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

H8/3028, H8/3028F-ZTAT™ Group

Hardware Manual

Renesas 16-Bit Single-Chip Microcomputer
H8 Family/H8/300H Series

Renesas Electronics
www.renesas.com

Rev.2.00 2003.09

Renesas 16-Bit Single-Chip Microcomp
H8 Family/H8/300H Series

H8/3028,
H8/3028F-ZTAT™ Grou

Hardware Manual

RENESAS

REJ09B00

Notes regarding these materials

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

The on-chip supporting functions include ROM, RAM, 16-bit timers, 8-bit timers, a programmable timing pattern controller (TPC), a watchdog timer (WDT), a serial com interface (SCI), an A/D converter, a D/A converter, I/O ports, a DMA controller (DM other facilities. The three-channel SCI has been expanded to support the ISO/IEC7816 card interface. Functions have also been added to reduce power consumption in batter applications: individual modules can be placed in standby, and the frequency of the sy supplied to the chip can be divided down under software control.

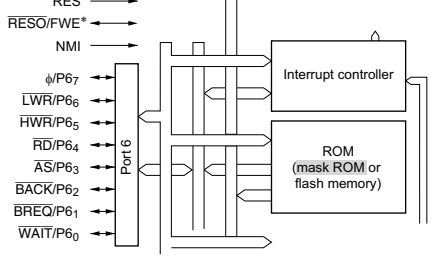
The address space is divided into eight areas. The data bus width and access cycle leng selected independently in each area, simplifying the connection of different types of m Seven MCU operating modes (modes 1 to 7) are provided, offering a choice of data bu address space size.

With these features, the H8/3028 Group offers easy implementation of compact, high- systems.

Versions with either flash memory (F-ZTAT™*) or mask ROM as the on-chip ROM a This enables users to respond quickly and flexibly to changing application specificatio initial production stage through full-scale volume production.

This manual describes the H8/3028 Group hardware. For details of the instruction set, H8/300H Series Programming Manual.

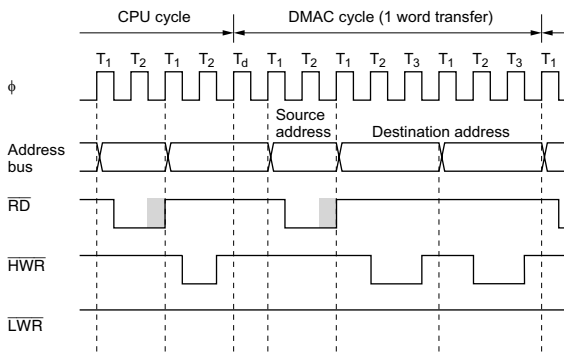
Note: * F-ZTAT™ (Flexible ZTAT) is a trademark of Renesas Technology Corp.



1.3.2 Pin Functions Table 1.2 Pin Functions	9	Table amended

System control	$\overline{\text{RES}}$	63	Input	**Reset input:** When this pin resets the c
	$\overline{\text{RESO}}$	10	Output	**Reset output (mas version):** Outputs t generated by the wa to an external devic
	FWE	10	Input	**Write enable signa version):** Flash me control signal

7.4.8 DMAC Bus Cycle Figure 7.13 DMA Transfer Bus Timing (Example)	229	Figure amended



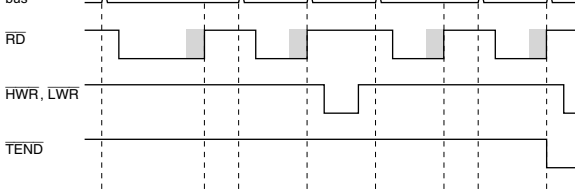


Figure 7.15 Burst DMA Bus Timing

231 Figure amended

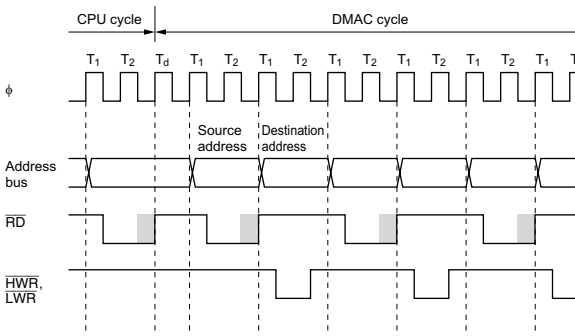
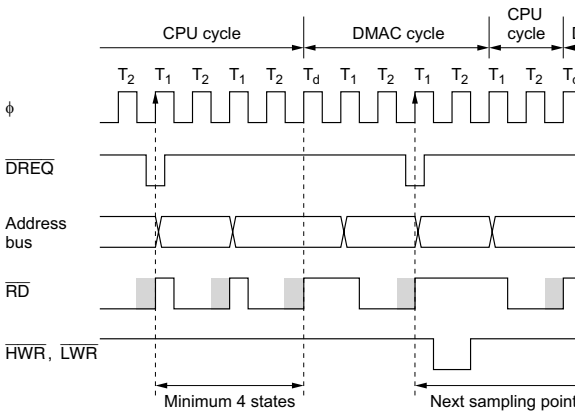


Figure 7.16 Timing of DMAC Activation by Falling Edge of DREQ in Normal Mode

232 Figure amended



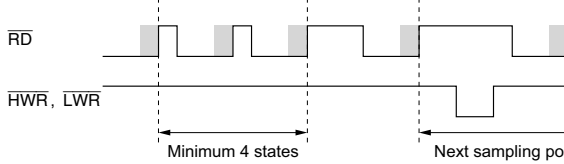
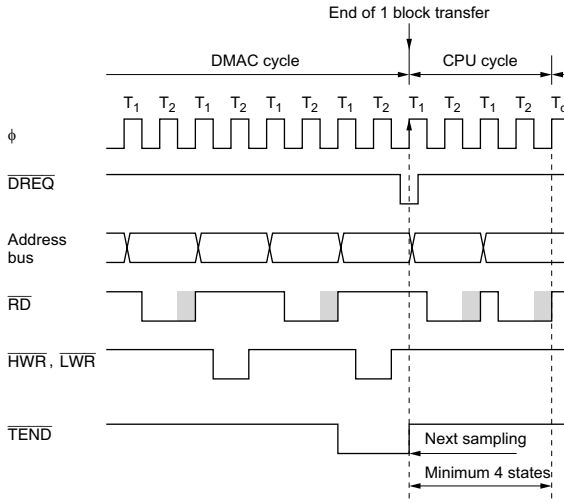


Figure 7.18 Timing of DMAC Activation by Falling Edge of DREQ in Block Transfer Mode

234

Figure amended

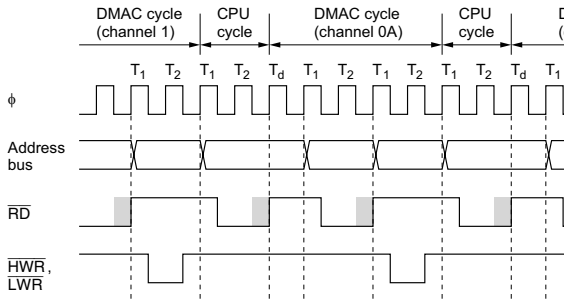


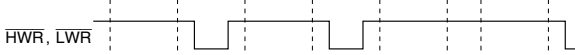
7.4.9 Multiple-Channel Operation

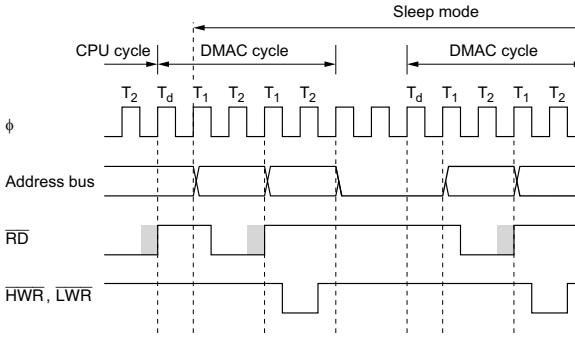
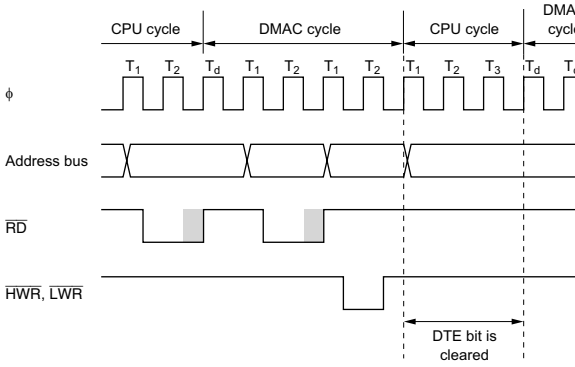
Figure 7.19 Timing of Multiple-Channel Operations

236

Figure amended





7.4.14 DMAC States in Reset State, Standby Modes, and Sleep Mode Figure 7.24 Timing of Cycle-Steal Transfer in Sleep Mode	240	Figure amended 
7.6.8 Bus Cycle when Transfer is Aborted Figure 7.27 Bus Timing at Abort of DMA Transfer in Cycle-Steal Mode	245	Figure amended 
12.2.3 Reset Control/ Status Register (RSTCSR) Bit 7—Watchdog Timer Reset (WRST)	442	Description amended At the same time, if the RSTOE bit is set to 1, the rese output from the RESO pin as low-level output to an ex device, making it possible to reset the entire system.

600	2	80	0.47
1200	1	162	–0.15
2400	1	80	0.47
4800	0	162	–0.15
9600	0	80	0.47

15.6 Usage Notes — 556

Note amended

Table 15.5 Analog Input Pin Ratings

Note: * When conversion time = 134 states, V_{CC} = 3.
and $\phi \leq$ 13 MHz. For details see section 21,
Characteristics.

18.12.1 Block Diagram — 622

Figure amended

Figure 18.19 ROM Block Diagram

Even addresses

19.2.1 Connecting a Crystal Resonator — 626

Note amended

Table 19.1(1) Damping Resistance Value

Note: A crystal resonator between 2 MHz and 25 M
used. If the chip is to be operated at less tha
on-chip frequency divider should be used. (A
resonator of less than 2 MHz cannot be used

19.2.2 External Clock Input — 629

Table amended

Table 19.3 Clock Timing

		V_{CC} = 3.0 V to 3.6 V			
Item	**Symbol**	**Min**	**Max**	**Unit**	**Test**
External clock input low pulse width	t_{EXL}	0.3	0.7	t_{cyc}	$\phi \geq 5$
		60	—	ns	$\phi < 5$
External clock input high pulse width	t_{EXH}	0.3	0.7	t_{cyc}	$\phi \geq 5$
		60	—	ns	$\phi < 5$
External clock rise time	t_{EXr}	—	5	ns	Figur
External clock fall time	t_{EXf}	—	5	ns	
Clock low pulse width	t_{CL}	0.4	0.6	t_{cyc}	$\phi \geq 5$
		80	—	ns	$\phi < 5$
Clock high pulse width	t_{CH}	0.4	0.6	t_{cyc}	$\phi \geq 5$
		80	—	ns	$\phi < 5$
External clock output settling delay time	t_{DEXT}*	500	—	µs	Figur

21.1.2 DC Characteristics Table 21.2 DC Characteristics	648, 649	Conditions amended Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC}*1 = 3.0 V to 3.6 V, V to AV_{CC}, V_{SS} = AV_{SS}*1 = 0 V, T_a = –20°C to +75° specifications), T_a = –40°C to +85°C (wide-rang specifications) FWE and *4 deleted Input high voltage — $\overline{STBY}$, $\overline{RES}$, NMI, MD_2 to MD_0 Input low voltage — $\overline{STBY}$, $\overline{RES}$, MD_2 to MD_0 Input leakage current — $\overline{STBY}$, $\overline{RES}$, NMI, MD_2 to MD_0
Table 21.3 Permissible Output Currents	650	Conditions amended Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{REF} AV_{CC}, V_{SS} = AV_{SS} = 0 V, T_a = –20°C to +75°C (r specifications), T_a = –40°C to +85°C (wide-rang specifications)
21.1.3 AC Characteristics Table 21.4 Clock Timing	652	Conditions amended Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{REF} AV_{CC}, V_{SS} = AV_{SS} = 0 V, T_a = –20°C to +75°C (r specifications), T_a = –40°C to +85°C (wide-rang specifications)

Table 21.6 Bus Timing 654, 655

Conditions amended

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_R
AV_{CC}, V_{SS} = AV_{SS} = 0 V, T_a = –20°C to +75°C
specifications), T_a = –40°C to +85°C (wide-ran
specifications)

Table amended

$\overline{RAS}$ precharge time	t_{RP}	1.5 t_{cyc} – 25	—	ns	Fi fig
$\overline{CAS}$ precharge time	t_{CP}	0.5 t_{cyc} – 15	—	ns	
Row address hold time	t_{RAH}	0.5 t_{cyc} – 15	—	ns	
Signal rise time (all input pins except EXTAL)	t_{SR}	—	100	ns	Fi
Signal fall time (all input pins except EXTAL)	t_{SF}	—	100	ns	

Table 21.7 Timing of On-Chip Supporting Modules 656

Conditions amended

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_R
AV_{CC}, V_{SS} = AV_{SS} = 0 V, T_a = –20°C to +75°C
specifications), T_a = –40°C to +85°C (wide-ran
specifications)

21.1.5 D/A Conversion Characterisitcs

Table 21.9 D/A Conversion Characteristics

659

Conditions amended

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{REF} AV_{CC}, V_{SS} = AV_{SS} = 0 V, T_a = –20°C to +75°C (r specifications), T_a = –40°C to +85°C (wide-rang specifications)

21.2.1 Absolute Maximum Ratings

Table 21.10 Absolute Maximum Ratings

660

Table and note amended

Operating temperature	T_{opr}	Regular specifications: –20 to +75*2
		Wide-range specificatio –40 to +85*2

2. The operating temperature range when programming and flash memory is: T_a = 0 to +75°C (regular specifications), +85°C (wide-range specifications).

21.2.2 DC Characteristics

Table 21.11 DC Characteristics

661, 662

Conditions amended

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC}*1 = 3.0 V to 3.6 V, V to AV_{CC}, V_{SS} = AV_{SS}*1 = 0 V, T_a = –20°C to +75° specifications), T_a = –40°C to +85°C (wide-rang specifications)
[Programming/erasing conditions: T_a = 0°C to + specifications), T_a = 0°C to +85°C (wide-range specifications)]

Table amended

Input capacitance	FWE
	NMI
	All input pins except NMI, and FWE

Timing Table 21.14 Control Signal Timing		specifications), T_a = –40°C to +85°C (wide-ran specifications)
Table 21.15 Bus Timing	666, 667	Conditions amended Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_R AV_{CC}, V_{SS} = AV_{SS} = 0 V, T_a = –20°C to +75°C specifications), T_a = –40°C to +85°C (wide-ran specifications) Table amended Row address hold time
Table 21.16 Timing of On-Chip Supporting Modules	668	Conditions amended Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_R AV_{CC}, V_{SS} = AV_{SS} = 0 V, T_a = –20°C to +75°C specifications), T_a = –40°C to +85°C (wide-ran specifications)
21.2.4 A/D Conversion Characteristics Table 21.17 A/D Conversion Characteristics	670	Conditions amended Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_R AV_{CC}, V_{SS} = AV_{SS} = 0 V, T_a = –20°C to +75°C specifications), T_a = –40°C to +85°C (wide-ran specifications) Table amended Conversion time: 70 states* — Permissible signal-source impedance — $\phi \leq 13$ MHz —
21.2.5 D/A Conversion Characteristics Table 21.18 D/A Conversion Characteristics	671	Conditions amended Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_R AV_{CC}, V_{SS} = AV_{SS} = 0 V, T_a = –20°C to +75°C specifications), T_a = –40°C to +85°C (wide-ran specifications)

Figure 21.23 SCI Input Clock Timing		SCK_0 to SCK_2
Figure 21.24 SCI Input/Output Timing in Synchronous Mode		SCK_0 to SCK_2 TxD_0 to TxD_2 RxD_0 to RxD_2
B.2 Addresses (EMC = 0)	733	Table amended

Address (Low)	Register Name	Data Bus Width	Bit Names	
			Bit 7	Bit 6
H'EE090	TCSR*2	8	OVF	WT/$\overline{IT}$
H'EE091	TCNT*2	8		
H'EE092	—		—	—
H'EE093	RSTCSR*2	8	WRST	RSTOE

	751	Note amended Notes: 1. These registers are only used by the fla version, and are not provided in the mask ROM ve

concise, optimized instruction set designed for speed. It can address a 16-Mbyte linear space. Its instruction set is upward-compatible at the object-code level with the H8/300 enabling easy porting of software from the H8/300 Series.

The on-chip system supporting functions include ROM, RAM, a 16-bit timer, an 8-bit programmable timing pattern controller (TPC), a watchdog timer (WDT), a serial com interface (SCI), an A/D converter, a D/A converter, I/O ports, a direct memory access (DMAC), and other facilities.

The H8/3028 Group has 384 kbytes of ROM and 16 kbytes of RAM.

Seven MCU operating modes offer a choice of bus width and address space size. The (modes 1 to 7) include two single-chip modes and five expanded modes.

In addition to the mask-ROM version of the H8/3028 Group, an F-ZTAT™* version w chip flash memory that can be freely programmed and reprogrammed by the user after installed is also available. This version enables users to respond quickly and flexibly t application specifications, growing production volumes, and other conditions.

Table 1.1 summarizes the features of the H8/3028 Group.

Note: * F-ZTAT™ (Flexible ZTAT) is a trademark of Renesas Technology Corp.

	• Maximum clock rate: 25 MHz • Add/subtract: 80 ns • Multiply/divide: 560 ns 16-Mbyte address space Instruction features • 8/16/32-bit data transfer, arithmetic, and logic instructions • Signed and unsigned multiply instructions (8 bits x 8 bits, 16 bits x • Signed and unsigned divide instructions (16 bits ÷ 8 bits, 32 bits ÷ • Bit accumulator function • Bit manipulation instructions with register-indirect specification of positions
Memory	H8/3028 Group • ROM: 384 kbytes • RAM: 16 kbytes
Interrupt controller	• Seven external interrupt pins: NMI, $\overline{IRQ}_0$ to $\overline{IRQ}_5$ • 36 internal interrupts • Three selectable interrupt priority levels

	• Direct connection of burst ROM • Direct connection of up to 8-Mbyte DRAM (or DRAM interface ca as interval timer) • Bus arbitration function
DMA controller (DMAC)	Short address mode • Maximum four channels available • Selection of I/O mode, idle mode, or repeat mode • Can be activated by compare match/input capture A interrupts fr timer channels 0 to 2, conversion-end interrupts from the A/D co transmit-data-empty and receive-data-full interrupts from the SC requests Full address mode • Maximum two channels available • Selection of normal mode or block transfer mode • Can be activated by compare match/input capture A interrupts fr timer channels 0 to 2, conversion-end interrupts from the A/D co external requests, or auto-request
16-bit timer, 3 channels	• Three 16-bit timer channels, capable of processing up to six puls six pulse inputs • 16-bit timer counter (channels 0 to 2) • Two multiplexed output compare/input capture pins (channels 0 • Operation can be synchronized (channels 0 to 2) • PWM mode available (channels 0 to 2) • Phase counting mode available (channel 2) • DMAC can be activated by compare match/input capture A inter (channels 0 to 2)

	• Output data can be transferred by DMAC
Watchdog timer (WDT), 1 channel	• Reset signal can be generated by overflow • Reset signal can be output externally (not in the F-ZTAT version) • Usable as an interval timer
Serial communication interface (SCI), 3 channels	• Selection of asynchronous or synchronous mode • Full duplex: can transmit and receive simultaneously • On-chip baud-rate generator • Smart card interface functions added
A/D converter	• Resolution: 10 bits • Eight channels, with selection of single or scan mode • Variable analog conversion voltage range • Sample-and-hold function • A/D conversion can be started by an external trigger or 8-bit timer match • DMAC can be activated by an A/D conversion end interrupt
D/A converter	• Resolution: 8 bits • Two channels • D/A outputs can be sustained in software standby mode
I/O ports	• 70 input/output pins • 9 input-only pins

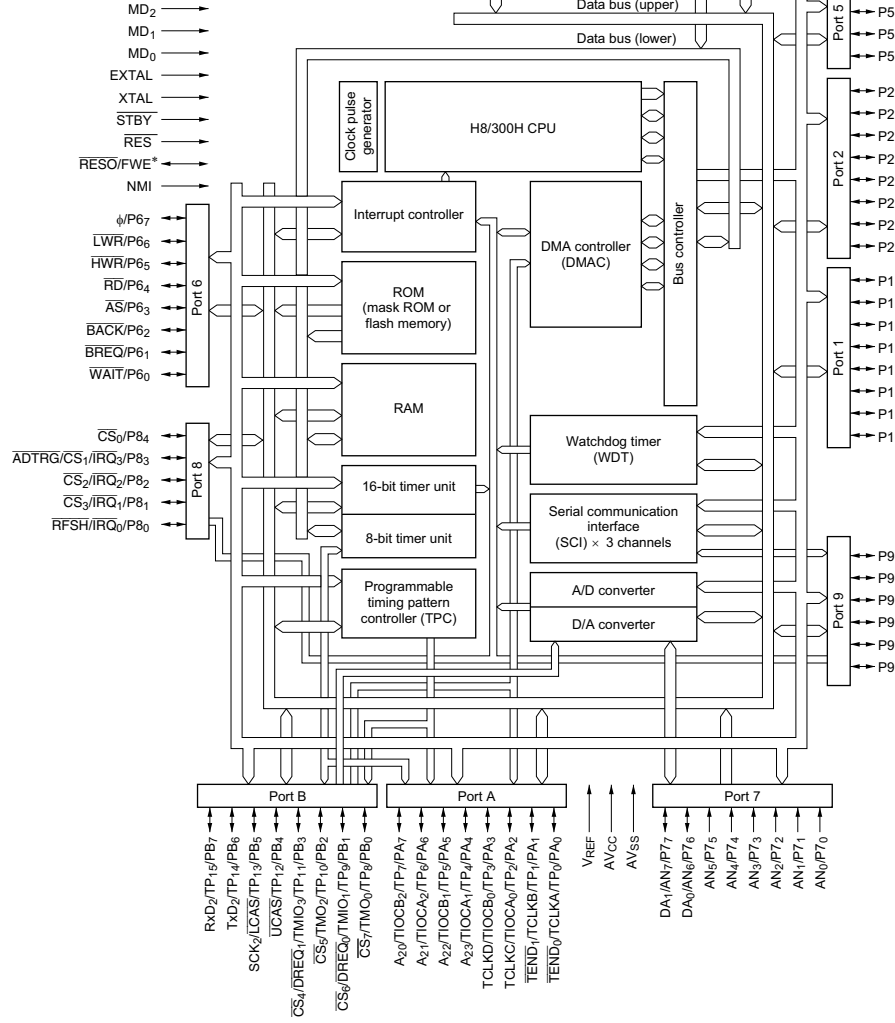
Mode 6	64 kbyte	—	—	—
Mode 7	1 Mbyte	—	—	—

Note: On-chip ROM is disabled in modes 1 to 4.

Power-down state	• Sleep mode • Software standby mode • Hardware standby mode • Module standby function • Programmable system clock frequency division
Other features	• On-chip clock pulse generator

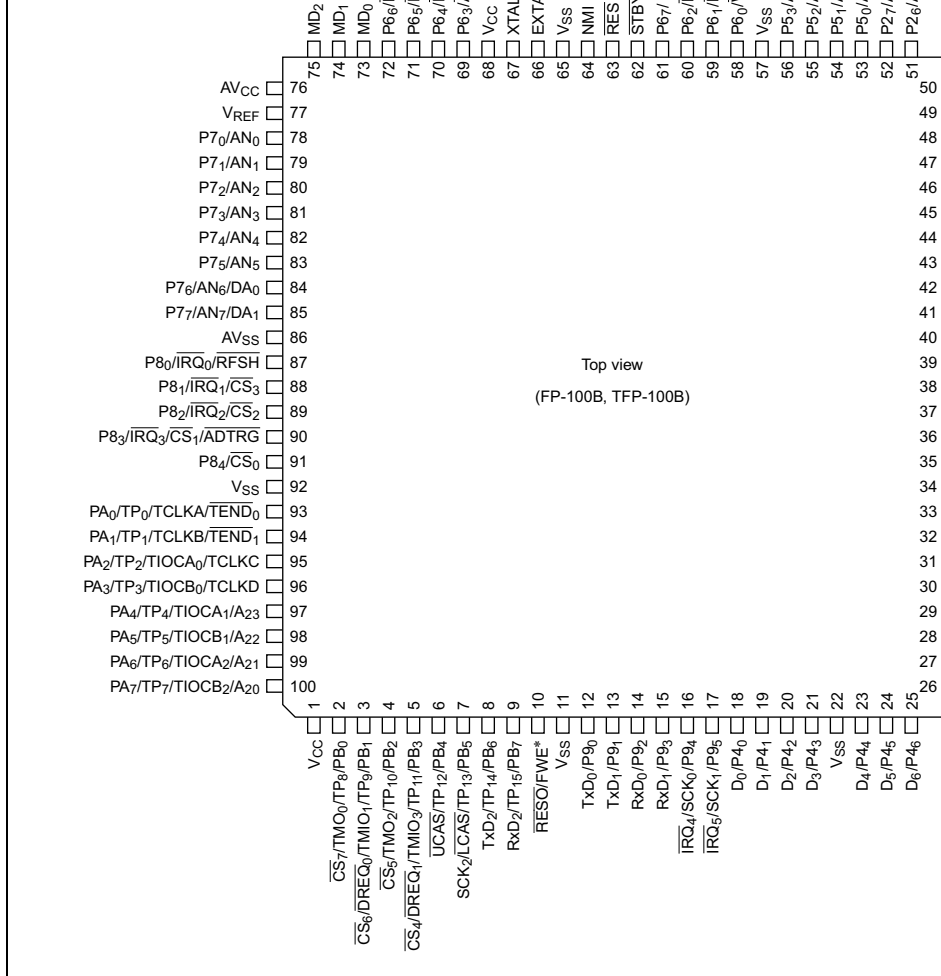
Product lineup

Product Type	Product Code	Package	ROM
H8/3028	HD64F3028F	100-pin QFP (FP-100B)	Flash mem
	HD64F3028TE	100-pin TQFP (TFP-100B)	
	HD6433028F	100-pin QFP (FP-100B)	Mask ROM
	HD6433028TE	100-pin TQFP (TFP-100B)	



Note: * Functions as $\overline{\text{RESO}}$ in mask ROM version and as FWE in flash memory version.

Figure 1.1 Block Diagram



Note: * Functions as $\overline{RESO}$ in mask ROM version and as FWE in flash memory version.

Figure 1.2 Pin Arrangement (FP-100B or TFP-100B, Top View)

Power	V_{CC}	1, 33, 68	Input	**Power:** For connection to the power Connect all V_{CC} pins to the system p supply.
	V_{SS}	11, 22, 44, 57, 65, 92	Input	**Ground:** For connection to ground (Connect all V_{SS} pins to the 0-V syste supply.
Clock	XTAL	67	Input	For connection to a crystal resonato For examples of crystal resonator ar clock input, see section 19, Clock Pu Generator.
	EXTAL	66	Input	For connection to a crystal resonato an external clock signal. For exampl resonator and external clock input, s 19, Clock Pulse Generator.
	ϕ	61	Output	**System clock:** Supplies the system external devices.
Operating mode control	MD_2 to MD_0	75 to 73	Input	**Mode 2 to mode 0:** For setting the o mode, as follows. Inputs at these pir be changed during operation.

MD_2	MD_1	MD_0	Operating Mod
0	0	0	—
0	0	1	Mode 1
0	1	0	Mode 2
0	1	1	Mode 3
1	0	0	Mode 4
1	0	1	Mode 5
1	1	0	Mode 6
1	1	1	Mode 7

				memory write control signal
	$\overline{STBY}$	62	Input	**Standby:** When driven low, this pin transition to hardware standby mod
	$\overline{BREQ}$	59	Input	**Bus request:** Used by an external request the bus right
	$\overline{BACK}$	60	Output	**Bus request acknowledge:** Indica bus has been granted to an externa
Interrupts	NMI	64	Input	**Nonmaskable interrupt:** Requests nonmaskable interrupt
	$\overline{IRQ}_5$ to $\overline{IRQ}_0$	17, 16, 90 to 87	Input	**Interrupt request 5 to 0:** Maskable request pins
Address bus	A_{23} to A_0	97 to 100, 56 to 45, 43 to 36	Output	**Address bus:** Outputs address sig
Data bus	D_{15} to D_0	34 to 23, 21 to 18	Input/ output	**Data bus:** Bidirectional data bus
Bus control	$\overline{CS}_7$ to $\overline{CS}_0$	2 to 5, 88 to 91	Output	**Chip select:** Select signals for area
	$\overline{AS}$	69	Output	**Address strobe:** Goes low to indic address output on the address bus
	$\overline{RD}$	70	Output	**Read:** Goes low to indicate reading external address space
	$\overline{HWR}$	71	Output	**High write:** Goes low to indicate w external address space; indicates v the upper data bus (D_{15} to D_8).
	$\overline{LWR}$	72	Output	**Low write:** Goes low to indicate wr external address space; indicates v the lower data bus (D_7 to D_0).
	$\overline{WAIT}$	58	Input	**Wait:** Requests insertion of wait sta cycles during access to the externa space

	$\overline{UCAS}$	8		address strobe signal for DRAM
	$\overline{LWR}$ $\overline{LCAS}$	72 7	Output	**Lower column address strobe $\overline{LCA}$** address strobe signal for DRAM
DMA controller (DMAC)	$\overline{DREQ_1}$, $\overline{DREQ_0}$	5, 3	Input	**DMA request 1 and 0:** DMAC activa requests
	$\overline{TEND_1}$, $\overline{TEND_0}$	94, 93	Output	**Transfer end 1 and 0:** These signal that the DMAC has ended a data tra
16-bit timer	TCLKD to TCLKA	96 to 93	Input	**Clock input D to A:** External clock i
	$TIOCA_2$ to $TIOCA_0$	99, 97, 95	Input/ output	**Input capture/output compare A2** GRA2 to GRA0 output compare or i or PWM output
	$TIOCB_2$ to $TIOCB_0$	100, 98, 96	Input/ output	**Input capture/output compare B2** GRB2 to GRB0 output compare or i or PWM output
8-bit timer	TMO_0, TMO_2	2, 4	Output	**Compare match output:** Compare output pins
	$TMIO_1$, $TMIO_3$	3, 5	Input/ output	**Input capture input/compare matc** Input capture input or compare matc pins
	TCLKD to TCLKA	96 to 93	Input	**Counter external clock input:** Thes an external clock to the counters.
Program-mable timing pattern controller (TPC)	TP_{15} to TP_0	9 to 2, 100 to 93	Output	**TPC output 15 to 0:** Pulse output
Serial com-munication interface (SCI)	TxD_2 to TxD_0	8, 13, 12	Output	**Transmit data (channels 0, 1, 2):** S output
	RxD_2 to RxD_0	9, 15, 14	Input	**Receive data (channels 0, 1, 2):** SC
	SCK_2 to SCK_0	7, 17, 16	Input/ output	**Serial clock (channels 0, 1, 2):** SC input/output

A/D and D/A converters	AV_{CC}	76	Input	Power supply pin for the A/D and D converters. Connect to the system when not using the A/D and D/A co
	AV_{SS}	86	Input	Ground pin for the A/D and D/A con Connect to system ground (0 V).
	V_{REF}	77	Input	Reference voltage input pin for the converters. Connect to the system when not using the A/D and D/A co
I/O ports	$P1_7$ to $P1_0$	43 to 36	Input/ output	**Port 1:** Eight input/output pins. The each pin can be selected in the por direction register (P1DDR).
	$P2_7$ to $P2_0$	52 to 45	Input/ output	**Port 2:** Eight input/output pins. The each pin can be selected in the por direction register (P2DDR).
	$P3_7$ to $P3_0$	34 to 27	Input/ output	**Port 3:** Eight input/output pins. The each pin can be selected in the por direction register (P3DDR).
	$P4_7$ to $P4_0$	26 to 23, 21 to 18	Input/ output	**Port 4:** Eight input/output pins. The each pin can be selected in the por direction register (P4DDR).
	$P5_3$ to $P5_0$	56 to 53	Input/ output	**Port 5:** Four input/output pins. The each pin can be selected in the por direction register (P5DDR).
	$P6_7$ to $P6_0$	61, 72 to 69, 60 to 58	Input/ output	**Port 6:** Eight input/output pins. The each pin can be selected in the por direction register (P6DDR).
	$P7_7$ to $P7_0$	85 to 78	Input	**Port 7:** Eight input pins
	$P8_4$ to $P8_0$	91 to 87	Input/ output	**Port 8:** Five input/output pins. The each pin can be selected in the por direction register (P8DDR).

	PB_7 to PB_0	9 to 2	Input/ output	**Port B:** Eight input/output pins. The each pin can be selected in the port direction register (PBDDR).

2	$PB_0/TP_8/TMO_0/\overline{CS_7}$	$PB_0/TP_8/TMO_0/\overline{CS_7}$	$PB_0/TP_8/TMO_0/\overline{CS_7}$	$PB_0/TP_8/TMO_0/\overline{CS_7}$	$PB_0/TP_8/TMO_0/\overline{CS_7}$	$PB_0/TP_8/TMO_0$
3	$PB_1/TP_9/TMIO_1/\overline{DREQ_0}/\overline{CS_6}$	$PB_1/TP_9/TMIO_1/\overline{DREQ_0}/\overline{CS_6}$	$PB_1/TP_9/TMIO_1/\overline{DREQ_0}/\overline{CS_6}$	$PB_1/TP_9/TMIO_1/\overline{DREQ_0}/\overline{CS_6}$	$PB_1/TP_9/TMIO_1/\overline{DREQ_0}/\overline{CS_6}$	$PB_1/TP_9/TMIO_1/\overline{DREQ_0}$
4	$PB_2/TP_{10}/TMO_2/\overline{CS_5}$	$PB_2/TP_{10}/TMO_2/\overline{CS_5}$	$PB_2/TP_{10}/TMO_2/\overline{CS_5}$	$PB_2/TP_{10}/TMO_2/\overline{CS_5}$	$PB_2/TP_{10}/TMO_2/\overline{CS_5}$	$PB_2/TP_{10}/TMO_2$
5	$PB_3/TP_{11}/TMIO_3/\overline{DREQ_1}/\overline{CS_4}$	$PB_3/TP_{11}/TMIO_3/\overline{DREQ_1}/\overline{CS_4}$	$PB_3/TP_{11}/TMIO_3/\overline{DREQ_1}/\overline{CS_4}$	$PB_3/TP_{11}/TMIO_3/\overline{DREQ_1}/\overline{CS_4}$	$PB_3/TP_{11}/TMIO_3/\overline{DREQ_1}/\overline{CS_4}$	$PB_3/TP_{11}/TMIO_3/\overline{DREQ_1}$
6	$PB_4/TP_{12}/\overline{UCAS}$	$PB_4/TP_{12}/\overline{UCAS}$	$PB_4/TP_{12}/\overline{UCAS}$	$PB_4/TP_{12}/\overline{UCAS}$	$PB_4/TP_{12}/\overline{UCAS}$	PB_4/TP_{12}
7	$PB_5/TP_{13}/\overline{LCAS}/SCK_2$	$PB_5/TP_{13}/\overline{LCAS}/SCK_2$	$PB_5/TP_{13}/\overline{LCAS}/SCK_2$	$PB_5/TP_{13}/\overline{LCAS}/SCK_2$	$PB_5/TP_{13}/\overline{LCAS}/SCK_2$	$PB_5/TP_{13}/SCK_2$
8	$PB_6/TP_{14}/TxD_2$	$PB_6/TP_{14}/TxD_2$	$PB_6/TP_{14}/TxD_2$	$PB_6/TP_{14}/TxD_2$	$PB_6/TP_{14}/TxD_2$	$PB_6/TP_{14}/TxD_2$
9	$PB_7/TP_{15}/RxD_2$	$PB_7/TP_{15}/RxD_2$	$PB_7/TP_{15}/RxD_2$	$PB_7/TP_{15}/RxD_2$	$PB_7/TP_{15}/RxD_2$	$PB_7/TP_{15}/RxD_2$
10	$\overline{RESO}/FWE$*1	$\overline{RESO}/FWE$*1	$\overline{RESO}/FWE$*1	$\overline{RESO}/FWE$*1	$\overline{RESO}/FWE$*1	$\overline{RESO}/FWE$*1
11	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}
12	$P9_0/TxD_0$	$P9_0/TxD_0$	$P9_0/TxD_0$	$P9_0/TxD_0$	$P9_0/TxD_0$	$P9_0/TxD_0$
13	$P9_1/TxD_1$	$P9_1/TxD_1$	$P9_1/TxD_1$	$P9_1/TxD_1$	$P9_1/TxD_1$	$P9_1/TxD_1$
14	$P9_2/RxD_0$	$P9_2/RxD_0$	$P9_2/RxD_0$	$P9_2/RxD_0$	$P9_2/RxD_0$	$P9_2/RxD_0$
15	$P9_3/RxD_1$	$P9_3/RxD_1$	$P9_3/RxD_1$	$P9_3/RxD_1$	$P9_3/RxD_1$	$P9_3/RxD_1$
16	$P9_4/\overline{IRQ_4}/SCK_0$	$P9_4/\overline{IRQ_4}/SCK_0$	$P9_4/\overline{IRQ_4}/SCK_0$	$P9_4/\overline{IRQ_4}/SCK_0$	$P9_4/\overline{IRQ_4}/SCK_0$	$P9_4/\overline{IRQ_4}/SCK_0$
17	$P9_5/\overline{IRQ_5}/SCK_1$	$P9_5/\overline{IRQ_5}/SCK_1$	$P9_5/\overline{IRQ_5}/SCK_1$	$P9_5/\overline{IRQ_5}/SCK_1$	$P9_5/\overline{IRQ_5}/SCK_1$	$P9_5/\overline{IRQ_5}/SCK_1$
18	$P4_0/D_0$*2	$P4_0/D_0$*3	$P4_0/D_0$*2	$P4_0/D_0$*3	$P4_0/D_0$*2	$P4_0$
19	$P4_1/D_1$*2	$P4_1/D_1$*3	$P4_1/D_1$*2	$P4_1/D_1$*3	$P4_1/D_1$*2	$P4_1$

26	$P4_7/D_7$*2	$P4_7/D_7$*3	$P4_7/D_7$*2	$P4_7/D_7$*3	$P4_7/D_7$*2	$P4_7$
27	D_8	D_8	D_8	D_8	D_8	$P3_0$
28	D_9	D_9	D_9	D_9	D_9	$P3_1$
29	D_{10}	D_{10}	D_{10}	D_{10}	D_{10}	$P3_2$
30	D_{11}	D_{11}	D_{11}	D_{11}	D_{11}	$P3_3$
31	D_{12}	D_{12}	D_{12}	D_{12}	D_{12}	$P3_4$
32	D_{13}	D_{13}	D_{13}	D_{13}	D_{13}	$P3_5$
33	D_{14}	D_{14}	D_{14}	D_{14}	D_{14}	$P3_6$
34	D_{15}	D_{15}	D_{15}	D_{15}	D_{15}	$P3_7$
35	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}
36	A_0	A_0	A_0	A_0	$P1_0/A_0$	$P1_0$
37	A_1	A_1	A_1	A_1	$P1_1/A_1$	$P1_1$
38	A_2	A_2	A_2	A_2	$P1_2/A_2$	$P1_2$
39	A_3	A_3	A_3	A_3	$P1_3/A_3$	$P1_3$
40	A_4	A_4	A_4	A_4	$P1_4/A_4$	$P1_4$
41	A_5	A_5	A_5	A_5	$P1_5/A_5$	$P1_5$
42	A_6	A_6	A_6	A_6	$P1_6/A_6$	$P1_6$
43	A_7	A_7	A_7	A_7	$P1_7/A_7$	$P1_7$
44	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}
45	A_8	A_8	A_8	A_8	$P2_0/A_8$	$P2_0$
46	A_9	A_9	A_9	A_9	$P2_1/A_9$	$P2_1$
47	A_{10}	A_{10}	A_{10}	A_{10}	$P2_2/A_{10}$	$P2_2$
48	A_{11}	A_{11}	A_{11}	A_{11}	$P2_3/A_{11}$	$P2_3$
49	A_{12}	A_{12}	A_{12}	A_{12}	$P2_4/A_{12}$	$P2_4$
50	A_{13}	A_{13}	A_{13}	A_{13}	$P2_5/A_{13}$	$P2_5$
51	A_{14}	A_{14}	A_{14}	A_{14}	$P2_6/A_{14}$	$P2_6$
52	A_{15}	A_{15}	A_{15}	A_{15}	$P2_7/A_{15}$	$P2_7$
53	A_{16}	A_{16}	A_{16}	A_{16}	$P5_0/A_{16}$	$P5_0$
54	A_{17}	A_{17}	A_{17}	A_{17}	$P5_1/A_{17}$	$P5_1$

60	$P6_2$/BACK	$P6_2$/BACK	$P6_2$/BACK	$P6_2$/BACK	$P6_2$/BACK	$P6_2$
61	ϕ	ϕ	ϕ	ϕ	$P6_7/\phi$	$P6_7/\phi$
62	$\overline{STBY}$	$\overline{STBY}$	$\overline{STBY}$	$\overline{STBY}$	$\overline{STBY}$	$\overline{STBY}$
63	$\overline{RES}$	$\overline{RES}$	$\overline{RES}$	$\overline{RES}$	$\overline{RES}$	$\overline{RES}$
64	NMI	NMI	NMI	NMI	NMI	NMI
65	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}
66	EXTAL	EXTAL	EXTAL	EXTAL	EXTAL	EXTAL
67	XTAL	XTAL	XTAL	XTAL	XTAL	XTAL
68	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}
69	$\overline{AS}$	$\overline{AS}$	$\overline{AS}$	$\overline{AS}$	$\overline{AS}$	$P6_3$
70	$\overline{RD}$	$\overline{RD}$	$\overline{RD}$	$\overline{RD}$	$\overline{RD}$	$P6_4$
71	$\overline{HWR}$	$\overline{HWR}$	$\overline{HWR}$	$\overline{HWR}$	$\overline{HWR}$	$P6_5$
72	$\overline{LWR}$	$\overline{LWR}$	$\overline{LWR}$	$\overline{LWR}$	$\overline{LWR}$	$P6_6$
73	MD_0	MD_0	MD_0	MD_0	MD_0	MD_0
74	MD_1	MD_1	MD_1	MD_1	MD_1	MD_1
75	MD_2	MD_2	MD_2	MD_2	MD_2	MD_2
76	AV_{CC}	AV_{CC}	AV_{CC}	AV_{CC}	AV_{CC}	AV_{CC}
77	V_{REF}	V_{REF}	V_{REF}	V_{REF}	V_{REF}	V_{REF}
78	$P7_0/AN_0$	$P7_0/AN_0$	$P7_0/AN_0$	$P7_0/AN_0$	$P7_0/AN_0$	$P7_0/AN_0$
79	$P7_1/AN_1$	$P7_1/AN_1$	$P7_1/AN_1$	$P7_1/AN_1$	$P7_1/AN_1$	$P7_1/AN_1$
80	$P7_2/AN_2$	$P7_2/AN_2$	$P7_2/AN_2$	$P7_2/AN_2$	$P7_2/AN_2$	$P7_2/AN_2$
81	$P7_3/AN_3$	$P7_3/AN_3$	$P7_3/AN_3$	$P7_3/AN_3$	$P7_3/AN_3$	$P7_3/AN_3$
82	$P7_4/AN_4$	$P7_4/AN_4$	$P7_4/AN_4$	$P7_4/AN_4$	$P7_4/AN_4$	$P7_4/AN_4$
83	$P7_5/AN_5$	$P7_5/AN_5$	$P7_5/AN_5$	$P7_5/AN_5$	$P7_5/AN_5$	$P7_5/AN_5$
84	$P7_6/AN_6$/ DA_0	$P7_6/AN_6$/ DA_0	$P7_6/AN_6$/ DA_0	$P7_6/AN_6$/ DA_0	$P7_6/AN_6$/ DA_0	$P7_6/AN_6$/ DA_0
85	$P7_7/AN_7$/ DA_1	$P7_7/AN_7$/ DA_1	$P7_7/AN_7$/ DA_1	$P7_7/AN_7$/ DA_1	$P7_7/AN_7$/ DA_1	$P7_7/AN_7$/ DA_1
86	AV_{SS}	AV_{SS}	AV_{SS}	AV_{SS}	AV_{SS}	AV_{SS}
87	$P8_0/\overline{IRQ_0}$/ $\overline{RFSH}$	$P8_0/\overline{IRQ_0}$/ $\overline{RFSH}$	$P8_0/\overline{IRQ_0}$/ $\overline{RFSH}$	$P8_0/\overline{IRQ_0}$/ $\overline{RFSH}$	$P8_0/\overline{IRQ_0}$/ $\overline{RFSH}$	$P8_0/\overline{IRQ_0}$

91	$P8_4/\overline{CS_0}$	$P8_4/\overline{CS_0}$	$P8_4/\overline{CS_0}$	$P8_4/\overline{CS_0}$	$P8_4/\overline{CS_0}$	$P8_4$
92	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}
93	PA_0/TP_0/ TCLKA/ $\overline{TEND_0}$	PA_0/TP_0/ TCLKA/ $\overline{TEND_0}$	PA_0/TP_0/ TCLKA/ $\overline{TEND_0}$	PA_0/TP_0/ TCLKA/ $\overline{TEND_0}$	PA_0/TP_0/ TCLKA/ $\overline{TEND_0}$	PA_0/TP_0/ TCLKA/ $\overline{TEND_0}$
94	PA_1/TP_1/ TCLKB/ $\overline{TEND_1}$	PA_1/TP_1/ TCLKB/ $\overline{TEND_1}$	PA_1/TP_1 /TCLKB/ $\overline{TEND_1}$	PA_1/TP_1/ TCLKB/ $\overline{TEND_1}$	PA_1/TP_1/ TCLKB/ $\overline{TEND_1}$	PA_1/TP_1/ TCLKB/ $\overline{TEND_1}$
95	PA_2/TP_2/ $TIOCA_0$/ TCLKC	PA_2/TP_2/ $TIOCA_0$/ TCLKC	PA_2/TP_2/ $TIOCA_0$/ TCLKC	PA_2/TP_2/ $TIOCA_0$/ TCLKC	PA_2/TP_2/ $TIOCA_0$/ TCLKC	PA_2/TP_2/ $TIOCA_0$/ TCLKC
96	PA_3/TP_3/ $TIOCB_0$/ TCLKD	PA_3/TP_3/ $TIOCB_0$/ TCLKD	PA_3/TP_3/ $TIOCB_0$/ TCLKD	PA_3/TP_3/ $TIOCB_0$/ TCLKD	PA_3/TP_3/ $TIOCB_0$/ TCLKD	PA_3/TP_3/ $TIOCB_0$/ TCLKD
97	PA_4/TP_4/ $TIOCA_1$	PA_4/TP_4/ $TIOCA_1$	PA_4/TP_4/ $TIOCA_1$/ A_{23}	PA_4/TP_4/ $TIOCA_1$/ A_{23}	PA_4/TP_4/ $TIOCA_1$/ A_{23}	PA_4/TP_4/ $TIOCA_1$
98	PA_5/TP_5/ $TIOCB_1$	PA_5/TP_5/ $TIOCB_1$	PA_5/TP_5/ $TIOCB_1$/ A_{22}	PA_5/TP_5/ $TIOCB_1$/ A_{22}	PA_5/TP_5/ $TIOCB_1$/ A_{22}	PA_5/TP_5/ $TIOCB_1$
99	PA_6/TP_6/ $TIOCA_2$	PA_6/TP_6/ $TIOCA_2$	PA_6/TP_6/ $TIOCA_2$/ A_{21}	PA_6/TP_6/ $TIOCA_2$/ A_{21}	PA_6/TP_6/ $TIOCA_2$/ A_{21}	PA_6/TP_6/ $TIOCA_2$
100	PA_7/TP_7/ $TIOCB_2$	PA_7/TP_7/ $TIOCB_2$	A_{20}	A_{20}	PA_7/TP_7/ $TIOCB_2$/ A_{20}	PA_7/TP_7/ $TIOCB_2$

Notes: 1. Functions as $\overline{RESO}$ in mask ROM version and as FWE in flash memory vers
2. In modes 1, 3, 5 the $P4_0$ to $P4_7$ functions of pins $P4_0/D_0$ to $P4_7/D_7$ are selecte reset, but they can be changed by software.
3. In modes 2 and 4 the D_0 to D_7 functions of pins $P4_0/D_0$ to $P4_7/D_7$ are selecte reset, but they can be changed by software.

2.1.1 Features

The H8/300H CPU has the following features.

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

- Two CPU operating modes
  - Normal mode
  - Advanced mode
- Low-power mode
  Transition to power-down state by SLEEP instruction

2.1.2 Differences from H8/300 CPU

In comparison to the H8/300 CPU, the H8/300H has the following enhancements.

- More general registers
  Eight 16-bit registers have been added.
- Expanded address space
  - Advanced mode supports a maximum 16-Mbyte address space.
  - Normal mode supports the same 64-kbyte address space as the H8/300 CPU.
- Enhanced addressing
  The addressing modes have been enhanced to make effective use of the 16-Mbyte a space.
- Enhanced instructions
  - Data transfer, arithmetic, and logic instructions can operate on 32-bit data.
  - Signed multiply/divide instructions and other instructions have been added.

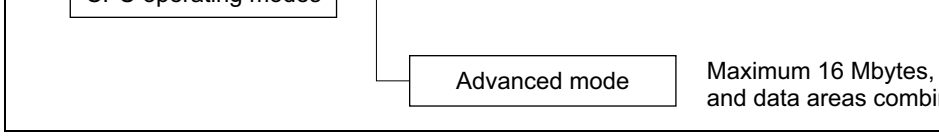


Figure 2.1 CPU Operating Modes

2.3 Address Space

Figure 2.2 shows a simple memory map for the H8/3028 Group. The H8/300H CPU c
linear address space with a maximum size of 64 kbytes in normal mode, and 16 Mbyt
advanced mode. For further details see section 3.6, Memory Map in Each Operating M

The 1-Mbyte operating modes use 20-bit addressing. The upper 4 bits of effective add
ignored.

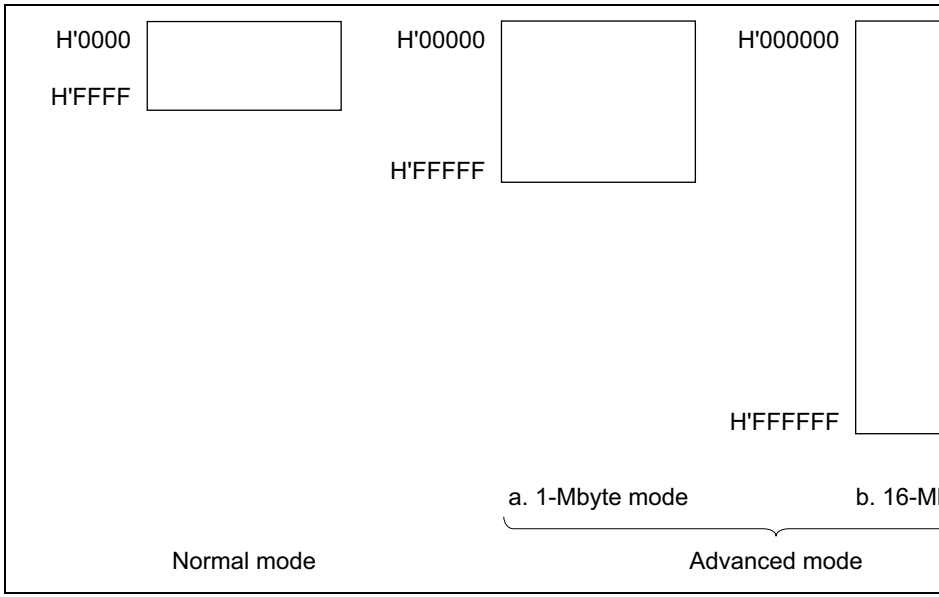


Figure 2.2 Memory Map

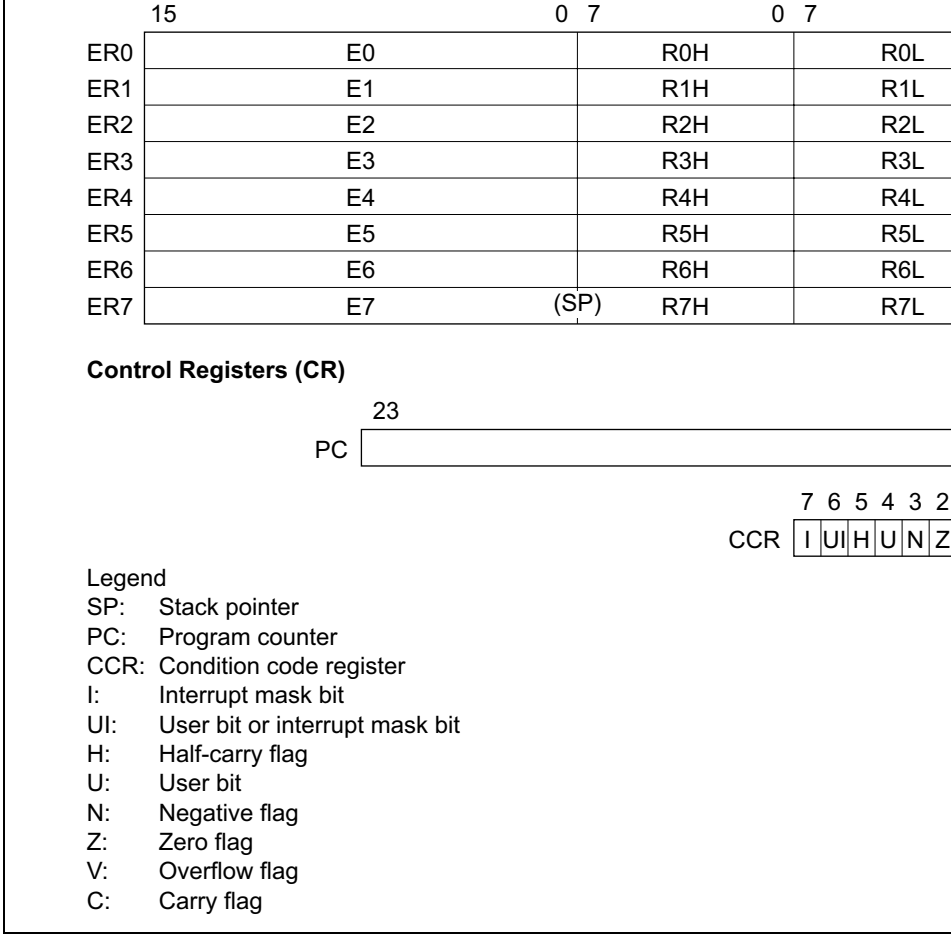


Figure 2.3 CPU Registers

(R0 to R7). These registers are functionally equivalent, providing a maximum sixteen 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 2.4 illustrates the usage of the general registers. The usage of each register can independently.

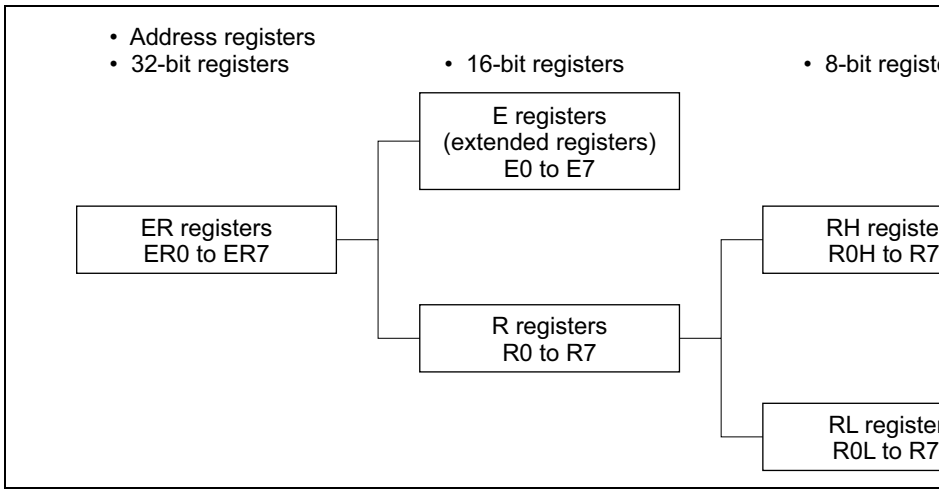


Figure 2.4 Usage of General Registers

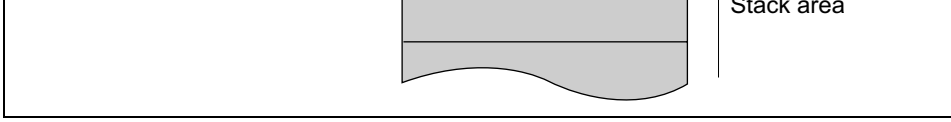


Figure 2.5 Stack

2.4.3 Control Registers

The control registers are the 24-bit program counter (PC) and the 8-bit condition code r
(CCR).

Program Counter (PC): This 24-bit counter indicates the address of the next instructi
will execute. The length of all CPU instructions is 2 bytes (one word), so the least signi
bit is ignored. When an instruction is fetched, the least significant PC bit is regarded as

Condition Code Register (CCR): This 8-bit register contains internal CPU status info
including the interrupt mask bit (I) and half-carry (H), negative (N), zero (Z), overflow
carry (C) flags.

Bit 7—Interrupt Mask Bit (I): Masks interrupts other than NMI when set to 1. NMI i
regardless of the I bit setting. The I bit is set to 1 at the start of an exception-handling s

Bit 6—User Bit or Interrupt Mask Bit (UI): Can be written and read by software usi
LDC, STC, ANDC, ORC, and XORC instructions. This bit can also be used as an inter
bit. For details see 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 clea
otherwise. When the ADD.W, SUB.W, CMP.W, or NEG.W instruction is executed, the
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 car
borrow at bit 27, and cleared to 0 otherwise.

Bit 0—Carry Flag (C): Set to 1 when a carry is generated by execution of an operati
cleared to 0 otherwise. Used by:

- Add instructions, to indicate a carry
- Subtract instructions, to indicate a borrow
- Shift and rotate instructions

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

Some instructions leave flag bits unchanged. Operations can be performed on CCR by
STC, ANDC, ORC, and XORC instructions. The N, Z, V, and C flags are used by con
branch (Bcc) instructions.

For the action of each instruction on the flag bits, see appendix A.1, Instruction List. F
UI bits, see section 5, Interrupt Controller.

2.4.4 Initial CPU Register Values

In reset exception handling, PC is initialized to a value loaded from the vector table, a
in CCR is set to 1. The other CCR bits and the general registers are not initialized. In p
the initial value of the stack pointer (ER7) is also undefined. The stack pointer (ER7)
therefore be initialized by an MOV.L instruction executed immediately after a reset.

Figures 2.6 and 2.7 show the data formats in general registers.

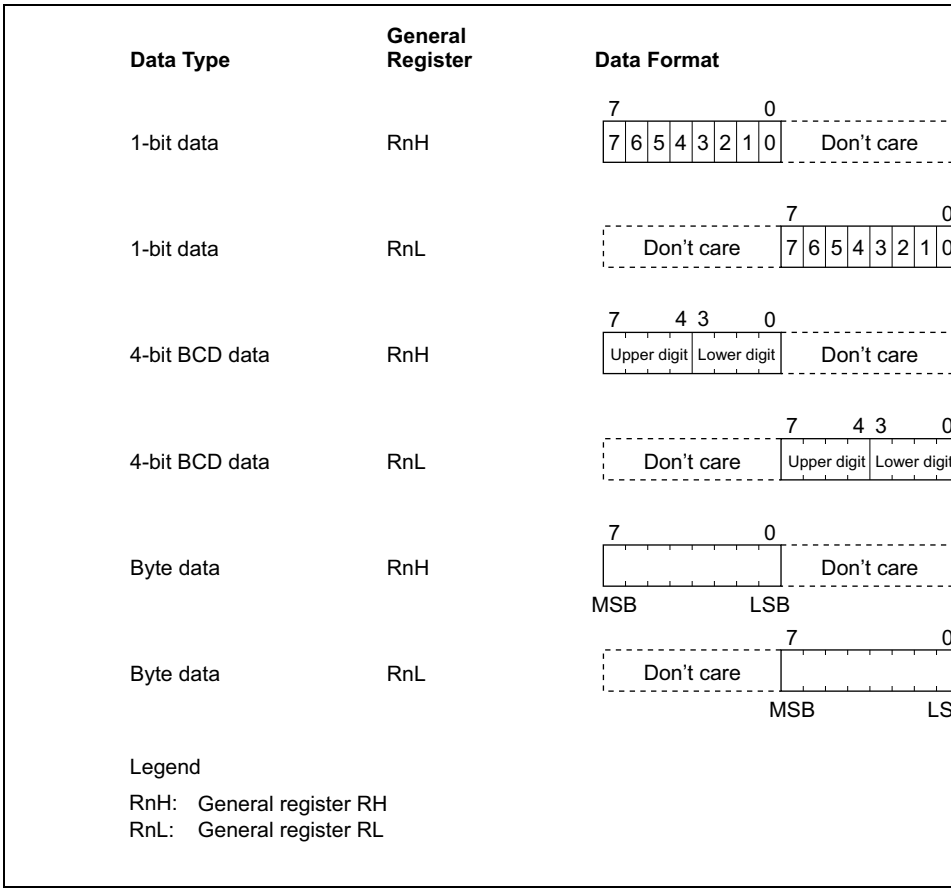


Figure 2.6 General Register Data Formats

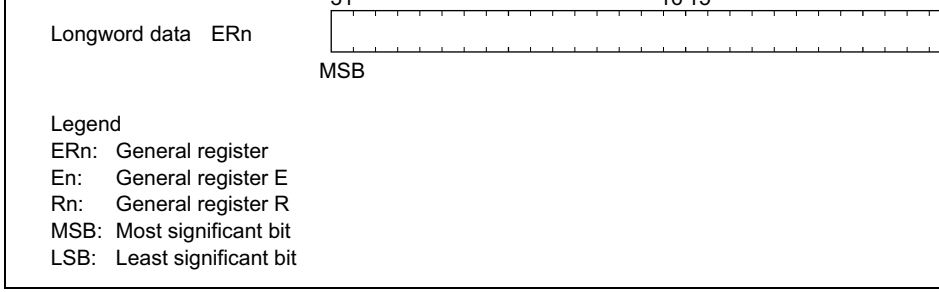


Figure 2.7 General Register Data Formats

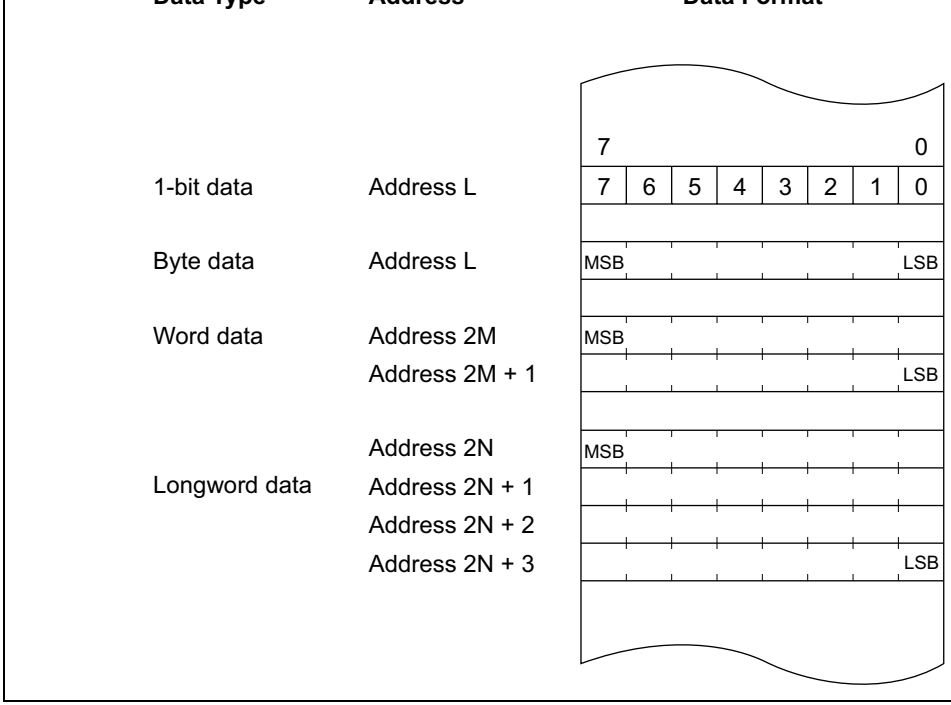


Figure 2.8 Memory Data Formats

When ER7 (SP) is used as an address register to access the stack, the operand size shou
size or longword size.

Data transfer	MOV, PUSH*1, POP*1, MOVTPE*2, MOVFPE*2
Arithmetic operations	ADD, SUB, ADDX, SUBX, INC, DEC, ADDS, SUBS, DAA, DAS MULXU, MULXS, DIVXU, DIVXS, CMP, NEG, EXTS, EXTU
Logic operations	AND, OR, XOR, NOT
Shift operations	SHAL, SHAR, SHLL, SHLR, ROTL, ROTR, ROTXL, ROTXR
Bit manipulation	BSET, BCLR, BNOT, BTST, BAND, BIAND, BOR, BIOR, BXO BIXOR, BLD, BILD, BST, BIST
Branch	Bcc*3, JMP, BSR, JSR, RTS
System control	TRAPA, RTE, SLEEP, LDC, STC, ANDC, ORC, XORC, NOP
Block data transfer	EEPMOV

T

Notes: 1. POP.W Rn is identical to MOV.W @SP+, Rn.
PUSH.W Rn is identical to MOV.W Rn, @–SP.
POP.L ERn is identical to MOV.L @SP+, Rn.
PUSH.L ERn is identical to MOV.L Rn, @–SP.
2. Not available in the H8/3028 Group.
3. Bcc is a generic branching instruction.

		#xx	Rn	@ERn	@(d:16,ERn)	@(d:32,ERn)	@ERn+/@-ERn	@aa:8	@aa:16	@aa:32	@(d:8,PC)	@(d:16,PC)
Data transfer	MOV	BWL	BWL	BWL	BWL	BWL	BWL	B	BWL	BWL	—	—
	POP, PUSH	—	—	—	—	—	—	—	—	—	—	—
	MOVFPE*, 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, MULXS, DIVXU, DIVXS	—	BW	—	—	—	—	—	—	—	—	—
	NEG	—	BWL	—	—	—	—	—	—	—	—	—
	EXTU, EXTS	—	WL	—	—	—	—	—	—	—	—	—
Logic operations	AND, OR, XOR	—	BWL	—	—	—	—	—	—	—	—	—
	NOT	—	BWL	—	—	—	—	—	—	—	—	—
Shift instructions		—	BWL	—	—	—	—	—	—	—	—	—
Bit manipulation		—	B	B	—	—	—	B	—	—	—	—
Branch	Bcc, BSR	—	—	—	—	—	—	—	—	—	—	—
	JMP, JSR	—	—	○	—	—	—	—	—	—	○	○
	RTS	—	—	—	—	—	—	—	—	○	—	—
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		—	—	—	—	—	—	—	—	—	—	—

Notes: * Not available in the H8/3028 Group
B: Byte
W: Word
L: Longword
—: No match
○: Match

Rn	General register
ERn	General register (32-bit register or address register)
(EAd)	Destination operand
(EAs)	Source operand
CCR	Condition code register
N	N (negative) flag of CCR
Z	Z (zero) flag of CCR
V	V (overflow) flag of CCR
C	C (carry) flag of CCR
PC	Program counter
SP	Stack pointer
#IMM	Immediate data
disp	Displacement
+	Addition
–	Subtraction
×	Multiplication
÷	Division
∧	AND logical
∨	OR logical
⊕	Exclusive OR logical
→	Move
¬	NOT (logical complement)
:3/:8/:16/:24	3-, 8-, 16-, or 24-bit length

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

		Cannot be used in this LSI.
POP	W/L	@SP+ → Rn Pops a general register from the stack. POP.W Rn is identical @SP+, Rn. Similarly, POP.L ERn is identical to MOV.L @SP+,
PUSH	W/L	Rn → @–SP Pushes a general register onto the stack. PUSH.W Rn is identi MOV.W Rn, @–SP. Similarly, PUSH.L ERn is identical to MOV @–SP.

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

SUBX		Performs addition or subtraction with carry or borrow on data registers, or on immediate data and data in a general register
INC, DEC	B/W/L	Rd ± 1 → Rd, Rd ± 2 → Rd Increments or decrements a general register by 1 or 2. (Byte be incremented or decremented by 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 from data in a 32-b
DAA, DAS	B	Rd decimal adjust → Rd Decimal-adjusts an addition or subtraction result in a general referring to CCR to produce 4-bit BCD data.
MULXU	B/W	Rd × Rs → Rd Performs unsigned multiplication on data in two general regis either 8 bits × 8 bits → 16 bits or 16 bits × 16 bits → 32 bits.
MULXS	B/W	Rd × Rs → Rd Performs signed multiplication on data in two general register either 8 bits × 8 bits → 16 bits or 16 bits × 16 bits → 32 bits.
DIVXU	B/W	Rd ÷ Rs → Rd Performs unsigned division on data in two general registers: e 8 bits → 8-bit quotient and 8-bit remainder or 32 bits ÷ 16 bits quotient and 16-bit remainder
DIVXS	B/W	Rd ÷ Rs → Rd Performs signed division on data in two general registers: eith bits → 8-bit quotient and 8-bit remainder, or 32 bits ÷ 16 bits quotient and 16-bit remainder
CMP	B/W/L	Rd – Rs, Rd – #IMM Compares data in a general register with data in another gene with immediate data, and sets CCR according to the result.
NEG	B/W/L	0 – Rd → Rd Takes the two's complement (arithmetic complement) of data register.

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

Table 2.5 Logic Operation Instructions

Instruction	Size*	Function
AND	B/W/L	Rd ∧ Rs → Rd, Rd ∧ #IMM → Rd Performs a logical AND operation on a general register and an general register or immediate data.
OR	B/W/L	Rd ∨ Rs → Rd, Rd ∨ #IMM → Rd Performs a logical OR operation on a general register and ano register or immediate data.
XOR	B/W/L	Rd ⊕ Rs → Rd, Rd ⊕ #IMM → Rd Performs a logical exclusive OR operation on a general registe another general register or immediate data.
NOT	B/W/L	¬ Rd → Rd Takes the one's complement (logical complement) of general r contents.

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

		Rotates general register contents.
ROTXL, ROTXR	B/W/L	Rd (rotate) → Rd Rotates general register contents, including the carry bit.

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

		number is specified by 3-bit immediate data or the lower 3 bits register.
BNOT	B	¬ (<bit-No.> of <EAd>) → (<bit-No.> of <EAd>) Inverts a specified bit in a general register or memory operand number is specified by 3-bit immediate data or the lower 3 bits register.
BTST	B	¬ (<bit-No.> of <EAd>) → Z Tests a specified bit in a general register or memory operand a clears the Z flag accordingly. The bit number is specified by 3-b immediate data or the lower 3 bits of a general register.
BAND	B	C ∧ (<bit-No.> of <EAd>) → C ANDs the carry flag with a specified bit in a general register or operand and stores the result in the carry flag.
BIAND	B	C ∧ [¬ (<bit-No.> of <EAd>)] → C ANDs the carry flag with the inverse of a specified bit in a gene or memory operand and stores the result in the carry flag. The bit number is specified by 3-bit immediate data.
BOR	B	C ∨ (<bit-No.> of <EAd>) → C ORs the carry flag with a specified bit in a general register or m operand and stores the result in the carry flag.
BIOR	B	C ∨ [¬ (<bit-No.> of <EAd>)] → C ORs the carry flag with the inverse of a specified bit in a genera memory operand and stores the result in the carry flag. The bit number is specified by 3-bit immediate data.
BXOR	B	C ⊕ (<bit-No.> of <EAd>) → C Exclusive-ORs the carry flag with a specified bit in a general re memory operand and stores the result in the carry flag.
BIXOR	B	C ⊕ [¬ (<bit-No.> of <EAd>)] → C Exclusive-ORs the carry flag with the inverse of a specified bit register or memory operand and stores the result in the carry fl The bit number is specified by 3-bit immediate data.

BST	B	C → (<bit-No.> of <EAd>) Transfers the carry flag value to a specified bit in a general re memory operand.
BIST	B	C → ¬ (<bit-No.> of <EAd>) Transfers the inverse of the carry flag value to a specified bit register or memory operand. The bit number is specified by 3-bit immediate data.

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

BLS	Low or same	C ∨ Z = 1
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 ⊕ V = 0
BLT	Less than	N ⊕ V = 1
BGT	Greater than	Z ∨ (N ⊕ V
BLE	Less or equal	Z ∨ (N ⊕ V

JMP	—	Branches unconditionally to a specified address
BSR	—	Branches to a subroutine at a specified address
JSR	—	Branches to a subroutine at a specified address
RTS	—	Returns from a subroutine

		is read by word access.
STC	B/W	CCR → (EAd) Transfers the CCR contents to a destination location. The cor register size is one byte, but in transfer to memory, data is wr access.
ANDC	B	CCR ∧ #IMM → CCR Logically ANDs the condition code register with immediate da
ORC	B	CCR ∨ #IMM → CCR Logically ORs the condition code register with immediate data
XORC	B	CCR ⊕ #IMM → CCR Logically exclusive-ORs the condition code register with imme
NOP	—	PC + 2 → PC Only increments the program counter.

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

Table 2.10 Block Transfer Instruction

Instruction	Size	Function
EEPMOV.B	—	if R4L ≠ 0 then repeat @ER5+ → @ER6+, R4L – 1 → R4L until R4L = 0 else next;
EEPMOV.W	—	if R4 ≠ 0 then repeat @ER5+ → @ER6+, R4 – 1 → R4 until R4 = 0 else next; Block transfer instruction. This instruction transfers the numbe bytes specified by R4L or R4, starting from the address indica to the location starting at the address indicated by ER6. At th transfer, the next instruction is executed.

Register Field: Specifies a general register. Address registers are specified by 3 bits, d by 3 bits or 4 bits. Some instructions have two register fields. Some have no register fie

Effective Address Extension: Eight, 16, or 32 bits specifying immediate data, an abso address, or a displacement. A 24-bit address or displacement is treated as 32-bit data in first 8 bits are 0 (H'00).

Condition Field: Specifies the branching condition of Bcc instructions.

Figure 2.9 shows examples of instruction formats.

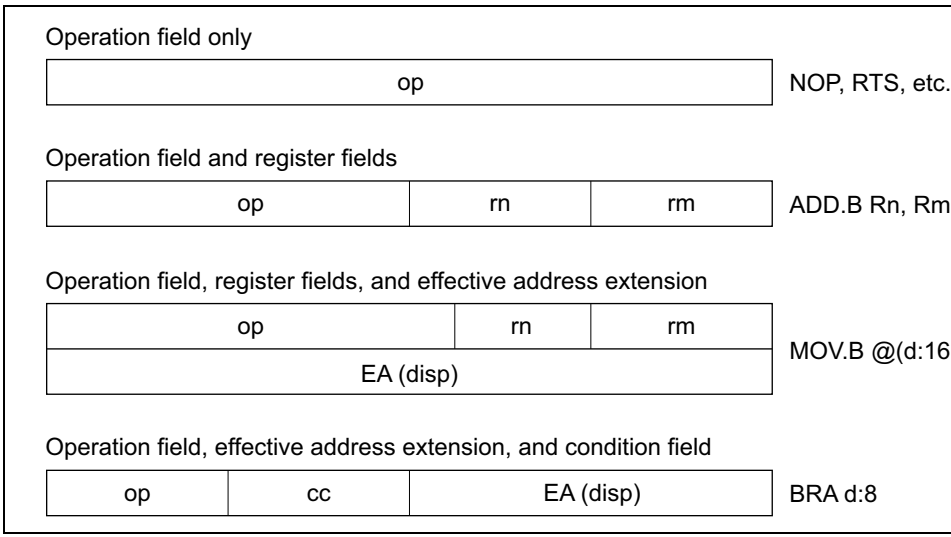


Figure 2.9 Instruction Formats

2	Modify	Modify one bit in the data byte
3	Write	Write the modified data byte back to the specified address

Example 1: BCLR is executed to clear bit 0 in the port 4 data direction register (P4DD
the following conditions.

$P4_7$, $P4_6$: Input pins
$P4_5 – P4_0$: Output pins

The intended purpose of this BCLR instruction is to switch $P4_0$ from output to input.

Before Execution of BCLR Instruction

	$P4_7$	**$P4_6$**	**$P4_5$**	**$P4_4$**	**$P4_3$**	**$P4_2$**	**$P4_1$**
Input/output	Input	Input	Output	Output	Output	Output	Output
DDR	0	0	1	1	1	1	1

Explanation: To execute the BCLR instruction, the CPU begins by reading P4DDR. S
P4DDR is a write-only register, it is read as H'FF, even though its true value is H'3F.

Next the CPU clears bit 0 of the read data, changing the value to H'FE.

Finally, the CPU writes this value (H'FE) back to P4DDR to complete the BCLR instru

As a result, $P4_0$DDR is cleared to 0, making $P4_0$ an input pin. In addition, $P4_7$DDR and
are set to 1, making $P4_7$ and $P4_6$ output pins.

The BCLR instruction can be used to clear flags in the on-chip registers to 0. In an inte
handling routine, for example, if it is known that the flag is set to 1, it is not necessary t
flag ahead of time.

indirect, or absolute (@aa:8) addressing mode to specify an operand, and register direc
BCLR, BNOT, and BTST instructions) or immediate (3-bit) addressing mode to speci
number in the operand.

Table 2.11 Addressing Modes

No.	Addressing Mode	Symbol
1	Register direct	Rn
2	Register indirect	@ERn
3	Register indirect with displacement	@(d:16, ERn)/@(d:24, ERn)
4	Register indirect with post-increment Register indirect with pre-decrement	@ERn+ @–ERn
5	Absolute address	@aa:8/@aa:16/@aa:24
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 code specifies an 8-, 16-
register containing the operand. R0H to R7H and R0L to R7L can be specified as 8-bi
R0 to R7 and E0 to E7 can be specified as 16-bit registers. ER0 to ER7 can be specifie
registers.

2 Register Indirect—@ERn: The register field of the instruction code specifies an a
register (ERn), the lower 24 bits of which contain the address of the operand.

3 Register Indirect with Displacement—@(d:16, ERn) or @(d:24, ERn): A 16-bi
displacement contained in the instruction code is added to the contents of an address r
(ERn) specified by the register field of the instruction, and the lower 24 bits of the sur
address of a memory operand. A 16-bit displacement is sign-extended when added.

The value 1, 2, or 4 is subtracted from an address register (ERn) specified by the reg in the instruction code, and the lower 24 bits of the result become the address of a m operand. The result is also stored in the address register. The value subtracted is 1 fo access, 2 for word access, or 4 for longword access. For word or longword access, t register value should be even.

5 Absolute Address—@aa:8, @aa:16, or @aa:24: The instruction code contains the address of a memory operand. The absolute address may be 8 bits long (@aa:8), 16 bits (@aa:16), or 24 bits long (@aa:24). For an 8-bit absolute address, the upper 16 bits are assumed to be 1 (H'FFFF). For a 16-bit absolute address the upper 8 bits are a sign exte 24-bit absolute address can access the entire address space. Table 2.12 indicates the acc address ranges.

Table 2.12 Absolute Address Access Ranges

Absolute Address	1-Mbyte Modes	16-Mbyte Modes
8 bits (@aa:8)	H'FFF00 to H'FFFFF (1048320 to 1048575)	H'FFFF00 to H'FFFFFF (16776960 to 16777215)
16 bits (@aa:16)	H'00000 to H'07FFF, H'F8000 to H'FFFFF (0 to 32767, 1015808 to 1048575)	H'000000 to H'007FFF, H'FF8000 to H'FFFFFF (0 to 32767, 16744448 to 1
24 bits (@aa:24)	H'00000 to H'FFFFF (0 to 1048575)	H'000000 to H'FFFFFF (0 to 16777215)

6 Immediate—#xx:8, #xx:16, or #xx:32: The instruction code contains 8-bit (#xx:8), (#xx:16), or 32-bit (#xx:32) immediate data as an operand.

The instruction codes of the ADDS, SUBS, INC, and DEC instructions contain immedi implicitly. The instruction codes of some bit manipulation instructions contain 3-bit im data specifying a bit number. The TRAPA instruction code contains 2-bit immediate da specifying a vector address.

instruction code contains an 8-bit absolute address specifying a memory operand. This
operand contains a branch address. The memory operand is accessed by longword acc
byte of the memory operand is ignored, generating a 24-bit branch address. See figure
upper bits of the 8-bit absolute address are assumed to be 0 (H'0000), so the address ra
255 (H'000000 to H'0000FF). Note that the first part of this range is also the exception
For further details see section 5, Interrupt Controller.

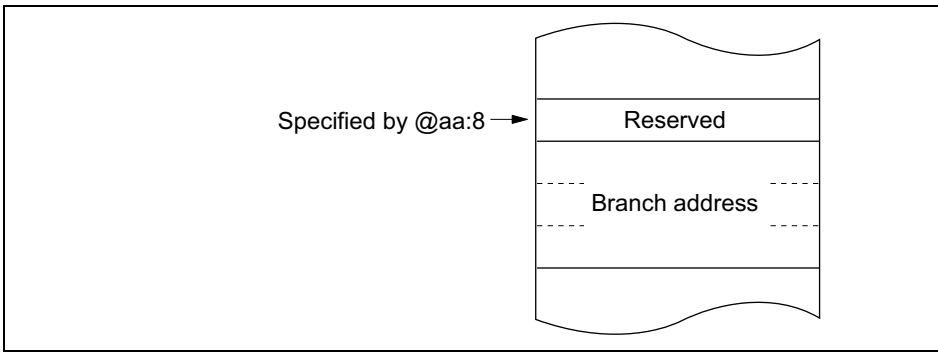


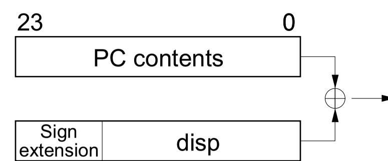
Figure 2.10 Memory-Indirect Branch Address Specification

When a word-size or longword-size memory operand is specified, or when a branch a
specified, if the specified memory address is odd, the least significant bit is regarded a
accessed data or instruction code therefore begins at the preceding address. See sectio
Memory Data Formats.

2.7.2 Effective Address Calculation

Table 2.13 explains how an effective address is calculated in each addressing mode. In
1-Mbyte operating modes the upper 4 bits of the calculated address are ignored in ord
generate a 20-bit effective address.

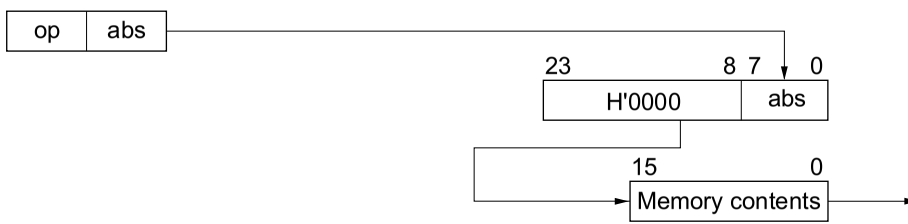
No.	Addressing Mode and Instruction Format	Effective Address Calculation	Effectiv
1	Register direct (Rn) op | rm | rn		Operand register
2	Register indirect (@ERn) op | r	31 … 0 → General register contents → 23	
3	Register indirect with displacement @(d:16, ERn)/@(d:24, ERn) op | r | |	31 … 0 → General register contents; Sign extension | disp; → 23	
4	Register indirect with post-increment or pre-decrement		
	Register indirect with post-increment @ERn+ op | r	31 … 0 → General register contents → 23; 1, 2, or 4	
	Register indirect with pre-decrement @–ERn op | r	31 … 0 → General register contents → 23; 1, 2, or 4	

No.	Addressing Mode and Instruction Format	Effective Address Calculation	E
5	Absolute address @aa:8 [op | abs] @aa:16 [op | abs] @aa:24 [op] [abs]		
6	Immediate #xx:8, #xx:16, or #xx:32 [op | IMM]		Operanc
7	Program-counter relative @(d:8, PC) or @(d:16, PC) [op | disp]		

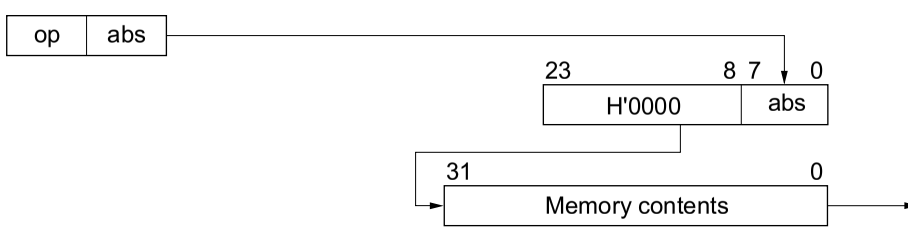
RENESAS

No.	Addressing Mode and Instruction Format	Effective Address Calculation	E
8	Memory indirect @@aa:8		

Normal mode



Advanced mode



Legend

r, rm, rn:	Register field
op:	Operation field
disp:	Displacement
IMM:	Immediate data
abs:	Absolute address

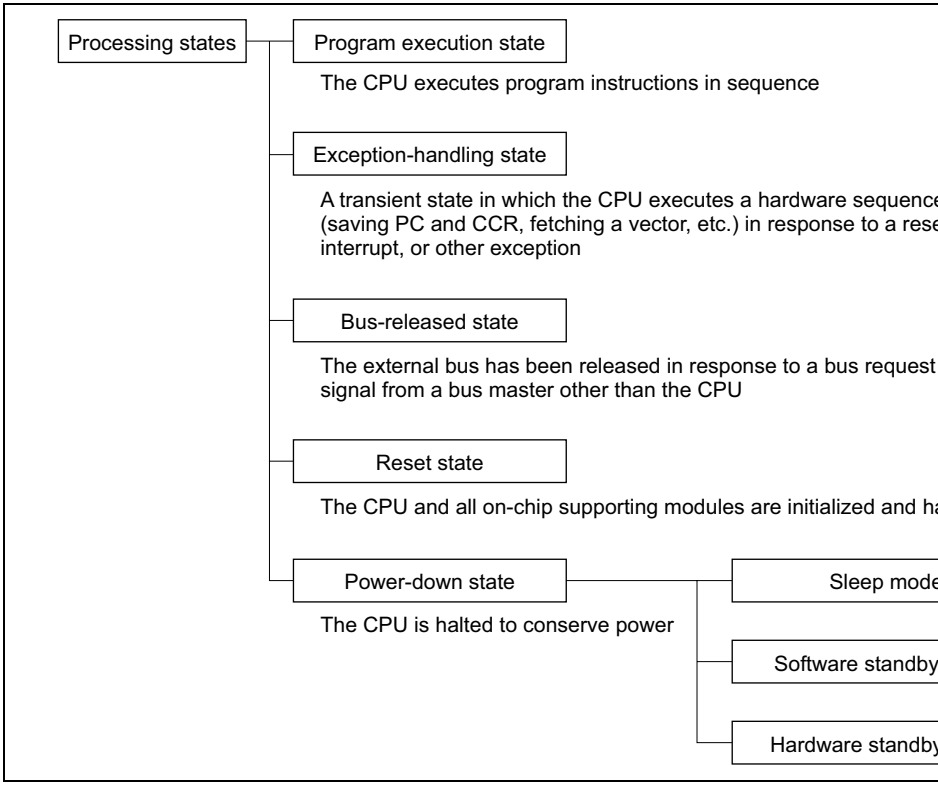


Figure 2.11 Processing States

the CPU references the stack pointer (ER7) and saves the program counter and conditio register.

Types of Exception Handling and Their Priority: Exception handling is performed f interrupts, and trap instructions. Table 2.14 indicates the types of exception handling ar priority. Trap instruction exceptions are accepted at all times in the program execution

Table 2.14 Exception Handling Types and Priority

Priority	Type of Exception	Detection Timing	Start of Exception Handl
High	Reset	Synchronized with clock	Exception handling starts when $\overline{\text{RES}}$ changes from l
	Interrupt	End of instruction execution or end of exception handling*	When an interrupt is reque exception handling starts a the current instruction or c exception-handling sequer
Low	Trap instruction	When TRAPA instruction is executed	Exception handling starts (TRAPA) instruction is exe

Note: * Interrupts are not detected at the end of the ANDC, ORC, XORC, and LDC instru immediately after reset exception handling.

Figure 2.12 classifies the exception sources. For further details about exception sources numbers, and vector addresses, see section 4, Exception Handling, and section 5, Interr Controller.

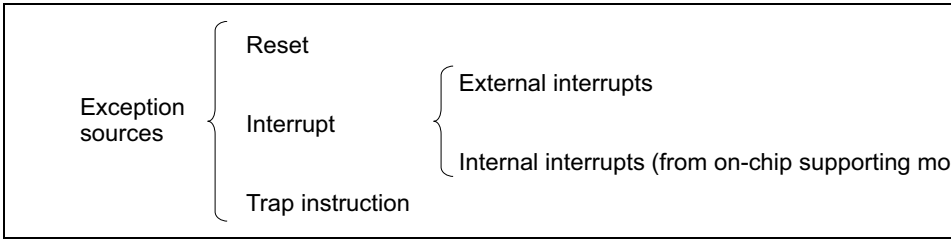
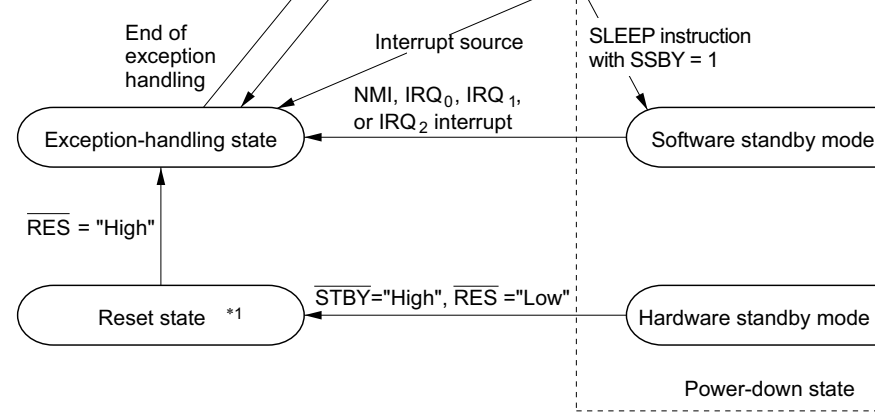


Figure 2.12 Classification of Exception Sources

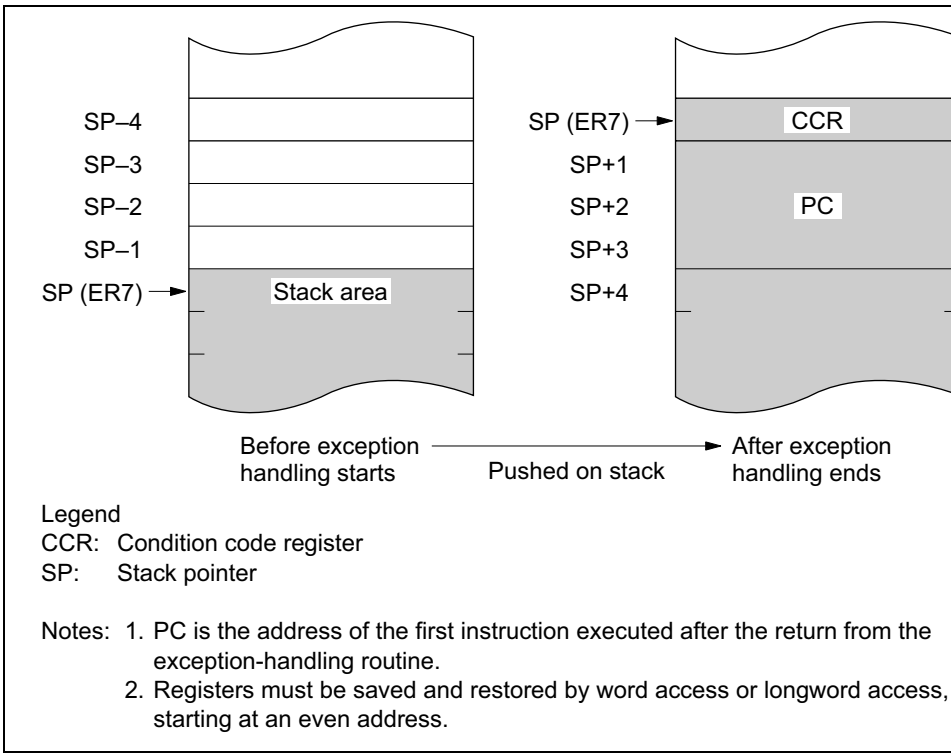


Notes: 1. From any state except hardware standby mode, a transition to the reset state occ
whenever RES goes low.
2. From any state, a transition to hardware standby mode occurs when STBY goes l

Figure 2.13 State Transitions

Interrupt Exception Handling and Trap Instruction Exception Handling: When th
exception-handling sequences begin, the CPU references the stack pointer (ER7) and pu
program counter and condition code register on the stack. Next, if the UE bit in the syst
register (SYSCR) is set to 1, the CPU sets the I bit in the condition code register to 1. If
is cleared to 0, the CPU sets both the I bit and the UI bit in the condition code register t
the CPU fetches a start address from the exception vector table and execution branches
address.

Figure 2.14 shows the stack after the exception-handling sequence.



Legend
CCR: Condition code register
SP: Stack pointer

Notes: 1. PC is the address of the first instruction executed after the return from the exception-handling routine.
2. Registers must be saved and restored by word access or longword access, starting at an even address.

Figure 2.14 Stack Structure after Exception Handling

When the $\overline{\text{RES}}$ input goes low all current processing stops and the CPU enters the rese
bit in the condition code register is set to 1 by a reset. All interrupts are masked in the
Reset exception handling starts when the $\overline{\text{RES}}$ signal changes from low to high.

The reset state can also be entered by a watchdog timer overflow. For details see secti
Watchdog Timer.

2.8.7 Power-Down State

In the power-down state the CPU stops operating to conserve power. There are three m
mode, software standby mode, and hardware standby mode.

Sleep Mode: A transition to sleep mode is made if the SLEEP instruction is executed
SSBY bit is cleared to 0 in the system control register (SYSCR). CPU operations stop
immediately after execution of the SLEEP instruction, but the contents of CPU registe
retained.

Software Standby Mode: A transition to software standby mode is made if the SLEE
instruction is executed while the SSBY bit is set to 1 in SYSCR. The CPU and clock h
on-chip supporting modules stop operating. The on-chip supporting modules are reset
as a specified voltage is supplied the contents of CPU registers and on-chip RAM are
The I/O ports also remain in their existing states.

Hardware Standby Mode: A transition to hardware standby mode is made when the
goes low. As in software standby mode, the CPU and all clocks halt and the on-chip s
modules are reset, but as long as a specified voltage is supplied, on-chip RAM conten
retained.

For further information see section 20, Power-Down State.

2.9.2 On-Chip Memory Access Timing

On-chip memory is accessed in two states. The data bus is 16 bits wide, permitting both
word access. Figure 2.15 shows the on-chip memory access cycle. Figure 2.16 indicate
states.

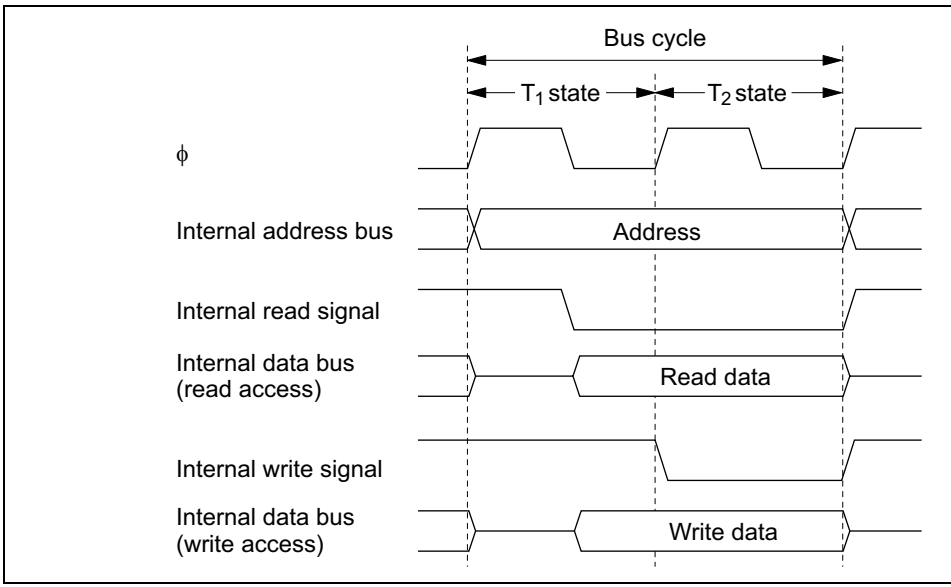


Figure 2.15 On-Chip Memory Access Cycle

Figure 2.16 Pin States during On-Chip Memory Access

2.9.3 On-Chip Supporting Module Access Timing

The on-chip supporting modules are accessed in three states. The data bus is 8 or 16 b
depending on the internal I/O register being accessed. Figure 2.17 shows the on-chip s
module access timing. Figure 2.18 indicates the pin states.

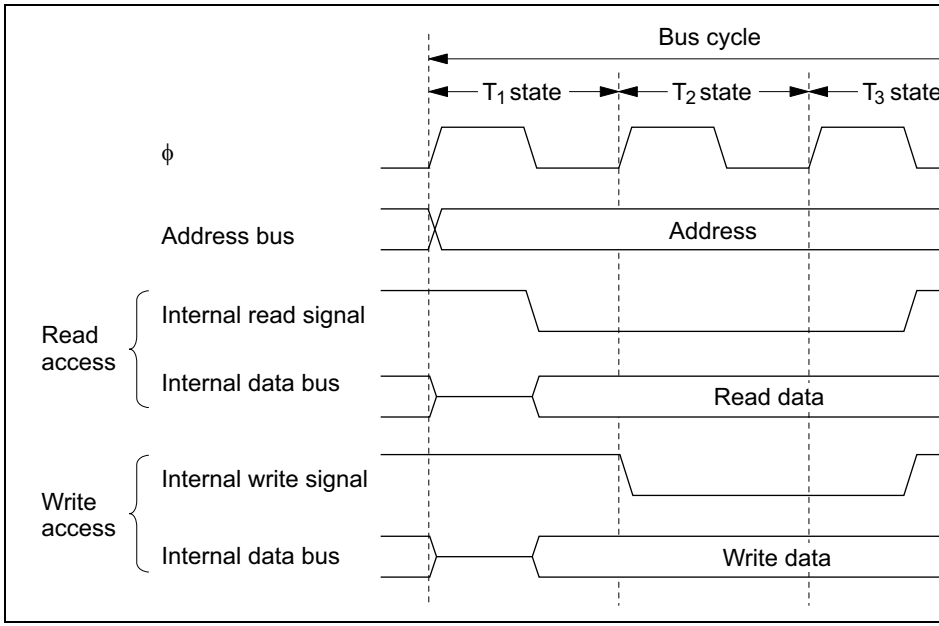


Figure 2.17 Access Cycle for On-Chip Supporting Modules

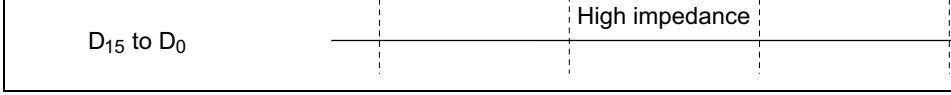


Figure 2.18 Pin States during Access to On-Chip Supporting Modules

2.9.4 Access to External Address Space

The external address space is divided into eight areas (areas 0 to 7). Bus-controller setti
determine whether each area is accessed via an 8-bit or 16-bit bus, and whether it is acc
two or three states. For details see section 6, Bus Controller.

(MD_2 to MD_0) as indicated in table 3.1. The input at these pins determines the size of space and the initial bus mode.

Table 3.1 Operating Mode Selection

Operating Mode	Mode Pins			Description		
	MD_2	MD_1	MD_0	Address Space	Initial Bus Mode*1	On-Chip ROM
—	0	0	0	—	—	—
Mode 1	0	0	1	Expanded mode	8 bits	Disabled
Mode 2	0	1	0	Expanded mode	16 bits	Disabled
Mode 3	0	1	1	Expanded mode	8 bits	Disabled
Mode 4	1	0	0	Expanded mode	16 bits	Disabled
Mode 5	1	0	1	Expanded mode	8 bits	Enabled
Mode 6	1	1	0	Single-chip normal mode	—	Enabled
Mode 7	1	1	1	Single-chip advanced mode	—	Enabled

Notes: 1. In modes 1 to 5, an 8-bit or 16-bit data bus can be selected on a per-area b settings made in the area bus width control register (ABWCR). For details s section 6, Bus Controller.

2. If the RAME bit in SYSCR is cleared to 0, these addresses become externa

For the address space size there are three choices: 64 kbytes, 1 Mbyte, or 16 Mbyte.Th data bus is either 8 or 16 bits wide depending on ABWCR settings. If 8-bit access is s all areas, 8-bit bus mode is used. For details see section 6, Bus Controller.

Modes 1 to 4 are externally expanded modes that enable access to external memory an devices and disable access to the on-chip ROM. Modes 1 and 2 support a maximum ad of 1 Mbyte. Modes 3 and 4 support a maximum address space of 16 Mbytes.

of these seven modes. The inputs at the mode pins must not be changed during operatio

3.1.2 Register Configuration

The H8/3028 Group has a mode control register (MDCR) that indicates the inputs at th pins (MD_2 to MD_0), and a system control register (SYSCR). Table 3.2 summarizes thes

Table 3.2 Registers

Address*	Name	Abbreviation	R/W	Initial Valu
H'EE011	Mode control register	MDCR	R	Undetermin
H'EE012	System control register	SYSCR	R/W	H'09

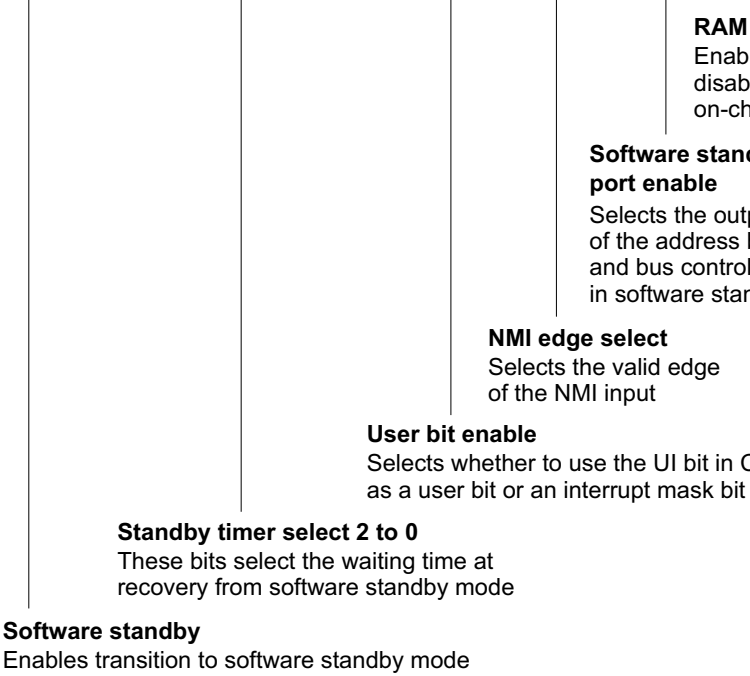
Note: * Lower 20 bits of the address in advanced mode.

Note: * Determined by pins MD_2 to MD_0.

Bits 7 and 6—Reserved: These bits can not be modified and are always read as 1.

Bits 5 to 3—Reserved: These bits can not be modified and are always read as 0.

Bits 2 to 0—Mode Select 2 to 0 (MDS2 to MDS0): These bits indicate the logic leve
MD_2 to MD_0 (the current operating mode). MDS2 to MDS0 correspond to MD_2 to MI
MDS0 are read-only bits. The mode pin (MD_2 to MD_0) levels are latched into these bi
MDCR is read.



Bit 7—Software Standby (SSBY): Enables transition to software standby mode. (For information about software standby mode see section 20, Power-Down State.)

When software standby mode is exited by an external interrupt, this bit remains set to 1 this bit, write 0.

Bit 7 SSBY	Description	
0	SLEEP instruction causes transition to sleep mode	(In
1	SLEEP instruction causes transition to software standby mode	

Bit 6 STS2	Bit 5 STS1	Bit 4 STS0	Description	
0	0	0	Waiting time = 8,192 states	(I
0	0	1	Waiting time = 16,384 states	
0	1	0	Waiting time = 32,768 states	
0	1	1	Waiting time = 65,536 states	
1	0	0	Waiting time = 131,072 states	
1	0	1	Waiting time = 262,144 states	
1	1	0	Waiting time = 1,024 states	
1	1	1	Illegal setting	

Bit 3—User Bit Enable (UE): Selects whether to use the UI bit in the condition code user bit or an interrupt mask bit.

Bit 3 UE	Description	
0	UI bit in CCR is used as an interrupt mask bit	
1	UI bit in CCR is used as a user bit	(I

Bit 2—NMI Edge Select (NMIEG): Selects the valid edge of the NMI input.

Bit 2 NMIEG	Description	
0	An interrupt is requested at the falling edge of NMI	(I
1	An interrupt is requested at the rising edge of NMI	

	signals are fixed high

Bit 0—RAM Enable (RAME): Enables or disables the on-chip RAM. The RAME bit initialized by the rising edge of the $\overline{RES}$ signal. It is not initialized in software standby

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

3.4 Operating Mode Descriptions

3.4.1 Mode 1

Ports 1, 2, and 5 function as address pins A_{19} to A_0, permitting access to a maximum 1- address space. The initial bus mode after a reset is 8 bits, with 8-bit access to all areas. one area is designated for 16-bit access in ABWCR, the bus mode switches to 16 bits.

3.4.2 Mode 2

Ports 1, 2, and 5 function as address pins A_{19} to A_0, permitting access to a maximum 1- address space. The initial bus mode after a reset is 16 bits, with 16-bit access to all area areas are designated for 8-bit access in ABWCR, the bus mode switches to 8 bits.

3.4.3 Mode 3

Ports 1, 2, and 5 and part of port A function as address pins A_{23} to A_0, permitting acces maximum 16-Mbyte address space. The initial bus mode after a reset is 8 bits, with 8-b all areas. If at least one area is designated for 16-bit access in ABWCR, the bus mode s 16 bits. A_{23} to A_{21} are valid when 0 is written in bits 7 to 5 of the bus release control re (BRCR). (In this mode A_{20} is always used for address output.)

Ports 1, 2, and 5 and part of port A can function as address pins A_{23} to A_0, permitting a maximum 16-Mbyte address space, but following a reset they are input ports. To use p and 5 as an address bus, the corresponding bits in their data direction registers (P1DD and P5DDR) must be set to 1. For A_{23} to A_{20} output, write 0 in bits 7 to 4 of BRCR. T mode after a reset is 8 bits, with 8-bit access to all areas. If at least one area is designa bit access in ABWCR, the bus mode switches to 16 bits.

3.4.6 Mode 6

This mode operates using the on-chip ROM, RAM, and registers. All I/O ports are av Mode 6 supports a maximum address space of 64 kbytes.

3.4.7 Mode 7

This mode operates using the on-chip ROM, RAM, and registers. All I/O ports are ava Mode 7 supports a 1-Mbyte address space.

Port 2	A_{15} to A_8	A_{15} to A_8	A_{15} to A_8	A_{15} to A_8	$P2_7$ to $P2_0$	$P2_7$ to $P2_0$
Port 3	D_{15} to D_8	D_{15} to D_8	D_{15} to D_8	D_{15} to D_8	D_{15} to D_8	$P3_7$ to $P3_0$
Port 4	$P4_7$ to $P4_0$ *1	D_7 to D_0 *1	$P4_7$ to $P4_0$ *1	D_7 to D_0 *1	$P4_7$ to $P4_0$ *1	$P4_7$ to $P4_0$
Port 5	A_{19} to A_{16}	A_{19} to A_{16}	A_{19} to A_{16}	A_{19} to A_{16}	$P5_3$ to $P5_0$ *2	$P5_3$ to $P5_0$
Port A	PA_7 to PA_4	PA_7 to PA_4	PA_6 to PA_4, A_{20} *3	PA_6 to PA_4, A_{20} *3	PA_7 to PA_4 *4	PA_7 to PA_4

Notes: 1. Initial state. The bus mode can be switched by settings in ABWCR. These pi
as $P4_7$ to $P4_0$ in 8-bit bus mode, and as D_7 to D_0 in 16-bit bus mode.
2. Initial state. These pins become address output pins when the correspondin
data direction registers (P1DDR, P2DDR, P5DDR) are set to 1.
3. Initial state. A_{20} is always an address output pin. PA_6 to PA_4 are switched ov
A_{21} output by writing 0 in bits 7 to 5 of BRCR.
4. Initial state. PA_7 to PA_4 are switched over to A_{23} to A_{20} output by writing 0 in
of BRCR.

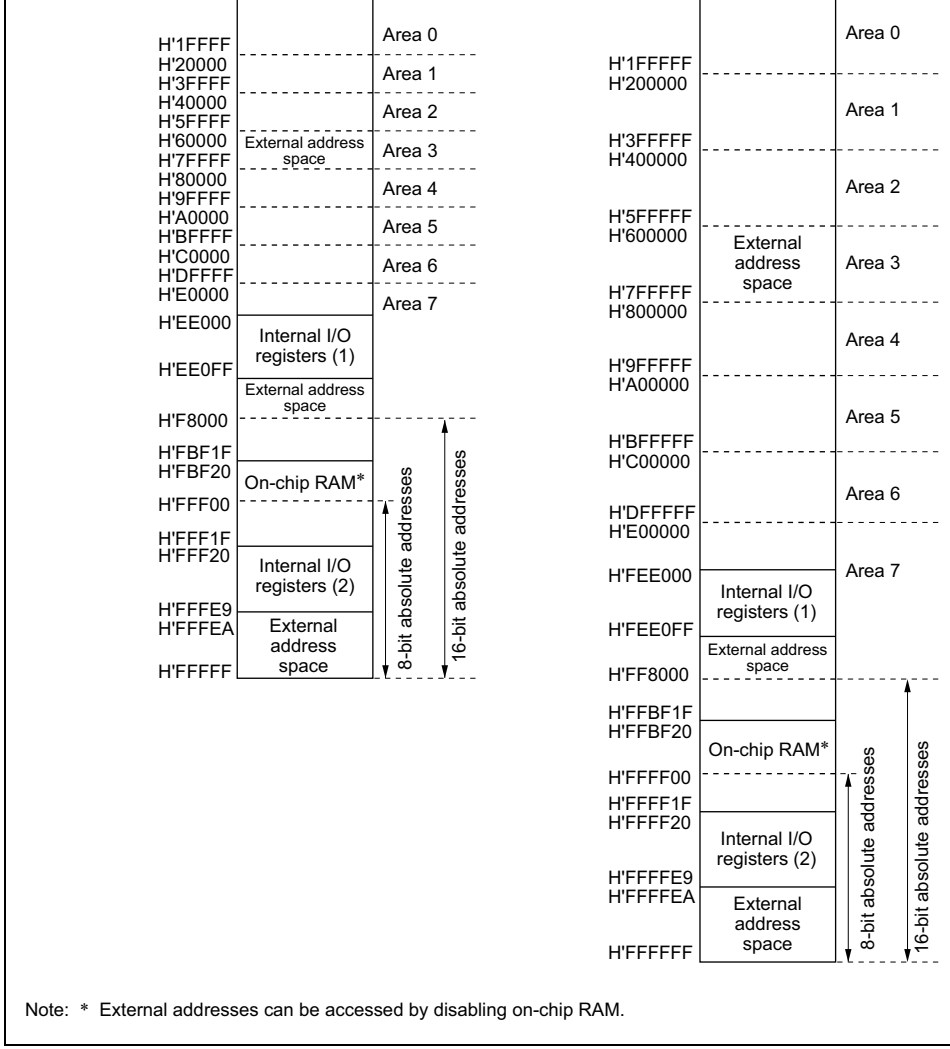
The address locations of the on-chip RAM and on-chip registers differ between the 64
(mode 6), the 1-Mbyte modes (modes 1, 2, and 7), and the 16-Mbyte modes (modes 3
The address range specifiable by the CPU in the 8- and 16-bit absolute addressing mo
and @aa:16) also differs.

3.6.1 Note on Reserved Areas

The H8/3028 Group memory map includes reserved areas to which read/write access
Note that normal operation is not guaranteed if the following reserved areas are acces

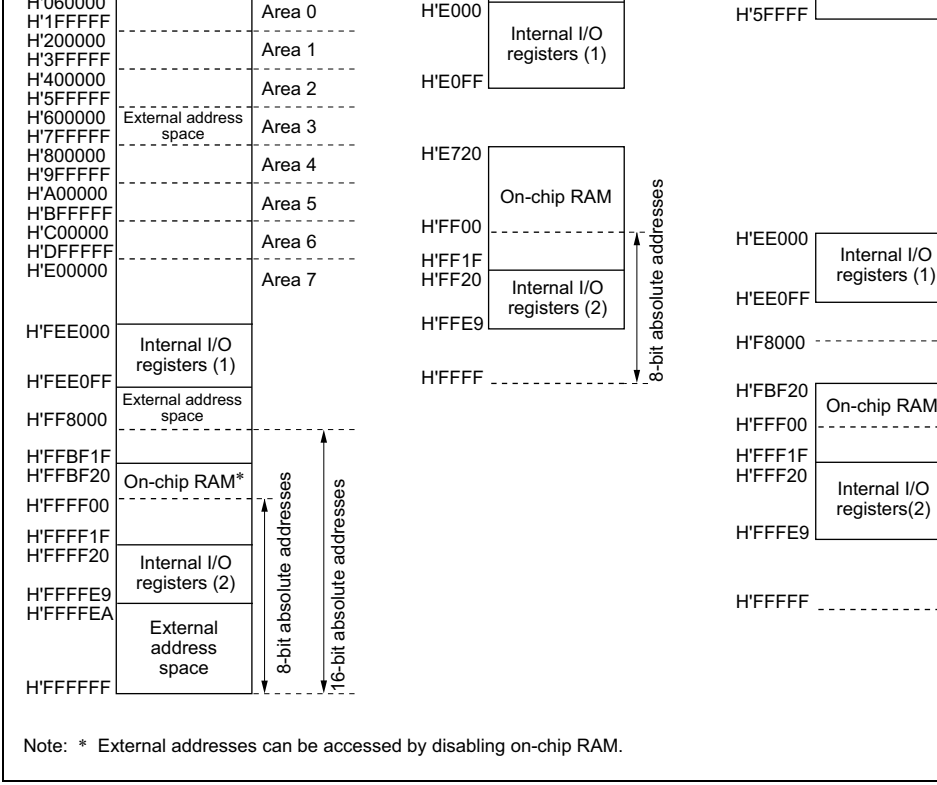
The reserved area in the internal I/O register space.

The H8/3028 Group internal I/O register space includes a reserved area to which acce
prohibited. For details see Appendix B, Internal I/O Registers.



Note: * External addresses can be accessed by disabling on-chip RAM.

Figure 3.1(1) H8/3028 Group Memory Map in Each Operating Mode



Note: * External addresses can be accessed by disabling on-chip RAM.

Figure 3.1(2) H8/3028 Group Memory Map in Each Operating Mode (EM

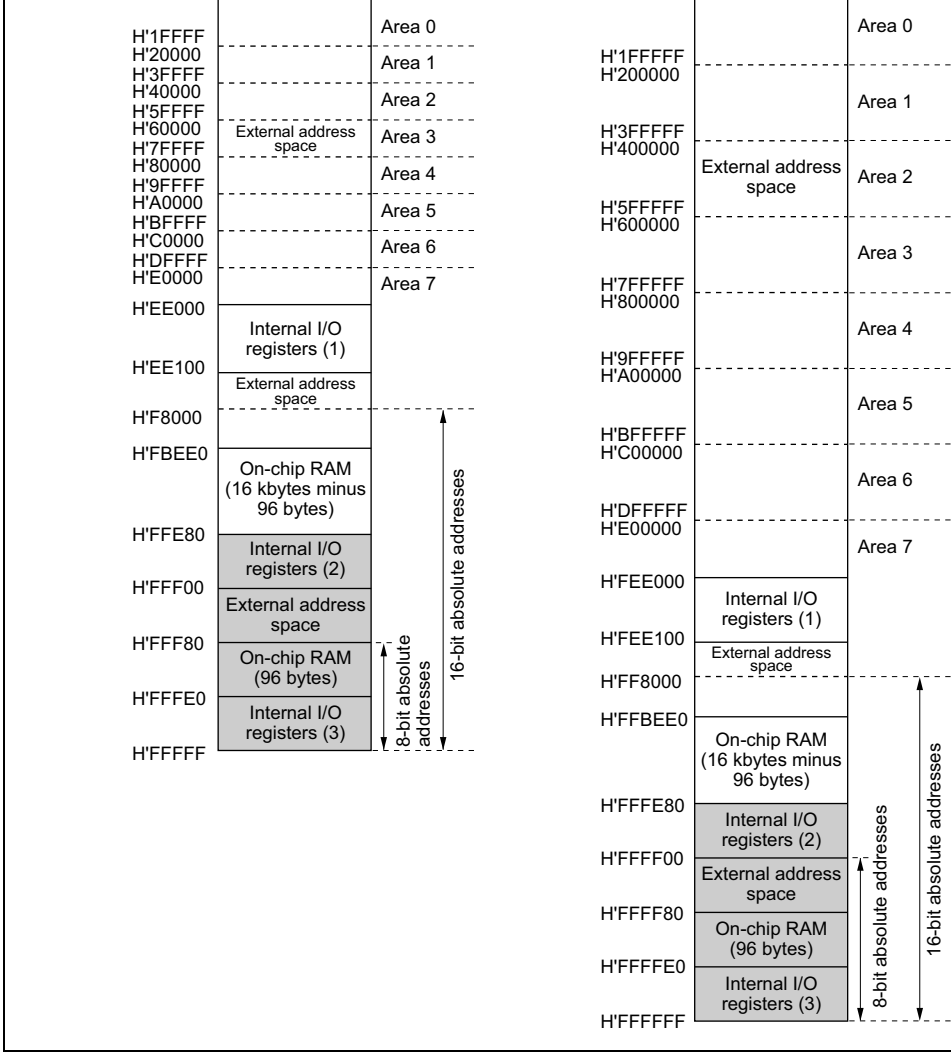


Figure 3.2(1) H8/3028 Group Memory Map in Each Operating Mode (EMC

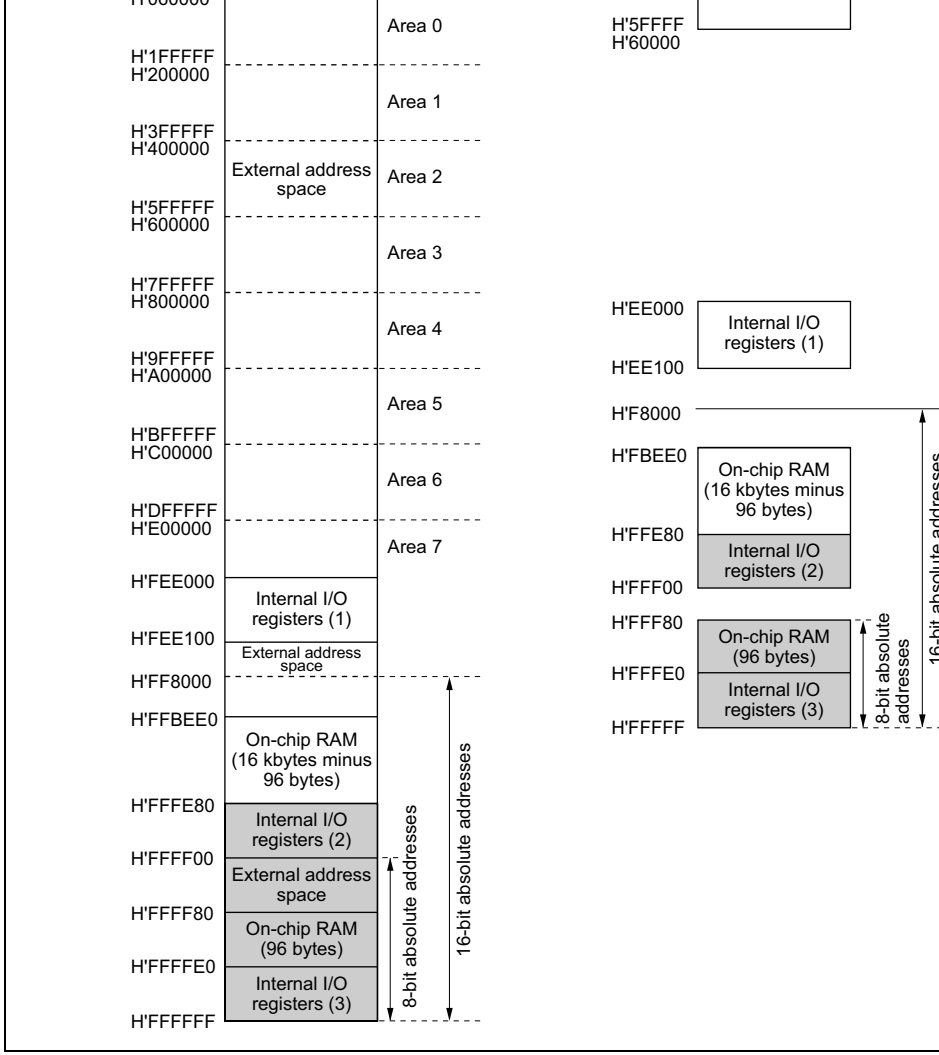


Figure 3.2(2) H8/3028 Group Memory Map in Each Operating Mode (EM

Exception handling is prioritized as shown in table 4.1. If two or more exceptions occ
simultaneously, they are accepted and processed in priority order. Trap instruction exc
accepted at all times in the program execution state.

Table 4.1 Exception Types and Priority

Priority	Exception Type	Start of Exception Handling
High	Reset	Starts immediately after a low-to-high transition at th
↕	Interrupt	Interrupt requests are handled when execution of the instruction or handling of the current exception is cor
Low	Trap instruction (TRAPA)	Started by execution of a trap instruction (TRAPA)

4.1.2 Exception Handling Operation

Exceptions originate from various sources. Trap instructions and interrupts are handle

1. The program counter (PC) and condition code register (CCR) are pushed onto the
2. The CCR interrupt mask bit is set to 1.
3. A vector address corresponding to the exception source is generated, and program
   starts from that address.

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

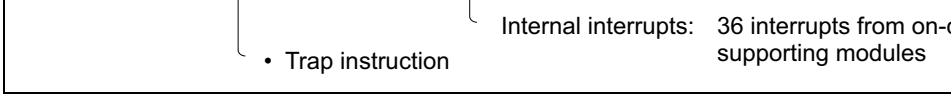


Figure 4.1 Exception Sources

	4	H'0010 to H'0013	H'0008
	5	H'0014 to H'0017	H'000A
	6	H'0018 to H'001B	H'000C
External interrupt (NMI)	7	H'001C to H'001F	H'000E
Trap instruction (4 sources)	8	H'0020 to H'0023	H'0010
	9	H'0024 to H'0027	H'0012
	10	H'0028 to H'002B	H'0014
	11	H'002C to H'002F	H'0016
External interrupt IRQ_0	12	H'0030 to H'0033	H'0018
External interrupt IRQ_1	13	H'0034 to H'0037	H'001A
External interrupt IRQ_2	14	H'0038 to H'003B	H'001C
External interrupt IRQ_3	15	H'003C to H'003F	H'001E
External interrupt IRQ_4	16	H'0040 to H'0043	H'0020
External interrupt IRQ_5	17	H'0044 to H'0047	H'0022
Reserved for system use	18	H'0048 to H'004B	H'0024
	19	H'004C to H'004F	H'0026
Internal interrupts*2	20 to 63	H'0050 to H'0053 to H'00FC to H'00FF	H'0028 to H'007E

Notes: 1. Lower 16 bits of the address.

2. For the internal interrupt vectors, see section 5.3.3, Interrupt Vector Table.

The chip can also be reset by overflow of the watchdog timer. For details see section 12 Watchdog Timer.

4.2.2 Reset Sequence

The chip enters the reset state when the $\overline{\text{RES}}$ pin goes low.

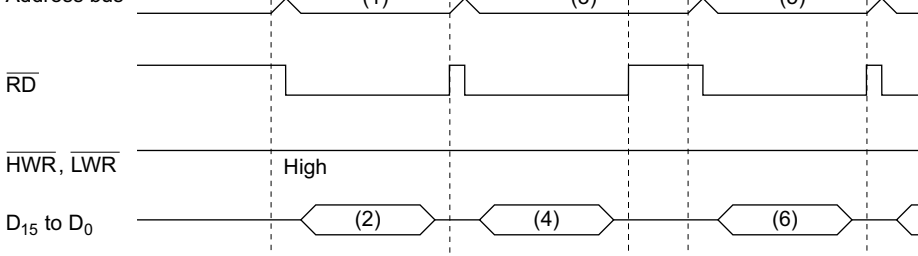
To ensure that the chip is reset, hold the $\overline{\text{RES}}$ pin low for at least 20 ms at power-up. To chip during operation, hold the $\overline{\text{RES}}$ pin low for at least 10 system clock (ϕ) cycles. Wh flash memory and flash memory R versions are used, the $\overline{\text{RES}}$ pin must be held low for system clock cycles. See appendix D.2, Pin States at Reset, for the states of the pins in t state.

When the $\overline{\text{RES}}$ pin goes high after being held low for the necessary time, the chip starts exception handling as follows.

- The internal state of the CPU and the registers of the on-chip supporting modules ar initialized, and the I bit is set to 1 in CCR.
- The contents of the reset vector address (H'0000 to H'0003 in advanced mode, H'00 H'0001 in normal mode) are read, and program execution starts from the address ind the vector address.

Figure 4.2 shows the reset sequence in modes 1 and 3. Figure 4.3 shows the reset seque modes 2 and 4. Figure 4.4 shows the reset sequence in mode 6.

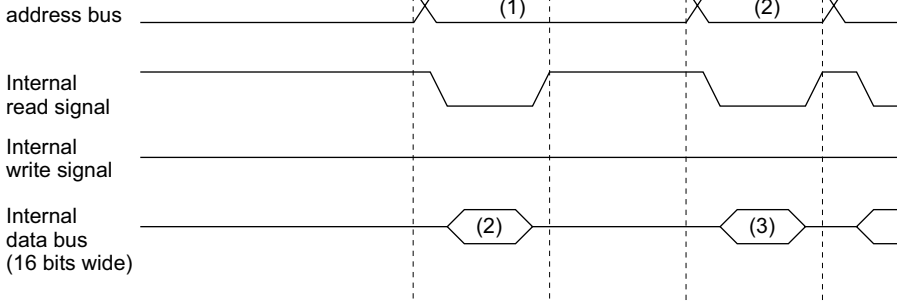
Figure 4.2 Reset Sequence (Modes 1 and 3)



(1), (3) Address of reset vector: (1) = H'000000, (3) = H'000002
(2), (4) Start address (contents of reset exception handling vector address)
(5) Start address
(6) First instruction of program

Note: After a reset, the wait-state controller inserts three wait states in every bus cycle.

Figure 4.3 Reset Sequence (Modes 2 and 4)



(1) Address of reset vector (H'0000)
(2) Start address (contents of reset exception handling vector address)
(3) First instruction of program

Figure 4.4 Reset Sequence (Mode 6)

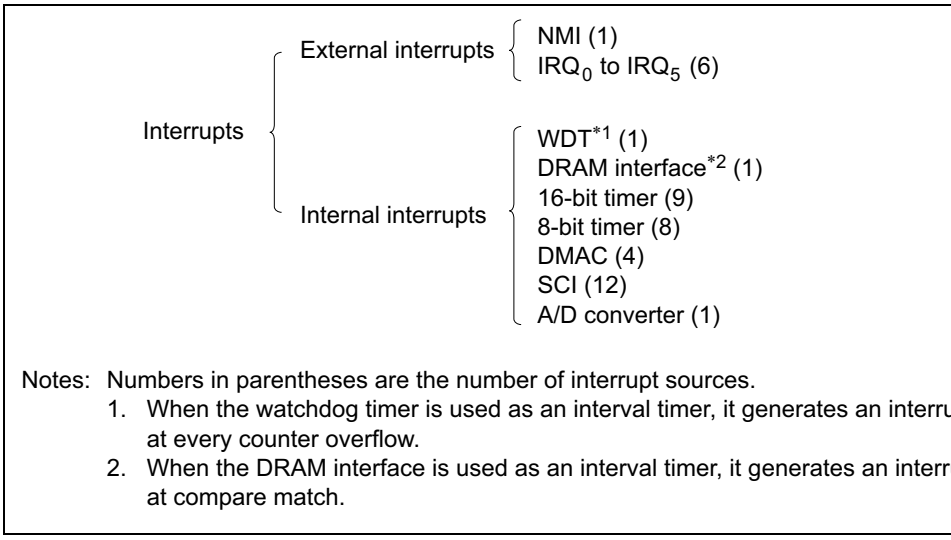
4.2.3 Interrupts after Reset

If an interrupt is accepted after a reset but before the stack pointer (SP) is initialized, P
will not be saved correctly, leading to a program crash. To prevent this, all interrupt re
including NMI, are disabled immediately after a reset. The first instruction of the prog
always executed immediately after the reset state ends. This instruction should initiali
pointer (example: MOV.L #xx:32, SP).

NMI is the highest-priority interrupt and is always accepted*. Interrupts are controlled
interrupt controller. The interrupt controller can assign interrupts other than NMI to tw
levels, and arbitrate between simultaneous interrupts. Interrupt priorities are assigned i
priority registers A and B (IPRA and IPRB) in the interrupt controller.

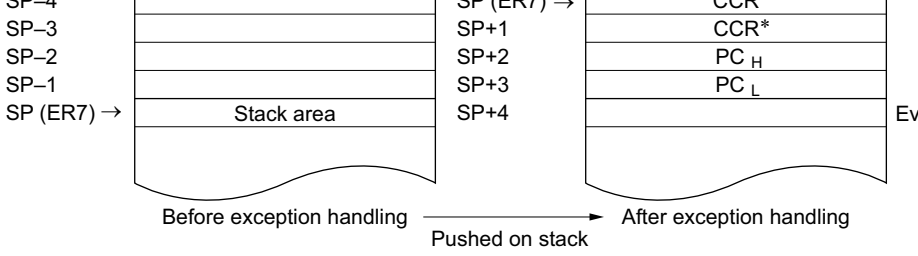
Note: * In the flash memory version, NMI input is sometimes disabled. For details see
18.9, NMI Input Disable Conditions.

For details on interrupts see section 5, Interrupt Controller.

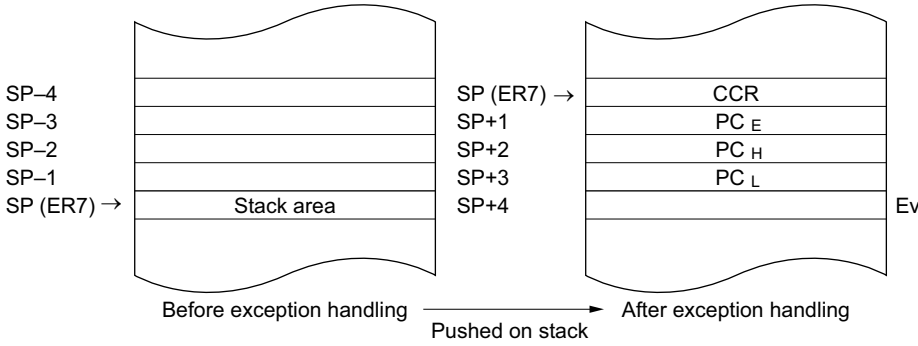


Notes: Numbers in parentheses are the number of interrupt sources.
1. When the watchdog timer is used as an interval timer, it generates an interru
at every counter overflow.
2. When the DRAM interface is used as an interval timer, it generates an interr
at compare match.

Figure 4.5 Interrupt Sources and Number of Interrupts



a. Normal mode



b. Advanced mode

Legend
PCE: Bits 23 to 16 of program counter (PC)
PCH: Bits 15 to 8 of program counter (PC)
PCL: Bits 7 to 0 of program counter (PC)
CCR: Condition code register
SP: Stack pointer

Notes: PC indicates the address of the first instruction that will be executed after return.
Registers must be saved in word or longword size at even addresses.
* Ignored at return.

Figure 4.6 Stack after Completion of Exception Handling

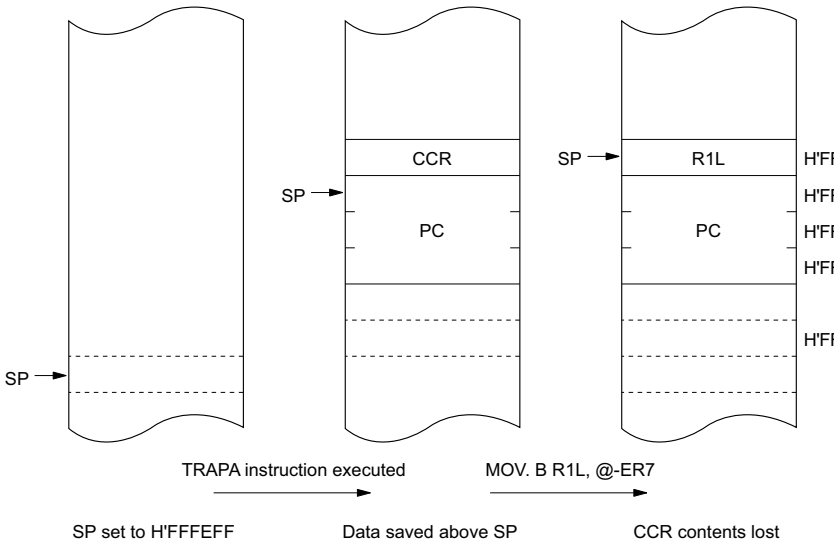
PUSH.L ERn (or MOV.L ERn, @–SP)

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 4.7 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: The diagram illustrates modes 3 and 4.

Figure 4.7 Operation when SP Value is Odd

- Interrupt priority registers (IPRs) for setting interrupt priorities

  Interrupts other than NMI can be assigned to two priority levels on a module-by-m
  in interrupt priority registers A and B (IPRA and IPRB).
- Three-level masking by the I and UI bits in the CPU condition code register (CCR
- Seven external interrupt pins

  NMI has the highest priority and is always accepted*; either the rising or falling ed
  selected. For each of IRQ_0 to IRQ_5, sensing of the falling edge or level sensing can
  independently.

Note: * In the flash memory, NMI input is sometimes disabled. For details see 18.9, N
Disable Conditions.

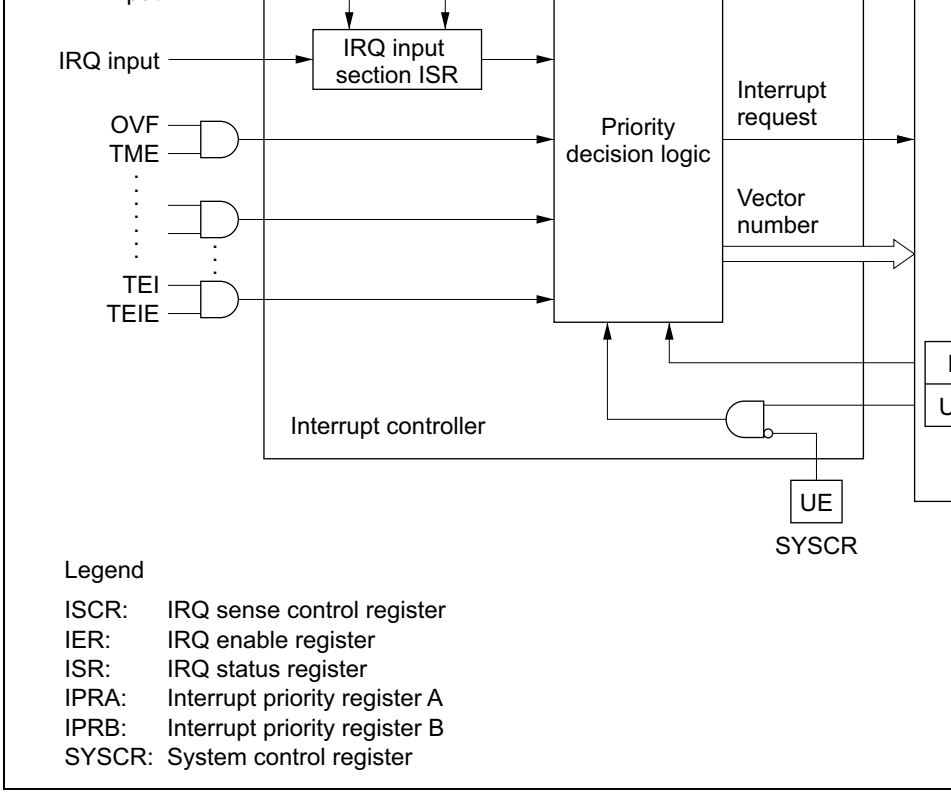


Figure 5.1 Interrupt Controller Block Diagram

External interrupt request 5 to 0	IRQ_5 to IRQ_0	Input	Maskable interrupts; falling e sensing selectable	

Note: * NMI input is sometimes disabled. For details see section 18.9, NMI Input Disab Conditions.

5.1.4 Register Configuration

Table 5.2 lists the registers of the interrupt controller.

Table 5.2 Interrupt Controller Registers

Address*1	Name	Abbreviation	R/W	Ini
H'EE012	System control register	SYSCR	R/W	H'0
H'EE014	IRQ sense control register	ISCR	R/W	H'0
H'EE015	IRQ enable register	IER	R/W	H'0
H'EE016	IRQ status register	ISR	R/(W)*2	H'0
H'EE018	Interrupt priority register A	IPRA	R/W	H'0
H'EE019	Interrupt priority register B	IPRB	R/W	H'0

Notes: 1. Lower 20 bits of the address in advanced mode.
2. Only 0 can be written, to clear flags.

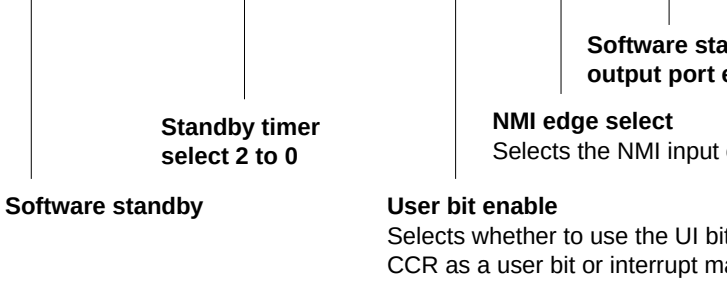
5.2 Register Descriptions

5.2.1 System Control Register (SYSCR)

SYSCR is an 8-bit readable/writable register that controls software standby mode, sele action of the UI bit in CCR, selects the NMI edge, and enables or disables the on-chip

Only bits 3 and 2 are described here. For the other bits, see section 3.3, System Contro (SYSCR).

SYSCR is initialized to H'09 by a reset and in hardware standby mode. It is not initiali software standby mode.



Bit 3—User Bit Enable (UE): Selects whether to use the UI bit in CCR as a user bit or interrupt mask bit.

Bit 3 UE	Description	
0	UI bit in CCR is used as interrupt mask bit	
1	UI bit in CCR is used as user bit	(In

Bit 2—NMI Edge Select (NMIEG): Selects the NMI input edge.

Bit 2 NMIEG	Description	
0	Interrupt is requested at falling edge of NMI input	(In
1	Interrupt is requested at rising edge of NMI input	

5.2.2 Interrupt Priority Registers A and B (IPRA, IPRB)

IPRA and IPRB are 8-bit readable/writable registers that control interrupt priority.

Priority level A7
Selects the priority level of IRQ_0 interrupt requests

Priority level A6
Selects the priority level of IRQ^1 interrupt requests

Priority level A5
Selects the priority level of IRQ^2 and IRQ^3 interrup

Priority level A4
Selects the priority level of IRQ^4 and IRQ
interrupt requests

Priority level A3
Selects the priority level of WD
DRAM interface, and A/D conv
interrupt requests

Priority level A2
Selects the priority leve
16-bit timer channel 0 i
requests

Priority leve
Selects the p
of 16-bit time
interrupt requ

Pr
le
Se
pr
of
ch
int
re

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

Bit 6 IPRA6	Description
0	IRQ_1 interrupt requests have priority level 0 (low priority) (In
1	IRQ_1 interrupt requests have priority level 1 (high priority)

Bit 5—Priority Level A5 (IPRA5): Selects the priority level of IRQ_2 and IRQ_3 interru

Bit 5 IPRA5	Description
0	IRQ_2 and IRQ_3 interrupt requests have priority level 0 (low priority) (In
1	IRQ_2 and IRQ_3 interrupt requests have priority level 1 (high priority)

Bit 4—Priority Level A4 (IPRA4): Selects the priority level of IRQ_4 and IRQ_5 interru

Bit 4 IPRA4	Description
0	IRQ_4 and IRQ_5 interrupt requests have priority level 0 (low priority) (In
1	IRQ_4 and IRQ_5 interrupt requests have priority level 1 (high priority)

Bit 3—Priority Level A3 (IPRA3): Selects the priority level of WDT, DRAM interfac converter interrupt requests.

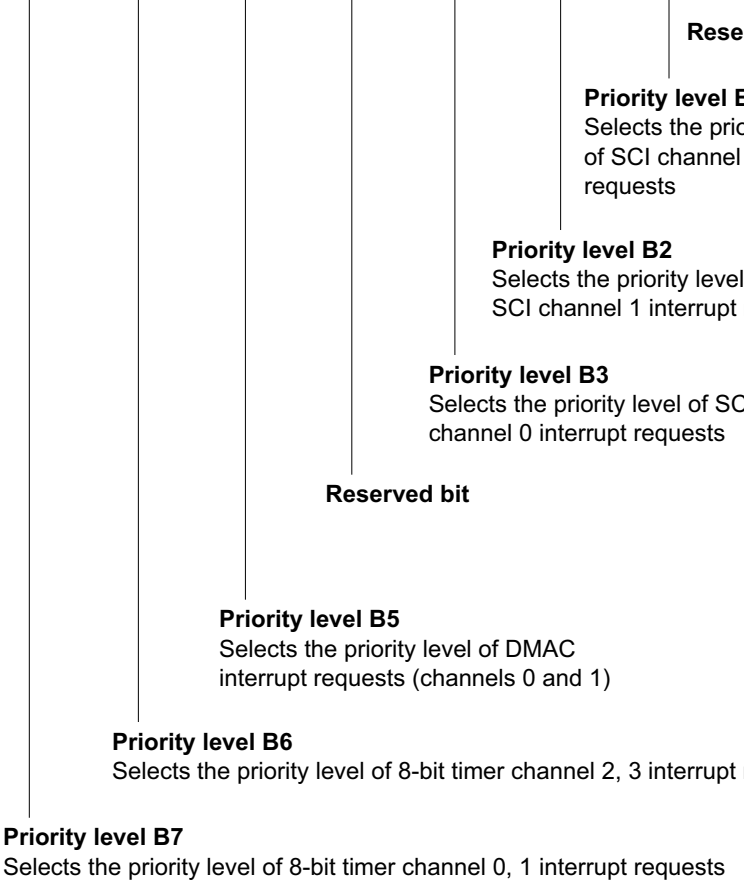
Bit 3 IPRA3	Description
0	WDT, DRAM interface, and A/D converter interrupt requests have priority lev (low priority) (In
1	WDT, DRAM interface, and A/D converter interrupt requests have priority lev (high priority)

requests.

Bit 1 IPRA1	Description
0	16-bit timer channel 1 interrupt requests have priority level 0 (low priority) (I
1	16-bit timer channel 1 interrupt requests have priority level 1 (high priority)

Bit 0—Priority Level A0 (IPRA0): Selects the priority level of 16-bit timer channel requests.

Bit 0 IPRA0	Description
0	16-bit timer channel 2 interrupt requests have priority level 0 (low priority) (I
1	16-bit timer channel 2 interrupt requests have priority level 1 (high priority)



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

Bit 6—Priority Level B6 (IPRB6): Selects the priority level of 8-bit timer channel 2, requests.

Bit 6 IPRB6	Description
0	8-bit timer channel 2, 3 interrupt requests have priority level 0 (low priority)(
1	8-bit timer channel 2, 3 interrupt requests have priority level 1 (high priority)

Bit 5—Priority Level B5 (IPRB5): Selects the priority level of DMAC interrupt requ (channels 0 and 1).

Bit 5 IPRB5	Description
0	DMAC interrupt requests (channels 0 and 1) have priority level 0 (low priority) (
1	DMAC interrupt requests (channels 0 and 1) have priority level 1 (high prior

Bit 4—Reserved: This bit can be written and read, but it does not affect interrupt prio

Bit 3—Priority Level B3 (IPRB3): Selects the priority level of SCI channel 0 interru

Bit 3 IPRB3	Description
0	SCI0 interrupt requests have priority level 0 (low priority) (
1	SCI0 interrupt requests have priority level 1 (high priority)

Bit 2—Priority Level B2 (IPRB2): Selects the priority level of SCI channel 1 interru

Bit 2 IPRB2	Description
0	SCI1 interrupt requests have priority level 0 (low priority) (
1	SCI1 interrupt requests have priority level 1 (high priority)

5.2.3 IRQ Status Register (ISR)

ISR is an 8-bit readable/writable register that indicates the status of IRQ_0 to IRQ_5 interr requests.

Bit	7	6	5	4	3	2	1
	—	—	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)*

Reserved bits

IRQ_5 to IRQ_0 flags
These bits indicate IRQ_5 to IR interrupt request status

Note: * Only 0 can be written, to clear flags.

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

Bits 7 and 6—Reserved: These bits can not be modified and are always read as 0.

Bits 5 to 0—IRQ_5 to IRQ_0 Flags (IRQ5F to IRQ0F): These bits indicate the status of IRQ_0 interrupt requests.

Bits 5 to 0 IRQ5F to IRQ0F	Description
0	[Clearing conditions] (In 0 is written in IRQnF after reading the IRQnF flag when IRQnF = 1. IRQnSC = 0, $\overline{IRQn}$ input is high, and interrupt exception handling is ca IRQnSC = 1 and IRQn interrupt exception handling is carried out.
1	[Setting conditions] IRQnSC = 0 and $\overline{IRQn}$ input is low. IRQnSC = 1 and $\overline{IRQn}$ input changes from high to low.

Note: n = 5 to 0

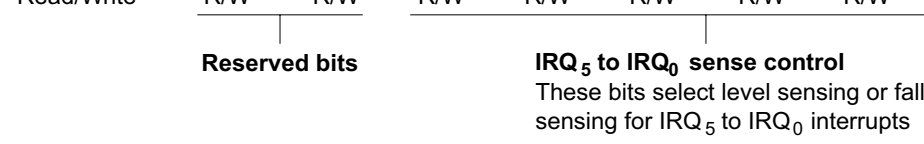
Reserved bits	IRQ_5 to IRQ_0 enable These bits enable or disable IRQ_5 to IRQ

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

Bits 7 and 6—Reserved: These bits can be written and read, but they do not enable o interrupts.

Bits 5 to 0—IRQ_5 to IRQ_0 Enable (IRQ5E to IRQ0E): These bits enable or disable IRQ_5 to IRQ_0 interrupts.

Bits 5 to 0 IRQ5E to IRQ0E	Description	
0	IRQ_5 to IRQ_0 interrupts are disabled	(I
1	IRQ_5 to IRQ_0 interrupts are enabled	



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

Bits 7 and 6—Reserved: These bits can be written and read, but they do not select leve falling-edge sensing.

Bits 5 to 0—IRQ_5 to IRQ_0 Sense Control (IRQ5SC to IRQ0SC): These bits select w interrupts IRQ_5 to IRQ_0 are requested by level sensing of pins $\overline{IRQ_5}$ to $\overline{IRQ_0}$, or by falli sensing.

Bits 5 to 0 IRQ5SC to IRQ0SC	Description	
0	Interrupts are requested when $\overline{IRQ_5}$ to $\overline{IRQ_0}$ inputs are low	(In
1	Interrupts are requested by falling-edge input at $\overline{IRQ_5}$ to $\overline{IRQ_0}$	

NMI: NMI is the highest-priority interrupt and is always accepted, regardless of the st
and UI bits in CCR*. The NMIEG bit in SYSCR selects whether an interrupt is reques
rising or falling edge of the input at the NMI pin. NMI interrupt exception handling ha
number 7.

Note: * NMI input is sometimes disabled. For details see section 18.9, NMI Input Dis
Conditions.

IRQ_0 to IRQ_5 Interrupts: These interrupts are requested by input signals at pins $\overline{IRQ}$
The IRQ_0 to IRQ_5 interrupts have the following features.

- ISCR settings can select whether an interrupt is requested by the low level of the i
$\overline{IRQ}_0$ to $\overline{IRQ}_5$, or by the falling edge.
- IER settings can enable or disable the IRQ_0 to IRQ_5 interrupts. Interrupt priority le
assigned by four bits in IPRA (IPRA7 to IPRA4).
- The status of IRQ_0 to IRQ_5 interrupt requests is indicated in ISR. The ISR flags ca
to 0 by software.

Figure 5.2 shows a block diagram of interrupts IRQ_0 to IRQ_5.

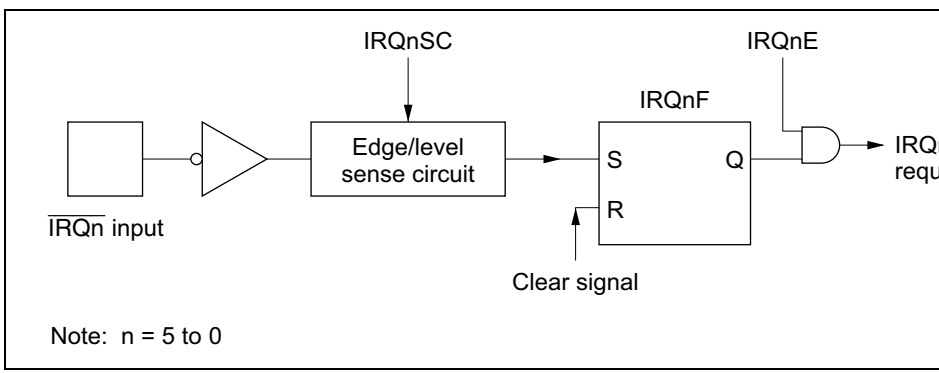
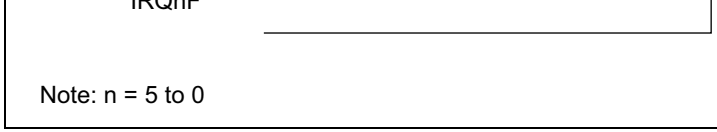


Figure 5.2 Block Diagram of Interrupts IRQ_0 to IRQ_5



Note: n = 5 to 0

Figure 5.3 Timing of Setting of IRQnF

Interrupts IRQ_0 to IRQ_5 have vector numbers 12 to 17. These interrupts are detected reg
whether the corresponding pin is set for input or output. When using a pin for external i
input, clear its DDR bit to 0 and do not use the pin for chip select output, refresh output
input/output, or A/D external trigger input.

5.3.2 Internal Interrupts

Thirty-Six internal interrupts are requested from the on-chip supporting modules.

- Each on-chip supporting module has status flags for indicating interrupt status, and
  for enabling or disabling interrupts.
- Interrupt priority levels can be assigned in IPRA and IPRB.
- 16-bit timer, SCI, and A/D converter interrupt requests can activate the DMAC, in v
  no interrupt request is sent to the interrupt controller, and the I and UI bits are disreg

5.3.3 Interrupt Vector Table

Table 5.3 lists the interrupt sources, their vector addresses, and their default priority orc
default priority order, smaller vector numbers have higher priority. The priority of inter
than NMI can be changed in IPRA and IPRB. The priority order after a reset is the defa
shown in table 5.3.

IRQ_3		15	H'003C to H'003F	H'001E to H'001F	
IRQ_4		16	H'0040 to H'0043	H'0020 to H'0021	IPR
IRQ_5		17	H'0044 to H'0047	H'0022 to H'0023	
Reserved	—	18	H'0048 to H'004B	H'0024 to H'0025	
		19	H'004C to H'004F	H'0026 to H'0027	
WOVI (interval timer)	Watchdog timer	20	H'0050 to H'0053	H'0028 to H'0029	IPR
CMI (compare match)	DRAM interface	21	H'0054 to H'0057	H'002A to H'002B	
Reserved	—	22	H'0058 to H'005B	H'002C to H'002D	
ADI (A/D end)	A/D	23	H'005C to H'005F	H'002E to H'002F	
IMIA0 (compare match/ input capture A0)	16-bit timer channel 0	24	H'0060 to H'0063	H'0030 to H'0031	IPR
IMIB0 (compare match/ input capture B0)		25	H'0064 to H'0067	H'0032 to H'0033	
OVI0 (overflow 0)		26	H'0068 to H'006B	H'0034 to H'0035	
Reserved	—	27	H'006C to H'006F	H'0036 to H'0037	
IMIA1 (compare match/ input capture A1)	16-bit timer channel 1	28	H'0070 to H'0073	H'0038 to H'0039	IPR
IMIB1 (compare match/ input capture B1)		29	H'0074 to H'0077	H'003A to H'003B	
OVI1 (overflow 1)		30	H'0078 to H'007B	H'003C to H'003D	
Reserved	—	31	H'007C to H'007F	H'003E to H'003F	

Note: * Lower 16 bits of the address.

(overflow 2)					
Reserved	—	35	H'008C to H'008F	H'0046 to H'0047	
CMIA0 (compare match A0)	8-bit timer channel 0/1	36	H'0090 to H'0093	H'0048 to H'0049	IPRE
CMIB0 (compare match B0)		37	H'0094 to H'0097	H'004A to H'004B	
CMIA1/CMIB1 (compare match A1/B1)		38	H'0098 to H'009B	H'004C to H'004D	
TOVI0/TOVI1 (overflow 0/1)		39	H'009C to H'009F	H'004E to H'004F	
CMIA2 (compare match A2)	8-bit timer channel 2/3	40	H'00A0 to H'00A3	H'0050 to H'0051	IPRE
CMIB2 (compare match B2)		41	H'00A4 to H'00A7	H'0052 to H'0053	
CMIA3/CMIB3 (compare match A3/B3)		42	H'00A8 to H'00AB	H'0054 to H'0055	
TOVI2/TOVI3 (overflow 2/3)		43	H'00AC to H'00AF	H'0056 to H'0057	
DEND0A	DMAC	44	H'00B0 to H'00B3	H'0058 to H'0059	IPRE
DEND0B		45	H'00B4 to H'00B7	H'005A to H'005B	
DEND1A		46	H'00B8 to H'00BB	H'005C to H'005D	
DEND1B		47	H'00BC to H'00BF	H'005E to H'005F	
Reserved	—	48	H'00C0 to H'00C3	H'0060 to H'0061	—
		49	H'00C4 to H'00C7	H'0062 to H'0063	
		50	H'00C8 to H'00CB	H'0064 to H'0065	
		51	H'00CC to H'00CF	H'0066 to H'0067	

Note: * Lower 16 bits of the address.

empty 0)					
TEI0 (transmit end 0)		55	H'00DC to H'00DF	H'006E to H'006F	
ERI1 (receive error 1)	SCI channel 1	56	H'00E0 to H'00E3	H'0070 to H'0071	IPR
RXI1 (receive data full 1)		57	H'00E4 to H'00E7	H'0072 to H'0073	
TXI1 (transmit data empty 1)		58	H'00E8 to H'00EB	H'0074 to H'0075	
TEI1 (transmit end 1)		59	H'00EC to H'00EF	H'0076 to H'0077	
ERI2 (receive error 2)	SCI channel 2	60	H'00F0 to H'00F3	H'0078 to H'0079	IPR
RXI2 (receive data full 2)		61	H'00F4 to H'00F7	H'007A to H'007B	
TXI2 (transmit data empty 2)		62	H'00F8 to H'00FB	H'007C to H'007D	
TEI2 (transmit end 2)		63	H'00FC to H'00FF	H'007E to H'007F	

Note: * Lower 16 bits of the address.

NMI interrupts are always accepted except in the reset and hardware standby states*. IR interrupts and interrupts from the on-chip supporting modules have their own enable bi requests are ignored when the enable bits are cleared to 0.

Note: * NMI input is sometimes disabled. For details see section 18.9, NMI Input Disa Conditions.

Table 5.4 UE, I, and UI Bit Settings and Interrupt Handling

SYSCR	CCR		
UE	**I**	**UI**	**Description**
1	0	—	All interrupts are accepted. Interrupts with priority level 1 hav priority.
	1	—	No interrupts are accepted except NMI.
0	0	—	All interrupts are accepted. Interrupts with priority level 1 hav priority.
	1	0	NMI and interrupts with priority level 1 are accepted.
		1	No interrupts are accepted except NMI.

UE = 1: Interrupts IRQ_0 to IRQ_5 and interrupts from the on-chip supporting modules ca masked by the I bit in the CPU's CCR. Interrupts are masked when the I bit is set to 1, unmasked when the I bit is cleared to 0. Interrupts with priority level 1 have higher prio 5.4 is a flowchart showing how interrupts are accepted when UE = 1.

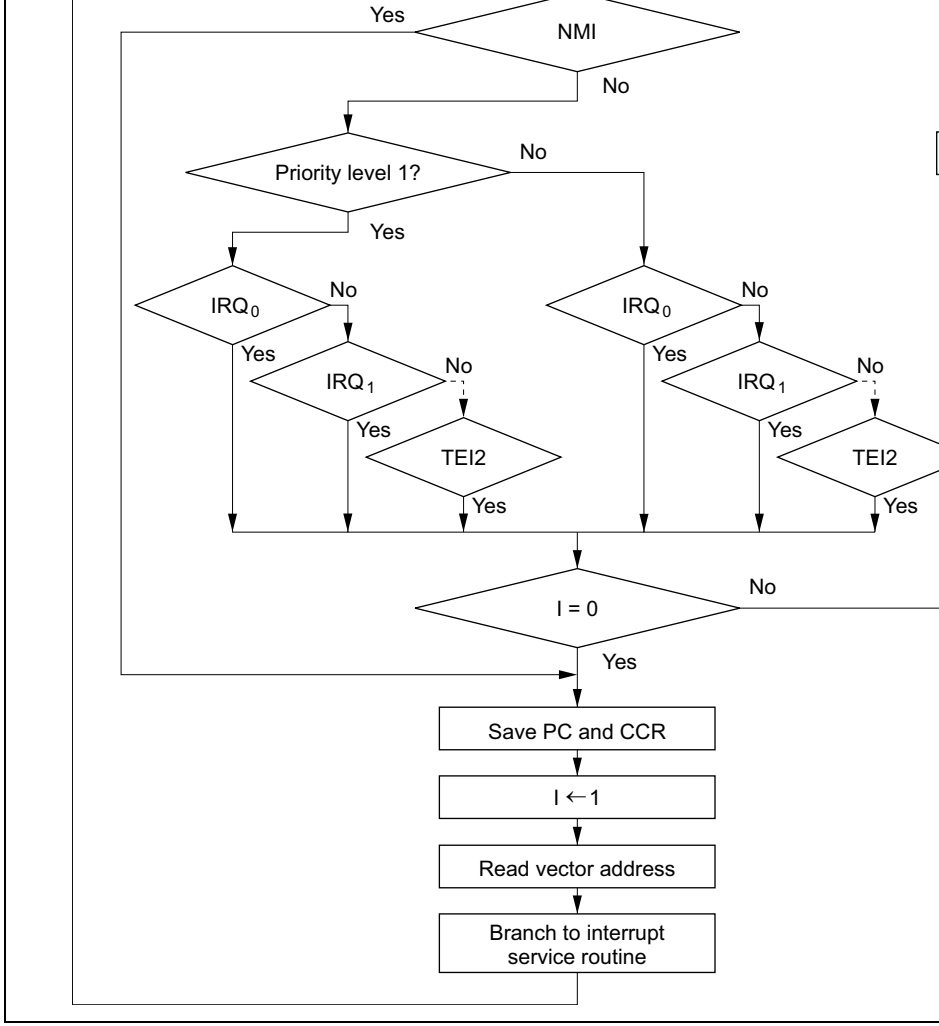


Figure 5.4 Process Up to Interrupt Acceptance when UE = 1

- When an interrupt request is accepted, interrupt exception handling starts after exec current instruction has been completed.
- In interrupt exception handling, PC and CCR are saved to the stack area. The PC va saved indicates the address of the first instruction that will be executed after the retu interrupt service routine.
- Next the I bit is set to 1 in CCR, masking all interrupts except NMI.
- The vector address of the accepted interrupt is generated, and the interrupt service r starts executing from the address indicated by the contents of the vector address.

UE = 0: The I and UI bits in the CPU's CCR and the IPR bits enable three-level maski IRQ_0 to IRQ_5 interrupts and interrupts from the on-chip supporting modules.

- Interrupt requests with priority level 0 are masked when the I bit is set to 1, and are when the I bit is cleared to 0.
- Interrupt requests with priority level 1 are masked when the I and UI bits are both s are unmasked when either the I bit or the UI bit is cleared to 0.

  For example, if the interrupt enable bits of all interrupt requests are set to 1, IPRA i H'20, and IPRB is set to H'00 (giving IRQ_2 and IRQ_3 interrupt requests priority ove interrupts), interrupts are masked as follows:

  a. If I = 0, all interrupts are unmasked (priority order: NMI > IRQ_2 > IRQ_3 >IRQ_0
  b. If I = 1 and UI = 0, only NMI, IRQ_2, and IRQ_3 are unmasked.
  c. If I = 1 and UI = 1, all interrupts are masked except NMI.

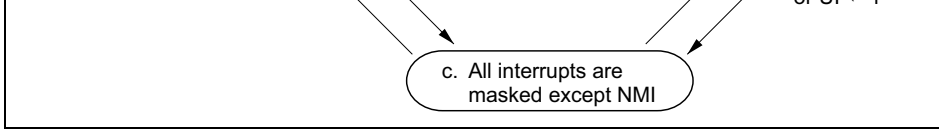


Figure 5.5 Interrupt Masking State Transitions (Example)

Figure 5.6 is a flowchart showing how interrupts are accepted when UE = 0.

- If an interrupt condition occurs and the corresponding interrupt enable bit is set to
interrupt request is sent to the interrupt controller.
- When the interrupt controller receives one or more interrupt requests, it selects the
priority request, following the IPR interrupt priority settings, and holds other reque
If two or more interrupts with the same IPR setting are requested simultaneously, t
controller follows the priority order shown in table 5.3.
- The interrupt controller checks the I bit. If the I bit is cleared to 0, the selected inte
is accepted regardless of its IPR setting, and regardless of the UI bit. If the I bit is
the UI bit is cleared to 0, only NMI and interrupts with priority level 1 are accepte
requests with priority level 0 are held pending. If the I bit and UI bit are both set to
NMI is accepted; all other interrupt requests are held pending.
- When an interrupt request is accepted, interrupt exception handling starts after exe
current instruction has been completed.
- In interrupt exception handling, PC and CCR are saved to the stack area. The PC v
saved indicates the address of the first instruction that will be executed after the re
interrupt service routine.
- The I and UI bits are set to 1 in CCR, masking all interrupts except NMI.
- The vector address of the accepted interrupt is generated, and the interrupt service
starts executing from the address indicated by the contents of the vector address.

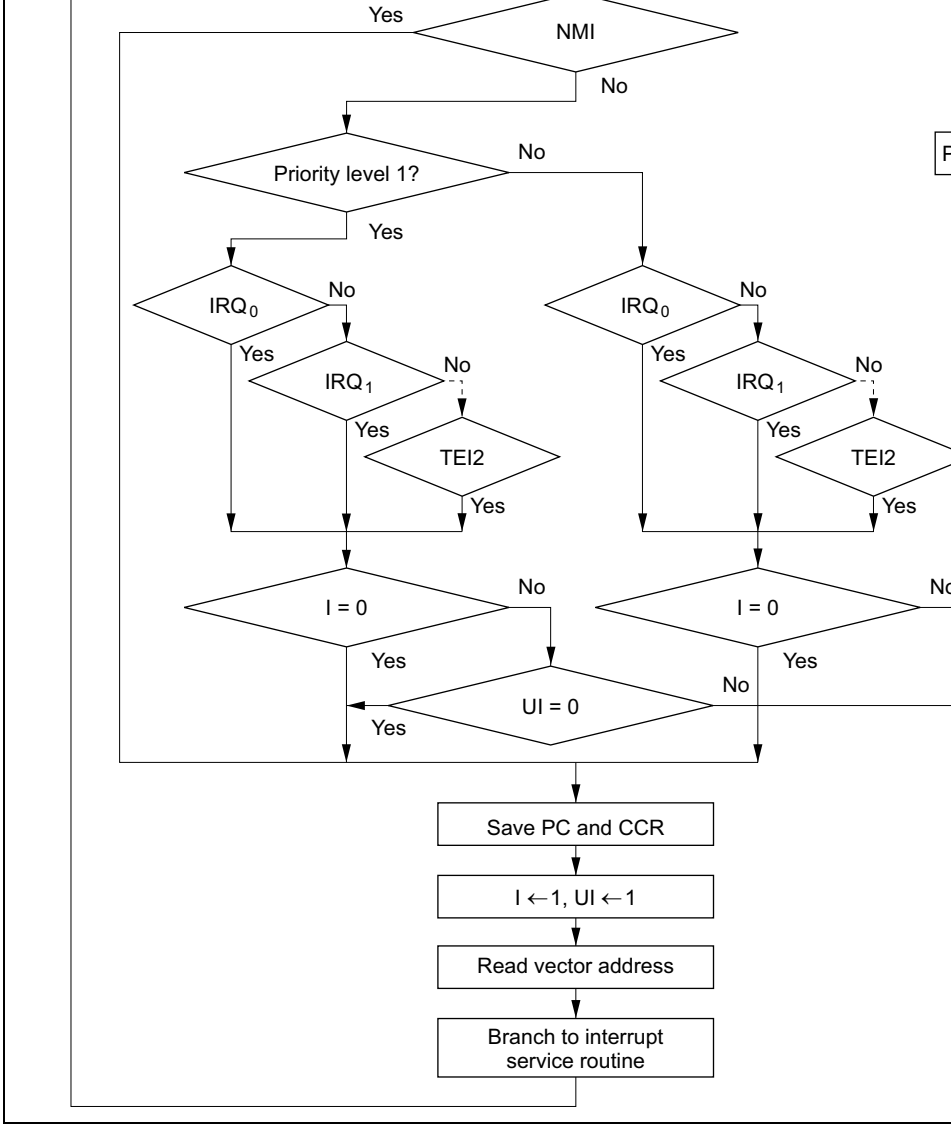


Figure 5.6 Process Up to Interrupt Acceptance when UE = 0

Figure 5.7 Interrupt Sequence

No.	Item	Memory	2 States	3 States	2 States	3
1	Interrupt priority decision	2*1	2*1	2*1	2*1	2
2	Maximum number of states until end of current instruction	1 to 23*5	1 to 27*5 *6	1 to 41*6	1 to 23*5	1
3	Saving PC and CCR to stack	4	8	12*4	4	6
4	Vector fetch	4	8	12*4	4	6
5	Instruction prefetch*2	4	8	12*4	4	6
6	Internal processing*3	4	4	4	4	4
Total		19 to 41	31 to 57	43 to 83	19 to 41	2

Notes: 1. 1 state for internal interrupts.
2. Prefetch after the interrupt is accepted and prefetch of the first instruction in interrupt service routine.
3. Internal processing after the interrupt is accepted and internal processing aft fetch.
4. The number of states increases if wait states are inserted in external memor
5. Example for DIVXS.W Rs,ERd and MULXS.W Rs,ERd
6. Example for MOV.L @(d:24,ERs),ERd and MOV.L ERs,@(d:24,ERd)

handling for the higher-priority interrupt is carried out, and the lower-priority interrup
This also applies to the clearing of an interrupt flag to 0.

