	↕	—	—
	BTST Rn,@ERd	B		4							¬ (Rn8 of @ERd)→Z	—	—	—	↕	—	—
	BTST Rn,@aa:8	B						4			¬ (Rn8 of @aa:8)→Z	—	—	—	↕	—	—
	BTST Rn,@aa:16	B						6			¬ (Rn8 of @aa:16)→Z	—	—	—	↕	—	—
	BTST Rn,@aa:32	B						8			¬ (Rn8 of @aa:32)→Z	—	—	—	↕	—	—
BLD	BLD #xx:3,Rd	B	2								(#xx:3 of Rd8)→C	—	—	—	—	—	↕
	BLD #xx:3,@ERd	B		4							(#xx:3 of @ERd)→C	—	—	—	—	—	↕
	BLD #xx:3,@aa:8	B						4			(#xx:3 of @aa:8)→C	—	—	—	—	—	↕
	BLD #xx:3,@aa:16	B						6			(#xx:3 of @aa:16)→C	—	—	—	—	—	↕
	BLD #xx:3,@aa:32	B						8			(#xx:3 of @aa:32)→C	—	—	—	—	—	↕
BILD	BILD #xx:3,Rd	B	2								¬ (#xx:3 of Rd8)→C	—	—	—	—	—	↕
	BILD #xx:3,@ERd	B		4							¬ (#xx:3 of @ERd)→C	—	—	—	—	—	↕
	BILD #xx:3,@aa:8	B						4			¬ (#xx:3 of @aa:8)→C	—	—	—	—	—	↕
	BILD #xx:3,@aa:16	B						6			¬ (#xx:3 of @aa:16)→C	—	—	—	—	—	↕
	BILD #xx:3,@aa:32	B						8			¬ (#xx:3 of @aa:32)→C	—	—	—	—	—	↕
BST	BST #xx:3,Rd	B	2								C→(#xx:3 of Rd8)	—	—	—	↕	—	—
	BST #xx:3,@ERd	B		4							C→(#xx:3 of @ERd)	—	—	—	↕	—	—
	BST #xx:3,@aa:8	B						4			C→(#xx:3 of @aa:8)	—	—	—	↕	—	—

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)								Operation	Condition Code					
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	|		I	H	N	Z	V	C
BST	BST #xx:3,@aa:16	B						6				C→(#xx:3 of @aa:16)	—	—	—	—	—	↕
	BST #xx:3,@aa:32	B						8				C→(#xx:3 of @aa:32)	—	—	—	—	—	↕
BIST	BIST #xx:3,Rd	B	2									¬ C→(#xx:3 of Rd8)	—	—	—	—	—	↕
	BIST #xx:3,@ERd	B			4							¬ C→(#xx:3 of @ERd)	—	—	—	—	—	↕
	BIST #xx:3,@aa:8	B						4				¬ C→(#xx:3 of @aa:8)	—	—	—	—	—	↕
	BIST #xx:3,@aa:16	B						6				¬ C→(#xx:3 of @aa:16)	—	—	—	—	—	↕
	BIST #xx:3,@aa:32	B						8				¬ C→(#xx:3 of @aa:32)	—	—	—	—	—	↕
BAND	BAND #xx:3,Rd	B	2									C∧(#xx:3 of Rd8)→C	—	—	—	—	—	↕
	BAND #xx:3,@ERd	B			4							C∧(#xx:3 of @ERd)→C	—	—	—	—	—	↕
	BAND #xx:3,@aa:8	B						4				C∧(#xx:3 of @aa:8)→C	—	—	—	—	—	↕
	BAND #xx:3,@aa:16	B						6				C∧(#xx:3 of @aa:16)→C	—	—	—	—	—	↕
	BAND #xx:3,@aa:32	B						8				C∧(#xx:3 of @aa:32)→C	—	—	—	—	—	↕
BIAND	BIAND #xx:3,Rd	B	2									C∧[¬ (#xx:3 of Rd8)]→C	—	—	—	—	—	↕
	BIAND #xx:3,@ERd	B			4							C∧[¬ (#xx:3 of @ERd)]→C	—	—	—	—	—	↕
	BIAND #xx:3,@aa:8	B						4				C∧[¬ (#xx:3 of @aa:8)]→C	—	—	—	—	—	↕
	BIAND #xx:3,@aa:16	B						6				C∧[¬ (#xx:3 of @aa:16)]→C	—	—	—	—	—	↕
	BIAND #xx:3,@aa:32	B						8				C∧[¬ (#xx:3 of @aa:32)]→C	—	—	—	—	—	↕
BOR	BOR #xx:3,Rd	B	2									C∨(#xx:3 of Rd8)→C	—	—	—	—	—	↕
	BOR #xx:3,@ERd	B			4							C∨(#xx:3 of @ERd)→C	—	—	—	—	—	↕

RENESAS

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)								Operation	Condition Code			
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa		I	H	N	Z
BOR	BOR #xx:3,@aa:8	B						4			C∨(#xx:3 of @aa:8)→C	—	—	—	—
	BOR #xx:3,@aa:16	B						6			C∨(#xx:3 of @aa:16)→C	—	—	—	—
	BOR #xx:3,@aa:32	B						8			C∨(#xx:3 of @aa:32)→C	—	—	—	—
BIOR	BIOR #xx:3,Rd	B	2								C∨[¬ (#xx:3 of Rd8)]→C	—	—	—	—
	BIOR #xx:3,@ERd	B		4							C∨[¬ (#xx:3 of @ERd)]→C	—	—	—	—
	BIOR #xx:3,@aa:8	B						4			C∨[¬ (#xx:3 of @aa:8)]→C	—	—	—	—
	BIOR #xx:3,@aa:16	B						6			C∨[¬ (#xx:3 of @aa:16)]→C	—	—	—	—
	BIOR #xx:3,@aa:32	B						8			C∨[¬ (#xx:3 of @aa:32)]→C	—	—	—	—
BXOR	BXOR #xx:3,Rd	B	2								C⊕(#xx:3 of Rd8)→C	—	—	—	—
	BXOR #xx:3,@ERd	B		4							C⊕(#xx:3 of @ERd)→C	—	—	—	—
	BXOR #xx:3,@aa:8	B						4			C⊕(#xx:3 of @aa:8)→C	—	—	—	—
	BXOR #xx:3,@aa:16	B						6			C⊕(#xx:3 of @aa:16)→C	—	—	—	—
	BXOR #xx:3,@aa:32	B						8			C⊕(#xx:3 of @aa:32)→C	—	—	—	—
BIXOR	BIXOR #xx:3,Rd	B	2								C⊕[¬ (#xx:3 of Rd8)]→C	—	—	—	—
	BIXOR #xx:3,@ERd	B		4							C⊕[¬ (#xx:3 of @ERd)]→C	—	—	—	—
	BIXOR #xx:3,@aa:8	B						4			C⊕[¬ (#xx:3 of @aa:8)]→C	—	—	—	—
	BIXOR #xx:3,@aa:16	B						6			C⊕[¬ (#xx:3 of @aa:16)]→C	—	—	—	—
	BIXOR #xx:3,@aa:32	B						8			C⊕[¬ (#xx:3 of @aa:32)]→C	—	—	—	—

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)										Operation		Condition Code				
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	—			Branching Condition	I	H	N	Z	V
Bcc	BRA d:8(BT d:8)	—							2			if condition is true then PC←PC+d else next;		Always	—	—	—	—	—
	BRA d:16(BT d:16)	—							4						—	—	—	—	—
	BRN d:8(BF d:8)	—							2					Never	—	—	—	—	—
	BRN d:16(BF d:16)	—							4						—	—	—	—	—
	BHI d:8	—							2					C∨Z=0	—	—	—	—	—
	BHI d:16	—							4						—	—	—	—	—
	BLS d:8	—							2					C∨Z=1	—	—	—	—	—
	BLS d:16	—							4						—	—	—	—	—
	BCC d:B(BHS d:8)	—							2					C=0	—	—	—	—	—
	BCC d:16(BHS d:16)	—							4						—	—	—	—	—
	BCS d:8(BLO d:8)	—							2					C=1	—	—	—	—	—
	BCS d:16(BLO d:16)	—							4						—	—	—	—	—
	BNE d:8	—							2					Z=0	—	—	—	—	—
	BNE d:16	—							4						—	—	—	—	—
	BEQ d:8	—							2					Z=1	—	—	—	—	—
	BEQ d:16	—							4						—	—	—	—	—
	BVC d:8	—							2					V=0	—	—	—	—	—
	BVC d:16	—							4						—	—	—	—	—

RENESAS

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition				
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	—	Branching Condition	I	H	N	Z	V	C
Bcc	BVS d:8	—	2									V=1	—	—	—	—	—	—
	BVS d:16	—	4										—	—	—	—	—	—
	BPL d:8	—	2									N=0	—	—	—	—	—	—
	BPL d:16	—	4										—	—	—	—	—	—
	BMI d:8	—	2									N=1	—	—	—	—	—	—
	BMI d:16	—	4										—	—	—	—	—	—
	BGE d:8	—	2									N⊕V=0	—	—	—	—	—	—
	BGE d:16	—	4										—	—	—	—	—	—
	BLT d:8	—	2									N⊕V=1	—	—	—	—	—	—
	BLT d:16	—	4										—	—	—	—	—	—
	BGT d:8	—	2									Z∨(N⊕V)=0	—	—	—	—	—	—
	BGT d:16	—	4										—	—	—	—	—	—
	BLE d:8	—	2									Z∨(N⊕V)=1	—	—	—	—	—	—
	BLE d:16	—	4										—	—	—	—	—	—

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code		
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	—		I	H	N
JMP	JMP @ERn	—			2							PC← ERn	—	—	—
	JMP @aa:24	—						4				PC← aa:24	—	—	—
	JMP @@aa:8	—								2		PC← @aa:8	—	—	—
BSR	BSR d:8	—							2			PC→@-SP,PC←PC+d:8	—	—	—
	BSR d:16	—							4			PC→@-SP,PC←PC+d:16	—	—	—
JSR	JSR @ERn	—			2							PC→@-SP,PC←ERn	—	—	—
	JSR @aa:24	—						4				PC→@-SP,PC←aa:24	—	—	—
	JSR @@aa:8	—								2		PC→@-SP,PC←@aa:8	—	—	—
RTS	RTS	—									2	PC← @SP+	—	—	—

RENESAS

Mnemonic	Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code			
		#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	|		I	H	N	Z
TRAPA	TRAPA #xx:2	—									PC→@-SP,CCR→@-SP, EXR→@-SP,<vector>→PC	1	—	—	—
RTE	RTE	—									EXR←@SP+,CCR←@SP+, PC←@SP+	↕	↕	↕	↕
SLEEP	SLEEP	—									Transition to power-down state	—	—	—	—
LDC	LDC #xx:8,CCR	B 2									#xx:8→CCR	↕	↕	↕	↕
	LDC #xx:8,EXR	B 4									#xx:8→EXR	—	—	—	—
	LDC Rs,CCR	B	2								Rs8→CCR	↕	↕	↕	↕
	LDC Rs,EXR	B	2								Rs8→EXR	—	—	—	—
	LDC @ERs,CCR	W		4							@ERs→CCR	↕	↕	↕	↕
	LDC @ERs,EXR	W		4							@ERs→EXR	—	—	—	—
	LDC @(d:16,ERs),CCR	W			6						@(d:16,ERs)→CCR	↕	↕	↕	↕
	LDC @(d:16,ERs),EXR	W			6						@(d:16,ERs)→EXR	—	—	—	—
	LDC @(d:32,ERs),CCR	W			10						@(d:32,ERs)→CCR	↕	↕	↕	↕
	LDC @(d:32,ERs),EXR	W			10						@(d:32,ERs)→EXR	—	—	—	—
	LDC @ERs+,CCR	W				4					@ERs→CCR,ERs32+2→ERs32	↕	↕	↕	↕
	LDC @ERs+,EXR	W				4					@ERs→EXR,ERs32+2→ERs32	—	—	—	—
	LDC @aa:16,CCR	W					6				@aa:16→CCR	↕	↕	↕	↕
	LDC @aa:16,EXR	W					6				@aa:16→EXR	—	—	—	—
	LDC @aa:32,CCR	W					8				@aa:32→CCR	↕	↕	↕	↕
	LDC @aa:32,EXR	W					8				@aa:32→EXR	—	—	—	—

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)								Operation	Condition Code		
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa		I	H	N
STC	STC CCR,Rd	B	2								CCR→Rd8	—	—	—
	STC EXR,Rd	B	2								EXR→Rd8	—	—	—
	STC CCR,@ERd	W		4							CCR→@ERd	—	—	—
	STC EXR,@ERd	W		4							EXR→@ERd	—	—	—
	STC CCR,@(d:16,ERd)	W			6						CCR→@(d:16,ERd)	—	—	—
	STC EXR,@(d:16,ERd)	W			6						EXR→@(d:16,ERd)	—	—	—
	STC CCR,@(d:32,ERd)	W			10						CCR→@(d:32,ERd)	—	—	—
	STC EXR,@(d:32,ERd)	W			10						EXR→@(d:32,ERd)	—	—	—
	STC CCR,@-ERd	W				4					ERd32-2→ERd32,CCR→@ERd	—	—	—
	STC EXR,@-ERd	W				4					ERd32-2→ERd32,EXR→@ERd	—	—	—
	STC CCR,@aa:16	W					6				CCR→@aa:16	—	—	—
	STC EXR,@aa:16	W					6				EXR→@aa:16	—	—	—
	STC CCR,@aa:32	W					8				CCR→@aa:32	—	—	—
	STC EXR,@aa:32	W					8				EXR→@aa:32	—	—	—
ANDC	ANDC #xx:8,CCR	B	2								CCR∧#xx:8→CCR	↔	↔	↔
	ANDC #xx:8,EXR	B	4								EXR∧#xx:8→EXR	—	—	—
ORC	ORC #xx:8,CCR	B	2								CCR∨#xx:8→CCR	↔	↔	↔
	ORC #xx:8,EXR	B	4								EXR∨#xx:8→EXR	—	—	—
XORC	XORC #xx:8,CCR	B	2								CCR⊕#xx:8→CCR	↔	↔	↔
	XORC #xx:8,EXR	B	4								EXR⊕#xx:8→EXR	—	—	—

Mnemonic	Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code			
		#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	—		I	H	N	Z
EEPMOV	EEPMOV.B	—								4	if R4L≠0 Repeat @ER5→@ER6 ER5+1→ER5 ER6+1→ER6 R4L-1→R4L Until R4L=0 else next;	—	—	—	—
	EEPMOV.W	—								4	if R4≠0 Repeat @ER5→@ER6 ER5+1→ER5 ER6+1→ER6 R4-1→R4 Until R4=0 else next;	—	—	—	—

Notes: 1. The number of states is the number of states required for execution when the instruction and its operands are loc
2. n is the initial value of R4L or R4.
[1] Seven states for saving or restoring two registers, nine states for three registers, or eleven states for four register
[2] Cannot be used in the H8S/2655 Group.
[3] Set to 1 when a carry or borrow occurs at bit 11; otherwise cleared to 0.
[4] Set to 1 when a carry or borrow occurs at bit 27; otherwise cleared to 0.
[5] Retains its previous value when the result is zero; otherwise cleared to 0.
[6] One additional state is required for execution immediately after a MULXU, MULXS, or STMAC instruction. Also, a
additional states are required for execution of a MULXU instruction within three states after execution of a MAC in
if there is a one-state instruction (such as NOP) between a MAC instruction and a MULXU instruction, the MULXU
states longer.
[7] A maximum of two additional states are required for execution of a MULXS instruction within two states after exec
instruction. For example, if there is a one-state instruction (such as NOP) between a MAC instruction and a MUL
MULXS instruction will be one state longer.
[8] Set to 1 when the divisor is negative; otherwise cleared to 0.
[9] Set to 1 when the divisor is zero; otherwise cleared to 0.

Instruction	Mnemonic	Size	Instruction Format 1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
ADD	ADD.B #xx:8,Rd	B	8 rd	IMM						
	ADD.B Rs,Rd	B	0 8	rs rd						
	ADD.W #xx:16,Rd	W	7 9	1 rd	IMM					
	ADD.W Rs,Rd	W	0 9	rs rd						
	ADD.L #xx:32,ERd	L	7 A	1 0 erd	IMM					
	ADD.L ERs,ERd	L	0 A	1 ers 0 erd						
ADDS	ADDS #1,ERd	L	0 B	0 0 erd						
	ADDS #2,ERd	L	0 B	8 0 erd						
	ADDS #4,ERd	L	0 B	9 0 erd						
ADDX	ADDX #xx:8,Rd	B	9 rd	IMM						
	ADDX Rs,Rd	B	0 E	rs rd						
AND	AND.B #xx:8,Rd	B	E rd	IMM						
	AND.B Rs,Rd	B	1 6	rs rd						
	AND.W #xx:16,Rd	W	7 9	6 rd	IMM					
	AND.W Rs,Rd	W	6 6	rs rd						
	AND.L #xx:32,ERd	L	7 A	6 0 erd	IMM					
	AND.L ERs,ERd	L	0 1	F 0	6 6	0 ers 0 erd				
ANDC	ANDC #xx:8,CCR	B	0 6	IMM						
	ANDC #xx:8,EXR	B	0 1	4 1	0 6	IMM				
BAND	BAND #xx:3,Rd	B	7 6	0 IMM rd						
	BAND #xx:3,@ERd	B	7 C	0 erd 0	7 6	0 IMM 0				
	BAND #xx:3,@aa:8	B	7 E	abs	7 6	0 IMM 0				
	BAND #xx:3,@aa:16	B	6 A	1 0	abs		7 6	0 IMM 0		
	BAND #xx:3,@aa:32	B	6 A	3 0	abs				7 6	0 IMM
Bcc	BRA d:8 (BT d:8)	—	4 0	disp						
	BRA d:16 (BT d:16)	—	5 8	0 0	disp					
	BRN d:8 (BF d:8)	—	4 1	disp						
	BRN d:16 (BF d:16)	—	5 8	1 0	disp					

Instruction	Mnemonic	Size	Instruction Format							
			1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8
Bcc	BHI d:8	—	4 2	disp						
	BHI d:16	—	5 8	2 0	disp					
	BLS d:8	—	4 3	disp						
	BLS d:16	—	5 8	3 0	disp					
	BCC d:8 (BHS d:8)	—	4 4	disp						
	BCC d:16 (BHS d:16)	—	5 8	4 0	disp					
	BCS d:8 (BLO d:8)	—	4 5	disp						
	BCS d:16 (BLO d:16)	—	5 8	5 0	disp					
	BNE d:8	—	4 6	disp						
	BNE d:16	—	5 8	6 0	disp					
	BEQ d:8	—	4 7	disp						
	BEQ d:16	—	5 8	7 0	disp					
	BVC d:8	—	4 8	disp						
	BVC d:16	—	5 8	8 0	disp					
	BVS d:8	—	4 9	disp						
	BVS d:16	—	5 8	9 0	disp					
	BPL d:8	—	4 A	disp						
	BPL d:16	—	5 8	A 0	disp					
	BMI d:8	—	4 B	disp						
	BMI d:16	—	5 8	B 0	disp					
	BGE d:8	—	4 C	disp						
	BGE d:16	—	5 8	C 0	disp					
	BLT d:8	—	4 D	disp						
	BLT d:16	—	5 8	D 0	disp					
	BGT d:8	—	4 E	disp						
	BGT d:16	—	5 8	E 0	disp					
	BLE d:8	—	4 F	disp						
	BLE d:16	—	5 8	F 0	disp					

RENESAS

Instruc-tion	Mnemonic	Size	Instruction Format																									
			1st byte			2nd byte			3rd byte			4th byte			5th byte			6th byte			7th byte			8				
BCLR	BCLR #xx:3,Rd	B	7		2	0:IMM		rd																				
	BCLR #xx:3,@ERd	B	7		D	0	erd		0	7		2	0:IMM		0													
	BCLR #xx:3,@aa:8	B	7		F	abs				7		2	0:IMM		0													
	BCLR #xx:3,@aa:16	B	6		A	1		8		abs						7		2	0:IMM		0							
	BCLR #xx:3,@aa:32	B	6		A	3		8		abs												7		2	0:I			
	BCLR Rn,Rd	B	6		2	m		rd																				
	BCLR Rn,@ERd	B	7		D	0	erd		0	6		2	rn		0													
	BCLR Rn,@aa:8	B	7		F	abs				6		2	rn		0													
	BCLR Rn,@aa:16	B	6		A	1		8		abs						6		2	rn		0							
	BCLR Rn,@aa:32	B	6		A	3		8		abs												6		2	r			
BIAND	BIAND #xx:3,Rd	B	7		6	1:IMM		rd																				
	BIAND #xx:3,@ERd	B	7		C	0	erd		0	7		6	1:IMM		0													
	BIAND #xx:3,@aa:8	B	7		E	abs				7		6	1:IMM		0													
	BIAND #xx:3,@aa:16	B	6		A	1		0		abs						7		6	1:IMM		0							
	BIAND #xx:3,@aa:32	B	6		A	3		0		abs												7		6	1:I			
BILD	BILD #xx:3,Rd	B	7		7	1:IMM		rd																				
	BILD #xx:3,@ERd	B	7		C	0	erd		0	7		7	1:IMM		0													
	BILD #xx:3,@aa:8	B	7		E	abs				7		7	1:IMM		0													
	BILD #xx:3,@aa:16	B	6		A	1		0		abs						7		7	1:IMM		0							
	BILD #xx:3,@aa:32	B	6		A	3		0		abs												7		7	1:I			
BIOR	BIOR #xx:3,Rd	B	7		4	1:IMM		rd																				
	BIOR #xx:3,@ERd	B	7		C	0	erd		0	7		4	1:IMM		0													
	BIOR #xx:3,@aa:8	B	7		E	abs				7		4	1:IMM		0													
	BIOR #xx:3,@aa:16	B	6		A	1		0		abs						7		4	1:IMM		0							
	BIOR #xx:3,@aa:32	B	6		A	3		0		abs												7		4	1:I			

Instruction	Mnemonic	Size	Instruction Format: 1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8
BIST	BIST #xx:3,Rd	B	6 7	1:IMM rd						
	BIST #xx:3,@ERd	B	7 D	0:erd 0	6 7	1:IMM 0				
	BIST #xx:3,@aa:8	B	7 F	abs	6 7	1:IMM 0				
	BIST #xx:3,@aa:16	B	6 A	1 8	abs		6 7	1:IMM 0		
	BIST #xx:3,@aa:32	B	6 A	3 8	abs				6 7	1:I
BIXOR	BIXOR #xx:3,Rd	B	7 5	1:IMM rd						
	BIXOR #xx:3,@ERd	B	7 C	0:erd 0	7 5	1:IMM 0				
	BIXOR #xx:3,@aa:8	B	7 E	abs	7 5	1:IMM 0				
	BIXOR #xx:3,@aa:16	B	6 A	1 0	abs		7 5	1:IMM 0		
	BIXOR #xx:3,@aa:32	B	6 A	3 0	abs				7 5	1:I
BLD	BLD #xx:3,Rd	B	7 7	0:IMM rd						
	BLD #xx:3,@ERd	B	7 C	0:erd 0	7 7	0:IMM 0				
	BLD #xx:3,@aa:8	B	7 E	abs	7 7	0:IMM 0				
	BLD #xx:3,@aa:16	B	6 A	1 0	abs		7 7	0:IMM 0		
	BLD #xx:3,@aa:32	B	6 A	3 0	abs				7 7	0:I
BNOT	BNOT #xx:3,Rd	B	7 1	0:IMM rd						
	BNOT #xx:3,@ERd	B	7 D	0:erd 0	7 1	0:IMM 0				
	BNOT #xx:3,@aa:8	B	7 F	abs	7 1	0:IMM 0				
	BNOT #xx:3,@aa:16	B	6 A	1 8	abs		7 1	0:IMM 0		
	BNOT #xx:3,@aa:32	B	6 A	3 8	abs				7 1	0:I
	BNOT Rn,Rd	B	6 1	rn rd						
	BNOT Rn,@ERd	B	7 D	0:erd 0	6 1	rn 0				
	BNOT Rn,@aa:8	B	7 F	abs	6 1	rn 0				
	BNOT Rn,@aa:16	B	6 A	1 8	abs		6 1	rn 0		
	BNOT Rn,@aa:32	B	6 A	3 8	abs				6 1	r

RENESAS

Instruc-tion	Mnemonic	Size	Instruction Format: 1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
BOR	BOR #xx:3,Rd	B	7 4	0 IMM rd						
	BOR #xx:3,@ERd	B	7 C	0 erd 0	7 4	0 IMM 0				
	BOR #xx:3,@aa:8	B	7 E	abs	7 4	0 IMM 0				
	BOR #xx:3,@aa:16	B	6 A	1 0	abs		7 4	0 IMM 0		
	BOR #xx:3,@aa:32	B	6 A	3 0	abs				7 4	0 I
BSET	BSET #xx:3,Rd	B	7 0	0 IMM rd						
	BSET #xx:3,@ERd	B	7 D	0 erd 0	7 0	0 IMM 0				
	BSET #xx:3,@aa:8	B	7 F	abs	7 0	0 IMM 0				
	BSET #xx:3,@aa:16	B	6 A	1 8	abs		7 0	0 IMM 0		
	BSET #xx:3,@aa:32	B	6 A	3 8	abs				7 0	0 I
	BSET Rn,Rd	B	6 0	rn rd						
	BSET Rn,@ERd	B	7 D	0 erd 0	6 0	rn 0				
	BSET Rn,@aa:8	B	7 F	abs	6 0	rn 0				
	BSET Rn,@aa:16	B	6 A	1 8	abs		6 0	rn 0		
	BSET Rn,@aa:32	B	6 A	3 8	abs				6 0	r
BSR	BSR d:8	—	5 5	disp						
	BSR d:16	—	5 C	0 0	disp					
BST	BST #xx:3,Rd	B	6 7	0 IMM rd						
	BST #xx:3,@ERd	B	7 D	0 erd 0	6 7	0 IMM 0				
	BST #xx:3,@aa:8	B	7 F	abs	6 7	0 IMM 0				
	BST #xx:3,@aa:16	B	6 A	1 8	abs		6 7	0 IMM 0		
	BST #xx:3,@aa:32	B	6 A	3 8	abs				6 7	0 I
BTST	BTST #xx:3,Rd	B	7 3	0 IMM rd						
	BTST #xx:3,@ERd	B	7 C	0 erd 0	7 3	0 IMM 0				
	BTST #xx:3,@aa:8	B	7 E	abs	7 3	0 IMM 0				
	BTST #xx:3,@aa:16	B	6 A	1 0	abs		7 3	0 IMM 0		
	BTST #xx:3,@aa:32	B	6 A	3 0	abs				7 3	0 I
	BTST Rn,Rd	B	6 3	rn rd						
	BTST Rn,@ERd	B	7 C	0 erd 0	6 3	rn 0				

Instruction	Mnemonic	Size	1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8…
BTST	BTST Rn,@aa:8	B	7 E	abs	6 3	rn 0				
	BTST Rn,@aa:16	B	6 A	1 0	abs		6 3	rn 0		
	BTST Rn,@aa:32	B	6 A	3 0	abs				6 3	r
BXOR	BXOR #xx:3,Rd	B	7 5	0:IMM rd						
	BXOR #xx:3,@ERd	B	7 C	0:erd 0	7 5	0:IMM 0				
	BXOR #xx:3,@aa:8	B	7 E	abs	7 5	0:IMM 0				
	BXOR #xx:3,@aa:16	B	6 A	1 0	abs		7 5	0:IMM 0		
	BXOR #xx:3,@aa:32	B	6 A	3 0	abs				7 5	0:I
CLRMAC	CLRMAC	—	0 1	A 0						
CMP	CMP.B #xx:8,Rd	B	A rd	IMM						
	CMP.B Rs,Rd	B	1 C	rs rd						
	CMP.W #xx:16,Rd	W	7 9	2 rd	IMM					
	CMP.W Rs,Rd	W	1 D	rs rd						
	CMP.L #xx:32,ERd	L	7 A	2 0:erd	IMM					
	CMP.L ERs,ERd	L	1 F	1:ers 0:erd						
DAA	DAA Rd	B	0 F	0 rd						
DAS	DAS Rd	B	1 F	0 rd						
DEC	DEC.B Rd	B	1 A	0 rd						
	DEC.W #1,Rd	W	1 B	5 rd						
	DEC.W #2,Rd	W	1 B	D rd						
	DEC.L #1,ERd	L	1 B	7 0:erd						
	DEC.L #2,ERd	L	1 B	F 0:erd						
DIVXS	DIVXS.B Rs,Rd	B	0 1	D 0	5 1	rs rd				
	DIVXS.W Rs,ERd	W	0 1	D 0	5 3	rs 0:erd				
DIVXU	DIVXU.B Rs,Rd	B	5 1	rs rd						
	DIVXU.W Rs,ERd	W	5 3	rs 0:erd						
EEPMOV	EEPMOV.B	—	7 B	5 C	5 9	8 F				
	EEPMOV.W	—	7 B	D 4	5 9	8 F				

RENESAS

Instruction	Mnemonic	Size	Instruction Format 1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte
EXTS	EXTS.W Rd	W	1 7	D rd					
	EXTS.L ERd	L	1 7	F 0 erd					
EXTU	EXTU.W Rd	W	1 7	5 rd					
	EXTU.L ERd	L	1 7	7 0 erd					
INC	INC.B Rd	B	0 A	0 rd					
	INC.W #1,Rd	W	0 B	5 rd					
	INC.W #2,Rd	W	0 B	D rd					
	INC.L #1,ERd	L	0 B	7 0 erd					
	INC.L #2,ERd	L	0 B	F 0 erd					
JMP	JMP @ERn	—	5 9	0 ern 0					
	JMP @aa:24	—	5 A	abs	abs	abs			
	JMP @@aa:8	—	5 B	abs					
JSR	JSR @ERn	—	5 D	0 ern 0					
	JSR @aa:24	—	5 E	abs	abs	abs			
	JSR @@aa:8	—	5 F	abs					
LDC	LDC #xx:8,CCR	B	0 7	IMM					
	LDC #xx:8,EXR	B	0 1	4 1	0 7	IMM			
	LDC Rs,CCR	B	0 3	0 rs					
	LDC Rs,EXR	B	0 3	1 rs					
	LDC @ERs,CCR	W	0 1	4 0	6 9	0 ers 0			
	LDC @ERs,EXR	W	0 1	4 1	6 9	0 ers 0			
	LDC @(d:16,ERs),CCR	W	0 1	4 0	6 F	0 ers 0	disp	disp	
	LDC @(d:16,ERs),EXR	W	0 1	4 1	6 F	0 ers 0	disp	disp	
	LDC @(d:32,ERs),CCR	W	0 1	4 0	7 8	0 ers 0	6 B	2 0	
	LDC @(d:32,ERs),EXR	W	0 1	4 1	7 8	0 ers 0	6 B	2 0	
	LDC @ERs+,CCR	W	0 1	4 0	6 D	0 ers 0			
	LDC @ERs+,EXR	W	0 1	4 1	6 D	0 ers 0			
	LDC @aa:16,CCR	W	0 1	4 0	6 B	0 0	disp	disp	
	LDC @aa:16,EXR	W	0 1	4 1	6 B	0 0	disp	disp	

Instruction	Mnemonic	Size	Instruction Format											
			1st byte		2nd byte		3rd byte		4th byte		5th byte	6th byte	7th byte	8
LDC	LDC @aa:32,CCR	W	0	1	4	0	6	B	2	0	abs			
	LDC @aa:32,EXR	W	0	1	4	1	6	B	2	0	abs			
LDM	LDM.L @SP+, (ERn-ERn+1)	L	0	1	1	0	6	D	7	0 ern+1				
	LDM.L @SP+, (ERn-ERn+2)	L	0	1	2	0	6	D	7	0 ern+2				
	LDM.L @SP+, (ERn-ERn+3)	L	0	1	3	0	6	D	7	0 ern+3				
LDMAC	LDMAC ERs,MACH	L	0	3	2	0 ers								
	LDMAC ERs,MACL	L	0	3	3	0 ers								
MAC	MAC @ERn+,@ERm+	—	0	1	6	0	6	D	0 ern	0 erm				
MOV	MOV.B #xx:8,Rd	B	F	rd	IMM									
	MOV.B Rs,Rd	B	0	C	rs	rd								
	MOV.B @ERs,Rd	B	6	8	0 ers	rd								
	MOV.B @(d:16,ERs),Rd	B	6	E	0 ers	rd	disp							
	MOV.B @(d:32,ERs),Rd	B	7	8	0 ers	0	6	A	2	rd	disp			
	MOV.B @ERs+,Rd	B	6	C	0 ers	rd								
	MOV.B @aa:8,Rd	B	2	rd	abs									
	MOV.B @aa:16,Rd	B	6	A	0	rd	abs							
	MOV.B @aa:32,Rd	B	6	A	2	rd	abs							
	MOV.B Rs,@ERd	B	6	8	1 erd	rs								
	MOV.B Rs,@(d:16,ERd)	B	6	E	1 erd	rs	disp							
	MOV.B Rs,@(d:32,ERd)	B	7	8	0 erd	0	6	A	A	rs	disp			
	MOV.B Rs,@-ERd	B	6	C	1 erd	rs								
	MOV.B Rs,@aa:8	B	3	rs	abs									
	MOV.B Rs,@aa :16	B	6	A	8	rs	abs							
	MOV.B Rs,@aa:32	B	6	A	A	rs	abs							
	MOV.W #xx:16,Rd	W	7	9	0	rd	IMM							
	MOV.W Rs,Rd	W	0	D	rs	rd								
	MOV.W @ERs,Rd	W	6	9	0 ers	rd								
	MOV.W @(d:16,ERs),Rd	W	6	F	0 ers	rd	disp							

RENESAS

Instruction	Mnemonic	Size	Instruction Format																													
			1st byte			2nd byte			3rd byte			4th byte			5th byte			6th byte			7th byte			8									
MOV	MOV.W @ERs+,Rd	W	6	D	0	ers		rd																								
	MOV.W @aa:16,Rd	W	6	B	0			rd	abs																							
	MOV.W @aa:32,Rd	W	6	B	2			rd	abs																							
	MOV.W Rs,@ERd	W	6	9	1	erd		rs																								
	MOV.W Rs,@(d:16,ERd)	W	6	F	1	erd		rs	disp																							
	MOV.W Rs,@(d:32,ERd)	W	7	8	0	erd		0	6	B		A		rs	disp																	
	MOV.W Rs,@-ERd	W	6	D	1	erd		rs																								
	MOV.W Rs,@aa:16	W	6	B	8			rs	abs																							
	MOV.W Rs,@aa:32	W	6	B	A			rs	abs																							
	MOV.L #xx:32,Rd	L	7	A	0		0	erd	IMM																							
	MOV.L ERs,ERd	L	0	F	1	ers	0	erd																								
	MOV.L @ERs,ERd	L	0	1	0		0		6	9		0	ers	0	erd																	
	MOV.L @(d:16,ERs),ERd	L	0	1	0		0		6	F		0	ers	0	erd	disp																
	MOV.L @(d:32,ERs),ERd	L	0	1	0		0		7	8		0	ers		0	6	B		2	0	erd											
	MOV.L @ERs+,ERd	L	0	1	0		0		6	D		0	ers	0	erd																	
	MOV.L @aa:16 ,ERd	L	0	1	0		0		6	B		0		0	erd	abs																
	MOV.L @aa:32 ,ERd	L	0	1	0		0		6	B		2		0	erd	abs																
	MOV.L ERs,@ERd	L	0	1	0		0		6	9		1	erd	0	ers																	
	MOV.L ERs,@(d:16,ERd)	L	0	1	0		0		6	F		1	erd	0	ers	disp																
	MOV.L ERs,@(d:32,ERd)*	L	0	1	0		0		7	8		0	erd		0	6	B		A	0	ers											
	MOV.L ERs,@-ERd	L	0	1	0		0		6	D		1	erd	0	ers																	
	MOV.L ERs,@aa:16	L	0	1	0		0		6	B		8		0	ers	abs																
	MOV.L ERs,@aa:32	L	0	1	0		0		6	B		A		0	ers	abs																
MOVFPE	MOVFPE @aa:16,Rd	B	6	A	4			rd	abs																							
MOVTPE	MOVTPE Rs,@aa:16	B	6	A	C			rs	abs																							
MULXS	MULXS.B Rs,Rd	B	0	1	C		0		5	0		rs			rd																	
	MULXS.W Rs,ERd	W	0	1	C		0		5	2		rs		0	erd																	
MULXU	MULXU.B Rs,Rd	B	5	0	rs			rd																								
	MULXU.W Rs,ERd	W	5	2	rs		0	erd																								

Instruction	Mnemonic	Size	Instruction Format								
			1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8
NEG	NEG.B Rd	B	1 7	8 rd						
	NEG.W Rd	W	1 7	9 rd						
	NEG.L ERd	L	1 7	B 0 erd						
NOP	NOP	—	0 0	0 0						
NOT	NOT.B Rd	B	1 7	0 rd						
	NOT.W Rd	W	1 7	1 rd						
	NOT.L ERd	L	1 7	3 0 erd						
OR	OR.B #xx:8,Rd	B	C rd	IMM						
	OR.B Rs,Rd	B	1 4	rs rd						
	OR.W #xx:16,Rd	W	7 9	4 rd	IMM					
	OR.W Rs,Rd	W	6 4	rs rd						
	OR.L #xx:32,ERd	L	7 A	4 0 erd	IMM					
	OR.L ERs,ERd	L	0 1	F 0	6 4	0 ers 0 erd				
ORC	ORC #xx:8,CCR	B	0 4	IMM						
	ORC #xx:8,EXR	B	0 1	4 1	0 4	IMM				
POP	POP.W Rn	W	6 D	7 rn						
	POP.L ERn	L	0 1	0 0	6 D	7 0 ern				
PUSH	PUSH.W Rn	W	6 D	F rn						
	PUSH.L ERn	L	0 1	0 0	6 D	F 0 ern				
ROTL	ROTL.B Rd	B	1 2	8 rd						
	ROTL.B #2, Rd	B	1 2	C rd						
	ROTL.W Rd	W	1 2	9 rd						
	ROTL.W #2, Rd	W	1 2	D rd						
	ROTL.L ERd	L	1 2	B 0 erd						
	ROTL.L #2, ERd	L	1 2	F 0 erd						

RENESAS

Instruction	Mnemonic	Size	Instruction Format								
			1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8	
ROTR	ROTR.B Rd	B	1 3	8 rd							
	ROTR.B #2, Rd	B	1 3	C rd							
	ROTR.W Rd	W	1 3	9 rd							
	ROTR.W #2, Rd	W	1 3	D rd							
	ROTR.L ERd	L	1 3	B 0: erd							
	ROTR.L #2, ERd	L	1 3	F 0: erd							
ROTXL	ROTXL.B Rd	B	1 2	0 rd							
	ROTXL.B #2, Rd	B	1 2	4 rd							
	ROTXL.W Rd	W	1 2	1 rd							
	ROTXL.W #2, Rd	W	1 2	5 rd							
	ROTXL.L ERd	L	1 2	3 0: erd							
	ROTXL.L #2, ERd	L	1 2	7 0: erd							
ROTXR	ROTXR.B Rd	B	1 3	0 rd							
	ROTXR.B #2, Rd	B	1 3	4 rd							
	ROTXR.W Rd	W	1 3	1 rd							
	ROTXR.W #2, Rd	W	1 3	5 rd							
	ROTXR.L ERd	L	1 3	3 0: erd							
	ROTXR.L #2, ERd	L	1 3	7 0: erd							
RTE	RTE	—	5 6	7 0							
RTS	RTS	—	5 4	7 0							
SHAL	SHAL.B Rd	B	1 0	8 rd							
	SHAL.B #2, Rd	B	1 0	C rd							
	SHAL.W Rd	W	1 0	9 rd							
	SHAL.W #2, Rd	W	1 0	D rd							
	SHAL.L ERd	L	1 0	B 0: erd							
	SHAL.L #2, ERd	L	1 0	F 0: erd							

Instruction	Mnemonic	Size	1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte
SHAR	SHAR.B Rd	B	1 1	8 rd					
	SHAR.B #2, Rd	B	1 1	C rd					
	SHAR.W Rd	W	1 1	9 rd					
	SHAR.W #2, Rd	W	1 1	D rd					
	SHAR.L ERd	L	1 1	B 0 erd					
	SHAR.L #2, ERd	L	1 1	F 0 erd					
SHLL	SHLL.B Rd	B	1 0	0 rd					
	SHLL.B #2, Rd	B	1 0	4 rd					
	SHLL.W Rd	W	1 0	1 rd					
	SHLL.W #2, Rd	W	1 0	5 rd					
	SHLL.L ERd	L	1 0	3 0 erd					
	SHLL.L #2, ERd	L	1 0	7 0 erd					
SHLR	SHLR.B Rd	B	1 1	0 rd					
	SHLR.B #2, Rd	B	1 1	4 rd					
	SHLR.W Rd	W	1 1	1 rd					
	SHLR.W #2, Rd	W	1 1	5 rd					
	SHLR.L ERd	L	1 1	3 0 erd					
	SHLR.L #2, ERd	L	1 1	7 0 erd					
SLEEP	SLEEP	—	0 1	8 0					
STC	STC.B CCR,Rd	B	0 2	0 rd					
	STC.B EXR,Rd	B	0 2	1 rd					
	STC.W CCR,@ERd	W	0 1	4 0	6 9	1 erd 0			
	STC.W EXR,@ERd	W	0 1	4 1	6 9	1 erd 0			
	STC.W CCR,@(d:16,ERd)	W	0 1	4 0	6 F	1 erd 0	disp		
	STC.W EXR,@(d:16,ERd)	W	0 1	4 1	6 F	1 erd 0	disp		
	STC.W CCR,@(d:32,ERd)	W	0 1	4 0	7 8	0 erd 0	6 B	A 0	
	STC.W EXR,@(d:32,ERd)	W	0 1	4 1	7 8	0 erd 0	6 B	A 0	
	STC.W CCR,@-ERd	W	0 1	4 0	6 D	1 erd 0			

Instruc-tion	Mnemonic	Size	Instruction Format								
			1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8
STC	STC.W CCR,@aa:16	W	0 1	4 0	6 B	8 0	abs			
	STC.W EXR,@aa:16	W	0 1	4 1	6 B	8 0	abs			
	STC.W CCR,@aa:32	W	0 1	4 0	6 B	A 0	abs			
	STC.W EXR,@aa:32	W	0 1	4 1	6 B	A 0	abs			
STM	STM.L(ERn-ERn+1), @-SP	L	0 1	1 0	6 D	F 0: ern				
	STM.L (ERn-ERn+2), @-SP	L	0 1	2 0	6 D	F 0: ern				
	STM.L (ERn-ERn+3), @-SP	L	0 1	3 0	6 D	F 0: ern				
STMAC	STMAC MACH,ERd	L	0 2	2 0: ers						
	STMAC MACL,ERd	L	0 2	3 0: ers						
SUB	SUB.B Rs,Rd	B	1 8	rs rd						
	SUB.W #xx:16,Rd	W	7 9	3 rd	IMM					
	SUB.W Rs,Rd	W	1 9	rs rd						
	SUB.L #xx:32,ERd	L	7 A	3 0: erd	IMM					
	SUB.L ERs,ERd	L	1 A	1: ers 0: erd						
SUBS	SUBS #1,ERd	L	1 B	0 0: erd						
	SUBS #2,ERd	L	1 B	8 0: erd						
	SUBS #4,ERd	L	1 B	9 0: erd						
SUBX	SUBX #xx:8,Rd	B	B rd	IMM						
	SUBX Rs,Rd	B	1 E	rs rd						
TAS	TAS @ERd	B	0 1	E 0	7 B	0: erd C				
TRAPA	TRAPA #x:2	—	5 7	00:IMM: 0						
XOR	XOR.B #xx:8,Rd	B	D rd	IMM						
	XOR.B Rs,Rd	B	1 5	rs rd						
	XOR.W #xx:16,Rd	W	7 9	5 rd	IMM					
	XOR.W Rs,Rd	W	6 5	rs rd						
	XOR.L #xx:32,ERd	L	7 A	5 0: erd	IMM					
	XOR.L ERs,ERd	L	0 1	F 0	6 5	0: ers 0: erd				

Instruc-tion	Mnemonic	Size	Instruction Format													
			1st byte		2nd byte		3rd byte		4th byte	5th byte	6th byte	7th byte	8
XORC	XORC #xx:8,CCR	B	0	5	IMM								
	XORC #xx:8,EXR	B	0	1	4	1	0	5	IMM				

Legend:

IMM: Immediate data (2, 3, 8, 16, or 32 bits)

abs: Absolute address (8, 16, 24, or 32 bits)

disp: Displacement (8, 16, or 32 bits)

rs, rd, rn: Register field (4 bits specifying an 8-bit or 16-bit register. The symbols rs, rd, and rn correspond to ope

ers, erd, ern, erm: Register field (3 bits specifying an address register or 32-bit register. The symbols ers, erd, ern, and e symbols ERs, ERd, ERn, and ERm.)