Figure 5.8 shows an example in which an IMIEA bit is cleared to 0 in the 16-bit timer
register.

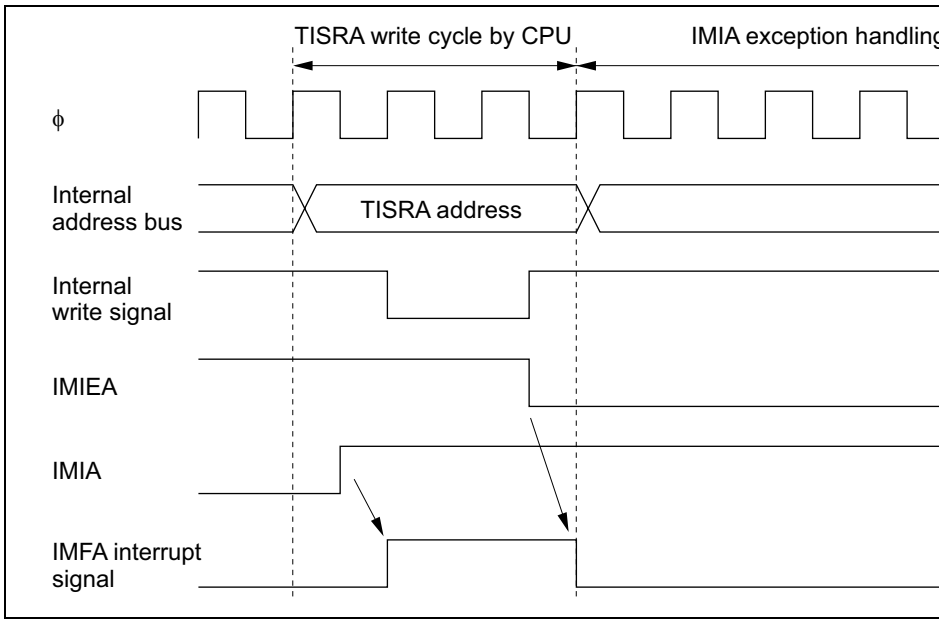


Figure 5.8 Contention between Interrupt and Interrupt-Disabling Instru

This type of contention will not occur if the interrupt is masked when the interrupt ena
flag is cleared to 0.

The EEPMOV.B and EEPMOV.W instructions differ in their reaction to interrupt requ

When the EEPMOV.B instruction is executing a transfer, no interrupts are accepted un transfer is completed, not even NMI.

When the EEPMOV.W instruction is executing a transfer, interrupt requests other than not accepted until the transfer is completed. If NMI is requested, NMI exception handli a transfer cycle boundary. The PC value saved on the stack is the address of the next in Programs should be coded as follows to allow for NMI interrupts during EEPMOV.W

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

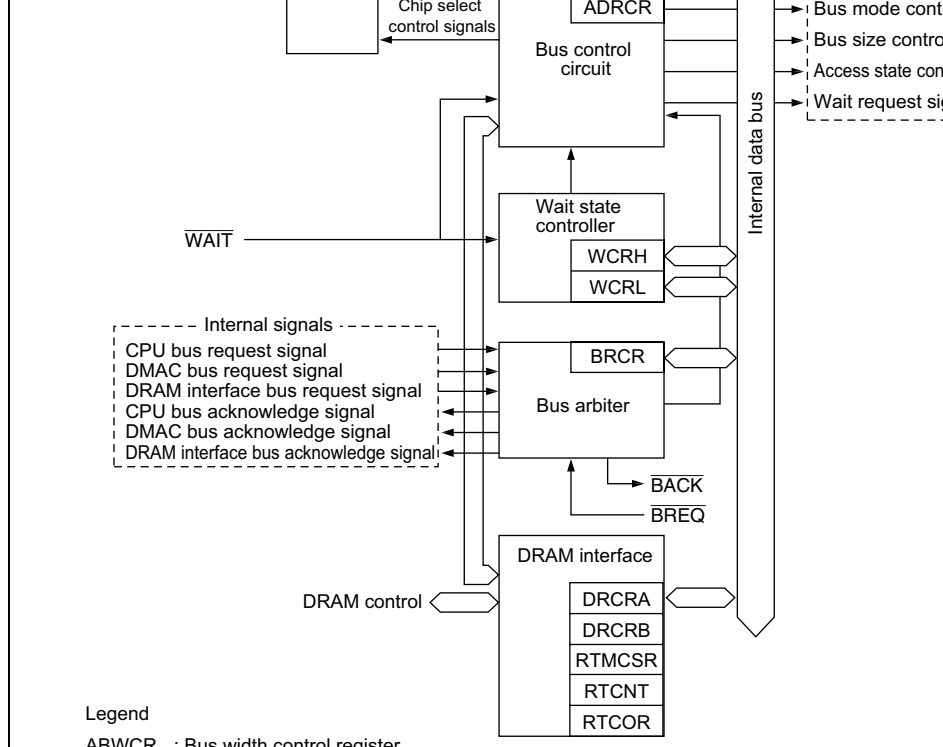
The bus controller also has a bus arbitration function that controls the operation of the masters-the CPU, DMA controller (DMAC), and DRAM interface and can release the external device.

6.1.1 Features

The features of the bus controller are listed below.

- Manages external address space in area units
  - Manages the external space as eight areas (0 to 7) of 128 kbytes in 1M-byte mo Mbytes in 16-Mbyte modes
  - Bus specifications can be set independently for each area
  - DRAM/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
  - Two-state access or three-state access can be selected for each area
  - Program wait states can be inserted for each area
  - Pin wait insertion capability is provided
- DRAM interface
  - DRAM interface can be set for areas 2 to 5
  - Row address/column address multiplexed output (8/9/10 bits)
  - 2-CAS byte access mode
  - Burst operation (fast page mode)
  - T_P cycle insertion to secure RAS precharging time
  - Choice of CAS-before-RAS refreshing or self-refreshing
- Burst ROM interface
  - Burst ROM interface can be set for area 0
  - Selection of two- or three-state burst access

— Refresh counter (refresh timer) can be used as interval timer
— Choice of two address update modes



Legend

ABWCR : Bus width control register
ASTCR : Access state control register
WCRH : Wait control register H
WCRL : Wait control register L
BRCR : Bus release control register
CSCR : Chip select control register
DRCRA : DRAM control register A
DRCRB : DRAM control register B
RTMCSR : Refresh timer control/status register
RTCNT : Refresh timer counter
RTCOR : Refresh time constant register
ADRCR : Address control register
BCR : Bus control register

Figure 6.1 Block Diagram of Bus Controller

			on the address bus
Read	$\overline{RD}$	Output	Strobe signal indicating reading fr external address space
High write	$\overline{HWR}$	Output	Strobe signal indicating writing to address space, with valid data on data bus (D_{15} to D_8)
Low write	$\overline{LWR}$	Output	Strobe signal indicating writing to address space, with valid data on data bus (D_7 to D_0)
Wait	$\overline{WAIT}$	Input	Wait request signal for access to e three-state access areas
Bus request	$\overline{BREQ}$	Input	Request signal for releasing the b external device
Bus acknowledge	$\overline{BACK}$	Output	Acknowledge signal indicating rele bus to an external device

H'EE022	Wait control register H	WCRH	R/W	H
H'EE023	Wait control register L	WCRL	R/W	H
H'EE013	Bus release control register	BRCR	R/W	H
H'EE01F	Chip select control register	CSCR	R/W	H
H'EE01E	Address control register	ADRCR	R/W	H
H'EE024	Bus control register	BCR	R/W	H
H'EE026	DRAM control register A	DRCRA	R/W	H
H'EE027	DRAM control register B	DRCRB	R/W	H
H'EE028	Refresh timer control/status register	RTMCSR	R(W)*4	H
H'EE029	Refresh timer counter	RTCNT	R/W	H
H'EE02A	Refresh time constant register	RTCOR	R/W	H

Notes: 1. Lower 20 bits of the address in advanced mode.
2. In modes 2 and 4, the initial value is H'00.
3. In modes 3 and 4, the initial value is H'EE.
4. For Bit 7, only 0 can be written to clear the flag.

Modes 1, 3, 5, 6, and 7	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
Modes 2 and 4	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

When ABWCR contains H'FF (selecting 8-bit access for all areas), the chip operates in mode: the upper data bus (D_{15} to D_8) is valid, and port 4 is an input/output port. When bit is cleared to 0 in ABWCR, the chip operates in 16-bit bus mode with a 16-bit data b D_0). In modes 1, 3, 5, 6, and 7, ABWCR is initialized to H'FF by a reset and in hardwa mode. In modes 2 and 4, ABWCR is initialized to H'00 by a reset and in hardware stan It is not initialized in software standby mode.

Bits 7 to 0—Area 7 to 0 Bus Width Control (ABW7 to ABW0): These bits select 8-b or 16-bit access for the corresponding areas.

Bits 7 to 0 ABW7 to ABW0	Description
0	Areas 7 to 0 are 16-bit access areas
1	Areas 7 to 0 are 8-bit access areas

ABWCR specifies the data bus width of external memory areas. The data bus width of memory and registers is fixed, and does not depend on ABWCR settings. These setting therefore meaningless in the single-chip modes (modes 6 and 7).

Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits selecting number of states for access to each area

ASTCR is initialized to H'FF by a reset and in hardware standby mode. It is not initia
software standby mode.

Bits 7 to 0—Area 7 to 0 Access State Control (AST7 to AST0): These bits select w
corresponding area is accessed in two or three states.

Bits 7 to 0 AST7 to AST0	Description
0	Areas 7 to 0 are accessed in two states
1	Areas 7 to 0 are accessed in three states (

ASTCR specifies the number of states in which external areas are accessed. On-chip
registers are accessed in a fixed number of states that does not depend on ASTCR sett
settings are therefore meaningless in the single-chip modes (modes 6 and 7).

When the corresponding area is designated as DRAM space by bits DRAS2 to DRAS
control register A (DRCRA), the number of access states does not depend on the AST
When an AST bit is cleared to 0, programmable wait insertion is not performed.

6.2.3 Wait Control Registers H and L (WCRH, WCRL)

WCRH and WCRL are 8-bit readable/writable registers that select the number of prog
states for each area.

On-chip memory and registers are accessed in a fixed number of states that does not d
WCRH/WCRL settings.

WCRH and WCRL are initialized to H'FF by a reset and in hardware standby mode.
initialized in software standby mode.

program wait states when area 7 in external space is accessed while the AST7 bit in AS
to 1.

Bit 7 W71	Bit 6 W70	Description
0	0	Program wait not inserted when external space area 7 is accessed
	1	1 program wait state inserted when external space area 7 is acces
1	0	2 program wait states inserted when external space area 7 is acce
	1	3 program wait states inserted when external space area 7 is acce (Initial v

Bits 5 and 4—Area 6 Wait Control 1 and 0 (W61, W60): These bits select the numb
program wait states when area 6 in external space is accessed while the AST6 bit in AS
to 1.

Bit 5 W61	Bit 4 W60	Description
0	0	Program wait not inserted when external space area 6 is accessed
	1	1 program wait state inserted when external space area 6 is acces
1	0	2 program wait states inserted when external space area 6 is acce
	1	3 program wait states inserted when external space area 6 is acce (Initial v

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.

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

Bit 1 W41	Bit 0 W40	Description
0	0	Program wait not inserted when external space area 4 is accesse
	1	1 program wait state inserted when external space area 4 is acce
1	0	2 program wait states inserted when external space area 4 is acc
	1	3 program wait states inserted when external space area 4 is acc (Initial

WCRL

Bit	7	6	5	4	3	2	1
	W31	W30	W21	W20	W11	W10	W01
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

Bits 7 and 6—Area 3 Wait Control 1 and 0 (W31, W30): These bits select the num
program wait states when area 3 in external space is accessed while the AST3 bit in A
to 1.

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

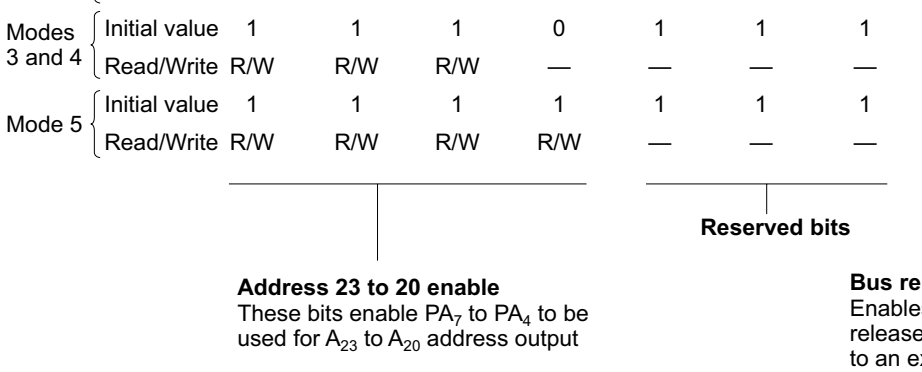
	1	3 program wait states inserted when external space area 2 is acce (Initial v

Bits 3 and 2—Area 1 Wait Control 1 and 0 (W11, W10): These bits select the numb program wait states when area 1 in external space is accessed while the AST1 bit in AS to 1.

Bit 3 W11	Bit 2 W10	Description
0	0	Program wait not inserted when external space area 1 is accessed
	1	1 program wait state inserted when external space area 1 is acces
1	0	2 program wait states inserted when external space area 1 is acce
	1	3 program wait states inserted when external space area 1 is acce (Initial v

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 W01	Bit 0 W00	Description
0	0	Program wait not inserted when external space area 0 is accessed
	1	1 program wait state inserted when external space area 0 is acces
1	0	2 program wait states inserted when external space area 0 is acce
	1	3 program wait states inserted when external space area 0 is acce (Initial v



BRCR is initialized to H'FE in modes 1, 2, 5, 6, and 7, and to H'EE in modes 3 and 4, and in hardware standby mode. It is not initialized in software standby mode.

Bit 7—Address 23 Enable (A23E): Enables PA_4 to be used as the A_{23} address output Writing 0 in this bit enables A_{23} output from PA_4. In modes other than 3, 4, and 5, thi be modified and PA_4 has its ordinary port functions.

Bit 7 A23E	Description
0	PA_4 is the A_{23} address output pin
1	PA_4 is an input/output pin (

Bit 6—Address 22 Enable (A22E): Enables PA_5 to be used as the A_{22} address output Writing 0 in this bit enables A_{22} output from PA_5. In modes other than 3, 4, and 5, thi be modified and PA_5 has its ordinary port functions.

Bit 6 A22E	Description
0	PA_5 is the A_{22} address output pin
1	PA_5 is an input/output pin (

Bit 4—Address 20 Enable (A20E): Enables PA_7 to be used as the A_{20} address output p
Writing 0 in this bit enables A_{20} output from PA_7. This bit can only be modified in mo

Bit 4 A20E	Description
0	PA_7 is the A_{20} address output pin (Initial value when in mode 3 or 4)
1	PA_7 is an input/output pin (Initial value when in mode 1, 2, 5, 6 or 7)

Bits 3 to 1—Reserved: These bits cannot be modified and are always read as 1.

Bit 0—Bus Release Enable (BRLE): Enables or disables release of the bus to an exter

Bit 0 BRLE	Description
0	The bus cannot be released to an external device $\overline{BREQ}$ and $\overline{BACK}$ can be used as input/output pins (Ir
1	The bus can be released to an external device

6.2.5 Bus Control Register (BCR)

Bit	7	6	5	4	3	2	1
	ICIS1	ICIS0	BROME	BRSTS1	BRSTS0	EMC	RDEA
Initial value	1	1	0	0	0	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W

BCR is an 8-bit readable/writable register that enables or disables idle cycle insertion, s
address map, selects the area division unit, and enables or disables $\overline{WAIT}$ pin input.

BCR is initialized to H'C6 by a reset and in hardware standby mode. It is not initialized
software standby mode.

Bit 6—Idle Cycle Insertion 0 (ICIS0): Selects whether one idle cycle state is to be in
between bus cycles in case of consecutive external read and write cycles.

Bit 6 ICIS0	Description
0	No idle cycle inserted in case of consecutive external read and writ
1	Idle cycle inserted in case of consecutive external read and write c (

Bit 5—Burst ROM Enable (BROME): Selects whether area 0 is a burst ROM interf

Bit 5 BROME	Description
0	Area 0 is a basic bus interface area (
1	Area 0 is a burst ROM interface area

Bit 4—Burst Cycle Select 1 (BRSTS1): Selects the number of burst cycle states for t
ROM interface.

Bit 4 BRSTS1	Description
0	Burst access cycle comprises 2 states (
1	Burst access cycle comprises 3 states

Bit 2—Expansion Memory Map Control (EMC): Selects either of the two memory m

Bit 2 EMC	Description
0	Selects the memory map shown in figure 3.2: see section 3.6, Mem Each Operating Mode
1	Selects the memory map shown in figure 3.1: see section 3.6, Mem Each Operating Mode (In

When EMC is cleared to 0, addresses of some internal I/O registers are moved. For det to appendix B.2, Address (when EMC = 0).

This bit is invalid in mode 6. In mode 6 and when the RDEA bit is 0, EMC must not b 0.

Bit 1—Area Division Unit Select (RDEA): Selects the memory map area division uni is valid in modes 3, 4, and 5, and is invalid in modes 1, 2, 6, and 7.

When the EMC bit is 0, RDEA must not be cleared to 0.

Bit 1 RDEA	Description		
0	Area divisions are as follows:	Area 0: 2 Mbytes	Area 4: 1.93 Mby
		Area 1: 2 Mbytes	Area 5: 4 kbytes
		Area 2: 8 Mbytes	Area 6: 23.75 kb
		Area 3: 2 Mbytes	Area 7: 22 bytes
1	Areas 0 to 7 are the same size (2 Mbytes)		(In

6.2.6 Chip Select Control Register (CSCR)

CSCR is an 8-bit readable/writable register that enables or disables output of chip sele
($\overline{CS}_7$ to $\overline{CS}_4$).

If output of a chip select signal is enabled by a setting in this register, the correspondi
functions as a chip select signal ($\overline{CS}_7$ to $\overline{CS}_4$) output regardless of any other settings. C
cannot be modified in single-chip mode.

Bit	7	6	5	4	3	2	1
	CS7E	CS6E	CS5E	CS4E	—	—	—
Initial value	0	0	0	0	1	1	1
Read/Write	R/W	R/W	R/W	R/W	—	—	—

Chip select 7 to 4 enable
These bits enable or disable chip select signal output

Reserved bits

CSCR is initialized to H'0F by a reset and in hardware standby mode. It is not initializ
software standby mode.

Bits 7 to 4—Chip Select 7 to 4 Enable (CS7E to CS4E): These bits enable or disabl
the corresponding chip select signal.

Bit n CSnE	Description	
0	Output of chip select signal $\overline{CSn}$ is disabled	(
1	Output of chip select signal $\overline{CSn}$ is enabled	

Note: n = 7 to 4

Bits 3 to 0—Reserved: These bits cannot be modified and are always read as 1.

function, and the access mode, and enables or disables self-refreshing and refresh pin o

DRCRA is initialized to H'10 by a reset and in hardware standby mode. It is not initial software standby mode.

Bits 7 to 5—DRAM Area Select (DRAS2 to DRAS0): These bits select which of area to function as DRAM interface areas (DRAM space) in expanded mode, and at the sam select the $\overline{RAS}$ output pin corresponding to each DRAM space.

			Description			
Bit 7 DRAS2	**Bit 6 DRAS1**	**Bit 5 DRAS0**	**Area 5**	**Area 4**	**Area 3**	**Area**
0	0	0	Normal	Normal	Normal	Norm
		1	Normal	Normal	Normal	DRAM ($\overline{CS_2}$)
	1	0	Normal	Normal	DRAM space ($\overline{CS_3}$)	DRAM ($\overline{CS_2}$)
		1	Normal	Normal	DRAM space ($\overline{CS_2}$)*	DRAM ($\overline{CS_2}$)
1	0	0	Normal	DRAM space ($\overline{CS_4}$)	DRAM space ($\overline{CS_3}$)	DRAM ($\overline{CS_2}$)
		1	DRAM space ($\overline{CS_5}$)	DRAM space ($\overline{CS_4}$)	DRAM space ($\overline{CS_3}$)	DRAM ($\overline{CS_2}$)
	1	0	DRAM space ($\overline{CS_4}$)*	DRAM space ($\overline{CS_4}$)*	DRAM space ($\overline{CS_2}$)*	DRAM ($\overline{CS_2}$)
		1	DRAM space ($\overline{CS_2}$)*	DRAM space ($\overline{CS_2}$)*	DRAM space ($\overline{CS_2}$)*	DRAM ($\overline{CS_2}$)

Note: * A single $\overline{CSn}$ pin serves as a common $\overline{RAS}$ output pin for a number of areas. U pins can be used as input/output ports.

When any of bits DRAS2 to DRAS0 is set to 1 in expanded mode, it is not possible to DRCRB, RTMCSR, RTCNT, or RTCOR. However, 0 can be written to the CMF flag RTMCSR to clear the flag.

0	Burst disabled (always full access)	(
1	DRAM space access performed in fast page mode	

Bit 2—RAS Down Mode (RDM): Selects whether to wait for the next DRAM access $\overline{\text{RAS}}$ signal held low (RAS down mode), or to drive the RAS signal high again (RAS when burst access is enabled for DRAM space (BE = 1), and access to DRAM is inter

Caution is required when the $\overline{\text{HWR}}$ and $\overline{\text{LWR}}$ are used as the $\overline{\text{UCAS}}$ and $\overline{\text{LCAS}}$ outpu details, see RAS Down Mode and RAS Up Mode in section 6.5.10, Burst Operation.

Bit 2 RDM	Description	
0	DRAM interface: RAS up mode selected	(
1	DRAM interface: RAS down mode selected	

Bit 1—Self-Refresh Mode (SRFMD): Specifies DRAM self-refreshing in software s mode.

When any of areas 2 to 5 is designated as DRAM space, DRAM self-refreshing is pos transition is made to software standby mode after the SRFMD bit has been set to 1.

The normal access state is restored when software standby mode is exited, regardless SRFMD setting.

Bit 1 SRFMD	Description	
0	DRAM self-refreshing disabled in software standby mode	(
1	DRAM self-refreshing enabled in software standby mode	

6.2.8 DRAM Control Register B (DRCRB)

Bit	7	6	5	4	3	2	1
	MXC1	MXC0	CSEL	RCYCE	—	TPC	RCW
Initial value	0	0	0	0	1	0	0
Read/Write	R/W	R/W	R/W	R/W	—	R/W	R/W

DRCRB is an 8-bit readable/writable register that selects the number of address multipl address bits for the DRAM interface, the column address strobe output pin, enabling or of refresh cycle insertion, the number of precharge cycles, enabling or disabling of wait insertion between $\overline{RAS}$ and $\overline{CAS}$, and enabling or disabling of wait state insertion in re cycles.

DRCRB is initialized to H'08 by a reset and in hardware standby mode. It is not initial software standby mode.

The settings in this register are invalid when bits DRAS2 to DRAS0 in DRCRA are all

Bits 7 and 6—Multiplex Control 1 and 0 (MXC1, MXC0): These bits select the row address/column address multiplexing method used on the DRAM interface. In burst op row address used for comparison is determined by the setting of these bits and the bus w relevant area set in ABWCR.

	1	Column address: 9 bits		
		Compared address:		
		Modes 1, 2	8-bit access space	A_{19} to A_9
			16-bit access space	A_{19} to A_{10}
		Modes 3, 4, 5	8-bit access space	A_{23} to A_9
			16-bit access space	A_{23} to A_{10}
1	0	Column address: 10 bits		
		Compared address:		
		Modes 1, 2	8-bit access space	A_{19} to A_{10}
			16-bit access space	A_{19} to A_{11}
		Modes 3, 4, 5	8-bit access space	A_{23} to A_{10}
			16-bit access space	A_{23} to A_{11}
	1	Illegal setting		

Bit 5—$\overline{\text{CAS}}$ Output Pin Select (CSEL): Selects the $\overline{\text{UCAS}}$ and $\overline{\text{LCAS}}$ output pins wi to 5 are designated as DRAM space.

Bit 5 CSEL	Description	
0	PB4 and PB5 selected as $\overline{\text{UCAS}}$ and $\overline{\text{LCAS}}$ output pins	(
1	$\overline{\text{HWR}}$ and $\overline{\text{LWR}}$ selected as $\overline{\text{UCAS}}$ and $\overline{\text{LCAS}}$ output pins	

Bit 4—Refresh Cycle Enable (RCYCE): Enables or disables CAS-before-RAS refre insertion. When none of areas 2 to 5 has been designated as DRAM space, refresh cy inserted regardless of the setting of this bit.

Bit 4 RCYCE	Description	
0	Refresh cycles disabled	(
1	DRAM refresh cycles enabled	

1	2-state precharge cycle inserted

Bit 1—$\overline{RAS}$-$\overline{CAS}$ Wait (RCW): Controls wait state (Trw) insertion between T_r and T_c read/write cycles. The setting of this bit does not affect refresh cycles.

Bit 1 RCW	Description
0	Wait state (Trw) insertion disabled (In
1	One wait state (Trw) inserted

Bit 0—Refresh Cycle Wait Control (RLW): Controls wait state (T_{RW}) insertion for C RAS refresh cycles. The setting of this bit does not affect DRAM read/write cycles.

Bit 0 RLW	Description
0	Wait state (T_{RW}) insertion disabled (In
1	One wait state (T_{RW}) inserted

6.2.9 Refresh Timer Control/Status Register (RTMCSR)

Bit	7	6	5	4	3	2	1
	CMF	CMIE	CKS2	CKS1	CKS0	—	—
Initial value	0	0	0	0	0	1	1
Read/Write	R(W)*	R/W	R/W	R/W	R/W	—	—

Note: * Only 0 can be written to clear the flag.

RTMCSR is an 8-bit readable/writable register that selects the refresh timer counter clo the refresh timer is used as an interval timer, RTMCSR also enables or disables interru Bits 7 and 6 of RTMCSR are initialized to 0 by a reset and in the standby modes. Bits

	When the chip is reset and in standby mode Read CMF when CMF = 1, then write 0 in CMF	(
1	[Setting condition] When RTCNT = RTCOR	

Bit 6—Compare Match Interrupt Enable (CMIE): Enables or disables the CMI int requested when the CMF flag is set to 1 in RTMCSR. The CMIE bit is always cleare any of areas 2 to 5 is designated as DRAM space.

Bit 6 CMIE	Description	
0	The CMI interrupt requested by CMF is disabled	(
1	The CMI interrupt requested by CMF is enabled	

Bits 5 to 3—Refresh Counter Clock Select (CKS2 to CKS0): These bits select the c input to RTCNT from among 7 clocks obtained by dividing the system clock (ϕ). Wh clock is selected with bits CKS2 to CKS0, RTCNT begins counting up.

Bit 5 CKS2	Bit 4 CKS1	Bit 3 CKS0	Description	
0	0	0	Count operation halted	(In
		1	ϕ/2 used as counter clock	
	1	0	ϕ/8 used as counter clock	
		1	ϕ/32 used as counter clock	
1	0	0	ϕ/128 used as counter clock	
		1	ϕ/512 used as counter clock	
	1	0	ϕ/2048 used as counter clock	
		1	ϕ/4096 used as counter clock	

Bits 2 to 0—Reserved: These bits cannot be modified and are always read as 1.

RTCNT is incremented by an internal clock selected by bits CKS2 to CKS0 in RTMCS
RTCNT matches RTCOR (compare match), the CMF flag in RTMCSR is set to 1 and I
cleared to H'00. If the RCYCE bit in DRCRB is set to 1 at this time, a refresh cycle is
Also, if the CMIE bit in RTMCSR is set to 1, a compare match interrupt (CMI) is gene

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

6.2.11 Refresh Time Constant Register (RTCOR)

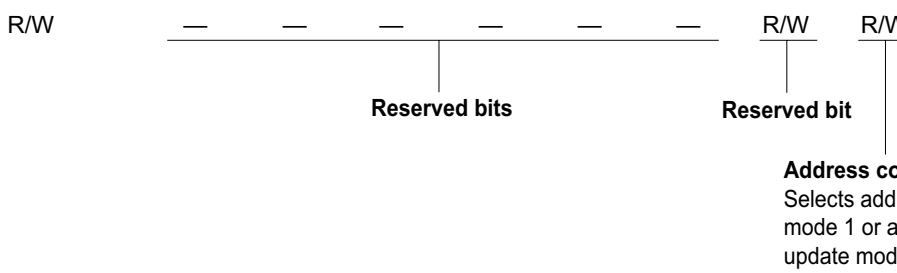
Bit	7	6	5	4	3	2	1
Initial value	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W

RTCOR is an 8-bit readable/writable register that determines the interval at which RTC
cleared.

RTCOR and RTCNT are constantly compared. When their values match, the CMF flag
in RTMCSR, and RTCNT is simultaneously cleared to H'00.

RTCOR is initialized to H'FF by a reset and in hardware standby mode. It is not initial
software standby mode.

Note: Only byte access can be used on this register.



ADRCR is initialized to H'FF by a reset and in hardware standby mode. It is not initia software standby mode.

Bits 7 to 2—Reserved: Read-only bits, always read as 1.

Bit 1—Reserved: Can be read or written to, but must not be cleared to 0.

Bit 0—Address Control (ADRCTL): Selects the address output method.

Bit 0 ADRCTL	Description	
0	Address update mode 2 is selected	
1	Address update mode 1 is selected	(

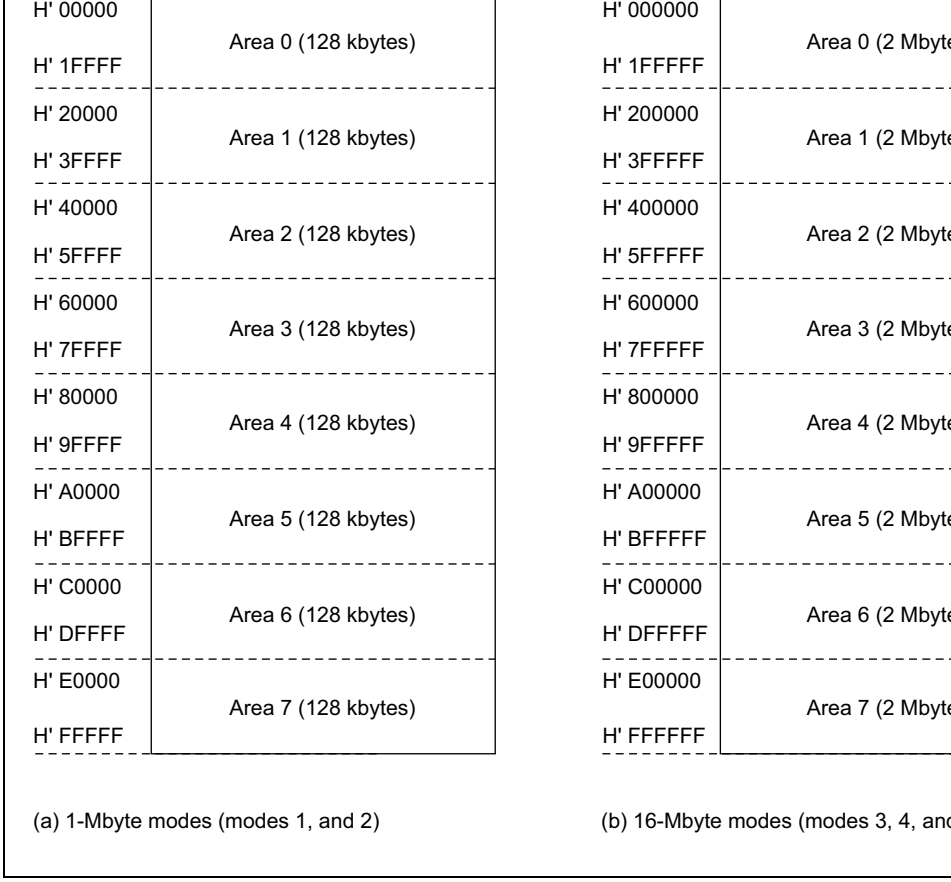
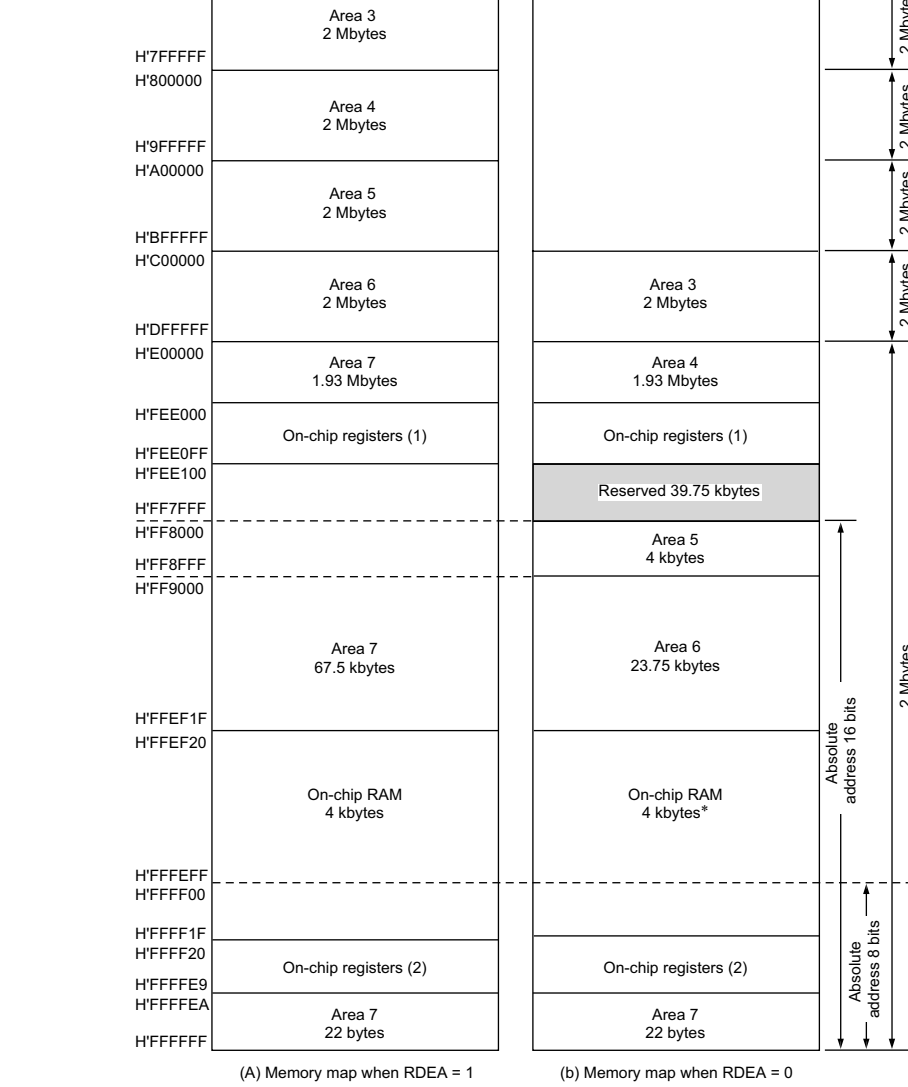


Figure 6.2 Access Area Map for Each Operating Mode

Chip select signals ($\overline{CS}_0$ to $\overline{CS}_7$) can be output for areas 0 to 7. The bus specifications f
area are selected in ABWCR, ASTCR, WCRH, and WCRL.

In 16-Mbyte mode, the area division units can be selected with the RDEA bit in BCR.



Note: * Area 6 when the RAME bit is cleared.

Figure 6.3 Memory Map in 16-Mbyte Mode

bus is selected functions as an 8-bit access space, and an area for which a 16-bit bus is
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 designat
bit access, 16-bit bus mode is set.

Number of Access States: Two or three access states can be selected with ASTCR. A
which two-state access is selected functions as a two-state access space, and an area for
three-state access is selected functions as a three-state access space.

DRAM space is accessed in four states regardless of the ASTCR settings.

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

Number of Program Wait States: When three-state access space is designated in AST
number of program wait states to be inserted automatically is selected with WCRH and
From 0 to 3 program wait states can be selected.

When ASTCR is cleared to 0 for DRAM space, a program wait (T_{c1}-T_{c2} wait) is not ins
Also, no program wait is inserted in burst ROM space burst cycles.

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

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

Note: n = 7 to 0

6.3.3 Memory Interfaces

The H8/3028 Group memory interfaces comprise a basic bus interface that allows dire connection of ROM, SRAM, and so on; a DRAM interface that allows direct connecti DRAM; and a burst ROM interface that allows direct connection of burst ROM. The be selected independently for each area.

An area for which the basic bus interface is designated functions as normal space, an a which the DRAM interface is designated functions as DRAM space, and area 0 for wh ROM interface is designated functions as burst ROM space.

In the expanded modes with on-chip ROM disabled, a reset leaves pin $\overline{CS}_0$ in the output pins $\overline{CS}_1$ to $\overline{CS}_3$ in the input state. To output chip select signals $\overline{CS}_1$ to $\overline{CS}_3$, the corresp DDR bits must be set to 1. In the expanded modes with on-chip ROM enabled, a reset $\overline{CS}_0$ to $\overline{CS}_3$ in the input state. To output chip select signals $\overline{CS}_0$ to $\overline{CS}_3$, the correspondi bits must be set to 1. For details, see section 8, I/O Ports.

Output of $\overline{CS}_4$ to $\overline{CS}_7$: Output of $\overline{CS}_4$ to $\overline{CS}_7$ is enabled or disabled in the chip select co register (CSCR). A reset leaves pins $\overline{CS}_4$ to $\overline{CS}_7$ in the input state. To output chip selec $\overline{CS}_4$ to $\overline{CS}_7$, the corresponding CSCR bits must be set to 1. For details, see section 8, I/

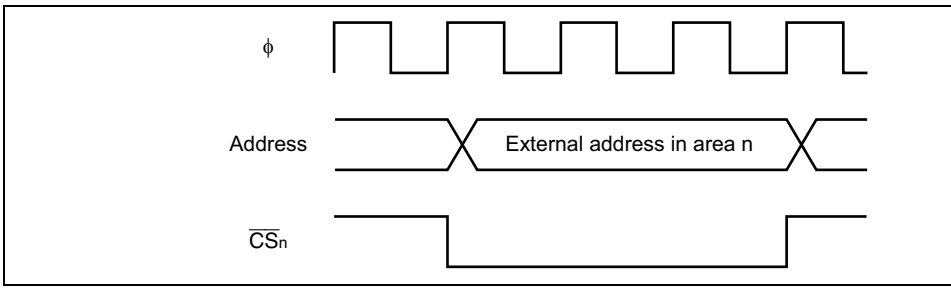


Figure 6.4 $\overline{CS}$n Signal Output Timing (n = 0 to 7)

When the on-chip ROM, on-chip RAM, and on-chip registers are accessed, $\overline{CS}_0$ to $\overline{CS}_7$ high. The $\overline{CS}_n$ signals are decoded from the address signals. They can be used as chip signals for SRAM and other devices.

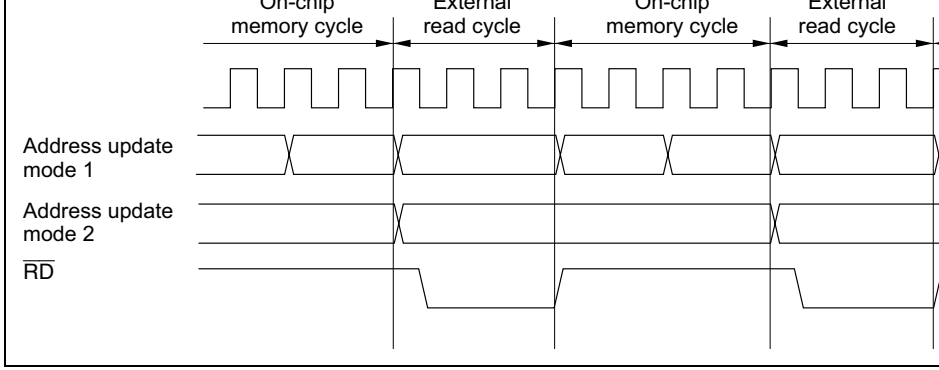


Figure 6.5 Sample Address Output in Each Address Update Mode (Basic Bus Interface, 3-State Space)

Address Update Mode 1: Address update mode 1 is compatible with the previous H Series. Addresses are always updated between bus cycles.

Address Update Mode 2: In address update mode 2, address updating is performed o external space accesses or self-refresh cycles. In this mode, the address can be retaine an external space read cycle and an instruction fetch cycle (on-chip memory) by placi program in on-chip memory. Address update mode 2 is therefore useful when connec that requires address hold time with respect to the rise of the $\overline{RD}$ strobe.

Switching between address update modes 1 and 2 is performed by means of the ADR ADRCR. The initial value of ADRCR is the address update mode 1 setting, providing compatibility with the previous H8/300H Series.

cycles (area inside the ellipse in the figure).

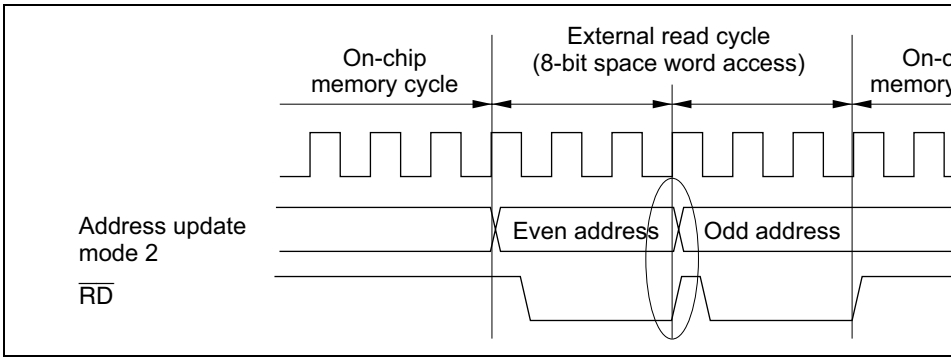


Figure 6.6 Example of Consecutive External Space Accesses in Address Updat

- When address update mode 2 is selected, in a DRAM space CAS-before-RAS (CBF cycle the previous address is retained (the area 2 start address is not output).

6.4.2 Data Size and Data Alignment

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

8-Bit Access Areas: Figure 6.7 illustrates data alignment control for 8-bit access spac
bit access space, the upper data bus (D_{15} to D_8) is always used for accesses. The amou
that can be accessed at one time is one byte: a word access is performed as two byte a
a longword access, as four byte accesses.

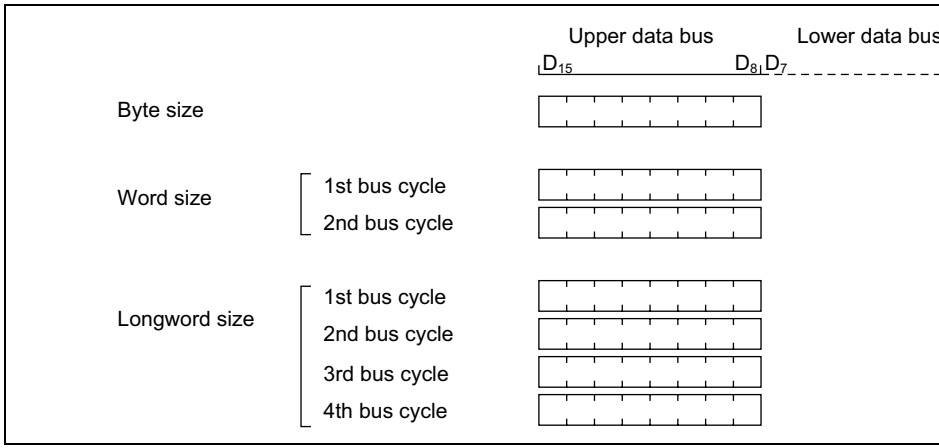


Figure 6.7 Access Sizes and Data Alignment Control (8-Bit Access Ar

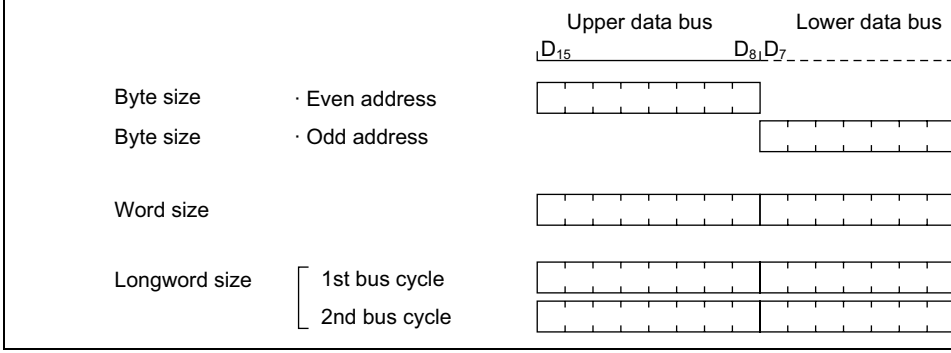


Figure 6.8 Access Sizes and Data Alignment Control (16-Bit Access Are

6.4.3 Valid Strobes

Table 6.4 shows the data buses used, and the valid strobes, for the access spaces.

In a read, the $\overline{RD}$ signal is valid for both the upper and the lower half of the data bus.

In a write, the $\overline{HWR}$ signal is valid for the upper half of the data bus, and the $\overline{LWR}$ sign
lower half.

area		Write	Even	$\overline{HWR}$	Valid	Und data
			Odd	$\overline{LWR}$	Undetermined data	Vali
	Word	Read	—	$\overline{RD}$	Valid	Vali
		Write	—	$\overline{HWR}$, $\overline{LWR}$	Valid	Vali

Notes: 1. Undetermined data means that unpredictable data is output.
2. Invalid means that the bus is in the input state and the input is ignored.

6.4.4 Memory Areas

The initial state of each area is basic bus interface, three-state access space. The initia is selected according to the operating mode. The bus specifications described here co items only, and the following sections should be referred to for further details: section Bus Interface, section 6.5, DRAM Interface, section 6.8, Burst ROM Interface.

Area 0: Area 0 includes on-chip ROM, and in ROM-disabled expansion mode, all of external space. In ROM-enabled expansion mode, the space excluding on-chip ROM space.
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 128 kbytes in modes 1 and 2, and 2 Mbytes in modes 3, 4, and 5.

Areas 1 and 6: In external expansion mode, areas 1 and 6 are entirely 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 128 kbytes in modes 1 and 2, and 2 Mbytes in modes 3, 4,

0, the on-chip RAM is disabled and the corresponding space becomes external space .
When area 7 external space is accessed, the $\overline{CS_7}$ signal can be output.
Only the basic bus interface can be used for the area 7 memory interface.
The size of area 7 is 128 kbytes in modes 1 and 2, and 2 Mbytes in modes 3, 4, and 5.

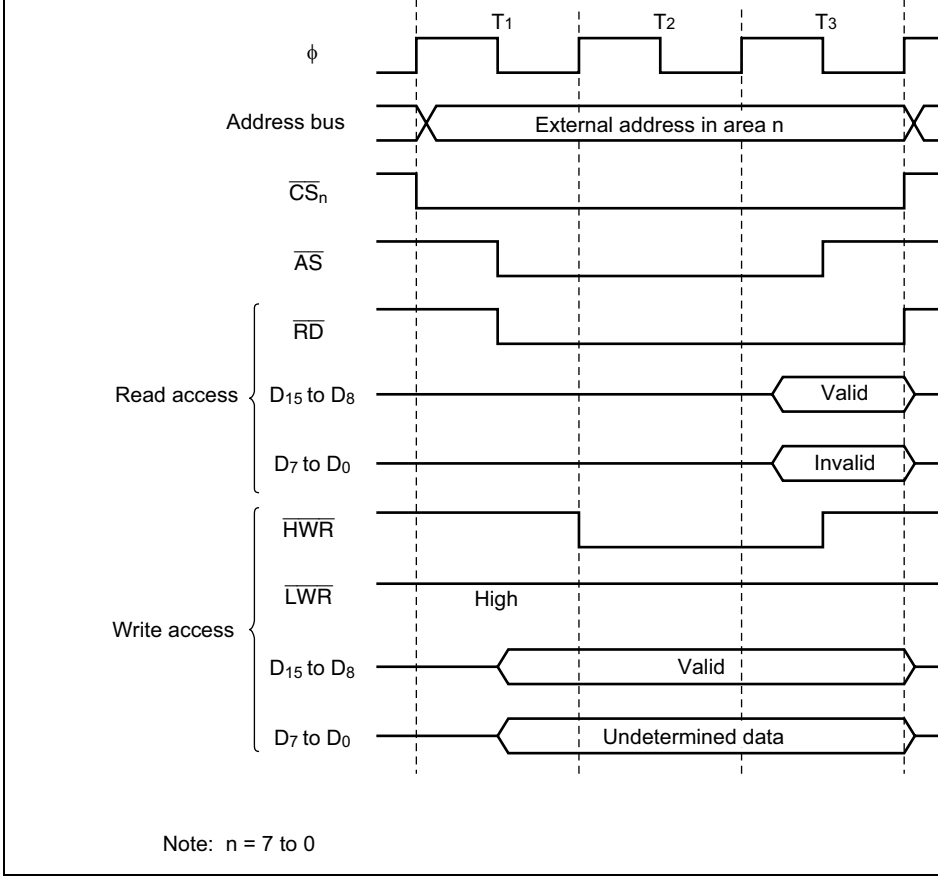


Figure 6.9 Bus Control Signal Timing for 8-Bit, Three-State-Access Ar

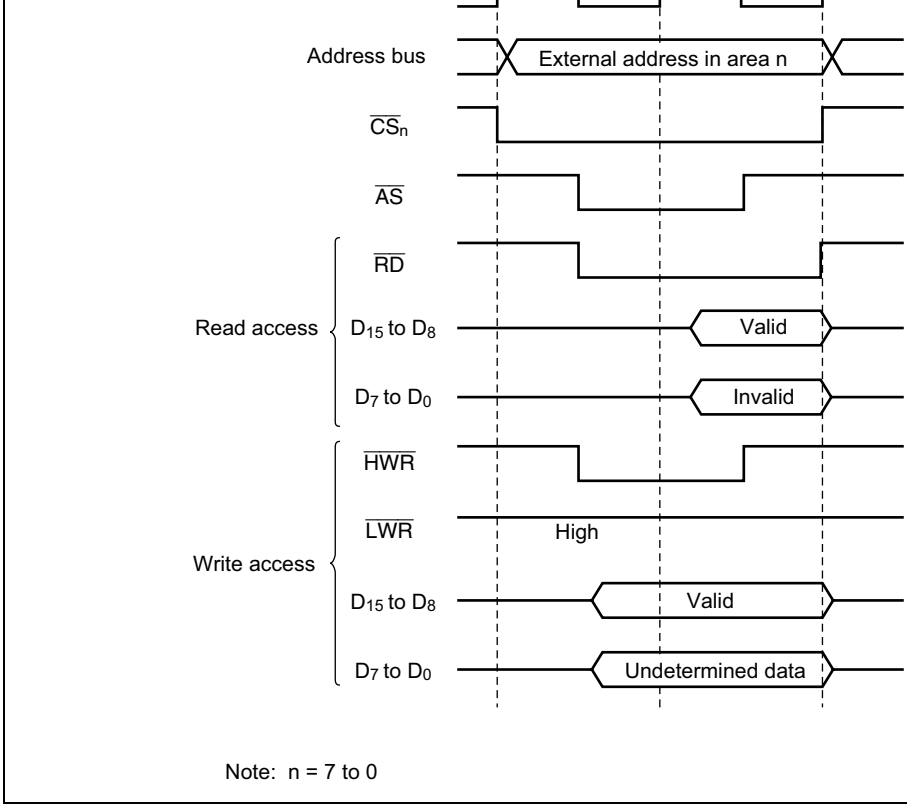


Figure 6.10 Bus Control Signal Timing for 8-Bit, Two-State-Access Are

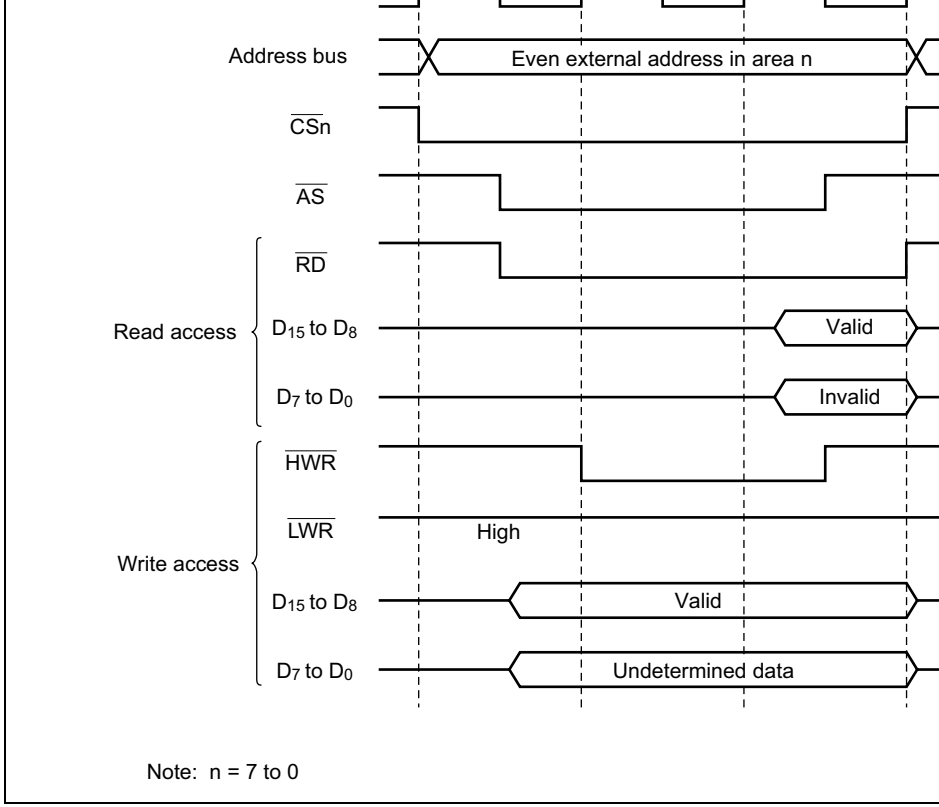


Figure 6.11 Bus Control Signal Timing for 16-Bit, Three-State-Access Ar
(Byte Access to Even Address)

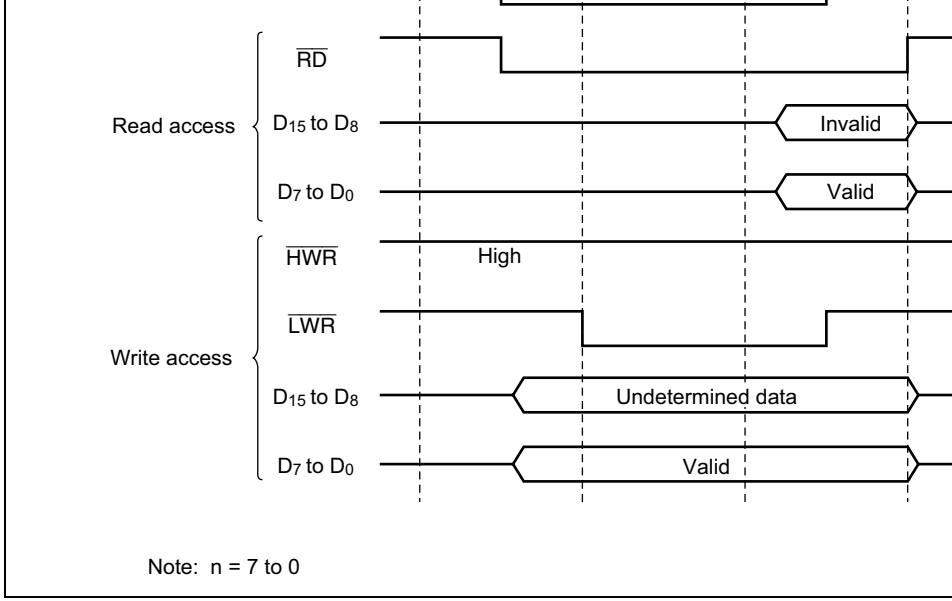


Figure 6.12 Bus Control Signal Timing for 16-Bit, Three-State-Access Are (Byte Access to Odd Address)

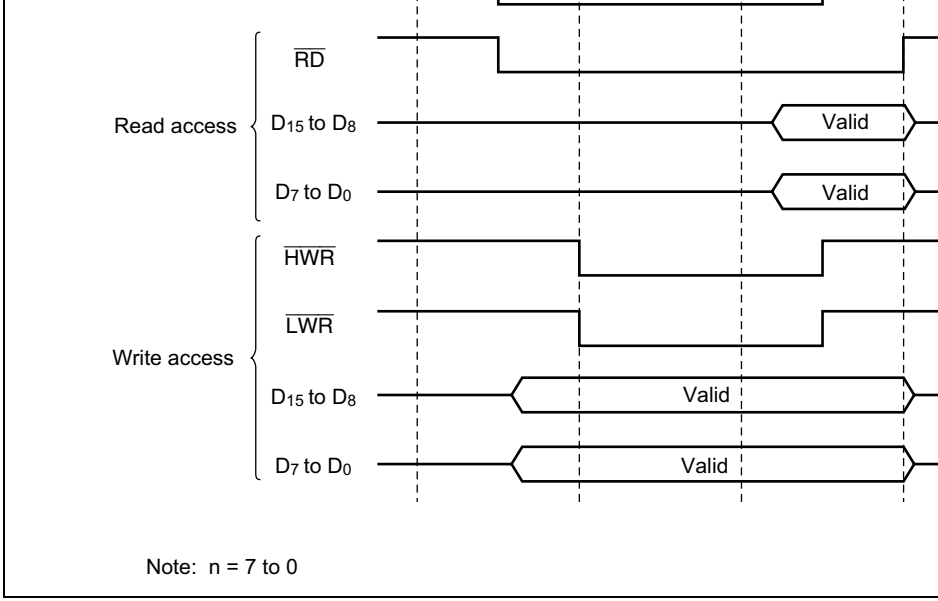


Figure 6.13 Bus Control Signal Timing for 16-Bit, Three-State-Access Ar (Word Access)

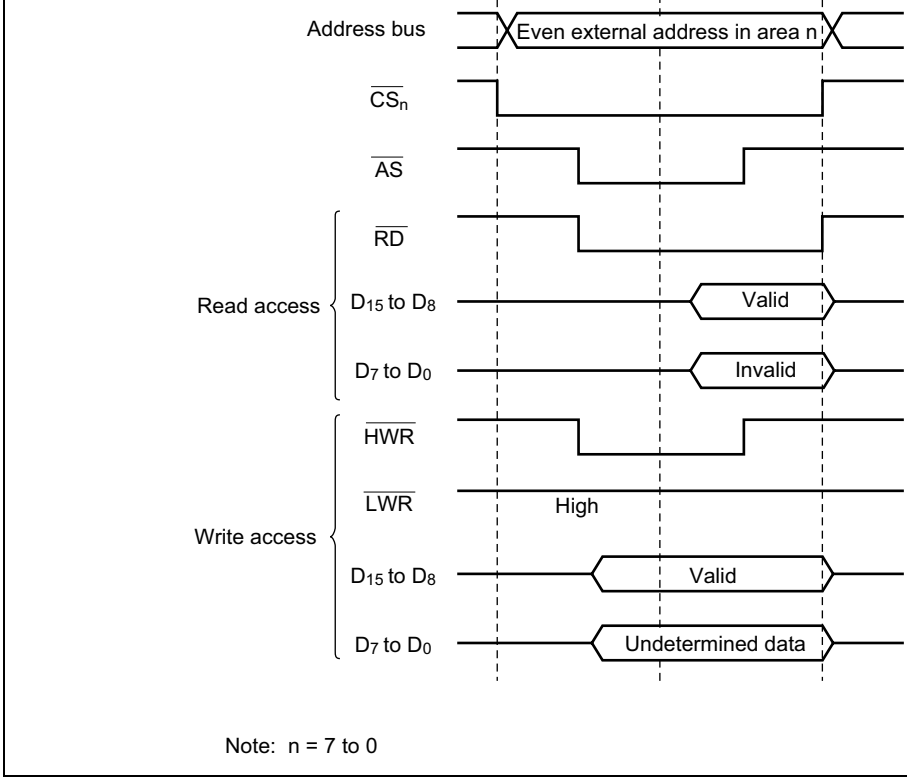


Figure 6.14 Bus Control Signal Timing for 16-Bit, Two-State-Access Area (Byte Access to Even Address)

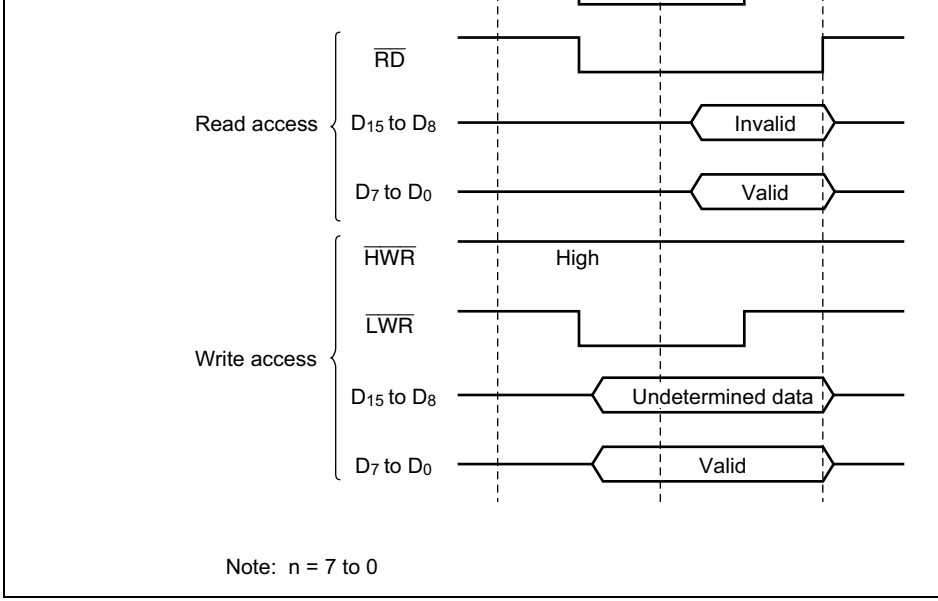


Figure 6.15 Bus Control Signal Timing for 16-Bit, Two-State-Access Are (Byte Access to Odd Address)

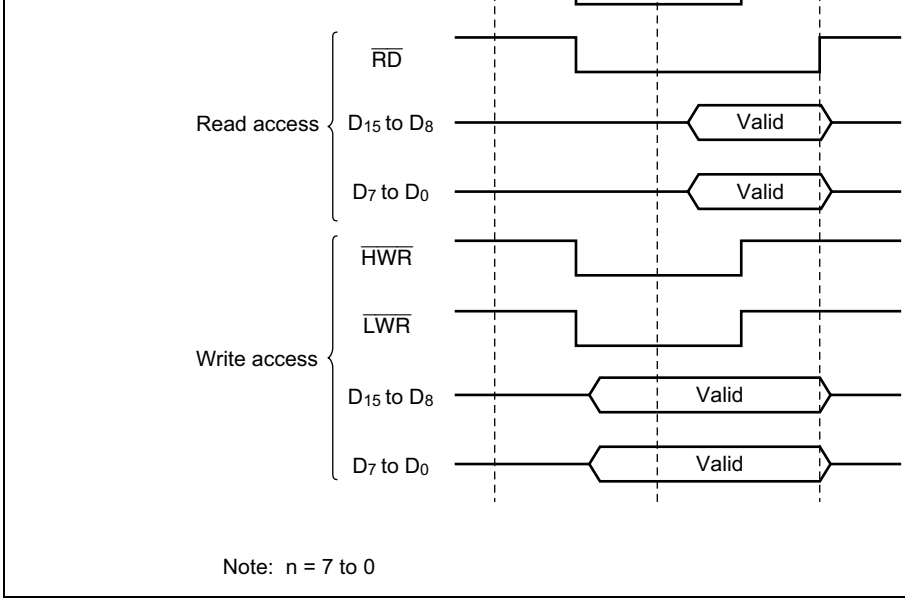


Figure 6.16 Bus Control Signal Timing for 16-Bit, Two-State-Access Area (Word Access)

6.4.6 Wait Control

When accessing external space, the H8/3028 Group can extend the bus cycle by insertin more wait states (T_w). There are two ways of inserting wait states: (1) program wait ins (2) pin wait insertion using the $\overline{\text{WAIT}}$ pin.

Program Wait Insertion: From 0 to 3 wait states can be inserted automatically betwee state and T_3 state on an individual area basis in three-state access space, according to th of WCRH and WCRL.

Figure 6.17 shows an example of the timing for insertion of one program wait state in space.

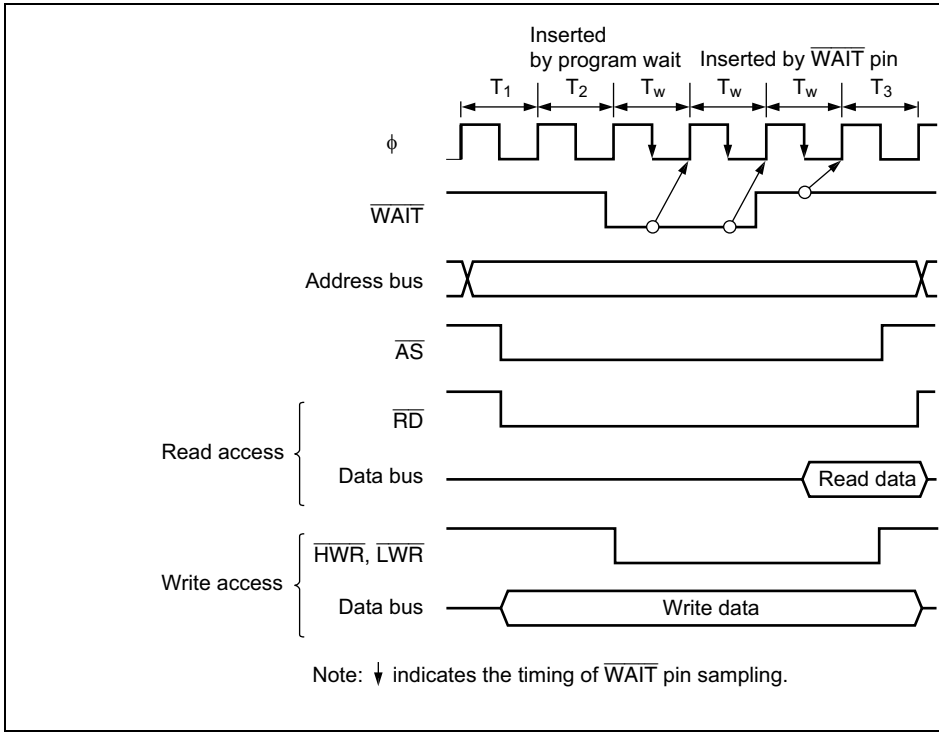


Figure 6.17 Example of Wait State Insertion Timing

used for byte access control. In the case of × 16-bit organization DRAM, therefore, the type can be connected. A fast page mode is supported in addition to the normal read ar access modes.

6.5.2 DRAM Space and $\overline{\text{RAS}}$ Output Pin Settings

Designation of areas 2 to 5 as DRAM space, and selection of the $\overline{\text{RAS}}$ output pin for ea designated as DRAM space, is performed by setting bits in DRCRA. Table 6.5 shows correspondence between the settings of bits DRAS2 to DRAS0 and the selected DRAM $\overline{\text{RAS}}$ output pin.

When an arbitrary value has been set in DRAS2 to DRAS0, a write of a different value 000 must not be performed.

		1	Normal space	Normal space	DRAM space ($\overline{CS_2}$)*	DRA ($\overline{CS}$
1	0	0	Normal space	DRAM space ($\overline{CS_4}$)	DRAM space ($\overline{CS_3}$)	DRA ($\overline{CS}$
		1	DRAM space ($\overline{CS_5}$)	DRAM space ($\overline{CS_4}$)	DRAM space ($\overline{CS_3}$)	DRA ($\overline{CS}$
	1	0	DRAM space ($\overline{CS_4}$)*	DRAM space ($\overline{CS_4}$)*	DRAM space ($\overline{CS_2}$)*	DRA ($\overline{CS}$
		1	DRAM space ($\overline{CS_2}$)*	DRAM space ($\overline{CS_2}$)*	DRAM space ($\overline{CS_2}$)*	DRA ($\overline{CS}$

Note: * A single $\overline{CS_n}$ pin serves as a common $\overline{RAS}$ output pin for a number of areas. U pins can be used as input/output ports.

	DRCRB		Address	Address Pins									
	MXC1	MXC0	Bits	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
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}
		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}
	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}
		1	Illegal setting	—	—	—	—	—	—	—	—	—	—	—
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

Note: * Row address bit A_{20} is not multiplexed in 1-Mbyte mode.

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 organization 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.

6.5.5 Pins Used for DRAM Interface

Table 6.7 shows the pins used for DRAM interfacing and their functions.

		address strobe		space access (when CSEL = 1
$\overline{LWR}$	$\overline{LCAS}$	Lower column address strobe	Output	Lower column address strobe f space access (when CSEL = 1
$\overline{CS_2}$	$\overline{RAS_2}$	Row address strobe 2	Output	Row address strobe for DRAM access
$\overline{CS_3}$	$\overline{RAS_3}$	Row address strobe 3	Output	Row address strobe for DRAM access
$\overline{CS_4}$	$\overline{RAS_4}$	Row address strobe 4	Output	Row address strobe for DRAM access
$\overline{CS_5}$	$\overline{RAS_5}$	Row address strobe 5	Output	Row address strobe for DRAM access
$\overline{RD}$	$\overline{WE}$	Write enable	Output	Write enable for DRAM space v access*
P80	$\overline{RFSH}$	Refresh	Output	Goes low in refresh cycle
A_{12} to A_0	A_{12} to A_0	Address	Output	Row address/column address r output
D_{15} to D_0	D_{15} to D_0	Data	I/O	Data input/output pins

Note: * Fixed high in a read access.

6.5.6 Basic Timing

Figure 6.18 shows the basic access timing for DRAM space. The basic DRAM access four states: one precharge cycle (T_p) state, one row address output cycle (T_r) state, and address output cycle (T_{c1}, T_{c2}) states. Unlike the basic bus interface, the correspondin ASTCR control only enabling or disabling of wait insertion between T_{c1} and T_{c2}, and the number of access states. When the corresponding bit in ASTCR is cleared to 0, w cannot be inserted between T_{c1} and T_{c2} in the DRAM access cycle.

If a DRAM read/write cycle is followed by an access cycle for an external area other t space when $\overline{HWR}$ and $\overline{LWR}$ are selected as the $\overline{UCAS}$ and $\overline{LCAS}$ output pins, an idle inserted unconditionally immediately after the DRAM access cycle. See section 6.9, for details.

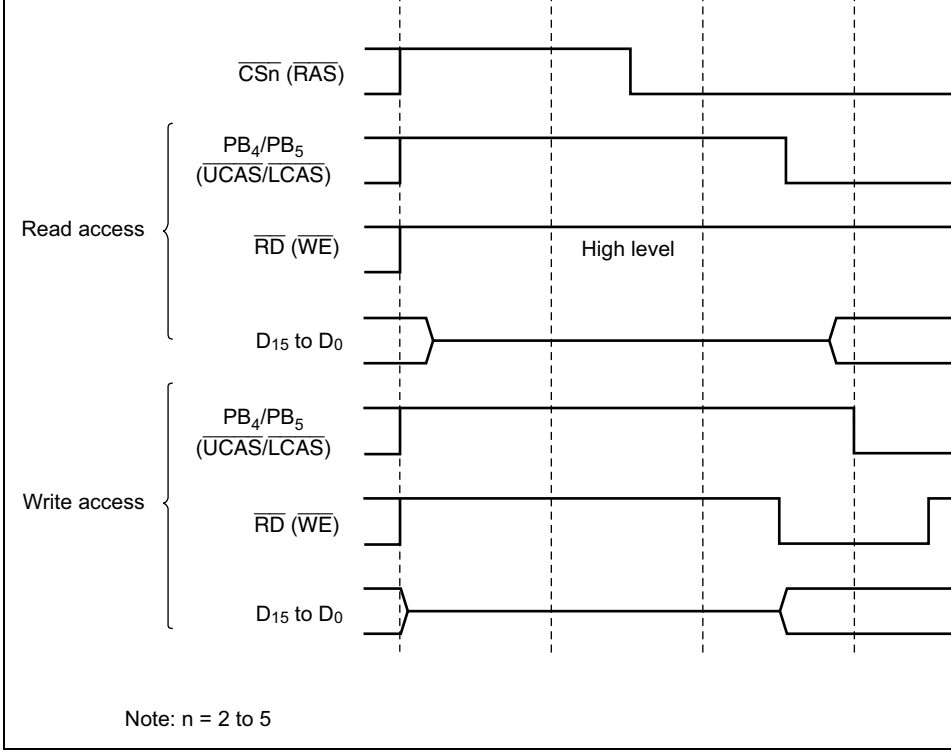


Figure 6.18 Basic Access Timing (CSEL = 0 in DRCRB)

6.5.7 Precharge State Control

In the H8/3028 Group, provision is made for the DRAM RAS precharge time by alway one RAS precharge state (T_p) when DRAM space is accessed. This can be changed to states by setting the TPC bit to 1 in DRCRB. The optimum number of T_p cycles should according to the DRAM connected and the operating frequency of the H8/3028 Group Figure 6.19 shows the timing when two T_p states are inserted.

When the TCP bit is set to 1, two T_p states are also used for CAS-before-RAS refresh

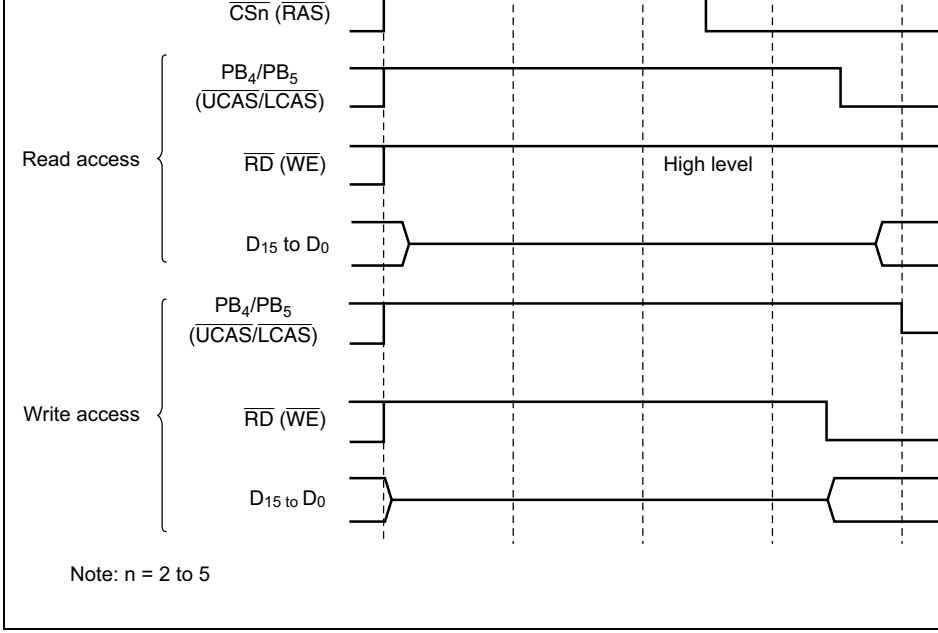


Figure 6.19 Timing with Two Precharge States (CSEL = 0 in DRCRB

6.5.8 Wait Control

In a DRAM access cycle, wait states can be inserted (1) between the T_r state and T_{c1} s between the T_{c1} state and T_{c2} state.

Insertion of T_{rw} Wait State between T_r and T_{c1}: One T_{rw} state can be inserted betwe T_{c1} by setting the RCW bit to 1 in DRCRB.

Insertion of T_w Wait State(s) between T_{c1} and T_{c2}: When the bit in ASTCR corresp area designated as DRAM space is set to 1, from 0 to 3 wait states can be inserted bet state and T_{c2} state by means of settings in WCRH and WCRL.

Figure 6.20 shows an example of the timing for wait state insertion.

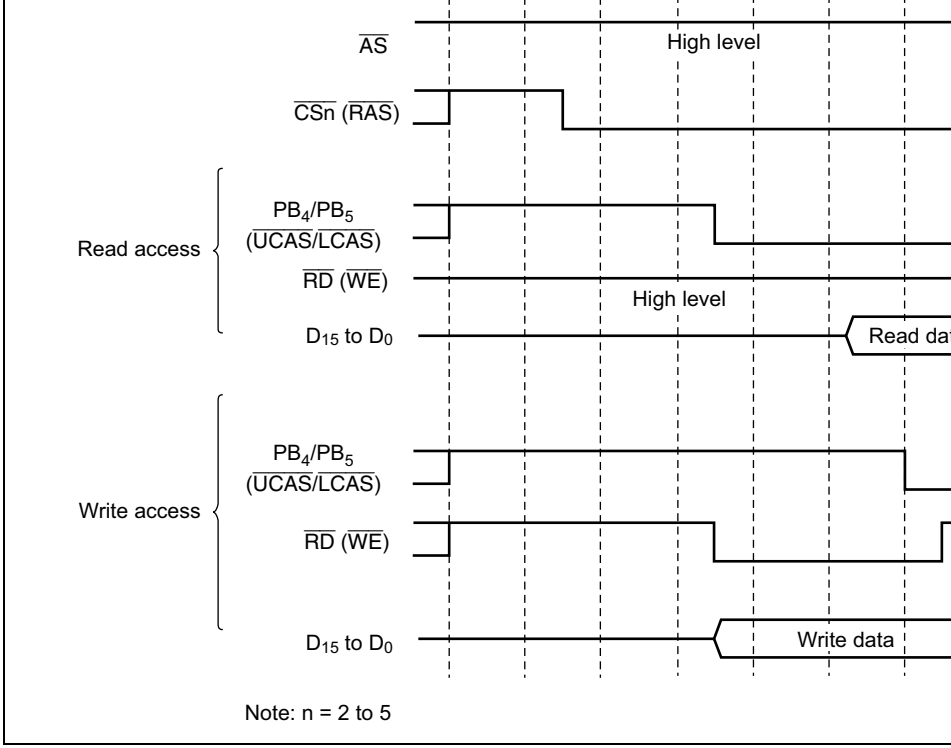


Figure 6.20 Example of Wait State Insertion Timing (CSEL = 0)

6.5.9 Byte Access Control and $\overline{\text{CAS}}$ Output Pin

When an access is made to DRAM space designated as a 16-bit-access area in ABWCR address strobes ($\overline{\text{UCAS}}$ and $\overline{\text{LCAS}}$) corresponding to the upper and lower halves of the data bus are output. In the case of × 16-bit organization DRAM, the 2-CAS type can be connected.

Either PB4 and PB5, or $\overline{\text{HWR}}$ and $\overline{\text{LWR}}$, can be used as the $\overline{\text{UCAS}}$ and $\overline{\text{LCAS}}$ output pi selection being made with the CSEL bit in DRCRB. Table 6.8 shows the CSEL bit sett corresponding output pin selections.

Table 6.8 CSEL Settings and UCAS and LCAS Output Pins

CSEL	$\overline{\text{UCAS}}$	$\overline{\text{LCAS}}$
0	PB_4	PB_5
1	$\overline{\text{HWR}}$	$\overline{\text{LWR}}$

Figure 6.21 shows the control timing.

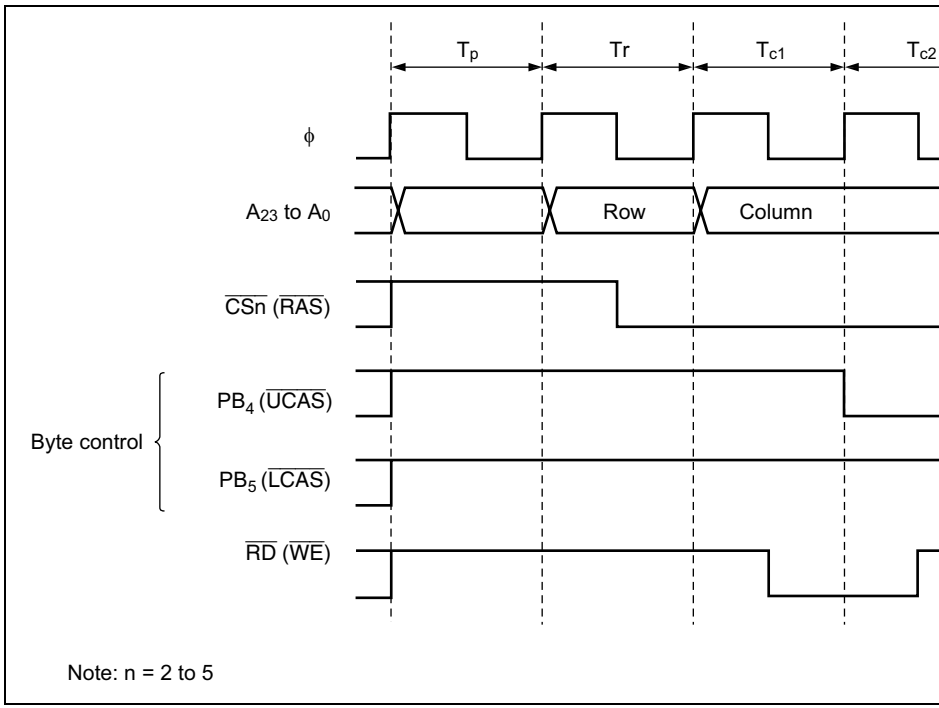


Figure 6.21 Control Timing (Upper-Byte Write Access When CSEL =

burst access. When there are consecutive access cycles for DRAM space, the column a
$\overline{CAS}$ signal output cycles (two states) continue as long as the row address is the same fo
consecutive access cycles. In burst access, too, the bus cycle can be extended by insert
states between T_{c1} and T_{c2}. The wait state insertion method and timing are the same as
access: see section 6.5.8, Wait Control, for details.

The row address used for the comparison is determined by the bus width of the relevan
bits MXC1 and MXC0 in BRCRB, and in ABWCR. Table 6.9 shows the compared ro
corresponding to the various settings of bits MXC1 and MXC0, and ABWCR.

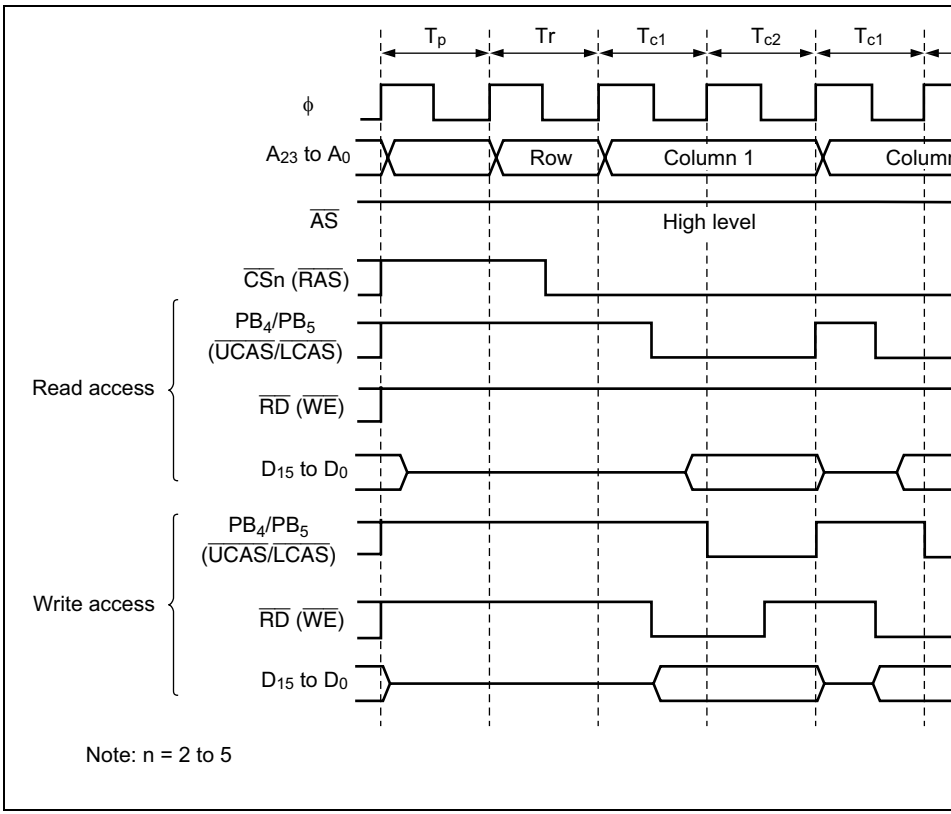


Figure 6.22 Operation Timing in Fast Page Mode

			1	8 bits	A_{19} to A_9
	1	0	0	16 bits	A_{19} to A_{11}
			1	8 bits	A_{19} to A_{10}
		1	—	—	Illegal setting
Modes 3, 4, and 5 (16-Mbyte)	0	0	0	16 bits	A_{23} to A_9
			1	8 bits	A_{23} to A_8
		1	0	16 bits	A_{23} to A_{10}
			1	8 bits	A_{23} to A_9
	1	0	0	16 bits	A_{23} to A_{11}
			1	8 bits	A_{23} to A_{10}
		1	—	—	Illegal setting

Note: n = 2 to 5

shows an example of the timing in RAS down mode.

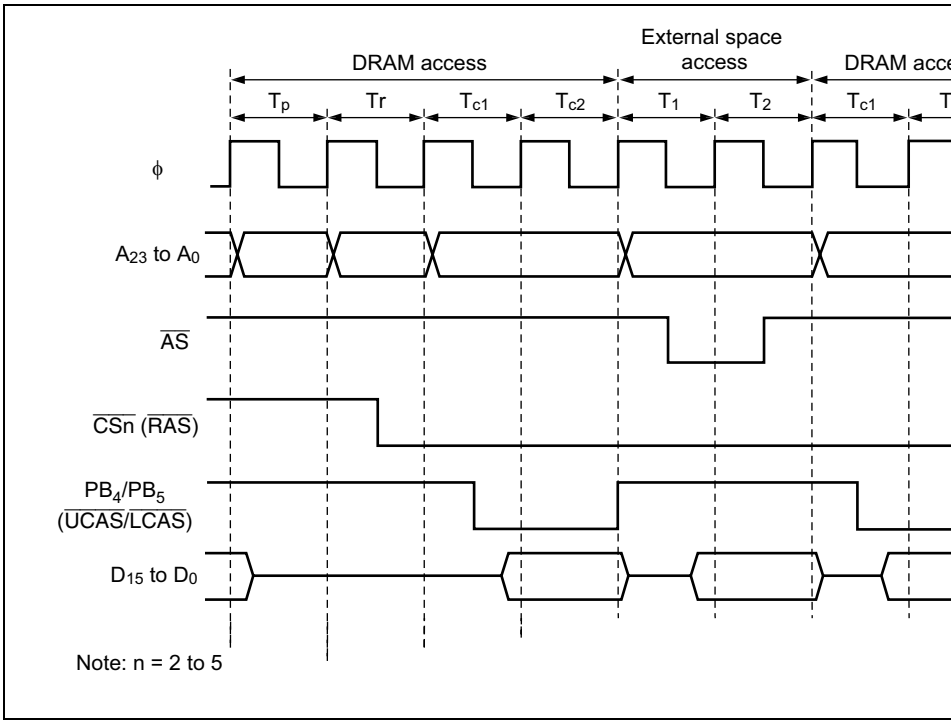
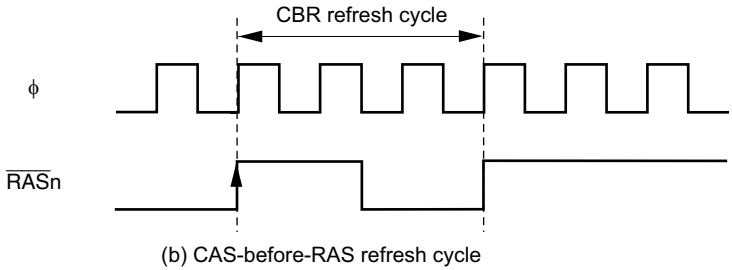


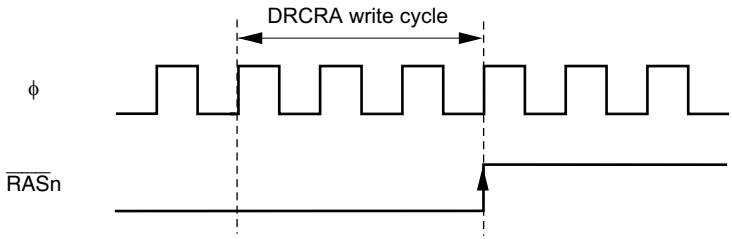
Figure 6.23 Example of Operation Timing in RAS Down Mode (CSEL =

When RAS down mode is selected, the conditions for an asserted RASn signal to re high level are as shown below. The timing in these cases is shown in figure 6.24.

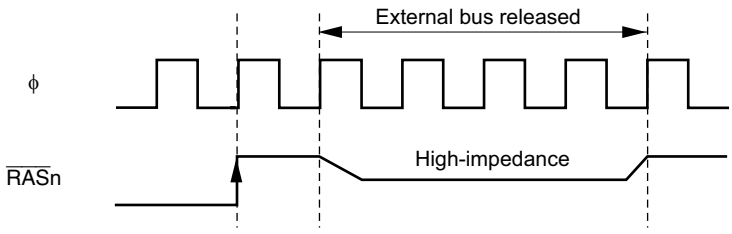
— When DRAM space with a different row address is accessed
— Immediately before a CAS-before-RAS refresh cycle
— When the BE bit or RDM bit is cleared to 0 in DRCRA
— Immediately before release of the external bus



(b) CAS-before-RAS refresh cycle



(c) BE bit or RDM bit cleared to 0 in DRCRA



(d) External bus released

Note: n = 2 to 5

Figure 6.24 $\overline{\text{RAS}}$n Negation Timing when RAS Down Mode is Selecte

$\overline{\text{LCAS}}$, a device other than DRAM is connected to external space, and $\overline{\text{HWR}}$ and $\overline{\text{LV}}$ used as write strobes.

- RAS Up Mode

  To select RAS up mode, clear the RDM bit to 0 in DRCRA. Each time access to D is interrupted and another space is accessed, the $\overline{\text{RAS}}$ signal returns to the high leve operation is only performed if DRAM space is continuous. Figure 6.25 shows an ex the timing in RAS up mode.

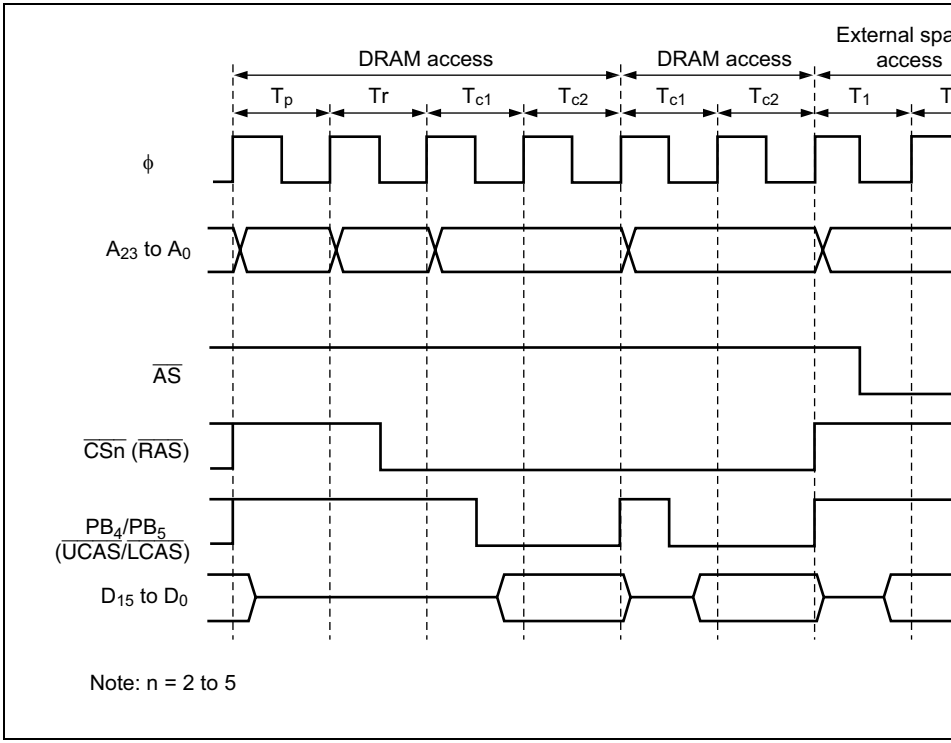


Figure 6.25 Example of Operation Timing in RAS Up Mode

RTMCSR, and a refresh request is generated when the count matches the value set in
(compare match). At the same time, RTCNT is reset and starts counting up again fro
Refreshing is thus repeated at fixed intervals determined by RTCOR and bits CKS2 t
refresh cycle is executed after this refresh request has been accepted and the DRAM i
acquired the bus. Set a value in bits CKS2 to CKS0 in RTCOR that will meet the refr
specification for the DRAM used. When RAS down mode is used, set the refresh inte
the maximum $\overline{\text{RAS}}$ pulse width specification is met.

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

Also note that a repeat refresh request generated during a bus request, or a refresh req
refresh cycle execution, will be ignored.

RTCNT operation is shown in figure 6.26, compare match timing in figure 6.27, and C
timing in figures 6.28 and 6.29.

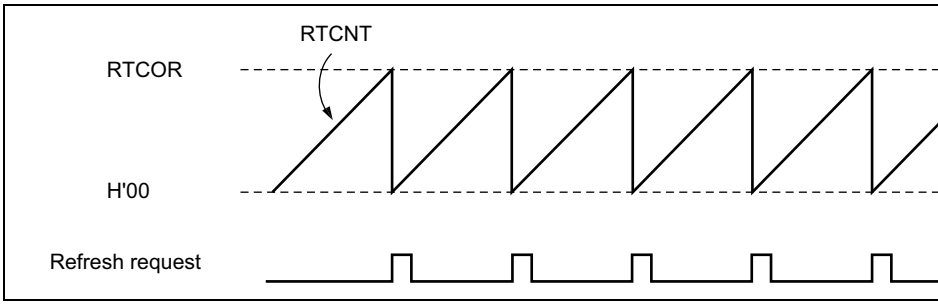


Figure 6.26 RTCNT Operation

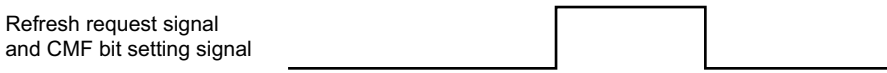
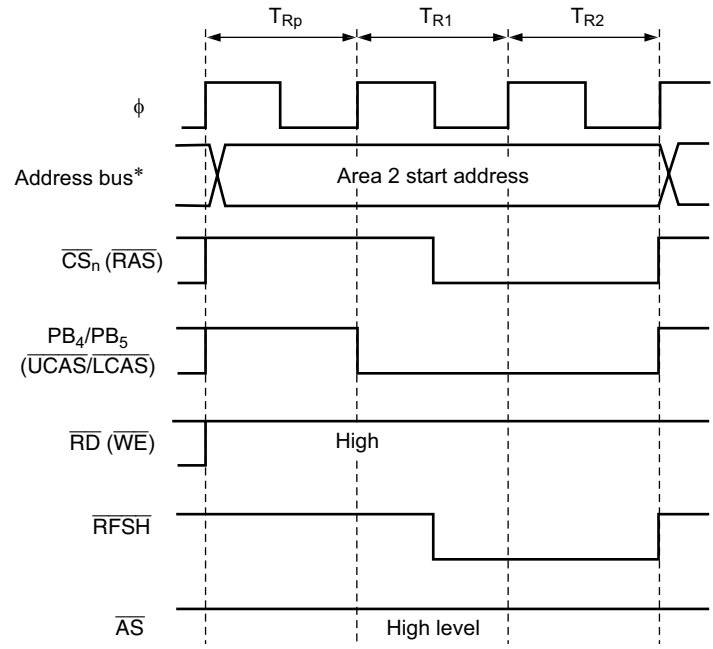


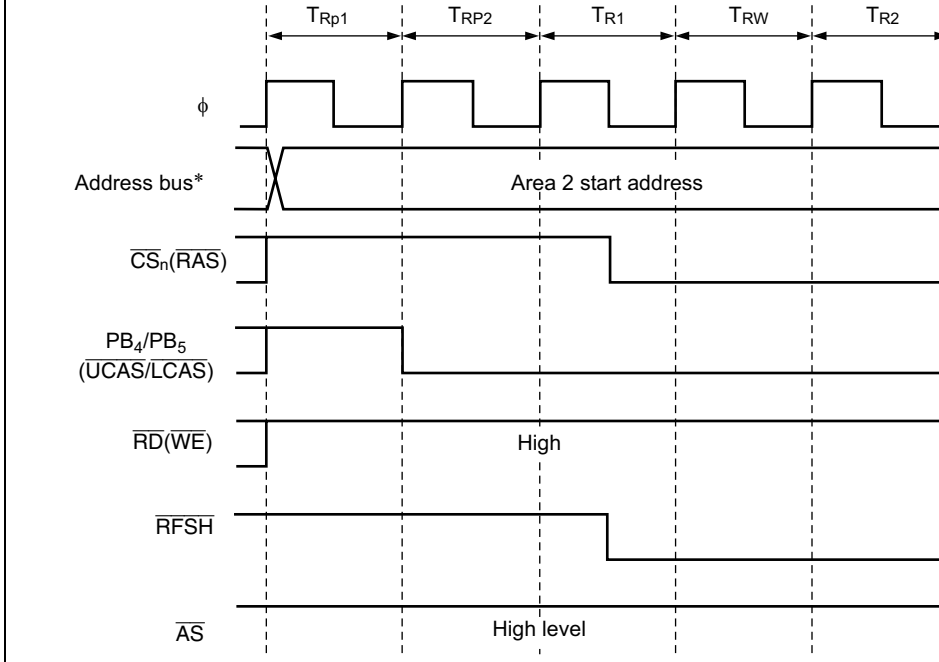
Figure 6.27 Compare Match Timing



Note: * In address update mode 1, the area 2 start address is output.
In address update mode 2, the address in the preceding bus cycle is retained.

Figure 6.28 CBR Refresh Timing (CSEL = 0, TPC = 0, RLW = 0)

The basic CBS refresh cycle timing comprises three states: one RAS precharge cycle (T
and two RAS output cycle (T_{R1}, T_{R2}) states. Either one or two states can be selected for
precharge cycle. When the TPC bit is set to 1 in DRCRB, $\overline{RAS}$ signal output is delayed
cycle. This does not affect the timing of $\overline{UCAS}$ and $\overline{LCAS}$ output.



Note: * In address update mode 1, the area 2 start address is output.
In address update mode 2, the address in the preceding bus cycle is retained.

Figure 6.29 CBR Refresh Timing (CSEL = 0, TPC = 1, RLW = 1)

DRAM must be refreshed immediately after powering on in order to stabilize its inter
When using the H8/3028 Group CAS-before-RAS refresh function, therefore, a DRA
stabilization period should be provided by means of interrupts by another timer modul
counting the number of times bit 7 (CMF) of RTMCSR is set, for instance, immediate
DRAS2 to DRAS0 have been set in DRCRA.

Self-Refreshing: A self-refresh mode (battery backup mode) is provided for DRAM a
standby mode. In this mode, refresh timing and refresh addresses are generated withi
DRAM. The H8/3028 Group has a function that places the DRAM in self-refresh mo
chip enters software standby mode.

selected, the RDM bit in DRCRA must be cleared to 0 and RAS up mode selected b
executing the SLEEP instruction. Select RAS down mode again after exiting softw
mode.

- The instruction immediately following a SLEEP instruction must not be located in a
designated as DRAM space.

The self-refresh function will not work properly unless the above conditions are observ

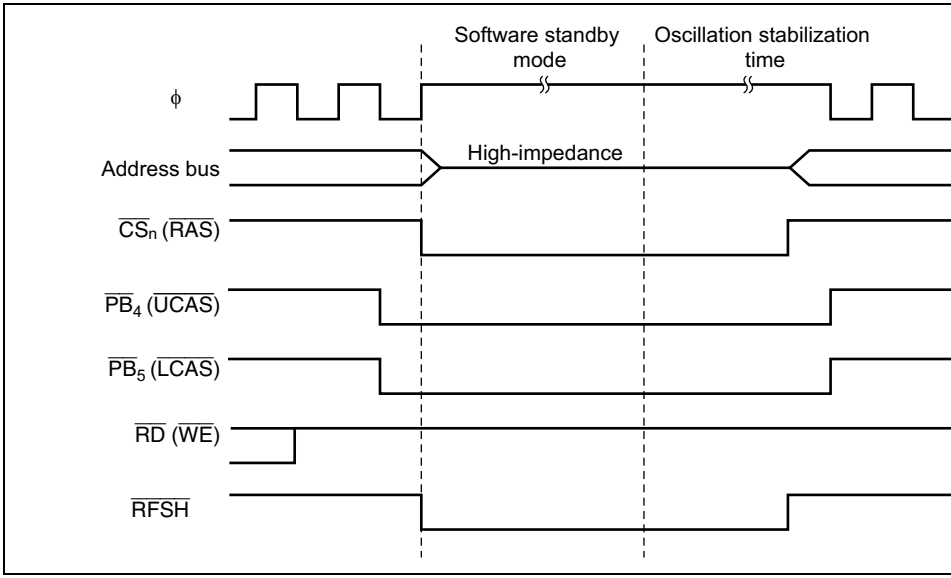


Figure 6.30 Self-Refresh Timing (CSEL = 0)

Refresh Signal ($\overline{\text{RFSH}}$): A refresh signal ($\overline{\text{RFSH}}$) that transmits a refresh cycle off-chi
output by setting the RFSHE bit to 1 in DRCRA. $\overline{\text{RFSH}}$ output timing is shown in figu
6.29, and 6.30.

using a × 16-bit organization, and the corresponding address map. The DRAMs u
example are of the 10-bit row address × 10-bit column address type. Up to four D
be connected by designating areas 2 to 5 as DRAM space.

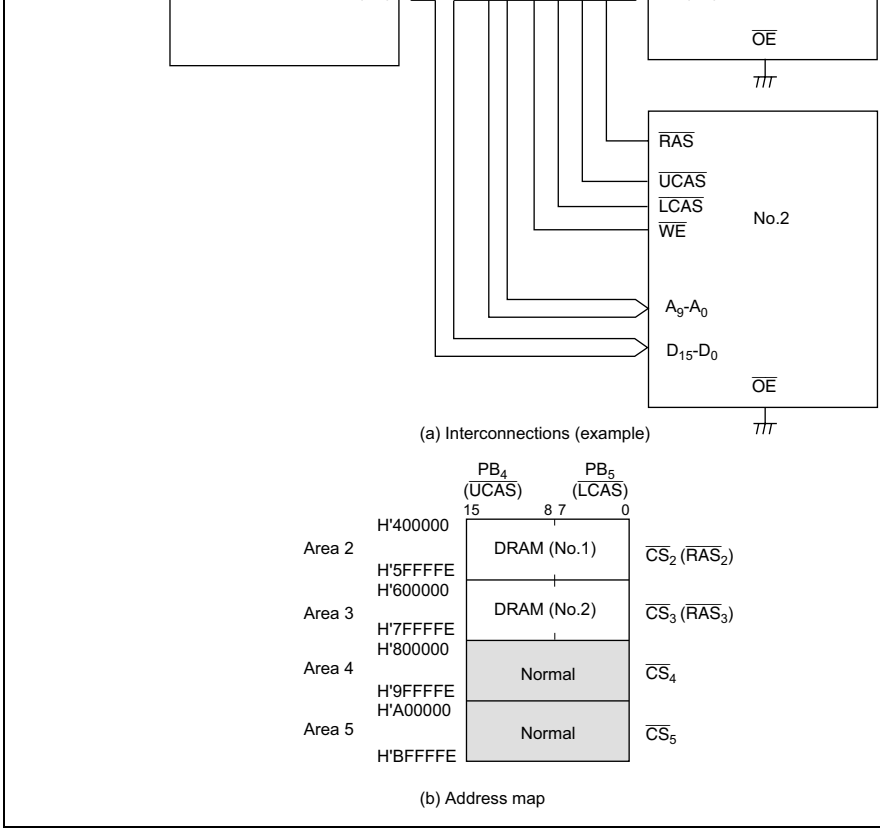


Figure 6.31 Interconnections and Address Map for 2-CAS 16-Mbit DRAMs wit Organization

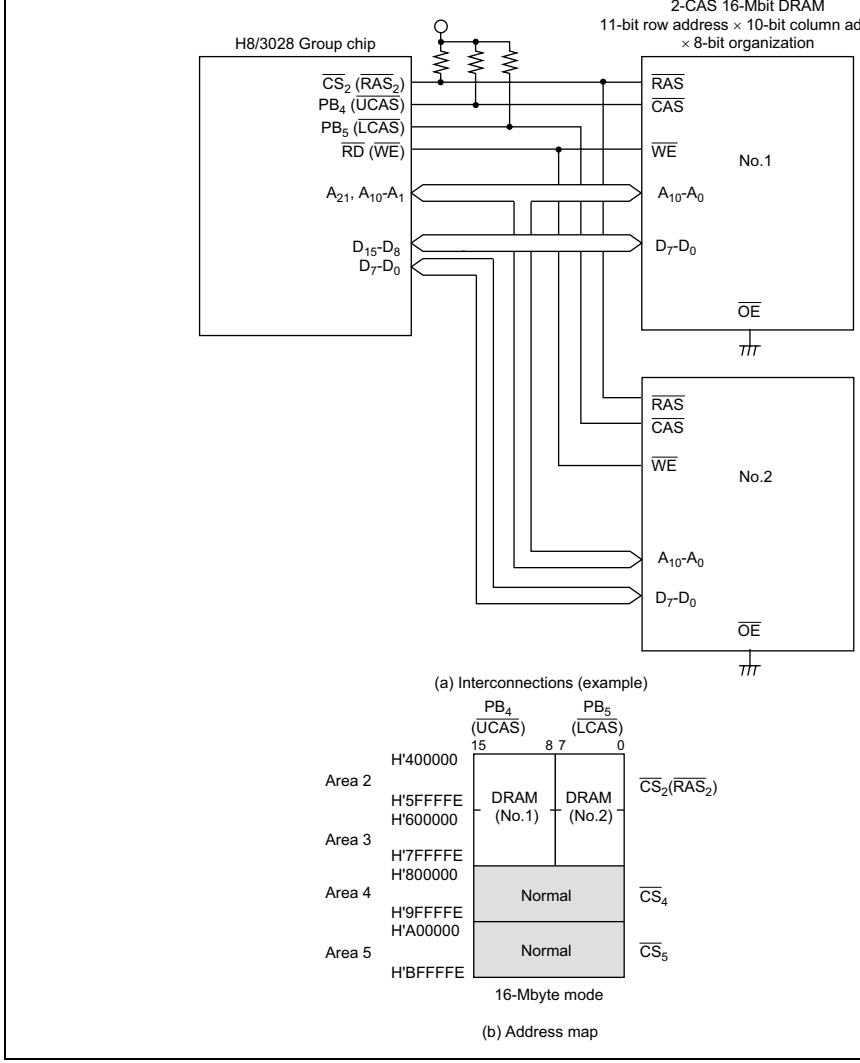
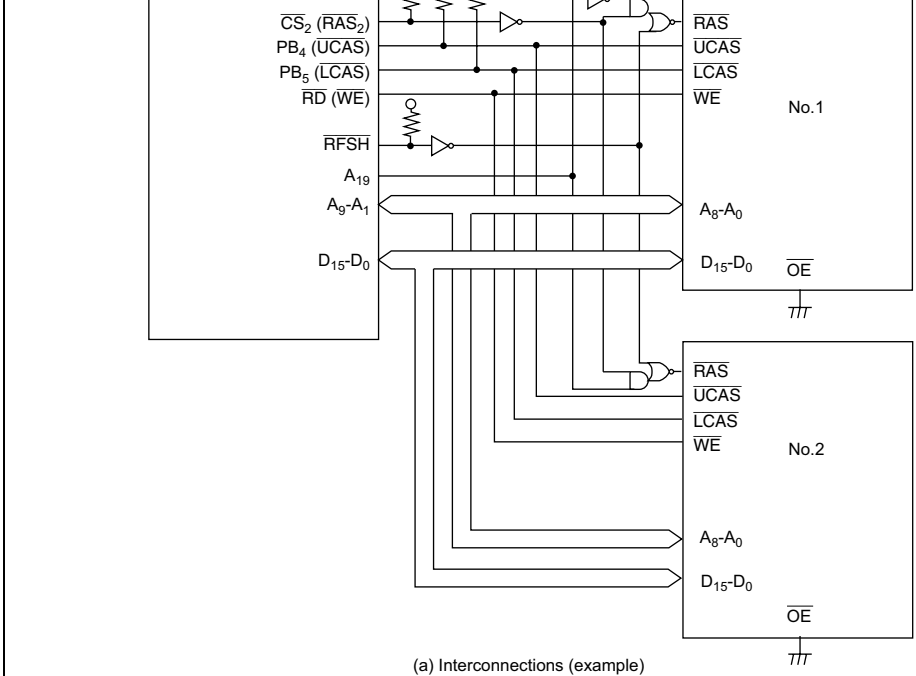
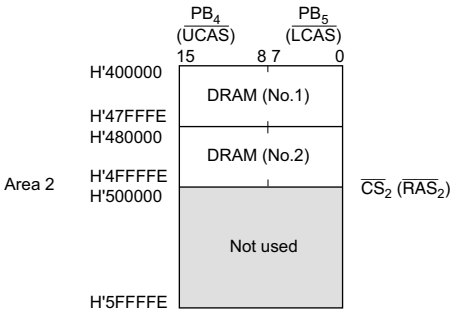


Figure 6.32 Interconnections and Address Map for 16-Mbit DRAMs with Organization



(a) Interconnections (example)



16-Mbyte mode

(b) Address map

Figure 6.33 Interconnections and Address Map for 2-CAS 4-Mbit DRAMs with Organization

RENESAS

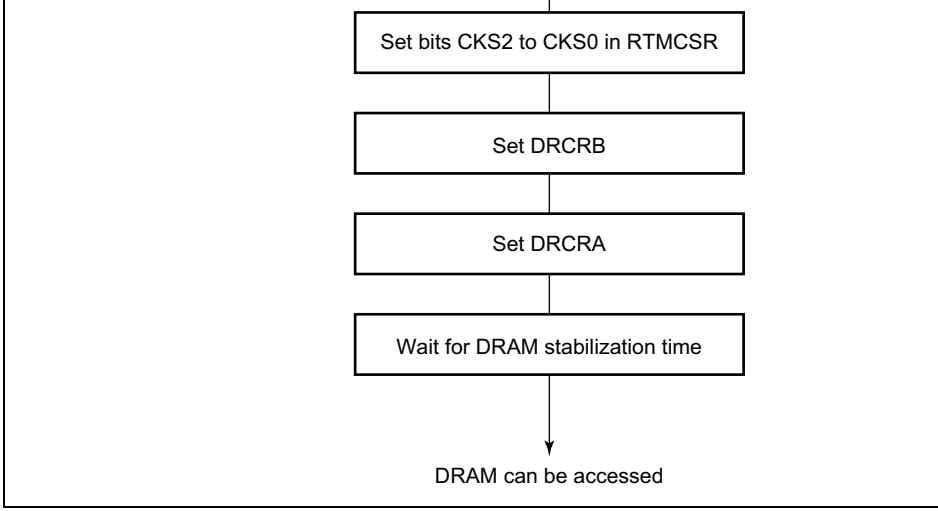


Figure 6.34 Example of Setup Procedure when Using DRAM Interfac

6.5.13 Usage Notes

Note the following points when using the DRAM refresh function.

- Refresh cycles will not be executed when the external bus released state, software mode, or a bus cycle is extended by means of wait state insertion. Refreshing mus be performed by other means in these cases.
- If a refresh request is generated internally while the external bus is released, the fi retained and a single refresh cycle will be executed after the bus-released state is c Figure 6.35 shows the bus cycle in this case.
- When a bus cycle is extended by means of wait state insertion, the first request is r the same way as when the external bus has been released.
- In the event of contention with a bus request from an external bus master when a t made to software standby mode, the $\overline{\text{BACK}}$ and strobe states may be indeterminat transition to software standby mode (see figure 6.36).

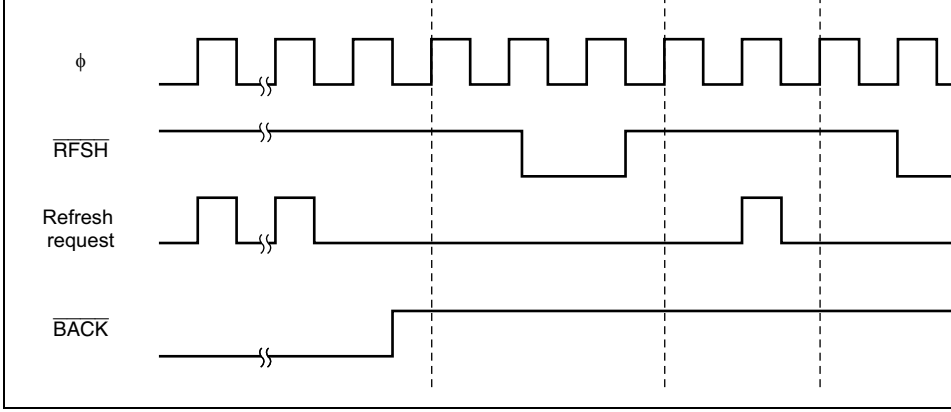


Figure 6.35 Bus-Released State and Refresh Cycles

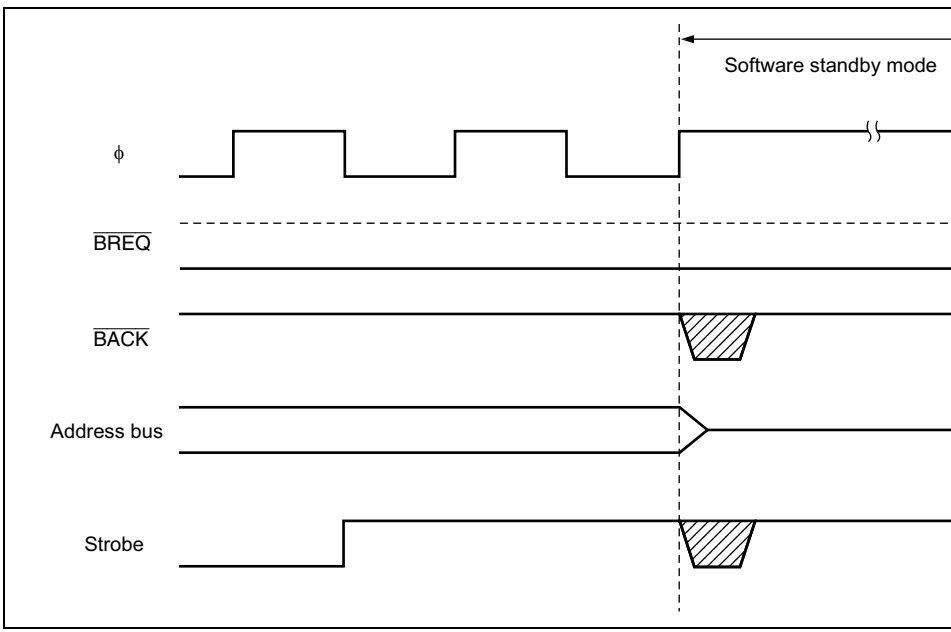


Figure 6.36 Bus-Released State and Software Standby Mode

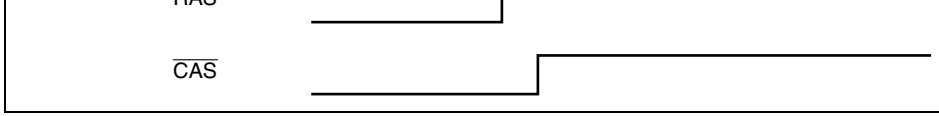


Figure 6.37 Self-Refresh Clearing

in RTMCSR is set to 1 by a compare match output when the RTCOR and RTCNT valu
The compare match signal is generated in the last state in which the values match (whe
is updated from the matching value to a new value). Accordingly, when RTCNT and R
match, the compare match signal is not generated until the next counter clock pulse. Fi
shows the timing.

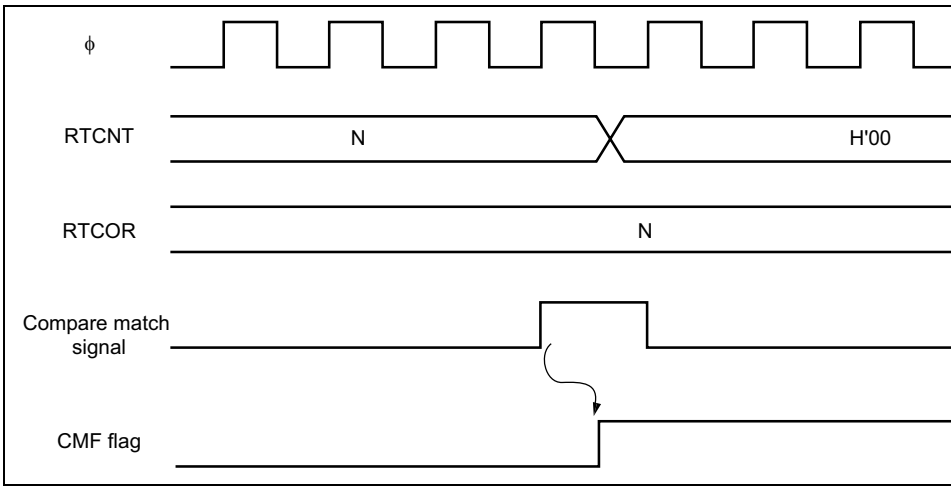


Figure 6.38 Timing of CMF Flag Setting

Operation in Power-Down State: The interval timer operates in sleep mode. It does n
in hardware standby mode. In software standby mode, RTCNT and RTMCSR bits 7 an
initialized, but RTMCSR bits 5 to 3 and RTCOR retain their settings prior to the transit
software standby mode.

Contention between RTCNT Write and Counter Clear: If a counter clear signal occ
T_3 state of an RTCNT write cycle, clearing of the counter takes priority and the write is
performed. See figure 6.39.

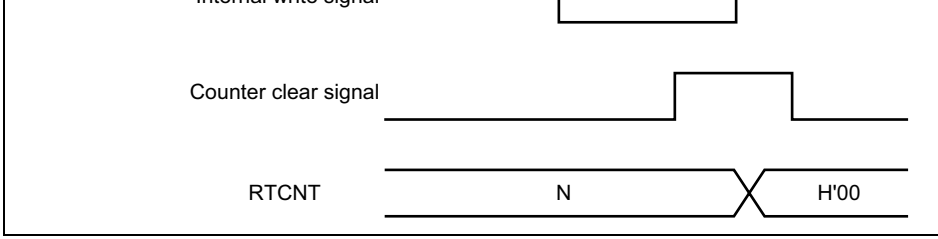


Figure 6.39 Contention between RTCNT Write and Clear

Contention between RTCNT Write and Increment: If an increment pulse occurs in of an RTCNT write cycle, writing takes priority and RTCNT is not incremented. See

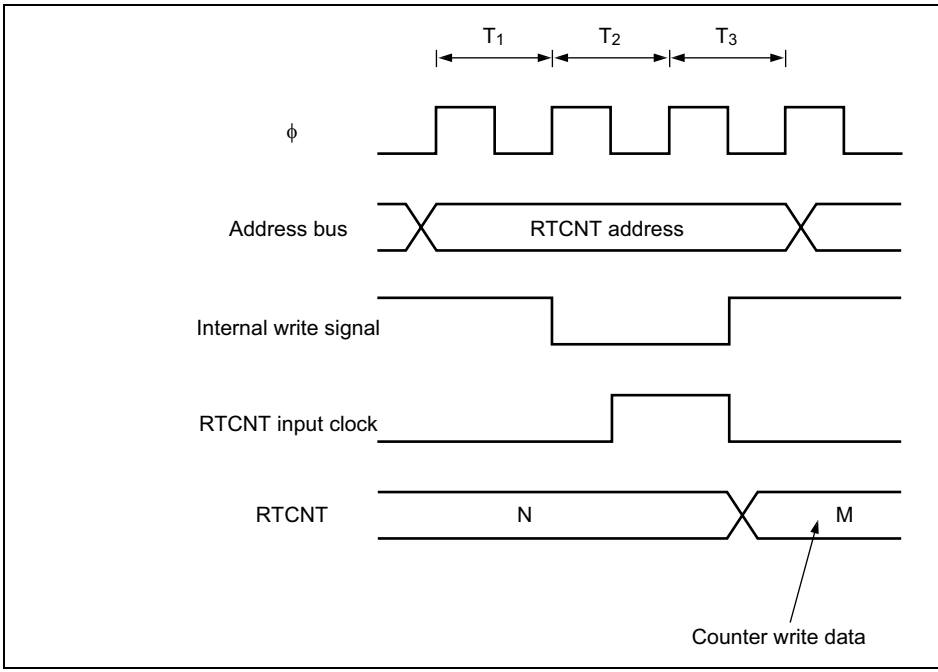


Figure 6.40 Contention between RTCNT Write and Increment

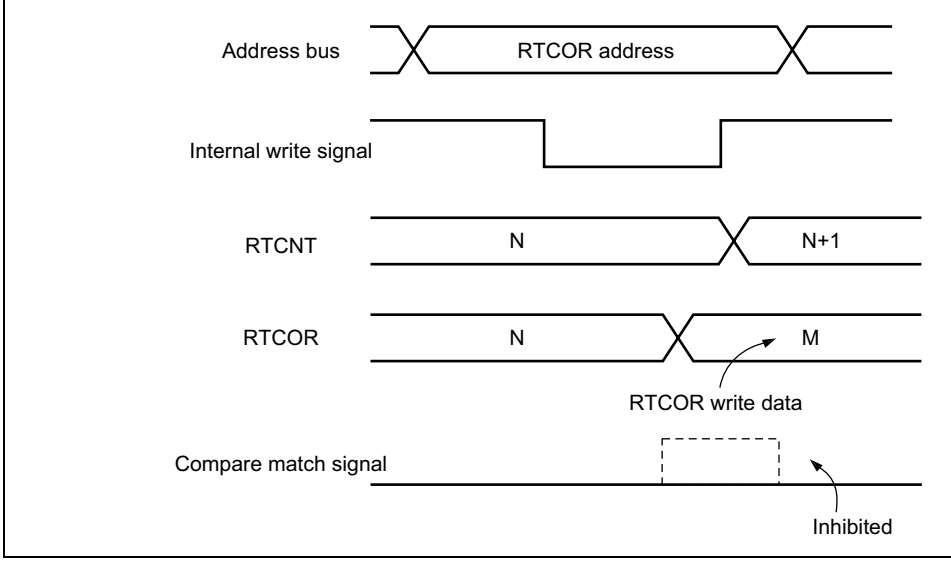
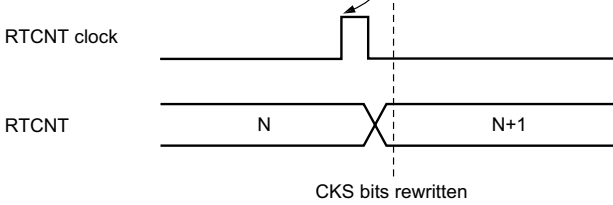
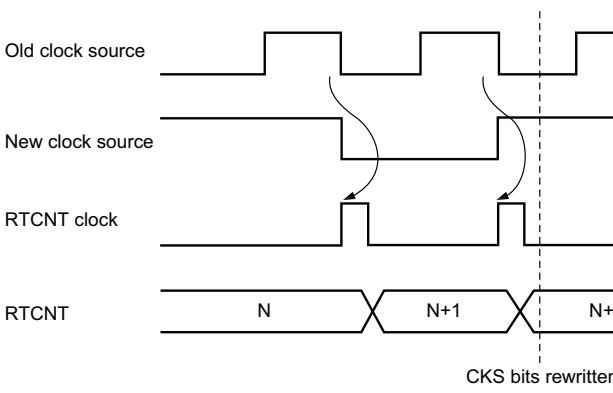
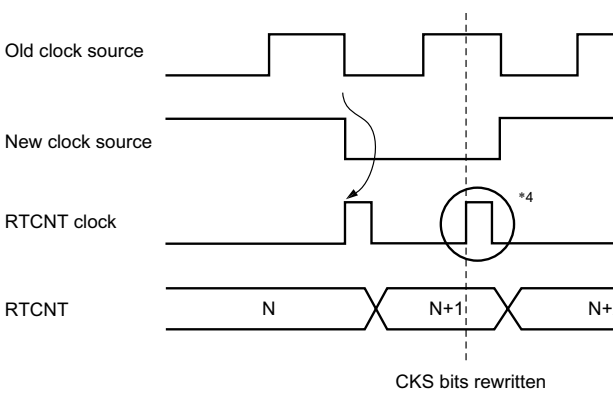
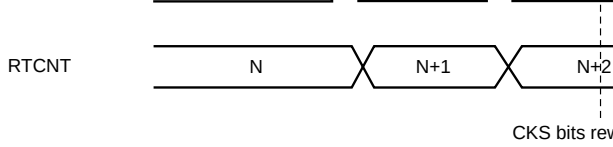


Figure 6.41 Contention between RTCOR Write and Compare Match

RTCNT Operation at Internal Clock Source Switchover: Switching internal clock s
cause RTCNT to increment, depending on the switchover timing. Table 6.10 shows the
between the time of the switchover (by writing to bits CKS2 to CKS0) and the operatio
RTCNT.

The RTCNT input clock is generated from the internal clock source by detecting the fa
of the internal clock. If a switchover is made from a high clock source to a low clock s
case No. 3 in table 6.10, the switchover will be regarded as a falling edge, an RTCNT c
will be generated, and RTCNT will be incremented.

		
2	Low → High switchover*2	
3	High → Low switchover*3	



Notes: 1. Including switchovers from a low clock source to the halted state, and from t
state to a low clock source.
2. Including switchover from the halted state to a high clock source.
3. Including switchover from a high clock source to the halted state.
4. The switchover is regarded as a falling edge, causing RTCNT to increment.

6.7 Interrupt Sources

Compare match interrupts (CMI) can be generated when the refresh timer is used as an
timer. Compare match interrupt requests are masked/unmasked with the CMIE bit in R

6.8 Burst ROM Interface

6.8.1 Overview

With the H8/3028 Group, external space area 0 can be designated as burst ROM space,
ROM space interfacing can be performed. The burst ROM space interface enables 16-
organization ROM with burst access capability to be accessed at high speed. Area 0 is
as burst ROM space by means of the BROME bit in BCR.

Continuous burst access of a maximum or four or eight words can be performed on exte
area 0. Two or three states can be selected for burst access.

The basic access timing for burst ROM space is shown in figure 6.42.

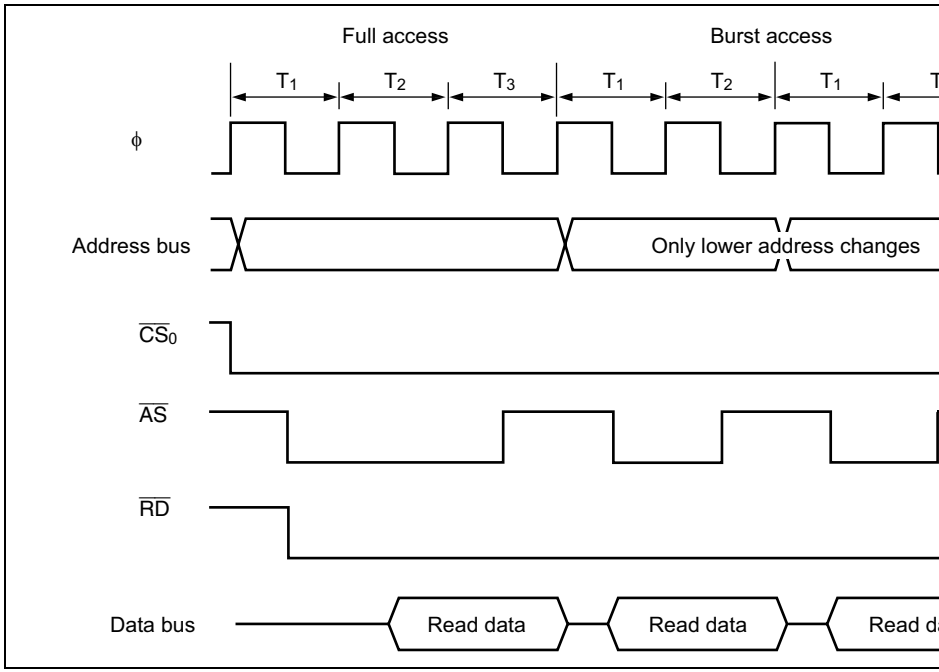


Figure 6.42 Example of Burst ROM Access Timing

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.

Wait states cannot be inserted in a burst cycle.

interfaces, and so on.

The ICIS1 and ICIS0 bits in BCR both have an initial value of 1, so that an idle cycle is the initial state. If there are no data collisions, the ICIS bits can be cleared.

Consecutive Reads between Different Areas: If consecutive reads between different a while the ICIS1 bit is set to 1 in BCR, an idle cycle is inserted at the start of the second

Figure 6.43 shows an example of the operation in this case. In this example, bus cycle 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 collisio cycle B between the read data from ROM and that from SRAM. In (b), an idle cycle is and a data collision is prevented.

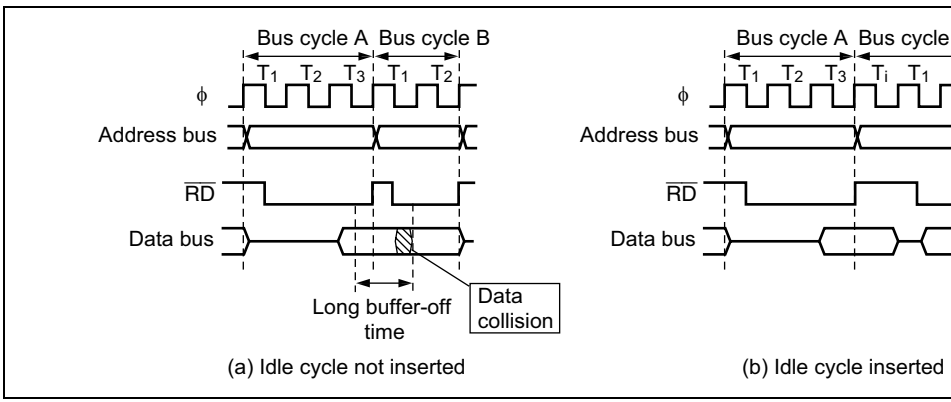


Figure 6.43 Example of Idle Cycle Operation (1) (ICIS1 = 1)

Write after Read: If an external write occurs after an external read while the ICIS0 bi in BCR, an idle cycle is inserted at the start of the write cycle.

Figure 6.44 shows an example of the operation in this case. In this example, bus cycle cycle from ROM with a long output floating time, and bus cycle B is a CPU write cycle

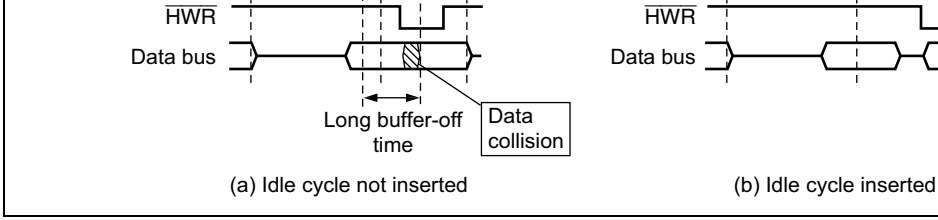


Figure 6.44 Example of Idle Cycle Operation (2) (ICIS0 = 1)

External Address Space Access Immediately after DRAM Space Access: If a DRA access is followed by a non-DRAM external access when $\overline{\text{HWR}}$ and $\overline{\text{LWR}}$ have been s the $\overline{\text{UCAS}}$ and $\overline{\text{LCAS}}$ output pins by means of the CSEL bit in DRCRB, a Ti cycle is i regardless of the settings of bits ICIS0 and ICIS1 in BCR. Figure 6.45 shows an exam operation.

This is done to prevent simultaneous changing of the $\overline{\text{HWR}}$ and $\overline{\text{LWR}}$ signals used as $\overline{\text{LCAS}}$ in DRAM space and $\overline{\text{CS}}$n for the space in the next cycle, and so avoid an erron the external device in the next cycle.

A T_i cycle is not inserted when PB4 and PB5 have been selected as the $\overline{\text{UCAS}}$ and $\overline{\text{LC}}$ pins.

In the case of consecutive DRAM space access precharge cycles (T_p), the ICIS0 and I settings are invalid. In the case of consecutive reads between different areas, for exam second access is a DRAM access, only a T_p cycle is inserted, and a T_i cycle is not. Th this case is shown in figure 6.46.

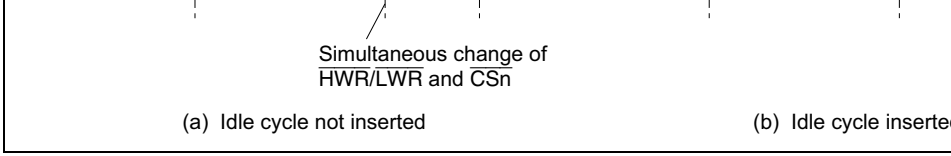


Figure 6.45 Example of Idle Cycle Operation (3) (HWR/LWR Used as UCAS/

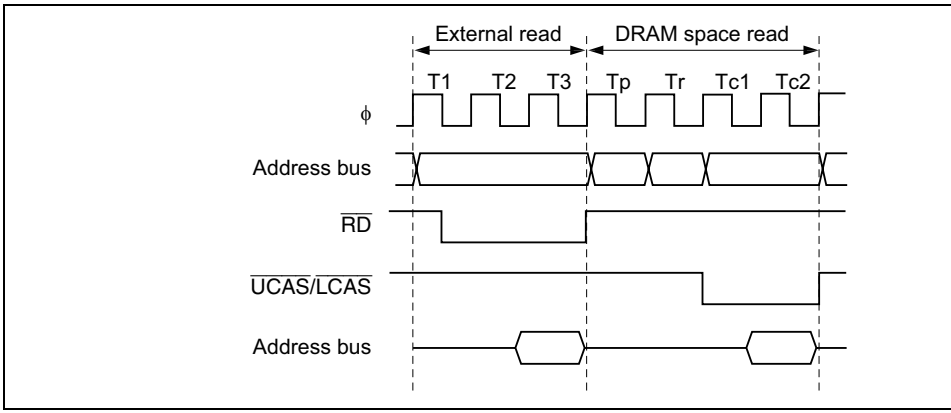


Figure 6.46 Example of Idle Cycle Operation (4) (Consecutive Precharge C

Usage Notes: When non-insertion of idle cycles is set, the rise (negation) of RD and th (assertion) of CSn may occur simultaneously. An example of the operation is shown in 6.47.

If consecutive reads between different external areas occur while the ICIS1 bit is cleare BCR, or if a write cycle to a different external area occurs after an external read while t bit is cleared to 0, the RD negation in the first read cycle and the CSn assertion in the fo bus cycle will occur simultaneously. Therefore, depending on the output delay time of it is possible that the low-level output of RD in the preceding read cycle and the low-le of CSn in the following bus cycle will overlap.

A setting whereby idle cycle insertion is not performed can be made only when RD and not change simultaneously, or when it does not matter if they do.

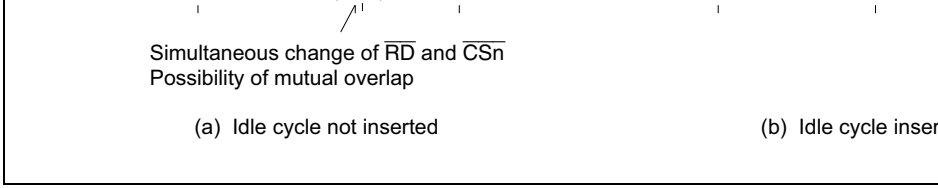


Figure 6.47 Example of Idle Cycle Operation (5)

6.9.2 Pin States in Idle Cycle

Table 6.11 shows the pin states in an idle cycle.

Table 6.11 Pin States in Idle Cycle

Pins	Pin State
A_{23} to A_0	Next cycle address value
D_{15} to D_0	High impedance
$\overline{CS}$n	High*
$\overline{UCAS}$, $\overline{LCAS}$	High
$\overline{AS}$	High
$\overline{RD}$	High
$\overline{HWR}$	High
$\overline{LWR}$	High

Note: * Remains low in DRAM space RAS down mode.

The bus arbiter checks whether the bus request signal from a bus master is active or ina
returns an acknowledge signal to the bus master. When two or more bus masters reque
the highest-priority bus master receives an acknowledge signal. The bus master that re
acknowledge signal can continue to use the bus until the acknowledge signal is deactiva

The bus master priority order is:

(High) External bus master > DRAM interface > DMAC > CPU (Low)

The bus arbiter samples the bus request signals and determines priority at all times, but
always grant the bus immediately, even when it receives a bus request from a bus maste
higher priority than the current bus master. Each bus master has certain times at which
release the bus to a higher-priority bus master.

6.10.1 Operation

CPU: The CPU is the lowest-priority bus master. If the DMAC, DRAM interface, or a
bus master requests the bus while the CPU has the bus right, the bus arbiter transfers th
to the bus master that requested it. The bus right is transferred at the following times:

- The bus right is transferred at the boundary of a bus cycle. If word data is accessed
consecutive byte accesses, however, the bus right is not transferred between the two
accesses.
- If another bus master requests the bus while the CPU is performing internal operatio
executing a multiply or divide instruction, the bus right is transferred immediately.
continues its internal operations.
- If another bus master requests the bus while the CPU is in sleep mode, the bus right
transferred immediately.

There is a priority order among the DMAC channels. For details see section 7.4.9, M
Channel Operation.

DRAM Interface: The DRAM interface requests the bus right from the bus arbiter w
cycle request is issued, and releases the bus at the end of the refresh cycle. For details
6.5, DRAM Interface.

External Bus Master: When the BRLE bit is set to 1 in BRCR, the bus can be releas
external bus master. The external bus master has highest priority, and requests the bu
the bus arbiter y driving the $\overline{\text{BREQ}}$ signal low. Once the external bus master acquires
keeps the bus until the $\overline{\text{BREQ}}$ signal goes high. While the bus is released to an extern
master, the H8/3028 Group chip holds the address bus, data bus, bus control signals ($\overline{\text{A}}$
$\overline{\text{HWR}}$, and $\overline{\text{LWR}}$), and chip select signals ($\overline{\text{CS}}$n: n = 7 to 0) in the high-impedance stat
the $\overline{\text{BACK}}$ pin in the low output state.

The bus arbiter samples the $\overline{\text{BREQ}}$ pin at the rise of the system clock (ϕ). If $\overline{\text{BREQ}}$ is
is released to the external bus master at the appropriate opportunity. The $\overline{\text{BREQ}}$ signa
held low until the $\overline{\text{BACK}}$ signal goes low.

When the $\overline{\text{BREQ}}$ pin is high in two consecutive samples, the $\overline{\text{BACK}}$ pin is driven high
bus-release cycle.

Figure 6.48 shows the timing when the bus right is requested by an external bus maste
read cycle in a two-state access area. There is a minimum interval of three states from
$\overline{\text{BREQ}}$ signal goes low until the bus is released.

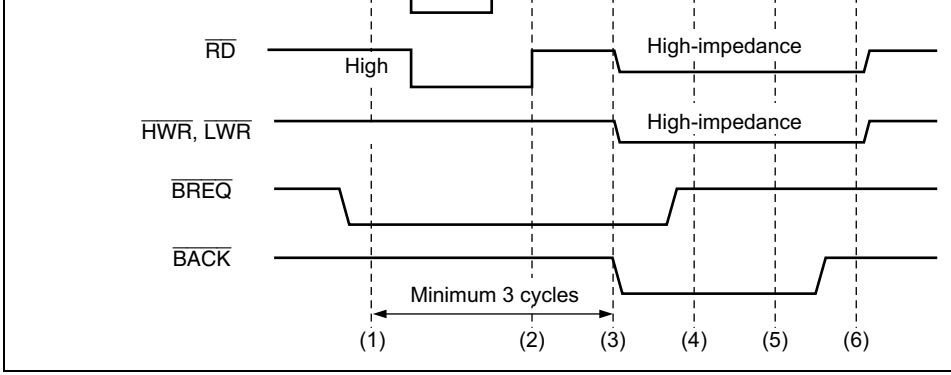


Figure 6.48 Example of External Bus Master Operation

In the event of contention with a bus request from an external bus master when a transi made to software standby mode, the $\overline{\text{BACK}}$ and strobe states may be indeterminate afte transition to software standby mode (see figure 6.36).

When software standby mode is used, the BRLE bit should be cleared to 0 in BRCR be executing the SLEEP instruction.

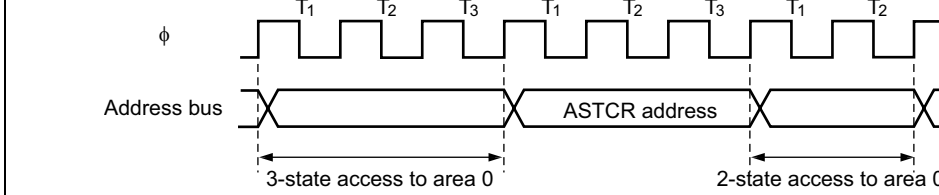


Figure 6.49 ASTCR Write Timing

DDR and CSCR Write Timing: Data written to DDR or CSCR for the port correspo $\overline{CS}$n pin to switch between $\overline{CS}$n output and generic input takes effect starting from the the DDR write cycle. Figure 6.50 shows the timing when the $\overline{CS}_1$ pin is changed from input to $\overline{CS}_1$ output.

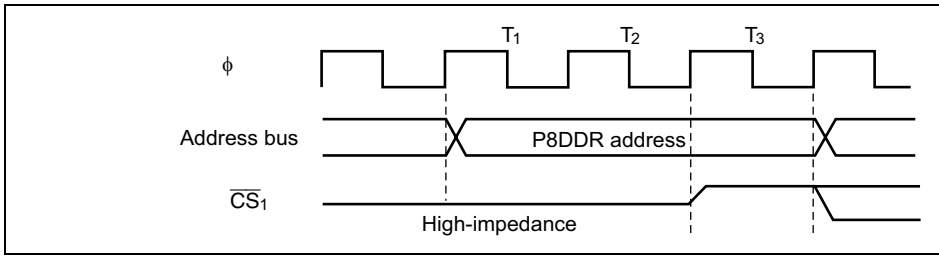


Figure 6.50 DDR Write Timing

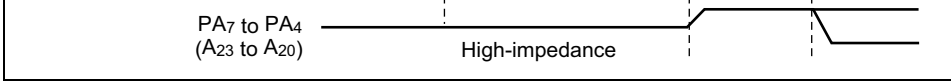


Figure 6.51 BRCR Write Timing

6.11.2 $\overline{\text{BREQ}}$ Pin Input Timing

After driving the $\overline{\text{BREQ}}$ pin low, hold it low until $\overline{\text{BACK}}$ goes low. If $\overline{\text{BREQ}}$ returns to level before $\overline{\text{BACK}}$ goes lows, the bus arbiter may operate incorrectly.

To terminate the external-bus-released state, hold the $\overline{\text{BREQ}}$ signal high for at least thre $\overline{\text{BREQ}}$ is high for too short an interval, the bus arbiter may operate incorrectly.

details see section 20.6, Module Standby Function.

7.1.1 Features

DMAC features are listed below.

- Selection of short address mode or full address mode
  Short address mode
  - 8-bit source address and 24-bit destination address, or vice versa
  - Maximum four channels available
  - Selection of I/O mode, idle mode, or repeat mode
- Full address mode
  - 24-bit source and destination addresses
  - Maximum two channels available
  - Selection of normal mode or block transfer mode
- Directly addressable 16-Mbyte address space
- Selection of byte or word transfer
- Activation by internal interrupts, external requests, or auto-request (depending on mode)
  - 16-bit timer compare match/input capture interrupts (×3)
  - Serial communication interface (SCI channel 0) transmit-data-empty/receive-d interrupts
  - External requests
  - Auto-request
  - A/D converter conversion-end interrupt

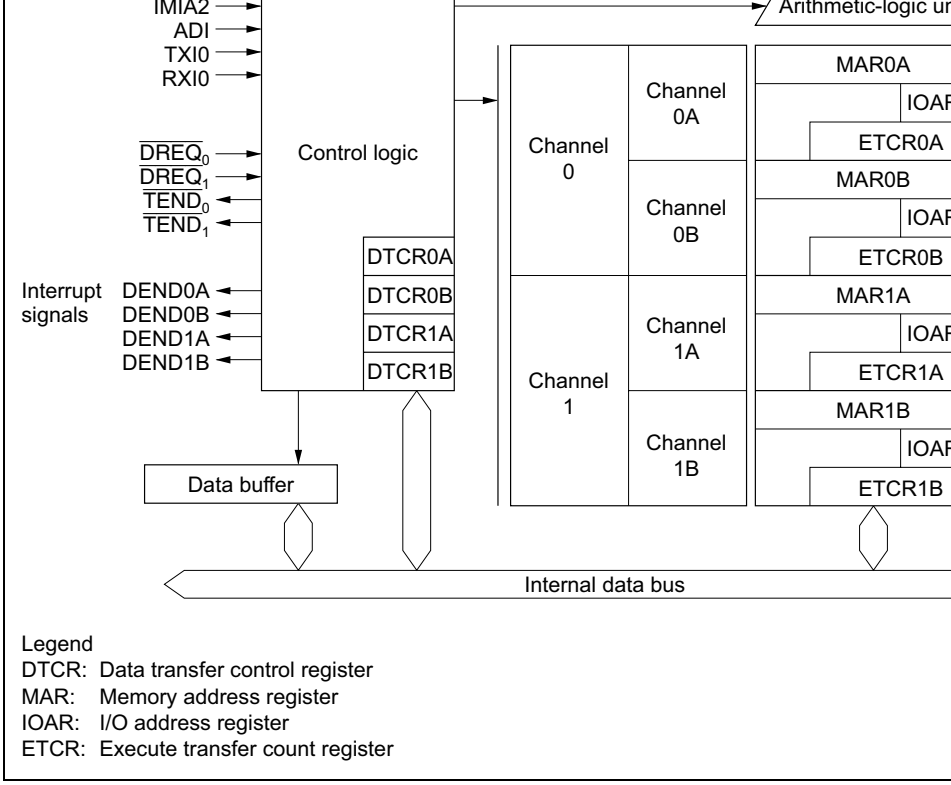


Figure 7.1 Block Diagram of DMAC

Short address mode	I/O mode • Transfers one byte or one word per request • Increments or decrements the memory address by 1 or 2 • Executes 1 to 65,536 transfers Idle mode • Transfers one byte or one word per request • Holds the memory address fixed • Executes 1 to 65,536 transfers Repeat mode • Transfers one byte or one word per request • Increments or decrements the memory address by 1 or 2 • Executes a specified number (1 to 255) of transfers, then returns to the initial state and continues	• Compare match/input capture A interrupts from 16-bit timer channels 0 to 2 • Transmit-data-empty interrupt from SCI channel 0	24
		• Conversion-end interrupt from A/D converter • Receive-data-full interrupt from SCI channel 0	8
		• External request	24
Full address mode	Normal mode • Auto-request — Retains the transfer request internally — Executes a specified number(1 to 65,536) of transfers continuously — Selection of burst mode or cycle-steal mode • External request — Transfers one byte or one word per request — Executes 1 to 65,536 transfers	• Auto-request • External request	24
	Block transfer • Transfers one block of a specified size per request • Executes 1 to 65,536 transfers • Allows either the source or destination to be a fixed block area • Block size can be 1 to 255 bytes or words	• Compare match/ input capture A interrupts from 16-bit timer channels 0 to 2 • External request • Conversion-end interrupt from A/D converter	24

	Transfer end 0	$\overline{TEND}_0$	Output	Transfer end on DMAC cha
1	DMA request 1	$\overline{DREQ}_1$	Input	External request for DMAC
	Transfer end 1	$\overline{TEND}_1$	Output	Transfer end on DMAC cha

Note: External requests cannot be made to channel A in short address mode.

7.1.5 Register Configuration

Table 7.3 lists the DMAC registers.

	H'FFF24	Execute transfer count register 0AH	ETCR0AH	R/W	Un
	H'FFF25	Execute transfer count register 0AL	ETCR0AL	R/W	Un
	H'FFF27	Data transfer control register 0A	DTCR0A	R/W	H'(
	H'FFF28	Memory address register 0BR	MAR0BR	R/W	Un
	H'FFF29	Memory address register 0BE	MAR0BE	R/W	Un
	H'FFF2A	Memory address register 0BH	MAR0BH	R/W	Un
	H'FFF2B	Memory address register 0BL	MAR0BL	R/W	Un
	H'FFF2E	I/O address register 0B	IOAR0B	R/W	Un
	H'FFF2C	Execute transfer count register 0BH	ETCR0BH	R/W	Un
	H'FFF2D	Execute transfer count register 0BL	ETCR0BL	R/W	Un
	H'FFF2F	Data transfer control register 0B	DTCR0B	R/W	H'(
1	H'FFF30	Memory address register 1AR	MAR1AR	R/W	Un
	H'FFF31	Memory address register 1AE	MAR1AE	R/W	Un
	H'FFF32	Memory address register 1AH	MAR1AH	R/W	Un
	H'FFF33	Memory address register 1AL	MAR1AL	R/W	Un
	H'FFF36	I/O address register 1A	IOAR1A	R/W	Un
	H'FFF34	Execute transfer count register 1AH	ETCR1AH	R/W	Un
	H'FFF35	Execute transfer count register 1AL	ETCR1AL	R/W	Un
	H'FFF37	Data transfer control register 1A	DTCR1A	R/W	H'(
	H'FFF38	Memory address register 1BR	MAR1BR	R/W	Un
	H'FFF39	Memory address register 1BE	MAR1BE	R/W	Un
	H'FFF3A	Memory address register 1BH	MAR1BH	R/W	Un
	H'FFF3B	Memory address register 1BL	MAR1BL	R/W	Un
	H'FFF3E	I/O address register 1B	IOAR1B	R/W	Un
	H'FFF3C	Execute transfer count register 1BH	ETCR1BH	R/W	Un
	H'FFF3D	Execute transfer count register 1BL	ETCR1BL	R/W	Un
	H'FFF3F	Data transfer control register 1B	DTCR1B	R/W	H'(

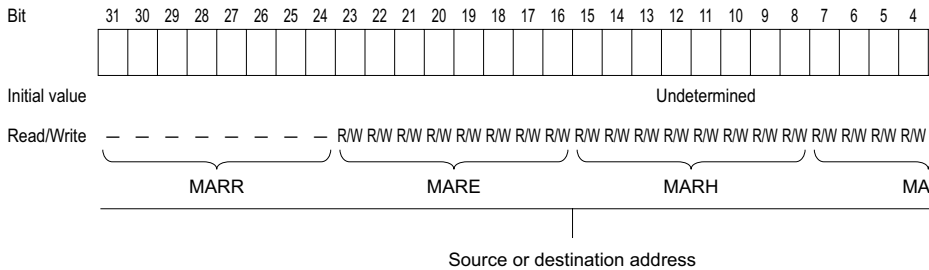
Note: * The lower 20 bits of the address are indicated.

0	1	1	DMAC channel 0 operates as one channel in full add
	Other than above		DMAC channels 0A and 0B operate as two independ channels in short address mode
1	1	1	DMAC channel 1 operates as one channel in full add
	Other than above		DMAC channels 1A and 1B operate as two independ channels in short address mode

7.2.1 Memory Address Registers (MAR)

A memory address register (MAR) is a 32-bit readable/writable register that specifies a destination address. The transfer direction is determined automatically from the activati

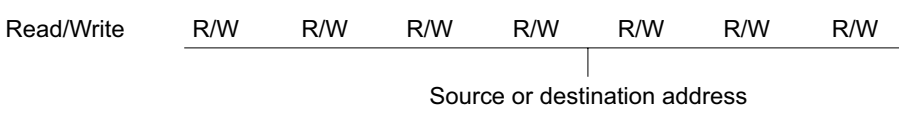
An MAR consists of four 8-bit registers designated MARR, MARE, MARH, and MAR of MARR are reserved; they cannot be modified and are always read as 1.



An MAR functions as a source or destination address register depending on how the DI activated: as a destination address register if activation is by a receive-data-full interrup serial communication interface (SCI) channel 0 or by an A/D converter conversion-end and as a source address register otherwise.

The MAR value is incremented or decremented each time one byte or word is transferr automatically updating the source or destination memory address. For details, see sectio Data Transfer Control Registers (DTCR).

The MARs are not initialized by a reset or in standby mode.



An IOAR functions as a source or destination address register depending on how the I activated: as a destination address register if activation is by a receive-data-full interru serial communication interface (SCI) channel 0 or by an A/D converter conversion-en and as a source address register otherwise.

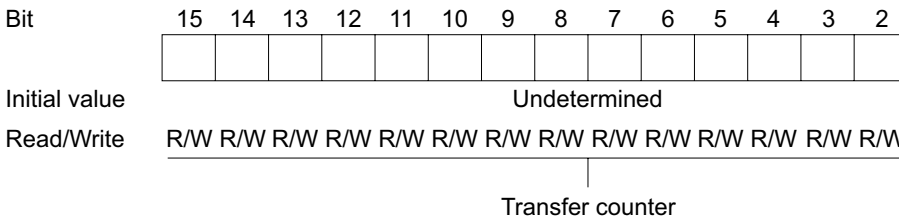
The IOAR value is held fixed. It is not incremented or decremented when a transfer is

The IOARs are not initialized by a reset or in standby mode.

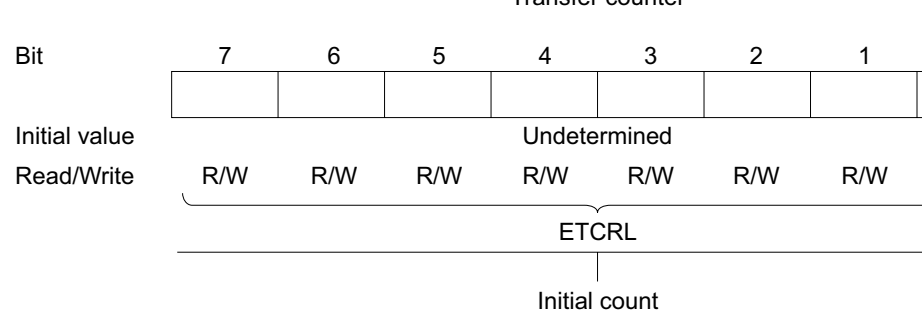
7.2.3 Execute Transfer Count Registers (ETCR)

An execute transfer count register (ETCR) is a 16-bit readable/writable register that sp number of transfers to be executed. These registers function in one way in I/O mode a mode, and another way in repeat mode.

- I/O mode and idle mode

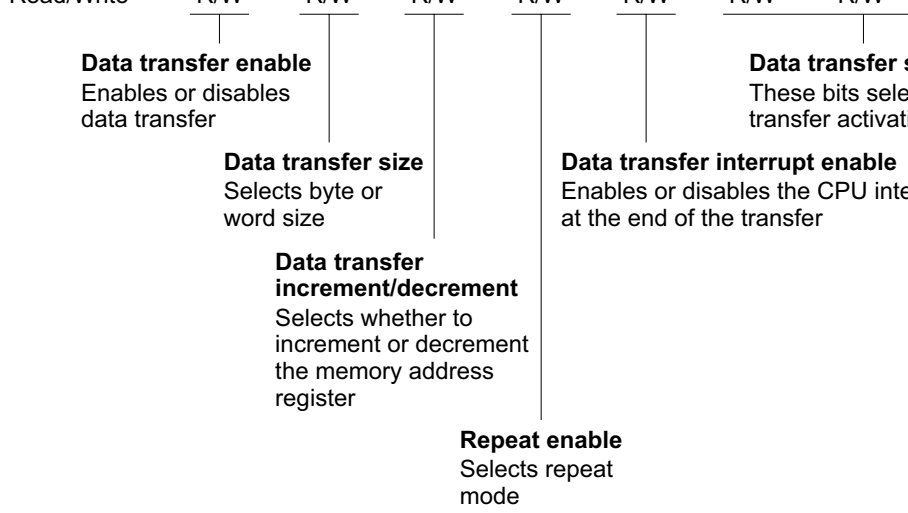


In I/O mode and idle mode, ETCR functions as a 16-bit counter. The count is decreme 1 each time one transfer is executed. The transfer ends when the count reaches H'0000



In repeat mode, ETCRH functions as an 8-bit transfer counter and ETCRL holds the ini transfer count. ETCRH is decremented by 1 each time one transfer is executed. When E reaches H'00, the value in ETCRL is reloaded into ETCRH and the same operation is re

The ETCRs are not initialized by a reset or in standby mode.



The DTCRs are initialized to H'00 by a reset and in standby mode.

Bit 7—Data Transfer Enable (DTE): Enables or disables data transfer on a channel. DTE bit is set to 1, the channel waits for a transfer to be requested, and executes the tr activated as specified by bits DTS2 to DTS0. When DTE is 0, the channel is disabled accept transfer requests. DTE is set to 1 by reading the register when DTE is 0, then w

Bit 7 DTE	Description	
0	Data transfer is disabled. In I/O mode or idle mode, DTE is cleared to 0 when the specified number of transfers have been completed	(I
1	Data transfer is enabled	

If DTIE is set to 1, a CPU interrupt is requested when DTIE is cleared to 0.

decrement the memory address register (MAR) after a data transfer in I/O mode or repe

Bit 5 DTID	Description	
0	MAR is incremented after each data transfer • If DTSZ = 0, MAR is incremented by 1 after each transfer • If DTSZ = 1, MAR is incremented by 2 after each transfer	(In
1	MAR is decremented after each data transfer • If DTSZ = 0, MAR is decremented by 1 after each transfer • If DTSZ = 1, MAR is decremented by 2 after each transfer	

MAR is not incremented or decremented in idle mode.

Bit 4—Repeat Enable (RPE): Selects whether to transfer data in I/O mode, idle mode mode.

Bit 4 RPE	Bit 3 DTIE	Description	
0	0	I/O mode	(In
	1		
1	0	Repeat mode	
	1	Idle mode	

Operations in these modes are described in sections 7.4.2, I/O Mode, 7.4.3, Idle Mode, Repeat Mode.

Bits 2 to 0 – Data Transfer Select (DTS2, DTS1, DTS0): These bits select the data t
activation source. Some of the selectable sources differ between channels A and B.

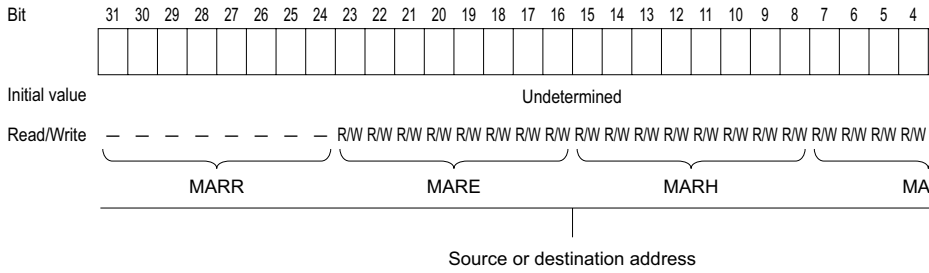
Bit 2 DTS2	Bit 1 DTS1	Bit 0 DTS0	Description
0	0	0	Compare match/input capture A interrupt from 16-bit timer (I
		1	Compare match/input capture A interrupt from 16-bit timer
	1	0	Compare match/input capture A interrupt from 16-bit timer
		1	Conversion-end interrupt from A/D converter
1	0	0	Transmit-data-empty interrupt from SCI channel 0
		1	Receive-data-full interrupt from SCI channel 0
	1	0	Falling edge of $\overline{\text{DREQ}}$ input (channel B) Transfer in full address mode (channel A)
		1	Low level of $\overline{\text{DREQ}}$ input (channel B) Transfer in full address mode (channel A)

Note: See section 7.3.4, Data Transfer Control Registers (DTCR).

The same internal interrupt can be selected as an activation source for two or more cha
once. In that case the channels are activated in a priority order, highest-priority channe
the priority order, see section 7.4.9, Multiple-Channel Operation.

When a channel is enabled (DTE = 1), its selected DMAC activation source cannot ge
CPU interrupt.

An MAR consists of four 8-bit registers designated MARR, MARE, MARH, and MAR
of MARR are reserved; they cannot be modified and are always read as 1. (Write is inv

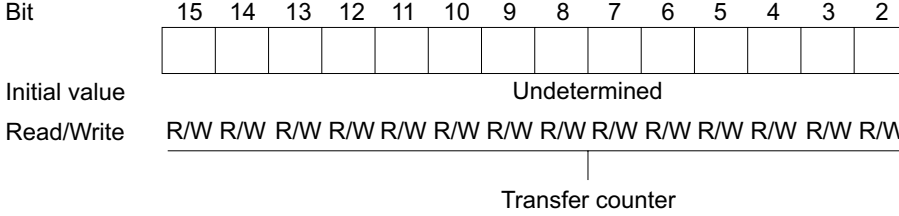


The MAR value is incremented or decremented each time one byte or word is transferr
automatically updating the source or destination memory address. For details, see sectio
Data Transfer Control Registers (DTCR).

The MARs are not initialized by a reset or in standby mode.

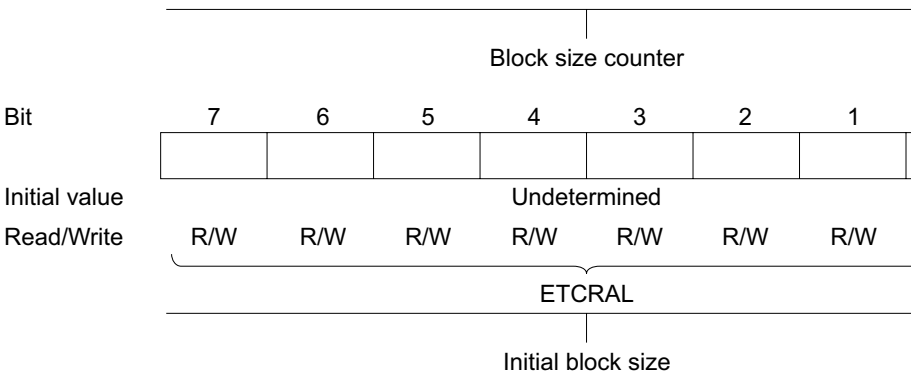
7.3.2 I/O Address Registers (IOAR)

The I/O address registers (IOARs) are not used in full address mode.

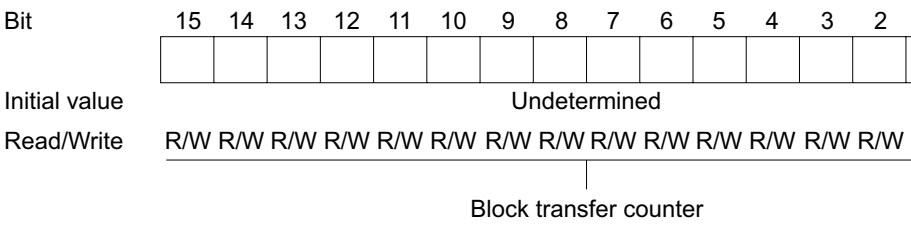


ETCRB: Is not used in normal mode.

In normal mode ETCRA functions as a 16-bit transfer counter. The count is decremen
each time one transfer is executed. The transfer ends when the count reaches H'0000.
not used.



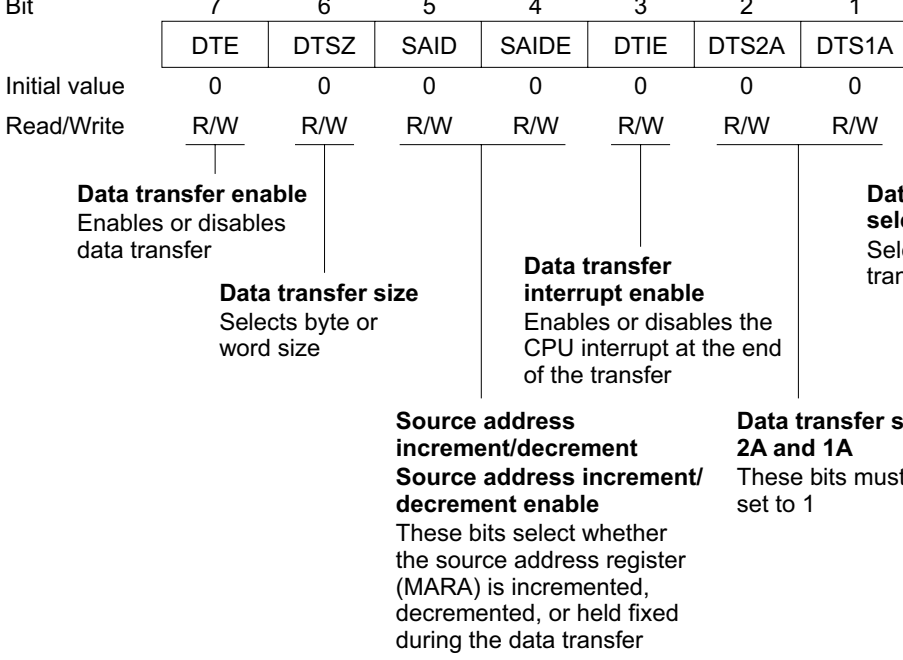
- ETCRB



In block transfer mode, ETCRAH functions as an 8-bit block size counter. ETCRAL ho initial block size. ETCRAH is decremented by 1 each time one byte or word is transfer the count reaches H'00, ETCRAH is reloaded from ETCRAL. Blocks consisting of an a number of bytes or words can be transferred repeatedly by setting the same initial block in ETCRAH and ETCRAL.

In block transfer mode ETCRB functions as a 16-bit block transfer counter. ETCRB is decremented by 1 each time one block is transferred. The transfer ends when the count H'0000.

The ETCRs are not initialized by a reset or in standby mode.



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

DTE	Description
0	Data transfer is disabled (DTE is cleared to 0 when the specified number (In of transfers have been completed)
1	Data transfer is enabled

If DTIE is set to 1, a CPU interrupt is requested when DTE is cleared to 0.

Bit 6—Data Transfer Size (DTSZ): Selects the data size of each transfer.

Bit 6 DTSZ	Description
0	Byte-size transfer (In
1	Word-size transfer

Bit 5—Source Address Increment/Decrement (SAID) and,
Bit 4—Source Address Increment/Decrement Enable (SAIDE): These bits select wh source address register (MARA) is incremented, decremented, or held fixed during the transfer.

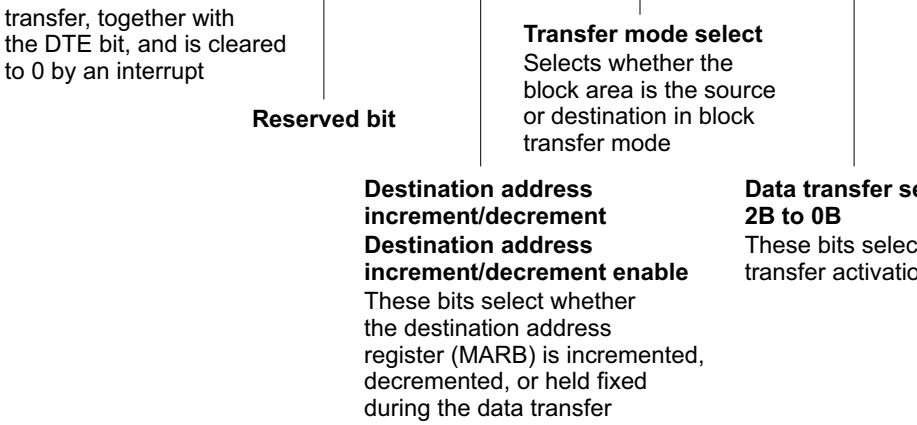
Bit 5 SAID	Bit 4 SAIDE	Description
0	0	MARA is held fixed (In
	1	MARA is incremented after each data transfer • If DTSZ = 0, MARA is incremented by 1 after each transfer • If DTSZ = 1, MARA is incremented by 2 after each transfer
1	0	MARA is held fixed
	1	MARA is decremented after each data transfer • If DTSZ = 0, MARA is decremented by 1 after each transfer • If DTSZ = 1, MARA is decremented by 2 after each transfer

Bits 2 and 1—Data Transfer Select 2A and 1A (DTS2A, DTS1A): A channel operat
address mode when DTS2A and DTS1A are both set to 1.

Bit 0—Data Transfer Select 0A (DTS0A): Selects normal mode or block transfer mo

Bit 0 DTS0A	Description
0	Normal mode (I
1	Block transfer mode

Operations in these modes are described in sections 7.4.5, Normal Mode, and 7.4.6, B
Transfer Mode.



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

Bit 7—Data Transfer Master Enable (DTME): Together with the DTE bit in DTCR enables or disables data transfer. When the DTME and DTE bits are both set to 1, the c enabled. When an NMI interrupt occurs DTME is cleared to 0, suspending the transfer CPU can use the bus. The suspended transfer resumes when DTME is set to 1 again. Fo information on operation in block transfer mode, see section 7.6.6, NMI Interrupts and Transfer Mode.

DTME is set to 1 by reading the register while DTME = 0, then writing 1.

Bit 7 DTME	Description
0	Data transfer is disabled (DTME is cleared to 0 when an NMI interrupt occurs) (In
1	Data transfer is enabled

0	0	MARB is held fixed
	1	MARB is incremented after each data transfer • If DTSZ = 0, MARB is incremented by 1 after each data trans • If DTSZ = 1, MARB is incremented by 2 after each data trans
1	0	MARB is held fixed
	1	MARB is decremented after each data transfer • If DTSZ = 0, MARB is decremented by 1 after each data tran • If DTSZ = 1, MARB is decremented by 2 after each data tran

Bit 3—Transfer Mode Select (TMS): Selects whether the source or destination is the in block transfer mode.

Bit 3 TMS	Description
0	Destination is the block area in block transfer mode
1	Source is the block area in block transfer mode

		1	Cannot be used
	1	0	Auto-request (cycle-steal mode)
		1	Cannot be used
1	0	0	Cannot be used
		1	Cannot be used
	1	0	Falling edge of $\overline{\text{DREQ}}$
		1	Low level input at $\overline{\text{DREQ}}$

- Block transfer mode

Bit 2 DTS2B	Bit 1 DTS1B	Bit 0 DTS0B	Description
0	0	0	Compare match/input capture A interrupt from 16-bit timer cha (In
		1	Compare match/input capture A interrupt from 16-bit timer cha
	1	0	Compare match/input capture A interrupt from 16-bit timer cha
		1	Conversion-end interrupt from A/D converter
1	0	0	Cannot be used
		1	Cannot be used
	1	0	Falling edge of $\overline{\text{DREQ}}$
		1	Cannot be used

The same internal interrupt can be selected to activate two or more channels. The chann activated in a priority order, highest priority first. For the priority order, see section 7.4 Multiple-Channel Operation.

Short address mode	I/O mode Idle mode Repeat mode	Compare match/input capture A interrupt from 16-bit timer channels 0 to 2	• Up to four cha can operate in • Only the B cha support extern
		Transmit-data-empty and receive-data-full interrupts from SCI channel 0	
		Conversion-end interrupt from A/D converter	
		External request	
Full address mode	Normal mode	Auto-request	• A and B chann paired; up to tw are available • Burst mode tra cycle-steal mo can be selecte requests
		External request	
	Block transfer mode	Compare match/input capture A interrupt from 16-bit timer channels 0 to 2	
		Conversion-end interrupt from A/D converter	
		External request	

A summary of operations in these modes follows.

I/O Mode: One byte or word is transferred per request. A designated number of these executed. A CPU interrupt can be requested at completion of the designated number o One 24-bit address and one 8-bit address are specified. The transfer direction is deterr automatically from the activation source.

Idle Mode: One byte or word is transferred per request. A designated number of these are executed. A CPU interrupt can be requested at completion of the designated numb transfers. One 24-bit address and one 8-bit address are specified. The addresses are he transfer direction is determined automatically from the activation source.

The DMAC is activated by register setup alone, and continues executing transfers u
designated number of transfers have been completed. A CPU interrupt can be reque
completion of the transfers. Both addresses are 24-bit addresses.

— Cycle-steal mode

The bus is released to another bus master after each byte or word is transferred.

— Burst mode

Unless requested by a higher-priority bus master, the bus is not released until th
designated number of transfers have been completed.

- External request

  One byte or word is transferred per request. A designated number of these transfers
  executed. A CPU interrupt can be requested at completion of the designated numbe
  transfers. Both addresses are 24-bit addresses.

Block Transfer Mode: One block of a specified size is transferred per request. A desig
number of block transfers are executed. At the end of each block transfer, one address i
to its initial value. When the designated number of blocks have been transferred, a CPU
can be requested. Both addresses are 24-bit addresses.

in MAR to the address specified in IOAR otherwise.

Table 7.6 indicates the register functions in I/O mode.

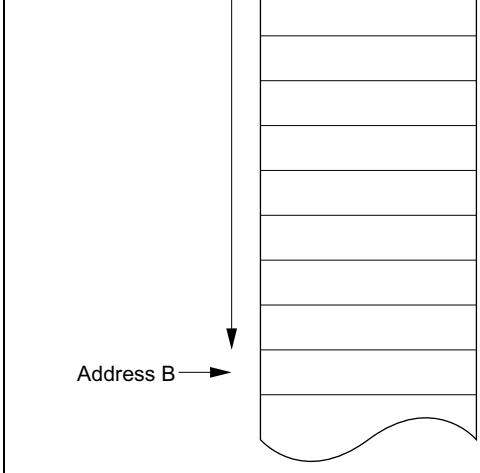
Table 7.6 Register Functions in I/O Mode

Register	Function: Activated by SCI 0 Receive-Data-Full Interrupt	Function: Other Activation	Initial Setting	Op
23 MAR 0	Destination address register	Source address register	Destination or source start address	Inc dec onc tra
23 All 1s 7 IOAR 0	Source address register	Destination address register	Source or destination address	Hel
15 ETCR 0	Transfer counter		Number of transfers	Dec onc tra H'0 rea tra

Legend
MAR: Memory address register
IOAR: I/O address register
ETCR: Execute transfer count register

MAR and IOAR specify the source and destination addresses. MAR specifies a 24-bit destination address, which is incremented or decremented as each byte or word is tran IOAR specifies the lower 8 bits of a fixed address. The upper 16 bits are all 1s. IOAR incremented or decremented.

Figure 7.2 illustrates how I/O mode operates.



Legend
L = initial setting of MAR
N = initial setting of ETCR
Address T = L
Address B = L + $(-1)^{DTID}$ • $(2^{DTSZ}$ • N – 1)

Figure 7.2 Operation in I/O Mode

The transfer count is specified as a 16-bit value in ETCR. The ETCR value is decremer each transfer. When the ETCR value reaches H'0000, the DTE bit is cleared and the tra If the DTIE bit is set to 1, a CPU interrupt is requested at this time. The maximum trans is 65,536, obtained by setting ETCR to H'0000.

Transfers can be requested (activated) by compare match/input capture A interrupts fro timer channels 0 to 2, transmit-data-empty and receive-data-full interrupts from SCI ch conversion-end interrupts from the A/D converter, and external request signals.

For the detailed settings see section 7.2.4, Data Transfer Control Registers (DTCR).

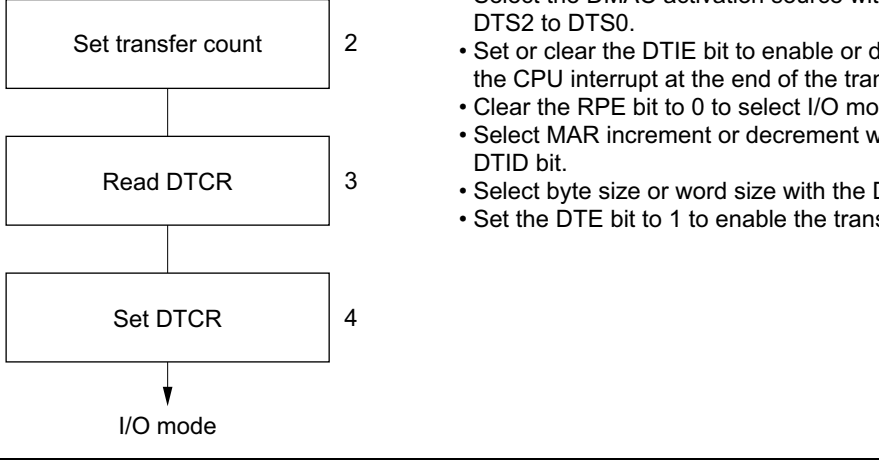


Figure 7.3 I/O Mode Setup Procedure (Example)

7.4.3 Idle Mode

Idle mode can be selected independently for each channel.

One byte or word is transferred at each transfer request in idle mode. A designated nu these transfers are executed. One address is specified in the memory address register (other in the I/O address register (IOAR). The direction of transfer is determined auton from the activation source. The transfer is from the address specified in IOAR to the a specified in MAR if activated by an SCI channel 0 receive-data-full interrupt, and fro address specified in MAR to the address specified in IOAR otherwise.

Table 7.7 indicates the register functions in idle mode.

23 7 0 All 1s | IOAR	Source address register	Destination address register	Source or destination address	Held
15 0 ETCR	Transfer counter		Number of transfers	Decr once trans H'00 reac trans

Legend

MAR: Memory address register

IOAR: I/O address register

ETCR: Execute transfer count register

MAR and IOAR specify the source and destination addresses. MAR specifies a 24-bit s
destination address. IOAR specifies the lower 8 bits of a fixed address. The upper 16 bi
1s. MAR and IOAR are not incremented or decremented.

Figure 7.4 illustrates how idle mode operates.

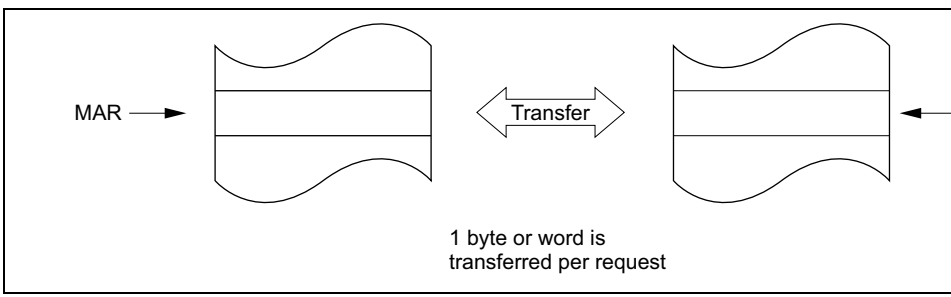


Figure 7.4 Operation in Idle Mode

For the detailed settings see section 7.3.4, Data Transfer Control Registers (DTCR).

Figure 7.5 shows a sample setup procedure for idle mode.

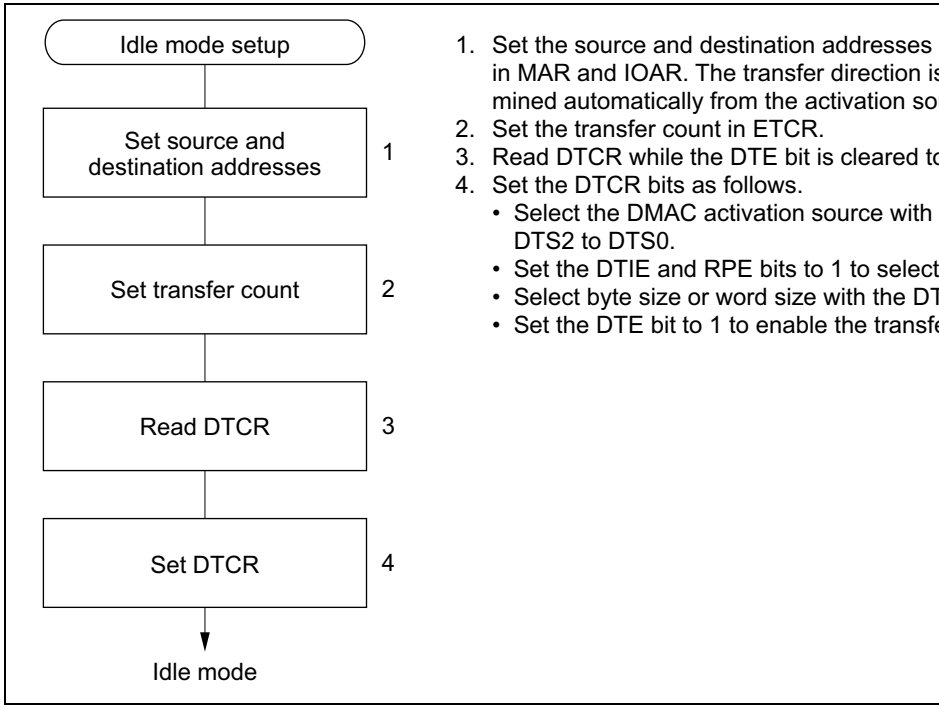
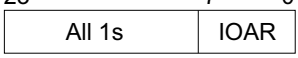
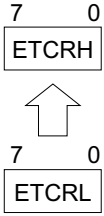


Figure 7.5 Idle Mode Setup Procedure (Example)

MAR and ETCRH are restored to their original values and operation continues. The dir
transfer is determined automatically from the activation source. The transfer is from the
specified in IOAR to the address specified in MAR if activated by an SCI channel 0 rec
full interrupt, and from the address specified in MAR to the address specified in IOAR

Table 7.8 indicates the register functions in repeat mode.

Table 7.8 Register Functions in Repeat Mode

	Function			
Register	**Activated by SCI 0 Receive-Data-Full Interrupt**	**Other Activation**	**Initial Setting**	**Operation**
	Destination address register	Source address register	Destination or source start address	Incrementd decrement each trans ETCRH re H'0000, th to initial va
	Source address register	Destination address register	Source or destination address	Held fixed
	Transfer counter		Number of transfers	Decremen per transfe H'0000 is then reloa ETCRL
	Initial transfer count		Number of transfers	Held fixed

Legend
MAR: Memory address register
IOAR: I/O address register
ETCR: Execute transfer count register

if the CPU sets DTE to 1 again, transfers resume from the state at which DTE was cle
CPU interrupt is requested.

As in I/O mode, MAR and IOAR specify the source and destination addresses. MAR s
24-bit source or destination address. IOAR specifies the lower 8 bits of a fixed addres
16 bits are all 1s. IOAR is not incremented or decremented.

Figure 7.6 illustrates how repeat mode operates.

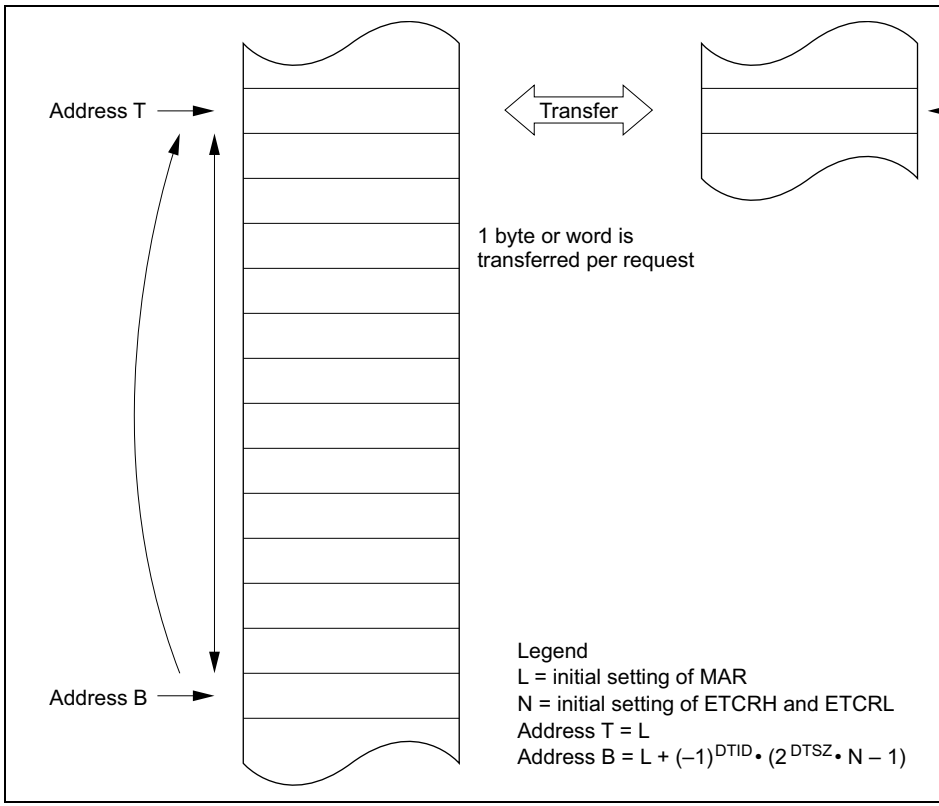


Figure 7.6 Operation in Repeat Mode

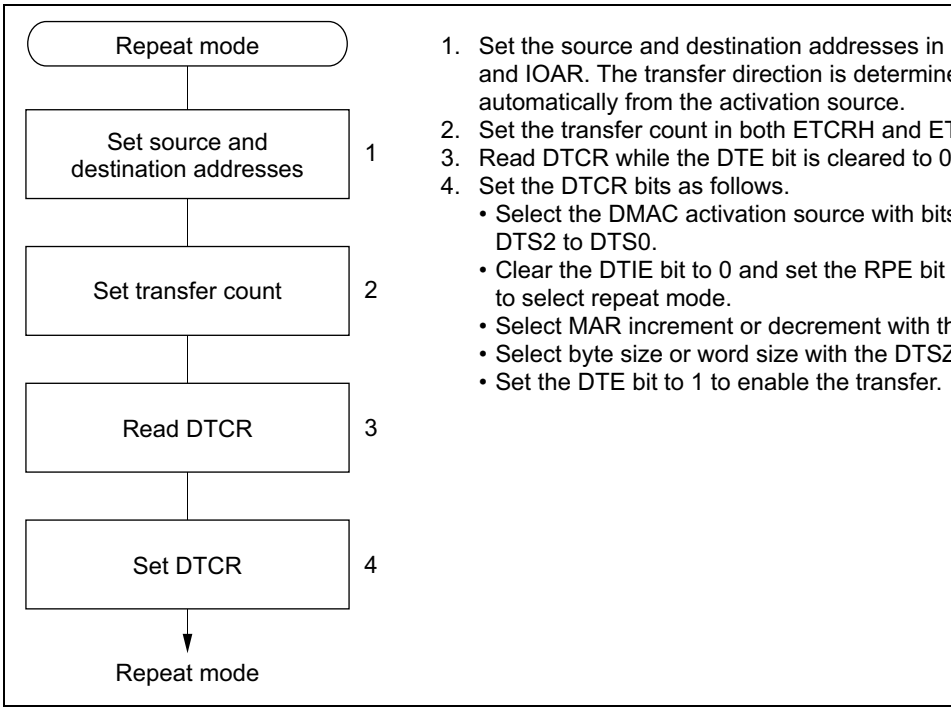


Figure 7.7 Repeat Mode Setup Procedure (Example)

23 0 MARA	Source address register	Source start address	Incremented decremented transfer, or h
23 0 MARB	Destination address register	Destination start address	Incremented decremented transfer, or h
15 0 ETCRA	Transfer counter	Number of transfers	Decremented transfer

Legend

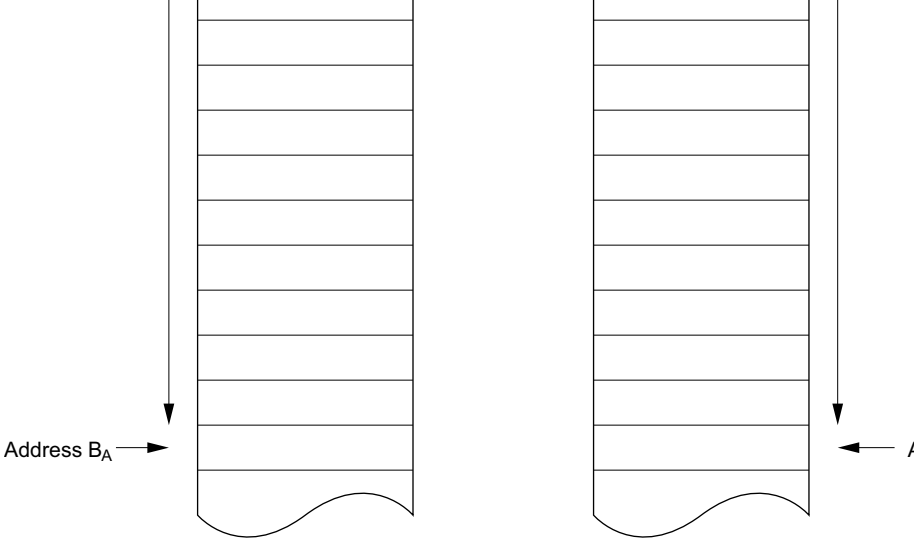
MARA: Memory address register A

MARB: Memory address register B

ETCRA: Execute transfer count register A

The source and destination addresses are both 24-bit addresses. MARA specifies the s address. MARB specifies the destination address. MARA and MARB can be indepen incremented, decremented, or held fixed as data is transferred.

The transfer count is specified as a 16-bit value in ETCRA. The ETCRA value is decr 1 at each transfer. When the ETCRA value reaches H'0000, the DTE bit is cleared and ends. If the DTIE bit is set to 1, a CPU interrupt is requested at this time. The maximu count is 65,536, obtained by setting ETCRA to H'0000.

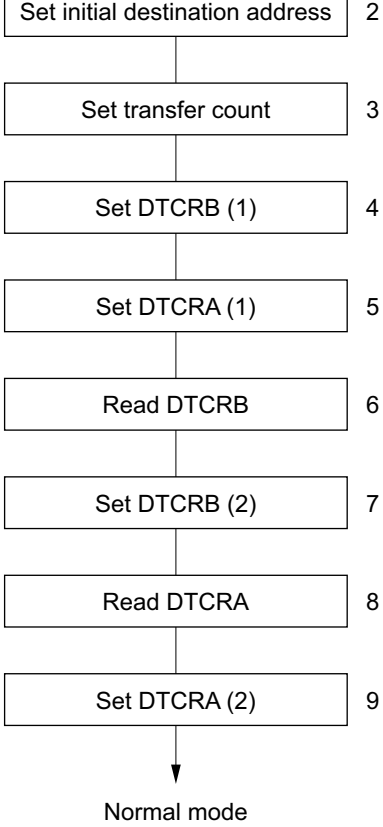


Legend
L_A = initial setting of MARA
L_B = initial setting of MARB
N = initial setting of ETCRA
$T_A = L_A$
$B_A = L_A + SAIDE \cdot (-1)^{SAID} \cdot (2^{DTSZ} \cdot N - 1)$
$T_B = L_B$
$B_B = L_B + DAIDE \cdot (-1)^{DAID} \cdot (2^{DTSZ} \cdot N - 1)$

Figure 7.8 Operation in Normal Mode

Transfers can be requested (activated) by an external request or auto-request. An auto-r transfer is activated by the register settings alone. The designated number of transfers a automatically. Either cycle-steal or burst mode can be selected. In cycle-steal mode the releases the bus temporarily after each transfer. In burst mode the DMAC keeps the bus transfers are completed, unless there is a bus request from a higher-priority bus master.

For the detailed settings see section 7.3.4, Data Transfer Control Registers (DTCR).

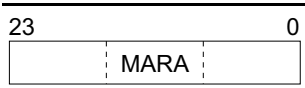
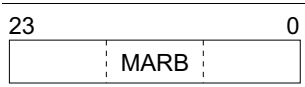
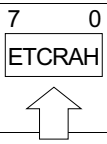
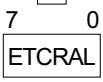
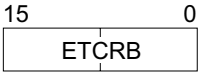


- Select the DMAC activation source wit DTS2B to DTS0B.

5. Set the DTCRA bits as follows.
   - Clear the DTE bit to 0.
   - Select byte or word size with the DTSZ
   - Set the SAID and SAIDE bits to select MARA is incremented, decremented, c
   - Set or clear the DTIE bit to enable or d CPU interrupt at the end of the transfe
   - Clear the DTS0A bit to 0 and set the D and DTS1A bits to 1 to select normal r
6. Read DTCRB with DTME cleared to 0.
7. Set the DTME bit to 1 in DTCRB.
8. Read DTCRA with DTE cleared to 0.
9. Set the DTE bit to 1 in DTCRA to enable

Note: Carry out settings 1 to 9 with the DEND interrupt masked in the CPU.
If an NMI interrupt occurs during the setup procedure, it may clear the DTME which case the transfer will not start.

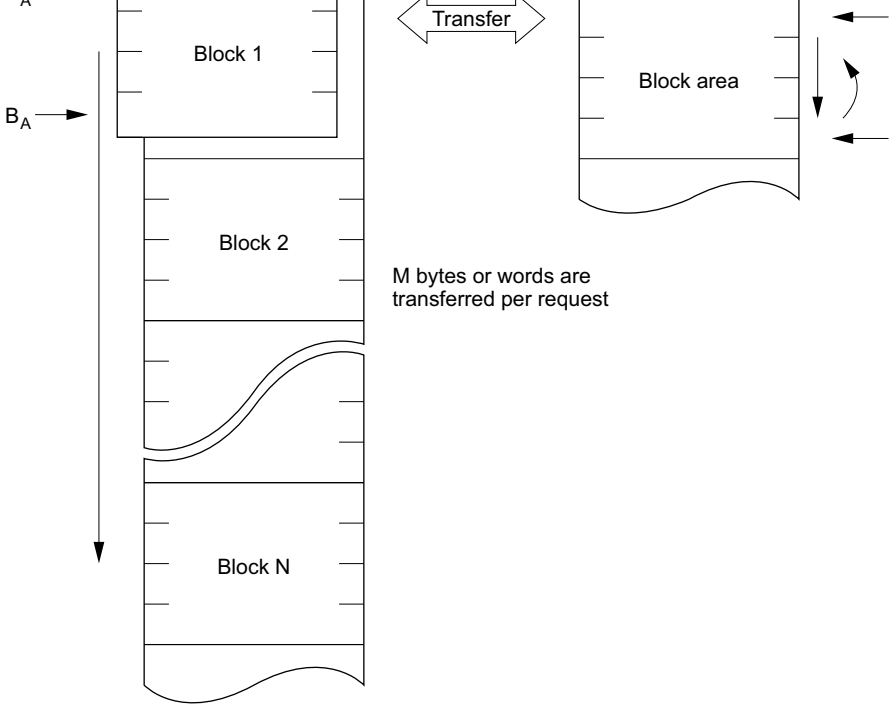
Figure 7.9 Normal Mode Setup Procedure (Example)

Register	Function	Initial Setting	Operation
	Source address register	Source start address	Incremented o decremented c transfer, or hel
	Destination address register	Destination start address	Incremented o decremented c transfer, or hel
	Block size counter	Block size	Decremented c transfer until H reached, then from ETCRL
	Initial block size	Block size	Held fixed
	Block transfer counter	Number of block transfers	Decremented c block transfer u is reached and transfer ends

Legend

MARA: Memory address register A
MARB: Memory address register B
ETCRA: Execute transfer count register A
ETCRB: Execute transfer count register B

The source and destination addresses are both 24-bit addresses. MARA specifies the so address. MARB specifies the destination address. MARA and MARB can be independe incremented, decremented, or held fixed as data is transferred. One of these registers op block area register: even if it is incremented or decremented, it is restored to its initial v end of each block transfer. The TMS bit in DTCRB selects whether the block area is th destination.



Legend
L_A = initial setting of MARA
L_B = initial setting of MARB
M = initial setting of ETCRAH and ETCRAL
N = initial setting of ETCRB
$T_A = L_A$
$B_A = L_A + SAIDE \cdot (-1)^{SAID} \cdot (2^{DTSZ} \cdot M - 1)$
$T_B = L_B$
$B_B = L_B + DAIDE \cdot (-1)^{DAID} \cdot (2^{DTSZ} \cdot M - 1)$

Figure 7.10 Operation in Block Transfer Mode

time.

Figure 7.11 shows examples of a block transfer with byte data size when the block area
destination. In (a) the block area address is cycled. In (b) the block area address is held

Transfers can be requested (activated) by compare match/input capture A interrupts fro
channels 0 to 2, by an A/D converter conversion-end interrupt, and by external request

For the detailed settings see section 7.3.4, Data Transfer Control Registers (DTCR).

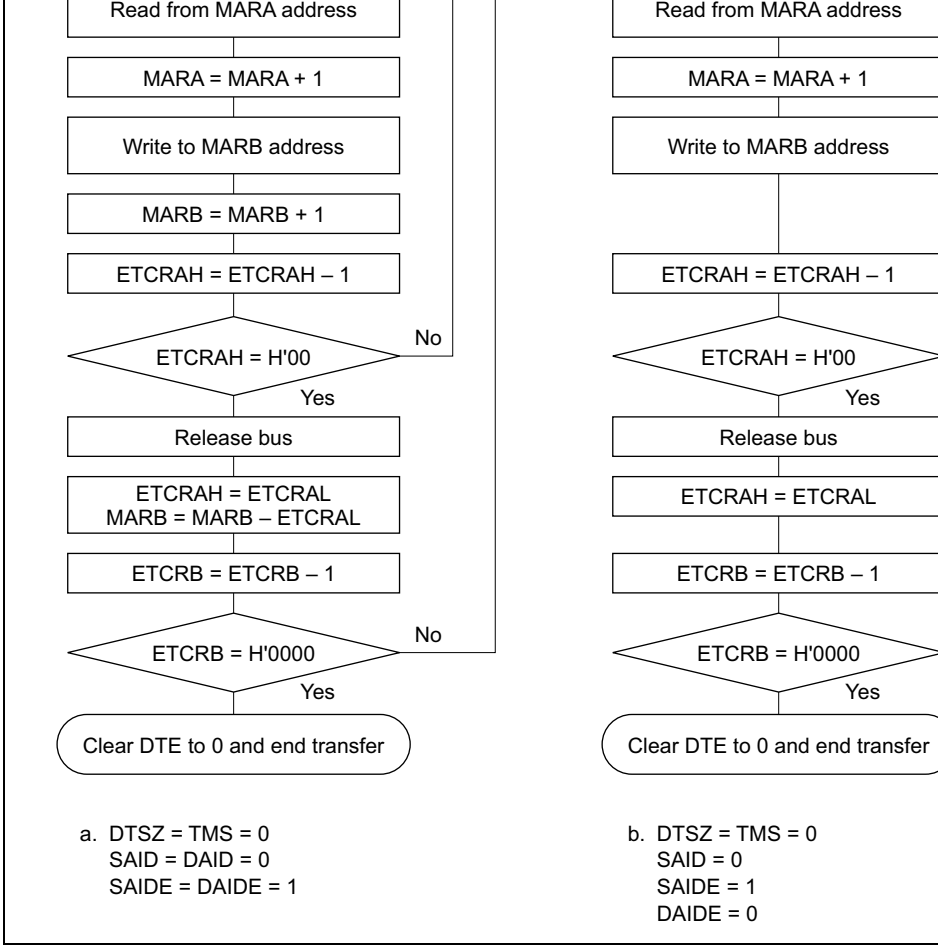
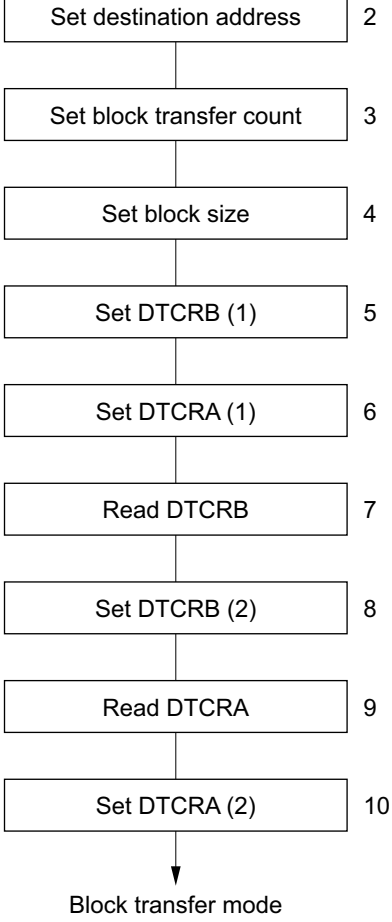


Figure 7.11 Block Transfer Mode Flowcharts (Examples)



- Set the DAID and DAIDE bits to select wh MARB is incremented, decremented, or h
- Set or clear the TMS bit to make the block the source or destination.
- Select the DMAC activation source with b DTS2B to DTS0B.

6. Set the DTCRA bits as follows.
   - Clear the DTE to 0.
   - Select byte size or word size with the DTS
   - Set the SAID and SAIDE bits to select wh MARA is incremented, decremented, or h
   - Set or clear the DTIE bit to enable or disa CPU interrupt at the end of the transfer.
   - Set bits DTS2A to DTS0A all to 1 to selec block transfer mode.
7. Read DTCRB with DTME cleared to 0.
8. Set the DTME bit to 1 in DTCRB.
9. Read DTCRA with DTE cleared to 0.
10. Set the DTE bit to 1 in DTCRA to enable the transfer.

Note: Carry out settings 1 to 10 with the DEND interrupt masked in the CPU.
If an NMI interrupt occurs during the setup procedure, it may clear the DTME b which case the transfer will not start.

Figure 7.12 Block Transfer Mode Setup Procedure (Example)

Activation Source		Channels 0A and 1A	Channels 0B and 1B	Full Address	
				Normal	Blo
Internal interrupts	IMIA0	○	○	×	○
	IMIA1	○	○	×	○
	IMIA2	○	○	×	○
	ADI	○	○	×	○
	TXI0	○	○	×	×
	RXI0	○	○	×	×
External requests	Falling edge of $\overline{\text{DREQ}}$	×	○	○	○
	Low input at $\overline{\text{DREQ}}$	×	○	○	×
Auto-request		×	×	○	×

Activation by Internal Interrupts: When an interrupt request is selected as a DMAC source and the DTE bit is set to 1, that interrupt request is not sent to the CPU. It is no for an interrupt request to activate the DMAC and simultaneously generate a CPU inte

When the DMAC is activated by an interrupt request, the interrupt request flag is clea automatically. If the same interrupt is selected to activate two or more channels, the in request flag is cleared when the highest-priority channel is activated, but the transfer r held pending on the other channels in the DMAC, which are activated in their priority

is completed. The bus is released temporarily after each byte or word has been transferi
however. If the $\overline{\text{DREQ}}$ input goes high during a transfer, the transfer is suspended after
byte or word has been transferred. When $\overline{\text{DREQ}}$ goes low, the request is held internally
byte or word has been transferred. The $\overline{\text{TEND}}$ signal goes low during the last write cyc

In block transfer mode, an external request operates as follows. Only edge-sensitive tra
requests are possible in block transfer mode. Each time a high-to-low transition of the $\overline{\text{D}}$
input is detected, a block of the specified size is transferred. The $\overline{\text{TEND}}$ signal goes low
last write cycle in each block.

Activation by Auto-Request: The transfer starts as soon as enabled by register setup, a
continues until completed. Cycle-steal mode or burst mode can be selected.

In cycle-steal mode the DMAC releases the bus temporarily after transferring each byte
Normally, DMAC cycles alternate with CPU cycles.

In burst mode the DMAC keeps the bus until the transfer is completed, unless there is a
priority bus request. If there is a higher-priority bus request, the bus is released after the
byte or word has been transferred.

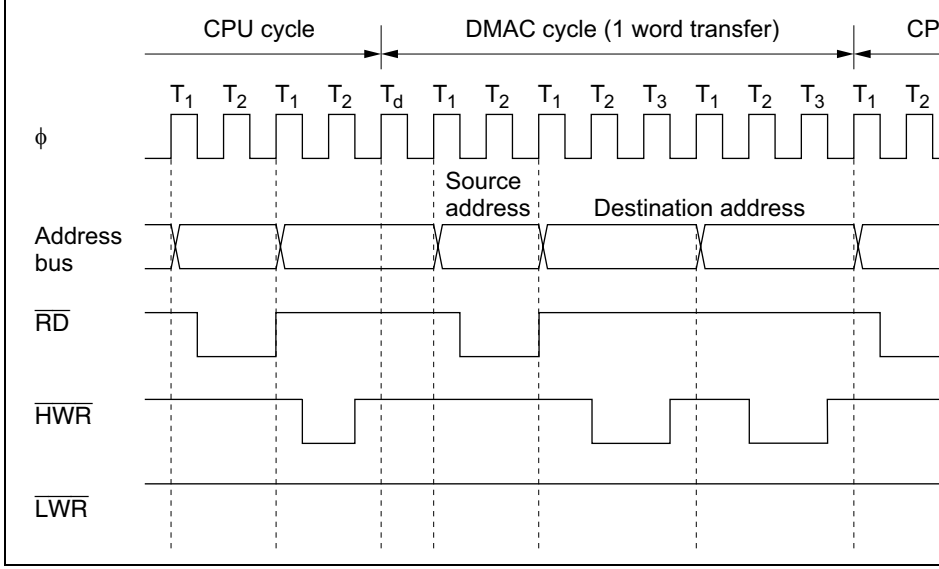


Figure 7.13 DMA Transfer Bus Timing (Example)

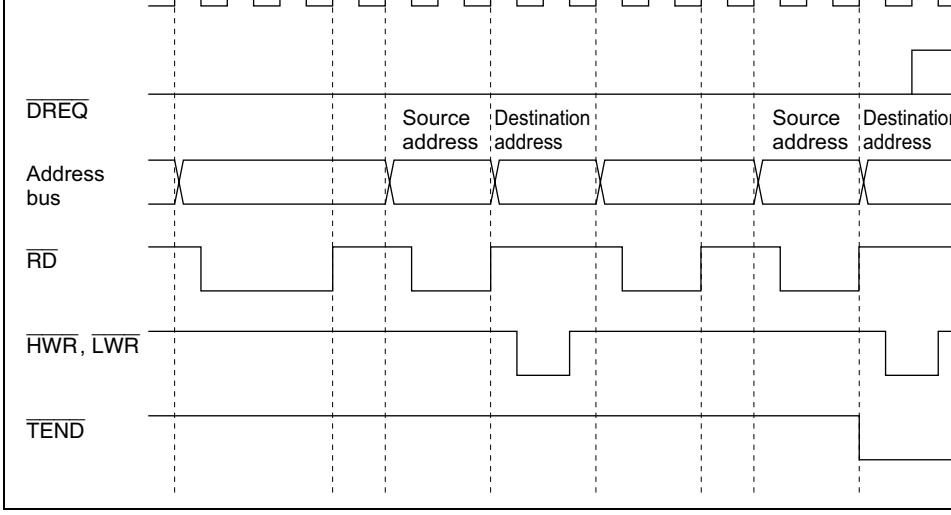


Figure 7.14 Bus Timing of DMA Transfer Requested by Low DREQ Inp

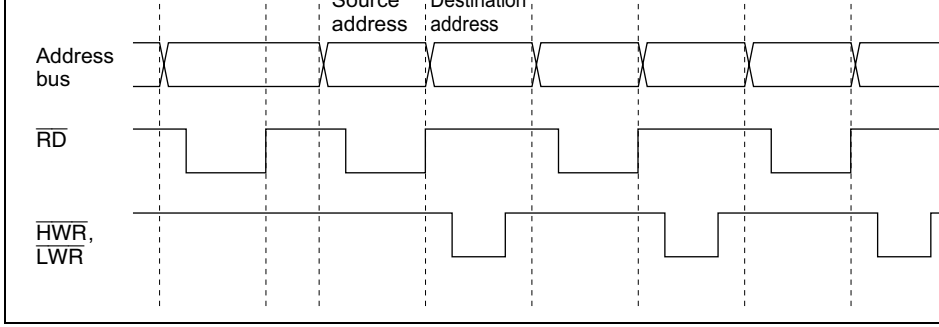


Figure 7.15 Burst DMA Bus Timing

When the DMAC is activated from a $\overline{\text{DREQ}}$ pin there is a minimum interval of four st
when the transfer is requested until the DMAC starts operating. The $\overline{\text{DREQ}}$ pin is not
during the time between the transfer request and the start of the transfer. In short addr
normal mode, the pin is next sampled at the end of the read cycle. In block transfer mo
is next sampled at the end of one block transfer.

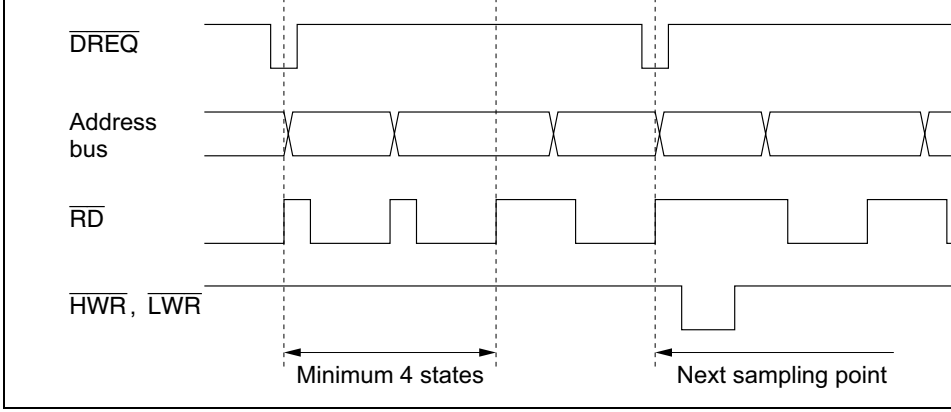


Figure 7.16 Timing of DMAC Activation by Falling Edge of DREQ in Norma

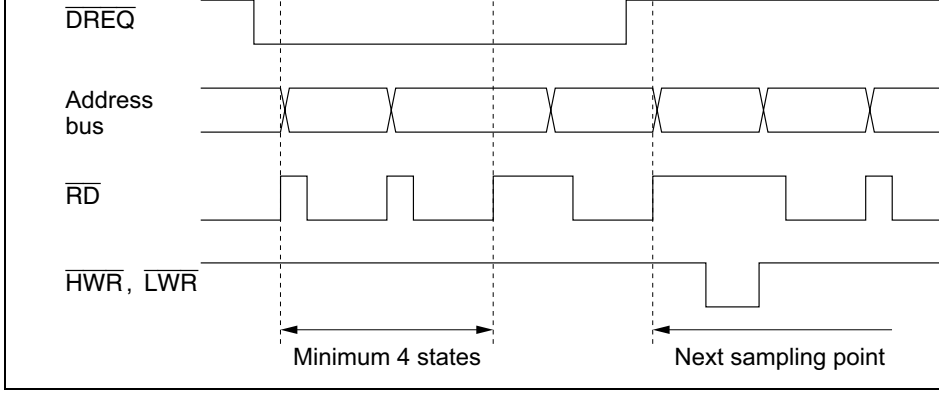


Figure 7.17 Timing of DMAC Activation by Low DREQ Level in Normal

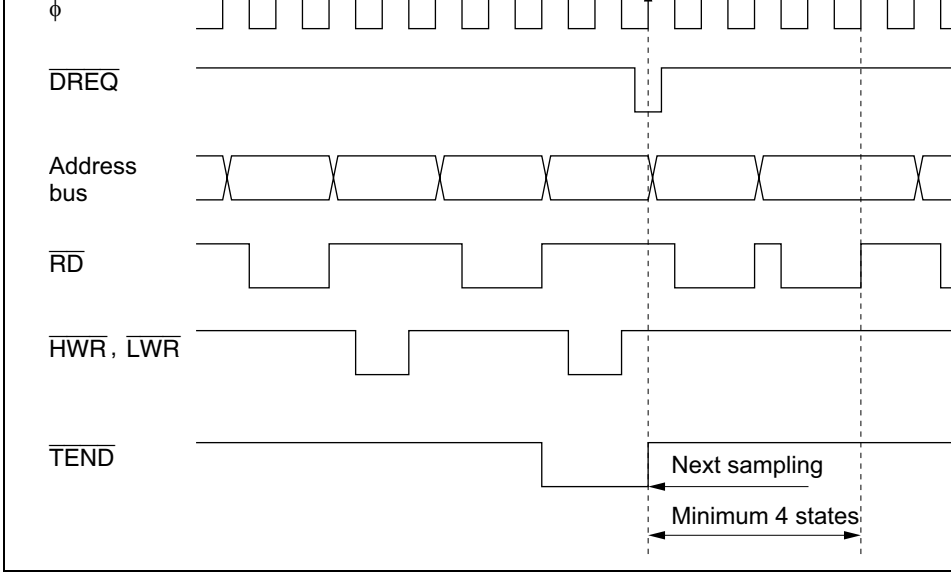


Figure 7.18 Timing of DMAC Activation by Falling Edge of DREQ in Block Tran

Channel 0B		
Channel 1A	Channel 1	
Channel 1B		Low

If transfers are requested on two or more channels simultaneously, or if a transfer on c is requested during a transfer on another channel, the DMAC operates as follows.

- When a transfer is requested, the DMAC requests the bus right. When it gets the b starts a transfer on the highest-priority channel at that time.
- Once a transfer starts on one channel, requests to other channels are held pending u channel releases the bus.
- After each transfer in short address mode, and each externally-requested or cycle-s in normal mode, the DMAC releases the bus and returns to step 1. After releasing there is a transfer request for another channel, the DMAC requests the bus again.
- After completion of a burst-mode transfer, or after transfer of one block in block tr mode, the DMAC releases the bus and returns to step 1. If there is a transfer reque higher-priority channel or a bus request from a higher-priority bus master, howeve DMAC releases the bus after completing the transfer of the current byte or word. A releasing the bus, if there is a transfer request for another channel, the DMAC requ again.

Figure 7.19 shows the timing when channel 0A is set up for I/O mode and channel 1 f mode, and a transfer request for channel 0A is received while channel 1 is active.

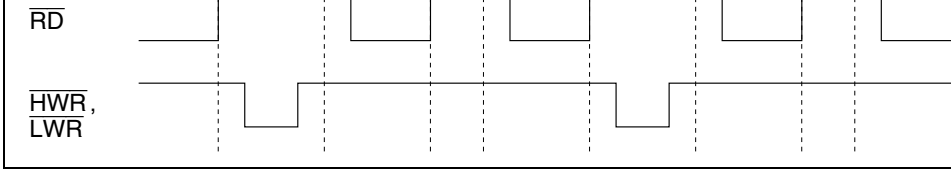


Figure 7.19 Timing of Multiple-Channel Operations

7.4.10 External Bus Requests, DRAM Interface, and DMAC

During a DMAC transfer, if the bus right is requested by an external bus request signal by the DRAM interface (refresh cycle), the DMAC releases the bus after completing th of the current byte or word. If there is a transfer request at this point, the DMAC reques right again. Figure 7.20 shows an example of the timing of insertion of a refresh cycle d burst transfer on channel 0.

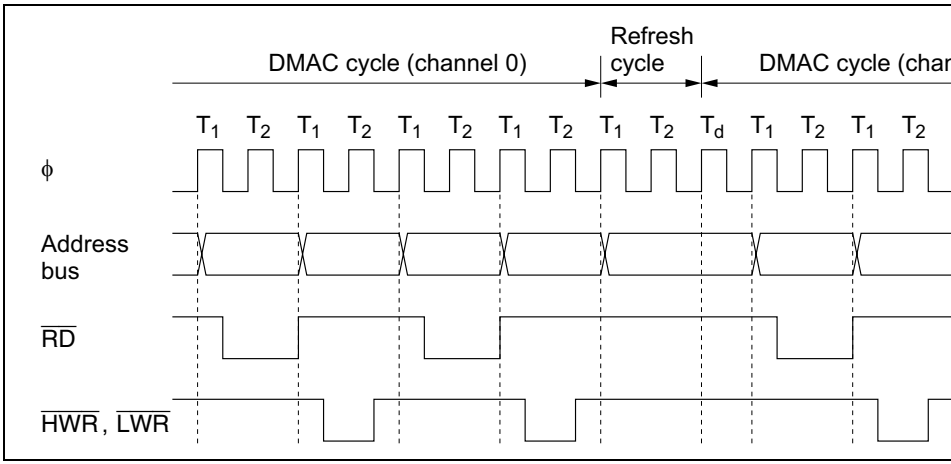


Figure 7.20 Bus Timing of DRAM Interface, and DMAC

the DTME bit to 1.

Figure 7.21 shows the procedure for resuming a DMAC transfer in normal mode on cl
after the transfer was halted by NMI input.

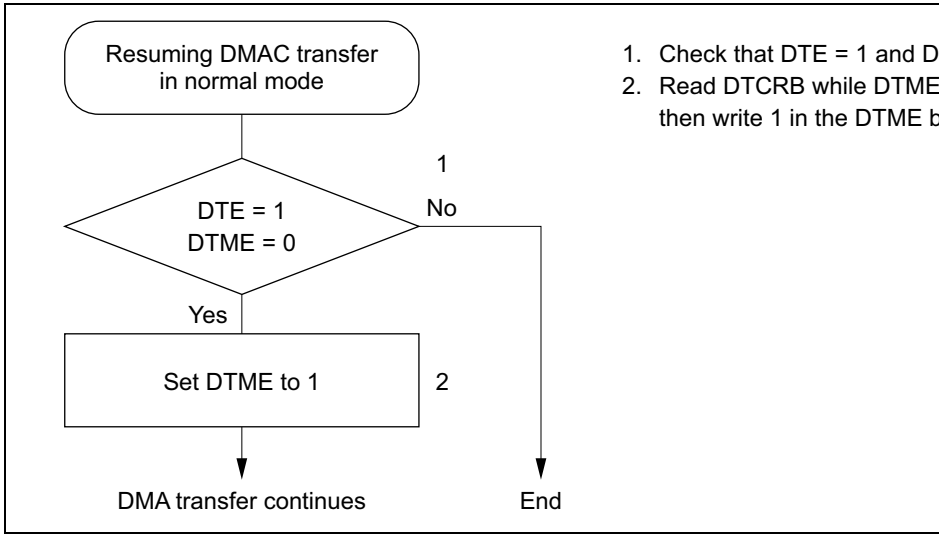


Figure 7.21 Procedure for Resuming a DMAC Transfer Halted by NMI (Ex

For information about NMI interrupts in block transfer mode, see section 7.6.6, NMI I
and Block Transfer Mode.

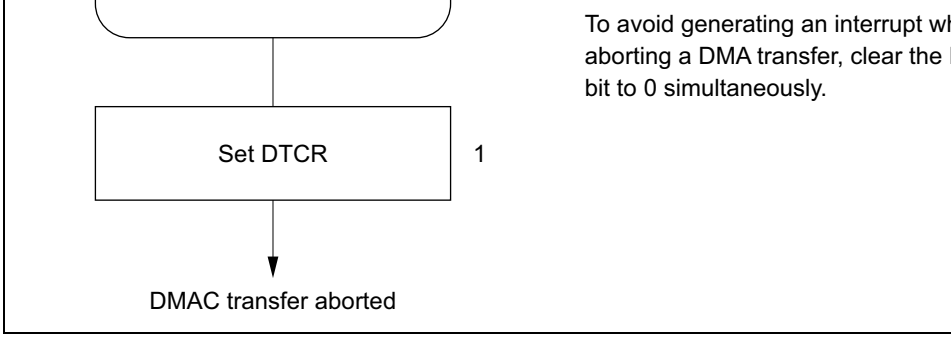


Figure 7.22 Procedure for Aborting a DMAC Transfer

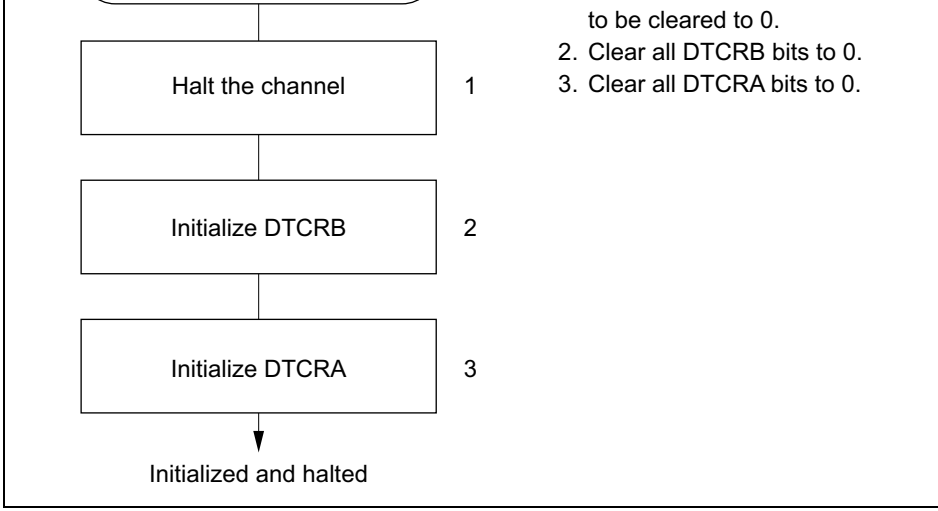


Figure 7.23 Procedure for Exiting Full Address Mode (Example)

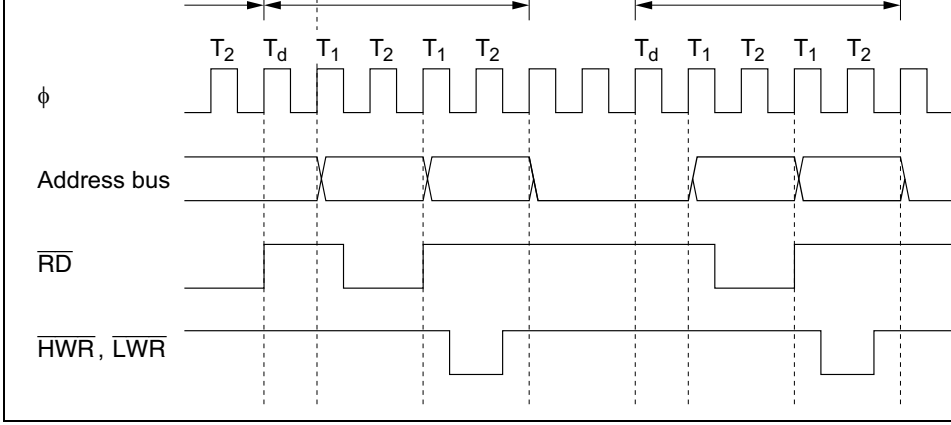


Figure 7.24 Timing of Cycle-Steal Transfer in Sleep Mode

DEND0B	End of transfer on channel 0B	—	
DEND1A	End of transfer on channel 1A	End of transfer on channel 1	
DEND1B	End of transfer on channel 1B	—	Low

Each interrupt is enabled or disabled by the DTIE bit in the corresponding data transfe
register (DTCR). Separate interrupt signals are sent to the interrupt controller.

The interrupt priority order among channels is channel 0 > channel 1 and channel A >

Figure 7.25 shows the DMA-end interrupt logic. An interrupt is requested whenever D
DTIE = 1.

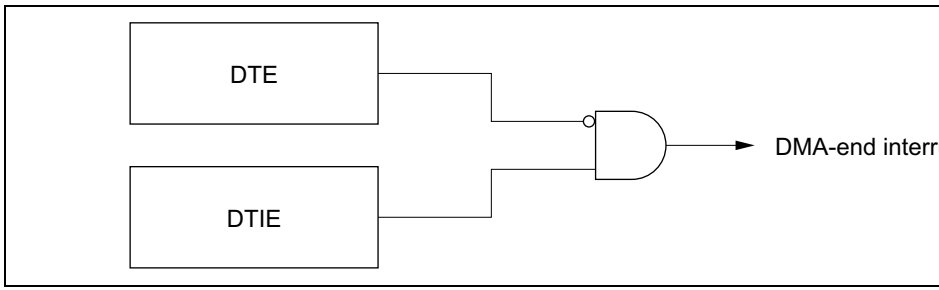


Figure 7.25 DMA-End Interrupt Logic

The DMA-end interrupt for the B channels (DENDB) is unavailable in full address mo
DTME bit does not affect interrupt operations.

The DMAC itself cannot be accessed during a DMAC cycle. DMAC registers cannot b
as source or destination addresses.

7.6.3 Longword Access to Memory Address Registers

A memory address register can be accessed as longword data at the MARR address.

Example

```
MOV.L  #LBL, ER0
MOV.L  ER0, @MARR
```

Four byte accesses are performed. Note that the CPU may release the bus between the s
(MARE) and third byte (MARH).

Memory address registers should be written and read only when the DMAC is halted.

7.6.4 Note on Full Address Mode Setup

Full address mode is controlled by two registers: DTCRA and DTCRB. Care must be ta
prevent the B channel from operating in short address mode during the register setup. T
bits (DTE and DTME) should not be set to 1 until the end of the setup procedure.

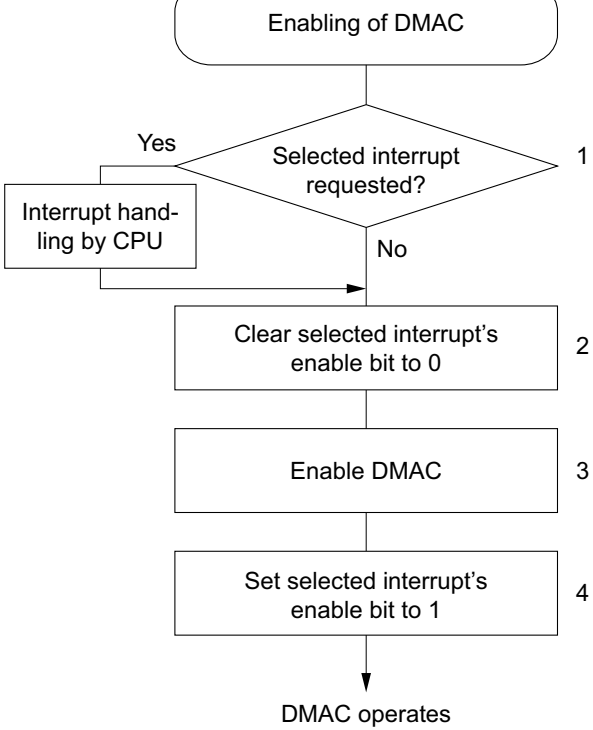


Figure 7.26 Procedure for Enabling DMAC while On-Chip Supporti Module is Operating (Example)

If the DTE bit is set to 1 but the DTME bit is cleared to 0, the DMAC is halted and th activating source cannot generate a CPU interrupt. If the DMAC is halted by an NMI example, the selected activating source cannot generate CPU interrupts. To terminate operations in this state, clear the DTE bit to 0 to allow CPU interrupts to be requested. DMAC operations, carry out steps 2 and 4 in figure 7.26 before and after setting the D to 1.

- When the NMI interrupt occurs, the DMAC finishes transferring the current byte or clears the DTME bit to 0 and halts. The halt may occur in the middle of a block.

  It is possible to find whether a transfer was halted in the middle of a block by check block size counter. If the block size counter does not have its initial value, the trans halted in the middle of a block.
- If the transfer is halted in the middle of a block, the activating interrupt flag is clear activation request is not held pending.
- While the DTE bit is set to 1 and the DTME bit is cleared to 0, the DMAC is halted not accept activating interrupt requests. If an activating interrupt occurs in this state DMAC does not operate and does not hold the transfer request pending internally. N CPU interrupt requested.

  For this reason, before setting the DTME bit to 1, first clear the enable bit of the act interrupt to 0. Then, after setting the DTME bit to 1, set the interrupt enable bit to 1 section 7.6.5, Note on Activating DMAC by Internal Interrupts.
- When the DTME bit is set to 1, the DMAC waits for the next transfer request. If it w in the middle of a block transfer, the rest of the block is transferred when the next tr request occurs. Otherwise, the next block is transferred when the next transfer reque

7.6.7 Memory and I/O Address Register Values

Table 7.14 indicates the address ranges that can be specified in the memory and I/O add registers (MAR and IOAR).

Table 7.14 Address Ranges Specifiable in MAR and IOAR

	1-Mbyte Mode	16-Mbyte Mode
MAR	H'00000 to H'FFFFF (0 to 1048575)	H'000000 to H'FFFFFF (0 to 16777215)
IOAR	H'FFF00 to H'FFFFF (1048320 to 1048575)	H'FFFF00 to H'FFFFFF (16776960 to 16777215)

MAR bits 23 to 20 are ignored in 1-Mbyte mode.

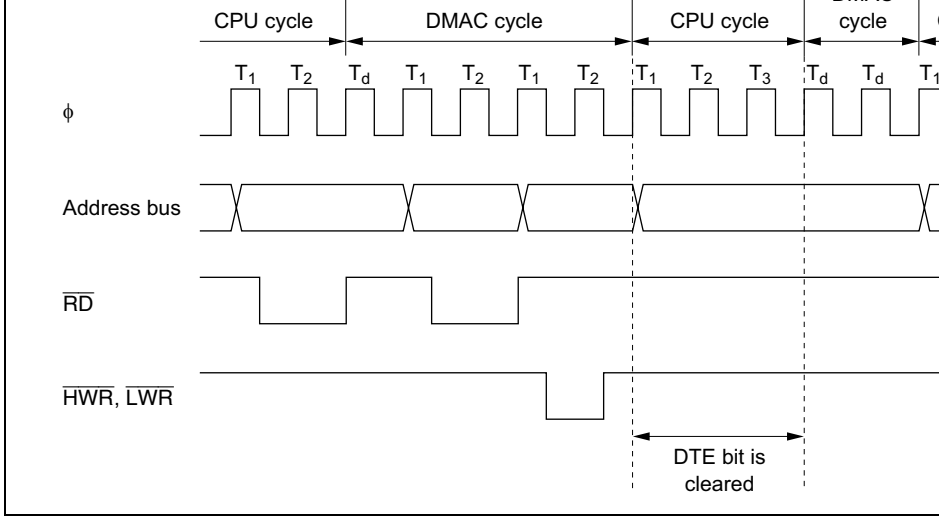


Figure 7.27 Bus Timing at Abort of DMA Transfer in Cycle-Steal Mo

7.6.9 Transfer Requests by A/D Converter

When the A/D converter is set to scan mode and conversion is performed on more tha channel, the A/D converter generates a transfer request when all conversions are comp converted data is stored in the appropriate ADDR registers. Block transfer mode and mode should therefore be used to transfer all the conversion results at one time.

(DR) for storing output data. In addition to these registers, ports 2, 4, and 5 have an in
control register (PCR) for switching input pull-up transistors on and off.

Ports 1 to 6 and port 8 can drive one TTL load and a 90-pF capacitive load. Ports 9, A
drive one TTL load and a 30-pF capacitive load. Ports 1 to 6 and 8 to B can drive a da
pair. Ports 1, 2, and 5 can drive LEDs (with 10-mA current sink). Pins $P8_2$ to $P8_0$, PA_7
Schmitt-trigger input circuits.

For block diagrams of the ports see appendix C, I/O Port Block Diagrams.

	up transistors • Can drive LEDs			DDR = 0: generic input DDR = 1: address output	
Port 3	• 8-bit I/O port	$P3_7$ to $P3_0$/ D_{15} to D_8	Data input/output (D_{15} to D_8)		Generic in
Port 4	• 8-bit I/O port • Built-in input pull-up transistors	$P4_7$ to $P4_0$/ D_7 to D_0	Data input/output (D_7 to D_0) and 8-bit generic input/output 8-bit bus mode: generic input/output 16-bit bus mode: data input/output		Generic in
Port 5	• 4-bit I/O port • Built-in input pull-up transistors • Can drive LEDs	$P5_3$ to $P5_0$/ A_{19} to A_{16}	Address output (A_{19} to A_{16})	Address output (A_{19} to A_{16}) and 4-bit generic input DDR = 0: generic input DDR = 1: address output	Generic in
Port 6	• 8-bit I/O port	$P6_7$/ϕ	Clock output (ϕ) and generic input		
		$P6_6$/$\overline{LWR}$ $P6_5$/$\overline{HWR}$ $P6_4$/$\overline{RD}$ $P6_3$/$\overline{AS}$	Bus control signal output ($\overline{LWR}$, $\overline{HWR}$, $\overline{RD}$, $\overline{AS}$)		Generic in
		$P6_2$/$\overline{BACK}$ $P6_1$/$\overline{BREQ}$ $P6_0$/$\overline{WAIT}$	Bus control signal input/output ($\overline{BACK}$, $\overline{BREQ}$, $\overline{WAIT}$) and 3-bit generic input/output		
Port 7	• 8-bit I/O port	$P7_7$/AN_7/DA_1 $P7_6$/AN_6/DA_0	Analog input (AN_7, AN_6) to A/D converter, analog output (DA_1, DA_0) from D/A converter, input		
		$P7_5$ to $P7_0$/ AN_5 to AN_0	Analog input (AN_5 to AN_0) to A/D converter, and generic input		
Port 8	• 5-bit I/O port • $P8_2$ to $P8_0$ have Schmitt inputs	$P8_4$/$\overline{CS_0}$	DDR = 0: generic input DDR = 1 (reset value): $\overline{CS_0}$ output	DDR = 0 (reset value): generic input DDR = 1: $\overline{CS_0}$ output	Generic in
		$P8_3$/$\overline{IRQ_3}$/ $\overline{CS_1}$/$\overline{ADTRG}$	$\overline{IRQ_3}$ input, $\overline{CS_1}$ output, external trigger input ($\overline{ADTRG}$) to A/D converter, and generic input DDR = 0 (after reset): generic input DDR = 1: $\overline{CS_1}$ output		$\overline{IRQ_3}$ input trigger inpu A/D conve generic inp
		$P8_2$/$\overline{IRQ_2}$/$\overline{CS_2}$ $P8_1$/$\overline{IRQ_1}$/$\overline{CS_3}$	$\overline{IRQ_2}$ and $\overline{IRQ_1}$ input, $\overline{CS_2}$ and $\overline{CS_3}$ output, and generic input* DDR = 0 (reset value): generic input DDR = 1: $\overline{CS_2}$ and $\overline{CS_3}$ output		$\overline{IRQ_2}$ and $\overline{I}$ generic inp
		$P8_0$/$\overline{IRQ_0}$/ $\overline{RFSH}$	$\overline{IRQ_0}$ input, $\overline{RFSH}$ output, and generic input/output		$\overline{IRQ_0}$ input input/outpu

Note: * $P8_1$ can be used as an output port by making a setting in DRCRA.

	• Schmitt inputs	$TIOCD_2/A_{20}$	grammable timing pattern controller (TPC), input or output ($TIOCB_2$) for 16-bit timer and generic input/output	(A_{20})	input or output ($TIOCB_2$) for 16-bit timer, and generic input/output	(TIOCE input/ou
		$PA_6/TP_6/TIOCA_2/A_{21}$ $PA_5/TP_5/TIOCB_1/A_{22}$ $PA_4/TP_4/TIOCA_1/A_{23}$	TPC output (TP_6 to TP_4), 16-bit timer input and output ($TIOCA_2$, $TIOCB_1$, $TIOCA_1$) , and generic input/output	TPC output (TP_6 to TP_4),16-bit timer input and output ($TIOCA_2$, $TIOCB_1$, $TIOCA_1$), address output (A_{23} to A_{21}), and generic input/output		TPC ou 16-bit ti output TIOCA input/ou
		$PA_3/TP_3/TIOCB_0/TCLKD$ $PA_2/TP_2/TIOCA_0/TCLKC$ $PA_1/TP_1/TCLKB/\overline{TEND}_1$ $PA_0/TP_0/TCLKA/\overline{TEND}_0$	TPC output (TP_3 to TP_0), 16-bit timer input and output ($TIOCB_0$, $TIOCA_0$, TCLKD, TCL TCLKA), 8-bit timer input (TCLKD, TCLKC, TCLKB, TCLKA), output ($\overline{TEND}_1$, $\overline{TEND}_0$) controller (DMAC), and generic input/output			
Port B	• 8-bit I/O port	$PB_7/TP_{15}/RXD_2$ $PB_6/TP_{14}/TXD_2$ $PB_5/TP_{13}/SCK_2/\overline{LCAS}$ $PB_4/TP_{12}/\overline{UCAS}$	TPC output (TP_{15} to TP_{12}), SCI2 input and output (SCK_2 , RxD_2, TxD_2), DRAM interface output ($\overline{LCAS}$, $\overline{UCAS}$), and generic input/output			TPC ou TP_{12}), S output TxD_2), a input/ou
		$PB_3/TP_{11}/TMIO_3/\overline{DREQ}_1/\overline{CS}_4$ $PB_2/TP_{10}/TMO_2/\overline{CS}_5$ $PB_1/TP_9/TMIO_1/\overline{DREQ}_0/\overline{CS}_6$ $PB_0/TP_8/TMO_0/\overline{CS}_7$	TPC output (TP_{11} to TP_8), 8-bit timer input and output ($TMIO_3$, TMO_2, $TMIO_1$, TMO_0), DMAC input ($\overline{DREQ}_1$, $\overline{DREQ}_0$), $\overline{CS}_7$ to $\overline{CS}_4$ output, and generic input/output			TPC ou TP_8), 8- and out TMO_2, DMAC $\overline{DREQ}_0$ input/ou

In mode 5 (expanded modes with on-chip ROM enabled), settings in the port 1 data dir register (P1DDR) can designate pins for address bus output (A_7 to A_0) or generic input. and 7 (single-chip mode), port 1 is a generic input/output port.

When DRAM is connected to area 2, 3, 4, 5, A_7 to A_0 output row and column addresses and write cycles. For details see section 6.5, DRAM Interface.

Pins in port 1 can drive one TTL load and a 90-pF capacitive load. They can also drive a darlington transistor pair.

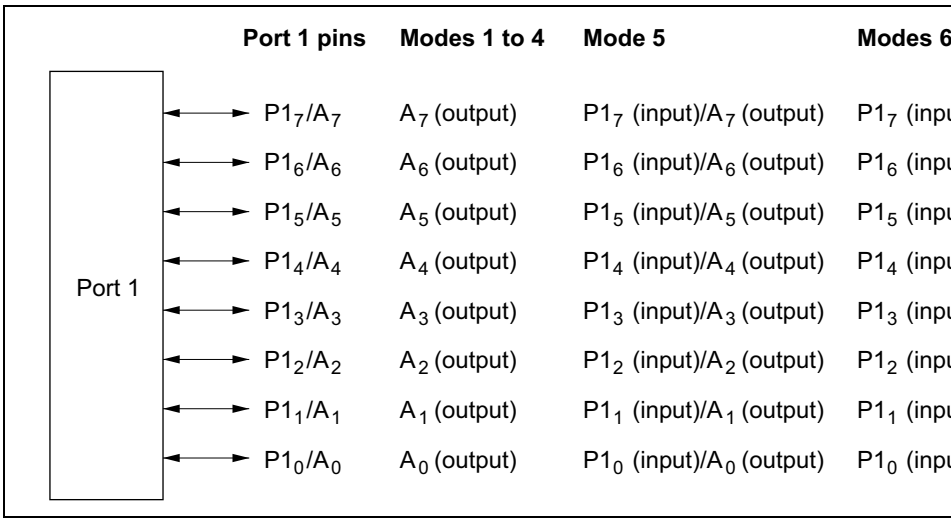


Figure 8.1 Port 1 Pin Configuration

H'FFFD0	Port 1 data register	P1DR	R/W	H'00	H'0

Note: * Lower 20 bits of the address in advanced mode.

Port 1 Data Direction Register (P1DDR): P1DDR is an 8-bit write-only register tha input or output for each pin in port 1.

Bit		7	6	5	4	3	2	1
		$P1_7DDR$	$P1_6DDR$	$P1_5DDR$	$P1_4DDR$	$P1_3DDR$	$P1_2DDR$	$P1_1DD$
Modes 1 to 4	Initial value	1	1	1	1	1	1	1
	Read/Write	—	—	—	—	—	—	—
Modes 5 to 7	Initial value	0	0	0	0	0	0	0
	Read/Write	W	W	W	W	W	W	W

Port 1 data direction 7 to 0
These bits select input or output for port 1 pins

Modes 1 to 4 (Expanded Modes with On-Chip ROM Disabled): P1DDR values are Port 1 functions as an address bus.

Mode 5 (Expanded Modes with On-Chip ROM Enabled): After a reset, port 1 func input port. A pin in port 1 becomes an address output pin if the corresponding P1DDR 1, and a generic input pin if this bit is cleared to 0.

Modes 6 and 7 (Single-Chip Mode): Port 1 functions as an input/output port. A pin i becomes an output port if the corresponding P1DDR bit is set to 1, and an input port i cleared to 0.

In modes 1 to 4, P1DDR bits are always read as 1, and cannot be modified.

In modes 5 to 7, P1DDR is a write-only register. Its value cannot be read. All bits retu read.

Bit	7	6	5	4	3	2	1
	$P1_7$	$P1_6$	$P1_5$	$P1_4$	$P1_3$	$P1_2$	$P1_1$
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

Port 1 data 7 to 0
These bits store data for port 1 pins

P1DR is initialized to H'00 by a reset and in hardware standby mode. In software stand
retains its previous setting.

port 2 data direction register (P2DDR) can designate pins for address bus output (A_{15} t
generic input. In modes 6 and 7 (single-chip mode), port 2 is a generic input/output po

When DRAM is connected to areas 2 to 5, A_{12} to A_8 output row and column addresses
write cycles. For details see section 6.5, DRAM Interface.

Port 2 has software-programmable built-in pull-up transistors.

Pins in port 2 can drive one TTL load and a 90-pF capacitive load. They can also driv
a darlington transistor pair.

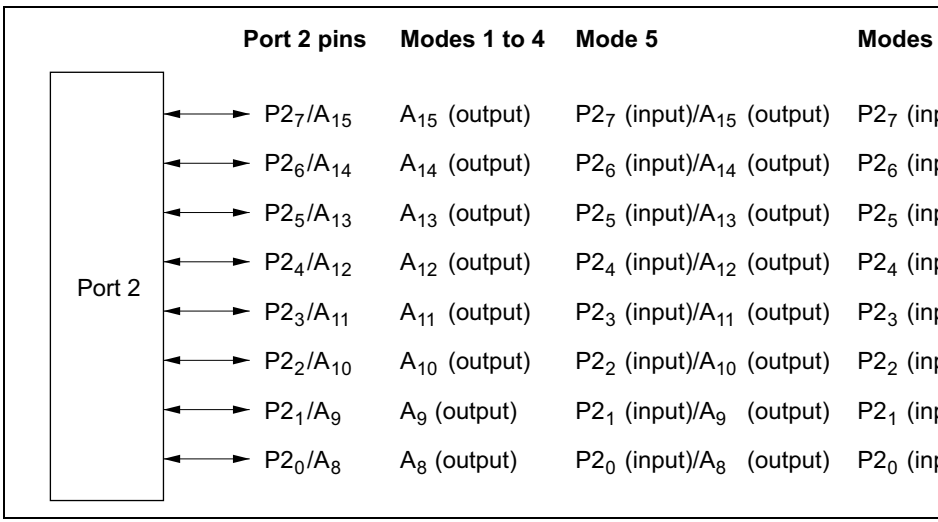


Figure 8.2 Port 2 Pin Configuration

H'FFFD1	Port 2 data register		P2DR	R/W	H'00	H'0
H'EE03C	Port 2 input pull-up register	MOS control	P2PCR	R/W	H'00	H'0

Note: * Lower 20 bits of the address in advanced mode.

Port 2 Data Direction Register (P2DDR): P2DDR is an 8-bit write-only register that input or output for each pin in port 2.

Bit		7	6	5	4	3	2	1
		$P2_7DDR$	$P2_6DDR$	$P2_5DDR$	$P2_4DDR$	$P2_3DDR$	$P2_2DDR$	$P2_1DDR$
Modes 1 to 4	Initial value	1	1	1	1	1	1	1
	Read/Write	—	—	—	—	—	—	—
Modes 5 to 7	Initial value	0	0	0	0	0	0	0
	Read/Write	W	W	W	W	W	W	W

Port 2 data direction 7 to 0
These bits select input or output for port 2 pins

Modes 1 to 4 (Expanded Modes with On-Chip ROM Disabled): P2DDR values are Port 2 functions as an address bus.

Mode 5 (Expanded Modes with On-Chip ROM Enabled): Following a reset, port 2 port. A pin in port 2 becomes an address output pin if the corresponding P2DDR bit is s a generic input port if this bit is cleared to 0.

Modes 6 and 7 (Single-Chip Mode): Port 2 functions as an input/output port. A pin in becomes an output port if the corresponding P2DDR bit is set to 1, and an input port if cleared to 0.

In modes 1 to 4, P2DDR bits are always read as 1, and cannot be modified.

In modes 5 to 7, P2DDR is a write-only register. Its value cannot be read. All bits retur read.

Bit	7	6	5	4	3	2	1
	$P2_7$	$P2_6$	$P2_5$	$P2_4$	$P2_3$	$P2_2$	$P2_1$
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

Port 2 data 7 to 0
These bits store data for port 2 pins

P2DR is initialized to H'00 by a reset and in hardware standby mode. In software stan
retains its previous setting.

Port 2 Input Pull-Up MOS Control Register (P2PCR): P2PCR is an 8-bit readable/
register that controls the MOS input pull-up transistors in port 2.

Bit	7	6	5	4	3	2	1
	$P2_7$PCR	$P2_6$PCR	$P2_5$PCR	$P2_4$PCR	$P2_3$PCR	$P2_2$PCR	$P2_1$PCR
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

Port 2 input pull-up MOS control 7 to 0
These bits control input pull-up
transistors built into port 2

In modes 5 to 7, when a P2DDR bit is cleared to 0 (selecting generic input), if the corr
bit in P2PCR is set to 1, the input pull-up transistor is turned on.

P2PCR is initialized to H'00 by a reset and in hardware standby mode. In software sta
retains its previous setting.

7	

Legend

Off: The input pull-up transistor is always off.

On/off: The input pull-up transistor is on if P2PCR = 1 and P2DDR = 0. Otherwise, it is o

darlington transistor pair.

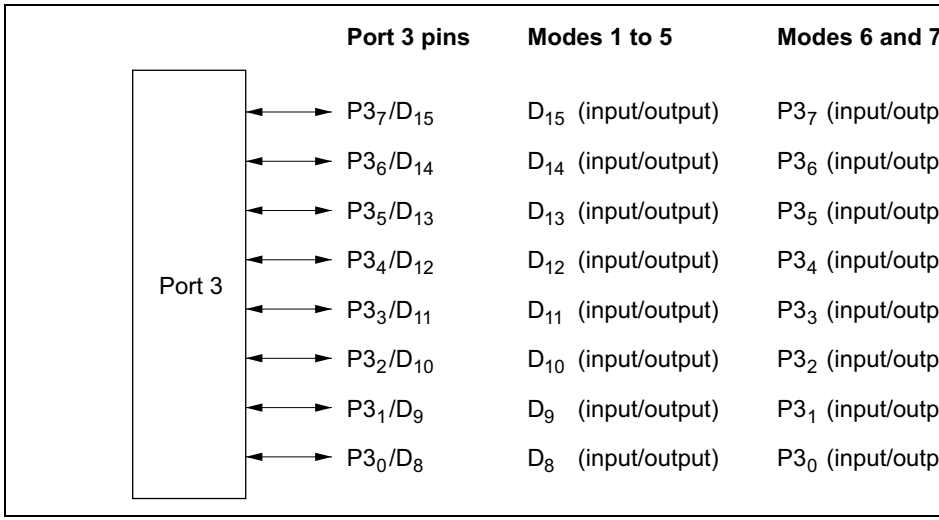


Figure 8.3 Port 3 Pin Configuration

8.4.2 Register Descriptions

Table 8.5 summarizes the registers of port 3.

Table 8.5 Port 3 Registers

Address*	Name	Abbreviation	R/W	In
H'EE002	Port 3 data direction register	P3DDR	W	H'
H'FFFD2	Port 3 data register	P3DR	R/W	H'

Note: * Lower 20 bits of the address in advanced mode.

Port 3 data direction 7 to 0
These bits select input or output for port 3 pin

Modes 1 to 5 (Expanded Modes): Port 3 functions as a data bus, regardless of the P3D settings.

Modes 6 and 7 (Single-Chip Mode): Port 3 functions as an input/output port. A pin in becomes an output port if the corresponding P3DDR bit is set to 1, and an input port if cleared to 0.

P3DDR is a write-only register. Its value cannot be read. All bits return 1 when read.

P3DDR is initialized to H'00 by a reset and in hardware standby mode. In software stan it retains its previous setting. Therefore, if a transition is made to software standby mo port 3 is functioning as an input/output port and a P3DDR bit is set to 1, the correspond maintains its output state.

Port 3 Data Register (P3DR): P3DR is an 8-bit readable/writable register that stores d for port 3. When port 3 functions as an output port, the value of this register is output. V in P3DDR is set to 1, if port 3 is read the value of the corresponding P3DR bit is return bit in P3DDR is cleared to 0, if port 3 is read the corresponding pin logic level is read.

Bit	7	6	5	4	3	2	1
	$P3_7$	$P3_6$	$P3_5$	$P3_4$	$P3_3$	$P3_2$	$P3_1$
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

Port 3 data 7 to 0
These bits store data for port 3 pins

P3DR is initialized to H'00 by a reset and in hardware standby mode. In software stand retains its previous setting.

input/output port. When at least one of areas 0 to 7 is designated as a 16-bit-access are
operates in 16-bit bus mode and port 4 becomes part of the data bus. In modes 6 and 7
mode), port 4 is a generic input/output port.

Port 4 has software-programmable built-in pull-up transistors.

Pins in port 4 can drive one TTL load and a 90-pF capacitive load. They can also drive
darlington transistor pair.

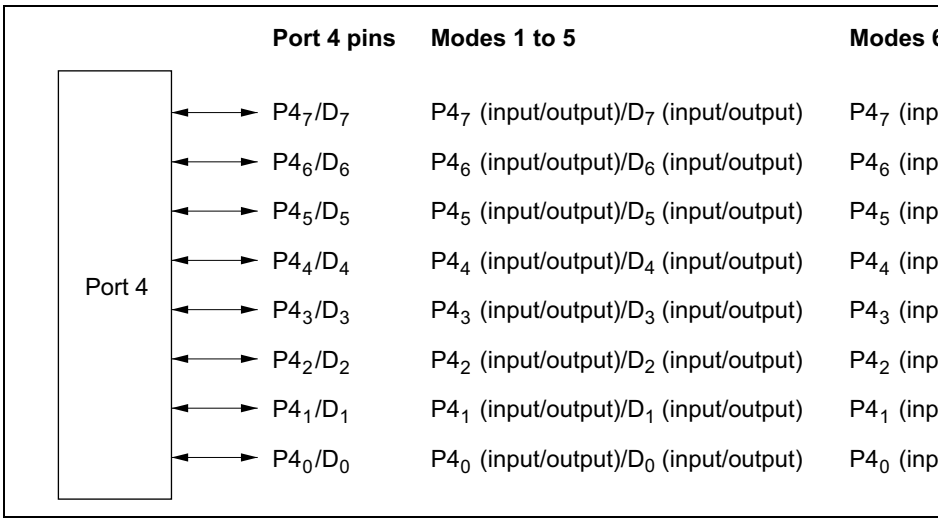


Figure 8.4 Port 4 Pin Configuration

H'EE03E	Port 4 input pull-up control register	P4PCR	R/W	H'0

Note: * Lower 20 bits of the address in advanced mode.

Port 4 Data Direction Register (P4DDR): P4DDR is an 8-bit write-only register that input or output for each pin in port 4.

Bit	7	6	5	4	3	2	1
	$P4_7DDR$	$P4_6DDR$	$P4_5DDR$	$P4_4DDR$	$P4_3DDR$	$P4_2DDR$	$P4_1DDR$
Initial value	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W

Port 4 data direction 7 to 0
These bits select input or output for port 4 pin

Modes 1 to 5 (Expanded Modes): When all areas are designated as 8-bit-access areas controller's bus width control register (ABWCR), selecting 8-bit bus mode, port 4 func input/output port. In this case, a pin in port 4 becomes an output port if the correspondi bit is set to 1, and an input port if this bit is cleared to 0.

When at least one area is designated as a 16-bit-access area, selecting 16-bit bus mode, functions as part of the data bus, regardless of the P4DDR settings.

Modes 6 and 7 (Single-Chip Mode): Port 4 functions as an input/output port. A pin in becomes an output port if the corresponding P4DDR bit is set to 1, and an input port if cleared to 0.

P4DDR is a write-only register. Its value cannot be read. All bits return 1 when read.

P4DDR is initialized to H'00 by a reset and in hardware standby mode. In software stan it retains its previous setting.

ABWCR and P4DDR are not initialized in software standby mode. Therefore, if a tran made to software standby mode while port 4 is functioning as an input/output port and bit is set to 1, the corresponding pin maintains its output state.

Bit	7	6	5	4	3	2	1
	$P4_7PCR$	$P4_6PCR$	$P4_5PCR$	$P4_4PCR$	$P4_3PCR$	$P4_2PCR$	$P4_1PCR$
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

Port 4 data 7 to 0
These bits store data for port 4 pins

P4DR is initialized to H'00 by a reset and in hardware standby mode. In software stan
retains its previous setting.

Port 4 Input Pull-Up MOS Control Register (P4PCR): P4PCR is an 8-bit readable/
register that controls the MOS input pull-up transistors in port 4.

Bit	7	6	5	4	3	2	1
	$P4_7PCR$	$P4_6PCR$	$P4_5PCR$	$P4_4PCR$	$P4_3PCR$	$P4_2PCR$	$P4_1PCR$
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

Port 4 input pull-up control 7 to 0
These bits control input pull-up transistors built

In modes 6 and 7 (single-chip mode), and in 8-bit bus mode in modes 1 to 5 (expande
when a P4DDR bit is cleared to 0 (selecting generic input), if the corresponding P4PC
to 1, the input pull-up transistor is turned on.

P4PCR is initialized to H'00 by a reset and in hardware standby mode. In software sta
retains its previous setting.

Legend

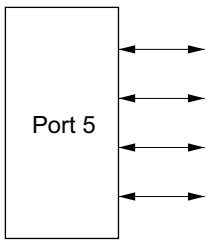
Off: The input pull-up transistor is always off.

On/off: The input pull-up transistor is on if P4PCR = 1 and P4DDR = 0. Otherwise, it is o

pins (A_{19} to A_{16}). In mode 5 (expanded modes with on-chip ROM enabled), settings in data direction register (P5DDR) designate pins for address bus output (A_{19} to A_{16}) or g input. In modes 6, 7 (single-chip mode), port 5 is a generic input/output port.

Port 5 has software-programmable built-in pull-up transistors.

Pins in port 5 can drive one TTL load and a 90-pF capacitive load. They can also drive a darlington transistor pair.



Port 5 pins	Modes 1 to 4	Mode 5	Modes
$P5_3/A_{19}$	A_{19} (output)	$P5_3$ (input)/A_{19} (output)	$P5_3$ (in
$P5_2/A_{18}$	A_{18} (output)	$P5_2$ (input)/A_{18} (output)	$P5_2$ (in
$P5_1/A_{17}$	A_{17} (output)	$P5_1$ (input)/A_{17} (output)	$P5_1$ (in
$P5_0/A_{16}$	A_{16} (output)	$P5_0$ (input)/A_{16} (output)	$P5_0$ (in

Figure 8.5 Port 5 Pin Configuration

8.6.2 Register Descriptions

Table 8.8 summarizes the registers of port 5.

Table 8.8 Port 5 Registers

Address*	Name	Abbreviation	R/W	Initial V: Modes 1 to 4	M
H'EE004	Port 5 data direction register	P5DDR	W	H'FF	H
H'FFFD4	Port 5 data register	P5DR	R/W	H'F0	H
H'EE03F	Port 5 input pull-up control register	P5PCR	R/W	H'F0	H

Note: * Lower 20 bits of the address in advanced mode.

Modes 5 to 7	Initial value	1	1	1	1	0	0	0
	Read/Write	—	—	—	—	W	W	W

Reserved bits

Port 5 data direction
These bits select inpu
output for port 5 pins

Modes 1 to 4 (Expanded Modes with On-Chip ROM Disabled): P5DDR values are
Port 5 functions as an address bus.

Modes 5 (Expanded Modes with On-Chip ROM Enabled): Following a reset, port 5
port. A pin in port 5 becomes an address output pin if the corresponding P5DDR bit is s
an input port if this bit is cleared to 0.

Mode 6 and 7 (Single-Chip Mode): Port 5 functions as an input/output port. A pin in p
becomes an output port if the corresponding P5DDR bit is set to 1, and an input port if
cleared to 0.

In modes 1 to 4, P5DDR bits are always read as 1, and cannot be modified.

In modes 5 to 7, P5DDR is a write-only register. Its value cannot be read. All bits retur
read.

P5DDR is initialized to H'FF in modes 1 to 4, and to H'F0 in modes 5 to 7, by a reset an
hardware standby mode. In software standby mode it retains its previous setting. There
transition is made to software standby mode while port 5 is functioning as an input/outp
a P5DDR bit is set to 1, the corresponding pin maintains its output state.

Initial value	1	1	1	1	0	0	0
Read/Write	—	—	—	—	R/W	R/W	R/W

Reserved bits

Port 5 data 3 to 0
These bits store d
for port 5 pins

P5DR is initialized to H'F0 by a reset and in hardware standby mode. In software stan
retains its previous setting.

Port 5 Input Pull-Up MOS Control Register (P5PCR): P5PCR is an 8-bit readable/
register that controls the MOS input pull-up transistors in port 5.

Bits 7 to 4 are reserved. They are fixed at 1, and cannot be modified.

Bit	7	6	5	4	3	2	1	
	—	—	—	—	$P5_3PCR$	$P5_2PCR$	$P5_1PCR$	
Initial value	1	1	1	1	0	0	0	
Read/Write	—	—	—	—	R/W	R/W	R/W	

Reserved bits

Port 5 input pull-up c
These bits control input
transistors built into por

In modes 5 to 7, when a P5DDR bit is cleared to 0 (selecting generic input), if the corr
bit in P5PCR is set to 1, the input pull-up transistor is turned on.

P5PCR is initialized to H'F0 by a reset and in hardware standby mode. In software sta
it retains its previous setting.

Table 8.9 summarizes the states of the input pull-ups in each mode.

Legend

Off: The input pull-up transistor is always off.

On/off: The input pull-up transistor is on if P5PCR = 1 and P5DDR = 0. Otherwise, it is o

In modes 6 and 7 (single-chip modes), $P6_7$ functions as a generic input port or ϕ outpu
$P6_0$ function as generic input/output ports.

When DRAM is connected to areas 2 to 5, $\overline{\text{LWR}}$, $\overline{\text{HWR}}$, and $\overline{\text{RD}}$ also function as $\overline{\text{LCA}}$
and $\overline{\text{WE}}$, respectively. For details see section 6.5, DRAM Interface.

Pins in port 6 can drive one TTL load and a 90-pF capacitive load. They can also drive
darlington transistor pair.

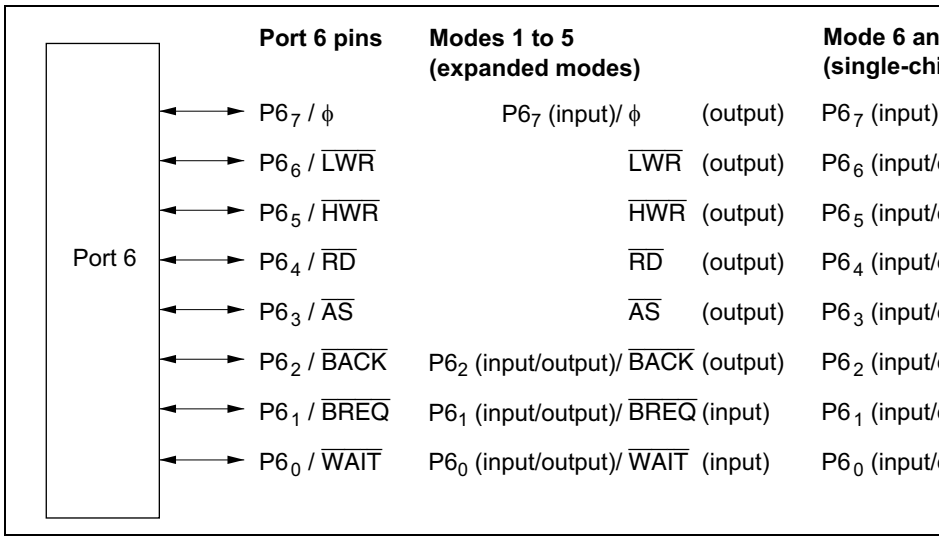
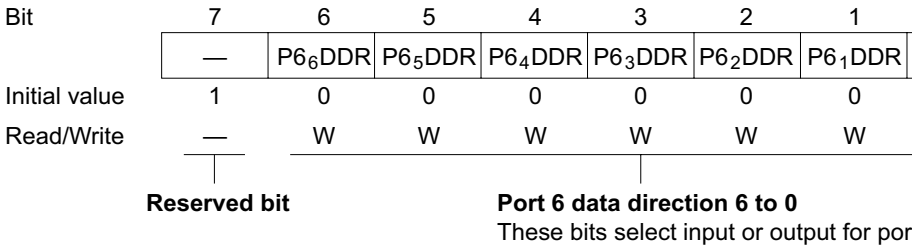


Figure 8.6 Port 6 Pin Configuration

Note: * Lower 20 bits of the address in advanced mode.

Port 6 Data Direction Register (P6DDR): P6DDR is an 8-bit write-only register that input or output for each pin in port 6.

Bit 7 is reserved. It is fixed at 1, and cannot be modified.



Modes 1 to 5 (Expanded Modes): P6$_7$ functions as the clock output pin (ϕ) or an input is the clock output pin (ϕ) if the PSTOP bit in MSTRCH is cleared to 0 (initial value), a port if this bit is set to 1.

P6$_6$ to P6$_3$ function as bus control output pins ($\overline{\text{LWR}}$, $\overline{\text{HWR}}$, $\overline{\text{RD}}$, and $\overline{\text{AS}}$), regardless o settings of bits P6$_6$DDR to P6$_3$DDR.

P6$_2$ to P6$_0$ function as bus control input/output pins ($\overline{\text{BACK}}$, $\overline{\text{BREQ}}$, and $\overline{\text{WAIT}}$) or inp ports. For the method of selecting the pin functions, see table 8.11.

When P6$_2$ to P6$_0$ function as input/output ports, the pin becomes an output port if the corresponding P6DDR bit is set to 1, and an input port if this bit is cleared to 0.

it retains its previous setting. Therefore, if a transition is made to software standby mo
port 6 is functioning as an input/output port and a P6DDR bit is set to 1, the correspon
maintains its output state.

Port 6 Data Register (P6DR): P6DR is an 8-bit readable/writable register that stores
for port 6. When port 6 functions as an output port, the value of this register is output.
value of 1 is returned if the bit is read while the PSTOP bit in MSTCRH is cleared to
$P6_7$ pin logic level is returned if the bit is read while the PSTOP bit is set to 1. Bit 7 c
modified. For bits 6 to 0, the pin logic level is returned if the bit is read while the corr
bit in P6DDR is cleared to 0, and the P6DR value is returned if the bit is read while th
corresponding bit in P6DDR is set to 1.

Bit	7	6	5	4	3	2	1
	$P6_7$	$P6_6$	$P6_5$	$P6_4$	$P6_3$	$P6_2$	$P6_1$
Initial value	1	0	0	0	0	0	0
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W

Port 6 data 7 to 0
These bits store data for port 6 pins

P6DR is initialized to H'80 by a reset and in hardware standby mode. In software stan
retains its previous setting.

$P6_6DDR$	0	1
Pin function	$\overline{LWR}$ output*	

Note: * If any of bits DRAS2 to DRAS0 in DRCRA is 1 and bit CSEL 1, LWR output functions as $\overline{LCAS}$.

$\overline{HWR}$ — Functions as $\overline{HWR}$ regardless of the setting of bit $P6_5DDR$

$P6_5DDR$	0	1
Pin function	$\overline{HWR}$ output*	

Note: * If any of bits DRAS2 to DRAS0 in DRCRA is 1 and bit CSEL 1, HWR output functions as $\overline{UCAS}$.

$\overline{RD}$ — Functions as $\overline{RD}$ regardless of the setting of bit $P6_4DDR$

$P6_4DDR$	0	1
Pin function	$\overline{RD}$ output*	

Note: * If any of bits DRAS2 to DRAS0 in DRCRA is 1, $\overline{RD}$ output fu $\overline{WE}$.

$\overline{AS}$ — Functions as $\overline{AS}$ regardless of the setting of bit $P6_3DDR$

$P6_3DDR$	0	1
Pin function	$\overline{AS}$ output	

$P6_2/\overline{BACK}$ — Bit BRLE in BRCR and bit $P6_2DDR$ select the pin function as follows

BRLE	0		1
$P6_2DDR$	0	1	—
Pin function	$P6_2$ input	$P6_2$ output	$\overline{BACK}$ ou

$P6_1/\overline{BREQ}$ — Bit BRLE in BRCR and bit $P6_1DDR$ select the pin function as follows

BRLE	0		1
$P6_1DDR$	0	1	—
Pin function	$P6_1$ input	$P6_1$ output	$\overline{BREQ}$ in

8.8 Port 7

8.8.1 Overview

Port 7 is an 8-bit input port that is also used for analog input to the A/D converter and output from the D/A converter. The pin functions are the same in all operating modes. shows the pin configuration of port 7.

See section 15, A/D Converter, for details of the A/D converter analog input pins, and D/A Converter, for details of the D/A converter analog output pins.

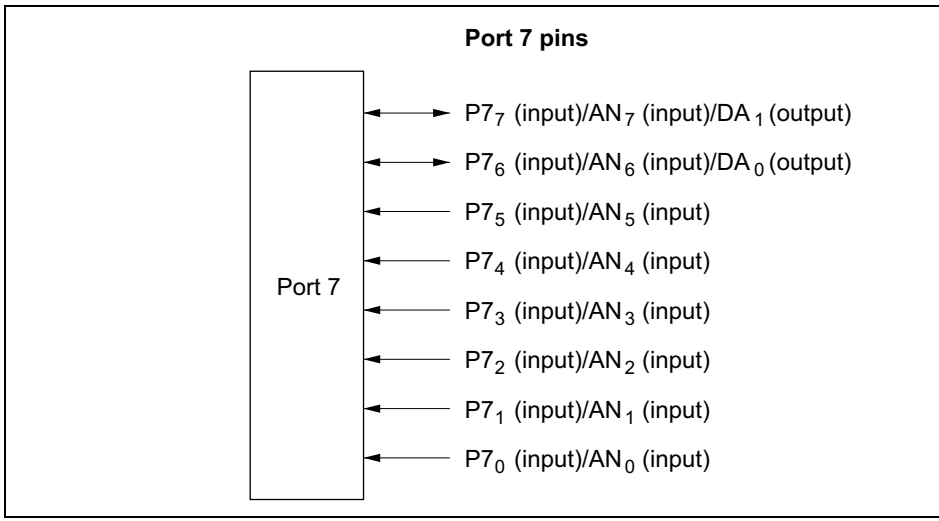


Figure 8.7 Port 7 Pin Configuration

Note: * Lower 20 bits of the address in advanced mode.

Port 7 Data Register (P7DR)

Bit	7	6	5	4	3	2	1
	$P7_7$	$P7_6$	$P7_5$	$P7_4$	$P7_3$	$P7_2$	$P7_1$
Initial value	—*	—*	—*	—*	—*	—*	—*
Read/Write	R	R	R	R	R	R	R

Note: * Determined by pins $P7_7$ to $P7_0$.

When port 7 is read, the pin logic levels are always read. P7DR cannot be modified.

modes.

In modes 6 and 7 (single-chip modes), port 8 can provide $\overline{IRQ_3}$ to $\overline{IRQ_0}$ input and $\overline{AD}$
See table 8.15 for the selection of pin functions in single-chip mode.

See section 15, A/D Converter, for a description of the A/D converter's $\overline{ADTRG}$ input

The $\overline{IRQ_3}$ to $\overline{IRQ_0}$ functions are selected by IER settings, regardless of whether the pi
input or output. Caution is therefore required. For details see section 5.3.1, External I

When DRAM is connected to areas 2 to 5, the $\overline{CS_3}$ and $\overline{CS_2}$ output pins function as $\overline{RA}$
pins for each area. For details see section 6.5, DRAM Interface.

Pins in port 8 can drive one TTL load and a 90-pF capacitive load. They can also driv
darlington transistor pair.

Pins $P8_2$ to $P8_0$ have Schmitt-trigger inputs.

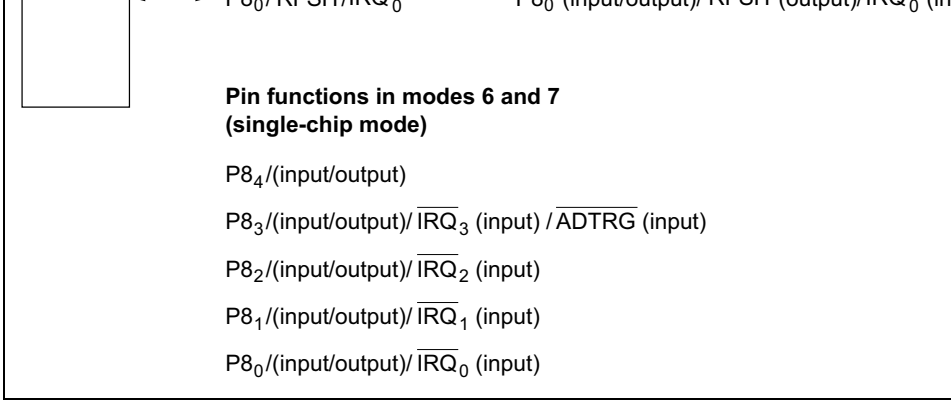


Figure 8.8 Port 8 Pin Configuration

	register				
H'FFFD7	Port 8 data register	P8DR	R/W	H'E0	H'E

Note: * Lower 20 bits of the address in advanced mode.

Port 8 Data Direction Register (P8DDR): P8DDR is an 8-bit write-only register tha input or output for each pin in port 8.

Bits 7 to 5 are reserved. They are fixed at 1, and cannot be modified.

Bit		7	6	5	4	3	2	1
		—	—	—	$P8_4$DDR	$P8_3$DDR	$P8_2$DDR	$P8_1$DD
Modes 1 to 4	Initial value	1	1	1	1	0	0	0
	Read/Write	—	—	—	W	W	W	W
Modes 5 to 7	Initial value	1	1	1	0	0	0	0
	Read/Write	—	—	—	W	W	W	W

Reserved bits

Port 8 data direction 4 to
These bits select input or output for port 8 pins

Modes 1 to 5 (Expanded Modes): When bits in P8DDR bit are set to 1, $P8_4$ to $P8_1$ be $\overline{CS}_3$ output pins. When bits in P8DDR are cleared to 0, the corresponding pins become However, $P8_1$ can also be used as an output port, depending on the setting of bits DRA DRAS0 in DRAM control register A (DRCRA). For details see section 6.5.2, DRAM $\overline{RAS}$ Output Pin Settings.

In modes 1 to 4 (expanded modes with on-chip ROM disabled), following a reset $P8_4$ the $\overline{CS}_0$ output, while $\overline{CS}_1$ to $\overline{CS}_3$ are input ports. In mode 5 (expanded mode with on- enabled), following a reset $\overline{CS}_0$ to $\overline{CS}_3$ are all input ports.

When the refresh enable bit (RFSHE) in DRCRA is set to 1, $P8_0$ is used for $\overline{RFSH}$ out RFSHE is cleared to 0, $P8_0$ becomes an input/output port according to the P8DDR sett details see table 8.14.

and a P8DDR bit is set to 1, the corresponding pin maintains its output state.

Port 8 Data Register (P8DR): P8DR is an 8-bit readable/writable register that stores d
for port 8. When port 8 functions as an output port, the value of this register is output. V
in P8DDR is set to 1, if port 8 is read the value of the corresponding P8DR bit is return
a bit in P8DDR is cleared to 0, if port 8 is read the corresponding pin logic level is read

Bits 7 to 5 are reserved. They are fixed at 1, and cannot be modified.

Bit	7	6	5	4	3	2	1
	—	—	—	$P8_4$	$P8_3$	$P8_2$	$P8_1$
Initial value	1	1	1	0	0	0	0
Read/Write	—	—	—	R/W	R/W	R/W	R/W

Reserved bits

Port 8 data 4 to 0
These bits store data for port 8 pins

P8DR is initialized to H'E0 by a reset and in hardware standby mode. In software stand
retains its previous setting.

$P8_3DDR$	0	1
Pin function	$P8_3$ input	$\overline{CS}_1$ ou
	$\overline{IRQ}_3$ input	
	$\overline{ADTRG}$ input	

$P8_2/\overline{CS}_2/\overline{IRQ}_2$ The DRAM interface settings by bits DRAS2 to DRAS0 in DRCRA, a $P8_2DDR$, select the pin function as follows.

DRAM interface settings	(1) in table below		(2) in table
$P8_2DDR$	0	1	—
Pin function	$P8_2$ input	$\overline{CS}_2$ output	$\overline{CS}_2$ out
	$\overline{IRQ}_3$ input		

Note: * $\overline{CS}_2$ is output as $\overline{RAS}_2$.

DRAM interface setting	(1)		(2)				
DRAS2	0				1		
DRAS1	0		1		0		
DRAS0	0	1	0	1	0	1	

	$\overline{IRQ}_1$ input pin						

Note: * $\overline{CS}_3$ is output as $\overline{RAS}_3$.

DRAM interface setting	(1)		(3)	(2)	(3)		
DRAS2	0				1		
DRAS1	0		1		0		
DRAS0	0	1	0	1	0	1	

$P8_0/\overline{RFSH}/\overline{IRQ}_0$ Bit RFSHE in DRCRA and bit $P8_0DDR$ select the pin function as follow

RFSHE	0		1*
$P8_0DDR$	0	1	—
Pin function	$P8_0$ input	$P8_0$ output	$\overline{RFSH}$ ou
	$\overline{IRQ}_0$ input		

Note: * If areas 2 to 5 are not designated as DRAM space, this bit sh
set to 1.

$P8_3$DDR	0	1
Pin function	$P8_3$ input	$P8_3$ ou
	$\overline{IRQ}_3$ input	
	$\overline{ADTRG}$ input	

$P8_2/\overline{IRQ}_2$ Bit $P8_2$DDR selects the pin function as follows

$P8_2$DDR	0	1
Pin function	$P8_2$ input	$P8_2$ ou
	$\overline{IRQ}_2$ input	

$P8_1/\overline{IRQ}_1$ Bit $P8_1$DDR selects the pin function as follows

$P8_1$DDR	0	1
Pin function	$P8_1$ input	$P8_1$ ou
	$\overline{IRQ}_1$ input	

$P8_0/\overline{IRQ}_0$ Bit $P8_0$DDR select the pin function as follows

$P8_0$DDR	0	1
Pin function	$P8_0$ input	$P8_0$ ou
	$\overline{IRQ}_0$ input	

for input or output. Caution is therefore required. For details see section 5.3.1, External

Port 9 has the same set of pin functions in all operating modes. Figure 8.9 shows the pi
configuration of port 9.

Pins in port 9 can drive one TTL load and a 30-pF capacitive load. They can also drive
darlington transistor pair.

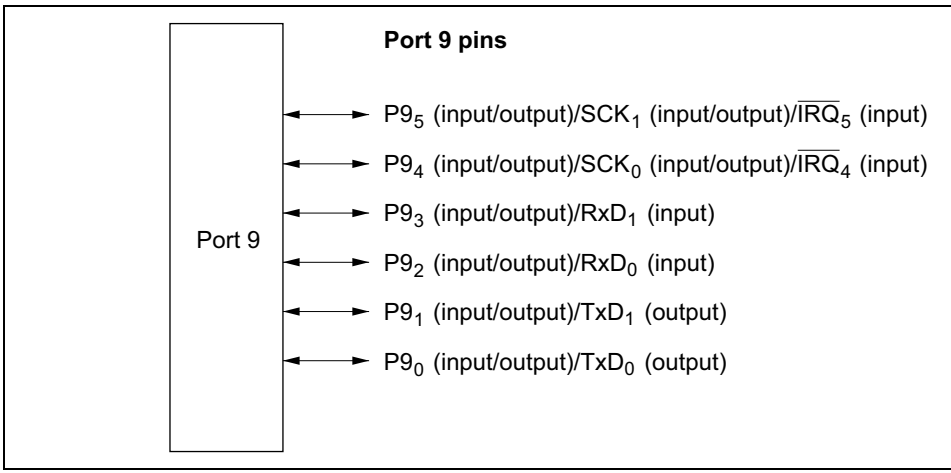
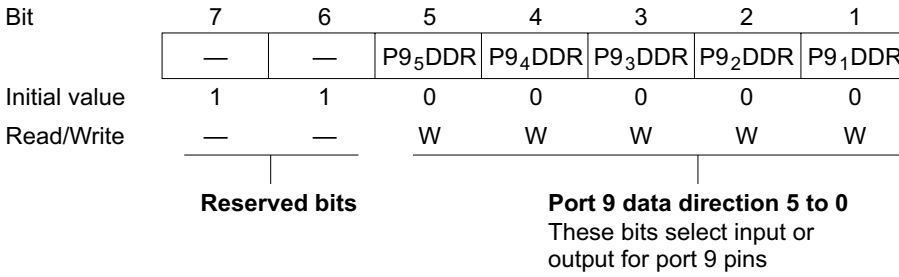


Figure 8.9 Port 9 Pin Configuration

Note: * Lower 20 bits of the address in advanced mode.

Port 9 Data Direction Register (P9DDR): P9DDR is an 8-bit write-only register tha input or output for each pin in port 9.

Bits 7 and 6 are reserved. They are fixed at 1, and cannot be modified.



When port 9 functions as an input/output port, a pin in port 9 becomes an output port i corresponding P9DDR bit is set to 1, and an input port if this bit is cleared to 0. For th selecting the pin functions, see table 8.17.

P9DDR is a write-only register. Its value cannot be read. All bits return 1 when read.

P9DDR is initialized to H'C0 by a reset and in hardware standby mode. In software st it retains its previous setting. Therefore, if a transition is made to software standby m port 9 is functioning as an input/output port and a P9DDR bit is set to 1, the correspon maintains its output state.

Initial value	1	1	0	0	0	0	0
Read/Write	—	—	R/W	R/W	R/W	R/W	R/W

Reserved bits

Port 9 data 5 to 0
These bits store data for port 9 pins

P9DR is initialized to H'C0 by a reset and in hardware standby mode. In software stand
retains its previous setting.

Pin function	$P9_5$ input	$P9_5$ output	SCK_1 output	SCK_1 output
	$\overline{IRQ_5}$ input			

$P9_4/SCK_0/\overline{IRQ_4}$

Bit C/$\overline{A}$ in SMR of SCI0, bits CKE0 and CKE1 in SCR, and bit $P9_4$DD pin function as follows

CKE1	0			
C/$\overline{A}$	0			1
CKE0	0		1	—
$P9_4$DDR	0	1	—	—
Pin function	$P9_4$ input	$P9_4$ output	SCK_0 output	SCK_0 output
	$\overline{IRQ_4}$ input			

$P9_3/RxD_1$

Bit RE in SCR of SCI1, bit SMIF in SCMR, and bit $P9_3$DDR select th as follows.

SMIF	0			
RE	0		1	
$P9_3$DDR	0	1	—	
Pin function	$P9_3$ input	$P9_3$ output	RxD_1 input	

$P9_2/RxD_0$

Bit RE in SCR of SCI0, bit SMIF in SCMR, and bit $P9_2$DDR select th as follows

SMIF	0			
RE	0		1	
$P9_2$DDR	0	1	—	
Pin function	$P9_2$ input	$P9_2$ output	RxD_0 input	

the pin is driven, and another in which the pin is at high-impe

$P9_0/TxD_0$

Bit TE in SCR of SCI0, bit SMIF in SCMR, and bit $P9_0DDR$ select the
as follows.

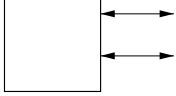
SMIF	0			
TE	0		1	
$P9_0DDR$	0	1	—	
Pin function	$P9_0$ input	$P9_0$ output	TxD_0 output	T

Note: * Functions as the TxD_0 output pin, but there are two states: o
the pin is driven, and another in which the pin is at highimpe

controller (DMAC), and address output (A_{23} to A_{20}). A reset or hardware standby tran
port A as an input port, except that in modes 3 and 4, one pin is always used for A_{20} ou
table 8.19 to 8.21 for the selection of pin functions.

Usage of pins for TPC, 16-bit timer, 8-bit timer, and DMAC input and output is descri
sections on those modules. For output of address bits A_{23} to A_{20} in modes 3, 4, and 5,
6.2.4, Bus Release Control Register (BRCR). Pins not assigned to any of these functi
available for generic input/output. Figure 8.10 shows the pin configuration of port A.

Pins in port A can drive one TTL load and a 30-pF capacitive load. They can also driv
darlington transistor pair. Port A has Schmitt-trigger inputs.



Pin functions in modes 1, 2, 6, and 7

PA 7 (input/output)/TP 7 (output)/TIOCB 2 (input/output)
PA 6 (input/output)/TP 6 (output)/TIOCA 2 (input/output)
PA 5 (input/output)/TP 5 (output)/TIOCB 1 (input/output)
PA 4 (input/output)/TP 4 (output)/TIOCA 1 (input/output)
PA 3 (input/output)/TP 3 (output)/TIOCB 0 (input/output)/TCLKD (input)
PA 2 (input/output)/TP 2 (output)/TIOCA 0 (input/output)/TCLKC (input)
PA 1 (input/output)/TP 1 (output)/$\overline{\text{TEND}}$ 1 (output)/TCLKB (input)
PA 0 (input/output)/TP 0 (output)/$\overline{\text{TEND}}$ 0 (output)/TCLKA (input)

Pin functions in modes 3 and 4

A_{20} (output)
PA 6 (input/output)/TP 6 (output)/TIOCA 2 (input/output)/A 21 (output)
PA 5 (input/output)/TP 5 (output)/TIOCB 1 (input/output)/A 22 (output)
PA 4 (input/output)/TP 4 (output)/TIOCA 1 (input/output)/A 23 (output)
PA 3 (input/output)/TP 3 (output)/TIOCB 0 (input/output)/TCLKD (input)
PA 2 (input/output)/TP 2 (output)/TIOCA 0 (input/output)/TCLKC (input)
PA 1 (input/output)/TP 1 (output)/$\overline{\text{TEND}}$ 1 (output)/TCLKB (input)
PA 0 (input/output)/TP 0 (output)/$\overline{\text{TEND}}$ 0 (output)/TCLKA (input)

Pin functions in mode 5

PA_7 (input/output)/TP_7 (output)/$TIOCB_2$ (input/output)/A_{20} (output)
PA_6 (input/output)/TP_6 (output)/$TIOCA_2$ (input/output)/A_{21} (output)
PA_5 (input/output)/TP_5 (output)/$TIOCB_1$ (input/output)/A_{22} (output)
PA_4 (input/output)/TP_4 (output)/$TIOCA_1$ (input/output)/A_{23} (output)
PA_3 (input/output)/TP_3 (output)/$TIOCB_0$ (input/output)/TCLKD (input)
PA_2 (input/output)/TP_2 (output)/$TIOCA_0$ (input/output)/TCLKC (input)
PA_1 (input/output)/TP_1 (output)/$\overline{TEND_1}$ (output)/TCLKB (input)
PA_0 (input/output)/TP_0 (output)/$\overline{TEND_0}$ (output)/TCLKA (input)

Figure 8.10 Port A Pin Configuration

H'FFFEE9	Port A data direction register	PADDR	W	H'00		H'80
H'FFFD9	Port A data register	PADR	R/W	H'00		H'00

Note: * Lower 20 bits of the address in advanced mode.

Port A Data Direction Register (PADDR): PADDR is an 8-bit write-only register th input or output for each pin in port A. When pins are used for TPC output, the corresp PADDR bits must also be set.

Bit		7	6	5	4	3	2	1
		PA_7DDR	PA_6DDR	PA_5DDR	PA_4DDR	PA_3DDR	PA_2DDR	PA_1DD
Modes 3 and 4	Initial value	1	0	0	0	0	0	0
	Read/Write	—	W	W	W	W	W	W
Modes 1, 2, 5, 6, and 7	Initial value	0	0	0	0	0	0	0
	Read/Write	W	W	W	W	W	W	W

Port A data direction 7 to 0
These bits select input or output for port A

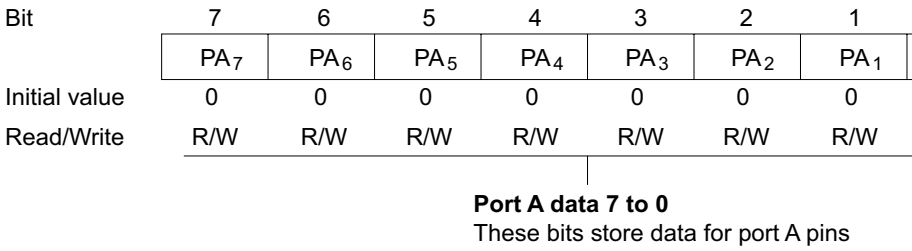
The pin functions that can be selected for pins PA_7 to PA_4 differ between modes 1, 2, modes 3 to 5. For the method of selecting the pin functions, see tables 8.19 and 8.20.

The pin functions that can be selected for pins PA_3 to PA_0 are the same in modes 1 to method of selecting the pin functions, see table 8.21.

When port A functions as an input/output port, a pin in port A becomes an output port corresponding PADDR bit is set to 1, and an input port if this bit is cleared to 0. In mo PA_7DDR is fixed at 1 and PA_7 functions as the A_{20} address output pin.

PADDR is a write-only register. Its value cannot be read. All bits return 1 when read.

When a bit in PADDR is cleared to 0, if port A is read the corresponding pin logic leve



Bit	7	6	5	4	3	2	1
	PA_7	PA_6	PA_5	PA_4	PA_3	PA_2	PA_1
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

Port A data 7 to 0
These bits store data for port A pins

PADR is initialized to H'00 by a reset and in hardware standby mode. In software stand
retains its previous setting.

Pin function	$TIOCB_2$ output			PA_7 input	PA outp
				$TIOCB_2$	

Note: * $TIOCB_2$ input when IOB2 = 1 and PWM2 = 0.

16-bit timer channel 2 settings	(2)	(1)		(2)
IOB2	0			1
IOB1	0	0	1	—
IOB0	0	1	—	—

$PA_6/TP_6/TIOCA_2$

Bit PWM2 in TMDR, bits IOA2 to IOA0 in TIOR2, bit NDER6 in NDE
PA_6DDR select the pin function as follows.

16-bit timer channel 2 settings	(1) in table below			(2) in table
PA_6DDR	—			0	1
NDER6	—			—	0
Pin function	$TIOCA_2$ output			PA_6 input	PA outp
				$TIOCA_2$	

Note: * $TIOCA_2$ input when IOA2 = 1.

16-bit timer channel 2 settings	(2)	(1)		(2)	
PWM2	0				
IOA2	0			1	
IOA1	0	0	1	—	
IOA0	0	1	—	—	

			input	outpu
			TIOCB$_1$ in	

Note: * TIOCB$_1$ input when IOB2 = 1 and PWM1 = 0.

16-bit timer channel 1 settings	(2)	(1)		
IOB2	0			
IOB1	0	0	1	
IOB0	0	1	—	

PA$_4$/TP$_4$/ TIOCA$_1$

Bit PWM1 in TMDR, bits IOA2 to IOA0 in TIOR1, bit NDER4 in NDER PA$_4$DDR select the pin function as follows.

16-bit timer channel 1 settings	(1) in table below		(2) in table	
PA$_4$DDR	—		0	1
NDER4	—		—	0
Pin function	TIOCA$_1$ output		PA$_4$ input	PA$_4$ outpu
			TIOCA$_1$ in	

Note: * TIOCA$_1$ input when IOA2 = 1.

16-bit timer channel 1 settings	(2)	(1)		(2)	
PWM1	0				
IOA2	0			1	
IOA1	0	0	1	—	
IOA0	0	1	—	—	

A20E	1			
16-bit timer channel 2 settings	(1) in table below	(2) in table below		
PA_7DDR	—	0	1	1
NDER7	—	—	0	1
Pin function	$TIOCB_2$ output	PA_7 input	PA_7 output	TP outp
		$TIOCB_2$ input*		

Note: * $TIOCB_2$ input when IOB2 = 1 and PWM2 = 0.

16-bit timer channel 2 settings	(2)	(1)		
IOB2	0			
IOB1	0	0	1	
IOB0	0	1	—	

Pin function	$TIOCA_2$ output	PA_6 input	PA_6 output	TP_6 outpu
		$TIOCA_2$ input*		

Note: * $TIOCA_2$ input when IOA2 = 1.

16-bit timer channel 2 settings	(2)	(1)		(2)
PWM2	0			
IOA2	0			1
IOA1	0	0	1	—
IOA0	0	1	—	—

$PA_5/TP_5/TIOCB_1/A_{22}$

Bit PWM1 in TMDR, bits IOB2 to IOB0 in TIOR1, bit NDER5 in NDER in BRCR, and bit PA_5DDR select the pin function as follows.

A22E	1			
16-bit timer channel 1 settings	(1) in table below	(2) in table below		
PA_5DDR	—	0	1	1
NDER5	—	—	0	1
Pin function	$TIOCB_1$ output	PA_5 input	PA_5 output	TP_5 outpu
		$TIOCB_1$ input*		

Note: * $TIOCB_1$ input when IOB2 = 1 and PWM1 = 0.

16-bit timer channel 1 settings	(2)	(1)		
IOB2	0			
IOB1	0	0	1	
IOB0	0	1	—	

Pin function	$TIOCA_1$ output	PA_4 input	PA_4 output	TP
		$TIOCA_1$ input*		

Note: * $TIOCA_1$ input when IOA2 = 1.

16-bit timer channel 1 settings	(2)	(1)		(2)	
PWM1	0				
IOA2	0			1	
IOA1	0	0	1	—	
IOA0	0	1	—	—	

NDER3	—	—	0
Pin function	$TIOCB_0$ output	PA_3 input	PA_3 output
		$TIOCB_0$ input	
	TCLKD input*2		

Notes: 1. $TIOCB_0$ input when IOB2 = 1 and PWM0 = 0.

2. TCLKD input when TPSC2 = TPSC1 = TPSC0 = 1 in any
to 16TCR0, or bits CKS2 to CKS0 in 8TCR2 are as shown
table below.

16-bit timer channel 0 settings	(2)	(1)		
IOB2	0			
IOB1	0	0	1	
IOB0	0	1	—	

8-bit timer channel 2 settings	(4)	(3)		
CKS2	0	1		
CKS1	—	0		
CKS0	—	0	1	-

Pin function	$TIOCA_0$ output	TPA_2 input	TPA_2 output
		$TIOCA_0$ inpu	
	TCLKC input*2		

Notes: 1. $TIOCA_0$ input when IOA2 = 1.

2. TCLKC input when TPSC2 = TPSC1 = 1 and TPSC0 = 0
16TCR2 to 16TCR0, or bits CKS2 to CKS0 in 8TCR0 are
(3) in the table below.

16-bit timer channel 0 settings	(2)	(1)		(2)	
PWM0	0				
IOA2	0			1	
IOA1	0	0	1	—	
IOA0	0	1	—	—	

8-bit timer channel 0 settings	(4)		(3)	
CKS2	0	1		
CKS1	—	0		
CKS0	—	0	1	

	$\overline{TEND}_1$ output*2	

Notes: 1. TCLKB input when MDF = 1 in TMDR, or TPSC2 = 1, TPS
TPSC0 = 1 in any of 16TCR2 to 16TCR0, or bits CKS2 to
8TCR3 are as shown in (1) in the table below.

2. When an external request is specified as a DMAC activati
$\overline{TEND}_1$ output regardless of bits PA_1DDR and NDER1.

8-bit timer channel 3 settings	(2)		(1)	
CKS2	0	1		
CKS1	—	0		
CKS0	—	0	1	

PA_0/TP_0/ TCLKA/ $\overline{TEND}_0$

Bit MDF in TMDR, bits TPSC2 to TPSC0 in 16TCR2 to 16TCR0 of the timer, bits CKS2 to CKS0 in 8TCR1 of the 8-bit timer, bit NDER0 in NDER and bit PA_0DDR select the pin function as follows.

PA_0DDR	0	1	
NDER0	—	0	
Pin function	PA_0 input	PA_0 output	TP
	TCLKA output*1		
	$\overline{TEND}_0$ output*2		

Notes: 1. TCLKA input when MDF = 1 in TMDR, or TPSC2 = 1, TPS
TPSC0 = 0 in any of 16TCR2 to 16TCR0, or bits CKS2 to
8TCR0 are as shown in (1) in the table below.

2. When an external request is specified as a DMAC activati
$\overline{TEND}_0$ output regardless of bits PA_0DDR and NDER0.

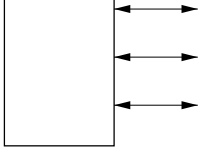
8-bit timer channel 1 settings	(2)		(1)	
CKS2	0	1		
CKS1	—	0		
CKS0	—	0	1	

A reset or hardware standby transition leaves port B as an input port.

For output of $\overline{CS}_7$ to $\overline{CS}_4$ in modes 1 to 5, see section 6.3.4, Chip Select Signals. Wher
connected to areas 2, 3, 4, and 5, the $\overline{CS}_4$ and $\overline{CS}_5$ output pins become $\overline{RAS}$ output pin
areas. For details see section 6.5, DRAM Interface. Pins not assigned to any of these
available for generic input/output. Figure 8.11 shows the pin configuration of port B.

When DRAM is connected to areas 2, 3, 4, and 5, the $\overline{CS}_4$ and $\overline{CS}_5$ output pins becom
output pins for these areas. For details see section 6.5, DRAM Interface.

Pins in port B can drive one TTL load and a 30-pF capacitive load. They can also driv
transistor pair.



Pin functions in modes 1 to 5

PB_7 (input/output)/TP_{15} (output) /RxD_2 (input)

PB_6 (input/output)/TP_{14} (output) /TxD_2 (output)

PB_5 (input/output)/TP_{13} (output) /SCK_2 (input/output) /$\overline{LCAS}$ (output)

PB_4 (input/output)/TP_{12} (output) /$\overline{UCAS}$ (output)

PB_3 (input/output)/TP_{11} (output) /$TMIO_3$ (input/output) /$\overline{DREQ}_1$ (input) /

PB_2 (input/output)/TP_{10} (output) /TMO_2 (output) /$\overline{CS}_5$ (output)

PB_1 (input/output)/TP_9 (output) /$TMIO_1$ (input/output) /$\overline{DREQ}_0$ (input) /

PB_0 (input/output)/TP_8 (output) /TMO_0 (output) /$\overline{CS}_7$ (output)

Pin functions in modes 6 and 7

PB_7 (input/output)/TP_{15} (output) /RxD_2 (input)

PB_6 (input/output)/TP_{14} (output) /TxD_2 (output)

PB_5 (input/output)/TP_{13} (output) /SCK_2 (input/output)

PB_4 (input/output)/TP_{12} (output)

PB_3 (input/output)/TP_{11} (output) /$TMIO_3$ (input/output) /$\overline{DREQ}_1$ (input)

PB_2 (input/output)/TP_{10} (output) /TMO_2 (output)

PB_1 (input/output)/TP_9 (output) /$TMIO_1$ (input/output) /$\overline{DREQ}_0$ (input)

PB_0 (input/output)/TP_8 (output) /TMO_0 (output)

Figure 8.11 Port B Pin Configuration

Note: * Lower 20 bits of the address in advanced mode.

Port B Data Direction Register (PBDDR): PBDDR is an 8-bit write-only register th
input or output for each pin in port B. When pins are used for TPC output, the corresp
PBDDR bits must also be set.

Bit	7	6	5	4	3	2	1
	PB_7DDR	PB_6DDR	PB_5DDR	PB_4DDR	PB_3DDR	PB_2DDR	PB_1DDR
Initial value	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W

Port B data direction 7 to 0
These bits select input or output for port B p

The pin functions that can be selected for port B differ between modes 1 to 5, and mo
For the method of selecting the pin functions, see tables 8.23 and 8.24.

When port B functions as an input/output port, a pin in port B becomes an output port
corresponding PBDDR bit is set to 1, and an input port if this bit is cleared to 0.

PBDDR is a write-only register. Its value cannot be read. All bits return 1 when read.

PBDDR is initialized to H'00 by a reset and in hardware standby mode. In software st
it retains its previous setting. Therefore, if a transition is made to software standby m
port B is functioning as an input/output port and a PBDDR bit is set to 1, the correspo
maintains its output state.

Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Port B data 7 to 0
These bits store data for port B pins

PBDR is initialized to H'00 by a reset and in hardware standby mode. In software stand
retains its previous setting.

Pin function	PB_7 input	PB_7 output	TP_{15} output	RxD_2 input

$PB_6/TP_{14}/TxD_2$

Bit TE in SCR of SCI2, bit SMIF in SCMR, bit NDER14 in NDERB, a PB_6DDR select the pin function as follows.

SMIF	0			
TE	0			1
PB_6DDR	0	1	1	—
NDER14	—	0	1	—
Pin function	PB_6 input	PB_6 output	TP_{14} output	TxD_2 output

Note: * Functions as the TxD_2 output pin, but there are two states: o the pin is driven, and another in which the pin is at high-imp

$PB_5/TP_{13}/SCK_2/\overline{LCAS}$

Bit C/$\overline{A}$ in SMR of SCI2, bits CKE0 and CKE1 in SCR, bit NDER13 in and bit PB_5DDR select the pin function as follows.

CKE1	0				
C/$\overline{A}$	0				1
CKE0	0			1	—
PB_5DDR	0	1	1	—	—
NDER13	—	0	1	—	—
Pin function	PB_5 input	PB_5 output	TP_{13} output	SCK_2 output	SCK output
	$\overline{LCAS}$ output*				

Note: * $\overline{LCAS}$ output depending on bits DRAS2 to DRAS0 in DRCR CSEL in DRCRB, and regardless of bits C/$\overline{A}$, CKE0 and CK and PB_5DDR. For details, see section 6, Bus Controller.

details, see section 6, Bus Controller.

$PB_3/TP_{11}/TMIO_3/\overline{DREQ}_1/\overline{CS}_4$

The DRAM interface settings by bits DRAS2 to DRAS0 in DRCRA, bit and OS1/0 in 8TCSR3, bits CCLR1 and CCLR0 in 8TCR3, bit CS4E i NDER11 in NDERB, and bit PB_3DDR select the pin function as follows

DRAM interface settings	(1) in table below				
OIS3/2 and OS1/0	All 0				Not all
CS4E	0			1	—
PB_3DDR	0	1	1	—	—
NDER11	—	0	1	—	—
Pin function	PB_3 input	PB_3 output	TP_{11} output	$\overline{CS}_4$ output	$TMIO_3$ output
	$TMIO_3$ input*1				
	$\overline{DREQ}_1$ input*2				

Notes: 1. $TMIO_3$ input when CCLR1 = CCLR0 = 1.

2. When an external request is specified as a DMAC activatio $\overline{DREQ}_1$ input regardless of bits OIS3 and OIS2, OS1 and O and CCLR0, CS4E, NDER11, and PB_3DDR.

3. $\overline{CS}_4$ is output as $\overline{RAS}_4$.

DRAM interface settings	(1)				(2)		
DRAS2	0				1		
DRAS1	0		1		0		
DRAS0	0	1	0	1	0	1	

RENESAS

CS5E	0			1	—
PB_2DDR	0	1	1	—	—
NDER10	—	0	1	—	—
Pin function	PB_2 input	PB_2 output	TP_{10} output	$\overline{CS}_5$ output	TMIO outpu

Note: * $\overline{CS}_5$ is output as $\overline{RAS}_5$.

DRAM interface settings	(1)						(2)	
DRAS2	0				1			
DRAS1	0		1		0			
DRAS0	0	1	0	1	0	1		

$PB_1/TP_9/TMIO_1/\overline{DREQ}_0/\overline{CS}_6$

Bits OIS3/2 and OS1/0 in 8TCSR1, bits CCLR1 and CCLR0 in TCR1 CSCR, bit NDER9 in NDERB, and bit PB_1DDR select the pin functio

OIS3/2 and OS1/0	All 0			
CS6E	0			1
PB_1DDR	0	1	1	—
NDER9	—	0	1	—
Pin function	PB_1 input	PB_1 output	TP_9 output	$\overline{CS}_6$ output
	$TMIO_1$ input*1			
	$\overline{DREQ}_0$ input*2			

Notes: 1. $TMIO_1$ input when CCLR1 = CCLR0 = 1.

2. When an external request is specified as a DMAC activat $\overline{DREQ}_0$ input regardless of bits OIS3/2 and OS1/0, bits C CS6E, bit NDER9, and bit PB_1DDR.

Pin function	PB_0 input	PB_0 output	TP_8 output	CS_7 output

Pin function	PB_7 input	PB_7 output	TP_{15} output	RxD_2 input

$PB_6/TP_{14}/TxD_2$

Bit TE in SCR of SCI2, bit SMIF in SCMR, bit NDER14 in NDERB, a PB_6DDR select the pin function as follows.

SMIF	0			
TE	0			1
PB_6DDR	0	1	1	—
NDER14	—	0	1	—
Pin function	PB_6 input	PB_6 output	TP_{14} output	TxD_2 outpu

Note: * Functions as the TxD2 output pin, but there are two states: the pin is driven, and another in which the pin is at high-imp

$PB_5/TP_{13}/SCK_2$

Bit C/$\overline{A}$ in SMR of SCI2, bits CKE0 and CKE1 in SCR, bit NDER13 i and bit PB_5DDR select the pin function as follows.

CKE1	0				
C/$\overline{A}$	0				1
CKE0	0			1	—
PB_5DDR	0	1	1	—	—
NDER13	—	0	1	—	—
Pin function	PB_5 input	PB_5 output	TP_{13} output	SCK_2 output	SCK outpu

PB_4/TP_{12}

Bit NDER12 in NDERB and bit PB_4DDR select the pin function as fo

PB_4DDR	0	1	
NDER12	—	0	
Pin function	PB_4 input	PB_4 output	TF

		$TMIO_3$ input*1		
		$\overline{DREQ}_1$ input*2		

Notes: 1. $TMIO_3$ input when CCLR1 = CCLR0 = 1.

2. When an external request is specified as a DMAC activati
$\overline{DREQ}_1$ input regardless of bits OIS3/2 and OS1/0, bit NDE
bit PB_3DDR.

$PB_2/TP_{10}/TMO_2$

Bits OIS3/2 and OS1/0 in TCSR2, bit NDER10 in NDERB, and bit PB_2
the pin function as follows.

OIS3/2 and OS1/0	All 0			
PB_2DDR	0	1	1	
NDER10	—	0	1	
Pin function	PB_2 input	PB_2 output	TP_{10} output	T

$PB_1/TP_9/TMIO_1/\overline{DREQ}_0$

Bits OIS3/2 and OS1/0 in TCSR1, bits CCLR1 and CCLR0 in TCR1, b
NDERB, and bit PB_1DDR select the pin function as follows.

OIS3/2 and OS1/0	All 0			
PB_1DDR	0	1	1	
NDER9	—	0	1	
Pin function	PB_1 input	PB_1 output	TP_9 output	
	$TMIO_1$ input*1			
	$\overline{DREQ}_0$ input*2			

Notes: 1. $TMIO_1$ input when CCLR1 = CCLR0 = 1.

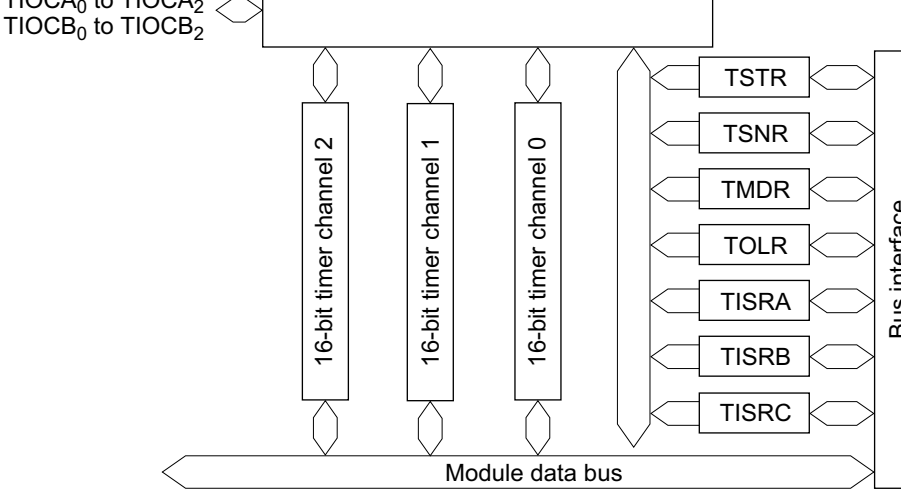
2. When an external request is specified as a DMAC activati
$\overline{DREQ}_0$ input regardless of bits OIS3/2 and OS1/0, bit NDE
PB_1DDR.

	Input	Output	Output	

16-bit timer features are listed below.

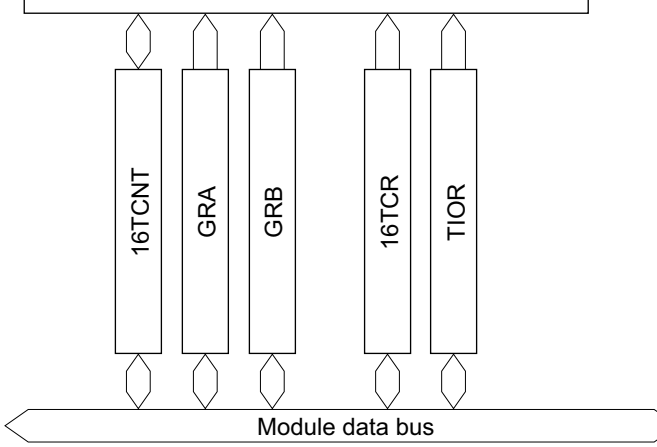
- Capability to process up to 6 pulse outputs or 6 pulse inputs
- Six general registers (GRs, two per channel) with independently-assignable output input capture functions
- Selection of eight counter clock sources for each channel:
  Internal clocks: ϕ, ϕ/2, ϕ/4, ϕ/8
  External clocks: TCLKA, TCLKB, TCLKC, TCLKD
- Five operating modes selectable in all channels:
  — Waveform output by compare match
    Selection of 0 output, 1 output, or toggle output (only 0 or 1 output in channel
  — Input capture function
    Rising edge, falling edge, or both edges (selectable)
  — Counter clearing function
    Counters can be cleared by compare match or input capture
  — Synchronization
    Two or more timer counters (16TCNTs) can be preset simultaneously, or clear simultaneously by compare match or input capture. Counter synchronization e synchronous register input and output.
  — PWM mode
    PWM output can be provided with an arbitrary duty cycle. With synchronizati three-phase PWM output is possible
- Phase counting mode selectable in channel 2
  Two-phase encoder output can be counted automatically.
- High-speed access via internal 16-bit bus
  The 16TCNTs and GRs can be accessed at high speed via a 16-bit bus.
- Any initial timer output value can be set
- Nine interrupt sources
  Each channel has two compare match/input capture interrupts and an overflow int interrupts can be requested independently.

		External clocks: TCLKA, TCLKB, TCLKC, TCLKD, selectable independently		
General registers (output compare/input capture registers)		GRA0, GRB0	GRA1, GRB1	GRA2, GRI
Input/output pins		$TIOCA_0$, $TIOCB_0$	$TIOCA_1$, $TIOCB_1$	$TIOCA_2$, TI
Counter clearing function		GRA0/GRB0 compare match or input capture	GRA1/GRB1 compare match or input capture	GRA2/GRE match or in
Initial output value setting function		Available	Available	Available
Compare match output	0	Available	Available	Available
	1	Available	Available	Available
	Toggle	Available	Available	Not availab
Input capture function		Available	Available	Available
Synchronization		Available	Available	Available
PWM mode		Available	Available	Available
Phase counting mode		Not available	Not available	Available
Interrupt sources		Three sources • Compare match/input capture A0 • Compare match/input capture B0 • Overflow	Three sources • Compare match/input capture A1 • Compare match/input capture B1 • Overflow	Three sour • Compare capture A • Compare capture B • Overflow



Legend:
TSTR: Timer start register (8 bits)
TSNR: Timer synchro register (8 bits)
TMDR: Timer mode register (8 bits)
TOLR: Timer output level setting register (8 bits)
TISRA: Timer interrupt status register A (8 bits)
TISRB: Timer interrupt status register B (8 bits)
TISRC: Timer interrupt status register C (8 bits)

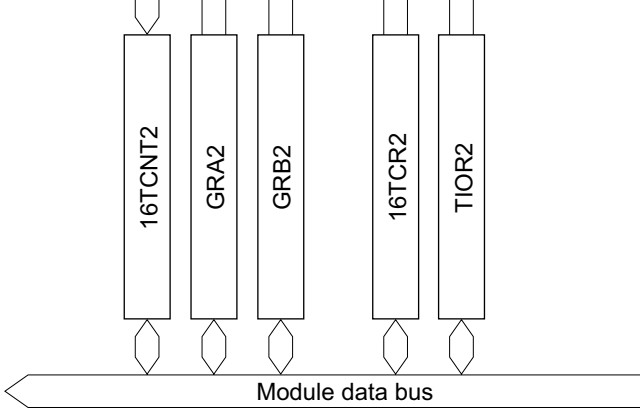
Figure 9.1 16-bit timer Block Diagram (Overall)



Legend:

16TCNT:	Timer counter (16 bits)
GRA, GRB:	General registers A and B (input capture/output compare registers) (16
TCR:	Timer control register (8 bits)
TIOR:	Timer I/O control register (8 bits)

Figure 9.2 Block Diagram of Channels 0 and 1



Legend:

16TCNT2:	Timer counter 2 (16 bits)
GRA2, GRB2:	General registers A2 and B2 (input capture/output compare register (16 bits × 2)
TCR2:	Timer control register 2 (8 bits)
TIOR2:	Timer I/O control register 2 (8 bits)

Figure 9.3 Block Diagram of Channel 2

	Clock input B	TCLKB	Input	External clock B input pin (phase-B input pin in phase cou
	Clock input C	TCLKC	Input	External clock C input pin
	Clock input D	TCLKD	Input	External clock D input pin
0	Input capture/output compare A0	$TIOCA_0$	Input/ output	GRA0 output compare or input PWM output pin in PWM mode
	Input capture/output compare B0	$TIOCB_0$	Input/ output	GRB0 output compare or input
1	Input capture/output compare A1	$TIOCA_1$	Input/ output	GRA1 output compare or input PWM output pin in PWM mode
	Input capture/output compare B1	$TIOCB_1$	Input/ output	GRB1 output compare or input
2	Input capture/output compare A2	$TIOCA_2$	Input/ output	GRA2 output compare or input PWM output pin in PWM mode
	Input capture/output compare B2	$TIOCB_2$	Input/ output	GRB2 output compare or input

	H'FFF61	Timer synchro register	TSNC	R/W
	H'FFF62	Timer mode register	TMDR	R/W
	H'FFF63	Timer output level setting register	TOLR	W
	H'FFF64	Timer interrupt status register A	TISRA	R/(W)*2
	H'FFF65	Timer interrupt status register B	TISRB	R/(W)*2
	H'FFF66	Timer interrupt status register C	TISRC	R/(W)*2
0	H'FFF68	Timer control register 0	16TCR0	R/W
	H'FFF69	Timer I/O control register 0	TIOR0	R/W
	H'FFF6A	Timer counter 0H	16TCNT0H	R/W
	H'FFF6B	Timer counter 0L	16TCNT0L	R/W
	H'FFF6C	General register A0H	GRA0H	R/W
	H'FFF6D	General register A0L	GRA0L	R/W
	H'FFF6E	General register B0H	GRB0H	R/W
	H'FFF6F	General register B0L	GRB0L	R/W
1	H'FFF70	Timer control register 1	16TCR1	R/W
	H'FFF71	Timer I/O control register 1	TIOR1	R/W
	H'FFF72	Timer counter 1H	16TCNT1H	R/W
	H'FFF73	Timer counter 1L	16TCNT1L	R/W
	H'FFF74	General register A1H	GRA1H	R/W
	H'FFF75	General register A1L	GRA1L	R/W
	H'FFF76	General register B1H	GRB1H	R/W
	H'FFF77	General register B1L	GRB1L	R/W

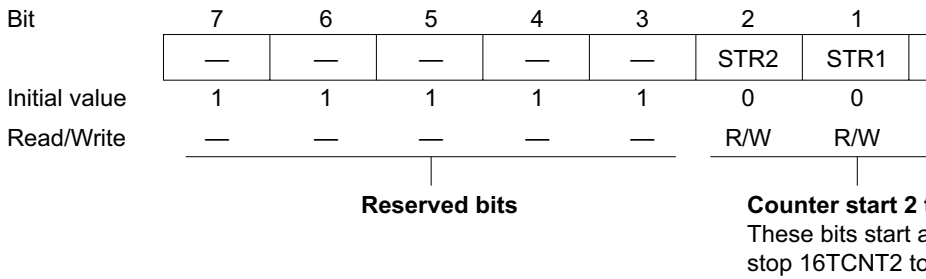
H'FFF7E	General register B2H	GRB2H	R/W
H'FFF7F	General register B2L	GRB2L	R/W

Notes: 1. The lower 20 bits of the address in advanced mode are indicated.
2. Only 0 can be written in bits 3 to 0, to clear the flags.

9.2 Register Descriptions

9.2.1 Timer Start Register (TSTR)

TSTR is an 8-bit readable/writable register that starts and stops the timer counter (16TC
channels 0 to 2.



TSTR is initialized to H'F8 by a reset and in standby mode.

Bits 7 to 3—Reserved: These bits cannot be modified and are always read as 1.

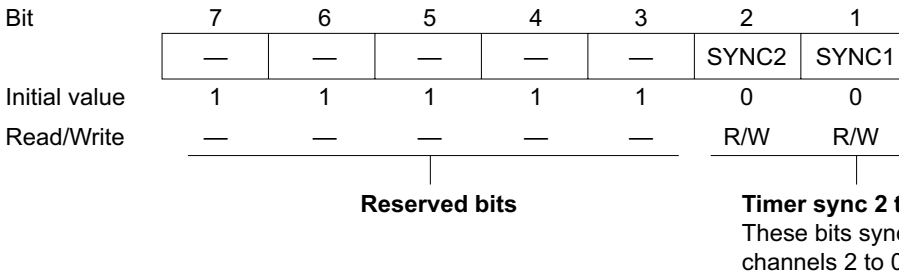
Bit 2—Counter Start 2 (STR2): Starts and stops timer counter 2 (16TCNT2).

Bit 2 STR2	Description	
0	16TCNT2 is halted	(In
1	16TCNT2 is counting	

Bit 0 STR0	Description	
0	16TCNT0 is halted	(I
1	16TCNT0 is counting	

9.2.2 Timer Synchro Register (TSNC)

TSNC is an 8-bit readable/writable register that selects whether channels 0 to 2 operat independently or synchronously. Channels are synchronized by setting the correspond



TSNC is initialized to H'F8 by a reset and in standby mode.

Bits 7 to 3—Reserved: These bits cannot be modified and are always read as 1.

Bit 2—Timer Sync 2 (SYNC2): Selects whether channel 2 operates independently or synchronously.

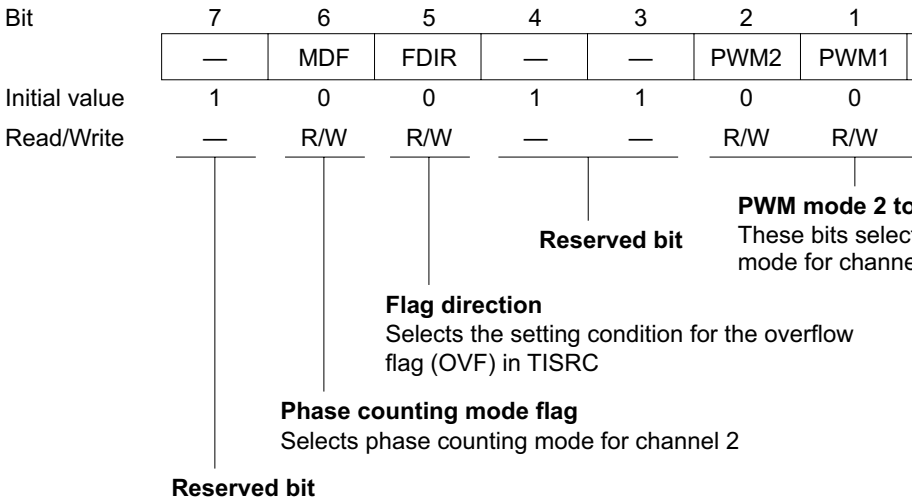
Bit 2 SYNC2	Description	
0	Channel 2's timer counter (16TCNT2) operates independently 16TCNT2 is preset and cleared independently of other channels	(I
1	Channel 2 operates synchronously 16TCNT2 can be synchronously preset and cleared	

Bit 0—Timer Sync 0 (SYNC0): Selects whether channel 0 operates independently or synchronously.

Bit 0 SYNC0	Description	
0	Channel 0's timer counter (16TCNT0) operates independently 16TCNT0 is preset and cleared independently of other channels	(In
1	Channel 0 operates synchronously 16TCNT0 can be synchronously preset and cleared	

9.2.3 Timer Mode Register (TMDR)

TMDR is an 8-bit readable/writable register that selects PWM mode for channels 0 to 2 selects phase counting mode and the overflow flag (OVF) setting conditions for channe



TMDR is initialized to H'98 by a reset and in standby mode.

When MDF is set to 1 to select phase counting mode, 16TCNT2 operates as an up/do and pins TCLKA and TCLKB become counter clock input pins. 16TCNT2 counts bot falling edges of TCLKA and TCLKB, and counts up or down as follows.

Counting Direction	Down-Counting				Up-Counting			
TCLKA pin	↑	High	↓	Low	Low	↑	High	
TCLKB pin	Low	↑	High	↓	↑	High	↓	

In phase counting mode, external clock edge selection by bits CKEG1 and CKEG0 in and counter clock selection by bits TPSC2 to TPSC0 are invalid, and the above phase mode operations take precedence.

The counter clearing condition selected by the CCLR1 and CCLR0 bits in 16TCR2 ar compare match/input capture settings and interrupt functions of TIOR2, TISRA, TISR remain effective in phase counting mode.

Bit 5—Flag Direction (FDIR): Designates the setting condition for the OVF flag in T FDIR designation is valid in all modes in channel 2.

Bit 5 FDIR	Description
0	OVF is set to 1 in TISRC when 16TCNT2 overflows or underflows (I
1	OVF is set to 1 in TISRC when 16TCNT2 overflows

Bits 4 and 3—Reserved: These bits cannot be modified and are always read as 1.

output goes to 1 at compare match with GRA2, and to 0 at compare match with GRB2.

Bit 1—PWM Mode 1 (PWM1): Selects whether channel 1 operates normally or in PW

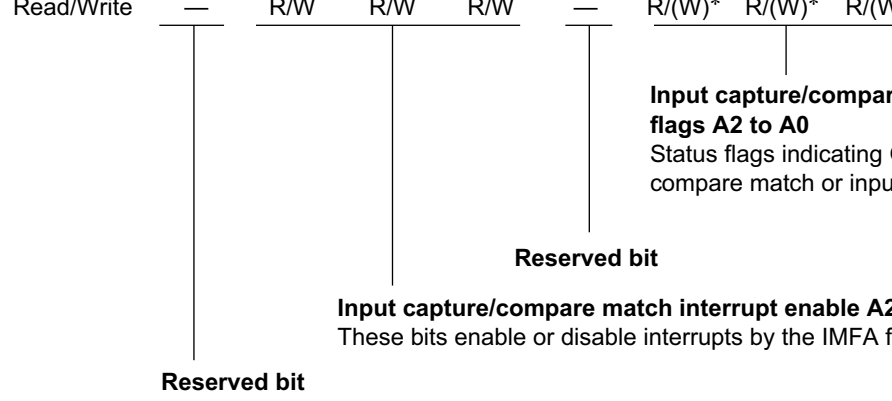
Bit 1 PWM1	Description	
0	Channel 1 operates normally	(In
1	Channel 1 operates in PWM mode	

When bit PWM1 is set to 1 to select PWM mode, pin $TIOCA_1$ becomes a PWM output output goes to 1 at compare match with GRA1, and to 0 at compare match with GRB1.

Bit 0—PWM Mode 0 (PWM0): Selects whether channel 0 operates normally or in PW

Bit 0 PWM0	Description	
0	Channel 0 operates normally	(In
1	Channel 0 operates in PWM mode	

When bit PWM0 is set to 1 to select PWM mode, pin $TIOCA_0$ becomes a PWM output output goes to 1 at compare match with GRA0, and to 0 at compare match with GRB0.



Note: * Only 0 can be written, to clear the flag.

TISRA is initialized to H'88 by a reset and in standby mode.

Bit 7—Reserved: This bit cannot be modified and is always read as 1.

Bit 6—Input Capture/Compare Match Interrupt Enable A2 (IMIEA2): Enables c
the interrupt requested by the IMFA2 when IMFA2 flag is set to 1.

Bit 6 IMIEA2	Description	
0	IMIA2 interrupt requested by IMFA2 flag is disabled	(I
1	IMIA2 interrupt requested by IMFA2 flag is enabled	

Bit 4—Input Capture/Compare Match Interrupt Enable A0 (IMIEA0): Enables or
the interrupt requested by the IMFA0 flag when IMFA0 is set to 1.

Bit 4 IMIEA0	Description	
0	IMIA0 interrupt requested by IMFA0 flag is disabled	(I
1	IMIA0 interrupt requested by IMFA0 flag is enabled	

Bit 3—Reserved: This bit cannot be modified and is always read as 1.

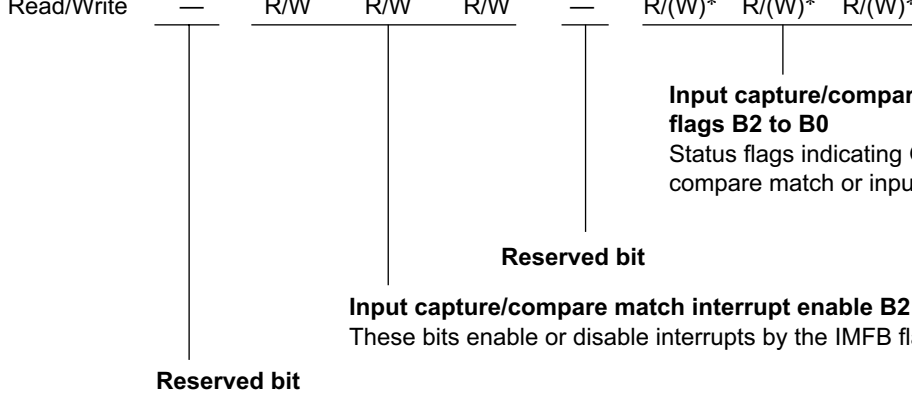
Bit 2—Input Capture/Compare Match Flag A2 (IMFA2): This status flag indicates
compare match or input capture events.

Bit 2 IMFA2	Description	
0	[Clearing conditions] • Read IMFA2 flag when IMFA2 =1, then write 0 in IMFA2 flag • DMAC is activated by an IMIA2 interrupt	(In
1	[Setting conditions] • 16TCNT2 = GRA2 when GRA2 functions as an output compare register • 16TCNT2 value is transferred to GRA2 by an input capture signal when G functions as an input capture register	

	• 16TCNT1 = GRA1 when GRA1 functions as an output compare register • 16TCNT1 value is transferred to GRA1 by an input capture signal when functions as an input capture register

Bit 0—Input Capture/Compare Match Flag A0 (IMFA0): This status flag indicate compare match or input capture events.

Bit 0 IMFA0	Description
0	[Clearing conditions] (I • Read IMFA0 flag when IMFA0 =1, then write 0 in IMFA0 flag • DMAC is activated by an IMIA0 interrupt
1	[Setting conditions] • 16TCNT0 = GRA0 when GRA0 functions as an output compare register • 16TCNT0 value is transferred to GRA0 by an input capture signal when functions as an input capture register



Note: * Only 0 can be written, to clear the flag.

TISRB is initialized to H'88 by a reset and in standby mode.

Bit 7—Reserved: This bit cannot be modified and is always read as 1.

Bit 6—Input Capture/Compare Match Interrupt Enable B2 (IMIEB2): Enables or the interrupt requested by the IMFB2 when IMFB2 flag is set to 1.

Bit 6 IMIEB2	Description	
0	IMIB2 interrupt requested by IMFB2 flag is disabled	(In
1	IMIB2 interrupt requested by IMFB2 flag is enabled	

Bit 4—Input Capture/Compare Match Interrupt Enable B0 (IMIEB0): Enables o
the interrupt requested by the IMFB0 when IMFB0 flag is set to 1.

Bit 4 IMIEB0	Description
0	IMIB0 interrupt requested by IMFB0 flag is disabled
1	IMIB0 interrupt requested by IMFB0 flag is enabled

Bit 3—Reserved: This bit cannot be modified and is always read as 1.

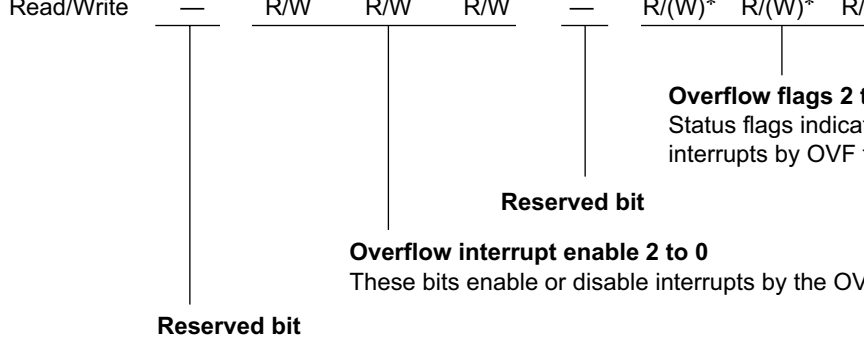
Bit 2—Input Capture/Compare Match Flag B2 (IMFB2): This status flag indicates
compare match or input capture events.

Bit 2 IMFB2	Description
0	[Clearing condition] Read IMFB2 flag when IMFB2 =1, then write 0 in IMFB2 flag
1	[Setting conditions] • 16TCNT2 = GRB2 when GRB2 functions as an output compare register • 16TCNT2 value is transferred to GRB2 by an input capture signal when functions as an input capture register

	• 16TCNT1 value is transferred to GRB1 by an input capture signal when G functions as an input capture register

Bit 0—Input Capture/Compare Match Flag B0 (IMFB0): This status flag indicates compare match or input capture events.

Bit 0 IMFB0	Description
0	[Clearing condition] (In Read IMFB0 flag when IMFB0 =1, then write 0 in IMFB0 flag
1	[Setting conditions] • 16TCNT0 = GRB0 when GRB0 functions as an output compare register • 16TCNT0 value is transferred to GRB0 by an input capture signal when G functions as an input capture register



Note: * Only 0 can be written, to clear the flag.

TISRC is initialized to H'88 by a reset and in standby mode.

Bit 7—Reserved: This bit cannot be modified and is always read as 1.

Bit 6—Overflow Interrupt Enable 2 (OVIE2): Enables or disables the interrupt req
OVF2 when OVF2 flag is set to 1.

Bit 6 OVIE2	Description	
0	OVI2 interrupt requested by OVF2 flag is disabled	(I
1	OVI2 interrupt requested by OVF2 flag is enabled	

Bit 5—Overflow Interrupt Enable 1 (OVIE1): Enables or disables the interrupt req
OVF1 when OVF1 flag is set to 1.

Bit 5 OVIE1	Description	
0	OVI1 interrupt requested by OVF1 flag is disabled	(I
1	OVI1 interrupt requested by OVF1 flag is enabled	

Bit 3—Reserved: This bit cannot be modified and is always read as 1.

Bit 2—Overflow Flag 2 (OVF2): This status flag indicates 16TCNT2 overflow.

Bit 2 OVF2	Description
0	[Clearing condition] (In Read OVF2 flag when OVF2 =1, then write 0 in OVF2 flag
1	[Setting condition] 16TCNT2 overflowed from H'FFFF to H'0000, or underflowed from H'0000 to

Note: 16TCNT underflow occurs when 16TCNT operates as an up/down-counter. Und occurs only when channel 2 operates in phase counting mode (MDF = 1 in TMD

Bit 1—Overflow Flag 1 (OVF1): This status flag indicates 16TCNT1 overflow.

Bit 1 OVF1	Description
0	[Clearing condition] (In Read OVF1 flag when OVF1 =1, then write 0 in OVF1 flag
1	[Setting condition] 16TCNT1 overflowed from H'FFFF to H'0000

Bit 0—Overflow Flag 0 (OVF0): This status flag indicates 16TCNT0 overflow.

Bit 0 OVF0	Description
0	[Clearing condition] (In Read OVF0 flag when OVF0 =1, then write 0 in OVF0 flag
1	[Setting condition] 16TCNT0 overflowed from H'FFFF to H'0000

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
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

Each 16TCNT is a 16-bit readable/writable register that counts pulse inputs from a clo
The clock source is selected by bits TPSC2 to TPSC0 in 16TCR.

16TCNT0 and 16TCNT1 are up-counters. 16TCNT2 is an up/down-counter in phase c
mode and an up-counter in other modes.

16TCNT can be cleared to H'0000 by compare match with GRA or GRB or by input c
GRA or GRB (counter clearing function).

When 16TCNT overflows (changes from H'FFFF to H'0000), the OVF flag is set to 1
the corresponding channel.

When 16TCNT underflows (changes from H'0000 to H'FFFF), the OVF flag is set to
of the corresponding channel.

The 16TCNTs are linked to the CPU by an internal 16-bit bus and can be written or re
word access or byte access.

Each 16TCNT is initialized to H'0000 by a reset and in standby mode.

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

A general register is a 16-bit readable/writable register that can function as either an ou
compare register or an input capture register. The function is selected by settings in TIO

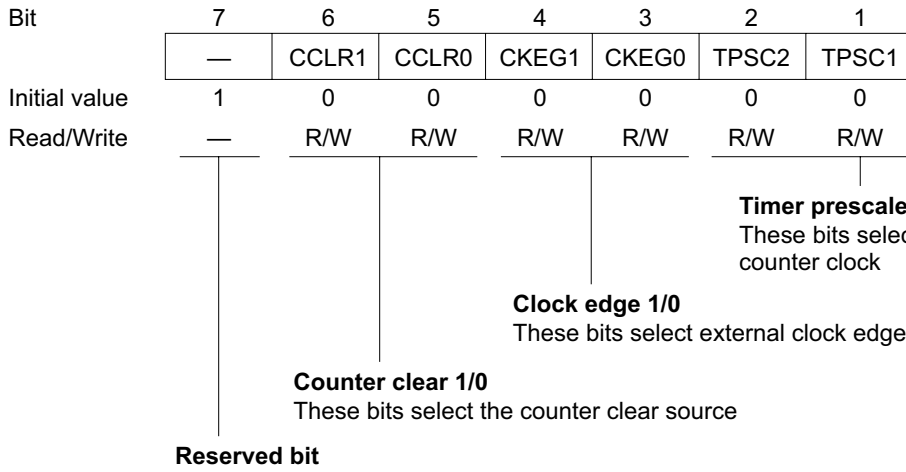
When a general register is used as an output compare register, its value is constantly co
with the 16TCNT value. When the two values match (compare match), the IMFA or IM
set to 1 in TISRA/TISRB. Compare match output can be selected in TIOR.

When a general register is used as an input capture register, an external input capture si
detected and the current 16TCNT value is stored in the general register. The correspond
or IMFB flag in TISRA/TISRB is set to 1 at the same time. The edges of the input capt
are selected in TIOR.

TIOR settings are ignored in PWM mode.

General registers are linked to the CPU by an internal 16-bit bus and can be written or r
either word access or byte access.

General registers are set as output compare registers (with no pin output) and initialized
by a reset and in standby mode.



Each 16TCR is an 8-bit readable/writable register that selects the timer counter clock s
selects the edge or edges of external clock sources, and selects how the counter is clea

16TCR is initialized to H'80 by a reset and in standby mode.

Bit 7—Reserved: This bit cannot be modified and is always read as 1.

		synchronized timers*2	

Notes: 1. 16TCNT is cleared by compare match when the general register functions as compare register, and by input capture when the general register functions a capture register.
2. Selected in TSNC.

Bits 4 and 3—Clock Edge 1 and 0 (CKEG1, CKEG0): These bits select external cloc edges when an external clock source is used.

Bit 4 CKEG1	Bit 3 CKEG0	Description	
0	0	Count rising edges	(In
	1	Count falling edges	
1	—	Count both edges	

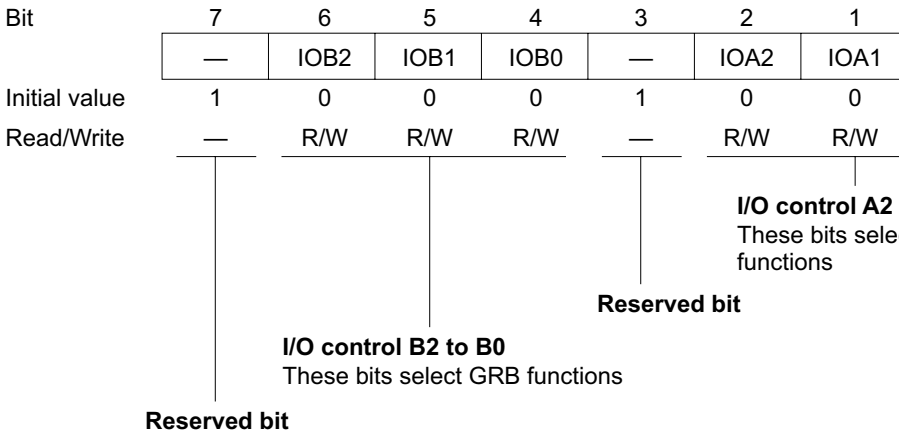
When channel 2 is set to phase counting mode, bits CKEG1 and CKEG0 in 16TCR2 ar Phase counting takes precedence.

Bits 2 to 0—Timer Prescaler 2 to 0 (TPSC2 to TPSC0): These bits select the counter source.

Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Function	
0	0	0	Internal clock: ϕ	(In
		1	Internal clock: $\phi/2$	
	1	0	Internal clock: $\phi/4$	
		1	Internal clock: $\phi/8$	
1	0	0	External clock A: TCLKA input	
		1	External clock B: TCLKB input	
	1	0	External clock C: TCLKC input	
		1	External clock D: TCLKD input	

TIOR is an 8-bit register. The 16-bit timer has three TIORs, one in each channel.

Channel	Abbreviation	Function
0	TIOR0	TIOR controls the general registers. Some functions differ i
1	TIOR1	mode.
2	TIOR2	



Each TIOR is an 8-bit readable/writable register that selects the output compare or inp function for GRA and GRB, and specifies the functions of the TIORA and TIORB pi output compare function is selected, TIOR also selects the type of output. If input cap selected, TIOR also selects the edges of the input capture signal.

TIOR is initialized to H'88 by a reset and in standby mode.

Bit 7—Reserved: This bit cannot be modified and is always read as 1.

1	0	0	GRB is an input compare register	GRB captures rising edge of input
		1		GRB captures falling edge of inpu
	1	0		GRB captures both edges of input
		1		

Notes: 1. After a reset, the output conforms to the TOLR setting until the first compare
2. Channel 2 output cannot be toggled by compare match. When this setting is output is selected automatically.

Bit 3—Reserved: This bit cannot be modified and is always read as 1.

Bits 2 to 0—I/O Control A2 to A0 (IOA2 to IOA0): These bits select the GRA functi

Bit 2 IOA2	Bit 1 IOA1	Bit 0 IOA0	Function	
0	0	0	GRA is an output compare register	No output at compare match (In
		1		0 output at GRA compare match*
	1	0		1 output at GRA compare match*
		1		Output toggles at GRA compare m (1 output in channel 2)*1 *2
1	0	0	GRA is an input compare register	GRA captures rising edge of input
		1		GRA captures falling edge of inpu
	1	0		GRA captures both edges of input
		1		

Notes: 1. After a reset, the output conforms to the TOLR setting until the first compare
2. Channel 2 output cannot be toggled by compare match. When this setting is output is selected automatically.

Output level setting A2 to A0, B2 to
These bits set the levels of the timer
(TIOCA$_2$ to TIOCA$_0$, and TIOCB$_2$ to T

Reserved bits

A TOLR setting can only be made when the corresponding bit in TSTR is 0.

TOLR is a write-only register, and cannot be read. If it is read, all bits will return a va

TOLR is initialized to H'C0 by a reset and in standby mode.

Bits 7 and 6—Reserved: These bits cannot be modified.

Bit 5—Output Level Setting B2 (TOB2): Sets the value of timer output TIOCB$_2$.

Bit 5 TOB2	Description	
0	TIOCB$_2$ is 0	(I
1	TIOCB$_2$ is 1	

Bit 4—Output Level Setting A2 (TOA2): Sets the value of timer output TIOCA$_2$.

Bit 4 TOA2	Description	
0	TIOCA$_2$ is 0	(I
1	TIOCA$_2$ is 1	

Bit 2 TOA1	Description	
0	$TIOCA_1$ is 0	(In
1	$TIOCA_1$ is 1	

Bit 1—Output Level Setting B0 (TOB0): Sets the value of timer output $TIOCB_0$.

Bit 0 TOB0	Description	
0	$TIOCB_0$ is 0	(In
1	$TIOCB_0$ is 1	

Bit 0—Output Level Setting A0 (TOA0): Sets the value of timer output $TIOCA_0$.

Bit 0 TOA0	Description	
0	$TIOCA_0$ is 0	(In
1	$TIOCA_0$ is 1	

Figures 9.6 to 9.9 show examples of byte read/write access to 16TCNTH and 16TCNT

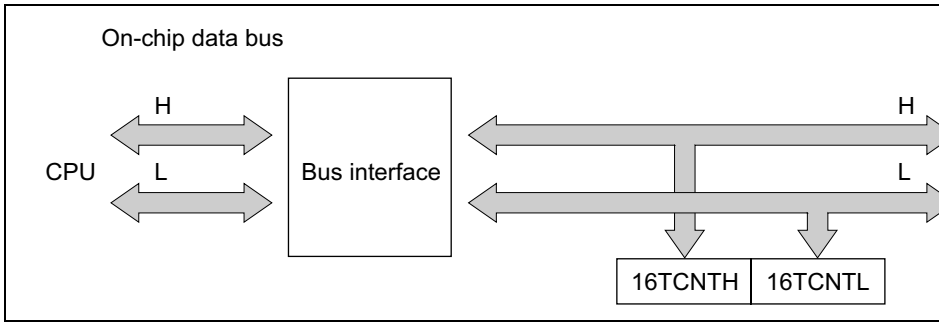


Figure 9.4 16TCNT Access Operation [CPU → 16TCNT (Word)]

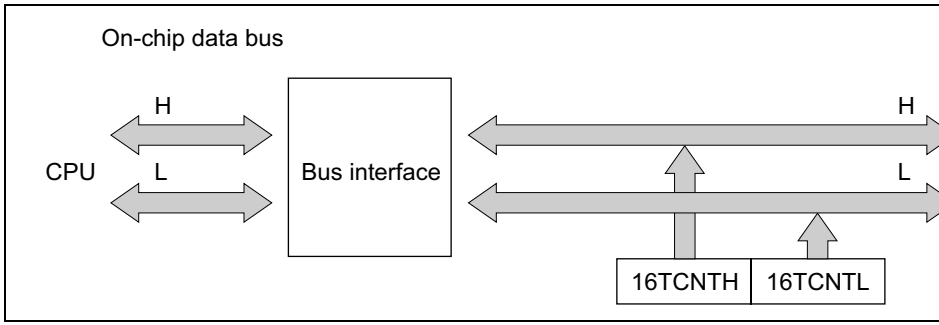


Figure 9.5 Access to Timer Counter (CPU Reads 16TCNT, Word)

Figure 9.6 Access to Timer Counter H (CPU Writes to 16TCNTH, Upper I

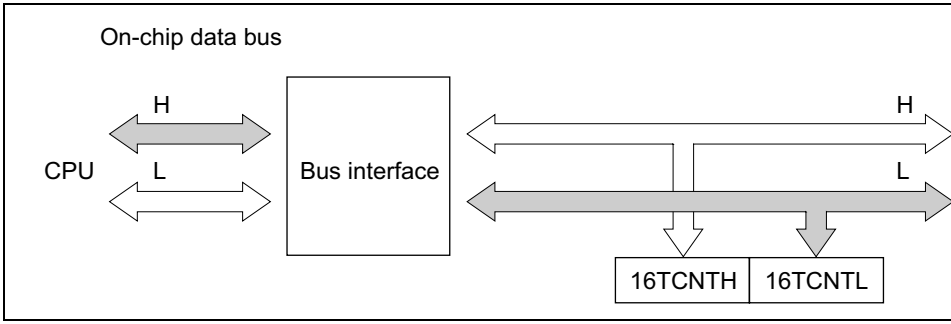


Figure 9.7 Access to Timer Counter L (CPU Writes to 16TCNTL, Lower I

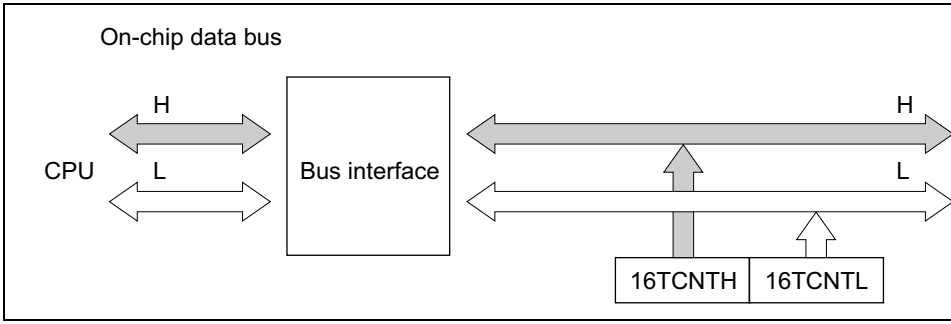


Figure 9.8 Access to Timer Counter H (CPU Reads 16TCNTH, Upper By

Figure 9.9 Access to Timer Counter L (CPU Reads 16TCNTL, Lower B

9.3.2 8-Bit Accessible Registers

The registers other than the timer counters and general registers are 8-bit registers. Th
are linked to the CPU by an internal 8-bit data bus.

Figures 9.10 and 9.11 show examples of byte read and write access to a 16TCR.

If a word-size data transfer instruction is executed, two byte transfers are performed.

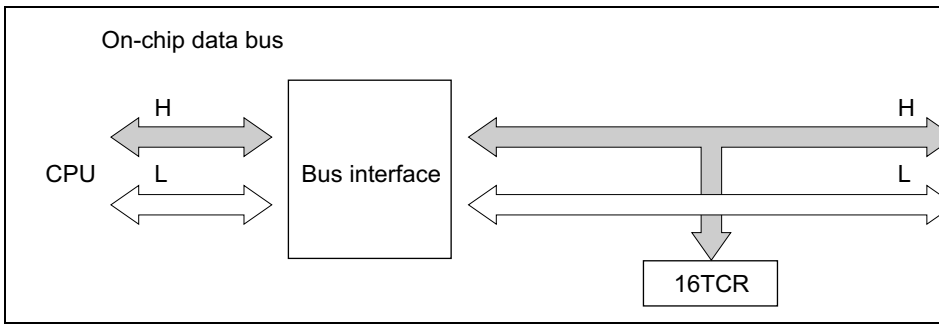


Figure 9.10 16TCR Access (CPU Writes to 16TCR)

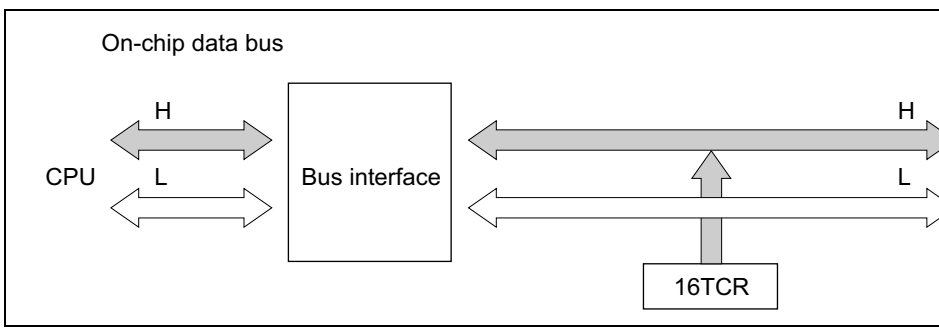


Figure 9.11 16TCR Access (CPU Reads 16TCR)

Synchronous Operation: The timer counters in designated channels are preset synchro
Data written to the timer counter in any one of these channels is simultaneously written
timer counters in the other channels as well. The timer counters can also be cleared syn
if so designated by the CCLR1 and CCLR0 bits in the TCRs.

PWM Mode: A PWM waveform is output from the TIOCA pin. The output goes to 1 a
match A and to 0 at compare match B. The duty cycle can be varied from 0% to 100%
on the settings of GRA and GRB. When a channel is set to PWM mode, its GRA and G
automatically become output compare registers.

Phase Counting Mode: The phase relationship between two clock signals input at TCl
TCLKB is detected and 16TCNT2 counts up or down accordingly. When phase countir
selected TCLKA and TCLKB become clock input pins and 16TCNT2 operates as an up
counter.

9.4.2 Basic Functions

Counter Operation: When one of bits STR0 to STR2 is set to 1 in the timer start regis
the timer counter (16TCNT) in the corresponding channel starts counting. The counting
free-running or periodic.

- Sample setup procedure for counter
  Figure 9.12 shows a sample procedure for setting up a counter.

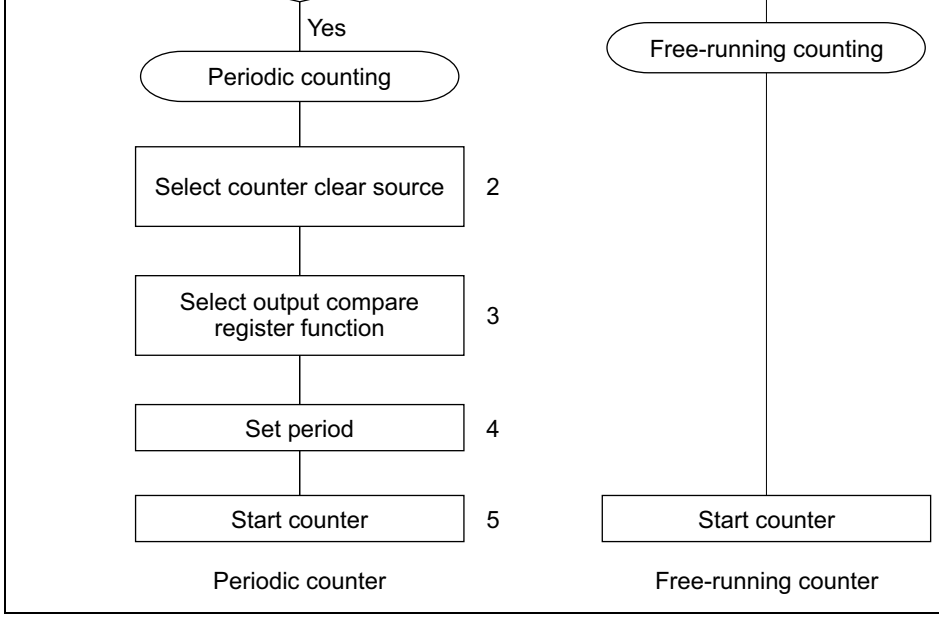


Figure 9.12 Counter Setup Procedure (Example)

1. Set bits TPSC2 to TPSC0 in 16TCR to select the counter clock source. If an extern source is selected, set bits CKEG1 and CKEG0 in 16TCR to select the desired edg external clock signal.
2. For periodic counting, set CCLR1 and CCLR0 in 16TCR to have 16TCNT cleared compare match or GRB compare match.
3. Set TIOR to select the output compare function of GRA or GRB, whichever was s step 2.
4. Write the count period in GRA or GRB, whichever was selected in step 2.
5. Set the STR bit to 1 in TSTR to start the timer counter.

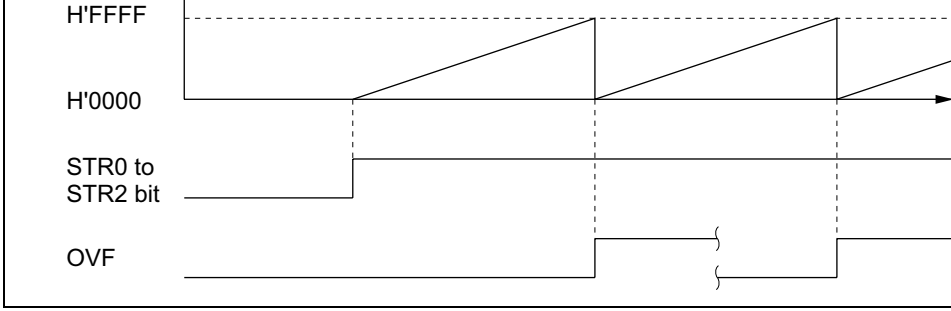


Figure 9.13 Free-Running Counter Operation

When a channel is set to have its counter cleared by compare match, in that channel operates as a periodic counter. Select the output compare function of GRA or GRB, CCLR1 or CCLR0 in 16TCR to have the counter cleared by compare match, and se period in GRA or GRB. After these settings, the counter starts counting up as a peri counter when the corresponding bit is set to 1 in TSTR. When the count matches GI GRB, the IMFA or IMFB flag is set to 1 in TISRA/TISRB and the counter is cleare H'0000. If the corresponding IMIEA or IMIEB bit is set to 1 in TISRA/TISRB, a CI interrupt is requested at this time. After the compare match, 16TCNT continues cou from H'0000. Figure 9.14 illustrates periodic counting.

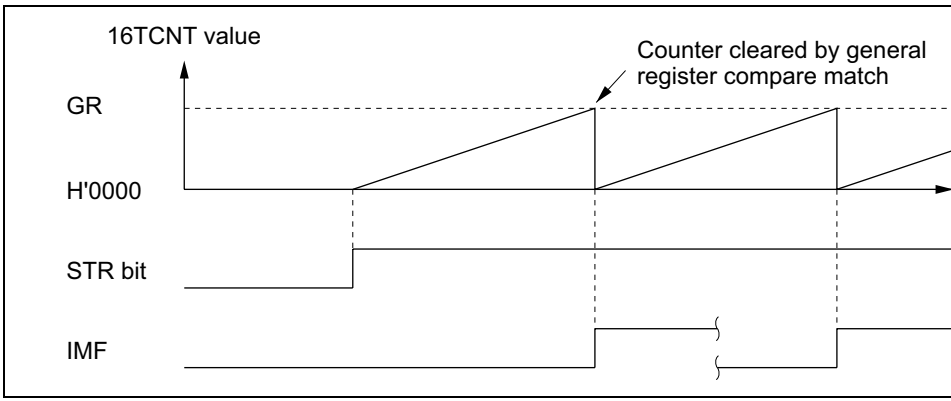


Figure 9.14 Periodic Counter Operation

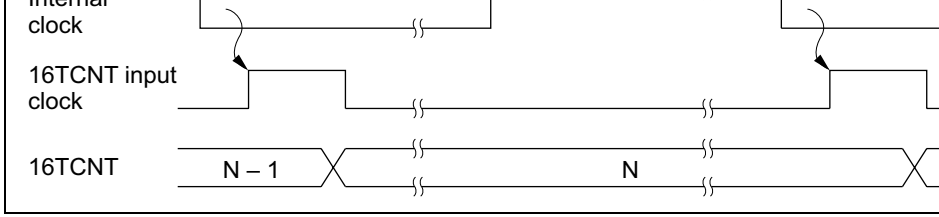


Figure 9.15 Count Timing for Internal Clock Sources

— External clock source

The external clock pin (TCLKA to TCLKD) can be selected by bits TPSC2 to 16TCR, and the detected edge by bits CKEG1 and CKEG0. The rising edge, fa or both edges can be selected.

The pulse width of the external clock signal must be at least 1.5 system clocks single edge is selected, and at least 2.5 system clocks when both edges are sele pulses will not be counted correctly.

Figure 9.16 shows the timing when both edges are detected.

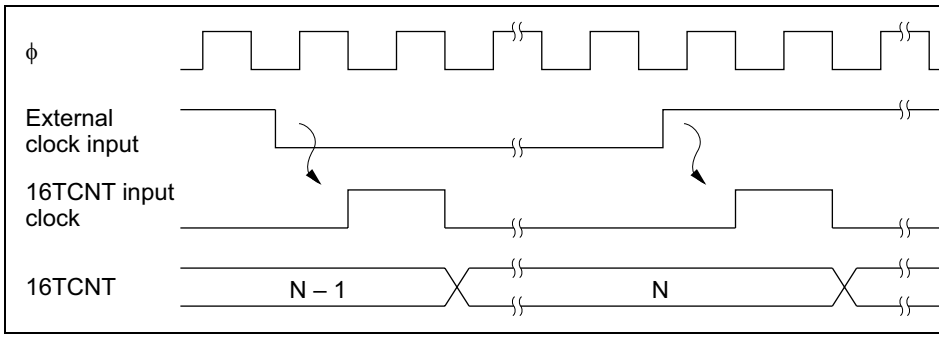


Figure 9.16 Count Timing for External Clock Sources (when Both Edges are

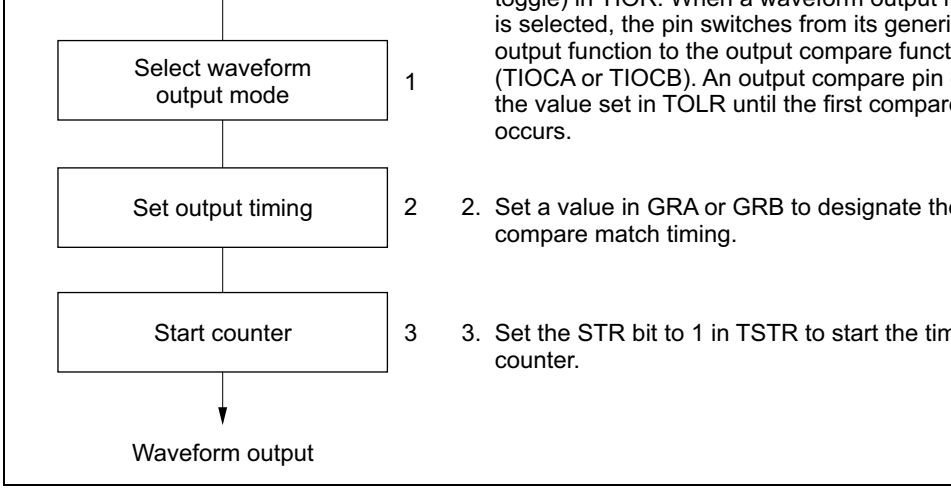


Figure 9.17 Setup Procedure for Waveform Output by Compare Match (Ex

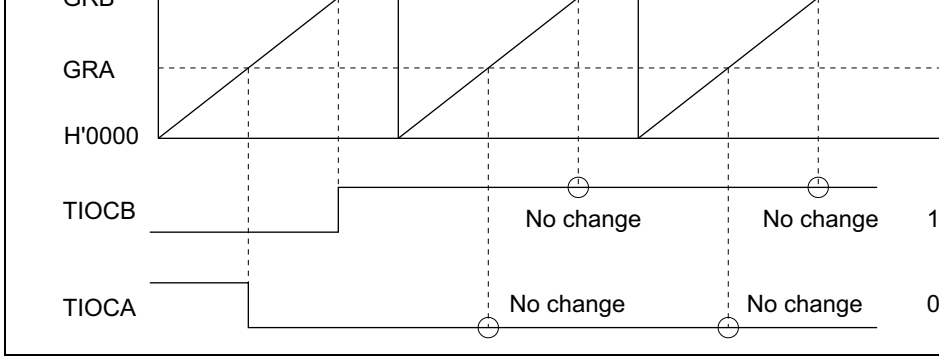


Figure 9.18 0 and 1 Output (TOA = 1, TOB = 0)

Figure 9.19 shows examples of toggle output. 16TCNT operates as a periodic coun
by compare match B. Toggle output is selected for both compare match A and B.

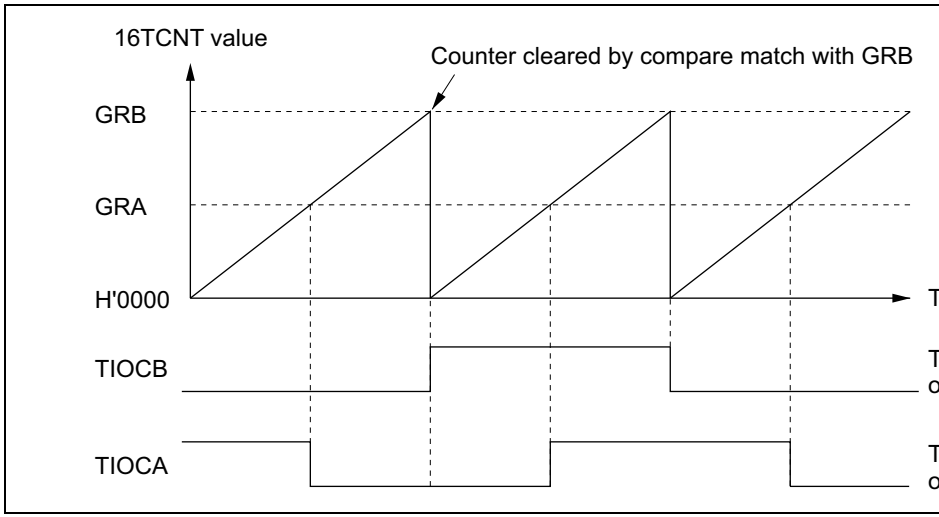


Figure 9.19 Toggle Output (TOA = 1, TOB = 0)

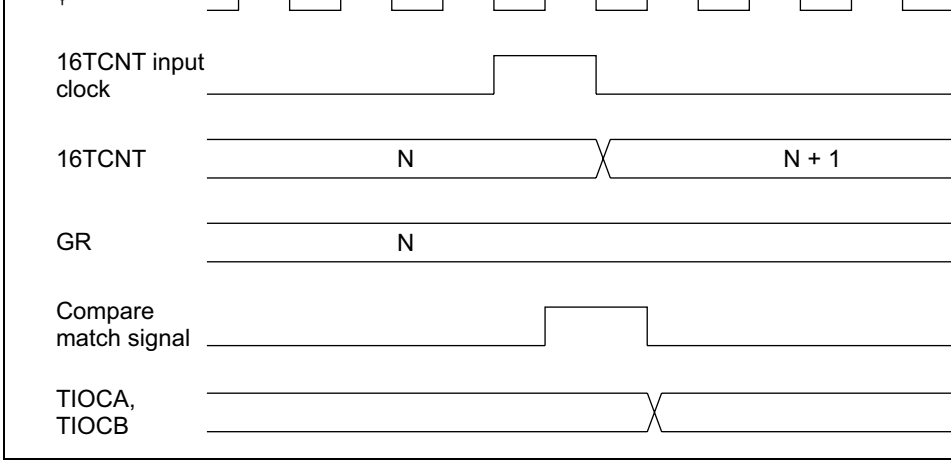


Figure 9.20 Output Compare Output Timing

Input Capture Function: The 16TCNT value can be transferred to a general register v
input edge is detected at an input capture input/output compare pin (TIOCA or TIOCB)
edge, falling-edge, or both-edge detection can be selected. The input capture function c
to measure pulse width or period.

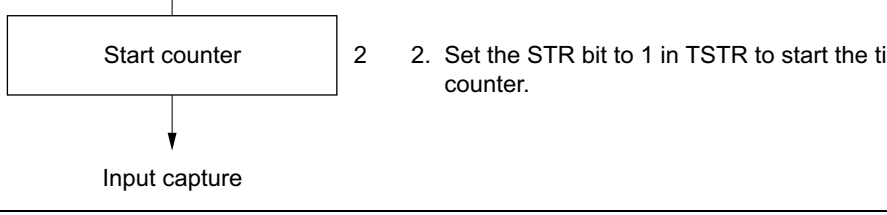


Figure 9.21 Setup Procedure for Input Capture (Example)

- Examples of input capture

  Figure 9.22 illustrates input capture when the falling edge of TIOCB and both edg
  are selected as capture edges. 16TCNT is cleared by input capture into GRB.

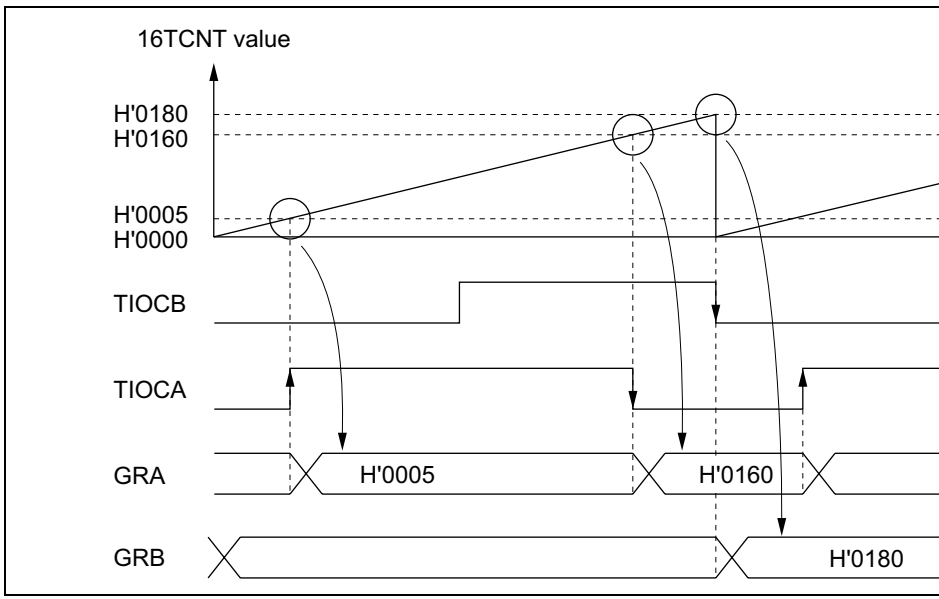


Figure 9.22 Input Capture (Example)

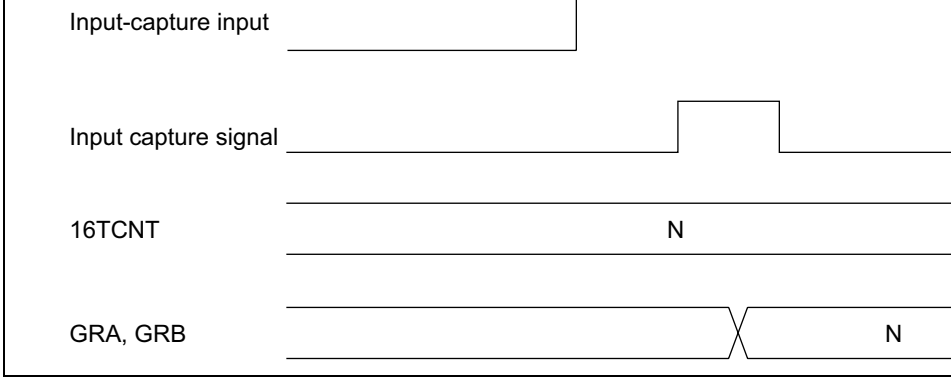
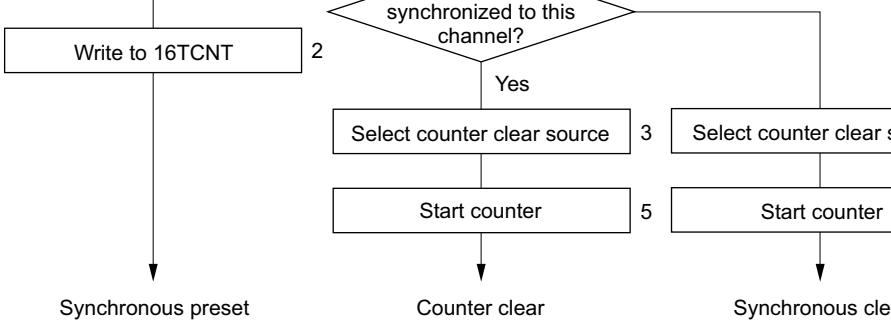


Figure 9.23 Input Capture Signal Timing

9.4.3 Synchronization

The synchronization function enables two or more timer counters to be synchronized by the same data to them simultaneously (synchronous preset). With appropriate 16TCR s or more timer counters can also be cleared simultaneously (synchronous clear). Synchr enables additional general registers to be associated with a single time base. Synchroni be selected for all channels (0 to 2).

Sample Setup Procedure for Synchronization: Figure 9.24 shows a sample procedur setting up synchronization.



1. Set the SYNC bits to 1 in TSNC for the channels to be synchronized.
2. When a value is written in 16TCNT in one of the synchronized channels, the same value is simultaneously written in 16TCNT in the other channels.
3. Set the CCLR1 or CCLR0 bit in 16TCR to have the counter cleared by compare match or in
4. Set the CCLR1 and CCLR0 bits in 16TCR to have the counter cleared synchronously.
5. Set the STR bits in TSTR to 1 to start the synchronized counters.

Figure 9.24 Setup Procedure for Synchronization (Example)

Example of Synchronization: Figure 9.25 shows an example of synchronization. Cha and 2 are synchronized, and are set to operate in PWM mode. Channel 0 is set for cou by compare match with GRB0. Channels 1 and 2 are set for synchronous counter clea timer counters in channels 0, 1, and 2 are synchronously preset, and are synchronously compare match with GRB0. A three-phase PWM waveform is output from pins TIOC and $TIOCA_2$. For further information on PWM mode, see section 9.4.4, PWM Mode.

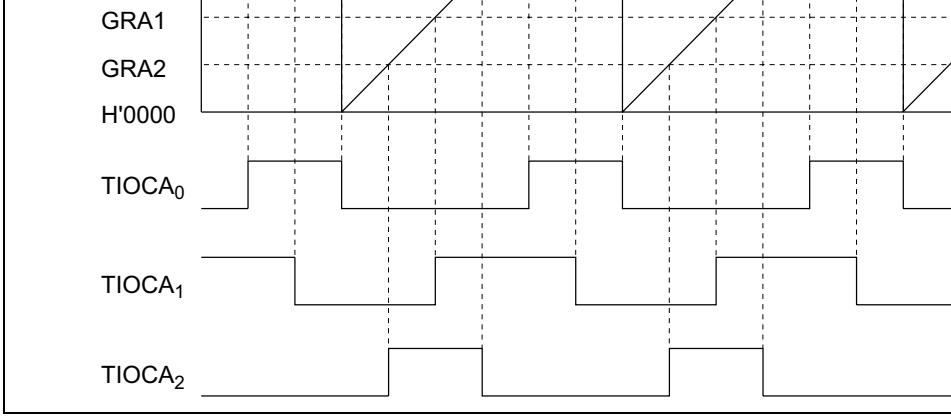


Figure 9.25 Synchronization (Example)

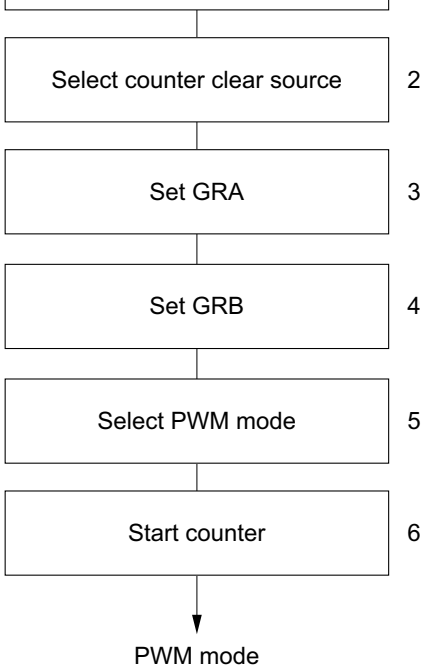
9.4.4 PWM Mode

In PWM mode GRA and GRB are paired and a PWM waveform is output from the TIO
GRA specifies the time at which the PWM output changes to 1. GRB specifies the time
the PWM output changes to 0. If either GRA or GRB compare match is selected as the
clear source, a PWM waveform with a duty cycle from 0% to 100% is output at the TIO
PWM mode can be selected in all channels (0 to 2).

Table 9.4 summarizes the PWM output pins and corresponding registers. If the same va
in GRA and GRB, the output does not change when compare match occurs.

Table 9.4 PWM Output Pins and Registers

Channel	Output Pin	1 Output	0 Output
0	$TIOCA_0$	GRA0	GRB0
1	$TIOCA_1$	GRA1	GRB1
2	$TIOCA_2$	GRA2	GRB2



2. Set bits CCLR1 and CCLR0 in 1
to select the counter clear source

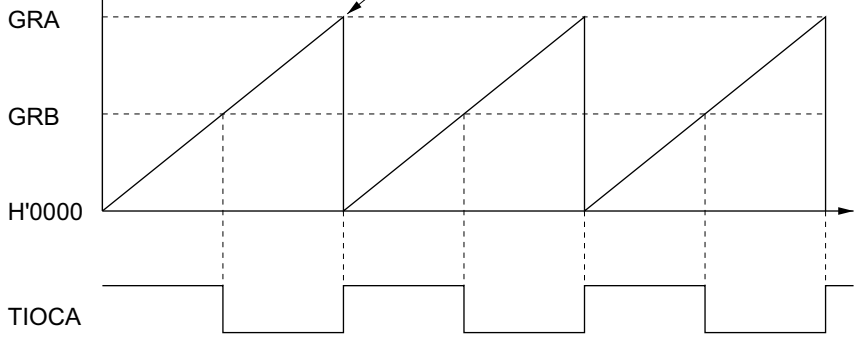
3. Set the time at which the PWM
waveform should go to 1 in GRA

4. Set the time at which the PWM
waveform should go to 0 in GRB

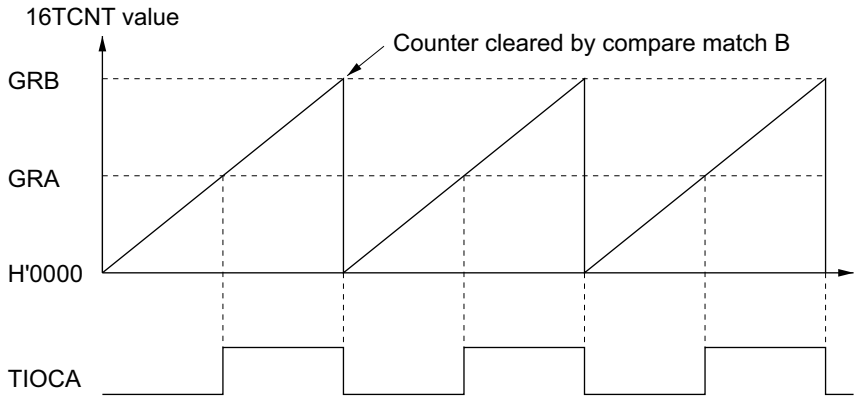
5. Set the PWM bit in TMDR to sele
PWM mode. When PWM mode i
selected, regardless of the TIOR
contents, GRA and GRB become
output compare registers specify
the times at which the PWM outp
goes to 1 and 0. The TIOCA pin
automatically becomes the PWM
output pin. The TIOCB pin confo
to the settings of bits IOB1 and I
in TIOR. If TIOCB output is not
desired, clear both IOB1 and IOB

6. Set the STR bit to 1 in TSTR to s
the timer counter.

Figure 9.26 Setup Procedure for PWM Mode (Example)



a. Counter cleared by GRA (TOA = 1)



b. Counter cleared by GRB (TOA = 0)

Figure 9.27 PWM Mode (Example 1)

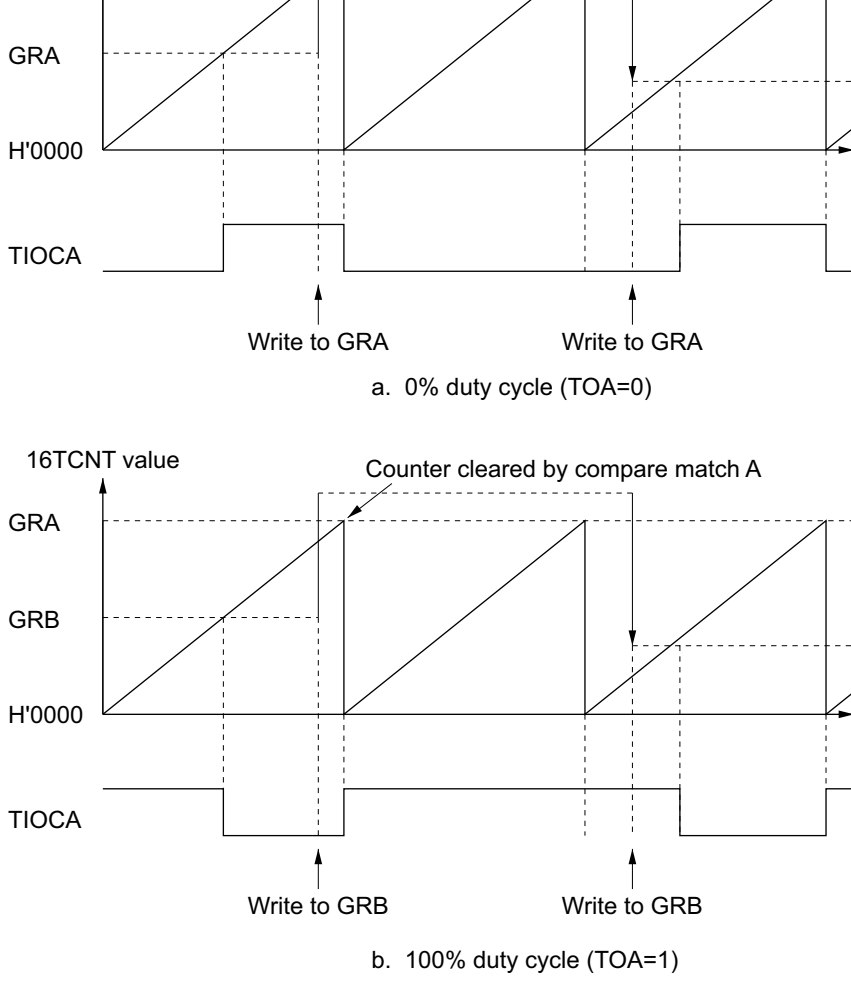


Figure 9.28 PWM Mode (Example 2)

valid. The input capture and output compare functions can be used, and interrupts can b
generated.

Phase counting is available only in channel 2.

Sample Setup Procedure for Phase Counting Mode: Figure 9.29 shows a sample pro
setting up phase counting mode.

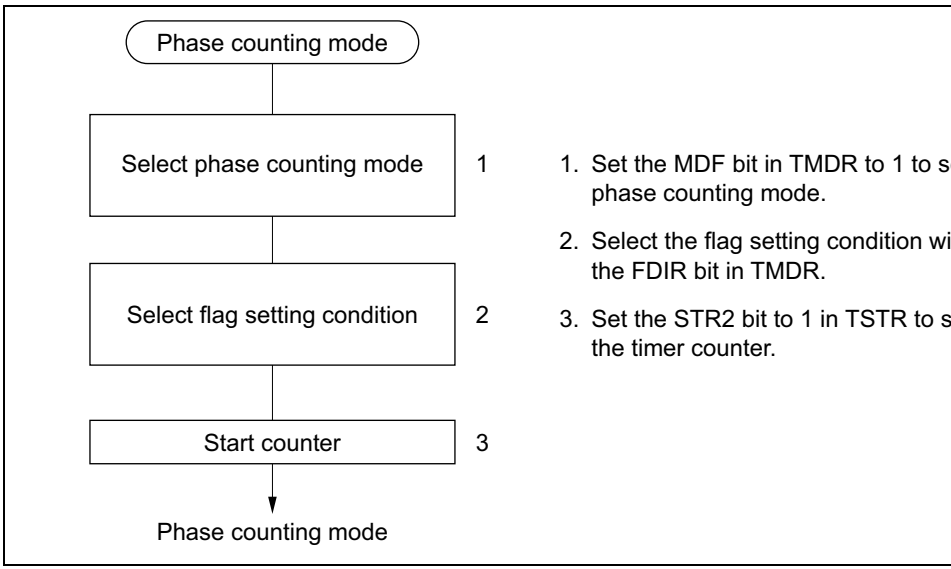


Figure 9.29 Setup Procedure for Phase Counting Mode (Example)

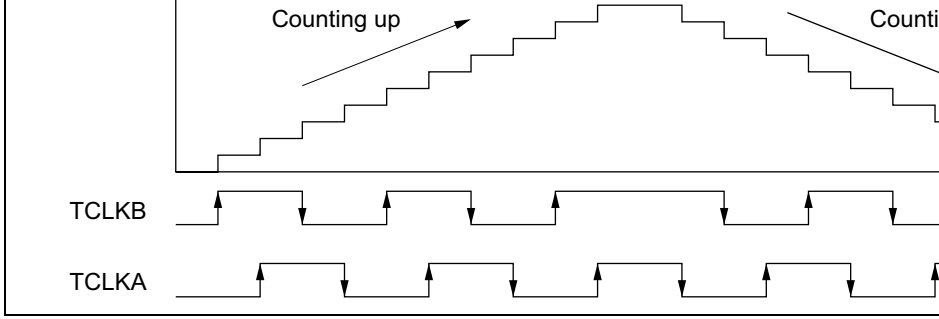


Figure 9.30 Operation in Phase Counting Mode (Example)

Table 9.5 Up/Down Counting Conditions

Counting Direction	Up-Counting				Down-Counting		
TCLKB pin	↑	High	↓	Low	HIgh	↓	Low
TCLKA pin	Low	↑	High	↓	↓	Low	↑

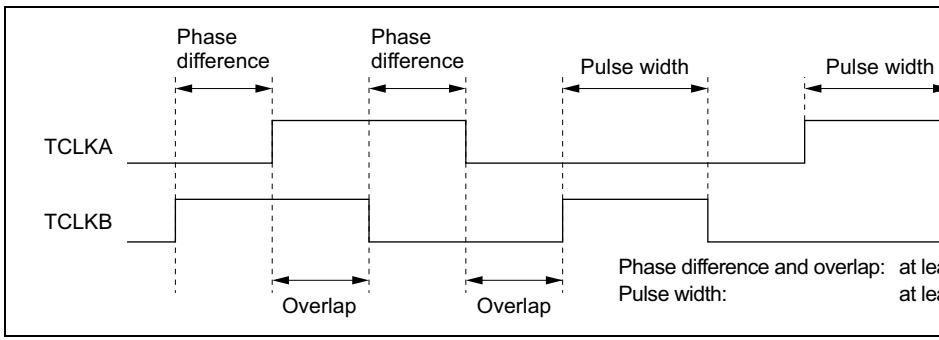


Figure 9.31 Phase Difference, Overlap, and Pulse Width in Phase Counting

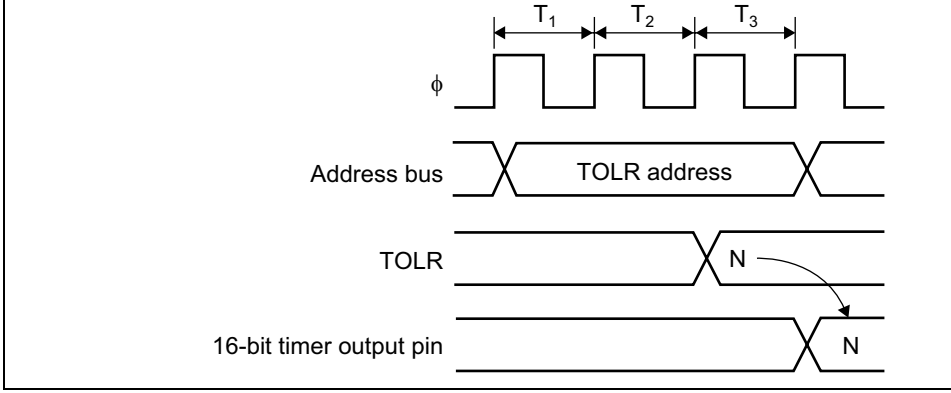


Figure 9.32 Timing for Setting 16-Bit Timer Output Level by Writing to T

match signal is generated in the last state in which the values match (when 16TCNT is from the matching count to the next count). Therefore, when 16TCNT matches a gene the compare match signal is not generated until the next 16TCNT clock input. Figure the timing of the setting of IMFA and IMFB.

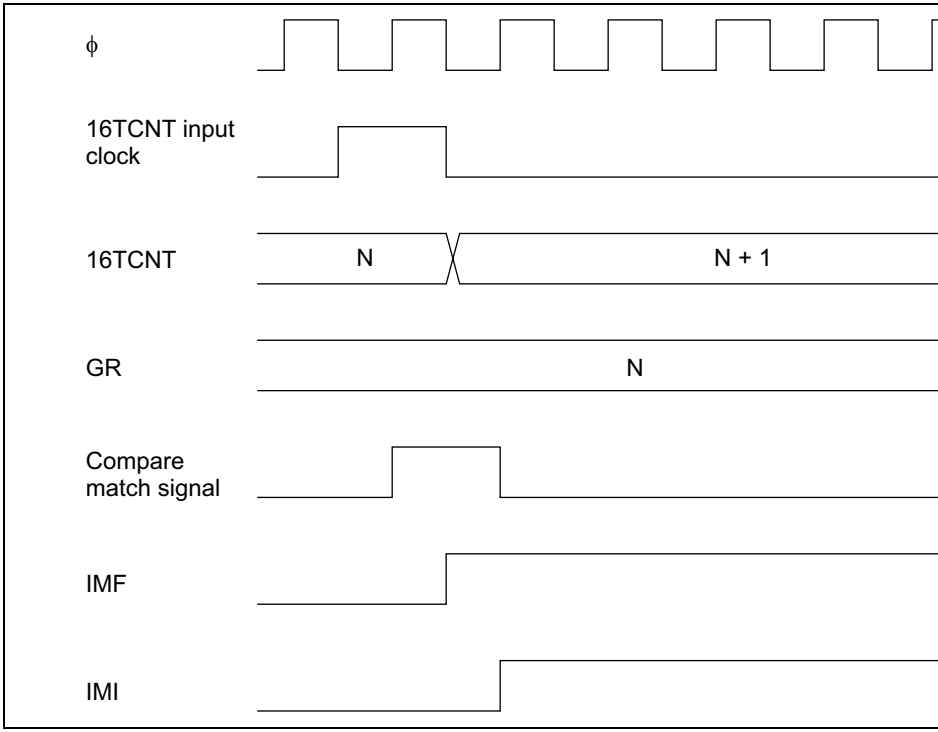


Figure 9.33 Timing of Setting of IMFA and IMFB by Compare Matc

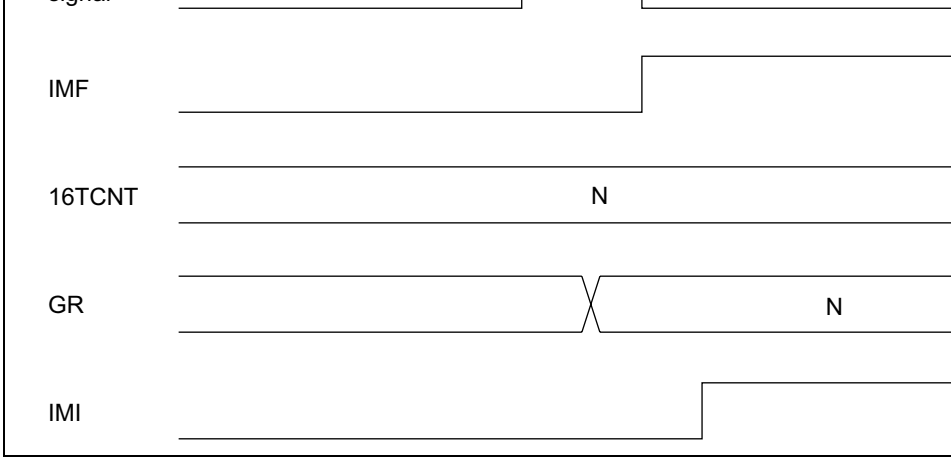


Figure 9.34 Timing of Setting of IMFA and IMFB by Input Capture

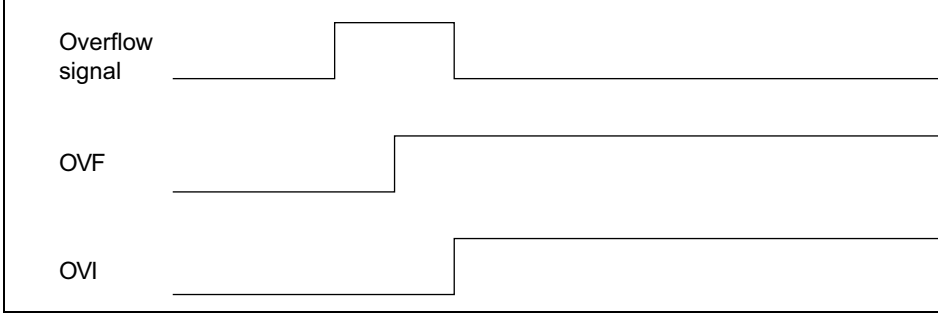


Figure 9.35 Timing of Setting of OVF

9.5.2 Timing of Clearing of Status Flags

If the CPU reads a status flag while it is set to 1, then writes 0 in the status flag, the sta
cleared. Figure 9.36 shows the timing.

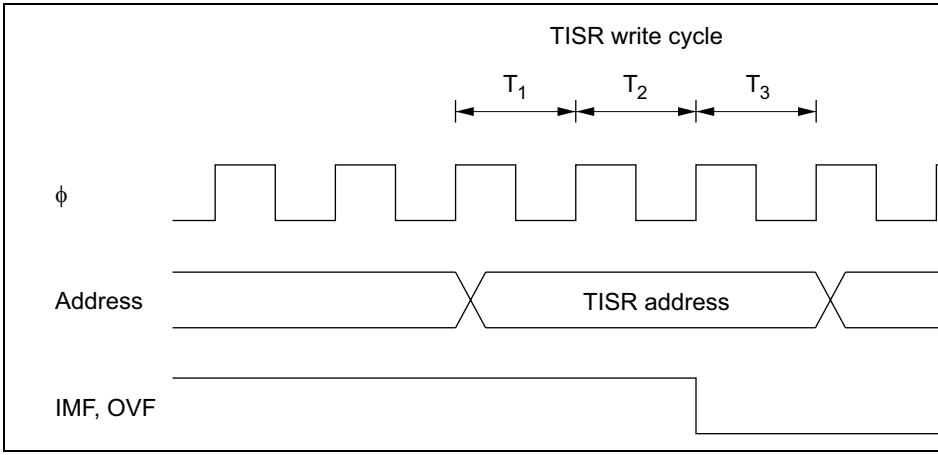


Figure 9.36 Timing of Clearing of Status Flags

Table 9.6 lists the interrupt sources.

Table 9.6 16-bit timer Interrupt Sources

Channel	Interrupt Source	Description	Pri
0	IMIA0	Compare match/input capture A0	Hig
	IMIB0	Compare match/input capture B0	
	OVI0	Overflow 0	
1	IMIA1	Compare match/input capture A1	
	IMIB1	Compare match/input capture B1	
	OVI1	Overflow 1	
2	IMIA2	Compare match/input capture A2	
	IMIB2	Compare match/input capture B2	
	OVI2	Overflow 2	Lov

Note: * The priority immediately after a reset is indicated. Inter-channel priorities can be by settings in IPRA.

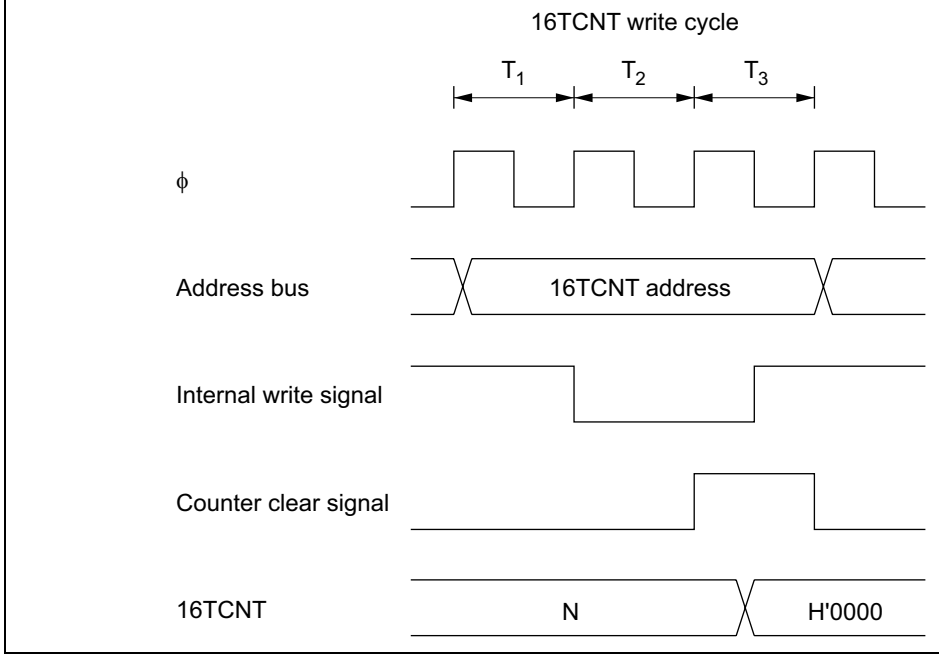


Figure 9.37 Contention between 16TCNT Write and Clear

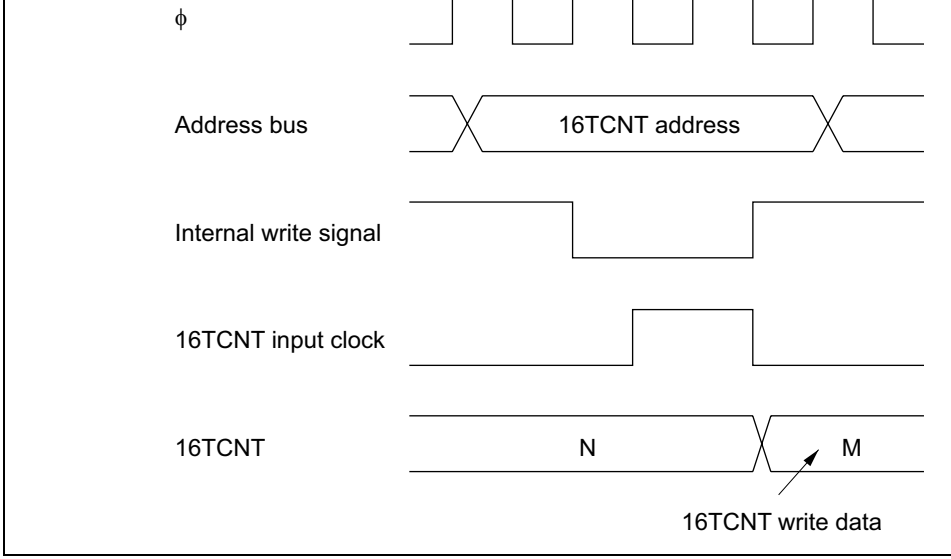


Figure 9.38 Contention between 16TCNT Word Write and Increment

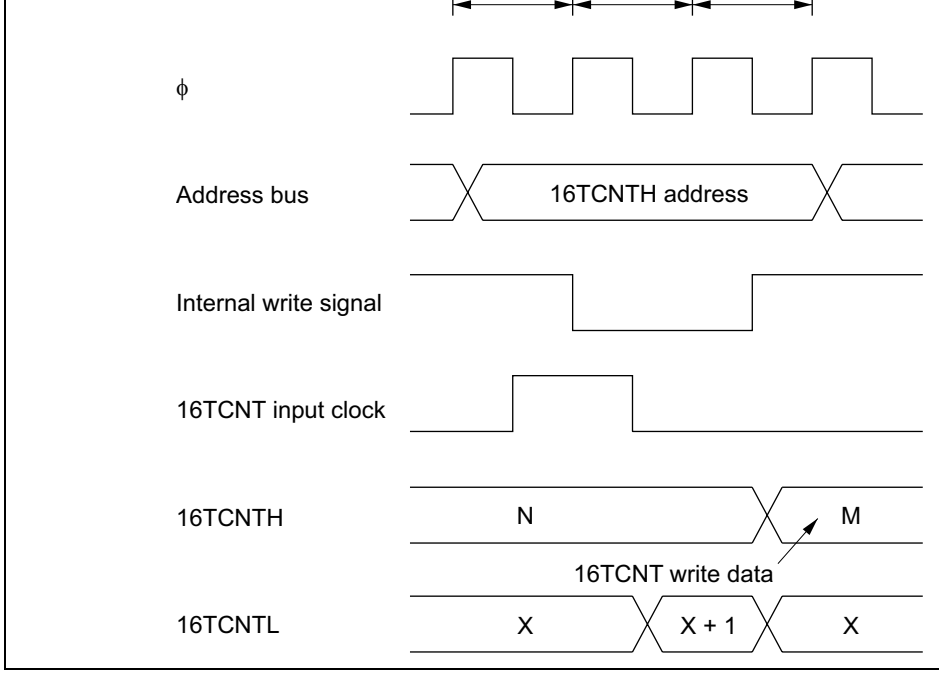


Figure 9.39 Contention between 16TCNT Byte Write and Incremen

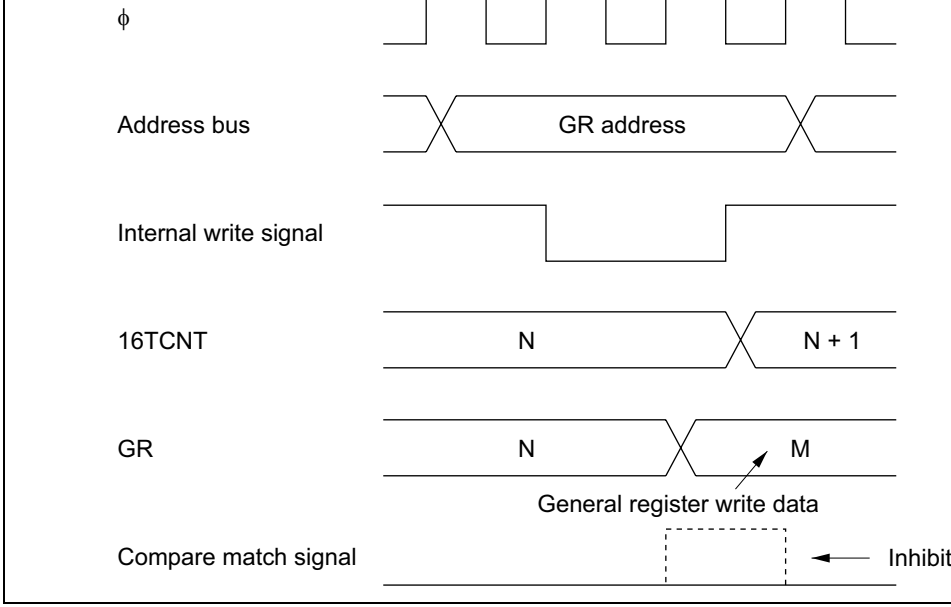


Figure 9.40 Contention between General Register Write and Compare Ma

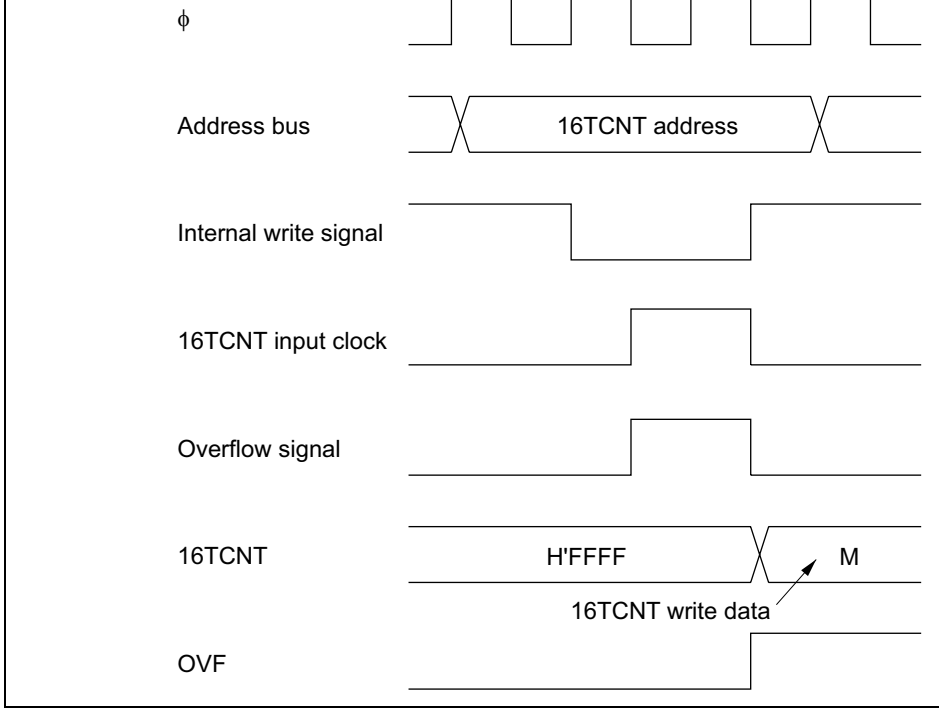


Figure 9.41 Contention between 16TCNT Write and Overflow

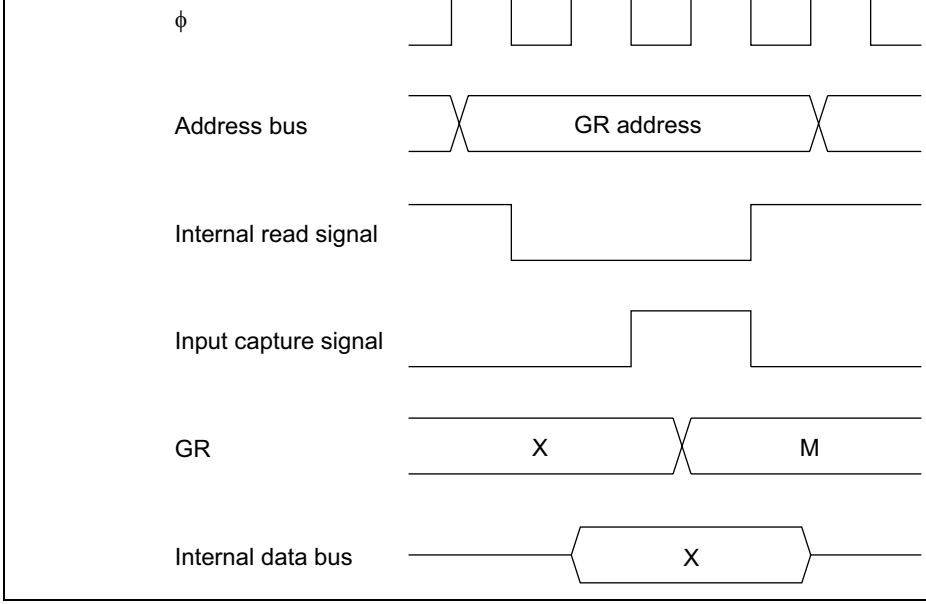


Figure 9.42 Contention between General Register Read and Input Captu

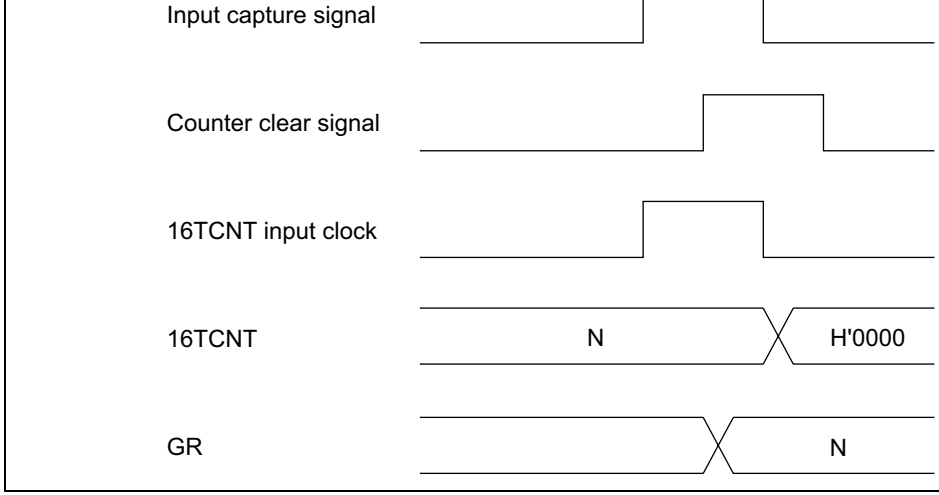


Figure 9.43 Contention between Counter Clearing by Input Capture and C Increment

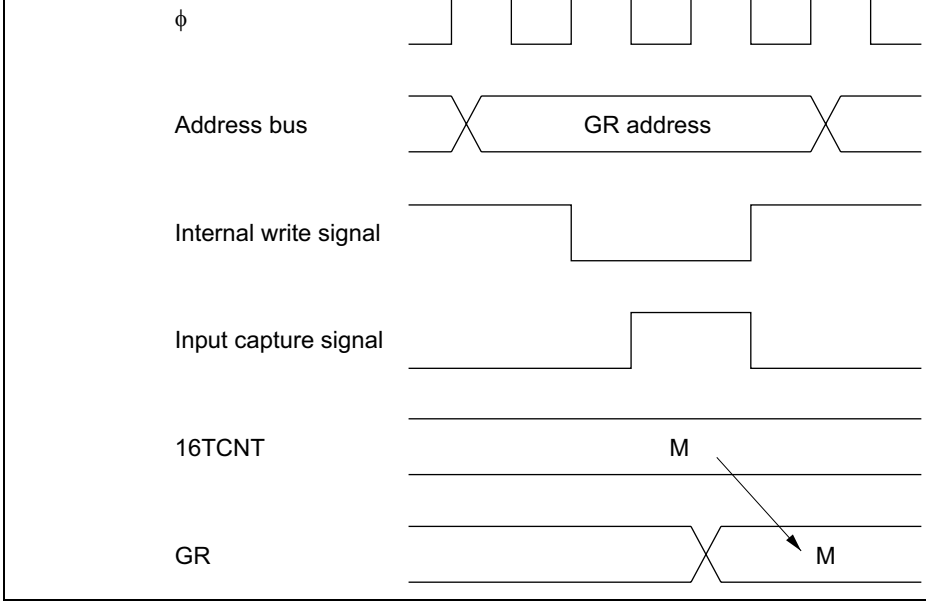
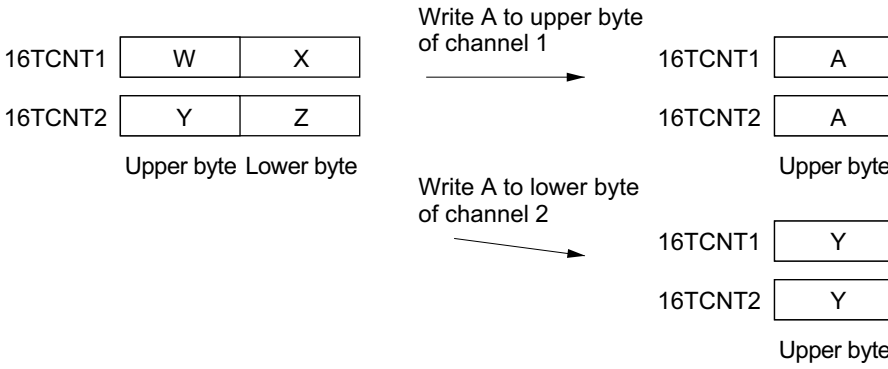


Figure 9.44 Contention between General Register Write and Input Captu

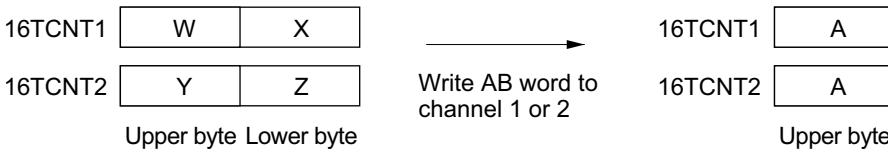
Note on Writes in Synchronized Operation: When channels are synchronized, if a 1
value is modified by byte write access, all 16 bits of all synchronized counters assume
value as the counter that was addressed.

(Example) When channels 1 and 2 are synchronized

- Byte write to channel 1 or byte write to channel 2



- Word write to channel 1 or word write to channel 2



PWM mode		○	—	—	PWM0 = 1	—	○	○
Output compare A		○	—	—	PWM0 = 0	IOA2 = 0 Other bits unrestricted	○	○
Output compare B		○	—	—	○	○	IOB2 = 0 Other bits unrestricted	○
Input capture A		○	—	—	PWM0 = 0	IOA2 = 1 Other bits unrestricted	○	○
Input capture B		○	—	—	PWM0 = 0	○	IOB2 = 1 Other bits unrestricted	○
Counter clearing	By compare match/input capture A	○	—	—	○	○	○	CCLR1 CCLR0
	By compare match/input capture B	○	—	—	○	○	○	CCLR1 CCLR0
	Synchronous clear	SYNC0 = 1	—	—	○	○	○	CCLR1 CCLR0

Legend: ○ Setting available (valid). — Setting does not affect this mode.

Note: * The input capture function cannot be used in PWM mode. If compare match A and compare m
simultaneously, the compare match signal is inhibited.

						unrestricted		
Output compare B		○	—	—	○	○	IOB2 = 0 Other bits unrestricted	○
Input capture A		○	—	—	PWM1 = 0	IOA2 = 1 Other bits unrestricted	○	○
Input capture B		○	—	—	PWM1 = 0	○	IOB2 = 1 Other bits unrestricted	○
Counter clearing	By compare match/input capture A	○	—	—	○	○	○	CCLR CCLR
	By compare match/input capture B	○	—	—	○	○	○	CCLR CCLR
	Synchronous clear	SYNC1 = 1	—	—	○	○	○	CCLR CCLR

Legend: ○ Setting available (valid). — Setting does not affect this mode.

Note: * The input capture function cannot be used in PWM mode. If compare match A and compare
occur simultaneously, the compare match signal is inhibited.

						unrestricted		
Output compare B		○	○	—	○	○	IOB2 = 0 Other bits unrestricted	○
Input capture A		○	○	—	PWM2 = 0	IOA2 = 1 Other bits unrestricted	○	○
Input capture B		○	○	—	PWM2 = 0	○	IOB2 = 1 Other bits unrestricted	○
Counter clearing	By compare match/input capture A	○	○	—	○	○	○	CCLR1 CCLR0
	By compare match/input capture B	○	○	—	○	○	○	CCLR1 CCLR0
	Synchronous clear	SYNC2 = 1	○	—	○	○	○	CCLR1 CCLR0
Phase counting mode		○	MDF = 1	○	○	○	○	○

Legend: ○ Setting available (valid). — Setting does not affect this mode.

Note: * The input capture function cannot be used in PWM mode. If compare match A and compare m simultaneously, the compare match signal is inhibited.

variety of applications, including the generation of a rectangular-wave output with an
duty cycle.

10.1.1 Features

The features of the 8-bit timer module are listed below.

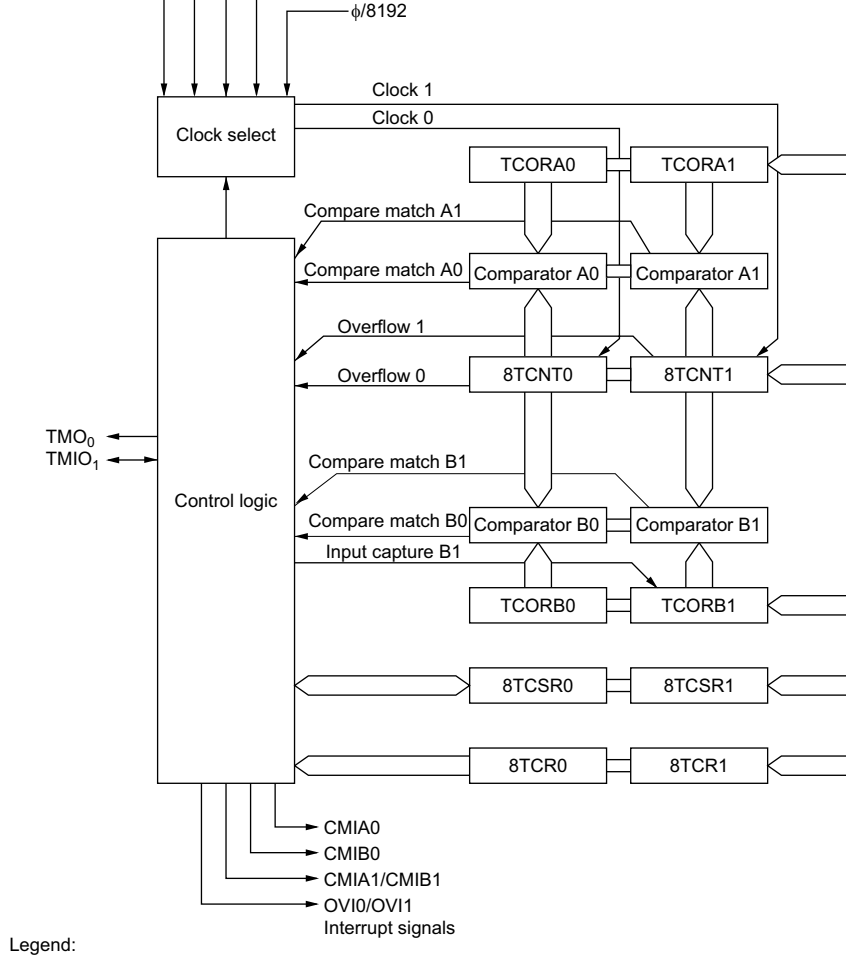
- Selection of four clock sources

  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 input capture B.
- Timer output controlled by two compare match signals

  The timer output signal in each channel is controlled by two independent compare
  signals, enabling the timer to generate output waveforms with an arbitrary duty cy
  output.
- A/D converter can be activated by a compare match
- Two channels can be cascaded
  - Channels 0 and 1 can be operated as the upper and lower halves of a 16-bit tim
    count mode).
  - Channels 2 and 3 can be operated as the upper and lower halves of a 16-bit tim
    count mode).
  - Channel 1 can count channel 0 compare match events (compare match count m
  - Channel 3 can count channel 2 compare match events (compare match count m
- Input capture function can be set

  8-bit or 16-bit input capture operation is available.



Legend:
TCORA: Time constant register A
TCORB: Time constant register B
8TCNT: Timer counter
8TCSR: Timer control/status register
8TCR: Timer control register

Figure 10.1 Block Diagram of 8-Bit Timer Unit (Two Channels: Group

	1	Timer input/output	$TMIO_1$	I/O	Compare match output capture input
		Timer clock input	TCLKA	Input	Counter external clock
1	2	Timer output	TMO_2	Output	Compare match output
		Timer clock input	TCLKD	Input	Counter external clock
	3	Timer input/output	$TMIO_3$	I/O	Compare match output capture input
		Timer clock input	TCLKB	Input	Counter external clock

	H'FFF84	Time constant register A0	TCORA0	R/W	I
	H'FFF86	Time constant register B0	TCORB0	R/W	I
	H'FFF88	Timer counter 0	8TCNT0	R/W	I
1	H'FFF81	Timer control register 1	8TCR1	R/W	I
	H'FFF83	Timer control/status register 1	8TCSR1	R/(W)*2	I
	H'FFF85	Time constant register A1	TCORA1	R/W	I
	H'FFF87	Time constant register B1	TCORB1	R/W	I
	H'FFF89	Timer counter 1	8TCNT1	R/W	I
2	H'FFF90	Timer control register 2	8TCR2	R/W	I
	H'FFF92	Timer control/status register 2	8TCSR2	R/(W)*2	I
	H'FFF94	Time constant register A2	TCORA2	R/W	I
	H'FFF96	Time constant register B2	TCORB2	R/W	I
	H'FFF98	Timer counter 2	8TCNT2	R/W	I
3	H'FFF91	Timer control register 3	8TCR3	R/W	I
	H'FFF93	Timer control/status register 3	8TCSR3	R/(W)*2	I
	H'FFF95	Time constant register A3	TCORA3	R/W	I
	H'FFF97	Time constant register B3	TCORB3	R/W	I
	H'FFF99	Timer counter 3	8TCNT3	R/W	I

Notes: 1. Indicates the lower 20 bits of the address in advanced mode.
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 comprises a 16-bit register with the
register as the upper 8 bits and the channel 1 register as the lower 8 bits, so they can b
together by word access.

Similarly, each pair of registers for channel 2 and channel 3 comprises a 16-bit registe
channel 2 register as the upper 8 bits and the channel 3 register as the lower 8 bits, so
accessed together by word access.

Initial value	0	0	0	0	0	0	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	R/W	R/W	R/W	R/W	R/W	R/W

	8TCNT2								8TCNT3					
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
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

The timer counters (8TCNT) are 8-bit readable/writable up-counters that increment on
generated from an internal or external clock source. The clock source is selected by clo
bits 2 to 0 (CKS2 to CKS0) in the timer control register (8TCR). The CPU can always
write to the timer counters.

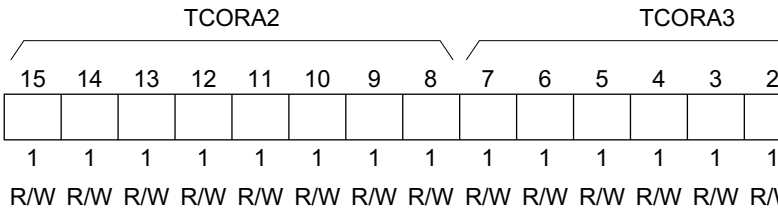
The 8TCNT0 and 8TCNT1 pair, and the 8TCNT2 and 8TCNT3 pair, can each be acces
16-bit register by word access.

8TCNT can be cleared by an input capture signal or compare match signal. Counter clo
and 0 (CCLR1 and CCLR0) in 8TCR select the method of clearing.

When 8TCNT overflows from H'FF to H'00, the overflow flag (OVF) in the timer contr
register (8TCSR) is set to 1.

Each 8TCNT is initialized to H'00 by a reset and in standby mode.

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	R/

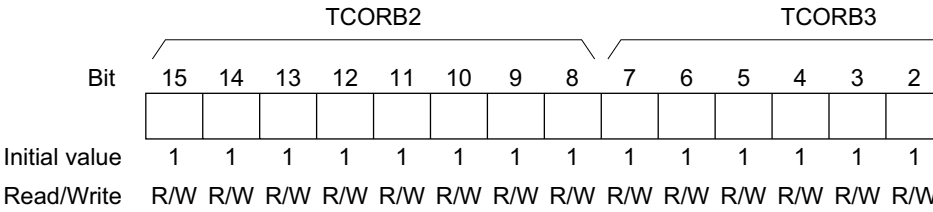


The TCORA0 and TCORA1 pair, and the TCORA2 and TCORA3 pair, can each be a 16-bit register by word access.

The TCORA value is constantly compared with the 8TCNT value. When a match is d corresponding compare match flag A (CMFA) is set to 1 in 8TCSR.

The timer output can be freely controlled by these compare match signals and the setti output select bits 1 and 0 (OS1, OS0) in 8TCSR.

Each TCORA register is initialized to H'FF by a reset and in standby mode.



TCORB0 to TCORB3 are 8-bit readable/writable registers. The TCORB0 and TCORB the TCORB2 and TCORB3 pair, can each be accessed as a 16-bit register by word acce

The TCORB value is constantly compared with the 8TCNT value. When a match is de corresponding compare match flag B (CMFB) is set to 1 in 8TCSR*.

The timer output can be freely controlled by these compare match signals and the settin output/input capture edge select bits 3 and 2 (OIS3, OIS2) in 8TCSR.

When TCORB is used for input capture, it stores the 8TCNT value on detection of an e input capture signal. At this time, the CMFB flag is set to 1 in the corresponding 8TCS The detected edge of the input capture signal is set in 8TCSR.

Each TCORB register is initialized to H'FF by a reset and in standby mode.

Note: * When channel 1 and channel 3 are designated for TCORB input capture, the CM not set by a channel 0 or channel 2 compare match B.

clearing specification, and enables interrupt requests.

8TCR is initialized to H'00 by a reset and in standby mode.

For the timing, see section 10.4, Operation.

Bit 7—Compare Match Interrupt Enable B (CMIEB): Enables or disables the CM request when the CMFB flag is set to 1 in 8TCSR.

Bit 7 CMIEB	Description	
0	CMIB interrupt requested by CMFB is disabled	(
1	CMIB interrupt requested by CMFB is enabled	

Bit 6—Compare Match Interrupt Enable A (CMIEA): Enables or disables the CM request when the CMFA flag is set to 1 in 8TCSR.

Bit 6 CMIEA	Description	
0	CMIA interrupt requested by CMFA is disabled	(
1	CMIA interrupt requested by CMFA is enabled	

Bit 5—Timer Overflow Interrupt Enable (OVIE): Enables or disables the OVI inte request when the OVF flag is set to 1 in 8TCSR.

Bit 5 OVIE	Description	
0	OVI interrupt requested by OVF is disabled	(
1	OVI interrupt requested by OVF is enabled	

Note: When input capture B is set as the 8TCNT1 and 8TCNT3 counter clear source, 8
and 8TCNT2 are not cleared by compare match B.

Bits 2 to 0—Clock Select 2 to 0 (CSK2 to CSK0): These bits select whether the clock
8TCNT is an internal or external clock.

Three internal clocks can be selected, all divided from the system clock (ϕ): ϕ/8, ϕ/64, a
The rising edge of the selected internal clock triggers the count.

When use of an external clock is selected, three types of count can be selected: at the ri
the falling edge, and both rising and falling edges.

When CKS2, CKS1, CKS0 = 1, 0, 0, channels 0 and 1 and channels 2 and 3 are cascad

The incrementing clock source is different when 8TCR0 and 8TCR2 are set, and when
8TCR3 are set.

			Channel 1 (compare match count mode): Count on 8T compare match A[*1]
			Channel 2 (16-bit count mode): Count on 8TCNT3 ove signal[*2]
			Channel 3 (compare match count mode): Count on 8T compare match A[*2]
		1	External clock, counted on rising edge
	1	0	External clock, counted on falling edge
		1	External clock, counted on both rising and falling edge

Notes: 1. If the clock input of channel 0 is the 8TCNT1 overflow signal and that of cha 8TCNT0 compare match signal, no incrementing clock is generated. Do no setting.

2. If the clock input of channel 2 is the 8TCNT3 overflow signal and that of cha 8TCNT2 compare match signal, no incrementing clock is generated. Do no setting.

8TCSR2

Bit	7	6	5	4	3	2	1
	CMFB	CMFA	OVF	—	OIS3	OIS2	OS1
Initial value	0	0	0	1	0	0	0
Read/Write	R/(W)*	R/(W)*	R/(W)*	—	R/W	R/W	R/W

8TCSR1, 8TCSR3

Bit	7	6	5	4	3	2	1
	CMFB	CMFA	OVF	ICE	OIS3	OIS2	OS1
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

Note: * Only 0 can be written to bits 7 to 5, to clear these flags.

The timer control/status registers 8TCSR are 8-bit registers that indicate compare matc
capture and overflow statuses, and control compare match output/input capture edge se

8TCSR2 is initialized to H'10, and 8TCSR0, 8TCSR1, and 8TCSR3 to H'00, by a reset
standby mode.

- The 8TCNT value is transferred to TCORB by an input capture
  TCORB functions as an input capture register

Note: * When bit ICE is set to 1 in 8TCSR1 and 8TCSR3, the CMFB flag is not set whe
TCORB0 or 8TCNT2 = TCORB2.

Bit 6—Compare Match Flag A (CMFA): Status flag that indicates the occurrence o
compare match.

Bit 6 CMFA	Description	
0	[Clearing condition] Read CMFA when CMFA = 1, then write 0 in CMFA	(
1	[Setting condition] 8TCNT = TCORA	

Bit 5—Timer Overflow Flag (OVF): Status flag that indicates that the 8TCNT has o
from H'FF to H'00.

Bit 5 OVF	Description	
0	[Clearing condition] Read OVF when OVF = 1, then write 0 in OVF	(
1	[Setting condition] 8TCNT overflows from H'FF to H'00	

		(ADTRG) input are disabled
1	0	A/D converter start requests by external trigger pin ($\overline{\text{ADTRG}}$) input enabled, and A/D converter start requests by compare match A ar
	1	A/D converter start requests by compare match A are enabled, and converter start requests by external trigger pin ($\overline{\text{ADTRG}}$) input are

Note: * TRGE is bit 7 of the A/D control register (ADCR).

Bit 4—Reserved (In 8TCSR1): This bit is a reserved bit, but can be read and written.

Bit 4—Input Capture Enable (ICE) (In 8TCSR1 and 8TCSR3): Selects the functio TCORB1 and TCORB3.

Bit 4 ICE	Description	
0	TCORB1 and TCORB3 are compare match registers	(I
1	TCORB1 and TCORB3 are input capture registers	

When bit ICE is set to 1 in 8TCSR1 or 8TCSR3, the operation of the TCORA and TC registers in channels 0 to 3 is as shown in the tables below.

TCORA1	Compare match operation	CMFA changed from 0 to 1 in 8TCSR1 by compare match	$TMIO_1$ is dedicated input capture pin	CMIA1 interru generated by match
TCORB1	Input capture operation	CMFB changed from 0 to 1 in 8TCSR1 by input capture	$TMIO_1$ is dedicated input capture pin	CMIB1 interru generated by capture

Table 10.4 Operation of Channels 2 and 3 when Bit ICE is Set to 1 in 8TCSR3 I

Register	Register Function	Status Flag Change	Timer Output Capture Input	Interrupt Re
TCORA2	Compare match operation	CMFA changed from 0 to 1 in 8TCSR2 by compare match	TMO_2 output controllable	CMIA2 interru generated by match
TCORB2	Compare match operation	CMFB not changed from 0 to 1 in 8TCSR2 by compare match	No output from TMO_2	CMIB2 interru not generate match
TCORA3	Compare match operation	CMFA changed from 0 to 1 in 8TCSR3 by compare match	$TMIO_3$ is dedicated input capture pin	CMIA3 interru generated by match
TCORB3	Input capture operation	CMFB changed from 0 to 1 in 8TCSR3 by input capture	$TMIO_3$ is dedicated input capture pin	CMIB3 interru generated by capture

0	0	0	No change when compare match B occurs (In
		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
1	0	0	TCORB input capture on rising edge
		1	TCORB input capture on falling edge
	1	0	TCORB input capture on both rising and falling edges
		1	

- When the compare match register function is used, the timer output priority order is output > 1 output > 0 output.
- If compare match A and B occur simultaneously, the output changes in accordance higher-priority compare match.
- When bits OIS3, OIS2, OS1, and OS0 are all cleared to 0, timer output is disabled.

Bits 1 and 0—Output Select A1 and A0 (OS1, OS0): These bits select the compare n output level.

Bit 1 OS1	Bit 0 OS0	Description
0	0	No change when compare match A occurs (In
	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)

- When the compare match register function is used, the timer output priority order is output > 1 output > 0 output.
- If compare match A and B occur simultaneously, the output changes in accordance higher-priority compare match.
- When bits OIS3, OIS2, OS1, and OS0 are all cleared to 0, timer output is disabled.

Figures 10.4 to 10.7 show the operation in byte read and write accesses to 8TCNT0 ar

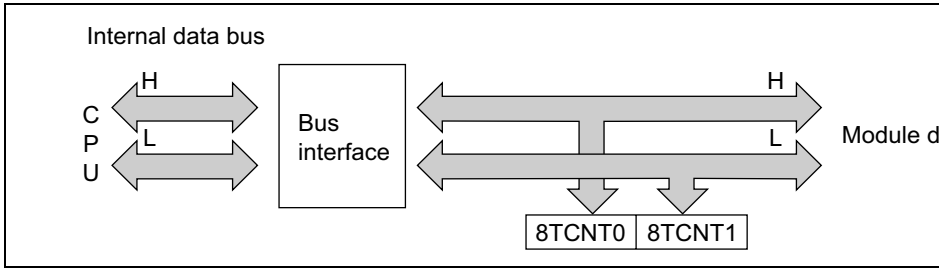


Figure 10.2 8TCNT Access Operation (CPU Writes to 8TCNT, Word

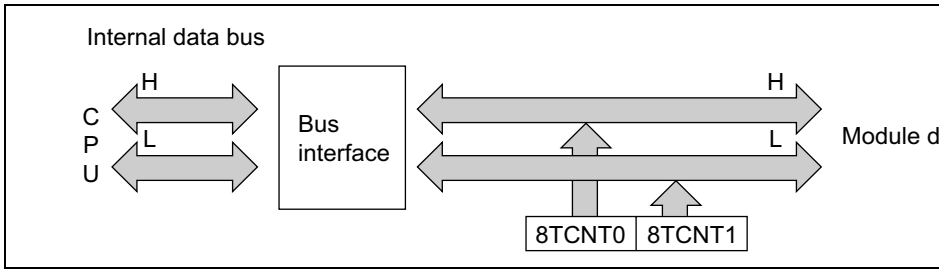


Figure 10.3 8TCNT Access Operation (CPU Reads 8TCNT, Word)

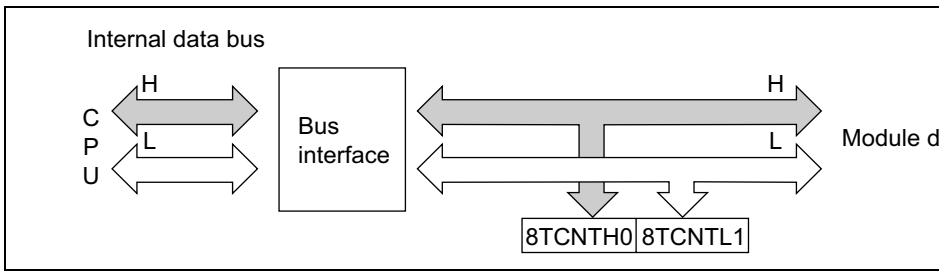


Figure 10.4 8TCNT0 Access Operation (CPU Writes to 8TCNT0, Upper

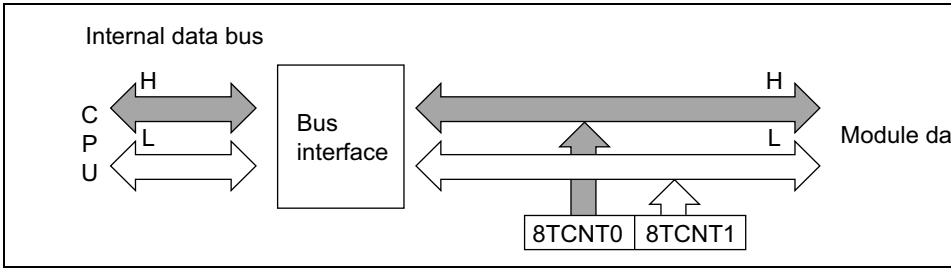


Figure 10.6 8TCNT0 Access Operation (CPU Reads 8TCNT0, Upper By

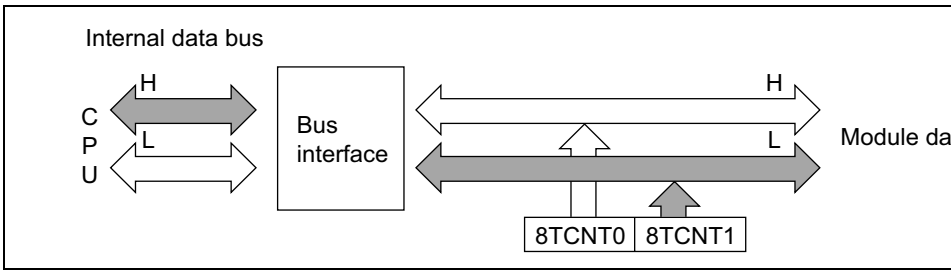
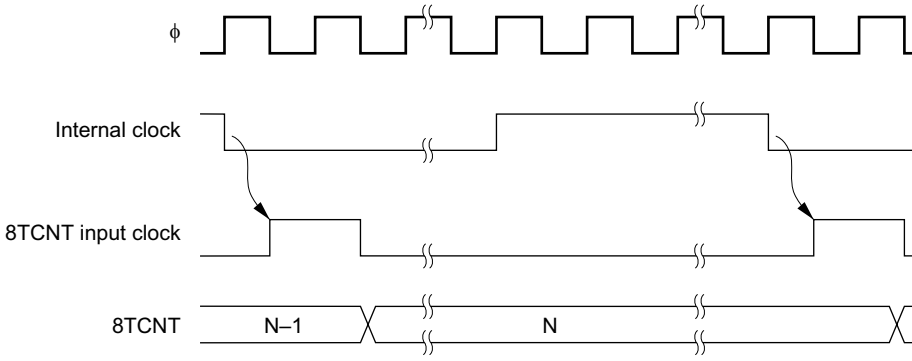


Figure 10.7 8TCNT1 Access Operation (CPU Reads 8TCNT1, Lower By



Note: Even if the same internal clock is selected for the 16-bit timer and the 8-bit timer, the same will not be performed since the incrementing edge is different in each case.

Figure 10.8 Count Timing for Internal Clock Input

External Clock: Three incrementation methods can be selected by setting bits CKS2 8TCR: on the rising edge, the falling edge, and both rising and falling edges.

The pulse width of the external clock signal must be at least 1.5 system clocks when a is selected, and at least 2.5 system clocks when both edges are selected. Shorter pulse counted correctly.

Figure 10.9 shows the timing for incrementation on both edges of the external clock si

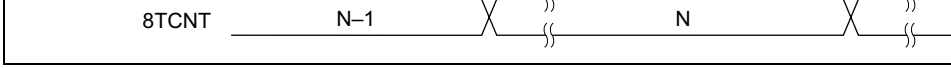


Figure 10.9 Count Timing for External Clock Input (Both-Edge Detectio

10.4.2 Compare Match Timing

Timer Output Timing: When compare match A or B occurs, the timer output is as spe
the OIS3, OIS2, OS1, and OS0 bits in 8TCSR (unchanged, 0 output, 1 output, or toggle

Figure 10.10 shows the timing when the output is set to toggle on compare match A.

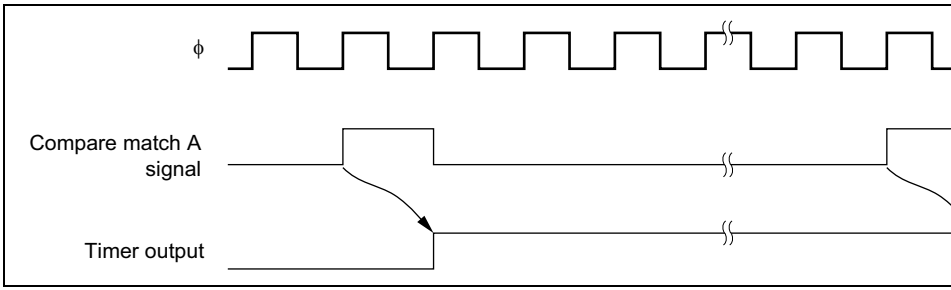


Figure 10.10 Timing of Timer Output

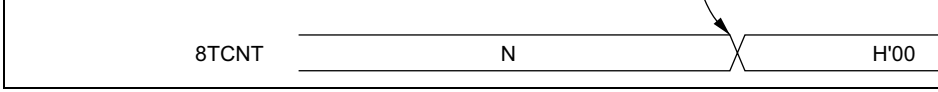


Figure 10.11 Timing of Clear by Compare Match

Clear by Input Capture: Depending on the setting of the CCLR1 and CCLR0 bits in
8TCNT can be cleared when input capture B occurs. Figure 10.12 shows the timing o
operation.

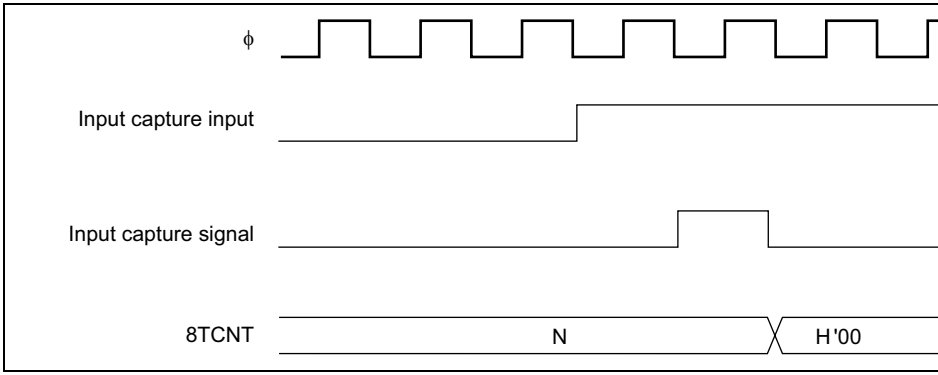


Figure 10.12 Timing of Clear by Input Capture

10.4.3 Input Capture Signal Timing

Input capture on the rising edge, falling edge, or both edges can be selected by setting

Figure 10.13 shows the timing when the rising edge is selected.

The pulse width of the input capture input signal must be at least 1.5 system clocks wh
edge is selected, and at least 2.5 system clocks when both edges are selected.

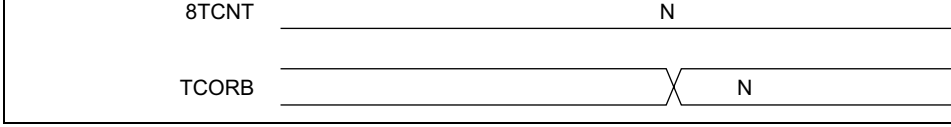


Figure 10.13 Timing of Input Capture Input Signal

10.4.4 Timing of Status Flag Setting

Timing of CMFA/CMFB Flag Setting when Compare Match Occurs: The CMFA a flags in 8TCSR are set to 1 by the compare match signal output when the TCORA or T 8TCNT values match. The compare match signal is generated in the last state of the m the matched 8TCNT count value is updated). Therefore, after the 8TCNT and TCORA TCORB values match, the compare match signal is not generated until an incrementing pulse signal is generated. Figure 10.14 shows the timing in this case.

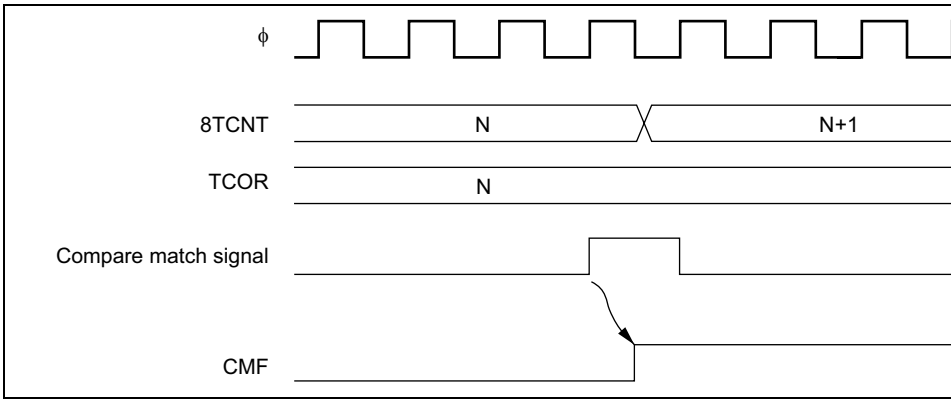


Figure 10.14 CMF Flag Setting Timing when Compare Match Occurs

Timing of CMFB Flag Setting when Input Capture Occurs: On generation of an inp signal, the CMFB flag is set to 1 and at the same time the 8TCNT value is transferred t Figure 10.15 shows the timing in this case.

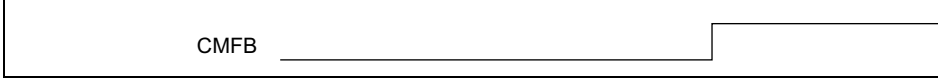


Figure 10.15 CMFB Flag Setting Timing when Input Capture Occur

Timing of Overflow Flag (OVF) Setting: The OVF flag in 8TCSR is set to 1 by the signal generated when 8TCNT overflows (from H'FF to H'00). Figure 10.16 shows th this case.

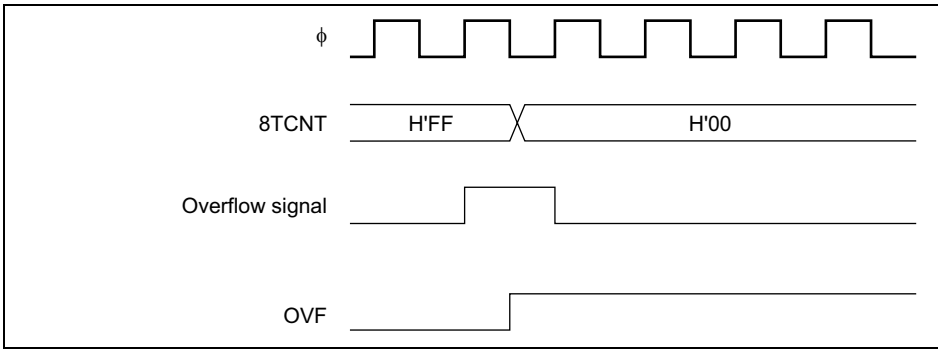


Figure 10.16 Timing of OVF Setting

10.4.5 Operation with Cascaded Connection

If bits CKS2 to CKS0 are set to (100) in either 8TCR0 or 8TCR1, the 8-bit timers of c and 1 are cascaded. With this configuration, the two timers can be used as a single 16 (16-bit timer mode), or channel 0 8-bit timer compare matches can be counted in chan (compare match count mode). Similarly, if bits CKS2 to CKS0 are set to (100) in eith 8TCR3, the 8-bit timers of channels 2 and 3 are cascaded. With this configuration, th can be used as a single 16-bit timer (16-bit timer mode),or channel 2 8-bit timer comp can be counted in channel 3 (compare match count mode). In this case, the timer oper below.

- TMO_0 pin output control by bits OIS3, OIS2, OS1, and OS0 in 8TCSR0 is in accordance with the 16-bit compare match conditions.
- $TMIO_1$ pin output control by bits OIS3, OIS2, OS1, and OS0 in 8TCSR1 is i accordance with the lower 8-bit compare match conditions.

— Setting when Input Capture Occurs

- The CMFB flag is set to 1 in 8TCSR0 and 8TCSR1 when the ICE bit is 1 in and input capture occurs.
- $TMIO_1$ pin input capture input signal edge detection is selected by bits OIS3 in 8TCSR0.

— Counter Clear Specification

- If counter clear on compare match or input capture has been selected by the and CCLR0 bits in 8TCR0, the 16-bit counter (both 8TCNT0 and 8TCNT1)
- The settings of the CCLR1 and CCLR0 bits in 8TCR1 are ignored. The low cannot be cleared independently.

— OVF Flag Operation

- The OVF flag is set to 1 in 8TCSR0 when the 16-bit counter (8TCNT0 and 8 overflows (from H'FFFF to H'0000).
- The OVF flag is set to 1 in 8TCSR1 when the 8-bit counter (8TCNT1) overf H'FF to H'00).

- Channels 2 and 3:

  When bits CKS2 to CKS0 are set to (100) in 8TCR2, the timer functions as a single timer with channel 2 occupying the upper 8 bits and channel 3 occupying the lower

  — Setting when Compare Match Occurs

  - The CMFA or CMFB flag is set to 1 in 8TCSR2 when a 16-bit compare mat
  - The CMFA or CMFB flag is set to 1 in 8TCSR3 when a lower 8-bit compar occurs.
  - TMO_2 pin output control by bits OIS3, OIS2, OS1, and OS0 in 8TCSR2 is in accordance with the 16-bit compare match conditions.
  - $TMIO_3$ pin output control by bits OIS3, OIS2, OS1, and OS0 in 8TCSR3 is i accordance with the lower 8-bit compare match conditions.

- The settings of the CCLR1 and CCLR0 bits in 8TCR3 are ignored. The lo
  cannot be cleared independently.

— OVF Flag Operation

- The OVF flag is set to 1 in 8TCSR2 when the 16-bit counter (8TCNT2 and
  overflows (from H'FFFF to H'0000).
- The OVF flag is set to 1 in 8TCSR3 when the 8-bit counter (8TCNT3) ove
  H'FF to H'00).

Compare Match Count Mode

- Channels 0 and 1:

  When bits CKS2 to CKS0 are set to (100) in 8TCR1, 8TCNT1 counts channel 0 c
  match A events.

  CMF flag setting, interrupt generation, TMO pin output, counter clearing, and so o
  accordance with the settings for each channel.

  Note: When bit ICE = 1 in 8TCSR1, the compare match register function of TC
  channel 0 cannot be used.

- Channels 2 and 3:

  When bits CKS2 to CKS0 are set to (100) in 8TCR3, 8TCNT3 counts channel 2 c
  match A events.

  CMF flag setting, interrupt generation, TMO pin output, counter clearing, and so o
  accordance with the settings for each channel.

  Note: When bit ICE = 1 in 8TCSR3, the compare match register function of TC
  channel 2 cannot be used.

Caution

Do not set 16-bit counter mode and compare match count mode simultaneously withi
group, as the 8TCNT input clock will not be generated and the counters will not opera

— Set TCORB1 as an 8-bit input capture register with the ICE bit in 8TCSR1.
— Select rising edge, falling edge, or both edges as the input edge(s) for the input c signal ($TMIO_1$) with bits OIS3 and OIS2 in 8TCSR1.
— Select the input clock with bits CKS2 to CKS0 in 8TCR1, and start the 8TCNT

- Channel 3:
  — Set TCORB3 as an 8-bit input capture register with the ICE bit in 8TCSR3.
  — Select rising edge, falling edge, or both edges as the input edge(s) for the input c signal ($TMIO_3$) with bits OIS3 and OIS2 in 8TCSR3.
  — Select the input clock with bits CKS2 to CKS0 in 8TCR3, and start the 8TCNT

Note: When TCORB1 in channel 1 is used for input capture, TCORB0 in channel 0 c used as a compare match register.
Similarly, when TCORB3 in channel 3 is used for input capture, TCORB2 in c cannot be used as a compare match register.

Setting Input Capture Operation in 16-Bit Count Mode

- Channels 0 and 1:
  — In 16-bit count mode, TCORB0 and TCORB1 function as a 16-bit input capture when the ICE bit is set to 1 in 8TCSR1.
  — Select rising edge, falling edge, or both edges as the input edge(s) for the input c signal ($TMIO_1$) with bits OIS3 and OIS2 in 8TCSR0. (In 16-bit count mode, th bits OIS3 and OIS2 in 8TCSR1 are ignored.)
  — Select the input clock with bits CKS2 to CKS0 in 8TCR1, and start the 8TCNT
- Channels 2 and 3:
  — In 16-bit count mode, TCORB2 and TCORB3 function as a 16-bit input capture when the ICE bit is set to 1 in 8TCSR3.
  — Select rising edge, falling edge, or both edges as the input edge(s) for the input c signal ($TMIO_3$) with bits OIS3 and OIS2 in 8TCSR2. (In 16-bit count mode, th bits OIS3 and OIS2 in 8TCSR3 are ignored.)
  — Select the input clock with bits CKS2 to CKS0 in 8TCR3, and start the 8TCNT

Table 10.5 Types of 8-Bit Timer Interrupt Sources and Priority Order

Interrupt Source	Description	Priority
CMIA	Interrupt by CMFA	High
CMIB	Interrupt by CMFB	↕
TOVI	Interrupt by OVF	Low

For compare match interrupts CMIA1/CMIB1 and CMIA3/CMIB3 and the overflow i
(TOVI0/TOVI1 and TOVI2/TOVI3), one vector is shared by two interrupts.

Table 10.6 lists the interrupt sources.

Table 10.6 8-Bit Timer Interrupt Sources

Channel	Interrupt Source	Description
0	CMIA0	TCORA0 compare match
	CMIB0	TCORB0 compare match/input capture
1	CMIA1/CMIB1	TCORA1 compare match, or TCORB1 compare m capture
0, 1	TOVI0/TOVI1	Counter 0 or counter 1 overflow
2	CMIA2	TCORA2 compare match
	CMIB2	TCORB2 compare match/input capture
3	CMIA3/CMIB3	TCORA3 compare match, or TCORB3 compare m capture
2, 3	TOVI2/TOVI3	Counter 2 or counter 3 overflow

10.6 8-Bit Timer Application Example

Figure 10.17 shows how the 8-bit timer module can be used to output pulses with any d
cycle. The settings for this example are as follows:

- Clear the CCLR1 bit to 0 and set the CCLR0 bit to 1 in 8TCR so that 8TCNT is cle
  TCORA compare match.
- Set bits OIS3, OIS2, OS1, and OS0 to (0110) in 8TCSR so that 1 is output on a TC
  compare match and 0 is output on a TCORB compare match.

The above settings enable a waveform with the cycle determined by TCORA and the p
detected by TCORB to be output without software intervention.

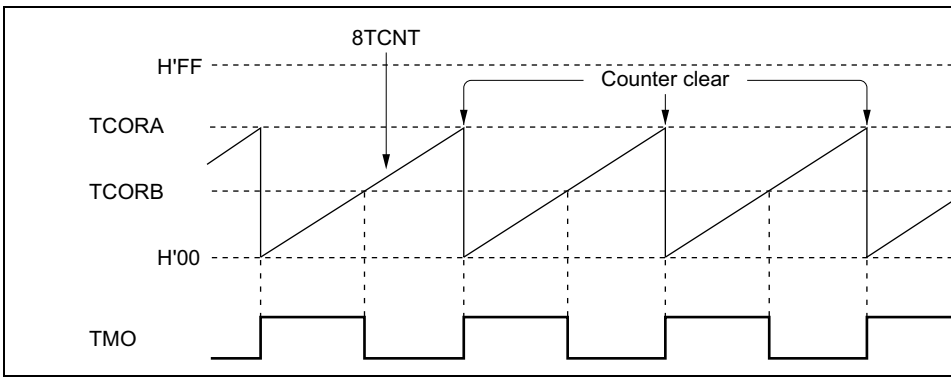


Figure 10.17 Example of Pulse Output

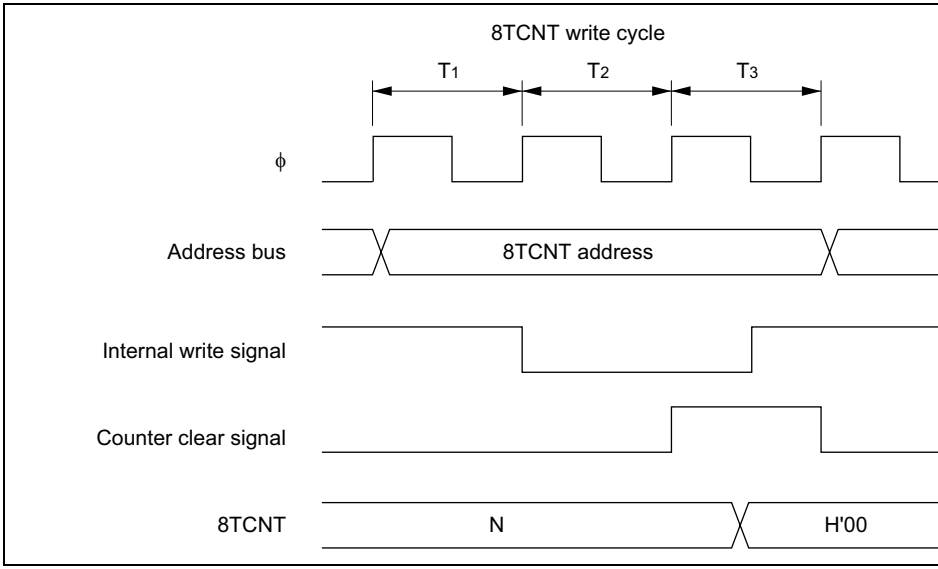


Figure 10.18 Contention between 8TCNT Write and Clear

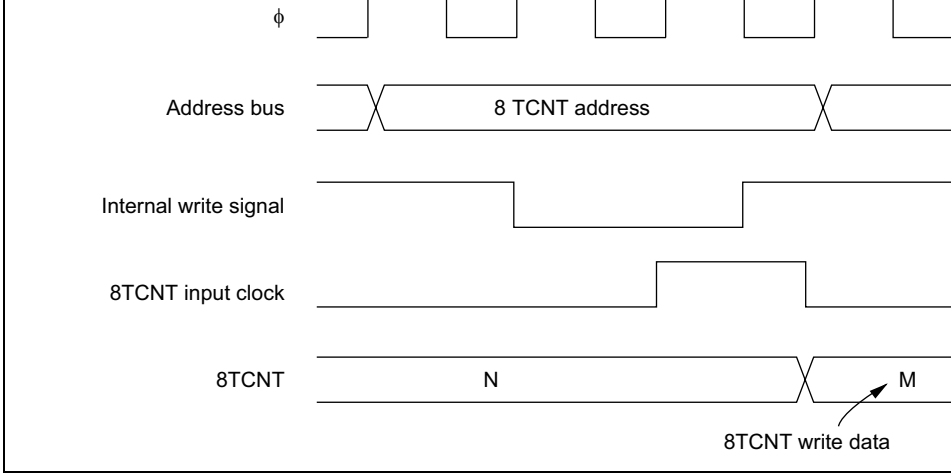


Figure 10.19 Contention between 8TCNT Write and Increment

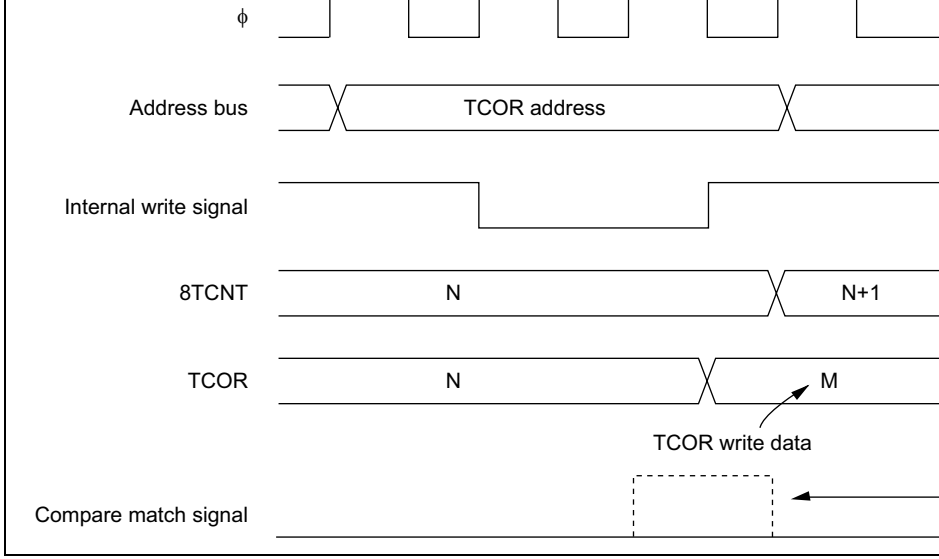


Figure 10.20 Contention between TCOR Write and Compare Match

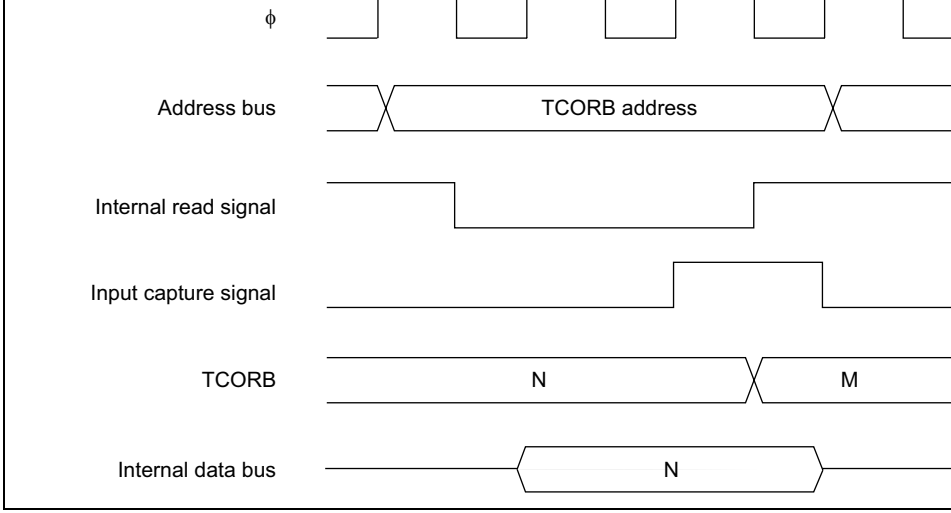


Figure 10.21 Contention between TCOR Read and Input Capture

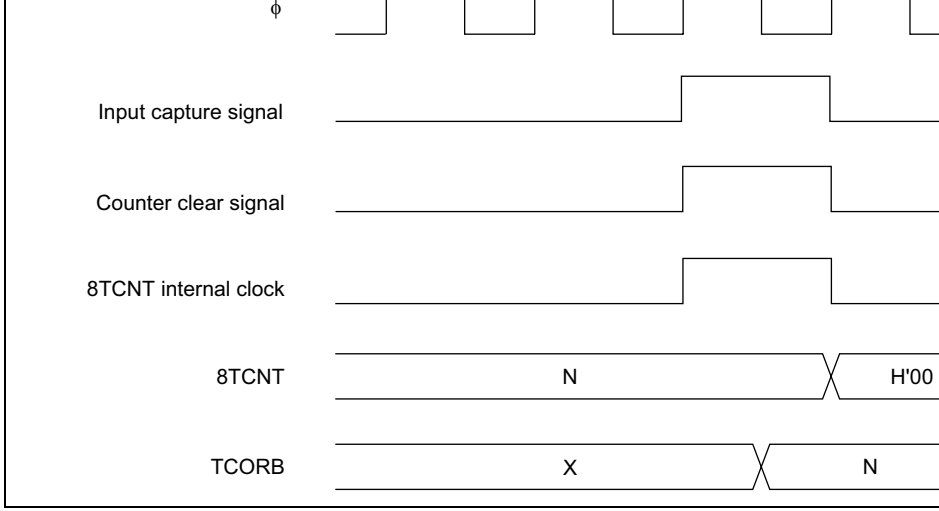


Figure 10.22 Contention between Counter Clearing by Input Capture and C Increment

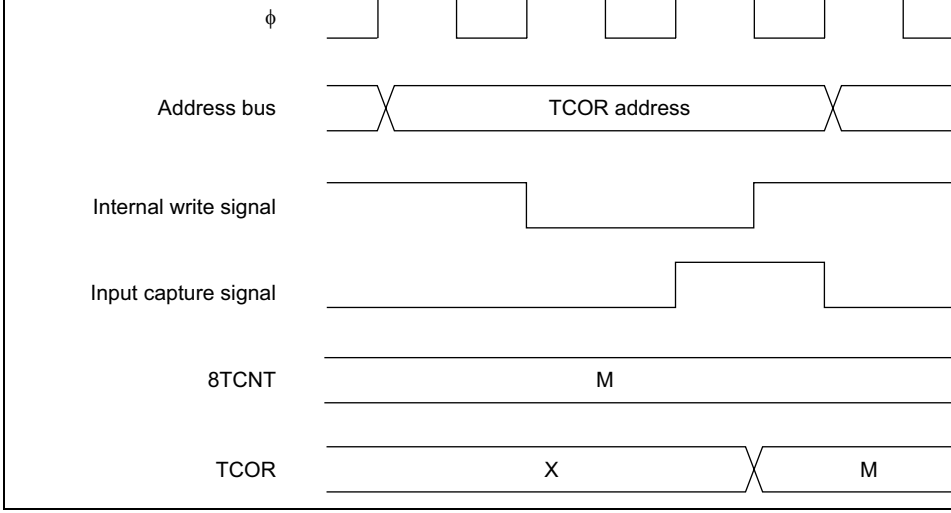


Figure 10.23 Contention between TCOR Write and Input Capture

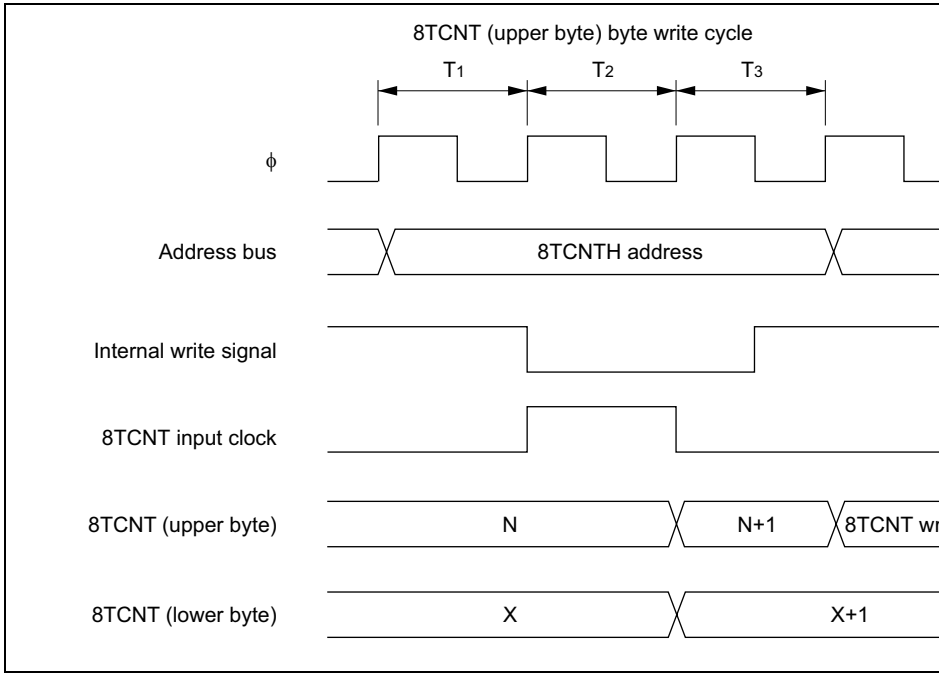


Figure 10.24 Contention between 8TCNT Byte Write and Increment in 16-Bit C

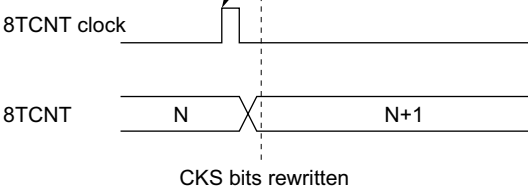
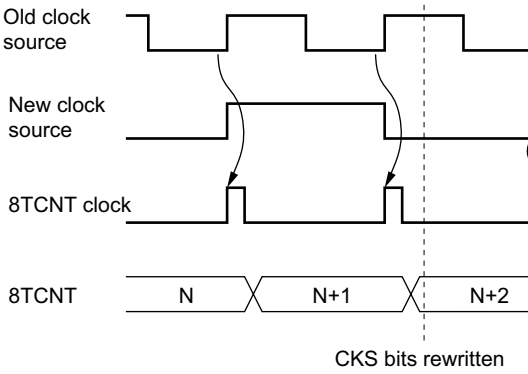
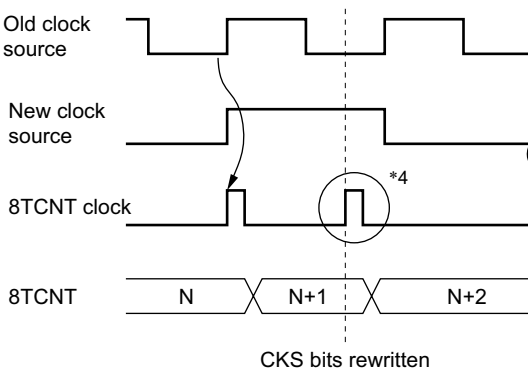
Toggle output	High
1 output	
0 output	
No change	Low

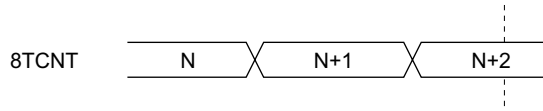
10.7.9 8TCNT Operation and Internal Clock Source Switchover

Switching internal clock sources may cause 8TCNT to increment, depending on the sw
timing. Table 10.8 shows the relation between the time of the switchover (by writing t
and CKS0) and the operation of 8TCNT.

The 8TCNT input clock is generated from the internal clock source by detecting the ris
the internal clock. If a switchover is made from a low clock source to a high clock sou
case no. 3 in table 10.8, the switchover will be regarded as a falling edge, a 8TCNT clo
will be generated, and 8TCNT will be incremented.

8TCNT may also be incremented when switching between internal and external clocks.

		
2	High → low switchover*2	
3	Low → high switchover*3	



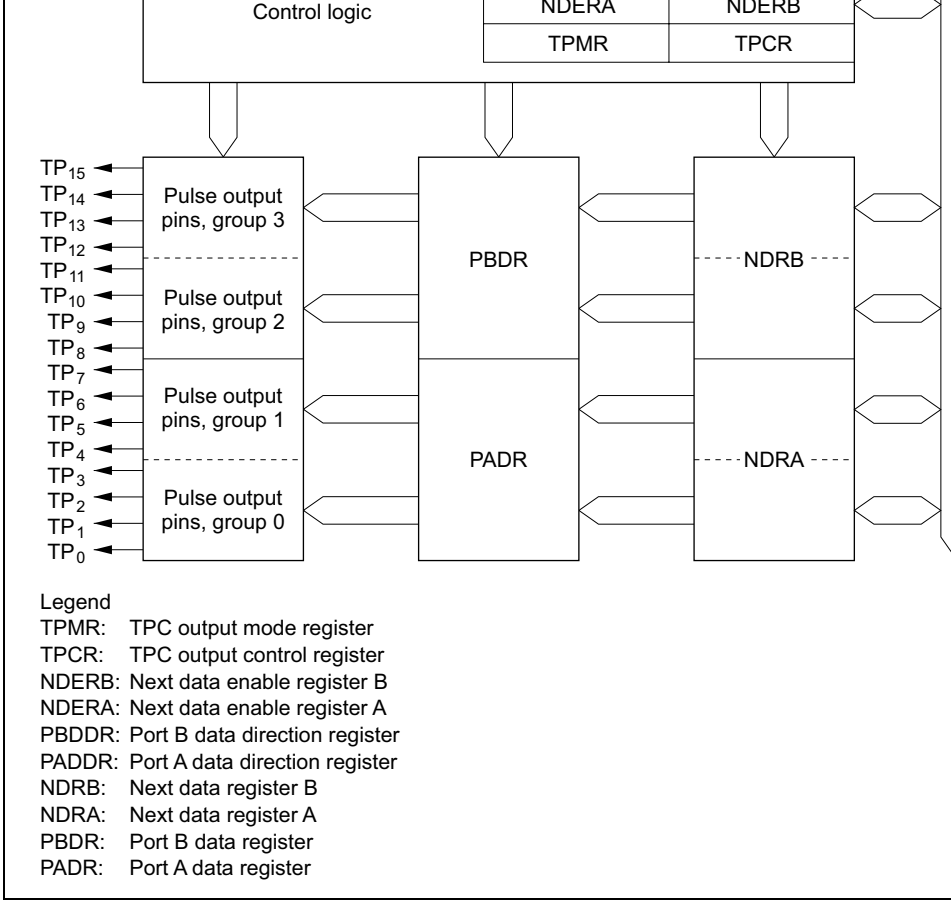
Notes: 1. Including switchovers from the high level to the halted state, and from the ha to the high level.
2. Including switchover from the halted state to the low level.
3. Including switchover from the low level to the halted state.
4. The switchover is regarded as a rising edge, causing 8TCNT to increment.

11.1.1 Features

TPC features are listed below.

- 16-bit output data
  Maximum 16-bit data can be output. TPC output can be enabled on a bit-by-bit ba
- 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
  16-bit timer channels.
- Non-overlap mode
  A non-overlap margin can be provided between pulse outputs.
- Can operate together with the DMA controller (DMAC)

The compare-match signals selected as trigger signals can activate the DMAC for seq
output of data without CPU intervention.



Legend
TPMR: TPC output mode register
TPCR: TPC output control register
NDERB: Next data enable register B
NDERA: Next data enable register A
PBDDR: Port B data direction register
PADDR: Port A data direction register
NDRB: Next data register B
NDRA: Next data register A
PBDR: Port B data register
PADR: Port A data register

Figure 11.1 TPC Block Diagram

TPC output 2	TP_2	Output	
TPC output 3	TP_3	Output	
TPC output 4	TP_4	Output	Group 1 pulse outp
TPC output 5	TP_5	Output	
TPC output 6	TP_6	Output	
TPC output 7	TP_7	Output	
TPC output 8	TP_8	Output	Group 2 pulse outp
TPC output 9	TP_9	Output	
TPC output 10	TP_{10}	Output	
TPC output 11	TP_{11}	Output	
TPC output 12	TP_{12}	Output	Group 3 pulse outp
TPC output 13	TP_{13}	Output	
TPC output 14	TP_{14}	Output	
TPC output 15	TP_{15}	Output	

H'EE00A	Port B data direction register	PBDDR	W	H'00
H'FFFDA	Port B data register	PBDR	R/(W)*2	H'00
H'FFFA0	TPC output mode register	TPMR	R/W	H'F0
H'FFFA1	TPC output control register	TPCR	R/W	H'FF
H'FFFA2	Next data enable register B	NDERB	R/W	H'00
H'FFFA3	Next data enable register A	NDERA	R/W	H'00
H'FFFA5/ H'FFFA7*3	Next data register A	NDRA	R/W	H'00
H'FFFA4/ H'FFFA6*3	Next data register B	NDRB	R/W	H'00

Notes: 1. Lower 20 bits of the address in advanced mode.

2. Bits used for TPC output cannot be written.

3. The NDRA address is H'FFFA5 when the same output trigger is selected for output groups 0 and 1 by settings in TPCR. When the output triggers are diff NDRA address is H'FFFA7 for group 0 and H'FFFA5 for group 1. Similarly, t of NDRB is H'FFFA4 when the same output trigger is selected for TPC outpu and 3 by settings in TPCR. When the output triggers are different, the NDRB H'FFFA6 for group 2 and H'FFFA4 for group 3.

Initial value	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W

Port A data direction 7 to 0
These bits select input or output for port A pins

Port A is multiplexed with pins TP_7 to TP_0. Bits corresponding to pins used for TPC o be set to 1. For further information about PADDR, see section 8.11, Port A.

11.2.2 Port A Data Register (PADR)

PADR is an 8-bit readable/writable register that stores TPC output data for groups 0 a these TPC output groups are used.

Bit	7	6	5	4	3	2	1
	PA_7	PA_6	PA_5	PA_4	PA_3	PA_2	PA_1
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)*

Port A data 7 to 0
These bits store output data for TPC output groups 0 and 1

Note: * Bits selected for TPC output by NDERA settings become read-only bits.

For further information about PADR, see section 8.11, Port A.

Port B direction 7 to 0
These bits select input or output for port B pins

Port B is multiplexed with pins TP_{15} to TP_8. Bits corresponding to pins used for TPC ou be set to 1. For further information about PBDDR, see section 8.12, Port B.

11.2.4 Port B Data Register (PBDR)

PBDR is an 8-bit readable/writable register that stores TPC output data for groups 2 an these TPC output groups are used.

Bit	7	6	5	4	3	2	1	
	PB_7	PB_6	PB_5	PB_4	PB_3	PB_2	PB_1	
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)*	

Port B data 7 to 0
These bits store output data for TPC output groups 2 and 3

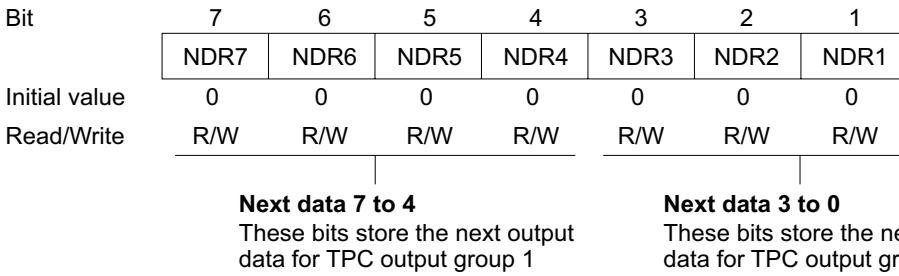
Note: * Bits selected for TPC output by NDERB settings become read-only bits.

For further information about PBDR, see section 8.12, Port B.

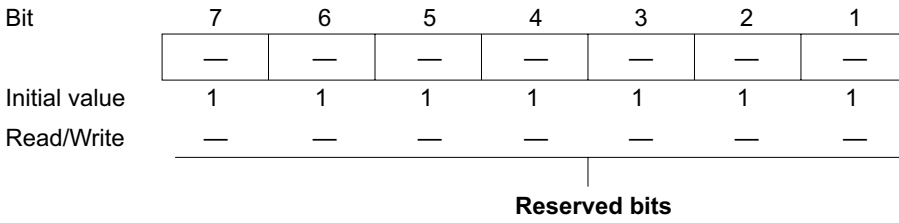
software standby mode.

Same Trigger for TPC Output Groups 0 and 1: If TPC output groups 0 and 1 are tr
the same compare match event, the NDRA address is H'FFFA5. The upper 4 bits belo
1 and the lower 4 bits to group 0. Address H'FFFA7 consists entirely of reserved bits t
be modified and always read 1.

Address H'FFFA5



Address H'FFFA7



Initial value	0	0	0	0	1	1	1
Read/Write	R/W	R/W	R/W	R/W	—	—	—

Next data 7 to 4
These bits store the next output data for TPC output group 1

Reserved bits

Address H'FFFA7

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

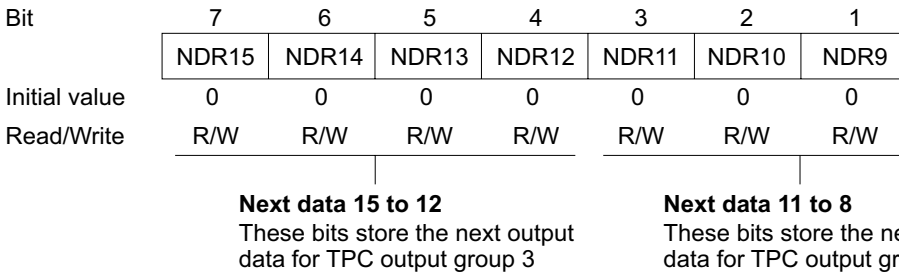
Reserved bits

Next data 3 to 0
These bits store the ne
data for TPC output gr

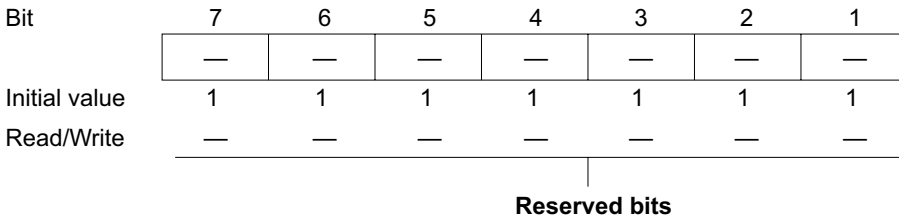
software standby mode.

Same Trigger for TPC Output Groups 2 and 3: If TPC output groups 2 and 3 are tr
the same compare match event, the NDRB address is H'FFFA4. The upper 4 bits belo
3 and the lower 4 bits to group 2. Address H'FFFA6 consists entirely of reserved bits t
be modified and always read 1.

Address H'FFFA4



Address H'FFFA6



Bit	7	6	5	4	3	2	1	
	—	—	—	—	NDR11	NDR10	NDR9	
Initial value	1	1	1	1	0	0	0	
Read/Write	—	—	—	—	R/W	R/W	R/W	

Reserved bits

Next data 11 to 8
These bits store the ne
data for TPC output gr

Address H'FFFA6

Bit	7	6	5	4	3	2	1	
	—	—	—	—	NDR11	NDR10	NDR9	
Initial value	1	1	1	1	0	0	0	
Read/Write	—	—	—	—	R/W	R/W	R/W	

Reserved bits

Next data 11 to 8
These bits store the ne
data for TPC output gr

Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Next data enable 7 to 0
These bits enable or disable
TPC output groups 1 and 0

If a bit is enabled for TPC output by NDERA, then when the 16-bit timer compare ma
selected in the TPC output control register (TPCR) occurs, the NDRA value is automa
transferred to the corresponding PADR bit, updating the output value. If TPC output i
the bit value is not transferred from NDRA to PADR and the output value does not ch

NDERA is initialized to H'00 by a reset and in hardware standby mode. It is not initia
software standby mode.

Bits 7 to 0—Next Data Enable 7 to 0 (NDER7 to NDER0): These bits enable or dis
output groups 1 and 0 (TP_7 to TP_0) on a bit-by-bit basis.

Bits 7 to 0 NDER7 to NDER0	Description
0	TPC outputs TP_7 to TP_0 are disabled (NDR7 to NDR0 are not transferred to PA_7 to PA_0)
1	TPC outputs TP_7 to TP_0 are enabled (NDR7 to NDR0 are transferred to PA_7 to PA_0)

Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W

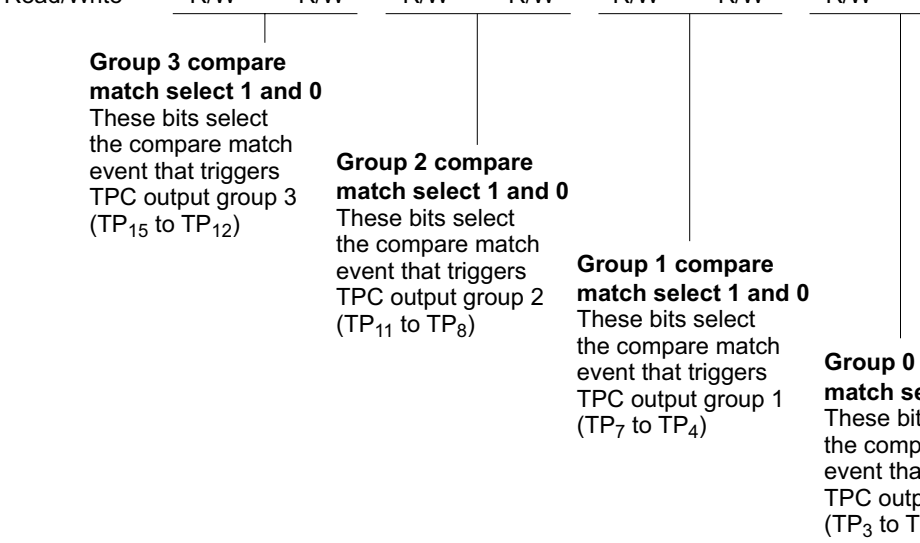
Next data enable 15 to 8
These bits enable or disable
TPC output groups 3 and 2

If a bit is enabled for TPC output by NDERB, then when the 16-bit timer compare mat selected in the TPC output control register (TPCR) occurs, the NDRB value is automat transferred to the corresponding PBDR bit, updating the output value. If TPC output is the bit value is not transferred from NDRB to PBDR and the output value does not cha

NDERB is initialized to H'00 by a reset and in hardware standby mode. It is not initiali software standby mode.

Bits 7 to 0—Next Data Enable 15 to 8 (NDER15 to NDER8): These bits enable or di output groups 3 and 2 (TP_{15} to TP_8) on a bit-by-bit basis.

Bits 7 to 0 NDER15 to NDER8	Description
0	TPC outputs TP_{15} to TP_8 are disabled (NDR15 to NDR8 are not transferred to PB_7 to PB_0)
1	TPC outputs TP_{15} to TP_8 are enabled (NDR15 to NDR8 are transferred to PB_7 to PB_0)



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

Bits 7 and 6—Group 3 Compare Match Select 1 and 0 (G3CMS1, G3CMS0): The select the compare match event that triggers TPC output group 3 (TP_{15} to TP_{12}).

Bit 7 G3CMS1	Bit 6 G3CMS0	Description
0	0	TPC output group 3 (TP_{15} to TP_{12}) is triggered by compare m timer channel 0
	1	TPC output group 3 (TP_{15} to TP_{12}) is triggered by compare m timer channel 1
1	0	TPC output group 3 (TP_{15} to TP_{12}) is triggered by compare m timer channel 2
	1	TPC output group 3 (TP_{15} to TP_{12}) is triggered by compare match in 16-bit timer channel 2

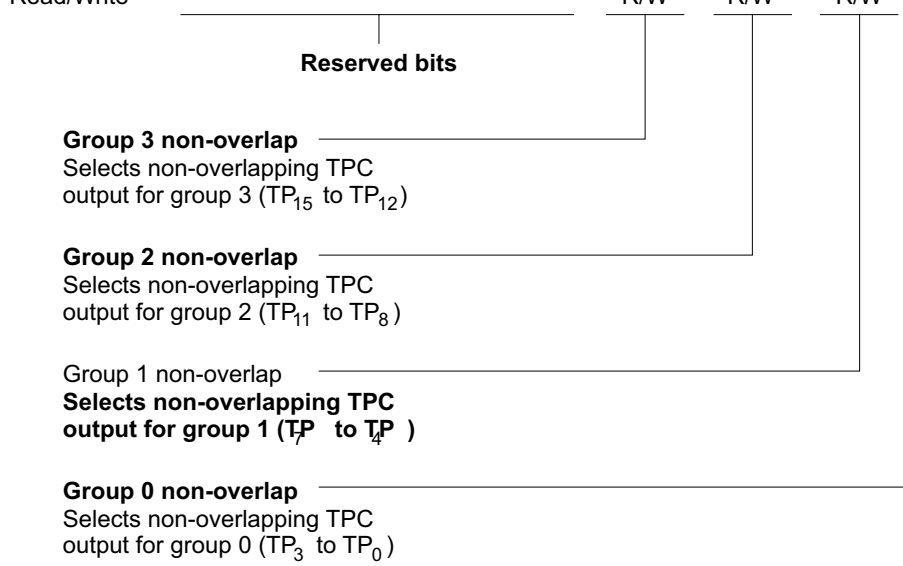
1	0	TPC output group 2 (TP_{11} to TP_8) is triggered by compare mat timer channel 2
	1	TPC output group 2 (TP_{11} to TP_8) is triggered by compare match in 16-bit timer channel 2 (

Bits 3 and 2—Group 1 Compare Match Select 1 and 0 (G1CMS1, G1CMS0): Thes select the compare match event that triggers TPC output group 1 (TP_7 to TP_4).

Bit 3 G1CMS1	Bit 2 G1CMS0	Description
0	0	TPC output group 1 (TP_7 to TP_4) is triggered by compare matc timer channel 0
	1	TPC output group 1 (TP_7 to TP_4) is triggered by compare matc timer channel 1
1	0	TPC output group 1 (TP_7 to TP_4) is triggered by compare matc timer channel 2
	1	TPC output group 1 (TP_7 to TP_4) is triggered by compare match in 16-bit timer channel 2 (

Bits 1 and 0—Group 0 Compare Match Select 1 and 0 (G0CMS1, G0CMS0): Thes select the compare match event that triggers TPC output group 0 (TP_3 to TP_0).

Bit 1 G0CMS1	Bit 0 G0CMS0	Description
0	0	TPC output group 0 (TP_3 to TP_0) is triggered by compare matc timer channel 0
	1	TPC output group 0 (TP_3 to TP_0) is triggered by compare matc timer channel 1
1	0	TPC output group 0 (TP_3 to TP_0) is triggered by compare matc timer channel 2
	1	TPC output group 0 (TP_3 to TP_0) is triggered by compare match in 16-bit timer channel 2 (



The output trigger period of a non-overlapping TPC output waveform is set in general (GRB) in the 16-bit timer channel selected for output triggering. The non-overlap mar general register A (GRA). The output values change at compare match A and B. For d section 11.3.4, Non-Overlapping TPC Output.

TPMR is initialized to H'F0 by a reset and in hardware standby mode. It is not initializ software standby mode.

Bits 7 to 4—Reserved: These bits cannot be modified and are always read as 1.

Bit 2—Group 2 Non-Overlap (G2NOV): Selects normal or non-overlapping TPC out group 2 (TP_{11} to TP_8).

Bit 2 G2NOV	Description	
0	Normal TPC output in group 2 (output values change at compare match A in the selected 16-bit timer channel)	(
1	Non-overlapping TPC output in group 2 (independent 1 and 0 output at compare match A and B in the selected 16-bit timer channel)	

Bit 1—Group 1 Non-Overlap (G1NOV): Selects normal or non-overlapping TPC out group 1 (TP_7 to TP_4).

Bit 1 G1NOV	Description	
0	Normal TPC output in group 1 (output values change at compare match A in the selected 16-bit timer channel)	(
1	Non-overlapping TPC output in group 1 (independent 1 and 0 output at compare match A and B in the selected 16-bit timer channel)	

Bit 0—Group 0 Non-Overlap (G0NOV): Selects normal or non-overlapping TPC out group 0 (TP_3 to TP_0).

Bit 0 G0NOV	Description	
0	Normal TPC output in group 0 (output values change at compare match A in the selected 16-bit timer channel)	(
1	Non-overlapping TPC output in group 0 (independent 1 and 0 output at compare match A and B in the selected 16-bit timer channel)	

Figure 11.2 illustrates the TPC output operation. Table 11.3 summarizes the TPC oper conditions.

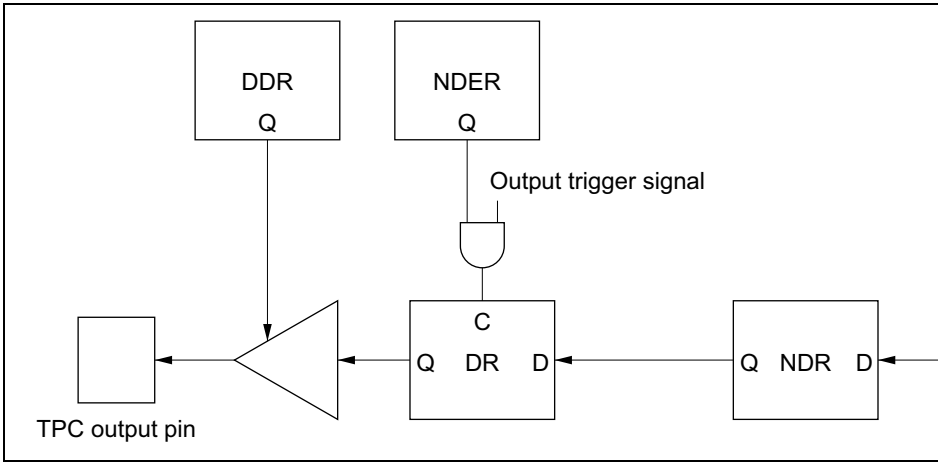


Figure 11.2 TPC Output Operation

Table 11.3 TPC Operating Conditions

NDER	DDR	Pin Function
0	0	Generic input port
	1	Generic output port
1	0	Generic input port (but the DR bit is a read-only bit, and whe match occurs, the NDR bit value is transferred to the DR bit)
	1	TPC pulse output

Sequential output of up to 16-bit patterns is possible by writing new output data to ND NDRB before the next compare match. For information on non-overlapping operation section 11.3.4, Non-Overlapping TPC Output.

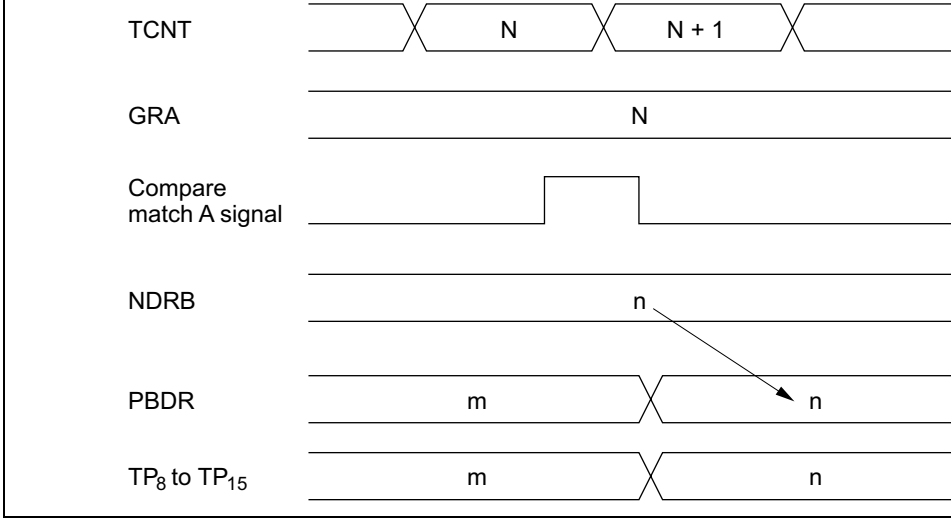
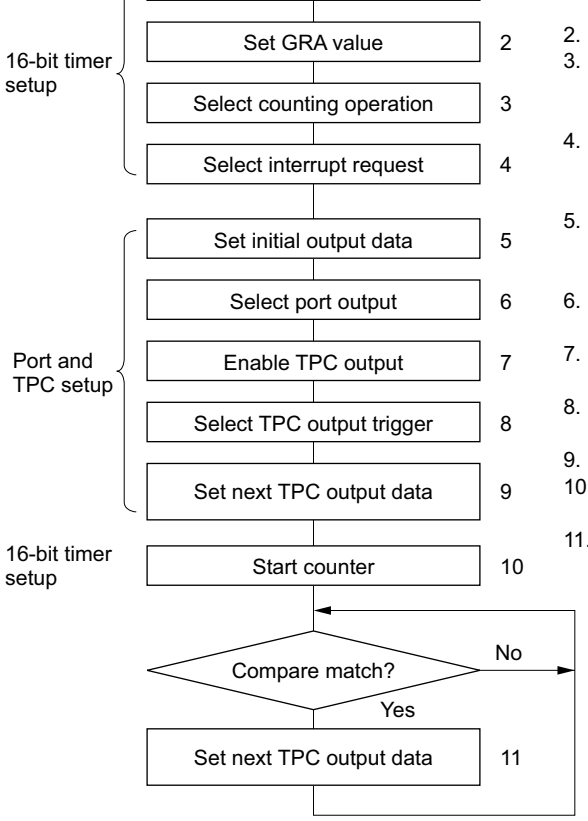
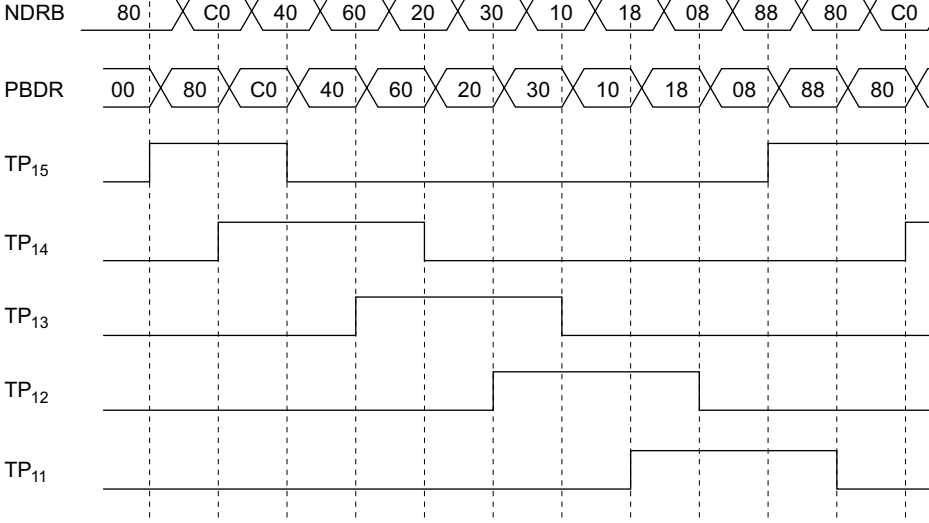


Figure 11.3 Timing of Transfer of Next Data Register Contents and Output (E



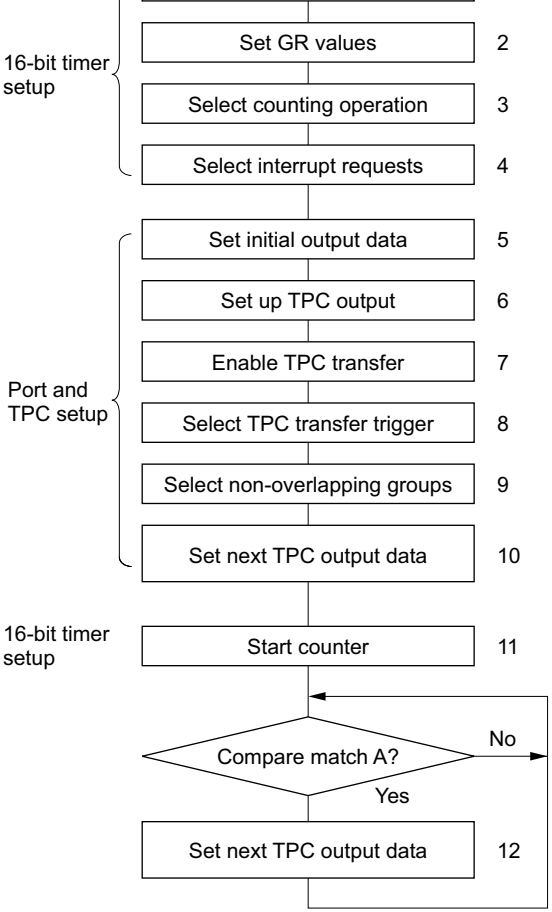
register (with output inhibited).
2. Set the TPC output trigger period
3. Select the counter clock source w
TPSC2 to TPSC0 in TCR. Select
clear source with bits CCLR1 and
4. Enable the IMFA interrupt in TIER
The DMAC can also be set up to
data to the next data register.
5. Set the initial output values in the
of the input/output port pins to be
TPC output.
6. Set the DDR bits of the input/outp
pins to be used for TPC output to
7. Set the NDER bits of the pins to
TPC output to 1.
8. Select the 16-bit timer compare m
to be used as the TPC output trig
9. Set the next TPC output values in
10. Set the STR bit to 1 in TSTR to st
timer counter.
11. At each IMFA interrupt, set the ne
values in the NDR bits.

Figure 11.4 Setup Procedure for Normal TPC Output (Example)



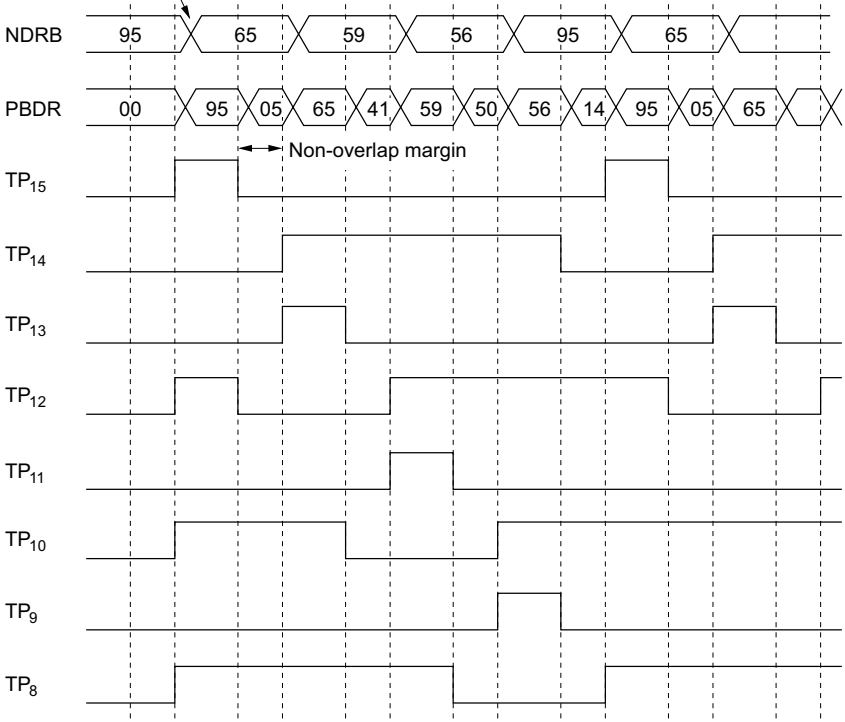
- The 16-bit timer channel to be used as the output trigger channel is set up so that GRA is an o compare register and the counter will be cleared by compare match A. The trigger period is se The IMIEA bit is set to 1 in TIER to enable the compare match A interrupt.
- H'F8 is written in PBDDR and NDERB, and bits G3CMS1, G3CMS0, G2CMS1, and G2CMS0 TPCR to select compare match in the 16-bit timer channel set up in step 1 as the output trigge Output data H'80 is written in NDRB.
- The timer counter in this 16-bit timer channel is started. When compare match A occurs, the N contents are transferred to PBDR and output. The compare match/input capture A (IMFA) inter service routine writes the next output data (H'C0) in NDRB.
- Five-phase overlapping pulse output (one or two phases active at a time) can be obtained by v H'40, H'60, H'20, H'30, H'10, H'18, H'08, H'88… at successive IMFA interrupts. If the DMAC is activation by this interrupt, pulse output can be obtained without loading the CPU.

Figure 11.5 Normal TPC Output Example (Five-Phase Pulse Output)



compare registers (with output inh
2. Set the TPC output trigger period
and the non-overlap margin in GR
3. Select the counter clock source w
TPSC2 to TPSC0 in TCR. Select
clear source with bits CCLR1 and
4. Enable the IMFA interrupt in TISR
The DMAC can also be set up to
data to the next data register.
5. Set the initial output values in the
of the input/output port pins to be
TPC output.
6. Set the DDR bits of the input/outp
to be used for TPC output to 1.
7. Set the NDER bits of the pins to b
TPC output to 1.
8. In TPCR, select the 16-bit timer c
event to be used as the TPC outp
9. In TPMR, select the groups that w
in non-overlap mode.
10. Set the next TPC output values in
bits.
11. Set the STR bit to 1 in TSTR to st
counter.
12. At each IMFA interrupt, write the n
value in the NDR bits.

Figure 11.6 Setup Procedure for Non-Overlapping TPC Output (Exam



- The 16-bit timer channel to be used as the output trigger channel is set up so that GRA and GRB output compare registers and the counter will be cleared by compare match B. The TPC output tri period is set in GRB. The non-overlap margin is set in GRA. The IMIEA bit is set to 1 in TISRA to IMFA interrupts.
- H'FF is written in PBDDR and NDERB, and bits G3CMS1, G3CMS0, G2CMS1, and G2CMS0 are TPCR to select compare match in the 16-bit timer channel set up in step 1 as the output trigger. B G3NOV and G2NOV are set to 1 in TPMR to select non-overlapping output. Output data H'95 is w NDRB.
- The timer counter in this 16-bit timer channel is started. When compare match B occurs, outputs c from 1 to 0. When compare match A occurs, outputs change from 0 to 1 (the change from 0 to 1 i by the value of GRA). The IMFA interrupt service routine writes the next output data (H'65) in NDR
- Four-phase complementary non-overlapping pulse output can be obtained by writing H'59, H'56, H at successive IMFA interrupts. If the DMAC is set for activation by this interrupt, pulse output can b obtained without loading the CPU.

Figure 11.7 Non-Overlapping TPC Output Example (Four-Phase Compleme Non-Overlapping Pulse Output)

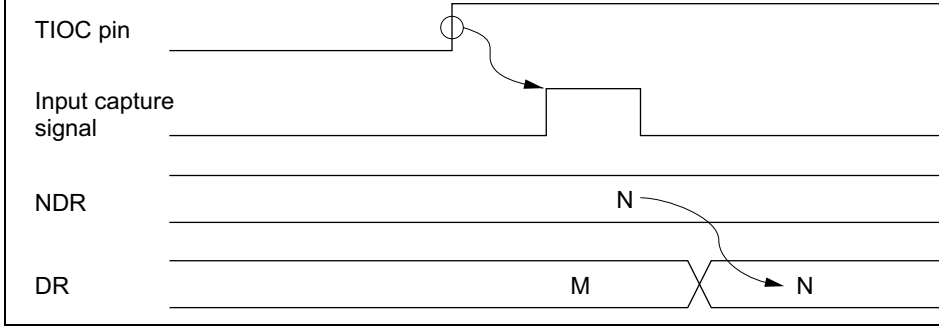


Figure 11.8 TPC Output Triggering by Input Capture (Example)

Pin functions should be changed only under conditions in which the output trigger even
occur.

11.4.2 Note on Non-Overlapping Output

During non-overlapping operation, the transfer of NDR bit values to DR bits takes plac
follows.

1. NDR bits are always transferred to DR bits at compare match A.
2. At compare match B, NDR bits are transferred only if their value is 0. Bits are not t
   if their value is 1.

Figure 11.9 illustrates the non-overlapping TPC output operation.

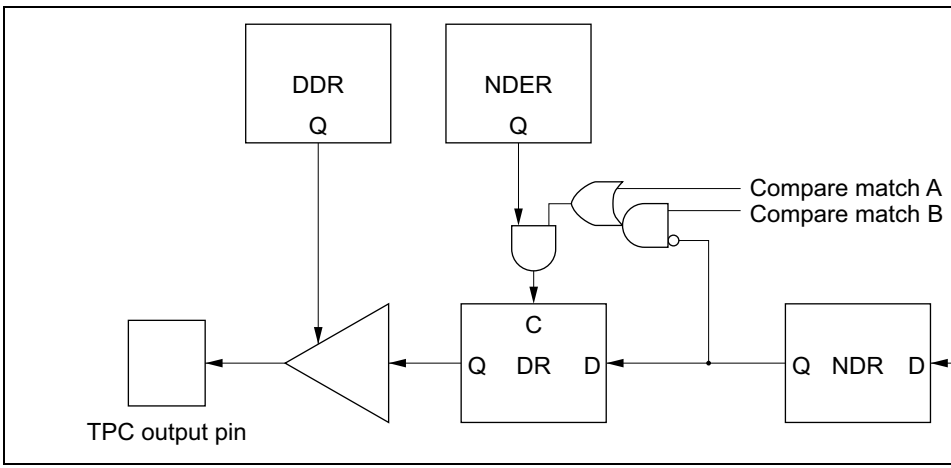


Figure 11.9 Non-Overlapping TPC Output

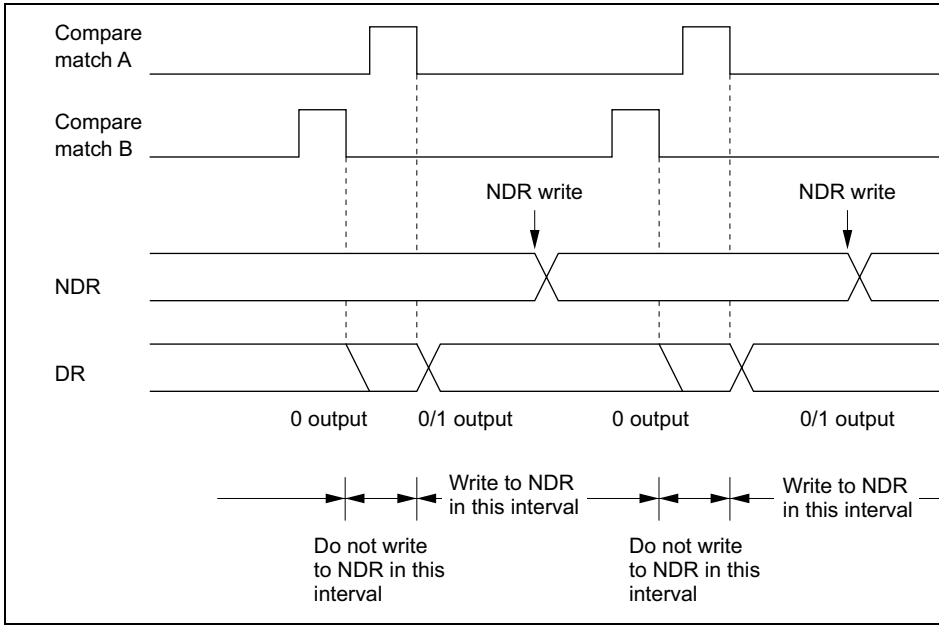


Figure 11.10 Non-Overlapping Operation and NDR Write Timing

timer operation, an interval timer interrupt is requested at each TCNT overflow.

12.1.1 Features

WDT features are listed below.

- Selection of eight counter clock sources
  φ/2, φ /32, φ /64, φ /128, φ /256, φ /512, φ /2048, or φ /4096
- Interval timer option
- Timer counter overflow generates a reset signal or interrupt.
  The reset signal is generated in watchdog timer operation. An interval timer interru
  generated in interval timer operation.
- It is possible to reset the entire H8/3028 Group using the reset signal generated by
  watchdog timer and simultaneously output the reset signal to an external device.*
  The reset signal generated by timer counter overflow during watchdog timer opera
  the entire H8/3028 Group internally.
  At the same time, a reset signal is output by pin $\overline{\text{RESO}}$ to an external device, maki
  possible to reset the entire system.