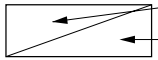
Note: * Bit 7 of the 4th byte of the MOV.L ERs, @(d:32,ERd) instruction can be either 1 or 0.

The register fields specify general registers as follows.

Address Register 32-Bit Register		16-Bit Register		8-Bit Register	
Register Field	General Register	Register Field	General Register	Register Field	General Register
000	ER0	0000	R0	0000	R0H
001	ER1	0001	R1	0001	R1H
•	•	•	•	•	•
•	•	•	•	•	•
•	•	•	•	•	•
111	ER7	0111	R7	0111	R7H
		1000	E0	1000	R0L
		1001	E1	1001	R1L
		•	•	•	•
		•	•	•	•
		•	•	•	•
		1111	E7	1111	R7L

Instruction code

1st byte		2nd byte	
AH	AL	BH	BL



Instruction when most significant bit of BH is 0.

Instruction when most significant bit of BH is 1.

AH \ AL	0	1	2	3	4	5	6	7	8	9	A	B	C	D
0	NOP	Table A.3(2)	STC / STMAC	LDC / LDMAC	ORC	XORC	ANDC	LDC	ADD		Table A.3(2)	Table A.3(2)	MOV	
1	Table A.3(2)	Table A.3(2)	Table A.3(2)	Table A.3(2)	OR	XOR	AND	Table A.3(2)	SUB		Table A.3(2)	Table A.3(2)	CMP	
2	MOV.B													
3														
4	BRA	BRN	BHI	BLS	BCC	BCS	BNE	BEQ	BVC	BVS	BPL	BMI	BGE	BLT
5	MULXU	DIVXU	MULXU	DIVXU	RTS	BSR	RTE	TRAPA	Table A.3(2)	JMP			BSR	
6	BSET	BNOT	BCLR	BTST	OR	XOR	AND	BST / BIST	MOV		Table A.3(2)	MOV		
7					BOR / BIOR	BXOR / BIXOR	BAND / BIAND	BLD / BILD	MOV	Table A.3(2)	Table A.3(2)	EEPMOV	Table A.	
8	ADD													
9	ADDX													
A	CMP													
B	SUBX													
C	OR													
D	XOR													
E	AND													
F	MOV													

Instruction code

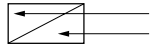
1st byte		2nd byte	
AH	AL	BH	BL

AH AL \ BH	0	1	2	3	4	5	6	7	8	9	A	B	C
01	MOV	LDM		STM	LDC / STC		MAC		SLEEP		CLRMAC		Table A.3(3)
0A	INC					INC		INC		ADDS		ADD	
0B	ADDS								ADDS				
0F	DAA												
10	SHLL				SHLL			SHLL	SHAL			SHAL	
11	SHLR				SHLR			SHLR	SHAR			SHAR	
12	ROTXL				ROTXL			ROTXL	ROTL			ROTL	
13	ROTXR				ROTXR			ROTXR	ROTR			ROTR	
17	NOT			NOT		EXTU		EXTU	NEG			NEG	
1A	DEC									SUB			
1B	SUBS					DEC		DEC	SUBS				
1F	DAS											CMP	
58	BRA	BRN	BHI	BLS	BCC	BCS	BNE	BEQ	BVC	BVS	BPL	BMI	BGE
6A	MOV	Table A.3(4)	MOV	Table A.3(4)	MOVFPE				MOV		MOV		MOVTPE
79	MOV	ADD	CMP	SUB	OR	XOR	AND						
7A	MOV	ADD	CMP	SUB	OR	XOR	AND						

RENESAS

Instruction code

1st byte		2nd byte		3rd byte		4th byte	
AH	AL	BH	BL	CH	CL	DH	DL



CL / AH AL BH BL CH	0	1	2	3	4	5	6	7	8	9	A	B	C
01C05	MULXS		MULXS										
01D05		DIVXS		DIVXS									
01F06					OR	XOR	AND						
7Cr06*1				BTST									
7Cr07*1				BTST	BOR / BIOR	BXOR / BIXOR	BAND / BIAND	BLD / BILD					
7Dr06*1	BSET	BNOT	BCLR					BST / BIST					
7Dr07*1	BSET	BNOT	BCLR										
7Eaa6*2				BTST									
7Eaa7*2				BTST	BOR / BIOR	BXOR / BIXOR	BAND / BIAND	BLD / BILD					
7Faa6*2	BSET	BNOT	BCLR					BST / BIST					
7Faa7*2	BSET	BNOT	BCLR										

Notes: 1. r is the register specification field.
2. aa is the absolute address specification.

Instruction code	1st byte		2nd byte		3rd byte		4th byte		5th byte		6th byte	
	AH	AL	BH	BL	CH	CL	DH	DL	EH	EL	FH	FL

Instruction whe
Instruction whe

AHALBHBLCHCLDHDLEH \ EL	0	1	2	3	4	5	6	7	8	9	A	B	C
6A10aaaa6*				BTST									
6A10aaaa7*					BOR / BIOR	BXOR / BIXOR	BAND / BIAND	BLD / BILD					
6A18aaaa6*	BSET	BNOT	BCLR					BST / BIST					
6A18aaaa7*													

Instruction code	1st byte		2nd byte		3rd byte		4th byte		5th byte		6th byte		7th byte		8th b
	AH	AL	BH	BL	CH	CL	DH	DL	EH	EL	FH	FL	GH	GL	HH

Instruction whe
Instruction whe

AHALBHBL ... FHFLGH \ GL	0	1	2	3	4	5	6	7	8	9	A	B	C
6A30aaaaaaaa6*				BTST									
6A30aaaaaaaa7*					BOR / BIOR	BXOR / BIXOR	BAND / BIAND	BLD / BILD					
6A38aaaaaaaa6*	BSET	BNOT	BCLR					BST / BIST					
6A38aaaaaaaa7*													

I J K L M N

Examples: Advanced mode, program code and stack located in external memory, on-c
supporting modules accessed in two states with 8-bit bus width, external devices access
states with one wait state and 16-bit bus width.

1. BSET #0, @FFFFC7:8

   From table A.5:

   $I = L = 2, \quad J = K = M = N = 0$

   From table A.4:

   $S_I = 4, \quad S_L = 2$

   Number of states required for execution = $2 \times 4 + 2 \times 2 = 12$

2. JSR @@30

   From table A.5:

   $I = J = K = 2, \quad L = M = N = 0$

   From table A.4:

   $S_I = S_J = S_K = 4$

   Number of states required for execution = $2 \times 4 + 2 \times 4 + 2 \times 4 = 24$

Branch address read	S_J							
Stack operation	S_K							
Byte data access	S_L		2			2	3 + m	
Word data access	S_M		4			4	6 + 2m	
Internal operation	S_N	1	1	1		1	1	1

Legend:

m: Number of wait states inserted into external device access

	ADD.L ERs,ERd	1		
ADDS	ADDS #1/2/4,ERd	1		
ADDX	ADDX #xx:8,Rd	1		
	ADDX Rs,Rd	1		
AND	AND.B #xx:8,Rd	1		
	AND.B Rs,Rd	1		
	AND.W #xx:16,Rd	2		
	AND.W Rs,Rd	1		
	AND.L #xx:32,ERd	3		
	AND.L ERs,ERd	2		
ANDC	ANDC #xx:8,CCR	1		
	ANDC #xx:8,EXR	2		
BAND	BAND #xx:3,Rd	1		
	BAND #xx:3,@ERd	2		1
	BAND #xx:3,@aa:8	2		1
	BAND #xx:3,@aa:16	3		1
	BAND #xx:3,@aa:32	4		1
Bcc	BRA d:8 (BT d:8)	2		
	BRN d:8 (BF d:8)	2		
	BHI d:8	2		
	BLS d:8	2		
	BCC d:8 (BHS d:8)	2		
	BCS d:8 (BLO d:8)	2		
	BNE d:8	2		
	BEQ d:8	2		
	BVC d:8	2		
	BVS d:8	2		
	BPL d:8	2		
	BMI d:8	2		
	BGE d:8	2		
	BLT d:8	2		
	BGT d:8	2		
	BLE d:8	2		

	BEQ d:16	2	
	BVC d:16	2	
	BVS d:16	2	
	BPL d:16	2	
	BMI d:16	2	
	BGE d:16	2	
	BLT d:16	2	
	BGT d:16	2	
	BLE d:16	2	
BCLR	BCLR #xx:3,Rd	1	
	BCLR #xx:3,@ERd	2	2
	BCLR #xx:3,@aa:8	2	2
	BCLR #xx:3,@aa:16	3	2
	BCLR #xx:3,@aa:32	4	2
	BCLR Rn,Rd	1	
	BCLR Rn,@ERd	2	2
	BCLR Rn,@aa:8	2	2
	BCLR Rn,@aa:16	3	2
	BCLR Rn,@aa:32	4	2
BIAND	BIAND #xx:3,Rd	1	
	BIAND #xx:3,@ERd	2	1
	BIAND #xx:3,@aa:8	2	1
	BIAND #xx:3,@aa:16	3	1
	BIAND #xx:3,@aa:32	4	1
BILD	BILD #xx:3,Rd	1	
	BILD #xx:3,@ERd	2	1
	BILD #xx:3,@aa:8	2	1
	BILD #xx:3,@aa:16	3	1
	BILD #xx:3,@aa:32	4	1

	BIST #xx:3,@aa:8	2	2
	BIST #xx:3,@aa:16	3	2
	BIST #xx:3,@aa:32	4	2
BIXOR	BIXOR #xx:3,Rd	1	
	BIXOR #xx:3,@ERd	2	1
	BIXOR #xx:3,@aa:8	2	1
	BIXOR #xx:3,@aa:16	3	1
	BIXOR #xx:3,@aa:32	4	1
BLD	BLD #xx:3,Rd	1	
	BLD #xx:3,@ERd	2	1
	BLD #xx:3,@aa:8	2	1
	BLD #xx:3,@aa:16	3	1
	BLD #xx:3,@aa:32	4	1
BNOT	BNOT #xx:3,Rd	1	
	BNOT #xx:3,@ERd	2	2
	BNOT #xx:3,@aa:8	2	2
	BNOT #xx:3,@aa:16	3	2
	BNOT #xx:3,@aa:32	4	2
	BNOT Rn,Rd	1	
	BNOT Rn,@ERd	2	2
	BNOT Rn,@aa:8	2	2
	BNOT Rn,@aa:16	3	2
	BNOT Rn,@aa:32	4	2
BOR	BOR #xx:3,Rd	1	
	BOR #xx:3,@ERd	2	1
	BOR #xx:3,@aa:8	2	1
	BOR #xx:3,@aa:16	3	1
	BOR #xx:3,@aa:32	4	1

	BSET Rn,@aa:8		2		2
	BSET Rn,@aa:16		3		2
	BSET Rn,@aa:32		4		2
BSR	BSR d:8	Normal	2	1	
		Advanced	2	2	
	BSR d:16	Normal	2	1	
		Advanced	2	2	
BST	BST #xx:3,Rd		1		
	BST #xx:3,@ERd		2		2
	BST #xx:3,@aa:8		2		2
	BST #xx:3,@aa:16		3		2
	BST #xx:3,@aa:32		4		2
BTST	BTST #xx:3,Rd		1		
	BTST #xx:3,@ERd		2		1
	BTST #xx:3,@aa:8		2		1
	BTST #xx:3,@aa:16		3		1
	BTST #xx:3,@aa:32		4		1
	BTST Rn,Rd		1		
	BTST Rn,@ERd		2		1
	BTST Rn,@aa:8		2		1
	BTST Rn,@aa:16		3		1
	BTST Rn,@aa:32		4		1
BXOR	BXOR #xx:3,Rd		1		
	BXOR #xx:3,@ERd		2		1
	BXOR #xx:3,@aa:8		2		1
	BXOR #xx:3,@aa:16		3		1
	BXOR #xx:3,@aa:32		4		1
CLRMAC	CLRMAC		1		

DAS	DAS Rd		1			
DEC	DEC.B Rd		1			
	DEC.W #1/2,Rd		1			
	DEC.L #1/2,ERd		1			
DIVXS	DIVXS.B Rs,Rd		2			
	DIVXS.W Rs,ERd		2			
DIVXU	DIVXU.B Rs,Rd		1			
	DIVXU.W Rs,ERd		1			
EEPMOV	EEPMOV.B		2			2n + 2*1
	EEPMOV.W		2			2n + 2*1
EXTS	EXTS.W Rd		1			
	EXTS.L ERd		1			
EXTU	EXTU.W Rd		1			
	EXTU.L ERd		1			
INC	INC.B Rd		1			
	INC.W #1/2,Rd		1			
	INC.L #1/2,ERd		1			
JMP	JMP @ERn		2			
	JMP @aa:24		2			
	JMP @@aa:8	Normal	2	1		
		Advanced	2	2		
JSR	JSR @ERn	Normal	2		1	
		Advanced	2		2	
	JSR @aa:24	Normal	2		1	
		Advanced	2		2	
	JSR @@aa:8	Normal	2	1	1	
		Advanced	2	2	2	

	LDC @(d:16,ERs),EXR	3				1
	LDC @(d:32,ERs),CCR	5				1
	LDC @(d:32,ERs),EXR	5				1
	LDC @ERs+,CCR	2				1
	LDC @ERs+,EXR	2				1
	LDC @aa:16,CCR	3				1
	LDC @aa:16,EXR	3				1
	LDC @aa:32,CCR	4				1
	LDC @aa:32,EXR	4				1
LDM	LDM.L @SP+, (ERn-ERn+1)	2	4			
	LDM.L @SP+, (ERn-ERn+2)	2	6			
	LDM.L @SP+, (ERn-ERn+3)	2	8			
LDMAC	LDMAC ERs,MACH	1				
	LDMAC ERs,MACL	1				
MAC	MAC @ERn+,@ERm+	2				2
MOV	MOV.B #xx:8,Rd	1				
	MOV.B Rs,Rd	1				
	MOV.B @ERs,Rd	1		1		
	MOV.B @(d:16,ERs),Rd	2		1		
	MOV.B @(d:32,ERs),Rd	4		1		
	MOV.B @ERs+,Rd	1		1		
	MOV.B @aa:8,Rd	1		1		
	MOV.B @aa:16,Rd	2		1		
	MOV.B @aa:32,Rd	3		1		
	MOV.B Rs,@ERd	1		1		
	MOV.B Rs,@(d:16,ERd)	2		1		
	MOV.B Rs,@(d:32,ERd)	4		1		
	MOV.B Rs,@-ERd	1		1		
	MOV.B Rs,@aa:8	1		1		
	MOV.B Rs,@aa:16	2		1		
	MOV.B Rs,@aa:32	3		1		
	MOV.W #xx:16,Rd	2				

	MOV.W Rs,@ERd	1	1
	MOV.W Rs,@(d:16,ERd)	2	1
	MOV.W Rs,@(d:32,ERd)	4	1
	MOV.W Rs,@-ERd	1	1
	MOV.W Rs,@aa:16	2	1
	MOV.W Rs,@aa:32	3	1
	MOV.L #xx:32,ERd	3	
	MOV.L ERs,ERd	1	
	MOV.L @ERs,ERd	2	2
	MOV.L @(d:16,ERs),ERd	3	2
	MOV.L @(d:32,ERs),ERd	5	2
	MOV.L @ERs+,ERd	2	2
	MOV.L @aa:16,ERd	3	2
	MOV.L @aa:32,ERd	4	2
	MOV.L ERs,@ERd	2	2
	MOV.L ERs,@(d:16,ERd)	3	2
	MOV.L ERs,@(d:32,ERd)	5	2
	MOV.L ERs,@-ERd	2	2
	MOV.L ERs,@aa:16	3	2
	MOV.L ERs,@aa:32	4	2
MOVFPE	MOVFPE @:aa:16,Rd	Can not be used in the H8S/2655 Group.	
MOVTPE	MOVTPE Rs,@:aa:16		
MULXS	MULXS.B Rs,Rd	2	
	MULXS.W Rs,ERd	2	
MULXU	MULXU.B Rs,Rd	1	
	MULXU.W Rs,ERd	1	
NEG	NEG.B Rd	1	
	NEG.W Rd	1	
	NEG.L ERd	1	
NOP	NOP	1	

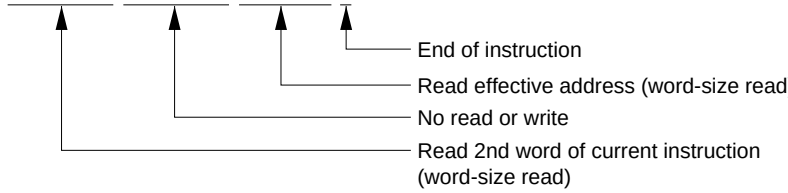
	OR.L #xx:32,ERd	3		
	OR.L ERs,ERd	2		
ORC	ORC #xx:8,CCR	1		
	ORC #xx:8,EXR	2		
POP	POP.W Rn	1		1
	POP.L ERn	2		2
PUSH	PUSH.W Rn	1		1
	PUSH.L ERn	2		2
ROTL	ROTL.B Rd	1		
	ROTL.B #2,Rd	1		
	ROTL.W Rd	1		
	ROTL.W #2,Rd	1		
	ROTL.L ERd	1		
	ROTL.L #2,ERd	1		
ROTR	ROTR.B Rd	1		
	ROTR.B #2,Rd	1		
	ROTR.W Rd	1		
	ROTR.W #2,Rd	1		
	ROTR.L ERd	1		
	ROTR.L #2,ERd	1		
ROTXL	ROTXL.B Rd	1		
	ROTXL.B #2,Rd	1		
	ROTXL.W Rd	1		
	ROTXL.W #2,Rd	1		
	ROTXL.L ERd	1		
	ROTXL.L #2,ERd	1		
ROTXR	ROTXR.B Rd	1		
	ROTXR.B #2,Rd	1		
	ROTXR.W Rd	1		
	ROTXR.W #2,Rd	1		
	ROTXR.L ERd	1		
	ROTXR.L #2,ERd	1		

	SHAL.L ERd	1		
	SHAL.L #2,ERd	1		
SHAR	SHAR.B Rd	1		
	SHAR.B #2,Rd	1		
	SHAR.W Rd	1		
	SHAR.W #2,Rd	1		
	SHAR.L ERd	1		
	SHAR.L #2,ERd	1		
SHLL	SHLL.B Rd	1		
	SHLL.B #2,Rd	1		
	SHLL.W Rd	1		
	SHLL.W #2,Rd	1		
	SHLL.L ERd	1		
	SHLL.L #2,ERd	1		
SHLR	SHLR.B Rd	1		
	SHLR.B #2,Rd	1		
	SHLR.W Rd	1		
	SHLR.W #2,Rd	1		
	SHLR.L ERd	1		
	SHLR.L #2,ERd	1		
SLEEP	SLEEP	1		
STC	STC.B CCR,Rd	1		
	STC.B EXR,Rd	1		
	STC.W CCR,@ERd	2		1
	STC.W EXR,@ERd	2		1
	STC.W CCR,@(d:16,ERd)	3		1
	STC.W EXR,@(d:16,ERd)	3		1
	STC.W CCR,@(d:32,ERd)	5		1
	STC.W EXR,@(d:32,ERd)	5		1
	STC.W CCR,@-ERd	2		1
	STC.W EXR,@-ERd	2		1

STMAC*3	STMAC MACH,ERd		1			
	STMAC MACL,ERd		1			
SUB	SUB.B Rs,Rd		1			
	SUB.W #xx:16,Rd		2			
	SUB.W Rs,Rd		1			
	SUB.L #xx:32,ERd		3			
	SUB.L ERs,ERd		1			
SUBS	SUBS #1/2/4,ERd		1			
SUBX	SUBX #xx:8,Rd		1			
	SUBX Rs,Rd		1			
TAS	TAS @ERd		2			2
TRAPA	TRAPA #x:2	Normal	2	1	2/3*1	
		Advanced	2	2	2/3*1	
XOR	XOR.B #xx:8,Rd		1			
	XOR.B Rs,Rd		1			
	XOR.W #xx:16,Rd		2			
	XOR.W Rs,Rd		1			
	XOR.L #xx:32,ERd		3			
	XOR.L ERs,ERd		2			
XORC	XORC #xx:8,CCR		1			
	XORC #xx:8,EXR		2			

Notes: 1. 2 when EXR is invalid, 3 when EXR is valid.
2. 5 for concatenated execution, 4 otherwise.
3. An internal operation may require between 0 and 3 additional states, deper preceding instruction.

Instruction	1	2	3	4	5	6	7
JMP@aa:24	R:W 2nd	Internal operation 1 state	R:W EA				



Legend

R:B	Byte-size read
R:W	Word-size read
W:B	Byte-size write
W:W	Word-size write
:M	Transfer of the bus is not performed immediately after this cycle
2nd	Address of 2nd word (3rd and 4th bytes)
3rd	Address of 3rd word (5th and 6th bytes)
4th	Address of 4th word (7th and 8th bytes)
5th	Address of 5th word (9th and 10th bytes)
NEXT	Address of next instruction
EA	Effective address
VEC	Vector address

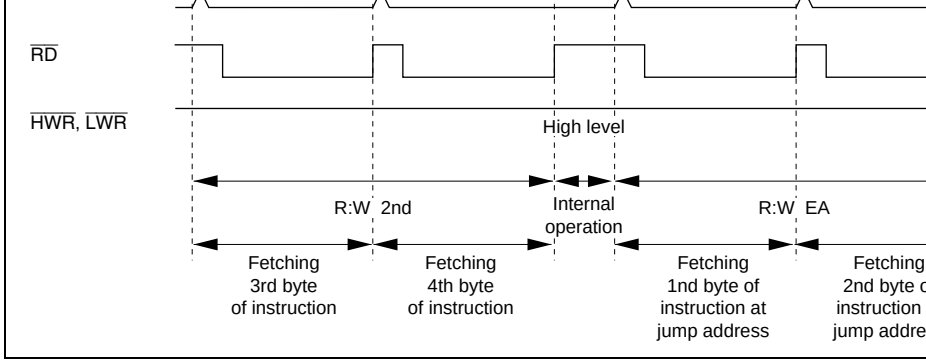


Figure A.1 Address Bus, $\overline{RD}$, $\overline{HWR}$, and $\overline{LWR}$ Timing (8-Bit Bus, Three-State Access, No Wait States)

Instruction	1	2	3	4	5	6	7
ADD.B #xx:8,Rd	R:W NEXT						
ADD.B Rs,Rd	R:W NEXT						
ADD.W #xx:16,Rd	R:W 2nd	R:W NEXT					
ADD.W Rs,Rd	R:W NEXT						
ADD.L #xx:32,ERd	R:W 2nd	R:W 3rd	R:W NEXT				
ADD.L ERs,ERd	R:W NEXT						
ADDS #1/2/4,ERd	R:W NEXT						
ADDX #xx:8,Rd	R:W NEXT						
ADDX Rs,Rd	R:W NEXT						
AND.B #xx:8,Rd	R:W NEXT						
AND.B Rs,Rd	R:W NEXT						
AND.W #xx:16,Rd	R:W 2nd	R:W NEXT					
AND.W Rs,Rd	R:W NEXT						
AND.L #xx:32,ERd	R:W 2nd	R:W 3rd	R:W NEXT				
AND.L ERs,ERd	R:W 2nd	R:W NEXT					
ANDC #xx:8,CCR	R:W NEXT						
ANDC #xx:8,EXR	R:W 2nd	R:W NEXT					
BAND #xx:3,Rd	R:W NEXT						
BAND #xx:3,@ERd	R:W 2nd	R:B EA	R:W:M NEXT				
BAND #xx:3,@aa:8	R:W 2nd	R:B EA	R:W:M NEXT				
BAND #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B EA	R:W:M NEXT			
BAND #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B EA	R:W:M NEXT		
BRA d:8 (BT d:8)	R:W NEXT	R:W EA					
BRN d:8 (BF d:8)	R:W NEXT	R:W EA					
BHI d:8	R:W NEXT	R:W EA					
BLS d:8	R:W NEXT	R:W EA					
BCC d:8 (BHS d:8)	R:W NEXT	R:W EA					
BCS d:8 (BLO d:8)	R:W NEXT	R:W EA					
BNE d:8	R:W NEXT	R:W EA					
BEQ d:8	R:W NEXT	R:W EA					
BVC d:8	R:W NEXT	R:W EA					
BVS d:8	R:W NEXT	R:W EA					
BPL d:8	R:W NEXT	R:W EA					
BMI d:8	R:W NEXT	R:W EA					
BGE d:8	R:W NEXT	R:W EA					
BLT d:8	R:W NEXT	R:W EA					
BGT d:8	R:W NEXT	R:W EA					

Instruction	1	2	3	4	5	6	7
BLE d:8	R:W NEXT	R:W EA					
BRA d:16 (BT d:16)	R:W 2nd	Internal operation, 1 state	R:W EA				
BRN d:16 (BF d:16)	R:W 2nd	Internal operation, 1 state	R:W EA				
BHI d:16	R:W 2nd	Internal operation, 1 state	R:W EA				
BLS d:16	R:W 2nd	Internal operation, 1 state	R:W EA				
BCC d:16 (BHS d:16)	R:W 2nd	Internal operation, 1 state	R:W EA				
BCS d:16 (BLO d:16)	R:W 2nd	Internal operation, 1 state	R:W EA				
BNE d:16	R:W 2nd	Internal operation, 1 state	R:W EA				
BEQ d:16	R:W 2nd	Internal operation, 1 state	R:W EA				
BVC d:16	R:W 2nd	Internal operation, 1 state	R:W EA				
BVS d:16	R:W 2nd	Internal operation, 1 state	R:W EA				
BPL d:16	R:W 2nd	Internal operation, 1 state	R:W EA				
BMI d:16	R:W 2nd	Internal operation, 1 state	R:W EA				
BGE d:16	R:W 2nd	Internal operation, 1 state	R:W EA				
BLT d:16	R:W 2nd	Internal operation, 1 state	R:W EA				
BGT d:16	R:W 2nd	Internal operation, 1 state	R:W EA				
BLE d:16	R:W 2nd	Internal operation, 1 state	R:W EA				
BCLR #xx:3,Rd	R:W NEXT						
BCLR #xx:3,@ERd	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA			

Instruction	1	2	3	4	5	6	7
BCLR #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B:M EA	R:W:M NEXT	W:B EA	
BCLR Rn,Rd	R:W NEXT						
BCLR Rn,@ERd	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA			
BCLR Rn,@aa:8	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA			
BCLR Rn,@aa:16	R:W 2nd	R:W 3rd	R:B:M EA	R:W:M NEXT	W:B EA		
BCLR Rn,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B:M EA	R:W:M NEXT	W:B EA	
BIAND #xx:3,Rd	R:W NEXT						
BIAND #xx:3,@ERd	R:W 2nd	R:B EA	R:W:M NEXT				
BIAND #xx:3,@aa:8	R:W 2nd	R:B EA	R:W:M NEXT				
BIAND #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B EA	R:W:M NEXT			
BIAND #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B EA	R:W:M NEXT		
BILD #xx:3,Rd	R:W NEXT						
BILD #xx:3,@ERd	R:W 2nd	R:B EA	R:W:M NEXT				
BILD #xx:3,@aa:8	R:W 2nd	R:B EA	R:W:M NEXT				
BILD #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B EA	R:W:M NEXT			
BILD #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B EA	R:W:M NEXT		
BIOR #xx:3,Rd	R:W NEXT						
BIOR #xx:3,@ERd	R:W 2nd	R:B EA	R:W:M NEXT				
BIOR #xx:3,@aa:8	R:W 2nd	R:B EA	R:W:M NEXT				
BIOR #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B EA	R:W:M NEXT			
BIOR #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B EA	R:W:M NEXT		
BIST #xx:3,Rd	R:W NEXT						
BIST #xx:3,@ERd	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA			
BIST #xx:3,@aa:8	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA			
BIST #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B:M EA	R:W:M NEXT	W:B EA		
BIST #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B:M EA	R:W:M NEXT	W:B EA	
BIXOR #xx:3,Rd	R:W NEXT						
BIXOR #xx:3,@ERd	R:W 2nd	R:B EA	R:W:M NEXT				
BIXOR #xx:3,@aa:8	R:W 2nd	R:B EA	R:W:M NEXT				
BIXOR #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B EA	R:W:M NEXT			
BIXOR #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B EA	R:W:M NEXT		
BLD #xx:3,Rd	R:W NEXT						
BLD #xx:3,@ERd	R:W 2nd	R:B EA	R:W:M NEXT				
BLD #xx:3,@aa:8	R:W 2nd	R:B EA	R:W:M NEXT				
BLD #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B EA	R:W:M NEXT			

Instruction		1	2	3	4	5	6	7
BNOT #xx:3,@ERd		R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA			
BNOT #xx:3,@aa:8		R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA			
BNOT #xx:3,@aa:16		R:W 2nd	R:W 3rd	R:B:M EA	R:W:M NEXT	W:B EA		
BNOT #xx:3,@aa:32		R:W 2nd	R:W 3rd	R:W 4th	R:B:M EA	R:W:M NEXT	W:B EA	
BNOT Rn,Rd		R:W NEXT						
BNOT Rn,@ERd		R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA			
BNOT Rn,@aa:8		R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA			
BNOT Rn,@aa:16		R:W 2nd	R:W 3rd	R:B:M EA	R:W:M NEXT	W:B EA		
BNOT Rn,@aa:32		R:W 2nd	R:W 3rd	R:W 4th	R:B:M EA	R:W:M NEXT	W:B EA	
BOR #xx:3,Rd		R:W NEXT						
BOR #xx:3,@ERd		R:W 2nd	R:B EA	R:W:M NEXT				
BOR #xx:3,@aa:8		R:W 2nd	R:B EA	R:W:M NEXT				
BOR #xx:3,@aa:16		R:W 2nd	R:W 3rd	R:B EA	R:W:M NEXT			
BOR #xx:3,@aa:32		R:W 2nd	R:W 3rd	R:W 4th	R:B EA	R:W:M NEXT		
BSET #xx:3,Rd		R:W NEXT						
BSET #xx:3,@ERd		R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA			
BSET #xx:3,@aa:8		R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA			
BSET #xx:3,@aa:16		R:W 2nd	R:W 3rd	R:B:M EA	R:W:M NEXT	W:B EA		
BSET #xx:3,@aa:32		R:W 2nd	R:W 3rd	R:W 4th	R:B:M EA	R:W:M NEXT	W:B EA	
BSET Rn,Rd		R:W NEXT						
BSET Rn,@ERd		R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA			
BSET Rn,@aa:8		R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA			
BSET Rn,@aa:16		R:W 2nd	R:W 3rd	R:B:M EA	R:W:M NEXT	W:B EA		
BSET Rn,@aa:32		R:W 2nd	R:W 3rd	R:W 4th	R:B:M EA	R:W:M NEXT	W:B EA	
BSR d:8	Normal	R:W NEXT	R:W EA	W:W stack				
	Advanced	R:W NEXT	R:W EA	W:W:M stack (H)	W:W stack (L)			
BSR d:16	Normal	R:W 2nd	Internal operation, 1 state	R:W EA	W:W stack			
	Advanced	R:W 2nd	Internal operation, 1 state	R:W EA	W:W:M stack (H)	W:W stack (L)		
BST #xx:3,Rd		R:W NEXT						
BST #xx:3,@ERd		R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA			
BST #xx:3,@aa:8		R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA			
BST #xx:3,@aa:16		R:W 2nd	R:W 3rd	R:B:M EA	R:W:M NEXT	W:B EA		
BST #xx:3,@aa:32		R:W 2nd	R:W 3rd	R:W 4th	R:B:M EA	R:W:M NEXT	W:B EA	

Instruction	1	2	3	4	5	6	7
BTST #xx:3,@aa:8	R:W 2nd	R:B EA	R:W:M NEXT				
BTST #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B EA	R:W:M NEXT			
BTST #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B EA	R:W:M NEXT		
BTST Rn,Rd	R:W NEXT						
BTST Rn,@ERd	R:W 2nd	R:B EA	R:W:M NEXT				
BTST Rn,@aa:8	R:W 2nd	R:B EA	R:W:M NEXT				
BTST Rn,@aa:16	R:W 2nd	R:W 3rd	R:B EA	R:W:M NEXT			
BTST Rn,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B EA	R:W:M NEXT		
BXOR #xx:3,Rd	R:W NEXT						
BXOR #xx:3,@ERd	R:W 2nd	R:B EA	R:W:M NEXT				
BXOR #xx:3,@aa:8	R:W 2nd	R:B EA	R:W:M NEXT				
BXOR #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B EA	R:W:M NEXT			
BXOR #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B EA	R:W:M NEXT		
CLRMAC	R:W NEXT	Internal operation, 1 state					
CMP.B #xx:8,Rd	R:W NEXT						
CMP.B Rs,Rd	R:W NEXT						
CMP.W #xx:16,Rd	R:W 2nd	R:W NEXT					
CMP.W Rs,Rd	R:W NEXT						
CMP.L #xx:32,ERd	R:W 2nd	R:W 3rd	R:W NEXT				
CMP.L ERs,ERd	R:W NEXT						
DAA Rd	R:W NEXT						
DAS Rd	R:W NEXT						
DEC.B Rd	R:W NEXT						
DEC.W #1/2,Rd	R:W NEXT						
DEC.L #1/2,ERd	R:W NEXT						
DIVXS.B Rs,Rd	R:W 2nd	R:W NEXT	Internal operation, 11 states				
DIVXS.W Rs,ERd	R:W 2nd	R:W NEXT	Internal operation, 19 states				
DIVXU.B Rs,Rd	R:W NEXT	Internal operation, 11 states					
DIVXU.W Rs,ERd	R:W NEXT	Internal operation, 19 states					
EEPMOV.B	R:W 2nd	R:B EAs*1	R:B EAd*1	R:B EAs*2	W:B EAd*2	R:W NEXT	
EEPMOV.W	R:W 2nd	R:B EAs*1	R:B EAd*1	R:B EAs*2	W:B EAd*2	R:W NEXT	
EXTS.W Rd	R:W NEXT			← Repeated n times*2 →			
EXTS.L ERd	R:W NEXT						
EXTU.W Rd	R:W NEXT						

Instruction		1	2	3	4	5	6	7
INC.W #1/2,Rd		R:W NEXT						
INC.L #1/2,ERd		R:W NEXT						
JMP @ERn		R:W NEXT	R:W EA					
JMP @aa:24		R:W 2nd	Internal operation, 1 state	R:W EA				
JMP @@aa:8	Normal	R:W NEXT	R:W aa:8	Internal operation, 1 state	R:W EA			
	Advanced	R:W NEXT	R:W:M aa:8	R:W aa:8	Internal operation, 1 state	R:W EA		
JSR @ERn	Normal	R:W NEXT	R:W EA	W:W stack				
	Advanced	R:W NEXT	R:W EA	W:W:M stack (H)	W:W stack (L)			
JSR @aa:24	Normal	R:W 2nd	Internal operation, 1 state	R:W EA	W:W stack			
	Advanced	R:W 2nd	Internal operation, 1 state	R:W EA	W:W:M stack (H)	W:W stack (L)		
JSR @@aa:8	Normal	R:W NEXT	R:W aa:8	W:W stack	R:W EA			
	Advanced	R:W NEXT	R:W:M aa:8	R:W aa:8	W:W:M stack (H)	W:W stack (L)	R:W EA	
LDC #xx:8,CCR		R:W NEXT						
LDC #xx:8,EXR		R:W 2nd	R:W NEXT					
LDC Rs,CCR		R:W NEXT						
LDC Rs,EXR		R:W NEXT						
LDC @ERs,CCR		R:W 2nd	R:W NEXT	R:W EA				
LDC @ERs,EXR		R:W 2nd	R:W NEXT	R:W EA				
LDC @(d:16,ERs),CCR		R:W 2nd	R:W 3rd	R:W NEXT	R:W EA			
LDC @(d:16,ERs),EXR		R:W 2nd	R:W 3rd	R:W NEXT	R:W EA			
LDC @(d:32,ERs),CCR		R:W 2nd	R:W 3rd	R:W 4th	R:W 5th	R:W NEXT	R:W EA	
LDC @(d:32,ERs),EXR		R:W 2nd	R:W 3rd	R:W 4th	R:W 5th	R:W NEXT	R:W EA	
LDC @ERs+,CCR		R:W 2nd	R:W NEXT	Internal operation, 1 state	R:W EA			
LDC @ERs+,EXR		R:W 2nd	R:W NEXT	Internal operation, 1 state	R:W EA			
LDC @aa:16,CCR		R:W 2nd	R:W 3rd	R:W NEXT	R:W EA			
LDC @aa:16,EXR		R:W 2nd	R:W 3rd	R:W NEXT	R:W EA			
LDC @aa:32,CCR		R:W 2nd	R:W 3rd	R:W 4th	R:W NEXT	R:W EA		
LDC @aa:32,EXR		R:W 2nd	R:W 3rd	R:W 4th	R:W NEXT	R:W EA		

RENESAS

Instruction	1	2	3	4	5	6	7
LDM.L @SP+,(ERn–ERn+2)	R:W 2nd	R:W NEXT	Internal operation, 1 state	R:W:M stack (H)*3	R:W stack (L)*3		
LDM.L @SP+,(ERn–ERn+3)	R:W 2nd	R:W NEXT	Internal operation, 1 state	R:W:M stack (H)*3	R:W stack (L)*3		
LDMAC ERs,MACH	R:W NEXT	Internal operation, 1 state	← Repeated n times*3 →				
LDMAC ERs,MACL	R:W NEXT	Internal operation, 1 state					
MAC @ERn+,@ERm+	R:W 2nd	R:W NEXT	R:W EAn	R:W EAm			
MOV.B #xx:8,Rd	R:W NEXT						
MOV.B Rs,Rd	R:W NEXT						
MOV.B @ERs,Rd	R:W NEXT	R:B EA					
MOV.B @(d:16,ERs),Rd	R:W 2nd	R:W NEXT	R:B EA				
MOV.B @(d:32,ERs),Rd	R:W 2nd	R:W 3rd	R:W 4th	R:W NEXT	R:B EA		
MOV.B @ERs+,Rd	R:W NEXT	Internal operation, 1 state	R:B EA				
MOV.B @aa:8,Rd	R:W NEXT	R:B EA					
MOV.B @aa:16,Rd	R:W 2nd	R:W NEXT	R:B EA				
MOV.B @aa:32,Rd	R:W 2nd	R:W 3rd	R:W NEXT	R:B EA			
MOV.B Rs,@ERd	R:W NEXT	W:B EA					
MOV.B Rs,@(d:16,ERd)	R:W 2nd	R:W NEXT	W:B EA				
MOV.B Rs,@(d:32,ERd)	R:W 2nd	R:W 3rd	R:W 4th	R:W NEXT	W:B EA		
MOV.B Rs,@–ERd	R:W NEXT	Internal operation, 1 state	W:B EA				
MOV.B Rs,@aa:8	R:W NEXT	W:B EA					
MOV.B Rs,@aa:16	R:W 2nd	R:W NEXT	W:B EA				
MOV.B Rs,@aa:32	R:W 2nd	R:W 3rd	R:W NEXT	W:B EA			
MOV.W #xx:16,Rd	R:W 2nd	R:W NEXT					
MOV.W Rs,Rd	R:W NEXT						
MOV.W @ERs,Rd	R:W NEXT	R:W EA					
MOV.W @(d:16,ERs),Rd	R:W 2nd	R:W NEXT	R:W EA				
MOV.W @(d:32,ERs),Rd	R:W 2nd	R:W 3rd	R:W 4th	R:W NEXT	R:W EA		
MOV.W @ERs+, Rd	R:W NEXT	Internal operation, 1 state	R:W EA				
MOV.W @aa:16,Rd	R:W 2nd	R:W NEXT	R:W EA				