Note: * In the F-ZTAT mask ROM version, the $\overline{\text{RESO}}$ pin is for FWE input only. Cor
is not possible to output reset signals to an external device from the F-ZTAT v

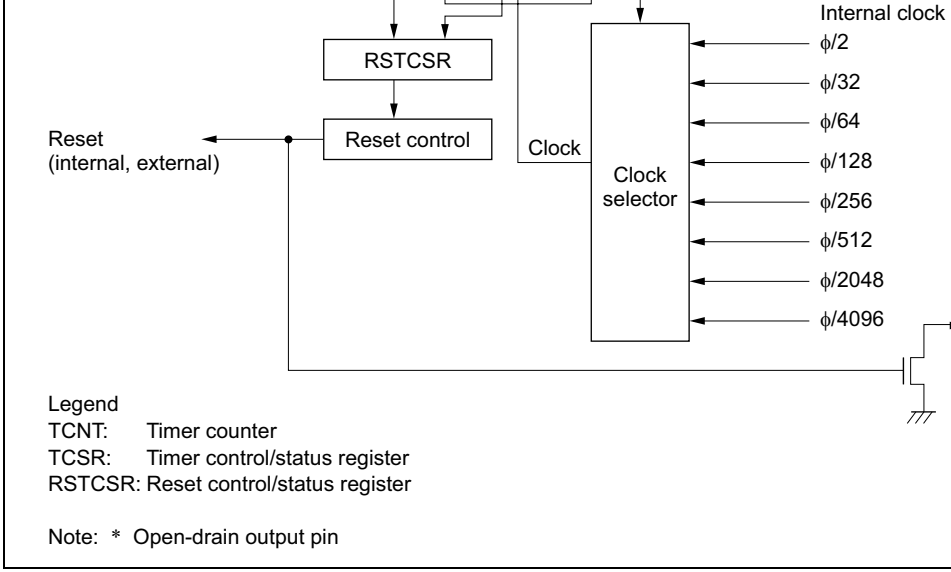


Figure 12.1 WDT Block Diagram

12.1.3 Pin Arrangement

The pins*[1] used by the watchdog timer are listed in table 12.1.

Table 12.1

Name	Abbreviation	I/O	Function
Reset output	$\overline{\text{RESO}}$	Output*[2]	Outputs watchdog timer rese external device

Notes: 1. Not available on flash memory version.
2. Open drain output pin.

	H'FFF8D	Timer counter	TCNT	R/W
H'FFF8E	H'FFF8F	Reset control/status register	RSTCSR	R/(W)*3

Notes: 1. Lower 20 bits of the address in advanced mode.
2. Write word data starting at this address.
3. Only 0 can be written in bit 7, to clear the flag.

12.2 Register Descriptions

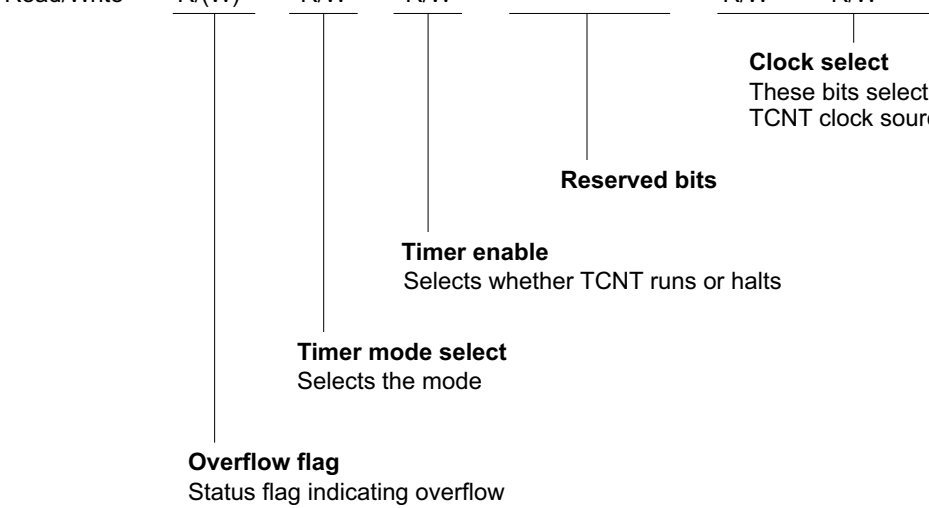
12.2.1 Timer Counter (TCNT)

TCNT is an 8-bit readable and writable up-counter.

Bit	7	6	5	4	3	2	1
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

Note: TCNT is write-protected by a password. For details see section 12.2.4, Notes o
Access.

When the TME bit is set to 1 in TCSR, TCNT starts counting pulses generated from a
clock source selected by bits CKS2 to CKS0 in TCSR. When the count overflows (cha
H'FF to H'00), the OVF bit is set to 1 in TCSR. TCNT is initialized to H'00 by a reset
the TME bit is cleared to 0.



Notes: TCSR is write-protected by a password. For details see section 12.2.4, Notes o
Access.
* Only 0 can be written, to clear the flag.

Bits 7 to 5 are initialized to 0 by a reset and in standby mode. Bits 2 to 0 are initialized
reset. In software standby mode bits 2 to 0 are not initialized, but retain their previous v

Bit 7—Overflow Flag (OVF): This status flag indicates that the timer counter has ove
from H'FF to H'00.

Bit 7 OVF	Description
0	[Clearing condition] Cleared by reading OVF when OVF = 1, then writing 0 in OVF (
1	[Setting condition] Set when TCNT changes from H'FF to H'00

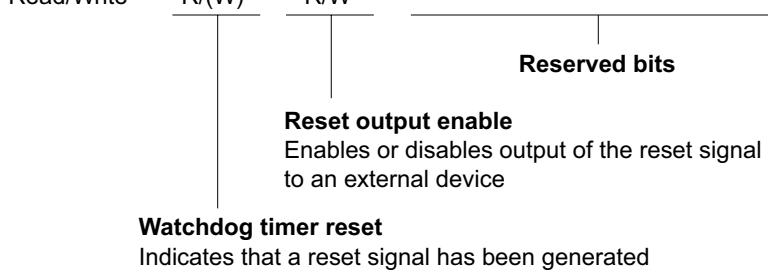
Bit 5—Timer Enable (TME): Selects whether TCNT runs or is halted. When WT/$\overline{IT}$ the software standby bit (SSBY) to 0 in SYSCR before setting TME. When setting SS TME should be cleared to 0.

Bit 5 TME	Description
0	TCNT is initialized to H'00 and halted
1	TCNT is counting

Bits 4 and 3—Reserved: These bits cannot be modified and are always read as 1.

Bits 2 to 0—Clock Select 2 to 0 (CKS2/1/0): These bits select one of eight internal c obtained by prescaling the system clock (ϕ), for input to TCNT.

Bit 2 CKS2	Bit 1 CKS1	Bit 0 CKS0	Description
0	0	0	ϕ/2
		1	ϕ /32
	1	0	ϕ /64
		1	ϕ /128
1	0	0	ϕ /256
		1	ϕ /512
	1	0	ϕ /2048
		1	ϕ /4096



Notes: The procedure for writing to RSTCSR differs from that for other registers in or prevent its contents from being overwritten accidentally. For details see sectio Notes on Register Access.
* Only 0 can be written to bit 7, to clear the flag.

Bits 7 and 6 are initialized by input of a reset signal to the $\overline{\text{RES}}$ pin. They are not initial reset signals generated by watchdog timer overflow.

Bit 7—Watchdog Timer Reset (WRST): During watchdog timer operation, this bit in TCNT has overflowed and generated a reset signal. This reset signal resets the entire H Group chip internally. At the same time, if the RSTOE bit is set to 1, the reset signal is from the $\overline{\text{RESO}}$ pin as low-level output to an external device, making it possible to rese system. Note that the flash memory version is not equipped with a $\overline{\text{RESO}}$ pin.

Bit 7 WRST	Description
0	[Clearing conditions] (I • Reset signal at $\overline{\text{RES}}$ pin. • Read WRST flag when WRST = 1, then write 0 to WRST.
1	[Setting condition] Set when TCNT overflow generates a reset signal during watchdog timer o

Bits 5 to 0—Reserved: These bits are reserved. They cannot be written to and are alw
as 1.

12.2.4 Notes on Register Access

The watchdog timer's TCNT, TCSR, and RSTCSR registers differ from other register
more difficult to write. The procedures for writing and reading these registers are give

Writing to TCNT and TCSR: These registers must be written by a word transfer ins
They cannot be written by byte instructions. Figure 12.2 shows the format of data writ
TCNT and TCSR. TCNT and TCSR both have the same write address. The write data
contained in the lower byte of the written word. The upper byte must contain H'5A (pa
TCNT) or H'A5 (password for TCSR). This transfers the write data from the lower by
or TCSR.

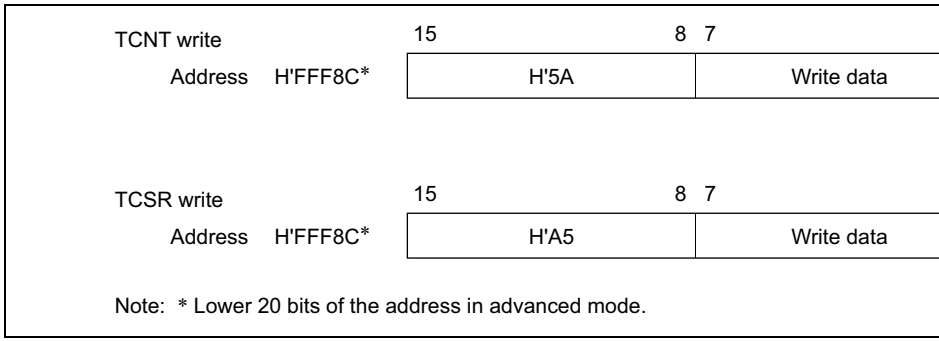


Figure 12.2 Format of Data Written to TCNT and TCSR

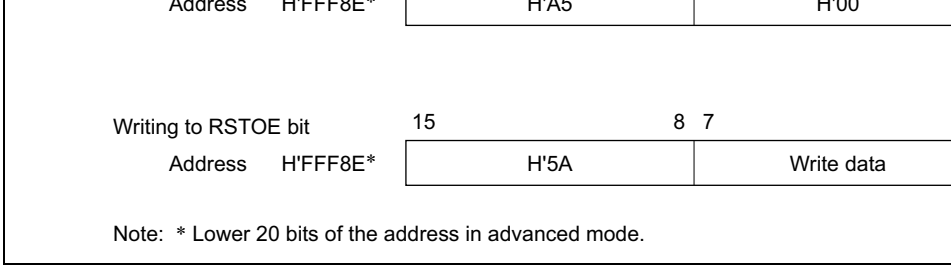


Figure 12.3 Format of Data Written to RSTCSR

Reading TCNT, TCSR, and RSTCSR: These registers are read like other registers. R
TCNT, TCSR, and RSTCSR: These registers are read like other registers. Byte transfe
instructions can be used. The read addresses are H'FFF8C for TCSR, H'FFF8D for TC
H'FFF8F for RSTCSR, as listed in table 12.3.

Table 12.3 Read Addresses of TCNT, TCSR, and RSTCSR

Address*	Register
H'FFF8C	TCSR
H'FFF8D	TCNT
H'FFF8F	RSTCSR

Note: * Lower 20 bits of the address in advanced mode.

TCNT value (normally by writing H'00) before overflow occurs. If TCNT fails to be r
overflows due to a system crash, etc., the H8/3028 Group is internally reset for a dura
states.

It is possible to output the reset signal generated by the WDT to an external device fro
pin and thereby reset the external system. The external reset signal is output for a dura
states. External output of the reset signal is enabled or disabled using the RSTOE bit i
Note, however, that the flash memory version is not equipped with a $\overline{\text{RESO}}$ pin.

A watchdog reset has the same vector as a reset generated by input at the $\overline{\text{RES}}$ pin. So
distinguish a $\overline{\text{RES}}$ reset from a watchdog reset by checking the WRST bit in RSTCSR

If a $\overline{\text{RES}}$ reset and a watchdog reset occur simultaneously, the $\overline{\text{RES}}$ reset takes priority

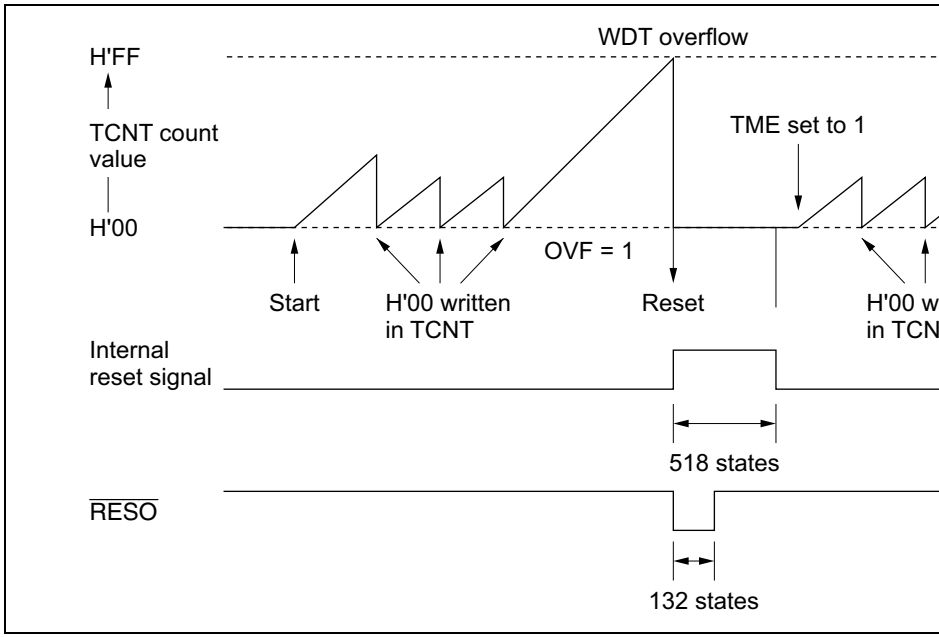


Figure 12.4 Operation in Watchdog Timer Mode

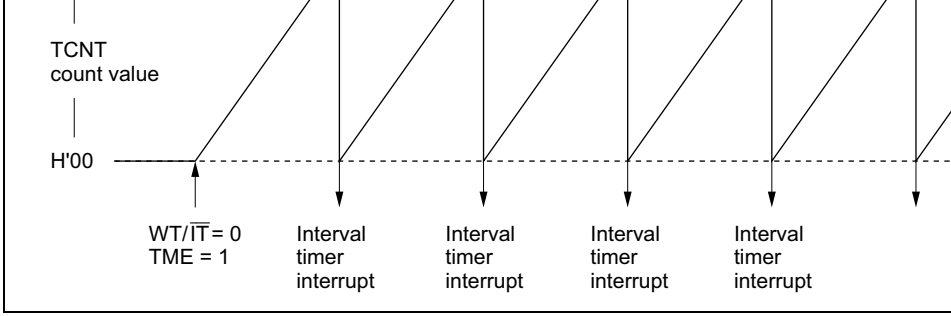


Figure 12.5 Interval Timer Operation

12.3.3 Timing of Setting of Overflow Flag (OVF)

Figure 12.6 shows the timing of setting of the OVF flag. The OVF flag is set to 1 when overflows. At the same time, a reset signal is generated in watchdog timer operation, or timer interrupt is generated in interval timer operation.

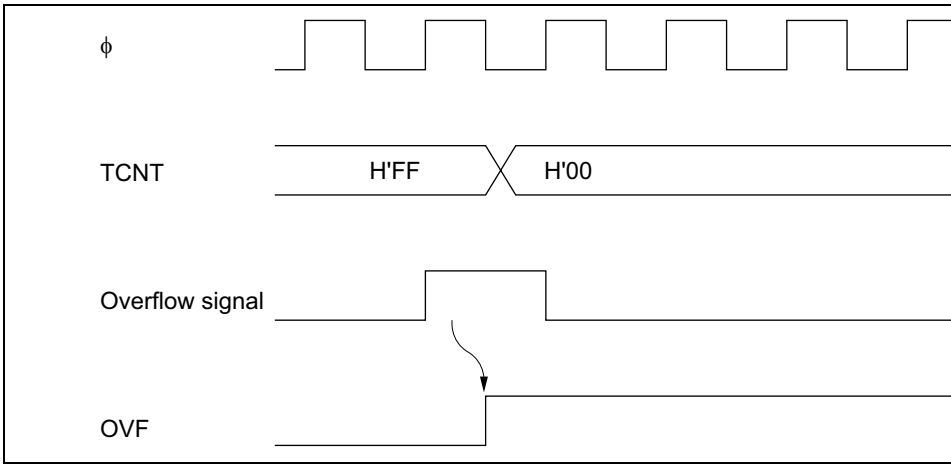


Figure 12.6 Timing of Setting of OVF

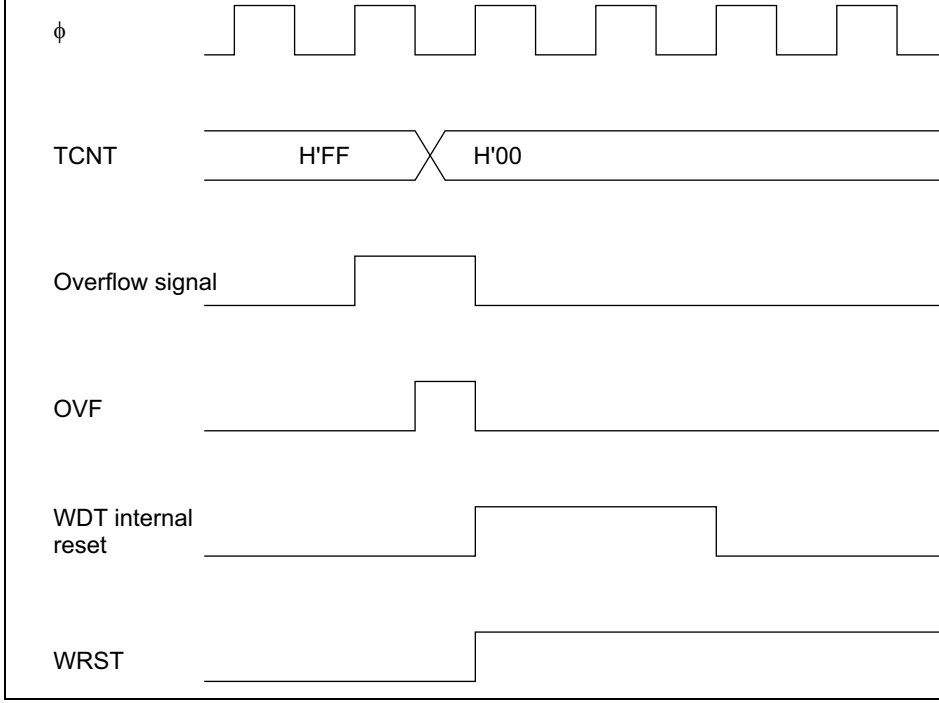


Figure 12.7 Timing of Setting of WRST Bit and Internal Reset

incremented. See figure 12.8.

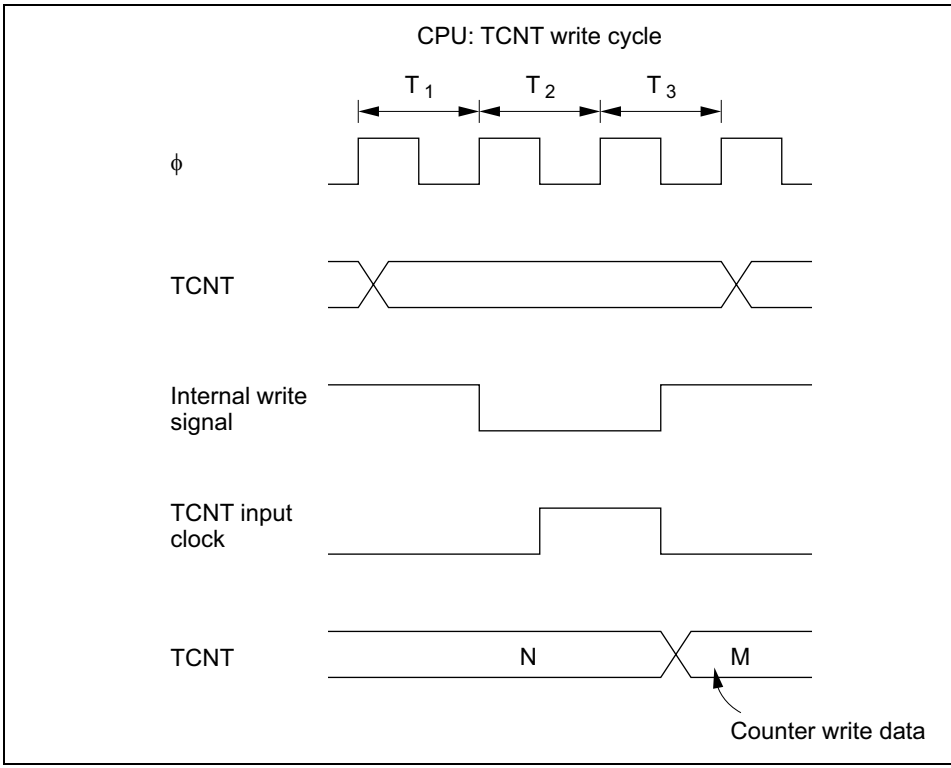


Figure 12.8 Contention between TCNT Write and Count up

Changing CKS2 to CKS0 Bit: Halt TCNT by clearing the TME bit to 0 in TCSR befo changing the values of bits CKS2 to CKS0.

When the SCI is not used, it can be halted to conserve power. Each SCI channel can b
independently. For details, see section 20.6, Module Standby Function.

The SCI also has a smart card interface function conforming to the ISO/IEC 7816-3 (I
Card) standard. This function supports serial communication with a smart card. Swit
between the normal serial communication interface and the smart card interface is car
means of a register setting.

13.1.1 Features

SCI features are listed below.

- Selection of synchronous or asynchronous mode for serial communication

  Asynchronous mode

  Serial data communication is synchronized one channel at a time. The SCI can co
  with a universal asynchronous receiver/transmitter (UART), asynchronous commu
  interface adapter (ACIA), or other chip that employs standard asynchronous comm
  It can also communicate with two or more other processors using the multiprocess
  communication function. There are twelve selectable serial data transfer formats.

  — Data length: 7 or 8 bits
  — Stop bit length: 1 or 2 bits
  — Parity: even/odd/none
  — Multiprocessor bit: 1 or 0
  — Receive error detection: parity, overrun, and framing errors
  — Break detection: by reading the RxD level directly when a framing err

  Synchronous mode

  Serial data communication is synchronized with a clock signal. The SCI can comm
  with other chips having a synchronous communication function.

  There is a single serial data communication format.

  — Data length: 8 bits
  — Receive error detection: overrun errors

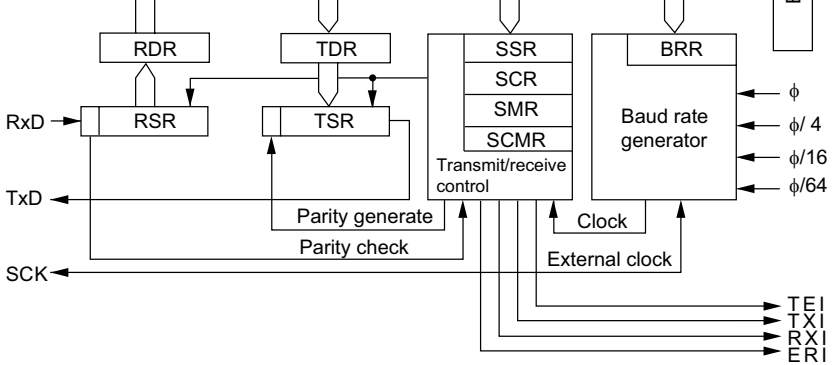
- Selectable transmit/receive clock sources: internal clock from baud rate generator, c
  clock from the SCK pin
- Four types of interrupts

  Transmit-data-empty, transmit-end, receive-data-full, and receive-error interrupts ar
  independently. The transmit-data-empty and receive-data-full interrupts from SCI0
  activate the DMA controller (DMAC) to transfer data.

Features of the smart card interface are listed below.

- Asynchronous communication
  - Data length: 8 bits
  - Parity bits generated and checked
  - Error signal output in receive mode (parity error)
  - Error signal detect and automatic data retransmit in transmit mode
  - Supports both direct convention and inverse convention
- Built-in baud rate generator with selectable bit rates
- Three types of interrupts

  Transmit-data-empty, receive-data-full, and transmit/receive-error interrupts are req
  independently. The transmit-data-empty and receive-data-full interrupts can activat
  controller (DMAC) to transfer data.



Legend
RSR : Receive shift register
RDR : Receive data register
TSR : Transmit shift register
TDR : Transmit data register
SMR : Serial mode register
SCR : Serial control register
SSR : Serial status register
BRR : Bit rate register
SCMR : Smart card mode register

Figure 13.1 SCI Block Diagram

	Transmit data pin	TxD_0	Output	SCI_0 transmit data
1	Serial clock pin	SCK_1	Input/output	SCI_1 clock input/ou
	Receive data pin	RxD_1	Input	SCI_1 receive data i
	Transmit data pin	TxD_1	Output	SCI_1 transmit data
2	Serial clock pin	SCK_2	Input/output	SCI_2 clock input/ou
	Receive data pin	RxD_2	Input	SCI_2 receive data i
	Transmit data pin	TxD_2	Output	SCI_2 transmit data

0	H'FFFB0	Serial mode register	SMR	R/W	H
	H'FFFB1	Bit rate register	BRR	R/W	H
	H'FFFB2	Serial control register	SCR	R/W	H
	H'FFFB3	Transmit data register	TDR	R/W	H
	H'FFFB4	Serial status register	SSR	R/(W)*2	H
	H'FFFB5	Receive data register	RDR	R	H
	H'FFFB6	Smart card mode register	SCMR	R/W	H
1	H'FFFB8	Serial mode register	SMR	R/W	H
	H'FFFB9	Bit rate register	BRR	R/W	H
	H'FFFBA	Serial control register	SCR	R/W	H
	H'FFFBB	Transmit data register	TDR	R/W	H
	H'FFFBC	Serial status register	SSR	R/(W)*2	H
	H'FFFBD	Receive data register	RDR	R	H
	H'FFFBE	Smart card mode register	SCMR	R/W	H
2	H'FFFC0	Serial mode register	SMR	R/W	H
	H'FFFC1	Bit rate register	BRR	R/W	H
	H'FFFC2	Serial control register	SCR	R/W	H
	H'FFFC3	Transmit data register	TDR	R/W	H
	H'FFFC4	Serial status register	SSR	R/(W)*2	H
	H'FFFC5	Receive data register	RDR	R	H
	H'FFFC6	Smart card mode register	SCMR	R/W	H

Notes: 1. Indicates the lower 20 bits of the address in advanced mode.
2. Only 0 can be written, to clear flags.

Read/Write

The SCI loads serial data input at the RxD pin into RSR in the order received, LSB (bit
thereby converting the data to parallel data. When one byte of data has been received,
automatically transferred to RDR. The CPU cannot read or write RSR directly.

13.2.2 Receive Data Register (RDR)

RDR is the register that stores received serial data.

Bit	7	6	5	4	3	2	1
Initial value	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R

When the SCI has received one byte of serial data, it transfers the received data from R
RDR for storage, completing the receive operation. RSR is then ready to receive the ne
This double-buffering allows data to be received continuously.

RDR is a read-only register. Its contents cannot be modified by the CPU. RDR is initi
H'00 by a reset and in standby mode.

LSB (bit 0) first. After transmitting one data byte, the SCI automatically loads the nex
data from TDR into TSR and starts transmitting it. If the TDRE flag is set to 1 in SSR
the SCI does not load the TDR contents into TSR. The CPU cannot read or write TSR

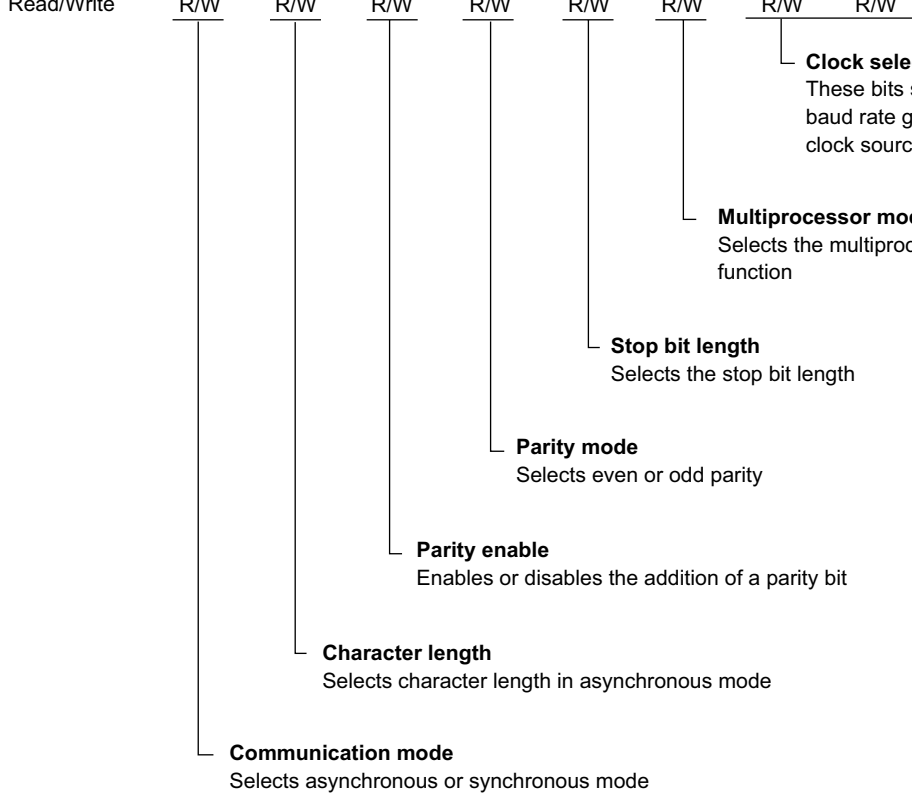
13.2.4 Transmit Data Register (TDR)

TDR is an 8-bit register that stores data for serial transmission.

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

When the SCI detects that TSR is empty, it moves transmit data written in TDR from
TSR and starts serial transmission. Continuous serial transmission is possible by writ
transmit data in TDR during serial transmission from TSR.

The CPU can always read and write TDR. TDR is initialized to H'FF by a reset and i
mode.



The CPU can always read and write SMR. SMR is initialized to H'00 by a reset and in mode.

Bit 7—Communication Mode (C/$\overline{A}$)/GSM Mode (GM): The function of this bit diffe normal serial communication interface and for the smart card interface. Its function is with the SMIF bit in SCMR.

For smart card interface (SMIF bit in SCMR set to 1). Selects GSM mode for the interface.

Bit 7 GM	Description	
0	The TEND flag is set 12.5 etu after the start bit	(
1	The TEND flag is set 11.0 etu after the start bit	

Note: etu (Elementary time unit: the time for transfer of one bit)

Bit 6—Character Length (CHR): Selects 7-bit or 8-bits data length in asynchronous synchronous mode, the data length is 8 bits regardless of the CHR setting,

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.

Bit 5—Parity Enable (PE): In asynchronous mode, this bit enables or disables the ad parity bit to transmit data, and the checking of the parity bit in receive data. In synchr the parity bit is neither added nor checked, regardless of the PE bit setting.

Bit 5 PE	Description	
0	Parity bit not added or checked	(
1	Parity bit added and checked*	

Note: * When PE bit is set to 1, an even or odd parity bit is added to transmit data acc even or odd parity mode selection by the O/$\overline{E}$ bit, and the parity bit in receive da checked to see that it matches the even or odd mode selected by the O/$\overline{E}$ bit.

Notes: 1. When even parity is selected, the parity bit added to transmit data makes an
number of 1s in the transmitted character and parity bit combined. Receive
have an even number of 1s in the received character and parity bit combine
2. When odd parity is selected, the parity bit added to transmit data makes an
of 1s in the transmitted character and parity bit combined. Receive data mu
odd number of 1s in the received character and parity bit combined.

Bit 3—Stop Bit Length (STOP): Selects one or two stop bits in asynchronous mode.
is used only in asynchronous mode. In synchronous mod no stop bit is added, so the S
setting is ignored.

Bit 3 STOP	Description	
0	1 stop bit*1	(In
1	2 stop bits*2	

Notes: 1. One stop bit (with value 1) is added to the end of each transmitted character
2. Two stop bits (with value 1) are added to the end of each transmitted charac

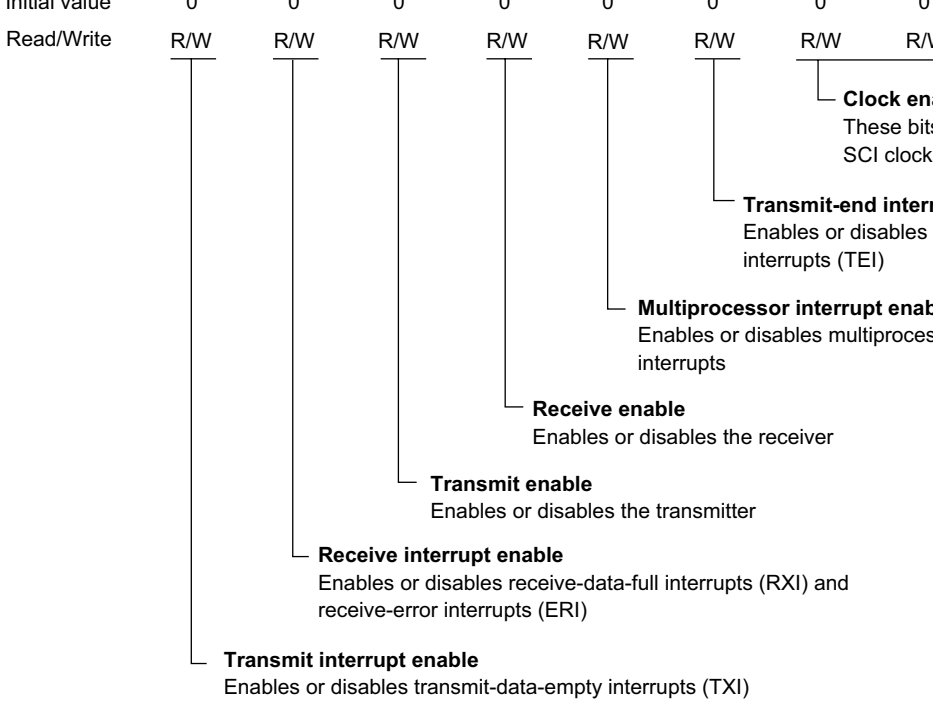
In receiving, only the first stop bit is checked, regardless of the STOP bit setting. If th
stop bit is 1, it is treated as a stop bit. If the second stop bit is 0, it is treated as the start
next incoming character.

Bit 2—Multiprocessor Mode (MP): Selects a multiprocessor format. When a multipr
format is selected, parity settings made by the PE and O/$\overline{E}$ bits are ignored. The MP bi
valid only in asynchronous mode. It is ignored in synchronous mode.

For further information on the multiprocessor communication function, see section 13.
Multiprocessor Communication.

Bit 2 MP	Description	
0	Multiprocessor function disabled	(In
1	Multiprocessor format selected	

0	1	φ/4
1	0	φ/16
1	1	φ/64



The CPU can always read and write SCR. SCR is initialized to H'00 by a reset and in s
mode.

clearing it to 0; or by clearing the TIE bit to 0.

Bit 6—Receive Interrupt Enable (RIE): Enables or disables the receive-data-full int
requested when the RDRF flag in SSR is set to 1 due to transfer of serial receive data
RDR; also enables or disables the receive-error interrupt (ERI).

Bit 6 RIE	Description
0	Receive-data-full (RXI) and receive-error (ERI) interrupt requests a (
1	Receive-data-full (RXI) and receive-error (ERI) interrupt requests a

Note: * RXI and ERI interrupt requests can be cleared by reading the value 1 from the
PER, or ORER flag, then clearing the flag to 0; or by clearing the RIE bit to 0.

Bit 5—Transmit Enable (TE): Enables or disables the start of SCI serial transmitting

Bit 5 TE	Description
0	Transmitting disabled*1 (
1	Transmitting enabled*2

Notes: 1. The TDRE flag is fixed at 1 in SSR.
2. In the enabled state, serial transmission starts when the TDRE flag in SSR
0 after writing of transmit data into TDR. Select the transmit format in SMR
setting the TE bit to 1.

mode, or serial clock input is detected in synchronous mode. Select the rece
in SMR before setting the RE bit to 1.

Bit 3—Multiprocessor Interrupt Enable (MPIE): Enables or disables multiprocesso
The MPIE bit setting is valid only in asynchronous mode, and only if the MP bit is set t
SMR. The MPIE bit setting is ignored in synchronous mode or when the MP bit is clea

Bit 3 MPIE	Description
0	Multiprocessor interrupts are disabled (normal receive operation) ([Clearing conditions] • The MPIE bit is cleared to 0 • MPB = 1 in received data
1	Multiprocessor interrupts are enabled* Receive-data-full interrupts (RXI), receive-error interrupts (ERI), and the RDRF, FER, and ORER status flags in SSR are disabled until da multiprocessor bit set to 1 is received.

Note: * The SCI does not transfer receive data from RSR to RDR, does not detect receiv
and does not set the RDRF, FER, and ORER flags in SSR. When it receives da
MPB = 1, the SCI sets the MPB bit to 1 in SSR, automatically clears the MPIE bi
enables RXI and ERI interrupts (if the TIE and RIE bits in SCR are set to 1), and
FER and ORER flags to be set.

Bit 2—Transmit-End interrupt Enable (TEIE): Enables or disables the transmit-end
(TEI) requested if TDR does not contain valid transmit data when the MSB is transmitt

Bit 2 TEIE	Description
0	Transmit-end interrupt requests (TEI) are disabled* (In
1	Transmit-end interrupt requests (TEI) are enabled*

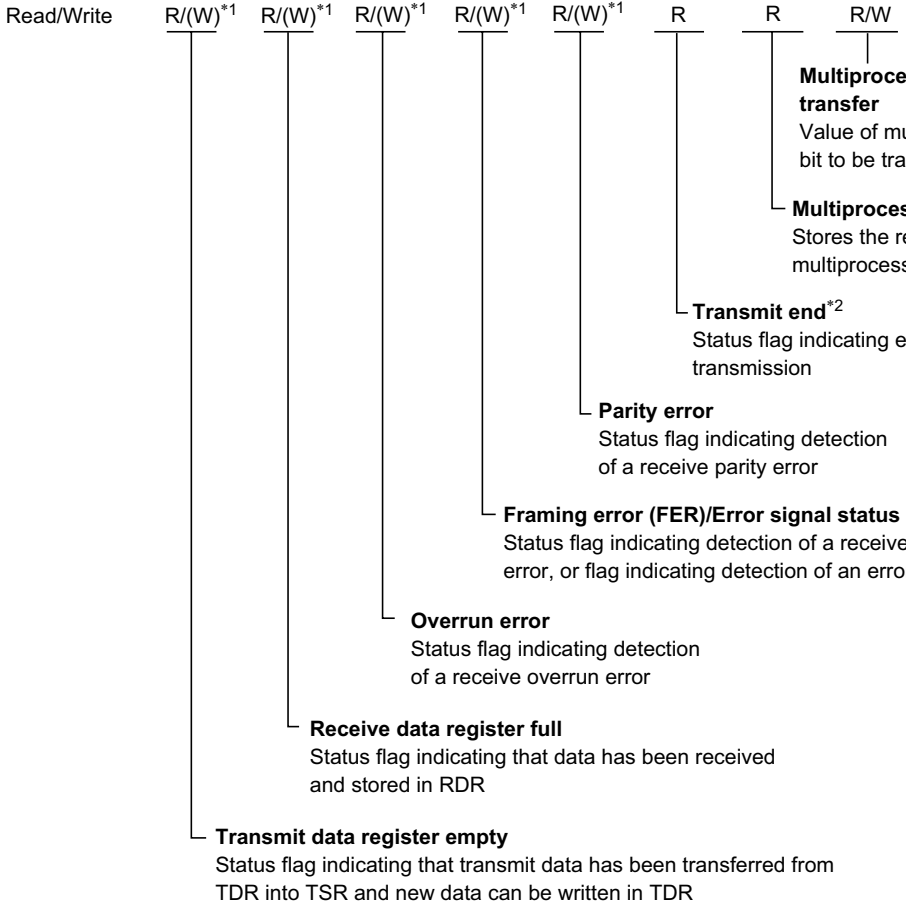
Note: * TEI interrupt requests can be cleared by reading the value 1 from the TDRE flag
then clearing the TDRE flag to 0, thereby also clearing the TEND flag to 0; or by
the TEIE bit to 0.

The CKE0 setting is valid only in asynchronous mode, and only when the SCI is inter
clocked (CKE1 = 0). The CKE0 setting is ignored in synchronous mode, or when an
clock source is selected (CKE1 = 1). Select the SCI operating mode in SMR before se
CKE1 and CKE0 bits . For further details on selection of the SCI clock source, see ta
section 13.3, Operation.

Bit 1 CKE1	Bit 0 CKE0	Description	
0	0	Asynchronous mode	Internal clock, SCK pin available for generic i
		Synchronous mode	Internal clock, SCK pin used for serial clock o
0	1	Asynchronous mode	Internal clock, SCK pin used for clock output*
		Synchronous mode	Internal clock, SCK pin used for serial clock o
1	0	Asynchronous mode	External clock, SCK pin used for clock input*3
		Synchronous mode	External clock, SCK pin used for serial clock i
1	1	Asynchronous mode	External clock, SCK pin used for clock input*3
		Synchronous mode	External clock, SCK pin used for serial clock i

Notes: 1. Initial value
2. The output clock frequency is the same as the bit rate.
3. The input clock frequency is 16 times the bit rate.

1	0	1	SCK pin used for clock output
1	1	0	SCK pin output fixed high
1	1	1	SCK pin used for clock output



Notes: 1. Only 0 can be written, to clear the flag.
2. Function differs between the normal serial communication interface and the smart c

The CPU can always read and write SSR, but cannot write 1 in the TDRE, RDRF, OR and FER flags. These flags can be cleared to 0 only if they have first been read while The TEND and MPB flags are read-only bits that cannot be written.

SSR is initialized to H'84 by a reset and in standby mode.

1	TDR does not contain valid transmit data [Setting conditions] • The chip is reset or enters standby mode • The TE bit in SCR is cleared to 0 • TDR contents are loaded into TSR, so new data can be written i	(In

Bit 6—Receive Data Register Full (RDRF): Indicates that RDR contains new receive

Bit 6 RDRF	Description	
0	RDR does not contain new receive data [Clearing conditions] • The chip is reset or enters standby mode • Read RDRF when RDRF = 1, then write 0 in RDRF • The DMAC reads data from RDR	(In
1	RDR contains new receive data [Setting condition] Serial data is received normally and transferred from RSR to RDR	

Note: The RDR contents and the RDRF flag are not affected by detection of receive er clearing of the RE bit to 0 in SCR. They retain their previous values. If the RDR still set to 1 when reception of the next data ends, an overrun error will occur and receive data will be lost.

1	A receive overrun error occurred*2 [Setting condition] Reception of the next serial data ends when RDRF = 1

Notes: 1. Clearing the RE bit to 0 in SCR does not affect the ORER flag, which retain previous value.
2. RDR continues to hold the receive data prior to the overrun error, so subse receive data is lost. Serial receiving cannot continue while the ORER flag i synchronous mode, serial transmitting is also disabled.

Bit 4—Framing Error (FER)/Error Signal Status (ERS): The function of this bit d normal serial communication interface and for the smart card interface. Its function is with the SMIF bit in SCMR.

For serial communication interface (SMIF bit in SCMR cleared to 0): Indicates th reception ended abnormally due to a framing error in asynchronous mode.

Bit 4 FER	Description
0	Receiving is in progress or has ended normally*1 ([Clearing conditions] • The chip is reset or enters standby mode • Read FER when FER = 1, then write 0 in FER
1	A receive framing error occurred*2 [Setting condition] The stop bit at the end of the receive data is checked and found to

Notes: 1. Clearing the RE bit to 0 in SCR does not affect the FER flag, which retains value.
2. When the stop bit length is 2 bits, only the first bit is checked. The second not checked. When a framing error occurs the SCI transfers the receive da but does not set the RDRF flag. Serial receiving cannot continue while the set to 1. In synchronous mode, serial transmitting is also disabled.

	• Read ERS when ERS = 1, then write 0 in ERS
1	An error signal has been sent from the receiving side indicating dete parity error [Setting condition] The error signal is low when sampled

Note: * Clearing the TE bit to 0 in SCR does not affect the ERS flag, which retains its pr value.

Bit 3—Parity Error (PER): Indicates that data reception ended abnormally due to a p in asynchronous mode.

Bit 3 PER	Description
0	Receiving is in progress or has ended normally*1 (In [Clearing conditions] • The chip is reset or enters standby mode • Read PER when PER = 1, then write 0 in PER
1	A receive parity error occurred*2 [Setting condition] The number of 1s in receive data, including the parity bit, does not m even or odd parity setting of O/$\overline{E}$ in SMR

Notes: 1. Clearing the RE bit to 0 in SCR does not affect the PER flag, which retains it value.
2. When a parity error occurs the SCI transfers the receive data into RDR but d the RDRF flag. Serial receiving cannot continue while the PER flag is set to synchronous mode, serial transmitting is also disabled.

Bit 2—Transmit End (TEND): The function of this bit differs for the normal serial communication interface and for the smart card interface. Its function is switched with bit in SCMR.

	• The DMAC writes data in TDR
1	End of transmission ([Setting conditions] • The chip is reset or enters standby mode • The TE bit in SCR is cleared to 0 • TDRE is 1 when the last bit of a 1-byte serial transmit characte transmitted

For smart card interface (SMIF bit in SCMR set to 1): Indicates that when the last serial character was transmitted TDR did not contain valid transmit data, so transmissi ended. The TEND flag is a read-only bit and cannot be written.

Bit 2 TEND	Description
0	Transmission is in progress [Clearing conditions] • Read TDRE when TDRE = 1, then write 0 in TDRE • The DMAC writes data in TDR
1	End of transmission ([Setting conditions] • The chip is reset or enters standby mode • The TE bit is cleared to 0 in SCR and the FER/ERS bit is also c • TDRE is 1 and FER/ERS is 0 (normal transmission) 2.5 etu (wh or 1.0 etu (when GM = 1) after a 1-byte serial character is trans

Note: etu (Elementary time unit: the time for transfer of one bit)

its previous value.

Bit 0—Multiprocessor Bit Transfer (MPBT): Stores the value of the multiprocessor transmit data when a multiprocessor format in selected for transmitting in asynchronou

The MPBT bit setting is ignored in synchronous mode, when a multiprocessor format is selected, or when the SCI cannot transmit.

Bit 1 MPBT	Description	
0	Multiprocessor bit value in transmit data is 0	(In
1	Multiprocessor bit value in transmit data is 1	

13.2.8 Bit Rate Register (BRR)

BRR is an 8-bit register that., together with the CKS1 and CKS0 bits in SMR that selec rate generator clock source, determines the serial communication bit rate.

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

The CPU can always read and write BRR. BRR is initialized to H'FF by a reset and in mode. Each SCI channel has independent baud rate generator control, so different valu set in the three channels.

Table 13.3 shows examples of BRR settings in asynchronous mode. Table 13.4 shows of BRR settings in synchronous mode.

600	0	103	0.16	0	108	0.21	0	127	0.00	0	155
1200	0	51	0.16	0	54	–0.70	0	63	0.00	0	77
2400	0	25	0.16	0	26	1.14	0	31	0.00	0	38
4800	0	12	0.16	0	13	–2.48	0	15	0.00	0	19
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	—	—

	ϕ (MHz)										
Bit Rate	**3.6864**			**4**			**4.9152**				
(bit/s)	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**
110	2	64	0.70	2	70	0.03	2	86	0.31	2	88
150	1	191	0.00	1	207	0.16	1	255	0.00	2	64
300	1	95	0.00	1	103	0.16	1	127	0.00	1	129
600	0	191	0.00	0	207	0.16	0	255	0.00	1	64
1200	0	95	0.00	0	103	0.16	0	127	0.00	0	129
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	0.00	0	4	–1.70	0	4
38400	0	2	0.00	0	2	8.51	0	3	0.00	0	3

2400	0	77	0.16	0	79	0.00	0	95	0.00	0	103
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

Bit Rate (bit/s)	φ (MHz)										
	9.8304			10			12			1	
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N
110	2	174	–0.26	2	177	–0.25	2	212	0.03	2	217
150	2	127	0.00	2	129	0.16	2	155	0.16	2	159
300	1	255	0.00	2	64	0.16	2	77	0.16	2	79
600	1	127	0.00	1	129	0.16	1	155	0.16	1	159
1200	0	255	0.00	1	64	0.16	1	77	0.16	1	79
2400	0	127	0.00	0	129	0.16	0	155	0.16	0	159
4800	0	63	0.00	0	64	0.16	0	77	0.16	0	79
9600	0	31	0.00	0	32	–1.36	0	38	0.16	0	39
19200	0	15	0.00	0	15	1.73	0	19	–2.34	0	19
31250	0	9	–1.70	0	9	0.00	0	11	0.00	0	11
38400	0	7	0.00	0	7	1.73	0	9	–2.34	0	9

1200	1	84	–0.43	1	90	0.16	1	95	0.00	1	103	0.16	1	116	0.16	1
2400	0	168	0.16	0	181	0.16	0	191	0.00	0	207	0.16	0	233	0.16	1
4800	0	84	–0.43	0	90	0.16	0	95	0.00	0	103	0.16	0	116	0.16	0
9600	0	41	0.76	0	45	–0.93	0	47	0.00	0	51	0.16	0	58	–0.69	0
19200	0	20	0.76	0	22	–0.93	0	23	0.00	0	25	0.16	0	28	1.02	0
31250	0	12	0.00	0	13	0.00	0	14	–1.70	0	15	0.00	0	17	0.00	0
38400	0	10	–3.82	0	10	3.57	0	11	0.00	0	12	0.16	0	14	–2.34	0

Bit Rate (bit/s)	ϕ (MHz) 25		
	n	N	Error (%)
110	3	110	–0.02
150	3	80	0.47
300	2	162	–0.15
600	2	80	0.47
1200	1	162	–0.15
2400	1	80	0.47
4800	0	162	–0.15
9600	0	80	0.47
19200	0	40	–0.76
31250	0	24	0.00
38400	0	19	1.73

1k	1	124	1	249	2	124	—	—	2	202	2	249	3	69	3	77
2.5k	0	199	1	99	1	199	1	249	2	80	2	99	2	112	2	124
5k	0	99	0	199	1	99	1	124	1	162	1	199	1	224	1	249
10k	0	49	0	99	0	199	0	249	1	80	1	99	1	112	1	124
25k	0	19	0	39	0	79	0	99	0	129	0	159	0	179	0	199
50k	0	9	0	19	0	39	0	49	0	64	0	79	0	89	0	99
100k	0	4	0	9	0	19	0	24	—	—	0	39	0	44	0	49
250k	0	1	0	3	0	7	0	9	0	12	0	15	0	17	0	19
500k	0	0*	0	1	0	3	0	4	—	—	0	7	0	8	0	9
1M			0	0*	0	1	—	—	—	—	0	3	0	4	0	4
2M					0	0*	—	—	—	—	0	1	—	—	—	—
2.5M					—	—	0	0*	—	—	—	—	—	—	—	—
4M											0	0*	—	—	—	—

Note: Settings with an error of 1% or less are recommended.

Legend

Blank : No setting available

— : Setting possible, but error occurs

* : Continuous transmission/reception not possible

$$N = \frac{\phi}{8 \times 2^{2n-1} \times B} \times 10^6 - 1$$

B: Bit rate (bit/s)
N: BRR setting for baud rate generator ($0 \leq N \leq 255$)
ϕ: System clock frequency (MHz)
n: Baud rate generator clock source (n = 0, 1, 2, 3)
(For the clock sources and values of n, see the following table.)

		SMR Settings	
n	**Clock Source**	**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 calculated as follows:

$$\text{Error (\%)} = \left\{ \frac{\phi \times 10^6}{(N + 1) \times B \times 64 \times 2^{2n-1}} - 1 \right\} \times 100$$

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
20	625000	0	0
25	781250	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
20	5.0000	312500
25	6.2500	390625

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
25	4.1667	4166666.7

Selection of asynchronous or synchronous mode and the transmission format for the n
communication interface is made in SMR, as shown in table 13.8. The SCI clock sou
selected by the C/$\overline{A}$ bit in SMR and the CKE1 and CKE0 bits in SCR, as shown in tab

For details of the procedures for switching between LSB-first and MSB-first mode an
the data logic level, see section 14.2.1, Smart Card Mode Register (SCMR).

For selection of the smart card interface format, see section 14.3.3, Data Format.

Asynchronous Mode

- Data length is selectable: 7 or 8 bits
- Parity and multiprocessor bits are selectable, and so is the stop bit length (1 or 2 bi
selections determine the communication format and character length.
- In receiving, it is possible to detect framing errors, parity errors, overrun errors, an
state.
- An internal or external clock can be selected as the SCI clock source.
  - When an internal clock is selected, the SCI operates using the on-chip baud rat
and can output a serial clock signal with a frequency matching the bit rate.
  - When an external clock is selected, the external clock input must have a freque
the bit rate. (The on-chip baud rate generator is not used.)

Synchronous Mode

- The communication format has a fixed 8-bit data length.
- In receiving, it is possible to detect overrun errors.
- An internal or external clock can be selected as the SCI clock source.
  - When an internal clock is selected, the SCI operates using the on-chip baud rat
and can output a serial clock signal to external devices.
  - When an external clock is selected, the SCI operates on the input serial clock.
baud rate generator is not used.

after at least 2 etu.

- Only asynchronous communication is supported. There is no synchronous commun function.

For details of smart card interface operation, see section 14, Smart Card Interface.

Table 13.8 SMR Settings and Serial Communication Formats

SMR Settings						SCI Communication For		
Bit 7 C/$\overline{A}$	Bit 6 CHR	Bit 2 MP	Bit 5 PE	Bit 3 STOP	Mode	Data Length	Multi-processor Bit	Parity Bit
0	0	0	0	0	Asyn-chronous mode	8-bit data	Absent	Absent
				1				
			1	0				Present
				1				
	1		0	0		7-bit data		Absent
				1				
			1	0				Present
				1				
	0	1	—	0	Asyn-chronous mode (multi-processor format)	8-bit data	Present	Absent
			—	1				
	1		—	0		7-bit data		
			—	1				
1	—	—	—	—	Synchronous mode	8-bit data	Absent	

		1			rate
1	0	0	Synchronous mode	Internal	Outputs the serial clock
		1			
	1	0		External	Inputs the serial clock
		1			

13.3.2 Operation in Asynchronous Mode

In asynchronous mode, each transmitted or received character begins with a start bit a one or two stop bits. Serial communication is synchronized one character at a time.

The transmitting and receiving sections of the SCI are independent, so full-duplex cor is possible. The transmitter and the receiver are both double-buffered, so data can be read while transmitting and receiving are in progress, enabling continuous transmittin receiving.

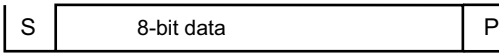
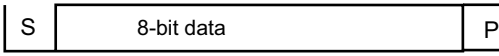
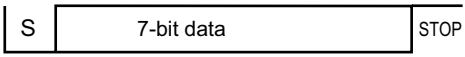
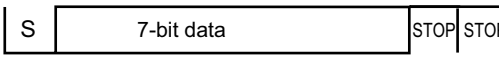
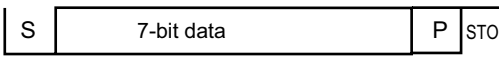
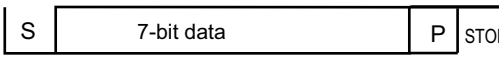
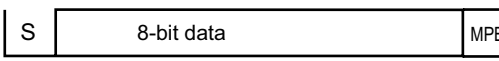
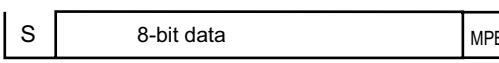
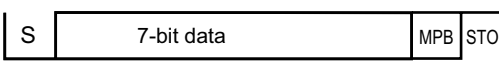
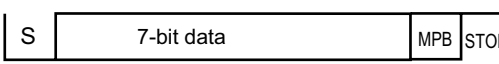
Figure 13.2 shows the general format of asynchronous serial communication. In asyn serial communication the communication line is normally held in the mark (high) stat monitors the line and starts serial communication when the line goes to the space (low indicating a start bit. One serial character consists of a start bit (low), data (LSB first) (high or low), and one or two stop bits (high), in that order.

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

Figure 13.2 Data Format in Asynchronous Communication (Example: 8-Bit Data with Parity and 2 Stop Bits)

Communication Formats

Table 13.10 shows the 12 communication formats that can be selected in asynchronous format is selected by settings in SMR.

0	1	0	0	
0	1	0	1	
1	0	0	0	
1	0	0	1	
1	1	0	0	
1	1	0	1	
0	—	1	0	
0	—	1	1	
1	—	1	0	
1	—	1	1	

Legend

S: Start bit

STOP: Stop bit

P: Parity bit

MPB: Multiprocessor bit

When the SCI is operated on an internal clock, it can output a clock signal at the SCK p
frequency of this output clock is equal to the bit rate. The phase is aligned as shown in
so that the rising edge of the clock occurs at the center of each transmit data bit.

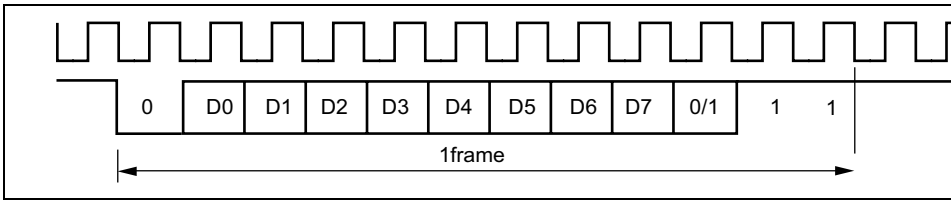


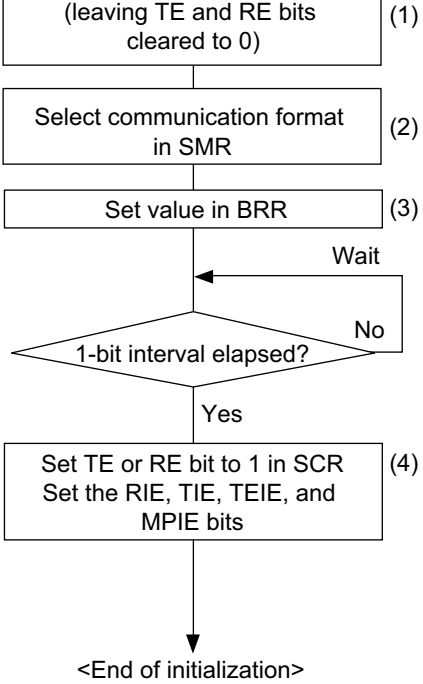
Figure 13.3 Phase Relationship between Output Clock and Serial Data (Asynchronous Mode)

Transmitting and Receiving Data

SCI Initialization (Asynchronous Mode): Before transmitting or receiving data, clear
RE bits to 0 in SCR, then initialize the SCI as follows.

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

When an external clock is used the clock should not be stopped during initialization or
operation, since operation will be unreliable in this case.



RIE, TIE, TEIE, MPIE, TE, and RE
0. If clock output is selected in
asynchronous mode, clock output
immediately after the setting is ma
SCR.

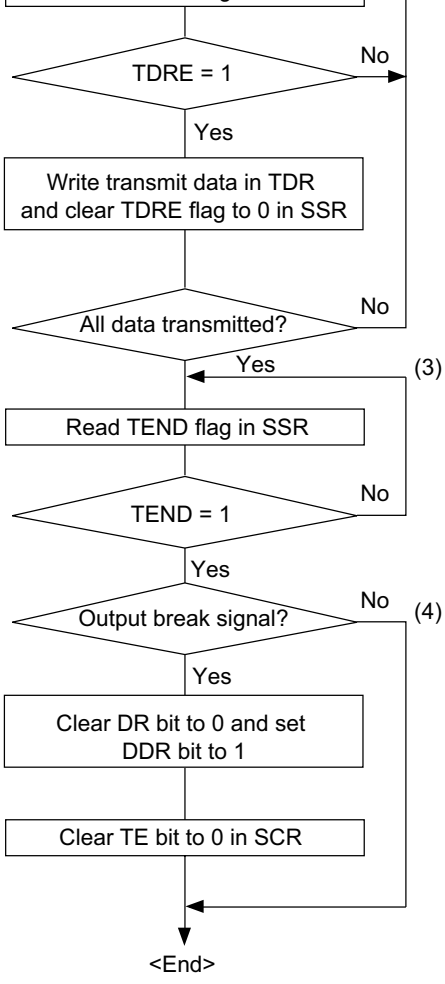
(2) Select the communication format i

(3) Write the value corresponding to t
rate in BRR.
This step is not necessary when a
external clock is used.

(4) Wait for at least the interval requir
transmit or receive one bit, then s
TE or RE bit to 1 in SCR. Set the
TIE, TEIE, and MPIE bits as nece
Setting the TE or RE bit enables t
to use the TxD or RxD pin.

Note: In simultaneous transmitting and receiving, the TE and RE bits should be cl
0 or set to 1 simultaneously.

Figure 13.4 Sample Flowchart for SCI Initialization



flag to 0.

(3) To continue transmitting serial data:
after checking that the TDRE flag is 1, inc
data can be written, write data in TDR, th
TDRE flag to 0. When the DMAC is activ
transmit-data-empty interrupt request (TX
data in TDR, the TDRE flag is checked ar
automatically.

(4) To output a break signal at the end of ser
transmission:
set the DDR bit to 1 and clear the DR bit
clear the TE bit to 0 in SCR.

Figure 13.5 Sample Flowchart for Transmitting Serial Data

— Transmit data: 7 or 8 bits are output, LSB first.

— Parity bit or multiprocessor bit: One parity bit (even or odd parity),or one mult bit is output. Formats in which neither a parity bit nor a multiprocessor bit is o also be selected.

— Stop bit(s): One or two 1 bits (stop bits) are output.

— Mark state: Output of 1 bits continues until the start bit of the next transmit dat

- The SCI checks the TDRE flag when it outputs the stop bit. If the TDRE flag is 0, loads new data from TDR into TSR, outputs the stop bit, then begins serial transm next frame. If the TDRE flag is 1, the SCI sets the TEND flag to 1 in SSR, output bit, then continues output of 1 bits in the mark state. If the TEIE bit is set to 1 in S transmit-end interrupt (TEI) is requested at this time.

Figure 13.6 shows an example of SCI transmit operation in asynchronous mode.

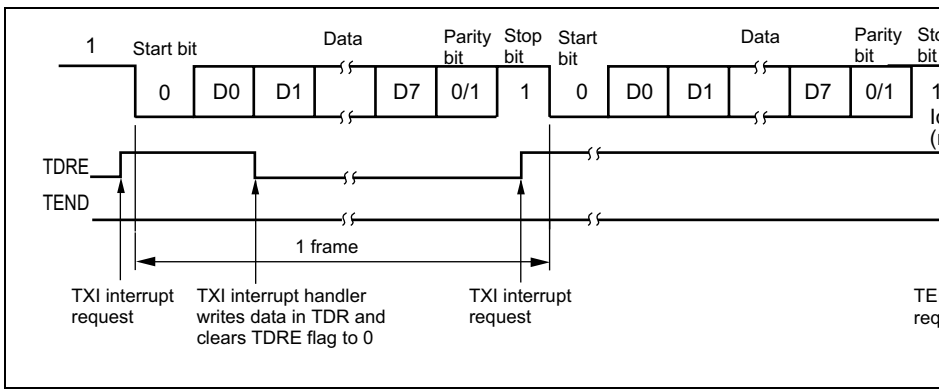
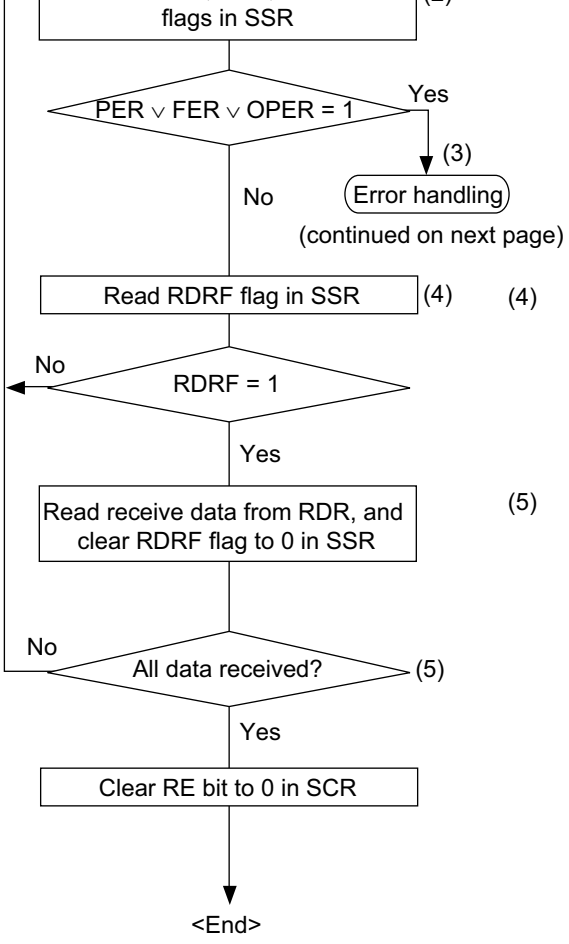


Figure 13.6 Example of SCI Transmit Operation in Asynchronous Mo (8-Bit Data with Parity and One Stop Bit)



PER, and FER flags in SSR to ide
error. After executing the necessa
handling, clear the ORER, PER, a
flags all to 0. Receiving cannot re
any of these flags remains set to
framing error occurs, the RxD pin
read to detect the break state.

(4) SCI status check and receive data
read SSR, check that the RDRF fl
to 1, then read receive data from
clear the RDRF flag to 0. Notifica
the RDRF flag has changed from
also be given by the RXI interrupt

(5) To continue receiving serial data:
check the RDRF flag, read RDR,
the RDRF flag to 0 before the stop
current frame is received. When
is activated by a receive-data-full
request (RXI) to read RDR, the R
is cleared automatically.

Figure 13.7 Sample Flowchart for Receiving Serial Data (1)

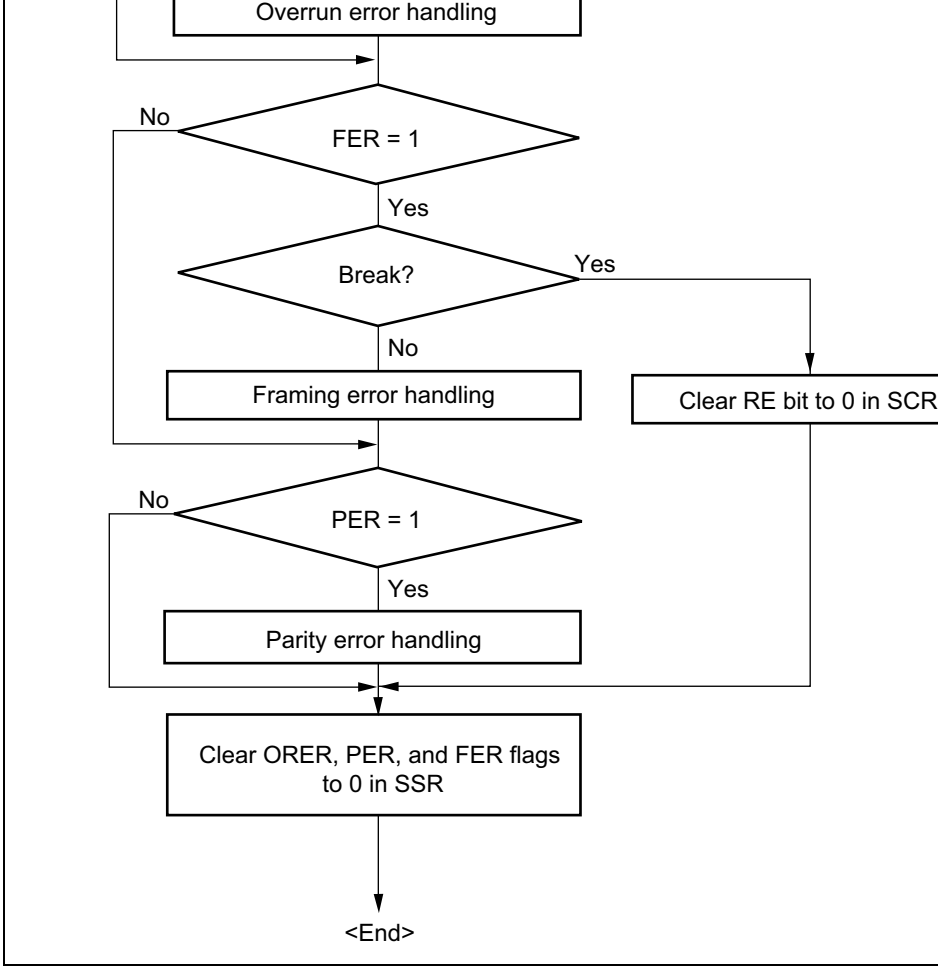


Figure 13.7 Sample Flowchart for Receiving Serial Data (2)

setting of the STOP bit in SMR.

— Stop bit check: The stop bit value must be 1. If there are two stop bits, only the
checked.

— Status check: The RDRF flag must be 0, indicating that the receive data can be t
from RSR into RDR.

If these all checks pass, the RDRF flag is set to 1 and the received data is stored in I
one of the checks fails (receive error*), the SCI operates as shown in table 13.11.

Note: * When a receive error occurs, further receiving is disabled. In receiving, the RD
not set to 1. Be sure to clear the error flags to 0.

- When the RDRF flag is set to 1, if the RIE bit is set to 1 in SCR, a receive-data-full
(RXI) is requested. If the ORER, PER, or FER flag is set to 1 and the RIE bit in SC
set to 1, a receive-error interrupt (ERI) is requested.

Table 13.11 Receive Error Conditions

Receive Error	Abbreviation	Condition	Data Transfer
Overrun error	ORER	Receiving of next data ends while RDRF flag is still set to 1 in SSR	Receive data is not tr from RSR to RDR
Framing error	FER	Stop bit is 0	Receive data is transf RSR to RDR
Parity error	PER	Parity of received data differs from even/odd parity setting in SMR	Receive data is transf RSR to RDR

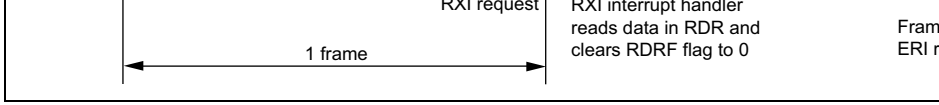


Figure 13.8 Example of SCI Receive Operation (8-Bit Data with Parity and One Stop Bit)

13.3.3 Multiprocessor Communication

The multiprocessor communication function enables several processors to share a sing communication line. The processors communicate in asynchronous mode using a forn additional multiprocessor bit (multiprocessor format).

In multiprocessor communication, each receiving processor is addressed by an ID. A communication cycle consists of an ID-sending cycle that identifies the receiving proc data-sending cycle. The multiprocessor bit distinguishes ID-sending cycles from data cycles.

The transmitting processor stars by sending the ID of the receiving processor with wh to communicate as data with the multiprocessor bit set to 1. Next the transmitting pro transmit data with the multiprocessor bit cleared to 0.

Receiving processors skip incoming data until they receive data with the multiprocess 1. When they receive data with the multiprocessor bit set to 1, receiving processors co data with their IDs. Processors with IDs not matching the received data skip further in until they again receive data with the multiprocessor bit set to 1. Multiple processors receive data in this way.

Figure 13.9 shows an example of communication among different processors using a multiprocessor format.

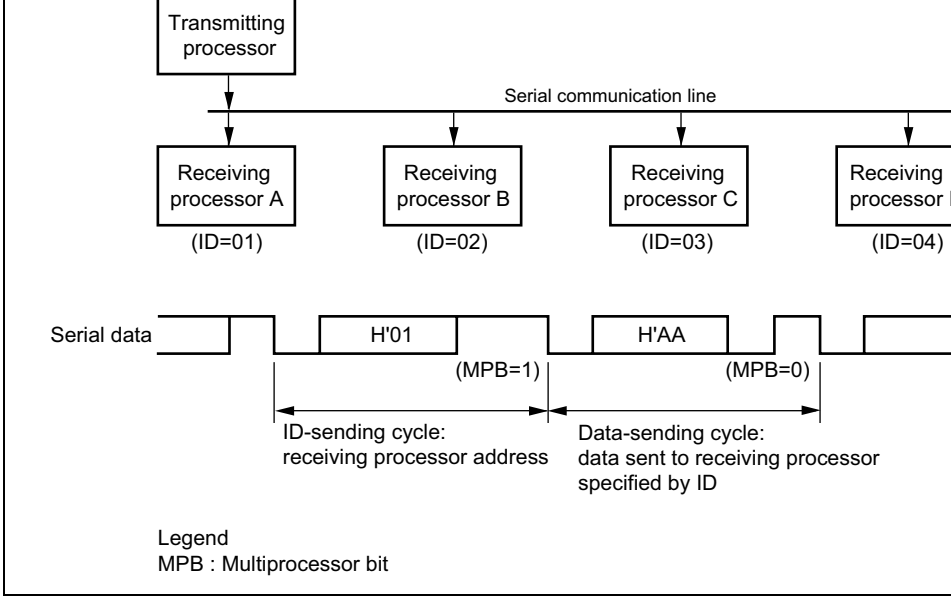
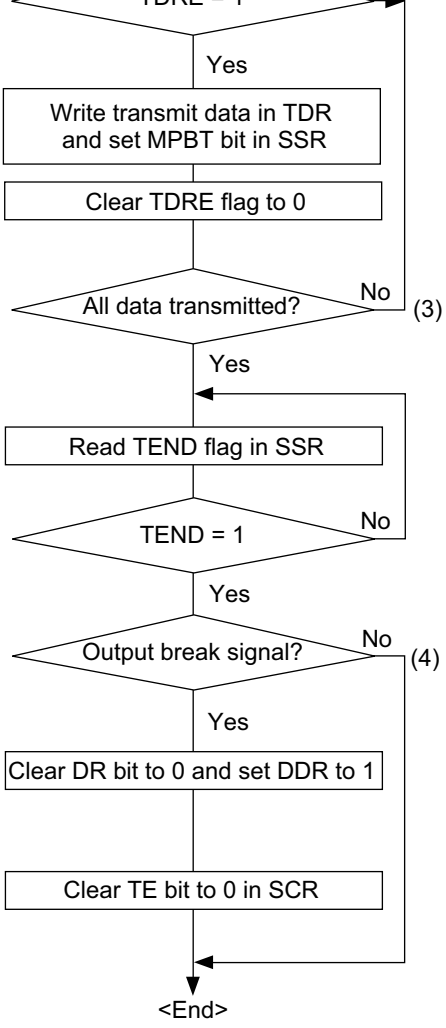


Figure 13.9 Example of Communication among Processors using Multiprocesso (Sending Data H'AA to Receiving Processor A)

Transmitting and Receiving Data

Transmitting Multiprocessor Serial Data: Figure 13.10 shows a sample flowchart fo transmitting multiprocessor serial data and indicates the procedure to follow.



after checking that the TDRE flag is
indicating that data can be written,
in TDR, then clear the TDRE flag to
the DMAC is activated by a transmi
empty interrupt request (TXI) to wri
TDR, the TDRE flag is checked and
automatically.

(4) To output a break signal at the end
transmission:
set the DDR bit to 1 and clear the D
then clear the TE bit to 0 in SCR.

Figure 13.10 Sample Flowchart for Transmitting Multiprocessor Serial

— Transmit data: 7 or 8 bits are output, LSB first.
— Multiprocessor bit: One multiprocessor bit (MPBT value) is output.
— Stop bit(s): One or two 1 bits (stop bits) are output.
— Mark state: Output of 1 bits continues until the start bit of the next transmit data

- The SCI checks the TDRE flag when it outputs the stop bit. If the TDRE flag is 0, t loads new data from TDR into TSR, outputs the stop bit, then begins serial transmis next frame. If the TDRE flag is 1, the SCI sets the TEND flag to 1 in SSR, outputs bit, then continues output of 1 bits in the mark state. If the TEIE bit is set to 1 in SC transmit-end interrupt (TEI) is requested at this time

Figure 13.11 shows an example of SCI transmit operation using a multiprocessor forma

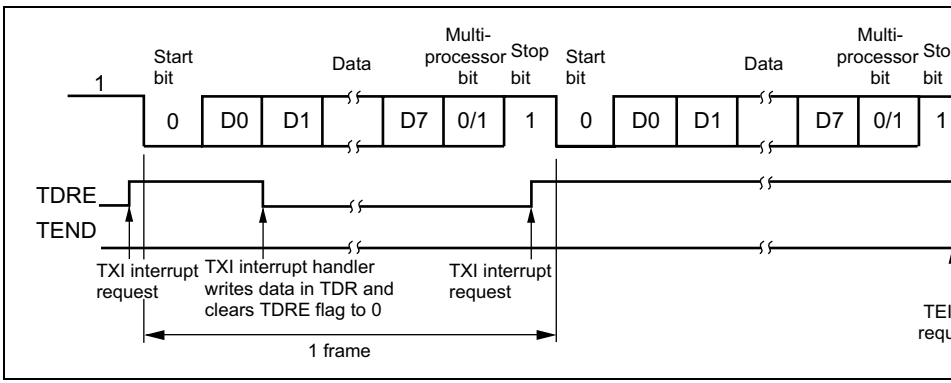
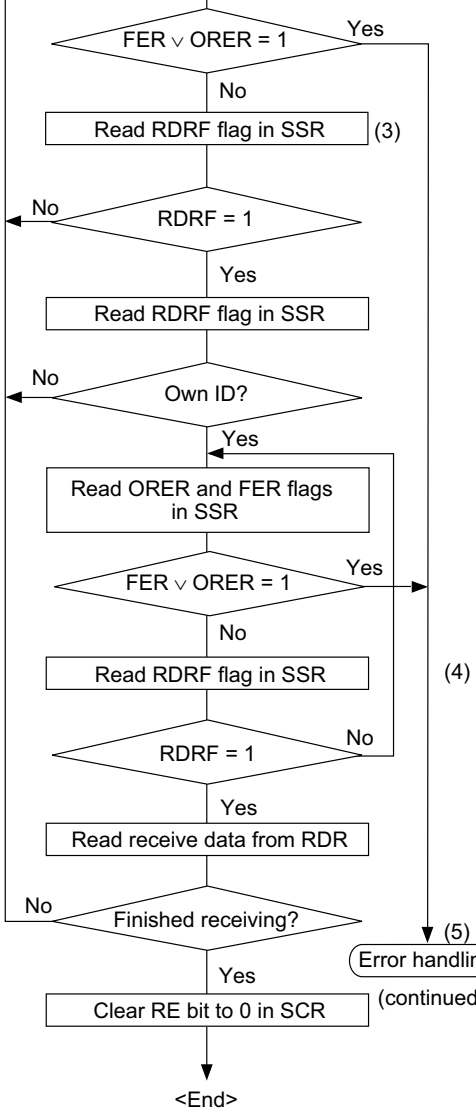


Figure 13.11 Example of SCI Transmit Operation (8-Bit Data with Multiprocessor Bit and One Stop Bit)

Receiving Multiprocessor Serial Data: Figure 13.12 shows a sample flowchart for re multiprocessor serial data and indicates the procedure to follow.



(4) SCI status check and data recei
read SSR, check that the RDRF
is set to 1, then read data from R

(5) Receive error handling and brea
detection:
if a receive error occurs, read th
ORER and FER flags in SSR to
identify the error. After executin
necessary error handling, clear t
ORER and FER flags both to 0.
Receiving cannot resume while
the ORER or FER flag remains s
1. When a framing error occurs
RxD pin can be read to detect th
break state.

Figure 13.12 Sample Flowchart for Receiving Multiprocessor Serial Dat

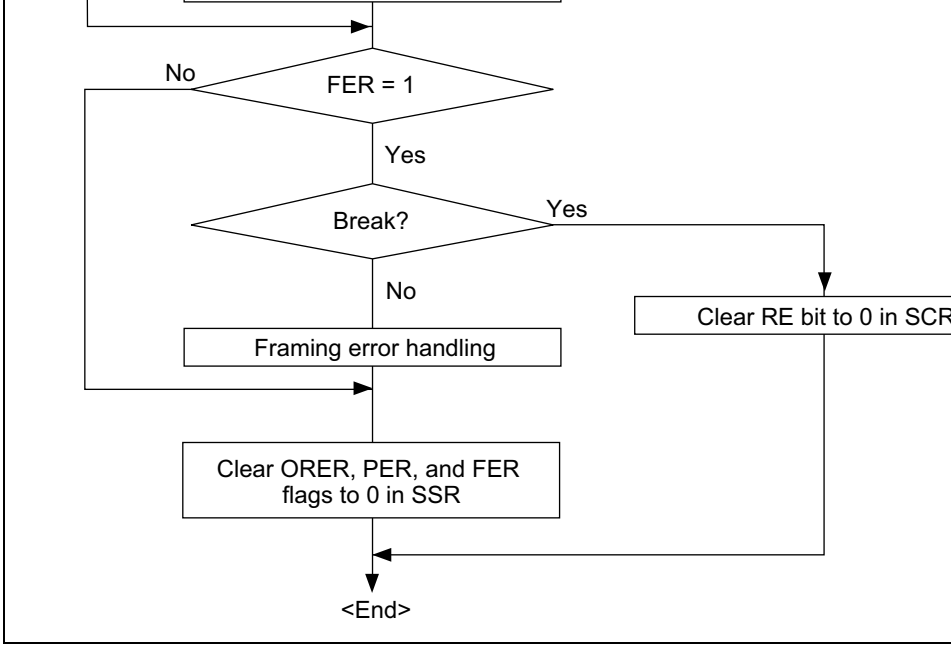


Figure 13.12 Sample Flowchart for Receiving Multiprocessor Serial Data

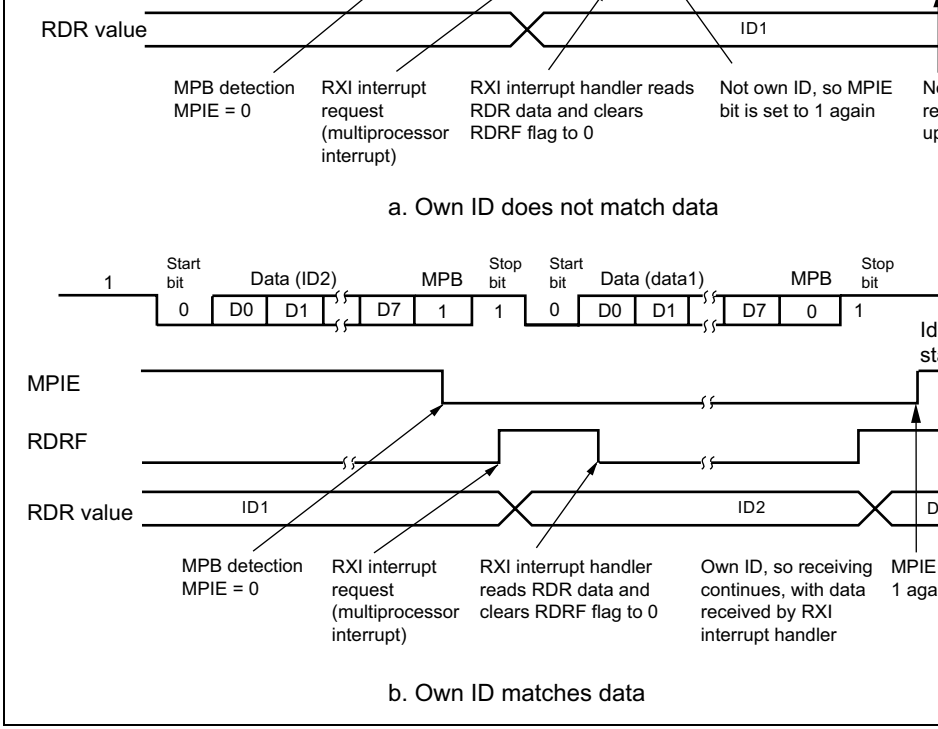


Figure 13.13 Example of SCI Receive Operation (8-Bit Data with Multiprocessor Bit and One Stop Bit)

Figure 13.14 shows the general format in synchronous serial communication.

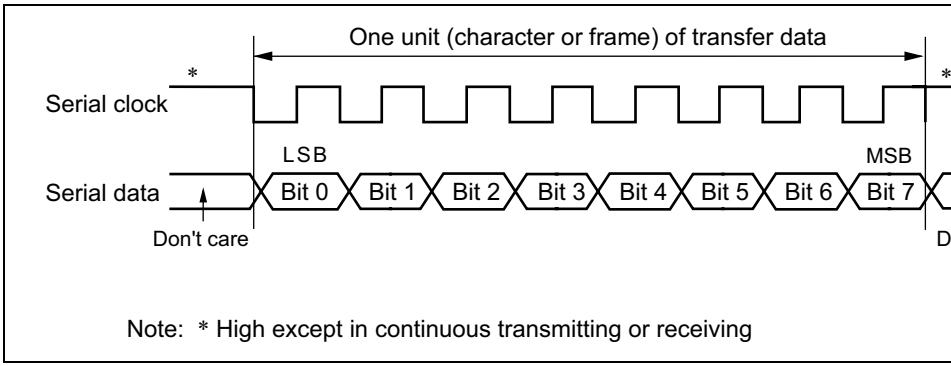


Figure 13.14 Data Format in Synchronous Communication

In synchronous serial communication, each data bit is placed on the communication lin falling edge of the serial clock to the next. Data is guaranteed valid at the rise of the se In each character, the serial data bits are transferred in order from LSB (first) to MSB (output of the MSB, the communication line remains in the state of the MSB. In synchr mode the SCI receives data by synchronizing with the rise of the serial clock.

Communication Format

The data length is fixed at 8 bits. No parity bit or multiprocessor bit can be added.

Clock

An internal clock generated by the on-chip baud rate generator or an external clock inp SCK pin can be selected by means of the C/$\overline{A}$ bit in SMR and the CKE1 and CKE0 bits See table 13.9 for details of SCI clock source selection.

When the SCI operates on an internal clock, it outputs the clock source at the SCK pin. clock pulses are output per transmitted or received character. When the SCI is not trans receiving, the clock signal remains in the high state. If receiving in single-character uni required, an external clock should be selected.

Figure 13.15 shows a sample flowchart for initializing the SCI.

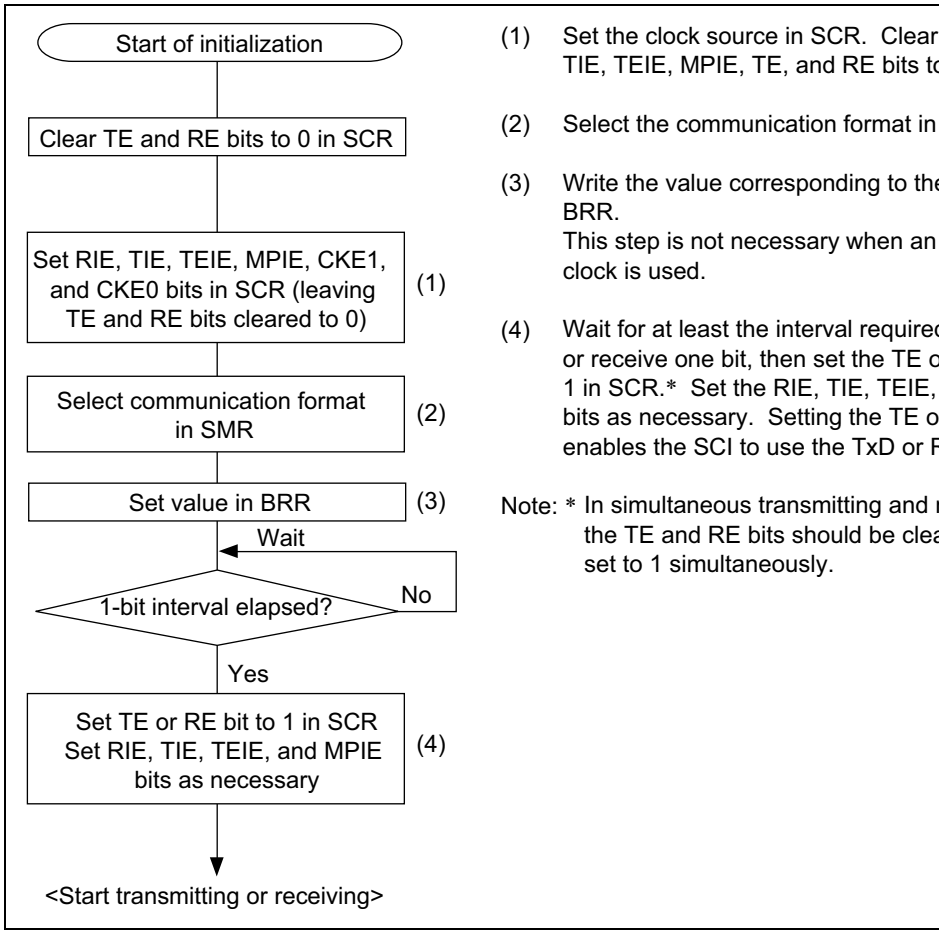
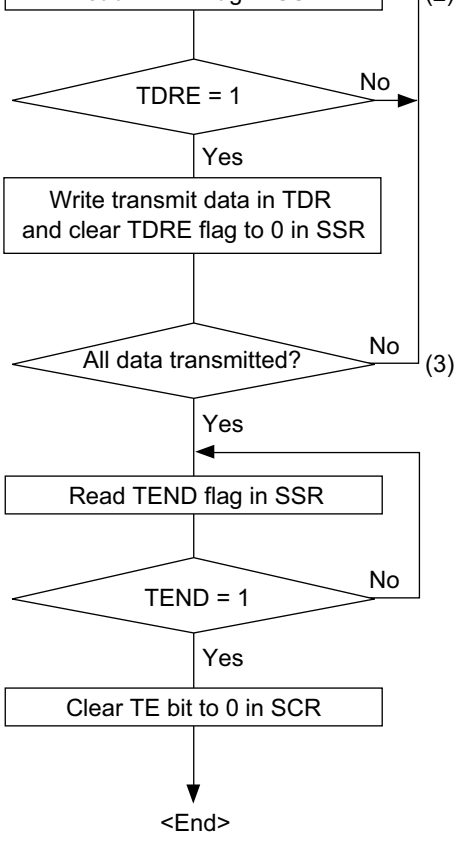


Figure 13.15 Sample Flowchart for SCI Initialization



write transmit data in TDR and clea
TDRE flag to 0.

(3) To continue transmitting serial data
checking that the TDRE flag is 1, in
that data can be written, write data
then clear the TDRE flag to 0. Whe
DMAC is activated by a transmit-da
interrupt request (TXI) to write data
the TDRE flag is checked and clear
automatically.

Figure 13.16 Sample Flowchart for Serial Transmitting

from the TxD pin n order from LSB (bit 0) to MSB (bit 7).

- The SCI checks the TDRE flag when it outputs the MSB (bit 7). If the TDRE flag SCI loads data from TDR into TSR and begins serial transmission of the next fram TDRE flag is 1, the SCI sets the TEND flag to 1 in SSR, and after transmitting the 7), holds the TxD pin in the MSB state. If the TEIE bit is set to 1 in SCR, a transm interrupt (TEI) is requested at this time
- After the end of serial transmission, the SCK pin is held in a constant state.

Figure 13.17 shows an example of SCI transmit operation.

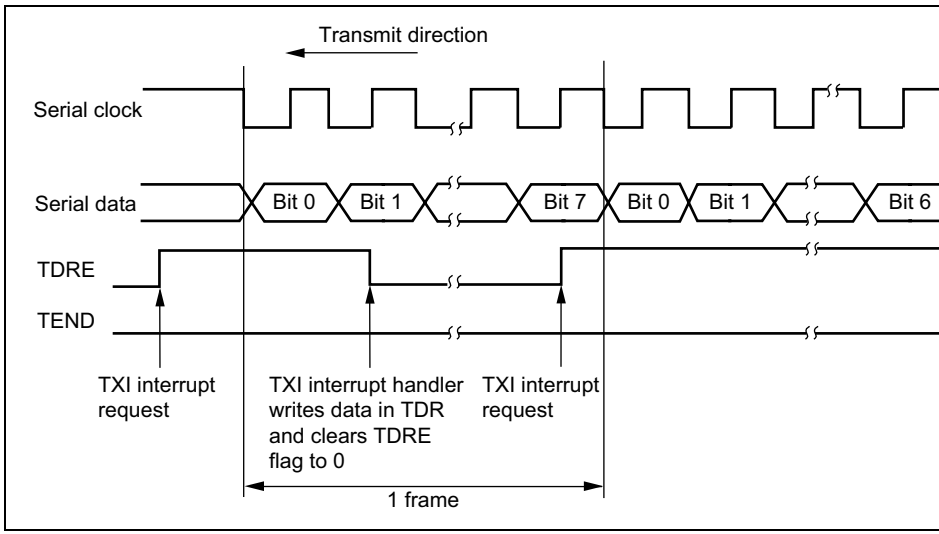
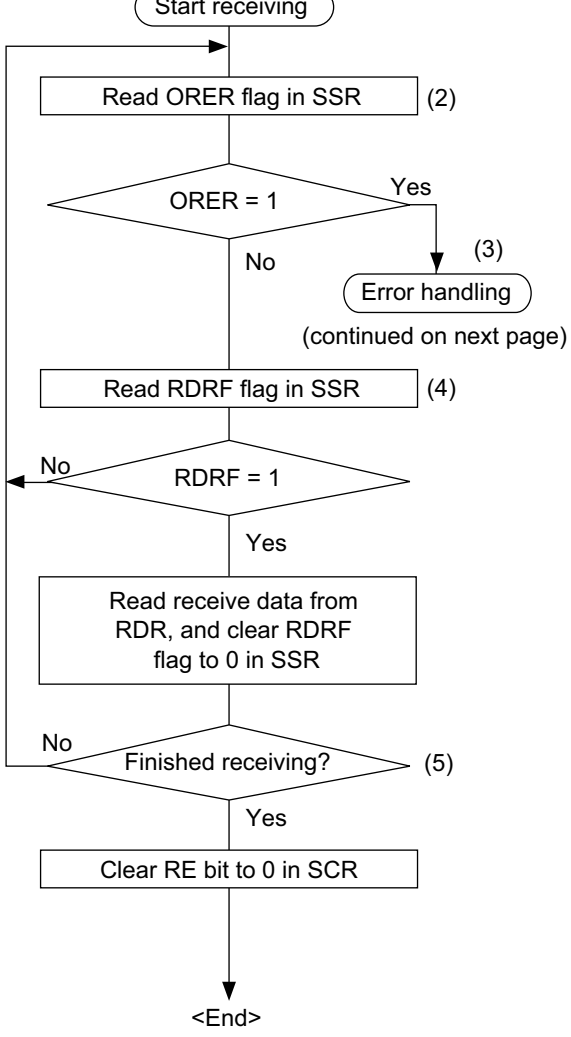


Figure 13.17 Example of SCI Transmit Operation



(2)(3) Receive error handling: if a error occurs, read the ORE SSR, then after executing t necessary error handling, c ORER flag to 0. Neither tra nor receiving can resume w ORER flag remains set to 1

(4) SCI status check and recei read: read SSR, check that flag is set to 1, then read re from RDR and clear the RD 0. Notification that the RDF has changed from 0 to 1 ca given by the RXI interrupt.

(5) To continue receiving seria check the RDRF flag, read clear the RDRF flag to 0 be MSB (bit 7) of the current fr received. When the DMAC activated by a receive-data interrupt request (RXI) to re the RDRF flag is cleared automatically.

Figure 13.18 Sample Flowchart for Serial Receiving (1)

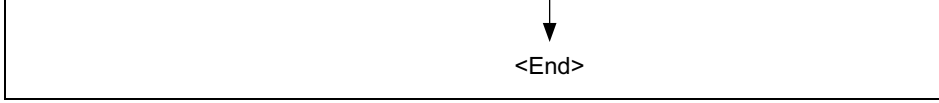


Figure 13.18 Sample Flowchart for Serial Receiving (2)

In receiving, the SCI operates as follows:

- The SCI synchronizes with serial clock input or output and synchronizes internally
- Receive data is stored in RSR in order from LSB to MSB.

  After receiving the data, the SCI checks that the RDRF flag is 0, so that receive da transferred from RSR to RDR. If this check passes, the RDRF flag is set to 1 and data is stored in RDR. If the checks fails (receive error), the SCI operates as show 13.11.

  When a receive error has been identified in the error check, subsequent transmit ar operations are disabled.
- When the RDRF flag is set to 1, if the RIE bit is set to 1 in SCR, a receive-data-ful (RXI) is requested. If the ORER flag is set to 1 and the RIE bit in SCR is also set receive-error interrupt (ERI) is requested.

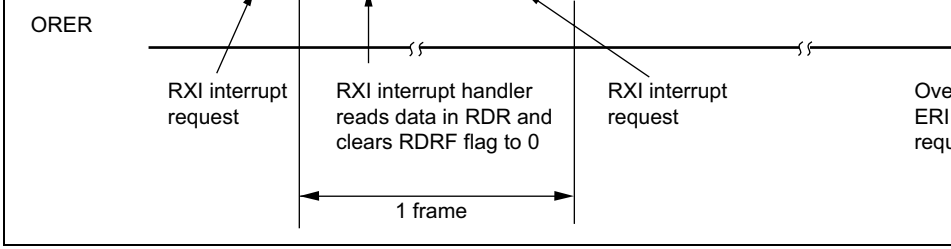
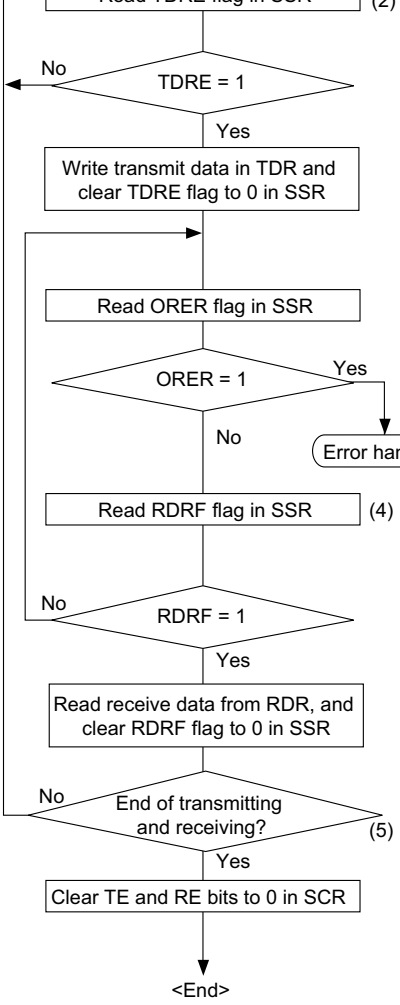


Figure 13.19 Example of SCI Receive Operation



that the TDRE flag is 1, then write transmit d
and clear the TDRE flag to 0.
Notification that the TDRE flag has changed
can also be given by the TXI interrupt.

(3) Receive error handling: if a receive error oc
ORER flag in SSR, then after executing the
error handling, clear the ORER flag to 0.
Neither transmitting nor receiving can resur
ORER flag remains set to 1.

(4) SCI status check and receive data read: rea
that the RDRF flag is 1, then read receive da
and clear the RDRF flag to 0. Notification th
flag has changed from 0 to 1 can also be giv
interrupt.

(5) To continue transmitting and receiving seria
the RDRF flag, read RDR, and clear the RD
before the MSB (bit 7) of the current frame i
Also check that the TDRE flag is set to 1, in
data can be written, write data in TDR, then
TDRE flag to 0 before the MSB (bit 7) of the
is transmitted. When the DMAC is activated
data-empty interrupt request (TXI) to write d
the TDRE flag is checked and cleared autor
When the DMAC is activated by a receive-d
interrupt request (RXI) to read RDR, the RD
cleared automatically.

Note: When switching from transmitting or receiving to simultaneous transmitting and receiving, clear b
and the RE bit to 0, then set both bits to 1 simultaneously.

Figure 13.20 Sample Flowchart for Simultaneous Serial Transmitting and R

when the TEND flag is set to 1 in SSR. A TXI interrupt request can activate the DMAC data. Data transfer by the DMAC automatically clears the TDRE flag to 0. A TEI interr cannot activate the DMAC.

An RXI interrupt is requested when the RDRF flag is set to 1 in SSR. An ERI interrupt requested when the ORER, PER, or FER flag is set to 1 in SSR. An RXI interrupt can a DMAC to transfer data. Data transfer by the DMAC automatically clears the RDRF fla ERI interrupt request cannot activate the DMAC.

The DMAC can be activated by interrupts from SCI channel 0.

Table 13.12 SCI Interrupt Sources

Interrupt Source	Description	Priorit
ERI	Receive error (ORER, FER, or PER)	High
RXI	Receive data register full (RDRF)	
TXI	Transmit data register empty (TDRE)	
TEI	Transmit end (TEND)	Low

13.5 Usage Notes

13.5.1 Notes on Use of SCI

Note the following points when using the SCI.

TDR Write and TDRE Flag: The TDRE flag in SSR is a status flag indicating the loa transmit data from TDR to TSR. The SCI sets the TDRE flag to 1 when it transfers dat TDR to TSR.

Data can be written into TDR regardless of the state of the TDRE flag. If new data is v TDR when the TDRE flag is 0, the old data stored in TDR will be lost because this data yet been transferred to TSR. Before writing transmit data in TDR, be sure to check tha flag is set to 1.

0	0	1	0	○	Framing error
0	0	0	1	○	Parity error
1	1	1	0	×	Overrun error + framing err
1	1	0	1	×	Overrun error + parity error
0	0	1	1	○	Framing error + parity error
1	1	1	1	×	Overrun error + framing err error

Notes: ○ : Receive data is transferred from RSR to RDR.
× : Receive data is not transferred from RSR to RDR.

Break Detection and Processing: Break signals can be detected by reading the RxD when a framing error (FER) is detected. In the break state the input from the RxD pin all 0s, so the FER flag is set and the parity error flag (PER) may also be set. In the bre SCI receiver continues to operate, so if the FER flag is cleared to 0 it will be set to 1 a

Sending a Break Signal: The input/output condition and level of the TxD pin are det DR and DDR bits. This feature can be used to send a break signal.

After the serial transmitter is initialized, the DR value substitutes for the mark state un bit is set to 1 (the TxD pin function is not selected until the TE bit is set to 1). The DD bits should therefore be set to 1 beforehand.

To send a break signal during serial transmission, clear the DR bit to 0 , then clear the When the TE bit is cleared to 0 the transmitter is initialized, regardless of its current st TxD pin becomes an input/output outputting the value 0.

Receive Error Flags and Transmitter Operation (Synchronous Mode Only): Whe error flag (ORER, PER, or FER) is set to 1 the SCI will not start transmitting, even if flag is cleared to 0. Be sure to clear the receive error flags to 0 when starting to transm that clearing the RE bit to 0 does not clear the receive error flags to 0.

Receive Data Sampling Timing in Asynchronous Mode and Receive Margin: In a mode the SCI operates on a base clock with 16 times the bit rate frequency. In receivi

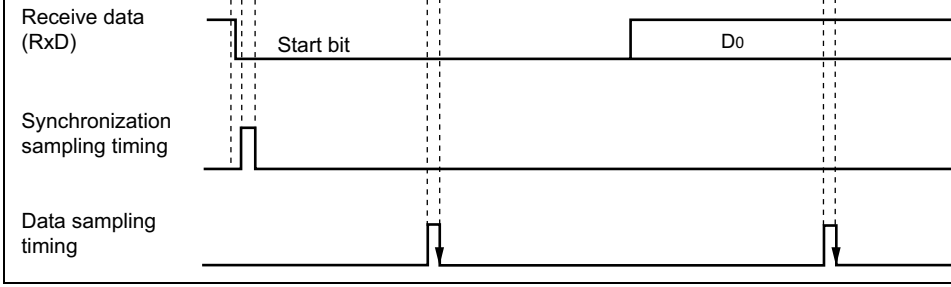


Figure 13.21 Receive Data Sampling Timing in Asynchronous Mode

The receive margin in asynchronous mode can therefore be expressed as shown in equa

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

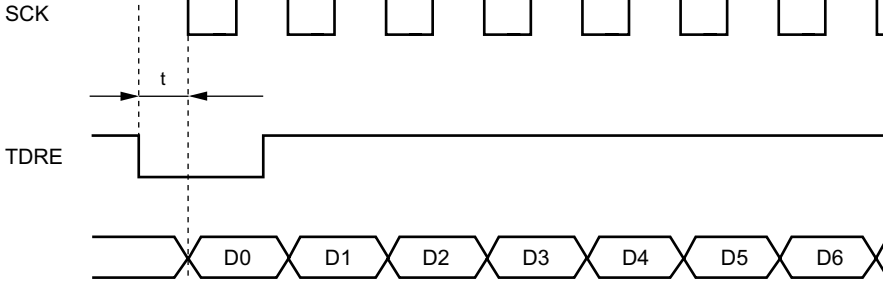
M: Receive margin (%)
N: Ratio of clock frequency to bit rate (N = 16)
D: Clock duty cycle (L = 0 to 1.0)
L: Frame length (L = 9 to 12)
F: Absolute deviation of clock frequency

From equation (1), if F = 0 and D = 0.5, the receive margin is 46.875%, as given by equ

$$D = 0.5,\ F = 0$$

$$M = (0.5 - \frac{1}{2 \times 16}) \times 100\% = 46.875\% \quad \ldots\ldots\ldots (2)$$

This is a theoretical value. A reasonable margin to allow in system designs is 20% to 3



Note: In operation with an external clock source, be sure that t >4 states.

Figure 13.22 Example of Synchronous Transmission Using DMAC

4. Occurrence of low-level output (see figure 13.23)

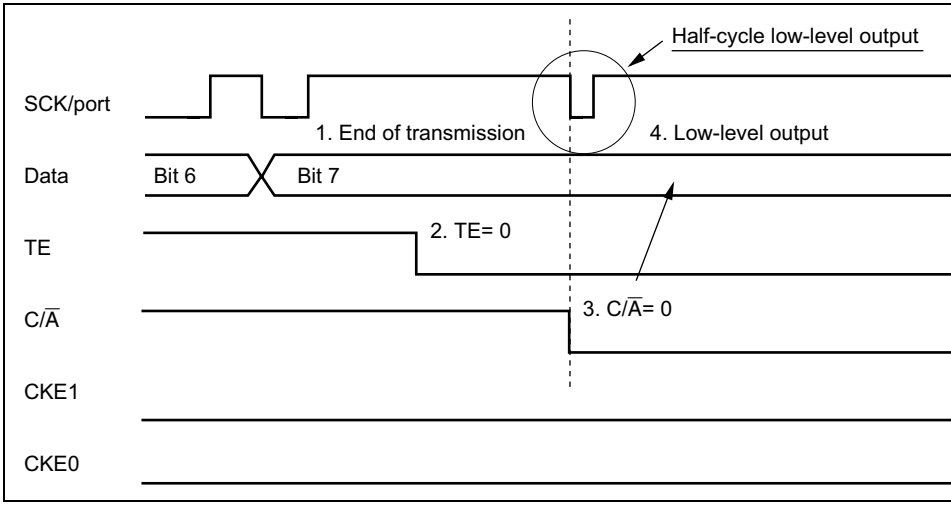


Figure 13.23 Operation when Switching from SCK Pin Function to Port Pin F

4. C/A bit = 0 ... switchover to port output

5. CKE1 bit = 0

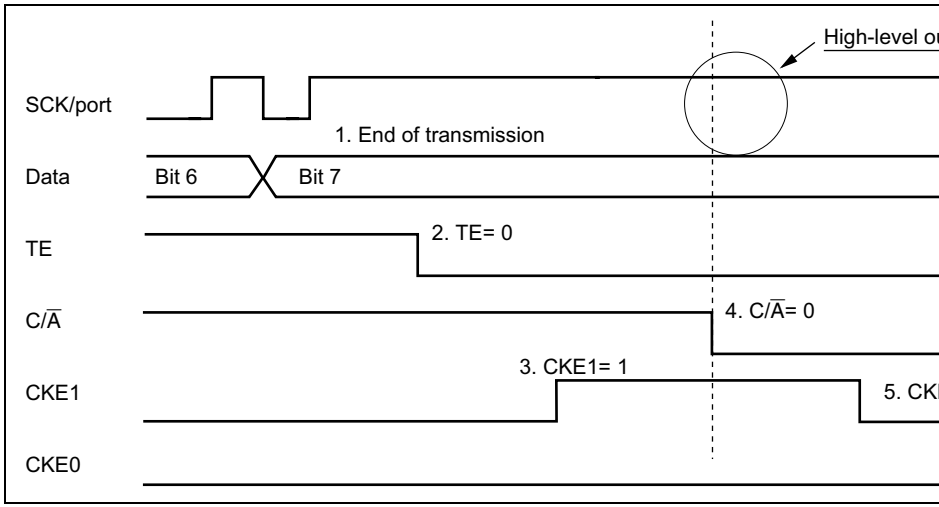


Figure 13.24 Operation when Switching from SCK Pin Function to Port Pin (Example of Preventing Low-Level Output)

controlled by a register setting.

14.1.1 Features

Features of the smart card interface supported by the H8/3028 Group are listed below.

- Asynchronous communication
  - 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
- Built-in baud rate generator allows any bit rate to be selected
- Three interrupt sources
  - There are three interrupt sources—transmit-data-empty, receive-data-full, and transmit/receive error—that can issue requests independently.
  - The transmit-data-empty interrupt and receive-data-full interrupt can activate t controller (DMAC) to execute data transfer.

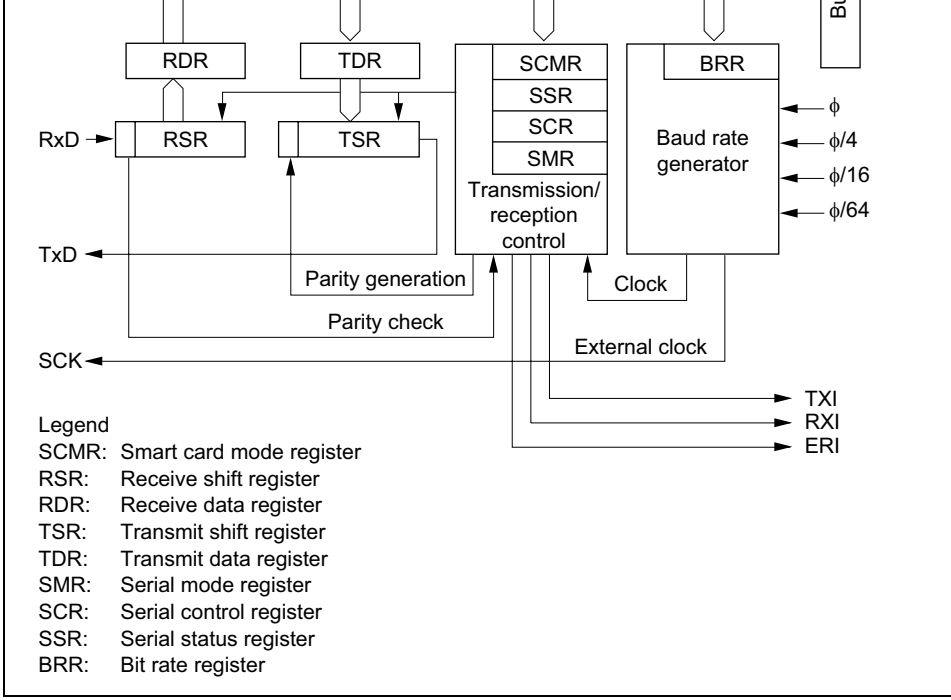


Figure 14.1 Block Diagram of Smart Card Interface

14.1.3 Pin Configuration

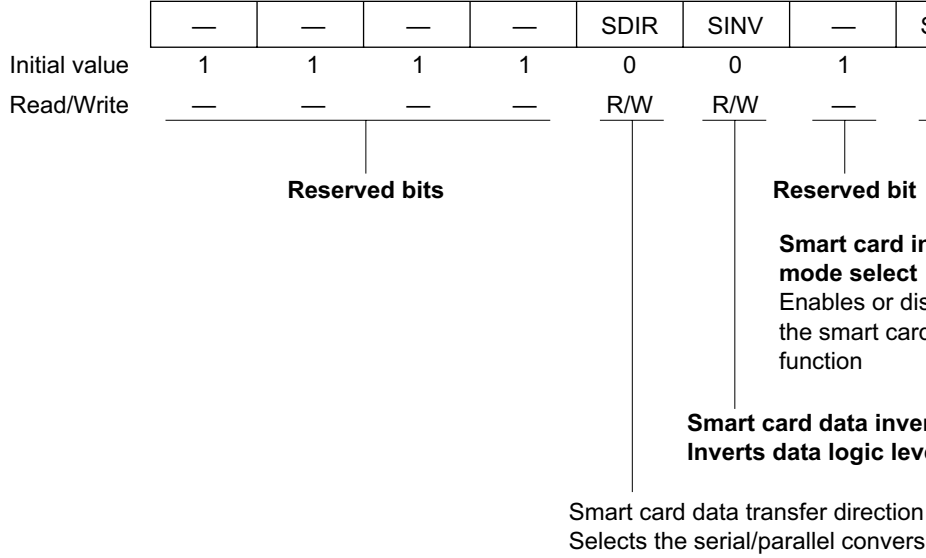
Table 14.1 shows the smart card interface pins.

Table 14.1 Smart Card Interface Pins

Pin Name	Abbreviation	I/O	Function
Serial clock pin	SCK	I/O	Clock input/output
Receive data pin	RxD	Input	Receive data input
Transmit data pin	TxD	Output	Transmit data output

0	H'FFFB0	Serial mode register	SMR	R/W	H'0
	H'FFFB1	Bit rate register	BRR	R/W	H'F
	H'FFFB2	Serial control register	SCR	R/W	H'0
	H'FFFB3	Transmit data register	TDR	R/W	H'F
	H'FFFB4	Serial status register	SSR	R/(W)*2	H'8
	H'FFFB5	Receive data register	RDR	R	H'0
	H'FFFB6	Smart card mode register	SCMR	R/W	H'F
1	H'FFFB8	Serial mode register	SMR	R/W	H'0
	H'FFFB9	Bit rate register	BRR	R/W	H'F
	H'FFFBA	Serial control register	SCR	R/W	H'0
	H'FFFBB	Transmit data register	TDR	R/W	H'F
	H'FFFBC	Serial status register	SSR	R/(W)*2	H'8
	H'FFFBD	Receive data register	RDR	R	H'0
	H'FFFBE	Smart card mode register	SCMR	R/W	H'F
2	H'FFFC0	Serial mode register	SMR	R/W	H'0
	H'FFFC1	Bit rate register	BRR	R/W	H'F
	H'FFFC2	Serial control register	SCR	R/W	H'0
	H'FFFC3	Transmit data register	TDR	R/W	H'F
	H'FFFC4	Serial status register	SSR	R/(W)*2	H'8
	H'FFFC5	Receive data register	RDR	R	H'0
	H'FFFC6	Smart card mode register	SCMR	R/W	H'F

Notes: 1. Lower 20 bits of the address in advanced mode.
2. Only 0 can be written in bits 7 to 3, to clear the flags.



SCMR is initialized to H'F2 by a reset and in standby 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.*[1]

Bit 3 SDIR	Description	
0	TDR contents are transmitted LSB-first	(In
	Receive data is stored LSB-first in RDR	
1	TDR contents are transmitted MSB-first	
	Receive data is stored MSB-first in RDR	

1	Inverted TDR contents are transmitted Receive data is inverted before storage in RDR

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

Bit 0—Smart Card Interface Mode Select (SMIF): Enables the smart card interface

Bit 0 SMIF	Description	
0	Smart card interface function is disabled	(
1	Smart card interface function is enabled	

Notes: 1. The function for switching between LSB-first and MSB-first mode can als
with the normal serial communication interface. Note that when the comm
format data length is set to 7 bits and MSB-first mode is selected for the se
be transferred, bit 0 of TDR is not transmitted, and only bits 7 to 1 of the re
are valid.

2. The data logic level inversion function can also be used with the normal se
communication interface. Note that, when inverting the serial data to be tr
parity transmission and parity checking is based on the number of high-lev
the serial data I/O pin, and not on the register value.

14.2.2 Serial Status Register (SSR)

The function of SSR bit 4 is modified in smart card interface mode. This change also
modification to the setting conditions for bit 2 (TEND).

Error signal status (ERS)
Status flag indicating that an error signal has been received

Note: * Only 0 can be written, to clear the flag.

Bits 7 to 5: These bits operate as in normal serial communication. For details see sectio Serial Status Register (SSR).

Bit 4—Error Signal Status (ERS): In smart card interface mode, this flag indicates th the error signal sent from the receiving device to the transmitting device. The smart car does not detection framing errors.

Bit 4 ERS	Description
0	Indicates normal transmission, with no error signal returned (In [Clearing conditions] The chip is reset, or enters standby mode or module stop mode Software reads ERS while it is set to 1, then writes 0.
1	Indicates that the receiving device sent an error signal reporting a parity err [Setting condition] A low error signal was sampled.

Note: Clearing the TE bit to 0 in SCR does not affect the ERS flag, which retains its pr value.

Bits 3 to 0: These bits operate as in normal serial communication. For details see sectio Serial Status Register (SSR). The setting conditions for transmit end (TEND), however modified as follows.

- The chip is reset or enters standby mode.
- The TE bit and FER/ERS bit are both cleared to 0 in SCR.
- TDRE is 1 and FER/ERS is 0 at a time 2.5 etu after the last bit of a 1-b character is transmitted (normal transmission).

Note: etu (Elementary time unit: the time for transfer of one bit)

14.2.3 Serial Mode Register (SMR)

The function of SMR bit 7 is modified in smart card interface mode. This change also modification to the function of bits 1 and 0 in the serial control register (SCR).

Bit	7	6	5	4	3	2	1
	GM	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1
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

Bit 7—GSM Mode (GM): With the normal smart card interface, this bit is cleared to this bit to 1 selects GSM mode, an additional mode for controlling the timing for setti TEND flag that indicates completion of transmission, and the type of clock output use details of the additional clock output control mode are specified by the CKE1 and CK the serial control register (SCR).

Bit 7 GM	Description
0	Normal smart card interface mode operation The TEND flag is set 12.5 etu after the beginning of the start bit. Clock output on/off control only.
1	GSM mode smart card interface mode operation The TEND flag is set 11.0 etu after the beginning of the start bit. Clock output on/off and fixed-high/fixed-low control.

Note: etu (Elementary time unit: the time for transfer of one bit)

	TIE	RIE	TE	RE	MPIE	TEIE	CKE1
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

Bits 7 to 2: These bits operate as in normal serial communication. For details see sectio Serial Control Register (SCR).

Bits 1 and 0—Clock Enable 1 and 0 (CKE1, CKE0): These bits select the SCI clock enable or disable clock output from the SCK pin. In smart card interface mode, it is pos specify a fixed high level or fixed low level for the clock output, in addition to the usua between enabling and disabling of the clock output.

Bit 7 GM	Bit 1 CKE1	Bit 0 CKE0	Description	
0	0	0	Internal clock/SCK pin is I/O port	(Init
		1	Internal clock/SCK pin is clock output	
1		0	Internal clock/SCK pin is fixed at low output	
		1	Internal clock/SCK pin is clock output	
	1	0	Internal clock/SCK pin is fixed at high output	
		1	Internal clock/SCK pin is clock output	

14.3 Operation

14.3.1 Overview

The main features 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 units: the time for tra one bit) is provided between the end of the parity bit and the start of the next frame.

Figure 14.2 shows a pin connection diagram for the smart card interface.

In communication with a smart card, since both transmission and reception are carried single data transmission line, the TxD pin and RxD pin should both be connected to th data transmission line should be pulled up to V_{CC} with a resistor.

When the smart card uses the clock generated on the smart card interface, the SCK pi input to the CLK pin of the smart card. If the smart card uses an internal clock, this co unnecessary.

The reset signal should be output from one of the H8/3028 Group's generic ports.

In addition to these pin connections, power and ground connections will normally also necessary.

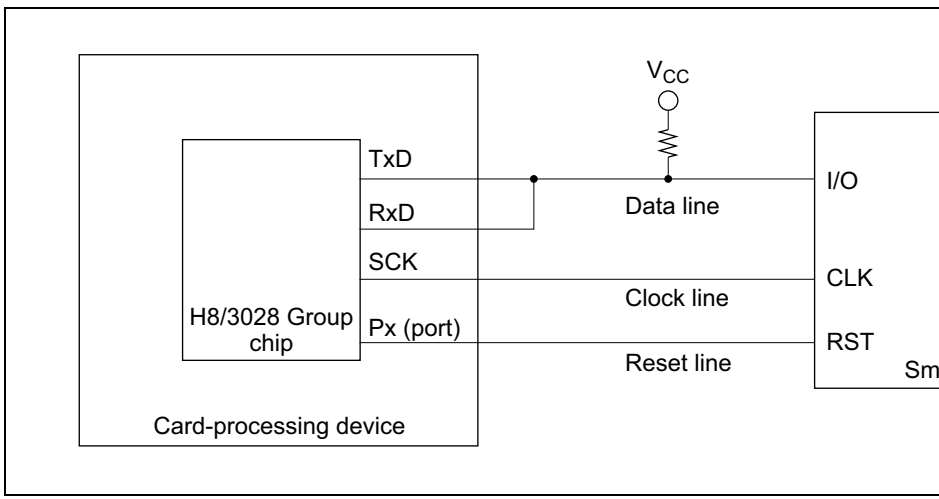


Figure 14.2 Smart Card Interface Connection Diagram

Note: A loop-back test can be performed by setting both RE and TE to 1 without co smart card.

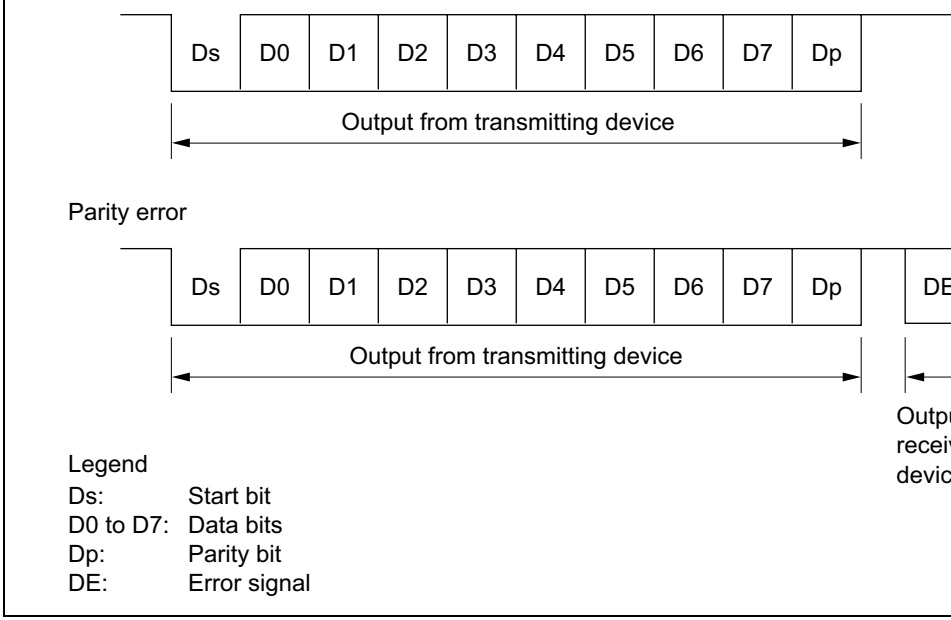


Figure 14.3 Smart Card Interface Data Format

The operating sequence is as follows.

1. When the data line is not in use it is in the high-impedance state, and is fixed high w up resistor.
2. The transmitting device 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 state. line is pulled high with a pull-up resistor.
4. The receiving device carries out a parity check. If there is no parity error and the da received normally, the receiving device waits for reception of the next data. If a par occurs, however, the receiving device outputs an error signal (DE, low-level) to req retransmission of the data. After outputting the error signal for the prescribed length the receiving device places the signal line in the high-impedance state again. The si pulled high again by a pull-up resistor.

Table 14.3 Smart Card Interface Register Settings

		Bit						
Register	**Address**[*1]	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**
SMR	H'FFFB0	GM	0	1	O/$\overline{E}$	1	0	CKS1
BRR	H'FFFB1	BRR7	BRR6	BRR5	BRR4	BRR3	BRR2	BRR1
SCR	H'FFFB2	TIE	RIE	TE	RE	0	0	CKE1
TDR	H'FFFB3	TDR7	TDR6	TDR5	TDR4	TDR3	TDR2	TDR1
SSR	H'FFFB4	TDRE	RDRF	ORER	ERS	PER	TEND	0
RDR	H'FFFB5	RDR7	RDR6	RDR5	RDR4	RDR3	RDR2	RDR1
SCMR	H'FFFB6	—	—	—	—	SDIR	SINV	—

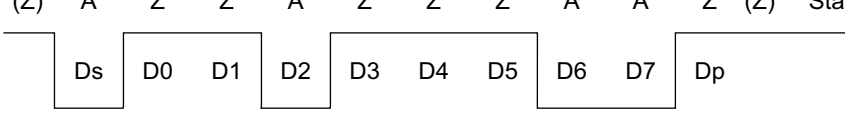
Notes: — Unused bit.
1. Lower 20 bits of the address in advanced mode.
2. When GM is cleared to 0 in SMR, the CKE1 bit must also be cleared to 0.

Serial Mode Register (SMR) Settings: Clear the GM bit to 0 when using the normal interface mode, or set to 1 when using GSM mode. Clear the O/$\overline{E}$ bit to 0 if the smart direct convention type, or set to 1 if of the inverse convention type.

Bits CKS1 and CKS0 select the clock source of the built-in baud rate generator. See s 14.3.5, Clock.

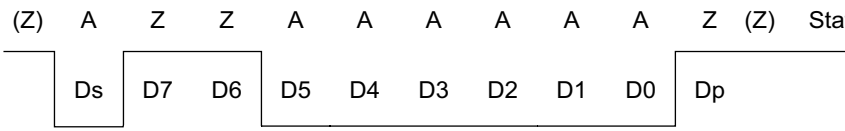
Bit Rate Register (BRR) Settings: BRR is used to set the bit rate. See section 14.3.5 the method of calculating the value to be set.

Serial Control Register (SCR) Settings: The TIE, RIE, TE, and RE bits have their n communication functions. See section 13, Serial Communication Interface, for details and CKE0 bits specify clock output. To disable clock output, clear these bits to 00; to output, set these bits to 01. Clock output is not performed when the GM bit is set to 1 Clock output can also be fixed low or high.



With the direct convention type, the logic 0 level corresponds to state Z and the log
state A, and transfer is performed in LSB-first order. In the example above, the first
data is H'3B. The parity bit is 1, following the even parity rule designated for smart

2. Indirect Convention (SDIR = SINV = O/$\overline{E}$ = 1)



With the indirect convention type, the logic 1 level corresponds to state Z and the lo
to state A, and transfer is performed in MSB-first order. In the example above, the f
character data is H'3F. The parity bit is 0, corresponding to state Z, following the ev
rule designated for smart cards.

In the H8/3028 Group, inversion specified by the SINV bit applies only to the data
D0. For parity bit inversion, the O/$\overline{E}$ bit in SMR must be set to odd parity mode. Th
to both transmission and reception.

$$B = \frac{\phi}{1488 \times 2^{2n-1} \times (N + 1)} \times 10^6$$

where, N: BRR setting ($0 \leq N \leq 255$)
B: Bit rate (bit/s)
ϕ: Operating frequency (MHz)
n: See table 14.4

Table 14.4 n-Values of CKS1 and CKS0 Settings

n	CKS1	CKS0
0	0	0
1		1
2	1	0
3		1

Note: If the gear function is used to divide the clock frequency, use the divided freque
calculate the bit rate. The equation above applies directly to 1/1 frequency divis

Table 14.5 Bit Rates (bits/s) for Various BRR Settings (When n = 0)

	ϕ (MHz)							
N	**7.1424**	**10.00**	**10.7136**	**13.00**	**14.2848**	**16.00**	**18.00**	**20.00**
0	9600.0	13440.9	14400.0	17473.1	19200.0	21505.4	24193.5	26881.
1	4800.0	6720.4	7200.0	8736.6	9600.0	10752.7	12096.8	13440.
2	3200.0	4480.3	4800.0	5824.4	6400.0	7168.5	8064.5	8960.6

Note: Bit rates are rounded off to one decimal place.

	7.1424		10.00		10.7136		13.00		14.2848		16.00		18.00		20.00	
bit/s	N	Error	N	Error	N	Error	N	Error	N	Error	N	Error	N	Error	N	Erro
9600	0	0.00	1	30	1	25	1	8.99	1	0.00	1	12.01	2	15.99	2	6.66

Table 14.7 Maximum Bit Rates for Various Frequencies (Smart Card Interface

ϕ (MHz)	Maximum Bit Rate (bits/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
25.00	33602	0	0

The bit rate error is given by the following equation:

$$\text{Error (\%)} = \left(\frac{\phi}{1488 \times 2^{2n-1} \times B \times (N + 1)} \times 10^6 - 1 \right) \times 100$$

mode register (SMR). Clear the C/A, CHR, and MP bits to 0, and set the STOP an
1.

4. Set the SMIF, SDIR, and SINV bits in the smart card mode register (SCMR).
   When the SMIF bit is set to 1, the TxD pin and RxD pin are both switched from po
   pin functions and go to the high-impedance state.
5. Set a value corresponding to the desired bit rate in the bit rate register (BRR).
6. Set the CKE0 bit in SCR. Clear the TIE, RIE, TE, RE, MPIE, TEIE, and CKE1 bi
   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 n
   bit and RE bit at the same time, except for self-diagnosis.

Transmitting Serial Data: As data transmission in smart card mode involves error si
sampling and retransmission processing, the processing procedure is different from th
normal SCI. Figure 14.5 shows a sample transmission processing flowchart.

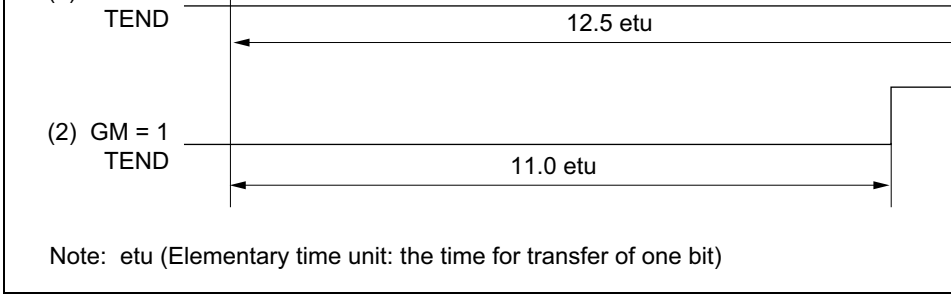
1. Perform smart card interface mode initialization as described in Initialization abov
2. Check that the FER/ERS error flag is cleared to 0 in SSR.
3. Repeat steps 2 and 3 until it can be confirmed that the TEND flag is set to 1 in SSI
4. Write the transmit data in TDR, clear the TDRE flag to 0, and perform the transmi
   The TEND flag is cleared to 0.
5. To continue transmitting data, go back to step 2.
6. To end transmission, clear the TE bit to 0.

The above processing may include interrupt handling DMA transfer.

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) will be requested. If an er
transmission and the ERS flag is set to 1 while the RIE bit is set to 1 and interrupt req
enabled, a transmit/receive-error interrupt (ERI) will be requested.

The timing of TEND flag setting depends on the GM bit in SMR (see figure 14.4).

If the TXI interrupt activates the DMAC, the number of bytes designated in the DMA
transmitted automatically, including automatic retransmission.



Note: etu (Elementary time unit: the time for transfer of one bit)

Figure 14.4 Timing of TEND Flag Setting

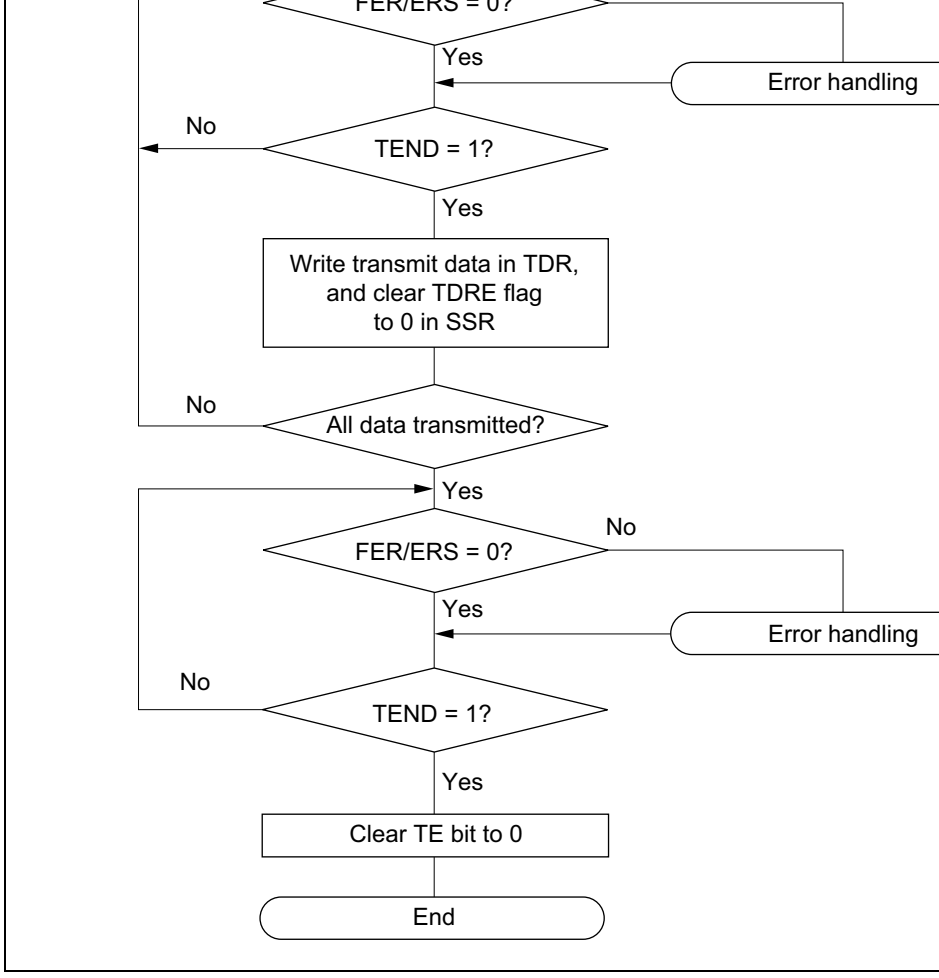
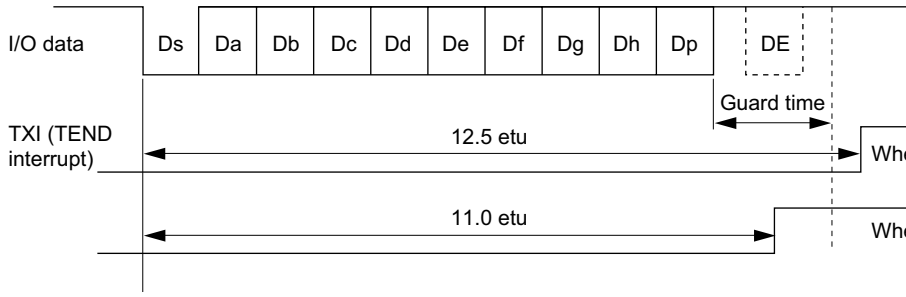


Figure 14.5 Sample Transmission Processing Flowchart

In case of normal transmission:	TEND flag is set
In case of transmit error:	ERS flag is set Steps 2 and 3 above are repeated unti TEND flag is set.

Note: When the ERS flag is set, it should be cleared until transfer of the last bit (D7 in L transmission, D0 in MSB-first transmission) of the retransmit data to be transmitte been completed.

Figure 14.6 Relation Between Transmit Operation and Internal Registe



Note: etu (Elementary time unit: the time for transfer of one bit)

Figure 14.7 Timing of TEND Flag Setting

Receiving Serial Data: Data reception in smart card mode uses the same processing pr for the normal SCI. Figure 14.8 shows a sample reception processing flowchart.

1. Perform smart card interface mode initialization as described in Initialization above
2. Check that the ORER flag and PER flag are cleared to 0 in SSR. If either is set, perf appropriate receive error handling, then clear both the ORER and the PER flag to 0.
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. To continue receiving data, clear the RDRF flag to 0 and go back to step 2.
6. To end reception, clear the RE bit to 0.

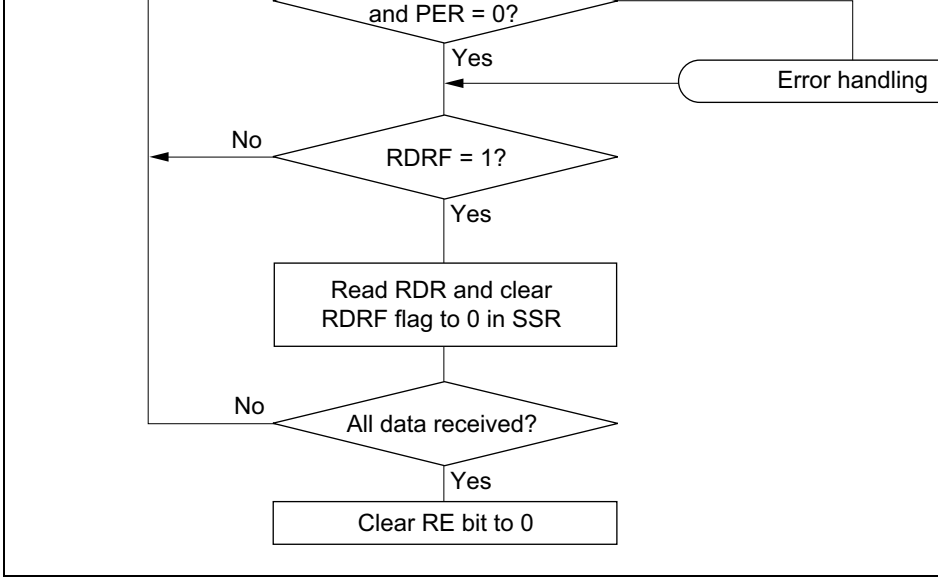


Figure 14.8 Sample Reception Processing Flowchart

The above procedure may include interrupt handling and DMA transfer.

If reception ends and the RDRF flag is set to 1 while the RIE bit is set to 1 and interru
are enabled, a receive-data-full interrupt (RXI) will be requested. If an error occurs in
and either the ORER flag or the PER flag is set to 1, a transmit/receive-error interrupt
be requested.

If the RXI interrupt activates the DMAC, the number of bytes designated in the DMA
transferred, skipping receive data in which an error occurred.

For details, see Interrupt Operations and Data Transfer by DMAC in this section.

If a parity error occurs during reception and the PER flag is set to 1, the received data
transferred to RDR, so the erroneous data can be read.

Fixing Clock Output: When the GM bit is set to 1 in SMR, clock output can be fixed t
of the CKE1 and CKE0 bits in SCR. The minimum clock pulse width can be set to the s
width in this case.

Figure 14.9 shows the timing for fixing clock output. In this example, GM = 1, CKE1 =
CKE0 bit is controlled.

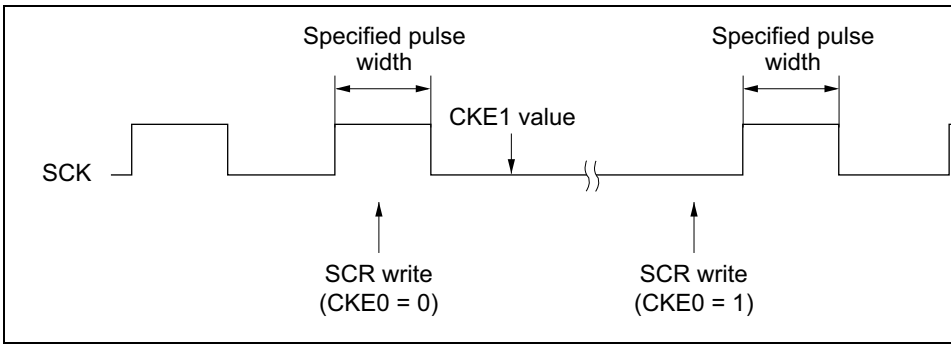


Figure 14.9 Timing for Fixing Cock Output

Interrupt Operations: The smart card interface has three interrupt sources: transmit-d
(TXI), transmit/receive-error (ERI), and receive-data-full (RXI). The transmit-end inter
request (TEI) is not available in smart card mode.

A TXI interrupt is requested when the TEND flag is set to 1 in SSR. An RXI interrupt i
when the RDRF flag is set to 1 in SSR. An ERI interrupt is requested when the ORER,
ERS flag is set to 1 in SSR. These relationships are shown in table 14.8.

	Error	PER, ORER	RIE	ERI	No

Data Transfer by DMAC: The DMAC can be used to transmit and receive data in sm
mode, as in normal SCI operations. In transmit mode, when the TEND flag is set to 1
TDRE flag is set simultaneously, generating a TXI interrupt. If the TXI request is desi
beforehand as a DMAC activation source, the DMAC will be activated by the TXI req
will transfer the next transmit data. This data transfer by the DMAC automatically cle
TDRE and TEND flags to 0. In the event of an error, the SCI automatically retransmit
data, keeping the TEND flag cleared to 0 so that the DMAC is not activated. The SCI
will therefore automatically transmit the designated number of bytes, including retrans
when an error occurs. When an error occurs, the ERS flag is not cleared automatically
bit should be set to 1 to enable the error to generate an ERI request, and the ERI interr
should clear ERS.

When using the DMAC to transmit or receive, first set up and enable the DMAC, then
settings. DMAC settings are described in section 7, DMA controller.

In receive operations, an RXI interrupt is requested when the RDRF flag is set to 1 in
RXI request is designated beforehand as a DMAC activation source, the DMAC will b
by the RXI request and will transfer the received data. This data transfer by the DMAC
automatically clears the RDRF flag to 0. When an error occurs, the RDRF flag is not s
error flag is set instead. The DMAC is not activated. The ERI interrupt request is direc
CPU. The ERI interrupt handler should clear the error flags.

software standby mode.

3. Write 0 in the CKE0 bit in SCR to stop the clock.
4. Wait for one serial clock cycle. During this period, the duty cycle is preserved and c is fixed at the specified level.
5. Write H'00 in the serial mode register (SMR) and smart card mode register (SCMR)
6. Make the transition to the software standby state.

- Returning from software standby mode to smart card interface mode

1. Clear the software standby state.
2. Set the CKE1 bit in SCR to the value for the fixed output state at the start of softwa (the current $P9_4$ pin state).
3. Set smart card interface mode and output the clock. Clock signal generation is starte normal duty cycle.

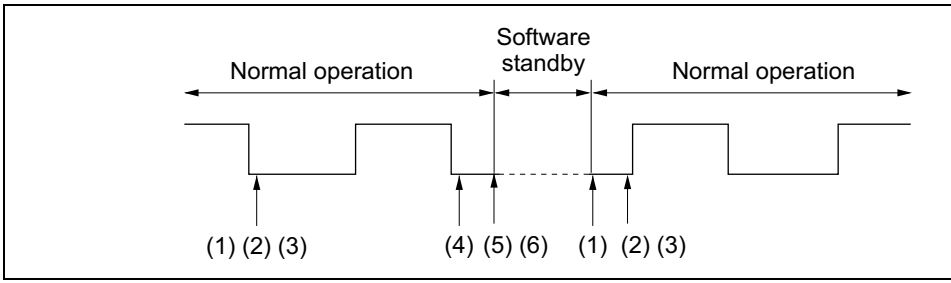


Figure 14.10 Procedure for Stopping and Restarting the Clock

Use the following procedure to secure the clock duty cycle after powering on.

1. The initial state is port input and high impedance. Use pull-up or pull-down resistor potential.
2. Fix at the output specified by the CKE1 bit in SCR.
3. Set SMR and SCMR, and switch to smart card interface mode operation.
4. Set the CKE0 bit to 1 in SCR to start clock output.

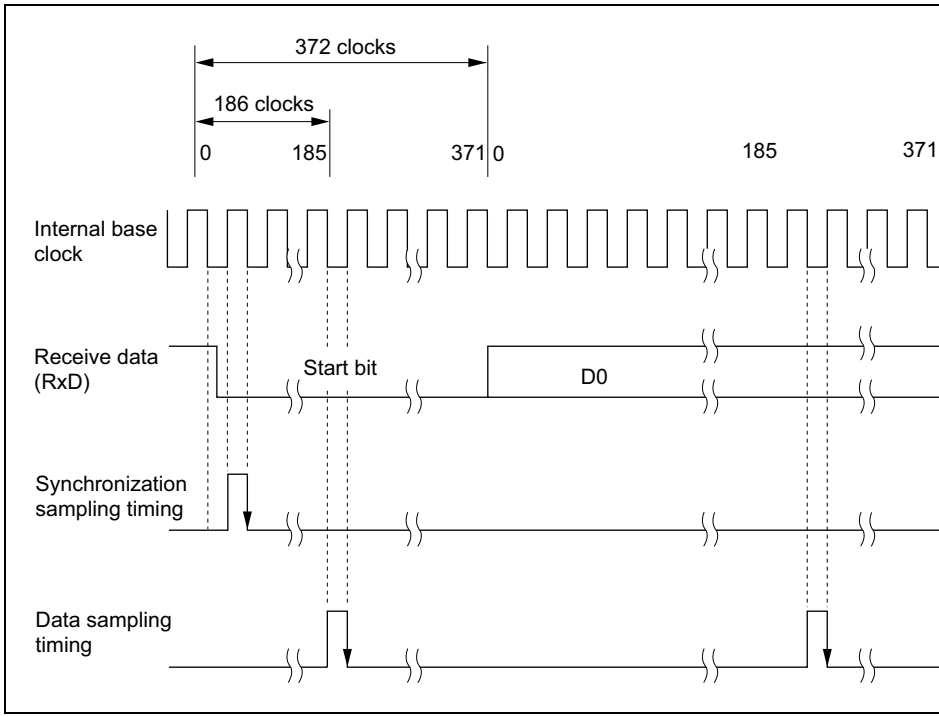


Figure 14.11 Receive Data Sampling Timing in Smart Card Interface M

L: Frame length (L =10)
F: Absolute deviation of clock frequency

From the above equation, if F = 0 and D = 0.5, the receive margin is as follows.

When D = 0.5 and F = 0:

$$M = (0.5 - 1/2 \times 372) \times 100\%$$
$$= 49.866\%$$

Retransmission: Retransmission is performed by the SCI in receive mode and transmi
described below.

- Retransmission when SCI is in Receive Mode
  Figure 14.12 illustrates retransmission when the SCI is in receive mode.

1. If an error is found when the received parity bit is checked, the PER bit is automatic
   1. If the RIE bit in SCR is set to the enable state, an ERI interrupt is requested. The
   should be cleared to 0 in SSR before the next parity bit sampling timing.
2. The RDRF bit in SSR is not set for the frame in which the error has occurred.
3. If no error is found when the received parity bit is checked, the PER bit is not set to
4. If no error is found when the received parity bit is checked, the receive operation is
   have been completed normally, and the RDRF bit is automatically set to 1 in SSR. I
   bit in SCR is set to the enable state, an RXI interrupt is requested. If RXI is enabled
   transfer activation source, the RDR contents can be read automatically. When the D
   reads the RDR data, the RDRF flag is automatically cleared to 0.
5. When a normal frame is received, the data pin is held in the high-impedance state a
   signal transmission timing.

Figure 14.12 Retransmission in SCI Receive Mode

- Retransmission when SCI is in Transmit Mode

  Figure 14.13 illustrates retransmission when the SCI is in transmit mode.

6. If an error signal is sent back from the receiving device after transmission of one f completed, the FER/ERS bit is set to 1 in SSR. If the RIE bit in SCR is set to the e an ERI interrupt is requested. The ERS bit should be cleared to 0 in SSR before th bit sampling timing.
7. The TEND bit in SSR is not set for the frame for which the error signal was receiv
8. If an error signal is not sent back from the receiving device, the ERS flag is not set
9. If an error signal is not sent back from the receiving device, transmission of one fr including retransmission, is assumed to have been completed, and the TEND bit is SSR. If the TIE bit in SCR is set to the enable state, a TXI interrupt is requested. I enabled as a DMA transfer activation source, the next data can be written in TDR automatically. When the DMAC writes data in TDR, the TDRE bit is automaticall 0.

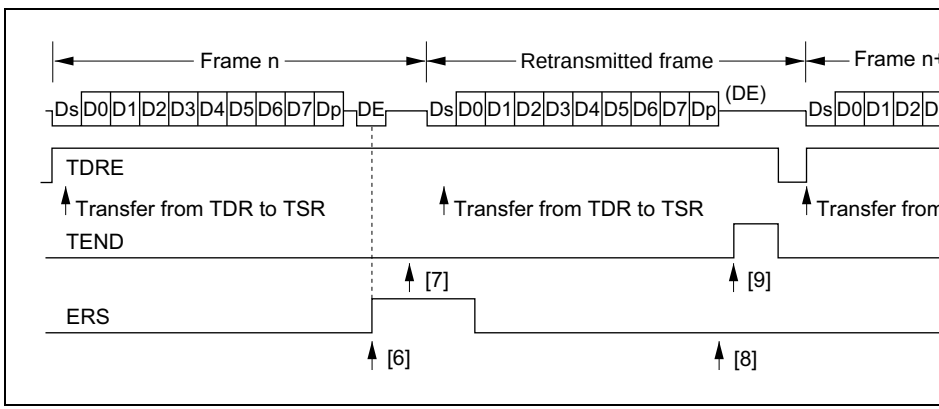


Figure 14.13 Retransmission in SCI Transmit Mode

see section 20.6, Module Standby Function.

15.1.1 Features

A/D converter features are listed below.

- 10-bit resolution
- Eight input channels
- Selectable analog conversion voltage range
  The analog voltage conversion range can be programmed by input of an analog ref
  voltage at the V_{REF} pin.
- High-speed conversion
  Conversion time: minimum 5.36 µs per channel (when operating at 25 MHz)
- Two conversion modes
  Single mode: A/D conversion of one channel
  Scan mode: continuous conversion on one to four channels
- Four 16-bit data registers
  A/D conversion results are transferred for storage into data registers corresponding
  channels.
- Sample-and-hold function
- Three conversion start sources
  The A/D converter can be activated by software, an external trigger, or an 8-bit tim
  match.
- A/D interrupt requested at end of conversion
  At the end of A/D conversion, an A/D end interrupt (ADI) can be requested.
- DMA controller (DMAC) activation
  The DMAC can be activated at the end of A/D conversion.

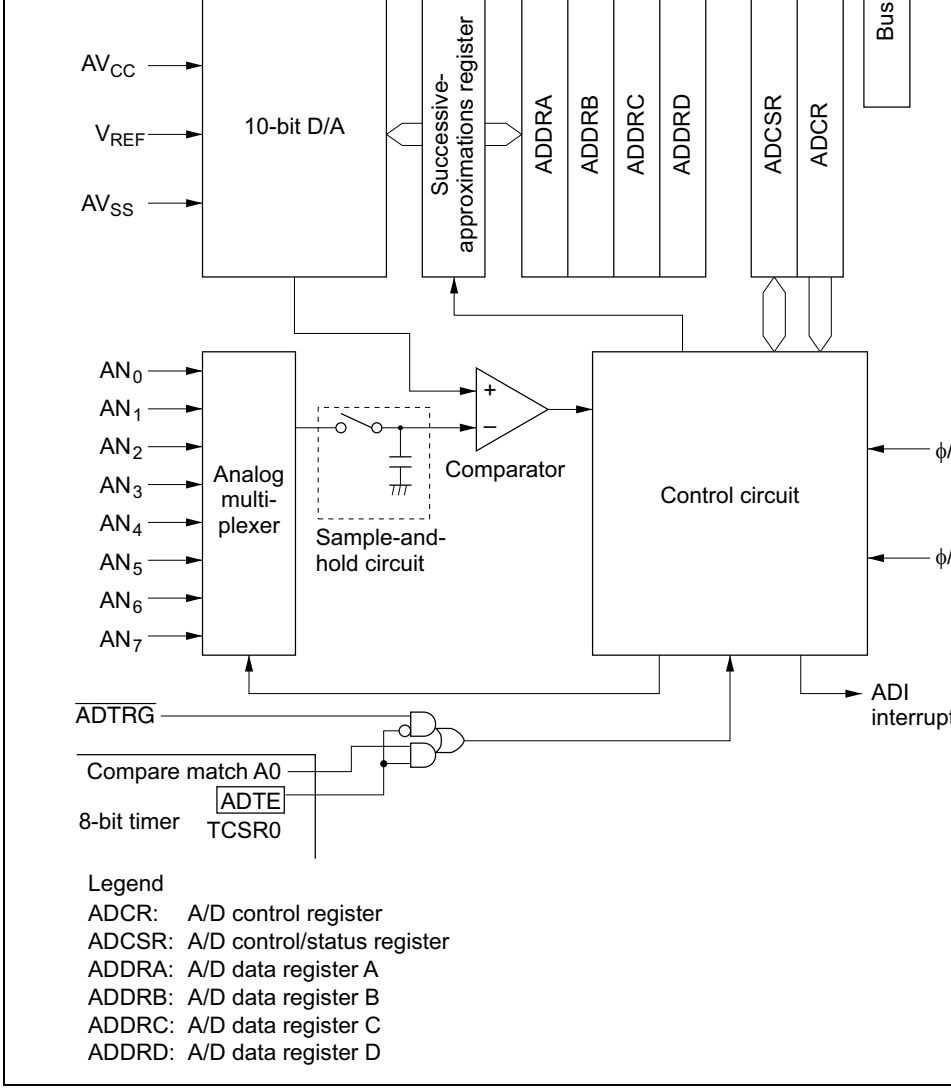


Figure 15.1 A/D Converter Block Diagram

Analog power supply pin	AV_{CC}	Input	Analog power supply
Analog ground pin	AV_{SS}	Input	Analog ground and reference v
Reference voltage pin	V_{REF}	Input	Analog reference voltage
Analog input pin 0	AN_0	Input	Group 0 analog inputs
Analog input pin 1	AN_1	Input	
Analog input pin 2	AN_2	Input	
Analog input pin 3	AN_3	Input	
Analog input pin 4	AN_4	Input	Group 1 analog inputs
Analog input pin 5	AN_5	Input	
Analog input pin 6	AN_6	Input	
Analog input pin 7	AN_7	Input	
A/D external trigger input pin	$\overline{ADTRG}$	Input	External trigger input for startin conversion

H'FFFE2	A/D data register B H	ADDRBH	R	H'00
H'FFFE3	A/D data register B L	ADDRBL	R	H'00
H'FFFE4	A/D data register C H	ADDRCH	R	H'00
H'FFFE5	A/D data register C L	ADDRCL	R	H'00
H'FFFE6	A/D data register D H	ADDRDH	R	H'00
H'FFFE7	A/D data register D L	ADDRDL	R	H'00
H'FFFE8	A/D control/status register	ADCSR	R/(W)*2	H'00
H'FFFE9	A/D control register	ADCR	R/W	H'7E

Notes: 1. Lower 20 bits of the address in advanced mode.
2. Only 0 can be written in bit 7, to clear the flag.

(n = A to D)

A/D conversion data
10-bit data giving an
A/D conversion result

Reserved b

The four A/D data registers (ADDRA to ADDRD) are 16-bit read-only registers that s
results of A/D conversion.

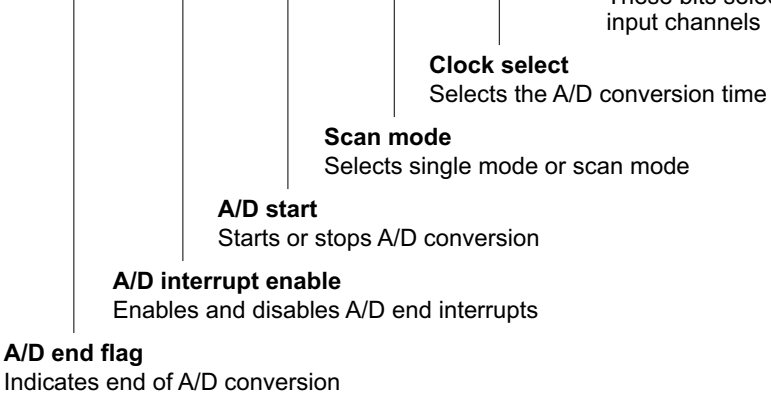
An A/D conversion produces 10-bit data, which is transferred for storage into the A/D
register corresponding to the selected channel. The upper 8 bits of the result are stored
byte of the A/D data register. The lower 2 bits are stored in the lower byte. Bits 5 to 0
data register are reserved bits that are always read as 0. Table 15.3 indicates the pairin
input channels and A/D data registers.

The CPU can always read and write the A/D data registers. The upper byte can be rea
but the lower byte is read through a temporary register (TEMP). For details see sectio
Interface.

The A/D data registers are initialized to H'0000 by a reset and in standby mode.

Table 15.3 Analog Input Channels and A/D Data Registers

Analog Input Channel		
Group 0	**Group 1**	**A/D Data Register**
AN_0	AN_4	ADDRA
AN_1	AN_5	ADDRB
AN_2	AN_6	ADDRC
AN_3	AN_7	ADDRD



Note: * Only 0 can be written, to clear the flag.

ADCSR is an 8-bit readable/writable register that selects the mode and controls the A/D
ADCSR is initialized to H'00 by a reset and in standby mode.

Bit 7—A/D End Flag (ADF): Indicates the end of A/D conversion.

Bit 7 ADF	Description	
0	[Clearing conditions] • Read ADF when ADF =1, then write 0 in ADF. • DMAC activated by ADI interrupt.	(I
1	[Setting conditions] • Single mode: A/D conversion ends • Scan mode: A/D conversion ends in all selected channels	

Bit 5—A/D Start (ADST): Starts or stops A/D conversion. The ADST bit remains se
A/D conversion. It can also be set to 1 by external trigger input at the $\overline{\text{ADTRG}}$ pin, or
timer compare match.

Bit 5 ADST	Description
0	A/D conversion is stopped
1	• Single mode: A/D conversion starts; ADST is automatically cleared to conversion ends. • Scan mode: A/D conversion starts and continues, cycling among the s channels, until ADST is cleared to 0 by software, by a reset, or by a tra standby mode.

Bit 4—Scan Mode (SCAN): Selects single mode or scan mode. For further informati
operation in these modes, see section 15.4, Operation. Clear the ADST bit to 0 before
the conversion mode.

Bit 4 SCAN	Description
0	Single mode
1	Scan mode

Bit 3—Clock Select (CKS): Selects the A/D conversion time. Clear the ADST bit to
switching the conversion time.

Bit 3 CKS	Description
0	Conversion time = 134 states (maximum)
1	Conversion time = 70 states (maximum)

		1	AN_3	AN_0 to AN_3
1	0	0	AN_4	AN_4
		1	AN_5	AN_4, AN_5
	1	0	AN_6	AN_4 to AN_6
		1	AN_7	AN_4 to AN_7

Trigger enable
Enables or disables starting of A/D conversion
by an external trigger or 8-bit timer compare match

ADCR is an 8-bit readable/writable register that enables or disables starting of A/D co
external trigger input or an 8-bit timer compare match signal. ADCR is initialized to H
reset and in standby mode.

Bit 7—Trigger Enable (TRGE): Enables or disables starting of A/D conversion by a
trigger or 8-bit timer compare match.

Bit 7 TRGE	Description
0	Starting of A/D conversion by an external trigger or 8-bit timer compare m disabled
1	A/D conversion is started at the falling edge of the external trigger signal by an 8-bit timer compare match

External trigger pin and 8-bit timer selection are performed by the 8-bit timer. For det
section 10, 8-Bit Timers.

Bits 6 to 1—Reserved: These bits cannot be modified and are always read as 1.

Bit 0—Reserved: This bit can be read or written, but must not be set to 1.

When reading an A/D data register, always read the upper byte before the lower byte. I
to read only the upper byte, but if only the lower byte is read, incorrect data may be obt

Figure 15.2 shows the data flow for access to an A/D data register.

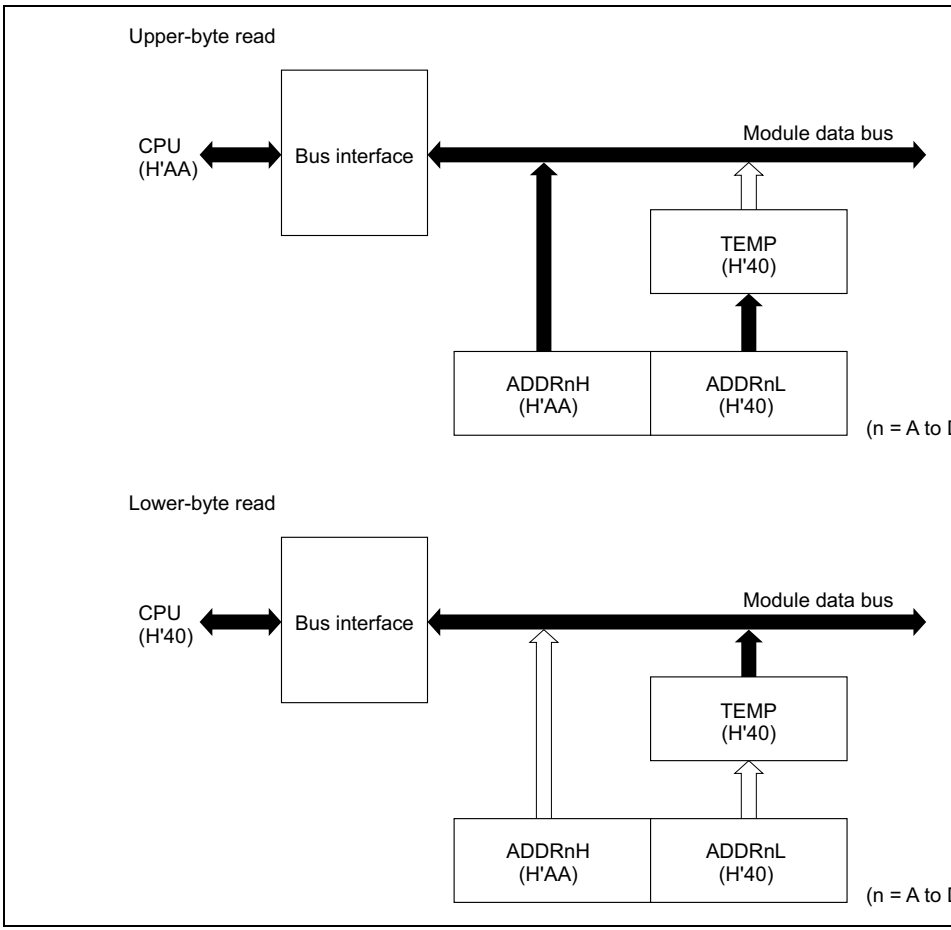


Figure 15.2 A/D Data Register Access Operation (Reading H'AA40)

ADST bit remains set to 1 during A/D conversion and is automatically cleared to 0 wh
conversion ends.

When conversion ends the ADF bit is set to 1. If the ADIE bit is also set to 1, an ADI
requested at this time. To clear the ADF flag to 0, first read ADCSR, then write 0 in A

When the mode or analog input channel must be switched during analog conversion, t
incorrect operation, first clear the ADST bit to 0 in ADCSR to halt A/D conversion. A
the necessary changes, set the ADST bit to 1 to start A/D conversion again. The ADS
set at the same time as the mode or channel is changed.

Typical operations when channel 1 (AN_1) is selected in single mode are described nex

Figure 15.3 shows a timing diagram for this example.

1. Single mode is selected (SCAN = 0), input channel AN_1 is selected (CH2 = CH1 =
CH0 = 1), the A/D interrupt is enabled (ADIE = 1), and A/D conversion is started
(ADST = 1).
2. When A/D conversion is completed, the result is transferred into ADDRB. At the s
the ADF flag is set to 1, the ADST bit is cleared to 0, and the A/D converter becor
3. Since ADF = 1 and ADIE = 1, an ADI interrupt is requested.
4. The A/D interrupt handling routine starts.
5. The routine reads ADCSR, then writes 0 in the ADF flag.
6. The routine reads and processes the conversion result (ADDRB).
7. Execution of the A/D interrupt handling routine ends. After that, if the ADST bit i
A/D conversion starts again and steps 2 to 7 are repeated.

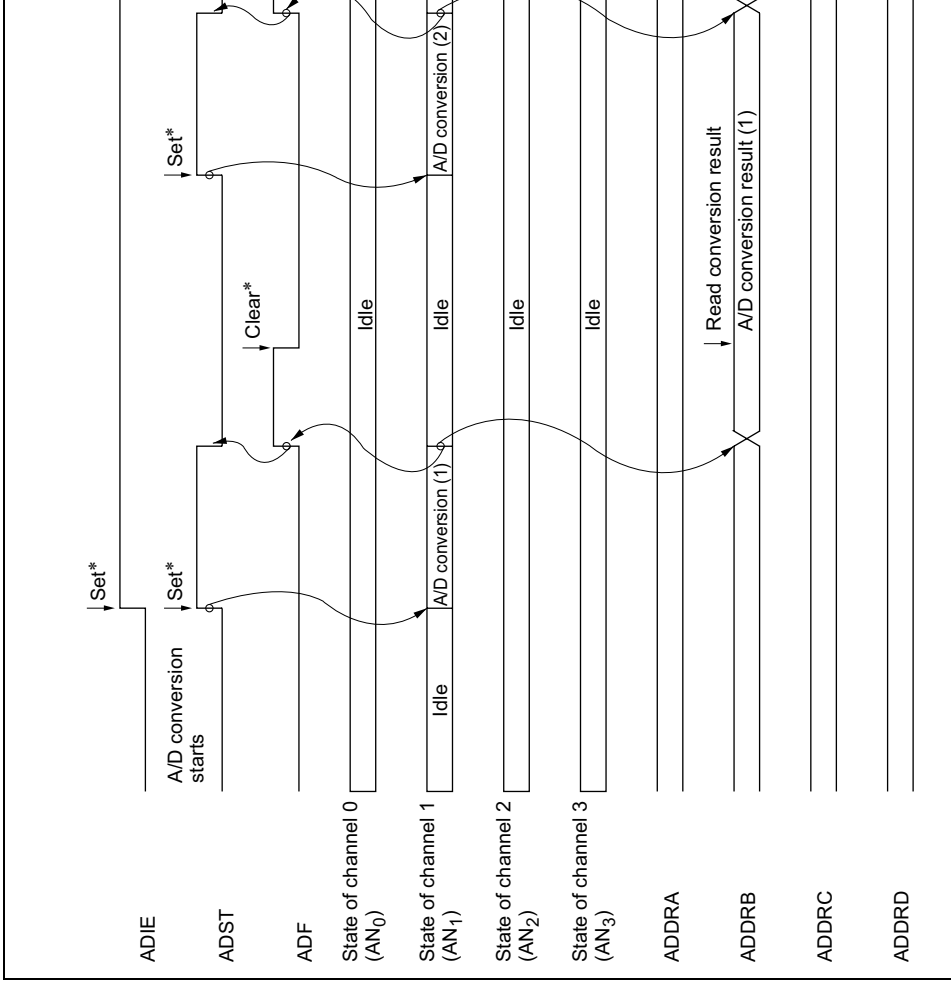


Figure 15.3 Example of A/D Converter Operation (Single Mode, Channel 1 S

When the mode or analog input channel selection must be changed during analog conv
prevent incorrect operation, first clear the ADST bit to 0 in ADCSR to halt A/D conve
making the necessary changes, set the ADST bit to 1. A/D conversion will start again
first channel in the group. The ADST bit can be set at the same time as the mode or ch
selection is changed.

Typical operations when three channels in group 0 (AN_0 to AN_2) are selected in scan m
described next. Figure 15.4 shows a timing diagram for this example.

1. Scan mode is selected (SCAN = 1), scan group 0 is selected (CH2 = 0), analog inp
   AN_0 to AN_2 are selected (CH1 = 1, CH0 = 0), and A/D conversion is started (ADS
2. When A/D conversion of the first channel (AN_0) is completed, the result is transfe
   ADDRA. Next, conversion of the second channel (AN_1) starts automatically.
3. Conversion proceeds in the same way through the third channel (AN_2).
4. When conversion of all selected channels (AN_0 to AN_2) is completed, the ADF fla
   and conversion of the first channel (AN_0) starts again. If the ADIE bit is set to 1, a
   interrupt is requested at this time.
5. Steps 2 to 4 are repeated as long as the ADST bit remains set to 1. When the ADST
   cleared to 0, A/D conversion stops. After that, if the ADST bit is set to 1, A/D con
   starts again from the first channel (AN_0).

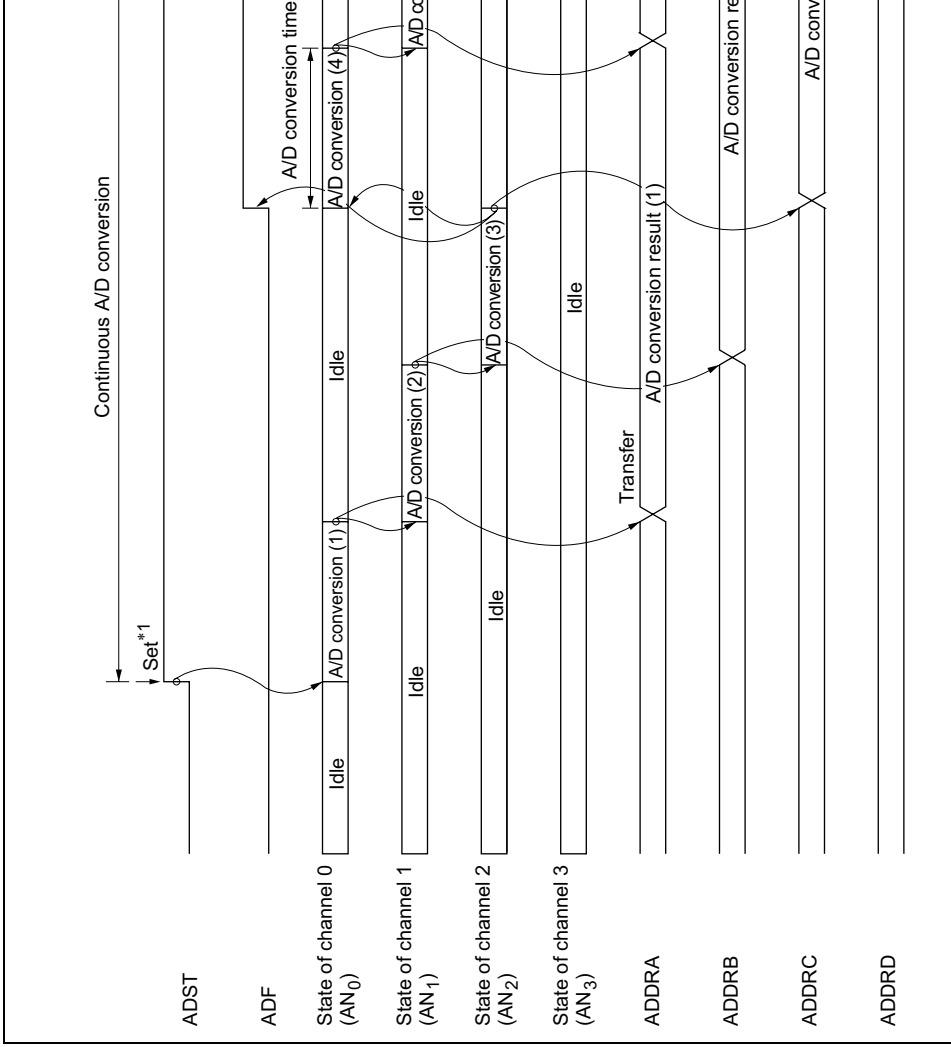
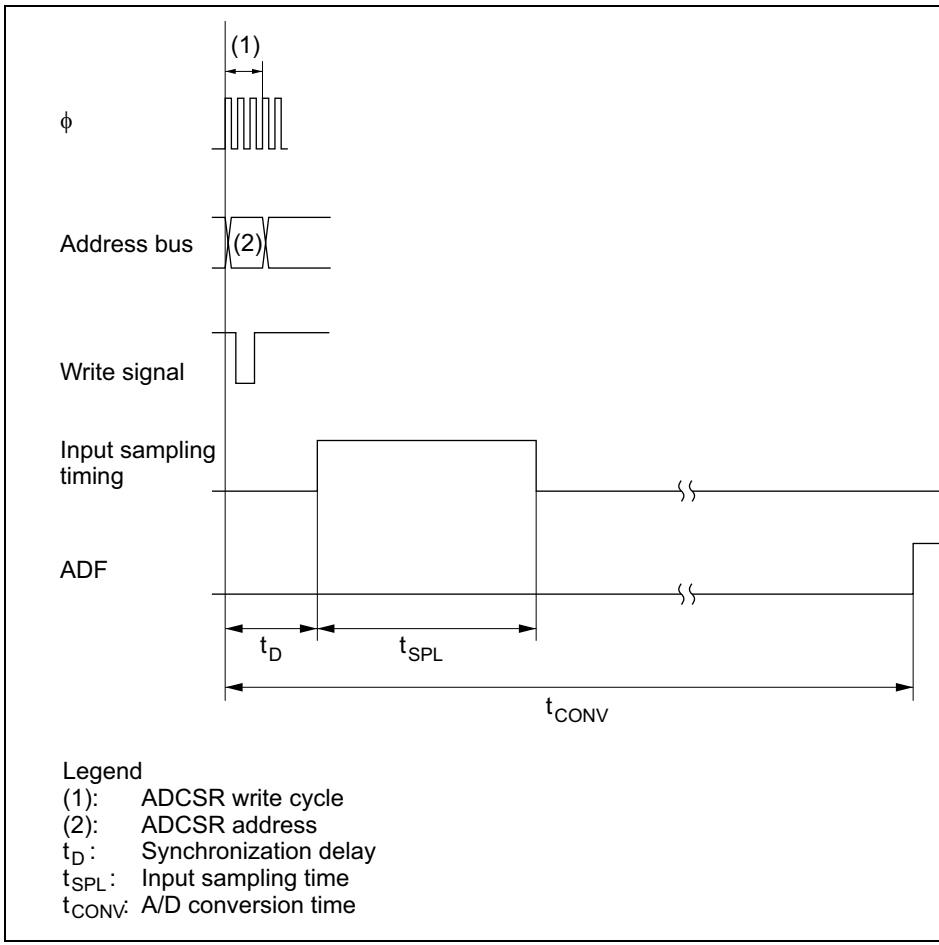


Figure 15.4 Example of A/D Converter Operation (Scan Mode, Channels AN_0 to AN_2 Selected)

In scan mode, the values given in table 15.4 apply to the first conversion. In the secon
subsequent conversions the conversion time is fixed at 128 states when CKS = 0 or 66
CKS = 1.



Legend
(1): ADCSR write cycle
(2): ADCSR address
t_D: Synchronization delay
t_{SPL}: Input sampling time
t_{CONV}: A/D conversion time

Figure 15.5 A/D Conversion Timing

15.4.4 External Trigger Input Timing

A/D conversion can be externally triggered. When the TRGE bit is set to 1 in ADCR ar
timer's ADTE bit is cleared to 0, external trigger input is enabled at the $\overline{\text{ADTRG}}$ pin. A
low transition at the $\overline{\text{ADTRG}}$ pin sets the ADST bit to 1 in ADCSR, starting A/D conve
Other operations, in both single and scan modes, are the same as if the ADST bit had b
by software. Figure 15.6 shows the timing.

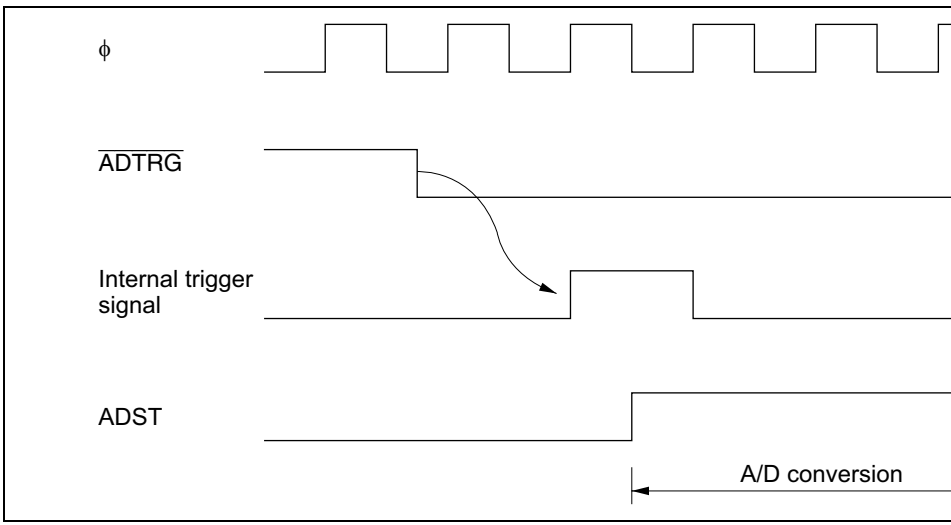


Figure 15.6 External Trigger Input Timing

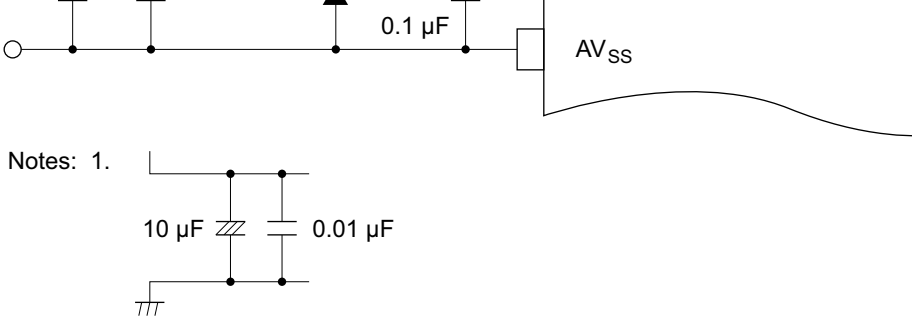
1. Analog Input Voltage Range: During A/D conversion, the voltages input to the ana pins should be in the range $AV_{SS} \leq AN_n \leq V_{REF}$.

2. Relationships of AV_{CC} and AV_{SS} to V_{CC} and V_{SS}: AV_{CC}, AV_{SS}, V_{CC}, and V_{SS} shou as follows: $AV_{SS} = V_{SS}$. AV_{CC} and AV_{SS} must not be left open, even if the A/D co used.

3. V_{REF} Programming Range: The reference voltage input at the V_{REF} pin should be i $V_{REF} \leq AV_{CC}$.

4. Note on Board Design: In board layout, separate the digital circuits from the analo much as possible. Particularly avoid layouts in which the signal lines of digital cir closely approach the signal lines of analog circuits. Induction and other effects ma analog circuits to operate incorrectly, or may adversely affect the accuracy of A/D The analog input signals (AN_0 to AN_7), analog reference voltage (V_{REF}), and analo voltage (AV_{CC}) must be separated from digital circuits by the analog ground (AV_S analog ground (AV_{SS}) should be connected to a stable digital ground (V_{SS}) at one p board.

5. Note on Noise: To prevent damage from surges and other abnormal voltages at the input pins (AN_0 to AN_7) and analog reference voltage pin (V_{REF}), connect a protec like the one in figure 15.7 between AV_{CC} and AV_{SS}. The bypass capacitors connec and V_{REF} and the filter capacitors connected to AN_0 to AN_7 must be connected to A filter capacitors like the ones in figure 15.7 are connected, the voltage values input analog input pins (AN_0 to AN_7) will be smoothed, which may give rise to error. Er occur if A/D conversion is frequently performed in scan mode so that the current t and discharges the capacitor in the sample-and-hold circuit of the A/D converter b greater than that input to the analog input pins via input impedance Rin. The circui should therefore be selected carefully.



Notes: 1.

2. Rin: input impedance

Figure 15.7 Example of Analog Input Protection Circuit

Table 15.5 Analog Input Pin Ratings

Item	Min	Max	Unit
Analog input capacitance	—	20	pF
Allowable signal-source impedance	—	10*	kΩ

Note: * When conversion time = 134 states, V_{CC} = 3.0 V to 3.6 V, and $\phi \leq 13$ MHz. For c
section 21, Electrical Characteristics.

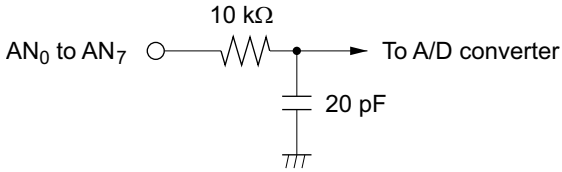


Figure 15.8 Analog Input Pin Equivalent Circuit

Note: Numeric values are approximate, except in table 15.5

voltage required to raise digital output from 1111111110 t
1111111111 (figure 15.10)

- Quantization error: ..... Intrinsic error of the A/D converter; 1/2 LSB (figure 15.9)
- Nonlinearity error: ..... Deviation from ideal A/D conversion characteristic in rang
volts to full scale, exclusive of offset error, full-scale error
quantization error.
- Absolute accuracy: ..... Deviation of digital value from analog input value, includi
error, full-scale error, quantization error, and nonlinearity

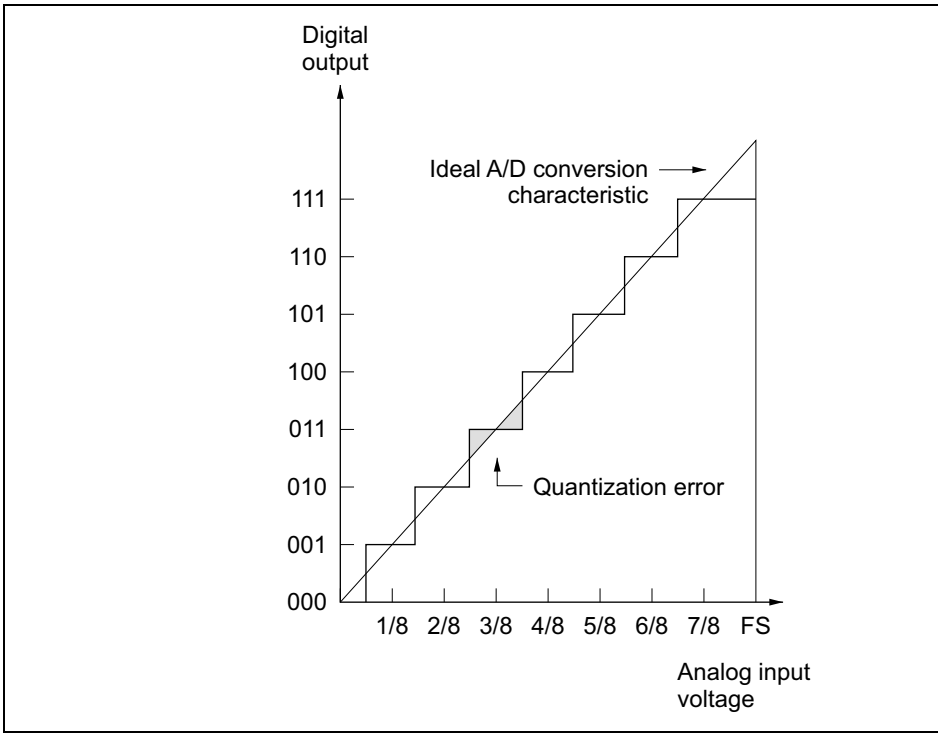


Figure 15.9 A/D Converter Accuracy Definitions (1)

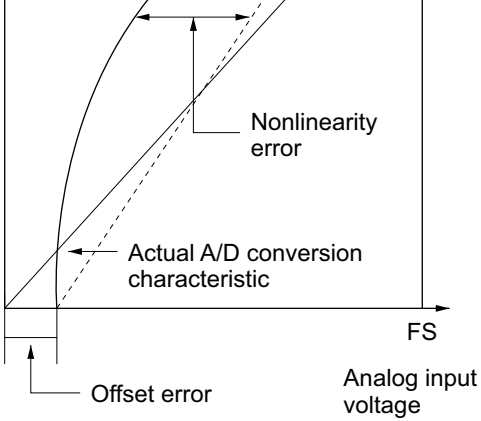


Figure 15.10 A/D Converter Accuracy Definitions (2)

7. Allowable Signal-Source Impedance: The analog inputs of the H8/3028 Group are d assure accurate conversion of input signals with a signal-source impedance not exce kΩ. The reason for this rating is that it enables the input capacitor in the sample-and circuit in the A/D converter to charge within the sampling time. If the sensor output exceeds 10 kΩ, charging may be inadequate and the accuracy of A/D conversion ca guaranteed.

   If a large external capacitor is provided in single mode, then the internal 10-kΩ inpu resistance becomes the only significant load on the input. In this case the impedance signal source is not a problem.

   A large external capacitor, however, acts as a low-pass filter. This may make it imp track analog signals with high dv/dt (e.g. a variation of 5 mV/µs) (figure 15.11). To high-speed analog signals or to use scan mode, insert a low-impedance buffer.

8. Effect on Absolute Accuracy: Attaching an external capacitor creates a coupling wi so if there is noise on the ground line, it may degrade absolute accuracy. The capaci connected to an electrically stable ground, such as AV_{SS}.

   If a filter circuit is used, be careful of interference with digital signals on the same b make sure the circuit does not act as an antenna.

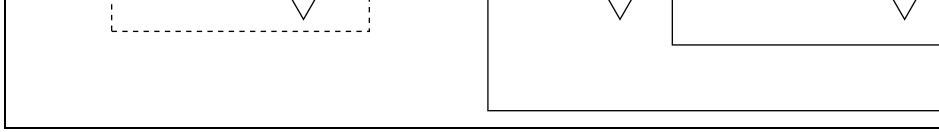


Figure 15.11 Analog Input Circuit (Example)

D/A converter features are listed below.

- Eight-bit resolution
- Two output channels
- Conversion time: maximum 10 μs (with 20-pF capacitive load)
- Output voltage: 0 V to $\frac{255}{256} \times V_{REF}$
- D/A outputs can be sustained in software standby mode

16.1.2 Block Diagram

Figure 16.1 shows a block diagram of the D/A converter.

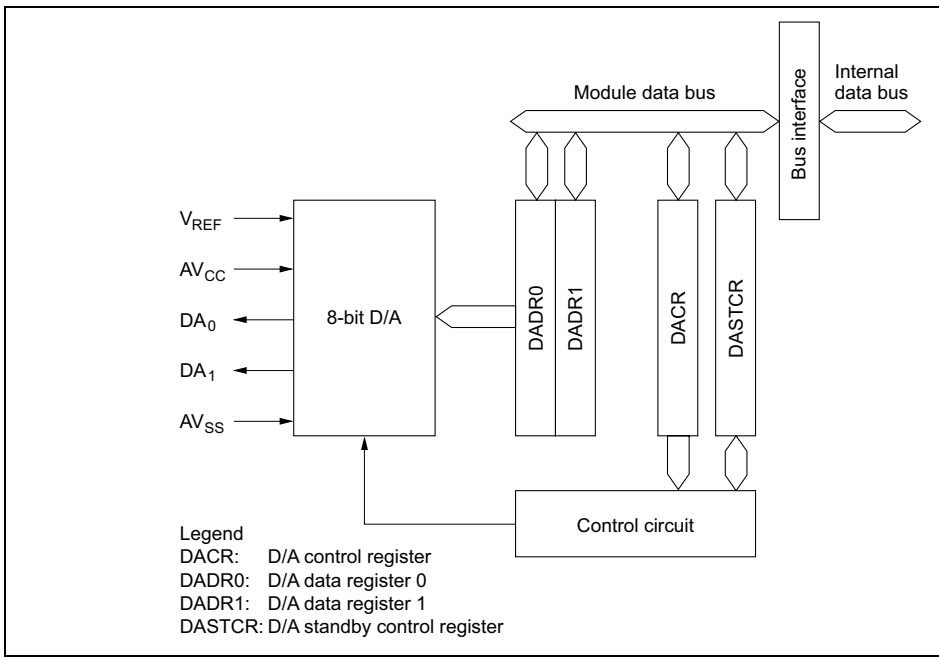


Figure 16.1 D/A Converter Block Diagram

Analog output pin 0	DA_0	Output	Analog output, channel 0
Analog output pin 1	DA_1	Output	Analog output, channel 1
Reference voltage input pin	V_{REF}	Input	Analog reference voltage

16.1.4 Register Configuration

Table 16.2 summarizes the D/A converter's registers.

Table 16.2 D/A Converter Registers

Address*	Name	Abbreviation	R/W	Initia
H'FFF9C	D/A data register 0	DADR0	R/W	H'00
H'FFF9D	D/A data register 1	DADR1	R/W	H'00
H'FFF9E	D/A control register	DACR	R/W	H'1F
H'EE01A	D/A standby control register	DASTCR	R/W	H'FE

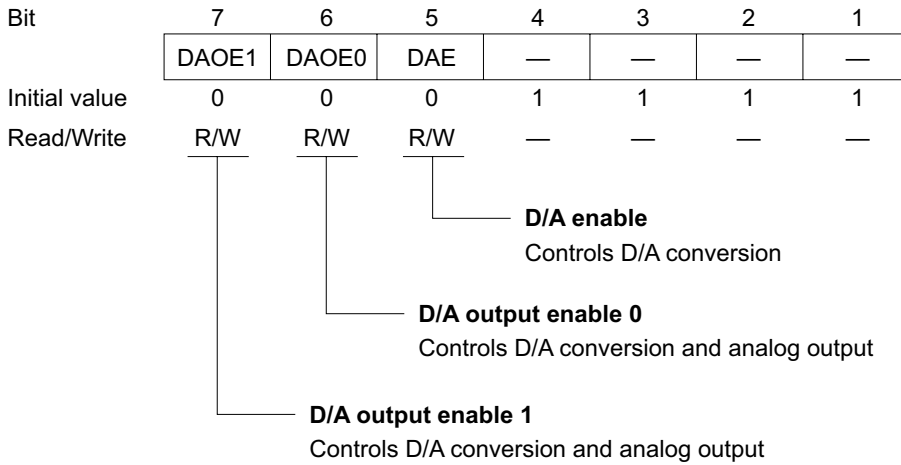
Note: * Lower 20 bits of the address in advanced mode.

The D/A data registers (DADR0 and DADR1) are 8-bit readable/writable registers tha data to be converted. When analog output is enabled, the D/A data register values are converted and output at the analog output pins.

The D/A data registers are initialized to H'00 by a reset and in standby mode.

When the DASTE bit is set to 1 in the D/A standby control register (DASTCR), the D are not initialized in software standby mode.

16.2.2 D/A Control Register (DACR)



DACR is an 8-bit readable/writable register that controls the operation of the D/A con DACR is initialized to H'1F by a reset and in standby mode.

When the DASTE bit is set to 1 in DASTCR, the DACR is not initialized in software mode.

Bit 6 DAOE0	Description
0	DA_0 analog output is disabled
1	Channel-0 D/A conversion and DA_0 analog output are enabled

Bit 5—D/A Enable (DAE): Controls D/A conversion, together with bits DAOE0 and D
When the DAE bit is cleared to 0, analog conversion is controlled independently in cha
and 1. When the DAE bit is set to 1, analog conversion is controlled together in channe
Output of the conversion results is always controlled independently by DAOE0 and DA

Bit 7 DAOE1	Bit 6 DAOE0	Bit 5 DAE	Description
0	0	—	D/A conversion is disabled in channels 0 and 1
	1	0	D/A conversion is enabled in channel 0
			D/A conversion is disabled in channel 1
		1	D/A conversion is enabled in channels 0 and 1
1	0	0	D/A conversion is disabled in channel 0
			D/A conversion is enabled in channel 1
		1	D/A conversion is enabled in channels 0 and 1
	1	—	D/A conversion is enabled in channels 0 and 1

When the DAE bit is set to 1, even if bits DAOE0 and DAOE1 in DACR and the ADS
ADCSR are cleared to 0, the same current is drawn from the analog power supply as du
and D/A conversion.

Bits 4 to 0—Reserved: These bits cannot be modified and are always read as 1.

Read/Write

Reserved bits

D/A standby enable
Enables or disables
in software standby

DASTCR is initialized to H'FE by a reset and in hardware standby mode. It is not initi
software standby mode.

Bits 7 to 1—Reserved: These bits cannot be modified and are always read as 1.

Bit 0—D/A Standby Enable (DASTE): Enables or disables D/A output in software s
mode.

Bit 0 DASTE	Description
0	D/A output is disabled in software standby mode
1	D/A output is enabled in software standby mode

An example of D/A conversion on channel 0 is given next. Timing is indicated in figur

1. Data to be converted is written in DADR0.
2. Bit DAOE0 is set to 1 in DACR. D/A conversion starts and DA0 becomes an outpu converted result is output after the conversion time.

   The output value is $\frac{\text{DADR contents}}{256} \times V_{REF}$

   Output of this conversion result continues until the value in DADR0 is modified or DAOE0 bit is cleared to 0.
3. If the DADR0 value is modified, conversion starts immediately, and the result is ou the conversion time.
4. When the DAOE0 bit is cleared to 0, DA0 becomes an input pin.

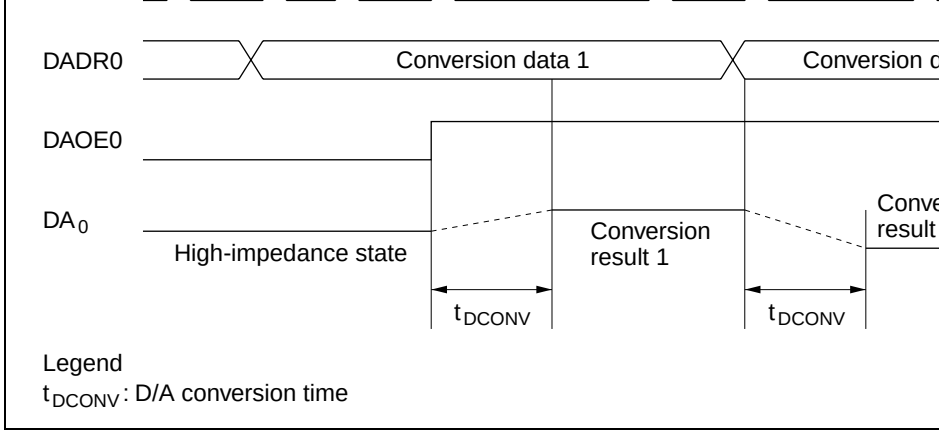


Figure 16.2 Example of D/A Converter Operation

16.4 D/A Output Control

In the H8/3028 Group, D/A converter output can be enabled or disabled in software st

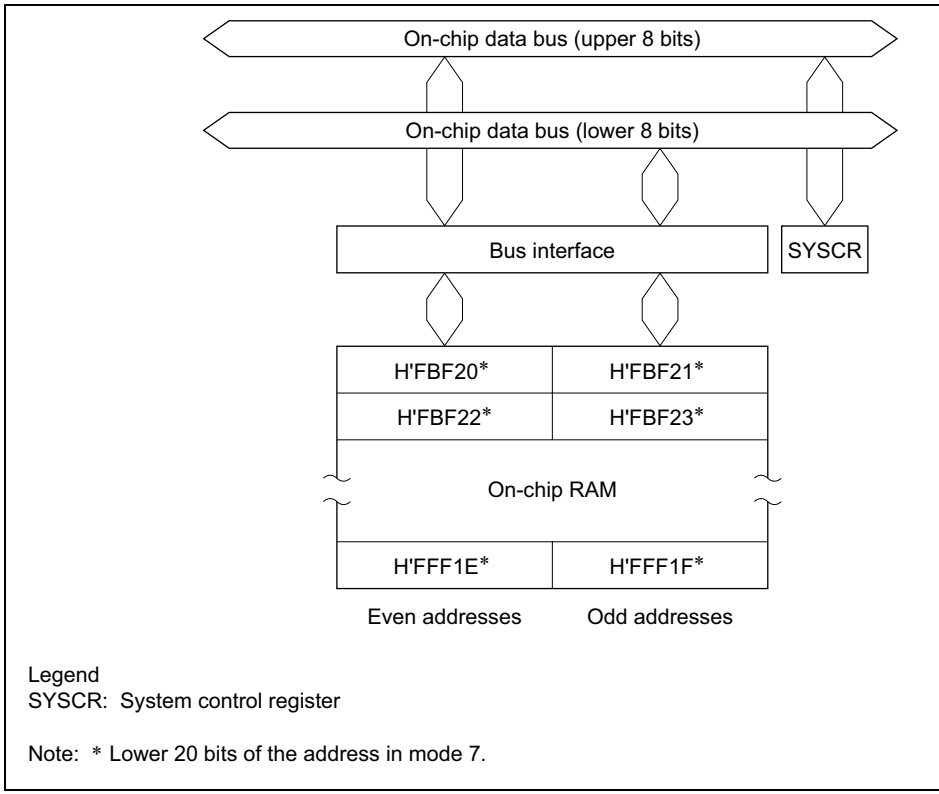
When the DASTE bit is set to 1 in DASTCR, D/A converter output is enabled in softw mode. The D/A converter registers retain the values they held prior to the transition to standby mode.

When D/A output is enabled in software standby mode, the reference supply current is during normal operation.

The on-chip RAM of the H8/3028 Group is assigned to addresses H'FBF20 to H'FFF1
1, 2, and 7, and to addresses H'FFBF20 to H'FFFF1F in modes 3, 4, and 5,and to addr
to H'FF1F in mode 6. The RAM enable bit (RAME) in the system control register (SY
enable or disable the on-chip RAM.

17.1.1 Block Diagram

Figure 17.1 shows a block diagram of the on-chip RAM.



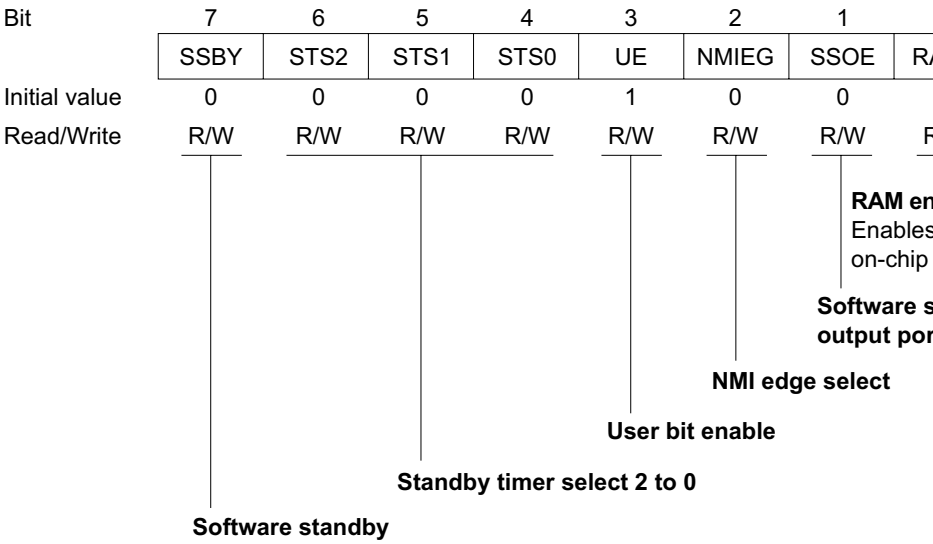
Legend
SYSCR: System control register

Note: * Lower 20 bits of the address in mode 7.

Figure 17.1 RAM Block Diagram

Note: * Lower 20 bits of the address in advanced mode.

17.2 System Control Register (SYSCR)



One function of SYSCR is to enable or disable access to the on-chip RAM. The on-chip
enabled or disabled by the RAME bit in SYSCR. For details about the other bits, see se
System Control Register (SYSCR).

Bit 0—RAM Enable (RAME): Enables or disables the on-chip RAM. The RAME bit
initialized at the rising edge of the input at the $\overline{\text{RES}}$ pin. It is not initialized in software
mode.

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

Since the on-chip RAM is connected to the CPU by an internal 16-bit data bus, it can
and read by word access. It can also be written and read by byte access. Byte data is a
two states using the upper 8 bits of the data bus. Word data starting at an even address
in two states using all 16 bits of the data bus.

enabling rapid data transfer.

The on-chip ROM is enabled and disabled by setting the mode pins (MD_2 to MD_0) as table 18.1.

The on-chip flash memory product (H8/3028F-ZTAT) can be erased and programmed as well as with a special-purpose PROM programmer.

Table 18.1 Operating Modes and ROM

	Mode Pins			
Mode	**MD2**	**MD1**	**MD0**	**On-Cl**
Mode 1 (expanded 1-Mbyte mode with on-chip ROM disabled)	0	0	1	Disab addre
Mode 2 (expanded 1-Mbyte mode with on-chip ROM disabled)	0	1	0	
Mode 3 (expanded 16-Mbyte mode with on-chip ROM disabled)	0	1	1	
Mode 4 (expanded 16-Mbyte mode with on-chip ROM disabled)	1	0	0	
Mode 5 (expanded 16-Mbyte mode with on-chip ROM enabled)	1	0	1	Enabl
Mode 6 (single-chip normal mode)	1	1	0	
Mode 7 (single-chip advanced mode)	1	1	1	

— Program-verify mode

— Erase-verify mode

- Programming/erase methods

  The flash memory is programmed 128 bytes at a time. Erasing is performed in bloc
  erase the entire flash memory, each block must be erased in turn. In block erasing, 4
  kbyte, and 64-kbyte blocks can be set arbitrarily.

- Programming/erase times

  The flash memory programming time is 10 ms (typ.) for simultaneous 128-byte pro
  equivalent approximately to 80 µs (typ.) per byte, and the erase time is 100 ms (typ.

- Reprogramming capability

  The flash memory can be reprogrammed up to 100 times.

- On-board programming modes

  There are two modes in which flash memory can be programmed/erased/verified on

  — Boot mode

  — User program mode

- Automatic bit rate adjustment

  For data transfer in boot mode, the H8/3028F-ZTAT chip's bit rate can be automati
  adjusted to match the transfer bit rate of the host.

- Flash memory emulation in RAM

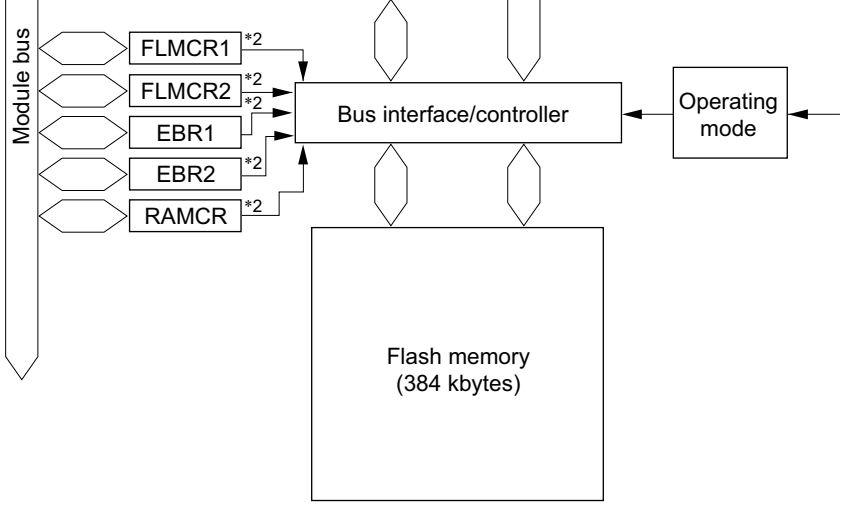
  Flash memory programming can be emulated in real time by overlapping a part of F
  flash memory.

- Protect modes

  There are three protect modes—hardware, software, and error—which allow protec
  to be designated for flash memory program/erase/verify operations

- PROM mode

  Flash memory can be programmed/erased in PROM mode, using a PROM program
  well as in on-board programming mode.



Legend
FLMCR1: Flash memory control register 1
FLMCR2: Flash memory control register 2
EBR1: Erase block register 1
EBR2: Erase block register 2
RAMCR: RAM control register

Notes: 1. Functions as FWE in flash memory version and as $\overline{\text{RESO}}$ in mask ROM v
2. The flash memory control registers (FLMCR1, FLMCR2, EBR1, EBR2, R/ are used only by the flash memory version and do not exist in the mask R version. In the mask ROM version reading these addresses always return value of 1, and it is not possible to write to them.

Figure 18.1 Block Diagram of Flash Memory

Mode 2	MD_2	Input	Sets H8/3028F-ZTAT operating m
Mode 1	MD_1	Input	Sets H8/3028F-ZTAT operating m
Mode 0	MD_0	Input	Sets H8/3028F-ZTAT operating m
Transmit data	TxD_1	Output	Serial transmit data output
Receive data	RxD_1	Input	Serial receive data input

18.2.3 Register Configuration

The registers used to control the on-chip flash memory when enabled are shown in tabl

Table 18.3 Flash Memory Registers

Register Name	Abbreviation	R/W	Initial Value	Ad
Flash memory control register 1	FLMCR1	R/W	H'00*2	H'E
Flash memory control register 2	FLMCR2	R	H'00	H'E
Erase block register 1	EBR1	R/W	H'00	H'E
Erase block register 2	EBR2	R/W	H'00	H'E
RAM control register	RAMCR	R/W	H'F0	H'E

Notes: FLMCR1, FLMCR2, EBR1, EBR2, and RAMCR are 8-bit registers, and should b accessed by byte access.

1. Lower 20 bits of address in advanced mode.
2. When a high level is input to the FWE pin, the initial value is H'80.

Note: * Determined by the state of the FWE pin.

FLMCR1 is an 8-bit register used for flash memory operating mode control.

Program-verify mode or erase-verify mode for addresses H'00000 to H'5FFFF is enter the SWE bit when FWE = 1, then setting the PV or EV bit. Program mode for address to H'5FFFF is entered by setting the SWE bit when FWE = 1, then setting the PSU bit setting the P bit. Erase mode for addresses H'00000 to H'5FFFF is entered by setting t when FWE = 1, then setting the ESU bit, and finally setting the E bit. FLMCR1 is init reset, and in hardware standby mode and software standby mode. Its initial value is H' high level is input to the FWE pin, and H'00 when a low level is input. In mode 6 the must be fixed low since flash memory on-board programming modes are not supporte on-chip flash memory is disabled, a read access to this register will return H'00, and w invalid.

When setting bits 6 to 0 in this register, one bit must be set one at a time. Writes to the FLMCR1 are enabled only when FWE = 1; writes to bits ESU, PSU, EV, and PV only = 1 and SWE = 1; writes to the E bit only when FWE = 1, SWE = 1, and ESU = 1; and the P bit only when FWE = 1, SWE = 1, and PSU = 1.

Notes: 1. The programming and erase flowcharts must be followed when setting the register to prevent erroneous programming or erasing.
2. Transitions are made to program mode, erase mode, program-verify mode, verify mode according to the settings in this register. When reading flash m normal on-chip ROM, bits 6 to 0 in this register must be cleared.

Bit 7—Flash Write Enable (FWE): Sets hardware protection against flash memory programming/erasing.

Bit 7 FWE	Description
0	When a low level is input to the FWE pin (hardware-protected state)
1	When a high level is input to the FWE pin

Note: Do not execute a SLEEP instruction while the SWE bit is set to 1.

Bit 5—Erase Setup (ESU): Prepares for a transition to erase mode. Set this bit to 1 be
the E bit to 1 in FLMCR1 (do not set the SWE, PSU, EV, PV, E, or P bit at the same ti

Bit 5 ESU	Description	
0	Erase setup cleared	(In
1	Erase setup [Setting condition] When FWE = 1 and SWE = 1	

Bit 4—Program Setup (PSU): Prepares for a transition to program mode. Set this bit t
setting the P bit to 1 in FLMCR1 (do not set the SWE, ESU, EV, PV, E, or P bit at the s

Bit 4 PSU	Description	
0	Program setup cleared	(In
1	Program setup [Setting condition] When FWE = 1 and SWE = 1	

Bit 3—Erase-Verify Mode (EV): Selects erase-verify mode transition or clearing. (Do
SWE, ESU, PSU, PV, E, or P bit at the same time.)

Bit 3 EV	Description	
0	Erase-verify mode cleared	(In
1	Transition to erase-verify mode [Setting condition] When FWE = 1 and SWE = 1	

Bit 1—Erase Mode (E): Selects erase mode transition or clearing. (Do not set the SW
PSU, EV, PV, or P bit at the same time.)

Bit 1 E	Description	
0	Erase mode cleared	(
1	Transition to erase mode [Setting condition] When FWE = 1, SWE = 1, and ESU = 1	

Note: Do not access the flash memory while the E bit is set.

Bit 0—Program (P): Selects program mode transition or clearing. (Do not set the SW
PSU, EV, PV, or E bit at the same time.)

Bit 0 P	Description	
0	Program mode cleared	(
1	Transition to program mode [Setting condition] When FWE = 1, SWE = 1, and PSU = 1	

Note: Do not access the flash memory while the P bit is set.

initialized to H'00 by a reset, and in hardware standby mode and software standby mod
the on-chip flash memory is disabled, a read will return H'00.

Note: FLMCR2 is a read-only register, and should not be written to.

Bit 7—Flash Memory Error (FLER): Indicates that an error has occurred during an o
flash memory (programming or erasing). When FLER is set to 1, flash memory goes to
protection state.

Bit 7 FLER	Description
0	Flash memory is operating normally Flash memory program/erase protection (error protection) is disabled [Clearing condition] Reset ($\overline{\text{RES}}$ pin or WDT reset) or hardware standby mode (In
1	An error occurred during flash memory programming/erasing Flash memory program/erase protection (error protection) is enabled [Setting conditions] • When flash memory is read during programming/erasing (including a ve or instruction fetch, but excluding a read of the RAM area overlapping fl memory space) • Immediately after the start of exception handling during programming/er (excluding reset, illegal instruction, trap instruction, and division-by-zero handling) • When a SLEEP instruction (including software standby) is executed dur programming/erasing • When the bus is released during programming/erasing

Bits 6 to 0—Reserved: These bits are always read as 0.

initialized to H'00 by a reset, in hardware standby mode and software standby mode, w
level is input to the FWE pin, and when a high level is input to the FWE pin and the S
FLMCR1 is not set. When a bit in EBR1 is set to 1, the corresponding block can be er
blocks are erase-protected. Only one bit can be set in EBR1 and EBR2 together; do no
more bits at the same time. When the on-chip flash memory is disabled, a read access
register will return H'00, and erasing is disabled.

The flash memory block configuration is shown in table 18.4. To erase the entire flash
each block must be erased in turn.

As the H8/3028F-ZTAT does not support on-board programming modes in mode 6, E
bits cannot be set to 1 in this mode.

18.3.4 Erase Block Register 2 (EBR2)

Bit	7	6	5	4	3	2	1
	—	—	EB13	EB12	EB11	EB10	EB9
Initial value	0	0	0	0	0	0	0
Read/Write	R	R	R/W	R/W	R/W	R/W	R/W

EBR2 is an 8-bit register that specifies the flash memory erase area block by block. EI
initialized to H'00 by a reset, in hardware standby mode and software standby mode, a
low level is input to the FWE pin. When a high level is input to the FWE pin and the S
FLMCR1 is not set, it is initialized to bit 0. When a bit in EBR2 is set to 1, the corresp
block can be erased. Other blocks are erase-protected. Only one bit can be set in EBR
together; do not set two or more bits at the same time. When the on-chip flash memor
a read will return H'00, and erasing is disabled.

The flash memory block configuration is shown in table 18.4. To erase the entire flash
each block must be erased in turn.

As the H8/3028F-ZTAT does not support on-board programming modes in mode 6, E
bits cannot be set to 1 in this mode.

EB3 (4 kbytes)	H'003000 to H'003FFF
EB4 (4 kbytes)	H'004000 to H'004FFF
EB5 (4 kbytes)	H'005000 to H'005FFF
EB6 (4 kbytes)	H'006000 to H'006FFF
EB7 (4 kbytes)	H'007000 to H'007FFF
EB8 (32 kbytes)	H'008000 to H'00FFFF
EB9 (64 kbytes)	H'010000 to H'01FFFF
EB10 (64 kbytes)	H'020000 to H'02FFFF
EB11 (64 kbytes)	H'030000 to H'03FFFF
EB12 (64 kbytes)	H'040000 to H'04FFFF
EB13 (64 kbytes)	H'050000 to H'05FFFF

18.3.5 RAM Control Register (RAMCR)

Bit	7	6	5	4	3	2	1
	—	—	—	—	RAMS	RAM2	RAM1
Initial value	1	1	1	1	0	0	0
Read/Write	R	R	R	R	R/W	R/W	R/W

RAMCR specifies the area of flash memory to be overlapped with part of RAM when e realtime flash memory programming. RAMCR is initialized to H'00 by a reset and in ha standby mode. RAMCR settings should be made in user mode or user program mode.

Flash memory area divisions are shown in table 18.5. To ensure correct operation of the function, the ROM for which RAM emulation is performed should not be accessed imm after this register has been modified. Normal execution of an access immediately after m modification is not guaranteed.

Bits 7 to 4—Reserved: These bits cannot be modified and are always read as 1.

Bits 2 to 0—Flash Memory Area Selection (RAM2 to RAM0): These bits are used
with bit 3 to select the flash memory area to be overlapped with RAM. (See table 18.5

Table 18.5 Flash Memory Area Divisions

RAM Area	Block Name	RAMS	RAM2	RAM1
H'FFE000 to H'FFEFFF	4-kbyte RAM area	0	*	*
H'000000 to H'000FFF	EB0 (4 kbytes)	1	0	0
H'001000 to H'001FFF	EB1 (4 kbytes)	1	0	0
H'002000 to H'002FFF	EB2 (4 kbytes)	1	0	1
H'003000 to H'003FFF	EB3 (4 kbytes)	1	0	1
H'004000 to H'004FFF	EB4 (4 kbytes)	1	1	0
H'005000 to H'005FFF	EB5 (4 kbytes)	1	1	0
H'006000 to H'006FFF	EB6 (4 kbytes)	1	1	1
H'007000 to H'007FFF	EB7 (4 kbytes)	1	1	1

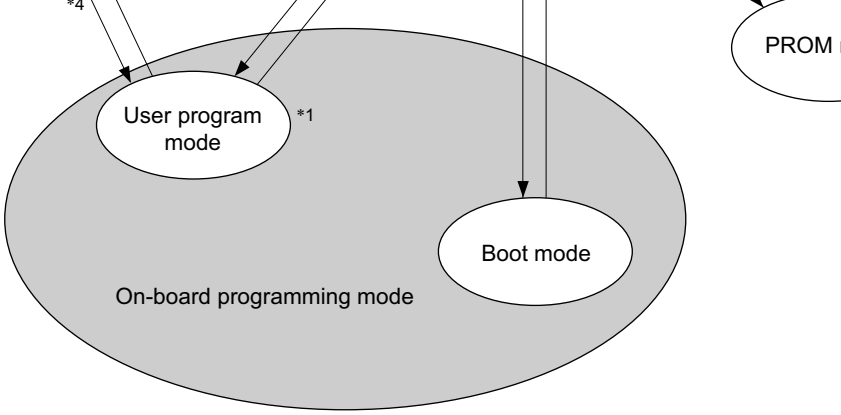
*: Don’t care

Note: Flash memory emulation by RAM is not supported in mode 6 (single-chip norm
therefore, although these bits can be written, they should not be set to 1.

When performing flash memory emulation by RAM, the RAME bit in SYSCR m
1.

mode.

Boot mode and user program mode cannot be used in the H8/3028F-ZTAT's mode 6 (r
mode with on-chip ROM enabled).



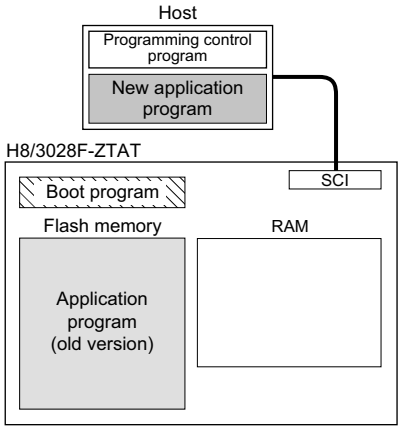
Notes: Only make a transition between user mode and user program mode when the CPU accessing the flash memory.

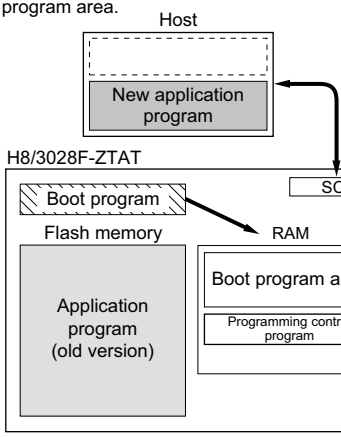
1. RAM emulation possible
2. The H8/3028F-ZTAT is placed in PROM mode by means of a dedicated PROM
3. MD_2, MD_1, MD_0 = (1, 0, 1) (1, 1, 0) (1, 1, 1)
   FWE = 0
4. MD_2, MD_1, MD_0 = (1, 0, 1) (1, 1, 1)
   FWE = 1
5. MD_2, MD_1, MD_0 (0, 0, 1) (0, 1, 1)
   FWE = 1

Figure 18.2 Flash Memory Related State Transitions

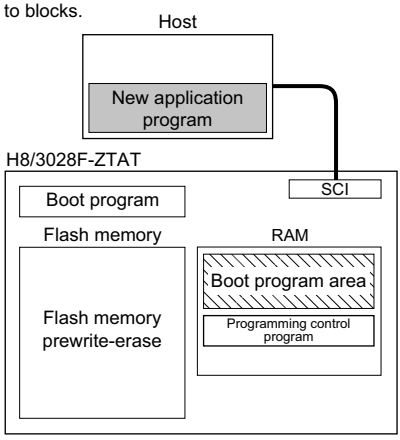
State transitions between the normal and user modes and on-board programming mod performed by changing the FWE pin level from high to low or from low to high. To p misoperation (erroneous programming or erasing) in these cases, the bits in the flash r control register (FLMCR1) should be cleared to 0 before making such a transition. Af are cleared, a wait time is necessary. Normal operation is not guaranteed if this wait ti insufficient.

automatically transferred to the RAM boo
program area.

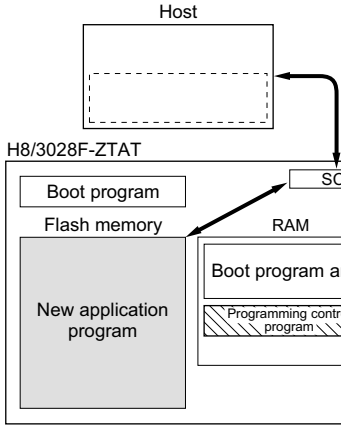


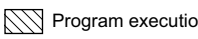


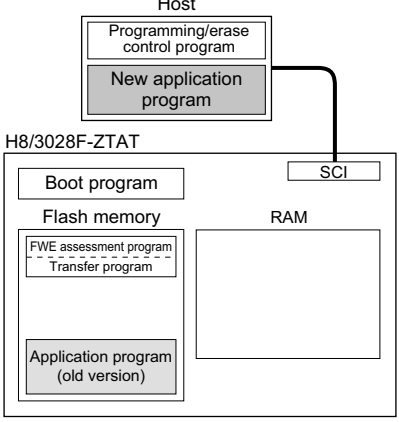
3. Flash memory initialization
The erase program in the boot program area (in RAM) is executed, and the flash memory is initialized (to H'FF). In boot mode, total flash memory erasure is performed, without regard to blocks.

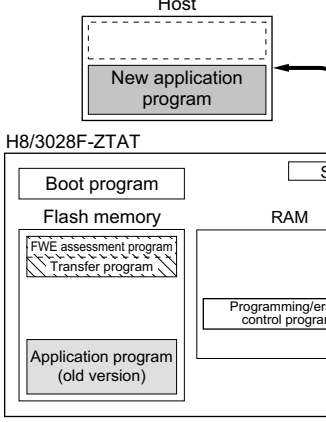


4. Writing new application program
The programming control program transfe
from the host to RAM is executed, and th
application program in the host is written
the flash memory.

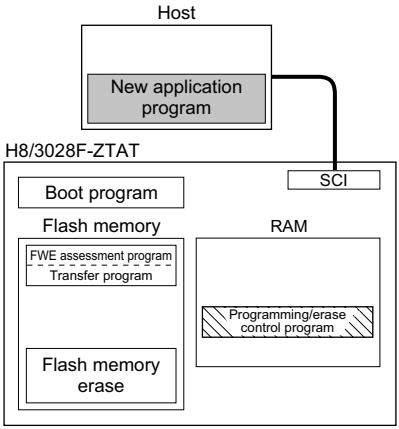




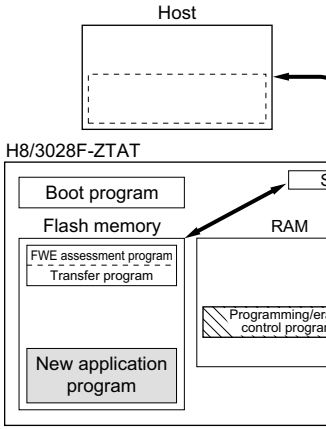




3. Flash memory initialization
   The programming/erase program in RAM is executed, and the flash memory is initialized (to H'FF). Erasing can be performed in block units, but not in byte units.



4. Writing new application program
   Next, the new application program in the
   written into the erased flash memory blo
   Do not write to unerased blocks.



Program execut

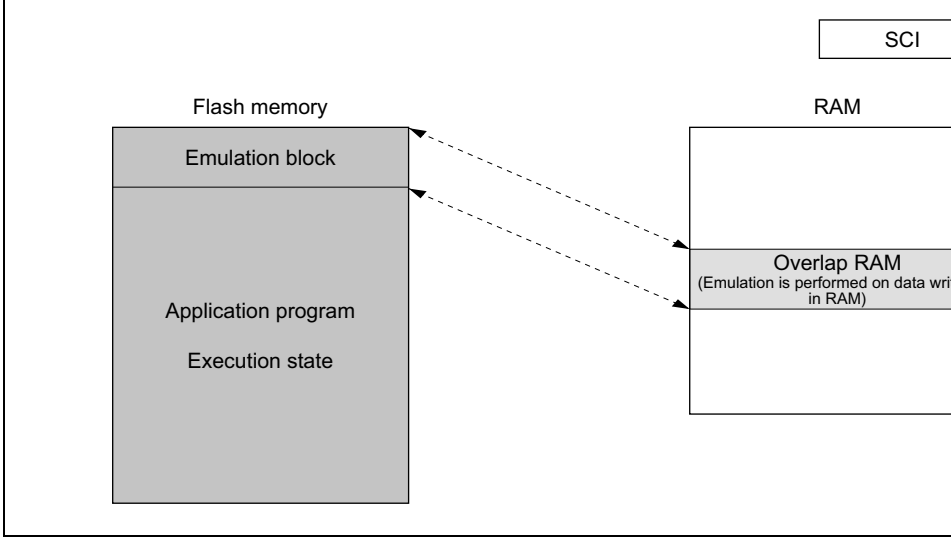


Figure 18.3 Reading Overlap RAM Data in User Mode/User Program M

When overlap RAM data is confirmed, clear the RAMS bit to cancel RAM overlap, an perform writes to the flash memory in user program mode.

When the programming control program is transferred to RAM in on-board programmi ensure that the transfer destination and the overlap RAM do not overlap, as this will ca the overlap RAM to be rewritten.

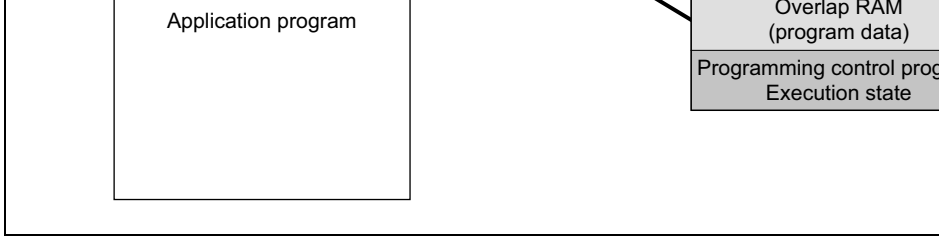
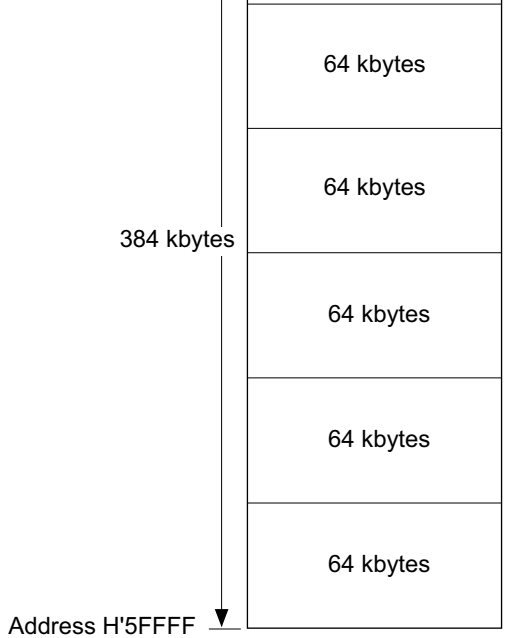


Figure 18.4 Writing Overlap RAM Data in User Program Mode


64 kbytes
64 kbytes
384 kbytes
64 kbytes
64 kbytes
64 kbytes
Address H'5FFFF

ROM enabled).

Table 18.6 On-Board Programming Mode Settings

Mode		FWE	MD_2	MD_1	M
Boot mode	Mode 5	1*1	0*2	0	1
	Mode 7		0*2	1	1
User program mode	Mode 5		1	0	1
	Mode 7		1	1	1

Notes: 1. For the High level input timing, see items 6 and 7 of Notes on Use of Boot M

2. In boot mode, the MD_2 setting should be the inverse of the input.
In the boot mode in the H8/3028F-ZTAT, the levels of the mode pins (MD_2 t
reflected in mode select bits 2 to 0 (MDS2 to MDS0) in the mode control re
(MDCR).

programming control program area in on-chip RAM. After the transfer is completed, co
branches to the start address (H'FFC720) of the programming control program area and
programming control program execution state is entered (flash memory programming/e
be performed).

Figure 18.5 shows a system configuration diagram when using boot mode, and figure 1
the boot program mode execution procedure.

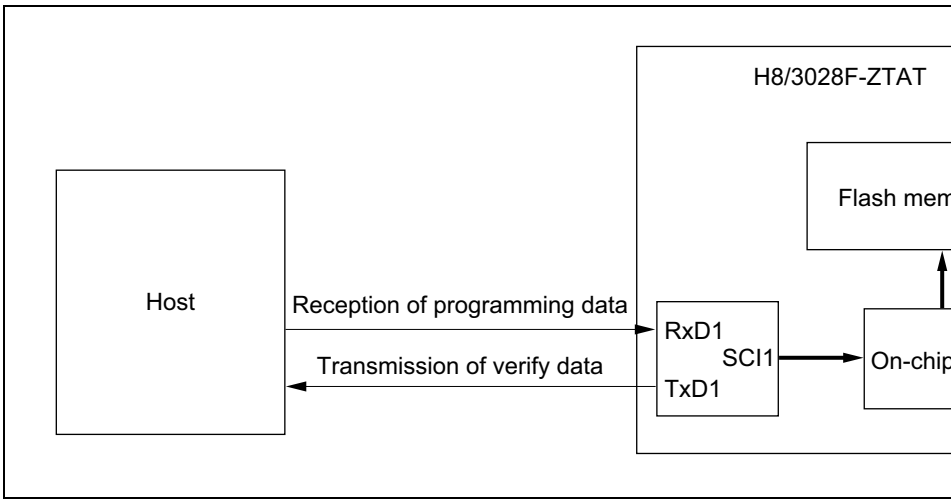
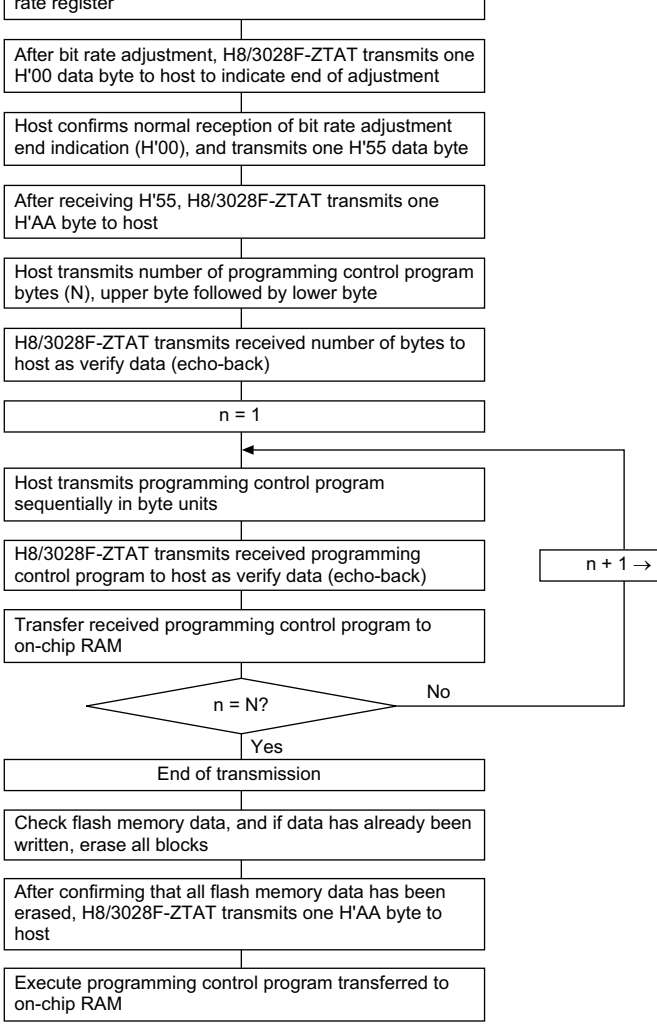


Figure 18.5 System Configuration When Using Boot Mode



Note: If a memory cell does not operate normally and cannot be erased, one H'FF byte is transmitted as an era indication, and the erase operation and subsequent operations are halted.

Figure 18.6 Boot Mode Execution Procedure

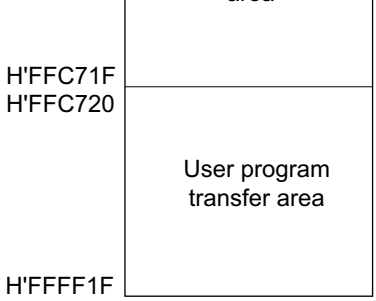
When boot mode is initiated, the H8/3028F-ZTAT measures the low period of the asyn
SCI communication data (H'00) transmitted continuously from the host. The SCI transm
format should be set as 8-bit data, 1 stop bit, no parity. The H8/3028F-ZTAT calculates
of the transmission from the host from the measured low period, and transmits one H'00
the host to indicate the end of bit rate adjustment. The host should confirm that this adj
end indication (H'00) has been received normally, and transmit one H'55 byte to the H8
ZTAT. If reception cannot be performed normally, initiate boot mode again (reset), and
above operations. Depending on the host's transmission bit rate and the H8/3028F-ZTA
clock frequency, there will be a discrepancy between the bit rates of the host and the H8
ZTAT. To ensure correct SCI operation, the host's transfer bit rate should be set to 480
19,200 bps*.

Table 18.7 shows typical host transfer bit rates and system clock frequencies for which
adjustment of the H8/3028F-ZTAT bit rate is possible. The boot program should be exe
within this system clock range.

Table 18.7 System Clock Frequencies for which Automatic Adjustment of H8/30 Bit Rate is Possible

Host Bit Rate (bps)	System Clock Frequency for which Automatic Adjustment of H8/3028F-ZTAT Bit Rate is Possible (M
19,200	16 to 25
9,600	8 to 25
4,800	4 to 25

Note: * Only use a setting of 4800, 9600, or 19200 bps for the host's bit rate. No other
be used.
Although the H8/3028F-ZTAT may also perform automatic bit rate adjustment
rate and system clock combinations other than those shown in table 18.7, a deg
will arise between the bit rates of the host and the H8/3028F-ZTAT, and subsec
transfer will not be performed normally. Therefore, only a combination of bit ra
system clock frequency within one of the ranges shown in table 18.7 can be use
mode execution.



Note: The boot program area cannot be used until a transition is made to the execu
for the user program transferred to RAM. Note also that the boot program rer
this area in RAM even after control branches to the user program.

Figure 18.7 RAM Areas in Boot Mode

Notes on Use of Boot Mode:

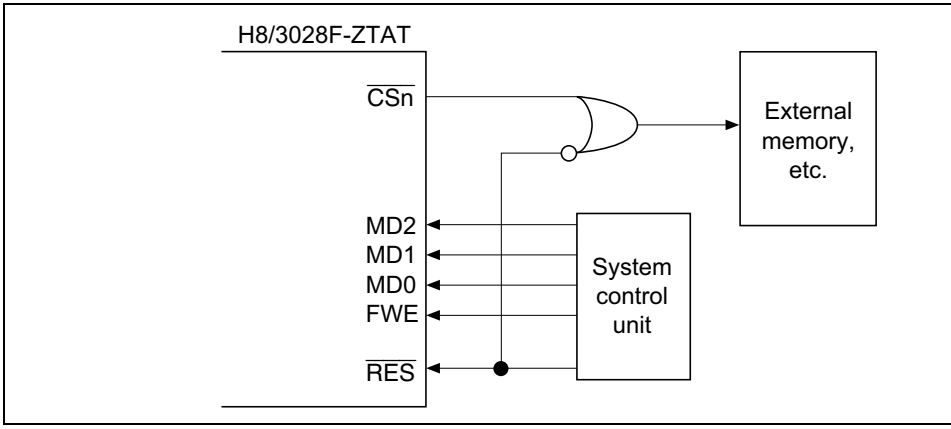
1. When the H8/3028F-ZTAT chip comes out of reset in boot mode, it measures the
of the input at the SCI's RxD_1 pin. The reset should end with RxD_1 high. After the
it takes about 100 states for the chip to get ready to measure the low period of the
2. In boot mode, if any data has been programmed into the flash memory (if all data i
flash memory blocks are erased. Boot mode is for use when user program mode is
such as the first time on-board programming is performed, or if the program activa
program mode is accidentally erased.
3. Interrupts cannot be used while the flash memory is being programmed or erased.
4. The RxD_1 and TxD_1 lines should be pulled up on the board.
5. Before branching to the user program the H8/3028F-ZTAT terminates transmit an
operations by the on-chip SCI (channel 1) (by clearing the RE and TE bits to 0 in t
control register (SCR)), but the adjusted bit rate value remains set in the bit rate re
(BRR). The transmit data output pin, TxD_1, goes to the high-level output state (P9
P9DDR, $P9_1DR$ = 1 in P9DR).

first be cleared by reset input via the RES pin*1. The RES pin must be held low
20 system clock cycles.*3

b. Do not change the input levels of the mode pins (MD_2 to MD_0) or the FWE pin
mode. To change the mode, the $\overline{RES}$ pin must first be driven low to set the reset
if a watchdog timer reset occurs in the boot mode state, the MCU's internal state
be cleared, and the on-chip boot program will be restarted regardless of the mod
states.

c. The FWE pin must not be driven low while the boot program is running or flash
being programmed or erased*2.

7. If the mode pin input levels are changed (for example, from low to high) during a re
state of ports with multiplexed address functions and bus control output signals ($\overline{CS}$
$\overline{LWR}$, $\overline{HWR}$) may also change according to the change in the MCU's operating mo
Therefore, care must be taken to make pin settings to prevent these pins from being
directly as output signal pins during a reset, or to prevent collision with signals outs
MCU.



Notes: 1. Mode pin and FWE pin input must satisfy the mode programming setup tim
with respect to the reset release timing.

2. For further information on FWE application and disconnection, see section
Flash Memory Programming and Erasing Precautions.

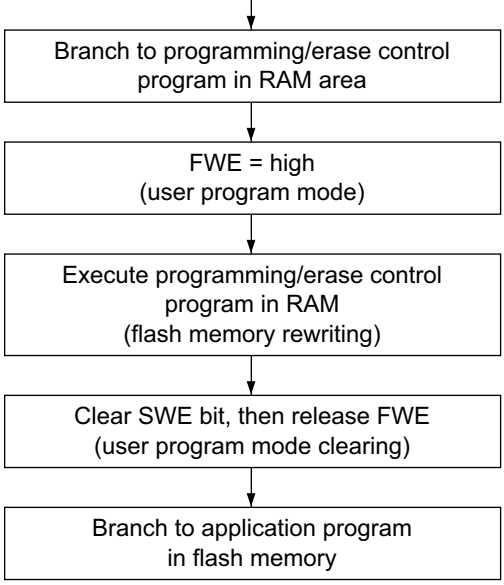
programming data, and storing a program/erase control program in part of the program necessary.

To select user program mode, select a mode that enables the on-chip ROM (mode 5 or apply a high level to the FWE pin. In this mode, on-chip supporting modules other tha memory operate as they normally would in modes 5 and 7.

Flash memory programming/erasing should not be carried out in mode 6. When mode FWE pin must be driven low.

The flash memory itself cannot be read while being programmed or erased, so the pro performs programming should be placed in external memory or transferred to RAM a there.

Figure 18.8 shows the execution procedure when user program mode is entered during execution in RAM. It is also possible to start from user program mode in a reset-start.



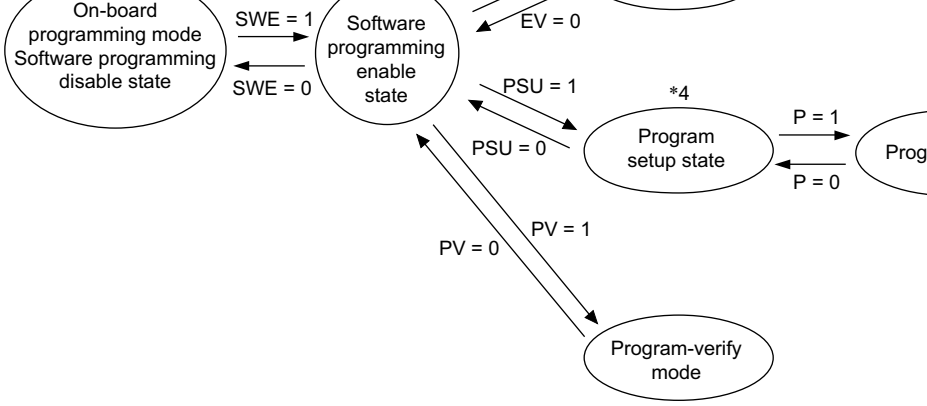
Notes: 1. Do not apply a constant high level to the FWE pin. A high level should be ap
FWE pin only when programming or erasing flash memory (including execut
memory emulation by RAM). Also, while a high level is applied to the FWE p
watchdog timer should be activated to prevent overprogramming or overeras
program runaway, etc.
2. For further information on FWE application and disconnection, see section 1
Memory Programming and Erasing Precautions.
3. In order to execute a normal read of flash memory in user program mode, th
programming/erase program must not be executing. It is thus necessary to e
bits 6 to 0 in FLMCR1 are cleared to 0.

Figure 18.8 Example of User Program Mode Execution Procedure

on-chip RAM or external memory.

See section 18.11, Flash Memory Programming and Erasing Precautions, for points to when programming or erasing the flash memory. In the following operation descriptio times after setting or clearing individual bits in FLMCR1 are given as parameters; for the wait times, see section 21.2.6, Flash Memory Characteristics.

Notes: 1. Operation is not guaranteed if setting/resetting of the SWE, ESU, PSU, EV P bits in FLMCR1 is executed by a program in flash memory.
2. When programming or erasing, set FWE to 1 (programming/erasing will n executed if FWE = 0).
3. Programming must be executed in the erased state. Do not perform additio programming on addresses that have already been programmed.



Notes: In order to perform a normal read of flash memory, SWE must be cleared to 0. Also note that ve
can be performed during the programming/erasing process.

1. : Normal mode : On-board programming mode
2. Do not make a state transition by setting or clearing multiple bits simultaneously.
3. After a transition from erase mode to the erase setup state, do not enter erase mode without
through the software programming enable state.
4. After a transition from program mode to the program setup state, do not enter program mode
passing through the software programming enable state.

Figure 18.9 FLMCR1 Bit Settings and State Transitions

the maximum number of programming operations (N) are shown in table 21.19 in sec
Flash Memory Characteristics.

Following the elapse of (t_{sswe}) µs or more after the SWE bit is set to 1 in FLMCR1, 12
is written consecutively to the write addresses. The lower 8 bits of the first address wr
be H'00 and H'80, 128 consecutive byte data transfers are performed. The program ad
program data are latched in the flash memory. A 128-byte data transfer must be perfor
writing fewer than 128 bytes; in this case, H'FF data must be written to the extra addre

Next, the watchdog timer (WDT) is set to prevent overprogramming due to program r
Set a value greater than ($t_{spsu} + t_{sp} + t_{cp} + t_{cpsu}$) µs as the WDT overflow period. Prepar
entering program mode (program setup) is performed next by setting the PSU bit in FL
The operating mode is then switched to program mode by setting the P bit in FLMCR
elapse of at least (t_{spsu}) µs. The time during which the P bit is set is the flash memory
programming time. Make a program setting so that the time for one programming ope
within the range of (t_{sp}) µs.

The wait time after P bit setting must be changed according to the degree of progress t
programming operation. For details see “Notes on Program/Program-Verify Procedur

of H'FF data should be made to the addresses to be read. The dummy write should be ex after the elapse of (t_{spv}) µs or more. When the flash memory is read in this state (verify in 16-bit units), the data at the latched address is read. Wait at least (t_{spvr}) µs after the du before performing this read operation. Next, the originally written data is compared wit data, and reprogram data is computed (see figure 18.10) and transferred to RAM. After verification of 128 bytes of data has been completed, exit program-verify mode, wait fo (t_{cpv}) µs, then clear the SWE bit in FLMCR1. If reprogramming is necessary, set progra again, and repeat the program/program-verify sequence as before. The maximum numb repetitions of the program/program-verify sequence is indicated by the maximum progr count (N). Leave a wait time of at least (t_{cswe}) µs after clearing SWE.

Notes on Program/Program-Verify Procedure

1. The program/program-verify procedure for the H8/3028F-ZTAT uses a 128-byte-un programming algorithm.

   In order to perform 128-byte-unit programming, the lower 8 bits of the write start ad be H'00 or H'80.

2. When performing continuous writing of 128-byte data to flash memory, byte-unit tr should be used.

   128-byte data transfer is necessary even when writing fewer than 128 bytes of data. H'FF data to the extra addresses.

3. Verify data is read in word units.

4. The write pulse is applied and a flash memory write executed while the P bit in FLM set. In the H8/3028F-ZTAT, write pulses should be applied as follows in the program/program-verify procedure to prevent voltage stress on the device and loss o data reliability.

   a. After write pulse application, perform a verify-read in program-verify mode and write pulse again for any bits read as 1 (reprogramming processing). When all th bits in the 128-byte write data are read as 0 in the verify-read operation, the program/program-verify procedure is completed. In the H8/3028F-ZTAT, the nu loops in reprogramming processing is guaranteed not to exceed the maximum va maximum programming count (N).

processing.

When programming is completed at a late stage in the program/program-verify

If programming is completed in the 7th or later reprogramming processing loo programming is not necessary for the relevant bits.

c. If programming of other bits is incomplete in the 128 bytes, reprogramming pr should be executed. If a bit for which programming has been judged to be com read as 1 in a subsequent verify-read, a write pulse should again be applied to t

5. The period for which the P bit in FLMCR1 is set (the write pulse width) should be according to the degree of progress through the program/program-verify procedure detailed wait time specifications, see section 21.2.6, Flash Memory Characteristics

Item	Symbol	Item
Wait time after P bit setting	t_{sp}	When reprogramming loop count (n) is 1 to 6
		When reprogramming loop count (n) is 7 or more
		In case of additional programming processing*

Note: * Additional programming processing is necessary only when the reprogramming (n) is 1 to 6.

6. The program/program-verify flowchart for the H8/3028F-ZTAT is shown in figure

To cover the points noted above, bits on which reprogramming processing is to be and bits on which additional programming is to be executed, must be determined a below.

Since reprogram data and additional-programming data vary according to the prog programming procedure, it is recommended that the following data storage areas (each) be provided in RAM.

1	0	1	—
1	1	1	Still in erased state: no action

Legend

(D): Source data of bits on which programming is executed

(X): Source data of bits on which reprogramming is executed

Additional-Programming Data Computation Table

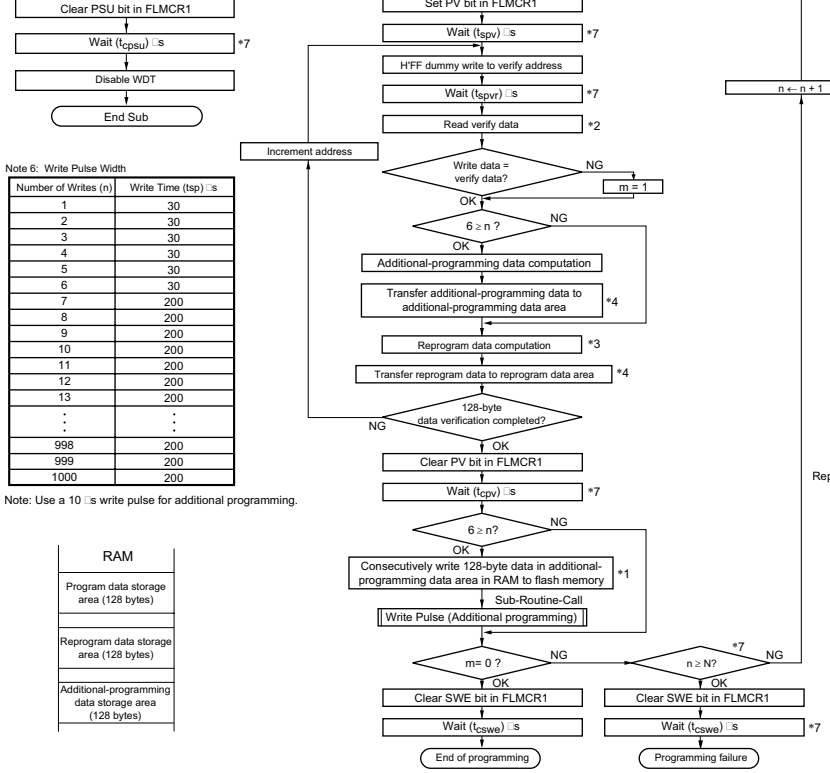
(X')	Result of Verify-Read after Write Pulse Application (V)	(Y) Result of Operation	Comments
0	0	0	Programming by write pulse appli judged to be completed: additiona programming processing to be ex
0	1	1	Programming by write pulse appli incomplete: additional programmi processing not to be executed
1	0	1	Programming already completed: programming processing not to be
1	1	1	Still in erased state: no action

Legend

(Y): Data of bits on which additional programming is executed

(X'): Data of bits on which reprogramming is executed in a certain reprogramming loop

7. It is necessary to execute additional programming processing during the course of t H8/3028F-ZTAT program/program-verify procedure. However, once 128-byte-unit programming is finished, additional programming should not be carried out on the s address area. When executing reprogramming, an erase must be executed first. Note normal operation of reads, etc., is not guaranteed if additional programming is perfo addresses for which a program/program-verify operation has finished.



Note 6: Write Pulse Width

Number of Writes (n)	Write Time (tsp) □s
1	30
2	30
3	30
4	30
5	30
6	30
7	200
8	200
9	200
10	200
11	200
12	200
13	200
⋮	⋮
998	200
999	200
1000	200

Note: Use a 10 □s write pulse for additional programming.

Notes: 1. Data transfer is performed by byte transfer. The lower 8 bits of the first address written to must be H'00 or H'80.
A 128-byte data transfer must be performed even if writing fewer than 128 bytes; in this case, H'FF data must be written to the extra addresses.
2. Verify data is read in 16-bit (word) units.
3. Reprogram data is determined by the operation shown in the table below (comparison between the data stored in the program data area and the verify data). B which the reprogram data is 0 are programmed in the next reprogramming loop. Therefore, even bits for which programming has been completed will be subjec programming once again if the result of the subsequent verify operation is NG.
4. A 128-byte area for storing program data, a 128-byte area for storing reprogram data, and a 128-byte area for storing additional-programming data must be pro RAM. The contents of the reprogram data area and additional-programming data area are modified as programming proceeds.
5. A write pulse of 30 □s or 200 □s is applied according to the progress of the programming operation. See Note 6 for details of the pulse widths. When writing of additional-programming data is executed, a 10 □s write pulse should be applied. Reprogram data X' means reprogram data when the write pulse is applied.
7. The wait times and value of N are shown in section 21.2.6, Flash Memory Characteristics.

Reprogram Data Computation Table

Original Data (D)	Verify Data (V)	Reprogram Data (X)	Comments
0	0	1	Programming completed
0	1	0	Programming incomplete; reprogram
1	0	1	
1	1	1	Still in erased state; no action

Additional-Programming Data Computation Table

Reprogram Data (X')	Verify Data (V)	Additional-Programming Data (Y)	Comme
0	0	0	Additional progra to be executed
0	1	1	Additional progra not to be execute
1	0	1	Additional progra not to be execute
1	1	1	Additional progra not to be execute

Figure 18.10 Program/Program-Verify Flowchart (128-Byte Programm

To erase flash memory contents, make a 1-bit setting for the flash memory area to be er
erase block register 1 and 2 (EBR1, EBR2) at least (t_{sswe}) µs after setting the SWE bit to
FLMCR1. Next, the watchdog timer (WDT) is set to prevent overerasing due to progra
runaway, etc. Set a value greater than (t_{se}) ms + (t_{sesu} + t_{ce} + t_{cesu}) µs as the WDT overfl
Preparation for entering erase mode (erase setup) is performed next by setting the ESU
FLMCR1. The operating mode is then switched to erase mode by setting the E bit in F
after the elapse of at least (t_{sesu}) µs. The time during which the E bit is set is the flash m
erase time. Ensure that the erase time does not exceed (t_{se}) ms.

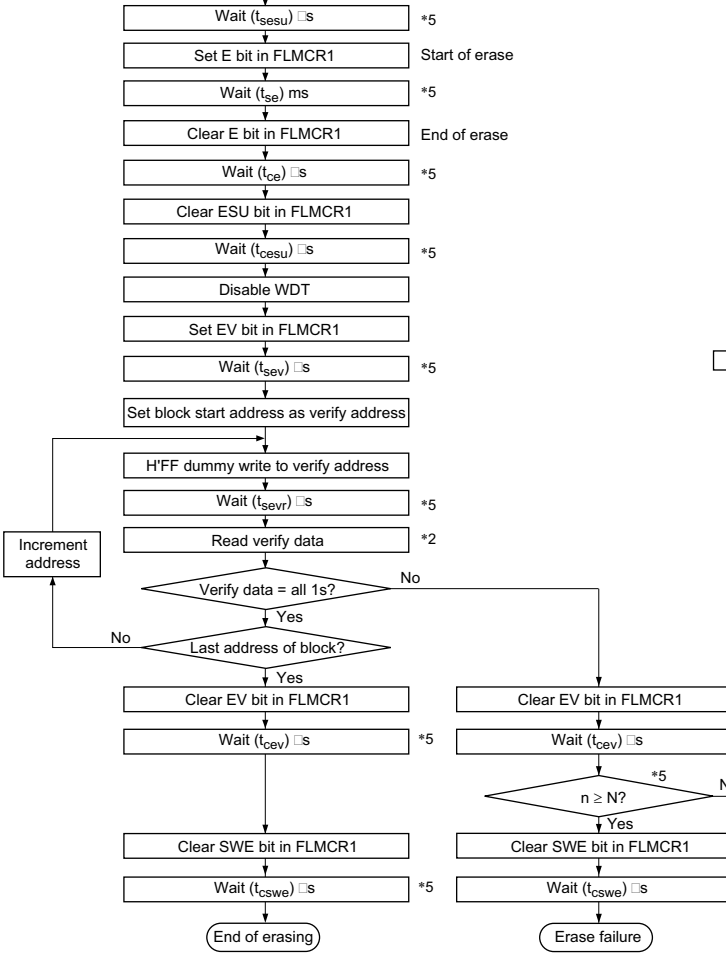
Note: With flash memory erasing, preprogramming (setting all memory data in the m
be erased to all 0) is not necessary before starting the erase procedure.

18.6.4 Erase-Verify Mode

In erase-verify mode, data is read after memory has been erased to check whether it has
correctly erased.

After the elapse of the fixed erase time, clear the E bit in FLMCR1, then wait for at leas
before clearing the ESU bit to exit erase mode. After exiting erase mode, the watchdog
setting is also cleared. The operating mode is then switched to erase-verify mode by se
EV bit in FLMCR1. Before reading in erase-verify mode, a dummy write of H'FF data
made to the addresses to be read. The dummy write should be executed after the elapse
or more. When the flash memory is read in this state (verify data is read in 16-bit units)
the latched address is read. Wait at least (t_{sevr}) µs after the dummy write before perform
read operation. If the read data has been erased (all 1), a dummy write is performed to t
address, and erase-verify is performed. If the read data is unerased, set erase mode agai
repeat the erase/erase-verify sequence as before. The maximum number of repetitions o
erase/erase-verify sequence is indicated by the maximum erase count (N). When verific
completed, exit erase-verify mode, and wait for at least (t_{cev}) µs. If erasure has been con
all the erase blocks, clear the SWE bit in FLMCR1, and leave a wait time of at least (t_{cs}

If erasing multiple blocks, set a single bit in EBR1/EBR2 for the next block to be erase
repeat the erase/erase-verify sequence as before.



Notes: 1. Prewriting (setting erase block data to all 0s) is not necessary.
2. Verify data is read in 16-bit (word) units.
3. Make only a single-bit specification in the erase block registers (EBR1 and EBR2). Two or more bits must not be set si
4. Erasing is performed in block units. To erase multiple blocks, each block must be erased in turn.
5. The wait times and the value of N are shown in section 21.2.6, Flash Memory Characteristics.

Figure 18.11 Erase/Erase-Verify Flowchart (Single-Block Erasing)

erase block registers 1 and 2 (EBR1, EBR2) are reset. In the error protection state, the F
EBR1, and EBR2 settings are retained; the P bit and E bit can be set, but a transition is
to program mode or erase mode. (See table 18.8.)

Table 18.8 Hardware Protection

		Function	
Item	**Description**	**Program**	**Erase**
FWE pin protection	• When a low level is input to the FWE pin, FLMCR1, EBR1, and EBR2 are initialized, and the program/erase-protected state is entered.	Not possible*1	Not possible*3
Reset/ standby protection	• In a reset (including a WDT overflow reset) and in standby mode, FLMCR1, FLMCR2, EBR1, and EBR2 are initialized, and the program/erase-protected state is entered. • In a reset via the $\overline{\text{RES}}$ pin, the reset state is not entered unless the $\overline{\text{RES}}$ pin is held low until oscillation stabilizes after powering on. In the case of a reset during operation, hold the $\overline{\text{RES}}$ pin low for the $\overline{\text{RES}}$ pulse width specified in the AC Characteristics section.*4	Not possible	Not possible*3
Error protection	• When a microcomputer operation error (error generation (FLER = 1)) was detected while flash memory was being programmed/erased, error protection is enabled. At this time, the FLMCR1, EBR1, and EBR2 settings are held, but programming/erasing is aborted at the time the error was generated. Error protection is released only by a reset via the $\overline{\text{RES}}$ pin or a WDT reset, or in the hardware standby mode.	Not possible	Not possible*3

Notes: 1. The RAM area that overlapped flash memory is deleted.
2. It is possible to perform a program-verify operation on the 128 bytes being programmed, or an erase-verify operation on the block being erased.

protection, setting the P or E bit in the flash memory control register 1 (FLMCR1) doe
a transition to program mode or erase mode. (See table 18.9.)

Table 18.9 Software Protection

		Functions	
Item	**Description**	**Program**	**Erase**
Block protection	• Erase protection can be set for individual blocks by settings in erase block register 1 (EBR1) and erase block register 2 (EBR2)*2. However, programming protection is disabled. • Setting EBR1 and EBR2 to H'00 places all blocks in the erase-protected state.	—	Not possible
Emulation protection	• Setting the RAMS bit 1 in RAMCR places all blocks in the program/erase-protected state.	Not possible*1	Not possible*3

Notes: 1. The RAM area overlapping flash memory can be written to.
2. When not erasing, set EBR1 and EBR2 to H'00.
3. All blocks are unerasable and block-by-block specification is not possible.

18.7.3 Error Protection

In error protection, an error is detected when MCU runaway occurs during flash mem
programming/erasing*1, or operation is not performed in accordance with the program
algorithm, and the program/erase operation is aborted. Aborting the program/erase op
prevents damage to the flash memory due to overprogramming or overerasing.

If the MCU malfunctions during flash memory programming/erasing, the FLER bit is
the flash memory status register (FLMSR2) and the error protection state is entered. F
FLMCR2, EBR1, and EBR2 settings*3 are retained, but program mode or erase mode
the point at which the error occurred. Program mode or erase mode cannot be re-enter

3. When a SLEEP instruction (including software standby) is executed during programming/erasing
4. When the bus is released during programming/erasing

   Error protection is released only by a $\overline{\text{RES}}$ pin or WDT reset, or in hardware standb

Notes: 1. State in which the P bit or E bit in FLMCR1 is set to 1. Note that NMI input in this state.
2. It is possible to perform a program-verify operation on the 128 bytes being programmed, or an erase-verify on the block being erased.
3. FLMCR1, EBR1, and EBR2 can be written to. However, the registers are in a transition is made to software standby mode while in the error protection s

Figure 18.12 shows the flash memory state transition diagram.

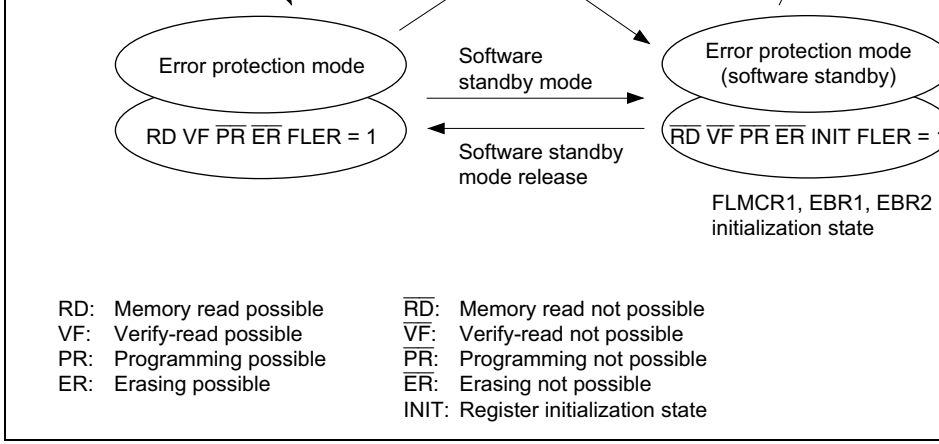


Figure 18.12 Flash Memory State Transitions
(When High Level is Applied to FWE Pin in Mode 5 or 7 (On-Chip ROM En

The error protection function is invalid for abnormal operations other than the FLER b conditions. Also, if a certain time has elapsed before this protection state is entered, da already have been caused to the flash memory. Consequently, this function cannot pro complete protection against damage to flash memory.

To prevent such abnormal operations, therefore, it is necessary to ensure correct opera accordance with the program/erase algorithm, with the flash write enable (FWE) volta and to conduct constant monitoring for MCU errors, internally and externally, using th timer or other means. There may also be cases where the flash memory is in an errone programming or erroneous erasing state at the point of transition to this protection mo programming or erasing is not properly carried out because of an abort. In cases such forced recovery (program rewrite) must be executed using boot mode. However, it ma happen that boot mode cannot be normally initiated because of overprogramming or o

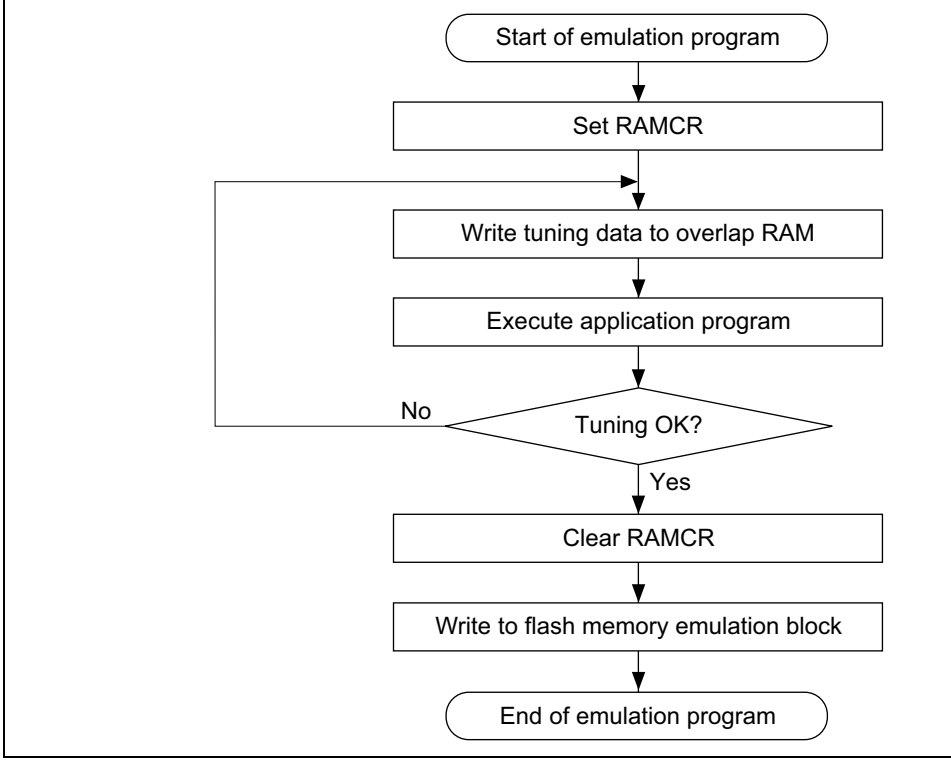


Figure 18.13 Flowchart of Flash Memory Emulation in RAM

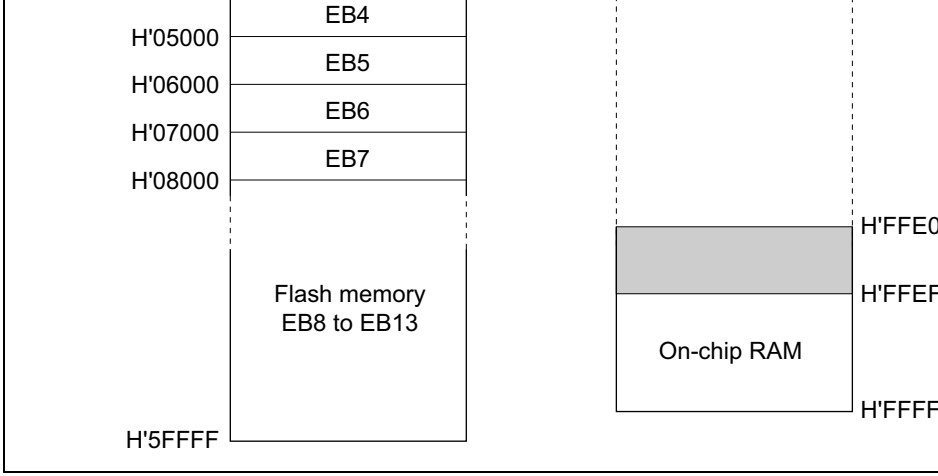


Figure 18.14 Example of RAM Overlap Operation

Example of Flash Memory Block Area EB0 Overlapping

1. Set bits RAMS and RAM2 to RAM0 in RAMCR to 1,0, 0, 0, to overlap part of RA area (EB0) for which realtime programming is required.
2. Realtime programming is performed using the overlapping RAM.
3. After the program data has been confirmed, the RAMS bit is cleared, releasing RA
4. The data written in the overlapping RAM is written into the flash memory space (I

Notes: 1. When the RAMS bit is set to 1, program/erase protection is enabled for all regardless of the value of RAM2 to RAM0 (emulation protection). In this s the P or E bit in FLMCR1 will not cause a transition to program mode or e When actually programming or erasing a flash memory area, the RAMS bi cleared to 0.

2. A RAM area cannot be erased by execution of software in accordance with algorithm while flash memory emulation in RAM is being used.
3. Block area EB0 contains the vector table. When performing RAM emulati vector table is needed in the overlap RAM.

FLMCR1 is 0 while a high level is being input to the FWE pin.

18.9 NMI Input Disabling Conditions

All interrupts, including NMI input, should be disabled while flash memory is being pr or erased (while the P bit or E bit is set in FLMCR1), and while the boot program is exe boot mode*[1], to give priority to the program or erase operation. There are three reasons

1. NMI input during programming or erasing might cause a violation of the programm erasing algorithm, with the result that normal operation could not be assured.
2. In the NMI exception handling sequence during programming or erasing, the vecto be read correctly*[2], possibly resulting in MCU runaway.
3. If NMI input occurred during boot program execution, it would not be possible to e normal boot mode sequence.

For these reasons, in on-board programming mode alone there are conditions for disabl input, as an exception to the general rule. However, this provision does not guarantee n erasing and programming or MCU operation. All interrupt requests (exception handling release), including NMI, must therefore be restricted inside and outside the MCU durin application. NMI input is also disabled in the error protection state and while the P or E remains set in FLMCR1 during flash memory emulation in RAM.

Notes: 1. This is the interval until a branch is made to the boot program area in the on (This branch takes place immediately after transfer of the user program is co Consequently, after the branch to the RAM area, NMI input is enabled exce programming and erasing. Interrupt requests must therefore be disabled insi outside the MCU until the user program has completed initial programming the vector table and the NMI interrupt handling routine).

2. The vector may not be read correctly in this case for the following two reaso
   - If flash memory is read while being programmed or erased (while the P b is set in FLMCR1), correct read data will not be obtained (undetermined be returned).

18.10.1 Socket Adapters and Memory Map

In PROM mode using a PROM writer, memory reading (verification) and writing and
memory initialization (total erasure) can be performed. For these operations, a special
adapter is mounted in the PROM writer. The socket adapter product codes are given in
18.10. In the H8/3028F-ZTAT PROM mode, only the socket adapters shown in this ta
be used.

Table 18.10 H8/3028F-ZTAT Socket Adapter Product Codes

Product Code	Package	Socket Adapter Product Code	Manufactur
HD64F3028F	100-pin QFP (FP-100B)	ME3024ESHF1H	MINATO ELECTRON
HD64F3028TE	100-pin TQFP (TFP-100B)	ME3024ESNF1H	
HD64F3028F	100-pin QFP (FP-100B)	HF302BQ100D4001	DATA I/O JA
HD64F3028TE	100-pin TQFP (TFP-100B)	HF302BT100D4001	

Figure 18.15 shows the memory map in PROM mode.

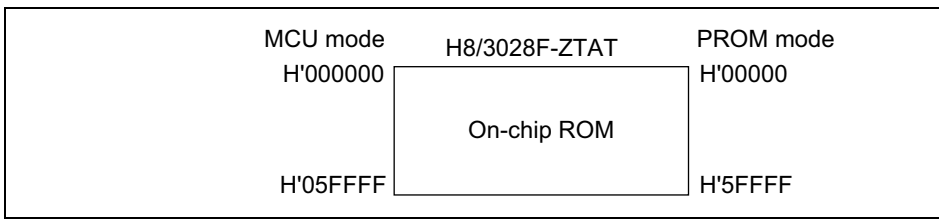


Figure 18.15 Memory Map in PROM Mode

For samples for which the erasure history is unknown, it is recommended that erasi executed to check and correct the initialization (erase) level.

4. The H8/3028F-ZTAT does not support a product identification mode as used with g purpose EPROMs, and therefore the device name cannot be set automatically in the writer.
5. Refer to the instruction manual provided with the socket adapter, or other relevant documentation, for information on PROM writers and associated program versions compatible with the PROM mode of the H8/3028F-ZTAT.

18.11 Flash Memory Programming and Erasing Precautions

Precautions concerning the use of on-board programming mode, the RAM emulation fu PROM mode are summarized below.

1. Use the specified voltages and timing for programming and erasing.

Applied voltages in excess of the rating can permanently damage the device. Use a programmer that supports the Renesas microcomputer device type "F-ZTAT512" w kbyte on-chip flash memory.

2. Powering on and off (see figures 18.16 to 18.18)

Do not apply a high level to the FWE pin until V_{CC} has stabilized. Also, drive the F low before turning off V_{CC}.

When applying or disconnecting V_{CC} power, fix the FWE pin low and place the flas in the hardware protection state.

The power-on and power-off timing requirements should also be satisfied in the eve power failure and subsequent recovery. Failure to do so may result in overprogramm overerasing due to MCU runaway, and loss of normal memory cell operation.

3. FWE application/disconnection

FWE application should be carried out when MCU operation is in a stable condition operation is not stable, fix the FWE pin low and set the protection state.

The following points must be observed concerning FWE application and disconnect prevent unintentional programming or erasing of flash memory:

unstable.

- In boot mode, apply and disconnect FWE during a reset.

  In a transition to boot mode, FWE = 1 input and MD_2–MD_0 setting should be p
  while the $\overline{RES}$ input is low. FWE and MD_2–MD_0 pin input must satisfy the mo
  programming setup time (t_{MDS}) with respect to the reset release timing. When r
  transition from boot mode to another mode, also, a mode programming setup ti
  necessary with respect to the reset release timing.

  In a reset during operation, the $\overline{RES}$ pin must be held low for a minimum of 20
  clock cycles.

- In user program mode, FWE can be switched between high and low level regar
  $\overline{RES}$ input.

  FWE input can also be switched during execution of a program in flash memor

- Do not apply FWE if program runaway has occurred.

  During FWE application, the program execution state must be monitored using
  watchdog timer or some other means.

- Disconnect FWE only when the SWE, ESU, PSU, EV, PV, E, and P bits in FL
  cleared.

  Make sure that the SWE, ESU, PSU, EV, PV, E, and P bits are not set by mista
  applying or disconnecting FWE.

4. Do not apply a constant high level to the FWE pin.

T prevent erroneous programming or erasing due to program runaway, etc., apply
to the FWE pin only when programming or erasing flash memory (including execu
memory emulation using RAM). A system configuration in which a high level is c
applied to the FWE pin should be avoided. Also, while a high level is applied to th
the watchdog timer should be activated to prevent overprogramming or overerasin
program runaway, etc.

5. Use the recommended algorithm when programming and erasing flash memo

The recommended algorithm enables programming and erasing to be carried out w
subjecting the device to voltage stress or sacrificing program data reliability. Wher
PSU or ESU bit in FLMCR1, the watchdog timer should be set beforehand as a pr
against program runaway, etc.

memory. However, the RAM area overlapping flash memory space can be read and
regardless of whether the SWE bit is set or cleared.

A wait time is necessary after the SWE bit is cleared. For details see table 21.19 in s
21.2.6, Flash Memory Characteristics.

7. **Do not use interrupts while flash memory is being programmed or erased.**

   All interrupt requests, including NMI, should be disabled during FWE application t
   priority to program/erase operations (including emulation in RAM).

   Bus release must also be disabled.

8. **Do not perform additional programming. Erase the memory before reprogram**

   In on-board programming, perform only one programming operation on a 128-byte
   programming unit block. Programming should be carried out with the entire progra
   block erased.

9. **Before programming, check that the chip is correctly mounted in the PROM w**

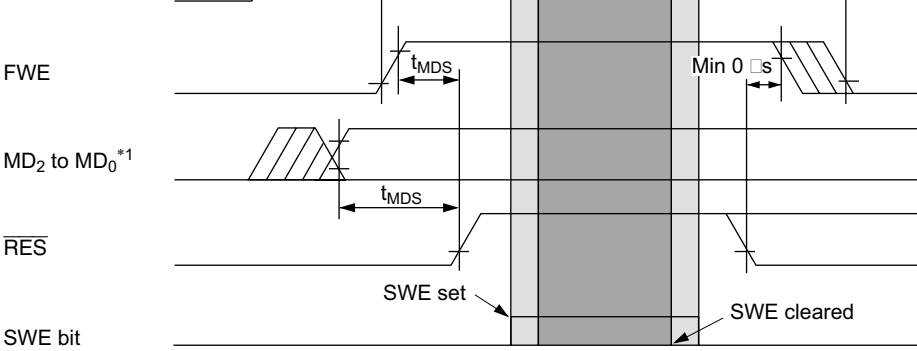
   Overcurrent damage to the device can result if the index marks on the PROM writer
   socket adapter, and chip are not correctly aligned.

10. **Do not touch the socket adapter or chip during programming.**

    Touching either of these can cause contact faults and write errors.

11. **A wait time of 100 µs or more is necessary when performing a read after a tra
    normal mode from program, erase, or verify mode.**

12. **Use byte access on the registers that control the flash memory (FLMCR1, FLM
    EBR1, EBR2, and RAMCR).**

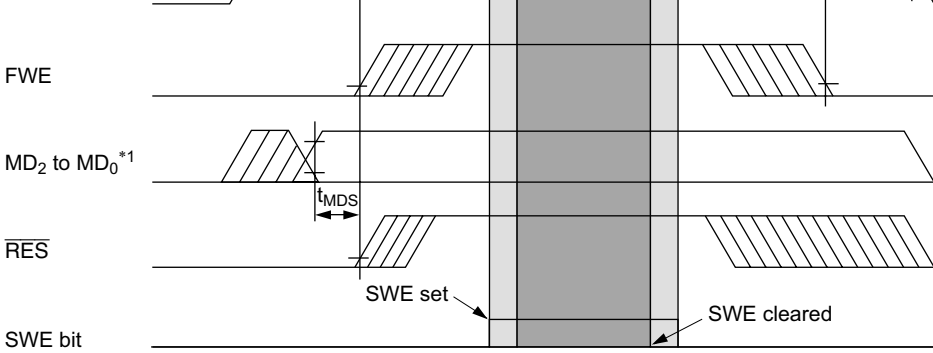


Period during which flash memory access is prohibited
(x: Wait time after setting SWE bit, y: Wait time after clearing SWE bit)*2

Period during which flash memory can be programmed
(Execution of program in flash memory prohibited, and data reads other than verify operatio
prohibited)

Notes: 1. Except when switching modes, the level of the mode pins (MD_2–MD_0) must be fixed u
by pulling the pins up or down.
2. See section 21.2.6, Flash Memory Characteristics.

Figure 18.16 Power-On/Off Timing (Boot Mode)

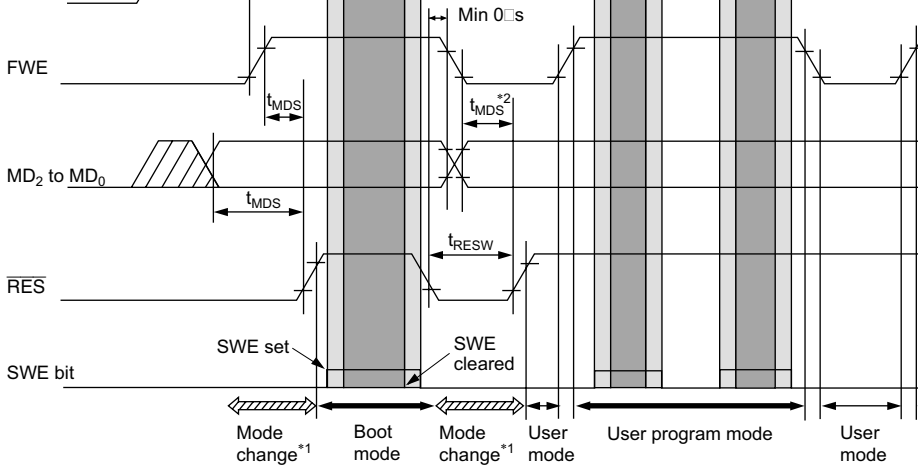


Period during which flash memory access is prohibited
(x: Wait time after setting SWE bit, y: Wait time after clearing SWE bit)*2

Period during which flash memory can be programmed
(Execution of program in flash memory prohibited, and data reads other than verify operation prohibited)

Notes: 1. Except when switching modes, the level of the mode pins (MD_2–MD_0) must be fixed un by pulling the pins up or down.
2. See section 21.2.6, Flash Memory Characteristics.

Figure 18.17 Power-On/Off Timing (User Program Mode)



Period during which flash memory access is prohibited
(x: Wait time after setting SWE bit, y: Wait time after clearing SWE bit)*3

Period during which flash memory can be programmed
(Execution of program in flash memory prohibited, and data reads other than verify operations prohibited)

Notes: 1. When entering boot mode or making a transition from boot mode to another mode, mode switching
out by means of RES input. The state of ports with multiplexed address functions and bus control o
(CSn, AS, RD, WR) will change during this switchover interval (the interval during which the RES pi
and therefore these pins should not be used as output signals during this time.
2. When making a transition from boot mode to another mode, the mode programming setup time t_{MD}
satisfied with respect to RES clearance timing.
3. See section 21.2.6, Flash Memory Characteristics.

Figure 18.18 Mode Transition Timing
(Example: Boot Mode → User Mode ↔ User Program Mode)

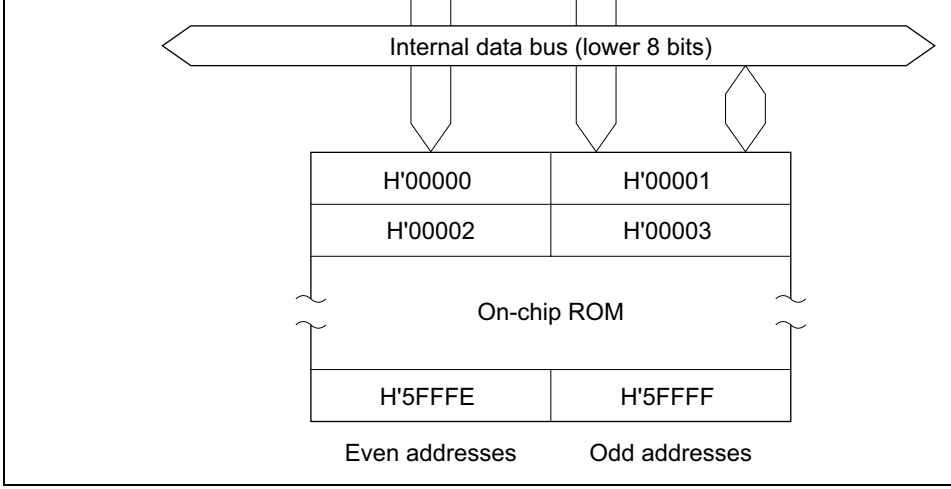
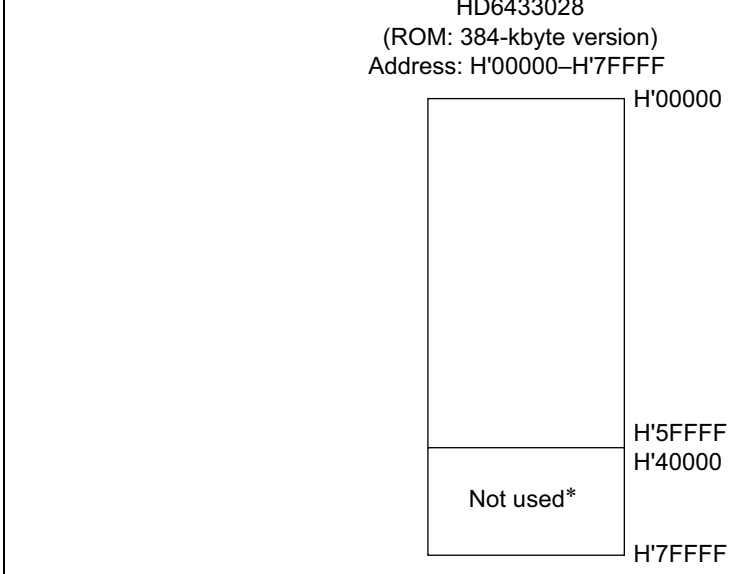


Figure 18.19 ROM Block Diagram



Note: * Write H'FF to all addresses in these areas.

Figure 18.20 ROM Addresses and Data

3. The flash memory control registers (RAMCR, FLMCR1, FLMCR2, EBR1, EBR2 provided in the mask ROM version. Reading these addresses always returns a valu is not possible to write to them. This must be borne in mind when switching from memory version to the mask ROM version.

Register	Bit	Value	F-ZTAT Version	Mask ROM Ver
FLMCR	FWE	0	Application status	— (Not readabl
		1	Overwritable	Application stat read as 1)

modules. Frequency division ratios of 1/1, 1/2, 1/4, and 1/8 can be selected for the fre
divider by settings in a division control register (DIVCR)*[2]. Power consumption in th
reduced in almost direct proportion to the frequency division ratio.

Notes: 1. Usage of the φ pin differs depending on the chip operating mode and the P
setting in the module standby control register (MSTCR). For details, see se
System Clock Output Disabling Function.

2. The division ratio of the frequency divider can be changed dynamically du
operation. The clock output at the φ pin also changes when the division rat
changed. The frequency output at the φ pin is shown below.

$$\phi = EXTAL \times n$$

where, EXTAL: Frequency of crystal resonator or external clock signal

n: Frequency division ratio (n = 1/1, 1/2, 1/4, or 1/8)

19.1.1 Block Diagram

Figure 19.1 shows a block diagram of the clock pulse generator.

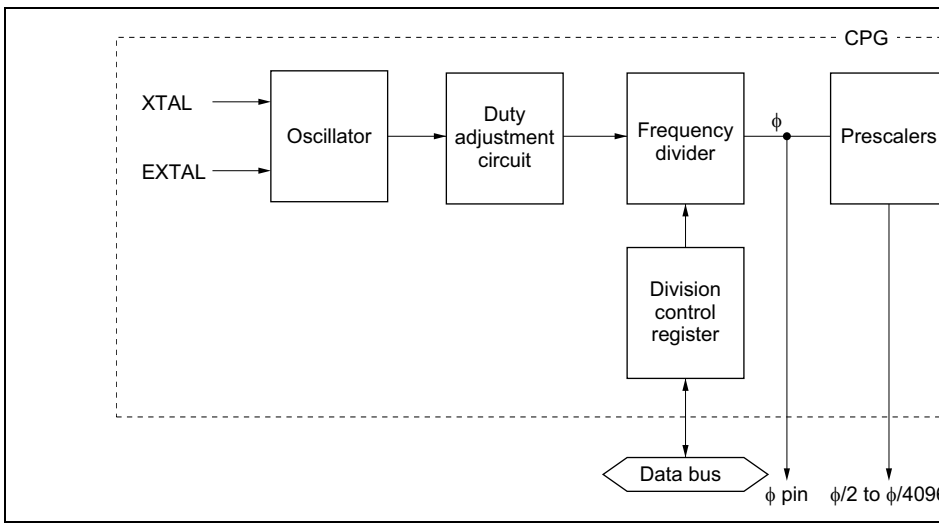


Figure 19.1 Block Diagram of Clock Pulse Generator

C_{L1} and C_{L2} according to table 19.1 (2). An AT-cut parallel-resonance crystal should be

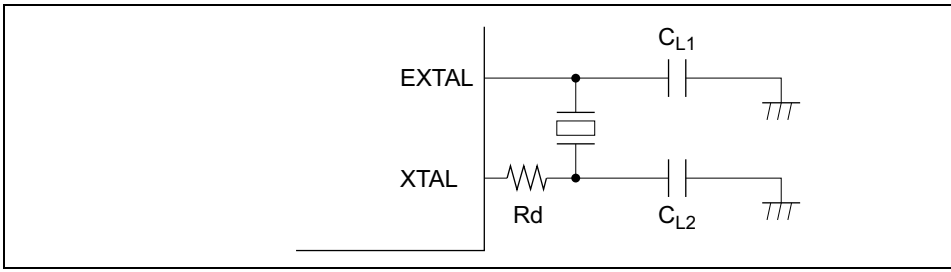


Figure 19.2 Connection of Crystal Resonator (Example)

If a crystal resonator with a frequency higher than 16 MHz is connected, the external lo
capacitance values in table 19.1 (2) should not exceed 10 [pF]. Also, in order to improv
accuracy of the oscillation frequency, a thorough study of oscillation matching evaluati
should be carried out when deciding the circuit constants.

Table 19.1 (1) Damping Resistance Value

Damping Resistance Value	Frequency f (MHz)						
	2	2 < f ≤ 4	4 < f ≤ 8	8 < f ≤ 10	10 < f ≤ 13	13 < f ≤ 16	16 < f ≤ 18
Rd (Ω)	1 k	500	200	0	0	0	0

Note: A crystal resonator between 2 MHz and 25 MHz can be used. If the chip is to be
at less than 2 MHz, the on-chip frequency divider should be used. (A crystal res
less than 2 MHz cannot be used.)

Table 19.1 (2) External Capacitance Values

External Capacitance Value	3.3 V Version	
Frequency f (MHz)	2 ≤ f ≤ 16	16 < f ≤
$C_{L1} = C_{L2}$ (pF)	22	10

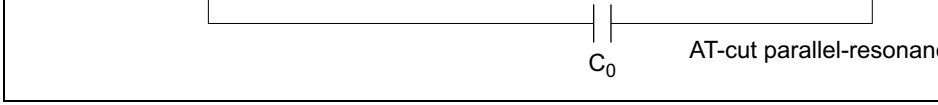


Figure 19.3 Crystal Resonator Equivalent Circuit

Table 19.2 Crystal Resonator Parameters

Frequency f (MHz)	2	4	8	10	12	16	18	
Rs max (Ω)	500	120	80	70	60	50	40	
Co max (pF)	7							

Use a crystal resonator with a frequency equal to the system clock frequency (ϕ).

Notes on Board Design: When a crystal resonator is connected, the following points noted:

Other signal lines should be routed away from the oscillator circuit to prevent inductio interfering with correct oscillation. See figure 19.4.

When the board is designed, the crystal resonator and its load capacitors should be pla as possible to the XTAL and EXTAL pins.

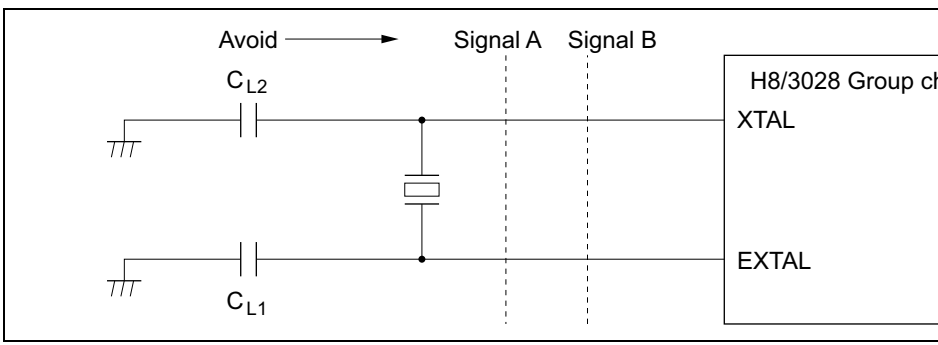


Figure 19.4 Oscillator Circuit Block Board Design Precautions

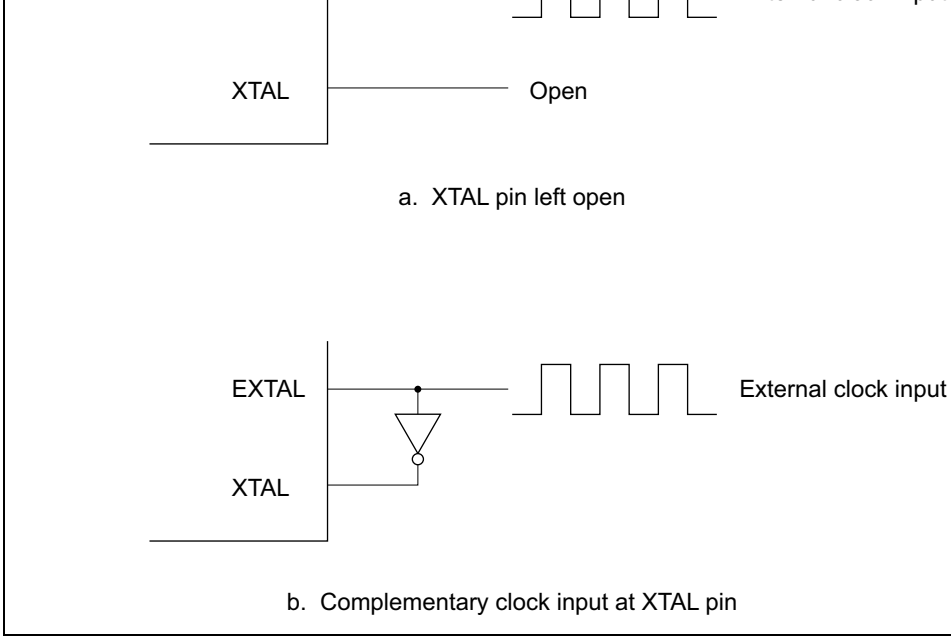


Figure 19.5 External Clock Input (Examples)

External Clock: The external clock frequency should be equal to the system clock frequ not divided by the on-chip frequency divider. Table 19.3 shows the clock timing, figure shows the external clock input timing, and figure 19.7 shows the external clock output delay timing. When the appropriate external clock is input via the EXTAL pin, its wav corrected by the on-chip oscillator and duty adjustment circuit.

When the appropriate external clock is input via the EXTAL pin, its waveform is corre on-chip oscillator and duty adjustment circuit. The resulting stable clock is output to ex devices after the external clock settling time (t_{DEXT}) has passed after the clock input. Th must remain reset with the reset signal low during t_{DEXT}, while the clock output is unsta

External clock rise time	t_{EXr}	—	5	ns	Figure 19
External clock fall time	t_{EXf}	—	5	ns	
Clock low pulse width	t_{CL}	0.4	0.6	t_{cyc}	$\phi \geq 5$ MHz
		80	—	ns	$\phi < 5$ MHz
Clock high pulse width	t_{CH}	0.4	0.6	t_{cyc}	$\phi \geq 5$ MHz
		80	—	ns	$\phi < 5$ MHz
External clock output settling delay time	t_{DEXT}*	500	—	µs	Figure 19

Note: * t_{DEXT} includes a $\overline{RES}$ pulse width (t_{RESW}). $t_{RESW} = 20\ t_{cyc}$

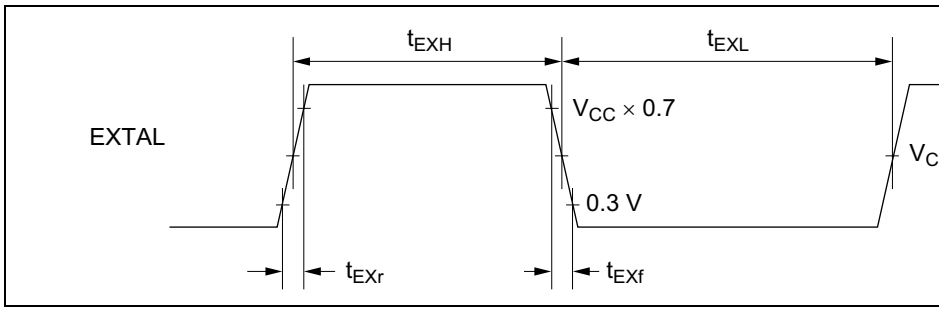


Figure 19.6 External Clock Input Timing

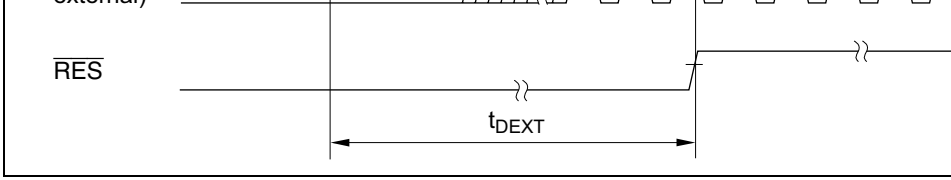


Figure 19.7 External Clock Output Settling Delay Timing

19.3 Duty Adjustment Circuit

When the oscillator frequency is 5 MHz or higher, the duty adjustment circuit adjusts th
cycle of the clock signal from the oscillator to generate ϕ.

19.4 Prescalers

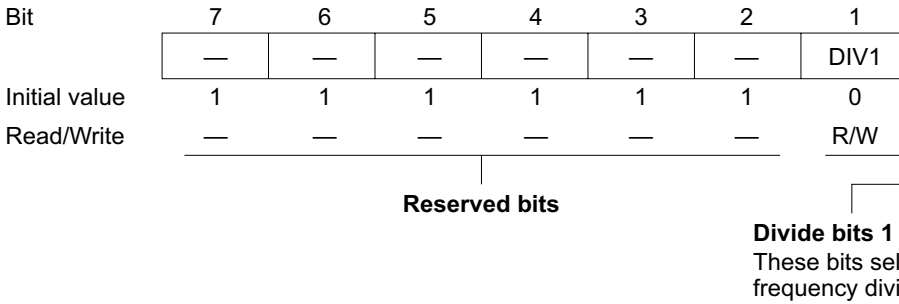
The prescalers divide the system clock (ϕ) to generate internal clocks (ϕ/2 to ϕ/4096).

19.5 Frequency Divider

The frequency divider divides the duty-adjusted clock signal to generate the system clo
frequency division ratio can be changed dynamically by modifying the value in DIVCR
described below. Power consumption in the chip is reduced in almost direct proportion
frequency division ratio. The system clock generated by the frequency divider can be o
ϕ pin.

19.5.2 Division Control Register (DIVCR)

DIVCR is an 8-bit readable/writable register that selects the division ratio of the frequ
divider.



DIVCR is initialized to H'FC by a reset and in hardware standby mode. It is not initial
software standby mode.

Bits 7 to 2—Reserved: These bits cannot be modified and are always read as 1.

Bits 1 and 0—Divide (DIV1, DIV0): These bits select the frequency division ratio, a

Bit 1 DIV1	Bit 0 DIV0	Frequency Division Ratio	
0	0	1/1	(I
0	1	1/2	
1	0	1/4	
1	1	1/8	

the division ratio. The waiting time for exit from software standby mode also chang
the division ratio is changed. For details, see section 20.4.3, Selection of Waiting Ti
from Software Standby Mode.

The power-down state includes the following three modes:

- Sleep mode
- Software standby mode
- Hardware standby mode

The module standby function can halt on-chip supporting modules independently of th
down state. The modules that can be halted are the 16-bit timer, 8-bit timer, SCI0, SCI
DMAC, DRAM interface, and A/D converter.

Table 20.1 indicates the methods of entering and exiting the power-down modes and r
standby mode, and gives the status of the CPU and on-chip supporting modules in eac

Mode	Entering Conditions	State													
		Clock	CPU	CPU Registers	DMAC	DRAM Interface	16-Bit Timer	8-Bit Timer	SCI0	SCI1	SCI2	A/D	Other Modules	RAM	φ clock output*4
Sleep mode	SLEEP instruction executed while SSBY = 0 in SYSCR	Active	Halted	Held	Active	Active	Active	Active	Active	Active	Active	Active	Active	Held	φ output
Software standby mode	SLEEP instruction executed while SSBY = 1 in SYSCR	Halted	Halted	Held	Halted and reset	Halted and held*1	Halted and reset	Halted and reset	Halted and reset	Halted and reset	Halted and reset	Halted and reset	Halted and reset	Held	High output
Hardware standby mode	Low input at $\overline{\text{STBY}}$ pin	Halted	Halted	Undetermined	Halted and reset	Halted and reset	Halted and reset	Halted and reset	Halted and reset	Halted and reset	Halted and reset	Halted and reset	Halted and reset	Held*3	High impedanc
Module standby	Corresponding bit set to 1 in MSTCR	Active	Active	—	Halted*2 and reset	Halted*2 and held*1	Halted*2 and reset	Halted*2 and reset	Halted*2 and reset	Halted*2 and reset	Halted*2 and reset	Halted*2 and reset	Active	—	High impedanc

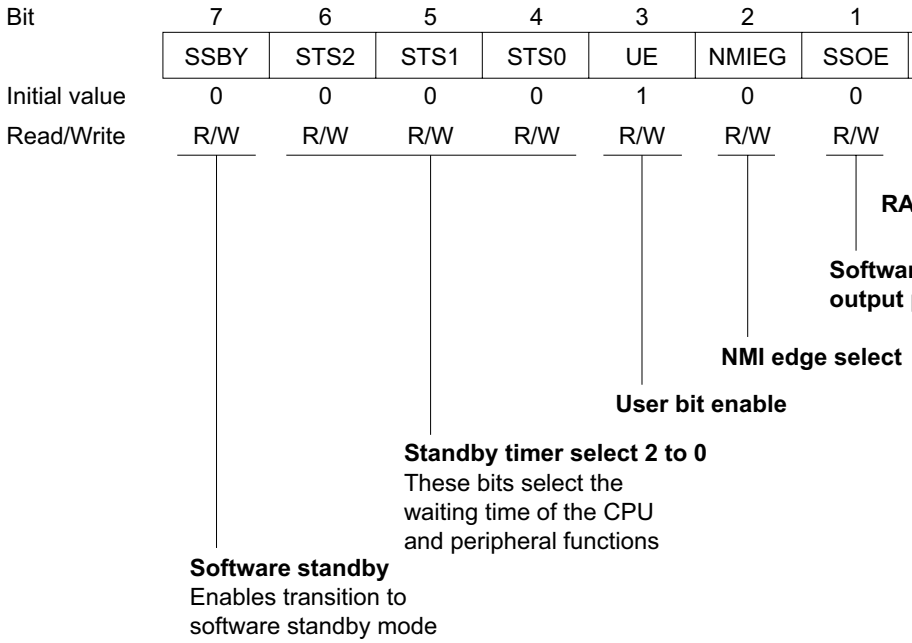
Notes: 1. RTCNT and bits 7 and 6 of RTMCSR are initialized. Other bits and registers hold their previous states.
2. State in which the corresponding MSTCR bit was set to 1. For details see section 20.2.2, Module Standby Control Register H (MS Module Standby Control Register L (MSTCRL).
3. The RAME bit must be cleared to 0 in SYSCR before the transition from the program execution state to hardware standby mode.
4. When $P6_7$ is used as the φ output pin.
5. When a MSTCR bit is set to 1, the registers of the corresponding on-chip supporting module are initialized. To restart the module, t then set up the module registers again.

Legend
SYSCR: System control register
SSBY: Software standby bit
MSTCRH: Module standby control register H
MSTCRL: Module standby control register L

H'EE012	System control register	SYSCR	R/W	H'09
H'EE01C	Module standby control register H	MSTCRH	R/W	H'78
H'EE01D	Module standby control register L	MSTCRL	R/W	H'00

Note: * Lower 20 bits of the address in advanced mode.

20.2.1 System Control Register (SYSCR)



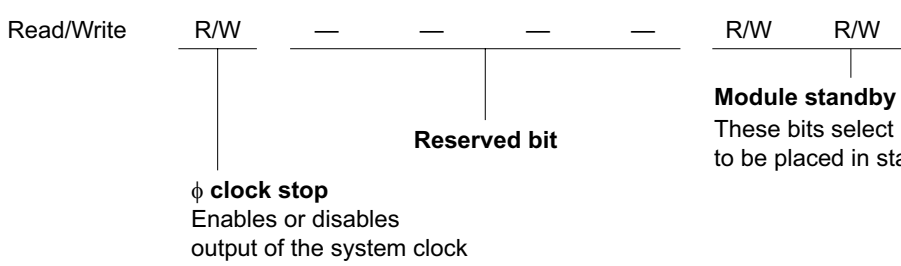
SYSCR is an 8-bit readable/writable register. Bit 7 (SSBY), bits 6 to 4 (STS2 to STS0 (SSOE) control the power-down state. For information on the other SYSCR bits, see s System Control Register (SYSCR).

Bits 6 to 4—Standby Timer Select (STS2 to STS0): These bits select the length of tir and on-chip supporting modules wait for the clock to settle when software standby mod by an external interrupt. If the clock is generated by a crystal resonator, set these bits ac the clock frequency so that the waiting time will be at least 7 ms (oscillation settling tir table 20.3. If an external clock is used, set these bits so that the waiting time will be at l 100 µs.

Bit 6 STS2	Bit 5 STS1	Bit 4 STS0	Description
0	0	0	Waiting time = 8,192 states (Ini
		1	Waiting time = 16,384 states
	1	0	Waiting time = 32,768 states
		1	Waiting time = 65,536 states
1	0	0	Waiting time = 131,072 states
		1	Waiting time = 262,144 states
	1	0	Waiting time = 1,024 states
		1	Illegal setting

Bit 1—Software Standby Output Port Enable (SSOE): Specifies whether the addres bus control signals ($\overline{CS}_0$ to $\overline{CS}_7$, $\overline{AS}$, $\overline{RD}$, $\overline{HWR}$, $\overline{LWR}$, $\overline{UCAS}$, $\overline{LCAS}$, and $\overline{RFSH}$) are l outputs or fixed high, or placed in the high-impedance state in software standby mode.

Bit 1 SSOE	Description
0	In software standby mode, the address bus and bus control signals are all l impedance (Ini
1	In software standby mode, the address bus retains its output state and bus signals are fixed high



MSTCRH is initialized to H'78 by a reset and in hardware standby mode. It is not initi
software standby mode.

Bit 7—ϕ Clock Stop (PSTOP): Enables or disables output of the system clock (ϕ).

Bit 1 PSTOP	Description	
0	System clock output is enabled	(In
1	System clock output is disabled	

Bits 6 to 3—Reserved: These bits cannot be modified and are always read as 1.

Bit 2—Module Standby H2 (MSTPH2): Selects whether to place the SCI2 in standb

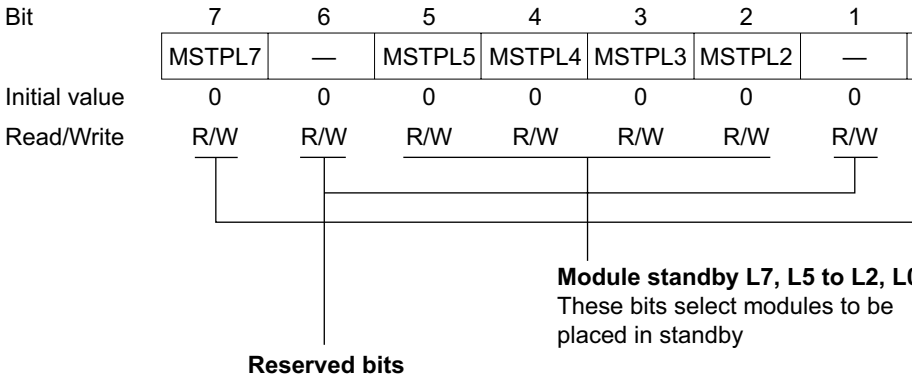
Bit 2 MSTPH2	Description	
0	SCI2 operates normally	(In
1	SCI2 is in standby state	

Bit 1—Module Standby H1 (MSTPH1): Selects whether to place the SCI1 in standb

0	SCI0 operates normally	(Ini
1	SCI0 is in standby state	

20.2.3 Module Standby Control Register L (MSTCRL)

MSTCRL is an 8-bit readable/writable register that controls the module standby functio places individual on-chip supporting modules in the standby state. Module standby can designated for the DMAC, 16-bit timer, DRAM interface, 8-bit timer, and A/D convert



MSTCRL is initialized to H'00 by a reset and in hardware standby mode. It is not initia software standby mode.

Bit 7—Module Standby L7 (MSTPL7): Selects whether to place the DMAC in stand

Bit 7 MSTPL7	Description	
0	DMAC operates normally	(Ini
1	DMAC is in standby state	

Bit 4—Module Standby L4 (MSTPL4): Selects whether to place the 16-bit timer in

Bit 4 MSTPL4	Description	
0	16-bit timer operates normally	(In
1	16-bit timer is in standby state	

Bit 3—Module Standby L3 (MSTPL3): Selects whether to place 8-bit timer channel
standby.

Bit 3 MSTPL3	Description	
0	8-bit timer channels 0 and 1 operate normally	(In
1	8-bit timer channels 0 and 1 are in standby state	

Bit 2—Module Standby L2 (MSTPL2): Selects whether to place 8-bit timer channel
standby.

Bit 2 MSTPL2	Description	
0	8-bit timer channels 2 and 3 operate normally	(In
1	8-bit timer channels 2 and 3 are in standby state	

Bit 1—Reserved: This bit can be written and read.

Bit 0—Module Standby L0 (MSTPL0): Selects whether to place the A/D converter

Bit 0 MSTPL0	Description	
0	A/D converter operates normally	(In
1	A/D converter is in standby state	

halted.

20.3.2 Exit from Sleep Mode

Sleep mode is exited by an interrupt, or by input at the $\overline{\text{RES}}$ or $\overline{\text{STBY}}$ pin.

Exit by Interrupt: An interrupt terminates sleep mode and causes a transition to the in exception handling state. Sleep mode is not exited by an interrupt source in an on-chip module if the interrupt is disabled in the on-chip supporting module. Sleep mode is not an interrupt other than NMI if the interrupt is masked by interrupt priority settings and of the I and UI bits in CCR, IPR.

Exit by $\overline{\text{RES}}$ Input: Low input at the $\overline{\text{RES}}$ pin exits from sleep mode to the reset state.

Exit by $\overline{\text{STBY}}$ Input: Low input at the $\overline{\text{STBY}}$ pin exits from sleep mode to hardware s mode.

are reset and halted. As long as the specified voltage is supplied, however, CPU regist and on-chip RAM data are retained. The settings of the I/O ports and DRAM interface held. When the WDT is used as a watchdog timer (WT/$\overline{IT}$ = 1), the TME bit must be c before setting SSBY. Also, when setting TME to 1, SSBY should be cleared to 0.

Clear the BRLE bit in BRCR (inhibiting bus release) before making a transition to sof standby mode.

Note: * RTCNT and bits 7 and 6 of RTMCSR are initialized. Other bits and registers r previous states.

20.4.2 Exit from Software Standby Mode

Software standby mode can be exited by input of an external interrupt at the NMI, $\overline{IRQ}$ $\overline{IRQ_2}$ pin, or by input at the $\overline{RES}$ or $\overline{STBY}$ pin.

Exit by Interrupt: When an NMI, IRQ_0, IRQ_1, or IRQ_2 interrupt request signal is rec clock oscillator begins operating. After the oscillator settling time selected by bits STS in SYSCR, stable clock signals are supplied to the entire chip, software standby mode interrupt exception handling begins. Software standby mode is not exited if the interru bits of interrupts IRQ_0, IRQ_1, and IRQ_2 are cleared to 0, or if these interrupts are mask CPU.

Exit by $\overline{RES}$ Input: When the $\overline{RES}$ input goes low, the clock oscillator starts and cloc supplied immediately to the entire chip. The $\overline{RES}$ signal must be held low long enough clock oscillator to stabilize. When $\overline{RES}$ goes high, the CPU starts reset exception hand

Exit by $\overline{STBY}$ Input: Low input at the $\overline{STBY}$ pin causes a transition to hardware stan

Table 20.3 Clock Frequency and Waiting Time for Clock to Settle

DIV1	DIV0	STS2	STS1	STS0	Waiting Time	25 MHz	20 MHz	18 MHz	16 MHz	12 MHz	10 MHz	8 MHz	6 MHz	4 MHz	2 M
0	0	0	0	0	8192 states	0.3	0.4	0.46	0.51	0.65	0.8	1.0	1.3	2.0	4.1
		0	0	1	16384 states	0.7	0.8	0.91	1.0	1.3	1.6	2.0	2.7	4.1	8.2
		0	1	0	32768 states	1.3	1.6	1.8	2.0	2.7	3.3	4.1	5.5	8.2*	16.
		0	1	1	65536 states	2.6	3.3	3.6	4.1	5.5	6.6	8.2*	10.9*	16.4	32.
		1	0	0	131072 states	5.2	6.6	7.3*	8.2*	10.9*	13.1*	16.4	21.8	32.8	65.
		1	0	1	262144 states	10.5*	13.1*	14.6	16.4	21.8	26.2	32.8	43.7	65.5	131
		1	1	0	1024 states	0.04	0.05	0.057	0.064	0.085	0.10	0.13	0.17	0.26	0.5
		1	1	1							Illegal setting				
0	1	0	0	0	8192 states	0.7	0.8	0.91	1.02	1.4	1.6	2.0	2.7	4.1	8.2
		0	0	1	16384 states	1.3	1.6	1.8	2.0	2.7	3.3	4.1	5.5	8.2*	16.
		0	1	0	32768 states	2.6	3.3	3.6	4.1	5.5	6.6	8.2*	10.9*	16.4	32.
		0	1	1	65536 states	5.2	6.6	7.3*	8.2*	10.9*	13.1*	16.4	21.8	32.8	65.
		1	0	0	131072 states	10.5*	13.1*	14.6	16.4	21.8	26.2	32.8	43.7	65.5	131
		1	0	1	262144 states	21.0	26.2	29.1	32.8	43.7	52.4	65.5	87.4	131.1	262
		1	1	0	1024 states	0.08	0.10	0.11	0.13	0.17	0.20	0.26	0.34	0.51	1.0
		1	1	1							Illegal setting				
1	0	0	0	0	8192 states	1.3	1.6	1.8	2.0	2.7	3.3	4.1	5.5	8.2*	16.
		0	0	1	16384 states	2.6	3.3	3.6	4.1	5.5	6.6	8.2*	10.9*	16.4	32.
		0	1	0	32768 states	5.2	6.6	7.3*	8.2*	10.9*	13.1*	16.4	21.8	32.8	65.
		0	1	1	65536 states	10.5*	13.1*	14.6	16.4	21.8	26.2	32.8	43.7	65.5	131
		1	0	0	131072 states	21.0	26.2	29.1	32.8	43.7	52.4	65.5	87.4	131.1	262
		1	0	1	262144 states	41.9	52.4	58.3	65.5	87.4	104.9	131.1	174.8	262.1	524
		1	1	0	1024 states	0.16	0.20	0.23	0.26	0.34	0.41	0.51	0.68	1.02	2.0
		1	1	1							Illegal setting				
1	1	0	0	0	8192 states	2.6	3.3	3.6	4.1	5.5	6.6	8.2*	10.9*	16.4*	32.
		0	0	1	16384 states	5.2	6.6	7.3*	8.2*	10.9*	13.1*	16.4	21.8	32.8	65.
		0	1	0	32768 states	10.5	13.1*	14.6	16.4	21.8	26.2	32.8	43.7	65.5	131
		0	1	1	65536 states	21.0*	26.2	29.1	32.8	43.7	52.4	65.5	87.4	131.1	262
		1	0	0	131072 states	41.9	52.4	58.3	65.5	87.4	104.9	131.1	174.8	262.1	524
		1	0	1	262144 states	83.9	104.9	116.5	131.1	174.8	209.7	262.1	349.5	524.3	104
		1	1	0	1024 states	0.33	0.41	0.46	0.51	0.68	0.82	1.0	1.4	2.0	4.1
		1	1	1							Illegal setting				

* : Recommended setting

Software standby mode is exited at the next rising edge of the NMI signal.

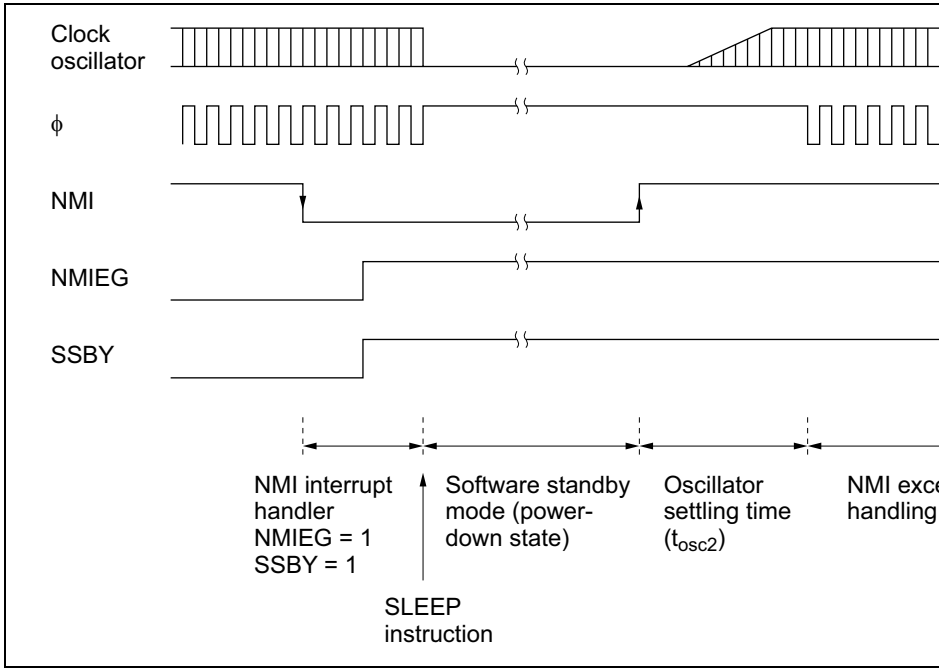


Figure 20.1 NMI Timing for Software Standby Mode (Example)

20.4.5 Note

The I/O ports retain their existing states in software standby mode. If a port is in the h state, its output current is not reduced.

Clear the RAME bit to 0 in SYSCR before $\overline{\text{STBY}}$ goes low to retain on-chip RAM data

The inputs at the mode pins (MD2 to MD0) should not be changed during hardware sta
mode.

20.5.2 Exit from Hardware Standby Mode

Hardware standby mode is exited by inputs at the $\overline{\text{STBY}}$ and $\overline{\text{RES}}$ pins. While $\overline{\text{RES}}$ is l
$\overline{\text{STBY}}$ goes high, the clock oscillator starts running. $\overline{\text{RES}}$ should be held low long enou
clock oscillator to settle. When $\overline{\text{RES}}$ goes high, reset exception handling begins, follow
transition to the program execution state.

20.5.3 Timing for Hardware Standby Mode

Figure 20.2 shows the timing relationships for hardware standby mode. To enter hardw
mode, first drive $\overline{\text{RES}}$ low, then drive $\overline{\text{STBY}}$ low. To exit hardware standby mode, first
$\overline{\text{STBY}}$ high, wait for the clock to settle, then bring $\overline{\text{RES}}$ from low to high.

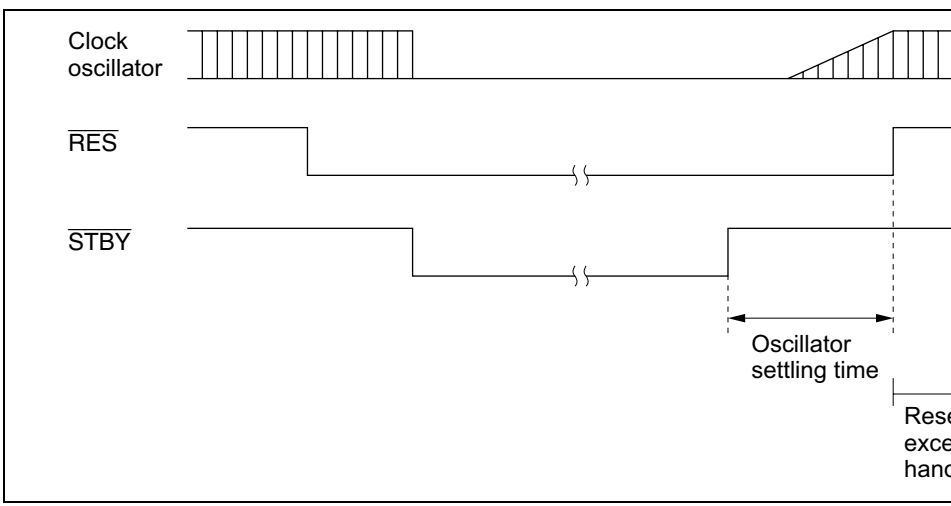


Figure 20.2 Hardware Standby Mode Timing

next bus cycle after the MSTCR write cycle.

20.6.2 Read/Write in Module Standby

When an on-chip supporting module is in module standby, read/write access to its reg
disabled. Read access always results in H'FF data. Write access is ignored.

20.6.3 Usage Notes

When using the module standby function, note the following points.

DMAC: When setting a bit in MSTCR to 1 to place the DMAC in module standby, m
the DMAC is not currently requesting the bus right. If the corresponding bit in MSTC
when a bus request is present, operation of the bus arbiter becomes ambiguous and a r
may occur.

DRAM Interface: When the module standby function is used on the DRAM interface
MSTCR bit to 1 while DRAM space is deselected.

On-Chip Supporting Module Interrupts: Before setting a module standby bit, first
interrupts by that module. When an on-chip supporting module is placed in standby by
standby function, its registers are initialized, including registers with interrupt request

Pin States: Pins used by an on-chip supporting module lose their module functions wi
module is placed in module standby. What happens after that depends on the particula
details, see section 8, I/O Ports. Pins that change from the input to the output state req
care. For example, if SCI1 is placed in module standby, the receive data pin loses its r
function and becomes a port pin. If its port DDR bit is set to 1, the pin becomes a data
and its output may collide with external SCI transmit data. Data collision should be pr
clearing the port DDR bit to 0 or taking other appropriate action.

Register Resetting: When an on-chip supporting module is halted by the module stan
function, all its registers are initialized. To restart the module, after its MSTCR bit is c
its registers must be set up again. It is not possible to write to the registers while the M
set to 1.

the state of the φ pin in various operating states.

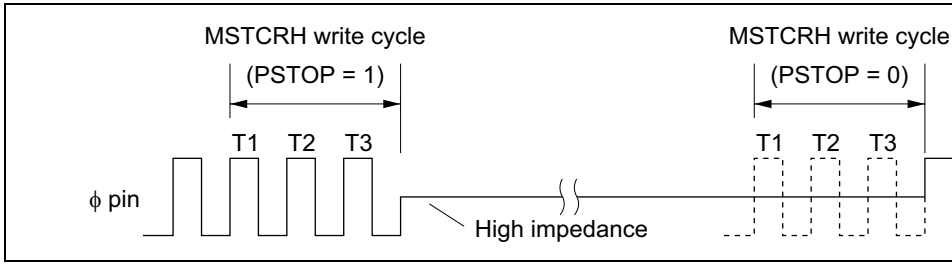


Figure 20.3 Starting and Stopping of System Clock Output

Table 20.4 φ Pin State in Various Operating States

Operating State	PSTOP = 0	PSTOP = 1
Hardware standby	High impedance	High impedance
Software standby	Always high	High impedance
Sleep mode	System clock output	High impedance
Normal operation	System clock output	High impedance

Table 21.1 Absolute Maximum Ratings

Item	Symbol	Value
Power supply voltage	V_{CC}	–0.3 to +4.6
Input voltage (except for port 7)*	V_{in}	–0.3 to V_{CC} +0.3
Input voltage (port 7)	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 +4.6
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 +8
Storage temperature	T_{stg}	–55 to +125

Caution: Permanent damage to the chip may result if absolute maximum ratings are ex

Note: * 12 V must not be applied to any pin, as this may cause permanent damage to

Item		Symbol	Min	Typ	Max	Unit	Test
Schmitt trigger input voltages	$P8_0$ to $P8_2$, Port A	V_T^-	$V_{CC} \times 0.2$	—	—	V	
		V_T^+	—	—	$V_{CC} \times 0.7$	V	
		$V_T^+ - V_T^-$	$V_{CC} \times 0.05$	—	—	V	
Input high voltage	$\overline{STBY}$, $\overline{RES}$, 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 7		$V_{CC} \times 0.7$	—	AV_{CC} + 0.3	V	
	Ports 1 to 6 $P8_3$, $P8_4$, $P9_0$ to $P9_5$, port B		$V_{CC} \times 0.7$	—	$V_{CC} + 0.3$	V	
Input low voltage	$\overline{STBY}$, $\overline{RES}$, MD_2 to MD_0	V_{IL}	–0.3	—	$V_{CC} \times 0.1$	V	
	NMI, EXTAL, ports 1 to 7 $P8_3$, $P8_4$, $P9_0$ to $P9_5$, port B		–0.3	—	$V_{CC} \times 0.2$	V	
Output high voltage	All output pins (except $\overline{RESO}$)	V_{OH}	V_{CC} – 0.5	—	—	V	I_{OH} =
			V_{CC} – 1.0	—	—	V	I_{OH} =
Output low voltage	All output pins (except $\overline{RESO}$)	V_{OL}	—	—	0.4	V	I_{OL} =
	Ports 1, 2, and 5		—	—	1.0	V	I_{OL} =
	$\overline{RESO}$		—	—	0.4	V	I_{OL} =
Input leakage current	$\overline{STBY}$, $\overline{RES}$, NMI, MD_2 to MD_0	$\|I_{in}\|$	—	—	1.0	µA	V_{in} = V_{CC}
	Port 7		—	—	1.0	µA	V_{in} = AV_{CC}

except NMI.

Current dissipation*2	Normal operation	I_{CC}*3	—	37 (3.3 V)	58	mA	f =
	Sleep mode		—	29 (3.3 V)	47	mA	
	Module standby mode		—	21 (3.3 V)	37	mA	
	Standby mode		—	1.0	10	µA	T_a ≤
			—	—	80	µA	50°
Analog power supply current	During A/D conversion	AI_{CC}	—	0.6	1.5	mA	
	During A/D and D/A conversion		—	0.6	1.5	mA	
	Idle		—	0.01	5.0	µA	DA
Reference current	During A/D conversion	AI_{CC}	—	0.45	0.8	mA	
	During A/D and D/A conversion		—	2.0	3.0	mA	
	Idle		—	0.01	5.0	µA	DA
RAM standby voltage		V_{RAM}	2.0	—	—	V	

Notes: 1. If the A/D converter is not used, do not leave the AV_{CC}, V_{REF}, and AV_{SS} pins
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 MOS pull-up transistors in the off stat
The values are for $V_{RAM} \leq V_{CC} < 3.0$ V, V_{IH} min = $V_{CC} \times 0.9$, and V_{IL} max = 0

3. I_{CC} max. (normal operation) = 6.0 (mA) + 0.577 (mA/(MHz × V)) × V_{CC}
I_{CC} max. (sleep mode) = 6.0 (mA) + 0.455 (mA/(MHz × V)) × V_{CC} × f
I_{CC} max. (sleep mode + module standby mode)
= 6.0 (mA) + 0.344 (mA/(MHz × V)) × V_{CC} × f
The Typ values for power consumption are reference values.

Permissible output low current (total)	Total of 20 pins in Ports 1, 2, and 5	ΣI_{OL}	—	—	8
	Total of all output pins, including the above		—	—	1
Permissible output high current (per pin)	All output pins	$\lvert -I_{OH} \rvert$	—	—	2
Permissible output high current (total)	Total of all output pins	$\lvert -\Sigma I_{OH} \rvert$	—	—	4

Notes: 1. To protect chip reliability, do not exceed the output current values in table 21
2. When directly driving a darlington pair or LED, always insert a current-limitin the output line, as shown in figures 21.1 and 21.2.

Figure 21.1 Darlington Pair Drive Circuit (Example)

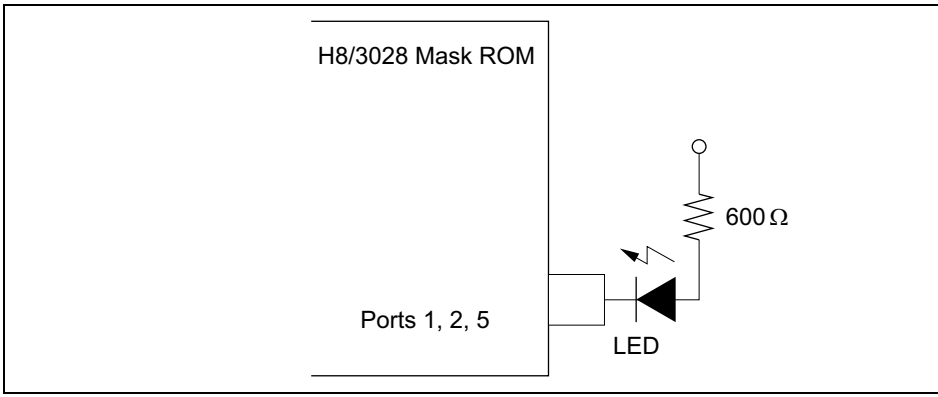


Figure 21.2 Sample LED Circuit

$V_{SS} = AV_{SS} = 0$ V, $T_a = -20°C$ to $+75°C$ (regular specifications),
$T_a = -40°C$ to $+85°C$ (wide-range specifications)

		Condition			
		f = 2 M to 25 MHz			
Item	**Symbol**	**Min**	**Max**	**Unit**	**Test C**
Clock cycle time	t_{cyc}	40	500	ns	Figure figure
Clock pulse low width	t_{CL}	10	—	ns	
Clock pulse high width	t_{CH}	10	—	ns	
Clock rise time	t_{Cr}	—	10	ns	
Clock fall time	t_{Cf}	—	10	ns	
Clock oscillator settling time at reset	t_{OSC1}	20	—	ms	Figure
Clock oscillator settling time in software standby	t_{OSC2}	7	—	ms	Figure

$\overline{\text{RES}}$ setup time	t_{RESS}	150	—	ns	Figure 2
$\overline{\text{RES}}$ pulse width	t_{RESW}	20	—	t_{cyc}	
Mode programming setup time	t_{MDS}	200	—	ns	
$\overline{\text{RESO}}$ output delay time	t_{RESO}	—	100	ns	Figure 2
$\overline{\text{RESO}}$ output pulse width	t_{RESOW}	132	—	t_{cyc}	
NMI, $\overline{\text{IRQ}}$ setup time	t_{NMIS}	150	—	ns	Figure 2
NMI, $\overline{\text{IRQ}}$ hold time	t_{NMIH}	10	—	ns	
NMI, $\overline{\text{IRQ}}$ pulse width	t_{NMIW}	200	—	ns	

Address delay time	t_{AD}	—	25	ns	Figu
Address hold time	t_{AH}	0.5 t_{cyc} – 20	—	ns	figur
Read strobe delay time	t_{RSD}	—	25	ns	figur
Address strobe delay time	t_{ASD}	—	25	ns	figur
Write strobe delay time	t_{WSD}	—	25	ns	
Strobe delay time	t_{SD}	—	25	ns	
Write strobe pulse width 1	t_{WSW1}	1.0 t_{cyc} – 25	—	ns	
Write strobe pulse width 2	t_{WSW2}	1.5 t_{cyc} – 25	—	ns	
Address setup time 1	t_{AS1}	0.5 t_{cyc} – 20	—	ns	
Address setup time 2	t_{AS2}	1.0 t_{cyc} – 20	—	ns	
Read data setup time	t_{RDS}	25	—	ns	
Read data hold time	t_{RDH}	0	—	ns	
Write data delay time	t_{WDD}	—	35	ns	
Write data setup time 1	t_{WDS1}	1.0 t_{cyc} – 30	—	ns	
Write data setup time 2	t_{WDS2}	2.0 t_{cyc} – 30	—	ns	
Write data hold time	t_{WDH}	0.5 t_{cyc} – 15	—	ns	
Read data access time 1	t_{ACC1}	—	2.0 t_{cyc} – 45	ns	
Read data access time 2	t_{ACC2}	—	3.0 t_{cyc} – 45	ns	
Read data access time 3	t_{ACC3}	—	1.5 t_{cyc} – 45	ns	
Read data access time 4	t_{ACC4}	—	2.5 t_{cyc} – 45	ns	
Precharge time 1	t_{PCH1}	1.0 t_{cyc} – 20	—	ns	
Precharge time 2	t_{PCH2}	0.5 t_{cyc} – 20	—	ns	
Wait setup time	t_{WTS}	25	—	ns	Figu
Wait hold time	t_{WTH}	5	—	ns	

Bus-floating time	t_{BZD}	—	30	ns	
$\overline{RAS}$ precharge time	t_{RP}	1.5 t_{cyc} – 25	—	ns	Figu figu
$\overline{CAS}$ precharge time	t_{CP}	0.5 t_{cyc} – 15	—	ns	
Row address hold time	t_{RAH}	0.5 t_{cyc} – 15	—	ns	
$\overline{RAS}$ delay time 1	t_{RAD1}	—	25	ns	
$\overline{RAS}$ delay time 2	t_{RAD2}	—	30	ns	
$\overline{CAS}$ delay time 1	t_{CASD1}	—	25	ns	
$\overline{CAS}$ delay time 2	t_{CASD2}	—	25	ns	
$\overline{WE}$ delay time	t_{WCD}	—	25	ns	
$\overline{CAS}$ pulse width 1	t_{CAS1}	1.5 t_{cyc} – 20	—	ns	
$\overline{CAS}$ pulse width 2	t_{CAS2}	1.0 t_{cyc} – 20	—	ns	
$\overline{CAS}$ pulse width 3	t_{CAS3}	1.0 t_{cyc} – 20	—	ns	
$\overline{RAS}$ access time	t_{RAC}	—	2.5 t_{cyc} – 40	ns	
Address access time	t_{AA}	—	2.0 t_{cyc} – 50	ns	
$\overline{CAS}$ access time	t_{CAC}	—	1.5 t_{cyc} – 50	ns	
$\overline{WE}$ setup time	t_{WCS}	0.5 t_{cyc} – 20	—	ns	
$\overline{WE}$ hold time	t_{WCH}	0.5 t_{cyc} – 15	—	ns	
Write data setup time	t_{WDS}	0.5 t_{cyc} – 20	—	ns	
$\overline{WE}$ write data hold time	t_{WDH}	0.5 t_{cyc} – 15	—	ns	
$\overline{CAS}$ setup time 1	t_{CSR1}	0.5 t_{cyc} – 20	—	ns	
$\overline{CAS}$ setup time 2	t_{CSR2}	0.5 t_{cyc} – 15	—	ns	
$\overline{CAS}$ hold time	t_{CHR}	0.5 t_{cyc} – 15	—	ns	
$\overline{RAS}$ pulse width	t_{RAS}	1.5 t_{cyc} – 15	—	ns	
Signal rise time (all input pins except EXTAL)	t_{SR}	—	100	ns	Figu
Signal fall time (all input pins except EXTAL)	t_{SF}	—	100	ns	

Note: In order to secure the address hold time relative to the rise of the $\overline{RD}$ strobe, ac update mode 2 should be used. For details see section 6.3.5, Address Output M

Ports and TPC	Output data delay time		t_{PWD}	—	50	ns	Figur
	Input data setup time		t_{PRS}	50	—	ns	
	Input data hold time		t_{PRH}	50	—	ns	
16-bit timer	Timer output delay time		t_{TOCD}	—	50	ns	Figur
	Timer input setup time		t_{TICS}	50	—	ns	
	Timer clock input setup time		t_{TCKS}	50	—	ns	Figur
	Timer clock pulse width	Single edge	t_{TCKWH}	1.5	—	t_{cyc}	
		Both edges	t_{TCKWL}	2.5	—	t_{cyc}	
8-bit timer	Timer output delay time		t_{TOCD}	—	50	ns	Figur
	Timer input setup time		t_{TICS}	50	—	ns	
	Timer clock input setup time		t_{TCKS}	50	—	ns	Figur
	Timer clock pulse width	Single edge	t_{TCKWH}	1.5	—	t_{cyc}	
		Both edges	t_{TCKWL}	2.5	—	t_{cyc}	

	Input clock fall time		t_{SCKf}	—	1.5	t_{cyc}	
	Input clock pulse width		t_{SCKW}	0.4	0.6	t_{Scyc}	
	Transmit data delay time		t_{TXD}	—	100	ns	Figu
	Receive data setup time (synchronous)		t_{RXS}	100	—	ns	
	Receive data hold time (synchronous)	Clock input	t_{RXH}	100	—	ns	
		Clock output		0	—	ns	
DMAC	$\overline{\text{TEND}}$ delay time 1		t_{TED1}	—	50	ns	Figu figu
	$\overline{\text{TEND}}$ delay time 2		t_{TED2}	—	50	ns	
	$\overline{\text{DREQ}}$ setup time		t_{DRQS}	25	—	ns	Figu
	$\overline{\text{DREQ}}$ hold time		t_{DRQH}	10	—	ns	

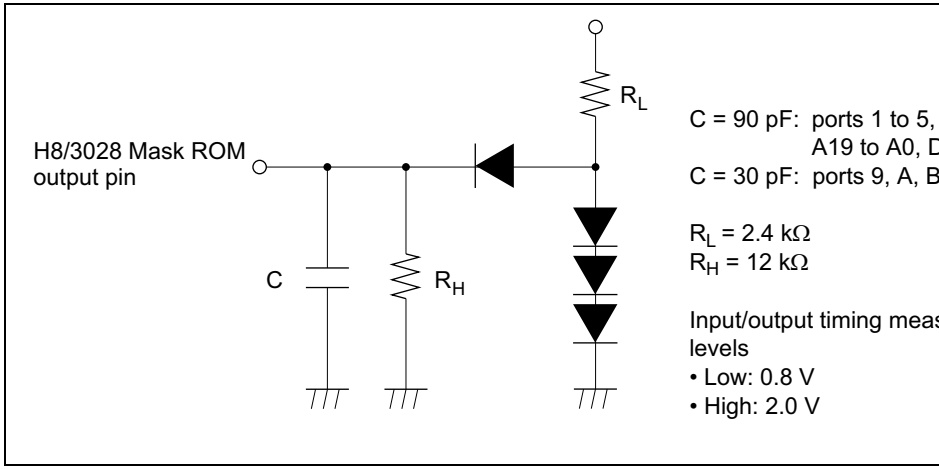


Figure 21.3 Output Load Circuit

Item			Condition: f = 2 M to 25 MHz Min	Typ	Max
Conversion time: 134 states	Resolution		10	10	10
	Conversion time (single mode)		5.36	—	—
	Analog input capacitance		—	—	20
	Permissible signal-source impedance	$\phi \leq 13$ MHz	—	—	10
		$\phi > 13$ MHz	—	—	5
	Nonlinearity error		—	—	±3.5
	Offset error		—	—	±3.5
	Full-scale error		—	—	±3.5
	Quantization error		—	—	±0.5
	Absolute accuracy		—	—	±4.0
Conversion time: 70 states*	Resolution		10	10	10
	Conversion time (single mode)		5.38	—	—
	Analog input capacitance		—	—	20
	Permissible signal-source impedance	$\phi \leq 13$ MHz	—	—	5
	Nonlinearity error		—	—	±7.5
	Offset error		—	—	±7.5
	Full-scale error		—	—	±7.5
	Quantization error		—	—	±0.5
	Absolute accuracy		—	—	±8.0

Note: * Do not select a conversion time of 70 states if the operating frequency exceeds (states)/5.38 (µs) = 13.0 (MHz).

	Condition				
	f = 2 M to 25 MHz				
Item	**Min**	**Typ**	**Max**	**Unit**	**Test Co**
Resolution	8	8	8	bits	
Conversion time (centering time)	—	—	10	µs	20 pF ca
Absolute accuracy	—	±2.0	±3.0	LSB	2 MΩ re
	—	—	±2.0	LSB	4 MΩ re

Power supply voltage	V_{CC}	–0.3 to +4.6
Input voltage (FWE)*1	V_{in}	–0.3 to V_{CC} +0.3
Input voltage (except for port 7)*1	V_{in}	–0.3 to V_{CC} +0.3
Input voltage (port 7)	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 +4.6
Analog input voltage	V_{AN}	–0.3 to AV_{CC} +0.3
Operating temperature	T_{opr}	Regular specifications: –20 to +75*2
		Wide-range specifications: –40 to +85
Storage temperature	T_{stg}	–55 to +125

Caution: Permanent damage to the chip may result if absolute maximum ratings are exc

Notes: 1. 12 V must not be applied to any pin, as this may cause permanent damage t device.