Instruction	1	2	3	4	5	6	7
MOV.W Rs,@(d:16,ERd)	R:W 2nd	R:W NEXT	W:W EA				
MOV.W Rs,@(d:32,ERd)	R:W 2nd	R:W 3rd	R:E 4th	R:W NEXT	W:W EA		
MOV.W Rs,@–ERd	R:W NEXT	Internal operation, 1 state	W:W EA				
MOV.W Rs,@aa:16	R:W 2nd	R:W NEXT	W:W EA				
MOV.W Rs,@aa:32	R:W 2nd	R:W 3rd	R:W NEXT	W:W EA			
MOV.L #xx:32,ERd	R:W 2nd	R:W 3rd	R:W NEXT				
MOV.L ERs,ERd	R:W NEXT						
MOV.L @ERs,ERd	R:W 2nd	R:W:M NEXT	R:W:M EA	R:W EA+2			
MOV.L @(d:16,ERs),ERd	R:W 2nd	R:W:M 3rd	R:W NEXT	R:W:M EA	R:W EA+2		
MOV.L @(d:32,ERs),ERd	R:W 2nd	R:W:M 3rd	R:W:M 4th	R:W 5th	R:W NEXT	R:W:M EA	R:W EA+2
MOV.L @ERs+,ERd	R:W 2nd	R:W:M NEXT	Internal operation, 1 state	R:W:M EA	R:W EA+2		
MOV.L @aa:16,ERd	R:W 2nd	R:W:M 3rd	R:W NEXT	R:W:M EA	R:W EA+2		
MOV.L @aa:32,ERd	R:W 2nd	R:W:M 3rd	R:W 4th	R:W NEXT	R:W:M EA	R:W EA+2	
MOV.L ERs,@ERd	R:W 2nd	R:W:M NEXT	W:W:M EA	W:W EA+2			
MOV.L ERs,@(d:16,ERd)	R:W 2nd	R:W:M 3rd	R:W NEXT	W:W:M EA	W:W EA+2		
MOV.L ERs,@(d:32,ERd)	R:W 2nd	R:W:M 3rd	R:W:M 4th	R:W 5th	R:W NEXT	W:W:M EA	W:W EA+2
MOV.L ERs,@–ERd	R:W 2nd	R:W:M NEXT	Internal operation, 1 state	W:W:M EA	W:W EA+2		
MOV.L ERs,@aa:16	R:W 2nd	R:W:M 3rd	R:W NEXT	W:W:M EA	W:W EA+2		
MOV.L ERs,@aa:32	R:W 2nd	R:W:M 3rd	R:W 4th	R:W NEXT	W:W:M EA	W:W EA+2	
MOVFPE @aa:16,Rd*4	R:W 2nd	R:W NEXT	R:B EA				
MOVTPE Rs,@aa:16*4	R:W 2nd	R:W NEXT	W:B EA				
MULXS.B Rs,Rd	R:W 2nd	R:W NEXT	Internal operation, 2 states				
MULXS.W Rs,ERd	R:W 2nd	R:W NEXT	Internal operation, 3 states				
MULXU.B Rs,Rd	R:W NEXT	Internal operation, 2 states					
MULXU.W Rs,ERd	R:W NEXT	Internal operation, 3 states					
NEG.B Rd	R:W NEXT						
NEG.W Rd	R:W NEXT						
NEG.L ERd	R:W NEXT						
NOP	R:W NEXT						
NOT.B Rd	R:W NEXT						
NOT.W Rd	R:W NEXT						
NOT.L ERd	R:W NEXT						

RENESAS

Instruction	1	2	3	4	5	6	7
OR.W #xx:16,Rd	R:W 2nd	R:W NEXT					
OR.W Rs,Rd	R:W NEXT						
OR.L #xx:32,ERd	R:W 2nd	R:W 3rd	R:W NEXT				
OR.L ERs,ERd	R:W 2nd	R:W NEXT					
ORC #xx:8,CCR	R:W NEXT						
ORC #xx:8,EXR	R:W 2nd	R:W NEXT					
POP.W Rn	R:W NEXT	Internal operation, 1 state	R:W EA				
POP.L ERn	R:W 2nd	R:W:M NEXT	Internal operation, 1 state	R:W:M EA	R:W EA+2		
PUSH.W Rn	R:W NEXT	Internal operation, 1 state	W:W EA				
PUSH.L ERn	R:W 2nd	R:W:M NEXT	Internal operation, 1 state	W:W:M EA	W:W EA+2		
ROTL.B Rd	R:W NEXT						
ROTL.B #2,Rd	R:W NEXT						
ROTL.W Rd	R:W NEXT						
ROTL.W #2,Rd	R:W NEXT						
ROTL.L ERd	R:W NEXT						
ROTL.L #2,ERd	R:W NEXT						
ROTR.B Rd	R:W NEXT						
ROTR.B #2,Rd	R:W NEXT						
ROTR.W Rd	R:W NEXT						
ROTR.W #2,Rd	R:W NEXT						
ROTR.L ERd	R:W NEXT						
ROTR.L #2,ERd	R:W NEXT						
ROTXL.B Rd	R:W NEXT						
ROTXL.B #2,Rd	R:W NEXT						
ROTXL.W Rd	R:W NEXT						
ROTXL.W #2,Rd	R:W NEXT						
ROTXL.L ERd	R:W NEXT						
ROTXL.L #2,ERd	R:W NEXT						
ROTXR.B Rd	R:W NEXT						
ROTXR.B #2,Rd	R:W NEXT						
ROTXR.W Rd	R:W NEXT						

Instruction		1	2	3	4	5	6	7
ROTXR.L #2,ERd		R:W NEXT						
RTE		R:W NEXT	R:W stack (EXR)	R:W stack (H)	R:W stack (L)	Internal operation, 1 state	R:W*5	
RTS	Normal	R:W NEXT	R:W stack	Internal operation, 1 state	R:W*5			
	Advanced	R:W NEXT	R:W:M stack (H)	R:W stack (L)	Internal operation, 1 state	R:W*5		
SHAL.B Rd		R:W NEXT						
SHAL.B #2,Rd		R:W NEXT						
SHAL.W Rd		R:W NEXT						
SHAL.W #2,Rd		R:W NEXT						
SHAL.L ERd		R:W NEXT						
SHAL.L #2,ERd		R:W NEXT						
SHAR.B Rd		R:W NEXT						
SHAR.B #2,Rd		R:W NEXT						
SHAR.W Rd		R:W NEXT						
SHAR.W #2,Rd		R:W NEXT						
SHAR.L ERd		R:W NEXT						
SHAR.L #2,ERd		R:W NEXT						
SHLL.B Rd		R:W NEXT						
SHLL.B #2,Rd		R:W NEXT						
SHLL.W Rd		R:W NEXT						
SHLL.W #2,Rd		R:W NEXT						
SHLL.L ERd		R:W NEXT						
SHLL.L #2,ERd		R:W NEXT						
SHLR.B Rd		R:W NEXT						
SHLR.B #2,Rd		R:W NEXT						
SHLR.W Rd		R:W NEXT						
SHLR.W #2,Rd		R:W NEXT						
SHLR.L ERd		R:W NEXT						
SHLR.L #2,ERd		R:W NEXT						
SLEEP		R:W NEXT	Internal operation:M					
STC CCR,Rd		R:W NEXT						
STC EXR,Rd		R:W NEXT						
STC CCR,@ERd		R:W 2nd	R:W NEXT	W:W EA				
STC EXR,@ERd		R:W 2nd	R:W NEXT	W:W EA				

Instruction		1	2	3	4	5	6	7
STC EXR,@(d:16,ERd)		R:W 2nd	R:W 3rd	R:W NEXT	W:W EA			
STC CCR,@(d:32,ERd)		R:W 2nd	R:W 3rd	R:W 4th	R:W 5th	R:W NEXT	W:W EA	
STC EXR,@(d:32,ERd)		R:W 2nd	R:W 3rd	R:W 4th	R:W 5th	R:W NEXT	W:W EA	
STC CCR,@–ERd		R:W 2nd	R:W NEXT	Internal operation, 1 state	W:W EA			
STC EXR,@–ERd		R:W 2nd	R:W NEXT	Internal operation, 1 state	W:W EA			
STC CCR,@aa:16		R:W 2nd	R:W 3rd	R:W NEXT	W:W EA			
STC EXR,@aa:16		R:W 2nd	R:W 3rd	R:W NEXT	W:W EA			
STC CCR,@aa:32		R:W 2nd	R:W 3rd	R:W 4th	R:W NEXT	W:W EA		
STC EXR,@aa:32		R:W 2nd	R:W 3rd	R:W 4th	R:W NEXT	W:W EA		
STM.L(ERn–ERn+1),@–SP		R:W 2nd	R:W:M NEXT	Internal operation, 1 state	W:W:M stack (H)*3	W:W stack (L)*3		
STM.L(ERn–ERn+2),@–SP		R:W 2nd	R:W:M NEXT	Internal operation, 1 state	W:W:M stack (H)*3	W:W stack (L)*3		
STM.L(ERn–ERn+3),@–SP		R:W 2nd	R:W:M NEXT	Internal operation, 1 state	W:W:M stack (H)*3	W:W stack (L)*3		
STMAC MACH,ERd		R:W NEXT		←Repeated n times*3→				
STMAC MACL,ERd		R:W NEXT						
SUB.B Rs,Rd		R:W NEXT						
SUB.W #xx:16,Rd		R:W 2nd	R:W NEXT					
SUB.W Rs,Rd		R:W NEXT						
SUB.L #xx:32,ERd		R:W 2nd	R:W 3rd	R:W NEXT				
SUB.L ERs,ERd		R:W NEXT						
SUBS #1/2/4,ERd		R:W NEXT						
SUBX #xx:8,Rd		R:W NEXT						
SUBX Rs,Rd		R:W NEXT						
TAS @ERd		R:W 2nd	R:W NEXT	R:B:M EA	W:B EA			
TRAPA #x:2	Normal	R:W NEXT	Internal operation, 1 state	W:W stack (L)	W:W stack (H)	W:W stack (EXR)	R:W VEC	Internal operation, 1 state
	Advanced	R:W NEXT	Internal operation, 1 state	W:W stack (L)	W:W stack (H)	W:W stack (EXR)	R:W:M VEC	R:W VEC+2
XOR.B #xx:8,Rd		R:W NEXT						
XOR.B Rs,Rd		R:W NEXT						
XOR.W #xx:16,Rd		R:W 2nd	R:W NEXT					

Instruction		1	2	3	4	5	6	7
XOR.L ERs,ERd		R:W 2nd	R:W NEXT					
XORC #xx:8,CCR		R:W NEXT						
XORC #xx:8,EXR		R:W 2nd	R:W NEXT					
Reset exception handling	Normal	R:W VEC	Internal operation, 1 state	R:W*6				
	Advanced	R:W VEC	R:W VEC+2	Internal operation, 1 state	R:W*6			
Interrupt exception handling	Normal	R:W*7	Internal operation, 1 state	W:W stack (L)	W:W stack (H)	W:W stack (EXR)	R:W VEC	Internal opera 1 state
	Advanced	R:W*7	Internal operation, 1 state	W:W stack (L)	W:W stack (H)	W:W stack (EXR)	R:W:M VEC	R:W VEC-

Notes: 1. EAs is the contents of ER5. EAd is the contents of ER6.
2. EAs is the contents of ER5. EAd is the contents of ER6. Both registers are incremented by 1 after execution o value of R4L or R4. If n = 0, these bus cycles are not executed.
3. Repeated two times to save or restore two registers, three times for three registers, or four times for four regis
4. Can not be used in the H8S/2655 Group.
5. Start address after return.
6. Start address of the program.
7. Prefetch address, equal to two plus the PC value pushed onto the stack. In recovery from sleep mode or softw operation is replaced by an internal operation.
8. Start address of the interrupt-handling routine.

⌊	7 for byte operands
Si	The i-th bit of the source operand
Di	The i-th bit of the destination operand
Ri	The i-th bit of the result
Dn	The specified bit in the destination operand
—	Not affected
↕	Modified according to the result of the instruction (see definition)
0	Always cleared to 0
1	Always set to 1
*	Undetermined (no guaranteed value)
Z'	Z flag before instruction execution
C'	C flag before instruction execution

ADDS	—	—	—	—	—	
ADDX	↕	↕	↕	↕	↕	H = Sm–4 · Dm–4 + Dm–4 · $\overline{Rm–4}$ + Sm–4 · $\overline{Rn}$
						N = Rm
						Z = Z' · $\overline{Rm}$ · ..... · $\overline{R0}$
						V = Sm · Dm · $\overline{Rm}$ + $\overline{Sm}$ · $\overline{Dm}$ · Rm
						C = Sm · Dm + Dm · $\overline{Rm}$ + Sm · $\overline{Rm}$
AND	—	↕	↕	0	—	N = Rm
						Z = $\overline{Rm}$ · $\overline{Rm–1}$ · ..... · $\overline{R0}$
ANDC	↕	↕	↕	↕	↕	Stores the corresponding bits of the result.
						No flags change when the operand is EXR.
BAND	—	—	—	—	↕	C = C' · Dn
Bcc	—	—	—	—	—	
BCLR	—	—	—	—	—	
BIAND	—	—	—	—	↕	C = C' · $\overline{Dn}$
BILD	—	—	—	—	↕	C = $\overline{Dn}$
BIOR	—	—	—	—	↕	C = C' + $\overline{Dn}$
BIST	—	—	—	—	—	
BIXOR	—	—	—	—	↕	C = C' · Dn + $\overline{C'}$ · $\overline{Dn}$
BLD	—	—	—	—	↕	C = Dn
BNOT	—	—	—	—	—	
BOR	—	—	—	—	↕	C = C' + Dn
BSET	—	—	—	—	—	
BSR	—	—	—	—	—	
BST	—	—	—	—	—	
BTST	—	—	↕	—	—	Z = $\overline{Dn}$
BXOR	—	—	—	—	↕	C = C' · $\overline{Dn}$ + $\overline{C'}$ · Dn

DAA	*	↕	↕	*	↕	N = Rm Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$ C: decimal arithmetic carry
DAS	*	↕	↕	*	↕	N = Rm Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$ C: decimal arithmetic borrow
DEC	—	↕	↕	↕	—	N = Rm Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$ V = Dm · $\overline{Rm}$
DIVXS	—	↕	↕	—	—	N = Sm · $\overline{Dm}$ + $\overline{Sm}$ · Dm Z = $\overline{Sm} \cdot \overline{Sm-1} \cdot \cdots\cdots \cdot \overline{S0}$
DIVXU	—	↕	↕	—	—	N = Sm Z = $\overline{Sm} \cdot \overline{Sm-1} \cdot \cdots\cdots \cdot \overline{S0}$
EEPMOV	—	—	—	—	—	
EXTS	—	↕	↕	0	—	N = Rm Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$
EXTU	—	0	↕	0	—	Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$
INC	—	↕	↕	↕	—	N = Rm Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$ V = $\overline{Dm}$ · Rm
JMP	—	—	—	—	—	
JSR	—	—	—	—	—	
LDC	↕	↕	↕	↕	↕	Stores the corresponding bits of the result. No flags change when the operand is EXR.
LDM	—	—	—	—	—	
LDMAC	—	—	—	—	—	

						Z = R2m · R2m–1 · ····· · R0
MULXU	—	—	—	—	—	
NEG	↕	↕	↕	↕	↕	H = Dm–4 + Rm–4 N = Rm Z = $\overline{Rm}$ · $\overline{Rm-1}$ · ····· · $\overline{R0}$ V = Dm · Rm C = Dm + Rm
NOP	—	—	—	—	—	
NOT	—	↕	↕	0	—	N = Rm Z = $\overline{Rm}$ · $\overline{Rm-1}$ · ····· · $\overline{R0}$
OR	—	↕	↕	0	—	N = Rm Z = $\overline{Rm}$ · $\overline{Rm-1}$ · ····· · $\overline{R0}$
ORC	↕	↕	↕	↕	↕	Stores the corresponding bits of the result. No flags change when the operand is EXR.
POP	—	↕	↕	0	—	N = Rm Z = $\overline{Rm}$ · $\overline{Rm-1}$ · ····· · $\overline{R0}$
PUSH	—	↕	↕	0	—	N = Rm Z = $\overline{Rm}$ · $\overline{Rm-1}$ · ····· · $\overline{R0}$
ROTL	—	↕	↕	0	↕	N = Rm Z = $\overline{Rm}$ · $\overline{Rm-1}$ · ····· · $\overline{R0}$ C = Dm (1-bit shift) or C = Dm–1 (2-bit shift)
ROTR	—	↕	↕	0	↕	N = Rm Z = $\overline{Rm}$ · $\overline{Rm-1}$ · ····· · $\overline{R0}$ C = D0 (1-bit shift) or C = D1 (2-bit shift)

RTE	↕	↕	↕	↕	↕	Stores the corresponding bits of the result.
RTS	—	—	—	—	—	
SHAL	—	↕	↕	↕	↕	N = Rm Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$ V = $\overline{Dm \cdot Dm-1 + \overline{Dm} \cdot \overline{Dm-1}}$ (1-bit shift) V = $\overline{Dm \cdot Dm-1 \cdot Dm-2 \cdot \overline{Dm} \cdot \overline{Dm-1} \cdot \overline{Dm-2}}$ (2-b C = Dm (1-bit shift) or C = Dm–1 (2-bit shift)
SHAR	—	↕	↕	0	↕	N = Rm Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$ C = D0 (1-bit shift) or C = D1 (2-bit shift)
SHLL	—	↕	↕	0	↕	N = Rm Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$ C = Dm (1-bit shift) or C = Dm–1 (2-bit shift)
SHLR	—	0	↕	0	↕	N = Rm Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$ C = D0 (1-bit shift) or C = D1 (2-bit shift)
SLEEP	—	—	—	—	—	
STC	—	—	—	—	—	
STM	—	—	—	—	—	
STMAC	—	↕	↕	↕	—	N = 1 if MAC instruction resulted in negative valu register Z = 1 if MAC instruction resulted in zero value in register V = 1 if MAC instruction resulted in overflow

SUBX	↕	↕	↕	↕	↕	H = $\overline{Sm–4}$ · Dm–4 + Dm–4 · $\overline{Rm–4}$ + $\overline{Sm–4}$ · Rm
						N = Rm
						Z = Z' · $\overline{Rm}$ · ····· · $\overline{R0}$
						V = $\overline{Sm}$ · Dm · $\overline{Rm}$ + Sm · $\overline{Dm}$ · Rm
						C = Sm · $\overline{Dm}$ + $\overline{Dm}$ · Rm + Sm · Rm
TAS	—	↕	↕	0	—	N = Dm
						Z = $\overline{Dm}$ · $\overline{Dm–1}$ · ····· · $\overline{D0}$
TRAPA	—	—	—	—	—	
XOR	—	↕	↕	0	—	N = Rm
						Z = $\overline{Rm}$ · $\overline{Rm–1}$ · ····· · $\overline{R0}$
XORC	↕	↕	↕	↕	↕	Stores the corresponding bits of the result.
						No flags change when the operand is EXR.

to H'FBFF	SAR									
	MRB	CHNE	DISEL	—	—	—	—	—	—	
	DAR									
	CRA									
	CRB									
H'FE80	TCR3	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU
H'FE81	TMDR3	—	—	BFB	BFA	MD3	MD2	MD1	MD0	
H'FE82	TIOR3H	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
H'FE83	TIOR3L	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0	
H'FE84	TIER3	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA	
H'FE85	TSR3	—	—	—	TCFV	TGFD	TGFC	TGFB	TGFA	
H'FE86	TCNT3									
H'FE87										
H'FE88	TGR3A									
H'FE89										
H'FE8A	TGR3B									
H'FE8B										
H'FE8C	TGR3C									
H'FE8D										
H'FE8E	TGR3D									
H'FE8F										

H'FE97										
H'FE98	TGR4A									
H'FE99										
H'FE9A	TGR4B									
H'FE9B										
H'FEA0	TCR5	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TP
H'FEA1	TMDR5	—	—	—	—	MD3	MD2	MD1	MD0	
H'FEA2	TIOR5	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
H'FEA4	TIER5	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA	
H'FEA5	TSR5	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA	
H'FEA6	TCNT5									
H'FEA7										
H'FEA8	TGR5A									
H'FEA9										
H'FEAA	TGR5B									
H'FEAB										
H'FEB0	P1DDR	P17DDR	P16DDR	P15DDR	P14DDR	P13DDR	P12DDR	P11DDR	P10DDR	Po
H'FEB1	P2DDR	P27DDR	P26DDR	P25DDR	P24DDR	P23DDR	P22DDR	P21DDR	P20DDR	
H'FEB2	P3DDR	—	—	P35DDR	P34DDR	P33DDR	P32DDR	P31DDR	P30DDR	
H'FEB4	P5DDR	—	—	—	—	P53DDR	P52DDR	P51DDR	P50DDR	
H'FEB5	P6DDR	P67DDR	P66DDR	P65DDR	P64DDR	P63DDR	P62DDR	P61DDR	P60DDR	
H'FEB9	PADDR	PA7DDR	PA6DDR	PA5DDR	PA4DDR	PA3DDR	PA2DDR	PA1DDR	PA0DDR	
H'FEBA	PBDDR	PB7DDR	PB6DDR	PB5DDR	PB4DDR	PB3DDR	PB2DDR	PB1DDR	PB0DDR	
H'FEBB	PCDDR	PC7DDR	PC6DDR	PC5DDR	PC4DDR	PC3DDR	PC2DDR	PC1DDR	PC0DDR	
H'FEBC	PDDDR	PD7DDR	PD6DDR	PD5DDR	PD4DDR	PD3DDR	PD2DDR	PD1DDR	PD0DDR	
H'FEBD	PEDDR	PE7DDR	PE6DDR	PE5DDR	PE4DDR	PE3DDR	PE2DDR	PE1DDR	PE0DDR	
H'FEBE	PFDDR	PF7DDR	PF6DDR	PF5DDR	PF4DDR	PF3DDR	PF2DDR	PF1DDR	PF0DDR	
H'FEBF	PGDDR	—	—	—	PG4DDR	PG3DDR	PG2DDR	PG1DDR	PG0DDR	

H'FEC7	IPRD	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0	
H'FEC8	IPRE	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0	
H'FEC9	IPRF	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0	
H'FECA	IPRG	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0	
H'FECB	IPRH	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0	
H'FECC	IPRI	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0	
H'FECD	IPRJ	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0	
H'FECE	IPRK	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0	
H'FED0	ABWCR	ABW7	ABW6	ABW5	ABW4	ABW3	ABW2	ABW1	ABW0	Bus cont
H'FED1	ASTCR	AST7	AST6	AST5	AST4	AST3	AST2	AST1	AST0	
H'FED2	WCRH	W71	W70	W61	W60	W51	W50	W41	W40	
H'FED3	WCRL	W31	W30	W21	W20	W11	W10	W01	W00	
H'FED4	BCRH	ICIS1	ICIS0	BRSTRM	BRSTS1	BRSTS0	RMTS2	RMTS1	RMST0	
H'FED5	BCRL	BLE	BREQOE	EAE	—	—	ASS	WDBE	WAITE	
H'FED6	MCR	TPC	BE	RCDM	CW2	MXC1	MXC0	RLW1	RLW0	
H'FED7	DRAMCR	RFSHE	RCW	RMODE	CMF	CMIE	CKS2	CKS1	CKS0	
H'FED8	RTCNT									
H'FED9	RTCOR									
H'FEE0	MAR0AH	—	—	—	—	—	—	—	—	DMA
H'FEE1										
H'FEE2	MAR0AL									
H'FEE3										
H'FEE4	IOAR0A									
H'FEE5										
H'FEE6	ETCR0A									
H'FEE7										
H'FEE8	MAR0BH	—	—	—	—	—	—	—	—	
H'FEE9										

H'FEF0	MAR1AH	—	—	—	—	—	—	—	—	
H'FEF1										
H'FEF2	MAR1AL									
H'FEF3										
H'FEF4	IOAR1A									
H'FEF5										
H'FEF6	ETCR1A									
H'FEF7										
H'FEF8	MAR1BH	—	—	—	—	—	—	—	—	
H'FEF9										
H'FEFA	MAR1BL									
H'FEFB										
H'FEFC	IOAR1B									
H'FEFD										
H'FEFE	ETCR1B									
H'FEFF										
H'FF00	DMAWER	—	—	—	—	WE1B	WE1A	WE0B	WE0A	
H'FF01	DMATCR	—	—	TEE1	TEE0	—	—	—	—	
H'FF02	DMACR0A	DTSZ	DTID	RPE	DTDIR	DTF3	DTF2	DTF1	DTF0	Sh ad mo
		DTSZ	SAID	SAIDE	BLKDIR	BLKE	—	—	—	Ful ad mo
H'FF03	DMACR0B	DTSZ	DTID	RPE	DTDIR	DTF3	DTF2	DTF1	DTF0	Sh ad mo
		—	DAID	DAIDE	—	DTF3	DTF2	DTF1	DTF0	Ful ad mo

										addr mod
		—	DAID	DAIDE	—	DTF3	DTF2	DTF1	DTF0	Full addr mod
H'FF06	DMABCRH	FAE1	FAE0	SAE1	SAE0	DTA1B	DTA1A	DTA0B	DTA0A	Shor addr mod
		FAE1	FAE0	—	—	DTA1	—	DTA0	—	Full addr mod
H'FF07	DMABCRL	DTE1B	DTE1A	DTE0B	DTE0A	DTIE1B	DTIE1A	DTIE0B	DTIE0A	Shor addr mod
		DTME1	DTE1	DTME0	DTE0	DTIE1B	DTIE1A	DTIE0B	DTIE0A	Full addr mod
H'FF2C	ISCRH	IRQ7SCB	IRQ7SCA	IRQ6SCB	IRQ6SCA	IRQ5SCB	IRQ5SCA	IRQ4SCB	IRQ4SCA	Inter cont
H'FF2D	ISCRL	IRQ3SCB	IRQ3SCA	IRQ2SCB	IRQ2SCA	IRQ1SCB	IRQ1SCA	IRQ0SCB	IRQ0SCA	
H'FF2E	IER	IRQ7E	IRQ6E	IRQ5E	IRQ4E	IRQ3E	IRQ2E	IRQ1E	IRQ0E	
H'FF2F	ISR	IRQ7F	IRQ6F	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	IRQ0F	
H'FF30 to H'FF35	DTCER	DTCE7	DTCE6	DTCE5	DTCE4	DTCE3	DTCE2	DTCE1	DTCE0	DTC
H'FF37	DTVECR	SWDTE	DTVEC6	DTVEC5	DTVEC4	DTVEC3	DTVEC2	DTVEC1	DTVEC0	
H'FF38	SBYCR	SSBY	STS2	STS1	STS0	OPE	—	—	—	MCU
H'FF39	SYSCR	MACS	—	INTM1	INTM0	NMIEG	—	—	RAME	
H'FF3A	SCKCR	PSTOP	—	—	—	—	SCK2	SCK1	SCK0	
H'FF3B	MDCR	—	—	—	—	—	MDS2	MDS1	MDS0	
H'FF3C	MSTPCRH	MSTP15	MSTP14	MSTP13	MSTP12	MSTP11	MSTP10	MSTP9	MSTP8	
H'FF3D	MSTPCRL	MSTP7	MSTP6	MSTP5	MSTP4	MSTP3	MSTP2	MSTP1	MSTP0	

H'FF4C*2	NDRH	NDR15	NDR14	NDR13	NDR12	NDR11	NDR10	NDR9	NDR8	
H'FF4D*2	NDRL	NDR7	NDR6	NDR5	NDR4	NDR3	NDR2	NDR1	NDR0	
H'FF4E*2	NDRH	—	—	—	—	NDR11	NDR10	NDR9	NDR8	
H'FF4F*2	NDRL	—	—	—	—	NDR3	NDR2	NDR1	NDR0	
H'FF50	PORT1	P17	P16	P15	P14	P13	P12	P11	P10	Po
H'FF51	PORT2	P27	P26	P25	P24	P23	P22	P21	P20	
H'FF52	PORT3	—	—	P35	P34	P33	P32	P31	P30	
H'FF53	PORT4	P47	P46	P45	P44	P43	P42	P41	P40	
H'FF54	PORT5	—	—	—	—	P53	P52	P51	P50	
H'FF55	PORT6	P67	P66	P65	P64	P63	P62	P61	P60	
H'FF59	PORTA	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	
H'FF5A	PORTB	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	
H'FF5B	PORTC	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	
H'FF5C	PORTD	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0	
H'FF5D	PORTE	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0	
H'FF5E	PORTF	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0	
H'FF5F	PORTG	—	—	—	PG4	PG3	PG2	PG1	PG0	
H'FF60	P1DR	P17DR	P16DR	P15DR	P14DR	P13DR	P12DR	P11DR	P10DR	
H'FF61	P2DR	P27DR	P26DR	P25DR	P24DR	P23DR	P22DR	P21DR	P20DR	
H'FF62	P3DR	—	—	P35DR	P34DR	P33DR	P32DR	P31DR	P30DR	
H'FF64	P5DR	—	—	—	—	P53DR	P52DR	P51DR	P50DR	
H'FF65	P6DR	P67DR	P66DR	P65DR	P64DR	P63DR	P62DR	P61DR	P60DR	
H'FF69	PADR	PA7DR	PA6DR	PA5DR	PA4DR	PA3DR	PA2DR	PA1DR	PA0DR	
H'FF6A	PBDR	PB7DR	PB6DR	PB5DR	PB4DR	PB3DR	PB2DR	PB1DR	PB0DR	
H'FF6B	PCDR	PC7DR	PC6DR	PC5DR	PC4DR	PC3DR	PC2DR	PC1DR	PC0DR	
H'FF6C	PDDR	PD7DR	PD6DR	PD5DR	PD4DR	PD3DR	PD2DR	PD1DR	PD0DR	
H'FF6D	PEDR	PE7DR	PE6DR	PE5DR	PE4DR	PE3DR	PE2DR	PE1DR	PE0DR	
H'FF6E	PFDR	PF7DR	PF6DR	PF5DR	PF4DR	PF3DR	PF2DR	PF1DR	PF0DR	
H'FF6F	PGDR	—	—	—	PG4DR	PG3DR	PG2DR	PG1DR	PG0DR	

H'FF77	PAODR	PA7ODR	PA6ODR	PA5ODR	PA4ODR	PA3ODR	PA2ODR	PA1ODR	PA0ODR	
H'FF78	SMR0	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0	SCI0 Sma inter
H'FF79	BRR0									
H'FF7A	SCR0	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
H'FF7B	TDR0									
H'FF7C	SSR0	TDRE	RDRF	ORER	FER/ ERS	PER	TEND	MPB	MPBT	
H'FF7D	RDR0									
H'FF7E	SCMR0	—	—	—	—	SDIR	SINV	—	SMIF	
H'FF80	SMR1	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0	SCI1 Sma inter
H'FF81	BRR1									
H'FF82	SCR1	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
H'FF83	TDR1									
H'FF84	SSR1	TDRE	RDRF	ORER	FER/ ERS	PER	TEND	MPB	MPBT	
H'FF85	RDR1									
H'FF86	SCMR1	—	—	—	—	SDIR	SINV	—	SMIF	
H'FF88	SMR2	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0	SCI2 Sma inter
H'FF89	BRR2									
H'FF8A	SCR2	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
H'FF8B	TDR2									
H'FF8C	SSR2	TDRE	RDRF	ORER	FER/ ERS	PER	TEND	MPB	MPBT	
H'FF8D	RDR2									
H'FF8E	SCMR2	—	—	—	—	SDIR	SINV	—	SMIF	

H'FF96	ADDRDH	—	—	—	—	—	—	AD9	AD8	
H'FF97	ADDRDL	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	
H'FF98	ADDREH	—	—	—	—	—	—	AD9	AD8	
H'FF99	ADDREL	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	
H'FF9A	ADDRFH	—	—	—	—	—	—	AD9	AD8	
H'FF9B	ADDRFL	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	
H'FF9C	ADDRGH	—	—	—	—	—	—	AD9	AD8	
H'FF9D	ADDRGL	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	
H'FF9E	ADDRHH	—	—	—	—	—	—	AD9	AD8	
H'FF9F	ADDRHL	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	
H'FFA0	ADCSR	ADF	ADIE	ADST	CKS	GRP	CH2	CH1	CH0	
H'FFA1	ADCR	—	PWR	TRGS1	TRGS0	SCAN	DSMP	BUFE1	BUFE0	
H'FFA4	DADR0									D/A
H'FFA5	DADR1									co
H'FFA6	DACR	DAOE1	DAOE0	DAE	—	—	—	—	—	
H'FFB0	TCR0	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	8-b
H'FFB1	TCR1	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	cha
H'FFB2	TCSR0	CMFB	CMFA	OVF	ADTE	OS3	OS2	OS1	OS0	0,
H'FFB3	TCSR1	CMFB	CMFA	OVF	—	OS3	OS2	OS1	OS0	
H'FFB4	TCORA0									
H'FFB5	TCORA1									
H'FFB6	TCORB0									
H'FFB7	TCORB1									
H'FFB8	TCNT0									
H'FFB9	TCNT1									

H'FFC1	TSYR	—	—	SYNC5	SYNC4	SYNC3	SYNC2	SYNC1	SYNC0	
H'FFD0	TCR0	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU
H'FFD1	TMDR0	—	—	BFB	BFA	MD3	MD2	MD1	MD0	
H'FFD2	TIOR0H	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
H'FFD3	TIOR0L	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0	
H'FFD4	TIER0	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA	
H'FFD5	TSR0	—	—	—	TCFV	TGFD	TGFC	TGFB	TGFA	
H'FFD6	TCNT0									
H'FFD7										
H'FFD8	TGR0A									
H'FFD9										
H'FFDA	TGR0B									
H'FFDB										
H'FFDC	TGR0C									
H'FFDD										
H'FFDE	TGR0D									
H'FFDF										
H'FFE0	TCR1	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU
H'FFE1	TMDR1	—	—	—	—	MD3	MD2	MD1	MD0	
H'FFE2	TIOR1	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
H'FFE4	TIER1	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA	
H'FFE5	TSR1	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA	
H'FFE6	TCNT1									
H'FFE7										
H'FFE8	TGR1A									
H'FFE9										
H'FFEA	TGR1B									
H'FFEB										

H'FFF7		
H'FFF8	TGR2A	
H'FFF9		
H'FFFA	TGR2B	
H'FFFB		

Notes: 1. Located in on-chip RAM. The bus width is 32 bits when the DTC accesses register information, and 16 bits otherwise.

2. If the pulse output group 2 and pulse output group 3 output triggers are the according to the PCR setting, the NDRH address will be H'FF4C, and if diff address of NDRH for group 2 will be H'FF4E, and that for group 3 will be H' Similarly, if the pulse output group 0 and pulse output group 1 output trigge same according to the PCR setting, the NDRL address will be H'FF4D, and the address of NDRL for group 0 will be H'FF4F, and that for group 1 will be

DTC D
Transfe

0	Byt tra
1	Wo tra

DTC Transfer Mode Sele

0	Destination side is area or block area
1	Source side is rep or block area

DTC Mode

0	0	Normal mode
	1	Repeat mode
1	0	Block transfer mode
	1	—

Destination Address Mode

0	—	DAR is fixed
1	0	DAR is incremented after a transfer (by +1 when Sz = 0; by +2 when Sz = 1)
	1	DAR is decremented after a transfer (by -1 when Sz = 0; by -2 when Sz = 1)

Source Address Mode

0	—	SAR is fixed
1	0	SAR is incremented after a transfer (by +1 when Sz = 0; by +2 when Sz = 1)
	1	SAR is decremented after a transfer (by -1 when Sz = 0; by -2 when Sz = 1)

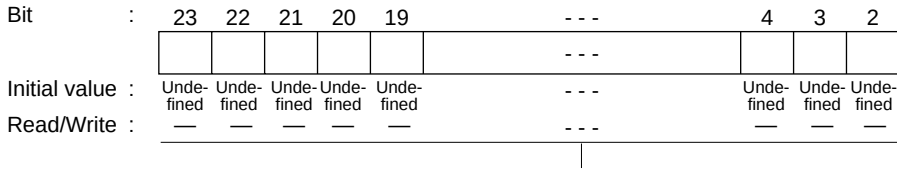
DTC Interrupt Select

0	After a data transfer ends, the CPU interrupt is disabled unless the transfer counter is 0
1	After a data transfer ends, the CPU interrupt is ena

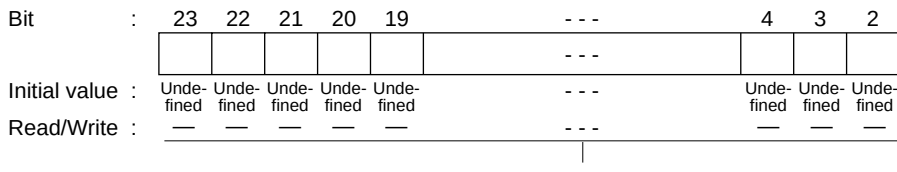
DTC Chain Transfer Enable

0	End of DTC data transfer
1	DTC chain transfer

SAR—DTC Source Address Register H'F800—H'FBFF

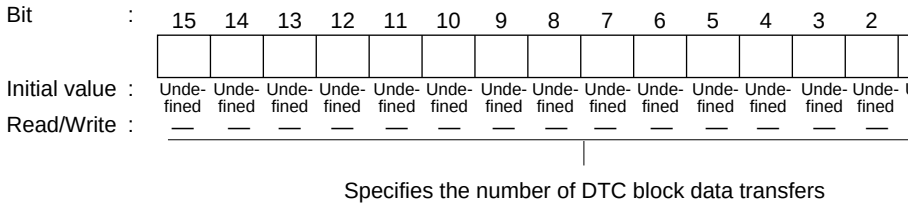


DAR—DTC Destination Address Register H'F800—H'FBFF



Specifies the number of DTC data transfers

CRB—DTC Transfer Count Register B H'F800—H'FBFF



		1	Internal clock: counts on φ/4
	1	0	Internal clock: counts on φ/1
		1	Internal clock: counts on φ/6
1	0	0	External clock: counts on T
		1	Internal clock: counts on φ/1
	1	0	Internal clock: counts on φ/2
		1	Internal clock: counts on φ/4

Clock Edge

0	0	Count at rising edge
	1	Count at falling edge
1	—	Count at both edges

Counter Clear

0	0	0	TCNT clearing disabled
		1	TCNT cleared by TGRA compare match/input capture
	1	0	TCNT cleared by TGRB compare match/input capture
		1	TCNT cleared by counter clearing for another channel performing synchronous clearing/synchronous operation*1
1	0	0	TCNT clearing disabled
		1	TCNT cleared by TGRC compare match/input capture*2
	1	0	TCNT cleared by TGRD compare match/input capture*2
		1	TCNT cleared by counter clearing for another channel performing synchronous clearing/synchronous operation*1

Notes: 1. Synchronous operation setting is performed by setting the SYNC bit in TSYR to 1.
2. When TGRC or TGRD is used as a buffer register, TCNT is not cleared because the buffer register setting has priority, and compare match/input capture does not occur.

0	0	0	0	Normal ope
			1	Reserved
		1	0	PWM mode
			1	PWM mode
	1	0	0	Phase coun
			1	Phase coun
		1	0	Phase coun
			1	Phase coun
1	*	*	*	—

Legend: *: Don't care

Notes: 1. MD3 is a reserved b
it should always be
2. Phase counting mo
set for channels 0 a
case, 0 should alwa
to MD2.

Buffer Operation A

0	TGRA operates normally
1	TGRA and TGRC used together for buffer operation

Buffer Operation B

0	TGRB operates normally
1	TGRB and TGRD used together for buffer operation

			1	TGR3B is output compare register	Initial output is 0 output	0 output at compar	
		1	0			1 output at compar	
			1			Toggle output at co	
	1	0	0		Output disabled		
			1		Initial output is 1 output	0 output at compar	
		1	0			1 output at compar	
			1			Toggle output at co	
1	0	0	0	TGR3A is input capture register	Capture input source is TIOCA3 pin	Input capture at ris	
			1			Input capture at fal	
		1	*			Input capture at bo	
	1	*	*		Capture input source is channel 4/count clock	Input capture at TC count-down	

Legend: *: Don't care

TGR3B I/O Control

0	0	0	0	TGR3B is output compare register	Output disabled	
			1		Initial output is 0 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
1	0	0	0	TGR3B is input capture register	Capture input source is TIOCB3 pin	Input capture at rising edge
			1			Input capture at falling edge
		1	*			Input capture at both edges
	1	*	*		Capture input source is channel 4/count clock	Input capture at TCNT4 count-up/ count-down

Legend: *: Don't care

Note: 1. When bits TPSC2 to TPSC0 in TCR4 are set to B'000, and ϕ/1 is used as the TCNT4 count clock, this setting will be invalid and input capture will not occur.

			1	compare register	Initial output is 0 output	0 output at compare match	
		1	0			1 output at compare match	
			1			Toggle output at compare r	
	1	0	0		Output disabled		
			1		Initial output is 1 output	0 output at compare match	
		1	0			1 output at compare match	
			1			Toggle output at compare r	
1	0	0	0	TGR3C is input capture register	Capture input source is TIOCC3 pin	Input capture at rising edge	
			1			Input capture at falling edg	
		1	*			Input capture at both edges	
	1	*	*		Capture input source is channel 4/count clock	Input capture at TCNT4 co count-down	

Legend: *: Don't care

Note: When the BFA bit in TMDR3 is set to 1 and TGR3C is used as a buf register, this setting is invalid and input capture/output compare is n generated.

TGR3D I/O Control

0	0	0	0	TGR3D is output compare register	Output disabled	
			1		Initial output is 0 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
1	0	0	0	TGR3D is input capture register*2	Capture input source is TIOCD3 pin	Input capture at rising edge
			1			Input capture at falling edge
		1	*			Input capture at both edges
	1	*	*		Capture input source is channel 4/count clock	Input capture at TCNT4 count-up/ count-down*1

Legend: *: Don't care

Notes: 1. When bits TPSC2 to TPSC0 in TCR4 are set to B'000 and φ/1 is used as the TCNT4 count clock, this setting is invalid and input capture is not generated.
2. When the BFB bit in TMDR3 is set to 1 and TGR3D is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Note: When GRC or GRD is designated for buffer operation, this setting is invalid and the register operates as a buffer register.

	by TGFA bit
1	Interrupt requ by TGFA bit

TGR Interrupt Enable B

0	Interrupt requests by TGFB bit disab
1	Interrupt requests by TGFB bit enabl

TGR Interrupt Enable C

0	Interrupt requests (TGIC) by TGFC bit disabled
1	Interrupt requests (TGIC) by TGFC bit enabled

TGR Interrupt Enable D

0	Interrupt requests (TGID) by TGFD bit disabled
1	Interrupt requests (TGID) by TGFD bit enabled

Overflow Interrupt Enable

0	Interrupt requests (TCIV) by TCFV disabled
1	Interrupt requests (TCIV) by TCFV enabled

A/D Conversion Start Request Enable

0	A/D conversion start request generation disabled
1	A/D conversion start request generation enabled

0	[Clearing conditions] • When DTC is activated by TGIA inte DISEL bit of MRB in DTC is 0 • When DMAC is activated by TGIA i DTA bit of DMABCR in DMAC is 1 • When 0 is written to TGFA after readi
1	[Setting conditions] • When TCNT=TGRA while TGRA is ing as output compare register • When TCNT value is transferred to input capture signal while TGRA is f input capture register

Input Capture/Output Compare Flag B

0	[Clearing conditions] • When DTC is activated by TGIB interrupt whil of MRB in DTC is 0 • When 0 is written to TGFB after reading TGF
1	[Setting conditions] • When TCNT = TGRB while TGRB is functioni output compare register • When TCNT value is transferred to TGRB by capture signal while TGRB is functioning as i register

Input Capture/Output Compare Flag C

0	[Clearing conditions] • When DTC is activated by TGIC interrupt while DISEL bit DTC is 0 • When 0 is written to TGFC after reading TGFC = 1
1	[Setting conditions] • When TCNT = TGRC while TGRC is functioning as outpu register • When TCNT value is transferred to TGRC by input captur while TGRC is functioning as input capture register

Input Capture/Output Compare Flag D

0	[Clearing conditions] • When DTC is activated by TGID interrupt while DISEL bit of MRB is 0 • When 0 is written to TGFD after reading TGFD = 1
1	[Setting conditions] • When TCNT = TGRD while TGRD is functioning as output compa • When TCNT value is transferred to TGRD by input capture signa TGRD is functioning as input capture register

Overflow Flag

0	[Clearing condition] When 0 is written to TCFV after reading TCFV = 1
1	[Setting condition] When the TCNT value overflows (changes from H'FFFF to H'0000)

Note: * Can only be written with 0 for flag clearing.

Up-counter

TGR3A—Timer General Register 3A	**H'FE88**
TGR3B—Timer General Register 3B	**H'FE8A**
TGR3C—Timer General Register 3C	**H'FE8C**
TGR3D—Timer General Register 3D	**H'FE8E**

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
Initial value	:	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Read/Write	:	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

		1	Internal clock: counts on φ/4
	1	0	Internal clock: counts on φ/16
		1	Internal clock: counts on φ/64
1	0	0	External clock: counts on TCl
		1	External clock: counts on TCl
	1	0	Internal clock: counts on φ/10
		1	Counts on TCNT5 overflow/u

Note: This setting is ignored when channel counting mode.

Clock Edge

0	0	Count at rising edge
	1	Count at falling edge
1	—	Count at both edges

Note: This setting is ignored when channel 4 is in phase counting mode.

Counter Clear

0	0	TCNT clearing disabled
	1	TCNT cleared by TGRA compare match/input capture
1	0	TCNT cleared by TGRB compare match/input capture
	1	TCNT cleared by counter clearing for another channel performing synchronous clearing/synchronous operation*

Note: * Synchronous operating setting is performed by setting the SYNC bit TSYR to 1.

0	0	0	0	Normal ope
			1	Reserved
		1	0	PWM mode
			1	PWM mode
	1	0	0	Phase coun
			1	Phase coun
		1	0	Phase coun
			1	Phase coun
1	*	*	*	—

Legend: *: Don't care

Note: MD3 is a reserved bit. I
it should always be writ

			1	is output compare register	Initial output is 0 output	0 output at compar
		1	0			1 output at compar
			1			Toggle output at co
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compar
		1	0			1 output at compar
			1			Toggle output at co
1	0	0	0	TGR4A is input capture register	Capture input source is TIOCA4 pin	Input capture at ris
			1			Input capture at fal
		1	*			Input capture at bo
	1	*	*		Capture input source is TGR3A compare match/ input capture	Input capture at ge TGR3A compare m capture

Legend: *: Don't care

TGR4B I/O Control

0	0	0	0	TGR4B is output compare register	Output disabled	
			1		Initial output is 0 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
1	0	0	0	TGR4B is input capture register	Capture input source is TIOCB4 pin	Input capture at rising edge
			1			Input capture at falling edge
		1	*			Input capture at both edges
	1	*	*		Capture input source is TGR3C compare match/ input capture	Input capture at generation of TGR3C compare match/input capture

Legend: *: Don't care

	by TGFA bit d
1	Interrupt requ by TGFA bit e

TGR Interrupt Enable B

0	Interrupt requests TGFB bit disabled
1	Interrupt requests TGFB bit enabled

Overflow Interrupt Enable

0	Interrupt requests (TCIV) by TCFV disabled
1	Interrupt requests (TCIV) by TCFV enabled

Underflow Interrupt Enable

0	Interrupt requests (TCIU) by TCFU disabled
1	Interrupt requests (TCIU) by TCFU enabled

A/D Conversion Start Request Enable

0	A/D conversion start request generation disabled
1	A/D conversion start request generation enabled

	• When DMAC is activated by TGIA DTA bit of DMABCR in DMAC is 1 • When 0 is written to TGFA after read
1	[Setting conditions] • When TCNT = TGRA while TGRA as output compare register • When TCNT value is transferred to input capture signal while TGRA is as input capture register

Input Capture/Output Compare Flag B

0	[Clearing conditions] • When DTC is activated by TGIB interrupt bit of MRB in DTC is 0 • When 0 is written to TGFB after reading T
1	[Setting conditions] • When TCNT = TGRB while TGRB is funct output compare register • When TCNT value is transferred to TGRB capture signal while TGRB is functioning a capture register

Overflow Flag

0	[Clearing condition] When 0 is written to TCFV after reading TCFV = 1
1	[Setting condition] When the TCNT value overflows (changes from H'FFFF to H'0000)

Underflow Flag

0	[Clearing condition] When 0 is written to TCFU after reading TCFU = 1
1	[Setting condition] When the TCNT value underflows (changes from H'0000 to H'FFFF)

Count Direction Flag

0	TCNT counts down
1	TCNT counts up

Note: * Can only be written with 0 for flag clearing.

Up/down-counter*

Note: * This timer counter can be used as an up/down-counter only in phase counti
mode or when performing overflow/underflow counting on another channel.
In other cases it functions as an up-counter.

TGR4A—Timer General Register 4A **H'FE98**
TGR4B—Timer General Register 4B **H'FE9A**

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
Initial value	:	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Read/Write	:	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

		1	Internal clock: counts on φ/4
	1	0	Internal clock: counts on φ/16
		1	Internal clock: counts on φ/64
1	0	0	External clock: counts on TCL
		1	External clock: counts on TCL
	1	0	Internal clock: counts on φ/256
		1	External clock: counts on TCL

Note: This setting is ignored when channel counting mode.

Clock Edge

0	0	Count at rising edge
	1	Count at falling edge
1	—	Count at both edges

Note: This setting is ignored when channel 5 is in phase counting mode.

Counter Clear

0	0	TCNT clearing disabled
	1	TCNT cleared by TGRA compare match/input capture
1	0	TCNT cleared by TGRB compare match/input capture
	1	TCNT cleared by counter clearing for another channel performing synchronous clearing/synchronous operation*

Note: * Synchronous operating setting is performed by setting the SYNC bit TSYR to 1.

0	0	0	0	Normal op
			1	Reserved
		1	0	PWM mod
			1	PWM mod
	1	0	0	Phase cou
			1	Phase cou
		1	0	Phase cou
			1	Phase cou
1	*	*	*	—

Legend: *: Don't care

Note: MD3 is a reserved bit.
it should always be wr

			1	is output compare register	Initial output is 0 output	0 output at compar
		1	0			1 output at compar
			1			Toggle output at co
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compar
		1	0			1 output at compar
			1			Toggle output at co
1	*	0	0	TGR5A is input capture register	Capture input source is TIOCA5 pin	Input capture at ris
			1			Input capture at fal
		1	*			Input capture at bo

Legend: *: Don't care

TGR5B I/O Control

0	0	0	0	TGR5B is output compare register	Output disabled	
			1		Initial output is 0 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
1	*	0	0	TGR5B is input capture register	Capture input source is TIOCB5 pin	Input capture at rising edge
			1			Input capture at falling edge
		1	*			Input capture at both edges

Legend: *: Don't care

	by TGFA bit d
1	Interrupt requ by TGFA bit e

TGR Interrupt Enable B

0	Interrupt requests by TGFB bit disabl
1	Interrupt requests by TGFB bit enable

Overflow Interrupt Enable

0	Interrupt requests (TCIV) by TCFV disabled
1	Interrupt requests (TCIV) by TCFV enabled

Underflow Interrupt Enable

0	Interrupt requests (TCIU) by TCFU disabled
1	Interrupt requests (TCIU) by TCFU enabled

A/D Conversion Start Request Enable

0	A/D conversion start request generation disabled
1	A/D conversion start request generation enabled

	DISEL bit of MRB in DTC is 0 • When DMAC is activated by TGIA DTA bit of DMABCR in DMAC is • When 0 is written to TGFA after rea
1	[Setting conditions] • When TCNT = TGRA while TGRA as output compare register • When TCNT value is transferred input capture signal while TGRA as input capture register

Input Capture/Output Compare Flag B

0	[Clearing conditions] • When DTC is activated by TGIB interrup bit of MRB in DTC is 0 • When 0 is written to TGFB after reading
1	[Setting conditions] • When TCNT = TGRB while TGRB is fun output compare register • When TCNT value is transferred to TGR capture signal while TGRB is functioning capture register

Overflow Flag

0	[Clearing condition] When 0 is written to TCFV after reading TCFV = 1
1	[Setting condition] When the TCNT value overflows (changes from H'FFFF to H'0000)

Underflow Flag

0	[Clearing condition] When 0 is written to TCFU after reading TCFU = 1
1	[Setting condition] When the TCNT value underflows (changes from H'0000 to H'FFFF)

Count Direction Flag

0	TCNT counts down
1	TCNT counts up

Note: * Can only be written with 0 for flag clearing.

Up/down-counter*

Note: * This timer counter can be used as an up/down-counter only in phase countir mode or when performing overflow/underflow counting on another channel. In other cases it functions as an up-counter.

TGR5A—Timer General Register 5A **H'FEA8**
TGR5B—Timer General Register 5B **H'FEAA**

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
Initial value	:	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Read/Write	:	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

P1DDR—Port 1 Data Direction Register **H'FEB0**

Bit	:	7	6	5	4	3	2	1
		P17DDR	P16DDR	P15DDR	P14DDR	P13DDR	P12DDR	P11DDR
Initial value	:	0	0	0	0	0	0	0
Read/Write	:	W	W	W	W	W	W	W

Specify input or output for individual port 1 pins

Specify input or output for individual port 2 pins

P3DDR—Port 3 Data Direction Register H'FEB2

Bit	7	6	5	4	3	2	1
	—	—	P35DDR	P34DDR	P33DDR	P32DDR	P31DDR
Initial value	1	1	0	0	0	0	0
Read/Write	—	—	W	W	W	W	W

Specify input or output for individual port 3 p

P5DDR—Port 5 Data Direction Register H'FEB4

Bit	7	6	5	4	3	2	1	
	—	—	—	—	P53DDR	P52DDR	P51DDR	P50
Initial value	1	1	1	1	0	0	0	
Read/Write	—	—	—	—	W	W	W	

Specify input or output for individual p

Specify input or output for individual port 6 pins

PADDR—Port A Data Direction Register H'FEB9

Bit	:	7	6	5	4	3	2	1
		PA7DDR	PA6DDR	PA5DDR	PA4DDR	PA3DDR	PA2DDR	PA1DD
Initial value	:	0	0	0	0	0	0	0
Read/Write	:	W	W	W	W	W	W	W

Specify input or output for individual port A pins

PBDDR—Port B Data Direction Register H'FEBA

Bit	:	7	6	5	4	3	2	1
		PB7DDR	PB6DDR	PB5DDR	PB4DDR	PB3DDR	PB2DDR	PB1DDR
Initial value	:	0	0	0	0	0	0	0
Read/Write	:	W	W	W	W	W	W	W

Specify input or output for individual port B pins

Specify input or output for individual port C pins

PDDDR—Port D Data Direction Register H'FEBC

Bit	:	7	6	5	4	3	2	1
		PD7DDR	PD6DDR	PD5DDR	PD4DDR	PD3DDR	PD2DDR	PD1DDR
Initial value	:	0	0	0	0	0	0	0
Read/Write	:	W	W	W	W	W	W	W

Specify input or output for individual port D pins

PEDDR—Port E Data Direction Register H'FEBD

Bit	:	7	6	5	4	3	2	1
		PE7DDR	PE6DDR	PE5DDR	PE4DDR	PE3DDR	PE2DDR	PE1DDR
Initial value	:	0	0	0	0	0	0	0
Read/Write	:	W	W	W	W	W	W	W

Specify input or output for individual port E pins

Initial value	:	0	0	0	0	0	0	0
Read/Write	:	W	W	W	W	W	W	W

Specify input or output for individual port F pins

PGDDR—Port G Data Direction Register H'FEBF

Bit	:	7	6	5	4	3	2	1
		—	—	—	PG4DDR	PG3DDR	PG2DDR	PG1DDR
Modes 1, 4, 5								
Initial value	:	1	1	1	1	0	0	0
Read/Write	:	—	—	—	W	W	W	W
Modes 2, 3, 6, 7								
Initial value	:	1	1	1	0	0	0	0
Read/Write	:	—	—	—	W	W	W	W

Specify input or output for individual po

Correspondence between Interrupt Sources and ICR Settings

Register	Bits						
	7	6	5	4	3	2	1
ICRA	IRQ_0	IRQ_1	IRQ_2 IRQ_3	IRQ_4 IRQ_5	IRQ_6 IRQ_7	DTC	Watchdog timer
ICRB	—	A/D converter	TPU channel 0	TPU channel 1	TPU channel 2	TPU channel 3	TPU channel 4
ICRC	8-bit timer channel 0	8-bit timer channel 1	DMAC	SCI channel 0	SCI channel 1	SCI channel 2	—

IPRI—Interrupt Priority Register I **H'FECC** **Interrup**
IPRJ—Interrupt Priority Register J **H'FECD** **Interrup**
IPRK—Interrupt Priority Register K **H'FECE** **Interrup**

Bit	:	7	6	5	4	3	2	1
		—	IPR6	IPR5	IPR4	—	IPR2	IPR1
Initial value	:	0	1	1	1	0	1	1
Read/Write	:	—	R/W	R/W	R/W	—	R/W	R/W

Set priority (levels 7 to 0) for interrupt sources

Correspondence between Interrupt Sources and IPR Settings

Register	Bits	
	6 to 4	2 to 0
IPRA	IRQ_0	IRQ_1
IPRB	IRQ_2 IRQ_3	IRQ_4 IRQ_5
IPRC	IRQ_6 IRQ_7	DTC
IPRD	WDT	Refresh timer
IPRE	—	A/D converter
IPRF	TPU channel 0	TPU channel 1
IPRG	TPU channel 2	TPU channel 3
IPRH	TPU channel 4	TPU channel 5
IPRI	8-bit timer channel 0	8-bit timer channel 1
IPRJ	DMAC	SCI channel 0
IPRK	SCI channel 1	SCI channel 2

Initial value :	0	0	0	0	0	0	0
Read/Write :	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Area 7 to 0 Bus Width Control

0	Area n is designated for 16-bit a
1	Area n is designated for 8-bit ac

Note: n = 7 to 0

ASTCR—Access State Control Register **H'FED1** **Bus**

Bit :	7	6	5	4	3	2	1
	AST7	AST6	AST5	AST4	AST3	AST2	AST1
Initial value :	1	1	1	1	1	1	1
Read/Write :	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Area 7 to 0 Access State Control

0	Area n is designated for 2-state access Wait state insertion in area n external space is disab
1	Area n is designated for 3-state access Wait state insertion in area n external space is enabl

Note: n = 7 to 0

Area 4 Wait Control

0	0	Program wait not in
	1	1 program wait stat
1	0	2 program wait stat
	1	3 program wait stat

Area 5 Wait Control

0	0	Program wait not inserted
	1	1 program wait state inserted
1	0	2 program wait states inserted
	1	3 program wait states inserted

Area 6 Wait Control

0	0	Program wait not inserted
	1	1 program wait state inserted
1	0	2 program wait states inserted
	1	3 program wait states inserted

Area 7 Wait Control

0	0	Program wait not inserted
	1	1 program wait state inserted
1	0	2 program wait states inserted
	1	3 program wait states inserted

Area 0 Wait Control

0	0	Program wait not inse
	1	1 program wait state
1	0	2 program wait states
	1	3 program wait states

Area 1 Wait Control

0	0	Program wait not inserted
	1	1 program wait state inserted
1	0	2 program wait states inserted
	1	3 program wait states inserted

Area 2 Wait Control

0	0	Program wait not inserted
	1	1 program wait state inserted
1	0	2 program wait states inserted
	1	3 program wait states inserted

Area 3 Wait Control

0	0	Program wait not inserted
	1	1 program wait state inserted
1	0	2 program wait states inserted
	1	3 program wait states inserted

RMTS2	RMTS1	RMTS0	Area 5	Area 4
0	0	0	Norma	
		1	Normal sp	
	1	0	Normal space	
		1	DRAM	
1	0	0	Normal	
		1	Normal spa	
	1	0	Normal space	
		1	PSRAM	

Burst Cycle Select 0

0	Max. 4 words in burst access
1	Max. 8 words in burst access

Burst Cycle Select 1

0	Burst cycle comprises 1 state
1	Burst cycle comprises 2 states

Area 0 Burst ROM Enable

0	Area 0 is basic bus interface
1	Area 0 is burst ROM interface

Idle Cycle Insert 0

0	Idle cycle not inserted in case of successive external read and extern
1	Idle cycle inserted in case of successive external read and external w

Idle Cycle Insert 1

0	Idle cycle not inserted in case of successive external read cycles in different are
1	Idle cycle inserted in case of successive external read cycles in different areas

1	Wait input by $\overline{WAIT}$ pin enabled

Write Data Buffer Enable

0	Write data buffer function not used
1	Write data buffer function used

Area Partition Unit Select

0	Area partition unit is 128 kbytes (1 Mb
1	Area partition unit is 2 Mbytes (16 Mb

DACK Timing Select

0	When DMAC single address transfer is performed DRAM/PSRAM space, full access is always execu $\overline{DACK}$ signal goes low from Tr or T1 cycle
1	Burst access is possible when DMAC single addr transfer is performed in DRAM/PSRAM space $\overline{DACK}$ signal goes low from Tc1 or T2 cycle

LCAS Pin Select

0	$\overline{LCAS}$ pin used for 2-CAS type DRAM interface $\overline{LCAS}$ signal ($\overline{BREQO}$ output and $\overline{WAIT}$ input cannot be used when $\overline{LCAS}$ signal is used)
1	$\overline{LWR}$ pin used for 2-CAS type DRAM interface $\overline{LCAS}$ signal (RAS down mode cannot be used)

External Addresses H'010000 to H'01FFFF Enable

0	On-chip ROM (H8S/2655) or reserved area* (H8S/2653)
1	External addresses (in external expansion mode) or reserved area (in single-chip mode)

Notes: * Do not access a reserved area.

BREQO Pin Enable

0	$\overline{BREQO}$ output disabled
1	$\overline{BREQO}$ output enabled

Bus Release Enable

0	External bus release is disabled
1	External bus release is enabled

0	0	No wait sta
	1	1 wait state
1	0	2 wait state
	1	3 wait state

Multiplex Shift Count

0	0	8-bit shift
	1	9-bit shift
1	0	10-bit shift
	1	—

2-CAS Method/2-WE Method Select

0	2-CAS method selected: $\overline{\text{CASH}}$, $\overline{\text{CASL}}$, $\overline{\text{WE}}$ signa
1	2-WE method selected: $\overline{\text{CAS}}$, $\overline{\text{UWE}}$, $\overline{\text{LWE}}$ signals

RAS/CS Down Mode

0	DRAM interface: RAS up mode selected
1	DRAM interface: RAS down mode selected

Burst Access Enable

0	Burst disabled (always full access)
1	• For DRAM space access Access in fast page mode • For PSRAM space access Access in static column mode

TP Cycle Control

0	1-state precharge cycle is inserted
1	2-state precharge cycle is inserted

0	0	0	Count operation
		1	Count uses ϕ/2
	1	0	Count uses ϕ/8
		1	Count uses ϕ/32
1	0	0	Count uses ϕ/12
		1	Count uses ϕ/51
	1	0	Count uses ϕ/20
		1	Count uses ϕ/40

Compare Match Interrupt Enable

0	Interrupt request (CMI) by CMF flag di
1	Interrupt request (CMI) by CMF flag e

Compare Match Flag

0	[Clearing condition] Cleared by reading the CMF flag when CMF = 1, writing 0 to the CMF flag
1	[Setting condition] Set when RTCNT = RTCOR

Refresh Mode

0	DRAM interface: CAS-before-RAS refreshing used
	PSRAM interface: Auto-refreshing used
1	Self-refreshing used

RAS-CAS Wait

0	Wait state insertion in CAS-before-RAS refreshing disabled $\overline{RAS}$ falls in T_{Rr} cycle
1	One wait state inserted in CAS-before-RAS refreshing $\overline{RAS}$ falls in T_{Rc1} cycle

Refresh Control

0	Refresh control is not performed
1	Refresh control is performed

RTCOR—Refresh Time Constant Register H'FED9 Bu

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

Sets the period for compare match operations with RTCN

MAR0AH—Memory Address Register 0AH H'FEE0
MAR0AL—Memory Address Register 0AL H'FEE2

Bit	:	31	30	29	28	27	26	25	24	23	22	21	20	19	18
MAR0AH	:	—	—	—	—	—	—	—	—						
Initial value	:	0	0	0	0	0	0	0	0	*	*	*	*	*	*
Read/Write	:	—	—	—	—	—	—	—	—	R/W	R/W	R/W	R/W	R/W	R/W

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
MAR0AL	:														
Initial value	:	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Read/Write	:	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

In short address mode: Specifies transfer source/transfer destination a
In full address mode: Specifies transfer source address

Legend: *: Undefined

In short address mode: Specifies transfer source/transfer destination ad
In full address mode: Not used

Legend: *: Undefined

ETCR0A—Transfer Count Register 0A H'FEE6

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
ETCR0A	:														
Initial value	:	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Read/Write	:	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

Sequential mode Idle mode Normal mode	Transfer counter	
Repeat mode	Transfer number storage register	Transfer counter
Block transfer mode	Block size storage register	Block size counter

Legend: *: Undefined

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
MAR0BL	:														
Initial value	:	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Read/Write	:	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

In short address mode: Specifies transfer source/transfer destination a
In full address mode: Specifies transfer destination address

Legend: *: Undefined

IOAR0B—I/O Address Register 0B **H'FEEC**

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
IOAR0B	:														
Initial value	:	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Read/Write	:	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

In short address mode: Specifies transfer source/transfer destination
In full address mode: Not used

Legend: *: Undefined

mode and idle mode		Transfer counter
Repeat mode	Transfer number storage register	Transfer counte
Block transfer mode	Block transfer counter	

Legend: *: Undefined

Note: Not used in normal mode.

MAR1AH—Memory Address Register 1AH **H'FEF0**
MAR1AL—Memory Address Register 1AL **H'FEF2**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18
MAR1AH	—	—	—	—	—	—	—	—						
Initial value	0	0	0	0	0	0	0	0	*	*	*	*	*	*
Read/Write	—	—	—	—	—	—	—	—	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2
MAR1AL														
Initial value	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Read/Write	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

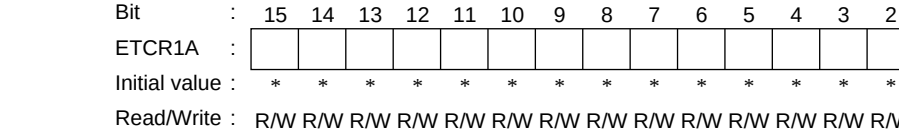
In short address mode: Specifies transfer source/transfer destination ad
In full address mode: Specifies transfer source address

Legend: *: Undefined

In short address mode: Specifies transfer source/transfer destination
In full address mode: Not used

Legend: *: Undefined

ETCR1A—Transfer Count Register 1A H'FEF6



Legend: *: Undefined

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
MAR1BL	:														
Initial value	:	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Read/Write	:	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

In short address mode: Specifies transfer source/transfer destination ad
In full address mode: Specifies transfer destination address

Legend: *: Undefined

IOAR1B—I/O Address Register 1B **H'FEFC**

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
IOAR1B	:														
Initial value	:	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Read/Write	:	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

In short address mode: Specifies transfer source/transfer destination ac
In full address mode: Not used

Legend: *: Undefined

Repeat mode	Transfer number storage register	Transfer coun
Block transfer mode	Block transfer counter	

Legend: *: Undefined

Note: Not used in normal mode.

	are disabled
1	Writes to all bits in D and bits 8, 4, and 0 i are enabled

Write Enable 0B

0	Writes to all bits in DMACR0B 5, and 1 in DMABCR, and bit DMATCR are disabled
1	Writes to all bits in DMACR0B 5, and 1 in DMABCR, and bit DMATCR are enabled

Write Enable 1A

0	Writes to all bits in DMACR1A, and bits 10, 6, and 2 in DMABCR are disabled
1	Writes to all bits in DMACR1A, and bits 10, 6, and 2 in DMABCR are enabled

Write Enable 1B

0	Writes to all bits in DMACR1B, bits 11, 7, and 3 in DMABCR, and bit 5 in DMATCR are disabled
1	Writes to all bits in DMACR1B, bits 11, 7, and 3 in DMABCR, and bit 5 in DMATCR are enabled

0	$\overline{TEND}_0$ pin output disabled
1	$\overline{TEND}_0$ pin output enabled

Transfer End Enable 1

0	$\overline{TEND}_1$ pin output disabled
1	$\overline{TEND}_1$ pin output enabled

Initial value :	0	0	0	0	0	0	0	0
Read/Write :	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Block Direction/Block Enable

0	0	Transfer in normal mode
	1	Transfer in block transfer mode, destination side i
1	0	Transfer in normal mode
	1	Transfer in block transfer mode, source side is bl

Source Address Increment/Decrement

0	0	MARA is fixed
	1	MARA is incremented after a data transfer
1	0	MARA is fixed
	1	MARA is decremented after a data transfer

Data Transfer Size

0	Byte-size transfer
1	Word-size transfer

0	0	0	0	—	—
			1	Activated by A/D converter conversion end interrupt	—
		1	0	Activated by $\overline{\text{DREQ}}$ pin falling edge input	Activa pin fal
			1	Activated by $\overline{\text{DREQ}}$ pin low-level input	Activa pin lo
	1	0	0	Activated by SCI channel 0 transmission complete interrupt	—
			1	Activated by SCI channel 0 reception complete interrupt	—
		1	0	Activated by SCI channel 1 transmission complete interrupt	Auto-r steal)
			1	Activated by SCI channel 1 reception complete interrupt	Auto-r
1	0	0	0	Activated by TPU channel 0 compare match/input capture A interrupt	—
			1	Activated by TPU channel 1 compare match/input capture A interrupt	—
		1	0	Activated by TPU channel 2 compare match/input capture A interrupt	—
			1	Activated by TPU channel 3 compare match/input capture A interrupt	—
	1	0	0	Activated by TPU channel 4 compare match/input capture A interrupt	—
			1	Activated by TPU channel 5 compare match/input capture A interrupt	—
		1	*	—	—

Legend: *: Don't care

Destination Address Increment/Decrement

0	0	MARB is fixed
	1	MARB is incremented after a data transfer
1	0	MARB is fixed
	1	MARB is decremented after a data transfer

0	Dual address mode: Transfer with MAR as source address and IOAR as destination address Single address mode: Transfer with MAR as source address and $\overline{\text{DACK}}$ pin as write strobe
1	Dual address mode: Transfer with IOAR as source address and MAR as destination address Single address mode: Transfer with $\overline{\text{DACK}}$ pin as read strobe and MAR as destination address

Repeat Enable

0	Transfer in sequential mode
1	Transfer in repeat mode or idle mode

Data Transfer Increment/Decrement

0	MAR is incremented after a data transfer
1	MAR is decremented after a data transfer

Data Transfer Size

0	Byte-size transfer
1	Word-size transfer

			1	Activated by A/D converte end interrupt	
		1	0	—	Activated falling ed
			1	—	Activated low-level
	1	0	0	Activated by SCI channel complete interrupt	
			1	Activated by SCI channel complete interrupt	
		1	0	Activated by SCI channel complete interrupt	
			1	Activated by SCI channel complete interrupt	
1	0	0	0	Activated by TPU channe match/input capture A int	
			1	Activated by TPU channe match/input capture A int	
		1	0	Activated by TPU channe match/input capture A int	
			1	Activated by TPU channe match/input capture A int	
	1	0	0	Activated by TPU channe match/input capture A int	
			1	Activated by TPU channe match/input capture A int	
		1	0	—	
			1	—	

0	Clearing of selected internal interrupt sou DMA transfer is disabled
1	Clearing of selected internal interrupt sou DMA transfer is enabled

Channel 1 Data Transfer Acknowledge

0	Clearing of selected internal interrupt source at t DMA transfer is disabled
1	Clearing of selected internal interrupt source at t DMA transfer is enabled

Channel 0 Full Address Enable

0	Short address mode
1	Full address mode

Channel 1 Full Address Enable

0	Short address mode
1	Full address mode

(Continued o

1	Transfer end interr

Channel 0 Data Transfer Interrup
Enable B

0	Transfer suspended inter
1	Transfer suspended inter

Channel 1 Data Transfer
Interrupt Enable A

0	Transfer end interrupt disabled
1	Transfer end interrupt enabled

Channel 1 Data Transfer Interrupt
Enable B

0	Transfer suspended interrupt disabled
1	Transfer suspended interrupt enabled

Channel 0 Data Transfer Enable

0	Data transfer disabled
1	Data transfer enabled

Channel 0 Data Transfer Master Enable

0	Data transfer disabled. In normal mode, cleared to 0 by an NMI interrupt
1	Data transfer enabled

Channel 1 Data Transfer Enable

0	Data transfer disabled
1	Data transfer enabled

Channel 1 Data Transfer Master Enable

0	Data transfer disabled. In normal mode, cleared to 0 by an NMI interrupt
1	Data transfer enabled

(Continued on

1	Clearing of selected interrupt source at ti transfer is enabled

Channel 0B Data Transfer Acknowl

0	Clearing of selected internal inte at time of DMA transfer is disable
1	Clearing of selected internal inte source at time of DMA transfer is

Channel 1A Data Transfer Acknowledge

0	Clearing of selected internal interrupt source at time of DMA transfer is disabled
1	Clearing of selected internal interrupt source at time of DMA transfer is enabled

Channel 1B Data Transfer Acknowledge

0	Clearing of selected internal interrupt source at time of DMA transfer is disabled
1	Clearing of selected internal interrupt source at time of DMA transfer is enabled

Channel 0B Single Address Enable

0	Transfer in dual address mode
1	Transfer in single address mode

Channel 1B Single Address Enable

0	Transfer in dual address mode
1	Transfer in single address mode

Channel 0 Full Address Enable

0	Short address mode
1	Full address mode

Channel 1 Full Address Enable

0	Short address mode
1	Full address mode

(Continued o

Channel 0B Data Transfer Interrupt Enable

0	Transfer end interrupt disabled
1	Transfer end interrupt enabled

Channel 1A Data Transfer Interrupt Enable

0	Transfer end interrupt disabled
1	Transfer end interrupt enabled

Channel 1B Data Transfer Interrupt Enable

0	Transfer end interrupt disabled
1	Transfer end interrupt enabled

Channel 0A Data Transfer Enable

0	Data transfer disabled
1	Data transfer enabled

Channel 0B Data Transfer Enable

0	Data transfer disabled
1	Data transfer enabled

Channel 1A Data Transfer Enable

0	Data transfer disabled
1	Data transfer enabled

Channel 1B Data Transfer Enable

0	Data transfer disabled
1	Data transfer enabled

Read/Write	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

IRQ_7 to IRQ_4 Sense Control

ISCRL

Bit	:	7	6	5	4	3	2	1
		IRQ3SCB	IRQ3SCA	IRQ2SCB	IRQ2SCA	IRQ1SCB	IRQ1SCA	IRQ0SCB
Initial value	:	0	0	0	0	0	0	0
Read/Write	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

IRQ_3 to IRQ_0 Sense Control

IRQ_nSCB	IRQ_nSCA	Interrupt Request Generation
0	0	$\overline{IRQ}_n$ input low level
	1	Falling edge of $\overline{IRQ}_n$ input
1	0	Rising edge of $\overline{IRQ}_n$ input
	1	Both falling and rising edges of $\overline{IRQ}_n$ input

Note: n = 7 to 0

0	IRQ_n interrupt disabled
1	IRQ_n interrupt enabled

Note: n = 7 to 0

ISR—IRQ Status Register **H'FF2F** **Interrupt**

Bit	:	7	6	5	4	3	2	1
		IRQ7F	IRQ6F	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F
Initial value	:	0	0	0	0	0	0	0
Read/Write	:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Indicate the status of IRQ_7 to IRQ_0 interrupt requests

Note: * Can only be written with 0 for flag clearing.

0	DTC activation by this interrupt is disabled [Clearing conditions] • When the DISEL bit is 1 and data transfer h • When the specified number of transfers hav
1	DTC activation by this interrupt is enabled [Holding condition] When the DISEL bit is 0 and the specified nur transfers have not ended

Correspondence between Interrupt Sources and DTCER

	Bits						
Register	**7**	**6**	**5**	**4**	**3**	**2**	**1**
DTCERA	IRQ0	IRQ1	IRQ2	IRQ3	IRQ4	IRQ5	IRQ6
DTCERB	—	ADI	TGI0A	TGI0B	TGI0C	TGI0D	TGI1
DTCERC	TGI2A	TGI2B	TGI3A	TGI3B	TGI3C	TGI3D	TGI4
DTCERD	—	—	TGI5A	TGI5B	CMIA0	CMIB0	CMIA
DTCERE	DMTEND0A	DMTEND0B	DMTEND1A	DMTEND1B	RXI0	TXI0	RXI1
DTCERF	RXI2	TXI2	—	—	—	—	—

DTC Software Activation Enable

0	DTC software activation is disabled [Clearing condition] When the DISEL bit is 0 and the specified number of transfers ha not ended
1	DTC software activation is enabled [Holding conditions] • When the DISEL bit is 1 and data transfer has ended • When the specified number of transfers have ended • During data transfer due to software activation

Note: * A value of 1 can always be written to the SWDTE bit, but 0 can only be writ
is read.

	control signals are high-impedance
1	In software standby mode, address bu control signals retain output state

Standby Timer Select

0	0	0	Standby time = 8192 states
		1	Standby time = 16384 states
	1	0	Standby time = 32768 states
		1	Standby time = 65536 states
1	0	0	Standby time = 131072 states
		1	Standby time = 262144 states
	1	0	Reserved
		1	Standby time = 16 states

Software Standby

0	Transition to sleep mode after execution of SLEEP instruction
1	Transition to software standby mode after execution of SLEEP instruction

1	On-chip RAM

NMI Input Edge Select

0	Falling edge
1	Rising edge

Interrupt Control Mode Selection

0	0	Interrupt control mode 0
	1	Interrupt control mode 1
1	0	Interrupt control mode 2
	1	Interrupt control mode 3

MAC Saturation

0	Non-saturating calculation for MAC instruction
1	Saturating calculation for MAC instruction

0	0	0	Bus master is in high-speed m
		1	Medium-speed clock is φ/2
	1	0	Medium-speed clock is φ/4
		1	Medium-speed clock is φ/8
1	0	0	Medium-speed clock is φ/16
		1	Medium-speed clock is φ/32
	1	—	—

φ Clock Output Control

PSTOP	Normal Operation	Sleep Mode	Software Standby Mode	Hai Stand
0	φ output	φ output	Fixed high	High i
1	Fixed high	Fixed high	Fixed high	High i

MDCR—Mode Control Register **H'FF3B**

Bit :	7	6	5	4	3	2	1	
	—	—	—	—	—	MDS2	MDS1	
Initial value :	1	0	0	0	0	—*	—*	
Read/Write :	—	—	—	—	—	R	R	

Current mode pin oper

Note: * Determined by pins MD_2 to MD_0

Specifies module stop mode

0	Module stop mode cleared
1	Module stop mode set

0	0	Compare match in
	1	Compare match in
1	0	Compare match in
	1	Compare match in

Output Trigger for Pulse Output Group 1

0	0	Compare match in TPU channel 0
	1	Compare match in TPU channel 1
1	0	Compare match in TPU channel 2
	1	Compare match in TPU channel 3

Output Trigger for Pulse Output Group 2

0	0	Compare match in TPU channel 0
	1	Compare match in TPU channel 1
1	0	Compare match in TPU channel 2
	1	Compare match in TPU channel 3

Output Trigger for Pulse Output Group 3

0	0	Compare match in TPU channel 0
	1	Compare match in TPU channel 1
1	0	Compare match in TPU channel 2
	1	Compare match in TPU channel 3

Pulse Output Group n Normal/Non-Overlap Operation Select

0	Normal operation in pulse output group n (values updated at compare match A in the TPU channel)
1	Non-overlapping operation in pulse output (independent 1 and 0 output at compare m or B in the selected TPU channel)

Note: n = 3 to 0

Pulse Output Group n Direct/Inverted Output

0	Inverted output for pulse output group n (low-level output at pin for a 1 in PODRH)
1	Direct output for pulse output group n (high-level output at pin for a 1 in PODRH)

Note: n = 3 to 0

Read/Write :	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Pulse Output Enable/Disable

0	Pulse outputs PO_{15} to PO_8 are disabled
1	Pulse outputs PO_{15} to PO_8 are enabled

NDERL

Bit :	7	6	5	4	3	2	1
	NDER7	NDER6	NDER5	NDER4	NDER3	NDER2	NDER1
Initial value :	0	0	0	0	0	0	0
Read/Write :	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Pulse Output Enable/Disable

0	Pulse outputs PO_7 to PO_0 are disabled
1	Pulse outputs PO_7 to PO_0 are enabled

Read/Write :	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Stores output data for use in pulse output

PODRL

Bit :	7	6	5	4	3	2	1
	POD7	POD6	POD5	POD4	POD3	POD2	POD1
Initial value :	0	0	0	0	0	0	0
Read/Write :	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Stores output data for use in pulse output

Note: * A bit that has been set for pulse output by NDER is read-only.