2. The operating temperature range when programming and erasing the flash m T_a = 0 to +75°C (regular specifications), T_a = 0 to +85°C (wide-range specific

[T_a = 0°C to +85°C (wide-range specifications)]

Item		Symbol	Min	Typ	Max	Unit	Tes
Schmitt trigger input voltages	$P8_0$ to $P8_2$, Port A	V_T^-	$V_{CC} \times 0.2$	—	—	V	
		V_T^+	—	—	$V_{CC} \times 0.7$	V	
		$V_T^+ - V_T^-$	$V_{CC} \times 0.05$	—	—	V	
Input high voltage	$\overline{STBY}$, $\overline{RES}$, NMI, MD_2 to MD_0, FWE	V_{IH}	$V_{CC} \times 0.9$	—	V_{CC} + 0.3	V	
	EXTAL		$V_{CC} \times 0.7$	—	V_{CC} + 0.3	V	
	Port 7		$V_{CC} \times 0.7$	—	AV_{CC} + 0.3	V	
	Ports 1 to 6 $P8_3$, $P8_4$, $P9_0$ to $P9_5$, port B		$V_{CC} \times 0.7$	—	V_{CC} + 0.3	V	
Input low voltage	$\overline{STBY}$, $\overline{RES}$, FWE, MD_2 to MD_0	V_{IL}	–0.3	—	$V_{CC} \times 0.1$	V	
	NMI, EXTAL, ports 1 to 7 $P8_3$, $P8_4$, $P9_0$ to $P9_5$, port B		–0.3	—	$V_{CC} \times 0.2$	V	
Output high voltage	All output pins	V_{OH}	V_{CC} – 0.5	—	—	V	I_{OH}
			V_{CC} – 1.0	—	—	V	I_{OH}
Output low voltage	All output pins	V_{OL}	—	—	0.4	V	I_{OL}
	Ports 1, 2, and 5		—	—	1.0	V	I_{OL}

Input pull-up MOS current	Ports 2, 4, and 5	$-I_p$	10	—	300	µA	V_{in} =
Input capacitance	FWE	C_{in}	—	—	80	pF	Vin =
	NMI		—	—	50	pF	f = f_r
	All input pins except NMI, and FWE		—	—	15	pF	Ta =
Current dissipation*2	Normal operation	I_{CC}*3	—	37 (3.3 V)	58	mA	f = 2
	Sleep mode		—	29 (3.3 V)	47		
	Module standby mode		—	21 (3.3 V)	37		
	Standby mode		—	1.0	10	µA	$T_a \leq$
			—	—	80	µA	50°C
	Flash memory programming/ erasing*4		—	47	68	mA	f = 2
Analog power supply current	During A/D conversion	AI_{CC}	—	0.6	1.5	mA	
	During A/D and D/A conversion		—	0.6	1.5	mA	
	Idle		—	0.01	5.0	µA	DAS
Reference current	During A/D conversion	AI_{CC}	—	0.45	0.8	mA	
	During A/D and D/A conversion		—	2.0	3.0	mA	
	Idle		—	0.01	5.0	µA	DAS
RAM standby voltage		V_{RAM}	2.0	—	—	V	

Notes: 1. If the A/D converter is not used, do not leave the AV_{CC}, V_{REF}, and AV_{SS} pins
Connect AV_{CC} and V_{REF} to V_{CC}, and connect AV_{SS} to V_{SS}.

Table 21.12 Permissible Output Currents

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{REF} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

Item		Symbol	Min	Typ
Permissible output low current (per pin)	Ports 1, 2, and 5	I_{OL}	—	—
	Other output pins		—	—
Permissible output low current (total)	Total of 20 pins in Ports 1, 2, and 5	ΣI_{OL}	—	—
	Total of all output pins, including the above		—	—
Permissible output high current (per pin)	All output pins	$\lvert -I_{OH} \rvert$	—	—
Permissible output high current (total)	Total of all output pins	$\lvert -\Sigma I_{OH} \rvert$	—	—

Notes: 1. To protect chip reliability, do not exceed the output current values in table 2
2. When directly driving a darlington pair or LED, always insert a current-limiti the output line, as shown in figures 21.4 and 21.5.

Figure 21.4 Darlington Pair Drive Circuit (Example)

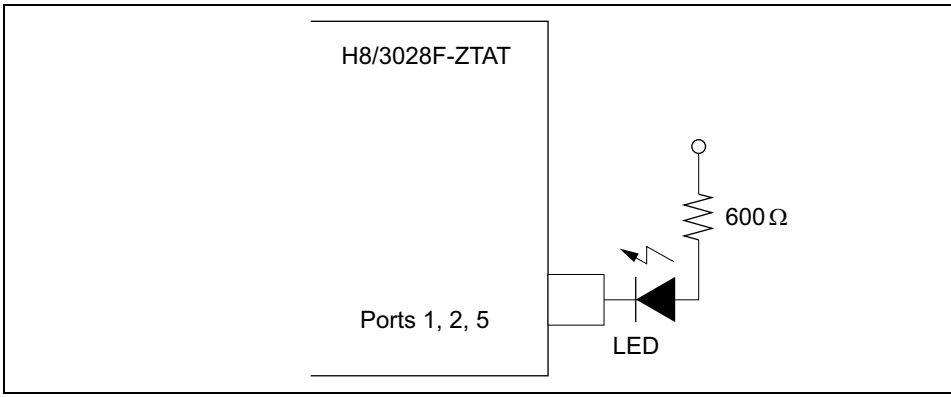


Figure 21.5 Sample LED Circuit

T_a = –40°C to +85°C (wide-range specifications)

Item	Symbol	Condition: f = 2 M to 25 MHz Min	Max	Unit	Test
Clock cycle time	t_{cyc}	40	500	ns	Figu
Clock pulse low width	t_{CL}	10	—	ns	figur
Clock pulse high width	t_{CH}	10	—	ns	
Clock rise time	t_{Cr}	—	10	ns	
Clock fall time	t_{Cf}	—	10	ns	
Clock oscillator settling time at reset	t_{OSC1}	20	—	ms	Figu
Clock oscillator settling time in software standby	t_{OSC2}	7	—	ms	Figu

Table 21.14 Control Signal Timing

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{REF} = 3.0 V to AV_{CC},
V_{SS} = AV_{SS} = 0 V, T_a = –20°C to +75°C (regular specifications),
T_a = –40°C to +85°C (wide-range specifications)

Item	Symbol	Condition: f = 2 M to 25 MHz Min	Max	Unit	Test Co
$\overline{RES}$ setup time	t_{RESS}	150	—	ns	Figure 2
$\overline{RES}$ pulse width	t_{RESW}	20	—	t_{cyc}	
Mode programming setup time	t_{MDS}	200	—	ns	
NMI, $\overline{IRQ}$ setup time	t_{NMIS}	150	—	ns	Figure 2
NMI, $\overline{IRQ}$ hold time	t_{NMIH}	10	—	ns	
NMI, $\overline{IRQ}$ pulse width	t_{NMIW}	200	—	ns	

Address delay time	t_{AD}	—	25	ns	Figu
Address hold time	t_{AH}	0.5 t_{cyc} – 20	—	ns	figur
Read strobe delay time	t_{RSD}	—	25	ns	figur
Address strobe delay time	t_{ASD}	—	25	ns	figur
Write strobe delay time	t_{WSD}	—	25	ns	
Strobe delay time	t_{SD}	—	25	ns	
Write strobe pulse width 1	t_{WSW1}	1.0 t_{cyc} – 25	—	ns	
Write strobe pulse width 2	t_{WSW2}	1.5 t_{cyc} – 25	—	ns	
Address setup time 1	t_{AS1}	0.5 t_{cyc} – 20	—	ns	
Address setup time 2	t_{AS2}	1.0 t_{cyc} – 20	—	ns	
Read data setup time	t_{RDS}	25	—	ns	
Read data hold time	t_{RDH}	0	—	ns	
Write data delay time	t_{WDD}	—	35	ns	
Write data setup time 1	t_{WDS1}	1.0 t_{cyc} – 30	—	ns	
Write data setup time 2	t_{WDS2}	2.0 t_{cyc} – 30	—	ns	
Write data hold time	t_{WDH}	0.5 t_{cyc} – 15	—	ns	
Read data access time 1	t_{ACC1}	—	2.0 t_{cyc} – 45	ns	
Read data access time 2	t_{ACC2}	—	3.0 t_{cyc} – 45	ns	
Read data access time 3	t_{ACC3}	—	1.5 t_{cyc} – 45	ns	
Read data access time 4	t_{ACC4}	—	2.5 t_{cyc} – 45	ns	
Precharge time 1	t_{PCH1}	1.0 t_{cyc} – 20	—	ns	
Precharge time 2	t_{PCH2}	0.5 t_{cyc} – 20	—	ns	
Wait setup time	t_{WTS}	25	—	ns	Figu
Wait hold time	t_{WTH}	5	—	ns	

Bus-floating time	t_{BZD}	—	30	ns	
$\overline{RAS}$ precharge time	t_{RP}	1.5 t_{cyc} – 25	—	ns	Figu figu
$\overline{CAS}$ precharge time	t_{CP}	0.5 t_{cyc} – 15	—	ns	
Row address hold time	t_{RAH}	0.5 t_{cyc} – 15	—	ns	
$\overline{RAS}$ delay time 1	t_{RAD1}	—	25	ns	
$\overline{RAS}$ delay time 2	t_{RAD2}	—	30	ns	
$\overline{CAS}$ delay time 1	t_{CASD1}	—	25	ns	
$\overline{CAS}$ delay time 2	t_{CASD2}	—	25	ns	
$\overline{WE}$ delay time	t_{WCD}	—	25	ns	
$\overline{CAS}$ pulse width 1	t_{CAS1}	1.5 t_{cyc} – 20	—	ns	
$\overline{CAS}$ pulse width 2	t_{CAS2}	1.0 t_{cyc} – 20	—	ns	
$\overline{CAS}$ pulse width 3	t_{CAS3}	1.0 t_{cyc} – 20	—	ns	
$\overline{RAS}$ access time	t_{RAC}	—	2.5 t_{cyc} – 40	ns	
Address access time	t_{AA}	—	2.0 t_{cyc} – 50	ns	
$\overline{CAS}$ access time	t_{CAC}	—	1.5 t_{cyc} – 50	ns	
$\overline{WE}$ setup time	t_{WCS}	0.5 t_{cyc} – 20	—	ns	
$\overline{WE}$ hold time	t_{WCH}	0.5 t_{cyc} – 15	—	ns	
Write data setup time	t_{WDS}	0.5 t_{cyc} – 20	—	ns	
$\overline{WE}$ write data hold time	t_{WDH}	0.5 t_{cyc} – 15	—	ns	
$\overline{CAS}$ setup time 1	t_{CSR1}	0.5 t_{cyc} – 20	—	ns	
$\overline{CAS}$ setup time 2	t_{CSR2}	0.5 t_{cyc} – 15	—	ns	
$\overline{CAS}$ hold time	t_{CHR}	0.5 t_{cyc} – 15	—	ns	
$\overline{RAS}$ pulse width	t_{RAS}	1.5 t_{cyc} – 15	—	ns	
Signal rise time (all input pins except EXTAL)	t_{SR}	—	100	ns	Figu
Signal fall time (all input pins except EXTAL)	t_{SF}	—	100	ns	

Note: In order to secure the address hold time relative to the rise of the $\overline{RD}$ strobe, ac update mode 2 should be used. For details see section 6.3.5, Address Output M

Ports and TPC	Output data delay time		t_{PWD}	—	50	ns	Figur
	Input data setup time		t_{PRS}	50	—	ns	
	Input data hold time		t_{PRH}	50	—	ns	
16-bit timer	Timer output delay time		t_{TOCD}	—	50	ns	Figur
	Timer input setup time		t_{TICS}	50	—	ns	
	Timer clock input setup time		t_{TCKS}	50	—	ns	Figur
	Timer clock pulse width	Single edge	t_{TCKWH}	1.5	—	t_{cyc}	
		Both edges	t_{TCKWL}	2.5	—	t_{cyc}	
8-bit timer	Timer output delay time		t_{TOCD}	—	50	ns	Figur
	Timer input setup time		t_{TICS}	50	—	ns	
	Timer clock input setup time		t_{TCKS}	50	—	ns	Figur
	Timer clock pulse width	Single edge	t_{TCKWH}	1.5	—	t_{cyc}	
		Both edges	t_{TCKWL}	2.5	—	t_{cyc}	

	Input clock fall time		t_{SCKf}	—	1.5	t_{cyc}	
	Input clock pulse width		t_{SCKW}	0.4	0.6	t_{Scyc}	
	Transmit data delay time		t_{TXD}	—	100	ns	Figu
	Receive data setup time (synchronous)		t_{RXS}	100	—	ns	
	Receive data hold time (synchronous)	Clock input	t_{RXH}	100	—	ns	
		Clock output		0	—	ns	
DMAC	$\overline{\text{TEND}}$ delay time 1		t_{TED1}	—	50	ns	Figu figu
	$\overline{\text{TEND}}$ delay time 2		t_{TED2}	—	50	ns	
	$\overline{\text{DREQ}}$ setup time		t_{DRQS}	25	—	ns	Figu
	$\overline{\text{DREQ}}$ hold time		t_{DRQH}	10	—	ns	

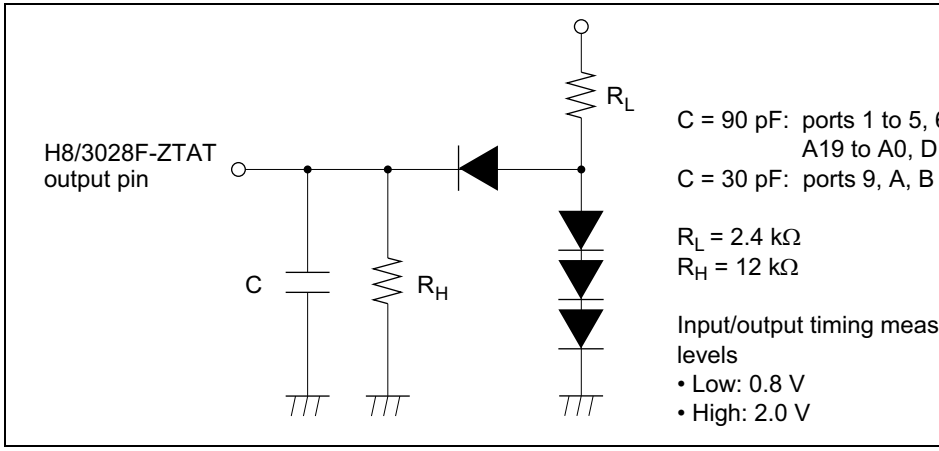


Figure 21.6 Output Load Circuit

Item			Condition: f = 2 M to 25 MHz Min	Typ	Max
Conversion time: 134 states	Resolution		10	10	10
	Conversion time (single mode)		5.36	—	—
	Analog input capacitance		—	—	20
	Permissible signal-source impedance	$\phi \leq 13$ MHz	—	—	10
		$\phi > 13$ MHz	—	—	5
	Nonlinearity error		—	—	±3.5
	Offset error		—	—	±3.5
	Full-scale error		—	—	±3.5
	Quantization error		—	—	±0.5
	Absolute accuracy		—	—	±4.0
Conversion time: 70 states*	Resolution		10	10	10
	Conversion time (single mode)		5.38	—	—
	Analog input capacitance		—	—	20
	Permissible signal-source impedance	$\phi \leq 13$ MHz	—	—	5
	Nonlinearity error		—	—	±7.5
	Offset error		—	—	±7.5
	Full-scale error		—	—	±7.5
	Quantization error		—	—	±0.5
	Absolute accuracy		—	—	±8.0

Note: * Do not select a conversion time of 70 states if the operating frequency exceeds (states)/5.38 (µs) = 13.0 (MHz).

	Condition				
	f = 2 M to 25 MHz				
Item	**Min**	**Typ**	**Max**	**Unit**	**Test Co**
Resolution	8	8	8	bits	
Conversion time (centering time)	—	—	10	µs	20 pF ca
Absolute accuracy	—	±2.0	±3.0	LSB	2 MΩ re
	—	—	±2.0	LSB	4 MΩ re

specification)

Item		Symbol	Min	Typ	Max	Unit	N
Programming time*1 *2 *4		t_P	—	10	200	ms/ 128 bytes	
Erase time*1 *3 *5		t_E	—	100	1200	ms/block	
Reprogramming count		N_{WEC}	—	—	100	Times	
Programming	Wait time after SWE bit setting*1	t_{sswe}	1	1	—	µs	
	Wait time after PSU bit setting*1	t_{spsu}	50	50	—	µs	
	Wait time after P bit setting*1 *4	t_{sp30}	28	30	32	µs	P t
		t_{sp200}	198	200	202	µs	P t
		t_{sp10}	8	10	12	µs	A p t
	Wait time after P bit clear*1	t_{cp}	5	5	—	µs	
	Wait time after PSU bit clear*1	t_{cpsu}	5	5	—	µs	
	Wait time after PV bit setting*1	t_{spv}	4	4	—	µs	
	Wait time after H'FF dummy write*1	t_{spvr}	2	2	—	µs	
	Wait time after PV bit clear*1	t_{cpv}	2	2	—	µs	
	Wait time after SWE bit clear*1	t_{cswe}	100	100	—	µs	
	Maximum programming count*1 *4	N	—	—	1000	Times	
Erase	Wait time after SWE bit setting*1	t_{sswe}	1	1	—	µs	
	Wait time after ESU bit setting*1	t_{sesu}	100	100	—	µs	
	Wait time after E bit setting*1 *5	t_{se}	10	10	100	ms	E w
	Wait time after E bit clear*1	t_{ce}	10	10	—	µs	
	Wait time after ESU bit clear*1	t_{cesu}	10	10	—	µs	
	Wait time after EV bit setting*1	t_{sev}	20	20	—	µs	
	Wait time after H'FF dummy write*1	t_{sevr}	2	2	—	µs	
	Wait time after EV bit clear*1	t_{cev}	4	4	—	µs	
	Wait time after SWE bit clear*1	t_{cswe}	100	100	—	µs	
	Maximum erase count*1 *5	N	12	—	120	Times	

the programming counter (n).

Programming counter (n) = 1 to 6: t_{sp30} =

Programming counter (n) = 7 to 1000: t_{sp200} =

Programming counter (n) [in additional programming] = 1 to 6: t_{sp10} =

5. For the maximum erase time (t_E(max)), the following relationship applies be wait time after E bit setting (t_{se}) and the maximum erase count (N):

   t_E(max) = Wait time after E bit setting (t_{se}) × maximum erase count (N

   To set the maximum erase time, the values of t_{se} and N should be set so as the above formula.

   Examples: When t_{se} = 100 [ms], N = 12 times

   When t_{se} = 10 [ms], N = 120 times

Figure 21.7 shows the oscillator settling timing.

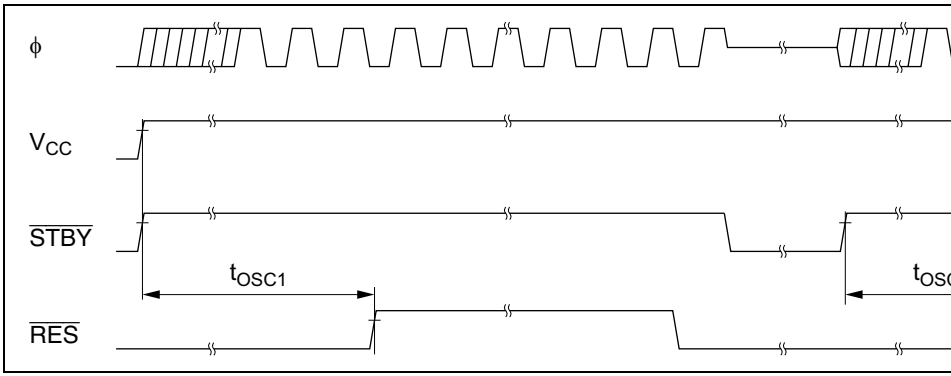


Figure 21.7 Oscillator Settling Timing

Figure 21.10 shows the interrupt input timing for NMI and $\overline{IRQ_5}$ to $\overline{IRQ_0}$.

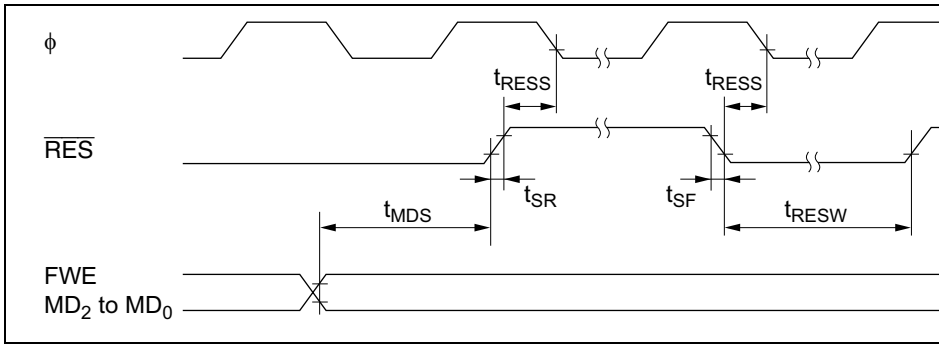


Figure 21.8 Reset Input Timing

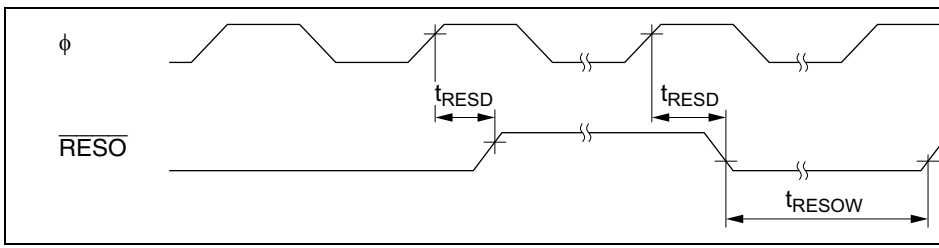


Figure 21.9 Reset Output Timing*

Note: * This function is used only in mask ROM models, and is not provided in flash models.

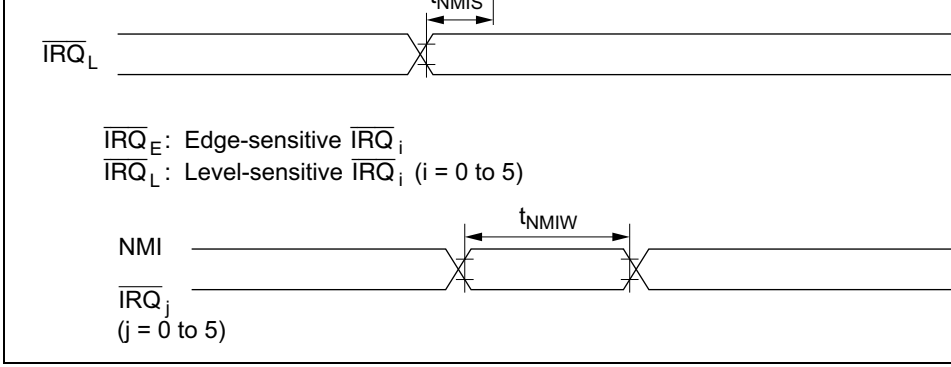
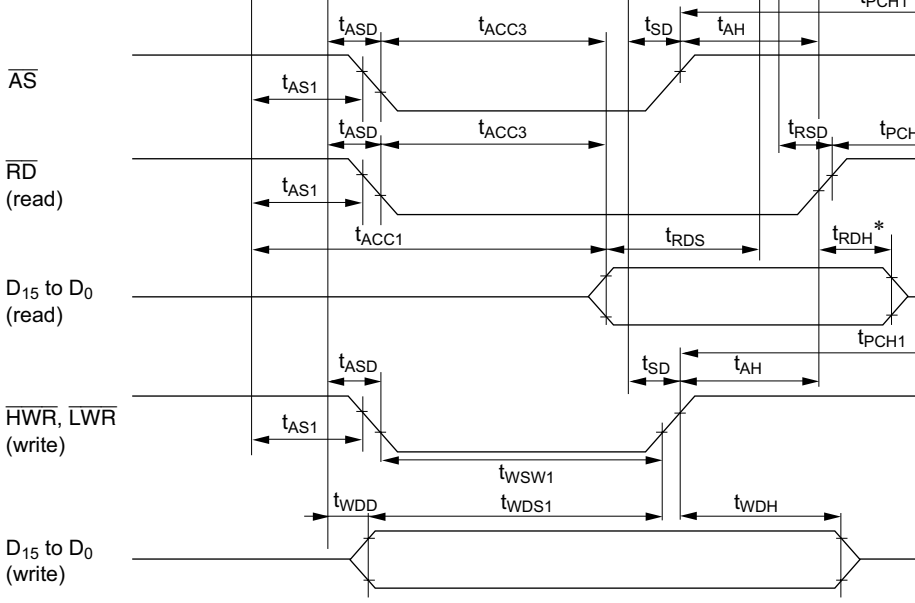


Figure 21.10 Interrupt Input Timing

21.3.3 Bus Timing

Bus timing is shown as follows:

- Basic bus cycle: two-state access
  Figure 21.11 shows the timing of the external two-state access cycle.
- Basic bus cycle: three-state access
  Figure 21.12 shows the timing of the external three-state access cycle.
- Basic bus cycle: three-state access with one wait state
  Figure 21.13 shows the timing of the external three-state access cycle with one wait inserted.
- Bus-release mode timing
  Figure 21.14 shows the bus-release mode timing.



Note: * Specification from the earliest negation timing of A_{23} to A_0, $\overline{CS}_n$, and $\overline{RD}$.

Figure 21.11 Basic Bus Cycle: Two-State Access

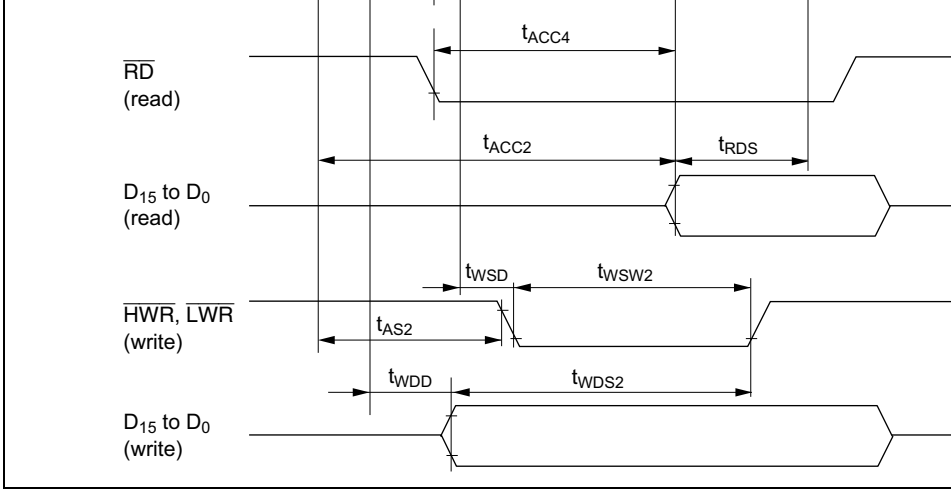


Figure 21.12 Basic Bus Cycle: Three-State Access

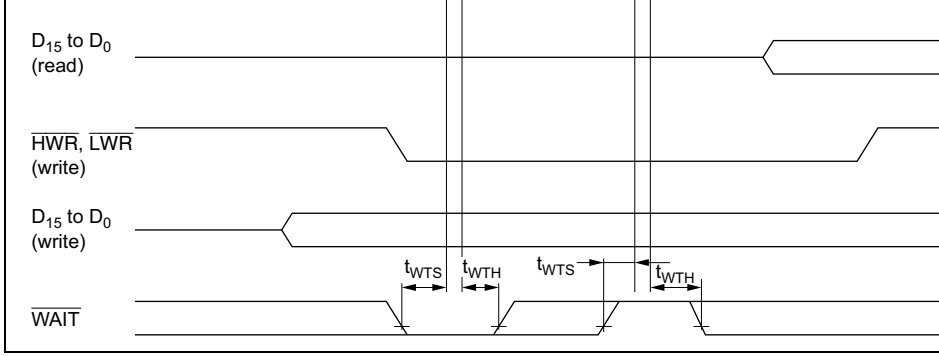
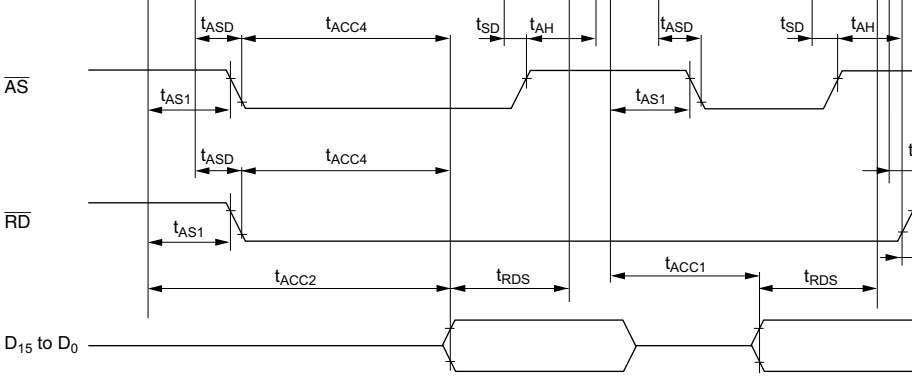
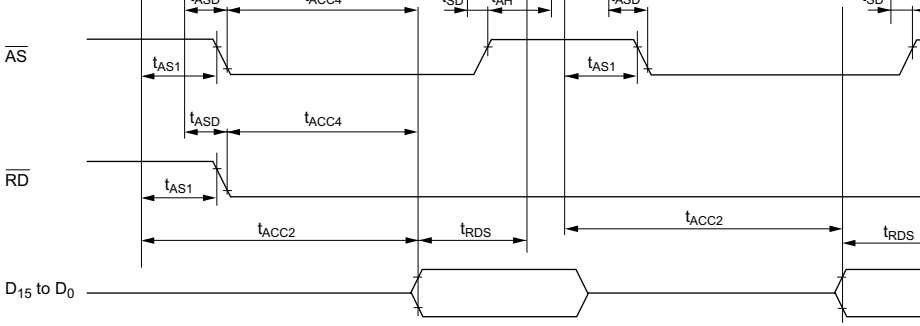


Figure 21.13 Basic Bus Cycle: Three-State Access with One Wait Sta



Note: * Specification from the earliest negation timing of A23 to A0, $\overline{CSn}$, and $\overline{RD}$.

Figure 21.14 Burst ROM Access Timing: Two-State Access



Note: * Specification from the earliest negation timing of A23 to A0, $\overline{CS}$n, and $\overline{RD}$.

Figure 21.15 Burst ROM Access Timing: Three-State Access

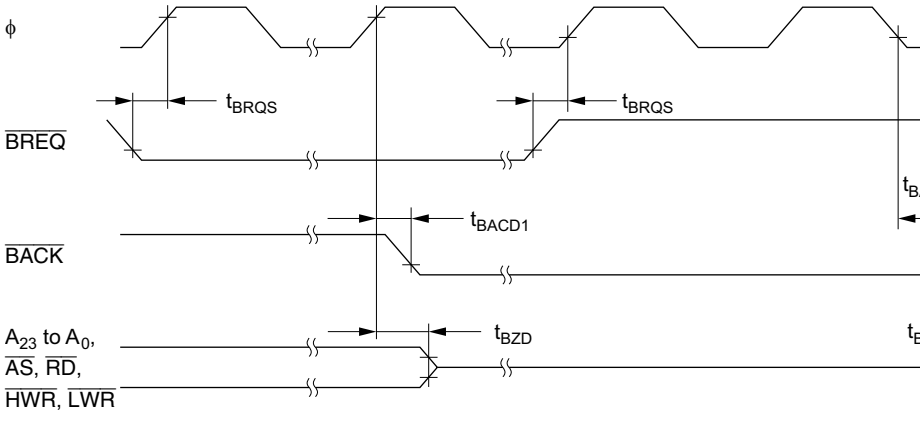
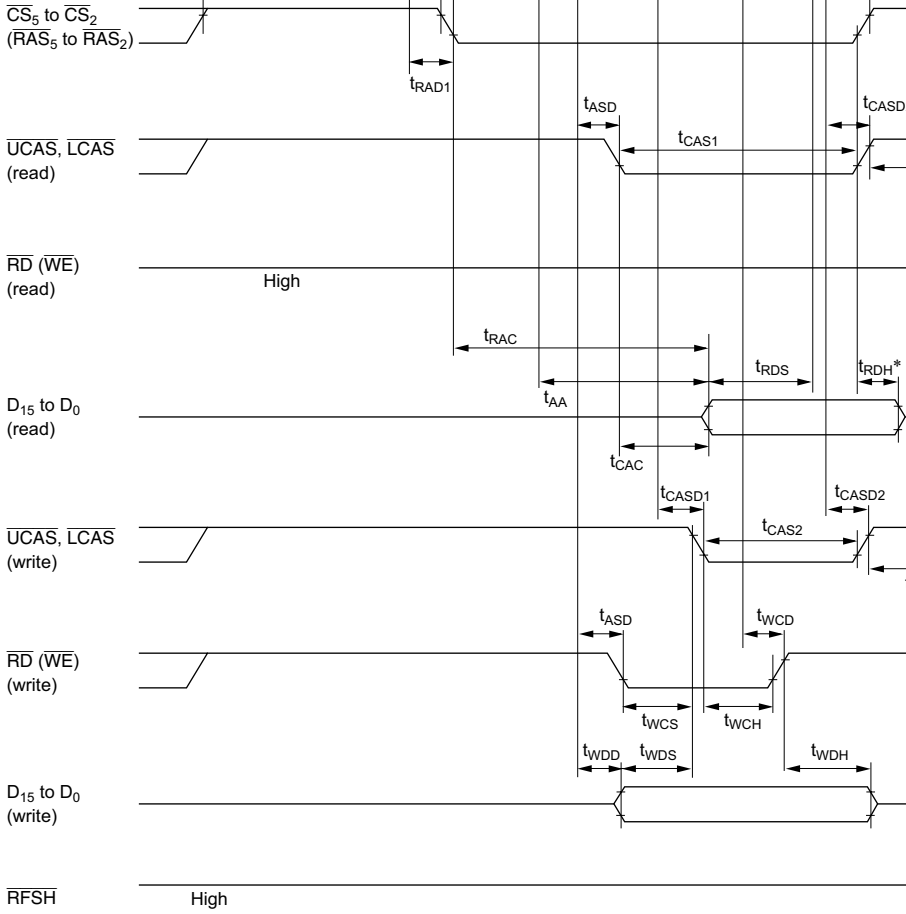


Figure 21.16 Bus-Release Mode Timing

Figure 21.19 shows the timing of the self-refresh.



Note: * Specification from the earliest negation timing of $\overline{RAS}$ and $\overline{CAS}$.

Figure 21.17 DRAM Bus Timing (Read/Write)

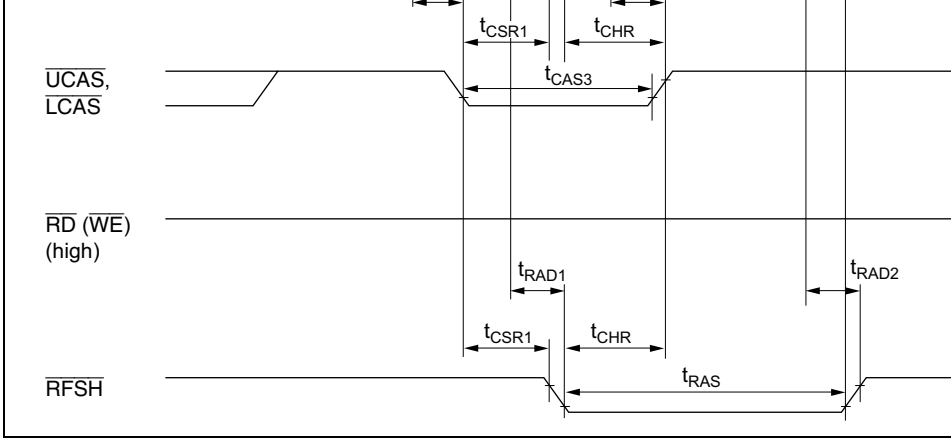


Figure 21.18 DRAM Bus Timing (CAS Before RAS Refresh)

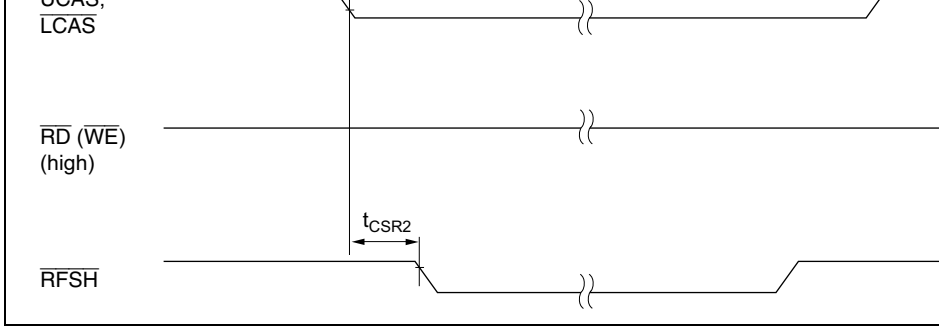


Figure 21.19 DRAM Bus Timing (Self-Refresh)

21.3.5 TPC and I/O Port Timing

Figure 21.20 shows the TPC and I/O port input/output timing.

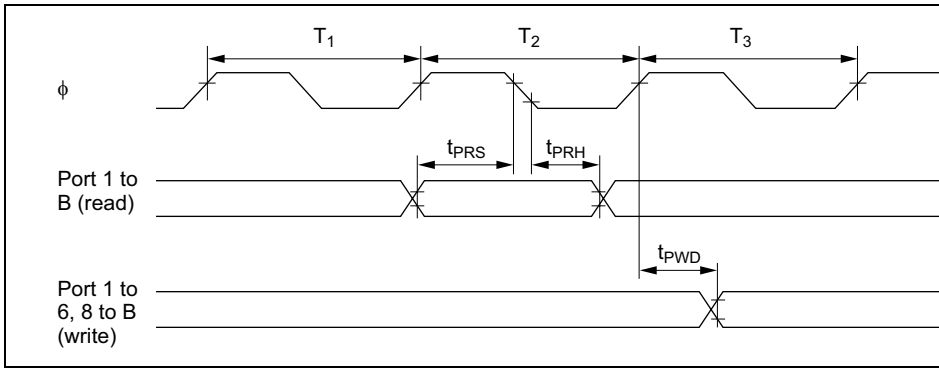
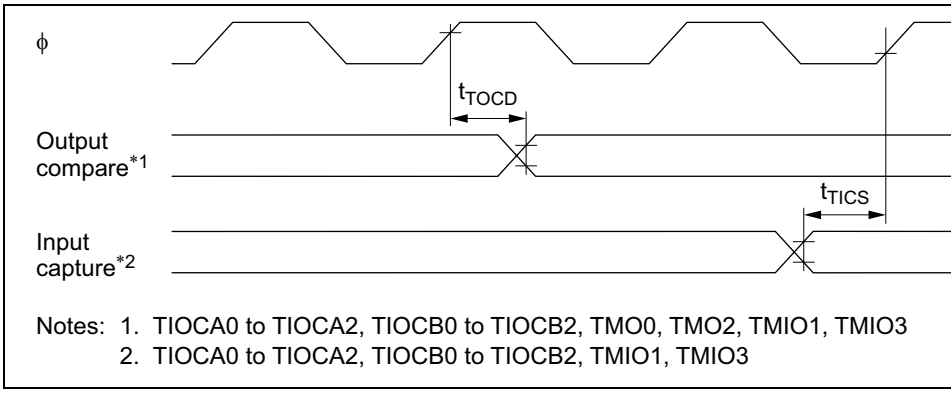


Figure 21.20 TPC and I/O Port Input/Output Timing



Notes: 1. TIOCA0 to TIOCA2, TIOCB0 to TIOCB2, TMO0, TMO2, TMIO1, TMIO3
2. TIOCA0 to TIOCA2, TIOCB0 to TIOCB2, TMIO1, TMIO3

Figure 21.21 Timer Input/Output Timing

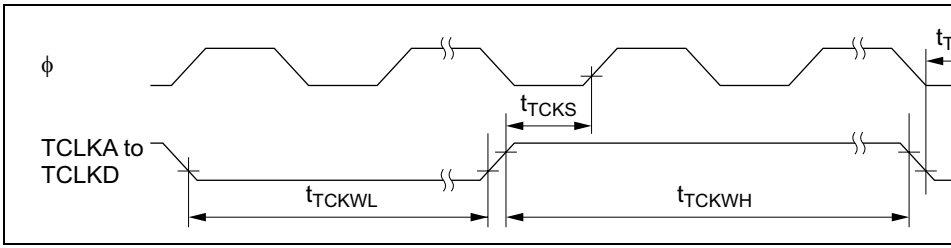


Figure 21.22 Timer External Clock Input Timing

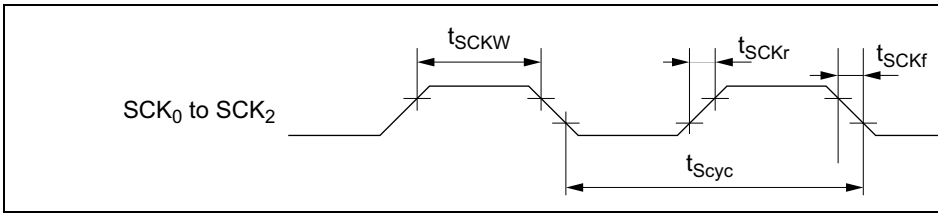


Figure 21.23 SCI Input Clock Timing

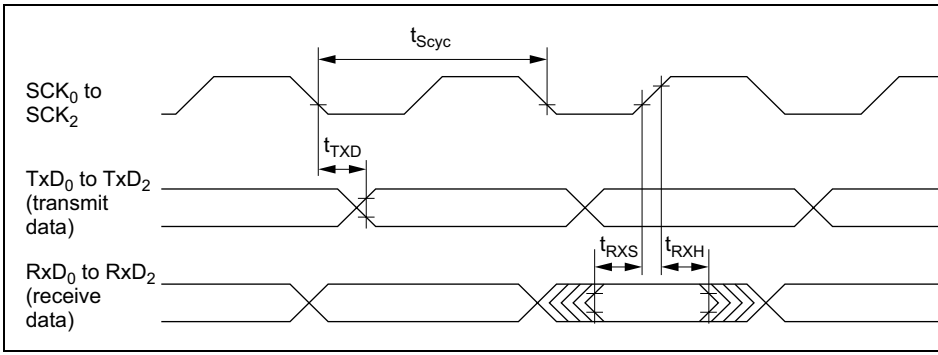


Figure 21.24 SCI Input/Output Timing in Synchronous Mode

Figure 21.27 shows DMAC $\overline{\text{DREQ}}$ input timing.

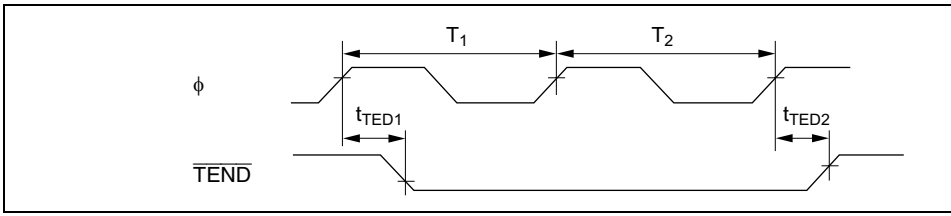


Figure 21.25 DMAC $\overline{\text{TEND}}$ Output Timing for 2-State Access

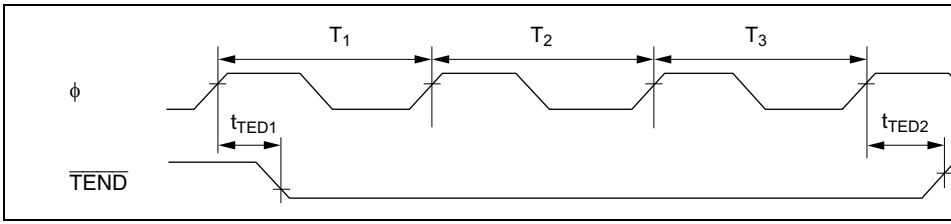


Figure 21.26 DMAC $\overline{\text{TEND}}$ Output Timing for 3-State Access

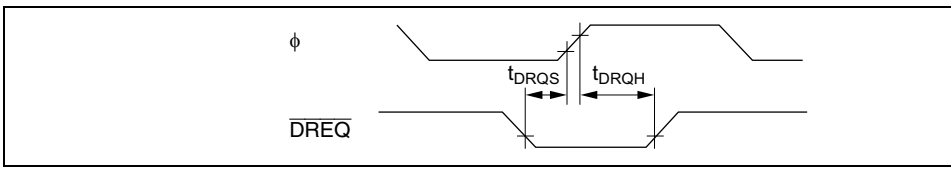


Figure 21.27 DMAC $\overline{\text{DREQ}}$ Input Timing

Figure 21.28 Input Signal Rise and Fall Timing

Rs	General source register
Rn	General register
ERd	General destination register (address register or 32-bit register)
ERs	General source register (address register or 32-bit register)
ERn	General register (32-bit register)
(EAd)	Destination operand
(EAs)	Source operand
PC	Program counter
SP	Stack pointer
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
disp	Displacement
→	Transfer from the operand on the left to the operand on the right, or trans the state on the left to the state on the right
+	Addition of the operands on both sides
–	Subtraction of the operand on the right from the operand on the left
×	Multiplication of the operands on both sides
÷	Division of the operand on the left by the operand on the right
∧	Logical AND of the operands on both sides
∨	Logical OR of the operands on both sides
⊕	Exclusive logical OR of the operands on both sides
¬	NOT (logical complement)
( ), < >	Contents of operand

Note: General registers include 8-bit registers (R0H to R7H and R0L to R7L) and 16-b (R0 to R7 and E0 to E7).

Δ		Varies depending on conditions, described in notes

Mnemonic	S	#	E	◎	◎	◎	◎	◎	↑	Operation	I	H	N	Z	V
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:24, ERs), Rd	B				8					@(d:24, 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:24, Rd	B						6			@aa:24 → 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:24, ERd)	B				8					Rs8 → @(d:24, 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:24	B						6			Rs8 → @aa:24	—	—	↕	↕	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
MOV.W @(d:16, ERs), Rd	W				4					@(d:16, ERs) → Rd16	—	—	↕	↕	0
MOV.W @(d:24, ERs), Rd	W				8					@(d:24, ERs) → Rd16	—	—	↕	↕	0
MOV.W @ERs+, Rd	W					2				@ERs → Rd16 ERs32+2 → @ERd32	—	—	↕	↕	0
MOV.W @aa:16, Rd	W						4			@aa:16 → Rd16	—	—	↕	↕	0

MOV.W Rs, @(d:16, ERd)	W				4						Rs16 → @(d:16, ERd)	—	—	↕	↕	0	–
MOV.W Rs, @(d:24, ERd)	W				8						Rs16 → @(d:24, ERd)	—	—	↕	↕	0	–
MOV.W Rs, @–ERd	W					2					ERd32–2 → ERd32 Rs16 → @ERd	—	—	↕	↕	0	–
MOV.W Rs, @aa:16	W						4				Rs16 → @aa:16	—	—	↕	↕	0	–
MOV.W Rs, @aa:24	W						6				Rs16 → @aa:24	—	—	↕	↕	0	–
MOV.L #xx:32, Rd	L	6									#xx:32 → Rd32	—	—	↕	↕	0	–
MOV.L ERs, ERd	L		2								ERs32 → ERd32	—	—	↕	↕	0	–
MOV.L @ERs, ERd	L			4							@ERs → ERd32	—	—	↕	↕	0	–
MOV.L @(d:16, ERs), ERd	L				6						@(d:16, ERs) → ERd32	—	—	↕	↕	0	–
MOV.L @(d:24, ERs), ERd	L				10						@(d:24, ERs) → ERd32	—	—	↕	↕	0	–
MOV.L @ERs+, ERd	L					4					@ERs → ERd32 ERs32+4 → ERs32	—	—	↕	↕	0	–
MOV.L @aa:16, ERd	L						6				@aa:16 → ERd32	—	—	↕	↕	0	–
MOV.L @aa:24, ERd	L						8				@aa:24 → ERd32	—	—	↕	↕	0	–
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:24, ERd)	L				10						ERs32 → @(d:24, 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:24	L						8				ERs32 → @aa:24	—	—	↕	↕	0	–
POP.W Rn	W									2	@SP → Rn16 SP+2 → SP	—	—	↕	↕	0	–
POP.L ERn	L									4	@SP → ERn32 SP+4 → SP	—	—	↕	↕	0	–

											ERn32 → @SP					
MOVFPE @aa:16, Rd	B						4				Cannot be used in the H8/3028 Seires	Cannot be used in t H8/3028 Seires				
MOVTPE Rs, @aa:16	B						4				Cannot be used in the H8/3028 Seires	Cannot be used in t H8/3028 Seires				

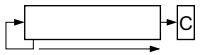
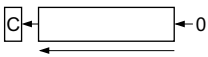
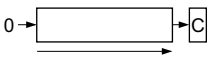
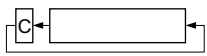
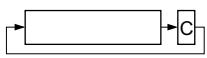
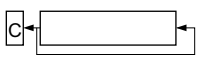
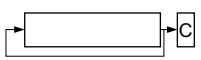
2. Arithmetic instructions

Mnemonic	Operand Size	Addressing Mode and Instruction Length (bytes)									Operation	Condition Code				
		#xx	Rn	@ERn	@(d, ERn)	@–ERn/@ERn+	@aa	@(d, PC)	@@aa	|		I	H	N	Z	V
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	—	(1)	↕	↕	↕
ADD.W Rs, Rd	W		2								Rd16+Rs16 → Rd16	—	(1)	↕	↕	↕
ADD.L #xx:32, ERd	L	6									ERd32+#xx:32 → ERd32	—	(2)	↕	↕	↕
ADD.L ERs, ERd	L		2								ERd32+ERs32 → ERd32	—	(2)	↕	↕	↕
ADDX.B #xx:8, Rd	B	2									Rd8+#xx:8 +C → Rd8	—	↕	↕	(3)	↕
ADDX.B Rs, Rd	B		2								Rd8+Rs8 +C → Rd8	—	↕	↕	(3)	↕
ADDS.L #1, ERd	L		2								ERd32+1 → ERd32	—	—	—	—	—
ADDS.L #2, ERd	L		2								ERd32+2 → ERd32	—	—	—	—	—
ADDS.L #4, ERd	L		2								ERd32+4 → ERd32	—	—	—	—	—
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	—	—	↕	↕	↕

DAA Rd	B		2								Rd8 decimal adjust → Rd8	—	*	↕	↕	*	–
SUB.B Rs, Rd	B		2								Rd8–Rs8 → Rd8	—	↕	↕	↕	↕	↕
SUB.W #xx:16, Rd	W	4									Rd16–#xx:16 → Rd16	—	(1)	↕	↕	↕	↕
SUB.W Rs, Rd	W		2								Rd16–Rs16 → Rd16	—	(1)	↕	↕	↕	↕
SUB.L #xx:32, ERd	L	6									ERd32–#xx:32 → ERd32	—	(2)	↕	↕	↕	↕
SUB.L ERs, ERd	L		2								ERd32–ERs32 → ERd32	—	(2)	↕	↕	↕	↕
SUBX.B #xx:8, Rd	B	2									Rd8–#xx:8–C → Rd8	—	↕	↕	(3)	↕	↕
SUBX.B Rs, Rd	B		2								Rd8–Rs8–C → Rd8	—	↕	↕	(3)	↕	↕
SUBS.L #1, ERd	L		2								ERd32–1 → ERd32	—	—	—	—	—	–
SUBS.L #2, ERd	L		2								ERd32–2 → ERd32	—	—	—	—	—	–
SUBS.L #4, ERd	L		2								ERd32–4 → ERd32	—	—	—	—	—	–
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.Rd	B		2								Rd8 decimal adjust → Rd8	—	*	↕	↕	*	–
MULXU. B Rs, Rd	B		2								Rd8 × Rs8 → Rd16 (unsigned multiplication)	—	—	—	—	—	–
MULXU. W Rs, ERd	W		2								Rd16 × Rs16 → ERd32 (unsigned multiplication)	—	—	—	—	—	–
MULXS. B Rs, Rd	B		4								Rd8 × Rs8 → Rd16 (signed multiplication)	—	—	↕	↕	—	–
MULXS. W Rs, ERd	W		4								Rd16 × Rs16 → ERd32 (signed multiplication)	—	—	↕	↕	—	–
DIVXU. B Rs, Rd	B		2								Rd16 ÷ Rs8 → Rd16 (RdH: remainder, RdL: quotient) (unsigned division)	—	—	(6)	(7)	—	–

											(unsigned division)					
DIVXS. B Rs, Rd	B		4								Rd16 ÷ Rs8 → Rd16 (RdH: remainder, RdL: quotient) (signed division)	—	—	(8)	(7)	—
DIVXS. W Rs, ERd	W		4								ERd32 ÷ Rs16 → ERd32 (Ed: remainder, Rd: quotient) (signed division)	—	—	(8)	(7)	—
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	—	(1)	↕	↕	↕
CMP.W Rs, Rd	W		2								Rd16–Rs16	—	(1)	↕	↕	↕
CMP.L #xx:32, ERd	L	6									ERd32–#xx:32	—	(2)	↕	↕	↕
CMP.L ERs, ERd	L		2								ERd32–ERs32	—	(2)	↕	↕	↕
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.W Rd	W		2								0 → (<bits 15 to 8> of Rd16)	—	—	0	↕	0
EXTU.L ERd	L		2								0 → (<bits 31 to 16> of ERd32)	—	—	0	↕	0
EXTS.W Rd	W		2								(<bit 7> of Rd16) → (<bits 15 to 8> of Rd16)	—	—	↕	↕	0
EXTS.L ERd	L		2								(<bit 15> of ERd32) → (<bits 31 to 16> of ERd32)	—	—	↕	↕	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.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.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.B Rd	B		2								¬Rd8 → Rd8	—	—	↕	↕	0	–
NOT.W Rd	W		2								¬Rd16 → Rd16	—	—	↕	↕	0	–
NOT.L ERd	L		2								¬Rd32 → Rd32	—	—	↕	↕	0	–

SHAL.W Rd	W		2								—	—	↕	↕	↕
SHAL.L ERd	L		2								—	—	↕	↕	↕
SHAR.B Rd	B		2								—	—	↕	↕	0
SHAR.W Rd	W		2								—	—	↕	↕	0
SHAR.L ERd	L		2								—	—	↕	↕	0
SHLL.B Rd	B		2								—	—	↕	↕	0
SHLL.W Rd	W		2								—	—	↕	↕	0
SHLL.L ERd	L		2								—	—	↕	↕	0
SHLR.B Rd	B		2								—	—	↕	↕	0
SHLR.W Rd	W		2								—	—	↕	↕	0
SHLR.L ERd	L		2								—	—	↕	↕	0
ROTXL.B Rd	B		2								—	—	↕	↕	0
ROTXL.W Rd	W		2								—	—	↕	↕	0
ROTXL.L ERd	L		2								—	—	↕	↕	0
ROTXR.B Rd	B		2								—	—	↕	↕	0
ROTXR.W Rd	W		2								—	—	↕	↕	0
ROTXR.L ERd	L		2								—	—	↕	↕	0
ROTL.B Rd	B		2								—	—	↕	↕	0
ROTL.W Rd	W		2								—	—	↕	↕	0
ROTL.L ERd	L		2								—	—	↕	↕	0
ROTR.B Rd	B		2								—	—	↕	↕	0
ROTR.W Rd	W		2								—	—	↕	↕	0
ROTR.L ERd	L		2								—	—	↕	↕	0

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 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	—	—	—	—	—	–
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 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	—	—	—	—	—	–
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 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)	—	—	—	—	—	–
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	—	—	—	↕	—	–
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	—	—	—	↕	—	–
BLD #xx:3, Rd	B		2								(#xx:3 of Rd8) → C	—	—	—	—	—	↕

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	—	—	—	—	—
BST #xx:3, Rd	B		2							C → (#xx:3 of Rd8)	—	—	—	—	—
BST #xx:3, @ERd	B			4						C → (#xx:3 of @ERd24)	—	—	—	—	—
BST #xx:3, @aa:8	B						4			C → (#xx:3 of @aa:8)	—	—	—	—	—
BIST #xx:3, Rd	B		2							¬ C → (#xx:3 of Rd8)	—	—	—	—	—
BIST #xx:3, @ERd	B			4						¬ C → (#xx:3 of @ERd24)	—	—	—	—	—
BIST #xx:3, @aa:8	B						4			¬ C → (#xx:3 of @aa:8)	—	—	—	—	—
BAND #xx:3, Rd	B		2							C∧(#xx:3 of Rd8) → C	—	—	—	—	—
BAND #xx:3, @ERd	B			4						C∧(#xx:3 of @ERd24) → C	—	—	—	—	—
BAND #xx:3, @aa:8	B						4			C∧(#xx:3 of @aa:8) → C	—	—	—	—	—
BIAND #xx:3, Rd	B		2							C∧ ¬ (#xx:3 of Rd8) → C	—	—	—	—	—
BIAND #xx:3, @ERd	B			4						C∧ ¬ (#xx:3 of @ERd24) → C	—	—	—	—	—
BIAND #xx:3, @aa:8	B						4			C∧ ¬ (#xx:3 of @aa:8) → C	—	—	—	—	—
BOR #xx:3, Rd	B		2							C∨(#xx:3 of Rd8) → C	—	—	—	—	—
BOR #xx:3, @ERd	B			4						C∨(#xx:3 of @ERd24) → C	—	—	—	—	—
BOR #xx:3, @aa:8	B						4			C∨(#xx:3 of @aa:8) → C	—	—	—	—	—
BIOR #xx:3, Rd	B		2							C∨ ¬ (#xx:3 of Rd8) → C	—	—	—	—	—
BIOR #xx:3, @ERd	B			4						C∨ ¬ (#xx:3 of @ERd24) → C	—	—	—	—	—
BIOR #xx:3, @aa:8	B						4			C∨ ¬ (#xx:3 of @aa:8) → C	—	—	—	—	—
BXOR #xx:3, Rd	B		2							C⊕(#xx:3 of Rd8) → C	—	—	—	—	—
BXOR #xx:3, @ERd	B			4						C⊕(#xx:3 of @ERd24) → C	—	—	—	—	—
BXOR #xx:3, @aa:8	B						4			C⊕(#xx:3 of @aa:8) → C	—	—	—	—	—
BIXOR #xx:3, Rd	B		2							C⊕ ¬ (#xx:3 of Rd8) → C	—	—	—	—	—
BIXOR #xx:3, @ERd	B			4						C⊕ ¬ (#xx:3 of @ERd24) → C	—	—	—	—	—
BIXOR #xx:3, @aa:8	B						4			C⊕ ¬ (#xx:3 of @aa:8) → C	—	—	—	—	—

BRA d:16 (BT d:16)	—						4			is true then PC ← PC+d else next;		—	—	—	—	—	—
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:8 (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					—	—	—	—	—	—
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					—	—	—	—	—	—

										else next;						
JMP @ERn	—			2							PC ← ERn	—	—	—	—	—
JMP @aa:24	—						4				PC ← aa:24	—	—	—	—	—
JMP @@aa:8	—								2		PC ← @aa:8	—	—	—	—	—
BSR d:8	—							2			PC → @–SP PC ← PC+d:8	—	—	—	—	—
BSR d:16	—							4			PC → @–SP PC ← PC+d:16	—	—	—	—	—
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	—									2	PC ← @SP+	—	—	—	—	—

											CCR → @–SP <vector> → PC						
RTE	—										CCR ← @SP+ PC ← @SP+	↕	↕	↕	↕	↕	↕
SLEEP	—										Transition to powerdown state	—	—	—	—	—	—
LDC #xx:8, CCR	B	2									#xx:8 → CCR	↕	↕	↕	↕	↕	↕
LDC Rs, CCR	B		2								Rs8 → CCR	↕	↕	↕	↕	↕	↕
LDC @ERs, CCR	W			4							@ERs → CCR	↕	↕	↕	↕	↕	↕
LDC @(d:16, ERs), CCR	W				6						@(d:16, ERs) → CCR	↕	↕	↕	↕	↕	↕
LDC @(d:24, ERs), CCR	W				10						@(d:24, ERs) → CCR	↕	↕	↕	↕	↕	↕
LDC @ERs+, CCR	W					4					@ERs → CCR ERs32+2 → ERs32	↕	↕	↕	↕	↕	↕
LDC @aa:16, CCR	W						6				@aa:16 → CCR	↕	↕	↕	↕	↕	↕
LDC @aa:24, CCR	W						8				@aa:24 → CCR	↕	↕	↕	↕	↕	↕
STC CCR, Rd	B		2								CCR → Rd8	—	—	—	—	—	—
STC CCR, @ERd	W			4							CCR → @ERd	—	—	—	—	—	—
STC CCR, @(d:16, ERd)	W				6						CCR → @(d:16, ERd)	—	—	—	—	—	—
STC CCR, @(d:24, ERd)	W				10						CCR → @(d:24, ERd)	—	—	—	—	—	—
STC CCR, @–ERd	W					4					ERd32–2 → ERd32 CCR → @ERd	—	—	—	—	—	—
STC CCR, @aa:16	W						6				CCR → @aa:16	—	—	—	—	—	—
STC CCR, @aa:24	W						8				CCR → @aa:24	—	—	—	—	—	—
ANDC #xx:8, CCR	B	2									CCR∧#xx:8 → CCR	↕	↕	↕	↕	↕	↕
ORC #xx:8, CCR	B	2									CCR∨#xx:8 → CCR	↕	↕	↕	↕	↕	↕
XORC #xx:8, CCR	B	2									CCR⊕#xx:8 → CCR	↕	↕	↕	↕	↕	↕
NOP	—									2	PC ← PC+2	—	—	—	—	—	—

											repeat @R5 @R6 R5+1 → R5 R6+1 → R6 R4L–1 → R4L until R4L=0 else next;							
EEPMOV. W	—									4	if R4 ≠ 0 repeat @R5 → @R6 R5+1 → R5 R6+1 → R6 R4–1 → R4 until R4=0 else next;	—	—	—	—	—	—	

Notes: 1. The number of states is the number of states required for execution when t
instruction and its operands are located in on-chip memory. For other cases
A.3.

2. n is the value set in register R4L or R4.

(1) Set to 1 when a carry or borrow occurs at bit 11; otherwise cleared to 0.

(2) Set to 1 when a carry or borrow occurs at bit 27; otherwise cleared to 0.

(3) Retains its previous value when the result is zero; otherwise cleared to 0.

(4) Set to 1 when the adjustment produces a carry; otherwise retains its previo

(5) The number of states required for execution of an instruction that transfers
synchronization with the E clock is variable.

(6) Set to 1 when the divisor is negative; otherwise cleared to 0.

(7) Set to 1 when the divisor is zero; otherwise cleared to 0.

(8) Set to 1 when the quotient is negative; otherwise cleared to 0.

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.2 (2)	STC	LDC	ORC	XORC	ANDC	LDC	ADD		Table A.2 (2)	Table A.2 (2)	MOV	
1	Table A.2 (2)	Table A.2 (2)	Table A.2 (2)	Table A.2 (2)	OR.B	XOR.B	AND.B	Table A.2 (2)	SUB		Table A.2 (2)	Table A.2 (2)	CMP	
2	MOV.B													
3														
4	BRA	BRN	BHI	BLS	BCC	BCS	BNE	BNQ	BVC	BVS	BPL	BMI	BGE	BLT
5	MULXU	DIVXU	MULXU	DIVXU	RTS	BSR	RTE	TRAPA	Table A.2 (2)	JMP			BSR	
6	BSET	BNOT	BCLR	BTST	OR	XOR	AND	BST / BIST	MOV					
7					BOR / BIOR	BXOR / BIXOR	BAND / BIAND	BLD / BILD	MOV		Table A.2 (2)	Table A.2 (2)	EEPMOV	Tab
8	ADD													
9	ADDX													
A	CMP													
B	SUBX													
C	OR													
D	XOR													

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				LDC/STC				SLEEP				Table A.2 (3)
0A	INC								ADD				
0B	ADDS					INC		INC	ADDS				
0F	DAA								MOV				
10	SHLL			SHLL					SHAL			SHAL	
11	SHLR			SHLR					SHAR			SHAR	
12	ROTXL			ROTXL					ROTL			ROTL	
13	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

Instruction code:

1st byte		2nd byte		3rd byte		4th byte	
AH	AL	BH	BL	CH	CL	DH	DL

Instruction when most signi

Instruction when most signi

CL / AH / ALBH / BLCH	0	1	2	3	4	5	6	7	8	9	A	B	C
01406										LDC / STC		LDC / STC	
01C05	MULXS		MULXS										
01D05		DIVIXS		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 designation field.
2. aa is the absolute address field.

Examples of Calculation of Number of States Required for Execution

Examples: Advanced mode, stack located in external address space, on-chip supporti
accessed with 8-bit bus width, external devices accessed in three states with one wait s
16-bit bus width.

BSET #0, @FFFFC7:8

From table A.4, I = L = 2 and J = K = M = N = 0
From table A.3, $S_I = 4$ and $S_L = 3$
Number of states = $2 \times 4 + 2 \times 3 = 14$

JSR @@30

From table A.4, I = J = K = 2 and L = M = N = 0
From table A.3, $S_I = S_J = S_K = 4$
Number of states = $2 \times 4 + 2 \times 4 + 2 \times 4 = 24$

Branch address read	S_J			
Stack operation	S_K			
Byte data access	S_L	3	2	3 + m
Word data access	S_M	6	4	6 + 2m
Internal operation	S_N	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		
BAND	BAND #xx:3, Rd	1		
	BAND #xx:3, @ERd	2		1
	BAND #xx:3, @aa:8	2		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 Rn, Rd	1		
	BCLR Rn, @ERd	2		2
	BCLR Rn, @aa:8	2		2
BIAND	BIAND #xx:3, Rd	1		
	BIAND #xx:3, @ERd	2		1
	BIAND #xx:3, @aa:8	2		1
BILD	BILD #xx:3, Rd	1		
	BILD #xx:3, @ERd	2		1
	BILD #xx:3, @aa:8	2		1
BIOR	BIOR #xx:8, Rd	1		
	BIOR #xx:8, @ERd	2		1
	BIOR #xx:8, @aa:8	2		1
BIST	BIST #xx:3, Rd	1		
	BIST #xx:3, @ERd	2		2
	BIST #xx:3, @aa:8	2		2
BIXOR	BIXOR #xx:3, Rd	1		
	BIXOR #xx:3, @ERd	2		1
	BIXOR #xx:3, @aa:8	2		1
BLD	BLD #xx:3, Rd	1		
	BLD #xx:3, @ERd	2		1
	BLD #xx:3, @aa:8	2		1

	BOR #xx:3, @ERd		2		1
	BOR #xx:3, @aa:8		2		1
BSET	BSET #xx:3, Rd		1		
	BSET #xx:3, @ERd		2		2
	BSET #xx:3, @aa:8		2		2
	BSET Rn, Rd		1		
	BSET Rn, @ERd		2		2
	BSET Rn, @aa:8		2		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
BTST	BTST #xx:3, Rd		1		
	BTST #xx:3, @ERd		2		1
	BTST #xx:3, @aa:8		2		1
	BTST Rn, Rd		1		
	BTST Rn, @ERd		2		1
	BTST Rn, @aa:8		2		1
BXOR	BXOR #xx:3, Rd		1		
	BXOR #xx:3, @ERd		2		1
	BXOR #xx:3, @aa:8		2		1
CMP	CMP.B #xx:8, Rd		1		
	CMP.B Rs, Rd		1		
	CMP.W #xx:16, Rd		2		
	CMP.W Rs, Rd		1		
	CMP.L #xx:32, ERd		3		
	CMP.L ERs, ERd		1		
DAA	DAA Rd		1		
DAS	DAS 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	LDC #xx:8, CCR		1						
	LDC Rs, CCR		1						
	LDC @ERs, CCR		2						1
	LDC @(d:16, ERs), CCR		3						1
	LDC @(d:24, ERs), CCR		5						1
	LDC @ERs+, CCR		2						1
	LDC @aa:16, CCR		3						1
	LDC @aa:24, CCR		4						1

MOV.B @aa:16, Rd	2		1		
MOV.B @aa:24, Rd	3		1		
MOV.B Rs, @ERd	1		1		
MOV.B Rs, @(d:16, ERd)	2		1		
MOV.B Rs, @(d:24, 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:24	3		1		
MOV.W #xx:16, Rd	2				
MOV.W Rs, Rd	1				
MOV.W @ERs, Rd	1				1
MOV.W @(d:16, ERs), Rd	2				1
MOV.W @(d:24, ERs), Rd	4				1
MOV.W @ERs+, Rd	1				1
MOV.W @aa:16, Rd	2				1
MOV.W @aa:24, Rd	3				1
MOV.W Rs, @ERd	1				1
MOV.W Rs, @(d:16, ERd)	2				1
MOV.W Rs, @(d:24, ERd)	4				1
MOV.W Rs, @–ERd	1				1
MOV.W Rs, @aa:16	2				1
MOV.W Rs, @aa:24	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:24, ERs), ERd	5				2
MOV.L @ERs+, ERd	2				2
MOV.L @aa:16, ERd	3				2
MOV.L @aa:24, ERd	4				2
MOV.L ERs, @ERd	2				2
MOV.L ERs, @(d:16, ERd)	3				2
MOV.L ERs, @(d:24, ERd)	5				2
MOV.L ERs, @–ERd	2				2
MOV.L ERs, @aa:16	3				2
MOV.L ERs, @aa:24	4				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				
NOT	NOT.B Rd	1				
	NOT.W Rd	1				
	NOT.L ERd	1				
OR	OR.B #xx:8, Rd	1				
	OR.B Rs, Rd	1				
	OR.W #xx:16, Rd	2				
	OR.W Rs, Rd	1				
	OR.L #xx:32, ERd	3				
	OR.L ERs, ERd	2				
ORC	ORC #xx:8, CCR	1				
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.W Rd	1				
	ROTL.L ERd	1				
ROTR	ROTR.B Rd	1				
	ROTR.W Rd	1				
	ROTR.L ERd	1				
ROTXL	ROTXL.B Rd	1				
	ROTXL.W Rd	1				
	ROTXL.L ERd	1				
ROTXR	ROTXR.B Rd	1				
	ROTXR.W Rd	1				
	ROTXR.L ERd	1				
RTE	RTE	2		2		

	SHAR.L ERd	1				
SHLL	SHLL.B Rd	1				
	SHLL.W Rd	1				
	SHLL.L ERd	1				
SHLR	SHLR.B Rd	1				
	SHLR.W Rd	1				
	SHLR.L ERd	1				
SLEEP	SLEEP	1				
STC	STC CCR, Rd	1				
	STC CCR, @ERd	2				1
	STC CCR, @(d:16, ERd)	3				1
	STC CCR, @(d:24, ERd)	5				1
	STC CCR, @–ERd	2				1
	STC CCR, @aa:16	3				1
	STC CCR, @aa:24	4				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				
TRAPA	TRAPA #x:2 Normal	2	1	2		
	Advanced	2	2	2		
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				

Notes: 1. n is the value set in register R4L or R4. The source and destination are acc
times each.

2. Not available in the H8/3028 Group.

H'EE001	P2DDR	8	$P2_7$DDR	$P2_6$DDR	$P2_5$DDR	$P2_4$DDR	$P2_3$DDR	$P2_2$DDR	$P2_1$DDR	$P2_0$DDR
H'EE002	P3DDR	8	$P3_7$DDR	$P3_6$DDR	$P3_5$DDR	$P3_4$DDR	$P3_3$DDR	$P3_2$DDR	$P3_1$DDR	$P3_0$DDR
H'EE003	P4DDR	8	$P4_7$DDR	$P4_6$DDR	$P4_5$DDR	$P4_4$DDR	$P4_3$DDR	$P4_2$DDR	$P4_1$DDR	$P4_0$DDR
H'EE004	P5DDR	8	—	—	—	—	$P5_3$DDR	$P5_2$DDR	$P5_1$DDR	$P5_0$DDR
H'EE005	P6DDR	8	—	$P6_6$DDR	$P6_5$DDR	$P6_4$DDR	$P6_3$DDR	$P6_2$DDR	$P6_1$DDR	$P6_0$DDR
H'EE006	—		—	—	—	—	—	—	—	—
H'EE007	P8DDR	8	—	—	—	$P8_4$DDR	$P8_3$DDR	$P8_2$DDR	$P8_1$DDR	$P8_0$DDR
H'EE008	P9DDR	8	—	—	$P9_5$DDR	$P9_4$DDR	$P9_3$DDR	$P9_2$DDR	$P9_1$DDR	$P9_0$DDR
H'EE009	PADDR	8	PA_7DDR	PA_6DDR	PA_5DDR	PA_4DDR	PA_3DDR	PA_2DDR	PA_1DDR	PA_0DDR
H'EE00A	PBDDR	8	PB_7DDR	PB_6DDR	PB_5DDR	PB_4DDR	PB_3DDR	PB_2DDR	PB_1DDR	PB_0DDR
H'EE00B	—		—	—	—	—	—	—	—	—
H'EE00C	—		—	—	—	—	—	—	—	—
H'EE00D	—		—	—	—	—	—	—	—	—
H'EE00E	—		—	—	—	—	—	—	—	—
H'EE00F	—		—	—	—	—	—	—	—	—
H'EE010	—		—	—	—	—	—	—	—	—
H'EE011	MDCR	8	—	—	—	—	—	MDS2	MDS1	MDS0
H'EE012	SYSCR	8	SSBY	STS2	STS1	STS0	UE	NMIEG	SSOE	RAME
H'EE013	BRCR	8	A23E	A22E	A21E	A20E	—	—	—	BRLE
H'EE014	ISCR	8	—	—	IRQ5SC	IRQ4SC	IRQ3SC	IRQ2SC	IRQ1SC	IRQ0SC
H'EE015	IER	8	—	—	IRQ5E	IRQ4E	IRQ3E	IRQ2E	IRQ1E	IRQ0E
H'EE016	ISR	8	—	—	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	IRQ0F
H'EE017	—		—	—	—	—	—	—	—	—
H'EE018	IPRA	8	IPRA7	IPRA6	IPRA5	IPRA4	IPRA3	IPRA2	IPRA1	IPRA0
H'EE019	IPRB	8	IPRB7	IPRB6	IPRB5	—	IPRB3	IPRB2	IPRB1	—
H'EE01A	DASTCR	8	—	—	—	—	—	—	—	DASTE
H'EE01B	DIVCR	8	—	—	—	—	—	—	DIV1	DIV0
H'EE01C	MSTCRH	8	PSTOP	—	—	—	—	MSTPH2	MSTPH1	MSTPH0
H'EE01D	MSTCRL	8	MSTPL7	—	MSTPL5	MSTPL4	MSTPL3	MSTPL2	—	MSTPL0
H'EE01E	ADRCR	8	—	—	—	—	—	—	—	ADRCTL
H'EE01F	CSCR	8	CS7E	CS6E	CS5E	CS4E	—	—	—	—

H'EE026	DRCRA	8	DRAS2	DRAS1	DRAS0	—	BE	RDM	SRFMD	RFSHE
H'EE027	DRCRB	8	MXC1	MXC0	CSEL	RCYCE	—	TPC	RCW	RLW
H'EE028	RTMCSR	8	CMF	CMIE	CKS2	CKS1	CKS0	—	—	—
H'EE029	RTCNT	8								
H'EE02A	RTCOR	8								
H'EE02B	Reserved area (access prohibited)									
H'EE02C										
H'EE02D										
H'EE02E										
H'EE02F										
H'EE030	FLMCR1	8	FWE	SWE	ESU	PSU	EV	PV	E	P
H'EE031	FLMCR2	8	FLER	—	—	—	—	—	—	—
H'EE032	EBR1	8	EB7	EB6	EB5	EB4	EB3	EB2	EB1	EB0
H'EE033	EBR2	8	—	—	EB13	EB12	EB11	EB10	EB9	EB8
H'EE034	Reserved area (access prohibited)									
H'EE035										
H'EE036										
H'EE037										
H'EE038										
H'EE039										
H'EE03A										
H'EE03B										
H'EE03C	P2PCR	8	$P2_7PCR$	$P2_6PCR$	$P2_5PCR$	$P2_4PCR$	$P2_3PCR$	$P2_2PCR$	$P2_7PCR$	$P2_0PCR$
H'EE03D	—		—	—	—	—	—	—	—	—
H'EE03E	P4PCR	8	$P4_7PCR$	$P4_6PCR$	$P4_5PCR$	$P4_4PCR$	$P4_3PCR$	$P4_2PCR$	$P4_1PCR$	$P4_0PCR$
H'EE03F	P5PCR	8	—	—	—	—	$P5_3PCR$	$P5_2PCR$	$P5_1PCR$	$P5_0PCR$

H'EE046	—	—	—	—	—	—	—	—
H'EE047	—	—	—	—	—	—	—	—
H'EE048	—	—	—	—	—	—	—	—
H'EE049	—	—	—	—	—	—	—	—
H'EE04A	—	—	—	—	—	—	—	—
H'EE04B	—	—	—	—	—	—	—	—
H'EE04C	—	—	—	—	—	—	—	—
H'EE04D	—	—	—	—	—	—	—	—
H'EE04E	—	—	—	—	—	—	—	—
H'EE04F	—	—	—	—	—	—	—	—
H'EE050	—	—	—	—	—	—	—	—
H'EE051	—	—	—	—	—	—	—	—
H'EE052	—	—	—	—	—	—	—	—
H'EE053	—	—	—	—	—	—	—	—
H'EE054	—	—	—	—	—	—	—	—
H'EE055	—	—	—	—	—	—	—	—
H'EE056	—	—	—	—	—	—	—	—
H'EE057	—	—	—	—	—	—	—	—
H'EE058	—	—	—	—	—	—	—	—
H'EE059	—	—	—	—	—	—	—	—
H'EE05A	—	—	—	—	—	—	—	—
H'EE05B	—	—	—	—	—	—	—	—
H'EE05C	—	—	—	—	—	—	—	—
H'EE05D	—	—	—	—	—	—	—	—
H'EE05E	—	—	—	—	—	—	—	—
H'EE05F	—	—	—	—	—	—	—	—

H'EE066	—	—	—	—	—	—	—	—	—
H'EE067	—	—	—	—	—	—	—	—	—
H'EE068	—	—	—	—	—	—	—	—	—
H'EE069	—	—	—	—	—	—	—	—	—
H'EE06A	—	—	—	—	—	—	—	—	—
H'EE06B	—	—	—	—	—	—	—	—	—
H'EE06C	—	—	—	—	—	—	—	—	—
H'EE06D	—	—	—	—	—	—	—	—	—
H'EE06E	—	—	—	—	—	—	—	—	—
H'EE06F	—	—	—	—	—	—	—	—	—
H'EE070	—	—	—	—	—	—	—	—	—
H'EE071	—	—	—	—	—	—	—	—	—
H'EE072	—	—	—	—	—	—	—	—	—
H'EE073	—	—	—	—	—	—	—	—	—
H'EE074	Reserved area (access prohibited)								
H'EE075									
H'EE076									
H'EE077	RAMCR	8	—	—	—	—	RAMS	RAM2	RAM1	RAM0
H'EE078	Reserved area (access prohibited)								
H'EE079									
H'EE07A									
H'EE07B									
H'EE07C									
H'EE07D									
H'EE07E									
H'EE07F									

H'FFF25	ETCR0AL	8								
H'FFF26	IOAR0A	8								
H'FFF27	DTCR0A	8	DTE	DTSZ	DTID	RPE	DTIE	DTS2	DTS1	DTS0	Short a
			DTE	DTSZ	SAID	SAIDE	DTIE	DTS2A	DTS1A	DTS0A	Full add
H'FFF28	MAR0BR	8									DMAC
H'FFF29	MAR0BE	8									
H'FFF2A	MAR0BH	8									
H'FFF2B	MAR0BL	8									
H'FFF2C	ETCR0BH	8									
H'FFF2D	ETCR0BL	8									
H'FFF2E	IOAR0B	8									
H'FFF2F	DTCR0B	8	DTE	DTSZ	DTID	RPE	DTIE	DTS2	DTS1	DTS0	Short a
			DTME	—	DAID	DAIDE	TMS	DTS2B	DTS1B	DTS0B	Full add
H'FFF30	MAR1AR	8									DMAC
H'FFF31	MAR1AE	8									
H'FFF32	MAR1AH	8									
H'FFF33	MAR1AL	8									
H'FFF34	ETCR1AH	8									
H'FFF35	ETCR1AL	8									
H'FFF36	IOAR1A	8									
H'FFF37	DTCR1A	8	DTE	DTSZ	DTID	RPE	DTIE	DTS2	DTS1	DTS0	Short a
			DTE	DTSZ	SAID	SAIDE	DTIE	DTS2A	DTS1A	DTS0A	Full add
H'FFF38	MAR1BR	8									DMAC
H'FFF39	MAR1BE	8									
H'FFF3A	MAR1BH	8									
H'FFF3B	MAR1BL	8									
H'FFF3C	ETCR1BH	8									
H'FFF3D	ETCR1BL	8									
H'FFF3E	IOAR1B	8									
H'FFF3F	DTCR1B	8	DTE	DTSZ	DTID	RPE	DTIE	DTS2	DTS1	DTS0	Short a
			DTME	—	DAID	DAIDE	TMS	DTS2B	DTS1B	DTS0B	Full add

H'FFF46	—	—	—	—	—	—	—	—
H'FFF47	—	—	—	—	—	—	—	—
H'FFF48	—	—	—	—	—	—	—	—
H'FFF49	—	—	—	—	—	—	—	—
H'FFF4A	—	—	—	—	—	—	—	—
H'FFF4B	—	—	—	—	—	—	—	—
H'FFF4C	—	—	—	—	—	—	—	—
H'FFF4D	—	—	—	—	—	—	—	—
H'FFF4E	—	—	—	—	—	—	—	—
H'FFF4F	—	—	—	—	—	—	—	—
H'FFF50	—	—	—	—	—	—	—	—
H'FFF51	—	—	—	—	—	—	—	—
H'FFF52	—	—	—	—	—	—	—	—
H'FFF53	—	—	—	—	—	—	—	—
H'FFF54	—	—	—	—	—	—	—	—
H'FFF55	—	—	—	—	—	—	—	—
H'FFF56	—	—	—	—	—	—	—	—
H'FFF57	—	—	—	—	—	—	—	—
H'FFF58	—	—	—	—	—	—	—	—
H'FFF59	—	—	—	—	—	—	—	—
H'FFF5A	—	—	—	—	—	—	—	—
H'FFF5B	—	—	—	—	—	—	—	—
H'FFF5C	—	—	—	—	—	—	—	—
H'FFF5D	—	—	—	—	—	—	—	—
H'FFF5E	—	—	—	—	—	—	—	—
H'FFF5F	—	—	—	—	—	—	—	—

H'FFF66	TISRC	8	—	OVIE2	OVIE1	OVIE0	—	OVF2	OVF1	OVF0
H'FFF67										
H'FFF68	TCR0	8	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
H'FFF69	TIOR0	8	—	IOB2	IOB1	IOB0	—	IOA2	IOA1	IOA0
H'FFF6A	TCNT0H	16								
H'FFF6B	TCNT0L									
H'FFF6C	GRA0H	16								
H'FFF6D	GRA0L									
H'FFF6E	GRB0H	16								
H'FFF6F	GRB0L									
H'FFF70	16TCR1	8	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
H'FFF71	TIOR1	8	—	IOB2	IOB1	IOB0	—	IOA2	IOA1	IOA0
H'FFF72	16TCNT1H	16								
H'FFF73	16TCNT1L									
H'FFF74	GRA1H	16								
H'FFF75	GRA1L									
H'FFF76	GRB1H	16								
H'FFF77	GRB1L									
H'FFF78	16TCR2	8	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
H'FFF79	TIOR2	8	—	IOB2	IOB1	IOB0	—	IOA2	IOA1	IOA0
H'FFF7A	16TCNT2H	16								
H'FFF7B	16TCNT2L									
H'FFF7C	GRA2H	16								
H'FFF7D	GRA2L									
H'FFF7E	GRB2H	16								
H'FFF7F	GRB2L									

H'FFF87	TCORB1	8								
H'FFF88	8TCNT0	8								
H'FFF89	8TCNT1	8								
H'FFF8A	—		—	—	—	—	—	—	—	—
H'FFF8B	—		—	—	—	—	—	—	—	—
H'FFF8C	TCSR*2	8	OVF	WT/$\overline{IT}$	TME	—	—	CKS2	CKS1	CKS0
H'FFF8D	TCNT*2	8								
H'FFF8E	—		—	—	—	—	—	—	—	—
H'FFF8F	RSTCSR*2	8	WRST	RSTOE	—	—	—	—	—	—
H'FFF90	8TCR2	8	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0
H'FFF91	8TCR3	8	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0
H'FFF92	8TCSR2	8	CMFB	CMFA	OVF	—	OIS3	OIS2	OS1	OS0
H'FFF93	8TCSR3	8	CMFB	CMFA	OVF	ICE	OIS3	OIS2	OS1	OS0
H'FFF94	TCORA2	8								
H'FFF95	TCORA3	8								
H'FFF96	TCORB2	8								
H'FFF97	TCORB3	8								
H'FFF98	8TCNT2	8								
H'FFF99	8TCNT3	8								
H'FFF9A	—		—	—	—	—	—	—	—	—
H'FFF9B	—		—	—	—	—	—	—	—	—
H'FFF9C	DADR0	8								
H'FFF9D	DADR1	8								
H'FFF9E	DACR	8	DAOE1	DAOE0	DAE	—	—	—	—	—
H'FFF9F	—	8	—	—	—	—	—	—	—	—

H'FFFA6	NDRB*3	8	—	—	—	—	—	—	—	—
			—	—	—	—	NDER11	NDER10	NDER9	NDE
H'FFFA7	NDRA*3	8	—	—	—	—	—	—	—	—
			—	—	—	—	NDER3	NDER2	NDER1	NDE
H'FFFA8										
H'FFFA9										
H'FFFAA										
H'FFFAB										
H'FFFAC										
H'FFFAD										
H'FFFAE										
H'FFFAF										
H'FFFB0	SMR	8	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS
H'FFFB1	BRR	8								
H'FFFB2	SCR	8	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE
H'FFFB3	TDR	8								
H'FFFB4	SSR	8	TDRE	RDRF	ORER	FER/ERS	PER	TEND	MPB	MPB
H'FFFB5	RDR	8								
H'FFFB6	SCMR	8	—	—	—	—	SDIR	SINV	—	SMIF
H'FFFB7	Reserved area (access prohibited)									
H'FFFB8	SMR	8	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS
H'FFFB9	BRR	8								
H'FFFBA	SCR	8	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE
H'FFFBB	TDR	8								
H'FFFBC	SSR	8	TDRE	RDRF	ORER	FER/ERS	PER	TEND	MPB	MPB
H'FFFBD	RDR	8								
H'FFFBE	SCMR	8	—	—	—	—	SDIR	SINV	—	SMIF
H'FFFBF	Reserved area (access prohibited)									

H'FFFC6	SCMR	8	—	—	—	—	SDIR	SINV	—	SMIF
H'FFFC7	Reserved area (access prohibited)									
H'FFFC8	—		—	—	—	—	—	—	—	—
H'FFFC9	—		—	—	—	—	—	—	—	—
H'FFFCA	—		—	—	—	—	—	—	—	—
H'FFFCB	—		—	—	—	—	—	—	—	—
H'FFFCC	—		—	—	—	—	—	—	—	—
H'FFFCD	—		—	—	—	—	—	—	—	—
H'FFFCE	—		—	—	—	—	—	—	—	—
H'FFFCF	—		—	—	—	—	—	—	—	—
H'FFFD0	P1DR	8	$P1_7$	$P1_6$	$P1_5$	$P1_4$	$P1_3$	$P1_2$	$P1_1$	$P1_0$
H'FFFD1	P2DR	8	$P2_7$	$P2_6$	$P2_5$	$P2_4$	$P2_3$	$P2_2$	$P2_1$	$P2_0$
H'FFFD2	P3DR	8	$P3_7$	$P3_6$	$P3_5$	$P3_4$	$P3_3$	$P3_2$	$P3_1$	$P3_0$
H'FFFD3	P4DR	8	$P4_7$	$P4_6$	$P4_5$	$P4_4$	$P4_3$	$P4_2$	$P4_1$	$P4_0$
H'FFFD4	P5DR	8	—	—	—	—	$P5_3$	$P5_2$	$P5_1$	$P5_0$
H'FFFD5	P6DR	8	$P6_7$	$P6_6$	$P6_5$	$P6_4$	$P6_3$	$P6_2$	$P6_1$	$P6_0$
H'FFFD6	P7DR	8	$P7_7$	$P7_6$	$P7_5$	$P7_4$	$P7_3$	$P7_2$	$P7_1$	$P7_0$
H'FFFD7	P8DR	8	—	—	—	$P8_4$	$P8_3$	$P8_2$	$P8_1$	$P8_0$
H'FFFD8	P9DR	8	—	—	$P9_5$	$P9_4$	$P9_3$	$P9_2$	$P9_1$	$P9_0$
H'FFFD9	PADR	8	PA_7	PA_6	PA_5	PA_4	PA_3	PA_2	PA_1	PA_0
H'FFFDA	PBDR	8	PB_7	PB_6	PB_5	PB_4	PB_3	PB_2	PB_1	PB_0
H'FFFDB	—		—	—	—	—	—	—	—	—
H'FFFDC	—		—	—	—	—	—	—	—	—
H'FFFDD	—		—	—	—	—	—	—	—	—
H'FFFDE	—		—	—	—	—	—	—	—	—
H'FFFDF	—		—	—	—	—	—	—	—	—

H'FFFE6	ADDRDH	8	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2
H'FFFE7	ADDRDL	8	AD1	AD0	—	—	—	—	—	—
H'FFFE8	ADCSR	8	ADF	ADIE	ADST	SCAN	CKS	CH2	CH1	CH0
H'FFFE9	ADCR	8	TRGE	—	—	—	—	—	—	—

Notes: 1. These registers are only used by the flash memory version, and are not prov
mask ROM versions.
2. For write access to TCSR, TCNT, and RSTCSR, see section 12.2.4, Notes o
Access.
3. The address depends on the output trigger setting.

Legend

WDT: Watchdog timer

TPC: Programmable timing pattern controller

SCI: Serial communication interface

H'EE004	P5DDR	8	—	—	—	—	$P5_3$DDR	$P5_2$DDR	$P5_1$DDR	$P5_0$DD
H'EE005	P6DDR	8	—	$P6_6$DDR	$P6_5$DDR	$P6_4$DDR	$P6_3$DDR	$P6_2$DDR	$P6_1$DDR	$P6_0$DD
H'EE006	—	—	—	—	—	—	—	—	—	—
H'EE007	P8DDR	8	—	—	—	$P8_4$DDR	$P8_3$DDR	$P8_2$DDR	$P8_1$DDR	$P8_0$DD
H'EE008	P9DDR	8	—	—	$P9_5$DDR	$P9_4$DDR	$P9_3$DDR	$P9_2$DDR	$P9_1$DDR	$P9_0$DD
H'EE009	PADDR	8	PA_7DDR	PA_6DDR	PA_5DDR	PA_4DDR	PA_3DDR	PA_2DDR	PA_1DDR	PA_0DD
H'EE00A	PBDDR	8	PB_7DDR	PB_6DDR	PB_5DDR	PB_4DDR	PB_3DDR	PB_2DDR	PB_1DDR	PB_0DD
H'EE00B	—		—	—	—	—	—	—	—	—
H'EE00C	—		—	—	—	—	—	—	—	—
H'EE00D	—		—	—	—	—	—	—	—	—
H'EE00E	—		—	—	—	—	—	—	—	—
H'EE00F	—		—	—	—	—	—	—	—	—
H'EE010	—		—	—	—	—	—	—	—	—
H'EE011	MDCR	8	—	—	—	—	—	MDS2	MDS1	MDS0
H'EE012	SYSCR	8	SSBY	STS2	STS1	STS0	UE	NMIEG	SSOE	RAME
H'EE013	BRCR	8	A23E	A22E	A21E	A20E	—	—	—	BRLE
H'EE014	ISCR	8	—	—	IRQ5SC	IRQ4SC	IRQ3SC	IRQ2SC	IRQ1SC	IRQ0S
H'EE015	IER	8	—	—	IRQ5E	IRQ4E	IRQ3E	IRQ2E	IRQ1E	IRQ0E
H'EE016	ISR	8	—	—	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	IRQ0F
H'EE017	—	8	—	—	—	—	—	—	—	—
H'EE018	IPRA	8	IPRA7	IPRA6	IPRA5	IPRA4	IPRA3	IPRA2	IPRA1	IPRA0
H'EE019	IPRB	8	IPRB7	IPRB6	IPRB5	—	IPRB3	IPRB2	IPRB1	—
H'EE01A	DASTCR	8	—	—	—	—	—	—	—	DASTE
H'EE01B	DIVCR	8	—	—	—	—	—	—	DIV1	DIV0
H'EE01C	MSTCRH	8	PSTOP	—	—	—	—	MSTPH2	MSTPH1	MSTPI
H'EE01D	MSTCRL	8	MSTPL7	—	MSTPL5	MSTPL4	MSTPL3	MSTPL2	—	MSTPI
H'EE01E	ADRCR	8	—	—	—	—	—	—	—	ADRC
H'EE01F	CSCR	8	CS7E	CS6E	CS5E	CS4E	—	—	—	—

H'EE027	DRCRB	8	MXC1	MXC0	CSEL	RCYCE	—	TPC	RCW	RLW
H'EE028	RTMCSR	8	CMF	CMIE	CKS2	CKS1	CKS0	—	—	—
H'EE029	RTCNT	8								
H'EE02A	RTCOR	8								
H'EE02B	—	8								
H'EE02C	DCR0	8								
H'EE02D	DCR1	8								
H'EE02E	DCR2	8								
H'EE02F	DCR3	8								
H'EE030	FLMCR1	8	FWE	SWE	ESU	PSU	EV	PV	E	P
H'EE031	FLMCR2	8	FLER							
H'EE032	EBR1	8	EB7	EB6	EB5	EB4	EB3	EB2	EB1	EB0
H'EE033	EBR2	8	—	—	EB13	EB12	EB11	EB10	EB9	EB8
H'EE034	Reserved area (access prohibited)									
H'EE035										
H'EE036										
H'EE037										
H'EE03C	P2PCR	8	$P2_7PCR$	$P2_6PCR$	$P2_5PCR$	$P2_4PCR$	$P2_3PCR$	$P2_2PCR$	$P2_7PCR$	$P2_0PCR$
H'EE03D	—	8	—	—	—	—	—	—	—	—
H'EE03E	P4PCR	8	$P4_7PCR$	$P4_6PCR$	$P4_5PCR$	$P4_4PCR$	$P4_3PCR$	$P4_2PCR$	$P4_1PCR$	$P4_0PCR$
H'EE03F	P5PCR	8	—	—	—	—	$P5_3PCR$	$P5_2PCR$	$P5_1PCR$	$P5_0PCR$

H'EE047	—	—	—	—	—	—	—	—	—
H'EE048	—	—	—	—	—	—	—	—	—
H'EE049	—	—	—	—	—	—	—	—	—
H'EE04A	—	—	—	—	—	—	—	—	—
H'EE04B	—	—	—	—	—	—	—	—	—
H'EE04C	—	—	—	—	—	—	—	—	—
H'EE04D	—	—	—	—	—	—	—	—	—
H'EE04E	—	—	—	—	—	—	—	—	—
H'EE04F	—	—	—	—	—	—	—	—	—
H'EE050	—	—	—	—	—	—	—	—	—
H'EE051	—	—	—	—	—	—	—	—	—
H'EE052	—	—	—	—	—	—	—	—	—
H'EE053	—	—	—	—	—	—	—	—	—
H'EE054	—	—	—	—	—	—	—	—	—
H'EE055	—	—	—	—	—	—	—	—	—
H'EE056	—	—	—	—	—	—	—	—	—
H'EE057	—	—	—	—	—	—	—	—	—
H'EE058	—	—	—	—	—	—	—	—	—
H'EE059	—	—	—	—	—	—	—	—	—
H'EE05A	—	—	—	—	—	—	—	—	—
H'EE05B	—	—	—	—	—	—	—	—	—
H'EE05C	—	—	—	—	—	—	—	—	—
H'EE05D	—	—	—	—	—	—	—	—	—
H'EE05E	—	—	—	—	—	—	—	—	—
H'EE05F	—	—	—	—	—	—	—	—	—

H'EE067	—		—	—	—	—	—	—	—	—
H'EE068	—		—	—	—	—	—	—	—	—
H'EE069	—		—	—	—	—	—	—	—	—
H'EE06A	—		—	—	—	—	—	—	—	—
H'EE06B	—		—	—	—	—	—	—	—	—
H'EE06C	—		—	—	—	—	—	—	—	—
H'EE06D	—		—	—	—	—	—	—	—	—
H'EE06E	—		—	—	—	—	—	—	—	—
H'EE06F	—		—	—	—	—	—	—	—	—
H'EE070	Reserved area (access prohibited)									
H'EE071										
H'EE072										
H'EE073										
H'EE074										
H'EE075										
H'EE076										
H'EE077	RAMCR	8	—	—	—	—	RAMS	RAM2	RAM1	RAM0
H'EE078	Reserved area (access prohibited)									
H'EE079										
H'EE07A										
H'EE07B										
H'EE07C										
H'EE07D										
H'EE07E										
H'EE07F										

H'EE097
H'EE098
H'EE099
H'EE09A
H'EE09B
H'EE09C
H'EE09D
H'EE09E
H'EE09F
H'EE0A0
H'EE0A1
H'EE0A2
H'EE0A3
H'EE0A4
H'EE0A5
H'EE0A6
H'EE0A7
H'EE0A8
H'EE0A9
H'EE0AA
H'EE0AB
H'EE0AC
H'EE0AD
H'EE0AE
H'EE0AF

H'EE0B7
H'EE0B8
H'EE0B9
H'EE0BA
H'EE0BB
H'EE0BC
H'EE0BD
H'EE0BE
H'EE0BF
H'EE0C0
H'EE0C1
H'EE0C2
H'EE0C3
H'EE0C4
H'EE0C5
H'EE0C6
H'EE0C7
H'EE0C8
H'EE0C9
H'EE0CA
H'EE0CB
H'EE0CC
H'EE0CD
H'EE0CE
H'EE0CF

H'EE0D7
H'EE0D8
H'EE0D9
H'EE0DA
H'EE0DB
H'EE0DC
H'EE0DD
H'EE0DE
H'EE0DF
H'EE0E0
H'EE0E1
H'EE0E2
H'EE0E3
H'EE0E4
H'EE0E5
H'EE0E6
H'EE0E7
H'EE0E8
H'EE0E9
H'EE0EA
H'EE0EB
H'EE0EC
H'EE0ED
H'EE0EE
H'EE0EF

H'EE0F7									
H'EE0F8									
H'EE0F9									
H'EE0FA									
H'EE0FB									
H'EE0FC									
H'EE0FD									
H'EE0FE									
H'EE0FF									
H'FFE00	—	—	—	—	—	—	—	—	—
H'FFE01	—	—	—	—	—	—	—	—	—
H'FFE02	—	—	—	—	—	—	—	—	—
H'FFE03	—	—	—	—	—	—	—	—	—
H'FFE04	—	—	—	—	—	—	—	—	—
H'FFE05	—	—	—	—	—	—	—	—	—
H'FFE06	—	—	—	—	—	—	—	—	—
H'FFE07	—	—	—	—	—	—	—	—	—
H'FFE08	—	—	—	—	—	—	—	—	—
H'FFE09	—	—	—	—	—	—	—	—	—
H'FFE0A	—	—	—	—	—	—	—	—	—
H'FFE0B	—	—	—	—	—	—	—	—	—
H'FFE0C	—	—	—	—	—	—	—	—	—
H'FFE0D	—	—	—	—	—	—	—	—	—
H'FFE0E	—	—	—	—	—	—	—	—	—
H'FFE0F	—	—	—	—	—	—	—	—	—

H'FFE17	—	—	—	—	—	—	—	—	—
H'FFE18	—	—	—	—	—	—	—	—	—
H'FFE19	—	—	—	—	—	—	—	—	—
H'FFE1A	—	—	—	—	—	—	—	—	—
H'FFE1B	—	—	—	—	—	—	—	—	—
H'FFE1C	—	—	—	—	—	—	—	—	—
H'FFE1D	—	—	—	—	—	—	—	—	—
H'FFE1E	—	—	—	—	—	—	—	—	—
H'FFE1F	—	—	—	—	—	—	—	—	—
H'FFE20	—	—	—	—	—	—	—	—	—
H'FFE21	—	—	—	—	—	—	—	—	—
H'FFE22	—	—	—	—	—	—	—	—	—
H'FFE23	—	—	—	—	—	—	—	—	—
H'FFE24	—	—	—	—	—	—	—	—	—
H'FFE25	—	—	—	—	—	—	—	—	—
H'FFE26	—	—	—	—	—	—	—	—	—
H'FFE27	—	—	—	—	—	—	—	—	—
H'FFE28	—	—	—	—	—	—	—	—	—
H'FFE29	—	—	—	—	—	—	—	—	—
H'FFE2A	—	—	—	—	—	—	—	—	—
H'FFE2B	—	—	—	—	—	—	—	—	—
H'FFE2C	—	—	—	—	—	—	—	—	—
H'FFE2D	—	—	—	—	—	—	—	—	—
H'FFE2E	—	—	—	—	—	—	—	—	—
H'FFE2F	—	—	—	—	—	—	—	—	—

H'FFE37	—	—	—	—	—	—	—	—
H'FFE38	—	—	—	—	—	—	—	—
H'FFE39	—	—	—	—	—	—	—	—
H'FFE3A	—	—	—	—	—	—	—	—
H'FFE3B	—	—	—	—	—	—	—	—
H'FFE3C	—	—	—	—	—	—	—	—
H'FFE3D	—	—	—	—	—	—	—	—
H'FFE3E	—	—	—	—	—	—	—	—
H'FFE3F	—	—	—	—	—	—	—	—
H'FFE40	—	—	—	—	—	—	—	—
H'FFE41	—	—	—	—	—	—	—	—
H'FFE42	—	—	—	—	—	—	—	—
H'FFE43	—	—	—	—	—	—	—	—
H'FFE44	—	—	—	—	—	—	—	—
H'FFE45	—	—	—	—	—	—	—	—
H'FFE46	—	—	—	—	—	—	—	—
H'FFE47	—	—	—	—	—	—	—	—
H'FFE48	—	—	—	—	—	—	—	—
H'FFE49	—	—	—	—	—	—	—	—
H'FFE4A	—	—	—	—	—	—	—	—
H'FFE4B	—	—	—	—	—	—	—	—
H'FFE4C	—	—	—	—	—	—	—	—
H'FFE4D	—	—	—	—	—	—	—	—
H'FFE4E	—	—	—	—	—	—	—	—
H'FFE4F	—	—	—	—	—	—	—	—

H'FFE57	—	—	—	—	—	—	—	—	—
H'FFE58	—	—	—	—	—	—	—	—	—
H'FFE59	—	—	—	—	—	—	—	—	—
H'FFE5A	—	—	—	—	—	—	—	—	—
H'FFE5B	—	—	—	—	—	—	—	—	—
H'FFE5C	—	—	—	—	—	—	—	—	—
H'FFE5D	—	—	—	—	—	—	—	—	—
H'FFE5E	—	—	—	—	—	—	—	—	—
H'FFE5F	—	—	—	—	—	—	—	—	—
H'FFE60	—	—	—	—	—	—	—	—	—
H'FFE61	—	—	—	—	—	—	—	—	—
H'FFE62	—	—	—	—	—	—	—	—	—
H'FFE63	—	—	—	—	—	—	—	—	—
H'FFE64	—	—	—	—	—	—	—	—	—
H'FFE65	—	—	—	—	—	—	—	—	—
H'FFE66	—	—	—	—	—	—	—	—	—
H'FFE67	—	—	—	—	—	—	—	—	—
H'FFE68	—	—	—	—	—	—	—	—	—
H'FFE69	—	—	—	—	—	—	—	—	—
H'FFE6A	—	—	—	—	—	—	—	—	—
H'FFE6B	—	—	—	—	—	—	—	—	—
H'FFE6C	—	—	—	—	—	—	—	—	—
H'FFE6D	—	—	—	—	—	—	—	—	—
H'FFE6E	—	—	—	—	—	—	—	—	—
H'FFE6F	—	—	—	—	—	—	—	—	—

H'FFE77	—		—	—	—	—	—	—	—	—
H'FFE78	—		—	—	—	—	—	—	—	—
H'FFE79	—		—	—	—	—	—	—	—	—
H'FFE7A	—		—	—	—	—	—	—	—	—
H'FFE7B	—		—	—	—	—	—	—	—	—
H'FFE7C	—		—	—	—	—	—	—	—	—
H'FFE7D	—		—	—	—	—	—	—	—	—
H'FFE7E	—		—	—	—	—	—	—	—	—
H'FFE7F	—		—	—	—	—	—	—	—	—
H'FFE80	MAR0AR	8								
H'FFE81	MAR0AE	8								
H'FFE82	MAR0AH	8								
H'FFE83	MAR0AL	8								
H'FFE84	ETCR0AH	8								
H'FFE85	ETCR0AL	8								
H'FFE86	IOAR0A	8								
H'FFE87	DTCR0A	8	DTE	DTSZ	DTID	RPE	DTIE	DTS2	DTS1	DTS0
			DTE	DTSZ	SAID	SAIDE	DTIE	DTS2A	DTS1A	DTS0A
H'FFE88	MAR0BR	8								
H'FFE89	MAR0BE	8								
H'FFE8A	MAR0BH	8								
H'FFE8B	MAR0BL	8								
H'FFE8C	ETCR0BH	8								
H'FFE8D	ETCR0BL	8								
H'FFE8E	IOAR0B	8								
H'FFE8F	DTCR0B	8	DTE	DTSZ	DTID	RPE	DTIE	DTS2	DTS1	DTS0
			DTME	—	DAID	DAIDE	TMS	DTS2B	DTS1B	DTS0B

H'FFE97	DTCR1A	8	DTE	DTSZ	DTID	RPE	DTIE	DTS2	DTS1	DTS0
			DTE	DTSZ	SAID	SAIDE	DTIE	DTS2A	DTS1A	DTS0A
H'FFE98	MAR1BR	8								
H'FFE99	MAR1BE	8								
H'FFE9A	MAR1BH	8								
H'FFE9B	MAR1BL	8								
H'FFE9C	ETCR1BH	8								
H'FFE9D	ETCR1BL	8								
H'FFE9E	IOAR1B	8								
H'FFE9F	DTCR1B	8	DTE	DTSZ	DTID	RPE	DTIE	DTS2	DTS1	DTS0
			DTME	—	DAID	DAIDE	TMS	DTS2B	DTS1B	DTS0B
H'FFEA0	TSTR	8	—	—	—	—	—	STR2	STR1	STR0
H'FFEA1	TSNC	8	—	—	—	—	—	SYNC2	SYNC1	SYNC0
H'FFEA2	TMDR	8	—	MDF	FDIR	—	—	PWM2	PWM1	PWM0
H'FFEA3	TOLR	8	—	—	TOB2	TOA2	TOB1	TOA1	TOB0	TOA0
H'FFEA4	TISRA	8	—	IMIEA2	IMIEA1	IMIEA0	—	IMFA2	IMFA1	IMFA0
H'FFEA5	TISRB	8	—	IMIEB2	IMIEB1	IMIEB0	—	IMFB2	IMFB1	IMFB0
H'FFEA6	TISRC	8	—	OVIE2	OVIE1	OVIE0	—	OVF2	OVF1	OVF0
H'FFEA7	—									
H'FFEA8	16TCR0	8	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
H'FFEA9	TIOR0	8	—	IOB2	IOB1	IOB0	—	IOA2	IOA1	IOA0
H'FFEAA	16TCNT0H	16								
H'FFEAB	16TCNT0L									
H'FFEAC	GRA0H	16								
H'FFEAD	GRA0L									
H'FFEAE	GRB0H	16								
H'FFEAF	GRB0L									

H'FFEB7	GRB1L									
H'FFEB8	16TCR2	8	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
H'FFEB9	TIOR2	8	—	IOB2	IOB1	IOB0	—	IOA2	IOA1	IOA0
H'FFEBA	16TCNT2H	16								
H'FFEBB	16TCNT2L									
H'FFEBC	GRA2H	16								
H'FFEBD	GRA2L									
H'FFEBE	GRB2H	16								
H'FFEBF	GRB2L									
H'FFEC0	8TCR0	8	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0
H'FFEC1	8TCR1	8	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0
H'FFEC2	8TCSR0	8	CMFB	CMFA	OVF	ADTE	OIS3	OIS2	OS1	OS0
H'FFEC3	8TCSR1	8	CMFB	CMFA	OVF	ICE	OIS3	OIS2	OS1	OS0
H'FFEC4	TCORA0	8								
H'FFEC5	TCORA1	8								
H'FFEC6	TCORB0	8								
H'FFEC7	TCORB1	8								
H'FFEC8	8TCNT0	8								
H'FFEC9	8TCNT1	8								
H'FFECA	Reserved area (access prohibited)									
H'FFECB										
H'FFECC										
H'FFECD										
H'FFECE										
H'FFECF										

H'FFED7	TCORB3	8								
H'FFED8	8TCNT2	8								
H'FFED9	8TCNT3	8								
H'FFEDA	Reserved area (access prohibited)									
H'FFEDB										
H'FFEDC										
H'FFEDD										
H'FFEDE										
H'FFEDF										
H'FFEE0	SMR	8	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0
H'FFEE1	BRR	8								
H'FFEE2	SCR	8	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
H'FFEE3	TDR	8								
H'FFEE4	SSR	8	TDRE	RDRF	ORER	FER/ ERS	PER	TEND	MPB	MPBT
H'FFEE5	RDR	8								
H'FFEE6	SCMR	8	—	—	—	—	SDIR	SINV	—	SMIF
H'FFEE7	—		—	—	—	—	—	—	—	—
H'FFEE8	SMR	8	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0
H'FFEE9	BRR	8								
H'FFEEA	SCR	8	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
H'FFEEB	TDR	8								
H'FFEEC	SSR	8	TDRE	RDRF	ORER	FER/ ERS	PER	TEND	MPB	MPBT
H'FFEED	RDR	8								
H'FFEEE	SCMR	8	—	—	—	—	SDIR	SINV	—	SMIF
H'FFEEF	—		—	—	—	—	—	—	—	—

H'FFEF6	SCMR	8	—	—	—	—	SDIR	SINV	—	SMIF
H'FFEF7	—		—	—	—	—	—	—	—	—
H'FFEF8	TPMR	8	—	—	—	—	G3NOV	G2NOV	G1NOV	G0NOV
H'FFEF9	TPCR	8	G3CMS1	G3CMS0	G2CMS1	G2CMS0	G1CMS1	G1CMS0	G0CMS1	G0CMS
H'FFEFA	NDERB	8	NDER15	NDER14	NDER13	NDER12	NDER11	NDER10	NDER9	NDER8
H'FFEFB	NDERA	8	NDER7	NDER6	NDER5	NDER4	NDER3	NDER2	NDER1	NDER0
H'FFEFC	NDRB*3	8	NDR15	NDR14	NDR13	NDR12	NDR11	NDR10	NDR9	NDR8
			NDR15	NDR14	NDR13	NDR12	—	—	—	—
H'FFEFD	NDRA*3	8	NDR7	NDR6	NDR5	NDR4	NDR3	NDR2	NDR1	NDR0
			NDR7	NDR6	NDR5	NDR4	—	—	—	—
H'FFEFE	NDRB*3	8	—	—	—	—	—	—	—	—
			—	—	—	—	NDR11	NDR10	NDR9	NDR8
H'FFEFF	NDRA*3	8	—	—	—	—	—	—	—	—
			—	—	—	—	NDR3	NDR2	NDR1	NDR0
H'FFF10	—		—	—	—	—	—	—	—	—
H'FFF11	—		—	—	—	—	—	—	—	—
H'FFF12	—		—	—	—	—	—	—	—	—
H'FFF13	—		—	—	—	—	—	—	—	—
H'FFF14	—		—	—	—	—	—	—	—	—
H'FFF15	—		—	—	—	—	—	—	—	—
H'FFF16	—		—	—	—	—	—	—	—	—
H'FFF17	—		—	—	—	—	—	—	—	—
H'FFF18	—		—	—	—	—	—	—	—	—
H'FFF19	—		—	—	—	—	—	—	—	—
H'FFF1A	—		—	—	—	—	—	—	—	—
H'FFF1B	—		—	—	—	—	—	—	—	—
H'FFF1C	—		—	—	—	—	—	—	—	—
H'FFF1D	—		—	—	—	—	—	—	—	—
H'FFF1E	—		—	—	—	—	—	—	—	—
H'FFF1F	—		—	—	—	—	—	—	—	—

H'FFF27	—	—	—	—	—	—	—	—	—
H'FFF28	—	—	—	—	—	—	—	—	—
H'FFF29	—	—	—	—	—	—	—	—	—
H'FFF2A	—	—	—	—	—	—	—	—	—
H'FFF2B	—	—	—	—	—	—	—	—	—
H'FFF2C	—	—	—	—	—	—	—	—	—
H'FFF2D	—	—	—	—	—	—	—	—	—
H'FFF2E	—	—	—	—	—	—	—	—	—
H'FFF2F	—	—	—	—	—	—	—	—	—
H'FFF30	—	—	—	—	—	—	—	—	—
H'FFF31	—	—	—	—	—	—	—	—	—
H'FFF32	—	—	—	—	—	—	—	—	—
H'FFF33	—	—	—	—	—	—	—	—	—
H'FFF34	—	—	—	—	—	—	—	—	—
H'FFF35	—	—	—	—	—	—	—	—	—
H'FFF36	—	—	—	—	—	—	—	—	—
H'FFF37	—	—	—	—	—	—	—	—	—
H'FFF38	—	—	—	—	—	—	—	—	—
H'FFF39	—	—	—	—	—	—	—	—	—
H'FFF3A	—	—	—	—	—	—	—	—	—
H'FFF3B	—	—	—	—	—	—	—	—	—
H'FFF3C	—	—	—	—	—	—	—	—	—
H'FFF3D	—	—	—	—	—	—	—	—	—
H'FFF3E	—	—	—	—	—	—	—	—	—
H'FFF3F	—	—	—	—	—	—	—	—	—

H'FFF47	—	—	—	—	—	—	—	—
H'FFF48	—	—	—	—	—	—	—	—
H'FFF49	—	—	—	—	—	—	—	—
H'FFF4A	—	—	—	—	—	—	—	—
H'FFF4B	—	—	—	—	—	—	—	—
H'FFF4C	—	—	—	—	—	—	—	—
H'FFF4D	—	—	—	—	—	—	—	—
H'FFF4E	—	—	—	—	—	—	—	—
H'FFF4F	—	—	—	—	—	—	—	—
H'FFF50	—	—	—	—	—	—	—	—
H'FFF51	—	—	—	—	—	—	—	—
H'FFF52	—	—	—	—	—	—	—	—
H'FFF53	—	—	—	—	—	—	—	—
H'FFF54	—	—	—	—	—	—	—	—
H'FFF55	—	—	—	—	—	—	—	—
H'FFF56	—	—	—	—	—	—	—	—
H'FFF57	—	—	—	—	—	—	—	—
H'FFF58	—	—	—	—	—	—	—	—
H'FFF59	—	—	—	—	—	—	—	—
H'FFF5A	—	—	—	—	—	—	—	—
H'FFF5B	—	—	—	—	—	—	—	—
H'FFF5C	—	—	—	—	—	—	—	—
H'FFF5D	—	—	—	—	—	—	—	—
H'FFF5E	—	—	—	—	—	—	—	—
H'FFF5F	—	—	—	—	—	—	—	—

H'FFF67	—	—	—	—	—	—	—	—	—
H'FFF68	—	—	—	—	—	—	—	—	—
H'FFF69	—	—	—	—	—	—	—	—	—
H'FFF6A	—	—	—	—	—	—	—	—	—
H'FFF6B	—	—	—	—	—	—	—	—	—
H'FFF6C	—	—	—	—	—	—	—	—	—
H'FFF6D	—	—	—	—	—	—	—	—	—
H'FFF6E	—	—	—	—	—	—	—	—	—
H'FFF6F	—	—	—	—	—	—	—	—	—
H'FFF70	—	—	—	—	—	—	—	—	—
H'FFF71	—	—	—	—	—	—	—	—	—
H'FFF72	—	—	—	—	—	—	—	—	—
H'FFF73	—	—	—	—	—	—	—	—	—
H'FFF74	—	—	—	—	—	—	—	—	—
H'FFF75	—	—	—	—	—	—	—	—	—
H'FFF76	—	—	—	—	—	—	—	—	—
H'FFF77	—	—	—	—	—	—	—	—	—
H'FFF78	—	—	—	—	—	—	—	—	—
H'FFF79	—	—	—	—	—	—	—	—	—
H'FFF7A	—	—	—	—	—	—	—	—	—
H'FFF7B	—	—	—	—	—	—	—	—	—
H'FFF7C	—	—	—	—	—	—	—	—	—
H'FFF7D	—	—	—	—	—	—	—	—	—
H'FFF7E	—	—	—	—	—	—	—	—	—
H'FFF7F	—	—	—	—	—	—	—	—	—

H'FFF87	—	—	—	—	—	—	—	—
H'FFF88	—	—	—	—	—	—	—	—
H'FFF89	—	—	—	—	—	—	—	—
H'FFF8A	—	—	—	—	—	—	—	—
H'FFF8B	—	—	—	—	—	—	—	—
H'FFF8C	—	—	—	—	—	—	—	—
H'FFF8D	—	—	—	—	—	—	—	—
H'FFF8E	—	—	—	—	—	—	—	—
H'FFF8F	—	—	—	—	—	—	—	—
H'FFF90	—	—	—	—	—	—	—	—
H'FFF91	—	—	—	—	—	—	—	—
H'FFF92	—	—	—	—	—	—	—	—
H'FFF93	—	—	—	—	—	—	—	—
H'FFF94	—	—	—	—	—	—	—	—
H'FFF95	—	—	—	—	—	—	—	—
H'FFF96	—	—	—	—	—	—	—	—
H'FFF97	—	—	—	—	—	—	—	—
H'FFF98	—	—	—	—	—	—	—	—
H'FFF99	—	—	—	—	—	—	—	—
H'FFF9A	—	—	—	—	—	—	—	—
H'FFF9B	—	—	—	—	—	—	—	—
H'FFF9C	—	—	—	—	—	—	—	—
H'FFF9D	—	—	—	—	—	—	—	—
H'FFF9E	—	—	—	—	—	—	—	—
H'FFF9F	—	—	—	—	—	—	—	—

H'FFFA7	—	—	—	—	—	—	—	—	—
H'FFFA8	—	—	—	—	—	—	—	—	—
H'FFFA9	—	—	—	—	—	—	—	—	—
H'FFFAA	—	—	—	—	—	—	—	—	—
H'FFFAB	—	—	—	—	—	—	—	—	—
H'FFFAC	—	—	—	—	—	—	—	—	—
H'FFFAD	—	—	—	—	—	—	—	—	—
H'FFFAE	—	—	—	—	—	—	—	—	—
H'FFFAF	—	—	—	—	—	—	—	—	—
H'FFFB0	—	—	—	—	—	—	—	—	—
H'FFFB1	—	—	—	—	—	—	—	—	—
H'FFFB2	—	—	—	—	—	—	—	—	—
H'FFFB3	—	—	—	—	—	—	—	—	—
H'FFFB4	—	—	—	—	—	—	—	—	—
H'FFFB5	—	—	—	—	—	—	—	—	—
H'FFFB6	—	—	—	—	—	—	—	—	—
H'FFFB7	—	—	—	—	—	—	—	—	—
H'FFFB8	—	—	—	—	—	—	—	—	—
H'FFFB9	—	—	—	—	—	—	—	—	—
H'FFFBA	—	—	—	—	—	—	—	—	—
H'FFFBB	—	—	—	—	—	—	—	—	—
H'FFFBC	—	—	—	—	—	—	—	—	—
H'FFFBD	—	—	—	—	—	—	—	—	—
H'FFFBE	—	—	—	—	—	—	—	—	—
H'FFFBF	—	—	—	—	—	—	—	—	—

H'FFFC7	—	—	—	—	—	—	—	—	—
H'FFFC8	—	—	—	—	—	—	—	—	—
H'FFFC9	—	—	—	—	—	—	—	—	—
H'FFFCA	—	—	—	—	—	—	—	—	—
H'FFFCB	—	—	—	—	—	—	—	—	—
H'FFFCC	—	—	—	—	—	—	—	—	—
H'FFFCD	—	—	—	—	—	—	—	—	—
H'FFFCE	—	—	—	—	—	—	—	—	—
H'FFFCF	—	—	—	—	—	—	—	—	—
H'FFFD0	—	—	—	—	—	—	—	—	—
H'FFFD1	—	—	—	—	—	—	—	—	—
H'FFFD2	—	—	—	—	—	—	—	—	—
H'FFFD3	—	—	—	—	—	—	—	—	—
H'FFFD4	—	—	—	—	—	—	—	—	—
H'FFFD5	—	—	—	—	—	—	—	—	—
H'FFFD6	—	—	—	—	—	—	—	—	—
H'FFFD7	—	—	—	—	—	—	—	—	—
H'FFFD8	—	—	—	—	—	—	—	—	—
H'FFFD9	—	—	—	—	—	—	—	—	—
H'FFFDA	—	—	—	—	—	—	—	—	—
H'FFFDB	—	—	—	—	—	—	—	—	—
H'FFFDC	—	—	—	—	—	—	—	—	—
H'FFFDD	—	—	—	—	—	—	—	—	—
H'FFFDE	—	—	—	—	—	—	—	—	—
H'FFFDF	—	—	—	—	—	—	—	—	—

H'FFFE7	ADDRDL	8	AD1	AD0	—	—	—	—	—	—
H'FFFE8	ADCSR	8	ADF	ADIE	ADST	SCAN	CKS	CH2	CH1	CH0
H'FFFE9	ADCR	8	TRGE	—	—	—	—	—	—	—
H'FFFEA	Reserved area (access prohibited)									
H'FFFEB										
H'FFFEC	DADR0	8								
H'FFFED	DADR1	8								
H'FFFEE	DACR	8	—	—	—	—	—	—	—	—
H'FFFEF	—	8	—	—	—	—	—	—	—	—
H'FFFF0	P1DR	8	$P1_7$	$P1_6$	$P1_5$	$P1_4$	$P1_3$	$P1_2$	$P1_1$	$P1_0$
H'FFFF1	P2DR	8	$P2_7$	$P2_6$	$P2_5$	$P2_4$	$P2_3$	$P2_2$	$P2_1$	$P2_0$
H'FFFF2	P3DR	8	$P3_7$	$P3_6$	$P3_5$	$P3_4$	$P3_3$	$P3_2$	$P3_1$	$P3_0$
H'FFFF3	P4DR	8	$P4_7$	$P4_6$	$P4_5$	$P4_4$	$P4_3$	$P4_2$	$P4_1$	$P4_0$
H'FFFF4	P5DR	8	—	—	—	—	$P5_3$	$P5_2$	$P5_1$	$P5_0$
H'FFFF5	P6DR	8	$P6_7$	$P6_6$	$P6_5$	$P6_4$	$P6_3$	$P6_2$	$P6_1$	$P6_0$
H'FFFF6	P7DR	8	$P7_7$	$P7_6$	$P7_5$	$P7_4$	$P7_3$	$P7_2$	$P7_1$	$P7_0$
H'FFFF7	P8DR	8	—	—	—	$P8_4$	$P8_3$	$P8_2$	$P8_1$	$P8_0$
H'FFFF8	P9DR	8	—	—	$P9_5$	$P9_4$	$P9_3$	$P9_2$	$P9_1$	$P9_0$
H'FFFF9	PADR	8	PA_7	PA_6	PA_5	PA_4	PA_3	PA_2	PA_1	PA_0
H'FFFFA	PBDR	8	PB_7	PB_6	PB_5	PB_4	PB_3	PB_2	PB_1	PB_0
H'FFFFB	—		—	—	—	—	—	—	—	—
H'FFFFC	—		—	—	—	—	—	—	—	—
H'FFFFD	—		—	—	—	—	—	—	—	—
H'FFFFE	—		—	—	—	—	—	—	—	—
H'FFFFF	—		—	—	—	—	—	—	—	—

Notes: 1. These registers are only used by the flash memory version, and are not pro
mask ROM versions.

2. For write access to TCSR, TCNT, and RSTCSR, see section 12.2.4, Notes
Access.

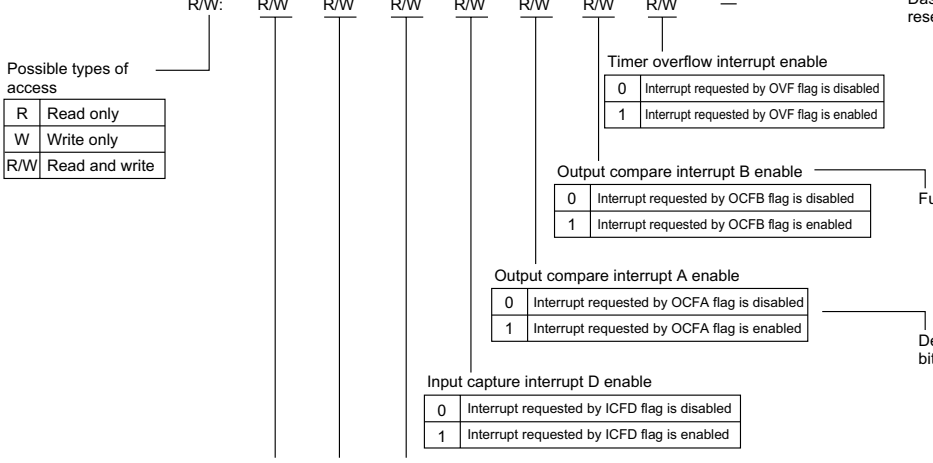
3. The address depends on the output trigger setting.

Legend

WDT: Watchdog timer

TPC: Programmable timing pattern controller

SCI: Serial communication interface

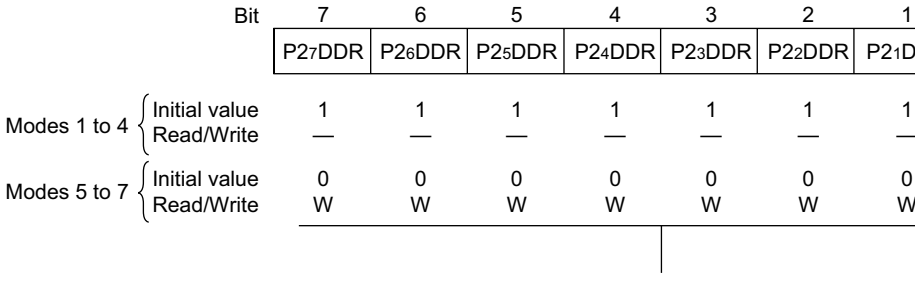


Note: * When the EMC bit in BCR is cleared to 0, addresses of some registers are changed.

Port 1 input/output select

0	Generic input
1	Generic output

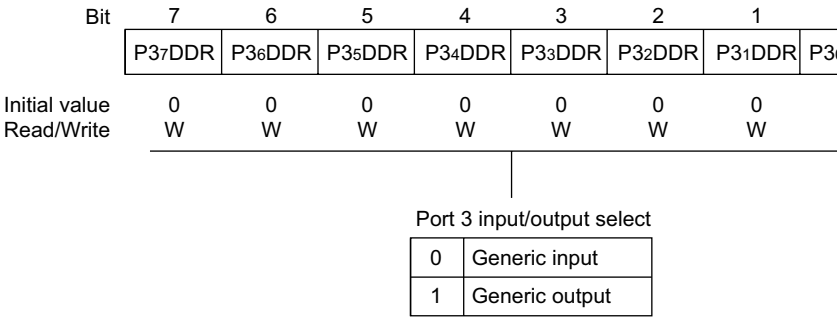
P2DDR—Port 2 Data Direction Register H'EE001



Port 2 input/output select

0	Generic input
1	Generic output

P3DDR—Port 3 Data Direction Register H'EE002



Port 3 input/output select

0	Generic input
1	Generic output

RENESAS

0	Generic input
1	Generic output

P5DDR—Port 5 Data Direction Register H'EE004

Bit	7	6	5	4	3	2	1
	—	—	—	—	P5$_3$DDR	P5$_2$DDR	P5$_1$DD
Modes 1 to 4 Initial value	1	1	1	1	1	1	1
Modes 1 to 4 Read/Write	—	—	—	—	—	—	—
Modes 5 to 7 Initial value	1	1	1	1	0	0	0
Modes 5 to 7 Read/Write	—	—	—	—	W	W	W

Port 5 input/output

0	Generic inpu
1	Generic outp

P6DDR—Port 6 Data Direction Register H'EE005

Bit	7	6	5	4	3	2	1	
	—	P6$_6$DDR	P6$_5$DDR	P6$_4$DDR	P6$_3$DDR	P6$_2$DDR	P6$_1$DDR	P6$_0$I
Initial value	1	0	0	0	0	0	0	0
Read/Write	—	W	W	W	W	W	W	W

Port 6 input/output select

0	Generic input
1	Generic output

Port 8 input/output sele

0	Generic input
1	Generic output

P9DDR—Port 9 Data Direction Register H'EE008

Bit	7	6	5	4	3	2	1	
	—	—	P9$_5$DDR	P9$_4$DDR	P9$_3$DDR	P9$_2$DDR	P9$_1$DDR	P9
Initial value	1	1	0	0	0	0	0	
Read/Write	—	—	W	W	W	W	W	

Port 9 input/output select

0	Generic input
1	Generic output

PADDR—Port A Data Direction Register H'EE009

	Bit	7	6	5	4	3	2	1
		PA$_7$DDR	PA$_6$DDR	PA$_5$DDR	PA$_4$DDR	PA$_3$DDR	PA$_2$DDR	PA$_1$DD
Modes 3, 4	Initial value	1	0	0	0	0	0	0
	Read/Write	—	W	W	W	W	W	W
Modes 1, 2, 5, 6, 7	Initial value	0	0	0	0	0	0	0
	Read/Write	W	W	W	W	W	W	W

Port A input/output select

0	Generic input
1	Generic output

RENESAS

0	Generic input
1	Generic output

MDCR—Mode Control Register H'EE011 Syst

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	MDS2	MDS1	MDS0
Initial value	1	1	0	0	0	—*	—*	—*
Read/Write	—	—	—	—	—	R	R	R

Mode select 2 to 0

Bit 2	Bit 1	Bit 0	Operat
MD_2	MD_1	MD_0	
0	0	0	
		1	Mo
	1	0	Mo
		1	Mo
1	0	0	Mo
		1	Mo
	1	0	Mo
		1	Mo

Note: * Determined by the state of the mode pins (MD_2 to MD_0).

1	On-chip RAM is

Software standby output port

0	In software standby mo all address bus and bus control signals are high- impedance
1	In software standby mo address bus retains out state and bus control signals are fixed high

NMI edge select

0	An interrupt is requested at the fallin
1	An interrupt is requested at the risin

User bit enable

0	CCR bit 6 (UI) is used as an interrupt mask bit
1	CCR bit 6 (UI) is used as a user bit

Standby timer select 2 to 0

Bit 6	Bit 5	Bit 4	Standby Timer
STS2	STS1	STS0	
0	0	0	Waiting Time = 8,192 states
		1	Waiting Time = 16,384 states
	1	0	Waiting Time = 32,768 states
		1	Waiting Time = 65,536 states
1	0	0	Waiting Time = 131,072 states
		1	Waiting Time = 26,2144 states
	1	0	Waiting Time = 1,024 states
		1	Illegal setting

Software standby

0	SLEEP instruction causes transition to sleep mode
1	SLEEP instruction causes transition to software standby mode

Mode 5	Initial value	1	1	1	1	1	1	1
	Read/Write	R/W	R/W	R/W	R/W	—	—	—

Address 23 to 20 enable

0	Address output
1	Other input/output

Bus release e

0	The bus released external
1	The bus released external

ISCR—IRQ Sense Control Register H'EE014 Interrupt C

Bit	7	6	5	4	3	2	1	0
	—	—	IRQ5SC	IRQ4SC	IRQ3SC	IRQ2SC	IRQ1SC	IRQ0SC
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

IRQ_5 to IRQ_0 sense control

0	Interrupts are requested when $\overline{IRQ}_5$ to $\overline{IRQ}_0$ are low
1	Interrupts are requested by falling-edge input at $\overline{IRQ}_5$ to

0	IRQ_5 to IRQ_0 interrupts are disabled
1	IRQ_5 to IRQ_0 interrupts are enabled

ISR—IRQ Status Register H'EE016 Interrupt

Bit	7	6	5	4	3	2	1	0
	—	—	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	IRQ0
Initial value	0	0	0	0	0	0	0	0
Read/Write	—	—	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)

IRQ5 to IRQ0 flags

Bits 5 to 0 IRQ5F to IRQ0F	Setting and Clearing Conditions
0	[Clearing conditions] • Read IRQnF when IRQnF = 1, then write 0 in IRQnF. • IRQnSC = 0, $\overline{IRQn}$ input is high, and interrupt exception handling is being carried out. • IRQnSC = 1 and IRQn interrupt exception handling is be carried out.
1	[Setting conditions] • IRQnSC = 0 and $\overline{IRQn}$ input is low. • IRQnSC = 1 and $\overline{IRQn}$ input changes from high to low.

(r

Note: * Only 0 can be written, to clear the flag.

0	Priority level 0 (low priority)
1	Priority level 1 (high priority)

• Interrupt sources controlled by each bit

IPRA	Bit	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1
		IPRA7	IPRA6	IPRA5	IPRA4	IPRA3	IPRA2	IPRA1
	Interrupt source	IRQ_0	IRQ_1	IRQ_2, IRQ_3	IRQ_4, IRQ_5	WDT, DRAM interface, A/D converter	16-bit timer channel 0	16-bit timer channel 1

IPRB—Interrupt Priority Register B H'EE019 Interrupt C

Bit	7	6	5	4	3	2	1	0
	IPRB7	IPRB6	IPRB5	—	IPRB3	IPRB2	IPRB1	—
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

Priority level B7 to B5, B3 to B1

0	Priority level 0 (low priority)
1	Priority level 1 (high priority)

• Interrupt sources controlled by each bit

IPRB	Bit	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1
		IPRB7	IPRB6	IPRB5	—	IPRB3	IPRB2	IPRB1
	Interrupt source	8-bit timer channels 0 and 1	8-bit timer channels 2 and 3	DMAC	—	SCI channel 0	SCI channel 1	SCI channel 2

0	D/A output is disabled in software standby mode (I
1	D/A output is enabled in software standby mode

DIVCR—Division Control Register H'EE01B Sys

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	DIV1	DIV0
Initial value	1	1	1	1	1	1	0	0
Read/Write	—	—	—	—	—	—	R/W	R/W

Divide 1 and 0

Bit 1	Bit 0	Frequency Div
DIV1	DIV0	
0	0	1/1 (Ir
	1	1/2
1	0	1/4
	1	1/8

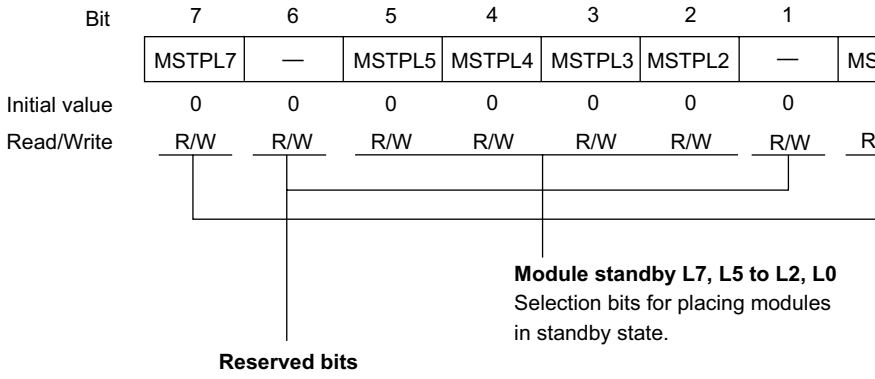
RENESAS

Module standby H
Selection bits for pla
in standby state.

Reserved bits

φ clock stop
Enables or disables φ clock output.

MSTCRL—Module Standby Control Register L **H'EE01D** **Syst**



Address control

Selects address update mode 1 or address update mo

ADRCTL	Description
0	Address update mode 2 is selected
1	Address update mode 1 is selected (Init

Note: * Can be read or written to, but must not be cleared to 0.

CSCR—Chip Select Control Register H'EE01F Bus

Bit	7	6	5	4	3	2	1	
	CS7E	CS6E	CS5E	CS4E	—	—	—	
Initial value	0	0	0	0	1	1	1	
Read/Write	R/W	R/W	R/W	R/W	—	—	—	

Chip select 7 to 4 enable

Bit n CSnE	Description
0	Output of chip select signal CSn is disabled (Initial val
1	Output of chip select signal CSn is enabled

(n = 7 to 4)

Bits 7 to 0	Bus Width of Access Area
ABW7 to ABW0	
0	Areas 7 to 0 are 16-bit access areas
1	Areas 7 to 0 are 8-bit access areas

ASTCR—Access State Control Register **H'EE021** **Bus**

Bit	7	6	5	4	3	2	1	0
	AST7	AST6	AST5	AST4	AST3	AST2	AST1	AS
Initial value	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/

Area 7 to 0 access state control

Bits 7 to 0	Number of States in Access Area
AST7 to AST0	
0	Areas 7 to 0 are two-state access areas
1	Areas 7 to 0 are three-state access areas

0	1	1 program wait state is
1	0	2 program wait states a
	1	3 program wait states a

Area 5 wait control 1 and 0

0	0	No program wait is inserted
	1	1 program wait state is inserted
1	0	2 program wait states are inserted
	1	3 program wait states are inserted

Area 6 wait control 1 and 0

0	0	No program wait is inserted
	1	1 program wait state is inserted
1	0	2 program wait states are inserted
	1	3 program wait states are inserted

Area 7 wait control 1 and 0

0	0	No program wait is inserted
	1	1 program wait state is inserted
1	0	2 program wait states are inserted
	1	3 program wait states are inserted

0	0	No program wait is ins
	1	1 program wait state i
1	0	2 program wait states
	1	3 program wait states

Area 1 wait control 1 and 0

0	0	No program wait is inserted
	1	1 program wait state is inserted
1	0	2 program wait states are inserted
	1	3 program wait states are inserted

Area 2 wait control 1 and 0

0	0	No program wait is inserted
	1	1 program wait state is inserted
1	0	2 program wait states are inserted
	1	3 program wait states are inserted

Area 3 wait control 1 and 0

0	0	No program wait is inserted
	1	1 program wait state is inserted
1	0	2 program wait states are inserted
	1	3 program wait states are inserted

1	WAIT pin wait in

Area division unit select

0	Area divisions are as follows: Area 0: 2 MB Area 4: 1.9 Area 1: 2 MB Area 5: 4 k Area 2: 8 MB Area 6: 23. Area 3: 2 MB Area 7: 22
1	Areas 0 to 7 are the same siz (2 MB)

Burst cycle select 0

0	Max. 4 words in burst access
1	Max. 8 words in burst access

Burst cycle select 1

0	Burst access cycle comprises 2 states
1	Burst access cycle comprises 3 states

Burst ROM enable

0	Area 0 is a basic bus interface area
1	Area 0 is a burst ROM interface area

Idle cycle insertion 0

0	No idle cycle is inserted in case of consecutive external read and write cycles
1	Idle cycle is inserted in case of consecutive external read and write cycles

Idle cycle insertion 1

0	No idle cycle is inserted in case of consecutive external read cycles for different areas
1	Idle cycle is inserted in case of consecutive external read cycles for different areas

Refresh pin enable

0	$\overline{\text{RFSH}}$ pin refresh signal output is
1	$\overline{\text{RFSH}}$ pin refresh signal output is

Self-refresh mode

0	DRAM self-refreshing is disabled in software star
1	DRAM self-refreshing is enabled in software stan

RAS 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	DRAM space access performed in fast page mode

DRAM area select

DRAS2	DRAS1	DRAS0	Area 5	Area 4	Area 3	Area 2
0	0	0	Normal	Normal	Normal	Normal
		1	Normal	Normal	Normal	DRAM space ($\overline{\text{CS}}_2$)
	1	0	Normal	Normal	DRAM space ($\overline{\text{CS}}_3$)	DRAM space ($\overline{\text{CS}}_2$)
		1	Normal	Normal	DRAM space($\overline{\text{CS}}_2$)*	
1	0	0	Normal	DRAM space ($\overline{\text{CS}}_4$)	DRAM space ($\overline{\text{CS}}_3$)	DRAM space ($\overline{\text{CS}}_2$)
		1	DRAM space ($\overline{\text{CS}}_5$)	DRAM space ($\overline{\text{CS}}_4$)	DRAM space ($\overline{\text{CS}}_3$)	DRAM space ($\overline{\text{CS}}_2$)
	1	0	DRAM space($\overline{\text{CS}}_4$)*		DRAM space($\overline{\text{CS}}_2$)*	
		1	DRAM space($\overline{\text{CS}}_2$)*			

Note: * A single $\overline{\text{CSn}}$ pin serves as a common $\overline{\text{RAS}}$ output pin for a number of areas. Unused $\overline{\text{CSn}}$ pins can be used as input/output ports.

1	1 wait state (T_{RW}) is inserted

$\overline{RAS}$-$\overline{CAS}$ wait

0	Wait state (T_{rw}) insertion is disabled
1	1 wait state (T_{rw}) is inserted

TP cycle control

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

Refresh cycle enable

0	Refresh cycles are disabled
1	DRAM refresh cycles are enabled

$\overline{CAS}$ output pin select

0	PB4 and PB5 selected as $\overline{UCAS}$ and $\overline{LCAS}$ output pins
1	$\overline{HWR}$ and $\overline{LWR}$ selected as $\overline{UCAS}$ and $\overline{LCAS}$ output pins

Multiplex control 1 and 0

MXC1	MXC0	Description
0	0	Column address: 8 bits Compared address: Modes 1, 2 — 8-bit access space — A_{19} to A_8 16-bit access space — A_{19} to A_9 Modes 3, 4, 5 — 8-bit access space — A_{23} to A_8 16-bit access space — A_{23} to A_9
	1	Column address: 9 bits Compared address: Modes 1, 2 — 8-bit access space — A_{19} to A_9 16-bit access space — A_{19} to A_{10} Modes 3, 4, 5 — 8-bit access space — A_{23} to A_9 16-bit access space — A_{23} to A_{10}
1	0	Column address: 10 bits Compared address: Modes 1, 2 — 8-bit access space — A_{19} to A_{10} 16-bit access space — A_{19} to A_{11} Modes 3, 4, 5 — 8-bit access space — A_{23} to A_{10} 16-bit access space — A_{23} to A_{11}
	1	Illegal setting

CKS2	CKS1	CKS0	Description
0	0	0	Count operation halted
		1	ϕ/2 used as counter clock
	1	0	ϕ/8 used as counter clock
		1	ϕ/32 used as counter clock
1	0	0	ϕ/128 used as counter clock
		1	ϕ/512 used as counter clock
	1	0	ϕ/2048 used as counter clock
		1	ϕ/4096 used as counter clock

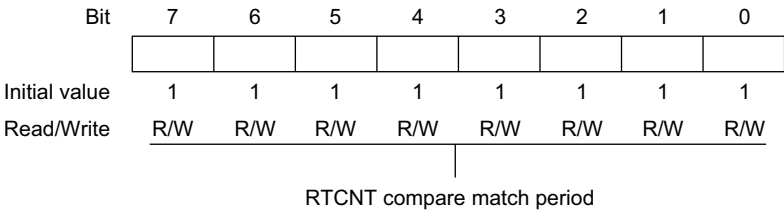
Compare match interrupt enable

0	The CMI interrupt requested by the CMF flag is disabled
1	The CMI interrupt requested by the CMF flag is enabled

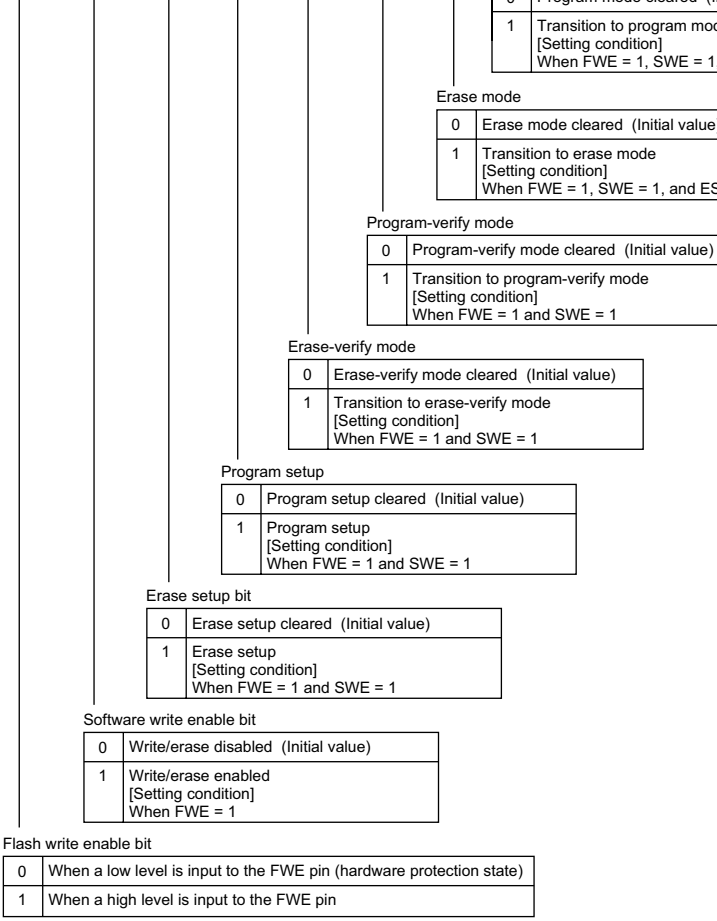
Compare match flag

0	[Clearing conditions] • Cleared by a reset and in standby mode • Cleared by reading CMF when CMF = 1, then writing 0 in CMF
1	[Setting condition] When RTCNT = RTCOR

Note: * Only 0 can be written to clear the flag.



Note: Only byte access can be used on this register.



Notes: 1. This register is used only in the flash memory.
Reading the corresponding address in a mask ROM version will always return 1s, and writes to this address are disabled
2. Fix the FWE pin low in mode 6.

Flash memory error

Notes: 1. Writes to FLMCR2 are prohibited.
2. This register is used only by the flash memory version and do not exist in the r
ROM version. In the mask ROM version reading these addresses always retur
value of 1, and it is not possible to write to them.

EBR (EBR1)—Erase Block Register H'EE032 Fla

Bit		7	6	5	4	3	2	1
		EB7	EB6	EB5	EB4	EB3	EB2	EB1
Modes 1 to 4, and 6	Initial value	0	0	0	0	0	0	0
	Read/Write	R	R	R	R	R	R	R
Modes 5 and 7	Initial value	0	0	0	0	0	0	0
	Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W

Block 7 to 0

0	Block EB7 to EB0 is not selected (Initial value)
1	Block EB7 to EB0 is selected

Notes: 1. When not erasing, clear EBR to H'00.
Writes are invalid.
A value of 1 cannot be set in this register in mode 6.
2. This register is used only by the flash memory version and do not exist in the
ROM version. In the mask ROM version reading these addresses always retu
value of 1, and it is not possible to write to them.

Block 13 to 8

0	Block EB13 to EB8 is not selected (Ini
1	Block EB13 to EB8 is selected

Notes: 1. When not erasing, clear EBR to H'00.
A value of 1 cannot be set in this register in mode 6.
2. This register is used only by the flash memory version and do not exist in the mask RO
version. In the mask ROM version reading these addresses always returns a value of
and it is not possible to write to them.

P2PCR—Port 2 Input Pull-Up Control Register H'EE03C Port

Bit	7	6	5	4	3	2	1	0
	$P2_7PCR$	$P2_6PCR$	$P2_5PCR$	$P2_4PCR$	$P2_3PCR$	$P2_2PCR$	$P2_1PCR$	$P2_0$P
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/

Port 2 input pull-up control 7 to 0

0	Input pull-up transistor is off
1	Input pull-up transistor is on

Note: Valid when the corresponding P2DDR bit is cleared t
(designating generic input).

0	Input pull-up transistor is off
1	Input pull-up transistor is on

Note: Valid when the corresponding P4DDR bit is cleared (designating generic input).

P5PCR—Port 5 Input Pull-Up Control Register H'EE03F

Bit	7	6	5	4	3	2	1	
	—	—	—	—	$P5_3PCR$	$P5_2PCR$	$P5_1PCR$	$P5_0$
Initial value	1	1	1	1	0	0	0	
Read/Write	—	—	—	—	R/W	R/W	R/W	R

Port 5 input pull-up control 3 to 0

0	Input pull-up transistor is off
1	Input pull-up transistor is on

Note: Valid when the corresponding P5DD cleared to 0 (designating generic inp

3 to 7 R/W R/W R/W R/W R/W

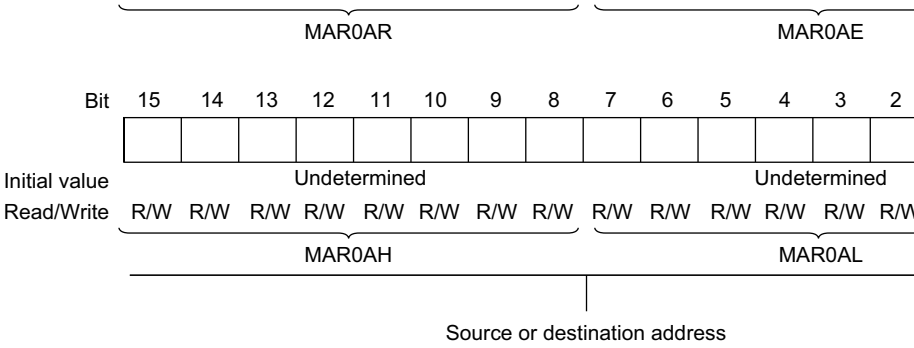
Reserved bits

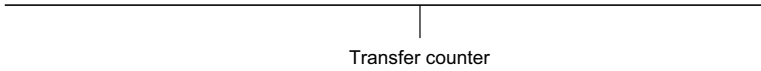
RAM select, RAM2 to RAM0

Bit 3	Bit 2	Bit 1	Bit 0	RAM Area	RAM Emulatio
RAMS	RAM2	RAM1	RAM0		
0	0/1	0/1	0/1	H'FFE000 to H'FFEFFF	Emulation
1	0	0	0	H'000000 to H'000FFF	Mapping RAM
			1	H'001000 to H'001FFF	
		1	0	H'002000 to H'002FFF	
			1	H'003000 to H'003FFF	
	1	0	0	H'004000 to H'004FFF	
			1	H'005000 to H'005FFF	
		1	0	H'006000 to H'006FFF	
			1	H'007000 to H'007FFF	

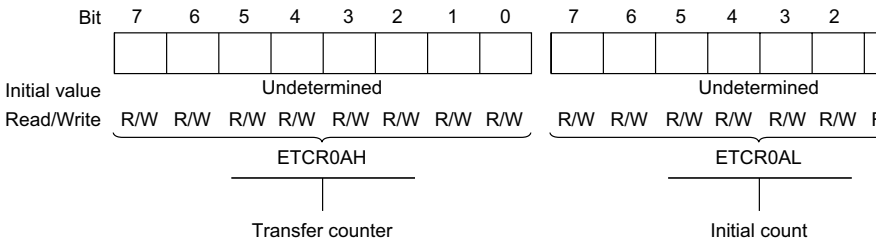
Notes: This register is used only in the flash memory and flash memory R versions.
Reading the corresponding address in a mask ROM version will always return 1s, and writes
address are disabled.

* In mode 6 (single-chip normal mode), flash memory emulation by RAM is not supported; tl
can be modified, but must not be set to 1.


MAR0AR
MAR0AE
Bit
15
14
13
12
11
10
9
8
7
6
5
4
3
2
Initial value
Undetermined
Undetermined
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
MAR0AH
MAR0AL
Source or destination address

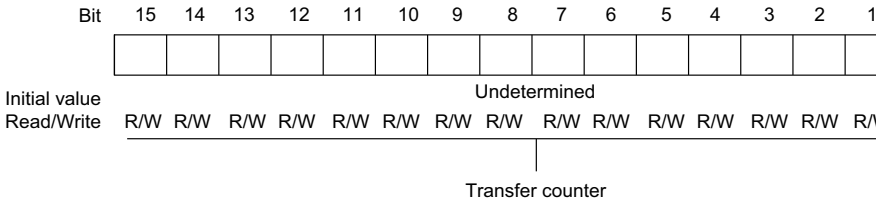


— Repeat mode

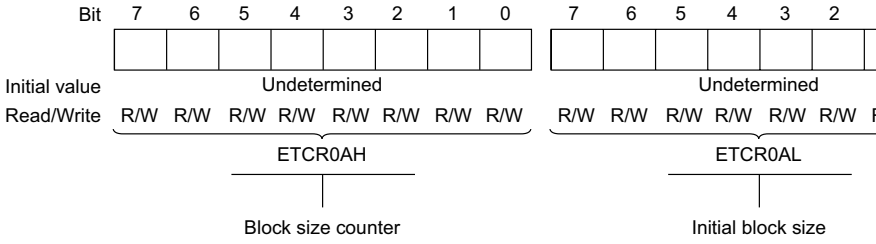


- Full address mode

— Normal mode



— Block transfer mode



Full address mode : not used

Bit 2	Bit 1	Bit 0	Data Transfer Activ
DTS2	DTS1	DTS0	
0	0	0	Compare match/input interrupt from 16-bit ti
		1	Compare match/input interrupt from 16-bit ti
	1	0	Compare match/input interrupt from 16-bit ti
		1	A/D converter convers
1	0	0	SCI0 transmit-data-er
		1	SCI0 receive-data-full
	1	0	Transfer in full addres
		1	Transfer in full addres

Data transfer interrupt enable

0	Interrupt requested by DTE bit is disabled
1	Interrupt requested by DTE bit is enabled

Repeat enable

RPE	DTIE	Description
0	0	I/O mode
	1	
1	0	Repeat mode
	1	Idle mode

Data transfer increment/decrement

0	Incremented:	If DTSZ = 0, MAR is incremented by 1 after each transfer If DTSZ = 1, MAR is incremented by 2 after each transfer
1	Decremented:	If DTSZ = 0, MAR is decremented by 1 after each transfer If DTSZ = 1, MAR is decremented by 2 after each transfer

Data transfer size

0	Byte-size transfer
1	Word-size transfer

Data transfer enable

0	Data transfer is disabled
1	Data transfer is enabled

Data transfe

0	Norm
1	Block

Data transfer select 2A an

Set both bits to 1

Data transfer interrupt enable

0	Interrupt requested by DTE bit is dis
1	Interrupt requested by DTE bit is en

Source address increment/decrement (bit 5)
Source address increment/decrement enable (bit 4)

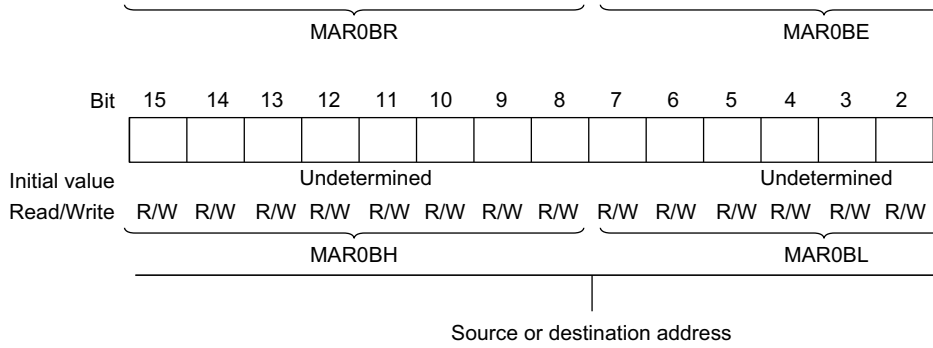
Bit 5	Bit 4	Increment/Decrement Enable
SAID	SAIDE	
0	0	MARA is held fixed
	1	Incremented: If DTSZ = 0, MARA is incremented by 1 after If DTSZ = 1, MARA is incremented by 2 after
1	0	MARA is held fixed
	1	Decremented: If DTSZ = 0, MARA is decremented by 1 after If DTSZ = 1, MARA is decremented by 2 after

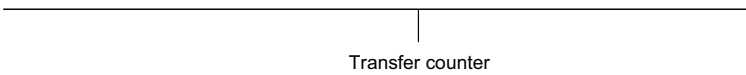
Data transfer size

0	Byte-size transfer
1	Word-size transfer

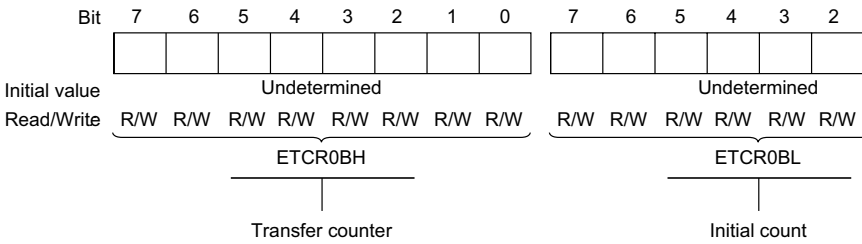
Data transfer enable

0	Data transfer is disabled
1	Data transfer is enabled


MAR0BR
MAR0BE
Bit
15
14
13
12
11
10
9
8
7
6
5
4
3
2
Initial value
Undetermined
Undetermined
Read/Write
R/W
MAR0BH
MAR0BL
Source or destination address

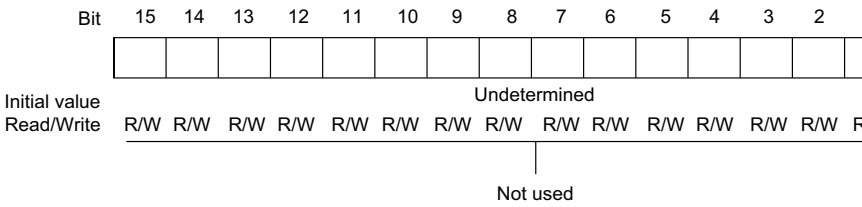


— Repeat mode

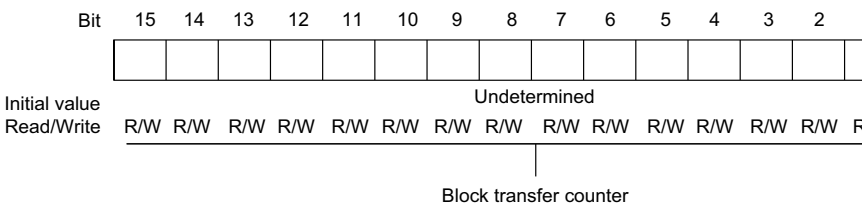


- Full address mode

— Normal mode



— Block transfer mode



Full address mode : not used

Bit 2	Bit 1	Bit 0	Data Transfer Activatio
DTS2	DTS1	DTS0	
0	0	0	Compare match/input ca from 16-bit timer channe
		1	Compare match/input ca from 16-bit timer channe
	1	0	Compare match/input ca from 16-bit timer channe
		1	A/D converter conversion
1	0	0	SCI0 transmit-data-empt
		1	SCI0 receive-data-full int
	1	0	Falling edge of $\overline{\text{DREQ}}$ in
		1	Low level of $\overline{\text{DREQ}}$ input

Data transfer interrupt enable

0	Interrupt requested by DTE bit is disabled
1	Interrupt requested by DTE bit is enabled

Repeat enable

RPE	DTIE	Description
0	0	I/O mode
	1	
1	0	Repeat mode
	1	Idle mode

Data transfer increment/decrement

0	Incremented: If DTSZ = 0, MAR is incremented by 1 after each transf If DTSZ = 1, MAR is incremented by 2 after each transf
1	Decremented: If DTSZ = 0, MAR is decremented by 1 after each trans If DTSZ = 1, MAR is decremented by 2 after each trans

Data transfer size

0	Byte-size transfer
1	Word-size transfer

Data transfer enable

0	Data transfer is disabled
1	Data transfer is enabled

0	Data transfer is disabled
1	Data transfer is enabled

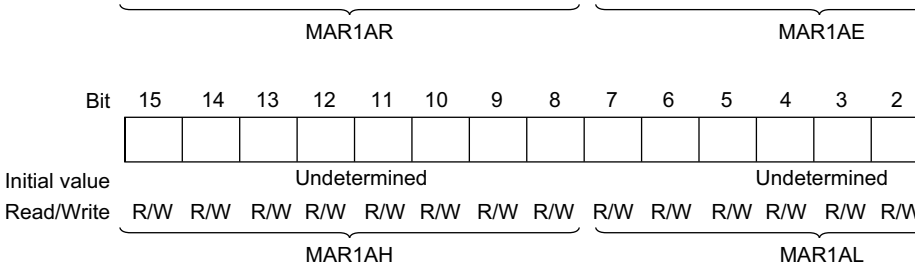
Bit 2	Bit 1	Bit 0	Data Transfer Activation S	
DTS2B	DTS1B	DTS0B	Normal Mode	Block T
0	0	0	Auto-request (burst mode)	Compare capture A 16-bit tim
		1	Not available	Compare capture A 16-bit tim
	1	0	Auto-request (cycle-steal mode)	Compare capture A 16-bit tim
		1	Not available	A/D conv end interr
1	0	0	Not available	Not availa
		1	Not available	Not availa
	1	0	Falling edge input of $\overline{\text{DREQ}}$	Falling ed $\overline{\text{DREQ}}$
		1	Low level input at $\overline{\text{DREQ}}$	Not availa

Transfer mode select

0	Destination is the block area in block transfer mode
1	Source is the block area in block transfer mode

Destination address increment/decrement (bit 5)
Destination address increment/decrement enable (bit 4)

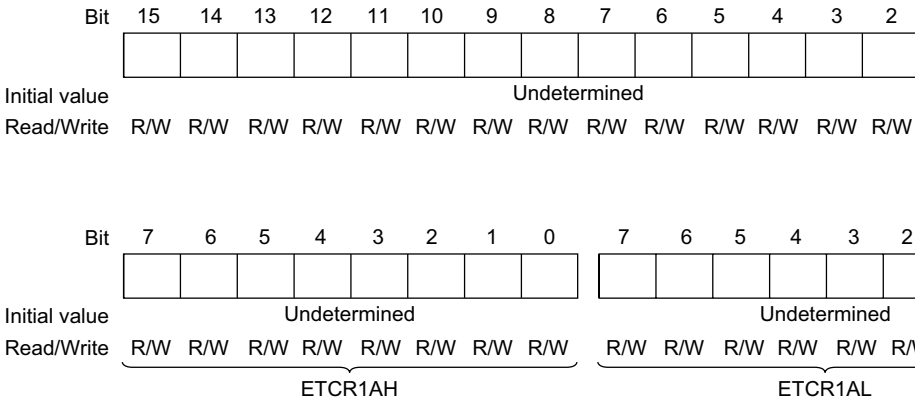
Bit 5	Bit 4	Increment/Decrement Enable
DAID	DAIDE	
0	0	MARB is held fixed
	1	Incremented: If DTSZ = 0, MARB is incremented by 1 after ea If DTSZ = 1, MARB is incremented by 2 after ea
1	0	MARB is held fixed
	1	Decremented: If DTSZ = 0, MARB is decremented by 1 after ea If DTSZ = 1, MARB is decremented by 2 after e



Note: Bit functions are the same as for DMAC0.

ETCR1A H/L—Execute Transfer Count Register 1A H/L

H'FFF34
H'FFF35



Note: Bit functions are the same as for DMAC0.

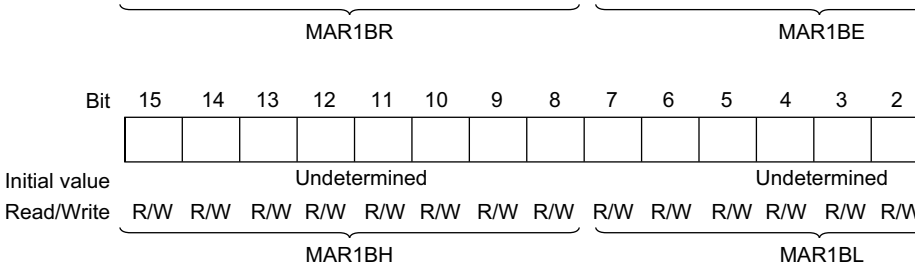
- Short address mode

Bit	7	6	5	4	3	2	1	
	DTE	DTSZ	DTID	RPE	DTIE	DTS2	DTS1	
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	

- Full address mode

Bit	7	6	5	4	3	2	1	
	DTE	DTSZ	SAID	SAIDE	DTIE	DTS2A	DTS1A	D
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	

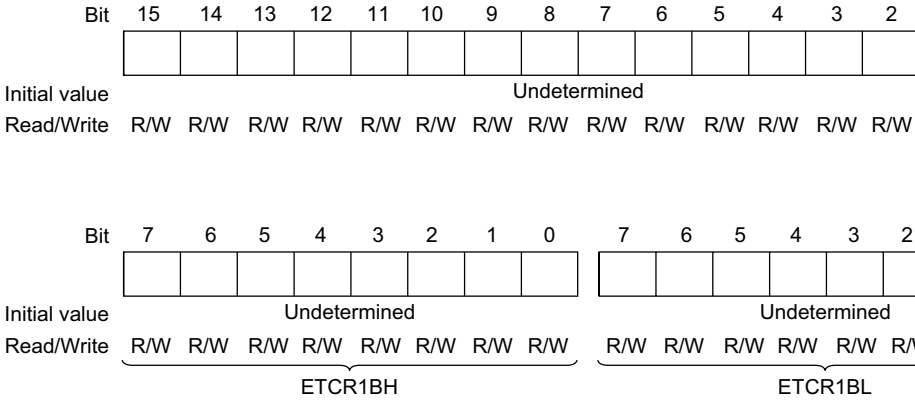
Note: Bit functions are the same as for DMAC0.



Note: Bit functions are the same as for DMAC0.

ETCR1B H/L—Execute Transfer Count Register 1B H/L

H'FFF3C
H'FFF3D



Note: Bit functions are the same as for DMAC0.

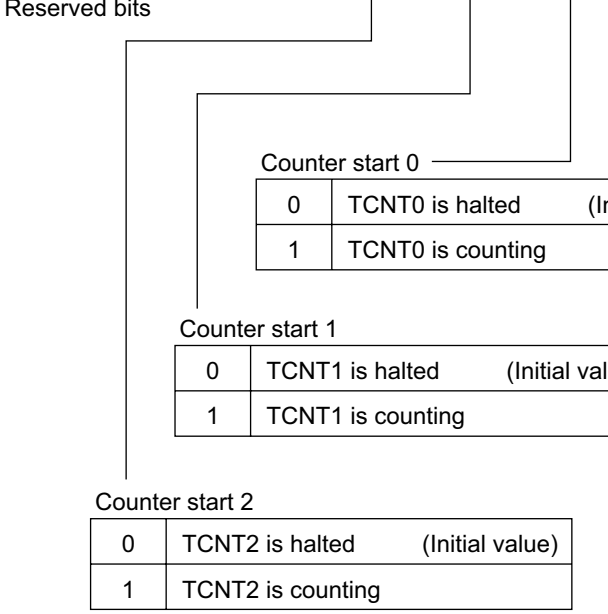
- Short address mode

Bit	7	6	5	4	3	2	1	
	DTE	DTSZ	DTID	RPE	DTIE	DTS2	DTS1	
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	

- Full address mode

Bit	7	6	5	4	3	2	1	
	DTME	—	DAID	DAIDE	TMS	DTS2B	DTS1B	D
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	

Note: Bit functions are the same as for DMAC0.