Initial value :	0	0	0	0	0	0	0
Read/Write :	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Stores the next data for pulse output groups 3 and 2

(b) Address: H'FF4E

Bit :	7	6	5	4	3	2	1
	—	—	—	—	—	—	—
Initial value :	1	1	1	1	1	1	1
Read/Write :	—	—	—	—	—	—	—

(2) When pulse output group output triggers are different

(a) Address: H'FF4C

Bit :	7	6	5	4	3	2	1
	NDR15	NDR14	NDR13	NDR12	—	—	—
Initial value :	0	0	0	0	1	1	1
Read/Write :	R/W	R/W	R/W	R/W	—	—	—

Stores the next data for pulse output group 3

(b) Address: H'FF4E

Bit :	7	6	5	4	3	2	1
	—	—	—	—	NDR11	NDR10	NDR
Initial value :	1	1	1	1	0	0	0
Read/Write :	—	—	—	—	R/W	R/W	R/W

Stores the next data for pulse

Initial value :	0	0	0	0	0	0	0
Read/Write :	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Stores the next data for pulse output groups 1 and 0

(b) Address: H'FF4F

Bit :	7	6	5	4	3	2	1
	—	—	—	—	—	—	—
Initial value :	1	1	1	1	1	1	1
Read/Write :	—	—	—	—	—	—	—

(2) When pulse output group output triggers are different

(a) Address: H'FF4D

Bit :	7	6	5	4	3	2	1
	NDR7	NDR6	NDR5	NDR4	—	—	—
Initial value :	0	0	0	0	1	1	1
Read/Write :	R/W	R/W	R/W	R/W	—	—	—

Stores the next data for pulse output group 1

(b) Address: H'FF4F

Bit :	7	6	5	4	3	2	1
	—	—	—	—	NDR3	NDR2	NDR1
Initial value :	1	1	1	1	0	0	0
Read/Write :	—	—	—	—	R/W	R/W	R/W

Stores the next data for pulse ou

Note: * Determined by the state of pins $P1_7$ to $P1_0$.

PORT2—Port 2 Register **H'FF51**

Bit :	7	6	5	4	3	2	1
	P27	P26	P25	P24	P23	P22	P21
Initial value :	—*	—*	—*	—*	—*	—*	—*
Read/Write :	R	R	R	R	R	R	R

State of port 2 pins

Note: * Determined by the state of pins $P2_7$ to $P2_0$.

PORT3—Port 3 Register **H'FF52**

Bit :	7	6	5	4	3	2	1
	—	—	P35	P34	P33	P32	P31
Initial value :	1	1	—*	—*	—*	—*	—*
Read/Write :	—	—	R	R	R	R	R

State of port 3 pins

Note: * Determined by the state of pins $P3_5$ to $P3_0$.

State of port 4 pins

Note: * Determined by the state of pins $P4_7$ to $P4_0$.

PORT5—Port 5 Register **H'FF54**

Bit :	7	6	5	4	3	2	1
	—	—	—	—	P53	P52	P51
Initial value :	1	1	1	1	—*	—*	—*
Read/Write :	—	—	—	—	R	R	R

State of port 5 pins

Note: * Determined by the state of pins $P5_3$ to $P5_0$.

PORT6—Port 6 Register **H'FF55**

Bit :	7	6	5	4	3	2	1
	P67	P66	P65	P64	P63	P62	P61
Initial value :	—*	—*	—*	—*	—*	—*	—*
Read/Write :	R	R	R	R	R	R	R

State of port 6 pins

Note: * Determined by the state of pins $P6_7$ to $P6_0$.

State of port A pins

Note: * Determined by the state of pins PA_7 to PA_0.

PORTB—Port B Register **H'FF5A**

Bit	:	7	6	5	4	3	2	1
		PB7	PB6	PB5	PB4	PB3	PB2	PB1
Initial value	:	—*	—*	—*	—*	—*	—*	—*
Read/Write	:	R	R	R	R	R	R	R

State of port B pins

Note: * Determined by the state of pins PB_7 to PB_0.

PORTC—Port C Register **H'FF5B**

Bit	:	7	6	5	4	3	2	1
		PC7	PC6	PC5	PC4	PC3	PC2	PC1
Initial value	:	—*	—*	—*	—*	—*	—*	—*
Read/Write	:	R	R	R	R	R	R	R

State of port C pins

Note: * Determined by the state of pins PC_7 to PC_0.

State of port D pins

Note: * Determined by the state of pins PD_7 to PD_0.

PORTE—Port E Register **H'FF5D**

Bit	:	7	6	5	4	3	2	1
		PE7	PE6	PE5	PE4	PE3	PE2	PE1
Initial value	:	—*	—*	—*	—*	—*	—*	—*
Read/Write	:	R	R	R	R	R	R	R

State of port E pins

Note: * Determined by the state of pins PE_7 to PE_0.

PORTF—Port F Register **H'FF5E**

Bit	:	7	6	5	4	3	2	1
		PF7	PF6	PF5	PF4	PF3	PF2	PF1
Initial value	:	—*	—*	—*	—*	—*	—*	—*
Read/Write	:	R	R	R	R	R	R	R

State of port F pins

Note: * Determined by the state of pins PF_7 to PF_0.

State of port G pins

Note: * Determined by the state of pins PG_4 to PG_0.

P1DR—Port 1 Data Register **H'FF60**

Bit	:	7	6	5	4	3	2	1
		P17DR	P16DR	P15DR	P14DR	P13DR	P12DR	P11DR
Initial value	:	0	0	0	0	0	0	0
Read/Write	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Stores output data for port 1 pins ($P1_7$ to $P1_0$)

P2DR—Port 2 Data Register **H'FF61**

Bit	:	7	6	5	4	3	2	1
		P27DR	P26DR	P25DR	P24DR	P23DR	P22DR	P21DR
Initial value	:	0	0	0	0	0	0	0
Read/Write	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Stores output data for port 2 pins ($P2_7$ to $P2_0$)

P5DR—Port 5 Data Register **H'FF64**

Bit	:	7	6	5	4	3	2	1
		—	—	—	—	P53DR	P52DR	P51DR
Initial value	:	1	1	1	1	0	0	0
Read/Write	:	—	—	—	—	R/W	R/W	R/W

Stores output data for port 5 pins (P5

P6DR—Port 6 Data Register **H'FF65**

Bit	:	7	6	5	4	3	2	1
		P67DR	P66DR	P65DR	P64DR	P63DR	P62DR	P61DR
Initial value	:	0	0	0	0	0	0	0
Read/Write	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Stores output data for port 6 pins ($P6_7$ to $P6_0$)

Stores output data for port A pins (PA_7 to PA_0)

PBDR—Port B Data Register **H'FF6A**

Bit	:	7	6	5	4	3	2	1
		PB7DR	PB6DR	PB5DR	PB4DR	PB3DR	PB2DR	PB1DR
Initial value	:	0	0	0	0	0	0	0
Read/Write	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Stores output data for port B pins (PB_7 to PB_0)

PCDR—Port C Data Register **H'FF6B**

Bit	:	7	6	5	4	3	2	1
		PC7DR	PC6DR	PC5DR	PC4DR	PC3DR	PC2DR	PC1DR
Initial value	:	0	0	0	0	0	0	0
Read/Write	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Stores output data for port C pins (PC_7 to PC_0)

Stores output data for port D pins (PD_7 to PD_0)

PEDR—Port E Data Register **H'FF6D**

Bit	:	7	6	5	4	3	2	1
		PE7DR	PE6DR	PE5DR	PE4DR	PE3DR	PE2DR	PE1DR
Initial value	:	0	0	0	0	0	0	0
Read/Write	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Stores output data for port E pins (PE_7 to PE_0)

PFDR—Port F Data Register **H'FF6E**

Bit	:	7	6	5	4	3	2	1
		PF7DR	PF6DR	PF5DR	PF4DR	PF3DR	PF2DR	PF1DR
Initial value	:	0	0	0	0	0	0	0
Read/Write	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Stores output data for port F pins (PF_7 to PF_0)

PAPCR—Port A MOS Pull-Up Control Register H'FF70

Bit	:	7	6	5	4	3	2	1	
		PA7PCR	PA6PCR	PA5PCR	PA4PCR	PA3PCR	PA2PCR	PA1PCR	P
Initial value	:	0	0	0	0	0	0	0	
Read/Write	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Controls the MOS input pull-up function incorporated into port A on a bit-b

PBPCR—Port B MOS Pull-Up Control Register H'FF71

Bit	:	7	6	5	4	3	2	1	
		PB7PCR	PB6PCR	PB5PCR	PB4PCR	PB3PCR	PB2PCR	PB1PCR	P
Initial value	:	0	0	0	0	0	0	0	
Read/Write	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Controls the MOS input pull-up function incorporated into port B on a bit-b

Controls the MOS input pull-up function incorporated into port C on a bit-by

PDPCR—Port D MOS Pull-Up Control Register H'FF73

Bit	:	7	6	5	4	3	2	1	
		PD7PCR	PD6PCR	PD5PCR	PD4PCR	PD3PCR	PD2PCR	PD1PCR	PD
Initial value	:	0	0	0	0	0	0	0	
Read/Write	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Controls the MOS input pull-up function incorporated into port D on a bit-by

PEPCR—Port E MOS Pull-Up Control Register H'FF74

Bit	:	7	6	5	4	3	2	1	
		PE7PCR	PE6PCR	PE5PCR	PE4PCR	PE3PCR	PE2PCR	PE1PCR	PE
Initial value	:	0	0	0	0	0	0	0	
Read/Write	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Controls the MOS input pull-up function incorporated into port E on a bit-by

PAODR—Port A Open Drain Control Register H'FF77

Bit	:	7	6	5	4	3	2	1
		PA7ODR	PA6ODR	PA5ODR	PA4ODR	PA3ODR	PA2ODR	PA1ODR
Initial value	:	0	0	0	0	0	0	0
Read/Write	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Controls the PMOS on/off status for each port A pin (PA_7 to P

0	0	φ clock
	1	φ/4 clock
1	0	φ/16 clock
	1	φ/64 clock

Multiprocessor Mode

0	Multiprocessor function disa
1	Multiprocessor format selec

Stop Bit Length

0	1 stop bit
1	2 stop bits

Parity Mode

0	Even parity
1	Odd parity

Parity Enable

0	Parity bit addition and checking disabled
1	Parity bit addition and checking enabled

Character Length

0	8-bit data
1	7-bit data*

Note: * When 7-bit data is selected, the MSB (bit 7) of TDR is not transmitt

Asynchronous Mode/Synchronous Mode Select

0	Asynchronous mode
1	Synchronous mode

0	0	φ clock
	1	φ/4 clock
1	0	φ/16 clock
	1	φ/64 clock

Multiprocessor Mode

0	Multiprocessor function
1	Multiprocessor format se

Stop Bit Length

0	1 stop bit
1	2 stop bits

Parity Mode

0	Even parity
1	Odd parity

Parity Enable

0	Parity bit addition and checking disabled
1	Parity bit addition and checking enabled

Character Length

0	8-bit data
1	7-bit data*

Note: * When 7-bit data is selected, the MSB (bit 7) of TDR is not trans

GSM Mode

0	Normal smart card interface mode operation • TEND flag generated 12.5 etu after beginning of start bit • Clock output on/off control only
1	GSM mode smart card interface mode operation • TEND flag generated 11.0 etu after beginning of start bit • Fixed high/low-level control possible (set in SCR) in addition to clock output on/off control

Note: etu (Elementary Time Unit): Interval for transfer of one bit

Note: See section 14.2.8, Bit Rate Register (BRR), for details.

		mode	as serial clock out
	1	Asynchronous mode	Internal clock/SCK as clock output*1
		Synchronous mode	Internal clock/SCK as serial clock out
1	0	Asynchronous mode	External clock/SC as clock input*2
		Synchronous mode	External clock/SC as serial clock inp
	1	Asynchronous mode	External clock/SC as clock input*2
		Synchronous mode	External clock/SC as serial clock inp

Notes: 1. Outputs a clock of the same frequen
2. Inputs a clock with a frequency 16 ti

Transmit End Interrupt Enable

0	Transmit end interrupt (TEI) request disabled
1	Transmit end interrupt (TEI) request enabled

Multiprocessor Interrupt Enable

0	Multiprocessor interrupts disabled (normal reception perfor [Clearing conditions] • When the MPIE bit is cleared to 0 • When MPB = 1 data is received
1	Multiprocessor interrupts enabled Receive interrupt (RXI) requests, receive error interrupt (ER requests, and setting of the RDRF, FER, and ORER flags i SSR are disabled until data with the multiprocessor bit set t is received

Receive Enable

0	Reception disabled
1	Reception enabled

Transmit Enable

0	Transmission disabled
1	Transmission enabled

Receive Interrupt Enable

0	Receive data full interrupt (RXI) request and receive error interrupt (ERI) request disabled
1	Receive data full interrupt (RXI) request and receive error interrupt (ERI) request enabled

Transmit Interrupt Enable

0	Transmit data empty interrupt (TXI) requests disabled
1	Transmit data empty interrupt (TXI) requests enabled

0	See SCI specification			
1	0	0	0	Opera input
1	0	0	1	Clock SCK
1	1	0	0	Fixed as SC
1	1	0	1	Clock SCK
1	1	1	0	Fixed as SC
1	1	1	1	Clock SCK

Transmit End Interrupt Enable

0	Transmit end interrupt (TEI) request disabled
1	Transmit end interrupt (TEI) request enabled

Multiprocessor Interrupt Enable

0	Multiprocessor interrupts disabled (normal reception perform [Clearing conditions] • When the MPIE bit is cleared to 0 • When MPB = 1 data is received
1	Multiprocessor interrupts enabled Receive interrupt (RXI) requests, receive error interrupt (ER requests, and setting of the RDRF, FER, and ORER flags in SSR are disabled until data with the multiprocessor bit set to is received

Receive Enable

0	Reception disabled
1	Reception enabled

Transmit Enable

0	Transmission disabled
1	Transmission enabled

Receive Interrupt Enable

0	Receive data full interrupt (RXI) request and receive error interrupt (ERI) request disabled
1	Receive data full interrupt (RXI) request and receive error interrupt (ERI) request enabled

Transmit Interrupt Enable

0	Transmit data empty interrupt (TXI) requests disabled
1	Transmit data empty interrupt (TXI) requests enabled

Stores data for serial transmission

Multiprocessor Bit

0	[Clearing condition] When data with a 0 multiprocessor bit is received
1	[Setting condition] When data with a 1 multiprocessor bit is received

Transmit End

0	[Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When the DMAC or DTC is activated by a TXI interrupt and wri
1	[Setting conditions] • When the TE bit in SCR is 0 • When TDRE = 1 at transmission of the last bit of a 1-byte serial transmit character

Parity Error

0	[Clearing condition] When 0 is written to PER after reading PER = 1
1	[Setting condition] When, in reception, the number of 1 bits in the receive data plus the parity does not match the parity setting (even or odd) specified by the O/$\overline{E}$ bit in

Framing Error

0	[Clearing condition] When 0 is written to FER after reading FER = 1
1	[Setting condition] When the SCI checks whether the stop bit at the end of the receive data is 1 when reception ends, and the stop bit is 0

Overrun Error

0	[Clearing condition] When 0 is written to ORER after reading ORER = 1
1	[Setting condition] When the next serial reception is completed while RDRF = 1

Receive Data Register Full

0	[Clearing conditions] • When 0 is written to RDRF after reading RDRF = 1 • When the DMAC or DTC is activated by an RXI interrupt and read data from RDR
1	[Setting condition] When serial reception ends normally and receive data is transferred from RSR to RDR

Transmit Data Register Empty

0	[Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When the DMAC or DTC is activated by a TXI interrupt and write data to TDR
1	[Setting conditions] • When the TE bit in SCR is 0 • When data is transferred from TDR to TSR and data can be written to TDR

Note: * Can only be written with 0 for flag clearing.

0	[Clearing condition] When data with a 0 multiprocessor bit is received
1	[Setting condition] When data with a 1 multiprocessor bit is received

Transmit End

0	[Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When the DMAC or DTC is activated by a TXI interrupt and w
1	[Setting conditions] • On reset, or in standby mode or module stop mode • When the TE bit in SCR is 0 and the ERS bit is also 0 • When TDRE = 1 and ERS = 0, 2.5 etu after a 1-byte serial transmit character is sent (normal transmission)

Note: etu: Elementary Time Unit (the time taken to transmit one bit)

Parity Error

0	[Clearing condition] When 0 is written to PER after reading PER = 1
1	[Setting condition] When, in reception, the number of 1 bits in the receive data plus the parity bi does not match the parity setting (even or odd) specified by the O/$\overline{E}$ bit in SM

Error Signal Status

0	[Clearing conditions] • On reset, or in standby mode or module stop mode • When 0 is written to ERS after reading ERS = 1
1	[Setting condition] When the error signal is sampled at the low level

Note: Clearing the TE bit in SCR to 0 does not affect the ERS flag, which retains its prior st

Overrun Error

0	[Clearing condition] When 0 is written to ORER after reading ORER = 1
1	[Setting condition] When the next serial reception is completed while RDRF = 1

Receive Data Register Full

0	[Clearing conditions] • When 0 is written to RDRF after reading RDRF = 1 • When the DMAC or DTC is activated by an RXI interrupt and read data from RDR
1	[Setting condition] When serial reception ends normally and receive data is transferred from RSR to RDR

Transmit Data Register Empty

0	[Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When the DMAC or DTC is activated by a TXI interrupt and write data to TDR
1	[Setting conditions] • When the TE bit in SCR is 0 • When data is transferred from TDR to TSR and data can be written to TDR

Note: * Can only be written with 0 for flag clearing.

Stores received serial data

SCMR0—Smart Card Mode Register 0 **H'FF7E** **SCI0, Smart Card**

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	SDIR	SINV	—	SMIF
Initial value	:	1	1	1	1	0	0	1	0
Read/Write	:	—	—	—	—	R/W	R/W	—	R/W

Smart Card Interface Mode Select

0	Smart Card interfac function is disabled
1	Smart Card interfac function is enabled

Smart Card Data Invert

0	TDR contents are transmitte they are Receive data is stored in RD
1	TDR contents are inverted b being transmitted Receive data is stored in RD in inverted form

Smart Card Data Direction

0	TDR contents are transmitted LSB-first Receive data is stored in RDR LSB-first
1	TDR contents are transmitted MSB-first Receive data is stored in RDR MSB-first

0	0	φ clock
	1	φ/4 clock
1	0	φ/16 clock
	1	φ/64 clock

Multiprocessor Mode

0	Multiprocessor function dis
1	Multiprocessor format sele

Stop Bit Length

0	1 stop bit
1	2 stop bits

Parity Mode

0	Even parity
1	Odd parity

Parity Enable

0	Parity bit addition and checking disabled
1	Parity bit addition and checking enabled

Character Length

0	8-bit data
1	7-bit data*

Note: * When 7-bit data is selected, the MSB (bit 7) of TDR is not transmi

Asynchronous Mode/Synchronous Mode Select

0	Asynchronous mode
1	Synchronous mode

0	0	φ clock
	1	φ/4 clock
1	0	φ/16 clock
	1	φ/64 clock

Multiprocessor Mode

0	Multiprocessor function disab
1	Multiprocessor format selecte

Stop Bit Length

0	1 stop bit
1	2 stop bits

Parity Mode

0	Even parity
1	Odd parity

Parity Enable

0	Parity bit addition and checking disabled
1	Parity bit addition and checking enabled

Character Length

0	8-bit data
1	7-bit data*

Note: * When 7-bit data is selected, the MSB (bit 7) of TDR is not transmitte

GSM Mode

0	Normal smart card interface mode operation • TEND flag generated 12.5 etu after beginning of start bit • Clock output on/off control only
1	GSM mode smart card interface mode operation • TEND flag generated 11.0 etu after beginning of start bit • Fixed high/low-level control possible (set in SCR) in addition to clock output on/off control

Note: etu (Elementary Time Unit): Interval for transfer of one bit

Note: See section 14.2.8, Bit Rate Register (BRR), for details.

		Synchronous mode	Internal clock/SCK pin as serial clock outp
	1	Asynchronous mode	Internal clock/SCK pin as clock output*1
		Synchronous mode	Internal clock/SCK pin as serial clock outp
1	0	Asynchronous mode	External clock/SCK pin as clock input*2
		Synchronous mode	External clock/SCK pin as serial clock input
	1	Asynchronous mode	External clock/SCK pin as clock input*2
		Synchronous mode	External clock/SCK pin as serial clock input

Notes: 1. Outputs a clock of the same frequenc
2. Inputs a clock with a frequency 16 tim

Transmit End Interrupt Enable

0	Transmit end interrupt (TEI) request disabled
1	Transmit end interrupt (TEI) request enabled

Multiprocessor Interrupt Enable

0	Multiprocessor interrupts disabled (normal reception performe [Clearing conditions] • When the MPIE bit is cleared to 0 • When MPB= 1 data is received
1	Multiprocessor interrupts enabled Receive interrupt (RXI) requests, receive error interrupt (ERI) requests, and setting of the RDRF, FER, and ORER flags in SSR are disabled until data with the multiprocessor bit set to is received

Receive Enable

0	Reception disabled
1	Reception enabled

Transmit Enable

0	Transmission disabled
1	Transmission enabled

Receive Interrupt Enable

0	Receive data full interrupt (RXI) request and receive error interrupt (ERI) request disabled
1	Receive data full interrupt (RXI) request and receive error interrupt (ERI) request enabled

Transmit Interrupt Enable

0	Transmit data empty interrupt (TXI) requests disabled
1	Transmit data empty interrupt (TXI) requests enabled

0	See SCI specificati			
1	0	0	0	Ope inpu
1	0	0	1	Cloc SCK
1	1	0	0	Fixe as S
1	1	0	1	Cloc SCK
1	1	1	0	Fixe as S
1	1	1	1	Cloc SCK

Transmit End Interrupt Enable

0	Transmit end interrupt (TEI) request disabled
1	Transmit end interrupt (TEI) request enabled

Multiprocessor Interrupt Enable

0	Multiprocessor interrupts disabled (normal reception perfo [Clearing conditions] • When the MPIE bit is cleared to 0 • When MPB= 1 data is received
1	Multiprocessor interrupts enabled Receive interrupt (RXI) requests, receive error interrupt (E requests, and setting of the RDRF, FER, and ORER flags SSR are disabled until data with the multiprocessor bit set is received

Receive Enable

0	Reception disabled
1	Reception enabled

Transmit Enable

0	Transmission disabled
1	Transmission enabled

Receive Interrupt Enable

0	Receive data full interrupt (RXI) request and receive error interrupt (ERI) request disabled
1	Receive data full interrupt (RXI) request and receive error interrupt (ERI) request enabled

Transmit Interrupt Enable

0	Transmit data empty interrupt (TXI) requests disabled
1	Transmit data empty interrupt (TXI) requests enabled

Stores data for serial transmission

Multiprocessor Bit

0	[Clearing condition] When data with a 0 multiprocessor bit is received
1	[Setting condition] When data with a 1 multiprocessor bit is received

Transmit End

0	[Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When the DMAC or DTC is activated by a TXI interrupt and
1	[Setting conditions] • When the TE bit in SCR is 0 • When TDRE = 1 at transmission of the last bit of a 1-byte serial transmit character

Parity Error

0	[Clearing condition] When 0 is written to PER after reading PER = 1
1	[Setting condition] When, in reception, the number of 1 bits in the receive data plus the parity does not match the parity setting (even or odd) specified by the O/$\overline{E}$ bit in S

Framing Error

0	[Clearing condition] When 0 is written to FER after reading FER = 1
1	[Setting condition] When the SCI checks whether the stop bit at the end of the receive data is 1 when reception ends, and the stop bit is 0

Overrun Error

0	[Clearing condition] When 0 is written to ORER after reading ORER = 1
1	[Setting condition] When the next serial reception is completed while RDRF = 1

Receive Data Register Full

0	[Clearing conditions] • When 0 is written to RDRF after reading RDRF = 1 • When the DMAC or DTC is activated by an RXI interrupt and read data from RDR
1	[Setting condition] When serial reception ends normally and receive data is transferred from RSR to RDR

Transmit Data Register Empty

0	[Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When the DMAC or DTC is activated by a TXI interrupt and write data to TDR
1	[Setting conditions] • When the TE bit in SCR is 0 • When data is transferred from TDR to TSR and data can be written to TDR

Note: * Can only be written with 0 for flag clearing.

0	[Clearing condition] When data with a 0 multiprocessor bit is received
1	[Setting condition] When data with a 1 multiprocessor bit is received

Transmit End

0	[Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When the DMAC or DTC is activated by a TXI interrupt and wri
1	[Setting conditions] • On reset, or in standby mode or module stop mode • When the TE bit in SCR is 0 and the ERS bit is also 0 • When TDRE = 1 and ERS = 0, 2.5 etu after a 1-byte serial transmit character is sent (normal transmission)

Note: etu: Elementary Time Unit (the time taken to transmit one bit)

Parity Error

0	[Clearing condition] When 0 is written to PER after reading PER = 1
1	[Setting condition] When, in reception, the number of 1 bits in the receive data plus the parity bit does not match the parity setting (even or odd) specified by the O/$\overline{E}$ bit in SMR

Error Signal Status

0	[Clearing conditions] • On reset, or in standby mode or module stop mode • When 0 is written to ERS after reading ERS =1
1	[Setting condition] When the error signal is sampled at the low level

Note: Clearing the TE bit in SCR to 0 does not affect the ERS flag, which retains its prior st

Overrun Error

0	[Clearing condition] When 0 is written to ORER after reading ORER = 1
1	[Setting condition] When the next serial reception is completed while RDRF = 1

Receive Data Register Full

0	[Clearing conditions] • When 0 is written to RDRF after reading RDRF = 1 • When the DMAC or DTC is activated by an RXI interrupt and read data from RDR
1	[Setting condition] When serial reception ends normally and receive data is transferred from RSR to RDR

Transmit Data Register Empty

0	[Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When the DMAC or DTC is activated by a TXI interrupt and write data to TDR
1	[Setting conditions] • When the TE bit in SCR is 0 • When data is transferred from TDR to TSR and data can be written to TDR

Note: * Can only be written with 0 for flag clearing.

Stores received serial data

SCMR1—Smart Card Mode Register 1 H'FF86 SCI1, Smart Card

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	SDIR	SINV	—	SMIF
Initial value	1	1	1	1	0	0	1	0
Read/Write	—	—	—	—	R/W	R/W	—	R/W

Smart Card Interface Mode Select

0	Smart Card interf function is disable
1	Smart Card interf function is enable

Smart Card Data Invert

0	TDR contents are transmi they are Receive data is stored in F
1	TDR contents are inverted being transmitted Receive data is stored in F in inverted form

Smart Card Data Direction

0	TDR contents are transmitted LSB-first Receive data is stored in RDR LSB-first
1	TDR contents are transmitted MSB-first Receive data is stored in RDR MSB-first

0	0	φ clock
	1	φ/4 clock
1	0	φ/16 clock
	1	φ/64 clock

Multiprocessor Mode

0	Multiprocessor function disa
1	Multiprocessor format selec

Stop Bit Length

0	1 stop bit
1	2 stop bits

Parity Mode

0	Even parity
1	Odd parity

Parity Enable

0	Parity bit addition and checking disabled
1	Parity bit addition and checking enabled

Character Length

0	8-bit data
1	7-bit data*

Note: * When 7-bit data is selected, the MSB (bit 7) of TDR is not transmitt

Asynchronous Mode/Synchronous Mode Select

0	Asynchronous mode
1	Synchronous mode

0	0	φ clock
	1	φ/4 clock
1	0	φ/16 clock
	1	φ/64 clock

Multiprocessor Mode

0	Multiprocessor function disa
1	Multiprocessor format selec

Stop Bit Length

0	1 stop bit
1	2 stop bits

Parity Mode

0	Even parity
1	Odd parity

Parity Enable

0	Parity bit addition and checking disabled
1	Parity bit addition and checking enabled

Character Length

0	8-bit data
1	7-bit data*

Note: * When 7-bit data is selected, the MSB (bit 7) of TDR is not transmitt

GSM Mode

0	Normal smart card interface mode operation • TEND flag generated 12.5 etu after beginning of start bit • Clock output on/off control only
1	GSM mode smart card interface mode operation • TEND flag generated 11.0 etu after beginning of start bit • Fixed high/low-level control possible (set in SCR) in addition to clock output on/off control

Note: etu (Elementary Time Unit): Interval for transfer of one bit

Note: See section 14.2.8, Bit Rate Register (BRR), for details.

		mode	as serial clock out
	1	Asynchronous mode	Internal clock/SCK as clock output*1
		Synchronous mode	Internal clock/SCK as serial clock out
1	0	Asynchronous mode	External clock/SC as clock input*2
		Synchronous mode	External clock/SC as serial clock inp
	1	Asynchronous mode	External clock/SC as clock input*2
		Synchronous mode	External clock/SC as serial clock inp

Notes: 1. Outputs a clock of the same freque
2. Inputs a clock with a frequency 16 t

Transmit End Interrupt Enable

0	Transmit end interrupt (TEI) request disabled
1	Transmit end interrupt (TEI) request enabled

Multiprocessor Interrupt Enable

0	Multiprocessor interrupts disabled (normal reception perform [Clearing conditions] • When the MPIE bit is cleared to 0 • When MPB= 1 data is received
1	Multiprocessor interrupts enabled Receive interrupt (RXI) requests, receive error interrupt (ER requests, and setting of the RDRF, FER, and ORER flags in SSR are disabled until data with the multiprocessor bit set t is received

Receive Enable

0	Reception disabled
1	Reception enabled

Transmit Enable

0	Transmission disabled
1	Transmission enabled

Receive Interrupt Enable

0	Receive data full interrupt (RXI) request and receive error interrupt (ERI) request disabled
1	Receive data full interrupt (RXI) request and receive error interrupt (ERI) request enabled

Transmit Interrupt Enable

0	Transmit data empty interrupt (TXI) requests disabled
1	Transmit data empty interrupt (TXI) requests enabled

0	See SCI specificatio			
1	0	0	0	Oper input
1	0	0	1	Cloc SCK
1	1	0	0	Fixe as S
1	1	0	1	Cloc SCK
1	1	1	0	Fixe as S
1	1	1	1	Cloc SCK

Transmit End Interrupt Enable

0	Transmit end interrupt (TEI) request disabled
1	Transmit end interrupt (TEI) request enabled

Multiprocessor Interrupt Enable

0	Multiprocessor interrupts disabled (normal reception perfor [Clearing conditions] • When the MPIE bit is cleared to 0 • When MPB= 1 data is received
1	Multiprocessor interrupts enabled Receive interrupt (RXI) requests, receive error interrupt (E requests, and setting of the RDRF, FER, and ORER flags SSR are disabled until data with the multiprocessor bit set is received

Receive Enable

0	Reception disabled
1	Reception enabled

Transmit Enable

0	Transmission disabled
1	Transmission enabled

Receive Interrupt Enable

0	Receive data full interrupt (RXI) request and receive error interrupt (ERI) request disabled
1	Receive data full interrupt (RXI) request and receive error interrupt (ERI) request enabled

Transmit Interrupt Enable

0	Transmit data empty interrupt (TXI) requests disabled
1	Transmit data empty interrupt (TXI) requests enabled

Stores data for serial transmission

Multiprocessor Bit

0	[Clearing condition] When data with a 0 multiprocessor bit is received
1	[Setting condition] When data with a 1 multiprocessor bit is received

Transmit End

0	[Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When the DMAC or DTC is activated by a TXI interrupt and w
1	[Setting conditions] • When the TE bit in SCR is 0 • When TDRE = 1 at transmission of the last bit of a 1-byte serial transmit character

Parity Error

0	[Clearing condition] When 0 is written to PER after reading PER = 1
1	[Setting condition] When, in reception, the number of 1 bits in the receive data plus the parity bit does not match the parity setting (even or odd) specified by the O/$\overline{E}$ bit in SM

Framing Error

0	[Clearing condition] When 0 is written to PER after reading PER = 1
1	[Setting condition] When the SCI checks whether the stop bit at the end of the receive data is 1 when reception ends, and the stop bit is 0

Overrun Error

0	[Clearing condition] When 0 is written to ORER after reading ORER = 1
1	[Setting condition] When the next serial reception is completed while RDRF = 1

Receive Data Register Full

0	[Clearing conditions] • When 0 is written to RDRF after reading RDRF = 1 • When the DMAC or DTC is activated by an RXI interrupt and read data from RDR
1	[Setting condition] When serial reception ends normally and receive data is transferred from RSR to RDR

Transmit Data Register Empty

0	[Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When the DMAC or DTC is activated by a TXI interrupt and write data to TDR
1	[Setting conditions] • When the TE bit in SCR is 0 • When data is transferred from TDR to TSR and data can be written to TDR

Note: * Can only be written with 0 for flag clearing.

0	[Clearing condition] When data with a 0 multiprocessor bit is received
1	[Setting condition] When data with a 1 multiprocessor bit is received

Transmit End

0	[Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When the DMAC or DTC is activated by a TXI interrupt and w
1	[Setting conditions] • On reset, or in standby mode or module stop mode • When the TE bit in SCR is 0 and the ERS bit is also 0 • When TDRE = 1 and ERS = 0, 2.5 etu after a 1-byte serial transmit character is sent (normal transmission)

Note: etu: Elementary Time Unit (the time taken to transmit one bit)

Parity Error

0	[Clearing condition] When 0 is written to PER after reading PER = 1
1	[Setting condition] When, in reception, the number of 1 bits in the receive data plus the parity b does not match the parity setting (even or odd) specified by the O/$\overline{E}$ bit in S

Error Signal Status

0	[Clearing conditions] • On reset, or in standby mode or module stop mode • When 0 is written to ERS after reading ERS =1
1	[Setting condition] When the error signal is sampled at the low level

Note: Clearing the TE bit in SCR to 0 does not affect the ERS flag, which retains its prior s

Overrun Error

0	[Clearing condition] When 0 is written to ORER after reading ORER = 1
1	[Setting condition] When the next serial reception completed while RDRF = 1

Receive Data Register Full

0	[Clearing conditions] • When 0 is written to RDRF after reading RDRF = 1 • When the DMAC or DTC is activated by an RXI interrupt and read data from RDR
1	[Setting condition] When serial reception ends normally and receive data is transferred from RSR to RDR

Transmit Data Register Empty

0	[Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When the DMAC or DTC is activated by a TXI interrupt and write data to TDR
1	[Setting conditions] • When the TE bit in SCR is 0 • When data is transferred from TDR to TSR and data can be written to TDR

Note: * Can only be written with 0 for flag clearing.

Stores received serial data

SCMR2—Smart Card Mode Register 2 H'FF8E SCI2, Smart Card

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	SDIR	SINV	—	SMIF
Initial value	:	1	1	1	1	0	0	1	0
Read/Write	:	—	—	—	—	R/W	R/W	—	R/W

Smart Card Interface Mode Select

0	Smart Card interface function is disabled
1	Smart Card interface function is enabled

Smart Card Data Invert

0	TDR contents are transmitted as Receive data is stored in RDR a
1	TDR contents are inverted befor transmitted Receive data is stored in RDR in inverted form

Smart Card Data Direction

0	TDR contents are transmitted LSB-first Receive data is stored in RDR LSB-first
1	TDR contents are transmitted MSB-first Receive data is stored in RDR MSB-first

ADDREH—A/D Data Register EH **H'FF98** **A/D**
ADDREL—A/D Data Register EL **H'FF99** **A/D**
ADDRFH—A/D Data Register FH **H'FF9A** **A/D**
ADDRFL—A/D Data Register FL **H'FF9B** **A/D**
ADDRGH—A/D Data Register GH **H'FF9C** **A/D**
ADDRGL—A/D Data Register GL **H'FF9D** **A/D**
ADDRHH—A/D Data Register HH **H'FF9E** **A/D**
ADDRHL—A/D Data Register HL **H'FF9F** **A/D**

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
		—	—	—	—	—	—	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2
Initial value	:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Read/Write	:	—	—	—	—	—	—	R	R	R	R	R	R	R	R

Stores the results of A/D conversi

Analog Input Channel	A/D Data Register
AN_0	ADDRA*
AN_1	ADDRB*
AN_2	ADDRC*
AN_3	ADDRD*
AN_4	ADDRE
AN_5	ADDRF
AN_6	ADDRG
AN_7	ADDRH

Note: * Except when buffer operation is used.

CH2	CH1	CH0	(GRP = 0)	(G
0	0	0	AN_0	AI
		1	AN_1	AI
	1	0	AN_2	AI
		1	AN_3	AI
1	0	0	AN_4	AI
		1	AN_5	AI
	1	0	AN_6	AI
		1	AN_7	AI

Group Mode

0	Select mode
1	Group mode

Clock Select

0	Conversion time = 20 states (A/D converter reference
1	Conversion time = 40 states (when ϕ/2 is selected)

A/D Start

0	A/D conversion stopped
1	• Single mode: A/D conversion is started. Cleared to 0 automati conversion on the specified channel ends • Scan mode: A/D conversion is started. Conversion continues u is cleared to 0 by software

A/D Interrupt Enable

0	A/D conversion end interrupt (ADI) request disabled
1	A/D conversion end interrupt (ADI) request enabled

A/D End Flag

0	[Clearing conditions] • When 0 is written to the ADF flag after reading ADF = 1 • When the DMAC or DTC is activated by an ADI interrupt, and the relevant register is
1	[Setting conditions] • Single mode: When conversion ends for all specified channels, and A/D conversion • Scan mode: When one round of conversion has been performed on all specified cha

Note: * Can only be written with 0 for flag clearing.

	1	ADDRA and ADDRB ar buffer operation (Conversion result → A ADDRB) ADDRB is buffer registe
1	0	ADDRA and ADDRC, a ADDRD, are used for b (Conversion result 1 → ADDRC; conversion re ADDRB → ADDRD) ADDRC and ADDRD ar
	1	ADDRA to ADDRD are operation (Conversion result → A ADDRB → ADDRC → A ADDRB to ADDRD are

Simultaneous Sampling

0	Normal sampling operation
1	Simultaneous sampling operation

Scan Mode

0	Single mode
1	Scan mode

Timer Trigger Select

0	0	A/D conversion start by software is enabled
	1	A/D conversion start by TPU conversion start trigger is enabled
1	0	A/D conversion start by 8-bit timer conversion start trigger is ena
	1	A/D conversion start by external trigger pin ($\overline{\text{ADTRG}}$) is enabled

Specifies conversion start mode

0	Low-power conversion mode
1	High-speed start mode

Stores data for D/A conversion

D/A Output Enable 0

0	Analog output DA_0 is disabled
1	Channel 0 D/A conversion is enabled Analog output DA_0 is enabled

D/A Output Enable 1

0	Analog output DA_1 is disabled
1	Channel 1 D/A conversion is enabled Analog output DA_1 is enabled

D/A Conversion Control

0	0	*	Channel 0 and 1 D/A conversion disabled
	1	0	Channel 0 D/A conversion enabled
			Channel 1 D/A conversion disabled
		1	Channel 0 and 1 D/A conversions enabled
1	0	0	Channel 0 D/A conversion disabled
			Channel 1 D/A conversion enabled
		1	Channel 0 and 1 D/A conversion enabled
	1	*	Channel 0 and 1 D/A conversion enabled

Legend: *: Don't care

0	0	0	Clock input disabled
		1	Internal clock: counted at fal of φ/8
	1	0	Internal clock: counted at fal of φ/64
		1	Internal clock: counted at fal of φ/8192
1	0	0	For channel 0: Count at TCNT1 overflow sig For channel 1: Count at TCNT0 compare m
		1	External clock: counted at ris
	1	0	External clock: counted at fa
		1	External clock: counted at bo falling edges

Note: * If the count input of channel 0 is the TC signal and that of channel 1 is the TCN match signal, no incrementing clock is Do not use this setting.

Counter Clear

0	0	Clear is disabled
	1	Clear by compare match A
1	0	Clear by compare match B
	1	Clear by rising edge of external reset input

Timer Overflow Interrupt Enable

0	OVF interrupt requests (OVI) are disabled
1	OVF interrupt requests (OVI) are enabled

Compare Match Interrupt Enable A

0	CMFA interrupt requests (CMIA) are disabled
1	CMFA interrupt requests (CMIA) are enabled

Compare Match Interrupt Enable B

0	CMFB interrupt requests (CMIB) are disabled
1	CMFB interrupt requests (CMIB) are enabled

	7	6	5	4	3	2	1	0
Initial value :	0	0	0	1	0	0	0	0
Read/Write :	R/(W)*	R/(W)*	R/(W)*	—	R/W	R/W	R/W	R/W

Output Select

0	0	No change when com occurs
	1	0 is output when comp occurs
1	0	1 is output when comp occurs
	1	Output is inverted whe match A occurs (toggl

Output Select

0	0	No change when compare match B
	1	0 is output when compare match B
1	0	1 is output when compare match B
	1	Output is inverted when compare m (toggle output)

A/D Trigger Enable (TCSR0 only)

0	A/D converter start requests by compare match A
1	A/D converter start requests by compare match A

Timer Overflow Flag

0	[Clearing condition] Cleared by reading OVF when OVF = 1, then writing 0 to OV
1	[Setting condition] Set when TCNT overflows (changes from H'FF to H'00)

Compare Match Flag A

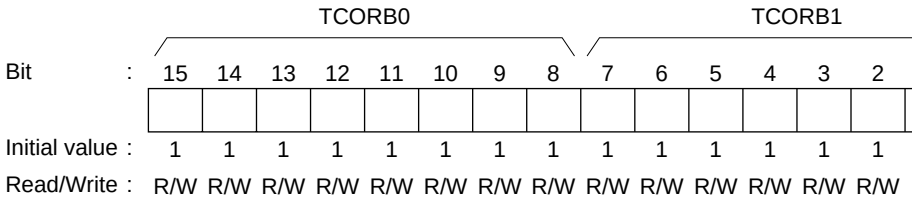
0	[Clearing conditions] • Cleared by reading CMFA when CMFA = 1, then writing 0 to CMFA • When the DTC is activated by a CMIA interrupt, while DISEL bit of MRB in
1	[Setting condition] Set when TCNT matches TCORA

Compare Match Flag B

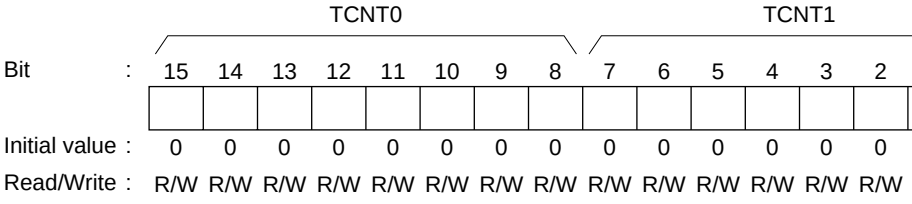
0	[Clearing conditions] • Cleared by reading CMFB when CMFB = 1, then writing 0 to CMFB • When the DTC is activated by a CMIB interrupt, while DISEL bit of MRB in DTC is 0.
1	[Setting condition] Set when TCNT matches TCORB

Note: * Only 0 can be written to bits 7 to 5, to clear these flags.

TCORB0—Time Constant Register B0 **H'FFB6** **8-Bit Timer**
TCORB1—Time Constant Register B1 **H'FFB7** **8-Bit Timer**



TCNT0—Timer Counter 0 **H'FFB8** **8-Bit Timer**
TCNT1—Timer Counter 1 **H'FFB9** **8-Bit Timer**



				(when ϕ =
0	0	0	ϕ/2 (initial value)	25.6 µs
		1	ϕ/64	819.2 µs
	1	0	ϕ/128	1.6 ms
		1	ϕ/512	6.6 ms
1	0	0	ϕ/2048	26.2 ms
		1	ϕ/8192	104.9 ms
	1	0	ϕ/32768	419.4 ms
		1	ϕ/131072	1.68 s

Note: * The overflow period is the time from
starts counting up from H'00 until ov

Timer Enable

0	TCNT is initialized to H'00 and halted
1	TCNT counts

Timer Mode Select

0	Interval timer mode: Sends the CPU an interval timer interrupt reques (WOVI) when TCNT overflows
1	Watchdog timer mode: Generates the $\overline{\text{WDTOVF}}$ signal when TCNT overflows

Overflow Flag

0	[Clearing condition] Cleared by reading TCSR when OVF = 1, then writing 0 to OVF
1	[Setting condition] Set when TCNT overflows from H'FF to H'00 in interval timer mode

The method for writing to TCSR is different from that for general registers to prevent inadvertent ove
For details see section 13.2.4, Notes on Register Access.

Note: * Can only be written with 0 for flag clearing.

overwriting. For details see section 15.2.4, Notes on Register Access.

Reset Select

0	Power-on reset
1	Manual reset

Reset Enable

0	Reset signal is not generated if TCNT overflows*
1	Reset signal is generated if TCNT overflows

Note: * The modules H8S/2655 Series are not reset, but
and TCSR in WDT are reset.

Watchdog Timer Overflow Flag

0	[Clearing condition] Cleared by reading TCSR when WOVF = 1, then writing 0 to W
1	[Setting condition] Set when TCNT overflows (changed from H'FF to H'00) during watchdog timer operation

Note: * Can only be written with 0 for flag clearing.

The method for writing to RSTCSR is different from that for general registers to pr
inadvertent overwriting. For details see section 13.2.4, Notes on Register Access

Counter Start

0	$TCNT_n$ count operation is
1	$TCNT_n$ performs count op

Note: n = 5 to 0

Note: If 0 is written to the CST bit during operation with the TIOC pin designated for outp the counter stops but the TIOC pin output compare output level is retained. If TIOR written to when the CST bit is cleared to 0, the pin output level will be changed to t set initial output value.

TSYR—Timer Synchro Register H'FFC1

Bit	:	7	6	5	4	3	2	1	0
		—	—	SYNC5	SYNC4	SYNC3	SYNC2	SYNC1	SYN
Initial value	:	0	0	0	0	0	0	0	0
Read/Write	:	—	—	R/W	R/W	R/W	R/W	R/W	R/W

Timer Synchronization

0	$TCNT_n$ operates independently (TCNT presetting clearing is unrelated to other channels)
1	$TCNT_n$ performs synchronous operation TCNT synchronous presetting/synchronous clea is possible

Note: n = 5 to 0

Notes: 1. To set synchronous operation, the SYNC bits for at least two channels must be To set synchronous clearing, in addition to the SYNC bit , the TCNT clearing
2. source must also be set by means of bits CCLR2 to CCLR0 in TCR.

		1	Internal clock: counts on φ/4
	1	0	Internal clock: counts on φ/16
		1	Internal clock: counts on φ/64
1	0	0	External clock: counts on TCI
		1	External clock: counts on TCI
	1	0	External clock: counts on TCI
		1	External clock: counts on TCI

Clock Edge

0	0	Count at rising edge
	1	Count at falling edge
1	—	Count at both edges

Counter Clear

0	0	0	TCNT clearing disabled
		1	TCNT cleared by TGRA compare match/input capture
	1	0	TCNT cleared by TGRB compare match/input capture
		1	TCNT cleared by counter clearing for another channel performing synchronous clearing/synchronous operation*1
1	0	0	TCNT clearing disabled
		1	TCNT cleared by TGRC compare match/input capture*2
	1	0	TCNT cleared by TGRD compare match/input capture*2
		1	TCNT cleared by counter clearing for another channel performing synchronous clearing/synchronous operation*1

Notes: 1. Synchronous operation setting is performed by setting the SYNC bit in TSYR to 1.
2. When TGRC or TGRD is used as a buffer register, TCNT is not cleared because the buffer register setting has priority, and compare match/input capture does not occur.

Mode

0	0	0	0	Normal oper
			1	Reserved
		1	0	PWM mode
			1	PWM mode
	1	0	0	Phase count
			1	Phase count
		1	0	Phase count
			1	Phase count
1	*	*	*	—

Legend: *: Don't care

Notes: 1. MD3 is a reserved bi
it should always be w
2. Phase counting mod
set for channels 0 an
case, 0 should alway
to MD2.

TGRA Buffer Operation

0	TGRA operates normally
1	TGRA and TGRC used together for buffer operation

TGRB Buffer Operation

0	TGRB operates normally
1	TGRB and TGRD used together for buffer operation

			1	compare register	Initial output is 0 output	0 output at compare
		1	0			1 output at compare
			1			Toggle output at com
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compare
		1	0			1 output at compare
			1			Toggle output at com
1	0	0	0	TGR0A is input capture register	Capture input source is TIOCA0 pin	Input capture at risin
			1			Input capture at fallin
		1	*			Input capture at both
	1	*	*		Capture input source is channel 1/count clock	Input capture at TCN count-down

Legend: *: Don't care

TGR0B I/O Control

0	0	0	0	TGR0B is output compare register	Output disabled	
			1		Initial output is 0 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
	1	0	0		Output disabled	
			1		Initial output is 0 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
1	0	0	0	TGR0B is input capture register	Capture input source is TIOCB0 pin	Input capture at rising edge
			1			Input capture at falling edge
		1	*			Input capture at both edges
	1	*	*		Capture input source is channel 1/count clock	Input capture at TCNT1 count-up/ count-down*1

Legend: *: Don't care

Note: 1. When bits TPSC2 to TPSC0 in TCR1 are set to B'000, and φ/1 is used as the TCNT1 count clock, this set input capture is not generated.

			1	compare register	0 output	
		1	0			1 output at compare mat
			1			Toggle output at compar
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compare mat
		1	0			1 output at compare mat
			1			Toggle output at compar
1	0	0	0	TGR0C is input capture register	Capture input source is TIOCC0 pin	Input capture at rising ed
			1			Input capture at falling ed
		1	*			Input capture at both edg
	1	*	*		Capture input source is channel 1/count clock	Input capture at TCNT1 c count-down

Legend: *: Don't care

Note: When the BFA bit in TMDR0 is set to 1 and TGR0C is used as a bu this setting is invalid and input capture/output compare is not gener

TGR0D I/O Control

0	0	0	0	TGR0D is output compare register	Output disabled	
			1		Initial output is 0 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
1	0	0	0	TGR0D is input capture registe*2	Capture input source is TIOCD0 pin	Input capture at rising edge
			1			Input capture at falling edge
		1	*			Input capture at both edges
	1	*	*		Capture input source is channel 1/count clock	Input capture at TCNT1 count-up/ count-down*1

Legend: *: Don't care

Note: 1. When bits TPSC2 to TPSC0 in TCR1 are set to B'000, and ϕ/1 is used as the TCNT1 count clock, this setting is invalid and input capture is not generated.
2. When the BFB bit in TMDR0 is set to 1 and TGR0D is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Note: When GRC or GRD is designated for buffer operation, this setting is invalid and the register operates as a buffer r

0	Interrupt req by TGFA bi
1	Interrupt req by TGFA bi

TGR Interrupt Enable

0	Interrupt reque by TGFB bit di
1	Interrupt reque by TGFB bit en

TGR Interrupt Enable C

0	Interrupt requests (TGIC) TGFC bit disabled
1	Interrupt requests (TGIC) TGFC bit enabled

TGR Interrupt Enable D

0	Interrupt requests (TGID) by TGFD bit disabled
1	Interrupt requests (TGID) by TGFD bit enabled

Overflow Interrupt Enable

0	Interrupt requests (TCIV) by TCFV disabled
1	Interrupt requests (TCIV) by TCFV enabled

A/D Conversion Start Request Enable

0	A/D conversion start request generation disabled
1	A/D conversion start request generation enabled

	• When DMAC is activated by TGIA inte while DTA bit of DMABCR in DMAC is • When 0 is written to TGFA after readi
1	[Setting conditions] • When TCNT = TGRA while TGRA is f as output compare register • When TCNT value is transferred to TC input capture signal while TGRA is fun as input capture register

TGR Input Capture/Output Compare Flag B

0	[Clearing conditions] • When DTC is activated by TGIB interrupt while D of MRB in DTC is 0 • When 0 is written to TGFB after reading TGFB =
1	[Setting conditions] • When TCNT = TGRB while TGRB is functioning output compare register • When TCNT value is transferred to TGRB by inp capture signal while TGRB is functioning as inpu register

TGR Input Capture/Output Compare Flag C

0	[Clearing conditions] • When DTC is activated by TGIC interrupt while DISEL bit of M • When 0 is written to TGFC after reading TGFC = 1
1	[Setting conditions] • When TCNT = TGRC while TGRC is functioning as output co • When TCNT value is transferred to TGRC by input capture sig TGRC is functioning as input capture register

TGR Input Capture/Output Compare Flag D

0	[Clearing conditions] • When DTC is activated by TGID interrupt while DISEL bit of MRB in • When 0 is written to TGFD after reading TGFD = 1
1	[Setting conditions] • When TCNT = TGRD while TGRD is functioning as output compare • When TCNT value is transferred to TGRD by input capture signal wh functioning as input capture register

Overflow Flag

0	[Clearing condition] When 0 is written to TCFV after reading TCFV = 1
1	[Setting condition] When the TCNT value overflows (changes from H'FFFF to H'0000)

Note: * Can only be written with 0 for flag clearing.

Up-counter

TGR0A—Timer General Register 0A	**H'FFD8**
TGR0B—Timer General Register 0B	**H'FFDA**
TGR0C—Timer General Register 0C	**H'FFDC**
TGR0D—Timer General Register 0D	**H'FFDE**

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
Initial value	:	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Read/Write	:	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

		1	Internal clock: counts on φ/4
	1	0	Internal clock: counts on φ/16
		1	Internal clock: counts on φ/64
1	0	0	External clock: counts on TC
		1	External clock: counts on TC
	1	0	Internal clock: counts on φ/25
		1	Counts on TCNT2 overflow/u

Note: This setting is ignored when channe counting mode.

Clock Edge

0	0	Count at rising edge
	1	Count at falling edge
1	*	Count at both edges

Legend: *: Don't care

Note: This setting is ignored when channel 1 is in phase counting mode.

Counter Clear

0	0	TCNT clearing disabled
	1	TCNT cleared by TGRA compare match/input capture
1	0	TCNT cleared by TGRB compare match/input capture
	1	TCNT cleared by counter clearing for another channel performing synchronous clearing/synchronous operation*

Note: * Synchronous operating setting is performed by setting the SYNC bit in TSYR to 1.

0	0	0	0	Normal ope
			1	Reserved
		1	0	PWM mode
			1	PWM mode
	1	0	0	Phase cou
			1	Phase cou
		1	0	Phase cou
			1	1Phase co
1	*	*	*	—

Legend: *: Don't care

Note: MD3 is a reserved bit. I
it should always be writ

				1	compare register	Initial output is 0 output	0 output at compare m
			1	0			1 output at compare m
				1			Toggle output at com
		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at compare m
			1	0			1 output at compare m
				1			Toggle output at com
1	0	0	0	0	TGR1A is input capture register	Capture input source is TIOCA1 pin	Input capture at rising
				1			Input capture at fallin
			1	*			Input capture at both
		1	*	*		Capture input source is TGR0A compare match/ input capture	Input capture at gene channel 0/TGR0A co input capture

Legend: *: Don't care

TGR1B I/O Control

0	0	0	0	TGR1B is output compare register	Output disabled	
			1		Initial output is 0 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
1	0	0	0	TGR1B is input capture register	Capture input source is TIOCB1 pin	Input capture at rising edge
			1			Input capture at falling edge
		1	*			Input capture at both edges
	1	*	*		Capture input source is TGR0C compare match/ input capture	Input capture at generation of TGR0B compare match/input capture

Legend: *: Don't care

	by TGFA bit d
1	Interrupt requ by TGFA bit e

TGR Interrupt Enable B

0	Interrupt requests by TGFB bit disal
1	Interrupt requests by TGFB bit enab

Overflow Interrupt Enable

0	Interrupt requests (TCIV) by TCFV disabled
1	Interrupt requests (TCIV) by TCFV enabled

Underflow Interrupt Enable

0	Interrupt requests (TCIU) by TCFU disabled
1	Interrupt requests (TCIU) by TCFU enabled

A/D Conversion Start Request Enable

0	A/D conversion start request generation disabled
1	A/D conversion start request generation enabled

	• When DMAC is activated by TGIA while DTA bit of DMABCR in DMA • When 0 is written to TGFA after re TGFA = 1
1	[Setting conditions] • When TCNT = TGRA while TGRA as output compare register • When TCNT value is transferred to input capture signal while TGRA is as input capture register

TGR Input Capture/Output Compare Flag B

0	[Clearing conditions] • When DTC is activated by TGIB interrupt bit of MRB in DTC is 0 • When 0 is written to TGFB after reading T
1	[Setting conditions] • When TCNT = TGRB while TGRB is funct output compare register • When TCNT value is transferred to TGRB capture signal while TGRB is functioning a capture register

Overflow Flag

0	[Clearing condition] When 0 is written to TCFV after reading TCFV = 1
1	[Setting condition] When the TCNT value overflows (changes from H'FFFF to H'0000)

Underflow Flag

0	[Clearing condition] When 0 is written to TCFU after reading TCFU = 1
1	[Setting condition] When the TCNT value underflows (changes from H'0000 to H'FFFF)

Count Direction Flag

0	TCNT counts down
1	TCNT counts up

Note: * Can only be written with 0 for flag clearing.

Up/down-counter*

Note: * This timer counter can be used as an up/down-counter only in phase counti
mode or when performing overflow/underflow counting on another channel.
In other cases it functions as an up-counter.

TGR1A—Timer General Register 1A **H'FFE8**
TGR1B—Timer General Register 1B **H'FFEA**

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
Initial value	:	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Read/Write	:	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

		1	Internal clock: counts on φ/4
	1	0	Internal clock: counts on φ/16
		1	Internal clock: counts on φ/64
1	0	0	External clock: counts on TCL
		1	External clock: counts on TCL
	1	0	External clock: counts on TCL
		1	Internal clock: counts on φ/10

Note: This setting is ignored when channel counting mode.

Clock Edge

0	0	Count at rising edge
	1	Count at falling edge
1	*	Count at both edges

Legend: *: Don't care

Note: This setting is ignored when channel 2 is in phase counting mode.

Counter Clear

0	0	TCNT clearing disabled
	1	TCNT cleared by TGRA compare match/input capture
1	0	TCNT cleared by TGRB compare match/input capture
	1	TCNT cleared by counter clearing for another channel performing synchronous clearing/synchronous operation*

Note: * Synchronous operating setting is performed by setting the SYNC bit TSYR to 1.

0	0	0	0	Normal op
			1	Reserved
		1	0	PWM mod
			1	PWM mod
	1	0	0	Phase cou
			1	Phase cou
		1	0	Phase cou
			1	Phase cou
1	*	*	*	—

Legend: *: Don't care

Note: MD3 is a reserved bit.
it should always be wr

			1	TGR2A is output compare register	Initial output is 0 output	0 output at compare	
		1	0			1 output at compare	
			1			Toggle output at com	
	1	0	0		Output disabled		
			1		Initial output is 1 output	0 output at compare	
		1	0			1 output at compare	
			1			Toggle output at com	
1	*	0	0	TGR2A is input capture register	Capture input source is TIOCA2 pin	Input capture at rising	
			1			Input capture at fallin	
		1	*			Input capture at both	

Legend: *: Don't care

TGR2B I/O Control

0	0	0	0	TGR2B is output compare register	Output disabled	
			1		Initial output is 0 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
1	*	0	0	TGR2B is input capture register	Capture input source is TIOCB2 pin	Input capture at rising edge
			1			Input capture at falling edge
		1	*			Input capture at both edges

Legend: *: Don't care

	by TGFA bit d
1	Interrupt requ by TGFA bit e

TGR Interrupt Enable B

0	Interrupt requests by TGFB bit disab
1	Interrupt requests by TGFB bit enab

Overflow Interrupt Enable

0	Interrupt requests (TCIV) by TCFV disabled
1	Interrupt requests (TCIV) by TCFV enabled

Underflow Interrupt Enable

0	Interrupt requests (TCIU) by TCFU disabled
1	Interrupt requests (TCIU) by TCFU enabled

A/D Conversion Start Request Enable

0	A/D conversion start request generation disabled
1	A/D conversion start request generation enabled

	while DISEL bit of MRB in DTC • When DMAC is activated by TC while DTA bit of DMABCR in D • When 0 is written to TGFA afte TGFA = 1
1	[Setting conditions] • When TCNT = TGRA while TG functioning as output compare • When TCNT value is transferre input capture signal while TGR as input capture register

TGR Input Capture/Output Compare Flag B

0	[Clearing conditions] • When DTC is activated by TGIB interru bit of MRB in DTC is 0 • When 0 is written to TGFB after readin TGFB = 1
1	[Setting conditions] • When TCNT = TGRB while TGRB is fu output compare register • When TCNT value is transferred to TG capture signal while TGRB is functionin capture register

Overflow Flag

0	[Clearing condition] When 0 is written to TCFV after reading TCFV = 1
1	[Setting condition] When the TCNT value overflows (changes from H'FFFF to H'0000)

Underflow Flag

0	[Clearing condition] When 0 is written to TCFU after reading TCFU = 1
1	[Setting condition] When the TCNT value underflows (changes from H'0000 to H'FFFF)

Count Direction Flag

0	TCNT counts down
1	TCNT counts up

Note: * Can only be written with 0 for flag clearing.

Up/down-counter*

Note: * This timer counter can be used as an up/down-counter only in phase count mode or when performing overflow/underflow counting on another channel. In other cases it functions as an up-counter.

TGR2A—Timer General Register 2A **H'FFF8**
TGR2B—Timer General Register 2B **H'FFFA**

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
Initial value	:	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Read/Write	:	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

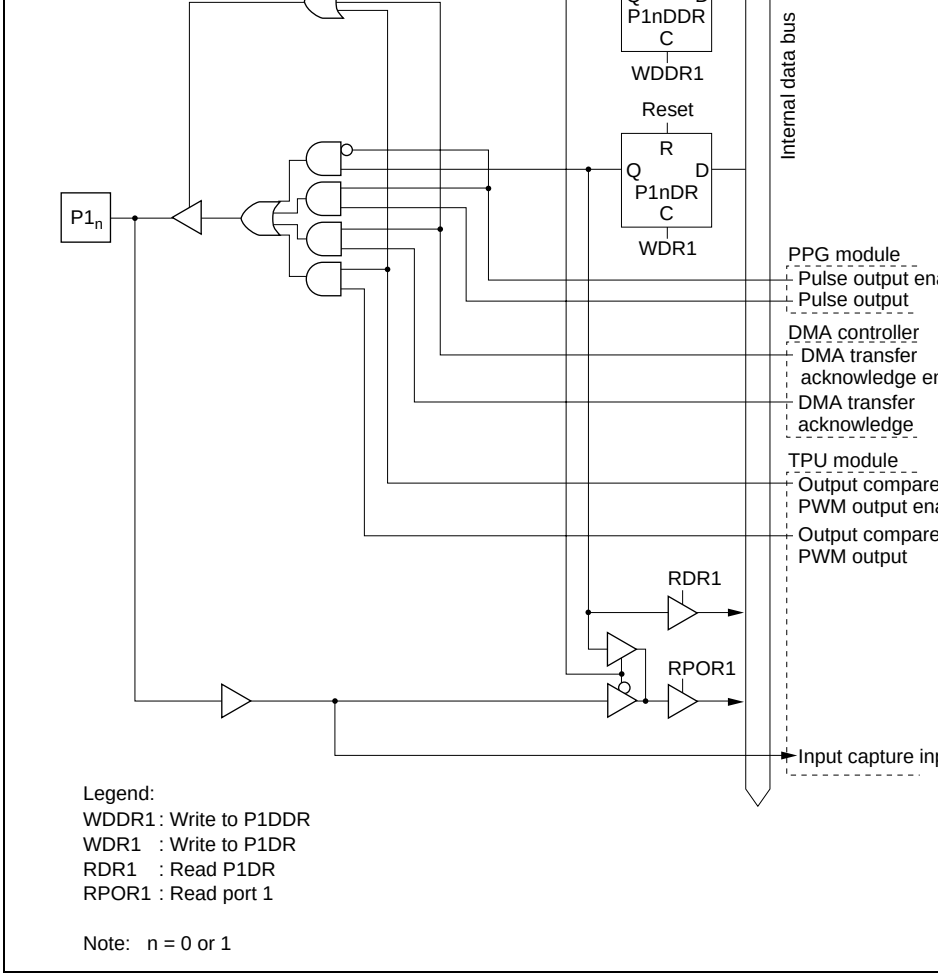


Figure C.1 (a) Port 1 Block Diagram (Pins $P1_0$ and $P1_1$)

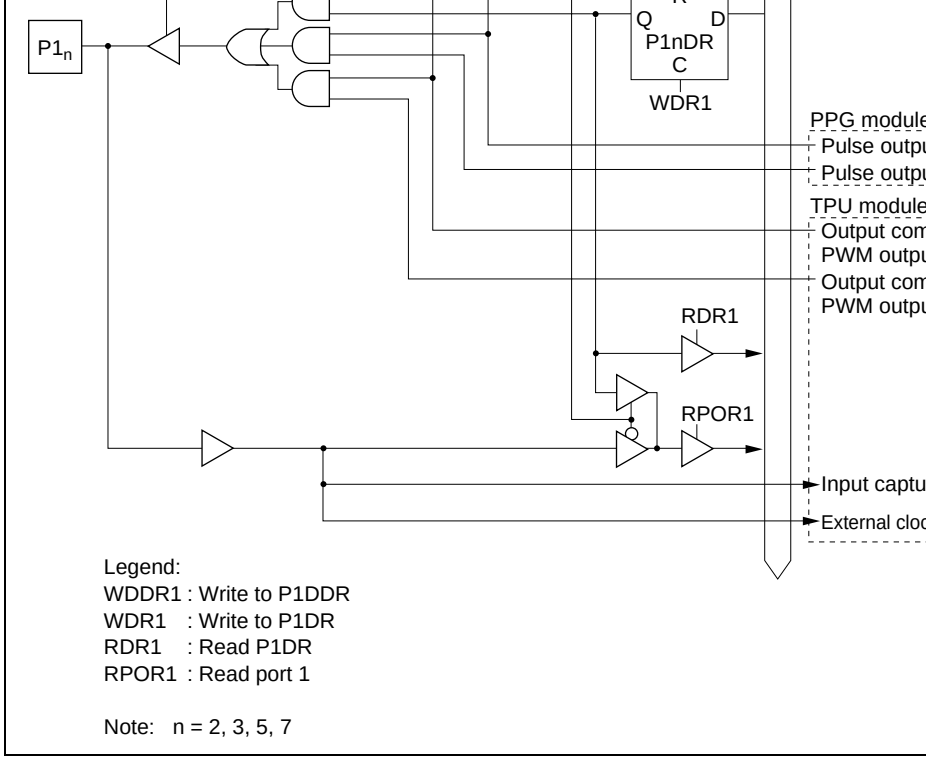


Figure C.1 (b) Port 1 Block Diagram (Pins $P1_2$, $P1_3$, $P1_5$, and $P1_7$)

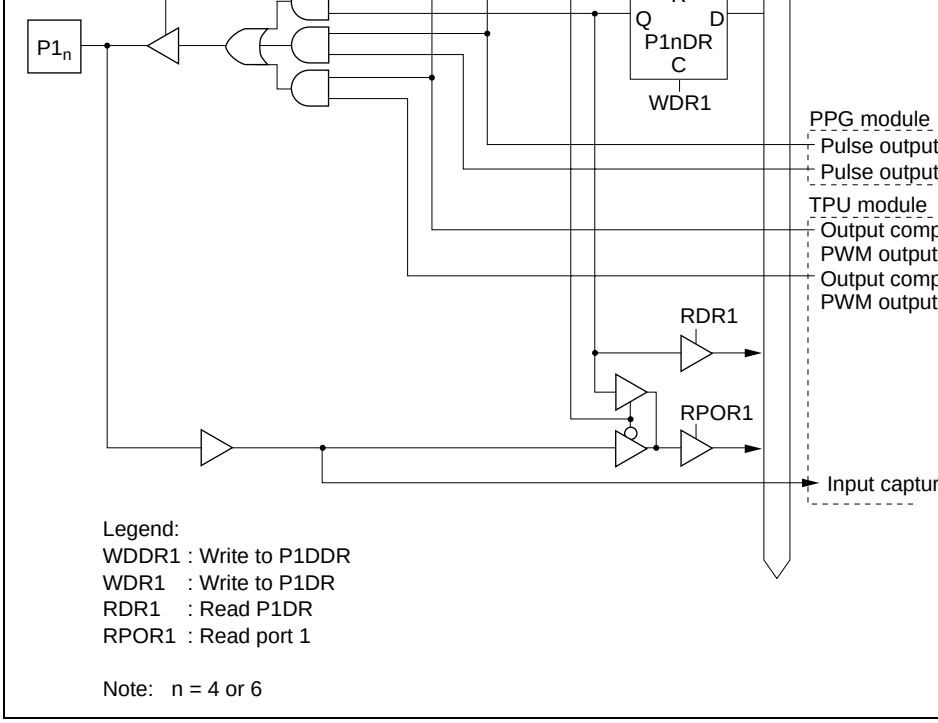


Figure C.1 (c) Port 1 Block Diagram (Pins $P1_4$ and $P1_6$)

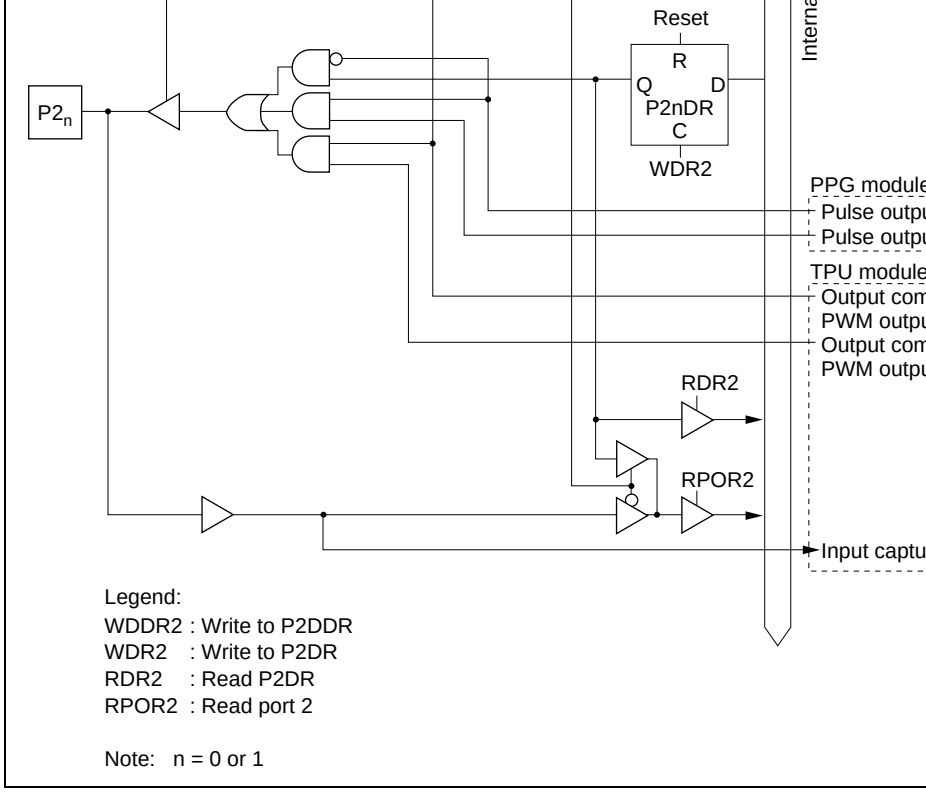
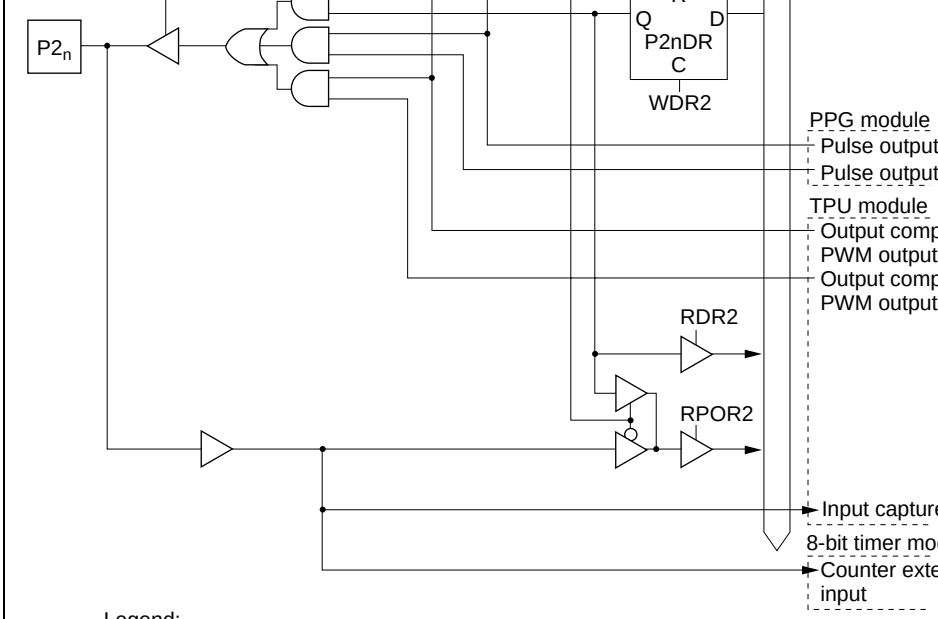


Figure C.2 (a) Port 2 Block Diagram (Pins $P2_0$ and $P2_1$)



Legend:
WDDR2 : Write to P2DDR
WDR2 : Write to P2DR
RDR2 : Read P2DR
RPOR2 : Read port 2

Note: n = 2 or 4

Figure C.2 (b) Port 2 Block Diagram (Pins $P2_2$ and $P2_4$)

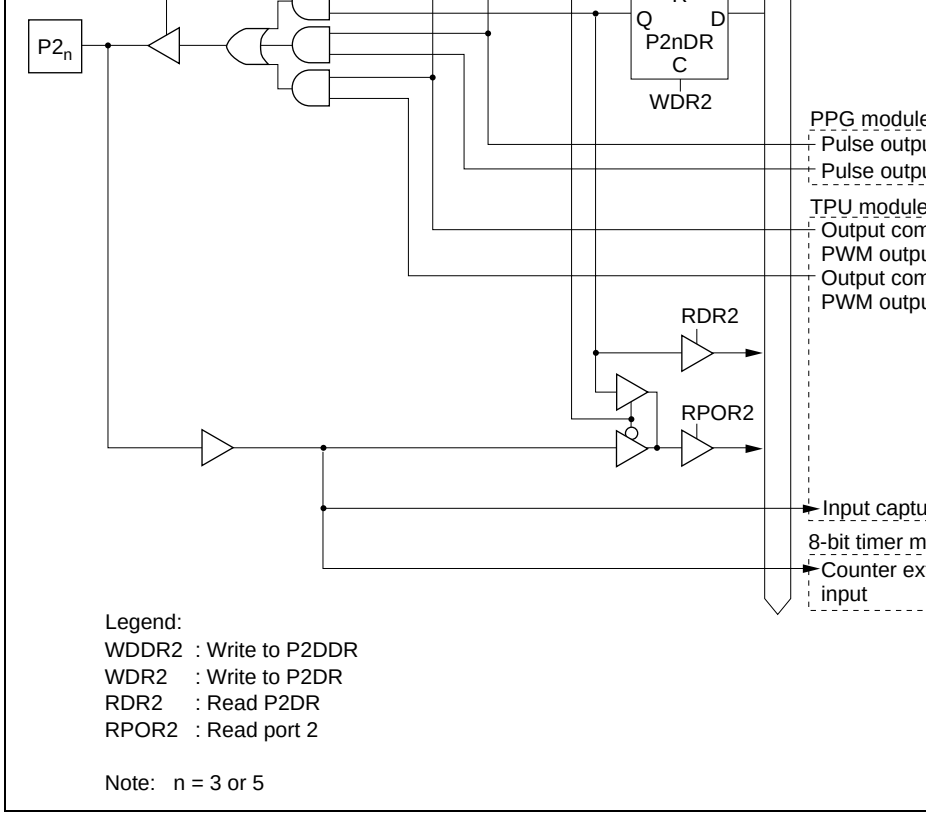


Figure C.2 (c) Port 2 Block Diagram (Pins $P2_3$ and $P2_5$)

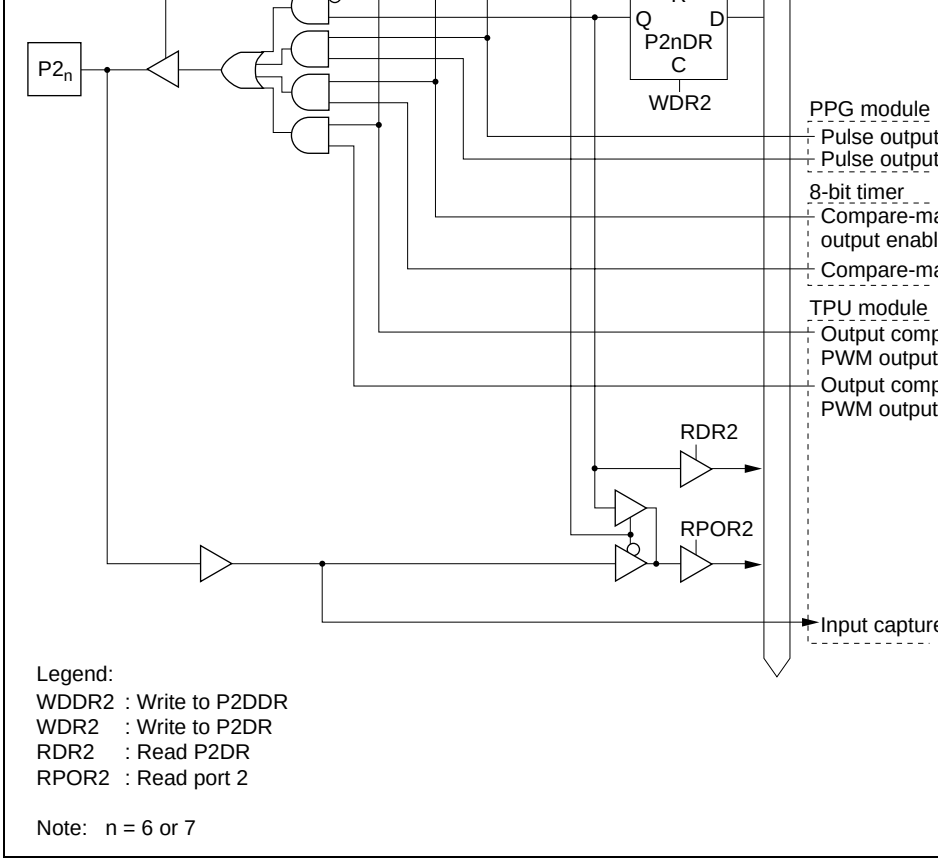


Figure C.2 (d) Port 2 Block Diagram (Pins $P2_6$ and $P2_7$)

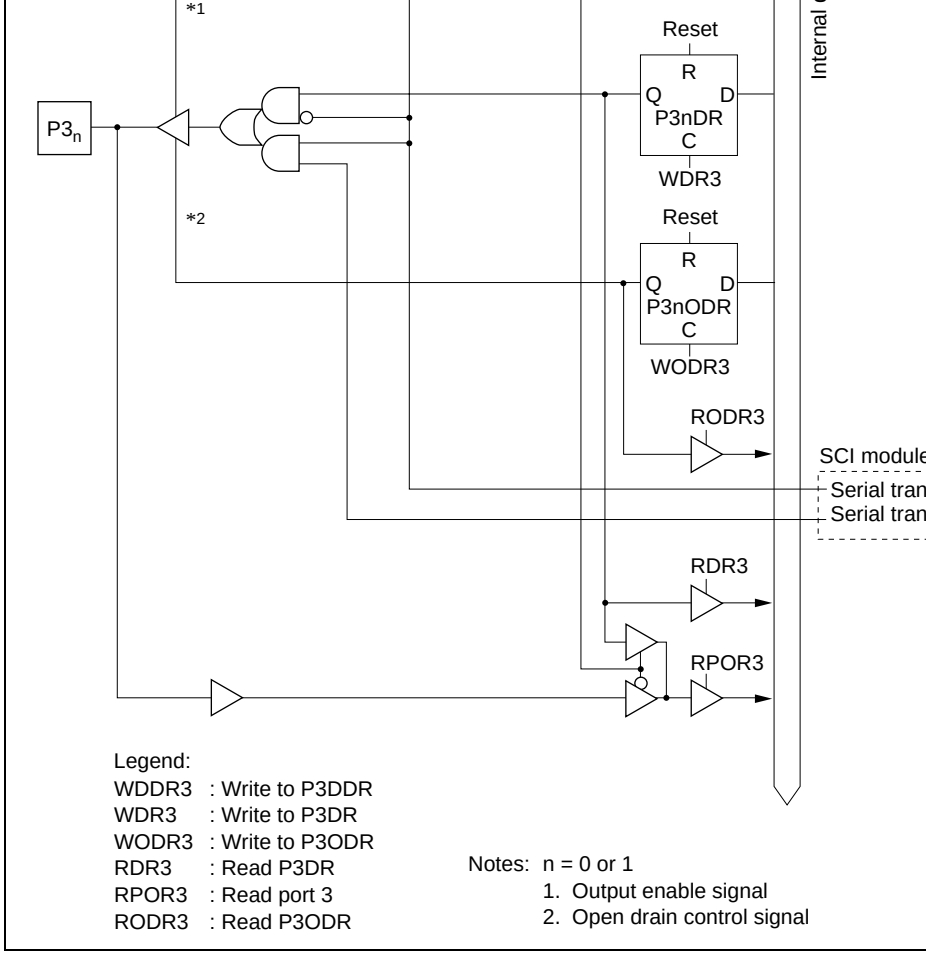


Figure C.3 (a) Port 3 Block Diagram (Pins $P3_0$ and $P3_1$)

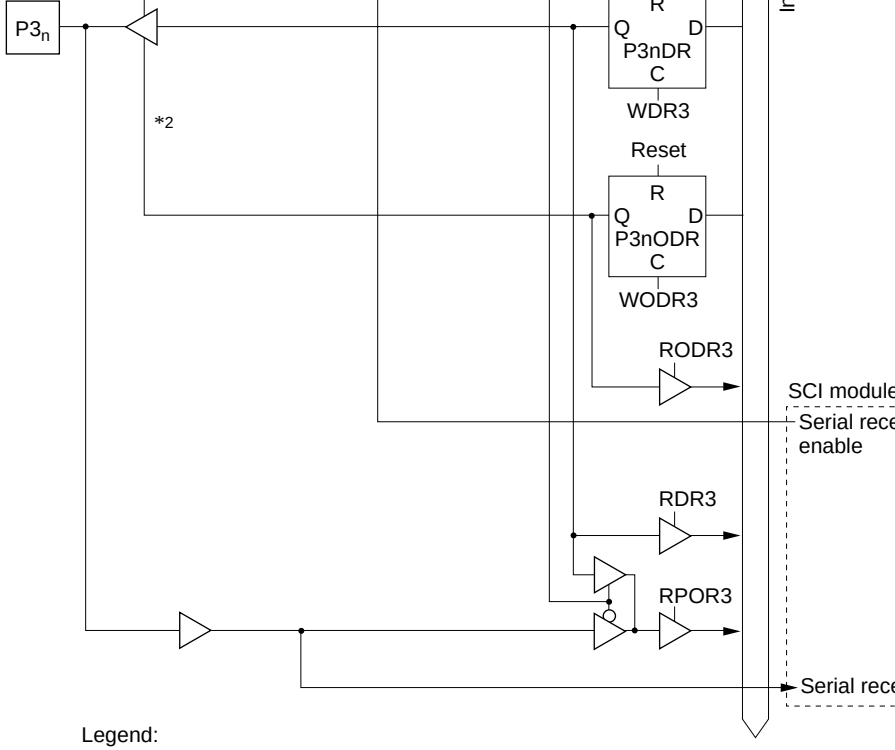


Figure C.3 (b) Port 3 Block Diagram (Pins $P3_2$ and $P3_3$)