Reserved bits
Counter start 0
0 TCNT0 is halted (I
1 TCNT0 is counting
Counter start 1
0 TCNT1 is halted (Initial val
1 TCNT1 is counting
Counter start 2
0 TCNT2 is halted (Initial value)
1 TCNT2 is counting

Reserved bits

Timer synchronization 0

0	Channel 0 timer counter (TCNT0) opera independently (TCNT0 presetting/cleari unrelated to other channels) (Ini
1	Channel 0 operates synchronously TCNT0 synchronous presetting/synchro clearing is possible

Timer synchronization 1

0	Channel 1 timer counter (TCNT1) operates independently (TCNT1 presetting/clearing is unrelated to other channels) (Initial valu
1	Channel 1 operates synchronously TCNT1 synchronous presetting/synchronous clearing is possible

Timer synchronization 2

0	Channel 2 timer counter (TCNT2) operates independently (TCNT2 presetting/clearing is unrelated to other channels) (Initial value)
1	Channel 2 operates synchronously TCNT2 synchronous presetting/synchronous clearing is possible

PWM mode 0

0	Channel 0 operates normally (I
1	Channel 0 operates in PWM mo

PWM mode 1

0	Channel 1 operates normally (Initial valu
1	Channel 1 operates in PWM mode

PWM mode 2

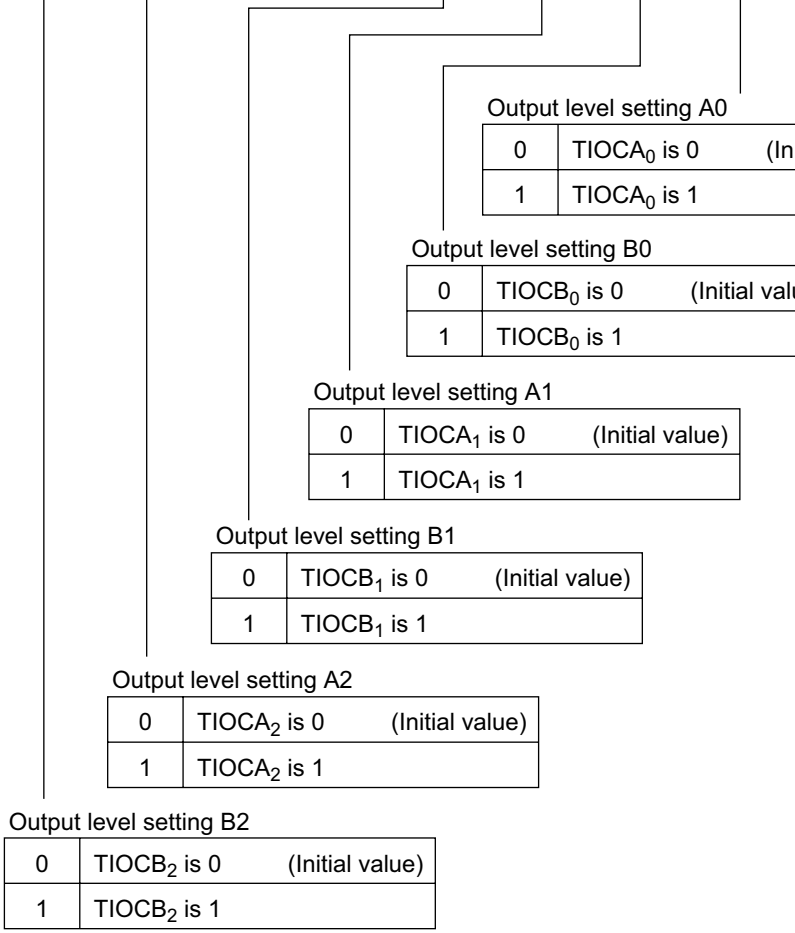
0	Channel 2 operates normally (Initial value)
1	Channel 2 operates in PWM mode

Flag direction

0	OVF is set to 1 in TISRC when TCNT2 overflows or underflows (Initial value)
1	OVF is set to 1 in TISRC when TCNT2 overflows

Phase counting mode flag

0	Channel 2 operates normally (Initial value)
1	Channel 2 operates in phase counting mode


Output level setting A0
0 TIOCA0 is 0 (In
1 TIOCA0 is 1
Output level setting B0
0 TIOCB0 is 0 (Initial valu
1 TIOCB0 is 1
Output level setting A1
0 TIOCA1 is 0 (Initial value)
1 TIOCA1 is 1
Output level setting B1
0 TIOCB1 is 0 (Initial value)
1 TIOCB1 is 1
Output level setting A2
0 TIOCA2 is 0 (Initial value)
1 TIOCA2 is 1
Output level setting B2
0 TIOCB2 is 0 (Initial value)
1 TIOCB2 is 1

0	[Clearing conditions] (Initial • Read IMFA0 when IMFA0=1, then write 0 in IMF • DMAC activated by IMIA0 interrupt.
1	[Setting conditions] • TCNT0=GRA0 when GRA0 functions as an outpu compare register. • TCNT0 value is transferred to GRA0 by an input signal when GRA0 functions as an input capture

Input capture/compare match flag A1

0	[Clearing conditions] (Initial valu • Read IMFA1 when IMFA1=1, then write 0 in IMFA1 • DMAC activated by IMIA1 interrupt.
1	[Setting conditions] • TCNT1=GRA1 when GRA1 functions as an output compare register. • TCNT1 value is transferred to GRA1 by an input captu signal when GRA1 functions as an input capture regis

Input capture/compare match flag A2

0	[Clearing conditions] (Initial value) • Read IMFA2 when IMFA2=1, then write 0 in IMFA2 • DMAC activated by IMIA2 interrupt.
1	[Setting conditions] • TCNT2=GRA2 when GRA2 functions as an output compare register. • TCNT2 value is transferred to GRA2 by an input capture signal when GRA2 functions as an input capture register.

Input capture/compare match interrupt enable A0

0	IMIA0 interrupt requested by IMFA0 flag is disabled (Initial valu
1	IMIA0 interrupt requested by IMFA0 is enabled

Input capture/compare match interrupt enable A1

0	IMIA1 interrupt requested by IMFA1 flag is disabled (Initial value)
1	IMIA1 interrupt requested by IMFA1 is enabled

Input capture/compare match interrupt enable A2

0	IMIA2 interrupt requested by IMFA2 flag is disabled (Initial value)
1	IMIA2 interrupt requested by IMFA2 is enabled

Note: * Only 0 can be written, to clear the flag.

Input capture/compare match flag B0

0	[Clearing condition] (Initi Read IMFB0 when IMFB0=1, then write 0 in IMF
1	[Setting conditions] • TCNT0=GRB0 when GRB0 functions as an ou compare register. • TCNT0 value is transferred to GRB0 by an inp signal when GRB0 functions as an input captu

Input capture/compare match flag B1

0	[Clearing condition] (Initial valu Read IMFB1 when IMFB1=1, then write 0 in IMFB1.
1	[Setting conditions] • TCNT1=GRB1 when GRB1 functions as an output compare register. • TCNT1 value is transferred to GRB1 by an input capt signal when GRB1 functions as an input capture regis

Input capture/compare match flag B2

0	[Clearing condition] (Initial value) Read IMFB2 when IMFB2=1, then write 0 in IMFB2.
1	[Setting conditions] • TCNT2=GRB2 when GRB2 functions as an output compare register. • TCNT2 value is transferred to GRB2 by an input capture signal when GRB2 functions as an input capture register.

Input capture/compare match interrupt enable B0

0	IMIB0 interrupt requested by IMFB0 flag is disabled (Initial value
1	IMIB0 interrupt requested by IMFB0 is enabled

Input capture/compare match interrupt enable B1

0	IMIB1 interrupt requested by IMFB1 flag is disabled (Initial value)
1	IMIB1 interrupt requested by IMFB1 is enabled

Input capture/compare match interrupt enable B2

0	IMIB2 interrupt requested by IMFB2 flag is disabled (Initial value)
1	IMIB2 interrupt requested by IMFB2 is enabled

Note: * Only 0 can be written, to clear the flag.

Overflow flag 0

0	[Clearing condition] Read OVF0 when OVF0 = 1, then write 0 in
1	[Setting condition] TCNT0 overflowed from H'FFFF to H'0000.

Overflow flag 1

0	[Clearing condition] (Initial Read OVF1 when OVF1 = 1, then write 0 in OVF
1	[Setting condition] TCNT1 overflowed from H'FFFF to H'0000.

Overflow flag 2

0	[Clearing condition] (Initial value) Read OVF2 when OVF2 = 1, then write 0 in OVF2.
1	[Setting condition] TCNT2 overflowed from H'FFFF to H'0000, or underflowe from H'0000 to H'FFFF.

Overflow interrupt enable 0

0	OVI0 interrupt requested by OVF0 flag is disabled (Initial val
1	OVI0 interrupt requested by OVF0 flag is enabled

Overflow interrupt enable 1

0	OVI1 interrupt requested by OVF1 flag is disabled (Initial value)
1	OVI1 interrupt requested by OVF1 flag is enabled

Overflow interrupt enable 2

0	OVI2 interrupt requested by OVF2 flag is disabled (Initial value)
1	OVI2 interrupt requested by OVF2 flag is enabled

Note: * Only 0 can be written, to clear the flag.

Timer prescaler 2 to 0

Bit 2	Bit 1	Bit 0	TCNT Clock Source
TPSC2	TPSC1	TPSC0	
0	0	0	Internal clock : ϕ (I
		1	Internal clock : $\phi/2$
	1	0	Internal clock : $\phi/4$
		1	Internal clock : $\phi/8$
1	0	0	External clock A : TCLKA input
		1	External clock B : TCLKB input
	1	0	External clock C : TCLKC input
		1	External clock D : TCLKD input

Clock edge 1 and 0

Bit 4	Bit 3	Counted Edges of External Clock
CKEG1	CKEG0	
0	0	Rising edges counted (Initial value)
0	1	Falling edges counted
1	—	Both edges counted

Counter clear 1 and 0

Bit 6	Bit 5	TCNT clear Sources
CCLR1	CCLR0	
0	0	TCNT is not cleared (Initial value)
	1	TCNT is cleared by GRA compare match or input capture
1	0	TCNT is cleared by GRB compare match or input capture
	1	Synchronous clear : TCNT is cleared in synchronization with other synchronized timers

I/O control A2 to A0

Bit 2	Bit 1	Bit 0	GRA Functions	
IOA2	IOA1	IOA0		
0	0	0	GRA is an output compare register	No output at compare match (
		1		0 output at GRA compare match
	1	0		1 output at GRA compare match
		1		Output toggles at GRA compare m (channel 2 only: 1 output)
1	0	0	GRA is an input capture register	GRA captures rising edges of inpu
		1		GRA captures falling edges of inp
	1	0		GRA captures both edges of input
		1		

I/O control B2 to B0

Bit 6	Bit 5	Bit 4	GRB Functions	
IOB2	IOB1	IOB0		
0	0	0	GRB is an output compare register	No output at compare match (Initial v
		1		0 output at GRB compare match
	1	0		1 output at GRB compare match
		1		Output toggles at GRB compare match (channel 2 only: 1 output)
1	0	0	GRB is an input capture register	GRB captures rising edges of input
		1		GRB captures falling edges of input
	1	0		GRB captures both edges of input
		1		

Up-counter

GRA0 H/L—General Register A0 H/L **H'FFF6C, H'FFF6D** **16-bit timer**

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

Output compare or input capture register

GRB0 H/L—General Register B0 H/L **H'FFF6E, H'FFF6F** **16-bit timer**

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

Output compare or input capture register

TCR1 Timer Control Register 1 **H'FFF70** **16-bit timer**

Bit	7	6	5	4	3	2	1	
	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TP
Initial value	1	0	0	0	0	0	0	
Read/Write	—	R/W	R/W	R/W	R/W	R/W	R/W	R

Note: Bit functions are the same as for 16-bit timer channel 0.

RENESAS

TCNT1 H/L—Timer Counter 1 H/L **H'FFF72, H'FFF73** **16-bit timer**

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
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

Note: Bit functions are the same as for 16-bit timer channel 0.

GRA1 H/L—General Register A1 H/L **H'FFF74, H'FFF75** **16-bit timer**

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

Note: Bit functions are the same as for 16-bit timer channel 0.

GRB1 H/L—General Register B1 H/L **H'FFF76, H'FFF77** **16-bit timer**

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

Note: Bit functions are the same as for 16-bit timer channel 0.

2. When phase counting mode is selected in channel 2, the settings of b
CKEG1 and CKEG0 and TPSC2 to TPSC0 in TCR2 are ignored.

TIOR2—Timer I/O Control Register 2 **H'FFF79** **16-bit timer**

Bit	7	6	5	4	3	2	1	
	—	IOB2	IOB1	IOB0	—	IOA2	IOA1	IO
Initial value	1	0	0	0	1	0	0	
Read/Write	—	R/W	R/W	R/W	—	R/W	R/W	R

Note: Bit functions are the same as for 16-bit timer channel 0.

TCNT2 H/L—Timer Counter 2 H/L **H'FFF7A, H'FFF7B** **16-bit timer**

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
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

Phase counting mode : up/down counter
Other mode : up-counter

GRA2 H/L—General Register A2 H/L **H'FFF7C, H'FFF7D** **16-bit timer**

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

Note: Bit functions are the same as for 16-bit timer channel 0.

Note: Bit functions are the same as for 16-bit timer channel 0.

0	0	0	Clock input is disabled
		1	Internal clock, counted on ris edge of ϕ/8
	1	0	Internal clock, counted on ris edge of ϕ/64
		1	Internal clock, counted on ris edge of ϕ/8192
1	0	0	Channel 0: Count on TCNT1 overflow s Channel 1: Count on TCNT0 compare A*
		1	External clock, counted on fallin
	1	0	External clock, counted on rising
		1	External clock, counted on b rising and falling edges

Notes: * If the clock input of channel 0 is the T overflow signal and that of channel 1 TCNT0 compare match signal, no incrementing clock is generated. Do this setting.

Counter clear 1 and 0

0	0	Clearing is disabled
	1	Cleared by compare match A
1	0	Cleared by compare match B/input capture B
	1	Cleared by input capture B

Timer overflow interrupt enable

0	OVI interrupt requested by OVF is disabled
1	OVI interrupt requested by OVF is enabled

Compare match interrupt enable A

0	CMIA interrupt requested by CMFA is disabled
1	CMIA interrupt requested by CMFA is enabled

Compare match interrupt enable B

0	CMIB interrupt requested by CMFB is disabled
1	CMIB interrupt requested by CMFB is enabled

1	0	1 output at compare match A
	1	Output toggles at compare match A

Output/input capture edge select B3 and B2

ICE in TCSR1	Bit 3 OIS3	Bit 2 OIS2	Description
0	0	0	No change at compare match B
		1	0 output at compare match B
	1	0	1 output at compare match B
		1	Output toggles at compare match B
1	0	0	TCORB input capture on rising edge
		1	TCORB input capture on falling edge
	1	0	TCORB input capture on both rising and falling edges
		1	

A/D trigger enable (TCSR0 only)

TRGE*	Bit 4 ADTE	Description
0	0	A/D converter start requests by compare match A or an external trigger are disabled
	1	A/D converter start requests by compare match A or an external trigger are enabled
1	0	A/D converter start requests by an external trigger are enal
	1	A/D converter start requests by compare match A are enab

Note: * TRGE is bit 7 of the A/D control register (ADCR).

Timer overflow flag

0	[Clearing condition] Read OVF when OVF = 1, then write 0 in OVF.
1	[Setting condition] TCNT overflows from H'FF to H'00.

Compare match flag A

0	[Clearing condition] Read CMFA when CMFA = 1, then write 0 in CMFA.
1	[Setting condition] TCNT = TCORA

Compare match/input capture flag B

0	[Clearing condition] Read CMFB when CMFB = 1, then write 0 in CMFB.
1	[Setting conditions] • TCNT = TCORB • The TCNT value is transferred to TCORB by an input capture signal when TCORB functions as an input capture register.

Note: * Only 0 can be written to bits 7 to 5, to clear these flags.

0	0	No change at compare match
	1	0 output at compare match A
1	0	1 output at compare match A
	1	Output toggles at compare ma

Output/input capture edge select B3 and B2

ICE in TCSR1	Bit 3 OIS3	Bit 2 OIS2	Description
0	0	0	No change at compare match B
		1	0 output at compare match B
	1	0	1 output at compare match B
		1	Output toggles at compare mat
1	0	0	TCORB input capture on rising
		1	TCORB input capture on falling
	1	0	TCORB input capture on both r
		1	and falling edges

Input capture enable

0	TCORB is a compare match register
1	TCORB is an input capture register

Timer overflow flag

0	[Clearing condition] Read OVF when OVF = 1, then write 0 in OVF.
1	[Setting condition] TCNT overflows from H'FF to H'00.

Compare match/input capture flag A

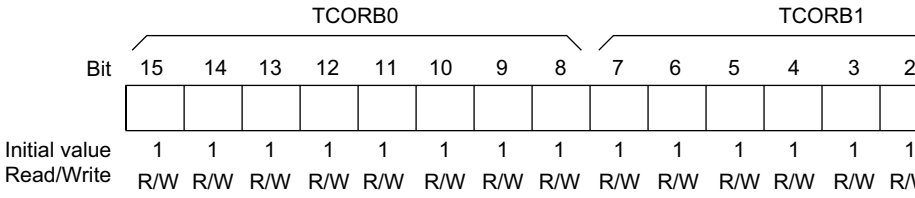
0	[Clearing condition] Read CMFA when CMFA = 1, then write 0 in CMFA.
1	[Setting condition] TCNT = TCORA

Compare match/input capture flag B

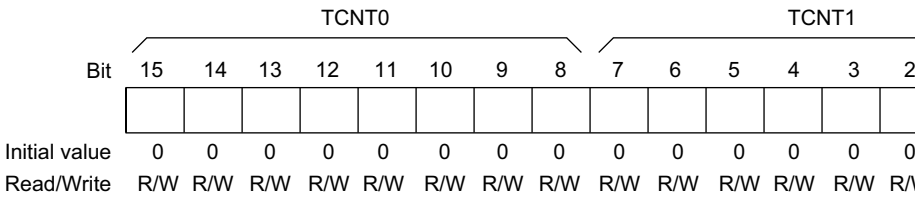
0	[Clearing condition] Read CMFB when CMFB = 1, then write 0 in CMFB.
1	[Setting conditions] • TCNT = TCORB • The TCNT value is transferred to TCORB by an input capture signal when TCORB functions as an input capture register.

Note: * Only 0 can be written to bits 7 to 5, to clear these flags.

TCORB0—Time Constant Register B0 **H'FFF86** **8-bit time**
TCORB1—Time Constant Register B1 **H'FFF87** **8-bit time**



TCNT0—Timer Counter 0 **H'FFF88** **8-bit time**
TCNT1—Timer Counter 1 **H'FFF89** **8-bit time**



CKS2	CKS1	CKS0	D
0	0	0	φ/2
		1	φ/32
	1	0	φ/64
		1	φ/128
1	0	0	φ/256
		1	φ/512
	1	0	φ/204
		1	φ/409

Timer enable

0	Timer disabled • TCNT is initialized to H'00 and halted
1	Timer enabled • TCNT is counting

Timer mode select

0	Interval timer: requests interval timer interrupts
1	Watchdog timer: generates a reset signal

Overflow flag

0	[Clearing condition] Read OVF when OVF = 1, then write 0 in OVF
1	[Setting condition] TCNT changes from H'FF to H'00

Note: * Only 0 can be written, to clear the flag.

Count value

RSTCSR—Reset Control/Status Register **H'FFF8F (read), H'FFF8E (write)**

Bit	7	6	5	4	3	2	1
	WRST	RSTOE	—	—	—	—	—
Initial value	0	0	1	1	1	1	1
Read/Write	R/(W)*	R/W	—	—	—	—	—

Reset output enable

0	External output of reset signal is disabled
1	External output of reset signal is enabled

Watchdog timer reset

0	[Clearing conditions] Reset signal at $\overline{\text{RES}}$ pin Read WRST when WRST = 1, then write 0 in WRST
1	[Setting condition] TCNT overflow generates a reset signal

Note: * Only 0 can be written in bit 7, to clear the flag.

CSK2	CSK1	CSK0	Description
0	0	0	Clock input is disabled
		1	Internal clock, counted on ris of ϕ/8
	1	0	Internal clock, counted on ris of ϕ/64
		1	Internal clock, counted on ris of ϕ/8192
1	0	0	Channel 2: Count on TCNT3 overflow Channel 3: Count on TCNT2 compare
		1	External clock, counted on fa
	1	0	External clock, counted on ris
		1	External clock, counted on bo rising and falling edges

Note: * If the clock input of channel 2 is the TCNT signal and that of channel 3 is the TCNT2 match signal, no incrementing clock is gen Do not use this setting.

Counter clear 1 and 0

0	0	Clearing is disabled
	1	Cleared by compare match A
1	0	Cleared by compare match B/input capture B
	1	Cleared by input capture B

Timer overflow interrupt enable

0	OVI interrupt requested by OVF is disabled
1	OVI interrupt requested by OVF is enabled

Compare match interrupt enable A

0	CMIA interrupt requested by CMFA is disabled
1	CMIA interrupt requested by CMFA is enabled

Compare match interrupt enable B

0	CMIB interrupt requested by CMFB is disabled
1	CMIB interrupt requested by CMFB is enabled

Read/Write	R/(W)*	R/(W)*	R/(W)*	R/W	R/W	R/W	R/W	R/W

Output select A1 and A0

Bit 1	Bit 0	Description
OS1	OS0	
0	0	No change at compare match
	1	0 output at compare match A
1	0	1 output at compare match A
	1	Output toggles at compare m

Output/input capture edge select B3 and B2

ICE in TCSR3	Bit 3	Bit 2	Description
	OIS3	OIS2	
0	0	0	No change at compare match B
		1	0 output at compare match B
	1	0	1 output at compare match B
		1	Output toggles at compare mat
1	0	0	TCORB input capture on rising
		1	TCORB input capture on falling
	1	0	TCORB input capture on both r and falling edges

Input capture enable (TCSR3 only)

0	TCORB is a compare match register
1	TCORB is an input capture register

Timer overflow flag

0	[Clearing condition] Read OVF when OVF = 1, then write 0 in OVF.
1	[Setting condition] TCNT overflows from H'FF to H'00.

Compare match/input capture flag A

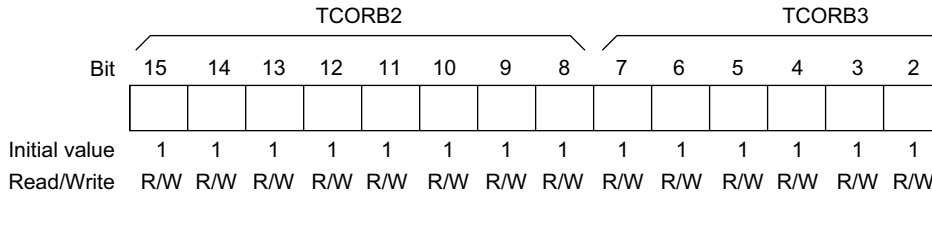
0	[Clearing condition] Read CMFA when CMFA = 1, then write 0 in CMFA.
1	[Setting condition] TCNT = TCORA

Compare match/input capture flag B

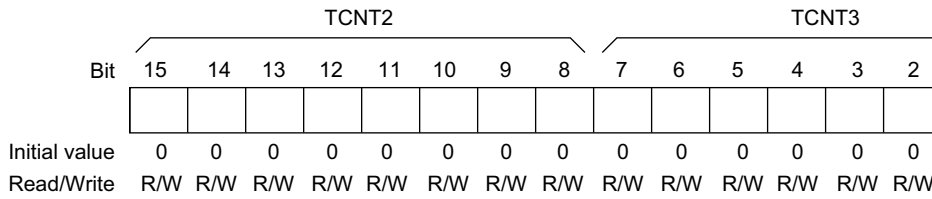
0	[Clearing condition] Read CMFB when CMFB = 1, then write 0 in CMFB.
1	[Setting conditions] • TCNT = TCORB • The TCNT value is transferred to TCORB by an input capture signal when TCORB functions as an input capture register.

Note: * Only 0 can be written to bits 7 to 5, to clear these flags.

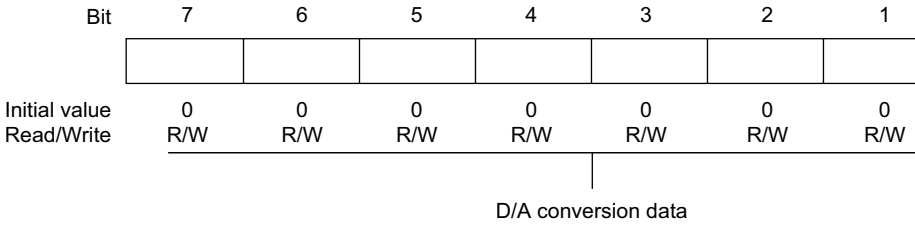
TCORB2—Time Constant Register B2 **H'FFF96** **8-bit timer**
TCORB3—Time Constant Register B3 **H'FFF97** **8-bit timer**



TCNT2—Timer Counter 2 **H'FFF98** **8-bit timer**
TCNT3—Timer Counter 3 **H'FFF99** **8-bit timer**



DADR1—D/A Data Register 1 **H'FFF9D** **D/A**



Bit 7	Bit 6	Bit 5	Description
DAOE1	DAOE0	DAE	
0	0	—	D/A conversion in channels 0 ar
0	1	0	D/A conversion in channel 0 D/A conversion in channel 1
0	1	1	D/A conversion in channels 0 ar
1	0	0	D/A conversion in channel 0 D/A conversion in channel 1
1	0	1	D/A conversion in channels 0 ar
1	1	—	D/A conversion in channels 0 ar

D/A output enable 0

0	DA_0 analog output is disabled
1	Channel-0 D/A conversion and DA_0 analog output are enabled

D/A output enable 1

0	DA_1 analog output is disabled
1	Channel-1 D/A conversion and DA_1 analog output are enabled

0	change at compare match A 16-bit timer channel
1	Non-overlapping TPC output controlled by compare match selected 16-bit timer channe

Group 1 non-overlap

0	Normal TPC output in group 1. Output valu at compare match A in the selected 16-bit
1	Non-overlapping TPC output in group 1, co compare match A and B in the selected 16

Group 2 non-overlap

0	Normal TPC output in group 2. Output values change compare match A in the selected 16-bit timer channel
1	Non-overlapping TPC output in group 2, controlled by compare match A and B in the selected 16-bit timer c

Group 3 non-overlap

0	Normal TPC output in group 3. Output values change at compare match A in the selected 16-bit timer channel
1	Non-overlapping TPC output in group 3, controlled by compare match A and B in the selected 16-bit timer channel

0	1	TPC output group 0 (TP3 to TP0 compare match in 16-bit timer c
1	0	TPC output group 0 (TP3 to TP0 compare match in 16-bit timer c
	1	

Group 1 compare match select 1 and 0

Bit 3	Bit 2	16-Bit Timer Channel Selected as Output Trig
G1CMS1	G1CMS0	
0	0	TPC output group 1 (TP7 to TP4) is triggered compare match in 16-bit timer channel 0
	1	TPC output group 1 (TP7 to TP4) is triggered compare match in 16-bit timer channel 1
1	0	TPC output group 1 (TP7 to TP4) is triggered compare match in 16-bit timer channel 2
	1	

Group 2 compare match select 1 and 0

Bit 5	Bit 4	16-Bit Timer Channel Selected as Output Trigger
G2CMS1	G2CMS0	
0	0	TPC output group 2 (TP11 to TP8) is triggered by compare match in 16-bit
	1	TPC output group 2 (TP11 to TP8) is triggered by compare match in 16-bit
1	0	TPC output group 2 (TP11 to TP8) is triggered by compare match in 16-bit
	1	

Group 3 compare match select 1 and 0

Bit 7	Bit 6	16-Bit Timer Channel Selected as Output Trigger
G3CMS1	G3CMS0	
0	0	TPC output group 3 (TP15 to TP12) is triggered by compare match in 16-bit timer channel 0
	1	TPC output group 3 (TP15 to TP12) is triggered by compare match in 16-bit timer channel 1
1	0	TPC output group 3 (TP15 to TP12) is triggered by compare match in 16-bit timer channel 2
	1	

Next data enable 15 to 8

Bits 7 to 0 NDER15 to NDER8	Description
0	TPC outputs TP_{15} to TP_8 are disabled (NDR15 to NDR8 are not transferred to PB_7 to PB_0)
1	TPC outputs TP_{15} to TP_8 are enabled (NDR15 to NDR8 are transferred to PB_7 to PB_0)

NDERA—Next Data Enable Register A H'FFFA3

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

Next data enable 7 to 0

Bits 7 to 0 NDER7 to NDER0	Description
0	TPC outputs TP_7 to TP_0 are disabled (NDR7 to NDR0 are not transferred to PA_7 to PA_0)
1	TPC outputs TP_7 to TP_0 are enabled (NDR7 to NDR0 are transferred to PA_7 to PA_0)

Store the next output data for TPC output group 3 Store the next output data for TPC o

— Address H'FFFA6

Bit	7	6	5	4	3	2	1	
	—	—	—	—	—	—	—	
Initial value	1	1	1	1	1	1	1	
Read/Write	—	—	—	—	—	—	—	

- Different triggers for TPC output groups 2 and 3

— Address H'FFFA4

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	—	—	—	

Store the next output data for TPC output group 3

— Address H'FFFA6

Bit	7	6	5	4	3	2	1	
	—	—	—	—	NDR11	NDR10	NDR9	
Initial value	1	1	1	1	0	0	0	
Read/Write	—	—	—	—	R/W	R/W	R/W	

Store the next output data for TPC out

Store the next output data for TPC output group 1 Store the next output data for TPC outp

— Address H'FFFA7

Bit	7	6	5	4	3	2	1	
	—	—	—	—	—	—	—	
Initial value	1	1	1	1	1	1	1	
Read/Write	—	—	—	—	—	—	—	

- Different triggers for TPC output groups 0 and 1

— Address H'FFFA5

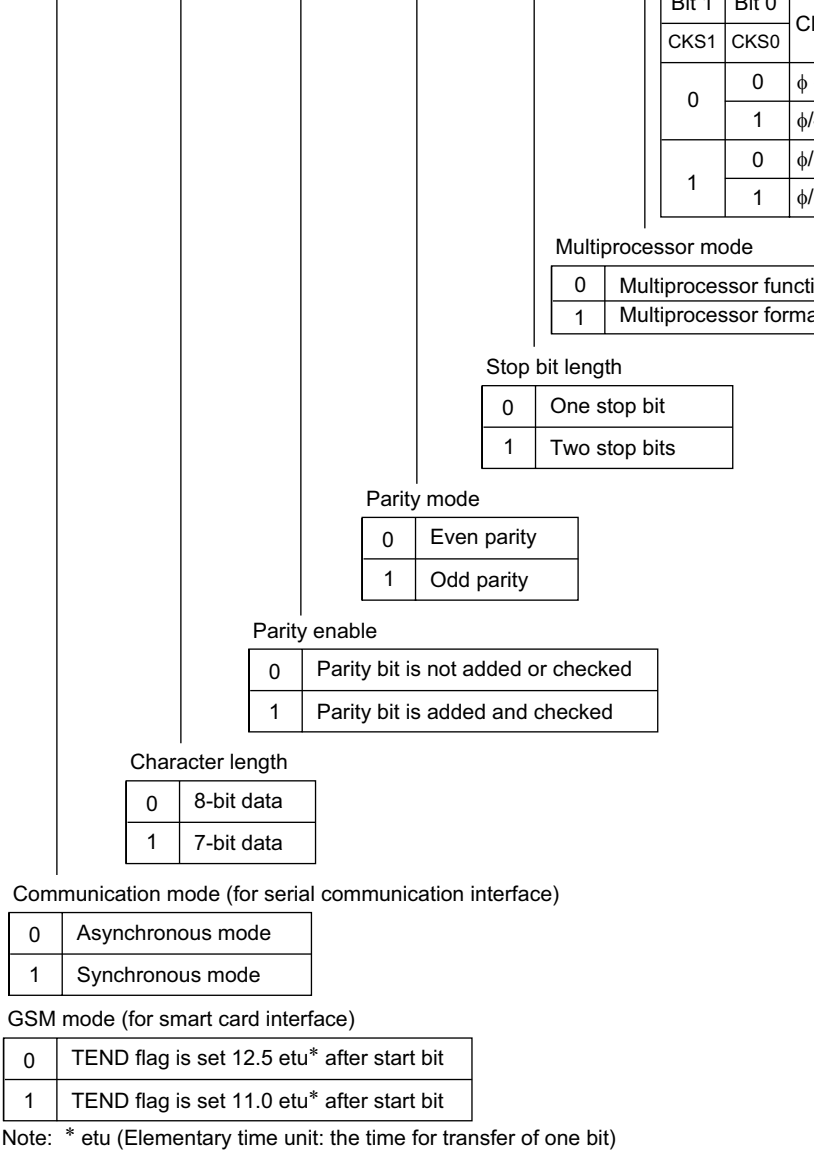
Bit	7	6	5	4	3	2	1	
	NDR7	NDR6	NDR5	NDR4	—	—	—	
Initial value	0	0	0	0	1	1	1	
Read/Write	R/W	R/W	R/W	R/W	—	—	—	

Store the next output data for TPC output group 1

— Address H'FFFA7

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	

Store the next output data for TPC o



Bit 1	Bit 0	CK
CKS1	CKS0	
0	0	φ
	1	φ/
1	0	φ/
	1	φ/

Multiprocessor mode

0	Multiprocessor functi
1	Multiprocessor forma

Stop bit length

0	One stop bit
1	Two stop bits

Parity mode

0	Even parity
1	Odd parity

Parity enable

0	Parity bit is not added or checked
1	Parity bit is added and checked

Character length

0	8-bit data
1	7-bit data

Communication mode (for serial communication interface)

0	Asynchronous mode
1	Synchronous mode

GSM mode (for smart card interface)

0	TEND flag is set 12.5 etu* after start bit
1	TEND flag is set 11.0 etu* after start bit

Note: * etu (Elementary time unit: the time for transfer of one bit)

1	Receiving is enabled

Transmit enable

0	Transmitting is disabled
1	Transmitting is enabled

0	0	Asynchronous mode	Internal clo available fo
		Synchronous mode	Internal clo used for se
	1	Asynchronous mode	Internal clo used for cl
		Synchronous mode	Internal clo used for se
1	0	Asynchronous mode	External cl used for cl
		Synchronous mode	External cl used for se
	1	Asynchronous mode	External cl used for cl
		Synchronous mode	External cl used for se

Clock enable 1 and 0 (for smart card interface)

SMR	Bit 1	Bit 0	Description
GM	CKE1	CKE0	
0	0	0	SCK pin available for ge
		1	SCK pin used for clock o
1	0	0	SCK pin output fixed low
		1	SCK pin used for clock o
	1	0	SCK pin output fixed hig
		1	SCK pin used for clock o

Transmit-end interrupt enable

0	Transmit-end interrupt requests (TEI) are disable
1	Transmit-end interrupt requests (TEI) are enable

Multiprocessor interrupt enable

0	Multiprocessor interrupts are disabled (normal receive operatio
1	Multiprocessor interrupts are enabled

Receive interrupt enable

0	Receive-data-full (RXI) and receive-error (ERI) interrupt requests are disabled
1	Receive-data-full (RXI) and receive-error (ERI) interrupt requests are enabled

Transmit interrupt enable

0	Transmit-data-empty interrupt request (TXI) is disabled
1	Transmit-data-empty interrupt request (TXI) is enabled

0	Multiprocessor bit value in receive data
1	Multiprocessor bit value in receive data

Transmit end (for serial communication interface)

0	[Clearing conditions] • Read TDRE when TDRE = 1, then write 0 in TDR • The DMAC writes data in TDR.
1	[Setting conditions] • Reset or transition to standby mode • TE is cleared to 0 in SCR. • TDRE is 1 when last bit of 1-byte serial character transmitted.

Transmit end (for smart card interface)

0	[Clearing conditions] • Read TDRE when TDRE = 1, then write 0 in TDR • The DMAC writes data in TDR.
1	[Setting conditions] • Reset or transition to standby mode • TE is cleared to 0 in SCR and FER/ERS is cleare • TDRE is 1 and FER/ERS is 0 (normal transmissi (when GM = 0) or 1.0 etu (when GM = 1) after 1- character is transmitted.

Note: * etu (Elementary time unit: the time for transfer of

Parity error

0	[Clearing conditions]	• Reset or transition to standby mode • Read PER when PER = 1, then write 0 in F
1	[Setting condition]	Parity error (parity of receive data does not m setting of O/E bit in SMR)

Framing error (for serial communication interface)

0	[Clearing conditions]	• Reset or transition to standby mode • Read FER when FER = 1, then write 0 in FER.
1	[Setting condition]	Framing error (stop bit is 0)

Error signal status (for smart card interface)

0	[Clearing conditions]	• Reset or transition to standby mode • Read ERS when ERS = 1, then write 0 in ERS.
1	[Setting condition]	A low error signal is received.

Overrun error

0	[Clearing conditions]	• Reset or transition to standby mode • Read ORER when ORER = 1, then write 0 in ORER.
1	[Setting condition]	Overrun error (reception of the next serial data ends when RDRF

Receive data register full

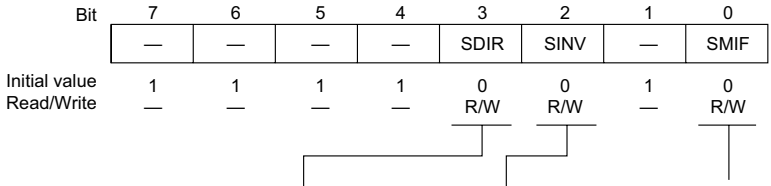
0	[Clearing conditions]	• Reset or transition to standby mode • Read RDRF when RDRF = 1, then write 0 in RDRF. • The DMAC reads data from RDR.
1	[Setting condition]	Serial data is received normally and transferred from RSR to RDR.

Transmit data register empty

0	[Clearing conditions]	• Read TDRE when TDRE = 1, then write 0 in TDRE. • The DMAC writes data in TDR.
1	[Setting conditions]	• Reset or transition to standby mode • TE is 0 in SCR. • Data is transferred from TDR to TSR, enabling new data to be written in TDR

Note: 1. Only 0 can be written, to clear the flag.

SCMR—Smart Card Mode Register H'FFFB6



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	Unmodified TDR contents are transmitted (Initial v Receive data is stored unmodified in RDR
1	Inverted 1/0 logic levels of TDR contents are transmitted 1/0 logic levels of received data are inverted before storage in

Smart card data transfer direction

0	TDR contents are transmitted LSB-first (Initial value) Receive data is stored LSB-first in RDR
1	TDR contents are transmitted MSB-first Receive data is stored MSB-first in RDR

BRR—Bit Rate Register **H'FFFB9**

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

Note: Bit functions are the same as for SCI0.

SCR—Serial Control Register **H'FFFBA**

Bit	7	6	5	4	3	2	1
	TIE	RIE	TE	RE	MPIE	TEIE	CKE1
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

Note: Bit functions are the same as for SCI0.

TDR—Transmit Data Register **H'FFFBB**

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

Note: Bit functions are the same as for SCI0.

RDR—Receive Data Register **H'FFFBD**

Bit	7	6	5	4	3	2	1
Initial value	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R

Note: Bit functions are the same as for SCI0.

SCMR—Smart Card Mode Register **H'FFFBE**

Bit	7	6	5	4	3	2	1
	—	—	—	—	SDIR	SINV	—
Initial value	1	1	1	1	0	0	1
Read/Write	—	—	—	—	R/W	R/W	—

Note: Bit functions are the same as for SCI0.

SMR—Serial Mode Register **H'FFFC0**

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

Note: Bit functions are the same as for SCI0.

SCR—Serial Control Register **H'FFFC2**

Bit	7	6	5	4	3	2	1
	TIE	RIE	TE	RE	MPIE	TEIE	CKE1
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

Note: Bit functions are the same as for SCI0.

TDR—Transmit Data Register **H'FFFC3**

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

Note: Bit functions are the same as for SCI0.

SSR—Serial Status Register **H'FFFC4**

Bit	7	6	5	4	3	2	1
	TDRE	RDRF	ORER	FER/ERS	PER	TEND	MPB
Initial value	1	0	0	0	0	1	0
Read/Write	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R

Note: Bit functions are the same as for SCI0.
* Only 0 can be written, to clear the flag.

SCMR—Smart Card Mode Register **H'FFFC6**

Bit	7	6	5	4	3	2	1
	—	—	—	—	SDIR	SINV	—
Initial value	1	1	1	1	0	0	1
Read/Write	—	—	—	—	R/W	R/W	—

Note: Bit functions are the same as for SCI0.

P1DR—Port 1 Data Register **H'FFFD0**

Bit	7	6	5	4	3	2	1
	$P1_7$	$P1_6$	$P1_5$	$P1_4$	$P1_3$	$P1_2$	$P1_1$
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

Data for port 1 pins

P2DR—Port 2 Data Register **H'FFFD1**

Bit	7	6	5	4	3	2	1
	$P2_7$	$P2_6$	$P2_5$	$P2_4$	$P2_3$	$P2_2$	$P2_1$
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

Data for port 2 pins

P4DR—Port 4 Data Register **H'FFFD3**

Bit	7	6	5	4	3	2	1
	$P4_7$	$P4_6$	$P4_5$	$P4_4$	$P4_3$	$P4_2$	$P4_1$
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

Data for port 4 pins

P5DR—Port 5 Data Register **H'FFFD4**

Bit	7	6	5	4	3	2	1
	—	—	—	—	$P5_3$	$P5_2$	$P5_1$
Initial value	1	1	1	1	0	0	0
Read/Write	—	—	—	—	R/W	R/W	R/W

Data for port 5 pins

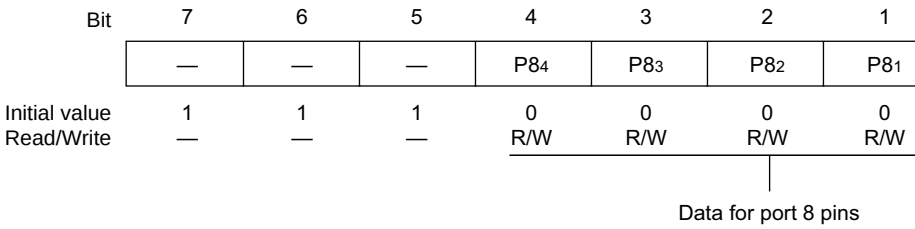
P6DR—Port 6 Data Register **H'FFFD5**

Bit	7	6	5	4	3	2	1
	$P6_7$	$P6_6$	$P6_5$	$P6_4$	$P6_3$	$P6_2$	$P6_1$
Initial value	1	0	0	0	0	0	0
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W

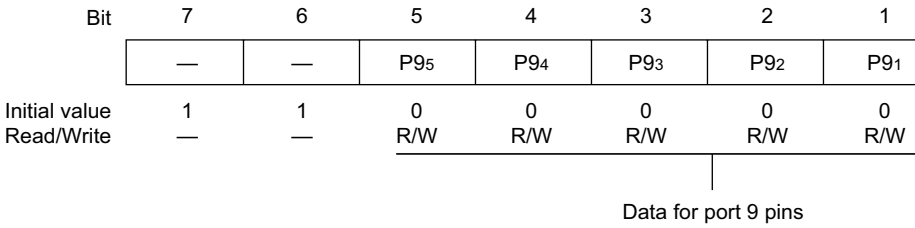
Data for port 6 pins

Note: * Determined by pins $P7_7$ to $P7_0$.

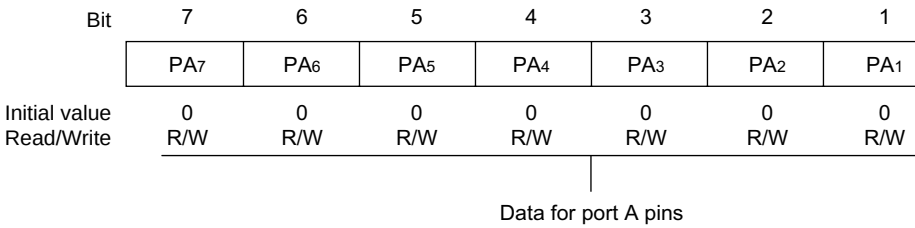
P8DR—Port 8 Data Register **H'FFFD7**



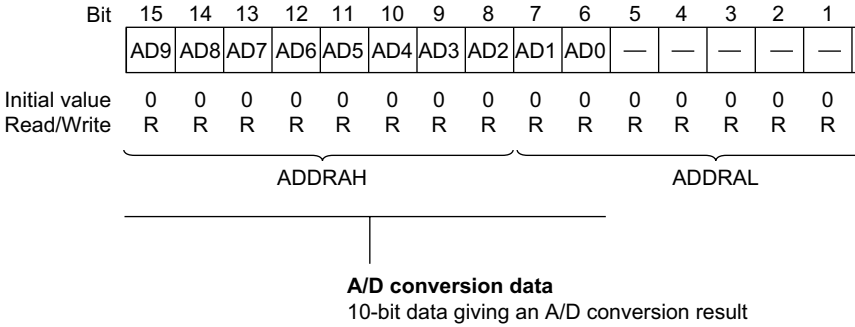
P9DR—Port 9 Data Register **H'FFFD8**



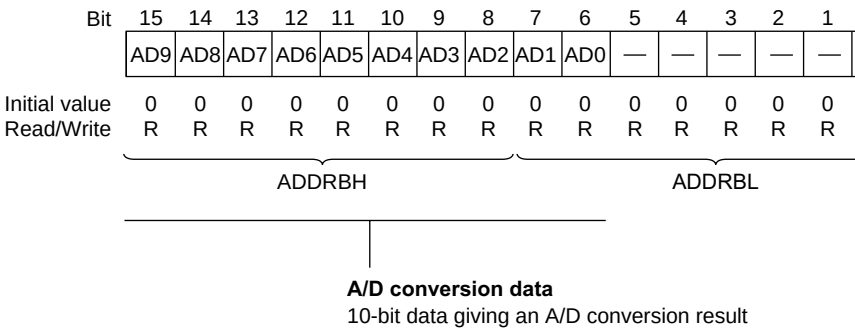
PADR—Port A Data Register **H'FFFD9**



ADDRA H/L—A/D Data Register A H/L **H'FFFE0, H'FFFE1**



ADDRB H/L—A/D Data Register B H/L **H'FFFE2, H'FFFE3**



A/D conversion data
10-bit data giving an A/D conversion result

ADDRD H/L—A/D Data Register D H/L

H'FFFE6, H'FFFE7

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
	ADDRDH								ADDRDL						

A/D conversion data
10-bit data giving an A/D conversion result

ADCR—A/D Control Register

H'FFFE9

Bit	7	6	5	4	3	2	1	
	TRGE	—	—	—	—	—	—	
Initial value	0	1	1	1	1	1	1	
Read/Write	R/W	—	—	—	—	—	—	R

Trigger Enable

0	A/D conversion start by external trigger or 8-bit timer compare match is disabled
1	A/D conversion is started by falling edge of external trigger signal ($\overline{\text{ADTRG}}$) or 8-bit timer compare match

CH2	CH1	CH0	Single Mode	Scan
0	0	0	AN_0	AN_0
		1	AN_1	$AN_{0,}$
	1	0	AN_2	AN_0 t
		1	AN_3	AN_0 t
1	0	0	AN_4	AN_4
		1	AN_5	$AN_{4,}$
	1	0	AN_6	AN_4 t
		1	AN_7	AN_4 t

Clock select

0	Conversion time = 134 states (maximum)
1	Conversion time = 70 states (maximum)

Scan mode

0	Single mode
1	Scan mode

A/D start

0	A/D conversion is stopped
1	A/D conversion is stopped Single mode: A/D conversion starts; ADST is automatically clea when conversion ends Scan mode: A/D conversion starts and continues, cycling amo selected channels ADST is cleared to 0 by softwa by a reset, or by a transition to standby mode

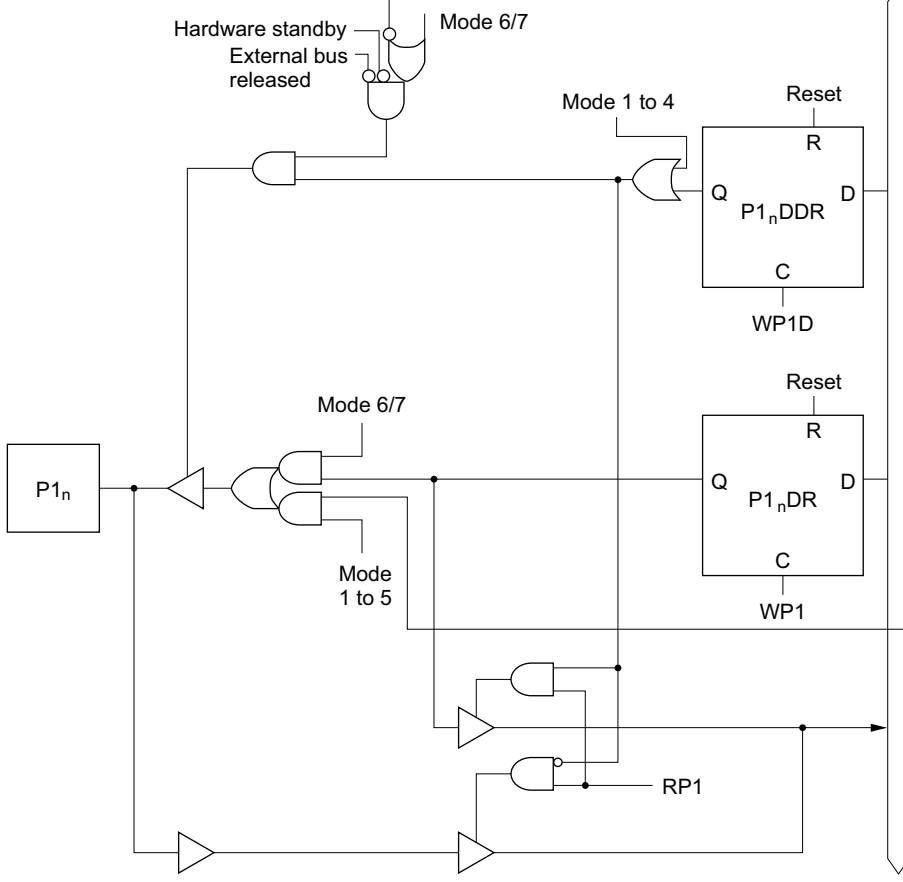
A/D interrupt enable

0	A/D end interrupt request is disabled
1	A/D end interrupt request is enabled

A/D end flag

0	[Clearing conditions] • Read ADF when ADF = 1, then write 0 in ADF • The DMAC is activated by an ADI interrupt
1	[Setting conditions] • Single mode: A/D conversion ends • Scan mode: A/D conversion ends in all selected channels

Note: * Only 0 can be written, to clear the flag.



WP1D: Write to P1DDR
WP1: Write to port 1
RP1: Read port 1
SSOE: Software standby output port enable
n = 0 to 7

Figure C.1 Port 1 Block Diagram

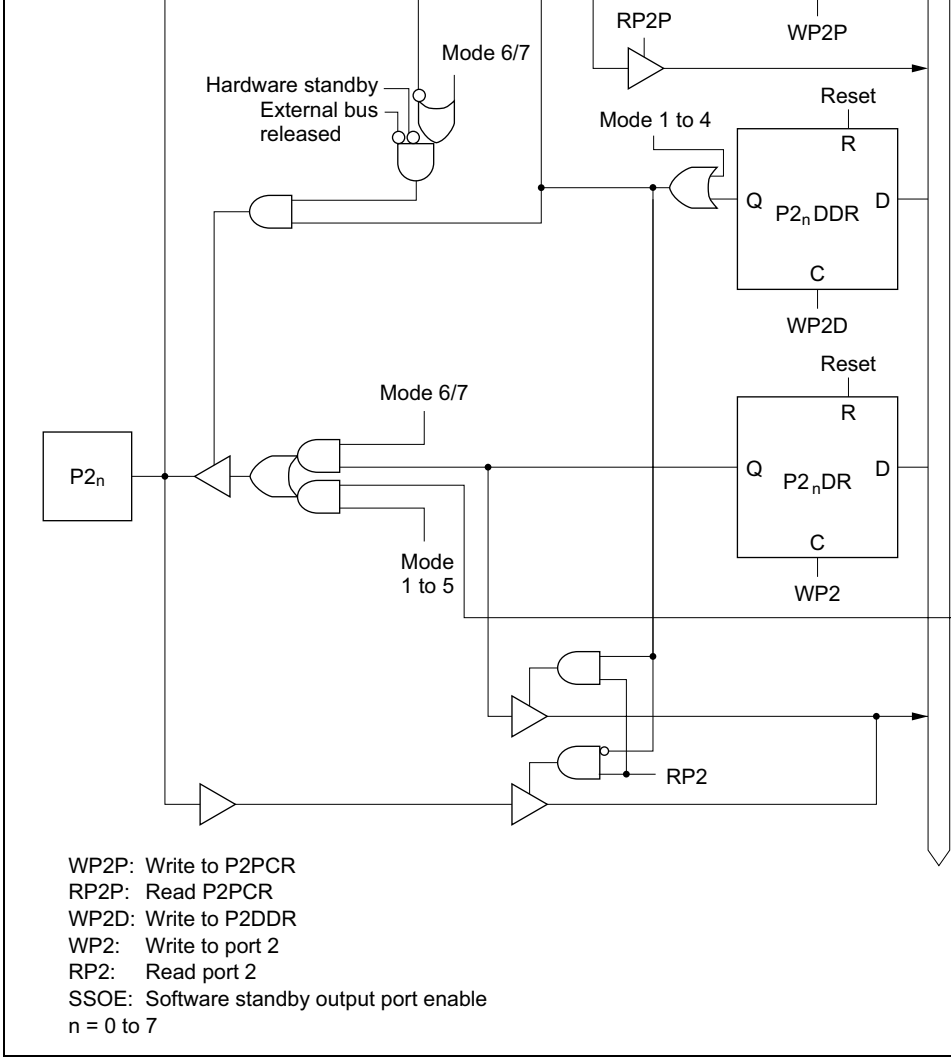


Figure C.2 Port 2 Block Diagram

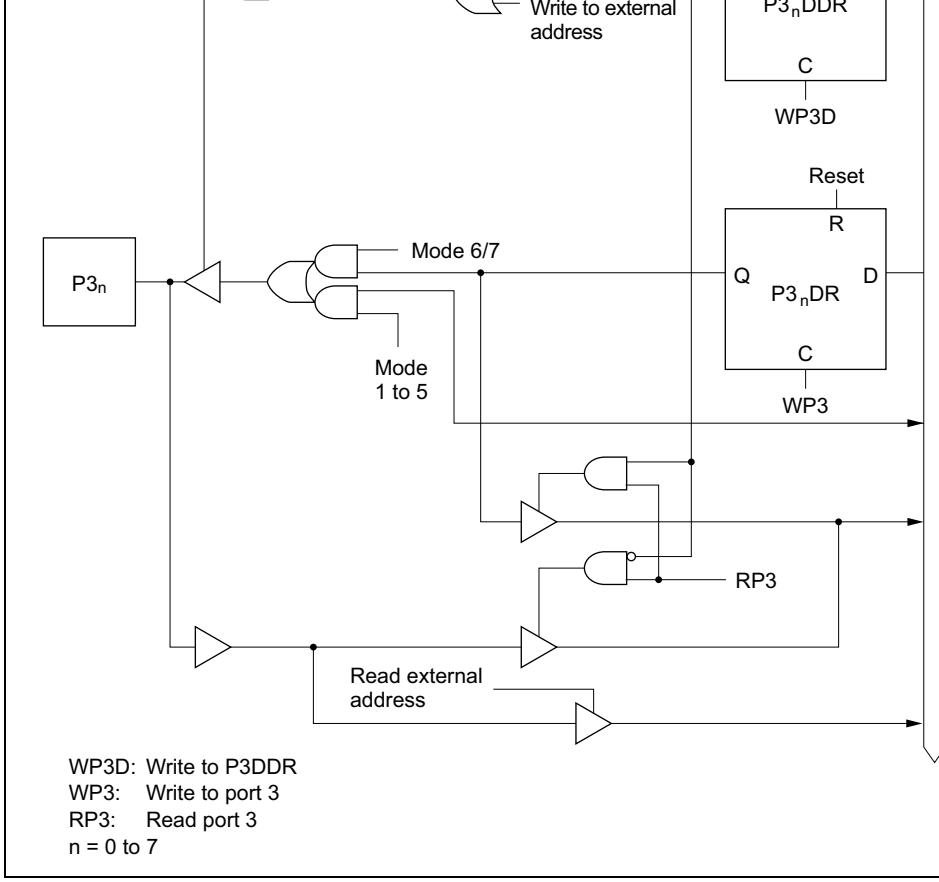


Figure C.3 Port 3 Block Diagram

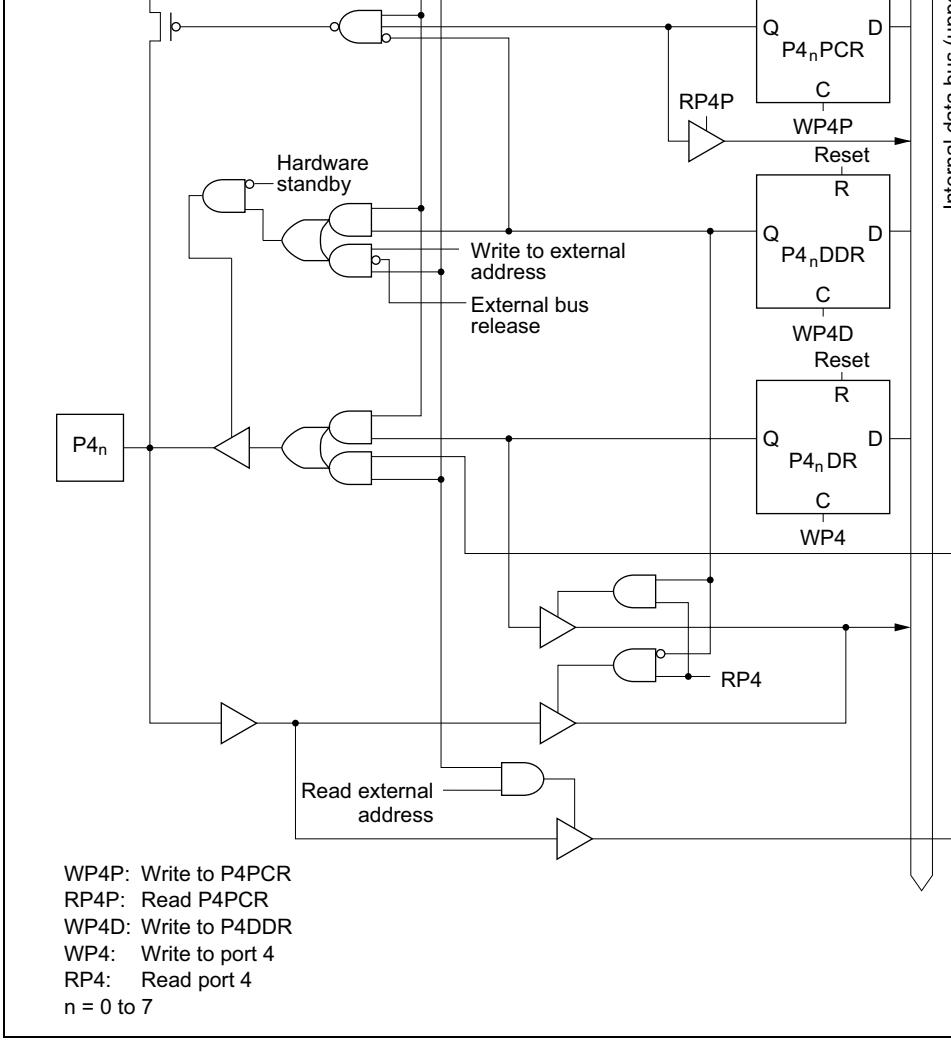
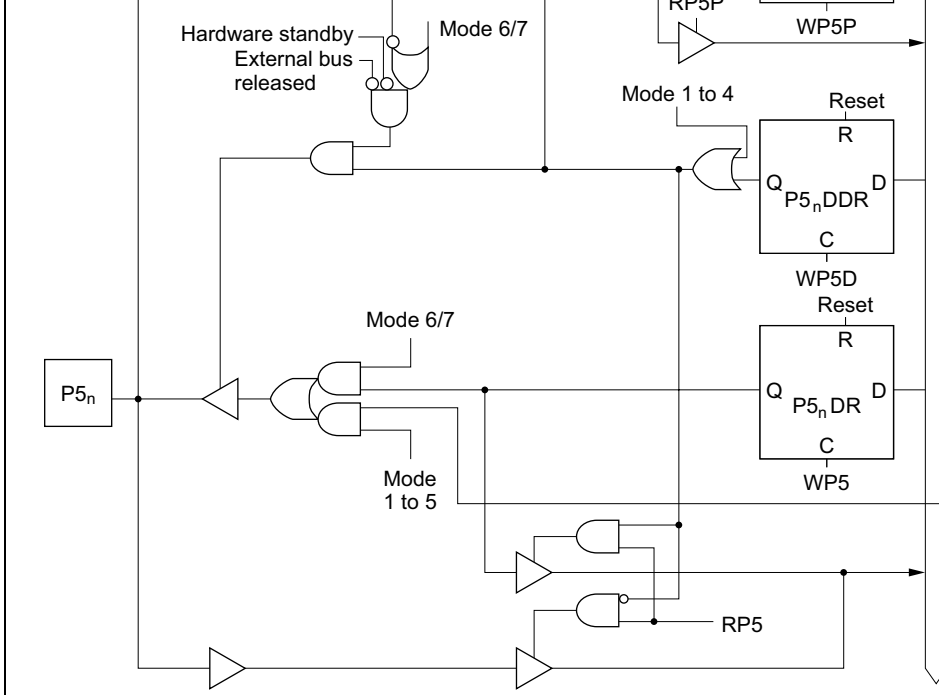


Figure C.4 Port 4 Block Diagram



WP5P: Write to P5PCR
RP5P: Read P5PCR
WP5D: Write to P5DDR
WP5: Write to port 5
RP5: Read port 5
SSOE: Software standby output port enable
n = 0 to 3

Figure C.5 Port 5 Block Diagram

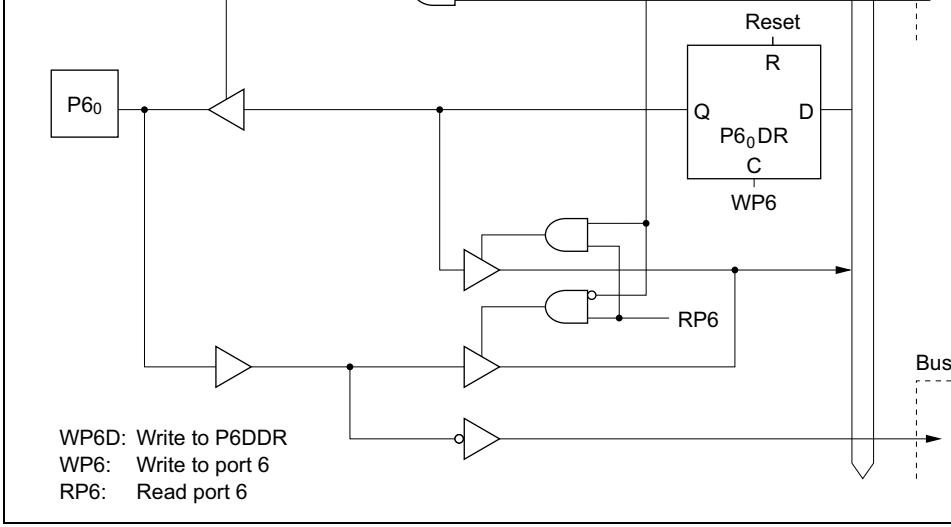


Figure C.6 (a) Port 6 Block Diagram (Pin $P6_0$)

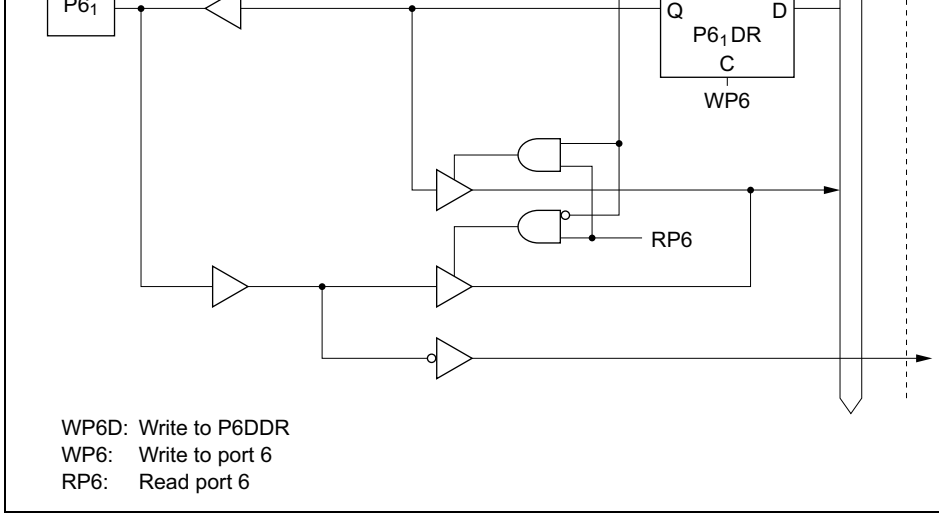


Figure C.6 (b) Port 6 Block Diagram (Pin P6$_1$)

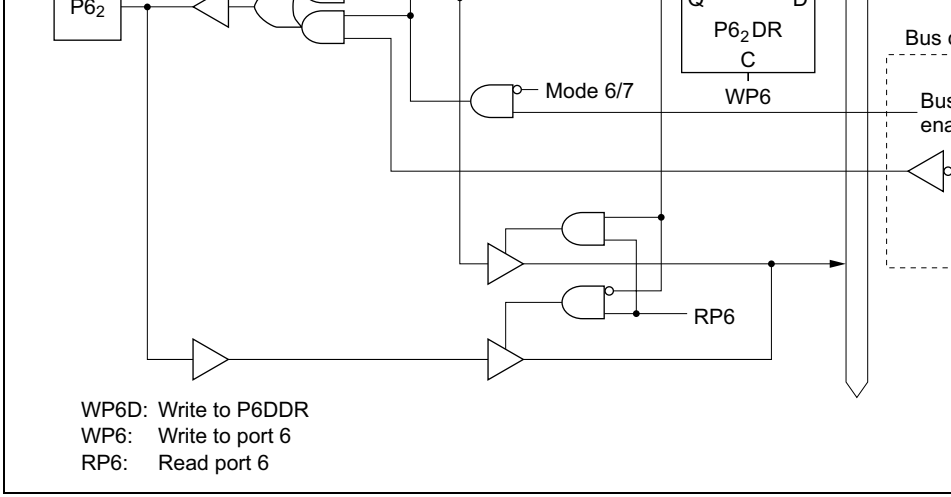


Figure C.6 (c) Port 6 Block Diagram (Pin P6$_2$)

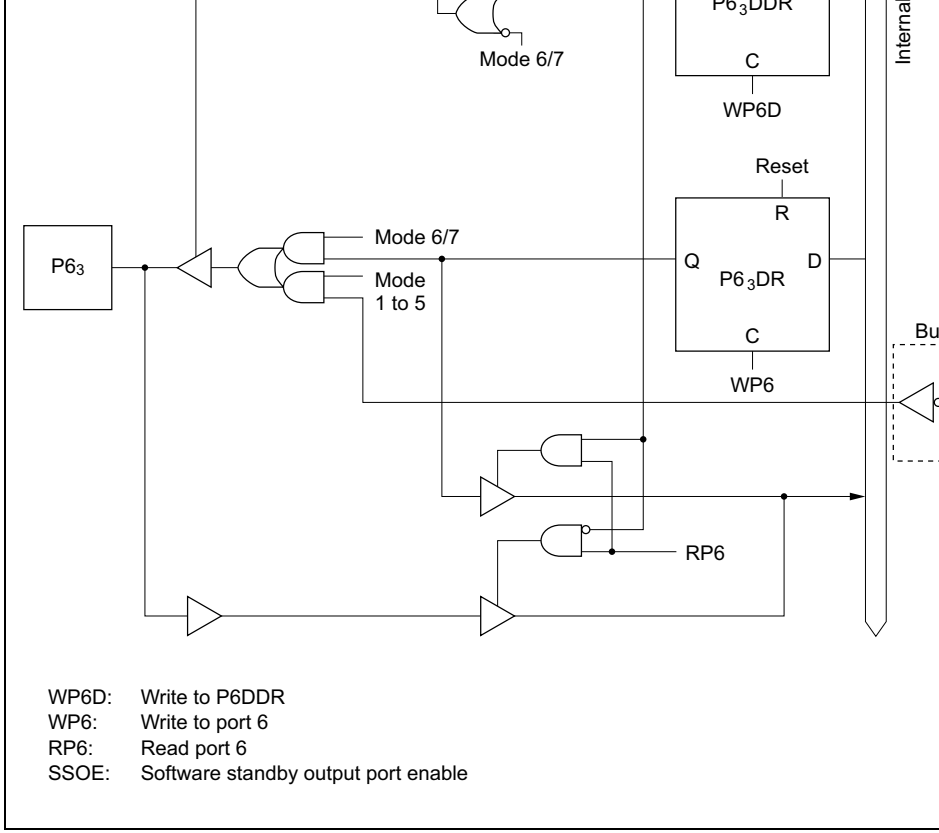


Figure C.6 (d) Port 6 Block Diagram (Pin $P6_3$)

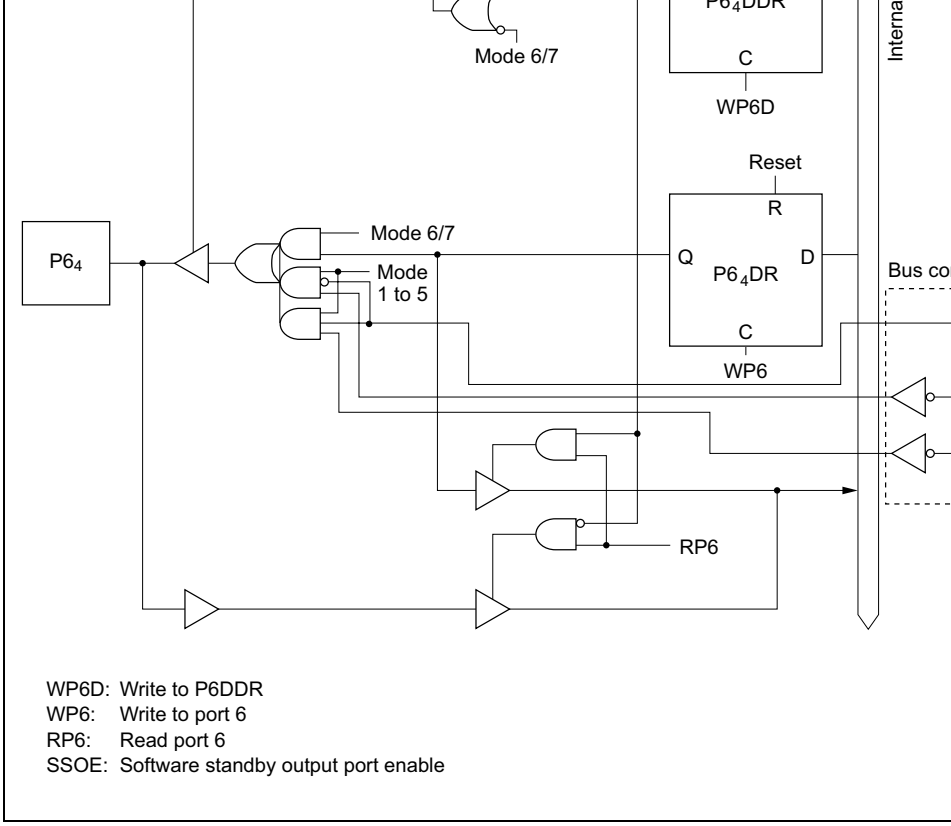


Figure C.6 (e) Port 6 Block Diagram (Pin P6$_4$)

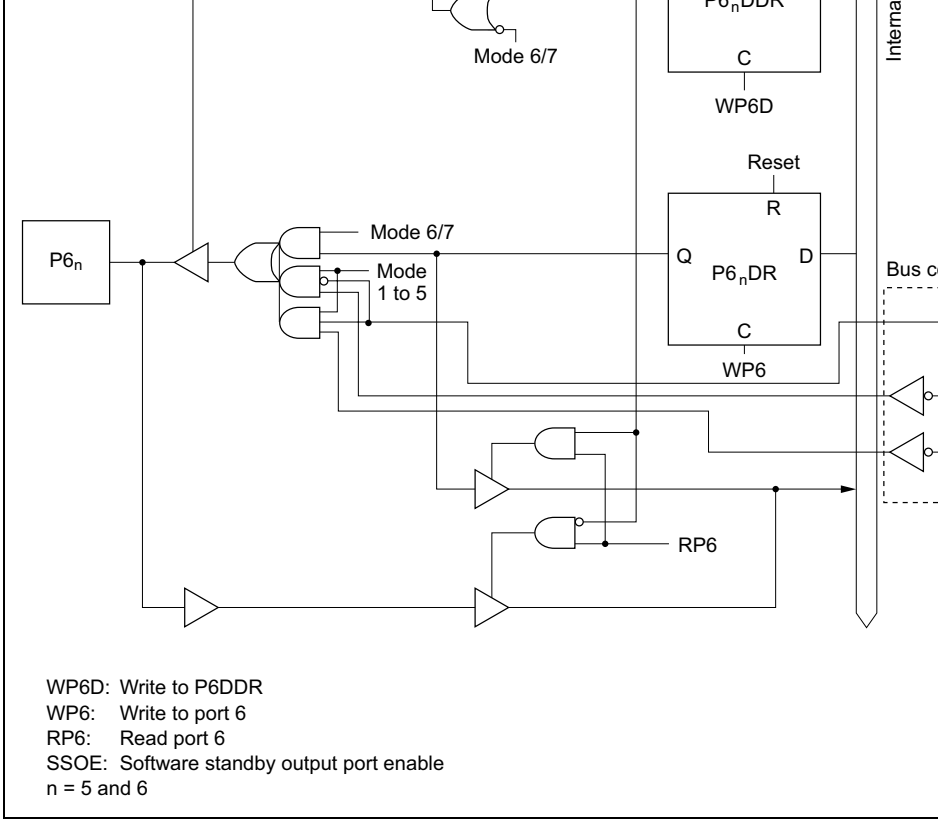


Figure C.6 (f) Port 6 Block Diagram (Pins $P6_5$ and $P6_6$)

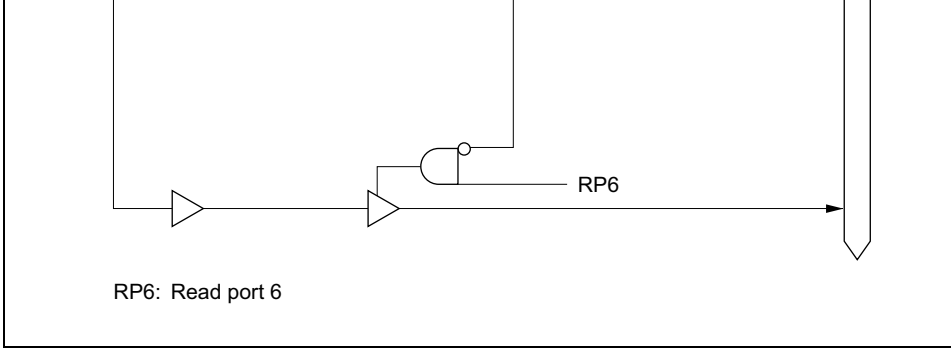


Figure C.6 (g) Port 6 Block Diagram (Pin $P6_7$)

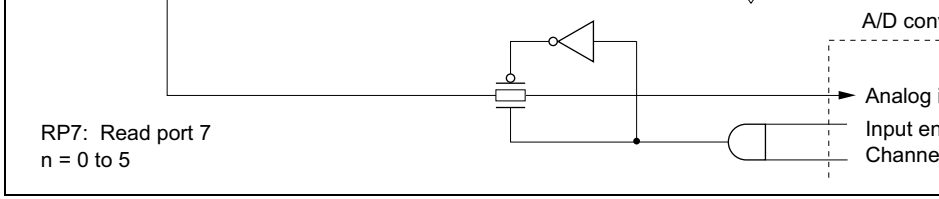


Figure C.7 (a) Port 7 Block Diagram (Pins $P7_0$ to $P7_5$)

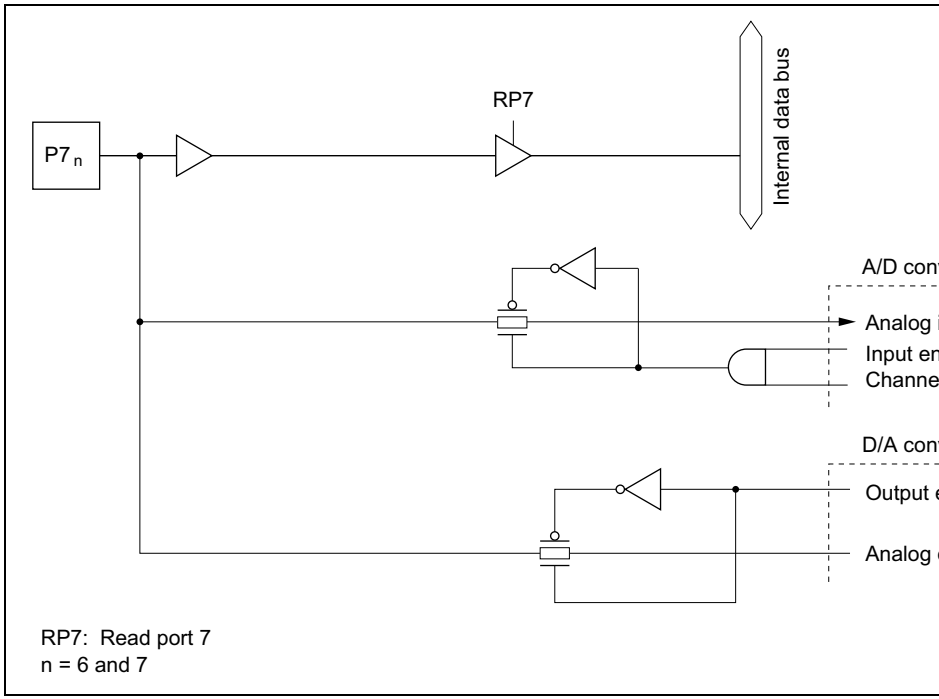


Figure C.7 (b) Port 7 Block Diagram (Pins $P7_6$ and $P7_7$)

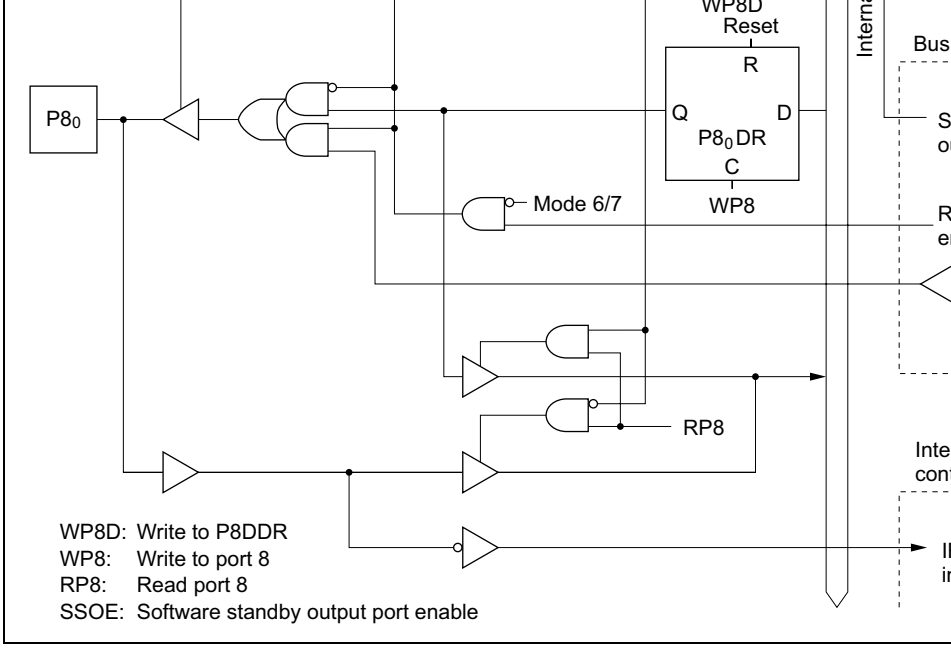


Figure C.8 (a) Port 8 Block Diagram (Pin $P8_0$)

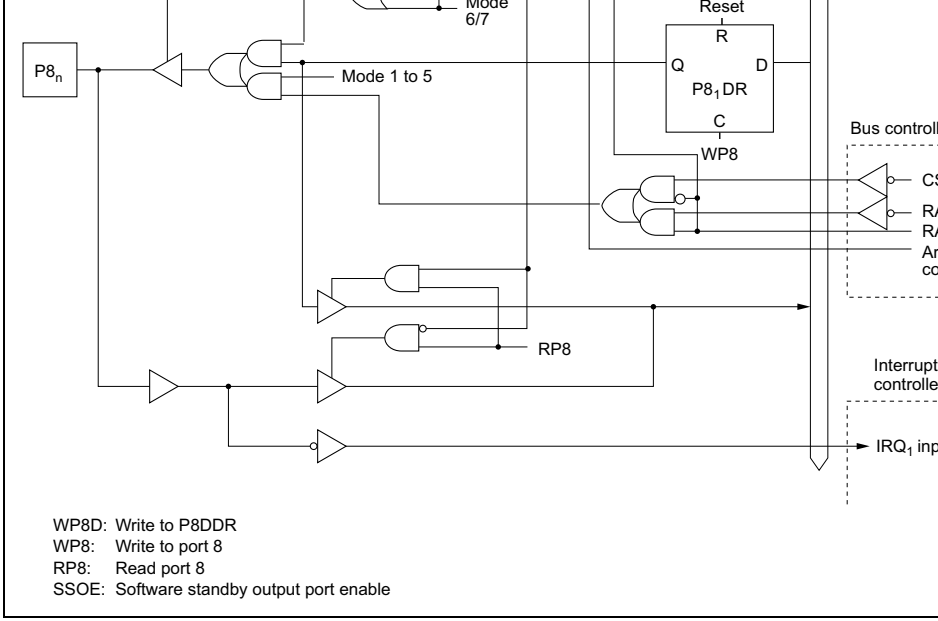


Figure C.8 (b) Port 8 Block Diagram (Pin P8$_1$)

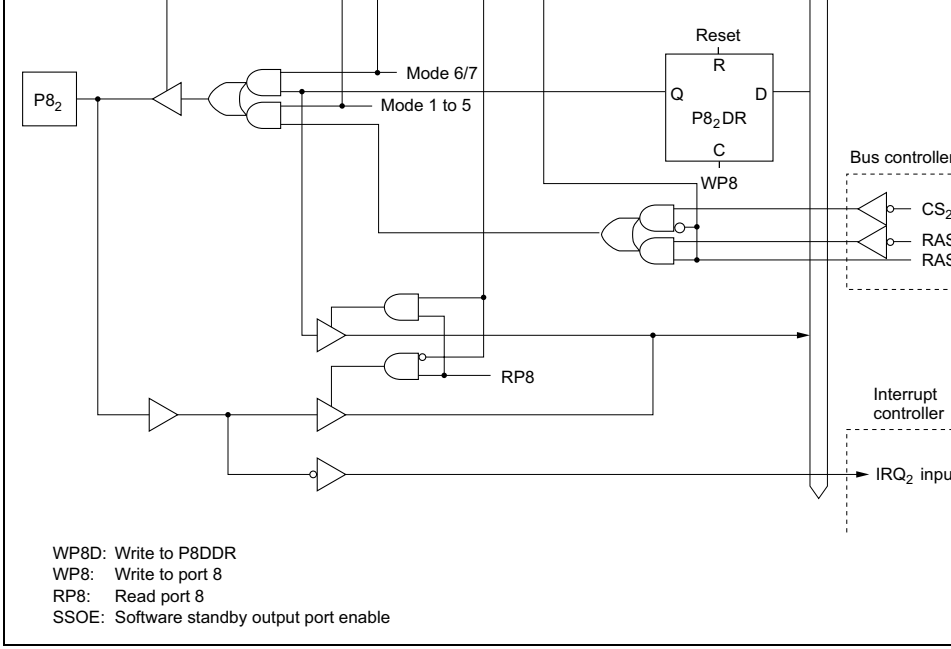
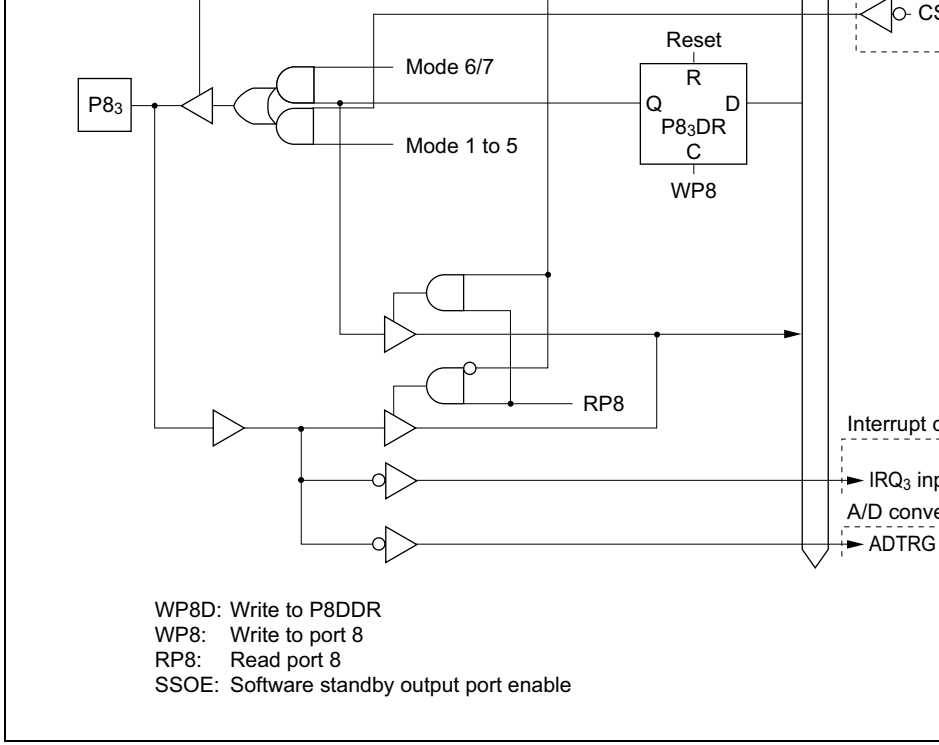
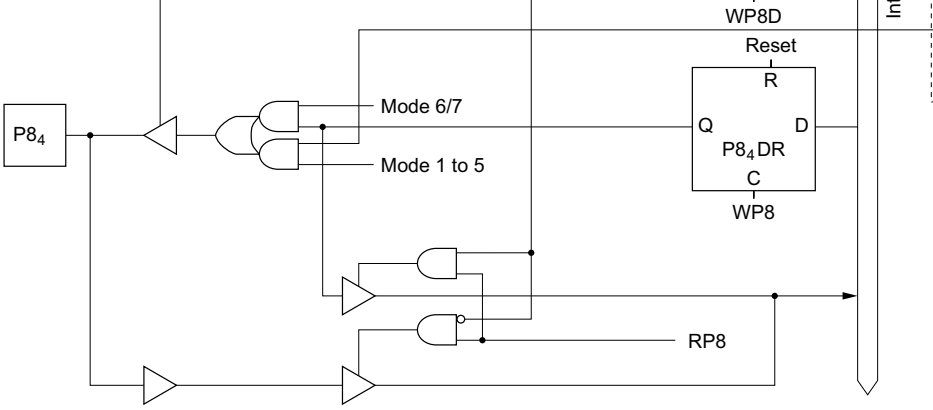


Figure C.8 (c) Port 8 Block Diagram (Pin P8$_2$)



WP8D: Write to P8DDR
WP8: Write to port 8
RP8: Read port 8
SSOE: Software standby output port enable

Figure C.8 (d) Port 8 Block Diagram (Pin $P8_3$)



WP8D: Write to P8DDR
WP8: Write to port 8
RP8: Read port 8
SSOE: Software standby output port enable

Figure C.8 (e) Port 8 Block Diagram (Pin P8$_4$)

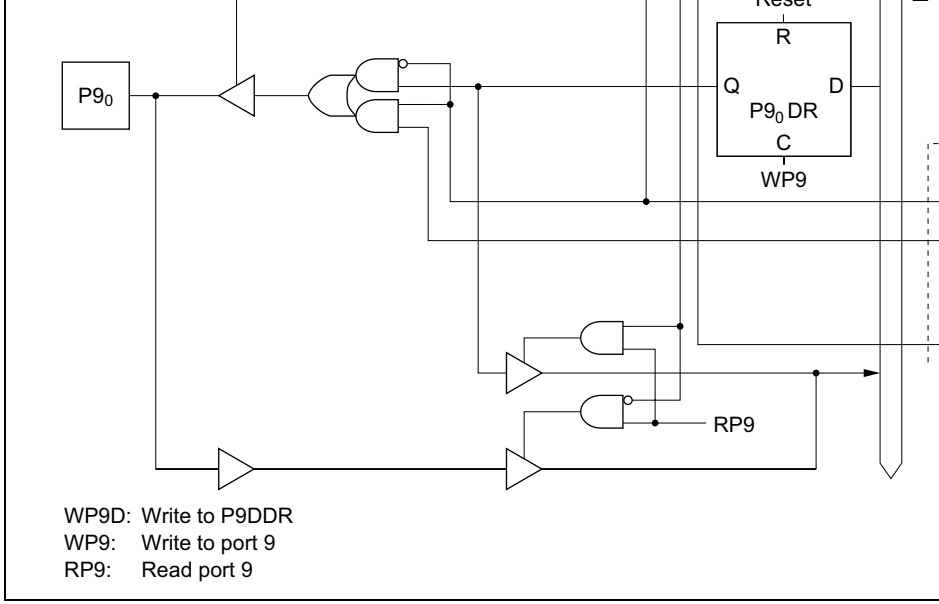
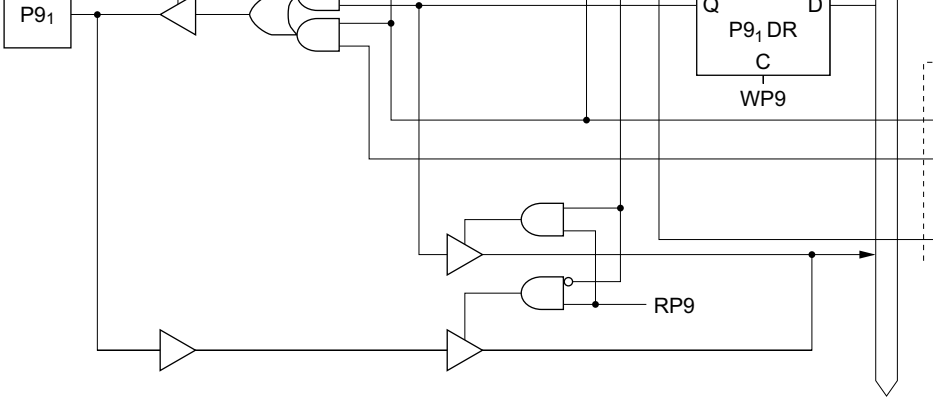


Figure C.9 (a) Port 9 Block Diagram (Pin P9$_0$)



WP9D: Write to P9DDR
WP9: Write to port 9
RP9: Read port 9

Figure C.9 (b) Port 9 Block Diagram (Pin P9$_1$)

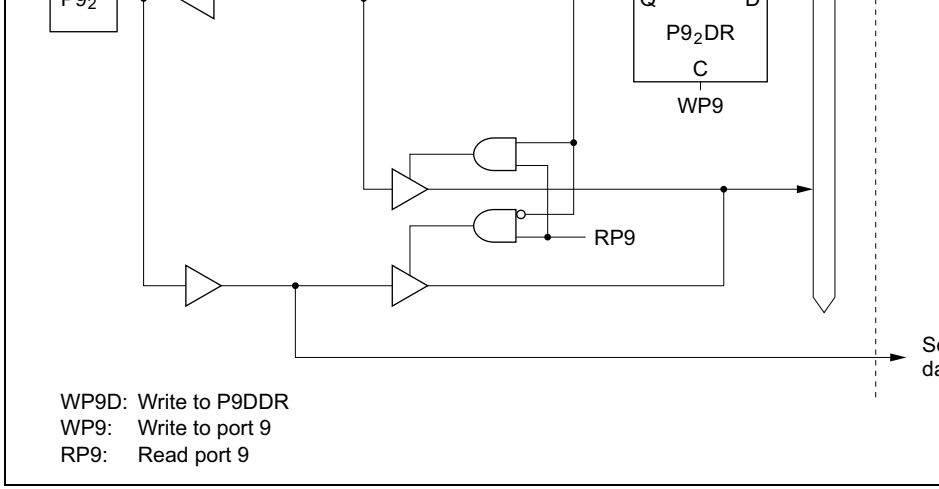


Figure C.9 (c) Port 9 Block Diagram (Pin P9$_2$)

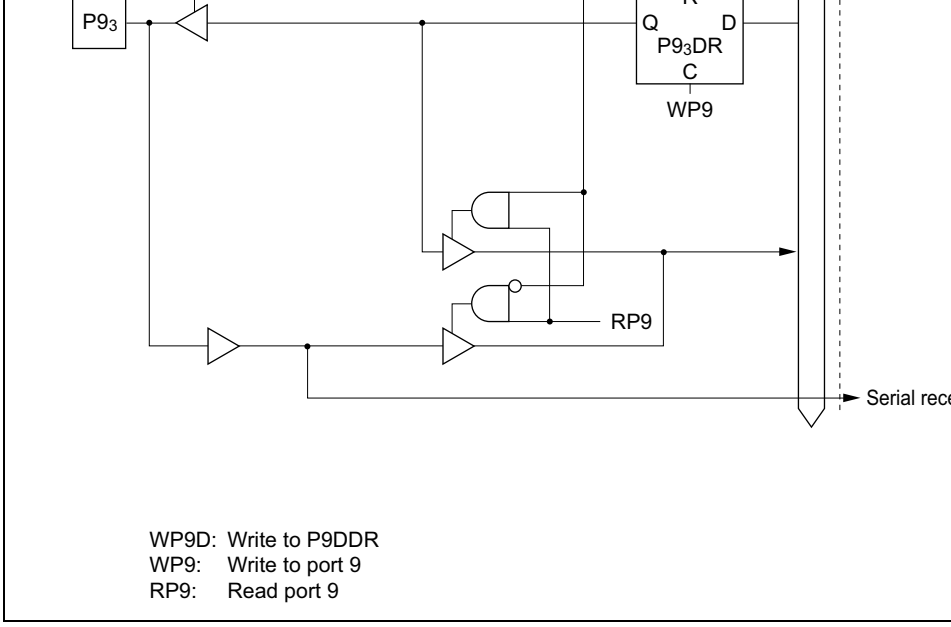


Figure C.9 (d) Port 9 Block Diagram (Pin P9$_3$)

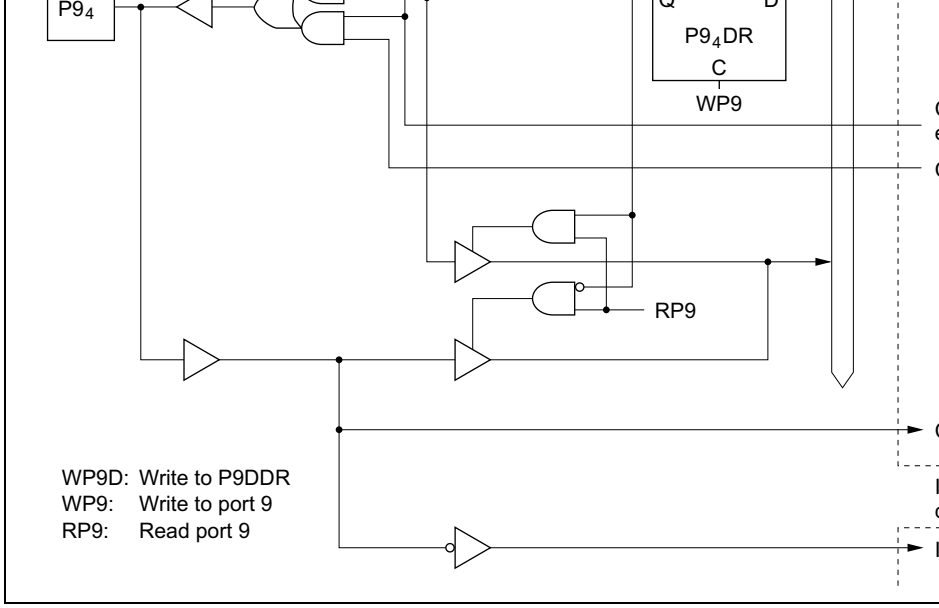


Figure C.9 (e) Port 9 Block Diagram (Pin P9$_4$)

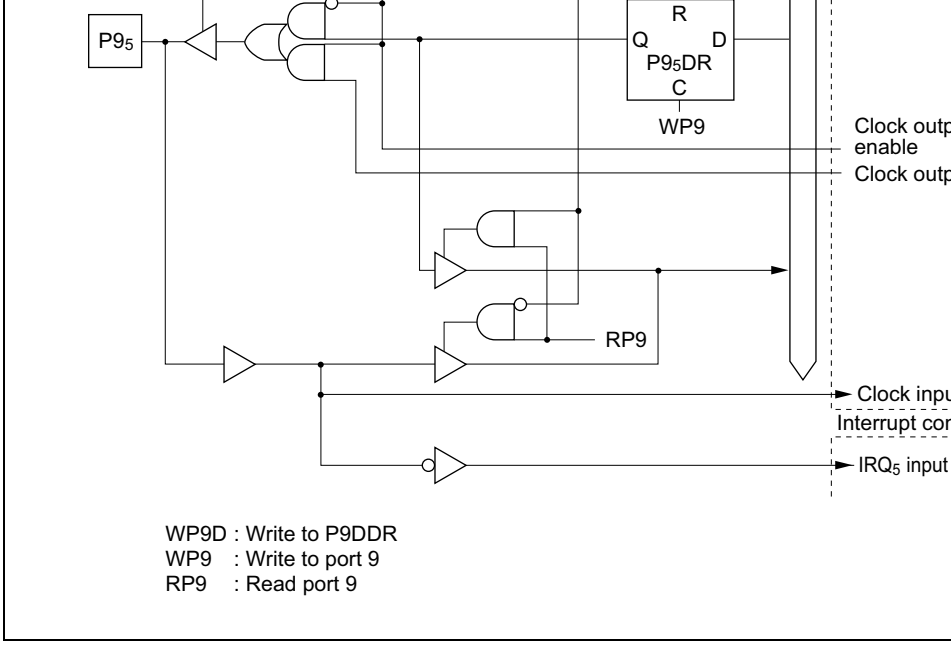


Figure C.9 (f) Port 9 Block Diagram (Pin P9$_5$)

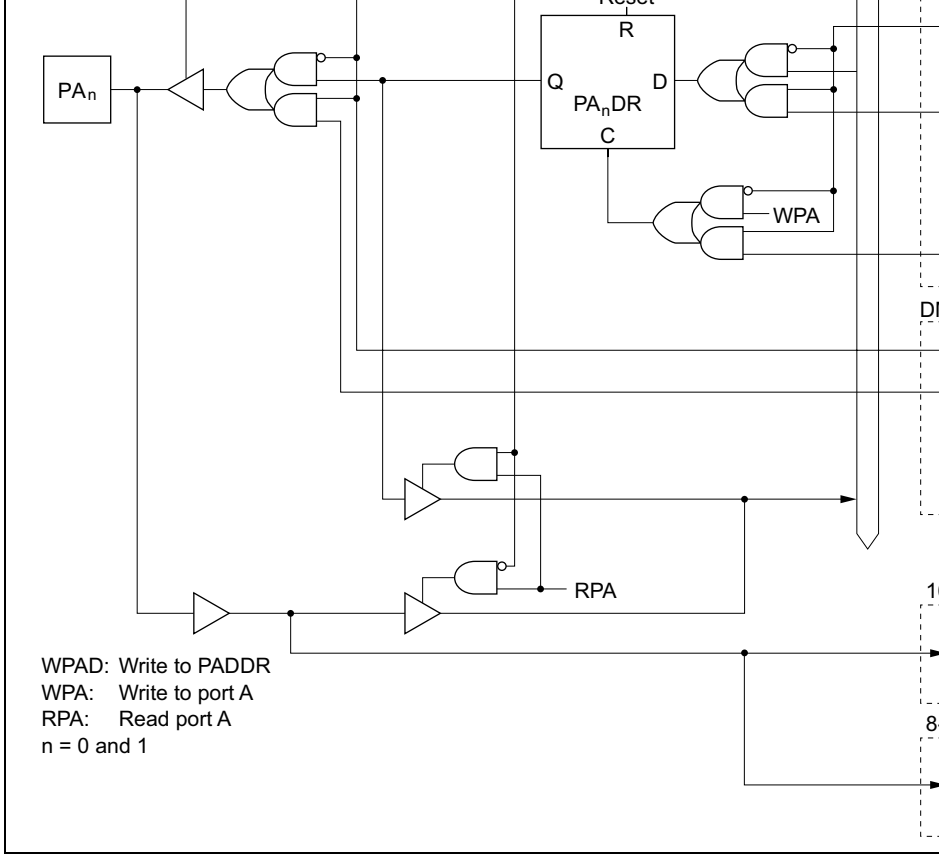


Figure C.10 (a) Port A Block Diagram (Pins PA_0, PA_1)

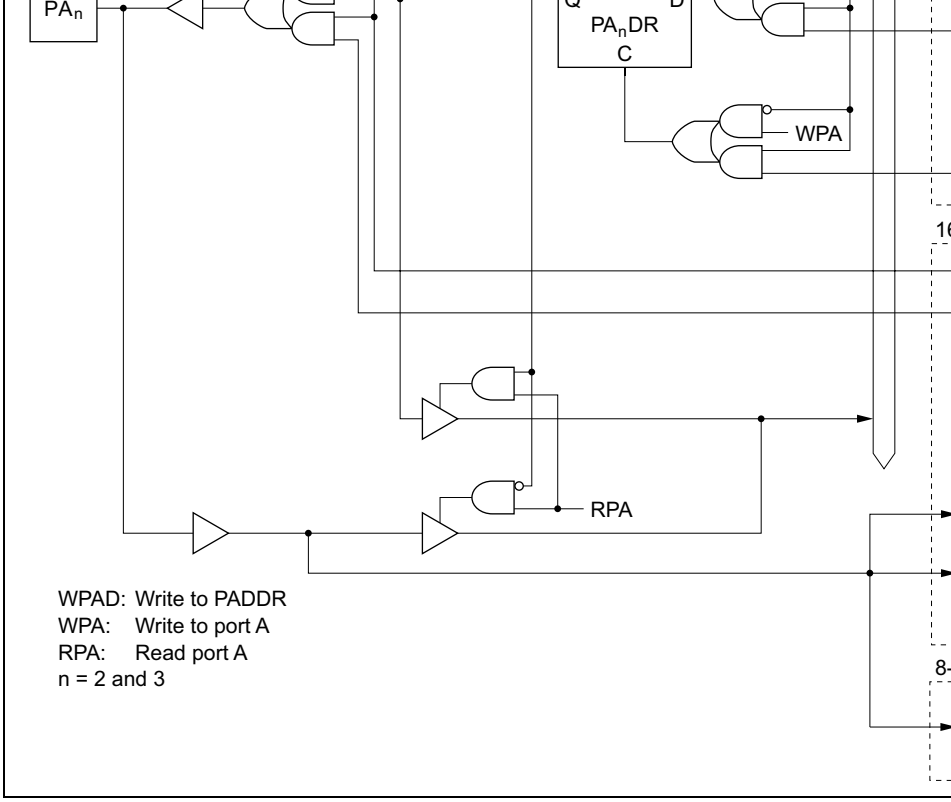
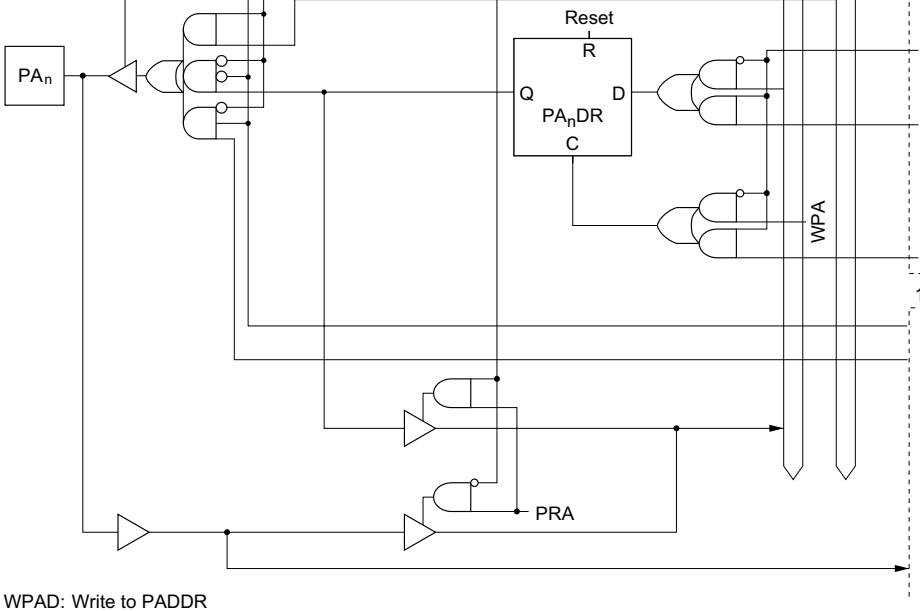


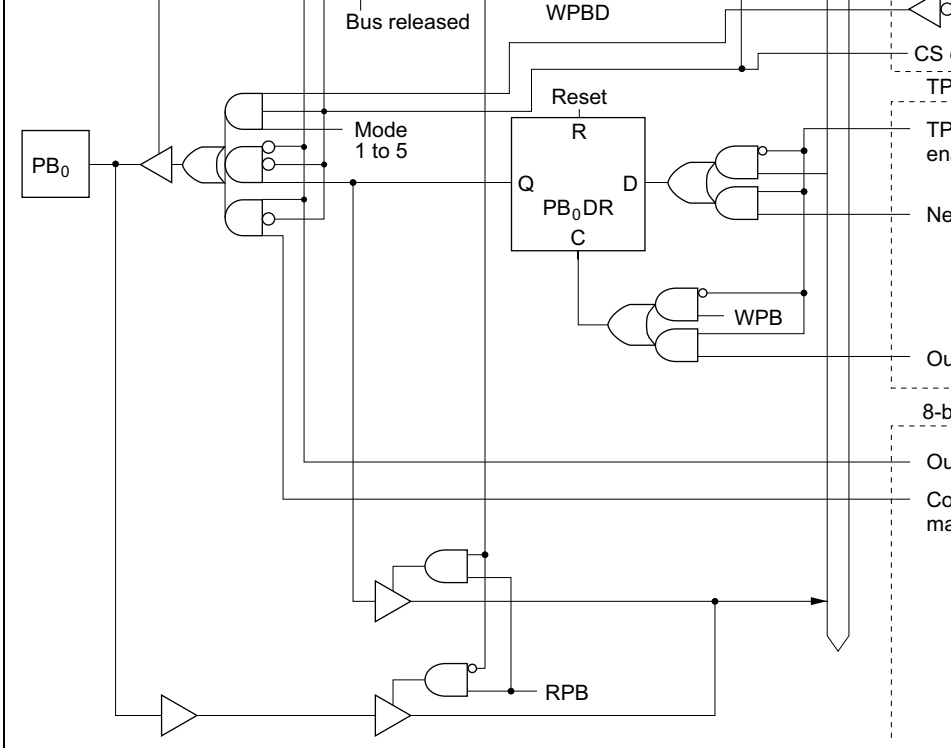
Figure C.10 (b) Port A Block Diagram (Pins PA_2, PA_3)



WPAD: Write to PADDR
WPA: Write to port A
RPA: Read port A
SSOE: Software standby output port enable
n = 4 to 7

Note: The PA_7 address output enable setting is fixed at 1 in modes 3 and 4.

Figure C.10 (c) Port A Block Diagram (Pins PA_4 to PA_7)



WPBD: Write to PBDDR
WPB: Write to port B
RPB: Read port B
SSOE: Software standby output port enable

Figure C.11 (a) Port B Block Diagram (Pin PB_0)

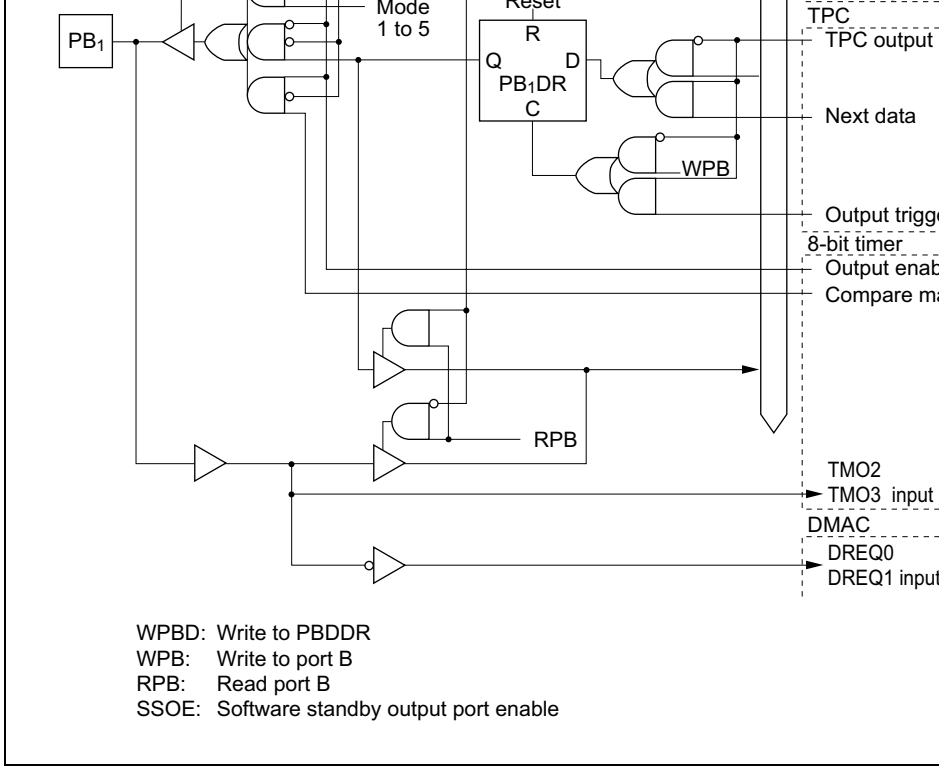
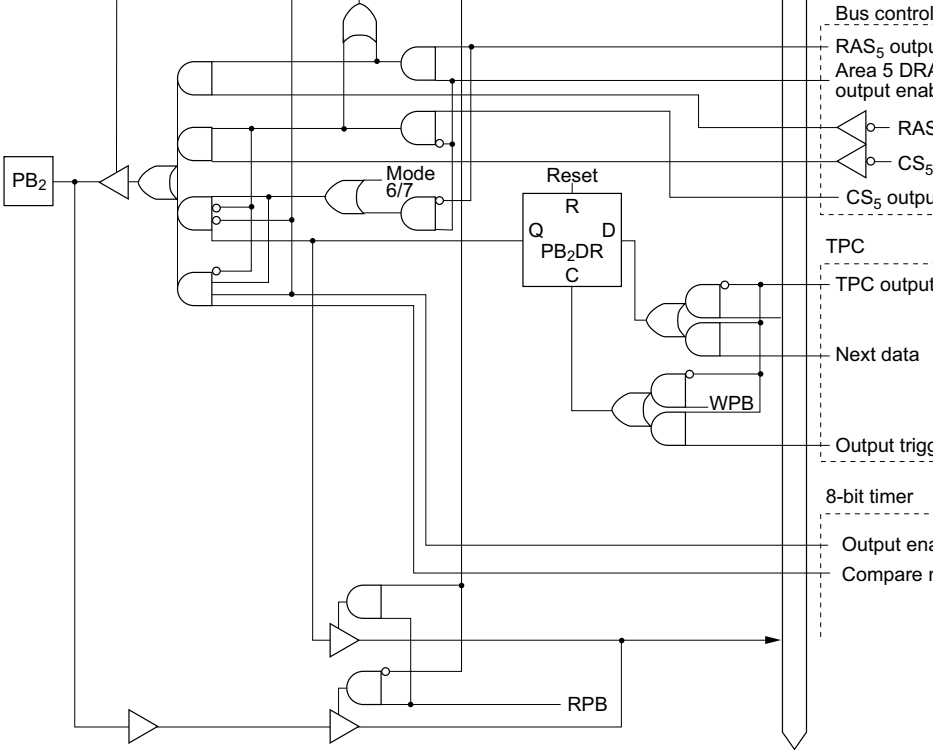


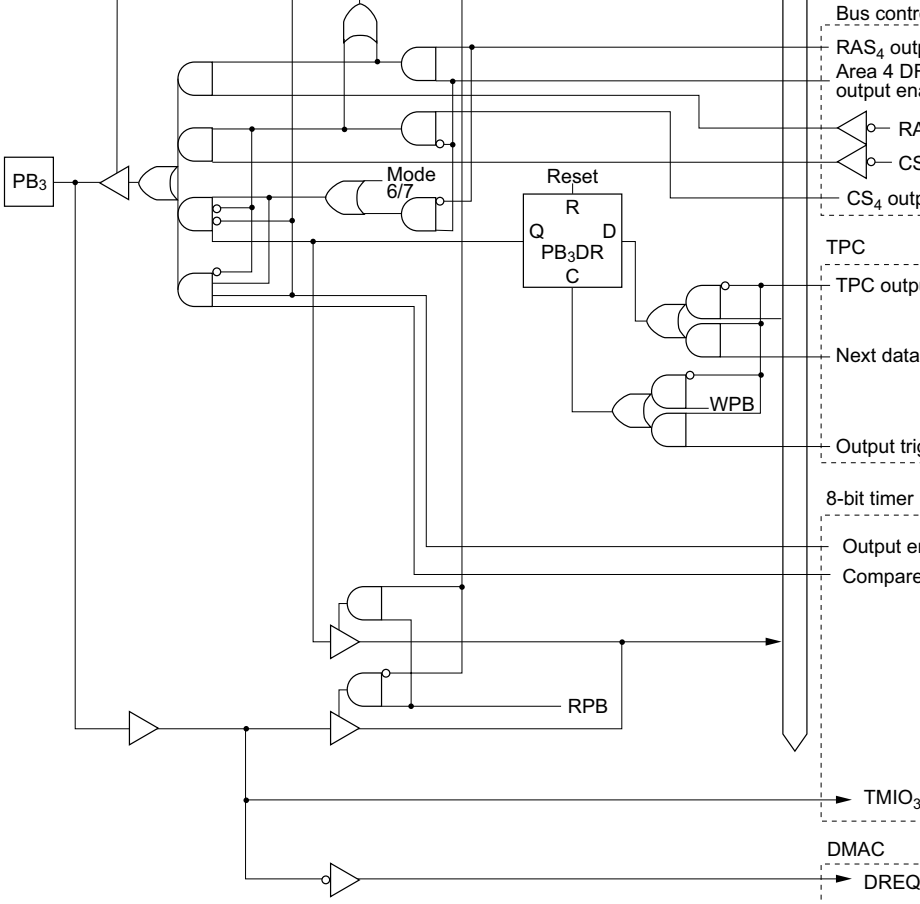
Figure C.11 (b) Port B Block Diagram (Pin PB_1)



WPBD: Write to PBDDR
WPB: Write to port B
RPB: Read port B
SSOE: Software standby output port enable

Note: Area 5 DRAM connection enable, RAS_5 output enable, and CS_5 output enable are fixed at 0 in modes 6 and 7.

Figure C.11 (c) Port B Block Diagram (Pin PB_2)



WPBD: Write to PBDDR
WPB: Write to port B
RPB: Read port B
SSOE: Software standby output port enable

Note: Area 4 DRAM connection enable, RAS_4 output enable, and CS_4 output enable are fixed at
in modes 6 and 7.

Figure C.11 (d) Port B Block Diagram (Pin PB_3)

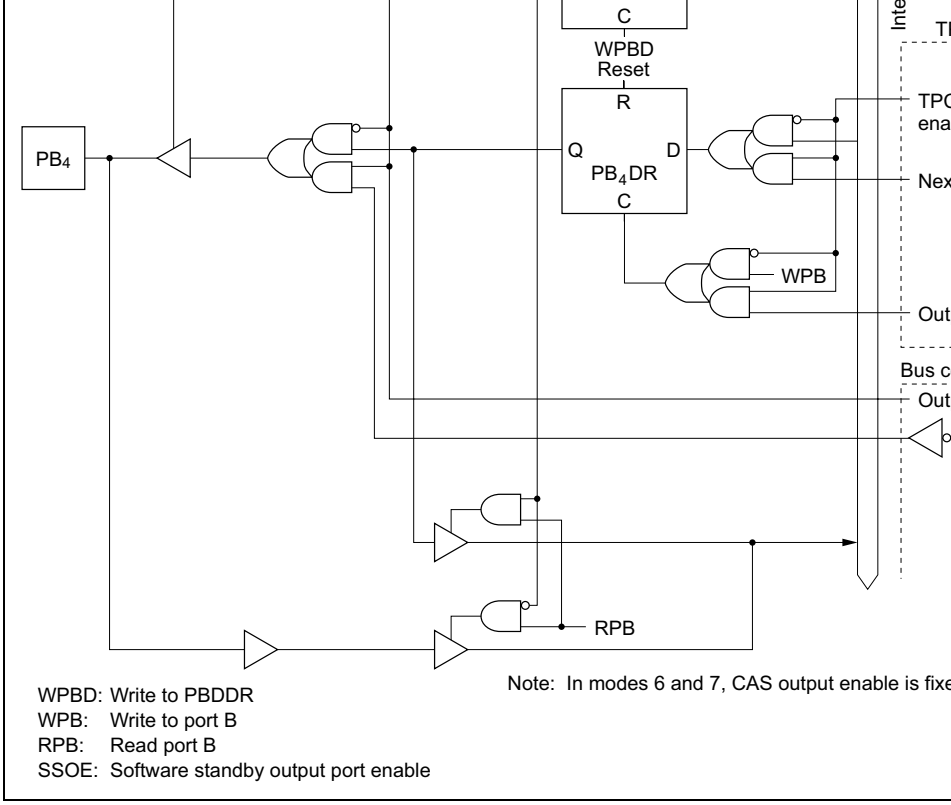


Figure C.11 (e) Port B Block Diagram (Pin PB_4)

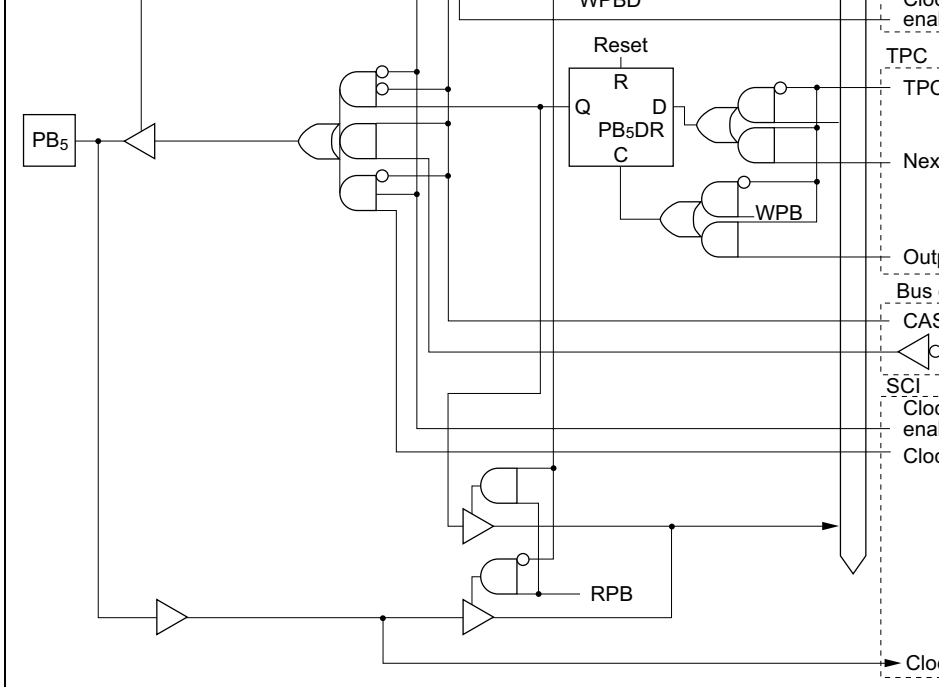


Figure C.11 (f) Port B Block Diagram (Pin PB_5)

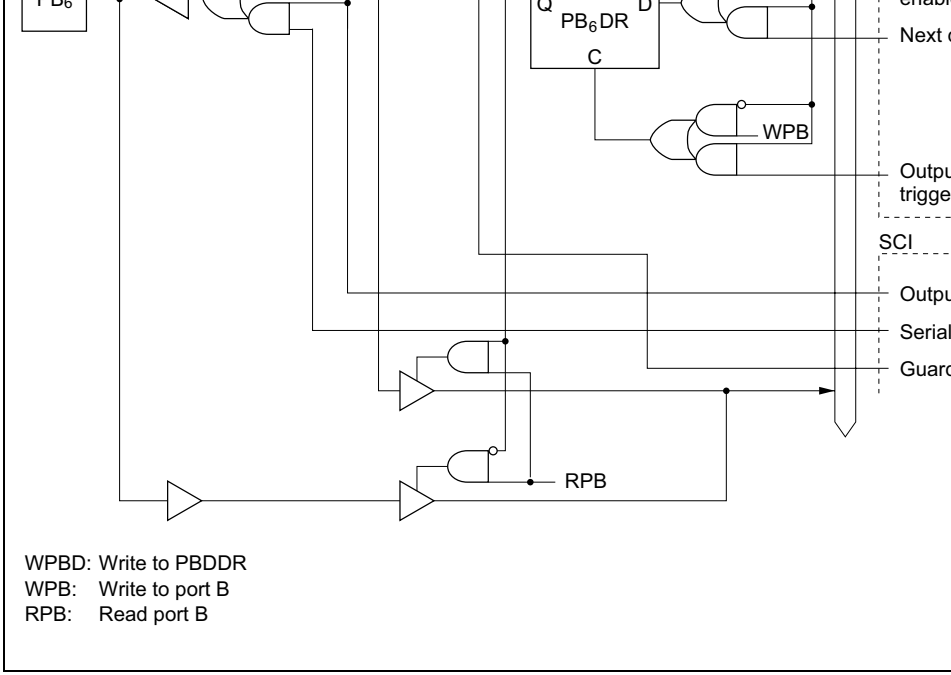


Figure C.11 (g) Port B Block Diagram (Pin PB_6)

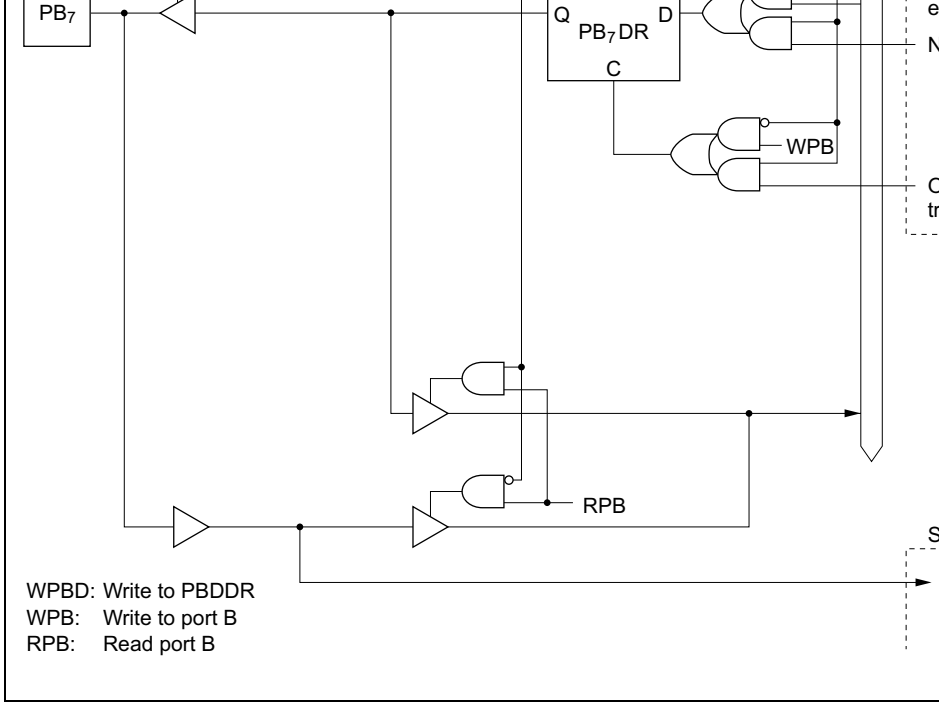


Figure C.11 (h) Port B Block Diagram (Pin PB_7)

$P1_7$ to $P1_0$	1 to 4	L	T	(SSOE=0) T (SSOE=1) Keep	T	A_7 to A_0
	5	T	T	(DDR = 0) Keep (DDR=1, SSOE=0) T (DDR=1, SSOE=1) Keep	T	(DDR=0) Input port (DDR=1) A_7 to A_0
	6, 7	T	T	Keep	—	I/O port
$P2_7$ to $P2_0$	1 to 4	L	T	(SSOE = 0) T (SSOE = 1) Keep	T	A_{15} to A_8
	5	T	T	(DDR = 0) Keep (DDR=1,SSOE=0) T (DDR=1,SSOE=1) Keep	T	(DDR=0) Input port (DDR=1) A_{15} to A_8
	6, 7	T	T	Keep	—	I/O port
$P3_7$ to $P3_0$	1 to 5	T	T	T	T	D_{15} to D_8
	6, 7	T	T	Keep	—	I/O port
$P4_7$ to $P4_0$	1, 3, 5	T	T	Keep	Keep	I/O port
	2, 4	T	T	T	T	D_7 to D_0
	6, 7	T	T	Keep	—	I/O port

				(DDR=1, SSOE=1) Keep		
	6, 7	T	T	Keep	—	I/O port
$P6_0$	1 to 5	T	T	Keep	Keep	I/O port $\overline{WAIT}$
	6, 7	T	T	Keep	—	I/O port
$P6_1$	1 to 5	T	T	(BRLE=0) Keep (BRLE=1) T	T	I/O port $\overline{BREQ}$
	6, 7	T	T	Keep	—	I/O port
$P6_2$	1 to 5	T	T	(BRLE=0) Keep (BRLE=1) H	L	(BRLE=0) I/O port (BRLE=1) $\overline{BACK}$
	6, 7	T	T	Keep	—	I/O port
$P6_6$ to $P6_3$	1 to 5	H	T	(SSOE=0) T (SSOE=1) H	T	$\overline{AS}$, $\overline{RD}$, $\overline{HWR}$, $\overline{LW}$
	6, 7	T	T	Keep	—	I/O port
$P6_7$	1 to 7	Clock output	T	(PSTOP=0) H (PSTOP=1) Keep	(PSTOP=0) ϕ (PSTOP=1) Keep	(PSTOP= ϕ (PSTOP= Input port
$P7_7$ to $P7_0$	1 to 7	T	T	T	T	Input port

				(RFSHE=0) Keep (RFSHE=1, SRFMD=0, SSOE=0) T (RFSHE=1, SRFMD=0, SSOE=1) H (RFSHE=1, SRFMD=1) $\overline{RFSH}$	(RFSHE=0) Keep (RFSHE=1) T	
	6, 7	T	T	Keep	—	I/O port
$P8_1$	1 to 5	T	T	• When DRAM space is selected*3 (SSOE=0) T (SSOE=1) H • When DRAM space is selected*4 Keep • Otherwise*5 *1 (DDR=0) T (DDR=1, SSOE=0) T (DDR=1, SSOE=1) H	• When DRAM space is selected*3 T • When DRAM space is selected*4 Keep • Otherwise*1 (DDR=0) Keep (DDR=1) T	• When DR is selecte is output $\overline{RAS}_3$ • When DR is selecte is not out I/O port • Otherwis (DDR=0) Input por (DDR=1) $\overline{CS}_3$
	6, 7	T	T	Keep	—	I/O port

				(DDR=1, SSOE=0) T (DDR=1, SSOE=1) H		
	6, 7	T	T	Keep	—	I/O port
$P8_3$	1 to 5	T	T	(DDR=0) T (DDR=1, SSOE=0) T (DDR=1, SSOE=1) H	(DDR=0) Keep (DDR=1) T	(DDR=0) Input port (DDR=1) $\overline{CS}_1$
	6, 7	T	T	Keep	—	I/O port
$P8_4$	1 to 4	H	T	(DDR=0) T (DDR=1, SSOE=0) T (DDR=1, SSOE=1) H	(DDR = 0) Keep (DDR = 1) T	(DDR = 0) Input port (DDR = 1) $\overline{CS}_0$
	5	T	T	(DDR=0) T (DDR=1, SSOE=0) T (DDR=1, SSOE=1) H	(DDR=0) Keep (DDR=1) T	(DDR=0) Input port (DDR=1) $\overline{CS}_0$
	6, 7	T	T	Keep	—	I/O port
$P9_5$ to $P9_0$	1 to 7	T	T	Keep	Keep	I/O port
PA_3 to PA_0	1 to 7	T	T	Keep	Keep	I/O port
PA_6 to PA_4	1, 2, 6, 7	T	T	Keep	Keep	I/O port

PA_7	1, 2	T	T	Keep	Keep	I/O port
	3, 4	L	T	(SSOE=0) T (SSOE=1) Keep	T	A_{20}
	5	L	T	• When A20E = 0 SSOE = 0 T SSOE = 1 Keep • When A20E = 1 Keep	• When A20E = 0 T • When A20E = 1 Keep	• When A2 A_{20} • When A2 I/O port
	6, 7	T	T	Keep	—	I/O port
PB_1 to PB_0	1 to 5	T	T	• CS output*7 (SSOE=0) T (SSOE=1) H • Otherwise*8 Keep	• CS output*7 T • Otherwise*8 Keep	• CS outpu $\overline{CS}_7$ to $\overline{C}$ • Otherwis I/O port
	6, 7	T	T	Keep	—	I/O port
PB_2	1 to 5	T	T	• RAS_5 output*9 (SSOE=0) T (SSOE=1) H • CS output*10 (SSOE=0) T (SSOE=1) H • Otherwise*11 Keep	• RAS_5 output*9 T • CS output*10 T • Otherwise*11 Keep	• RAS_5 ou $\overline{RAS}_5$ • CS outpu $\overline{CS}_5$ • Otherwis I/O port
	6, 7	T	T	Keep	—	I/O port

				(SSOE=1) H • Otherwise*14 Keep		
	6, 7	T	T	Keep	—	I/O port
PB_5 to PB_4	1 to 5	T	T	• CAS output*15 (SSOE=0) T (SSOE=1) H • Otherwise*16 Keep	• CAS output*15 T • Otherwise*16 Keep	• CAS ou $\overline{UCAS}$, • Otherw I/O por
	6, 7	T	T	Keep	—	I/O port
PB_7 to PB_4	1 to 7	T	T	Keep	Keep	I/O port
$\overline{RESO}$*1	—	T*1	T	T	T*1	T

Legend
H: High
L: Low
T: High-impedance state
Keep: Input pins are in the high-impedance state; output pins maintain their previous s
DDR: Data direction register

Notes:
1. When bits DRAS2, DRAS1, and DRAS0 in DRCRA (DRAM control registe cleared to 0.
2. When any of bits DRAS2, DRAS1, or DRAS0 in DRCRA (DRAM control re set to 1.
3. When the setting of bits DRAS2, DRAS1, and DRAS0 in DRCRA (DRAM register A) is 010, 100, or 101.
4. When the setting of bits DRAS2, DRAS1, and DRAS0 in DRCRA (DRAM register A) is other than 010, 100, 101, or 000.
5. When bit A23E, A22E, or A21E, respectively, in BRCR (bus release contro cleared to 0.
6. When bit A23E, A22E, or A21E, respectively, in BRCR (bus release contro set to 1.
7. When bit CS7E or CS6E, respectively, in CSCR (chip select control regist 1.

12. When the setting of bits DRAS2, DRAS1, and DRAS0 in DRCRA (DRAM co register A) is 100, 101, or 110.
13. When the setting of bits DRAS2, DRAS1, and DRAS0 in DRCRA (DRAM co register A) is other than 100, 101, or 110, and bit CS4E in CSCR (chip sele register) is set to 1.
14. When the setting of bits DRAS2, DRAS1, and DRAS0 in DRCRA (DRAM co register A) is other than 100, 101, or 110, and bit CS4E in CSCR (chip sele register) is cleared to 0.
15. When any of bits DRAS2, DRAS1, or DRAS0 in DRCRA (DRAM control reg set to 1, and bit CSEL in DRCRB (DRAM control register B) is cleared to 0.
16. When any of bits DRAS2, DRAS1, or DRAS0 in DRCRA (DRAM control reg set to 1, and bit CSEL in DRCRB (DRAM control register B) is set to 1; or, v DRAS2, DRAS1, and DRAS0 are all cleared to 0.

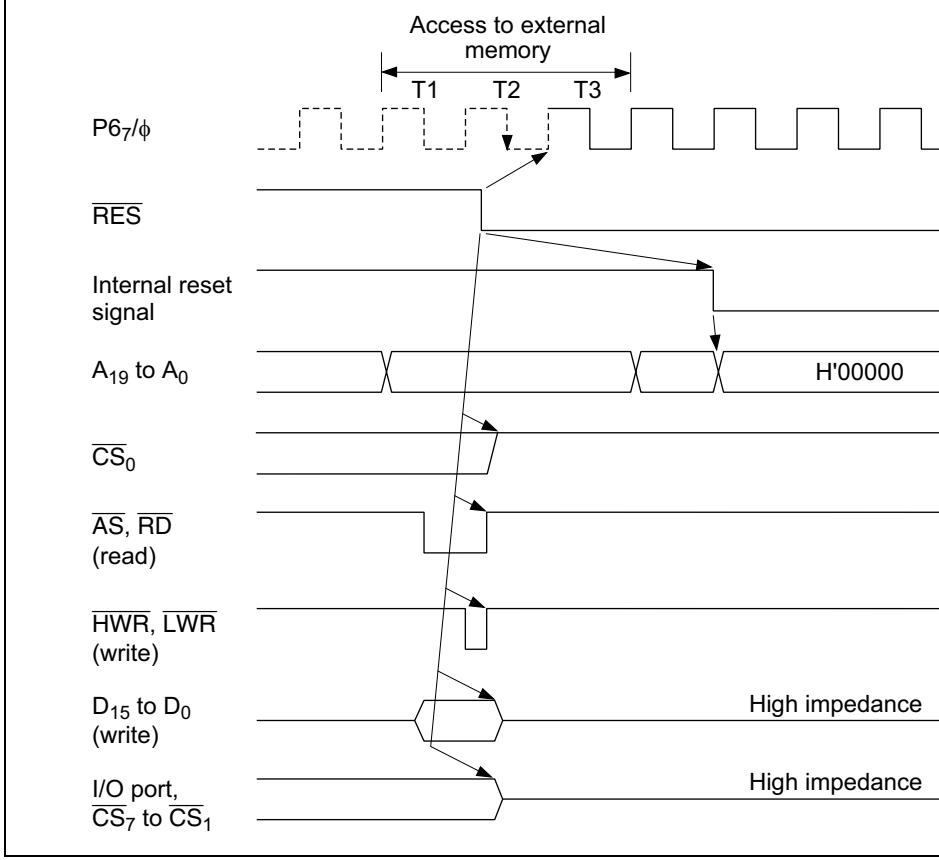


Figure D.1 Reset during Memory Access (Modes 1 and 2)

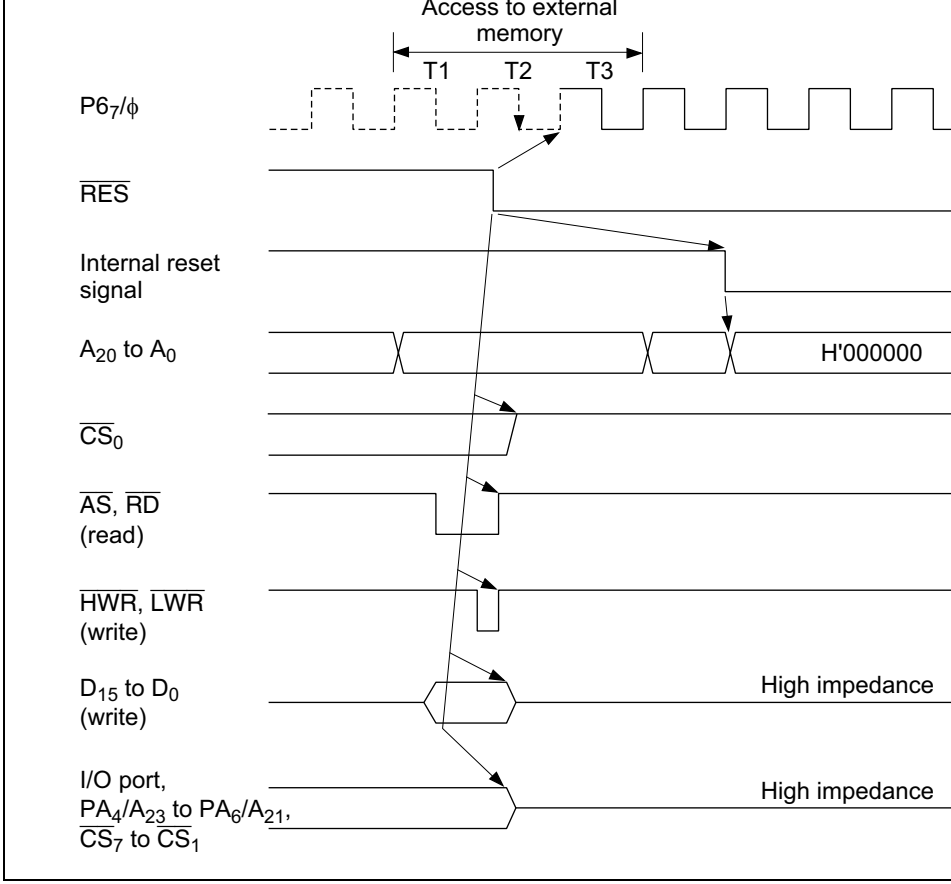


Figure D.2 Reset during Memory Access (Modes 3 and 4)

Mode 5: Figure D.3 is a timing diagram for the case in which $\overline{RES}$ goes low during an memory access in mode 5. As soon as $\overline{RES}$ goes low, all ports are initialized to the inpu $\overline{RD}$, $\overline{HWR}$, and $\overline{LWR}$ go high, and the address bus and D_{15} to D_0 go to the high-impeda Clock pin P6$_7$/ϕ goes to the output state at the next rise of ϕ after $\overline{RES}$ goes low.

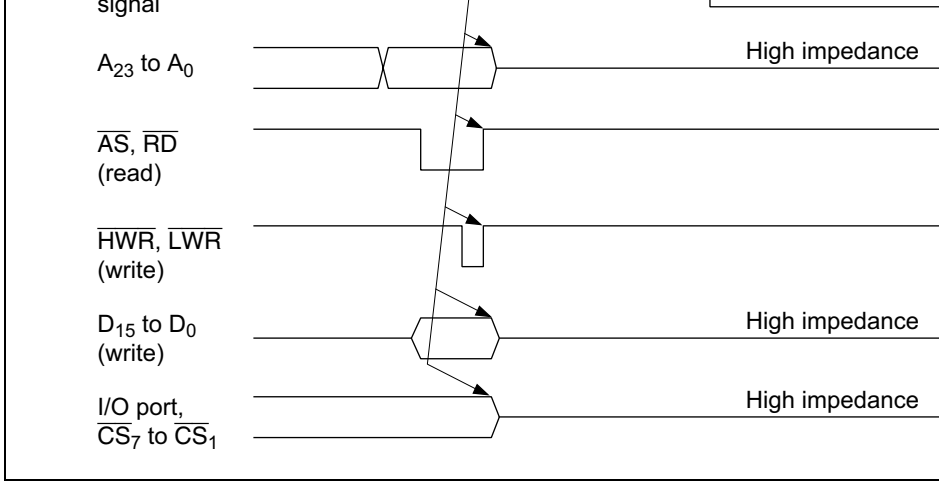


Figure D.3 Reset during Memory Access (Mode 5)

Modes 6 and 7: Figure D.4 is a timing diagram for the case in which $\overline{RES}$ goes low d operation in mode 6 or 7. As soon as $\overline{RES}$ goes low, all ports are initialized to the inpu Clock pin $P6_7/\phi$ goes to the output state at the next rise of ϕ after $\overline{RES}$ goes low.

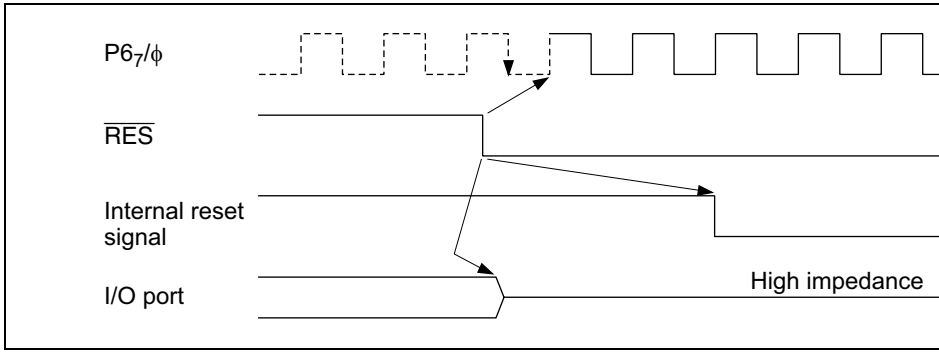
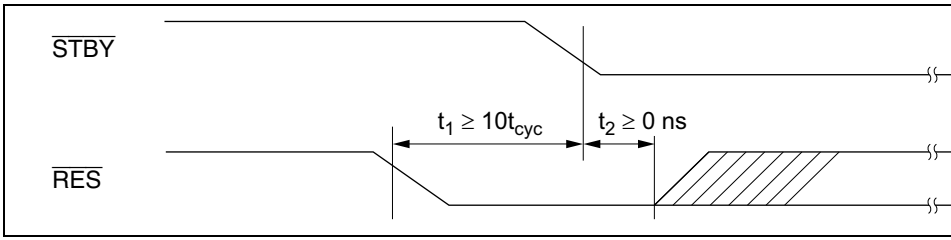
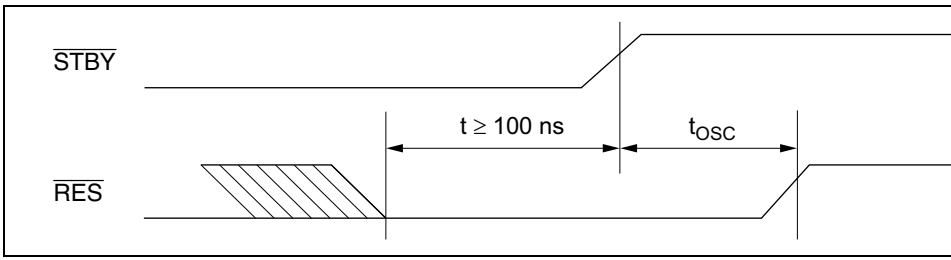


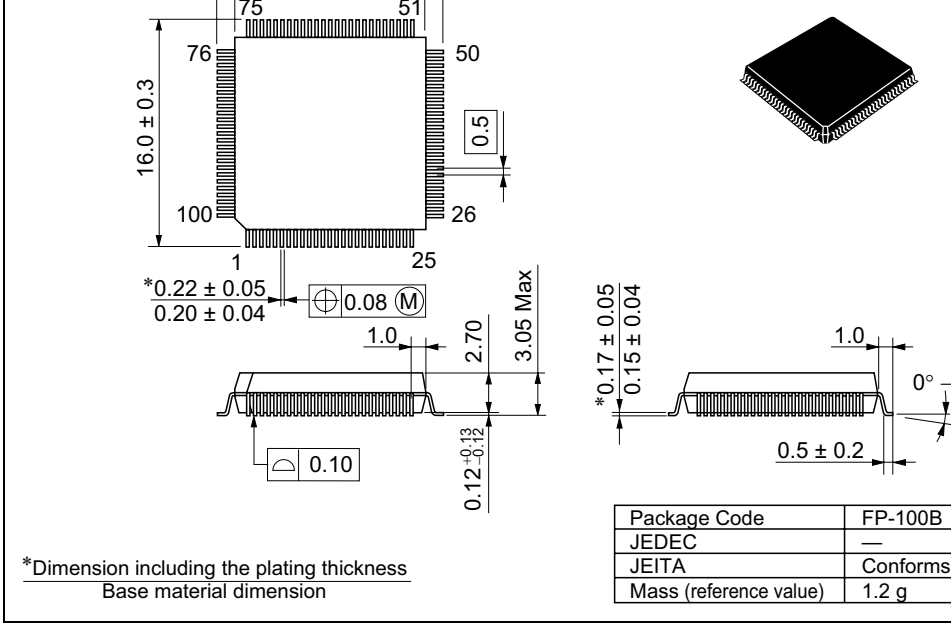
Figure D.4 Reset during Operation (Modes 6 and 7)



2. To retain RAM contents with the RAME bit cleared to 0 in SYSCR, RES does not l
   driven low as in (1).

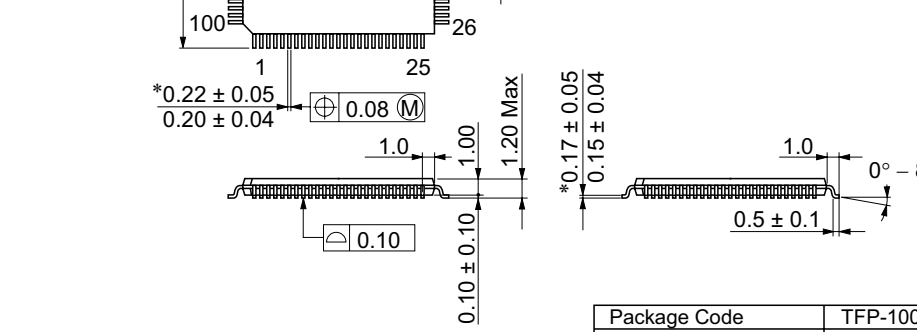
Timing of Recovery from Hardware Standby Mode: Drive the RES signal low appr
100 ns before STBY goes high.





Package Code	FP-100B
JEDEC	—
JEITA	Conforms
Mass (reference value)	1.2 g

Figure G.1 Package Dimensions (FP-100B)



Package Code	TFP-100
JEDEC	—
JEITA	Conform
Mass (reference value)	0.5 g

Figure G.2 Package Dimensions (TFP-100B)

1	Operating mode	Mode 5	16 Mbytes ROM enabled expanded mode	16 Mbytes ROM enabled expanded mode	1 e e m
		Mode 6	64 kbytes single-chip mode	64 kbytes single-chip mode	1 I e m
2	Interrupt controller	Internal interrupt sources	36	36 (H8/3067) 27 (H8/3024)	3
3	Bus controller	Burst ROM interface	Yes	Yes (H8/3067) No (H8/3024)	I
		Idle cycle insertion function	Yes	Yes	I
		Wait mode	2 modes	2 modes	4
		Wait state number setting	Per area	Per area	C t
		Address output method	Choice of address update fixed	Choice of address update mode (H8/3024 Group)	I
4	DRAM interface	Connectable areas	Area 2/3/4/5	Area 2/3/4/5 (H8/3067 only)	A
		Precharge cycle insertion function	Yes	Yes (H8/3067 only)	I
		Fast page mode	Yes	Yes (H8/3067 only)	I
		Address shift amount	8 bit/9 bit/10 bit	8 bit/9 bit/10 bit (H8/3067 only)	8

		External clock	4 systems (selectable)	4 systems (fixed)	4 systems (selectable)	4 systems (fixed)
		Internal clock	ϕ, ϕ/2, ϕ/4, ϕ/8	ϕ/8, ϕ/64, ϕ/8192	ϕ, ϕ/2, ϕ/4, ϕ/8	ϕ/8, ϕ/64, ϕ/8192
		Complementary PWM function	No	No	No	No
		Reset-synchronous PWM function	No	No	No	No
		Buffer operation	No	No	No	No
		Output initialization function	Yes	No	Yes	No
		PWM output	3	4 (2)	3	4 (2)
		DMAC activation	3 channels	No	3 channels (H8/3067 only)	No
		A/D conversion activation	No	Yes	No	Yes
		Interrupt sources	3 sources × 3	8 sources	3 sources × 3	8 sources
6	TPC	Time base	3 kinds, 16-bit timer base		3 kinds, 16-bit timer base	
7	WDT	Reset signal external output function	Yes (but not present in the flash memory version)		Yes (except products with on-chip flash memory)	
8	SCI	Number of channels	3 channels		3 channels (H8/3067) 2 channels (H8/3024)	
		Smart card interface	Supported on all channels		Supported on all channels	

		Address bus, $\overline{AS}$, $\overline{RD}$, $\overline{HWR}$, $\overline{LWR}$, $\overline{CS_7}$–$\overline{CS_0}$, $\overline{RFSH}$ in software standby state	High-level output/high-impedance selectable	High-level output/high-impedance selectable ($\overline{RFSH}$: H8/3067 only)	H c (L c
		$\overline{CS_7}$–$\overline{CS_0}$ in bus-released state	High-impedance	High-impedance	H c
11	Flash memory functions	Program/erase voltage	12 V application unnecessary. Single-power-supply programming.	12 V application unnecessary. Single-power-supply programming.	1 a f
		Block divisions	14 blocks	8 blocks	1

3	$PB_1/TP_9/TMIO_1/\overline{DREQ_0}/\overline{CS_6}$	$PB_1/TP_9/TMIO_1/\overline{DREQ_0}/\overline{CS_6}$	$PB_1/TP_9/TMIO_1/\overline{CS_6}$	$PB_1/TP_9/TIOCB_3$	$PB_1/$ TIOC
4	$PB_2/TP_{10}/TMO_2/\overline{CS_5}$	$PB_2/TP_{10}/TMO_2/\overline{CS_5}$	$PB_2/TP_{10}/TMO_2/\overline{CS_5}$	$PB_2/TP_{10}/TIOCA_4$	$PB_2/$ TIOC
5	$PB_3/TP_{11}/TMIO_3/\overline{DREQ_1}/\overline{CS_4}$	$PB_3/TP_{11}/TMIO_3/\overline{DREQ_1}/\overline{CS_4}$	$PB_3/TP_{11}/TMIO_3/\overline{CS_4}$	$PB_3/TP_{11}/TIOCB_4$	$PB_3/$
6	$PB_4/TP_{12}/\overline{UCAS}$	$PB_4/TP_{12}/\overline{UCAS}$	PB_4/TP_{12}	$PB_4/TP_{12}/TOCXA_4$	$PB_4/$
7	$PB_5/TP_{13}/\overline{LCAS}/SCK_2$	$PB_5/TP_{13}/\overline{LCAS}/SCK_2$	PB_5/TP_{13}	$PB_5/TP_{13}/TOCXB_4$	$PB_5/$
8	$PB_6/TP_{14}/TxD_2$	$PB_6/TP_{14}/TxD_2$	PB_6/TP_{14}	$PB_6/TP_{14}/\overline{DREQ_0}/\overline{CS_7}$	$PB_6/$
9	$PB_7/TP_{15}/RxD_2$	$PB_7/TP_{15}/RxD_2$	PB_7/TP_{15}	$PB_7/TP_{15}/\overline{DREQ_1}/\overline{ADTRG}$	$PB_7/$ ADT
10	$\overline{RESO}$/FWE*	$\overline{RESO}$/FWE*	$\overline{RESO}$/FWE*	$\overline{RESO}/V_{PP}$*	$\overline{RES}$
11	Vss	Vss	Vss	Vss	Vss
12	$P9_0/TxD_0$	$P9_0/TxD_0$	$P9_0/TxD_0$	$P9_0/TxD_0$	$P9_0/$
13	$P9_1/TxD_1$	$P9_1/TxD_1$	$P9_1/TxD_1$	$P9_1/TxD_1$	$P9_1/$
14	$P9_2/RxD_0$	$P9_2/RxD_0$	$P9_2/RxD_0$	$P9_2/RxD_0$	$P9_2/$
15	$P9_3/RxD_1$	$P9_3/RxD_1$	$P9_3/RxD_1$	$P9_3/RxD_1$	$P9_3/$
16	$P9_4/SCK_0/\overline{IRQ_4}$	$P9_4/SCK_0/\overline{IRQ_4}$	$P9_4/SCK_0/\overline{IRQ_4}$	$P9_4/SCK_0/\overline{IRQ_4}$	$P9_4/$
17	$P9_5/SCK_1/\overline{IRQ_5}$	$P9_5/SCK_1/\overline{IRQ_5}$	$P9_5/SCK_1/\overline{IRQ_5}$	$P9_5/SCK_1/\overline{IRQ_5}$	$P9_5/$
18	$P4_0/D_0$	$P4_0/D_0$	$P4_0/D_0$	$P4_0/D_0$	$P4_0/$
19	$P4_1/D_1$	$P4_1/D_1$	$P4_1/D_1$	$P4_1/D_1$	$P4_1/$
20	$P4_2/D_2$	$P4_2/D_2$	$P4_2/D_2$	$P4_2/D_2$	$P4_2/$
21	$P4_3/D_3$	$P4_3/D_3$	$P4_3/D_3$	$P4_3/D_3$	$P4_3/$
22	Vss	Vss	Vss	Vss	Vss
23	$P4_4/D_4$	$P4_4/D_4$	$P4_4/D_4$	$P4_4/D_4$	$P4_4/$
24	$P4_5/D_5$	$P4_5/D_5$	$P4_5/D_5$	$P4_5/D_5$	$P4_5/$
25	$P4_6/D_6$	$P4_6/D_6$	$P4_6/D_6$	$P4_6/D_6$	$P4_6/$
26	$P4_7/D_7$	$P4_7/D_7$	$P4_7/D_7$	$P4_7/D_7$	$P4_7/$
27	$P3_0/D_8$	$P3_0/D_8$	$P3_0/D_8$	$P3_0/D_8$	$P3_0/$
28	$P3_1/D_9$	$P3_1/D_9$	$P3_1/D_9$	$P3_1/D_9$	$P3_1/$

36	$P1_0/A_0$	$P1_0/A_0$	$P1_0/A_0$	$P1_0/A_0$	$P1_0/A_0$
37	$P1_1/A_1$	$P1_1/A_1$	$P1_1/A_1$	$P1_1/A_1$	$P1_1/A_1$
38	$P1_2/A_2$	$P1_2/A_2$	$P1_2/A_2$	$P1_2/A_2$	$P1_2/A_2$
39	$P1_3/A_3$	$P1_3/A_3$	$P1_3/A_3$	$P1_3/A_3$	$P1_3/A_3$
40	$P1_4/A_4$	$P1_4/A_4$	$P1_4/A_4$	$P1_4/A_4$	$P1_4/A_4$
41	$P1_5/A_5$	$P1_5/A_5$	$P1_5/A_5$	$P1_5/A_5$	$P1_5/A_5$
42	$P1_6/A_6$	$P1_6/A_6$	$P1_6/A_6$	$P1_6/A_6$	$P1_6/A_6$
43	$P1_7/A_7$	$P1_7/A_7$	$P1_7/A_7$	$P1_7/A_7$	$P1_7/A_7$
44	Vss	Vss	Vss	Vss	Vss
45	$P2_0/A_8$	$P2_0/A_8$	$P2_0/A_8$	$P2_0/A_8$	$P2_0/A_8$
46	$P2_1/A_9$	$P2_1/A_9$	$P2_1/A_9$	$P2_1/A_9$	$P2_1/A_9$
47	$P2_2/A_{10}$	$P2_2/A_{10}$	$P2_2/A_{10}$	$P2_2/A_{10}$	$P2_2/A_{10}$
48	$P2_3/A_{11}$	$P2_3/A_{11}$	$P2_3/A_{11}$	$P2_3/A_{11}$	$P2_3/A_{11}$
49	$P2_4/A_{12}$	$P2_4/A_{12}$	$P2_4/A_{12}$	$P2_4/A_{12}$	$P2_4/A_{12}$
50	$P2_5/A_{13}$	$P2_5/A_{13}$	$P2_5/A_{13}$	$P2_5/A_{13}$	$P2_5/A_{13}$
51	$P2_6/A_{14}$	$P2_6/A_{14}$	$P2_6/A_{14}$	$P2_6/A_{14}$	$P2_6/A_{14}$
52	$P2_7/A_{15}$	$P2_7/A_{15}$	$P2_7/A_{15}$	$P2_7/A_{15}$	$P2_7/A_{15}$
53	$P5_0/A_{16}$	$P5_0/A_{16}$	$P5_0/A_{16}$	$P5_0/A_{16}$	$P5_0/A_{16}$
54	$P5_1/A_{17}$	$P5_1/A_{17}$	$P5_1/A_{17}$	$P5_1/A_{17}$	$P5_1/A_{17}$
55	$P5_2/A_{18}$	$P5_2/A_{18}$	$P5_2/A_{18}$	$P5_2/A_{18}$	$P5_2/A_{18}$
56	$P5_3/A_{19}$	$P5_3/A_{19}$	$P5_3/A_{19}$	$P5_3/A_{19}$	$P5_3/A_{19}$
57	Vss	Vss	Vss	Vss	Vss
58	$P6_0/\overline{WAIT}$	$P6_0/\overline{WAIT}$	$P6_0/\overline{WAIT}$	$P6_0/\overline{WAIT}$	$P6_0/\overline{WAIT}$
59	$P6_1/\overline{BREQ}$	$P6_1/\overline{BREQ}$	$P6_1/\overline{BREQ}$	$P6_1/\overline{BREQ}$	$P6_1/\overline{BREQ}$
60	$P6_2/\overline{BACK}$	$P6_2/\overline{BACK}$	$P6_2/\overline{BACK}$	$P6_2/\overline{BACK}$	$P6_2/\overline{BACK}$
61	$P6_7/\phi$	$P6_7/\phi$	$P6_7/\phi$	ϕ	ϕ
62	$\overline{STBY}$	$\overline{STBY}$	$\overline{STBY}$	$\overline{STBY}$	$\overline{STBY}$
63	$\overline{RES}$	$\overline{RES}$	$\overline{RES}$	$\overline{RES}$	$\overline{RES}$
64	NMI	NMI	NMI	NMI	NMI
65	Vss	Vss	Vss	Vss	Vss

73	MD_0	MD_0	MD_0	MD_0	MD_0
74	MD_1	MD_1	MD_1	MD_1	MD_1
75	MD_2	MD_2	MD_2	MD_2	MD_2
76	AVcc	AVcc	AVcc	AVcc	AVcc
77	V_{REF}	V_{REF}	V_{REF}	V_{REF}	V_{REF}
78	$P7_0/AN_0$	$P7_0/AN_0$	$P7_0/AN_0$	$P7_0/AN_0$	$P7_0/A$
79	$P7_1/AN_1$	$P7_1/AN_1$	$P7_1/AN_1$	$P7_1/AN_1$	$P7_1/A$
80	$P7_2/AN_2$	$P7_2/AN_2$	$P7_2/AN_2$	$P7_2/AN_2$	$P7_2/A$
81	$P7_3/AN_3$	$P7_3/AN_3$	$P7_3/AN_3$	$P7_3/AN_3$	$P7_3/A$
82	$P7_4/AN_4$	$P7_4/AN_4$	$P7_4/AN_4$	$P7_4/AN_4$	$P7_4/A$
83	$P7_5/AN_5$	$P7_5/AN_5$	$P7_5/AN_5$	$P7_5/AN_5$	$P7_5/A$
84	$P7_6/AN_6/DA_0$	$P7_6/AN_6/DA_0$	$P7_6/AN_6/DA_0$	$P7_6/AN_6/DA_0$	$P7_6/A$
85	$P7_7/AN_7/DA_1$	$P7_7/AN_7/DA_1$	$P7_7/AN_7/DA_1$	$P7_7/AN_7/DA_1$	$P7_7/A$
86	AVss	AVss	AVss	AVss	AVss
87	$P8_0/\overline{RFSH}/\overline{IRQ}_0$	$P8_0/\overline{RFSH}/\overline{IRQ}_0$	$P8_0/\overline{IRQ}_0$	$P8_0/\overline{RFSH}/\overline{IRQ}_0$	$P8_0/\overline{I}$
88	$P8_1/\overline{CS}_3/\overline{IRQ}_1$	$P8_1/\overline{CS}_3/\overline{IRQ}_1$	$P8_1/\overline{CS}_3/\overline{IRQ}_1$	$P8_1/\overline{CS}_3/\overline{IRQ}_1$	$P8_1/\overline{C}$
89	$P8_2/\overline{CS}_2/\overline{IRQ}_2$	$P8_2/\overline{CS}_2/\overline{IRQ}_2$	$P8_2/\overline{CS}_2/\overline{IRQ}_2$	$P8_2/\overline{CS}_2/\overline{IRQ}_2$	$P8_2/\overline{C}$
90	$P8_3/\overline{CS}_1/\overline{IRQ}_3/$ $\overline{ADTRG}$	$P8_3/\overline{CS}_1/\overline{IRQ}_3/$ $\overline{ADTRG}$	$P8_3/\overline{CS}_1/\overline{IRQ}_3/$ $\overline{ADTRG}$	$P8_3/\overline{CS}_1/\overline{IRQ}_3$	$P8_3/\overline{C}$
91	$P8_4/\overline{CS}_0$	$P8_4/\overline{CS}_0$	$P8_4/\overline{CS}_0$	$P8_4/\overline{CS}_0$	$P8_4/\overline{C}$
92	Vss	Vss	Vss	Vss	Vss
93	$PA_0/TP_0/\overline{TEND}_0/$ TCLKA	$PA_0/TP_0/\overline{TEND}_0/$ TCLKA	$PA_0/TP_0/TCLKA$	$PA_0/TP_0/\overline{TEND}_0/$ TCLKA	$PA_0/$ TCL
94	$PA_1/TP_1/\overline{TEND}_1/$ TCLKB	$PA_1/TP_1/\overline{TEND}_1/$ TCLKB	$PA_1/TP_1/TCLKB$	$PA_1/TP_1/\overline{TEND}_1/$ TCLKB	$PA_1/$ TCL
95	$PA_2/TP_2/TIOCA_0/$ TCLKC	$PA_2/TP_2/TIOCA_0/$ TCLKC	$PA_2/TP_2/TIOCA_0/$ TCLKC	$PA_2/TP_2/TIOCA_0/$ TCLKC	$PA_2/$ TCL
96	$PA_3/TP_3/TIOCB_0/$ TCLKD	$PA_3/TP_3/TIOCB_0/$ TCLKD	$PA_3/TP_3/TIOCB_0/$ TCLKD	$PA_3/TP_3/TIOCB_0/$ TCLKD	$PA_3/$ TCL

Note: * Functions as RESO in the mask ROM versions, and as FWE in the flash memor

H8/3028, H8/3028F-ZTAT™ Group Hardware Manual

Publication Date: 1st Edition, September 2002
Rev.2.00, September 19, 2003
Published by: Sales Strategic Planning Div.
Renesas Technology Corp.
Edited by: Technical Documentation & Information Department
Renesas Kodaira Semiconductor Co., Ltd.

Renesas Technology Corp. Sales Strategic Planning Div. Nippon Bldg., 2-6-2, Ohte-machi, Chiyoda-ku, Tokyo 100-000

RENESAS SALES OFFICES

http://www.

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, United Kingdom
Tel: <44> (1628) 585 100, Fax: <44> (1628) 585 900

Renesas Technology Europe GmbH
Dornacher Str. 3, D-85622 Feldkirchen, Germany
Tel: <49> (89) 380 70 0, Fax: <49> (89) 929 30 11

Renesas Technology Hong Kong Ltd.
7/F., North Tower, World Finance Centre, Harbour City, Canton Road, Hong Kong
Tel: <852> 2265-6688, Fax: <852> 2375-6836

Renesas Technology Taiwan Co., Ltd.
FL 10, #99, Fu-Hsing N. Rd., Taipei, Taiwan
Tel: <886> (2) 2715-2888, Fax: <886> (2) 2713-2999

Renesas Technology (Shanghai) Co., Ltd.
26/F., Ruijin Building, No.205 Maoming Road (S), Shanghai 200020, China
Tel: <86> (21) 6472-1001, Fax: <86> (21) 6415-2952

Renesas Technology Singapore Pte. Ltd.
1, Harbour Front Avenue, #06-10, Keppel Bay Tower, Singapore 098632
Tel: <65> 6213-0200, Fax: <65> 6278-8001

H8/3028, H8/3028F-ZTAT™ Group
Hardware Manual

Renesas Electronics Corporation
1753, Shimonumabe, Nakahara-ku, Kawasaki-shi, Kanagawa 211-8668 Japan

REJ09B0083-0200O

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 :

M30302FCPFP#U3 MB90F036APMC-GSE1 MB90F428GCPFR-GSE1 MB96F683RBPMC-GSAE1 R5F10MMGDFB#30 R5F111PGGFB#30 R5F117BCGNA#20 DF3026XBL25V DF36014GFTV DF36014GFXV DF36034GFPV R5F11B7EANA#U0 R5F21172DSP#U0 MB90092PF-G-BNDE1 MB90F335APMC1-G-SPE1 MB90F345CAPFR-GSE1 MB90F568PMCR-GE1 MB96F395RSAPMC-GSE2 DF36024GFXV UPD78F1018F1-BA4-A MB96F018RBPMC-GSE1 MB90F867ASPFR-GE1 M30290FCHP#U3A DF2239FA20IV R5F117BCGFP#30 LC88F58B0AU-SQFPH MB90F548GPF-GE1 MB90214PF-GT-310-BND-AE1 MB90F342CESPQC-GSE2 MB90F428GAPF-GSE1 ML620Q504H-NNNTBWBX S912ZVH128F2VLL UPD78F1500AGK-GAK-AX HD64F3337SF16V MB90F428GCPF-GSE1 MB90F342ESPMC-G-JNE1 MB90022PF-GS-358E1 MB96F395RWAPMC-GSE2 MB96395RSAPMC-GS-110E2 MB90F883CSPMC-GE1 S912ZVHY64F1CLL S912ZVHY64F1VLQ ST10F280 MB96F338RSAPMCR-GK5E2 CY90096PF-G-002-BND-ERE1 ML62Q1569-NNNGAZ0AX ML62Q1739-NNNGAZ0AX ML62Q1749-NNNGAZ0AX ML62Q1579-NNNGAZ0AX ML62Q1559-NNNGAZ0AX