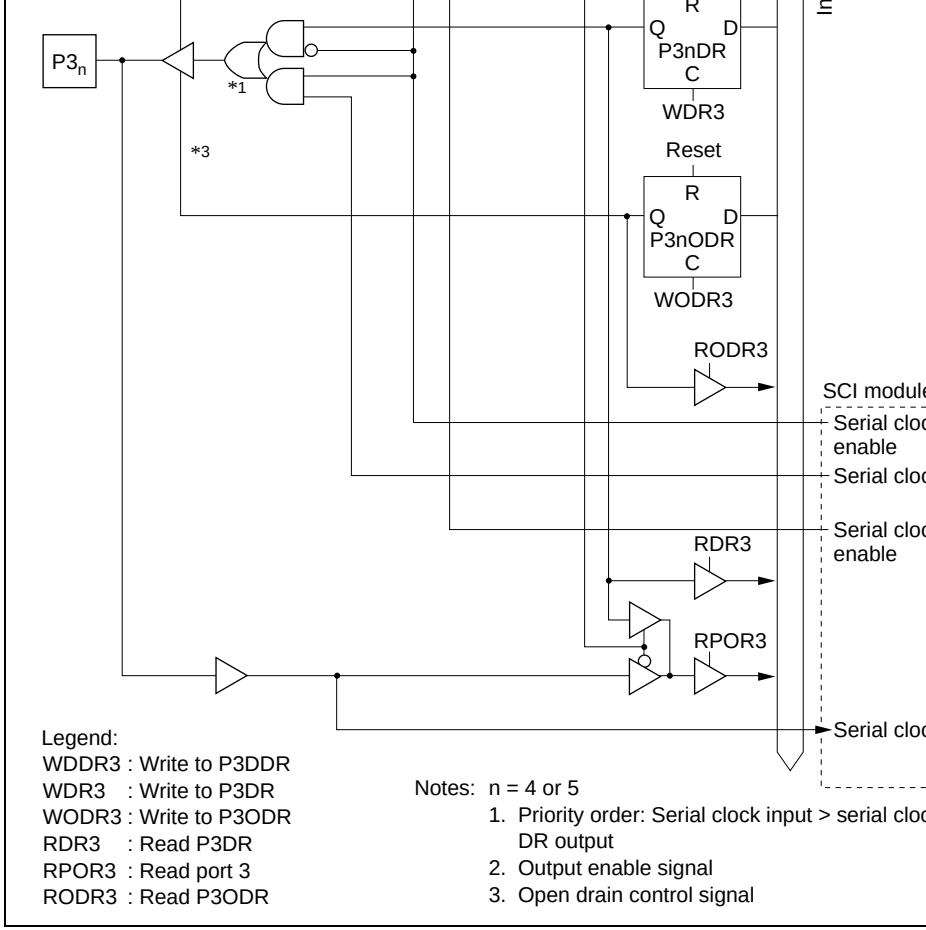
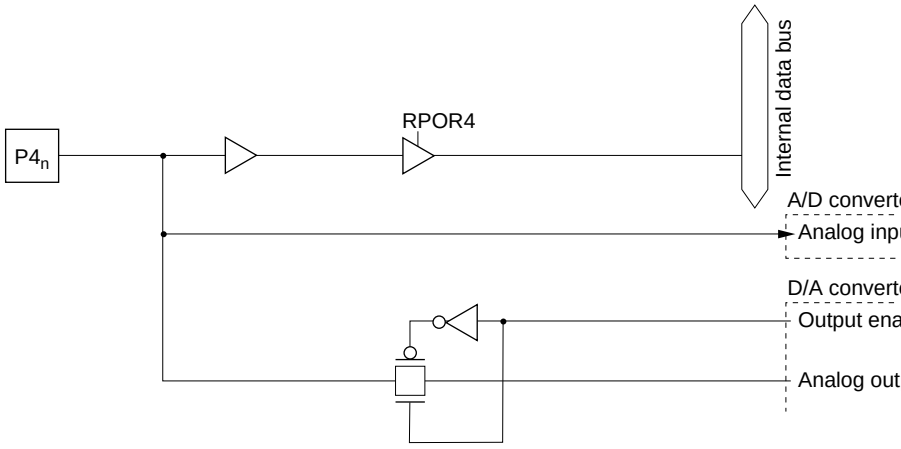


Figure C.3 (c) Port 3 Block Diagram (Pins $P3_4$ and $P3_5$)

Legennd:
RPOR4 : Read port 4

Note: n = 0 to 5

Figure C.4 (a) Port 4 Block Diagram (Pins $P4_0$ to $P4_5$)



Legend:
RPOR4 : Read port 4

Note: n = 6 or 7

Figure C.4 (b) Port 4 Block Diagram (Pins $P4_6$ and $P4_7$)

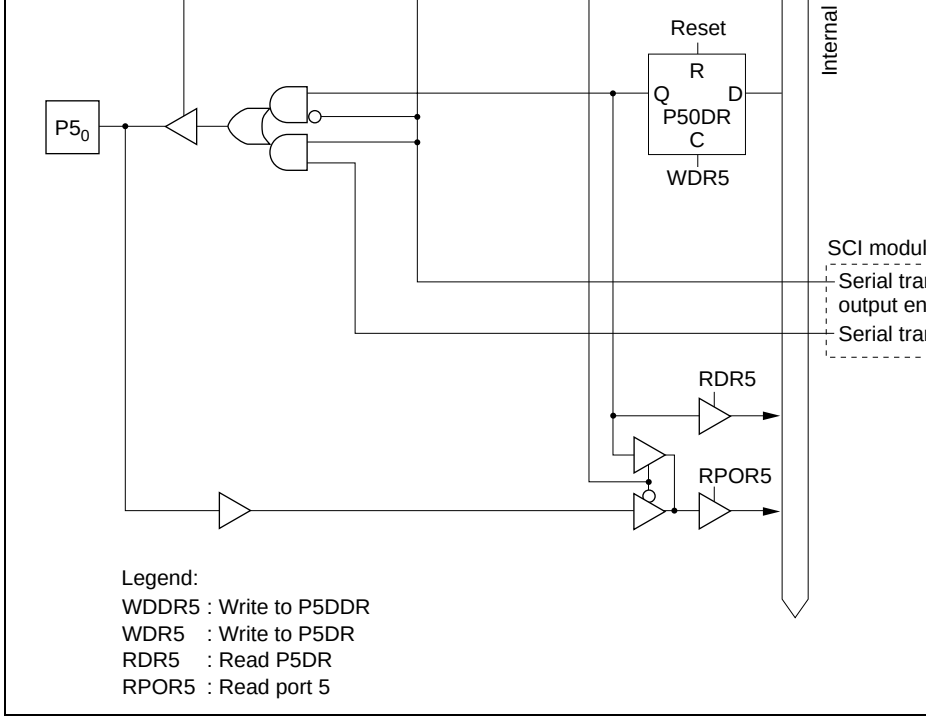


Figure C.5 (a) Port 5 Block Diagram (Pin P5$_0$)

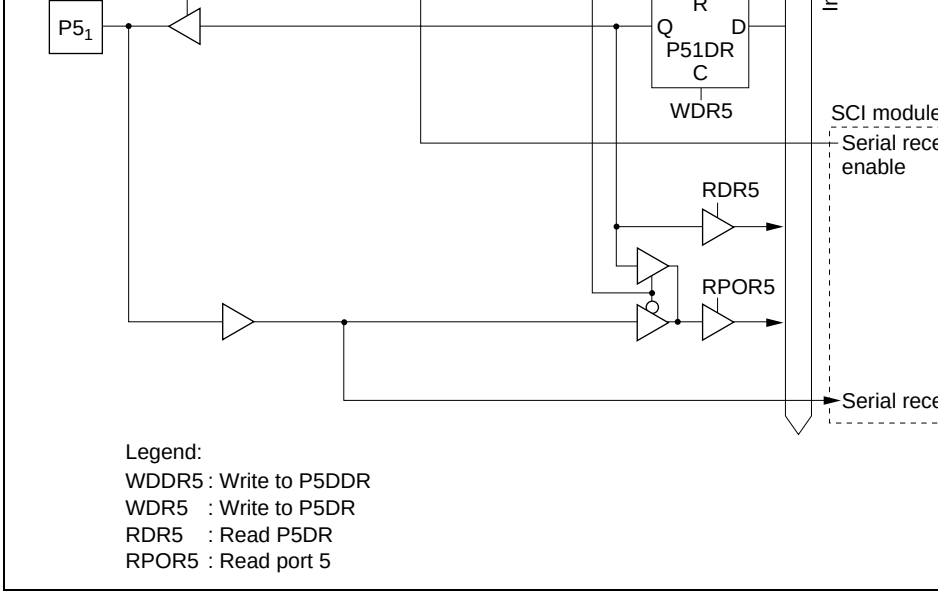
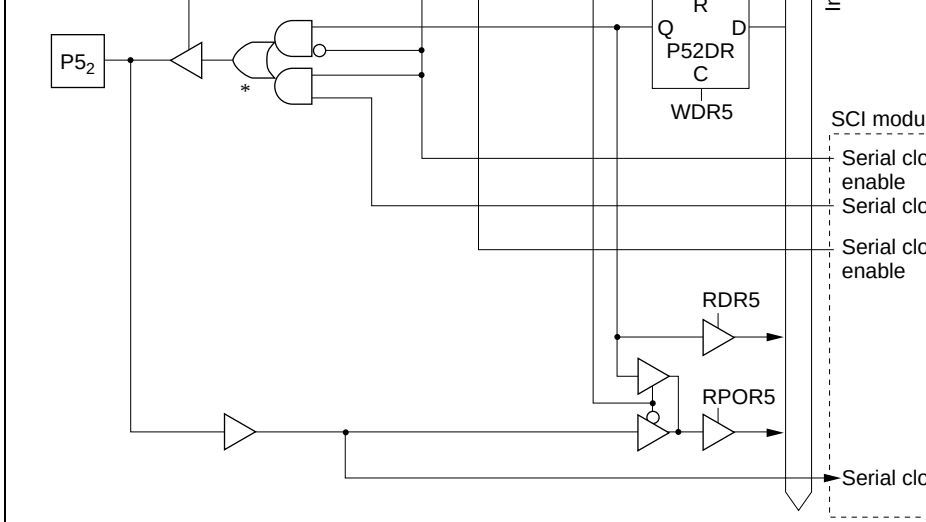


Figure C.5 (b) Port 5 Block Diagram (Pin P5$_1$)



Legend:
WDDR5 : Write to P5DDR
WDR5 : Write to P5DR
RDR5 : Read P5DR
RPOR5 : Read port 5

Note: * Priority order: Serial clock input > serial clock output > DR output

Figure C.5 (c) Port 5 Block Diagram (Pin P5$_2$)

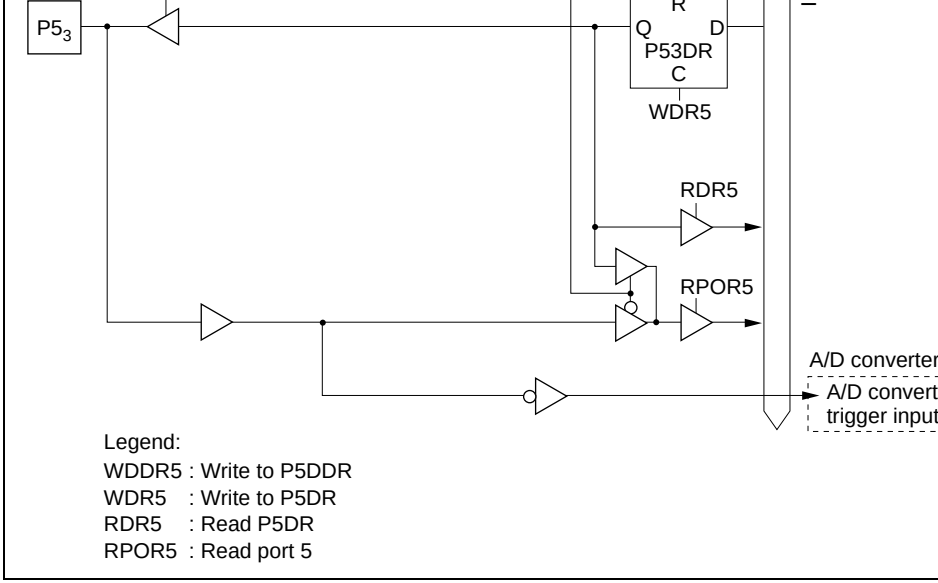


Figure C.5 (d) Port 5 Block Diagram (Pin P5$_3$)

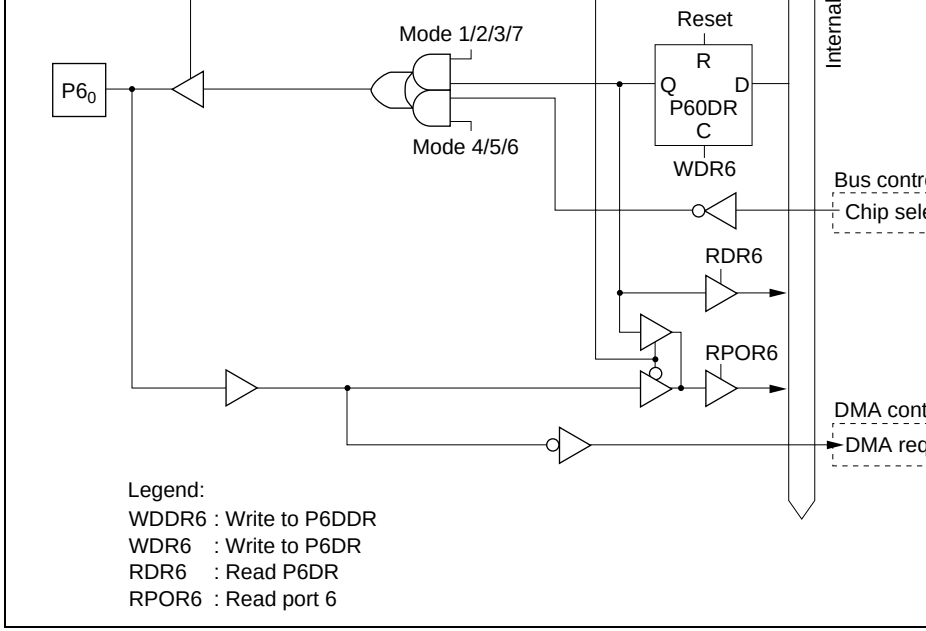


Figure C.6 (a) Port 6 Block Diagram (Pin P6$_0$)

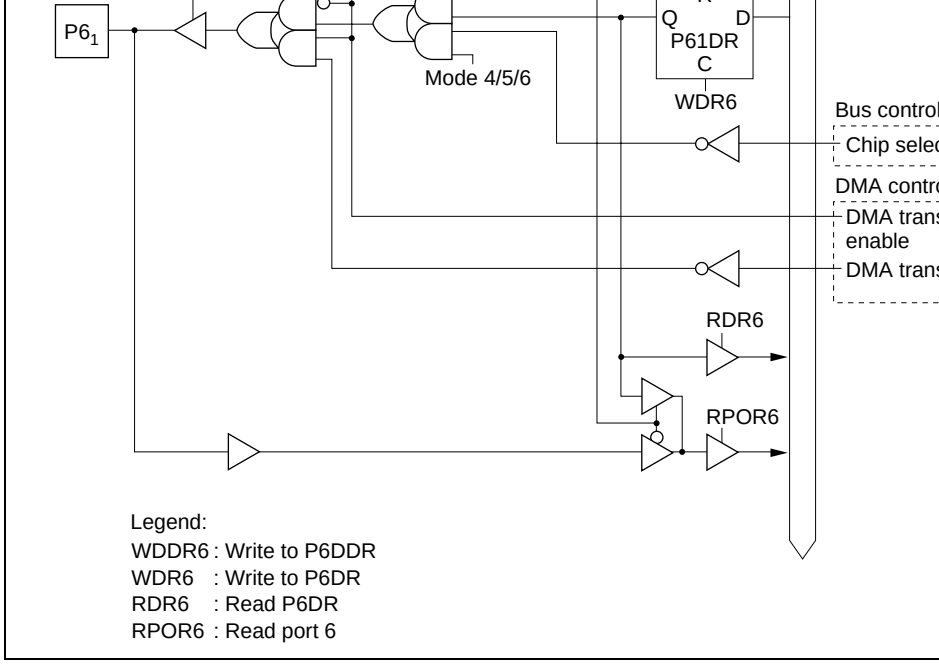


Figure C.6 (b) Port 6 Block Diagram (Pin P6$_1$)

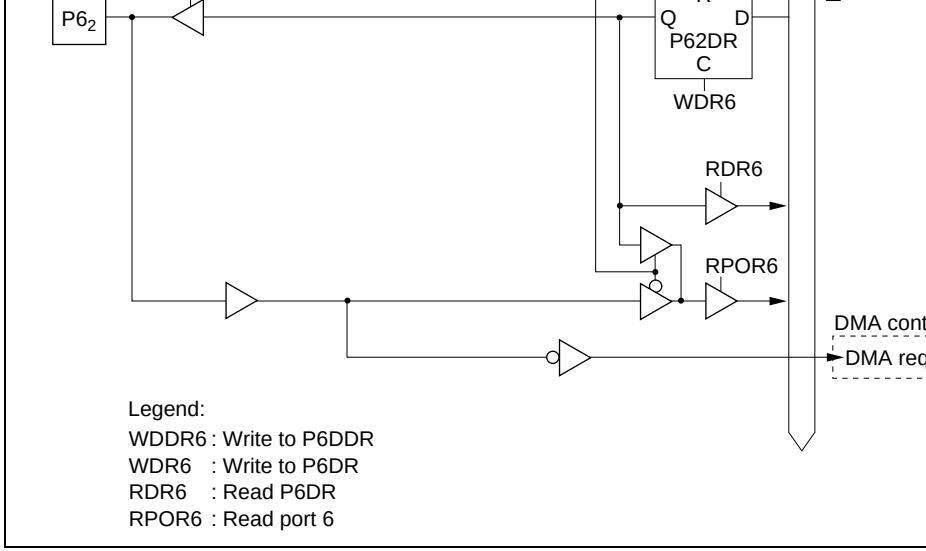


Figure C.6 (c) Port 6 Block Diagram (Pin P6$_2$)

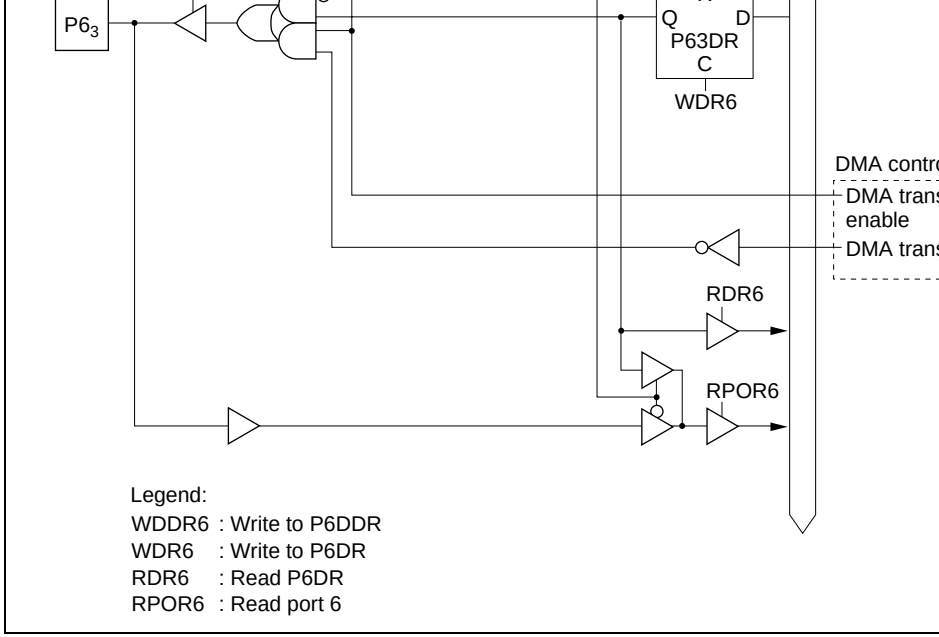
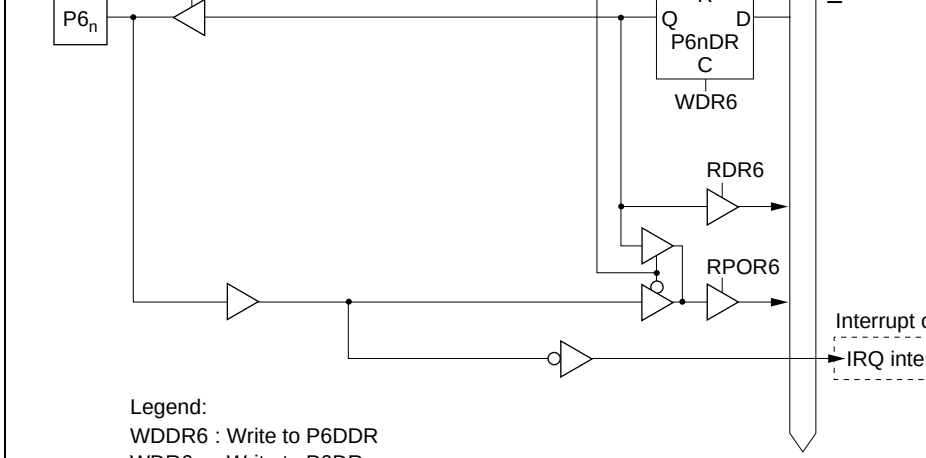


Figure C.6 (d) Port 6 Block Diagram (Pin P6$_3$)



Legend:
WDDR6 : Write to P6DDR
WDR6 : Write to P6DR
RDR6 : Read P6DR
RPOR6 : Read port 6

Note: n = 4 or 5

Figure C.6 (e) Port 6 Block Diagram (Pins $P6_4$ and $P6_5$)

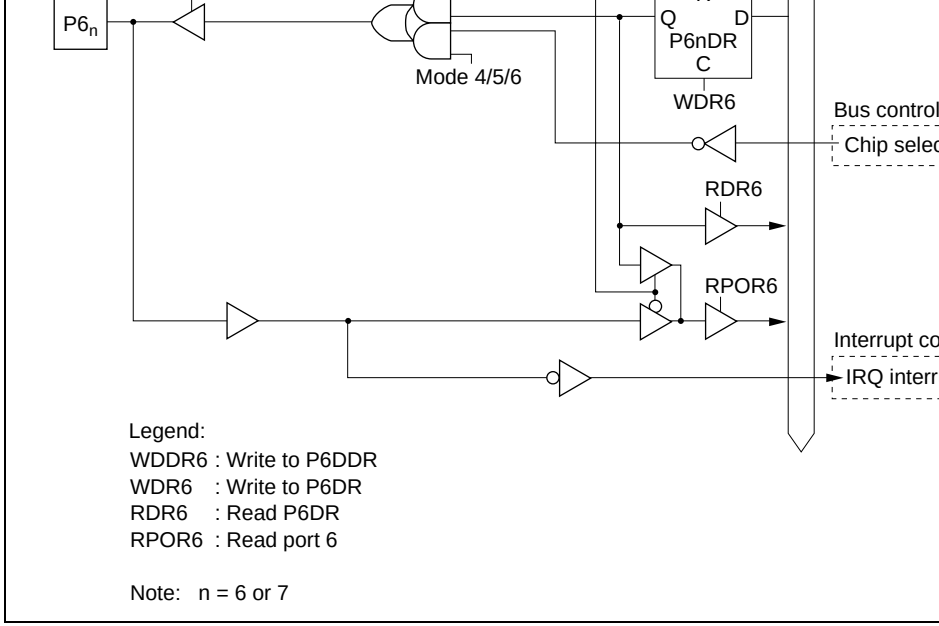


Figure C.6 (f) Port 6 Block Diagram (Pins $P6_6$ and $P6_7$)

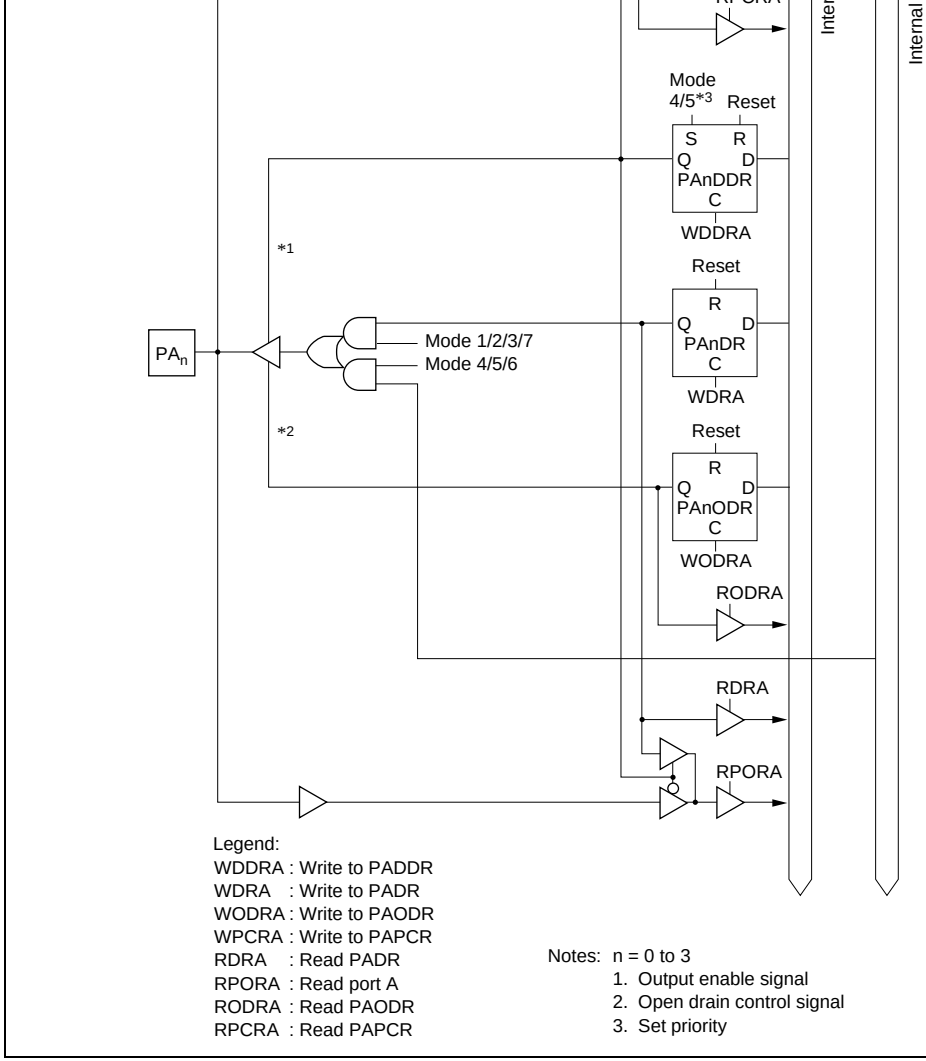


Figure C.7 (a) Port A Block Diagram (Pins PA_0 to PA_3)

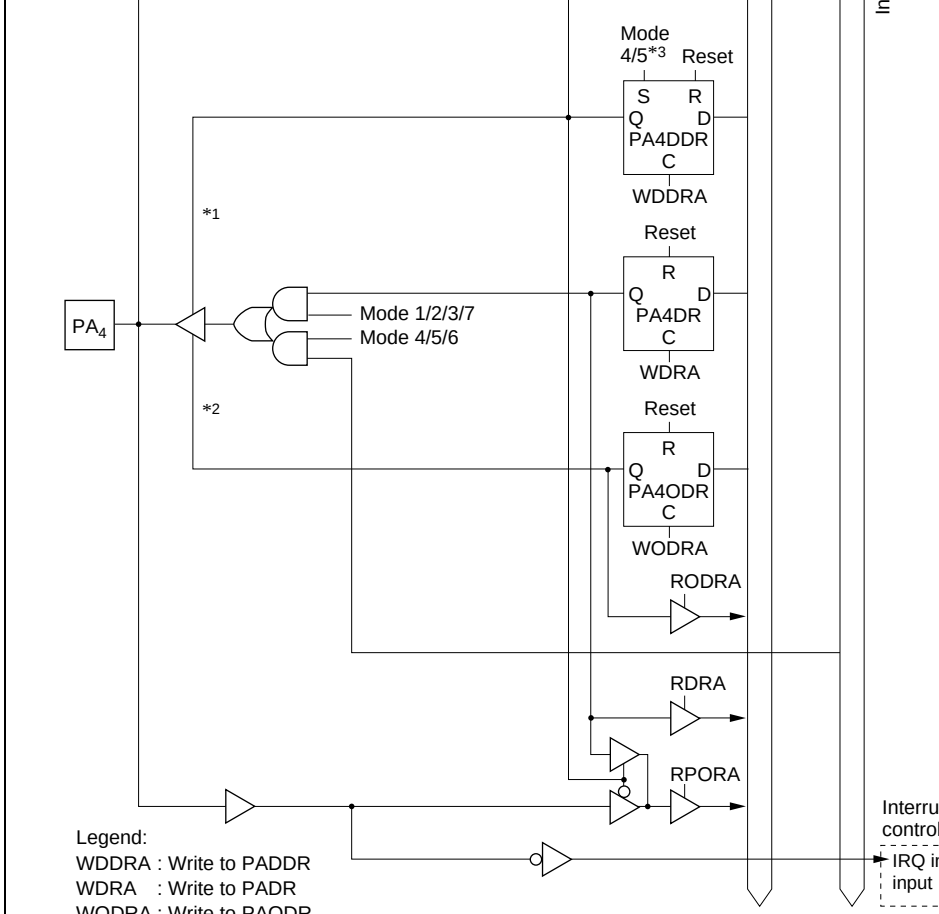


Figure C.7 (b) Port A Block Diagram (Pin PA_4)

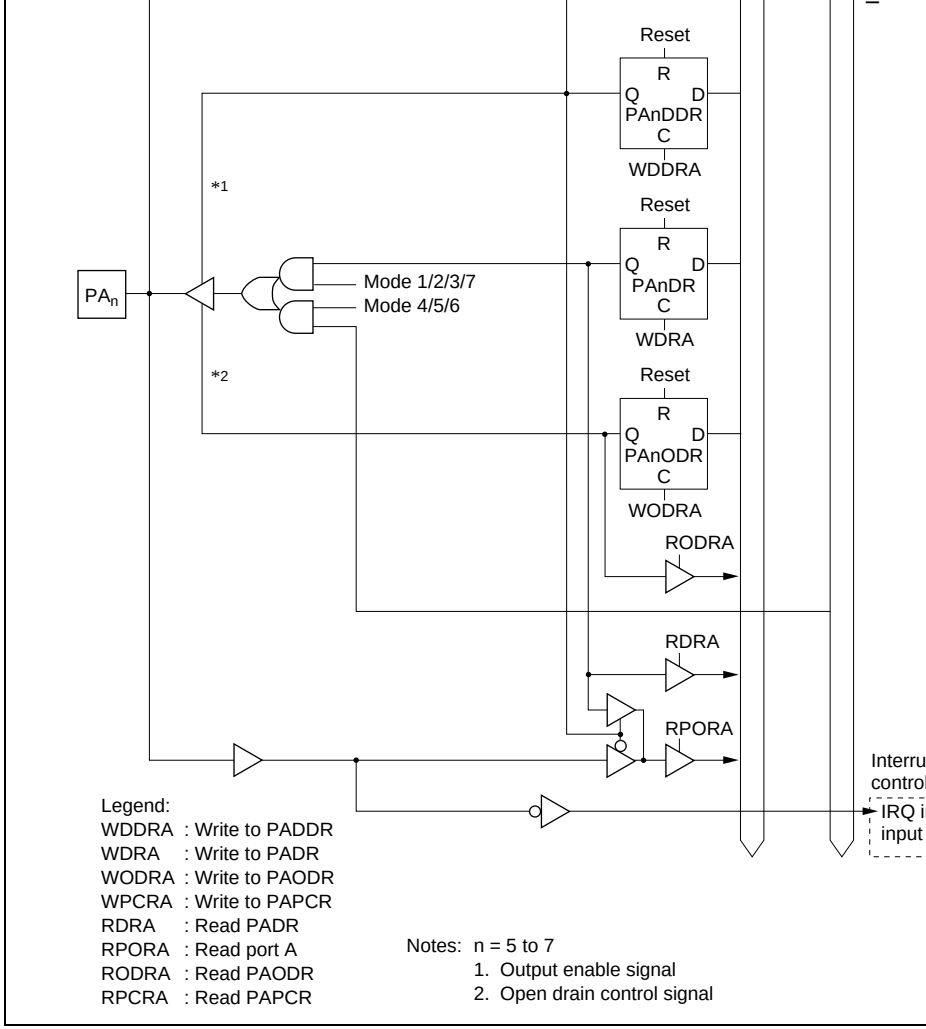


Figure C.7 (c) Port A Block Diagram (Pins PA_5 to PA_7)

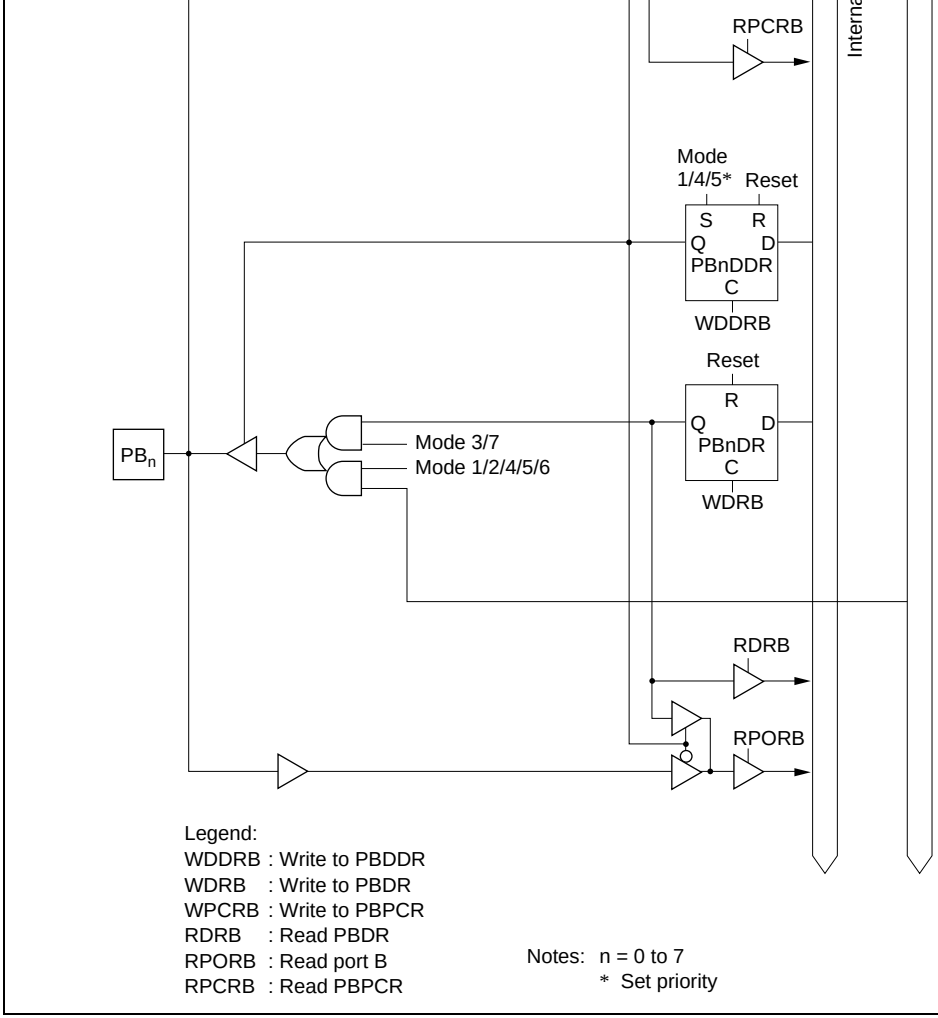


Figure C.8 Port B Block Diagram (Pin PB_n)

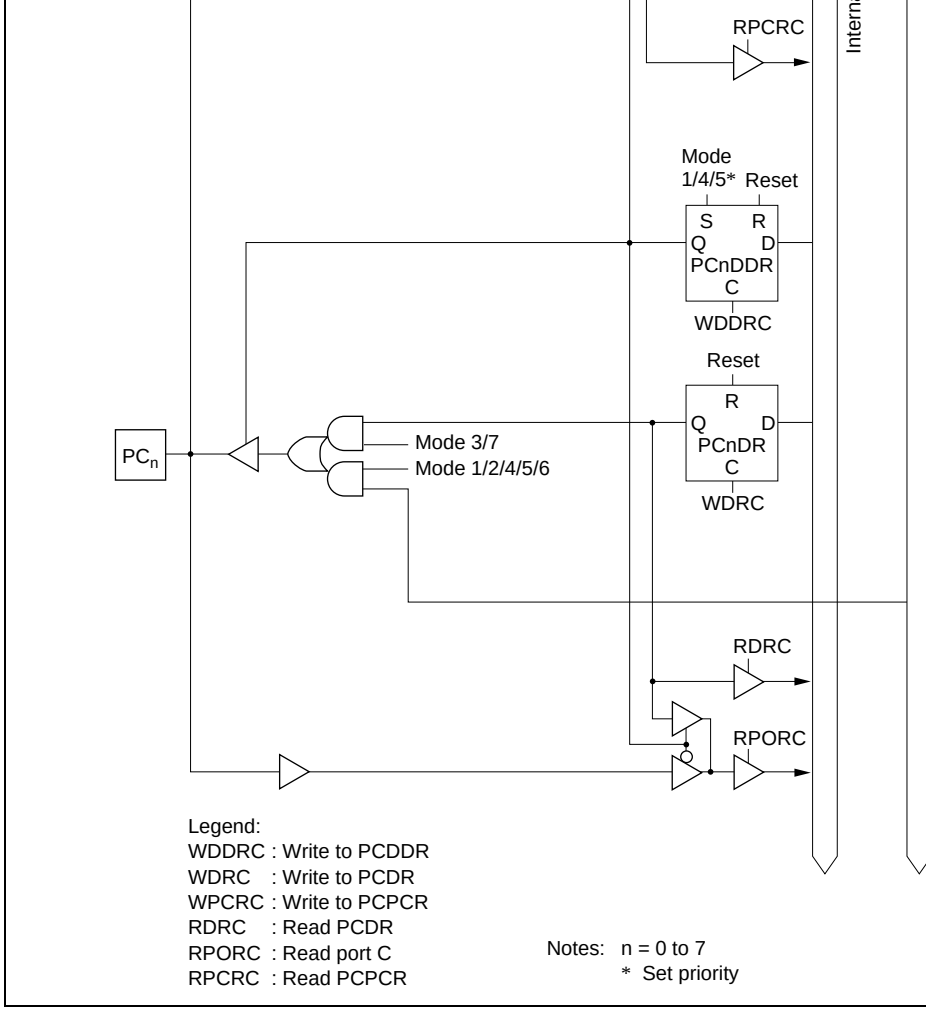


Figure C.9 Port C Block Diagram (Pin PC_n)

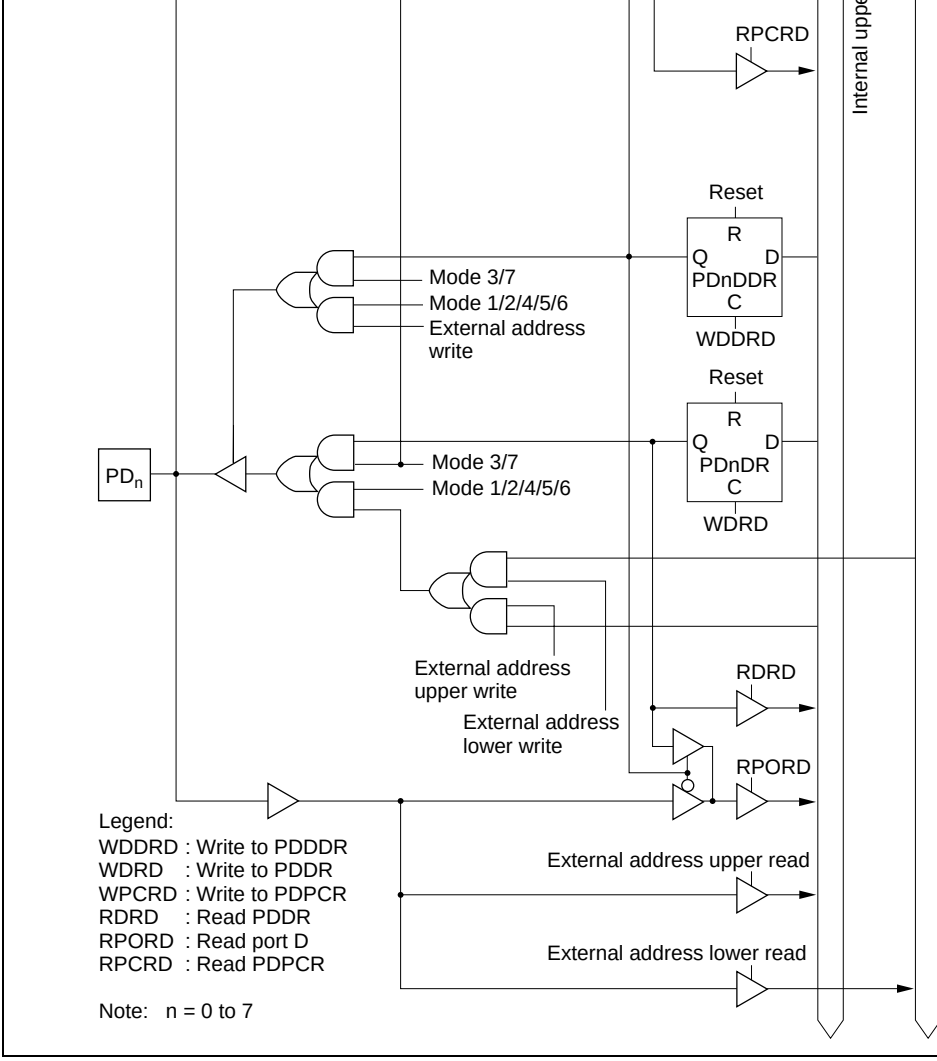


Figure C.10 Port D Block Diagram (Pin PD_n)

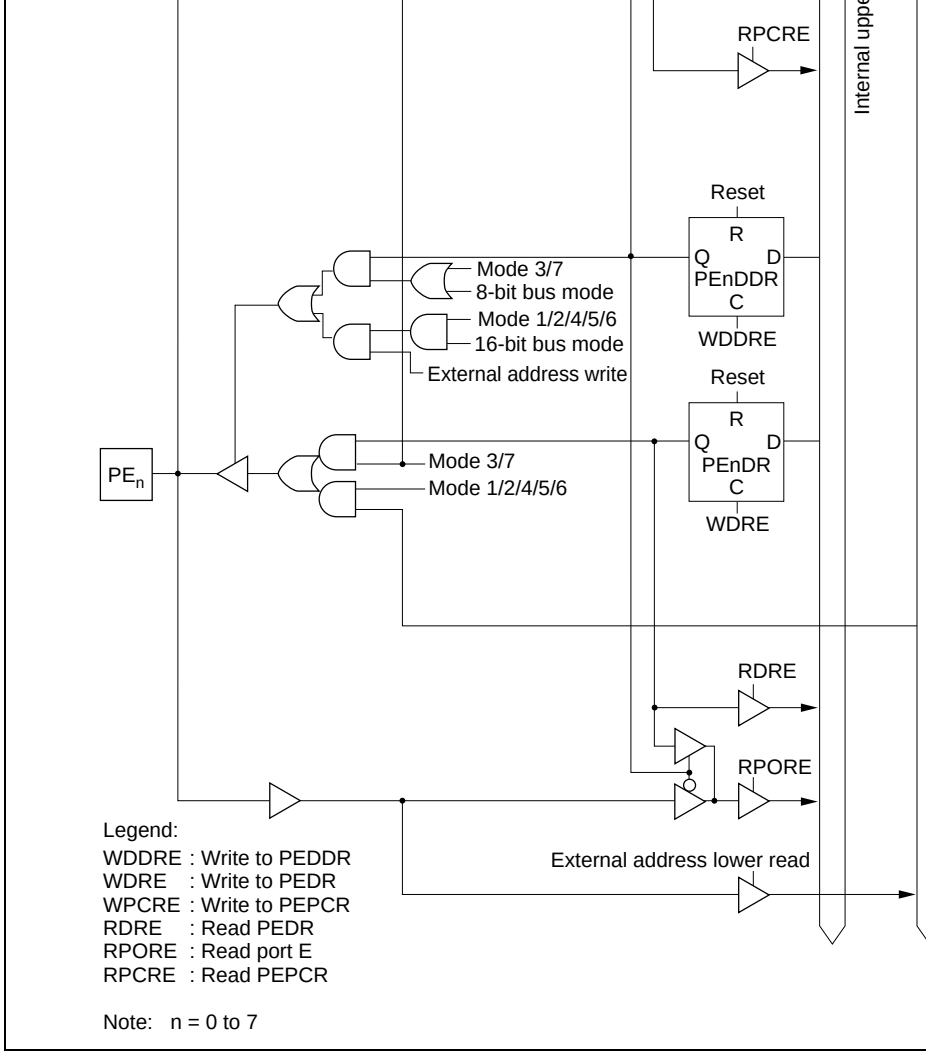


Figure C.11 Port E Block Diagram (Pin PE_n)

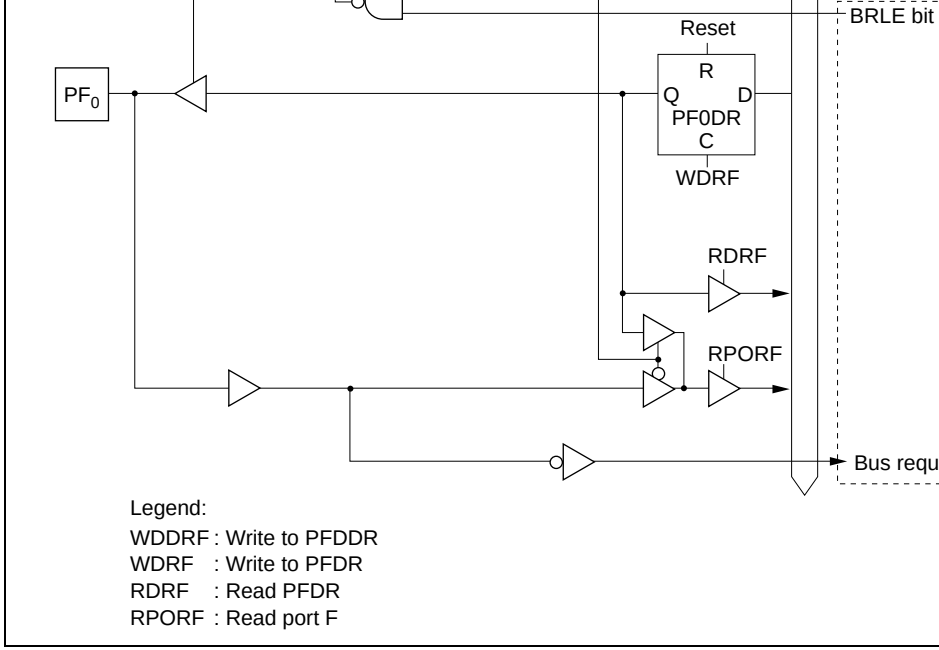


Figure C.12 (a) Port F Block Diagram (Pin PF_0)

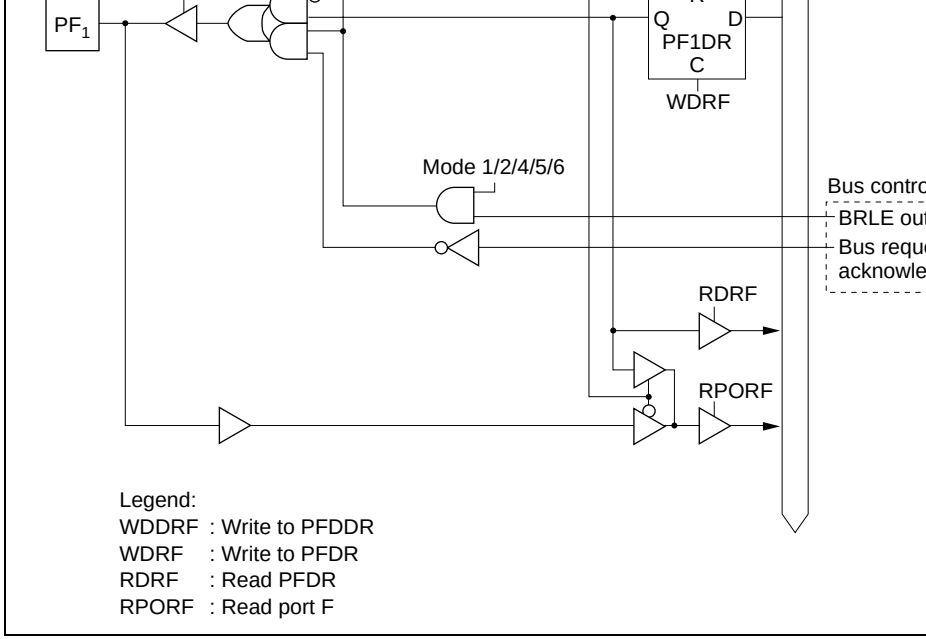


Figure C.12 (b) Port F Block Diagram (Pin PF_1)

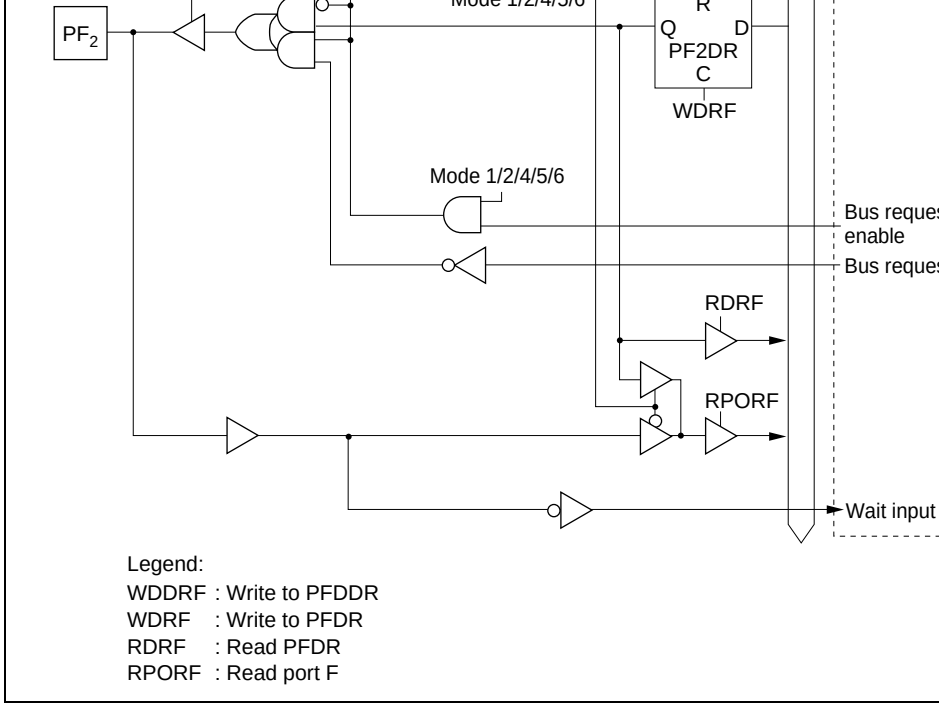


Figure C.12 (c) Port F Block Diagram (Pin PF_2)

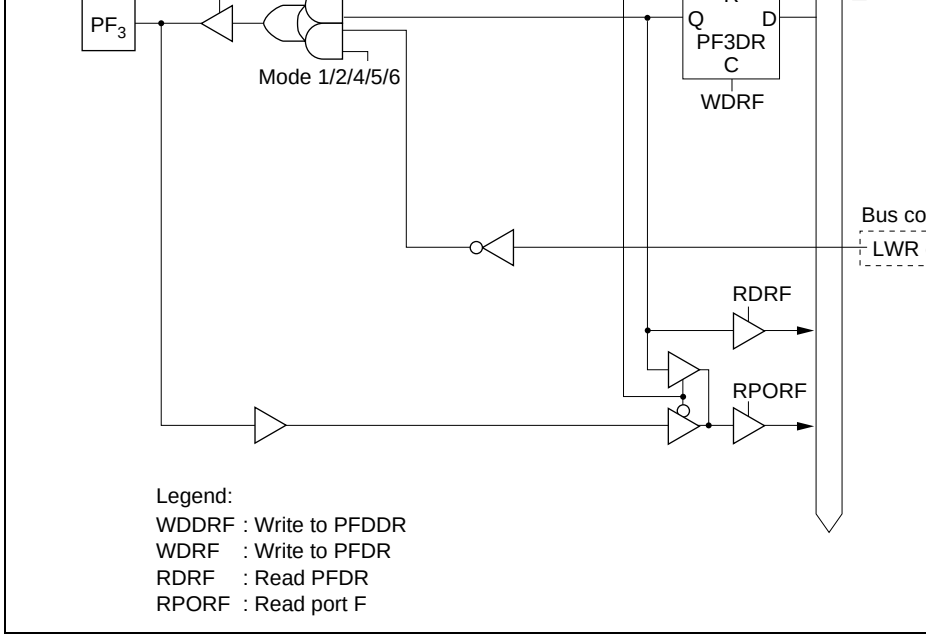


Figure C.12 (d) Port F Block Diagram (Pin PF_3)

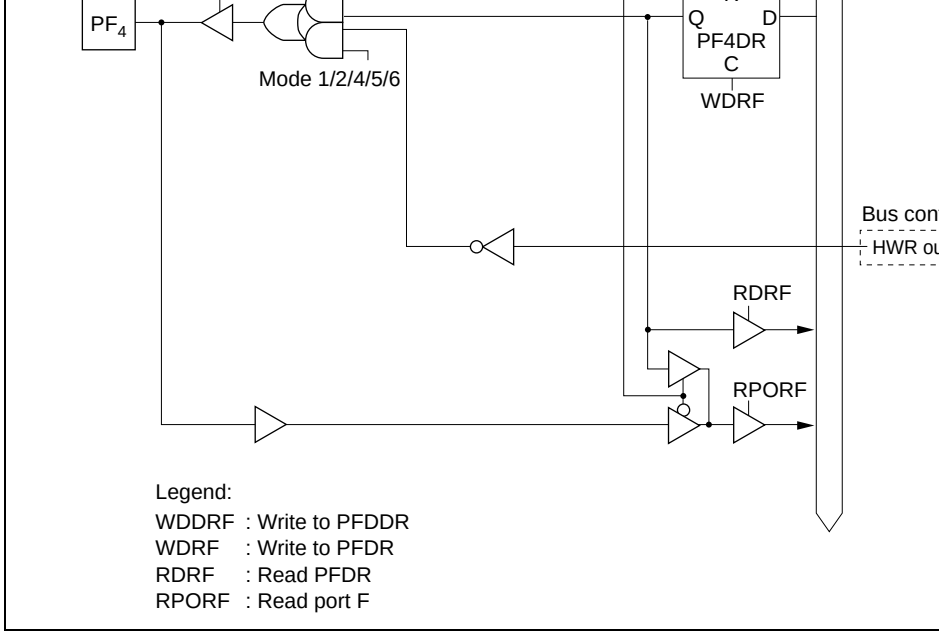


Figure C.12 (e) Port F Block Diagram (Pin PF_4)

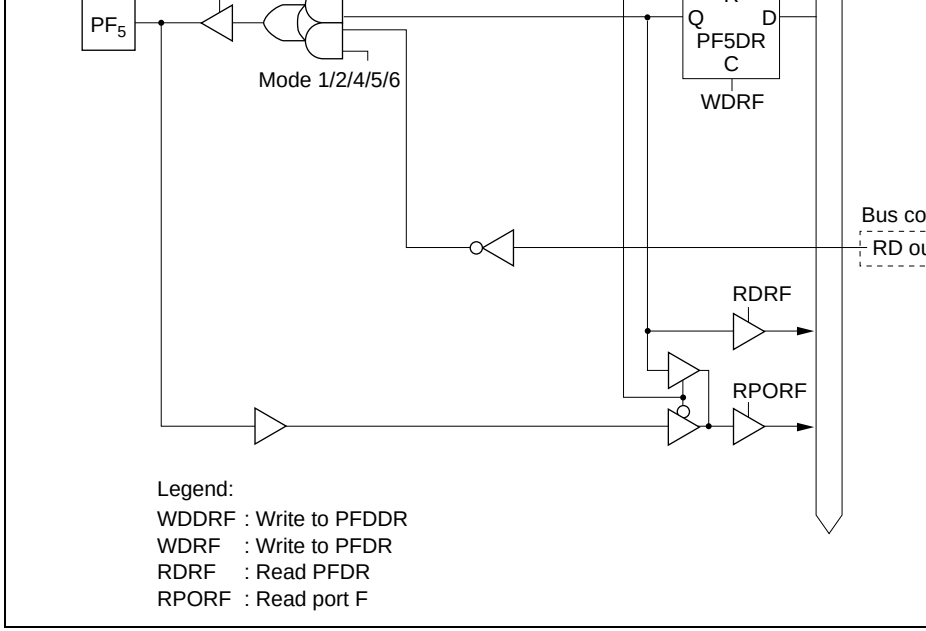


Figure C.12 (f) Port F Block Diagram (Pin PF_5)

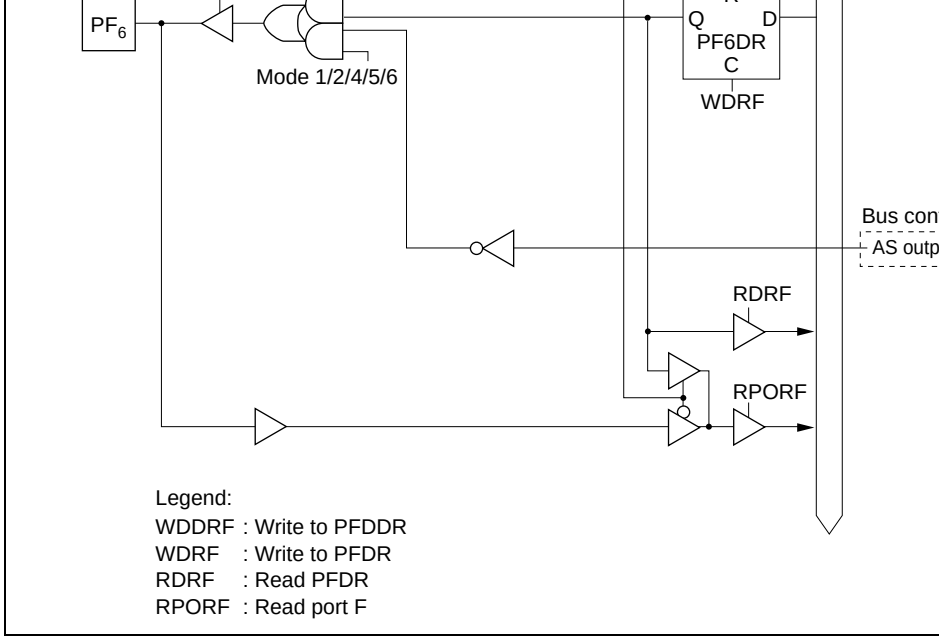
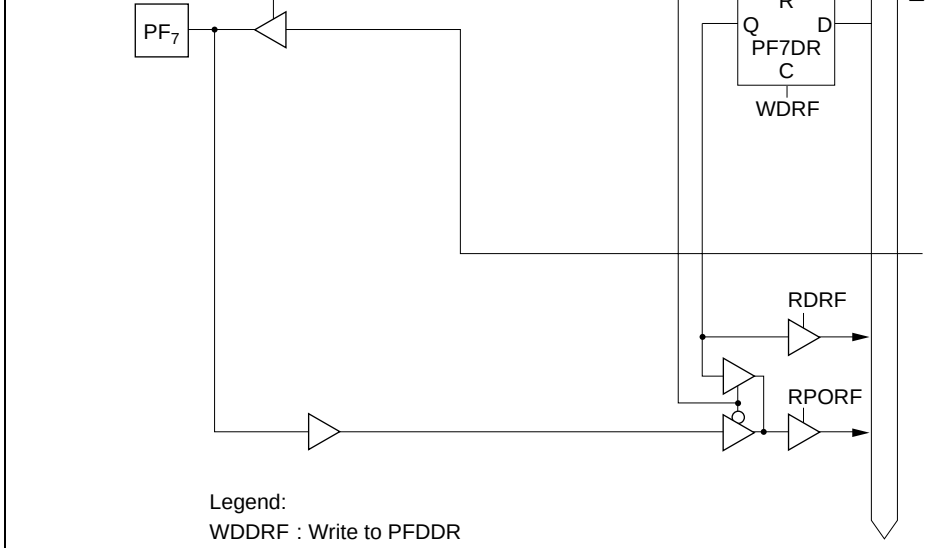


Figure C.12 (g) Port F Block Diagram (Pin PF_6)



Legend:
WDDRF : Write to PFDDR
WDRF : Write to PFDR
RDRF : Read PFDR
RPORF : Read port F

Note: * Set priority

Figure C.12 (h) Port F Block Diagram (Pin PF_7)

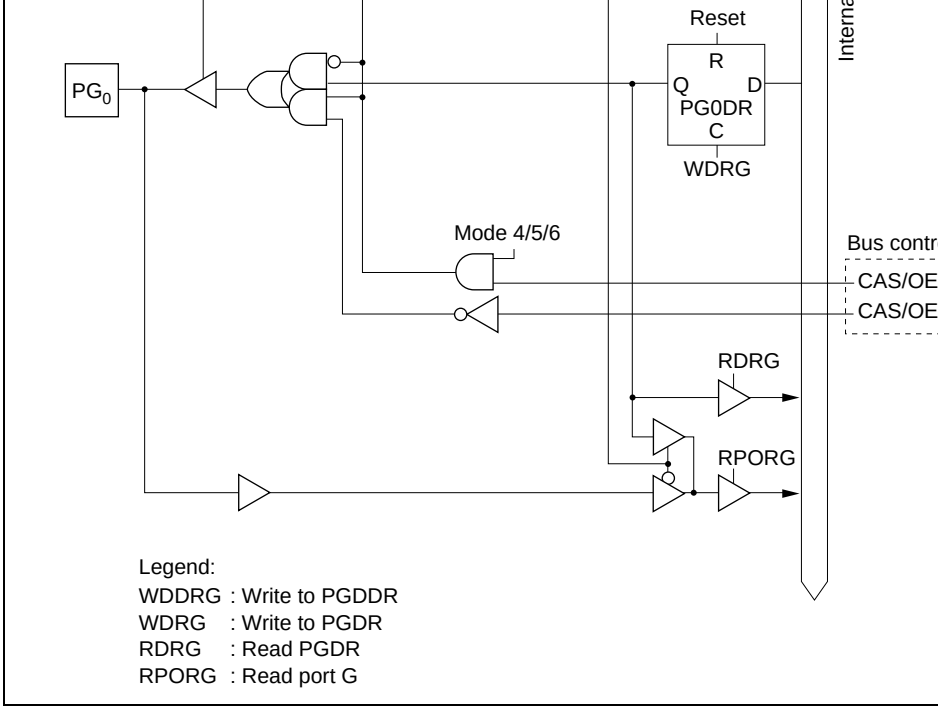
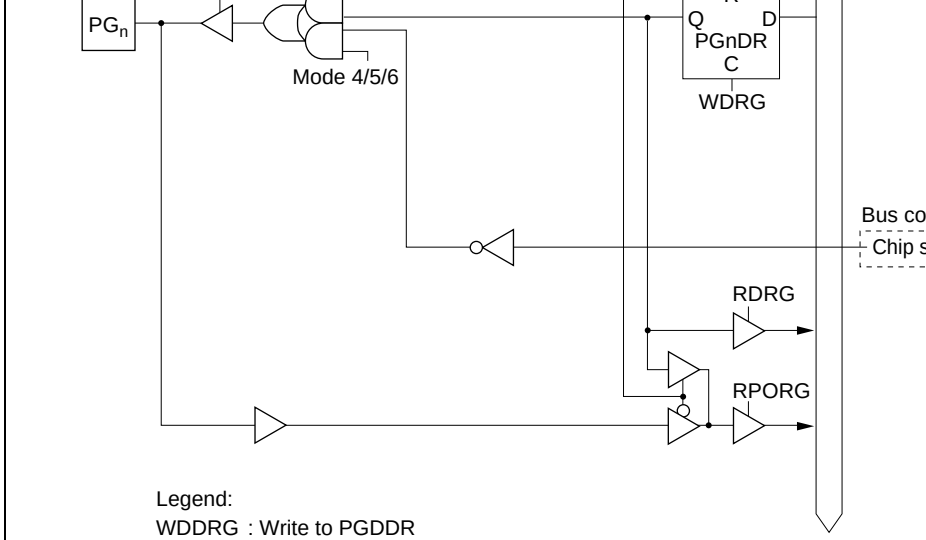


Figure C.13 (a) Port G Block Diagram (Pin PG_0)



Legend:
WDDRG : Write to PGDDR
WDRG : Write to PGDR
RDRG : Read PGDR
RPORG : Read port G

Note: n = 1, 2, 3

Figure C.13 (b) Port G Block Diagram (Pins PG_1 to PG_3)

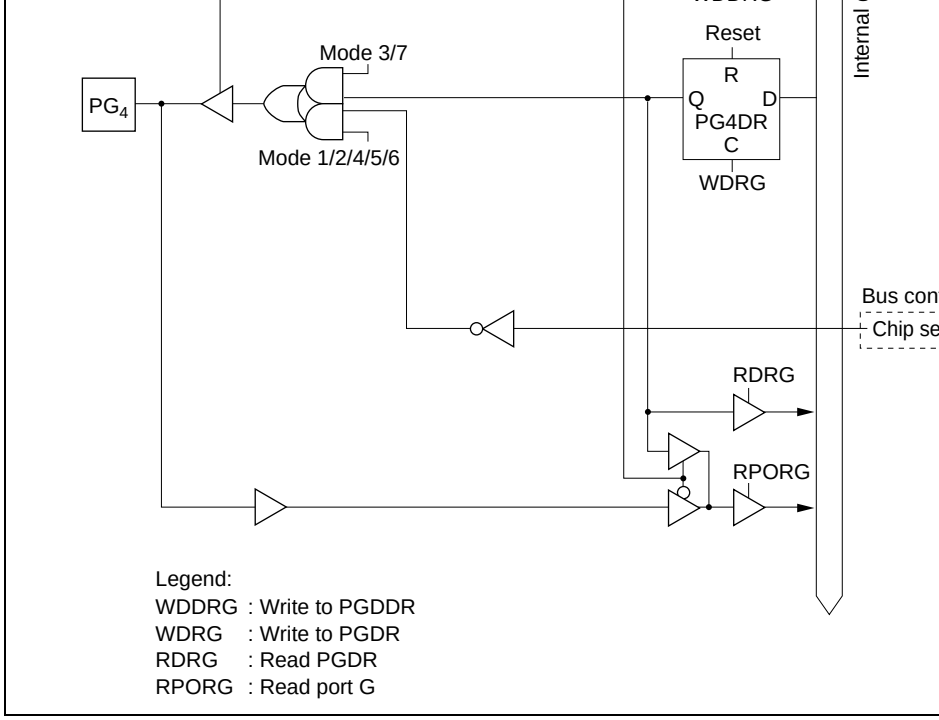


Figure C.13 (c) Port G Block Diagram (Pin PG_4)

Port Name Pin Name	Operating Mode	On Reset	Manual Reset	Standby Mode	Standby Mode	Release State
Port 1	1 to 7	T	kept	T	kept	kept
Port 2	1 to 7	T	kept	T	kept	kept
Port 3	1 to 7	T	kept	T	kept	kept
Port 4	1 to 7	T	T	T	T	T
Port 5	1 to 7	T	kept	T	kept	kept
$P6_5$ to $P6_2$	1 to 7	T	kept	T	kept	kept
	1 to 3, 7	T	kept	T	kept	kept
$P6_7/\overline{CS_7}$ $P6_6/\overline{CS_6}$ $P6_1/\overline{CS_5}$ $P6_0/\overline{CS_4}$	4 to 6	T	kept	T	[DDR · OPE = 0] T [DDR · OPE = 1] H	T
Port A	1 to 3, 7	T	kept	T	kept	kept
PA0 to PA3	4, 5	L	kept	T	[OPE = 0] T [OPE = 1] kept	T
	6	T	kept	T	[DDR · OPE = 0] T [DDR · OPE = 1] kept	T

	0	T	kept	T	[DDR · OPE = 0] T [DDR · OPE = 1] kept	T	[D In [D A ou
Port A	1 to 3, 7	T	kept	T	kept	kept	I/
PA5 to PA7	4, 5	T	kept	T	[DDR · OPE = 0] T [DDR · OPE = 1] kept	T	[D In [D A ou
	6	T	kept	T	[DDR · OPE = 0] T [DDR · OPE = 1] kept	T	[D In [D A ou
Port B	1, 4, 5	L	kept	T	[OPE = 0] T [OPE = 1] kept	T	A ou
	2, 6	T	kept	T	[DDR · OPE = 0] T [DDR · OPE = 1] kept	T	[D In [D A ou
	3, 7	T	kept	T	kept	kept	I/

						[DDR · OPE = 1] kept	
	3, 7		T	kept	T	kept	kept
Port D	1, 2, 4 to 6		T	T*	T	T	T
	3, 7		T	kept	T	kept	kept
Port E	1, 2, 4 to 6	8-bit bus	T	kept	T	kept	kept
		16-bit bus	T	T*	T	T	T
	3, 7		T	kept	T	kept	kept
PF_7/ϕ	1, 2, 4 to 6		Clock output	[DDR = 0] T [DDR = 1] Clock output	T	[DDR = 0] Input port [DDR = 1] H	[DDR = 0] Input port [DDR = 1] Clock output
	3, 7		T	kept	T	[DDR = 0] Input port [DDR = 1] H	[DDR = 0] Input port [DDR = 1] Clock output
$PF_6/\overline{AS}$ $PF_5/\overline{RD}$ $PF_4/\overline{HWR}$	1, 2, 4 to 6		H	H*	T	[OPE = 0] T [OPE = 1] H	T
$PF_3/\overline{LWR}$	3, 7		T	kept	T	kept	kept

			[WAITE = 1] T [LCASE = 1] H*		[WAITE = 1] T [LCASE = 1, OPE = 0] T [LCASE = 1, OPE = 1] $\overline{\text{LCAS}}$	[WAITE = 1] T [LCASE = 1] T	[W $\overline{W}$ [L $\overline{L}$
	3, 7	T	kept	T	kept	kept	I/
$PF_1/\overline{BACK}$	1, 2, 4 to 6	T	[BRLE = 0] kept [BRLE = 1] $\overline{\text{BACK}}$	T	[BRLE = 0] kept [BRLE = 1] $\overline{\text{BACK}}$	L	[B I/ [B $\overline{B}$
	3, 7	T	kept	T	kept	kept	I/
$PF_0/\overline{BREQ}$	1, 2, 4 to 6	T	[BRLE = 0] kept [BRLE = 1] $\overline{\text{BREQ}}$	T	[BRLE = 0] kept [BRLE = 1] T	T	[B I/ [B $\overline{B}$
	3, 7	T	kept	T	kept	kept	I/
$PG_4/\overline{CS_0}$	1, 4, 5	H	[DDR = 0]	T	[DDR · OPE = 0]	T	[D
	2, 6	T	T [DDR = 1] H*		T [DDR · OPE = 1] H		In [D $\overline{C}$
	3, 7	T	kept	T	kept	kept	I/
$PG_3/\overline{CS_1}$	1 to 3, 7	T	kept	T	kept	kept	I/
$PG_2/\overline{CS_2}$ $PG_1/\overline{CS_3}$	4 to 6	T	[DDR = 0] T [DDR = 1] H*	T	[DDR · OPE = 0] T [DDR · OPE = 1] H	T	[D In [D $\overline{C}$

		[PSRAME = 1] H*	[DRAME · OPE= 1] $\overline{CAS}$ [PSRAME · OPE = 1] $\overline{OE}$

Legend:

H	: High level
L	: Low level
T	: High impedance
kept	: Input port becomes high-impedance, output port retains state
DDR	: Data direction register
OPE	: Output port enable
WAITE	: Wait input enable
BRLE	: Bus release enable
BREQOE	: BREQO pin enable
DRAME	: DRAM space setting
LCASE	: DRAM space setting, CW2 = LCASS = 0
PSRAME	: PSRAM space setting

Note: * Indicates the state after completion of the executing bus cycle.

STBY signal goes low (delay from STBY low to RES high: 0 ns or more).

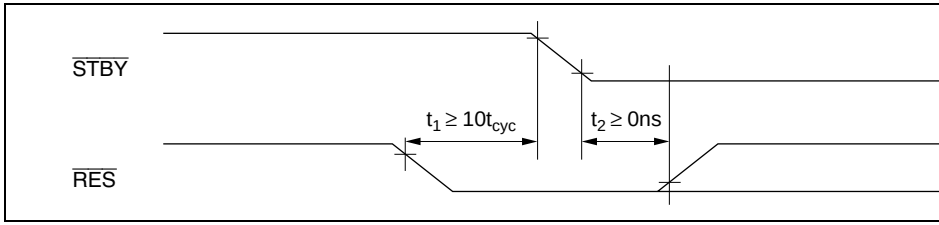


Figure E.1 Timing of Transition to Hardware Standby Mode

(2) To retain RAM contents with the RAME bit cleared to 0 in SYSCR, or when RAM not need to be retained, RES does not have to be driven low as in (1).

Timing of Recovery from Hardware Standby Mode

Drive the RES signal low and the NMI signal high approximately 100 ns or more befor goes high to execute a power-on reset.

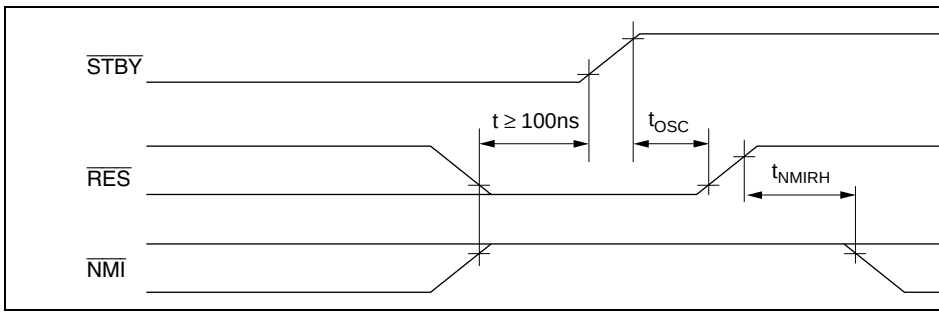


Figure E.2 Timing of Recovery from Hardware Standby Mode

				HD6432655(***)F	128-p (FP-1
		Low-voltage version (V_{CC} = 2.7 to 5.5 V)		HD6432655(***)TE	120-p (TFP
				HD6432655(***)F	128-p (FP-1
	ZTAT™	5 V version (V_{CC} = 5.0 V ±10%)	HD6472655	HD6472655TE	120-p (TFP
				HD6472655F	128-p (FP-1
		Low-voltage version (V_{CC} = 2.7 to 5.5 V)		HD6472655VTE	120-p (TFP
				HD6472655VF	128-p (FP-1
H8S/2653	Mask ROM	5 V version (V_{CC} = 5.0 V ±10%)	HD6432653	HD6432653(***)TE	120-p (TFP
				HD6432653(***)F	128-p (FP-1
		Low-voltage version (V_{CC} = 2.7 to 5.5 V)		HD6432653(***)TE	120-p (TFP
				HD6432653(***)F	128-p (FP-1

Note: (***) indicates the ROM code.

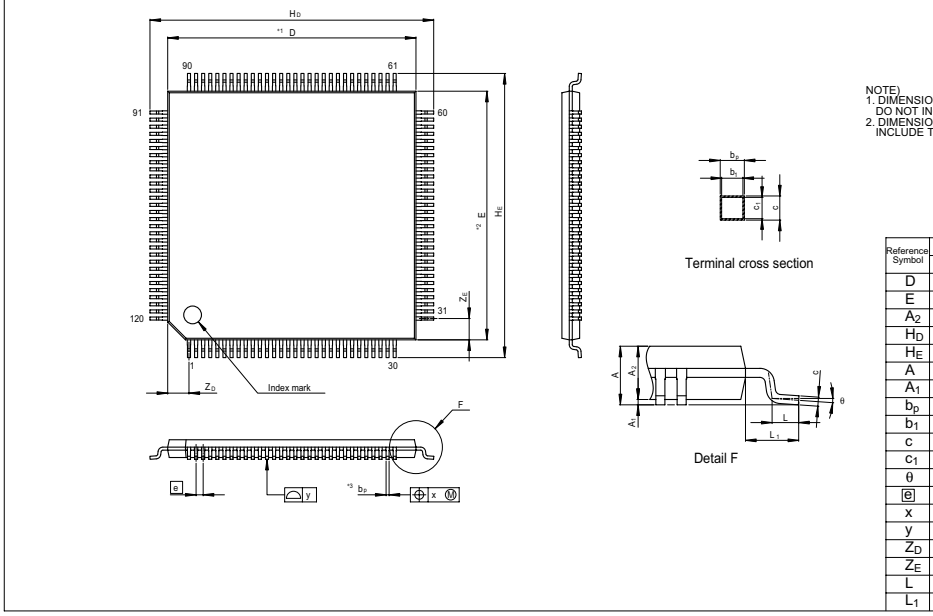
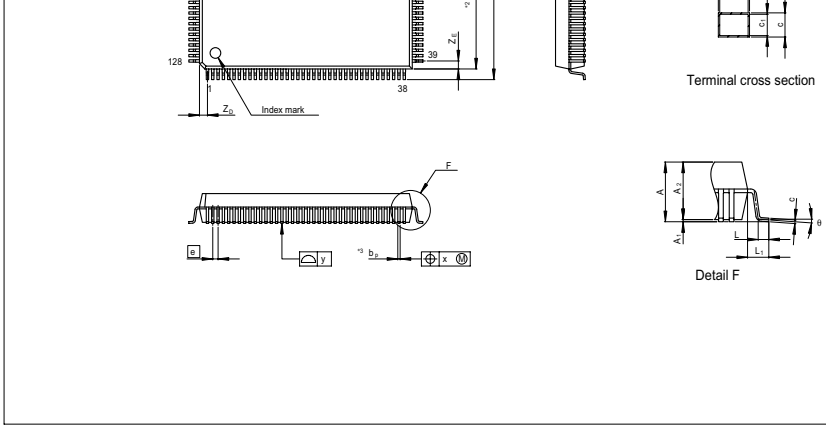


Figure G.1 TFP-120 Package Dimensions



Referen Symbo
D
E
A_2
H_D
H_E
A
A_1
b_p
b_1
c
c_1
θ
e
x
y
Z_D
Z_E
L
L_1

Figure G.2 FP-128 Package Dimensions

Renesas 16-Bit Single-Chip Microcomputer
Hardware Manual
H8S/2655 Group

Publication Date: 1st Edition, November 1995
Rev.5.00, September 14, 2006
Published by: Sales Strategic Planning Div.
Renesas Technology Corp.
Edited by: Customer Support Department
Global Strategic Communication Div.
Renesas Solutions Corp.

RENESAS SALES OFFICES

Refer to "**http://www.renesas.com/en/network**" for the latest and detailed information.

Renesas Technology America, Inc.
450 Holger Way, San Jose, CA 95134-1368, U.S.A
Tel: <1> (408) 382-7500, Fax: <1> (408) 382-7501

Renesas Technology Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K.
Tel: <44> (1628) 585-100, Fax: <44> (1628) 585-900

Renesas Technology (Shanghai) Co., Ltd.
Unit 204, 205, AZIACenter, No.1233 Lujiazui Ring Rd, Pudong District, Shanghai, China 200120
Tel: <86> (21) 5877-1818, Fax: <86> (21) 6887-7898

Renesas Technology Hong Kong Ltd.
7th Floor, North Tower, World Finance Centre, Harbour City, 1 Canton Road, Tsimshatsui, Kowloon, Hong Kong
Tel: <852> 2265-6688, Fax: <852> 2730-6071

Renesas Technology Taiwan Co., Ltd.
10th Floor, No.99, Fushing North Road, Taipei, Taiwan
Tel: <886> (2) 2715-2888, Fax: <886> (2) 2713-2999

Renesas Technology Singapore Pte. Ltd.
1 Harbour Front Avenue, #06-10, Keppel Bay Tower, Singapore 098632
Tel: <65> 6213-0200, Fax: <65> 6278-8001

Renesas Technology Korea Co., Ltd.
Kukje Center Bldg. 18th Fl., 191, 2-ka, Hangang-ro, Yongsan-ku, Seoul 140-702, Korea
Tel: <82> (2) 796-3115, Fax: <82> (2) 796-2145

Renesas Technology Malaysia Sdn. Bhd
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No.18, Jalan Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehs
Tel: <603> 7955-9390, Fax: <603> 7955-9510

H8S/2655 Group
Hardware Manual

Renesas Electronics Corporation
1753, Shimonumabe, Nakahara-ku, Kawasaki-shi, Kanagawa 211-8668 Japan

REJ09B0331-0500

X-ON Electronics

Largest Supplier of Electrical and Electronic Components

Click to view similar products for 16-bit Microcontrollers - MCU ***category:***

Click to view products by Renesas ***manufacturer:***

Other Similar products are found below :

MB90F036APMC-GSE1 MB90F342CASPMC-GSE1 MB90F345CESPMC-GE1 MB90F349CAPFR-GSE1 MB90F428GCPFR-GSE1 MB90F462APFM-GE1 MB90F462APMC-G-SNE1 MB90F497GPF-GE1 MB90F546GSPFR-GE1 MB90F947APFR-GS-SPE1 MB96F346RSBPMC-GS-N2E2 MB96F683RBPMC-GSAE1 R5F11BGEAFB#30 DF3026XBL25V S912ZVFP64F1VLL R4F24268NVRFQV R5F107DEGSP#X0 R5F11B7EANA#U0 R5F21172DSP#U0 M30622F8PGP#U3C MB90092PF-G-BNDE1 MB90F335APMC1-G-SPE1 MB90F342CASPFR-GS-N2E1 MB90F345CAPFR-GSE1 MB90F543GPF-GE1 MB90F546GSPF-GE1 MB90F568PMCR-GE1 MB90F594APFR-GE1 MB90F882ASPMC-GE1 MB96F346RSAPQCR-GS-N2E2 MB96F387RSBPMC-GSE2 MB96F387RSBPMC-GS-N2E2 MB96F395RSAPMC-GSE2 MB96F623RBPMC1-GSE1 MB96F646RBPMC-GSE1 XE167F96F66LACFXUMA1 MB96F696RBPMC-GSAE1 MB96F018RBPMC-GSE1 MB90F962SPMCR-GE1 MB90F867ASPFR-GE1 MB90F543GPF-G-FLE1 MB90F345CESPF-GE1 M30290FCHP#U3A DF2239FA20IV HD64F3672FPV R5F104AEASP#V0 R5F100BCANA#U0 R5F100BFANA#U0 S9S12H256J2VFVER R5F100ACASP#V0